To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

16

R8C/18 Group, R8C/19 Group

Hardware Manual

RENESAS 16-BIT SINGLE-CHIP MCU
R8C FAMILY / R8C/1x 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 information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.1.30 2006.04

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

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

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

2. Processing at Power-on

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

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

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

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual.
The manual comprises an overview of the product; descriptions of the CPU, system control functions, peripheral functions, and electrical characteristics; and usage notes.

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

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

The following documents apply to the R8C/18 Group, R8C/19 Group. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Technology Web site.

Document Type	Description	Document Title	Document No.
Datasheet	Hardware overview and electrical characteristics	R8C/18 Group, R8C/19 Group Datasheet	REJ03B0124
Hardware manual	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	R8C/18 Group, R8C/19 Group Hardware Manual	This hardware manual
Software manual	Description of CPU instruction set	R8C/Tiny Series Software Manual	REJ09B0001
Application note	Information on using peripheral functions and application examples Sample programs Information on writing programs in assembly language and C	Available from Renesas Technology Web site.	
Renesas technical update	Product specifications, updates on documents, etc.		

2. Notation of Numbers and Symbols

The notation conventions for register names, bit names, numbers, and symbols used in this manual are described below.

(1) Register Names, Bit Names, and Pin Names

Registers, bits, and pins are referred to in the text by symbols. The symbol is accompanied by the word “register,” “bit,” or “pin” to distinguish the three categories.

Examples the PM03 bit in the PM0 register
P3_5 pin, VCC pin

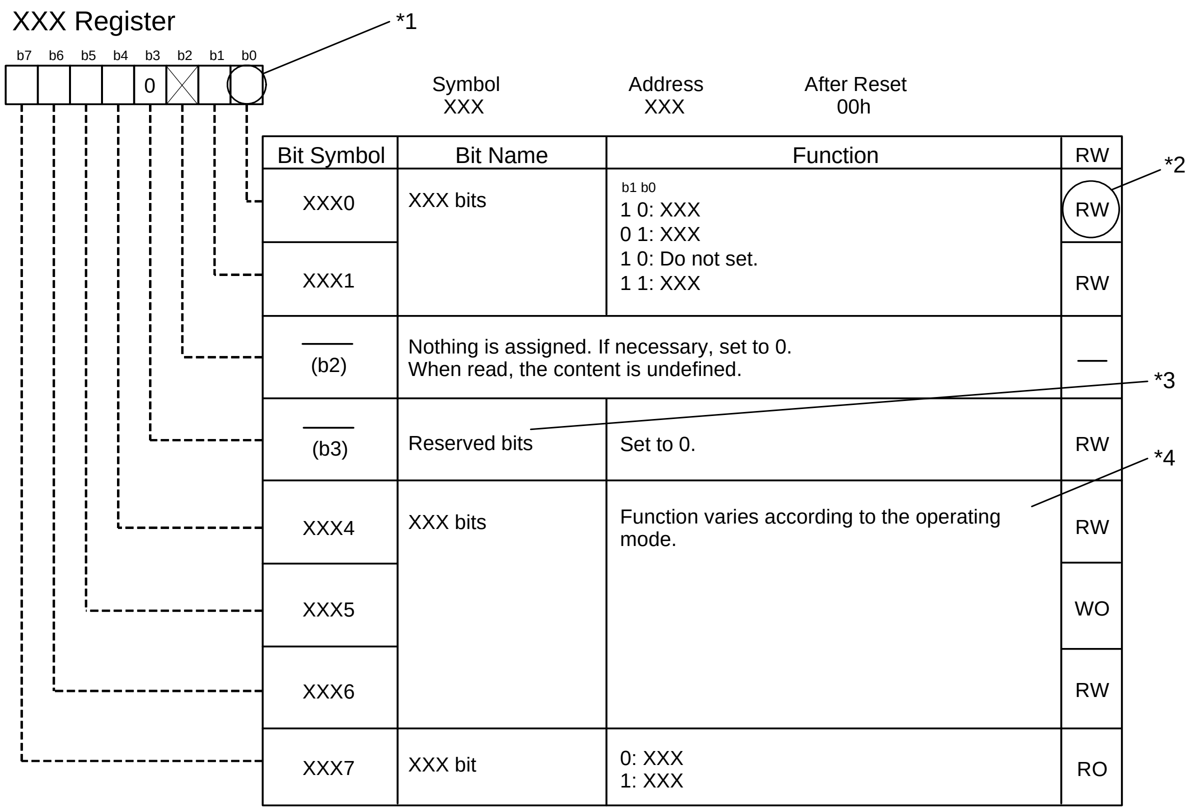
(2) Notation of Numbers

The indication “b” is appended to numeric values given in binary format. However, nothing is appended to the values of single bits. The indication “h” is appended to numeric values given in hexadecimal format. Nothing is appended to numeric values given in decimal format.

Examples Binary: 11b
Hexadecimal: EFA0h
Decimal: 1234

3. Register Notation

The symbols and terms used in register diagrams are described below.



Symbol	Address	After Reset
XXX	XXX	00h

Bit Symbol	Bit Name	Function	RW
XXX0	XXX bits	b1 b0 1 0: XXX 0 1: XXX 1 0: Do not set. 1 1: XXX	RW
XXX1			RW
— (b2)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		—
— (b3)	Reserved bits	Set to 0.	RW
XXX4	XXX bits	Function varies according to the operating mode.	RW
XXX5			WO
XXX6			RW
XXX7	XXX bit	0: XXX 1: XXX	RO

*1

Blank: Set to 0 or 1 according to the application.
0: Set to 0.
1: Set to 1.
X: Nothing is assigned.

*2

RW: Read and write.
RO: Read only.
WO: Write only.
–: Nothing is assigned.

*3

- Reserved bit
  Reserved bit. Set to specified value.

*4

- Nothing is assigned
  Nothing is assigned to the bit. As the bit may be used for future functions, if necessary, set to 0.
- Do not set to a value
  Operation is not guaranteed when a value is set.
- Function varies according to the operating mode.
  The function of the bit varies with the peripheral function mode. Refer to the register diagram for information on the individual modes.

4. List of Abbreviations and Acronyms

Abbreviation	Full Form
ACIA	Asynchronous Communication 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
SFR	Special Function Registers
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

Table of Contents

SFR Page Reference

Address	Register	Symbol	Page
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	51
0005h	Processor Mode Register 1	PM1	51
0006h	System Clock Control Register 0	CM0	55
0007h	System Clock Control Register 1	CM1	56
0008h			
0009h	Address Match Interrupt Enable Register	AIER	93
000Ah	Protect Register	PRCR	71
000Bh			
000Ch	Oscillation Stop Detection Register	OCD	57
000Dh	Watchdog Timer Reset Register	WDTR	99
000Eh	Watchdog Timer Start Register	WDTS	99
000Fh	Watchdog Timer Control Register	WDC	98
0010h	Address Match Interrupt Register 0	RMAD0	93
0011h			
0012h			
0013h			
0014h	Address Match Interrupt Register 1	RMAD1	93
0015h			
0016h			
0017h			
0018h			
0019h			
001Ah			
001Bh			
001Ch	Count Source Protection Mode Register	CSPR	99
001Dh			
001Eh	$\overline{\text{INT0}}$ Input Filter Select Register	INT0F	85
001Fh			
0020h	High-Speed On-Chip Oscillator Control Register 0	HRA0	58
0021h	High-Speed On-Chip Oscillator Control Register 1	HRA1	59
0022h	High-Speed On-Chip Oscillator Control Register 2	HRA2	59
0023h			
0024h			
0025h			
0026h			
0027h			
0028h			
0029h			
002Ah			
002Bh			
002Ch			
002Dh			
002Eh			
002Fh			
0030h			
0031h	Voltage Detection Register 1	VCA1	43
0032h	Voltage Detection Register 2	VCA2	43
0033h			
0034h			
0035h			
0036h	Voltage Monitor 1 Circuit Control Register	VW1C	44
0037h	Voltage Monitor 2 Circuit Control Register	VW2C	45
0038h			
0039h			
003Ah			
003Bh			
003Ch			
003Dh			
003Eh			
003Fh			

NOTE:

1. The blank regions are reserved. Do not access locations in these regions.

Address	Register	Symbol	Page
0040h			
0041h			
0042h			
0043h			
0044h			
0045h			
0046h			
0047h			
0048h			
0049h			
004Ah			
004Bh			
004Ch			
004Dh	Key Input Interrupt Control Register	KUPIC	77
004Eh	Comparator Conversion Interrupt Control Register	ADIC	77
004Fh			
0050h	Compare 1 Interrupt Control Register	CMP1IC	77
0051h	UART0 Transmit Interrupt Control Register	S0TIC	77
0052h	UART0 Receive Interrupt Control Register	S0RIC	77
0053h	UART1 Transmit Interrupt Control Register	S1TIC	77
0054h	UART1 Receive Interrupt Control Register	S1RIC	77
0055h			
0056h	Timer X Interrupt Control Register	TXIC	77
0057h			
0058h	Timer Z Interrupt Control Register	TZIC	77
0059h	$\overline{\text{INT1}}$ Interrupt Control Register	INT1IC	77
005Ah	$\overline{\text{INT3}}$ Interrupt Control Register	INT3IC	77
005Bh	Timer C Interrupt Control Register	TCIC	77
005Ch	Compare 0 Interrupt Control Register	CMP0IC	77
005Dh	$\overline{\text{INT0}}$ Interrupt Control Register	INT0IC	78
005Eh			
005Fh			
0060h			
0061h			
0062h			
0063h			
0064h			
0065h			
0066h			
0067h			
0068h			
0069h			
006Ah			
006Bh			
006Ch			
006Dh			
006Eh			
006Fh			
0070h			
0071h			
0072h			
0073h			
0074h			
0075h			
0076h			
0077h			
0078h			
0079h			
007Ah			
007Bh			
007Ch			
007Dh			
007Eh			
007Fh			

Address	Register	Symbol	Page
0080h	Timer Z Mode Register	TZMR	118
0081h			
0082h			
0083h			
0084h	Timer Z Waveform Output Control Register	PUM	120
0085h	Prescaler Z Register	PREZ	119
0086h	Timer Z Secondary Register	TZSC	119
0087h	Timer Z Primary Register	TZPR	119
0088h			
0089h			
008Ah	Timer Z Output Control Register	TZOC	120
008Bh	Timer X Mode Register	TXMR	104
008Ch	Prescaler X Register	PREX	105
008Dh	Timer X Register	TX	105
008Eh	Timer Count Source Set Register	TCSS	105,121
008Fh			
0090h	Timer C Register	TC	137
0091h			
0092h			
0093h			
0094h			
0095h			
0096h	External Input Enable Register	INTEN	85
0097h			
0098h	Key Input Enable Register	KIEN	91
0099h			
009Ah	Timer C Control Register 0	TCC0	138
009Bh	Timer C Control Register 1	TCC1	139
009Ch	Capture, Compare 0 Register	TM0	137
009Dh			
009Eh	Compare 1 Register	TM1	137
009Fh			
00A0h	UART0 Transmit/Receive Mode Register	U0MR	149
00A1h	UART0 Bit Rate Register	U0BRG	148
00A2h	UART0 Transmit Buffer Register	U0TB	148
00A3h			
00A4h	UART0 Transmit/Receive Control Register 0	U0C0	150
00A5h	UART0 Transmit/Receive Control Register 1	U0C1	151
00A6h	UART0 Receive Buffer Register	U0RB	148
00A7h			
00A8h	UART1 Transmit/Receive Mode Register	U1MR	149
00A9h	UART1 Bit Rate Register	U1BRG	148
00AAh	UART1 Transmit Buffer Register	U1TB	148
00ABh			
00ACh	UART1 Transmit/Receive Control Register 0	U1C0	150
00ADh	UART1 Transmit/Receive Control Register 1	U1C1	151
00AEh	UART1 Receive Buffer Register	U1RB	148
00AFh			
00B0h	UART Transmit/Receive Control Register 2	UCON	151
00B1h			
00B2h			
00B3h			
00B4h			
00B5h			
00B6h			
00B7h			
00B8h			
00B9h			
00BAh			
00BBh			
00BCh			
00BDh			
00BEh			
00BFh			

NOTE:

1. The blank regions, 0100h to 01AFh, and 01C0h to 02FFh are reserved. Do not access locations in these regions.

Address	Register	Symbol	Page
00C0h	A/D Register	AD	166
00C1h			
00C2h			
00C3h			
00C4h			
00C5h			
00C6h			
00C7h			
00C8h			
00C9h			
00CAh			
00CBh			
00CCh			
00CDh			
00CEh			
00CFh			
00D0h			
00D1h			
00D2h			
00D3h			
00D4h	A/D Control Register 2	ADCON2	166
00D5h			
00D6h	A/D Control Register 0	ADCON0	165
00D7h	A/D Control Register 1	ADCON1	165
00D8h			
00D9h			
00DAh			
00DBh			
00DCh			
00DDh			
00DEh			
00DFh			
00E0h			
00E1h	Port P1 Register	P1	33
00E2h			
00E3h	Port P1 Direction Register	PD1	33
00E4h			
00E5h	Port P3 Register	P3	33
00E6h			
00E7h	Port P3 Direction Register	PD3	33
00E8h	Port P4 Register	P4	33
00E9h			
00EAh	Port P4 Direction Register	PD4	33
00EBh			
00ECh			
00EDh			
00EEh			
00EFh			
00F0h			
00F1h			
00F2h			
00F3h			
00F4h			
00F5h			
00F6h			
00F7h			
00F8h			
00F9h			
00FAh			
00FBh			
00FCh	Pull-Up Control Register 0	PUR0	34
00FDh	Pull-Up Control Register 1	PUR1	34
00FEh	Port P1 Drive Capacity Control Register	DRR	34
00FFh	Timer C Output Control Register	TCOUT	140
01B3h	Flash Memory Control Register 4	FMR4	184
01B4h			
01B5h	Flash Memory Control Register 1	FMR1	183
01B6h			
01B7h	Flash Memory Control Register 0	FMR0	182

0FFFFh	Optional Function Select Register	OFS	98,177

R8C/18 Group, R8C/19 Group

SINGLE-CHIP 16-BIT CMOS MCU

REJ09B0222-0130
Rev.1.30
Apr 14, 2006

1. Overview

These MCUs are fabricated using a high-performance silicon gate CMOS process, embedding the R8C/Tiny Series CPU core, and is packaged in a 20-pin molded-plastic LSSOP, SDIP or a 28-pin plastic molded-HWQFN. It implements sophisticated instructions for a high level of instruction efficiency. With 1 Mbyte of address space, they are capable of executing instructions at high speed.
Furthermore, the R8C/19 Group has on-chip data flash ROM (1 KB × 2 blocks).
The difference between the R8C/18 Group and R8C/19 Group is only the presence or absence of data flash ROM. Their peripheral functions are the same.

1.1 Applications

Electric household appliances, office equipment, housing equipment (sensors, security systems), general industrial equipment, audio equipment, etc.

1.2 Performance Overview

Table 1.1 outlines the Functions and Specifications for R8C/18 Group and Table 1.2 outlines the Functions and Specifications for R8C/19 Group.

Table 1.1 Functions and Specifications for R8C/18 Group

Item		Specification
CPU	Number of fundamental instructions	89 instructions
	Minimum instruction execution time	50 ns (f(XIN) = 20 MHz, VCC = 3.0 to 5.5 V) 100 ns (f(XIN) = 10 MHz, VCC = 2.7 to 5.5 V)
	Operation mode	Single-chip
	Address space	1 Mbyte
	Memory capacity	Refer to **Table 1.3 Product Information for R8C/18 Group**
Peripheral Functions	Ports	I/O ports: 13 pins (including LED drive port) Input port: 3 pins
	LED drive ports	I/O ports: 4 pins
	Timers	Timer X: 8 bits × 1 channel, timer Z: 8 bits × 1 channel (Each timer equipped with 8-bit prescaler) Timer C: 16 bits × 1 channel (Input capture and output compare circuits)
	Serial interfaces	1 channel Clock synchronous serial I/O, UART 1 channel UART
	Comparator	1-bit comparator: 1 circuit, 4 channels
	Watchdog timer	15 bits × 1 channel (with prescaler) Reset start selectable, count source protection mode
	Interrupts	Internal: 10 sources, External: 4 sources, Software: 4 sources, Priority levels: 7 levels
	Clock generation circuits	2 circuits • Main clock oscillation circuit (with on-chip feedback resistor) • On-chip oscillator (high speed, low speed) High-speed on-chip oscillator has frequency adjustment function
	Oscillation stop detection function	Main clock oscillation stop detection function
	Voltage detection circuit	On-chip
	Power-on reset circuit	On-chip

Electric Characteristics	Supply voltage	VCC = 3.0 to 5.5 V (f(XIN) = 20 MHz)
		VCC = 2.7 to 5.5 V (f(XIN) = 10 MHz)

	Current consumption	Typ. 9 mA (VCC = 5.0 V, f(XIN) = 20 MHz, comparator stopped) Typ. 5 mA (VCC = 3.0V, f(XIN) = 10 MHz, comparator stopped) Typ. 35 μA (VCC = 3.0 V, wait mode, peripheral clock off) Typ. 0.7 μA (VCC = 3.0 V, stop mode)
Flash Memory	Programming and erasure voltage	VCC = 2.7 to 5.5 V
	Programming and erasure endurance	100 times
Operating Ambient Temperature		-20 to 85°C -40 to 85°C (D version)
Package		20-pin molded-plastic LSSOP
		20-pin molded-plastic SDIP
		28-pin molded-plastic HWQFN

Table 1.2 Functions and Specifications for R8C/19 Group

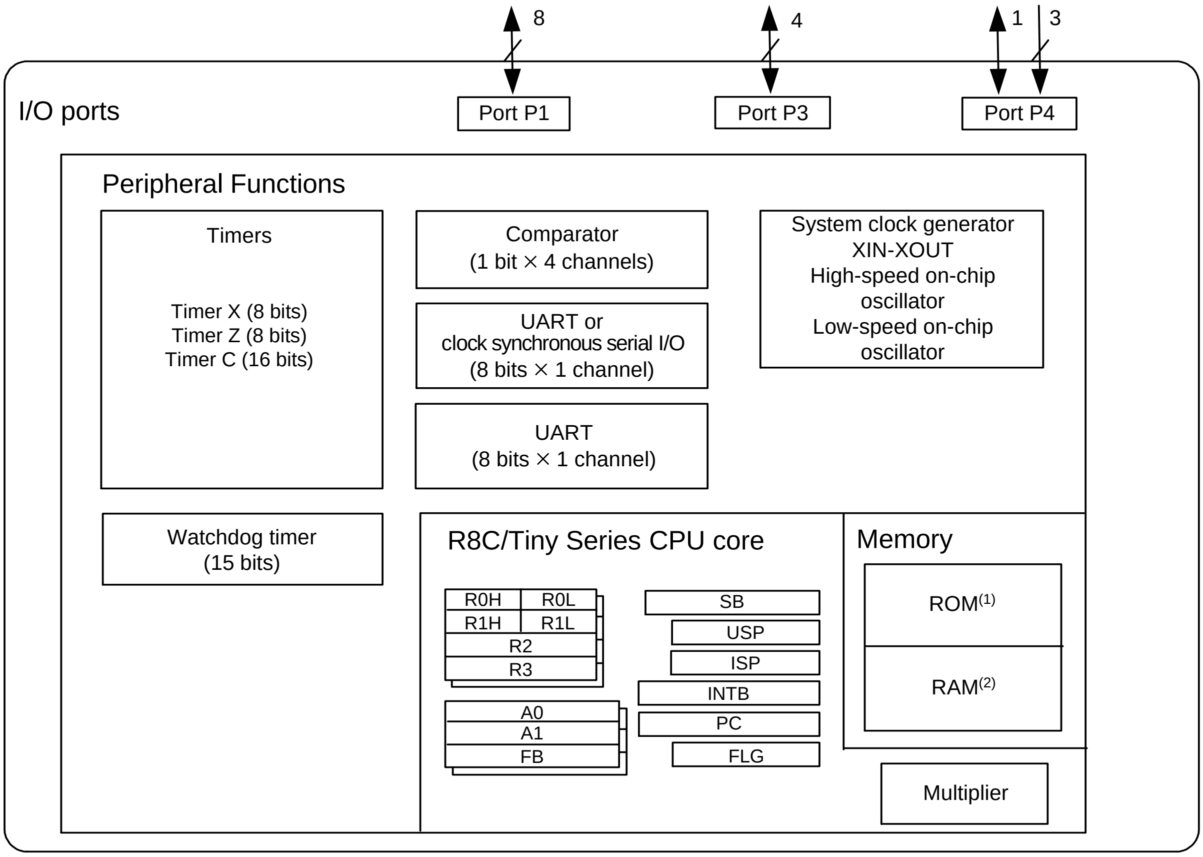
Item		Specification
CPU	Number of fundamental instructions	89 instructions
	Minimum instruction execution time	50 ns (f(XIN) = 20 MHz, VCC = 3.0 to 5.5 V) 100 ns (f(XIN) = 10 MHz, VCC = 2.7 to 5.5 V)
	Operation mode	Single-chip
	Address space	1 Mbyte
	Memory capacity	Refer to **Table 1.4 Product Information for R8C/19 Group**
Peripheral Functions	Ports	I/O ports: 13 pins (including LED drive port) Input port: 3 pins
	LED drive ports	I/O ports: 4 pins
	Timers	Timer X: 8 bits × 1 channel, timer Z: 8 bits × 1 channel (Each timer equipped with 8-bit prescaler) Timer C: 16 bits × 1 channel (Input capture and output compare circuits)
	Serial interfaces	1 channel Clock synchronous serial I/O, UART 1 channel UART
	Comparator	1-bit comparator: 1 circuit, 4 channels
	Watchdog timer	15 bits × 1 channel (with prescaler) Reset start selectable, count source protection mode
	Interrupts	Internal: 10 sources, External: 4 sources, Software: 4 sources, Priority levels: 7 levels
	Clock generation circuits	2 circuits • Main clock generation circuit (with on-chip feedback resistor) • On-chip oscillator (high speed, low speed) High-speed on-chip oscillator has frequency adjustment function
	Oscillation stop detection function	Main clock oscillation stop detection function
	Voltage detection circuit	On-chip
	Power-on reset circuit	On-chip

Electric Characteristics	Supply voltage	VCC = 3.0 to 5.5 V (f(XIN) = 20 MHz)
		VCC = 2.7 to 5.5 V (f(XIN) = 10 MHz)

	Current consumption	Typ. 9 mA (VCC = 5.0 V, f(XIN) = 20 MHz, comparator stopped) Typ. 5 mA (VCC = 3.0 V, f(XIN) = 10MHz, comparator stopped) Typ. 35 µA (VCC = 3.0 V, wait mode, peripheral clock off) Typ. 0.7 µA (VCC = 3.0 V, stop mode)
Flash Memory	Programming and erasure voltage	VCC = 2.7 to 5.5 V
	Programming and erasure endurance	10,000 times (data flash)
		1,000 times (program ROM)
Operating Ambient Temperature		-20 to 85°C -40 to 85°C (D version)
Package		20-pin molded-plastic LSSOP
		20-pin molded-plastic SDIP
		28-pin molded-plastic HWQFN

1.3 Block Diagram

Figure 1.1 shows a Block Diagram.



NOTES:
1. ROM size varies with MCU type.
2. RAM size varies with MCU type.

Figure 1.1 Block Diagram

1.4 Product Information

Table 1.3 lists Product Information for R8C/18 Group and Table 1.4 lists Product Information for R8C/19 Group.

Table 1.3 Product Information for R8C/18 Group **Current of Apr. 2006**

Type No.	ROM Capacity	RAM Capacity	Package Type	Remarks
R5F21181SP	4 Kbytes	384 bytes	PLSP0020JB-A	Flash memory version
R5F21182SP	8 Kbytes	512 bytes	PLSP0020JB-A	
R5F21183SP	12 Kbytes	768 bytes	PLSP0020JB-A	
R5F21184SP	16 Kbytes	1 Kbyte	PLSP0020JB-A	
R5F21181DSP (D)	4 Kbytes	384 bytes	PLSP0020JB-A	D version
R5F21182DSP (D)	8 Kbytes	512 bytes	PLSP0020JB-A	
R5F21183DSP (D)	12 Kbytes	768 bytes	PLSP0020JB-A	
R5F21184DSP (D)	16 Kbytes	1 Kbyte	PLSP0020JB-A	
R5F21181DD	4 Kbytes	384 bytes	PRDP0020BA-A	Flash memory version
R5F21182DD	8 Kbytes	512 bytes	PRDP0020BA-A	
R5F21183DD	12 Kbytes	768 bytes	PRDP0020BA-A	
R5F21184DD	16 Kbytes	1 Kbyte	PRDP0020BA-A	
R5F21182NP	8 Kbytes	512 bytes	PWQN0028KA-B	Flash memory version
R5F21183NP	12 Kbytes	768 bytes	PWQN0028KA-B	
R5F21184NP	16 Kbytes	1 Kbyte	PWQN0028KA-B	

(D): Under Development

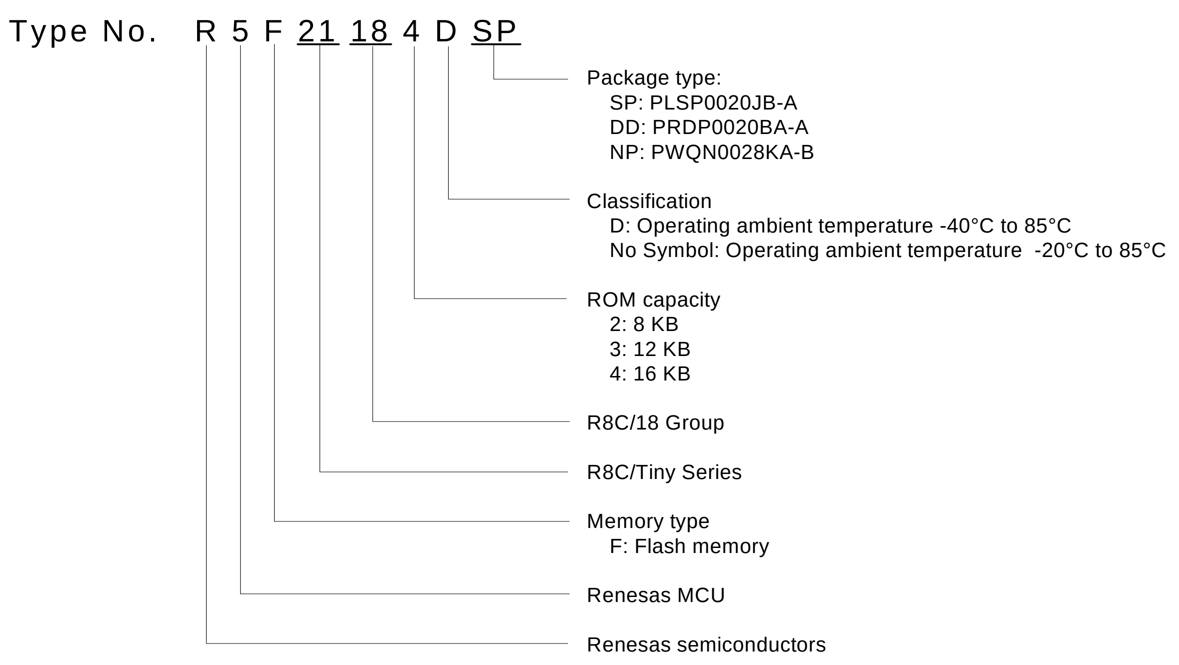
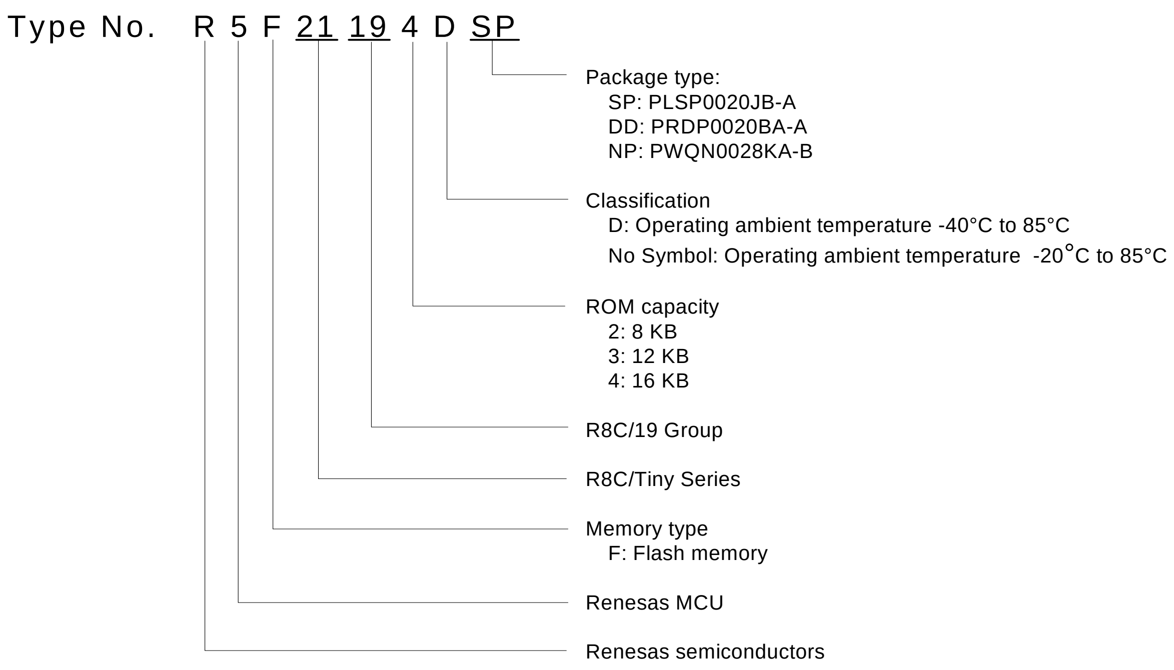


Figure 1.2 Type Number, Memory Size, and Package of R8C/18 Group

Table 1.4 Product Information for R8C/19 Group **Current of Apr. 2006**

Type No.	ROM Capacity		RAM Capacity	Package Type	Remarks
	Program ROM	Data flash			
R5F21191SP	4 Kbytes	1 Kbyte × 2	384 bytes	PLSP0020JB-A	Flash memory version
R5F21192SP	8 Kbytes	1 Kbyte × 2	512 bytes	PLSP0020JB-A	
R5F21193SP	12 Kbytes	1 Kbyte × 2	768 bytes	PLSP0020JB-A	
R5F21194SP	16 Kbytes	1 Kbyte × 2	1 Kbyte	PLSP0020JB-A	
R5F21191DSP (D)	4 Kbytes	1 Kbyte × 2	384 bytes	PLSP0020JB-A	D version
R5F21192DSP (D)	8 Kbytes	1 Kbyte × 2	512 bytes	PLSP0020JB-A	
R5F21193DSP (D)	12 Kbytes	1 Kbyte × 2	768 bytes	PLSP0020JB-A	
R5F21194DSP (D)	16 Kbytes	1 Kbyte × 2	1 Kbyte	PLSP0020JB-A	
R5F21191DD	4 Kbytes	1 Kbyte × 2	384 bytes	PRDP0020BA-A	Flash memory version
R5F21192DD	8 Kbytes	1 Kbyte × 2	512 bytes	PRDP0020BA-A	
R5F21193DD	12 Kbytes	1 Kbyte × 2	768 bytes	PRDP0020BA-A	
R5F21194DD	16 Kbytes	1 Kbyte × 2	1 Kbyte	PRDP0020BA-A	
R5F21192NP	8 Kbytes	1 Kbyte × 2	512 bytes	PWQN0028KA-B	Flash memory version
R5F21193NP	12 Kbytes	1 Kbyte × 2	768 bytes	PWQN0028KA-B	
R5F21194NP	16 Kbytes	1 Kbyte × 2	1 Kbyte	PWQN0028KA-B	

(D): Under Development



Type No. R 5 F 21 19 4 D SP

Package type:
SP: PLSP0020JB-A
DD: PRDP0020BA-A
NP: PWQN0028KA-B

Classification
D: Operating ambient temperature -40°C to 85°C
No Symbol: Operating ambient temperature -20°C to 85°C

ROM capacity
2: 8 KB
3: 12 KB
4: 16 KB

R8C/19 Group

R8C/Tiny Series

Memory type
F: Flash memory

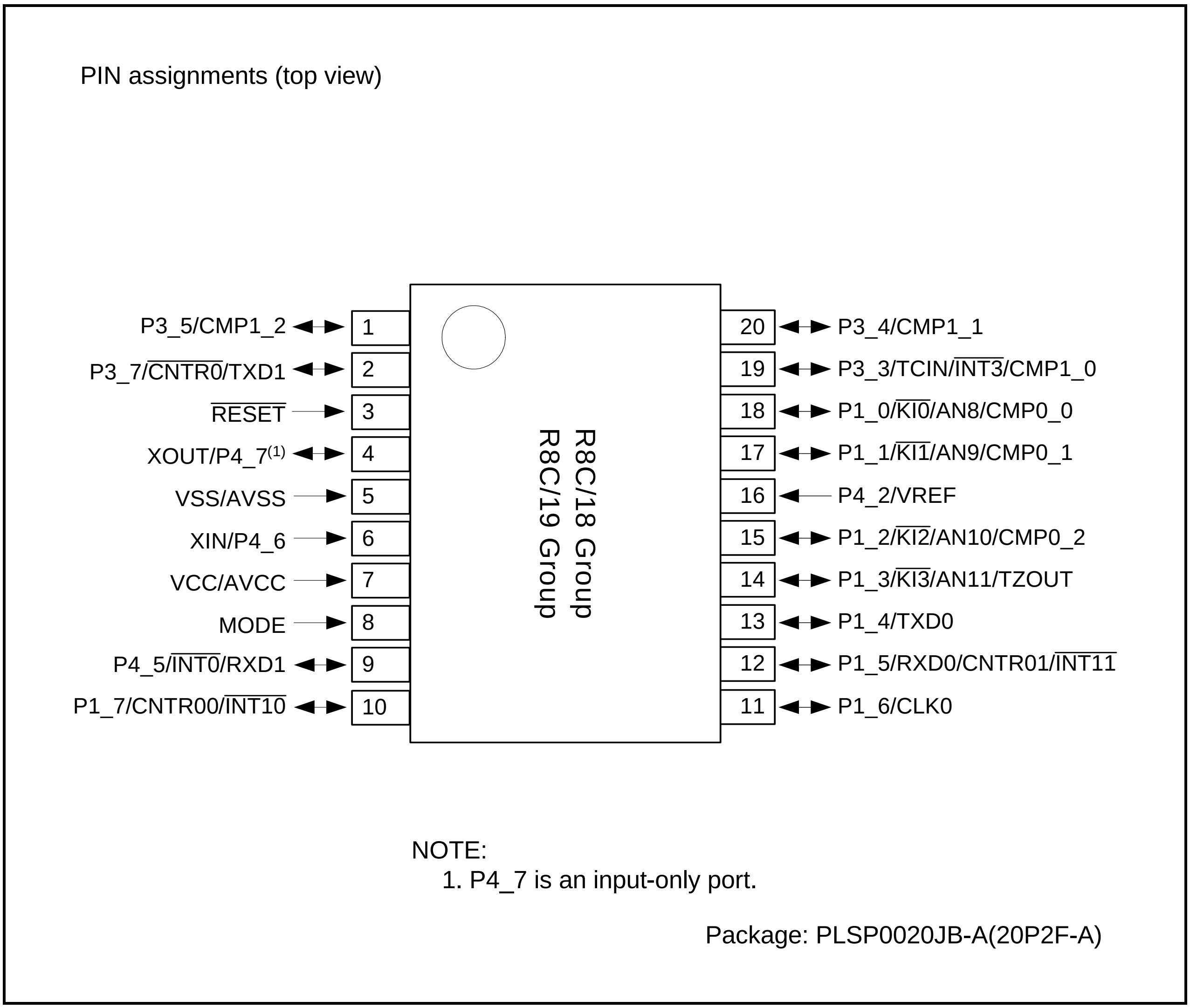
Renesas MCU

Renesas semiconductors

Figure 1.3 Type Number, Memory Size, and Package of R8C/19 Group

1.5 Pin Assignments

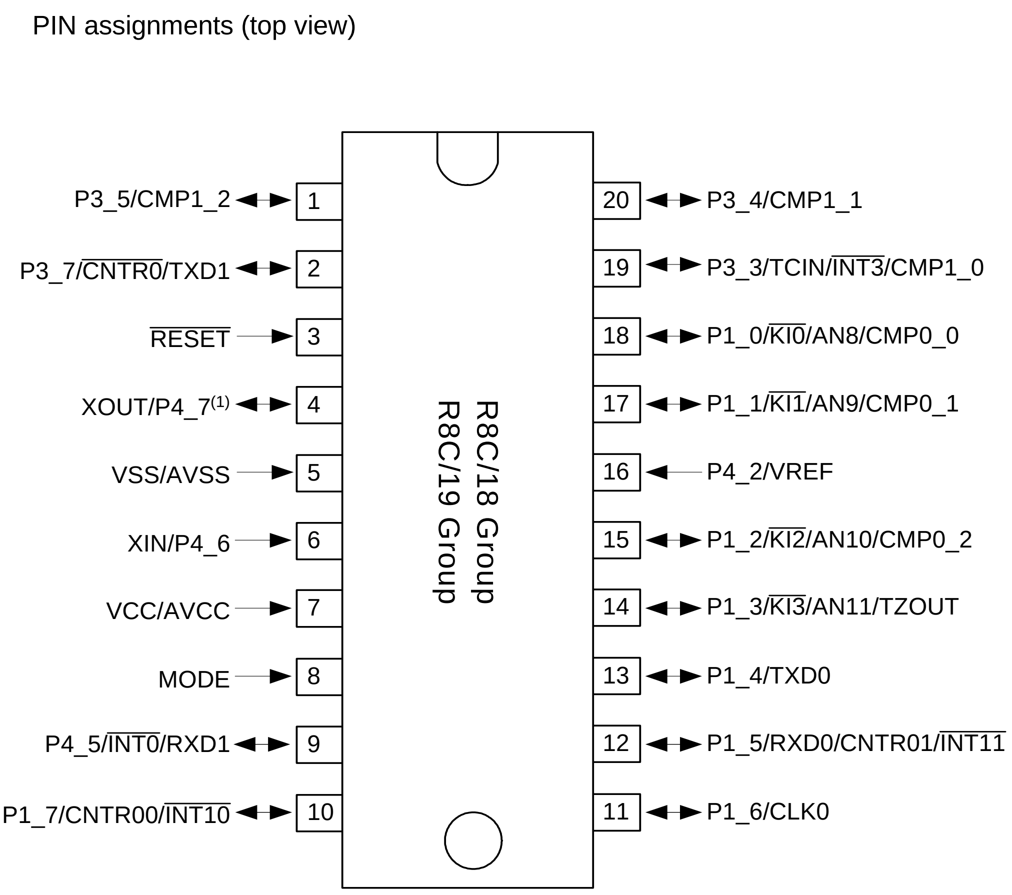
Figure 1.4 shows Pin Assignments for PLSP0020JB-A Package (Top View), Figure 1.5 shows Pin Assignments for PRDP0020BA-A Package (Top View) and Figure 1.6 shows Pin Assignments for PWQN0028KA-B Package (Top View).



NOTE:
1. P4_7 is an input-only port.

Package: PLSP0020JB-A(20P2F-A)

Figure 1.4 Pin Assignments for PLSP0020JB-A Package (Top View)



NOTE:
1. P4_7 is an input-only port.

Package: PRDP0020BA-A(20P4B)

Figure 1.5 Pin Assignments for PRDP0020BA-A Package (Top View)

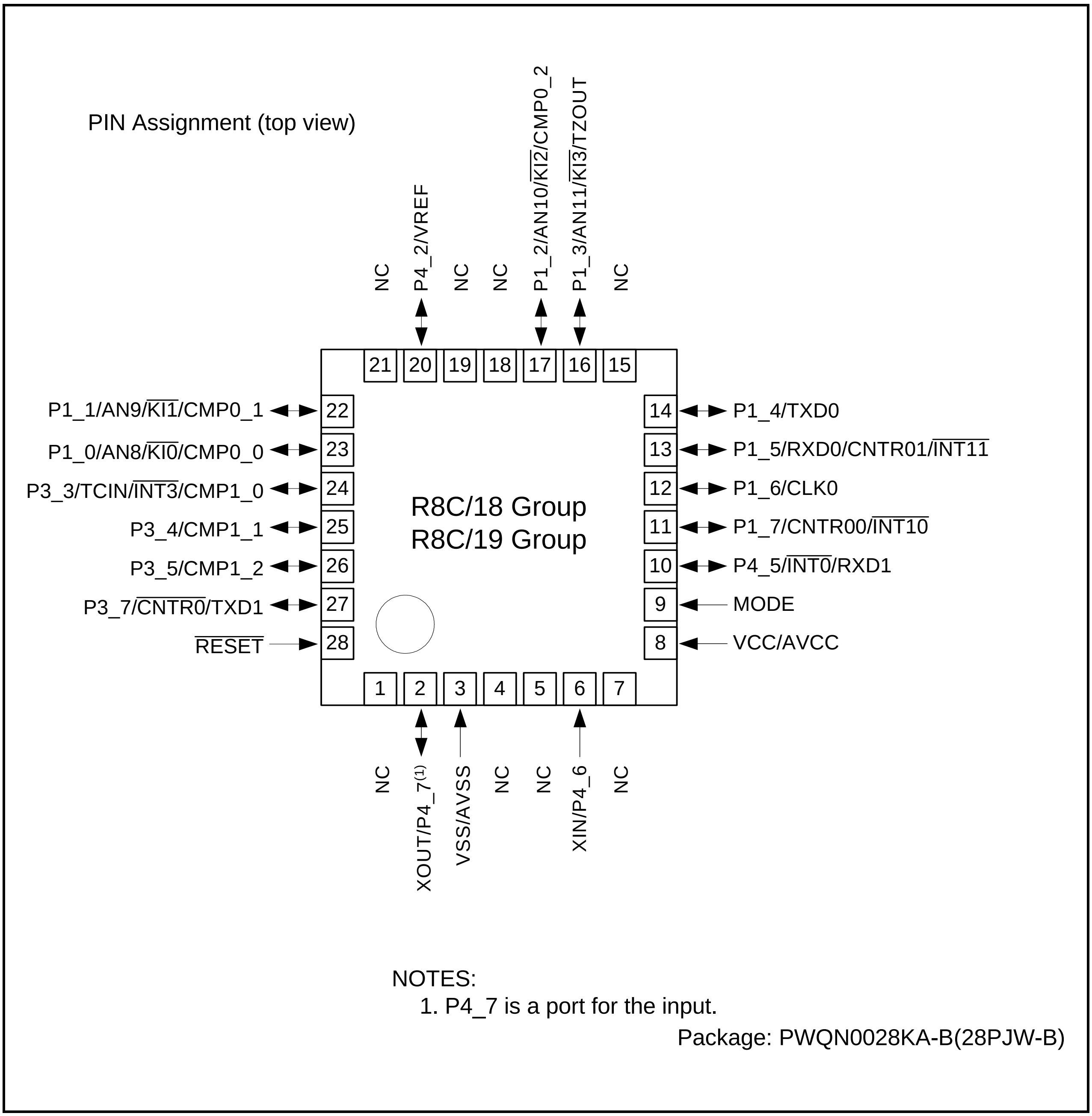


Figure 1.6 Pin Assignments for PWQN0028KA-B Package (Top View)

1.6 Pin Functions

Table 1.5 lists Pin Functions, Table 1.6 lists Pin Name Information by Pin Number of PLSP0020JB-A, PRDP0020BA-A packages, and Table 1.7 lists Pin Name Information by Pin Number of PWQN0028KA-B package.

Table 1.5 Pin Functions

Type	Symbol	I/O Type	Description
Power supply input	VCC VSS	I	Apply 2.7 V to 5.5 V to the VCC pin. Apply 0 V to the VSS pin.
Analog power supply input	AVCC, AVSS	I	Power supply for the comparator Connect a capacitor between AVCC and AVSS.
Reset input	$\overline{\mathrm{RESET}}$	I	Input “L” on this pin resets the MCU.
MODE	MODE	I	Connect this pin to VCC via a resistor.
Main clock input	XIN	I	These pins are provided for main clock generation circuit I/O. Connect a ceramic resonator or a crystal oscillator between the XIN and XOUT pins. To use an external clock, input it to the XIN pin and leave the XOUT pin open.
Main clock output	XOUT	O	
$\overline{\mathrm{INT}}$ interrupt	$\overline{\mathrm{INT0}}$, $\overline{\mathrm{INT1}}$, $\overline{\mathrm{INT3}}$	I	$\overline{\mathrm{INT}}$ interrupt input pins
Key input interrupt	$\overline{\mathrm{KI0}}$ to $\overline{\mathrm{KI3}}$	I	Key input interrupt input pins
Timer X	CNTR0	I/O	Timer X I/O pin
	$\overline{\mathrm{CNTR0}}$	O	Timer X output pin
Timer Z	TZOUT	O	Timer Z output pin
Timer C	TCIN	I	Timer C input pin
	CMP0_0 to CMP0_2, CMP1_0 to CMP1_2	O	Timer C output pins
Serial interface	CLK0	I/O	Transfer clock I/O pin
	RXD0, RXD1	I	Serial data input pins
	TXD0, TXD1	O	Serial data output pins
Reference voltage input	VREF	I	Reference voltage input pin to comparator
Comparator	AN8 to AN11	I	Analog input pins to comparator
I/O port	P1_0 to P1_7, P3_3 to P3_5, P3_7, P4_5	I/O	CMOS I/O ports. Each port has an I/O select direction register, allowing each pin in the port to be directed for input or output individually. Any port set to input can be set to use a pull-up resistor or not by a program. P1_0 to P1_3 also function as LED drive ports.
Input port	P4_2, P4_6, P4_7	I	Input-only ports

I: Input O: Output I/O: Input and output

Table 1.6 Pin Name Information by Pin Number of PLSP0020JB-A, PRDP0020BA-A packages

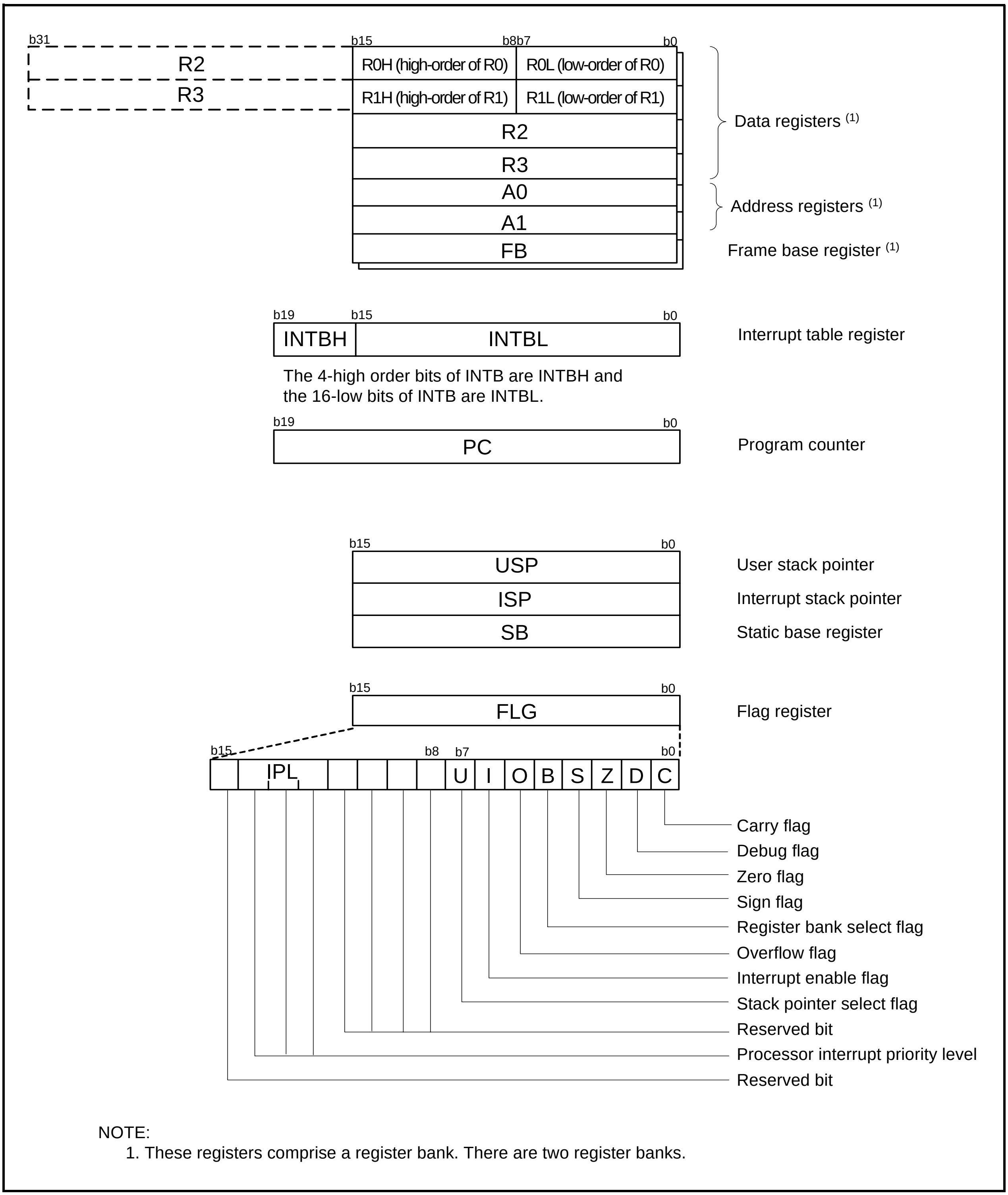
Pin Number	Control Pin	Port	I/O Pin Functions for Peripheral Modules			
			Interrupt	Timer	Serial Interface	Comparator
1		P3_5		CMP1_2		
2		P3_7		$\overline{\mathrm{CNTR0}}$	TXD1	
3	$\overline{\mathrm{RESET}}$					
4	XOUT	P4_7				
5	VSS/AVSS					
6	XIN	P4_6				
7	VCC/AVCC					
8	MODE					
9		P4_5	$\overline{\mathrm{INT0}}$		RXD1	
10		P1_7	$\overline{\mathrm{INT10}}$	CNTR00		
11		P1_6			CLK0	
12		P1_5	$\overline{\mathrm{INT11}}$	CNTR01	RXD0	
13		P1_4			TXD0	
14		P1_3	$\overline{\mathrm{KI3}}$	TZOUT		AN11
15		P1_2	$\overline{\mathrm{KI2}}$	CMP0_2		AN10
16	VREF	P4_2				
17		P1_1	$\overline{\mathrm{KI1}}$	CMP0_1		AN9
18		P1_0	$\overline{\mathrm{KI0}}$	CMP0_0		AN8
19		P3_3	$\overline{\mathrm{INT3}}$	TCIN/CMP1_0		
20		P3_4		CMP1_1		

Table 1.7 Pin Name Information by Pin Number of PWQN0028KA-B package

Pin Number	Control Pin	Port	I/O Pin of Peripheral Function			
			Interrupt	Timer	Serial Interface	Comparator
1	NC					
2	XOUT	P4_7				
3	VSS/AVSS					
4	NC					
5	NC					
6	XIN	P4_6				
7	NC					
8	VCC/AVCC					
9	MODE					
10		P4_5	$\overline{\text{INT0}}$		RXD1	
11		P1_7	$\overline{\text{INT10}}$	CNTR00		
12		P1_6			CLK0	
13		P1_5	$\overline{\text{INT11}}$	CNTR01	RXD0	
14		P1_4			TXD0	
15	NC					
16		P1_3	$\overline{\text{KI3}}$	TZOUT		AN11
17		P1_2	$\overline{\text{KI2}}$	CMP0_2		AN10
18	NC					
19	NC					
20	VREF	P4_2				
21	NC					
22		P1_1	$\overline{\text{KI1}}$	CMP0_1		AN9
23		P1_0	$\overline{\text{KI0}}$	CMP0_0		AN8
24		P3_3	$\overline{\text{INT3}}$	TCIN/CMP1_0		
25		P3_4		CMP1_1		
26		P3_5		CMP1_2		
27		P3_7		$\overline{\text{CNTR0}}$	TXD1	
28	$\overline{\text{RESET}}$					

2. Central Processing Unit (CPU)

Figure 2.1 shows the CPU Registers. The CPU contains 13 registers. R0, R1, R2, R3, A0, A1, and FB configure a register bank. There are two sets of register bank.



NOTE:
1. These registers comprise a register bank. There are two register banks.

Figure 2.1 CPU Registers

2.1 Data Registers (R0, R1, R2, and R3)

R0 is a 16-bit register for transfer, arithmetic, and logic operations. The same applies to R1 to R3. R0 can be split into high-order bits (R0H) and low-order bits (R0L) to be used separately as 8-bit data registers. R1H and R1L are analogous to R0H and R0L. R2 can be combined with R0 and used as a 32-bit data register (R2R0). R3R1 is analogous to R2R0.

2.2 Address Registers (A0 and A1)

A0 is a 16-bit register for address register indirect addressing and address register relative addressing. It is also used for transfer, arithmetic and logic operations. A1 is analogous to A0. A1 can be combined with A0 and used as a 32-bit address register (A1A0).

2.3 Frame Base Register (FB)

FB is a 16-bit register for FB relative addressing.

2.4 Interrupt Table Register (INTB)

INTB is a 20-bit register that indicates the start address of an interrupt vector table.

2.5 Program Counter (PC)

PC is 20 bits wide, indicates the address of the next instruction to be executed.

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

The stack pointer (SP), USP, and ISP, are each 16 bits wide. The U flag of FLG is used to switch between USP and ISP.

2.7 Static Base Register (SB)

SB is a 16-bit register for SB relative addressing.

2.8 Flag Register (FLG)

FLG is an 11-bit register indicating the CPU state.

2.8.1 Carry Flag (C)

The C flag retains a carry, borrow, or shift-out bits that have been generated by the arithmetic and logic unit.

2.8.2 Debug Flag (D)

The D flag is for debugging only. Set it to 0.

2.8.3 Zero Flag (Z)

The Z flag is set to 1 when an arithmetic operation results in 0; otherwise to 0.

2.8.4 Sign Flag (S)

The S flag is set to 1 when an arithmetic operation results in a negative value; otherwise to 0.

2.8.5 Register Bank Select Flag (B)

Register bank 0 is selected when the B flag is 0. Register bank 1 is selected when this flag is set to 1.

2.8.6 Overflow Flag (O)

The O flag is set to 1 when the operation results in an overflow; otherwise to 0.

2.8.7 Interrupt Enable Flag (I)

The I flag enables maskable interrupts.

Interrupts are disabled when the I flag is set to 0, and are enabled when the I flag is set to 1. The I flag is set to 0 when an interrupt request is acknowledged.

2.8.8 Stack Pointer Select Flag (U)

ISP is selected when the U flag is set to 0; USP is selected when the U flag is set to 1.

The U flag is set to 0 when a hardware interrupt request is acknowledged or the INT instruction of software interrupt numbers 0 to 31 is executed.

2.8.9 Processor Interrupt Priority Level (IPL)

IPL is 3 bits wide, assigns processor interrupt priority levels from level 0 to level 7.

If a requested interrupt has higher priority than IPL, the interrupt is enabled.

2.8.10 Reserved Bit

If necessary, set to 0. When read, the content is undefined.

3. Memory

3.1 R8C/18 Group

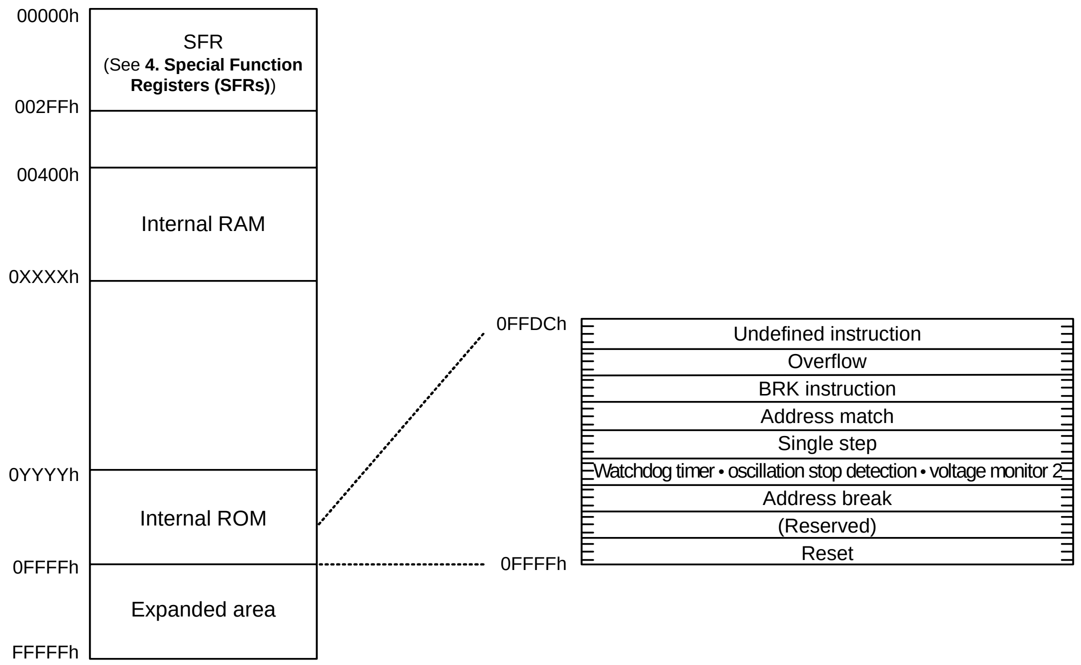
Figure 3.1 is a Memory Map of R8C/18 Group. The R8C/18 Group has 1 Mbyte of address space from addresses 00000h to FFFFFh.

The internal ROM area is allocated lower addresses, beginning with address 0FFFFh. For example, a 16-Kbyte internal ROM is allocated addresses 0C000h to 0FFFFh.

The fixed interrupt vector table is allocated addresses 0FFDCh to 0FFFFh. They store the starting address of each interrupt routine.

The internal RAM is allocated higher addresses, beginning with address 00400h. For example, a 1-Kbyte internal RAM area is allocated addresses 00400h to 007FFh. The internal RAM is used not only for storing data but also for calling subroutines and as stacks when interrupt requests are acknowledged.

Special function registers (SFRs) are allocated addresses 00000h to 002FFh. The peripheral function control registers are allocated here. All addresses within the SFR, which have nothing allocated are reserved for future use and cannot be accessed by users.



NOTE:

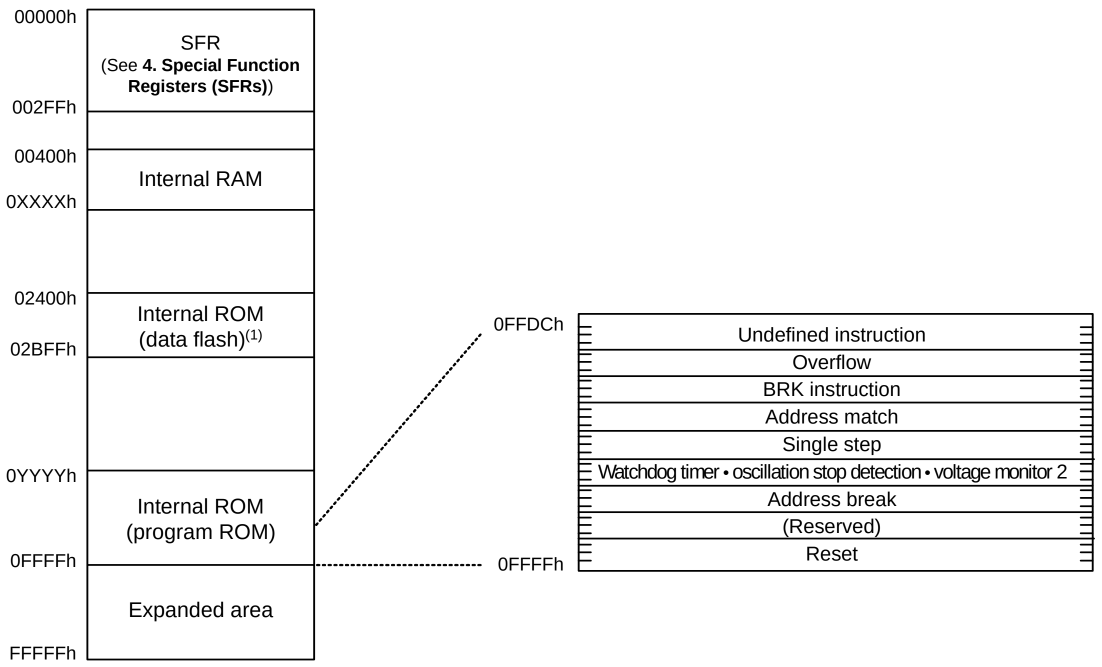
1. The blank regions are reserved. Do not access locations in these regions.

Part Number	Internal ROM		Internal RAM	
	Size	Address 0YYYYh	Size	Address 0XXXXh
R5F21184SP, R5F21184DSP, R5F21184DD, R5F21184NP	16 Kbytes	0C000h	1 Kbyte	007FFh
R5F21183SP, R5F21183DSP, R5F21183DD, R5F21183NP	12 Kbytes	0D000h	768 bytes	006FFh
R5F21182SP, R5F21182DSP, R5F21182DD, R5F21182NP	8 Kbytes	0E000h	512 bytes	005FFh
R5F21181SP, R5F21181DSP, R5F21181DD	4 Kbytes	0F000h	384 bytes	0057Fh

Figure 3.1 Memory Map of R8C/18 Group

3.2 R8C/19 Group

Figure 3.2 is a Memory Map of R8C/19 Group. The R8C/19 group has 1 Mbyte of address space from addresses 00000h to FFFFFh.
The internal ROM (program ROM) is allocated lower addresses, beginning with address 0FFFFh. For example, a 16-Kbyte internal ROM area is allocated addresses 0C000h to 0FFFFh.
The fixed interrupt vector table is allocated addresses 0FFDCh to 0FFFFh. They store the starting address of each interrupt routine.
The internal ROM (data flash) is allocated addresses 02400h to 02BFFh.
The internal RAM is allocated higher addresses, beginning with address 00400h. For example, a 1-Kbyte internal RAM area is allocated addresses 00400h to 007FFh. The internal RAM is used not only for storing data but also for calling subroutines and as stacks when interrupt requests are acknowledged.
Special function registers (SFRs) are allocated addresses 00000h to 002FFh. The peripheral function control registers are allocated here. All addresses within the SFR, which have nothing allocated are reserved for future use and cannot be accessed by users.



NOTES:
1. Data flash block A (1 Kbyte) and B (1 Kbyte) are shown.
2. The blank regions are reserved. Do not access locations in these regions.

Part Number	Internal ROM		Internal RAM	
	Size	Address 0YYYYh	Size	Address 0XXXXh
R5F21194SP, R5F21194DSP, R5F21194DD, R5F21194NP	16 Kbytes	0C000h	1 Kbyte	007FFh
R5F21193SP, R5F21193DSP, R5F21193DD, R5F21193NP	12 Kbytes	0D000h	768 bytes	006FFh
R5F21192SP, R5F21192DSP, R5F21192DD, R5F21192NP	8 Kbytes	0E000h	512 bytes	005FFh
R5F21191SP, R5F21191DSP, R5F21191DD	4 Kbytes	0F000h	384 bytes	0057Fh

Figure 3.2 Memory Map of R8C/19 Group

4. Special Function Registers (SFRs)

An SFR (special function register) is a control register for a peripheral function. Tables 4.1 to 4.4 list the special function registers.

Table 4.1 SFR Information (1)(1)

Address	Register	Symbol	After reset
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	00h
0005h	Processor Mode Register 1	PM1	00h
0006h	System Clock Control Register 0	CM0	01101000b
0007h	System Clock Control Register 1	CM1	00100000b
0008h			
0009h	Address Match Interrupt Enable Register	AIER	00h
000Ah	Protect Register	PRCR	00h
000Bh			
000Ch	Oscillation Stop Detection Register	OCD	00000100b
000Dh	Watchdog Timer Reset Register	WDTR	XXh
000Eh	Watchdog Timer Start Register	WDTS	XXh
000Fh	Watchdog Timer Control Register	WDC	00011111b
0010h	Address Match Interrupt Register 0	RMAD0	00h
0011h			00h
0012h			X0h
0013h			
0014h	Address Match Interrupt Register 1	RMAD1	00h
0015h			00h
0016h			X0h
0017h			
0018h			
0019h			
001Ah			
001Bh			
001Ch	Count Source Protection Mode Register	CSPR	00h
001Dh			
001Eh	$\overline{\text{INT0}}$ Input Filter Select Register	INT0F	00h
001Fh			
0020h	High-Speed On-Chip Oscillator Control Register 0	HRA0	00h
0021h	High-Speed On-Chip Oscillator Control Register 1	HRA1	When shipping
0022h	High-Speed On-Chip Oscillator Control Register 2	HRA2	00h
0023h			
002Ah			
002Bh			
002Ch			
002Dh			
002Eh			
002Fh			
0030h			
0031h	Voltage Detection Register 1(2)	VCA1	00001000b

0032h	Voltage Detection Register 2(2)	VCA2	00h(3)
			01000000b(4)

0033h			
0034h			
0035h			

0036h	Voltage Monitor 1 Circuit Control Register (2)	VW1C	0000X000b(3)
			0100X001b(4)

0037h	Voltage Monitor 2 Circuit Control Register (5)	VW2C	00h
0038h			
0039h			
003Ah			
003Bh			
003Ch			
003Dh			
003Eh			
003Fh			

X: Undefined

NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. Software reset, watchdog timer reset, and voltage monitor 2 reset do not affect this register.
3. After hardware reset.
4. After power-on reset or voltage monitor 1 reset.
5. Software reset, watchdog timer reset, and voltage monitor 2 reset do not affect b2 and b3.

Table 4.2 SFR Information (2)[1]

Address	Register	Symbol	After reset
0040h			
0041h			
0042h			
0043h			
0044h			
0045h			
0046h			
0047h			
0048h			
0049h			
004Ah			
004Bh			
004Ch			
004Dh	Key Input Interrupt Control Register	KUPIC	XXXXX000b
004Eh	Comparator Conversion Interrupt Control Register	ADIC	XXXXX000b
004Fh			
0050h	Compare 1 Interrupt Control Register	CMP1IC	XXXXX000b
0051h	UART0 Transmit Interrupt Control Register	S0TIC	XXXXX000b
0052h	UART0 Receive Interrupt Control Register	S0RIC	XXXXX000b
0053h	UART1 Transmit Interrupt Control Register	S1TIC	XXXXX000b
0054h	UART1 Receive Interrupt Control Register	S1RIC	XXXXX000b
0055h			
0056h	Timer X Interrupt Control Register	TXIC	XXXXX000b
0057h			
0058h	Timer Z Interrupt Control Register	TZIC	XXXXX000b
0059h	$\overline{\text{INT1}}$ Interrupt Control Register	INT1IC	XXXXX000b
005Ah	$\overline{\text{INT3}}$ Interrupt Control Register	INT3IC	XXXXX000b
005Bh	Timer C Interrupt Control Register	TCIC	XXXXX000b
005Ch	Compare 0 Interrupt Control Register	CMP0IC	XXXXX000b
005Dh	$\overline{\text{INT0}}$ Interrupt Control Register	INT0IC	XX00X000b
005Eh			
005Fh			
0060h			
0061h			
0062h			
0063h			
0064h			
0065h			
0066h			
0067h			
0068h			
0069h			
006Ah			
006Bh			
006Ch			
006Dh			
006Eh			
006Fh			
0070h			
0071h			
0072h			
0073h			
0074h			
0075h			
0076h			
0077h			
0078h			
0079h			
007Ah			
007Bh			
007Ch			
007Dh			
007Eh			
007Fh			

X: Undefined

NOTE:

1. The blank regions are reserved. Do not access locations in these regions.

Table 4.3 SFR Information (3)[1]

Address	Register	Symbol	After reset
0080h	Timer Z Mode Register	TZMR	00h
0081h			
0082h			
0083h			
0084h	Timer Z Waveform Output Control Register	PUM	00h
0085h	Prescaler Z Register	PREZ	FFh
0086h	Timer Z Secondary Register	TZSC	FFh
0087h	Timer Z Primary Register	TZPR	FFh
0088h			
0089h			
008Ah	Timer Z Output Control Register	TZOC	00h
008Bh	Timer X Mode Register	TXMR	00h
008Ch	Prescaler X Register	PREX	FFh
008Dh	Timer X Register	TX	FFh
008Eh	Timer Count Source Setting Register	TCSS	00h
008Fh			
0090h	Timer C Register	TC	00h
0091h			00h
0092h			
0093h			
0094h			
0095h			
0096h	External Input Enable Register	INTEN	00h
0097h			
0098h	Key Input Enable Register	KIEN	00h
0099h			
009Ah	Timer C Control Register 0	TCC0	00h
009Bh	Timer C Control Register 1	TCC1	00h
009Ch	Capture, Compare 0 Register	TM0	00h
009Dh			00h[2]
009Eh	Compare 1 Register	TM1	FFh
009Fh			FFh
00A0h	UART0 Transmit/Receive Mode Register	U0MR	00h
00A1h	UART0 Bit Rate Register	U0BRG	XXh
00A2h	UART0 Transmit Buffer Register	U0TB	XXh
00A3h			XXh
00A4h	UART0 Transmit/Receive Control Register 0	U0C0	00001000b
00A5h	UART0 Transmit/Receive Control Register 1	U0C1	00000010b
00A6h	UART0 Receive Buffer Register	U0RB	XXh
00A7h			XXh
00A8h	UART1 Transmit/Receive Mode Register	U1MR	00h
00A9h	UART1 Bit Rate Register	U1BRG	XXh
00AAh	UART1 Transmit Buffer Register	U1TB	XXh
00ABh			XXh
00ACh	UART1 Transmit/Receive Control Register 0	U1C0	00001000b
00ADh	UART1 Transmit/Receive Control Register 1	U1C1	00000010b
00AEh	UART1 Receive Buffer Register	U1RB	XXh
00AFh			XXh
00B0h	UART Transmit/Receive Control Register 2	UCON	00h
00B1h			
00B2h			
00B3h			
00B4h			
00B5h			
00B6h			
00B7h			
00B8h			
00B9h			
00BAh			
00BBh			
00BCh			
00BDh			
00BEh			
00BFh			

X: Undefined

NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. When the output compare mode is selected (the TCC13 bit in the TCC1 register = 1), the value is set to $FFFF_{16}$.

Table 4.4 SFR Information (4)[1]

Address	Register	Symbol	After reset
00C0h	A/D Register	AD	XXh
00C1h			
00C2h			
00C3h			
00C4h			
00C5h			
00C6h			
00C7h			
00C8h			
00C9h			
00CAh			
00CBh			
00CCh			
00CDh			
00CEh			
00CFh			
00D0h			
00D1h			
00D2h			
00D3h			
00D4h	A/D Control Register 2	ADCON2	00h
00D5h			
00D6h	A/D Control Register 0	ADCON0	00000XXXb
00D7h	A/D Control Register 1	ADCON1	00h
00D8h			
00D9h			
00DAh			
00DBh			
00DCh			
00DDh			
00DEh			
00DFh			
00E0h			
00E1h	Port P1 Register	P1	XXh
00E2h			
00E3h	Port P1 Direction Register	PD1	00h
00E4h			
00E5h	Port P3 Register	P3	XXh
00E6h			
00E7h	Port P3 Direction Register	PD3	00h
00E8h	Port P4 Register	P4	XXh
00E9h			
00EAh	Port P4 Direction Register	PD4	00h
00EBh			
00ECh			
00EDh			
00EEh			
00EFh			
00F0h			
00F1h			
00F2h			
00F3h			
00F4h			
00F5h			
00F6h			
00F7h			
00F8h			
00F9h			
00FAh			
00FBh			
00FCh	Pull-Up Control Register 0	PUR0	00XX0000b
00FDh	Pull-Up Control Register 1	PUR1	XXXXXX0Xb
00FEh	Port P1 Drive Capacity Control Register	DRR	00h
00FFh	Timer C Output Control Register	TCOUT	00h
01B3h	Flash Memory Control Register 4	FMR4	01000000b
01B4h			
01B5h	Flash Memory Control Register 1	FMR1	1000000Xb
01B6h			
01B7h	Flash Memory Control Register 0	FMR0	00000001b
0FFFFh	Optional Function Select Register	OFS	(Note 2)

X: Undefined

NOTES:

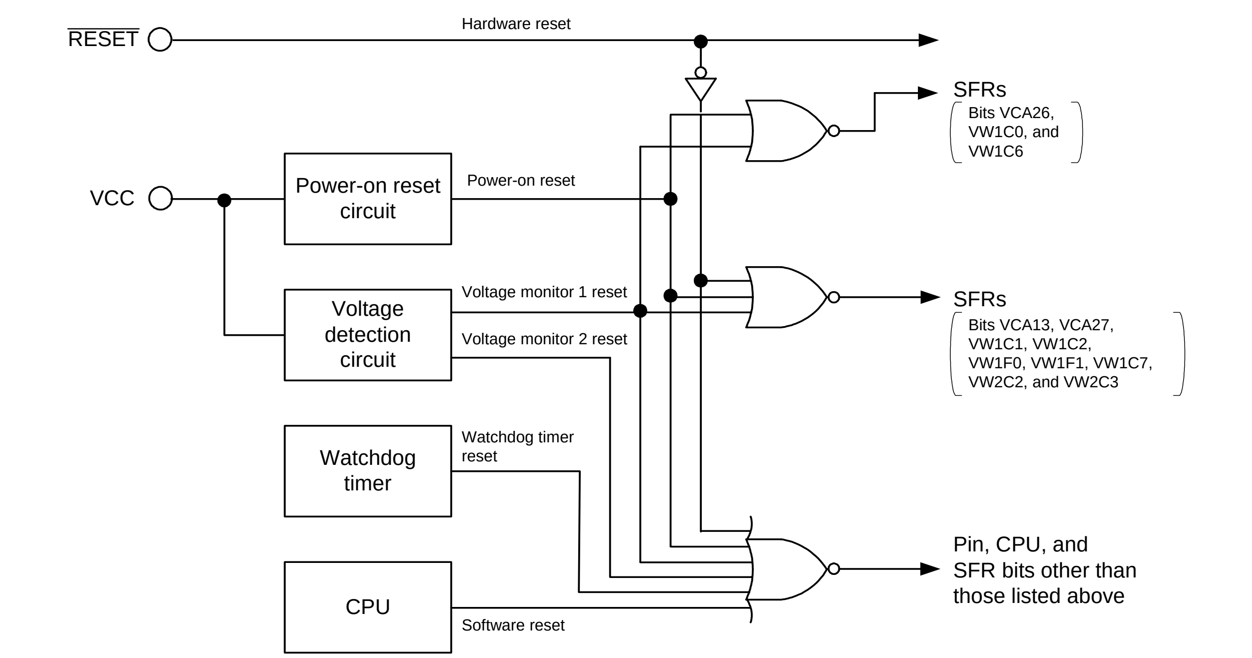
1. The blank regions, 0100h to 01B2h and 01B8h to 02FFh are all reserved. Do not access locations in these regions.
2. The OFS register cannot be changed by a program. Use a flash programmer to write to it.

5. Resets

The following resets are implemented: hardware reset, power-on reset, voltage monitor 1 reset, voltage monitor 2 reset, watchdog timer reset, and software reset. Table 5.1 lists the Reset Names and Sources.

Table 5.1 Reset Names and Sources

Reset Name	Source
Hardware reset	Input voltage of $\overline{\text{RESET}}$ pin is held “L”
Power-on reset	VCC rises.
Voltage monitor 1 reset	VCC falls (monitor voltage: Vdet1).
Voltage monitor 2 reset	VCC falls (monitor voltage: Vdet2).
Watchdog timer reset	Underflow of watchdog timer
Software reset	Write 1 to PM03 bit in PM0 register.


$\overline{\text{RESET}}$
Hardware reset
VCC
Power-on reset circuit
Power-on reset
Voltage detection circuit
Voltage monitor 1 reset
Voltage monitor 2 reset
Watchdog timer
Watchdog timer reset
CPU
Software reset
SFRs
Bits VCA26, VW1C0, and VW1C6
SFRs
Bits VCA13, VCA27, VW1C1, VW1C2, VW1F0, VW1F1, VW1C7, VW2C2, and VW2C3
Pin, CPU, and SFR bits other than those listed above

VCA13: Bit in VCA1 register
VCA26, VCA27: Bits in VCA2 register
VW1C0 to VW1C2, VW1F0, VW1F1, VW1C6, VW1C7: Bits in VW1C register
VW2C2, VW2C3: Bits in VW2C register

Figure 5.1 Block Diagram of Reset Circuit

Table 5.2 shows the Pin Functions after Reset, Figure 5.2 shows CPU Register Status after Reset and Figure 5.3 shows Reset Sequence.

Table 5.2 Pin Functions after Reset

Pin Name	Pin Functions
P1	Input port
P3_3 to P3_5, P3_7	Input port
P4_2, P4_5 to P4_7	Input port

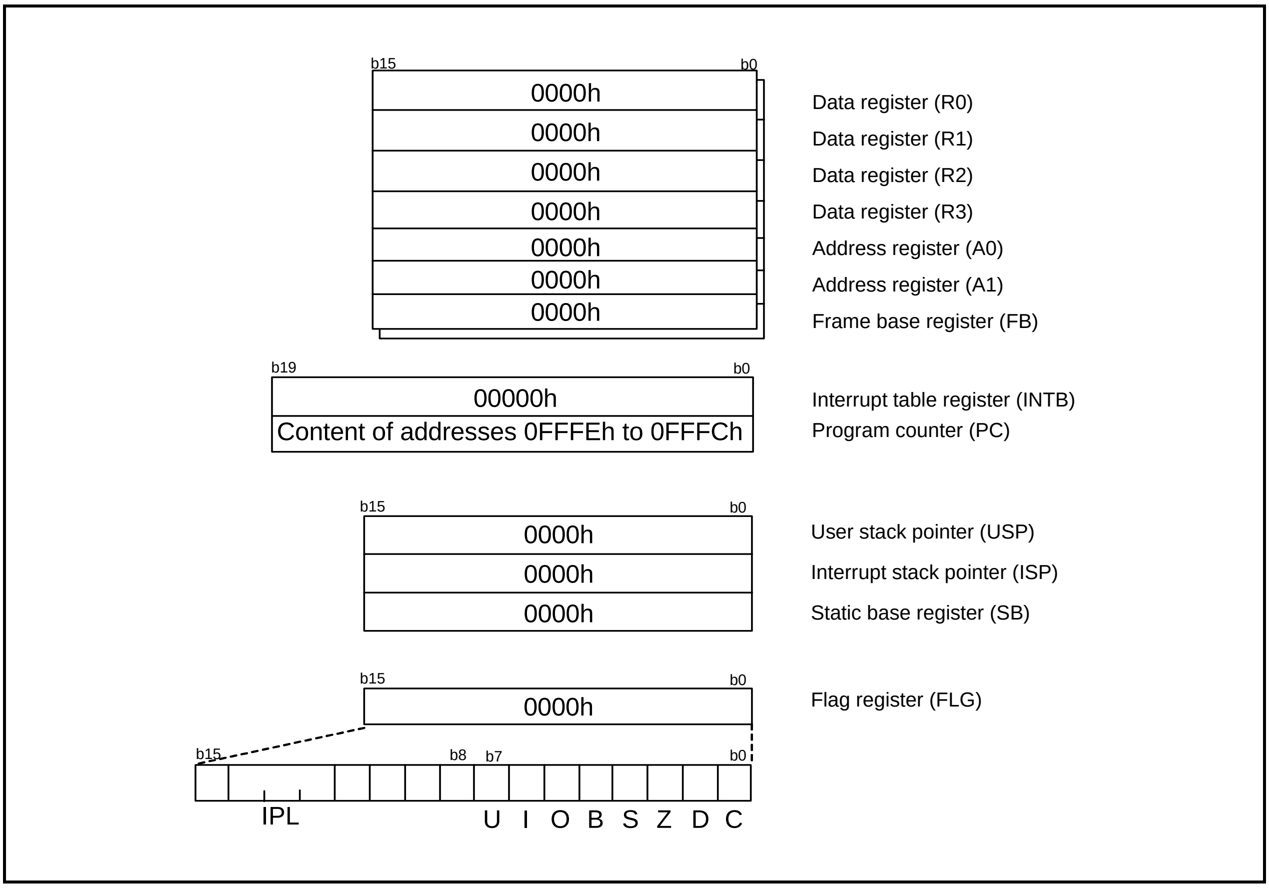
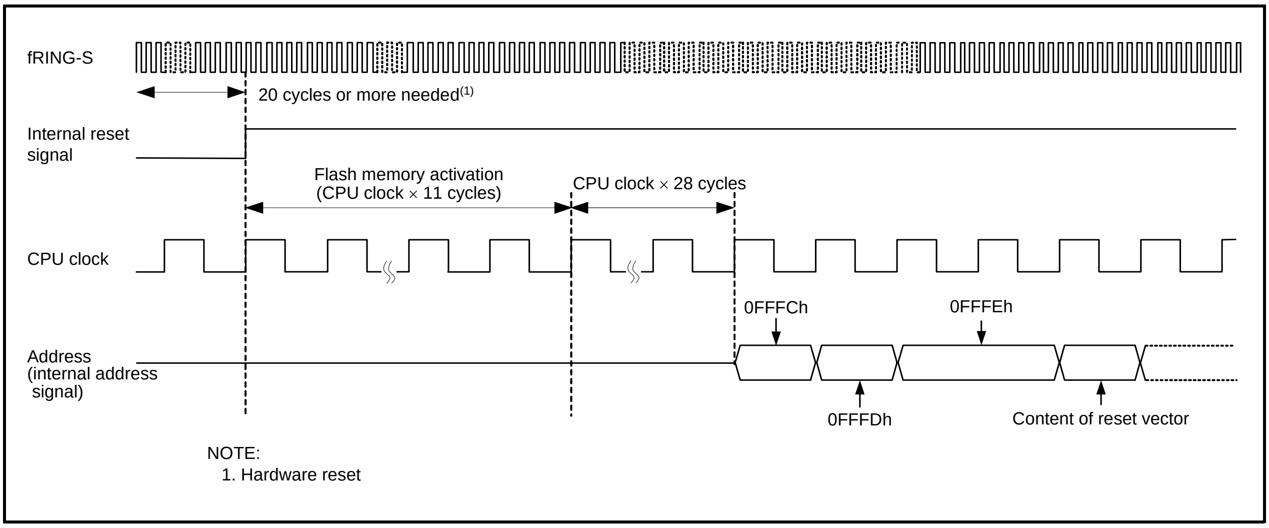


Figure 5.2 CPU Register Status after Reset



NOTE:
1. Hardware reset

Figure 5.3 Reset Sequence

5.1 Hardware Reset

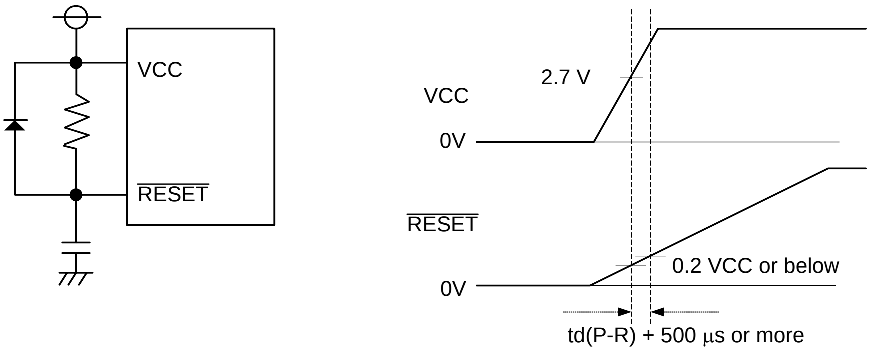
A reset is applied using the RESET pin. When an “L” signal is applied to the RESET pin while the supply voltage meets the recommended operating conditions, pins, CPU, and SFRs are reset (refer to **Table 5.2 Pin Functions after Reset**). When the input level applied to the RESET pin changes from “L” to “H”, a program is executed beginning with the address indicated by the reset vector. After reset, the low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock.
Refer to **4. Special Function Registers (SFRs)** for the state of the SFRs after reset.
The internal RAM is not reset. If the RESET pin is pulled “L” while writing to the internal RAM is in progress, the contents of internal RAM will be undefined.
Figure 5.4 shows an Example of Hardware Reset Circuit and Operation and Figure 5.5 shows an Example of Hardware Reset Circuit (Usage Example of External Supply Voltage Detection Circuit) and Operation.

5.1.1 When Power Supply is Stable

(1) Apply “L” to the RESET pin.
(2) Wait for 500 µs (1/fRING-S × 20).
(3) Apply “H” to the RESET pin.

5.1.2 Power On

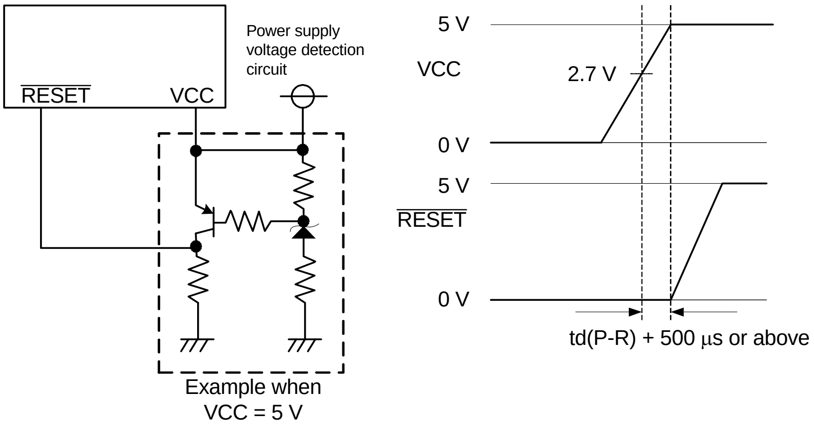
(1) Apply “L” to the RESET pin.
(2) Let the supply voltage increase until it meets the recommended operating condition.
(3) Wait for td(P-R) or more to allow the internal power supply to stabilize (refer to **18. Electrical Characteristics**).
(4) Wait for 500 µs (1/fRING-S × 20).
(5) Apply “H” to the RESET pin.



NOTE:

1. Refer to **18. Electrical Characteristics.**

Figure 5.4 Example of Hardware Reset Circuit and Operation



NOTE:

1. Refer to **18. Electrical Characteristics.**

Figure 5.5 Example of Hardware Reset Circuit (Usage Example of External Supply Voltage Detection Circuit) and Operation

5.2 Power-On Reset Function

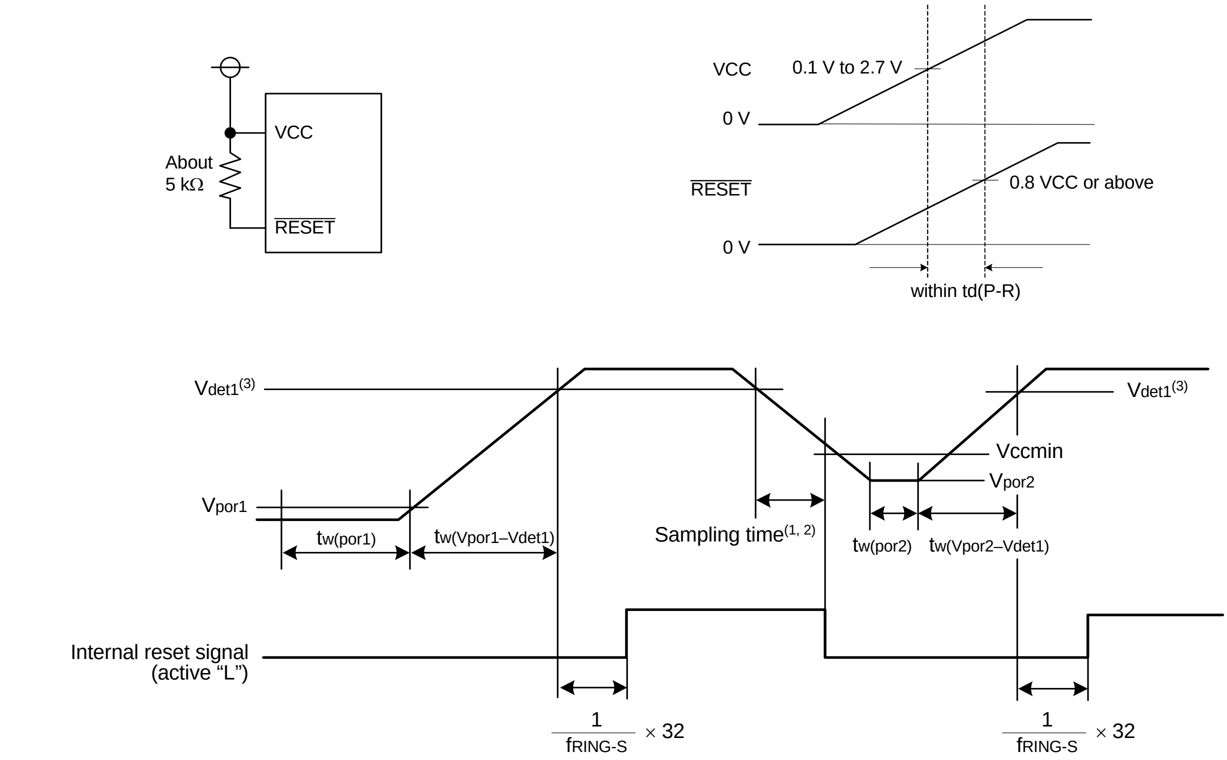
When the $\overline{\text{RESET}}$ pin is connected to the VCC pin via a pull-up resistor of about 5 kΩ, and the VCC pin voltage level rises, the power-on reset function is enabled and the MCU resets its pins, CPU, and SFR. When a capacitor is connected to the $\overline{\text{RESET}}$ pin, always keep the voltage to the $\overline{\text{RESET}}$ pin 0.8VCC or more.

When the input voltage to the VCC pin reaches the Vdet1 level or above, the low-speed on-chip oscillator clock starts counting. When the low-speed on-chip oscillator clock count reaches 32, the internal reset signal is held “H” and the MCU enters the reset sequence (refer to Figure 5.3). The low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU after reset.

Refer to **4. Special Function Registers (SFRs)** for the status of the SFR after power-on reset.

The voltage monitor 1 reset is enabled after power-on reset.

Figure 5.6 shows an Example of Power-On Reset Circuit and Operation.



NOTES:
1. The supply voltage must be held within the MCU’s operating voltage range (Vccmin or above) over the sampling time.
2. A sampling clock can be selected. Refer to **7. Voltage Detection Circuit** for details.
3. Vdet1 indicates voltage detection level for the voltage detection 1 circuit. Refer to **7. Voltage Detection Circuit** for details.
4. Refer to **18. Electrical Characteristics.**

Figure 5.6 Example of Power-On Reset Circuit and Operation

5.3 Voltage Monitor 1 Reset

A reset is applied using the on-chip voltage detection 1 circuit. The voltage detection 1 circuit monitors the input voltage to the VCC pin. The voltage to monitor is Vdet1.

When the input voltage to the VCC pin reaches the Vdet1 level or below, the pins, CPU, and SFR are reset.

When the input voltage to the VCC pin reaches the Vdet1 level or above, the low-speed on-chip oscillator clock starts counting. When the low-speed on-chip oscillator clock count reaches 32, the internal reset signal is held "H" and the MCU enters the reset sequence (refer to Figure 5.3). The low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU after reset.

Refer to **4. Special Function Registers (SFRs)** for the status of the SFR after voltage monitor 1 reset.

The internal RAM is not reset. When the input voltage to the VCC pin reaches the Vdet1 level or below while writing to the internal RAM is in progress, the contents of internal RAM are undefined.

Refer to **7. Voltage Detection Circuit** for details of voltage monitor 1 reset.

5.4 Voltage Monitor 2 Reset

A reset is applied using the on-chip voltage detection 2 circuit. The voltage detection 2 circuit monitors the input voltage to the VCC pin. The voltage to monitor is Vdet2.

When the input voltage to the VCC pin reaches the Vdet2 level or below, pins, CPU, and SFR are reset and the program beginning with the address indicated by the reset vector is executed. After reset, the low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock.

The voltage monitor 2 does not reset some SFRs. Refer to **4. Special Function Registers (SFRs)** for details.

The internal RAM is not reset. When the input voltage to the VCC pin reaches the Vdet2 level or below while writing to the internal RAM is in progress, the contents of internal RAM are undefined.

Refer to **7. Voltage Detection Circuit** for details of voltage monitor 2 reset.

5.5 Watchdog Timer Reset

When the PM12 bit in the PM1 register is set to 1 (reset when watchdog timer underflows), the MCU resets its pins, CPU, and SFR if the watchdog timer underflows. Then the program beginning with the address indicated by the reset vector is executed. After reset, the low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock.

The watchdog timer reset does not reset some SFRs. Refer to **4. Special Function Registers (SFRs)** for details.

The internal RAM is not reset. When the watchdog timer underflows, the contents of internal RAM are undefined.

Refer to **13. Watchdog Timer** for details of watchdog timer.

5.6 Software Reset

When the PM03 bit in the PM0 register is set to 1 (MCU reset), the MCU resets its pins, CPU, and SFR. The program beginning with the address indicated by the reset vector is executed. After reset, the low-speed on-chip oscillator clock divided by 8 is automatically selected for the CPU clock.

The software reset does not reset some SFRs. Refer to **4. Special Function Registers (SFRs)** for details.

The internal RAM is not reset.

6. Programmable I/O Ports

There are 13 programmable Input/Output ports (I/O ports) P1, P3_3 to P3_5, P3_7, and P4_5. P4_2 can be used as an input-only port. Also, P4_6 and P4_7 can be used as input-only ports if the main clock oscillation circuit is not used. Table 6.1 lists an Overview of Programmable I/O Ports.

Table 6.1 Overview of Programmable I/O Ports

Ports	I/O	Type of Output	I/O Setting	Internal Pull-Up Resistor	Drive Capacity Selection
P1	I/O	CMOS3 State	Set per bit	Set every 4 bits(1)	Set every bit(2) of P1_0 to P1_3
P3_3, P4_5	I/O	CMOS3 State	Set per bit	Set every bit(1)	None
P3_4, P3_5, P3_7	I/O	CMOS3 State	Set per bit	Set every 3 bits(1)	None
P4_2, P4_6, P4_7(3)	I	(No output function)	None	None	None

NOTES:

1. In input mode, whether an internal pull-up resistor is connected or not can be selected by registers PUR0 and PUR1.
2. These ports can be used as the LED drive port by setting the DRR register to 1 (high).
3. When the main clock oscillation circuit is not used, P4_6 and P4_7 can be used as input-only ports.

6.1 Functions of Programmable I/O Ports

The PDi_j (j=0 to 7) bit in the PDi (i=1, 3, and 4) register controls I/O of ports P1, P3_3 to P3_5, P3_7, and P4_5. The Pi register consists of a port latch to hold output data and a circuit to read pin states. Figures 6.1 to 6.3 show the Configurations of Programmable I/O Ports.

Table 6.2 lists the Functions of Programmable I/O Ports. Also, Figure 6.5 shows Registers PD1, PD3, and PD4. Figure 6.6 shows Registers P1, P3, and P4, Figure 6.7 shows Registers PUR0 and PUR1, and Figure 6.8 shows the DRR Register.

Table 6.2 Functions of Programmable I/O Ports

Operation when Accessing Pi Register	Value of PDi_j Bit in PDi Register(1)	
	When PDi_j Bit is Set to 0 (Input Mode)	When PDi_j Bit is Set to 1 (Output Mode)
Reading	Read pin input level	Read the port latch
Writing	Write to the port latch	Write to the port latch. The value written to the port latch is output from the pin.

NOTE:

1. Nothing is assigned to bits PD3_0 to PD3_2, PD3_6, PD4_0 to PD4_4, PD4_6, and PD4_7.

6.2 Effect on Peripheral Functions

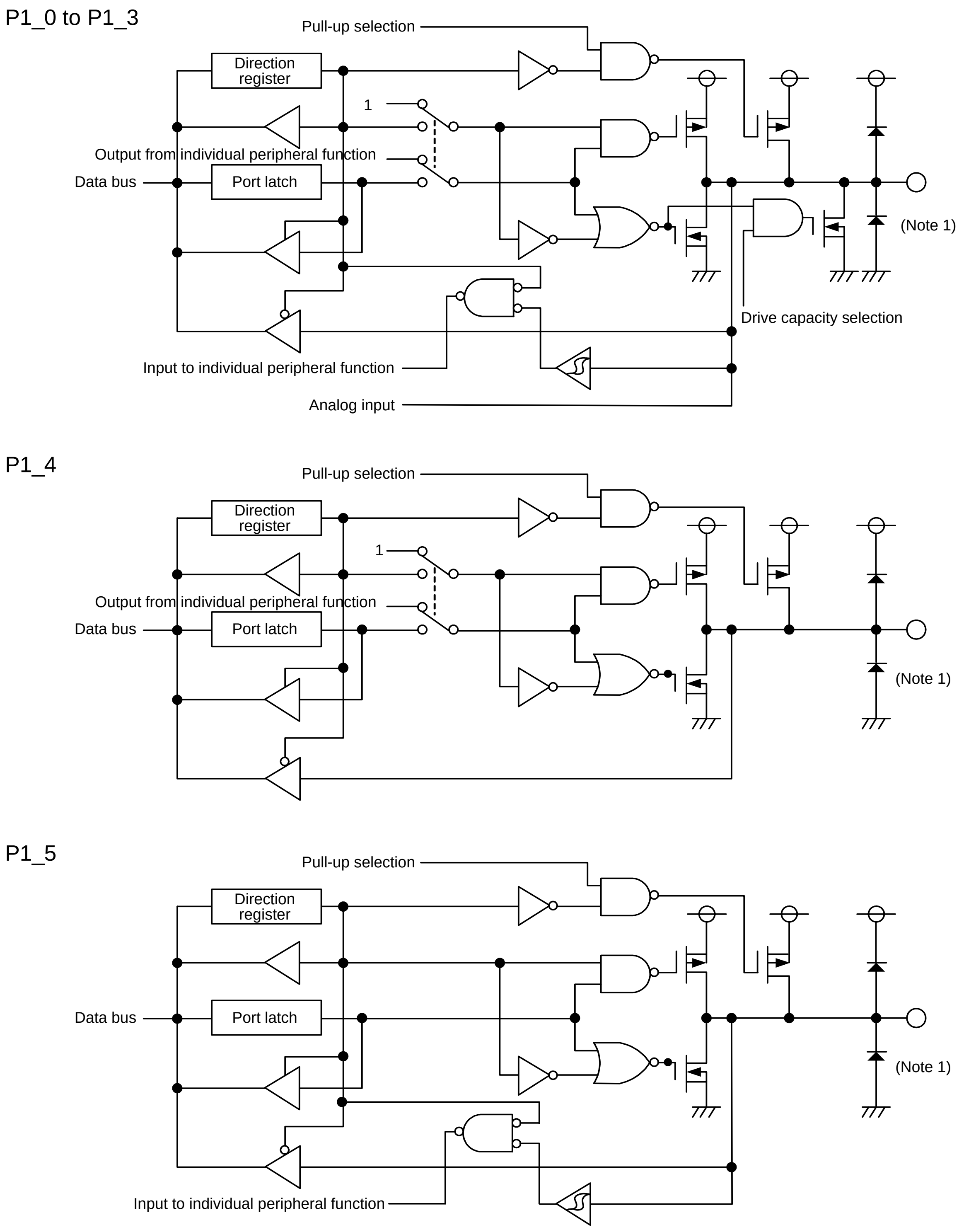
Programmable I/O ports function as I/O ports for peripheral functions (Refer to **Table 1.6 Pin Name Information by Pin Number of PLSP0020JB-A, PRDP0020BA-A packages**). Table 6.3 lists the Settings of PDi_j Bit when Functioning as I/O Ports for Peripheral Functions. Refer to the description of each function for information on how to set peripheral functions.

Table 6.3 Settings of PDi_j Bit when Functioning as I/O Ports for Peripheral Functions

I/O of Peripheral Functions	PPDi_j Bit Settings for Shared Pin Functions
Input	Set this bit to 0 (input mode).
Output	This bit can be set to either 0 or 1 (output regardless of the port setting).

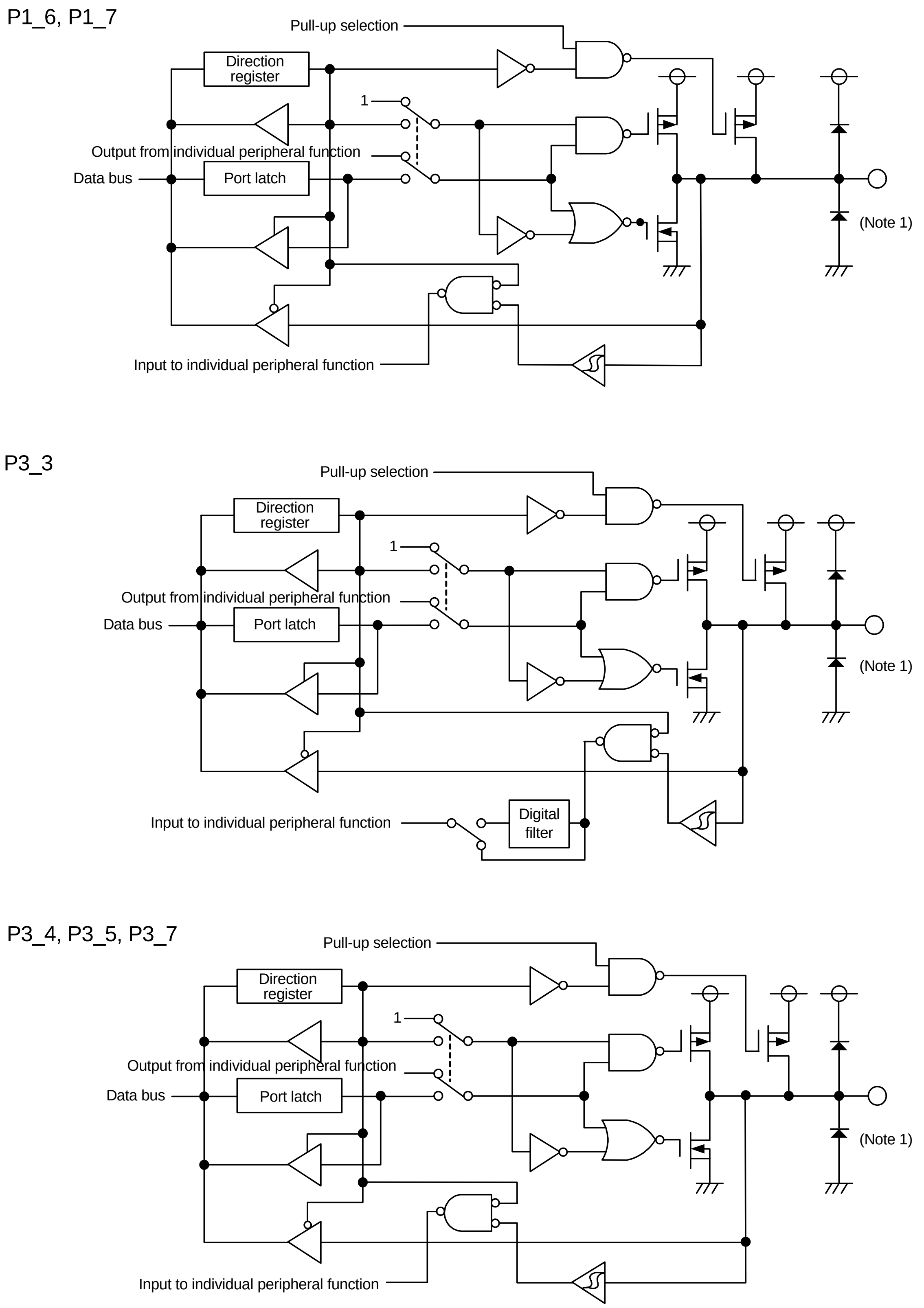
6.3 Pins Other than Programmable I/O Ports

Figure 6.4 shows the Configuration of I/O Pins.



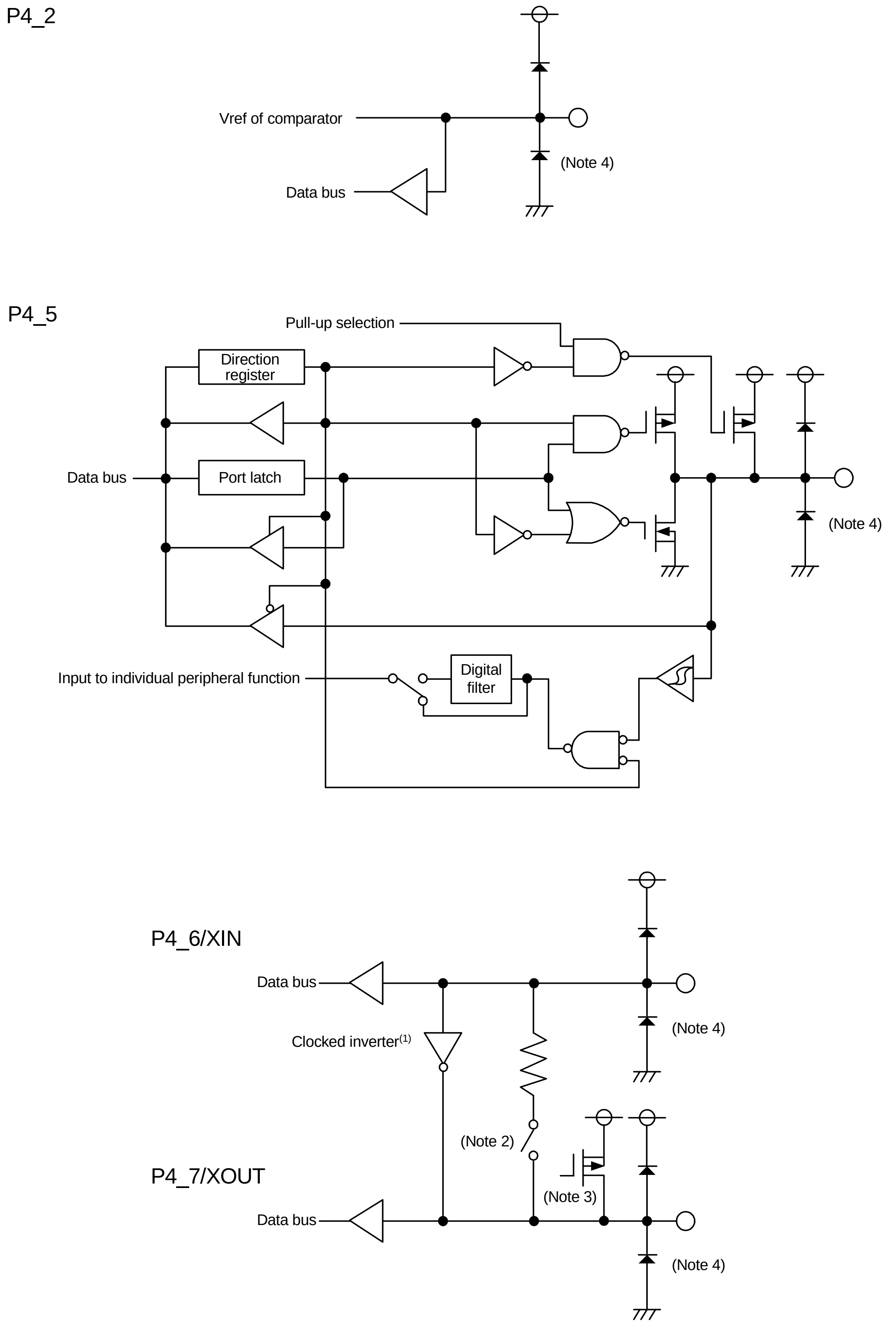
NOTE:
1. ─┤◄── symbolizes a parasitic diode.
Ensure the input voltage to each port will not exceed VCC.

Figure 6.1 Configuration of Programmable I/O Ports (1)


P1_6, P1_7
Pull-up selection
Direction register
1
Output from individual peripheral function
Data bus
Port latch
(Note 1)
Input to individual peripheral function
P3_3
Pull-up selection
Direction register
1
Output from individual peripheral function
Data bus
Port latch
(Note 1)
Input to individual peripheral function
Digital filter
P3_4, P3_5, P3_7
Pull-up selection
Direction register
1
Output from individual peripheral function
Data bus
Port latch
(Note 1)
Input to individual peripheral function

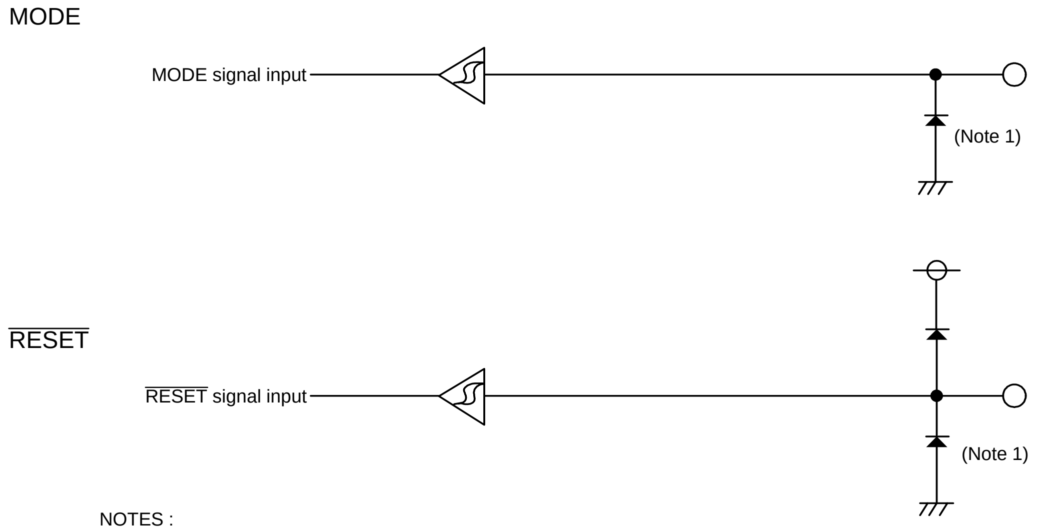
NOTE:
1. symbolizes a parasitic diode.
Ensure the input voltage to each port will not exceed VCC.

Figure 6.2 Configuration of Programmable I/O Ports (2)



NOTES:
1. When CM05 = 1, CM10 = 1, or CM13 = 0, the clocked inverter is cut off.
2. When CM10 = 1 or CM13 = 0, the feedback resistor is disconnected.
3. When CM05 = CM13 = 1 or CM10 = CM13 = 1, this pin is pulled up.
4. ——|◀—— symbolizes a parasitic diode.
Ensure the input voltage to each port does not exceed VCC.

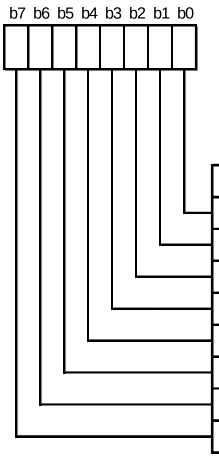
Figure 6.3 Configuration of Programmable I/O Ports (3)


MODE
MODE signal input
(Note 1)
$\overline{\text{RESET}}$
$\overline{\text{RESET}}$ signal input
(Note 1)

NOTES :
1. symbolizes a parasitic diode.
Ensure the input voltage to each port does not exceed VCC.

Figure 6.4 Configuration of I/O Pins

Port Pi Direction Register (i = 1, 3, 4)(1, 2)



Symbol	Address	After Reset
PD1	00E3h	00h
PD3	00E7h	00h
PD4	00EAh	00h

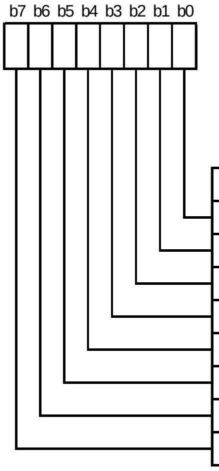
Bit Symbol	Bit Name	Function	RW
PDi_0	Port Pi0 direction bit	0 : Input mode (functions as an input port) 1 : Output mode (functions as an output port)	RW
PDi_1	Port Pi1 direction bit		RW
PDi_2	Port Pi2 direction bit		RW
PDi_3	Port Pi3 direction bit		RW
PDi_4	Port Pi4 direction bit		RW
PDi_5	Port Pi5 direction bit		RW
PDi_6	Port Pi6 direction bit		RW
PDi_7	Port Pi7 direction bit		RW

NOTES:

1. Bits PD3_0 to PD3_2, and PD3_6 in the PD3 register are unavailable on this MCU. If it is necessary to set bits PD3_0 to PD3_2, and PD3_6, set to 0 (input mode). When read, the content is 0.
2. Bits PD4_0 to PD4_4, PD4_6, and PD4_7 in the PD4 register are unavailable on this MCU. If it is necessary to set bits PD4_0 to PD4_4, PD4_6, and PD4_7, set to 0 (input mode). When read, the content is 0.

Figure 6.5 Registers PD1, PD3, and PD4

Port Pi Register (i = 1, 3, 4)(1, 2)



Symbol	Address	After Reset
P1	00E1h	Undefined
P3	00E5h	Undefined
P4	00E8h	Undefined

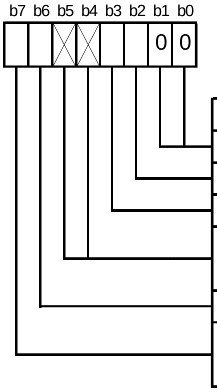
Bit Symbol	Bit Name	Function	RW
Pi_0	Port Pi0 bit	The pin level of any I/O port which is set to input mode can be read by reading the corresponding bit in this register. The pin level of any I/O port which is set to output mode can be controlled by writing to the corresponding bit in this register. 0 : "L" level 1 : "H" level(1)	RW
Pi_1	Port Pi1 bit		RW
Pi_2	Port Pi2 bit		RW
Pi_3	Port Pi3 bit		RW
Pi_4	Port Pi4 bit		RW
Pi_5	Port Pi5 bit		RW
Pi_6	Port Pi6 bit		RW
Pi_7	Port Pi7 bit		RW

NOTES:

1. Bits P3_0 to P3_2, and P3_6 in the P3 register are unavailable on this MCU. If it is necessary to set bits P3_0 to P3_2, and P3_6, set to 0 ("L" level). When read, the content is 0.
2. Bits P4_0 to P4_1, and P4_3 to P4_4 in the P4 register are unavailable on this MCU. If it is necessary to set bits P4_0 to PD4_1, and P4_3 to P4_4, set to 0 ("L" level). When read, the content is 0.

Figure 6.6 Registers P1, P3, and P4

Pull-Up Control Register 0



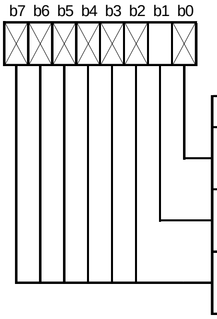
Symbol	Address	After Reset
PUR0	00FCh	00XX0000b

Bit Symbol	Bit Name	Function	RW
(b1-b0)	Reserved bits	Set to 0.	RW
PU02	P1_0 to P1_3 pull-up[1]	0 : Not pulled up 1 : Pulled up	RW
PU03	P1_4 to P1_7 pull-up[1]		RW
— (b5-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		—
PU06	P3_3 pull-up[1]	0 : Not pulled up 1 : Pulled up	RW
PU07	P3_4 to P3_5, and P3_7 pll-up[1]		RW

NOTE:

1. When this bit is set to 1 (pulled up), the pin w hose direction bit is set to 0 (input mode) is pulled up.

Pull-Up Control Register 1



Symbol	Address	After Reset
PUR1	00FDh	XXXXXX0Xb

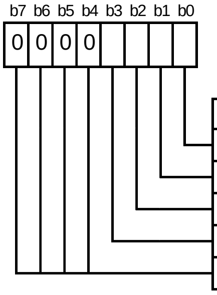
Bit Symbol	Bit Name	Function	RW
— (b0)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		—
PU11	P4_5 pull-up[1]	0 : Not pulled up 1 : Pulled up	RW
— (b7-b2)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		—

NOTE:

1. When the PU11 bit is set to 1 (pulled up), and the PD4_5 bit is set to 0 (input mode), the P4_5 pin is pulled up.

Figure 6.7 Registers PUR0 and PUR1

Port P1 Drive Capacity Control Register



Symbol	Address	After Reset
DRR	00FEh	00h

Bit Symbol	Bit Name	Function	RW
DRR0	P1_0 drive capacity	Set P1 N-channel output transistor drive capacity. 0 : Low 1 : High	RW
DRR1	P1_1 drive capacity		RW
DRR2	P1_2 drive capacity		RW
DRR3	P1_3 drive capacity		RW
(b7-b4)	Reserved bits	Set to 0.	RW

Figure 6.8 DRR Register

6.4 Port Settings

Tables 6.4 to 6.17 list the port settings.

Table 6.4 Port P1_0/$\overline{\text{KI0}}$/AN8/CMP0_0

Register	PD1	PUR0	DRR	KIEN	ADCON0	TCOUT	Function
Bit	PD1_0	PU02	DRR0	KI0EN	CH2, CH1, CH0, ADGSEL0	TCOUT0	
Setting Value	0	0	X	X	XXXX	0	Input port (not pulled up)
	0	1	X	X	XXXX	0	Input port (pulled up)
	0	0	X	1	XXXX	0	$\overline{\text{KI0}}$ input
	0	0	X	X	1001b	0	Comparator input (AN8)
	1	X	0	X	XXXX	0	Output port
	1	X	1	X	XXXX	0	Output port (high drive)
	X	X	X	X	XXXX	1	CMP0_0 output

X: 0 or 1

Table 6.5 Port P1_1/$\overline{\text{KI1}}$/AN9/CMP0_1

Register	PD1	PUR0	DRR	KIEN	ADCON0	TCOUT	Function
Bit	PD1_1	PU02	DRR1	KI1EN	CH2, CH1, CH0, ADGSEL0	TCOUT1	
Setting Value	0	0	X	X	XXXX	0	Input port (not pulled up)
	0	1	X	X	XXXX	0	Input port (pulled up)
	0	0	X	1	XXXX	0	$\overline{\text{KI1}}$ input
	0	0	X	X	1011b	0	Comparator input (AN9)
	1	X	0	X	XXXX	0	Output port
	1	X	1	X	XXXX	0	Output port (high drive)
	X	X	X	X	XXXX	1	CMP0_1 output

X: 0 or 1

Table 6.6 Port P1_2/$\overline{\text{KI2}}$/AN10/CMP0_2

Register	PD1	PUR0	DRR	KIEN	ADCON0	TCOUT	Function
Bit	PD1_2	PU02	DRR2	KI2EN	CH2, CH1, CH0, ADGSEL0	TCOUT2	
Setting Value	0	0	X	X	XXXX	0	Input port (not pulled up)
	0	1	X	X	XXXX	0	Input port (pulled up)
	0	0	X	1	XXXX	0	$\overline{\text{KI2}}$ input
	0	0	X	X	1101b	0	Comparator input (AN10)
	1	X	0	X	XXXX	0	Output port
	1	X	1	X	XXXX	0	Output port (high drive)
	X	X	X	X	XXXX	1	CMP0_2 input

X: 0 or 1

Table 6.7 Port P1_3/$\overline{\text{KI3}}$/AN11/TZOUT

Register	PD1	PUR0	DRR	KIEN	ADCON0	TZMR	TZOC	Function
Bit	PD1_3	PU02	DRR3	KI3EN	CH2, CH1, CH0, ADGSEL0	TZMOD1, TZMOD0	TZOCNT	
Setting Value	0	0	X	X	XXXX	00b	X	Input port (not pulled up)
	0	1	X	X	XXXX	00b	X	Input port (pulled up)
	0	0	X	1	XXXX	00b	X	$\overline{\text{KI3}}$ input
	0	0	X	X	1111b	00b	X	Comparator input (AN11)
	1	X	0	X	XXXX	00b	X	Output port
	1	X	1	X	XXXX	00b	X	Output port (high drive)
	X	X	0	X	XXXX	01b	1	Output port
	X	X	1	X	XXXX	01b	1	Output port (high drive)
	X	X	X	X	XXXX	01b	0	TZOUT output
	X	X	X	X	XXXX	1Xb	X	TZOUT output

X: 0 or 1

Table 6.8 Port P1_4/TXD0

Register	PD1	PUR0	U0MR	U0C0	Function
Bit	PD1_4	PU03	SMD2 to SMD0	NCH	
Setting Value	0	0	000b	X	Input port (not pulled up)
	0	1	000b	X	Input port (pulled up)
	1	X	000b	X	Output port
	X	X	001b	0	TXD0 output, CMOS output
			100b		
			101b		
			110b		
	X	X	001b	1	TXD0 output, N-channel open output
			100b		
			101b		
			110b		

X: 0 or 1

Table 6.9 Port P1_5/RXD0/CNTR01/$\overline{\text{INT11}}$

Register	PD1	PUR0	UCON	TXMR	Function
Bit	PD1_5	PU03	CNTRSEL	TXMOD1, TXMOD0	
Setting Value	0	0	X	XX	Input port (not pulled up)
	0	1	X	XX	Input port (pulled up)
	0	X	X	Other than 01b	RXD0 input
	0	X	1	Other than 01b	CNTR01/$\overline{\text{INT11}}$ input
	1	X	X	Other than 01b	Output port
	1	X	1	01b	CNTR01 output

X: 0 or 1

Table 6.10 Port P1_6/CLK0

Register	PD1	PUR0	U0MR	Function
Bit	PD1_6	PU03	SMD2, SMD0, CKDIR	
Setting Value	0	0	Other than 010b	Input port (not pulled up)
	0	1	Other than 010b	Input port (pulled up)
	0	0	XX1	CLK0 (external clock) input
	1	X	Other than 010b	Output port
	X	X	010b	CLK0 (internal clock) output

X: 0 or 1

Table 6.11 Port P1_7/CNTR00/$\overline{\text{INT10}}$

Register	PD1	PUR0	TXMR	UCON	Function
Bit	PD1_7	PU03	TXMOD1, TXMOD0	CNTRSEL	
Setting Value	0	0	Other than 01b	X	Input port (not pulled up)
	0	1	Other than 01b	X	Input port (pulled up)
	0	0	Other than 01b	0	CNTR00/$\overline{\text{INT10}}$ input
	1	X	Other than 01b	X	Output port
	X	X	01b	0	CNTR00 output

X: 0 or 1

Table 6.12 Port P3_3/TCIN/$\overline{\text{INT3}}$/CMP1_0

Register	PD3	PUR0	TCOUT	Function
Bit	PD3_3	PU06	TCOUT3	
Setting Value	0	0	0	Input port (not pulled up)
	0	1	0	Input port (pulled up)
	1	X	0	Output port
	X	X	1	CMP1_0 output
	0	X	0	TCIN input/$\overline{\text{INT3}}$

X: 0 or 1

Table 6.13 Port P3_4/CMP1_1

Register	PD3	PUR0	TCOUT	Function
Bit	PD3_4	PU07	TCOUT4	
Setting Value	0	0	0	Input port (not pulled up)
	0	1	0	Input port (pulled up)
	1	X	0	Output port
	X	X	1	CMP1_1 output

X: 0 or 1

Table 6.14 Port P3_5/CMP1_2

Register	PD3	PUR0	TCOUT	Function
Bit	PD3_5	PU07	TCOUT5	
Setting Value	0	0	0	Input port (not pulled up)
	0	1	0	Input port (pulled up)
	1	X	0	Output port
	X	X	1	CMP1_2 output

X: 0 or 1

Table 6.15 Port P3_7/$\overline{\text{CNTR0}}$/TXD1

Register	PD3	PUR0	U1MR	TXMR	UCON	Function
Bit	PD3_7	PU07	SMD2 to SMD0	TXOCNT	U1SEL1, U1SEL0	
Setting Value	0	0	000b	0	0X	Input port (not pulled up)
	0	1	000b	0	0X	Input port (pulled up)
	1	X	000b	0	0X	Output port
	X	X	001b	X	11b	TXD1 output pin
			100b			
			101b			
			110b			
	X	X	000b	1	XX	$\overline{\text{CNTR0}}$ output pin

X: 0 or 1

Table 6.16 Port XIN/P4_6, XOUT/P4_7

Register	CM1	CM1	CM0	Circuit specification		Function
Bit	CM13	CM10	CM05	Oscillation buffer	Feedback resistance	
Setting Value	1	1	1	OFF	OFF	XIN-XOUT oscillation stop
	1	0	1	OFF	ON	External input to XIN pin, “H” output from XOUT pin
	1	0	1	OFF	ON	XIN-XOUT oscillation stop
	1	0	0	ON	ON	XIN-XOUT oscillation
	0	X	X	OFF	OFF	Input port

X: 0 or 1

Table 6.17 Port P4_5/$\overline{\text{INT0}}$/RXD1

Register	PD4	PUR1	UCON	INTEN	Function
Bit	PD4_5	PU11	U1SEL1, U1SEL0	INT0EN	
Setting Value	0	0	00b	0	Input port (not pulled up)
	0	1	00b	0	Input port (pulled up)
	0	0	00b	1	$\overline{\text{INT0}}$ input
	X	0	01b	0	RXD1 input
			10b		
	1	X	00b	X	Output port

X: 0 or 1

6.5 Unassigned Pin Handling

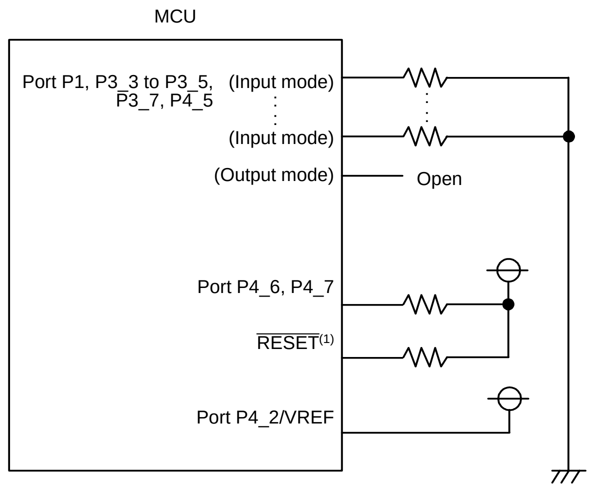
Table 6.18 lists Unassigned Pin Handling. Figure 6.9 shows Unassigned Pin Handling.

Table 6.18 Unassigned Pin Handling

Pin Name	Connection
Ports P1, P3_3 to P3_5, P3_7, P4_5	• After setting to input mode, connect each pin to VSS via a resistor (pull-down) or connect each pin to VCC via a resistor (pull-up).(2) • After setting to output mode, leave these pins open.(1, 2)
Ports P4_6, P4_7	Connect to VCC via a pull-up resistor(2)
Port P4_2/VREF	Connect to VCC
$\overline{\text{RESET}}$ (3)	Connect to VCC via a pull-up resistor(2)

NOTES:

1. If these ports are set to output mode and left open, they remain in input mode until they are switched to output mode by a program. The voltage level of these pins may be undefined and the power supply current may increase while the ports remain in input mode.
   The content of the direction registers may change due to noise or program runaway caused by noise. In order to enhance program reliability, the program should periodically repeat the setting of the direction registers.
2. Connect these unassigned pins to the MCU using the shortest wire length (2 cm or less) possible.
3. When the power-on reset function is in use.



NOTE:
1. When the power-on reset function is in use.

Figure 6.9 Unassigned Pin Handling

7. Voltage Detection Circuit

The voltage detection circuit monitors the input voltage to the VCC pin. This circuit can be used to monitor the VCC input voltage by a program. Alternately, voltage monitor 1 reset, voltage monitor 2 interrupt, and voltage monitor 2 reset can also be used.
Table 7.1 lists the Specifications of Voltage Detection Circuit and Figures 7.1 to 7.3 show the Block Diagrams. Figures 7.4 to 7.6 show the Associated Registers.

Table 7.1 Specifications of Voltage Detection Circuit

Item		Voltage Detection 1	Voltage Detection 2
VCC monitor	Voltage to monitor	Vdet1	Vdet2
	Detection target	Passing through Vdet1 by rising or falling	Passing through Vdet2 by rising or falling
	Monitor	None	VCA13 bit in VCA1 register
			Whether VCC is higher or lower than Vdet2
Process when voltage is detected	Reset	Voltage monitor 1 reset	Voltage monitor 2 reset
		Reset at Vdet1 > VCC; restart CPU operation at VCC > Vdet1	Reset at Vdet2 > VCC; restart CPU operation after a specified time
	Interrupt	None	Voltage monitor 2 interrupt
			Interrupt request at Vdet2 > VCC and VCC > Vdet2 when digital filter is enabled; interrupt request at Vdet2 > VCC or VCC > Vdet2 when digital filter is disabled
Digital filter	Switch enabled/disabled	Available	Available
	Sampling time	(Divide-by-n of fRING-S) x 4 n: 1, 2, 4, and 8	(Divide-by-n of fRING-S) x 4 n: 1, 2, 4, and 8

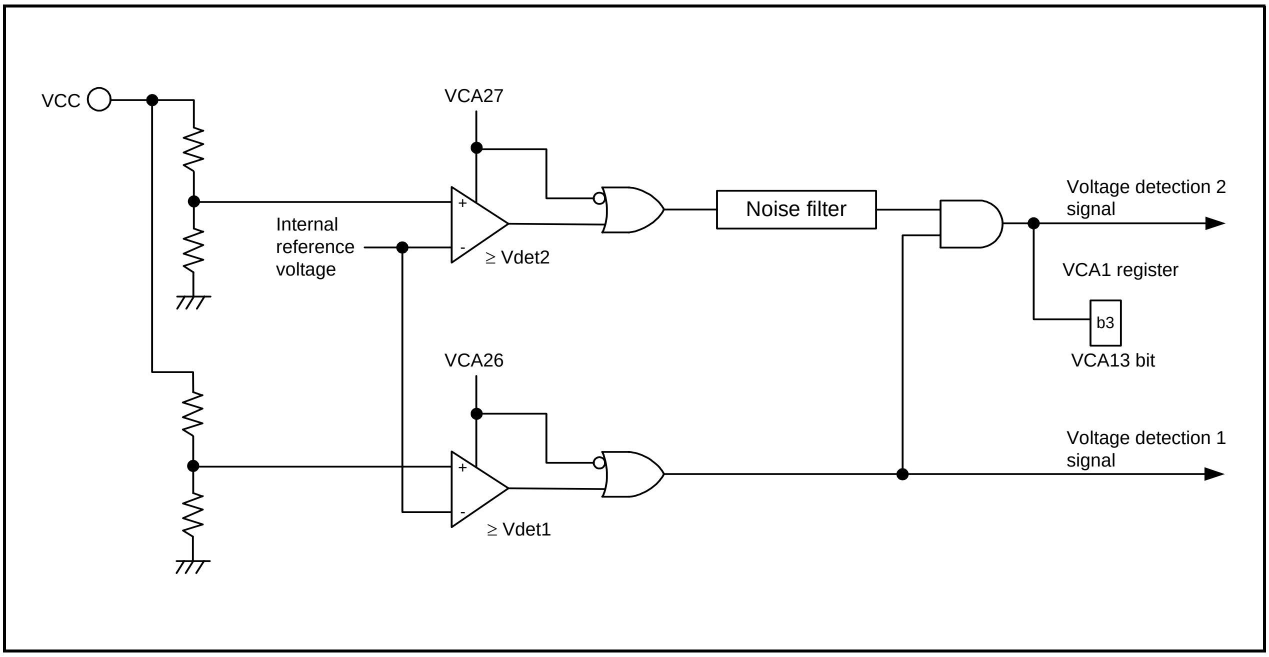


Figure 7.1 Block Diagram of Voltage Detection Circuit

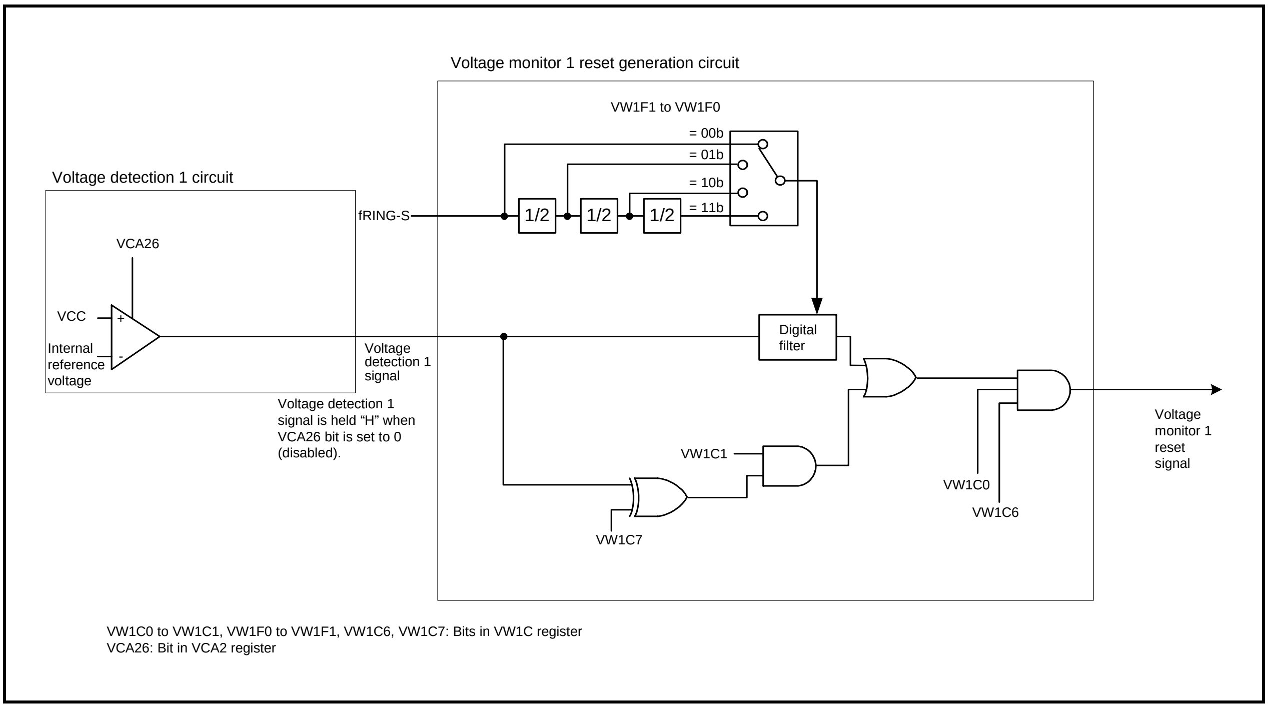
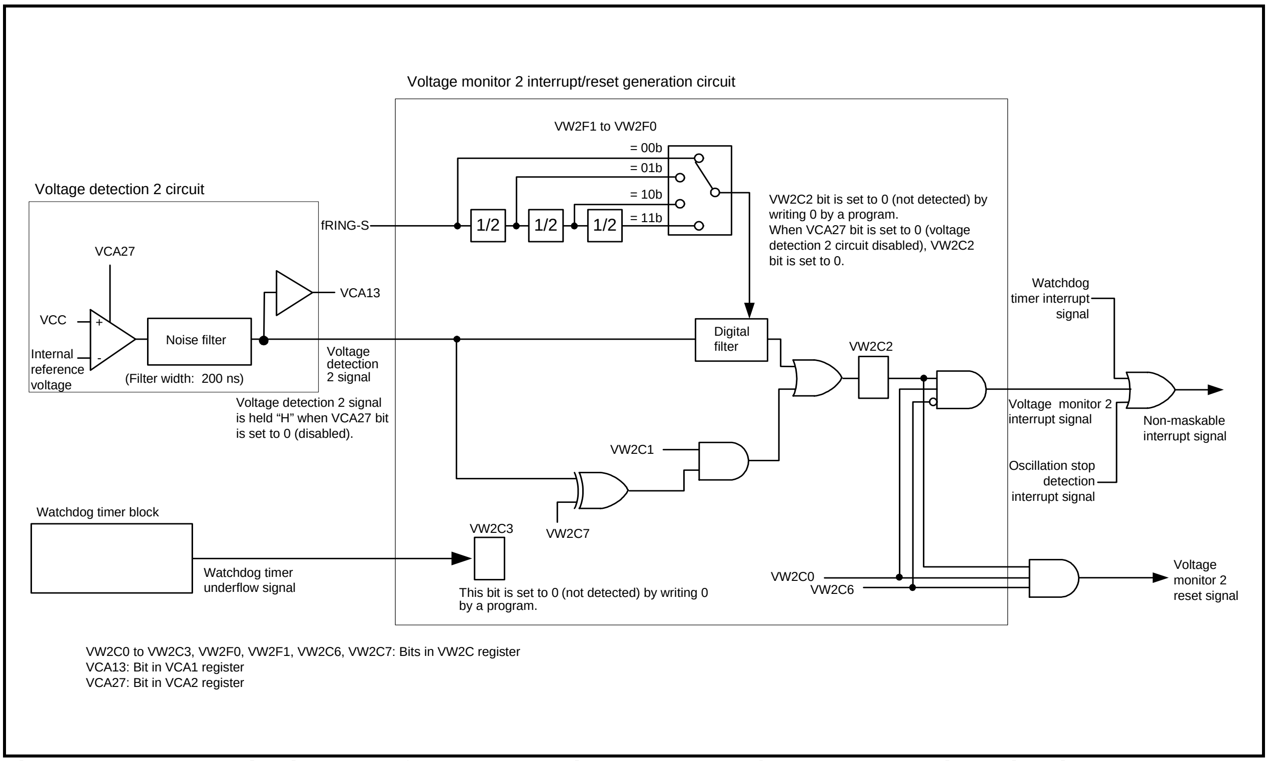


Figure 7.2 Block Diagram of Voltage Monitor 1 Reset Generation Circuit


Voltage monitor 2 interrupt/reset generation circuit
VW2F1 to VW2F0
= 00b
= 01b
= 10b
= 11b
fRING-S
1/2
1/2
1/2
VW2C2 bit is set to 0 (not detected) by writing 0 by a program.
When VCA27 bit is set to 0 (voltage detection 2 circuit disabled), VW2C2 bit is set to 0.
Voltage detection 2 circuit
VCA27
VCC
+
-
Internal reference voltage
Noise filter
(Filter width: 200 ns)
VCA13
Voltage detection 2 signal
Voltage detection 2 signal is held "H" when VCA27 bit is set to 0 (disabled).
Digital filter
VW2C2
Watchdog timer interrupt signal
Voltage monitor 2 interrupt signal
Non-maskable interrupt signal
Oscillation stop detection interrupt signal
VW2C1
VW2C7
Watchdog timer block
Watchdog timer underflow signal
VW2C3
This bit is set to 0 (not detected) by writing 0 by a program.
VW2C0
VW2C6
Voltage monitor 2 reset signal

VW2C0 to VW2C3, VW2F0, VW2F1, VW2C6, VW2C7: Bits in VW2C register
VCA13: Bit in VCA1 register
VCA27: Bit in VCA2 register

Figure 7.3 Block Diagram of Voltage Monitor 2 Interrupt/Reset Generation Circuit

Voltage Detection Register 1

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0		0	0	0

Symbol	Address	After Reset[2]
VCA1	0031h	00001000b

Bit Symbol	Bit Name	Function	RW
— (b2-b0)	Reserved bits	Set to 0.	RW
VCA13	Voltage detection 2 signal monitor flag[1]	0 : VCC < Vdet2 1 : VCC ≥ Vdet2 or voltage detection 2 circuit disabled	RO
— (b7-b4)	Reserved bits	Set to 0.	RW

NOTES:

1. The VCA13 bit is enabled w hen the VCA27 bit in the VCA2 register is set to 1 (voltage detection 2 circuit enabled). The VCA13 bit is set to 1 (VCC ≥ Vdet 2) w hen the VCA27 bit in the VCA2 register is set to 0 (voltage detection 2 circuit disabled).
2. The softw are reset, w atchdog timer reset, and voltage monitor 2 reset do not affect this register.

Voltage Detection Register 2[1]

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0	0	0	0

Symbol	Address	After Reset[4]
VCA2	0032h	Hardw are reset : 00h Pow er-on reset, voltage monitor 1 reset : 01000000b

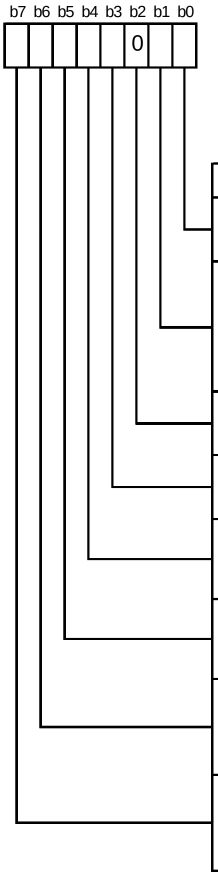
Bit Symbol	Bit Name	Function	RW
— (b5-b0)	Reserved bits	Set to 0.	RW
VCA26	Voltage detection 1 enable bit[2]	0 : Voltage detection 1 circuit disabled 1 : Voltage detection 1 circuit enabled	RW
VCA27	Voltage detection 1 enable bit[3]	0 : Voltage detection 2 circuit disabled 1 : Voltage detection 2 circuit enabled	RW

NOTES:

1. Set the PRC3 bit in the PRCR register to 1 (w rite enable) before w riting to this register.
2. To use the voltage monitor 1 reset, set the VCA26 bit to 1.
   After the VCA26 bit is set to 1 from 0, the voltage detection circuit w aits for td(E-A) to elapse before starting operation.
3. To use the voltage monitor 2 interrupt/reset or the VCA13 bit in the VCA1 register, set the VCA27 bit to 1.
   After the VCA27 bit is set to 1 from 0, the voltage detection circuit w aits for td(E-A) to elapse before starting operation.
4. Softw are reset, w atchdog timer reset, and voltage monitor 2 reset do not affect this register.

Figure 7.4 Registers VCA1 and VCA2

Voltage Monitor 1 Circuit Control Register[1]



Symbol	Address	After reset[2]
VW1C	0036h	Hardware reset : 0000X000b Power-on reset, voltage monitor 1 reset : 0100X001b

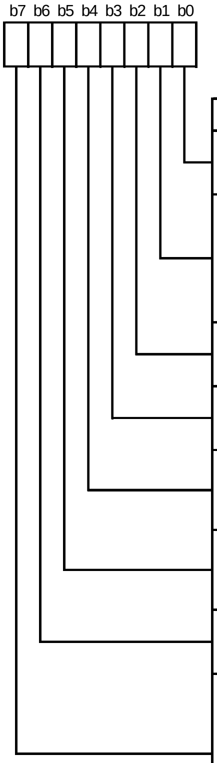
Bit Symbol	Bit Name	Function	RW
VW1C0	Voltage monitor 1 reset enable bit[3]	0 : Disable 1 : Enable	RW
VW1C1	Voltage monitor 1 digital filter disable mode select bit	0 : Digital filter enabled mode (digital filter circuit enabled) 1 : Digital filter disabled mode (digital filter circuit disabled)	RW
VW1C2	Reserved bit	Set to 0.	RW
— (b3)	Reserved bit	When read, the content is undefined.	RO
VW1F0	Sampling clock select bits	b5 b4 0 0 : fRING-S divided by 1 0 1 : fRING-S divided by 2 1 0 : fRING-S divided by 4 1 1 : fRING-S divided by 8	RW
VW1F1			RW
VW1C6	Voltage monitor 1 circuit mode select bit	When the VW1C0 bit is set to 1 (voltage monitor 1 reset enabled), set to 1.	RW
VW1C7	Voltage monitor 1 reset generation condition select bit	When the VW1C1 bit is set to 1 (digital filter disabled mode), set to 1.	RW

NOTES:

1. Set the PRC3 bit in the PRCR register to 1 (write enable) before writing to this register.
   When rewriting the VW1C register, the VW1C2 bit may be set to 1. Set the VW1C2 bit to 0 after rewriting the VW1C register.
2. The value remains unchanged after a software reset, watchdog timer reset, or voltage monitor 2 reset.
3. The VW1C0 bit is enabled when the VCA26 bit in the VCA2 register is set to 1 (voltage detection 1 circuit enabled). Set the VW1C0 bit to 0 (disable), when the VCA26 bit is set to 0 (voltage detection 1 circuit disabled).

Figure 7.5 VW1C Register

Voltage Monitor 2 Circuit Control Register[1]



Symbol	Address	After Reset[8]
VW2C	0037h	00h

Bit Symbol	Bit Name	Function	RW
VW2C0	Voltage monitor 2 interrupt/reset enable bit[6, 10]	0 : Disable 1 : Enable	RW
VW2C1	Voltage monitor 2 digital filter disable mode select bit[2]	0 : Digital filter enabled mode (digital filter circuit enabled) 1 : Digital filter disabled mode (digital filter circuit disabled)	RW
VW2C2	Voltage change detection flag[3, 4, 8]	0 : Not detected 1 : Vdet2 crossing detected	RW
VW2C3	WDT detection flag[4, 8]	0 : Not detected 1 : Detected	RW
VW2F0	Sampling clock select bits	b5 b4 0 0 : fRING-S divide by 1 0 1 : fRING-S divide by 2 1 0 : fRING-S divide by 4 1 1 : fRING-S divide by 8	RW
VW2F1			RW
VW2C6	Voltage monitor 2 circuit mode select bit[5]	0 : Voltage monitor 2 interrupt mode 1 : Voltage monitor 2 reset mode	RW
VW2C7	Voltage monitor 2 interrupt/reset generation condition select bit[7, 9]	0 : When VCC reaches Vdet2 or above. 1 : When VCC reaches Vdet2 or below .	RW

NOTES:

1. Set the PRC3 bit in the PRCR register to 1 (rewrite enable) before writing to this register.
   When rewriting the VW2C register, the VW2C2 bit may be set to 1. Set the VW2C2 bit to 0 after rewriting the VW2C register.
2. When the voltage monitor 2 interrupt is used to exit stop mode and to return again, write 0 to the VW2C1bit before writing 1.
3. This bit is enabled when the VCA27 bit in the VCA2 register is set to 1 (voltage detection 2 circuit enabled).
4. Set this bit to 0 by a program. When 0 is written by a program, it is set to 0 (and remains unchanged even if 1 is written to it).
5. This bit is enabled when the VW2C0 bit is set to 1 (voltage monitor 2 interrupt/enabled reset).
6. The VW2C0 bit is enabled when the VCA27 bit in the VCA2 register is set to 1 (voltage detection 2 circuit enabled). Set the VW2C0 bit to 0 (disable) when the VCA27 bit is set to 0 (voltage detection 2 circuit disabled).
7. The VW2C7 bit is enabled when the VW2C1 bit is set to 1 (digital filter disabled mode).
8. Bits VW2C2 and VW2C3 remain unchanged after a software reset, watchdog timer reset, or voltage monitor 2
9. When the VW2C6 bit is set to 1 (voltage monitor 2 reset mode), set the VW2C7 bit to 1 (when VCC reaches Vdet2 or below). (Do not set to 0.)
10. Set the VW2C0 bit to 0 (disabled) when the VCA13 bit in the VCA1 register is set to 1 (VCC $\geq$ Vdet2 or voltage detection 2 circuit disabled), the VW2C1 bit is set to 1 (digital filter disabled mode), and the VW2C7 bit is set to 0 (when VCC reaches Vdet2 or above).
    Set the VW2C0 bit to 0 (disabled) when the VCA13 bit is set to 0 (VCC < Vdet2), the VW2C1 bit is set to 1 (digital filter disabled mode), and the VW2C7 bit is set to 1 (when VCC reaches Vdet2 or below).

Figure 7.6 VW2C Register

7.1 VCC Input Voltage

7.1.1 Monitoring Vdet1

Vdet1 cannot be monitored.

7.1.2 Monitoring Vdet2

Set the VCA27 bit in the VCA2 register to 1 (voltage detection 2 circuit enabled). After td(E-A) has elapsed (refer to **18. Electrical Characteristics**), Vdet2 can be monitored by the VCA13 bit in the VCA1 register.

7.1.3 Digital Filter

A digital filter can be used for monitoring the VCC input voltage. When the VW1C1 bit in the VW1C register is set to 0 (digital filter enabled) for the voltage monitor 1 circuit and the VW2C1 bit in the VW2C register is set to 0 (digital filter enabled) for the voltage monitor 2 circuit, the digital filter circuit is enabled.
fRING-S divided by 1, 2, 4, or 8 may be selected as a sampling clock.
The level of VCC input voltage is sampled every sampling clock cycle, and when the sampled input level matches two times, the internal reset signal changes to “L” or a voltage monitor 2 interrupt request is generated.

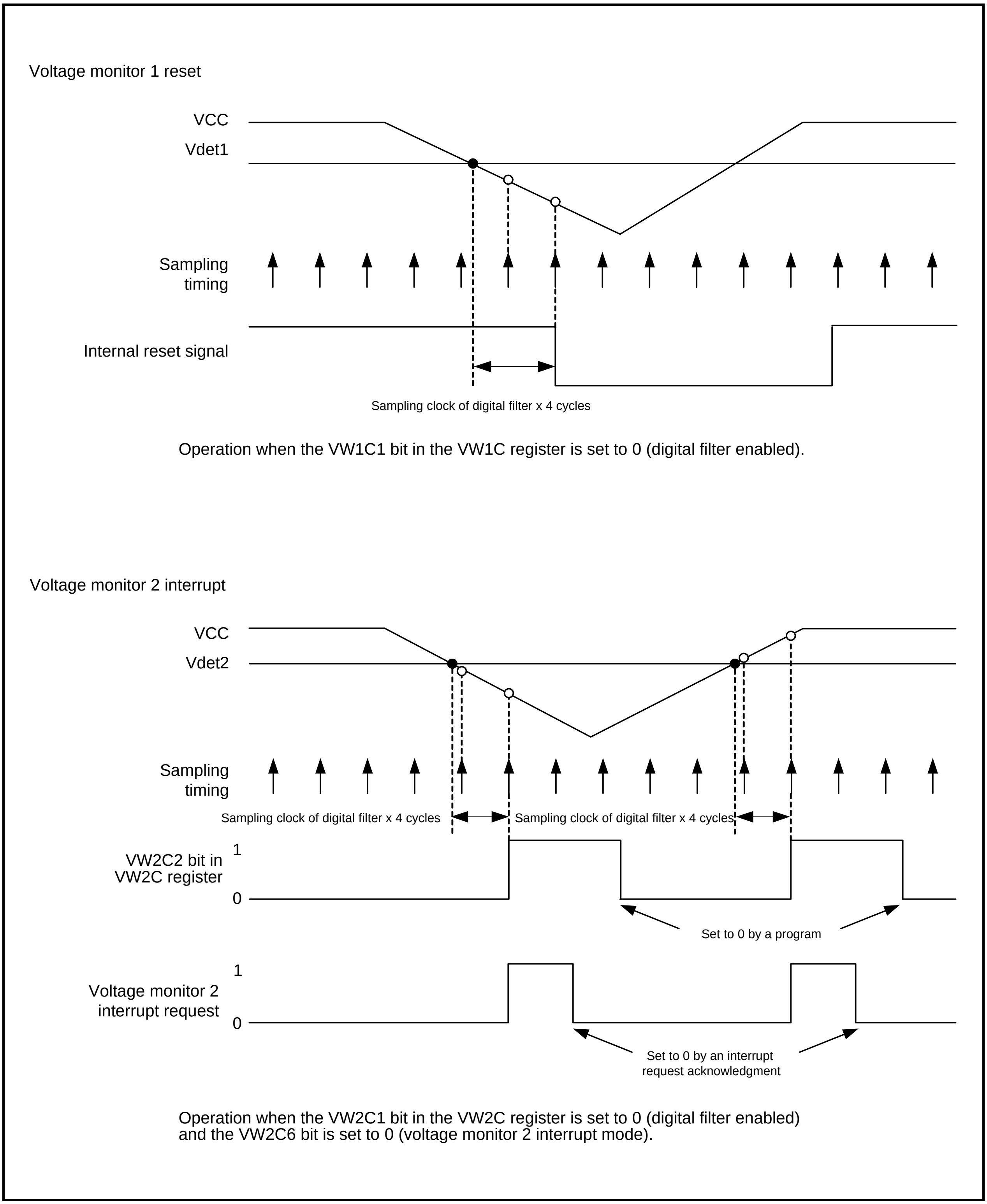

Voltage monitor 1 reset
VCC
Vdet1
Sampling timing
Internal reset signal
Sampling clock of digital filter x 4 cycles
Operation when the VW1C1 bit in the VW1C register is set to 0 (digital filter enabled).
Voltage monitor 2 interrupt
VCC
Vdet2
Sampling timing
Sampling clock of digital filter x 4 cycles
Sampling clock of digital filter x 4 cycles
VW2C2 bit in VW2C register
1
0
Set to 0 by a program
Voltage monitor 2 interrupt request
1
0
Set to 0 by an interrupt request acknowledgment
Operation when the VW2C1 bit in the VW2C register is set to 0 (digital filter enabled) and the VW2C6 bit is set to 0 (voltage monitor 2 interrupt mode).

Figure 7.7 Operating Example of Digital Filter

7.2 Voltage Monitor 1 Reset

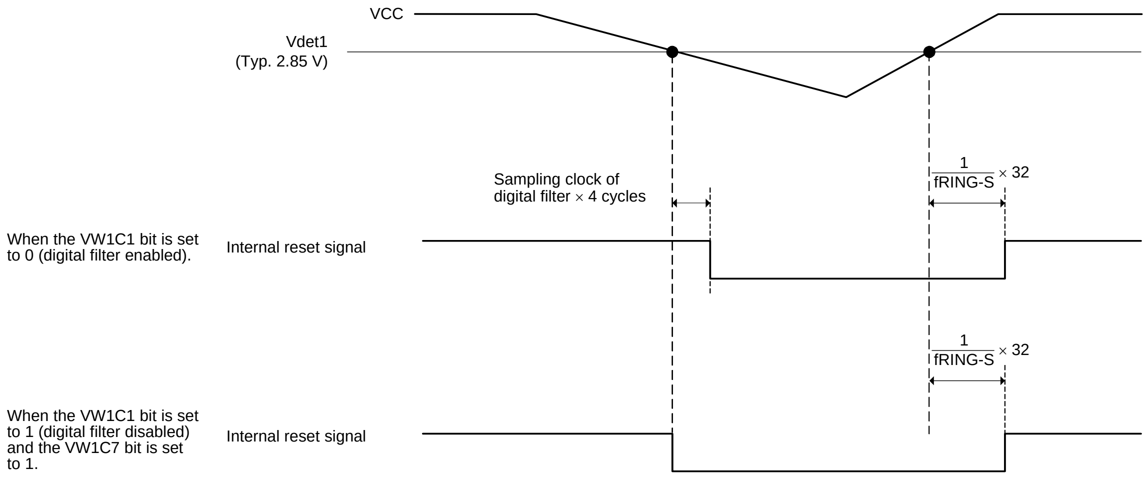
Table 7.2 lists the Setting Procedure of Voltage Monitor 1 Reset Associated Bits and Figure 7.8 shows an Operating Example of Voltage Monitor 1 Reset. To use voltage monitor 1 reset to exit stop mode, set the VW1C1 bit in the VW1C register to 1 (digital filter disabled).

Table 7.2 Setting Procedure of Voltage Monitor 1 Reset Associated Bits

Step	When Using Digital Filter	When Not Using Digital Filter
1	Set the VCA26 bit in the VCA2 register to 1 (voltage detection 1 circuit enabled).	
2	Wait for td(E-A)	
3[1]	Select the sampling clock of the digital filter by bits VW1F0 to VW1F1 in the VW1C register.	Set the VW1C7 bit in the VW1C register to 1.
4[1]	Set the VW1C1 bit in the VW1C register to 0 (digital filter enabled).	Set the VW1C1 bit in the VW1C register to 1 (digital filter disabled).
5[1]	Set the VW1C6 bit in the VW1C register to 1 (voltage monitor 1 reset mode).	
6	Set the VW1C2 bit in the VW1C register to 0.	
7	Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on).	–
8	Wait for 4 cycles of the sampling clock of the digital filter.	– (No wait time)
9	Set the VW1C0 bit in the VW1C register to 1 (voltage monitor 1 reset enabled).	

NOTE:

1. When the VW1C0 bit is set to 0 (disabled), steps 3, 4, and 5 can be executed simultaneously (with 1 instruction).


VCC
Vdet1
(Typ. 2.85 V)
Sampling clock of
digital filter × 4 cycles
1 / fRING-S × 32
When the VW1C1 bit is set to 0 (digital filter enabled).
Internal reset signal
1 / fRING-S × 32
When the VW1C1 bit is set to 1 (digital filter disabled) and the VW1C7 bit is set to 1.
Internal reset signal

VW1C1 and VW1C7: Bits in VW1C register

The above applies under the following conditions.
- VCA26 bit in VCA2 register = 1 (voltage detection 1 circuit enabled)
- VW1C0 bit in VW1C register = 1 (voltage monitor 1 reset enabled)
- VW1C6 bit in VW1C register = 1 (voltage monitor 1 reset mode)

When the internal reset signal is held “L”, the pins, CPU and SFR are reset.
The internal reset signal level changes from “L” to “H”, and a program is executed beginning with the address indicated by the reset vector.
Refer to **4. Special Function Registers (SFRs),** for the SFR status after reset.

Figure 7.8 Operating Example of Voltage Monitor 1 Reset

7.3 Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset

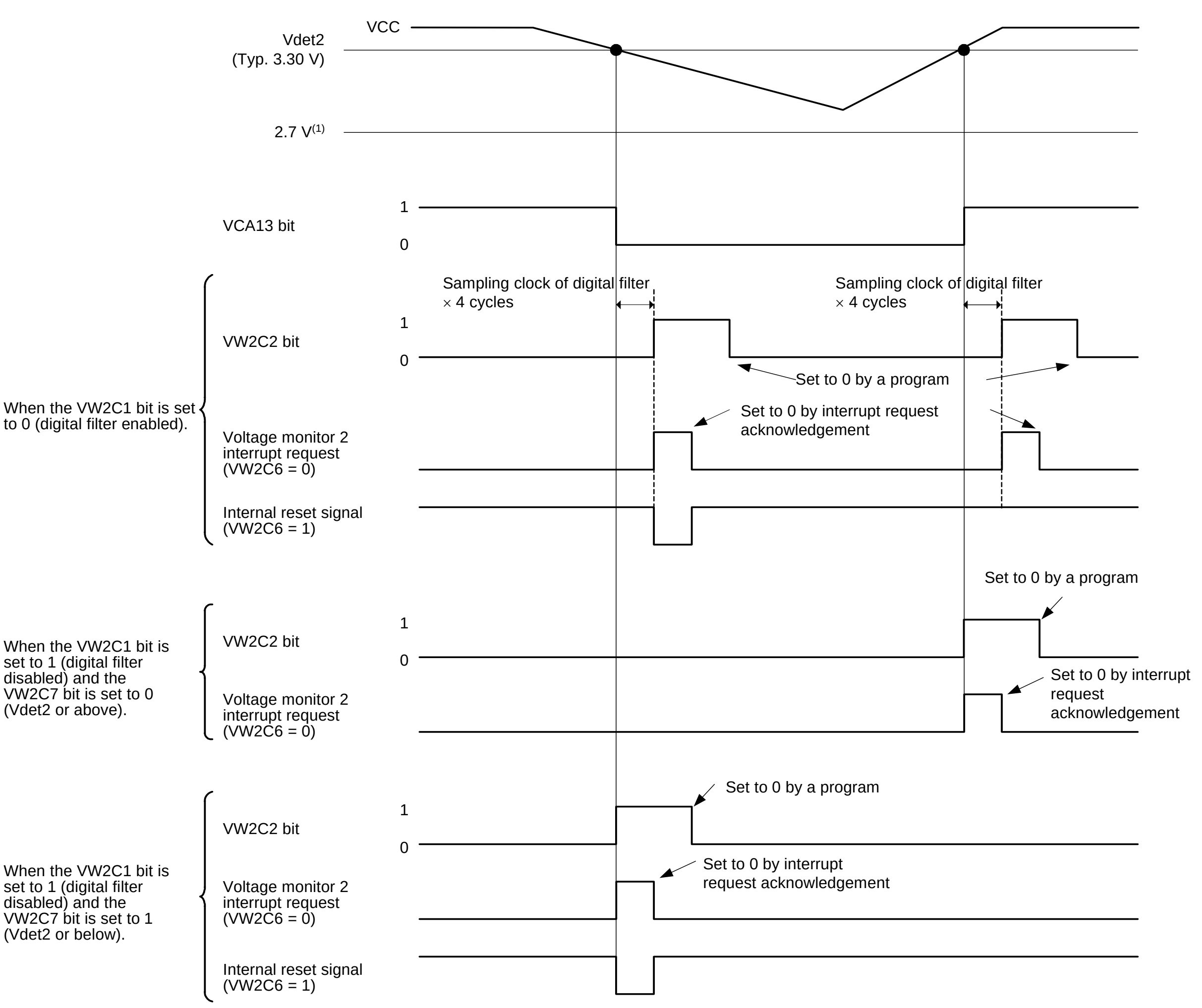
Table 7.3 lists the Setting Procedure of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset Associated Bits. Figure 7.9 shows an Operating Example of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset. To use voltage monitor 2 interrupt or voltage monitor 2 reset to exit stop mode, set the VW2C1 bit in the VW2C register to 1 (digital filter disabled).

Table 7.3 Setting Procedure of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset Associated Bits

Step	When Using Digital Filter		When Not Using Digital Filter	
	Voltage Monitor 2 Interrupt	Voltage Monitor 2 Reset	Voltage Monitor 2 Interrupt	Voltage Monitor 2 Reset
1	Set the VCA27 bit in the VCA2 register to 1 (voltage detection 2 circuit enabled).			
2	Wait for td(E-A)			
3[2]	Select the sampling clock of the digital filter by bits VW2F0 to VW2F1 in the VW2C register.		Select the timing of the interrupt and reset request by the VW2C7 bit in the VW2C register[1].	
4[2]	Set the VW2C1 bit in the VW2C register to 0 (digital filter enabled).		Set the VW2C1 bit in the VW2C register to 1 (digital filter disabled).	
5[2]	Set the VW2C6 bit in the VW2C register to 0 (voltage monitor 2 interrupt mode).	Set the VW2C6 bit in the VW2C register to 1 (voltage monitor 2 reset mode).	Set the VW2C6 bit in the VW2C register to 0 (voltage monitor 2 interrupt mode).	Set the VW2C6 bit in the VW2C register to 1 (voltage monitor 2 reset mode).
6	Set the VW2C2 bit in the VW2C register to 0 (passing of Vdet2 is not detected).			
7	Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on).		–	
8	Wait for 4 cycles of the sampling clock of the digital filter.		– (No wait time)	
9	Set the VW2C0 bit in the VW2C register to 1 (voltage monitor 2 interrupt/reset enabled).			

NOTES:

1. Set the VW2C7 bit to 1 (when VCC reaches Vdet2 or below) for the voltage monitor 2 reset.
2. When the VW2C0 bit is set to 0 (disabled), steps 3, 4 and 5 can be executed simultaneously (with 1 instruction).


VCC
Vdet2
(Typ. 3.30 V)
2.7 V(1)
VCA13 bit
1
0
Sampling clock of digital filter
× 4 cycles
Sampling clock of digital filter
× 4 cycles
When the VW2C1 bit is set to 0 (digital filter enabled).
VW2C2 bit
Set to 0 by a program
Set to 0 by interrupt request acknowledgement
Voltage monitor 2 interrupt request (VW2C6 = 0)
Internal reset signal (VW2C6 = 1)
When the VW2C1 bit is set to 1 (digital filter disabled) and the VW2C7 bit is set to 0 (Vdet2 or above).
Set to 0 by a program
Set to 0 by interrupt request acknowledgement
When the VW2C1 bit is set to 1 (digital filter disabled) and the VW2C7 bit is set to 1 (Vdet2 or below).
Set to 0 by a program
Set to 0 by interrupt request acknowledgement

VCA13: Bit in VCA1 register
VW2C1, VW2C2, VW2C6, VW2C7: Bit in VW2C register

The above applies under the following conditions.
- VCA27 bit in VCA2 register = 1 (voltage detection 2 circuit enabled)
- VW2C0 bit in VW2C register = 1 (voltage monitor 2 interrupt and voltage monitor 2 reset enabled)

NOTE:
1. If voltage monitor 1 reset is not used, set the power supply to VCC ≥ 2.7.

Figure 7.9 Operating Example of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset

8. Processor Mode

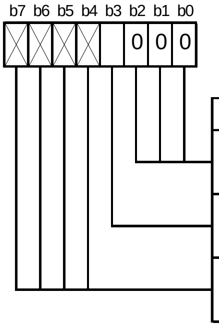
8.1 Processor Modes

Single-chip mode can be selected as the processor mode. Table 8.1 lists Features of Processor Mode. Figure 8.1 shows the PM0 Register and Figure 8.2 shows the PM1 Register.

Table 8.1 Features of Processor Mode

Processor Mode	Accessible Areas	Pins Assignable as I/O Port Pins
Single-chip mode	SFR, internal RAM, internal ROM	All pins are I/O ports or peripheral function I/O pins.

Processor Mode Register 0[1]



Symbol	Address	After Reset
PM0	0004h	00h

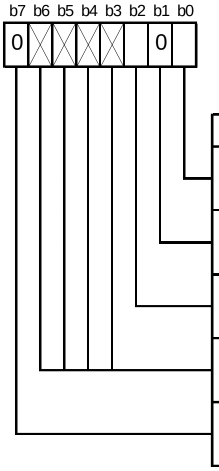
Bit Symbol	Bit Name	Function	RW
— (b2-b0)	Reserved bits	Set to 0.	RW
PM03	Software reset bit	The MCU is reset when this bit is set to 1. When read, the content is 0.	RW
— (b7-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		—

NOTE:

1. Set the PRC1 bit in the PRCR register to 1 (write enable) before rewriting the PM0 register.

Figure 8.1 PM0 Register

Processor Mode Register 1[1]



Symbol	Address	After Reset
PM1	0005h	00h

Bit Symbol	Bit Name	Function	RW
— (b0)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		—
— (b1)	Reserved bit	Set to 0.	RW
PM12	WDT interrupt/reset switch bit	0 : Watchdog timer interrupt 1 : Watchdog timer reset[2]	RW
— (b6-b3)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		—
— (b7)	Reserved bit	Set to 0.	RW

NOTES:

1. Set the PRC1 bit in the PRCR register to 1 (write enable) before rewriting the PM1 register.
2. The PM12 bit is set to 1 by a program (and remains unchanged even if 0 is written to it). When the CSPRO bit in the CSPR register is set to 1 (count source protect mode enabled), the PM12 bit is automatically set to 1.

Figure 8.2 PM1 Register

9. Bus

The bus cycles differ when accessing ROM/RAM, and when accessing SFR. Table 9.1 lists Bus Cycles by Access Space of the R8C/18 Group and Table 9.2 lists Bus Cycles by Access Space of the R8C/19 Group. ROM/RAM and SFR are connected to the CPU by an 8-bit bus. When accessing in word (16-bit) units, these areas are accessed twice in 8-bit units. Table 9.3 lists Access Units and Bus Operations.

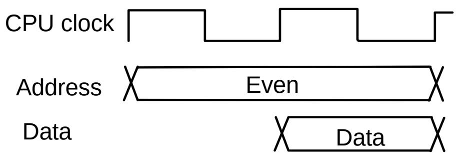
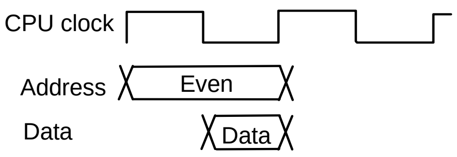
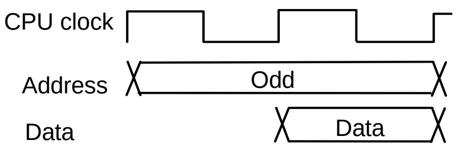
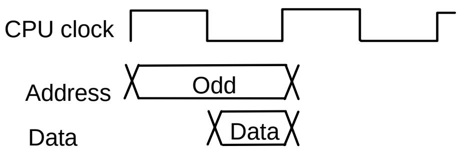
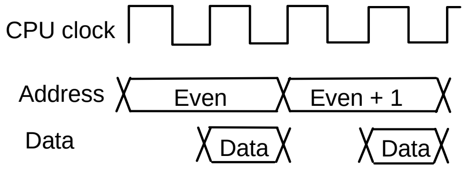
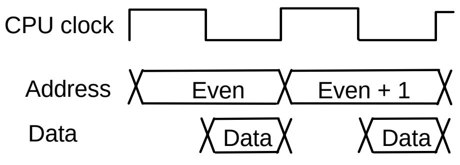
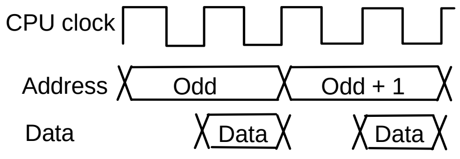
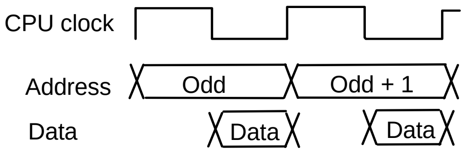
Table 9.1 Bus Cycles by Access Space of the R8C/18 Group

Access Area	Bus Cycle
SFR	2 cycles of CPU clock
ROM/RAM	1 cycle of CPU clock

Table 9.2 Bus Cycles by Access Space of the R8C/19 Group

Access Area	Bus Cycle
SFR/data flash	2 cycles of CPU clock
Program ROM/RAM	1 cycle of CPU clock

Table 9.3 Access Units and Bus Operations

Area	SFR, data flash	ROM (program ROM), RAM
Even address Byte access		
Odd address Byte access		
Even address Word access	 CPU clock Address Even Even + 1 Data Data Data	 CPU clock Address Even Even + 1 Data Data Data
Odd address Word access	 CPU clock Address Odd Odd + 1 Data Data Data	 CPU clock Address Odd Odd + 1 Data Data Data

10. Clock Generation Circuit

The clock generation circuit has:

- Main clock oscillation circuit
- On-chip oscillator (oscillation stop detection function)

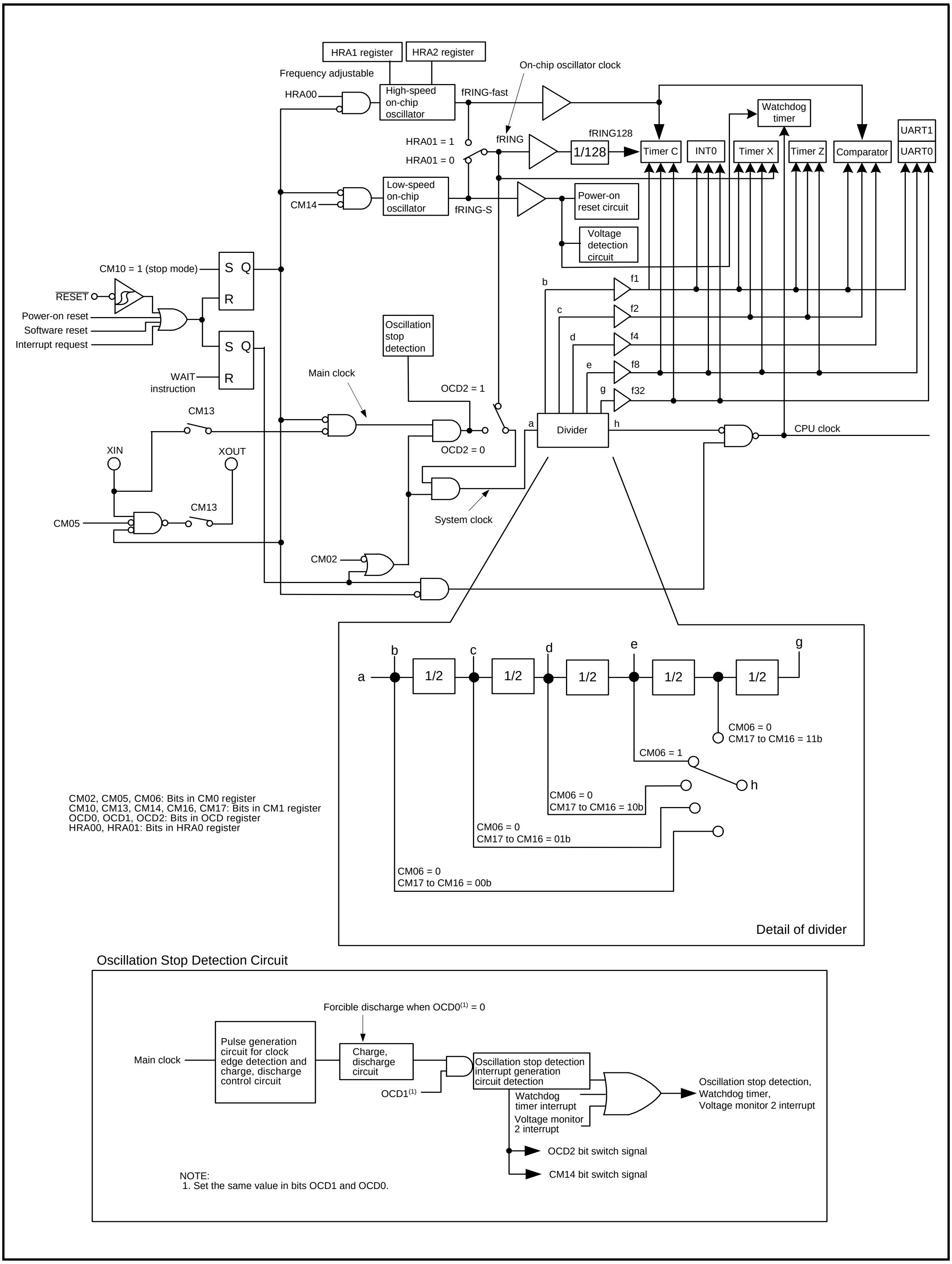
Table 10.1 lists Specifications of Clock Generation Circuit. Figure 10.1 shows a Clock Generation Circuit. Figures 9.2 to 10.5 show clock associated registers.

Table 10.1 Specifications of Clock Generation Circuit

Item	Main Clock Oscillation Circuit	On-Chip Oscillator	
		High-Speed On-Chip Oscillator	Low-Speed On-Chip Oscillator
Applications	• CPU clock source • Peripheral function clock source	• CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when main clock stops oscillating	• CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when main clock stops oscillating
Clock frequency	0 to 20 MHz	Approx. 8 MHz	Approx. 125 kHz
Connectable oscillator	• Ceramic resonator • Crystal oscillator	–	–
Oscillator connect pins	XIN, XOUT(1)	(Note 1)	(Note 1)
Oscillation stop, restart function	Usable	Usable	Usable
Oscillator status after reset	Stop	Stop	Oscillate
Others	Externally generated clock can be input	–	–

NOTE:

1. These pins can be used as P4_6 or P4_7 when using the on-chip oscillator clock as the CPU clock while the main clock oscillation circuit is not used.

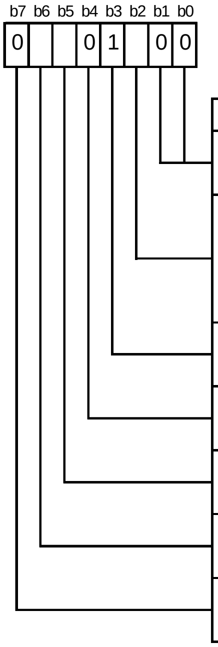


CM02, CM05, CM06: Bits in CM0 register
CM10, CM13, CM14, CM16, CM17: Bits in CM1 register
OCD0, OCD1, OCD2: Bits in OCD register
HRA00, HRA01: Bits in HRA0 register

NOTE:
1. Set the same value in bits OCD1 and OCD0.

Figure 10.1 Clock Generation Circuit

System Clock Control Register 0[1]



Symbol	Address	After Reset
CM0	0006h	68h

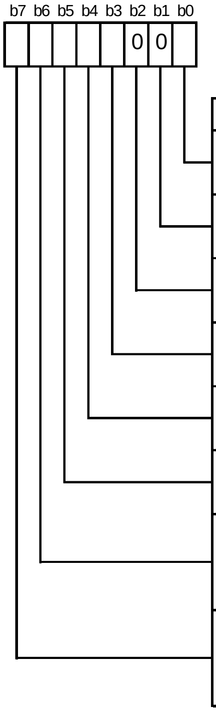
Bit Symbol	Bit Name	Function	RW
— (b1-b0)	Reserved bits	Set to 0.	RW
CM02	WAIT peripheral function clock stop bit	0 : Peripheral function clock does not stop in w ait mode. 1 : Peripheral function clock stops in w ait mode.	RW
— (b3)	Reserved bit	Set to 1.	RW
— (b4)	Reserved bit	Set to 0.	RW
CM05	Main clock (XIN-XOUT) stop bit[2, 4]	0 : Main clock oscillates. 1 : Main clock stops.[3]	RW
CM06	System clock division select bit 0[5]	0 : CM16, CM17 enabled 1 : Divide-by-8 mode	RW
— (b7)	Reserved bit	Set to 0.	RW

NOTES:

1. Set the PRC0 bit in the PRCR register to 1 (w rite enable) before rew riting the CM0 register.
2. The CM05 bit stops the main clock w hen the on-chip oscillator mode is selected.
   Do not use this bit to detect w hether the main clock is stopped. To stop the main clock, set the bits in the follow ing order:
   (a) Set bits OCD1 and OCD0 in the OCD register to 00b (oscillation stop detection function disabled).
   (b) Set the OCD2 bit to 1 (selects on-chip oscillator clock).
3. To input an external clock, set the CM05 bit to 1 (main clock stops) and the CM13 bit in the CM1 register to 1 (XIN-XOUT pin).
4. When the CM05 bit is set to 1 (main clock stops), P4_6 and P4_7 can be used as input ports.
5. When entering stop mode from high or medium speed mode, the CM06 bit is set to 1 (divide-by-8 mode).

Figure 10.2 CM0 Register

System Clock Control Register 1[1]



Symbol	Address	After Reset
CM1	0007h	20h

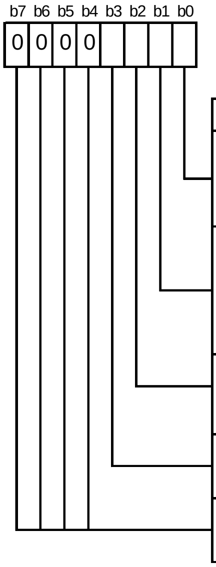
Bit Symbol	Bit Name	Function	RW
CM10	All clock stop control bit[4, 7, 8]	0 : Clock operates. 1 : Stops all clocks (stop mode).	RW
— (b1)	Reserved bit	Set to 0.	RW
— (b2)	Reserved bit	Set to 0.	RW
CM13	Port XIN-XOUT sw itch bit[7]	0 : Input port P4_6, P4_7 1 : XIN-XOUT Pin	RW
CM14	Low -speed on-chip oscillation stop bit[5, 6, 8]	0 : Low -speed on-chip oscillator on 1 : Low -speed on-chip oscillator off	RW
CM15	XIN-XOUT drive capacity select bit[2]	0 : Low 1 : High	RW
CM16	System clock division select bits 1[3]	b7 b6 0 0 : No division mode 0 1 : Divide-by-2 mode 1 0 : Divide-by-4 mode 1 1 : Divide-by-16 mode	RW
CM17			RW

NOTES:

1. Set the PRC0 bit in the PRCR register to 1 (w rite enable) before rew riting the CM1 register.
2. When entering stop mode from high or medium speed mode, this bit is set to 1 (drive capacity high).
3. When the CM06 bit is set to 0 (bits CM16, CM17 enabled), bits CM16 to CM17 are enabled.
4. If the CM10 bit is set to 1 (stop mode), the on-chip feedback resistor is disabled.
5. When the OCD2 bit is set to 0 (main clock selected), the CM14 bit is set to 1 (low -speed on-chip oscillator stopped). When the OCD2 bit is set to 1 (on-chip oscillator clock selected), the CM14 bit is set to 0 (low -speed on-chip oscillator on). And remains unchanged even if 1 is w ritten to it.
6. When using the voltage detection interrupt, set the CM14 bit to 0 (low -speed on-chip oscillator on).
7. When the CM10 bit is set to 1 (stop mode), or the CM05 bit in the CM0 register to 1 (main clock stops) and the CM13 bit is set to 1 (XIN-XOUT pin), the XOUT (P4_7) pin becomes "H".
   When the CM13 bit is set to 0 (input ports, P4_6, P4_7), P4_7 (XOUT) enters input mode.
8. In count source protect mode (refer to **13.2 Count Source Protection Mode Enabled**), the value remains unchanged even if bits CM10 and CM14 are set.

Figure 10.3 CM1 Register

Oscillation Stop Detection Register[1]



Symbol	Address	After Reset
OCD	000Ch	04h

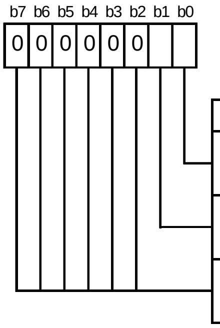
Bit Symbol	Bit Name	Function	RW
OCD0	Oscillation stop detection enable bits	b1 b0 0 0 : Oscillation stop detection function disabled 0 1 : Do not set. 1 0 : Do not set. 1 1 : Oscillation stop detection function enabled(4, 7)	RW
OCD1			RW
OCD2	System clock select bit(6)	0 : Selects main clock.(7) 1 : Selects on-chip oscillator clock.(2)	RW
OCD3	Clock monitor bit(3, 5)	0 : Main clock oscillates. 1 : Main clock stops.	RO
— (b7-b4)	Reserved bits	Set to 0.	RW

NOTES:

1. Set the PRC0 bit in the PRCR register to 1 (w rite enable) before rew riting to this register.
2. The OCD2 bit is automatically set to 1 (on-chip oscillator clock selected) if a main clock oscillation stop is detected w hile bits OCD1 to OCD0 are set to 11b (oscillation stop detection function enabled). If the OCD3 bit is set to 1 (main clock stops), the OCD2 bit remains unchanged even w hen set to 0 (main clock selected).
3. The OCD3 bit is enabled w hen bits OCD1 to OCD0 are set to 11b (oscillation stop detection function enabled).
4. Set bits OCD1 to OCD0 to 00b (oscillation stop detection function disabled) before entering stop or on-chip oscillator mode (main clock stops).
5. The OCD3 bit remains 0 (main clock oscillates) if bits OCD1 to OCD0 are set to 00b.
6. The CM14 bit is set to 0 (low -speed on-chip oscillator on) if the OCD2 bit is set to 1 (on-chip oscillator clock selected).
7. Refer to **Figure 10.9 Procedure for Sw itching Clock Source from Low-Speed On-Chip Oscillator to Main Clock** for the sw itching procedure w hen the main clock re-oscillates after detecting an oscillation stop.

Figure 10.4 OCD Register

High-Speed On-Chip Oscillator Control Register 0[1]



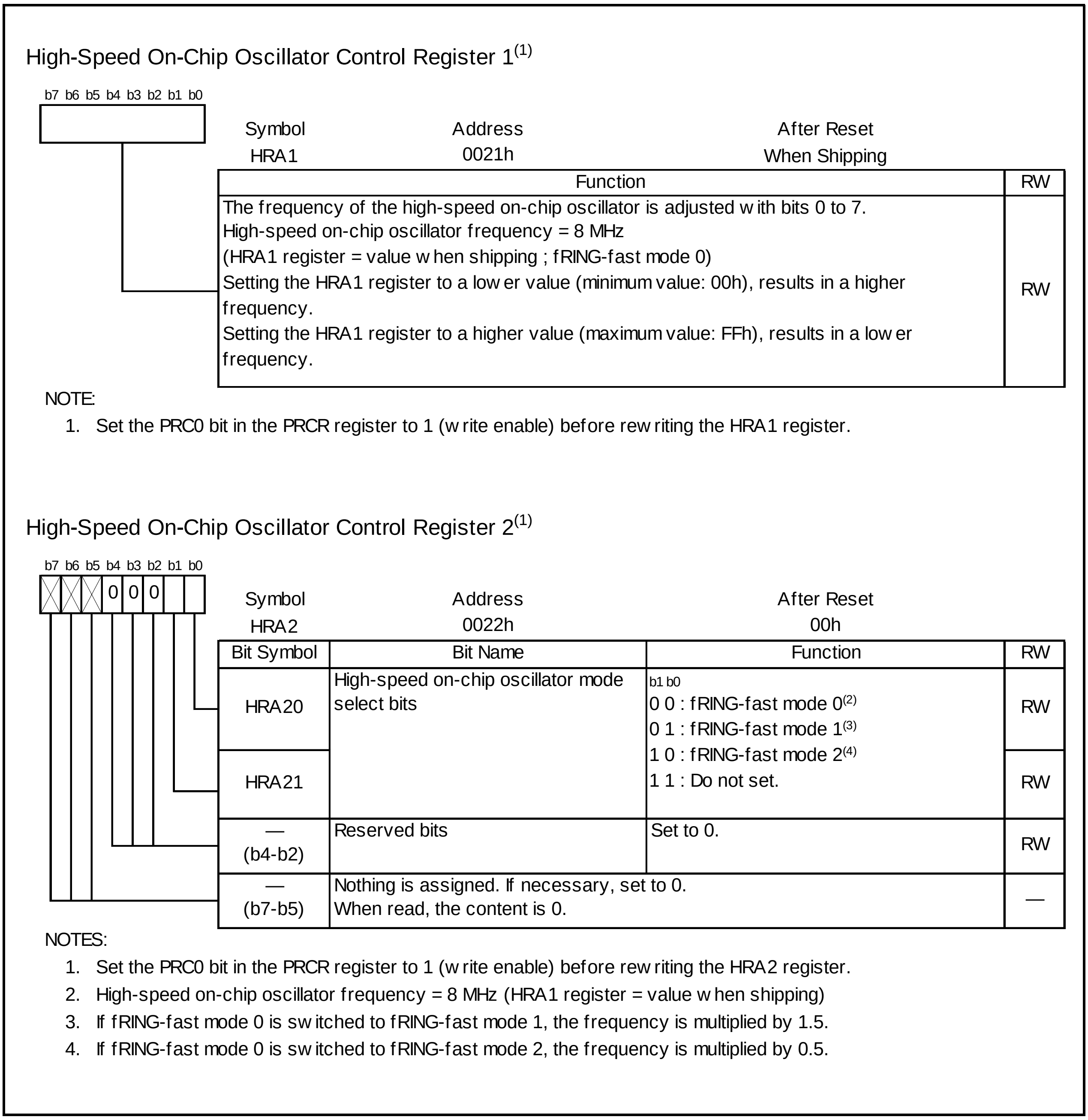
Symbol	Address	After Reset
HRA0	0020h	00h

Bit Symbol	Bit Name	Function	RW
HRA00	High-speed on-chip oscillator enable bit	0 : High-speed on-chip oscillator off 1 : High-speed on-chip oscillator on	RW
HRA01	High-speed on-chip oscillator select bit[2]	0 : Selects low-speed on-chip oscillator.[3] 1 : Selects high-speed on-chip oscillator.	RW
— (b7-b2)	Reserved bits	Set to 0.	RW

NOTES:

1. Set the PRC0 bit in the PRCR register to 1 (write enable) before rewriting the HRA0 register.
2. Change the HRA01 bit under the following conditions.
   • HRA00 = 1 (high-speed on-chip oscillation)
   • The CM14 bit in the CM1 register = 0 (low-speed on-chip oscillator on)
3. When setting the HRA01 bit to 0 (low-speed on-chip oscillator selected), do not set the HRA00 bit to 0 (high-speed on-chip oscillator off) at the same time.
   Set the HRA00 bit to 0 after setting the HRA01 bit to 0.

Figure 10.5 HRA0 Register



High-Speed On-Chip Oscillator Control Register 1[1]

Symbol	Address	After Reset
HRA1	0021h	When Shipping

Function	RW
The frequency of the high-speed on-chip oscillator is adjusted w ith bits 0 to 7. High-speed on-chip oscillator frequency = 8 MHz (HRA1 register = value w hen shipping ; fRING-fast mode 0) Setting the HRA1 register to a low er value (minimum value: 00h), results in a higher frequency. Setting the HRA1 register to a higher value (maximum value: FFh), results in a low er frequency.	RW

NOTE:

1. Set the PRC0 bit in the PRCR register to 1 (w rite enable) before rew riting the HRA1 register.

High-Speed On-Chip Oscillator Control Register 2[1]

Symbol	Address	After Reset
HRA2	0022h	00h

Bit Symbol	Bit Name	Function	RW
HRA20	High-speed on-chip oscillator mode select bits	b1 b0 0 0 : fRING-fast mode 0[2] 0 1 : fRING-fast mode 1[3] 1 0 : fRING-fast mode 2[4] 1 1 : Do not set.	RW
HRA21			RW
— (b4-b2)	Reserved bits	Set to 0.	RW
— (b7-b5)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		—

NOTES:

1. Set the PRC0 bit in the PRCR register to 1 (w rite enable) before rew riting the HRA2 register.
2. High-speed on-chip oscillator frequency = 8 MHz (HRA1 register = value w hen shipping)
3. If fRING-fast mode 0 is sw itched to fRING-fast mode 1, the frequency is multiplied by 1.5.
4. If fRING-fast mode 0 is sw itched to fRING-fast mode 2, the frequency is multiplied by 0.5.

Figure 10.6 Registers HRA1 and HRA2

The clocks generated by the clock generation circuits are described below.

10.1 Main Clock

This clock is supplied by a main clock oscillation circuit. This clock is used as the clock source for the CPU and peripheral function clocks. The main clock oscillation circuit is configured by connecting resonator between the XIN and XOUT pins. The main clock oscillation circuit includes an on-chip feedback resistor, which is disconnected from the oscillation circuit in stop mode in order to reduce the amount of power consumed by the chip. The main clock oscillation circuit may also be configured by feeding an externally generated clock to the XIN pin. Figure 10.7 shows Examples of Main Clock Connection Circuit. During reset and after reset, the main clock stops.

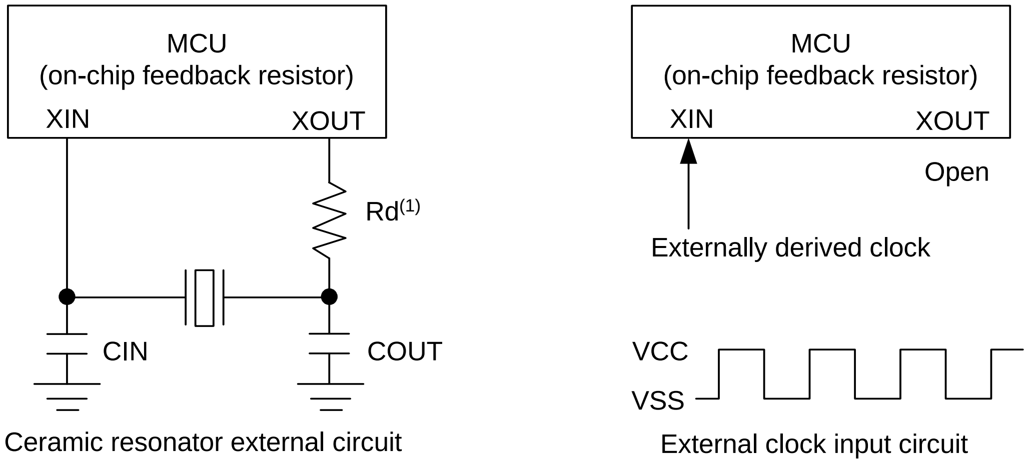
The main clock starts oscillating when the CM05 bit in the CM0 register is set to 0 (main clock on) after setting the CM13 bit in the CM1 register to 1 (XIN- XOUT pin).

To use the main clock for the CPU clock source, set the OCD2 bit in the OCD register to 0 (selects main clock) after the main clock is oscillating stably.

The power consumption can be reduced by setting the CM05 bit in the CM0 register to 1 (main clock stops) if the OCD2 bit is set to 1 (select on-chip oscillator clock).

When an external clock is input to the XIN pin, the main clock does not stop if the CM05 bit is set to 1. If necessary, use an external circuit to stop the clock.

In stop mode, all clocks including the main clock stop. Refer to **10.4 Power Control** for details.


MCU
(on-chip feedback resistor)
XIN
XOUT
Rd(1)
CIN
COUT
Ceramic resonator external circuit
MCU
(on-chip feedback resistor)
XIN
XOUT
Open
Externally derived clock
VCC
VSS
External clock input circuit

NOTE:
1. Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the manufacturer of the oscillator. When the oscillation drive capacity is set to low, check that oscillation is stable. Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added to the chip externally, insert a feedback resistor between XIN and XOUT following the instructions.

Figure 10.7 Examples of Main Clock Connection Circuit

10.2 On-Chip Oscillator Clocks

These clocks are supplied by the on-chip oscillators (high-speed on-chip oscillator and a low-speed on-chip oscillator). The on-chip oscillator clock is selected by the HRA01 bit in the HRA0 register.

10.2.1 Low-Speed On-Chip Oscillator Clock

The clock generated by the low-speed on-chip oscillator is used as the clock source for the CPU clock, peripheral function clock, fRING, fRING128, and fRING-S.

After reset, the on-chip oscillator clock generated by the low-speed on-chip oscillator divided by 8 is selected as the CPU clock.

If the main clock stops oscillating when bits OCD1 to OCD0 in the OCD register are set to 11b (oscillation stop detection function enabled), the low-speed on-chip oscillator automatically starts operating, supplying the necessary clock for the MCU.

The frequency of the low-speed on-chip oscillator varies depending on the supply voltage and the operating ambient temperature. Application products must be designed with sufficient margin to allow for the frequency changes.

10.2.2 High-Speed On-Chip Oscillator Clock

The clock generated by the high-speed on-chip oscillator is used as the clock source for the CPU clock, peripheral function clock, fRING, fRING128, and fRING1-fast.

After reset, the on-chip oscillator clock generated by the high-speed on-chip oscillator stops. Oscillation is started by setting the HRA00 bit in the HRA0 register to 1 (high-speed on-chip oscillator on). The frequency can be adjusted by registers HRA1 and HRA2.

Since there are differences in delay among the bits in the HRA1 register, make adjustments by changing the settings of individual bits.

The high-speed on-chip oscillator frequency may be changed in flash memory CPU rewrite mode during auto-program operation or auto-erase operation. Refer to **10.6.4 High-Speed On-Chip Oscillator Clock** for details.

10.3 CPU Clock and Peripheral Function Clock

There are a CPU clock to operate the CPU and a peripheral function clock to operate the peripheral functions. Refer to **Figure 10.1 Clock Generation Circuit**.

10.3.1 System Clock

The system clock is the clock source for the CPU and peripheral function clocks. Either the main clock or the on-chip oscillator clock can be selected.

10.3.2 CPU Clock

The CPU clock is an operating clock for the CPU and watchdog timer.
The system clock can be divided by 1 (no division), 2, 4, 8, or 16 to produce the CPU clock. Use the CM06 bit in the CM0 register and bits CM16 to CM17 in the CM1 register to select the value of the division.
After reset, the low-speed on-chip oscillator clock divided by 8 provides the CPU clock. When entering stop mode from high-speed or medium-speed mode, the CM06 bit is set to 1 (Divide-by-8 mode).

10.3.3 Peripheral Function Clock (f1, f2, f4, f8, and f32)

The peripheral function clock is the operating clock for the peripheral functions.
The clock fi (i = 1, 2, 4, 8, and 32) is generated by the system clock divided by i. The clock fi is used for timers X, Y, Z, and C, the serial interface and the comparator.
When the WAIT instruction is executed after setting the CM02 bit in the CM0 register to 1 (peripheral function clock stops in wait mode), the clock fi stops.

10.3.4 fRING and fRING128

fRING and fRING128 are operating clocks for the peripheral functions.
fRING runs at the same frequency as the on-chip oscillator clock and can be used as the source for the timer X. fRING128 is generated from fRING by dividing it by 128, and it can be used as timer C.
When the WAIT instruction is executed, the clocks fRING and fRING128 do not stop.

10.3.5 fRING-fast

fRING-fast is used as the count source for timer C. fRING-fast is generated by the high-speed on-chip oscillator and supplied by setting the HRA00 bit to 1.
When the WAIT instruction is executed, the clock fRING-fast does not stop.

10.3.6 fRING-S

fRING-S is an operating clock for the watchdog timer and voltage detection circuit. fRING-S is supplied by setting the CM14 bit to 0 (low-speed on-chip oscillator on) and uses the clock generated by the low-speed on-chip oscillator. When the WAIT instruction is executed or in count source protect mode of the watchdog timer, fRING-S does not stop.

10.4 Power Control

There are three power control modes. All modes other than wait mode and stop mode are referred to as standard operating mode.

10.4.1 Standard Operating Mode

Standard operating mode is further separated into four modes.
In standard operating mode, the CPU clock and the peripheral function clock are supplied to operate the CPU and the peripheral function clocks. Power consumption control is enabled by controlling the CPU clock frequency. The higher the CPU clock frequency, the more processing power increases. The lower the CPU clock frequency, the more power consumption decreases. When unnecessary oscillator circuits stop, power consumption is further reduced.
Before the clock sources for the CPU clock can be switched over, the new clock source needs to be oscillating and stable. If the new clock source is the main clock, allow sufficient wait time in a program until oscillation is stabilized before exiting.

Table 10.2 Settings and Modes of Clock Associated Bits

Modes		OCD Register	CM1 Register		CM0 Register	
		OCD2	CM17, CM16	CM13	CM06	CM05
High-speed mode		0	00b	1	0	0
Medium-speed mode	Divide-by-2	0	01b	1	0	0
	Divide-by-4	0	10b	1	0	0
	Divide-by-8	0	–	1	1	0
	Divide-by-16	0	11b	1	0	0
High-speed, low-speed on-chip oscillator mode[1]	No division	1	00b	–	0	–
	Divide-by-2	1	01b	–	0	–
	Divide-by-4	1	10b	–	0	–
	Divide-by-8	1	–	–	1	–
	Divide-by-16	1	11b	–	0	–

NOTE:

1. The low-speed on-chip oscillator is used as the on-chip oscillator clock when the CM14 bit in the CM1 register is set to 0 (low-speed on-chip oscillator on) and the HRA01 bit in the HRA0 register is set to 0. The high-speed on-chip oscillator is used as the on-chip oscillator clock when the HRA00 bit in the HRA0 register is set to 1 (high-speed on-chip oscillator A on) and the HRA01 bit in the HRA0 register is set to 1.

10.4.1.1 High-Speed Mode

The main clock divided by 1 (no division) provides the CPU clock. If the CM14 bit is set to 0 (low-speed on-chip oscillator on) or the HRA00 bit in the HRA0 register is set to 1 (high-speed on-chip oscillator on), fRING and fRING128 can be used as timers X and C. When the HRA00 bit is set to 1, fRING-fast can be used as timer C. When the CM14 bit is set to 0 (low-speed on-chip oscillator on), fRING-S can be used for the watchdog timer and voltage detection circuit.

10.4.1.2 Medium-Speed Mode

The main clock divided by 2, 4, 8, or 16 provides the CPU clock. If the CM14 bit is set to 0 (low-speed on-chip oscillator on) or the HRA00 bit in the HRA0 register is set to 1 (high-speed on-chip oscillator on), fRING and fRING128 can be used as timers X and C. When the HRA00 bit is set to 1, fRING-fast can be used as timer C. When the CM14 bit is set to 0 (low-speed on-chip oscillator on), fRING-S can be used for the watchdog timer and voltage detection circuit.

10.4.1.3 High-Speed and Low-Speed On-Chip Oscillator Modes

The on-chip oscillator clock divided by 1 (no division), 2, 4, 8, or 16 provides the CPU clock. The on-chip oscillator clock is also the clock source for the peripheral function clocks. When the HRA00 bit is set to 1, fRING-fast can be used as timer C. When the CM14 bit is set to 0 (low-speed on-chip oscillator on), fRING-S can be used for the watchdog timer and voltage detection circuit.

10.4.2 Wait Mode

Since the CPU clock stops in wait mode, the CPU, which operates using the CPU clock and the watchdog timer when count source protection mode is disabled stop. The main clock and on-chip oscillator clock do not stop and the peripheral functions using these clocks continue operating.

10.4.2.1 Peripheral Function Clock Stop Function

If the CM02 bit is set to 1 (peripheral function clock stops in wait mode), the f1, f2, f4, f8, and f32 clocks stop in wait mode. This reduces power consumption.

10.4.2.2 Entering Wait Mode

The MCU enters wait mode when the WAIT instruction is executed.

10.4.2.3 Pin Status in Wait Mode

The status before wait mode was entered is maintained.

10.4.2.4 Exiting Wait Mode

The MCU exits wait mode by a hardware reset or a peripheral function interrupt. To use a hardware reset to exit wait mode, set bits ILVL2 to ILVL0 for the peripheral function interrupts to 000b (interrupts disabled) before executing the WAIT instruction.
The peripheral function interrupts are affected by the CM02 bit. When the CM02 bit is set to 0 (peripheral function clock does not stop in wait mode), all peripheral function interrupts can be used to exit wait mode. When the CM02 bit is set to 1 (peripheral function clock stops in wait mode), the peripheral functions using the peripheral function clock stop operating and the peripheral functions operated by external signals can be used to exit wait mode.
Table 10.3 lists Interrupts to Exit Wait Mode and Usage Conditions.
To use a peripheral function interrupt to exit wait mode, set up the following before executing the WAIT instruction.

(1) Set the interrupt priority level in bits ILVL2 to ILVL0 in the interrupt control registers of the peripheral function interrupts to be used for exiting wait mode. Set bits ILVL2 to ILVL0 of the peripheral function interrupts that are not to be used for exiting wait mode to 000b (interrupt disabled).
(2) Set the I flag to 1.
(3) Operate the peripheral function to be used for exiting wait mode.

When exiting by a peripheral function interrupt, the interrupt sequence is executed when an interrupt request is generated and the CPU clock supply is started.
The CPU clock, when exiting wait mode by a peripheral function interrupt, is the same clock as the CPU clock when the WAIT instruction is executed.

Table 10.3 Interrupts to Exit Wait Mode and Usage Conditions

Interrupt	CM02 = 0	CM02 = 1
Serial interface interrupt	Usable when operating with internal or external clock	Usable when operating with external clock
Key input interrupt	Usable	Usable
Comparator conversion interrupt	Usable in one-shot mode	(Do not use)
Timer X interrupt	Usable in all modes	Usable in event counter mode
Timer Z interrupt	Usable in all modes	(Do not use)
Timer C interrupt	Usable in all modes	(Do not use)
$\overline{\text{INT}}$ interrupt	Usable	Usable ($\overline{\text{INT0}}$ and $\overline{\text{INT3}}$ can be used if there is no filter.)
Voltage monitor 2 interrupt	Usable	Usable
Oscillation stop detection interrupt	Usable	(Do not use)
Watchdog timer interrupt	Usable in count source protect mode	Usable in count source protect mode

10.4.3 Stop Mode

Since the oscillator circuits stop in stop mode, the CPU clock and peripheral function clock stop and the CPU and peripheral functions that use these clocks stop operating. The least power required to operate the MCU is in stop mode. If the voltage applied to the VCC pin is VRAM or more, the contents of internal RAM is maintained.

The peripheral functions clocked by external signals continue operating. Table 10.4 lists Interrupts to Exit Stop Mode and Usage Conditions.

Table 10.4 Interrupts to Exit Stop Mode and Usage Conditions

Interrupt	Usage Conditions
Key input interrupt	–
$\overline{\text{INT0}}$ to $\overline{\text{INT1}}$ interrupts	$\overline{\text{INT0}}$ can be used if there is no filter.
$\overline{\text{INT3}}$ interrupt	No filter. Interrupt request is generated at $\overline{\text{INT3}}$ input (TCC06 bit in TCC0 register is set to 1).
Timer X interrupt	When external pulse is counted in event counter mode.
Serial interface interrupt	When external clock is selected.
Voltage monitor 2 interrupt	Usable in digital filter disabled mode (VW2C1 bit in VW2C register is set to 1)

10.4.3.1 Entering Stop Mode

The MCU enters stop mode when the CM10 bit in the CM1 register is set to 1 (all clocks stop). At the same time, the CM06 bit in the CM0 register is set to 1 (Divide-by-8 mode) and the CM15 bit in the CM10 register is set to 1 (main clock oscillation circuit drive capacity high).

When using stop mode, set bits OCD1 to OCD0 to 00b (oscillation stop detection function disabled) before entering stop mode.

10.4.3.2 Pin Status in Stop Mode

The status before wait mode was entered is maintained.

However, when the CM13 bit in the CM1 register is set to 1 (XIN-XOUT pins), the XOUT(P4_7) pin is held “H”. When the CM13 bit is set to 0 (input ports P4_6 and P4_7), the P4_7(XOUT) pin is held in input status.

10.4.3.3 Exiting Stop Mode

The MCU exits stop mode by a hardware reset or peripheral function interrupt.

When using a hardware reset to exit stop mode, set bits ILVL2 to ILVL0 for the peripheral function interrupts to 000b (interrupts disabled) before setting the CM10 bit to 1.

When using a peripheral function interrupt to exit stop mode, set up the following before setting the CM10 bit to 1.

(1) Set the interrupt priority level in bits ILVL2 to ILVL0 of the peripheral function interrupts to be used for exiting stop mode. Set bits ILVL2 to ILVL0 of the peripheral function interrupts that are not to be used for exiting stop mode to 000b (interrupt disabled).

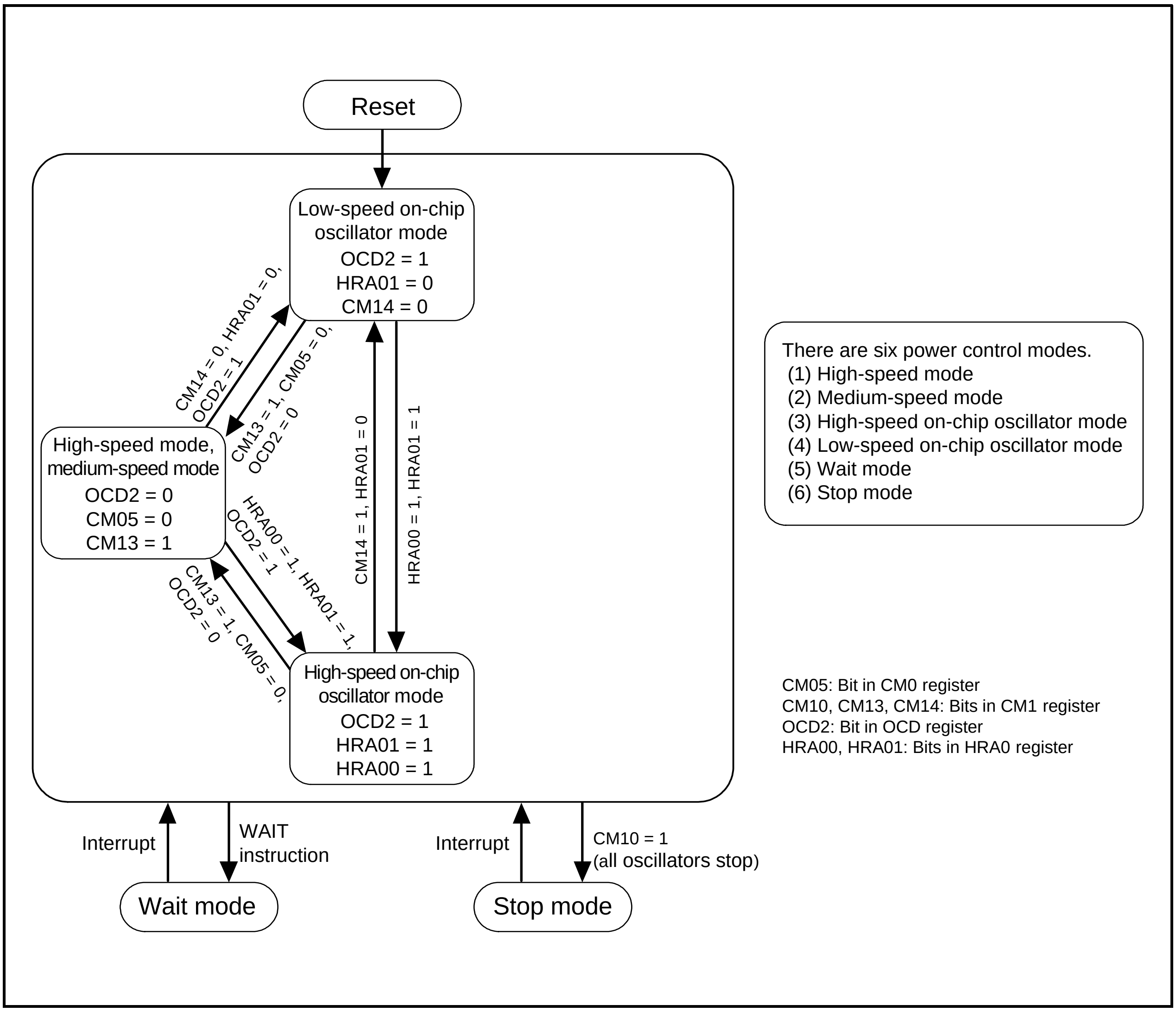
(2) Set the I flag to 1.

(3) Operate the peripheral function to be used for exiting stop mode.

When exiting by a peripheral function interrupt, the interrupt sequence is executed when an interrupt request is generated and the CPU clock supply is started.

The CPU clock, when exiting stop mode by a peripheral function interrupt, is the Divide-by-8 of the clock which was used before stop mode was entered.

Figure 10.8 shows the State Transitions in Power Control.


Reset
Low-speed on-chip oscillator mode
OCD2 = 1
HRA01 = 0
CM14 = 0
High-speed mode, medium-speed mode
OCD2 = 0
CM05 = 0
CM13 = 1
High-speed on-chip oscillator mode
OCD2 = 1
HRA01 = 1
HRA00 = 1
CM14 = 0, HRA01 = 0, OCD2 = 1
CM13 = 1, CM05 = 0, OCD2 = 0
CM14 = 1, HRA01 = 0
HRA00 = 1, HRA01 = 1
HRA00 = 1, HRA01 = 1, OCD2 = 1
CM13 = 1, CM05 = 0, OCD2 = 0
Interrupt
WAIT instruction
Interrupt
CM10 = 1 (all oscillators stop)
Wait mode
Stop mode

There are six power control modes.
(1) High-speed mode
(2) Medium-speed mode
(3) High-speed on-chip oscillator mode
(4) Low-speed on-chip oscillator mode
(5) Wait mode
(6) Stop mode

CM05: Bit in CM0 register
CM10, CM13, CM14: Bits in CM1 register
OCD2: Bit in OCD register
HRA00, HRA01: Bits in HRA0 register

Figure 10.8 State Transitions in Power Control

10.5 Oscillation Stop Detection Function

The oscillation stop detection function detects the stop of the main clock oscillation circuit. The oscillation stop detection function can be enabled and disabled by bits OCD1 to OCD0 in the OCD register.
Table 10.5 lists the Specifications of Oscillation Stop Detection Function.

When the main clock is the CPU clock source and bits OCD1 to OCD0 are set to 11b (oscillation stop detection function enabled), the system is placed in the following state if the main clock stops.

- OCD2 bit in OCD register = 1 (on-chip oscillator clock selected)
- OCD3 bit in OCD register = 1 (main clock stops)
- CM14 bit in CM1 register = 0 (low-speed on-chip oscillator oscillates)
- Oscillation stop detection interrupt request is generated.

Table 10.5 Specifications of Oscillation Stop Detection Function

Item	Specification
Oscillation stop detection enable clock and frequency bandwidth	f(XIN) ≥ 2 MHz
Enabled condition for oscillation stop detection function	Set bits OCD1 to OCD0 to 11b (oscillation stop detection function enabled).
Operation at oscillation stop detection	Oscillation stop detection interrupt is generated

10.5.1 How to Use Oscillation Stop Detection Function

- The oscillation stop detection interrupt shares a vector with the voltage monitor 2 interrupt, and the watchdog timer interrupt. When using the oscillation stop detection interrupt and watchdog timer interrupt, the interrupt source needs to be determined. Table 10.6 lists Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, and Voltage Monitor 2 Interrupts.
- When the main clock restarts after oscillation stop, switch the main clock to the clock source of the CPU clock and peripheral functions by a program.
- Figure 10.9 shows the Procedure for Switching Clock Source from Low-Speed On-Chip Oscillator to Main Clock.
- To enter wait mode while using the oscillation stop detection function, set the CM02 bit to 0 (peripheral function clock does not stop in wait mode).
- Since the oscillation stop detection function is a function for cases where the main clock is stopped by an external cause, set bits OCD1 to OCD0 to 00b (oscillation stop detection function disabled) when the main clock stops or is started by a program, (stop mode is selected or the CM05 bit is changed).
- This function cannot be used when the main clock frequency is 2 MHz or below. In this case, set bits OCD1 to OCD0 to 00b (oscillation stop detection function disabled).
- To use the low-speed on-chip oscillator clock for the CPU clock and clock sources of peripheral functions after detecting the oscillation stop, set the HRA01 bit in the HRA0 register to 0 (low-speed on-chip oscillator selected) and bits OCD1 to OCD0 to 11b (oscillation stop detection function enabled).
  To use the high-speed on-chip oscillator clock for the CPU clock and clock sources of peripheral functions after detecting the oscillation stop, set the HRA01 bit to 1 (high-speed on-chip oscillator selected) and bits OCD1 to OCD0 to 11b (oscillation stop detection function enabled).

Table 10.6 Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, and Voltage Monitor 2 Interrupts

Generated Interrupt Source	Bit Showing Interrupt Cause
Oscillation stop detection ((a) or (b))	(a) OCD3 bit in OCD register = 1 (b) Bits OCD1 to OCD0 in OCD register = 11b and OCD2 bit = 1
Watchdog timer	VW2C3 bit in VW2C register = 1
Voltage monitor 2	VW2C2 bit in VW2C register = 1

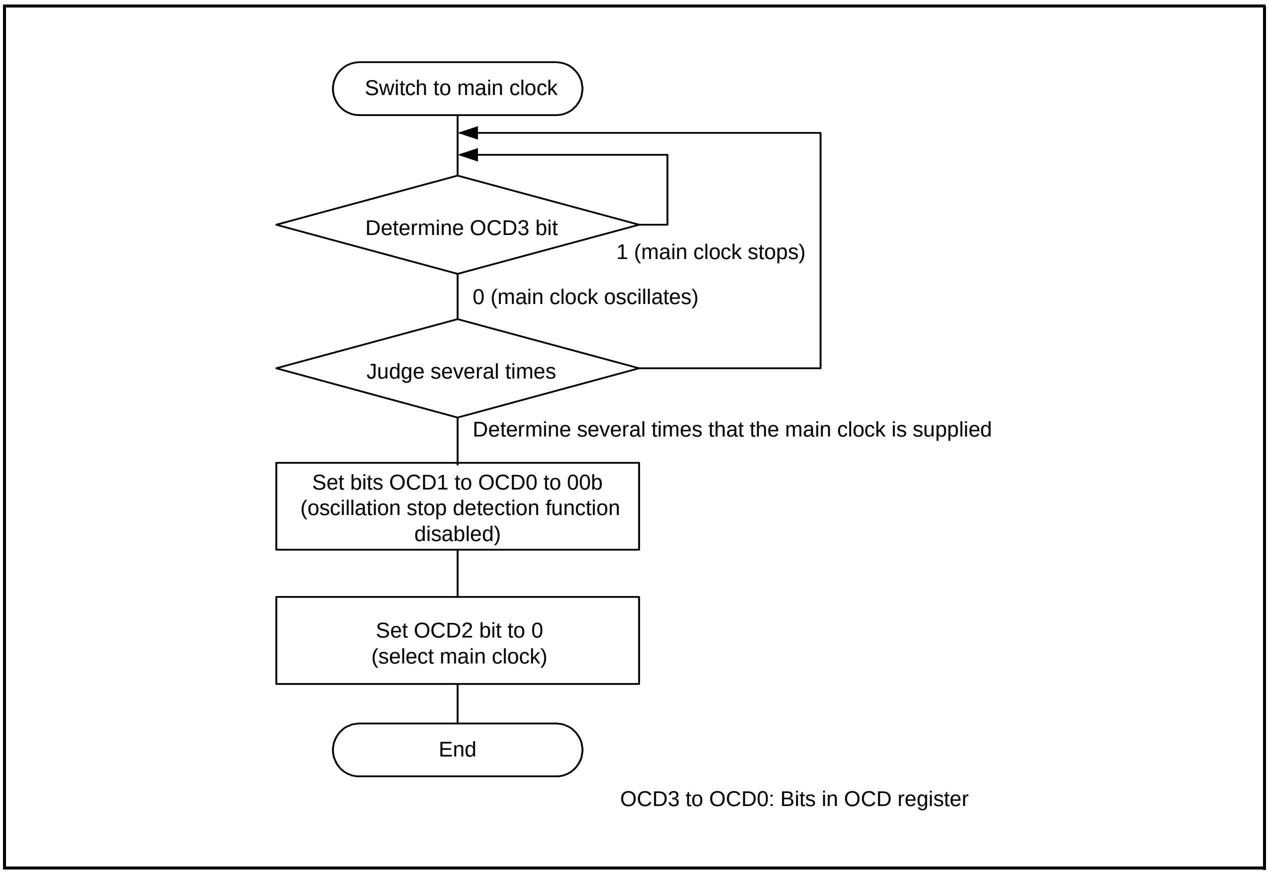

Switch to main clock
Determine OCD3 bit
1 (main clock stops)
0 (main clock oscillates)
Judge several times
Determine several times that the main clock is supplied
Set bits OCD1 to OCD0 to 00b (oscillation stop detection function disabled)
Set OCD2 bit to 0 (select main clock)
End
OCD3 to OCD0: Bits in OCD register

Figure 10.9 Procedure for Switching Clock Source from Low-Speed On-Chip Oscillator to Main Clock

10.6 Notes on Clock Generation Circuit

10.6.1 Stop Mode and Wait Mode

When entering stop mode or wait mode, an instruction queue pre-reads 4 bytes from the WAIT instruction or an instruction that sets the CM10 bit in the CM1 register to 1 (stops all clocks) before the program stops. Therefore, insert at least four NOPs after the WAIT instruction or an instruction that sets the CM10 bit to 1.

10.6.2 Oscillation Stop Detection Function

Since the oscillation stop detection function cannot be used if the main clock frequency is below 2 MHz, set bits OCD1 to OCD0 to 00b (oscillation stop detection function disabled) in this case.

10.6.3 Oscillation Circuit Constants

Ask the manufacturer of the oscillator to specify the best oscillation circuit constants for your system.

10.6.4 High-Speed On-Chip Oscillator Clock

The high-speed on-chip oscillator frequency may be changed up to 10%(1) in flash memory CPU rewrite mode during auto-program operation or auto-erase operation.
The high-speed on-chip oscillator frequency after auto-program operation ends or auto-erase operation ends is held the state before the program command or block erase command is generated. Also, this note is not applicable when the read array command, read status register command, or clear status register command is generated. The application products must be designed with careful considerations for the frequency change.

NOTE:

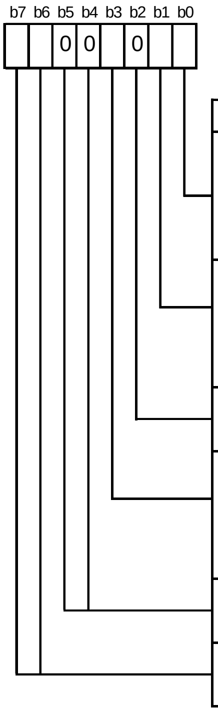
1. Change ratio to 8 MHz frequency adjusted in shipping.

11. Protection

The protection function protects important registers from being easily overwritten when a program runs out of control. Figure 11.1 shows the PRCR Register. The registers protected by the PRCR register are listed below.

- Registers protected by PRC0 bit: Registers CM0, CM1, OCD, HRA0, HRA1, and HRA2
- Registers protected by PRC1 bit: Registers PM0 and PM1
- Registers protected by PRC3 bit: Registers VCA2, VW1C, and VW2C

Protect Register



Symbol	Address	After Reset
PRCR	000Ah	00h

Bit Symbol	Bit Name	Function	RW
PRC0	Protect bit 0	Writing to registers CM0, CM, OCD, HRA0, HRA1, and HRA2 is enabled. 0 : Disables w riting 1 : Enables w riting	RW
PRC1	Protect bit 1	Writing to registers PM0 and PM1 is enabled. 0 : Disables w riting 1 : Enables w riting	RW
— (b2)	Reserved bit	Set to 0.	RW
PRC3	Protect bit 3	Writing to registers VCA2, VW1C, and VW2C is enabled. 0 : Disables w riting 1 : Enables w riting	RW
— (b5-b4)	Reserved bits	Set to 0.	RW
— (b7-b6)	Reserved bits	When read, the content is 0.	RO

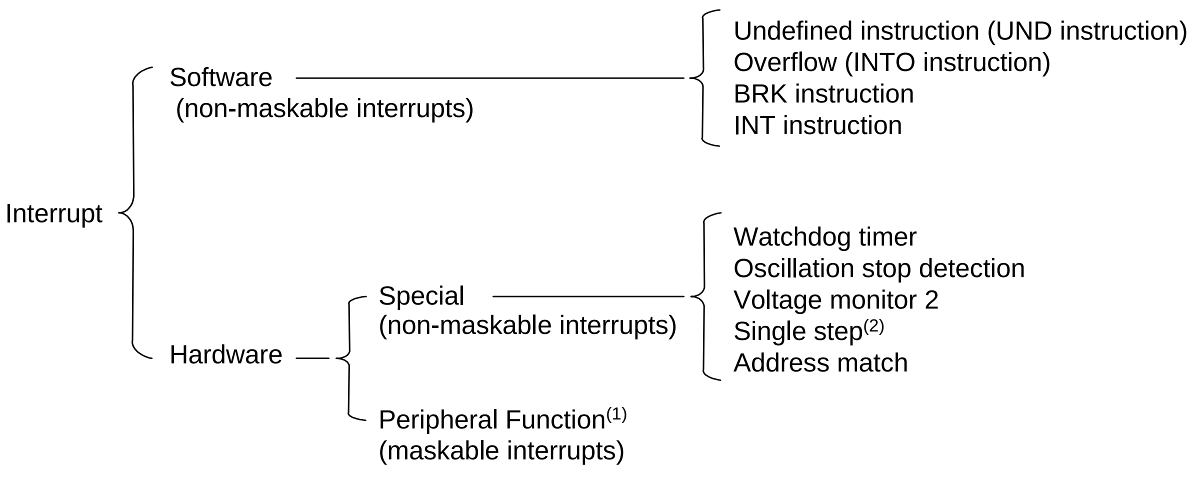
Figure 11.1 PRCR Register

12. Interrupts

12.1 Interrupt Overview

12.1.1 Types of Interrupts

Figure 12.1 shows the types of Interrupts.



NOTES:

1. Peripheral function interrupts in the MCU are used to generate peripheral interrupts.
2. Do not use this interrupt. This is for use with development tools only.

Figure 12.1 Interrupts

- Maskable interrupts: The interrupt enable flag (I flag) enables or disables these interrupts. The interrupt priority order can be changed based on the interrupt priority level.
- Non-maskable interrupts: The interrupt enable flag (I flag) does not enable or disable interrupts. The interrupt priority order cannot be changed based on interrupt priority level.

12.1.2 Software Interrupts

A software interrupt is generated when an instruction is executed. Software interrupts are non-maskable.

12.1.2.1 Undefined Instruction Interrupt

The undefined instruction interrupt is generated when the UND instruction is executed.

12.1.2.2 Overflow Interrupt

The overflow interrupt is generated when the O flag is set to 1 (arithmetic operation overflow) and the INTO instruction is executed. Instructions that set the O flag are: ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, and SUB.

12.1.2.3 BRK Interrupt

A BRK interrupt is generated when the BRK instruction is executed.

12.1.2.4 INT Instruction Interrupt

An INT instruction interrupt is generated when the INT instruction is executed. The INT instruction can select software interrupt numbers 0 to 63. Software interrupt numbers 4 to 31 are assigned to the peripheral function interrupt. Therefore, the MCU executes the same interrupt routine when the INT instruction is executed as when a peripheral function interrupt is generated. For software interrupt numbers 0 to 31, the U flag is saved to the stack during instruction execution and the U flag is set to 0 (ISP selected) before the interrupt sequence is executed. The U flag is restored from the stack when returning from the interrupt routine. For software interrupt numbers 32 to 63, the U flag does not change state during instruction execution, and the selected SP is used.

12.1.3 Special Interrupts

Special interrupts are non-maskable.

12.1.3.1 Watchdog Timer Interrupt

The watchdog timer interrupt is generated by the watchdog timer. Reset the watchdog timer after the watchdog timer interrupt is generated. For details, refer to **13. Watchdog Timer**.

12.1.3.2 Oscillation Stop Detection Interrupt

The oscillation stop detection interrupt is generated by the oscillation stop detection function. For details of the oscillation stop detection function, refer to **10. Clock Generation Circuit**.

12.1.3.3 Voltage Monitor 2 Interrupt

The voltage monitor 2 interrupt is generated by the voltage detection circuit. For details of the voltage detection circuit, refer to **7. Voltage Detection Circuit**.

12.1.3.4 Single-Step Interrupt, and Address Break Interrupt

Do not use these interrupts. They are for use by development tools only.

12.1.3.5 Address Match Interrupt

The address match interrupt is generated immediately before executing an instruction that is stored at an address indicated by registers RMAD0 to RMAD1 when the AIER0 or AIER1 bit in the AIER register is set to 1 (address match interrupt enable). For details of the address match interrupt, refer to **12.4 Address Match Interrupt**.

12.1.4 Peripheral Function Interrupt

The peripheral function interrupt is generated by the internal peripheral function of the MCU and is a maskable interrupt. Refer to **Table 12.2 Relocatable Vector Tables** for sources of the peripheral function interrupt. For details of peripheral functions, refer to the descriptions of individual peripheral functions.

12.1.5 Interrupts and Interrupt Vectors

There are 4 bytes in each vector. Set the starting address of an interrupt routine in each interrupt vector. When an interrupt request is acknowledged, the CPU branches to the address set in the corresponding interrupt vector. Figure 12.2 shows an Interrupt Vector.

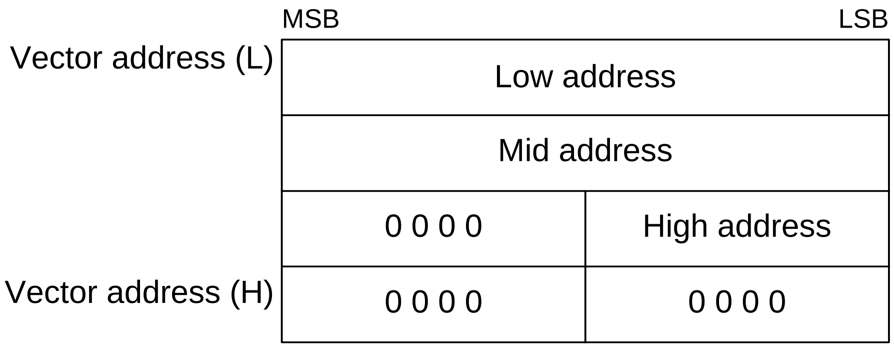


Figure 12.2 Interrupt Vector

12.1.5.1 Fixed Vector Tables

The fixed vector tables are allocated addresses 0FFDCh to 0FFFFh. Table 12.1 lists the Fixed Vector Tables. The vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to **17.3 Functions to Prevent Rewriting of Flash Memory**.

Table 12.1 Fixed Vector Tables

Interrupt Source	Vector Addresses Address (L) to (H)	Remarks	Reference
Undefined instruction	0FFDCh to 0FFDFh	Interrupt on UND instruction	R8C/Tiny Series Software Manual
Overflow	0FFE0h to 0FFE3h	Interrupt on INTO instruction	
BRK instruction	0FFE4h to 0FFE7h	If the content of address 0FFE7h is FFh, program execution starts from the address shown by the vector in the relocatable vector table.	
Address match	0FFE8h to 0FFEBh		12.4 Address Match Interrupt
Single step[1]	0FFECh to 0FFEFh		
• Watchdog timer • Oscillation stop detection • Voltage monitor 2	0FFF0h to 0FFF3h		• 13. Watchdog Timer • 10. Clock Generation Circuit • 7. Voltage Detection Circuit
Address break[1]	0FFF4h to 0FFF7h		
(Reserved)	0FFF8h to 0FFFBh		
Reset	0FFFCh to 0FFFFh		5. Resets

NOTE:

1. Do not use these interrupts. They are for use by development tools only.

12.1.5.2 Relocatable Vector Tables

The relocatable vector tables occupy 256 bytes beginning from the starting address set in the INTB register. Table 12.2 lists the Relocatable Vector Tables.

Table 12.2 Relocatable Vector Tables

Interrupt Source	Vector Address[1] Address (L) to Address (H)	Software Interrupt Number	Reference
BRK instruction[2]	+0 to +3 (0000h to 0003h)	0	R8C/Tiny Series Software Manual
(Reserved)		1 to 12	
Key input	+52 to +55 (0034h to 0037h)	13	12.3 Key Input Interrupt
Comparator conversion	+56 to +59 (0038h to 003Bh)	14	16. Comparator
(Reserved)		15	
Compare 1	+64 to +67 (0040h to 0043h)	16	14.3 Timer C
UART0 transmit	+68 to +71 (0044h to 0047h)	17	15. Serial Interface
UART0 receive	+72 to +75 (0048h to 004Bh)	18	
UART1 transmit	+76 to +79 (004Ch to 004Fh)	19	
UART1 receive	+80 to +83 (0050h to 0053h)	20	
(Reserved)		21	
Timer X	+88 to +91 (0058h to 005Bh)	22	14.1 Timer X
(Reserved)		23	
Timer Z	+96 to +99 (0060h to 0063h)	24	14.2 Timer Z
$\overline{\text{INT1}}$	+100 to +103 (0064h to 0067h)	25	12.2 $\overline{\text{INT}}$ interrupt
$\overline{\text{INT3}}$	+104 to +107 (0068h to 006Bh)	26	
Timer C	+108 to +111 (006Ch to 006Fh)	27	14.3 Timer C
Compare 0	+112 to +115 (0070h to 0073h)	28	
$\overline{\text{INT0}}$	+116 to +119 (0074h to 0077h)	29	12.2 $\overline{\text{INT}}$ interrupt
(Reserved)		30	
(Reserved)		31	
Software interrupt[2]	+128 to +131 (0080h to 0083h) to +252 to +255 (00FCh to 00FFh)	32 to 63	R8C/Tiny Series Software Manual

NOTES:

1. These addresses are relative to those in the INTB register.
2. The I flag does not disable these interrupts.

12.1.6 Interrupt Control

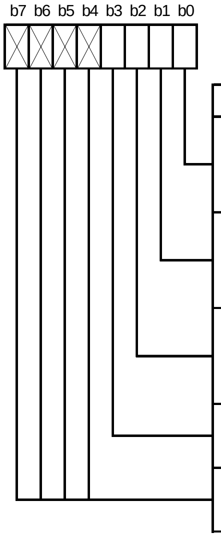
The following describes enabling and disabling the maskable interrupts and setting the priority for acknowledgement. The explanation does not apply to nonmaskable interrupts.

Use the I flag in the FLG register, IPL, and bits ILVL2 to ILVL0 in each interrupt control register to enable or disable maskable interrupts. Whether an interrupt is requested is indicated by the IR bit in each interrupt control register.

Figure 12.3 shows the Interrupt Control Register and Figure 12.4 shows the INT0IC Register

Interrupt Control Register[(2)]

Symbol	Address	After Reset
KUPIC	004Dh	XXXXX000b
ADIC	004Eh	XXXXX000b
CMP1IC	0050h	XXXXX000b
S0TIC, S1TIC	0051h, 0053h	XXXXX000b
S0RIC, S1RIC	0052h, 0054h	XXXXX000b
TXIC	0056h	XXXXX000b
TZIC	0058h	XXXXX000b
INT1IC	0059h	XXXXX000b
INT3IC	005Ah	XXXXX000b
TCIC	005Bh	XXXXX000b
CMP0IC	005Ch	XXXXX000b



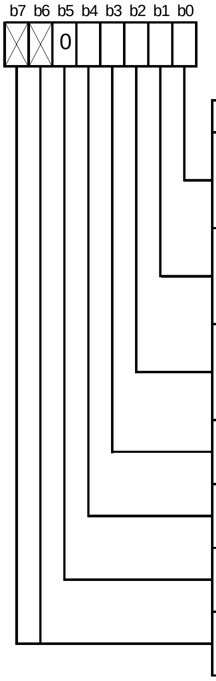
Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (interrupt disable) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0 : Requests no interrupt 1 : Requests interrupt	RW[(1)]
— (b7-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		—

NOTES:

1. Only 0 can be written to the IR bit. Do not write 1.
2. Rewrite the interrupt control register when the interrupt request which is applicable for the register is not generated. Refer to **12.5.6 Changing Interrupt Control Register Contents.**

Figure 12.3 Interrupt Control Register

INT0 Interrupt Control Register[2]



Symbol	Address	After Reset
INT01C	005Dh	XX00X000b

Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (interrupt disable) 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 bit	0 : Requests no interrupt. 1 : Requests interrupt.	RW[1]
POL	Polarity sw itch bit[4]	0 : Selects falling edge. 1 : Selects rising edge.[3]	RW
— (b5)	Reserved bit	Set to 0.	RW
— (b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		—

NOTES:

1. Only 0 can be w ritten to the IR bit. (Do not w rite 1.)
2. Rew rite the interrupt control register w hen the interrupt request w hich is applicable for the register is not generated. Refer to **12.5.6 Changing Interrupt Control Register Contents.**
3. If the INTOPL bit in the INTEN register is set to 1 (both edges), set the POL bit to 0 (selects falling edge).
4. The IR bit may be set to 1 (requests interrupt) w hen the POL bit is rew ritten. Refer to **12.5.5 Changing Interrupt Sources**.

Figure 12.4 INT0IC Register

12.1.6.1 I Flag

The I flag enables or disables maskable interrupts. Setting the I flag to 1 (enabled) enables maskable interrupts. Setting the I flag to 0 (disabled) disables all maskable interrupts.

12.1.6.2 IR Bit

The IR bit is set to 1 (interrupt requested) when an interrupt request is generated. Then, when the interrupt request is acknowledged and the CPU branches to the corresponding interrupt vector, the IR bit is set to 0 (= interrupt not requested).
The IR bit can be set to 0 by a program. Do not write 1 to this bit.

12.1.6.3 Bits ILVL2 to ILVL0 and IPL

Interrupt priority levels can be set using bits ILVL2 to ILVL0.
Table 12.3 lists the Settings of Interrupt Priority Levels and Table 12.4 lists the Interrupt Priority Levels Enabled by IPL.

The following are conditions under which an interrupt is acknowledged:

- I flag = 1
- IR bit = 1
- Interrupt priority level > IPL

The I flag, IR bit, bits ILVL2 to ILVL0 and IPL are independent of each other. They do not affect one another.

Table 12.3 Settings of Interrupt Priority Levels

ILVL2 to ILVL0 Bits	Interrupt Priority Level	Priority Order
000b	Level 0 (interrupt disabled)	–
001b	Level 1	Low
010b	Level 2	
011b	Level 3	
100b	Level 4	
101b	Level 5	
110b	Level 6	
111b	Level 7	High

Table 12.4 Interrupt Priority Levels Enabled by IPL

IPL	Enabled Interrupt Priority Levels
000b	Interrupt level 1 and above
001b	Interrupt level 2 and above
010b	Interrupt level 3 and above
011b	Interrupt level 4 and above
100b	Interrupt level 5 and above
101b	Interrupt level 6 and above
110b	Interrupt level 7 and above
111b	All maskable interrupts are disabled

12.1.6.4 Interrupt Sequence

An interrupt sequence is performed between an interrupt request acknowledgement and interrupt routine execution.
When an interrupt request is generated while an instruction is being executed, the CPU determines its interrupt priority level after the instruction is completed. The CPU starts the interrupt sequence from the following cycle. However, for the SMOVB, SMOVF, SSTR, or RMPA instruction, if an interrupt request is generated while the instruction is being executed, the MCU suspends the instruction to start the interrupt sequence. The interrupt sequence is performed as indicated below.
Figure 12.5 shows the Time Required for Executing Interrupt Sequence.

(1) The CPU gets interrupt information (interrupt number and interrupt request level) by reading address 00000h. The IR bit for the corresponding interrupt is set to 0 (interrupt not requested).

(2) The FLG register is saved to a temporary register[1] in the CPU immediately before entering the interrupt sequence.

(3) The I, D, and U flags in the FLG register are set as follows:
The I flag is set to 0 (interrupts disabled).
The D flag is set to 0 (single-step interrupt disabled).
The U flag is set to 0 (ISP selected).
However, the U flag does not change state if an INT instruction for software interrupt numbers 32 to 63 is executed.

(4) The CPU’s internal temporary register[1] is saved to the stack.

(5) The PC is saved to the stack.

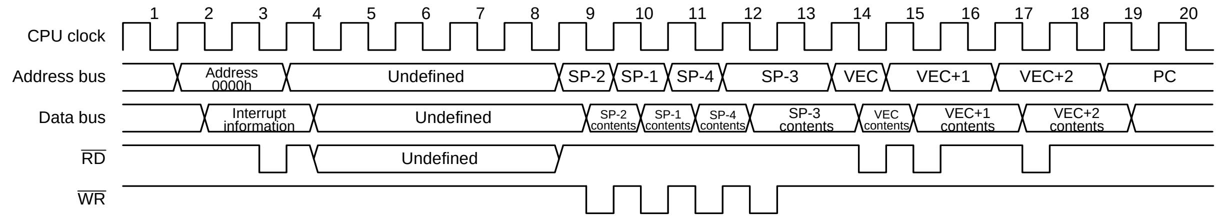
(6) The interrupt priority level of the acknowledged interrupt is set in the IPL.

(7) The starting address of the interrupt routine set in the interrupt vector is stored in the PC.

After the interrupt sequence is completed, instructions are executed from the starting address of the interrupt routine.

NOTE:

1. **This register cannot be used by user.**

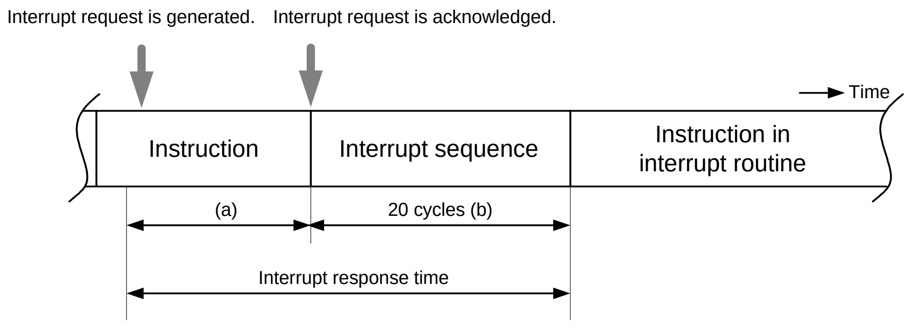


The undefined state depends on the instruction queue buffer. A read cycle occurs when the instruction queue buffer is ready to acknowledge instructions.

Figure 12.5 Time Required for Executing Interrupt Sequence

12.1.6.5 Interrupt Response Time

Figure 12.6 shows the Interrupt Response Time. The interrupt response time is the period between an interrupt request generation and the execution of the first instruction in the interrupt routine. The interrupt response time includes the period between interrupt request generation and the completion of execution of the instruction (refer to (a) in Figure 12.6) and the period required to perform the interrupt sequence (20 cycles, refer to (b) in Figure 12.6).



Interrupt request is generated.
Interrupt request is acknowledged.
Time
Instruction
Interrupt sequence
Instruction in interrupt routine
(a)
20 cycles (b)
Interrupt response time

(a) Period between interrupt request generation and the completion of execution of an instruction. The length of time varies depending on the instruction being executed. The DIVX instruction requires the longest time, 30 cycles (assuming no wait states and that a register is set as the divisor).
(b) 21 cycles for address match and single-step interrupts.

Figure 12.6 Interrupt Response Time

12.1.6.6 IPL Change when Interrupt Request is Acknowledged

When an interrupt request of a maskable interrupt is acknowledged, the interrupt priority level of the acknowledged interrupt is set in the IPL.
When a software interrupt or special interrupt request is acknowledged, the level listed in Table 12.5 is set in the IPL. Table 12.5 lists the IPL Value When Software or Special Interrupt Is Acknowledged.

Table 12.5 IPL Value When Software or Special Interrupt Is Acknowledged

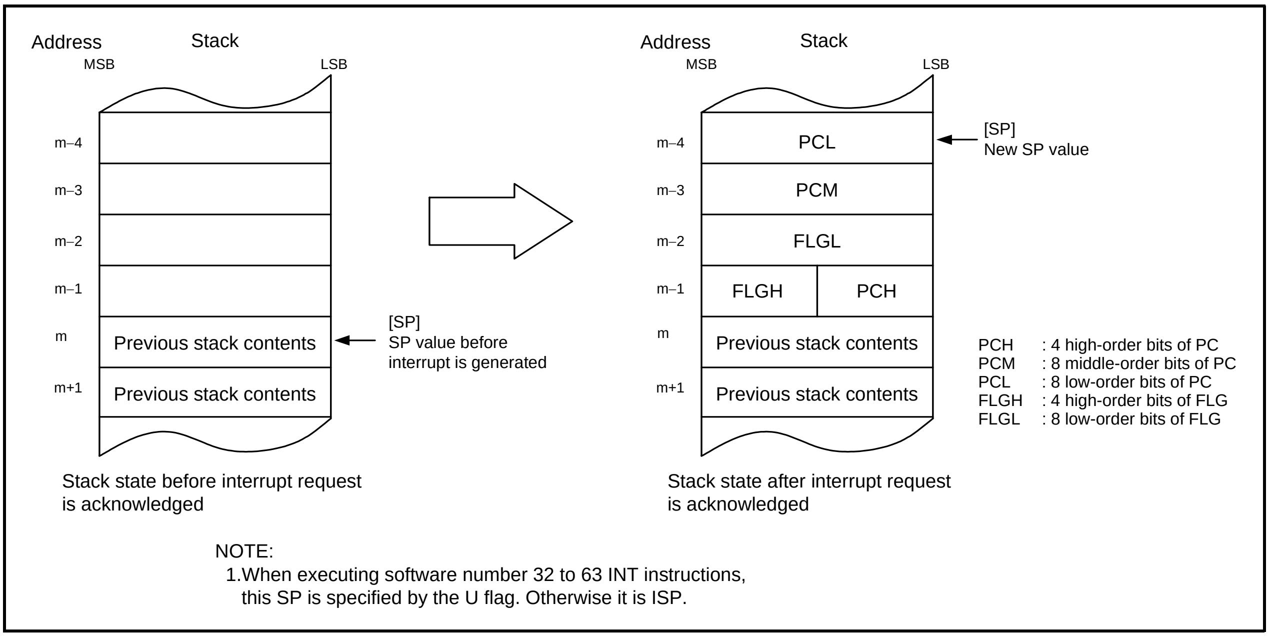
Interrupt Source	Value Set in IPL
Watchdog timer, oscillation stop detection, voltage monitor 2	7
Software, address match, single-step, address break	Not changed

12.1.6.7 Saving a Register

In the interrupt sequence, the FLG register and PC are saved to the stack.
After an extended 16 bits, 4 high-order bits in the PC and 4 high-order (IPL) and 8 low-order bits in the FLG register, are saved to the stack, the 16 low-order bits in the PC are saved. Figure 12.7 shows the Stack State Before and After Acknowledgement of Interrupt Request.
The other necessary registers are saved by a program at the beginning of the interrupt routine. The PUSHM instruction can save several registers in the register bank being currently used(1) with a single instruction.

NOTE:

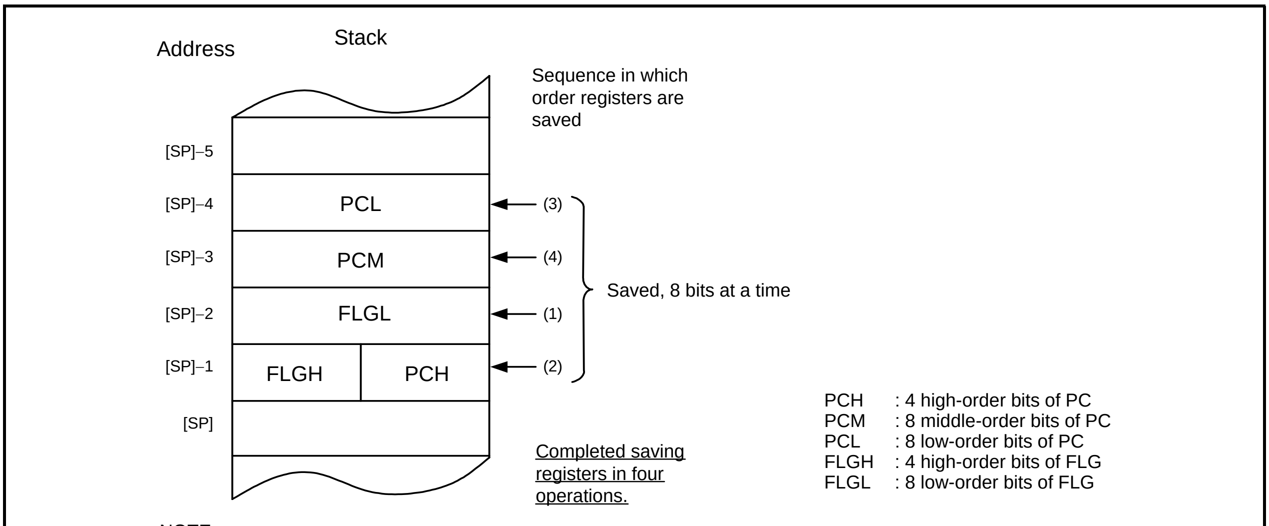
1. **Selectable from registers R0, R1, R2, R3, A0, A1, SB, and FB.**


Address
Stack
MSB
LSB
m–4
m–3
m–2
m–1
m
m+1
Previous stack contents
Previous stack contents
[SP]
SP value before interrupt is generated
Stack state before interrupt request is acknowledged
Address
Stack
MSB
LSB
m–4
m–3
m–2
m–1
m
m+1
PCL
PCM
FLGL
FLGH
PCH
Previous stack contents
Previous stack contents
[SP]
New SP value
PCH : 4 high-order bits of PC
PCM : 8 middle-order bits of PC
PCL : 8 low-order bits of PC
FLGH : 4 high-order bits of FLG
FLGL : 8 low-order bits of FLG
Stack state after interrupt request is acknowledged

NOTE:
1.When executing software number 32 to 63 INT instructions, this SP is specified by the U flag. Otherwise it is ISP.

Figure 12.7 Stack State Before and After Acknowledgement of Interrupt Request

The register saving operation, which is performed as part of the interrupt sequence, saved in 8 bits at a time in four steps. Figure 12.8 shows the Register Saving Operation.
.


Address
Stack
Sequence in which order registers are saved
[SP]–5
[SP]–4
[SP]–3
[SP]–2
[SP]–1
[SP]
PCL
PCM
FLGL
FLGH
PCH
(3)
(4)
(1)
(2)
Saved, 8 bits at a time
Completed saving registers in four operations.
PCH : 4 high-order bits of PC
PCM : 8 middle-order bits of PC
PCL : 8 low-order bits of PC
FLGH : 4 high-order bits of FLG
FLGL : 8 low-order bits of FLG

NOTE:
1. [SP] indicates the initial value of the SP when an interrupt request is acknowledged.
After registers are saved, the SP content is [SP] minus 4. When executing software number 32 to 63 INT instructions, this SP is specified by the U flag. Otherwise it is ISP.

Figure 12.8 Register Saving Operation

12.1.6.8 Returning from an Interrupt Routine

When the REIT instruction is executed at the end of an interrupt routine, the FLG register and PC, which have been saved to the stack, are automatically restored. The program, that was running before the interrupt request was acknowledged, starts running again.
Restore registers saved by a program in an interrupt routine using the POPM instruction or others before executing the REIT instruction.

12.1.6.9 Interrupt Priority

If two or more interrupt requests are generated while a single instruction is being executed, the interrupt with the higher priority is acknowledged.
Set bits ILVL2 to ILVL0 to select the desired priority level for maskable interrupts (peripheral functions). However, if two or more maskable interrupts have the same priority level, their interrupt priority is resolved by hardware, and the higher priority interrupts acknowledged.
The priority levels of special interrupts, such as reset (reset has the highest priority) and watchdog timer, are set by hardware. Figure 12.9 shows the Priority Levels of Hardware Interrupts.
The interrupt priority does not affect software interrupts. The MCU jumps to the interrupt routine when the instruction is executed.

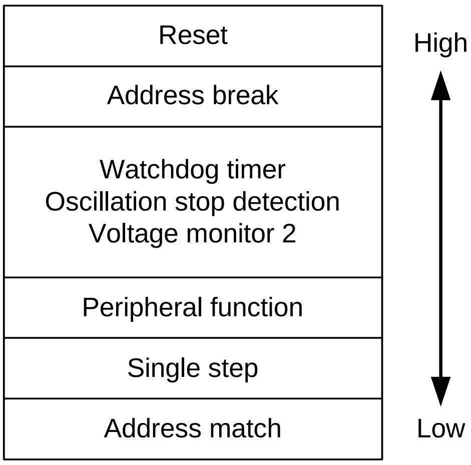


Figure 12.9 Priority Levels of Hardware Interrupts

12.1.6.10 Interrupt Priority Judgement Circuit

The interrupt priority judgement circuit selects the highest priority interrupt, as shown in Figure 12.10.

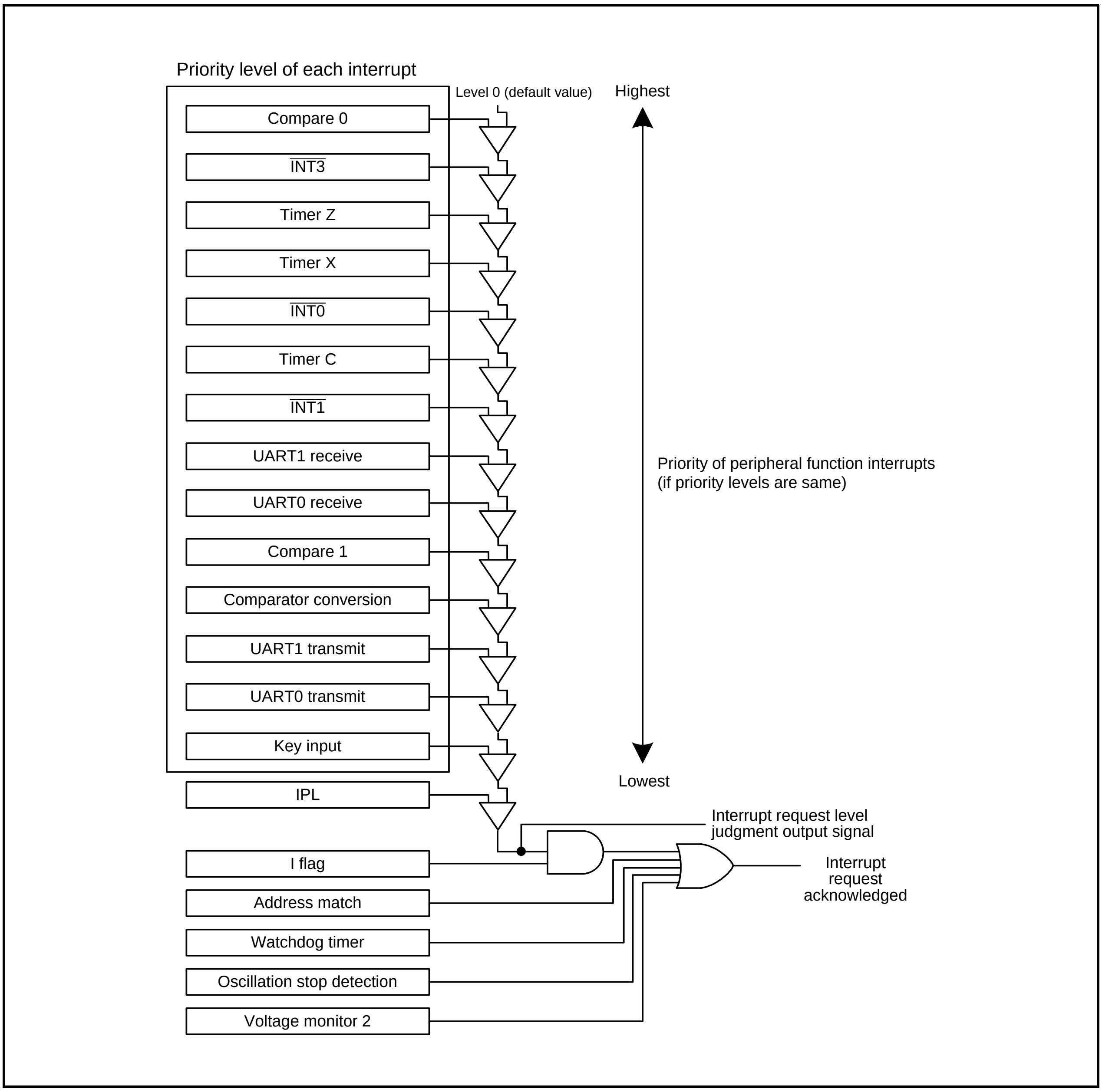


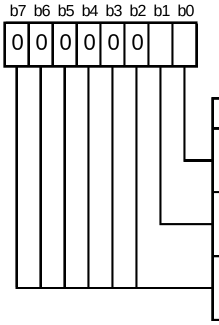
Figure 12.10 Interrupt Priority Level Judgement Circuit

12.2 $\overline{\text{INT}}$ Interrupt

12.2.1 $\overline{\text{INT0}}$ Interrupt

The $\overline{\text{INT0}}$ interrupt is generated by an $\overline{\text{INT0}}$ input. When using the $\overline{\text{INT0}}$ interrupt, the INT0EN bit in the INTEN register is set to 1 (enable). The edge polarity is selected using the INT0PL bit in the INTEN register and the POL bit in the INT0IC register.
Inputs can be passed through a digital filter with three different sampling clocks.
The $\overline{\text{INT0}}$ pin is shared with the external trigger input pin of timer Z.
Figure 12.11 shows Registers INTEN and INT0F.

External Input Enable Register

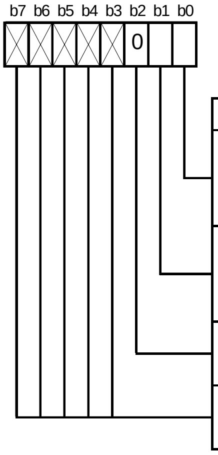


Symbol	Address	After Reset
INTEN	0096h	00h

Bit Symbol	Bit Name	Function	RW
INT0EN	$\overline{\text{INT0}}$ input enable bit[1]	0 : Disable 1 : Enable	RW
INT0PL	$\overline{\text{INT0}}$ input polarity select bit[2, 3]	0 : One edge 1 : Both edges	RW
— (b7-b2)	Reserved bits	Set to 0.	RW

NOTES:
1. Set the INT0EN bit while the INOSTG bit in the PUM register is set to 0 (one-shot trigger disabled).
2. When setting the INT0PL bit to 1 (both edges), set the POL bit in the INT0IC register to 0 (selects falling edge).
3. The IR bit in the INT0IC register may be set to 1 (requests interrupt) when the INT0PL bit is rewritten. Refer to **12.5.5 Changing Interrupt Sources**.

$\overline{\text{INT0}}$ Input Filter Select Register



Symbol	Address	After Reset
INT0F	001Eh	00h

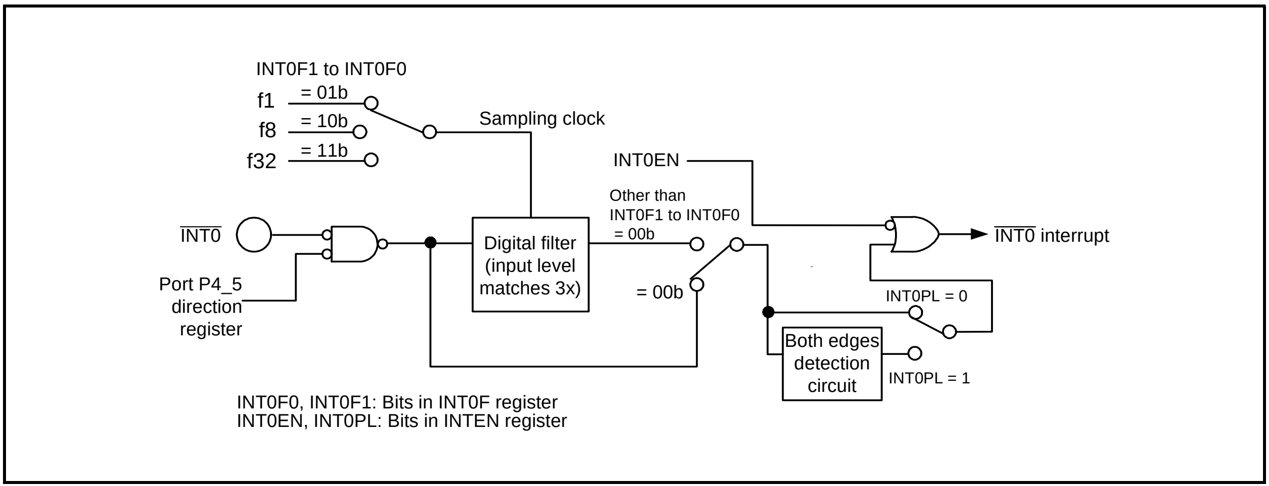
Bit Symbol	Bit Name	Function	RW
INT0F0	$\overline{\text{INT0}}$ input filter select bits	b1 b0 0 0 : No filter 0 1 : Filter with f1 sampling 1 0 : Filter with f8 sampling 1 1 : Filter with f32 sampling	RW
INT0F1			RW
— (b2)	Reserved bit	Set to 0.	RW
— (b7-b3)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		—

Figure 12.11 Registers INTEN and INT0F

12.2.2 $\overline{\text{INT0}}$ Input Filter

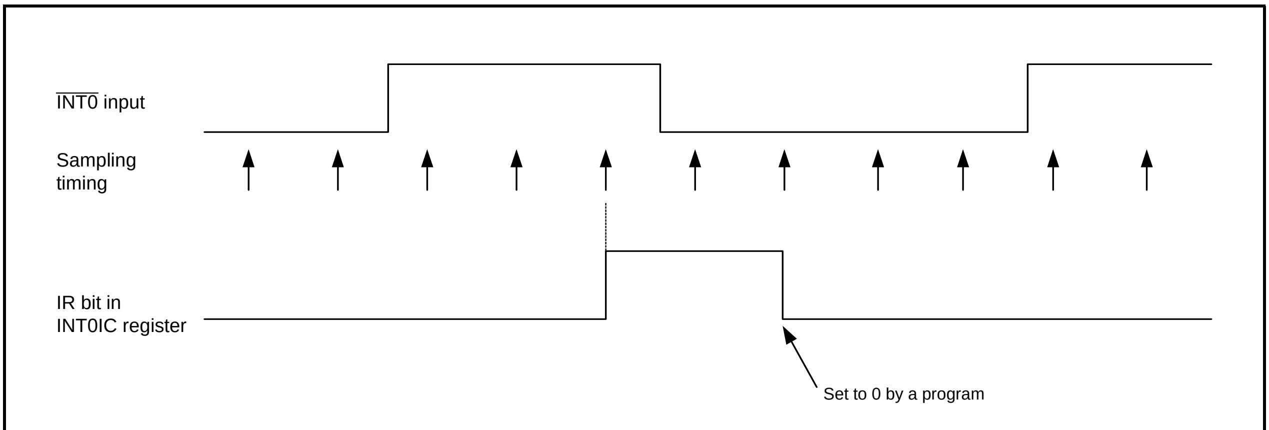
The $\overline{\text{INT0}}$ input contains a digital filter. The sampling clock is selected by bits INT0F1 to INT0F0 in the INT0F register. The $\overline{\text{INT0}}$ level is sampled every sampling clock cycle and if the sampled input level matches three times, the IR bit in the INT0IC register is set to 1 (interrupt requested).

Figure 12.12 shows the Configuration of INT0 Input Filter. Figure 12.13 shows an Operating Example of INT0 Input Filter.



INT0F0, INT0F1: Bits in INT0F register
INT0EN, INT0PL: Bits in INTEN register

Figure 12.12 Configuration of $\overline{\text{INT0}}$ Input Filter



This is an operating example in which bits INT0F1 to INT0F0 in the INT0F register are set to 01b, 10b, or 11b (digital filter enabled).

Figure 12.13 Operating Example of $\overline{\text{INT0}}$ Input Filter

12.2.3 INT1 Interrupt

The INT1 interrupt is generated by an INT1 input. The edge polarity is selected by the R0EDG bit in the TXMR register.
When the CNTRSEL bit in the UCON register is set to 0, the INT10 pin becomes the INT1 input pin. When the CNTRSEL bit is set to 1, the INT11 pin becomes the INT1 input pin.
The INT10 pin is shared with the CNTR00 pin and the INT11 pin is shared with the CNTR01 pin.
Figure 12.14 shows the TXMR Register when INT1 Interrupt is Used.

Timer X Mode Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
TXMR	008Bh	00h

Bit Symbol	Bit Name	Function	RW
TXMOD0	Operating mode select bits 0, 1[1]	b1 b0 0 0 : Timer mode or pulse period measurement mode 0 1 : Do not set. 1 0 : Event count mode 1 1 : Pulse width measurement mode	RW
TXMOD1			RW
R0EDG	INT1/CNTR0 polarity switch bit[2]	0 : Rising edge 1 : Falling edge	RW
TXS	Timer X count start flag[3]	0 : Stops counting. 1 : Starts counting.	RW
TXOCNT	P3_7/CNTR0 select bit	Function varies depending on operating mode.	RW
TXMOD2	Operating mode select bit 2	0 : Other than pulse period measurement mode 1 : Pulse period measurement mode	RW
TXEDG	Active edge reception flag	Function varies depending on operating mode.	RW
TXUND	Timer X underflow flag	Function varies depending on operating mode.	RW

NOTES:

1. When using INT1 interrupt, select modes other than pulse output mode.
2. The IR bit in the INT1IC register may be set to 1 (requests interrupt) when the R0EDG bit is rewritten. Refer to **12.5.5 Changing Interrupt Sources**.
3. Refer to **14.1.6 Notes on Timer X** for precautions regarding the TXS bit.

Figure 12.14 TXMR Register when INT1 Interrupt is Used

12.2.4 INT3 Interrupt

The INT3 interrupt is generated by an INT3 input. Set the TCC07 bit in the TCC0 register to 0 (INT3). When the TCC06 bit in the TCC0 register is set to 0, an INT3 interrupt request is generated in synchronization with the count source of timer C. If the TCC06 bit is set to 1, the INT3 interrupt request is generated when an INT3 input occurs.

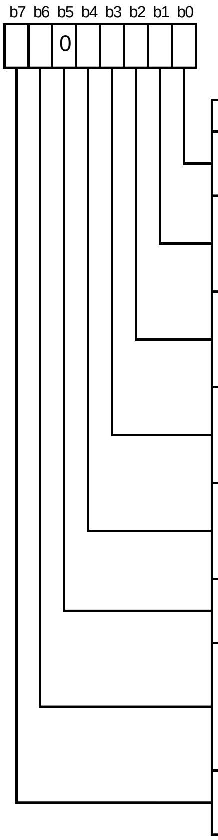
The INT3 input contains a digital filter. The INT3 level is sampled every sampling clock cycle and if the sampled input level matches three times, the IR bit in the INT3IC register is set to 1 (interrupt requested). The sampling clock is selected by bits TCC11 to TCC10 in the TCC1 register. If filter is selected, the interrupt request is generated in synchronization with the sampling clock, even if the TCC06 bit is set to 1. The P3_3 bit in the P3 register indicates the value before filtering regardless of the contents set in bits TCC11 to TCC10.

The INT3 pin is used with the TCIN pin.

If the TCC07 bit is set to 1 (fRING128), the INT3 interrupt is generated by the fRING128 clock. The IR bit in the INT3IC register is set to 1 (interrupt requested) every fRING128 clock cycle or every half fRING128 clock cycle.

Figure 12.15 shows the TCC0 Register and Figure 12.16 shows the TCC1 Register.

Timer C Control Register 0



Symbol	Address	After Reset
TCC0	009Ah	00h

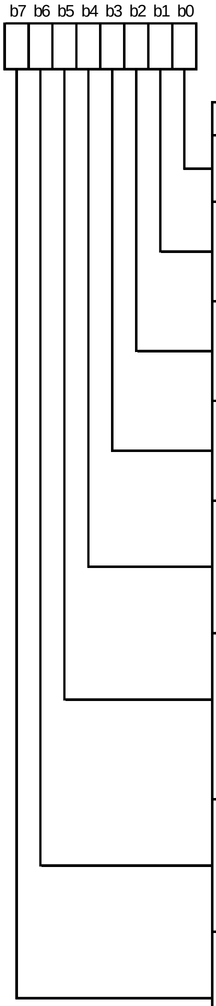
Bit Symbol	Bit Name	Function	RW
TCC00	Timer C count start bit	0 : Stops counting. 1 : Starts counting.	RW
TCC01	Timer C count source select bits[1]	b2 b1 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : fRING-fast	RW
TCC02			RW
TCC03	INT3 interrupt and capture polarity select bits[1, 2]	b4 b3 0 0 : Rising edge 0 1 : Falling edge 1 0 : Both edges 1 1 : Do not set.	RW
TCC04			RW
— (b5)	Reserved bit	Set to 0.	RW
TCC06	INT3 interrupt request generation timing select bit[2, 3]	0 : INT3 interrupt is generated in synchronization with timer C count source. 1 : INT3 interrupt is generated when INT3 interrupt is input.[4]	RW
TCC07	INT3 interrupt and capture input switch bit[1, 2]	0 : INT3 1 : fRING128	RW

NOTES:

1. Change this bit when the TCC00 bit is set to 0 (count stops).
2. The IR bit in the INT3IC register may be set to 1 (requests interrupt) when the TCC03, TCC04, TCC06, or TCC07 bit is rewritten. Refer to **12.5.5 Changing Interrupt Sources**.
3. When the TCC13 bit is set to 1 (output compare mode) and an INT3 interrupt is input, regardless of the setting value of the TCC06 bit, an interrupt request is generated.
4. When using the INT3 filter, the INT3 interrupt is generated in synchronization with the clock for the digital filter.

Figure 12.15 TCC0 Register

Timer C Control Register 1



Symbol	Address	After Reset
TCC1	009Bh	00h

Bit Symbol	Bit Name	Function	RW
TCC10	$\overline{\mathrm{INT3}}$ filter select bits[1]	b1b0 0 0 : No filter 0 1 : Filter with f1 sampling 1 0 : Filter with f8 sampling 1 1 : Filter with f32 sampling	RW
TCC11			RW
TCC12	Timer C counter reload select bit[2, 3]	0 : No reload 1 : Set TC register to 0000h when compare 1 is matched.	RW
TCC13	Compare 0/capture select bit	0 : Capture select (input capture mode)[2] 1 : Compare 0 output select (output compare mode)	RW
TCC14	Compare 0 output mode select bits[3]	b5 b4 0 0 : CMP output remains unchanged even when compare 0 is matched. 0 1 : CMP output is reversed when compare 0 signal is matched. 1 0 : CMP output is set to "L" when compare 0 signal is matched. 1 1 : CMP output is set to "H" when compare 0 signal is matched.	RW
TCC15			RW
TCC16	Compare 1 output mode select bits[3]	b7 b6 0 0 : CMP output remains unchanged even when compare 1 is matched. 0 1 : CMP output is reversed when compare 1 signal is matched. 1 0 : CMP output is set to "L" when compare 1 signal is matched. 1 1 : CMP output is set to "H" when compare 1 signal is matched.	RW
TCC17			RW

NOTES:

1. When the same value from the $\overline{\mathrm{INT3}}$ pin is sampled three times continuously, the input is determined.
2. When the TCC00 bit in the TCC0 register is set to 0 (count stops), rewrite the TCC13 bit.
3. When the TCC13 bit is set to 0 (input capture mode), set bits TCC12 and TCC14 to TCC17 to 0.

Figure 12.16 TCC1 Register

12.3 Key Input Interrupt

A key input interrupt request is generated by one of the input edges of pins $\overline{\text{K10}}$ to $\overline{\text{K13}}$. The key input interrupt can be used as a key-on wake-up function to exit wait or stop mode.
The KIiEN (i = 0 to 3) bit in the KIEN register can select whether or not the pins are used as $\overline{\text{KIi}}$ input.
The KIiPL bit in the KIEN register can select the input polarity.
When “L” is input to the $\overline{\text{KIi}}$ pin, which sets the KIiPL bit to 0 (falling edge), input to the other pins $\overline{\text{K10}}$ to $\overline{\text{K13}}$ is not detected as interrupts. Also, when “H” is input to the $\overline{\text{KIi}}$ pin, which sets the KIiPL bit to 1 (rising edge), input to the other pins $\overline{\text{K10}}$ to $\overline{\text{K13}}$ is not detected as interrupts.
Figure 12.17 shows a Block Diagram of Key Input Interrupt.

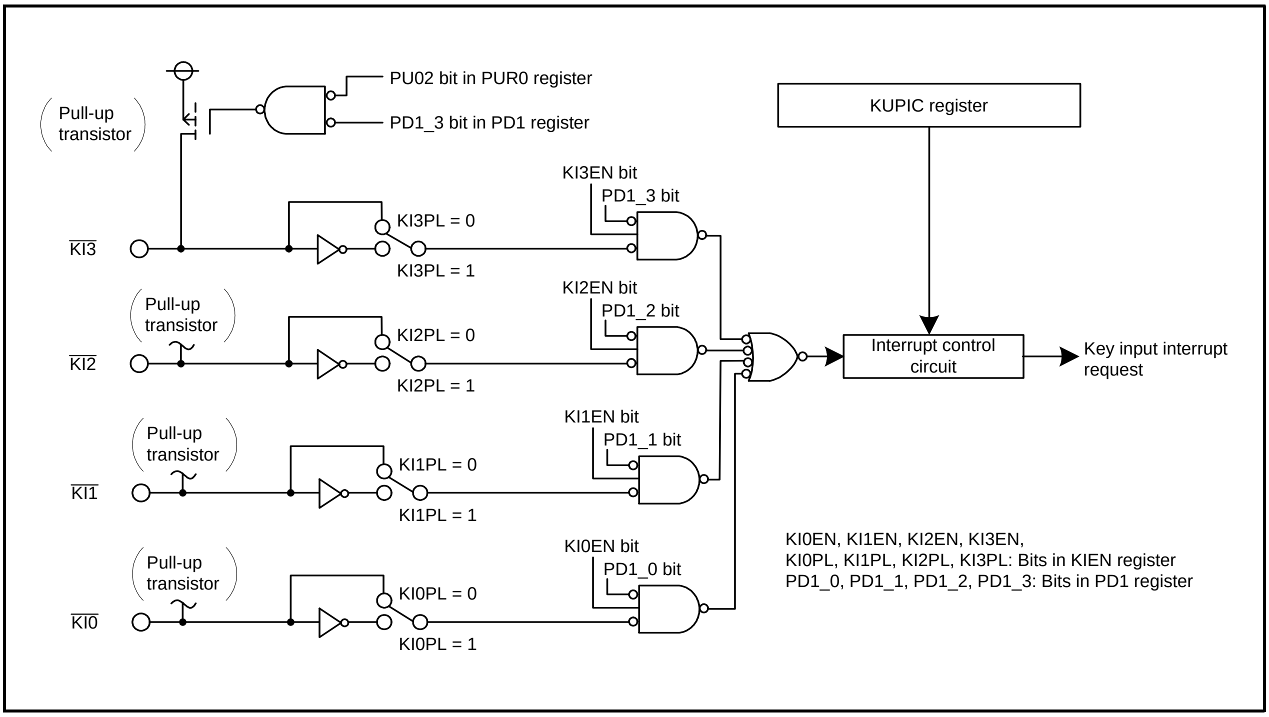
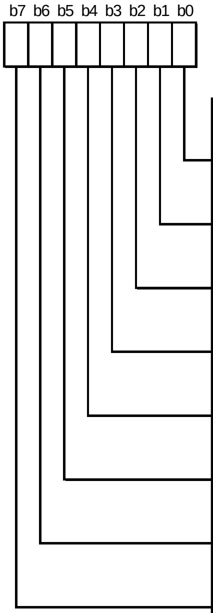


Figure 12.17 Block Diagram of Key Input Interrupt

Key Input Enable Register[1]



Symbol	Address	After Reset
KIEN	0098h	00h

Bit Symbol	Bit Name	Function	RW
KI0EN	KI0 input enable bit	0 : Disable 1 : Enable	RW
KI0PL	KI0 input polarity select bit	0 : Falling edge 1 : Rising edge	RW
KI1EN	KI1 input enable bit	0 : Disable 1 : Enable	RW
KI1PL	KI1 input polarity select bit	0 : Falling edge 1 : Rising edge	RW
KI2EN	KI2 input enable bit	0 : Disable 1 : Enable	RW
KI2PL	KI2 input polarity select bit	0 : Falling edge 1 : Rising edge	RW
KI3EN	KI3 input enable bit	0 : Disable 1 : Enable	RW
KI3PL	KI3 input polarity select bit	0 : Falling edge 1 : Rising edge	RW

NOTE:

1. The IR bit in the KUPIC register may be set to 1 (requests interrupt) when the KIEN register is rewritten. Refer to **12.5.5 Changing Interrupt Sources.**

Figure 12.18 KIEN Register

12.4 Address Match Interrupt

An address match interrupt request is generated immediately before execution of the instruction at the address indicated by the RMADi register (i = 0, 1). This interrupt is used as a break function by the debugger. When using the on-chip debugger, do not set an address match interrupt (registers of AIER, RMAD0, and RMAD1 and fixed vector tables) in a user system.

Set the starting address of any instruction in the RMADi register. Bits AIER0 and AIER1 in the AIER0 register can be used to select enable or disable of the interrupt. The I flag and IPL do not affect the address match interrupt.
The value of the PC (Refer to **12.1.6.7 Saving a Register** for the value of the PC) which is saved to the stack when an address match interrupt is acknowledged varies depending on the instruction at the address indicated by the RMADi register. (The appropriate return address is not saved on the stack.)
When returning from the address match interrupt, return by one of the following means:

- Change the content of the stack and use the REIT instruction.
- Use an instruction such as POP to restore the stack as it was before the interrupt request was acknowledged. Then use a jump instruction.

Table 12.6 lists the Values of PC Saved to Stack when Address Match Interrupt is Acknowledged.
Figure 12.19 shows Registers AIER and RMAD0 to RMAD1.

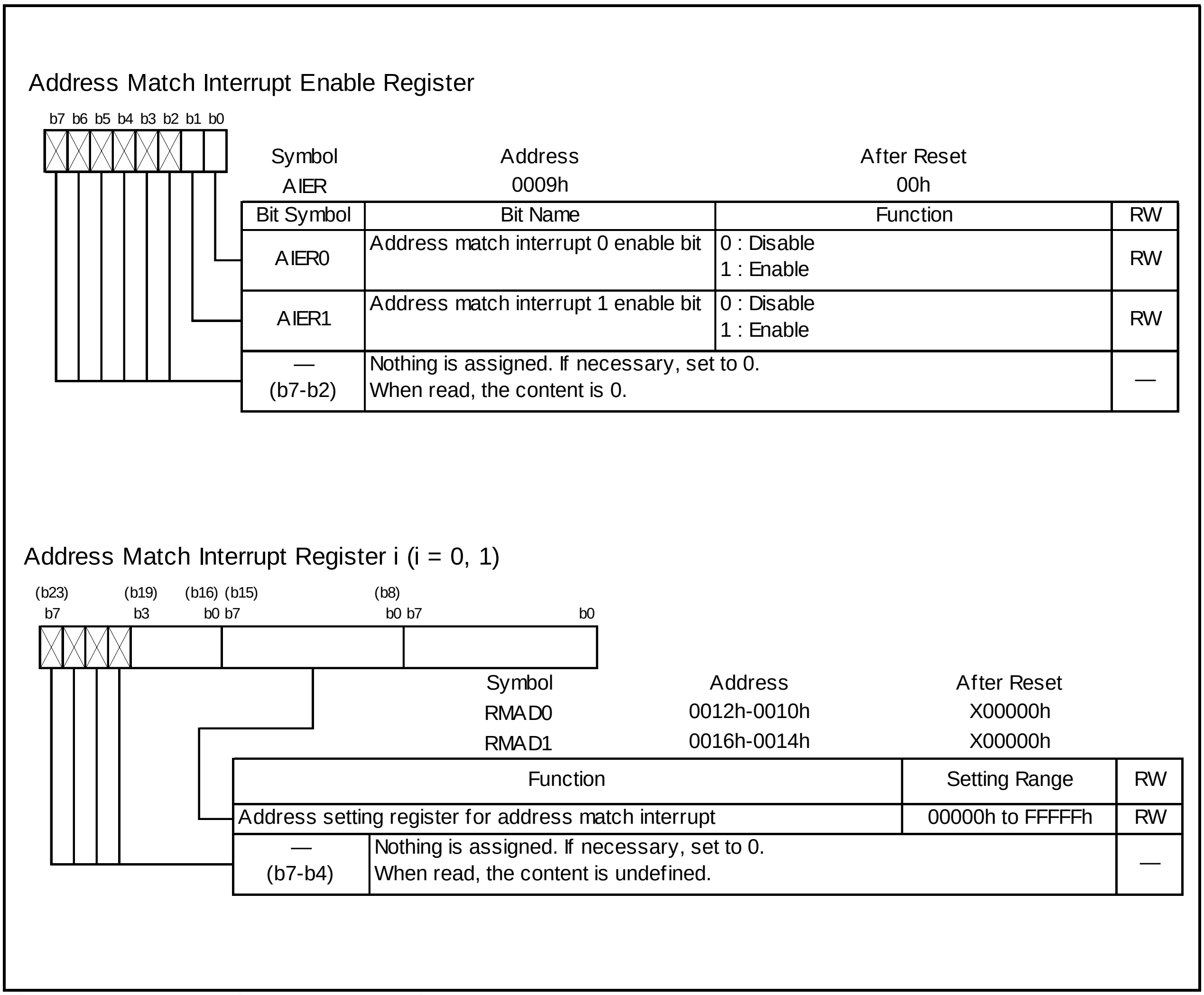
Table 12.6 Values of PC Saved to Stack when Address Match Interrupt is Acknowledged

Address Indicated by RMADi Register (i = 0, 1)	PC Value Saved[1]
• 16-bit operation code instruction • Instruction shown below among 8-bit operation code instructions ADD.B:S #IMM8,dest SUB.B:S #IMM8,dest AND.B:S #IMM8,dest OR.B:S #IMM8,dest MOV.B:S #IMM8,dest STZ.B:S #IMM8,dest STNZ.B:S #IMM8,dest STZX.B:S #IMM81,#IMM82,dest CMP.B:S #IMM8,dest PUSHM src POPM dest JMPS #IMM8 JSRS #IMM8 MOV.B:S #IMM,dest (however, dest = A0 or A1)	Address indicated by RMADi register + 2
• Instructions other than the above	Address indicated by RMADi register + 1

NOTE:
1. Refer to the **12.1.6.7 Saving a Register** for the PC value saved.

Table 12.7 Correspondence Between Address Match Interrupt Sources and Associated Registers

Address Match Interrupt Source	Address Match Interrupt Enable Bit	Address Match Interrupt Register
Address match interrupt 0	AIER0	RMAD0
Address match interrupt 1	AIER1	RMAD1



Address Match Interrupt Enable Register

Symbol	Address	After Reset
AIER	0009h	00h

Bit Symbol	Bit Name	Function	RW
AIER0	Address match interrupt 0 enable bit	0 : Disable 1 : Enable	RW
AIER1	Address match interrupt 1 enable bit	0 : Disable 1 : Enable	RW
— (b7-b2)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		—

Address Match Interrupt Register i (i = 0, 1)

Symbol	Address	After Reset
RMAD0	0012h-0010h	X00000h
RMAD1	0016h-0014h	X00000h

Function		Setting Range	RW
Address setting register for address match interrupt		00000h to FFFFFh	RW
— (b7-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		—

Figure 12.19 Registers AIER and RMAD0 to RMAD1

12.5 Notes on Interrupts

12.5.1 Reading Address 00000h

Do not read address 00000h by a program. When a maskable interrupt request is acknowledged, the CPU reads interrupt information (interrupt number and interrupt request level) from 00000h in the interrupt sequence. At this time, the acknowledged interrupt IR bit is set to 0.
If address 00000h is read by a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is set to 0. This may cause the interrupt to be canceled, or an unexpected interrupt to be generated.

12.5.2 SP Setting

Set any value in the SP before an interrupt is acknowledged. The SP is set to 0000h after reset. Therefore, if an interrupt is acknowledged before setting a value in the SP, the program may run out of control.

12.5.3 External Interrupt and Key Input Interrupt

Either “L” level or “H” level of at least 250 ns width is necessary for the signal input to pins $\overline{\text{INT0}}$ to $\overline{\text{INT3}}$ and pins $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$ regardless of the CPU clock.

12.5.4 Watchdog Timer Interrupt

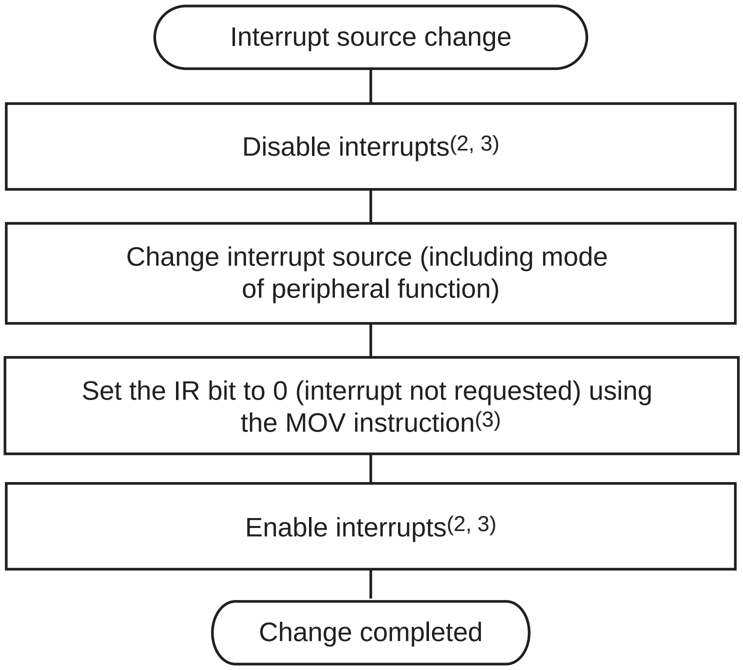
Reset the watchdog timer after a watchdog timer interrupt is generated.

12.5.5 Changing Interrupt Sources

The IR bit in the interrupt control register may be set to 1 (interrupt requested) when the interrupt source changes. When using an interrupt, set the IR bit to 0 (no interrupt requested) after changing the interrupt source.

In addition, changes of interrupt sources include all factors that change the interrupt sources assigned to individual software interrupt numbers, polarities, and timing. Therefore, if a mode change of a peripheral function involves interrupt sources, edge polarities, and timing, set the IR bit to 0 (no interrupt requested) after the change. Refer to the individual peripheral function for its related interrupts.

Figure 12.20 shows an Example of Procedure for Changing Interrupt Sources.


Interrupt source change
Disable interrupts(2, 3)
Change interrupt source (including mode of peripheral function)
Set the IR bit to 0 (interrupt not requested) using the MOV instruction(3)
Enable interrupts(2, 3)
Change completed

IR bit: The interrupt control register bit of an interrupt whose source is changed.

NOTES:

1. Execute the above settings individually. Do not execute two or more settings at once (by one instruction).
2. Use the I flag for the $\overline{\text{INTi}}$ (i = 0 to 3) interrupts.
   To prevent interrupt requests from being generated when using peripheral function interrupts other than the $\overline{\text{INTi}}$ interrupt, disable the peripheral function before changing the interrupt source. In this case, use the I flag if all maskable interrupts can be disabled. If all maskable interrupts cannot be disabled, use bits ILVL0 to ILVL2 of the interrupt whose source is changed.
3. Refer to **12.5.6 Changing Interrupt Control Register Contents** for the instructions to be used and usage notes.

Figure 12.20 Example of Procedure for Changing Interrupt Sources

12.5.6 Changing Interrupt Control Register Contents

(a) The contents of an interrupt control register can only be changed while no interrupt requests corresponding to that register are generated. If interrupt requests may be generated, disable interrupts before changing the interrupt control register contents.

(b) When changing the contents of an interrupt control register after disabling interrupts, be careful to choose appropriate instructions.
Changing any bit other than IR bit
If an interrupt request corresponding to a register is generated while executing the instruction, the IR bit may not be set to 1 (interrupt requested), and the interrupt request may be ignored. If this causes a problem, use the following instructions to change the register: AND, OR, BCLR, BSET
Changing IR bit
If the IR bit is set to 0 (interrupt not requested), it may not be set to 0 depending on the instruction used. Therefore, use the MOV instruction to set the IR bit to 0.

(c) When disabling interrupts using the I flag, set the I flag as shown in the sample programs below. Refer to (b) regarding changing the contents of interrupt control registers by the sample programs.

Sample programs 1 to 3 are for preventing the I flag from being set to 1 (interrupts enabled) before the interrupt control register is changed for reasons of the internal bus or the instruction queue buffer.

Example 1: Use NOP instructions to prevent I flag from being set to 1 before interrupt control register is changed

```
INT_SWITCH1:
    FCLR    I               ; Disable interrupts
    AND.B   #00H,0056H      ; Set TXIC register to 00h
    NOP                     ;
    NOP
    FSET    I               ; Enable interrupts
```

Example 2: Use dummy read to delay FSET instruction

```
INT_SWITCH2:
    FCLR    I               ; Disable interrupts
    AND.B   #00H,0056H      ; Set TXIC register to 00h
    MOV.W   MEM,R0          ; Dummy read
    FSET    I               ; Enable interrupts
```

Example 3: Use POPC instruction to change I flag

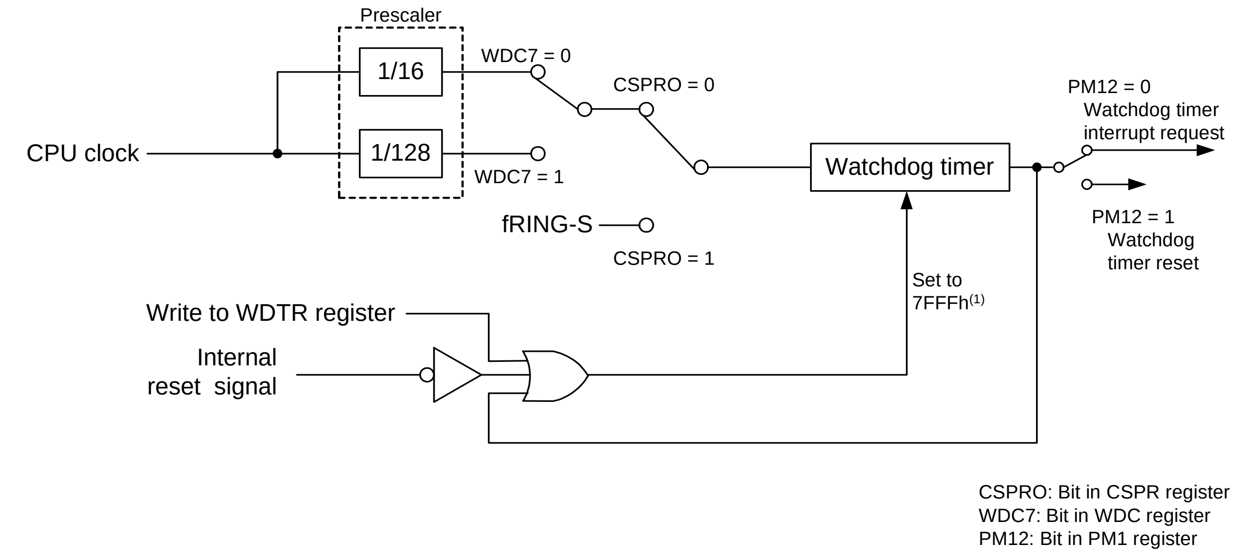
```
INT_SWITCH3:
    PUSHC   FLG
    FCLR    I               ; Disable interrupts
    AND.B   #00H,0056H      ; Set TXIC register to 00h
    POPC    FLG             ; Enable interrupts
```

13. Watchdog Timer

The watchdog timer is a function that detects when a program is out of control. Use of the watchdog timer is recommended to improve the reliability of the system. The watchdog timer contains a 15-bit counter and allows selection of count source protection mode enable or disable. Table 13.1 lists information on the Count Source Protection Mode.
Refer to **5.5 Watchdog Timer Reset** for details on the watchdog timer reset.
Figure 13.1 shows the Block Diagram of Watchdog Timer and Figures 13.2 to 13.3 show Registers OFS, WDC, WDTR, WDTS, and CSPR.

Table 13.1 Count Source Protection Mode

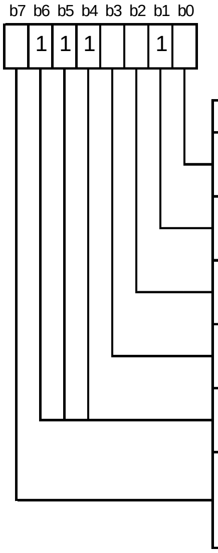
Item	Count Source Protection Mode Disabled	Count Source Protection Mode Enabled
Count source	CPU clock	Low-speed on-chip oscillator clock
Count operation	Decrement	
Reset condition of watchdog timer	• Reset • Write 00h to the WDTR register before writing FFh • underflow	
Count start condition	Either of the following can be selected • After reset, count starts automatically • Count starts by writing to WDTS register	
Count stop condition	Stop mode, wait mode	None
Operation at time of underflow	Watchdog timer interrupt or watchdog timer reset	Watchdog timer reset



NOTE:
1. When the CSPRO bit is set to 1 (count source protection mode enabled), 0FFFh is set.

Figure 13.1 Block Diagram of Watchdog Timer

Option Function Select Register[1]



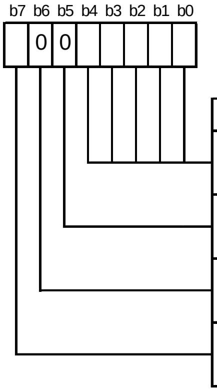
Symbol	Address	Before Shipment
OFS	0FFFFh	FFh[2]

Bit Symbol	Bit Name	Function	RW
WDTON	Watchdog timer start select bit	0 : Starts w atchdog timer automatically after reset. 1 : Watchdog timer is inactive after reset.	RW
— (b1)	Reserved bit	Set to 1.	RW
ROMCR	ROM code protect disabled bit	0 : ROM code protect disabled 1 : ROMCP1 enabled	RW
ROMCP1	ROM code protect bit	0 : ROM code protect enabled 1 : ROM code protect disabled	RW
— (b6-b4)	Reserved bits	Set to 1.	RW
CSPROINI	Count source protect mode after reset select bit	0 : Count source protect mode enabled after reset 1 : Count source protect mode disabled after reset	RW

NOTES:

1. The OFS register is on the flash memory. Write to the OFS register w ith a program.
2. If the block including the OFS register is erased, FFh is set to the OFS register.

Watchdog Timer Control Register

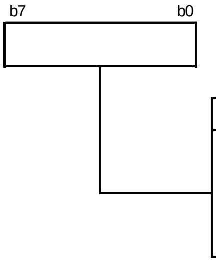


Symbol	Address	After Reset
WDC	000Fh	00011111b

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	High-order bits of w atchdog timer		RO
— (b5)	Reserved bit	Set to 0.	RW
— (b6)	Reserved bit	Set to 0.	RW
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	RW

Figure 13.2 Registers OFS and WDC

Watchdog Timer Reset Register

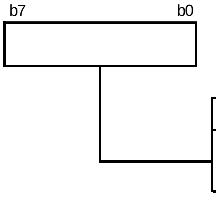


Symbol	Address	After Reset
WDTR	000Dh	Undefined

Function	RW
When 00h is w ritten before w riting FFh, the w atchdog timer is reset.[1] The default value of the w atchdog timer is 7FFFh w hen count source protection mode is disabled and 0FFFh w hen count source protection mode is enabled.[2]	WO

NOTES:
1. Do not generate an interrupt betw een w hen 00h and FFh are w ritten.
2. When the CSPRO bit in the CSPR register is set to 1 (count source protection mode enabled), 0FFFh is set in the w atchdog timer.

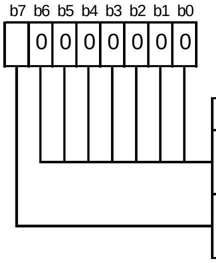
Watchdog Timer Start Register



Symbol	Address	After Reset
WDTS	000Eh	Undefined

Function	RW
The w atchdog timer starts counting after a w rite instruction to this register.	WO

Count Source Protection Mode Register



Symbol	Address	After Reset[1]
CSPR	001Ch	00h

Bit Symbol	Bit Name	Function	RW
— (b6-b0)	Reserved bits	Set to 0.	RW
CSPRO	Count source protection mode select bit[2]	0 : Count source protection mode disabled 1 : Count source protection mode enabled	RW

NOTES:
1. When 0 is w ritten to the CSPROINI bit in the OFS register, the value after reset is 10000000b.
2. Write 0 before w riting 1 to set the CSPRO bit to 1. 0 cannot be set by a program.

Figure 13.3 Registers WDTR, WDTS, and CSPR

13.1 Count Source Protection Mode Disabled

The count source of the watchdog timer is the CPU clock when count source protection mode is disabled. Table 13.2 lists the Watchdog Timer Specifications (with Count Source Protection Mode Disabled).

Table 13.2 Watchdog Timer Specifications (with Count Source Protection Mode Disabled)

Item	Specification
Count source	CPU clock
Count operation	Decrement
Period	$\frac{\text{Division ratio of prescaler (n)} \times \text{count value of watchdog timer (32768)}}{\text{CPU clock}}$ (1) n: 16 or 128 (selected by WDC7 bit in WDC register) Example: When the CPU clock frequency is 16 MHz and prescaler divides by 16, the period is approximately 32.8 ms.
Count start conditions	The WDTON bit(2) in the OFS register (0FFFFh) selects the operation of the watchdog timer after a reset. • When the WDTON bit is set to 1 (watchdog timer is in stop state after reset). The watchdog timer and prescaler stop after a reset and the count starts when the WDTS register is written to. • When the WDTON bit is set to 0 (watchdog timer starts automatically after exiting). The watchdog timer and prescaler start counting automatically after reset.
Reset condition of watchdog timer	• Reset • Write 00h to the WDTR register before writing FFh. • Underflow
Count stop condition	Stop and wait modes (inherit the count from the held value after exiting modes)
Operation at time of underflow	• When the PM12 bit in the PM1 register is set to 0. Watchdog timer interrupt • When the PM12 bit in the PM1 register is set to 1. Watchdog timer reset (Refer to **5.5 Watchdog Timer Reset**.)

NOTES:

1. The watchdog timer is reset when 00h is written to the WDTR register before FFh. The prescaler is reset after the MCU is reset. Some errors in the period of the watchdog timer may be caused by the prescaler.
2. The WDTON bit cannot be changed by a program. To set the WDTON bit, write 0 to bit 0 of address 0FFFFh with a flash programmer.

13.2 Count Source Protection Mode Enabled

The count source of the watchdog timer is the low-speed on-chip oscillator clock when count source protection mode is enabled. If the CPU clock stops when a program is out of control, the clock can still be supplied to the watchdog timer. Table 13.3 lists the Watchdog Timer Specifications (with Count Source Protection Mode Enabled).

Table 13.3 Watchdog Timer Specifications (with Count Source Protection Mode Enabled)

Item	Specification
Count source	Low-speed on-chip oscillator clock
Count operation	Decrement
Period	$\frac{\text{Count value of watchdog timer (4096)}}{\text{Low-speed on-chip oscillator clock}}$ Example: Period is approximately 32.8 ms when the low-speed on-chip oscillator clock frequency is 125 kHz
Count start conditions	The WDTON bit[1] in the OFS register (0FFFFh) selects the operation of the watchdog timer after a reset. • When the WDTON bit is set to 1 (watchdog timer is in stop state after reset). The watchdog timer and prescaler stop after a reset and the count starts when the WDTS register is written to. • When the WDTON bit is set to 0 (watchdog timer starts automatically after reset). The watchdog timer and prescaler start counting automatically after a reset.
Reset condition of watchdog timer	• Reset • Write 00h to the WDTR register before writing FFh. • Underflow
Count stop condition	None (The count does not stop in wait mode after the count starts. The MCU does not enter stop mode.)
Operation at time of underflow	Watchdog timer reset (Refer to **5.5 Watchdog Timer Reset**.)
Registers, bits	• When setting the CSPPRO bit in the CSPR register to 1 (count source protection mode is enabled)[2], the following are set automatically - Set 0FFFh to the watchdog timer - Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on) - Set the PM12 bit in the PM1 register to 1 (The watchdog timer is reset when watchdog timer underflows.) • The following conditions apply in count source protection mode - Writing to the CM10 bit in the CM1 register is disabled. (It remains unchanged even if it is set to 1. The MCU does not enter stop mode.) - Writing to the CM14 bit in the CM1 register is disabled. (It remains unchanged even if it is set to 1. The low-speed on-chip oscillator does not stop.)

NOTES:

1. The WDTON bit cannot be changed by a program. To set the WDTON bit, write 0 to bit 0 of address 0FFFFh with a flash programmer.
2. Even if 0 is written to the CSPROINI bit in the OFS register, the CSPRO bit is set to 1. The CSPROINI bit cannot be changed by a program. To set the CSPROINI bit, write 0 to bit 7 of address 0FFFFh with a flash programmer.

14. Timers

The MCU has two 8-bit timers with 8-bit prescalers, and a 16-bit timer. The two 8-bit timers with 8-bit prescalers are timer X and timer Z. These timers contain a reload register to store the default value of the counter. The 16-bit timer is timer C, and has input capture and output compare functions. All the timers operate independently. The count source for each timer is the operating clock that regulates the timing of timer operations such as counting and reloading.

Table 14.1 lists Functional Comparison of Timers.

Table 14.1 Functional Comparison of Timers

Item		Timer X	Timer Z	Timer C
Configuration		8-bit timer with 8-bit prescaler (with reload register)	8-bit timer with 8-bit prescaler (with reload register)	16-bit free-run timer (with input capture and output compare)
Count		Decrement	Decrement	Increment
Count sources		• f1 • f2 • f8 • fRING	• f1 • f2 • f8 • Timer X underflow	• f1 • f8 • f32 • fRING-fast
Function	Timer mode	Provided	Provided	Not provided
	Pulse output mode	Provided	Not provided	Not provided
	Event counter mode	Provided	Not provided	Not provided
	Pulse width measurement mode	Provided	Not provided	Not provided
	Pulse period measurement mode	Provided	Not provided	Not provided
	Programmable waveform generation mode	Not provided	Provided	Not provided
	Programmable one-shot generation mode	Not provided	Provided	Not provided
	Programmable wait one-shot generation mode	Not provided	Provided	Not provided
	Input capture mode	Not provided	Not provided	Provided
	Output compare mode	Not provided	Not provided	Provided
Input pin		CNTR0	$\overline{\text{INT0}}$	TCIN
Output pin		CNTR0 $\overline{\text{CNTR0}}$	TZOUT	CMP0_0 to CMP0_2 CMP1_0 to CMP1_2
Related interrupt		Timer X interrupt $\overline{\text{INT1}}$ interrupt	Timer Z interrupt $\overline{\text{INT0}}$ interrupt	Timer C interrupt $\overline{\text{INT3}}$ interrupt Compare 0 interrupt Compare 1 interrupt
Timer stop		Provided	Provided	Provided

14.1 Timer X

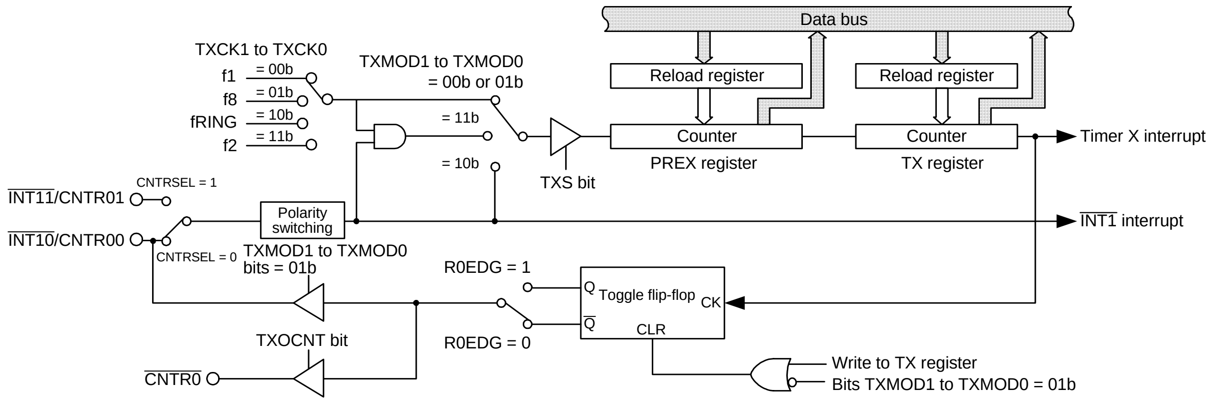
Timer X is an 8-bit timer with an 8-bit prescaler.

The prescaler and timer each consist of a reload register and counter. The reload register and counter are allocated at the same address, and can be accessed when accessing registers PREX and TX (refer to **Tables 14.2 to 14.6 the Specifications of Each Modes**).

Figure 14.1 shows a Block Diagram of Timer X. Figures 14.2 and 14.3 show the registers associated with Timer X.

Timer X has the following five operating modes:

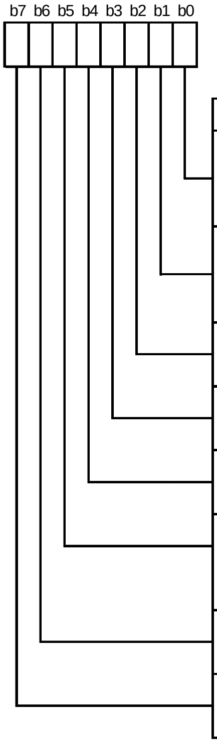
• Timer mode:	The timer counts the internal count source.
• Pulse output mode:	The timer counts the internal count source and outputs pulses which invert the polarity by underflow of the timer.
• Event counter mode:	The timer counts external pulses.
• Pulse width measurement mode:	The timer measures the pulse width of an external pulse.
• Pulse period measurement mode:	The timer measures the pulse period of an external pulse.



TXMOD0 to TXMOD1, R0EDG, TXS, TXOCNT: Bits in TXMR register
TXCK0 to TXCK1: Bits in TCSS register
CNTRSEL: Bit in UCON register

Figure 14.1 Block Diagram of Timer X

Timer X Mode Register



Symbol	Address	After Reset
TXMR	008Bh	00h

Bit Symbol	Bit Name	Function	RW
TXMOD0	Operating mode select bits 0, 1	b1 b0 0 0 : Timer mode or pulse period measurement mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse w idth measurement mode	RW
TXMOD1			RW
R0EDG	$\overline{\text{INT1}}$/CNTR0 signal polarity sw itch bit[1]	Function varies depending on operating mode.	RW
TXS	Timer X count start flag[2]	0 : Stops counting. 1 : Starts counting.	RW
TXOCNT	P3_7/$\overline{\text{CNTR0}}$ select bit	Function varies depending on operating mode.	RW
TXMOD2	Operating mode select bit 2	0 : Other than pulse period measurement mode 1 : Pulse period measurement mode	RW
TXEDG	Active edge judgment flag	Function varies depending on operating mode.	RW
TXUND	Timer X underflow flag	Function varies depending on operating mode.	RW

NOTES:

1. The IR bit in the INT1IC register may be set to 1 (requests interrupt) w hen the R0EDG bit is rew ritten. Refer to **12.5.5 Changing Interrupt Sources**.
2. Refer to **14.1.6 Notes on Timer X** for precautions regarding the TXS bit.

Figure 14.2 TXMR Register

Prescaler X Register

b7 … b0

Symbol	Address	After Reset
PREX	008Ch	FFh

Mode	Function	Setting Range	RW
Timer mode	Counts internal count source.	00h to FFh	RW
Pulse output mode	Counts internal count source.	00h to FFh	RW
Event counter mode	Counts input pulses from external clock.	00h to FFh	RW
Pulse w idth measurement mode	Measures pulse w idth of input pulses from external clock (counts internal count source).	00h to FFh	RW
Pulse period measurement mode	Measures pulse period of input pulses from external clock (counts internal count source).	00h to FFh	RW

Timer X Register

b7 … b0

Symbol	Address	After Reset
TX	008Dh	FFh

Function	Setting Range	RW
Counts underflow of prescaler X	00h to FFh	RW

Timer Count Source Setting Register

b7	b6	b5	b4	b3	b2	b1	b0
0	0			0	0		

Symbol	Address	After Reset
TCSS	008Eh	00h

Bit Symbol	Bit Name	Function	RW
TXCK0	Timer X count source select bits[1]	b1 b0 0 0 : f1 0 1 : f8 1 0 : fRING 1 1 : f2	RW
TXCK1			RW
— (b3-b2)	Reserved bits	Set to 0.	RW
TZCK0	Timer Z count source select bits[1]	b5 b4 0 0 : f1 0 1 : f8 1 0 : Selects timer X underflow . 1 1 : f2	RW
TZCK1			RW
— (b7-b6)	Reserved bits	Set to 0.	RW

NOTE:

1. Do not sw itch count sources during a count operation. Stop the timer count before sw itching count sources.

Figure 14.3 Registers PREX, TX, and TCSS

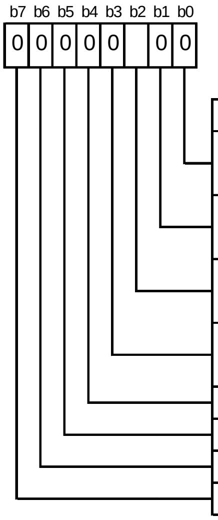
14.1.1 Timer Mode

Timer mode, the internally generated count source is counted (refer to **Table 14.2 Timer Mode Specifications**). Figure 14.4 shows the TXMR Register in Timer Mode.

Table 14.2 Timer Mode Specifications

Item	Specification
Count sources	f1, f2, f8, fRING
Count operations	• Decrement • When the timer underflows, the contents of the reload register are reloaded and the count is continued.
Divided ratio	1/(n+1)(m+1) n: value set in PREX register, m: value set in TX register
Count start condition	1 (count starts) is written to the TXS bit in the TXMR register.
Count stop condition	0 (count stops) is written to the TXS bit in the TXMR register.
Interrupt request generation timing	When timer X underflows [timer X interrupt].
$\overline{\text{INT10}}$/CNTR00, $\overline{\text{INT11}}$/CNTR01 pin functions	Programmable I/O port, or $\overline{\text{INT1}}$ interrupt input
$\overline{\text{CNTR0}}$ pin function	Programmable I/O port
Read from timer	The count value can be read out by reading registers TX and PREX.
Write to timer	• When registers TX and PREX are written while the count is stopped, values are written to both the reload register and counter. • When registers TX and PREX are written during the count, the value is written to each reload register of registers TX and PREX at the following count source input, the data is transferred to the counter at the second count source input, and the count re-starts at the third count source input.

Timer X Mode Register



Symbol	Address	After Reset
TXMR	008Bh	00h

Bit Symbol	Bit Name	Function	RW
TXMOD0	Operating mode select bits 0, 1	b1 b0 0 0 : Timer mode or pulse period measurement mode	RW
TXMOD1			RW
R0EDG	$\overline{\text{INT1}}$/CNTR0 signal polarity switch bit[1, 2]	0 : Rising edge 1 : Falling edge	RW
TXS	Timer X count start flag[3]	0 : Stops counting. 1 : Starts counting.	RW
TXOCNT	Set to 0 in timer mode.		RW
TXMOD2	Operating mode select bit 2	0 : Other than pulse period measurement mode	RW
TXEDG	Set to 0 in timer mode.		RW
TXUND	Set to 0 in timer mode.		RW

NOTES:

1. The IR bit in the INT1IC register may be set to 1 (requests interrupt) when the R0EDG bit is rewritten. Refer to **12.5.5 Changing Interrupt Sources**.
2. This bit is used to select the polarity of $\overline{\text{INT1}}$ interrupt in timer mode.
3. Refer to **14.1.6 Notes on Timer X** for precautions regarding the TXS bit.

Figure 14.4 TXMR Register in Timer Mode

14.1.2 Pulse Output Mode

In pulse output mode, the internally generated count source is counted, and a pulse with inverted polarity is output from the CNTR0 pin each time the timer underflows (refer to **Table 14.3 Pulse Output Mode Specifications**). Figure 14.5 shows the TXMR Register in Pulse Output Mode.

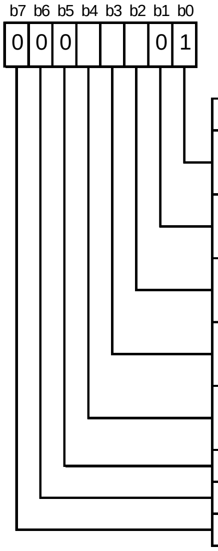
Table 14.3 Pulse Output Mode Specifications

Item	Specification
Count sources	f1, f2, f8, fRING
Count operations	• Decrement • When the timer underflows, the contents of the reload register are reloaded and the count is continued.
Divided ratio	1/(n+1)(m+1) n: value set in PREX register, m: value set in TX register
Count start condition	1 (count starts) is written to the TXS bit in the TXMR register.
Count stop condition	0 (count stops) is written to the TXS bit in the TXMR register.
Interrupt request generation timing	When timer X underflows [timer X interrupt].
$\overline{\text{INT1}}$0/CNTR00 pin function	Pulse output
$\overline{\text{CNTR0}}$ pin function	Programmable I/O port, or inverted output of CNTR0
Read from timer	The count value can be read out by reading registers TX and PREX.
Write to timer	• When registers TX and PREX are written while the count is stopped, values are written to both the reload register and counter. • When registers TX and PREX are written during the count, the value is written to each reload register of registers TX and PREX at the following count source input, the data is transferred to the counter at the second count source input, and the count re-starts at the third count source input.
Select functions	• $\overline{\text{INT1}}$/CNTR0 signal polarity switch function The R0EDG bit can select the polarity level when the pulse output starts.(1) • Inverted pulse output function The pulse which inverts the polarity of the CNTR0 output can be output from the $\overline{\text{CNTR0}}$ pin (selected by TXOCNT bit).

NOTE:

1. The level of the output pulse becomes the level when the pulse output starts when the TX register is written to.

Timer X Mode Register



Symbol	Address	After Reset
TXMR	008Bh	00h

Bit Symbol	Bit Name	Function	RW
TXMOD0	Operating mode select bits 0, 1	b1 b0 0 1 : Pulse output mode	RW
TXMOD1			RW
R0EDG	$\overline{\text{INT1}}$/CNTR0 signal polarity switch bit[1]	0 : CNTR0 signal output starts at “H”. 1 : CNTR0 signal output starts at “L”.	RW
TXS	Timer X count start flag[2]	0 : Stops counting. 1 : Starts counting.	RW
TXOCNT	P3_7/$\overline{\text{CNTR0}}$ select bit	0 : Port P3_7 1 : $\overline{\text{CNTR0}}$ output	RW
TXMOD2	Set to 0 in pulse output mode.		RW
TXEDG	Set to 0 in pulse output mode.		RW
TXUND	Set to 0 in pulse output mode.		RW

NOTES:

1. The IR bit in the INT1IC register may be set to 1 (requests interrupt) when the R0EDG bit is rewritten. Refer to **12.5.5 Changing Interrupt Sources**.
2. Refer to **14.1.6 Notes on Timer X** for precautions regarding the TXS bit.

Figure 14.5 TXMR Register in Pulse Output Mode

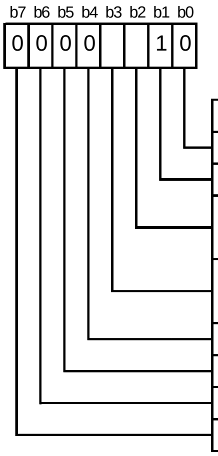
14.1.3 Event Counter Mode

In event counter mode, external signal inputs to the $\overline{\text{INT1}}$/CNTR0 pin are counted (refer to **Table 14.4 Event Counter Mode Specifications**). Figure 14.6 shows the TXMR Register in Event Counter Mode.

Table 14.4 Event Counter Mode Specifications

Item	Specification
Count source	External signal which is input to CNTR0 pin (Active edge is selectable by software)
Count operations	• Decrement • When the timer underflows, the contents of the reload register are reloaded and the count is continued.
Divided ratio	1/(n+1)(m+1) n: value set in PREX register, m: value set in TX register
Count start condition	1 (count starts) is written to the TXS bit in the TXMR register.
Count stop condition	0 (count stops) is written to the TXS bit in the TXMR register.
Interrupt request generation timing	When timer X underflows [timer X interrupt].
$\overline{\text{INT10}}$/CNTR00, INT11/CNTR01 pin functions	Count source input ($\overline{\text{INT1}}$ interrupt input)
$\overline{\text{CNTR0}}$ pin function	Programmable I/O port
Read from timer	The count value can be read out by reading registers TX and PREX.
Write to timer	• When registers TX and PREX are written while the count is stopped, values are written to both the reload register and counter. • When registers TX and PREX are written during the count, the value is written to each reload register of registers TX and PREX at the following count source input, the data is transferred to the counter at the second count source input, and the count re-starts at the third count source input.
Select functions	• $\overline{\text{INT1}}$/CNTR0 signal polarity switch function The R0EDG bit can select the active edge of the count source. • Count source input pin select function The CNTRSEL bit in the UCON register can select the CNTR00 or CNTR01 pin.

Timer X Mode Register



Symbol	Address	After Reset
TXMR	008Bh	00h

Bit Symbol	Bit Name	Function	RW
TXMOD0	Operating mode select bits 0, 1	b1 b0 1 0 : Event counter mode	RW
TXMOD1			RW
R0EDG	$\overline{\text{INT1}}$/CNTR0 signal polarity sw itch bit[1]	0 : Rising edge 1 : Falling edge	RW
TXS	Timer X count start flag[2]	0 : Stops counting. 1 : Starts counting.	RW
TXOCNT	Set to 0 in event counter mode.		RW
TXMOD2	Set to 0 in event counter mode.		RW
TXEDG	Set to 0 in event counter mode.		RW
TXUND	Set to 0 in event counter mode.		RW

NOTES:

1. The IR bit in the INT1IC register may be set to 1 (requests interrupt) w hen the R0EDG bit is rew ritten. Refer to **12.5.5 Changing Interrupt Sources**.
2. Refer to **14.1.6 Notes on Timer X** for precautions regarding the TXS bit.

Figure 14.6 TXMR Register in Event Counter Mode

14.1.4 Pulse Width Measurement Mode

In pulse width measurement mode, the pulse width of an external signal input to the $\overline{\text{INT1}}$/CNTR0 pin is measured (refer to **Table 14.5 Pulse Width Measurement Mode Specifications**). Figure 14.7 shows the TXMR Register in Pulse Width Measurement Mode. Figure 14.8 shows an Operating Example in Pulse Width Measurement Mode.

Table 14.5 Pulse Width Measurement Mode Specifications

Item	Specification
Count sources	f1, f2, f8, fRING
Count operations	• Decrement • Continuously counts the selected signal only when the measured pulse is “H” level, or conversely only “L” level. • When the timer underflows, the contents of the reload register are reloaded and the count is continued.
Count start condition	1 (count starts) is written to the TXS bit in the TXMR register.
Count stop condition	0 (count stops) is written to the TXS bit in the TXMR register.
Interrupt request generation timing	• When timer X underflows [timer X interrupt]. • Rising or falling of the CNTR0 input (end of measurement period) [$\overline{\text{INT1}}$ interrupt]
$\overline{\text{INT10}}$/CNTR00, $\overline{\text{INT11}}$/CNTR01 pin functions	Measured pulse input ($\overline{\text{INT1}}$ interrupt input)
$\overline{\text{CNTR0}}$ pin function	Programmable I/O port
Read from timer	The count value can be read out by reading registers TX and PREX.
Write to timer	• When registers TX and PREX are written while the count is stopped, values are written to both the reload register and counter. • When registers TX and PREX are written during the count, the value is written to each reload register of registers TX and PREX at the following count source input, the data is transferred to the counter at the second count source input, and the count re-starts at the third count source input.
Select functions	• $\overline{\text{INT1}}$/CNTR0 signal polarity switch function The R0EDG bit can select “H” or “L” level period for the input pulse width measurement. • Measured pulse input pin select function The CNTRSEL bit in the UCON register can select the CNTR00 or CNTR01 pin.

Timer X Mode Register

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0			1	1

Symbol	Address	After Reset
TXMR	008Bh	00h

Bit Symbol	Bit Name	Function	RW
TXMOD0	Operating mode select bits 0, 1	b1 b0 1 1 : Pulse w idth measurement mode	RW
TXMOD1			RW
R0EDG	$\overline{\text{INT1}}$/CNTR0 signal polarity sw itch bit(1)	[CNTR0] 0 : Measures “L” level w idth 1 : Measures “H” level w idth [$\overline{\text{INT1}}$] 0 : Rising edge 1 : Falling edge	RW
TXS	Timer X count start flag(2)	0 : Stops counting. 1 : Starts counting.	RW
TXOCNT	Set to 0 in pulse w idth measurement mode.		RW
TXMOD2	Set to 0 in pulse w idth measurement mode.		RW
TXEDG	Set to 0 in pulse w idth measurement mode.		RW
TXUND	Set to 0 in pulse w idth measurement mode.		RW

NOTES:

1. The IR bit in the INT1IC register may be set to 1 (requests interrupt) w hen the R0EDG bit is rew ritten. Refer to **12.5.5 Changing Interrupt Sources**.
2. Refer to **14.1.6 Notes on Timer X** for precautions regarding the TXS bit.

Figure 14.7 TXMR Register in Pulse Width Measurement Mode

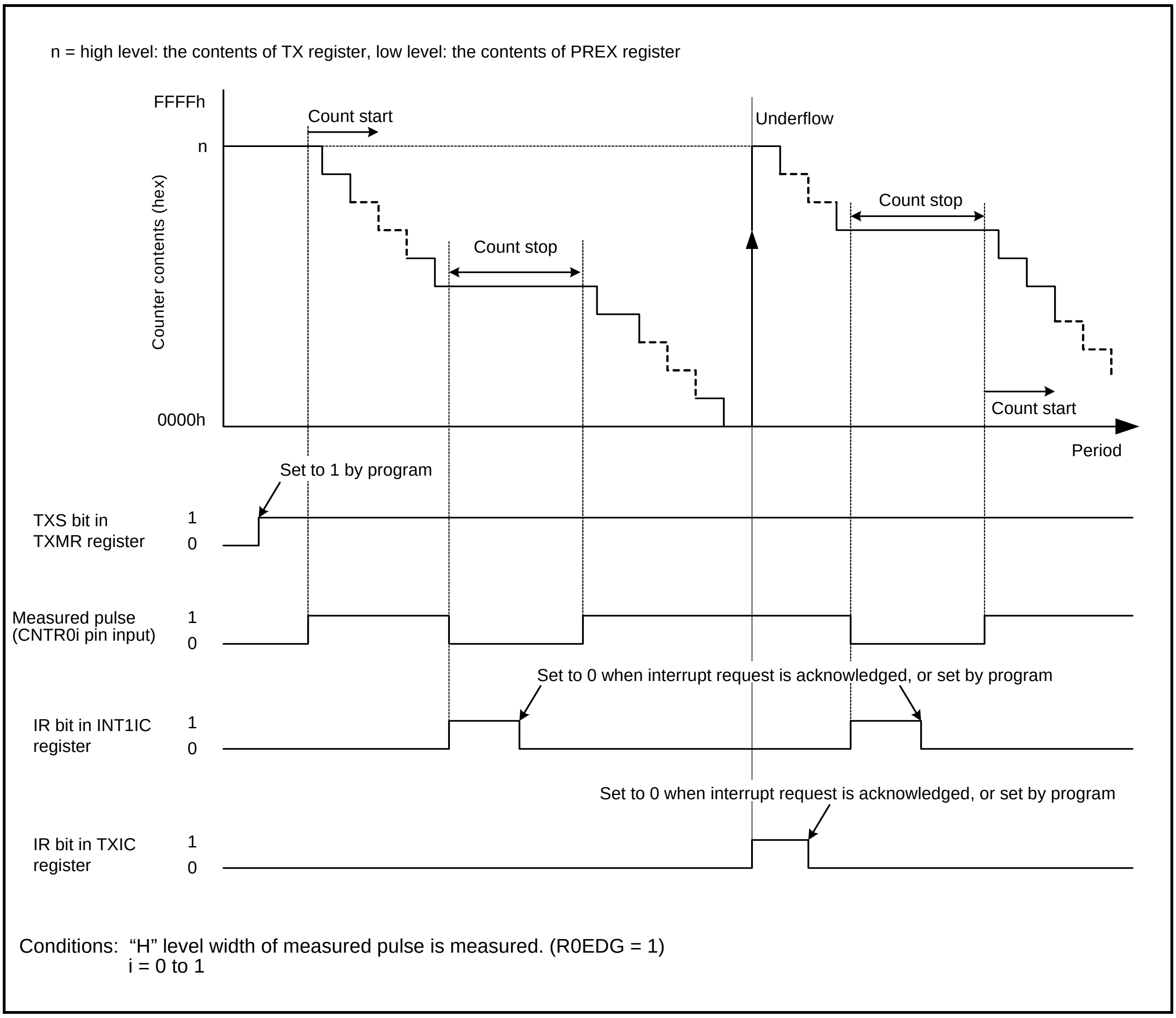

n = high level: the contents of TX register, low level: the contents of PREX register
FFFFh
n
0000h
Counter contents (hex)
Count start
Underflow
Count stop
Count stop
Count start
Period
Set to 1 by program
TXS bit in TXMR register
1
0
Measured pulse (CNTR0i pin input)
1
0
Set to 0 when interrupt request is acknowledged, or set by program
IR bit in INT1IC register
1
0
Set to 0 when interrupt request is acknowledged, or set by program
IR bit in TXIC register
1
0
Conditions: “H” level width of measured pulse is measured. (R0EDG = 1)
i = 0 to 1

Figure 14.8 Operating Example in Pulse Width Measurement Mode

14.1.5 Pulse Period Measurement Mode

In pulse period measurement mode, the pulse period of an external signal input to the $\overline{\mathrm{INT1}}$/CNTR0 pin is measured (refer to **Table 14.6 Pulse Period Measurement Mode Specifications**). Figure 14.9 shows the TXMR Register in Pulse Period Measurement Mode. Figure 14.10 shows an Operating Example in Pulse Period Measurement Mode.

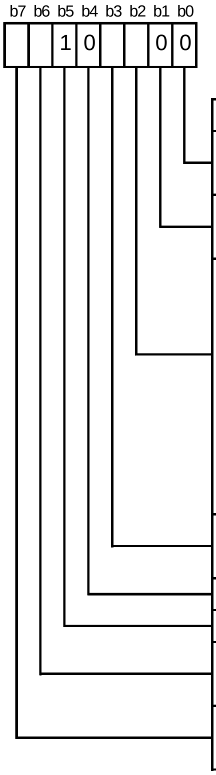
Table 14.6 Pulse Period Measurement Mode Specifications

Item	Specification
Count sources	f1, f2, f8, fRING
Count operations	• Decrement • After an active edge of the measured pulse is input, contents for the read-out buffer are retained at the first underflow of prescaler X. Then timer X reloads contents in the reload register at the second underflow of prescaler X and continues counting.
Count start condition	1 (count starts) is written to the TXS bit in the TXMR register.
Count stop condition	0 (count stops) is written to the TXS bit in the TXMR register.
Interrupt request generation timing	• When timer X underflows or reloads [timer X interrupt]. • Rising or falling of CNTR0 input (end of measurement period) [$\overline{\mathrm{INT1}}$ interrupt]
$\overline{\mathrm{INT10}}$/CNTR00, $\overline{\mathrm{INT11}}$/CNTR01 pin functions	Measured pulse input[1] ($\overline{\mathrm{INT1}}$ interrupt input)
$\overline{\mathrm{CNTR0}}$ pin function	Programmable I/O port
Read from timer	Contents of the read-out buffer can be read out by reading the TX register. The value retained in the read-out buffer is released by reading the TX register.
Write to timer	• When registers TX and PREX are written while the count is stopped, values are written to both the reload register and counter. • When registers TX and PREX are written during the count, the value is written to each reload register of registers TX and PREX at the following count source input, the data is transferred to the counter at the second count source input, and the count re-starts at the third count source input.
Select functions	• $\overline{\mathrm{INT1}}$/CNTR0 polarity switch function The R0EDG bit can select the measurement period for the input pulse. • Measured pulse input pin select function The CNTRSEL bit in the UCON register can select the CNTR00 or CNTR01 pin.

NOTE:

1. Input a pulse with a period longer than twice of the prescaler X period. Input a pulse with a longer "H" and "L" width than the prescaler X period. If a pulse with a shorter period is input to the CNTR0 pin, the input may be ignored.

Timer X Mode Register



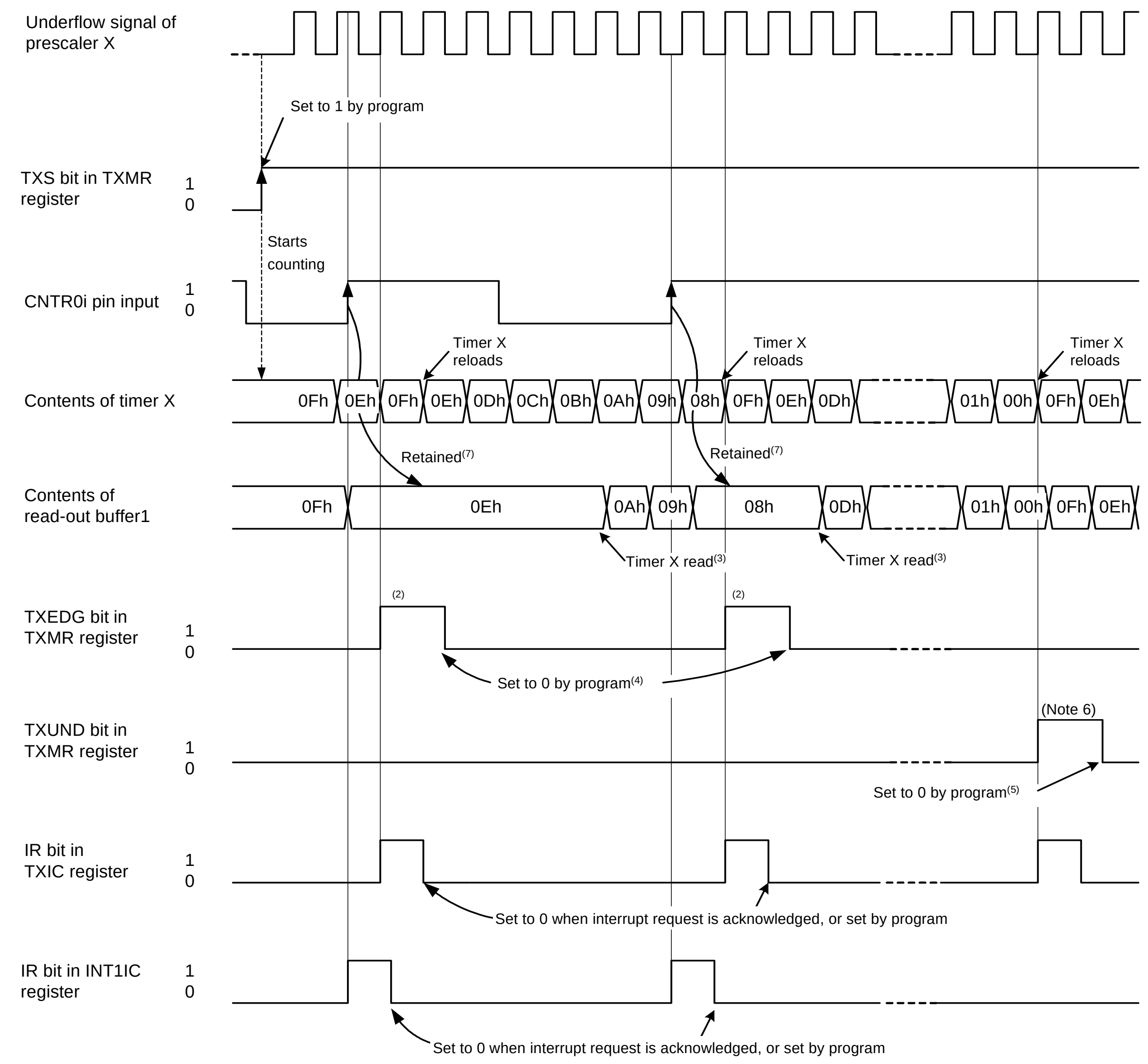
Symbol	Address	After Reset
TXMR	008Bh	00h

Bit Symbol	Bit Name	Function	RW
TXMOD0	Operating mode select bits 0, 1	b1 b0 0 0 : Timer mode or pulse period measurement mode	RW
TXMOD1			RW
R0EDG	$\overline{\text{INT1}}$/CNTR0 signal polarity switch bit[1]	[CNTR0] 0 : Measures measured pulse from one rising edge to next rising edge. 1 : Measures measured pulse from one falling edge to next falling edge. [$\overline{\text{INT1}}$] 0 : Rising edge 1 : Falling edge	RW
TXS	Timer X count start flag[3]	0 : Stops counting. 1 : Starts counting.	RW
TXOCNT	Set to 0 in pulse width measurement mode.		RW
TXMOD2	Operating mode select bit 2	1 : Pulse period measurement mode	RW
TXEDG[2]	Active edge judgment flag	0 : Active edge not received 1 : Active edge received	RW
TXUND[2]	Timer X underflow flag	0 : No underflow 1 : Underflow	RW

NOTES:

1. The IR bit in the INT1IC register may be set to 1 (requests interrupt) when the R0EDG bit is rewritten. Refer to **12.5.5 Changing Interrupt Sources**.
2. This bit is set to 0 by writing 0 in a program (and remains unchanged even if 1 is written to it).
3. Refer to **14.1.6 Notes on Timer X** for precautions regarding the TXS bit.

Figure 14.9 TXMR Register in Pulse Period Measurement Mode



Conditions: The period from one rising edge to the next rising edge of the measured pulse is measured (R0EDG = 0) with the default value of the TX register as 0Fh.
i = 0 to 1

NOTES:
1. The contents of the read-out buffer can be read by reading the TX register in pulse period measurement mode.
2. After an active of edge the measured pulse is input, the TXEDG bit in the TXMR register is set to 1 (active edge found) when the prescale X underflows for the second time.
3. The TX register should be read before the next active edge is input after the TXEDG bit is set to 1 (active edge found). The contents in the read-out buffer are retained until the TX register is read. If the TX register is not read before the next active edge is input, the measured result of the previous period is retained.
4. To set to 0 by a program, use a MOV instruction to write 0 to the TXEDG in the TXMR register. At the same time, write 1 to the TXUND bit.
5. To set to 0 by a program, use a MOV instruction to write 0 to the TXUND in the TXMR register. At the same time, write 1 to the TXEDG bit.
6. Bits TXUND and TXEDG are both set to 1 if timer X underflows and reloads on an active edge simultaneously. In this case, the validity of the TXUND bit should be determined by the contents of the read-out buffer.
7. If the prescaler X underflow signal is “H” level when the CNTR0 active edge is input, the value in the read buffer is the count value at that point. If the prescaler X underflow signal is “L” level when the CNTR0 active edge is input, the value in the read buffer is the next count value.

Figure 14.10 Operating Example in Pulse Period Measurement Mode

14.1.6 Notes on Timer X

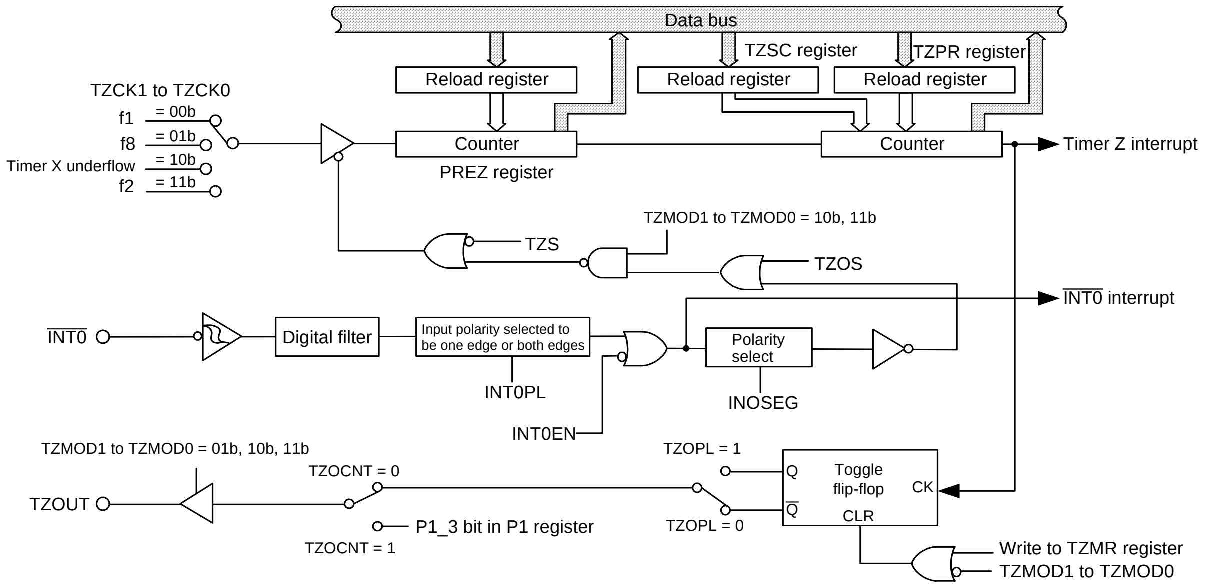
- Timer X stops counting after a reset. Set the values in the timer and prescaler before the count starts.
- Even if the prescaler and timer are read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- Do not rewrite bits TXMOD0 to TXMOD1, and bits TXMOD2 and TXS simultaneously.
- In pulse period measurement mode, bits TXEDG and TXUND in the TXMR register can be set to 0 by writing 0 to these bits by a program. However, these bits remain unchanged if 1 is written. When using the READ-MODIFY-WRITE instruction for the TXMR register, the TXEDG or TXUND bit may be set to 0 although these bits are set to 1 while the instruction is being executed. In this case, write 1 to the TXEDG or TXUND bit which is not supposed to be set to 0 with the MOV instruction.
- When changing to pulse period measurement mode from another mode, the contents of bits TXEDG and TXUND are undefined. Write 0 to bits TXEDG and TXUND before the count starts.
- The TXEDG bit may be set to 1 by the prescaler X underflow generated after the count starts.
- When using the pulse period measurement mode, leave two or more periods of the prescaler X immediately after the count starts, then set the TXEDG bit to 0.
- The TXS bit in the TXMR register has a function to instruct timer X to start or stop counting and a function to indicate that the count has started or stopped.
  0 (count stops) can be read until the following count source is applied after 1 (count starts) is written to the TXS bit while the count is being stopped. If the following count source is applied, 1 can be read from the TXS bit. After writing 1 to the TXS bit, do not access registers associated with timer X (registers TXMR, PREX, TX, TCSS, and TXIC) except for the TXS bit, until 1 can be read from the TXS bit. The count starts at the following count source after the TXS bit is set to 1.
  Also, after writing 0 (count stops) to the TXS bit during the count, timer X stops counting at the following count source.
  1 (count starts) can be read by reading the TXS bit until the count stops after writing 0 to the TXS bit. After writing 0 to the TXS bit, do not access registers associated with timer X except for the TXS bit, until 0 can be read from the TXS bit.

14.2 Timer Z

Timer Z is an 8-bit timer with an 8-bit prescaler. The prescaler and timer each consist of a reload register and counter. The reload register and counter are allocated at the same address. Refer to the **Tables 14.7 to 14.10 for the Specifications of Each Mode**. Timer Z contains timer Z primary and timer Z secondary reload registers.
Figure 14.11 shows a Block Diagram of Timer Z. Figures 14.12 to 14.15 show registers TZMR, PREZ, TZSC, TZPR, TZOC, PUM, and TCSS.

Timer Z has the following four operating modes:

• Timer mode:	The timer counts an internal count source or timer X underflows.
• Programmable waveform generation mode:	The timer outputs pulses of a given width successively.
• Programmable one-shot generation mode:	The timer outputs a one-shot pulse.
• Programmable wait one-shot generation mode:	The timer outputs a delayed one-shot pulse.

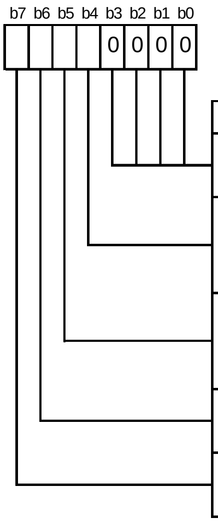


TZMOD0 to TZMOD1, TZS: Bits in TZMR register
TZOS, TZOCNT: Bits in TZOC register

TZOPL, INOSTG: Bits in PUM register
TZCK0 to TZCK1: Bits in TCSS register
INT0EN, INT0PL: Bits in INTEN register

Figure 14.11 Block Diagram of Timer Z

Timer Z Mode Register



Symbol	Address	After Reset
TZMR	0080h	00h

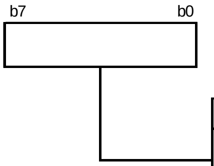
Bit Symbol	Bit Name	Function	RW
— (b3-b0)	Reserved bits	Set to 0.	RW
TZMOD0	Timer Z operating mode bits	b5 b4 0 0 : Timer mode 0 1 : Programmable w aveform generation mode 1 0 : Programmable one-shot generation mode 1 1 : Programmable w ait one-shot generation mode	RW
TZMOD1			RW
TZWC	Timer Z w rite control bit	Functions varies depending on operating mode.	RW
TZS	Timer Z count start flag[1]	0 : Stops counting. 1 : Starts counting.	RW

NOTE:

1. Refer to **14.2.5 Notes on Timer Z** for precautions regarding the TZS bit.

Figure 14.12 TZMR Register

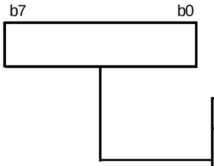
Prescaler Z Register



Symbol	Address	After Reset
PREZ	0085h	FFh

Mode	Function	Setting Range	RW
Timer mode	Counts internal count source or timer X underflows.	00h to FFh	RW
Programmable waveform generation mode	Counts internal count source or timer X underflows.	00h to FFh	RW
Programmable one-shot generation mode	Counts internal count source or timer X underflows.	00h to FFh	RW
Programmable wait one-shot generation mode	Counts internal count source or timer X underflows.	00h to FFh	RW

Timer Z Secondary Register



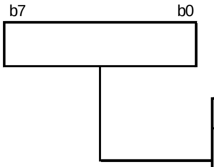
Symbol	Address	After Reset
TZSC	0086h	FFh

Mode	Function	Setting Range	RW
Timer mode	Disabled	—	—
Programmable waveform generation mode	Counts underflow of prescaler Z.[1]	00h to FFh	WO[2]
Programmable one-shot generation mode	Disabled	—	—
Programmable wait one-shot generation mode	Counts underflows of prescaler Z (counts one-shot width).	00h to FFh	WO

NOTES:

1. Each value in the TZPR register and TZSC register is reloaded to the counter alternately and counted.
2. The count value can be read out by reading the TZPR register even when the secondary period is being counted.

Timer Z Primary Register



Symbol	Address	After Reset
TZPR	0087h	FFh

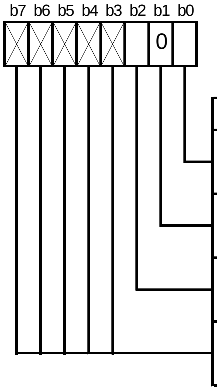
Mode	Function	Setting Range	RW
Timer mode	Counts underflows of prescaler Z.	00h to FFh	RW
Programmable waveform generation mode	Counts underflows of prescaler Z.[1]	00h to FFh	RW
Programmable one-shot generation mode	Counts underflows of prescaler Z (counts one-shot width).	00h to FFh	RW
Programmable wait one-shot generation mode	Counts underflows of prescaler Z (counts wait period).	00h to FFh	RW

NOTE:

1. Each value in registers TZPR and TZSC is reloaded to the counter alternately and counted.

Figure 14.13 Registers PREZ, TZSC, and TZPR

Timer Z Output Control Register[3]



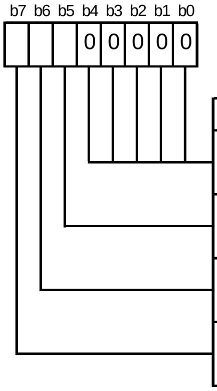
Symbol	Address	After Reset
TZOC	008Ah	00h

Bit Symbol	Bit Name	Function	RW
TZOS	Timer Z one-shot start bit[1]	0 : One-shot stops. 1 : One-shot starts.	RW
— (b1)	Reserved bit	Set to 0.	RW
TZOCNT	Timer Z programmable waveform generation output switch bit[2]	0 : Outputs programmable waveform. 1 : Outputs value in P1_3 port register.	RW
— (b7-b3)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		—

NOTES:

1. This bit is set to 0 when the output of one-shot waveform is completed. If the TZS bit in the TZMR register was set to 0 (count stops) to stop the waveform output during one-shot waveform output, set the TZOS bit to 0.
2. This bit is enabled only when operating in programmable waveform generation mode.
3. When executing an instruction which changes this register when the TZOS bit is set to 1 (during count), the TZOS bit is automatically set to 0 (one-shot stop) if the count is completed while the instruction is being executed. If this causes problems, execute an instruction which changes the contents of this register when the TZOS bit is set to 0 (one-shot stop).

Timer Z Waveform Output Control Register



Symbol	Address	After Reset
PUM	0084h	00h

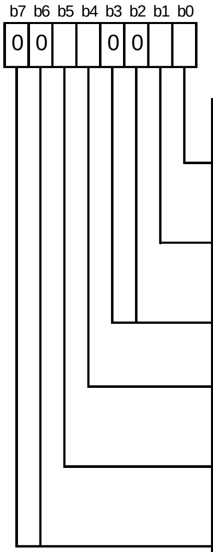
Bit Symbol	Bit Name	Function	RW
— (b4-b0)	Reserved bits	Set to 0.	RW
TZOPL	Timer Z output level latch	Function varies depending on operating mode.	RW
INOSTG	$\overline{\text{INT0}}$ pin one-shot trigger control bit (timer Z)[2]	0 : $\overline{\text{INT0}}$ pin one-shot trigger disabled 1 : $\overline{\text{INT0}}$ pin one-shot trigger enabled	RW
INOSEG	$\overline{\text{INT0}}$ pin one-shot trigger polarity select bit (timer Z)[1]	0 : Falling edge trigger 1 : Rising edge trigger	RW

NOTES:

1. The INOSEG bit is enabled only when the INT0PL bit in the INTEN register is set to 0 (one edge).
2. Set the INOSTG bit to 1 after setting the INT0EN bit in the INTEN register and the INOSEG bit in the PUM register.

Figure 14.14 Registers TZOC and PUM

Timer Count Source Setting Register



Symbol	Address	After Reset
TCSS	008Eh	00h

Bit Symbol	Bit Name	Function	RW
TXCK0	Timer X count source select bits[1]	b1 b0 0 0 : f1 0 1 : f8 1 0 : fRING 1 1 : f2	RW
TXCK1			RW
— (b3-b2)	Reserved bits	Set to 0.	RW
TZCK0	Timer Z count source select bits[1]	b5 b4 0 0 : f1 0 1 : f8 1 0 : Selects Timer X underflow . 1 1 : f2	RW
TZCK1			RW
— (b7-b6)	Reserved bits	Set to 0.	RW

NOTE:

1. Do not sw itch count sources during a count operation. Stop the timer count before sw itching count sources.

Figure 14.15 TCSS Register

14.2.1 Timer Mode

In timer mode, a count source which is internally generated or timer X underflow is counted (refer to **Table 14.7 Timer Mode Specifications**). The TZSC register is not used in timer mode. Figure 14.16 shows Registers TZMR and PUM in Timer Mode.

Table 14.7 Timer Mode Specifications

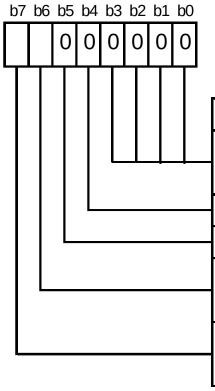
Item	Specification
Count sources	f1, f2, f8, Timer X underflow
Count operations	• Decrement • When the timer underflows, it reloads the reload register contents before the count continues. (When timer Z underflows, the contents of timer Z primary reload register is reloaded.)
Divided ratio	1/(n+1)(m+1) fi: Count source frequency n: Value set in PREZ register, m: value set in TZPR register
Count start condition	1 (count starts) is written to the TZS bit in the TZMR register.
Count stop condition	0 (count stops) is written to the TZS bit in the TZMR register.
Interrupt request generation timing	• When timer Z underflows [timer Z interrupt].
TZOUT pin function	Programmable I/O port
$\overline{\text{INT0}}$ pin function	Programmable I/O port, or $\overline{\text{INT0}}$ interrupt input
Read from timer	The count value can be read out by reading registers TZPR and PREZ.
Write to timer(1)	• When registers TZPR and PREZ are written while the count is stopped, values are written to both the reload register and counter. • When registers TZPR and PREZ are written during the count while the TZWC bit is set to 0 (writing to the reload register and counter simultaneously), the value is written to each reload register of registers TZPR and PREZ at the following count source input, the data is transferred to the counter at the second count source input, and the count re-starts at the third count source input. When the TZWC bit is set to 1 (writing to only the reload register), the value is written to each reload register of registers TZPR and PREZ (the data is transferred to the counter at the following reload).

NOTE:

1. The IR bit in the TZIC register is set to 1 (interrupt requested) when writing to the TZPR or PREZ register while both of the following conditions are met.
   - TZWC bit in TZMR register is set to 0 (write to reload register and counter simultaneously)
   - TZS bit in TZMR register is set to 1 (count starts)

   Disable interrupts before writing to the TZPR or PREZ register in the above state.

Timer Z Mode Register



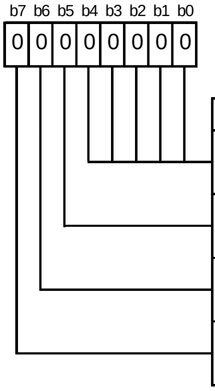
Symbol	Address	After Reset
TZMR	0080h	00h

Bit Symbol	Bit Name	Function	RW
— (b3-b0)	Reserved bits	Set to 0.	RW
TZMOD0	Timer Z operating mode bits	b5 b4 0 0 : Timer mode	RW
TZMOD1			RW
TZWC	Timer Z w rite control bit[(1)]	0 : Write to reload register and counter 1 : Write to reload register only	RW
TZS	Timer Z count start flag[(2)]	0 : Stops counting. 1 : Starts counting.	RW

NOTES:

1. When the TZS bit is set to 1 (count starts), the setting value in the TZWC bit is enabled. When the TZWC bit is set to 0, timer Z count value is w ritten to both reload register and counter. Timer Z count value is w ritten to the reload register only w hen the TZWC bit is set to 1. When the TZS bit is set to 0 (count stops), timer Z count value is w ritten to both reload register and counter regardless of the setting value of the TZWC bit.
2. Refer to **14.2.5 Notes on Timer Z** for precautions regarding the TZS bit.

Timer Z Waveform Output Control Register



Symbol	Address	After Reset
PUM	0084h	00h

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	Reserved bits	Set to 0.	RW
TZOPL	Timer Z output level latch	Set to 0 in timer mode.	RW
INOSTG	$\overline{\text{INT0}}$ pin one-shot trigger control bit	Set to 0 in timer mode.	RW
INOSEG	$\overline{\text{INT0}}$ pin one-shot trigger polarity select bit	Set to 0 in timer mode.	RW

Figure 14.16 Registers TZMR and PUM in Timer Mode

14.2.2 Programmable Waveform Generation Mode

In programmable waveform generation mode, the signal output from the TZOUT pin is inverted each time the counter underflows, while the values in registers TZPR and TZSC are counted alternately (refer to **Table 14.8 Programmable Waveform Generation Mode Specifications**). Counting starts by counting the value set in the TZPR register. Figure 14.17 shows Registers TZMR and PUM in Programmable Waveform Generation Mode. Figure 14.18 shows an Operating Example of Timer Z in Programmable Waveform Generation Mode.

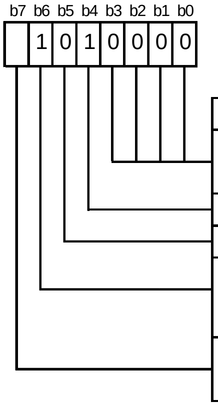
Table 14.8 Programmable Waveform Generation Mode Specifications

Item	Specification
Count sources	f1, f2, f8, timer X underflow
Count operations	• Decrement • When the timer underflows, it reloads the contents of the primary reload and secondary reload registers alternately before the count is continued.
Width and period of output waveform	Primary period: (n+1)(m+1)/fi Secondary period: (n+1)(p+1)/fi Period: (n+1){(m+1)+(p+1)}/fi fi: Count source frequency n: Value set in PREZ register, m: value set in TZPR register, p: value set in TZSC register
Count start condition	1 (count starts) is written to the TZS bit in the TZMR register.
Count stop condition	0 (count stops) is written to the TZS bit in the TZMR register.
Interrupt request generation timing	In half a cycle of count source, after timer Z underflows during the secondary period (at the same time as the TZOUT output change) [timer Z interrupt].
TZOUT pin function	Pulse output (To use this pin as a programmable I/O port, select timer mode.)
$\overline{\text{INT0}}$ pin function	Programmable I/O port, or $\overline{\text{INT0}}$ interrupt input
Read from timer	The count value can be read out by reading registers TZPR and PREZ[1].
Write to timer	The value written to registers TZSC, PREZ, and TZPR is written to the reload register only[2]
Select functions	• Output level latch select function The TZOPL bit can select the output level during primary and secondary periods. • Programmable waveform generation output switch function When the TZOCNT bit in the TZOC register is set to 0, the output from the TZOUT pin is inverted synchronously when timer Z underflows. When set to 1, the value in the P1_3 bit is output from the TZOUT pin[3]

NOTES:

1. Even when counting the secondary period, the TZPR register may be read.
2. The value set in registers TZPR and TZSC are made effective by writing a value to the TZPR register. The set values are reflected in the waveform output beginning with the following primary period after writing to the TZPR register.
3. The TZOCNT bit is enabled by the following.
   - When counting starts.
   - When a timer Z interrupt request is generated. The contents after the TZOCNT bit is changed are reflected from the output of the following primary period.

Timer Z Mode Register



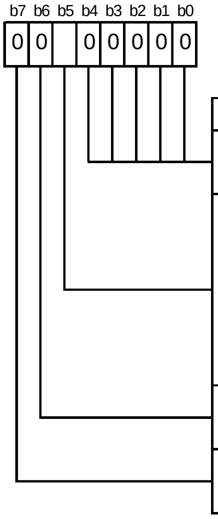
Symbol	Address	After Reset
TZMR	0080h	00h

Bit Symbol	Bit Name	Function	RW
— (b3-b0)	Reserved bits	Set to 0.	RW
TZMOD0	Timer Z operating mode bits	b5 b4 0 1 : Programmable w aveform generation mode	RW
TZMOD1			RW
TZWC	Timer Z w rite control bit	Set to 1 in programmable w aveform generation mode.[1]	RW
TZS	Timer Z count start flag[2]	0 : Stops counting. 1 : Starts counting.	RW

NOTES:

1. When the TZS bit is set to 1 (count starts), the count value is w ritten to the reload register only. When the TZS bit is set to 0 (count stops), the count value is w ritten to both reload register and counter.
2. Refer to **14.2.5 Notes on Timer Z** for precautions regarding the TZS bit.

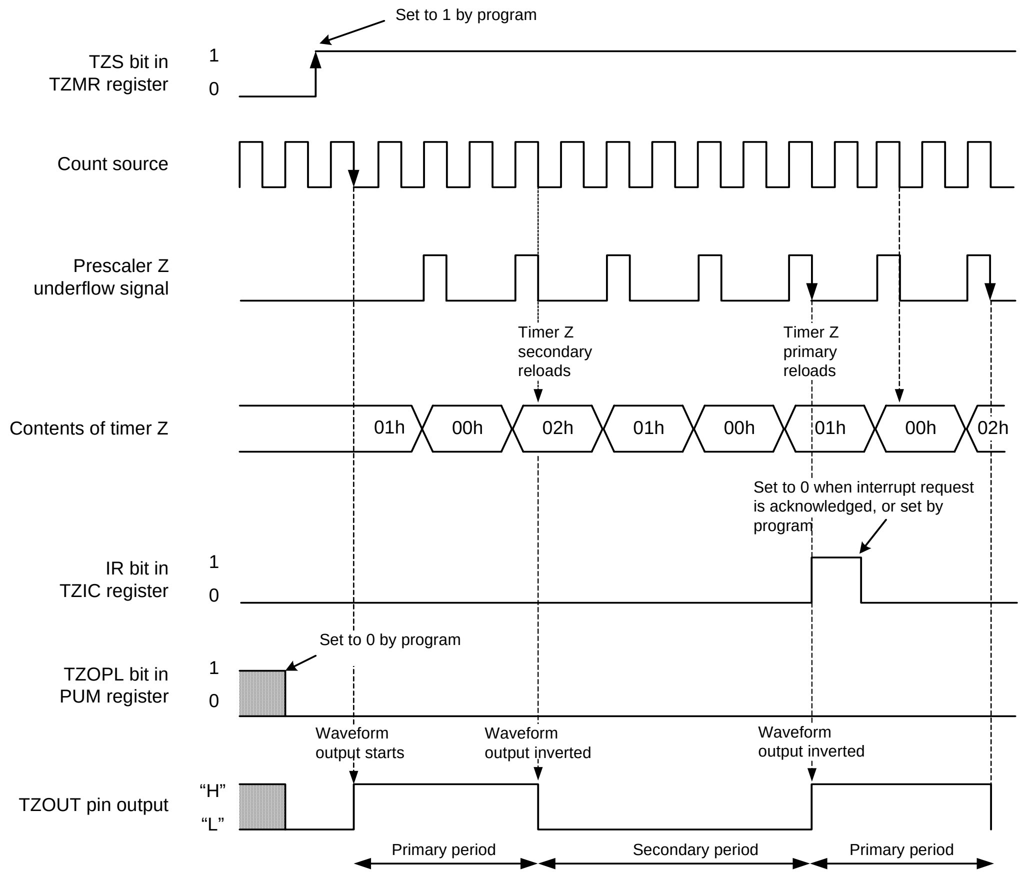
Timer Z Waveform Output Control Register



Symbol	Address	After Reset
PUM	0084h	00h

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	Reserved bits	Set to 0.	RW
TZOPL	Timer Z output level latch	0 : Outputs “H” for primary period. Outputs “L” for secondary period. Outputs “L” w hen the timer is stopped. 1 : Outputs “L” for primary period. Outputs “H” for secondary period. Outputs “H” w hen the timer is stopped.	RW
INOSTG	$\overline{\text{INT0}}$ pin one-shot trigger control bit	Set to 0 in programmable w aveform generation mode.	RW
INOSEG	$\overline{\text{INT0}}$ pin one-shot trigger polarity select bit	Set to 0 in programmable w aveform generation mode.	RW

Figure 14.17 Registers TZMR and PUM in Programmable Waveform Generation Mode



The above applies under the following conditions.
PREZ = 01h, TZPR = 01h, TZSC = 02h
TZOC register TZOCNT bit = 0

Figure 14.18 Operating Example of Timer Z in Programmable Waveform Generation Mode

14.2.3 Programmable One-shot Generation Mode

In programmable one-shot generation mode, one-shot pulse is output from the TZOUT pin by a program or an external trigger input (input to the $\overline{\text{INT0}}$ pin) (refer to **Table 14.9 Programmable One-Shot Generation Mode Specifications**). When a trigger is generated, the timer starts operating from the point only once for a given period equal to the set value in the TZPR register. The TZSC register is not used in this mode. Figure 14.19 shows Registers TZMR and PUM in Programmable One-Shot Generation Mode. Figure 14.20 shows an Operating Example in Programmable One-Shot Generation Mode.

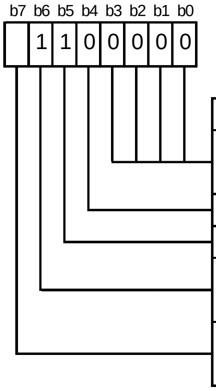
Table 14.9 Programmable One-Shot Generation Mode Specifications

Item	Specification
Count sources	f1, f2, f8, Timer X underflow
Count operations	• Decrement the value set in the TZPR register • When the timer underflows, it reloads the contents of the reload register before the count completes and the TZOS bit is set to 0 (one-shot stops). • When the count stops, the timer reloads the contents of the reload register before it stops.
One-shot pulse output time	(n+1)(m+1)/fi fi: Count source frequency, n: value set in PREZ register, m: value set in TZPR register
Count start conditions	• Set the TZOS bit in the TZOC register to 1 (one-shot starts).(1) • Input active trigger to the $\overline{\text{INT0}}$ pin(2)
Count stop conditions	• When reloading completes after the count value is set to 00h. • When the TZS bit in the TZMR register is set to 0 (count stops). • When the TZOS bit in the TZOC register is set to 0 (one-shot stops).
Interrupt request generation timing	In half a cycle of the count source, after the timer underflows (at the same time as the TZOUT output ends) [timer Z interrupt].
TZOUT pin function	Pulse output (To use this pin as a programmable I/O port, select timer mode.)
$\overline{\text{INT0}}$ pin function	• When the INOSTG bit in the PUM register is set to 0 ($\overline{\text{INT0}}$ one-shot trigger disabled): programmable I/O port or $\overline{\text{INT0}}$ interrupt input • When the INOSTG bit in the PUM register is set to 1 ($\overline{\text{INT0}}$ one-shot trigger enabled): external trigger ($\overline{\text{INT0}}$ interrupt input)
Read from timer	The count value can be read out by reading registers TZPR and PREZ.
Write to timer	The value written to registers TZPR and PREZ is written to the reload register only(3).
Select functions	• Output level latch select function The TZOPL bit can select the output level of the one-shot pulse waveform. • $\overline{\text{INT0}}$ pin one-shot trigger control and polarity select functions The INOSTG bit can select the trigger as active or inactive from the $\overline{\text{INT0}}$ pin. Also, the INOSEG bit can select the active trigger polarity.

NOTES:

1. Set the TZS bit in the TZMR register to 1 (count starts).
2. Set the TZS bit to 1 (count starts), the INT0EN bit in the INTEN register to 1 (enables $\overline{\text{INT0}}$ input), and the INOSTG bit in the PUM register to 1 ($\overline{\text{INT0}}$ one-shot trigger enabled). A trigger which is input during the count cannot be acknowledged, however an $\overline{\text{INT0}}$ interrupt request is generated.
3. The set value is reflected at the following one-shot pulse after writing to the TZPR register.

Timer Z Mode Register



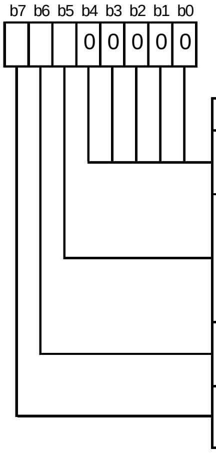
Symbol	Address	After Reset
TZMR	0080h	00h

Bit Symbol	Bit Name	Function	RW
— (b3-b0)	Reserved bits	Set to 0.	RW
TZMOD0	Timer Z operating mode bits	b5 b4 1 0 : Programmable one-shot generation mode	RW
TZMOD1			RW
TZWC	Timer Z w rite control bit	Set to 1 in programmable one-shot generation mode.[1]	RW
TZS	Timer Z count start flag[2]	0 : Stops counting. 1 : Starts counting.	RW

NOTES:

1. When the TZS bit is set to 1 (count starts), the count value is w ritten to the reload register only. When the TZS bit is set to 0 (count stops), the count value is w ritten to both reload register and counter.
2. Refer to **14.2.5 Notes on Timer Z** for precautions regarding the TZS bit.

Timer Z Waveform Output Control Register



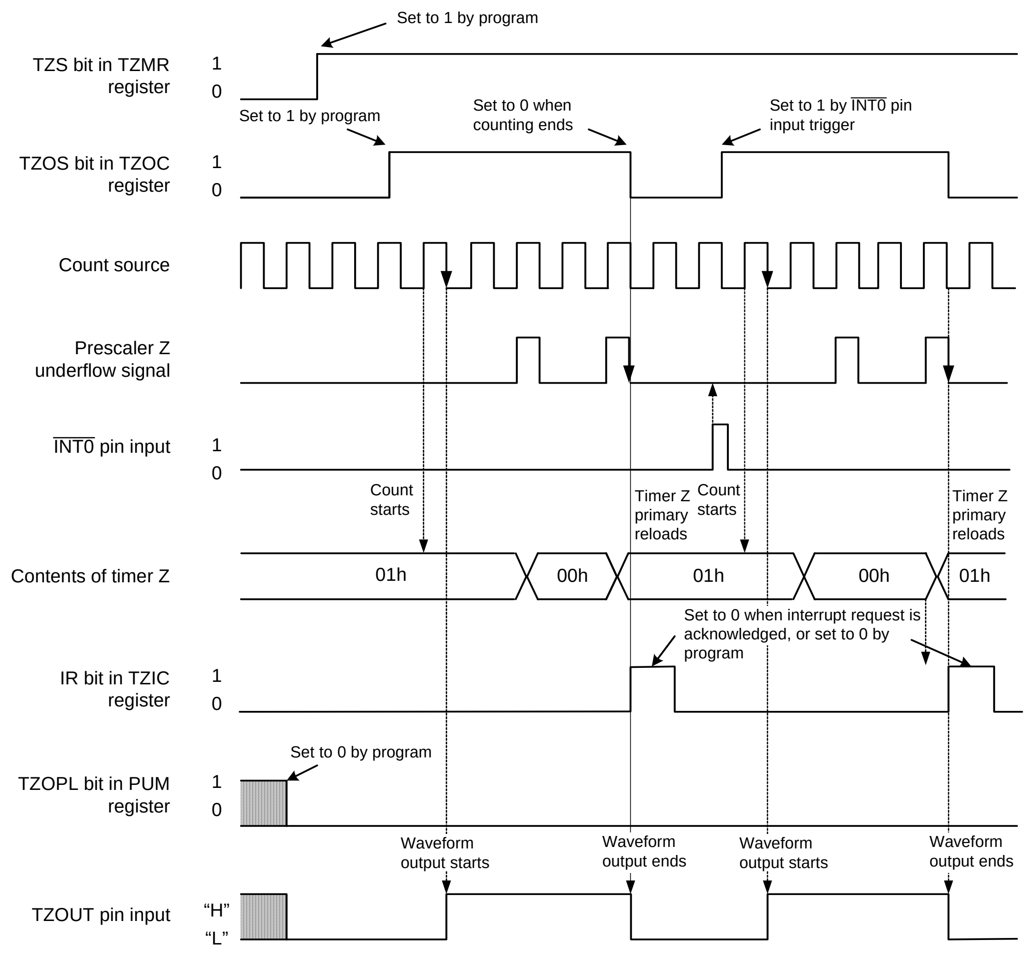
Symbol	Address	After Reset
PUM	0084h	00h

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	Reserved bits	Set to 0.	RW
TZOPL	Timer Z output level latch	0 : Outputs one-shot pulse “H”. Outputs “L” w hen the timer is stopped. 1 : Outputs one-shot pulse “L”. Outputs “H” w hen the timer is stopped.	RW
INOSTG	$\overline{\mathrm{INT0}}$ pin one-shot trigger control bit[1]	0 : $\overline{\mathrm{INT0}}$ pin one-shot trigger disabled 1 : $\overline{\mathrm{INT0}}$ pin one-shot trigger enabled	RW
INOSEG	$\overline{\mathrm{INT0}}$ pin one-shot trigger polarity select bit[2]	0 : Falling edge trigger 1 : Rising edge trigger	RW

NOTES:

1. Set the INOSTG bit to 1 after the INT0EN bit in the INTEN register and the INOSEG bit in the PUM register are set. When setting the INOSTG bit to 1 ($\overline{\mathrm{INT0}}$ pin one-shot trigger enabled), set bits INT0F0 to INT0F1 in the INT0F register. Set the INOSTG bit to 0 ($\overline{\mathrm{INT0}}$ pin one-shot trigger disabled) after the TZS bit in the TZMR register is set to 0 (count stops).
2. The INOSEG bit is enabled only w hen the INT0PL bit in the INTEN register is set to 0 (one edge).

Figure 14.19 Registers TZMR and PUM in Programmable One-Shot Generation Mode



The above applies under the following conditions.
PREZ = 01h, TZPR = 01h
TZOPL bit in PUM register = 0, INOSTG bit = 1 (INT0 one-shot trigger enabled)
INOSEG bit = 1 (rising edge trigger)

Figure 14.20 Operating Example in Programmable One-Shot Generation Mode

14.2.4 Programmable Wait One-Shot Generation Mode

In programmable wait one-shot generation mode, one-shot pulse is output from the TZOUT pin by a program or an external trigger input (input to the $\overline{\text{INT0}}$ pin) (refer to **Table 14.10 Programmable Wait One-Shot Generation Mode Specifications**). When a trigger is generated, from that point, the timer outputs a pulse only once for a given length of time equal to the value set in the TZSC register after waiting for a given length of time equal to the setting value in the TZPR register. Figure 14.21 shows the Registers TZMR and PUM in Programmable Wait One-Shot Generation Mode. Figure 14.22 shows an Operating Example in Programmable Wait One-Shot Generation Mode.

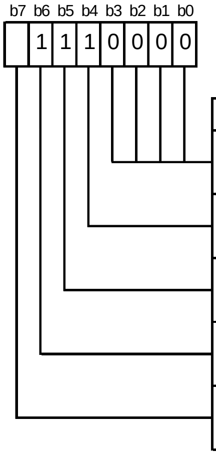
Table 14.10 Programmable Wait One-Shot Generation Mode Specifications

Item	Specification
Count sources	f1, f2, f8, Timer X underflow
Count operations	• Decrement the value set in Timer Z primary • When the count of TZPR register underflows, the timer reloads the contents of the TZSC register before the count is continued. • When the count of the TZSC register underflows, the timer reloads the contents of the TZPR register before the count completes and the TZOS bit is set to 0. • When the count stops, the timer reloads the contents of the reload register before it stops.
Wait time	(n+1)(m+1)/fi fi: Count source frequency n: Value set in PREZ register, m: value set in TZPR register
One-shot pulse output time	(n+1)(p+1)/fi fi: Count source frequency n: Value set in PREZ register, p: value set in TZSC register
Count start conditions	• Set the TZOS bit in the TZOC register to 1 (one-shot starts).(1) • Input active trigger to the $\overline{\text{INT0}}$ pin(2)
Count stop conditions	• When reloading completes after timer Z underflows during secondary period. • When the TZS bit in the TZMR register is set to 0 (count stops). • When the TZOS bit in the TZOC register is set to 0 (one-shot stops).
Interrupt request generation timing	In half a cycle of the count source after timer Z underflows during secondary period (complete at the same time as waveform output from the TZOUT pin) [timer Z interrupt].
TZOUT pin function	Pulse output (To use this pin as a programmable I/O port, select timer mode.)
$\overline{\text{INT0}}$ pin function	• When the INOSTG bit in the PUM register is set to 0 ($\overline{\text{INT0}}$ one-shot trigger disabled): programmable I/O port or $\overline{\text{INT0}}$ interrupt input • When the INOSTG bit in the PUM register is set to 1 ($\overline{\text{INT0}}$ one-shot trigger enabled): external trigger ($\overline{\text{INT0}}$ interrupt input)
Read from timer	The count value can be read out by reading registers TZPR and PREZ.
Write to timer	The value written to registers TZPR and PREZ is written to the reload register only(3).
Select functions	• Output level latch select function The output level of the one-shot pulse waveform is selected by the TZOPL bit. • $\overline{\text{INT0}}$ pin one-shot trigger control function and polarity select function Trigger input from the $\overline{\text{INT0}}$ pin can be set to active or inactive by the INOSTG bit. Also, the active trigger's polarity can be selected by the INOSEG bit.

NOTES:

1. The TZS bit in the TZMR register must be set to 1 (start counting).
2. The TZS bit must be set to 1 (start counting), the INT0EN bit in the INTEN register to 1 (enabling $\overline{\text{INT0}}$ input), and the INOSTG bit in the PUM register to 1 (enabling $\overline{\text{INT0}}$ one-shot trigger). A trigger which is input during the count cannot be acknowledged, however an $\overline{\text{INT0}}$ interrupt request is generated.
3. The set values are reflected at the following one-shot pulse after writing to the TZPR register.

Timer Z Mode Register



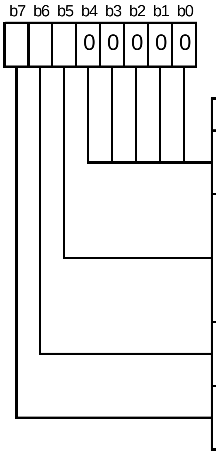
Symbol	Address	After Reset
TZMR	0080h	00h

Bit Symbol	Bit Name	Function	RW
— (b3-b0)	Reserved bits	Set to 0.	RW
TZMOD0	Timer Z operating mode bits	b5 b4 1 1 : Programmable wait one-shot generation mode	RW
TZMOD1			RW
TZWC	Timer Z write control bit	Set to 1 in programmable wait one-shot generation mode.[1]	RW
TZS	Timer Z count start flag[2]	0 : Stops counting. 1 : Starts counting.	RW

NOTES:

1. When the TZS bit is set to 1 (count starts), the count value is written to the reload register only. When the TZS bit is set to 0 (count stops), the count value is written to both reload register and counter.
2. Refer to **14.2.5 Notes on Timer Z** for precautions regarding the TZS bit.

Timer Z Waveform Output Control Register



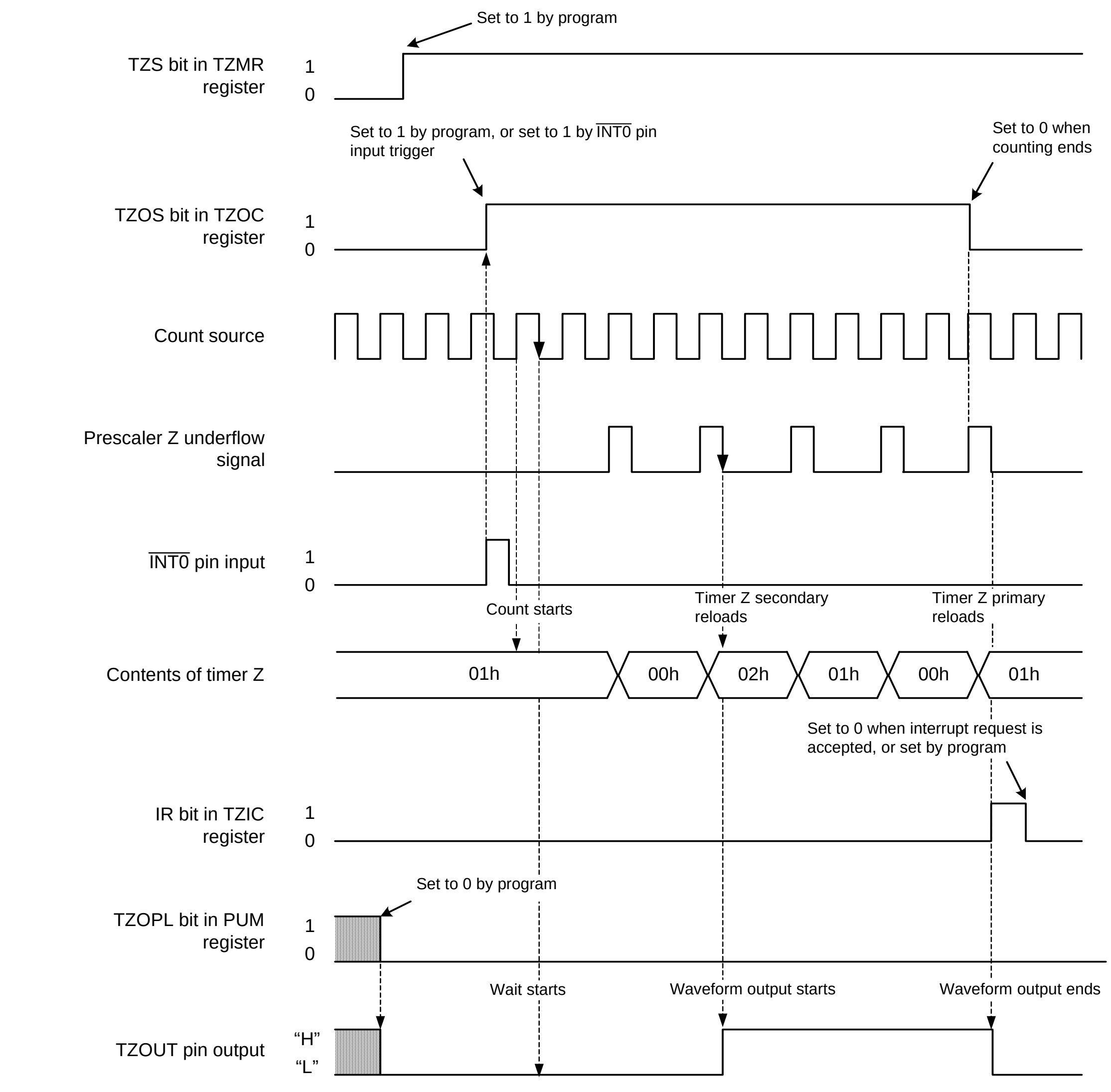
Symbol	Address	After Reset
PUM	0084h	00h

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	Reserved bits	Set to 0.	RW
TZOPL	Timer Z output level latch	0 : Outputs one-shot pulse “H”. Outputs “L” when the timer is stopped. 1 : Outputs one-shot pulse “L”. Outputs “H” when the timer is stopped.	RW
INOSTG	$\overline{\text{INT0}}$ pin one-shot trigger control bit[1]	0 : $\overline{\text{INT0}}$ pin one-shot trigger disabled 1 : $\overline{\text{INT0}}$ pin one-shot trigger enabled	RW
INOSEG	$\overline{\text{INT0}}$ pin one-shot trigger polarity select bit[2]	0 : Falling edge trigger 1 : Rising edge trigger	RW

NOTES:

1. Set the INOSTG bit to 1 after the INT0EN bit in the INTEN register and the INOSEG bit in the PUM register are set. When setting the INOSTG bit to 1 ($\overline{\text{INT0}}$ pin one-shot trigger enabled), set bits INT0F0 to INT0F1 in the INT0F register. Set the INOSTG bit to 0 ($\overline{\text{INT0}}$ pin one-shot trigger disabled) after the TZS bit in the TZMR register is set to 0 (count stops).
2. The INOSEG bit is enabled only when the INT0PL bit in the INTEN register is set to 0 (one edge).

Figure 14.21 Registers TZMR and PUM in Programmable Wait One-Shot Generation Mode


Set to 1 by program
TZS bit in TZMR register
1
0
Set to 1 by program, or set to 1 by $\overline{\text{INT0}}$ pin input trigger
Set to 0 when counting ends
TZOS bit in TZOC register
1
0
Count source
Prescaler Z underflow signal
$\overline{\text{INT0}}$ pin input
1
0
Count starts
Timer Z secondary reloads
Timer Z primary reloads
Contents of timer Z
01h
00h
02h
01h
00h
01h
Set to 0 when interrupt request is accepted, or set by program
IR bit in TZIC register
1
0
Set to 0 by program
TZOPL bit in PUM register
1
0
Wait starts
Waveform output starts
Waveform output ends
TZOUT pin output
"H"
"L"

The above applies under the following conditions.
PREZ = 01h, TZPR = 01h, TZSC = 02h
PUM register TZOPL bit = 0, INOSTG bit = 1 ($\overline{\text{INT0}}$ one-shot trigger enabled)
INOSEG bit = 1 (rising edge trigger)

Figure 14.22 Operating Example in Programmable Wait One-Shot Generation Mode

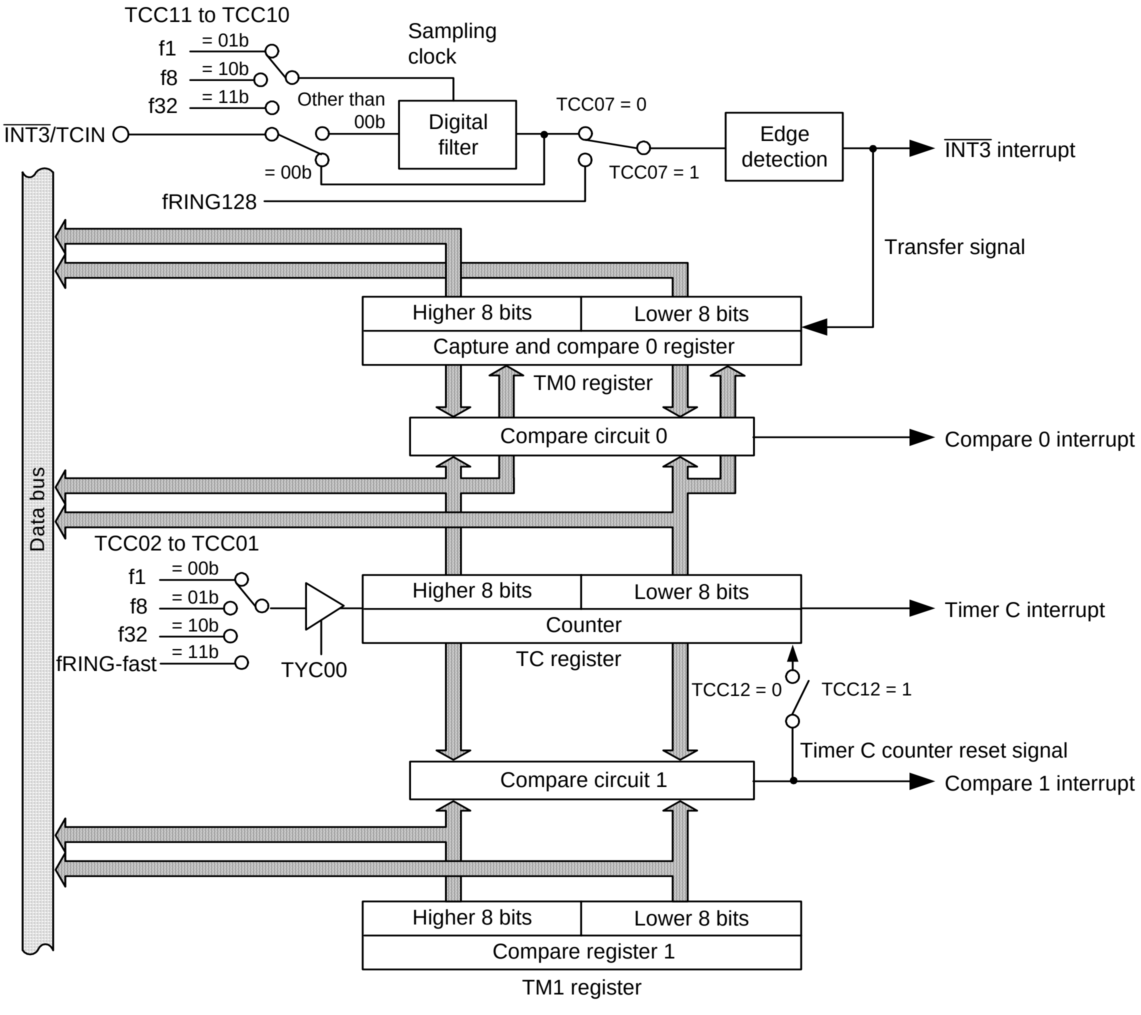
14.2.5 Notes on Timer Z

- Timer Z stops counting after a reset. Set the values in the timer and prescaler before the count starts.
- Even if the prescaler and timer are read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- Do not rewrite bits TZMOD0 to TZMOD1, and the TZS bit simultaneously.
- In programmable one-shot generation mode and programmable wait one-shot generation mode, when setting the TZS bit in the TZMR register to 0 (stops counting) or setting the TZOS bit in the TZOC register to 0 (stops one-shot), the timer reloads the value of the reload register and stops. Therefore, in programmable one-shot generation mode and programmable wait one-shot generation mode read the timer count value before the timer stops.
- The TZS bit in the TZMR register has a function to instruct timer Z to start or stop counting and a function to indicate that the count has started or stopped.
  0 (count stops) can be read until the following count source is applied after 1 (count starts) is written to the TZS bit while the count is being stopped. If the following count source is applied, 1 can be read from the TZS bit. After writing 1 to the TZS bit, do not access registers associated with timer Z (registers TZMR, PREZ, TZSC, TZPR, TZOC, PUM, TCSC, and TZIC) except for the TZS bit, until 1 can be read from the TZS bit. The count starts at the following count source after the TZS bit is set to 1.
  Also, after writing 0 (count stops) to the TZS bit during the count, timer Z stops counting at the following count source.
  1 (count starts) can be read by reading the TZS bit until the count stops after writing 0 to the TZS bit. After writing 0 to the TZS bit, do not access registers associated with timer Z except for the TZS bit, until 0 can be read from the TZS bit.

14.3 Timer C

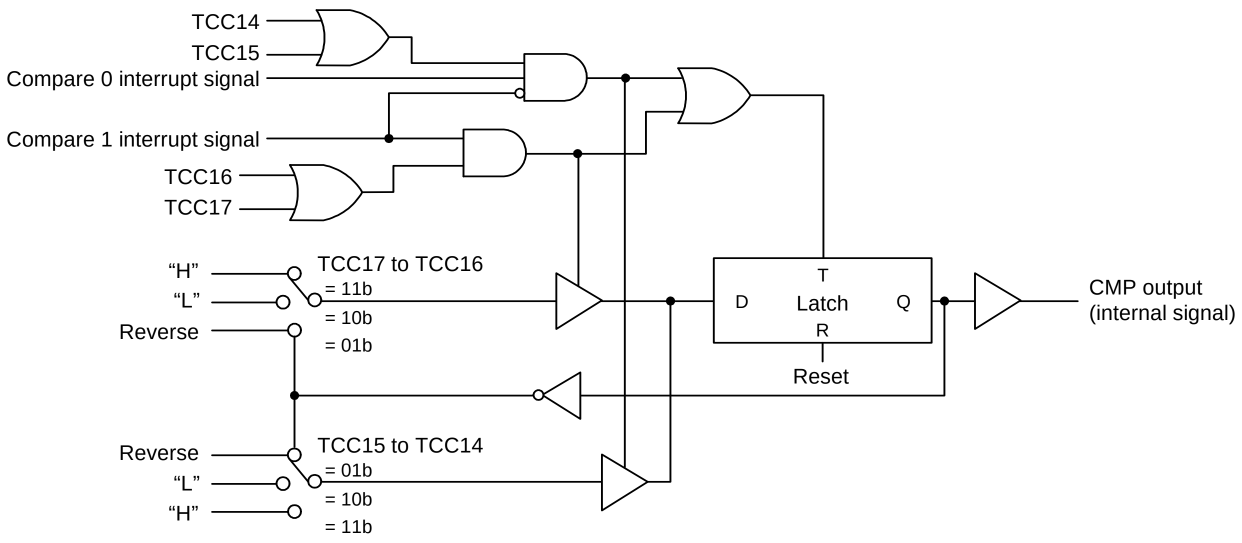
Timer C is a 16-bit timer. Figure 14.23 shows a Block Diagram of Timer C. Figure 14.24 shows a Block Diagram of CMP Waveform Generation Unit. Figure 14.25 shows a Block Diagram of CMP Waveform Output Unit.

Timer C has two modes: input capture mode and output compare mode. Figures 14.26 to 14.29 show the Timer C-associated registers.



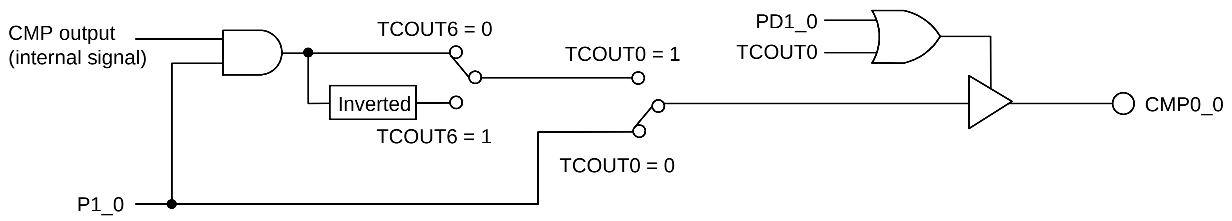
TCC01 to TCC02, TCC07: Bits in TCC0 register
TCC10 to TCC12: Bits in TCC1 register

Figure 14.23 Block Diagram of Timer C



TCC14 to TCC17: Bits in TCC1 register

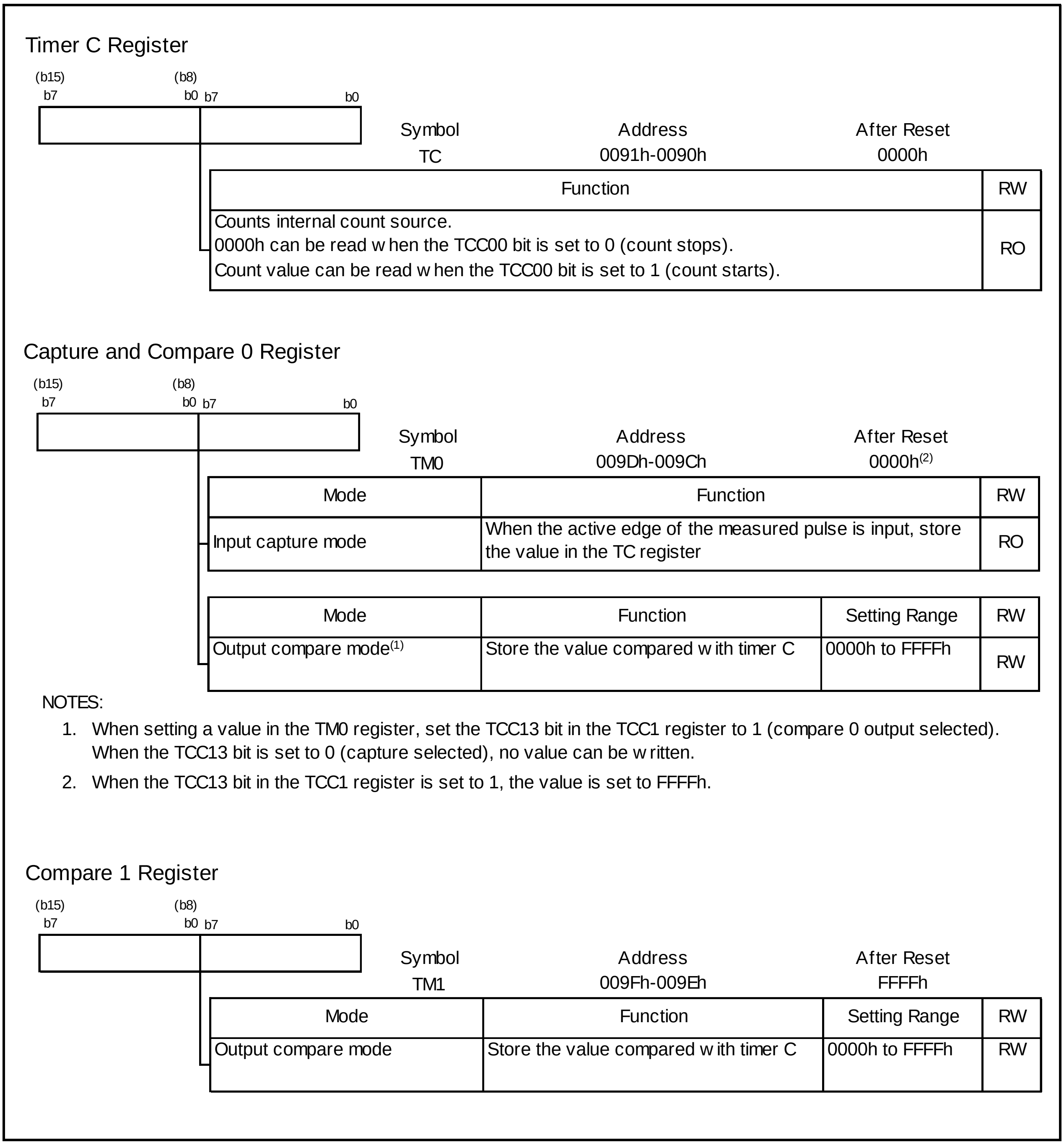
Figure 14.24 Block Diagram of CMP Waveform Generation Unit



Register	TCOUT	P1	TCOUT	CMP0_0 Output
Bit	TCOUT0	P1_0	TCOUT6	
Setting Value	1	1	0	CMP0_0 waveform output
	1	1	1	CMP0_0 reversed waveform output
	1	0	0	"L" output
	1	0	1	"H" output

This diagram is a block diagram of the CMP0_0 waveform output unit.
The CMP0_1 to CMP0_2 and CMP1_0 to CMP1_2 waveform output units have the same configuration.

Figure 14.25 Block Diagram of CMP Waveform Output Unit



Timer C Register

Symbol	Address	After Reset
TC	0091h-0090h	0000h

Function	RW
Counts internal count source. 0000h can be read w hen the TCC00 bit is set to 0 (count stops). Count value can be read w hen the TCC00 bit is set to 1 (count starts).	RO

Capture and Compare 0 Register

Symbol	Address	After Reset
TM0	009Dh-009Ch	0000h[2]

Mode	Function	RW
Input capture mode	When the active edge of the measured pulse is input, store the value in the TC register	RO

Mode	Function	Setting Range	RW
Output compare mode[1]	Store the value compared w ith timer C	0000h to FFFFh	RW

NOTES:

1. When setting a value in the TM0 register, set the TCC13 bit in the TCC1 register to 1 (compare 0 output selected). When the TCC13 bit is set to 0 (capture selected), no value can be w ritten.
2. When the TCC13 bit in the TCC1 register is set to 1, the value is set to FFFFh.

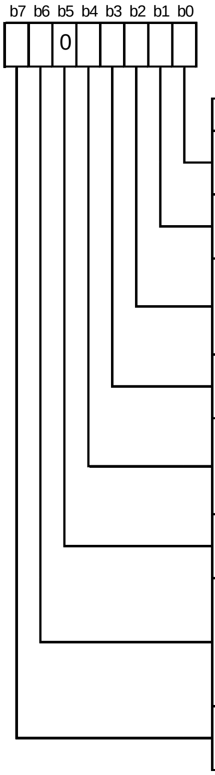
Compare 1 Register

Symbol	Address	After Reset
TM1	009Fh-009Eh	FFFFh

Mode	Function	Setting Range	RW
Output compare mode	Store the value compared w ith timer C	0000h to FFFFh	RW

Figure 14.26 Registers TC, TM0, and TM1

Timer C Control Register 0



Symbol	Address	After Reset
TCC0	009Ah	00h

Bit Symbol	Bit Name	Function	RW
TCC00	Timer C count start bit	0 : Stops counting. 1 : Starts counting.	RW
TCC01	Timer C count source select bits[1]	b2 b1 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : fRING-fast	RW
TCC02			RW
TCC03	$\overline{\text{INT3}}$ interrupt / capture polarity select bits[1, 2]	b4 b3 0 0 : Rising edge 0 1 : Falling edge 1 0 : Both edges 1 1 : Do not set.	RW
TCC04			RW
— (b5)	Reserved bit	Set to 0.	RW
TCC06	$\overline{\text{INT3}}$ interrupt generation timing select bit[2, 3]	0 : $\overline{\text{INT3}}$ interrupt is generated in synchronization with timer C count source. 1 : $\overline{\text{INT3}}$ interrupt is generated when $\overline{\text{INT3}}$ interrupt is input[4].	RW
TCC07	$\overline{\text{INT3}}$ interrupt / capture input switch bit[1, 2]	0 : $\overline{\text{INT3}}$ 1 : fRING128	RW

NOTES:

1. Change this bit when the TCC00 bit is set to 0 (count stops).
2. The IR bit in the INT3IC register may be set to 1 (requests interrupt) when the TCC03, TCC04, TCC06, or TCC07 bit is rewritten. Refer to **12.5.5 Changing Interrupt Sources**.
3. When the TCC13 bit is set to 1 (output compare mode) and $\overline{\text{INT3}}$ interrupt is input, regardless of the setting value of the TCC06 bit, an interrupt request is generated.
4. When using the $\overline{\text{INT3}}$ filter, the $\overline{\text{INT3}}$ interrupt is generated is synchronization with the clock for the digital filter.

Figure 14.27 TCC0 Register

Timer C Control Register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
TCC1	009Bh	00h

Bit Symbol	Bit Name	Function	RW
TCC10	$\overline{\text{INT3}}$ filter select bits[1]	b1 b0 0 0 : No filter 0 1 : Filter with f1 sampling 1 0 : Filter with f8 sampling 1 1 : Filter with f32 sampling	RW
TCC11			RW
TCC12	Timer C counter reload select bit[3]	0 : No reload 1 : Set TC register to 0000h when compare 1 is matched.	RW
TCC13	Compare 0 / capture select bit[2]	0 : Selects capture (input capture mode).[3] 1 : Selects compare 0 output (output compare mode).	RW
TCC14	Compare 0 output mode select bits[3]	b5 b4 0 0 : CMP output remains unchanged even when compare 0 is matched. 0 1 : CMP output is inverted when compare 0 signal is matched. 1 0 : CMP output is set to “L” when compare 0 signal is matched. 1 1 : CMP output is set to “H” when compare 0 signal is matched.	RW
TCC15			
TCC16	Compare 1 output mode select bits[3]	b7 b6 0 0 : CMP output remains unchanged even when compare 1 is matched. 0 1 : CMP output is inverted when compare 1 signal is matched. 1 0 : CMP output is set to “L” when compare 1 signal is matched. 1 1 : CMP output is set to “H” when compare 1 signal is matched.	RW
TCC17			

NOTES:

1. When the same value is sampled from the $\overline{\text{INT3}}$ pin three times continuously, the input is determined.
2. When the TCC00 bit in the TCC0 register is set to 0 (count stops), rewrite the TCC13 bit.
3. When the TCC13 bit is set to 0 (input capture mode), set bits TCC12, and TCC14 to TCC17 to 0.

Figure 14.28 TCC1 Register

Timer C Output Control Register[1]

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
TCOUT	00FFh	00h

Bit Symbol	Bit Name	Function	RW
TCOUT0	CMP output enable bit 0	0 : Disables CMP output from CMP0_0. 1 : Enables CMP output from CMP0_0.	RW
TCOUT1	CMP output enable bit 1	0 : Disables CMP output from CMP0_1. 1 : Enables CMP output from CMP0_1.	RW
TCOUT2	CMP output enable bit 2	0 : Disables CMP output from CMP0_2. 1 : Enables CMP output from CMP0_2.	RW
TCOUT3	CMP output enable bit 3	0 : Disables CMP output from CMP1_0. 1 : Enables CMP output from CMP1_0.	RW
TCOUT4	CMP output enable bit 4	0 : Disables CMP output from CMP1_1. 1 : Enables CMP output from CMP1_1.	RW
TCOUT5	CMP output enable bit 5	0 : Disables CMP output from CMP1_2. 1 : Enables CMP output from CMP1_2.	RW
TCOUT6	CMP output invert bit 0	0 : Does not invert CMP output from CMP0_0 to CMP0_2. 1 : Inverts CMP output from CMP0_0 to CMP0_2.	RW
TCOUT7	CMP output invert bit 1	0 : Does not invert CMP output from CMP1_0 to CMP1_2. 1 : Inverts CMP output from CMP1_0 to CMP1_2.	RW

NOTE:

1. Set the bits w hich are not used for CMP output to 0.

Figure 14.29 TCOUT Register

14.3.1 Input Capture Mode

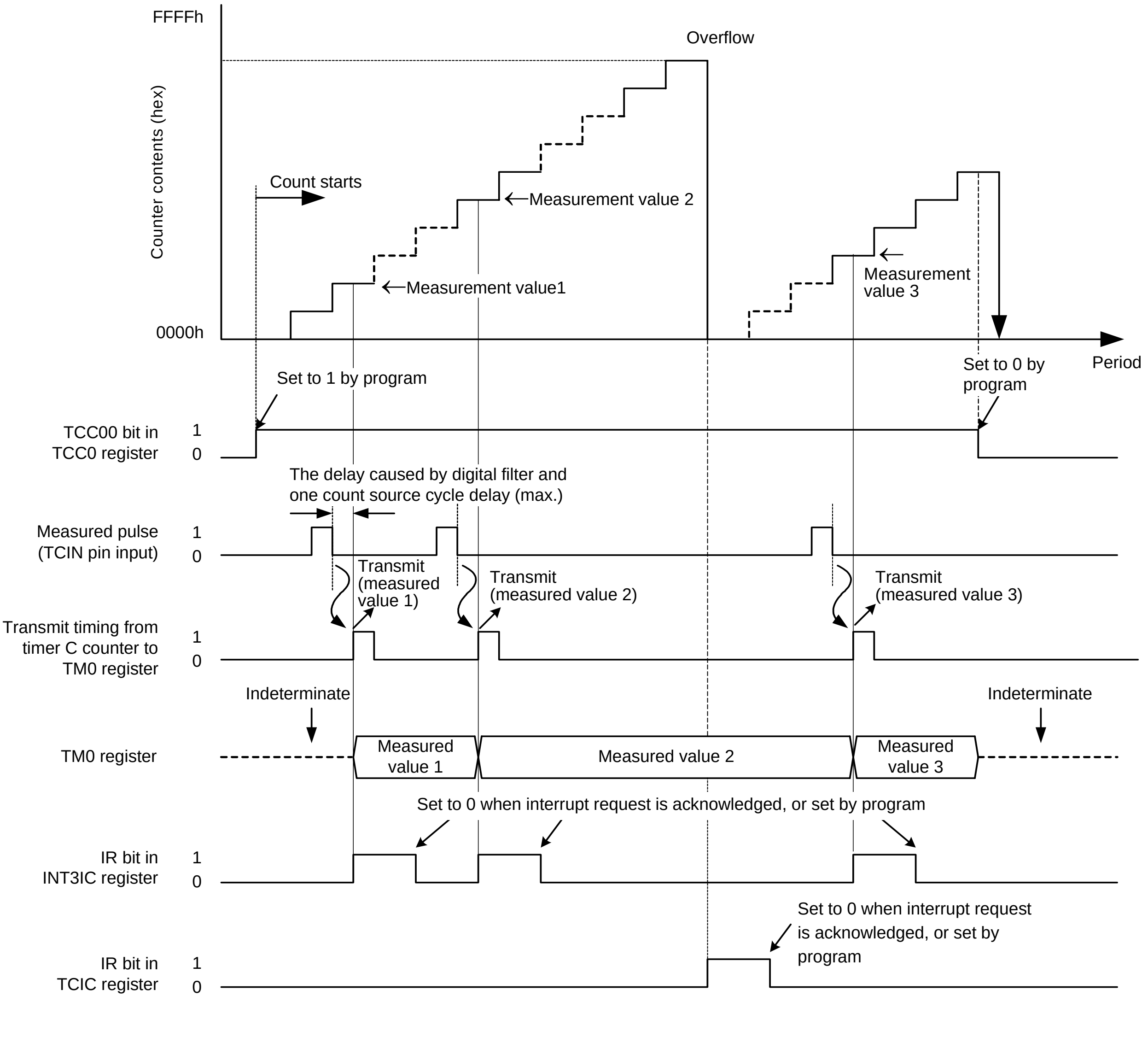
In input capture mode, the edge of the TCIN pin input signal or the fRING128 clock is used as a trigger to latch the timer value and generate an interrupt request. The TCIN input contains a digital filter, and this prevents errors caused by noise or the like from occurring. Table 14.11 shows the Input Capture Mode Specifications. Figure 14.30 shows an Operating Example in Input Capture Mode.

Table 14.11 Input Capture Mode Specifications

Item	Specification
Count sources	f1, f8, f32, fRING-fast
Count operations	• Increment • Transfer the value in the TC register to the TM0 register at the active edge of the measured pulse. • The value in the TC register is set to 0000h when the count stops.
Count start condition	The TCC00 bit in the TCC0 register is set to 1 (count starts).
Count stop condition	The TCC00 bit in the TCC0 register is set to 0 (count stops).
Interrupt request generation timing	• When the active edge of the measured pulse is input [$\overline{\text{INT3}}$ interrupt].[1] • When timer C overflows [timer C interrupt].
$\overline{\text{INT3}}$/TCIN pin function	Programmable I/O port or the measured pulse input ($\overline{\text{INT3}}$ interrupt input)
P1_0 to P1_2, P3_3 to P3_5 pin functions	Programmable I/O port
Counter value reset timing	When the TCC00 bit in the TCC0 register is set to 0 (capture disabled).
Read from timer[2]	• The count value can be read out by reading the TC register. • The count value at the measured pulse active edge input can be read out by reading the TM0 register.
Write to timer	Write to the TC and TM0 registers is disabled.
Select functions	• $\overline{\text{INT3}}$/TCIN polarity select function Bits TCC03 to TCC04 can select the active edge of the measured pulse. • Digital filter function Bits TCC11 to TCC10 can select the digital filter sampling frequency. • Trigger select function The TCC07 bit can select the TCIN input or the fRING128.

NOTES:

1. The $\overline{\text{INT3}}$ interrupt includes a digital filter delay and one count source (max.) delay.
2. Read registers TC and TM0 in 16-bit unit.


FFFFh
Counter contents (hex)
0000h
Overflow
Count starts
Measurement value 2
Measurement value1
Measurement value 3
Period
Set to 1 by program
Set to 0 by program
TCC00 bit in TCC0 register
1
0
The delay caused by digital filter and one count source cycle delay (max.)
Measured pulse (TCIN pin input)
1
0
Transmit (measured value 1)
Transmit (measured value 2)
Transmit (measured value 3)
Transmit timing from timer C counter to TM0 register
1
0
Indeterminate
Indeterminate
TM0 register
Measured value 1
Measured value 2
Measured value 3
Set to 0 when interrupt request is acknowledged, or set by program
IR bit in INT3IC register
1
0
Set to 0 when interrupt request is acknowledged, or set by program
IR bit in TCIC register
1
0

The above applies under the following conditions.
TCC0 register TCC04 to TCC03 bits = 01b (capture input polarity is set for falling edge).
TCC07 = 0 ($\overline{\text{INT3}}$/TCIN input as capture input trigger)

Figure 14.30 Operating Example in Input Capture Mode

14.3.2 Output Compare Mode

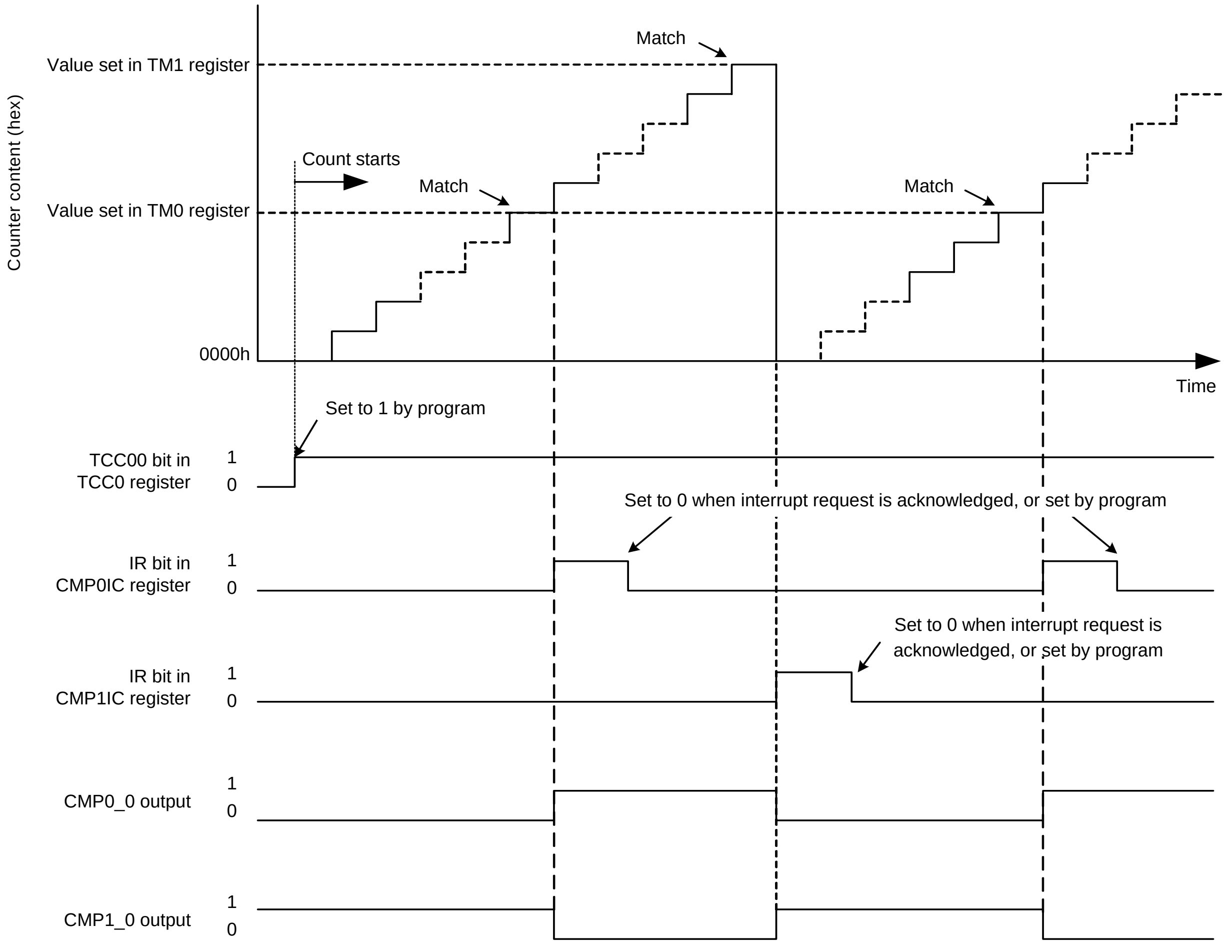
In output compare mode, an interrupt request is generated when the value of the TC register matches the value of the TM0 or TM1 register. Table 14.12 shows the Output Compare Mode Specifications. Figure 14.31 shows an Operating Example in Output Compare Mode.

Table 14.12 Output Compare Mode Specifications

Item	Specification
Count sources	f1, f8, f32, fRING-fast
Count operations	• Increment • The value in the TC register is set to 0000h when the count stops.
Count start condition	The TCC00 bit in the TCC0 register is set to 1 (count starts).
Counter stop condition	The TCC00 bit in the TCC0 register is set to 0 (count stops).
Waveform output start condition	Bits TCOUT0 to TCOUT5 in the TCOUT register are set to 1 (enables CMP output).(2)
Waveform output stop condition	Bits TCOUT0 to TCOUT5 in the TCOUT register are set to 0 (disables CMP output).
Interrupt request generation timing	• When a match occurs in compare circuit 0 [compare 0 interrupt]. • When a match occurs in compare circuit 1 [compare 1 interrupt]. • When time C overflows [timer C interrupt].
$\overline{\text{INT3}}$/TCIN pin function	Programmable I/O port or $\overline{\text{INT3}}$ interrupt input
P1_0 to P1_2 pins and P3_0 to P3_2 pins functions	Programmable I/O port or CMP output(1)
Counter value reset timing	When the TCC00 bit in the TCC0 register is set to 0 (count stops).
Read from timer(2)	• The value in the compare register can be read out by reading registers TM0 and TM1. • The count value can be read out by reading the TC register.
Write to timer(2)	• Write to the TC register is disabled. • The values written to registers TM0 and TM1 are stored in the compare register in the following timings: - When registers TM0 and TM1 are written to, if the TCC00 bit is set to 0 (count stops). - When the counter overflows, if the TCC00 bit is set to 1 (during counting) and the TCC12 bit in the TCC1 register is set to 0 (free-run). - When the compare 1 matches a counter, if the TCC00 bit is set to 1 and the TCC12 bit is set to 1 (the TC register is set to 0000h at compare 1 match).
Select functions	• Timer C counter reload select function The TCC12 bit in the TCC1 register can select whether the counter value in the TC register is set to 0000h when the compare circuit 1 match. • Bits TCC14 to TCC15 in the TCC1 register can be used to select the output level when compare circuit 0 matches. Bits TCC16 to TCC17 in the TCC1 register can be used to select the output level when compare circuit 1 matches. • Bits TCOUT6 to TCOUT7 in the TCOUT register can select whether the output is inverted or not.

NOTES:

1. When the corresponding port data is 1, the waveform is output depending on the setting of the registers TCC1 and TCOUT. When the corresponding port data is 0, the fixed level is output (refer to **Figure 14.25 Block Diagram of CMP Waveform Output Unit**).
2. Access registers TC, TM0, and TM1 in 16-bit units.



Conditions:

TCC12 bit in TCC1 register = 1 (TC register is set to 0000h at compare 1 match occurrence)
TCC13 bit in TCC1 register = 1 (Compare 0 output selected)
TCC15 to TCC14 bits in TCC1 register = 11b (CMP output level is set to high at compare 0 match occurrence)
TCC17 to TCC16 bits in TCC1 register = 10b (CMP output level is set to low at compare 1 match occurrence)
TCOUT6 bit in TCOUT register = 0 (not inverted)
TCOUT7 bit in TCOUT register = 1 (inverted)
TCOUT0 bit in TCOUT register = 1 (CMP0_0 output enabled)
TCOUT3 bit in TCOUT register = 1 (CMP1_0 output enabled)
P1_0 bit in P1 register = 1 (high)
P3_0 bit in P3 register = 1 (high)

Figure 14.31 Operating Example in Output Compare Mode

14.3.3 Notes on Timer C

Access registers TC, TM0, and TM1 in 16-bit units.
The TC register can be read in 16-bit units. This prevents the timer value from being updated between when the low-order bytes and high-order bytes are being read.

Example of reading timer C:

```
    MOV.W       0090H,R0        ; Read out timer C
```

15. Serial Interface

The serial interface consists of two channels (UART0 and UART1). Each UARTi (i = 0 or 1) has an exclusive timer to generate the transfer clock and operates independently.
Figure 15.1 shows a UARTi (i = 0 or 1) Block Diagram. Figure 15.2 shows a UARTi Transmit/Receive Unit.
UART0 has two modes: clock synchronous serial I/O mode and clock asynchronous serial I/O mode (UART mode).
UART1 has only clock asynchronous serial I/O mode (UART mode).
Figures 15.3 to 15.6 show the Registers Associated with UARTi.

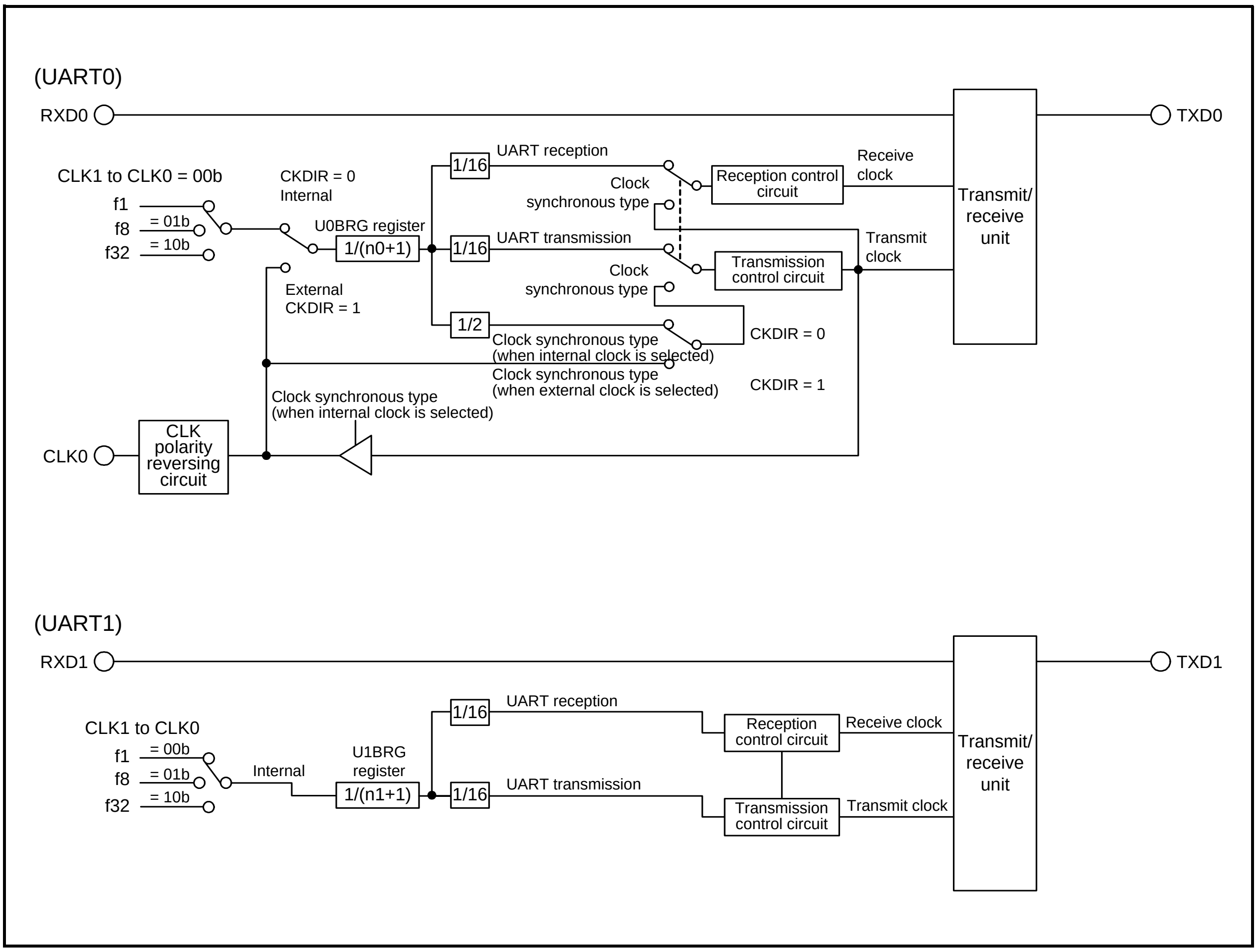
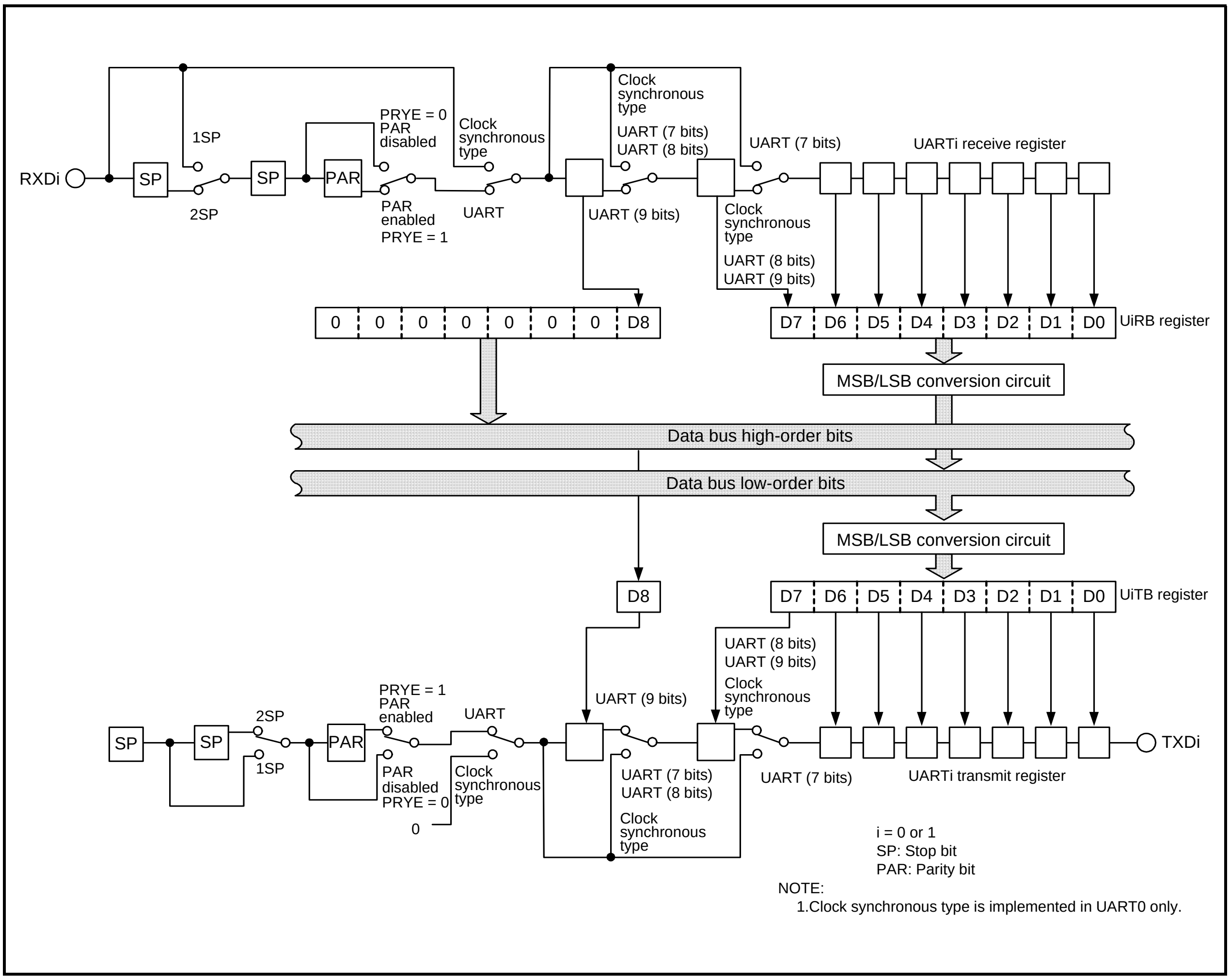


Figure 15.1 UARTi (i = 0 or 1) Block Diagram

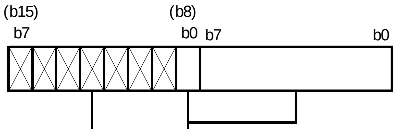


i = 0 or 1
SP: Stop bit
PAR: Parity bit

NOTE:
1.Clock synchronous type is implemented in UART0 only.

Figure 15.2 UARTi Transmit/Receive Unit

UARTi Transmit Buffer Register (i = 0 or 1)(1, 2)



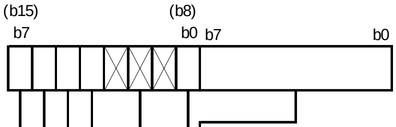
Symbol	Address	After Reset
U0TB	00A3h-00A2h	Undefined
U1TB	00ABh-00AAh	Undefined

Bit Symbol	Function	RW
— (b8-b0)	Transmit data	WO
— (b15-b9)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.	—

NOTES:

1. When the transfer data length is 9 bits, w rite data to high byte first, then low byte.
2. Use the MOV instruction to w rite to this register.

UARTi Receive Buffer Register (i = 0 or 1)(1)



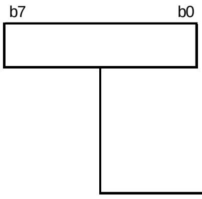
Symbol	Address	After Reset
U0RB	00A7h-00A6h	Undefined
U1RB	00AFh-00AEh	Undefined

Bit Symbol	Bit Name	Function	RW
— (b7-b0)	—	Receive data (D7 to D0)	RO
— (b8)	—	Receive data (D8)	RO
— (b11-b9)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		—
OER	Overrun error flag(2)	0 : No overrun error 1 : Overrun error	RO
FER	Framing error flag(2)	0 : No framing error 1 : Framing error	RO
PER	Parity error flag(2)	0 : No parity error 1 : Parity error	RO
SUM	Error sum flag(2)	0 : No error 1 : Error	RO

NOTES:

1. Read out the UiRB register in 16-bit units.
2. Bits SUM, PER, FER, and OER are set to 0 (no error) w hen 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 (receive disabled). The SUM bit is set to 0 (no error) w hen bits PER, FER, and OER are set to 0 (no error). Bits PER and FER are set to 0 even w hen the higher byte of the UiRB register is read out.
   Also, bits PER and FER are set to 0 w hen reading the high-order byte of the UiRB register.

UARTi Bit Rate Register (i = 0 or 1)(1, 2, 3)



Symbol	Address	After Reset
U0BRG	00A1h	Undefined
U1BRG	00A9h	Undefined

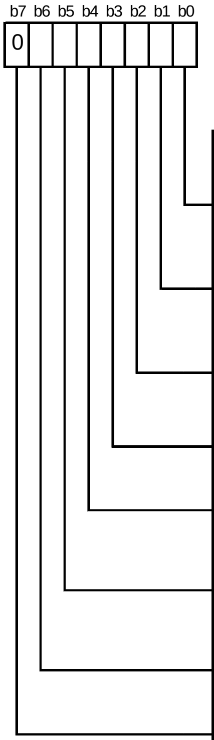
Function	Setting Range	RW
Assuming the set value is n, UiBRG divides the count source by n+1	00h to FFh	WO

NOTES:

1. Write to this register w hile the serial I/O is neither transmitting nor receiving.
2. Use the MOV instruction to w rite to this register.
3. After setting the CLK0 to CLK1 bits of the UiC0 register, w rite to the UiBRG register.

Figure 15.3 Registers U0TB to U1TB, U0RB to U1RB, and U0BRG to U1BRG

UARTi Transmit/Receive Mode Register (i = 0 or 1)



Symbol	Address	After Reset
U0MR	00A0h	00h
U1MR	00A8h	00h

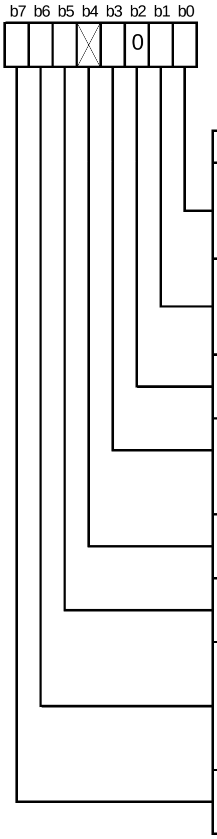
Bit Symbol	Bit Name	Function	RW
SMD0	Serial interface mode select bits[2]	b2 b1 b0 0 0 0 : Serial interface disabled 0 0 1 : Clock synchronous serial I/O mode 1 0 0 : UART mode transfer data 7 bits long 1 0 1 : UART mode transfer data 8 bits long 1 1 0 : UART mode transfer data 9 bits long Other than above : Do not set.	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/external clock select bit[3]	0 : Internal clock 1 : External clock[1]	RW
STPS	Stop bit length select bit	0 : 1 stop bit 1 : 2 stop bits	RW
PRY	Odd/even parity select bit	Enables w hen PRYE = 1. 0 : Odd parity 1 : Even parity	RW
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	RW
— (b7)	Reserved bit	Set to 0.	RW

NOTES:

1. Set the PD1_6 bit in the PD1 register to 0 (input).
2. Do not set bits SMD2 to SMD0 in the U1MR register to any values other than 000b, 100b, 101b, and 110b.
3. Set the CKDIR bit in UART1 to 0 (internal clock).

Figure 15.4 Registers U0MR to U1MR

UARTi Transmit/Receive Control Register 0 (i = 0 or 1)



Symbol	Address	After Reset
U0C0	00A4h	08h
U1C0	00ACh	08h

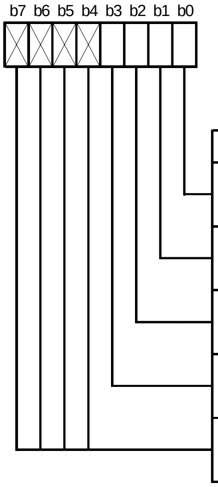
Bit Symbol	Bit Name	Function	RW
CLK0	BRG count source select bits[1]	b1 b0 0 0 : Selects f1. 0 1 : Selects f8. 1 0 : Selects f32. 1 1 : Do not set.	RW
CLK1			RW
— (b2)	Reserved bit	Set to 0.	RW
TXEPT	Transmit register empty flag	0 : Data in transmit register (during transmit) 1 : No data in transmit register (transmit completed)	RO
— (b4)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		—
NCH	Data output select bit	0 : TXDi pin is for CMOS output. 1 : TXDi pin is for N-channel open drain output.	RW
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge. 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge.	RW
UFORM	Transfer format select bit	0 : LSB first 1 : MSB first	RW

NOTE:

1. If the BRG count source is sw itched, set the UiBRG register again.

Figure 15.5 Registers U0C0 to U1C0

UARTi Transmit/Receive Control Register 1 (i = 0 or 1)



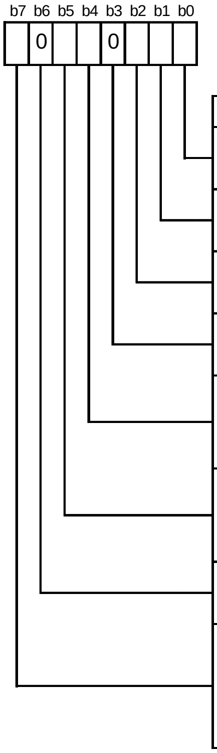
Symbol	Address	After Reset
U0C1	00A5h	02h
U1C1	00ADh	02h

Bit Symbol	Bit Name	Function	RW
TE	Transmit enable bit	0 : Disables transmission. 1 : Enables transmission.	RW
TI	Transmit buffer empty flag	0 : Data in UiTB register 1 : No data in UiTB register	RO
RE	Receive enable bit	0 : Disables reception. 1 : Enables reception.	RW
RI	Receive complete flag[1]	0 : No data in UiRB register 1 : Data in UiRB register	RO
— (b7-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		—

NOTE:

1. The RI bit is set to 0 when the higher byte of the UiRB register is read out. Set the PD1_6 bit in the PD1 register to 0 (input).

UART Transmit/Receive Control Register 2



Symbol	Address	After Reset
UCON	00B0h	00h

Bit Symbol	Bit Name	Function	RW
U0IRS	UART0 transmit interrupt source select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmit completed (TXEPT = 1)	RW
U1IRS	UART1 transmit interrupt source select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmit completed (TXEPT = 1)	RW
U0RRM	UART0 continuous receive mode enable bit	0 : Disables continuous receive mode. 1 : Enables continuous receive mode.	RW
— (b3)	Reserved bit	Set to 0.	RW
U1SEL0	UART1 pin (P3_7/TXD1, P4_5/RXD1) select bits	b5 b4 0 0 : P3_7, P4_5 0 1 : P3_7, RXD1 1 0 : Do not set. 1 1 : TXD1, RXD1	RW
U1SEL1			RW
— (b6)	Reserved bit	Set to 0.	RW
CNTRSEL	CNTR0 signal pin select bit[1]	0 : P1_5/RXD0 P1_7/CNTR00/$\overline{\text{INT10}}$ 1 : P1_5/RXD0/CNTR01/$\overline{\text{INT11}}$ P1_7	RW

NOTE:

1. The CNTRSEL bit selects the input pin of the CNTR0 ($\overline{\text{INT1}}$) signal.
   When the CNTR0 signal is output, it is output from the CNTR00 pin regardless of the CNTRSEL bit setting.

Figure 15.6 Registers U0C1 to U1C1, and UCON

15.1 Clock Synchronous Serial I/O Mode

In clock synchronous serial I/O mode, data is transmitted and received using a transfer clock. Table 15.1 lists the Clock Synchronous Serial I/O Mode Specifications. Table 15.2 lists the Registers Used and Settings in Clock Synchronous Serial I/O Mode(1).

Table 15.1 Clock Synchronous Serial I/O Mode Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clocks	• CKDIR bit in U0MR register is set to 0 (internal clock): fi/(2(n+1)). fi = f1, f8, f32 n = value set in U0BRG register: 00h to FFh • The CKDIR bit is set to 1 (external clock): input from CLK0 pin.
Transmit start conditions	• Before transmission starts, the following requirements must be met.(1) - The TE bit in the U0C1 register is set to 1 (transmission enabled). - The TI bit in the U0C1 register is set to 0 (data in the U0TB register).
Receive start conditions	• Before reception starts, the following requirements must be met.(1) - The RE bit in the U0C1 register is set to 1 (reception enabled). - The TE bit in the U0C1 register is set to 1 (transmission enabled). - The TI bit in the U0C1 register is set to 0 (data in the U0TB register).
Interrupt request generation timing	• When transmitting, one of the following conditions can be selected. - The U0IRS bit is set to 0 (transmit buffer empty): When transferring data from the U0TB register to UART0 transmit register (when transmission starts). - The U0IRS bit is set to 1 (transmission completes): When completing data transmission from UARTi transmit register. • When receiving When data transfer from the UART0 receive register to the U0RB register (when reception completes).
Error detection	• Overrun error(2) This error occurs if the serial interface starts receiving the next data item before reading the U0RB register and receives the 7th bit of the next data.
Select functions	• CLK polarity selection Transfer data input/output can be selected to occur synchronously with the rising or the falling edge of the transfer clock. • LSB first, MSB first selection Whether transmitting or receiving data begins with bit 0 or begins with bit 7 can be selected. • Continuous receive mode selection. Receive is enabled immediately by reading the U0RB register.

NOTES:

1. The programming and erasure endurance is defined on a per-block basis. If the programming and erasure endurance is n (n = 100 or 10,000), each block can be erased n times. For example, if 1,024 1-byte writes are performed to block A, a 1-Kbyte block, and then the block is erased, the erase count stands at one. When performing 100 or more rewrites, the actual erase count can be reduced by executing programming operations in such a way that all blank areas are used before performing an erase operation. Avoid rewriting only particular blocks and try to average out the programming and erasure endurance of the blocks. It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.
2. If an overrun error occurs, the value of the U0RB register will be undefined. The IR bit in the S0RIC register remains unchanged.

Table 15.2 Registers Used and Settings in Clock Synchronous Serial I/O Mode[1]

Register	Bit	Function
U0TB	0 to 7	Set data transmission.
U0RB	0 to 7	Data reception can be read.
	OER	Overrun error flag
U0BRG	0 to 7	Set bit rate.
U0MR	SMD2 to SMD0	Set to 001b.
	CKDIR	Select the internal clock or external clock.
U0C0	CLK1 to CLK0	Select the count source in the U0BRG register.
	TXEPT	Transmit register empty flag
	NCH	Select TXD0 pin output mode.
	CKPOL	Select the transfer clock polarity.
	UFORM	Select the LSB first or MSB first.
U0C1	TE	Set this bit to 1 to enable transmission/reception.
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception.
	RI	Reception complete flag
UCON	U0IRS	Select the UART0 transmit interrupt source.
	U0RRM	Set this bit to 1 to use continuous receive mode.
	CNTRSEL	Set this bit to 1 to select P1_5/RXD0/CNTR01/$\overline{\text{INT11}}$.

NOTE:

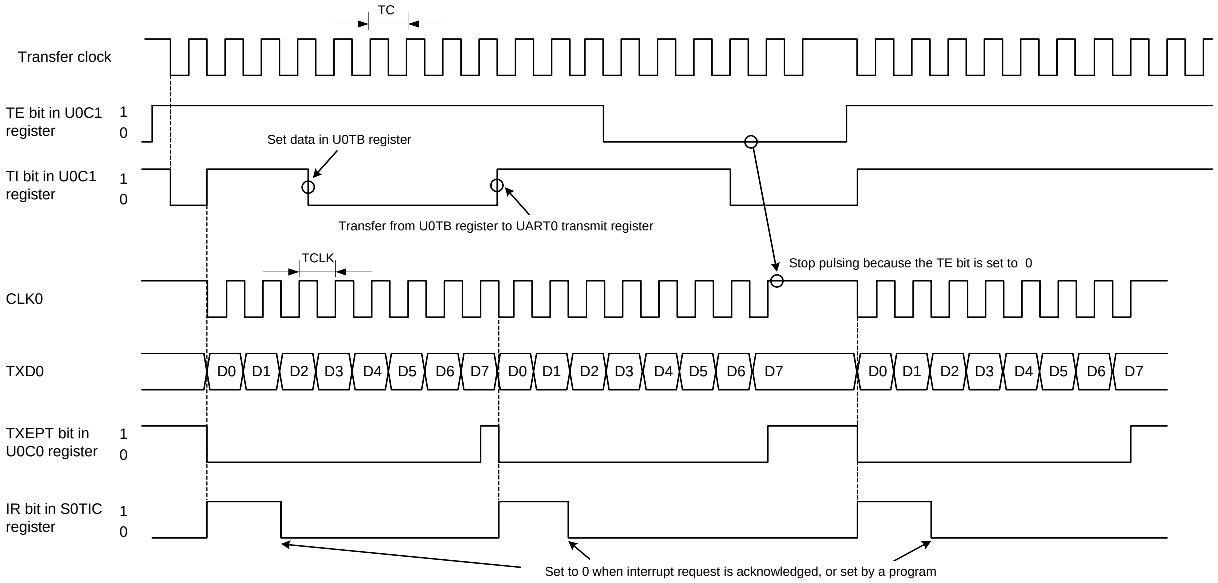
1. Set bits which are not in this table to 0 when writing to the above registers in clock synchronous serial I/O mode.

Table 15.3 lists the I/O Pin Functions in Clock Synchronous Serial I/O Mode. The TXD0 pin outputs “H” level between the operating mode selection of UART0 and transfer start. (If the NCH bit is set to 1 (N-channel open-drain output), this pin is in a high-impedance state.)

Table 15.3 I/O Pin Functions in Clock Synchronous Serial I/O Mode

Pin Name	Function	Selection Method
TXD0 (P1_4)	Output serial data	(Outputs dummy data when performing reception only.)
RXD0 (P1_5)	Input serial data	PD1_5 bit in PD1 register = 0 (P1_5 can be used as an input port when performing transmission only.)
CLK0 (P1_6)	Output transfer clock	CKDIR bit in U0MR register = 0
	Input transfer clock	CKDIR bit in U0MR register = 1 PD1_6 bit in PD1 register = 0

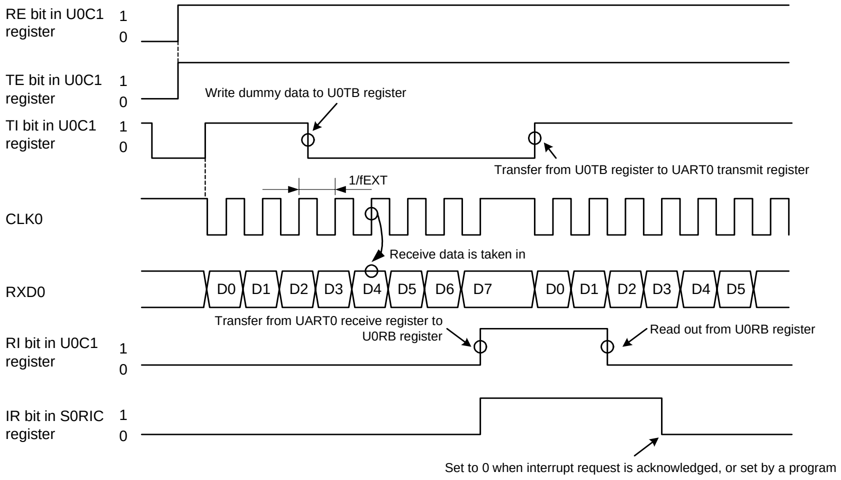
• Example of transmit timing (when internal clock is selected)


TC
Transfer clock
TE bit in U0C1 register
1
0
Set data in U0TB register
TI bit in U0C1 register
1
0
Transfer from U0TB register to UART0 transmit register
TCLK
Stop pulsing because the TE bit is set to 0
CLK0
TXD0
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7
TXEPT bit in U0C0 register
1
0
IR bit in S0TIC register
1
0
Set to 0 when interrupt request is acknowledged, or set by a program

TC=TCLK=2(n+1)/fi
fi: Frequency of U0BRG count source (f1, f8, f32)
n: Setting value to U0BRG register

The above applies under the following settings:
- CKDIR bit in U0MR register = 0 (internal clock)
- CKPOL bit in U0C0 register = 0 (output transmit data at the falling edge and input receive data at the rising edge of the transfer clock)
- U0IRS bit in UCON register = 0 (an interrupt request is generated when the transmit buffer is empty)

• Example of receive timing (when external clock is selected)


RE bit in U0C1 register
1
0
TE bit in U0C1 register
1
0
Write dummy data to U0TB register
TI bit in U0C1 register
1
0
Transfer from U0TB register to UART0 transmit register
1/fEXT
CLK0
Receive data is taken in
RXD0
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5
Transfer from UART0 receive register to U0RB register
Read out from U0RB register
RI bit in U0C1 register
1
0
IR bit in S0RIC register
1
0
Set to 0 when interrupt request is acknowledged, or set by a program

The above applies under the following settings:
- CKDIR bit in U0MR register = 1 (external clock)
- CKPOL bit in U0C0 register = 0 (output transmit data at the falling edge and input receive data at the rising edge of the transfer clock)

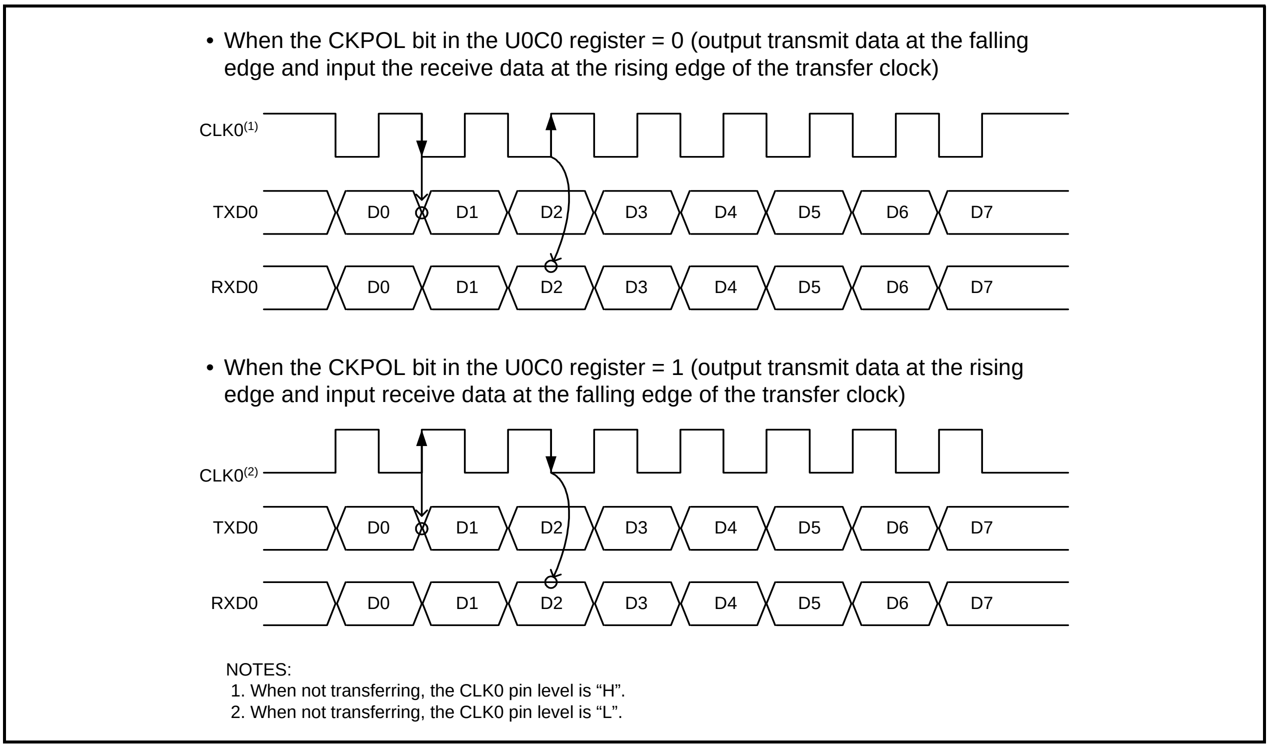
The following conditions are met when “H” is applied to the CLK0 pin before receiving data:
- TE bit in U0C1 register = 1 (enables transmit)
- RE bit in U0C1 register = 1 (enables receive)
- Write dummy data to the U0TB register

fEXT: Frequency of external clock

Figure 15.7 Transmit and Receive Timing Example in Clock Synchronous Serial I/O Mode

15.1.1 Polarity Select Function

Figure 15.8 shows the Transfer Clock Polarity. Use the CKPOL bit in the U0C0 register to select the transfer clock polarity.



- When the CKPOL bit in the U0C0 register = 0 (output transmit data at the falling edge and input the receive data at the rising edge of the transfer clock)

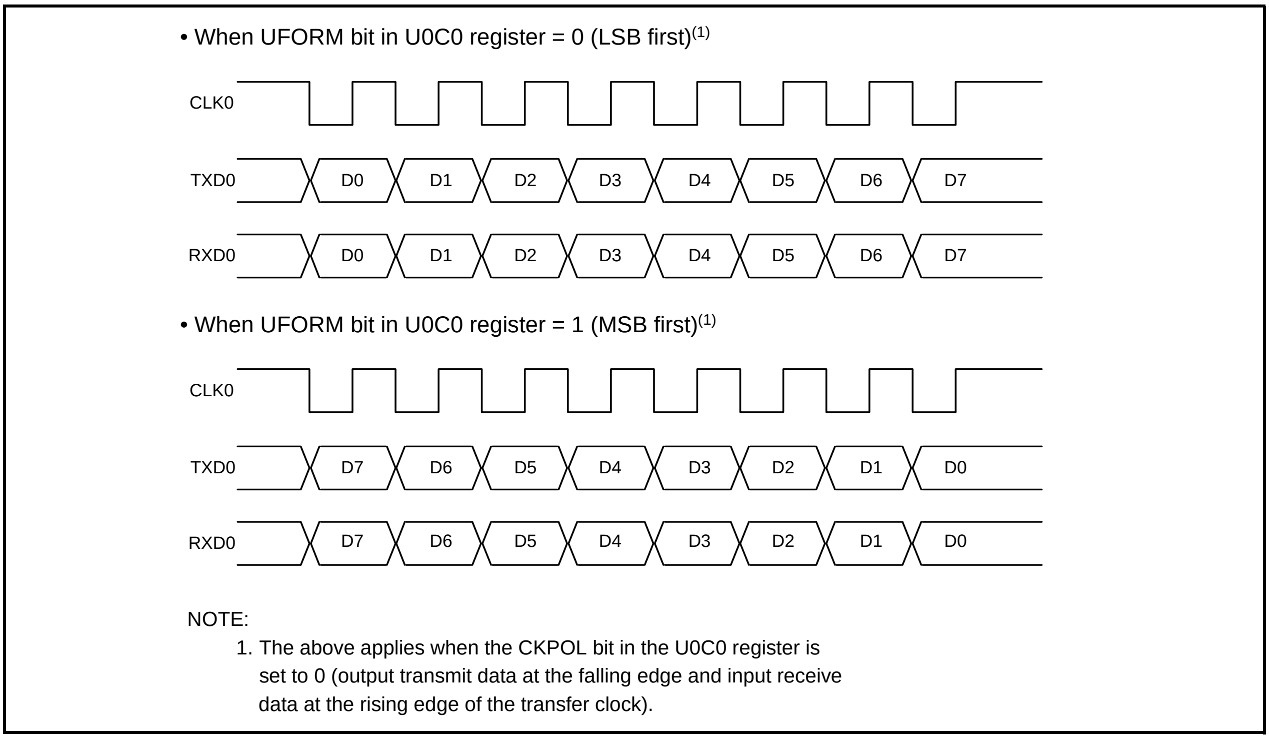
- When the CKPOL bit in the U0C0 register = 1 (output transmit data at the rising edge and input receive data at the falling edge of the transfer clock)

NOTES:
1. When not transferring, the CLK0 pin level is “H”.
2. When not transferring, the CLK0 pin level is “L”.

Figure 15.8 Transfer Clock Polarity

15.1.2 LSB First/MSB First Select Function

Figure 15.9 shows the Transfer Format. Use the UFORM bit in the U0C0 register to select the transfer format.



- When UFORM bit in U0C0 register = 0 (LSB first)[1]

- When UFORM bit in U0C0 register = 1 (MSB first)[1]

NOTE:
1. The above applies when the CKPOL bit in the U0C0 register is set to 0 (output transmit data at the falling edge and input receive data at the rising edge of the transfer clock).

Figure 15.9 Transfer Format

15.1.3 Continuous Receive Mode

Continuous receive mode is selected by setting the U0RRM bit in the UCON register to 1 (enables continuous receive mode). In this mode, reading the U0RB register sets the TI bit in the U0C1 register to 0 (data in the U0TB register). When the U0RRM bit is set to 1, do not write dummy data to the U0TB register by a program.

15.2 Clock Asynchronous Serial I/O (UART) Mode

The UART mode allows data transmission and reception after setting the desired bit rate and transfer data format. Table 15.4 lists the UART Mode Specifications. Table 15.5 lists the Registers Used and Settings for UART Mode.

Table 15.4 UART Mode Specifications

Item	Specification
Transfer data format	• Character bit (transfer data): Selectable among 7, 8, or 9 bits • Start bit: 1 bit • Parity bit: Selectable among odd, even, or none • Stop bit: Selectable among 1 or 2 bits
Transfer clocks	• CKDIR bit in UiMR register is set to 0 (internal clock): fj/(16(n+1)) fj = f1, f8, f32 n = value set in UiBRG register: 00h to FFh • CKDIR bit is set to 1 (external clock): fEXT/(16(n+1)) fEXT: input from CLKi pin n = value set in UiBRG register: 00h to FFh
Transmit start conditions	• Before transmission starts, the following are required. - TE bit in UiC1 register is set to 1 (transmission enabled). - TI bit in UiC1 register is set to 0 (data in UiTB register).
Receive start conditions	• Before reception starts, the following are required. - RE bit in UiC1 register is set to 1 (reception enabled). - Start bit deleted
Interrupt request generation timing	• When transmitting, one of the following conditions can be selected. - UiIRS bit is set to 0 (transmit buffer empty): When transferring data from the UiTB register to UARTi transmit register (when transmit starts). - UiIRS bit is set to 1 (transfer ends): When serial interface completes transmitting data from the UARTi transmit register. • When receiving When transferring data from the UARTi receive register to UiRB register (when receive ends).
Error detection	• Overrun error(1) This error occurs if the serial interface starts receiving the next data item before reading the UiRB register and receives the bit preceding the final stop bit of the next data item. • Framing error This error occurs when the set number of stop bits is not detected. • Parity error This error occurs when parity is enabled, and the number of 1's in parity and character bits do not match the number of 1's set. • Error sum flag This flag is set is set to 1 when an overrun, framing, or parity error is generated.

i = 0 to 1

NOTE:

1. If an overrun error occurs, the contents of the UiRB register will be undefined. The IR bit in the SiRIC register remains unchanged.

Table 15.5 Registers Used and Settings for UART Mode

Register	Bit	Function
UiTB	0 to 8	Set transmit data[1].
UiRB	0 to 8	Receive data can be read[1].
	OER,FER,PER,SUM	Error flag
UiBRG	0 to 7	Set a bit rate.
UiMR	SMD2 to SMD0	Set to 100b when transfer data is 7 bits long. Set to 101b when transfer data is 8 bits long. Set to 110b when transfer data is 9 bits long.
	CKDIR	Select the internal clock or external clock.[2]
	STPS	Select the stop bit.
	PRY, PRYE	Select whether parity is included and whether odd or even.
UiC0	CLK0, CLK1	Select the count source for the UiBRG register.
	TXEPT	Transmit register empty flag
	NCH	Select TXDi pin output mode.
	CKPOL	Set to 0.
	UFORM	LSB first or MSB first can be selected when transfer data is 8 bits long. Set to 0 when transfer data is 7 or 9 bits long.
UiC1	TE	Set to 1 to enable transmit.
	TI	Transmit buffer empty flag
	RE	Set to 1 to enable receive.
	RI	Receive complete flag
UCON	U0IRS, U1IRS	Select the source of UART0 transmit interrupt.
	U0RRM	Set to 0.
	CNTRSEL	Set to 1 to select P1_5/RXD0/CNTR01/$\overline{\text{INT11}}$.

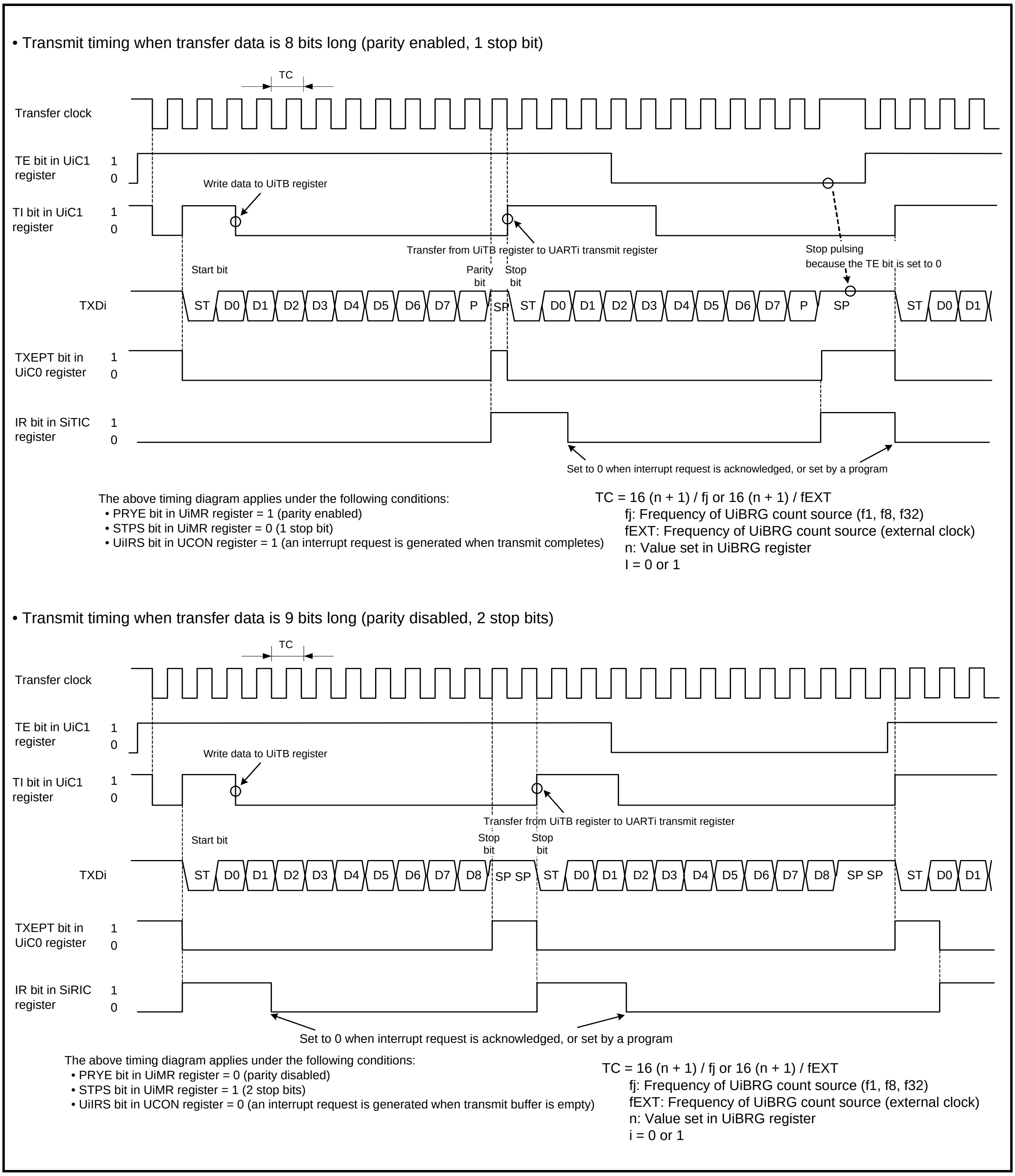
NOTES:

1. The bits used for transmit/receive data are as follows: Bits 0 to 6 when transfer data is 7 bits long; bits 0 to 7 when transfer data is 8 bits long; bits 0 to 8 when transfer data is 9 bits long.
2. An external clock can be selected in UART0 only.

Table 15.6 lists the I/O Pin Functions in Clock Asynchronous Serial I/O Mode. The TXDi pin outputs “H” level between the operating mode selection of UARTi (i = 0 or 1) and transfer start. (If the NCH bit is set to 1 (N-channel open-drain output), this pin is in a high-impedance state.)

Table 15.6 I/O Pin Functions in Clock Asynchronous Serial I/O Mode

Pin Name	Function	Selection Method
TXD0 (P1_4)	Output serial data	(Cannot be used as a port when performing reception only.)
RXD0 (P1_5)	Input serial data	PD1_5 bit in PD1 register = 0 (P1_5 can be used as an input port when performing transmission only.)
CLK0 (P1_6)	Programmable I/O port	CKDIR bit in U0MR register = 0
	Input transfer clock	CKDIR bit in U0MR register = 1 PD1_6 bit in PD1 register = 0
TXD1 (P3_7)	Output serial data	Bits U1SEL1 to U1SEL0 in UCON register = 11b (P3_7 can be used as a port when bits U1SEL1 to U1SEL0 = 01b and performing reception only.)
RXD1 (P4_5)	Input serial data	PD4_5 bit in PD4 register = 0 Bits U1SEL1 to U1SEL0 in UCON register = 01b or 11b (Cannot be used as a port when performing transmission only.)


• Transmit timing when transfer data is 8 bits long (parity enabled, 1 stop bit)
TC
Transfer clock
TE bit in UiC1 register
1
0
Write data to UiTB register
TI bit in UiC1 register
1
0
Transfer from UiTB register to UARTi transmit register
Stop pulsing because the TE bit is set to 0
Start bit
Parity bit
Stop bit
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
D1
TXEPT bit in UiC0 register
1
0
IR bit in SiTIC register
1
0
Set to 0 when interrupt request is acknowledged, or set by a program
• Transmit timing when transfer data is 9 bits long (parity disabled, 2 stop bits)
TC
Transfer clock
TE bit in UiC1 register
1
0
Write data to UiTB register
TI bit in UiC1 register
1
0
Transfer from UiTB register to UARTi transmit register
Start bit
Stop bit
Stop bit
TXDi
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
D1
TXEPT bit in UiC0 register
1
0
IR bit in SiRIC register
1
0
Set to 0 when interrupt request is acknowledged, or set by a program

The above timing diagram applies under the following conditions:
- PRYE bit in UiMR register = 1 (parity enabled)
- STPS bit in UiMR register = 0 (1 stop bit)
- UiIRS bit in UCON register = 1 (an interrupt request is generated when transmit completes)

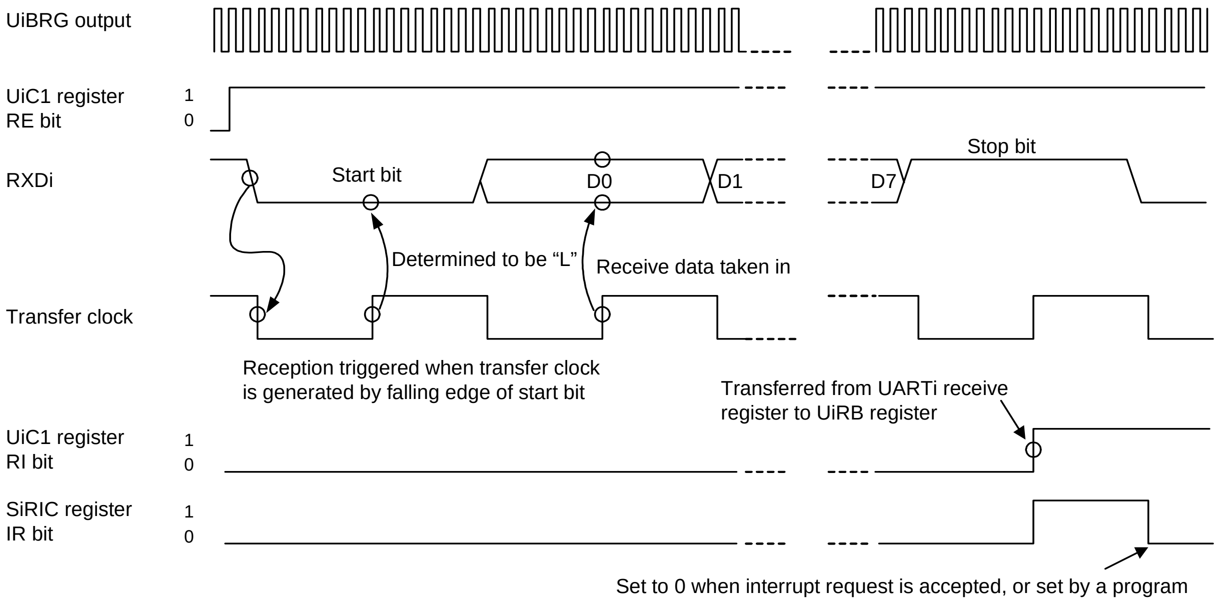
TC = 16 (n + 1) / fj or 16 (n + 1) / fEXT
fj: Frequency of UiBRG count source (f1, f8, f32)
fEXT: Frequency of UiBRG count source (external clock)
n: Value set in UiBRG register
I = 0 or 1

The above timing diagram applies under the following conditions:
- PRYE bit in UiMR register = 0 (parity disabled)
- STPS bit in UiMR register = 1 (2 stop bits)
- UiIRS bit in UCON register = 0 (an interrupt request is generated when transmit buffer is empty)

TC = 16 (n + 1) / fj or 16 (n + 1) / fEXT
fj: Frequency of UiBRG count source (f1, f8, f32)
fEXT: Frequency of UiBRG count source (external clock)
n: Value set in UiBRG register
i = 0 or 1

Figure 15.10 Transmit Timing in UART Mode

• Example of receive timing when transfer data is 8 bits long (parity disabled, one stop bit)


UiBRG output
UiC1 register
RE bit
1
0
RXDi
Start bit
D0
D1
D7
Stop bit
Determined to be "L"
Receive data taken in
Transfer clock
Reception triggered when transfer clock
is generated by falling edge of start bit
Transferred from UARTi receive
register to UiRB register
UiC1 register
RI bit
1
0
SiRIC register
IR bit
1
0
Set to 0 when interrupt request is accepted, or set by a program

The above timing diagram applies when the register bits are set as follows:
- PRYE bit in UiMR register = 0 (parity disabled)
- STPS bit in UiMR register = 0 (1 stop bit)

i = 0 or 1

Figure 15.11 Receive Timing in UART Mode

15.2.1 CNTR0 Pin Select Function

The CNTRSEL bit in the UCON register selects whether P1_7 is used as the CNTR00/$\overline{\text{INT10}}$ input pin or P1_5 is used as the CNTR01/$\overline{\text{INT11}}$ input pin.

When the CNTRSEL bit is set to 0, P1_7 is used as the CNTR00/$\overline{\text{INT10}}$ pin and when the CNTRSEL bit is set to 1, P1_5 is used as the CNTR01/$\overline{\text{INT11}}$ pin.

15.2.2 Bit Rate

In UART mode, the bit rate is the frequency divided by the UiBRG (i = 0 or 1) register.

UART Mode

- Internal clock selected

UiBRG register setting value = $\dfrac{fj}{\text{Bit Rate} \times 16} - 1$

Fj: Count source frequency of the UiBRG register (f1, f8, or f32)

- External clock selected

UiBRG register setting value = $\dfrac{fEXT}{\text{Bit Rate} \times 16} - 1$

fEXT: Count source frequency of the UiBRG register (external clock)

i = 0 or 1

Figure 15.12 Calculation Formula of UiBRG (i = 0 or 1) Register Setting Value

Table 15.7 Bit Rate Setting Example in UART Mode (Internal Clock Selected)

Bit Rate (bps)	BRG Count Source	System Clock = 20 MHz			System Clock = 8 MHz		
		UiBRG Setting Value	Actual Time (bps)	Error (%)	UiBRG Setting Value	Actual Time (bps)	Error (%)
1200	f8	129(81h)	1201.92	0.16	51(33h)	1201.92	0.16
2400	f8	64(40h)	2403.85	0.16	25(19h)	2403.85	0.16
4800	f8	32(20h)	4734.85	-1.36	12(0Ch)	4807.69	0.16
9600	f1	129(81h)	9615.38	0.16	51(33h)	9615.38	0.16
14400	f1	86(56h)	14367.82	-0.22	34(22h)	14285.71	-0.79
19200	f1	64(40h)	19230.77	0.16	25(19h)	19230.77	0.16
28800	f1	42(2Ah)	29069.77	0.94	16(10h)	29411.76	2.12
31250	f1	39(27h)	31250.00	0.00	15(0Fh)	31250.00	0.00
38400	f1	32(20h)	37878.79	-1.36	12(0Ch)	38461.54	0.16
51200	f1	23(17h)	52083.33	1.73	9(09h)	50000.00	-2.34

i = 0 or 1

15.3 Notes on Serial Interface

- When reading data from the U0RB register either in the clock asynchronous serial I/O mode or in the clock synchronous serial I/O mode. Ensure the data is read in 16-bit units. When the high-order byte of the U0RB register is read, bits PER and FER in the U0RB register and the RI bit in the U0C1 register are set to 0.

Example (when reading receive buffer register):

```
    MOV.W       00A6H,R0        ; Read the U0RB register
```

- When writing data to the U0TB register in the clock asynchronous serial I/O mode with 9-bit transfer data length, write data to the high-order byte first then the low-order byte, in 8-bit units.

Example (when reading transmit buffer register):

```
    MOV.B       #XXH,00A3H    ; Write the high-order byte of U0TB register
    MOV.B       #XXH,00A2H    ; Write the low-order byte of U0TB register
```

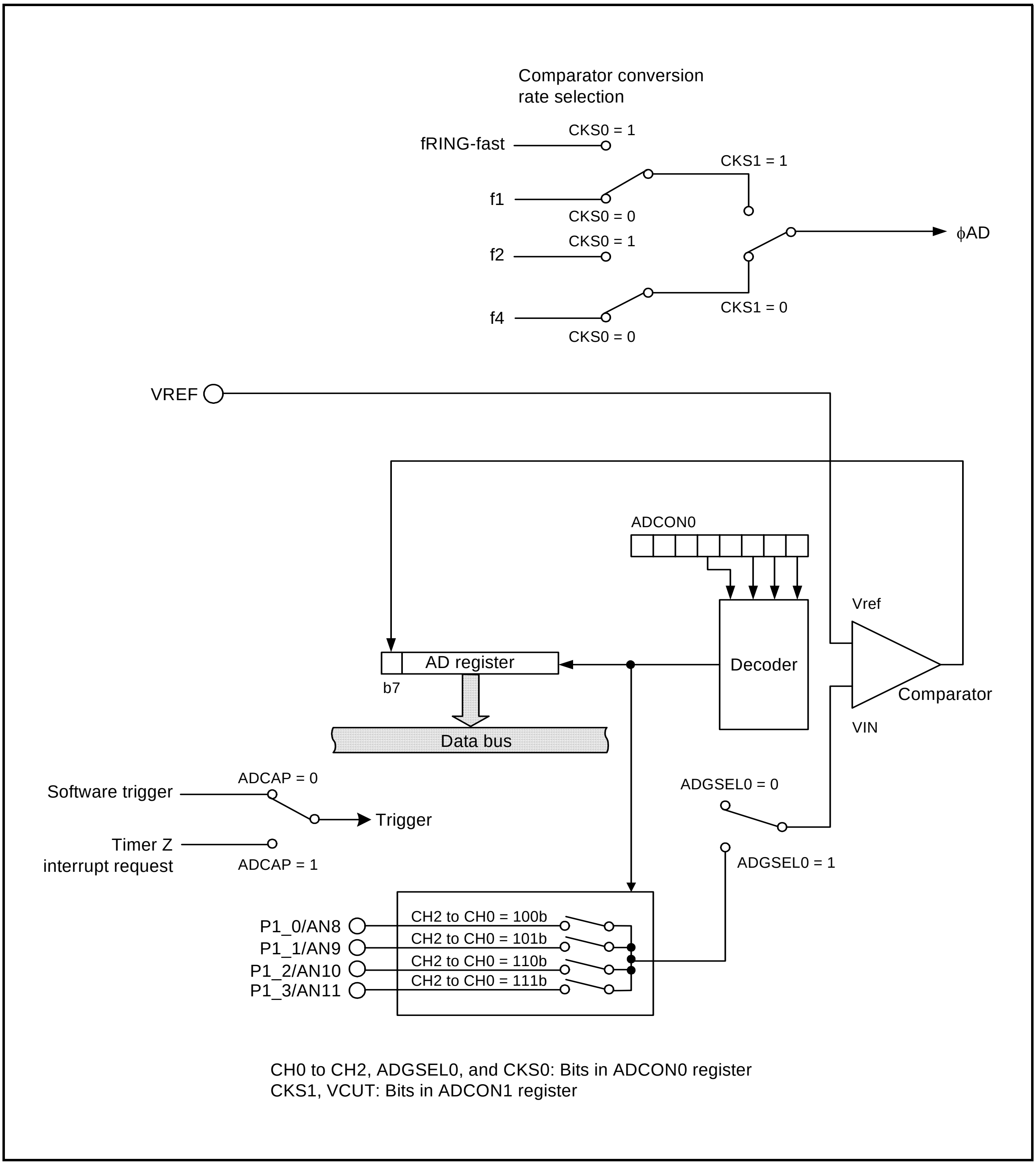
16. Comparator

The comparator compares the electric potential input from the VREF pin with analog input.
The analog input shares pins P1_0 to P1_3. Therefore, when using these pins, ensure the corresponding port direction bits are set to 0 (input mode).
The result of comparator conversion is stored in the AD register.
Table 16.1 lists the Comparator Performance. Figure 16.1 shows a Comparator Block Diagram.
Figures 16.2 and 16.3 show the Associated Comparator Registers.

Table 16.1 Comparator Performance

Item	Performance
Comparator conversion method	Comparator
Analog input voltage	0 V to AVCC
Operating clock φAD(1)	4.2 V ≤ AVCC ≤ 5.5 V fRING-fast, f1, f2, f4 2.7 V ≤ AVCC < 4.2 V f2, f4
Absolute accuracy	AVCC = 2.7 to 5.5 V ± 20 mV
Operating mode	One-shot and repeat modes
Analog input pin	4 pins (AN8 to AN11)
Comparator conversion start conditions	• Software trigger Set the ADST bit in the ADCON0 register to 1 (comparator conversion starts). • Capture A timer Z interrupt request is generated while the ADST bit is set to 1.
Conversion rate per pin	10φAD cycles

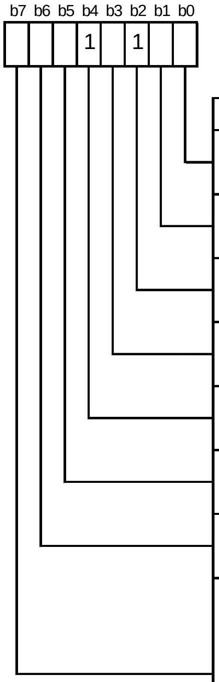
NOTE:
1. The φAD frequency must be 10 MHz or below.



CH0 to CH2, ADGSEL0, and CKS0: Bits in ADCON0 register
CKS1, VCUT: Bits in ADCON1 register

Figure 16.1 Comparator Block Diagram

A/D Control Register 0[1]



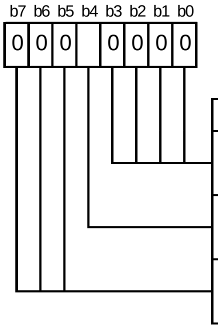
Symbol	Address	After Reset
ADCON0	00D6h	00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bits[2]	b2 b1 b0 1 0 0 : AN8 1 0 1 : AN9 1 1 0 : AN10 1 1 1 : AN11 Other than above : Do not set.	RW
CH1			RW
CH2			RW
MD	Comparator conversion operating mode select bit[3]	0 : On-shot mode 1 : Repeat mode	RW
ADGSEL0	Analog input group select bit[5]	0 : Disabled 1 : Enabled (AN8 to AN11)	RW
ADCAP	Comparator conversion automatic start bit	0 : Starts at software trigger (ADST bit). 1 : Starts at capture (requests timer Z interrupt).	RW
ADST	Comparator conversion start flag	0 : Disables comparator conversion. 1 : Starts comparator conversion.	RW
CKS0	Frequency select bit 0	[When CKS1 in ADCON1 register = 0] 0 : Selects f4. 1 : Selects f2. [When CKS1 in ADCON1 register = 1] 0 : Selects f1.[4] 1 : fRING-fast	RW

NOTES:

1. If the ADCON0 register is rewritten during comparator conversion, the conversion result is undefined.
2. Bits CH0 to CH2 are enabled when the ADGSEL0 bit is set to 1. After setting the ADGSEL0 bit to 1, write to bits CH0 to CH2.
3. When changing comparator conversion operating mode, set the analog input pin again.
4. Set the øAD frequency to 10 MHz or below.
5. To use the comparator, set the ADGSEL0 bit to 1.

A/D Control Register 1[1]



Symbol	Address	After Reset
ADCON1	00D7h	00h

Bit Symbol	Bit Name	Function	RW
— (b3-b0)	Reserved bits	Set to 0.	RW
CKS1	Frequency select bit 1	Refer to the description of the CKS0 bit in the ADCON0 register function	RW
— (b7-b5)	Reserved bits	Set to 0.	RW

NOTE:

1. If the ADCON1 register is rewritten during comparator conversion, the conversion result is undefined.

Figure 16.2 Registers ADCON0 and ADCON1

A/D Control Register 2[1]

b7	b6	b5	b4	b3	b2	b1	b0
☒	☒	☒	☒	0	0		0

Symbol	Address	After Reset
ADCON2	00D4h	00h

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved bit	Set to 0.	RW
CMPSEL	Comparator function select bit	0 : Not used 1 : Used	RW
— (b3-b2)	Reserved bits	Set to 0.	RW
— (b7-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		—

NOTE:

1. If the ADCON2 register is rewritten during comparator conversion, the conversion result is undefined.

A/D Register

b7	b6	b5	b4	b3	b2	b1	b0
	☒	☒	☒	☒	☒	☒	☒

Symbol	Address	After Reset
AD	00C0h	Undefined

Function	RW
Nothing is assigned. If necessary, set to 0. When read, the content is 0.	—
Comparator conversion result	RO

Figure 16.3 Registers ADCON2 and AD

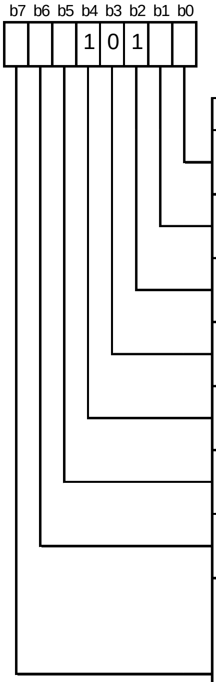
16.1 One-Shot Mode

In one-shot mode, the input voltage on one selected pin is comparator converted once. Table 16.2 lists the One-Shot Mode Specifications. Figure 16.4 shows Registers ADCON0 and ADCON1 in One-Shot Mode.

Table 16.2 One-Shot Mode Specifications

Item	Specification
Function	The input voltage on one pin selected by bits CH2 to CH0 is comparator converted once.
Start conditions	• When the ADCAP bit is set to 0 (software trigger), set the ADST bit to 1 (comparator conversion starts). • When the ADCAP bit is set to 1 (capture), a timer Z interrupt request is generated while the ADST bit is set to 1.
Stop conditions	• Comparator conversion is completed (the ADST bit is set to 0). • The ADST bit is set to 0.
Interrupt request generation timing	Comparator conversion completed
Input pin	Select one of AN8 to AN11
Reading of comparator conversion result	Read AD register

A/D Control Register 0[1]



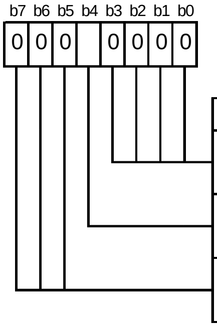
Symbol	Address	After Reset
ADCON0	00D6h	00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bits[2]	b2 b1 b0 1 0 0 : AN8 1 0 1 : AN9 1 1 0 : AN10 1 1 1 : AN11 Other than above : Do not set.	RW
CH1			RW
CH2			RW
MD	Comparator conversion operating mode select bit[3]	0 : One-shot mode	RW
ADGSEL0	Analog input group select bit[5]	0 : Disabled 1 : Enabled (AN8 to AN11)	RW
ADCAP	Comparator conversion automatic start bit	0 : Starts at softw are trigger (ADST bit). 1 : Starts at capture (requests timer Z interrupt).	RW
ADST	Comparator conversion start flag	0 : Disables comparator conversion. 1 : Starts comparator conversion.	RW
CKS0	Frequency select bit 0	[When CKS1 in ADCON1 register = 0] 0 : Selects f4. 1 : Selects f2. [When CKS1 in ADCON1 register = 1] 0 : Selects f1.[4] 1 : fRING-fast	RW

NOTES:

1. If the ADCON0 register is rew ritten during comparator conversion, the conversion result is undefined.
2. Bits CH0 to CH2 are enabled w hen the ADGSEL0 bit is set to 1. After setting the ADGSEL0 bit to 1, w rite to bits CH0 to CH2.
3. When changing comparator conversion operating mode, set the analog input pin again.
4. Set the øAD frequency to 10 MHz or below .
5. To use the comparator, set the ADGSEL0 bit to 1.

A/D Control Register 1[1]



Symbol	Address	After Reset
ADCON1	00D7h	00h

Bit Symbol	Bit Name	Function	RW
— (b3-b0)	Reserved bits	Set to 0.	RW
CKS1	Frequency select bit 1	Refer to the description of the CKS0 bit in the ADCON0 register function	RW
— (b7-b5)	Reserved bits	Set to 0.	RW

NOTE:

1. If the ADCON1 register is rew ritten during comparator conversion, the conversion result is undefined.

Figure 16.4 Registers ADCON0 and ADCON1 in One-Shot Mode

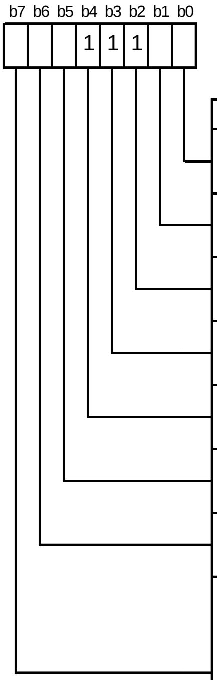
16.2 Repeat Mode

In repeat mode, the input voltage on one selected pin is comparator converted repeatedly. Table 16.3 lists the Repeat Mode Specifications. Figure 16.5 shows Registers ADCON0 and ADCON1 in Repeat Mode.

Table 16.3 Repeat Mode Specifications

Item	Specification
Function	The Input voltage on one pin selected by bits CH2 to CH0, and the ADGSEL0 bit is comparator converted repeatedly
Start conditions	• When the ADCAP bit is set to 0 (software trigger), set the ADST bit to 1 (comparator conversion starts). • When the ADCAP bit is set to 1 (capture), a timer Z interrupt request is generated while the ADST bit is set to 1.
Stop condition	Set the ADST bit to 0.
Interrupt request generation timing	Not generated
Input pin	Selects one of AN8 to AN11.
Reading of result of comparator conversion	Read AD register.

A/D Control Register 0[1]



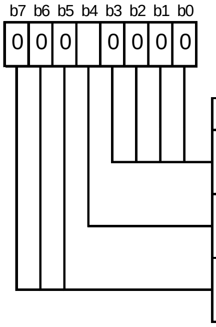
Symbol	Address	After Reset
ADCON0	00D6h	00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bits[2]	b2 b1 b0 1 0 0 : AN8 1 0 1 : AN9 1 1 0 : AN10 1 1 1 : AN11 Other than above : Do not set.	RW
CH1			RW
CH2			RW
MD	Comparator conversion operating mode select bit[3]	1 : Repeat mode	RW
ADGSEL0	Analog input group select bit[5]	0 : Disabled 1 : Enabled (AN8 to AN11)	RW
ADCAP	Comparator conversion automatic start bit	0 : Starts at softw are trigger (ADST bit). 1 : Starts at capture (requests timer Z interrupt).	RW
ADST	Comparator conversion start flag	0 : Disables comparator conversion. 1 : Starts comparator conversion.	RW
CKS0	Frequency select bit 0	[When CKS1 in ADCON1 register = 0] 0 : Selects f4. 1 : Selects f2. [When CKS1 in ADCON1 register = 1] 0 : Selects f1.[4] 1 : fRING-fast	RW

NOTES:

1. If the ADCON0 register is rew ritten during comparator conversion, the conversion result is undefined.
2. Bits CH0 to CH2 are enabled w hen the ADGSEL0 bit is set to 1. After setting the ADGSEL0 bit to 1, w rite to bits CH0 to CH2.
3. When changing comparator conversion operating mode, set the analog input pin again.
4. Set the øAD frequency to 10 MHz or below .
5. To use the comparator, set the ADGSEL0 bit to 1.

A/D Control Register 1[1]



Symbol	Address	After Reset
ADCON1	00D7h	00h

Bit Symbol	Bit Name	Function	RW
— (b3-b0)	Reserved bits	Set to 0.	RW
CKS1	Frequency select bit 1	Refer to the description of the CKS0 bit in the ADCON0 register function.	RW
— (b7-b5)	Reserved bits	Set to 0.	RW

NOTE:

1. If the ADCON1 register is rew ritten during comparator conversion, the conversion result is undefined.

Figure 16.5 Registers ADCON0 and ADCON1 in Repeat Mode

16.3 Notes on Comparator

- Write to each bit (other than bit 6) in the ADCON0 register, each bit in the ADCON1 register, or the CMPSEL bit in the ADCON2 register when the comparator conversion stops (before a trigger occurs).
- When changing comparator conversion operating mode, select an analog input pin again.
- To use in one-shot mode, ensure that the comparator conversion is completed and the AD register is read. The IR bit in the ADIC register or the ADST bit in the ADCON0 register can determine whether the comparator conversion is completed.
- To use in repeat mode, use the undivided main clock as the CPU clock.
- If the ADST bit in the ADCON0 register is set to 0 (comparator conversion stops) by a program and the comparator conversion is forcibly terminated during the comparator conversion operation, the conversion result of the comparator will be indeterminate. If the ADST bit is set to 0 by a program, do not use the AD register value.
- Connect a 0.1 µF capacitor between the VCC/AVCC pin and VSS/AVSS pin.

17. Flash Memory Version

17.1 Overview

In the flash memory version, rewrite operations to the flash memory can be performed in three modes; CPU rewrite, standard serial I/O, and parallel I/O.

Table 17.1 lists the Flash Memory Version Performance (refer to **Table 1.1 Functions and Specifications for R8C/18 Group** and **Table 1.2 Functions and Specifications for R8C/19 Group** for items not listed in Table 17.1).

Table 17.1 Flash Memory Version Performance

Item		Specification
Flash memory operating mode		3 modes (CPU rewrite, standard serial I/O, and parallel I/O mode)
Division of erase block		Refer to **Figure 17.1** and **Figure 17.2**
Programming method		Byte unit
Erase method		Block erase
Programming and erasure control method		Program and erase control by software command
Rewrite control method		Rewrite control for blocks 0 and 1 by FMR02 bit in FMR0 register.
		Rewrite control for block 0 by FMR15 bit and block 1 by FMR16 bit in FMR1 register.
Number of commands		5 commands
Programming and erasure endurance(1)	Blocks 0 and 1 (program ROM)	R8C/18 Group: 100 times; R8C/19 Group: 1,000 times
	Blocks A and B (data flash)(2)	10,000 times
ID code check function		Standard serial I/O mode supported
ROM code protect		Parallel I/O mode supported

NOTES:

1. Definition of programming and erasure endurance
   The programming and erasure endurance is defined on a per-block basis. If the programming and erasure endurance is n (n = 100 or 10,000), each block can be erased n times. For example, if 1,024 1-byte writes are performed to block A, a 1-Kbyte block, and then the block is erased, the erase count stands at one. When performing 100 or more rewrites, the actual erase count can be reduced by executing programming operations in such a way that all blank areas are used before performing an erase operation. Avoid rewriting only particular blocks and try to average out the programming and erasure endurance of the blocks. It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.
2. Blocks A and B are implemented only in the R8C/19 Group.

Table 17.2 Flash Memory Rewrite Modes

Flash Memory Rewrite Mode	CPU Rewrite Mode	Standard Serial I/O Mode	Parallel I/O Mode
Function	User ROM area is rewritten by executing software commands from the CPU. EW0 mode: Rewritable in any area other than flash memory EW1 mode: Rewritable in flash memory	User ROM area is rewritten by a dedicated serial programmer.	User ROM area is rewritten by a dedicated parallel programmer.
Areas which can be rewritten	User ROM area	User ROM area	User ROM area
Operating mode	Single chip mode	Boot mode	Parallel I/O mode
ROM programmer	None	Serial programmer	Parallel programmer

17.2 Memory Map

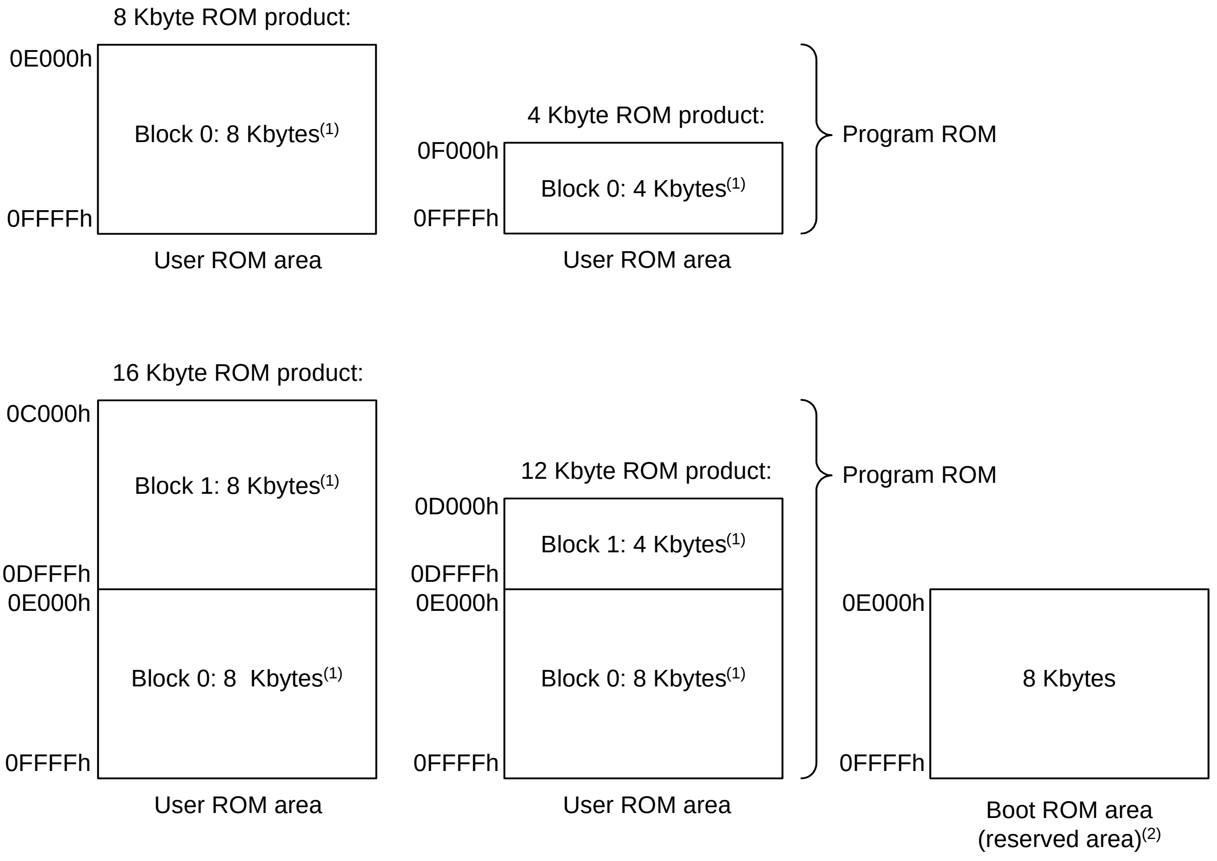
The flash memory contains a user ROM area and a boot ROM area (reserved area). Figure 17.1 shows a Flash Memory Block Diagram for R8C/18 Group. Figure 17.2 shows a Flash Memory Block Diagram for R8C/19 Group.

The user ROM area of the R8C/19 Group contains an area (program ROM) which stores MCU operating programs and the blocks A and B (data flash) each 1 byte in size.

The user ROM area is divided into several blocks. The user ROM area can be rewritten in CPU rewrite mode and standard serial I/O and parallel I/O modes.

When rewriting blocks 0 and 1 in CPU rewrite mode, set the FMR02 bit in the FMR0 register to 1 (rewrite enabled). When the FMR15 bit in the FMR1 register is set to 0 (rewrite enabled), block 0 is rewritable. When the FMR16 bit is set to 0 (rewrite enabled), block 1 is rewritable.

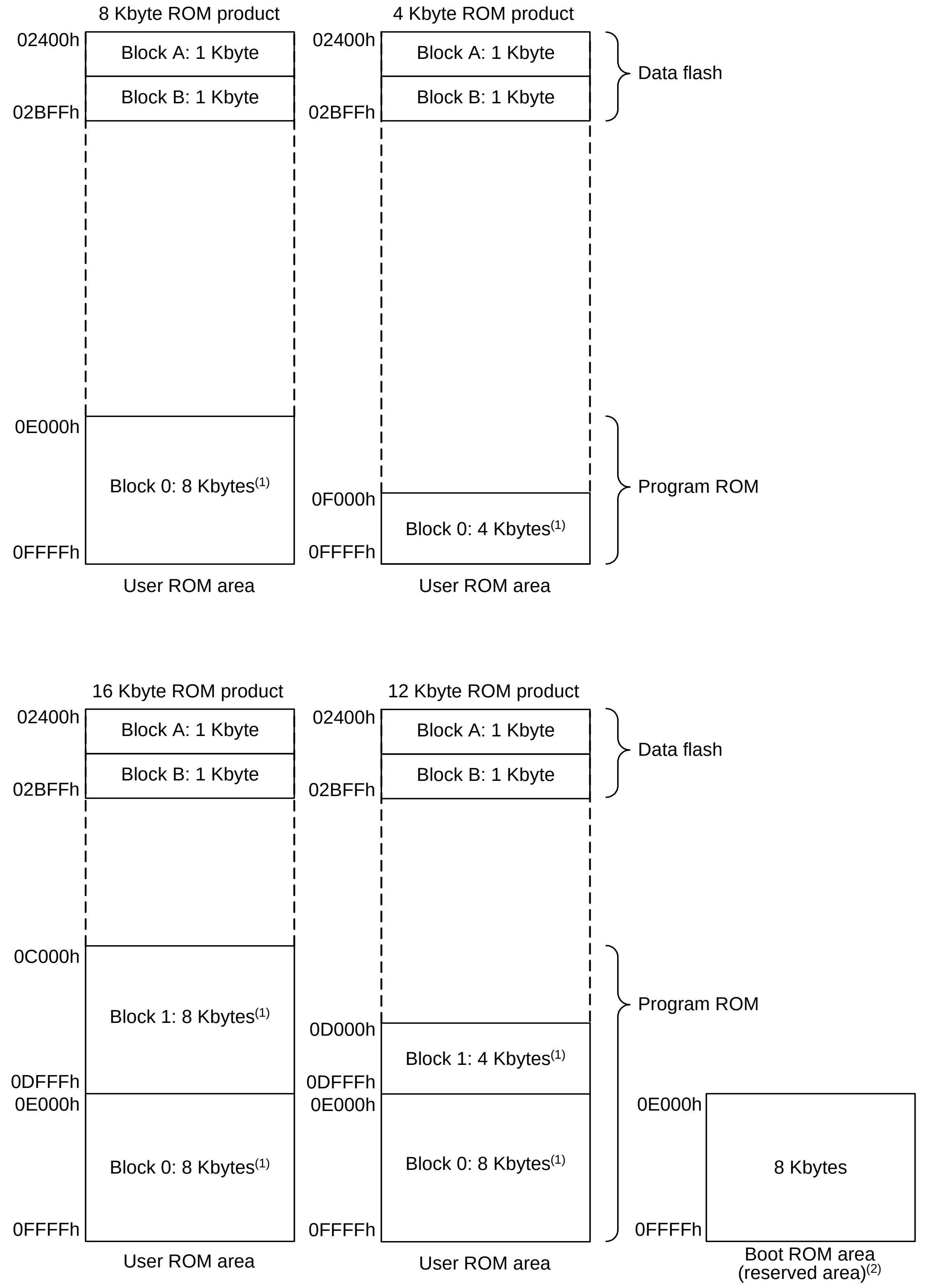
The rewrite control program for standard serial I/O mode is stored in the boot ROM area before shipment. The boot ROM area and the user ROM area share the same address, but have separate memory areas.



NOTES:

1. When the FMR02 bit in the FMR0 register is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register to 0 (rewrite enabled), block 0 is rewritable. When the FMR16 bit is set to 0 (rewrite enabled), block 1 is rewritable (only for CPU rewrite mode).
2. This area is for storing the boot program provided by Renesas Technology.

Figure 17.1 Flash Memory Block Diagram for R8C/18 Group



NOTES:
1. When the FMR02 bit in the FMR0 register is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register to 0 (rewrite enabled), block 0 is rewritable. When the FMR16 bit is set to 0 (rewrite enabled), block 1 is rewritable (only for CPU rewrite mode).
2. This area is for storing the boot program provided by Renesas Technology.

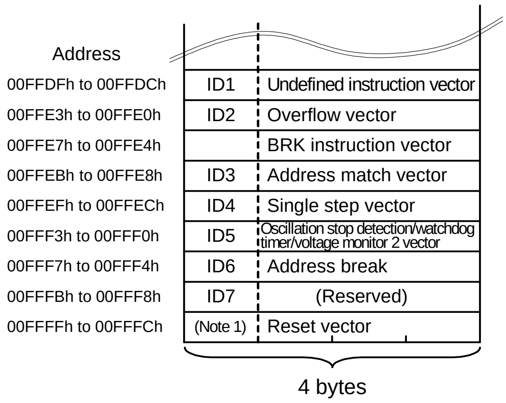
Figure 17.2 Flash Memory Block Diagram for R8C/19 Group

17.3 Functions to Prevent Rewriting of Flash Memory

Standard serial I/O mode has an ID code check function, and parallel I/O mode has a ROM code protect function to prevent the flash memory from being read or rewritten easily.

17.3.1 ID Code Check Function

This function is used in standard serial I/O mode. Unless the flash memory is blank, the ID codes sent from the programmer and the ID codes written in the flash memory are checked to see if they match. If the ID codes do not match, the commands sent from the programmer are not acknowledged. The ID codes consist of 8 bits of data each, the areas of which, beginning with the first byte, are 00FFDFh, 00FFE3h, 00FFEBh, 00FFEFh, 00FFF3h, 00FFF7h, and 00FFFBh. Write programs in which the ID codes are set at these addresses and write them to the flash memory.



Address		
00FFDFh to 00FFDCh	ID1	Undefined instruction vector
00FFE3h to 00FFE0h	ID2	Overflow vector
00FFE7h to 00FFE4h		BRK instruction vector
00FFEBh to 00FFE8h	ID3	Address match vector
00FFEFh to 00FFECh	ID4	Single step vector
00FFF3h to 00FFF0h	ID5	Oscillation stop detection/watchdog timer/voltage monitor 2 vector
00FFF7h to 00FFF4h	ID6	Address break
00FFFBh to 00FFF8h	ID7	(Reserved)
00FFFFh to 00FFFCh	(Note 1)	Reset vector

NOTE:

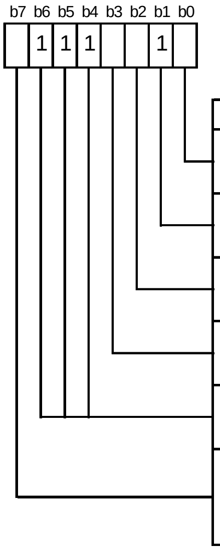
1. The OFS register is assigned to 00FFFFh. Refer to **Figure 13.2 Registers OFS and WDC** and **Figure 13.3 Registers WDTR and WDTS** for OFS register details.

Figure 17.3 Address for Stored ID Code

17.3.2 ROM Code Protect Function

The ROM code protect function disables reading or changing the contents of the on-chip flash memory by the OFS register in parallel I/O mode. Figure 17.4 shows the OFS Register.
The ROM code protect function is enabled by writing 0 to the ROMCP1 bit and 1 to the ROMCR bit. It disables reading or changing the contents of the on-chip flash memory.
Once ROM code protect is enabled, the content in the internal flash memory cannot be rewritten in parallel I/O mode. To disable ROM code protect, erase the block including the OFS register with CPU rewrite mode or standard serial I/O mode.

Option Function Select Register[1]



Symbol	Address	Before Shipment
OFS	0FFFFh	FFh[2]

Bit Symbol	Bit Name	Function	RW
WDTON	Watchdog timer start select bit	0 : Starts w atchdog timer automatically after reset. 1 : Watchdog timer is inactive after reset.	RW
— (b1)	Reserved bit	Set to 1.	RW
ROMCR	ROM code protect disabled bit	0 : ROM code protect disabled 1 : ROMCP1enabled	RW
ROMCP1	ROM code protect bit	0 : ROM code protect enabled 1 : ROM code protect disabled	RW
— (b6-b4)	Reserved bits	Set to 1.	RW
CSPROINI	Count source protect mode after reset select bit	0 : Count source protect mode enabled after reset. 1 : Count source protect mode disabled after reset.	RW

NOTES:
1. The OFS register is on the flash memory. Write to the OFS register w ith a program.
2. If the block including the OFS register is erased, FFh is set to the OFS register.

Figure 17.4 OFS Register

17.4 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten by executing software commands from the CPU. Therefore, the user ROM area can be rewritten directly while the MCU is mounted on a board without using a ROM programmer. Execute the program and block erase commands only to blocks in the user ROM area.

The flash module has an erase-suspend function when an interrupt request is generated during an erase operation in CPU rewrite mode. It performs an interrupt process after the erase operation is halted temporarily. During erase-suspend, the user ROM area can be read by a program.

In case an interrupt request is generated during an auto-program operation in CPU rewrite mode, the flash module has a program-suspend function which performs the interrupt process after the auto-program operation. During program-suspend, the user ROM area can be read by a program.

CPU rewrite mode has an erase write 0 mode (EW0 mode) and an erase write 1 mode (EW1 mode). Table 17.3 lists the Differences between EW0 Mode and EW1 Mode.

Table 17.3 Differences between EW0 Mode and EW1 Mode

Item	EW0 Mode	EW1 Mode
Operating mode	Single-chip mode	Single-chip mode
Areas in which a rewrite control program can be located	User ROM area	User ROM area
Areas in which a rewrite control program can be executed	Necessary to transfer to any area other than the flash memory (e.g., RAM) before executing.	Executing directly in user ROM area is possible.
Areas which can be rewritten	User ROM area	User ROM area However, blocks which contain a rewrite control program are excluded.[1]
Software command restrictions	None	• Program and block erase commands • Cannot be run on any block which contains a rewrite control program • Read status register command cannot be executed
Modes after program or erase	Read status register mode	Read array mode
CPU status during auto-write and auto-erase	Operating	Hold state (I/O ports hold state before the command is executed)
Flash memory status detection	• Read bits FMR00, FMR06, and FMR07 in the FMR0 register by a program. • Execute the read status register command and read bits SR7, SR5, and SR4 in the status register.	Read bits FMR00, FMR06, and FMR07 in the FMR0 register by a program.
Conditions for transition to erase-suspend	Set bits FMR40 and FMR41 in the FMR4 register to 1 by a program.	The FMR40 bit in the FMR4 register is set to 1 and the interrupt request of the enabled maskable interrupt is generated.
Conditions for transitions to program-suspend	Set bits FMR40 and FMR42 in the FMR4 register to 1 by a program.	The FMR40 bit in the FMR4 register is set to 1 and the interrupt request of the enabled maskable interrupt is generated.
CPU clock	5 MHz or below	No restriction (on clock frequency to be used)

NOTE:

1. When the FMR02 bit in the FMR0 register is set to 1 (rewrite enabled), rewriting block 0 is enabled by setting the FMR15 bit in the FMR1 register to 0 (rewrite enabled), and rewriting block 1 is enabled by setting the FMR16 bit to 0 (rewrite enabled).

17.4.1 EW0 Mode

The MCU enters CPU rewrite mode and software commands can be acknowledged by setting the FMR01 bit in the FMR0 register to 1 (CPU rewrite mode enabled). In this case, since the FMR11 bit in the FMR1 register is set to 0, EW0 mode is selected.

Use software commands to control program and erase operations. The FMR0 register or the status register can be used to determine when program and erase operations complete.

During auto-erasure, set the FMR40 bit to 1 (erase-suspend enabled) and the FMR41 bit to 1 (request erase-suspend). Wait for td(SR-ES) and ensure that the FMR46 bit is set to 1 (read enabled) before accessing the user ROM area. The auto-erase operation can be restarted by setting the FMR41 bit to 0 (erase restarts).

To enter program-suspend during the auto-program operation, set the FMR40 bit to 1 (suspend enabled) and the FMR42 bit to 1 (request program-suspend). Wait for td(SR-ES) and ensure that the FMR46 bit is set to 1 (read enabled) before accessing the user ROM area. The auto-program operation can be restarted by setting the FMR42 bit to 0 (program restarts).

17.4.2 EW1 Mode

The MCU is switched to EW1 mode by setting the FMR11 bit to 1 (EW1 mode) after setting the FMR01 bit to 1 (CPU rewrite mode enabled).

The FMR0 register can be used to determine when program and erase operations complete. Do not execute the read status register command in EW1 mode.

To enable the erase-suspend function during auto-erasure, execute the block erase command after setting the FMR40 bit to 1 (erase-suspend enabled). The interrupt to enter erase-suspend should be in interrupt enabled status. After waiting for td(SR-ES) after the block erase command is executed, the interrupt request is acknowledged.

When an interrupt request is generated, the FMR41 bit is automatically set to 1 (request erase-suspend) and the auto-erase operation suspends. If an auto-erase operation does not complete (FMR00 bit is 0) after an interrupt process completes, the auto-erase operation restarts by setting the FMR41 bit to 0 (erase restarts).

To enable the program-suspend function during auto-programming, execute the program command after setting the FMR40 bit to 1 (suspend enabled). The interrupt to enter a program-suspend should be in interrupt enabled status. After waiting for td(SR-ES) after the program command is executed, an interrupt request is acknowledged.

When an interrupt request is generated, the FMR42 bit is automatically set to 1 (request program-suspend) and the auto-program operation suspends. When the auto-program operation does not complete (FMR00 bit is 0) after the interrupt process completes, the auto-program operation can be restarted by setting the FMR42 bit to 0 (programming restarts).

Figure 17.5 shows the FMR0 Register. Figure 17.7 shows the FMR4 Register.

17.4.2.1 FMR00 Bit

This bit indicates the operating status of the flash memory. The bits value is 0 during programming, erasure, or erase-suspend mode; otherwise, it is 1.

17.4.2.2 FMR01 Bit

The MCU is made ready to accept commands by setting the FMR01 bit to 1 (CPU rewrite mode).

17.4.2.3 FMR02 Bit

Rewriting of blocks 0 and 1 does not accept the program or block erase commands if the FMR02 bit is set to 0 (rewrite disabled).
Rewriting of blocks 0 and 1 is controlled by bits FMR15 and FMR16 if the FMR02 bit is set to 1 (rewrite enabled).

17.4.2.4 FMSTP Bit

This bit is used to initialize the flash memory control circuits, and also to reduce the amount of current consumed by the flash memory. Access to the flash memory is disabled by setting the FMSTP bit to 1. Therefore, the FMSTP bit must be written to by a program located outside of the flash memory.
In the following cases, set the FMSTP bit to 1:

- When flash memory access resulted in an error while erasing or programming in EW0 mode (FMR00 bit not reset to 1 (ready)).
- When entering on-chip oscillator mode (main clock stops).

Figure 17.11 shows a flowchart to be followed before and after entering on-chip oscillator mode (main clock stop). Note that when going to stop or wait mode while the CPU rewrite mode is disabled, the FMR0 register does not need to be set because the power for the flash memory is automatically turned off and is turned back on again after returning from stop or wait mode.

17.4.2.5 FMR06 Bit

This is a read-only bit indicating the status of an auto-program operation. The bit is set to 1 when a program error occurs; otherwise, it is set to 0. Refer to **17.4.5 Full Status Check** for details.

17.4.2.6 FMR07 Bit

This is a read-only bit indicating the status of an auto-erase operation. The bit is set to 1 when an erase error occurs; otherwise, it is set to 0. Refer to **17.4.5 Full Status Check** for details.

17.4.2.7 FMR11 Bit

Setting this bit to 1 (EW1 mode) places the MCU in EW1 mode.

17.4.2.8 FMR15 Bit

When the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit is set to 0 (rewrite enabled), block 0 accepts program and block erase commands.

17.4.2.9 FMR16 Bit

When the FMR02 bit is set to 1 (rewrite enabled) and the FMR16 bit is set to 0 (rewrite enabled), block 1 accepts program and block erase commands.

17.4.2.10 FMR40 Bit

The suspend function is enabled by setting the FMR40 bit to 1 (enable).

17.4.2.11 FMR41 Bit

In EW0 mode, the MCU enters erase-suspend mode when the FMR41 bit is set to 1 by a program. The FMR41 bit is automatically set to 1 (request erase-suspend) when an interrupt request of an enabled interrupt is generated in EW1 mode, and then the MCU enters erase-suspend mode.
Set the FMR41 bit to 0 (erase restarts) when the auto-erase operation restarts.

17.4.2.12 FMR42 Bit

In EW0 mode, the MCU enters program-suspend mode when the FMR42 bit is set to 1 by a program. The FMR42 bit is automatically set to 1 (request program-suspend) when an interrupt request of an enabled interrupt is generated in EW1 mode, and then the MCU enters program-suspend mode.
Set the FMR42 bit to 0 (program restart) when the auto-program operation restarts.

17.4.2.13 FMR43 Bit

When the auto-erase operation starts, the FMR43 bit is set to 1 (erase execution in progress). The FMR43 bit remains set to 1 (erase execution in progress) during erase-suspend operation.
When the auto-erase operation ends, the FMR43 bit is set to 0 (erase not executed).

17.4.2.14 FMR44 Bit

When the auto-program operation starts, the FMR44 bit is set to 1 (program execution in progress). The FMR44 bit remains set to 1 (program execution in progress) during program-suspend operation.
When the auto-program operation ends, the FMR44 bit is set to 0 (program not executed).

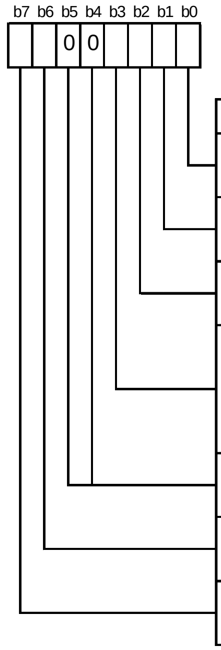
17.4.2.15 FMR46 Bit

The FMR46 bit is set to 0 (reading disabled) during auto-erase execution and set to 1 (reading enabled) in erase-suspend mode. Do not access the flash memory while this bit is set to 0.

17.4.2.16 FMR47 BIt

Power consumption when reading flash memory can be reduced by setting the FMR47 bit to 1 (enabled).

Flash Memory Control Register 0



Symbol	Address	After Reset
FMR0	01B7h	00000001b

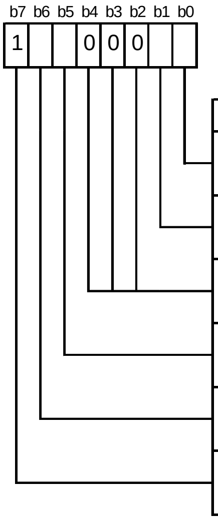
Bit Symbol	Bit Name	Function	RW
FMR00	RY/$\overline{\text{BY}}$ status flag	0 : Busy (writing or erasing in progress) 1 : Ready	RO
FMR01	CPU rewrite mode select bit[1]	0 : CPU rewrite mode disabled 1 : CPU rewrite mode enabled	RW
FMR02	Block 0, 1 rewrite enable bit[2, 6]	0 : Disables rewrite. 1 : Enables rewrite.	RW
FMSTP	Flash memory stop bit[3, 5]	0 : Enables flash memory operation. 1 : Stops flash memory (enters low-power consumption state and flash memory is reset).	RW
— (b5-b4)	Reserved bits	Set to 0.	RW
FMR06	Program status flag[4]	0 : Completed successfully 1 : Terminated by error	RO
FMR07	Erase status flag[4]	0 : Completed successfully 1 : Terminated by error	RO

NOTES:

1. To set this bit to 1, set it to 1 immediately after setting it first to 0. Do not generate an interrupt between setting the bit to 0 and setting it to 1. Enter read array mode and set this bit to 0.
2. Set this bit to 1 immediately after setting it first to 0 while the FMR01 bit is set to 1.
   Do not generate an interrupt between setting the bit to 0 and setting it to 1.
3. Set this bit by a program located in a space other than the flash memory.
4. This bit is set to 0 by executing the clear status command.
5. This bit is enabled when the FMR01 bit is set to 1 (CPU rewrite mode). When the FMR01 bit is set to 0, writing 1 to the FMSTP bit causes the FMSTP bit to be set to 1. The flash memory does not enter low-power consumption state nor is it reset.
6. When setting the FMR01 bit to 0 (CPU rewrite mode disabled), the FMR02 bit is set to 0 (disables rewrite).

Figure 17.5 FMR0 Register

Flash Memory Control Register 1



Symbol	Address	After Reset
FMR1	01B5h	1000000Xb

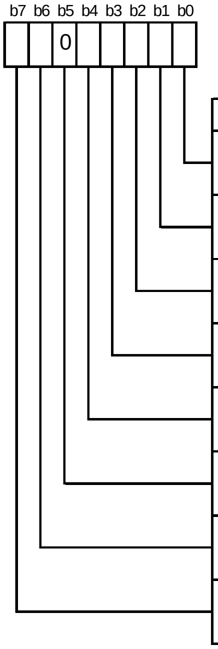
Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved bit	When read, the content is undefined.	RO
FMR11	EW1 mode select bit(1, 2)	0 : EW0 mode 1 : EW1 mode	RW
— (b4-b2)	Reserved bits	Set to 0.	RW
FMR15	Block 0 rewrite disable bit(2, 3)	0 : Enables rewrite. 1 : Disables rewrite.	RW
FMR16	Block 1rewrite disable bit(2, 3)	0 : Enables rewrite. 1 : Disables rewrite.	RW
— (b7)	Reserved bit	Set to 1.	RW

NOTES:

1. To set this bit to 1, set it to 1 immediately after setting it first to 0 while the FMR01 bit is set to 1 (CPU rewrite mode enable) . Do not generate an interrupt between setting the bit to 0 and setting it to 1.
2. This bit is set to 0 by setting the FMR01 bit to 0 (CPU rewrite mode disabled).
3. When the FMR01 bit is set to 1 (CPU rewrite mode enabled), bits FMR15 and FMR16 can be written to.
   To set this bit to 0, set it to 0 immediately after setting it first to 1.
   To set this bit to 1, set it to 1.

Figure 17.6 FMR1 Register

Flash Memory Control Register 4



Symbol	Address	After Reset
FMR4	01B3h	01000000b

Bit Symbol	Bit Name	Function	RW
FMR40	Erase-suspend function enable bit(1)	0 : Disable 1 : Enable	RW
FMR41	Erase-suspend request bit(2)	0 : Erase restart 1 : Erase-suspend request	RW
FMR42	Program-suspend request bit(3)	0 : Program restart 1 : Program-suspend request	RW
FMR43	Erase command flag	0 : Erase not executed 1 : Erase execution in progress	RO
FMR44	Program command flag	0 : Program not executed 1 : Program execution in progress	RO
— (b5)	Reserved bit	Set to 0.	RO
FMR46	Read status flag	0 : Disables reading. 1 : Enables reading.	RO
FMR47	Low -pow er consumption read mode enable bit(1, 4)	0 : Disable 1 : Enable	RW

NOTES:

1. To set this bit to 1, set it to 1 immediately after setting it first to 0. Do not generate an interrupt betw een setting the bit to 0 and setting it to 1.
2. This bit is enabled w hen the FMR40 bit is set to 1 (enable) and it can be w ritten to during the period betw een issuing an erase command and completing the erase. (This bit is set to 0 during the periods other than the above.)
   In EW0 mode, it can be set to 0 and 1 by a program.
   In EW1 mode, it is automatically set to 1 if a maskable interrupt is generated during an erase operation w hile the FMR40 bit is set to 1. Do not set this bit to 1 by a program (0 can be w ritten).
3. The FMR42 bit is enabled only w hen the FMR40 bit is set to 1 (enable) and programming to the FMR42 bit is enabled until auto-programming ends after a program command is generated. (This bit is set to 0 during periods other than the above.)
   In EW0 mode, 0 or 1 can be programmed to the FMR42 bit by a program.
   In EW1 mode, the FMR42 bit is automatically set to 1 by generating a maskable interrupt during auto-programming w hen the FMR40 bit is set to 1. 1 cannot be w ritten to the FMR42 bit by a program.
4. Use this mode only in low -speed on-chip oscillator mode.

Figure 17.7 FMR4 Register

Figure 17.8 shows the Timing of Suspend Operation.

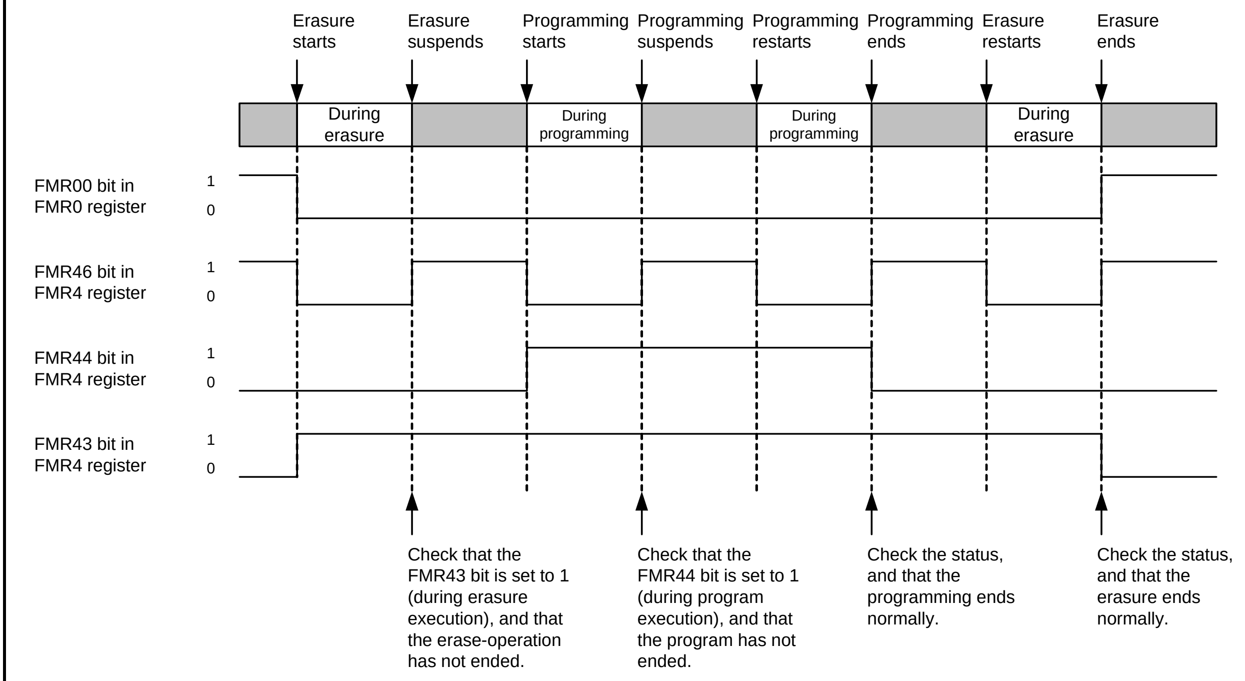
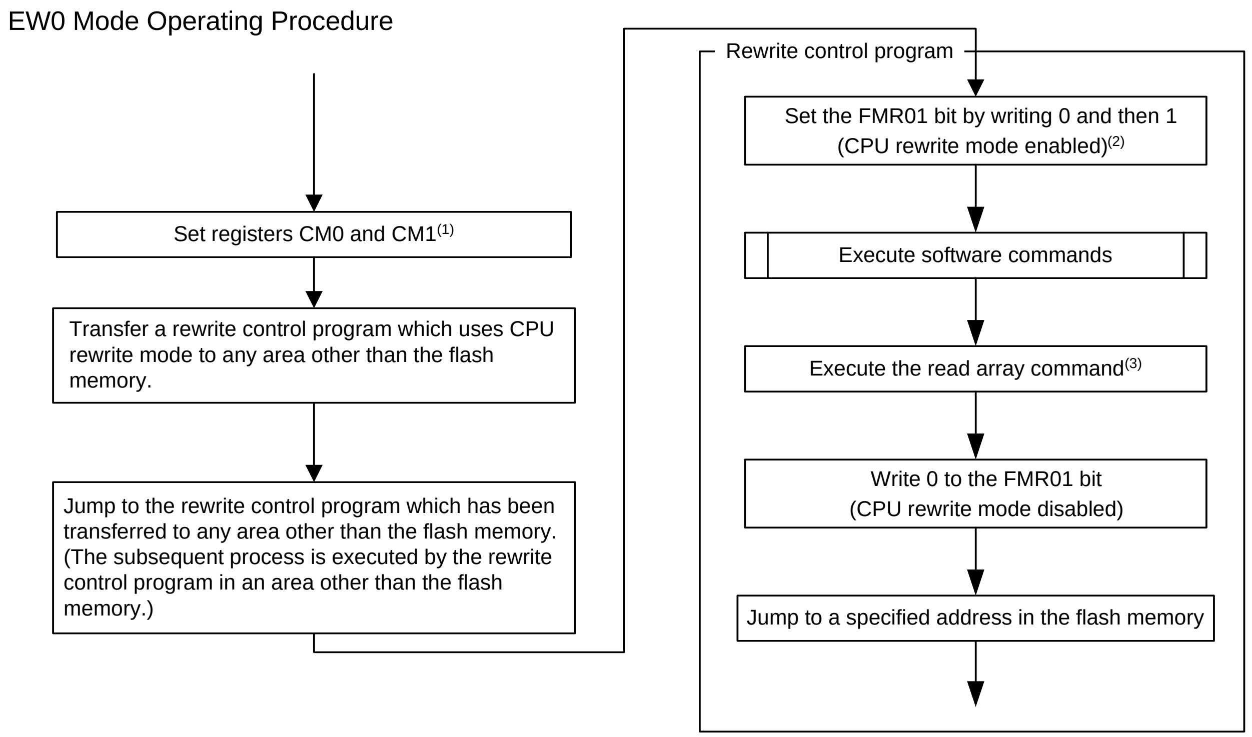

Erasure starts
Erasure suspends
Programming starts
Programming suspends
Programming restarts
Programming ends
Erasure restarts
Erasure ends
During erasure
During programming
During programming
During erasure
FMR00 bit in FMR0 register
FMR46 bit in FMR4 register
FMR44 bit in FMR4 register
FMR43 bit in FMR4 register
1
0
Check that the FMR43 bit is set to 1 (during erasure execution), and that the erase-operation has not ended.
Check that the FMR44 bit is set to 1 (during program execution), and that the program has not ended.
Check the status, and that the programming ends normally.
Check the status, and that the erasure ends normally.

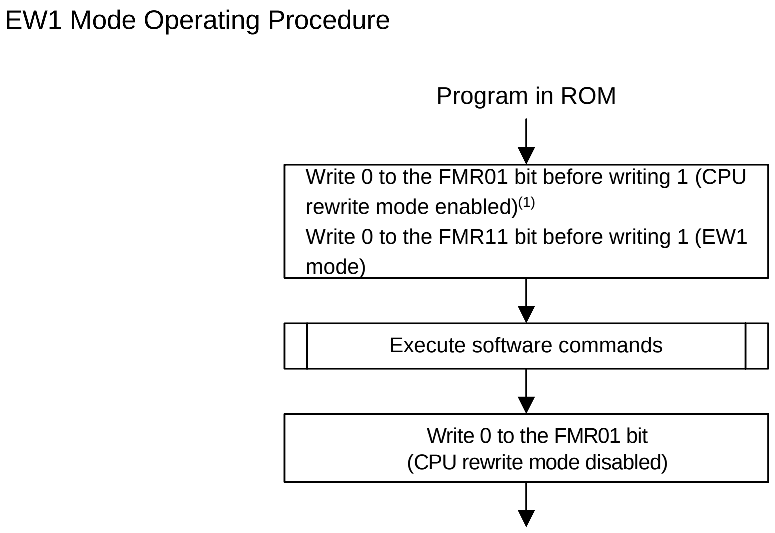
Figure 17.8 Timing of Suspend Operation

Figure 17.9 shows How to Set and Exit EW0 Mode. Figure 17.10 shows How to Set and Exit EW1 Mode.


EW0 Mode Operating Procedure
Set registers CM0 and CM1(1)
Transfer a rewrite control program which uses CPU rewrite mode to any area other than the flash memory.
Jump to the rewrite control program which has been transferred to any area other than the flash memory. (The subsequent process is executed by the rewrite control program in an area other than the flash memory.)
Rewrite control program
Set the FMR01 bit by writing 0 and then 1 (CPU rewrite mode enabled)(2)
Execute software commands
Execute the read array command(3)
Write 0 to the FMR01 bit (CPU rewrite mode disabled)
Jump to a specified address in the flash memory

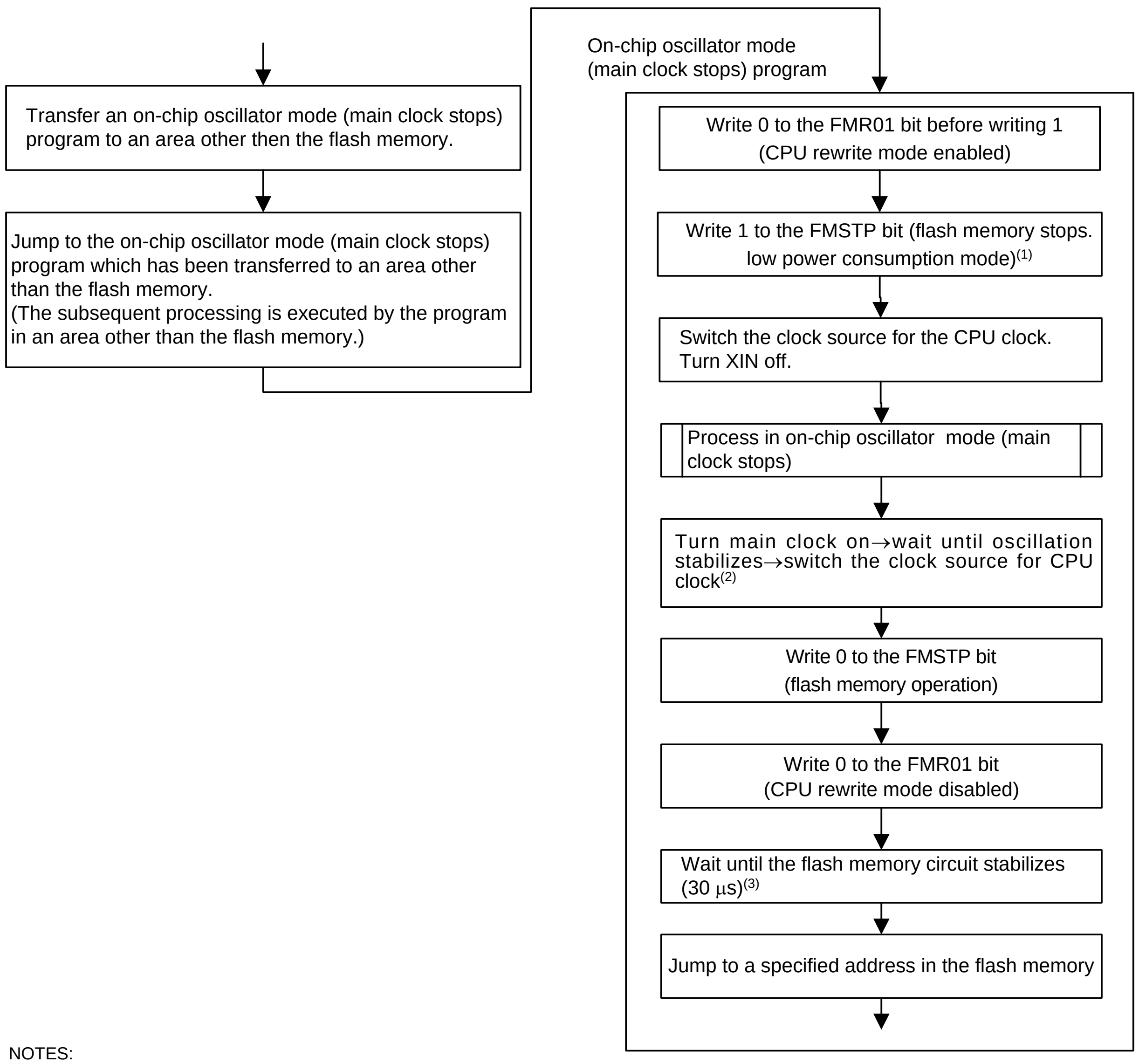
NOTES:
1. Select 5 MHz or below for the CPU clock by the CM06 bit in the CM0 register and bits CM16 to CM17 in the CM1 register.
2. To set the FMR01 bit to 1, write 0 to the FMR01 bit before writing 1. Do not generate an interrupt between writing 0 and 1.
3. Disable the CPU rewrite mode after executing the read array command.

Figure 17.9 How to Set and Exit EW0 Mode


EW1 Mode Operating Procedure
Program in ROM
Write 0 to the FMR01 bit before writing 1 (CPU rewrite mode enabled)(1)
Write 0 to the FMR11 bit before writing 1 (EW1 mode)
Execute software commands
Write 0 to the FMR01 bit (CPU rewrite mode disabled)

NOTE:
1. To set the FMR01 bit to 1, write 0 to the FMR01 bit before writing 1. Do not generate an interrupt between writing 0 and 1.

Figure 17.10 How to Set and Exit EW1 Mode


Transfer an on-chip oscillator mode (main clock stops) program to an area other then the flash memory.
Jump to the on-chip oscillator mode (main clock stops) program which has been transferred to an area other than the flash memory.
(The subsequent processing is executed by the program in an area other than the flash memory.)
On-chip oscillator mode (main clock stops) program
Write 0 to the FMR01 bit before writing 1 (CPU rewrite mode enabled)
Write 1 to the FMSTP bit (flash memory stops. low power consumption mode)(1)
Switch the clock source for the CPU clock. Turn XIN off.
Process in on-chip oscillator mode (main clock stops)
Turn main clock on→wait until oscillation stabilizes→switch the clock source for CPU clock(2)
Write 0 to the FMSTP bit (flash memory operation)
Write 0 to the FMR01 bit (CPU rewrite mode disabled)
Wait until the flash memory circuit stabilizes (30 µs)(3)
Jump to a specified address in the flash memory

NOTES:

1. Set the FMR01 bit to 1 (CPU rewrite mode) before setting the FMSTP bit to 1 .
2. Before switching to a different clock source for the CPU, make sure the designated clock is stable.
3. Insert a 30 µs wait time in a program. Do not access the flash memory during this wait time.

Figure 17.11 Process to Reduce Power Consumption in On-Chip Oscillator Mode (Main Clock Stops)

17.4.3 Software Commands

The software commands are described below. Read or write commands and data in 8-bit units.

Table 17.4 Software Commands

Command	First Bus Cycle			Second Bus Cycle		
	Mode	Address	Data (D7 to D0)	Mode	Address	Data (D7 to D0)
Read array	Write	×	FFh			
Read status register	Write	×	70h	Read	×	SRD
Clear status register	Write	×	50h			
Program	Write	WA	40h	Write	WA	WD
Block erase	Write	×	20h	Write	BA	D0h

SRD: Status register data (D7 to D0)

WA: Write address (ensure the address specified in the first bus cycle is the same address as the write address specified in the second bus cycle.)

WD: Write data (8 bits)

BA: Given block address

×: Any specified address in the user ROM area

17.4.3.1 Read Array Command

The read array command reads the flash memory.
The MCU enters read array mode when FFh is written in the first bus cycle. When the read address is entered in the following bus cycles, the content of the specified address can be read in 8-bit units. Since the MCU remains in read array mode until another command is written, the contents of multiple addresses can be read continuously.

17.4.3.2 Read Status Register Command

The read status register command is used to read the status register.
When 70h is written in the first bus cycle, the status register can be read in the second bus cycle. (Refer to **17.4.4 Status Register**.) When reading the status register, specify an address in the user ROM area.
Do not execute this command in EW1 mode.

17.4.3.3 Clear Status Register Command

The clear status register command sets the status register to 0.
When 50h is written in the first bus cycle, bits FMR06 to FMR07 in the FMR0 register and SR4 to SR5 in the status register are set to 0.

17.4.3.4 Program Command

The program command writes data to the flash memory in 1-byte units.
By writing 40h in the first bus cycle and data to the write address in the second bus cycle, an auto-program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same address as the write address specified in the second bus cycle.
The FMR00 bit in the FMR0 register can be used to determine whether auto-programming has completed. The FMR00 bit is set to 0 during auto-programming and set to 1 when auto-programming completes.
The FMR06 bit in the FMR0 register can be used to determine the result of auto-programming after it has been finished. (Refer to **17.4.5 Full Status Check**.)
Do not write additions to the already programmed addresses.
When the FMR02 bit in the FMR0 register is set to 0 (rewriting disabled), or the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register is set to 1 (rewriting disabled), program commands targeting block 0 are not acknowledged. When the FMR16 bit is set to 1 (rewriting disabled), program commands targeting block 1 are not acknowledged.
In EW1 mode, do not execute this command for any address which a rewrite control program is allocated.
In EW0 mode, the MCU enters read status register mode at the same time auto-programming starts and the status register can be read. The status register bit 7 (SR7) is set to 0 at the same time auto-programming starts and set back to 1 when auto-programming completes. In this case, the MCU remains in read status register mode until the next read array command is written. The status register can be read to determine the result of auto-programming after/auto-programming has completed.

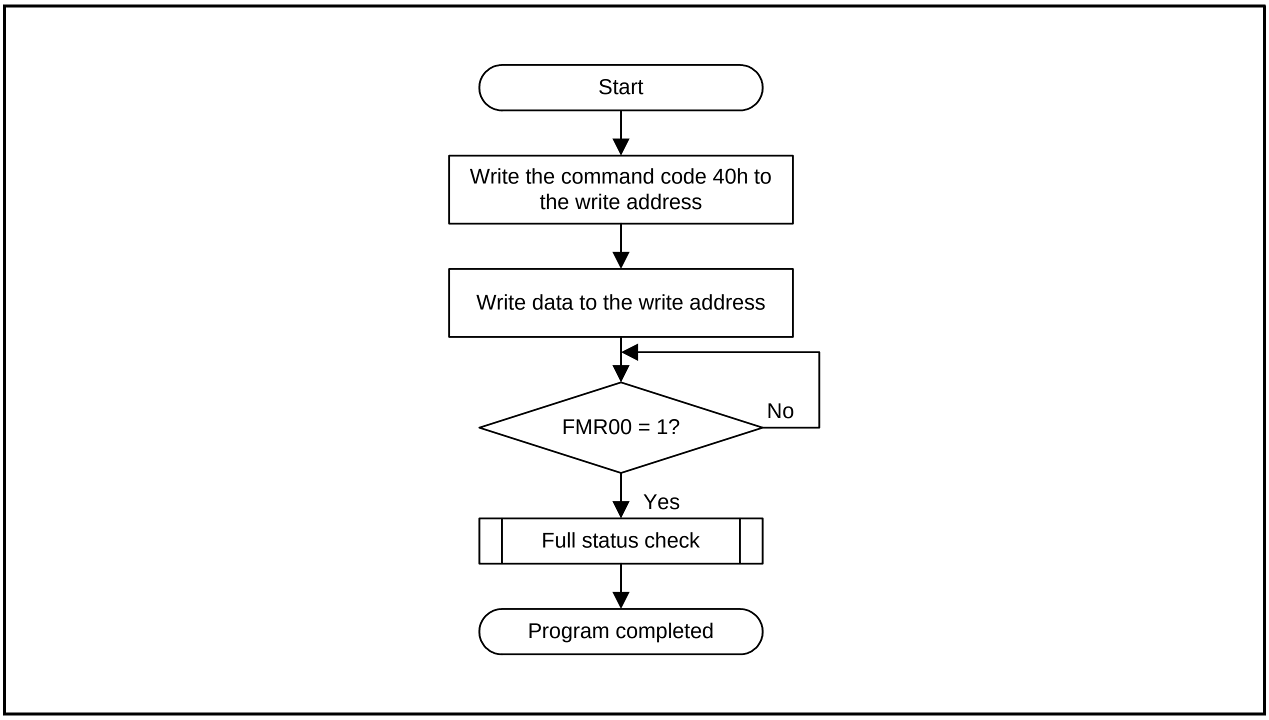

Start
Write the command code 40h to the write address
Write data to the write address
FMR00 = 1?
No
Yes
Full status check
Program completed

Figure 17.12 Program Command

17.4.3.5 Block Erase

When 20h is written in the first bus cycle and D0h is written to a given address of a block in the second bus cycle, an auto-erase operation (erase and verify) of the specified block starts.
The FMR00 bit in the FMR0 register can be used to determine whether auto-erasure has completed.
The FMR00 bit is set to 0 during auto-erasure and set to 1 when auto-erasure completes.
The FMR07 bit in the FMR0 register can be used to determine the result of auto-erasure after auto-erasure has completed. (Refer to **17.4.5 Full Status Check**.)
When the FMR02 bit in the FMR0 register is set to 0 (rewriting disable) or the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register is set to 1 (rewriting disable), the block erase commands targeting block 0 are not acknowledged. When the FMR16 bit is set to 1 (rewriting disable), the block erase commands targeting block 1 are not acknowledged.
Do not use the block erase command during program-suspend.
Figure 17.13 shows the Block Erase Command (When Not Using Erase-Suspend Function). Figure 17.14 shows the Block Erase Command (When Using Erase-Suspend Function).
In EW1 mode, do not execute this command for any address to which a rewrite control program is allocated.
In EW0 mode, the MCU enters read status register mode at the same time auto-erasure starts and the status register can be read. The status register bit 7 (SR7) is set to 0 at the same time auto-erasure starts and set back to 1 when auto-erasure completes. In this case, the MCU remains in read status register mode until the next read array command is written.

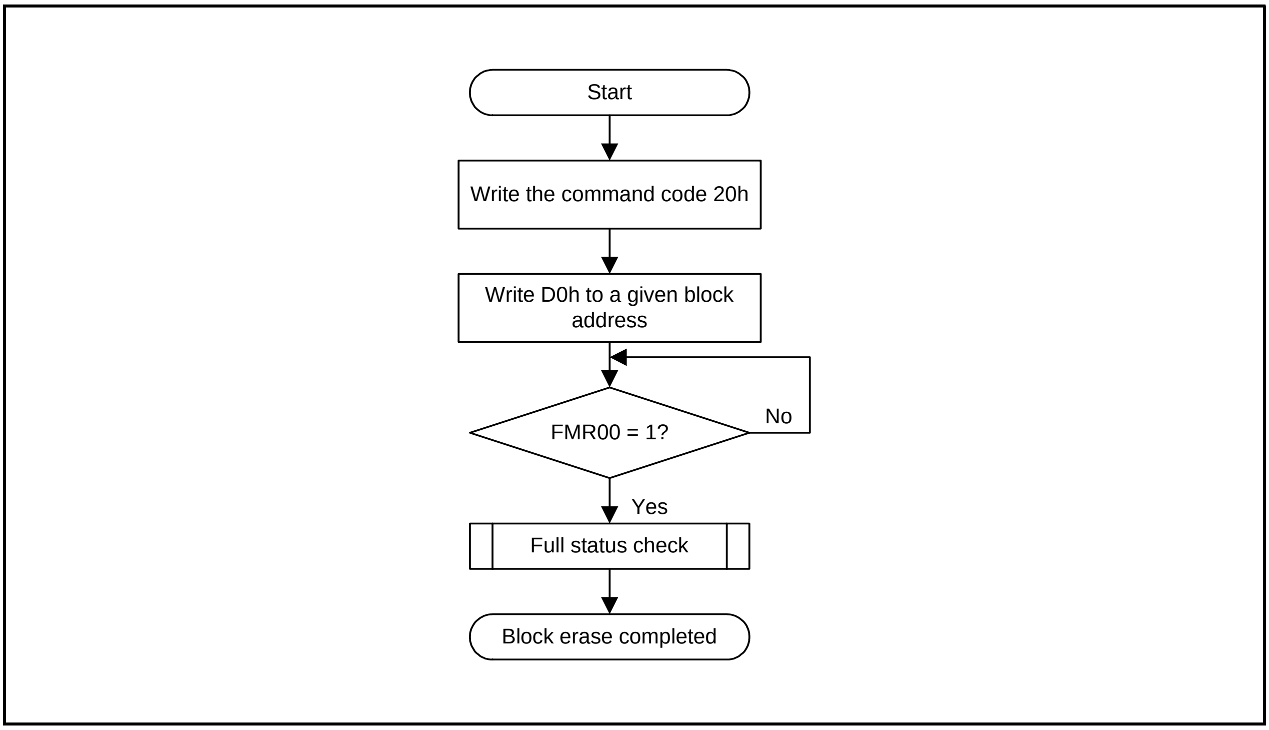
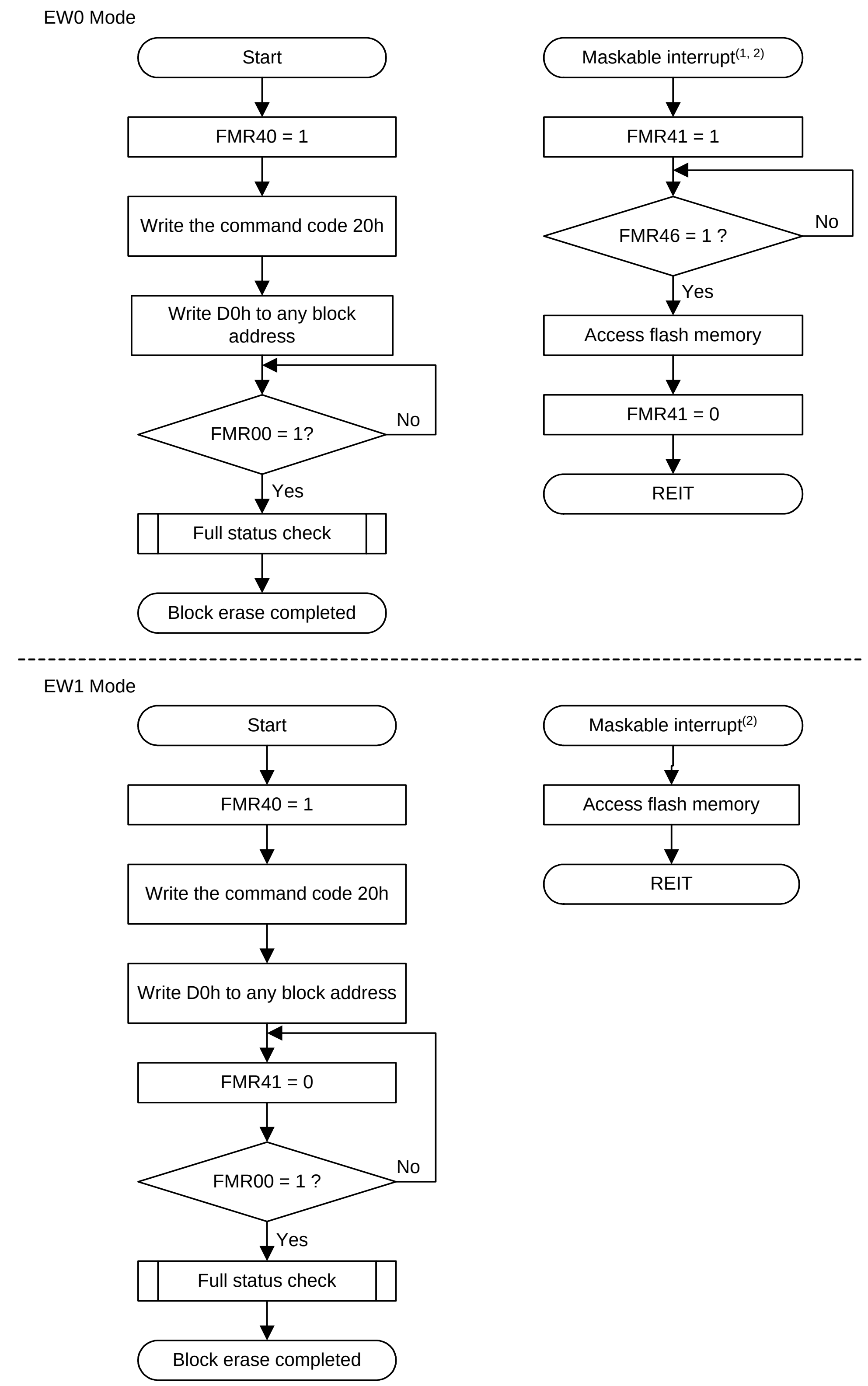

Start
Write the command code 20h
Write D0h to a given block address
FMR00 = 1?
No
Yes
Full status check
Block erase completed

Figure 17.13 Block Erase Command (When Not Using Erase-Suspend Function)



NOTES:

1. In EW0 mode, the interrupt vector table and interrupt routine for interrupts to be used should be allocated to the RAM area.
2. td(SR-ES) is needed until the interrupt request is acknowledged after it is generated. The interrupt to enter erase-suspend should be in interrupt enabled status.

Figure 17.14 Block Erase Command (When Using Erase-Suspend Function)

17.4.4 Status Register

The status register indicates the operating status of the flash memory and whether an erase or program operation has completed normally or in error. Status of the status register can be read by bits FMR00, FMR06, and FMR07 in the FMR0 register.

Table 17.5 lists the Status Register Bits.

In EW0 mode, the status register can be read in the following cases:

- When a given address in the user ROM area is read after writing the read status register command
- When a given address in the user ROM area is read after executing the program or block erase command but before executing the read array command.

17.4.4.1 Sequencer Status (Bits SR7 and FMR00)

The sequencer status bits indicate the operating status of the flash memory. SR7 is set to 0 (busy) during/auto-programming and auto-erasure, and is set to 1 (ready) at the same time the operation completes.

17.4.4.2 Erase Status (Bits SR5 and FMR07)

Refer to **17.4.5 Full Status Check**.

17.4.4.3 Program Status (Bits SR4 and FMR06)

Refer to **17.4.5 Full Status Check**.

Table 17.5 Status Register Bits

Status Register Bit	FMR0 Register Bit	Status Name	Description		Value after Reset
			0	1	
SR0 (D0)	–	Reserved	–	–	–
SR1 (D1)	–	Reserved	–	–	–
SR2 (D2)	–	Reserved	–	–	–
SR3 (D3)	–	Reserved	–	–	–
SR4 (D4)	FMR06	Program status	Completed normally	Error	0
SR5 (D5)	FMR07	Erase status	Completed normally	Error	0
SR6 (D6)	–	Reserved	–	–	–
SR7 (D7)	FMR00	Sequencer status	Busy	Ready	1

- D0 to D7: Indicate the data bus which is read when the read status register command is executed.
- Bits FMR07 (SR5) to FMR06 (SR4) are set to 0 by executing the clear status register command.
- When the FMR07 bit (SR5) or FMR06 bit (SR4) is set to 1, the program and block erase commands cannot be accepted.

17.4.5 Full Status Check

When an error occurs, bits FMR06 to FMR07 in the FMR0 register are set to 1, indicating the occurrence of an error. Therefore, checking these status bits (full status check) can be used to determine the execution result.

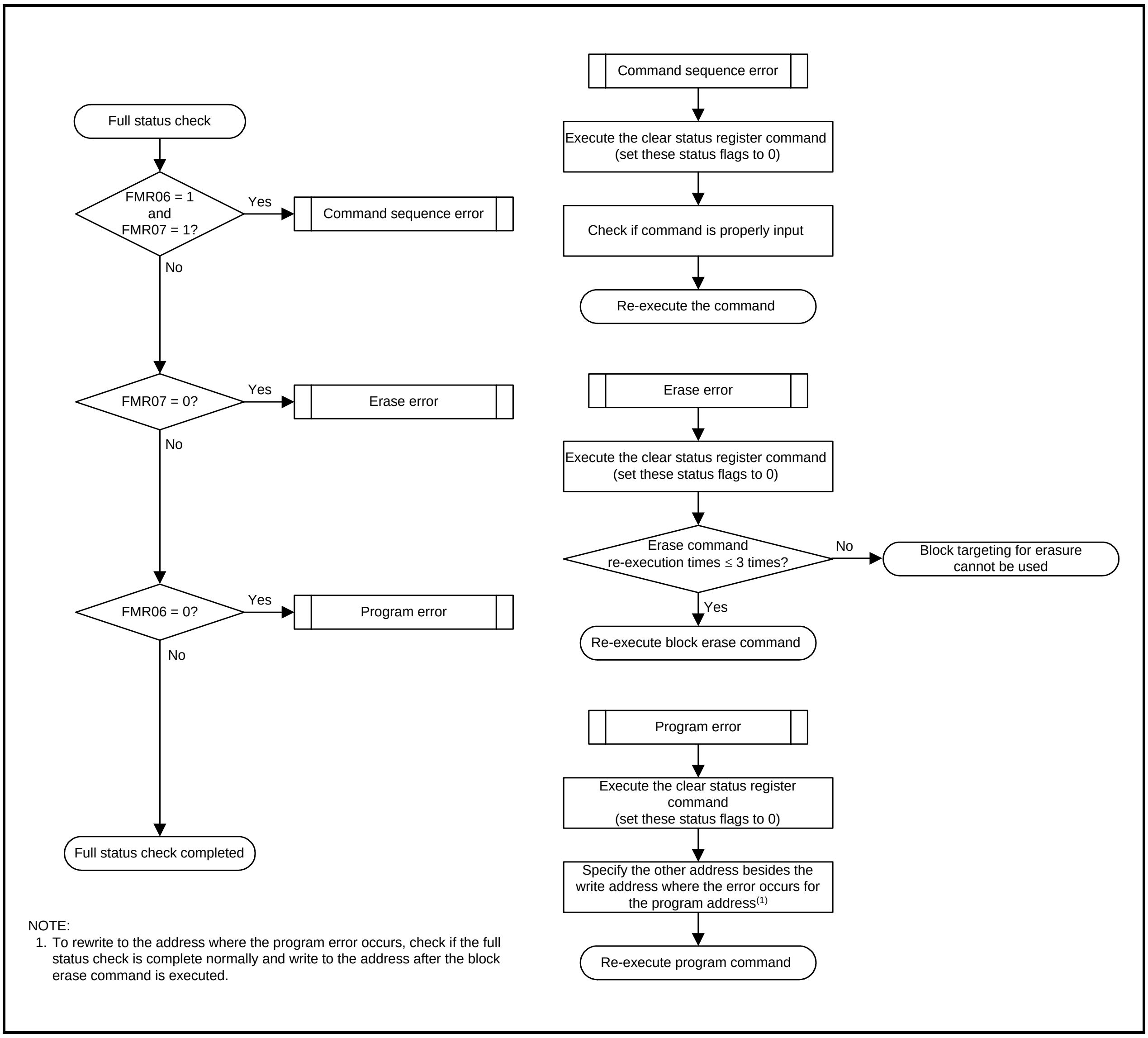
Table 17.6 lists the Errors and FMR0 Register Status. Figure 17.15 shows the Full Status Check and Handling Procedure for Individual Errors.

Table 17.6 Errors and FMR0 Register Status

FRM00 Register (Status Register) Status		Error	Error Occurrence Condition
FMR07(SR5)	FMR06(SR4)		
1	1	Command sequence error	• When a command is not written correctly. • When invalid data other than that which can be written in the second bus cycle of the block erase command is written (i.e., other than D0h or FFh)[1] • When the program command or block erase command is executed while rewriting is disabled using the FMR02 bit in the FMR0 register, or the FMR15 or FMR16 bit in the FMR1 register. • When an address not allocated in flash memory is input during erase command input. • When attempting to erase the block for which rewriting is disabled during erase command input. • When an address not allocated in flash memory is input during write command input. • When attempting to write the block for which rewriting is disabled during write command input.
1	0	Erase error	• When the block erase command is executed but auto-erasure does not complete correctly.
0	1	Program error	• When the program command is executed but auto-programming does not complete correctly.

NOTE:

1. The MCU enters read array mode when FFh is written in the second bus cycle of these commands. At the same time, the command code written in the first bus cycle is disabled.


Full status check
FMR06 = 1 and FMR07 = 1?
Yes
Command sequence error
No
FMR07 = 0?
Yes
Erase error
No
FMR06 = 0?
Yes
Program error
No
Full status check completed
Command sequence error
Execute the clear status register command (set these status flags to 0)
Check if command is properly input
Re-execute the command
Erase error
Execute the clear status register command (set these status flags to 0)
Erase command re-execution times ≤ 3 times?
No
Block targeting for erasure cannot be used
Yes
Re-execute block erase command
Program error
Execute the clear status register command (set these status flags to 0)
Specify the other address besides the write address where the error occurs for the program address(1)
Re-execute program command

NOTE:

1. To rewrite to the address where the program error occurs, check if the full status check is complete normally and write to the address after the block erase command is executed.

Figure 17.15 Full Status Check and Handling Procedure for Individual Errors

17.5 Standard Serial I/O Mode

In standard serial I/O mode, the user ROM area can be rewritten while the MCU is mounted on-board by using a serial programmer which is suitable for the MCU.
Standard serial I/O mode is used to connect with a serial programmer using a special clock asynchronous serial I/O.
There are three standard serial I/O modes:

- Standard serial I/O mode 1.......... Clock synchronous serial I/O used to connect with a serial programmer
- Standard serial I/O mode 2.......... Clock asynchronous serial I/O used to connect with a serial programmer
- Standard serial I/O mode 3.......... Special clock asynchronous serial I/O used to connect with a serial programmer

This MCU uses standard serial I/O mode 2 and standard serial I/O mode 3.

Refer to **Appendix 2. Connection Examples between Serial Writer and On-Chip Debugging Emulator.** Contact the manufacturer of your serial programmer for additional information. Refer to the user’s manual of your serial programmer for details on how to use it.

Table 17.7 lists the Pin Functions (Flash Memory Standard Serial I/O Mode 2), Table 17.8 lists the Pin Functions (Flash Memory Standard Serial I/O Mode 3). Figure 17.16 shows Pin Connections for Standard Serial I/O Mode 3.
After processing the pins shown in Table 17.8 and rewriting the flash memory using a programmer, apply “H” to the MODE pin and reset the hardware to run a program in the flash memory in single-chip mode.

17.5.1 ID Code Check Function

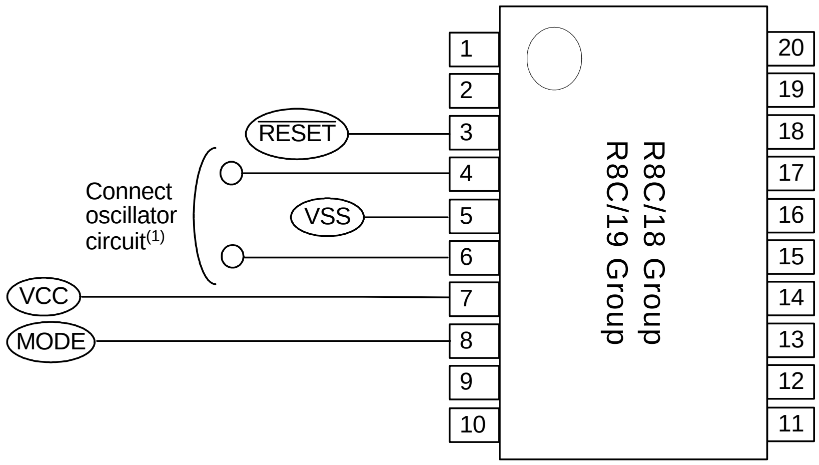
The ID code check function determines whether the ID codes sent from the serial programmer and those written in the flash memory match (refer to **17.3 Functions to Prevent Rewriting of Flash Memory**).

Table 17.7 Pin Functions (Flash Memory Standard Serial I/O Mode 2)

Pin	Name	I/O	Description
VCC,VSS	Power input		Apply the voltage guaranteed for programming and erasure to the VCC pin and 0 V to the VSS pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin.
P4_6/XIN	P4_6 input/clock input	I	Connect a ceramic resonator or crystal oscillator between pins XIN and XOUT.
P4_7/XOUT	P4_7 input/clock output	I/O	
AVCC, AVSS	Analog power supply input	I	Connect AVSS to VSS and AVCC to VCC, respectively.
P1_0 to P1_7	Input port P1	I	Input “H” or “L” level signal or leave the pin open.
P3_3 to P3_5	Input port P3	I	Input “H” or “L” level signal or leave the pin open.
P4_2/VREF	Input port P4	I	Input “H” or “L” level signal or leave the pin open.
MODE	MODE	I/O	Input “L”.
P3_7	TXD output	O	Serial data output pin.
P4_5	RXD input	I	Serial data input pin.

Table 17.8 Pin Functions (Flash Memory Standard Serial I/O Mode 3)

Pin	Name	I/O	Description
VCC,VSS	Power input		Apply the voltage guaranteed for programming and erasure to the VCC pin and 0 V to the VSS pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin.
P4_6/XIN	P4_6 input/clock input	I	Connect a ceramic resonator or crystal oscillator between pins XIN and XOUT when connecting external oscillator. Apply “H” and “L” or leave the pin open when using as input port
P4_7/XOUT	P4_7 input/clock output	I/O	
AVCC, AVSS	Analog power supply input	I	Connect AVSS to VSS and AVCC to VCC, respectively.
P1_0 to P1_7	Input port P1	I	Input “H” or “L” level signal or leave the pin open.
P3_3 to P3_5, P3_7	Input port P3	I	Input “H” or “L” level signal or leave the pin open.
P4_2/VREF, P4_5	Input port P4	I	Input “H” or “L” level signal or leave the pin open.
MODE	MODE	I/O	Serial data I/O pin. Connect to flash programmer.



Package: PLSP0020JB-A

NOTE:
1. It is not necessary to connect an oscillating circuit when operating with the on-chip oscillator clock.

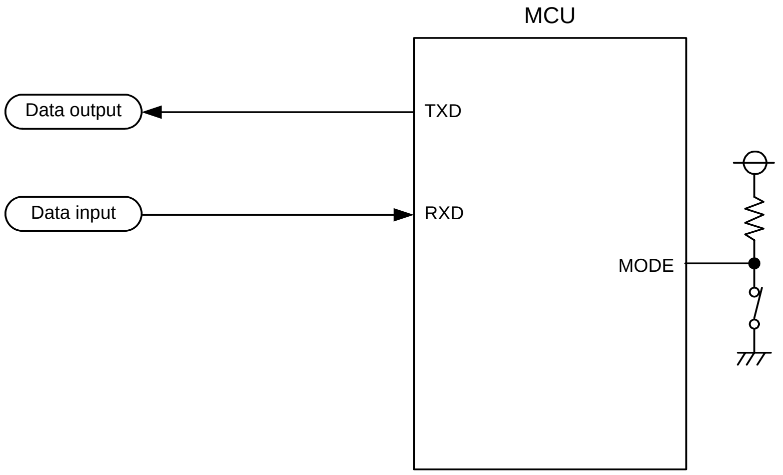
Mode Setting

Signal	Value
MODE	Voltage from programmer
$\overline{\text{RESET}}$	VSS → VCC

Figure 17.16 Pin Connections for Standard Serial I/O Mode 3

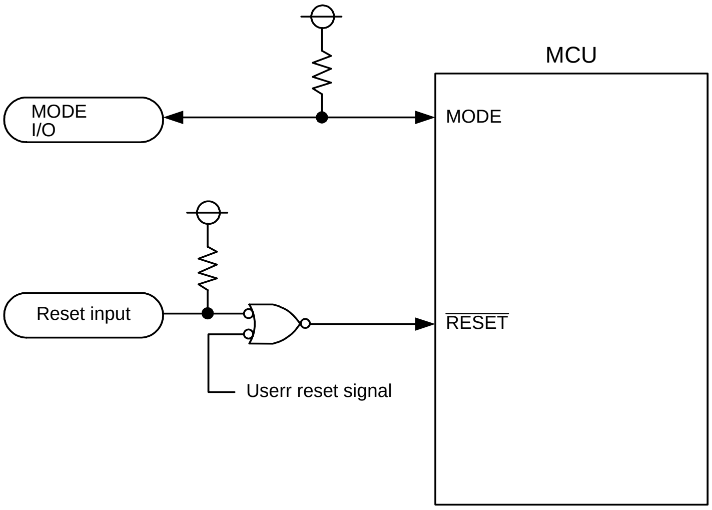
17.5.1.1 Example of Circuit Application in the Standard Serial I/O Mode

Figure 17.17 shows an example of Pin Processing in Standard Serial I/O Mode 2, and Figure 17.18 shows Pin Processing in Standard Serial I/O Mode 3. Since the controlled pins vary depending on the programmer, refer to the manual of your serial programmer for details.



(1) In this example, modes are switched between single-chip mode and standard serial I/O mode by controlling the MODE input with a switch.
(2) Connecting an oscillator is necessary. Set the main clock frequency to between 1 MHz and 20 MHz. Refer to **Appendix 2.1 Connection Example with M16C Flash Starter (M3A-0806)**.

Figure 17.17 Pin Processing in Standard Serial I/O Mode 2



(1) Controlled pins and external circuits vary depending on the programmer. Refer to the programmer manual for details.
(2) In this example, modes are switched between single-chip mode and standard serial I/O mode by connecting a programmer.
(3) When operating with the on-chip oscillator clock, it is not necessary to connection oscillating circuit.

Figure 17.18 Pin Processing in Standard Serial I/O Mode 3

17.6 Parallel I/O Mode

Parallel I/O mode is used to input and output software commands, addresses, and data necessary to control (read, program, and erase) the on-chip flash memory. Use a parallel programmer which supports this MCU. Contact the manufacturer of the parallel programmer for more information, and refer to the user’s manual of the parallel programmer for details on how to use it.

ROM areas shown in Figures 17.1 and 17.2 can be rewritten in parallel I/O mode.

17.6.1 ROM Code Protect Function

The ROM code protect function disables the reading and rewriting of the flash memory. (Refer to the **17.3 Functions to Prevent Rewriting of Flash Memory**.)

17.7 Notes on Flash Memory Version

17.7.1 CPU Rewrite Mode

17.7.1.1 Operating Speed

Before entering CPU rewrite mode (EW0 mode), select 5 MHz or below for the CPU clock using the CM06 bit in the CM0 register and bits CM16 to CM17 in the CM1 register. This does note apply to EW1 mode.

17.7.1.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because they reference data in the flash memory: UND, INTO, and BRK.

17.7.1.3 Interrupts

Table 17.9 lists the EW0 Mode Interrupts and Table 17.10 lists the EW1 Mode Interrupts.

Table 17.9 EW0 Mode Interrupts

Mode	Status	When Maskable Interrupt Request is Acknowledged	When Watchdog Timer, Oscillation Stop Detection and Voltage Monitor 2 Interrupt Request is Acknowledged
EW0	During auto-erasure	Any interrupt can be used by allocating a vector in RAM	Once an interrupt request is acknowledged, auto-programming or auto-erasure is forcibly stopped immediately and the flash memory is reset. Interrupt handing starts after the fixed period and the flash memory restarts. Since the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be read. Execute auto-erasure again and ensure it completes normally. Since the watchdog timer does not stop during the command operation, interrupt requests may be generated. Reset the watchdog timer regularly.
	Auto-programming		

NOTES:

1. Do not use the address match interrupt while a command is being executed because the vector of the address match interrupt is allocated in ROM.
2. Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

Table 17.10 EW1 Mode Interrupts

Mode	Status	When Maskable Interrupt Request is Acknowledged	When Watchdog Timer, Oscillation Stop Detection and Voltage Monitor 2 Interrupt Request are Acknowledged
EW1	During auto-erasure (erase- suspend function enabled)	Auto-erasure is suspended after td(SR-ES) and interrupt handing is executed. Auto-erasure can be restarted by setting the FMR41 bit in the FMR4 register to 0 (erase restart) after interrupt handing completes.	Once an interrupt request is acknowledged, auto-programming or auto-erasure is forcibly stopped immediately and the flash memory is reset. Interrupt handing starts after the fixed period and the flash memory restarts. Since the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be read. Execute auto-erasure again and ensure it completes normally. Since the watchdog timer does not stop during the command operation, interrupt requests may be generated. Reset the watchdog timer regularly using the erase-suspend function.
	During auto-erasure (erase- suspend function disabled)	Auto-erasure has priority and the interrupt request acknowledgement is put on standby. Interrupt handing is executed after auto-erasure completes.	
	During auto-programming (program suspend function enabled)	Auto-programming is suspended after td(SR-SUS) and interrupt handing is executed. Auto-programming can be restarted by setting the FMR42 bit in the FMR4 register to 0 (program restart) after interrupt handing completes.	
	During auto-programming (program suspend function disabled)	Auto-programming has priority and the interrupt request acknowledgement is put on standby. Interrupt handing is executed after auto-programming completes.	

NOTES:

1. Do not use the address match interrupt while a command is executing because the vector of the address match interrupt is allocated in ROM.
2. Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

17.7.1.4 How to Access

Write 0 before writing 1 when setting the FMR01, FMR02, or FMR11 bit to 1. Do not generate an interrupt between writing 0 and 1.

17.7.1.5 Rewriting User ROM Area

In EW0 Mode, if the supply voltage drops while rewriting any block in which a rewrite control program is stored, it may not be possible to rewrite the flash memory because the rewrite control program cannot be rewritten correctly. In this case, use standard serial I/O mode.

17.7.1.6 Program

Do not write additions to the already programmed address.

17.7.1.7 Entering Stop Mode or Wait Mode

Do not enter stop mode or wait mode during erase-suspend.

18. Electrical Characteristics

Table 18.1 Absolute Maximum Ratings

Symbol	Parameter	Condition	Rated Value	Unit
V_{CC}	Supply voltage	V_{CC} = AV_{CC}	-0.3 to 6.5	V
AV_{CC}	Analog supply voltage	V_{CC} = AV_{CC}	-0.3 to 6.5	V
V_I	Input voltage		-0.3 to V_{CC}+0.3	V
V_O	Output voltage		-0.3 to V_{CC}+0.3	V
P_d	Power dissipation	T_{opr} = 25°C	300	mW
T_{opr}	Operating ambient temperature		-20 to 85 / -40 to 85 (D version)	°C
T_{stg}	Storage temperature		-65 to 150	°C

Table 18.2 Recommended Operating Conditions

Symbol	Parameter		Conditions	Standard			Unit
				Min.	Typ.	Max.	
V_{CC}	Supply voltage			2.7	–	5.5	V
AV_{CC}	Analog supply voltage			–	V_{CC}	–	V
V_{SS}	Supply voltage			–	0	–	V
AV_{SS}	Analog supply voltage			–	0	–	V
V_{IH}	Input “H” voltage			0.8V_{CC}	–	V_{CC}	V
V_{IL}	Input “L” voltage			0	–	0.2V_{CC}	V
$I_{OH(sum)}$	Peak sum output “H” current	Sum of all pins I_{OH} (peak)		–	–	-60	mA
$I_{OH(peak)}$	Peak output “H” current			–	–	-10	mA
$I_{OH(avg)}$	Average output “H” current			–	–	-5	mA
$I_{OL(sum)}$	Peak sum output “L” currents	Sum of all pins I_{OL} (peak)		–	–	60	mA
$I_{OL(peak)}$	Peak output “L” currents	Except P1_0 to P1_3		–	–	10	mA
		P1_0 to P1_3	Drive capacity HIGH	–	–	30	mA
			Drive capacity LOW	–	–	10	mA
$I_{OL(avg)}$	Average output “L” current	Except P1_0 to P1_3		–	–	5	mA
		P1_0 to P1_3	Drive capacity HIGH	–	–	15	mA
			Drive capacity LOW	–	–	5	mA
$f_{(XIN)}$	Main clock input oscillation frequency		3.0 V ≤ V_{CC} ≤ 5.5 V	0	–	20	MHz
			2.7 V ≤ V_{CC} < 3.0 V	0	–	10	MHz

NOTES:

1. V_{CC} = 2.7 to 5.5 V at T_{opr} = -20 to 85 °C / -40 to 85 °C, unless otherwise specified.
2. Typical values when average output current is 100 ms.

Table 18.3 Comparator Characteristics

Symbol	Parameter	Conditions	Standard			Unit
			Min.	Typ.	Max.	
–	Resolution		–	–	1	Bit
–	Absolute accuracy	ϕ_{AD} = 10 MHz(3)	–	–	±20	mV
t_{conv}	Conversion time	ϕ_{AD} = 10 MHz(3)	1	–	–	µs
V_{ref}	Reference voltage		0	–	AV_{CC}	V
V_{IA}	Analog input voltage		0	–	AV_{CC}	V
–	Comparator conversion operating clock frequency(2)		1	–	10	MHz

NOTES:

1. V_{CC} = 2.7 to 5.5 V at T_{opr} = -20 to 85 °C / -40 to 85 °C, unless otherwise specified.
2. If f1 exceeds 10 MHz, divided f1 and ensure the comparator conversion operating clock frequency (ϕ_{AD}) is 10 MHz or below.
3. If AVcc is less than 4.2 V, divided f1 and ensure the comparator conversion operating clock frequency (ϕ_{AD}) is f1/2 or below.

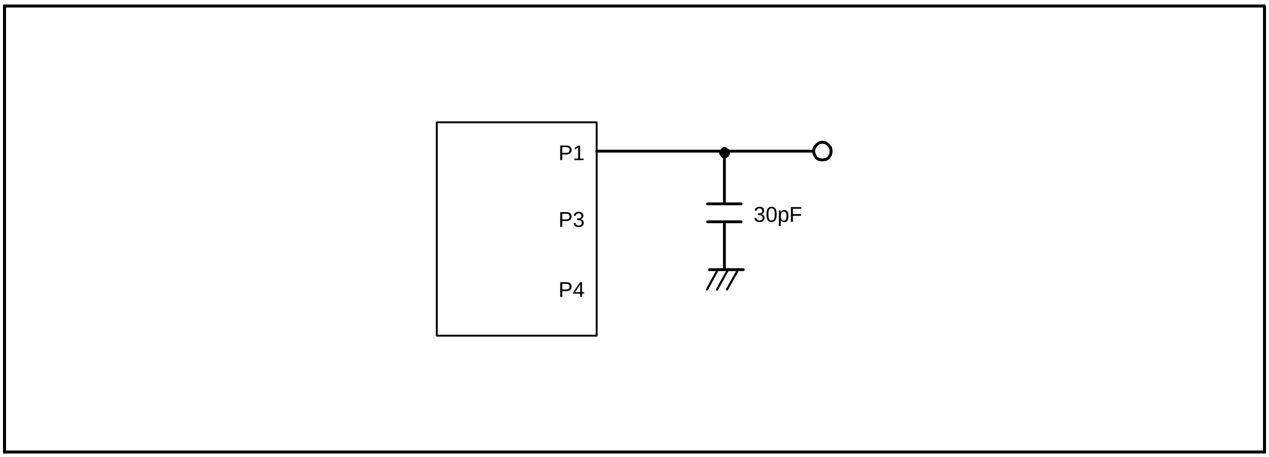


Figure 18.1 Port P1, P3, and P4 Measurement Circuit

Table 18.4 Flash Memory (Program ROM) Electrical Characteristics

Symbol	Parameter	Conditions	Standard			Unit
			Min.	Typ.	Max.	
–	Program/erase endurance[2]	R8C/18 Group	100[3]	–	–	times
		R8C/19 Group	1,000[3]	–	–	times
–	Byte program time		–	50	400	µs
–	Block erase time		–	0.4	9	s
$t_{d(SR-SUS)}$	Time delay from suspend request until suspend		–	–	97+CPU clock × 6 cycles	µs
–	Interval from erase start/restart until following suspend request		650	–	–	µs
–	Interval from program start/restart until following suspend request		0	–	–	ns
–	Time from suspend until program/erase restart		–	–	3+CPU clock × 4 cycles	µs
–	Program, erase voltage		2.7	–	5.5	V
–	Read voltage		2.7	–	5.5	V
–	Program, erase temperature		0	–	60	°C
–	Data hold time[8]	Ambient temperature = 55 °C	20	–	–	year

NOTES:

1. V_{CC} = 2.7 to 5.5 V at T_{opr} = 0 to 60 °C, unless otherwise specified.
2. Definition of programming/erasure endurance
   The programming and erasure endurance is defined on a per-block basis.
   If the programming and erasure endurance is n (n = 100 or 10,000), each block can be erased n times. For example, if 1,024 1-byte writes are performed to block A, a 1 Kbyte block, and then the block is erased, the programming/erasure endurance still stands at one. However, the same address must not be programmed more than once per erase operation (overwriting prohibited).
3. Endurance to guarantee all electrical characteristics after program and erase. (1 to Min. value can be guaranteed).
4. If emergency processing is required, a suspend request can be generated independent of this characteristic. In that case the normal time delay to Suspend can be applied to the request.However, we recommend that a suspend request with an interval of less than 650 µs is only used once because, if the suspend state continues, erasure cannot operate and the incidence of erasure error rises.
5. In a system that executes multiple programming operations, the actual erasure count can be reduced by writing to sequential addresses in turn so that as much of the block as possible is used up before performing an erase operation. For example, when programming groups of 16 bytes, the effective number of rewrites can be minimized by programming up to 128 groups before erasing them all in one operation. In addition, averaging the number of erase operations between block A and block B can further reduce the effective number of rewrites. It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.
6. If an error occurs during block erase, attempt to execute the clear status register command, then execute the block erase command at least three times until the erase error does not occur.
7. Customers desiring programming/erasure failure rate information should contact their Renesas technical support representative.
8. The data hold time includes time that the power supply is off or the clock is not supplied.

Table 18.5 Flash Memory (Data flash Block A, Block B) Electrical Characteristics

Symbol	Parameter	Conditions	Standard			Unit
			Min.	Typ.	Max.	
–	Program/erase endurance[2]		10,000[3]	–	–	times
–	Byte program time (Program/erase endurance ≤ 1,000 times)		–	50	400	µs
–	Byte program time (Program/erase endurance > 1,000 times)		–	65	–	µs
–	Block erase time (Program/erase endurance ≤ 1,000 times)		–	0.2	9	s
–	Block erase time (Program/erase endurance > 1,000 times)		–	0.3	–	s
$t_{d(SR-SUS)}$	Time delay from suspend request until suspend		–	–	97+CPU clock × 6 cycles	µs
–	Interval from erase start/restart until following suspend request		650	–	–	µs
–	Interval from program start/restart until following suspend request		0	–	–	ns
–	Time from suspend until program/erase restart		–	–	3+CPU clock × 4 cycles	µs
–	Program, erase voltage		2.7	–	5.5	V
–	Read voltage		2.7	–	5.5	V
–	Program, erase temperature		-20[8]	–	85	°C
–	Data hold time[9]	Ambient temperature = 55 °C	20	–	–	year

NOTES:

1. V_{CC} = 2.7 to 5.5 V at T_{opr} = -20 to 85 °C / -40 to 85 °C, unless otherwise specified.
2. Definition of programming/erasure endurance
   The programming and erasure endurance is defined on a per-block basis.
   If the programming and erasure endurance is n (n = 100 or 10,000), each block can be erased n times. For example, if 1,024 1-byte writes are performed to block A, a 1 Kbyte block, and then the block is erased, the programming/erasure endurance still stands at one. However, the same address must not be programmed more than once per erase operation (overwriting prohibited).
3. Endurance to guarantee all electrical characteristics after program and erase. (1 to Min. value can be guaranteed).
4. If emergency processing is required, a suspend request can be generated independent of this characteristic. In that case the normal time delay to suspend can be applied to the request. However, we recommend that a suspend request with an interval of less than 650 µs is only used once because, if the suspend state continues, erasure cannot operate and the incidence of erasure error rises.
5. In a system that executes multiple programming operations, the actual erasure count can be reduced by writing to sequential addresses in turn so that as much of the block as possible is used up before performing an erase operation. For example, when programming groups of 16 bytes, the effective number of rewrites can be minimized by programming up to 128 groups before erasing them all in one operation. It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.
6. If an error occurs during block erase, attempt to execute the clear status register command, then execute the block erase command at least three times until the erase error does not occur.
7. Customers desiring programming/erasure failure rate information should contact their Renesas technical support representative.
8. -40 °C for D version.
9. The data hold time includes time that the power supply is off or the clock is not supplied.

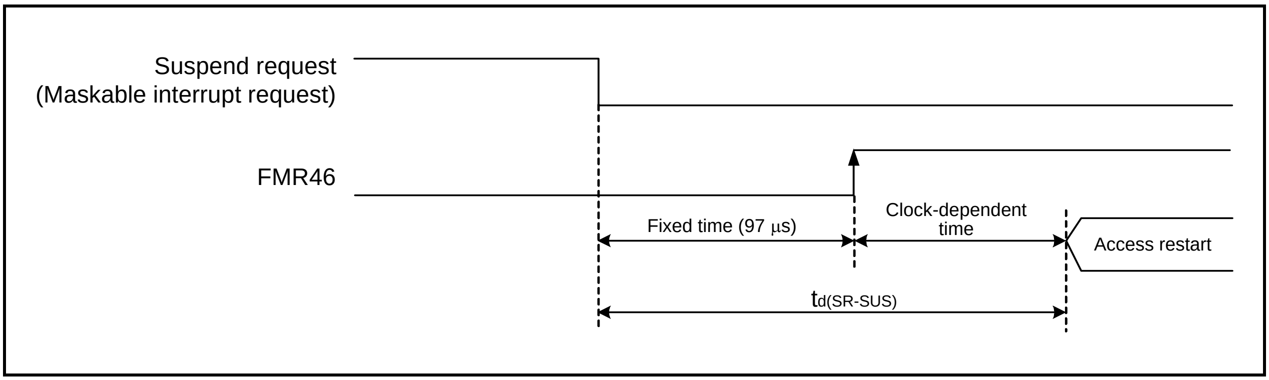


Figure 18.2 Transition Time to Suspend

Table 18.6 Voltage Detection 1 Circuit Electrical Characteristics

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V_{det1}	Voltage detection level(3)		2.70	2.85	3.00	V
–	Voltage detection circuit self power consumption	VCA26 = 1, V_{CC} = 5.0 V	–	600	–	nA
$t_{d(E-A)}$	Waiting time until voltage detection circuit operation starts(2)		–	–	100	µs
Vccmin	MCU operating voltage minimum value		2.7	–	–	V

NOTES:
1. The measurement condition is V_{CC} = 2.7 V to 5.5 V and T_{opr} = -40°C to 85 °C.
2. Necessary time until the voltage detection circuit operates when setting to 1 again after setting the VCA26 bit in the VCA2 register to 0.
3. Ensure that $V_{det2} > V_{det1}$.

Table 18.7 Voltage Detection 2 Circuit Electrical Characteristics

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V_{det2}	Voltage detection level(4)		3.00	3.30	3.60	V
–	Voltage monitor 2 interrupt request generation time(2)		–	40	–	µs
–	Voltage detection circuit self power consumption	VCA27 = 1, V_{CC} = 5.0 V	–	600	–	nA
$t_{d(E-A)}$	Waiting time until voltage detection circuit operation starts(3)		–	–	100	µs

NOTES:
1. The measurement condition is V_{CC} = 2.7 V to 5.5 V and T_{opr} = -40°C to 85 °C.
2. Time until the voltage monitor 2 interrupt request is generated after the voltage passes V_{det1}.
3. Necessary time until the voltage detection circuit operates when setting to 1 again after setting the VCA27 bit in the VCA2 register to 0.
4. Ensure that $V_{det2} > V_{det1}$.

Table 18.8 Reset Circuit Electrical Characteristics (When Using Voltage Monitor 1 Reset)

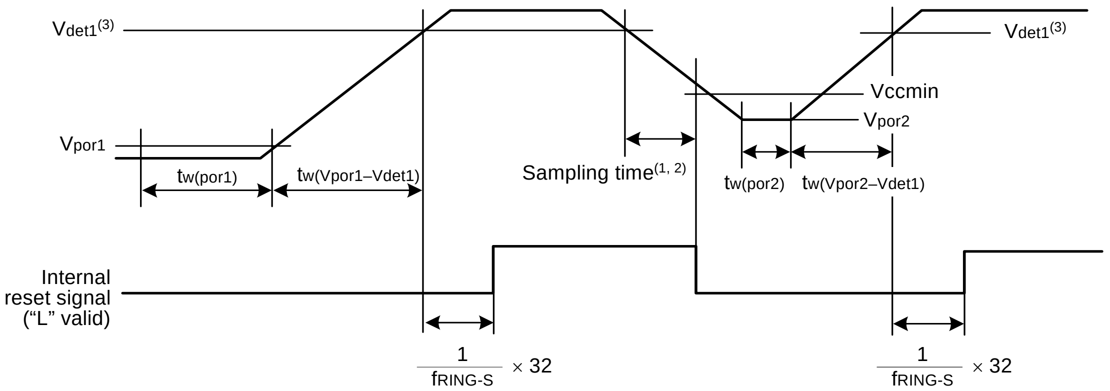
Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V_{por2}	Power-on reset valid voltage	-20°C ≤ Topr ≤ 85°C	–	–	V_{det1}	V
$t_{w(Vpor2-Vdet1)}$	Supply voltage rising time when power-on reset is deasserted(1)	-20°C ≤ Topr ≤ 85°C, $t_{w(por2)}$ ≥ 0s(3)	–	–	100	ms

NOTES:
1. This condition is not applicable when using with Vcc ≥ 1.0 V.
2. When turning power on after the time to hold the external power below effective voltage (V_{por1}) exceeds10 s, refer to **Table 18.9 Reset Circuit Electrical Characteristics (When Not Using Voltage Monitor 1 Reset)**.
3. $t_{w(por2)}$ is the time to hold the external power below effective voltage (V_{por2}).

Table 18.9 Reset Circuit Electrical Characteristics (When Not Using Voltage Monitor 1 Reset)

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V_{por1}	Power-on reset valid voltage	-20°C ≤ Topr ≤ 85°C	–	–	0.1	V
$t_{w(Vpor1-Vdet1)}$	Supply voltage rising time when power-on reset is deasserted	0°C ≤ Topr ≤ 85°C, $t_{w(por1)}$ ≥ 10 s(2)	–	–	100	ms
$t_{w(Vpor1-Vdet1)}$	Supply voltage rising time when power-on reset is deasserted	-20°C ≤ Topr < 0°C, $t_{w(por1)}$ ≥ 30 s(2)	–	–	100	ms
$t_{w(Vpor1-Vdet1)}$	Supply voltage rising time when power-on reset is deasserted	-20°C ≤ Topr < 0°C, $t_{w(por1)}$ ≥ 10 s(2)	–	–	1	ms
$t_{w(Vpor1-Vdet1)}$	Supply voltage rising time when power-on reset is deasserted	0°C ≤ Topr ≤ 85°C, $t_{w(por1)}$ ≥ 1 s(2)	–	–	0.5	ms

NOTES:
1. When not using voltage monitor 1, use with Vcc ≥ 2.7 V.
2. $t_{w(por1)}$ is the time to hold the external power below effective voltage (V_{por1}).



NOTES:
1. Hold the voltage inside the MCU operation voltage range (Vccmin or above) within the sampling time.
2. The sampling clock can be selected. Refer to **7. Voltage Detection Circuit** for details.
3. V_{det1} indicates the voltage detection level of the voltage detection 1 circuit. Refer to **7. Voltage Detection Circuit** for details.

Figure 18.3 Reset Circuit Electrical Characteristics

Table 18.10 High-speed On-Chip Oscillator Circuit Electrical Characteristics

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
–	High-speed on-chip oscillator frequency when the reset is deasserted	V_{CC} = 5.0 V, T_{opr} = 25 °C	–	8	–	MHz
–	High-speed on-chip oscillator frequency temperature supply voltage dependence(2)	0 to +60 °C/5 V ± 5 %(3)	7.76	–	8.24	MHz
		-20 to +85 °C/2.7 to 5.5 V(3)	7.68	–	8.32	MHz
		-40 to +85 °C/2.7 to 5.5 V(3)	7.44	–	8.32	MHz

NOTES:

1. The measurement condition is V_{CC} = 5.0 V and T_{opr} = 25 °C.
2. Refer to **10.6.4 High-Speed On-Chip Oscillator Clock** for notes on high-speed on-chip oscillator clock.
3. The standard value shows when the HRA1 register is assumed as the value in shipping and the HRA2 register value is set to 00h.

Table 18.11 Power Supply Circuit Timing Characteristics

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
$t_{d(P-R)}$	Time for internal power supply stabilization during power-on(2)		1	–	2000	µs
$t_{d(R-S)}$	STOP exit time(3)		–	–	150	µs

NOTES:

1. The measurement condition is V_{CC} = 2.7 to 5.5 V and T_{opr} = 25 °C.
2. Waiting time until the internal power supply generation circuit stabilizes during power-on.
3. Time until CPU clock supply starts after the interrupt is acknowledged to exit stop mode.

Table 18.12 Electrical Characteristics (1) [V_{CC} = 5 V]

Symbol	Parameter		Condition		Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	Output “H” voltage	Except X_{OUT}	I_{OH} = -5 mA		V_{CC} – 2.0	–	V_{CC}	V
			I_{OH} = -200 µA		V_{CC} – 0.3	–	V_{CC}	V
		X_{OUT}	Drive capacity HIGH	I_{OH} = -1 mA	V_{CC} – 2.0	–	V_{CC}	V
			Drive capacity LOW	I_{OH} = -500 µA	V_{CC} – 2.0	–	V_{CC}	V
V_{OL}	Output “L” voltage	Except P1_0 to P1_3, X_{OUT}	I_{OL} = 5 mA		–	–	2.0	V
			I_{OL} = 200 µA		–	–	0.45	V
		P1_0 to P1_3	Drive capacity HIGH	I_{OL} = 15 mA	–	–	2.0	V
			Drive capacity LOW	I_{OL} = 5 mA	–	–	2.0	V
			Drive capacity LOW	I_{OL} = 200 µA	–	–	0.45	V
		X_{OUT}	Drive capacity HIGH	I_{OL} = 1 mA	–	–	2.0	V
			Drive capacity LOW	I_{OL} = 500 µA	–	–	2.0	V
V_{T+}-V_{T-}	Hysteresis	$\overline{\text{INT0}}$, $\overline{\text{INT1}}$, $\overline{\text{INT2}}$, $\overline{\text{INT3}}$, KI0, KI1, KI2, KI3, CNTR0, CNTR1, TCIN, RXD0			0.2	–	1.0	V
		$\overline{\text{RESET}}$			0.2	–	2.2	V
I_{IH}	Input “H” current		VI = 5 V		–	–	5.0	µA
I_{IL}	Input “L” current		VI = 0 V		–	–	-5.0	µA
R_{PULLUP}	Pull-up resistance		VI = 0 V		30	50	167	kΩ
R_{fXIN}	Feedback resistance	XIN			–	1.0	–	MΩ
f_{RING-S}	Low-speed on-chip oscillator frequency				40	125	250	kHz
V_{RAM}	RAM hold voltage		During stop mode		2.0	–	–	V

NOTE:

1. V_{CC} = 4.2 to 5.5 V at T_{opr} = -20 to 85 °C / -40 to 85 °C, f(XIN) = 20 MHz, unless otherwise specified.

Table 18.13 Electrical Characteristics (2) [Vcc = 5 V] (Topr = -40 to 85 °C, unless otherwise specified.)

Symbol	Parameter	Condition		Standard			Unit
				Min.	Typ.	Max.	
I_{CC}	Power supply current (V_{CC} = 3.3 to 5.5 V) Single-chip mode, output pins are open, other pins are V_{SS}, comparator is stopped	High-speed mode	XIN = 20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	9	15	mA
			XIN = 16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	8	14	mA
			XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	5	–	mA
		Medium-speed mode	XIN = 20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	4	–	mA
			XIN = 16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	3	–	mA
			XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	2	–	mA
		High-speed on-chip oscillator mode	Main clock off High-speed on-chip oscillator on = 8 MHz Low-speed on-chip oscillator on = 125 kHz No division	–	4	8	mA
			Main clock off High-speed on-chip oscillator on = 8 MHz Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	1.5	–	mA
		Low-speed on-chip oscillator mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8 FMR47 = 1	–	110	300	µA
		Wait mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock operation VCA27 = VCA26 = 0	–	40	80	µA
		Wait mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock off VCA27 = VCA26 = 0	–	38	76	µA
		Stop mode	Main clock off, Topr = 25 °C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = 0	–	0.8	3.0	µA

Timing Requirements
(Unless Otherwise Specified: VCC = 5 V, VSS = 0 V at Ta = 25 °C) [VCC = 5 V]

Table 18.14 XIN Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(XIN)}$	XIN input cycle time	50	–	ns
$t_{WH(XIN)}$	XIN input “H” width	25	–	ns
$t_{WL(XIN)}$	XIN input “L” width	25	–	ns

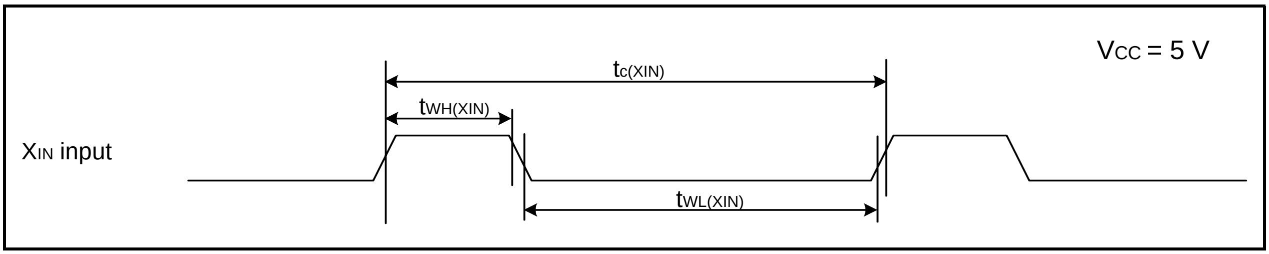


Figure 18.4 XIN Input Timing Diagram when VCC = 5 V

Table 18.15 CNTR0 Input, CNTR1 Input, $\overline{\text{INT1}}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CNTR0)}$	CNTR0 input cycle time	100	–	ns
$t_{WH(CNTR0)}$	CNTR0 input “H” width	40	–	ns
$t_{WL(CNTR0)}$	CNTR0 input “L” width	40	–	ns

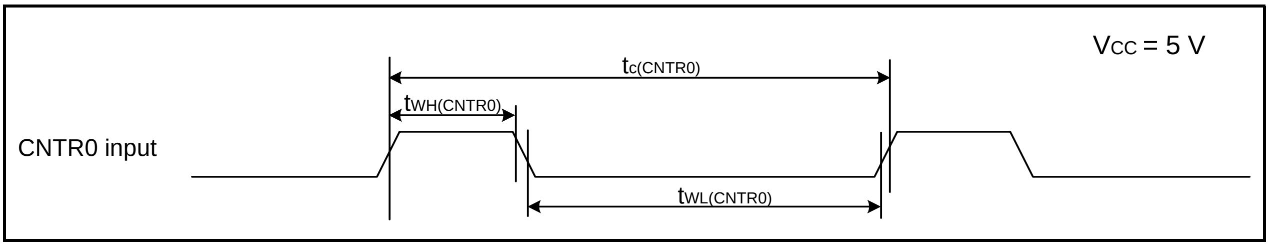


Figure 18.5 CNTR0 Input, CNTR1 Input, $\overline{\text{INT1}}$ Input Timing Diagram when VCC = 5 V

Table 18.16 TCIN Input, $\overline{\text{INT3}}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TCIN)}$	TCIN input cycle time	400[1]	–	ns
$t_{WH(TCIN)}$	TCIN input “H” width	200[2]	–	ns
$t_{WL(TCIN)}$	TCIN input “L” width	200[2]	–	ns

NOTES:
1. When using timer C input capture mode, adjust the cycle time to (1/timer C count source frequency x 3) or above.
2. When using timer C input capture mode, adjust the pulse width to (1/timer C count source frequency x 1.5) or above.

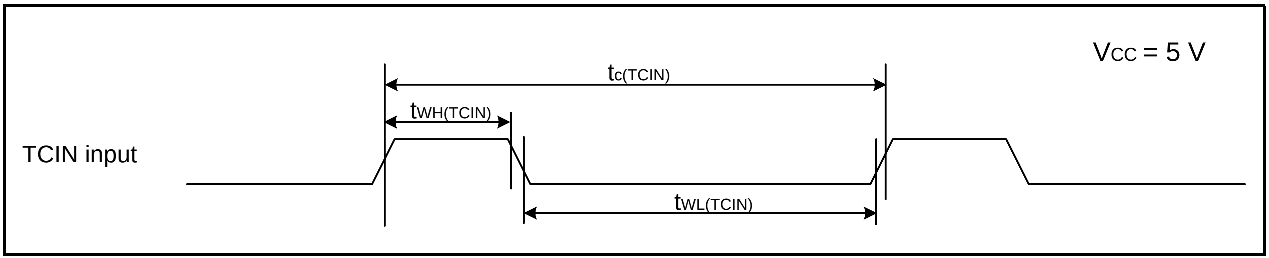


Figure 18.6 TCIN Input, $\overline{\text{INT3}}$ Input Timing Diagram when VCC = 5 V

Table 18.17 Serial Interface

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200	–	ns
$t_{W(CKH)}$	CLKi input “H” width	100	–	ns
$t_{W(CKL)}$	CLKi input “L” width	100	–	ns
$t_{d(C-Q)}$	TXDi output delay time	–	50	ns
$t_{h(C-Q)}$	TXDi hold time	0	–	ns
$t_{su(D-C)}$	RXDi input setup time	50	–	ns
$t_{h(C-D)}$	RXDi input hold time	90	–	ns

i = 0 or 1

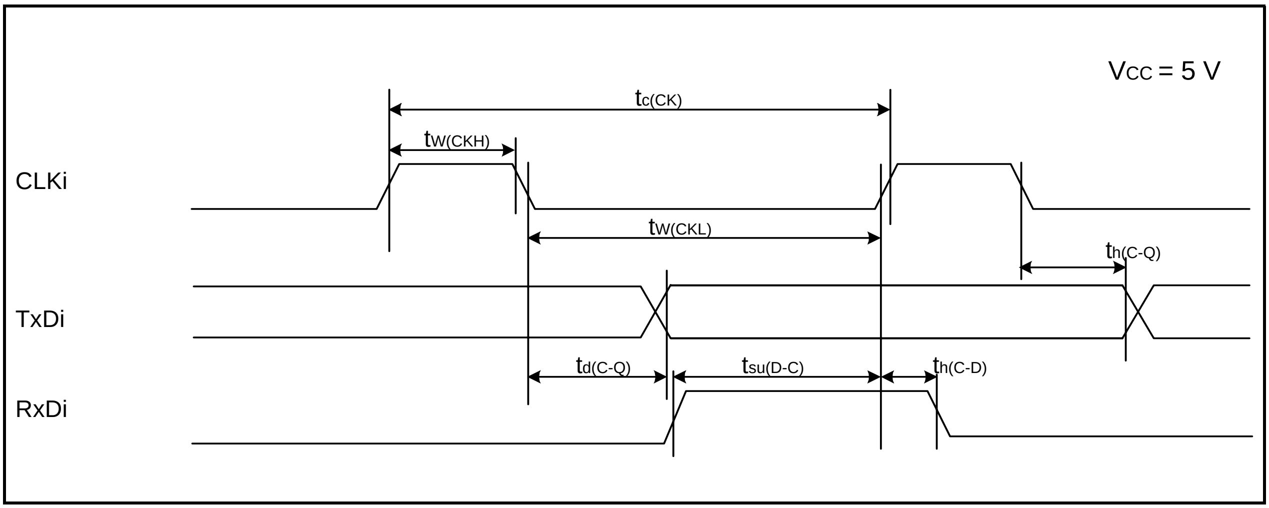


Figure 18.7 Serial Interface Timing Diagram when VCC = 5 V

Table 18.18 External Interrupt $\overline{\text{INT0}}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{W(INH)}$	$\overline{\text{INT0}}$ input “H” width	250[1]	–	ns
$t_{W(INL)}$	$\overline{\text{INT0}}$ input “L” width	250[2]	–	ns

NOTES:
1. When selecting the digital filter by the $\overline{\text{INT0}}$ input filter select bit, use an $\overline{\text{INT0}}$ input HIGH width of either (1/digital filter clock frequency x 3) or the minimum value of standard, whichever is greater.
2. When selecting the digital filter by the $\overline{\text{INT0}}$ input filter select bit, use an $\overline{\text{INT0}}$ input LOW width of either (1/digital filter clock frequency x 3) or the minimum value of standard, whichever is greater.

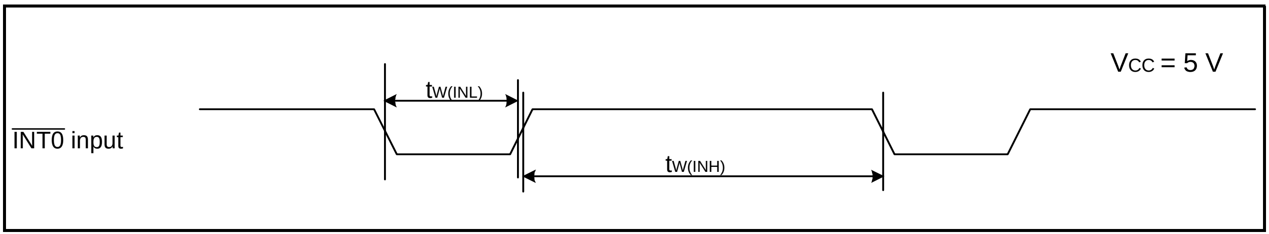


Figure 18.8 External Interrupt $\overline{\text{INT0}}$ Input Timing Diagram when VCC = 5 V

Table 18.19 Electrical Characteristics (3) [V_{CC} = 3V]

Symbol	Parameter		Condition		Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	Output “H” voltage	Except XOUT	I_{OH} = -1 mA		V_{CC} – 0.5	–	V_{CC}	V
		XOUT	Drive capacity HIGH	I_{OH} = -0.1 mA	V_{CC} – 0.5	–	V_{CC}	V
			Drive capacity LOW	I_{OH} = -50 μA	V_{CC} – 0.5	–	V_{CC}	V
V_{OL}	Output “L” voltage	Except P1_0 to P1_3, XOUT	I_{OL} = 1mA		–	–	0.5	V
		P1_0 to P1_3	Drive capacity HIGH	I_{OL} = 2 mA	–	–	0.5	V
			Drive capacity LOW	I_{OL} = 1 mA	–	–	0.5	V
		XOUT	Drive capacity HIGH	I_{OL} = 0.1 mA	–	–	0.5	V
			Drive capacity LOW	I_{OL} = 50 μA	–	–	0.5	V
V_{T+}-V_{T-}	Hysteresis	$\overline{\text{INT0}}$, $\overline{\text{INT1}}$, $\overline{\text{INT2}}$, $\overline{\text{INT3}}$, KI0, KI1, KI2, KI3, CNTR0, CNTR1, TCIN, RXD0			0.2	–	0.8	V
		$\overline{\text{RESET}}$			0.2	–	1.8	V
I_{IH}	Input “H” current		VI = 3 V		–	–	4.0	μA
I_{IL}	Input “L” current		VI = 0 V		–	–	-4.0	μA
R_{PULLUP}	Pull-up resistance		VI = 0 V		66	160	500	kΩ
R_{fXIN}	Feedback resistance	XIN			–	3.0	–	MΩ
f_{RING-S}	Low-speed on-chip oscillator frequency				40	125	250	kHz
V_{RAM}	RAM hold voltage		During stop mode		2.0	–	–	V

NOTE:

1. V_{CC} = 2.7 to 3.3 V at T_{opr} = -20 to 85 °C / -40 to 85 °C, f(XIN) = 10 MHz, unless otherwise specified.

Table 18.20 Electrical Characteristics (4) [Vcc = 3V] (Topr = -40 to 85 °C, unless otherwise specified.)

Symbol	Parameter	Condition		Standard			Unit
				Min.	Typ.	Max.	
ICC	Power supply current (VCC = 2.7 to 3.3 V) Single-chip mode, output pins are open, other pins are VSS, comparator is stopped	High-speed mode	XIN = 20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	8	13	mA
			XIN = 16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	7	12	mA
			XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	5	–	mA
		Medium-speed mode	XIN = 20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	3	–	mA
			XIN = 16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	2.5	–	mA
			XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	1.6	–	mA
		High-speed on-chip oscillator mode	Main clock off High-speed on-chip oscillator on = 8 MHz Low-speed on-chip oscillator on = 125 kHz No division	–	3.5	7.5	mA
			Main clock off High-speed on-chip oscillator on = 8 MHz Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	1.5	–	mA
		Low-speed on-chip oscillator mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8 FMR47 = 1	–	100	280	µA
		Wait mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock operation VCA27 = VCA26 = 0	–	37	74	µA
		Wait mode	Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock off VCA27 = VCA26 = 0	–	35	70	µA
		Stop mode	Main clock off, Topr = 25 °C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = 0	–	0.7	3.0	µA

Timing requirements
(Unless Otherwise Specified: V_{CC} = 3 V, V_{SS} = 0 V at Ta = 25 °C) [V_{CC} = 3 V]

Table 18.21 XIN Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(XIN)}$	XIN input cycle time	100	–	ns
$t_{WH(XIN)}$	XIN input “H” width	40	–	ns
$t_{WL(XIN)}$	XIN input “L” width	40	–	ns

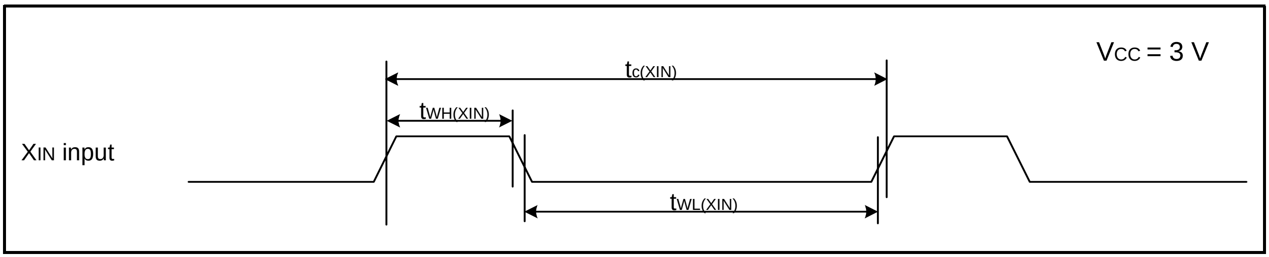


Figure 18.9 XIN Input Timing Diagram when V_{CC} = 3 V

Table 18.22 CNTR0 Input, CNTR1 Input, $\overline{INT1}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CNTR0)}$	CNTR0 input cycle time	300	–	ns
$t_{WH(CNTR0)}$	CNTR0 input “H” width	120	–	ns
$t_{WL(CNTR0)}$	CNTR0 input “L” width	120	–	ns

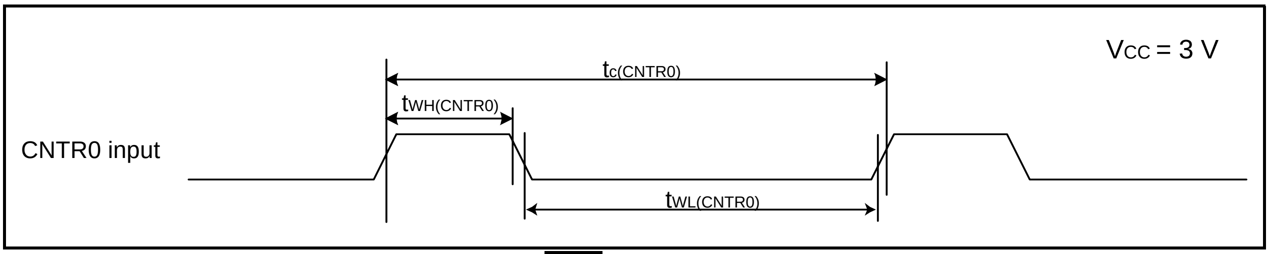


Figure 18.10 CNTR0 Input, CNTR1 Input, $\overline{INT1}$ Input Timing Diagram when V_{CC} = 3 V

Table 18.23 TCIN Input, $\overline{INT3}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TCIN)}$	TCIN input cycle time	1,200[1]	–	ns
$t_{WH(TCIN)}$	TCIN input “H” width	600[2]	–	ns
$t_{WL(TCIN)}$	TCIN input “L” width	600[2]	–	ns

NOTES:
1. When using the timer C input capture mode, adjust the cycle time to (1/timer C count source frequency × 3) or above.
2. When using the timer C input capture mode, adjust the width to (1/timer C count source frequency × 1.5) or above.

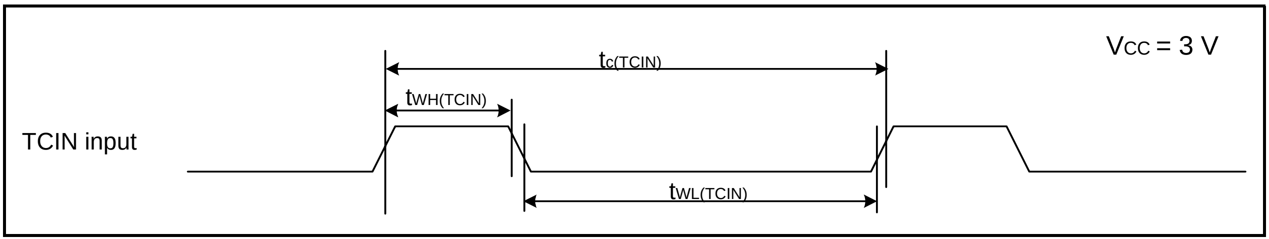


Figure 18.11 TCIN Input, $\overline{INT3}$ Input Timing Diagram when V_{CC} = 3 V

Table 18.24 Serial Interface

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300	–	ns
$t_{W(CKH)}$	CLKi input “H” width	150	–	ns
$t_{W(CKL)}$	CLKi input “L” width	150	–	ns
$t_{d(C-Q)}$	TXDi output delay time	–	80	ns
$t_{h(C-Q)}$	TXDi hold time	0	–	ns
$t_{su(D-C)}$	RXDi input setup time	70	–	ns
$t_{h(C-D)}$	RXDi input hold time	90	–	ns

i = 0 or 1

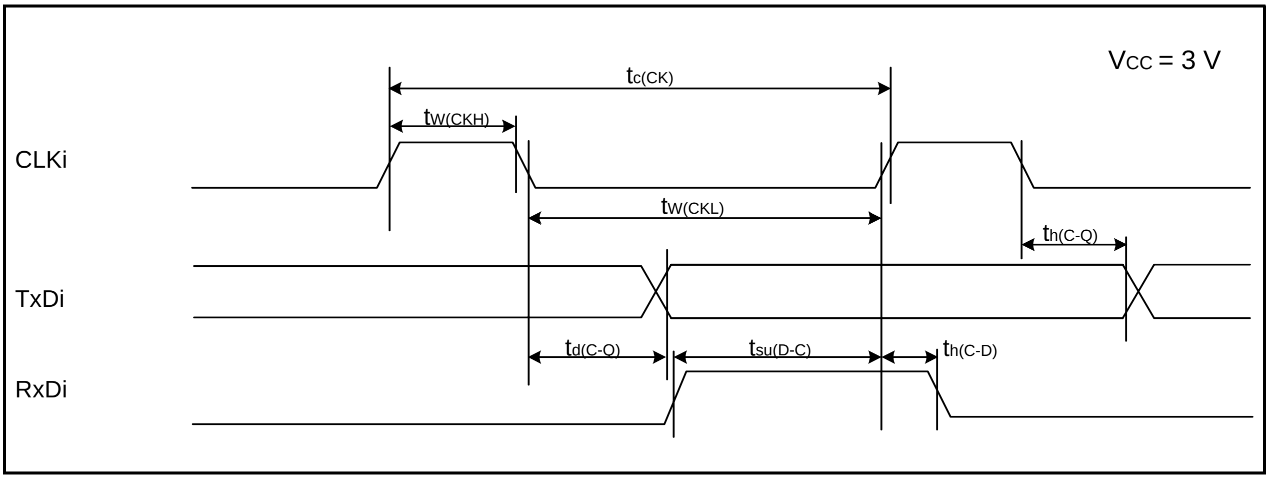


Figure 18.12 Serial Interface Timing Diagram when V_{CC} = 3 V

Table 18.25 External Interrupt $\overline{INT0}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{W(INH)}$	$\overline{INT0}$ input “H” width	380[1]	–	ns
$t_{W(INL)}$	$\overline{INT0}$ input “L” width	380[2]	–	ns

NOTES:

1. When selecting the digital filter by the $\overline{INT0}$ input filter select bit, use an $\overline{INT0}$ input HIGH width of either (1/digital filter clock frequency x 3) or the minimum value of standard, whichever is greater.
2. When selecting the digital filter by the $\overline{INT0}$ input filter select bit, use an $\overline{INT0}$ input LOW width of either (1/digital filter clock frequency x 3) or the minimum value of standard, whichever is greater.

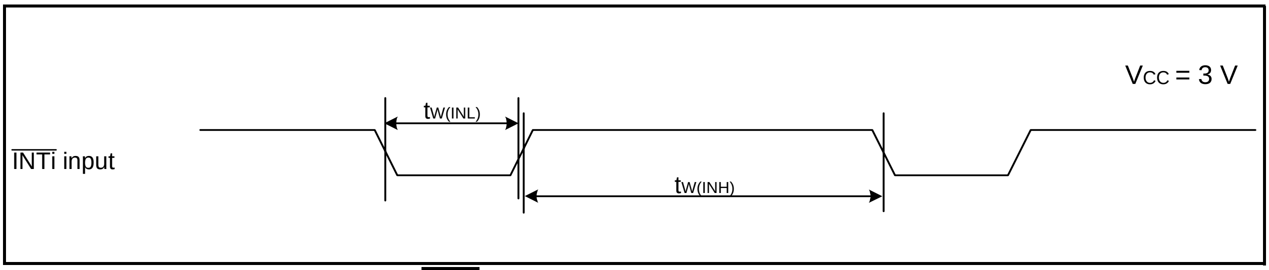


Figure 18.13 External Interrupt $\overline{INT0}$ Input Timing Diagram when V_{CC} = 3 V

19. Usage Notes

19.1 Notes on Clock Generation Circuit

19.1.1 Stop Mode and Wait Mode

When entering stop mode or wait mode, an instruction queue pre-reads 4 bytes from the WAIT instruction or an instruction that sets the CM10 bit in the CM1 register to 1 (stops all clocks) before the program stops. Therefore, insert at least four NOPs after the WAIT instruction or an instruction that sets the CM10 bit to 1.

19.1.2 Oscillation Stop Detection Function

Since the oscillation stop detection function cannot be used if the main clock frequency is below 2 MHz, set bits OCD1 to OCD0 to 00b (oscillation stop detection function disabled) in this case.

19.1.3 Oscillation Circuit Constants

Ask the manufacturer of the oscillator to specify the best oscillation circuit constants for your system.

19.1.4 High-Speed On-Chip Oscillator Clock

The high-speed on-chip oscillator frequency may be changed up to 10%(1) in flash memory CPU rewrite mode during auto-program operation or auto-erase operation.
The high-speed on-chip oscillator frequency after auto-program operation ends or auto-erase operation ends is held the state before the program command or block erase command is generated. Also, this note is not applicable when the read array command, read status register command, or clear status register command is generated. The application products must be designed with careful considerations for the frequency change.

NOTE:

1. Change ratio to 8 MHz frequency adjusted in shipping.

19.2 Notes on Interrupts

19.2.1 Reading Address 00000h

Do not read address 00000h by a program. When a maskable interrupt request is acknowledged, the CPU reads interrupt information (interrupt number and interrupt request level) from 00000h in the interrupt sequence. At this time, the acknowledged interrupt IR bit is set to 0.
If address 00000h is read by a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is set to 0. This may cause the interrupt to be canceled, or an unexpected interrupt to be generated.

19.2.2 SP Setting

Set any value in the SP before an interrupt is acknowledged. The SP is set to 0000h after reset. Therefore, if an interrupt is acknowledged before setting a value in the SP, the program may run out of control.

19.2.3 External Interrupt and Key Input Interrupt

Either "L" level or "H" level of at least 250 ns width is necessary for the signal input to pins $\overline{\text{INT0}}$ to $\overline{\text{INT3}}$ and pins $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$ regardless of the CPU clock.

19.2.4 Watchdog Timer Interrupt

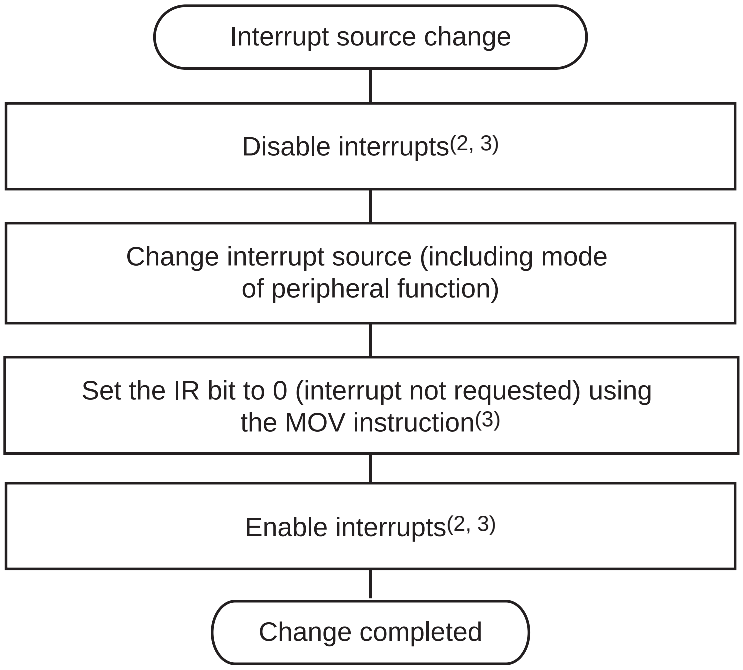
Reset the watchdog timer after a watchdog timer interrupt is generated.

19.2.5 Changing Interrupt Sources

The IR bit in the interrupt control register may be set to 1 (interrupt requested) when the interrupt source changes. When using an interrupt, set the IR bit to 0 (no interrupt requested) after changing the interrupt source.

In addition, changes of interrupt sources include all factors that change the interrupt sources assigned to individual software interrupt numbers, polarities, and timing. Therefore, if a mode change of a peripheral function involves interrupt sources, edge polarities, and timing, set the IR bit to 0 (no interrupt requested) after the change. Refer to the individual peripheral function for its related interrupts.

Figure 19.1 shows an Example of Procedure for Changing Interrupt Sources.


Interrupt source change
Disable interrupts(2, 3)
Change interrupt source (including mode of peripheral function)
Set the IR bit to 0 (interrupt not requested) using the MOV instruction(3)
Enable interrupts(2, 3)
Change completed

IR bit: The interrupt control register bit of an interrupt whose source is changed.

NOTES:

1. Execute the above settings individually. Do not execute two or more settings at once (by one instruction).
2. Use the I flag for the $\overline{\text{INTi}}$ (i = 0 to 3) interrupts.
   To prevent interrupt requests from being generated when using peripheral function interrupts other than the $\overline{\text{INTi}}$ interrupt, disable the peripheral function before changing the interrupt source. In this case, use the I flag if all maskable interrupts can be disabled. If all maskable interrupts cannot be disabled, use bits ILVL0 to ILVL2 of the interrupt whose source is changed.
3. Refer to **12.5.6 Changing Interrupt Control Register Contents** for the instructions to be used and usage notes.

Figure 19.1 Example of Procedure for Changing Interrupt Sources

19.2.6 Changing Interrupt Control Register Contents

(a) The contents of an interrupt control register can only be changed while no interrupt requests corresponding to that register are generated. If interrupt requests may be generated, disable interrupts before changing the interrupt control register contents.

(b) When changing the contents of an interrupt control register after disabling interrupts, be careful to choose appropriate instructions.

Changing any bit other than IR bit

If an interrupt request corresponding to a register is generated while executing the instruction, the IR bit may not be set to 1 (interrupt requested), and the interrupt request may be ignored. If this causes a problem, use the following instructions to change the register: AND, OR, BCLR, BSET

Changing IR bit

If the IR bit is set to 0 (interrupt not requested), it may not be set to 0 depending on the instruction used. Therefore, use the MOV instruction to set the IR bit to 0.

(c) When disabling interrupts using the I flag, set the I flag as shown in the sample programs below. Refer to (b) regarding changing the contents of interrupt control registers by the sample programs.

Sample programs 1 to 3 are for preventing the I flag from being set to 1 (interrupts enabled) before the interrupt control register is changed for reasons of the internal bus or the instruction queue buffer.

Example 1: Use NOP instructions to prevent I flag from being set to 1 before interrupt control register is changed

```
INT_SWITCH1:
    FCLR    I              ; Disable interrupts
    AND.B   #00H,0056H     ; Set TXIC register to 00h
    NOP                    ;
    NOP
    FSET    I              ; Enable interrupts
```

Example 2: Use dummy read to delay FSET instruction

```
INT_SWITCH2:
    FCLR    I              ; Disable interrupts
    AND.B   #00H,0056H     ; Set TXIC register to 00h
    MOV.W   MEM,R0         ; Dummy read
    FSET    I              ; Enable interrupts
```

Example 3: Use POPC instruction to change I flag

```
INT_SWITCH3:
    PUSHC   FLG
    FCLR    I              ; Disable interrupts
    AND.B   #00H,0056H     ; Set TXIC register to 00h
    POPC    FLG            ; Enable interrupts
```

19.3 Notes on Timers

19.3.1 Notes on Timer X

- Timer X stops counting after a reset. Set the values in the timer and prescaler before the count starts.
- Even if the prescaler and timer are read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- Do not rewrite bits TXMOD0 to TXMOD1, and bits TXMOD2 and TXS simultaneously.
- In pulse period measurement mode, bits TXEDG and TXUND in the TXMR register can be set to 0 by writing 0 to these bits by a program. However, these bits remain unchanged if 1 is written. When using the READ-MODIFY-WRITE instruction for the TXMR register, the TXEDG or TXUND bit may be set to 0 although these bits are set to 1 while the instruction is being executed. In this case, write 1 to the TXEDG or TXUND bit which is not supposed to be set to 0 with the MOV instruction.
- When changing to pulse period measurement mode from another mode, the contents of bits TXEDG and TXUND are undefined. Write 0 to bits TXEDG and TXUND before the count starts.
- The TXEDG bit may be set to 1 by the prescaler X underflow generated after the count starts.
- When using the pulse period measurement mode, leave two or more periods of the prescaler X immediately after the count starts, then set the TXEDG bit to 0.
- The TXS bit in the TXMR register has a function to instruct timer X to start or stop counting and a function to indicate that the count has started or stopped.
  0 (count stops) can be read until the following count source is applied after 1 (count starts) is written to the TXS bit while the count is being stopped. If the following count source is applied, 1 can be read from the TXS bit. After writing 1 to the TXS bit, do not access registers associated with timer X (registers TXMR, PREX, TX, TCSS, and TXIC) except for the TXS bit, until 1 can be read from the TXS bit. The count starts at the following count source after the TXS bit is set to 1.
  Also, after writing 0 (count stops) to the TXS bit during the count, timer X stops counting at the following count source.
  1 (count starts) can be read by reading the TXS bit until the count stops after writing 0 to the TXS bit. After writing 0 to the TXS bit, do not access registers associated with timer X except for the TXS bit, until 0 can be read from the TXS bit.

19.3.2 Notes on Timer Z

- Timer Z stops counting after a reset. Set the values in the timer and prescaler before the count starts.
- Even if the prescaler and timer are read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- Do not rewrite bits TZMOD0 to TZMOD1, and the TZS bit simultaneously.
- In programmable one-shot generation mode and programmable wait one-shot generation mode, when setting the TZS bit in the TZMR register to 0 (stops counting) or setting the TZOS bit in the TZOC register to 0 (stops one-shot), the timer reloads the value of the reload register and stops. Therefore, in programmable one-shot generation mode and programmable wait one-shot generation mode read the timer count value before the timer stops.
- The TZS bit in the TZMR register has a function to instruct timer Z to start or stop counting and a function to indicate that the count has started or stopped.
  0 (count stops) can be read until the following count source is applied after 1 (count starts) is written to the TZS bit while the count is being stopped. If the following count source is applied, 1 can be read from the TZS bit. After writing 1 to the TZS bit, do not access registers associated with timer Z (registers TZMR, PREZ, TZSC, TZPR, TZOC, PUM, TCSC, and TZIC) except for the TZS bit, until 1 can be read from the TZS bit. The count starts at the following count source after the TZS bit is set to 1.
  Also, after writing 0 (count stops) to the TZS bit during the count, timer Z stops counting at the following count source.
  1 (count starts) can be read by reading the TZS bit until the count stops after writing 0 to the TZS bit. After writing 0 to the TZS bit, do not access registers associated with timer Z except for the TZS bit, until 0 can be read from the TZS bit.

19.3.3 Notes on Timer C

Access registers TC, TM0, and TM1 in 16-bit units.
The TC register can be read in 16-bit units. This prevents the timer value from being updated between when the low-order bytes and high-order bytes are being read.

Example of reading timer C:

```
    MOV.W     0090H,R0       ; Read out timer C
```

19.4 Notes on Serial Interface

- When reading data from the U0RB register either in the clock asynchronous serial I/O mode or in the clock synchronous serial I/O mode. Ensure the data is read in 16-bit units. When the high-order byte of the U0RB register is read, bits PER and FER in the U0RB register and the RI bit in the U0C1 register are set to 0.

Example (when reading receive buffer register):

```
MOV.W      00A6H,R0        ; Read the U0RB register
```

- When writing data to the U0TB register in the clock asynchronous serial I/O mode with 9-bit transfer data length, write data to the high-order byte first then the low-order byte, in 8-bit units.

Example (when reading transmit buffer register):

```
MOV.B      #XXH,00A3H    ; Write the high-order byte of U0TB register
MOV.B      #XXH,00A2H    ; Write the low-order byte of U0TB register
```

19.5 Notes on Comparator

- Write to each bit (other than bit 6) in the ADCON0 register, each bit in the ADCON1 register, or the CMPSEL bit in the ADCON2 register when the comparator conversion stops (before a trigger occurs).
- When changing comparator conversion operating mode, select an analog input pin again.
- To use in one-shot mode, ensure that the comparator conversion is completed and the AD register is read. The IR bit in the ADIC register or the ADST bit in the ADCON0 register can determine whether the comparator conversion is completed.
- To use in repeat mode, use the undivided main clock as the CPU clock.
- If the ADST bit in the ADCON0 register is set to 0 (comparator conversion stops) by a program and the comparator conversion is forcibly terminated during the comparator conversion operation, the conversion result of the comparator will be indeterminate. If the ADST bit is set to 0 by a program, do not use the AD register value.
- Connect a 0.1 μF capacitor between the VCC/AVCC pin and VSS/AVSS pin.

19.6 Notes on Flash Memory Version

19.6.1 CPU Rewrite Mode

19.6.1.1 Operating Speed

Before entering CPU rewrite mode (EW0 mode), select 5 MHz or below for the CPU clock using the CM06 bit in the CM0 register and bits CM16 to CM17 in the CM1 register. This does note apply to EW1 mode.

19.6.1.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because they reference data in the flash memory: UND, INTO, and BRK.

19.6.1.3 Interrupts

Table 19.1 lists the EW0 Mode Interrupts and Table 19.2 lists the EW1 Mode Interrupts.

Table 19.1 EW0 Mode Interrupts

Mode	Status	When Maskable Interrupt Request is Acknowledged	When Watchdog Timer, Oscillation Stop Detection and Voltage Monitor 2 Interrupt Request is Acknowledged
EW0	During auto-erasure	Any interrupt can be used by allocating a vector in RAM	Once an interrupt request is acknowledged, auto-programming or auto-erasure is forcibly stopped immediately and the flash memory is reset. Interrupt handing starts after the fixed period and the flash memory restarts. Since the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be read. Execute auto-erasure again and ensure it completes normally. Since the watchdog timer does not stop during the command operation, interrupt requests may be generated. Reset the watchdog timer regularly.
	Auto-programming		

NOTES:

1. Do not use the address match interrupt while a command is being executed because the vector of the address match interrupt is allocated in ROM.
2. Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

Table 19.2 EW1 Mode Interrupts

Mode	Status	When Maskable Interrupt Request is Acknowledged	When Watchdog Timer, Oscillation Stop Detection and Voltage Monitor 2 Interrupt Request are Acknowledged
EW1	During auto-erasure (erase-suspend function enabled)	Auto-erasure is suspended after td(SR-ES) and interrupt handing is executed. Auto-erasure can be restarted by setting the FMR41 bit in the FMR4 register to 0 (erase restart) after interrupt handing completes.	Once an interrupt request is acknowledged, auto-programming or auto-erasure is forcibly stopped immediately and the flash memory is reset. Interrupt handing starts after the fixed period and the flash memory restarts. Since the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be read. Execute auto-erasure again and ensure it completes normally. Since the watchdog timer does not stop during the command operation, interrupt requests may be generated. Reset the watchdog timer regularly using the erase-suspend function.
	During auto-erasure (erase-suspend function disabled)	Auto-erasure has priority and the interrupt request acknowledgement is put on standby. Interrupt handing is executed after auto-erasure completes.	
	During auto-programming (program suspend function enabled)	Auto-programming is suspended after td(SR-SUS) and interrupt handing is executed. Auto-programming can be restarted by setting the FMR42 bit in the FMR4 register to 0 (program restart) after interrupt handing completes.	
	During auto-programming (program suspend function disabled)	Auto-programming has priority and the interrupt request acknowledgement is put on standby. Interrupt handing is executed after auto-programming completes.	

NOTES:

1. Do not use the address match interrupt while a command is executing because the vector of the address match interrupt is allocated in ROM.
2. Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

19.6.1.4 How to Access

Write 0 before writing 1 when setting the FMR01, FMR02, or FMR11 bit to 1. Do not generate an interrupt between writing 0 and 1.

19.6.1.5 Rewriting User ROM Area

In EW0 Mode, if the supply voltage drops while rewriting any block in which a rewrite control program is stored, it may not be possible to rewrite the flash memory because the rewrite control program cannot be rewritten correctly. In this case, use standard serial I/O mode.

19.6.1.6 Program

Do not write additions to the already programmed address.

19.6.1.7 Entering Stop Mode or Wait Mode

Do not enter stop mode or wait mode during erase-suspend.

19.7 Notes on Noise

19.7.1 Inserting a Bypass Capacitor between VCC and VSS Pins as a Countermeasure against Noise and Latch-Up

Connect a bypass capacitor (at least 0.1 μF) using the shortest and thickest wire possible.

19.7.2 Countermeasures against Noise Error of Port Control Registers

During rigorous noise testing or the like, external noise (mainly power supply system noise) can exceed the capacity of the MCU's internal noise control circuitry. In such cases the contents of the port related registers may be changed.

As a firmware countermeasure, it is recommended that the port registers, port direction registers, and pull-up control registers be reset periodically. However, examine the control processing fully before introducing the reset routine as conflicts may be created between the reset routine and interrupt routines.

20. Notes on On-chip Debugger

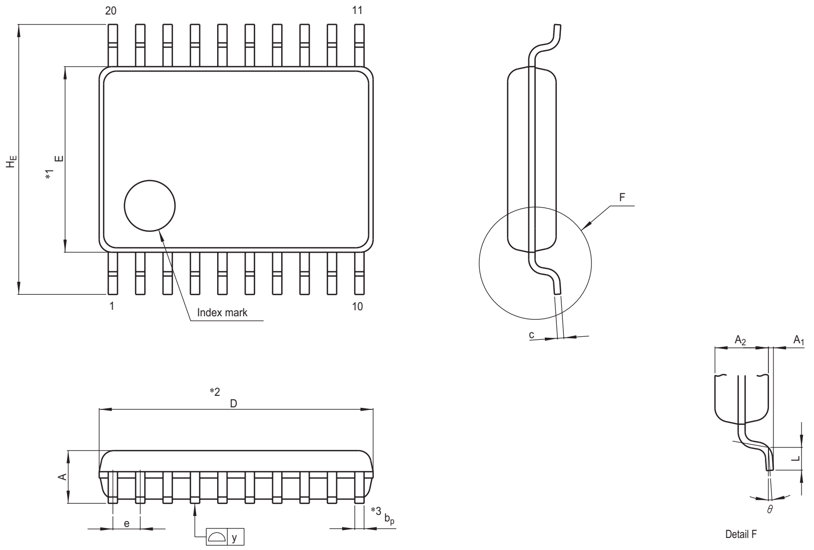
When using on-chip debugger to develop and debug programs for the R8C/18 Group and R8C/19 Group, take note of the following.

(1) Do not access the related UART1 registers.
(2) Do not use from addresses OC000h address to OC7FFh because the on-chip debugger uses these addresses.
(3) Do not set the address match interrupt (registers AIER, RMAD0, and RMAD1 and fixed vector tables) in a user system.
(4) Do not use the BRK instruction in a user system.
(5) A stack pointer of up to 8 bytes is used during user program breaks. Therefore, leave 8 bytes free for the stack area.

Connecting and using the on-chip debugger has some special restrictions. Refer to the on-chip debugger manual for on-chip debugger details.

Appendix 1. Package Dimensions

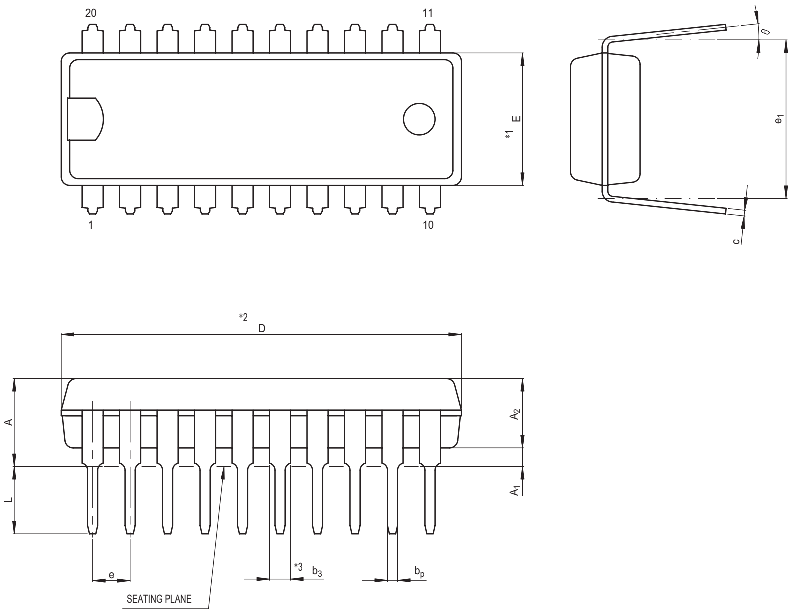
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LSSOP20-4.4x6.5-0.65	PLSP0020JB-A	20P2F-A	0.1g



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	6.4	6.5	6.6
E	4.3	4.4	4.5
A_2	—	1.15	—
A	—	—	1.45
A_1	0	0.1	0.2
b_p	0.17	0.22	0.32
c	0.13	0.15	0.2
θ	0°	—	10°
H_E	6.2	6.4	6.6
e	0.53	0.65	0.77
y	—	—	0.10
L	0.3	0.5	0.7

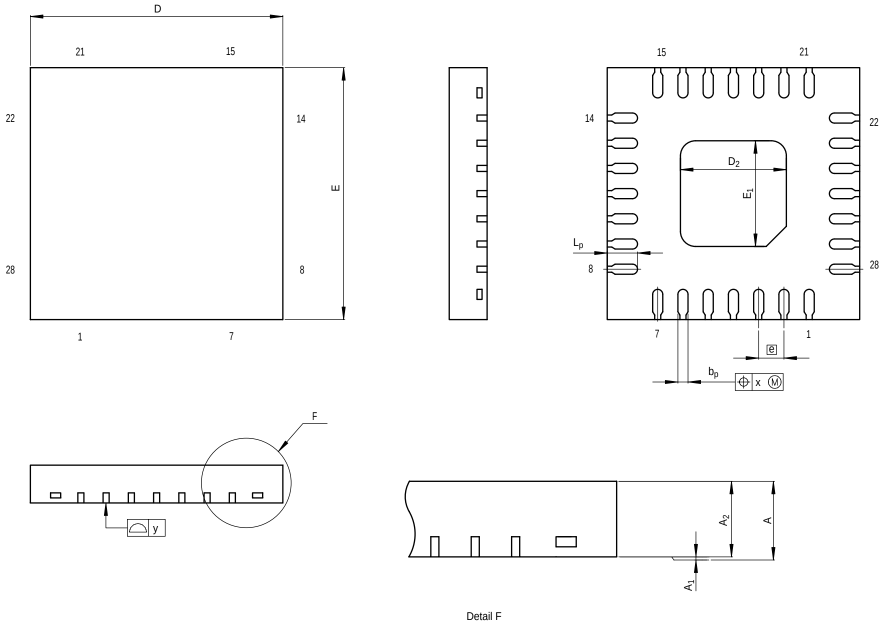
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-SDIP20-6.3x19-1.78	PRDP0020BA-A	20P4B	1.0g



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
e_1	7.32	7.62	7.92
D	18.8	19.0	19.2
E	6.15	6.3	6.45
A	—	—	4.5
A_1	0.51	—	—
A_2	—	3.3	—
b_p	0.38	0.48	0.58
b_3	0.9	1.0	1.3
c	0.22	0.27	0.34
θ	0°	—	15°
e	1.528	1.778	2.028
L	3.0	—	—

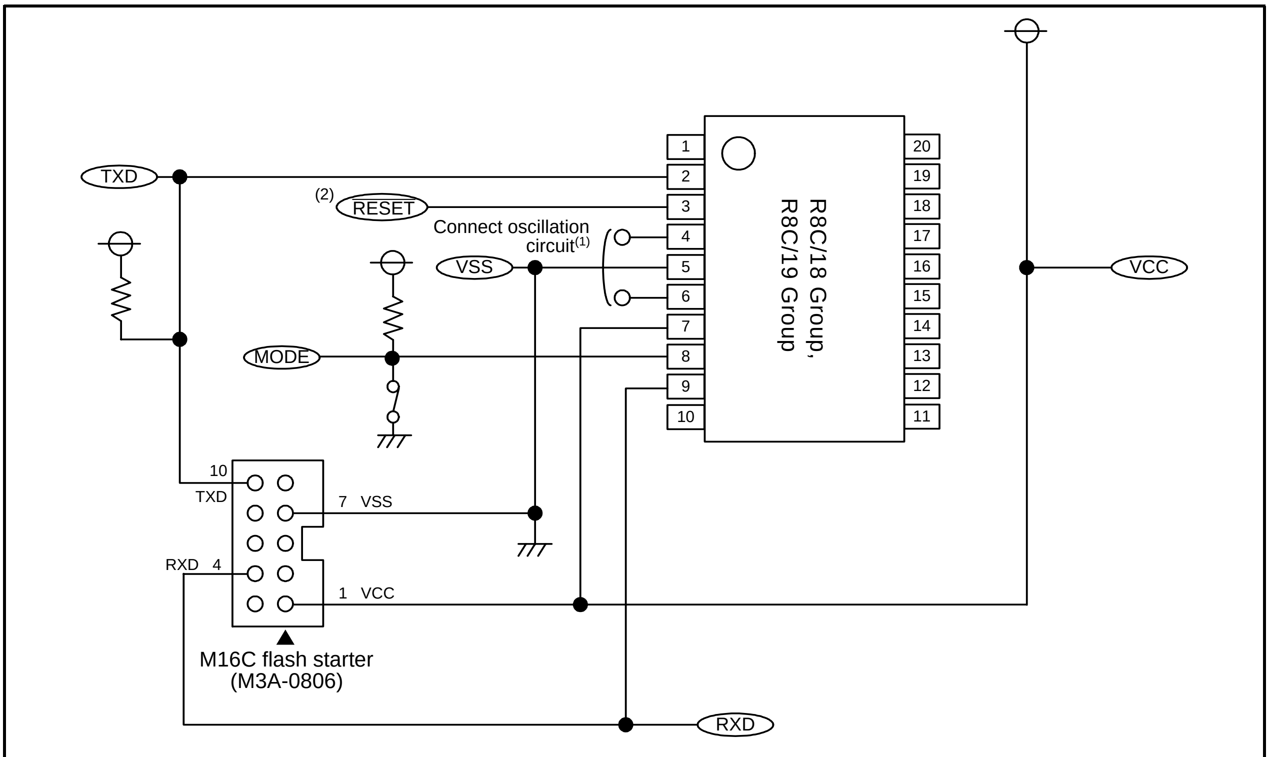
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-HWQFN28-5x5-0.50	PWQN0028KA-B	28PJW-B	0.05g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	4.9	5.0	5.1
E	4.9	5.0	5.1
A_2	—	0.75	—
A	—	—	0.8
A_1	0	0	0.05
b_p	0.15	0.2	0.25
$\boxed{e}$	—	0.5	—
L_p	0.5	0.6	0.7
x	—	—	0.05
y	—	—	0.05
D_2	—	—	2.0
E_1	—	—	2.0

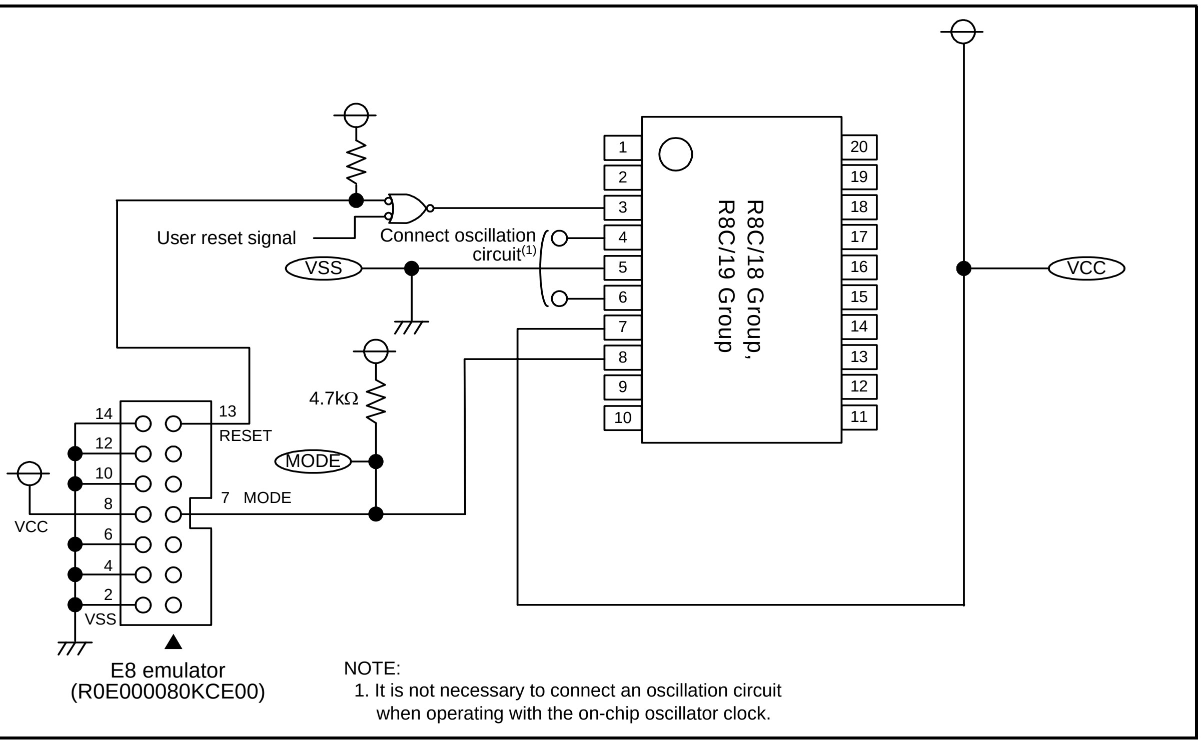
Appendix 2. Connection Examples between Serial Writer and On-Chip Debugging Emulator

Appendix Figure 2.1 shows a Connection Example with M16C Flash Starter (M3A-0806) and Appendix Figure 2.2 shows a Connection Example with E8 Emulator (R0E000080KCE00).



NOTES:
1. An oscillation circuit must be connected, even when operating with the on-chip oscillator clock.
2. Connect an external reset circuit.

Appendix Figure 2.1 Connection Example with M16C Flash Starter (M3A-0806)

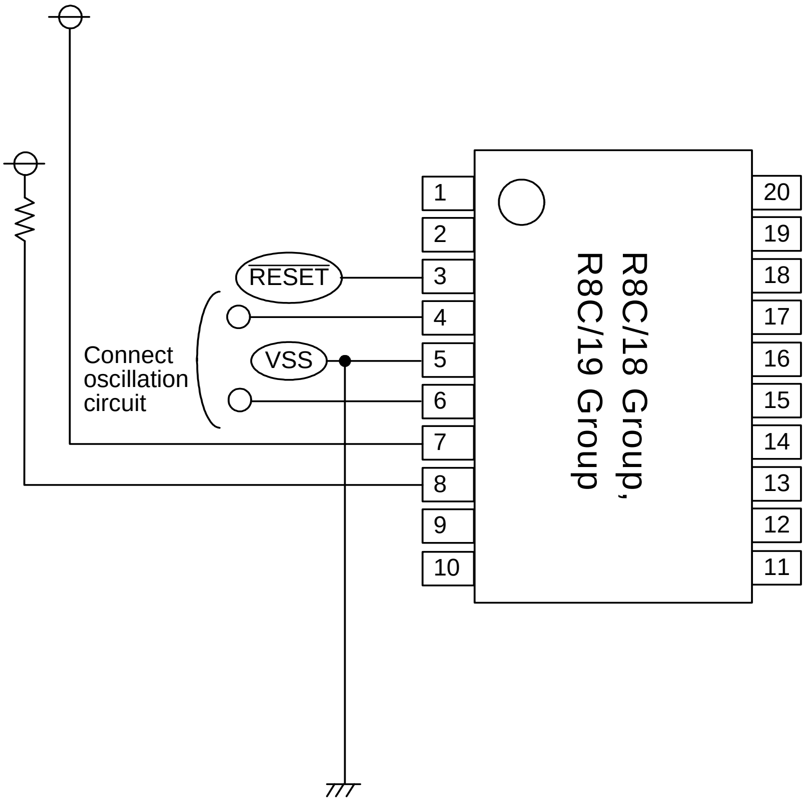


NOTE:
1. It is not necessary to connect an oscillation circuit when operating with the on-chip oscillator clock.

Appendix Figure 2.2 Connection Example with E8 Emulator (R0E000080KCE00)

Appendix 3. Example of Oscillation Evaluation Circuit

Appendix Figure 3.1 shows an Example of Oscillation Evaluation Circuit.



NOTE:
1. Write a program to perform the evaluation.

Appendix Figure 3.1 Example of Oscillation Evaluation Circuit

Register Index

REVISION HISTORY	R8C/18 Group, R8C/19 Group Hardware

Rev.	Date	Description	
		Page	Summary
0.10	Feb 15, 2005	–	First Edition issued
0.21	Apr 04, 2005	4	Figure 1.1 is partly revised.
		5, 6	Table 1.3, Table1.4 are partly revised.
		18	Table 4.3 is partly revised.
		49	Figure 10.1 is partly revised.
		55	“10.1 Main clock” is partly revised.
		61	Table 10.4 is partly revised.
		88	“12.4 Address Match Interrupt” is partly revised.
		93	Table 13.1 is partly revised.
		123, 127	Table 14.9, Table 14.10 are partly revised.
		130	“14.2.5 Precautions on Timer Z” is partly revised.
		133	Figure 14.26 is partly revised.
		137	Table 14.11 is partly revised.
		142	Figure 15.1 is partly revised.
		147	Figure 15.6 is partly revised.
		154	Table 15.6 is partly revised.
		159	Table 16.1 is partly revised.
		160	Figure 16.1 is partly revised.
		161	Figure 16.2 is partly revised.
		164	Figure 16.4 is partly revised.
		166	Figure 16.5 is partly revised.
		199	Table 18.4 is partly revised.
		200	Table 18.5 is partly revised. Figure 18.2 is revised.
		206	Title of Table 18.15 is partly revised. Title of Figure 18.5 is partly revised.
		210	Title of Table 18.22 is partly revised. Title of Figure 18.10 is partly revised.
		216	“19.3.2 Precautions on Timer Z” is partly revised.
		223	“20 Precautions on On-Chip Debugger” is partly revised.
1.00	May 27, 2005	5, 6	Table 1.3, Table1.4 are partly revised.
		9	Table 1.5 is partly revised.
		21	Figure 5.3 revised
		33 to 36	Table 6.4 to Table 6.17 are added.
		42	Figure 7.5 is partly revised.
		43	Figure 7.6 Note 10 added.
		46	Table 7.2 is partly revised.
		47	Table 7.3 is partly revised.
		54	Figure 10.2 is partly deleted.
		56	Figure 10.4 is partly deleted.

Rev.	Date	Description	
		Page	Summary
1.00	May 27, 2005	57	Figure 10.5 is partly deleted.
		61	“10.3.2 CPU Clock” is partly deleted.
		62	Table 10.2 is partly revised.
		63	“10.4.1.1 High-speed Mode” is partly deleted.
			“10.4.1.2 Medium-speed Mode” is partly deleted.
			“10.4.1.3 High-speed, Low-speed On-chip Oscillator Mode” is partly deleted.
		66	Figure 10.8 is revised.
		67	Figure 10.9 is deleted.
		69	“10.6.1 Stop Mode and Wait Mode” is revised.
		102	Figure 14.1 is partly revised.
		105	Table 14.2 is partly revised.
		106	Table 14.3 is partly revised.
		108	Table 14.4 is partly revised.
		109	Table 14.5 is partly revised.
		112	Table 14.6 is partly revised.
		164	Figure 16.2 is partly revised.
		167	Figure 16.4 is partly revised.
		169	Figure 16.5 is partly revised.
		183	Figure 17.7 is partly revised.
		185	Figure 17.9 is partly revised.
		186	Figure 17.11 is partly revised.
		190	Figure 17.14 is partly revised.
		194	“17.5 Standard Serial I/O Mode” is revised.
			Table 17.7 is added.
		195	Table 17.8 is partly revised.
		196	Title of Figure 17.16 is partly revised.
		197	“17.5.1.1 Example of Circuit Application in the Standard Serial I/O Mode” is revised.
			Figure 17.17 is added.
			Title of Figure 17.18 is partly revised.
		200	Former “17.7.1.7” is deleted.
		206	Table 18.9 is revised.
		207	Table 18.10 is partly revised.
		209	Table 18.13 is partly revised.
		213	Table 18.20 is partly revised.
		216	“19.1.1 Stop Mode and Wait Mode” is revised.
			“19.1.3 Oscillation Circuit Constants” is added.
		225	Former “19.6.1.7” is deleted.“
		227	“20. Precautions on On-chip Debugger” is partly added.

Rev.	Date	Description	
		Page	Summary
1.10	Jun 09, 2005	27	Figure 6.1 Note 1 added.
		28	Figure 6.2 Note 1 added.
		30	Figure 6.3 Note 4 added.
		34	Table 6.7 is partly revised (register name).
		36	Table 6.15 is partly revised (UCON → PD3).
		105	Table 14.2 is partly revised (Write to Timer).
		106	Table 14.3 is partly revised (Write to Timer).
		108	Table 14.4 is partly revised (Write to Timer).
		109	Table 14.5 is partly revised (Write to Timer).
		112	Table 14.6 is partly revised (Write to Timer).
		121	Table 14.7 is partly revised (Write to Timer).
		196	Figure 17.16 is partly revised.
		207	Table 18.10 is partly revised.
		229	Appendix Figure 2.1, 2.2 are partly revised.
1.20	Nov 01, 2005	3	Table 1.2 Performance Outline of the R8C/19 Group; Flash Memory: (Data area) → (Data flash) (Program area) → (Program ROM) revised
		4	Figure 1.1 Block Diagram; “Peripheral Function” added, “System Clock Generation” → “System Clock Generator” revised
		6	Table 1.4 Product Information of R8C/19 Group; ROM capacity: “Program area” → “Program ROM”, “Data area” → “Data flash” revised
		9	Table 1.5 Pin Description; Power Supply Input: “VCC/AVCC” → “VCC”, “VSS/AVSS” → “VSS” revised Analog Power Supply Input: added
		11	Figure 2.1 CPU Register; “Reserved Area” → “Reserved Bit” revised
		13	2.8.10 Reserved Area; “Reserved Area” → “Reserved Bit” revised
		15	3.2 R8C/19 Group, Figure 3.2 Memory Map of R8C/19 Group; “Data area” → “Data flash”, “Program area” → “Program ROM” revised
		16	Table 4.1 SFR Information(1); 0009h: “XXXXXX00b” → “00h” 000Ah: “00XXX000b” → “00h” 001Eh: “XXXXX000b” → “00h” revised

REVISION HISTORY	R8C/18 Group, R8C/19 Group Hardware

Rev.	Date	Description	
		Page	Summary
1.20	Nov 01, 2005	18	Table 4.3 SFR Information(3); 0085h: “Prescaler Z” → “Prescaler Z Register” 0086h: “Timer Z Secondary” → “Timer Z Secondary Register” 0087h: “Timer Z Primary” → “Timer Z Primary Register” 008Ch: “Prescaler X” → “Prescaler X Register” 008Dh: “Timer X” → “Timer X Register” 0090h, 0091h: “Timer C” → “Timer C Register” revised
		36	Table 6.16 Port XIN/P4_6, XOUT/P4_7 Setting; Setting value CM1: “1” → “0”, CM0: “0” → “1”, Feedback resistance: “OFF” → “ON”,
		37	Table 6.18 Unassigned Pin Handling, Figure 6.9 Unassigned Pin Handling; “Port P4_2, P4_6, P4_7” → “Port P4_6, P4_7” “VREF” → “Port P4_2/VREF” revised
		51	Table 9.2 Bus Cycles for Access Space of the R8C/1B (19) Group added, Table 9.3 Access Unit and Bus Operation; “SFR” → “SFR, Data flash”, “ROM/RAM” → “ROM (Program ROM), RAM” revised
		52	Table 10.1 Specification of Clock Generation Circuit; Note 2: deleted
		56	Figure 10.4 OCD Register; Note 3: partly deleted
		60	10.2.1 Low-speed On-Chip Oscillator Clock; “The application products ... to accommodate the frequency range.” → “The application products ... for the frequency change.” revised 10.2.2 High-Speed On-Chip Oscillator Clock; “The high-speed on-chip oscillator frequency ... for details.” added
		62	Table 10.2 Setting and Mode of Clock Associated Bit; Medium-speed Mode/devide-by-16: “00b”” → “11b” High-speed,... Mode/devide-by-2: “00b”” → “01b” CM13 added
		67	10.5.1 How to Use Oscillation Stop Detection Function; “This function cannot ... is 2 MHz or below.” → “This function cannot be ... is below 2 MHz.” revised
		68	Figure 10.9 Procedure of Switching Clock Source From Low-Speed On-Chip Oscillator to Main Clock revised
		69	10.6.2 Oscillation Stop Detection Function; “Since the oscillation ...frequency is 2MHz or below, ...” → “Since the oscillation ...frequency is below 2MHz, ...” revised 10.6.4 High-Speed On-Ship Oscillator Clock added.
		70	Figure 11.1 PRCR Register; After Reset: “XXXXX000b”” → “00h” revised
		84	Figure 12.11 INTEN and INT0F Registers; After Reset: “XXXXX000b”” → “00h” revised

REVISION HISTORY	R8C/18 Group, R8C/19 Group Hardware

Rev.	Date	Description	
		Page	Summary
1.20	Nov 01, 2005	92	Figure 12.19 AIER, RMAD0 to RMAD1 Registers; Address Match Interrupt Enable Register and Address Match Interrupt Register i(i=0, 1) revised
		102	Figure 14.1 Block Diagram of Timer X; “Peripheral data bus” → “Data Bus” revised
		115	14.1.6 Precautions on Timer X; “When writing “1” (count starts) to ... writing “1” to the TXS bit.” → ‘ “0” (count stops) can be read ... after the TXS bit is set to “1”.’ revised
		116	Figure 14.11 Block Diagram of Timer Z; “Peripheral Data Bus” → “Data Bus” revised
		133	14.2.5 Precautions on Timer Z; “When writing “1” (count starts) to ... writing “1” to the TZS bit.” → ‘ “0” (count stops) can be read ... after the TZS bit is set to “1”.’ revised
		147	Figure 15.3 U0TB to U1TB, U0RB to U1RB and U0BRG to U1BRG Registers; “UARTi Transmit Buffer Register (i=0 to 1)” and “UARTi Receive Buffer Register (i=0 to 1)” revised
		150	Figure 15.6 U0C1 to U1C1 and UCON Registers; UARTi Transmit / Receive Control Register 1 (i=0 to 1) revised
		157	Table 15.5 Registers to Be Used and Settings in UART Mode; UiBRG: “–” → “0 to 7” revised
		162	Table 16.1 Performance of Comparator Analog Input Voltage: “0V to Vref” → “0V to AVCC” revised
		171	Table 17.1 Flash Memory Version Performance; Program and Erase Endurance: (Program area) → (Program ROM), (Data area) → (Data Flash) revised
		173	17.2 Memory Map; “The user ROM ... area ... Block A and B.” → “The user ROM ... area (program ROM) ... Block A and B (data flash).” revised Figure 17.1 Flash Memory Block Diagram for R8C/18 Group revised
		174	Figure 17.2 Flash Memory Block Diagram for R8C/19 Group revised
		189	17.4.3.5 Block Erase “The block erase command cannot ... program-suspend.” added
		200	Table 17.10 Interrupt in EW1 Mode; During automatic programming (program suspend function enabled) and During automatic programming (program suspend function disabled) revised
		203	Table 18.4 Flash Memory (Program ROM) Electrical Characteristics; NOTES 3 and 5 revised, NOTE8 deleted
		204	Table 18.5 Flash Memory (Data flash Block A, Block B) Electrical Characteristics; NOTES 1 and 3 revised
		206	Table 18.8 Reset Circuit Electrical Characteristics (When Using Voltage Monitor 1 Reset); NOTE 2 revised

Rev.	Date	Description	
		Page	Summary
1.20	Nov 01, 2005	207	Table 18.10 High-speed On-Chip Oscillator Circuit Electrical Characteristics; “High-Speed On-Chip Oscillator ...” → “High-Speed On-Chip Oscillator Frequency ...” revised NOTE 2, 3 added
		209	Table 18.13 Electrical Characteristics (2) [Vcc = 5V]; NOTE 1 deleted
		212	Table 18.20 Electrical Characteristics (4) [Vcc = 3V]; NOTE 1 deleted
		216	19.1.2 Oscillation Stop Detection Function; “Since the oscillation ...frequency is 2MHz or below, ...” → “Since the oscillation ...frequency is below 2MHz, ...” revised
		220	19.3.1 Precautions on Timer X; “When writing “1” (count starts) to ... writing “1” to the TXS bit.” → ‘ “0” (count stops) can be read ... after the TXS bit is set to “1”.’ revised
		221	19.3.2 Precautions on Timer Z; “When writing “1” (count starts) to ... writing “1” to the TZS bit.” → ‘ “0” (count stops) can be read ... after the TZS bit is set to “1”.’ revised
		225	Table 19.2 Interrupt in EW1 Mode; During automatic programming (program suspend function enabled) and During automatic programming (program suspend function disabled) revised
		227	20.Precautions on On-Chip Debugger; (1) added
1.30	Apr 14, 2006	–	Products of PWQN0028KA-B package included
		1	“or SDIP” → “SDIP or a 28-pin plastic molded-HWQFN”
		2, 3	Table 1.1, Table 1.2; Interrupts: Internal 8 → 10 sources, Package: “28-pin molded-plastic HWQFN” added
		5, 6	Table 1.3, Table 1.4; Type No. added, deleted
		9	Figure 1.6 added
		12	Table 1.7 added
		16, 17	Figure 3.1, Figure 3.2; Part Number added, deleted
		26	5.2 “When a capacitor is connected to ... pin 0.8VCC or more.” added
		35 to 38	Table 6.4, Table 6.5, Table 6.6, Table 6.12, Table 6.13, Table 6.14; revised
		54	Figure 10.1 revised
		97	Figure 13.1 revised
		98	Figure 13.2; Option Function Select Register: NOTE 1 revised, NOTE 2 added Watchdog Timer Control Register: NOTE 1 deleted
		107	Table 14.3; NOTE 1 added
		136	Figure 14.25 revised
		143	Table 14.12; NOTE 1 revised

Rev.	Date	Description	
		Page	Summary
1.30	Apr 14, 2006	148	Figure 15.3; NOTE 3 added
		150	Figure 15.5; NOTE 1 added
		164	Table 16.1 revised
		177	17.3.2; “The ROM code protect function is disabled .... on-chip flash memory.” deleted “To disable ROM code protect ....” revised Figure 17.4; NOTE 1 revised, NOTE 2 added
		182	Figure 17.5; NOTE 6 added
		192	Table 17.5; Value after Reset revised
		194	Figure 17.15 revised
		204, 205	Table 18.4, Table 18.5; “Ta” → “Ambient temperature”, Conditions: V_{CC} = 5.0 V at Topr = 25 °C deleted
		210, 214	Table 18.13, Table 18.20; The title revised, Condition of Stop Mode “Topr = 25 °C” added
		212, 216	Table 18.17, Table 18.24; Standard of td(C-Q) and tsu(D-C) revised
		229, 230	Package Dimensions revised, added
		231	Appendix Figure 2.1 revised
		232	Appendix Figure 3.1 revised

R8C/18 Group, R8C/19 Group Hardware Manual

Publication Data : Rev.0.10 Feb 15, 2005
Rev.1.30 Apr 14, 2006

Published by : Sales Strategic Planning Div.
Renesas Technology Corp.

R8C/18 Group, R8C/19 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan