

# R32C/111 Group

User's Manual: Hardware

RENESAS MCU

M16C Family / R32C/100 Series

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

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.  
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.  
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.  

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

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

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

## 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 document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

### 1. Handling of Unused Pins

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

- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of 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 a product with a different part number, confirm that the change will not lead to problems.

- The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

# About This Manual

## 1. Purpose and Target User

This manual is designed to be read primarily by application developers who have an understanding of this microcomputer (MCU) including its hardware functions and electrical characteristics. The user should have a basic understanding of electric circuits, logic circuits and, MCUs.

This manual consists of 27 chapters covering six main categories: Overview, CPU, System Control, Peripherals, Electrical Characteristics, and Usage Notes.

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

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

The R32C/111 Group includes the documents listed below. Verify this manual is the latest version by visiting the Renesas Electronics website.

Type of Document	Contents	Document Name	Document Number
Datasheet	Overview of Hardware and Electrical Characteristics	R32C/111 Group Datasheet	R01DS0062EJ0130
User's Manual: Hardware	Specifications and detailed descriptions of: -pin layout -memory map -peripherals -electrical characteristics -timing characteristics Refer to the Application Manual for peripheral usage.	R32C/111 Group User's Manual: Hardware	This publication
User's Manual: Software/Software Manual	Descriptions of instruction set	R32C/100 Series Software Manual	REJ09B0267-0100
Application Note	-Usages -Applications -Sample programs -Programming technics using Assembly language or C programming language	Available on the Renesas Electronics website.	
Renesas Technical Update	Bulletins on product specifications, documents, etc.		

## 2. Numbers and Symbols

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

(1) Registers, bits, and pins

Registers, bits, and pins are indicated by symbols. Each symbol has a register/bit/pin identifier after the symbol.

Example: PM03 bit in the PM0 register

P3\_5 pin, VCC pin

(2) Numbers

A binary number has the suffix “b” except for a 1-bit value.

A hexadecimal number has the suffix “h”.

A decimal number has no suffix.

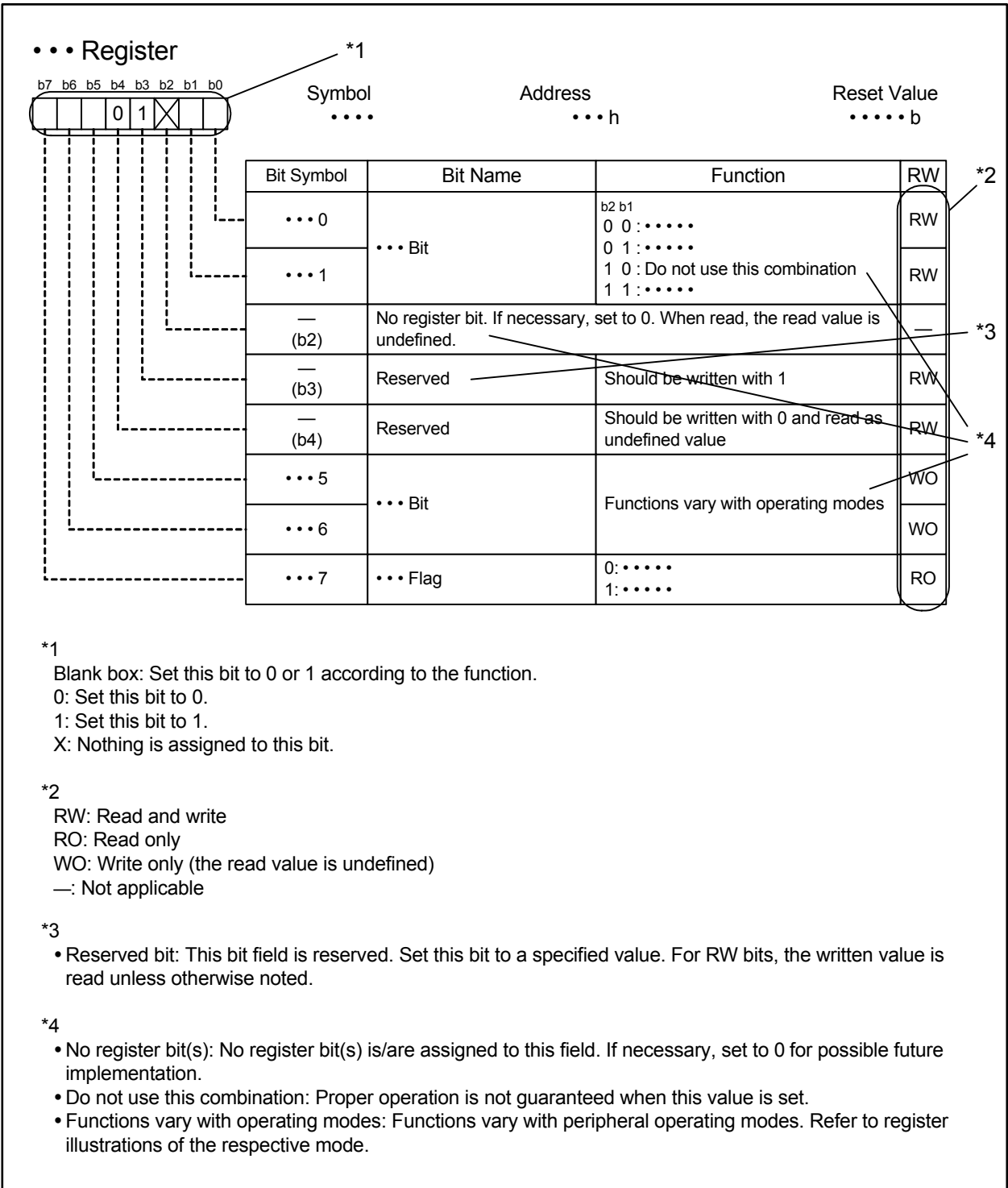
Example: Binary notation: 11b

Hexadecimal notation: EFA0h

Decimal notation: 1234

### 3. Registers

The following illustration describes registers used throughout this manual.



## 4. Abbreviations and Acronyms

The following acronyms and terms are used throughout this manual.

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

# TABLE OF CONTENTS

1.	Overview	1
1.1	Features.....	1
1.1.1	Applications .....	1
1.1.2	Performance Overview .....	2
1.2	Product Information .....	6
1.3	Block Diagram .....	8
1.4	Pin Assignments .....	10
1.5	Pin Definitions and Functions .....	18
2.	Central Processing Unit (CPU)	25
2.1	General Purpose Registers .....	26
2.1.1	Data Registers (R2R0, R3R1, R6R4, and R7R5).....	26
2.1.2	Address Registers (A0, A1, A2, and A3) .....	26
2.1.3	Static Base Register (SB).....	26
2.1.4	Frame Base Register (FB).....	26
2.1.5	Program Counter (PC).....	26
2.1.6	Interrupt Vector Table Base Register (INTB) .....	26
2.1.7	User Stack Pointer (USP) and Interrupt Stack Pointer (ISP) .....	26
2.1.8	Flag Register (FLG).....	26
2.2	Fast Interrupt Registers .....	28
2.2.1	Save Flag Register (SVF).....	28
2.2.2	Save PC Register (SVP) .....	28
2.2.3	Vector Register (VCT) .....	28
2.3	DMAC-associated Registers.....	28
2.3.1	DMA Mode Registers (DMD0, DMD1, DMD2, and DMD3) .....	28
2.3.2	DMA Terminal Count Registers (DCT0, DCT1, DCT2, and DCT3) .....	28
2.3.3	DMA Terminal Count Reload Registers (DCR0, DCR1, DCR2, and DCR3) .....	28
2.3.4	DMA Source Address Registers (DSA0, DSA1, DSA2, and DSA3).....	28
2.3.5	DMA Source Address Reload Registers (DSR0, DSR1, DSR2, and DSR3).....	28
2.3.6	DMA Destination Address Registers (DDA0, DDA1, DDA2, and DDA3) .....	28
2.3.7	DMA Destination Address Reload Registers (DDR0, DDR1, DDR2, and DDR3) .....	28
3.	Memory	29
4.	Special Function Registers (SFRs)	30
5.	Resets	54
5.1	Hardware Reset.....	54
5.2	Software Reset .....	57
5.3	Watchdog Timer Reset .....	57
5.4	Reset Vector .....	58



<b>6.</b>	<b>Power Management</b>	<b>59</b>
6.1	Voltage Regulators for Internal Logic.....	59
6.1.1	Decoupling Capacitor .....	60
6.2	Low Voltage Detector.....	61
6.2.1	Operational State of Low Voltage Detector.....	64
6.2.2	Low Voltage Detection Interrupt .....	64
6.2.3	Application Example of the Low Voltage Detector.....	65
<b>7.</b>	<b>Processor Mode</b>	<b>66</b>
7.1	Types of Processor Modes .....	66
7.2	Processor Mode Setting .....	66
<b>8.</b>	<b>Clock Generator</b>	<b>69</b>
8.1	Clock Generator Types .....	69
8.1.1	Main Clock.....	78
8.1.2	Sub Clock (fC).....	79
8.1.3	PLL Clock .....	80
8.1.4	On-chip Oscillator Clock .....	83
8.2	Oscillator Stop Detection .....	84
8.2.1	How to Use Oscillator Stop Detection.....	84
8.3	Base Clock.....	84
8.4	CPU Clock and Peripheral Bus Clock.....	85
8.5	Peripheral Clock .....	85
8.6	Clock Output Function .....	86
8.7	Power Control .....	87
8.7.1	Normal Operating Mode .....	88
8.7.2	Wait Mode.....	93
8.7.3	Stop Mode .....	96
8.8	System Clock Protection.....	98
8.9	Notes on Clock Generator .....	99
8.9.1	Sub Clock .....	99
8.9.2	Power Control.....	99
<b>9.</b>	<b>Bus</b>	<b>100</b>
9.1	Bus Settings.....	100
9.2	Peripheral Bus Timing Setting .....	101
9.3	External Bus Setting .....	102
9.3.1	External Address Space Setting .....	102
9.3.2	External Data Bus Width Setting .....	108
9.3.3	Separate Bus/Multiplexed Bus Selection.....	110
9.3.4	Read and Write Signals.....	113
9.3.5	External Bus Timing.....	114

9.3.6	ALE Signal .....	118
9.3.7	$\overline{\text{RDY}}$ Signal .....	119
9.3.8	$\overline{\text{HOLD}}$ Signal .....	122
9.3.9	BCLK Output .....	122
9.4	External Bus State when Accessing Internal Space .....	122
9.5	Notes on Bus .....	123
9.5.1	Notes on Designing a System .....	123
9.5.2	Notes on Register Settings .....	123
<b>10.</b>	<b>Protection</b> .....	<b>124</b>
10.1	Protect Register (PRCR Register) .....	124
10.2	Protect Register 2 (PRCR2 Register) .....	125
10.3	Protect Register 3 (PRCR3 Register) .....	125
10.4	Protect Release Register (PRR Register) .....	126
<b>11.</b>	<b>Interrupts</b> .....	<b>127</b>
11.1	Interrupt Types .....	127
11.2	Software Interrupts .....	128
11.3	Hardware Interrupts .....	129
11.3.1	Special Interrupts .....	129
11.3.2	Peripheral Interrupts .....	129
11.4	Fast Interrupt .....	130
11.5	Interrupt Vectors .....	130
11.5.1	Fixed Vector Table .....	131
11.5.2	Relocatable Vector Table .....	131
11.6	Interrupt Request Acceptance .....	136
11.6.1	I Flag and IPL .....	136
11.6.2	Interrupt Control Registers .....	137
11.6.3	Wake-up IPL Setting Register .....	140
11.6.4	Interrupt Sequence .....	141
11.6.5	Interrupt Response Time .....	142
11.6.6	IPL after Accepting an Interrupt Request .....	143
11.6.7	Register Saving .....	143
11.7	Register Restoring from Interrupt Handler .....	144
11.8	Interrupt Priority .....	144
11.9	Priority Resolver .....	144
11.10	External Interrupt .....	146
11.11	NMI .....	147
11.12	Key Input Interrupt .....	148
11.13	Intelligent I/O Interrupt .....	149
11.14	Notes on Interrupts .....	152
11.14.1	ISP Setting .....	152

11.14.2	NMI .....	152
11.14.3	External Interrupts .....	152
<b>12.</b>	<b>Watchdog Timer</b>	<b>153</b>
<b>13.</b>	<b>DMAC</b>	<b>155</b>
13.1	Transfer Cycle.....	164
13.1.1	Effect of Transfer Address and Data Bus Width .....	164
13.1.2	Effect of Bus Timing.....	165
13.1.3	Effect of $\overline{\text{RDY}}$ Signal .....	165
13.2	DMA Transfer Cycle.....	167
13.3	Channel Priority and DMA Transfer Timing .....	168
13.4	Notes on DMAC.....	169
13.4.1	DMAC-associated Register Settings .....	169
13.4.2	Reading DMAC-associated Registers .....	169
<b>14.</b>	<b>DMAC II</b>	<b>170</b>
14.1	DMAC II Settings .....	170
14.1.1	Registers RIPL1 and RIPL2 .....	171
14.1.2	DMAC II Index .....	172
14.1.3	Interrupt Control Register of the Peripherals .....	175
14.1.4	Relocatable Vector Table of the Peripherals.....	175
14.1.5	IRLT Bit in the IIOiE Register (i = 0 to 11).....	175
14.2	DMAC II Operation .....	175
14.3	Transfer Types .....	175
14.3.1	Memory-to-memory Transfer .....	175
14.3.2	Immediate Data Transfer .....	176
14.3.3	Calculation Result Transfer .....	176
14.4	Transfer Modes.....	176
14.4.1	Single Transfer .....	176
14.4.2	Burst Transfer .....	176
14.4.3	Multiple Transfer.....	176
14.5	Chain Transfer .....	177
14.6	DMA II Transfer Complete Interrupt.....	177
14.7	Execution Time .....	178
<b>15.</b>	<b>Programmable I/O Ports</b>	<b>179</b>
15.1	Port Pi Register (Pi register, i = 0 to 10) .....	181
<b>16.</b>	<b>Timers</b>	<b>182</b>
16.1	Timer A .....	184
16.1.1	Timer Mode.....	191
16.1.2	Event Counter Mode.....	193

16.1.3	One-shot Timer Mode.....	197
16.1.4	Pulse-width Modulation Mode.....	199
16.2	Timer B .....	202
16.2.1	Timer Mode.....	205
16.2.2	Event Counter Mode.....	207
16.2.3	Pulse Period/Pulse-width Measure Mode.....	209
16.3	Notes on Timers.....	212
16.3.1	Timer A and Timer B.....	212
16.3.2	Timer A .....	212
16.3.3	Timer B .....	214
<b>17.</b>	<b>Three-phase Motor Control Timers</b> .....	<b>215</b>
17.1	Modulation Modes of Three-phase Motor Control Timers .....	222
17.2	Timer B2 .....	223
17.3	Timers A4, A1, and A2.....	225
17.4	Simultaneous Conduction Prevention and Dead Time Timer .....	228
17.5	Three-phase Motor Control Timer Operation.....	229
17.6	Notes on Three-phase Motor Control Timers .....	232
17.6.1	Shutdown.....	232
17.6.2	Register Setting .....	232
17.6.3	Restriction for the Use of 64-pin Package's Product .....	232
<b>18.</b>	<b>Serial Interface</b> .....	<b>233</b>
18.1	Synchronous Serial Interface Mode.....	250
18.1.1	Reset Procedure on Transmit/Receive Error.....	255
18.1.2	CLK Polarity.....	255
18.1.3	LSB First and MSB First Selection .....	256
18.1.4	Continuous Receive Mode .....	256
18.1.5	Serial Data Logic Inversion.....	257
18.1.6	CTS/RTS Function.....	257
18.2	Asynchronous Serial Interface Mode (UART Mode).....	258
18.2.1	Bit Rate .....	263
18.2.2	Reset Procedure on Transmit/Receive Error.....	264
18.2.3	LSB First and MSB First Selection .....	264
18.2.4	Serial Data Logic Inversion.....	265
18.2.5	TXD and RXD I/O Polarity Inversion .....	266
18.2.6	CTS/RTS Function.....	266
18.3	Special Mode 1 (I <sup>2</sup> C Mode).....	267
18.3.1	START Condition and STOP Condition Detection .....	273
18.3.2	START Condition and STOP Condition Generation .....	273
18.3.3	Arbitration .....	274
18.3.4	SCL Control and Clock Synchronization .....	275

18.3.5	SDA Output .....	277
18.3.6	SDA Input .....	277
18.3.7	Acknowledge .....	277
18.3.8	Transmit/Receive Operation Reset.....	277
18.4	Special Mode 2 .....	278
18.4.1	$\overline{\text{SSi}}$ Input Pin Function (i = 0 to 6).....	280
18.4.2	Clock Phase Setting .....	281
18.5	Notes on Serial Interface .....	283
18.5.1	Changing the UiBRG Register (i = 0 to 8) .....	283
18.5.2	Restriction for the Use of 64-pin Package's Products .....	283
18.5.3	Synchronous Serial Interface Mode .....	283
18.5.4	Special Mode 1 (I <sup>2</sup> C Mode) .....	283
18.5.5	Reset Procedure or Suspend/Resume Procedure .....	284
<b>19.</b>	<b>A/D Converter</b> .....	<b>285</b>
19.1	Mode Descriptions .....	293
19.1.1	One-shot Mode .....	293
19.1.2	Repeat Mode .....	294
19.1.3	Single Sweep Mode.....	295
19.1.4	Repeat Sweep Mode 0 .....	296
19.1.5	Repeat Sweep Mode 1 .....	297
19.1.6	Multi-port Single Sweep Mode.....	298
19.1.7	Multi-port Repeat Sweep Mode 0 .....	299
19.2	Functions .....	300
19.2.1	Resolution Selection .....	300
19.2.2	Sample and Hold Function .....	300
19.2.3	Trigger Selection.....	300
19.2.4	DMAC Operating Mode .....	300
19.2.5	Function-extended Analog Input Pins.....	301
19.2.6	External Operating Amplifier (Op-Amp) Connection Mode.....	301
19.2.7	Power Saving .....	302
19.2.8	Output Impedance of Sensor Equivalent Circuit under A/D Conversion .....	302
19.3	Notes on A/D Converter.....	304
19.3.1	Notes on Designing Boards.....	304
19.3.2	Notes on Programming.....	305
<b>20.</b>	<b>D/A Converter</b> .....	<b>306</b>
<b>21.</b>	<b>CRC Calculator</b> .....	<b>308</b>
<b>22.</b>	<b>X-Y Conversion</b> .....	<b>311</b>
22.1	Data Conversion When Reading .....	312
22.2	Data Conversion When Writing.....	314

<b>23. Intelligent I/O</b>	<b>315</b>
23.1 Base Timer for Groups 0 to 2.....	330
23.2 Time Measurement for Groups 0 and 1.....	336
23.3 Waveform Generation for Groups 0 to 2.....	340
23.3.1 Single-phase Waveform Output Mode for Groups 0 to 2.....	341
23.3.2 Inverted Waveform Output Mode for Groups 0 to 2.....	343
23.3.3 Set/Reset Waveform Output Mode (SR Waveform Output Mode) for Groups 0 to 2.....	345
23.3.4 Bit Modulation PWM Output Mode for Group 2.....	348
23.3.5 Real-time Port Output Mode (RTP Output Mode) for Group 2.....	350
23.3.6 Parallel Real-time Port Output Mode (RTP Output Mode) for Group 2.....	352
23.4 Group 2 Serial Interface.....	354
23.4.1 Variable Synchronous Serial Interface Mode for Group 2.....	359
<b>24. I/O Pins</b>	<b>362</b>
24.1 Port Pi Direction Register (PDi Register, i = 0 to 10).....	363
24.2 Output Function Select Registers.....	364
24.3 Input Function Select Registers.....	378
24.4 Pull-up Control Registers 0 to 3 (Registers PUR0 to PUR3).....	382
24.5 Port Control Register (PCR Register).....	385
24.6 Configuring Unused Pins.....	386
<b>25. Flash Memory</b>	<b>389</b>
25.1 Overview.....	389
25.2 Flash Memory Protection.....	391
25.2.1 Lock Bit Protection.....	391
25.2.2 ROM Code Protection.....	391
25.2.3 ID Code Protection.....	392
25.2.4 Forcible Erase Function.....	393
25.2.5 Standard Serial I/O Mode Disable Function.....	393
25.3 CPU Rewrite Mode.....	394
25.3.1 CPU Operating Mode and Flash Memory Rewrite.....	401
25.3.2 Flash Memory Rewrite Bus Timing.....	402
25.3.3 Software Commands.....	406
25.3.4 Mode Transition.....	407
25.3.5 Issuing Software Commands.....	408
25.3.6 Status Check.....	414
25.4 Standard Serial I/O Mode.....	415
25.5 Parallel I/O mode.....	421
25.6 Notes on Flash Memory Rewriting.....	422
25.6.1 Note on Power Supply.....	422
25.6.2 Note on Hardware Reset.....	422
25.6.3 Note on Flash Memory Protection.....	422

25.6.4	Notes on Programming.....	422
25.6.5	Notes on Interrupts.....	422
25.6.6	Notes on Rewrite Control Program.....	423
25.6.7	Notes on Number of Program/Erase Cycles and Software Command Execution Time .....	423
25.6.8	Other Notes .....	423
<b>26.</b>	<b>Electrical Characteristics</b>	<b>424</b>
<b>27.</b>	<b>Usage Notes</b>	<b>462</b>
27.1	Notes on Board Designing.....	462
27.1.1	Power Supply Pins .....	462
27.1.2	Supply Voltage.....	462
27.2	Notes on Register Setting.....	463
27.2.1	Registers with Write-only Bits .....	463
27.3	Notes on Clock Generator .....	465
27.3.1	Sub Clock .....	465
27.3.2	Power Control.....	465
27.4	Notes on Bus .....	466
27.4.1	Notes on Designing a System .....	466
27.4.2	Notes on Register Settings.....	466
27.5	Notes on Interrupts .....	467
27.5.1	ISP Setting.....	467
27.5.2	NMI .....	467
27.5.3	External Interrupts .....	467
27.6	Notes on DMAC.....	468
27.6.1	DMAC-associated Register Settings .....	468
27.6.2	Reading DMAC-associated Registers .....	468
27.7	Notes on Timers.....	469
27.7.1	Timer A and Timer B.....	469
27.7.2	Timer A .....	469
27.7.3	Timer B .....	471
27.8	Notes on Three-phase Motor Control Timers .....	472
27.8.1	Shutdown.....	472
27.8.2	Register Setting .....	472
27.8.3	Restriction for the Use of 64-pin Package's Product .....	472
27.9	Notes on Serial Interface .....	473
27.9.1	Changing the UiBRG Register (i = 0 to 8) .....	473
27.9.2	Restriction for the Use of 64-pin Package's Products .....	473
27.9.3	Synchronous Serial Interface Mode .....	473
27.9.4	Special Mode 1 (I <sup>2</sup> C Mode) .....	473
27.9.5	Reset Procedure or Suspend/Resume Procedure .....	474
27.10	Notes on A/D Converter.....	475

27.10.1	Notes on Designing Boards.....	475
27.10.2	Notes on Programming.....	476
27.11	Notes on Flash Memory Rewriting.....	477
27.11.1	Note on Power Supply.....	477
27.11.2	Note on Hardware Reset.....	477
27.11.3	Note on Flash Memory Protection.....	477
27.11.4	Notes on Programming.....	477
27.11.5	Notes on Interrupts.....	477
27.11.6	Notes on Rewrite Control Program.....	478
27.11.7	Notes on Number of Program/Erase Cycles and Software Command Execution Time.....	478
27.11.8	Other Notes.....	478
Appendix 1. Package Dimensions		479
INDEX		481



## **1. Overview**

### **1.1 Features**

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

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

The R32C/111 Group is the standard MCU within the R32C/100 Series. This product, provided as 100-pin plastic molded LGA, and 100-/64-pin plastic molded LQFP packages, has a maximum of nine channels of serial interface.

#### **1.1.1 Applications**

Audio, cameras, television, home appliance, printer, meter, office/industrial equipment, communication/portable devices

### 1.1.2 Performance Overview

Tables 1.1 to 1.4 show the performance overview of the R32C/111 Group.

**Table 1.1 Performance Overview for the 100-pin Package (1/2)**

Unit	Function	Explanation
CPU	Central processing unit	R32C/100 Series CPU Core <ul style="list-style-type: none"> <li>• Basic instructions: 108</li> <li>• Minimum instruction execution time: 20 ns (<math>f(\text{CPU}) = 50 \text{ MHz}</math>)</li> <li>• Multiplier: 32-bit <math>\times</math> 32-bit <math>\rightarrow</math> 64-bit</li> <li>• Multiply-accumulate unit: 32-bit <math>\times</math> 32-bit + 64-bit <math>\rightarrow</math> 64-bit</li> <li>• IEEE-754 compatible FPU: Single precision</li> <li>• 32-bit barrel shifter</li> <li>• Operating mode: Single-chip mode, memory expansion mode, microprocessor mode (optional <sup>(1)</sup>)</li> </ul>
Memory		Flash memory: 256 to 512 Kbytes RAM: 32 to 63 Kbytes Data flash: 4 Kbytes $\times$ 2 blocks Refer to Table 1.5 for memory size of each product group
Voltage Detector	Low voltage detector	Optional <sup>(1)</sup> Low voltage detection interrupt
Clock	Clock generator	<ul style="list-style-type: none"> <li>• 4 circuits (main clock, sub clock, PLL, on-chip oscillator)</li> <li>• Oscillation stop detector: Main clock oscillator stop/restart detection</li> <li>• Frequency divide circuit: Divide-by-2 to divide-by-24 selectable</li> <li>• Low power modes: Wait mode, stop mode</li> </ul>
External Bus Expansion	Bus and memory expansion	<ul style="list-style-type: none"> <li>• Address space: 4 Gbytes (of which up to 64 Mbytes is user accessible)</li> <li>• External bus Interface: Support for wait-state insertion, 4 chip select outputs, 3V/5V interface</li> <li>• Bus format: Separate bus/Multiplexed bus selectable, data bus width selectable (8/16 bits)</li> </ul>
Interrupts		Interrupt vectors: 261 External interrupt inputs: $\overline{\text{NMI}}$ , $\overline{\text{INT}} \times 6$ , key input $\times 4$ Interrupt priority levels: 7
Watchdog Timer		15 bits $\times$ 1 (selectable input frequency from prescaler output)
DMA	DMAC	4 channels <ul style="list-style-type: none"> <li>• Cycle-steal transfer mode</li> <li>• Request sources: 51</li> <li>• 2 transfer modes: Single transfer, repeat transfer</li> </ul>
	DMAC II	<ul style="list-style-type: none"> <li>• Triggered by an interrupt request of any peripheral</li> <li>• 3 characteristic transfer functions: Immediate data transfer, calculation result transfer, chain transfer</li> </ul>
I/O Ports	Programmable I/O ports	<ul style="list-style-type: none"> <li>• 2 input-only ports</li> <li>• 82 CMOS I/O ports</li> <li>• 2 N-channel open drain ports</li> <li>• A pull-up resistor is selectable for every 4 input ports</li> </ul>

Note:

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

**Table 1.2 Performance Overview for the 100-pin Package (2/2)**

Unit	Function	Explanation
Timer	Timer A	16-bit timer × 5 Timer mode, event counter mode, one-shot timer mode, pulse-width modulation (PWM) mode Two-phase pulse signal processing in event counter mode (two-phase encoder input) × 3
	Timer B	16-bit timer × 6 Timer mode, event counter mode, pulse frequency measurement mode, pulse-width measurement mode
	Three-phase motor control timer	Three-phase motor control timer × 1 (timers A1, A2, A4, and B2 used) 8-bit programmable dead time timer
Serial Interface	UART0 to UART8	Asynchronous/synchronous serial interface × 9 channels <ul style="list-style-type: none"> <li>• I<sup>2</sup>C-bus (UART0 to UART6)</li> <li>• Special mode 2 (UART0 to UART6)</li> <li>• IEBus (optional <sup>(1)</sup>) (UART0 to UART6)</li> </ul>
A/D Converter		10-bit resolution × 26 channels Sample and hold functionality integrated
D/A Converter		8-bit resolution × 2
CRC Calculator		CRC-CCITT ( $X^{16} + X^{12} + X^5 + 1$ )
X-Y Converter		16 bits × 16 bits
Intelligent I/O		Time measurement (input capture): 16 bits × 16 Waveform generation (output compare): 16 bits × 19 Serial interface: Variable-length synchronous serial I/O mode, IEBus mode (optional <sup>(1)</sup> )
Flash Memory		Programming and erasure supply voltage: VCC1 = VCC2 = 3.0 to 5.5 V Minimum endurance: 1,000 program/erase cycles Security protection: ROM code protect, ID code protect Debugging: On-chip debug, on-board flash programming
Operating Frequency/Supply Voltage		50 MHz/VCC1 = 3.0 to 5.5 V, VCC2 = 3.0 V to VCC1
Operating Temperature		-20°C to 85°C (N version) -40°C to 85°C (D version)
Current Consumption		32 mA (VCC1 = VCC2 = 5.0 V, f(CPU) = 50 MHz) 8 μA (VCC1 = VCC2 = 3.3 V, f(XCIN) = 32.768 kHz, in wait mode)
Package		100-pin plastic molded LQFP (PLQP0100KB-A) 100-pin plastic molded TFLGA (PTLG0100KA-A)

Note:

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

**Table 1.3 Performance Overview for the 64-pin Package (1/2)**

Unit	Function	Explanation
CPU	Central processing unit	R32C/100 Series CPU Core <ul style="list-style-type: none"> <li>• Basic instructions: 108</li> <li>• Minimum instruction execution time: 20 ns (<math>f(\text{CPU}) = 50 \text{ MHz}</math>)</li> <li>• Multiplier: 32-bit <math>\times</math> 32-bit <math>\rightarrow</math> 64-bit</li> <li>• Multiply-accumulate unit: 32-bit <math>\times</math> 32-bit + 64-bit <math>\rightarrow</math> 64-bit</li> <li>• IEEE-754 compatible FPU: Single precision</li> <li>• 32-bit barrel shifter</li> <li>• Operating mode: Single-chip mode</li> </ul>
Memory		Flash memory: 128/256 Kbytes RAM: 32 Kbytes Data flash: 4 Kbytes $\times$ 2 blocks Refer to Table 1.5 for memory size of each product group
Voltage Detector	Low voltage detector	Optional <sup>(1)</sup> Low voltage detection interrupt
Clock	Clock generator	<ul style="list-style-type: none"> <li>• 4 circuits (main clock, sub clock, PLL, on-chip oscillator)</li> <li>• Oscillation stop detector: Main clock oscillator stop/restart detection</li> <li>• Frequency divide circuit: Divide-by-2 to divide-by-24 selectable</li> <li>• Low power modes: Wait mode, stop mode</li> </ul>
Interrupts		Interrupt vectors: 261 External interrupt inputs: $\overline{\text{NMI}}$ , $\overline{\text{INT}} \times 6$ , key input $\times 4$ Interrupt priority levels: 7
Watchdog Timer		15 bits $\times$ 1 (selectable input frequency from prescaler output)
DMA	DMAC	4 channels <ul style="list-style-type: none"> <li>• Cycle-steal transfer mode</li> <li>• Request sources: 45</li> <li>• 2 transfer modes: Single transfer, repeat transfer</li> </ul>
	DMAC II	<ul style="list-style-type: none"> <li>• Triggered by an interrupt request of any peripheral</li> <li>• 3 characteristic transfer functions: Immediate data transfer, calculation result transfer, chain transfer</li> </ul>
I/O Ports	Programmable I/O ports	<ul style="list-style-type: none"> <li>• 1 input-only port</li> <li>• 49 CMOS I/O ports</li> <li>• 2 N-channel open drain ports</li> <li>• A pull-up resistor is selectable for every 4 input ports</li> </ul>

Note:

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

**Table 1.4 Performance Overview for the 64-pin Package (2/2)**

Unit	Function	Explanation
Timer	Timer A	16-bit timer × 5 Timer mode, event counter mode, one-shot timer mode, pulse-width modulation (PWM) mode Two-phase pulse signal processing in event counter mode (two-phase encoder input) × 3
	Timer B	16-bit timer × 6 <sup>(1)</sup> Timer mode, event counter mode, pulse frequency measurement mode, pulse-width measurement mode
	Three-phase motor control timer	Three-phase motor control timer × 1 (timers A1, A2, A4, and B2 used) 8-bit programmable dead time timer
Serial Interface	UART0 to UART3, UART5, UART8	Asynchronous/synchronous serial interface × 6 channels • I <sup>2</sup> C-bus (UART0 to UART3, UART5) • Special mode 2 (UART0 to UART3, UART5) • IEBus (optional <sup>(2)</sup> ) (UART0 to UART3, UART5)
A/D Converter		10-bit resolution × 20 channels Sample and hold functionality integrated
D/A Converter		8-bit resolution × 1
CRC Calculator		CRC-CCITT ( $X^{16} + X^{12} + X^5 + 1$ )
X-Y Converter		16 bits × 16 bits
Intelligent I/O		Time measurement (input capture): 16 bits × 16 Waveform generation (output compare): 16 bits × 19 Serial interface: Variable-length synchronous serial I/O mode, IEBus mode (optional <sup>(2)</sup> )
Flash Memory		Programming and erasure supply voltage: VCC1 = 3.0 to 5.5 V Minimum endurance: 1,000 program/erase cycles Security protection: ROM code protect, ID code protect Debugging: On-chip debug, on-board flash programming
Operating Frequency/Supply Voltage		50 MHz/VCC1 = 3.0 to 5.5 V
Operating Temperature		-40°C to 85°C (D version)
Current Consumption		32 mA (VCC1 = 5.0 V, f(CPU) = 50 MHz) 8 μA (VCC1 = 3.3 V, f(XCIN) = 32.768 kHz, in wait mode)
Package		64-pin plastic molded LQFP (PLQP0064KB-A)

## Notes:

1. Timer B4 is available in timer mode only.
2. Contact a Renesas Electronics sales office to use the optional features.

## 1.2 Product Information

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

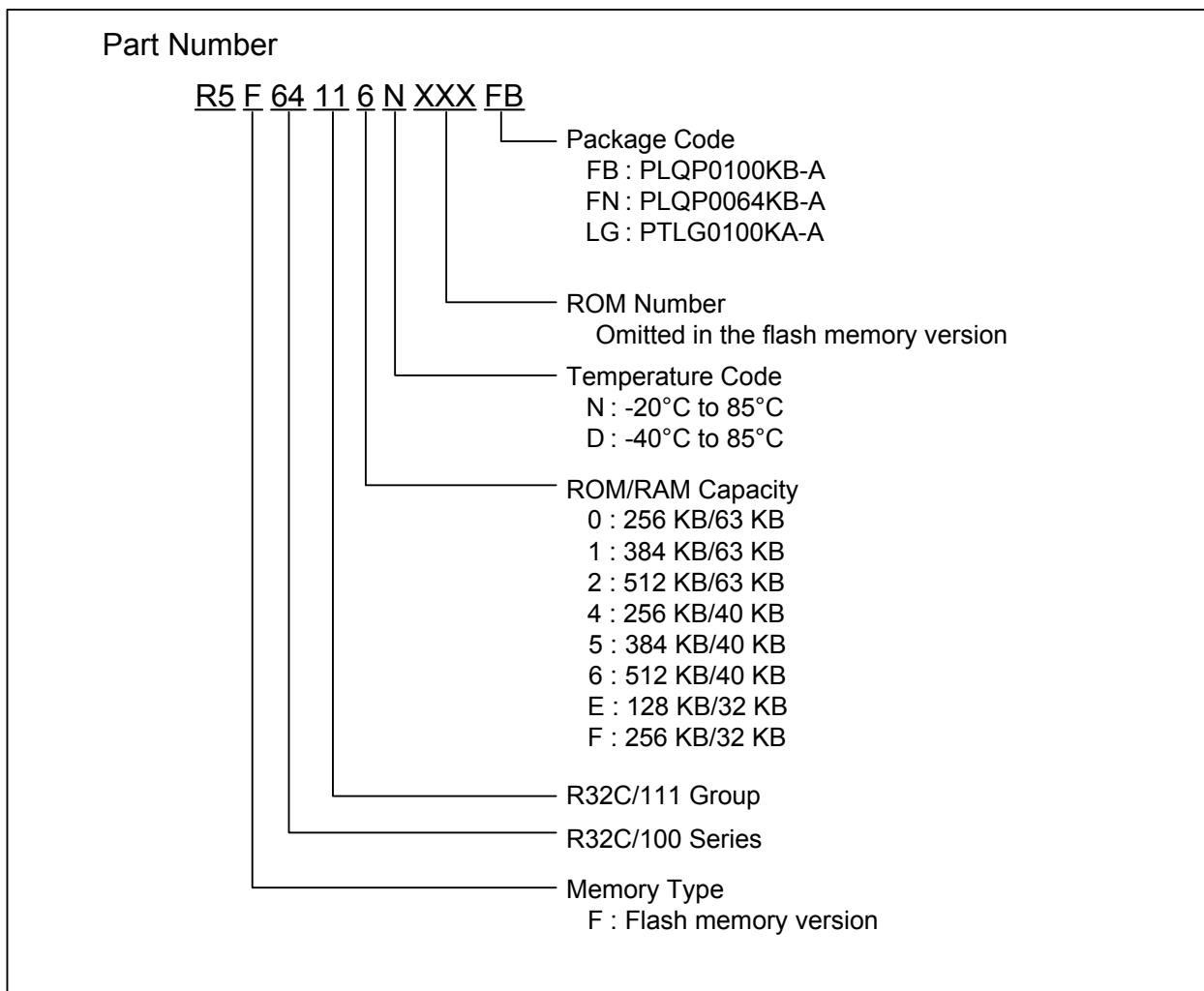
**Table 1.5 R32C/111 Group Product List**

**As of March, 2014**

Part Number	Package Code (1)	ROM Capacity (2)	RAM Capacity	Remarks
R5F64110DFB	PLQP0100KB-A	256 Kbytes +8 Kbytes	63 Kbytes	-40°C to 85°C (D version)
R5F64111DFB		384 Kbytes +8 Kbytes		
R5F64112DFB		512 Kbytes +8 Kbytes		
R5F64114DFB		256 Kbytes +8 Kbytes	40 Kbytes	
R5F64115DFB		384 Kbytes +8 Kbytes		
R5F64116DFB		512 Kbytes +8 Kbytes		
R5F64111NLG	PTLG0100KA-A	384 Kbytes +8 Kbytes	63 Kbytes	-20°C to 85°C (N version)
R5F64112NLG		512 Kbytes +8 Kbytes		
R5F6411FNLG		256 Kbytes +8 Kbytes	32 Kbytes	
R5F6411EDFN	PLQP064KB-A	128 Kbytes +8 Kbytes	32 Kbytes	-40°C to 85°C (D version)
R5F6411FDFN		256 Kbytes +8 Kbytes		

Notes:

- The old package codes are as follows:  
 PLQP0100KB-A: 100P6Q-A  
 PTLG0100KA-A: 100F0M  
 PLQP0064KB-A: 64P6Q-A
- “+ 8 Kbytes” in the ROM capacity column indicates the data flash capacity.



**Figure 1.1 Part Numbering**

### 1.3 Block Diagram

Figures 1.2 and 1.3 show block diagram of the R32C/111 Group.

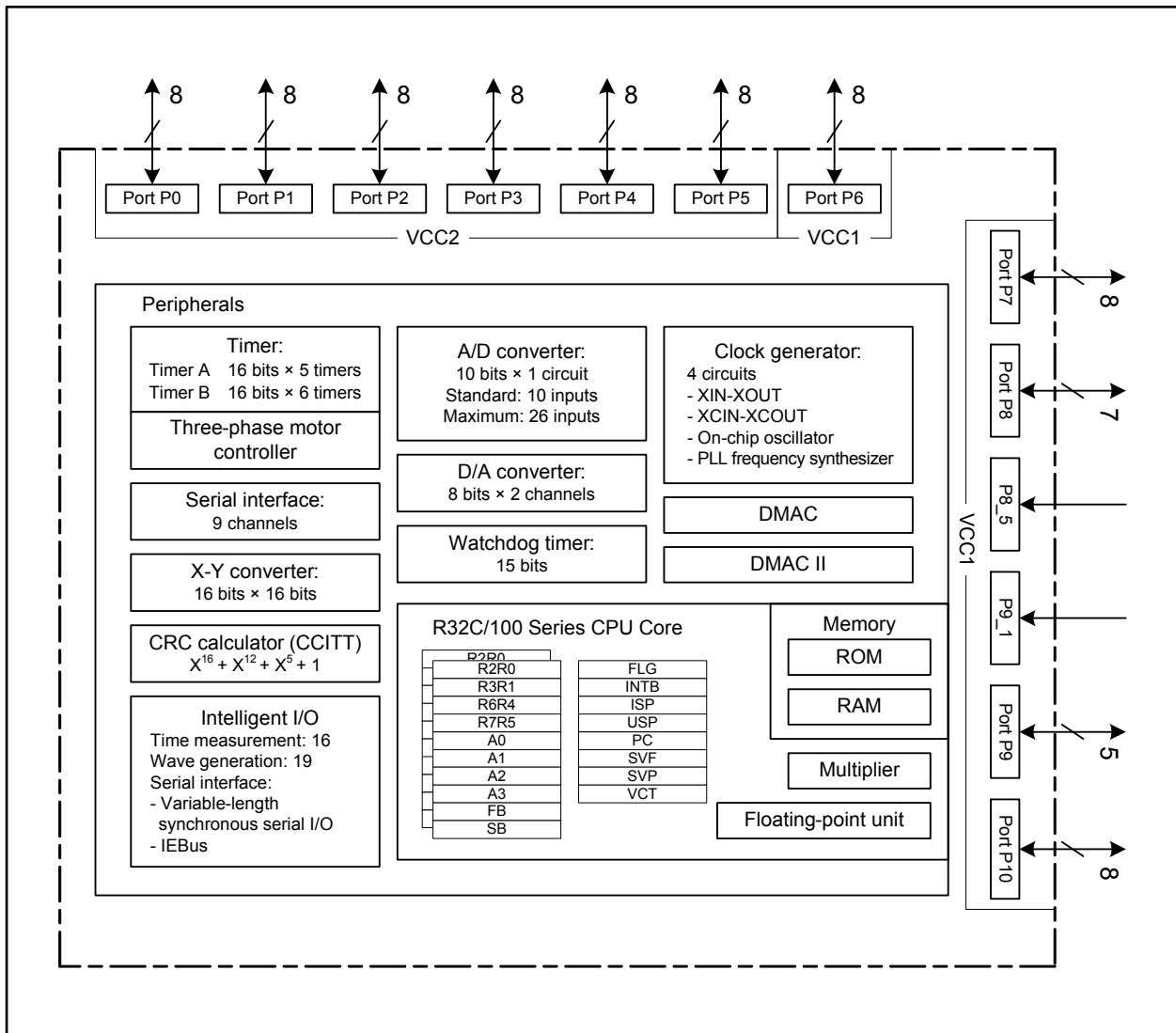


Figure 1.2 R32C/111 Group Block Diagram for the 100-pin Package



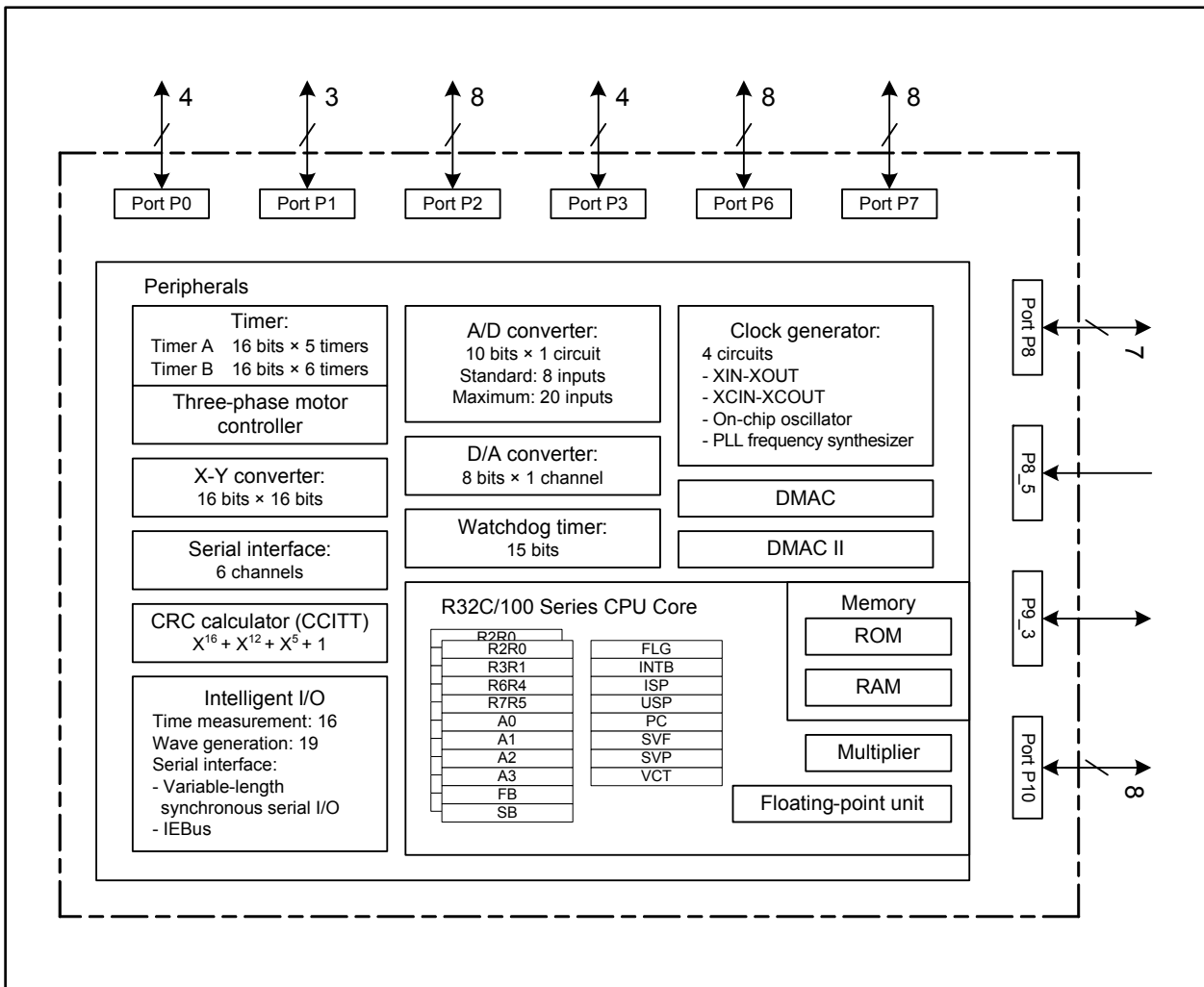


Figure 1.3 R32C/111 Group Block Diagram for the 64-pin Package

### 1.4 Pin Assignments

Figures 1.4 to 1.6 show the pin assignments (top view) and Tables 1.6 to 1.10 show the pin characteristics.

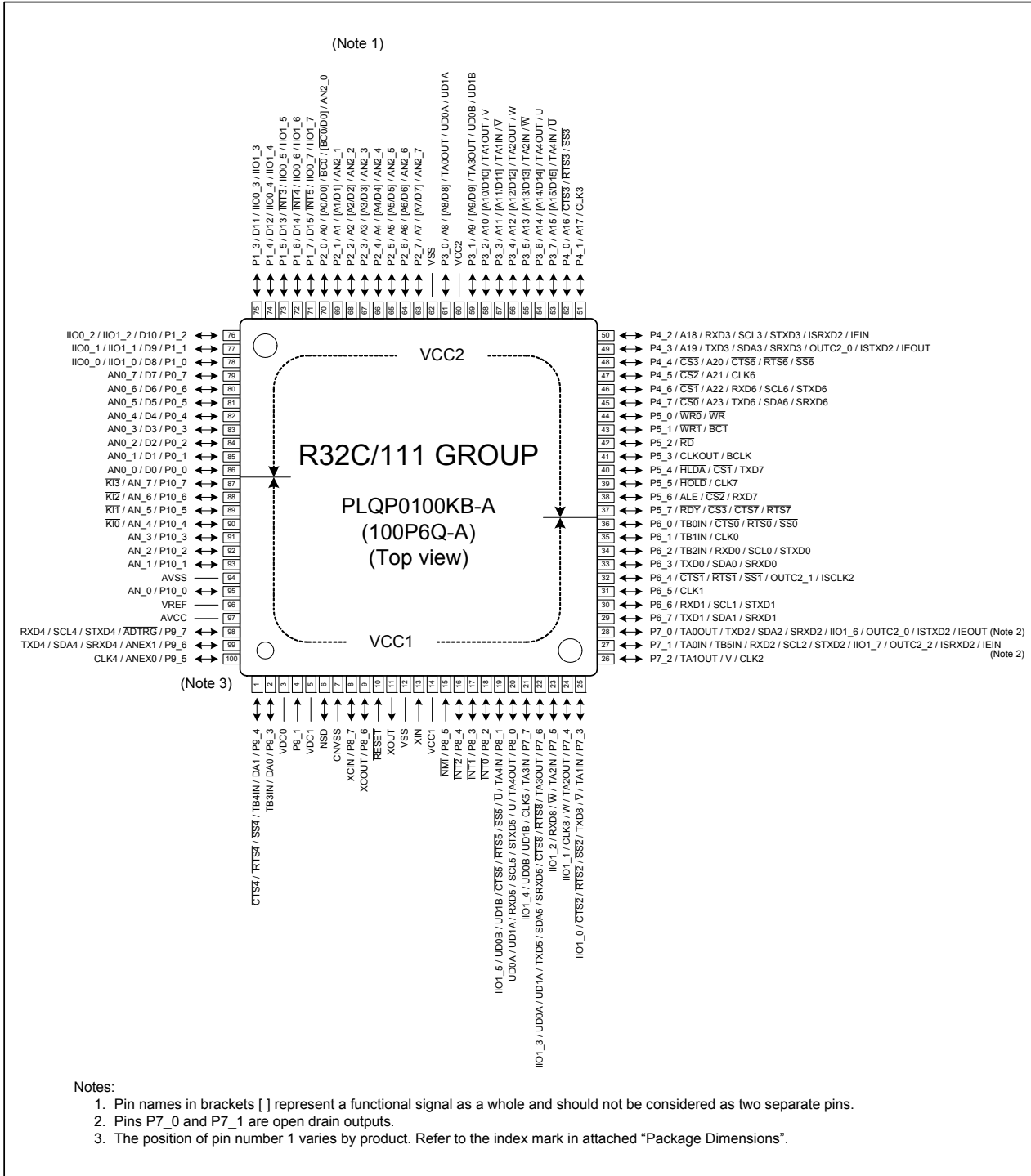


Figure 1.4 Pin Assignment for the 100-pin Package (top view)

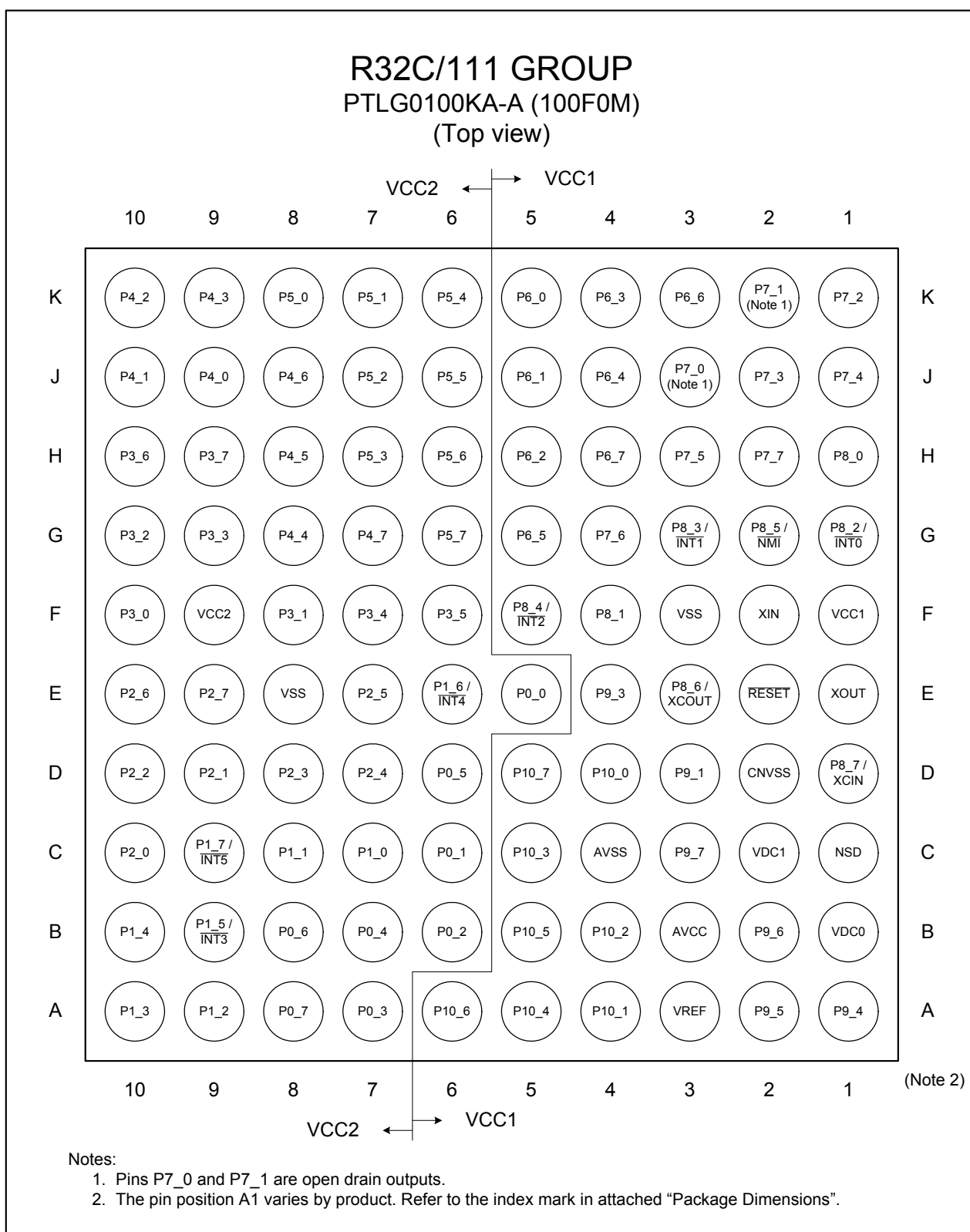


Figure 1.5 Pin Assignment for the 100-pin LGA Package (top view)

**Table 1.6 Pin Characteristics for the 100-pin Package (1/3)**

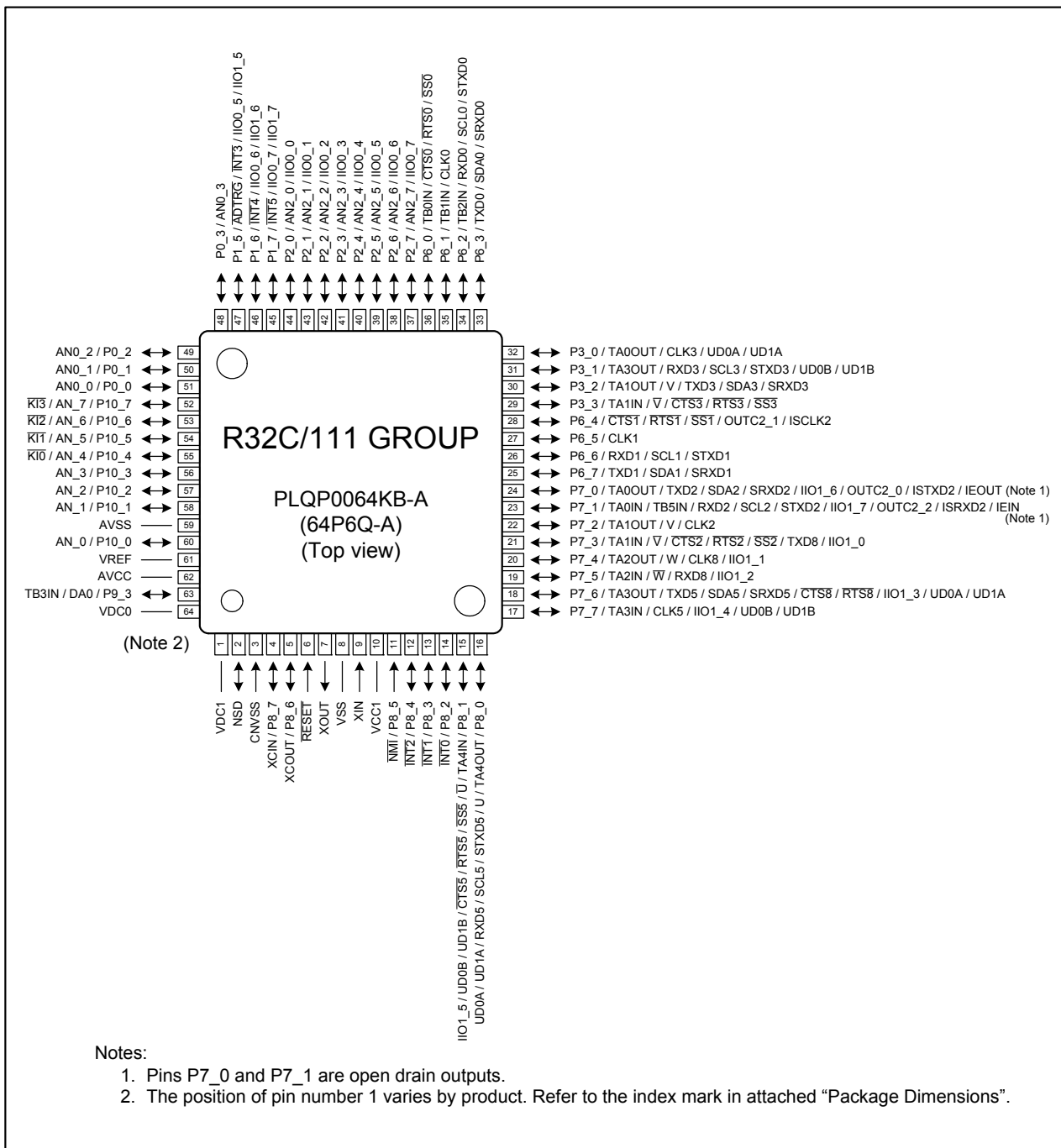
Pin No.		Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
QFP	LGA								
1	A1		P9_4		TB4IN	CTS4/RTS4/SS4		DA1	
2	E4		P9_3		TB3IN			DA0	
3	B1	VDC0							
4	D3		P9_1						
5	C2	VDC1							
6	C1	NSD							
7	D2	CNVSS							
8	D1	XCIN	P8_7						
9	E3	XCOU	P8_6						
10	E2	RESET							
11	E1	XOUT							
12	F3	VSS							
13	F2	XIN							
14	F1	VCC1							
15	G2		P8_5	NMI					
16	F5		P8_4	INT2					
17	G3		P8_3	INT1					
18	G1		P8_2	INT0					
19	F4		P8_1		TA4IN/U	CTS5/RTS5/SS5	IIO1_5/UD0B/UD1B		
20	H1		P8_0		TA4OUT/U	RXD5/SCL5/STXD5	UD0A/UD1A		
21	H2		P7_7		TA3IN	CLK5	IIO1_4/UD0B/UD1B		
22	G4		P7_6		TA3OUT	TXD5/SDA5/ SRXD5/CTS8/RTS8	IIO1_3/UD0A/UD1A		
23	H3		P7_5		TA2IN/W	RXD8	IIO1_2		
24	J1		P7_4		TA2OUT/W	CLK8	IIO1_1		
25	J2		P7_3		TA1IN/V	CTS2/RTS2/SS2/ TXD8	IIO1_0		
26	K1		P7_2		TA1OUT/V	CLK2			
27	K2		P7_1		TA0IN/ TB5IN	RXD2/SCL2/STXD2	IIO1_7/OUTC2_2/ ISRXD2/IEIN		
28	J3		P7_0		TA0OUT	TXD2/SDA2/SRXD2	IIO1_6/OUTC2_0/ ISTXD2/IEOUT		
29	H4		P6_7			TXD1/SDA1/SRXD1			
30	K3		P6_6			RXD1/SCL1/STXD1			
31	G5		P6_5			CLK1			
32	J4		P6_4			CTS1/RTS1/SS1	OUTC2_1/ISCLK2		
33	K4		P6_3			TXD0/SDA0/SRXD0			
34	H5		P6_2		TB2IN	RXD0/SCL0/STXD0			
35	J5		P6_1		TB1IN	CLK0			
36	K5		P6_0		TB0IN	CTS0/RTS0/SS0			
37	G6		P5_7			CTS7/RTS7			RDY/CS3
38	H6		P5_6			RXD7			ALE/CS2
39	J6		P5_5			CLK7			HOLD

**Table 1.7 Pin Characteristics for the 100-pin Package (2/3)**

Pin No.		Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
QFP	LGA								
40	K6		P5_4			TXD7			HLDA/CS1
41	H7		P5_3						CLKOUT/ BCLK
42	J7		P5_2						RD
43	K7		P5_1						WR1/BC1
44	K8		P5_0						WR0/WR
45	G7		P4_7			TXD6/SDA6/SRXD6			CS0/A23
46	J8		P4_6			RXD6/SCL6/STXD6			CS1/A22
47	H8		P4_5			CLK6			CS2/A21
48	G8		P4_4			CTS6/RTS6/SS6			CS3/A20
49	K9		P4_3			TXD3/SDA3/SRXD3	OUTC2_0/ISTXD2/ IEOUT		A19
50	K10		P4_2			RXD3/SCL3/STXD3	ISRXD2/IEIN		A18
51	J10		P4_1			CLK3			A17
52	J9		P4_0			CTS3/RTS3/SS3			A16
53	H9		P3_7		TA4IN/U				A15(/D15)
54	H10		P3_6		TA4OUT/U				A14(/D14)
55	F6		P3_5		TA2IN/W				A13(/D13)
56	F7		P3_4		TA2OUT/W				A12(/D12)
57	G9		P3_3		TA1IN/V				A11(/D11)
58	G10		P3_2		TA1OUT/V				A10(/D10)
59	F8		P3_1		TA3OUT		UD0B/UD1B		A9(/D9)
60	F9	VCC2							
61	F10		P3_0		TA0OUT		UD0A/UD1A		A8(/D8)
62	E8	VSS							
63	E9		P2_7					AN2_7	A7(/D7)
64	E10		P2_6					AN2_6	A6(/D6)
65	E7		P2_5					AN2_5	A5(/D5)
66	D7		P2_4					AN2_4	A4(/D4)
67	D8		P2_3					AN2_3	A3(/D3)
68	D10		P2_2					AN2_2	A2(/D2)
69	D9		P2_1					AN2_1	A1(/D1)
70	C10		P2_0					AN2_0	A0(/D0)/ BC0(/D0)
71	C9		P1_7	INT5			IIO0_7/IIO1_7		D15
72	E6		P1_6	INT4			IIO0_6/IIO1_6		D14
73	B9		P1_5	INT3			IIO0_5/IIO1_5		D13
74	B10		P1_4				IIO0_4/IIO1_4		D12
75	A10		P1_3				IIO0_3/IIO1_3		D11
76	A9		P1_2				IIO0_2/IIO1_2		D10
77	C8		P1_1				IIO0_1/IIO1_1		D9
78	C7		P1_0				IIO0_0/IIO1_0		D8
79	A8		P0_7					AN0_7	D7

**Table 1.8 Pin Characteristics for the 100-pin Package (3/3)**

Pin No.		Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
QFP	LGA								
80	B8		P0_6					AN0_6	D6
81	D6		P0_5					AN0_5	D5
82	B7		P0_4					AN0_4	D4
83	A7		P0_3					AN0_3	D3
84	B6		P0_2					AN0_2	D2
85	C6		P0_1					AN0_1	D1
86	E5		P0_0					AN0_0	D0
87	D5		P10_7	KI3				AN_7	
88	A6		P10_6	KI2				AN_6	
89	B5		P10_5	KI1				AN_5	
90	A5		P10_4	KI0				AN_4	
91	C5		P10_3					AN_3	
92	B4		P10_2					AN_2	
93	A4		P10_1					AN_1	
94	C4	AVSS							
95	D4		P10_0					AN_0	
96	A3	VREF							
97	B3	AVCC							
98	C3		P9_7			RXD4/SCL4/STXD4		ADTRG	
99	B2		P9_6			TXD4/SDA4/SRXD4		ANEX1	
100	A2		P9_5			CLK4		ANEX0	



**Figure 1.6 Pin Assignment for the 64-pin Package (top view)**

**Table 1.9 Pin Characteristics for the 64-pin Package (1/2)**

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin
1	VDC1						
2	NSD						
3	CNVSS						
4	XCIN	P8_7					
5	XCOU	P8_6					
6	RESET						
7	XOUT						
8	VSS						
9	XIN						
10	VCC1						
11		P8_5	NMI				
12		P8_4	INT2				
13		P8_3	INT1				
14		P8_2	INT0				
15		P8_1		TA4IN/U	CTS5/RTS5/SS5	IIO1_5/UD0B/UD1B	
16		P8_0		TA4OUT/U	RXD5/SCL5/STXD5	UD0A/UD1A	
17		P7_7		TA3IN	CLK5	IIO1_4/UD0B/UD1B	
18		P7_6		TA3OUT	TXD5/SDA5/SRXD5/ CTS8/RTS8	IIO1_3/UD0A/UD1A	
19		P7_5		TA2IN/W	RXD8	IIO1_2	
20		P7_4		TA2OUT/W	CLK8	IIO1_1	
21		P7_3		TA1IN/V	CTS2/RTS2/SS2/TXD8	IIO1_0	
22		P7_2		TA1OUT/V	CLK2		
23		P7_1		TA0IN/ TB5IN	RXD2/SCL2/STXD2	IIO1_7/OUTC2_2/ ISRXD2/IEIN	
24		P7_0		TA0OUT	TXD2/SDA2/SRXD2	IIO1_6/OUTC2_0/ ISTXD2/IEOUT	
25		P6_7			TXD1/SDA1/SRXD1		
26		P6_6			RXD1/SCL1/STXD1		
27		P6_5			CLK1		
28		P6_4			CTS1/RTS1/SS1	OUTC2_1/ISCLK2	
29		P3_3		TA1IN/V	CTS3/RTS3/SS3		
30		P3_2		TA1OUT/V	TXD3/SDA3/SRXD3		
31		P3_1		TA3OUT	RXD3/SCL3/STXD3	UD0B/UD1B	
32		P3_0		TA0OUT	CLK3	UD0A/UD1A	
33		P6_3			TXD0/SDA0/SRXD0		
34		P6_2		TB2IN	RXD0/SCL0/STXD0		
35		P6_1		TB1IN	CLK0		
36		P6_0		TB0IN	CTS0/RTS0/SS0		
37		P2_7				IIO0_7	AN2_7
38		P2_6				IIO0_6	AN2_6
39		P2_5				IIO0_5	AN2_5
40		P2_4				IIO0_4	AN2_4



**Table 1.10 Pin Characteristics for the 64-pin Package (2/2)**

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Intelligent I/O Pin	Analog Pin
41		P2_3				IIO0_3	AN2_3
42		P2_2				IIO0_2	AN2_2
43		P2_1				IIO0_1	AN2_1
44		P2_0				IIO0_0	AN2_0
45		P1_7	$\overline{\text{INT5}}$			IIO0_7/IIO1_7	
46		P1_6	$\overline{\text{INT4}}$			IIO0_6/IIO1_6	
47		P1_5	$\overline{\text{INT3}}$			IIO0_5/IIO1_5	$\overline{\text{ADTRG}}$
48		P0_3					AN0_3
49		P0_2					AN0_2
50		P0_1					AN0_1
51		P0_0					AN0_0
52		P10_7	$\overline{\text{KI3}}$				AN_7
53		P10_6	$\overline{\text{KI2}}$				AN_6
54		P10_5	$\overline{\text{KI1}}$				AN_5
55		P10_4	$\overline{\text{KI0}}$				AN_4
56		P10_3					AN_3
57		P10_2					AN_2
58		P10_1					AN_1
59	AVSS						
60		P10_0					AN_0
61	VREF						
62	AVCC						
63		P9_3		TB3IN			DA0
64	VDC0						

## 1.5 Pin Definitions and Functions

Tables 1.11 to 1.17 show the pin definitions and functions.

**Table 1.11 Pin Definitions and Functions for the 100-pin Package (1/4)**

Function	Symbol	I/O	Power Supply	Description
Power supply	VCC1, VCC2, VSS	I	—	Applicable as follows: VCC1 and VCC2 = 3.0 to 5.5 V ( $VCC1 \geq VCC2$ ), VSS = 0 V
Connecting pins for decoupling capacitor	VDC0, VDC1	—	—	A decoupling capacitor for internal voltage should be connected between VDC0 and VDC1
Analog power supply	AVCC, AVSS	I	VCC1	Power supply for the A/D converter. AVCC and AVSS should be connected to VCC1 and VSS, respectively
Reset input	RESET	I	VCC1	The MCU is reset when this pin is driven low
CNVSS	CNVSS	I	VCC1	This pin should be connected to VSS via a resistor
Debug port	NSD	I/O	VCC1	This pin is to communicate with a debugger. It should be connected to VCC1 via a resistor of 1 to 4.7 k $\Omega$
Main clock input	XIN	I	VCC1	Input/output for the main clock oscillator. A crystal, or a ceramic resonator should be connected between pins XIN and XOUT. An external clock should be input at the XIN while leaving the XOUT open
Main clock output	XOUT	O	VCC1	
Sub clock input	XCIN	I	VCC1	Input/output for the sub clock oscillator. A crystal oscillator should be connected between pins XCIN and XCOU. An external clock should be input at the XCIN while leaving the XCOU open
Sub clock output	XCOU	O	VCC1	
BCLK output	BCLK	O	VCC2	BCLK output
Clock output	CLKOUT	O	VCC2	Output of the clock with the same frequency as low speed clocks, f8, or f32
External interrupt input	INT0 to INT5	I	VCC1 VCC2	Input for external interrupts
NMI input	P8_5/NMI	I	VCC1	Input for NMI
Key input interrupt	KI0 to KI3	I	VCC1	Input for the key input interrupt
Bus control pins	D0 to D7	I/O	VCC2	Input/output of data (D0 to D7) while accessing an external memory space with a separate bus
	D8 to D15	I/O	VCC2	Input/output of data (D8 to D15) while accessing an external memory space with 16-bit separate bus
	A0 to A23	O	VCC2	Output of address bits A0 to A23

**Table 1.12 Pin Definitions and Functions for the 100-pin Package (2/4)**

Function	Symbol	I/O	Power Supply	Description
Bus control pins	A0/D0 to A7/D7	I/O	VCC2	Output of address bits (A0 to A7) and input/output of data (D0 to D7) by time-division while accessing an external memory space with multiplexed bus
	A8/D8 to A15/D15	I/O	VCC2	Output of address bits (A8 to A15) and input/output of data (D8 to D15) by time-division while accessing an external memory space with 16-bit multiplexed bus
	$\overline{BC0}/D0$	I/O	VCC2	Output of byte control ( $\overline{BC0}$ ) and input/output of data (D0) by time-division while accessing an external memory space with multiplexed bus
	$\overline{CS0}$ to $\overline{CS3}$	O	VCC2	Chip select output
	$\overline{WR0}/\overline{WR1}/\overline{WR}/\overline{BC0}/\overline{BC1}/\overline{RD}$	O	VCC2	Output of write, byte control, and read signals. Either $\overline{WRx}$ or $\overline{WR}$ and $\overline{BCx}$ can be selected by a program. Data is read when $\overline{RD}$ is low.  <ul style="list-style-type: none"> <li>When <math>\overline{WR0}</math>, <math>\overline{WR1}</math>, and <math>\overline{RD}</math> are selected, data is written to the following address: an even address, when <math>\overline{WR0}</math> is low an odd address, when <math>\overline{WR1}</math> is low on 16-bit external data bus</li> <li>When <math>\overline{WR}</math>, <math>\overline{BC0}</math>, <math>\overline{BC1}</math>, and <math>\overline{RD}</math> are selected, data is written, when <math>\overline{WR}</math> is low and the following address is accessed: an even address, when <math>\overline{BC0}</math> is low an odd address, when <math>\overline{BC1}</math> is low on 16-bit external data bus</li> </ul>
	ALE	O	VCC2	Latch enable signal in multiplexed bus format
	$\overline{HOLD}$	I	VCC2	The MCU is in a hold state while this pin is held low
	HLDA	O	VCC2	This pin is driven low while the MCU is held in a hold state
	RDY	I	VCC2	Bus cycle is extended by the CPU if this pin is low on the falling edge of BCLK

**Table 1.13 Pin Definitions and Functions for the 100-pin Package (3/4)**

Function	Symbol	I/O	Power Supply	Description
I/O port	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7,	I/O	VCC2	I/O ports in CMOS. Each port can be programmed to input or output under the control of the direction register. Pull-up resistors are selected for the following 4-pin units: Pi_0 to Pi_3 and Pi_4 to Pi_7 (i = 0 to 10); however, they are enabled only for the input pins.
	P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3 to P9_7, P10_0 to P10_7	I/O	VCC1	P7_0 and P7_1 outputs are N-channel open drain
Input port	P9_1	I	VCC1	Input port in CMOS. Pull-up resistors are selectable for P9_1 and P9_3
Timer A	TA0OUT to TA4OUT	I/O	VCC1 VCC2	Timers A0 to A4 input/output. TA0OUT output assigned for port P7_0 is N-channel open drain
	TA0IN to TA4IN	I	VCC1 VCC2	Timers A0 to A4 input
Timer B	TB0IN to TB5IN	I	VCC1	Timers B0 to B5 input
Three-phase motor control timer output	U, $\bar{U}$ , V, $\bar{V}$ , W, $\bar{W}$	O	VCC1 VCC2	Three-phase motor control timer output
Serial interface	$\overline{CTS0}$ to $\overline{CTS8}$	I	VCC1 VCC2	Handshake input
	$\overline{RTS0}$ to $\overline{RTS8}$	O	VCC1 VCC2	Handshake output
	CLK0 to CLK8	I/O	VCC1 VCC2	Transmit/receive clock input/output
	RXD0 to RXD8	I	VCC1 VCC2	Serial data input
	TXD0 to TXD8	O	VCC1 VCC2	Serial data output. TXD2 output is N-channel open drain
I <sup>2</sup> C-bus (simplified)	SDA0 to SDA6	I/O	VCC1 VCC2	Serial data input/output. SDA2 output is N-channel open drain
	SCL0 to SCL6	I/O	VCC1 VCC2	Transmit/receive clock input/output. SCL2 output is N-channel open drain

**Table 1.14 Pin Definitions and Functions for the 100-pin Package (4/4)**

Function	Symbol	I/O	Power Supply	Description
Serial interface special functions	STXD0 to STXD6	O	VCC1 VCC2	Serial data output in slave mode. STXD2 is N-channel open drain
	SRXD0 to SRXD6	I	VCC1 VCC2	Serial data input in slave mode
	$\overline{SS}0$ to $\overline{SS}6$	I	VCC1 VCC2	Input to control serial interface special functions
A/D converter	AN_0 to AN_7	I	VCC1	Analog input for the A/D converter
	AN0_0 to AN0_7, AN2_0 to AN2_7	I	VCC2	
	ADTRG	I	VCC1	External trigger input for the A/D converter
	ANEX0	I/O	VCC1	Expanded analog input for the A/D converter and output in external op-amp connection mode
	ANEX1	I	VCC1	Expanded analog input for the A/D converter
D/A converter	DA0, DA1	O	VCC1	Output for the D/A converter
Reference voltage input	VREF	I	–	Reference voltage input for the A/D converter and D/A converter
Intelligent I/O	IIO0_0 to IIO0_7	I/O	VCC1 VCC2	Input/output for Intelligent I/O group 0. Either input capture or output compare is selectable
	IIO1_0 to IIO1_7	I/O	VCC1 VCC2	Input/output for Intelligent I/O group 1. Either input capture or output compare is selectable. IIO1_6 and IIO1_7 outputs assigned for ports P7_0 and P7_1 are N-channel open drain
	UD0A, UD0B, UD1A, UD1B	I	VCC1 VCC2	Input for the two-phase encoder
	OUTC2_0 to OUTC2_2	O	VCC1 VCC2	Output for OC (output compare) of Intelligent I/O group 2. OUTC2_0 and OUTC2_2 assigned for ports P7_0 and P7_1 are N-channel open drain
	ISCLK2	I/O	VCC1 VCC2	Clock input/output for the serial interface
	ISRXD2	I		Receive data input for the serial interface
	ISTXD2	O		Transmit data output for the serial interface. ISTXD2 assigned for port P7_0 is N-channel open drain
	IEIN	I	VCC1 VCC2	Receive data input for the serial interface
	IEOUT	O		Transmit data output for the serial interface. IEOUT assigned for port P7_0 is N-channel open drain

**Table 1.15 Pin Definitions and Functions for 64-pin Package (1/3)**

Function	Symbol	I/O	Description
Power supply	VCC1, VSS	I	Applicable as follows: VCC1 = 3.0 to 5.5 V, VSS = 0 V
Connecting pins for decoupling capacitor	VDC0, VDC1	—	A decoupling capacitor for internal voltage should be connected between VDC0 and VDC1
Analog power supply	AVCC, AVSS	I	Power supply for the A/D converter. AVCC and AVSS should be connected to VCC and VSS, respectively
Reset input	$\overline{\text{RESET}}$	I	The MCU is reset when this pin is driven low
CNVSS	CNVSS	I	This pin should be connected to VSS via a resistor
Debug port	NSD	I/O	This pin is to communicate with a debugger. It should be connected to VCC1 via a resistor of 1 to 4.7 k $\Omega$
Main clock input	XIN	I	Input/output for the main clock oscillator. A crystal, or a ceramic resonator should be connected between pins XIN and XOUT. An external clock should be input at the XIN while leaving the XOUT open
Main clock output	XOUT	O	
Sub clock input	XCIN	I	Input/output for the sub clock oscillator. A crystal oscillator should be connected between pins XCIN and XCOU. An external clock should be input at the XCIN while leaving the XCOU open
Sub clock output	XCOU	O	
External interrupt input	INT0 to INT5	I	Input for external interrupts
NMI input	P8_5/ $\overline{\text{NMI}}$	I	Input for NMI
Key input interrupt	KI0 to KI3	I	Input for the key input interrupt
I/O port	P0_0 to P0_3, P1_5 to P1_7, P2_0 to P2_7, P3_0 to P3_3, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3, P10_0 to P10_7	I/O	I/O ports in CMOS. Each port can be programmed to input or output under the control of the direction register. Pull-up resistors are selected for the following 4-pin units: Pi_0 to Pi_3 and Pi_4 to Pi_7 (i = 0 to 3, 6 to 10); however, they are enabled only for the input pins. P7_0 and P7_1 outputs are N-channel open drain
Timer A	TA0OUT to TA4OUT	I/O	Timers A0 to A4 input/output. TA0OUT output assigned for port P7_0 is N-channel open drain
	TA0IN to TA4IN	I	Timers A0 to A4 input
Timer B	TB0IN to TB3IN, TB5IN	I	Timers B0 to B3, and B5 input
Three-phase motor control timer output	U, $\overline{\text{U}}$ , V, $\overline{\text{V}}$ , W, $\overline{\text{W}}$	O	Three-phase motor control timer output

**Table 1.16 Pin Definitions and Functions for the 64-pin Package (2/3)**

Function	Symbol	I/O	Description
Serial interface	$\overline{\text{CTS0}}$ to $\overline{\text{CTS3}}$ , $\overline{\text{CTS5}}$ , $\overline{\text{CTS8}}$	I	Handshake input
	$\overline{\text{RTS0}}$ to $\overline{\text{RTS3}}$ , $\overline{\text{RTS5}}$ , $\overline{\text{RTS8}}$	O	Handshake output
	CLK0 to CLK3, CLK5, CLK8	I/O	Transmit/receive clock input/output
	RXD0 to RXD3, RXD5, RXD8	I	Serial data input
	TXD0 to TXD3, TXD5, TXD8	O	Serial data output. TXD2 output is N-channel open drain
I <sup>2</sup> C-bus (simplified)	SDA0 to SDA3, SDA5	I/O	Serial data input/output. SDA2 output is N-channel open drain
	SCL0 to SCL3, SCL5	I/O	Transmit/receive clock input/output. SCL2 output is N-channel open drain
Serial interface special functions	STXD0 to STXD3, STXD5	O	Serial data output in slave mode. STXD2 is N-channel open drain
	SRXD0 to SRXD3, SRXD5	I	Serial data input in slave mode
	$\overline{\text{SS0}}$ to $\overline{\text{SS3}}$ , $\overline{\text{SS5}}$	I	Input to control serial interface special functions
A/D converter	AN_0 to AN_7, AN0_0 to AN0_3, AN2_0 to AN2_7	I	Analog input for the A/D converter
	ADTRG	I	External trigger input for the A/D converter
D/A converter	DA0	O	Output for the D/A converter
Reference voltage input	VREF	I	Reference voltage input for the A/D converter and D/A converter

**Table 1.17 Pin Definitions and Functions for the 64-pin Package (3/3)**

Function	Symbol	I/O	Description
Intelligent I/O	IIO0_0 to IIO0_7	I/O	Input/output for Intelligent I/O group 0. Either input capture or output compare is selectable
	IIO1_0 to IIO1_7	I/O	Input/output for Intelligent I/O group 1. Either input capture or output compare is selectable. IIO1_6 and IIO1_7 outputs assigned for ports P7_0 and P7_1 are N-channel open drain
	UD0A, UD0B, UD1A, UD1B	I	Input for the two-phase encoder
	OUTC2_0 to OUTC2_2	O	Output for OC (output compare) of Intelligent I/O group 2. OUTC2_0 and OUTC2_2 assigned for ports P7_0 and P7_1 are N-channel open drain
	ISCLK2	I/O	Clock input/output for the serial interface
	ISRXD2	I	Receive data input for the serial interface
	ISTXD2	O	Transmit data output for the serial interface. ISTXD2 assigned for port P7_0 is N-channel open drain
	IEIN	I	Receive data input for the serial interface
	IEOUT	O	Transmit data output for the serial interface. IEOUT assigned for port P7_0 is N-channel open drain



## 2. Central Processing Unit (CPU)

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

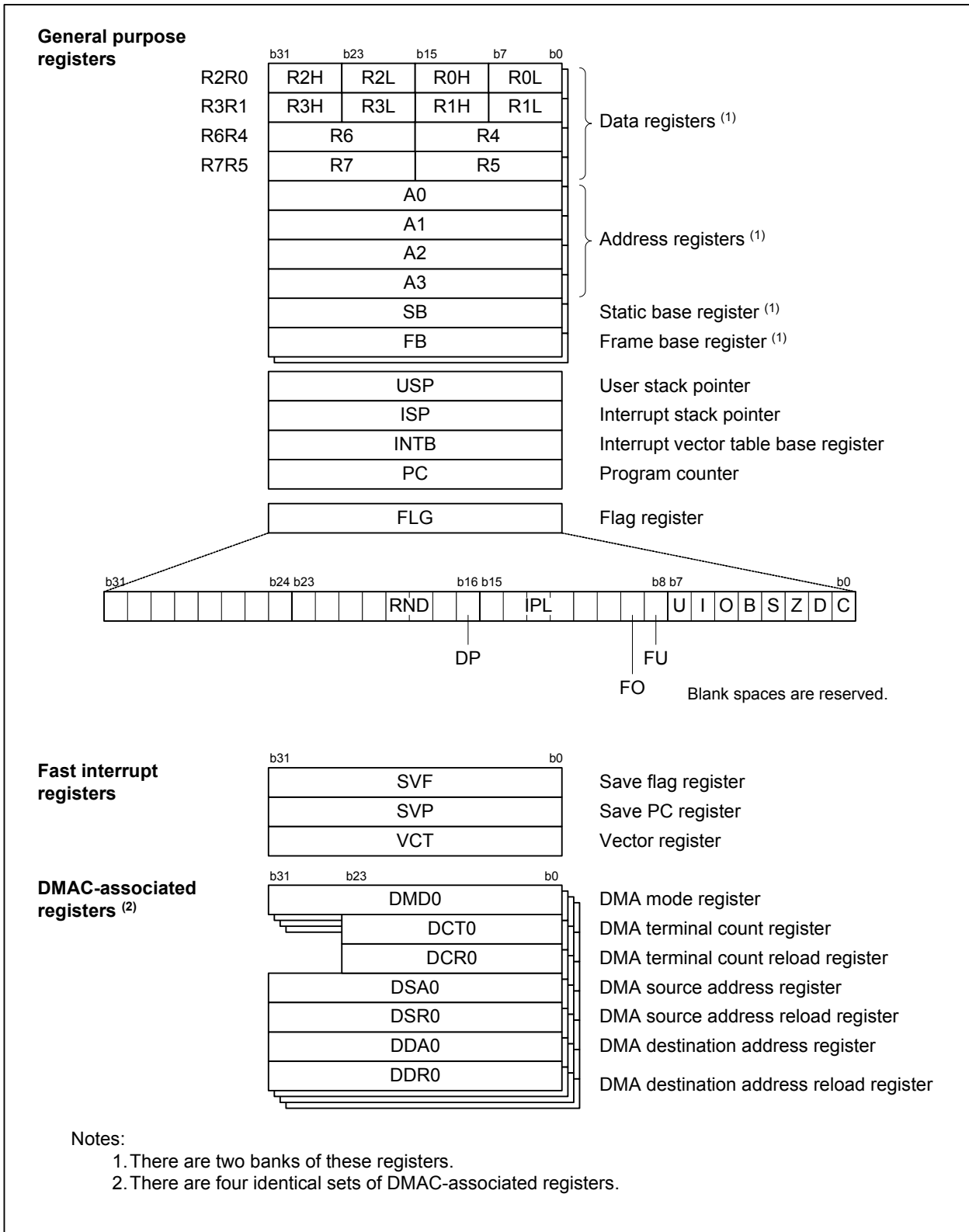


Figure 2.1 CPU Registers

## 2.1 General Purpose Registers

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

These 32-bit registers are primarily used for transfers and arithmetic/logic operations.

Each of the registers can be divided into upper and lower 16-bit registers, e.g. R2R0 can be divided into R2 and R0, R3R1 can be divided into R3 and R1, etc.

Moreover, data registers R2R0 and R3R1 can be divided into four 8-bit data registers: upper (R2H and R3H), mid-upper (R2L and R3L), mid-lower (R0H and R1H), and lower (R0L and R1L).

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

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

### 2.1.3 Static Base Register (SB)

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

### 2.1.4 Frame Base Register (FB)

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

### 2.1.5 Program Counter (PC)

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

### 2.1.6 Interrupt Vector Table Base Register (INTB)

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

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

Two types of 32-bit stack pointers (SPs) are provided: user stack pointer (USP) and interrupt stack pointer (ISP).

Use the stack pointer select flag (U flag) to select either the user stack pointer (USP) or the interrupt stack pointer (ISP). The U flag is bit 7 in the flag register (FLG). Refer to 2.1.8 "Flag Register (FLG)" for details.

To minimize the overhead of interrupt sequence due to less memory access, set the user stack pointer (USP) or the interrupt stack pointer (ISP) to a multiple of 4.

### 2.1.8 Flag Register (FLG)

This 32-bit register indicates the CPU status.

#### 2.1.8.1 Carry Flag (C flag)

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

#### 2.1.8.2 Debug Flag (D flag)

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

#### 2.1.8.3 Zero Flag (Z flag)

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

#### 2.1.8.4 Sign Flag (S flag)

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

### 2.1.8.5 Register Bank Select Flag (B flag)

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

### 2.1.8.6 Overflow Flag (O flag)

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

### 2.1.8.7 Interrupt Enable Flag (I flag)

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

### 2.1.8.8 Stack Pointer Select Flag (U flag)

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

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

### 2.1.8.9 Floating-point Underflow Flag (FU flag)

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

### 2.1.8.10 Floating-point Overflow Flag (FO flag)

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

### 2.1.8.11 Processor Interrupt Priority Level (IPL)

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

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

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

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

### 2.1.8.13 Floating-point Rounding Mode (RND)

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

### 2.1.8.14 Reserved

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

## 2.2 Fast Interrupt Registers

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

### 2.2.1 Save Flag Register (SVF)

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

### 2.2.2 Save PC Register (SVP)

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

### 2.2.3 Vector Register (VCT)

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

## 2.3 DMAC-associated Registers

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### 3. Memory

Figure 3.1 shows the memory map of the R32C/111 Group.

The R32C/111 Group provides a 4-Gbyte address space from 00000000h to FFFFFFFFh.

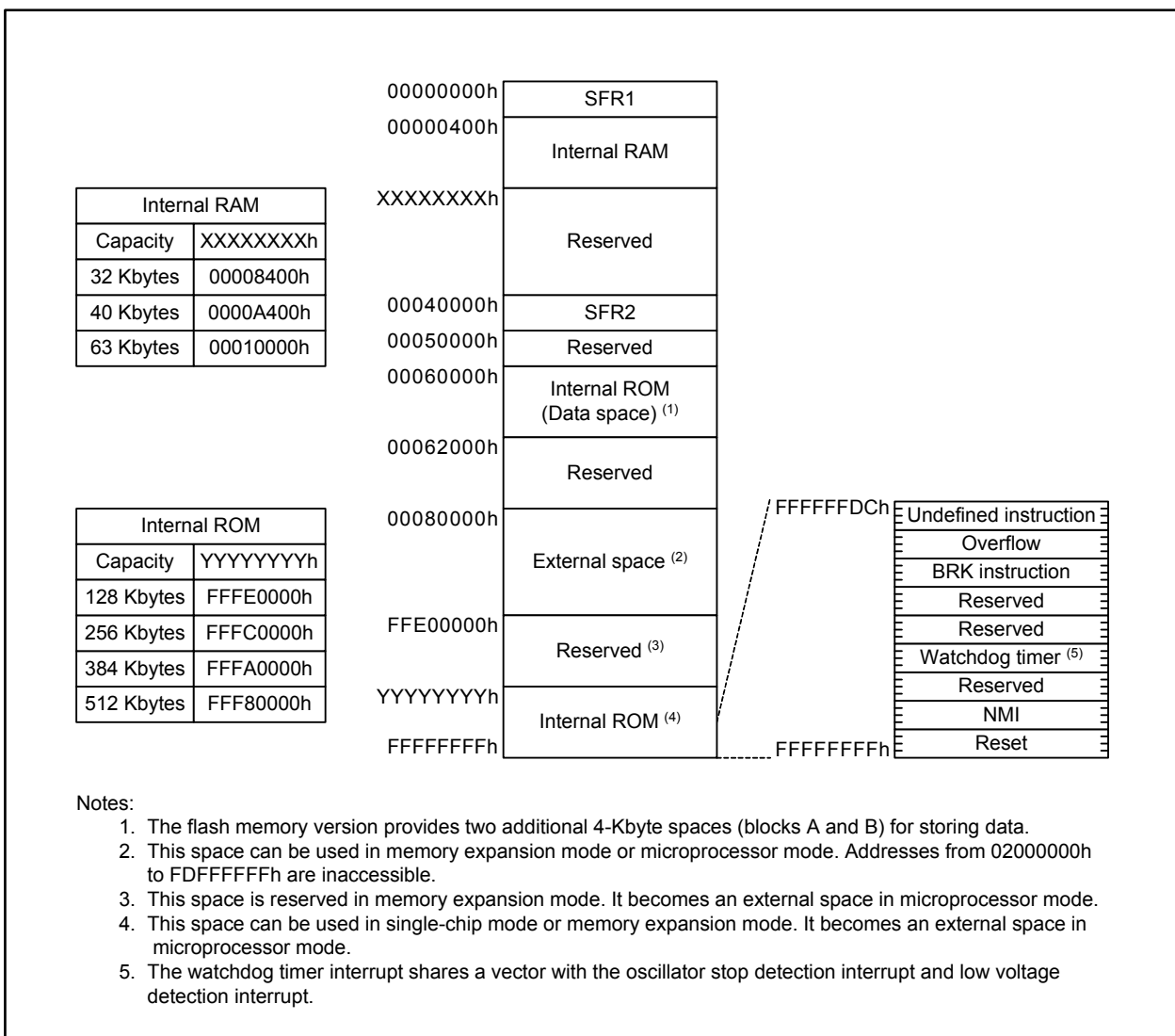
The internal ROM is mapped from address FFFFFFFFh in the inferior direction. For example, the 512-Kbyte internal ROM is mapped from FFF80000h to FFFFFFFFh.

The fixed interrupt vector table contains the start address of interrupt handlers and is mapped from FFFFFFFDCh to FFFFFFFFh.

The internal RAM is mapped from address 00000400h in the superior direction. For example, the 63-Kbyte internal RAM is mapped from 00000400h to 0000FFFFh. Besides being used for data storage, the internal RAM functions as a stack(s) for subroutine calls and/or interrupt handlers.

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

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



**Figure 3.1 Memory Map**

## 4. Special Function Registers (SFRs)

SFRs are memory-mapped peripheral registers that control the operation of peripherals. Tables 4.1 SFR List (1) to 4.24 SFR List (24) list the SFR details.

**Table 4.1 SFR List (1)**

Address	Register	Symbol	Reset Value
000000h			
000001h			
000002h			
000003h			
000004h	Clock Control Register	CCR	0001 1000b
000005h			
000006h	Flash Memory Control Register	FMCR	0000 0001b
000007h	Protect Release Register	PRR	00h
000008h			
000009h			
00000Ah			
00000Bh			
00000Ch			
00000Dh			
00000Eh			
00000Fh			
000010h	External Bus Control Register 3/Flash Memory Rewrite Bus	EBC3/FEBC3	0000h
000011h	Control Register 3		
000012h	Chip Selects 2 and 3 Boundary Setting Register	CB23	00h
000013h			
000014h	External Bus Control Register 2	EBC2	0000h
000015h			
000016h	Chip Selects 1 and 2 Boundary Setting Register	CB12	00h
000017h			
000018h	External Bus Control Register 1	EBC1	0000h
000019h			
00001Ah	Chip Selects 0 and 1 Boundary Setting Register	CB01	00h
00001Bh			
00001Ch	External Bus Control Register 0/Flash Memory Rewrite Bus	EBC0/FEBC0	0000h
00001Dh	Control Register 0		
00001Eh	Peripheral Bus Control Register	PBC	0504h
00001Fh			
000020h to 00005Fh			

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.2 SFR List (2)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.3 SFR List (3)**

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

X: Undefined

Blanks are reserved. No access is allowed.



**Table 4.4 SFR List (4)**

Address	Register	Symbol	Reset Value
0000B0h	Intelligent I/O Interrupt Enable Register 0	IIO0IE	00h
0000B1h	Intelligent I/O Interrupt Enable Register 1	IIO1IE	00h
0000B2h	Intelligent I/O Interrupt Enable Register 2	IIO2IE	00h
0000B3h	Intelligent I/O Interrupt Enable Register 3	IIO3IE	00h
0000B4h	Intelligent I/O Interrupt Enable Register 4	IIO4IE	00h
0000B5h	Intelligent I/O Interrupt Enable Register 5	IIO5IE	00h
0000B6h	Intelligent I/O Interrupt Enable Register 6	IIO6IE	00h
0000B7h	Intelligent I/O Interrupt Enable Register 7	IIO7IE	00h
0000B8h	Intelligent I/O Interrupt Enable Register 8	IIO8IE	00h
0000B9h	Intelligent I/O Interrupt Enable Register 9	IIO9IE	00h
0000BAh	Intelligent I/O Interrupt Enable Register 10	IIO10IE	00h
0000BBh	Intelligent I/O Interrupt Enable Register 11	IIO11IE	00h
0000BCh			
0000BDh			
0000BEh			
0000BFh			
0000C0h			
0000C1h			
0000C2h			
0000C3h			
0000C4h			
0000C5h			
0000C6h			
0000C7h			
0000C8h			
0000C9h			
0000CAh			
0000CBh			
0000CCh			
0000CDh			
0000CEh			
0000CFh			
0000D0h			
0000D1h			
0000D2h			
0000D3h			
0000D4h			
0000D5h			
0000D6h			
0000D7h			
0000D8h			
0000D9h			
0000DAh			
0000DBh			
0000DCh			
0000DDh	UART7 Transmit Interrupt Control Register	S7TIC	XXXX X00b
0000DEh			
0000DFh	UART8 Transmit Interrupt Control Register	S8TIC	XXXX X00b

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.5 SFR List (5)**

Address	Register	Symbol	Reset Value
0000E0h			
0000E1h			
0000E2h			
0000E3h			
0000E4h			
0000E5h			
0000E6h			
0000E7h			
0000E8h			
0000E9h			
0000EAh			
0000EBh			
0000ECh			
0000EDh			
0000EEh			
0000EFh			
0000F0h			
0000F1h			
0000F2h			
0000F3h			
0000F4h			
0000F5h			
0000F6h			
0000F7h			
0000F8h			
0000F9h			
000FAh			
000FBh			
000FCh			
000FDh	UART7 Receive Interrupt Control Register	S7RIC	XXXX X000b
000FEh			
000FFh	UART8 Receive Interrupt Control Register	S8RIC	XXXX X000b
000100h	Group 1 Time Measurement/Waveform Generation Register 0	G1TM0/G1PO0	XXXXh
000101h			
000102h	Group 1 Time Measurement/Waveform Generation Register 1	G1TM1/G1PO1	XXXXh
000103h			
000104h	Group 1 Time Measurement/Waveform Generation Register 2	G1TM2/G1PO2	XXXXh
000105h			
000106h	Group 1 Time Measurement/Waveform Generation Register 3	G1TM3/G1PO3	XXXXh
000107h			

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.6 SFR List (6)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.7 SFR List (7)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.8 SFR List (8)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.9 SFR List (9)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.10 SFR List (10)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.11 SFR List (11)**

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

X: Undefined

Blanks are reserved. No access is allowed.



**Table 4.12 SFR List (12)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.13 SFR List (13)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.14 SFR List (14)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.15 SFR List (15)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.16 SFR List (16)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.17 SFR List (17)**

Address	Register	Symbol	Reset Value
0003E0h			
0003E1h			
0003E2h			
0003E3h			
0003E4h			
0003E5h			
0003E6h			
0003E7h			
0003E8h			
0003E9h			
0003EAh			
0003EBh			
0003ECh			
0003EDh			
0003EEh			
0003EFh			
0003F0h	Pull-up Control Register 0	PUR0	0000 0000b
0003F1h	Pull-up Control Register 1	PUR1	XXXX 0000b
0003F2h	Pull-up Control Register 2	PUR2	0000 0000b
0003F3h	Pull-up Control Register 3	PUR3	XXXX XX00b
0003F4h			
0003F5h			
0003F6h			
0003F7h			
0003F8h			
0003F9h			
0003FAh			
0003FBh			
0003FCh			
0003FDh			
0003FEh			
0003FFh	Port Control Register	PCR	XXXX XXX0b

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.18 SFR List (18)**

Address	Register	Symbol	Reset Value
040000h	Flash Memory Control Register 0	FMR0	0X01 XX00b
040001h	Flash Memory Status Register 0	FMSR0	1000 0000b
040002h			
040003h			
040004h			
040005h			
040006h			
040007h			
040008h	Flash Register Protection Unlock Register 0	FPR0	00h
040009h	Flash Memory Control Register 1	FMR1	0000 0010b
04000Ah	Block Protect Bit Monitor Register 0	FBPM0	??? ????b (1)
04000Bh	Block Protect Bit Monitor Register 1	FBPM1	XXX? ????b (1)
04000Ch			
04000Dh			
04000Eh			
04000Fh			
040010h			
040011h			
040012h			
040013h			
040014h			
040015h			
040016h			
040017h			
040018h			
040019h			
04001Ah			
04001Bh			
04001Ch			
04001Dh			
04001Eh			
04001Fh			
040020h	PLL Control Register 0	PLC0	0000 0001b
040021h	PLL Control Register 1	PLC1	0001 1111b
040022h			
040023h			
040024h			
040025h			
040026h			
040027h			
040028h			
040029h			
04002Ah			
04002Bh			
04002Ch			
04002Dh			
04002Eh			
04002Fh			

X: Undefined

Blanks are reserved. No access is allowed.

Note:

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

**Table 4.19 SFR List (19)**

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

X: Undefined

Blanks are reserved. No access is allowed.

Note:

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



**Table 4.20 SFR List (20)**

Address	Register	Symbol	Reset Value
040094h			
040095h			
040096h			
040097h	Three-phase Output Buffer Control Register	IOBC	0XXX XXXXb
040098h	Input Function Select Register 0	IFS0	X000 0000b <sup>(1)</sup>
040099h			
04009Ah	Input Function Select Register 2	IFS2	0000 00X0b <sup>(2)</sup>
04009Bh	Input Function Select Register 3	IFS3	XXXX XX00b
04009Ch			
04009Dh			
04009Eh			
04009Fh	Input Function Select Register 7 <sup>(3)</sup>	IFS7	XXXX XX0Xb
0400A0h	Port P0_0 Function Select Register	P0_0S	0XXX X000b
0400A1h	Port P1_0 Function Select Register	P1_0S	XXXX X000b
0400A2h	Port P0_1 Function Select Register	P0_1S	0XXX X000b
0400A3h	Port P1_1 Function Select Register	P1_1S	XXXX X000b
0400A4h	Port P0_2 Function Select Register	P0_2S	0XXX X000b
0400A5h	Port P1_2 Function Select Register	P1_2S	XXXX X000b
0400A6h	Port P0_3 Function Select Register	P0_3S	0XXX X000b
0400A7h	Port P1_3 Function Select Register	P1_3S	XXXX X000b
0400A8h	Port P0_4 Function Select Register	P0_4S	0XXX X000b
0400A9h	Port P1_4 Function Select Register	P1_4S	XXXX X000b
0400AAh	Port P0_5 Function Select Register	P0_5S	0XXX X000b
0400ABh	Port P1_5 Function Select Register	P1_5S	XXXX X000b
0400ACh	Port P0_6 Function Select Register	P0_6S	0XXX X000b
0400ADh	Port P1_6 Function Select Register	P1_6S	XXXX X000b
0400AEh	Port P0_7 Function Select Register	P0_7S	0XXX X000b
0400AFh	Port P1_7 Function Select Register	P1_7S	XXXX X000b
0400B0h	Port P2_0 Function Select Register	P2_0S	0XXX X000b
0400B1h	Port P3_0 Function Select Register	P3_0S	XXXX X000b
0400B2h	Port P2_1 Function Select Register	P2_1S	0XXX X000b
0400B3h	Port P3_1 Function Select Register	P3_1S	XXXX X000b
0400B4h	Port P2_2 Function Select Register	P2_2S	0XXX X000b
0400B5h	Port P3_2 Function Select Register	P3_2S	XXXX X000b
0400B6h	Port P2_3 Function Select Register	P2_3S	0XXX X000b
0400B7h	Port P3_3 Function Select Register	P3_3S	XXXX X000b
0400B8h	Port P2_4 Function Select Register	P2_4S	0XXX X000b
0400B9h	Port P3_4 Function Select Register	P3_4S	XXXX X000b
0400BAh	Port P2_5 Function Select Register	P2_5S	0XXX X000b
0400BBh	Port P3_5 Function Select Register	P3_5S	XXXX X000b
0400BCh	Port P2_6 Function Select Register	P2_6S	0XXX X000b
0400BDh	Port P3_6 Function Select Register	P3_6S	XXXX X000b
0400BEh	Port P2_7 Function Select Register	P2_7S	0XXX X000b
0400BFh	Port P3_7 Function Select Register	P3_7S	XXXX X000b

X: Undefined

Blanks are reserved. No access is allowed.

**Notes:**

1. The reset value is 0000 0000b in the 64-pin package.
2. The reset value is 0000 000Xb in the 64-pin package.
3. This register is provided for the 64-pin package only. No access is allowed in the 100-pin package.

**Table 4.21 SFR List (21)**

Address	Register	Symbol	Reset Value
0400C0h	Port P4_0 Function Select Register	P4_0S	XXXX X000b
0400C1h	Port P5_0 Function Select Register	P5_0S	XXXX X000b
0400C2h	Port P4_1 Function Select Register	P4_1S	XXXX X000b
0400C3h	Port P5_1 Function Select Register	P5_1S	XXXX X000b
0400C4h	Port P4_2 Function Select Register	P4_2S	XXXX X000b
0400C5h	Port P5_2 Function Select Register	P5_2S	XXXX X000b
0400C6h	Port P4_3 Function Select Register	P4_3S	XXXX X000b
0400C7h	Port P5_3 Function Select Register	P5_3S	XXXX X000b
0400C8h	Port P4_4 Function Select Register	P4_4S	XXXX X000b
0400C9h	Port P5_4 Function Select Register	P5_4S	XXXX X000b
0400CAh	Port P4_5 Function Select Register	P4_5S	XXXX X000b
0400CBh	Port P5_5 Function Select Register	P5_5S	XXXX X000b
0400CCh	Port P4_6 Function Select Register	P4_6S	XXXX X000b
0400CDh	Port P5_6 Function Select Register	P5_6S	XXXX X000b
0400CEh	Port P4_7 Function Select Register	P4_7S	XXXX X000b
0400CFh	Port P5_7 Function Select Register	P5_7S	XXXX X000b
0400D0h	Port P6_0 Function Select Register	P6_0S	XXXX X000b
0400D1h	Port P7_0 Function Select Register	P7_0S	XXXX X000b
0400D2h	Port P6_1 Function Select Register	P6_1S	XXXX X000b
0400D3h	Port P7_1 Function Select Register	P7_1S	XXXX X000b
0400D4h	Port P6_2 Function Select Register	P6_2S	XXXX X000b
0400D5h	Port P7_2 Function Select Register	P7_2S	XXXX X000b
0400D6h	Port P6_3 Function Select Register	P6_3S	XXXX X000b
0400D7h	Port P7_3 Function Select Register	P7_3S	XXXX X000b
0400D8h	Port P6_4 Function Select Register	P6_4S	XXXX X000b
0400D9h	Port P7_4 Function Select Register	P7_4S	XXXX X000b
0400DAh	Port P6_5 Function Select Register	P6_5S	XXXX X000b
0400DBh	Port P7_5 Function Select Register	P7_5S	XXXX X000b
0400DCh	Port P6_6 Function Select Register	P6_6S	XXXX X000b
0400DDh	Port P7_6 Function Select Register	P7_6S	XXXX X000b
0400DEh	Port P6_7 Function Select Register	P6_7S	XXXX X000b
0400DFh	Port P7_7 Function Select Register	P7_7S	XXXX X000b
0400E0h	Port P8_0 Function Select Register	P8_0S	XXXX X000b
0400E1h			
0400E2h	Port P8_1 Function Select Register	P8_1S	XXXX X000b
0400E3h			
0400E4h	Port P8_2 Function Select Register	P8_2S	XXXX X000b
0400E5h			
0400E6h	Port P8_3 Function Select Register	P8_3S	XXXX X000b
0400E7h	Port P9_3 Function Select Register	P9_3S	0XXX X000b
0400E8h	Port P8_4 Function Select Register	P8_4S	XXXX X000b
0400E9h	Port P9_4 Function Select Register	P9_4S	0XXX X000b
0400EAh			
0400EBh	Port P9_5 Function Select Register	P9_5S	0XXX X000b
0400ECh	Port P8_6 Function Select Register	P8_6S	XXXX X000b
0400EDh	Port P9_6 Function Select Register	P9_6S	0XXX X000b
0400EEh	Port P8_7 Function Select Register	P8_7S	XXXX X000b
0400EFh	Port P9_7 Function Select Register	P9_7S	XXXX X000b

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.22 SFR List (22)**

Address	Register	Symbol	Reset Value
0400F0h	Port P10_0 Function Select Register	P10_0S	0XXX X000b
0400F1h			
0400F2h	Port P10_1 Function Select Register	P10_1S	0XXX X000b
0400F3h			
0400F4h	Port P10_2 Function Select Register	P10_2S	0XXX X000b
0400F5h			
0400F6h	Port P10_3 Function Select Register	P10_3S	0XXX X000b
0400F7h			
0400F8h	Port P10_4 Function Select Register	P10_4S	0XXX X000b
0400F9h			
0400FAh	Port P10_5 Function Select Register	P10_5S	0XXX X000b
0400FBh			
0400FCh	Port P10_6 Function Select Register	P10_6S	0XXX X000b
0400FDh			
0400FEh	Port P10_7 Function Select Register	P10_7S	0XXX X000b
0400FFh			
040100h			
040101h			
040102h			
040103h			
040104h			
040105h			
040106h			
040107h			
040108h			
040109h			
04010Ah			
04010Bh			
04010Ch			
04010Dh			
04010Eh			
04010Fh			
040110h			
040111h			
040112h			
040113h			
040114h			
040115h			
040116h			
040117h			
040118h			
040119h			
04011Ah			
04011Bh			
04011Ch			
04011Dh			
04011Eh			
04011Fh			

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.23 SFR List (23)**

Address	Register	Symbol	Reset Value
040120h to 04403Fh			
044040h			
044041h			
044042h			
044043h			
044044h			
044045h			
044046h			
044047h			
044048h			
044049h			
04404Ah			
04404Bh			
04404Ch			
04404Dh			
04404Eh	Watchdog Timer Start Register	WDTS	XXXX XXXXb
04404Fh	Watchdog Timer Control Register	WDC	000X XXXXb
044050h			
044051h			
044052h			
044053h			
044054h			
044055h			
044056h			
044057h			
044058h			
044059h			
04405Ah			
04405Bh			
04405Ch			
04405Dh			
04405Eh			
04405Fh	Protect Register 2	PRCR2	0XXX XXXXb

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.24 SFR List (24)**

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

X: Undefined

Blanks are reserved. No access is allowed.

## 5. Resets

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

### 5.1 Hardware Reset

A hardware reset is generated using the  $\overline{\text{RESET}}$  pin. When a low signal is applied to the  $\overline{\text{RESET}}$  pin while the supply voltage satisfies the operating conditions described in 26. "Electrical Characteristics", the pins are reset (refer to Table 5.1 for details). Also, oscillators are reset, and the main clock starts oscillating.

The CPU and SFRs are reset by a low-to-high transition on the  $\overline{\text{RESET}}$  pin. Then, the CPU starts executing the program from the address indicated by the reset vector. Internal RAM is not affected by a hardware reset. However, if a hardware reset occurs during a write operation to the internal RAM, the value is undefined.

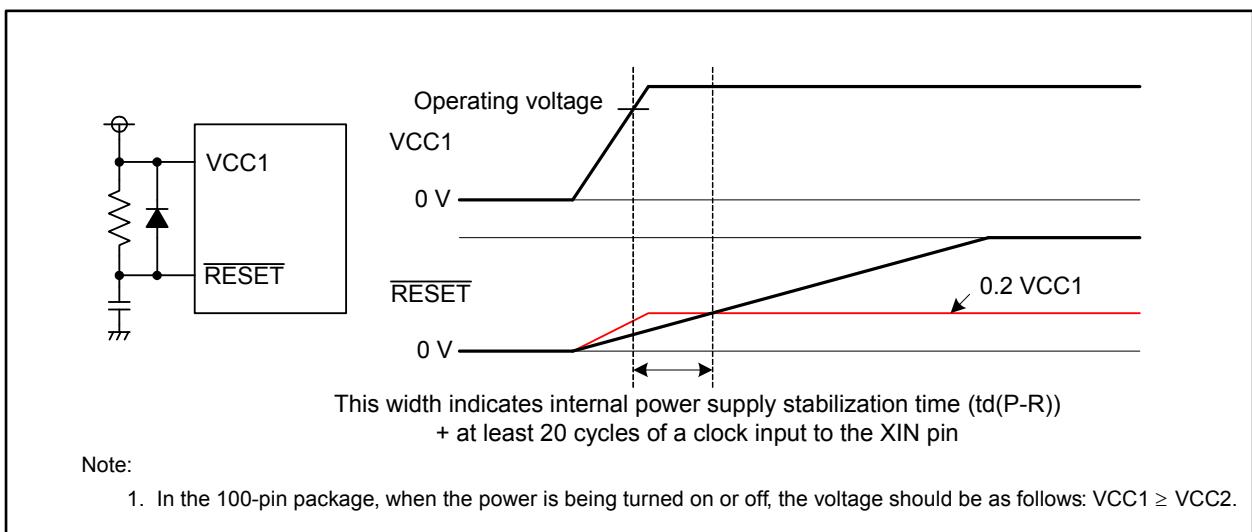
Figure 5.1 shows an example of the reset circuit. Figure 5.2 shows the reset sequence. Table 5.1 lists pin states while the  $\overline{\text{RESET}}$  pin is held low. Figure 5.3 shows CPU register states after a reset. Refer to 4. "Special Function Registers (SFRs)" for details on the states of SFRs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

#### A. Reset when the supply voltage is stable

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

#### B. Reset when turning on the power

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



**Figure 5.1** Reset Circuitry

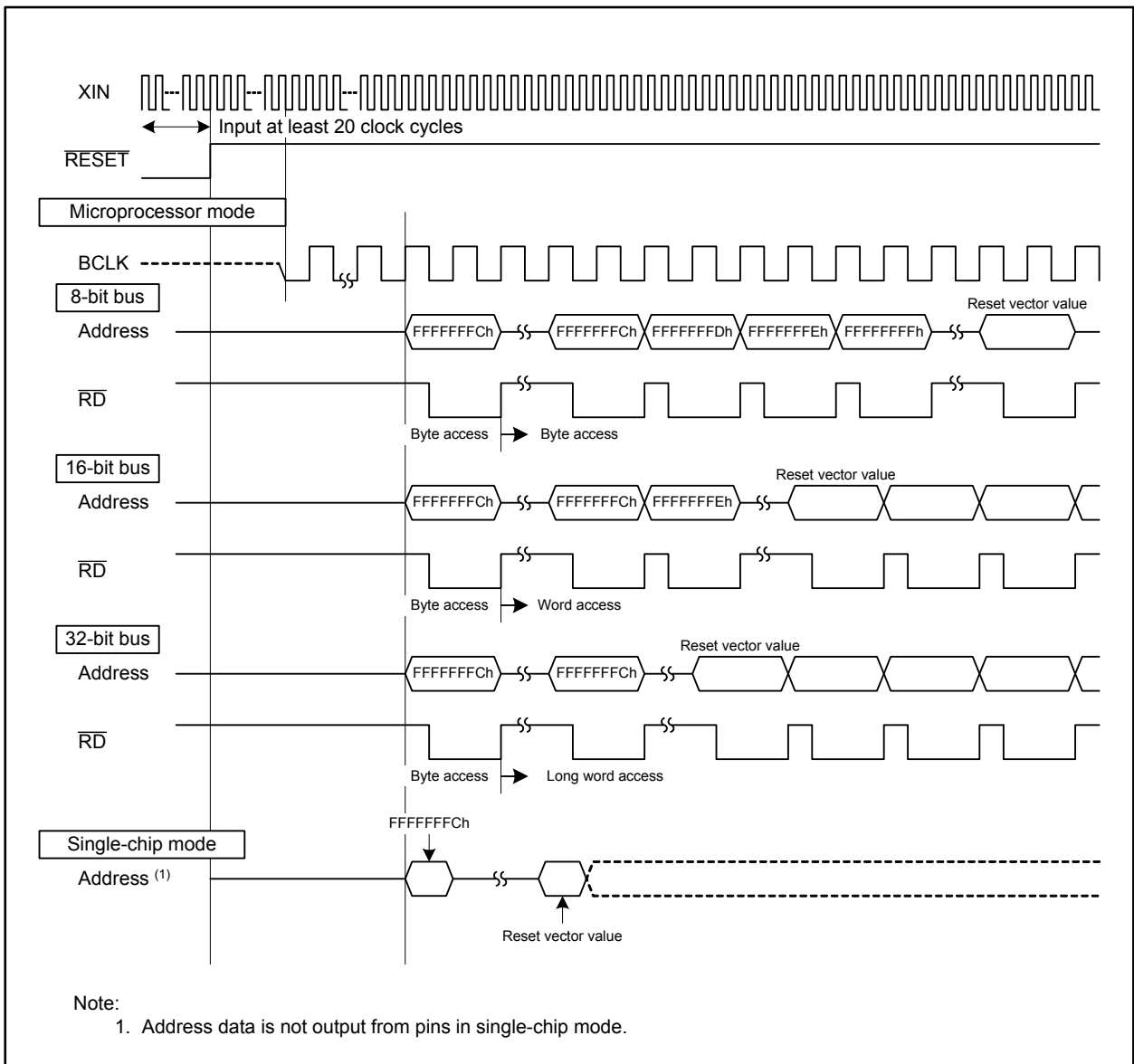


Figure 5.2 Reset Sequence

**Table 5.1 Pin States while  $\overline{\text{RESET}}$  Pin is Held Low (1)**

Pin Name	Pin States	
	CVSS = VSS	CVSS = VCC1 (2)
P0	Input port (high-impedance)	Inputs data
P1	Input port (high-impedance)	Input port (high-impedance)
P2, P3	Input port (high-impedance)	Output addresses (undefined)
P4_0 to P4_6 (3)	Input port (high-impedance)	Output addresses (undefined)
P4_7 (3)	Input port (high-impedance)	Outputs the $\overline{\text{CS0}}$ signal (high)
P5_0 (3)	Input port (high-impedance)	Outputs the $\overline{\text{WR}}$ signal (high)
P5_1 (3)	Input port (high-impedance)	Outputs the $\overline{\text{BC1}}$ signal (undefined)
P5_2 (3)	Input port (high-impedance)	Outputs the $\overline{\text{RD}}$ signal (high)
P5_3 (3)	Input port (high-impedance)	Outputs BCLK (4)
P5_4 (3)	Input port (high-impedance)	Outputs the $\overline{\text{HLDA}}$ signal (output signal depends on an input signal to the $\overline{\text{HOLD}}$ pin) (4)
P5_5 (3)	Input port (high-impedance)	Inputs the $\overline{\text{HOLD}}$ signal (high-impedance)
P5_6 (3)	Input port (high-impedance)	Outputs the $\overline{\text{CS2}}$ signal (high)
P5_7 (3)	Input port (high-impedance)	Inputs the $\overline{\text{RDY}}$ signal (high-impedance)
P6 to P10	Input port (high-impedance)	Input port (high-impedance)

## Notes:

- Whether a pull-up resistor is enabled or not is undefined until the internal voltage is stabilized.
- Do not use this setting in the 64-pin package.
- Ports P4 and P5 are only available in the 100-pin package.
- State after power is on and the internal voltage has stabilized. It is undefined until the internal voltage is stabilized



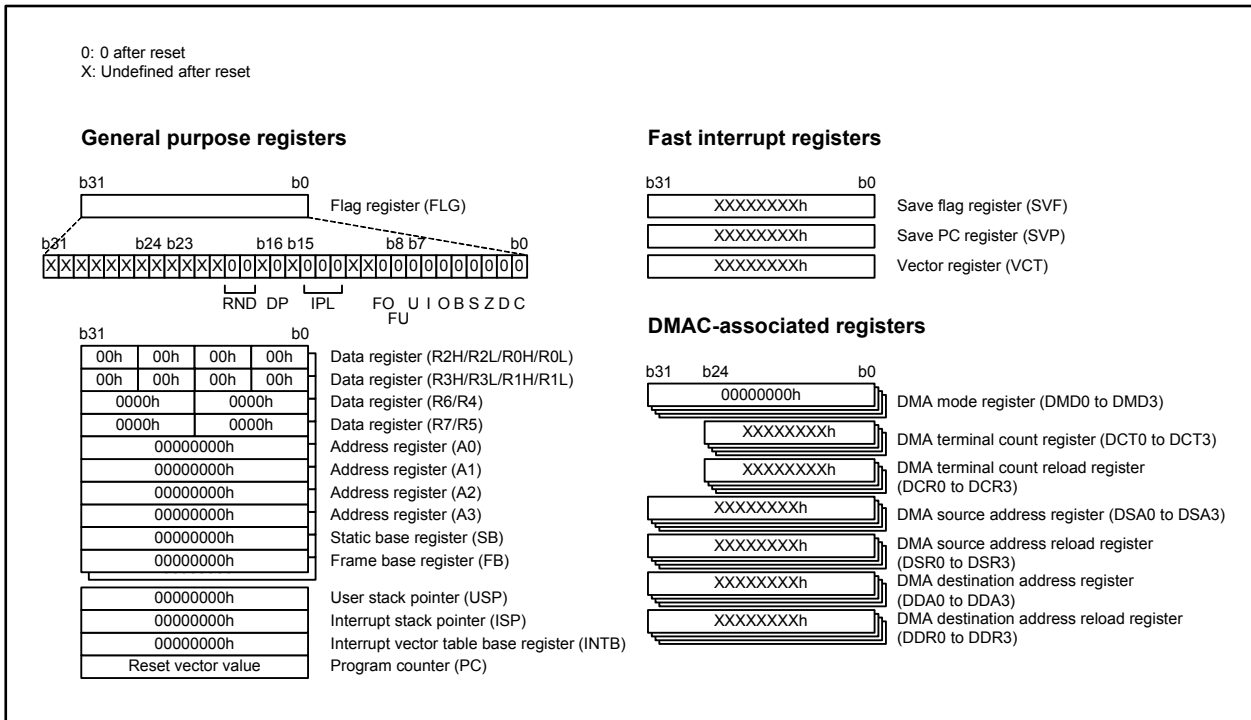


Figure 5.3 CPU Registers after Reset

## 5.2 Software Reset

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

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

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

## 5.3 Watchdog Timer Reset

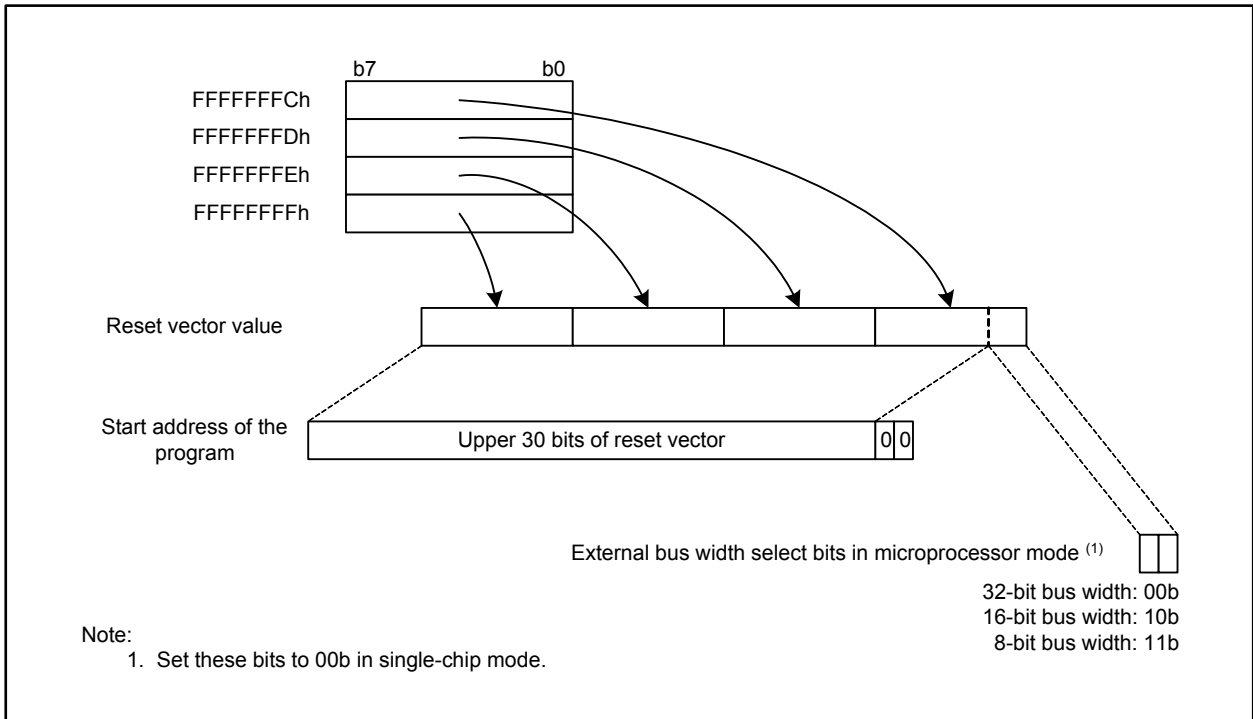
The CPU, SFRs, and pins are reset when the watchdog timer underflows while the CM06 bit in the CM0 register is 1 (reset when watchdog timer underflows). Then, the CPU executes the program from the address indicated by the reset vector.

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

## 5.4 Reset Vector

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

The start address of a program consists of the upper 30 bits of the reset vector and 00b as lower 2 bits. The lower 2 bits of the reset vector are bits to select the external bus width in microprocessor mode. Therefore, the start address of a program requires 4-byte alignment so that the lower 2 bits are 00b. In single-chip mode, set the external bus width select bits to 00b.



**Figure 5.4 Reset Vector Configuration**

## 6. Power Management

### 6.1 Voltage Regulators for Internal Logic

The supply voltage for internal logic is generated by reducing the input voltage from the VCC1 pin with the voltage regulators. Figure 6.1 shows a block diagram of the voltage regulators for internal logic, and Figure 6.2 shows the VRCCR register.

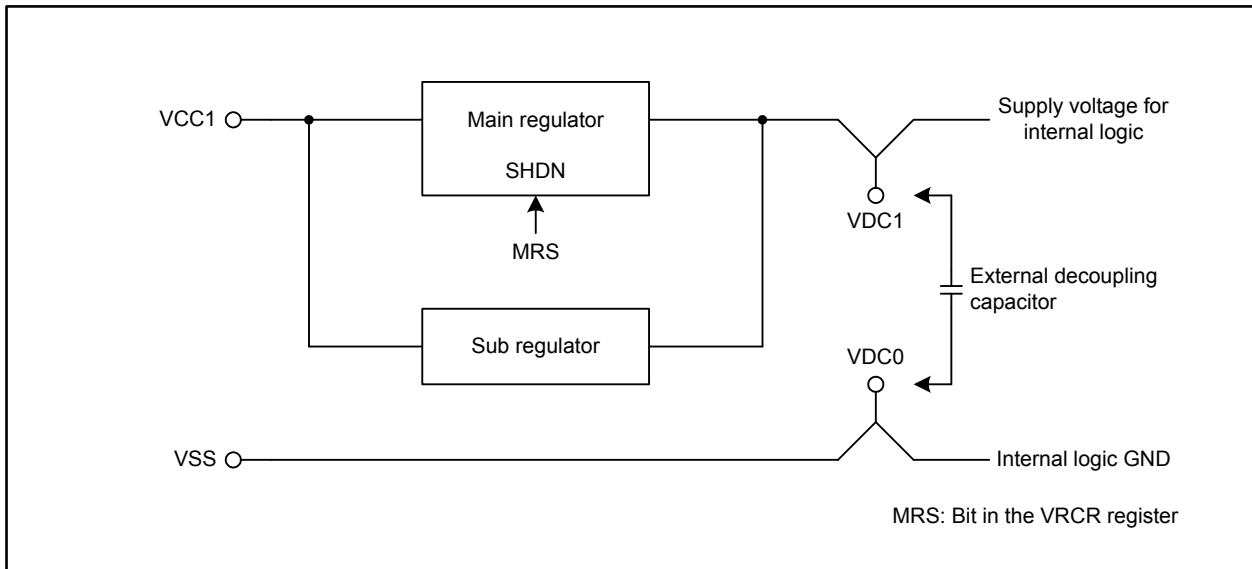


Figure 6.1 Block Diagram of Voltage Regulators for Internal Logic

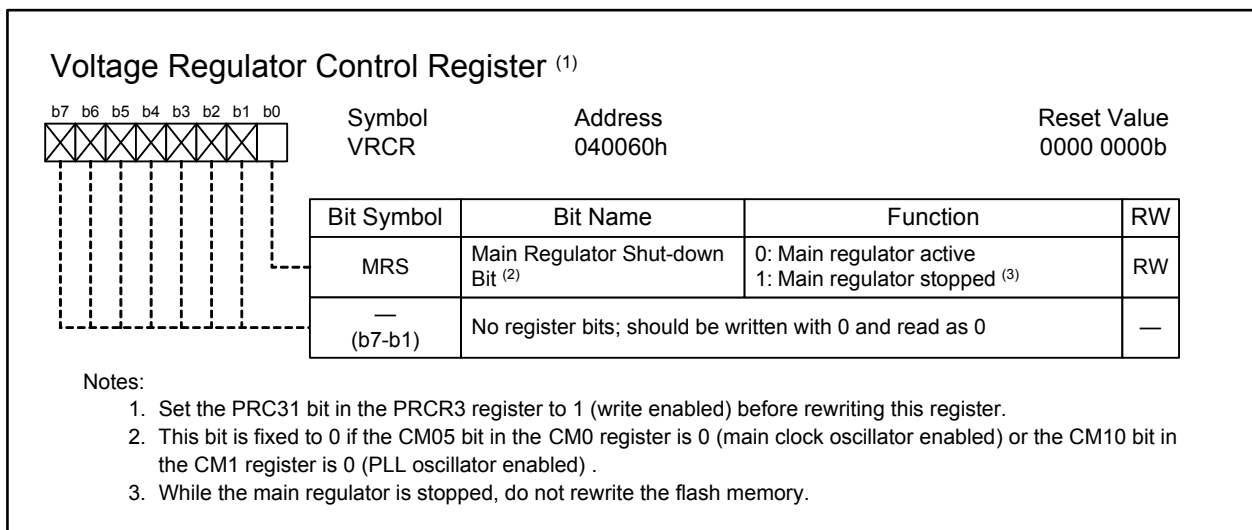


Figure 6.2 VRCCR Register

### 6.1.1 Decoupling Capacitor

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

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

**Table 6.1 Recommended Capacitor Specifications**

Applicable standard		Temperature Characteristics		Rated Voltage (V)	Nominal Capacitance ( $\mu$ F)	Capacitance Tolerance (%)
		Operating temperature range ( $^{\circ}$ C)	Capacitance change (%)			
B	JIS	-25 to 85	$\pm$ 10	6.3 or higher	4.7	$\pm$ 20 or better
R	JIS	-55 to 125	$\pm$ 15	6.3 or higher	4.7	$\pm$ 20 or better
X5R	EIA	-55 to 85	$\pm$ 15	6.3 or higher	4.7	$\pm$ 20 or better
X7R	EIA	-55 to 125	$\pm$ 15	6.3 or higher	4.7	$\pm$ 20 or better
X8R	EIA	-55 to 150	$\pm$ 15	6.3 or higher	4.7	$\pm$ 20 or better
X6S	EIA	-55 to 105	$\pm$ 22	6.3 or higher	4.7	$\pm$ 20 or better
X7S	EIA	-55 to 125	$\pm$ 22	6.3 or higher	4.7	$\pm$ 20 or better

## 6.2 Low Voltage Detector

The low voltage detector monitors the supply voltage input to the VCC1 pin.

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

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

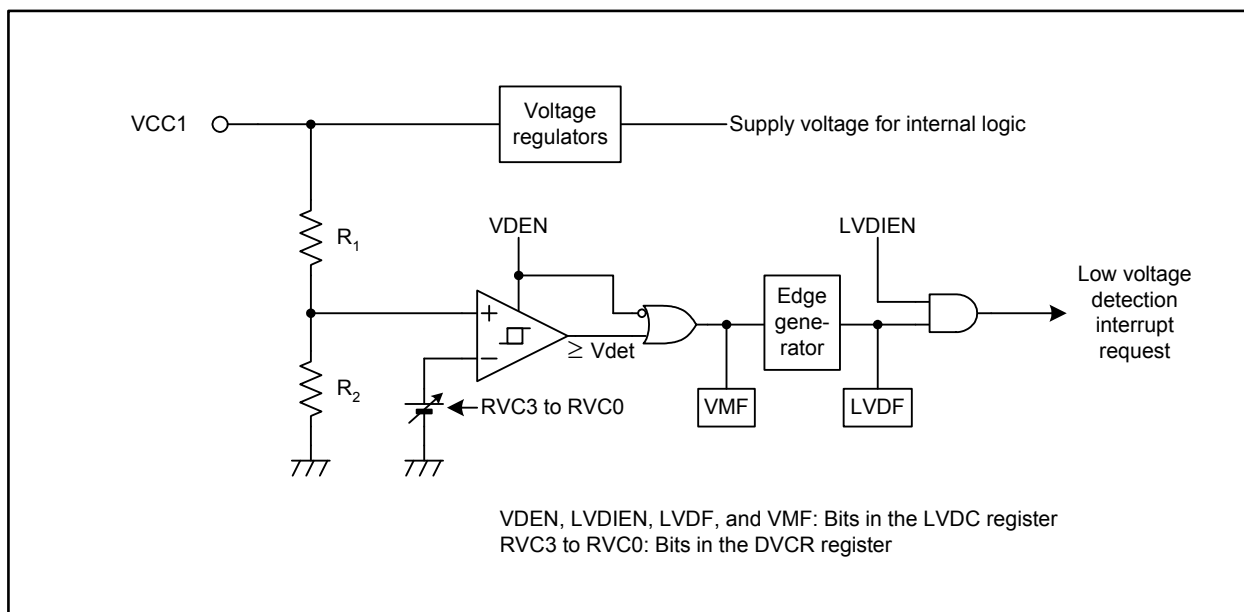
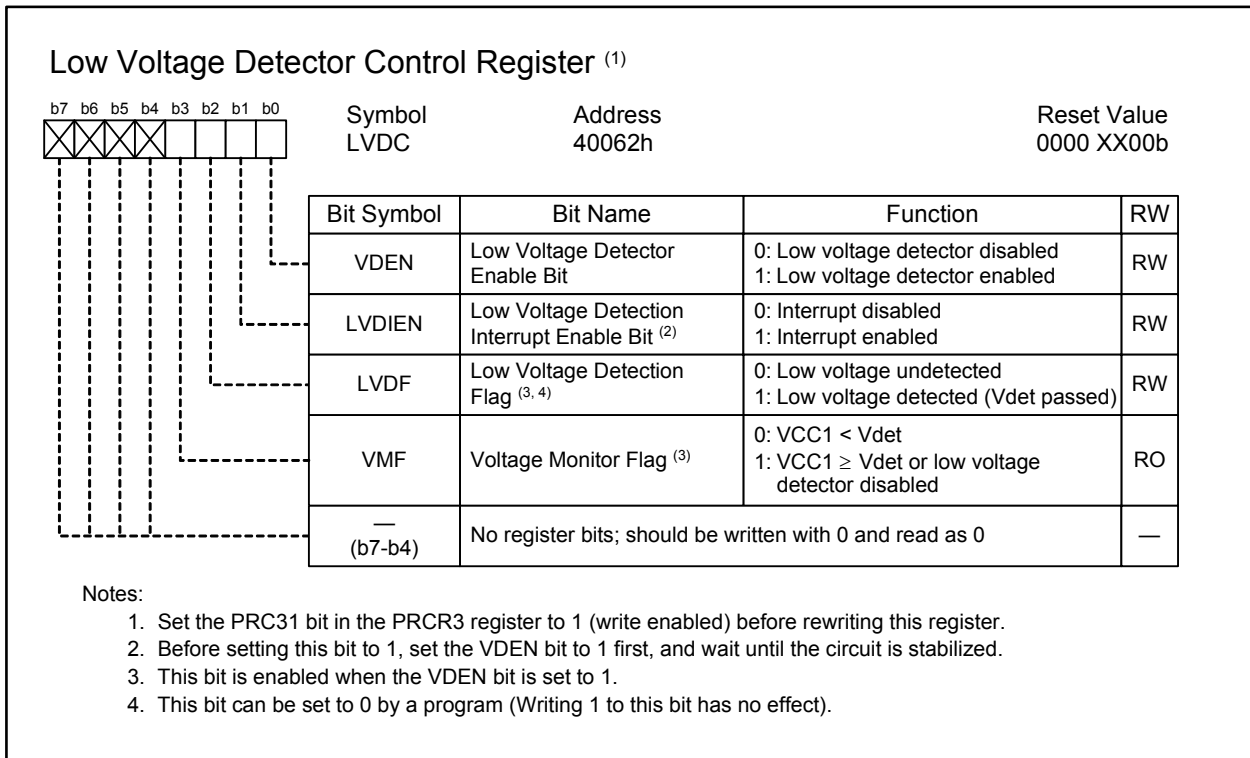


Figure 6.3 Low Voltage Detector Block Diagram



**Figure 6.4 LVDC Register**

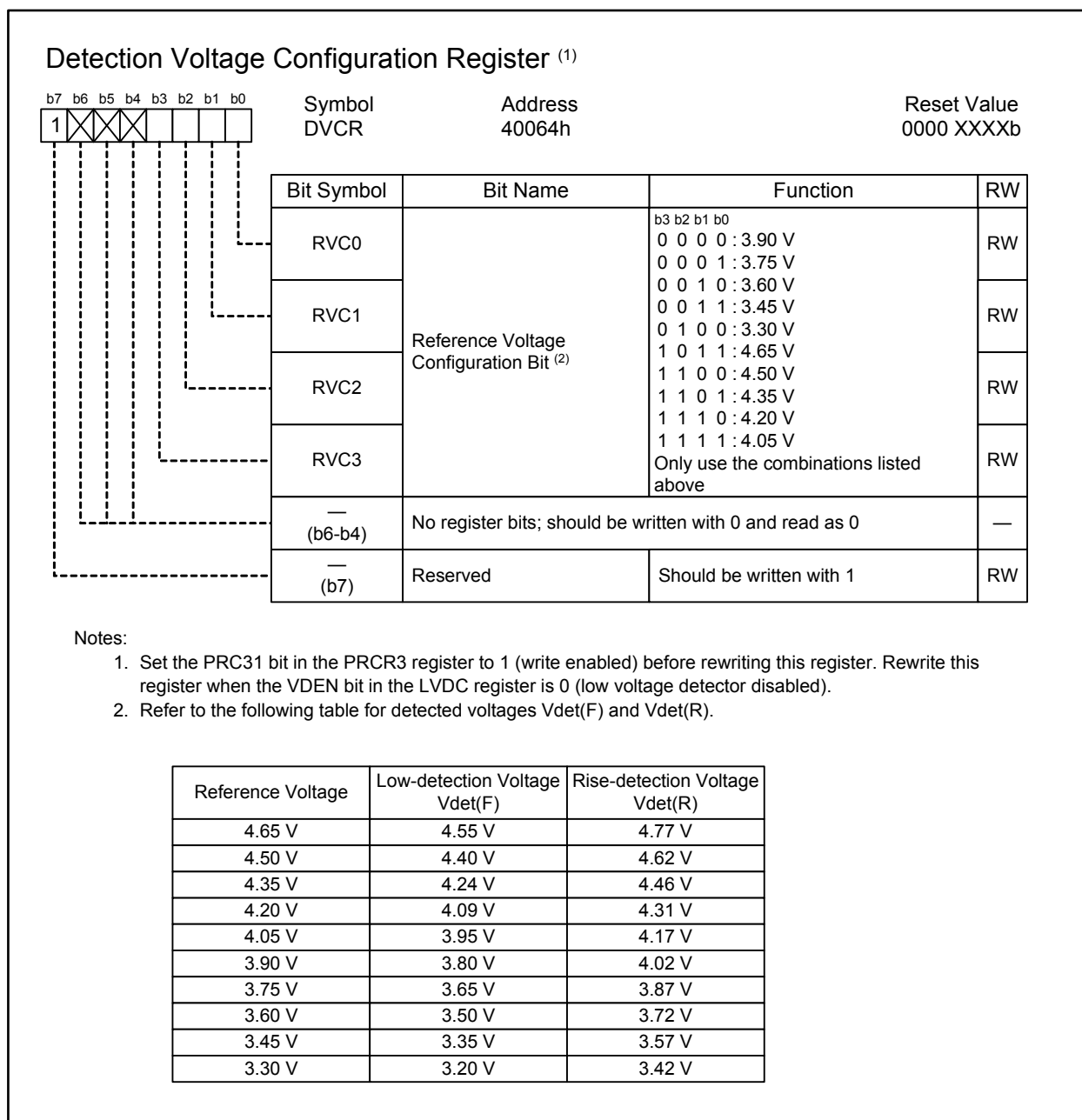


Figure 6.5 DVCR Register

### 6.2.1 Operational State of Low Voltage Detector

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

When the input voltage to the VCC1 pin drops below  $V_{det(F)}$ , the VMF bit becomes 0 ( $V_{CC1} < V_{det}$ ) and the LVDF bit becomes 1 (low voltage detected ( $V_{det}$  passed)). At this point an interrupt request is generated when the LVDIEN bit is 1 (low voltage detection interrupt enabled). Set the LVDF bit to 0 (low voltage undetected) by a program.

When the voltage rises to or above  $V_{det(R)}$  again, the VMF bit becomes 1 ( $V_{CC1} \geq V_{det}$ ) and the LVDF bit becomes 1. At this point an interrupt request is generated when the LVDIEN bit is 1.

Figure 6.6 shows the operation of the low voltage detector.

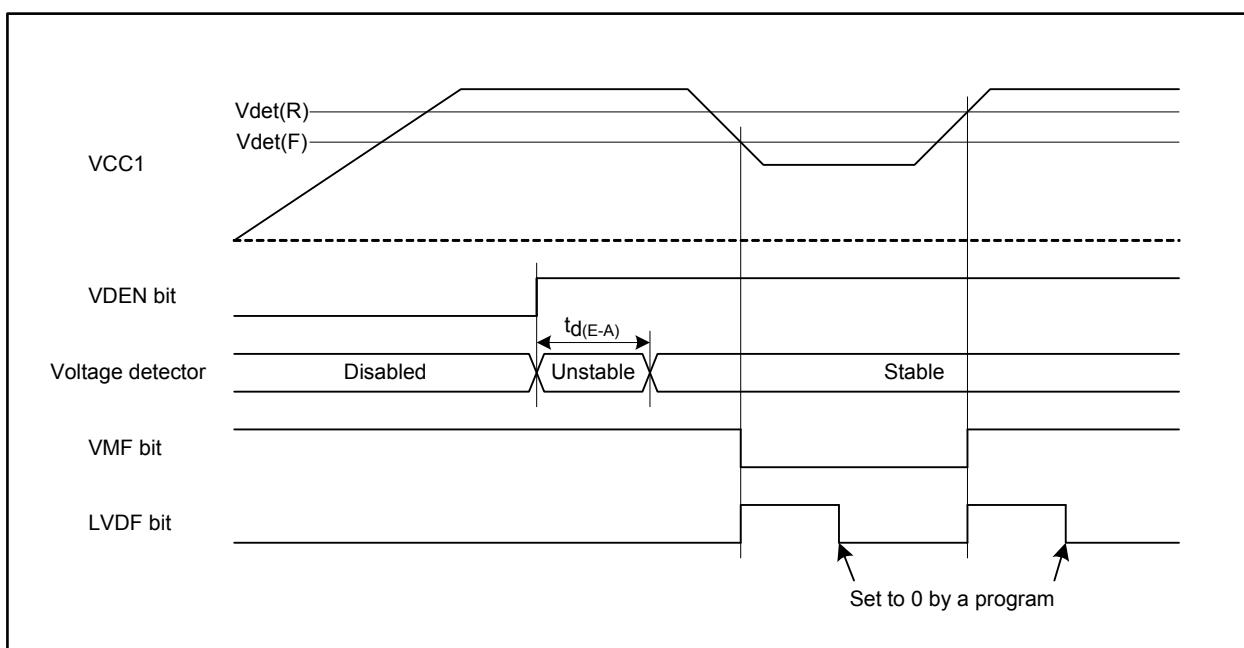


Figure 6.6 Low Voltage Detector Operation

### 6.2.2 Low Voltage Detection Interrupt

A low voltage detection interrupt request is generated when the input voltage at the VCC1 pin rises to or above the  $V_{det(R)}$  level, or falls below the  $V_{det(F)}$  level while the LVDIEN bit in the LVDC register is 1 (low voltage detection interrupt enabled).

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

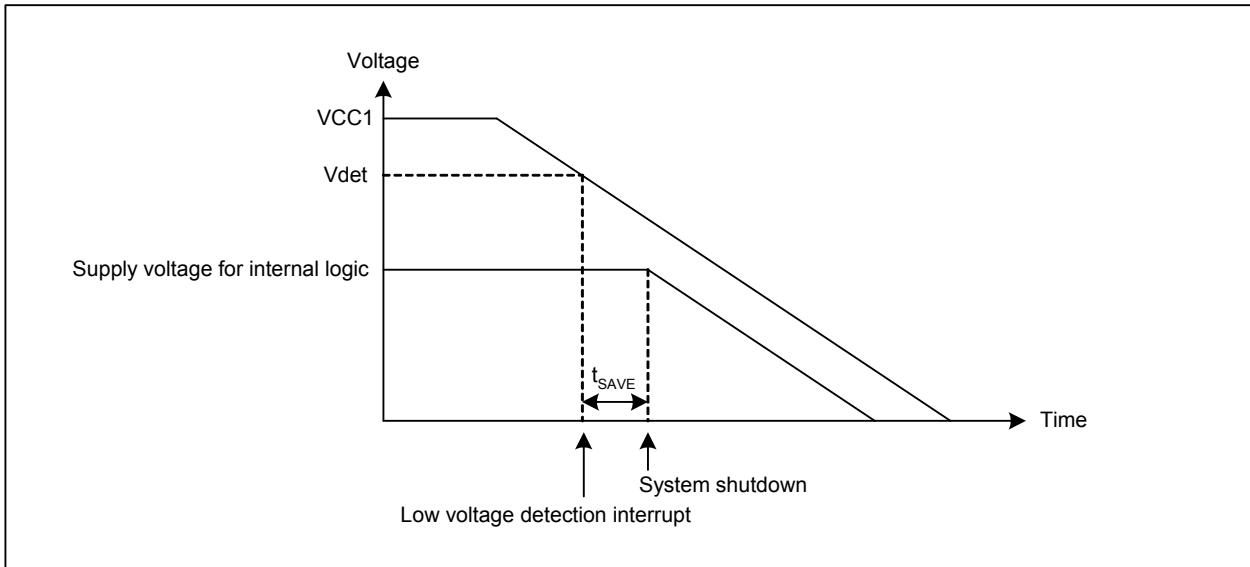
The LVDF bit becomes 1 when the input voltage at the VCC1 pin passes the  $V_{det(R)}$  level or  $V_{det(F)}$  level. When the LVDF bit changes from 0 to 1, a low voltage detection interrupt request is generated. Set this bit to 0 (low voltage undetected) by a program.



### 6.2.3 Application Example of the Low Voltage Detector

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

The supply voltage for internal logic is generated by reducing the input voltage from the VCC1 pin with the voltage regulators. When the input voltage begins to fall, the internal voltage remains steady. However, as the VCC1 input voltage continues to fall, the supply voltage for the internal logic also begins to fall, which may affect MCU operation. Consequently, the system can be safely shut down between when the VCC1 input voltage begins to fall and when the supply voltage for internal logic begins to fall. The low voltage detection interrupt can be applied to detect the falling input voltage.



**Figure 6.7** Example of the Low Voltage Detection Interrupt

## 7. Processor Mode

### 7.1 Types of Processor Modes

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

**Table 7.1 Processor Mode Characteristics**

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

Note:

1. Refer to 9. "Bus" for details.

The R32C/111 Group supports two standard processor modes: single-chip mode and memory expansion mode in the 100-pin package. Microprocessor mode is optional. Contact a Renesas Electronics sales office to use this mode. In the 64-pin package, only single-chip mode is supported.

### 7.2 Processor Mode Setting

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

**Table 7.2 Processor Mode after Hardware Reset**

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

Note:

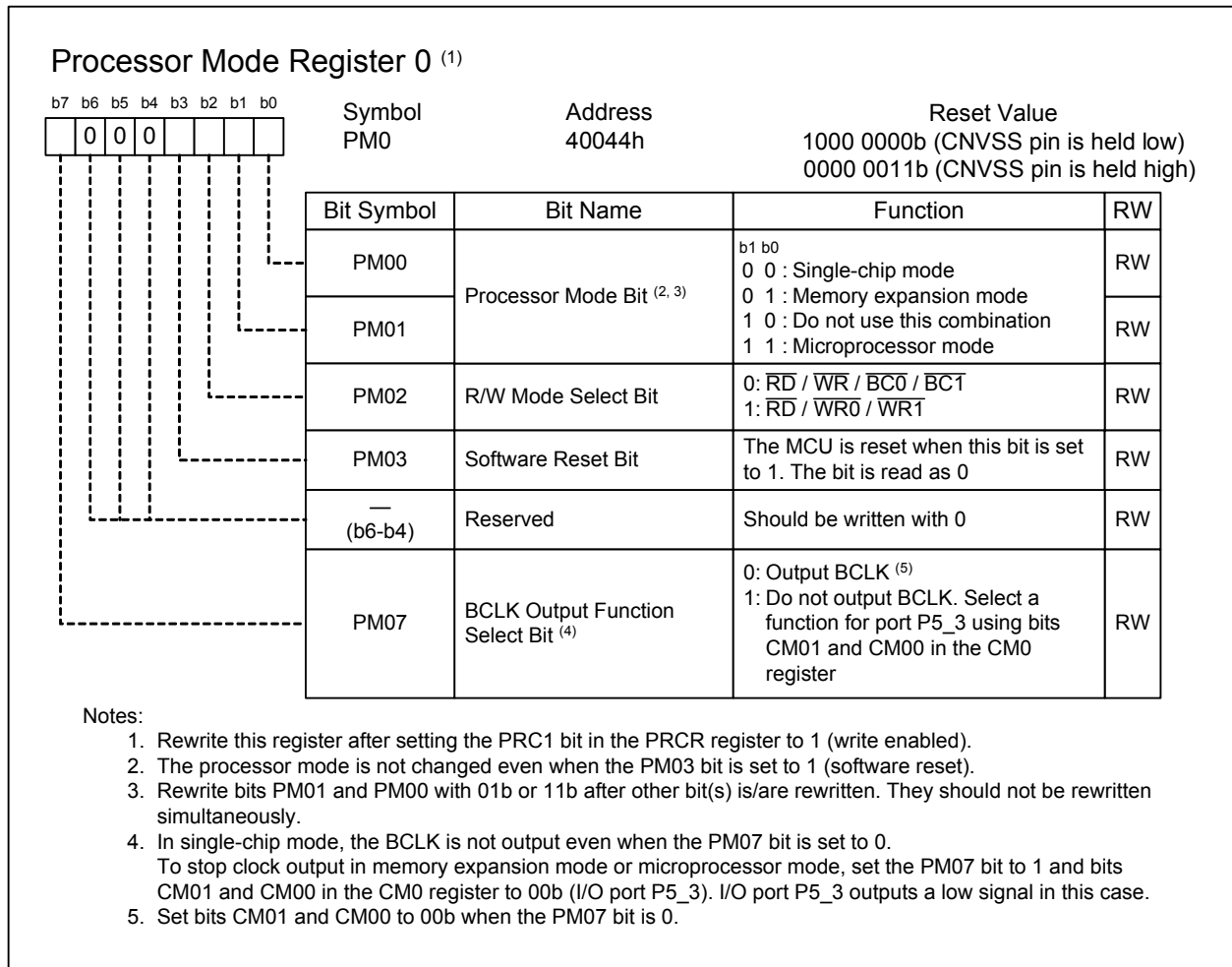
1. The CNVSS pin should be connected to VCC1 or VSS via a resistor.

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

Notes on changing processor mode are as follows:

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

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



**Figure 7.1 PM0 Register**

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

Notes:

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

**Figure 7.2 Memory Map of Each Processor Mode**

## 8. Clock Generator

### 8.1 Clock Generator Types

The clock generator consists of four circuits:

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

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

**Table 8.1 Clock Generator Specifications**

Item	Main Clock Oscillator	Sub Clock Oscillator	PLL Frequency Synthesizer	On-chip Oscillator
Used as	<ul style="list-style-type: none"> <li>• PLL reference clock source</li> <li>• Peripheral clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Clock source for timers A and B</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Clock source for timers A and B</li> </ul>
Clock frequency	4 to 16 MHz	32.768 kHz	$f_{SO(PLL)}$ or $f_{(PLL)}$	Approx. 125 kHz
Connectable oscillators or additional circuits	Ceramic resonator Crystal oscillator	Crystal oscillator	—	—
Pins for oscillators or additional circuits	XIN, XOUT	XCIN, XCOU	—	—
Oscillator stop/restart function	Available	Available	Available	Available
Oscillator state after a reset	Running	Stopped	Running	Stopped
Note	Externally generated clock can be input	Externally generated clock can be input	When the main clock oscillator stops running, the PLL frequency synthesizer oscillates at its own frequency of $f_{SO(PLL)}$	

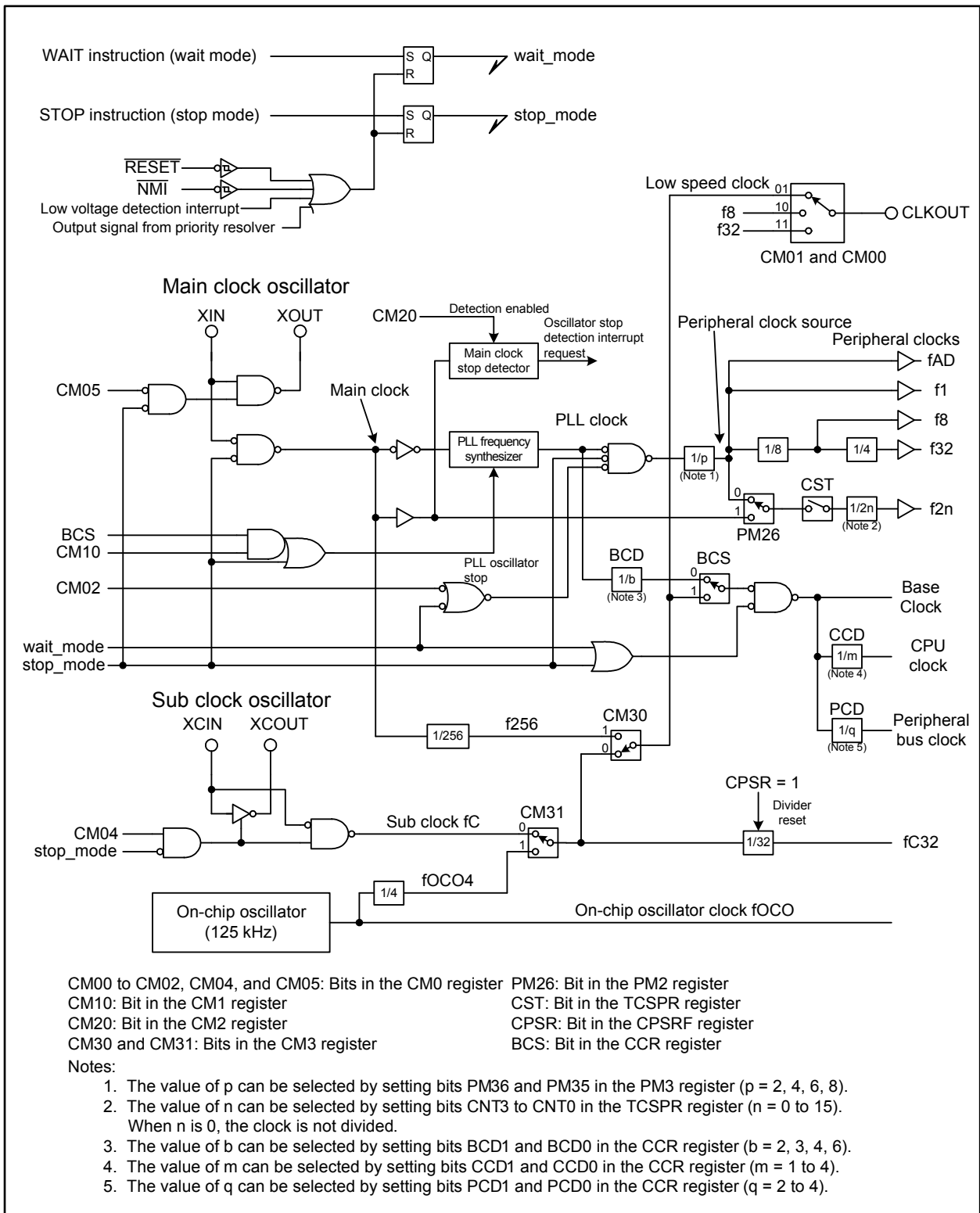
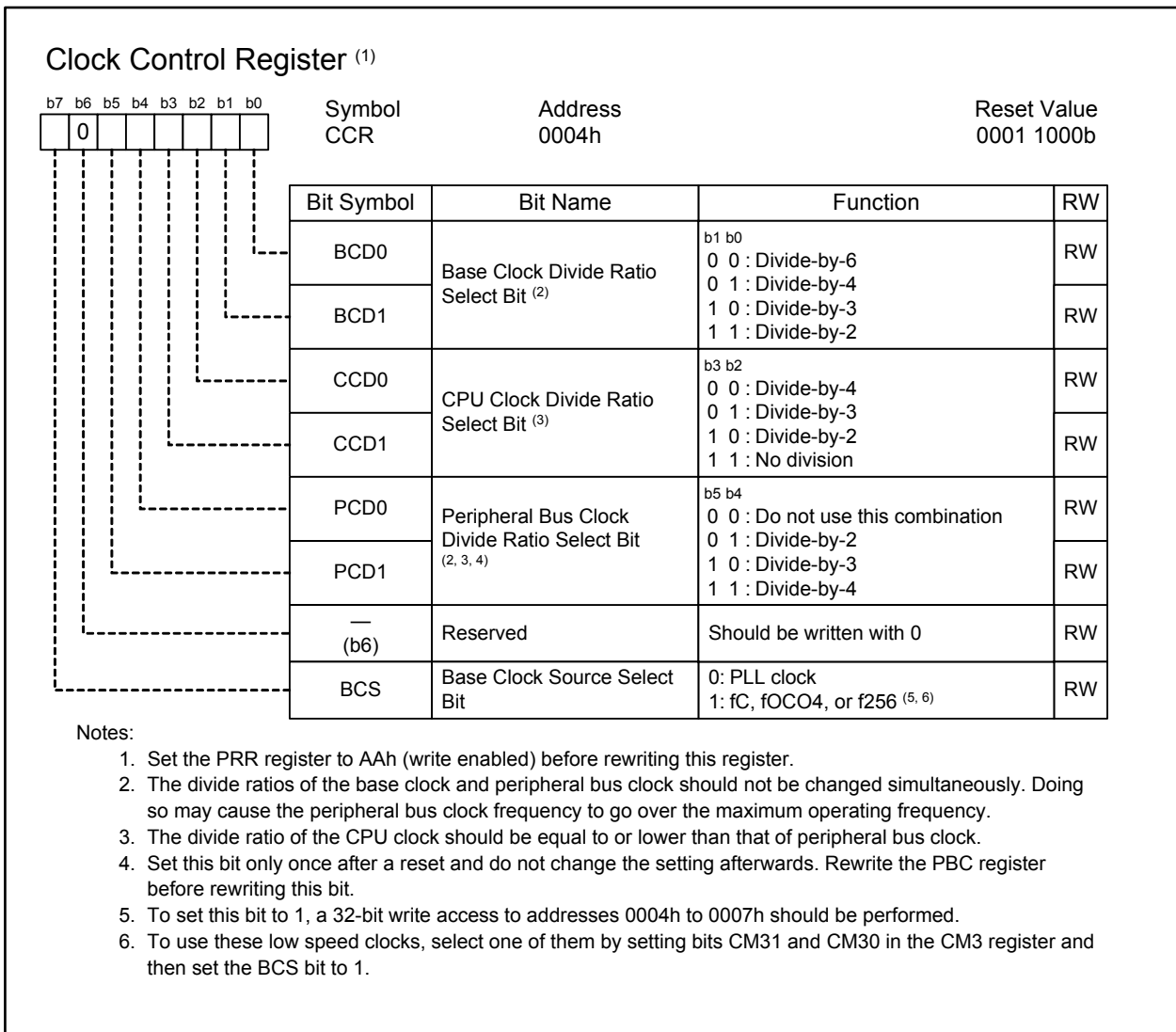
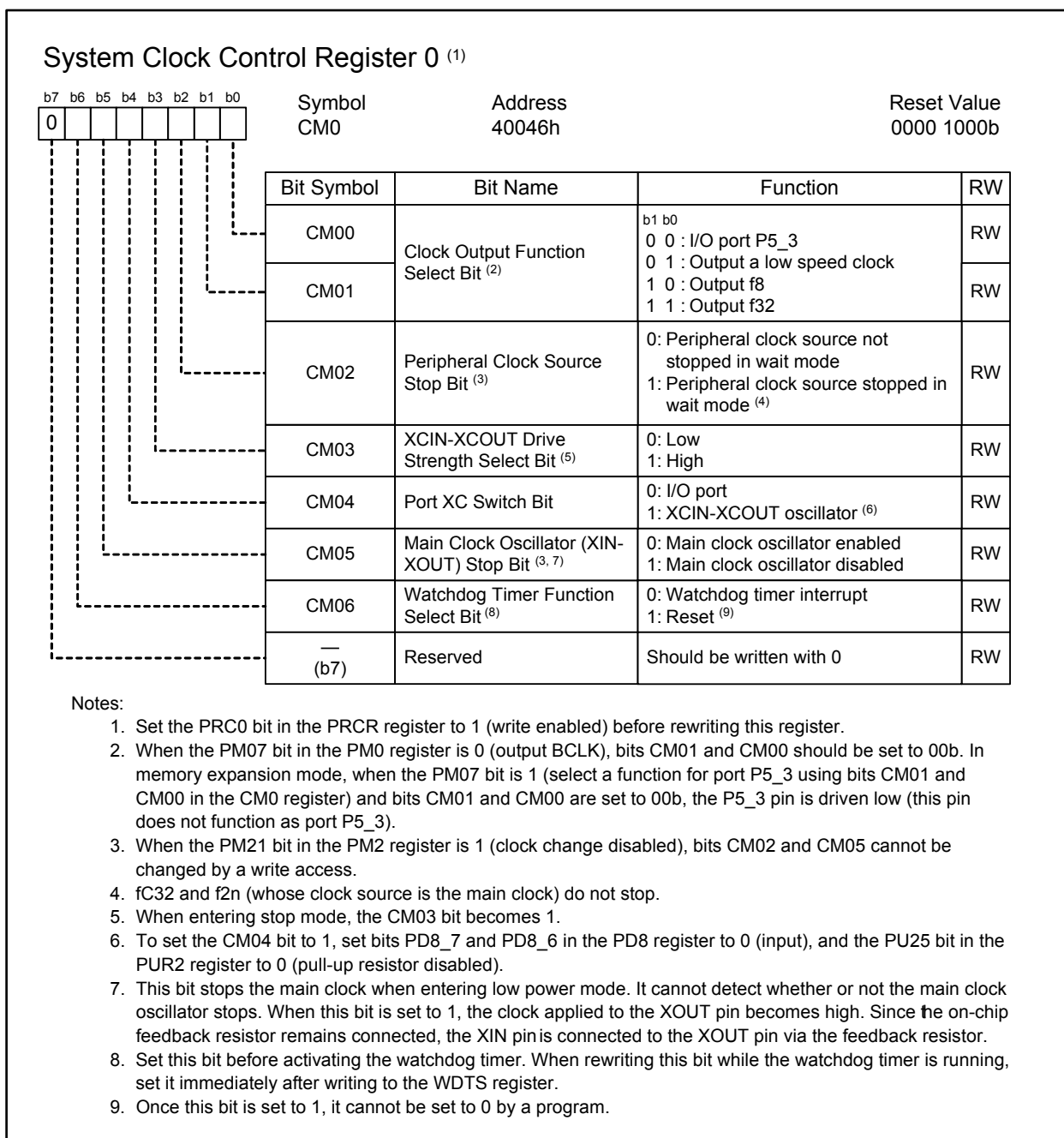


Figure 8.1 Clock Generation Circuitry

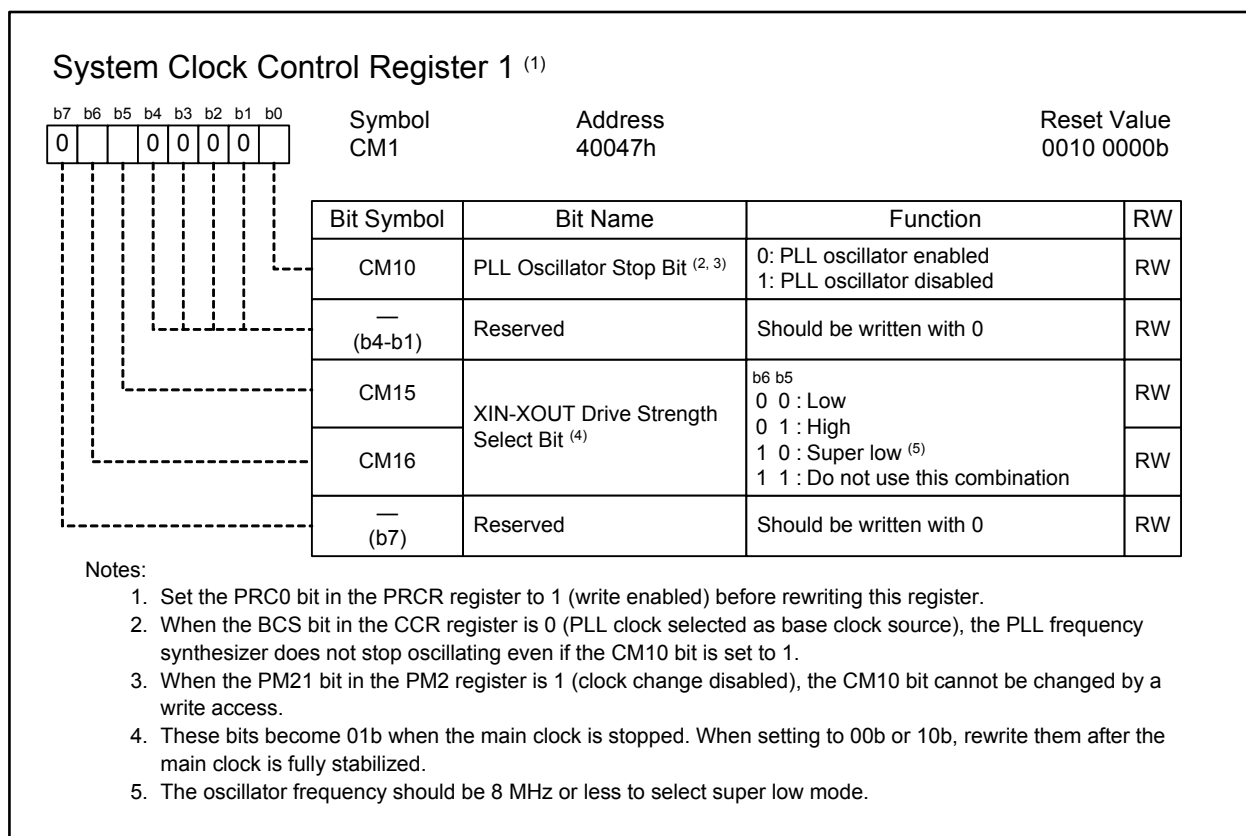


**Figure 8.2 CCR Register**



**Figure 8.3 CM0 Register**





**Figure 8.4 CM1 Register**

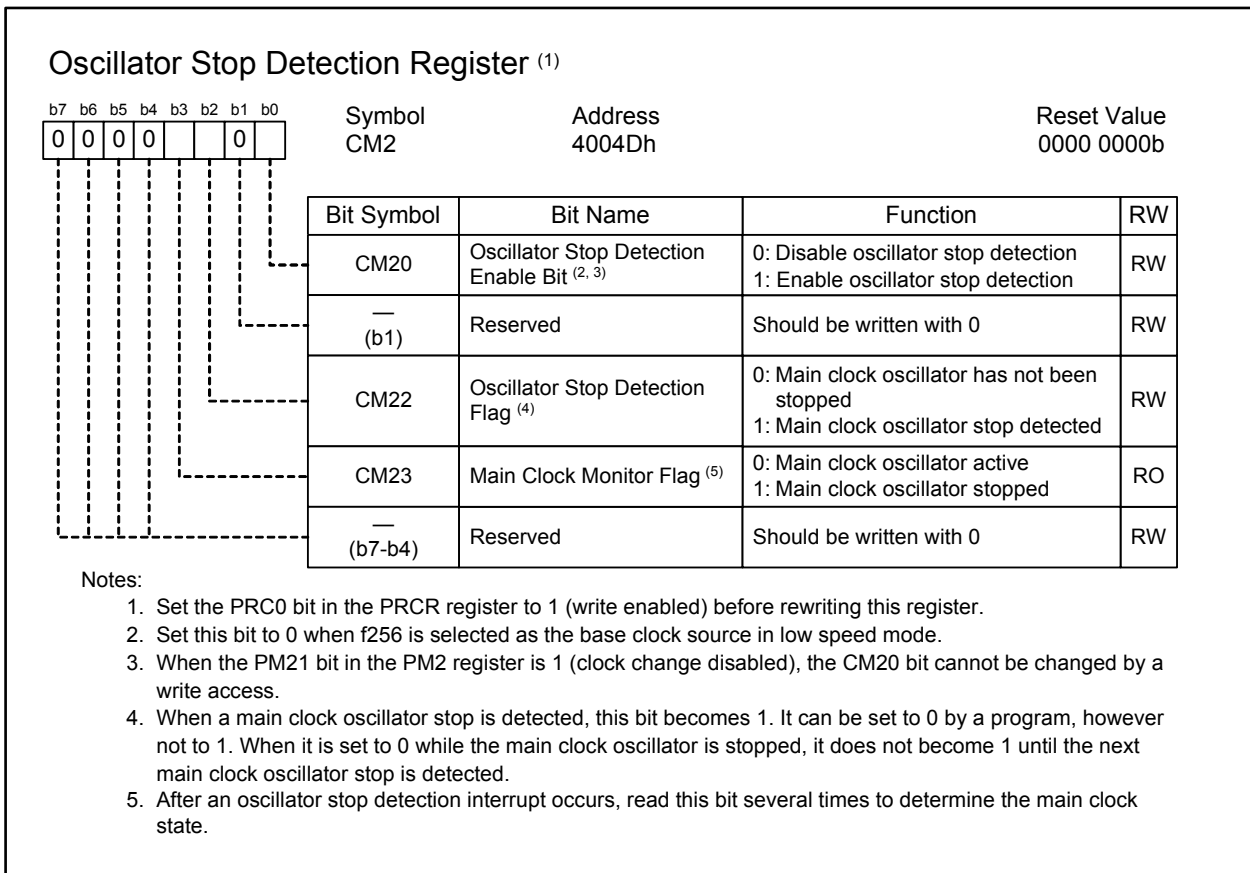


Figure 8.5 CM2 Register

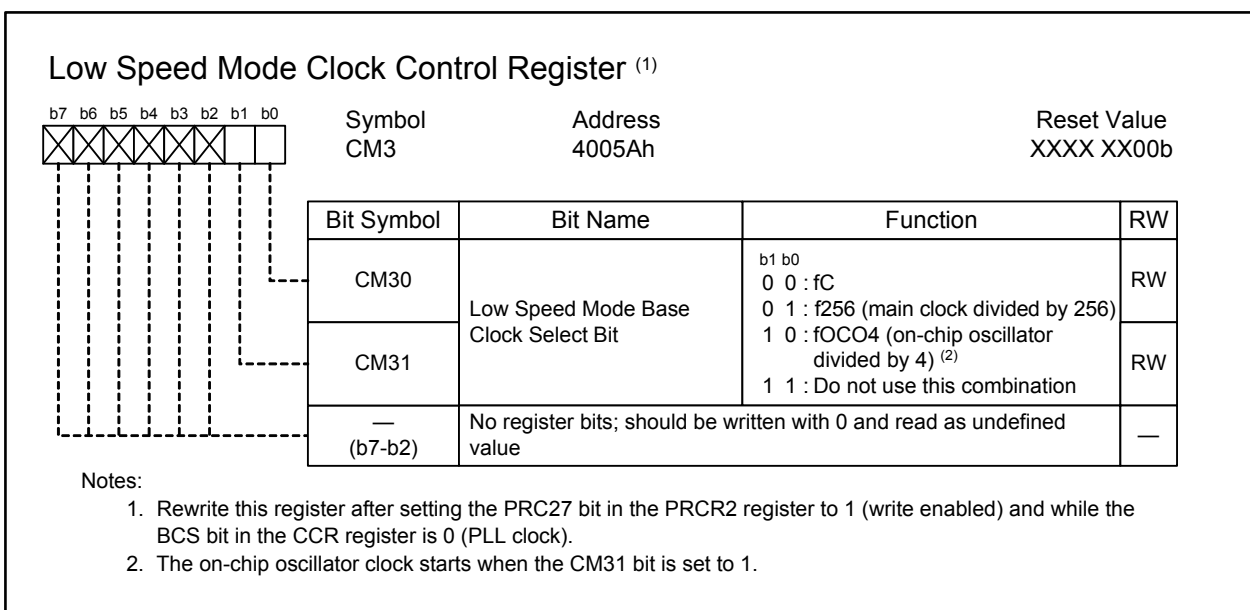


Figure 8.6 CM3 Register

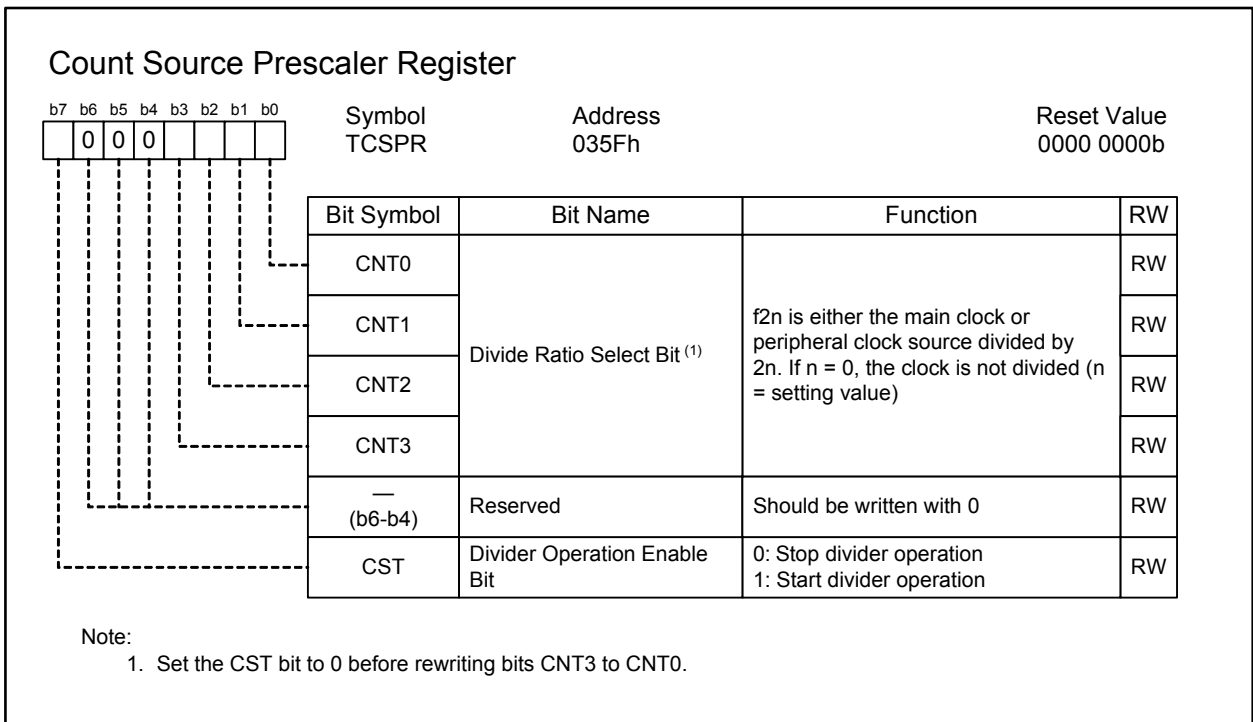


Figure 8.7 TCSPR Register

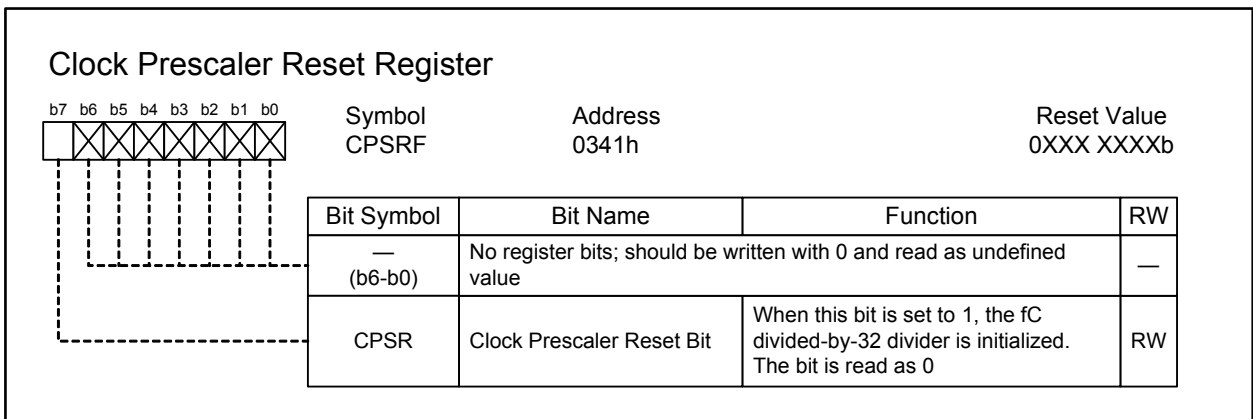
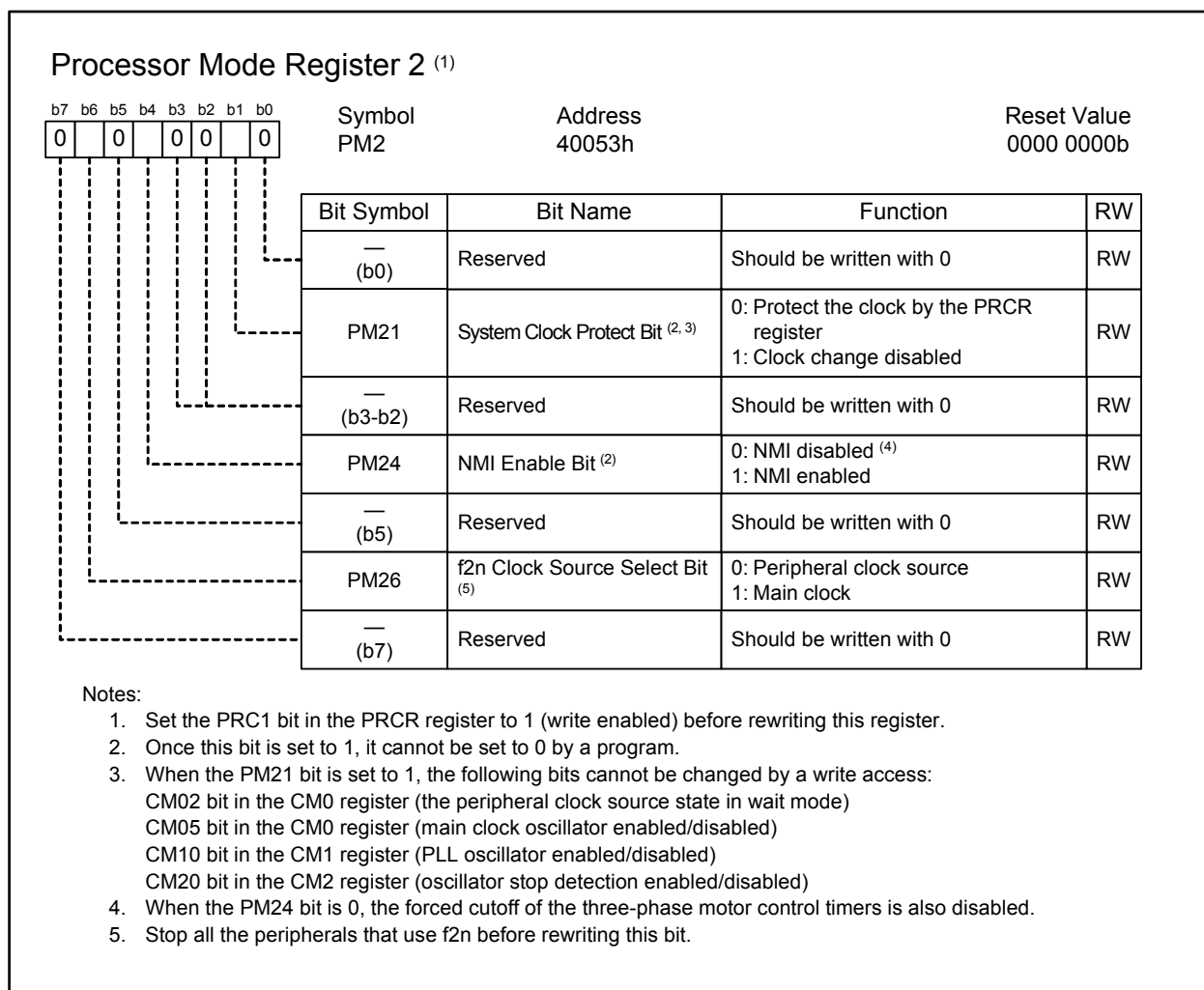
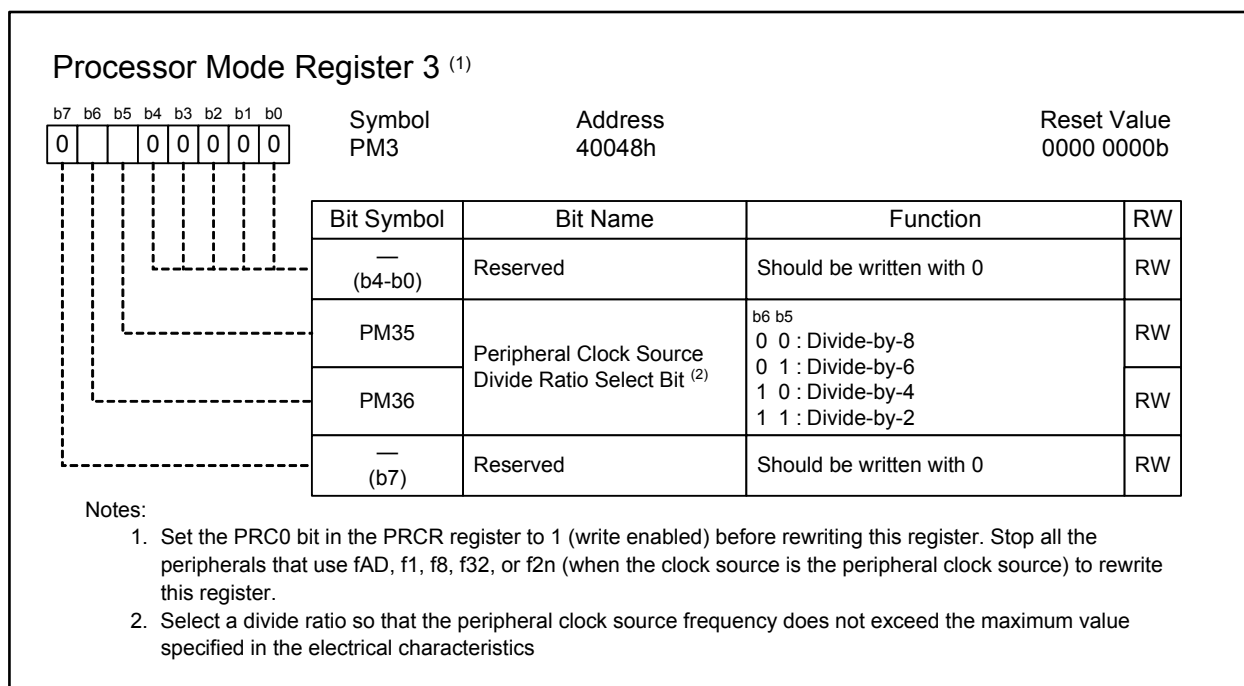


Figure 8.8 CPSRF Register

**Figure 8.9 PM2 Register**



**Figure 8.10 PM3 Register**

The following sections illustrate clocks generated in clock generators.

### 8.1.1 Main Clock

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

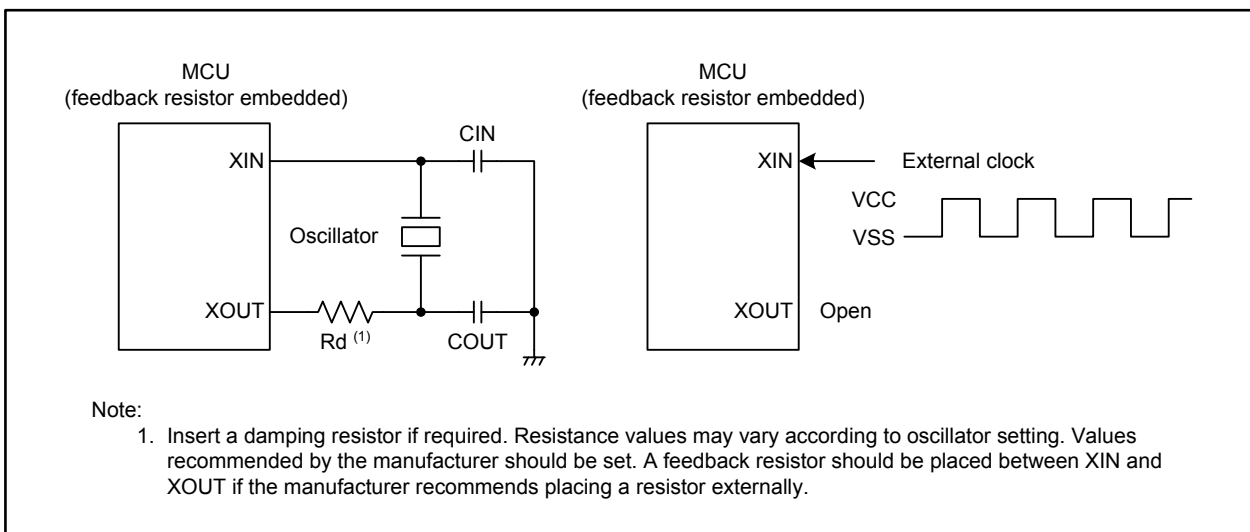
The main clock oscillator is configured with two pins, XIN and XOUT, connected by an oscillator or resonator. The circuit has an on-chip feedback resistor which is separated from the oscillator in stop mode to save power consumption. An external clock can be applied to the XIN pin in this circuit. Figure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

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

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

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

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



**Figure 8.11 Main Clock Circuit Connection**

### 8.1.2 Sub Clock (fC)

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

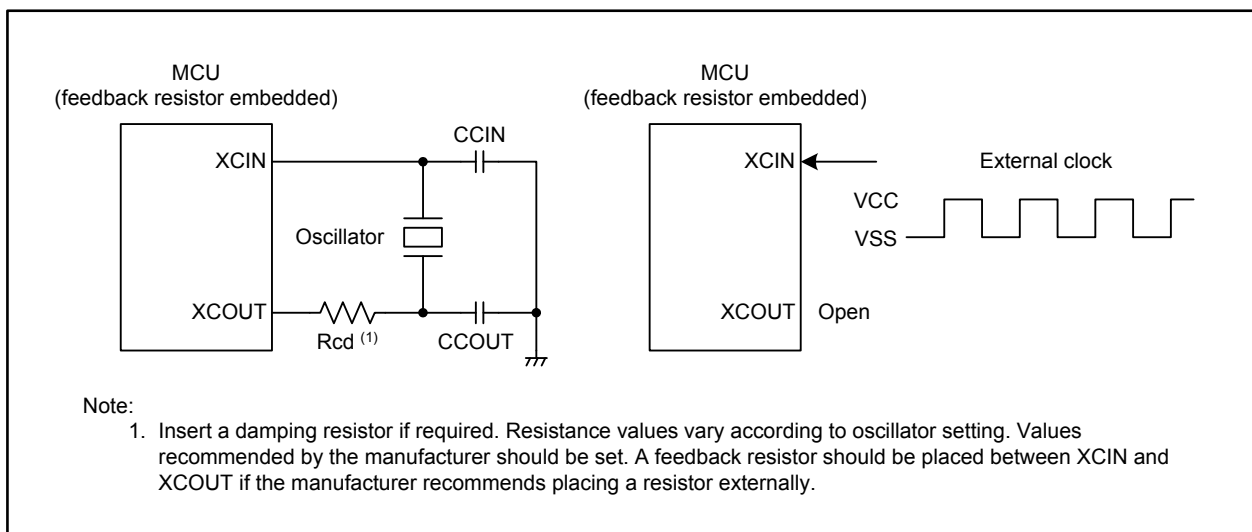
The sub clock oscillator is configured with pins XCIN and XCOU connected by a crystal oscillator. The circuit has a on-chip feedback resistor which is separated from the oscillator in stop mode to save power consumption. An external clock can be applied to the XCIN pin. Figure 8.12 shows an example of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

After a reset, the sub clock is stopped and the feedback resistor is separated from the oscillator. In order to start the sub clock oscillation, first set bits PD8\_6 and PD8\_7 in the PD8 register to 0 (input mode), and the PU25 bit in the PUR2 register to 0 (pull-up resistor disabled). Then, set the CM04 bit in the CM0 register to 1 (XCIN-XCOU oscillator).

To input an external clock to the XCIN pin, set bits PD8\_7 and PU25 to 0 and then the CM04 bit to 1. The clock applied to the XCIN pin becomes a clock source for the sub clock.

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

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



**Figure 8.12 Sub Clock Circuit Connection**

### 8.1.3 PLL Clock

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

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

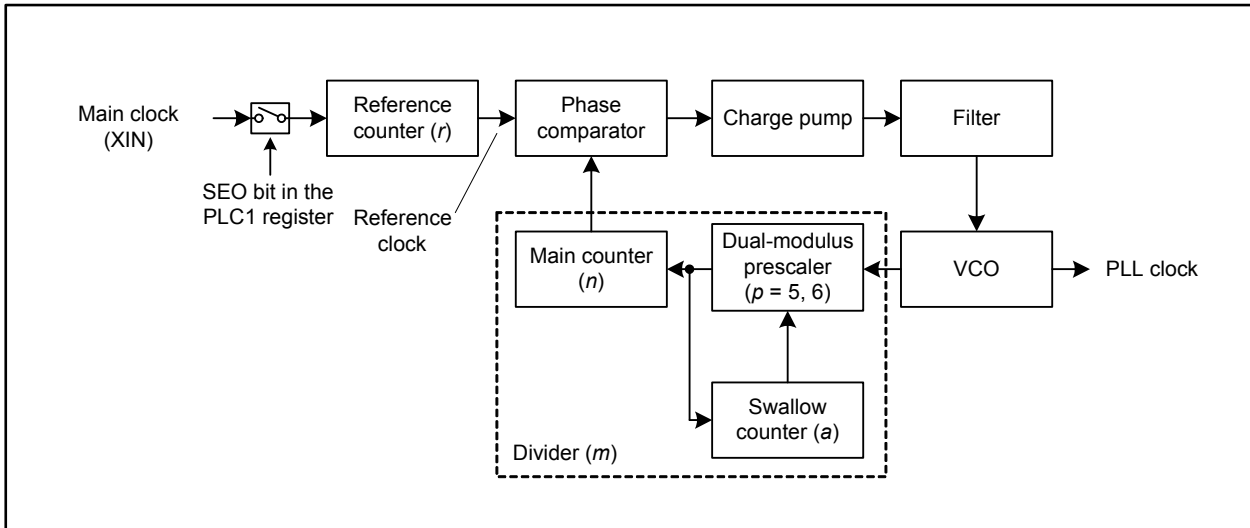


Figure 8.13 PLL Frequency Synthesizer Block Diagram

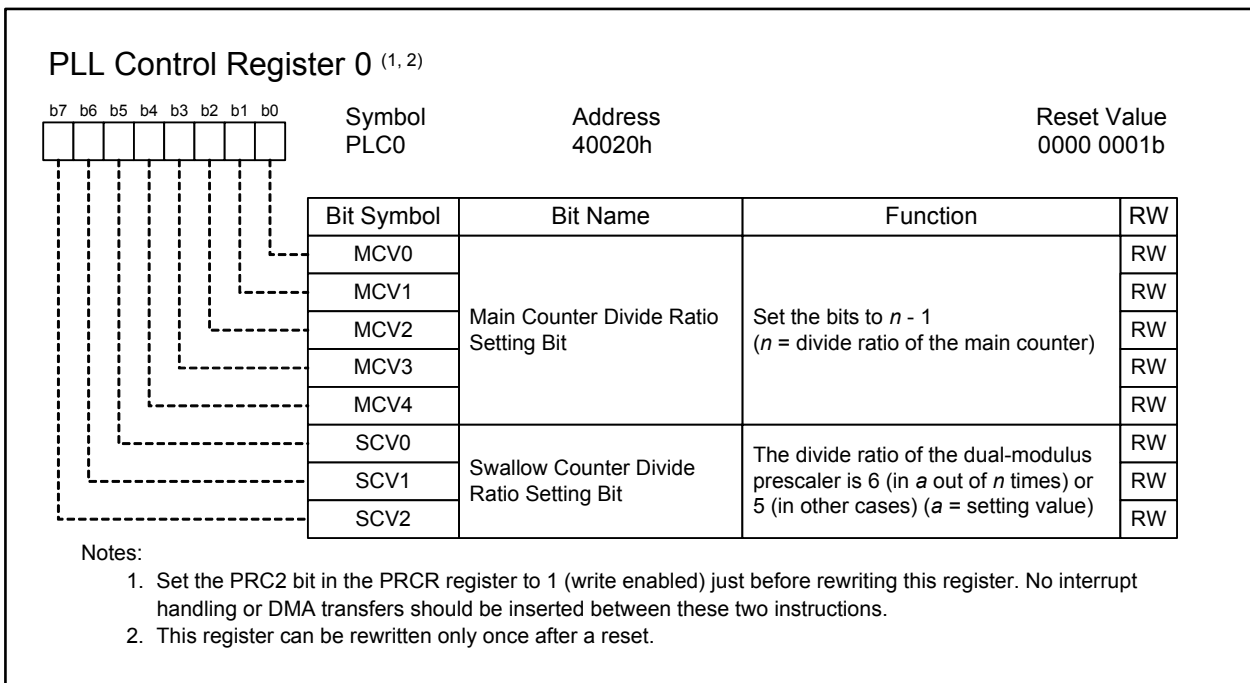
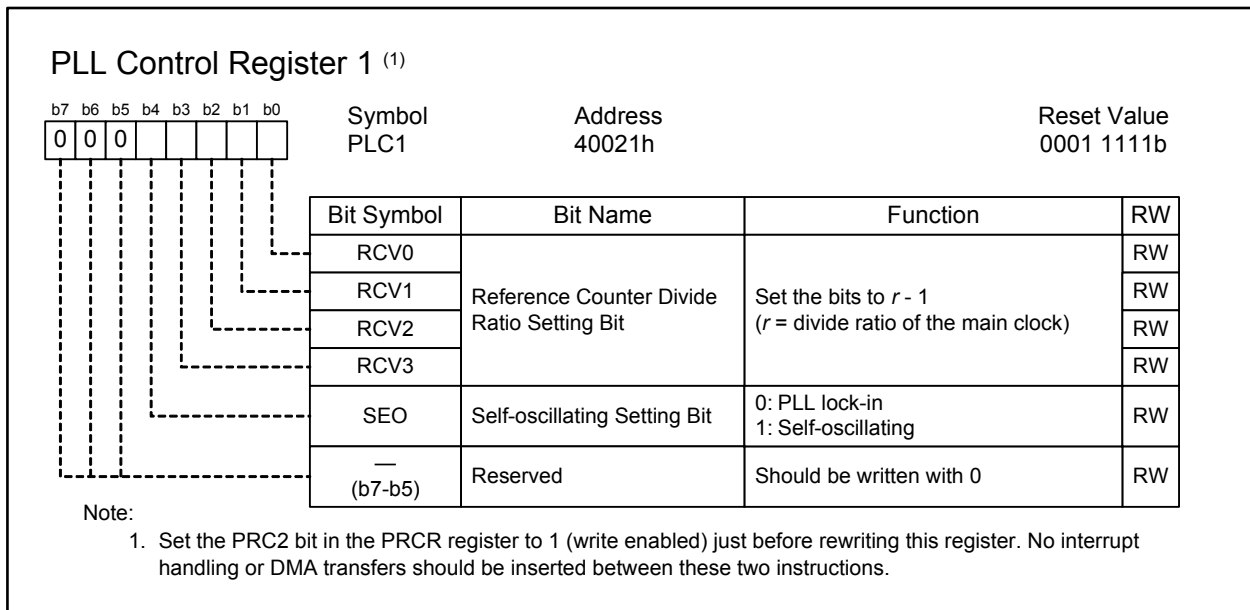


Figure 8.14 PLC0 Register





**Figure 8.15 PLC1 Register**

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

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

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

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

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

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

Each register should be set to meet the following conditions:

- The reference clock, which is the main clock divided by  $r$ , should be between 2 to 4 MHz.
- The divide ratio  $m$  is  $25 \leq m \leq 100$ .

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

**Table 8.2 PLC1 and PLC0 Register Settings (1)**

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

Note:

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

#### 8.1.4 On-chip Oscillator Clock

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

The on-chip oscillator clock is stopped after a reset. It starts running when setting the CM31 bit in the CM3 register to 1. It is not necessary to wait for stabilization because the on-chip oscillator instantly starts oscillating.

## 8.2 Oscillator Stop Detection

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

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

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

### 8.2.1 How to Use Oscillator Stop Detection

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

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

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

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

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

## 8.3 Base Clock

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

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

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

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

## 8.4 CPU Clock and Peripheral Bus Clock

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

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

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

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

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

## 8.5 Peripheral Clock

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

### (1) f1, f8, f32, f2n

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

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

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

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

### (2) fAD

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

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

### (3) fC32

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

## 8.6 Clock Output Function

Low speed clocks, f8, and f32 can be output from the CLKOUT pin in the 100-pin package.

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

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

**Table 8.3 CLKOUT Pin Functions in Single-chip Mode**

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

Notes:

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

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

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

Notes:

1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
3. When the PM07 bit is set to 0 (output BCLK), set bits CM01 and CM00 to 00b (I/O port P5\_3).

## 8.7 Power Control

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

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

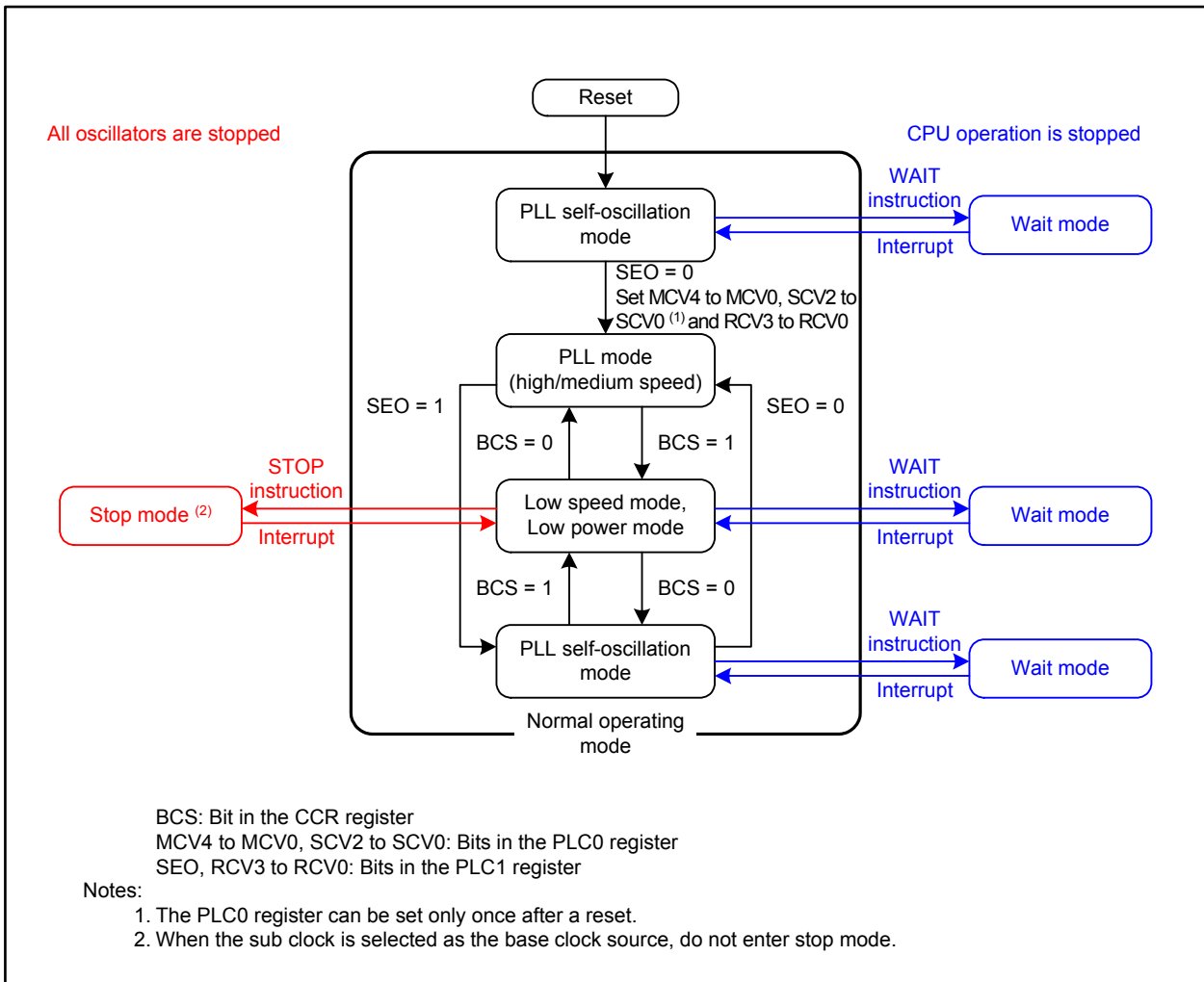


Figure 8.16 State Transition in Stop Mode and Wait Mode

### 8.7.1 Normal Operating Mode

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

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

#### (1) PLL Mode (high speed mode)

In this mode, the PLL clock is selected as the base clock source, and the main clock is provided as the reference clock source for the PLL frequency synthesizer. High speed mode enables the CPU to operate at the maximum operating frequency. The PLL clock divided by 2 becomes the base clock. The base clock frequency should be identical to that of the CPU clock.  $f_{AD}$ ,  $f_1$ ,  $f_8$ ,  $f_{32}$ , and  $f_{2n}$  can be used as the peripheral clocks. When the sub clock or the on-chip oscillator clock is provided,  $f_{C32}$  can be used as the count source for timers A and B.

#### (2) PLL Mode (medium speed mode)

This mode indicates all modes in PLL mode except high speed mode. The PLL clock divided by 2, 3, 4, or 6 becomes the base clock and the base clock divided by 1 to 4 becomes the CPU clock.  $f_{AD}$ ,  $f_1$ ,  $f_8$ ,  $f_{32}$ , and  $f_{2n}$  can be used as the peripheral clocks. When the sub clock or the on-chip oscillator clock is provided,  $f_{C32}$  can be used as the count source for timers A and B.

#### (3) Low Speed Mode

In this mode, a low speed clock is used as the base clock source. The low speed clock becomes the base clock and the base clock divided by 1 to 4 becomes the CPU clock.  $f_{AD}$ ,  $f_1$ ,  $f_8$ ,  $f_{32}$ , and  $f_{2n}$  can be used as the peripheral clocks. When the sub clock or the on-chip oscillator clock is provided,  $f_{C32}$  can be used as the count source for timers A and B.

#### (4) Low Power Mode

This is a state where the main clock oscillator and the PLL frequency synthesizer are stopped after switching to low speed mode. The sub clock or the on-chip oscillator clock divided by 4 becomes the base clock and the base clock divided by 1 to 4 becomes the CPU clock.  $f_{C32}$ , which is the only peripheral clock available, can be used as the count source for timers A and B. By setting the MRS bit in the VRCCR register to 1 (main regulator stopped), this mode consumes even less power than the modes above.

#### (5) PLL Self-oscillation Mode

In this mode, the PLL clock is selected as the base clock source, and the main clock is not provided as the reference clock source for the PLL frequency synthesizer. The PLL frequency synthesizer self-oscillates at its own frequency. The PLL clock divided by 2, 3, 4, or 6 becomes the base clock and the base clock divided by 1 to 4 becomes the CPU clock.  $f_{AD}$ ,  $f_1$ ,  $f_8$ ,  $f_{32}$ , and  $f_{2n}$  can be used as the peripheral clocks. When the sub clock or the on-chip oscillator clock is provided,  $f_{C32}$  can be used as the count source for timers A and B.



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

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

Note:

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

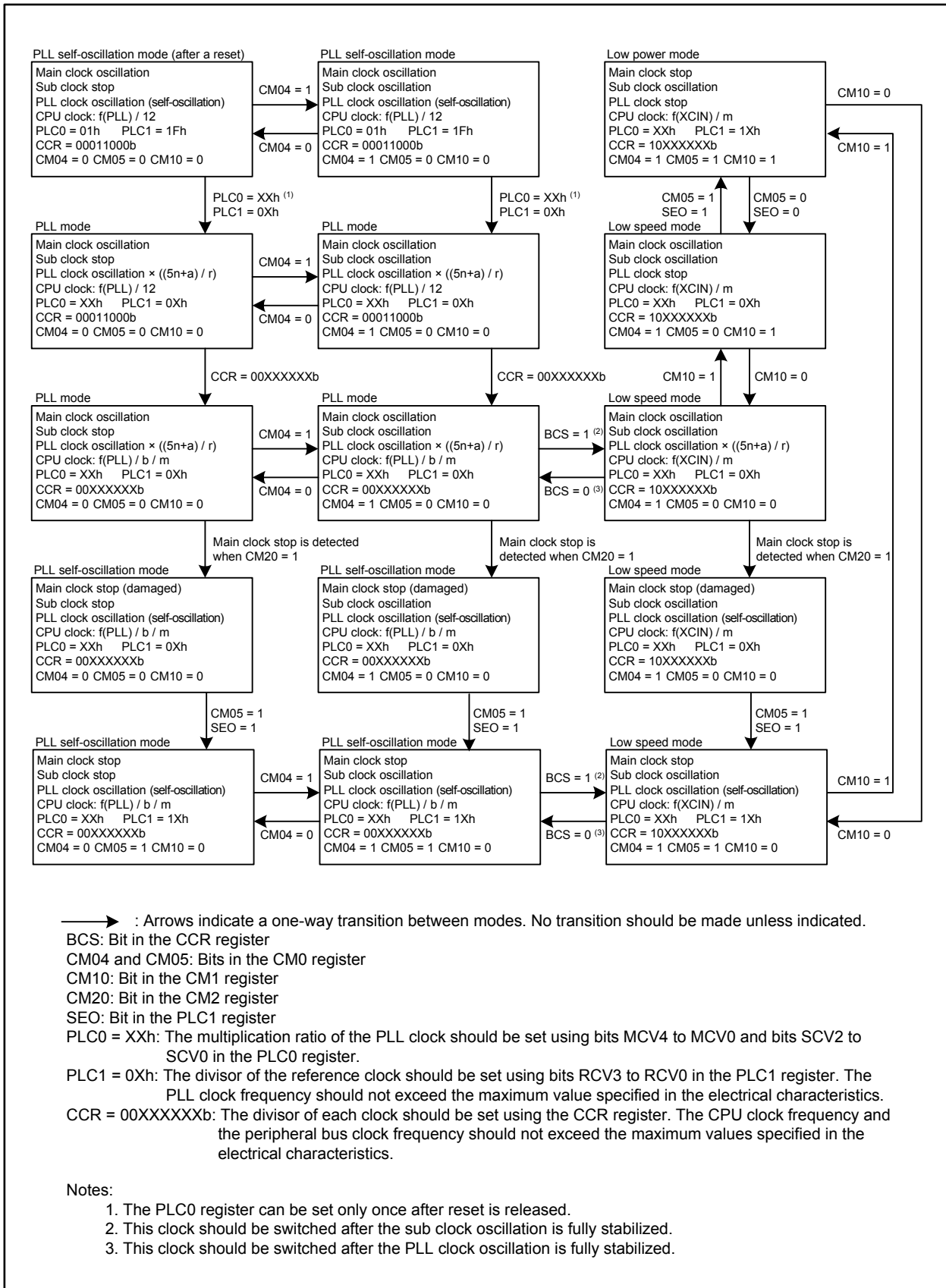


Figure 8.17 State Transition When Using the Sub Clock

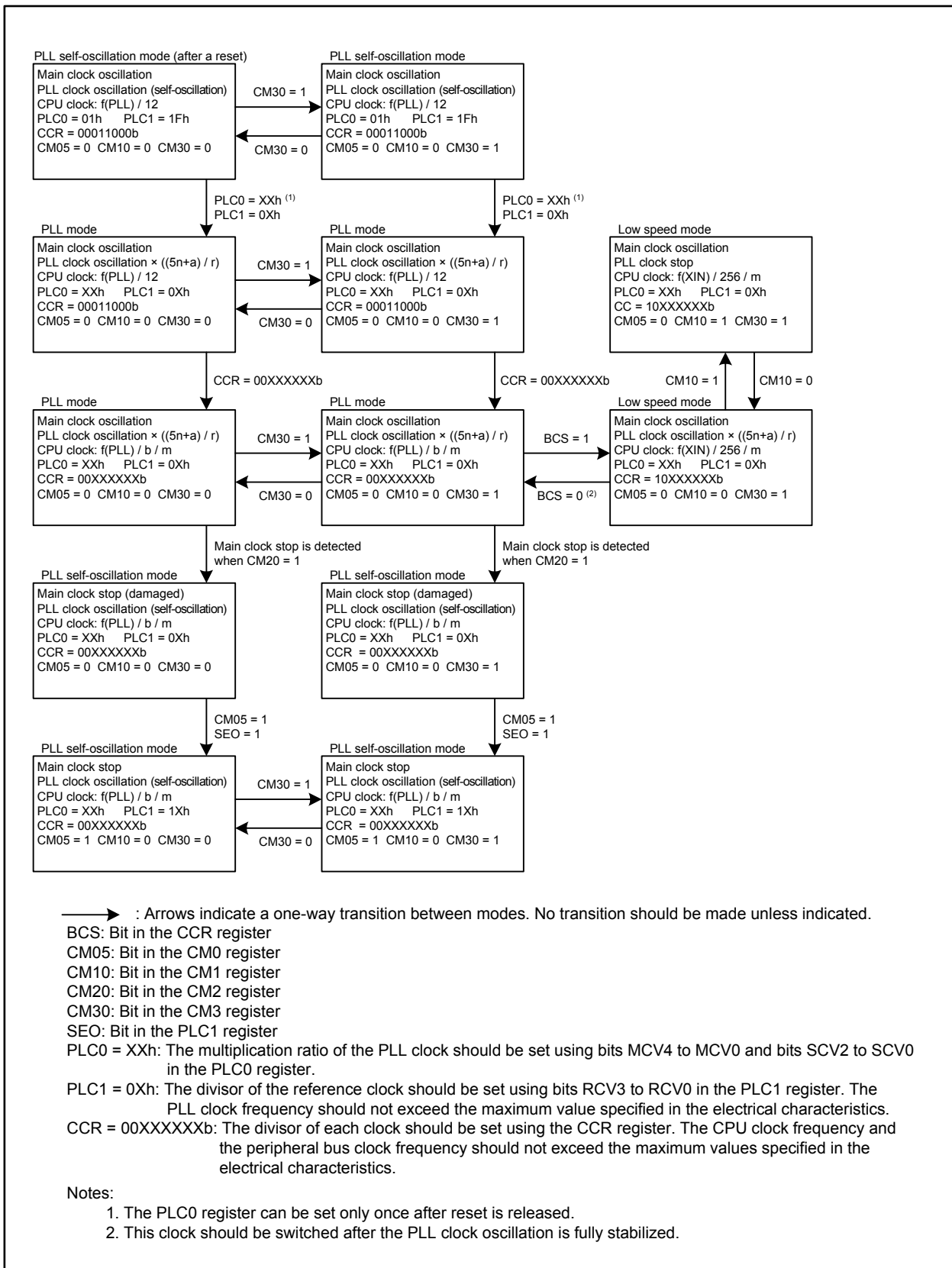


Figure 8.18 State Transition When Using the Main Clock Divided by 256

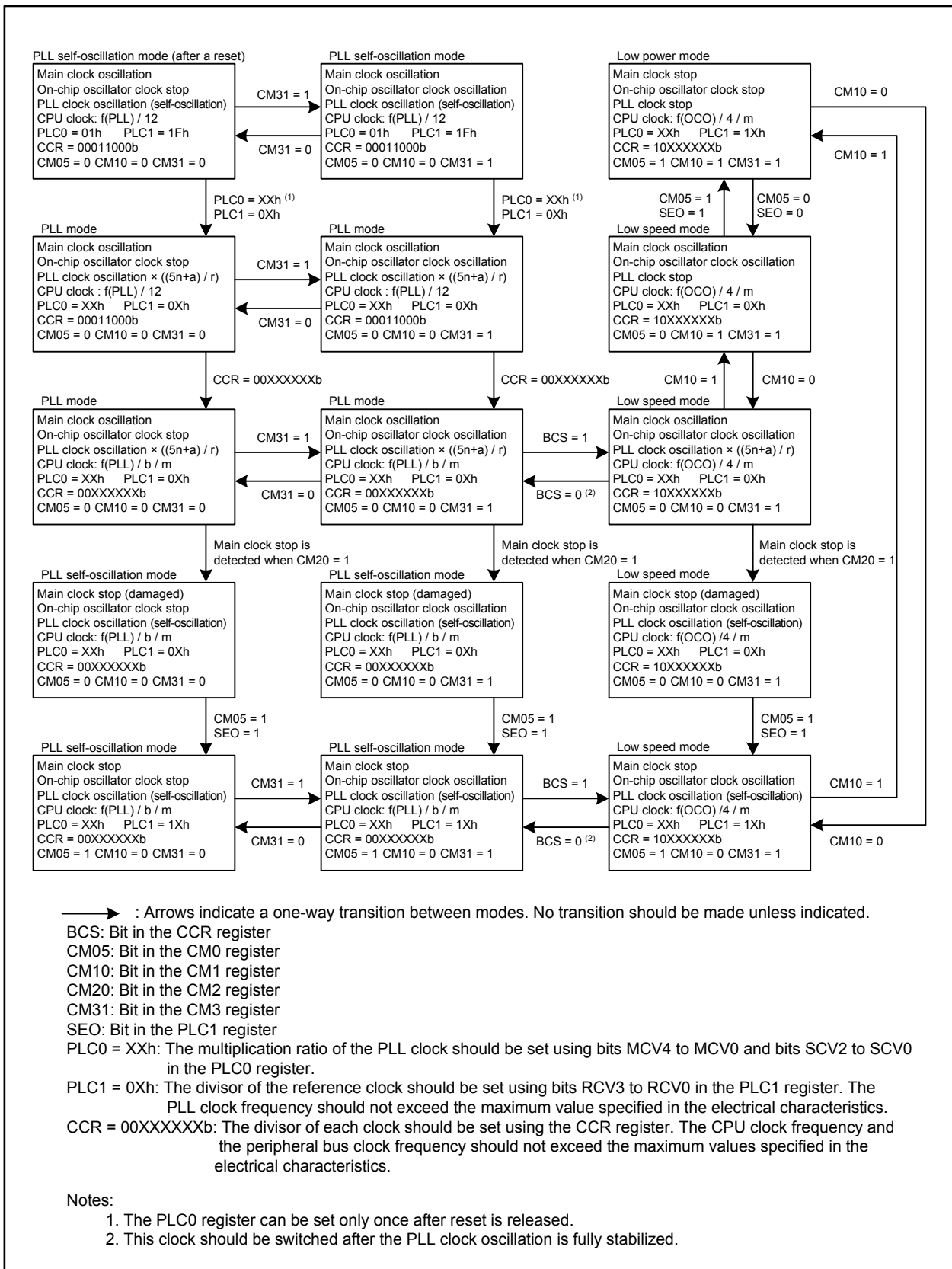


Figure 8.19 State Transition When Using the On-chip Oscillator Clock

## 8.7.2 Wait Mode

The base clock stops in wait mode, so clocks generated by the base clock, the CPU clock and peripheral bus clock, stop running as well. Thus the CPU and watchdog timer, operated by these two clocks, also stop. Since the main clock, sub clock, PLL clock, and on-chip oscillator clock continue running, the peripherals using these clocks also continue operating.

### 8.7.2.1 Peripheral Clock Source Stop Function

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

### 8.7.2.2 Entering Wait Mode

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

- Initial setting
  - Set the wake-up interrupt priority level (bits RLVL2 to RLVL0 in registers RIPL1 and RIPL2) to 7. Then set each interrupt request level.
- Steps before entering wait mode
  - (1) Set the I flag to 0.
  - (2) Set the interrupt request level for each interrupt source (interrupt number from 1 to 127) to 0, if its interrupt request level is not 0.
  - (3) Perform a dummy read of any of the interrupt control registers.
  - (4) Set the processor interrupt priority level (IPL) in the flag register to 0.
  - (5) Enable interrupts temporarily by executing the following instructions:
 

```
FSET I
NOP
NOP
FCLR I
```
  - (6) Set the interrupt request level for the interrupt to exit wait mode. Do not rewrite the interrupt control register after this step.
  - (7) Set the IPL in the flag register.
  - (8) Set the interrupt priority level for resuming to the same level as the IPL.
 

Interrupt request level for the interrupt to exit wait mode > IPL = Interrupt priority level for resuming
  - (9) Set the CM20 bit in the CM2 register to 0 (disable oscillator stop detection) when the oscillator stop detection is used.
  - (10) Enter either PLL self-oscillation mode, low speed mode, or low power mode.
  - (11) Set the I flag to 1.
  - (12) Execute the WAIT instruction.
- After exiting wait mode
  - Set the wake-up interrupt priority level to 7 immediately after exiting wait mode.

### 8.7.2.3 Pin State in Wait Mode

Table 8.5 lists the pin state in wait mode.

**Table 8.5 Pin State in Wait Mode**

Pin		Memory Expansion Mode/ Microprocessor Mode	Single-chip Mode
Address bus, data bus, $\overline{CS0}$ to $\overline{CS3}$ , $\overline{BC0}$ to $\overline{BC1}$		The state immediately before entering wait mode is held	—
$\overline{RD}$ , $\overline{WR}$ , $\overline{WR0}$ to $\overline{WR1}$		High	—
HLDA, BCLK		High	—
ALE		High	—
Ports		The state immediately before entering wait mode is held	
DA0, DA1		The state immediately before entering wait mode is held	
CLKOUT	When a low speed clock is selected	The clock is output	
	When f8 or f32 is selected	The clock is output when the CM02 bit in the CM0 register is 0 (no peripheral clock source stopped in wait mode). The state immediately before entering wait mode is held when the CM02 bit is 1 (peripheral clock source stopped in wait mode)	

### 8.7.2.4 Exiting Wait Mode

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

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

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

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

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

**Table 8.6 Interrupts for Exiting Wait Mode and Usage Conditions**

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

Note:

1. UART7 and UART8 are excluded.

### 8.7.3 Stop Mode

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

#### 8.7.3.1 Entering Stop Mode

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

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



### 8.7.3.2 Pin State in Stop Mode

Table 8.7 lists the pin state in stop mode.

**Table 8.7 Pin State in Stop Mode**

Pin	Memory Expansion Mode/ Microprocessor Mode	Single-chip Mode
Address bus, data bus, $\overline{CS0}$ to $\overline{CS3}$ , $\overline{BC0}$ to $\overline{BC1}$	The state immediately before entering stop mode is held	—
$\overline{RD}$ , $\overline{WR}$ , $\overline{WR0}$ to $\overline{WR1}$	High	—
$\overline{HLDA}$ , $\overline{BCLK}$	High	—
ALE	High	—
Ports	The state immediately before entering stop mode is held	
DA0, DA1	The state immediately before entering stop mode is held	
CLKOUT	When a low speed clock is selected	High
	When f8 or f32 is selected	The state immediately before entering stop mode is held
XIN	High-impedance	
XOUT	High	
XCIN, XCOU	High-impedance	

### 8.7.3.3 Exiting Stop Mode

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

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

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

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

**Table 8.8 Interrupts for Exiting Stop Mode and Usage Conditions**

Interrupt	Usage Condition
NMI	
Low voltage detection interrupt	
External interrupt	
Key input interrupt	
Timer A interrupt Timer B interrupt	Available when a timer counts an external pulse with a frequency of 100 Hz or less in event counter mode
Serial interface interrupt (1)	Available when an external clock is used

Note:

1. UART7 and UART8 are excluded.

## 8.8 System Clock Protection

The system clock protection disables clock change when the PLL clock is selected as the base clock source. This prevents the CPU clock from stopping due to a runaway program.

When the PM21 bit in the PM2 register is set to 1 (clock change disabled), the following bits cannot be written to:

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

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

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

## 8.9 Notes on Clock Generator

### 8.9.1 Sub Clock

#### 8.9.1.1 Oscillator Constant Matching

The constant matching of the sub clock oscillator should be evaluated in both cases when the drive strength is high and low.

Contact the oscillator manufacturer for details on the oscillation circuit constant matching.

### 8.9.2 Power Control

Do not switch the base clock source until the oscillation of the clock to be used has stabilized. However, this does not apply to the on-chip oscillator since it starts running immediately after the CM31 bit in the CM3 register is set to 1.

To switch the base clock source from the PLL clock to a low speed clock, use the MOV.L or OR.L instruction to set the BCS bit in the CCR register to 1.

- Program example in assembly language

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

- Program example in C language

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

#### 8.9.2.1 Stop Mode

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

#### 8.9.2.2 Suggestions for Power Saving

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

- I/O pins:

If inputs are floating, both transistors may be conducting. Set unassigned pins to input mode and connect each of them to VSS via a resistor, or set them to output mode and leave them open.

- A/D converter:

When not performing the A/D conversion, set the VCUT bit in the AD0CON1 register to 0 (VREF disconnected). To perform the A/D conversion, set the VCUT bit to 1 (VREF connected) and wait at least 1  $\mu\text{s}$  before starting conversion.

- D/A converter:

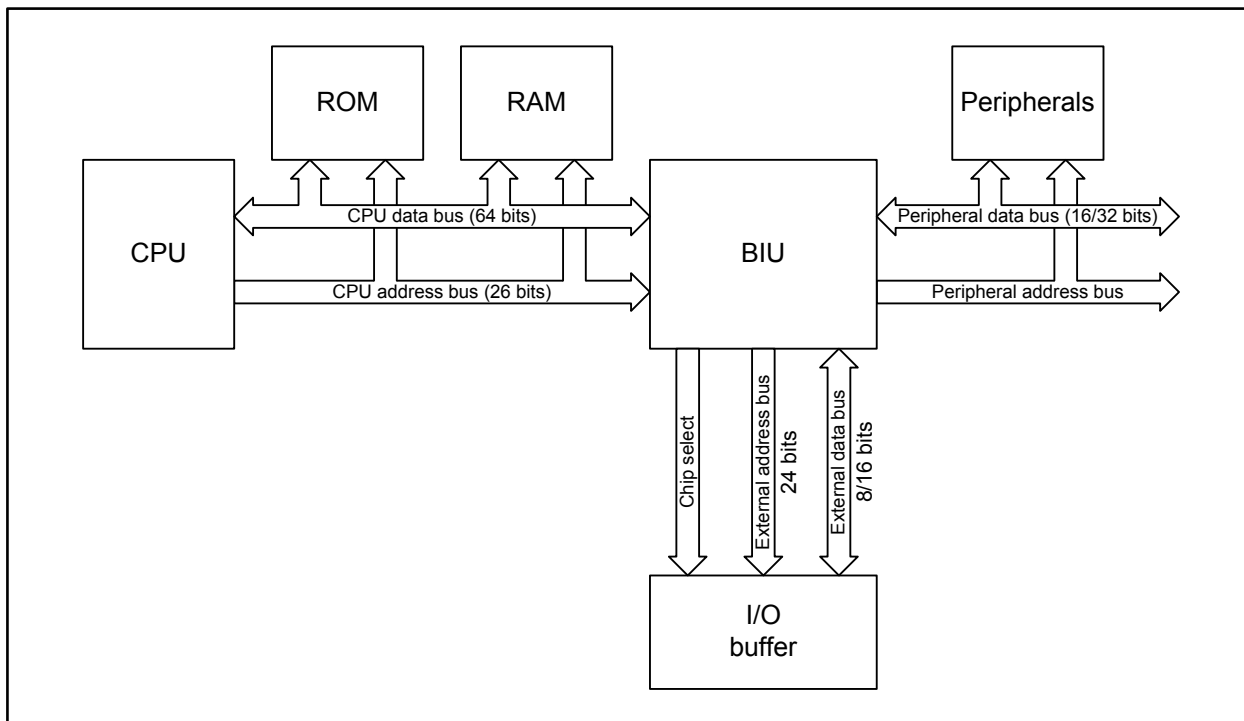
When not performing the D/A conversion, set the DAiE bit in the DACON register ( $i = 0, 1$ ) to 0 (output disabled) and the DAi register to 00h.

- Peripheral clock stop:

When entering wait mode, power consumption can be reduced by setting the CM02 bit in the CM0 register to 1 to stop the peripheral clock source. However, this setting does not stop the fC32.

## 9. Bus

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



**Figure 9.1 Bus Block Diagram**

In memory expansion mode or microprocessor mode, some pins function as bus control pin to control the address bus and the data bus. The bus control pins are as follows: A0 to A23, D0 to D15,  $\overline{CS0}$  to  $\overline{CS3}$ ,  $\overline{WR0}/\overline{WR}$ ,  $\overline{BC0}$ ,  $\overline{WR1}/\overline{BC1}$ ,  $\overline{RD}$ ,  $\overline{BCLK}$ ,  $\overline{HLDA}$ ,  $\overline{HOLD}$ ,  $\overline{ALE}$ , and  $\overline{RDY}$ .

### 9.1 Bus Settings

The bus settings are controlled by the two lowest bits of the reset vector, the PBC register, registers EBC0 to EBC3, and CSOP0 and CSOP1.

Table 9.1 lists bus settings and their sources.

**Table 9.1 Bus Settings and Sources**

Bus Settings	Sources
Internal SFR bus timing	PBC register
External bus timing	Registers EBC0 to EBC3
External data bus width	PBC register, registers EBC0 to EBC3
External data bus width after reset	Two lowest bits of the reset vector
Separate bus/multiplexed bus selection	PBC register, registers EBC0 to EBC3
Pins outputting chip select signals	Registers CSOP0 and CSOP1

## 9.2 Peripheral Bus Timing Setting

The 16-/32-bit wide peripheral bus operates at a frequency up to 32 MHz (the theoretical value and the maximum frequency of each product group are as defined by  $f(\text{BCLK})$  in 26. "Electrical Characteristics"). The timing adjustment and bus-width conversion with the faster, 64-bit wide CPU bus are controlled in the bus interface unit (BIU).

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

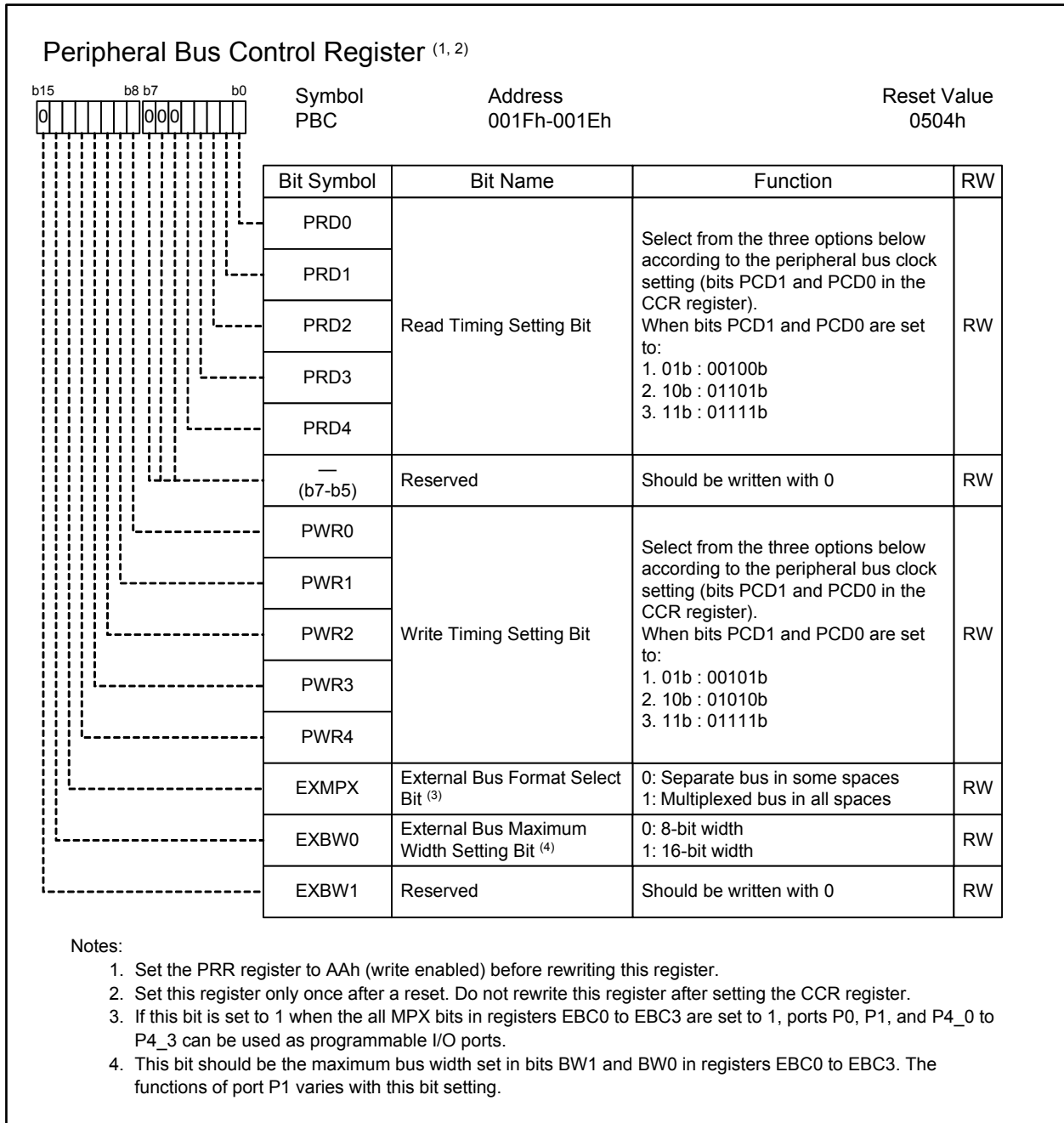


Figure 9.2 PBC Register

### 9.3 External Bus Setting

The 8-/16-bit wide external bus operates at a frequency up to 32 MHz (the theoretical value and the maximum frequency of each product group are as defined by  $f(\text{BCLK})$  in 26. “Electrical Characteristics”). The timing adjustment and bus-width conversion with the faster 64-bit wide CPU bus are controlled in the bus interface unit (BIU).

#### 9.3.1 External Address Space Setting

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

Up to 24 address lines from A0 to A23 can be used for external output. Decoded A18 to A25 function as 4 chip select signals ( $\overline{\text{CS}}_3$  to  $\overline{\text{CS}}_0$ ). If a 16 MB space is assigned to each chip select signal, up to 63.5 MB can be used as external address space. When the processor mode is changed from single-chip mode to memory expansion mode, the address bus status is undefined until an external space is accessed.

Chip select signals  $\overline{\text{CS}}_3$  to  $\overline{\text{CS}}_0$  share pins with A20 to A23, respectively. Other combinations of signal and output port are also available as follows: signals  $\overline{\text{CS}}_1$  to  $\overline{\text{CS}}_3$  with ports P5\_4, P5\_6, and P5\_7.

In microprocessor mode, the  $\overline{\text{CS}}_0$  signal is output from port P4\_7 after a reset. The maximum space per chip select signal is 8 MB since A23 is not available. Signals  $\overline{\text{CS}}_1$  to  $\overline{\text{CS}}_3$  are output only when being set.

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

Set registers CSOP0 and CSOP1 to select a chip select signal to be used and its output pin. Set registers CB01, CB12, and CB23 to set the address space for each chip select signal.

Figures 9.4 and 9.5 show registers CSOP0 and CSOP1. Figures 9.6, 9.7, and 9.8 show registers CB01, CB12, and CB23, respectively. Figures 9.9 and 9.10 show the chip select space.

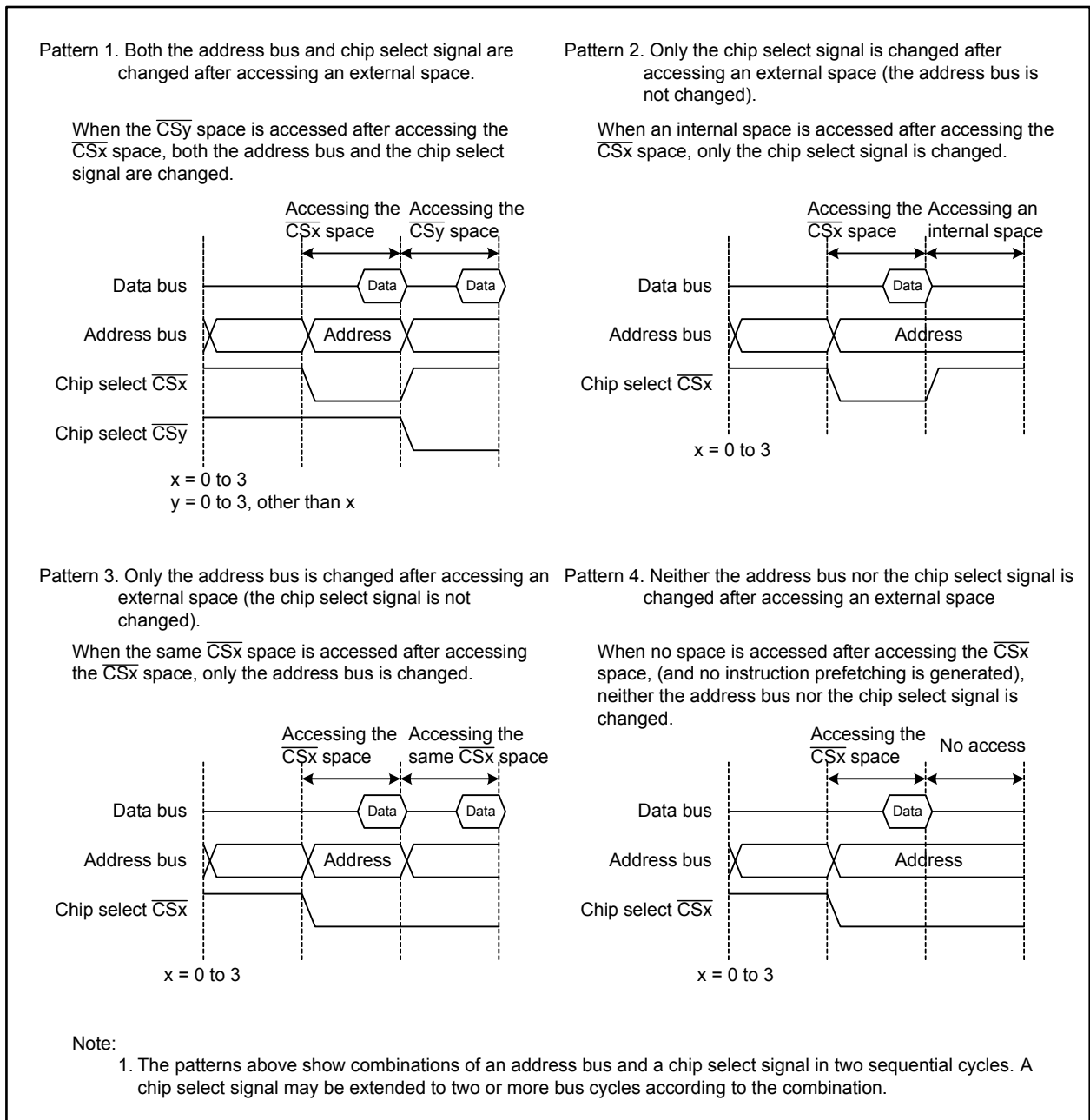
A chip select signal should not be set for more than two output pins in registers CSOP0 and CSOP1. Registers CB01, CB12, and CB23 should be set to meet the conditions below:

- In memory expansion mode

$$0080000\text{h} \leq (\text{CB}23 \times 2^{18}) \leq (\text{CB}12 \times 2^{18}) \leq (\text{CB}01 \times 2^{18}) \leq 3\text{DC}0000\text{h}$$

- In microprocessor mode

$$0080000\text{h} \leq (\text{CB}23 \times 2^{18}) \leq (\text{CB}12 \times 2^{18}) \leq (\text{CB}01 \times 2^{18}) \leq 3\text{FC}0000\text{h}$$



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

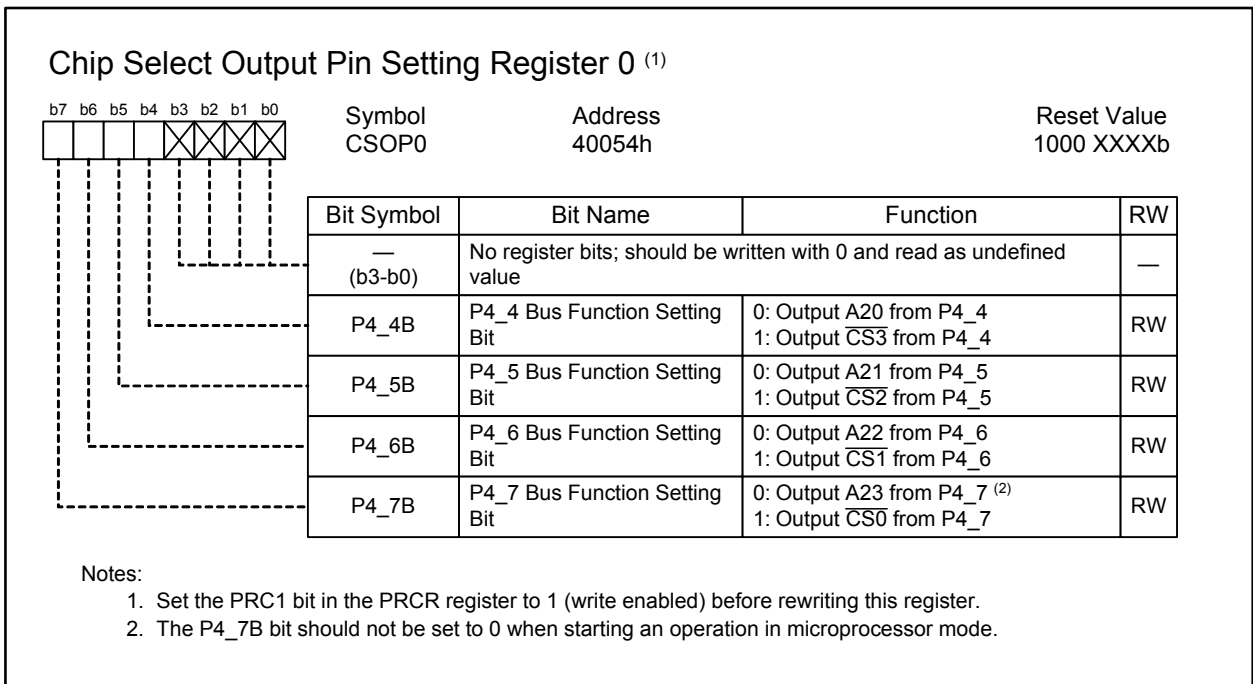


Figure 9.4 CSOP0 Register

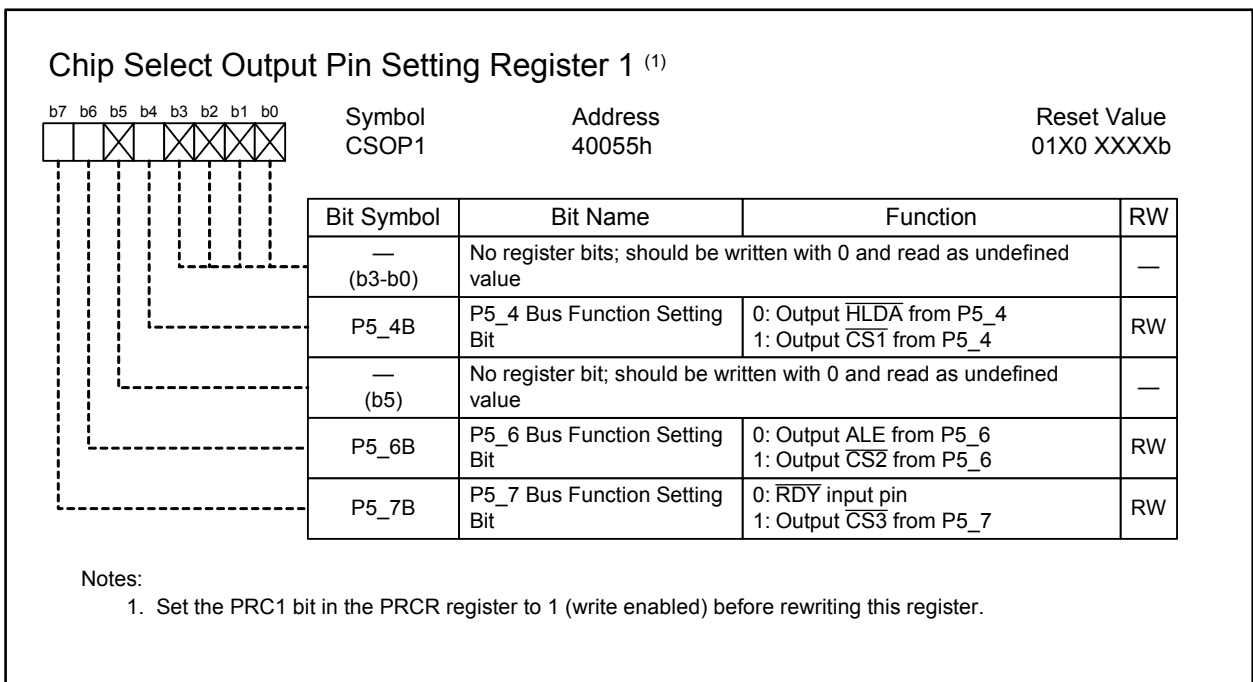
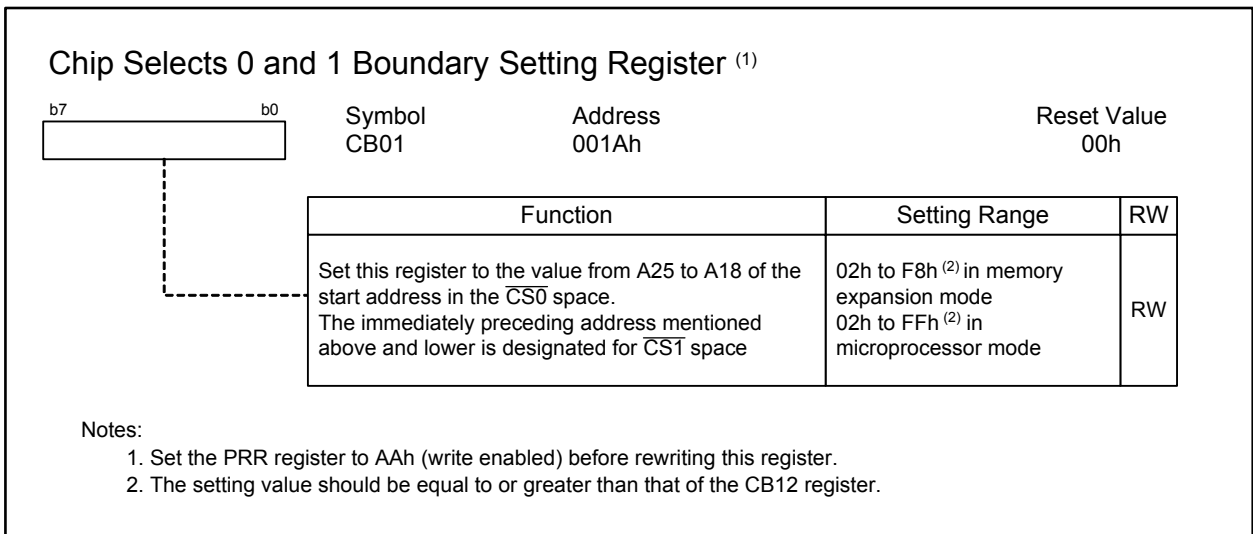
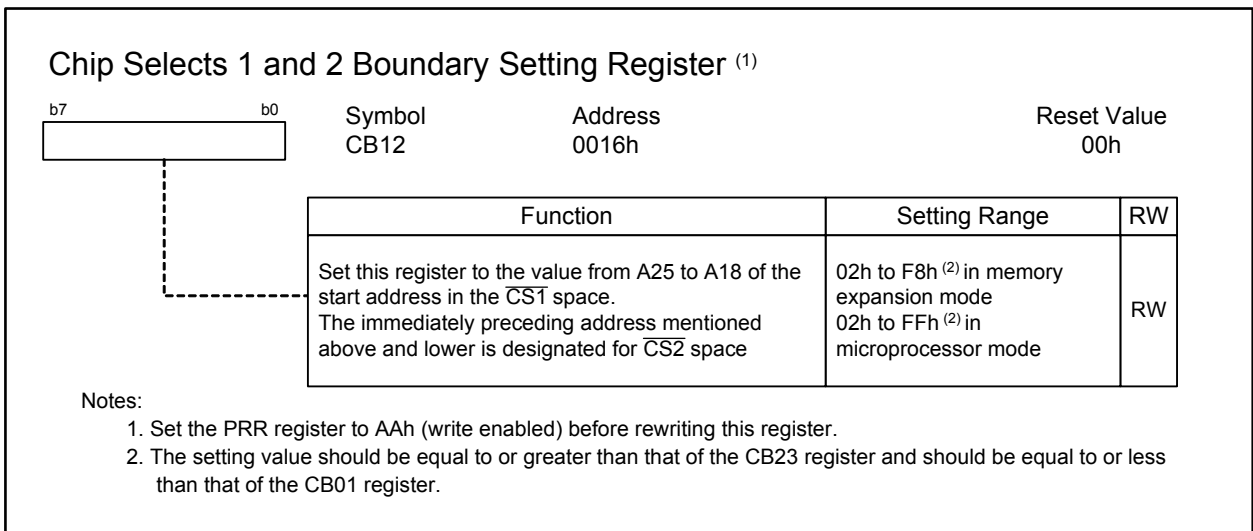
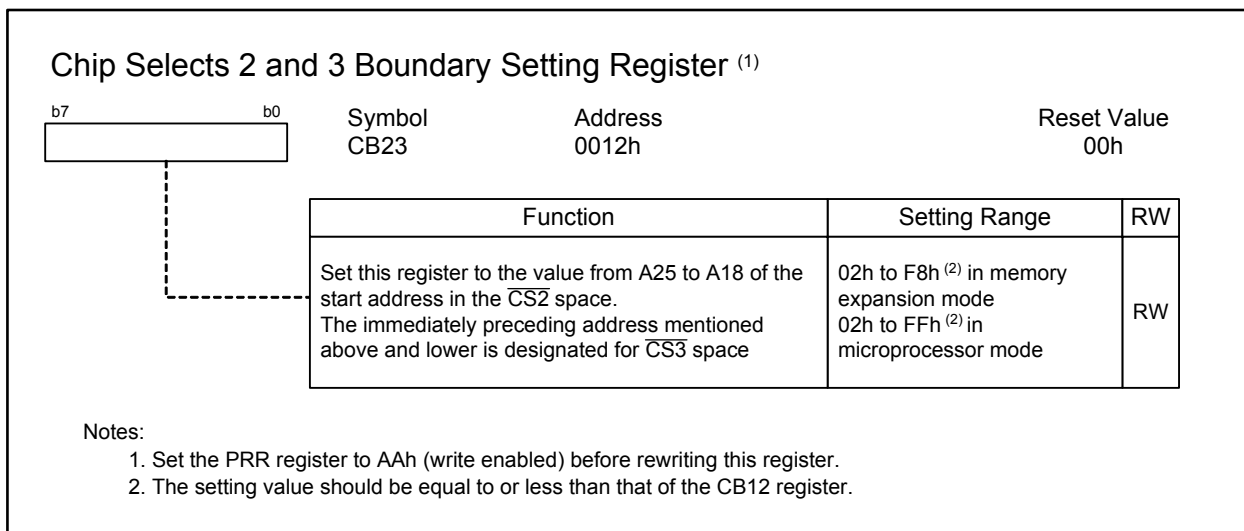


Figure 9.5 CSOP1 Register



**Figure 9.6 CB01 Register****Figure 9.7 CB12 Register**

**Figure 9.8 CB23 Register**

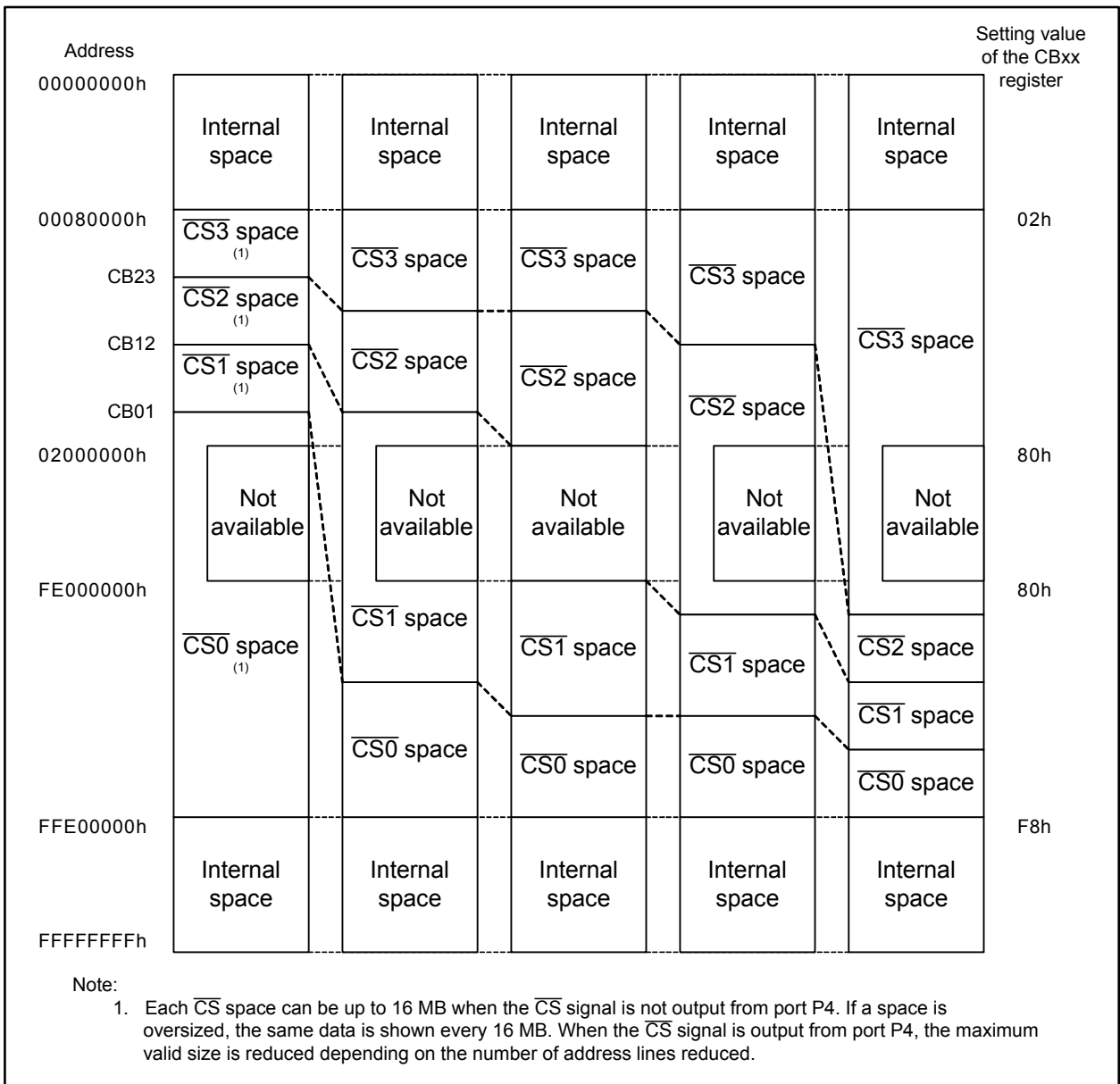
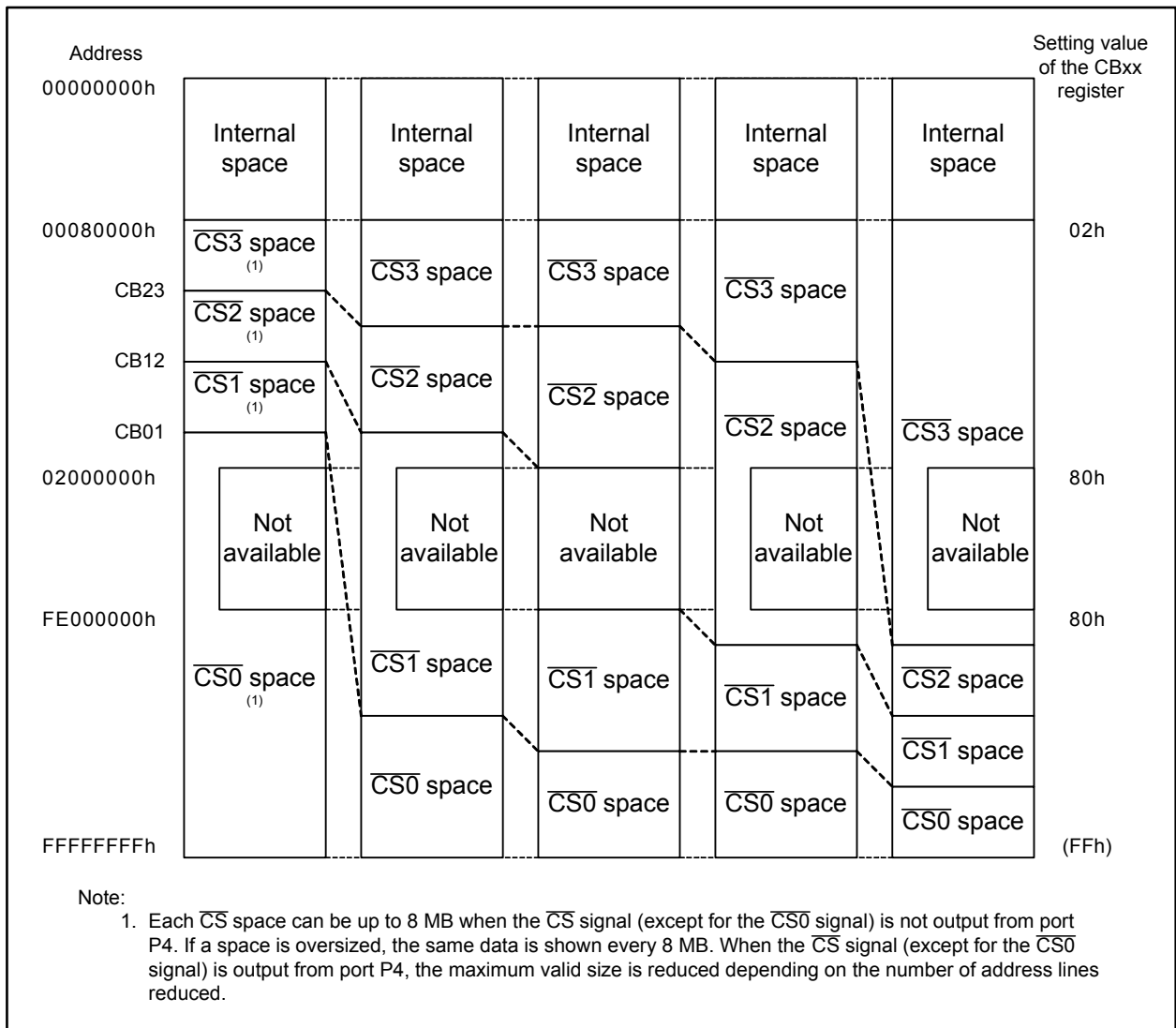


Figure 9.9 Chip Select Spaces in Memory Expansion Mode



**Figure 9.10 Chip Select Spaces in Microprocessor Mode**

### 9.3.2 External Data Bus Width Setting

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

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

Figure 9.11 shows registers EBC0 to EBC3.

**External Bus Control Register i (i = 0 to 3) <sup>(1)</sup>**

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

Bit Symbol	Bit Name	Function	RW
ESUR0	Address Setup Cycles Before Read Setting Bit <sup>(2)</sup>	b1 b0 0 0 : Do not use this combination 0 1 : <i>sur</i> = 1 1 0 : <i>sur</i> = 2 1 1 : <i>sur</i> = 3	RW
EWR0	Read Pulse Width Setting Bit <sup>(2)</sup>	b3 b2 0 0 : <i>wr</i> = 1 0 1 : <i>wr</i> = 2 1 0 : <i>wr</i> = 3 1 1 : <i>wr</i> = 4	RW
— (b4)	Reserved	Should be written with 1	RW
RDY	$\overline{\text{RDY}}$ Monitor Bit	0: Ignore RDY 1: Use RDY	RW
MPY0	Multiplied Cycle Setting Bit <sup>(2)</sup>	b7 b6 0 0 : <i>mpy</i> = 1 0 1 : <i>mpy</i> = 2 1 0 : <i>mpy</i> = 3 1 1 : <i>mpy</i> = 4	RW
ESUW0	Address Setup Cycles Before Write Setting Bit <sup>(2)</sup>	b9 b8 0 0 : Do not use this combination 0 1 : <i>suw</i> = 1 1 0 : <i>suw</i> = 2 1 1 : <i>suw</i> = 3	RW
EWW0	Write Pulse Width Setting Bit <sup>(2)</sup>	b11b10 0 0 : <i>ww</i> = 1 0 1 : <i>ww</i> = 2 1 0 : <i>ww</i> = 3 1 1 : <i>ww</i> = 4	RW
— (b12)	Reserved	Should be written with 1	RW
MPX	External Bus Format Select Bit	0: Separate bus 1: Multiplexed bus	RW
BW0	External Bus Width Setting Bit <sup>(3)</sup>	0: 8-bit width 1: 16-bit width	RW
— (b15)	Reserved	Should be written with 0	RW

Notes:

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

**Figure 9.11 Registers EBC0 to EBC3**

### 9.3.3 Separate Bus/Multiplexed Bus Selection

The bus format is selectable between separate bus format and multiplexed bus format. The bus format for each space is selected by setting the MPX bit in registers EBC0 to EBC3. To select the multiplexed bus format for all spaces, the EXPMX bit in the PBC register should be set to 1 (multiplexed bus in all spaces). In this case, ports P0, P1, and P4\_0 to P4\_3 can be used as programmable I/O ports.

#### (1) Separate Bus

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

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

When the EXBW0 bit in the PBC register is 0 (8-bit width), port P0 is the data bus, and port P1 is programmable I/O port.

When the EXBW0 bit is 1 (16-bit width), ports P0 and P1 are data buses. Note that port P1 (D8 to D15) becomes undefined if the MCU accesses an space where the BW0 bit is to 0 (8-bit width).

#### (2) Multiplexed Bus

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

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

When the BW0 bit in registers EBC0 to EBC3 is 0 (8-bit width), D0 to D7 are multiplexed with A0 to A7. When the BW0 bit is 1 (16-bit width), D0 to D15 are multiplexed with  $\overline{BC0}$ , A1 to A15.

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

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

**Table 9.2 Processor Mode and Pin Functions**

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

Note:

1. An undefined value is output.

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

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

Note:

1. An undefined value is output.



### 9.3.4 Read and Write Signals

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

When the EXBW0 bit in the PBC register is 0 (8-bit width), the PM02 bit should be set to 0 ( $\overline{RD}/\overline{WR}/\overline{BC0}/\overline{BC1}$ ). When accessing an 8-bit space while the EXBW0 bit is 1 (16-bit width), the combination of  $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{BC0}$ , and  $\overline{BC1}$  is selected irrespective of the PM02 bit setting.

Tables 9.4 and 9.5 list the operation of each signal.

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

**Table 9.4**  $\overline{RD}$ ,  $\overline{WR0}$ , and  $\overline{WR1}$  Signals

Data Bus Width	$\overline{RD}$	$\overline{WR0}$	$\overline{WR1}$	External Data Bus Status
16 bits	L	H	H	Read 2-byte data
	H	L	H	Write 1-byte data to even address
	H	H	L	Write 1-byte data to odd address
	H	L	L	Write 2-byte data to both even and odd addresses
8 bits	L	H ( $\overline{WR}$ )	—	Read 1-byte data
	H	L ( $\overline{WR}$ )	—	Write 1-byte data

**Table 9.5**  $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{BC0}$ , and  $\overline{BC1}$  Signals

Data Bus Width	$\overline{RD}$	$\overline{WR}$	$\overline{BC0}$	$\overline{BC1}$	External Data Bus Status
16 bits	L	H	L	L	Read 2-byte data
	H	L	L	H	Write 1-byte data to even address
	H	L	H	L	Write 1-byte data to odd address
	H	L	L	L	Write 2-byte data to both even and odd addresses
8 bits	L	H	H/L (A0)	—	Read 1-byte data
	H	L	H/L (A0)	—	Write 1-byte data

### 9.3.5 External Bus Timing

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

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

**Table 9.6  $T_{su}(A-R)$  and Bit Settings: MPY1, MPY0, ESUR1, and ESUR0 (unit: cycles)**

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

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

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

Note:

1. Do not set this value.

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

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

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

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

Note:

1. Do not set this value.

Figure 9.12 and 9.13 show examples of external bus timing in separate bus format (the MPX bit is set to 0) and in multiplexed bus format (the MPX bit is set to 1), respectively.

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

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

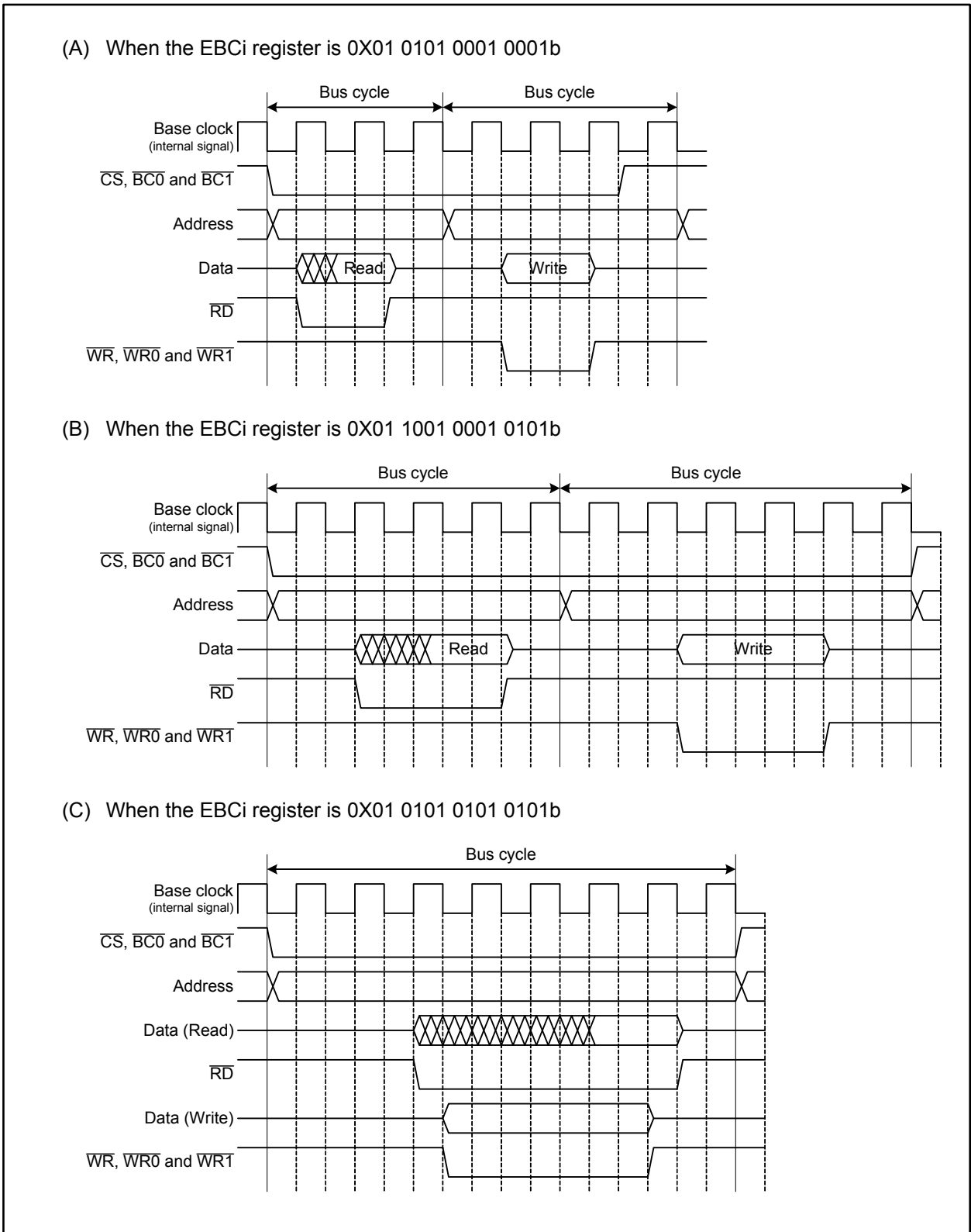


Figure 9.12 External Bus Timing in Separate Bus Format (i = 0 to 3)

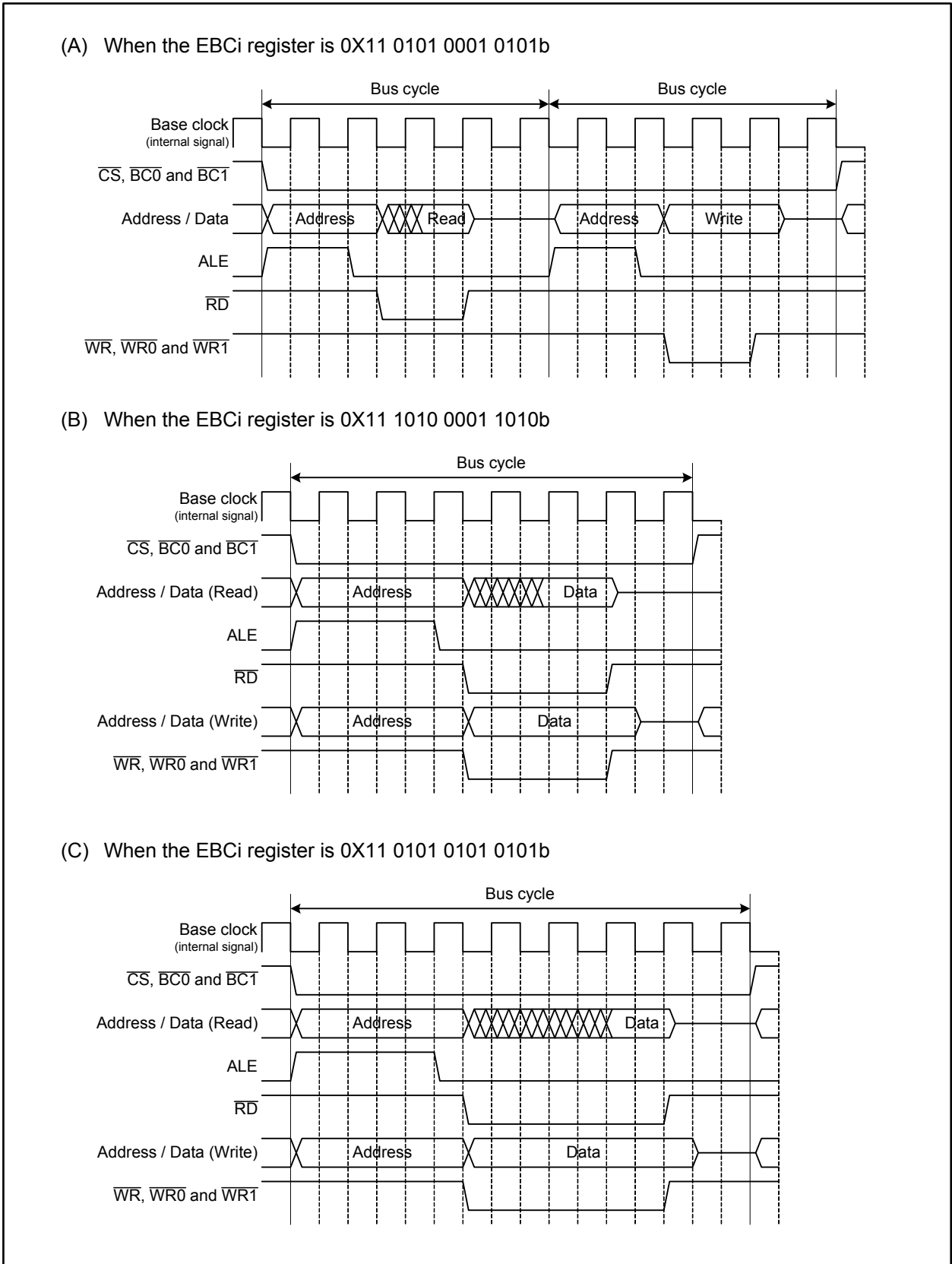
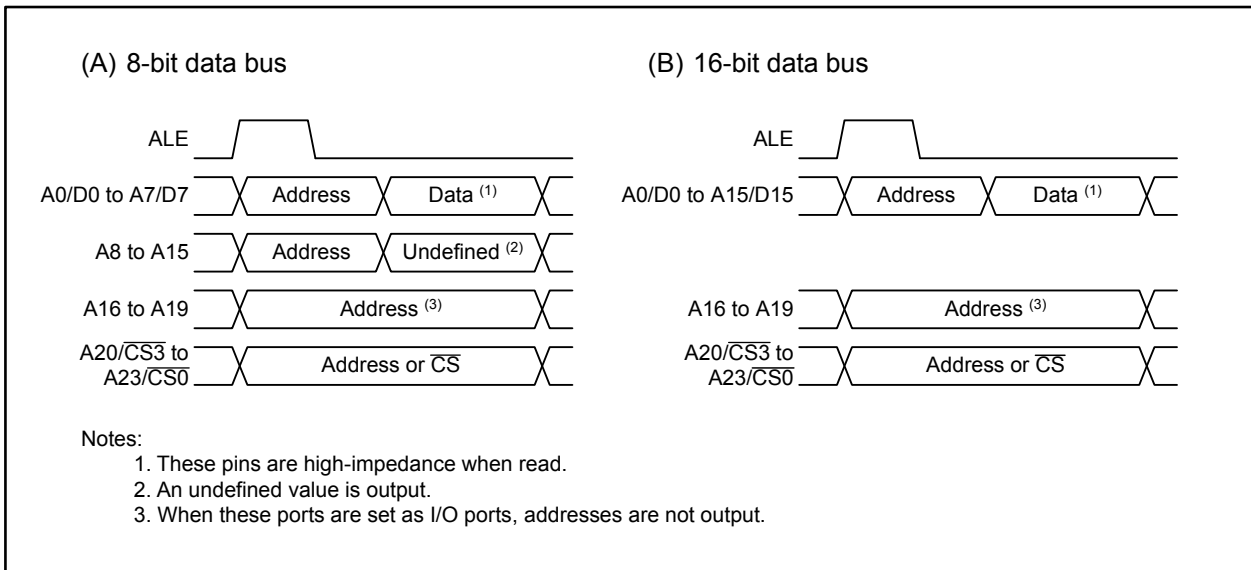


Figure 9.13 External Bus Timing in Multiplexed Bus Format (i = 0 to 3)

### 9.3.6 ALE Signal

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



**Figure 9.14 ALE Signal and Address Bus/Data Bus**

The ALE signal becomes high when a bus cycle is started and changes to low at 1/2 base clock before  $\overline{RD}$  or  $\overline{WR}$  becomes low.

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

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

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

Since the base clock is not output to external pins, drive the  $\overline{\text{RDY}}$  signal low when the  $\overline{\text{RD}}$ ,  $\overline{\text{WR}}$ , and  $\overline{\text{WR0}}$  to  $\overline{\text{WR1}}$  signals are held in a low level, and drive the  $\overline{\text{RDY}}$  signal high synchronizing the rise of the BCLK signal.

Figure 9.15 shows an example of  $\overline{\text{RDY}}$  signal generator and Table 9.10 lists setting conditions of registers EBC0 to EBC3 to use this circuit. Figure 9.16 shows examples of bus cycle that is extended by the  $\overline{\text{RDY}}$  signal.

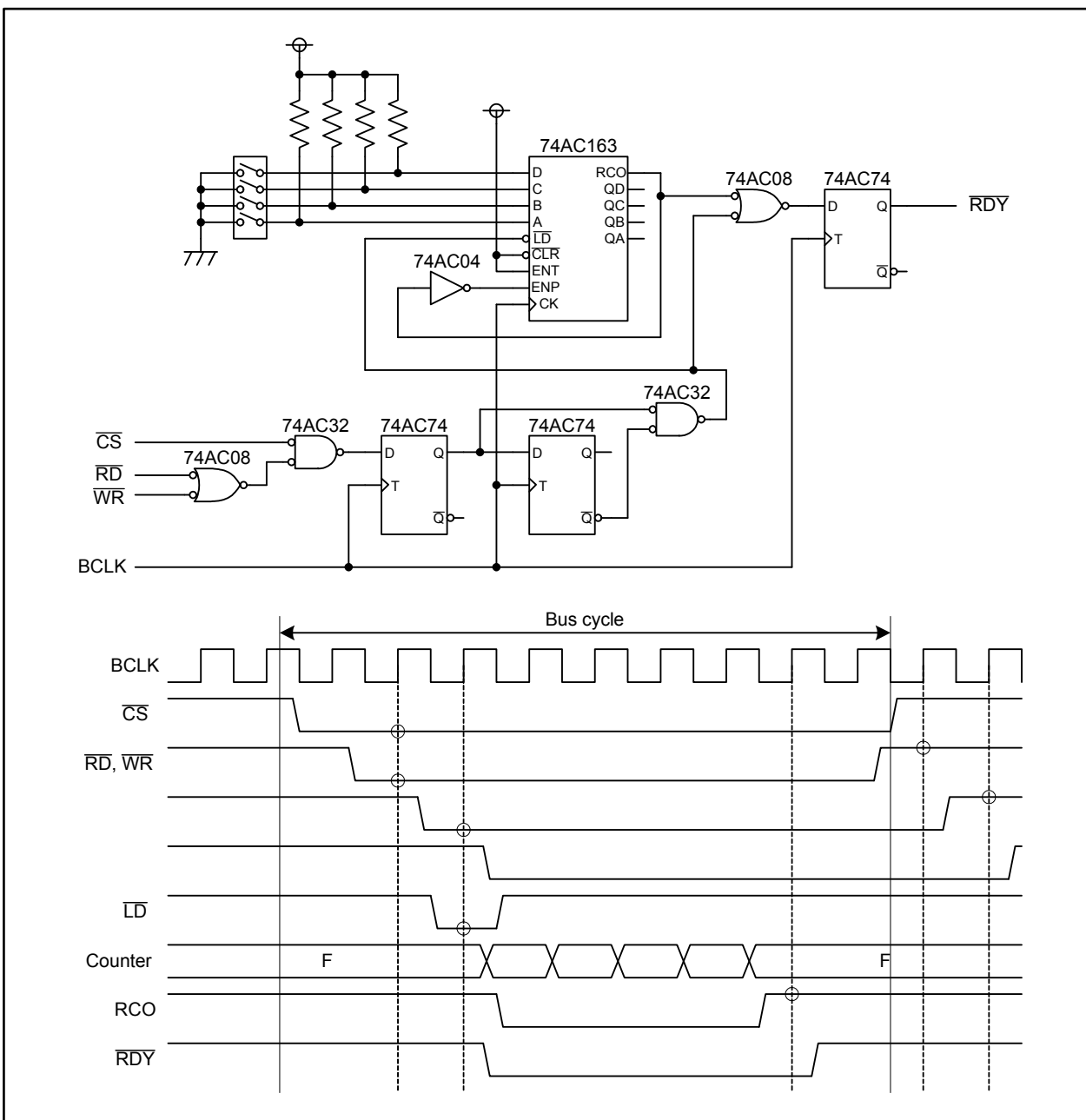


Figure 9.15  $\overline{\text{RDY}}$  Signal Generation Circuitry

**Table 9.10 EBCi Register Setting Conditions when Using the Circuit in Figure 9.15 (i = 0 to 3)**

Peripheral Bus Clock Frequency	Setting Condition	Setting Example
BCLK = 1/2 base clock	$mpy = 3$ In separate bus format $\overline{RD}$ pulse width $\geq 9.5$ $\overline{WR}$ pulse width $\geq 11.5$ $\overline{RD}/\overline{WR}$ high level width $\geq 2.5$ In multiplexed bus format $\overline{RD}$ pulse width $\geq 11.5$ $\overline{WR}$ pulse width $\geq 11.5$	In separate bus format EBCi = 0X01 1101 1011 1001b etc.  In multiplexed bus format EBCi = 0X11 1101 1011 1101b etc.
BCLK = 1/3 base clock	$mpy = 3$ In separate bus format $\overline{RD}$ pulse width $\geq 12.5$ $\overline{WR}$ pulse width $\geq 11.5$ $\overline{RD}/\overline{WR}$ high level width $\geq 3.5$ In multiplexed bus format $\overline{RD}$ pulse width $\geq 11.5$ $\overline{WR}$ pulse width $\geq 11.5$	In separate bus format EBCi = 0X01 1101 1011 1101b etc.  In multiplexed bus format EBCi = 0X11 1101 1011 1101b etc.
BCLK = 1/4 base clock	$mpy = 4$ In separate bus format $\overline{RD}$ pulse width $\geq 20.5$ $\overline{WR}$ pulse width $\geq 19.5$ $\overline{RD}/\overline{WR}$ high level width $\geq 4.5$ In multiplexed bus format $\overline{RD}$ pulse width $\geq 19.5$ $\overline{WR}$ pulse width $\geq 19.5$	In separate bus format Not available  In multiplexed bus format Not available

X: Given value



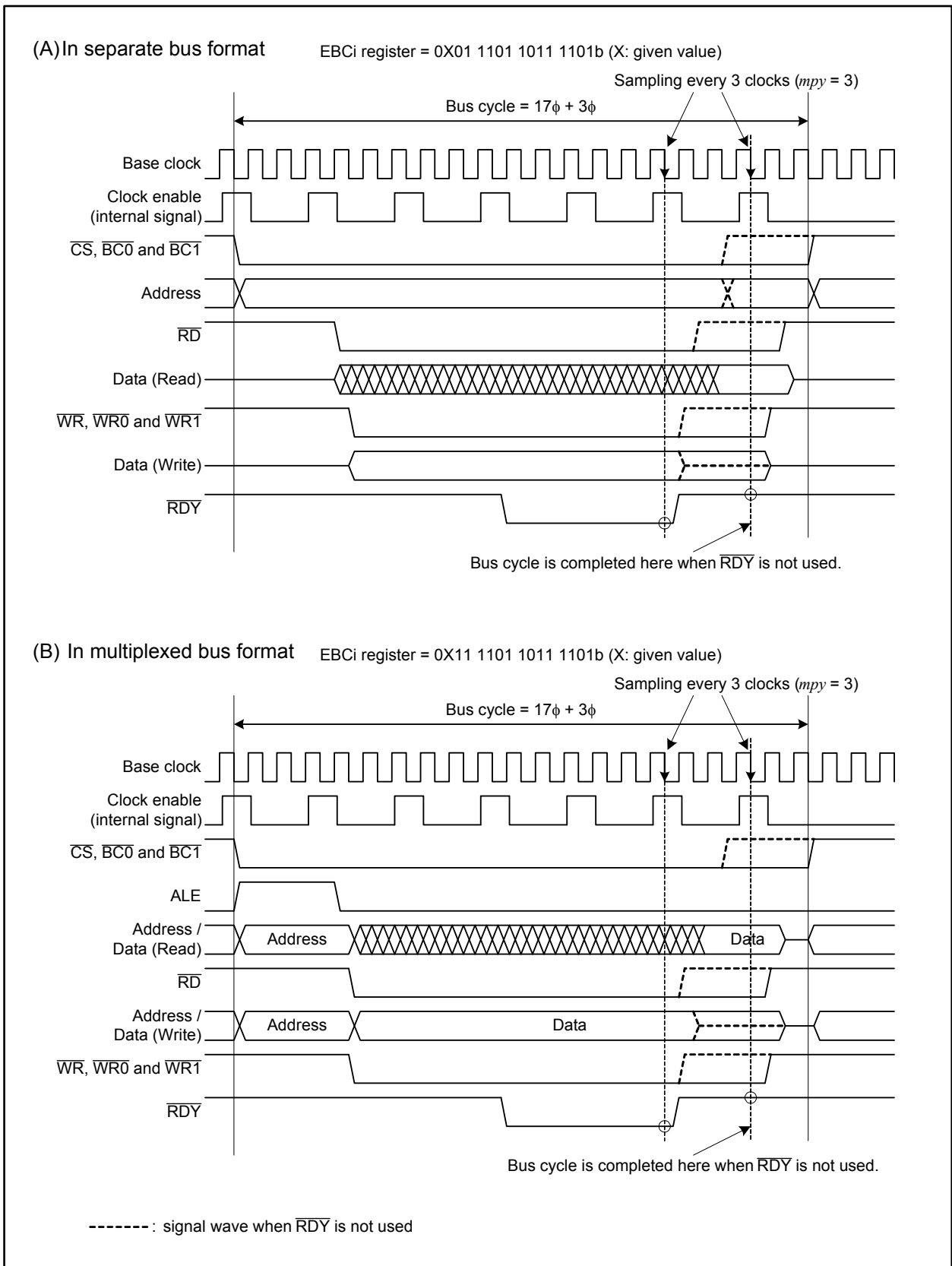


Figure 9.16 An Example of Bus Cycle Extended by  $\overline{RDY}$  Signal ( $f(\text{BCLK}) = 1/2 f(\text{Base})$ ) ( $i = 0$  to 3)

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

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

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

Table 9.11 lists the MCU state in a hold state.

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

**Table 9.11 MCU State in Hold State**

Item	State
Oscillation	On
Address bus, data bus, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ , $\overline{\text{BC0}}$ to $\overline{\text{BC1}}$	High-impedance
$\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{WR0}}$ to $\overline{\text{WR1}}$	High-impedance
Programmable I/O port	The state when $\overline{\text{HOLD}}$ was received is held
$\overline{\text{HLDA}}$ pin	Low is output
Internal peripheral circuit	On (excluding the watchdog timer)
ALE pin	Low is output

### 9.3.9 BCLK Output

The BCLK, which has the same frequency as peripheral bus clock, is a divided clock derived from the PLL clock. In memory expansion mode or microprocessor mode, BCLK is output from port P5\_3 when the PM07 bit in the PM0 register is set to 0 (output BCLK) and bits CM01 and CM00 in the CM0 register are set to 00b (I/O port P5\_3). In single-chip mode, BCLK cannot be output. Refer to 8. "Clock Generator" for details.

## 9.4 External Bus State when Accessing Internal Space

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

**Table 9.12 External Bus State when Accessing Internal Space**

Pin		Pin State when Accessing SFR	Pin State when Accessing Internal Memory
Address bus		Address is output	The address of an SFR or external space last accessed is held
Data bus	Read cycle	High-impedance	High-impedance
	Write cycle	Data is output	Undefined
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$		High is output	High is output
$\overline{\text{BC0}}$ to $\overline{\text{BC1}}$		$\overline{\text{BC0}}$ to $\overline{\text{BC1}}$ are output	The address of SFR or external space last accessed is held
$\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{WR0}}$ to $\overline{\text{WR1}}$		$\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{WR0}}$ to $\overline{\text{WR1}}$ are output	High is output
ALE		The ALE signal is output	The ALE signal is output

## 9.5 Notes on Bus

### 9.5.1 Notes on Designing a System

When a flash memory rewrite is performed in CPU rewrite mode using memory expansion mode, the use of  $\overline{CS0}$  space and  $\overline{CS3}$  space has the following restrictions:

- If the FEBC0 and/or FEBC3 registers are set in CPU rewrite mode, the bus format for the corresponding space functions as separate bus. Any external devices connected in multiplexed bus format become inaccessible.
- If the FEBC0 and/or FEBC3 registers are set in CPU rewrite mode, the bus timing for the corresponding space changes. This may cause external devices to become inaccessible depending on the register settings.

Devices required to be accessed in CPU rewrite mode should be allocated in  $\overline{CS1}$  space and/or  $\overline{CS2}$  space.

### 9.5.2 Notes on Register Settings

#### 9.5.2.1 Chip Select Boundary Select Registers

When not using memory expansion mode, do not change values after a reset for registers CB01, CB12, and CB23.

When using memory expansion mode, set all of these registers to a value within the specified range whether or not each chip select space is used.

#### 9.5.2.2 External Bus Control Registers

Registers EBC0 and EBC3 share respective addresses with registers FEBC0 and FEBC3. If the FEBC0 and/or FEBC3 registers are set while the flash memory is being rewritten, set the EBC0 and/or EBC3 registers again after rewriting the flash memory.

## 10. Protection

This function protects important registers from being easily overwritten when a program goes out of control. Registers used to protect other registers from being rewritten are as follows: PRCR, PRCR2 to PRCR3, and PRR.

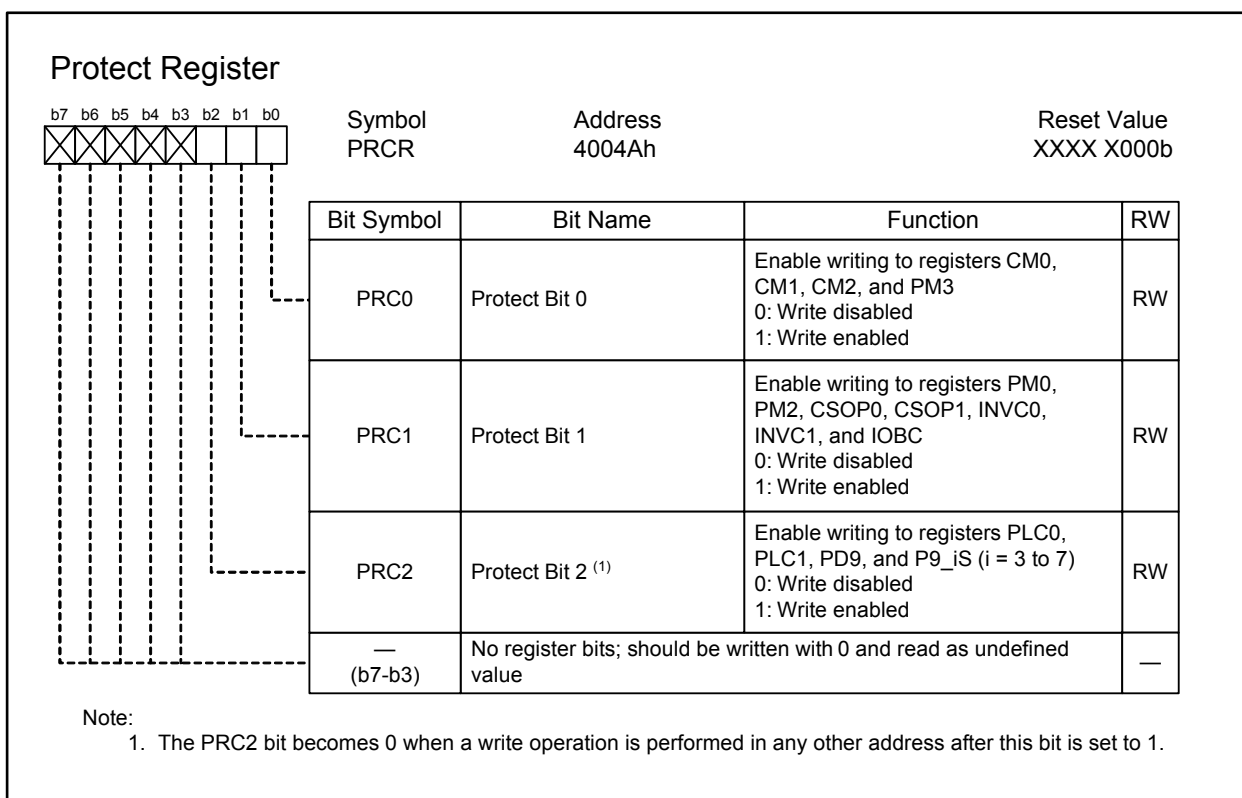
### 10.1 Protect Register (PRCR Register)

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

**Table 10.1 Registers Protected by the PRCR Register**

Bit	Protected Registers
PRC0	CM0, CM1, CM2, and PM3
PRC1	PM0, PM2, CSOP0, CSOP1, INVC0, INVC1, and IOBC
PRC2	PLC0, PLC1, PD9, and P9_iS (i = 3 to 7)

The PRC2 bit becomes 0 (write disabled) when a write operation is performed in any other address after this bit is set to 1 (write enabled). Set the PRC2 bit to 1 just before rewriting registers PD9, P9\_iS, PLC0, and PLC1 (i = 3 to 7). No interrupt handling or DMA transfers should be inserted between these two instructions. Bits PRC1 and PRC0 do not become 0 even if a write operation is performed in any other address. These bits should be set to 0 by a program.



**Figure 10.1 PRCR Register**

## 10.2 Protect Register 2 (PRCR2 Register)

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

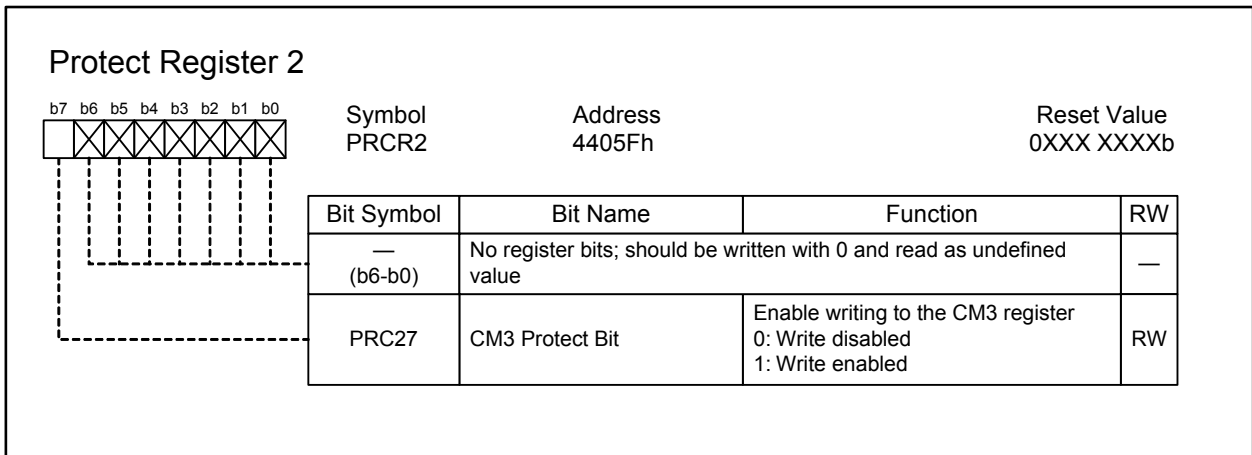


Figure 10.2 PRCR2 Register

## 10.3 Protect Register 3 (PRCR3 Register)

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

Table 10.2 Registers Protected by the PRCR3 Register

Bit	Protected Registers
PRC31	VRCCR, LVDC, and DVCR

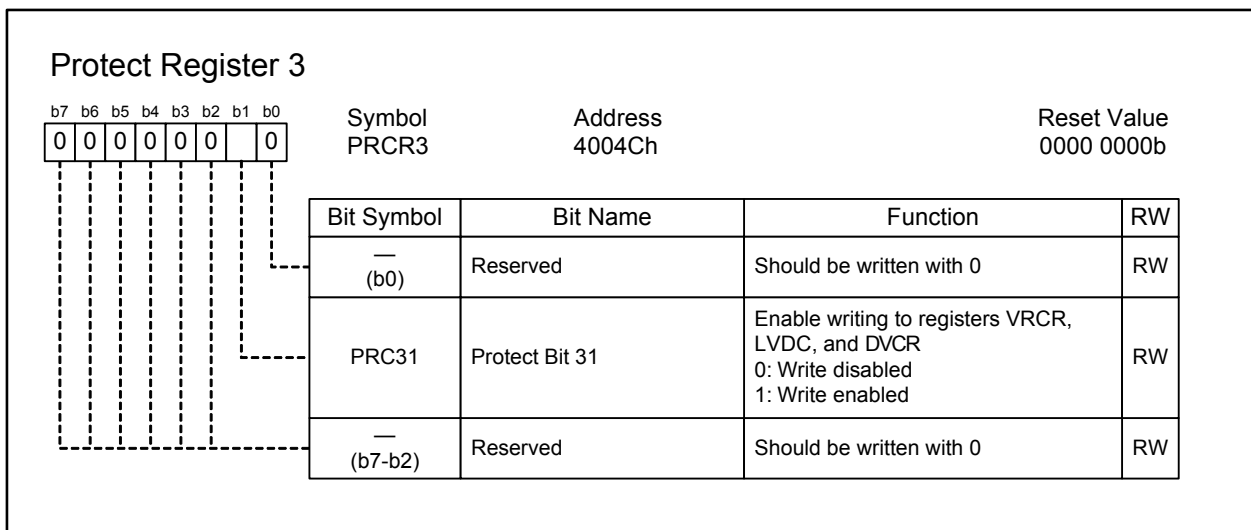
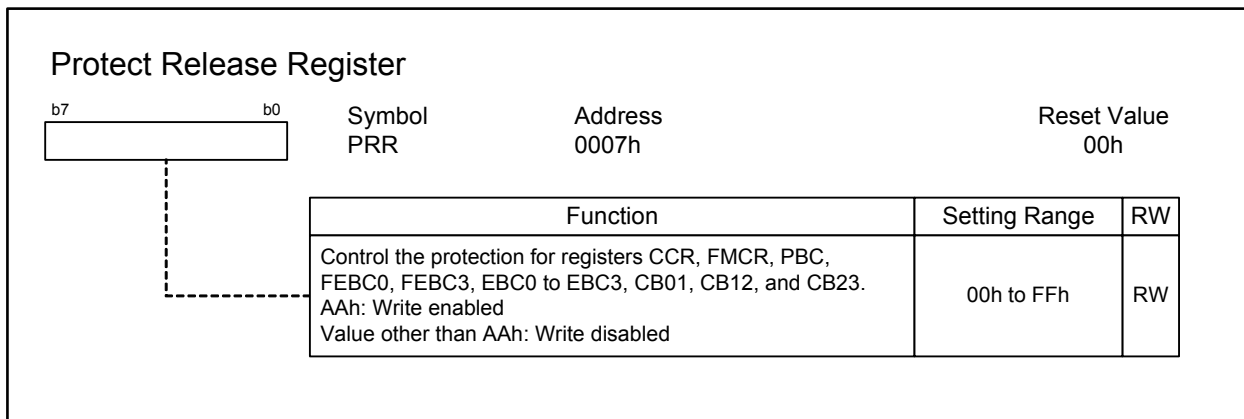


Figure 10.3 PRCR3 Register

### 10.4 Protect Release Register (PRR Register)

Figure 10.4 shows the PRR register. Registers protected by the PRR register are as follows: CCR, FMCR, PBC, FEBC0, FEBC3, EBC0 to EBC3, CB01, CB12, and CB23.

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

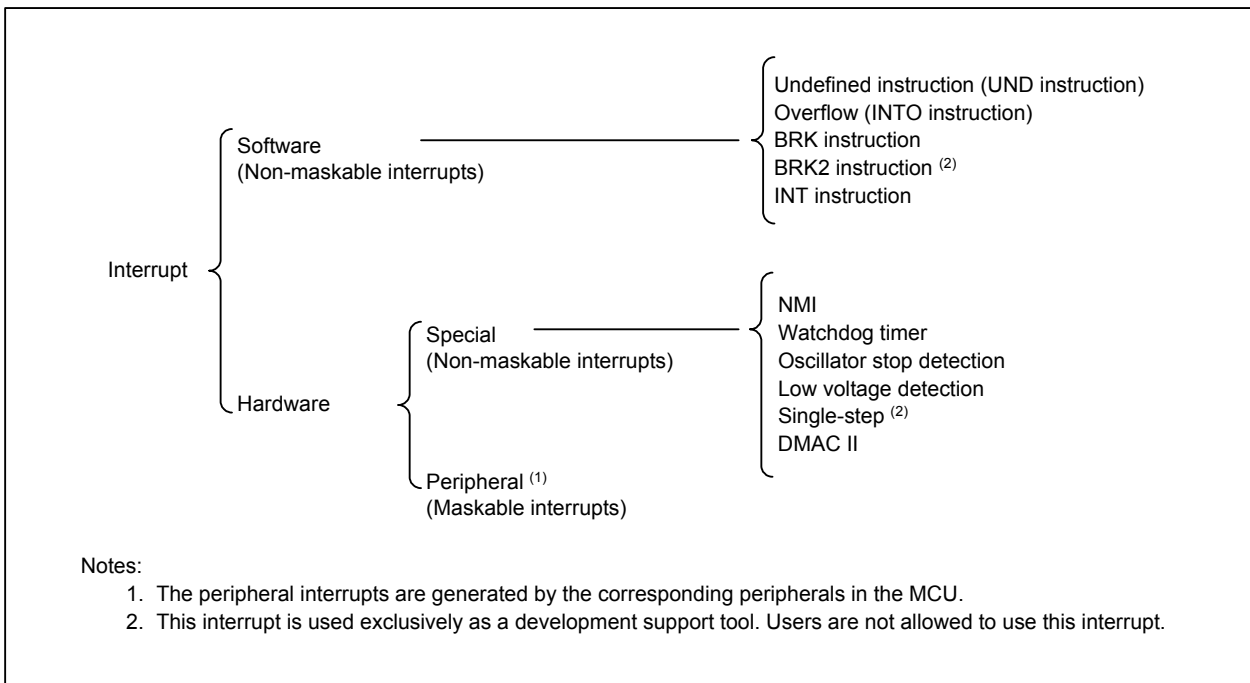


**Figure 10.4 PRR Register**

## 11. Interrupts

### 11.1 Interrupt Types

Figure 11.1 shows the types of interrupts.



**Figure 11.1 Interrupts**

Interrupts are also classified into maskable/non-maskable.

#### (1) Maskable Interrupts

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

#### (2) Non-maskable Interrupts

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

## 11.2 Software Interrupts

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

### (1) Undefined Instruction Interrupt

This interrupt occurs when the UND instruction is executed.

### (2) Overflow Interrupt

This interrupt occurs when the INTO instruction is executed while the O flag is 1. The following instructions may change the O flag to 1, depending on the operation result:

ABS, ADC, ADCF, ADD, ADDF, ADSF, CMP, CMPF, CNVIF, DIV, DIVF, DIVU, DIVX, EDIV, EDIVU, EDIVX, MUL, MULF, MULU, MULX, NEG, RMPA, ROUND, SBB, SCMPU, SHA, SUB, SUBF, SUNTIL, and SWHILE

### (3) BRK Instruction Interrupt

This interrupt occurs when the BRK instruction is executed.

### (4) BRK2 Instruction Interrupt

This interrupt occurs when the BRK2 instruction is executed.

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

### (5) INT Instruction Interrupt

This interrupt occurs when the INT instruction is executed with a selected software interrupt number from 0 to 255. Software interrupt numbers 0 to 127 are designated for peripheral interrupts. That is, the INT instruction with a software interrupt number from 0 to 127 has the same interrupt handler as that for peripheral interrupts.

The stack pointer (SP) used for this interrupt differs depending on the software interrupt numbers. For software interrupt numbers 0 to 127, when an interrupt request is accepted, the U flag is saved and set to 0 to select the interrupt stack pointer (ISP) during the interrupt sequence. The saved data of the U flag is restored upon returning from the interrupt handler. For software interrupt numbers 128 to 255, the stack pointer does not change during the interrupt sequence.



## 11.3 Hardware Interrupts

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

### 11.3.1 Special Interrupts

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

(1) NMI (Non Maskable Interrupt)

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

(2) Watchdog Timer Interrupt

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

(3) Oscillator Stop Detection Interrupt

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

(4) Low Voltage Detection Interrupt

This interrupt occurs when a low voltage input to VCC1 is detected by the voltage detector. Refer to 6.2 “Low Voltage Detector” for details.

(5) Single-step Interrupt

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

### 11.3.2 Peripheral Interrupts

Peripheral interrupts occur when an interrupt request from a peripheral in the MCU is accepted. They share the interrupt vector with software interrupt numbers 0 to 127 for the INT instruction. Peripheral interrupts are maskable.

Refer to Tables 11.2 to 11.5 for details on the interrupt sources. Refer to the relevant descriptions for details on each function.

## 11.4 Fast Interrupt

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

Use the following procedure to enable a fast interrupt:

- (1) Set the both FSIT bit in registers RIPL1 and RIPL2 to 1 (interrupt request level 7 available for fast interrupt).
- (2) Set the both DMAII bit in registers RIPL1 and RIPL2 to 0 (interrupt request level 7 available for interrupts).
- (3) Set the start address of the fast interrupt handler to the VCT register.

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

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

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

## 11.5 Interrupt Vectors

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

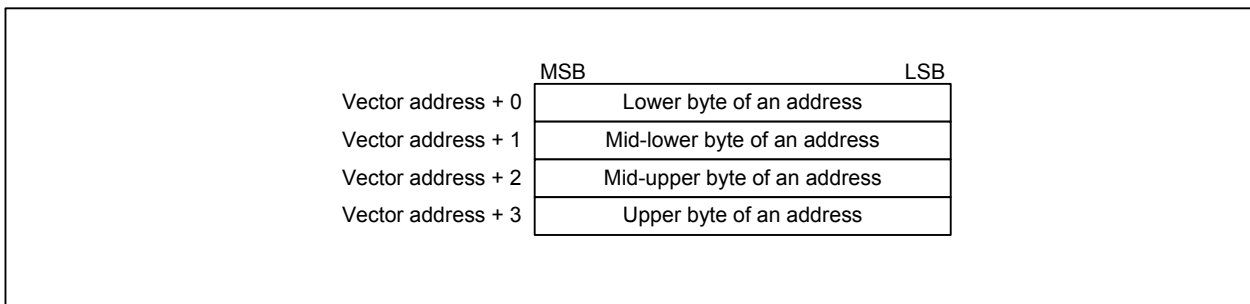


Figure 11.2 Interrupt Vector

### 11.5.1 Fixed Vector Table

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

**Table 11.1 Fixed Vector Table**

Interrupt Source	Vector Addresses (Address (L) to Address (H))	Remarks	Reference
Undefined instruction	FFFFFFDCh to FFFFFFFDFh	Interrupt by the UND instruction	R32C/100 Series Software Manual
Overflow	FFFFFFE0h to FFFFFFFE3h	Interrupt by the INTO instruction	
BRK instruction	FFFFFFE4h to FFFFFFFE7h	If address FFFFFFFE7h is FFh, a jump to the interrupt vector of software interrupt number 0 in the relocatable vector table takes place	
—	FFFFFFE8h to FFFFFFFEBh	Reserved	
—	FFFFFFECh to FFFFFFFEFh	Reserved	
Watchdog timer Oscillator stop detection Low voltage detection	FFFFFFF0h to FFFFFFFF3h	These addresses are shared by the watchdog timer interrupt, oscillator stop detection interrupt, and low voltage detection interrupt	12. "Watchdog Timer" 8. "Clock Generator" 6.2 "Low Voltage Detector"
—	FFFFFFF4h to FFFFFFFF7h	Reserved	
NMI	FFFFFFF8h to FFFFFFFBh	External interrupt by the $\overline{\text{NMI}}$ pin	
Reset	FFFFFFFCh to FFFFFFFFh		5. "Resets"

### 11.5.2 Relocatable Vector Table

The relocatable vector table occupies a 1024-byte memory space from the start address set in the INTB register. Tables 11.2 to 11.5 list the relocatable vector table entries.

An address in a multiple of 4 should be set in the INTB register for a faster interrupt sequence.

**Table 11.2 Relocatable Vector Table (1/4)**

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

## Notes:

- Each entry is relative to the base address in the INTB register.
- Interrupts from this source cannot be disabled by the I flag.
- In I<sup>2</sup>C mode, interrupts are generated by NACK, ACK, or detection of a START condition/STOP condition.
- The IFSR16 bit in the IFSR1 register selects either the interrupt source in UART5 or UART6.

**Table 11.3 Relocatable Vector Table (2/4)**

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

**Notes:**

- Each entry is relative to the base address in the INTB register.
- In I<sup>2</sup>C mode, interrupts are generated by NACK, ACK, or detection of a START condition/STOP condition.
- The IFSR06 bit in the IFSR0 register selects either the interrupt source in UART0 or UART3. The IFSR07 bit selects either the interrupt source in UART1 or that in UART4.

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

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

## Notes:

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

**Table 11.5 Relocatable Vector Table (4/4) (1)**

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

## Notes:

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

## 11.6 Interrupt Request Acceptance

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

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

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

The following section describes these flag and bits.

### 11.6.1 I Flag and IPL

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

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

Table 11.6 lists interrupt request levels classified by the IPL.

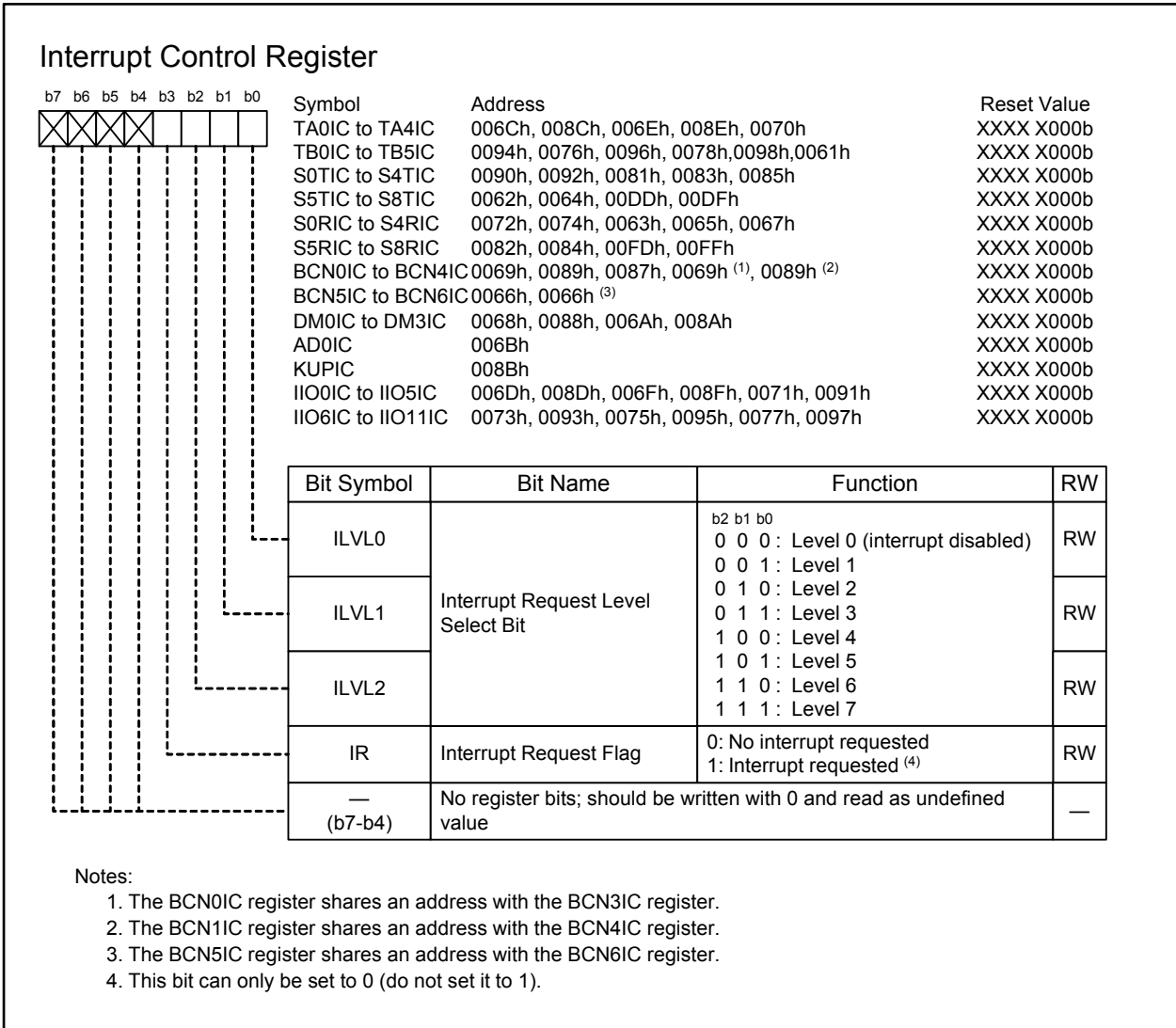
**Table 11.6 Acceptable Interrupt Request Levels and IPL**

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

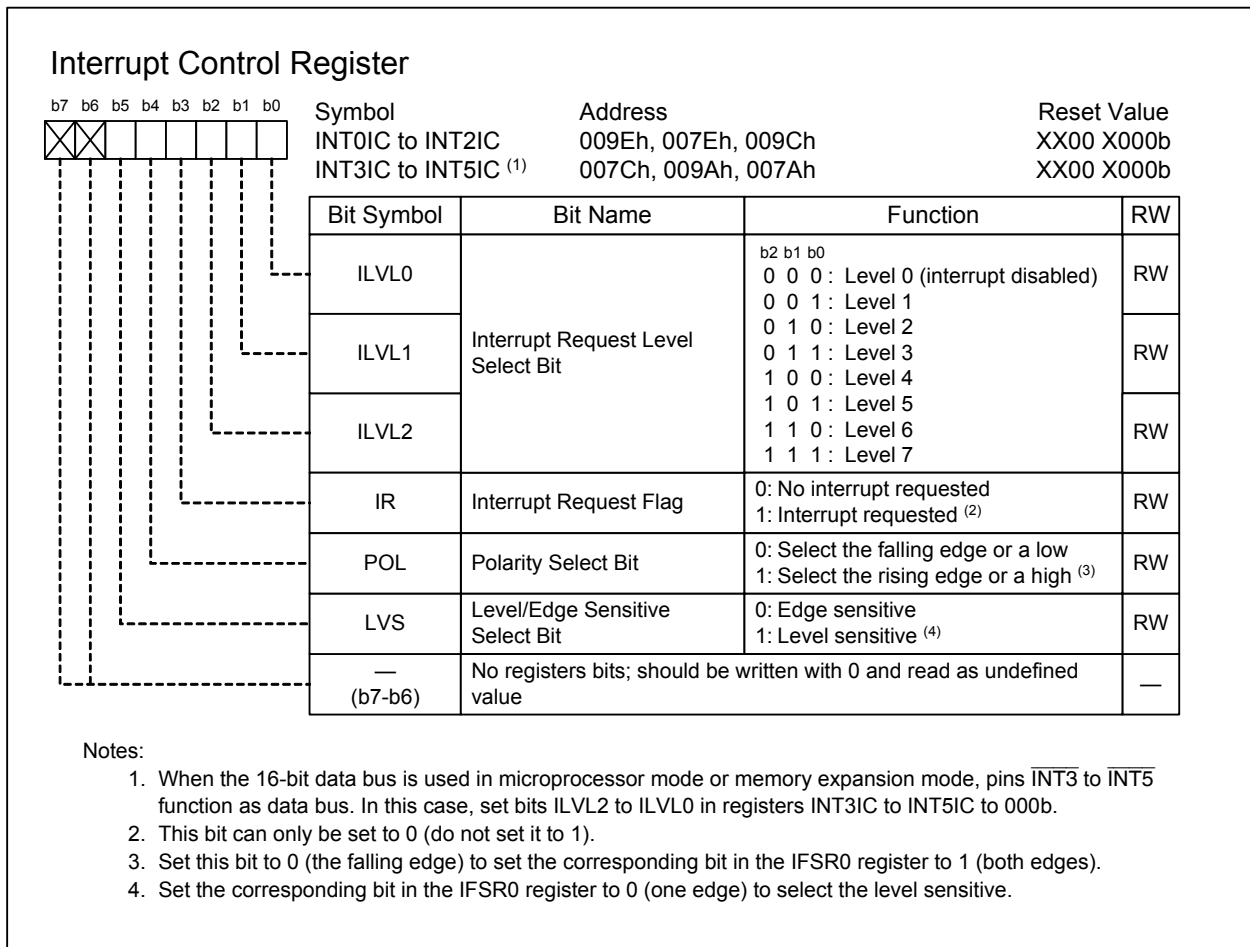


## 11.6.2 Interrupt Control Registers

Each peripheral interrupt is controlled by an interrupt control register. Figures 11.3 and 11.4 show the interrupt control registers.



**Figure 11.3 Interrupt Control Register (1/2)**



**Figure 11.4 Interrupt Control Register (2/2)**

#### Bits ILVL2 to ILVL0

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

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

#### IR bit

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

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

When rewriting the interrupt control register, no corresponding interrupt request should be generated. If there is a possibility that an interrupt request may be generated, disable the interrupt request before rewriting the register.

When enabling an interrupt immediately after changing the interrupt control register, insert NOPs between two instructions or perform a dummy read of the interrupt control register so that the interrupt enable flag (I flag) cannot become 1 (interrupt enabled) before writing to the interrupt control register is completed.

If an interrupt request is generated for the register being rewritten, the IR bit may not become 1 depending on the instruction being used. If it matters, use one of the following instructions to rewrite the register:

- AND
- OR
- BCLR
- BSET

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

### 11.6.3 Wake-up IPL Setting Register

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

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

Figure 11.5 shows registers RIPL1 and RIPL2.

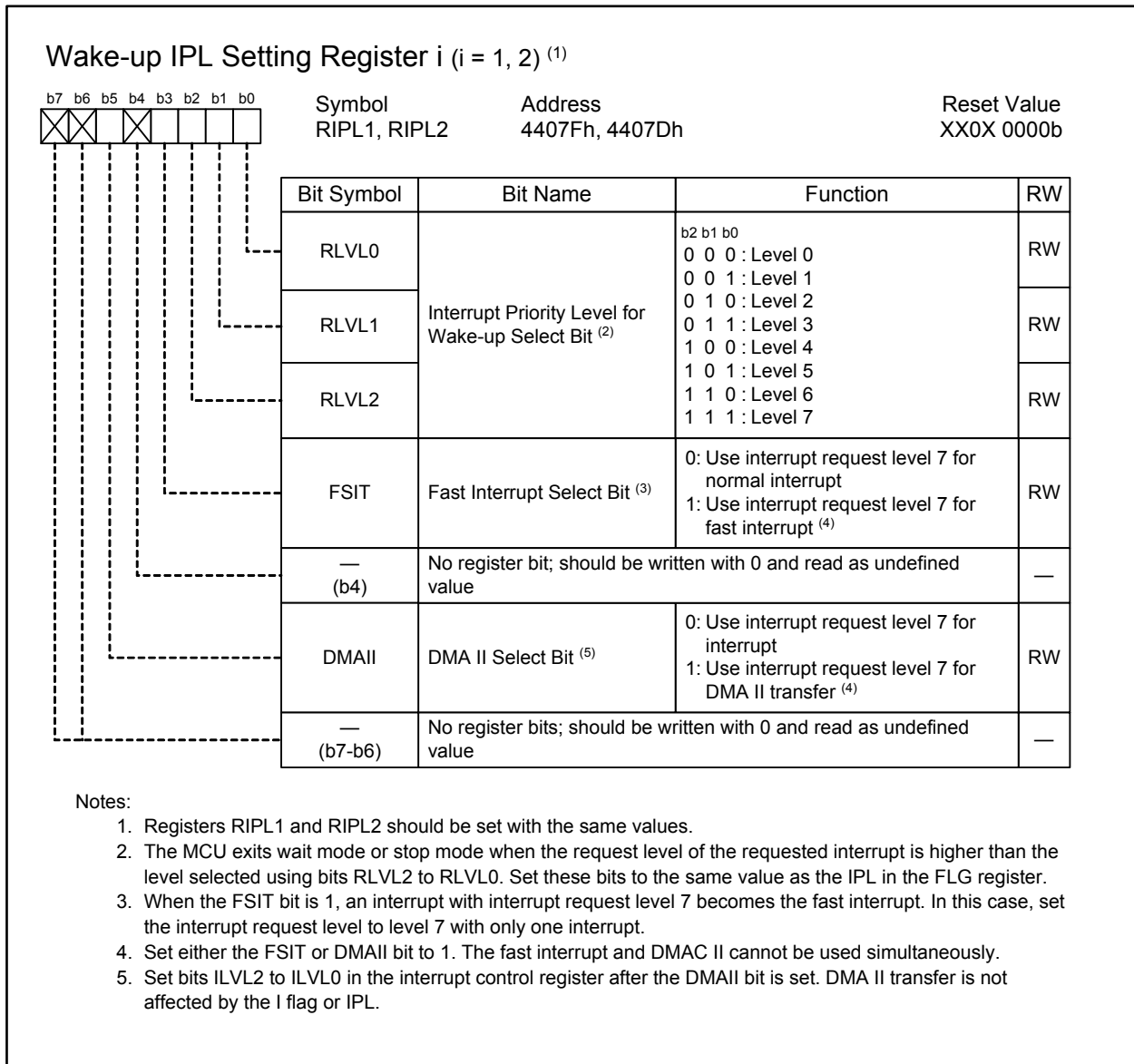


Figure 11.5 Registers RIPL1 and RIPL2

### 11.6.4 Interrupt Sequence

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

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

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

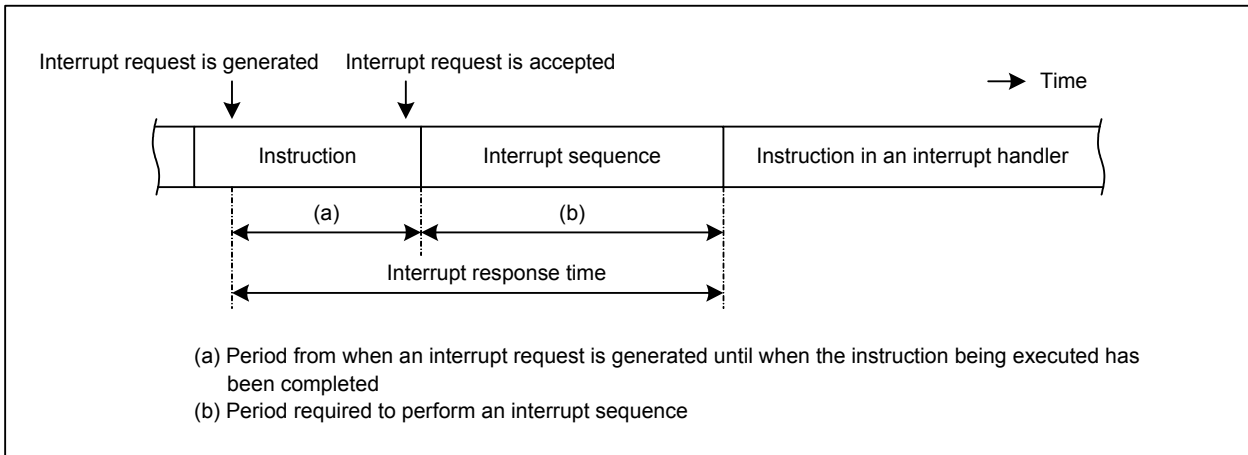
The interrupt sequence is as follows:

- (1) The CPU acknowledges the interrupt request to obtain the interrupt information (the interrupt number, and the interrupt request level) from the interrupt controller. Then the corresponding IR bit becomes 0 (no interrupt requested).
- (2) The FLG register value before the interrupt sequence is stored to a temporary register in the CPU. The temporary register is inaccessible to users.
- (3) The following bits in the FLG register become 0:
  - The I flag (interrupt enable flag): interrupt disabled
  - The D flag (debug flag): single-step interrupt disabled
  - The U flag (stack pointer select flag): ISP selected
- (4) The temporary register value in the CPU is saved to the stack, or to the SVF register in case of the fast interrupt.
- (5) The PC value is saved to the stack, or to the SVP register in case of the fast interrupt.
- (6) The interrupt request level for the accepted interrupt is set in the IPL (processor interrupt priority level).
- (7) The corresponding interrupt vector is read from the interrupt vector table.
- (8) This interrupt vector is stored into the PC.

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

### 11.6.5 Interrupt Response Time

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



**Figure 11.6** Interrupt Response Time

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

Period (b) is listed in Table 11.7.

**Table 11.7** Interrupt Sequence Execution Time <sup>(1)</sup>

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

Notes:

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

### 11.6.6 IPL after Accepting an Interrupt Request

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

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

**Table 11.8 Interrupts without Interrupt Request Level and IPL**

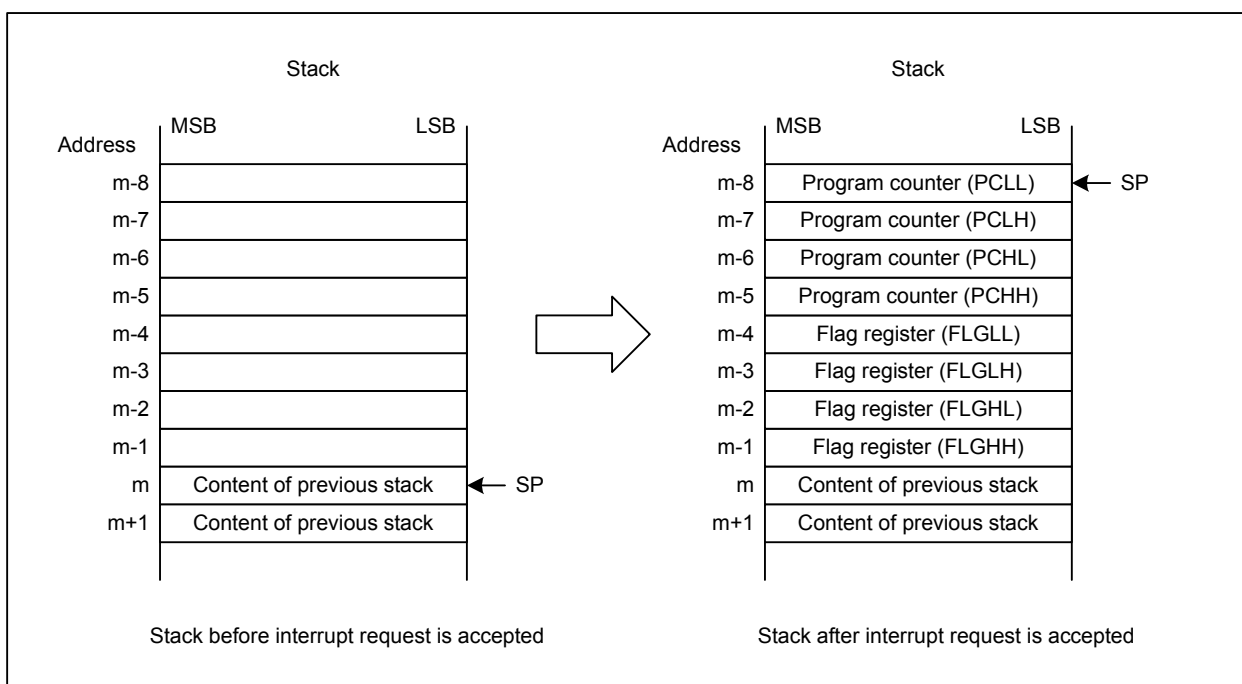
Interrupt Sources without Interrupt Request Level	IPL Value to be Set
NMI, watchdog timer, oscillator stop detection, low voltage detection	7
Reset	0
Software	Unchanged

### 11.6.7 Register Saving

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

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

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



**Figure 11.7 Stack Before and After an Interrupt Request is Accepted**

## 11.7 Register Restoring from Interrupt Handler

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

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

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

## 11.8 Interrupt Priority

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

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

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

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

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

## 11.9 Priority Resolver

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

Figure 11.8 shows the priority resolver.



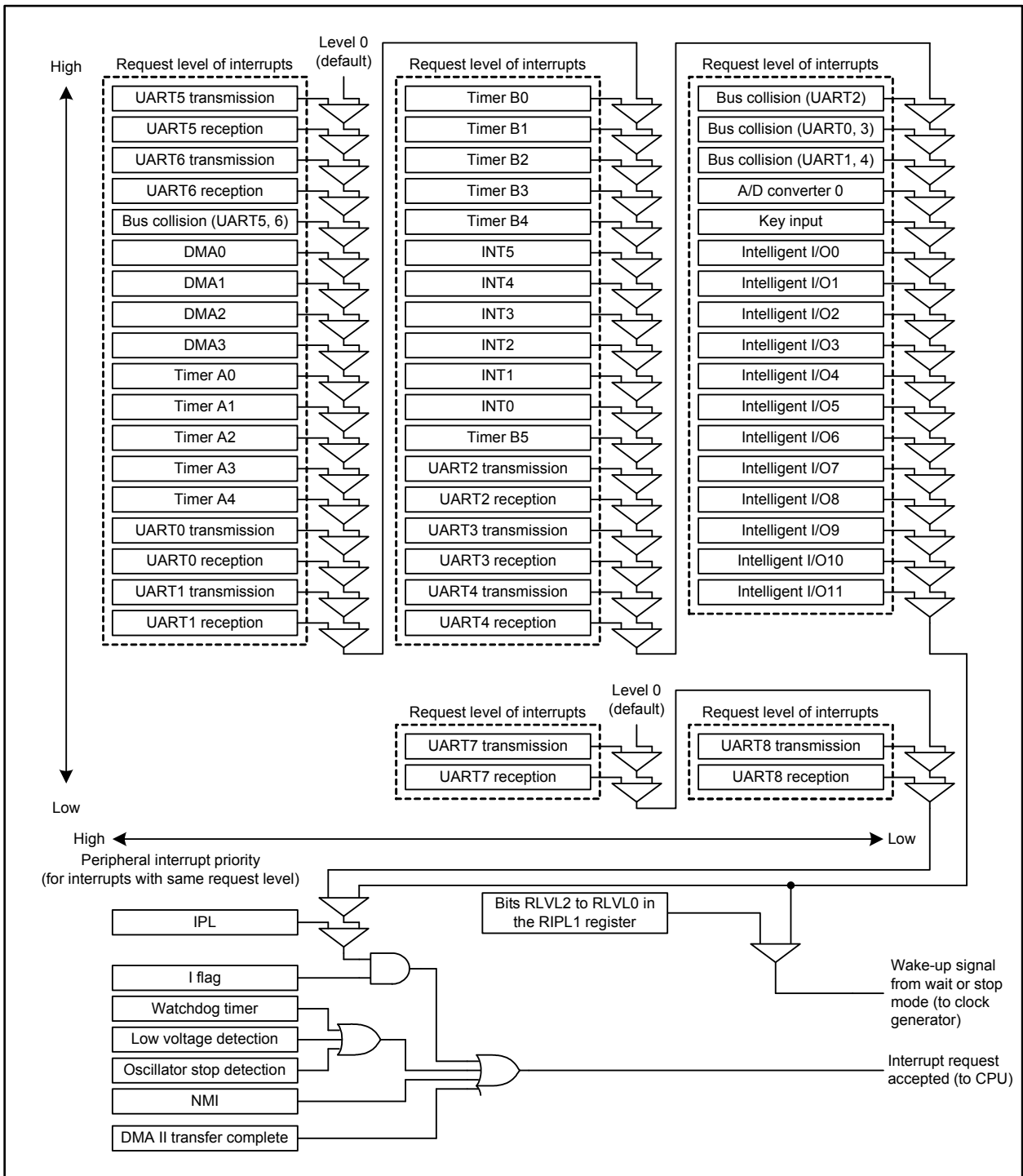


Figure 11.8 Priority Resolver

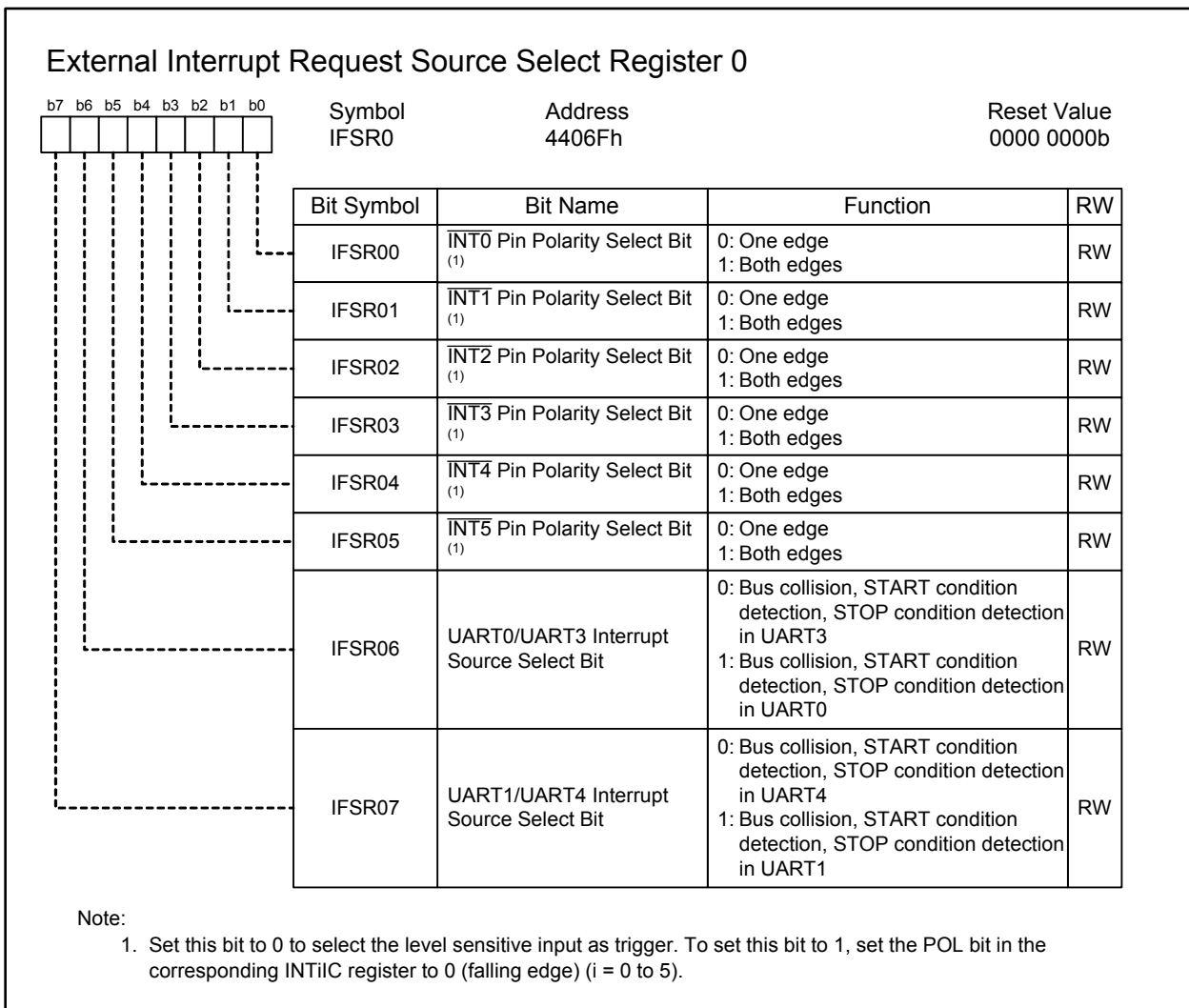
## 11.10 External Interrupt

An external interrupt occurs by an external input applied to the  $\overline{\text{INT}}_i$  pin ( $i = 0$  to 5). Set the LVS bit in the INTiIC register to select whether an interrupt is triggered by the effective edge(s) (edge sensitive), or by the effective level (level sensitive) of the input signal. The polarity of the input signal is selected by setting the POL bit in the same register.

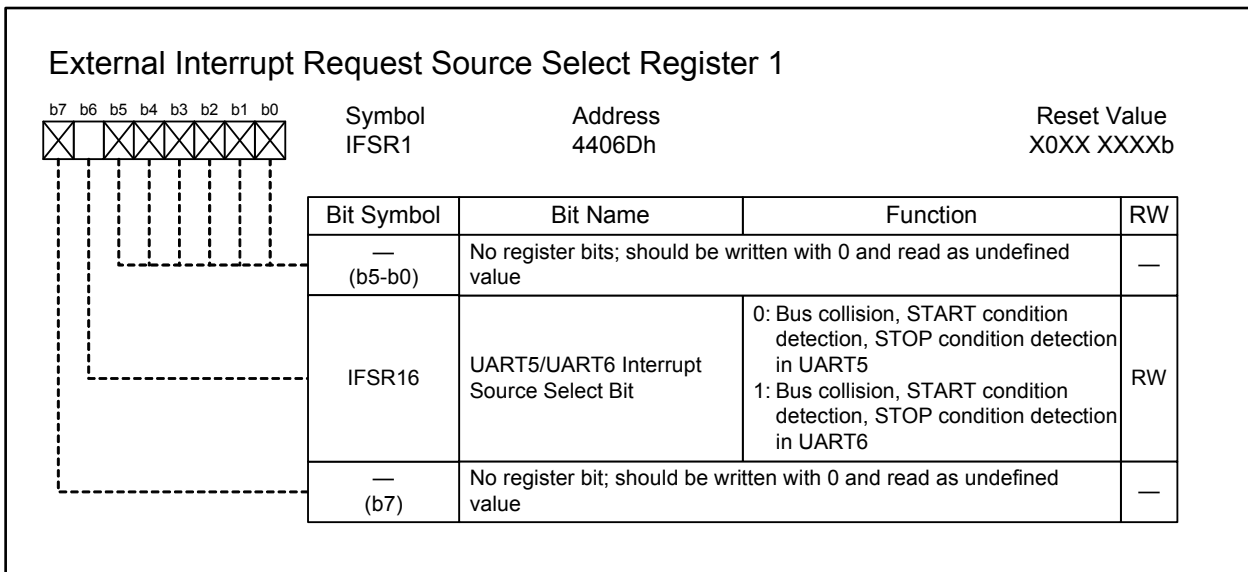
When using edge-triggered interrupts, setting the IFSR0i bit in the IFSR0 register to 1 (both edges) causes interrupt requests to be generated on both rising and falling edges of the external input. Set the POL bit in the corresponding register to 0 (falling edge) to set the IFSR0i bit to 1.

When using level-triggered interrupts, set the IFSR0i bit to 0 (one edge). When an effective level, which is selected by the POL bit, is detected on the  $\overline{\text{INT}}_i$  pin, the IR bit in the INTiIC register becomes 1. The IR bit does not become 0 even if the signal level at the  $\overline{\text{INT}}_i$  pin changes. This bit becomes 0 when the INTi interrupt is accepted or it is set to 0 by a program.

Figure 11.9 shows the IFSR0 register.



**Figure 11.9 IFSR0 Register**



**Figure 11.10 IFSR1 Register**

### 11.11 NMI

The NMI (non maskable interrupt) occurs when an input signal at the  $\overline{\text{NMI}}$  pin switches from high to low. This non maskable interrupt is disabled after a reset. To enable this interrupt, set the PM24 bit in the PM2 register to 1 after setting the interrupt stack pointer (ISP) at the beginning of the program. The  $\overline{\text{NMI}}$  pin shares a pin with port P8\_5, which enables the P8\_5 bit in the P8 register to indicate the input level at the  $\overline{\text{NMI}}$  pin.

Note:

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

## 11.12 Key Input Interrupt

The key input interrupt is enabled by setting ports P10\_4 to P10\_7 as input ports.

The interrupt request is generated if any of the signals applied to ports P10\_4 to P10\_7 switch from high to low. This interrupt also functions as key wake-up to exit wait or stop mode. Figure 11.11 shows a block diagram of the key input interrupt. If any of the ports are held low, signals applied to other ports are not detected as interrupt request signals.

To use the key input interrupt, every register from P10\_4S to P10\_7S should be set to 00h (I/O port) and bits PD10\_4 to PD10\_7 should be set to 0 (input). This is the only setting available for the key input interrupt.

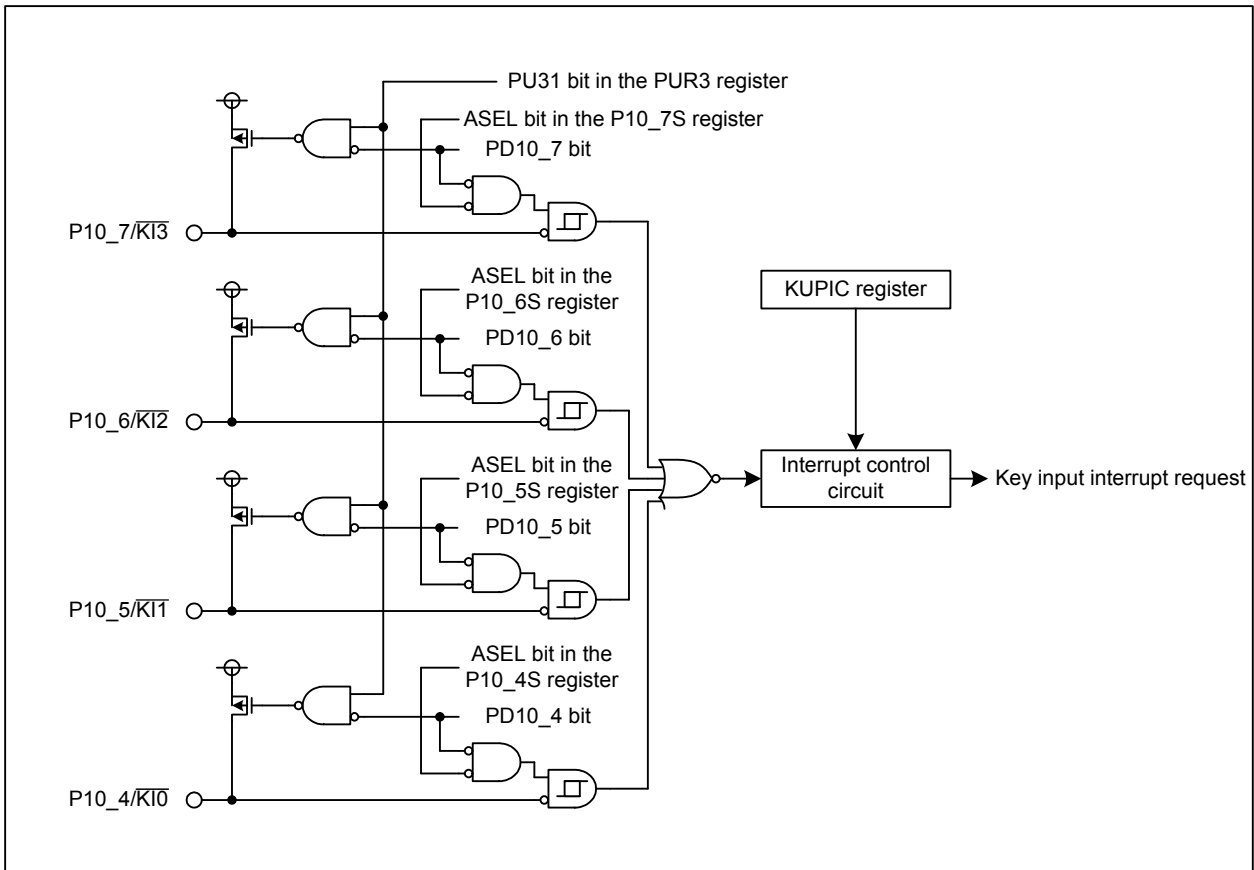


Figure 11.11 Key Input Interrupt Block Diagram

### 11.13 Intelligent I/O Interrupt

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

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

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

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

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

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

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

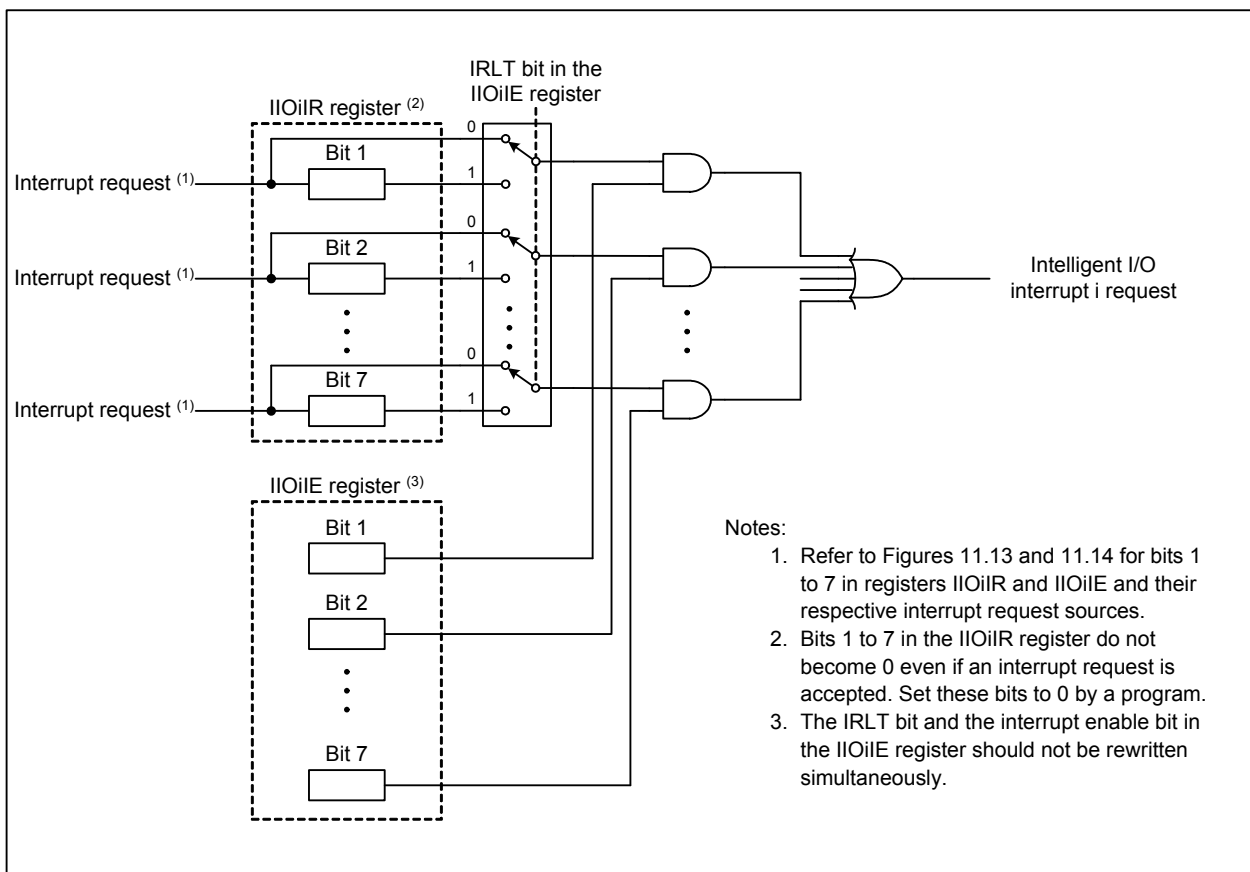
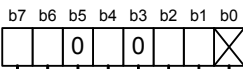


Figure 11.12 Intelligent I/O Interrupt Block Diagram ( $i = 0$  to 11)

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



Symbol IIO0IR to IIO11IR  
Address Refer to the table below

Reset Value ??? 0??1b<sup>(1)</sup>

Bit Symbol	Bit Name	Function	RW
— (b0)	No register bit; this bit is read as 1		—
(Note 2)	0: No interrupt requested 1: Interrupt requested <sup>(3)</sup>		RW
(Note 2)	0: No interrupt requested 1: Interrupt requested <sup>(3)</sup>		RW
— (b3)	Reserved	Should be written with 0	RW
(Note 2)	0: No interrupt requested 1: Interrupt requested <sup>(3)</sup>		RW
— (b5)	Reserved	Should be written with 0	RW
(Note 2)	0: No interrupt requested 1: Interrupt requested <sup>(3)</sup>		RW
(Note 2)	0: No interrupt requested 1: Interrupt requested <sup>(3)</sup>		RW

## Notes:

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

## Bit Symbols for the Intelligent I/O Interrupt Request Register

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

BTxR: Intelligent I/O group x base timer interrupt request (x = 0 to 2)

TMxyR: Intelligent I/O group x time measurement channel y interrupt request (x = 0, 1; y = 0 to 7)

POxyR: Intelligent I/O group x waveform generation channel y interrupt request (x = 0 to 2; y = 0 to 7)

IEzR: Intelligent I/O group 2 IEBus interrupt request (z = 0 to 2)

SIO2RR: Intelligent I/O group 2 receive interrupt request

SIO2TR: Intelligent I/O group 2 transmit interrupt request

Figure 11.13 Registers IIO0IR to IIO11IR

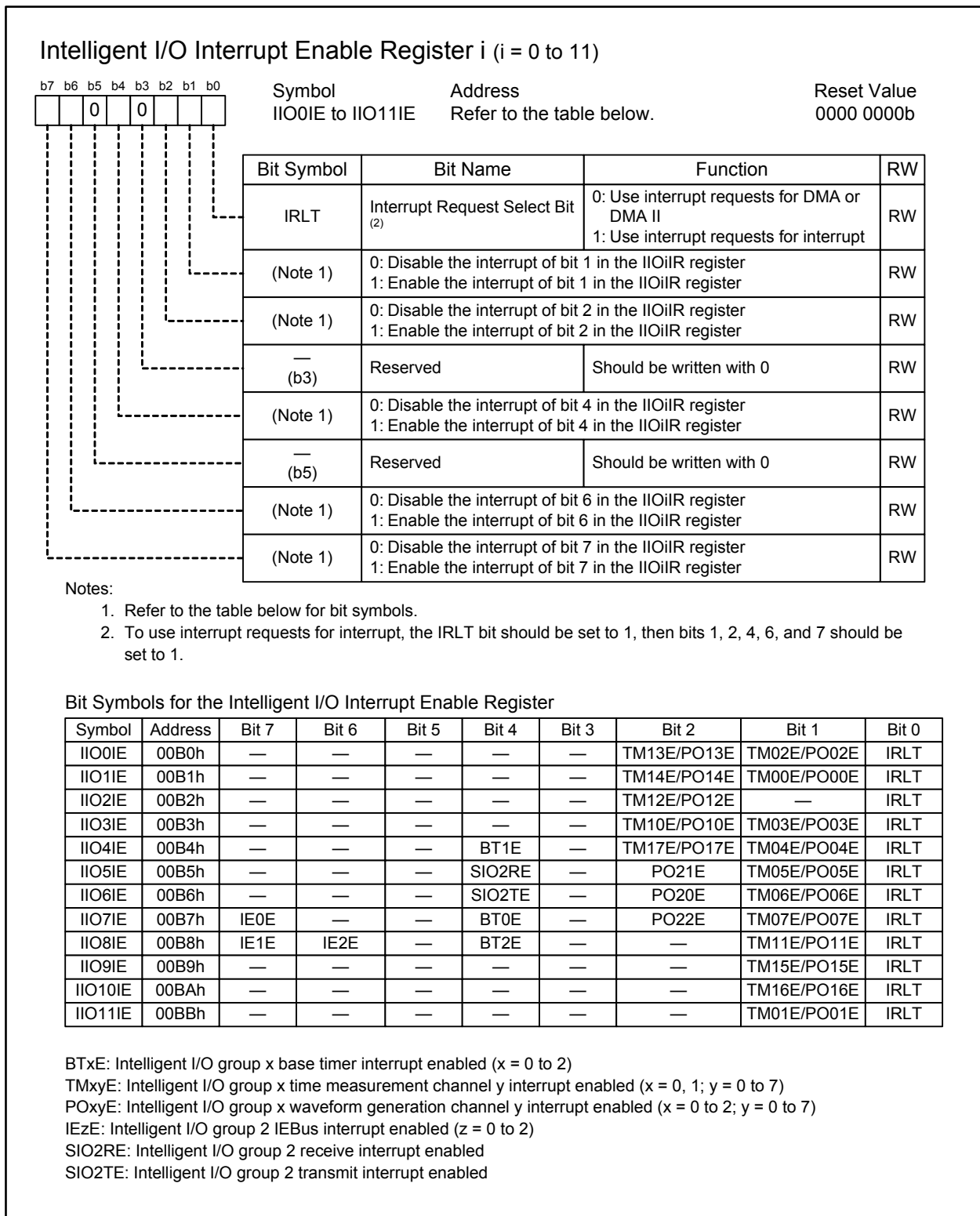


Figure 11.14 Registers IIO0IE to IIO11IE

## 11.14 Notes on Interrupts

### 11.14.1 ISP Setting

The interrupt stack pointer (ISP) is initialized to 00000000h after a reset. Set a value to the ISP before an interrupt is accepted, otherwise the program may go out of control. A multiple of 4 should be set to the ISP, which enables faster interrupt sequence due to less memory access.

When using NMI, in particular, since this interrupt cannot be disabled, set the PM24 bit in the PM2 register to 1 (NMI enabled) after setting the ISP at the beginning of the program.

### 11.14.2 NMI

- NMI cannot be disabled once the PM24 bit in the PM2 register is set to 1 (NMI enabled). This bit setting should be done only when using NMI.
- When the PM24 bit in the PM2 register is 1 (NMI enabled), the P8\_5 bit in the P8 register is enabled just for monitoring the  $\overline{\text{NMI}}$  pin state. It is not enabled as a general port.

### 11.14.3 External Interrupts

- The input signal to the  $\overline{\text{INTi}}$  pin requires the pulse width specified in the electrical characteristics ( $i = 0$  to  $5$ ). If the pulse width is narrower than the specification, an external interrupt may not be accepted.
- When the effective level or edge of the  $\overline{\text{INTi}}$  pin ( $i = 0$  to  $5$ ) is changed by the following bits: bits POL, LVS in the INTiIC register, the IFSR0i bit ( $i = 0$  to  $5$ ) in the IFSR0 register, the corresponding IR bit may become 1 (interrupt requested). When setting the above mentioned bits, preset bits ILVL2 to ILVL0 in the INTiIC register to 000b (interrupt disabled). After setting the above mentioned bits, set the corresponding IR bit to 0 (no interrupt requested), then rewrite bits ILVL2 to ILVL0.



## 12. Watchdog Timer

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

Select either an interrupt request or a reset with the CM06 bit in the CM0 register for when the watchdog timer underflows. Once the CM06 bit is set to 1 (reset), it cannot be changed to 0 (watchdog timer interrupt) by a program. It can be set to 0 only by a reset.

The watchdog timer has a prescaler which divides the peripheral bus clock by 16 or 128. To select the divide ratio, set the WDC7 bit in the WDC register.

The watchdog timer is stopped in wait mode, stop mode, or when the  $\overline{\text{HOLD}}$  signal is driven low. It resumes counting from the value held when exiting the mode or state.

The general formula to calculate a watchdog timer period is:

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

For example, when the peripheral bus clock is 1/2 of 50 MHz CPU clock and the prescaler has a divide-by-16 operation, the watchdog timer period is approximately 21 ms. Depending on the timing of when a value is written to the WDTS register, a marginal error of one prescaler output cycle (maximum) may occur in the watchdog timer period.

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

After a reset, both the watchdog timer and the prescaler are stopped. They start counting when a write operation to the WDTS register is performed.

Figure 12.1 shows a block diagram of the watchdog timer. Figures 12.2 and 12.3 show registers associated with the watchdog timer.

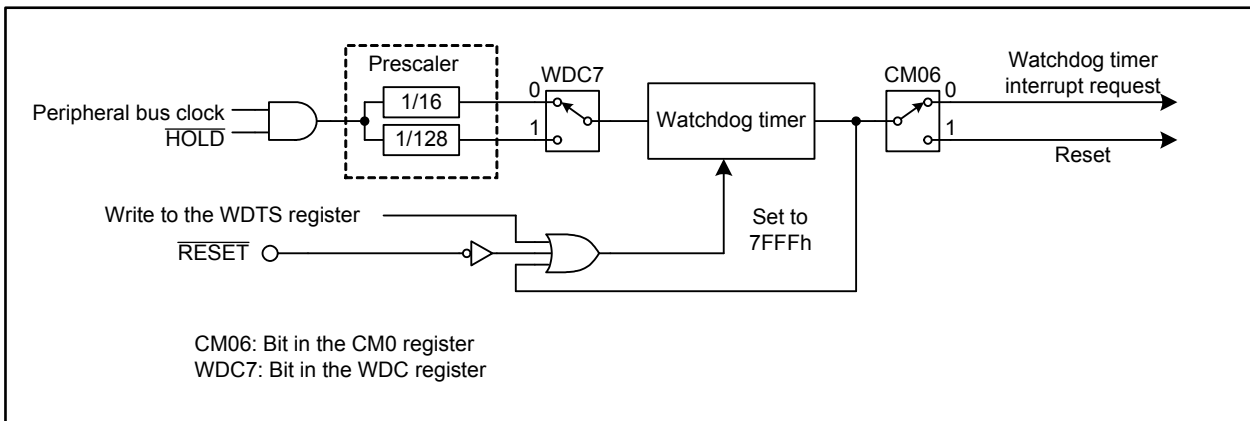


Figure 12.1 Watchdog Timer Block Diagram

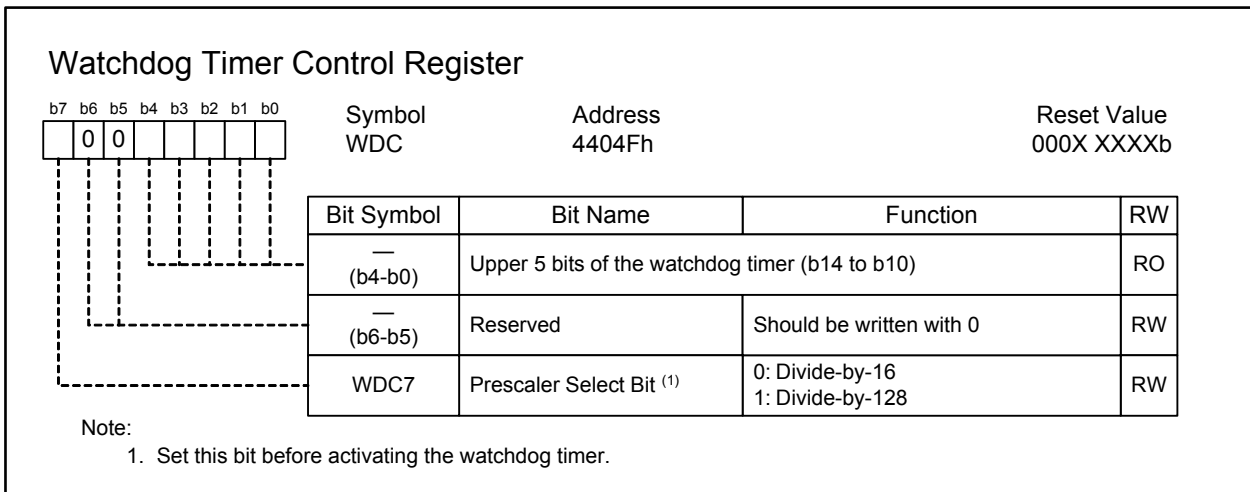


Figure 12.2 WDC Register

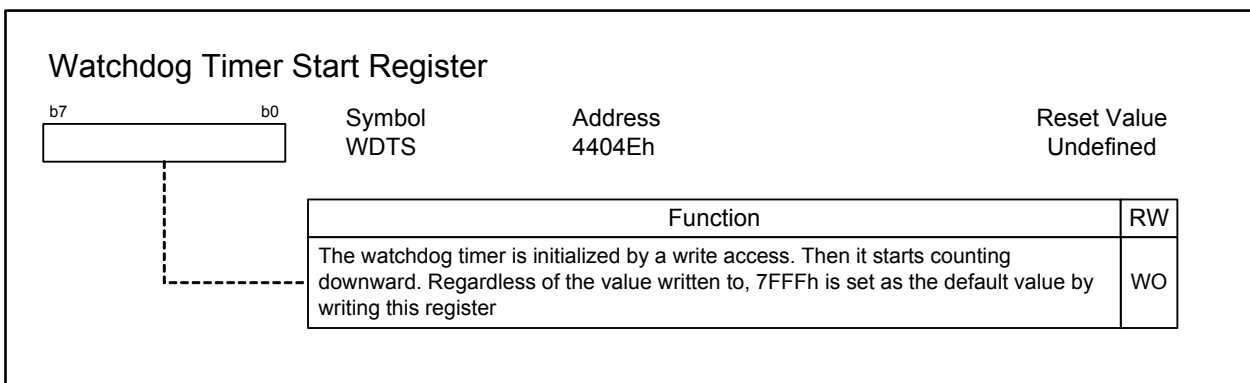


Figure 12.3 WDTS Register

## 13. DMAC

Direct memory access (DMA) is a system that can control data transfer without using a CPU instruction. The R32C/100 Series' four channel DMA controller (DMAC) transmits 8-bit (byte), 16-bit (word), or 32-bit (long word) data in cycle-steal mode from a source address to a destination address each time a transfer request is generated.

The DMAC, which shares a data bus with the CPU, has a higher bus access priority than the CPU. This allows the DMAC to perform fast data transfer when a transfer request is generated.

Figure 13.1 shows a map of the CPU-internal registers associated with DMAC. Table 13.1 lists DMAC specifications. Figures 13.2 to 13.10 show registers associated with DMAC. Since the registers shown in Figure 13.1 are allocated in the CPU, the LDC or STC instruction should be used to write to the registers.

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

Note:

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

**Figure 13.1 CPU-internal Registers for DMAC**

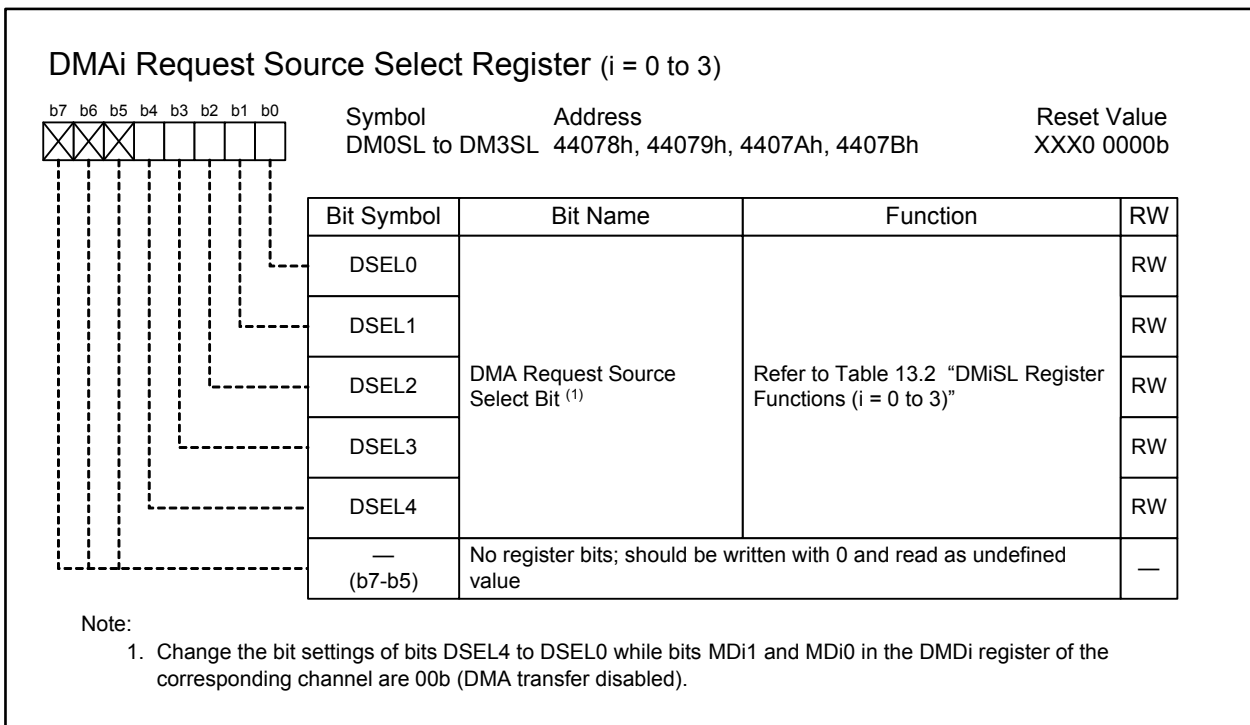
**Table 13.1 DMAC Specifications (i = 0 to 3)**

Item		Specification
Channels		4
Bus request mode		Cycle-steal mode
Transfer memory spaces		From a given address in a 64-Mbyte space (00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh) to another given address in the same space
Maximum transfer bytes		64-Mbytes (when 32-bit data is transferred), 32-Mbytes (when 16-bit data is transferred), 16-Mbytes (when 8-bit data is transferred)
DMA request sources <sup>(1)</sup>		Falling edge or both edges of signals applied to pins INT0 to INT3 Interrupt requests from timers A0 to A4 Interrupt requests from timers B0 to B5 Transmit/receive interrupt requests from UART0 to UART8 A/D conversion interrupt requests Intelligent I/O interrupt requests Software trigger
Channel priority		DMA0 > DMA1 > DMA2 > DMA3 (DMA0 has the highest priority)
Transfer sizes		8 bits, 16 bits, or 32 bits
Addressing modes		Incrementing addressing or non-incrementing addressing
Transfer modes	Single transfer	Transfer is completed when the DCTi register becomes 00000000h
	Repeat transfer	When the DCTi register becomes 00000000h, the value of the DCRi register is reloaded into the DCTi register to continue the DMA transfer
DMA transfer complete interrupt request generation timing		When the DCTi register changes from 00000001h to 00000000h
DMA transfer start-up	Single transfer	When a DMA transfer request is generated after the DCTi register is set to a value other than 00000000h and bits MDi1 and MDi0 in the DMDi register are set to 01b (single transfer)
	Repeat transfer	When a DMA transfer request is generated after the DCTi register is set to a value other than 00000000h and bits MDi1 and MDi0 are set to 11b (repeat transfer)
DMA transfer stop	Single transfer	When bits MDi1 and MDi0 are set to 00b (DMA transfer disabled)
	Repeat transfer	When bits MDi1 and MDi0 are set to 00b (DMA transfer disabled)
Reload timing to DCTi, DSAi, or DDAi register		When the DCTi register changes from 00000001h to 00000000h in repeat transfer mode
Minimum DMA transfer cycles		3

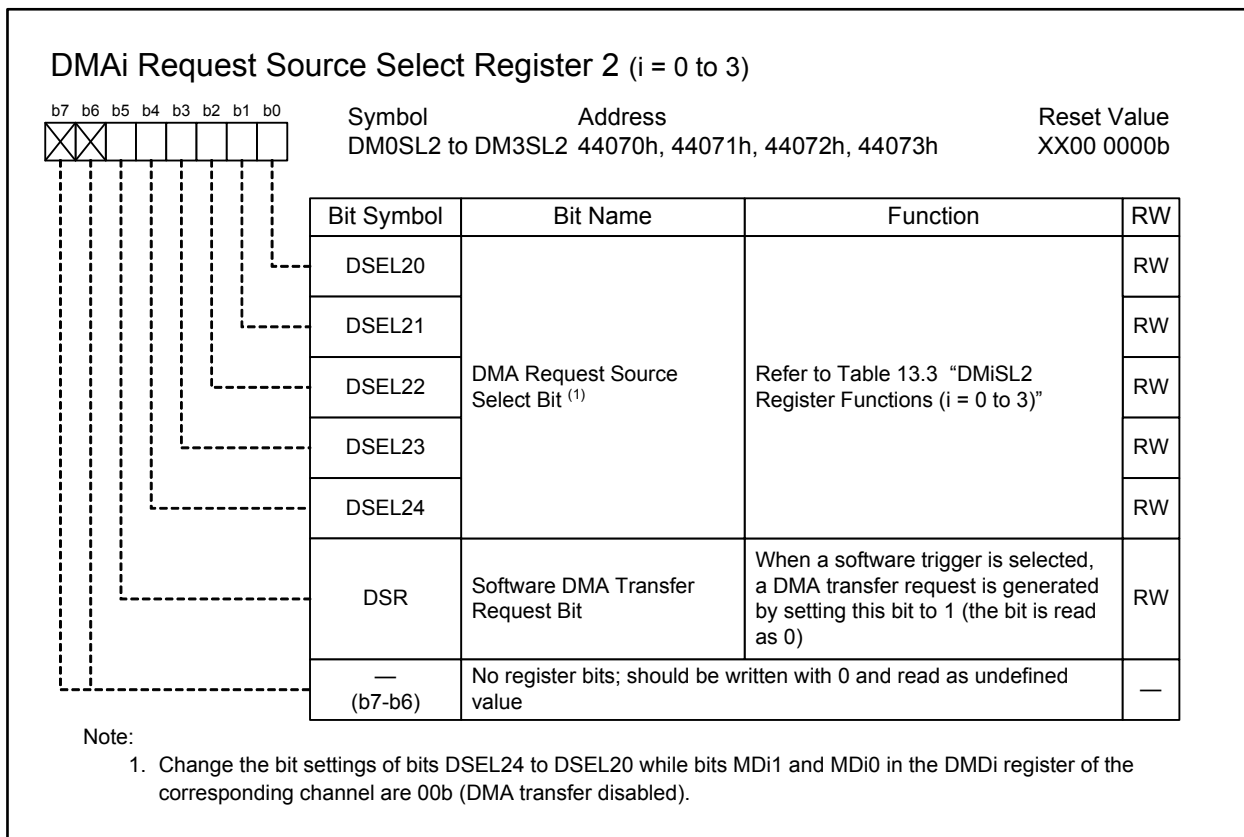
Note:

1. DMA transfer does not affect any interrupts.

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



**Figure 13.2 Registers DM0SL to DM3SL**



**Figure 13.3 Registers DM0SL2 to DM3SL2**

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

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

**Notes:**

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

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

Setting Value	DMA Request Source			
b4 b3 b2 b1 b0	DMA0	DMA1	DMA2	DMA3
0 0 0 0 0	Software trigger			
0 0 0 0 1	Reserved			
0 0 0 1 0	Reserved			
0 0 0 1 1	Reserved			
0 0 1 0 0	Reserved			
0 0 1 0 1	Reserved			
0 0 1 1 0	Reserved			
0 0 1 1 1	Reserved			
0 1 0 0 0	Reserved			
0 1 0 0 1	Reserved			
0 1 0 1 0	Reserved			
0 1 0 1 1	Reserved			
0 1 1 0 0	Reserved			
0 1 1 0 1	Reserved			
0 1 1 1 0	Reserved			
0 1 1 1 1	Reserved			
1 0 0 0 0	Reserved			
1 0 0 0 1	Reserved			
1 0 0 1 0	Reserved			
1 0 0 1 1	Reserved			
1 0 1 0 0	Reserved			
1 0 1 0 1	Reserved			
1 0 1 1 0	Reserved			
1 0 1 1 1	Reserved			
1 1 0 0 0	UART7 transmit interrupt request			
1 1 0 0 1	UART7 receive interrupt request			
1 1 0 1 0	UART8 transmit interrupt request			
1 1 0 1 1	UART8 receive interrupt request			
1 1 1 0 0	Reserved			
1 1 1 0 1	Reserved			
1 1 1 1 0	Reserved			
1 1 1 1 1	Reserved			



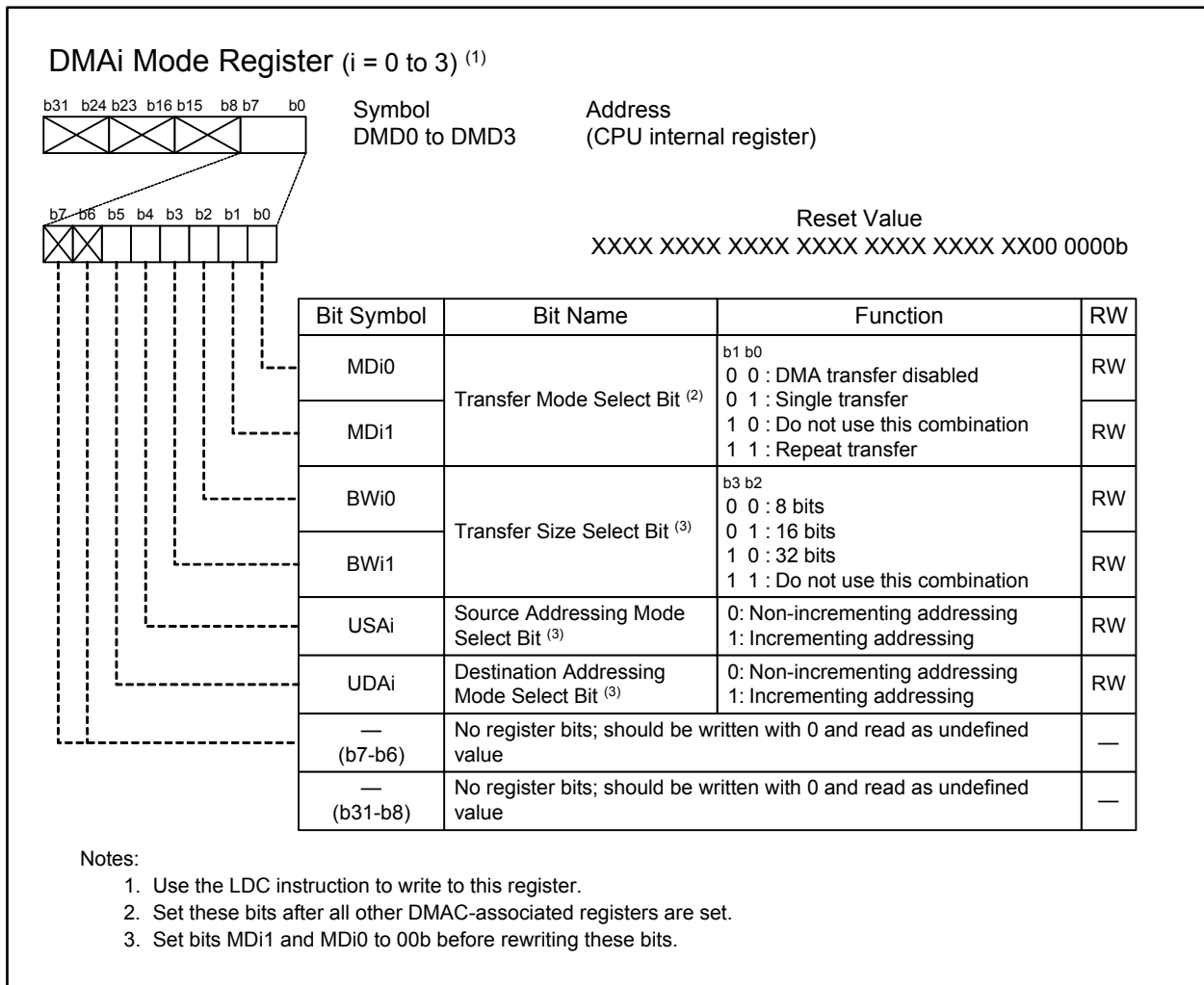


Figure 13.4 Registers DMD0 to DMD3

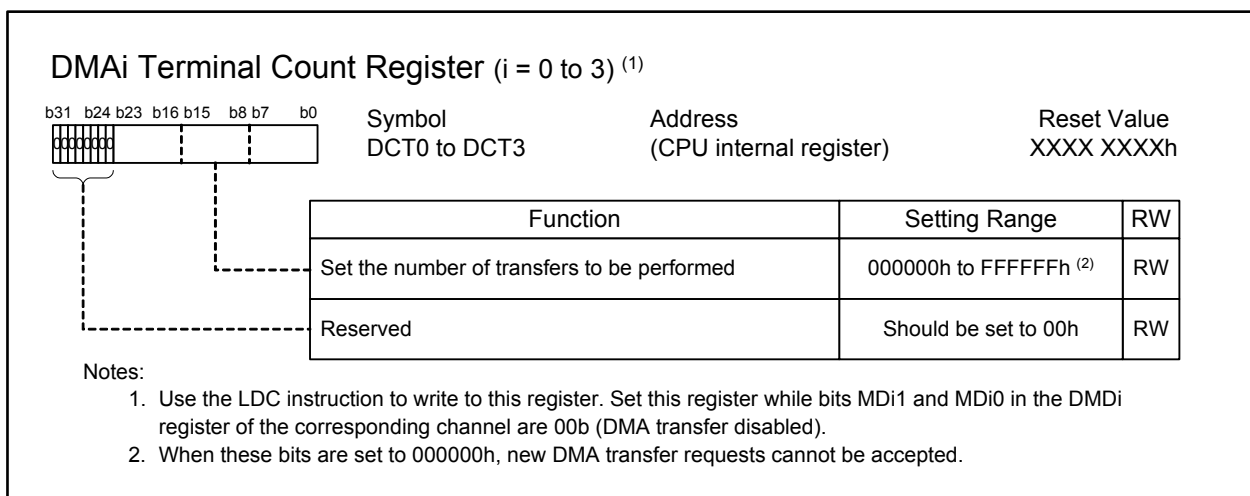


Figure 13.5 Registers DCT0 to DCT3

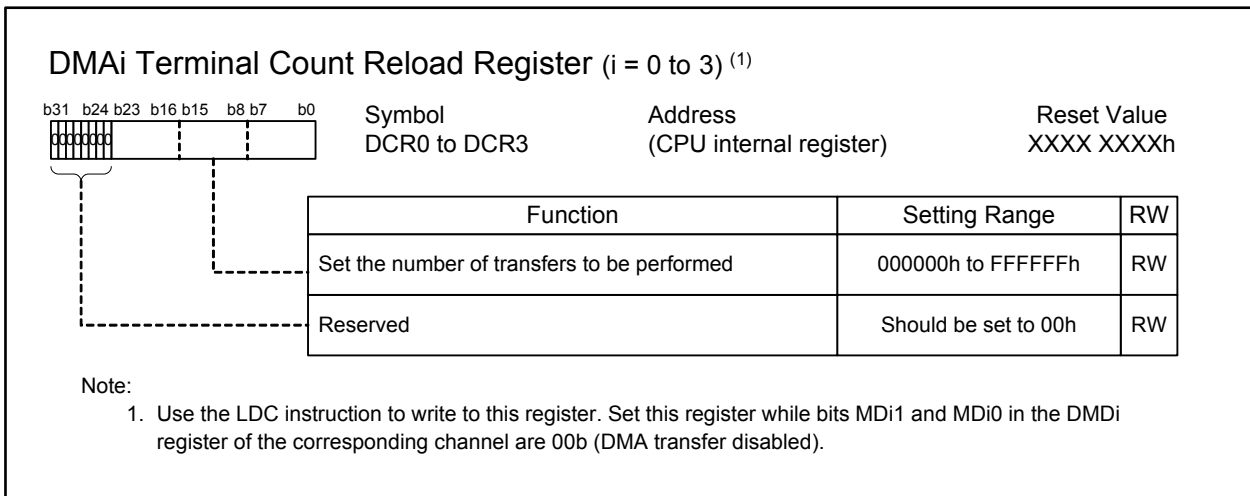


Figure 13.6 Registers DCR0 to DCR3

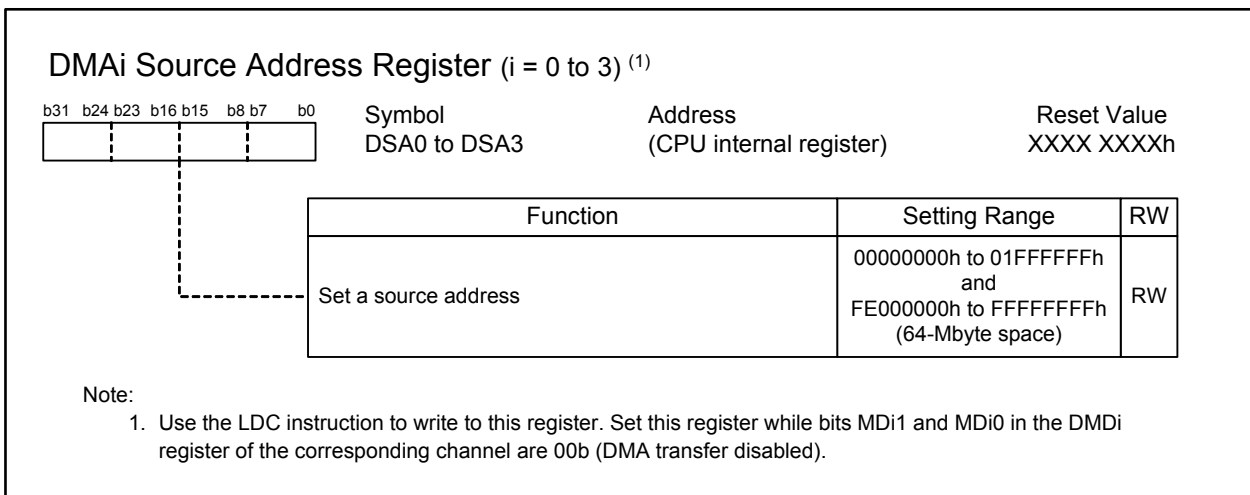
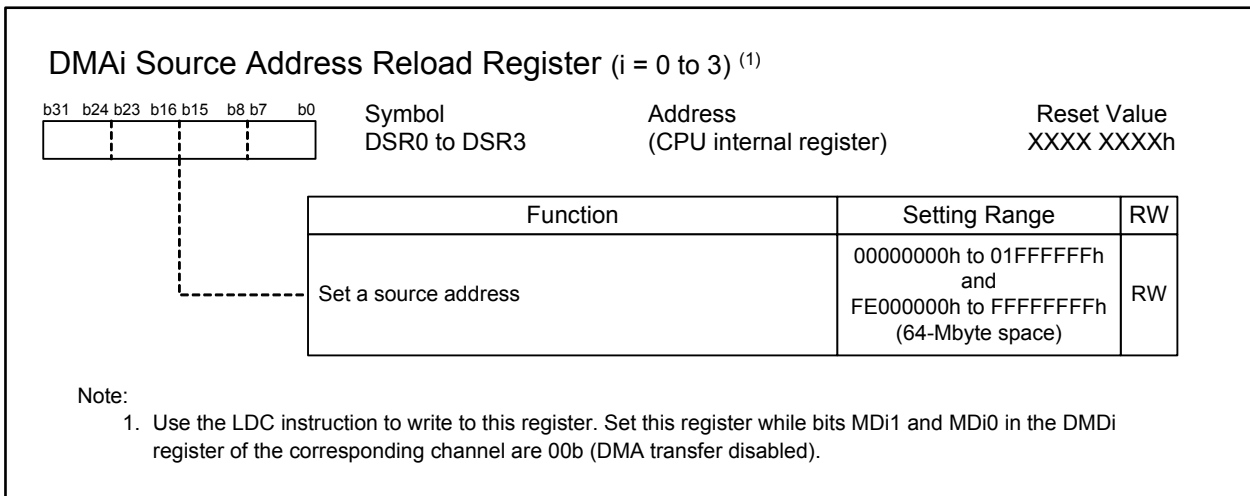
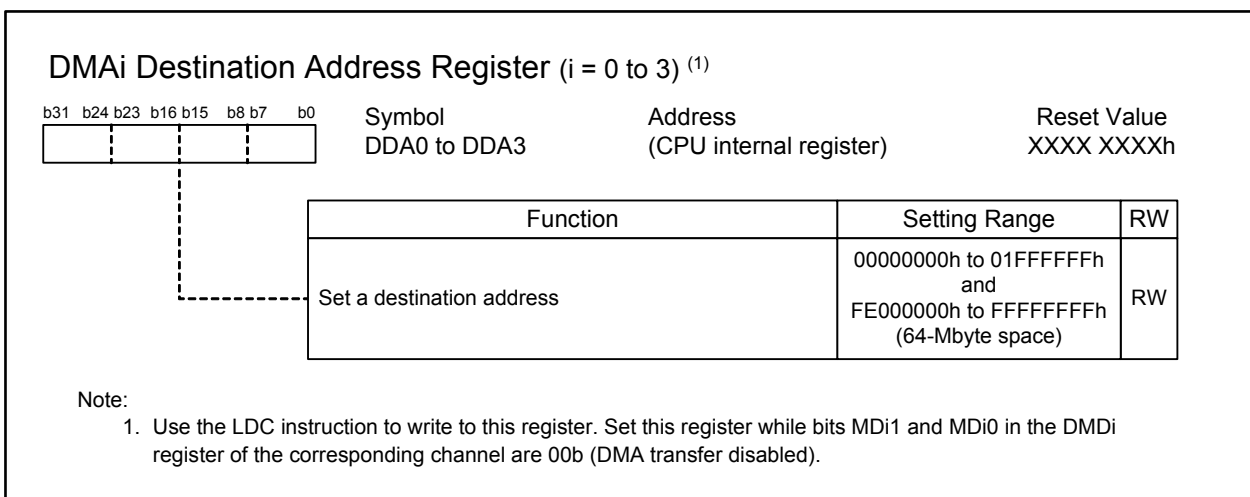
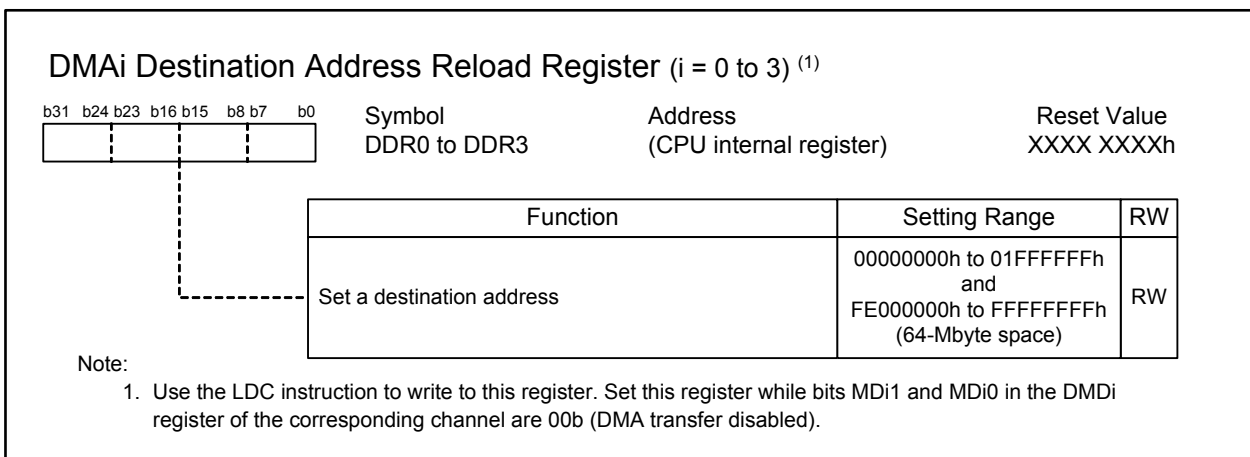


Figure 13.7 Registers DSA0 to DSA3

**Figure 13.8 Registers DSR0 to DSR3****Figure 13.9 Registers DDA0 to DDA3****Figure 13.10 Registers DDR0 to DDR3**

### 13.1 Transfer Cycle

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

The read and write bus cycles vary with the setting of registers DSA<sub>i</sub> and DDA<sub>i</sub>, the width and timing of the data bus connected to the relevant device ( $i = 0$  to 3).

#### 13.1.1 Effect of Transfer Address and Data Bus Width

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

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

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

### 13.1.2 Effect of Bus Timing

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

**Table 13.5 Bus Width and Bus Cycles**

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

Notes:

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

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

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

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

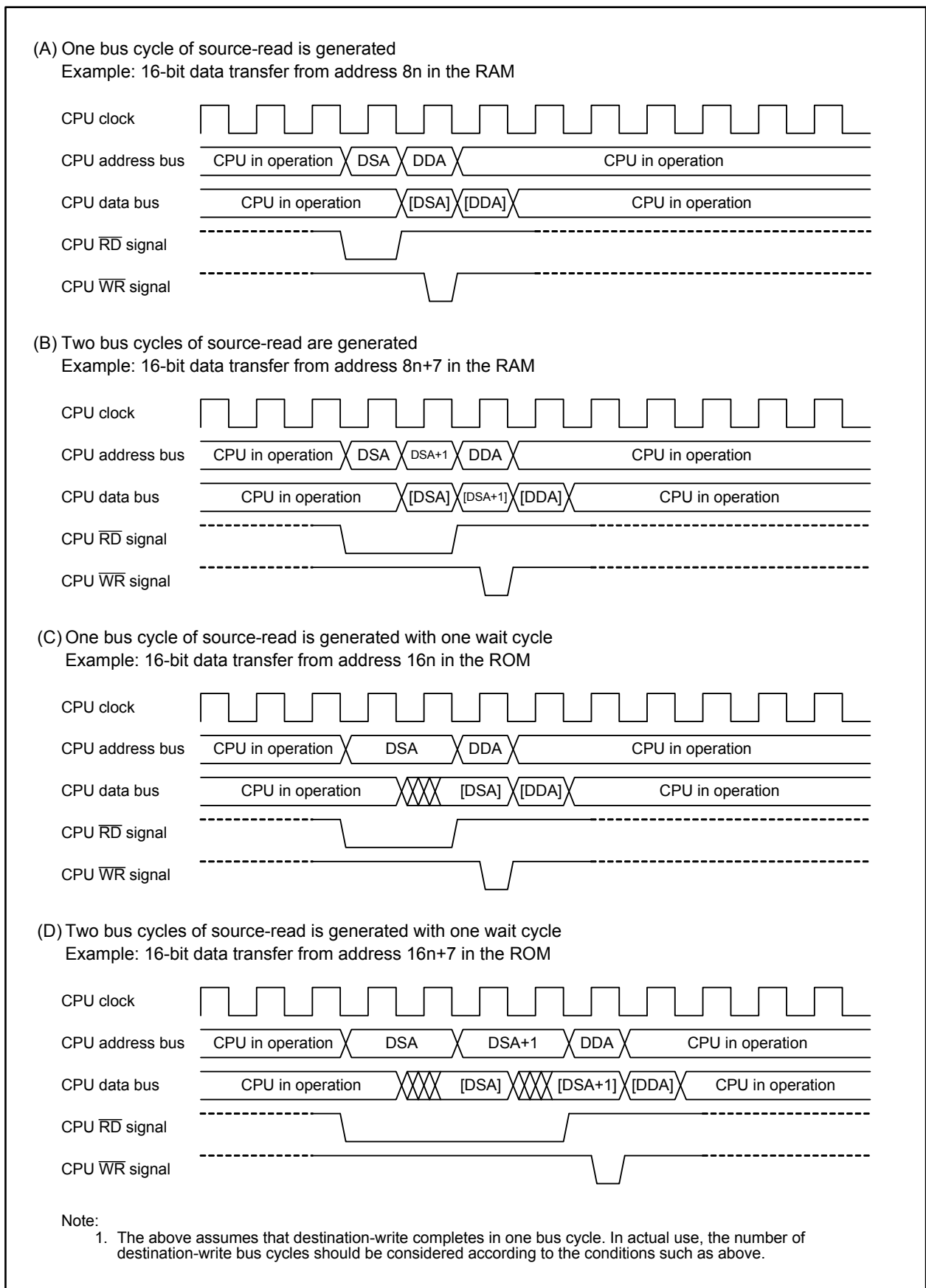


Figure 13.11 Source-read Bus Cycles in a Transfer Cycle

## 13.2 DMA Transfer Cycle

The DMA transfer cycles are calculated as follows:

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

where:

$j$  = access cycles for read

$k$  = access cycles for write (refer to Table 13.5)

Each bus cycle, source-read and destination-write, requires at least one cycle. In addition, more cycles may be required depending on the transfer address. Refer to Table 13.4 for details on the required bus cycles.

“+1” in the formula above means a cycle required to decrement the value of DCT<sub>i</sub> register (i = 0 to 3).

The following are calculation examples:

To transfer 32-bit data from address 400h in the RAM to address 800h in the RAM,

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

Thus, there are three cycles.

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

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

Thus, there are nine cycles.

### 13.3 Channel Priority and DMA Transfer Timing

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

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

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

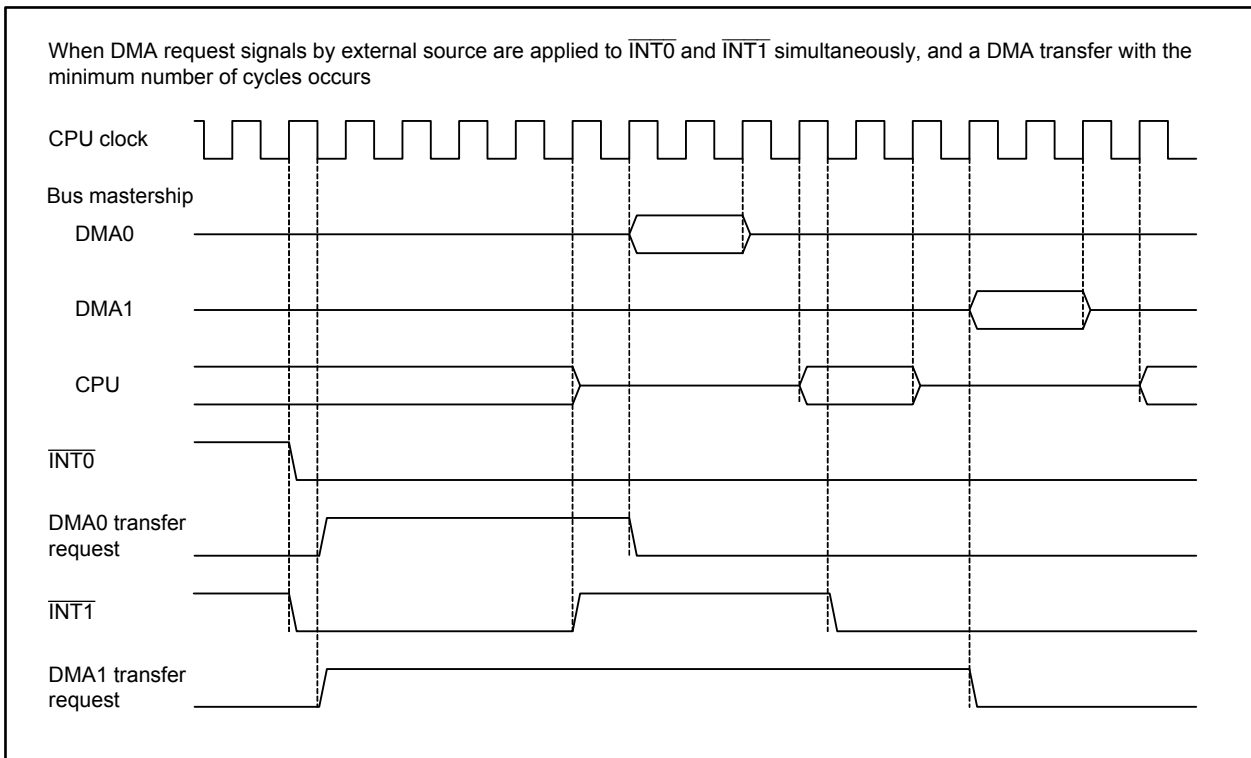


Figure 13.12 DMA Transfer by External Source



## 13.4 Notes on DMAC

### 13.4.1 DMAC-associated Register Settings

- Set DMAC-associated registers while bits MDi1 and MDi0 in the DMDi register are 00b (DMA transfer disabled) (i = 0 to 3). Then, set bits MDi1 and MDi0 to 01b (single transfer) or 11b (repeat transfer) at the end of the setup procedure. This procedure also applies when rewriting bits UDAi, USAi, and BWi1 and BWi0 in the DMDi register.
- When rewriting the DMAC-associated registers while DMA transfer is enabled, stop the peripherals that can be DMA triggers so that no DMA transfer request is generated, then set bits MDi1 and MDi0 in the DMDi register of the corresponding channel to 00b (DMA transfer disabled).
- Once a DMA transfer request is accepted, DMA transfer cannot be disabled even if setting bits MDi1 and MDi0 in the DMDi register to 00b (DMA transfer disabled). Do not change the settings of any DMAC-associated registers other than bits MDi1 and MDi0 until the DMA transfer is completed.
- After setting registers DMiSL and DMiSL2, wait at least six peripheral bus clocks to set bits MDi1 and MDi0 in the DMDi register to 01b (single transfer) or 11b (repeat transfer).

### 13.4.2 Reading DMAC-associated Registers

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

## 14. DMAC II

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

Table 14.1 lists specifications of DMAC II.

**Table 14.1 DMAC II Specifications**

Item	Specification
DMAC II request sources	Interrupt requests from the peripherals of which bits ILVL2 to ILVL0 in the corresponding interrupt control register are set to 111b (level 7)
Transfer types	<ul style="list-style-type: none"> <li>• Data in memory is transferred to memory (memory-to-memory transfer)</li> <li>• Immediate data is transferred to memory (immediate data transfer)</li> <li>• Data in memory + data in memory are transferred to memory (calculation result transfer)</li> <li>• Immediate data + data in memory are transferred to memory (calculation result transfer)</li> </ul>
Transfer sizes	8 bits or 16 bits
Transfer memory spaces	From a given address in a 64-Mbyte space (00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh) to another given address in the same space <sup>(1)</sup>
Addressing modes	Individually selectable for each source address and destination address from the following two modes: <ul style="list-style-type: none"> <li>• Non-incrementing addressing: Address is held constant throughout a data transfer/DMAC II transaction</li> <li>• Incrementing addressing: Address increments by 1 (when 8-bit data is transferred) or 2 (when 16-bit data is transferred) after each data transfer</li> </ul>
Transfer modes	<ul style="list-style-type: none"> <li>• Single transfer: Only one data transfer is performed by one transfer request</li> <li>• Burst transfer: Data transfers are continuously performed for the number of times set in the transfer counter by one transfer request</li> <li>• Multiple transfer: Multiple memory-to-memory transfers are performed from different source addresses to different destination addresses by one transfer request</li> </ul>
Chain transfer	Data transfer is sequentially performed by switching among multiple DMAC II indexes (transfer information)
DMA II transfer complete interrupt request	An interrupt request is generated when the transfer counter reaches 0000h

Note:

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

### 14.1 DMAC II Settings

To use DMAC II, set the following:

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

### 14.1.1 Registers RIPL1 and RIPL2

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

Figure 14.1 shows registers RIPL1 and RIPL2.

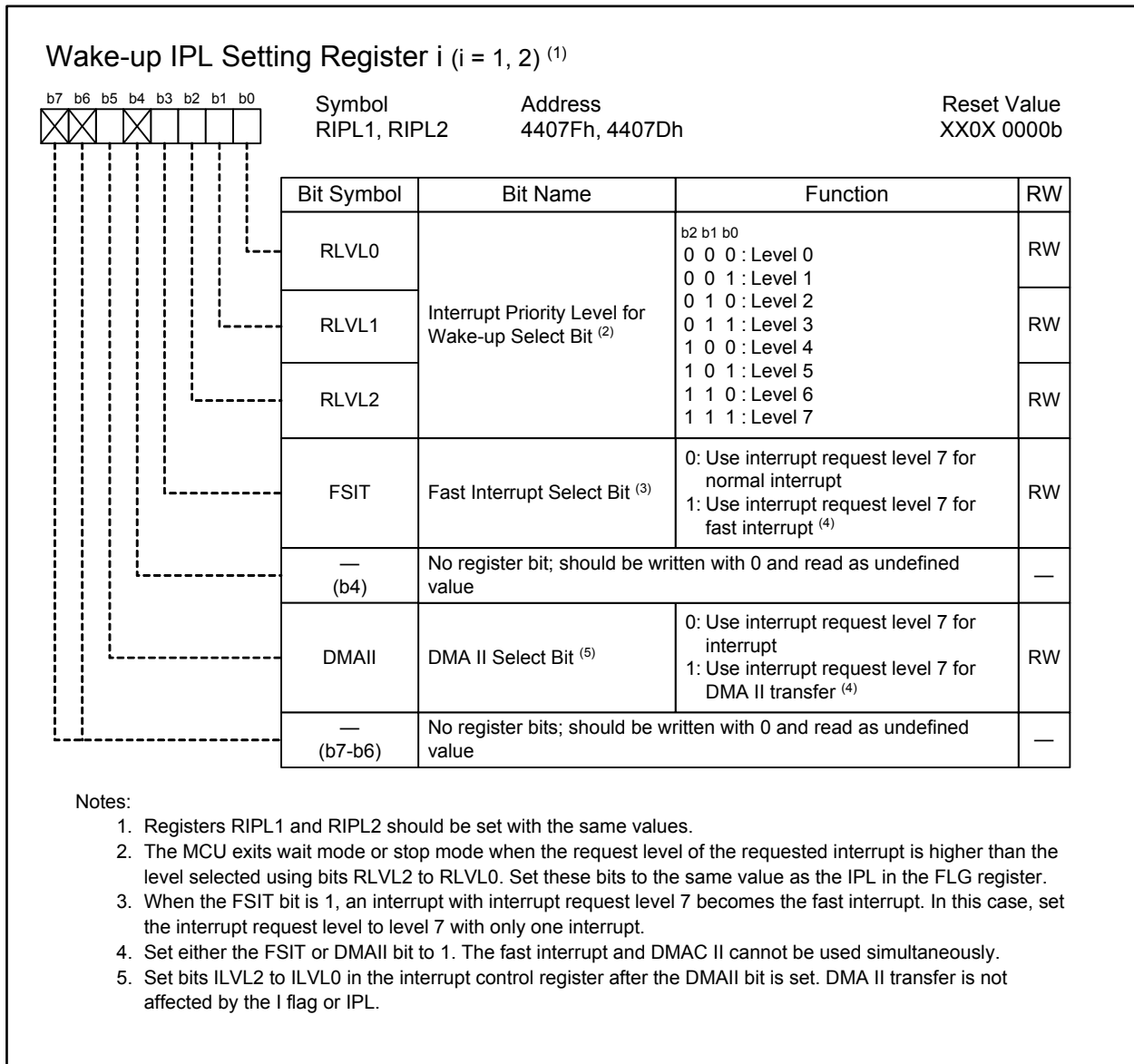


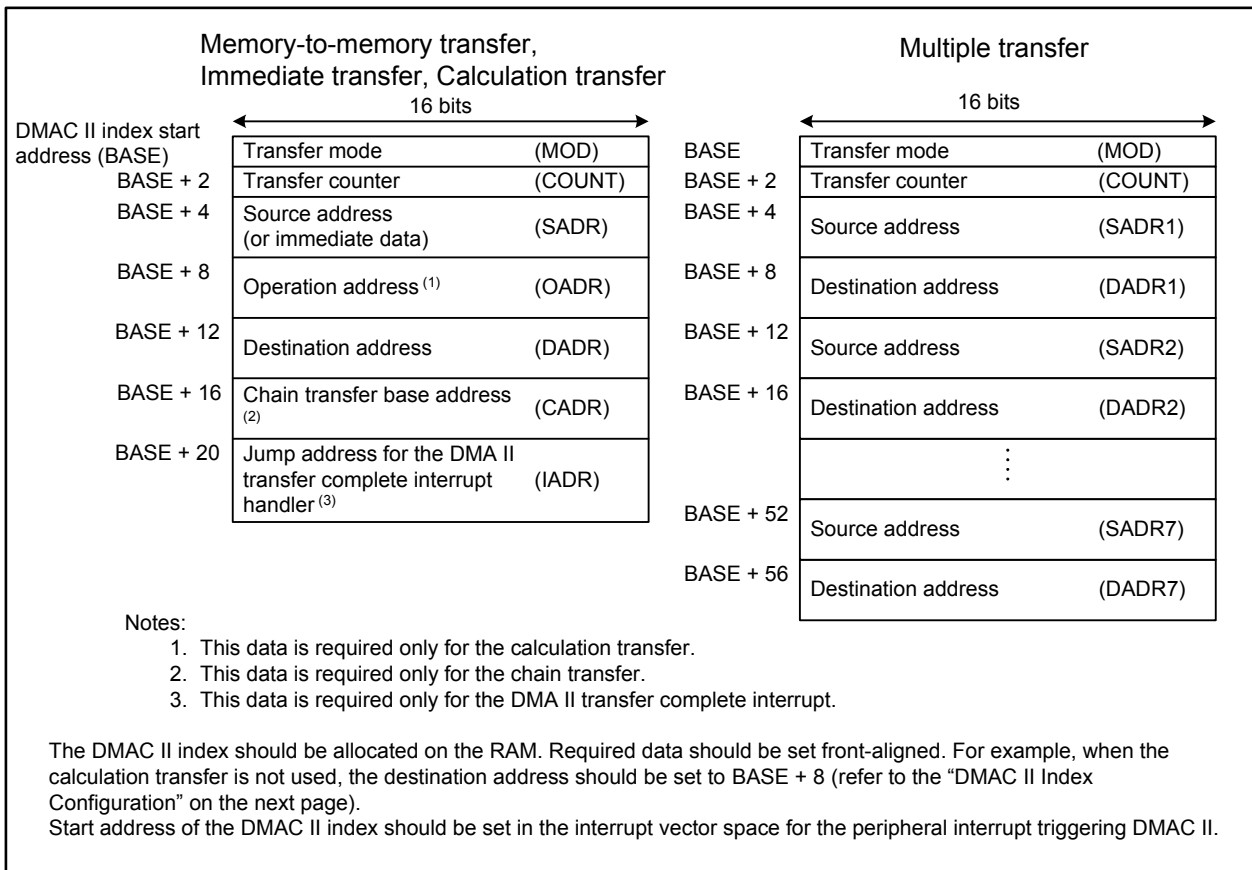
Figure 14.1 Registers RIPL1 and RIPL2

### 14.1.2 DMAC II Index

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

This DMAC II index should be allocated on the RAM.

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



**Figure 14.2 DMAC II Index**

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

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

The symbols above are hereinafter used in place of their respective parameters.

**Table 14.2 DMAC II Index Configuration**

Transfer Data	Memory-to-memory Transfer/ Immediate Data Transfer				Calculation Transfer				Multiple Transfer	
	Not used	Used	Not used	Used	Not used	Used	Not used	Used		
Chain transfer	Not used	Used	Not used	Used	Not used	Used	Not used	Used	Not available	
DMA II transfer complete interrupt	Not used	Not used	Used	Used	Not used	Not used	Used	Used	Not available	
DMAC II index	MOD	MOD	MOD	MOD	MOD	MOD	MOD	MOD	MOD	
	COUNT	COUNT	COUNT	COUNT	COUNT	COUNT	COUNT	COUNT	COUNT	
	SADR	SADR	SADR	SADR	SADR	SADR	SADR	SADR	SADR1	
	DADR	DADR	DADR	DADR	OADR	OADR	OADR	OADR	DADR1	
	12 bytes	CADR	IADR	CADR	IADR	DADR	DADR	DADR	DADR	SADRi

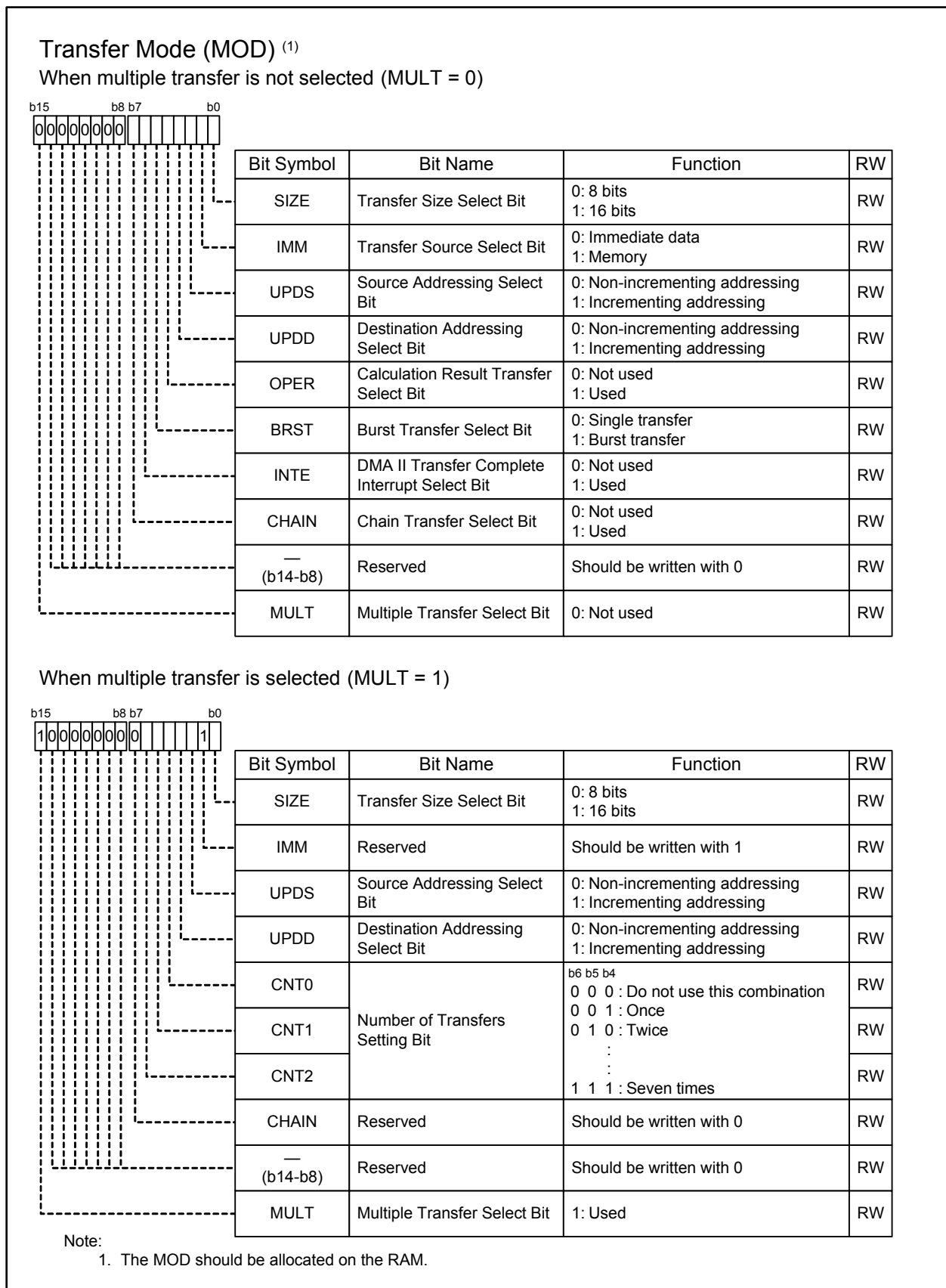


Figure 14.3 MOD

### 14.1.3 Interrupt Control Register of the Peripherals

Set bits ILVL2 to ILVL0 in the interrupt control register for the peripheral interrupt triggering DMAC II to 111b (level 7).

### 14.1.4 Relocatable Vector Table of the Peripherals

Set the start address of the DMAC II index in the interrupt vector space for the peripheral interrupt triggering DMAC II.

To use the chain transfer, allocate the relocatable vector table on the RAM.

### 14.1.5 IRLT Bit in the IIOiE Register (i = 0 to 11)

To use the intelligent I/O interrupt as a trigger for DMAC II, set the IRLT bit in the corresponding IIOiE register to 0 (interrupt request for DMA or DMA II used).

## 14.2 DMAC II Operation

Set the DMAII bit in registers RIPL1 and RIPL2 to 1 (interrupt request level 7 used for DMA II transfer) to perform a DMA II transfer. DMAC II starts by an interrupt request from any peripheral whose bits ILVL2 to ILVL0 in the corresponding interrupt control register are set to 111b (level 7). These peripheral interrupt requests are available only for DMA II transfer and cannot be used for the CPU.

When an interrupt request is generated with interrupt request level 7, DMAC II starts irrespective of the state of the I flag or IPL.

When a peripheral interrupt request triggering DMAC II and a higher-priority request such as the watchdog timer interrupt, low voltage detection interrupt, oscillator stop detection interrupt, or NMI are simultaneously generated, the higher-priority interrupt is accepted prior to the DMA II transfer, and the DMA II transfer starts after the higher-priority interrupt sequence.

## 14.3 Transfer Types

DMAC II transfers three types of 8-bit or 16-bit data as follows:

- Memory-to-memory transfer: Data is transferred from a given memory location in a 64-Mbyte space (addresses 00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh) to another given memory location in the same space.
- Immediate data transfer: Immediate data is transferred to a given memory location in a 64-Mbyte space.
- Calculation transfer: Two data are added together and the result is transferred to a given memory location in a 64-Kbyte space.

When 16-bit data is transferred to DADR at FFFFFFFFh, it is transferred to 00000000h as well as FFFFFFFFh. The same transfer is performed when SADR is FFFFFFFFh.

### 14.3.1 Memory-to-memory Transfer

Data transfer between any two memory locations can be:

- A transfer from a fixed address to another fixed address
- A transfer from a fixed address to an address range in memory
- A transfer from an address range in memory to a fixed address
- A transfer from an address range in memory to another address range in memory

When increment addressing mode is selected, SADR and DADR increment by 1 in an 8-bit transfer and by 2 in a 16-bit transfer after a data transfer for the next transfer. When SADR or DADR exceeds FFFFFFFFh by the incrementation, it returns to 00000000h. Likewise, when SADR or DADR exceeds 01FFFFFFh, it becomes 02000000h, but an actual transfer is performed for FE000000h.

### 14.3.2 Immediate Data Transfer

DMAC II transfers immediate data to a given memory location. Either incrementing or non-incrementing addressing mode can be selected for the destination address. Store the immediate data to be transferred into SADR. To transfer 8-bit immediate data, set the data to the lower 1 byte of SADR. The upper 3 bytes are ignored. To transfer 16-bit immediate data, set the data to the lower 2 bytes. The upper 2 bytes are ignored.

### 14.3.3 Calculation Result Transfer

After two memory data or immediate data and memory data are added together, DMAC II transfers the calculated result to a given memory location. Set an address to be calculated or immediate data to SADR and set the other address to be calculated to OADR. Either incrementing or non-incrementing addressing mode can be selected for source and destination addresses when performing data in memory + data in memory calculation transfer. If the source addressing is in incrementing mode, the operation addressing is also in incrementing mode. When performing immediate data + data in memory calculation transfer, the addressing mode is selectable only for the destination address.

## 14.4 Transfer Modes

DMAC II provides three types of basic transfer mode: single transfer, burst transfer, and multiple transfer. COUNT determines the number of transfers to be performed. Transfers are not performed when COUNT is 0000h.

### 14.4.1 Single Transfer

Set the BRST bit in the MOD to 0.

A single data transfer is performed by one transfer request.

When incrementing addressing mode is selected for the source and/or destination address, the address or addresses increment after a data transfer for the next transfer.

COUNT is decremented each time a data transfer is performed. When COUNT reaches 0000h, the DMA II transfer complete interrupt request is generated if the INTE bit in the MOD is 1 (DMA II transfer complete interrupt used).

### 14.4.2 Burst Transfer

Set the BRST bit in the MOD to 1.

DMAC II continuously transfers data for the number of times determined by COUNT with one transfer request. COUNT decrements each time a data transfer is performed. When COUNT reaches 0000h, the burst transfer is completed. The DMA II transfer complete interrupt request is generated if the INTE bit is 1 (DMA II transfer complete interrupt used).

No interrupts are accepted during a burst transfer.

### 14.4.3 Multiple Transfer

Set the MULT bit in the MOD to 1.

Multiple memory-to-memory transfers are performed from different source addresses to different destination addresses using one transfer request.

Set bits CNT2 to CNT0 in the MOD to select the number of transfers to be performed from 001b (once) to 111b (seven times). Do not set these bits to 000b.

Allocate the required number of SDARs and DADRs alternately following MOD and COUNT.

When the multiple transfer is selected, the following transfer functions are not available: calculation result transfer, burst transfer, chain transfer, and DMA II transfer complete interrupt.



## 14.5 Chain Transfer

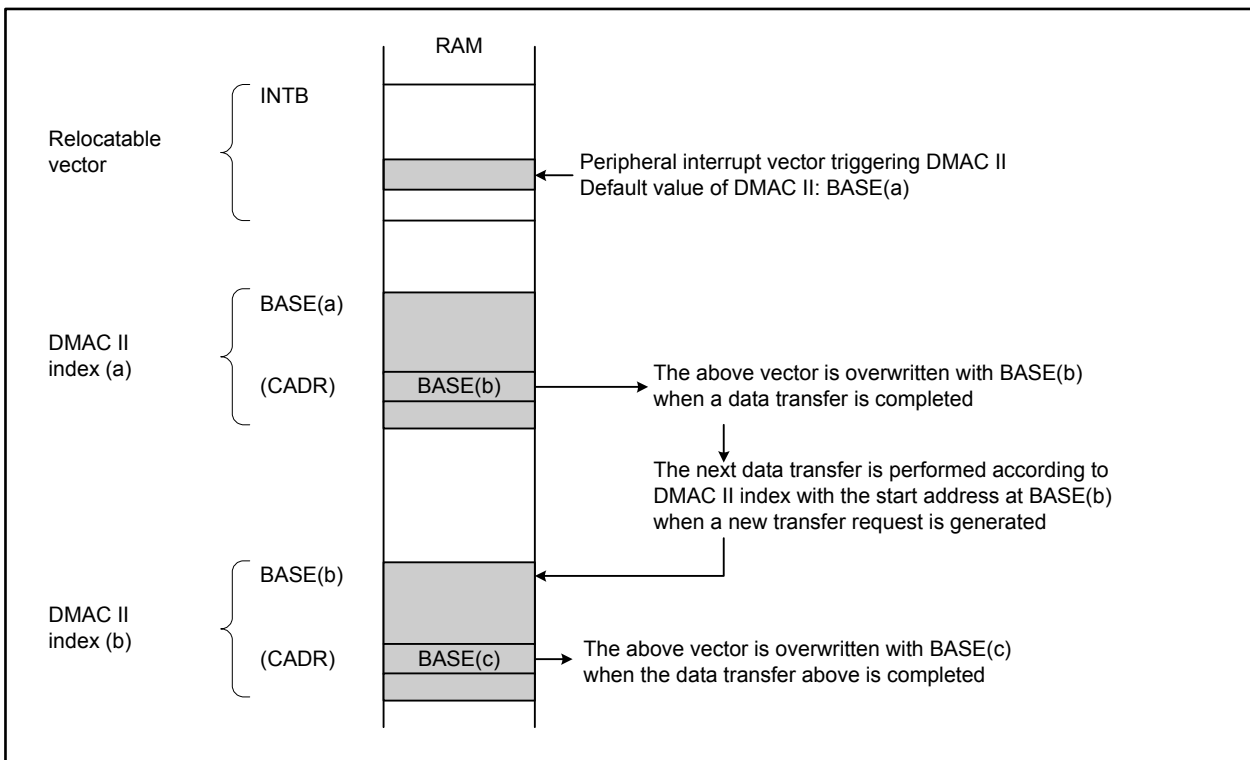
The chain transfer is available when the CHAIN bit in the MOD is 1.

The chain transfer is performed as follows:

- (1) When a transfer request is generated, a data transfer is performed according to the DMAC II index specified by the corresponding interrupt vector. Either a single transfer (the BRST bit in the MOD is 0) or burst transfer (the BRST bit is 1) is performed according to the BRST bit setting.
- (2) When COUNT reaches 0000h, the value in the interrupt vector in (1) above is overwritten with the value in CADR. Simultaneously, the DMA II transfer complete interrupt request is generated when the INTE bit in the MOD is 1.
- (3) When the next DMA II transfer request is generated, the data transfer is performed according to the DMAC II index specified by the peripheral interrupt vector in (2) above.

Figure 14.4 shows the relocatable vector and DMAC II index in a chain transfer.

To use the chain transfer, the relocatable vector table should be allocated on the RAM.



**Figure 14.4 Relocatable Vector and DMAC II Index in a Chain Transfer**

## 14.6 DMA II Transfer Complete Interrupt

The DMA II transfer complete interrupt is available when the INTE bit in the MOD is 1.

Set the start address of the DMA II transfer complete interrupt handler to IADR. The interrupt request is generated when COUNT reaches 0000h.

The initial instruction of the interrupt handler is executed in the eighth cycle after a DMA II transfer is completed.

## 14.7 Execution Time

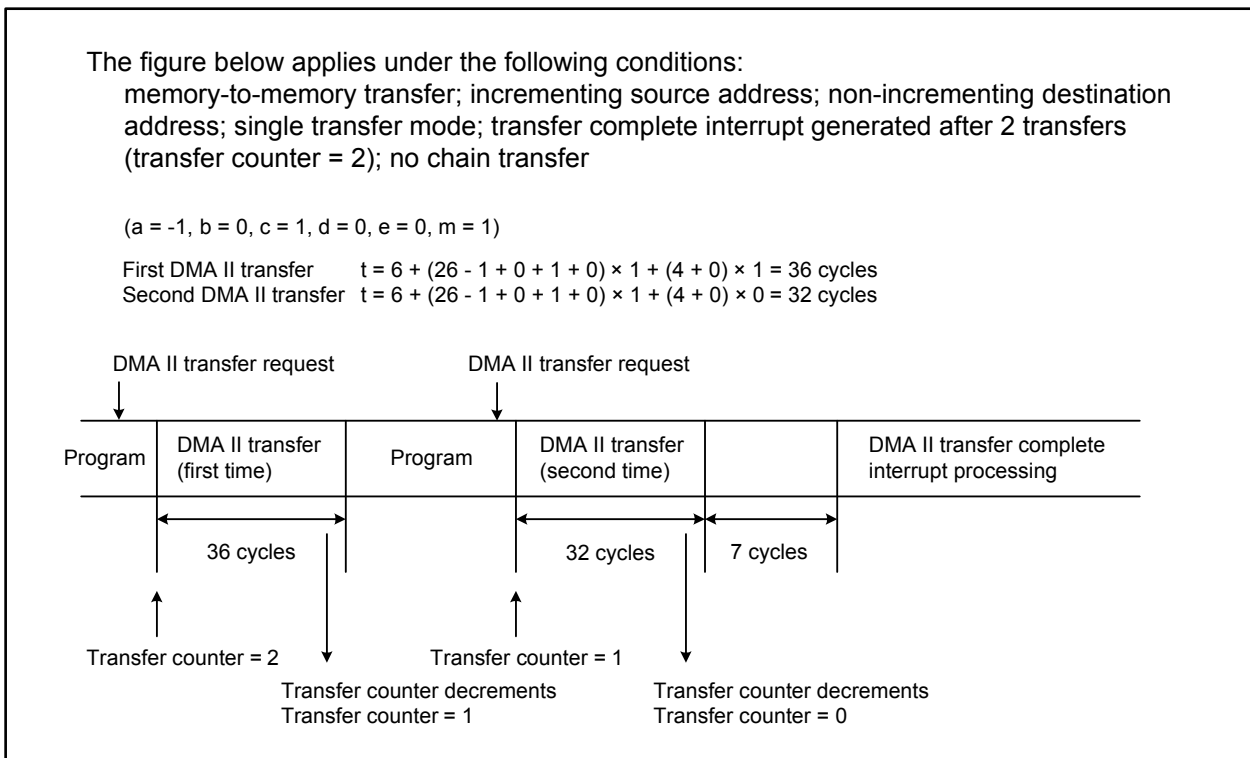
The DMAC II execution cycle is calculated by the following equations:

Mode other than multiple transfer:  $t = 6 + (26 + a + b + c + d) \times m + (4 + e) \times n$  cycles

When using multiple transfer:  $t = 21 + (11 + b + c) \times k$  cycles

- a: When IMM is 0 (transfer source is immediate data), a is 0;  
When IMM is 1 (transfer source is memory), a is -1
- b: When UPDS is 1 (source addressing is incrementing), b is 0;  
When UPDS is 0 (source addressing is non-incrementing), b is 1
- c: When UPDD is 1 (destination addressing is incrementing), c is 0;  
When UPDD is 0 (destination addressing is non-incrementing), c is 1
- d: When OPER is 0 (calculation transfer is not selected), d is 0;  
When OPER is 1 (calculation transfer is selected) and UPDS is 0 (source addressing is immediate data or non-incrementing), d is 7;  
When OPER is 1 (calculation transfer is selected) and UPDS is 1 (source addressing is incrementing), d is 8
- e: When CHAIN is 0 (chain transfer is not selected), e is 0;  
When CHAIN is 1 (chain transfer is selected), e is 4
- m: When BRST is 0 (single transfer), m is 1;  
When BRST is 1 (burst transfer), m is COUNT
- n: When COUNT is 0001h, n is 0; if COUNT is 0002h or more, n is 1
- k: The number of transfers set using bits CNT2 to CNT0

The equations above are estimations. The number of cycles may vary depending on CPU state, bus wait state, and DMAC II index allocation.



**Figure 14.5 Transfer Cycles**

## 15. Programmable I/O Ports

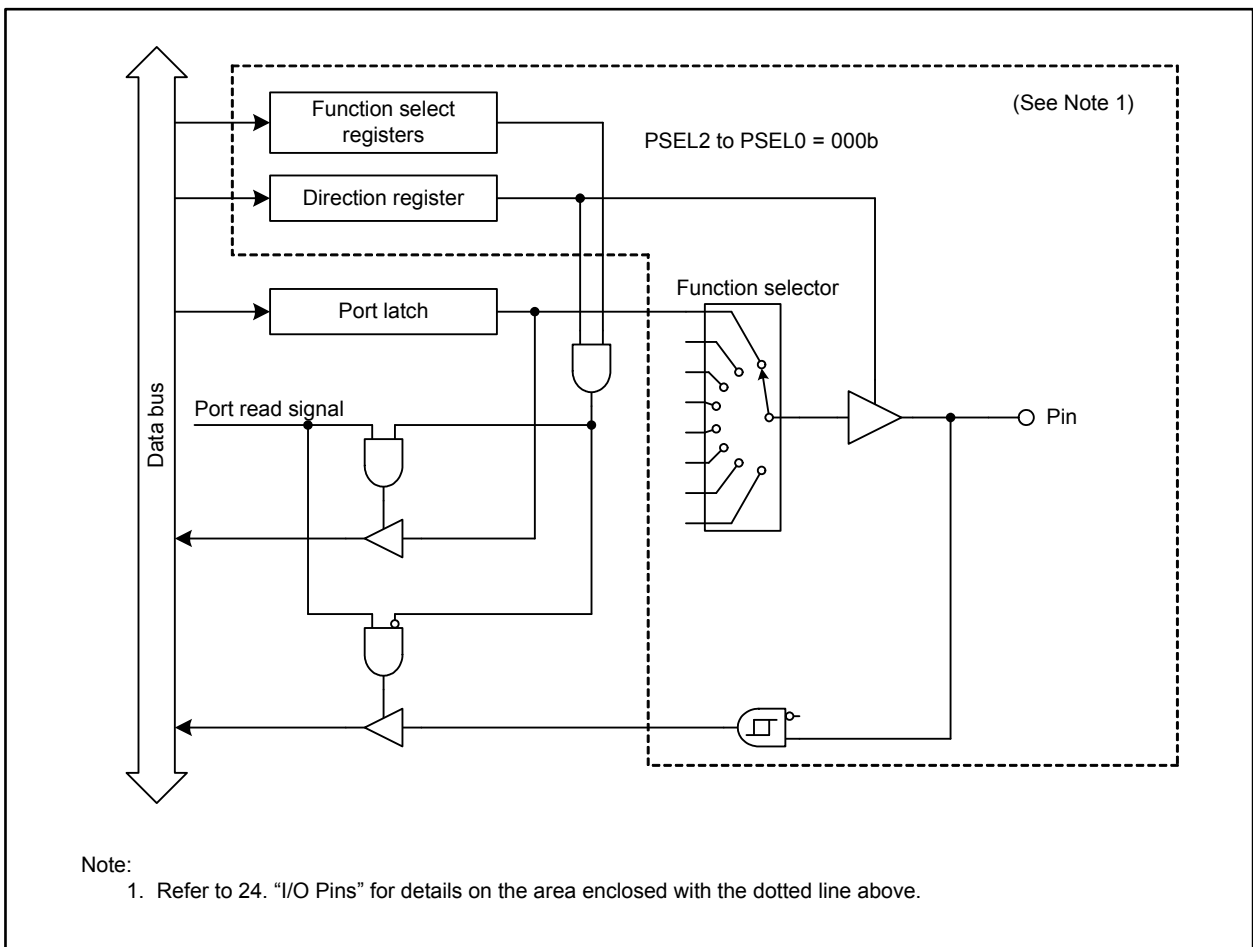
The programmable I/O ports in each pin package are designated as follows:

64-pin package: 51 ports from P0 to P3 (excluding P0\_4 to P0\_7, P1\_0 to P1\_4, and P3\_4 to P3\_7), and P6 to P10 (excluding P8\_5, P9\_0 to P9\_2, and P9\_4 to P9\_7)

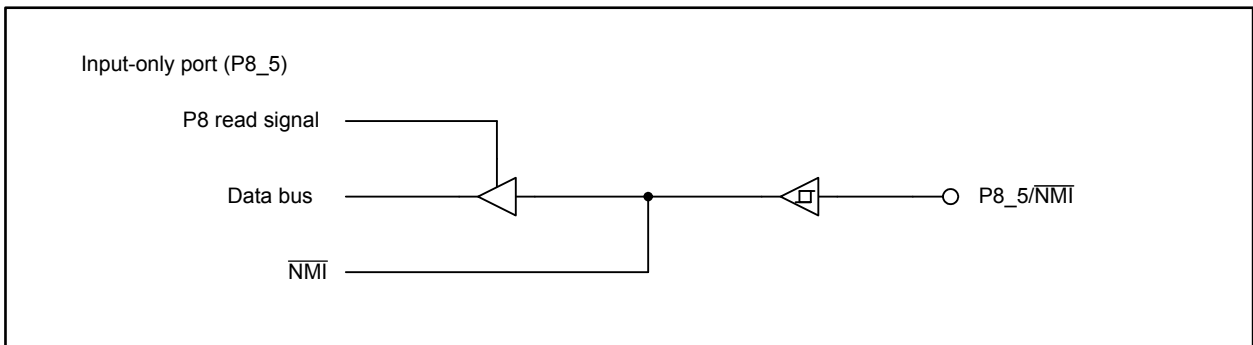
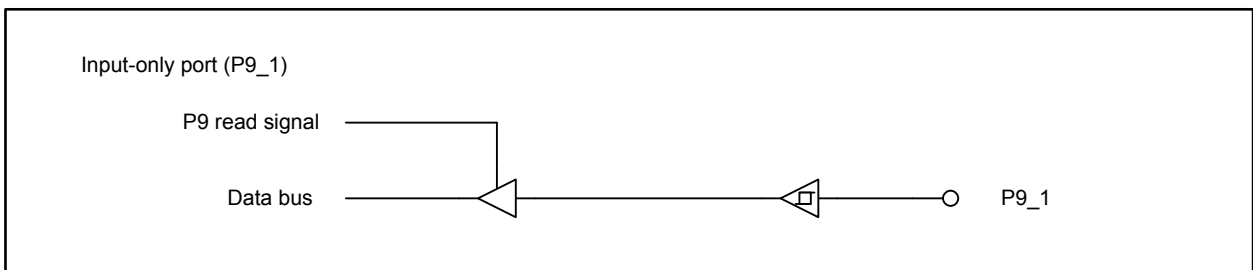
100-pin package: 84 ports from P0 to P10 (excluding P8\_5 and P9\_0 to P9\_2)

Each port status, input or output, can be selected using the direction register except P8\_5 and P9\_1 which are input only. The P8\_5 bit in the P8 register indicates an NMI input level since the P8\_5 shares a pin with the NMI.

Figure 15.1 shows a configuration of programmable I/O ports, and Figures 15.2 and 15.3 show a configuration of each input-only port.



**Figure 15.1 Programmable I/O Port Configuration**

**Figure 15.2 Input-only Port Configuration (1/2)****Figure 15.3 Input-only Port Configuration (2/2) (in the 100-pin package only)**

## 15.1 Port Pi Register (Pi register, i = 0 to 10)

A write/read operation to the Pi register is required to communicate with external devices. This register consists of a port latch to hold output data and a circuit to read pin states. Bits in the Pi register correspond to respective ports.

When a programmable I/O port is selected in the output function select registers, the value in the port latch is read for output and the pin state is read for input.

In memory expansion mode and microprocessor mode, this register cannot control pins being assigned bus control signals ( $\overline{A0}$  to  $\overline{A23}$ ,  $\overline{D0}$  to  $\overline{D15}$ ,  $\overline{CS0}$  to  $\overline{CS3}$ ,  $\overline{WR}/\overline{WR0}$ ,  $\overline{BC0}$ ,  $\overline{BC1}/\overline{WR1}$ ,  $\overline{RD}$ ,  $\overline{CLKOUT}/\overline{BCLK}$ ,  $\overline{HLDA}$ ,  $\overline{HOLD}$ ,  $\overline{ALE}$ , and  $\overline{RDY}$ ).

Figure 15.4 shows the Pi register.

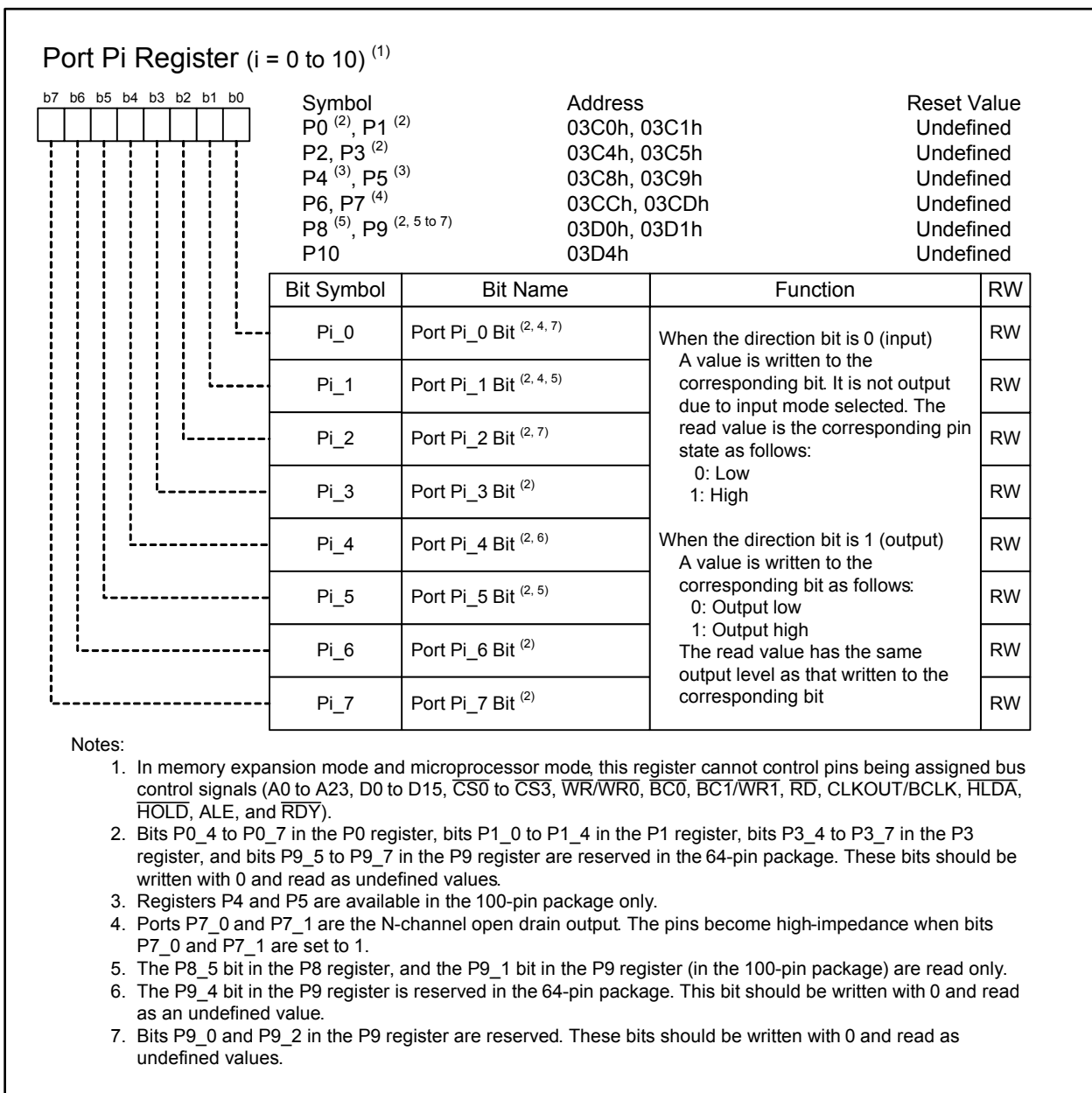
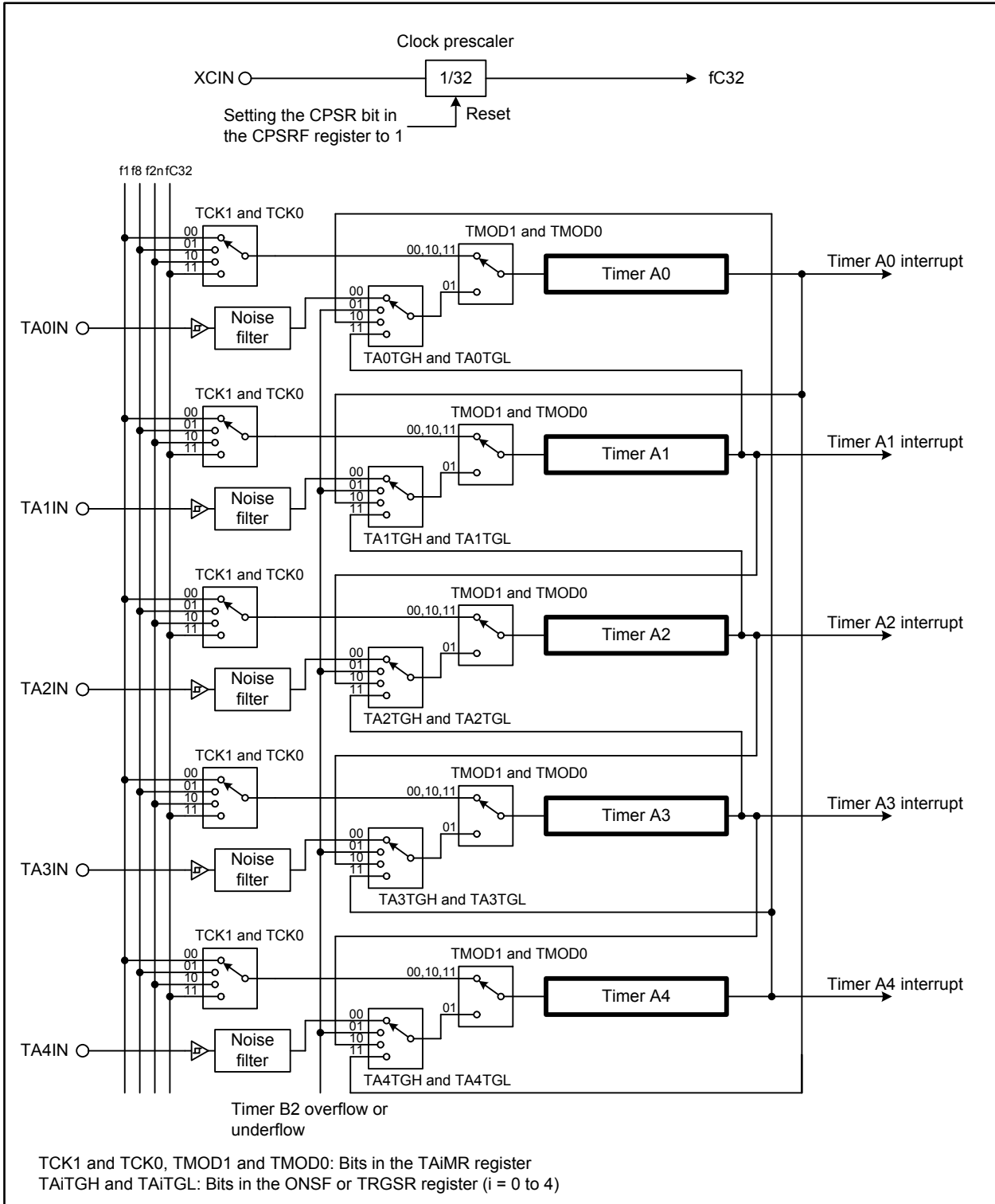


Figure 15.4 Registers P0 to P10

## 16. Timers

This MCU has eleven 16-bit timers which are divided into two groups according to their functions: five timer As and six timer Bs. Each timer functions individually. The count source of each timer provides the clock for timer operations such as counting and reloading.

Figures 16.1 and 16.2 show the configuration of timers A and B, respectively.



**Figure 16.1** Timer A Configuration

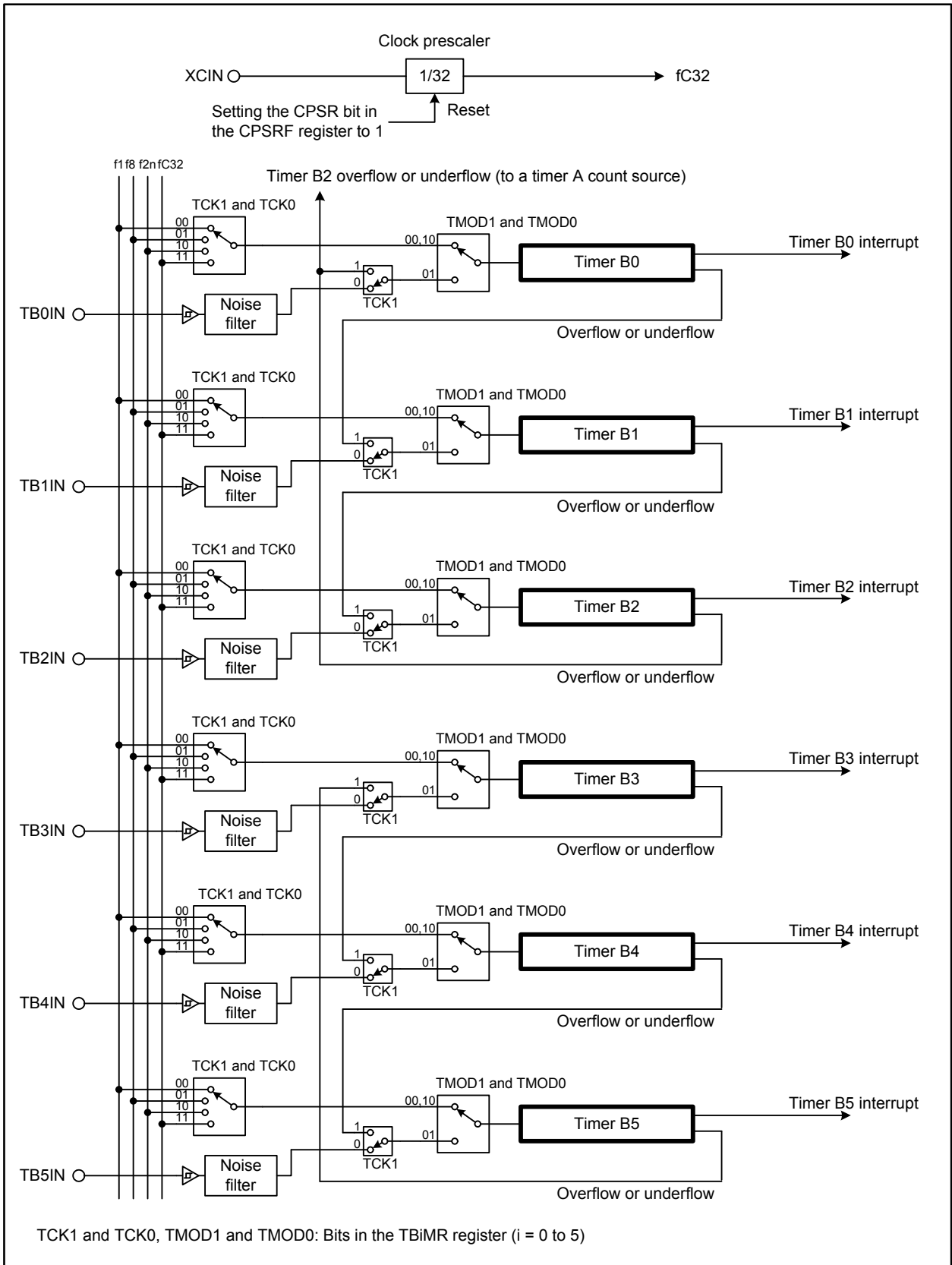


Figure 16.2 Timer B Configuration

## 16.1 Timer A

Figure 16.3 shows a block diagram of timer A and Figure 16.4 to Figure 16.10 show registers associated with timer A.

Timer A supports the four modes shown below. Timers A0 to A4 in any mode other than the event counter mode have the same function. Select a mode by setting bits TMOD1 and TMOD0 in the TAI<sub>i</sub>MR register (i = 0 to 4).

- Timer mode: The timer counts an internal count source
- Event counter mode: The timer counts an external pulse or overflow and underflow of other timers
- One-shot timer mode: The timer outputs a pulse after a trigger input until the counter reaches 0000h
- Pulse-width modulation mode: The timer successively outputs pulses of a given width

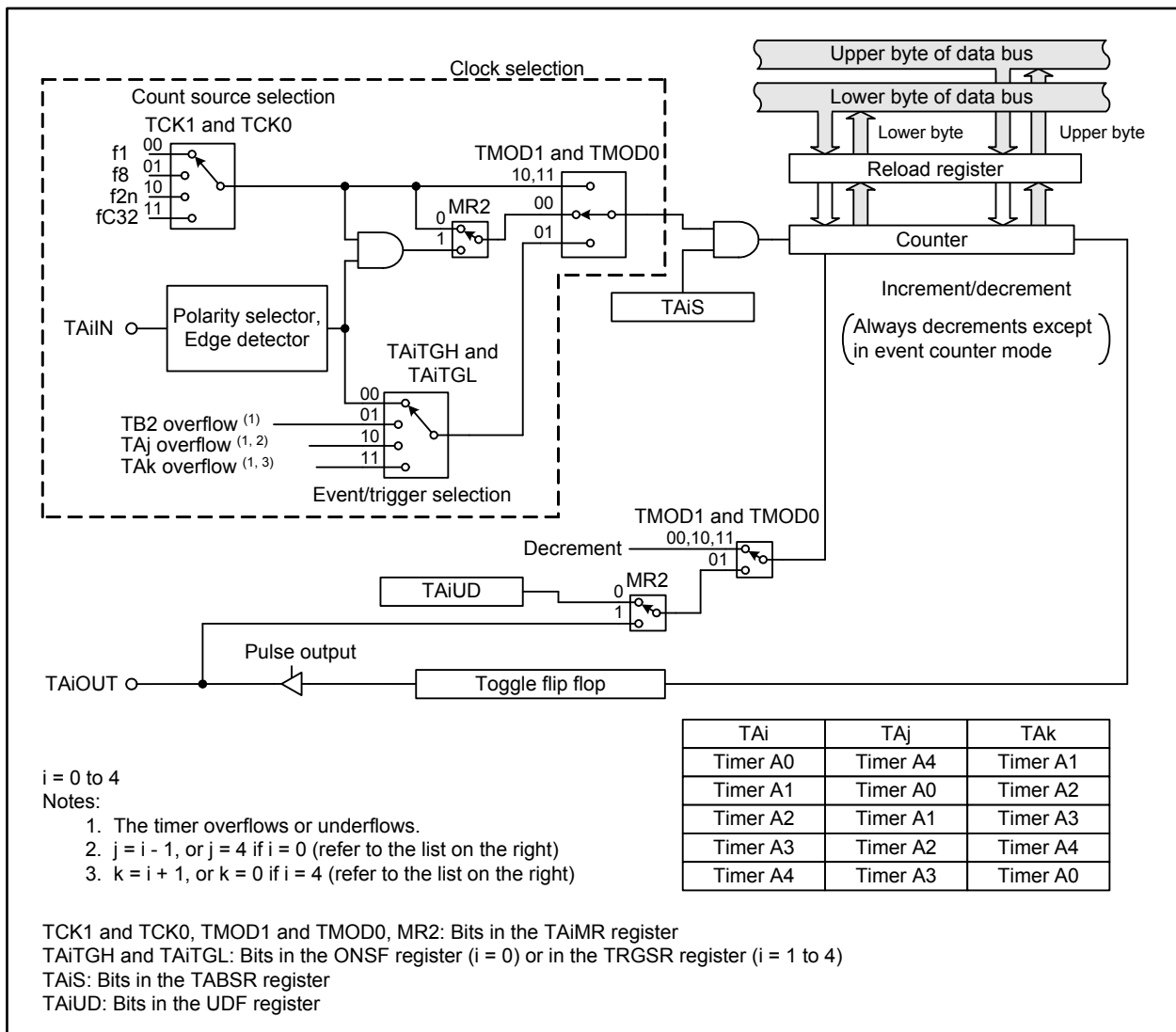
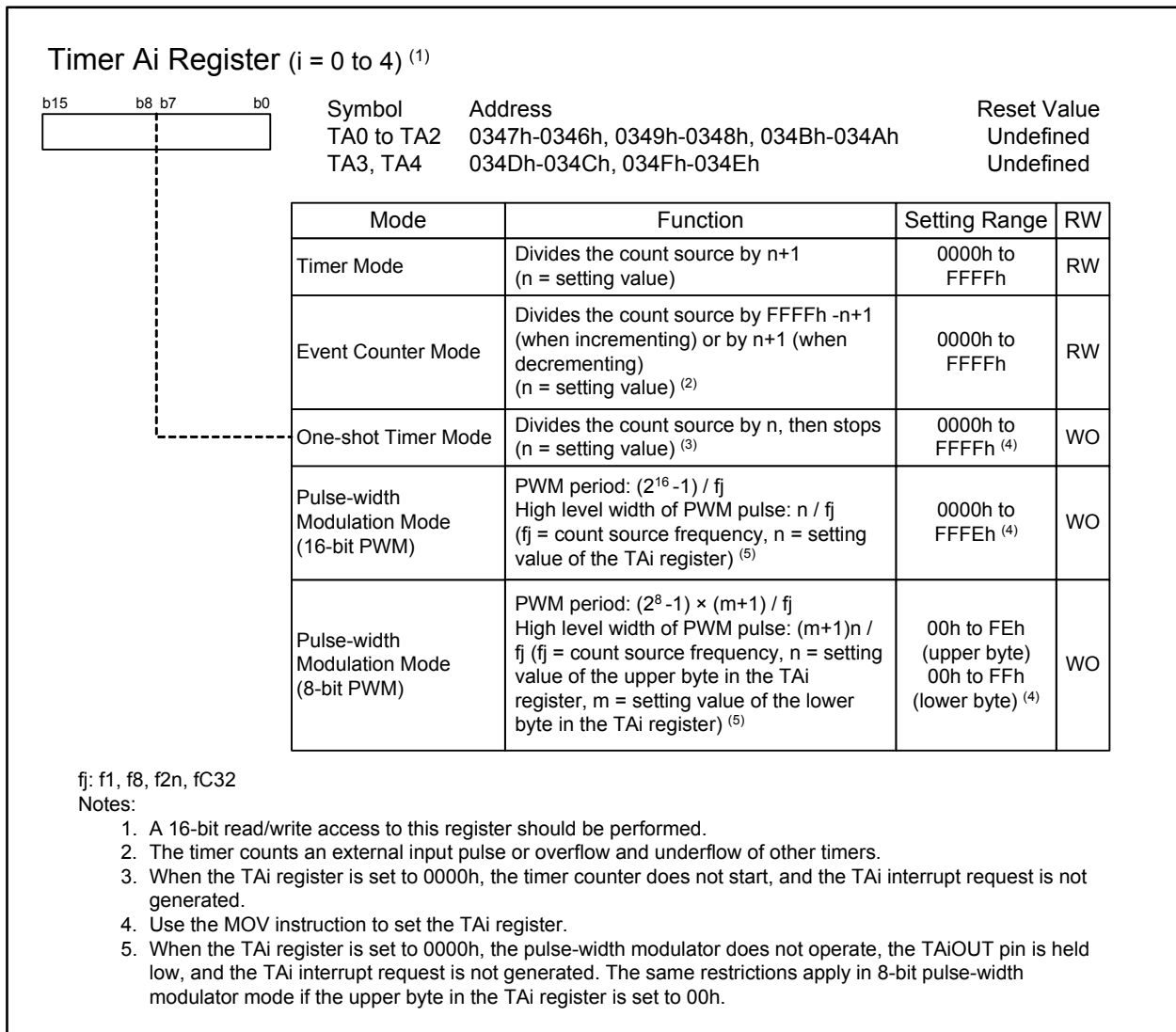


Figure 16.3 Timer A Block Diagram





**Figure 16.4 Registers TA0 to TA4**

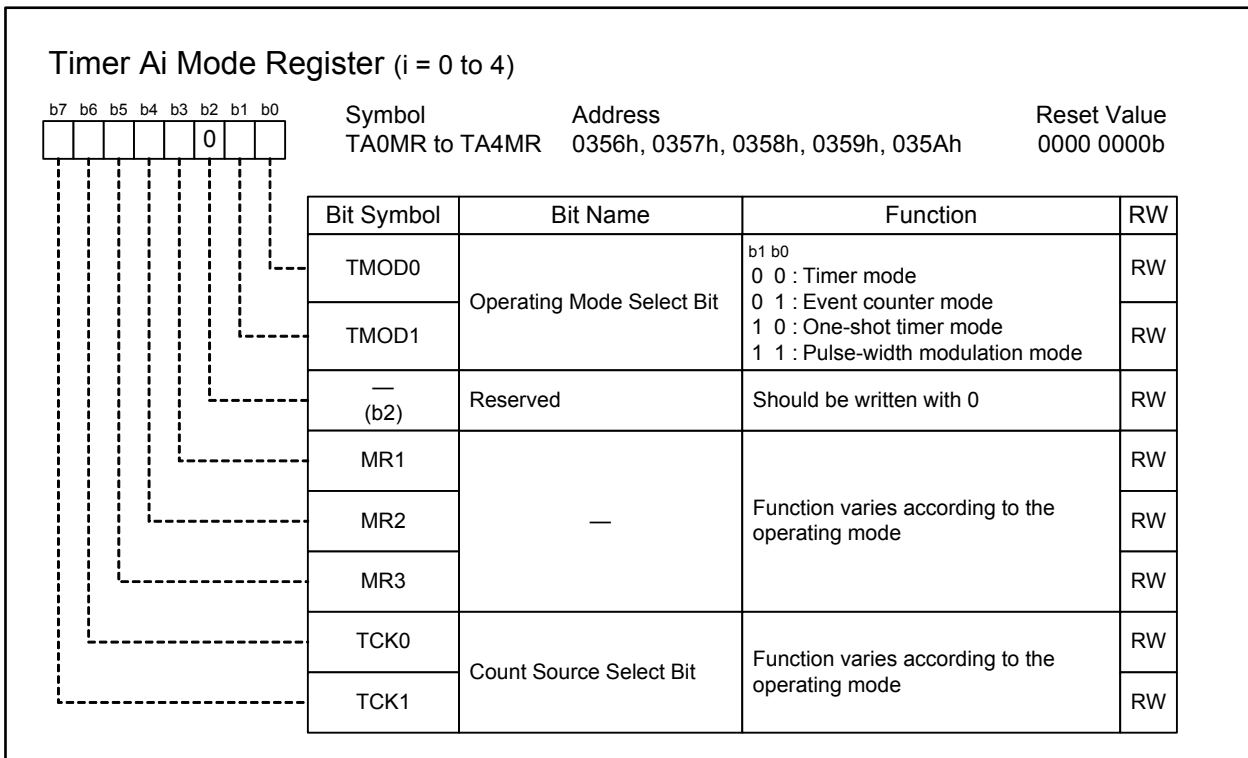


Figure 16.5 Registers TA0MR to TA4MR

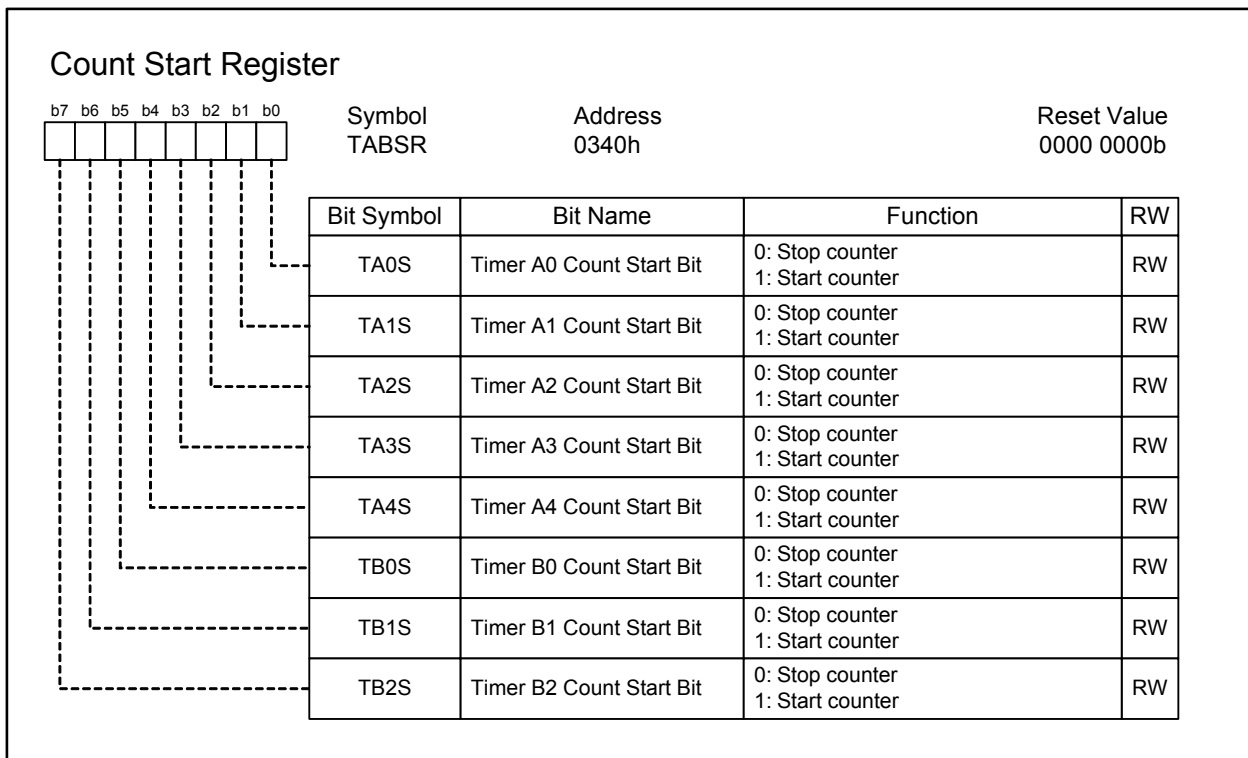


Figure 16.6 TABSR Register

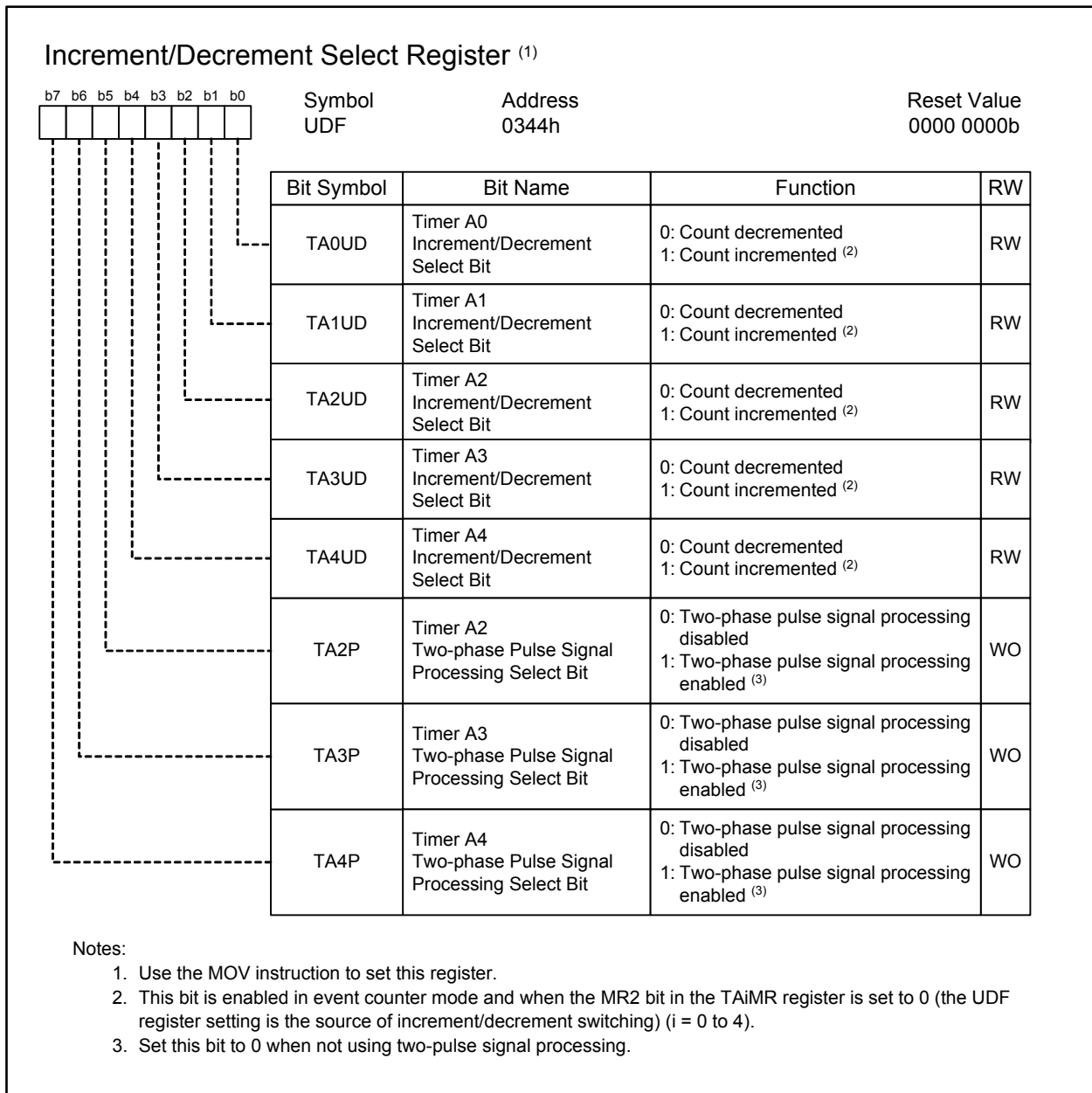


Figure 16.7 UDF Register

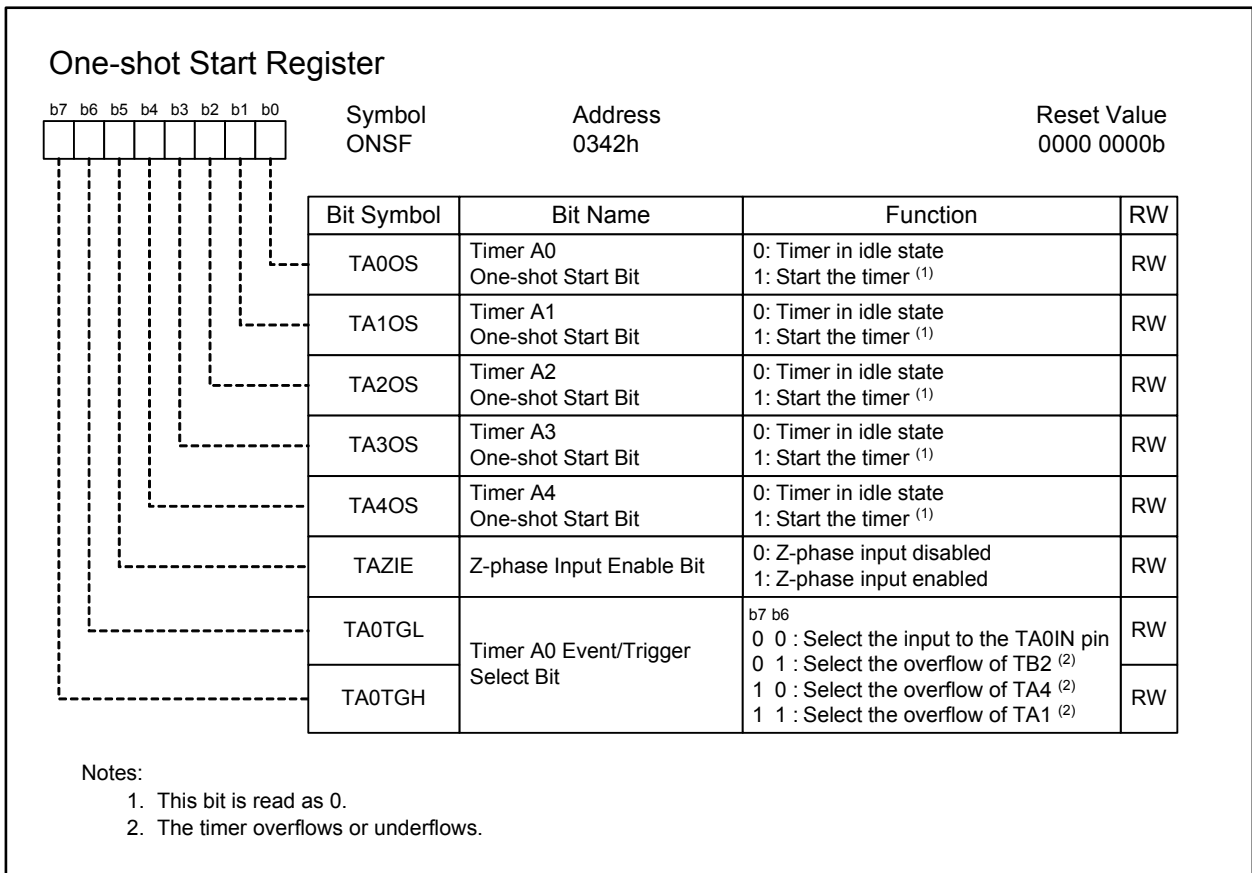


Figure 16.8 ONSF Register

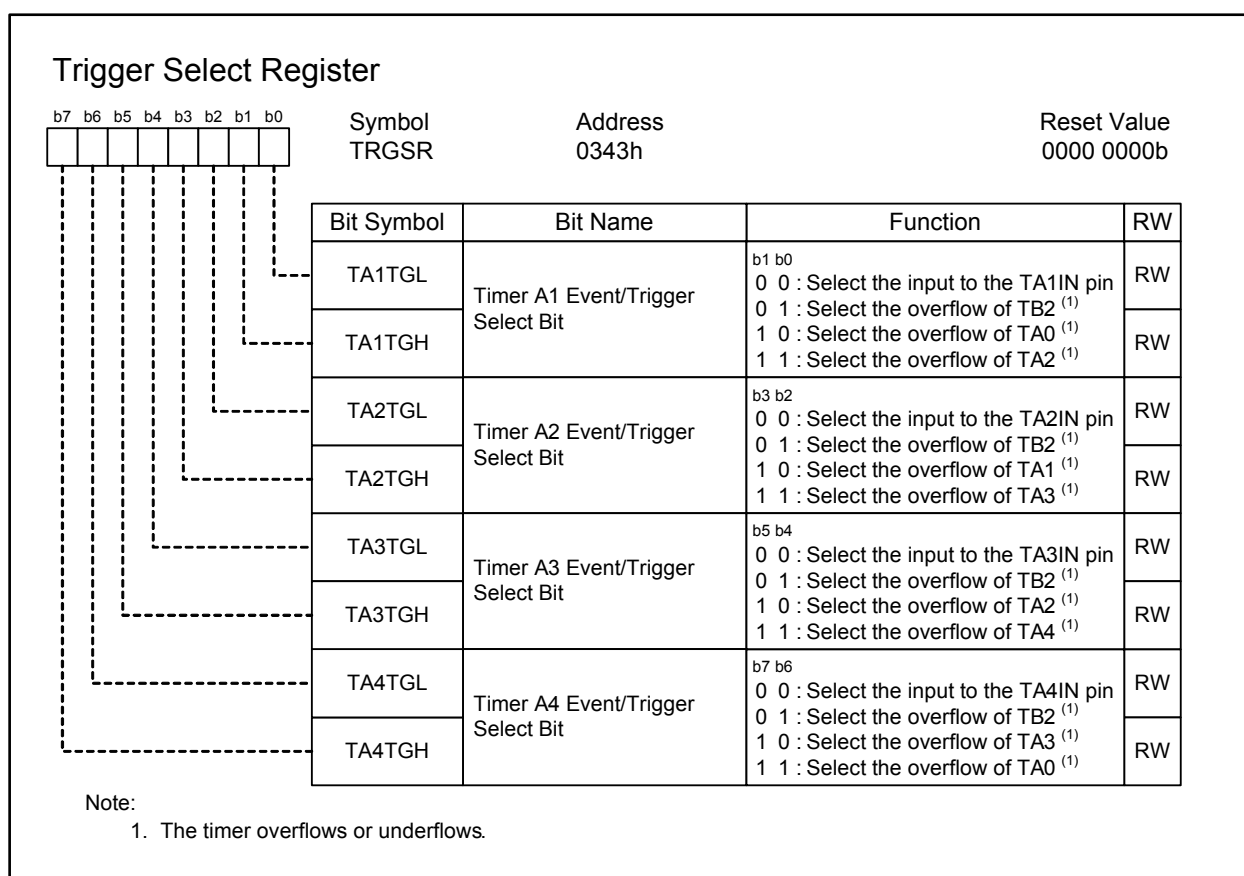


Figure 16.9 TRGSR Register

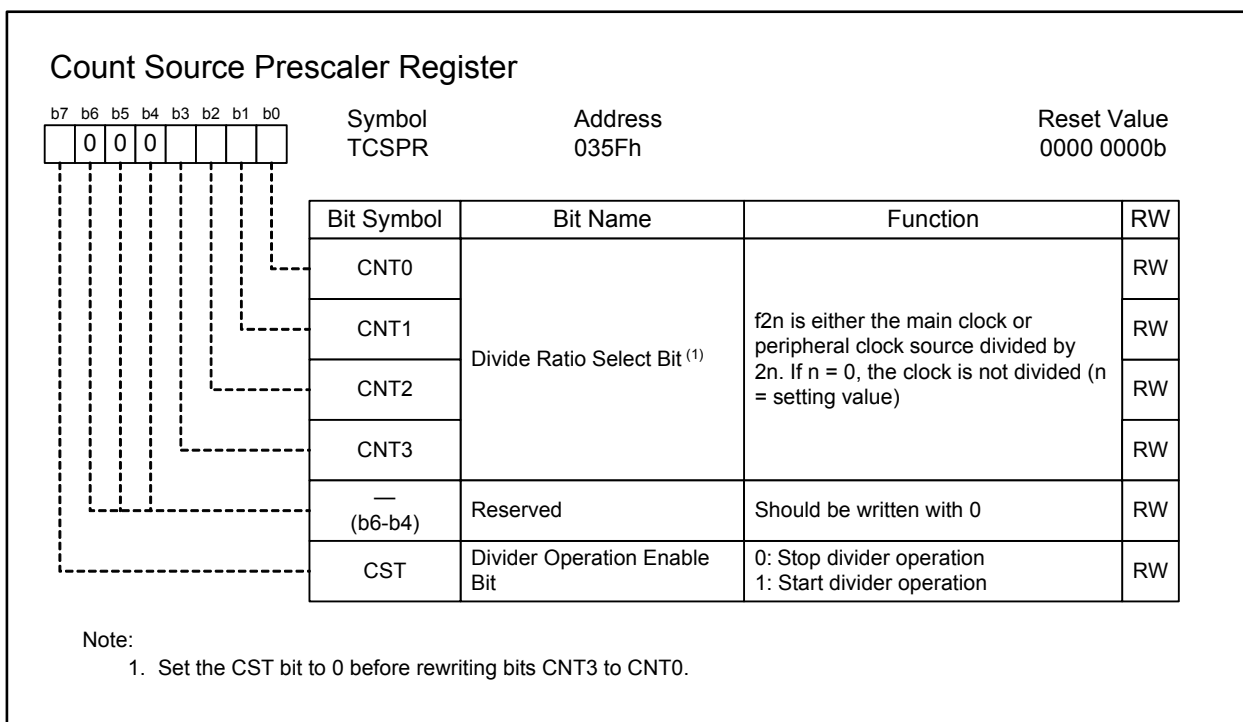


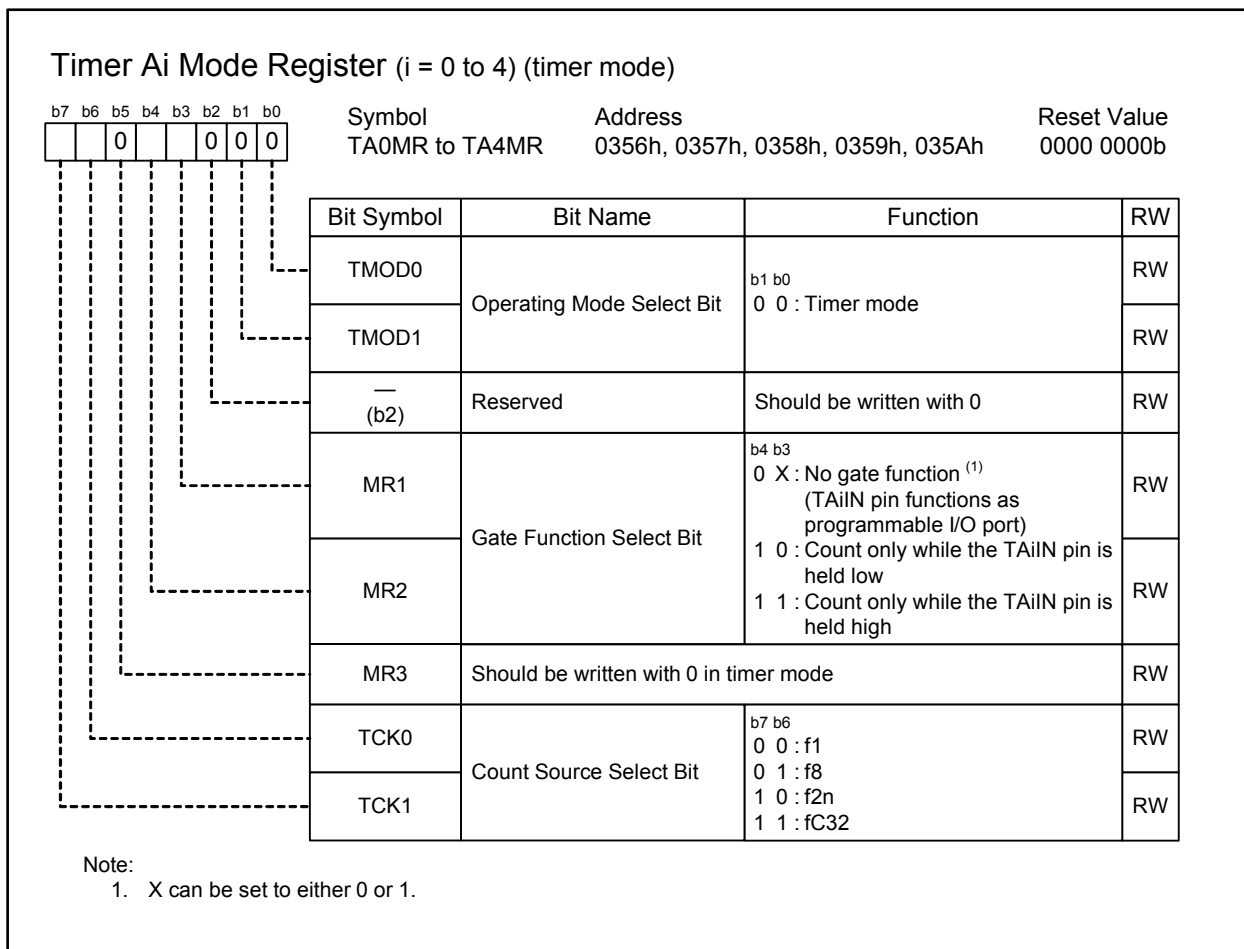
Figure 16.10 TCSPR Register

### 16.1.1 Timer Mode

In timer mode, the timer counts an internally generated count source. Table 16.1 lists the specifications of timer mode. Figure 16.11 shows registers TA0MR to TA4MR in this mode.

**Table 16.1 Timer Mode Specifications (i = 0 to 4)**

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	<ul style="list-style-type: none"> <li>• Decrement</li> <li>• When the timer counter underflows, the reload register value is reloaded into the counter to continue counting</li> </ul>
Divide ratio	$\frac{1}{n+1}$ n: TAI register setting value, 0000h to FFFFh
Count start condition	The TAI <sub>S</sub> bit in the TABSR register is 1 (start counter)
Count stop condition	The TAI <sub>S</sub> bit in the TABSR register is 0 (stop counter)
Interrupt request generating timing	When the timer counter underflows
TAiIN pin function	Functions as a programmable I/O port or a gate input
TAiOUT pin function	Functions as a programmable I/O port or a pulse output
Read from timer	The TAI register indicates the counter value
Write to timer	<ul style="list-style-type: none"> <li>• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAI register is written to both the reload register and the counter</li> <li>• While the timer counter is running, the value written to the TAI register is written to the reload register (it is transferred to the counter at the next reload timing)</li> </ul>
Other functions	<ul style="list-style-type: none"> <li>• Gate function Input signal to the TAI<sub>IN</sub> pin can control the count start/stop</li> <li>• Pulse output function The polarity of the TAI<sub>OUT</sub> pin is inverted each time the timer counter underflows. A low is output while the TAI<sub>S</sub> bit holds 0 (stop counter)</li> </ul>



**Figure 16.11 Registers TA0MR to TA4MR in Timer Mode**



### 16.1.2 Event Counter Mode

In event counter mode, the timer counts an external signal or an overflow and underflow of other timers. Timers A2, A3, and A4 can count two-phase external signals. Table 16.2 lists the specifications in event count mode and Table 16.3 also lists the specifications when the timers use two-phase pulse signal processing. Figure 16.12 shows registers TA0MR to TA4MR in this mode.

**Table 16.2 Event Counter Mode Specifications (without two-phase pulse signal processing)  
(i = 0 to 4)**

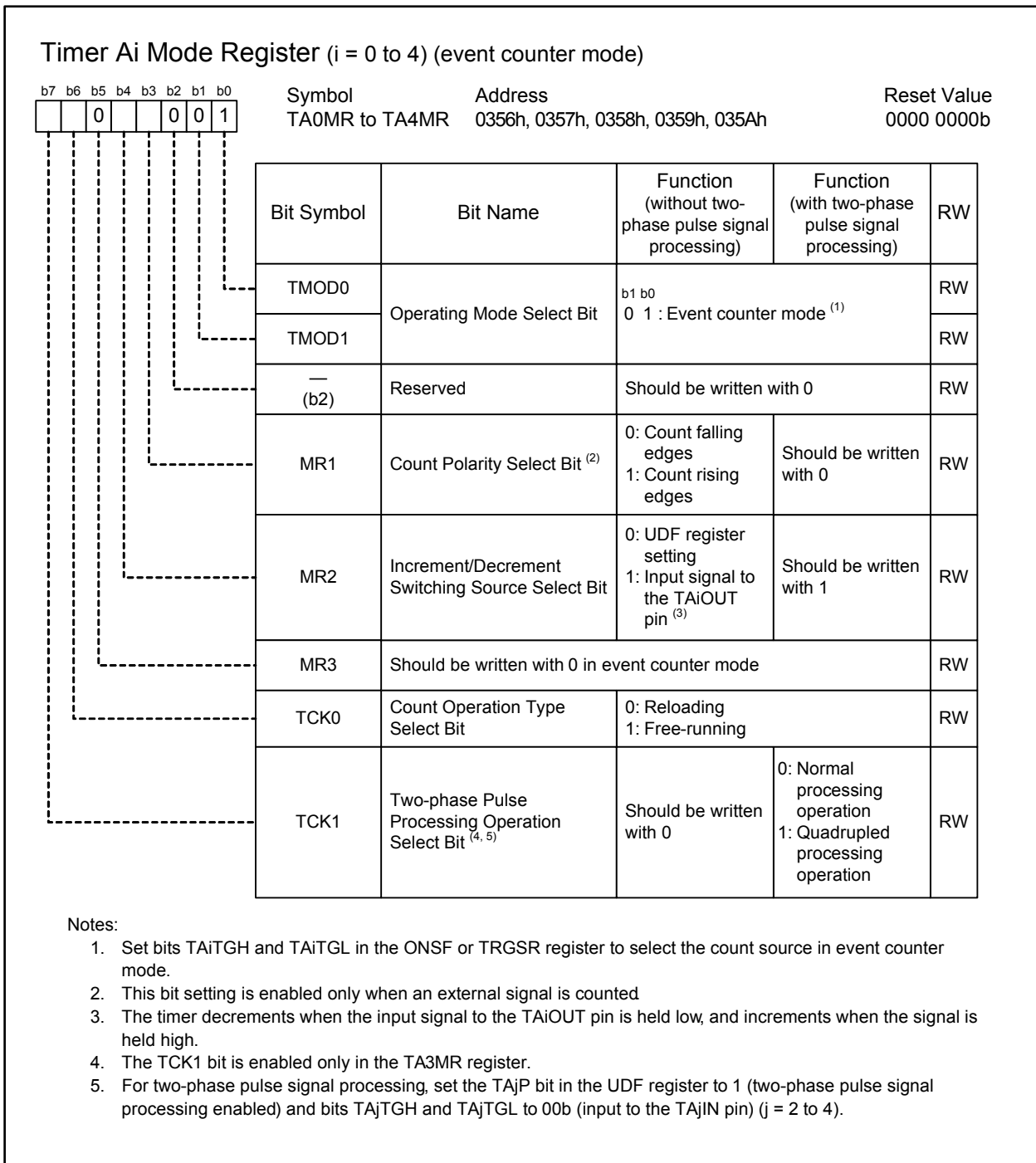
Item	Specification
Count sources	<ul style="list-style-type: none"> <li>External signal applied to the TAIIN pin (valid edge is selectable by a program)</li> <li>One of the following: the overflow and/or underflow signal of timer B2, the overflow and/or underflow signal of timer Aj (j = i - 1, or j = 4 if i = 0), or the overflow and/or underflow signal of timer Ak (k = i + 1, or k = 0 if i = 4)</li> </ul>
Count operations	<ul style="list-style-type: none"> <li>Increment/decrement can be switched by an external signal or program</li> <li>When the timer counter underflows or overflows, the reload register value is reloaded into the counter to continue counting. In a free-running count operation, the timer counter continues counting without reloading</li> </ul>
Divide ratio	<ul style="list-style-type: none"> <li><math>\frac{1}{FFFFh - n + 1}</math> when incrementing</li> <li><math>\frac{1}{n + 1}</math> when decrementing</li> </ul> n: TAI register setting value, 0000h to FFFFh
Count start condition	The TAI S bit in the TABSR register is 1 (start counter)
Count stop condition	The TAI S bit in the TABSR register is 0 (stop counter)
Interrupt request generating timing	When the timer counter overflows or underflows
TAiIN pin function	Functions as a programmable I/O port or a count source input
TAiOUT pin function	Functions as a programmable I/O port, a pulse output, or an input for switching between increment/decrement
Read from timer	The TAI register indicates a counter value
Write to timer	<ul style="list-style-type: none"> <li>While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAI register is written to both the reload register and the counter</li> <li>While the timer counter is running, the value written to the TAI register is written to the reload register (it is transferred to the counter at the next reload timing)</li> </ul>
Other functions	<ul style="list-style-type: none"> <li>Free-running count function The reload register value is not reloaded even if the timer counter overflows or underflows</li> <li>Pulse output function The polarity of the TAIOUT pin is inverted whenever the timer counter overflows or underflows. A low is output while the TAI S bit holds 0 (stop counter)</li> </ul>

**Table 16.3 Event Counter Mode Specifications (with two-phase pulse signal processing on timers A2 to A4) (i = 2 to 4)**

Item	Specification
Count sources	Two-phase pulse signal applied to pins TAIIN and TAIOUT
Count operations	<ul style="list-style-type: none"> <li>Increment/decrement can be switched by a two-phase pulse signal</li> <li>When the timer counter underflows or overflows, the reload register value is reloaded into the counter to continue counting. In a free-running count operation, the timer counter continues counting without reloading</li> </ul>
Divide ratio	<ul style="list-style-type: none"> <li><math>\frac{1}{FFFFh - n + 1}</math> when incrementing</li> <li><math>\frac{1}{n + 1}</math> when decrementing</li> </ul> <i>n</i> : TAI register setting value, 0000h to FFFFh
Count start condition	The TAI <sub>S</sub> bit in the TABSR register is 1 (start counter)
Count stop condition	The TAI <sub>S</sub> bit in the TABSR register is 0 (stop counter)
Interrupt request generating timing	When the timer counter overflows or underflows
TAiIN pin function	A two-phase pulse input
TAiOUT pin function	A two-phase pulse input
Read from timer	The TAI register indicates a counter value
Write to timer	<ul style="list-style-type: none"> <li>While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAI register is written to both the reload register and the counter</li> <li>While the timer counter is running, the value written to the TAI register is written to the reload register (it is transferred to the counter at the next reload timing)</li> </ul>
Other functions (1)	<ul style="list-style-type: none"> <li>Normal processing operation (timers A2 and A3) While the input signal applied to the TAJOUT pin is held high, the timer increments on the rising edge of the TAJIN pin and decrements on the falling edge (j = 2 or 3)</li> </ul> <p style="text-align: right;">IC: Increments DC: Decrements</p> <ul style="list-style-type: none"> <li>Quadrupled processing operation (timers A3 and A4) When the input signal applied to the TAKOUT pin is held high on the rising edge of the TAKIN pin, the timer increments on both the rising and falling edges of pins TAKOUT and TAKIN (k = 3 or 4). When the signal is held high on the falling edge of the TAKIN pin, the timer decrements on both the rising and falling edges of pins TAKOUT and TAKIN</li> </ul> <ul style="list-style-type: none"> <li>Counter reset by Z-phase input (timer A3) The counter value is set to 0 by Z-phase input</li> </ul>

Note:

- Only timer A3 is available for any of the other functions. Timer A2 is exclusively for normal processing operations and timer A4 is for the quadrupled processing operation.



**Figure 16.12 Registers TA0MR to TA4MR in Event Counter Mode**

### 16.1.2.1 Counter Reset by Two-phase Pulse Signal Processing

A Z-phase input signal resets the timer counter when a two-phase pulse signal is being processed.

This function can be used under the following conditions: timer A3 event counter mode, two-phase pulse signal processing, free-running count operation, and quadrupled processing. The Z-phase signal is applied to the  $\overline{\text{INT2}}$  pin.

When the TAZIE bit in the ONSF register is set to 1 (Z-phase input enabled), the timer counter can be reset by Z-phase input. To reset the counter, set the TA3 register to 0000h beforehand.

A Z-phase signal applied to the  $\overline{\text{INT2}}$  pin is detected on an edge. The edge polarity is selected using the POL bit in the INT2IC register. The Z-phase signal should be input in order to have a pulse width of at least one count source cycle for timer A3. Figure 16.13 shows the two-phase pulse (phases A and B) and the Z-phase.

The timer counter is reset at the initial count source input after Z-phase input is detected. Figure 16.14 shows the counter reset timing.

When timer A3 overflows or underflows during a reset by the Z-phase input, two timer A3 interrupt requests are successively generated. To avoid this, the timer A3 interrupt request should not be used when using this function.

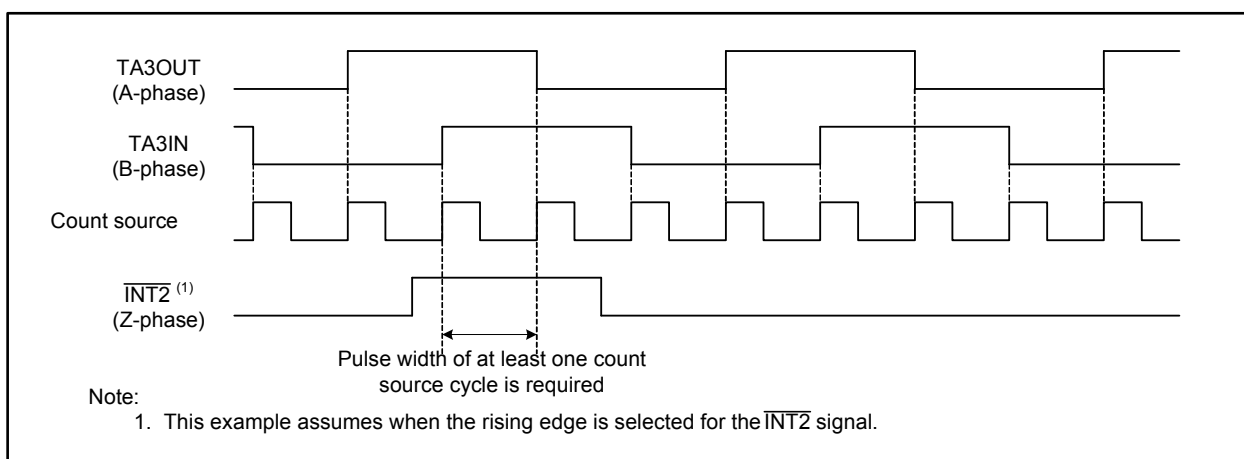


Figure 16.13 Two-phase Pulse (phases A and B) and Z-phase

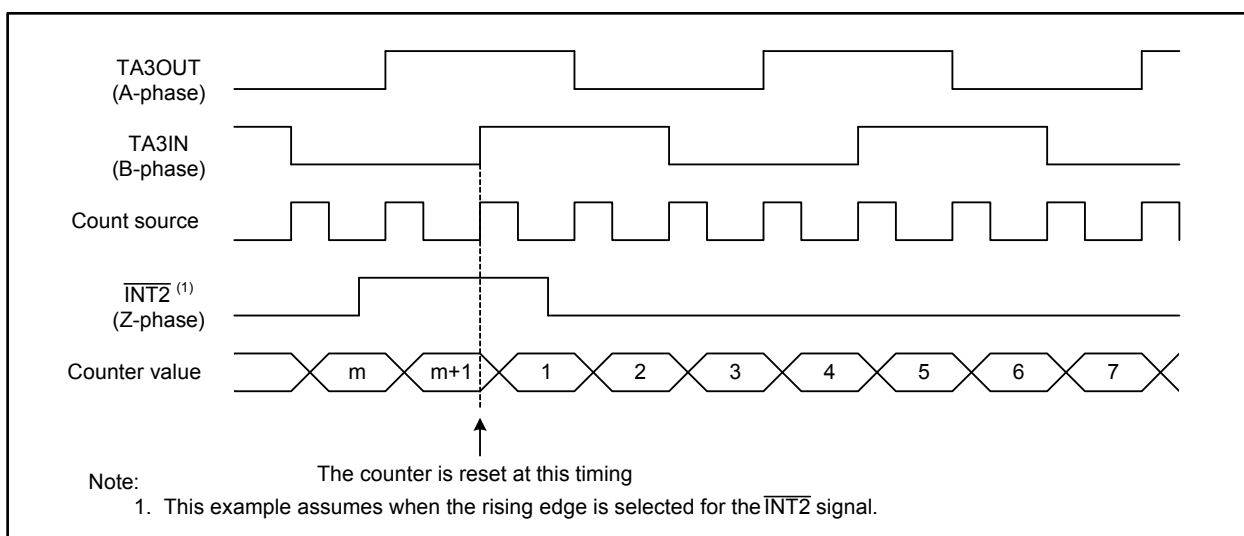


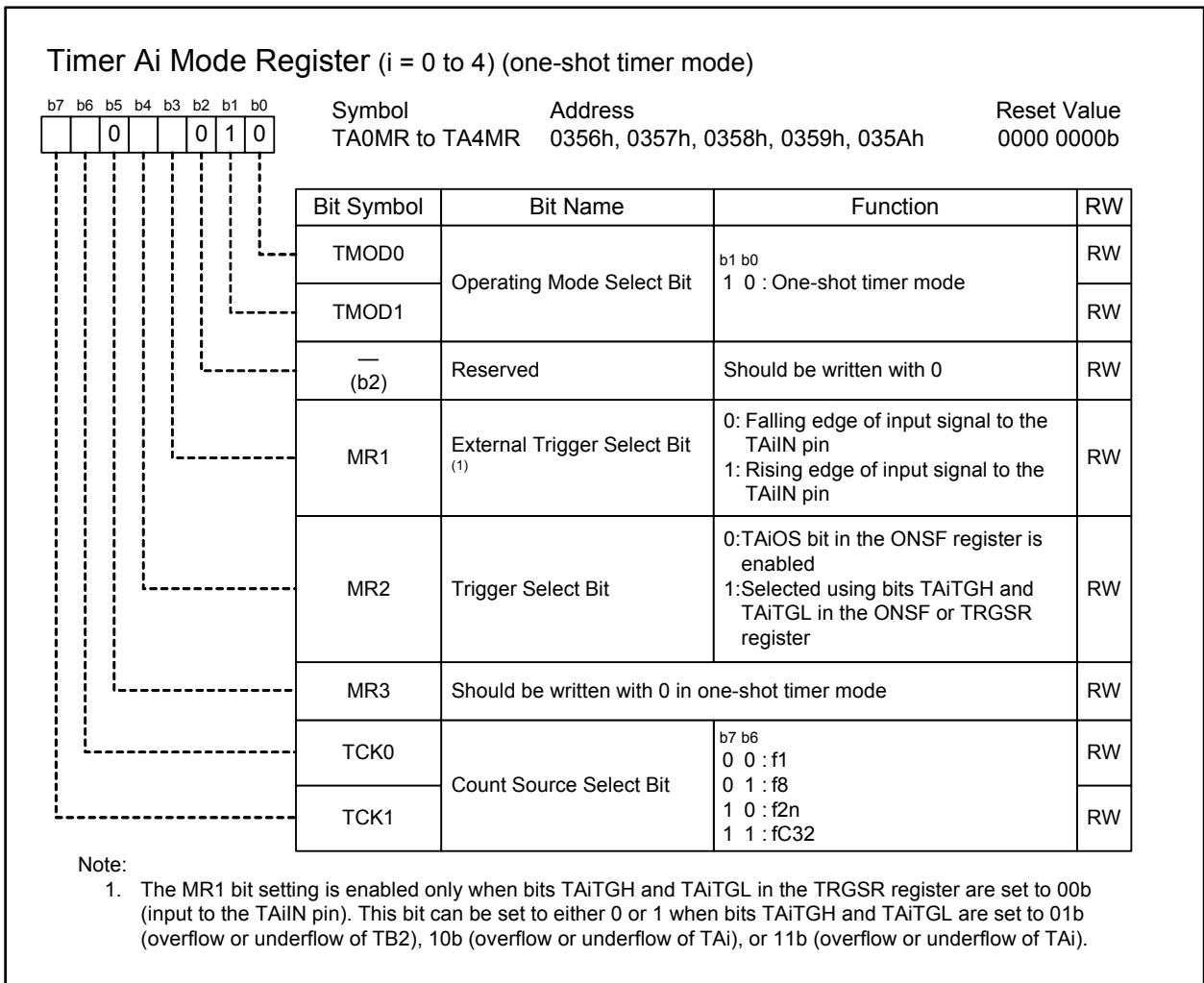
Figure 16.14 Counter Reset Timing

### 16.1.3 One-shot Timer Mode

In one-shot timer mode, the timer operates only once for each trigger. Table 16.4 lists specifications of one-shot timer mode. Once a trigger occurs, the timer starts and operates for a given period. Figure 16.15 shows registers TA0MR to TA4MR in this mode.

**Table 16.4 One-shot Timer Mode Specifications (i = 0 to 4)**

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	<ul style="list-style-type: none"> <li>• Decrement</li> <li>• When the timer counter reaches 0000h, it stops running after the reload register value is reloaded</li> <li>• When a trigger occurs while counting, the reload register value is reloaded into the counter to continue counting</li> </ul>
Divide ratio	$\frac{1}{n}$ n: TAI register setting value, 0000h to FFFFh (Note that the timer counter does not run if n = 0000h)
Count start conditions	<p>The TAI<sub>S</sub> bit in the TABSR register is 1 (start counter) and any of following triggers occurs:</p> <ul style="list-style-type: none"> <li>• An external trigger applied to the TAI<sub>IN</sub> pin</li> <li>• One of the following: the overflow and/or underflow signal of timer B2, the overflow and/or underflow signal of timer A<sub>j</sub> (j = i - 1, or j = 4 if i = 0), or the overflow and/or underflow signal of timer A<sub>k</sub> (k = i + 1, or k = 0 if i = 4)</li> <li>• The TAI<sub>OS</sub> bit in the ONSF register is 1 (start the timer)</li> </ul>
Count stop conditions	<ul style="list-style-type: none"> <li>• The timer counter reaches 0000h and the reload register value is reloaded</li> <li>• The TAI<sub>S</sub> bit in the TABSR register is 0 (stop counter)</li> </ul>
Interrupt request generating timing	When the timer counter reaches 0000h
TAI <sub>IN</sub> pin function	A programmable I/O port or a trigger input
TAI <sub>OUT</sub> pin function	A programmable I/O port or a pulse output
Read from timer	The TAI register indicates an undefined value
Write to timer	<ul style="list-style-type: none"> <li>• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAI register is written to both the reload register and the counter</li> <li>• While the timer counter is running, the value written to the TAI register is written to the reload register (it is transferred to the counter at the next reload timing)</li> </ul>
Other function	<ul style="list-style-type: none"> <li>• Pulse output function</li> </ul> <p>A low is output while the timer counter is stopped and a high is output while the timer counter is running</p>



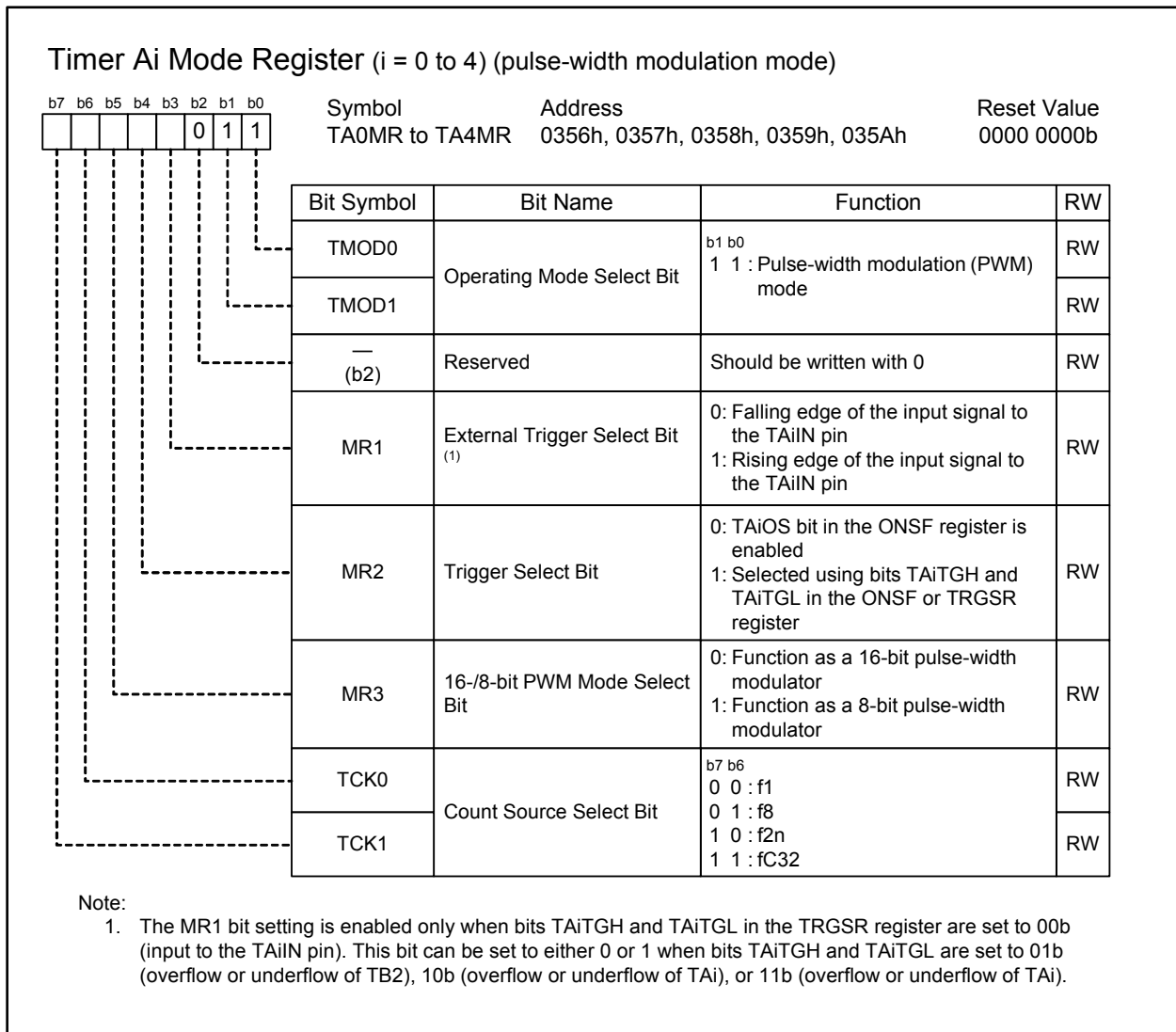
**Figure 16.15 Registers TA0MR to TA4MR in One-shot Timer Mode**

### 16.1.4 Pulse-width Modulation Mode

In pulse-width modulation mode, the timer outputs pulses of given width successively. Table 16.5 lists specifications of pulse-width modulation mode. The timer counter functions as either a 16-bit or 8-bit pulse-width modulator. Figure 16.16 shows registers TA0MR to TA4MR in this mode. Figures 16.17 and 16.18 show operation examples of 16-bit and 8-bit pulse-width modulators.

**Table 16.5 Pulse-width Modulation Mode Specifications (i = 0 to 4)**

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	<ul style="list-style-type: none"> <li>Decrement (the timer counter functions as an 8-bit or a 16-bit pulse-width modulator)</li> <li>The reload register value is reloaded on the rising edge of a PWM pulse to continue counting</li> <li>The timer is not affected by a trigger that occurs while the counter is running</li> </ul>
16-bit PWM	<ul style="list-style-type: none"> <li>High level width: <math>\frac{n}{fj}</math> <i>n</i>: TAI register setting value, 0000h to FFFEh <i>fj</i>: Count source frequency</li> <li>Period: fixed to <math>\frac{2^{16} - 1}{fj}</math></li> </ul>
8-bit PWM	<ul style="list-style-type: none"> <li>High level width: <math>\frac{n \times (m + 1)}{fj}</math></li> <li>Period: <math>\frac{(2^8 - 1) \times (m + 1)}{fj}</math> <i>n</i>: Upper byte of the TAI register setting value, 00h to FEh <i>m</i>: Lower byte of the TAI register setting value, 00h to FFh</li> </ul>
Count start conditions	<ul style="list-style-type: none"> <li>The TAI<sub>S</sub> bit in the TABSR register is 1 (start counter)</li> <li>The TAI<sub>S</sub> bit is 1 and an external trigger is applied to the TAI<sub>IN</sub> pin</li> <li>The TAI<sub>S</sub> bit is 1 and any of following triggers occurs: the overflow and/or underflow signal of timer B2, the overflow and/or underflow signal of timer Aj (j = i - 1, or j = 4 if i = 0), or the overflow and/or underflow signal of timer Ak (k = i + 1, or k = 0 if i = 4)</li> </ul>
Count stop condition	The TAI <sub>S</sub> bit in the TABSR register is 0 (stop counter)
Interrupt request generating timing	On the falling edge of the PWM pulse
TAI <sub>IN</sub> pin function	A programmable I/O port or trigger input
TAI <sub>OUT</sub> pin function	A pulse output
Read from timer	The TAI register indicates an undefined value
Write to timer	<ul style="list-style-type: none"> <li>While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAI register is written to both the reload register and the counter</li> <li>While the timer counter is running, the value written to the TAI register is written to the reload register (it is transferred to the counter at the next reload timing)</li> </ul>



**Figure 16.16 Registers TA0MR to TA4MR in Pulse-width Modulation Mode**



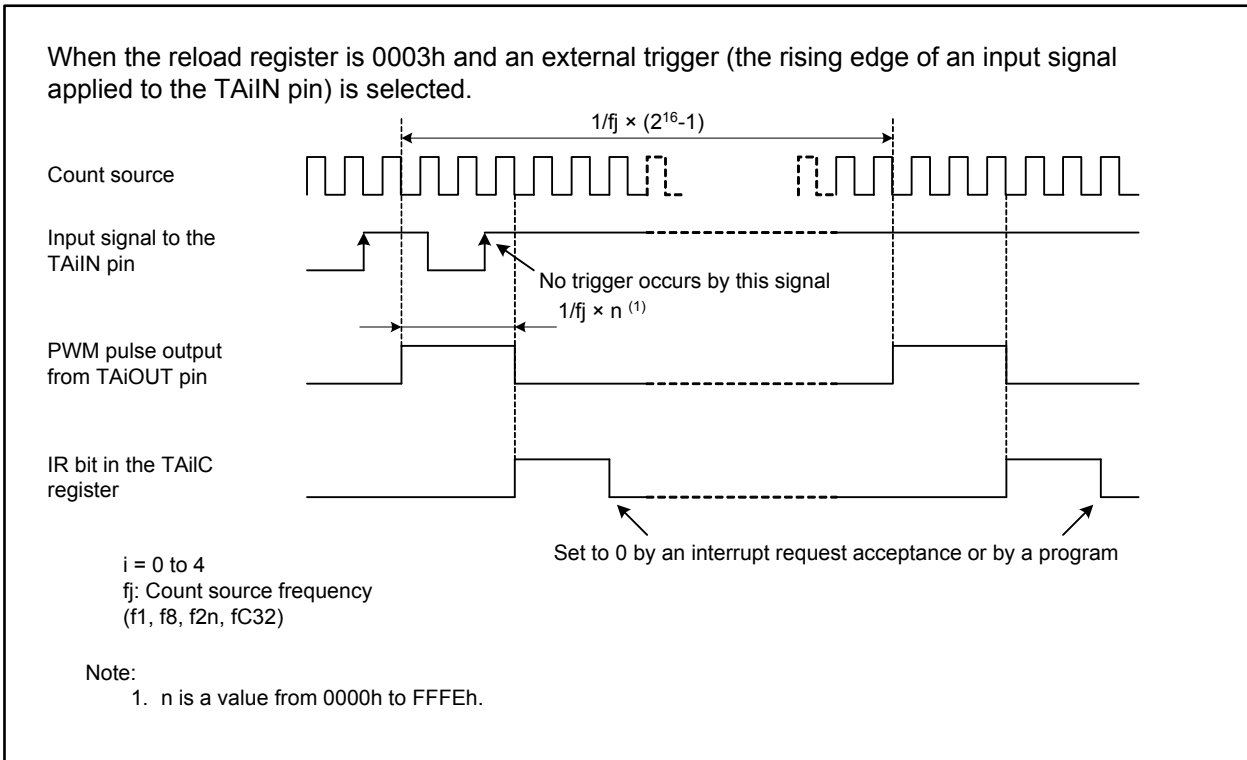


Figure 16.17 16-bit Pulse-width Modulator Operation

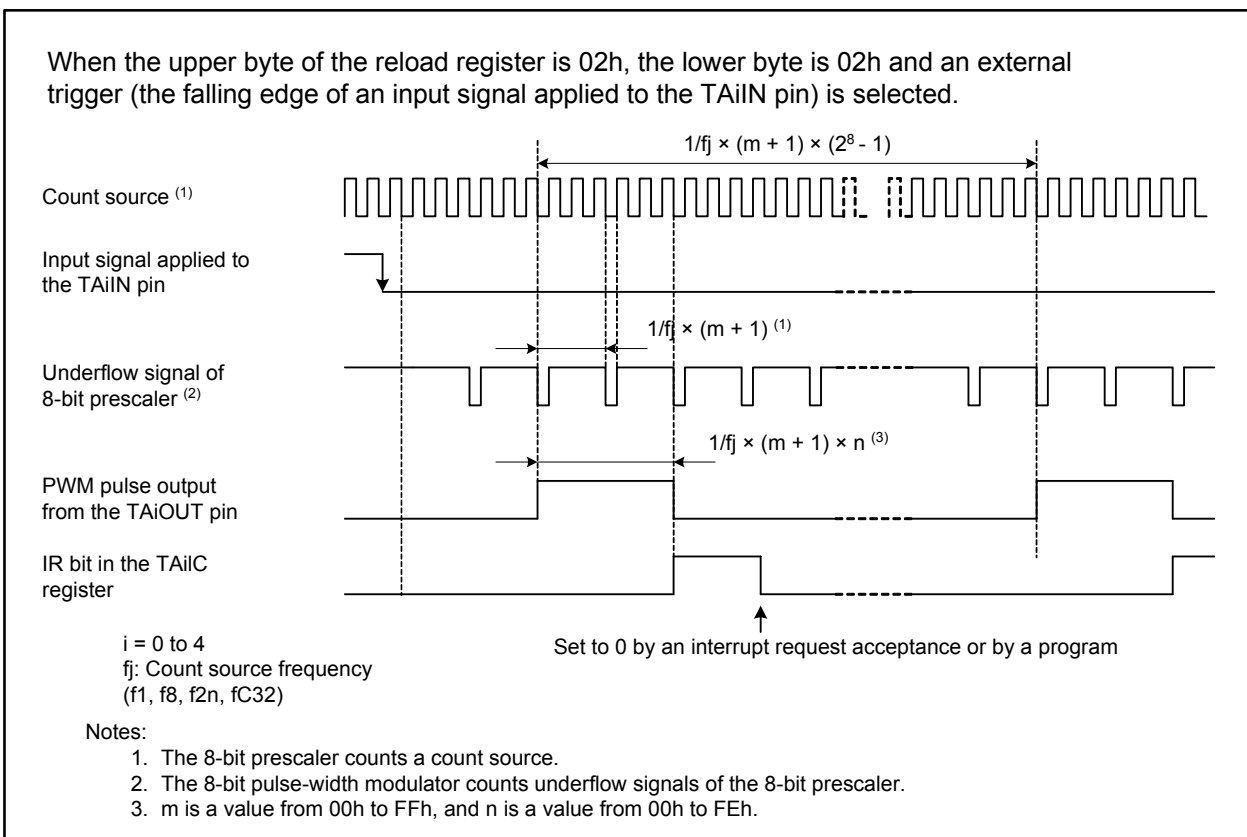


Figure 16.18 8-bit Pulse-width Modulator Operation

## 16.2 Timer B

Figure 16.19 shows a block diagram of timer B, and Figure 16.20 to Figure 16.23 show registers associated with timer B.

Timer B supports the three modes shown below. Select a mode by setting bits TMOD1 and TMOD0 in the TBiMR register ( $i = 0$  to 5).

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts an external pulse or an overflow and underflow of other timers.
- Pulse period/pulse-width measure mode: The timer measures the pulse period or pulse width of an external signal.

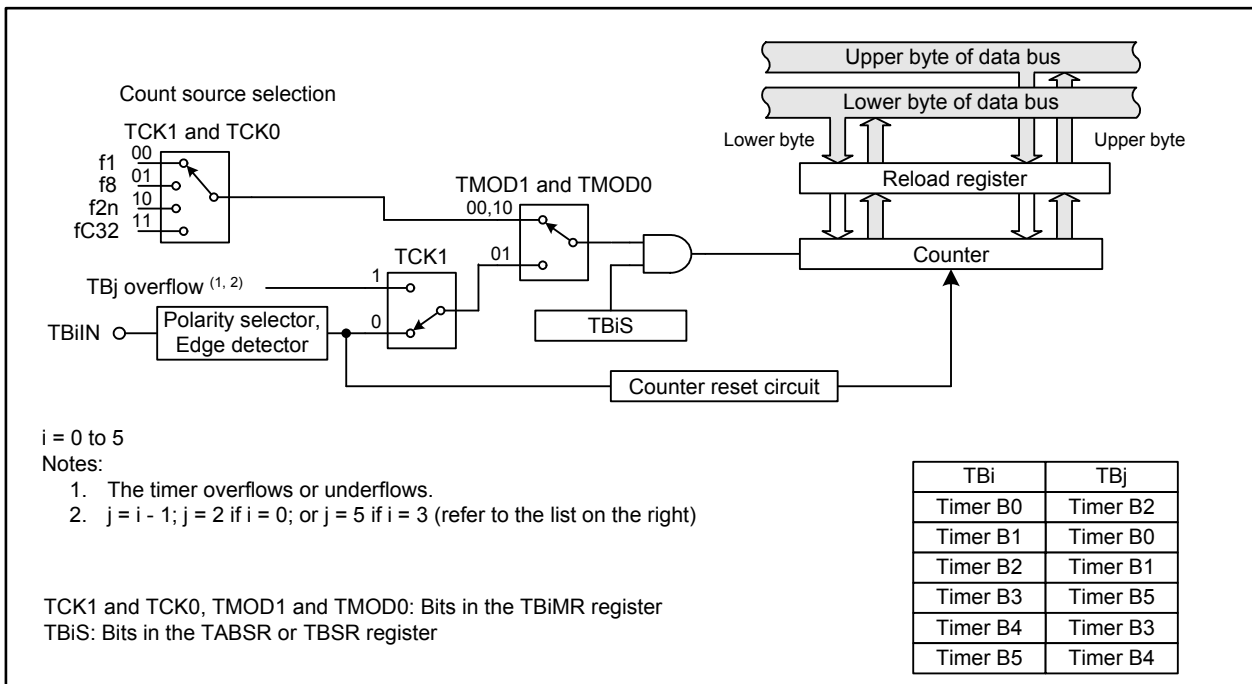


Figure 16.19 Timer B Block Diagram

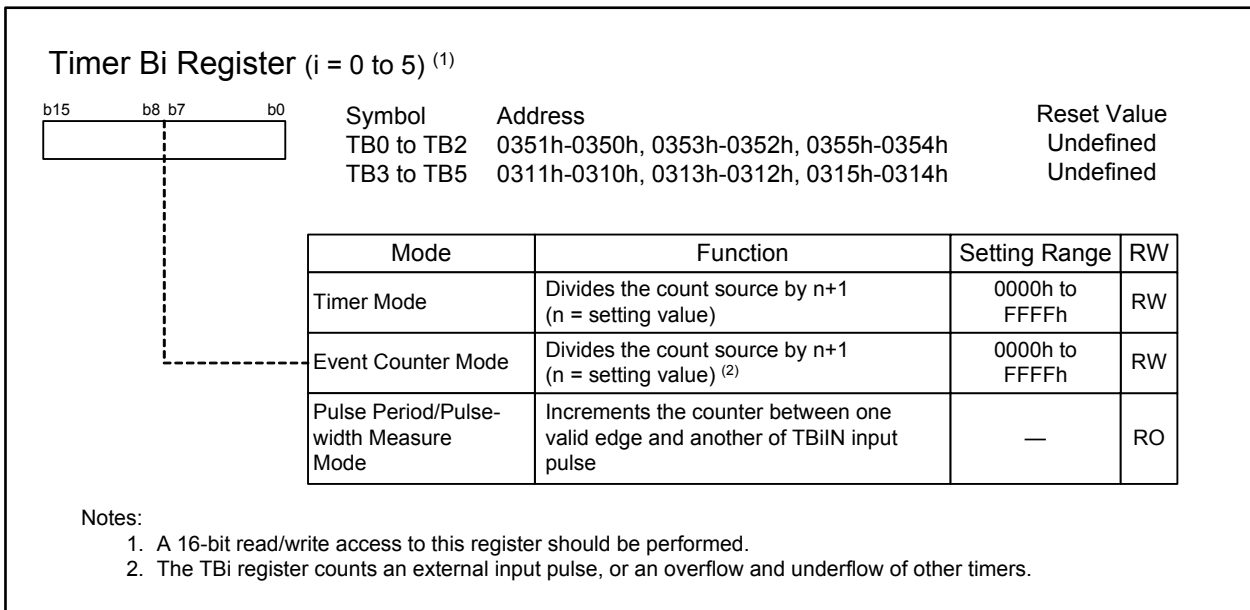


Figure 16.20 Registers TB0 to TB5

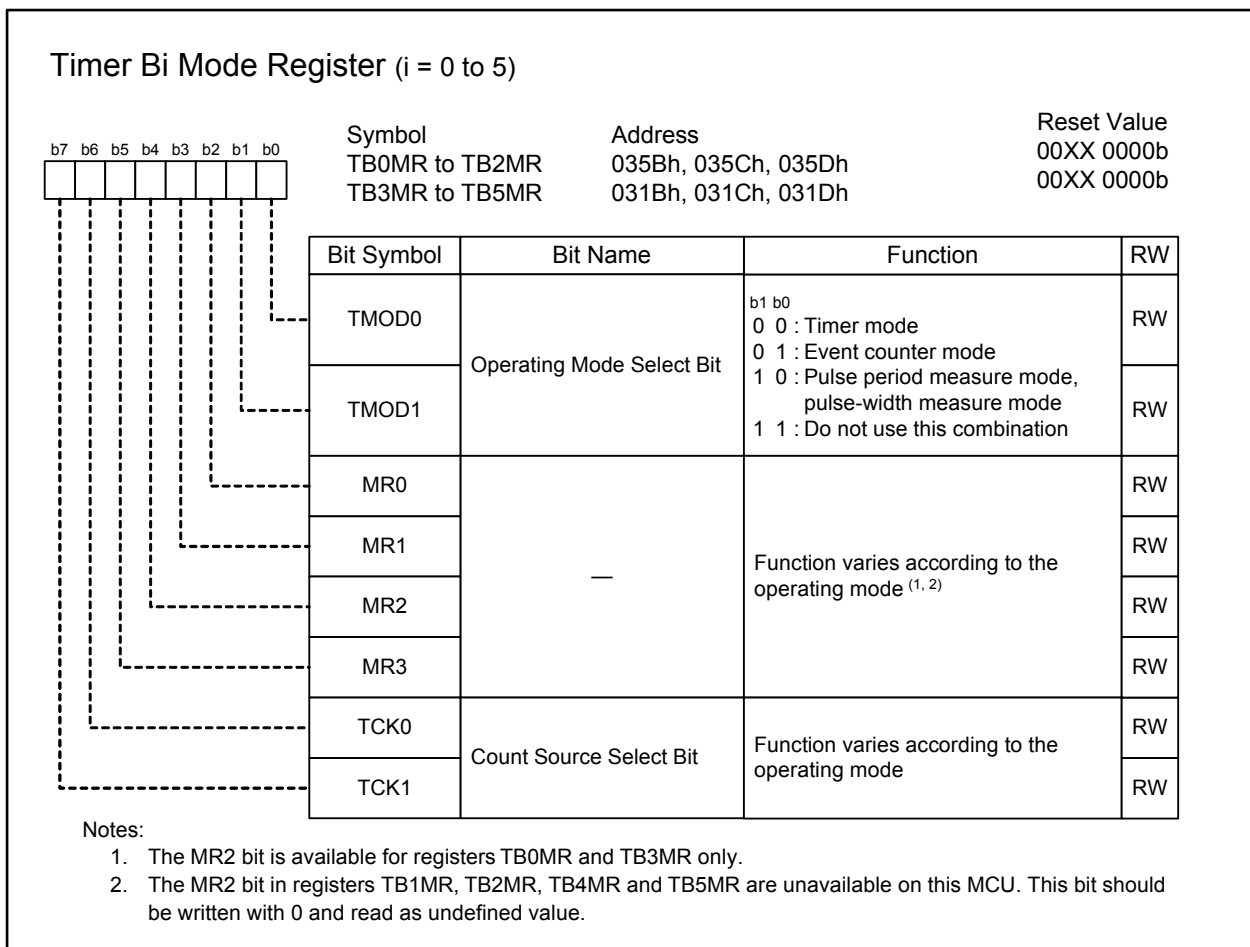


Figure 16.21 Registers TB0MR to TB5MR

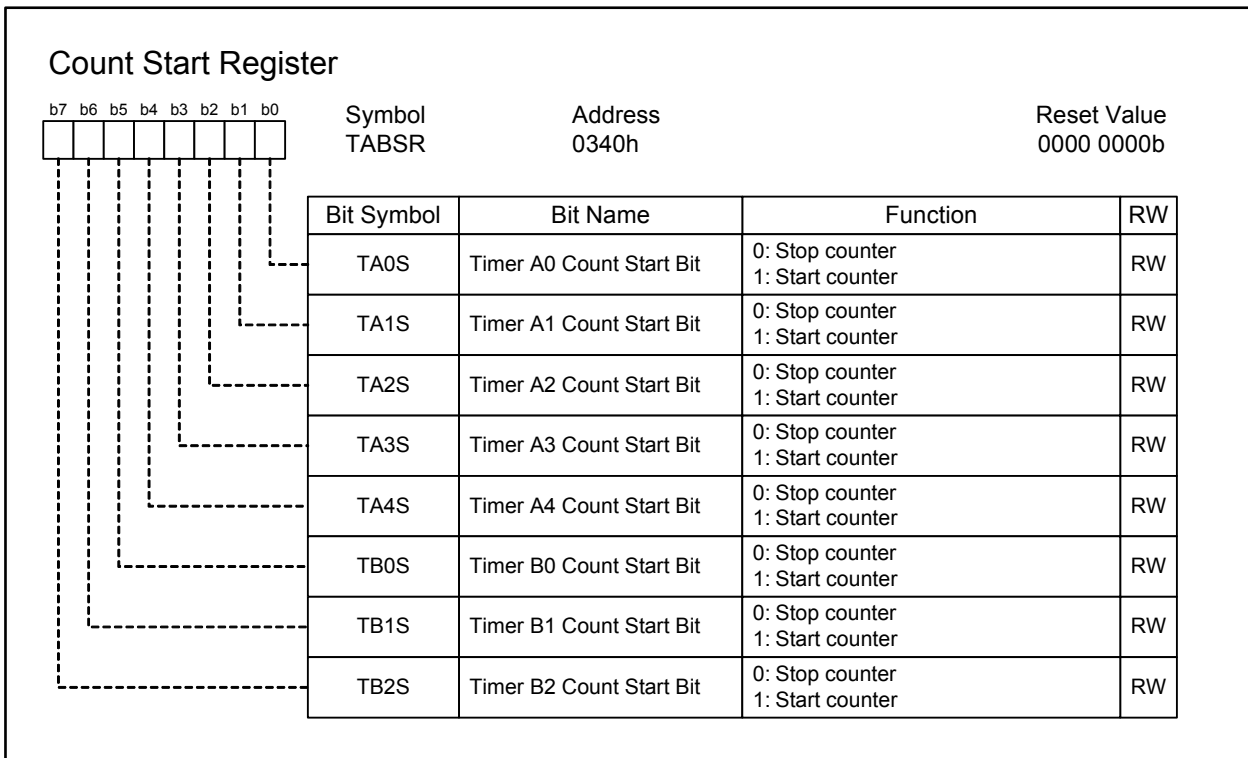


Figure 16.22 TABSR Register

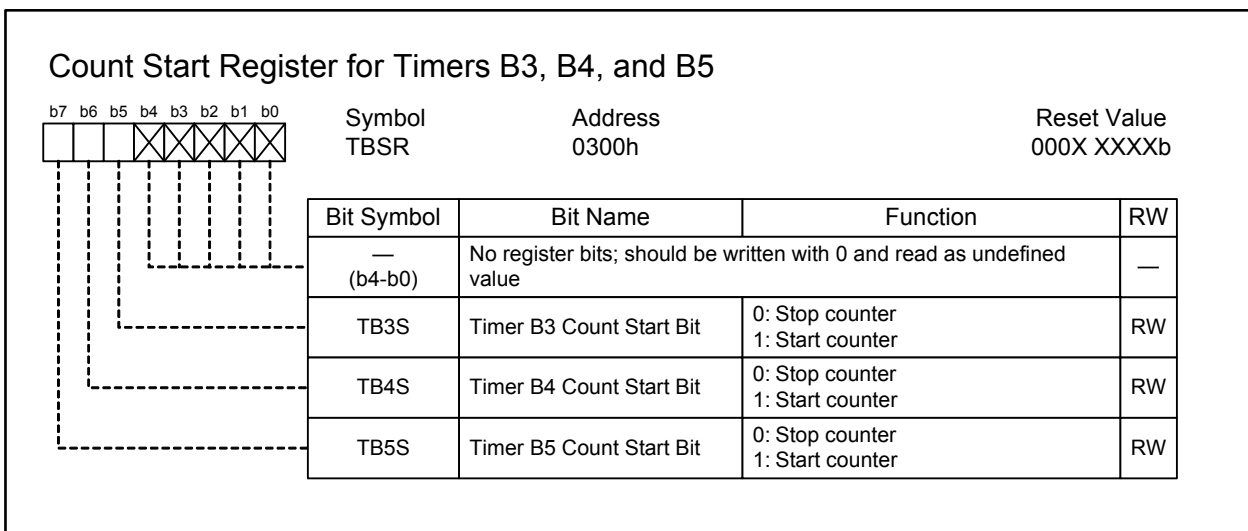


Figure 16.23 TBSR Register

### 16.2.1 Timer Mode

In timer mode, the timer counts an internally generated count source. Table 16.6 lists specifications of timer mode. Figure 16.24 shows registers TB0MR to TB5MR in this mode.

**Table 16.6 Timer Mode Specifications (i = 0 to 5)**

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	<ul style="list-style-type: none"> <li>• Decrement</li> <li>• When the timer counter underflows, the reload register value is reloaded into the counter to continue counting</li> </ul>
Divide ratio	$\frac{1}{n+1}$ n: TBi register setting value, 0000h to FFFFh
Count start condition	The TBiS bit in the TABSR or TBSR register is 1 (start counter)
Count stop condition	The TBiS bit in the TABSR or TBSR register is 0 (stop counter)
Interrupt request generating timing	When the timer counter underflows
TBiIN pin function	Functions as a programmable I/O port
Read from timer	The TBi register indicates a counter value
Write to timer	<ul style="list-style-type: none"> <li>• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TBi register is written to both the reload register and the counter</li> <li>• While the timer counter is running, the value written to the TBi register is written to the reload register (it is transferred to the counter at the next reload timing)</li> </ul>

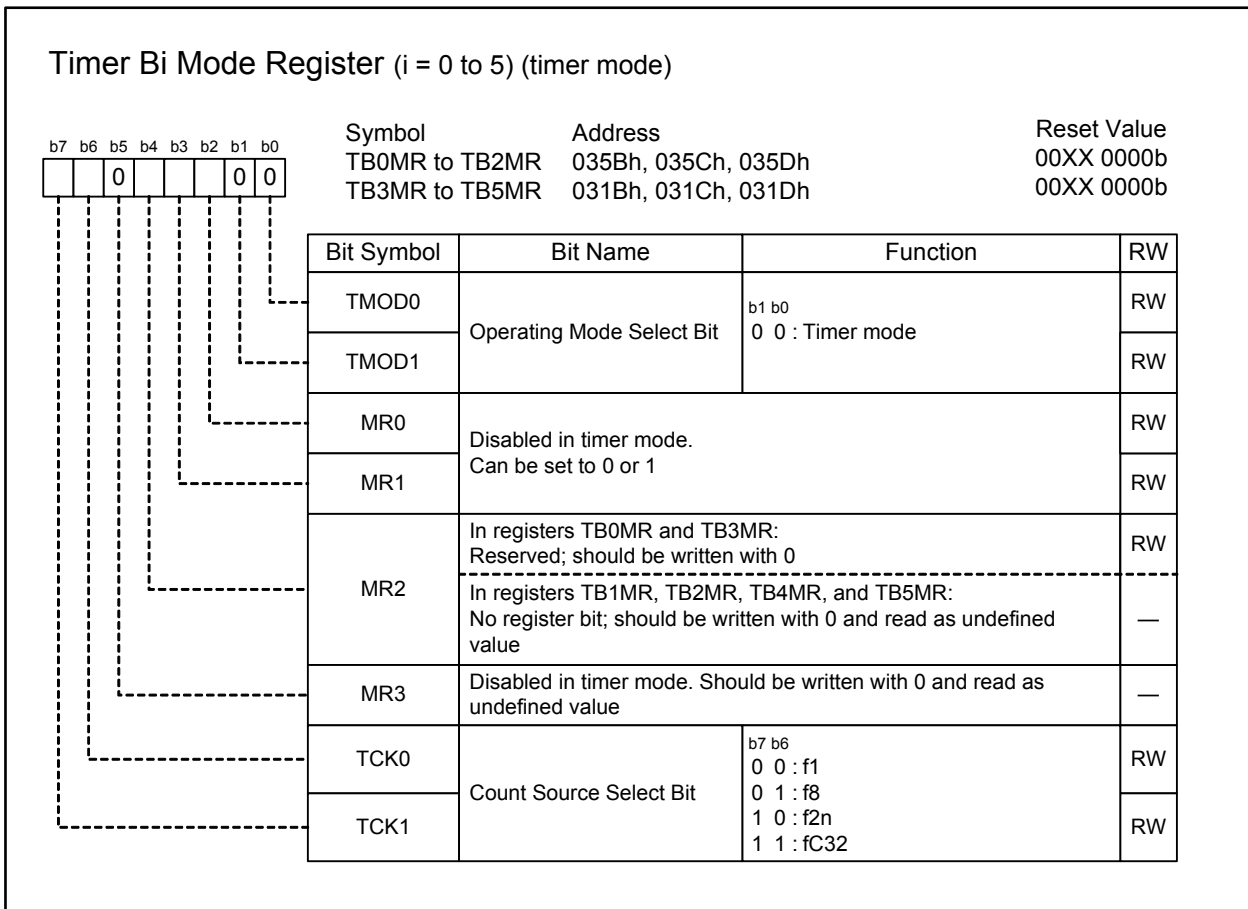


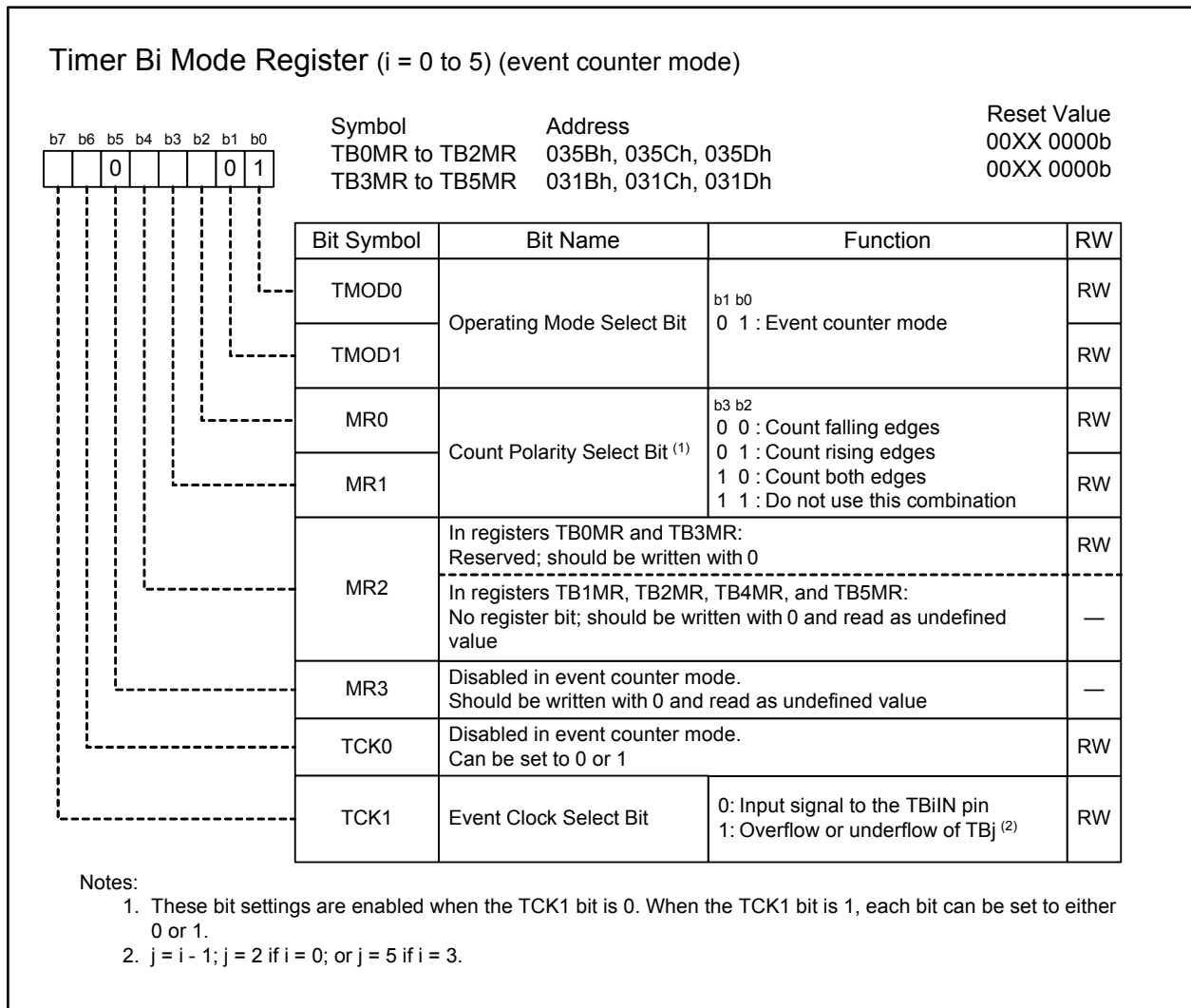
Figure 16.24 Registers TB0MR to TB5MR in Timer Mode

### 16.2.2 Event Counter Mode

In event counter mode, the timer counts an external signal or the overflow or underflow of other timers. Table 16.7 lists specifications of event counter mode. Figure 16.25 shows the TBiMR register in this mode (i = 0 to 5).

**Table 16.7 Event Counter Mode Specifications (i = 0 to 5)**

Item	Specification
Count sources	<ul style="list-style-type: none"> <li>External signal applied to the TBiIN pin (valid edge is selectable among the falling edge, the rising edge, or both)</li> <li>The overflow or underflow signal of TBj (j = i - 1; j = 2 if i = 0; or j = 5 if i = 3)</li> </ul>
Count operations	<ul style="list-style-type: none"> <li>Decrement</li> <li>When the timer counter underflows, the reload register value is reloaded into the counter to continue counting</li> </ul>
Divide ratio	$\frac{1}{n+1}$ n: TBi register setting value, 0000h to FFFFh
Count start condition	The TBiS bit in the TABSR or TBSR register is 1 (start counter)
Count stop condition	The TBiS bit in the TABSR or TBSR register is 0 (stop counter)
Interrupt request generation timing	When the timer counter underflows
TBiIN pin function	Functions as a programmable I/O port or count source input
Read from timer	The TBi register indicates a counter value
Write to timer	<ul style="list-style-type: none"> <li>While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TBi register is written to both the reload register and the counter</li> <li>While the timer counter is running, the value written to the TBi register is written to the reload register (it is transferred to the counter at the next reload timing)</li> </ul>



**Figure 16.25 Registers TB0MR to TB5MR in Event Counter Mode**



### 16.2.3 Pulse Period/Pulse-width Measure Mode

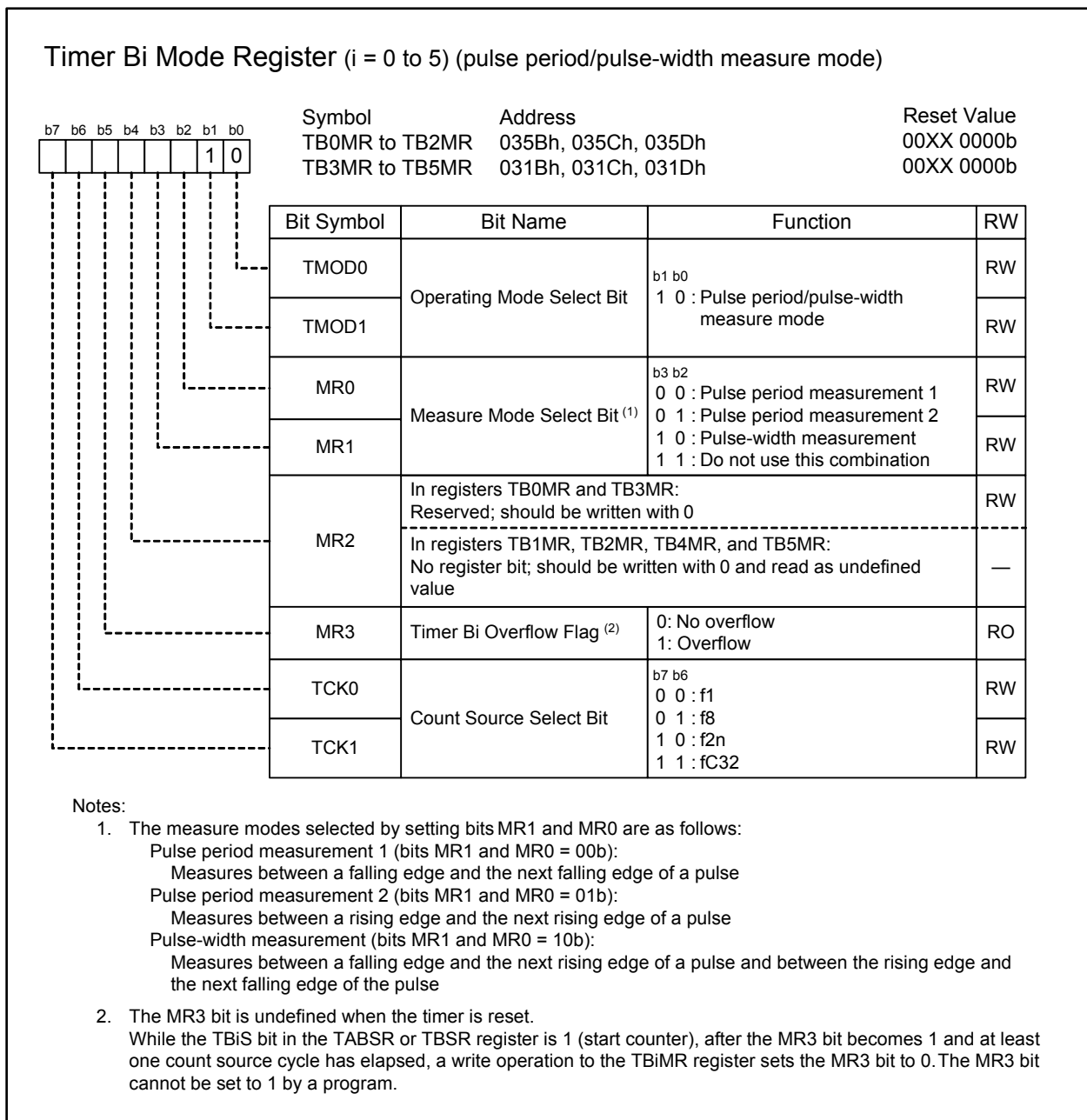
In pulse period/pulse-width measure mode, the timer measures the pulse period or pulse width of an external signal. Table 16.8 lists specifications of the pulse period/pulse-width measure mode. Figure 16.26 shows registers TB0MR to TB5MR in this mode. Figures 16.27 and 16.28 show an operation example of pulse period measurement and pulse-width measurement, respectively.

**Table 16.8 Pulse Period/Pulse-width Measure Mode Specifications (i = 0 to 5)**

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	<ul style="list-style-type: none"> <li>• Increment</li> <li>• The counter value is transferred to the reload register on the valid edge of a pulse to be measured, then it is set to 0000h to resume counting</li> </ul>
Count start condition	The TBiS bit in the TABSR or TBSR register is 1 (start counter)
Count stop condition	The TBiS bit in the TABSR or TBSR register is 0 (stop counter)
Interrupt request generating timing	<ul style="list-style-type: none"> <li>• On the valid edge of a pulse to be measured <sup>(1)</sup></li> <li>• When the timer counter overflows (when the MR3 bit in the TBIMR register becomes 1 (overflow)) <sup>(2)</sup></li> </ul>
TBiIN pin function	A pulse input to be measured
Read from timer	The TBi register indicates a reload register value (measurement results) <sup>(3)</sup>
Write to timer	The value written to the TBi register is written to neither the reload register nor the counter

Notes:

1. No interrupt request is generated when the pulse to be measured is applied on the initial valid edge after the timer counter starts.
2. While the TBiS bit is 1 (start counter), after the MR3 bit becomes 1 (overflow) and at least one count source cycle has elapsed, a write operation to the TBIMR register sets the MR3 bit to 0 (no overflow).
3. The TBi register indicates an undefined value until the pulse to be measured is applied on the second valid edge after the timer counter starts.



**Figure 16.26 Registers TB0MR to TB5MR in Pulse Period/Pulse-width Measure Mode**

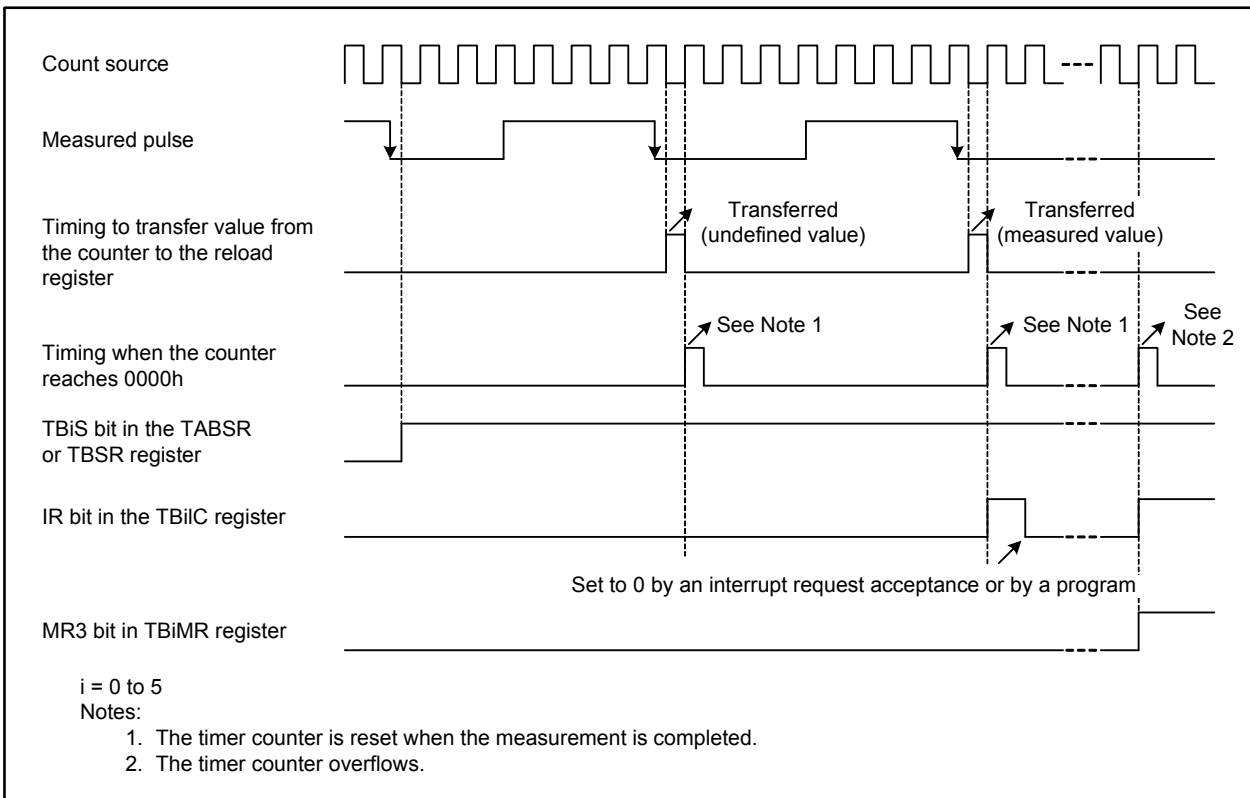


Figure 16.27 Operation Example in Pulse Period Measurement

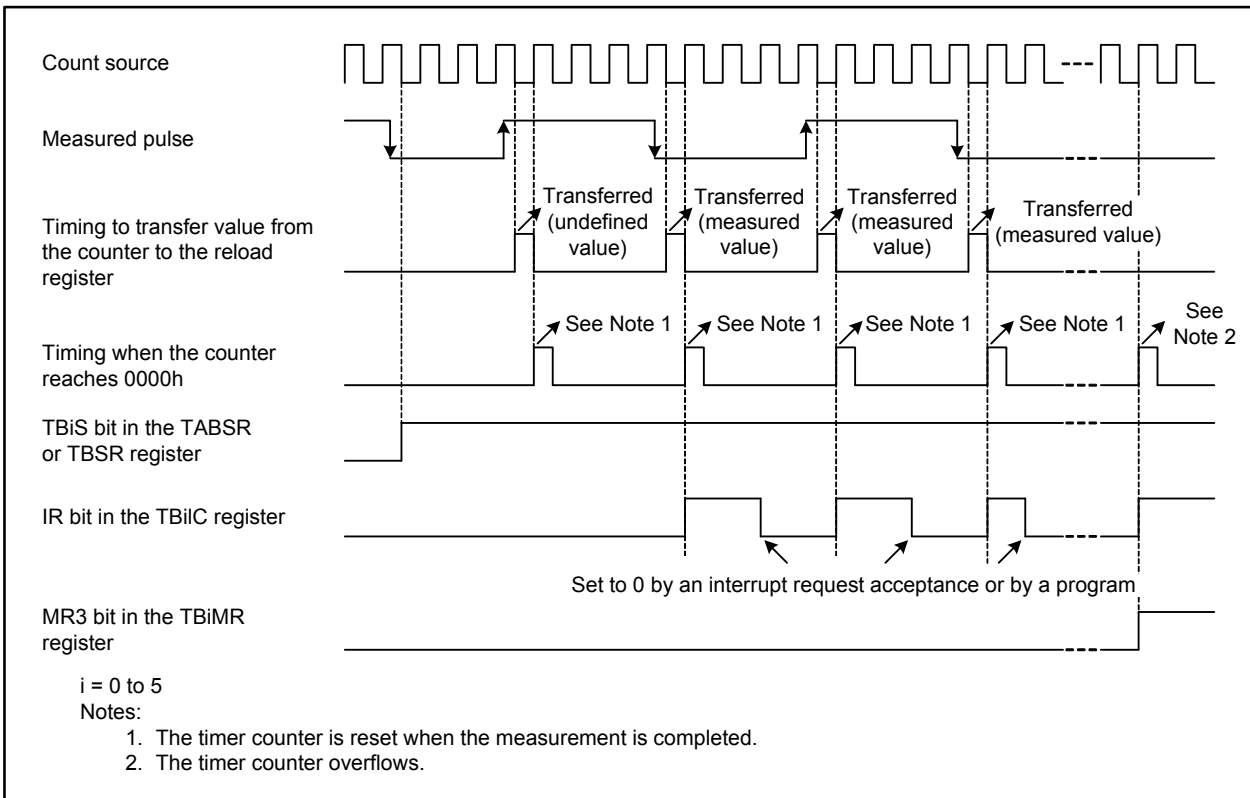


Figure 16.28 Operation Example in Pulse-width Measurement

## 16.3 Notes on Timers

### 16.3.1 Timer A and Timer B

All timers are stopped after a reset. To restart timers, configure parameters such as operating mode, count source, and counter value, then set the TAI<sub>S</sub> bit or TB<sub>J</sub>S bit in the TABSR or TBSR register to 1 (count starts) (i = 0 to 4; j = 0 to 5).

The following registers and bits should be set while the TAI<sub>S</sub> bit or TB<sub>J</sub>S bit is 0 (count stops):

- Registers TAI<sub>MR</sub> and TB<sub>J</sub>MR
- UDF register
- Bits TAZIE, TA0TGL, and TA0TGH in the ONSF register
- TRGSR register

### 16.3.2 Timer A

#### 16.3.2.1 Timer Mode

- While the timer counter is running, the TAI register indicates a counter value at any given time. However, FFFFh is read while reloading is in progress. A set value is read if the TAI register is set while the timer counter is stopped.

#### 16.3.2.2 Event Counter Mode

- While the timer counter is running, the TAI register indicates a counter value at any given time. However, FFFFh is read if the timer counter underflows or 0000h if overflows while reloading is in progress. A set value is read if the TAI register is set while the timer counter is stopped.

#### 16.3.2.3 One-shot Timer Mode

- If the TAI<sub>S</sub> bit in the TABSR register is set to 0 (count stops) while the timer counter is running, the following operations are performed:
  - The timer counter stops and the setting value of the TAI register is reloaded.
  - A low signal is output at the TAI<sub>OUT</sub> pin.
  - The IR bit in the TAI<sub>IC</sub> register becomes 1 (interrupts requested) after one CPU clock cycle.
- The one-shot timer is operated by an internal count source. When the trigger is an input to the TAI<sub>IN</sub> pin, the signal is output with a maximum one count source clock delay after a trigger input to the TAI<sub>IN</sub> pin.
- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt, set the IR bit to 0 after one of the settings below is done:
  - Select one-shot timer mode after a reset.
  - Switch operating modes from timer mode to one-shot timer mode.
  - Switch operating modes from event counter mode to one-shot timer mode.
- If a retrigger occurs while counting, the timer counter decrements by one, reloads the setting value of the TAI register, and then continues counting. To generate a retrigger while counting, wait at least one count source cycle after the last trigger is generated.
- When an external trigger input is selected to start counting in timer A one-shot mode, do not provide an external retrigger for 300 ns before the timer counter reaches 0000h. Otherwise, it may stop counting.

#### 16.3.2.4 Pulse-width Modulation Mode

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt, set the IR bit to 0 after one of the settings below is done (i = 0 to 4):
  - Select pulse-width modulation mode after a reset.
  - Switch operating modes from timer mode to pulse-width modulation mode.
  - Switch operating modes from event counter mode to pulse-width modulation mode.
  
- If the TAI<sub>S</sub> bit in the TABSR register is set to 0 (count stops) while PWM pulse is output, the following operations are performed:
  - The timer counter stops.
  - The output level at the TAI<sub>OUT</sub> pin changes from high to low. The IR bit becomes 1.
  - When a low signal is output at the TAI<sub>OUT</sub> pin, it does not change. The IR bit does not change, either.

### 16.3.3 Timer B

#### 16.3.3.1 Restriction for the Use of 64-pin Package's Products

- TB4IN pin is not available in the 64-pin package. Timer B4 is available in timer mode only.

#### 16.3.3.2 Timer Mode and Event Counter Mode

- While the timer counter is running, the TBJ register indicates a counter value at any given time ( $j = 0$  to  $5$ ). However, FFFFh is read while reloading is in progress. When a value is set to the TBJ register while the timer counter is stopped, if the TBJ register is read before the count starts, the set value is read.

#### 16.3.3.3 Pulse Period/Pulse-width Measure Mode

- To set the MR3 bit in the TBJMR register to 0 (no overflow), write the TBJMR register after the MR3 bit becomes 1 (overflow) and at least one count source cycle has elapsed while the TBJS bit in the TABSR or TBSR register is 1 (start counter).
- Use the IR bit in the TBJIC register to detect overflow. The MR3 bit is used only to determine an interrupt request source within the interrupt handler.
- The counter value is undefined when the timer counter starts. Therefore, the timer counter may overflow before a measured pulse is applied on the initial valid edge and cause a timer B $j$  interrupt request to be generated.
- When the measured pulse is applied on the initial valid edge after the timer counter starts, an undefined value is transferred to the reload register. At this time, a timer B $j$  interrupt request is not generated.
- The IR bit may become 1 (interrupt requested) by changing bits MR1 and MR0 in the TBJMR register after the timer counter starts. However, if the same value is rewritten to bits MR1 and MR0, the IR bit does not change.
- Pulse width is continuously measured in pulse-width measure mode. Whether the measurement result is high-level width or not is determined by a program.
- When an overflow occurs at the same time a pulse is applied on the valid edge, this pulse is not recognized since an interrupt request is generated only once. Do not let an overflow occur in pulse period measure mode.
- In pulse-width measure mode, determine whether an interrupt source is a pulse applied on the valid edge or an overflow by reading the port level in the timer B $j$  interrupt handler.

## 17. Three-phase Motor Control Timers

A three-phase motor driving waveform can be output using timers A1, A2, A4, and B2. The three-phase motor control timers are enabled by setting the INV02 bit in the INVC0 register to 1. Timer B2 is used for carrier wave control, and timers A1, A2, and A4 for three-phase PWM output (U,  $\bar{U}$ , V,  $\bar{V}$ , W, and  $\bar{W}$ ) control. Table 17.1 lists the specifications of the three-phase motor control timers and Figure 17.1 shows its block diagram. Figures 17.2 to 17.6 show registers associated with this function.

**Table 17.1 Specifications for Three-phase Motor Control Timers**

Item	Specification
Three-phase PWM waveform output pins	Six pins: U, $\bar{U}$ , V, $\bar{V}$ , W, and $\bar{W}$
Forced cutoff (1)	A low input to the $\overline{\text{NMI}}$ pin
Timers	Timers A4, A1, and A2 are used in one-shot timer mode: Timer A4 is used for U- and $\bar{U}$ -phase waveform control Timer A1 is used for V- and $\bar{V}$ -phase waveform control Timer A2 is used for W- and $\bar{W}$ -phase waveform control Timer B2 is used in timer mode Carrier wave cycle control Dead time timer (three 8-bit timers share a reload register): Dead time control
Output waveforms	Triangular wave modulation and sawtooth wave modulation • Output of a high or a low waveform for one cycle • Separately settable levels of high side and low side
Carrier wave periods	Triangular wave modulation: count source $\times (m + 1) \times 2$ Sawtooth wave modulation: count source $\times (m + 1)$ m: TB2 register setting value from 0000h to FFFFh Count source: f1, f8, f2n, or fC32
Three-phase PWM output width	Triangular wave modulation: count source $\times n \times 2$ Sawtooth wave modulation: count source $\times n$ n: Setting value of registers TA4, TA1, and TA2 (registers TA4, TA41, TA1, TA11, TA2, and TA21 when the INV11 bit in the INVC1 register is 1) from 0001h to FFFFh Count source: f1, f8, f2n, or fC32
Dead time (width)	Count source $\times p$ or no dead time p: DTT register setting value from 01h to FFh Count source: f1 or f1 divided by 2
Active level	Selectable either active high or active low
Simultaneous conduction prevention	Function to detect simultaneous turn-on signal outputs, function to disable signal output when simultaneous turn-on signal outputs are detected
Interrupt frequency	Selectable from one through 15 time-carrier wave cycle-to-cycle basis for the timer B2 interrupt

Note:

1. Forced cutoff by a signal input to the  $\overline{\text{NMI}}$  pin can be performed when the PM24 bit in the PM2 register is 1 (NMI enabled), the INV02 bit in the INVC0 register is 1 (three-phase motor control timers used), and the INV03 bit is 1 (three-phase motor control timer output enabled).

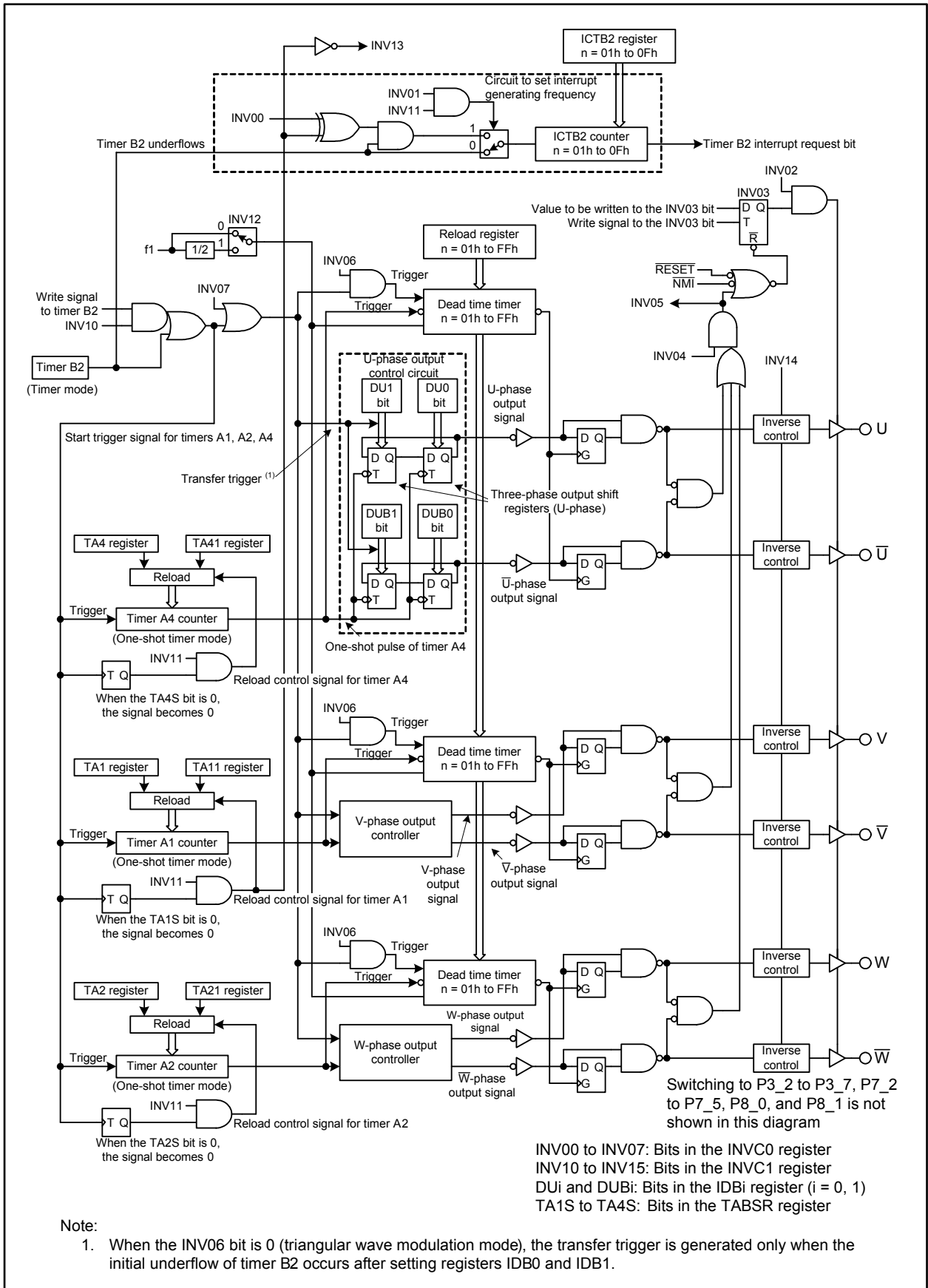


Figure 17.1 Block Diagram for Three-phase Motor Control Timers



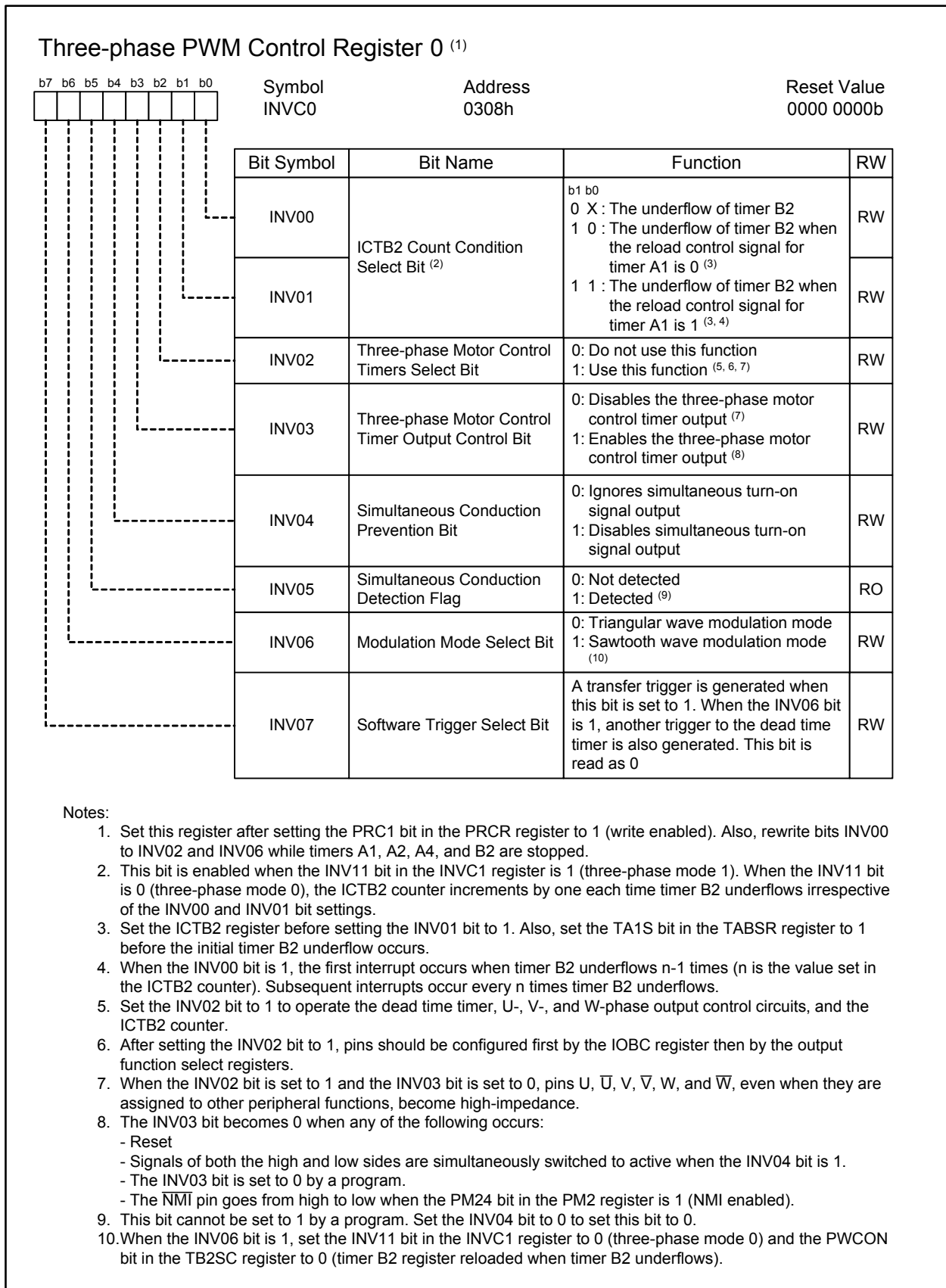
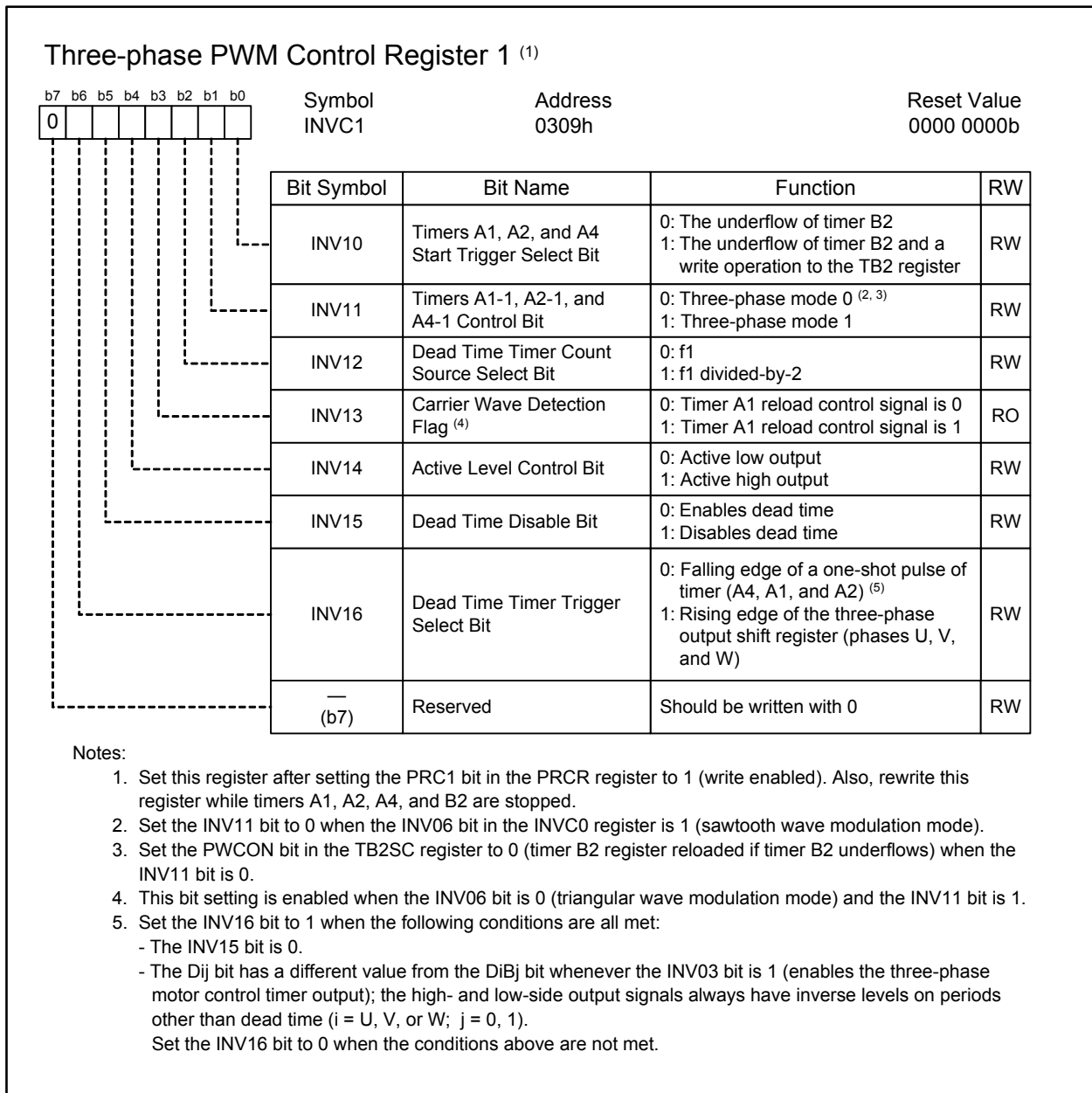
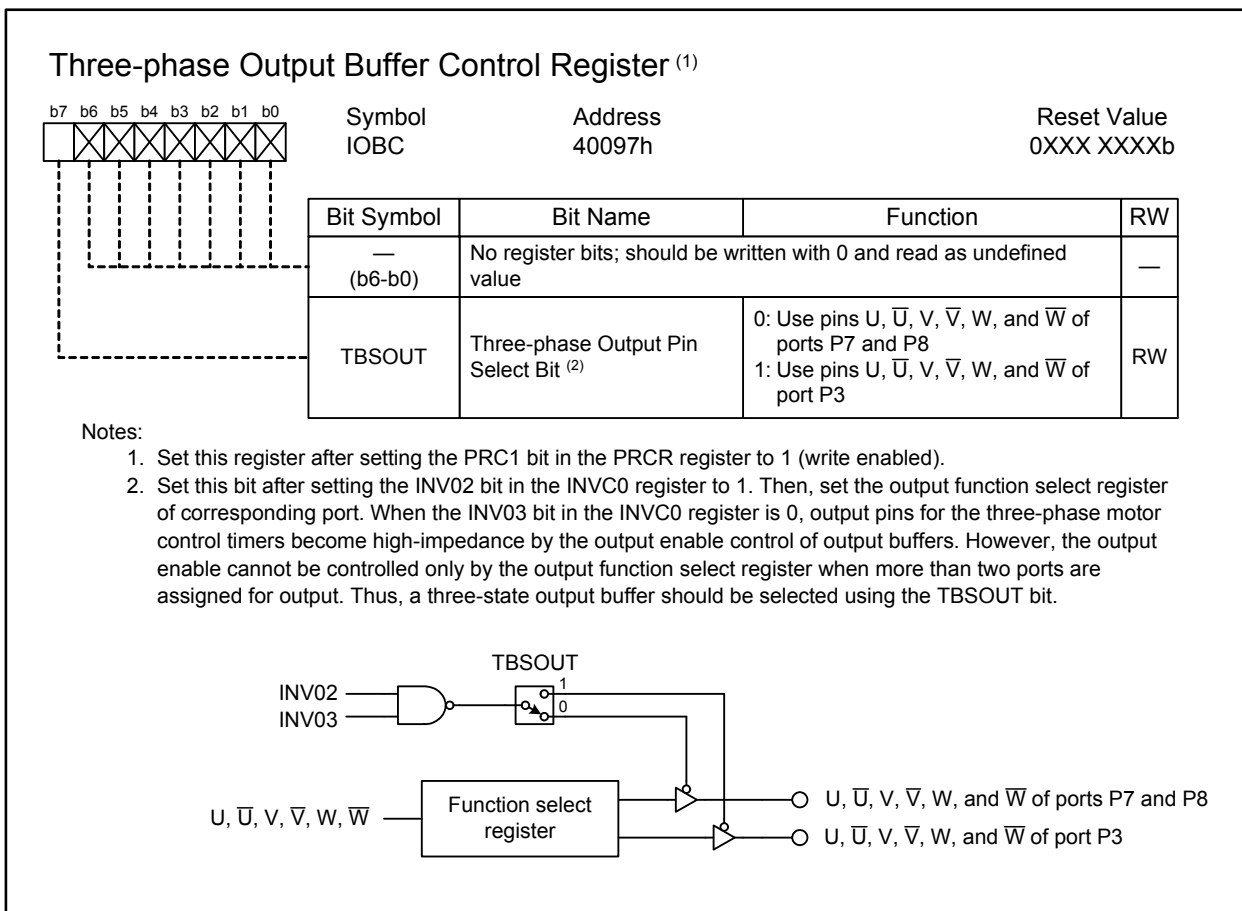


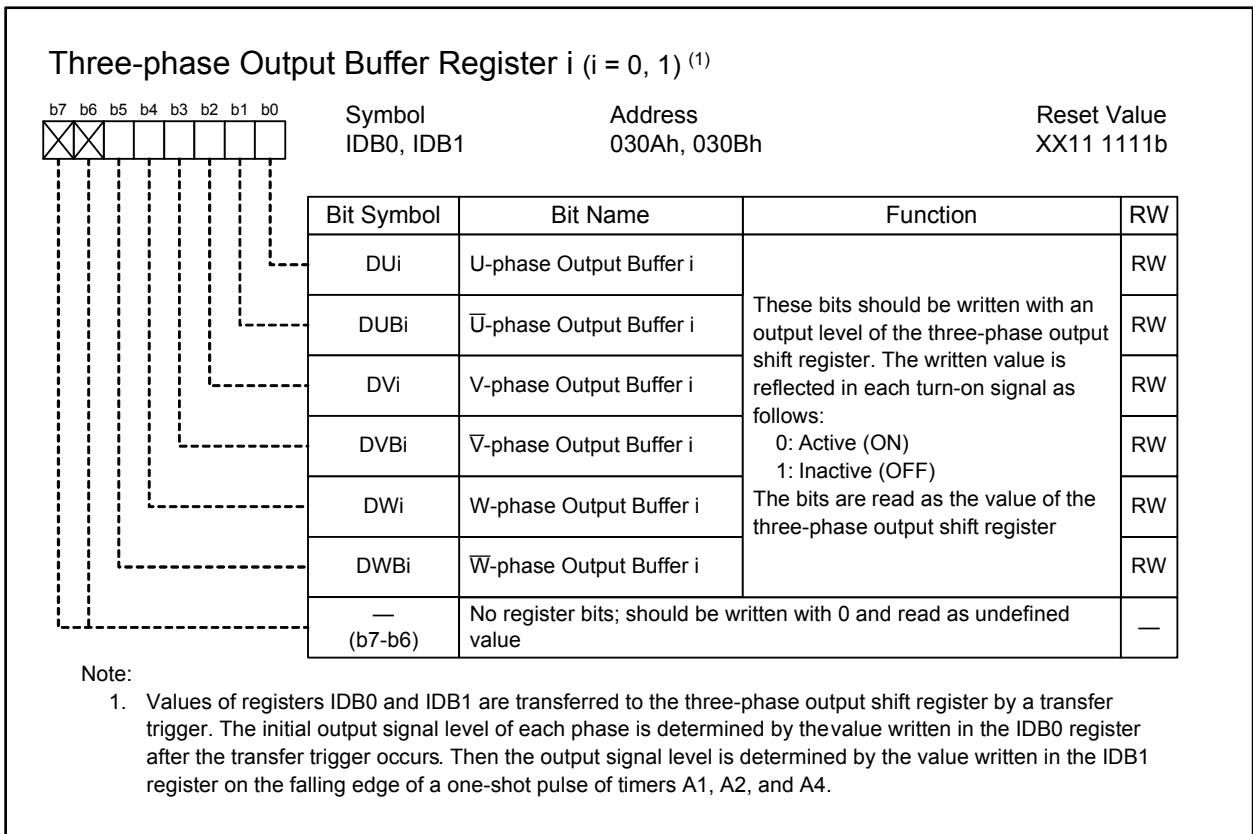
Figure 17.2 INVC0 Register



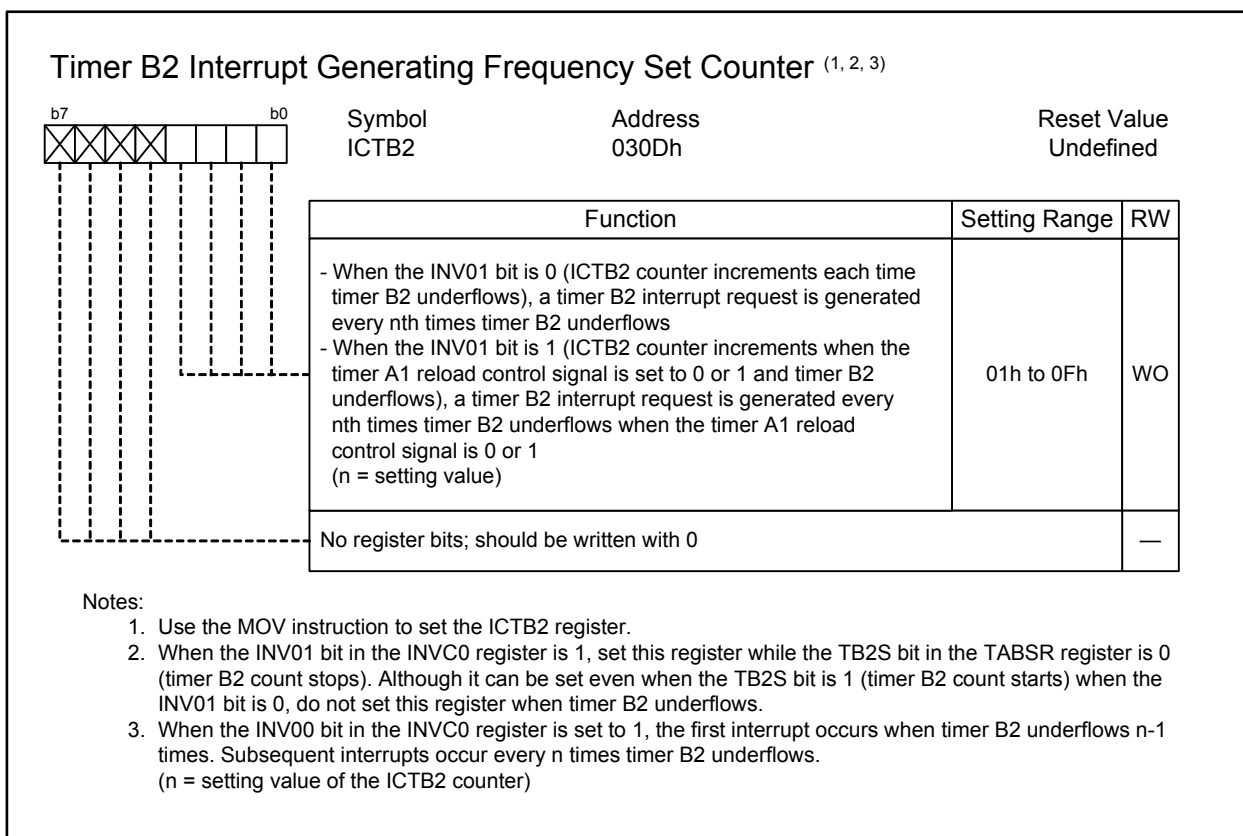
**Figure 17.3 INVC1 Register**



**Figure 17.4 IOBC Register**



**Figure 17.5 Registers IDB0 and IDB1**



**Figure 17.6 ICTB2 Register**

## 17.1 Modulation Modes of Three-phase Motor Control Timers

The three-phase motor control timers support two modulation modes: triangular wave modulation mode and sawtooth wave modulation mode. The triangular wave modulation mode has two modes: three-phase mode 0 and three-phase mode 1. Table 17.2 lists bit settings and characteristics of each mode.

**Table 17.2 Modulation Modes**

Item	Triangular Wave Modulation Mode		Sawtooth Wave Modulation Mode
	Three-phase mode 0	Three-phase mode 1	(Three-phase mode 0)
Bit settings	INV06 is 0, INV11 is 0, PWCON is 0	INV06 is 0, INV11 is 1	INV06 is 1, INV11 is 0, PWCON is 0
Waveform	Triangular wave		Sawtooth wave
Registers TA11, TA21, and TA41	Not used	Used	Not used
Timing to transfer data from registers IDB0 and IDB1 to the three-phase output shift register	Only once when a transfer trigger <sup>(1)</sup> occurs after setting registers IDB0 and IDB1		Whenever a transfer trigger <sup>(1)</sup> occurs
Timing to trigger the dead time timer when the INV16 bit is 0	On the falling edge of a one-shot pulse of timers A1, A2, and A4		When a transfer trigger occurs, or on the falling edge of a one-shot pulse of timers A1, A2, and A4
Bits INV00 and INV01 in the INVC0 register	Disabled. The ICTB2 counter increments each time timer B2 underflows, irrespective of the INV00 and INV01 bit settings	Enabled	Disabled. The ICTB2 counter increments each time timer B2 underflows, irrespective of the INV00 and INV01 bit settings
INV13 bit	Disabled	Enabled	Disabled

Note:

1. The transfer trigger is a timer B2 underflow, a write operation to the INV07 bit, or a write operation to the TB2 register when the INV10 bit is 1.

## 17.2 Timer B2

Timer B2, which operates in timer mode, is used for carrier wave control in the three-phase motor control timers.

Figures 17.7 and 17.8 show registers TB2 and TB2MR in this function, respectively. Figure 17.9 shows the TB2SC register which switches timing to change the carrier wave frequency in three-phase mode 1.

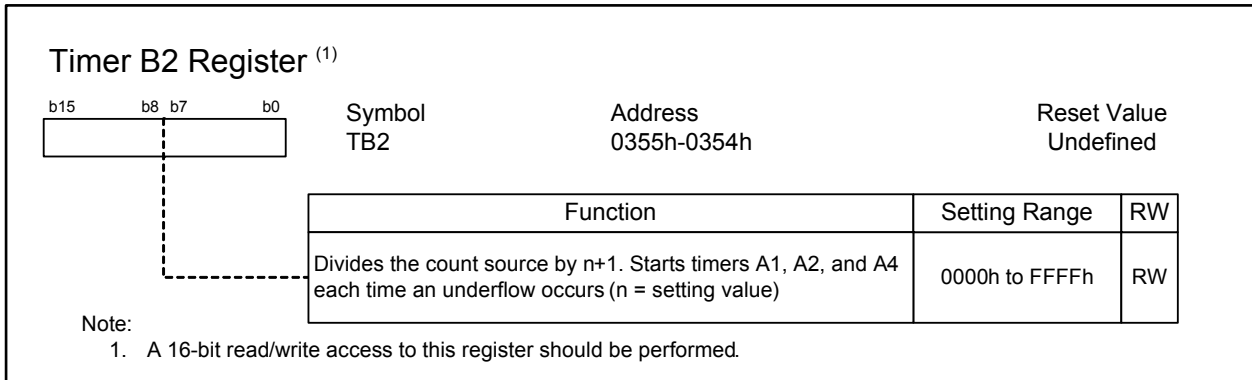


Figure 17.7 TB2 Register When Using Three-phase Motor Control Timers

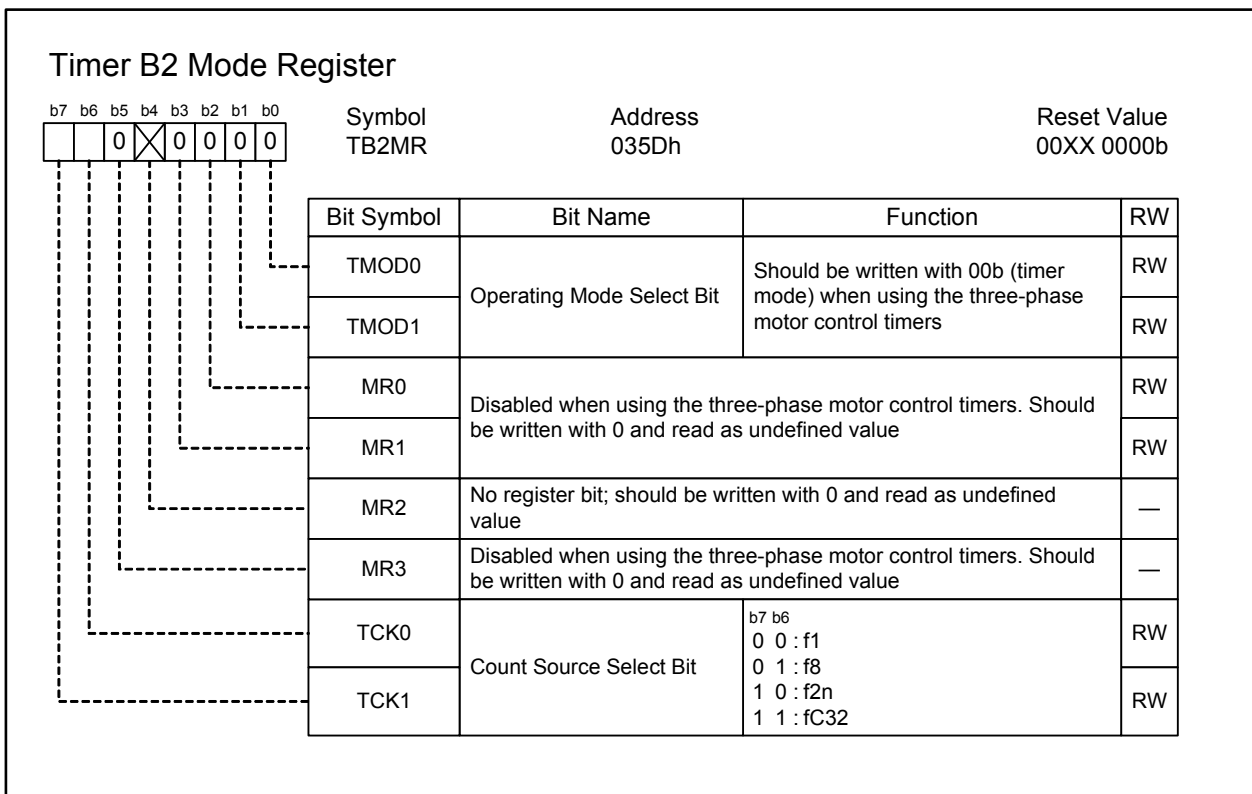
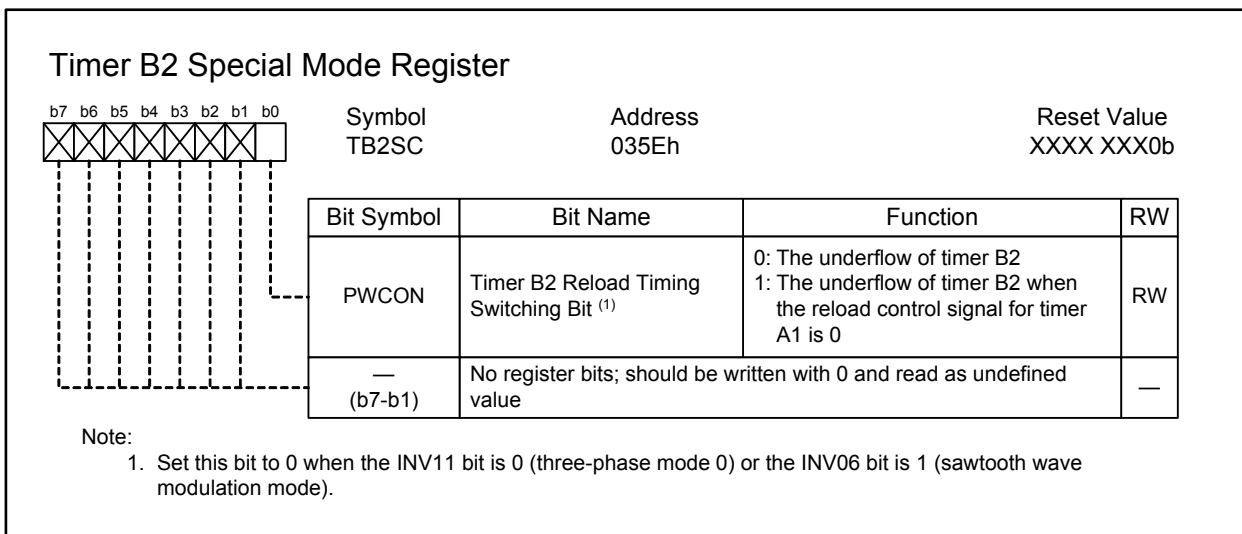


Figure 17.8 TB2MR Register When Using Three-phase Motor Control Timers



**Figure 17.9 TB2SC Register**



### 17.3 Timers A4, A1, and A2

Timers A4, A1, and A2 are used for three-phase PWM output (U,  $\bar{U}$ , V,  $\bar{V}$ , W, and  $\bar{W}$ ) control when using the three-phase motor control timers.

These timers should be operated in one-shot timer mode. Every time timer B2 underflows, a trigger is input to timers A4, A1, and A2 to generate a one-shot pulse. If the values of registers TA4, TA1, and TA2 are rewritten every time a timer B2 interrupt occurs, the duty cycle of the PWM waveform can be varied. In three-phase mode 1, the value of registers TAI and TAI-1 is alternately reloaded to the counter at each timer B2 interrupt, which halves the timer B2 interrupt frequency ( $i = 4, 1, 2$ ).

Figure 17.10 shows registers TA1, TA2, TA4, TA11, TA21, and TA41 in the three-phase motor control timers. Figure 17.11 shows registers TA1MR, TA2MR, and TA4MR in this function. Figures 17.12 and 17.13 show registers TRGSR and TABSR, respectively, in this function.

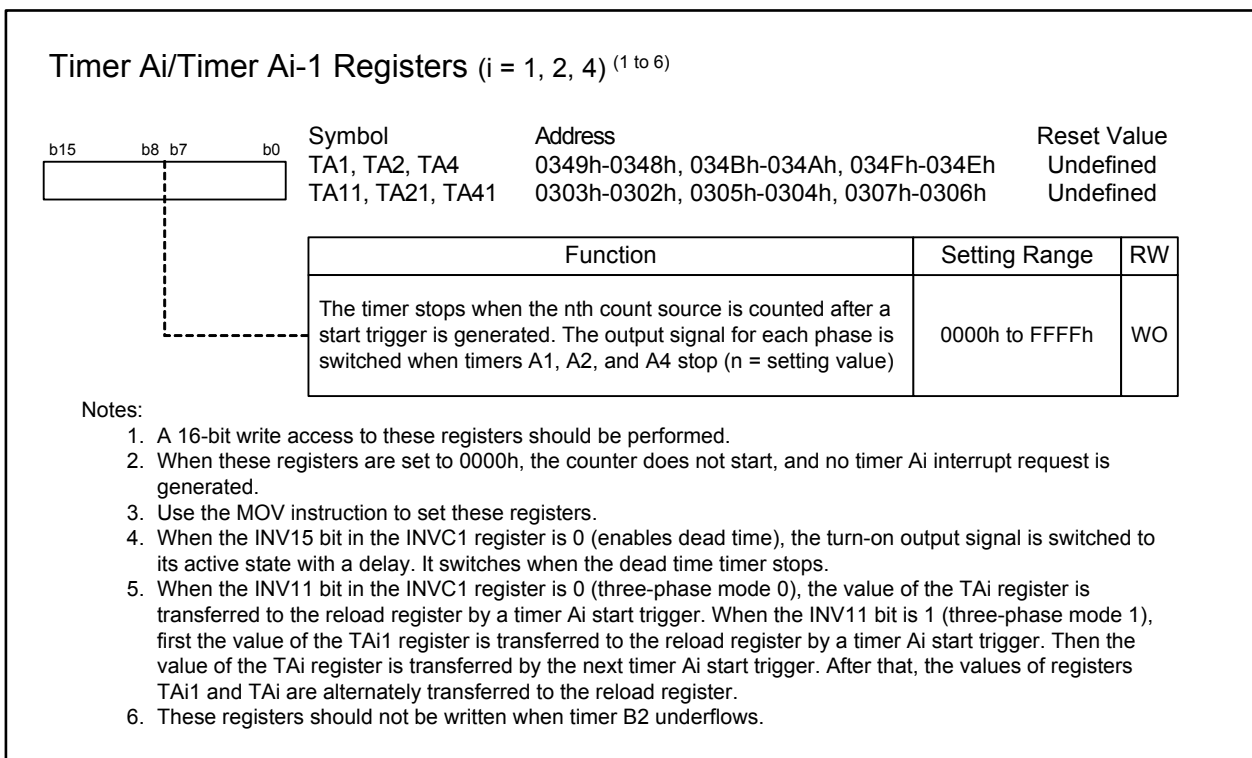
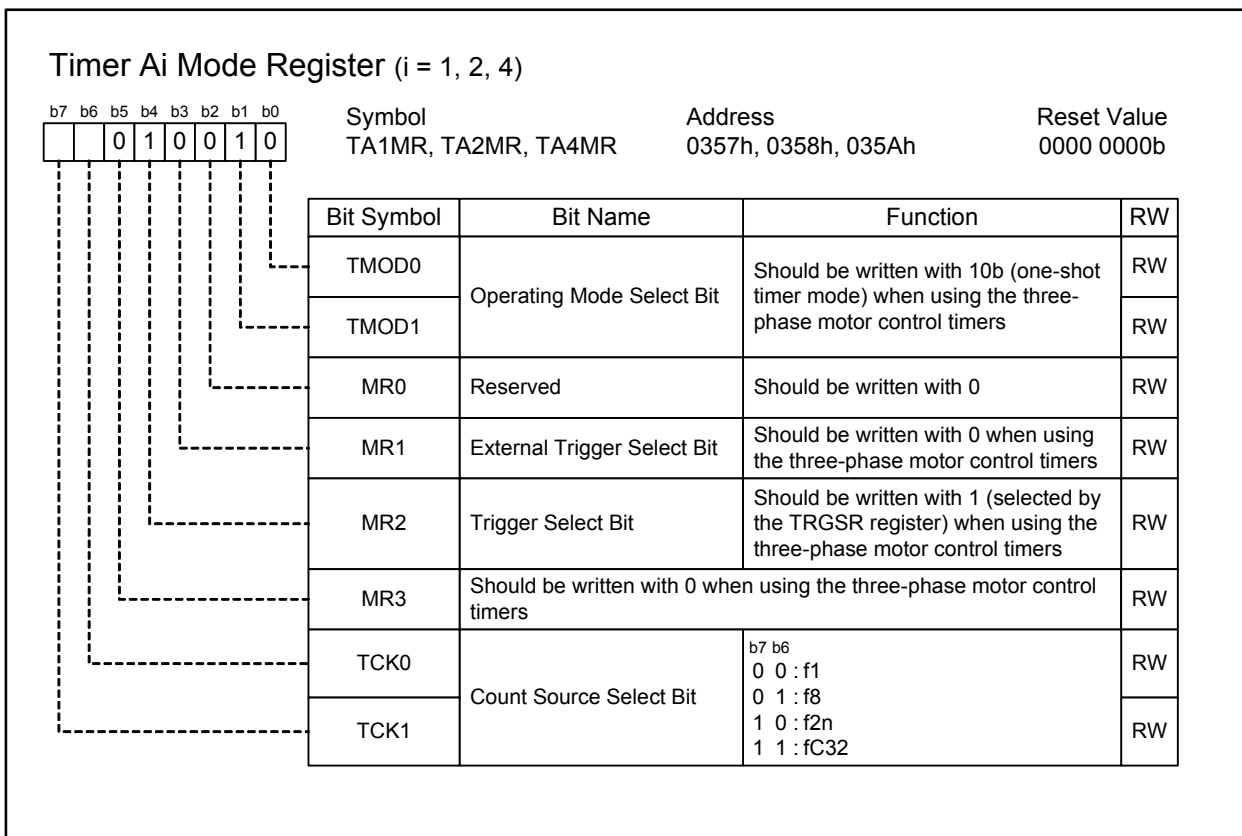


Figure 17.10 Registers TA1, TA2, TA4, TA11, TA21, and TA41



**Figure 17.11 Registers TA1MR, TA2MR, and TA4MR When Using Three-phase Motor Control Timers**

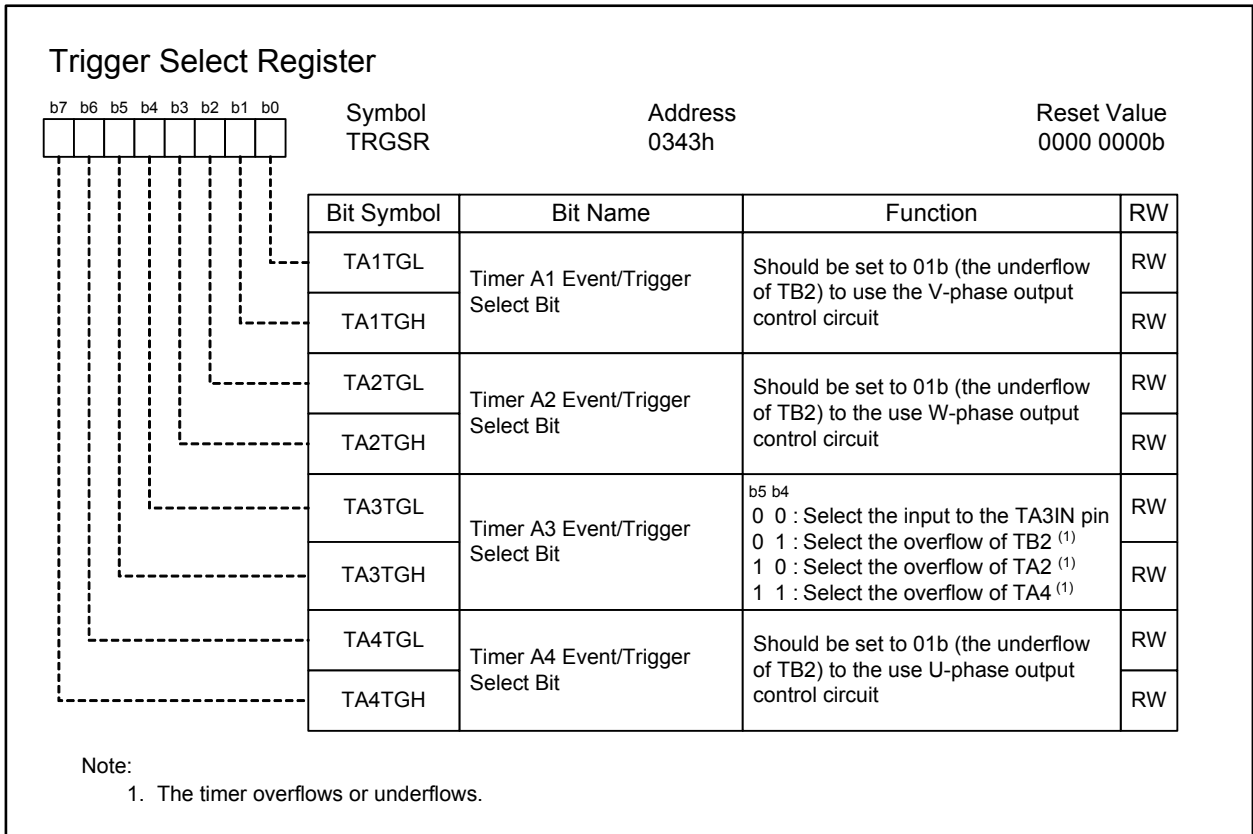


Figure 17.12 TRGSR Register in Three-phase Motor Control Timers

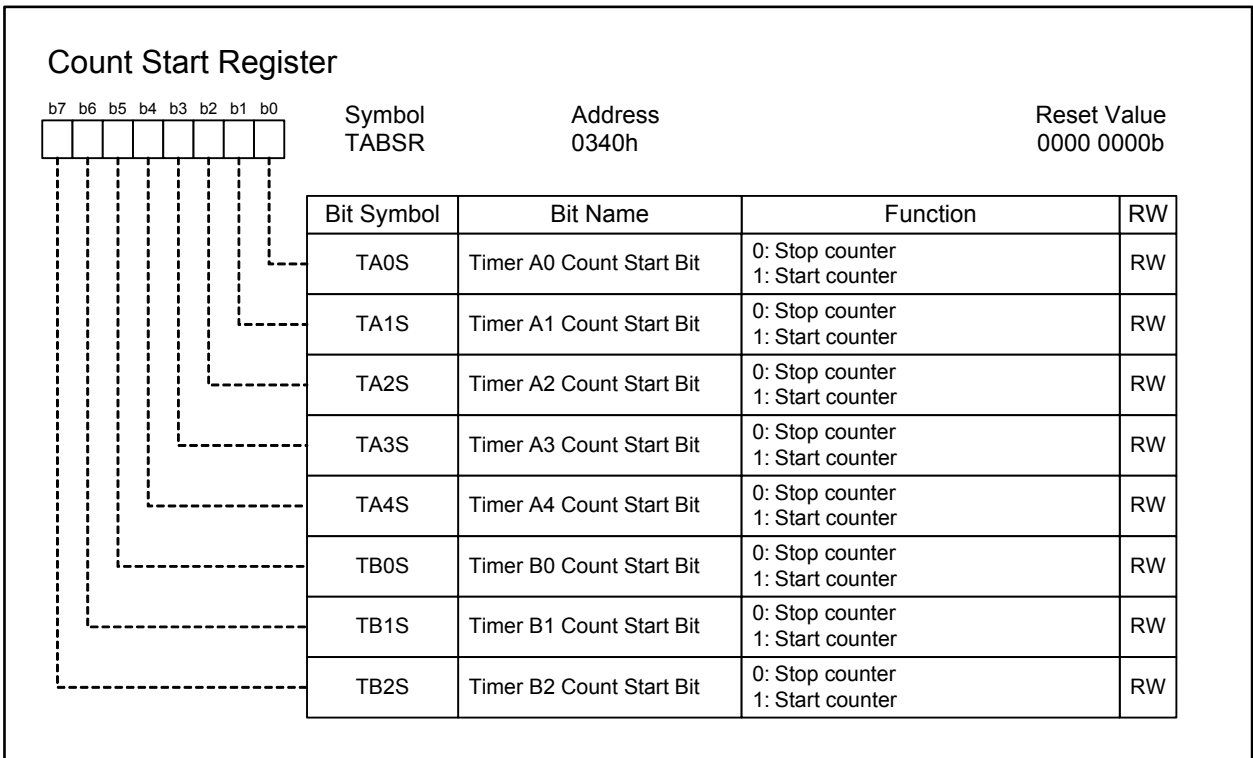


Figure 17.13 TABSR Register

## 17.4 Simultaneous Conduction Prevention and Dead Time Timer

The three-phase motor control timers offer two ways to avoid shoot-through, which occurs when high-side and low-side transistors are simultaneously turned on.

One is “simultaneous turn-on signal output disable function”. This function prevents high-side and low-side transistors from being inadvertently switched to active due to events like program errors. The other is by the use of dead time timers. A dead time timer delays the turn-on of one transistor in order to ensure that an adequate time (the dead time) passes after the other is turned off.

To disable simultaneous turn-on output signals, the INV04 bit in the INVC0 register should be set to 1. If outputs for any pair of phases (U and  $\bar{U}$ , V and  $\bar{V}$ , or W and  $\bar{W}$ ) are simultaneously switched to an active state, every three-phase motor control output pin becomes high-impedance. Figure 17.14 shows an example of output waveform when simultaneous turn-on signal output is disabled.

To enable the dead time timer, the INV15 bit in the INVC1 register should be set to 0. The DTT register determines the dead time. Figure 17.15 shows the DTT register and Figure 17.16 shows an example of output waveform on using dead time timer.

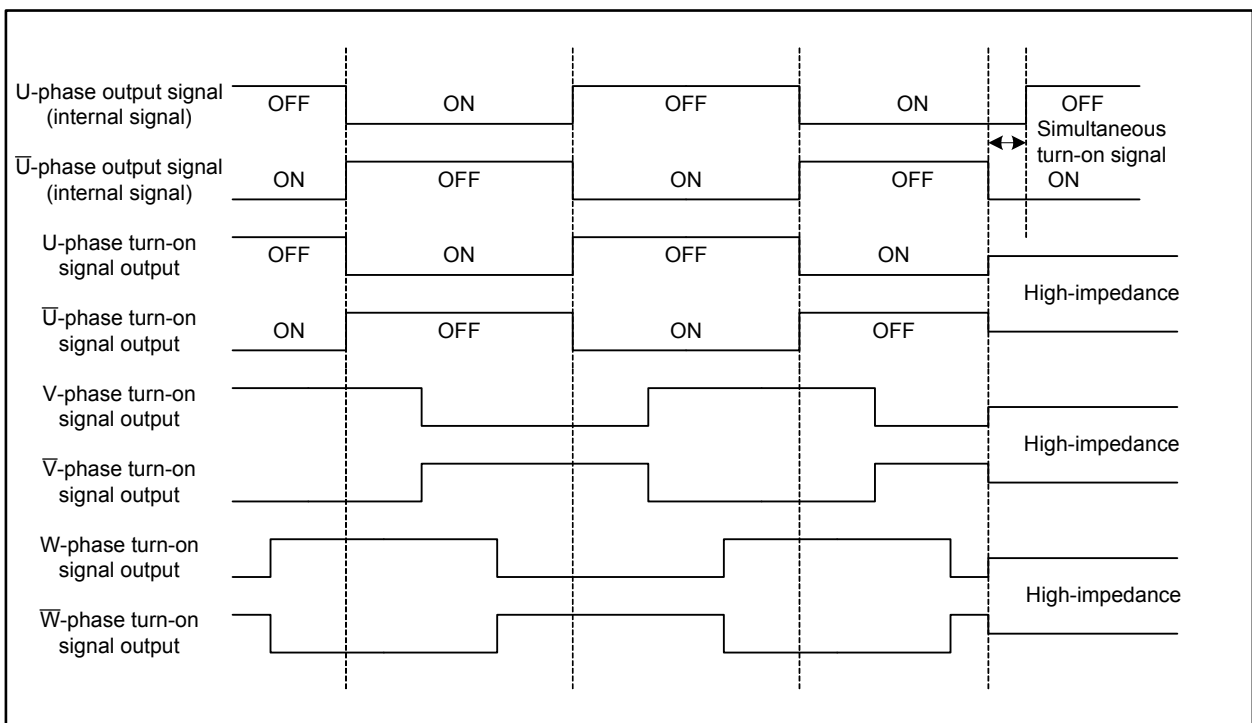


Figure 17.14 Output Waveform When Simultaneous Turn-on Signal Output is Disabled

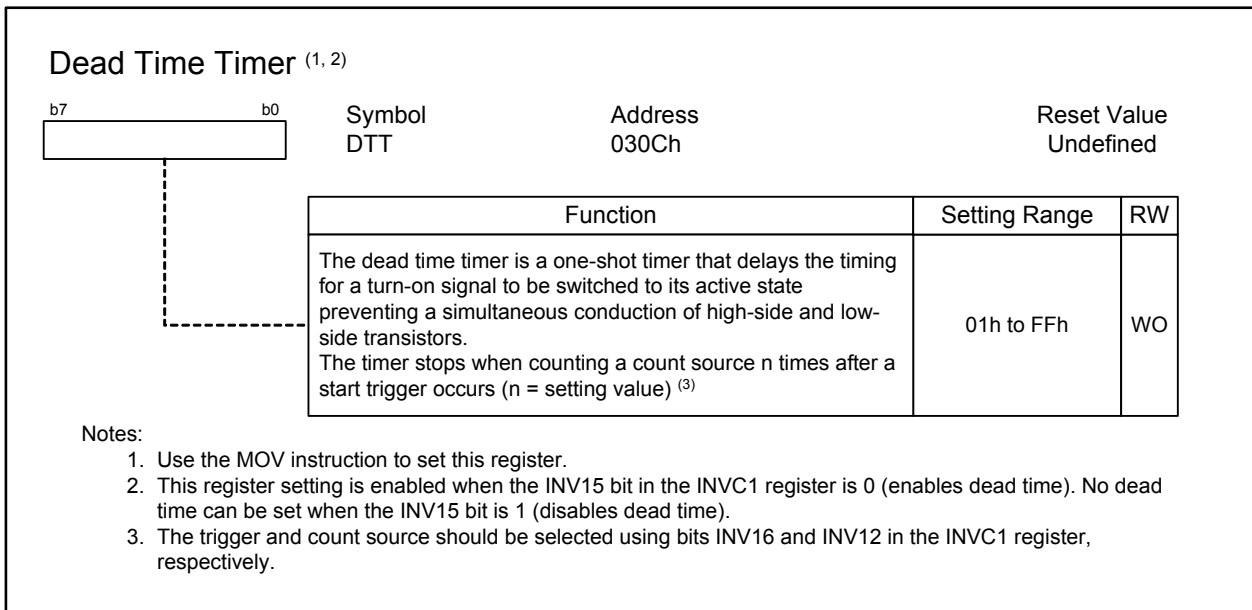


Figure 17.15 DTT Register

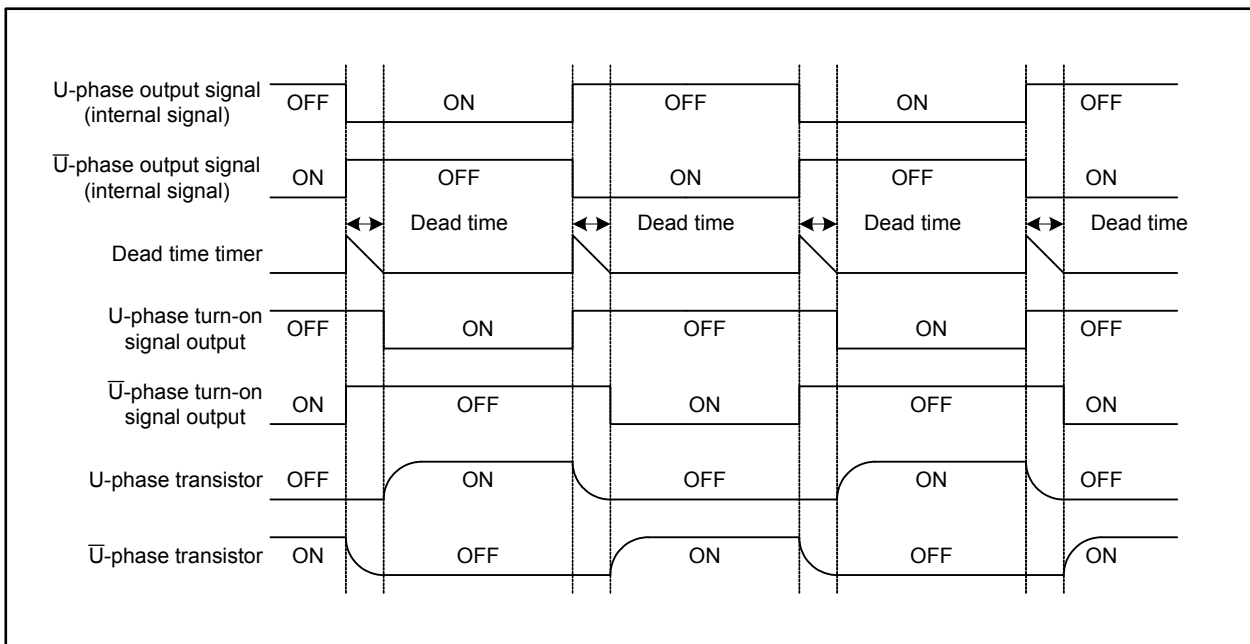


Figure 17.16 Output Waveform When Using Dead Time Timer

### 17.5 Three-phase Motor Control Timer Operation

Figures 17.17 and 17.18 show an operation example of triangular wave modulation and sawtooth wave modulation, respectively.

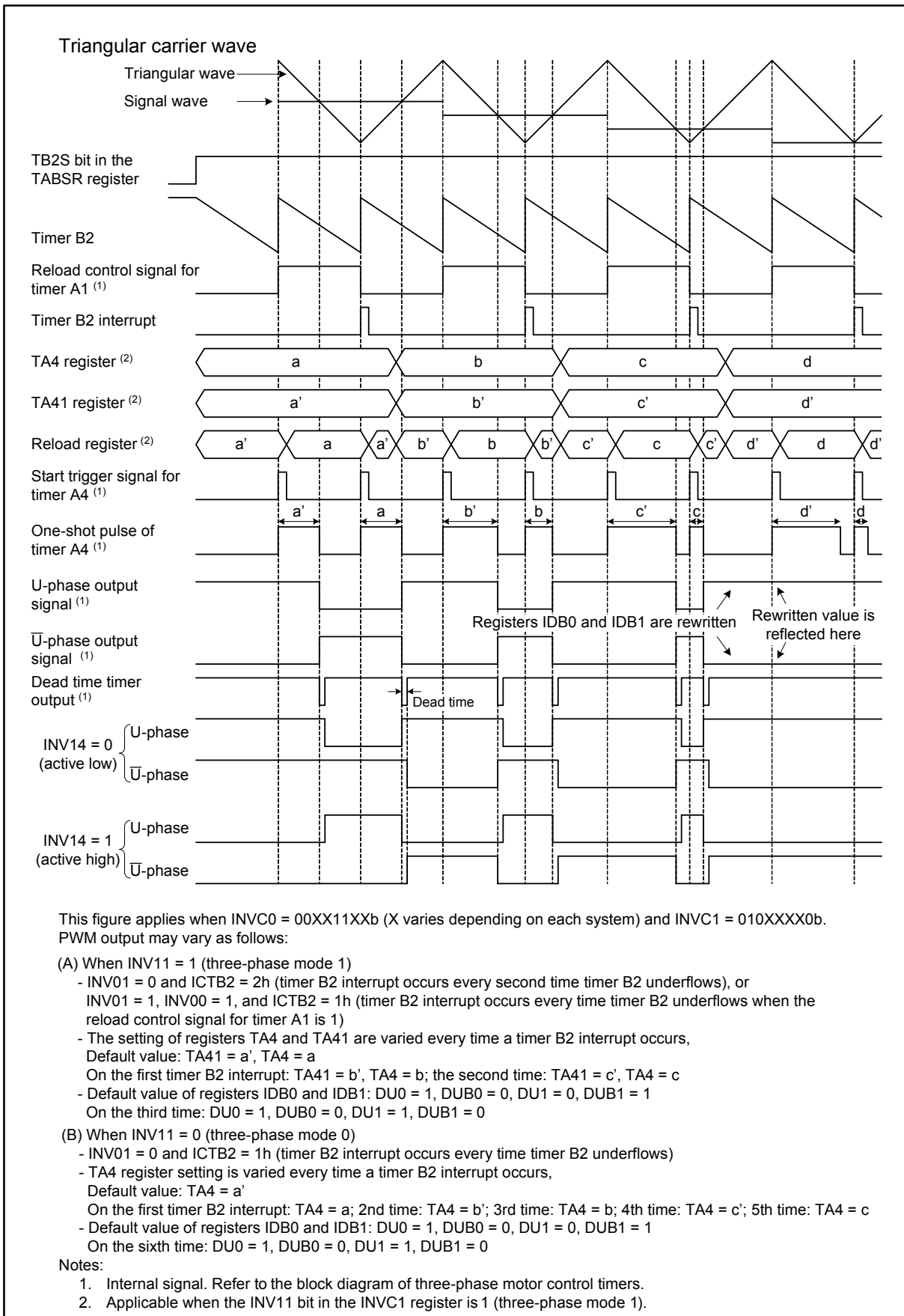
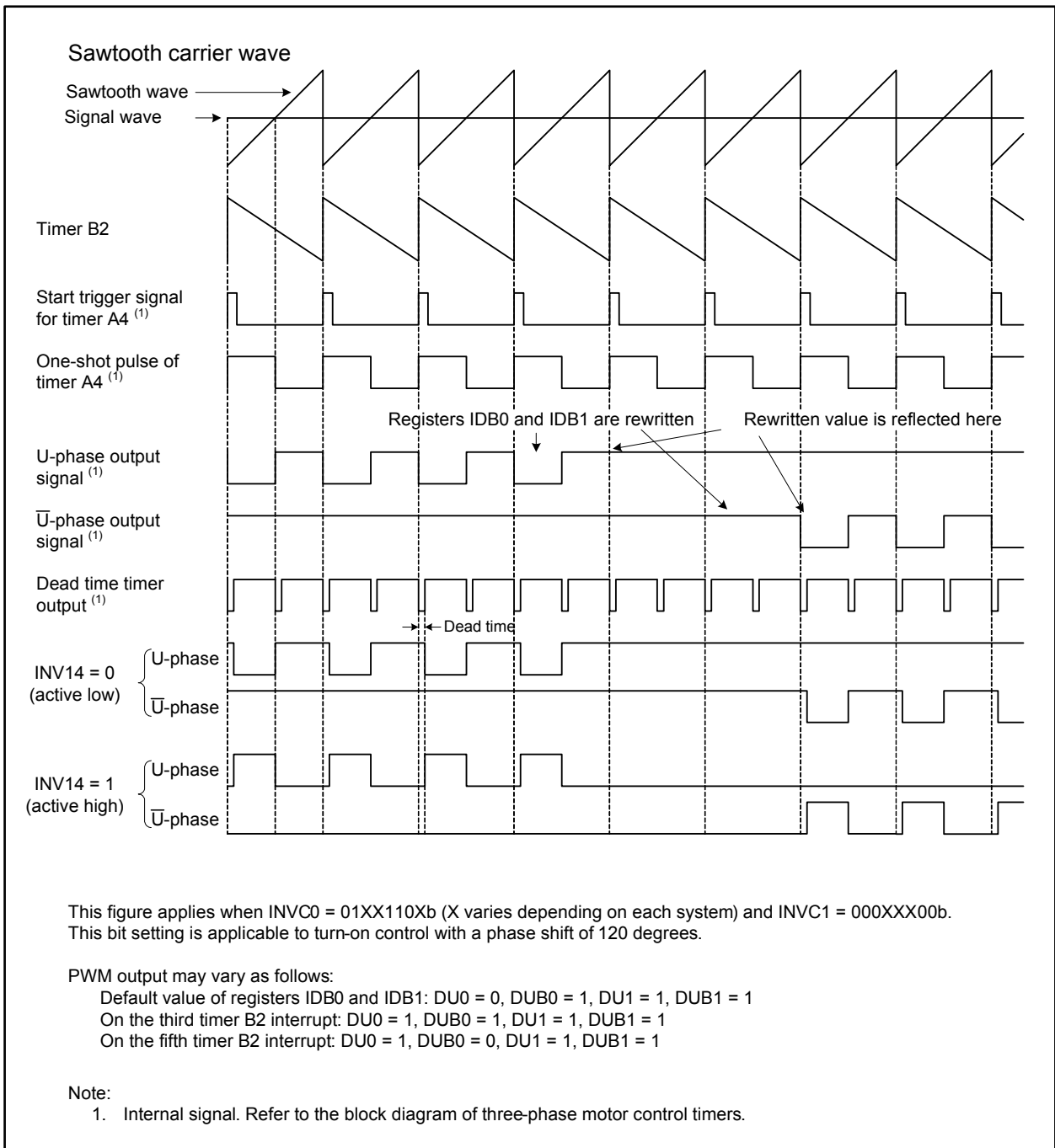


Figure 17.17 Triangular Wave Modulation Operation



**Figure 17.18 Sawtooth Wave Modulation Operation**

## 17.6 Notes on Three-phase Motor Control Timers

### 17.6.1 Shutdown

- When a low signal is applied to the  $\overline{\text{NMI}}$  pin with the following bit settings, pins TA1OUT, TA2OUT, and TA4OUT become high-impedance: the PM24 bit in the PM2 register is 1 (NMI enabled), the INV02 bit in the INVC0 register is 1 (three-phase motor control timers used), and the INV03 bit is 1 (three-phase motor control timer output enabled).

### 17.6.2 Register Setting

- Do not write to the TAI1 register before and after timer B2 underflows ( $i = 1, 2, 4$ ). Before writing to the TAI1 register, read the TB2 register to verify that sufficient time remains until timer B2 underflows. Then, immediately write to the TAI1 register so no interrupt handling is performed during this write procedure. If the TB2 register indicates little time remains until the underflow, write to the TAI1 register after timer B2 underflows.

### 17.6.3 Restriction for the Use of 64-pin Package's Product

Port P3 is available only for output pins V and  $\overline{\text{V}}$  in the 64-pin package. Use ports P7 and P8 for other output pins U,  $\overline{\text{U}}$ , W, and  $\overline{\text{W}}$ .



## 18. Serial Interface

The serial interface consists of nine channels: UART0 to UART8.

Each channel has an exclusive timer to generate the transmit/receive clock and operates independently. Figures 18.1 and 18.2 show block diagrams of UART0 to UART6 and UART7 and UART8, respectively.

UARTi supports the following modes:

- Synchronous serial interface mode (for UART0 to UART8)
- Asynchronous serial interface mode (UART mode) (for UART0 to UART8)
- Special mode 1 (I<sup>2</sup>C mode) (for UART0 to UART6)
- Special mode 2 (for UART0 to UART6)
- Special mode 4 (Bus collision detection: IE mode) (optional) <sup>(1)</sup> (for UART0 to UART6)

Figures 18.3 to 18.19 show registers associated with UARTi (i = 0 to 8).

Refer to the tables listing each mode for registers and pin settings.

Note:

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

**Table 18.1 Comparison of UART0 to UART8 Functions**

Mode/Function	UART0, UART1, UART3 to UART6	UART2	UART7, UART8
Synchronous serial interface mode	Available	Available	Available
Serial data logic inversion	Available	Available	Not available
Asynchronous serial interface mode (UART mode)	Available	Available	Available
CTS/RTS function selection	Available	Available	Available
TXD and RXD I/O polarity selection	Available	Available	Not available
Special mode 1 (I <sup>2</sup> C mode)	Available	Available	Not available
Special mode 2	Available	Available	Not available
Special mode 4 (IE mode) (optional) <sup>(1)</sup>	Available	Available	Not available
Pins TXD and RXD output mode	Push-pull output	N-channel open drain output	Push-pull output

Note:

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

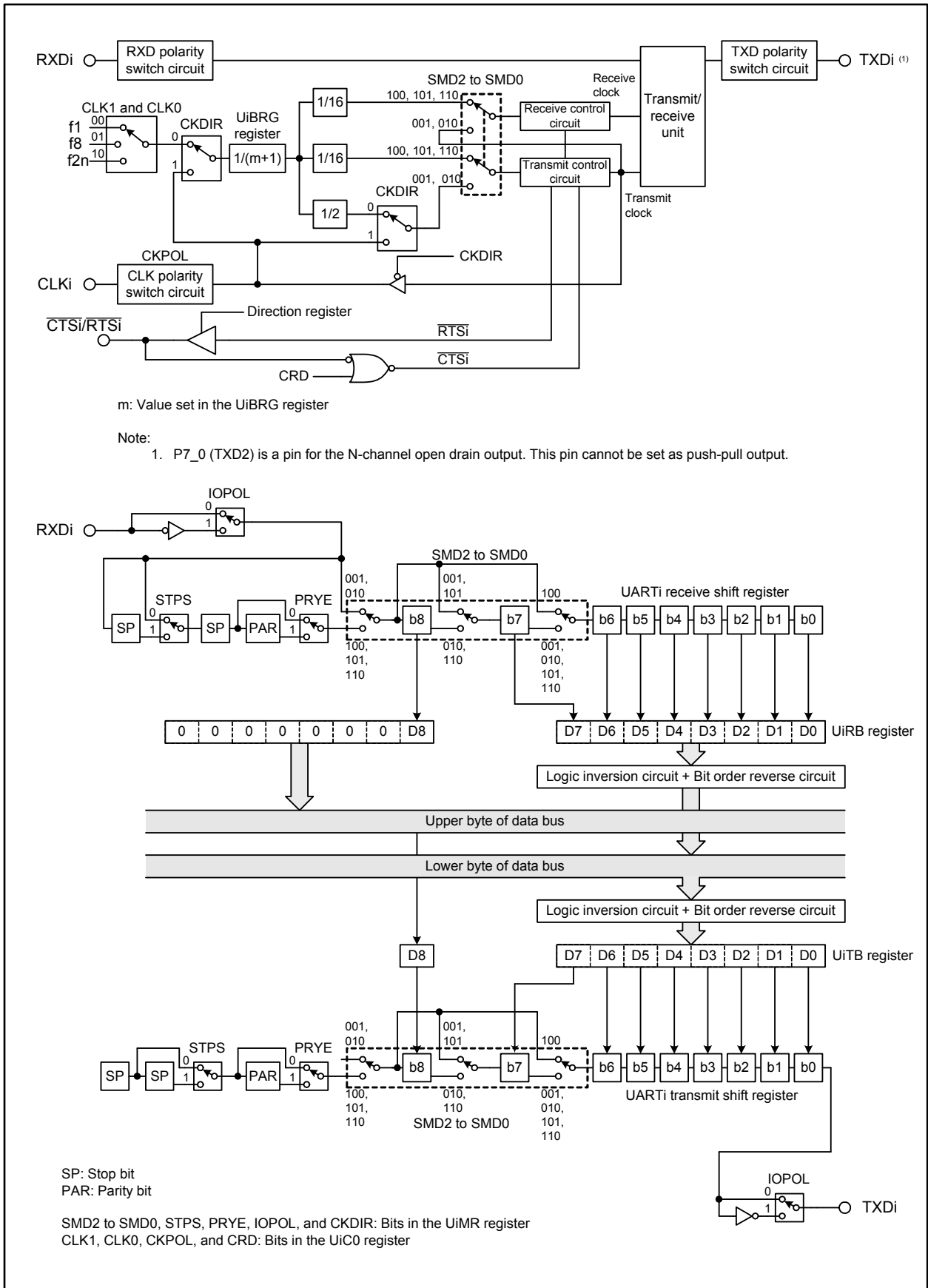


Figure 18.1 UARTi Block Diagram (i = 0 to 6)

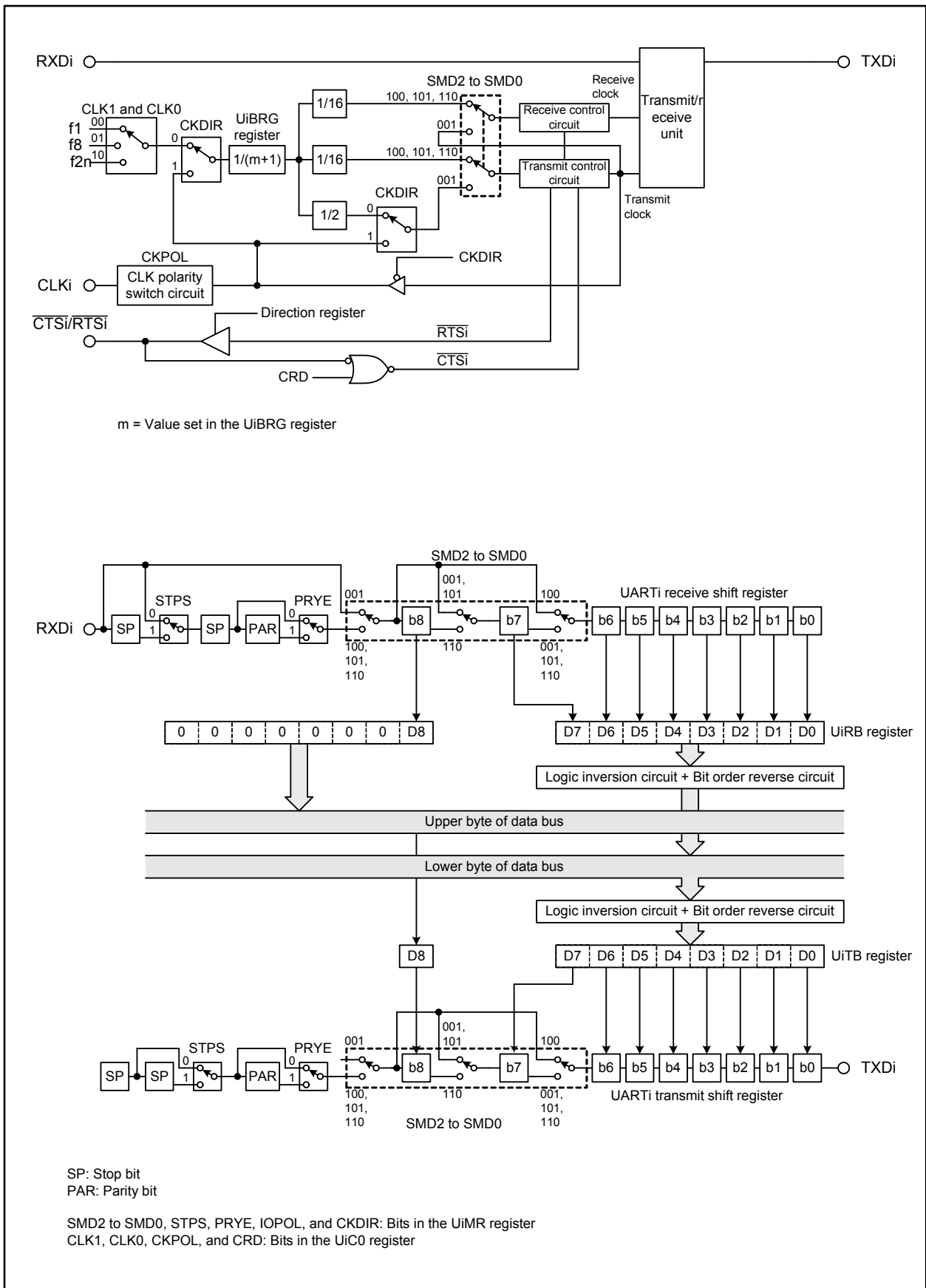
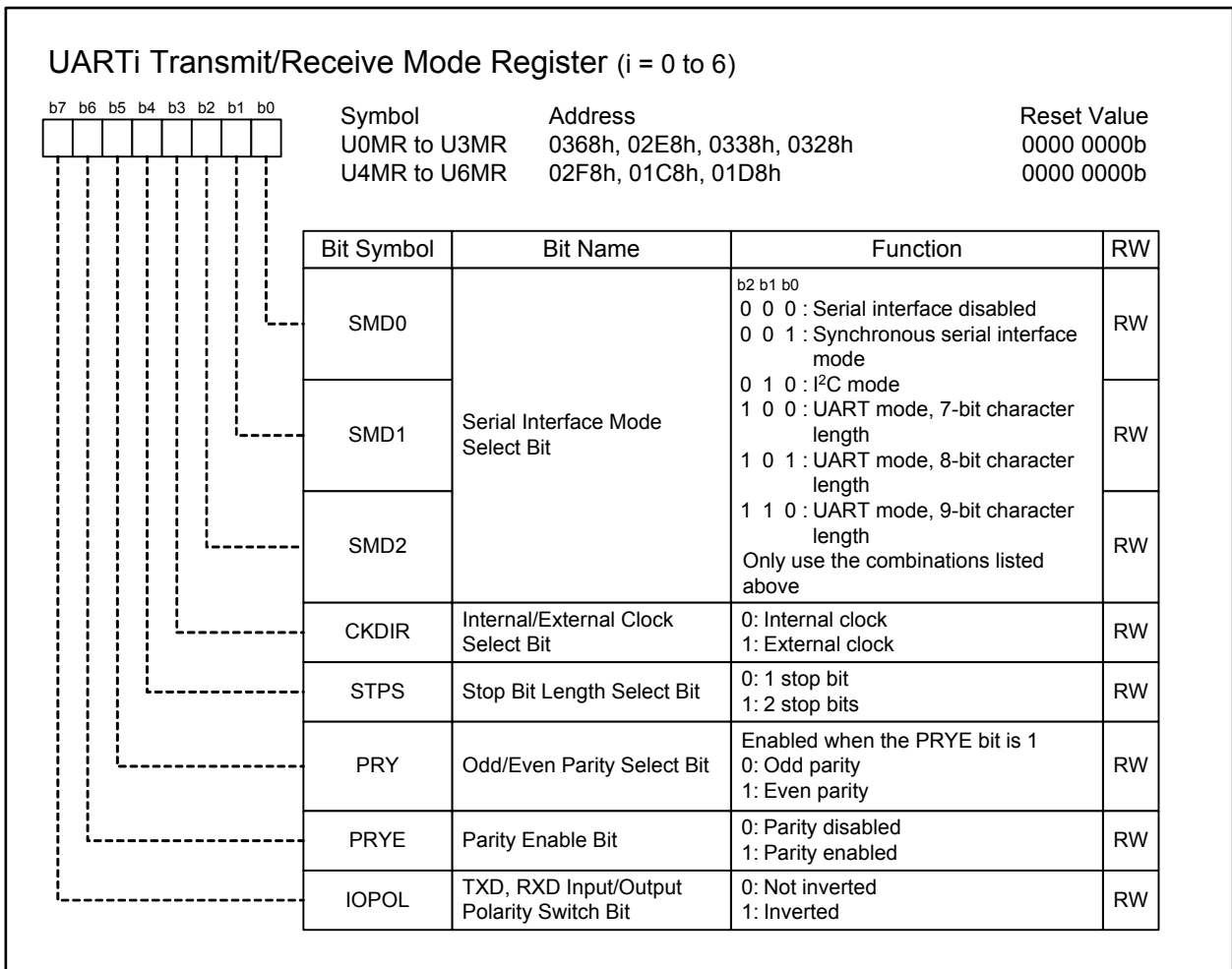
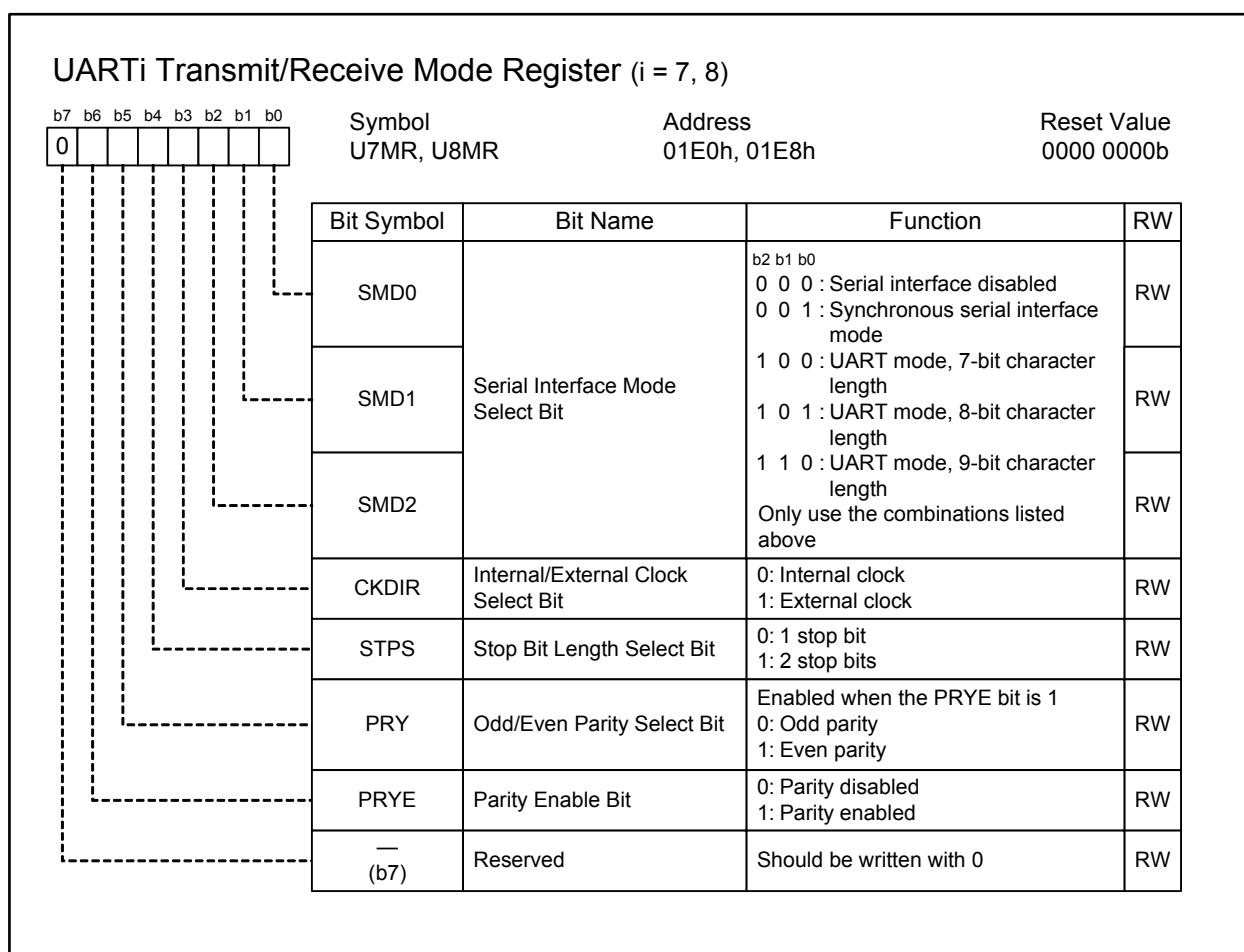


Figure 18.2 UARTi Block Diagram (i = 7, 8)



**Figure 18.3 Registers U0MR to U6MR**



**Figure 18.4 Registers U7MR and U8MR**

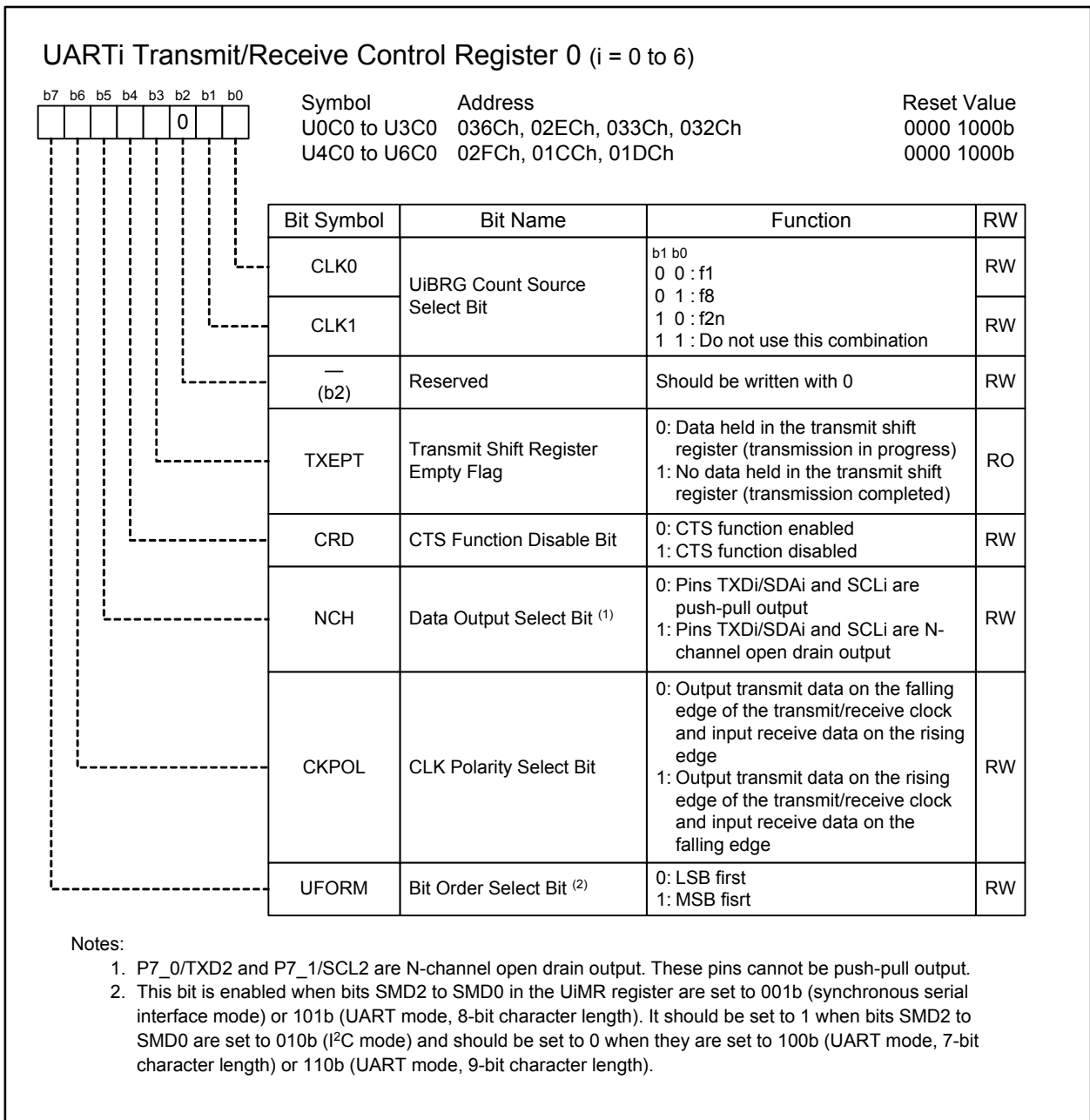
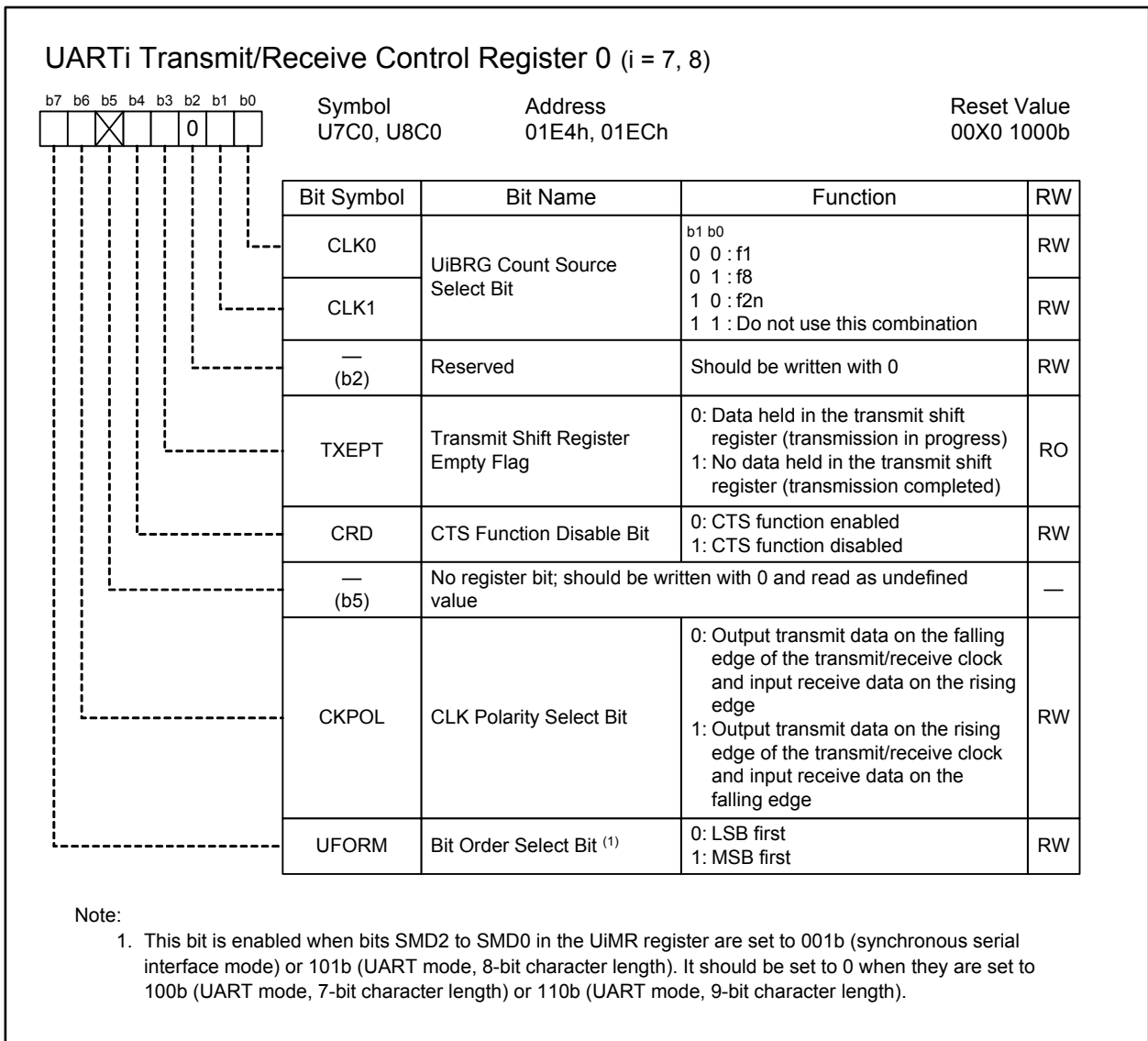


Figure 18.5 Registers U0C0 to U6C0



**Figure 18.6 Registers U7C0 and U8C0**

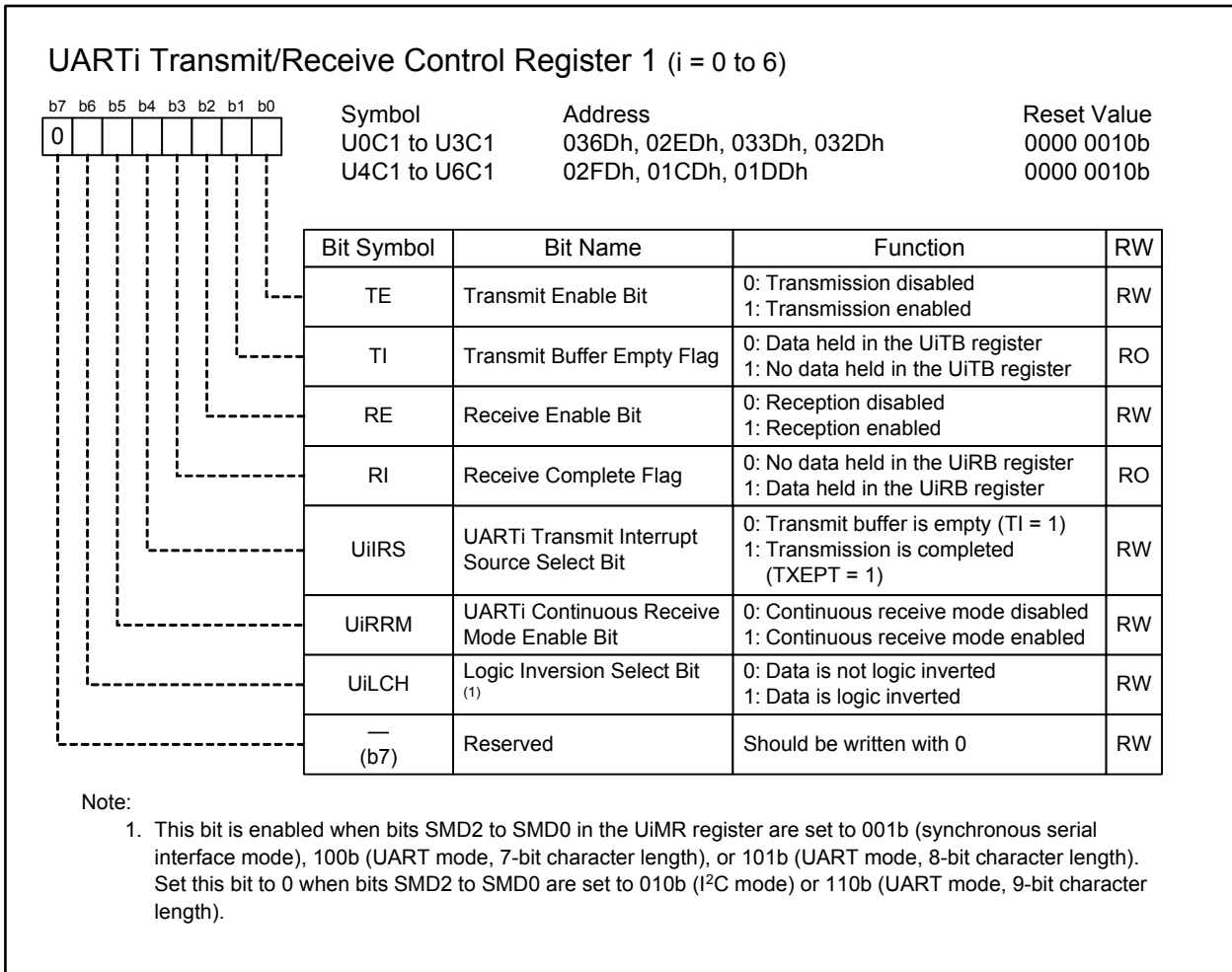


Figure 18.7 Registers U0C1 to U6C1

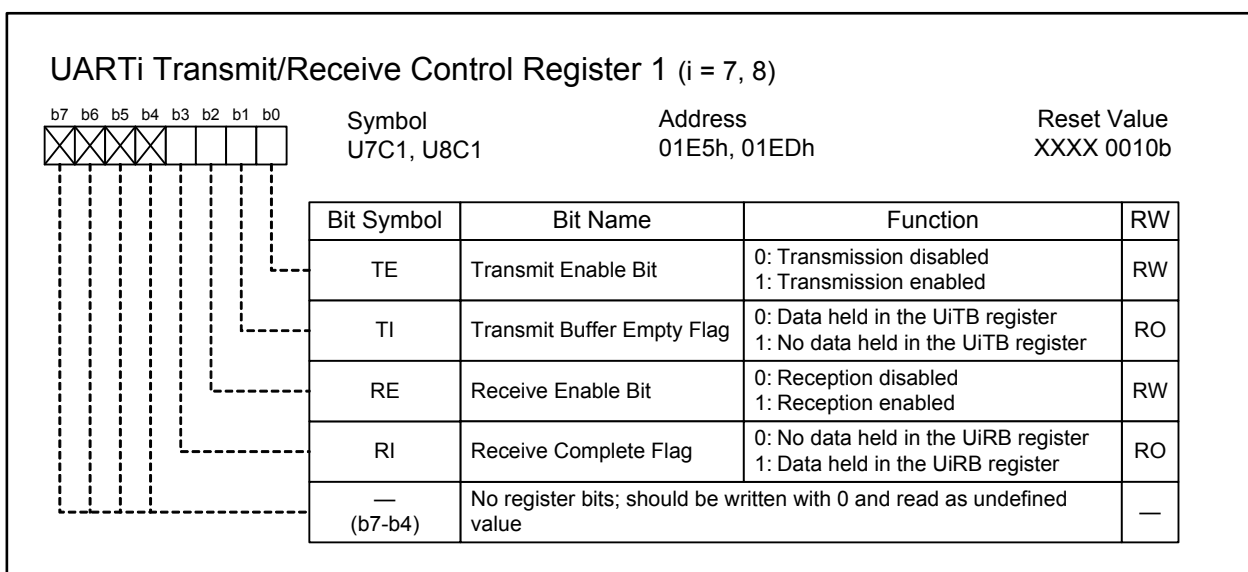


Figure 18.8 Registers U7C1 and U8C1



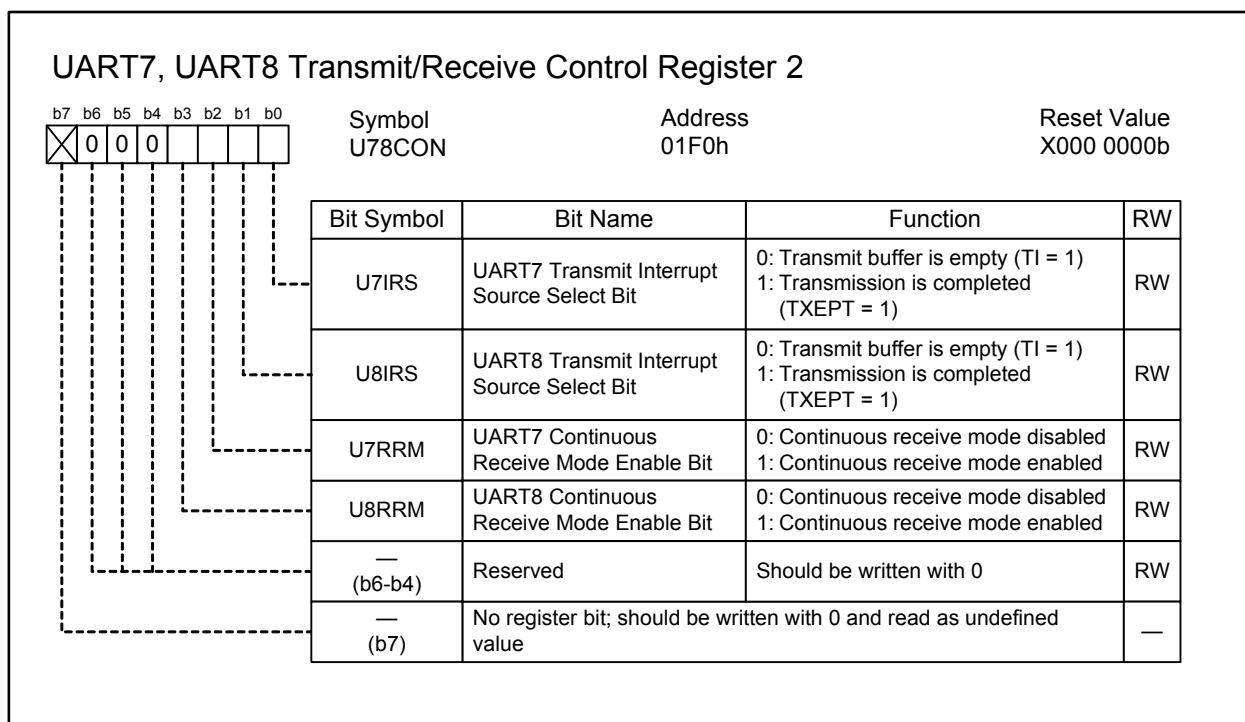


Figure 18.9 U78CON Register

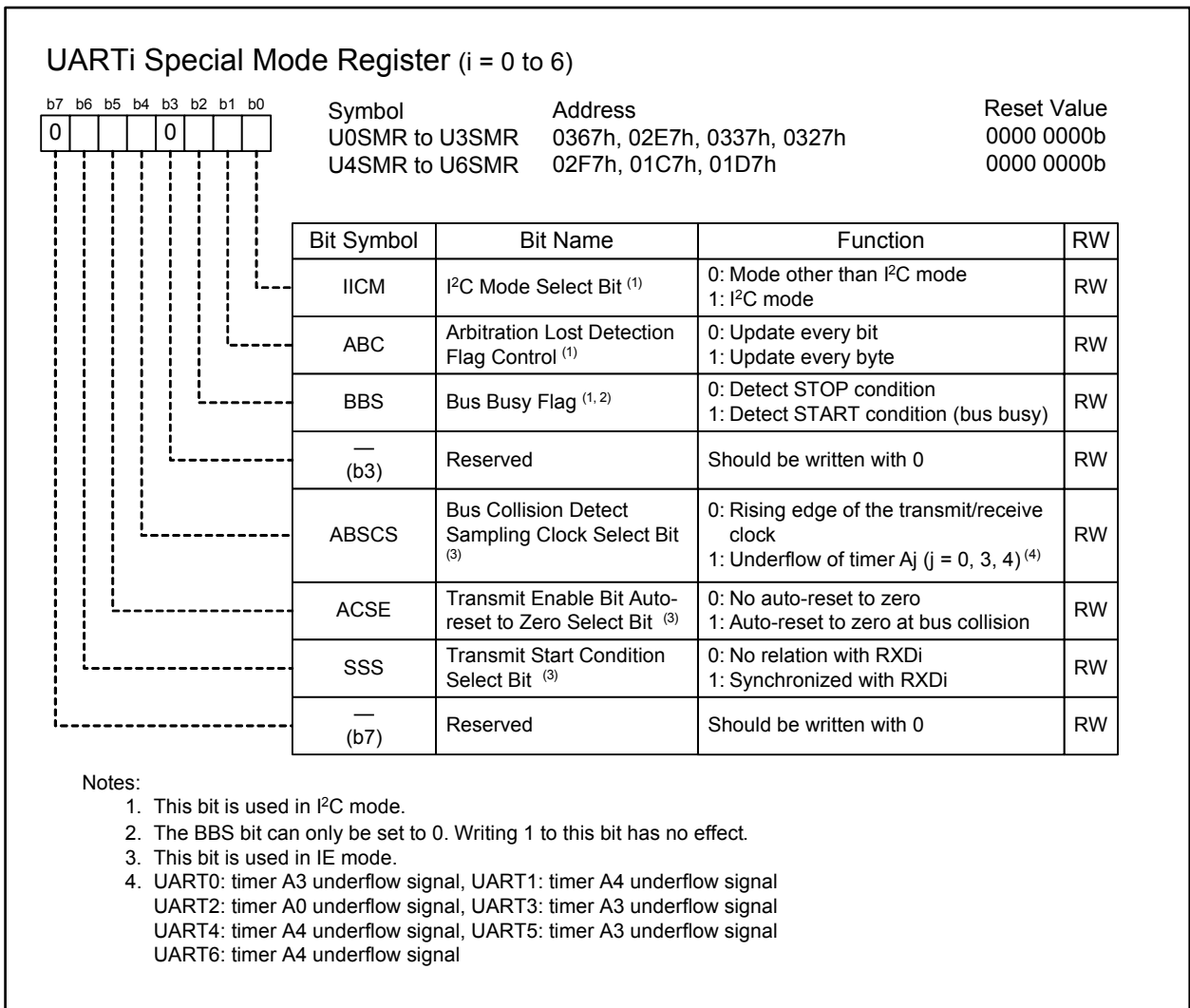
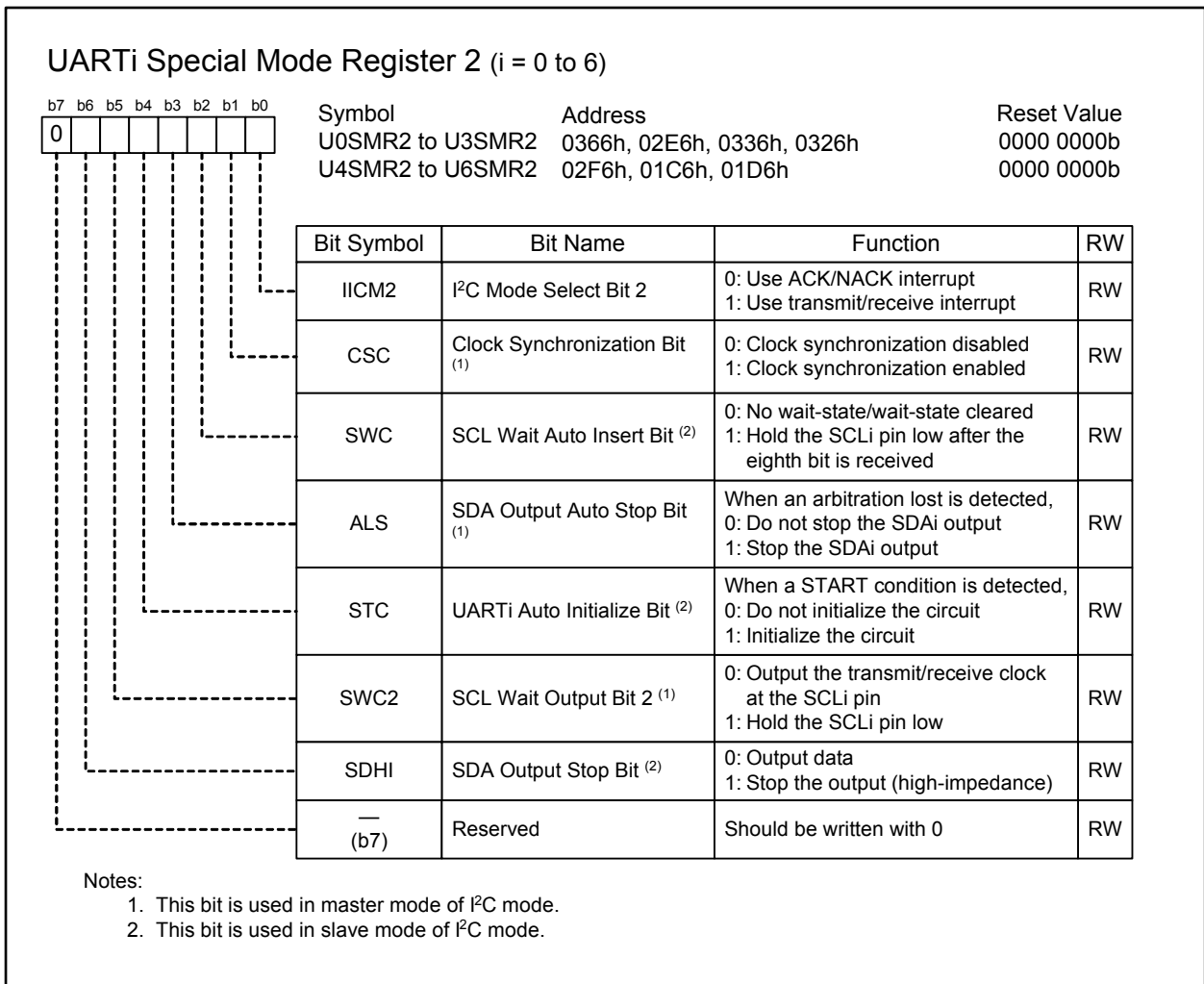


Figure 18.10 Registers U0SMR to U6SMR



**Figure 18.11 Registers U0SMR2 to U6SMR2**

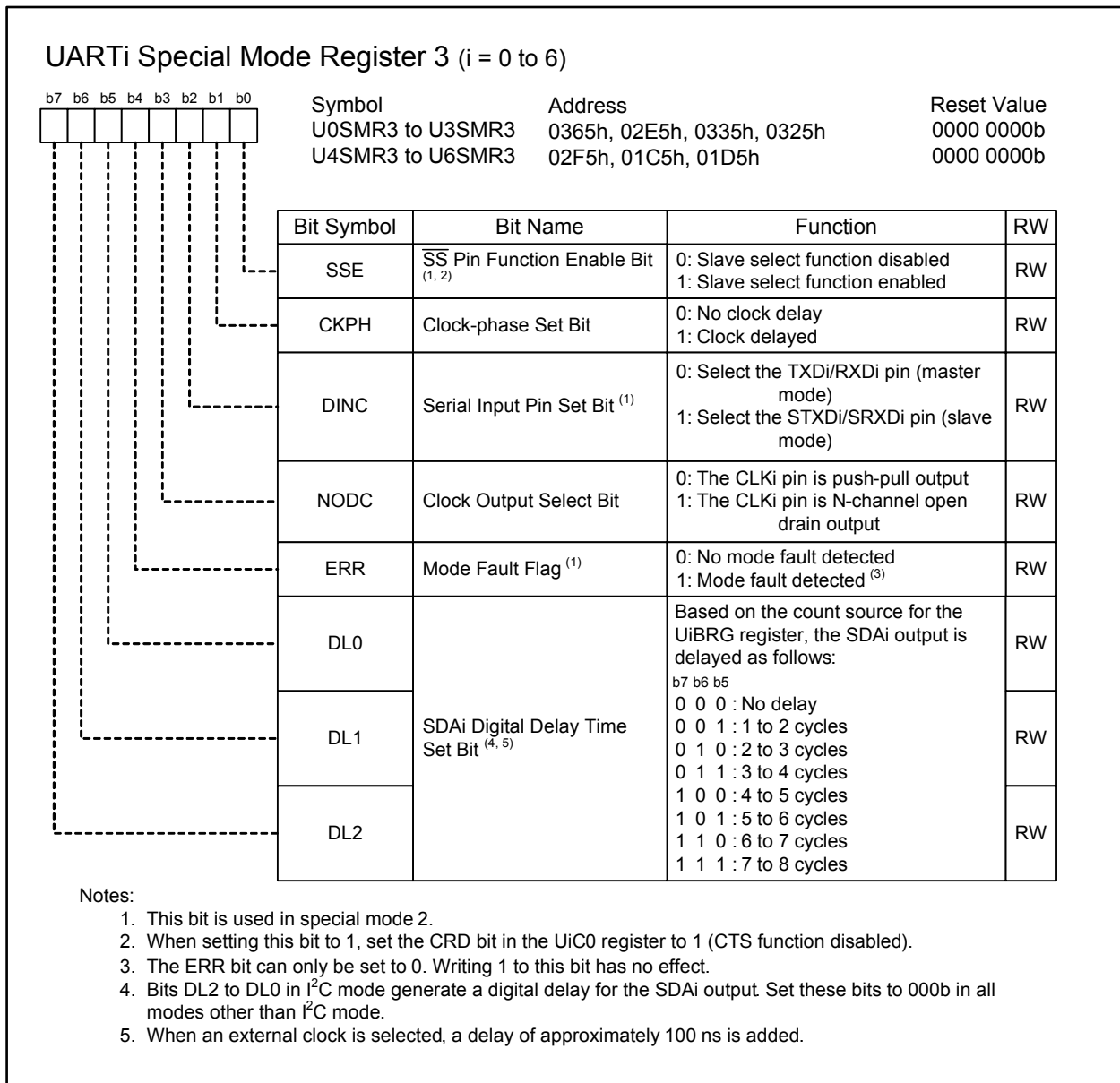


Figure 18.12 Registers U0SMR3 to U6SMR3

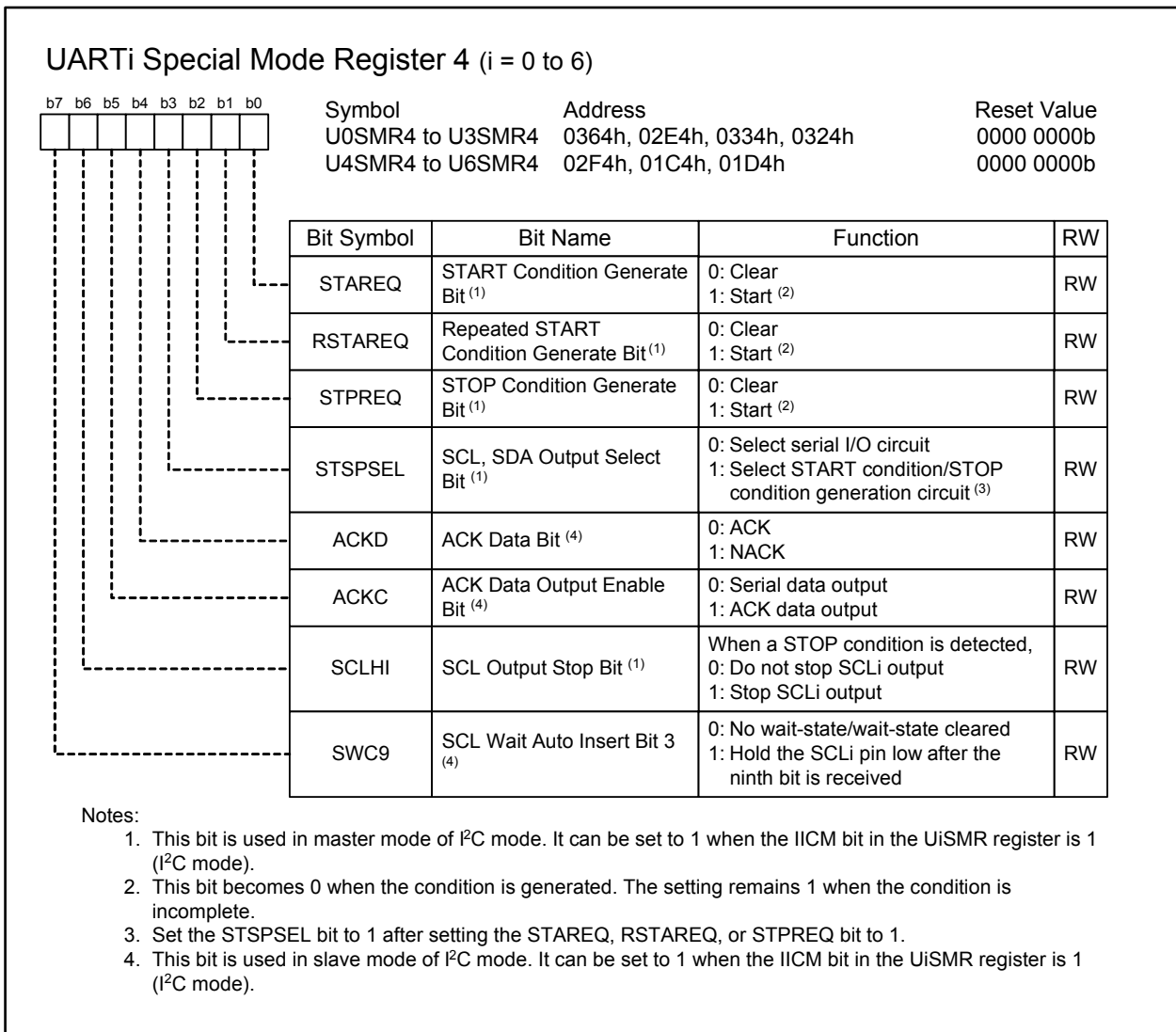


Figure 18.13 Registers U0SMR4 to U6SMR4

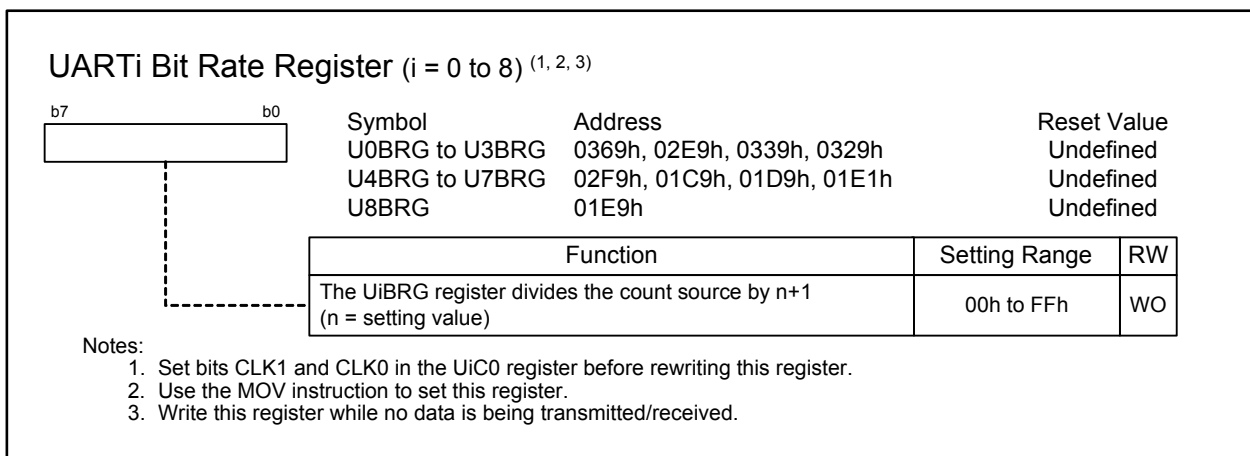


Figure 18.14 Registers U0BRG to U8BRG

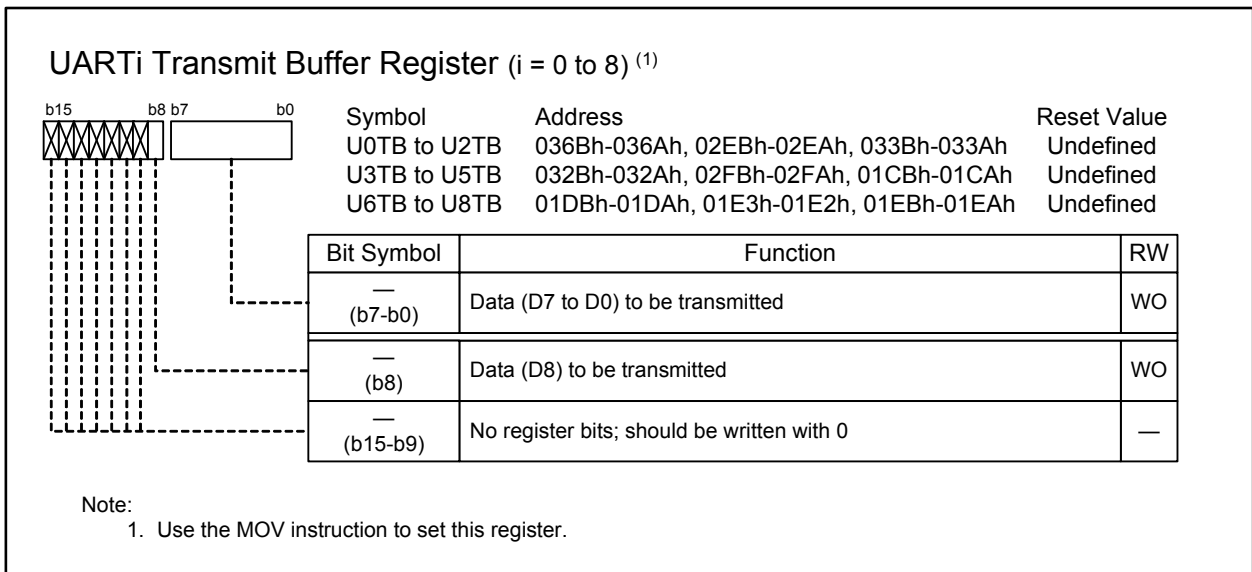


Figure 18.15 Registers U0TB to U8TB

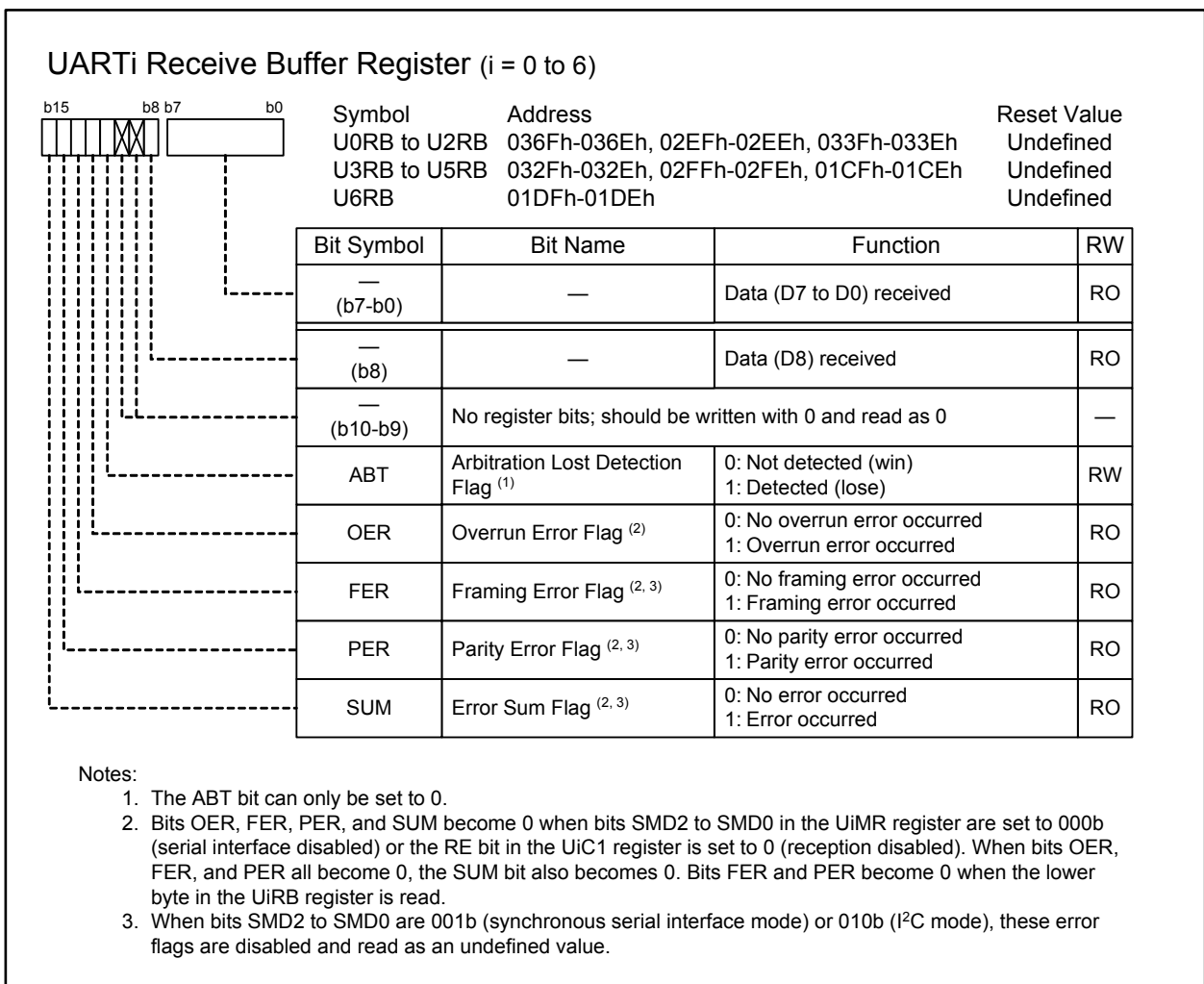
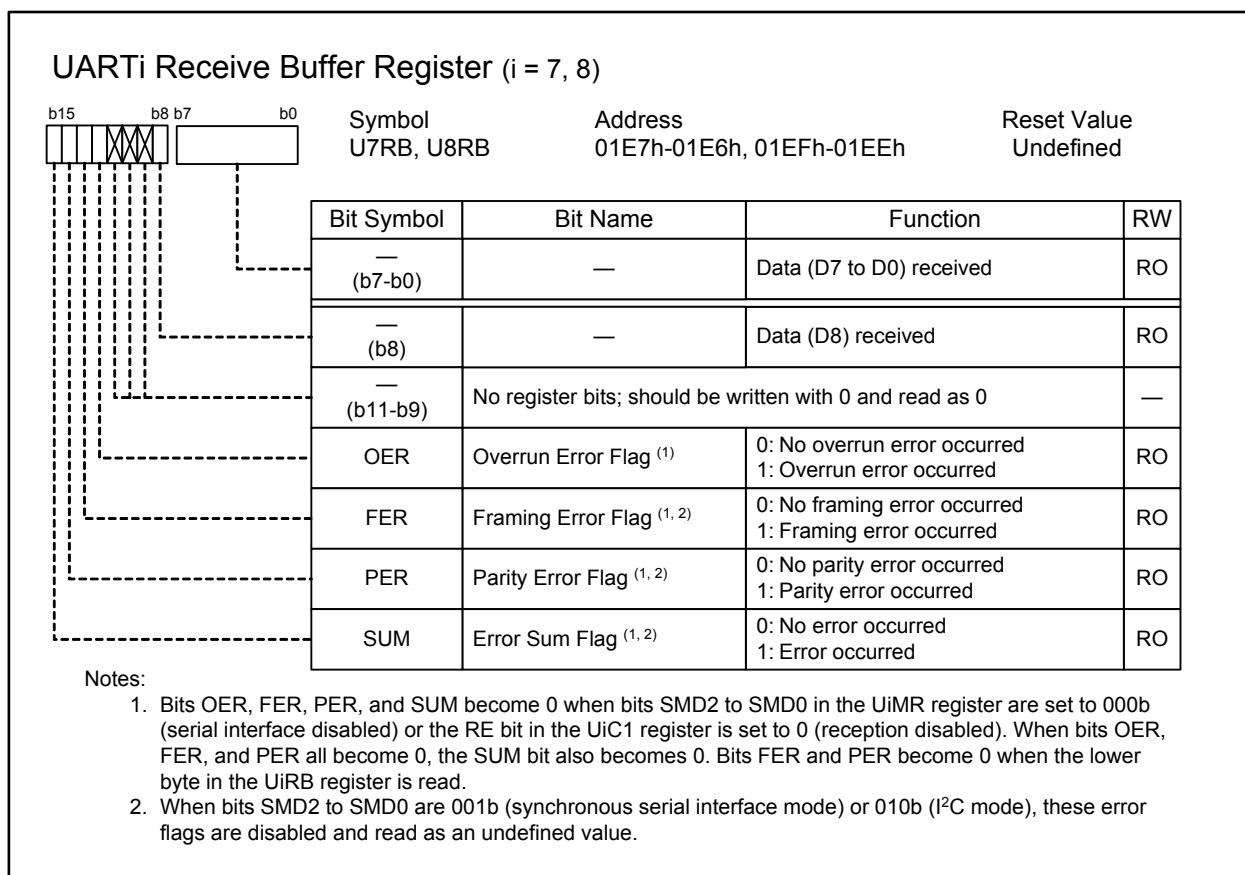
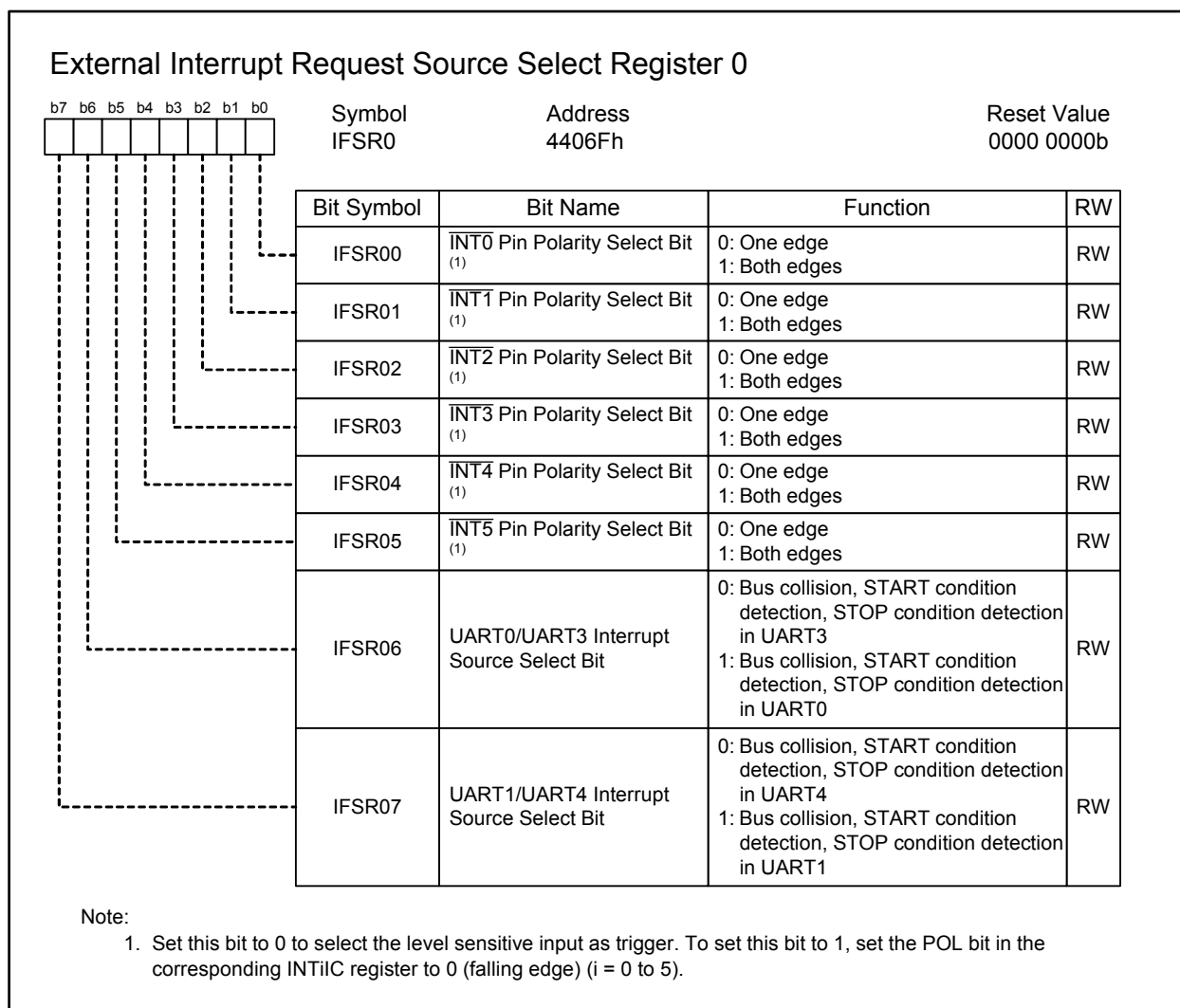


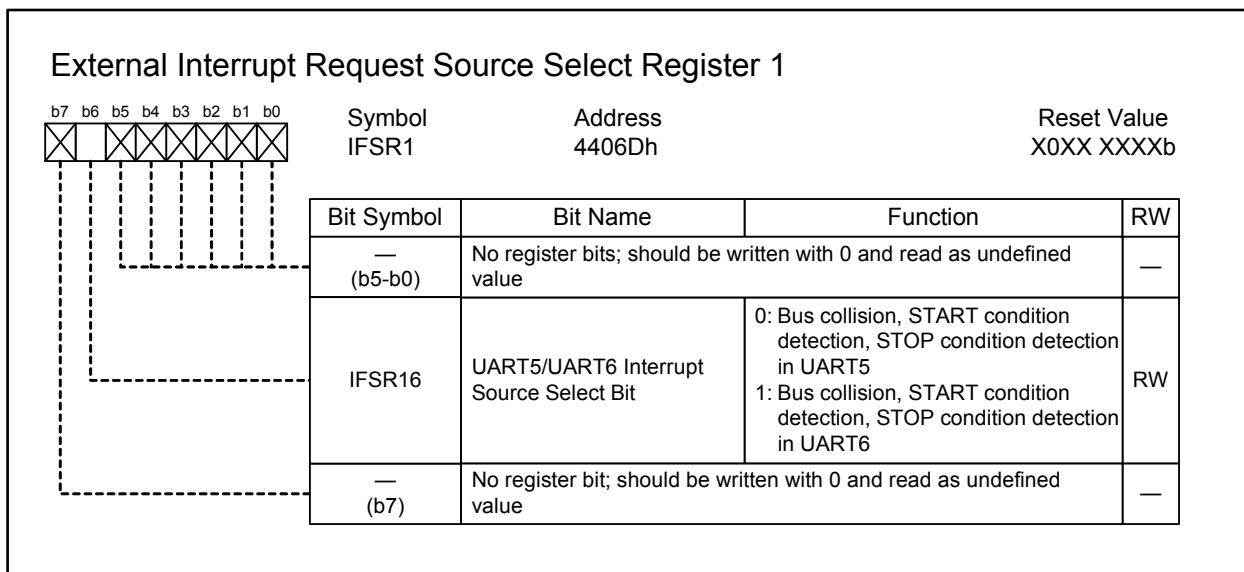
Figure 18.16 Registers U0RB to U6RB



**Figure 18.17 Registers U7RB and U8RB**

**Figure 18.18 IFSR0 Register**





**Figure 18.19 IFSR1 Register**

## 18.1 Synchronous Serial Interface Mode

The synchronous serial interface mode allows data transmission/reception synchronized with the transmit/receive clock. Table 18.2 lists specifications of synchronous serial interface mode.

**Table 18.2 Synchronous Serial Interface Mode Specifications**

Item	Specification
Data format	8-bit character length
Transmit/receive clock	<ul style="list-style-type: none"> <li>The CKDIR bit in the UiMR register is 0 (internal clock) (<math>i = 0</math> to 8):  <math display="block">\frac{fx}{2(m+1)} \quad fx = f1, f8, f2n; m: \text{UiBRG register setting value, } 00h \text{ to } FFh</math> </li> <li>The CKDIR bit is 1 (external clock): input to the CLKi pin</li> </ul>
Transmit/receive control	CTS function enabled, RTS function enabled, or CTS/RTS function disabled
Transmit start conditions	<p>The conditions for starting data transmission are as follows <sup>(1)</sup>:</p> <ul style="list-style-type: none"> <li>The TE bit in the UiC1 register is 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register is 0 (data held in the UiTB register)</li> <li>Input level at the <math>\overline{\text{CTS}}_i</math> pin is low when the CTS function is selected</li> </ul>
Receive start conditions	<p>The conditions for starting data reception are as follows <sup>(1)</sup>:</p> <ul style="list-style-type: none"> <li>The RE bit in the UiC1 register is 1 (reception enabled)</li> <li>The TE bit in the UiC1 register is 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register is 0 (data held in the UiTB register)</li> <li>Input level at the <math>\overline{\text{CTS}}_i</math> pin is low when the CTS function is selected</li> </ul>
Interrupt request generating timing	<p>In transmit interrupt, one of the following conditions can be selected by setting the UiIRS bit in registers U0C1 to U6C1 and U78CON:</p> <ul style="list-style-type: none"> <li>The UiIRS bit is 0 (transmit buffer is empty): when data is transferred from the UiTB register to the UARTi transmit shift register (when the transmission has started)</li> <li>The UiIRS bit is 1 (transmission is completed): when data transmission from the UARTi transmit shift register is completed</li> </ul> <p>In receive interrupt,</p> <ul style="list-style-type: none"> <li>When data is transferred from the UARTi receive shift register to the UiRB register (when the reception is completed)</li> </ul>
Error detection	<p>Overflow error <sup>(2)</sup></p> <p>This error occurs when the seventh bit of the next data is received before the UiRB register is read</p>
Other functions	<ul style="list-style-type: none"> <li>CLK polarity Rising or falling edge of the transmit/receive clock for output and input of transmit/receive data</li> <li>Bit order selection LSB first or MSB first</li> <li>Continuous receive mode Data reception is enabled by a read access to the UiRB register</li> <li>Serial data logic inversion (UART0 to UART6) This function logically inverts transmit/receive data</li> </ul>

Notes:

- When selecting an external clock, the following preconditions should be met:
  - The CLKi pin is held high when the CKPOL bit in the UiC0 register is set to 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge).
  - The CLKi pin is held low when the CKPOL bit is set to 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge).
- The UiRB register is undefined when an overrun error occurs. The IR bit in the SiRIC register does not change to 1 (interrupt requested).

Tables 18.3 and 18.4 list register settings. When UART<sub>i</sub> operating mode is selected, a high is output at the TXD<sub>i</sub> pin until transmission starts (the TXD<sub>i</sub> pin is high-impedance when the N-channel open drain output is selected) (i = 0 to 8).

Figures 18.20 and 18.21 show examples of transmit and receive operations in synchronous serial interface mode, respectively.

**Table 18.3 Register Settings in Synchronous Serial Interface Mode (for UART0 to UART6)**

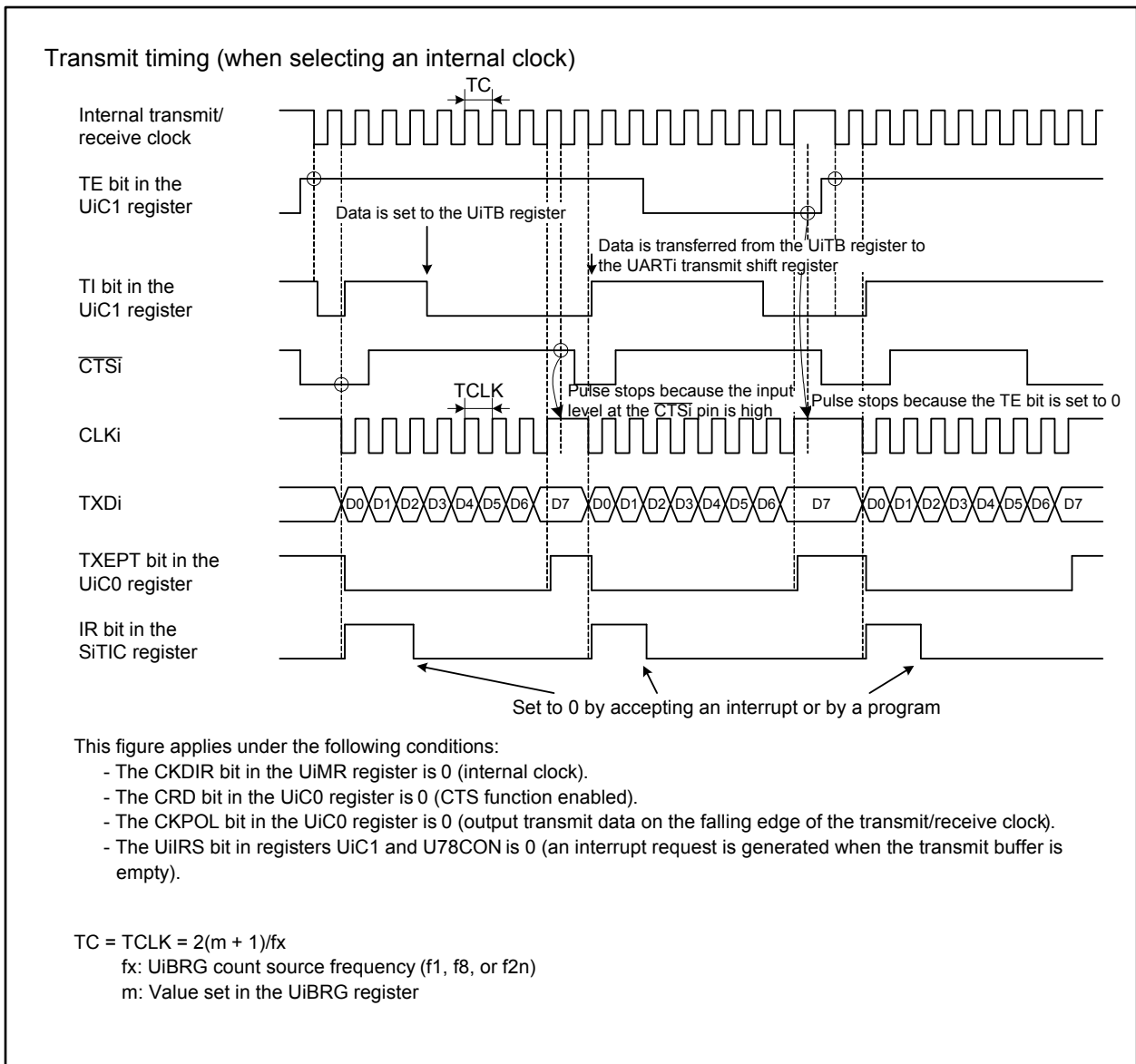
Register	Bits	Function
UiMR	(b7 to b4)	Set the bits to 0000b
	CKDIR	Select either an internal clock or external clock
	SMD2 to SMD0	Set the bits to 001b
UiC0	UFORM	Select either LSB first or MSB first
	CKPOL	Select a transmit/receive clock polarity
	NCH	Select an output mode of the TXD <sub>i</sub> pin
	CRD	Select CTS function enabled or disabled
	TXEPT	Transmit shift register empty flag
	(b2)	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	(b7)	Set the bit to 0
	UiLCH	Set the bit to 1 to use logic inversion
	UiRRM	Set the bit to 1 to use continuous receive mode
	UiIRS	Select a source for the UART <sub>i</sub> transmit interrupt
	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission/reception
UiSMR	—	Set the bits to 00h
UiSMR2	—	Set the bits to 00h
UiSMR3	(b7 to b4)	Set the bits to 0000b
	NODC	Select a clock output mode
	(b2 to b0)	Set the bits to 000b
UiSMR4	(b7 to b0)	Set the bits to 00h
UiBRG	(b7 to b0)	Set the divide ratio according to the bit rate
UiTB	(b7 to b0)	Set the data to be transmitted
UiRB	OER	Overrun error flag
	(b7 to b0)	Received data is read

i = 0 to 6

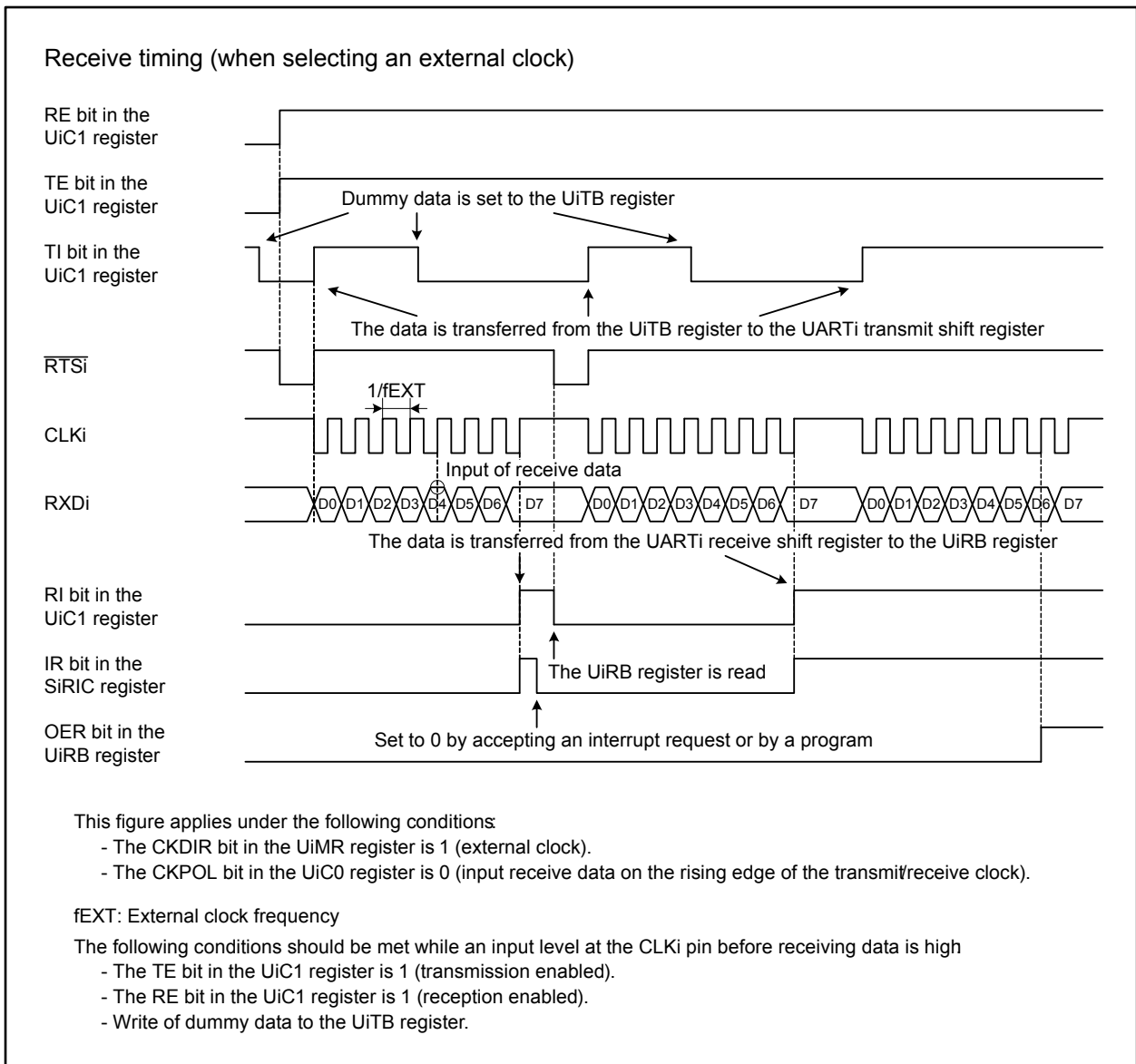
**Table 18.4 Register Settings in Synchronous Serial Interface Mode (for UART7 and UART8)**

Register	Bits	Function
UiMR	(b7 to b4)	Set the bits to 0000b
	CKDIR	Select an internal clock or external clock
	SMD2 to SMD0	Set the bits to 001b
UiC0	UFORM	Select either LSB first or MSB first
	CKPOL	Select a transmit/receive clock polarity
	(b5)	Set the bit to 0
	CRD	Select CTS function enabled or disabled
	TXEPT	Transmit shift register empty flag
	(b2)	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	(b7 to b4)	Set the bits to 0000b
	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission/reception
U78CON	(b7 to b4)	Set the bits to 0000b
	UiRRM	Set the bit to 1 to use continuous receive mode
	UiIRS	Select an interrupt source for UAR <i>T</i> <sub>i</sub> transmit
UiBRG	—	Set the divide ratio according to the bit rate
UiTB	(b7 to b0)	Set the data to be transmitted
UiRB	OER	Overrun error flag
	(b7 to b0)	Received data can be read

i = 7, 8



**Figure 18.20 Transmit Operation in Synchronous Serial Interface Mode**



**Figure 18.21 Receive Operation in Synchronous Serial Interface Mode**

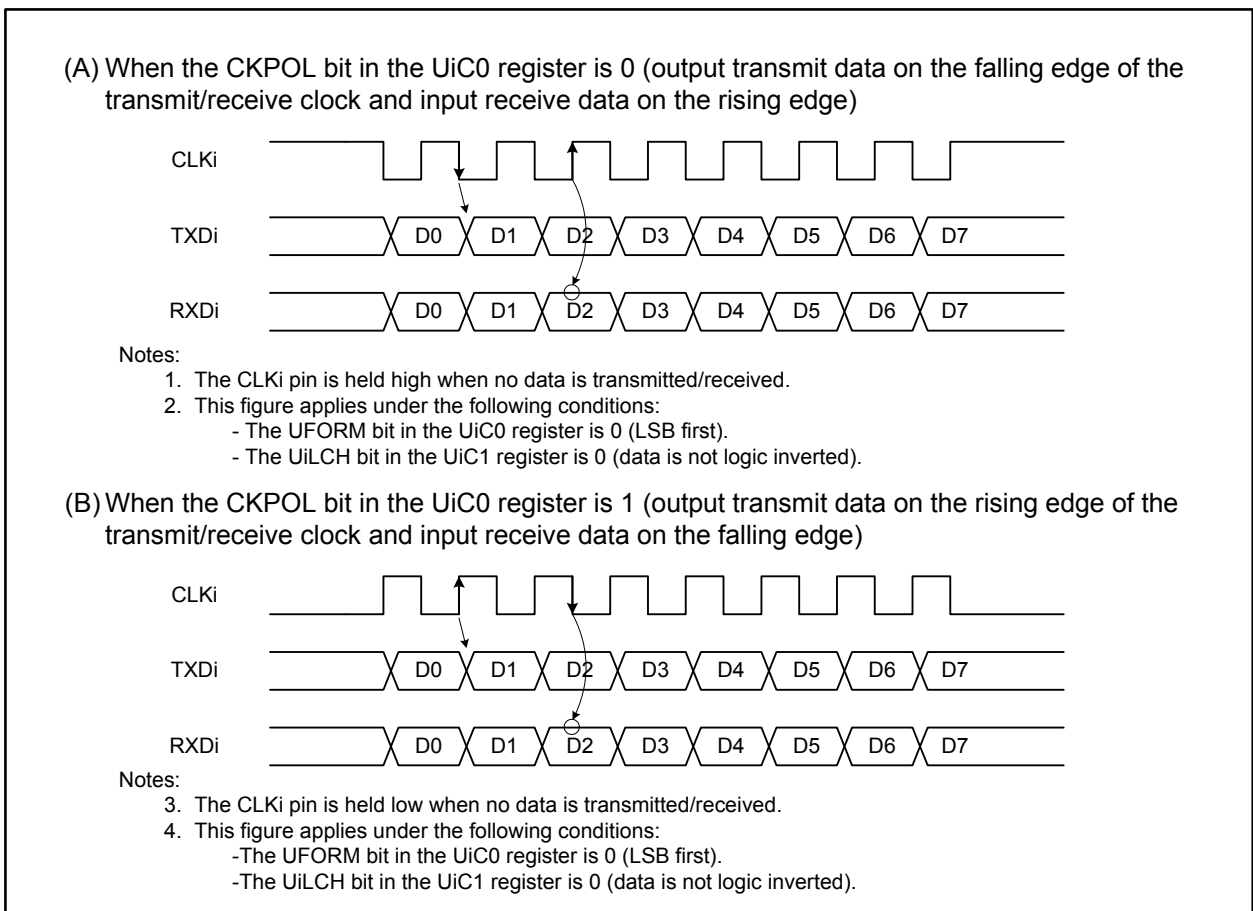
### 18.1.1 Reset Procedure on Transmit/Receive Error

When a transmit/receive error occurs in synchronous serial interface mode, follow the procedures below to perform a reset:

- (1) Set the TE bit to 0 (transmission disabled) and the RE bit to 0 (reception disabled) in the UiC1 register ( $i = 0$  to 8).
- (2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
- (3) Set bits SMD2 to SMD0 to 001b (synchronous serial interface mode).
- (4) Set the TE bit to 1 (transmission enabled) and the RE bit to 1 (reception enabled) in the UiC1 register.

### 18.1.2 CLK Polarity

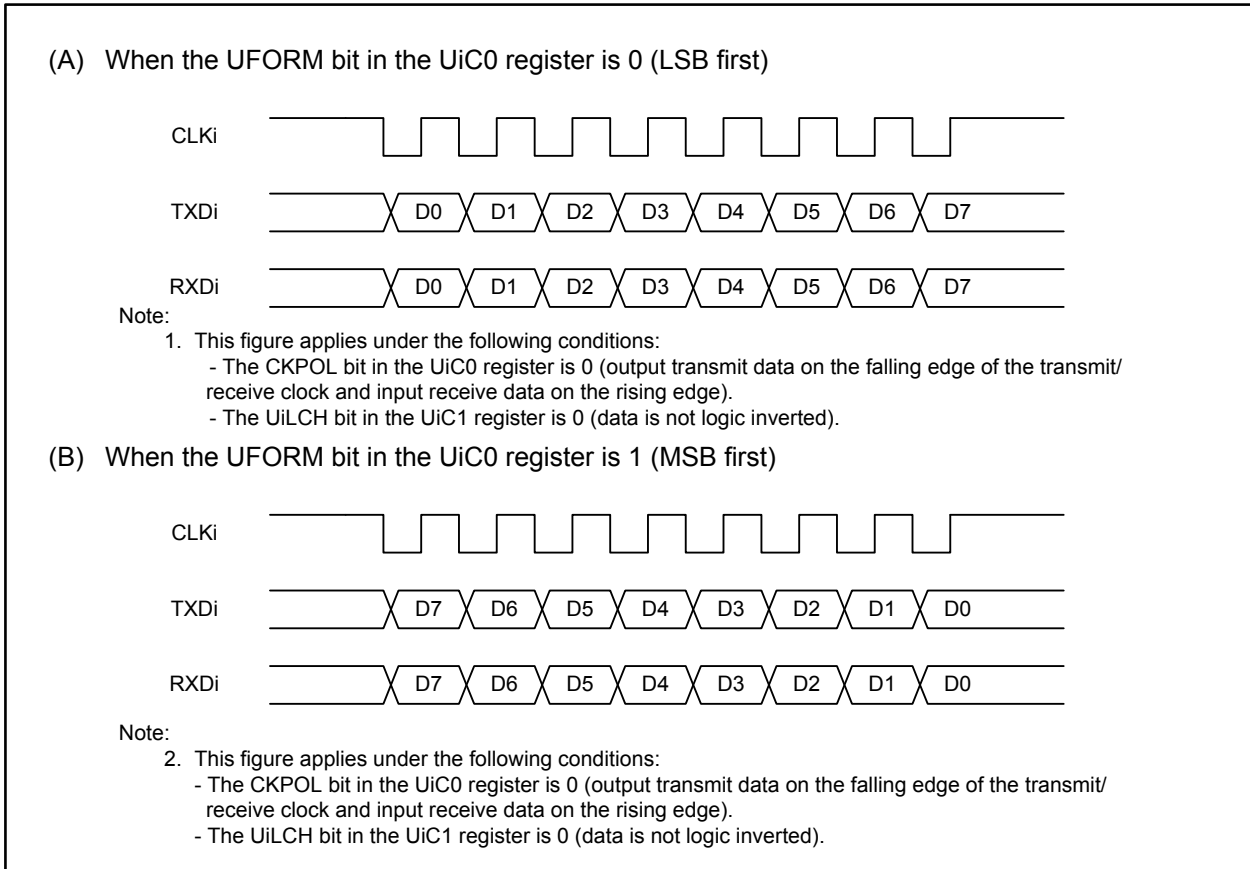
As shown in Figure 18.22, the polarity of the transmit/receive clock is selected using the CKPOL bit in the UiC0 register ( $i = 0$  to 8).



**Figure 18.22 Transmit/Receive Clock Polarity ( $i = 0$  to 8)**

### 18.1.3 LSB First and MSB First Selection

As shown in Figure 18.23, the bit order is selected by setting the UFORM bit in the UiC0 register ( $i = 0$  to 8).



**Figure 18.23 Bit Order ( $i = 0$  to 8)**

### 18.1.4 Continuous Receive Mode

In continuous receive mode, data reception is automatically enabled by a read access to the receive buffer register without writing dummy data to the transmit buffer register. To start data reception, however, dummy data is required to read the receive buffer register.

When the UiRRM bit in registers U0C1 to U6C1 and the U78CON register is set to 1 (continuous receive mode enabled), the TI bit in the UiC1 register becomes 0 (data held in the UiTB register) by a read access to the UiRB register ( $i = 0$  to 8). In this UiRRM bit setting, no dummy data should be written to the UiTB register.



### 18.1.5 Serial Data Logic Inversion

When the UiLCH bit in the UiC1 register is 1 (data is logic inverted), the logical value written in the UiTB register is inverted before being transmitted ( $i = 0$  to 6). The UiRB register is read as logic-inverted receive data. Figure 18.24 shows the logic inversion of serial data.

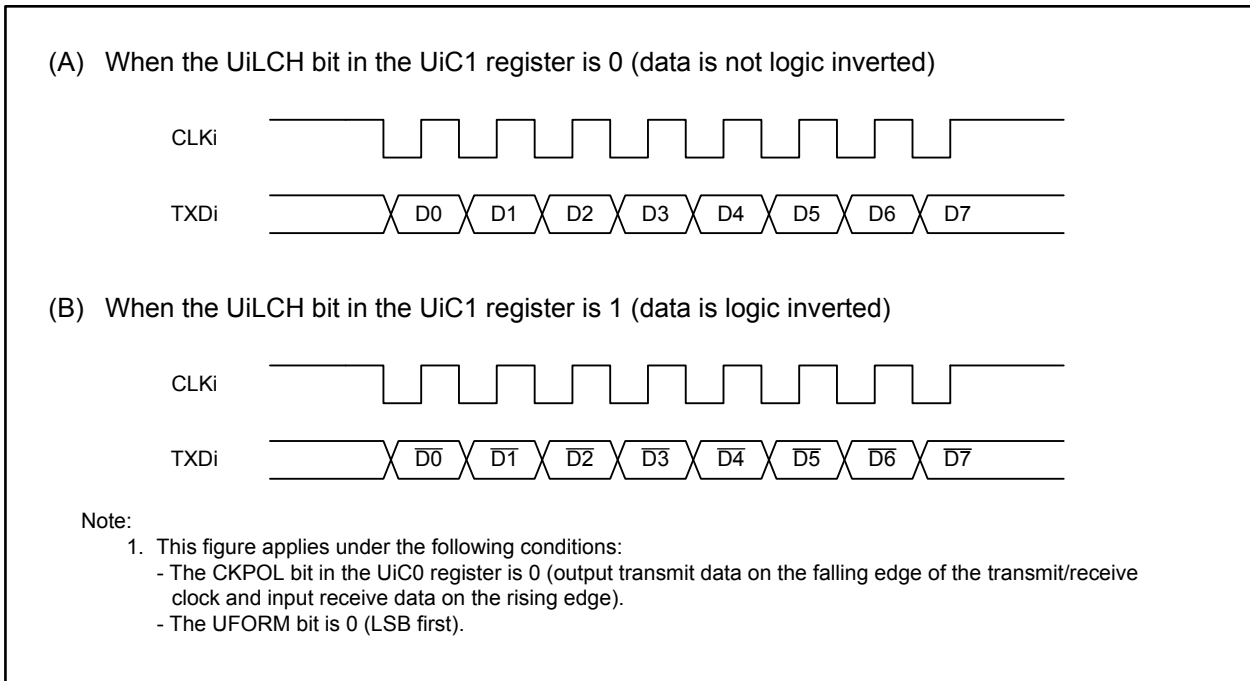


Figure 18.24 Serial Data Logic Inversion ( $i = 0$  to 6)

### 18.1.6 CTS/RTS Function

CTS function controls data transmission using the  $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$  pin ( $i = 0$  to 8). When an input level at the pin becomes low, data transmission starts. If the input level changes to high during transmission, the transmission of the next data is stopped after the last bit is transmitted.

In synchronous serial interface mode, the transmitter is required to operate even during the receive operation. If CTS function is enabled, the input level at the  $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$  pin should be low to start data reception as well.

RTS function indicates receiver status using the  $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$  pin. When data reception is ready, the output level at the pin becomes low. It becomes high on the first falling edge of the CLKi pin.

## 18.2 Asynchronous Serial Interface Mode (UART Mode)

The UART mode enables data transmission/reception synchronized with an internal clock generated by a trigger on the falling edge of the start bit. Table 18.5 lists specifications of UART mode.

**Table 18.5 UART Mode Specifications**

Item	Specification
Data format	<ul style="list-style-type: none"> <li>Start bit: 1-bit</li> <li>Data bit (data character): 7-bit, 8-bit, or 9-bit</li> <li>Parity bit: odd, even, or none</li> <li>Stop bit: 1-bit or 2-bit</li> </ul>
Transmit/receive clock	<ul style="list-style-type: none"> <li>The CKDIR bit in the UiMR register is 0 (internal clock) (<math>i = 0</math> to 8):  <math display="block">\frac{f_x}{16(m+1)} \quad f_x = f1, f8, f2n; m: \text{UiBRG register setting value, } 00\text{h to } FF\text{h}</math> </li> <li>The CKDIR bit is 1 (external clock)  <math display="block">\frac{f_{EXT}}{16(m+1)} \quad f_{EXT}: \text{Clock applied to the CLKi pin}</math> </li> </ul>
Transmit/receive control	CTS function enabled, RTS function enabled, or CTS/RTS function disabled
Transmit start conditions	<p>The conditions for starting data transmission are as follows:</p> <ul style="list-style-type: none"> <li>The TE bit in the UiC1 register is 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register is 0 (data held in the UiTB register)</li> <li>Input level at the CTS<math>\bar{i}</math> pin is low when CTS function is selected</li> </ul>
Receive start conditions	<p>The conditions for starting data reception are as follows:</p> <ul style="list-style-type: none"> <li>The RE bit in the UiC1 register is 1 (reception enabled)</li> <li>The start bit is detected</li> </ul>
Interrupt request generating timing	<p>In transmit interrupt, one of the following conditions can be selected by setting the UiIRS bit in registers U0C1 to U6C1 and the U78CON register:</p> <ul style="list-style-type: none"> <li>The UiIRS bit is 0 (transmit buffer is empty): when data is transferred from the UiTB register to the UART<math>i</math> transmit shift register (when the transmission has started)</li> <li>The UiIRS bit is 1 (transmission is completed): when data transmission from the UART<math>i</math> transmit shift register is completed</li> </ul> <p>In receive interrupt,</p> <ul style="list-style-type: none"> <li>When data is transferred from the UART<math>i</math> receive shift register to the UiRB register (when reception is completed)</li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Overrun error <sup>(1)</sup> This error occurs when 1 bit prior to the stop bit (when 1 stop bit length is selected) or the first stop bit (when 2 stop bit length is selected) of the next data is received before the UiRB register is read</li> <li>Framing error This error occurs when the required number of stop bits is not detected</li> <li>Parity error This error occurs when an even number of 1's in parity and character bits is detected while the odd number is set, or vice versa. The parity should be enabled</li> <li>Error sum flag This flag becomes 1 when any of overrun error, framing error, or parity error occurs</li> </ul>
Other functions	<ul style="list-style-type: none"> <li>Bit order selection LSB first or MSB first</li> <li>Serial data logic inversion This function logically inverts transmit/receive data. The start bit and stop bit are not inverted</li> <li>TXD/RXD I/O polarity switching The output level from the TXD pin and the input level to the RXD pin are inverted. All I/O levels are inverted</li> </ul>

Note:

- The UiRB register is undefined when an overrun error occurs. The IR bit in the SiRIC register does not change to 1 (interrupt requested).

Tables 18.6 and 18.7 list register settings. When UART<sub>i</sub> operating mode is selected, a high is output at the TXD<sub>i</sub> pin until transmission starts (the TXD<sub>i</sub> pin is high-impedance when the N-channel open drain output is selected) (i = 0 to 8). Figures 18.25 and 18.26 show examples of transmit operations in UART mode. Figure 18.27 shows an example of receive operation.

**Table 18.6 Register Settings in UART Mode (UART0 to UART6)**

Register	Bits	Function	
UiMR	IOPOL	Select I/O polarity of pins TXD and RXD	
	PRY and PRYE	Select parity enabled or disabled, and odd or even	
	STPS	Select a stop bit length	
	CKDIR	Select an internal clock or external clock	
	SMD2 to SMD0		Set the bits to 100b in 7-bit character length
			Set the bits to 101b in 8-bit character length
		Set the bits to 110b in 9-bit character length	
UiC0	UFORM	Select LSB first or MSB first in 8-bit character length. Set the bit to 0 in 7-bit or 9-bit character length	
	CKPOL	Set the bit to 0	
	NCH	Select an output mode for the TXD <sub>i</sub> pin	
	CRD	Select CTS function enabled or disabled	
	TXEPT	Transmit shift register empty flag	
	(b2)	Set the bit to 0	
	CLK1 and CLK0	Select a count source for the UiBRG register	
UiC1	(b7)	Set the bit to 0	
	UiLCH	Set the bit to 1 to use logic inversion	
	UiRRM	Set the bit to 0	
	UiIRS	Select an interrupt source for UART <sub>i</sub> transmission	
	RI	Receive complete flag	
	RE	Set the bit to 1 to enable data reception	
	TI	Transmit buffer empty flag	
	TE	Set the bit to 1 to enable data transmission	
UiSMR	—	Set the bits to 00h	
UiSMR2	—	Set the bits to 00h	
UiSMR3	—	Set the bits to 00h	
UiSMR4	—	Set the bits to 00h	
UiBRG	—	Set the divide ratio according to the bit rate	
UiTB	(b8 to b0)	Set the data to be transmitted <sup>(1)</sup>	
UiRB	OER, FER, PER, and SUM	Error flag	
	(b8 to b0)	Received data is read <sup>(1)</sup>	

i = 0 to 6

Note:

- The bits used are as follows: 7-bit character length: bits 6 to 0  
8-bit character length: bits 7 to 0  
9-bit character length: bits 8 to 0

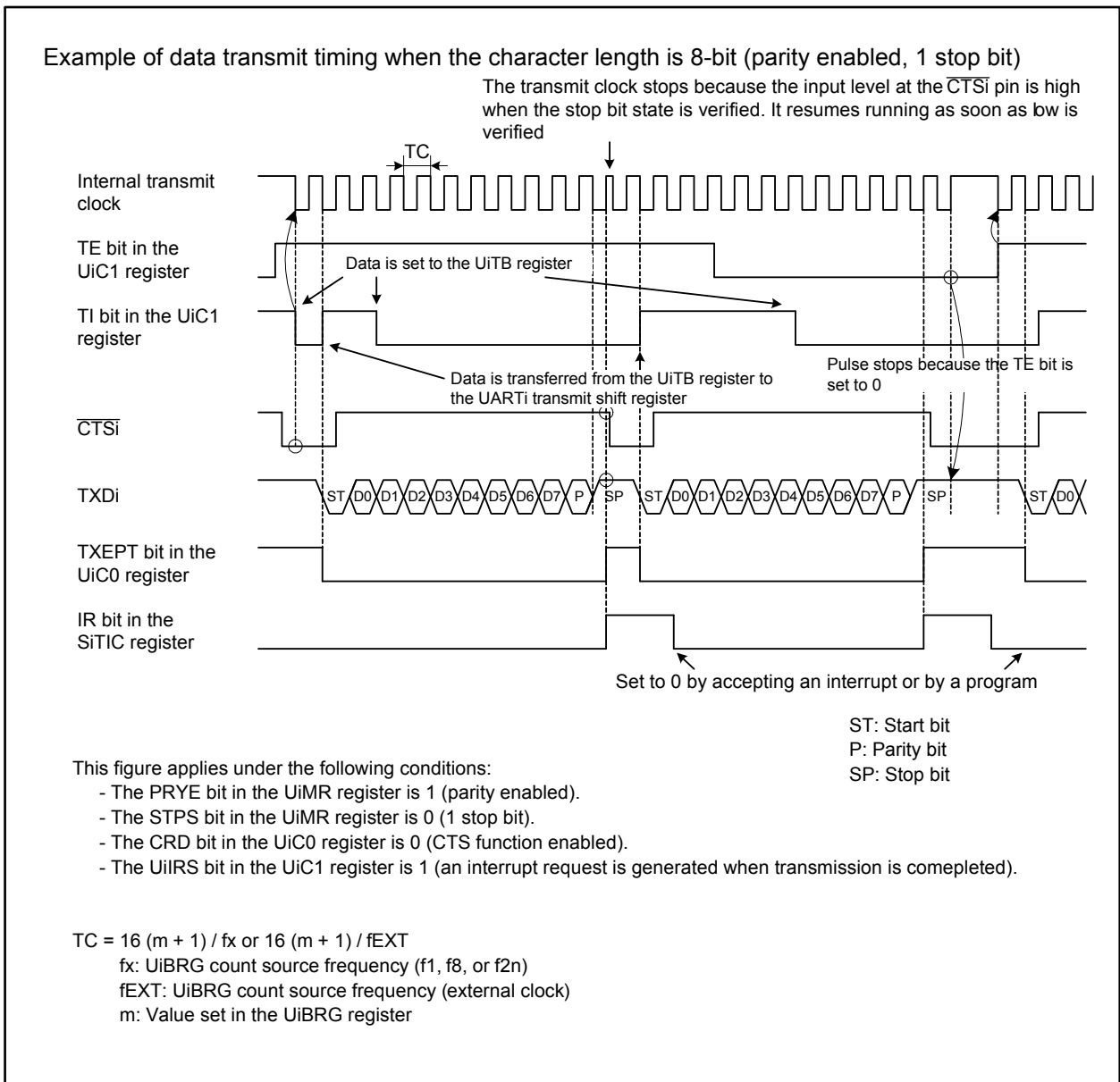
**Table 18.7 Register Settings in UART Mode (UART7, UART8)**

Register	Bits	Function	
UiMR	(b7)	Set the bit to 0	
	PRY and PRYE	Select parity enabled or disabled, and odd or even	
	STPS	Select a stop bit length	
	CKDIR	Select an internal clock or external clock	
	SMD2 to SMD0		Set the bits to 100b in 7-bit character length
			Set the bits to 101b in 8-bit character length
		Set the bits to 110b in 9-bit character length	
UiC0	UFORM	Select LSB first or MSB first in 8-bit character length. Set the bit to 0 in 7-bit or 9-bit character length	
	CKPOL	Set the bit to 0	
	(b5)	Set the bit to 0	
	CRD	Select CTS function enabled or disabled	
	TXEPT	Transmit shift register empty flag	
	(b2)	Set the bit to 0	
	CLK1 and CLK0	Select a count source for the UiBRG register	
UiC1	(b7 to b4)	Set the bits to 0000b	
	RI	Receive complete flag	
	RE	Set the bit to 1 to enable data reception	
	TI	Transmit buffer empty flag	
	TE	Set the bit to 1 to enable data transmission	
U78CON	(b7 to b4)	Set the bits to 0000b	
	UiRRM	Set the bit to 0	
	UiIRS	Select an interrupt source for UARTi transmission	
UiBRG	—	Set the divide ratio according to the bit rate	
UiTB	(b8 to b0)	Set the data to be transmitted <sup>(1)</sup>	
UiRB	OER, FER, PER, and SUM	Error flag	
	(b8 to b0)	Received data is read <sup>(1)</sup>	

i = 7, 8

Note:

- The bits used are as follows: 7-bit character length: bits 6 to 0  
8-bit character length: bits 7 to 0  
9-bit character length: bits 8 to 0



**Figure 18.25 Transmit Operation in UART Mode (1/2) (i = 0 to 8)**

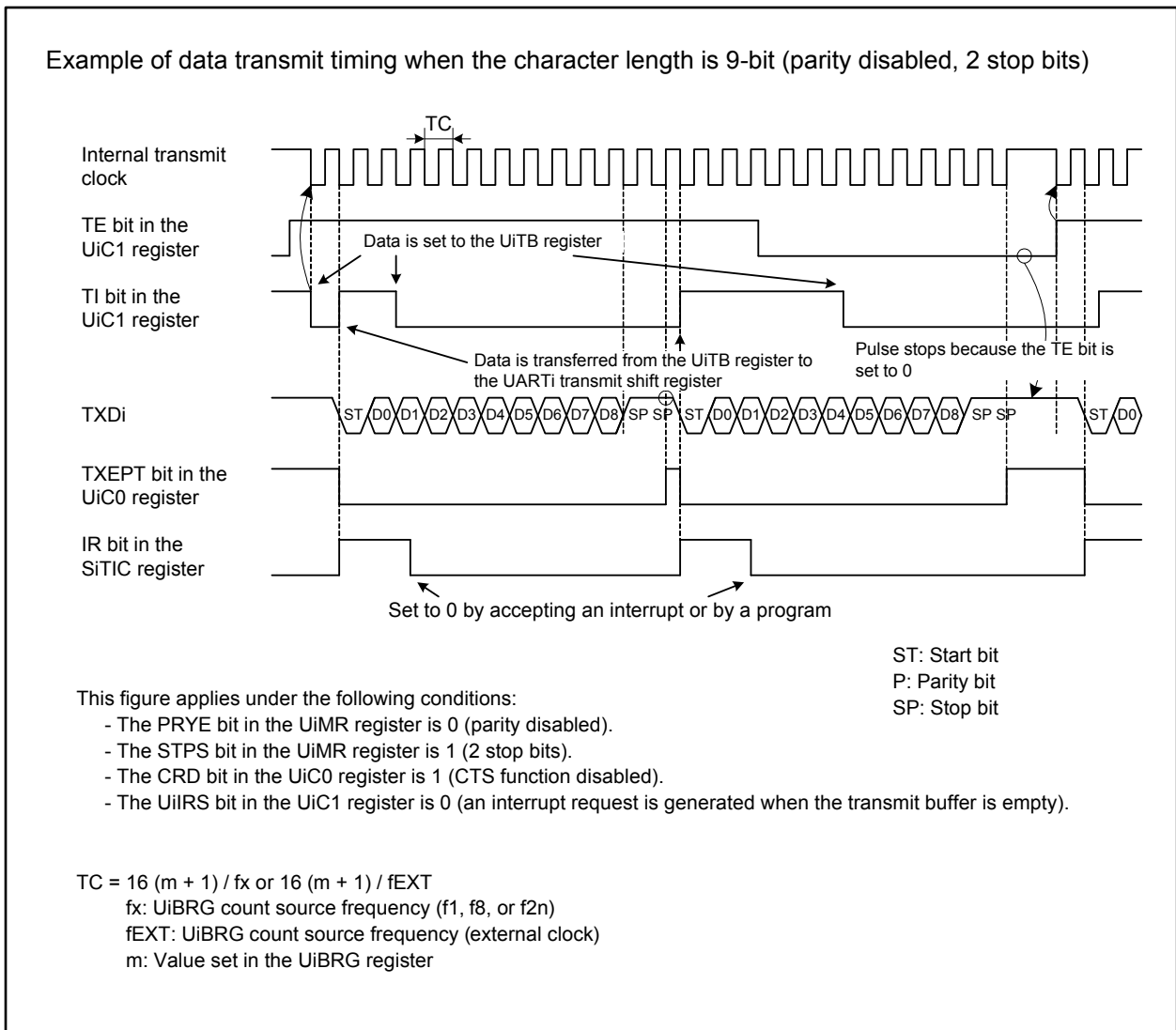


Figure 18.26 Transmit Operation in UART Mode (2/2) (i = 0 to 8)

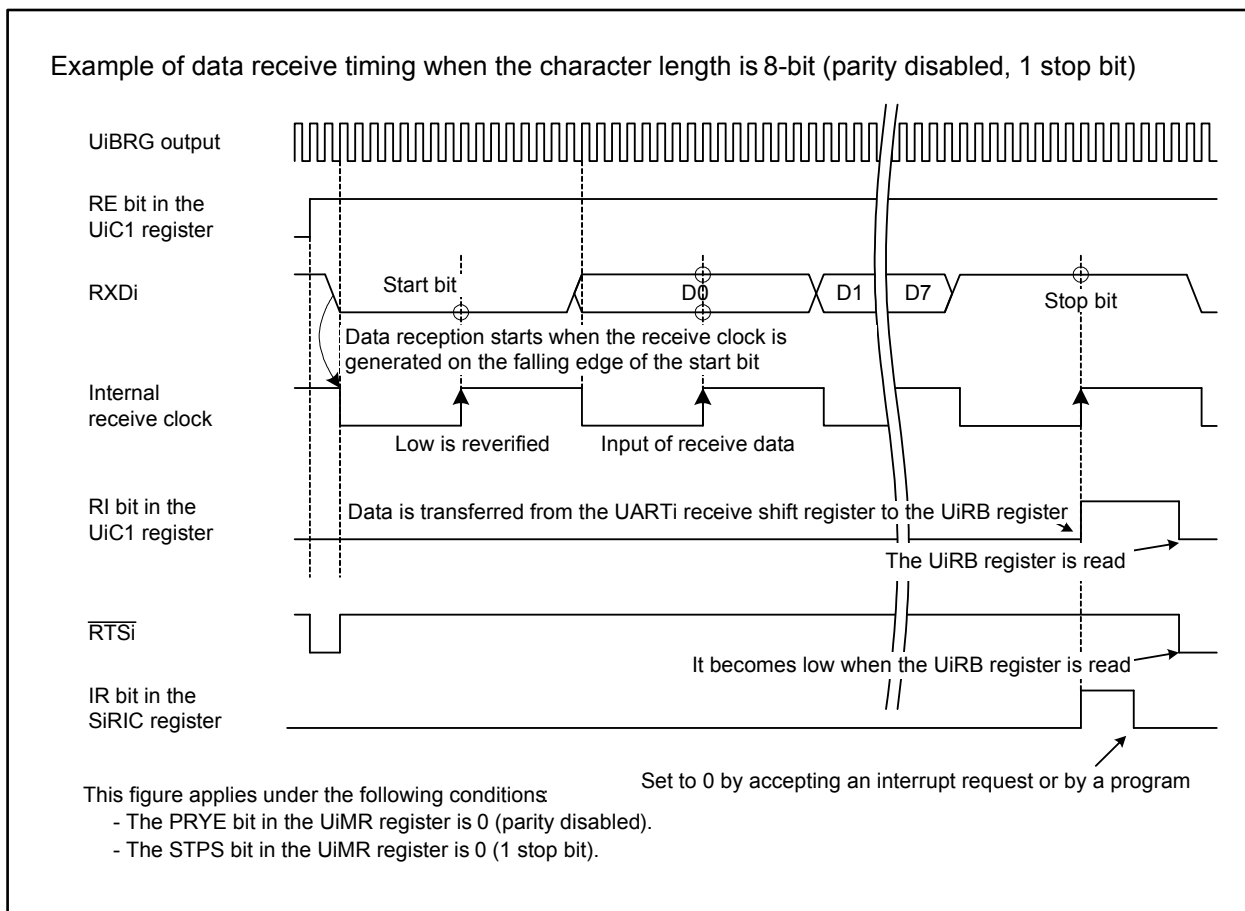


Figure 18.27 Receive Operation in UART Mode (i = 0 to 8)

### 18.2.1 Bit Rate

In UART mode, the bit rate is a clock frequency which is divided by a setting value of the UiBRG register and again divided by 16 (i = 0 to 8). Table 18.8 lists an example of bit rate setting.

Table 18.8 Bit Rate Setting

Bit Rate (bps)	Count Source of UiBRG	Peripheral Clock: 30 MHz		Peripheral Clock: 32 MHz	
		Setting value of UiBRG: n	Actual bit rate (bps)	Setting value of UiBRG: n	Actual bit rate (bps)
1200	f8	194 (C2h)	1202	207 (CHh)	1202
2400	f8	97 (61h)	2392	103 (67h)	2404
4800	f8	48 (30h)	4783	51 (33h)	4808
9600	f1	194 (C2h)	9615	207 (CFh)	9615
14400	f1	129 (81h)	14423	138 (8Ah)	14388
19200	f1	97 (61h)	19133	103 (67h)	19231
28800	f1	64 (40h)	28846	68 (44h)	28986
31250	f1	59 (3Bh)	31250	63 (3Fh)	31250
38400	f1	48 (30h)	38265	51 (33h)	38462
51200	f1	36 (24h)	50676	38 (26h)	51282

### 18.2.2 Reset Procedure on Transmit/Receive Error

When a transmit/receive error occurs in UART mode, follow the procedure below to perform a reset:

- (1) Set the TE bit to 0 (transmission disabled) and the RE bit to 0 (reception disabled) in the UiC1 register ( $i = 0$  to 8).
- (2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
- (3) Set again bits SMD2 to SMD0 to either of 001b, 101b, or 110b.
- (4) Set the TE bit to 1 (transmission enabled) and the RE bit to 1 (reception enabled) in the UiC1 register.

### 18.2.3 LSB First and MSB First Selection

As shown in Figure 18.28, the bit order is selected by setting the UFORM bit in the UiC0 register ( $i = 0$  to 8). This function is available when the character length is 8-bit.

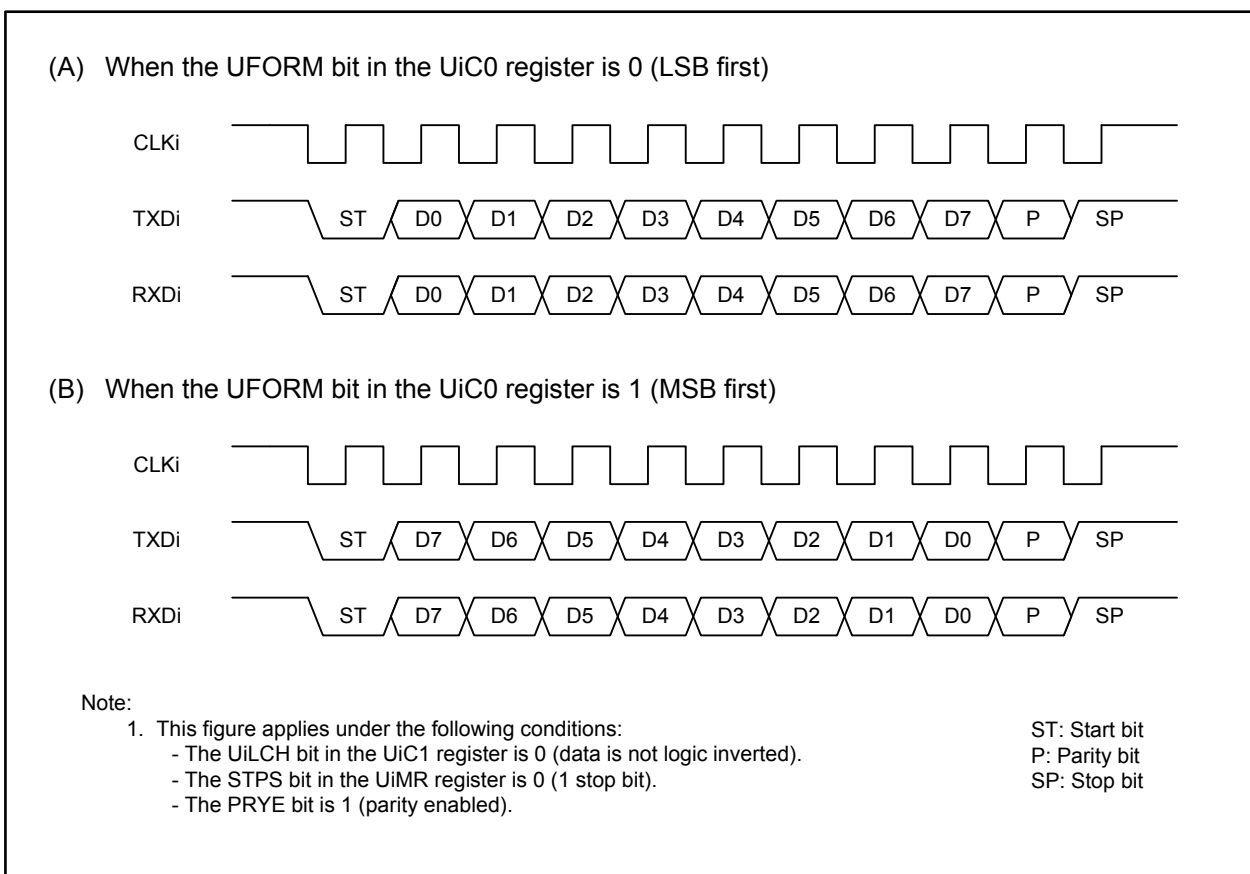


Figure 18.28 Bit Order ( $i = 0$  to 8)



### 18.2.4 Serial Data Logic Inversion

When the UiLCH bit in the UiC1 register is 1 (data is logic inverted), the logical value written in the UiTB register is inverted before being transmitted ( $i = 0$  to 6). The UiRB register is read as logic-inverted receive data. The parity bit is not inverted. Figure 18.29 shows the logic inversion of serial data.

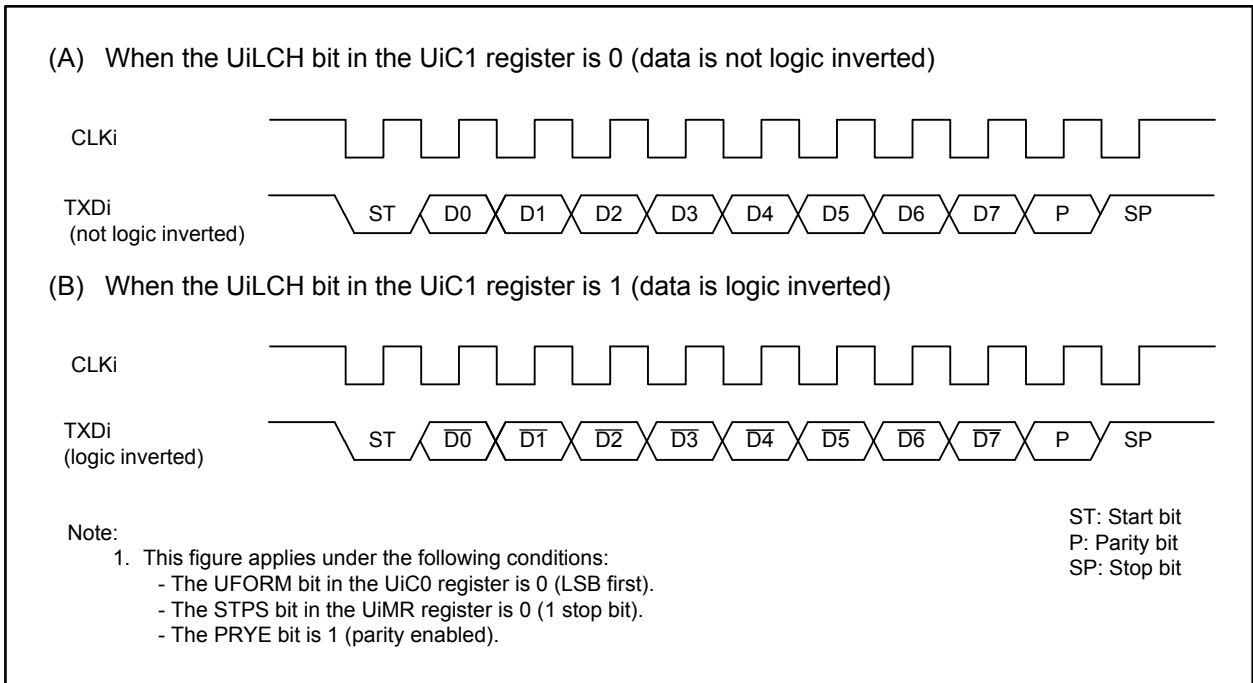


Figure 18.29 Serial Data Logic Inversion ( $i = 0$  to 6)

### 18.2.5 TXD and RXD I/O Polarity Inversion

The output level at the TXD pin and the input level at the RXD pin are inverted by this function. All I/O data levels, including the start bit, stop bit, and parity bit are inverted by setting the IOPOL bit in the UiMR register to 1 (inverted) ( $i = 0$  to 6). Figure 18.30 shows TXD and RXD I/O polarity inversion.

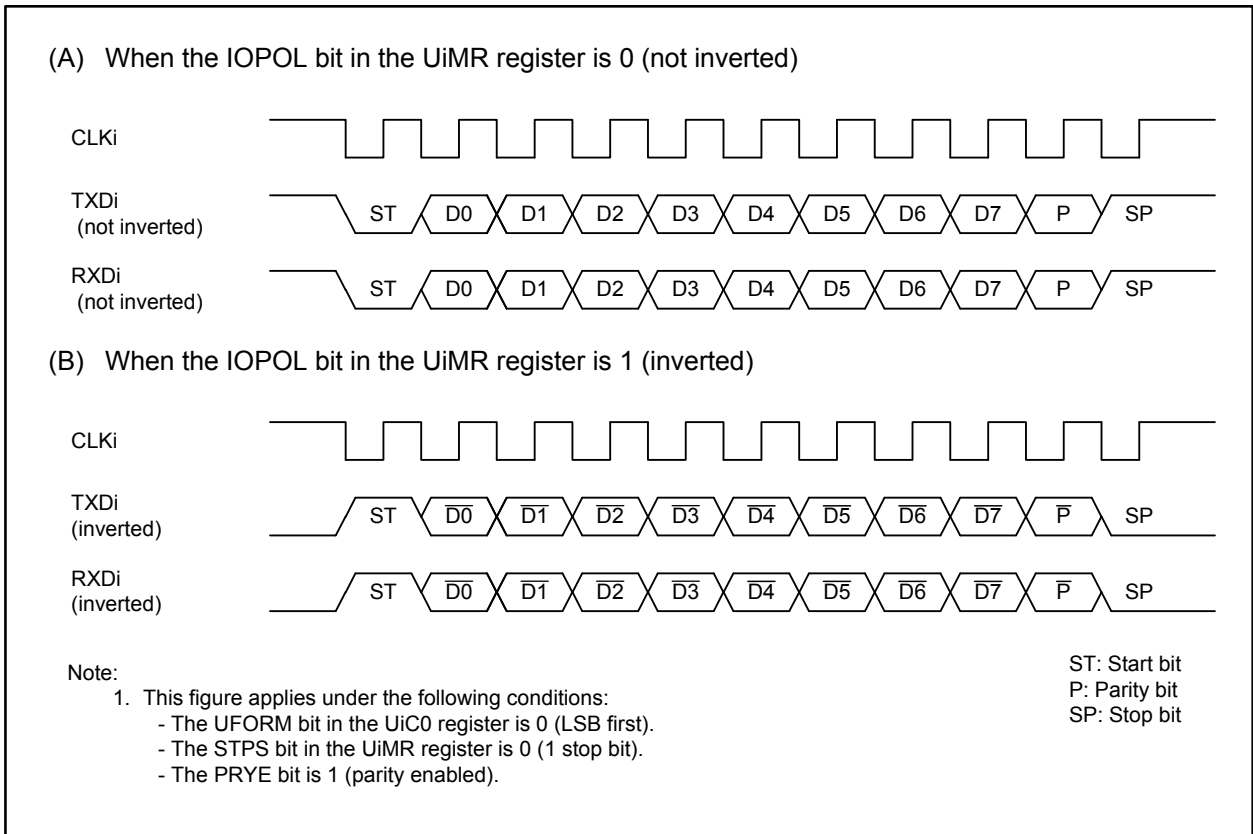


Figure 18.30 TXD and RXD I/O Polarity Inversion ( $i = 0$  to 6)

### 18.2.6 CTS/RTS Function

CTS function controls data transmission using the  $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$  pin ( $i = 0$  to 8). When an input level at the pin becomes low, data transmission starts. If the input level changes to high during transmit operation, transmission of the next data is stopped.

RTS function indicates receiver status using the  $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$  pin. When the MCU is ready to receive data, the output level at the pin becomes low. It becomes high on the first falling edge of the CLKi pin.

### 18.3 Special Mode 1 (I<sup>2</sup>C Mode)

This mode uses an I<sup>2</sup>C-typed interface for communication. Table 18.9 lists specifications of the I<sup>2</sup>C mode.

**Table 18.9 I<sup>2</sup>C Mode Specifications**

Item	Specification
Data format	8-bit character length
Transmit/receive clock	In master mode <ul style="list-style-type: none"> <li>The CKDIR bit in the UiMR register is 0 (internal clock) (i = 0 to 6):               <math display="block">\frac{fx}{2(m+1)} \quad fx = f1, f8, f2n</math> <p style="text-align: center;"><i>m</i>: UiBRG register setting value, 00h to FFh</p> </li> </ul> In slave mode <ul style="list-style-type: none"> <li>The CKDIR bit is 1 (external clock): input to the SCLi pin</li> </ul>
Transmit start conditions	The conditions for starting data transmission are as follows <sup>(1)</sup> : <ul style="list-style-type: none"> <li>The TE bit in the UiC1 register is 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register is 0 (data held in the UiTB register)</li> </ul>
Receive start conditions	The conditions for starting data reception are as follows <sup>(1)</sup> : <ul style="list-style-type: none"> <li>The RE bit in the UiC1 register is 1 (reception enabled)</li> <li>The TE bit in the UiC1 register is 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register is 0 (data held in the UiTB register)</li> </ul>
Interrupt request generating timing	<ul style="list-style-type: none"> <li>START condition is detected</li> <li>STOP condition is detected</li> <li>ACK (acknowledge) is detected, or reception is completed</li> <li>NACK (not-acknowledge) is detected, or transmission is completed</li> </ul>
Error detection	Overrun error <sup>(2)</sup> This error occurs when the eighth bit of the next data is received before the UiRB register is read
Other functions	<ul style="list-style-type: none"> <li>Arbitration lost Update timing of the ABT bit in the UiRB register can be selected</li> <li>SDAi digital delay No digital delay or two to eight cycles of digital delay of UiBRG count source</li> <li>Clock phase setting Clock delayed or no clock delay</li> </ul>

Notes:

- When an external clock is selected, the conditions should be met while the external clock signal is held high.
- The UiRB register is undefined when an overrun error occurs. The IR bit in the SiRIC register does not change to 1 (interrupt requested).

Table 18.10 lists register settings in I<sup>2</sup>C mode, and Tables 18.11 and 18.12 list I<sup>2</sup>C mode functions. Figure 18.31 shows a block diagram of I<sup>2</sup>C mode, and Figure 18.32 shows timings for the transfer to the UiRB register and the interrupt (i = 0 to 6).

As shown in Tables 18.11 and 18.12, UARTi enters this mode when bits SMD2 to SMD0 in the UiMR register are set to 010b, and the IICM bit in the UiSMR register is set to 1 (i = 0 to 6). Since a transmit signal at the SDAi pin is output via the delay circuit, it changes after the SCLi pin is stably held low.

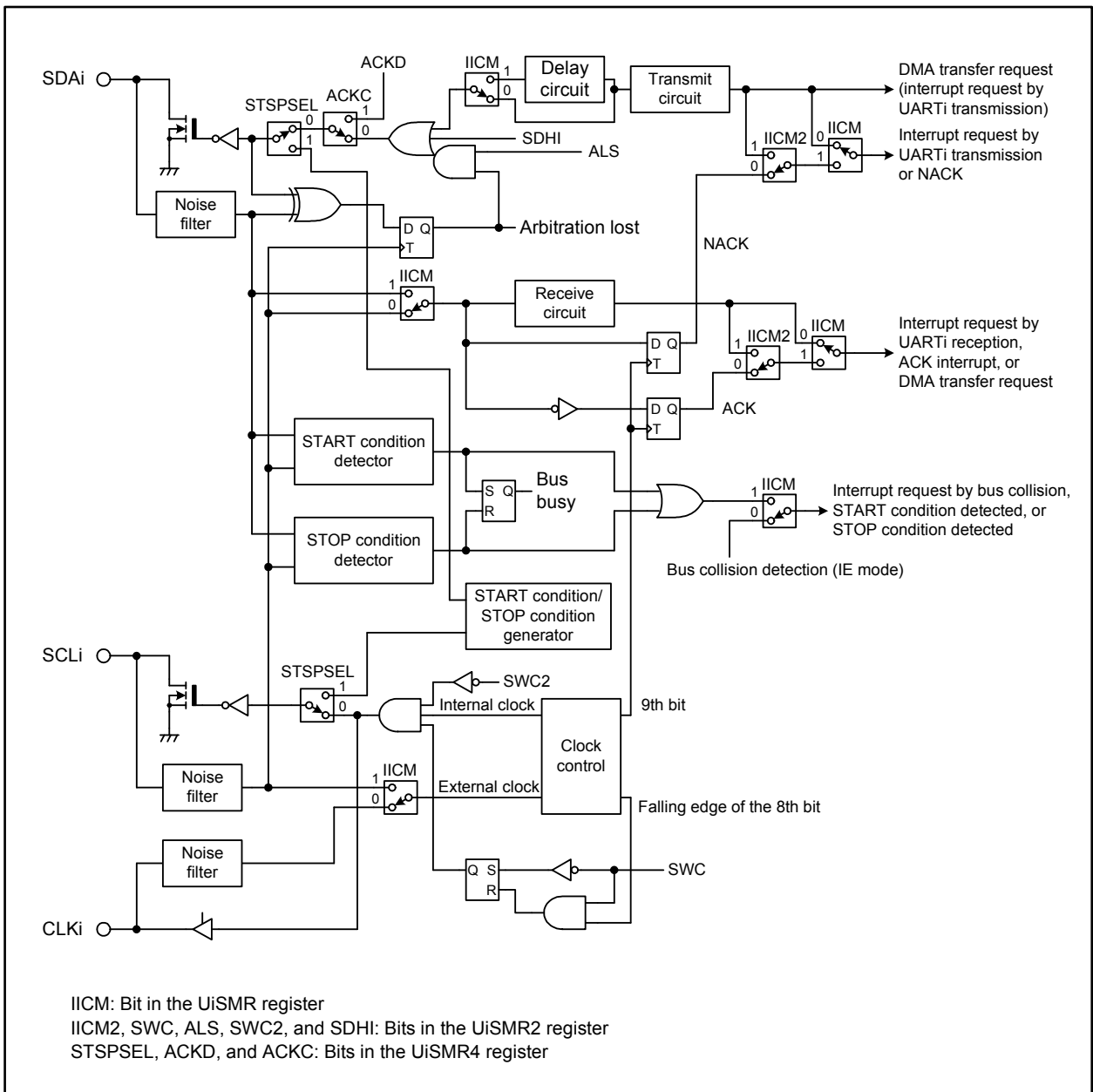


Figure 18.31 I<sup>2</sup>C Mode Block Diagram (i = 0 to 6)

**Table 18.10 Register Settings in I<sup>2</sup>C Mode (i = 0 to 6)**

Register	Bits	Function	
		Master	Slave
UiMR	IOPOL	Set the bit to 0	
	(b6 to b4)	Set the bits to 000b	
	CKDIR	Set the bit to 0	Set the bit to 1
	SMD2 to SMD0	Set the bit to 010b	
UiC0	(b7 to b4)	Set the bits to 1011b	
	TXEPT	Transmit shift register empty flag	
	(b2)	Set the bit to 0	
	CLK1 and CLK0	Select a count source for the UiBRG register	Disabled
UiC1	(b7 to b5)	Set the bits to 000b	
	UIIRS	Set the bit to 1	
	RI	Receive complete flag	
	RE	Set the bit to 1 to enable data reception	
	TI	Transmit buffer empty flag	
	TE	Set the bit to 1 to enable data transmission/reception	
UiSMR	(b7 to b3)	Set the bits to 00000b	
	BBS	Bus busy flag	
	ABC	Select an arbitration lost detection timing	Disabled
	IICM	Set the bit to 1	
UiSMR2	(b7)	Set the bit to 0	
	SDHI	Set the bit to 1 to disable the SDA output	
	SWC2	Set the bit to 1 to hold the SCL output at a forcible low	
	STC	Set the bit to 0	Set the bit to 1 to reset UARTi by detecting the START condition
	ALS	Set the bit to 1 to stop the output at the SDAi pin to detect an arbitration lost	Set the bit to 0
	SWC	Set the bit to 1 to hold a low output at the SCLi pin after receiving the eighth bit of the clock	
	CSC	Set the bit to 1 to enable clock synchronization	Set the bit to 0
	IICM2	Refer to Tables 18.11 and 18.12	
UiSMR3	DL2 to DL0	Set the digital delay value of SDAi	
	(b4 to b2)	Set the bit to 000b	
	CKPH	Refer to Tables 18.11 and 18.12	
	SSE	Set the bit to 0	
UiSMR4	SWC9	Set the bit to 0	Set the bit to 1 to hold a low output at the SCLi pin after receiving the ninth bit of the clock
	SCLHI	Set the bit to 1 to stop the SCL output to detect STOP condition	Set the bit to 0
	ACKC	Set the bit to 1 for ACK data output	
	ACKD	Select ACK or NACK	
	STSPSEL	Set the bit to 1 when any condition is output	Set the bit to 0
	STPREQ	Set the bit to 1 to generate a STOP condition	Set the bit to 0
	RSTAREQ	Set the bit to 1 to generate a repeated START condition	Set the bit to 0
	STAREQ	Set the bit to 1 to generate a START condition	Set the bit to 0
UiBRG	—	Set the divide ratio according to the bit rate	Disabled
IFSR0	IFSR06 and IFSR07	Select a UART as interrupt source	
IFSR1	IFSR16	Select a UART as interrupt source	
UiTB	(b8)	Set the bit to 1 when transmitting. Set the bit to the value of the ACK bit when receiving	
	(b7 to b0)	Set the data to be transmitted when transmitting. Set the register to FFh when receiving	
UiRB	OER	Overrun error flag	
	ABT	Arbitration lost detection flag	Disabled
	(b8)	D0 is loaded immediately after a receive interrupt occurs. ACK or NACK is loaded after a transmit interrupt occurs	
	(b7 to b0)	D7 to D1 are read immediately after a receive interrupt occurs. D7 to D0 are read after a transmit interrupt occurs	

**Table 18.11 I<sup>2</sup>C Mode Functions (i = 0 to 6) (1/2)**

Function	Synchronous Serial Interface Mode (SMD2 to SMD0 are 001b, IICM is 0)	I <sup>2</sup> C Mode (SMD2 to SMD0 are 010b, IICM is 1)			
		IICM2 is 0 (ACK/NACK interrupt)		IICM2 is 1 (Transmit/receive interrupt)	
		CKPH is 0 (No clock delay)	CKPH is 1 (Clock delayed)	CKPH is 0 (No clock delay)	CKPH is 1 (Clock delayed)
Source of software interrupt numbers 6 and 39 to 41 <sup>(1)</sup> (refer to Figure 18.32)	—	START condition or STOP condition detection (refer to Table 18.13)			
Source of software interrupt numbers 2, 4, 17, 19, 33, 35, and 37 <sup>(1)</sup> (refer to Figure 18.32)	UARTi transmission: Transmission started or completed (selected using the UiIRS register)	NACK detection: Rising edge of the ninth bit of SCLi	UARTi transmission: Rising edge of the ninth bit of SCLi	UARTi transmission: Falling edge of the ninth bit of SCLi	
Source of software interrupt numbers 3, 5, 18, 20, 34, 36, and 38 <sup>(1)</sup> (refer to Figure 18.32)	UARTi reception: Receiving at eighth bit CKPOL is 0 (rising edge) CKPOL is 1 (falling edge)	ACK detection: Rising edge of the ninth bit of SCLi	UARTi reception: Falling edge of the eighth bit of SCLi		
Data transfer timing from the UARTi receive shift register to the UiRB register	CKPOL is 0 (rising edge) CKPOL is 1 (falling edge)	Rising edge of the ninth bit of SCLi	Falling edge of the eighth bit of SCLi	Falling edge of the eighth bit and rising edge of the ninth bit of SCLi	
UARTi transmit output delay	No delay	Delayed			
Pins P6_3, P6_7, P7_0, P7_3, P7_6, P9_2, P9_6, P11_0, P12_0, P15_0, and P15_4	TXDi output	SDAi I/O			
Pins P6_2, P6_6, P7_1, P7_5, P8_0, P9_1, P9_7, P11_2, P12_2, P15_2, and P15_5	RXDi input	SCLi I/O			
Pins P6_1, P6_5, P7_2, P7_4, P7_7, P9_0, P9_5, P11_1, P12_1, P15_1, and P15_6	Select CLKi input or output	— (Not used in I <sup>2</sup> C mode)			

Note:

1. Steps to change an interrupt source are as follows:
  - (1) Disable the interrupt of the corresponding software interrupt number.
  - (2) Change the source of interrupt.
  - (3) Set the IR bit of the corresponding software interrupt number to 0 (no interrupt requested).
  - (4) Set bits ILVL2 to ILVL0 of the corresponding software interrupt number.

**Table 18.12 I<sup>2</sup>C Mode Functions (i = 0 to 6) (2/2)**

Function	Synchronous Serial Interface Mode (SMD2 to SMD0 are 001b, IICM is 0)	I <sup>2</sup> C Mode (SMD2 to SMD0 are 010b, IICM is 1)			
		IICM2 is 0 (ACK/NACK interrupt)		IICM2 is 1 (Transmit/receive interrupt)	
		CKPH is 0 (No clock delay)	CKPH is 1 (Clock delayed)	CKPH is 0 (No clock delay)	CKPH is 1 (Clock delayed)
Read level at pins RXDi and SCLi	Readable irrespective of the port direction bit				
Default output value at the SDAi pin	—	High (Value set in the port Pi register if the I/O port is selected by output function select registers (i = 0 to 7))			
SCLi default and end values	—	High	Low	High	Low
DMA source (refer to Figure 18.32)	UARTi reception	ACK detection		UARTi reception: Falling edge of the eighth bit of SCLi	
Store received data	The first to eighth bits of received data are stored into bits 0 to 7 in the UiRB register	The first to eighth bits of received data are stored into bits 7 to 0 in the UiRB register		The first to seventh bits of received data are stored into bits 6 to 0 in the UiRB register and the eighth bit is stored into bit 8	Same as on the left column on the first data storing. <sup>(1)</sup> The first to eighth bits of received data are stored into 7 to 0 bits in the UiRB register and the ninth bit is stored into bit 8 on the second data storing <sup>(2)</sup>
Read received data	The UiRB register status is read as it is			Bits 6 to 0 in the UiRB register are read as bits 7 to 1 and bit 8 is read as bit 0	Same as on the left column on the first read. <sup>(1)</sup> The UiRB register status is read as it is on the second read <sup>(2)</sup>

**Notes:**

1. The first data transfer to the UiRB register starts on the rising edge of the eighth bit of SCLi.
2. The second data transfer to the UiRB register starts on the rising edge of the ninth bit of SCLi.

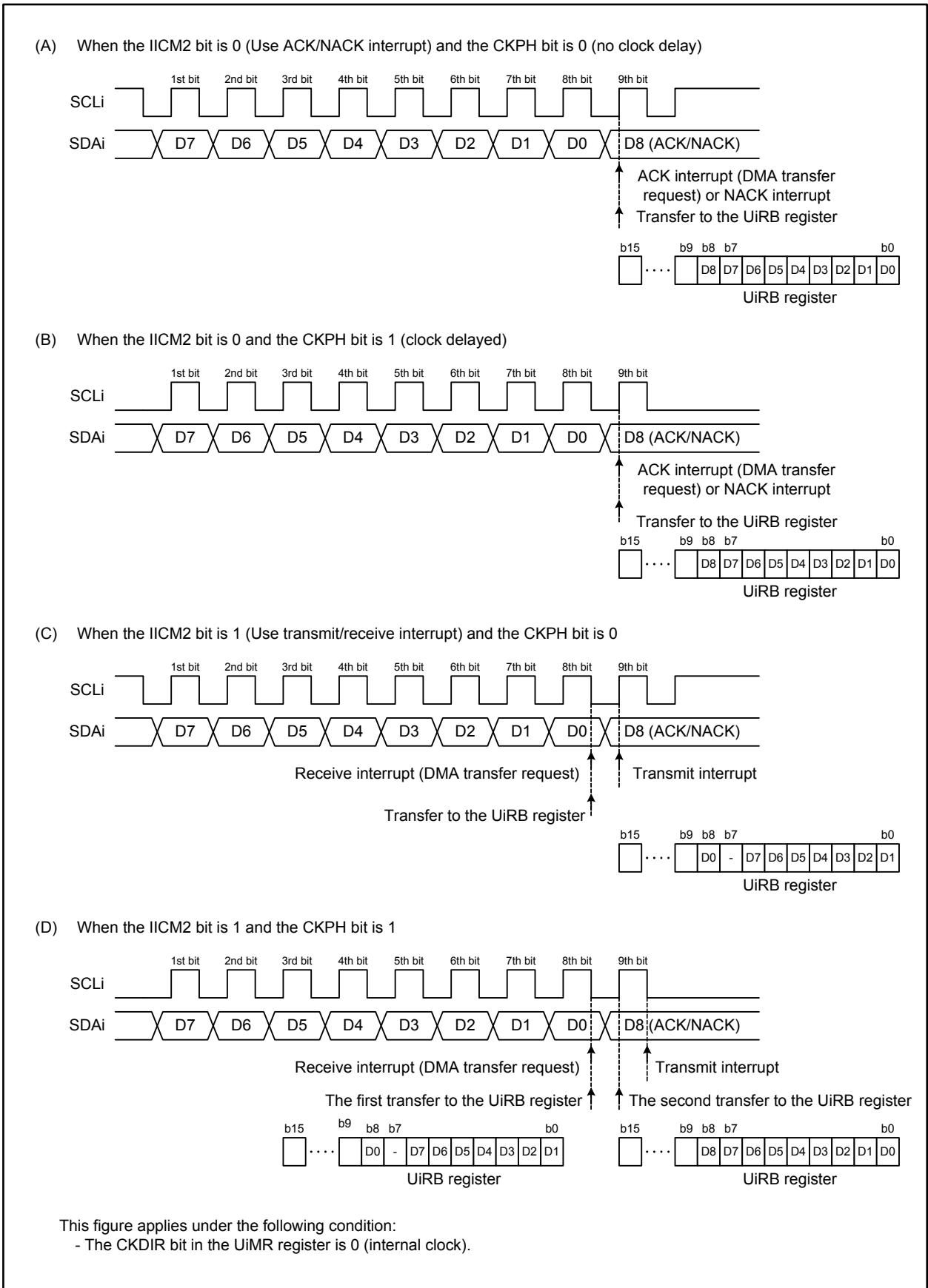


Figure 18.32 Timings for the Transfer and Interrupt to the UIRB Register (i = 0 to 6)



### 18.3.1 START Condition and STOP Condition Detection

The START condition and STOP condition are detected by their respective detectors.

The START condition detection interrupt request is generated by a high-to-low transition at the SDA<sub>i</sub> pin while the SCL<sub>i</sub> pin is held high ( $i = 0$  to 6). The STOP condition detection interrupt request is generated by a low-to-high transition at the SDA<sub>i</sub> pin while the SCL<sub>i</sub> pin is held high.

The START condition detection interrupt shares interrupt control registers and vectors with the STOP condition detection interrupt. The BBS bit in the UiSMR register determines which interrupt is requested.

To detect a START condition or STOP condition, both set-up and hold times require at least six cycles of the peripheral clock ( $f_1$ ) as shown in Figure 18.33. To meet the condition for the Fast-mode specification,  $f_1$  must be at least 10 MHz.

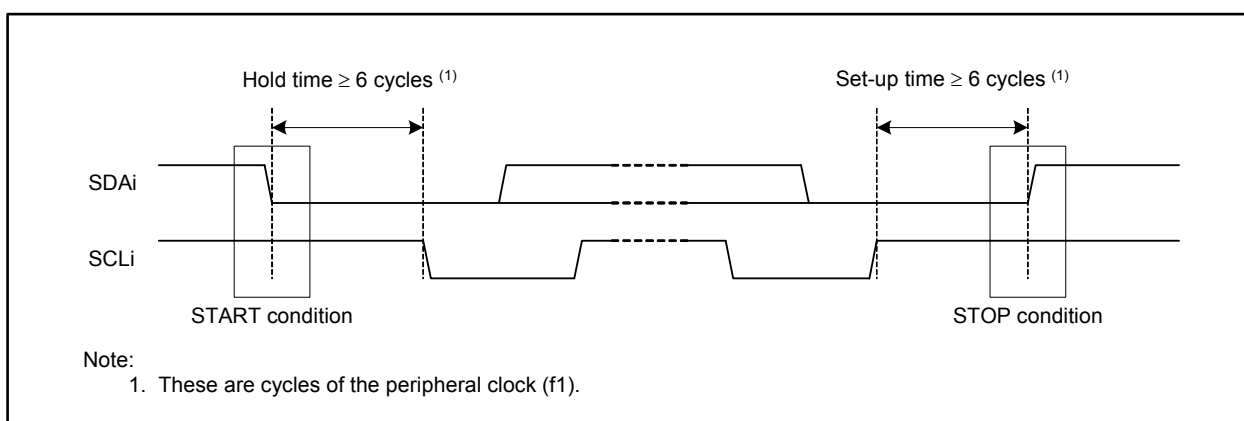


Figure 18.33 START Condition and STOP Condition Detection Timing ( $i = 0$  to 6)

### 18.3.2 START Condition and STOP Condition Generation

The START condition, repeated START condition, and STOP condition are generated by bits STAREQ, RSTAREQ, and STPREQ in the UiSMR4 register, respectively ( $i = 0$  to 6). To output a START condition, set the STSPSEL bit in the UiSMR4 register to 1 (select START condition/STOP condition generation circuit) after setting the STAREQ bit to 1 (start). To output a repeated START condition or STOP condition, set the STSPSEL bit to 1 after setting RSTAREQ bit or STPREQ bit to 1, respectively.

Table 18.13 and Figure 18.34 show the functions of the STSPSEL bit.

Table 18.13 STSPSEL Bit Functions

Function	STSPSEL is 0	STSPSEL is 1
START condition and STOP condition generation	Output is provided by the program with port (no auto generation by hardware)	START condition or STOP condition is output according to the STAREQ, RSTAREQ, or STPREQ bit
START condition and STOP condition interrupt request generating timing	When START condition or STOP condition is detected	When START condition or STOP condition generation is completed

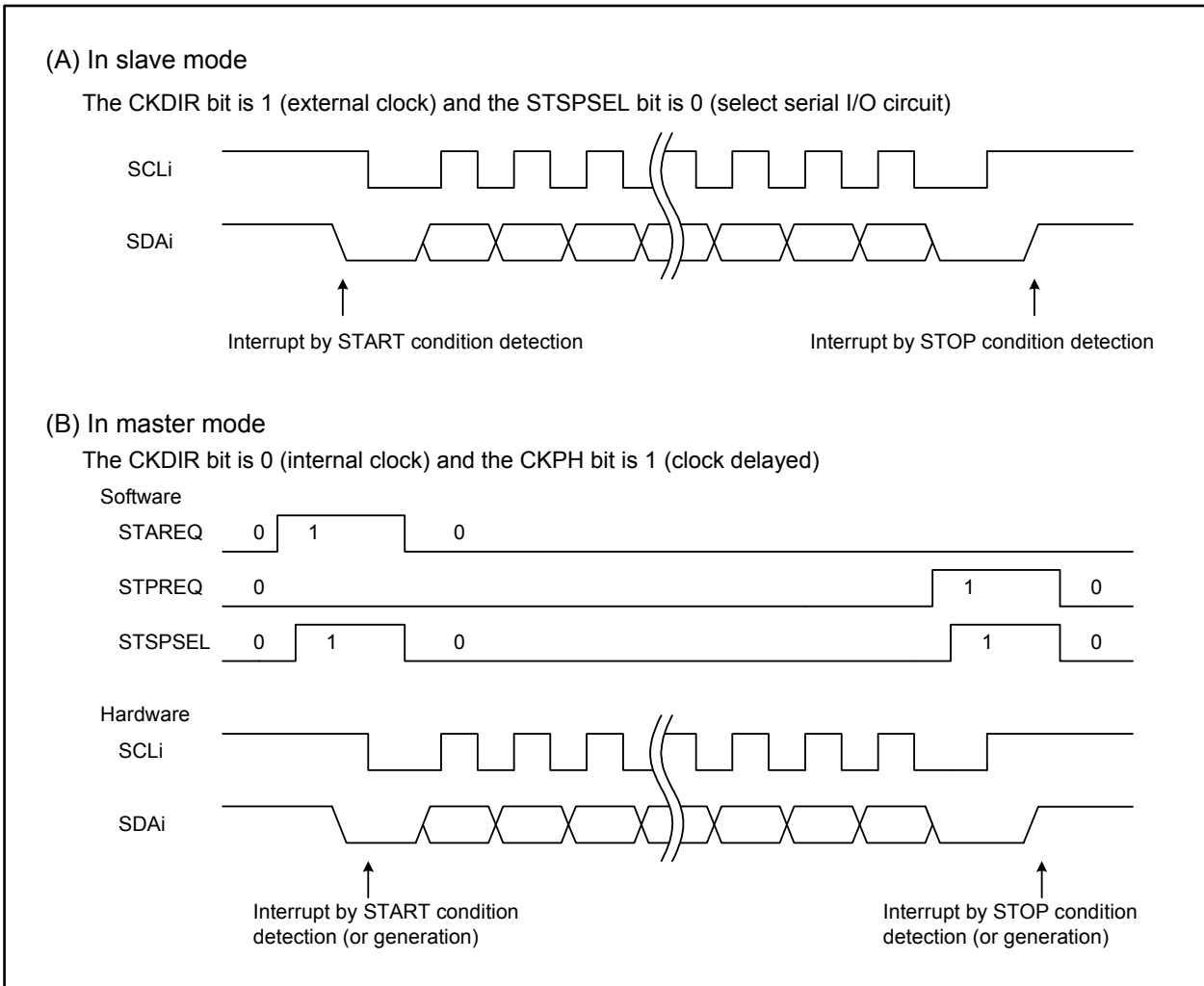


Figure 18.34 STSPSEL Bit Functions (i = 0 to 6)

### 18.3.3 Arbitration

On the rising edge of the SCLi, the MCU compares the transmit data with the data input from the SDAi pin. If no match is found, the MCU performs arbitration by stopping the SDAi output.

The update timing for the ABT bit in the UiRB register is selected by setting the ABC bit in the UiSMR register (i = 0 to 6).

When the ABC bit is 0 (update every bit), the ABT bit becomes 1 (detected (lose)) as soon as a data discrepancy is detected. If not detected, the ABT bit becomes 0 (not detected (win)). When the ABC bit is 1 (update every byte), the ABT bit becomes 1 on the falling edge of the eighth bit of the SCLi if any discrepancy is detected. In this ABC bit setting, set the ABT bit to 0 before starting the next 1-byte transfer.

When the ALS bit in the UiSMR2 register is 1 (stop the SDAi output), the SDAi pin becomes high-impedance as the ABT bit becomes 1 when an arbitration lost occurs.

### 18.3.4 SCL Control and Clock Synchronization

Data transmission/reception in I<sup>2</sup>C mode uses the transmit/receive clock as shown in Figure 18.32. The clock speed increase makes it difficult to secure the required time for ACK generation and data transmit procedure. I<sup>2</sup>C mode supports a function of wait-state insertion to secure this required time and a function of clock synchronization with a wait-state inserted by other devices.

The SWC bit in the UiSMR2 register is used to insert a wait-state for ACK generation ( $i = 0$  to 6). When the SWC bit is 1 (hold the SCLi pin low after the eighth bit is received), the SCLi pin is held low on the falling edge of the eighth bit of the SCLi. When the SWC bit is 0 (no wait-state/wait-state cleared), the SCLi line is released.

When the SWC2 bit in the UiSMR2 register is 1 (hold the SCLi pin low), the SCLi pin is forced low even during transmission or reception. When the SWC2 bit is 0 (output the transmit/receive clock at the SCLi pin), the SCLi line is released to output the transmit/receive clock.

The SWC9 bit in the UiSMR4 register is used to insert a wait-state for checking received acknowledge bits. While the CKPH bit in the UiSMR3 register is 1 (clock delayed), when the SWC9 bit is set to 1 (hold the SCLi pin low after the ninth bit is received), the SCLi pin is held low on the falling edge of the ninth bit of the SCLi. When the SWC9 bit is set to 0 (no wait-state/wait-state cleared), the SCLi line is released.

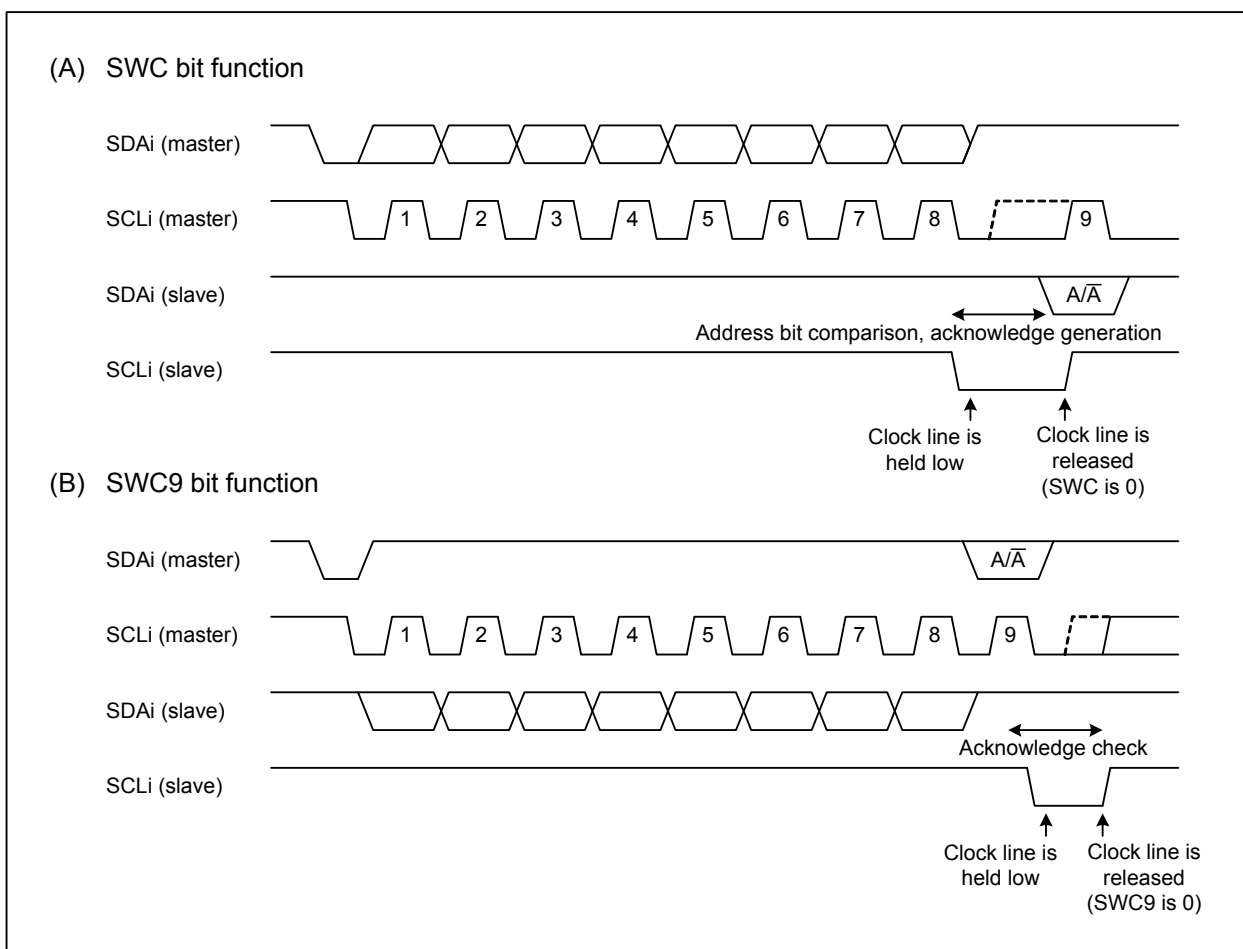
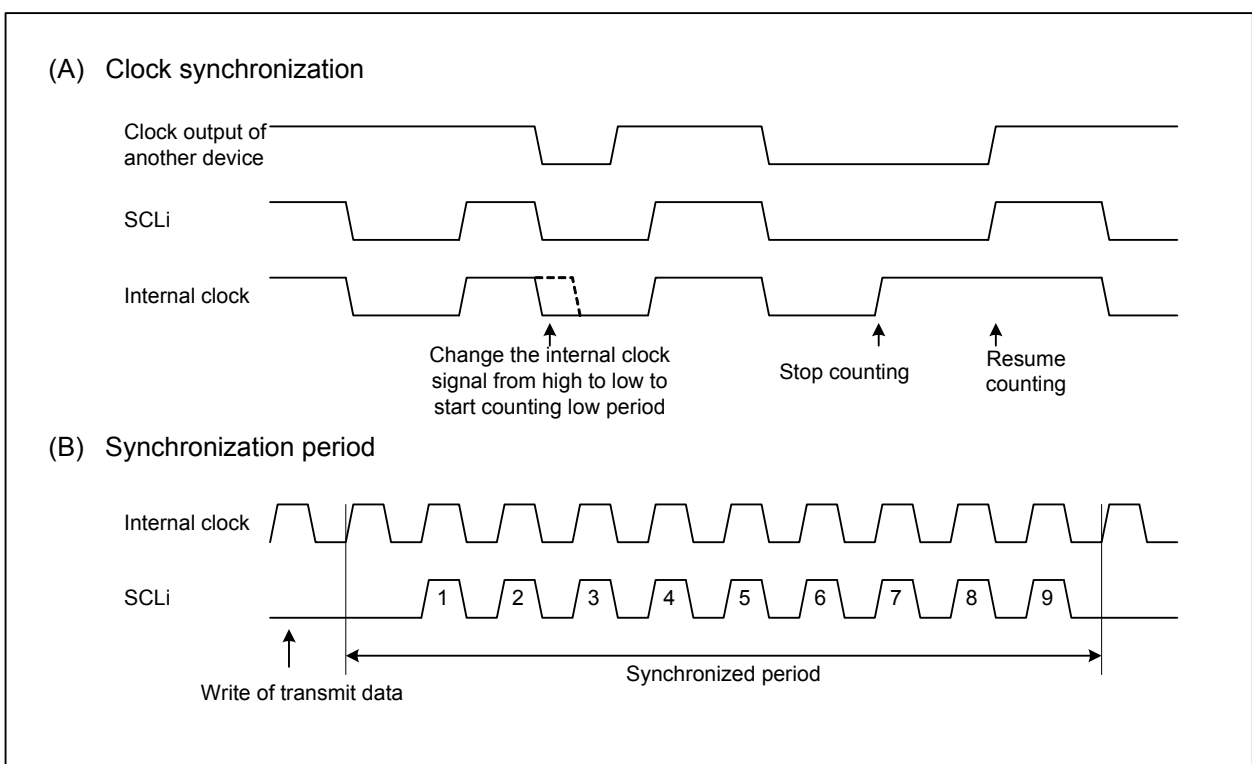


Figure 18.35 Wait-state Insertion Using the SWC or SWC9 Bit ( $i = 0$  to 6)

The CSC bit in the UiSMR2 register is used to synchronize an internally generated clock with the clock applied to the SCLi pin. For example, if a wait-state is inserted from another device, the two clocks are not synchronized. While the CSC bit is 1 (clock synchronization enabled) and the internal clock is held high, when a high at the SCLi pin changes to low, the internal clock becomes low in order to reload the value of the UiBRG register and to resume counting. While the SCLi pin is held low, when the internal clock changes from low to high, the count is stopped until the SCLi pin becomes high. That is, the UARTi transmit/receive clock is the logical AND of the internal clock and the SCLi. The synchronized period starts from one clock prior to the first synchronized clock and ends when the ninth clock is completed. The CSC bit can be set to 1 only when the CKDIR bit in the UiMR register is 0 (internal clock).

The SCLHI bit in the UiSMR4 register is used to leave the SCLi pin open when another master generates a STOP condition while the master is in transmit/receive operation. If the SCLHI bit is set to 1 (stop SCLi output), the SCLi pin is open (the pin is high-impedance) when a STOP condition is detected and the clock output is stopped.



**Figure 18.36 Clock Synchronization (i = 0 to 6)**

### 18.3.5 SDA Output

Values set to bits 8 to 0 (D8 to D0) in the UiTB register are output starting from D7 to D0, and lastly D8, which is a bit for the acknowledge signal ( $i = 0$  to 6). When transmitting, D8 should be set to 1 to free the bus. When receiving, D8 should be set to ACK or NACK.

Bits DL2 to DL0 in the UiSMR3 register set a delay time of the SDA<sub>i</sub> on the falling edge of the SCL<sub>i</sub>. Based on the UiBRG count source, the delay time can be selected from zero cycles (no delay) or two to eight cycles.

The SDA<sub>i</sub> pin can be high-impedance at any given time once the SDHI bit in the UiSMR2 register is set to 1 (stop the output). Output at the SDA<sub>i</sub> pin is low if an I/O port is selected for the SDA<sub>i</sub> and the pin is specified as the output port after selecting I<sup>2</sup>C mode. In this case, if the SDHI bit is 1, the SDA<sub>i</sub> pin becomes high-impedance.

When the SDHI bit is rewritten while the SCL<sub>i</sub> pin is held high, a START condition or STOP condition is generated. When it is rewritten immediately before the rising edge of SCL<sub>i</sub>, arbitration lost may be accidentally detected. Therefore, the SDHI bit should be rewritten so the SDA<sub>i</sub> pin level changes while the SCL<sub>i</sub> pin is low.

### 18.3.6 SDA Input

When the IICM2 bit in the UiSMR2 register is 0 (use ACK/NACK interrupt), the first 8 bits of received data (D7 to D0) are stored into bits 7 to 0 in the UiRB register and the ninth bit (ACK/NACK) is stored into bit 8 ( $i = 0$  to 6).

When the IICM2 bit is 1, the first 7 bits of received data (D7 to D1) are stored into bits 6 to 0 in the UiRB register and eighth bit (D0) is stored into bit 8.

If the IICM2 bit is 1 and the CKPH bit in the UiSMR3 register is 1 (clock delayed), the same data that is set when the IICM2 bit is 0 can be read. To read this data, read the UiRB register after data in the ninth bit is latched on the rising edge of the SCL<sub>i</sub>.

### 18.3.7 Acknowledge

When data is to be received in master mode, ACK is output after 8 bits are received by setting the UiTB register to 00FFh as dummy data. When the STSPSEL bit in the UiSMR4 register is 0 (select serial I/O circuit) and the ACKC bit is 1 (ACK data output), the value of the ACKD bit is output at the SDA<sub>i</sub> pin ( $i = 0$  to 6).

If the IICM2 bit is 0, a NACK interrupt request is generated when the SDA<sub>i</sub> pin is high on the rising edge of the ninth bit of the SCL<sub>i</sub>. An ACK interrupt request is generated when the SDA<sub>i</sub> pin is low.

When the DMA request source is "UART<sub>i</sub> receive interrupt request or ACK interrupt request", the DMA transfer starts when an ACK is detected.

### 18.3.8 Transmit/Receive Operation Reset

When the CKDIR bit in the UiMR register is 1 (external clock), the STC bit in the UiSMR2 register is 1 (initialize the circuit), and a START condition is detected, the following three operations are performed ( $i = 0$  to 6):

- The transmit shift register is reset and the UiTB register value is transferred to the transmit shift register. New data transmission starts on the falling edge of the first bit of the next SCL<sub>i</sub> as transmit clock. The transmit shift register value before the reset is output at the SDA<sub>i</sub> pin in the period from the falling edge of the SCL<sub>i</sub> until the first data output.
- The receive shift register is reset and the new data reception starts on the falling edge of the first bit of the next SCL<sub>i</sub>.
- The SWC bit in the UiSMR2 register becomes 1 (hold the SCL<sub>i</sub> pin low after the eighth bit is received).

The TI bit in the UiC1 register does not change when using this function to start the UART<sub>i</sub> transmission/reception.

## 18.4 Special Mode 2

Special mode 2 enables serial communication between one or multiple masters and multiple slaves. The  $\overline{SS}_i$  input pin controls serial bus communication ( $i = 0$  to  $6$ ). Table 18.14 lists specifications of special mode 2.

**Table 18.14 Special Mode 2 Specifications**

Item	Specification
Data format	8-bit character length
Transmit/receive clock	<ul style="list-style-type: none"> <li>The CKDIR bit in the UiMR register is set to 0 (internal clock) (<math>i = 0</math> to <math>6</math>):  <math display="block">\frac{f_x}{2^{(m+1)}} \quad f_x = f_1, f_8, f_{2n} \quad m: \text{UiBRG register setting value, 00h to FFh}</math> </li> <li>The CKDIR bit is set to 1 (external clock): input to the CLK<sub>i</sub> pin</li> </ul>
Transmit/receive control	Slave select function
Transmit start conditions	<p>The conditions for starting data transmission are as follows <sup>(1)</sup>:</p> <ul style="list-style-type: none"> <li>The TE bit in the UiC1 register is 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register is 0 (data held in the UiTB register)</li> </ul>
Receive start conditions	<p>The conditions for starting data reception are as follows <sup>(1)</sup>:</p> <ul style="list-style-type: none"> <li>The RE bit in the UiC1 register is 1 (reception enabled)</li> <li>The TE bit in the UiC1 register is 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register is 0 (data held in the UiTB register)</li> </ul>
Interrupt request generating timing	<p>In transmit interrupt, one of the following conditions can be selected by setting the UiIRS bit in registers U0C1 to U6C1:</p> <ul style="list-style-type: none"> <li>The UiIRS bit is 0 (transmit buffer is empty): when data is transferred from the UiTB register to the UART<sub>i</sub> transmit shift register (when the transmission has started)</li> <li>The UiIRS bit is 1 (transmission is completed): when data transmission from the UART<sub>i</sub> transmit shift register is completed</li> </ul> <p>In receive interrupt,</p> <ul style="list-style-type: none"> <li>When data is transferred from the UART<sub>i</sub> receive shift register to the UiRB register (when the reception is completed)</li> </ul>
Error detection	<p>Overflow error <sup>(2)</sup></p> <p>This error occurs when the seventh bit of the next data has been received before reading the UiRB register</p>
Other functions	<ul style="list-style-type: none"> <li>CLK polarity Rising or falling edge of the transmit/receive clock for transfer data input and output</li> <li>Bit order selection LSB first or MSB first</li> <li>Continuous receive mode Data reception is enabled by a read access to the UiRB register</li> <li>Serial data logic inversion This function logically inverts transmit/receive data</li> <li>Clock phase selection One of four combinations of transmit/receive clock polarity and phases</li> <li><math>\overline{SS}_i</math> input pin function Output pin can be high-impedance when the <math>\overline{SS}_i</math> pin is high</li> </ul>

Notes:

- When selecting an external clock, the following preconditions should be met:
  - The CLK<sub>i</sub> pin is held high when the CKPOL bit in the UiC0 register is 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge).
  - The CLK<sub>i</sub> pin is held low when the CKPOL bit is 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge).
- The UiRB register is undefined when an overrun error occurs. The IR bit in the SiRIC register does not change to 1 (interrupt requested).

Table 18.15 lists register settings in special mode 2.

**Table 18.15 Register Settings in Special Mode 2 (i = 0 to 6)**

Register	Bits	Function
UiMR	(b7 to b4)	Set the bits to 0000b
	CKDIR	Set the bit to 0 in master mode and set it to 1 in slave mode
	SMD2 to SMD0	Set the bits to 001b
UiC0	UFORM	Select either LSB first or MSB first
	CKPOL	Clock phase can be set by the combination of bits CKPOL and CKPH in the UiSMR3 register
	NCH	Select an output mode for the TXDi pin
	CRD	Set the bit to 1
	TXEPT	Transmit shift register empty flag
	(b2)	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	(b7 and b6)	Set the bits to 00b
	UiRRM	Set the bit to 1 to use continuous receive mode
	UiIRS	Select a source for UARTi transmit interrupt
	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission/reception
UiSMR	—	Set the bits to 00h
UiSMR2	—	Set the bits to 00h
UiSMR3	(b7 to b5)	Set the bits to 000b
	ERR	Mode fault flag
	NODC	Set the bit to 0
	DINC	Set to 0 in master mode and set to 1 in slave mode
	CKPH	Clock phase can be set by the combination of bits CKPH and CKPOL in the UiC0 register
	SSE	Set the bit to 1
UiSMR4	—	Set the bits to 00h
UiBRG	—	Set the divide ratio according to the bit rate
UiTB	(b7 to b0)	Set the data to be transmitted
UiRB	OER	Overrun error flag
	(b7 to b0)	Received data is read

### 18.4.1 $\overline{SSi}$ Input Pin Function (i = 0 to 6)

Special mode 2 is selected by setting the SSE bit in the UiSMR3 register to 1 (slave select function enabled). The  $\overline{CTSi}/\overline{RTSi}/\overline{SSi}$  pin functions as  $\overline{SSi}$  input.

The DINC bit in the UiSMR3 register determines which MCU performs as a master or slave.

When multiple MCUs perform as masters (multi-master system), the  $\overline{SSi}$  pin setting determines which master MCU is active and when.

#### 18.4.1.1 Slave Select Function in Slave Mode

When the DINC bit is 1 (slave mode) while input at the  $\overline{SSi}$  pin is high, the STXD<sub>i</sub> pin becomes high-impedance and the clock input at the CLK<sub>i</sub> pin is ignored. When input at the  $\overline{SSi}$  pin is low, the clock input is valid and serial data is output from the STXD<sub>i</sub> pin to enable serial communication.

#### 18.4.1.2 Slave Select Function in Master Mode

When the DINC bit is 0 (master mode) while input at the  $\overline{SSi}$  pin is high, which means there is the only one master MCU or no other master MCU is active, the MCU as master starts communication. The master provides the transmit/receive clock output at the CLK<sub>i</sub> pin. When input at the  $\overline{SSi}$  pin is low, which means that there are more masters, pins TXD<sub>i</sub> and CLK<sub>i</sub> become high-impedance. This error is called a mode fault. It can be verified using the ERR bit in the UiSMR3 register. The ongoing data transmission/reception does not stop even if a mode fault occurs. To stop transmission/reception, bits SMD2 to SMD0 in the UiMR register should be set to 000b (serial interface disabled).

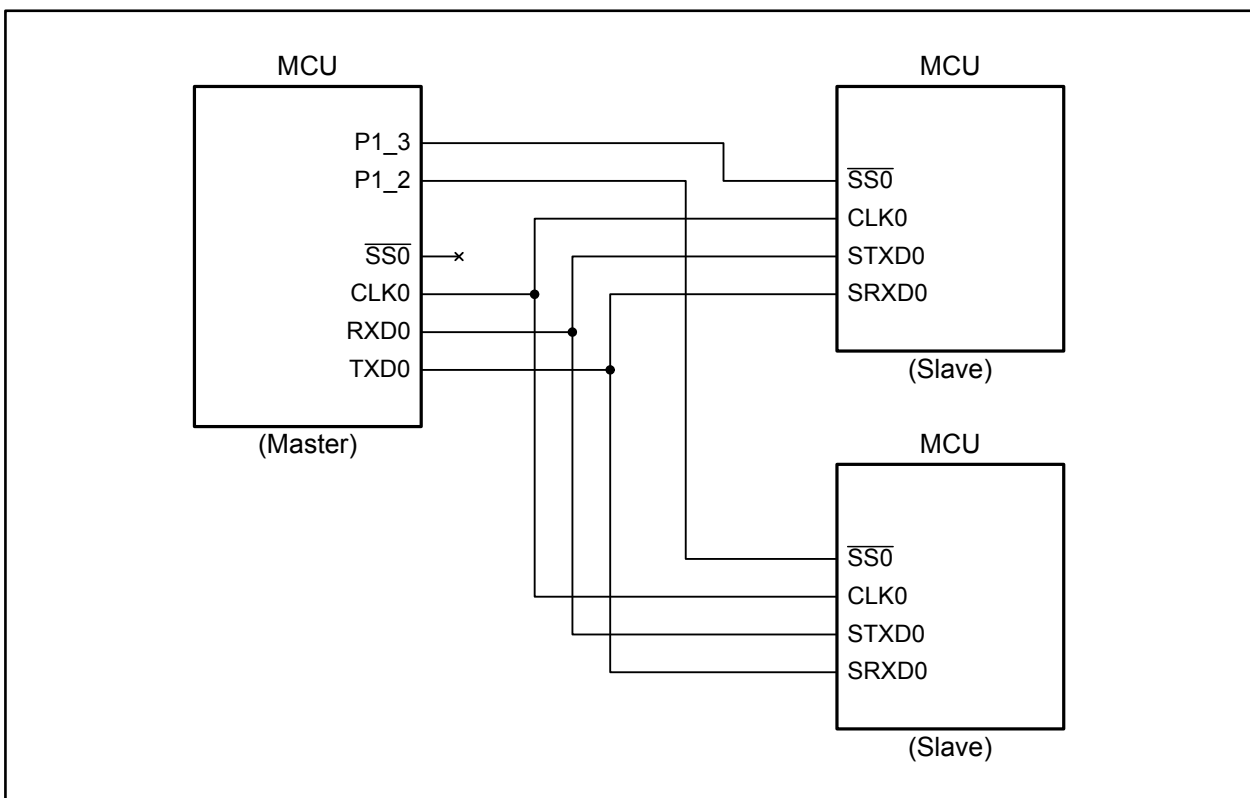


Figure 18.37 Serial Bus Communication Control with the  $\overline{SSi}$  Pin



## 18.4.2 Clock Phase Setting

The CKPH bit in the UiSMR3 register and the CKPOL bit in the UiC0 register select one of four combinations of transmit/receive clock polarity and serial clock phase ( $i = 0$  to 6).

The transmit/receive clock phase and polarity should be identical for the master device and the communicating slave device.

### 18.4.2.1 Transmit/Receive Timing in Master Mode

When the DINC bit is 0 (master mode), the CKDIR bit in the UiMR register should be set to 0 (internal clock) to generate the clock. Figure 18.38 shows transmit/receive timing of each clock phase.

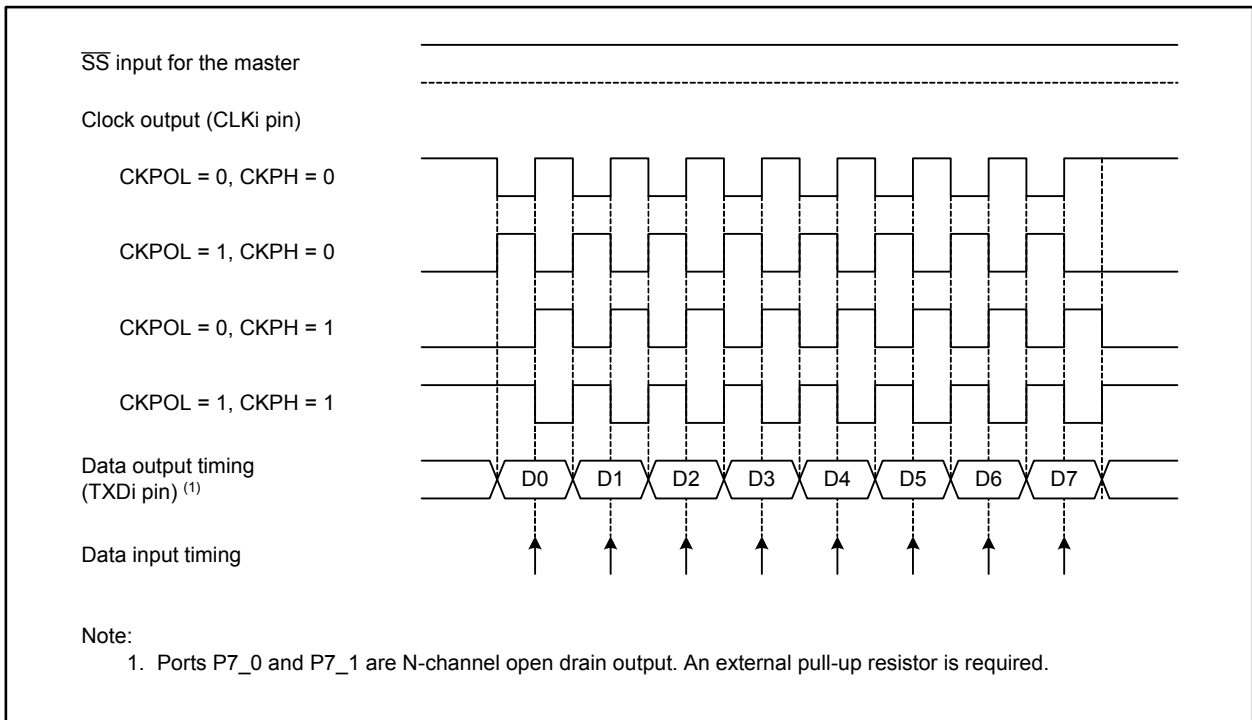


Figure 18.38 Transmit/Receive Timing in Master Mode

### 18.4.2.2 Transmit/Receive Timing in Slave Mode

When the DINC bit is 1 (slave mode), the CKDIR bit in the UIMR register should be set to 1 (external clock).

When the CKPH bit is 0 (no clock delay) while input at the  $\overline{SS}_i$  pin is high, the STXDi pin becomes high-impedance. When input at the  $\overline{SS}_i$  pin is low, the conditions for data transmission are all met, but output is undefined. Then the data transmission/reception starts synchronizing with the clock. Figure 18.39 shows the transmit/receive timing.

When the CKPH bit is 1 (clock delayed) while input at the  $\overline{SS}_i$  pin is high, the STXDi pin becomes high-impedance. When input at the  $\overline{SS}_i$  pin is low, the first data is output. Then the data transmission starts synchronizing with the clock. Figure 18.40 shows the transmit/receive timing.

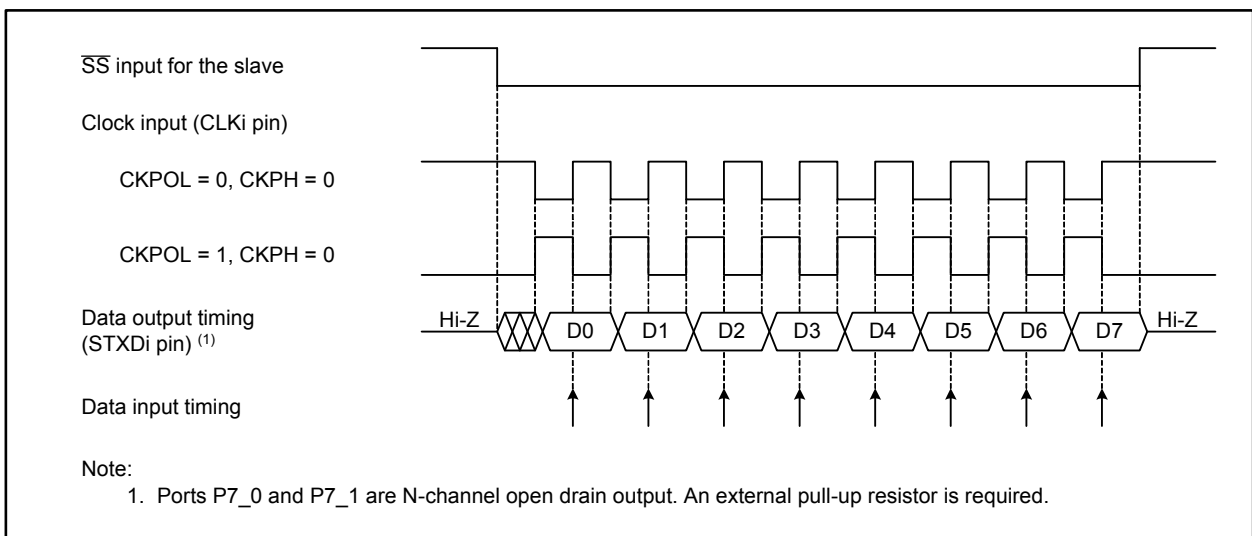


Figure 18.39 Transmit/Receive Timing in Slave Mode (CKPH = 0)

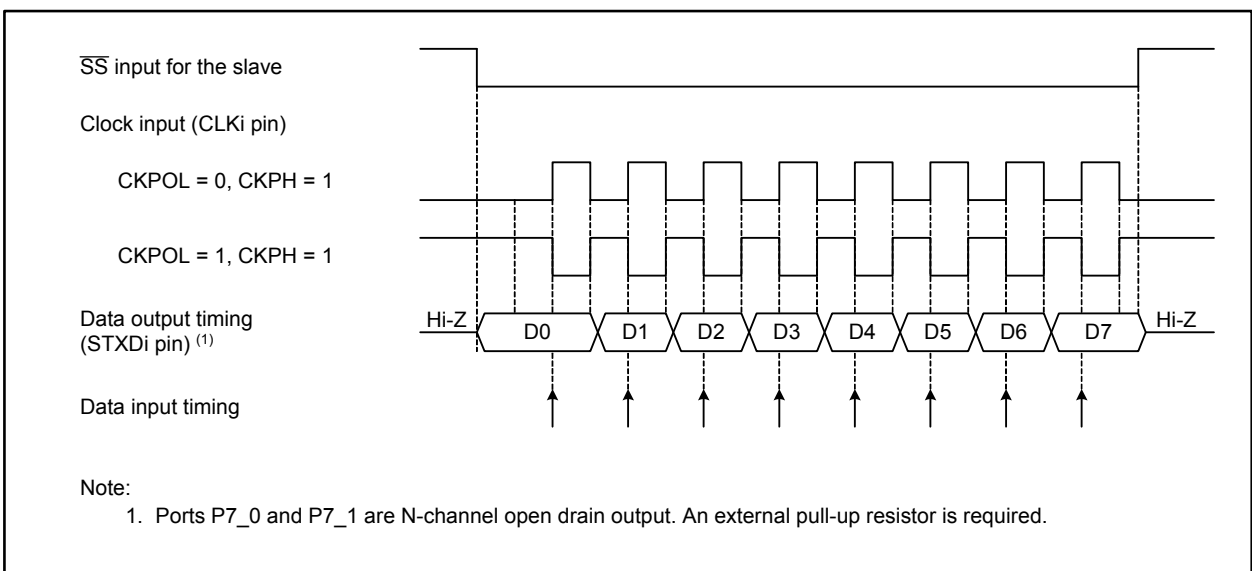


Figure 18.40 Transmit/Receive Timing in Slave Mode (CKPH = 1)

## 18.5 Notes on Serial Interface

### 18.5.1 Changing the UiBRG Register (i = 0 to 8)

- Set the UiBRG register after setting bits CLK1 and CLK0 in the UiC0 register. When these bits are changed, the UiBRG register must be set again.
- When a clock is input immediately after the UiBRG register is set to 00h, the counter may become FFh. In this case, it requires extra 256 clocks to reload 00h to the register. Once 00h is reloaded, the counter performs the operation without dividing the count source according to the setting.

### 18.5.2 Restriction for the Use of 64-pin Package's Products

In the 64-pin package, all signals of UART4, UART6, and UART7 are not connected to respective external pins. Use UART0 to UART3, UART5, or UART8.

### 18.5.3 Synchronous Serial Interface Mode

#### 18.5.3.1 Selecting an External Clock

- If an external clock is selected, the following conditions must be met while the external clock is held high when the CKPOL bit in the UiC0 register is 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge), or while the external clock is held low when the CKPOL bit is 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge) (i = 0 to 8):
  - The TE bit in the UiC1 register is 1 (transmission enabled).
  - The RE bit in the UiC1 register is 1 (reception enabled). This bit setting is not required when only transmitting.
  - The TI bit in the UiC1 register is 0 (data held in the UiTB register).

#### 18.5.3.2 Receive Operation

- In synchronous serial interface mode, the transmit/receive clock is controlled by the transmit control circuit. Set UARTi-associated registers for a transmit operation, even if the MCU is used only for receive operation (i = 0 to 8). Dummy data is output from the TXDi pin while receiving when the TXDi pin is set to output mode.
- When data is received continuously, an overrun error occurs when the RI bit in the UiC1 register is 1 (data held in the UiRB register) and the seventh bit of the next data is received in the UARTi receive shift register. Then, the OER bit in the UiRB register becomes 1 (overrun error occurred). In this case, the UiRB register becomes undefined. If an overrun error occurs, the IR bit in the SiRIC register does not change to 1.

### 18.5.4 Special Mode 1 (I<sup>2</sup>C Mode)

- To generate a START condition, STOP condition, or repeated START condition, set the STSPSEL bit in the UiSMR4 register to 0 (i = 0 to 6). Then, wait at least a half clock cycle of the transmit/receive clock to change the condition generate bits (STAREQ, RSTAREQ, or STPREQ bit) from 0 to 1.

### 18.5.5 Reset Procedure or Suspend/Resume Procedure

Operations which result in communication errors such as rewriting function select registers during transmission/reception should not be performed. Follow the procedure below to reset the internal circuit once the communication error occurs in the following cases: when the operation above is performed by a receiver or transmitter or when a bit slip is caused by noise.

Also follow the procedure below when suspending and resuming communication in an emergency.

#### A. Synchronous Serial Interface Mode

- (1) Set the TE bit in the UiC1 register to 0 (transmission disabled) and the RE bit to 0 (reception disabled) (i = 0 to 8).
- (2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
- (3) Set bits SMD2 to SMD0 in the UiMR register to 001b (synchronous serial interface mode).
- (4) Set the TE bit in the UiC1 register to 1 (transmission enabled) and the RE bit to 1 (reception enabled) if necessary.

#### B. UART Mode

- (1) Set the TE bit in the UiC1 register to 0 (transmission disabled) and the RE bit to 0 (reception disabled).
- (2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
- (3) Set bits SMD2 to SMD0 in the UiMR register to 100b (UART mode, 7-bit character length), 101b (UART mode, 8-bit character length), or 110b (UART mode, 9-bit character length).
- (4) Set the TE bit in the UiC1 register to 1 (transmission enabled) and the RE bit to 1 (reception enabled) if necessary.

## 19. A/D Converter

The A/D converter consists of one 10-bit successive approximation A/D converter with a capacitive coupling amplifier.

A/D converted results are stored in the A/D registers corresponding to selected pins. Results are stored in the AD00 register only when the DMAC operating mode is enabled.

When the A/D converter is not in use, power consumption can be reduced by setting the VCUT bit in the AD0CON1 register to 0 (VREF disconnected). This bit setting enables the power supply from the VREF pin to the resistor ladder to stop.

Table 19.1 lists specifications of the A/D converter. Figure 19.1 shows a block diagram of the A/D converter. Figures 19.2 to 19.7 show registers associated with the A/D converter.

**Table 19.1 A/D Converter Specifications**

Item	Specification
A/D conversion method	Capacitance-based successive approximation
Analog input voltage <sup>(1)</sup>	0 V to AVCC (VCC1)
Operating clock, $\phi_{AD}$ <sup>(2)</sup>	fAD, fAD divided by 2, fAD divided by 3, fAD divided by 4, fAD divided by 6, or fAD divided by 8
Resolution	8 bits or 10 bits
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, repeat sweep mode 1, multi-port single sweep mode, multi-port repeat sweep mode 0
Analog input pins <sup>(3)</sup>	26 <sup>(4)</sup> 8 pins each for AN, AN0, and AN2 <sup>(5)</sup> 2 function-extended input pins (ANEX0 and ANEX1) <sup>(5)</sup>
A/D conversion start conditions	<ul style="list-style-type: none"> <li>• Software trigger The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program</li> <li>• External trigger (retrigger is enabled) An input signal at the ADTRG pin switches from high to low after the ADST bit is set to 1 by a program</li> <li>• Hardware trigger (retrigger is enabled) <ul style="list-style-type: none"> <li>• Generation of a timer B2 interrupt request which has passed through the circuit to set an interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program</li> </ul> </li> </ul>
Conversion rates per pin	<ul style="list-style-type: none"> <li>• Without sample and hold function 49 <math>\phi_{AD}</math> cycles at 8-bit resolution 59 <math>\phi_{AD}</math> cycles at 10-bit resolution including 2 <math>\phi_{AD}</math> cycles for sampling time</li> <li>• With sample and hold function 28 <math>\phi_{AD}</math> cycles at 8-bit resolution 33 <math>\phi_{AD}</math> cycles at 10-bit resolution including 3 <math>\phi_{AD}</math> cycles for sampling time</li> </ul>

## Notes:

1. The analog input voltage is not dependent on whether the sample and hold function is enabled or disabled.
2. The  $\phi_{AD}$  frequency should be as follows:
  - When VCC1 = 4.2 to 5.5 V, 16 MHz or below
  - When VCC1 = 3.0 to 4.2 V, 10 MHz or below
  - When not using the sample and hold function, 250 kHz or above
  - When using the sample and hold function, 1 MHz or above
3. In the 100-pin package, when AVCC = VREF = VCC1  $\geq$  VCC2, A/D input voltage for pins AN\_0 to AN\_7, ANEX0, and ANEX1 should be VCC1 or lower, and A/D input voltage for pins AN0\_0 to AN0\_7, and AN2\_0 to AN2\_7 should be VCC2 or lower. In the 64-pin package, when AVCC = VREF = VCC1, all A/D input voltages should be VCC1 or lower.
4. Specification of the 100-pin package. In the 64-pin package, 20 channels are available.
5. Pins AN0\_4 to AN0\_7, ANEX0, and ANEX1 are not available in the 64-pin package.

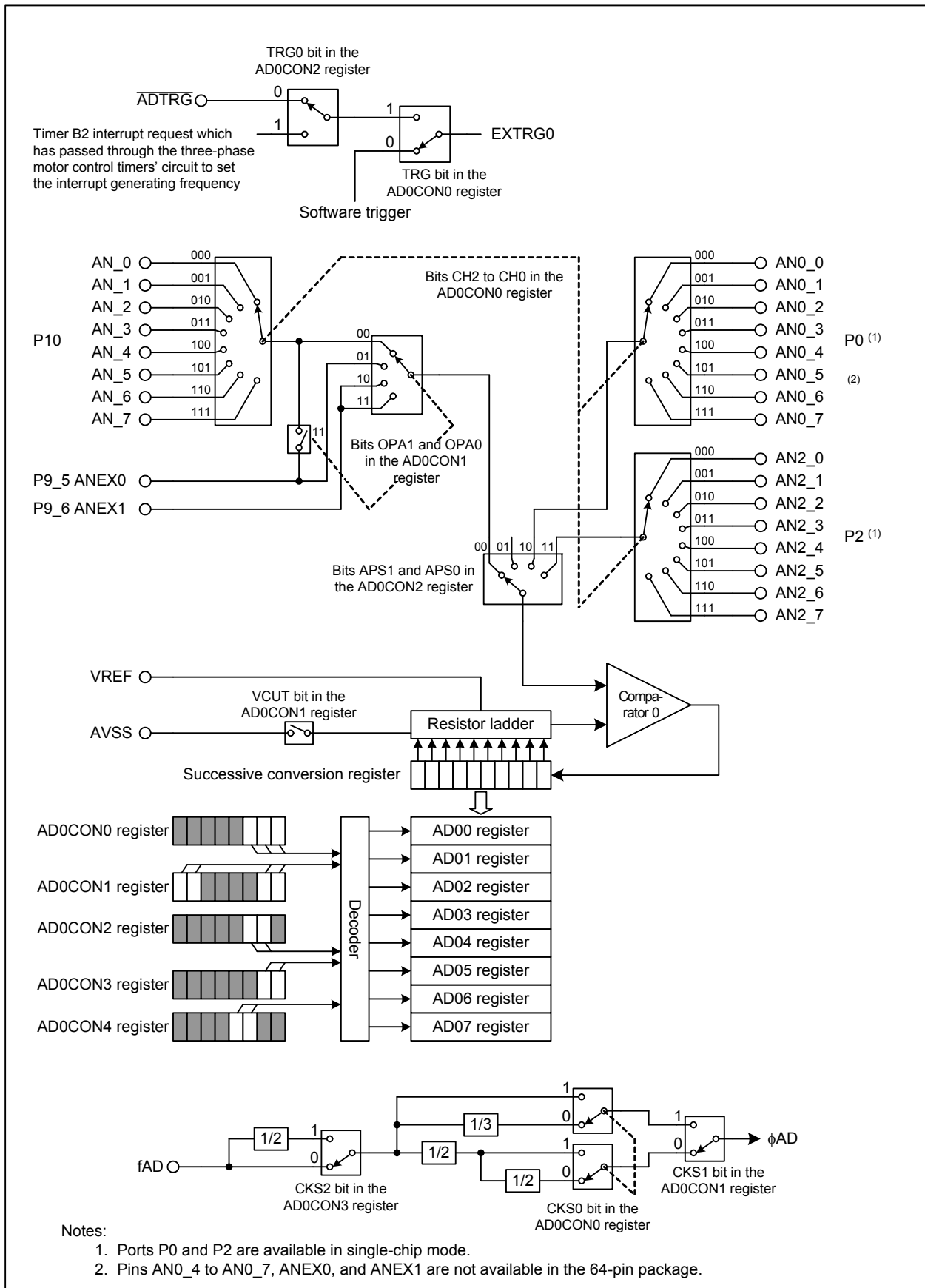


Figure 19.1 A/D Converter Block Diagram

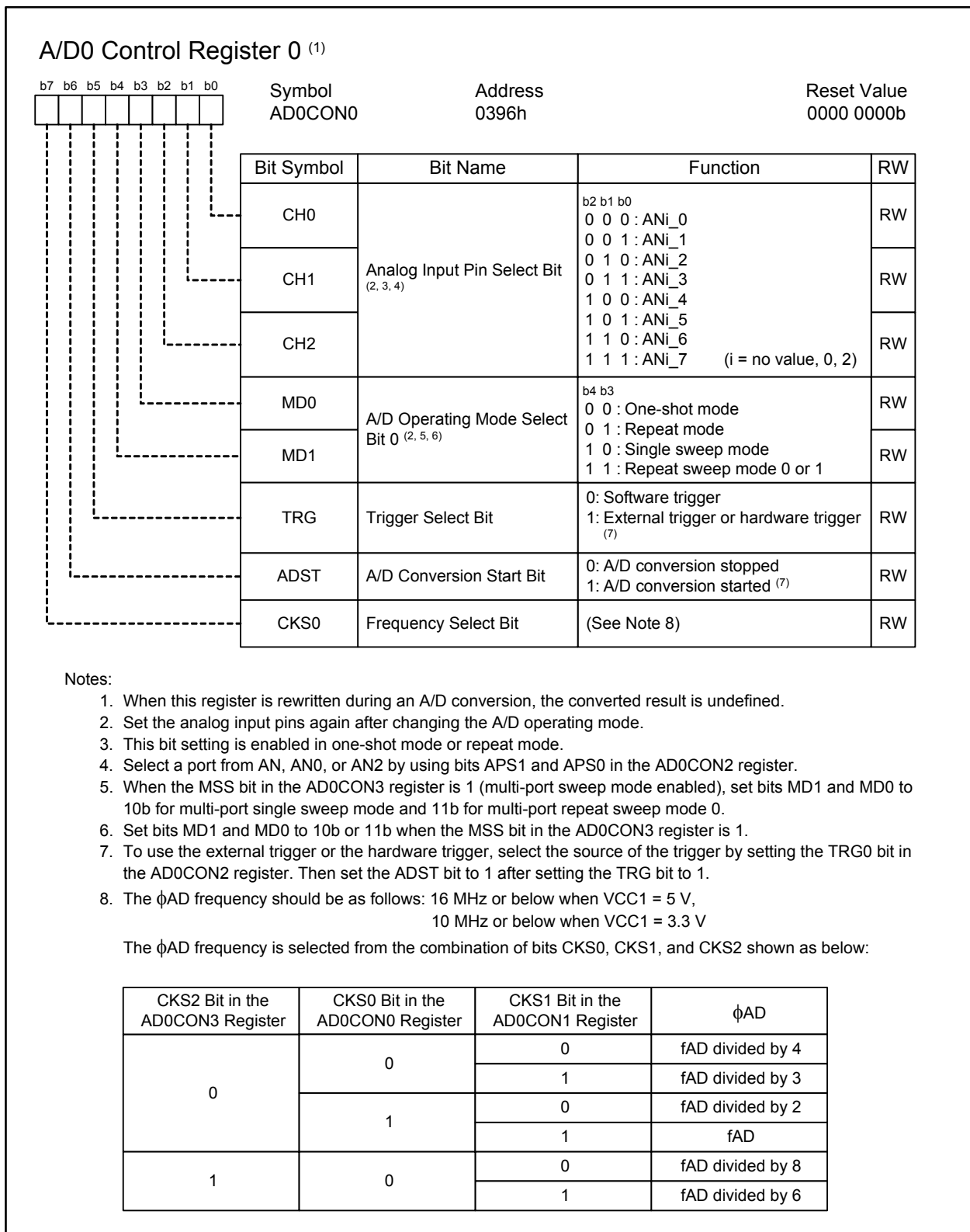


Figure 19.2 AD0CON0 Register



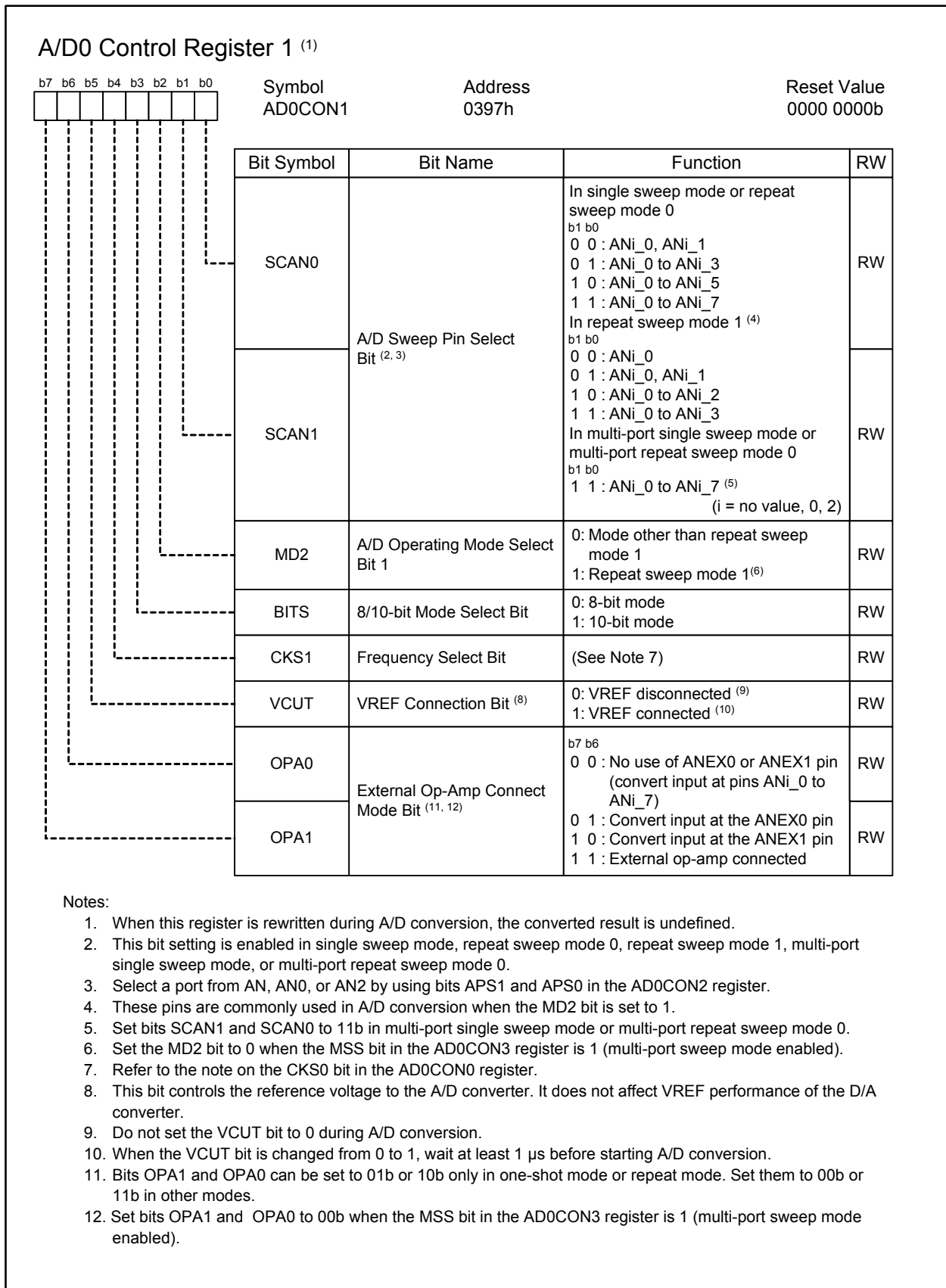


Figure 19.3 AD0CON1 Register

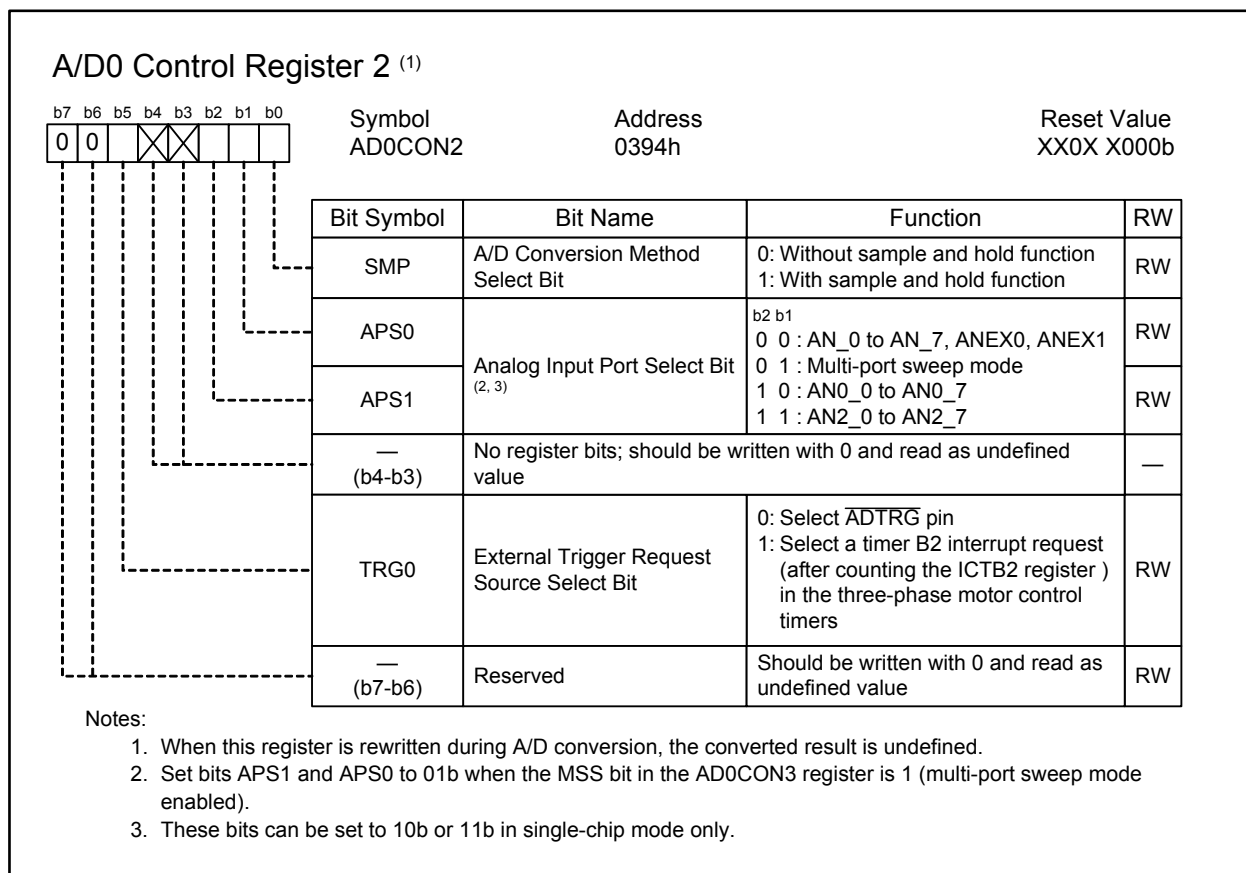
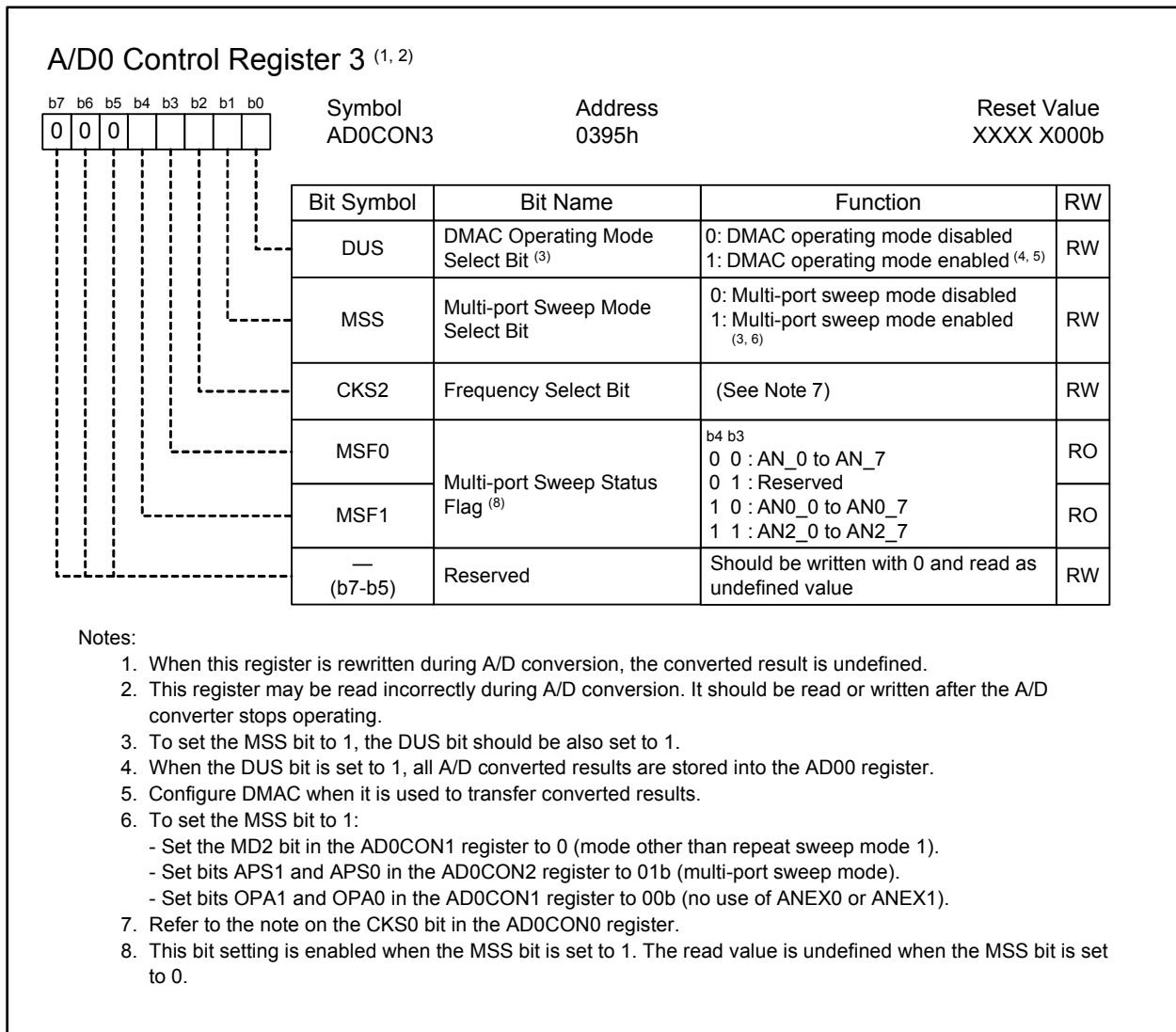


Figure 19.4 AD0CON2 Register



**Figure 19.5 AD0CON3 Register**

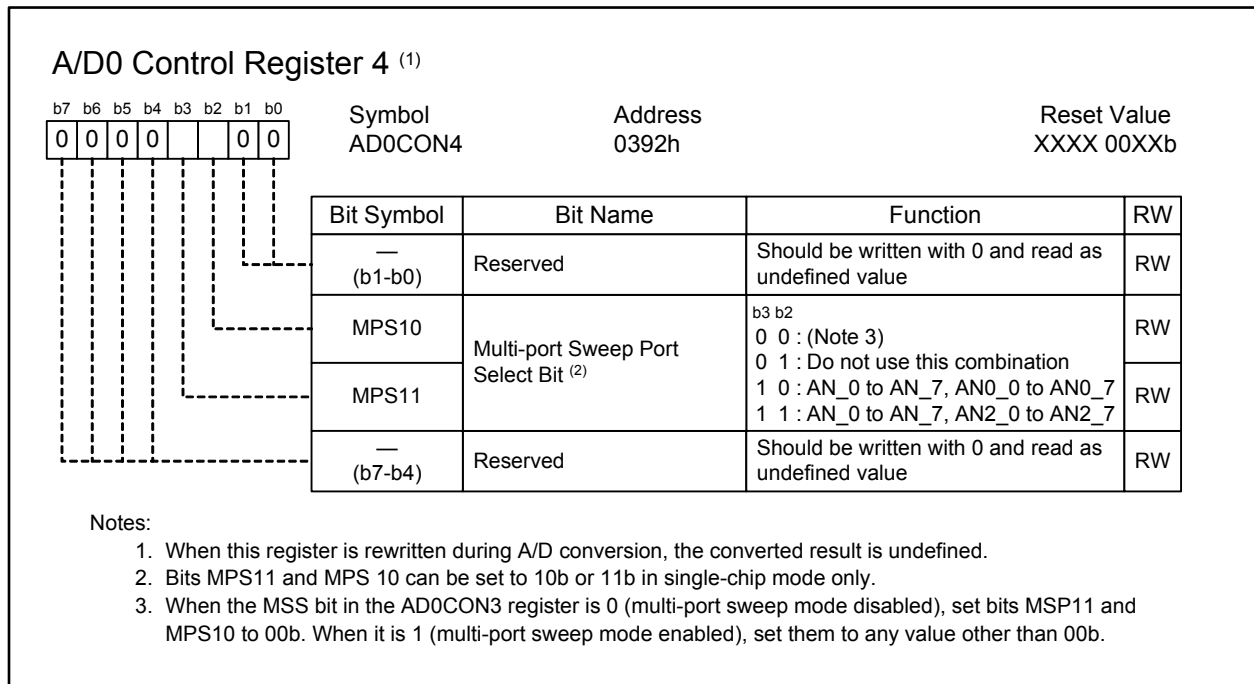


Figure 19.6 AD0CON4 Register

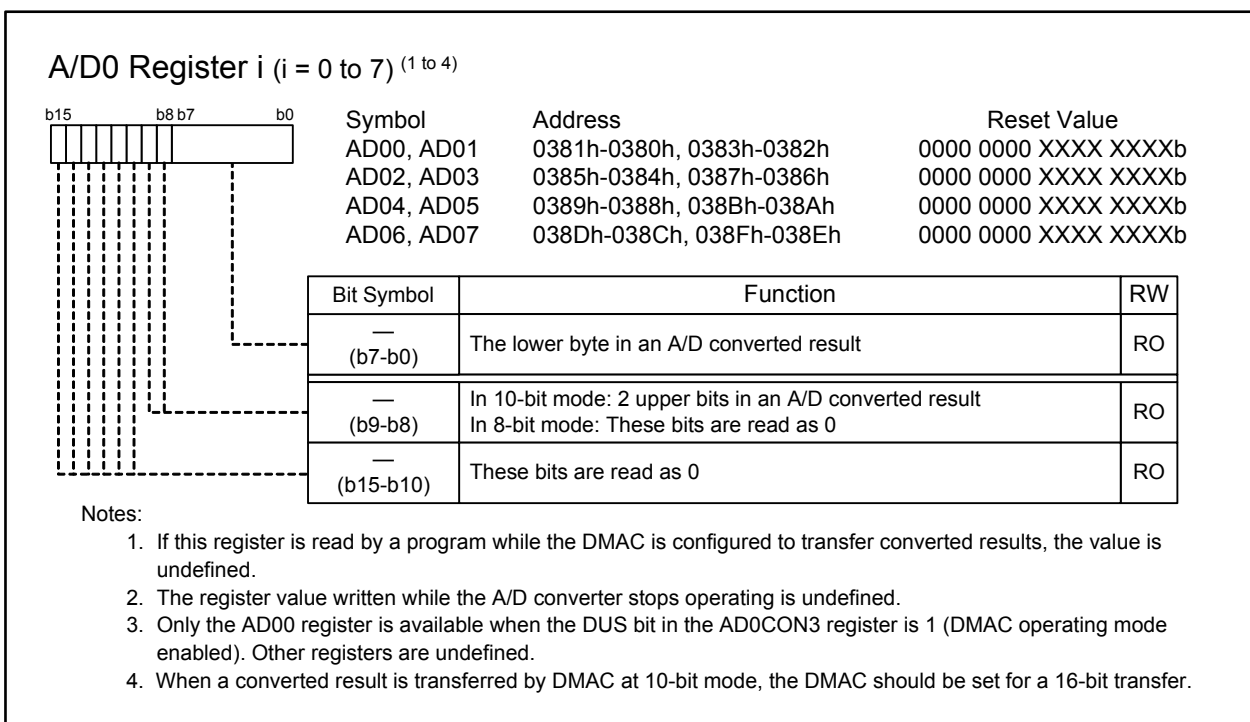


Figure 19.7 Registers AD00 to AD07

## 19.1 Mode Descriptions

### 19.1.1 One-shot Mode

In one-shot mode, the analog voltage applied to a selected pin is converted into a digital code only once. Table 19.2 lists specifications of one-shot mode.

**Table 19.2 One-shot Mode Specifications**

Item	Specification
Function	Converts the analog voltage applied to a pin into a digital code only once. The pin is selected by setting bits CH2 to CH0 in the AD0CON0 register, bits OPA1 and OPA0 in the AD0CON1 register, and bits APS1 and APS0 in the AD0CON2 register
Start conditions	When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. When the TRG bit is 1 (external trigger or hardware trigger) Set the TRG0 bit in the AD0CON2 register to select external trigger request source. <ul style="list-style-type: none"> <li>When 0 is selected, an input signal at the <math>\overline{\text{ADTRG}}</math> pin switches from high to low after the ADST bit is set to 1 by a program.</li> <li>When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.</li> </ul>
Stop conditions	<ul style="list-style-type: none"> <li>A/D conversion is completed (the ADST bit is set to 0 when the software trigger is selected)</li> <li>The ADST bit is set to 0 (A/D conversion stopped) by a program</li> </ul>
Interrupt request generation timing	When A/D conversion is completed, an interrupt request is generated
Input pin to be selected	One pin is selected from among AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0, and ANEX1
Reading A/D converted result	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled) Read the AD0j register corresponding to the selected pin (j = 0 to 7) When the DUS bit is 1 (DMAC operating mode enabled) Configure the DMAC (refer to 13. "DMAC"). Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space. Do not read the AD00 register by a program

### 19.1.2 Repeat Mode

In repeat mode, the analog voltage applied to a selected pin is repeatedly converted into a digital code. Table 19.3 lists specifications of repeat mode.

**Table 19.3 Repeat Mode Specifications**

Item	Specification
Function	Converts the analog voltage input to a pin into a digital code repeatedly. The pin is selected by setting bits CH2 to CH0 in the AD0CON0 register, bits OPA1 and OPA0 in the AD0CON1 register, and bits APS1 and APS0 in the AD0CON2 register
Start conditions	When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. When the TRG bit is 1 (external trigger or hardware trigger) Set the TRG0 bit in the AD0CON2 register to select external trigger request source. <ul style="list-style-type: none"> <li>• When 0 is selected, an input signal at the <math>\overline{\text{ADTRG}}</math> pin switches from high to low after the ADST bit is set to 1 by a program.</li> <li>• When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.</li> </ul>
Stop conditions	The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled), no interrupt request is generated. When the DUS bit is 1 (DMAC operating mode enabled), each time A/D conversion is completed, an interrupt request is generated
Analog voltage input pins	One pin is selected from among AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0, and ANEX1
Reading A/D converted result	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled) Read the AD0j register corresponding to the selected pin (j = 0 to 7) When the DUS bit is 1 (DMAC operating mode enabled) <ul style="list-style-type: none"> <li>• When the converted result is transferred by DMAC Configure the DMAC (refer to 13. "DMAC"). Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space. Do not read the AD00 register by a program</li> <li>• When the converted result is transferred by a program Read the AD00 register after the IR bit in the AD0IC register becomes 1. Set the IR bit back to 0</li> </ul>

### 19.1.3 Single Sweep Mode

In single sweep mode, the analog voltage applied to selected pins is converted one-by-one into a digital code. Table 19.4 lists specifications of single sweep mode.

**Table 19.4 Single Sweep Mode Specifications**

Item	Specification
Function	Converts the analog voltage input to a set of pins into a digital code one-by-one. The pins are selected by setting bits SCAN1 and SCAN0 in the AD0CON1 register and bits APS1 and APS0 in the AD0CON2 register
Start conditions	When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. When the TRG bit is 1 (external trigger or hardware trigger) Set the TRG0 bit in the AD0CON2 register to select external trigger request source. <ul style="list-style-type: none"> <li>• When 0 is selected, an input signal at the <math>\overline{\text{ADTRG}}</math> pin switches from high to low after the ADST bit is set to 1 by a program.</li> <li>• When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.</li> </ul>
Stop conditions	<ul style="list-style-type: none"> <li>• A/D conversion is completed (the ADST bit is set to 0 when the software trigger is selected)</li> <li>• The ADST bit is set to 0 (A/D conversion stopped) by a program</li> </ul>
Interrupt request generation timing	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled) when a sweep is completed, an interrupt request is generated. When the DUS bit is 1 (DMAC operating mode enabled), each time A/D conversion is completed, an interrupt request is generated
Analog voltage input pins	Selected from a group of 2 pins (ANi_0 and ANi_1), 4 pins (ANi_0 to ANi_3), 6 pins (ANi_0 to ANi_5), or 8 pins (ANi_0 to ANi_7) (i = no value, 0, 2)
Reading A/D converted result	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled) Read the AD0j register corresponding to the selected pin (j = 0 to 7) When the DUS bit is 1 (DMAC operating mode enabled) Configure the DMAC (refer to 13. "DMAC"). Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space. Do not read the AD00 register by a program

### 19.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, the analog voltage applied to selected pins is repeatedly converted into a digital code. Table 19.5 lists specifications of repeat sweep mode 0.

**Table 19.5 Repeat Sweep Mode 0 Specifications**

Item	Specification
Function	Converts the analog voltage input to a set of pins into a digital code repeatedly. The pins are selected by setting bits SCAN1 and SCAN0 in the AD0CON1 register and APS1 and APS0 in the AD0CON2 register
Start conditions	When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. When the TRG bit is 1 (external trigger or hardware trigger) Set the TRG0 bit in the AD0CON2 register to select external trigger request source. <ul style="list-style-type: none"> <li>• When 0 is selected, an input signal at the ADTRG pin switches from high to low after the ADST bit is set to 1 by a program.</li> <li>• When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.</li> </ul>
Stop conditions	The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled), no interrupt request is generated. When the DUS bit is 1 (DMAC operating mode enabled), each time A/D conversion is completed, an interrupt request is generated
Analog voltage input pins	Selected from a group of 2 pins (ANi_0 and ANi_1), 4 pins (ANi_0 to ANi_3), 6 pins (ANi_0 to ANi_5), or 8 pins (ANi_0 to ANi_7) (i = no value, 0, 2)
Reading A/D converted result	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled) Read the AD0j register corresponding to the selected pin (j = 0 to 7) When the DUS bit is 1 (DMAC operating mode enabled) <ul style="list-style-type: none"> <li>• When the converted result is transferred by DMAC Configure the DMAC (refer to 13. "DMAC"). Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space. Do not read the AD00 register by a program</li> <li>• When the converted result is transferred by a program Read the AD00 register after the IR bit in the AD0IC register becomes 1. Set the IR bit back to 0</li> </ul>



### 19.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, the analog voltage applied to eight selected pins including one to four prioritized pins is repeatedly converted into a digital code. Table 19.6 lists specifications of repeat sweep mode 1.

**Table 19.6 Repeat Sweep Mode 1 Specifications**

Item	Specification
Function	The analog voltage applied to eight selected pins including one to four prioritized pins is repeatedly converted into a digital code. The prioritized pins are selected by setting bits SCAN1 and SCAN0 in the AD0CON1 register and bits APS1 and APS0 in the AD0CON2 register For example, when AN_0 is selected, the A/D conversion is performed in the following order: AN_0→AN_1→AN_0→AN_2→AN_0→AN_3...
Start conditions	When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. When the TRG bit is 1 (external trigger or hardware trigger) Set the TRG0 bit in the AD0CON2 register to select external trigger request source. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. Retrigger is invalid. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled), no interrupt request is generated. When the DUS bit is 1 (DMAC operating mode enabled), each time A/D conversion is completed, an interrupt request is generated
Analog voltage input pins	8 (ANi_0 to ANi_7) (i = no value, 0, 2)
Prioritized pin(s)	Selected from a group of 1 pin (ANi_0), 2 pins (ANi_0 and ANi_1), 3 pins (ANi_0 to ANi_2), or 4 pins (ANi_0 to ANi_3)
Reading A/D converted result	When the DUS bit in the AD0CON3 register is 0 (DMAC operating mode disabled) Read the AD0j register corresponding to the selected pin (j = 0 to 7) When the DUS bit is 1 (DMAC operating mode enabled) • When the converted result is transferred by DMAC Configure the DMAC (refer to 13. "DMAC"). Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space. Do not read the AD00 register by a program • When the converted result is transferred by a program Read the AD00 register after the IR bit in the AD0IC register becomes 1. Set the IR bit back to 0

### 19.1.6 Multi-port Single Sweep Mode

In multi-port single sweep mode, the analog voltage applied to 16 selected pins is converted one-by-one into a digital code. The DUS bit in the AD0CON3 register should be set to 1 (DMAC operating mode enabled). Table 19.7 lists specifications of multi-port single sweep mode.

**Table 19.7 Multi-port Single Sweep Mode Specifications**

Item	Specification
Function	<p>Converts the analog voltage input to a set of 16 selected pins into a digital code one-by-one in the following order: AN_0 to AN_7→AN<sub>i</sub>_0 to AN<sub>i</sub>_7 (i = 0, 2)</p> <p>The 16 pins are selected by setting bits MPS11 and MPS10 in the AD0CON4 register</p> <p>For example, when bits MPS11 and MPS10 are set to 10b (AN_0 to AN_7, AN0_0 to AN0_7), the analog voltage is converted into a digital code in the following order:</p> <p>AN_0→AN_1→AN_2→AN_3→AN_4→AN_5→AN_6→AN_7→AN0_0→...→AN0_6→AN0_7</p>
Start conditions	<p>When the TRG bit in the AD0CON0 register is 0 (software trigger) The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program.</p> <p>When the TRG bit is 1 (external trigger or hardware trigger) Set the TRG0 bit in the AD0CON2 register to select external trigger request source.</p> <ul style="list-style-type: none"> <li>• When 0 is selected, an input signal at the <u>ADTRG</u> pin switches from high to low after the ADST bit is set to 1 by a program.</li> <li>• When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.</li> </ul>
Stop conditions	<ul style="list-style-type: none"> <li>• A/D conversion is completed (the ADST bit is set to 0 when the software trigger is selected)</li> <li>• The ADST bit is set to 0 (A/D conversion stopped) by a program</li> </ul>
Interrupt request generation timing	Every time A/D conversion is completed (set the DUS bit to 1)
Analog voltage input pins	A combination of pin group is selected from AN_0 to AN_7→AN0_0 to AN0_7, or AN_0 to AN_7→AN2_0 to AN2_7
Reading A/D converted result	<p>Set the DUS bit to 1 and configure the DMAC (refer to 13. "DMAC").</p> <p>Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space.</p> <p>Do not read the AD00 register by a program</p>

### 19.1.7 Multi-port Repeat Sweep Mode 0

In multi-port repeat sweep mode 0, the analog voltage applied to 16 selected pins is repeatedly converted into a digital code. The DUS bit in the AD0CON3 register should be set to 1 (DMAC operating mode enabled). Table 19.8 lists specifications of multi-port repeat sweep mode 0.

**Table 19.8 Multi-port Repeat Sweep Mode 0 Specifications**

Item	Specification
Function	<p>Converts the analog voltage input to a set of 16 selected pins into a digital code repeatedly in the following order: AN_0 to AN_7→ANi_0 to ANi_7 (i = 0, 2)</p> <p>The 16 pins are selected by setting bits MPS11 and MPS10 in the AD0CON4 register</p> <p>For example, when bits MPS11 and MPS10 are set to 10b (AN_0 to AN_7, AN0_0 to AN0_7), the analog voltage is converted into a digital code repeatedly in the following order:</p> <p>AN_0→AN_1→AN_2→AN_3→AN_4→AN_5→AN_6→AN_7→AN0_0→...→AN0_6→AN0_7</p>
Start conditions	<p>When the TRG bit in the AD0CON0 register is 0 (software trigger)</p> <p>The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program.</p> <p>When the TRG bit is 1 (external trigger or hardware trigger)</p> <p>Set the TRG0 bit in the AD0CON2 register to select external trigger request source.</p> <ul style="list-style-type: none"> <li>• When 0 is selected, an input signal at the <u>ADTRG</u> pin switches from high to low after the ADST bit is set to 1 by a program.</li> <li>• When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.</li> </ul>
Stop conditions	The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	Every time A/D conversion is completed (set the DUS bit to 1)
Analog voltage input pins	A combination of pin group is selected from AN_0 to AN_7→AN0_0 to AN0_7, or AN_0 to AN_7→AN2_0 to AN2_7
Reading A/D converted result	<p>Set the DUS bit to 1 and configure the DMAC (refer to 13. "DMAC").</p> <p>Then the A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to a given memory space.</p> <p>Do not read the AD00 register by a program</p>

## 19.2 Functions

### 19.2.1 Resolution Selection

Resolution is selected by setting the BITS bit in the AD0CON1 register. When the BITS bit is set to 1 (10-bit precision), the A/D converted result is stored into bits 9 to 0 in the AD0i register (i = 0 to 7). When the BITS bit is set to 0 (8-bit precision), the result is stored into bits 7 to 0 in the AD0i register.

### 19.2.2 Sample and Hold Function

This function improves the conversion rate per pin to 28  $\phi$ AD cycles at 8-bit resolution and 33  $\phi$ AD cycles for 10-bit resolution. This function is available in all operating modes and is enabled by setting the SMP bit in the AD0CON2 register to 1 (with sample and hold function). Start A/D conversion after setting the SMP bit.

### 19.2.3 Trigger Selection

A trigger to start A/D conversion is specified by the combination of TRG bit in the AD0CON0 register and the TRG0 bit in the AD0CON2 register. Table 19.9 lists the settings of the trigger selection.

**Table 19.9 Trigger Selection Settings**

Bit and Setting		Trigger
AD0CON0 register	AD0CON2 register	
TRG = 0	—	Software trigger The ADST bit in the AD0CON0 register is set to 1
TRG = 1 (1, 2)	TRG0 = 0	External trigger Falling edge of a signal applied to the $\overline{\text{ADTRG}}$ pin
	TRG0 = 1	Hardware trigger Generation of a timer B2 interrupt request which has passed through the circuit to set the interrupt generating frequency in the three-phase motor control timers

Notes:

1. A/D conversion starts when a trigger is generated while the ADST bit is 1 (A/D conversion started).
2. When an external trigger or a hardware trigger is generated during A/D conversion, the A/D converter aborts the operation in progress. Then, it restarts the operation.

### 19.2.4 DMAC Operating Mode

DMAC operating mode can be used in all operating modes. DMAC operating mode is highly recommended when the A/D converter is in multi-port single sweep mode or multi-port repeat sweep mode 0. When the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode enabled), all A/D converted results are stored in the AD00 register. The DMAC transfers the data from the AD00 register to a given memory space every time A/D conversion is completed at a pin. 8-bit DMA transfer should be selected for 8-bit resolution. For 10-bit resolution, 16-bit DMA transfer should be selected. Refer to 13. "DMAC" for details.

### 19.2.5 Function-extended Analog Input Pins

In one-shot mode and repeat mode, pins ANEX0 and ANEX1 can be used as analog input pins by setting bits OPA1 and OPA0 in the AD0CON1 register (refer to Table 19.10). The A/D converted results of pins ANEX0 and ANEX1 are stored into registers AD00 and AD01, respectively. However, when the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode enabled), all results are stored into the AD00 register.

To use function-extended analog input pins, bits APS1 and APS0 in the AD0CON2 register should be set to 00b (AN0 to AN7, ANEX0, ANEX1 function as analog input ports) and the MSS bit in the AD0CON3 register to 0 (multi-port sweep mode disabled).

**Table 19.10 Function-extended Analog Input Pin Settings**

AD0CON1 Register		ANEX0	ANEX1
OPA1	OPA0		
0	0	Not used	Not used
0	1	Analog input	Not used
1	0	Not used	Analog input
1	1	Output to an external op-amp	Input from an external op-amp

### 19.2.6 External Operating Amplifier (Op-Amp) Connection Mode

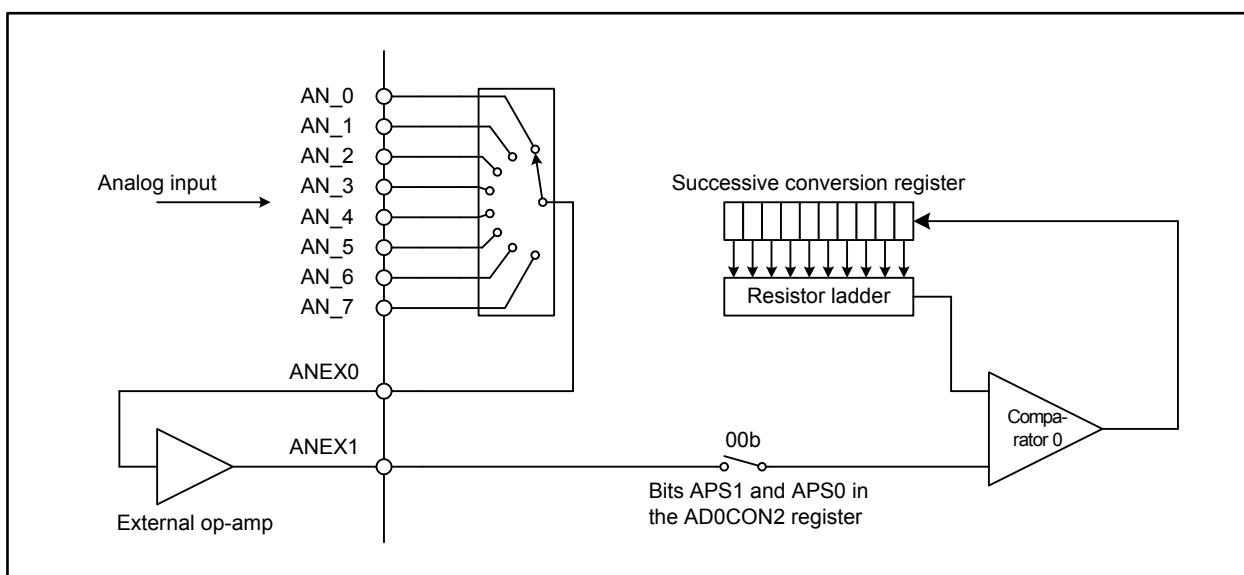
In external op-amp connection mode, multiple analog inputs can be amplified by one external op-amp using function-extended analog input pins ANEX0 and ANEX1.

When bits OPA1 and OPA0 in the AD0CON1 register are 11b (external op-amp connected), the voltage applied to pins AN0 to AN7 is output from the ANEX0 pin. This output signal should be amplified by an external op-amp and applied to the ANEX1 pin.

The analog voltage applied to the ANEX1 pin is converted into a digital code. The converted result is stored in the corresponding AD0i register ( $i = 0$  to 7). The conversion rate varies with the response of the external op-amp. Note that the ANEX0 pin should not be connected to the ANEX1 pin directly.

To use external op-amp connection mode, set bits APS1 and APS0 in the AD0CON2 register to 00b.

Figure 19.8 shows an example of an external op-amp connection.



**Figure 19.8 External Op-Amp Connection**

### 19.2.7 Power Saving

When the A/D converter is not in use, power consumption can be reduced by setting the VCUT bit in the AD0CON1 to 0 (VREF disconnected). With this bit setting, the reference voltage input pin (VREF) can be disconnected from the resistor ladder, which enables the power supply from the VREF to the resistor ladder to stop.

To use the A/D converter, set the VCUT bit to 1 (VREF connected) and wait at least 1  $\mu\text{s}$  before setting the ADST bit in the AD0CON0 register to 1 (A/D conversion started). Bits ADST and VCUT should not be set to 1 simultaneously. The VCUT bit should not be set to 0 during A/D conversion.

The VCUT bit does not affect VREF performance of the D/A converter (refer to Figure 19.9).

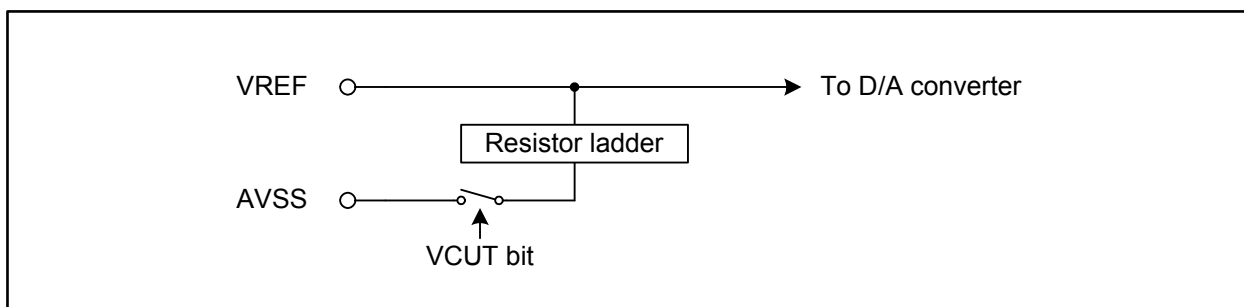


Figure 19.9 Power Supply by VCUT Bit

### 19.2.8 Output Impedance of Sensor Equivalent Circuit under A/D Conversion

Figure 19.10 shows an analog input pin and external sensor equivalent circuit.

To perform A/D conversion correctly, the internal capacitor (C) charging, shown in Figure 19.10, should be completed within the specified period. This period, called the sampling time, is 2  $\phi\text{AD}$  cycles for conversion without the sample and hold function and 3  $\phi\text{AD}$  cycles for conversion with this function.

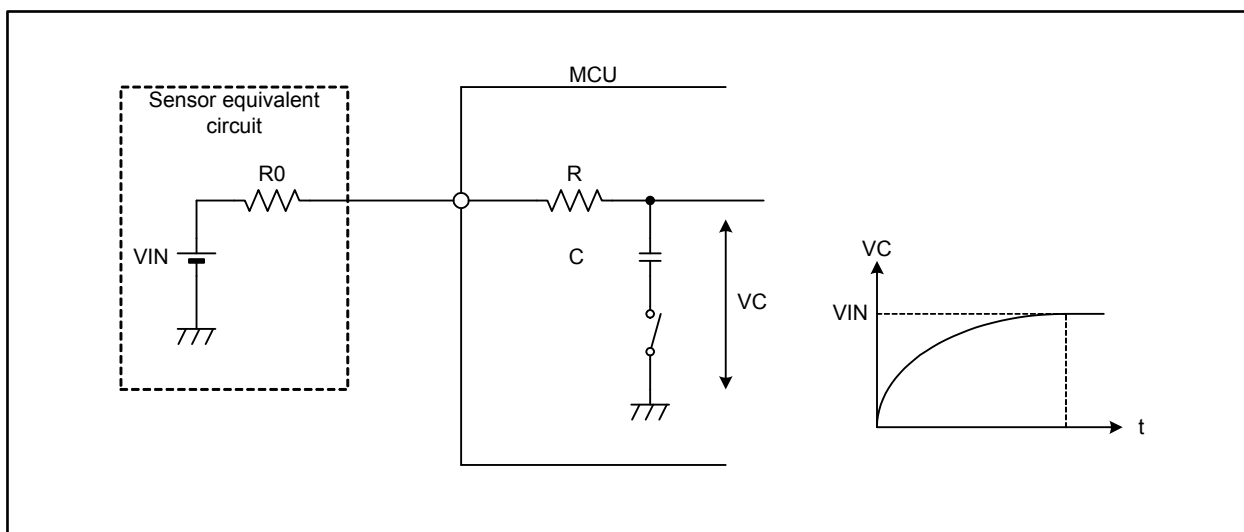


Figure 19.10 Analog Input Pin and External Sensor Equivalent Circuitry

The voltage between pins (VC) is expressed as follows:

$$VC = VIN \left\{ 1 - e^{-\frac{t}{C(R0+R)}} \right\}$$

When  $t = T$  and the precision (error) is  $x$  or less,

$$VC = VIN - \frac{x}{y} VIN = VIN \left( 1 - \frac{x}{y} \right)$$

Thus, output impedance of the sensor equivalent circuit (R0) is determined by the following formulas:

$$e^{-\frac{T}{C(R0+R)}} = \frac{x}{y}$$

$$-\frac{T}{C(R0+R)} = \ln \frac{x}{y}$$

$$R0 = -\frac{T}{C \ln \frac{x}{y}} - R$$

where:

T[s] = Sampling time

R0[Ω] = Output impedance of the sensor equivalent circuit

VC = Potential difference between edges of capacitor C

R[Ω] = Internal resistance of the MCU

x[LSB] = Precision (error) of the A/D converter

y[step] = Resolution of the A/D converter (1024 steps at 10-bit mode, 256 steps at 8-bit mode)

When  $\phi_{AD} = 10$  MHz, the A/D conversion mode is 10-bit resolution with the sample and hold function, the output impedance (R0) with the precision (error) of 0.1 LSB or less is determined by the following formula:

Using  $T = 0.3 \mu\text{s}$ ,  $R = 2.0 \text{ k}\Omega$  (reference value),  $C = 8.0 \text{ pF}$  (reference value),  $x = 0.1$ ,  $y = 1024$ ,

$$R0 = -\frac{0.3 \times 10^{-6}}{8.0 \times 10^{-12} \times \ln \frac{0.1}{1024}} - 2.0 \times 10^3$$

$$= 2061$$

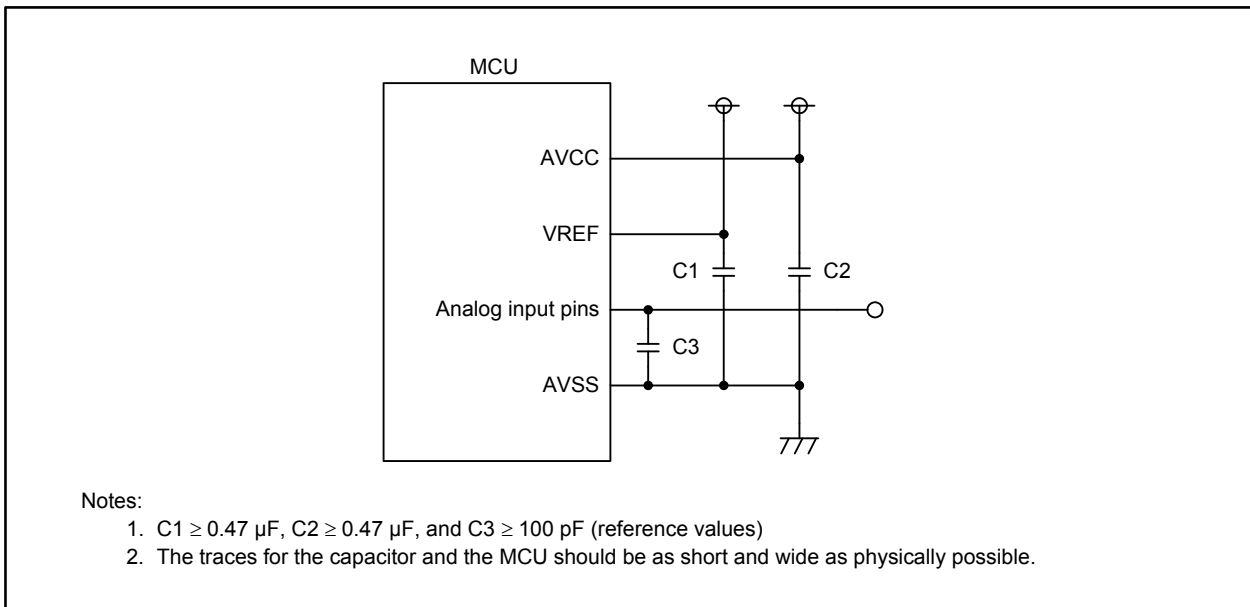
Thus, the allowable output impedance of the sensor equivalent circuit (R0), making the precision (error) of 0.1 LSB or less, should be less than 2 kΩ.

The actual error, however, is the value of absolute precision added to the 0.1 LSB mentioned above.

## 19.3 Notes on A/D Converter

### 19.3.1 Notes on Designing Boards

- Three capacitors should be placed between the AVSS pin and pins such as AVCC, VREF, and analog inputs (AN\_0 to AN\_7, AN0\_0 to AN0\_7, and AN2\_0 to AN2\_7) to avoid erroneous operations caused by noise or latchup, and to reduce conversion errors. Figure 19.11 shows an example of pin configuration for A/D converter.



**Figure 19.11 Pin Configuration for the A/D Converter**

- Do not use AN\_4 to AN\_7 for analog input if the key input interrupt is to be used. Otherwise, a key input interrupt request occurs when the A/D input voltage becomes VIL or lower.
- In the 100-pin package, when  $AVCC = VREF = VCC1 \geq VCC2$ , A/D input voltage for pins AN\_0 to AN\_7, ANEX0, and ANEX1 should be VCC1 or lower, and A/D input voltage for pins AN0\_0 to AN0\_7, and AN2\_0 to AN2\_7 should be VCC2 or lower. In the 64-pin package, under the same condition as that in the 100-pin package, all A/D input voltages should be VCC1 or lower.



### 19.3.2 Notes on Programming

- The following registers should be written while A/D conversion is stopped. That is, before a trigger occurs: AD0CON0 (except the ADST bit), AD0CON1, AD0CON2, AD0CON3, and AD0CON4.
- When the VCUT bit in the AD0CON1 register is changed from 0 (VREF connected) to 1 (VREF disconnected), wait for at least 1  $\mu$ s before starting A/D conversion. When not performing A/D conversion, set the VCUT bit to 0 to reduce power consumption.
- Set the port direction bit for the pin to be used as an analog input pin to 0 (input). Set the ASEL bit of the corresponding port function select register to 1 (port is used as A/D input).
- When the TRG bit in the AD0CON0 register is 1 (external trigger or hardware trigger), set the corresponding port direction bit (PD9\_7 bit) for the  $\overline{\text{ADTRG}}$  pin to 0 (input).
- The  $\phi_{\text{AD}}$  frequency should be 16 MHz or lower when VCC1 is 4.2 to 5.5 V, and 10 MHz or lower when VCC1 is 3.0 to 4.2 V. It should be 1 MHz or higher when the sample and hold function is enabled. If not, it should be 250 kHz or higher.
- When A/D operating mode (bits MD1 and MD0 in the AD0CON0 register or the MD2 bit in the AD0CON1 register) has been changed, reselect analog input pins by setting bits CH2 to CH0 in the AD0CON0 register or bits SCAN1 and SCAN0 in the AD0CON1 register.
- If the AD0i register is read when the A/D converted result is stored to the register, the stored value may have an error ( $i = 0$  to 7). Read the AD0i register after A/D conversion is completed. In one-shot mode or single sweep mode, read the AD0i register after the IR bit in the AD0IC register becomes 1 (interrupt requested). In repeat mode, repeat sweep mode 0, or repeat sweep mode 1, an interrupt request can be generated each time A/D conversion is completed when the DUS bit in the AD0CON3 register is 1 (DMAC operating mode enabled). Similar to the other modes above, read the AD00 register after the IR bit in the AD0IC register becomes 1 (interrupt requested).
- When an A/D conversion is halted by setting the ADST bit in the AD0CON0 register to 0, the converted result is undefined. In addition, the unconverted AD0i register may also become undefined. Consequently, the AD0i register should not be used just after A/D conversion is halted.
- External triggers cannot be used in DMAC operating mode. When the DMAC is configured to transfer converted results, do not read the AD00 register by a program.
- While in single sweep mode, if A/D conversion is halted by setting the ADST bit in the AD0CON0 register to 0 (A/D conversion is stopped), an interrupt request may be generated even though the sweep is not completed. To halt A/D conversion, disable interrupts before setting the ADST bit to 0.

## 20. D/A Converter

The MCU has two separate 8-bit R-2R resistor ladder D/A converters in the 100-pin package. In the 64-pin package, it has a single D/A converter.

Digital code is converted to an analog voltage when a value is written to the corresponding DA<sub>i</sub> register (i = 0, 1). The DA<sub>i</sub>E bit in the DACON register determines whether the D/A conversion result is output or not. Set the DA<sub>i</sub>E bit to 1 (output enabled) to output the converted value. This bit setting disables a pull-up resistor for the corresponding port.

Analog voltage to be output (V) is calculated based on the value (n) set in the DA<sub>i</sub> register (n is a decimal number).

$$V = \frac{V_{REF} \times n}{256} \quad (n = 0 \text{ to } 255)$$

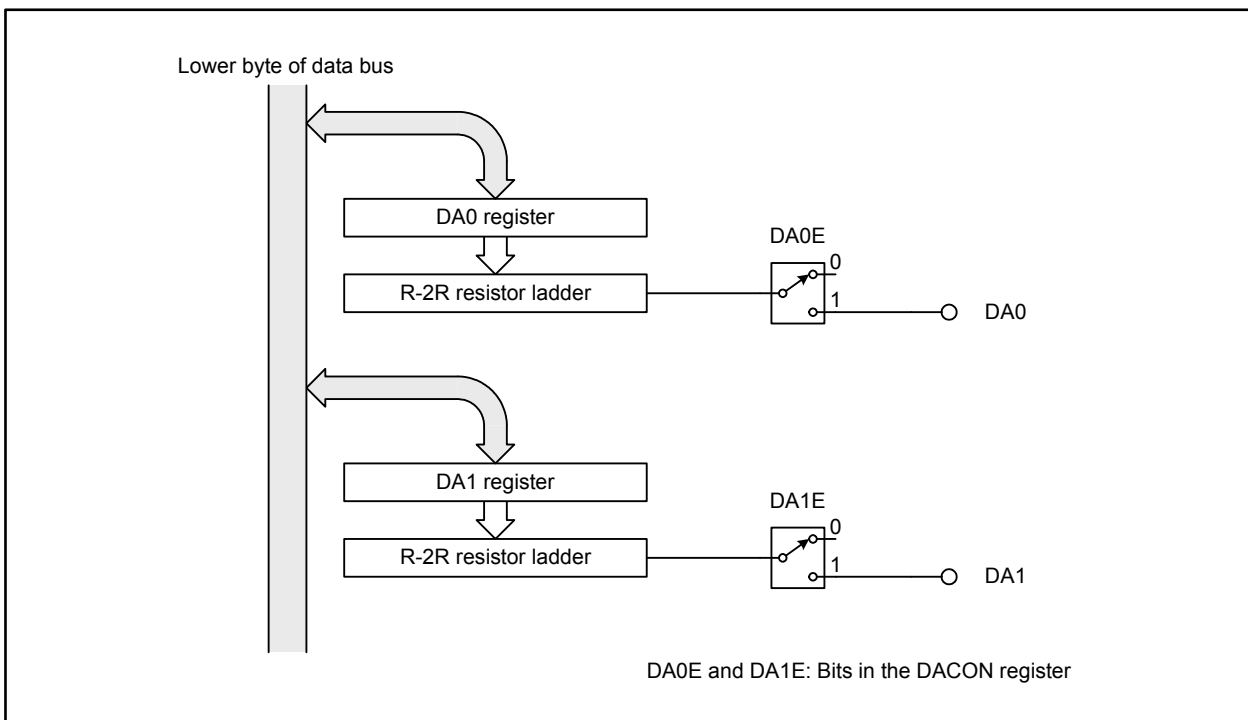
*V<sub>REF</sub>*: reference voltage

Table 20.1 lists specifications of the D/A converter. Figure 20.1 shows a block diagram of the D/A converter. Figures 20.2 and 20.3 show registers associated with the D/A converter. Figure 20.4 shows a D/A converter equivalent circuit.

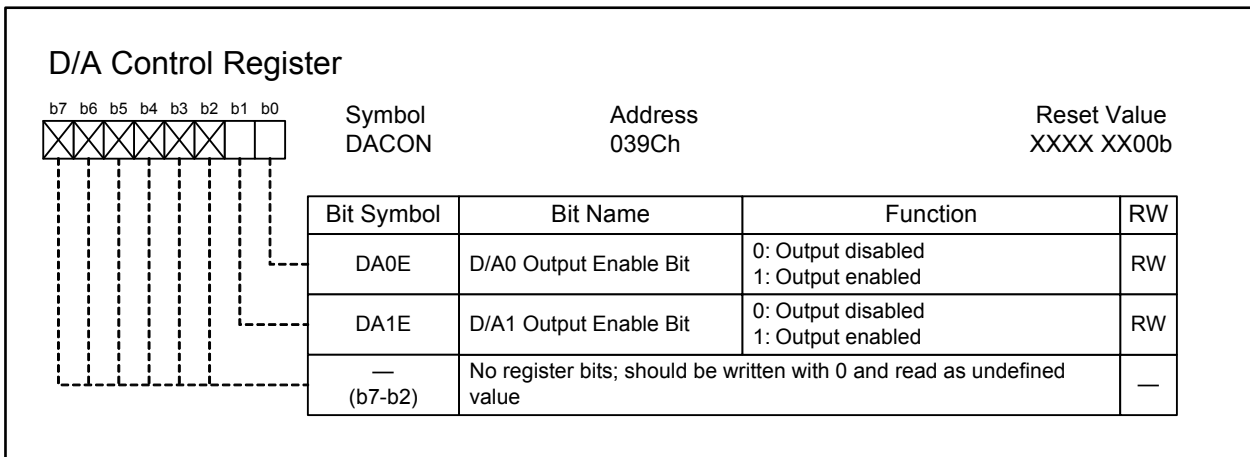
When the D/A converter is not used, set the DA<sub>i</sub> register to 00h and the DA<sub>i</sub>E bit to 0 (output disabled).

**Table 20.1 D/A Converter Specifications**

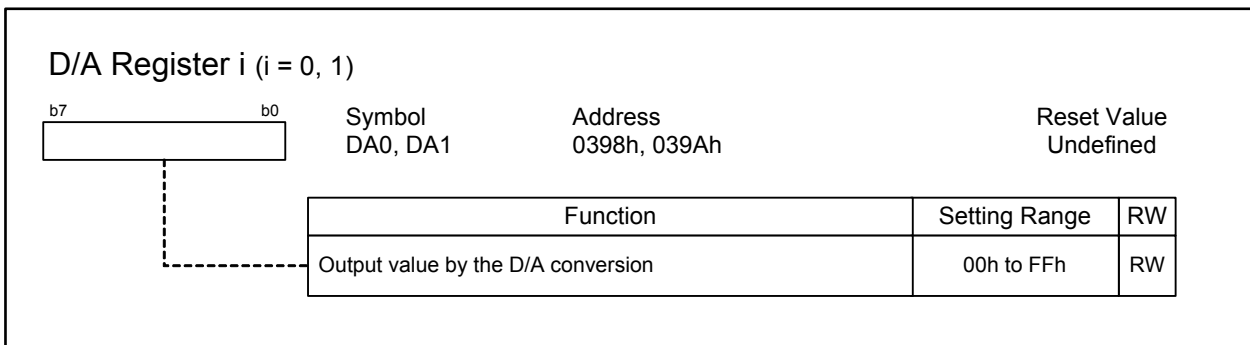
Item	Specification
D/A conversion method	R-2R resistor ladder
Resolution	8 bits
Analog output pins	2 channels



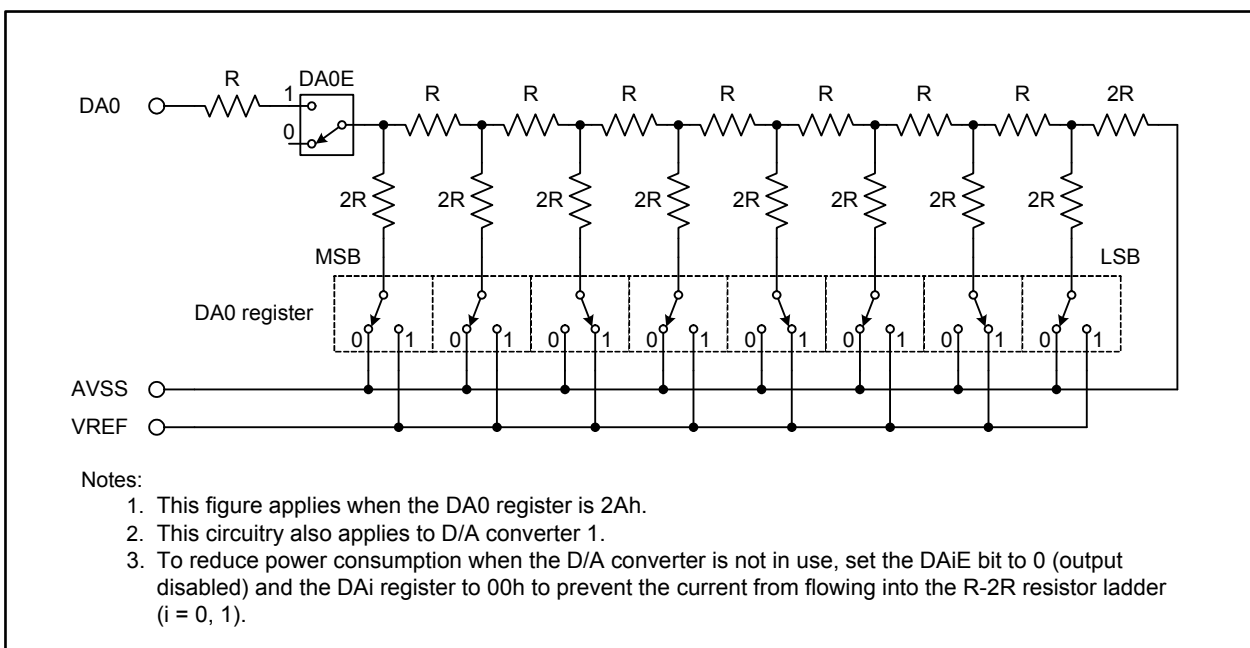
**Figure 20.1 D/A Converter Block Diagram**



**Figure 20.2 DACON Register**



**Figure 20.3 Registers DA0 and DA1**



**Figure 20.4 D/A Converter Equivalent Circuitry**

## 21. CRC Calculator

The Cyclic Redundancy Check (CRC) calculator is used for detecting errors in data blocks. A generator polynomial of CRC-CCITT ( $X^{16} + X^{12} + X^5 + 1$ ) generates the CRC.

The CRC is a 16-bit code generated for a given set of blocks of 8-bit data. It is set in the CRCD register every time 1-byte data is written to the CRCIN register after a default value is set to the CRCD register.

Figure 21.1 shows a block diagram of the CRC calculator. Figures 21.2 and 21.3 show registers associated with the CRC. Figure 21.4 shows an example of the CRC calculation.

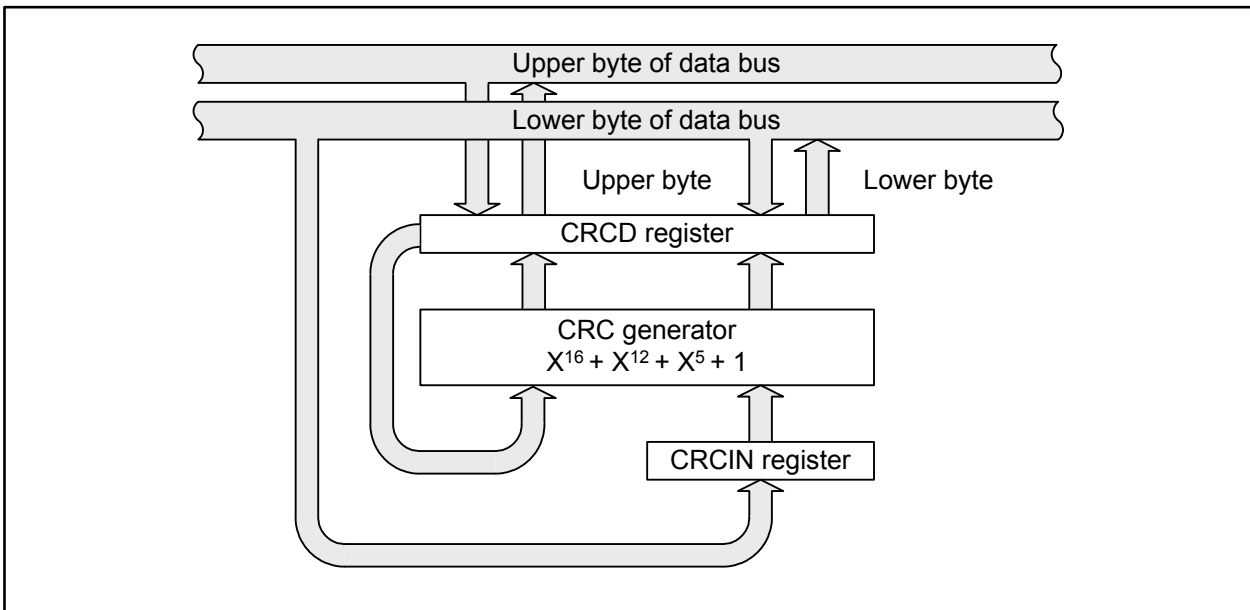


Figure 21.1 CRC Calculator Block Diagram

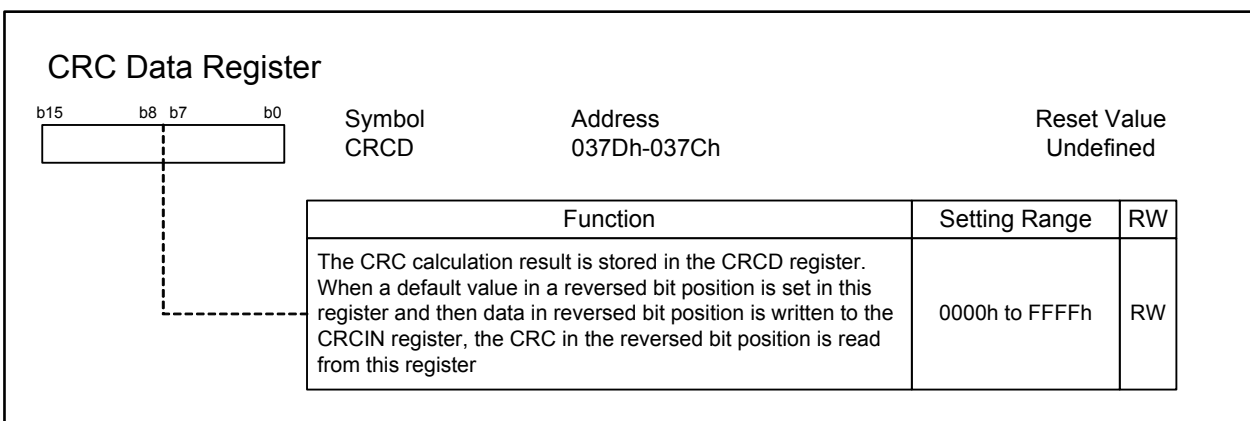
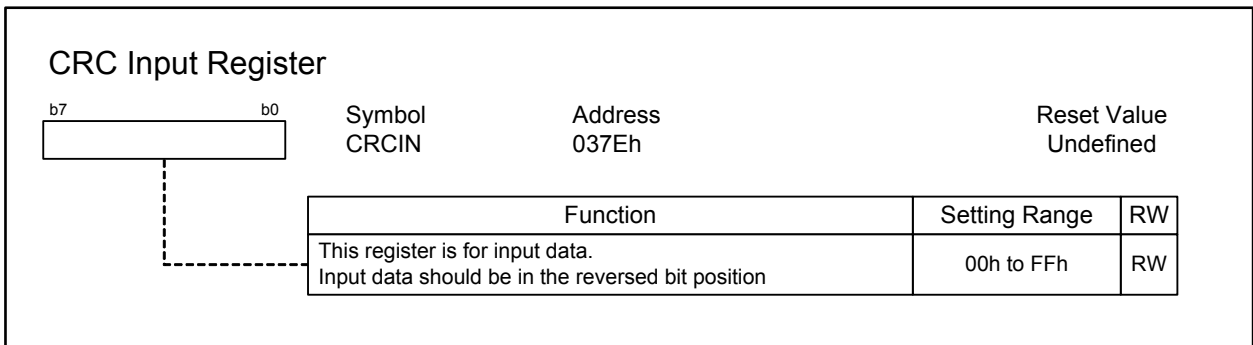


Figure 21.2 CRCD Register



**Figure 21.3 CRCIN Register**

### CRC Calculation and Setting Procedure to Generate CRC for 80C4h

- CRC Calculation for R32C

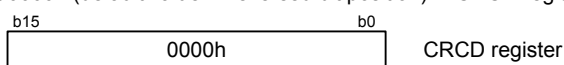
CRC: a remainder of the division as follows: 
$$\frac{\text{reversed-bit-position value in the CRCIN register}}{\text{generator polynomial}}$$

Generator Polynomial:  $X^{16} + X^{12} + X^5 + 1$  (1 0001 0000 0010 0001b)

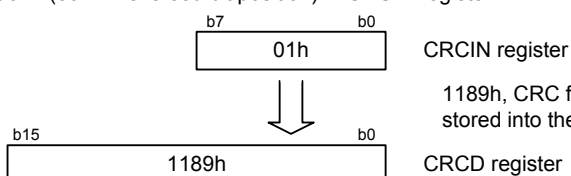
- Setting Procedure

(1) Reverse the bit position of 80C4h in 1-byte units by a program  
80h to 01h, C4h to 23h

(2) Set 0000h (default value in reversed bit position) in CRCD register

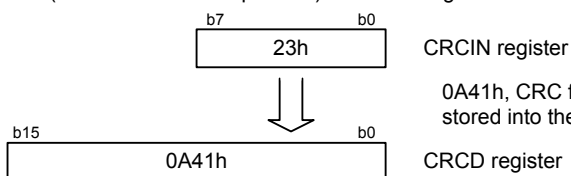


(3) Set 01h (80h in reversed bit position) in CRCIN register



1189h, CRC for 80h (9188h) in reversed bit position is stored into the CRCD register in the third cycle.

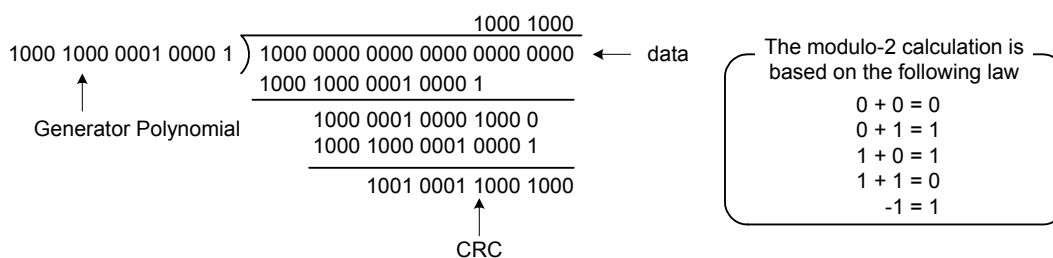
(4) Set 23h (C4h in reversed bit position) in CRCIN register



0A41h, CRC for 80C4h (8250h) in reversed bit position is stored into the CRCD register in the third cycle.

- Details of the CRC Calculation

As shown in (3) above, add 1000 0000 0000 0000 0000 0000b as 80h (1000 0000b) plus 16 digits to 0000 0000 0000 0000 0000 0000b as the default value of the CRCD register, 0000h plus eight digits to perform the modulo-2 division.



0001 0001 1000 1001b (1189h), the reversed-bit-position value of remainder 1001 0001 1000 1000b (9188h) can be read from the CRCD register.

When continuing on to (4) above, add 1100 0100 0000 0000 0000 0000b as C4h (1100 0100b) plus 16 digits to 1001 0001 1000 1000 0000 0000b as the remainder of (3) left in the CRCD register plus eight digits to perform the modulo-2 division. 0000 1010 0100 0001b (0A41h), the reversed-bit-position value of remainder 1000 0010 0101 0000b (8250h) can be read from the CRCD register.

Figure 21.4 CRC Calculation

## 22. X-Y Conversion

X-Y conversion rotates a 16 × 16-bit matrix data 90 degrees or reverses the bit position of 16-bit data.

X-Y conversion is set using the XYC register shown in Figure 22.1.

Data is written to the write-only XiR registers and converted data is read from the read-only YjR register (i = 0 to 15; j = 0 to 15). These registers are allocated to the same address. Figures 22.2 and 22.3 show registers XiR and YjR, respectively. A write/read access to registers XiR and YjR should be performed in 16-bit units from an even address. 8-bit access operation results are undefined.

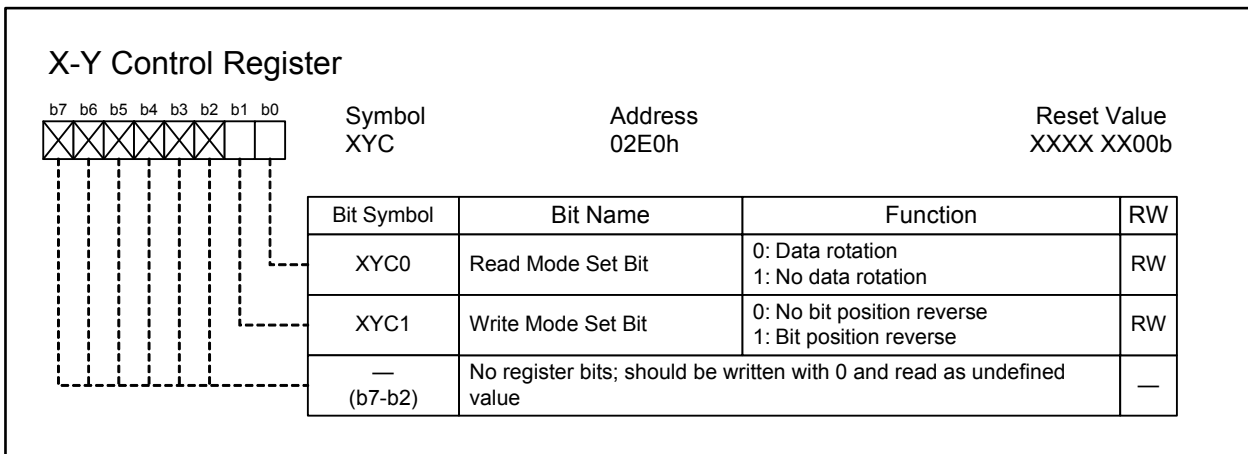


Figure 22.1 XYC Register

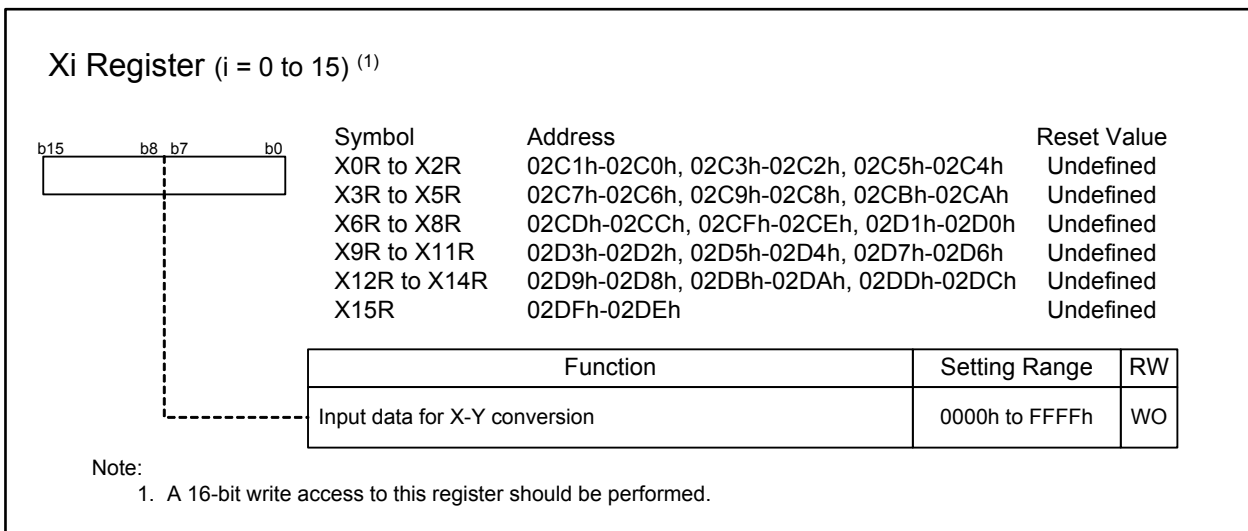
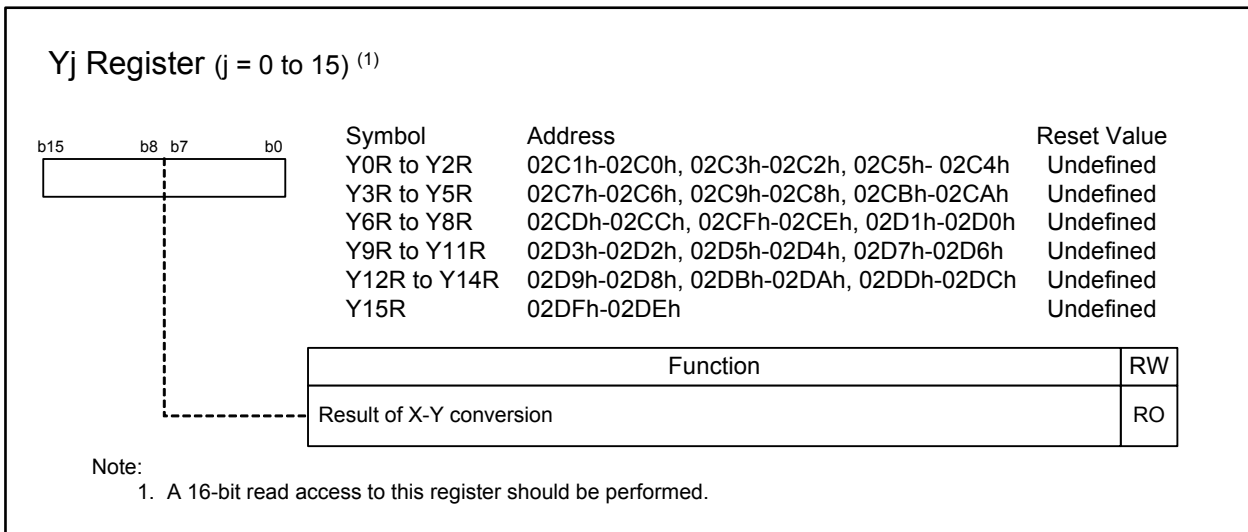


Figure 22.2 Registers X0R to X15R



**Figure 22.3 Registers Y0R to Y15R**

## 22.1 Data Conversion When Reading

Set the XYC0 bit in the XYC register to select a read mode for the Y<sub>j</sub>R register. When the XYC0 bit is 0 (data rotation), bit j in the corresponding registers X0R to X15R is automatically read upon reading the Y<sub>j</sub>R register (j = 0 to 15).

More concretely, upon reading bit i (i = 0 to 15) in the Y0R register, the data of bit 0 in the X<sub>i</sub>R register is read. That is, the read data of bit 0 in the Y15R register means the data of bit 15 in the X0R register and the data of bit 15 in the Y0R register is identical to that of bit 0 in the X15R register.

Figure 22.4 shows the conversion table when the XYC0 bit is 0 and Figure 22.5 shows an example of X-Y conversion.



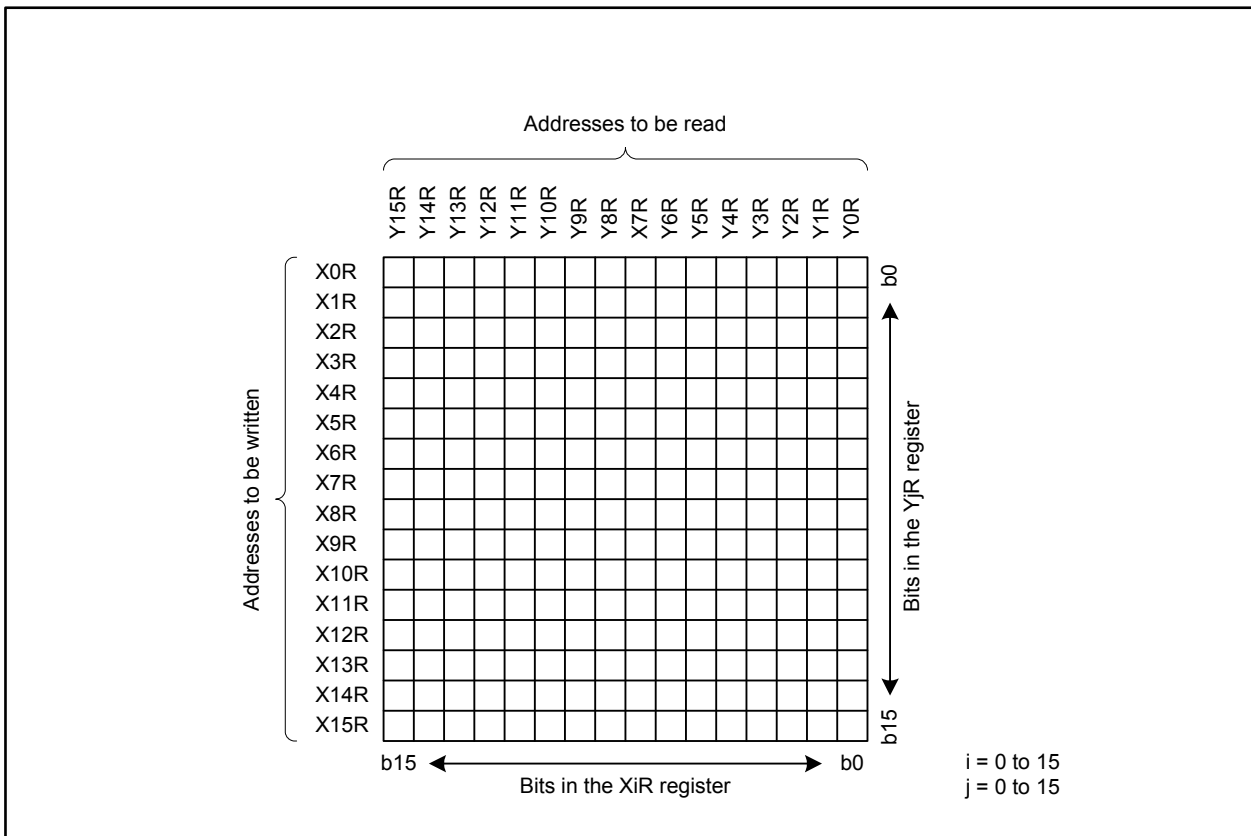


Figure 22.4 Conversion Table (XYC0 Bit is 0)

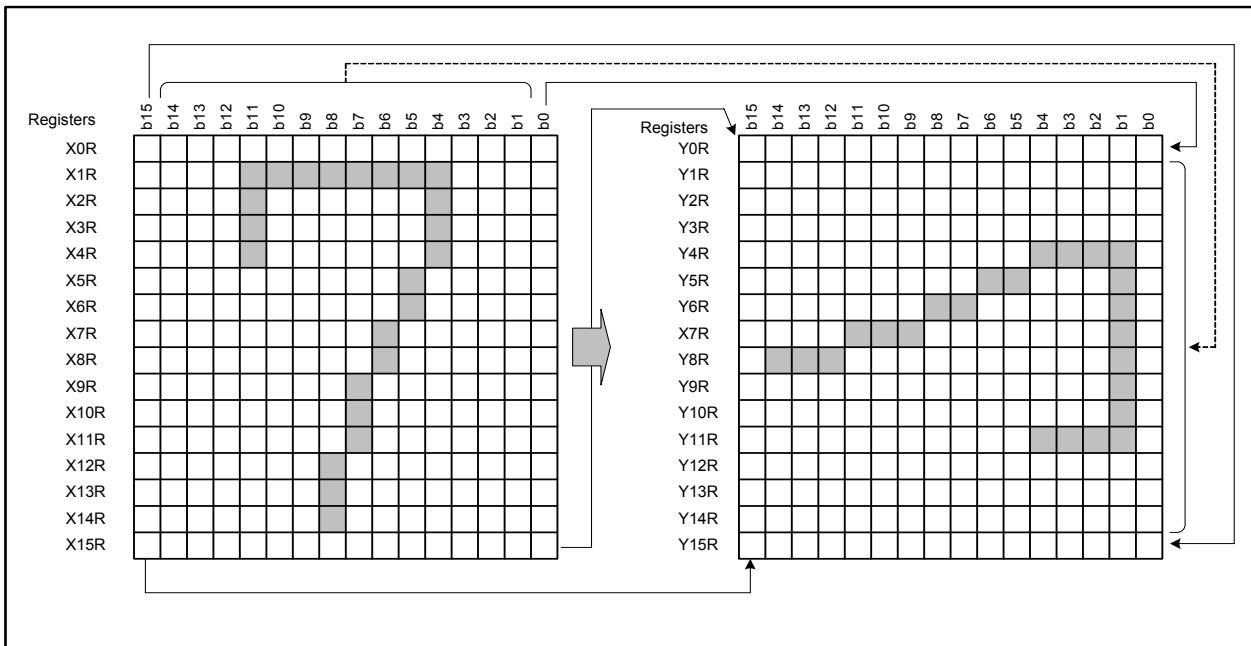


Figure 22.5 X-Y Conversion

When the XYC0 bit is set to 1 (no data rotation), the data of each bit in the YjR register is identical to that written in the XiR register. Figure 22.6 shows the conversion table when the XYC0 bit is set to 1.

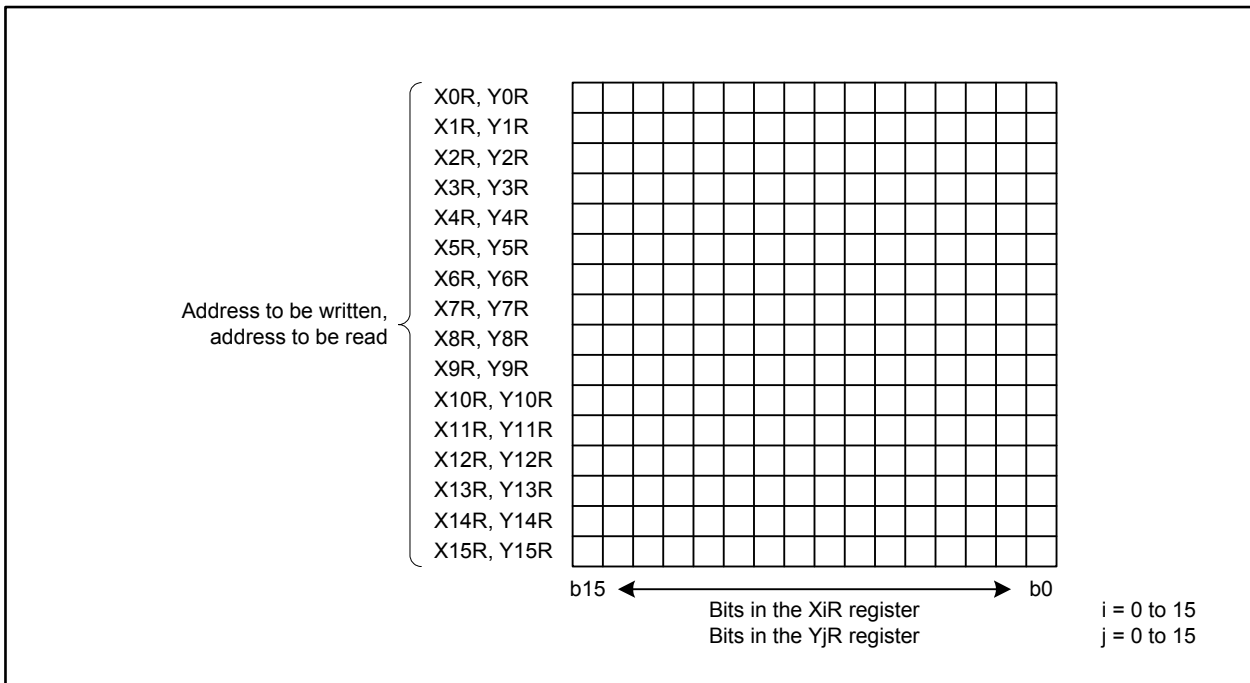


Figure 22.6 Conversion Table (XYC0 Bit is 1)

### 22.2 Data Conversion When Writing

Set the XYC1 bit in the XYC register to select a write mode for the XiR register.

When the XYC1 bit is set to 0 (no bit position reverse), the data is written in order. When it is set to 1 (bit position reverse), the data is written in reversed order. Figure 22.7 shows the conversion table when the XYC1 bit is set to 1.

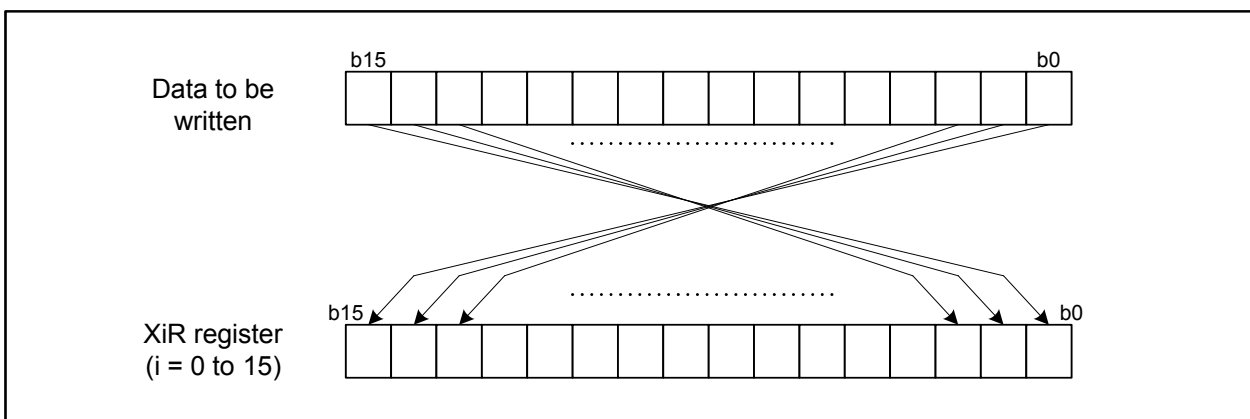


Figure 22.7 Conversion Table (XYC1 Bit is 1)

## 23. Intelligent I/O

The intelligent I/O is a multifunctional I/O port for time measurement, waveform generation, variable character length synchronous serial interface, and IEBus.

It consists of three groups each of which has one free-running 16-bit base timer and eight 16-bit registers for time measurement or waveform generation.

Table 23.1 lists the functions and channels of the intelligent I/O.

**Table 23.1 Intelligent I/O Functions and Channels**

Functions		Group 0	Group 1	Group 2
Time measurement (1)	Digital filter	8 channels	8 channels	Not available
	Prescaler	2 channels	2 channels	
	Gating	2 channels	2 channels	
Waveform generation (1)	Single-phase waveform output mode	8 channels	8 channels	3 channels
	Inverted waveform output mode	8 channels	8 channels	3 channels
	SR waveform output mode	8 channels	8 channels	3 channels
	Bit modulation PWM mode	Not available	Not available	3 channels
	RTP mode			3 channels
	Parallel RTP mode			3 channels
Serial interface	Variable character length synchronous serial interface mode	Not available	Not available	Available
	IEBus mode (optional (2))			

Notes:

1. The time measurement and waveform generation functions share a pin.
2. Contact a Renesas Electronics sales office to use the optional features.

Each channel can be individually assigned for time measurement or waveform generation function.

Figures 23.1 to 23.3 show block diagrams of the intelligent I/O.

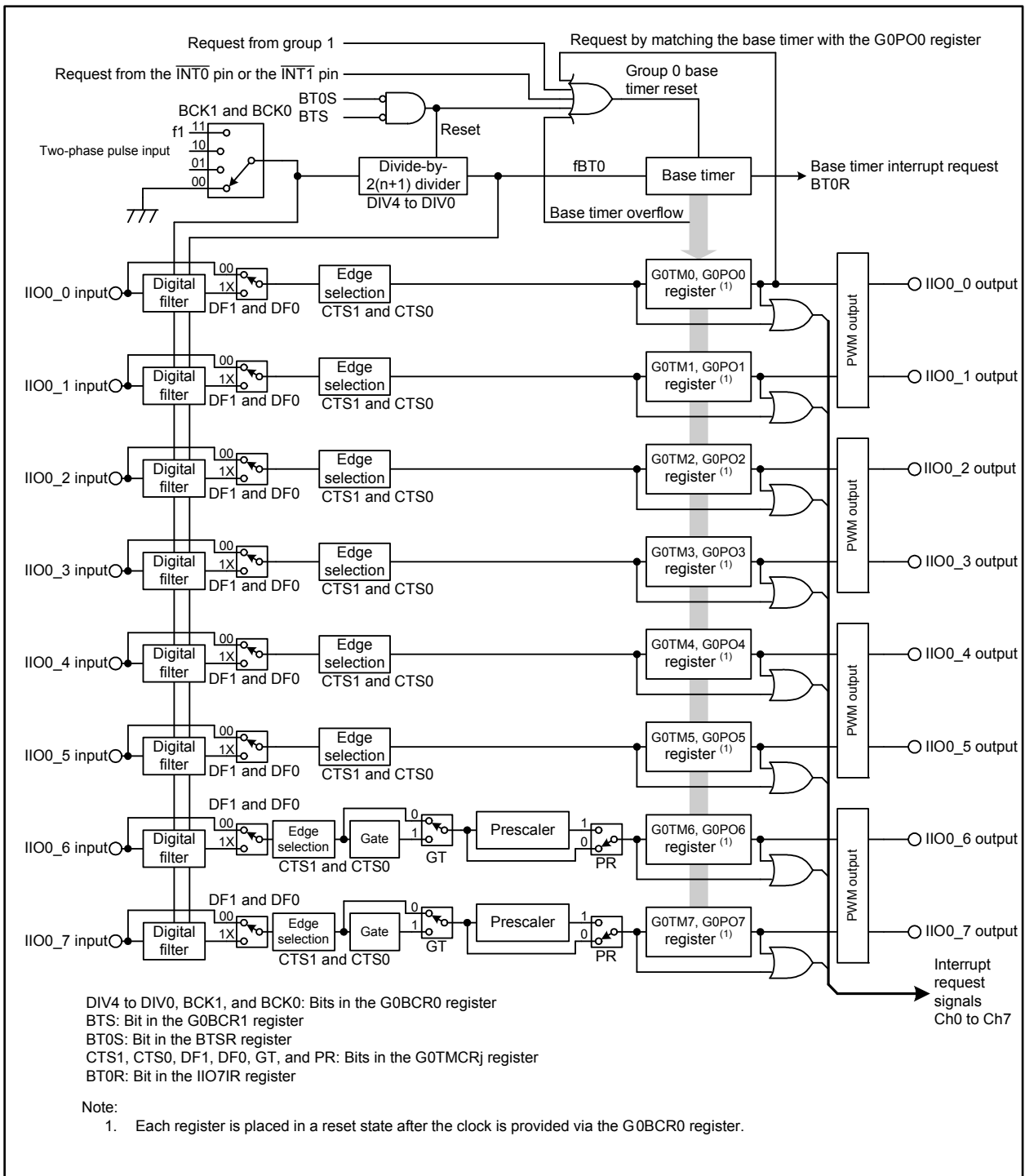


Figure 23.1 Intelligent I/O Group 0 Block Diagram (j = 0 to 7)

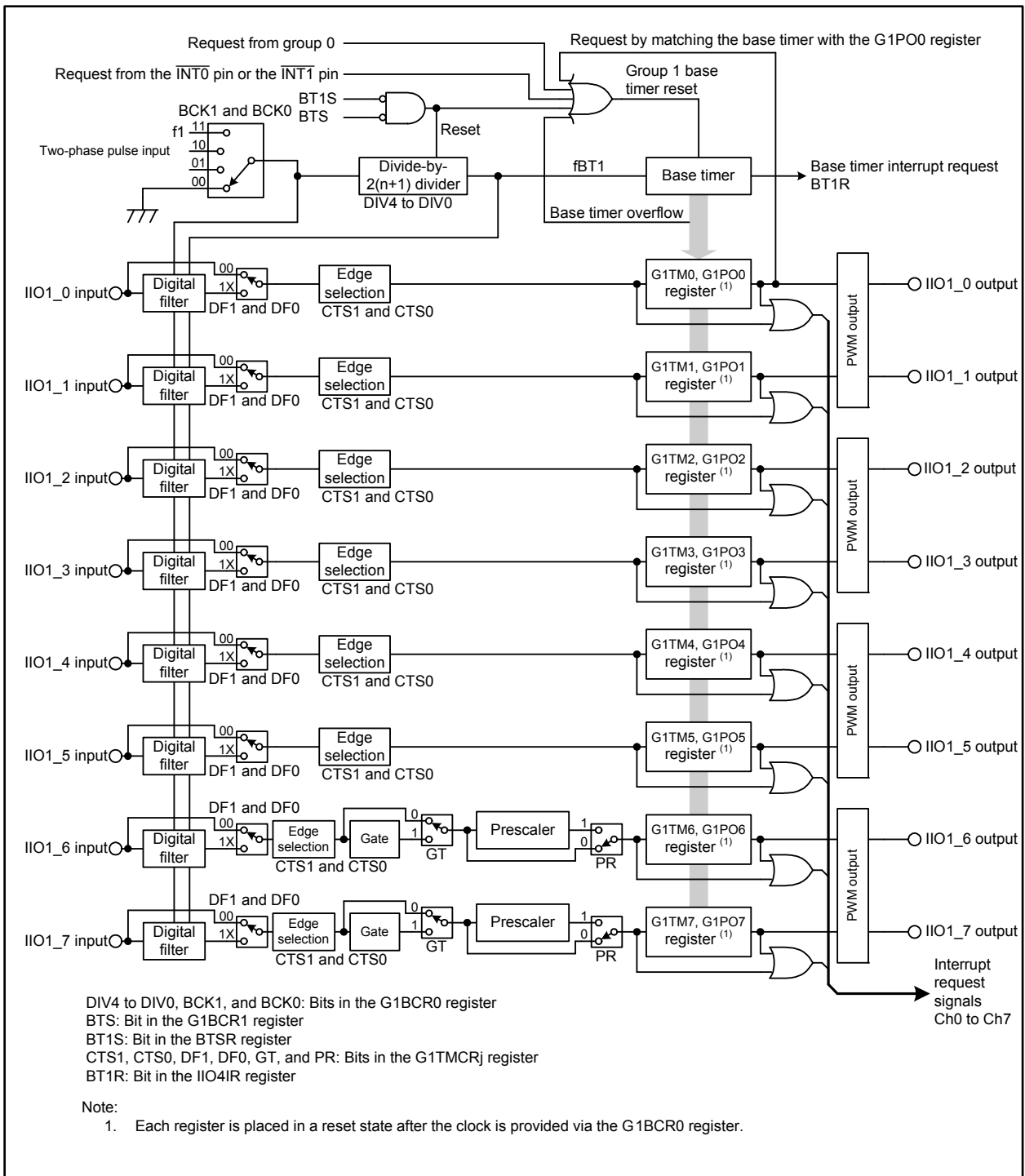


Figure 23.2 Intelligent I/O Group 1 Block Diagram (j = 0 to 7)

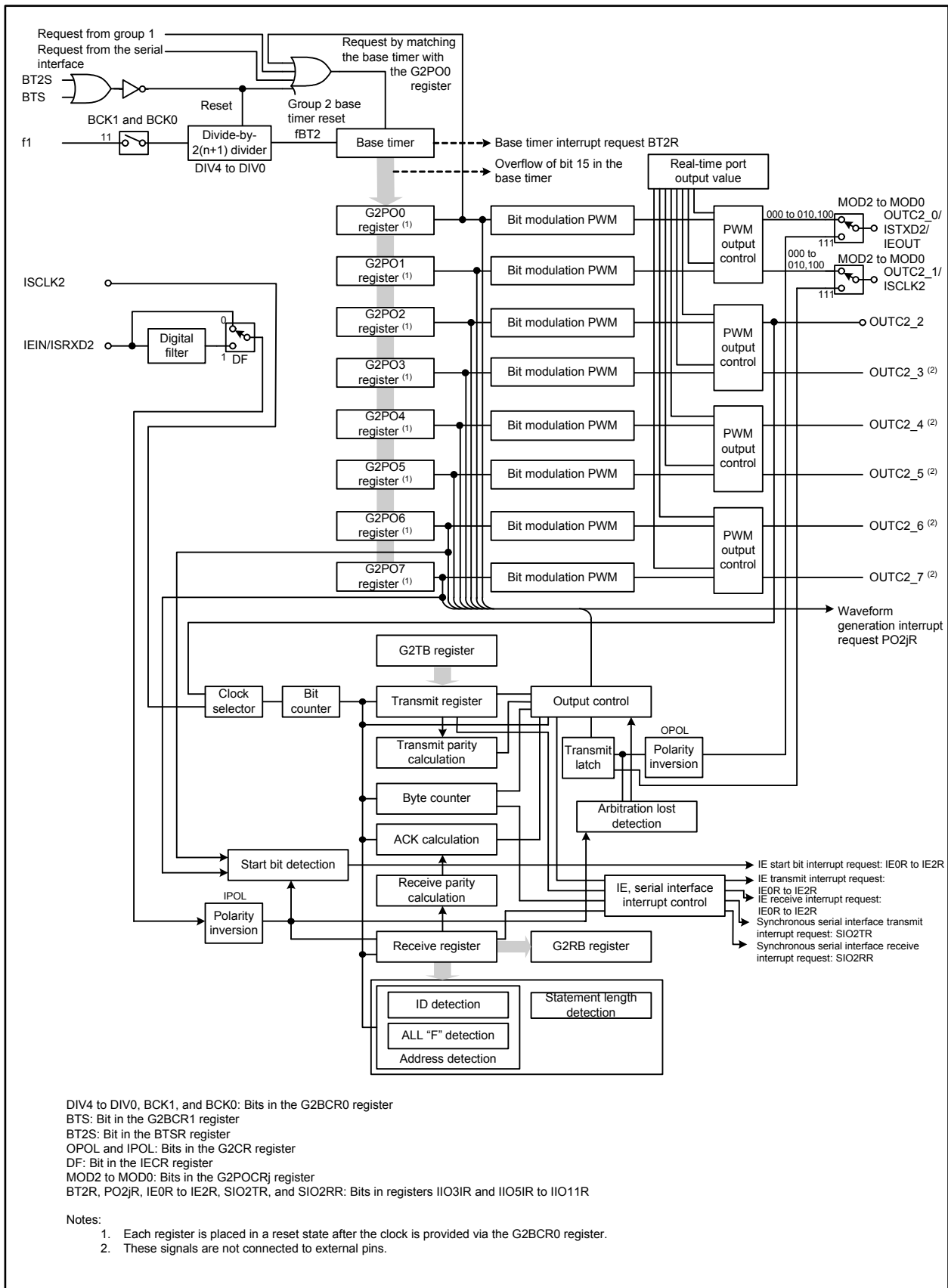
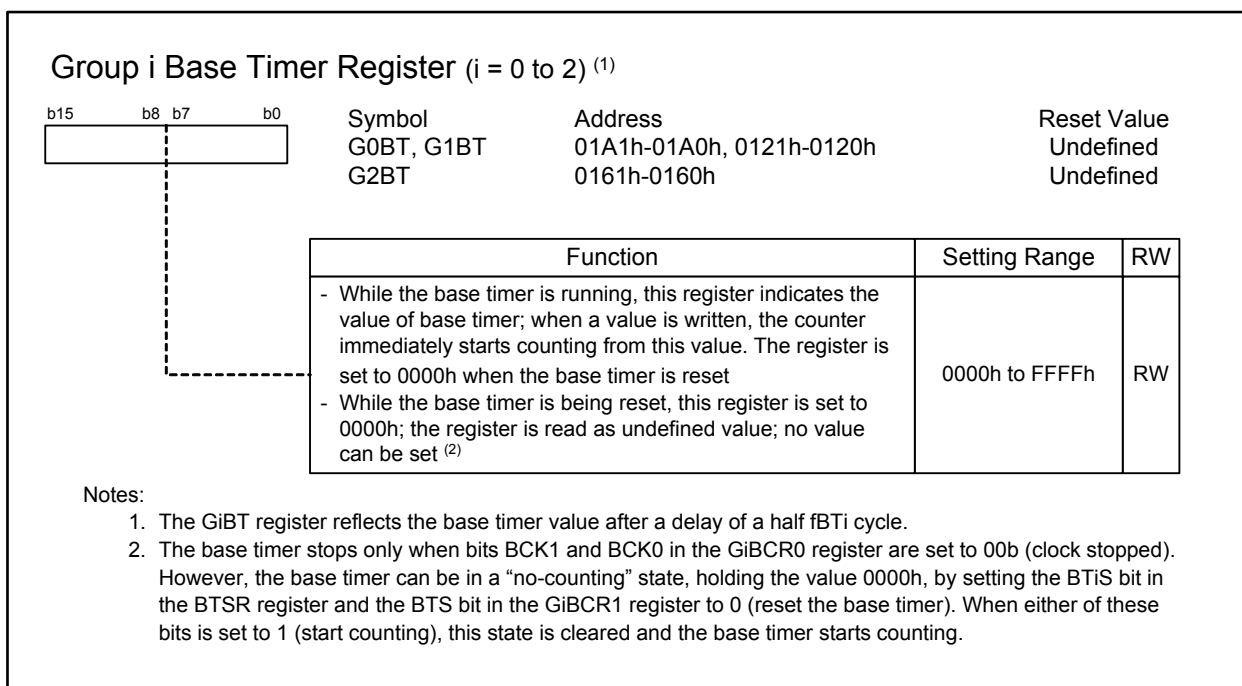
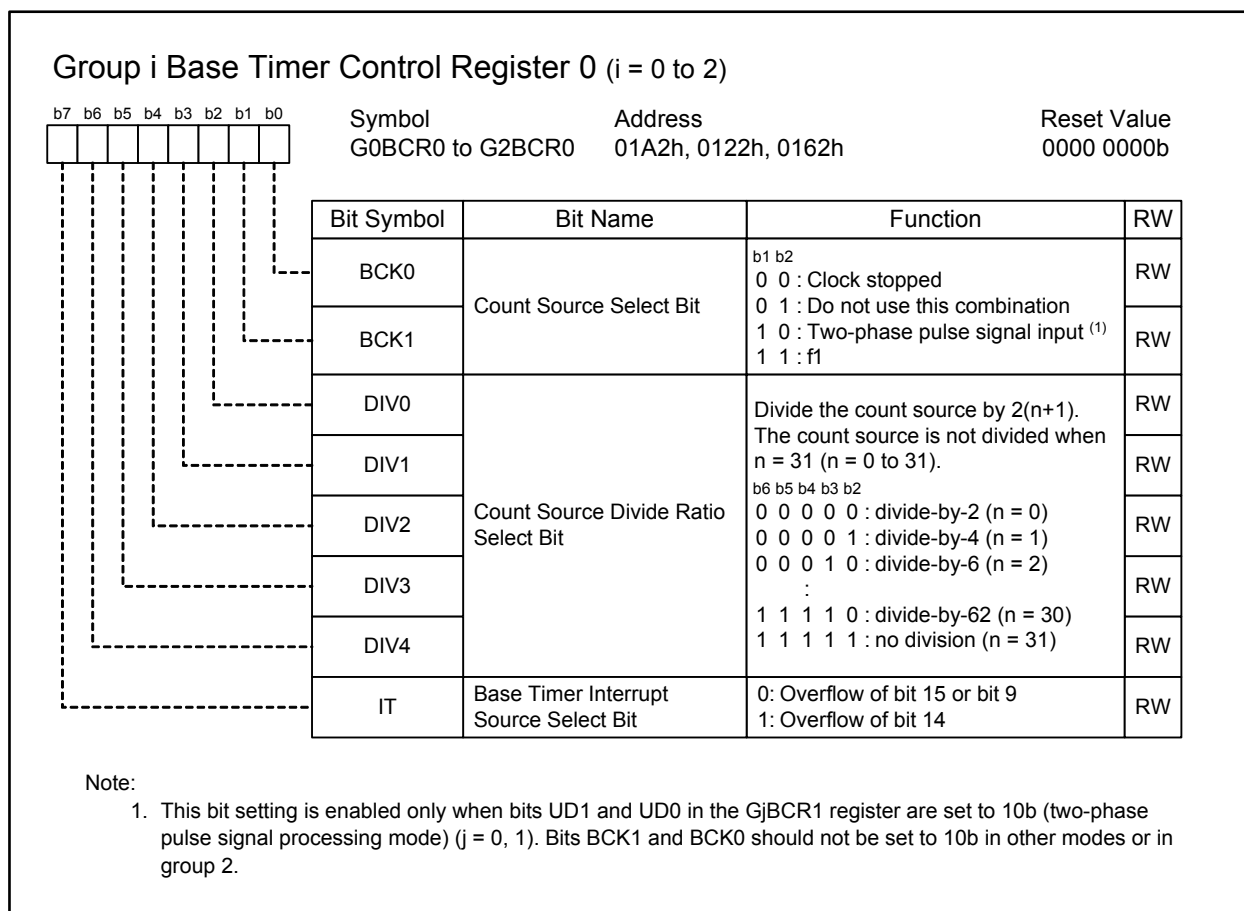


Figure 23.3 Intelligent I/O Group 2 Block Diagram (j = 0 to 7)

Figures 23.4 to 23.17 show registers associated with the intelligent I/O base timer, time measurement, and waveform generation (for registers associated with the serial interface, refer to Figures 23.33 to 23.40).

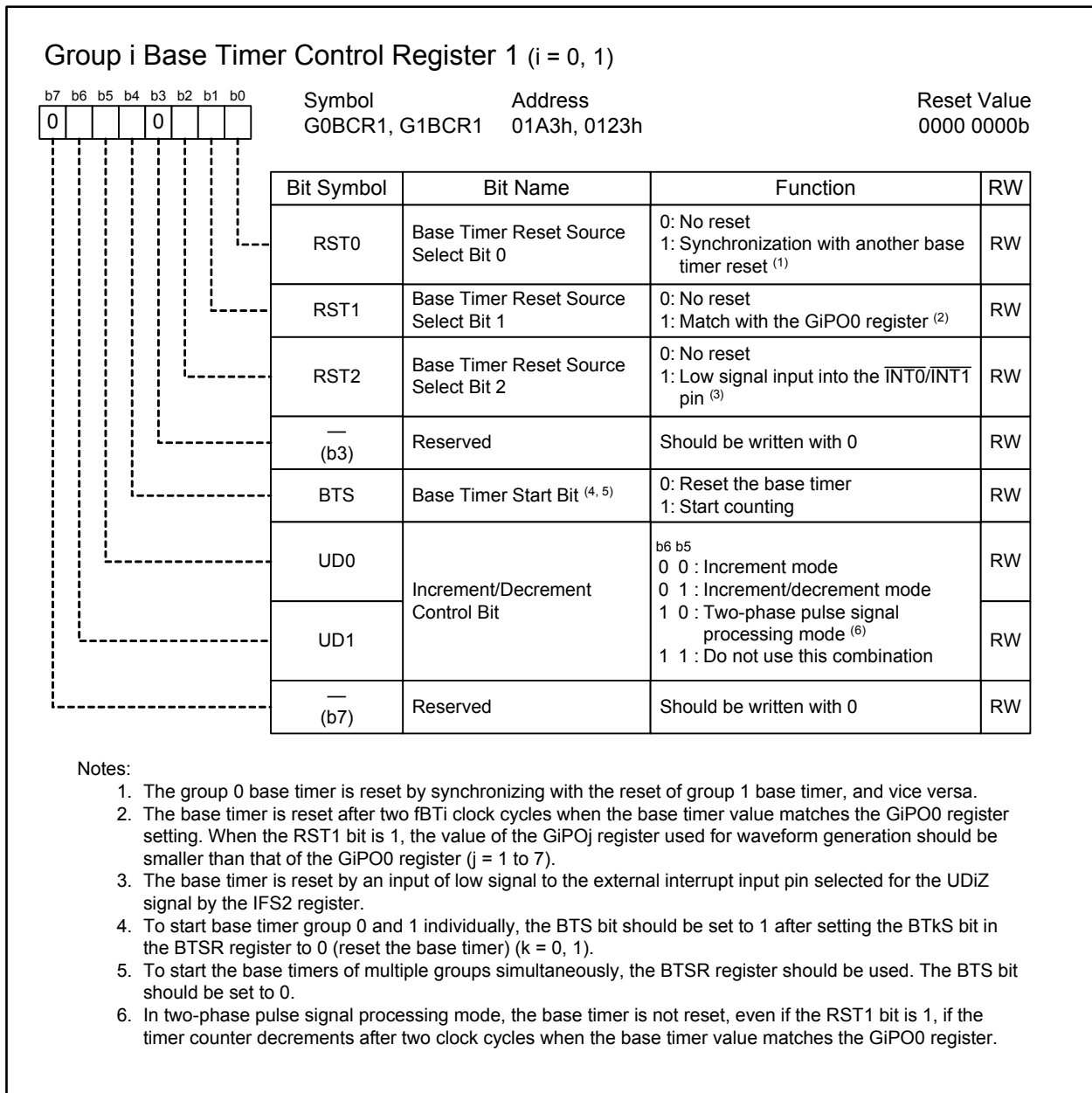


**Figure 23.4 Registers G0BT to G2BT**

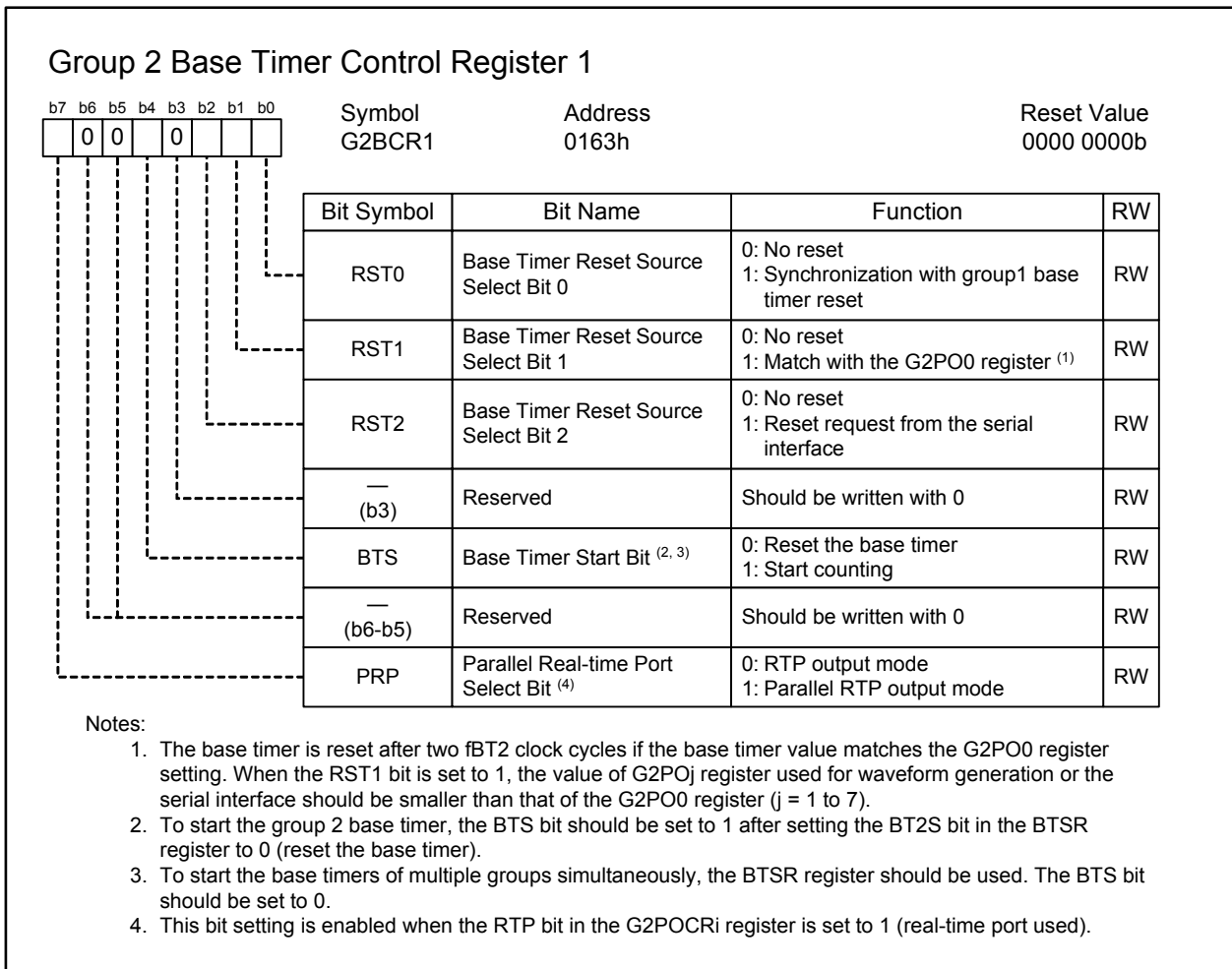


**Figure 23.5 Registers G0BCR0 to G2BCR0**

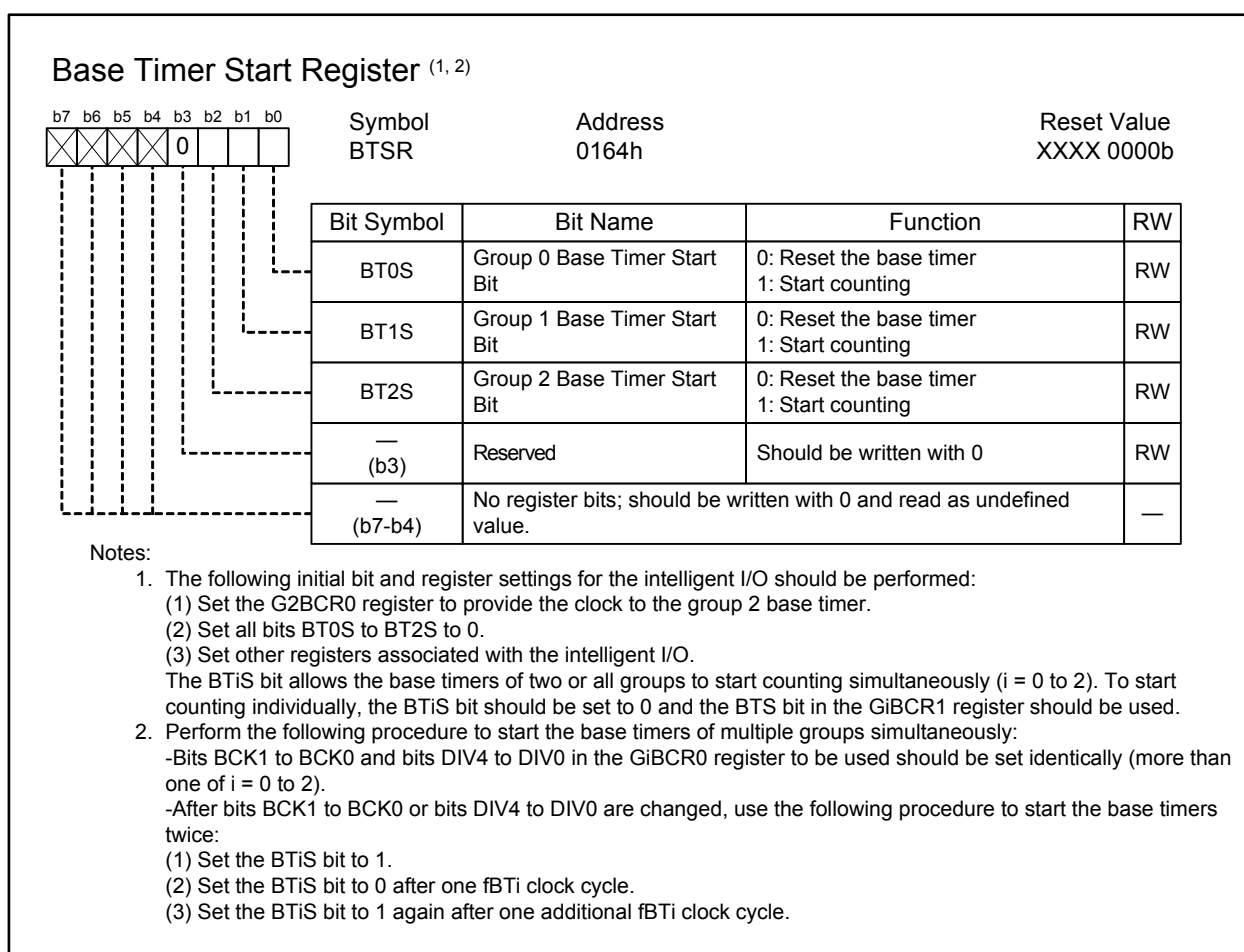




**Figure 23.6 Registers G0BCR1 and G1BCR1**



**Figure 23.7 G2BCR1 Register**

**Figure 23.8 BTISR Register**

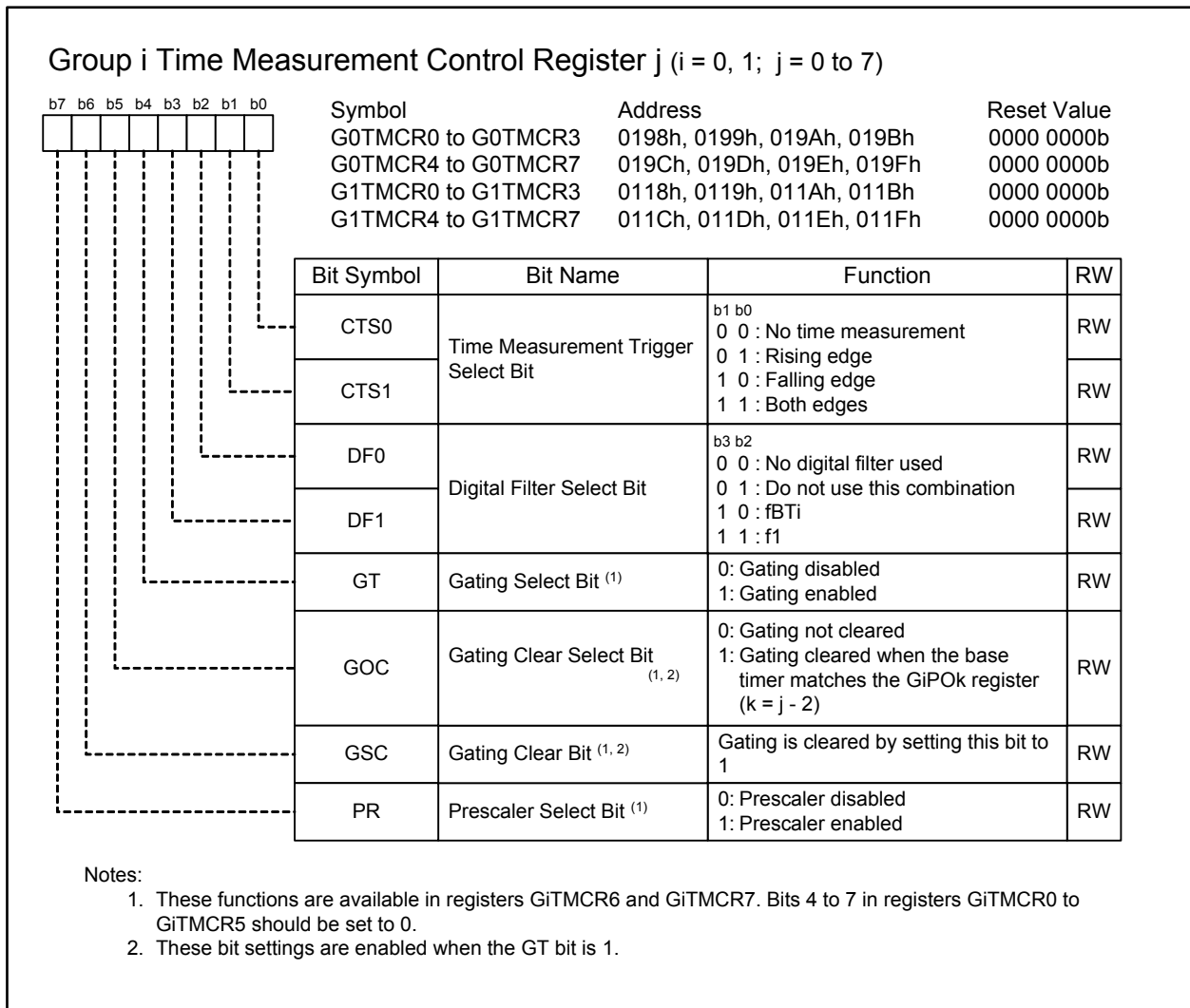


Figure 23.9 Registers G0TMCR0 to G0TMCR7 and G1TMCR0 to G1TMCR7

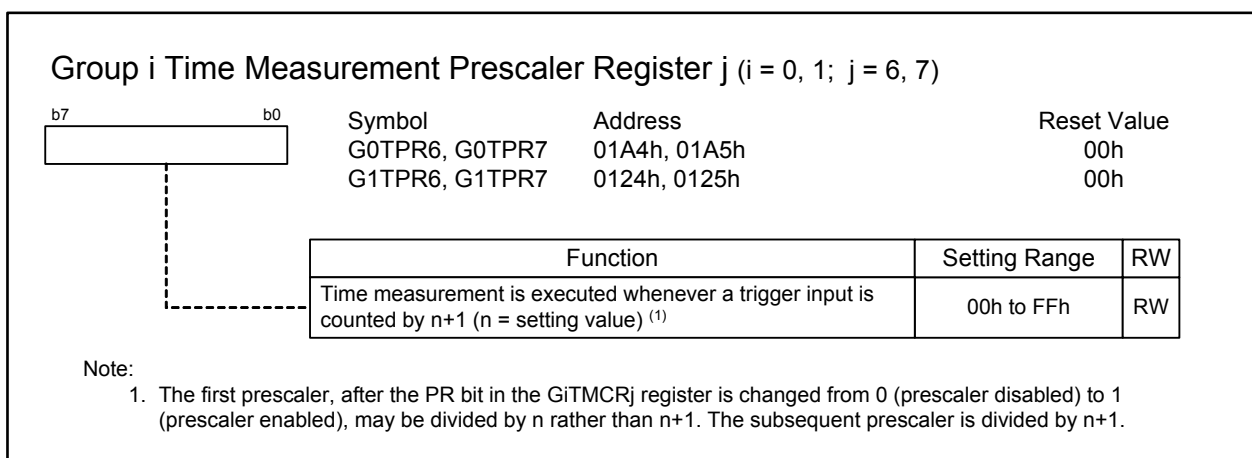


Figure 23.10 Registers G0TPR6, G0TPR7, G1TPR6, and G1TPR7

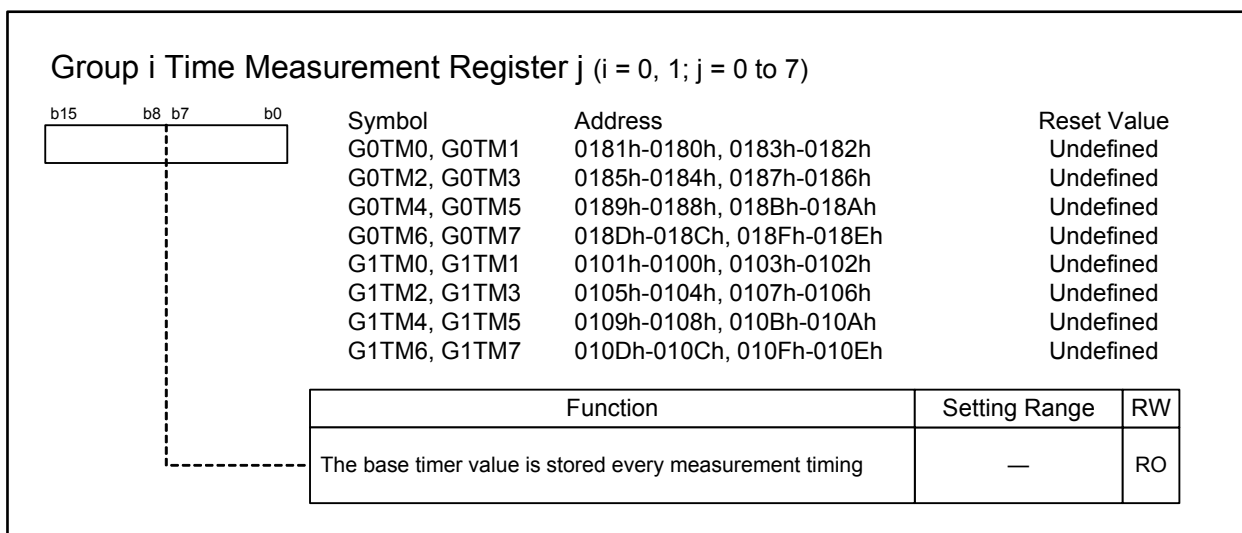


Figure 23.11 Registers G0TM0 to G0TM7 and G1TM0 to G1TM7

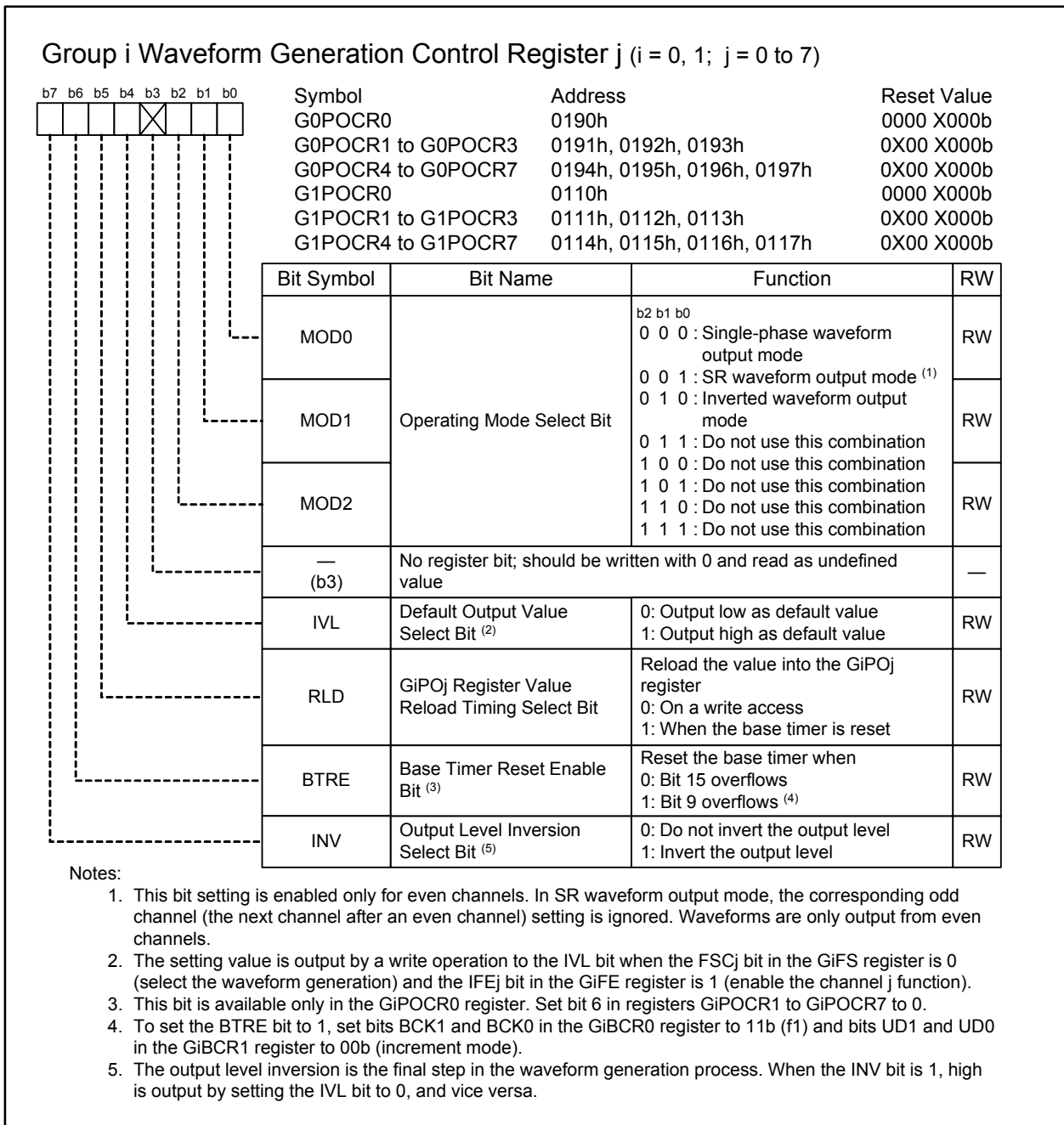


Figure 23.12 Registers G0POCR0 to G0POCR7 and G1POCR0 to G1POCR7

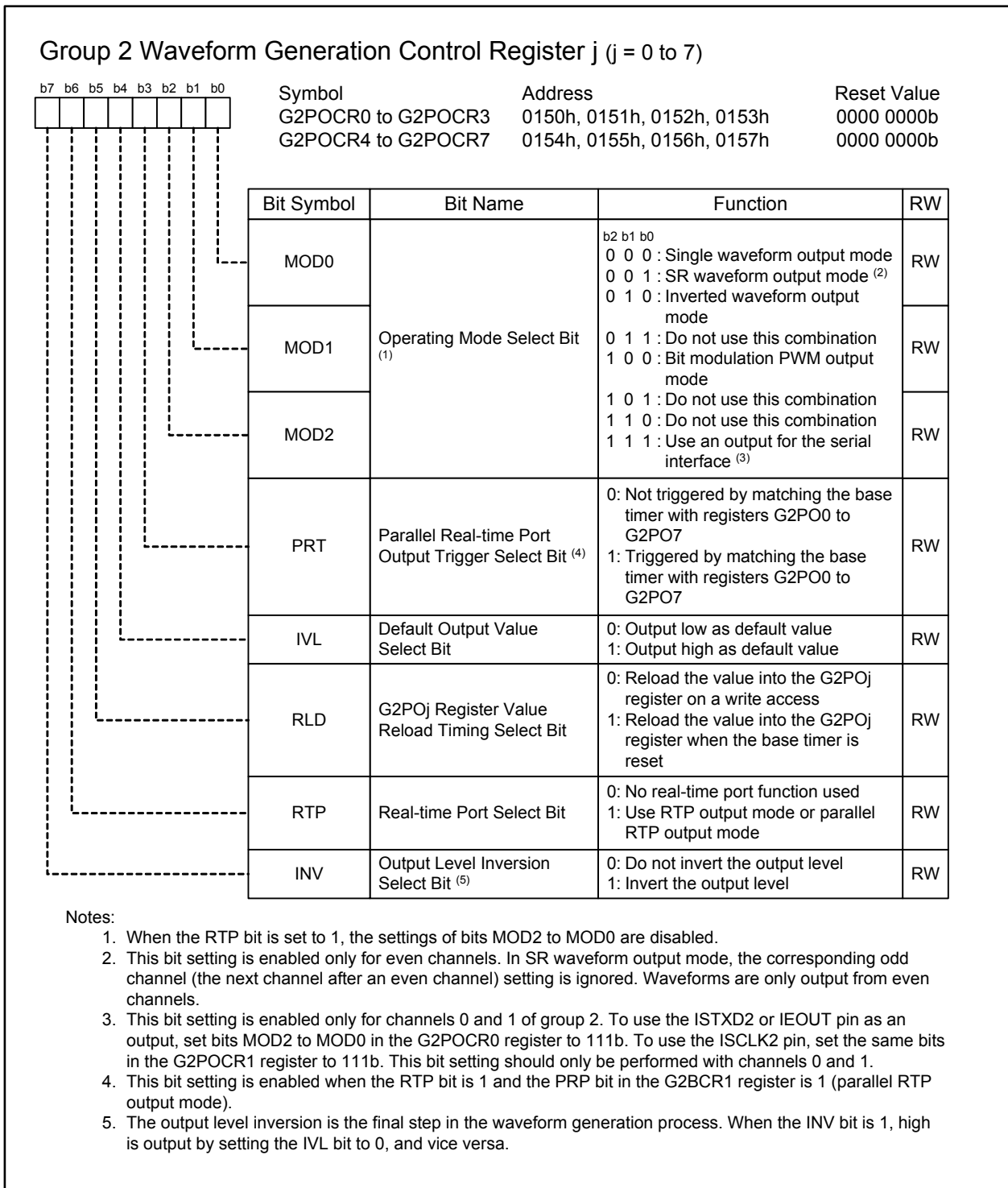
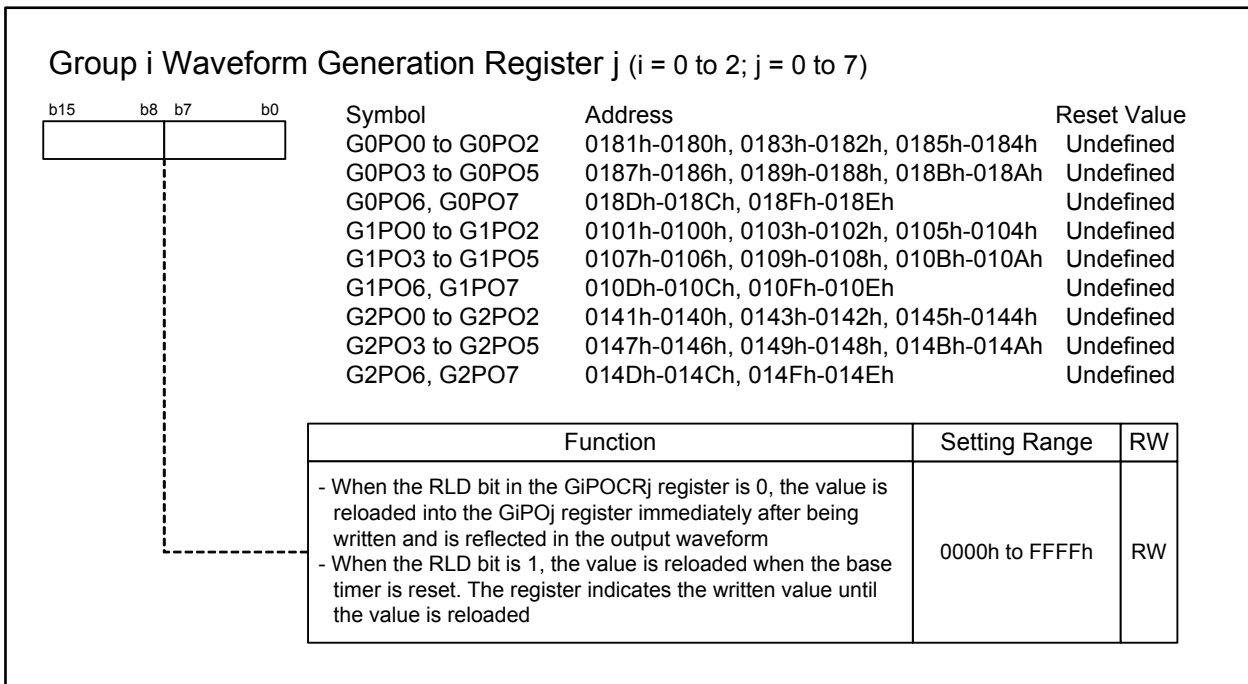
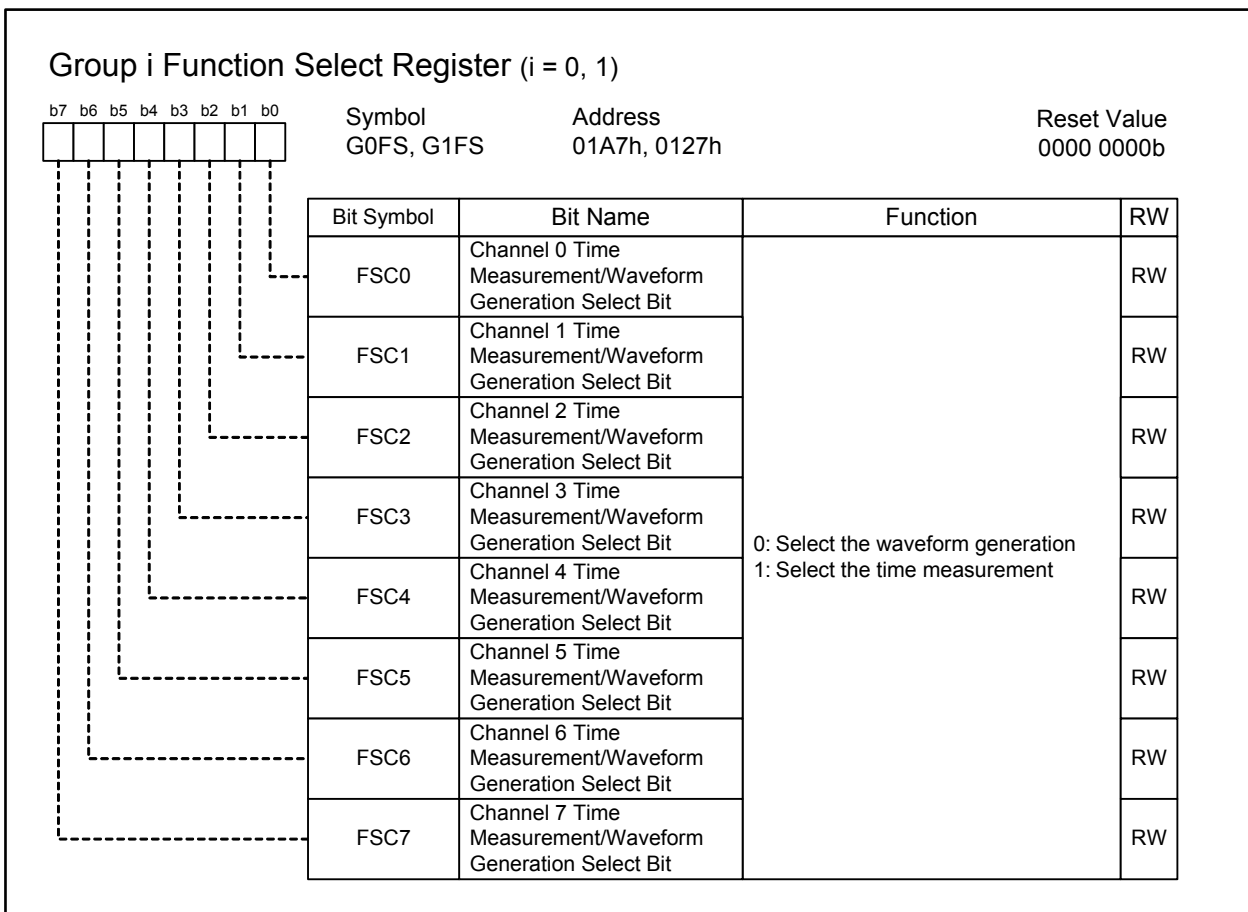


Figure 23.13 Registers G2POCR0 to G2POCR7



**Figure 23.14 Registers G0PO0 to G0PO7, G1PO0 to G1PO7, and G2PO0 to G2PO7**



**Figure 23.15 Registers G0FS and G1FS**



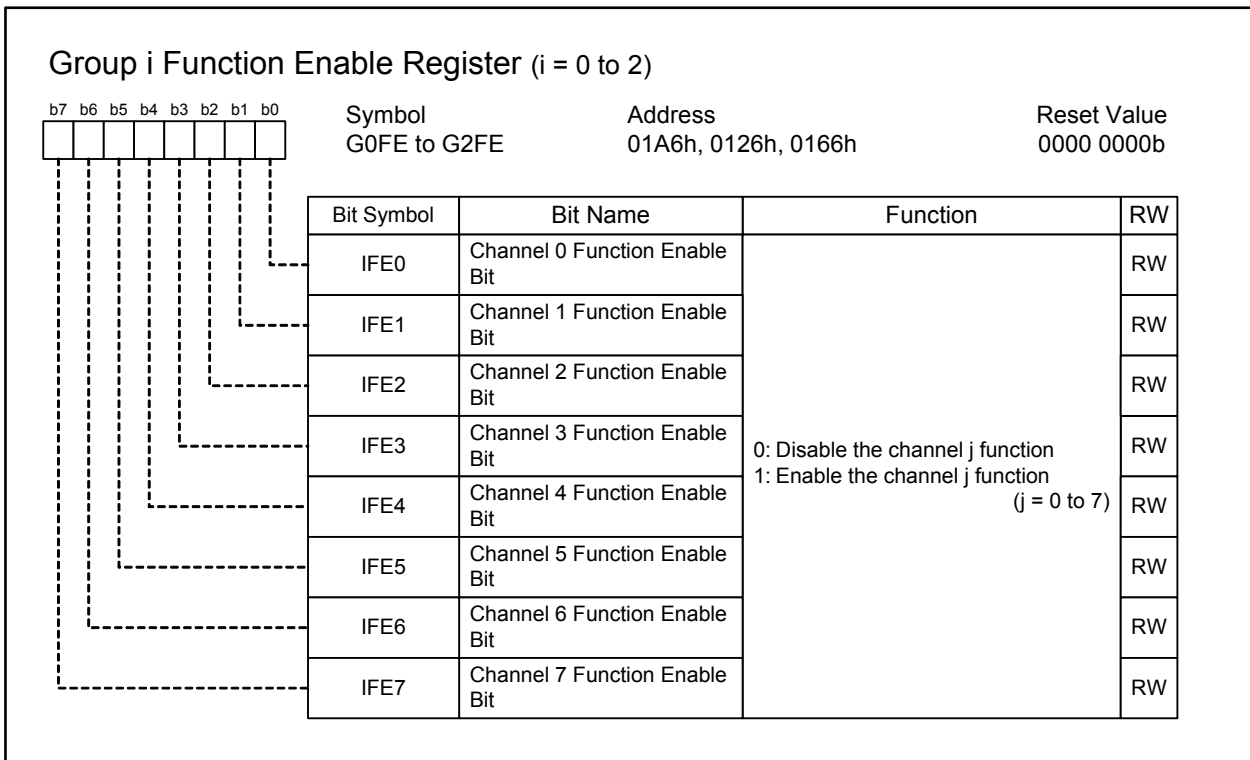


Figure 23.16 Registers G0FE to G2FE

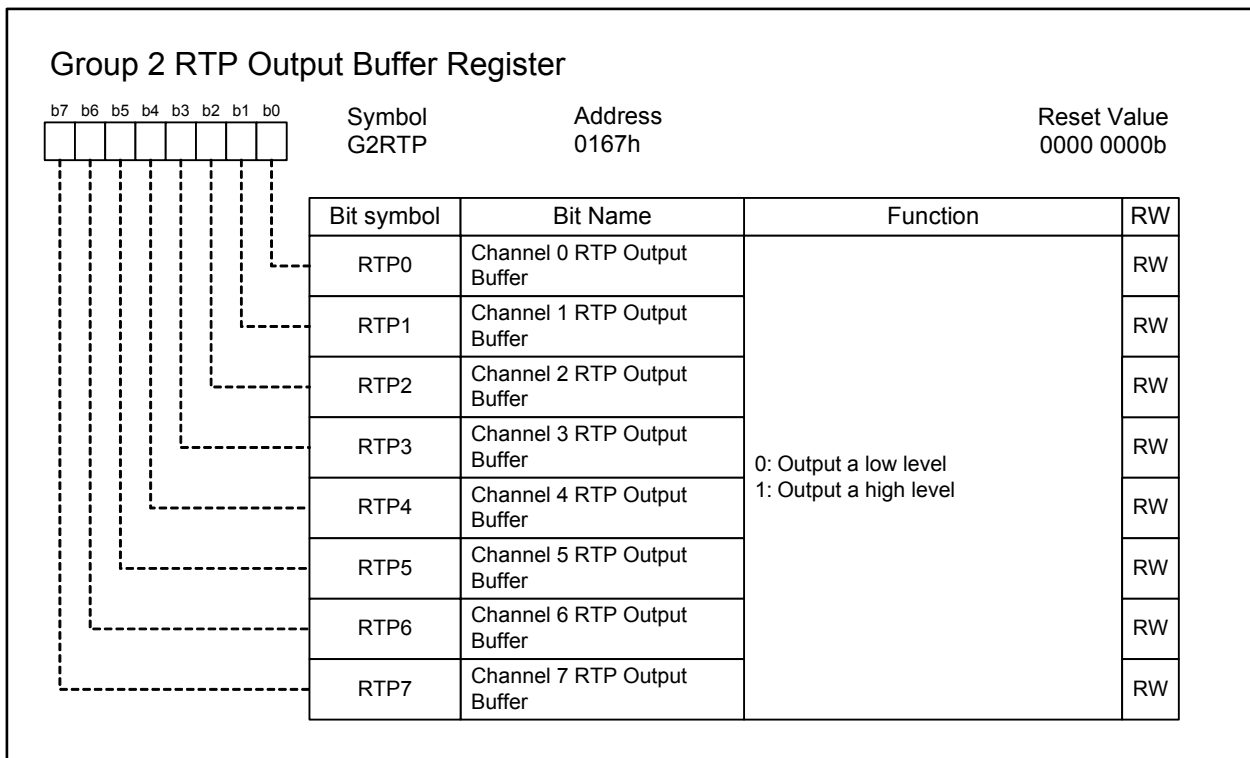


Figure 23.17 G2RTP Register

### 23.1 Base Timer for Groups 0 to 2

The base timer is a free-running counter that counts an internally generated count source. Table 23.2 lists specifications of the base timer. Figures 23.4 to 23.17 show registers associated with the base timer. Figure 23.18 shows a block diagram of the base timer. Figures 23.19, 23.20, and 23.21 show operation examples of the base timer for groups 0 and 1 in increment mode, increment/decrement mode, and two-phase pulse signal processing mode, respectively.

**Table 23.2 Base Timer Specifications (i = 0 to 2)**

Item	Specification
Count source (fBTi)	f1 divided by 2(n+1) for groups 0 to 2, two-phase pulse input divided by 2(n+1) for groups 0 and 1 n: setting value using bits DIV4 to DIV0 in the GiBCR0 register n = 0 to 31; however no division when n = 31
Count operations	<ul style="list-style-type: none"> <li>• Increment</li> <li>• Increment/decrement</li> <li>• Two-phase pulse signal processing</li> </ul>
Count start conditions	<ul style="list-style-type: none"> <li>• To start each base timer individually, The BTS bit in the GiBCR1 register is 1 (start counting)</li> <li>• To start the base timers of multiple groups simultaneously, The BTiS bit in the B TSR register is 1 (start counting)</li> </ul>
Count stop condition	The BTiS bit in the B TSR register and the BTS bit in the GiBCR1 register are 0 (reset the base timer)
Reset conditions	<ul style="list-style-type: none"> <li>• The base timer value matches the GiPO0 register setting</li> <li>• An input of low signal into the external interrupt pin (<math>\overline{\text{INT0}}</math> or <math>\overline{\text{INT1}}</math>) as follows: for group 0: selected using bits IFS23 and IFS22 in the IFS2 register for group 1: selected using bits IFS27 and IFS26 in the IFS2 register</li> <li>• The overflow of bit 15 or bit 9 in the base timer</li> <li>• The base timer reset request from the communication functions (group 2)</li> </ul>
Reset value	0000h
Interrupt request	When the BTiR bit in the interrupt request register becomes 1 (interrupt requested) by the overflow of bit 9, 14, or 15 in the base timer (refer to Figure 11.12)
Read from base timer	<ul style="list-style-type: none"> <li>• The GiBT register indicates a counter value while the base timer is running</li> <li>• The GiBT register is undefined while the base timer is being reset</li> </ul>
Write to base timer	When a value is written while the base timer is running, the timer counter immediately starts counting from this value. No value can be written while the base timer is being reset
Other functions	<ul style="list-style-type: none"> <li>• Increment/decrement mode for groups 0 and 1 The base timer starts counting when the BTS or BTiS bit is set to 1. When the base timer reaches FFFFh, it starts decrementing. When the RST1 bit in the GiBCR1 register is 1 (the base timer is reset by matching with the GiPO0 register), the timer counter starts decrementing two counts after the base timer value matches the GiPO0 register setting. When the timer counter reaches 0000h, it starts incrementing again (refer to Figure 23.20).</li> <li>• Two-phase pulse signal processing mode for groups 0 and 1 Two-phase pulse signals at pins UDiA and UDiB are counted (refer to Figure 23.21).</li> </ul> <div style="text-align: center;"> <p>UDiA</p> <p>UDiB</p> <p>The timer counter increments on all edges      The timer counter decrements on all edges</p> </div>

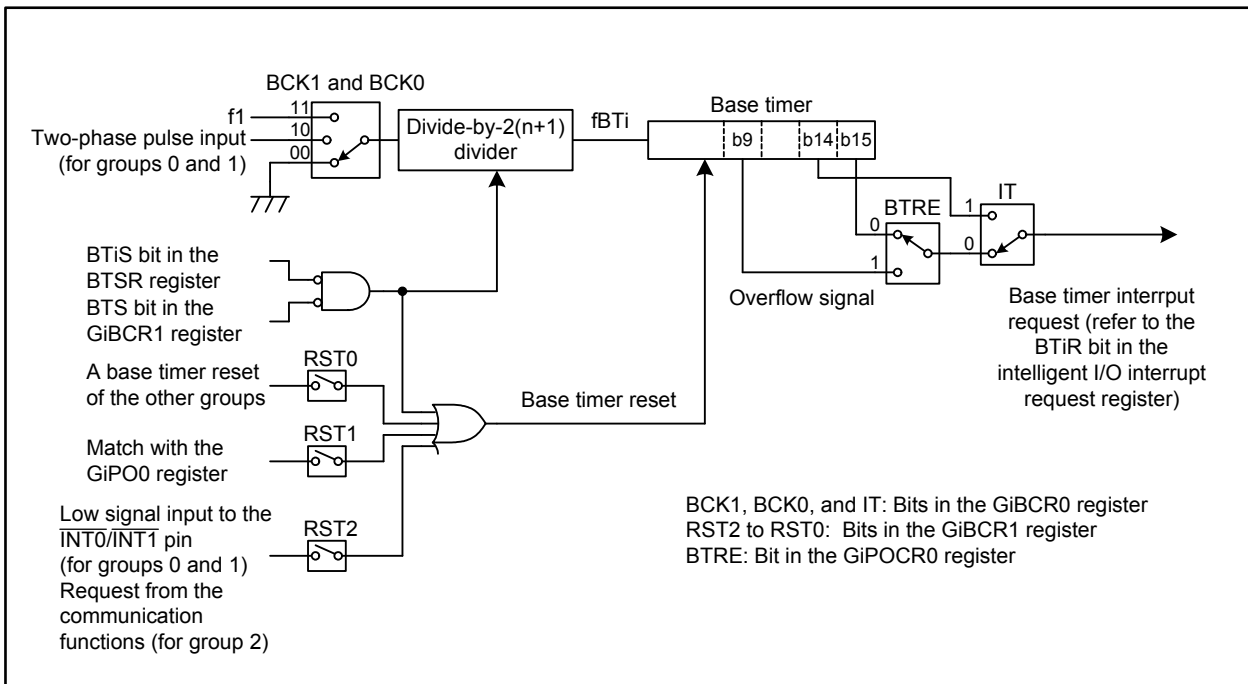


Figure 23.18 Base Timer Block Diagram (i = 0 to 2)

Table 23.3 Base Timer Associated Register Settings (Common Settings for Time Measurement, Waveform Generation, and Serial Interface) (i = 0 to 2)

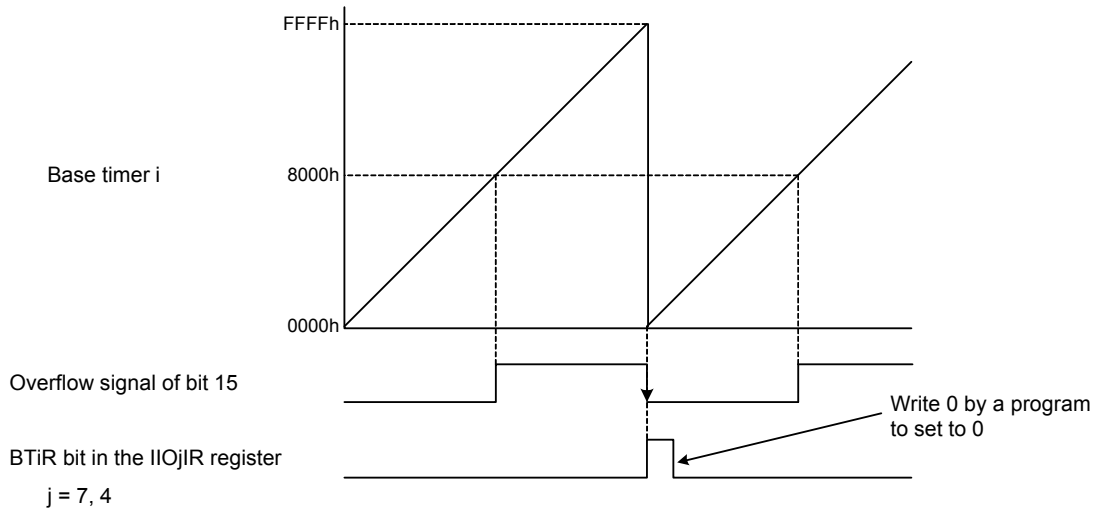
Register	Bits	Function
G2BCR0	—	Provide an operating clock to the B TSR register. Set to 0111 1111b
BTSR	—	Set to 0000 0000b
GiBCR0	BCK1 and BCK0	Select a count source
	DIV4 to DIV0	Select a count source divide ratio
	IT	Select a base timer interrupt source
GiBCR1	RST2 to RST0	Select a timing for base timer reset
	BTS	Use this bit when each base timer individually starts counting
	UD1 and UD0	Select a count mode in groups 0 and 1
GiPOCR0	BTRE	Select a source for base timer reset
GiBT	—	Read or write the base timer value

The following register settings are required to set the RST1 bit to 1 (the base timer is reset by matching with the GiPO0 register).

GiPOCR0	MOD2 to MOD0	Set to 000b (single-phase waveform output mode)
GiPO0	—	Set the reset cycle
GiFS	FSC0	Set the bit to 0 (select the waveform generation)
GiFE	IFE0	Set the bit to 1 (channel operation starts)

Bit configurations and functions vary by group.

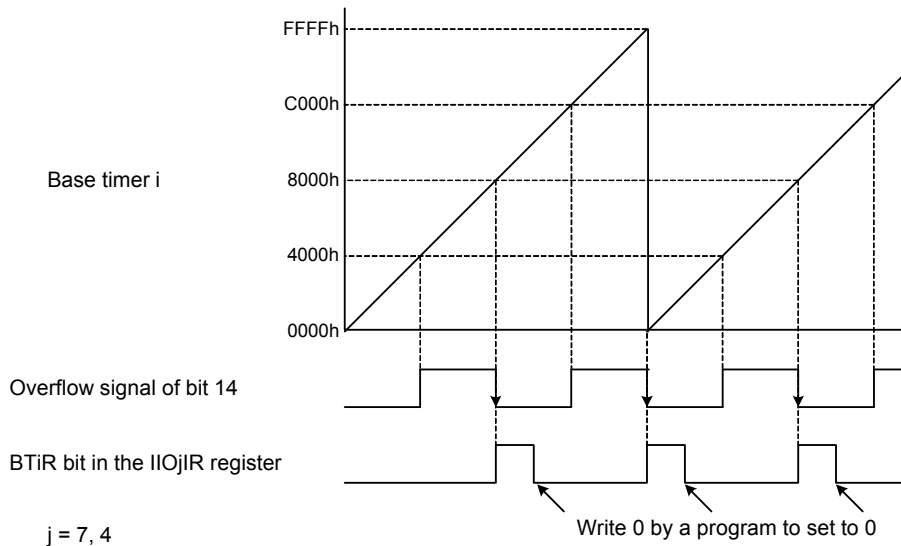
(A) When the IT bit in the GiBCR0 register is 0  
(an interrupt is requested by the overflow of bit 15 in the base timer)



This figure applies under the following conditions:

- The RST1 bit in the GiBCR1 register is 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are 00b (increment mode)

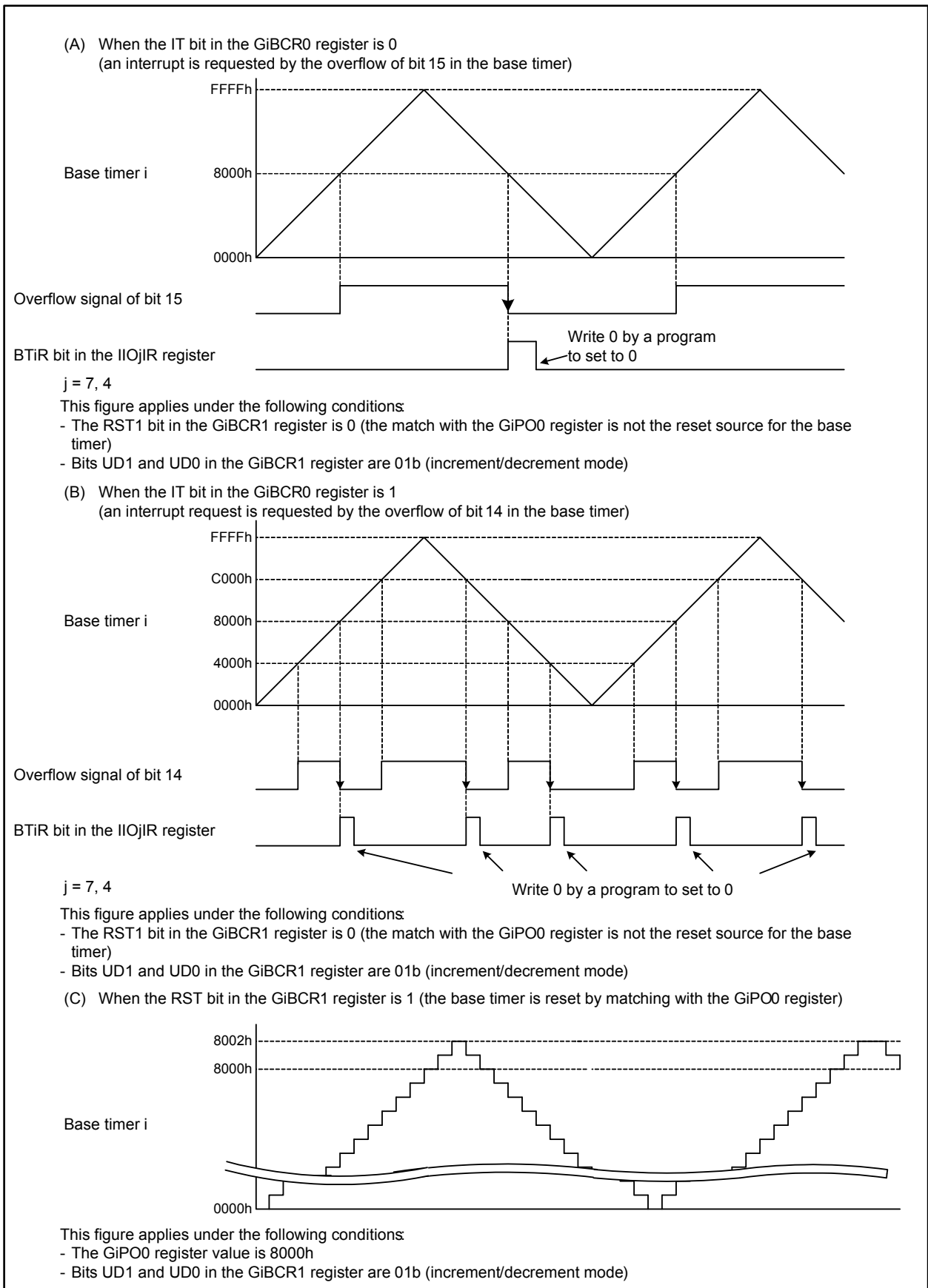
(B) When the IT bit in the GiBCR0 register is 1  
(an interrupt is requested by the overflow of bit 14 in the base timer)



This figure applies under the following conditions:

- The RST1 bit in the GiBCR1 register is 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are 00b (increment mode)

**Figure 23.19 Base Timer Increment Mode for Groups 0 and 1 ( $i = 0, 1$ )**



**Figure 23.20 Base Timer Increment/Decrement for Groups 0 and 1 ( $i = 0, 1$ )**

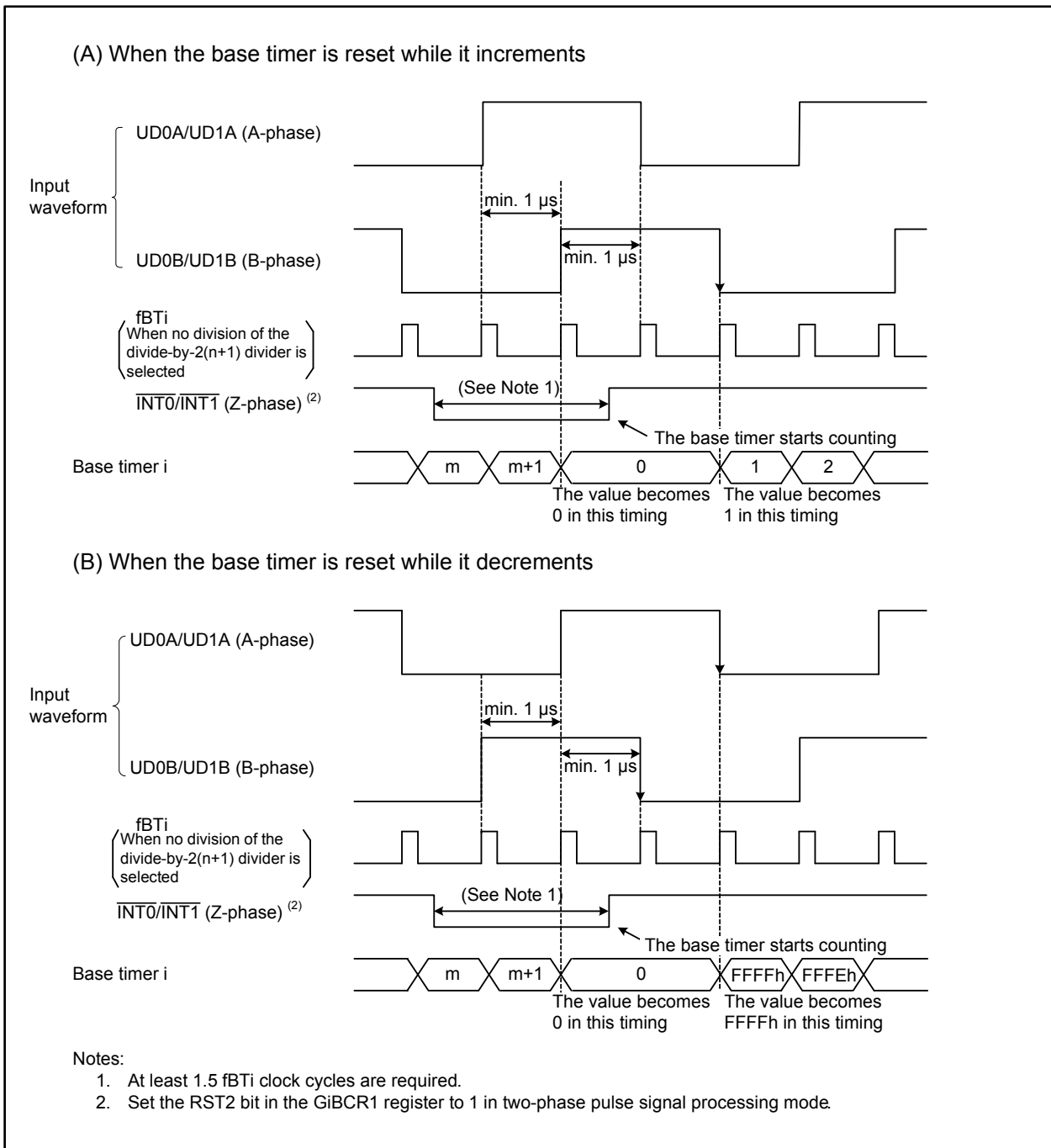


Figure 23.21 Base Timer Two-phase Pulse Signal Processing Mode for Groups 0 and 1 (i = 0, 1)

## 23.2 Time Measurement for Groups 0 and 1

Every time an external trigger is input, the base timer value is stored into the GiTMj register ( $i = 0, 1; j = 0$  to 7). Table 23.4 lists specifications of the time measurement and Table 23.5 lists its register settings. Figures 23.22 and 23.23 show operation examples of the time measurement and Figure 23.24 shows operation examples with the prescaler or gate function.

**Table 23.4 Time Measurement Specifications ( $i = 0, 1; j = 0$  to 7)**

Item	Specification
Time measurement channels	Group 0: Channels 0 to 7 Group 1: Channels 0 to 7
Trigger input polarity	Rising edge, falling edge, or both edges of the IIOi_j pin
Time measurement start condition	The IFEj bit in the GiFE register is 1 (enable the channel j function) while the FSCj bit in the GiFS register is 1 (select the time measurement)
Time measurement stop condition	The IFEj bit is 0 (disable the channel j function)
Time measurement timing	<ul style="list-style-type: none"> <li>• Without the prescaler: every time a trigger is input</li> <li>• With the prescaler for channels 6 and 7: every [GiTPRk register value + 1] times a trigger is input (<math>k = 6, 7</math>)</li> </ul>
Interrupt request	When the TMijR bit in the interrupt request register becomes 1 (interrupt requested) (refer to Figure 11.12)
IIOi_j input pin function	Trigger input
Other functions	<ul style="list-style-type: none"> <li>• Digital filter The digital filter determines a trigger input level every f1 or fBTi cycle and passes the signals holding the same level during three sequential cycles</li> <li>• Prescaler for channels 6 and 7 Time measurement is executed every [GiTPRk register value + 1] times a trigger is input</li> <li>• Gating for channels 6 and 7 This function disables any trigger input to be accepted after the time measurement by the first trigger input. However, the trigger input can be accepted again if any of following conditions are met while the GOC bit in the GiTMCRk register is 1 (the gating is cleared when the base timer matches the GiPOp register) (<math>p = 4, 5; p = 4</math> when <math>k = 6; p = 5</math> when <math>k = 7</math>): <ul style="list-style-type: none"> <li>• The base timer value matches the GiPOp register setting</li> <li>• The GSC bit in the GiTMCRk register is 1</li> </ul> </li> </ul>

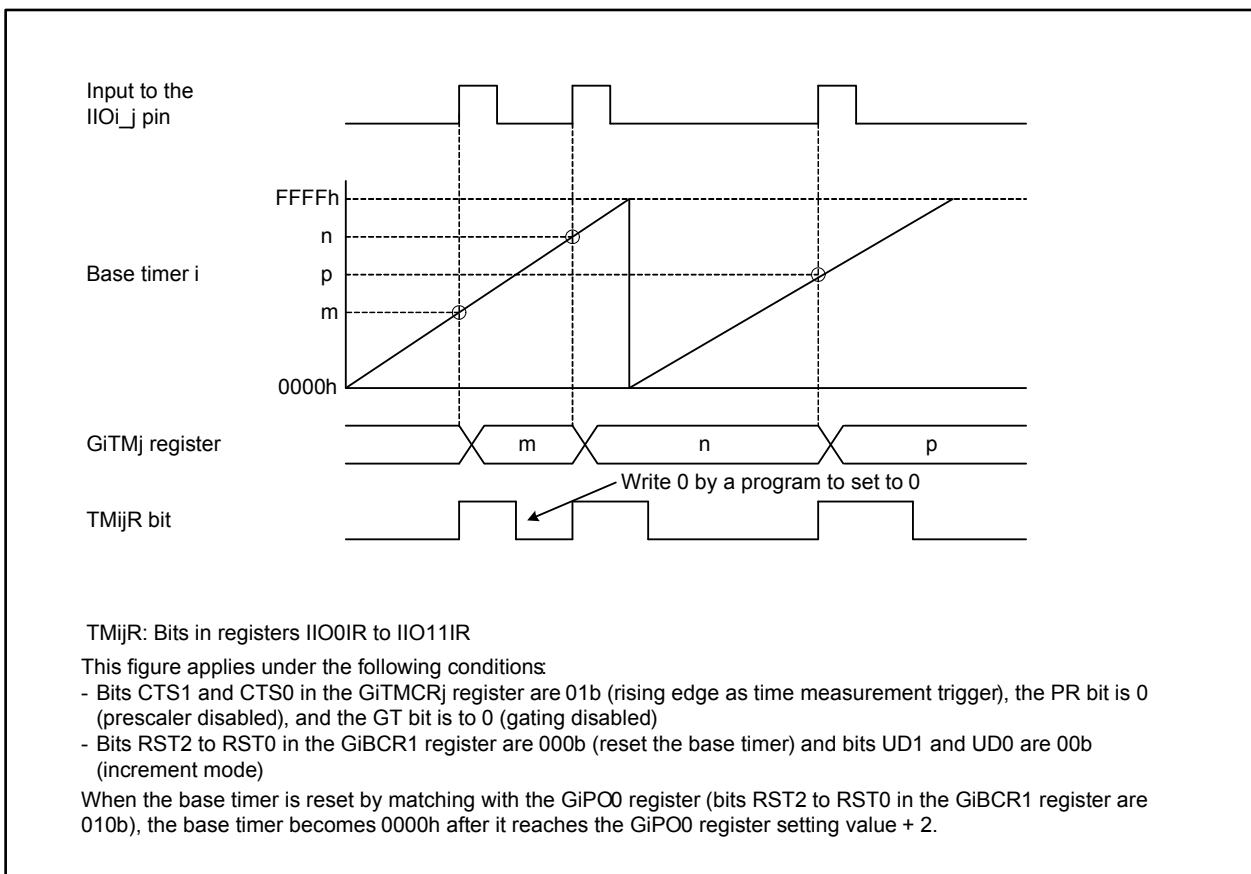


**Table 23.5 Time Measurement (for Groups 0 and 1) Associated Register Settings (i = 0, 1; j = 0 to 7; k = 6, 7)**

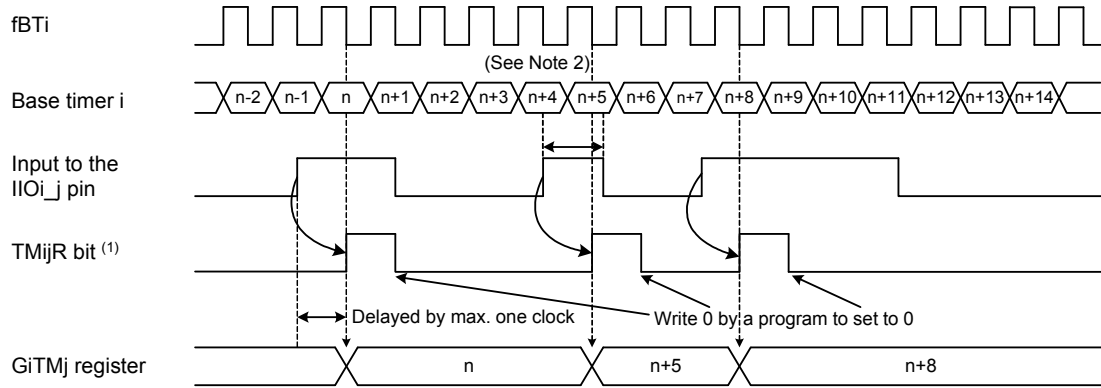
Register	Bits	Function
GiTMCRj	CTS1 and CTS0	Select a time measurement trigger
	DF1 and DF0	Select a digital filter
	GT, GOC, GSC	Select if the gating is used
	PR	Select if the prescaler is used
GiTPRk	—	Set the prescaler value
GiFS	FSCj	Set the bit to 1 (select the time measurement)
GiFE	IFEj	Set the bit to 1 (enable the channel j function)

Bit configurations and functions vary with channels and groups.

Registers associated with the time measurement should be set after setting the base timer-associated registers.

**Figure 23.22 Time Measurement Operation (1/2) (i = 0, 1; j = 0 to 7)**

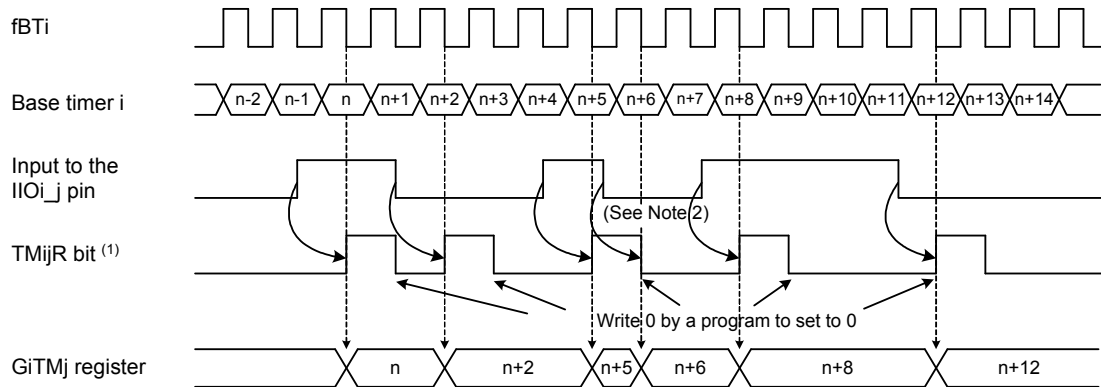
(A) When selecting the rising edge as a time measurement trigger  
(bits CTS1 and CTS0 in the GiTMCRj register are 01b)



Notes:

1. Bits in registers IIO0IR to IIO11IR.
2. Input pulse applied to the IIOi\_j pin requires at least 1.5 fBTi clock cycles.

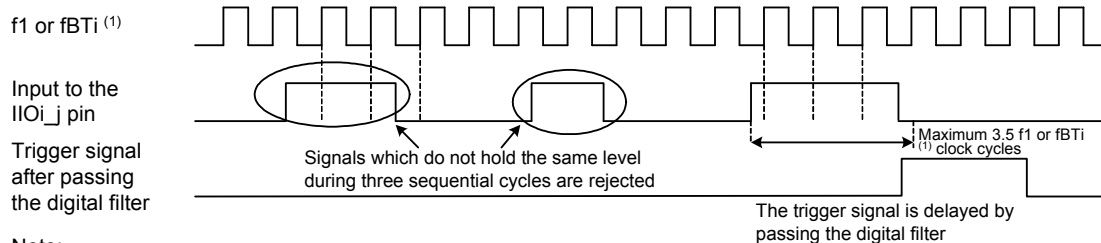
(B) When selecting both edges as a time measurement trigger  
(bits CTS1 and CTS0 in the GiTMCRj register are 11b)



Notes:

1. Bits in registers IIO0IR to IIO11IR.
2. No interrupt occurs if the MCU receives a trigger signal when the TMijR bit is 1. However, the value of GiTMj register changes.

(C) Trigger signal when using the digital filter  
(bits DF1 and DF0 in the GiTMCRj register are 10b or 11b)



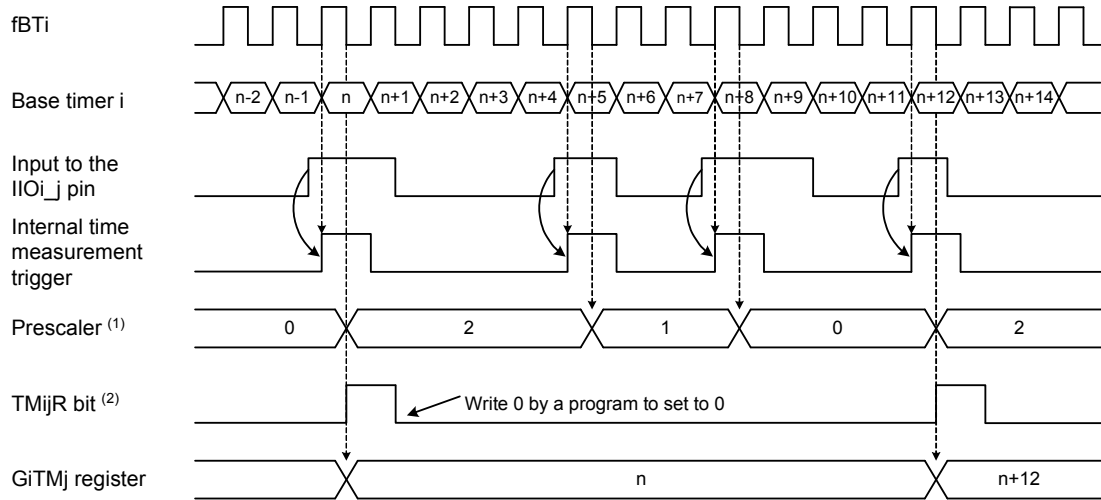
Note:

1. fBTi when bits DF1 and DF0 are 10b, f1 when the bits are 11b.

Figure 23.23 Time Measurement Operation (2/2) (i = 0, 1; j = 0 to 7)

(A) Operation with the prescaler

(the GiTPRj register is 02h and the PR bit in the GiTMCRj register is 1)

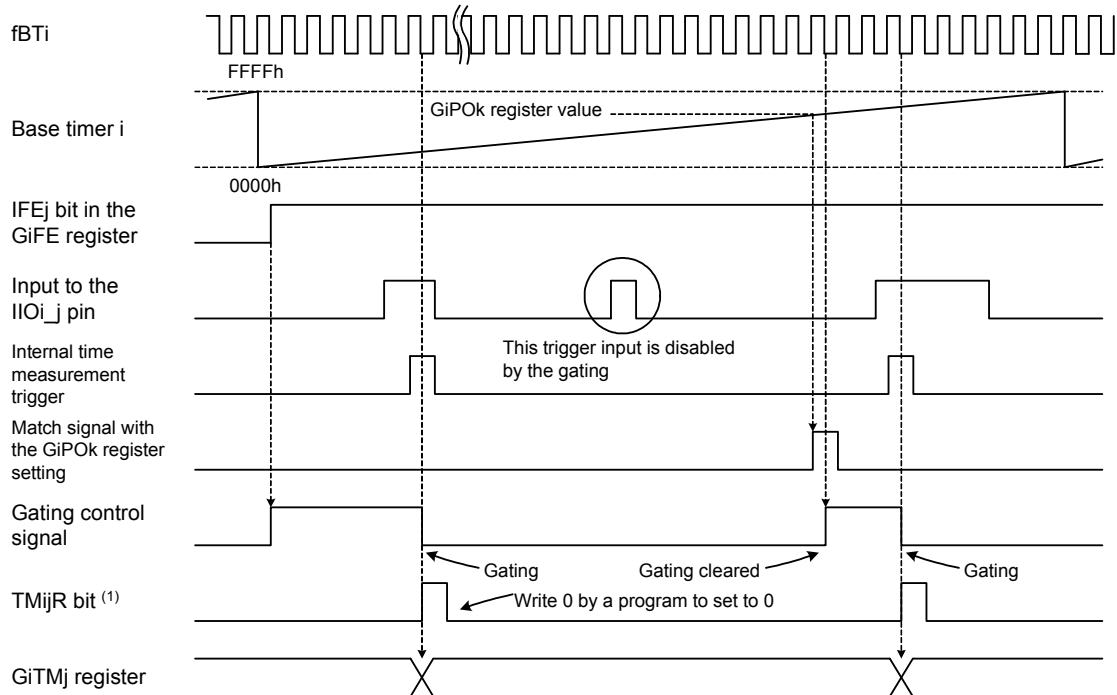


Notes:

1. This example applies to cycles following the first cycle after the PR bit in the GiTMCRj register is set to 1 (prescaler enabled).
2. Bits in registers IIO0IR to IIO11IR.

(B) Operation with the gating

(the gating is cleared by matching the base timer value with the GiPOk register setting, and bits GT and GOC in the GiTMCRj register are 1, respectively)



Note:

1. Bits in registers IIO0IR to IIO11IR.

Figure 23.24 Prescaler and Gate Operations (i = 0, 1; j = 6, 7; k = 4, 5)

### 23.3 Waveform Generation for Groups 0 to 2

Waveforms are generated when the base timer value matches the GiPOj register setting (i = 0 to 2; j = 0 to 7).

Waveform generation has the following six modes:

- Single-phase waveform output mode for groups 0 to 2
- Inverted waveform output mode for groups 0 to 2
- Set/reset waveform output (SR waveform output) mode for groups 0 to 2
- Bit modulation PWM output mode for group 2
- Real-time port output (RTP output) mode for group 2
- Parallel real-time port output (parallel RTP output) mode for group 2

Table 23.6 lists registers associated with the waveform generation.

**Table 23.6 Waveform Generation Associated Register Settings (i = 0 to 2; j = 0 to 7)**

Register	Bits	Function
GiPOCRj	MOD2 to MOD0	Select a waveform output mode
	PRT (1)	Set the bit to 1 to use parallel RTP output mode
	IVL	Select a default value
	RLD	Select a timing to reload the value into the GiPOj register
	RTP (1)	Set the bit to 1 to use RTP output mode or parallel RTP output mode. The settings of bits MOD2 to MOD0 are disabled when this bit is set to 1
	INV	Select if output level is inverted
G2BCR1	PRP	Set the bit to 1 to use parallel RTP output mode
GiPOj	—	Set the timing to invert output waveform level
GiFS	FSCj	Set the bit to 0 (select the waveform generation) for groups 0 and 1 only
GiFE	IFEj	Set the bit to 1 (enable the channel j function)
G2RTP	RTP0 to RTP7	Set the RTP output value in RTP output mode or parallel RTP output mode

Bit configurations and functions vary with channels and groups.

Registers associated with the waveform generation should be set after setting the base timer-associated registers.

Note:

1. This bit is available in the G2POCRj register only. Neither the G0POCRj nor G1POCRj register has it.

### 23.3.1 Single-phase Waveform Output Mode for Groups 0 to 2

The output level at the IIOi\_j pin (or OUTC2\_j pin for group 2) becomes high when the base timer value matches the GiPOj register (i = 0 to 2; j = 0 to 7). It switches to low when the base timer reaches 0000h. If the IVL bit in the GiPOCRj register is set to 1 (output high as default value), a high level output is provided when a waveform output starts. If the INV bit is set to 1 (invert the output level), a waveform with an inverted level is output. Refer to Figure 23.25 for details on single-phase waveform mode operation.

Table 23.7 lists specifications of single-phase waveform output mode.

**Table 23.7 Single-phase Waveform Output Mode Specifications (i = 0 to 2)**

Item	Specification
Output waveform (1)	<ul style="list-style-type: none"> <li>Free-running operation (when bits RST2 to RST0 in the GiBCR1 register are 000b)               <ul style="list-style-type: none"> <li>Cycle: <math>\frac{65536}{fBTi}</math></li> <li>Low level width: <math>\frac{m}{fBTi}</math></li> <li>High level width: <math>\frac{65536 - m}{fBTi}</math></li> </ul> </li> <li><i>m</i>: GiPOj register setting value (j = 0 to 7), 0000h to FFFFh</li> <li>The base timer is reset by matching the base timer value with the GiPO0 register setting (when bits RST2 to RST0 are 010b)               <ul style="list-style-type: none"> <li>Cycle: <math>\frac{n + 2}{fBTi}</math></li> <li>Low level width: <math>\frac{m}{fBTi}</math></li> <li>High level width: <math>\frac{n + 2 - m}{fBTi}</math></li> </ul> </li> <li><i>m</i>: GiPOj register setting value (j = 1 to 7), 0000h to FFFFh</li> <li><i>n</i>: GiPO0 register setting value, 0001h to FFFDh</li> <li>If <math>m \geq n + 2</math>, the output level is fixed to low</li> </ul>
Waveform output start condition (2)	The IFEj bit in the GiFE register is 1 (enable the channel j function) (j = 0 to 7)
Waveform output stop condition	The IFEj bit is 0 (disable the channel j function)
Interrupt request	When the POijR bit in the intelligent I/O interrupt request register becomes 1 (interrupt requested) by matching the base timer value with the GiPOj register setting (refer to Figure 11.12)
IIOi_j output pin (or OUTC2_j pin for group 2) function	Pulse signal output
Other functions	<ul style="list-style-type: none"> <li>Default value setting This function determines the starting waveform output level</li> <li>Output level inversion This function inverts the waveform output level and outputs the inverted signal from the IIOi_j pin (or OUTC2_j pin for group 2)</li> </ul>

**Notes:**

- When the INV bit in the GiPOCRj register is 1 (invert the output level), the high and low widths are inverted.
- To use channels shared by time measurement and waveform generation, set the FSCj bit in the GiFS register to 0 (select the waveform generation).

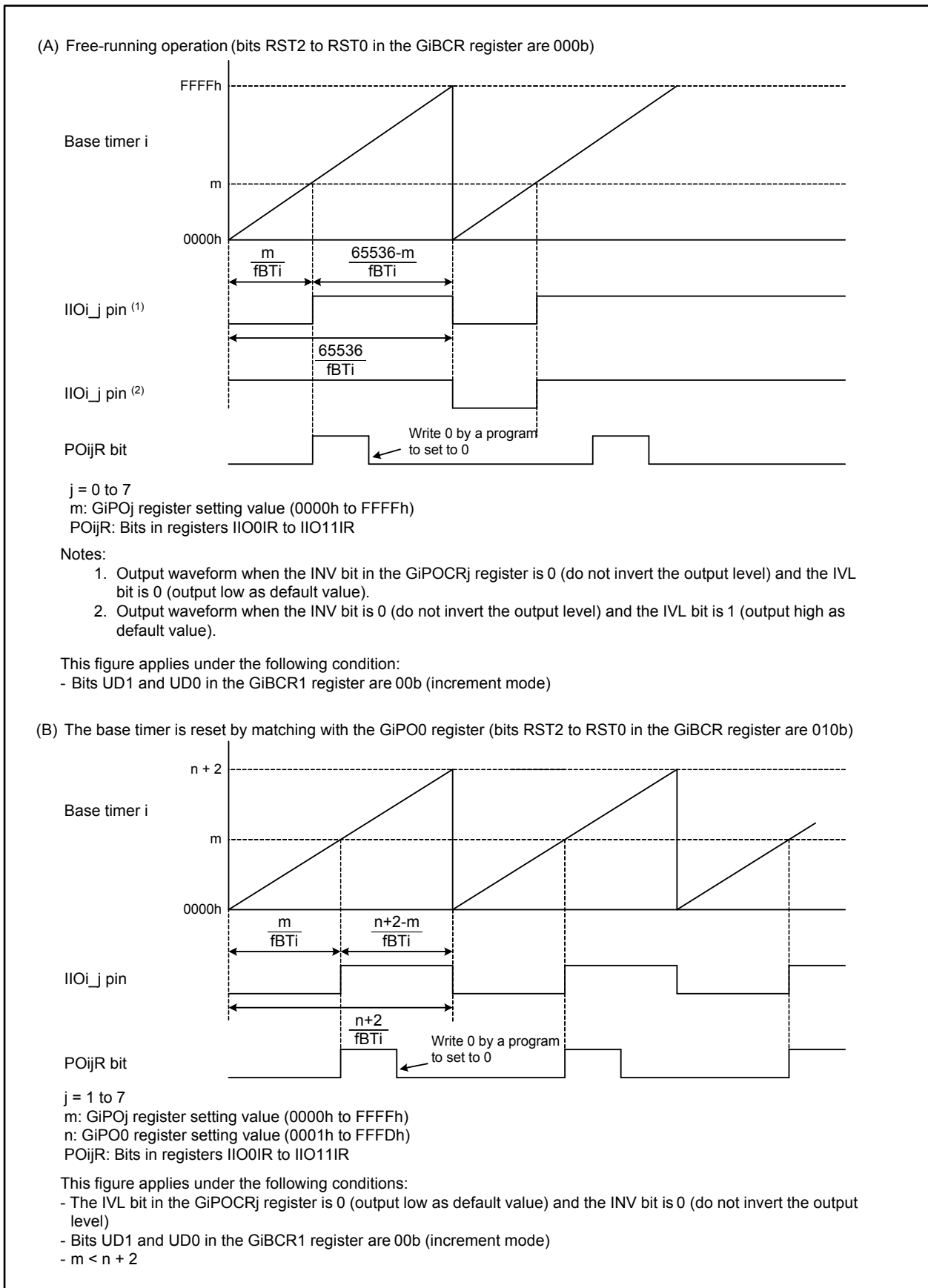


Figure 23.25 Single-phase Waveform Output Mode Operation (i = 0 to 2)

### 23.3.2 Inverted Waveform Output Mode for Groups 0 to 2

The output level at the IIOi\_j pin (or OUTC2\_j pin for group 2) is inverted every time the base timer value matches the GiPOj register setting (i = 0 to 2; j = 0 to 7).

Table 23.8 lists specifications of the inverted waveform output mode. Figure 23.26 shows an example of the inverted waveform output mode operation.

**Table 23.8 Inverted Waveform Output Mode Specifications (i = 0 to 2)**

Item	Specification
Output waveform	<ul style="list-style-type: none"> <li>Free-running operation (when bits RST2 to RST0 in the GiBCR1 register are 000b)           <ul style="list-style-type: none"> <li>Cycle: <math>\frac{65536 \times 2}{fBTi}</math></li> <li>High or low level width: <math>\frac{65536}{fBTi}</math></li> <li>m: GiPOj register setting value (j = 0 to 7), 0000h to FFFFh</li> </ul> </li> <li>The base timer is reset by matching the base timer value with the GiPO0 register setting (when bits RST2 to RST0 are 010b)           <ul style="list-style-type: none"> <li>Cycle: <math>\frac{2(n+2)}{fBTi}</math></li> <li>High or low level width: <math>\frac{n+2}{fBTi}</math></li> <li>n: GiPO0 register setting value, 0001h to FFFDh</li> <li>GiPOj register setting value (j = 1 to 7), 0000h to FFFFh</li> <li>If the GiPOj register setting <math>\geq n+2</math>, the output level is not inverted</li> </ul> </li> </ul>
Waveform output start condition <sup>(1)</sup>	The IFEj bit in the GiFE register is 1 (enable the channel j function) (j = 0 to 7)
Waveform output stop condition	The IFEj bit is 0 (disable the channel j function)
Interrupt request	When the POijR bit in the intelligent I/O interrupt request register becomes 1 (interrupt requested) by matching the base timer value with the GiPOj register setting (refer to Figure 11.12)
IIOi_j output pin (or OUTC2_j pin for group 2) function	Pulse signal output
Other functions	<ul style="list-style-type: none"> <li>Default value setting This function determines the starting waveform output level</li> <li>Output level inversion This function inverts the waveform output level and outputs the inverted signal from the IIOi_j pin (or OUTC2_j pin for group 2)</li> </ul>

Note:

- To use channels shared by time measurement and waveform generation, set the FSCj bit in the GiFS register to 0 (select the waveform generation).

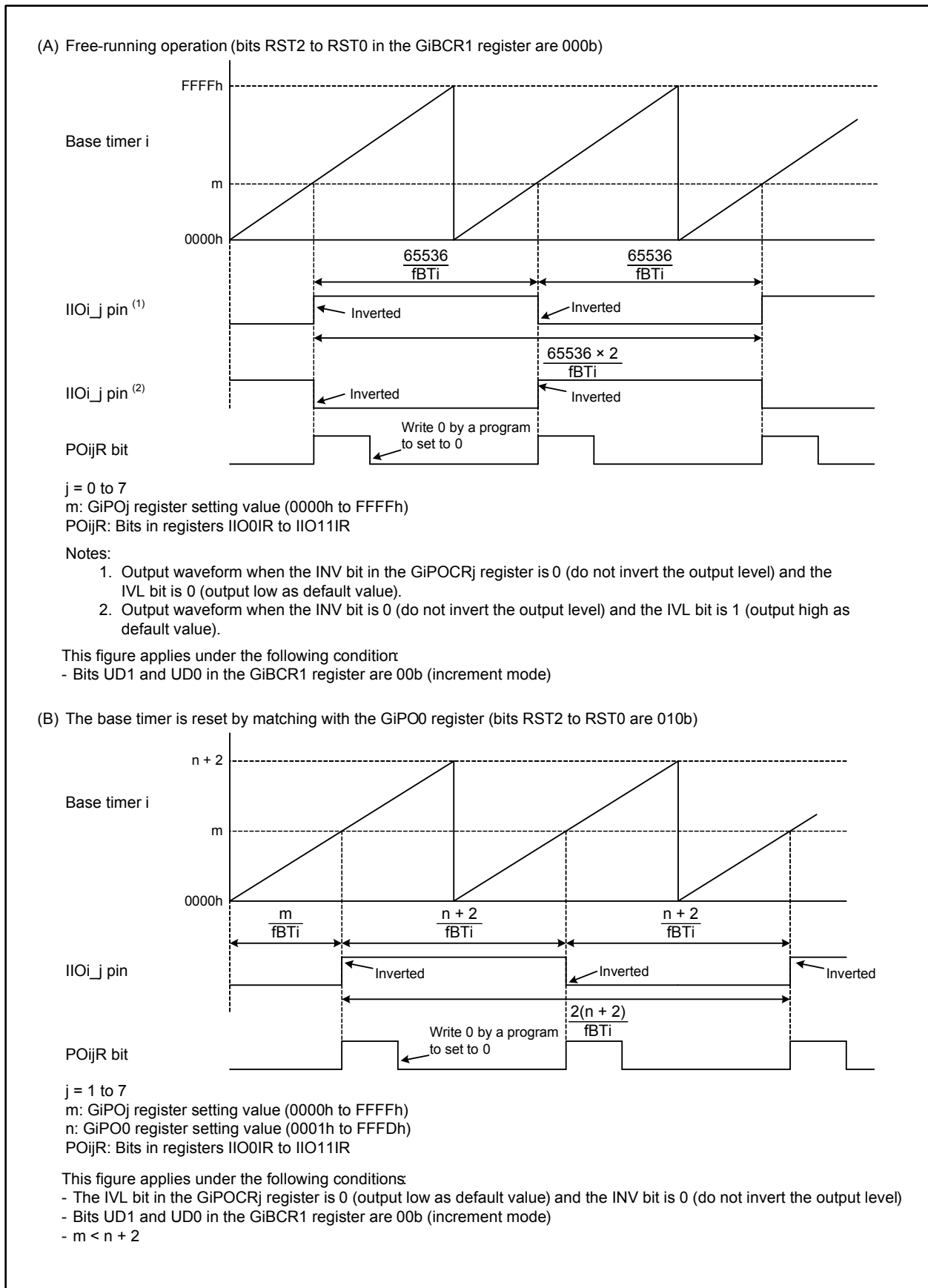


Figure 23.26 Inverted Waveform Output Mode Operation (i = 0 to 2)



### 23.3.3 Set/Reset Waveform Output Mode (SR Waveform Output Mode) for Groups 0 to 2

The output level at the IIOi\_j pin (or OUTC2\_j pin for group 2) becomes high when the base timer value matches the GiPOj register setting (i = 0 to 2; j = 0, 2, 4, 6). It becomes low when the base timer value matches the GiPOk register setting or the base timer reaches 0000h (k = j + 1). When the IVL bit in the GiPOCRj register is set to 1 (output high as default value), a high output level is provided when a waveform output starts (j = 0 to 7). When the INV bit is set to 1 (invert the output level), a waveform with inverted level is output. Refer to Figure 23.27 for details on SR waveform mode operation. Tables 23.9 and 23.10 list specifications of SR waveform output mode.

**Table 23.9 SR Waveform Output Mode Specifications (i = 0 to 2) (1/2)**

Item	Specification
Output waveform <sup>(1)</sup>	<ul style="list-style-type: none"> <li>• Free-running operation (when bits RST2 to RST0 in the GiBCR1 register are 000b)               <ul style="list-style-type: none"> <li>(A) <math>m &lt; n</math> <ul style="list-style-type: none"> <li>High level width: <math>\frac{n - m}{fBTi}</math></li> <li>Low level width: <math>\frac{m}{fBTi}</math> (See Note 2) + <math>\frac{65536 - n}{fBTi}</math> (See Note 3)</li> </ul> </li> <li>(B) <math>m \geq n</math> <ul style="list-style-type: none"> <li>High level width: <math>\frac{65536 - m}{fBTi}</math></li> <li>Low level width: <math>\frac{m}{fBTi}</math></li> </ul> </li> </ul> </li> <li><math>m</math>: GiPOj register setting value (j = 0, 2, 4, 6), 0000h to FFFFh  <math>n</math>: GiPOk register setting value (k = j + 1), 0000h to FFFFh</li> <li>• The base timer is reset by matching with the GiPO0 register (when bits RST2 to RST0 are 010b) <sup>(4)</sup> <ul style="list-style-type: none"> <li>(A) <math>m &lt; n &lt; p + 2</math> <ul style="list-style-type: none"> <li>High level width: <math>\frac{n + m}{fBTi}</math></li> <li>Low width: <math>\frac{m}{fBTi}</math> (See Note 2) + <math>\frac{p + 2 - n}{fBTi}</math> (See Note 3)</li> </ul> </li> <li>(B) <math>m &lt; p + 2 \leq n</math> <ul style="list-style-type: none"> <li>High level width: <math>\frac{p + 2 - m}{fBTi}</math></li> <li>Low level width: <math>\frac{m}{fBTi}</math></li> </ul> </li> <li>(C) <math>m \geq p + 2</math>, output level is fixed to low  <math>p</math>: GiPO0 register setting value, 0001h to FFFDh  <math>m</math>: GiPOj register setting value (j = 2, 4, 6), 0000h to FFFFh  <math>n</math>: GiPOk register setting value (k = j + 1), 0000h to FFFFh</li> </ul> </li> </ul>

Notes:

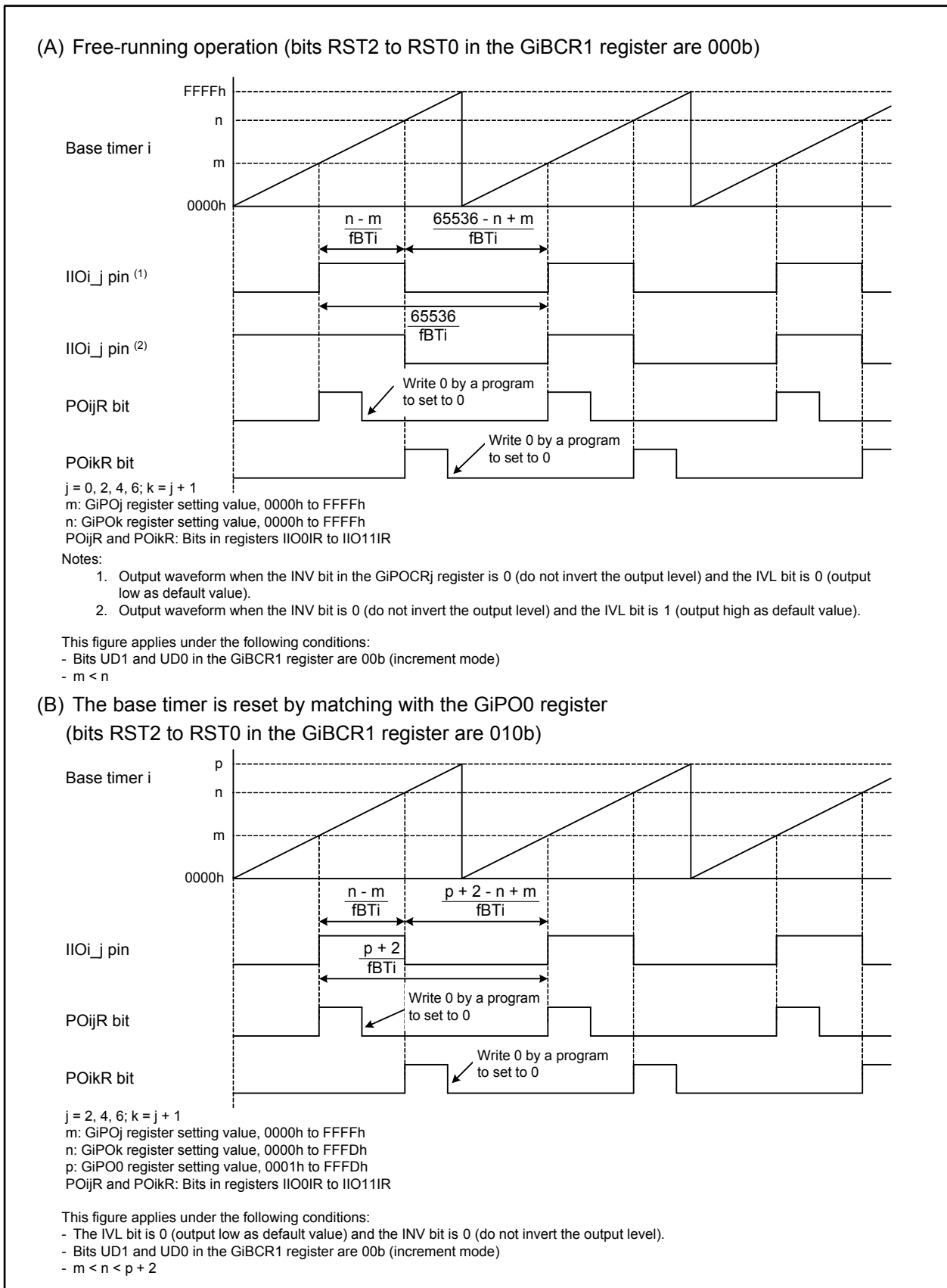
1. When the INV bit in the GiPOCRj register is 1 (invert the output level), the high and low widths are inverted.
2. Output period from a base timer reset until when the output level becomes high.
3. Output period from when the output level becomes low until the next base timer reset.
4. When the GiPO0 register resets the base timer, channel 0 and channel 1 SR waveform generation functions are not available.

**Table 23.10 SR Waveform Output Mode Specifications (i = 0 to 2) (2/2)**

Item	Specification
Waveform output start condition (1)	The IFEq bit in the GiFE register is 1 (enable the channel q function) (q = 0 to 7)
Waveform output stop condition	The IFEq bit is 0 (disable the channel q function)
Interrupt request	When the POijR bit in the intelligent I/O interrupt request register becomes 1 (interrupt requested) by matching the base timer value with the GiPOj register setting. When the POikR bit becomes 1 (interrupt requested) by matching the base timer value with the GiPOk register setting (refer to Figure 11.12)
IIOi_j output pin (or OUTC2_j pin for group 2) function	Pulse signal output
Other functions	<ul style="list-style-type: none"> <li>• Default value setting This function determines the starting waveform output level</li> <li>• Output level inversion This function inverts the waveform output level and outputs the inverted signal from the IIOi_j pin (or OUTC2_j pin for group 2)</li> </ul>

**Note:**

1. To use channels shared by time measurement and waveform generation, set the FSCj bit in the GiFS register to 0 (select the waveform generation).

Figure 23.27 SR Waveform Output Mode Operation ( $i = 0$  to 2)

### 23.3.4 Bit Modulation PWM Output Mode for Group 2

In bit modulation PWM output mode, a PWM output has 16-bit resolution.

Pulses are repeatedly output in a period of 1024 consecutive periods of span  $t$ . The period of span  $t$  is

$\frac{64}{fBT2}$ . The 6 upper bits in the G2POj register determine the base low width ( $j = 0$  to 7). The 10 lower

bits determine the number of span  $t$ , within a period, in which the low width is extended by the minimum resolution bit width, that is, one clock cycle.

When the INV bit is set to 1 (invert the output level), the waveform with an inverted level is output.

Table 23.11 lists specifications of bit modulation PWM output mode. Table 23.12 lists the number of modulated spans and span  $t$ s to be extended with the minimum resolution bit width. Figure 23.28 shows an example of bit modulation PWM output mode operation.

**Table 23.11 Bit Modulation PWM Output Mode Specifications ( $j = 0$  to 7)**

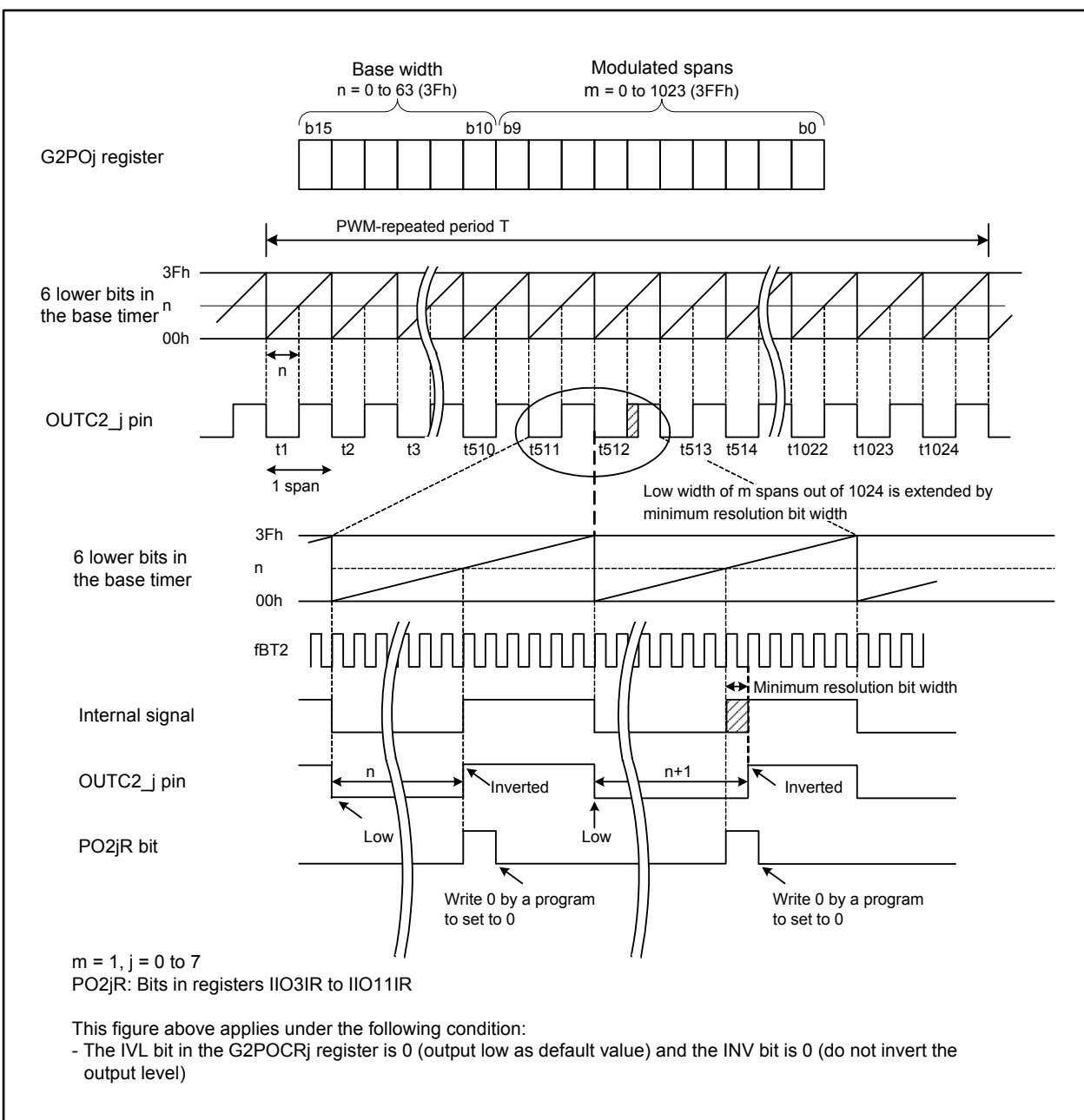
Item	Specification
Output waveform (1,2)	PWM-repeated period T: $\frac{65536}{fBT2} \left( = \frac{64}{fBT2} \times 1024 \right)$ Period of span $t$ : $\frac{64}{fBT2}$ Low width: $\frac{n+1}{fBT2}$ of $m$ spans $\frac{n}{fBT2}$ of $(1024 - m)$ spans Mean low width: $\frac{1}{fBT2} \times \left( n + \frac{m}{1024} \right)$ $n$ : G2POj register setting value (6 upper bits), 00h to 3Fh $m$ : G2POj register setting value (10 lower bits), 000h to 3FFh
Waveform output start condition	The IFEj bit in the G2FE register is 1 (enable the channel $j$ function)
Waveform output stop condition	The IFEj bit is 0 (disable the channel $j$ function)
Interrupt request	When the PO2jR bit in the interrupt request register becomes 1 (interrupt requested) by matching the 6 lower bits of the base timer value with the 6 upper bits of the G2POj register setting (refer to Figure 11.12)
OUTC2_j pin function	Pulse signal output pin
Other functions	<ul style="list-style-type: none"> <li>• Default value setting This function determines the starting waveform output level</li> <li>• Output level inversion This function inverts the waveform output level and outputs the inverted signal from the OUTC2_j pin</li> </ul>

**Notes:**

1. Bits RST2 and RST0 in the G2BCR1 register should be set to 000b to use bit modulation PWM output mode.
2. When the INV bit in the G2POCRj register is set to 1 (invert the output level), the high and low widths are inverted.

**Table 23.12 Number of Modulated Spans and Span t Extended Minimum Resolution Bit Width**

Modulated Spans	Span ts to be Extended with Minimum Resolution Bit Width
00 0000 0000b	none
00 0000 0001b	t512
00 0000 0010b	t256 and t768
00 0000 0100b	t128, t384, t640, and t896
00 0000 1000b	t64, t192, t320, t448, t576, t704, t832, and t960
:	:
10 0000 0000b	t1, t3, t5, t7, ... t1019, t1021, and t1023



**Figure 23.28 Bit Modulation PWM Output Mode Operation**

### 23.3.5 Real-time Port Output Mode (RTP Output Mode) for Group 2

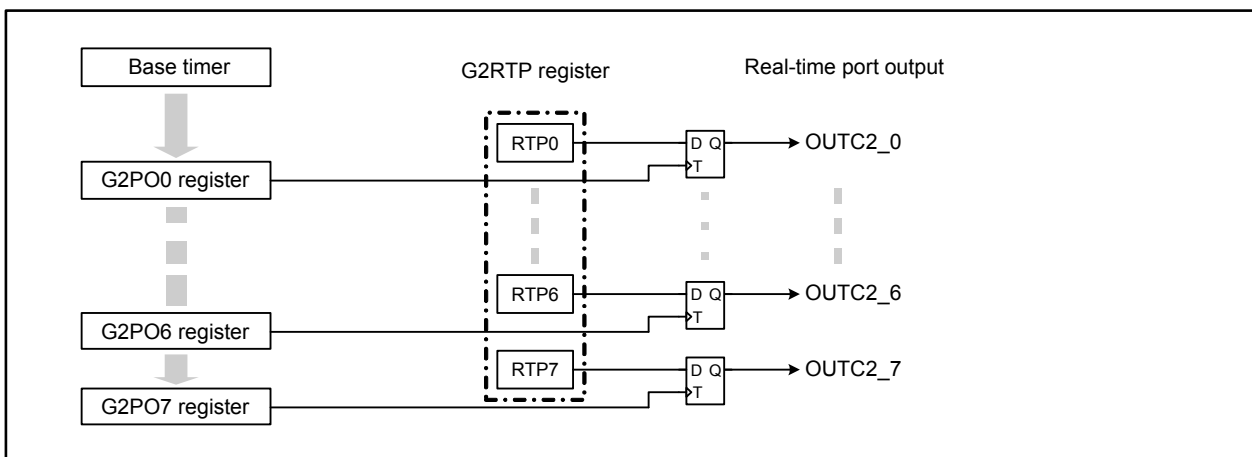
The OUTC2\_j pin outputs the G2RTP register setting value in 1-bit units when the base timer value matches the G2POj register setting (j = 0 to 7). Table 23.13 lists specifications of RTP output mode. Figure 23.29 shows a block diagram of RTP output and Figure 23.30 shows an example of RTP output mode operation.

**Table 23.13 RTP Output Mode Specifications (j = 0 to 7)**

Item	Specification
Waveform output start condition	The IFEj bit in the G2FE register is 1 (enable the channel j function)
Waveform output stop condition	The IFEj bit is 0 (disable the channel j function)
Interrupt request	When the PO2jR bit in the interrupt request register becomes 1 (interrupt requested) by matching the base timer value with the G2POj register setting (0000h to FFFFh <sup>(1)</sup> ) (refer to Figure 11.12)
OUTC2_j pin function	RTP output pin
Other functions	<ul style="list-style-type: none"> <li>• Default value setting This function determines the starting waveform output level</li> <li>• Output level inversion This function inverts the waveform output level and outputs the inverted signal from the OUTC2_j pin</li> </ul>

Note:

1. The G2PO0 register should be set to between 0001h and FFFDh to set the base timer value to 0000h (bits RST2 to RST0 are set to 010b) when the base timer value matches the G2PO0 register setting.



**Figure 23.29 RTP Output Block Diagram**

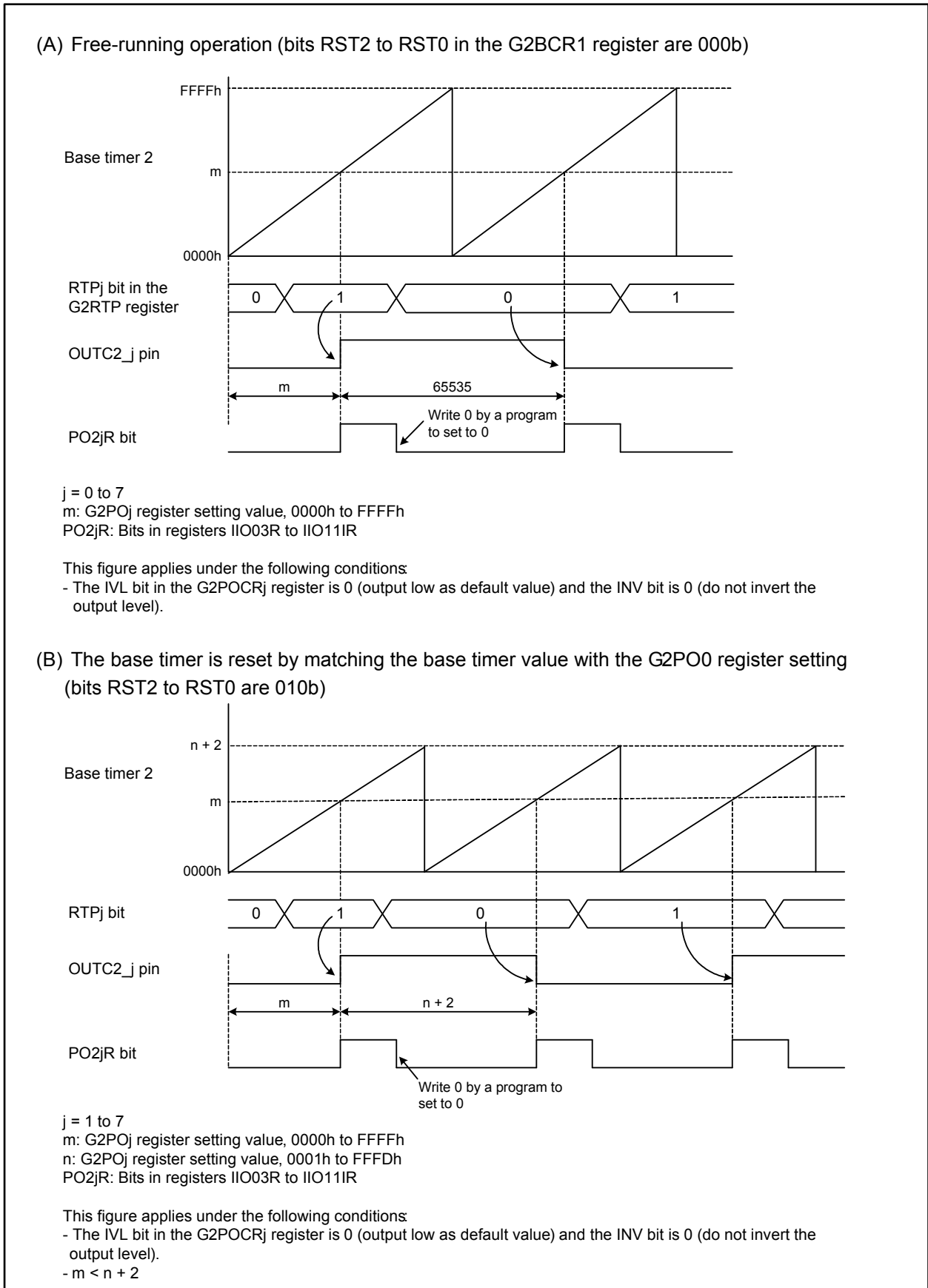


Figure 23.30 RTP Output Mode Operation

### 23.3.6 Parallel Real-time Port Output Mode (RTP Output Mode) for Group 2

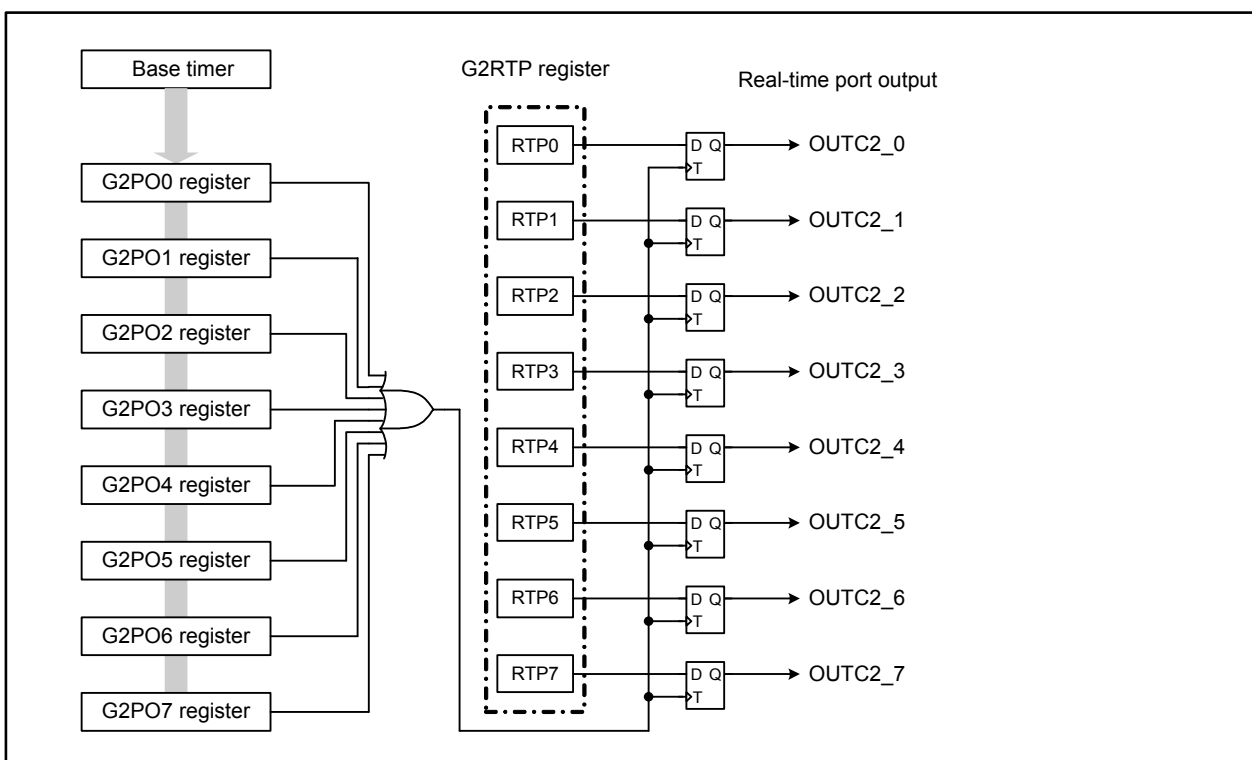
The OUTC2\_j pin outputs all the G2RTP register setting values in 1-byte units when the base timer value matches the G2POj register setting (j = 0 to 7). Table 23.14 lists specifications of parallel RTP output mode. Figure 23.7 shows the G2BCR1 register. Figure 23.31 shows a block diagram of parallel RTP output and Figure 23.32 shows an example of parallel RTP output mode operation.

**Table 23.14 Parallel RTP Output Mode Specifications (j = 0 to 7)**

Item	Specification
Waveform output start condition	The IFEj bit in the G2FE register is 1 (enable the channel j function)
Waveform output stop Condition	The IFEj bit is 0 (disable the channel j function)
Interrupt request	The PO2jR bit in the interrupt request register becomes 1 (interrupt requested) when the base timer value matches the G2POj register setting (0000h to FFFFh <sup>(1)</sup> ) (refer to Figure 11.12)
OUTC2_j pin function	RTP output pin
Other functions	<ul style="list-style-type: none"> <li>• Default value setting This function determines the starting waveform output level</li> <li>• Output level inversion This function inverts the waveform output level and outputs the inverted signal from the OUTC2_j pin</li> </ul>

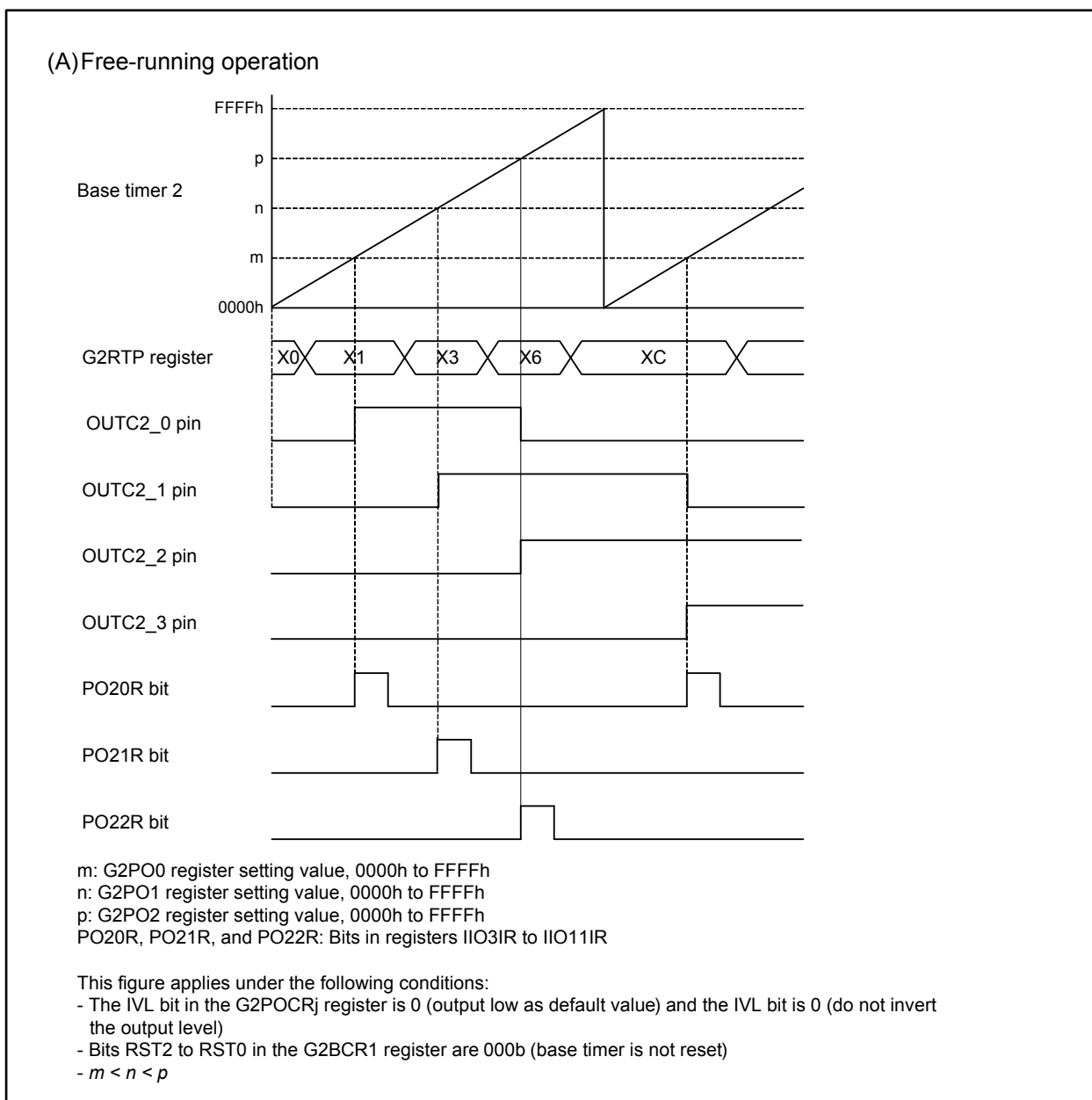
Note:

1. The G2PO0 register should be set to between 0001h and FFFDh to set the base timer value to 0000h (bits RST2 to RST0 are set to 010b) when the base timer value matches the G2PO0 register setting.



**Figure 23.31 Parallel RTP Output Mode Block Diagram**





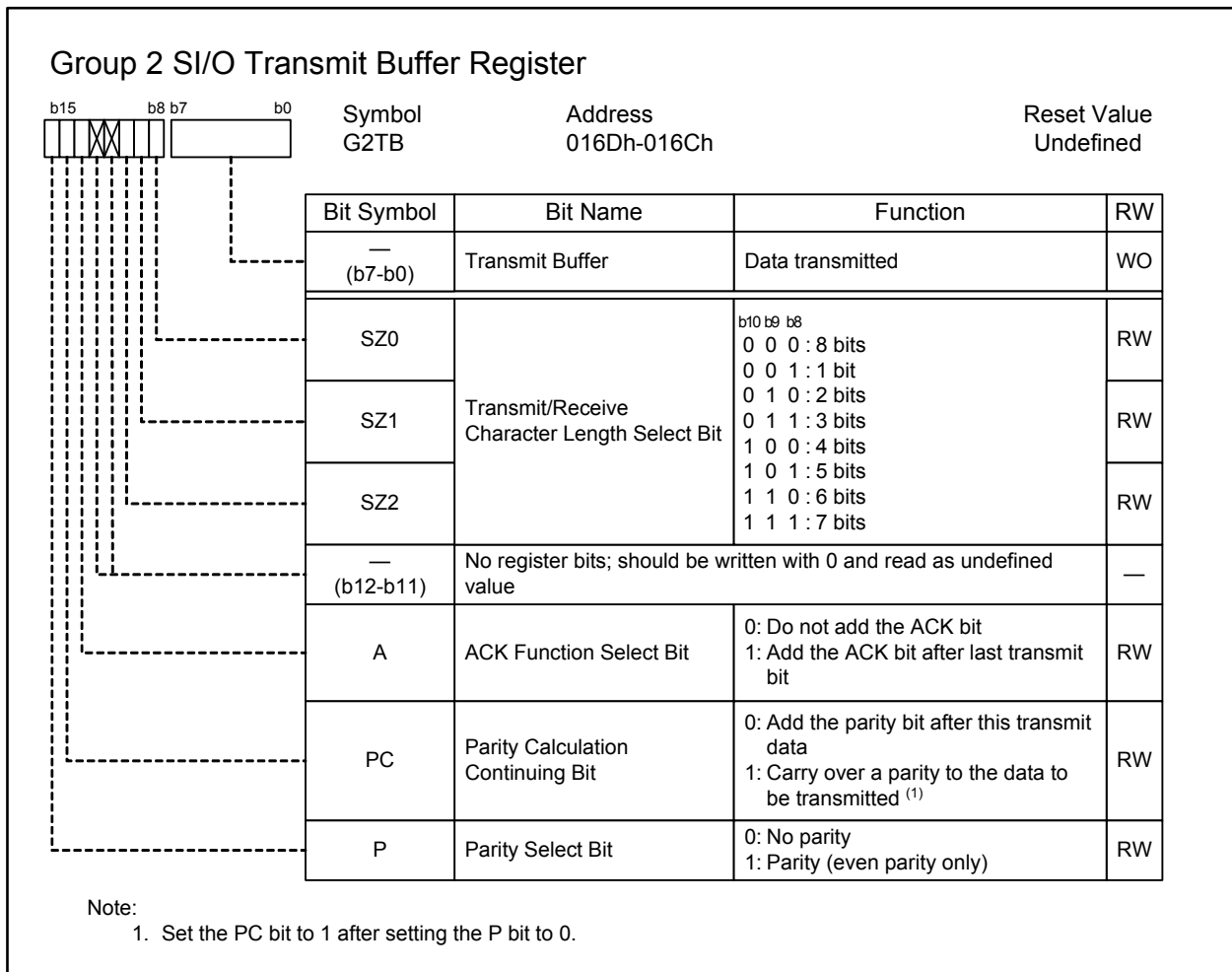
**Figure 23.32 Parallel RTP Output Mode Operation**

## 23.4 Group 2 Serial Interface

Two 8-bit shift registers and waveform generation enable the serial interface function. In group 2 of the intelligent I/O, the variable synchronous serial interface and IEBus (optional <sup>(1)</sup>) are available. Figures 23.33 to 23.40 show associated registers.

Note:

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



**Figure 23.33 G2TB Register**

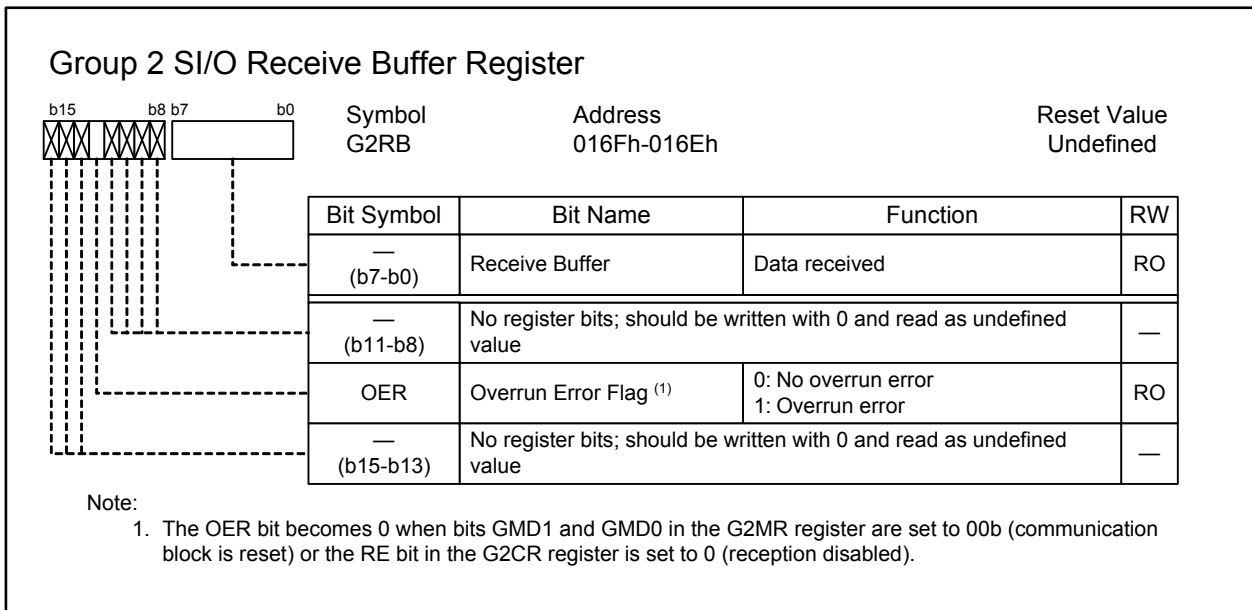


Figure 23.34 G2RB Register

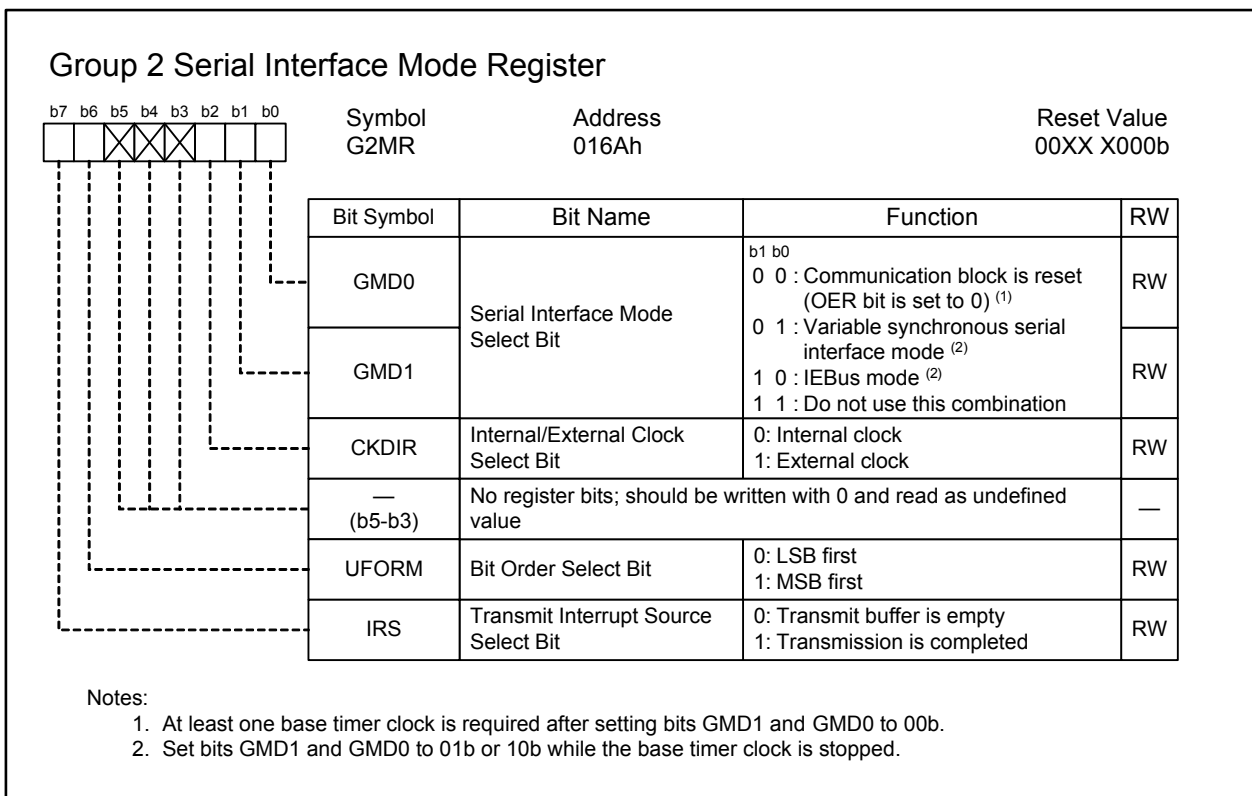
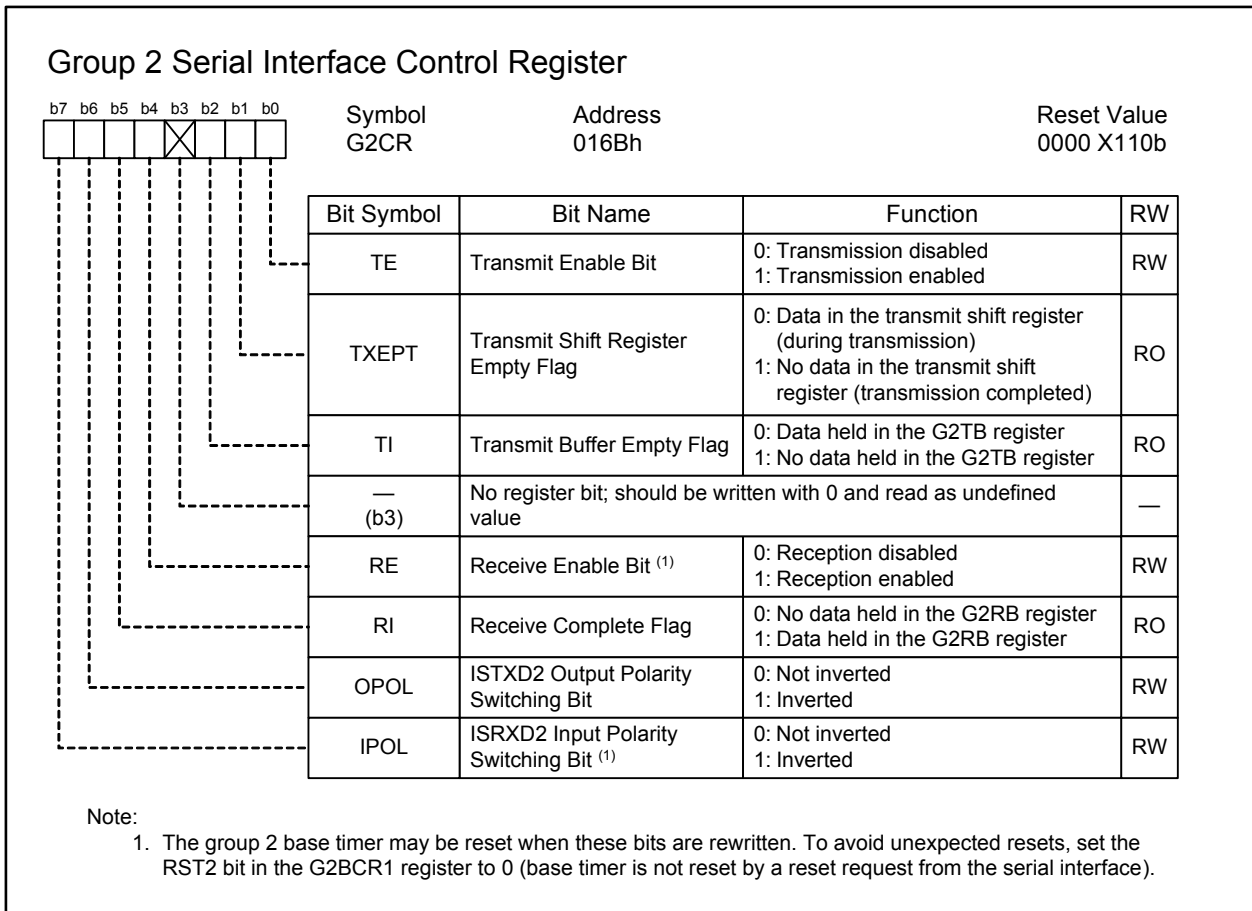


Figure 23.35 G2MR Register



**Figure 23.36 G2CR Register**

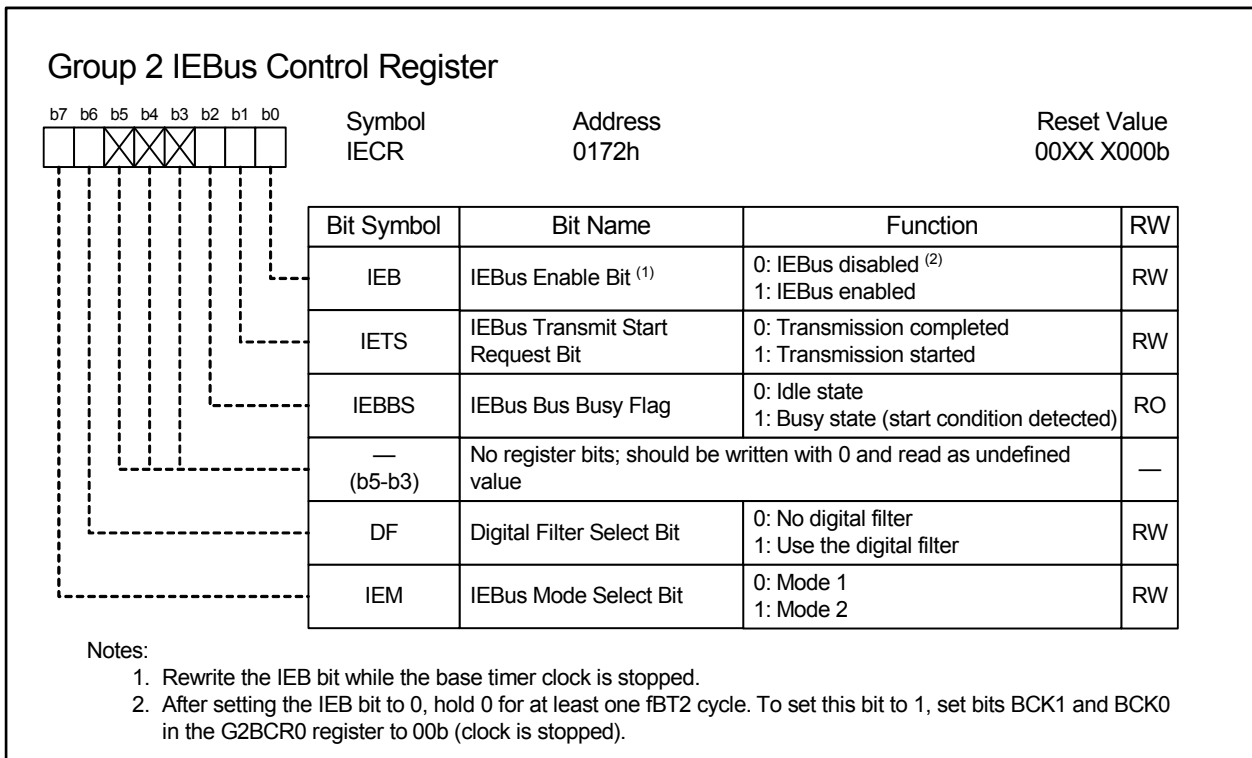


Figure 23.37 IECR Register

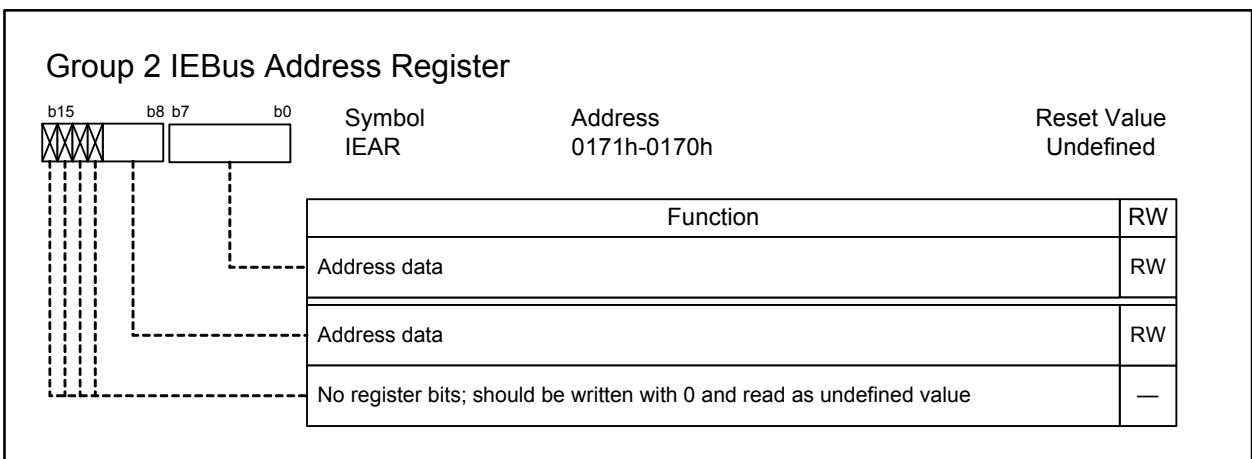


Figure 23.38 IEAR Register

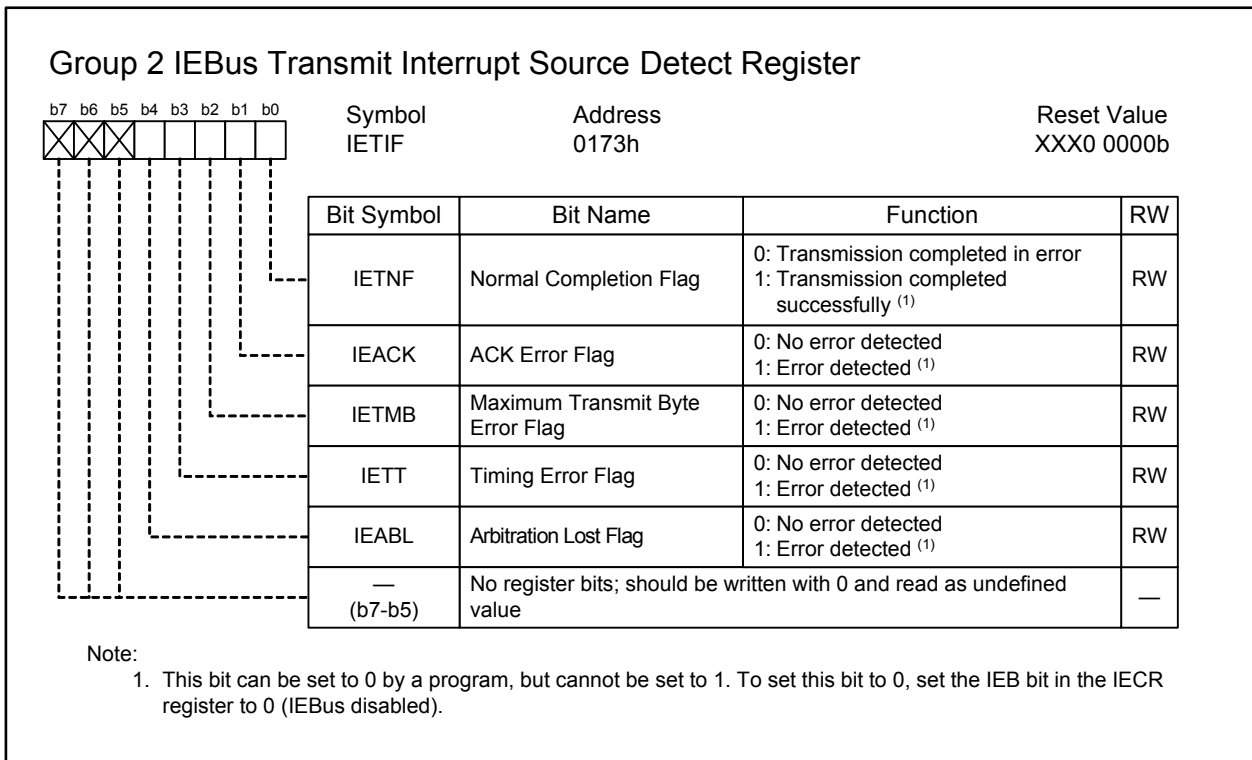


Figure 23.39 IETIF Register

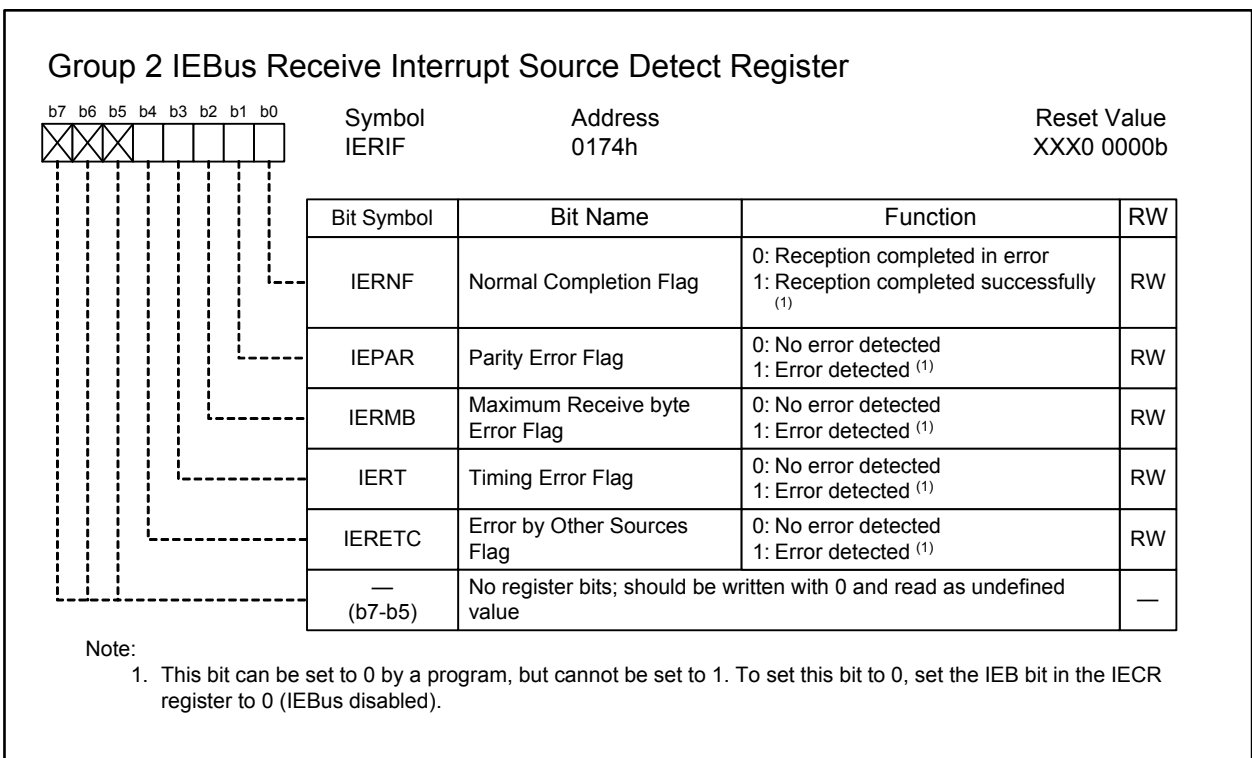


Figure 23.40 IERIF Register

### 23.4.1 Variable Synchronous Serial Interface Mode for Group 2

This mode allows data transmission/reception synchronized with the transmit/receive clock. The character length is selectable from 1 to 8 bits. Table 23.15 lists specifications of the group 2 variable synchronous serial interface mode and Table 23.16 lists its settings. Figure 23.41 shows an operation example of data transmission/reception.

**Table 23.15 Group 2 Variable Synchronous Serial Interface Mode Specifications**

Item	Specification
Data format	1- to 8-bit character length
Transmit/receive clock	<ul style="list-style-type: none"> <li>The CKDIR bit in the G2MR register is 0 (internal clock selected):  <math display="block">\frac{f_{BT2}}{2(n+2)}</math> <i>n</i>: G2PO0 register setting value, 0000h to FFFFh <sup>(1)</sup>            The bit rate is set using the G2PO0 register. The clock is generated in the inverted waveform output mode of the channel 2 waveform generation</li> <li>The CKDIR bit is 1 (external clock selected): input into the ISCLK2 pin <sup>(2)</sup></li> </ul>
Transmit start conditions	The conditions for starting data transmission are as follows: <ul style="list-style-type: none"> <li>The TE bit in the G2CR register is 1 (transmission enabled)</li> <li>The TI bit in the G2CR register is 0 (data held in the G2TB register)</li> </ul>
Receive start conditions	The conditions for starting data reception are as follows: <ul style="list-style-type: none"> <li>The RE bit in the G2CR register is 1 (reception enabled)</li> <li>The TE bit in the G2CR register is 1 (transmission enabled)</li> <li>The TI bit in the G2CR register is 0 (data held in the G2TB register)</li> </ul>
Interrupt request	In transmit interrupt, either of the following conditions is selected to set the SIO2TR bit in the IIO6IR register to 1 (interrupt requested) (refer to Figure 11.12): <ul style="list-style-type: none"> <li>The IRS bit in the G2MR register is 0 (transmit buffer in the G2TB register is empty):              when data is transferred from the G2TB register to the transmit shift register (when the transmission has started)</li> <li>The IRS bit is 1 (transmission is completed):              when data transmission from the transmit shift register is completed</li> </ul> In receive interrupt, When data is transferred from the receive shift register to the G2RB register (when the reception is completed), the SIO2PR bit in the IIO5IR register becomes 1 (interrupt requested) (refer to Figure 11.12)
Error detection	Overrun error <sup>(3)</sup> This error occurs when the last bit of the next data has been received before reading the G2RB register
Other functions	<ul style="list-style-type: none"> <li>Bit order selection              LSB first or MSB first</li> <li>ISTXD2 and ISRXD2 I/O polarity              Output levels from the ISTXD2 pin and input levels to the ISRXD2 pin can be inverted</li> <li>Character length for data transmission/reception              1- to 8-bit character length</li> </ul>

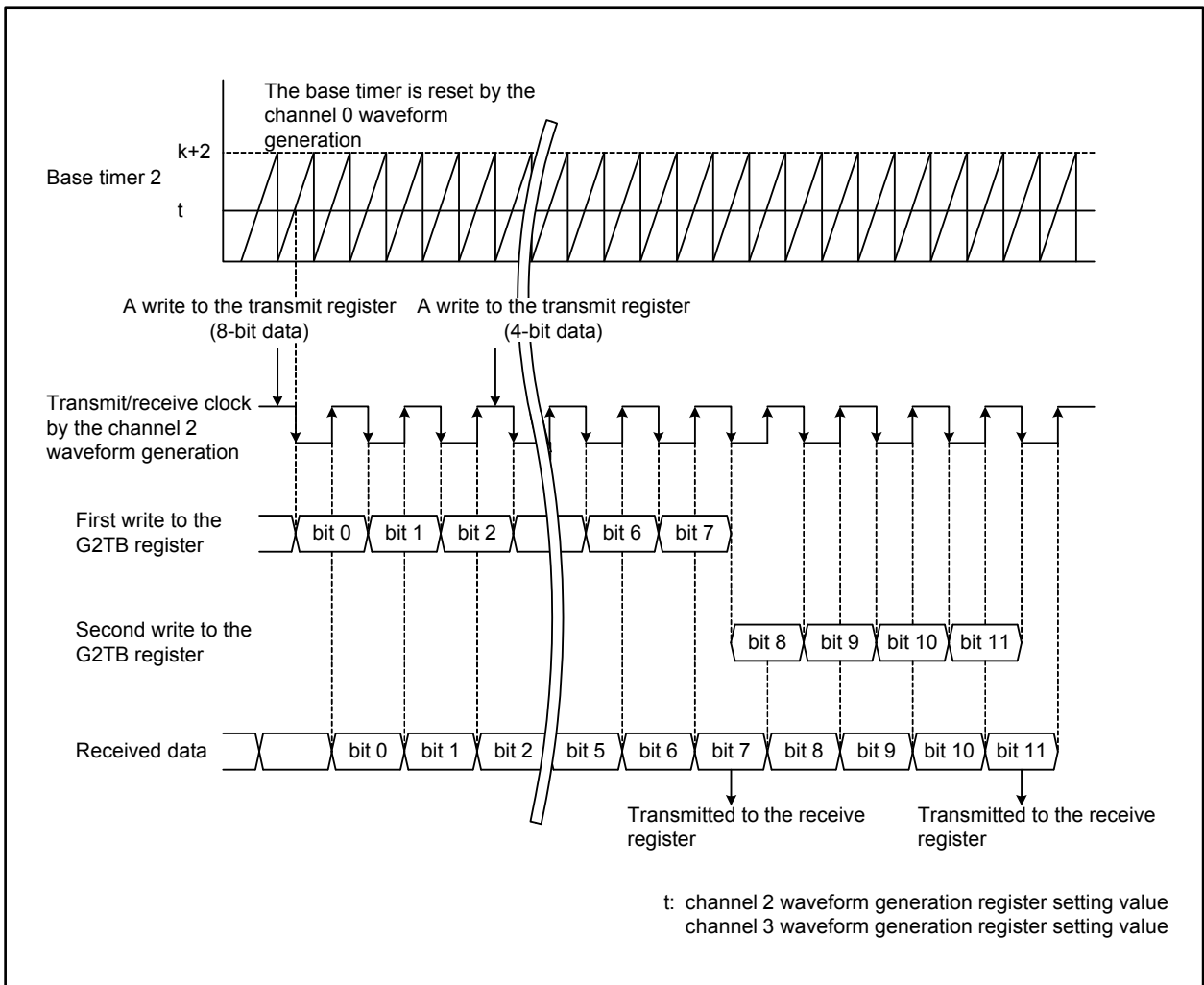
Notes:

- When using the serial interface, set a value greater than or equal to 1 to the G2PO0 register.
- The highest transmit/receive clock frequency should be  $f_{BT2}$  divided by 20.
- If an overrun error occurs, the G2RB register is undefined.

**Table 23.16 Register Settings in Group 2 Variable Synchronous Serial Interface Mode**

Register	Bits	Function
G2BCR0	BCK1 and BCK0	Set the bits to 11b
	DIV4 to DIV0	Select a divide ratio of count source
	IT	Set the bit to 0
G2BCR1	7 to 0	Set the bits to 0001 0010b
G2POCR0	7 to 0	Set the bits to 0000 0111b
G2POCR1	7 to 0	Set the bits to 0000 0111b
G2POCR2	7 to 0	Set the bits to 0000 0010b
G2PO0	15 to 0	Set a comparative value for waveform generation $\frac{f_{BT2}}{2 \times (\text{setting value} + 2)} = \text{transmit/receive clock frequency}$
G2PO2	15 to 0	Set to a value smaller than that in the G2PO0 register setting
G2FE	IFE2 to IFE0	Set the bits to 111b
G2MR	GMD1 and GMD0	Set the bits to 01b
	CKDIR	Select either the internal clock or the external clock
	UFORM	Select either LSB first or MSB first
	IRS	Select a source for transmit interrupt
G2CR	TE	Set the bit to 1 to enable data transmission/reception
	TXEPT	Transmit shift register empty flag
	TI	Transmit buffer empty flag
	RE	Set the bit to 1 to enable data reception
	RI	Receive complete flag
	OPOL	Select if the output level at the ISTXD2 pin is inverted (usually set the bit to 0)
	IPOL	Select if the input level at the ISRXD2 pin is inverted (usually set the bit to 0)
G2TB	15 to 0	Set the data to be transmitted/received and its character length
G2RB	15 to 0	Store received data and error flag





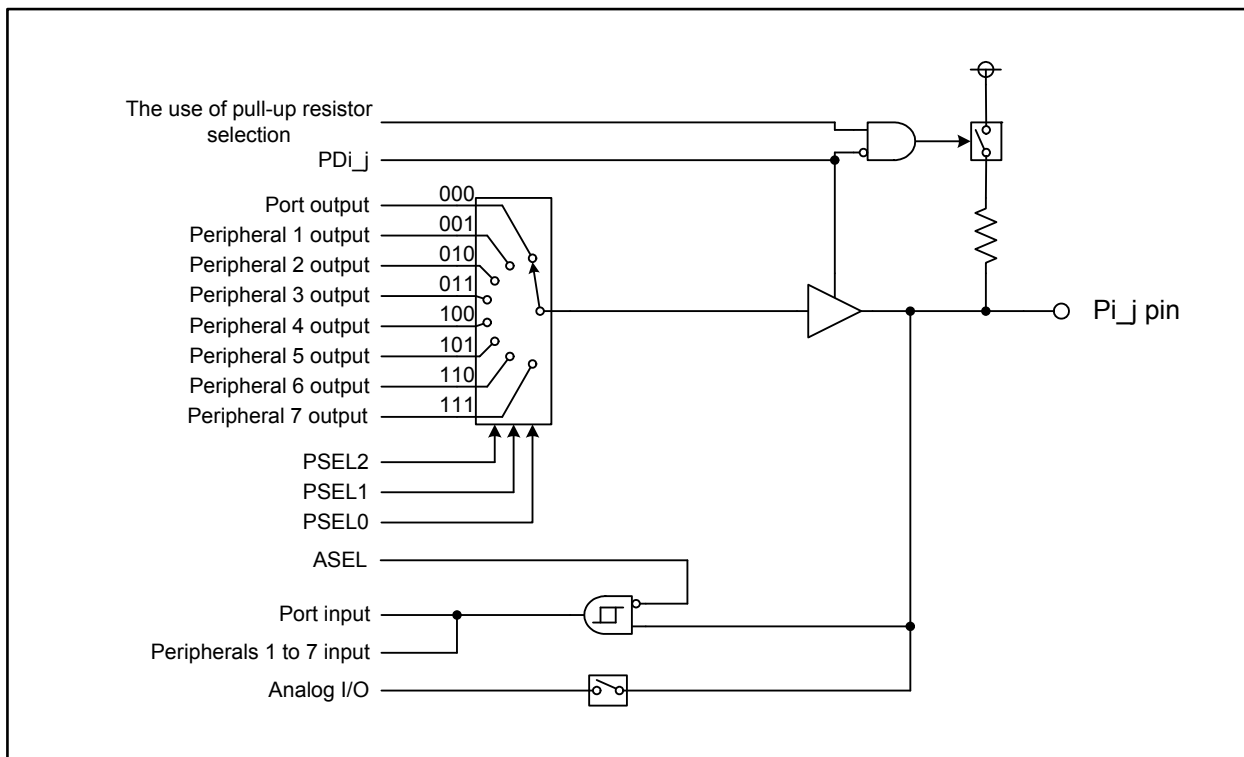
**Figure 23.41 Group 2 Variable Synchronous Serial Interface Mode Transmit/Receive Operation**

## 24. I/O Pins

Each pin of the MCU functions as a programmable I/O port, an I/O pin for integrated peripherals, or a bus control pin. These functions can be switched by the function select registers or the processor mode registers. This chapter particularly addresses the function select registers. For the use as a bus control pin, refer to 7. "Processor Mode" and 9. "Bus".

The pull-up resistors are enabled for every group of four pins. However, when a pin functions as an output pin, a pull-up resistor is disabled regardless of the register settings.

Figure 24.1 shows a block diagram of typical I/O pin.



**Figure 24.1** Typical I/O Pin Block Diagram ( $i = 0$  to  $10$ ;  $j = 0$  to  $7$ )

The registers to control I/O pins are as follows: port  $P_i$  direction register ( $PDi$  register), output function select registers, and pull-up control registers. The  $PDi$  register selects the input or output state of pins. The output function select registers which select output function consist of bits  $PSEL2$  to  $PSEL0$ , and  $ASEL$ . Bits  $PSEL2$  to  $PSEL0$  select a function as a programmable I/O or peripheral output (except analog output). The  $ASEL$  bit prevents the increase in power consumption of input buffer caused by an intermediate potential when a pin functions as an analog I/O pin. The pull-up control registers enable/disable the pull-up resistors. To use a pin as an analog I/O pin, set the  $PDi_j$  bit to 0 (input), bits  $PSEL2$  to  $PSEL0$  to 000b, and the  $ASEL$  bit to 1.

The input-only port  $P8\_5$  shares a pin with  $\overline{NMI}$  and has neither function select register nor the corresponding direction bit. Port  $P9\_1$  also functions as an input-only port. The function select register and bit 1 in the  $PD9$  register are reserved. Port  $P9$  is protected from unexpected write accesses by the  $PRC2$  bit in the  $PRCR$  register (refer to 10. "Protection").

## 24.1 Port Pi Direction Register (PDi Register, i = 0 to 10)

The PDi register selects the input or output state of pins. Bits in this register correspond to respective pins.

In memory expansion mode or microprocessor mode, this register cannot control pins being assigned bus control signals (A0 to A23, D0 to D15,  $\overline{CS0}$  to  $\overline{CS3}$ ,  $\overline{WR}/\overline{WR0}$ ,  $\overline{BC0}$ ,  $\overline{BC1}/\overline{WR1}$ ,  $\overline{RD}$ , CLKOUT/BCLK, HLDA,  $\overline{HOLD}$ , ALE, and  $\overline{RDY}$ ).

Figure 24.2 shows the PDi register.

No register bit is provided for port P8\_5. For port P9\_1, a reserved bit is provided.

The PD9 register is protected from unexpected write accesses by setting the PRC2 bit in the PRCR register (refer to 10. "Protection").

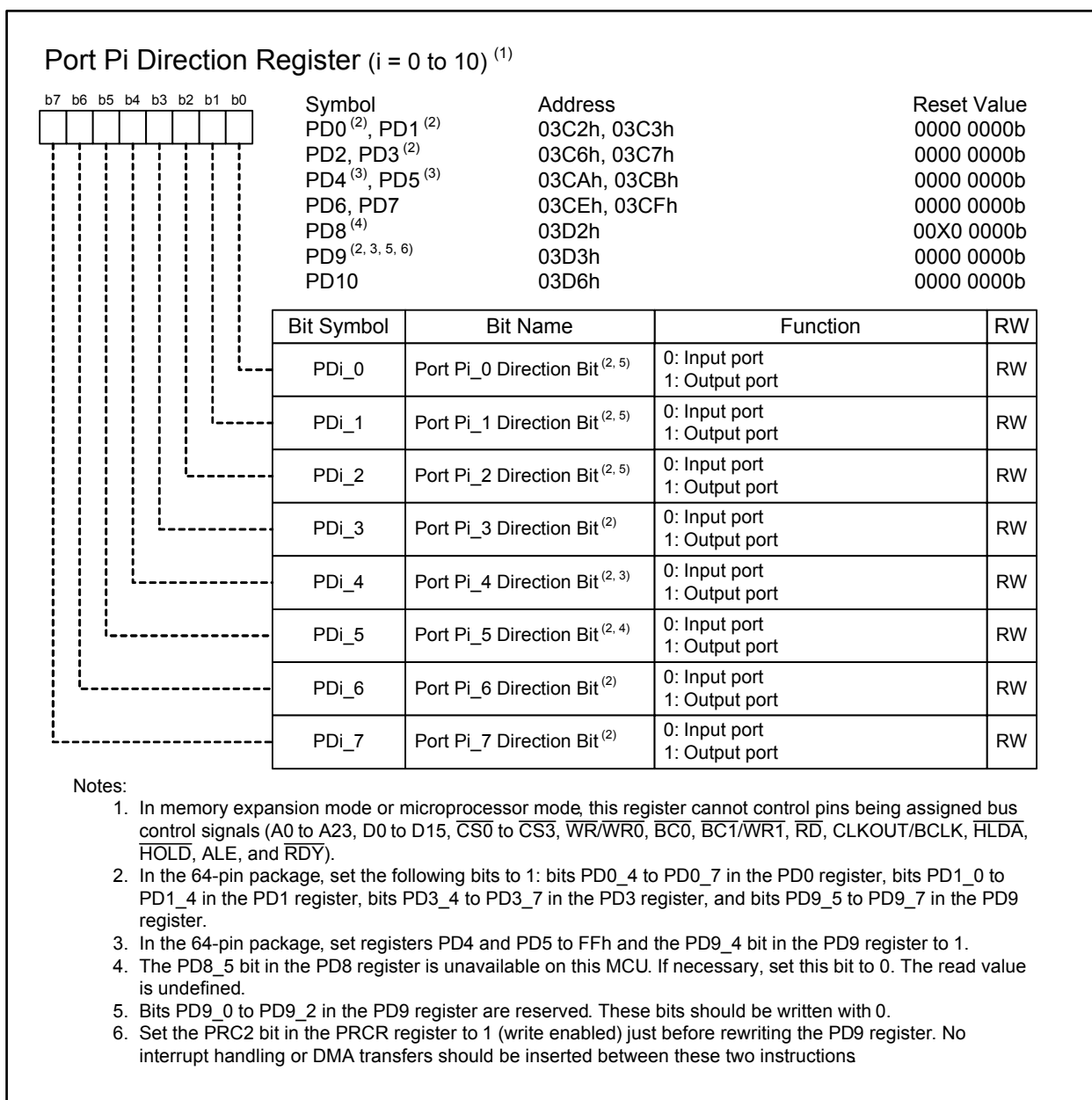


Figure 24.2 Registers PD0 to PD10

## 24.2 Output Function Select Registers

When a programmable I/O port and peripheral output share a pin, these registers select the output function of the pin. Regardless of the register settings, signals are input to all the connected peripherals. An output function select register consists of bits PSEL2 to PSEL0, and ASEL. Bits PSEL2 to PSEL0 select a function as programmable I/O or peripheral output (except analog output). The ASEL bit prevents the increase in power consumption caused by an intermediate potential generated when a pin functions as an analog I/O pin.

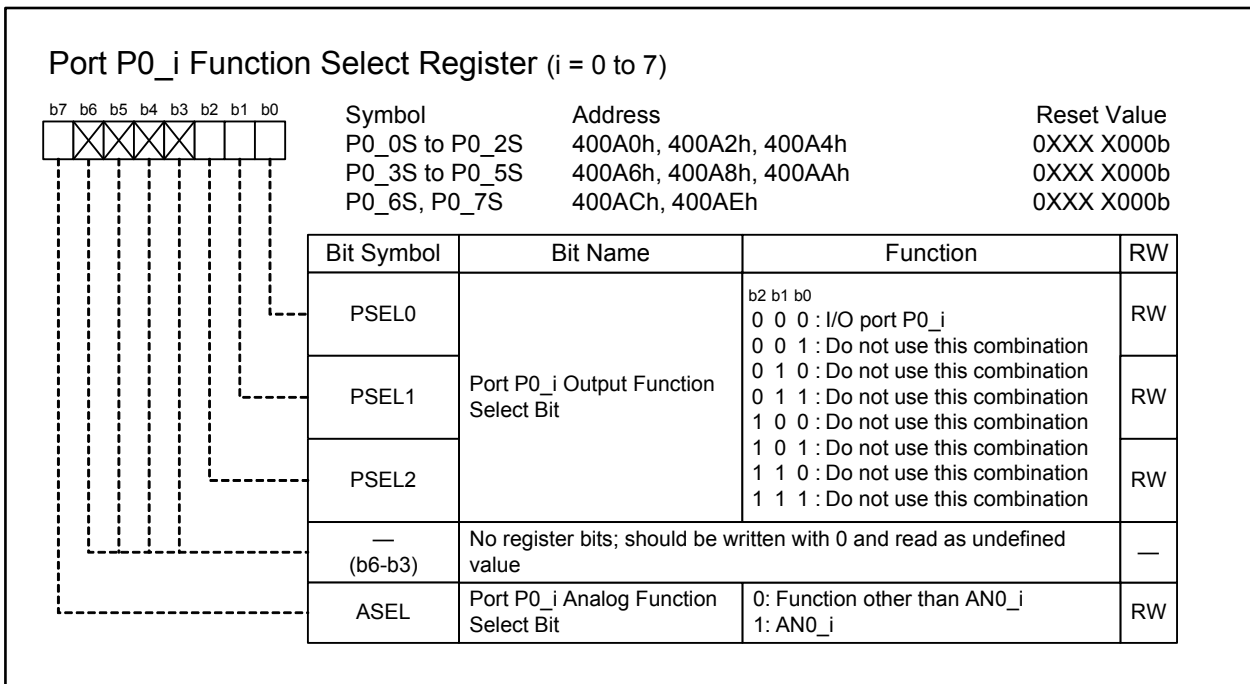
Table 24.1 shows the peripherals assigned to each PSEL2 to PSEL0 bit combination, and Figures 24.3 to 24.15 show the function select registers.

Note that ports P8\_5 and P9\_1 (input only) have no output function select registers.

The P9\_iS register is protected from unexpected write accesses by setting the PRC2 bit in the PRCR register (refer to 10. "Protection").

**Table 24.1 Peripheral Assignment**

Bits PSEL2 to PSEL0	Peripherals
001b	Timer
010b	Three-phase motor control timers
011b	UART
100b	UART special function
101b	Intelligent I/O groups 0 and 2
110b	Intelligent I/O group 1
111b	UART8

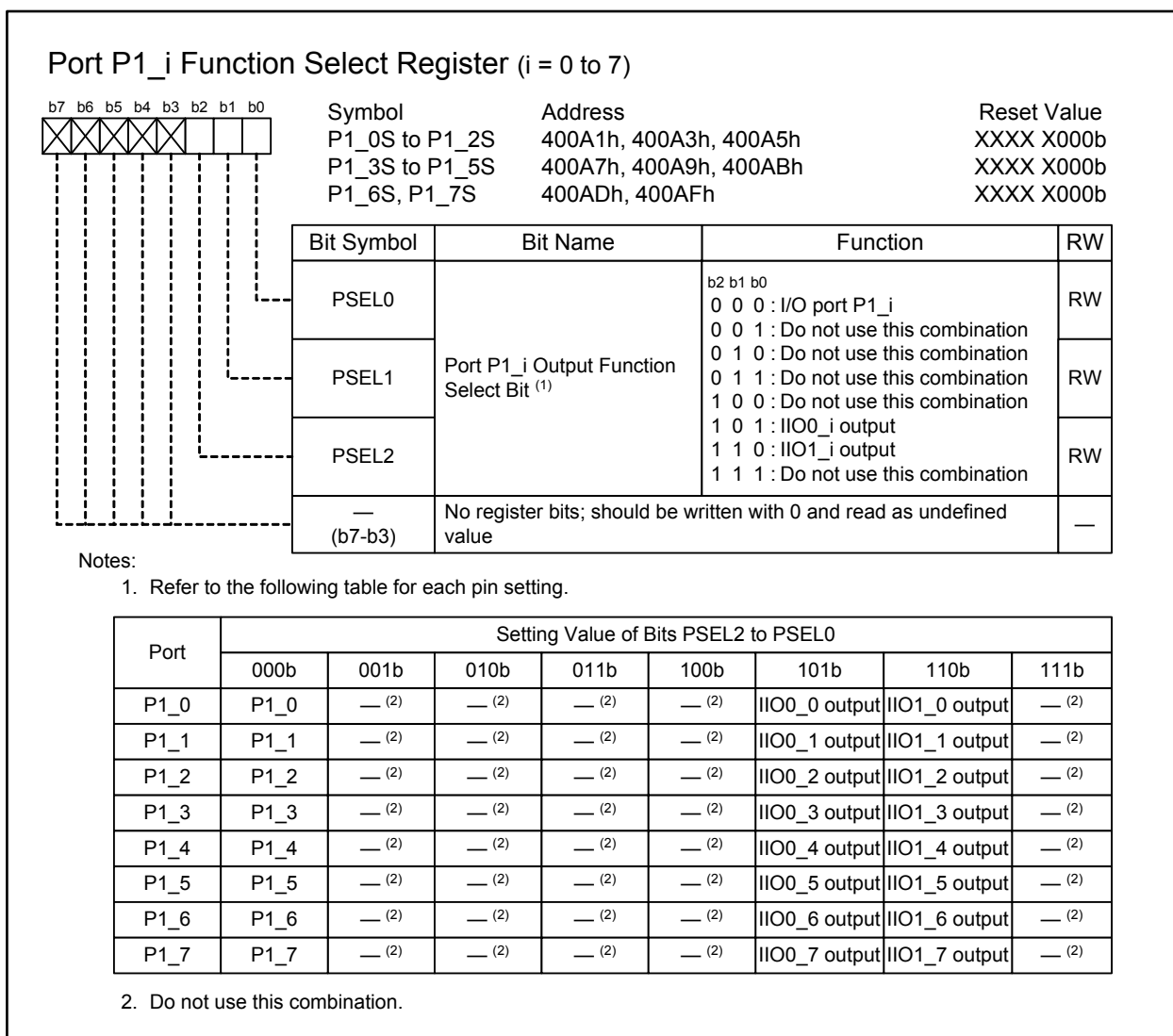


**Figure 24.3 Registers P0\_0S to P0\_7S**

Port P0<sub>i</sub> shares a pin with the AN0<sub>i</sub> input for the A/D converter (i = 0 to 7).

To use it as a programmable I/O port, set the P0<sub>i</sub>S register to 00h. To use it as an A/D converter input pin, set this register to 80h and the PD0<sub>i</sub> bit to 0 (port P0<sub>i</sub> functions as an input port).

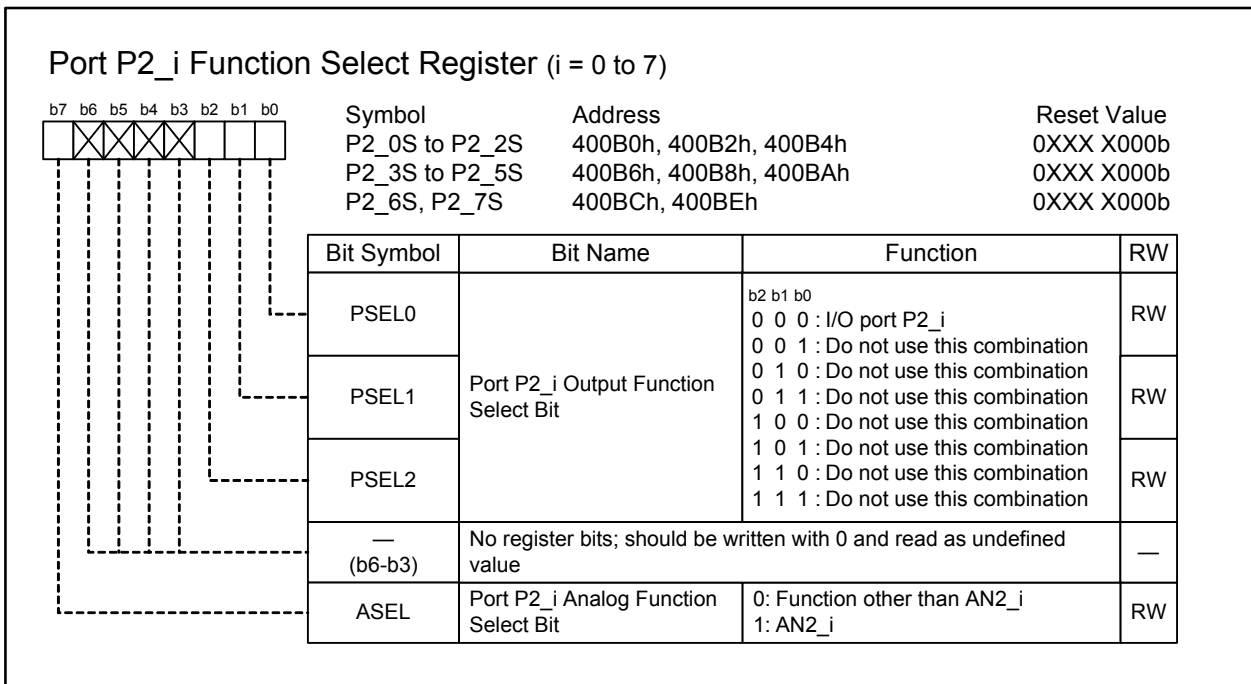
Registers P0\_4S to P0\_7S are reserved in the 64-pin package. Do not change the values in these registers.



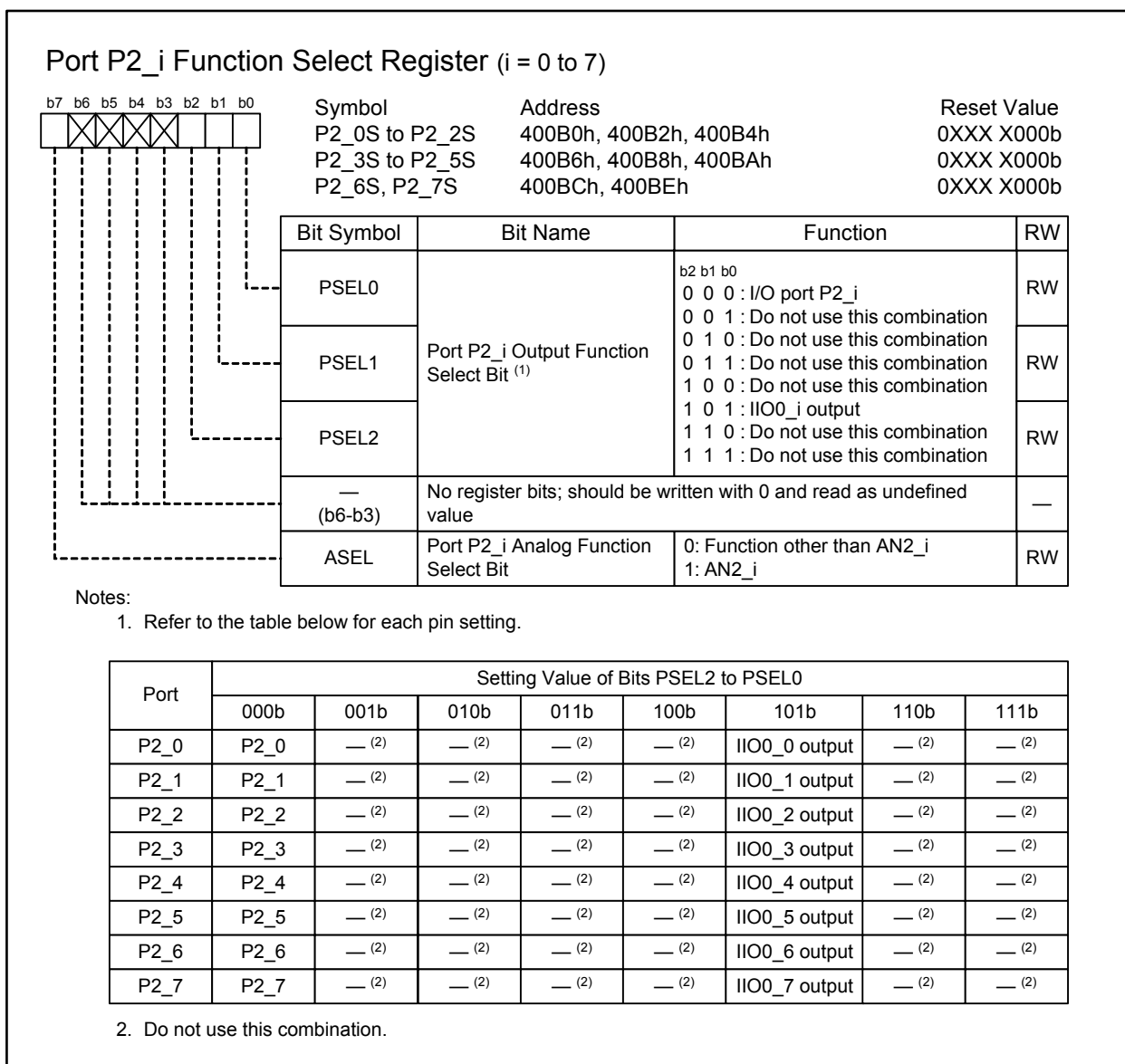
**Figure 24.4 Registers P1\_0S to P1\_7S**

Port P1<sub>i</sub> shares a pin with intelligent I/O groups 0 and 1 (IIO0 and IIO1) and the external interrupt input (i = 0 to 7).

To use it as an output pin, set the PD1<sub>i</sub> bit to 1 (port P1<sub>i</sub> functions as an output port) and select a function according to Figure 24.4. To use it as an input pin, set the PD1<sub>i</sub> bit to 0 (port P1<sub>i</sub> functions as an input port). Registers P1\_0S to P1\_4S are reserved in the 64-pin package. Do not change the values in these registers.



**Figure 24.5 Registers P2\_0S to P2\_7S (100-pin package)**

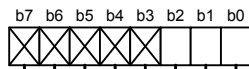


**Figure 24.6 Registers P2\_0S to P2\_7S (64-pin package)**

Port P2<sub>i</sub> shares a pin with the AN2<sub>i</sub> for the A/D converter (i = 0 to 7). This port also shares a pin with intelligent I/O group 0 (IIO0) in the 64-pin package.

To use it as a programmable I/O port, set the P2<sub>i</sub>S register to 00h. To use it as an output pin in the 64-pin package, set the PD2<sub>i</sub> bit to 1 (port P2<sub>i</sub> functions as an output port) and select a function according to Figure 24.6. To use it as an input pin of functions other than the A/D converter, set the PD2<sub>i</sub> bit to 0 (port P2<sub>i</sub> functions as an input port). To use it as an A/D converter input pin, set this register to 80h and the PD2<sub>i</sub> bit to 0 (port P2<sub>i</sub> functions as an input port).



Port P3<sub>i</sub> Function Select Register (i = 0 to 7)

Symbol	Address	Reset Value
P3_0S to P3_2S	400B1h, 400B3h, 400B5h	XXXX X000b
P3_3S to P3_5S	400B7h, 400B9h, 400BBh	XXXX X000b
P3_6S, P3_7S	400BDh, 400BFh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P3 <sub>i</sub> Output Function Select Bit <sup>(1)</sup>	b2 b1 b0 0 0 0 : I/O port P3 <sub>i</sub> 0 0 1 : Timer output 0 1 0 : Three-phase motor control output	RW
PSEL1		0 1 1 : Do not use this combination 1 0 0 : Do not use this combination	RW
PSEL2		1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
— (b7-b3)		No register bits; should be written with 0 and read as undefined value	—

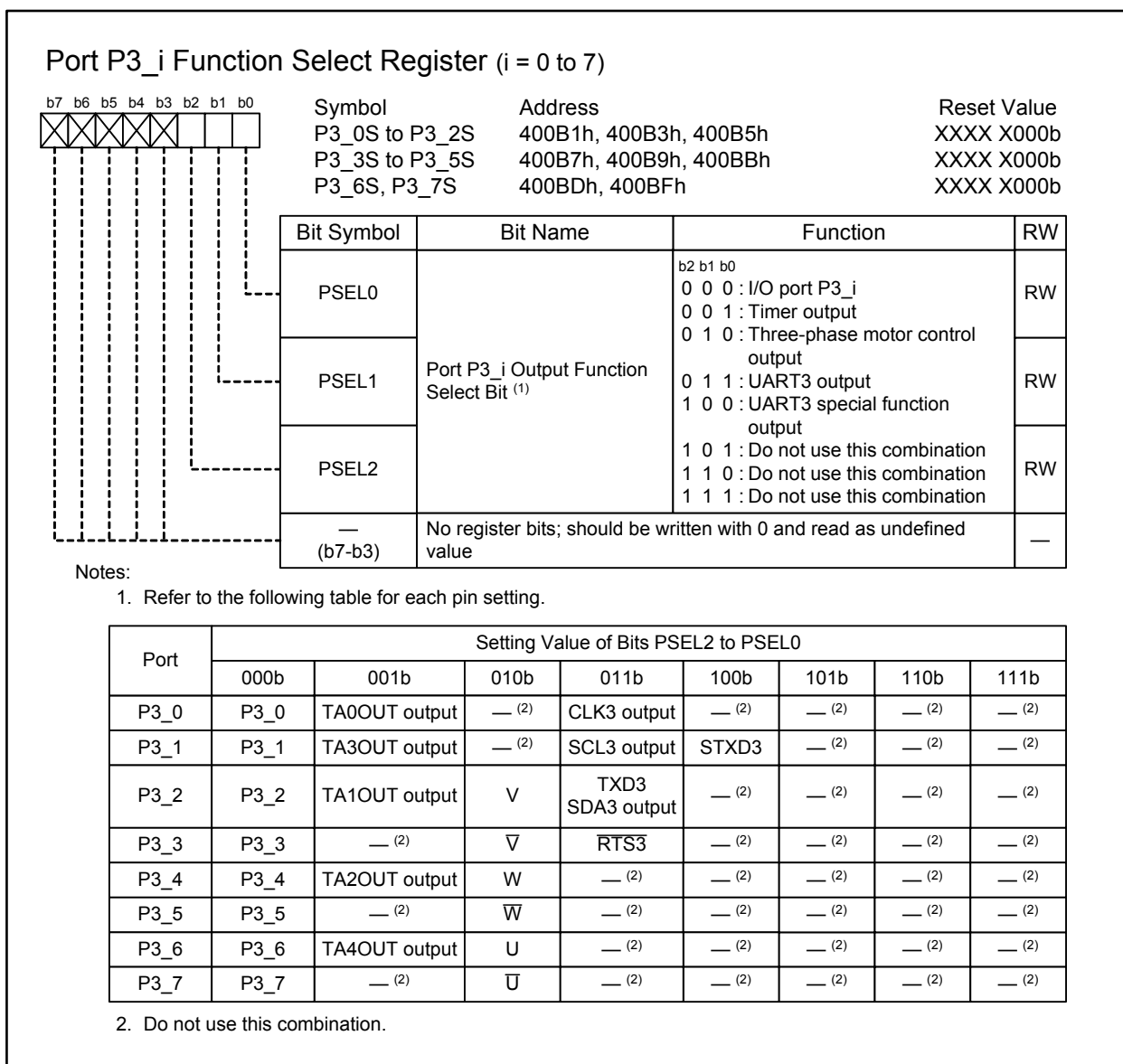
## Notes:

1. Refer to the following table for each pin setting.

Port	Setting Value of Bits PSEL2 to PSEL0							
	000b	001b	010b	011b	100b	101b	110b	111b
P3_0	P3_0	TA0OUT output	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>
P3_1	P3_1	TA3OUT output	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>
P3_2	P3_2	TA1OUT output	V	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>
P3_3	P3_3	— <sup>(2)</sup>	$\bar{V}$	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>
P3_4	P3_4	TA2OUT output	W	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>
P3_5	P3_5	— <sup>(2)</sup>	$\bar{W}$	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>
P3_6	P3_6	TA4OUT output	U	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>
P3_7	P3_7	— <sup>(2)</sup>	$\bar{U}$	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>

2. Do not use this combination.

Figure 24.7 Registers P3\_0S to P3\_7S (100-pin package)

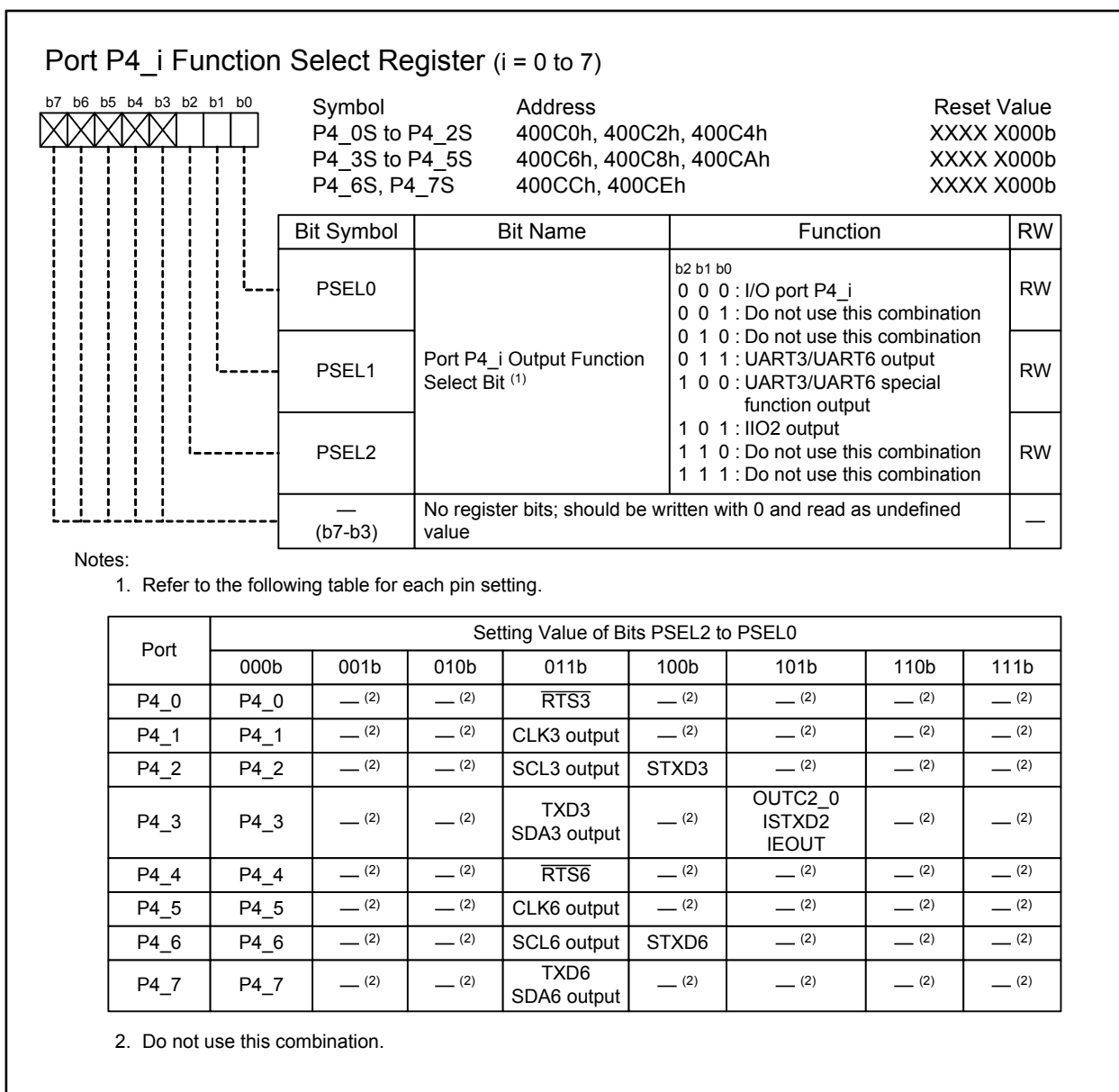


**Figure 24.8 Registers P3\_0S to P3\_7S (64-pin package)**

Port P3<sub>i</sub> shares a pin with the timer output and three-phase motor control output (i = 0 to 7). This port also shares a pin with the serial interface (UART3) in the 64-pin package.

To use it as an output pin, set the PD3<sub>i</sub> bit to 1 (port P3<sub>i</sub> functions as an output port) and select a function according to Figures 24.7 and 24.8. To use it as an input pin, set the PD3<sub>i</sub> bit to 0 (port P3<sub>i</sub> functions as an input port).

Registers P3\_4S to P3\_7S are reserved in the 64-pin package. Do not change the values in these registers.

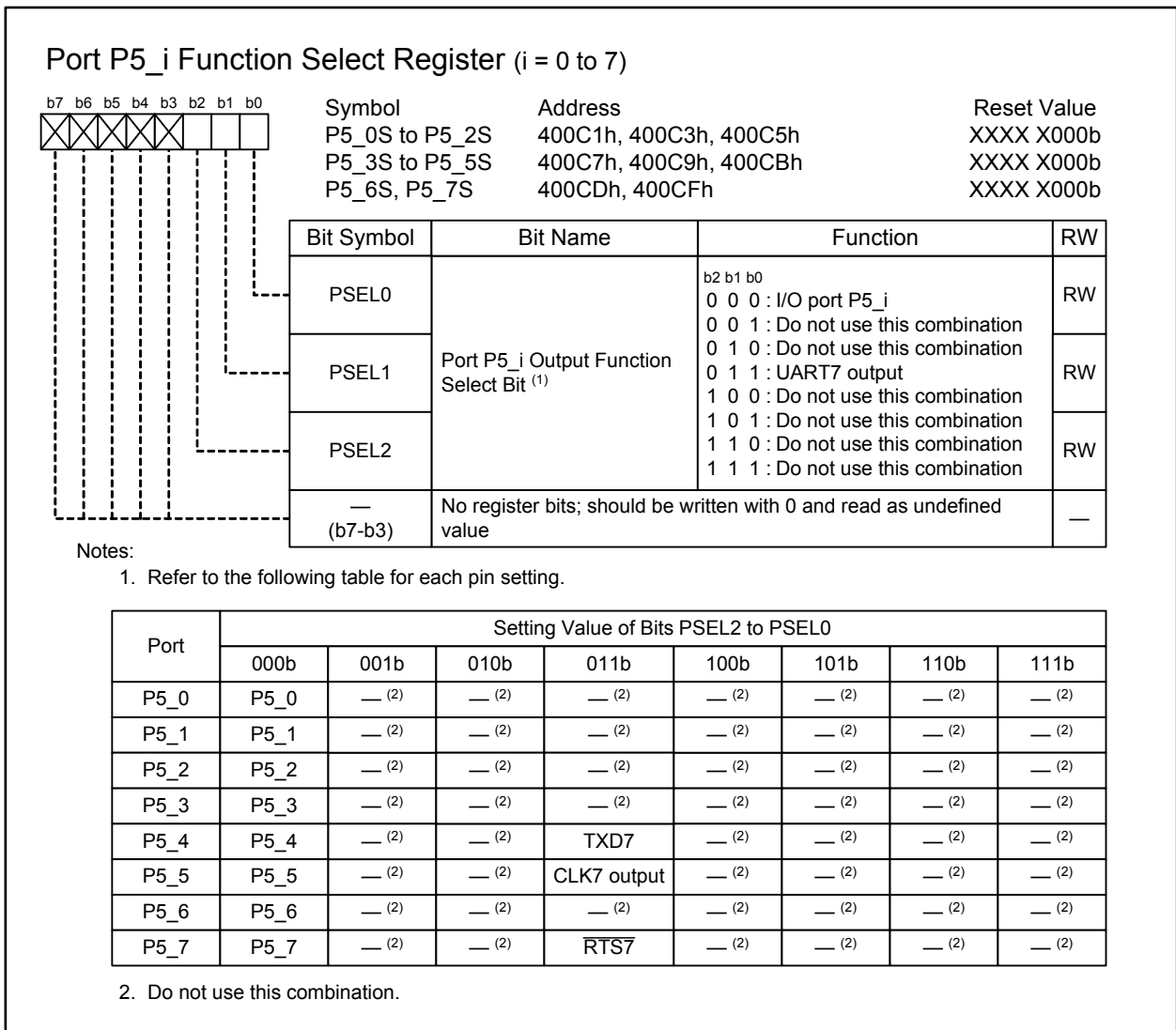


**Figure 24.9 Registers P4\_0S to P4\_7S**

Port P4<sub>i</sub> shares a pin with the serial interface (UART3 and UART6) and intelligent I/O group 2 (IIO2) (i = 0 to 7).

To use it as an output pin, set the PD4<sub>i</sub> bit to 1 (port P4<sub>i</sub> functions as an output port) and select a function according to Figure 24.9. To use it as an input pin, set the PD4<sub>i</sub> bit to 0 (port P4<sub>i</sub> functions as an input port).

Registers P4\_0S to P4\_7S are reserved in the 64-pin package. Do not change the values in these registers.

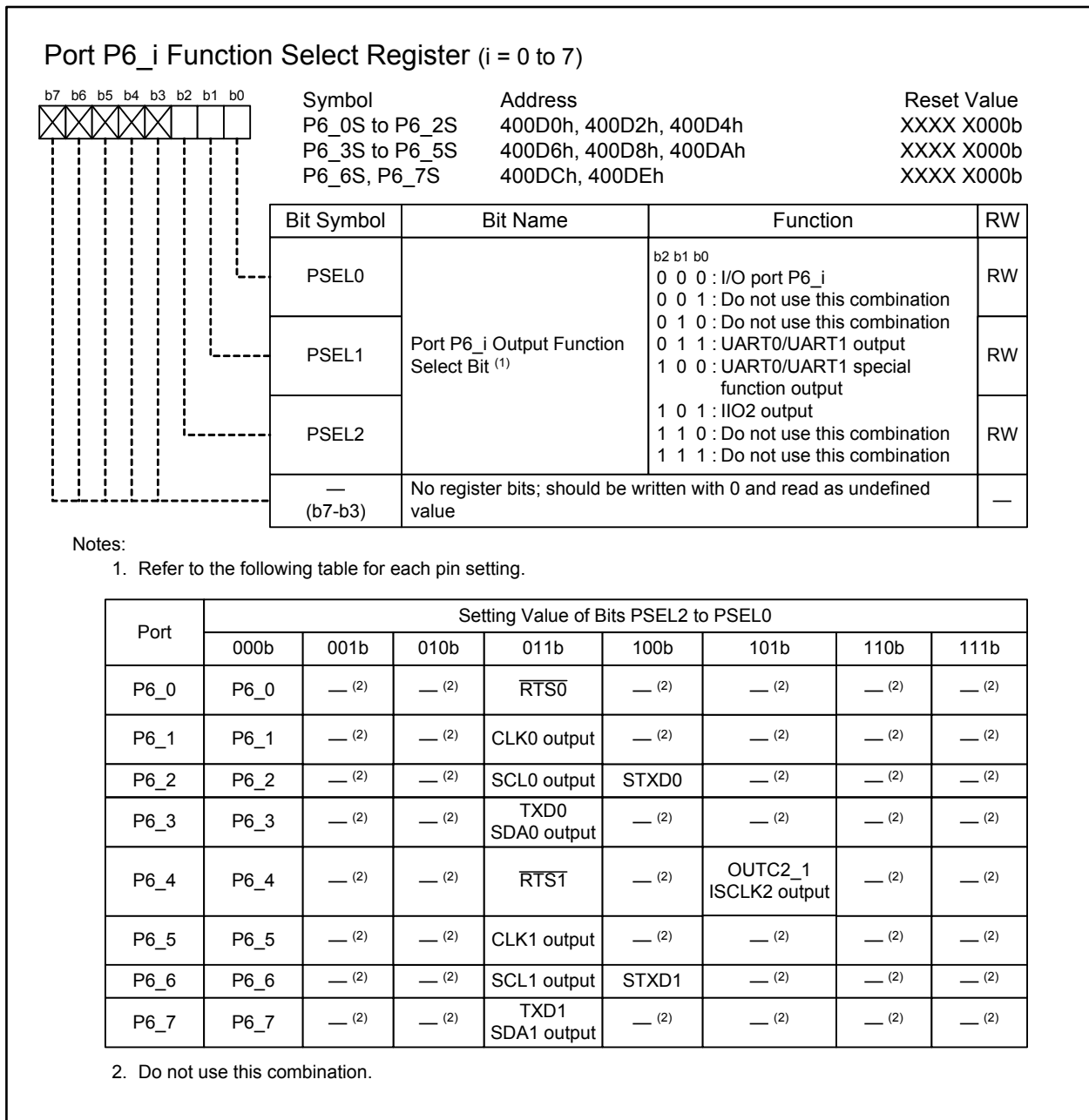


**Figure 24.10 Registers P5\_0S to P5\_7S**

Port P5<sub>i</sub> shares a pin with the serial interface (UART7) (i = 0 to 7).

To use it as an output pin, set the PD5<sub>i</sub> bit to 1 (port P5<sub>i</sub> functions as an output port) and select a function according to Figure 24.10. To use it as an input pin, set the PD5<sub>i</sub> bit to 0 (port P5<sub>i</sub> functions as an input port).

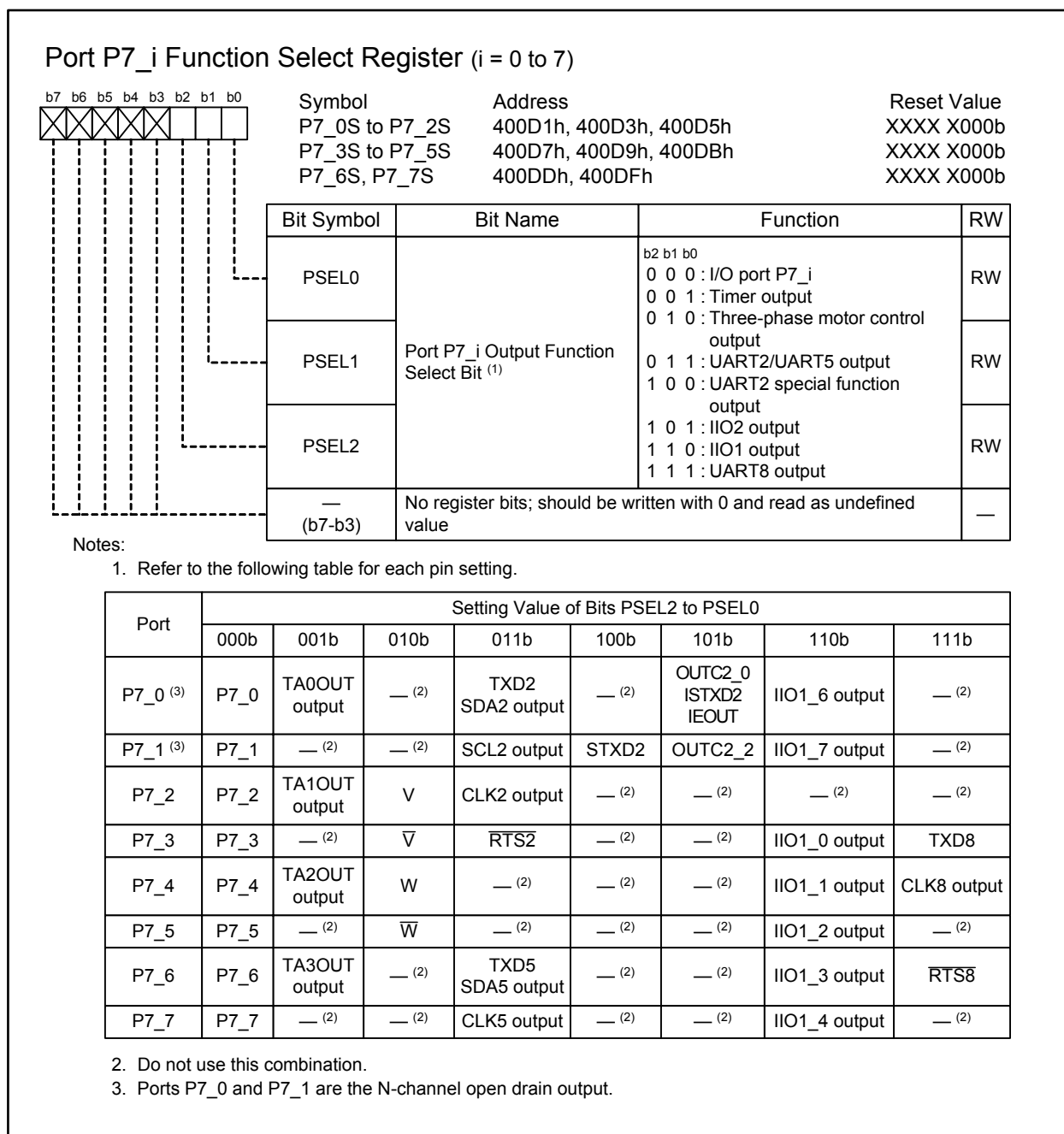
Registers P5\_0S to P5\_7S are reserved in the 64-pin package. Do not change the values in these registers.



**Figure 24.11 Registers P6\_0S to P6\_7S**

Port P6<sub>i</sub> shares a pin with the serial interface (UART0 and UART1) and intelligent I/O group 2 (IIO2) (i = 0 to 7).

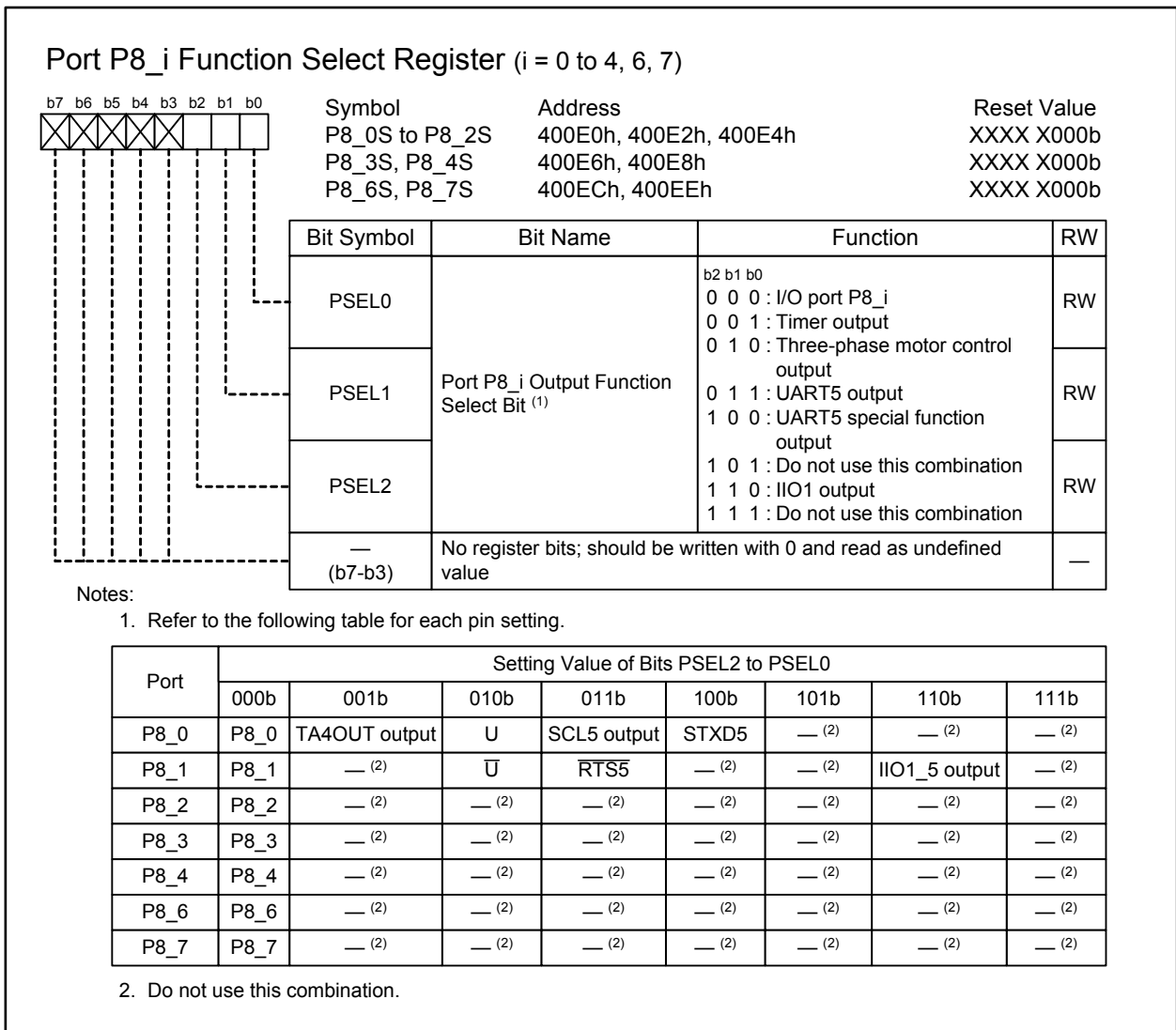
To use it as an output pin, set the PD6<sub>i</sub> bit to 1 (port P6<sub>i</sub> functions as an output port) and select a function according to Figure 24.11. To use it as an input pin, set the PD6<sub>i</sub> bit to 0 (port P6<sub>i</sub> functions as an input port).



**Figure 24.12 Registers P7\_0S to P7\_7S**

Port P7<sub>i</sub> shares a pin with the timer, three-phase motor control, serial interface (UART2, UART5, and UART8), and intelligent I/O groups 1 and 2 (IIO1 and IIO2) (i = 0 to 7).

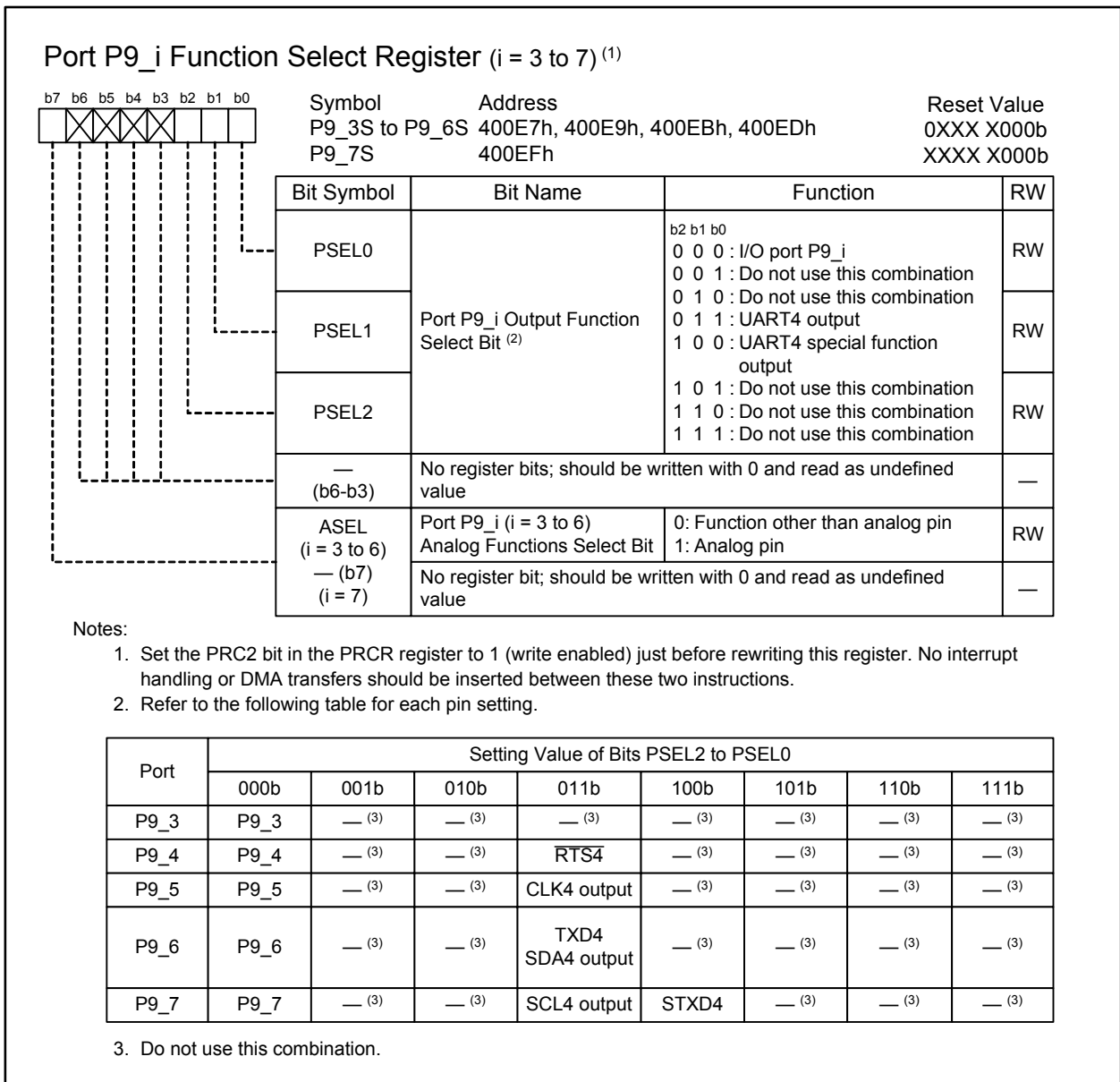
To use it as an output pin, set the PD7<sub>i</sub> bit to 1 (port P7<sub>i</sub> functions as an output port) and select a function according to Figure 24.12. To use it as an input pin, set the PD7<sub>i</sub> bit to 0 (port P7<sub>i</sub> functions as an input port).



**Figure 24.13 Registers P8\_0S to P8\_4S, P8\_6S, and P8\_7S**

Port P8<sub>i</sub> shares a pin with the timer, three-phase motor control, serial interface (UART5), intelligent I/O group 1 (IIO1), and external interrupt input (i = 0 to 4, 6, 7).

To use it as an output pin, set the PD8<sub>i</sub> bit to 1 (port P8<sub>i</sub> functions as an output port) and select a function according to Figure 24.13. To use it as an input pin, set the PD8<sub>i</sub> bit to 0 (port P8<sub>i</sub> functions as an input port).



**Figure 24.14 Registers P9\_3S to P9\_7S**

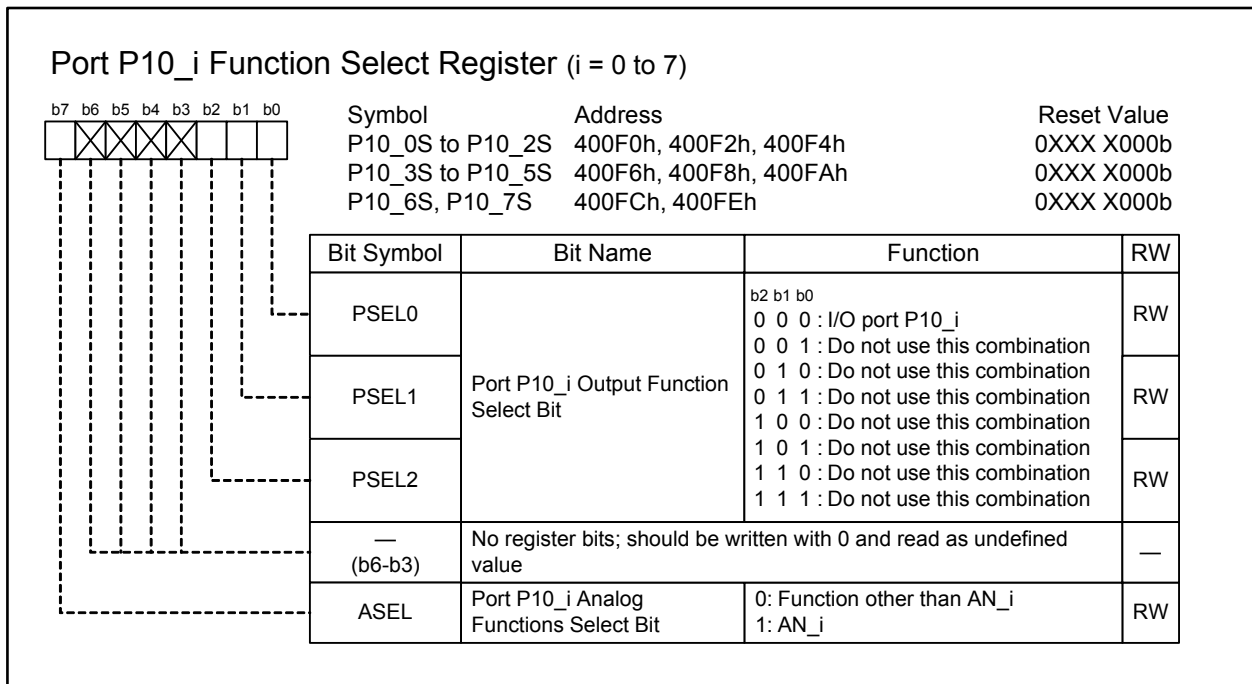
Port P9<sub>i</sub> shares a pin with the serial interface (UART4) (i = 3 to 7). Ports P9\_3 to P9\_6 also share a pin with the A/D converter I/O (ANEX0 and ANEX1) and D/A converter output.

To use it as the A/D converter pin or the D/A converter pin, set the P9<sub>i</sub>S register to 80h and the PD9<sub>i</sub> bit to 0 (port P9<sub>i</sub> functions as an input port) irrespective of the I/O state.

To use it as an output pin for functions other than the A/D converter or the D/A converter, set the PD9<sub>i</sub> bit to 1 (port P9<sub>i</sub> functions as an output port) and select a function according to Figure 24.14. To use it as an input pin of functions other than the A/D converter or the D/A converter, set the PD9<sub>i</sub> bit to 0 (port P9<sub>i</sub> functions as an input port).

Registers P9\_4S to P9\_7S in the 64-pin package are reserved. Do not change the values in these registers.





**Figure 24.15 Registers P10\_0S to P10\_7S**

Port P10<sub>i</sub> shares a pin with the AN<sub>i</sub> input for the A/D converter and key input interrupt (i = 0 to 7).

To use it as a programmable I/O port, set the P10<sub>i</sub>S register to 00h. To use it as an input pin (except for the A/D converter), set the PD10<sub>i</sub> bit to 0 (port P10<sub>i</sub> functions as an input port). To use it as an input pin for the A/D converter, set the P10<sub>i</sub>S register to 80h and the PD10<sub>i</sub> bit to 0 (port P10<sub>i</sub> functions as an input port).

### 24.3 Input Function Select Registers

When a peripheral input is assigned to multiple pins, these registers select which input pin should be connected to the peripheral.

Figures 24.16 to 24.21 show the input function select registers.

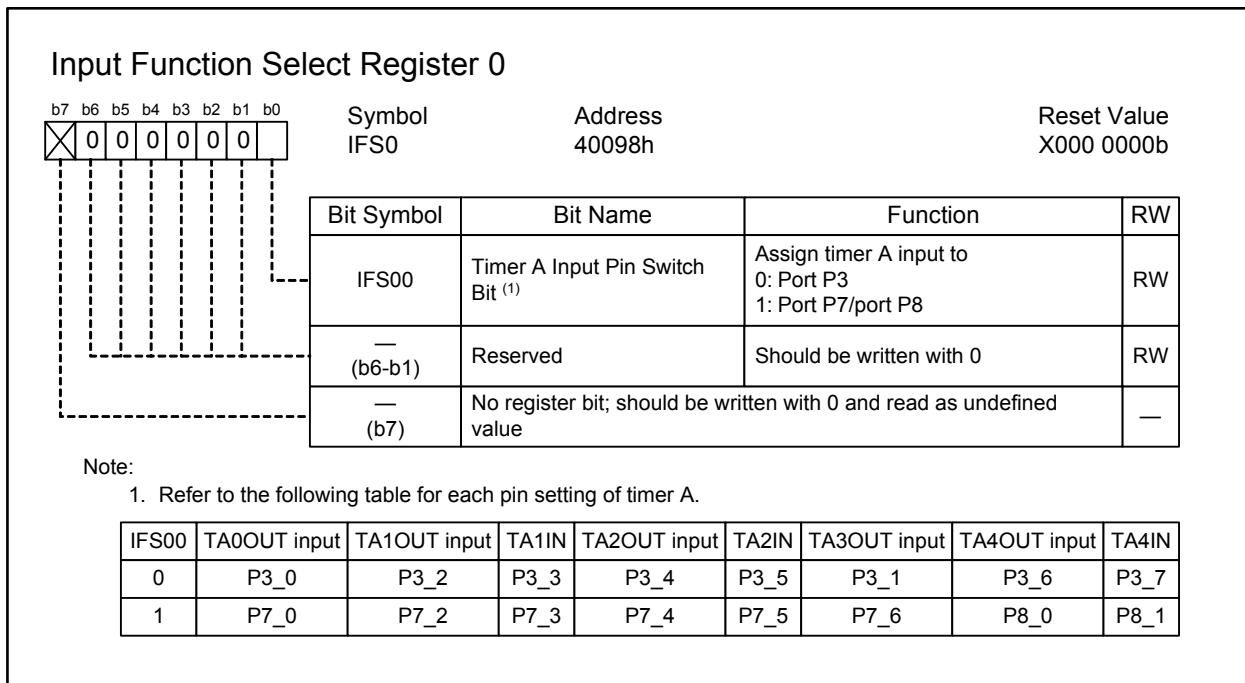


Figure 24.16 IFS0 Register (100-pin package)

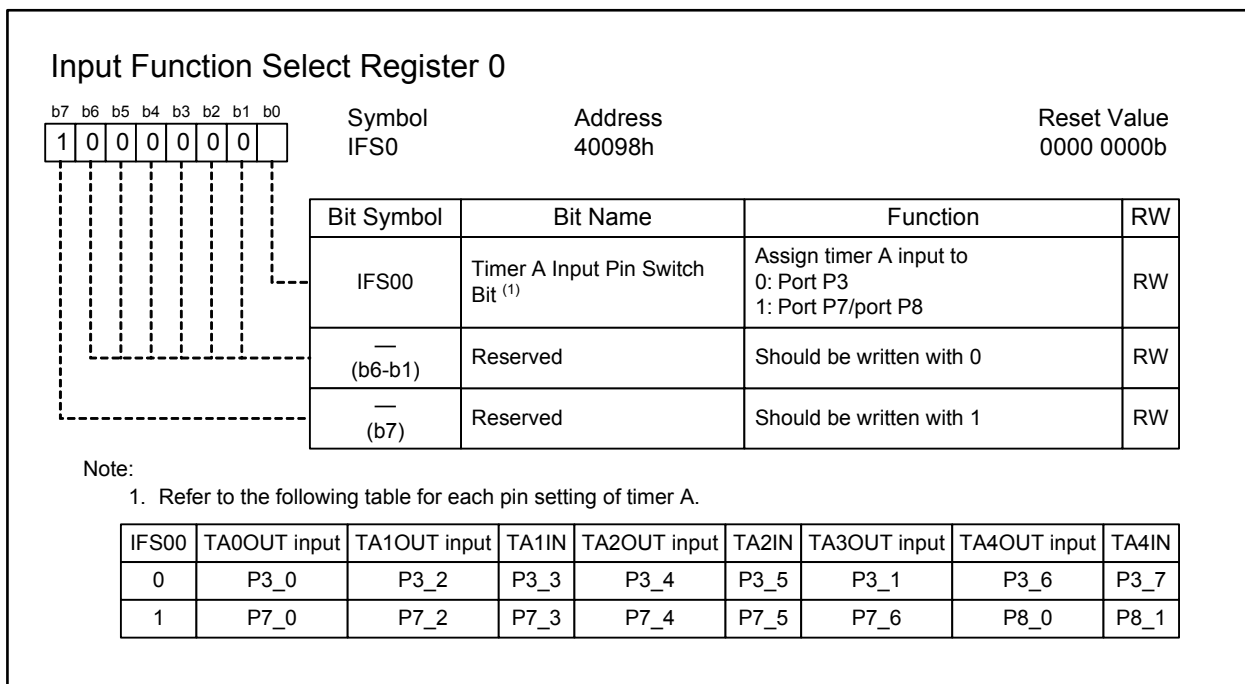


Figure 24.17 IFS0 Register (80-/64-pin package)

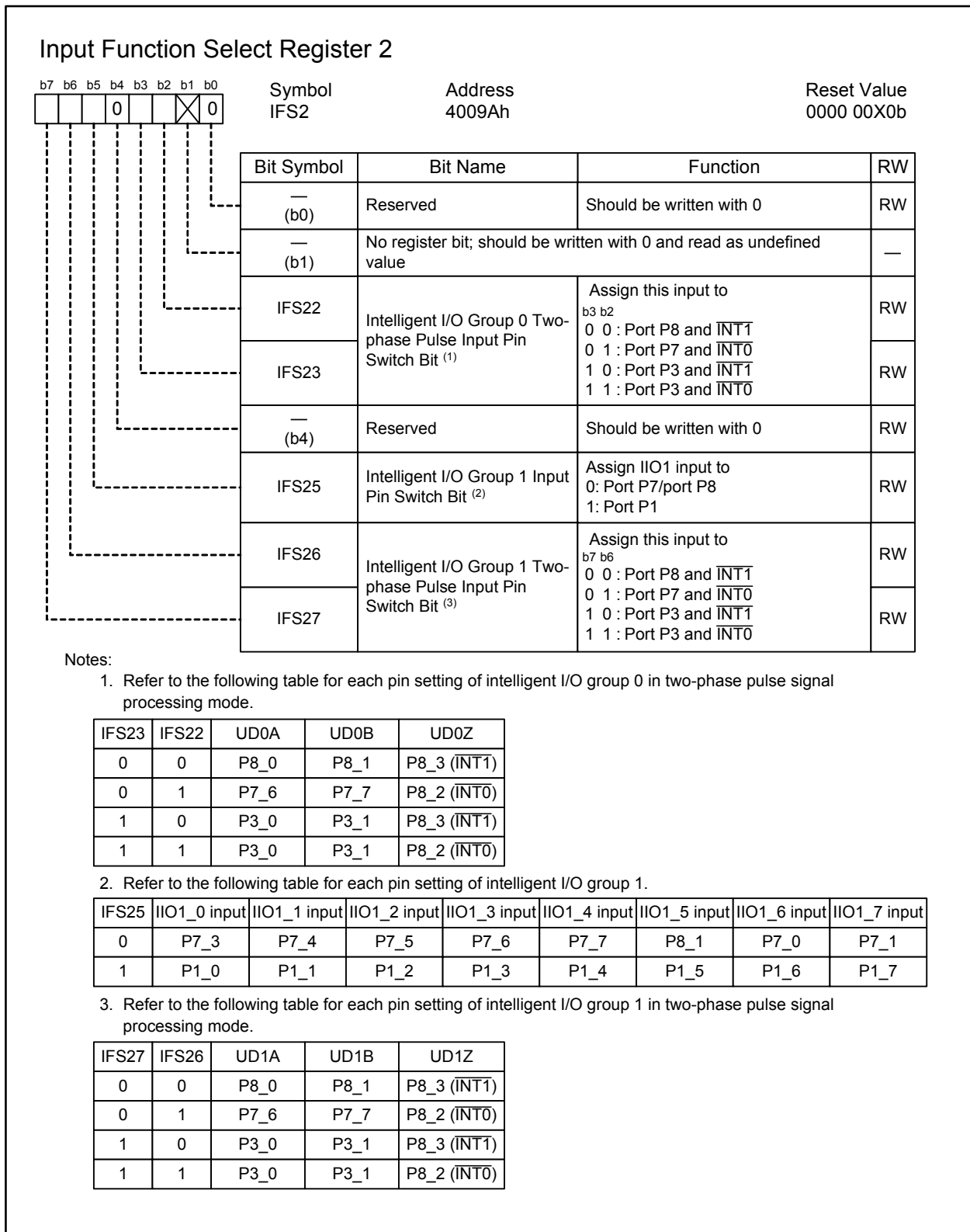
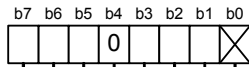


Figure 24.18 IFS2 Register (100-pin package)

## Input Function Select Register 2

b7	b6	b5	b4	b3	b2	b1	b0	Symbol IFS2	Address 4009Ah	Reset Value 0000 000Xb
----	----	----	----	----	----	----	----	----------------	-------------------	---------------------------



Bit Symbol	Bit Name	Function	RW
— (b0)	No register bit; should be written with 0 and read as undefined value		—
IFS21	Intelligent I/O Group 0 Input Pin Switch Bit <sup>(1)</sup>	Assign IIO0 input to 0: Port P1 1: Port P2	RW
IFS22	Intelligent I/O Group 0 Two-Phase Pulse Input Pin Switch Bit <sup>(2)</sup>	Assign this input to b3 b2 0 0 : Port P8 and $\overline{\text{INT1}}$ 0 1 : Port P7 and $\overline{\text{INT0}}$ 1 0 : Port P3 and $\overline{\text{INT1}}$ 1 1 : Port P3 and $\overline{\text{INT0}}$	RW
IFS23			RW
— (b4)	Reserved	Should be written with 0	RW
IFS25	Intelligent I/O Group 1 Input Pin Switch Bit <sup>(3)</sup>	Assign IIO1 input to 0: Port P7/port P8 1: Port P1	RW
IFS26	Intelligent I/O Group 1 Two-Phase Pulse Input Pin Switch Bit <sup>(4)</sup>	Assign this input to b7 b6 0 0 : Port P8 and $\overline{\text{INT1}}$ 0 1 : Port P7 and $\overline{\text{INT0}}$ 1 0 : Port P3 and $\overline{\text{INT1}}$ 1 1 : Port P3 and $\overline{\text{INT0}}$	RW
IFS27			RW

## Notes:

1. Refer to the following table for each pin setting of intelligent I/O group 0.

IFS21	IIO0_0 input	IIO0_1 input	IIO0_2 input	IIO0_3 input	IIO0_4 input	IIO0_5 input	IIO0_6 input	IIO0_7 input
0	—	—	—	—	—	P1_5	P1_6	P1_7
1	P2_0	P2_1	P2_2	P2_3	P2_4	P2_5	P2_6	P2_7

2. Refer to the following table for each pin setting of intelligent I/O group 0 in two-phase pulse signal processing mode.

IFS23	IFS22	UD0A	UD0B	UD0Z
0	0	P8_0	P8_1	P8_3 ( $\overline{\text{INT1}}$ )
0	1	P7_6	P7_7	P8_2 ( $\overline{\text{INT0}}$ )
1	0	P3_0	P3_1	P8_3 ( $\overline{\text{INT1}}$ )
1	1	P3_0	P3_1	P8_2 ( $\overline{\text{INT0}}$ )

3. Refer to the following table for each pin setting of intelligent I/O group 1.

IFS25	IIO1_0 input	IIO1_1 input	IIO1_2 input	IIO1_3 input	IIO1_4 input	IIO1_5 input	IIO1_6 input	IIO1_7 input
0	P7_3	P7_4	P7_5	P7_6	P7_7	P8_1	P7_0	P7_1
1	P1_0	P1_1	P1_2	P1_3	P1_4	P1_5	P1_6	P1_7

4. Refer to the following table for each pin setting of intelligent I/O group 1 in two-phase pulse signal processing mode.

IFS27	IFS26	UD1A	UD1B	UD1Z
0	0	P8_0	P8_1	P8_3 ( $\overline{\text{INT1}}$ )
0	1	P7_6	P7_7	P8_2 ( $\overline{\text{INT0}}$ )
1	0	P3_0	P3_1	P8_3 ( $\overline{\text{INT1}}$ )
1	1	P3_0	P3_1	P8_2 ( $\overline{\text{INT0}}$ )

Figure 24.19 IFS2 Register (64-pin package)

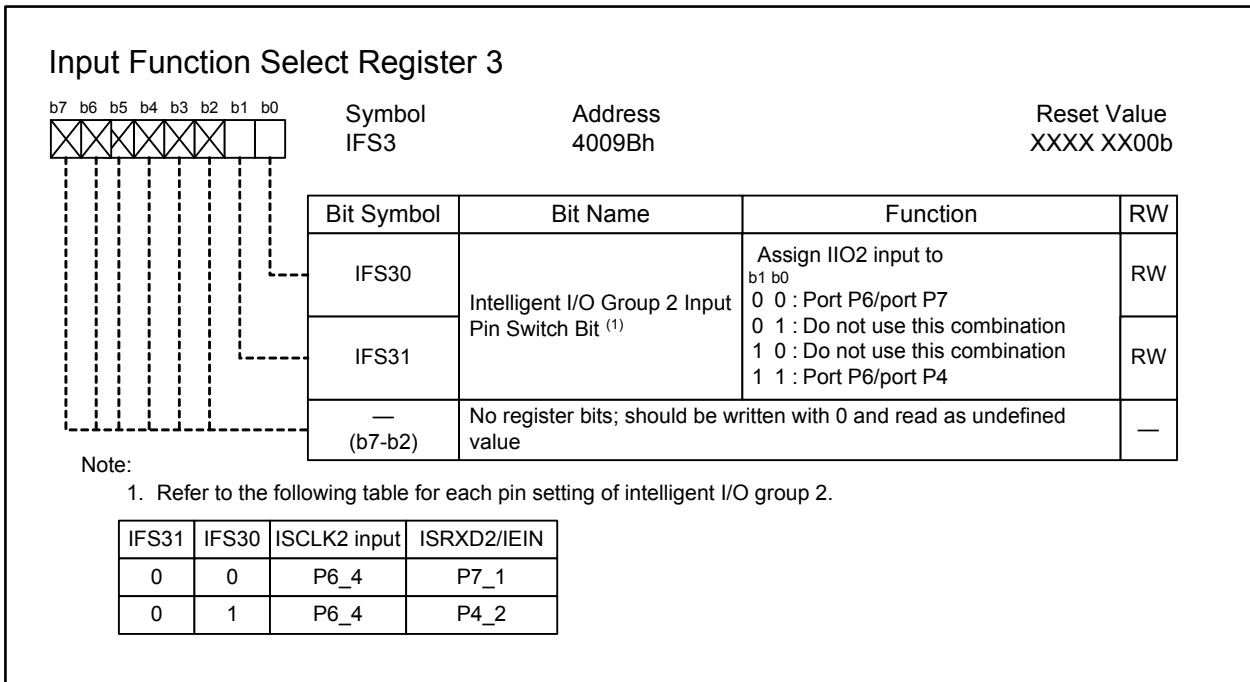


Figure 24.20 IFS3 Register

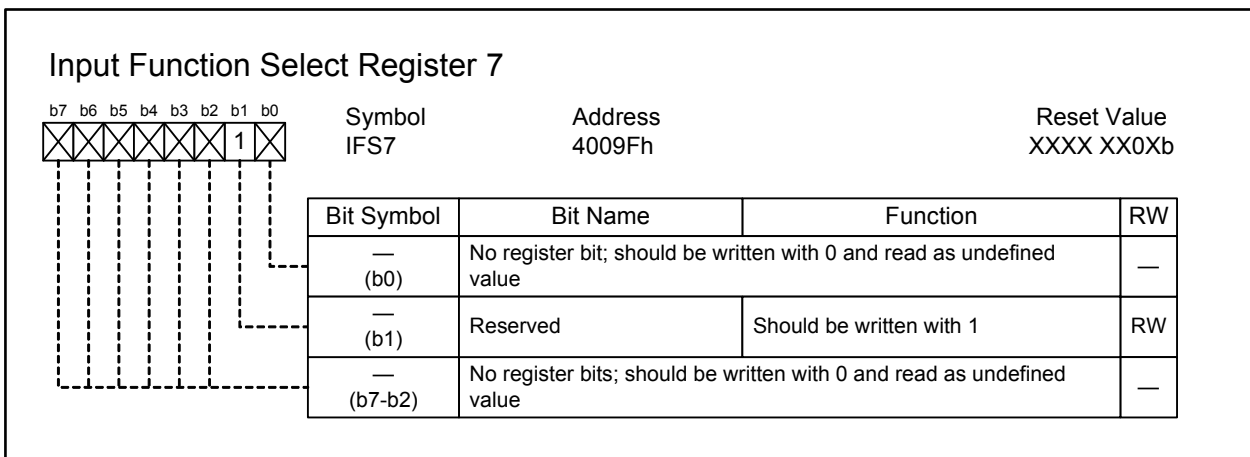


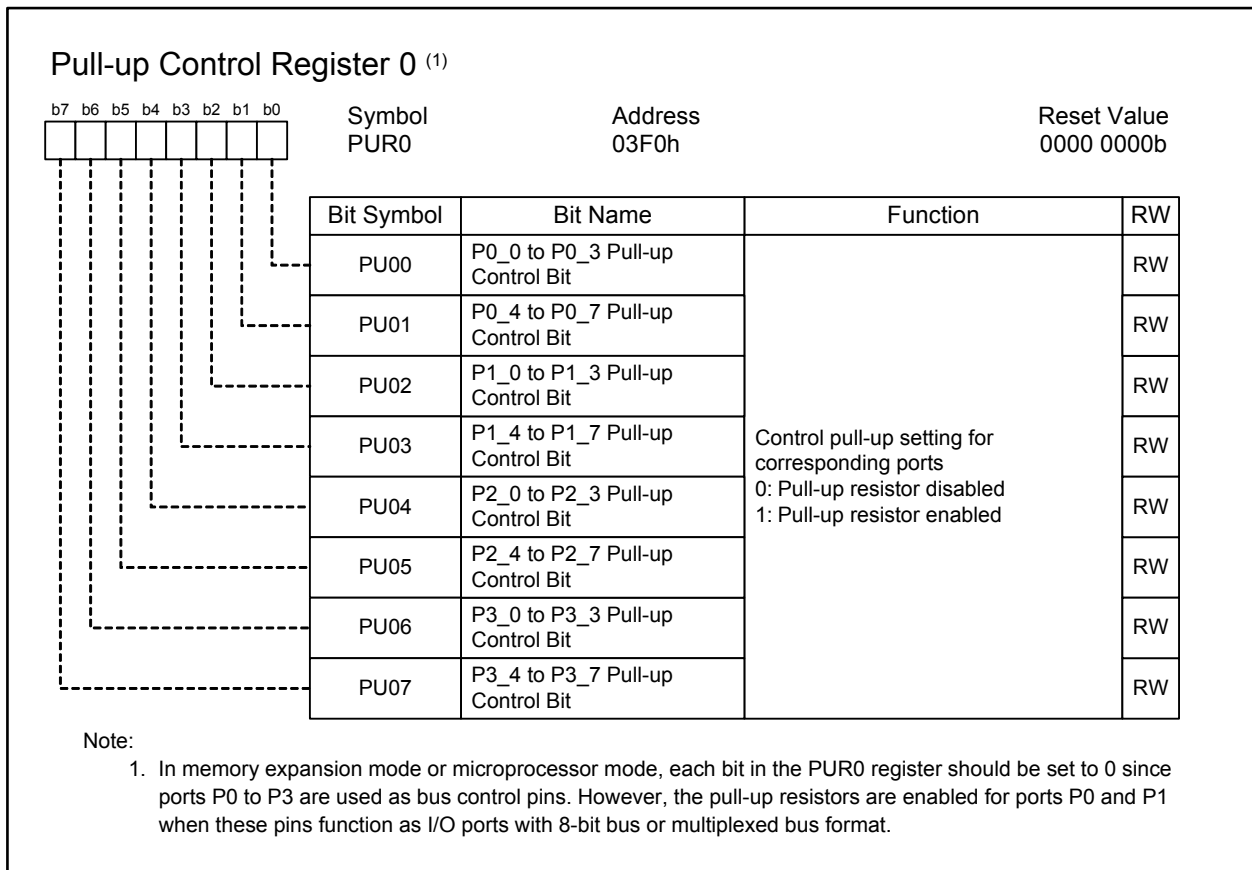
Figure 24.21 IFS7 Register (64-pin package)

## 24.4 Pull-up Control Registers 0 to 3 (Registers PUR0 to PUR3)

Figures 24.22 to 24.25 show registers PUR0 to PUR3.

These registers enable/disable the pull-up resistors for every group of four pins. To enable the pull-up resistors, set the corresponding bits in registers PUR0 to PUR3 to 1 (pull-up resistor enabled) and the respective bits in the direction register to 0 (input).

In memory expansion mode or microprocessor mode, set 0 (pull-up resistor disabled) to the pull-up control bits for ports P0 to P5 operating as bus control pins. The pull-up resistors are enabled for ports P0, P1, and P4\_0 to P4\_3 when these pins function as input ports in these modes.



**Figure 24.22 PUR0 Register**

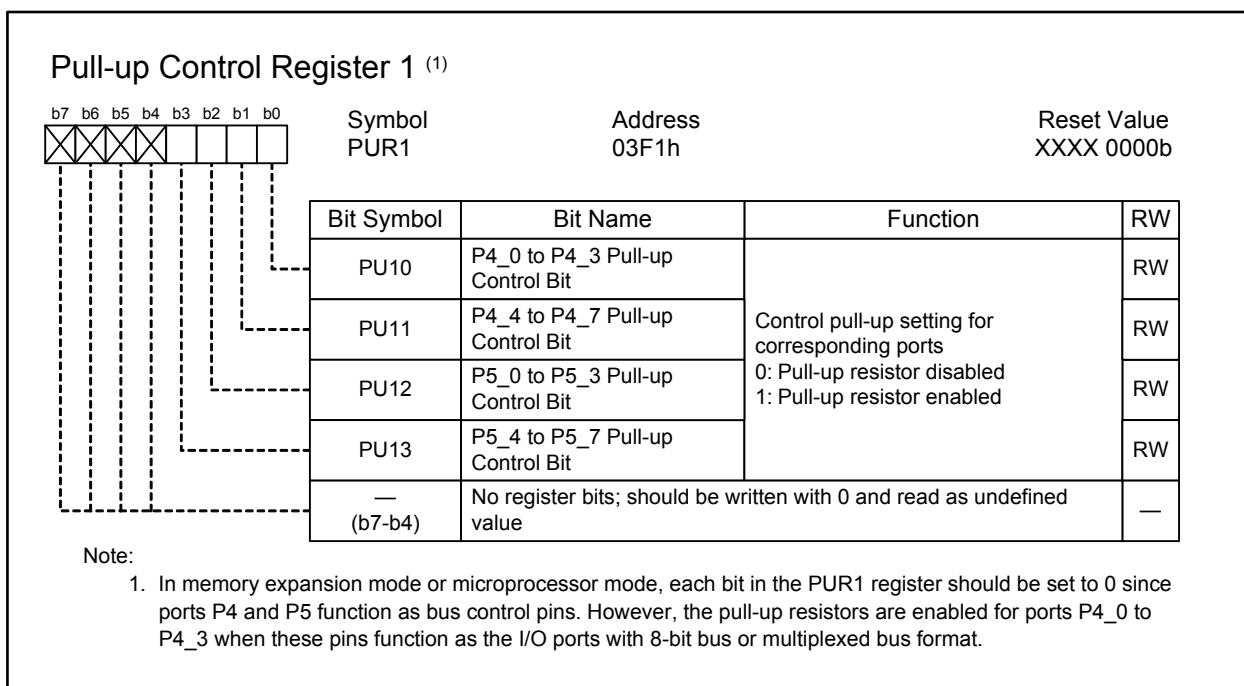


Figure 24.23 PUR1 Register

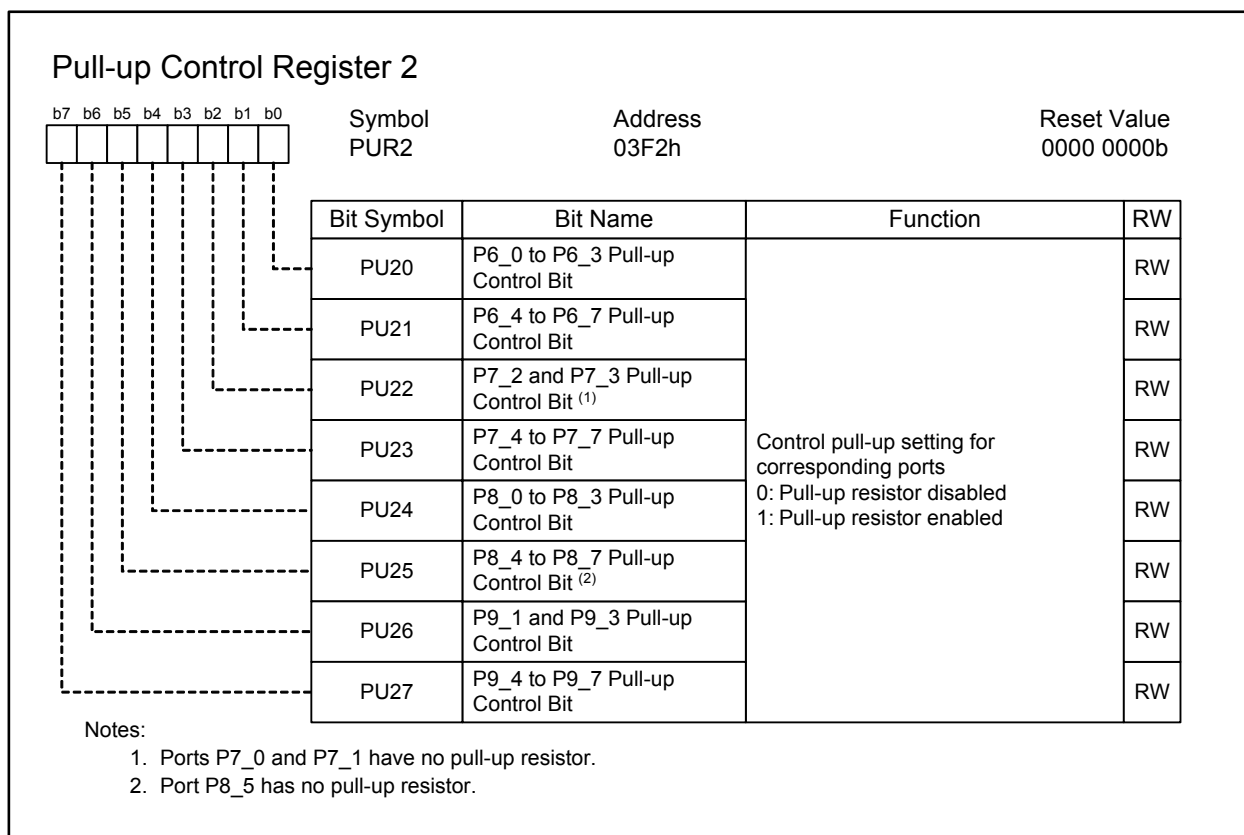
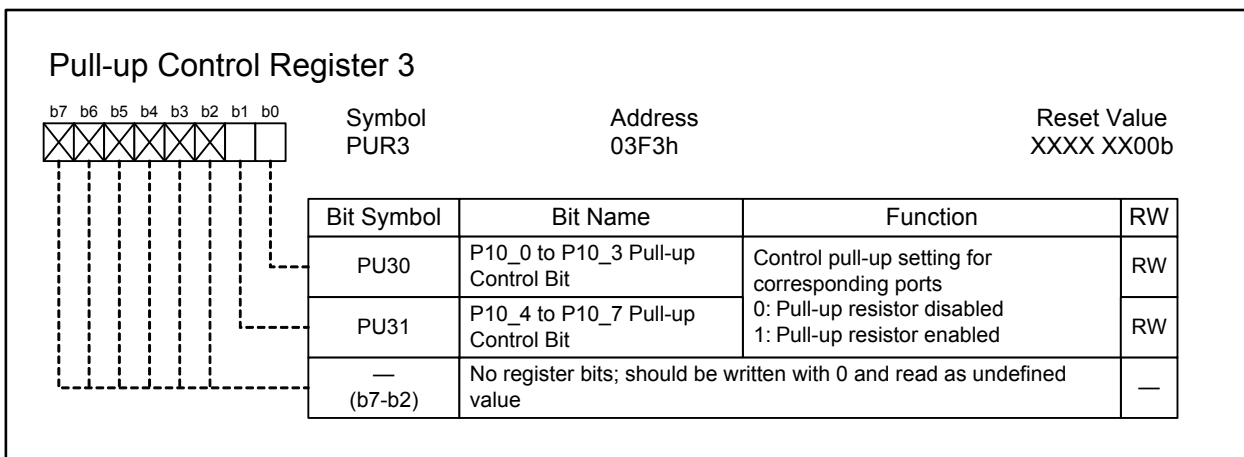


Figure 24.24 PUR2 Register



**Figure 24.25 PUR3 Register**



### 24.5 Port Control Register (PCR Register)

Figure 24.26 shows the PCR register.

This register selects an output mode for port P1 between push-pull output and pseudo-N-channel open drain output. When the PCR0 bit is set to 1, the P-channel transistor in the output buffer is turned off. Note that port P1 cannot be a perfect open drain output due to remaining parasitic diode. The absolute maximum rating of the input voltage is, therefore,  $-0.3\text{ V}$  to  $VCC2 + 0.3\text{ V}$  (refer to Figure 24.27).

In memory expansion mode or microprocessor mode, when port P1 is used for the data bus, the PCR0 bit should be set to 0. However, when port P1 is used as a programmable I/O port or an I/O pin for the peripheral functions, the output mode can be selected by setting the PCR0 bit even in these operating modes.

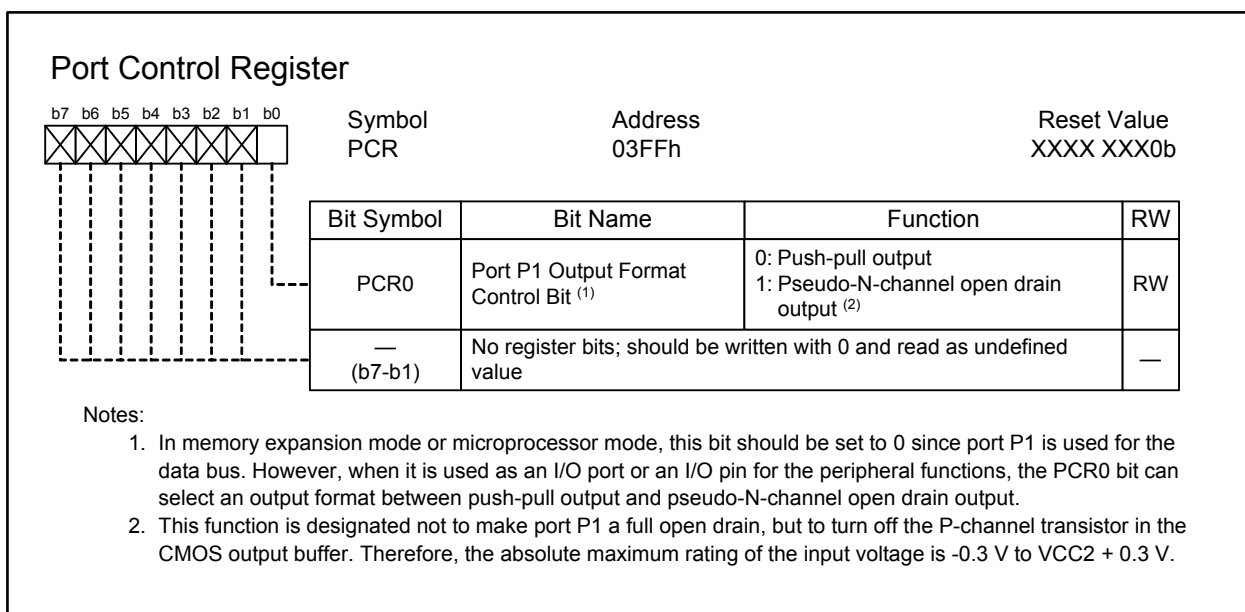


Figure 24.26 PCR Register

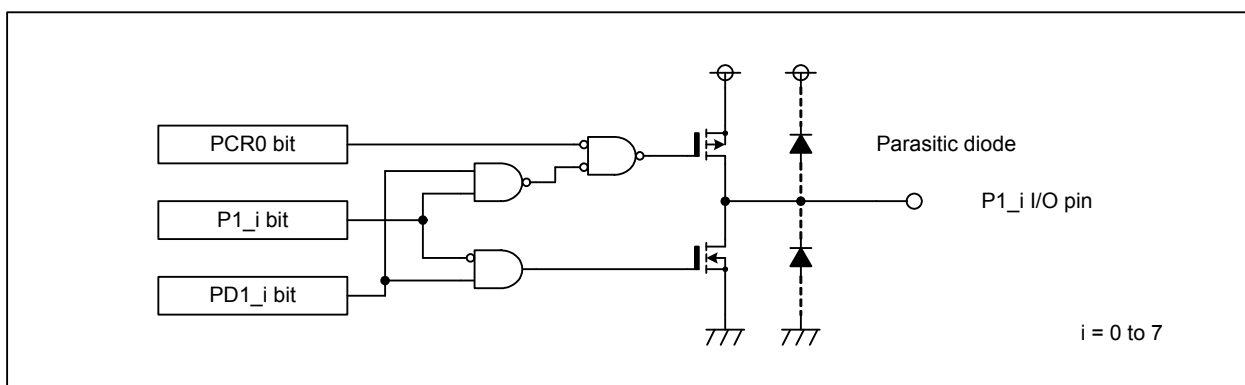


Figure 24.27 Port P1 Output Buffer Configuration

## 24.6 Configuring Unused Pins

Tables 24.2 and 24.3, and Figure 24.29 show examples of configuring unused pins on the board.

**Table 24.2 Unused Pin Configuration in Single-chip Mode (1)**

Pin Name	Setting
Ports P0 to P10 (excluding ports P8_5 and P9_1) (2, 3, 4, 5)	Configure as input ports so that each pin is connected to VSS via its own resistor; (6) or configure as output ports to leave the pins open
P9_1 (in the 100-pin package)	Connect the pin to VSS via a resistor (6)
XOUT (7)	Leave pin open
NMI (P8_5)	Connect the pin to VCC1 via a resistor (6)
AVCC	Connect the pin to VCC1
AVSS, VREF	Connect the pin to VSS
NSD	Connect the pin to VCC1 via a resistor of 1 to 4.7 kΩ

Notes:

1. Unused pins should be wired within 2 cm of the MCU.
2. When configuring as an output port to release the pin open, it remains as an input port until it is set as an output port after a reset is released. Therefore, while it remains as an input port, the power supply current may increase due to the undefined voltage level of the pin. In addition, the direction register value may change due to noise or program runaway caused by the noise. To avoid these situations, reconfigure the direction register regularly by software, which may achieve higher program reliability.
3. Ports P7\_0 and P7\_1 are N-channel open drain output. To use these ports as output ports, a low signal should be output.
4. Ports P0\_4 to P0\_7, P1\_0 to P1\_4, P3\_4 to P3\_7, P4, P5, P9\_4 to P9\_7 are available in the 100-pin package.
5. In the 64-pin package, set registers PD4 and PD5 to FFh. Also set the following bits to 1: Bits PD0\_4 to PD0\_7 in the PD0 register, bits PD1\_0 to PD1\_4 in the PD1 register, bits PD3\_4 to PD3\_7 in the PD3 register, and bits PD9\_4 to PD9\_7 in the PD9 register.
6. Select a resistance value that is appropriate for the system. A range from 10 to 100 kΩ is recommended.
7. This setting is applicable when an external clock is applied to the XIN pin.

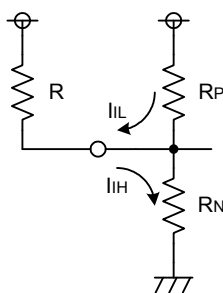
**Table 24.3 Unused Pin Configuration in Memory Expansion Mode or Microprocessor Mode (in the 100-pin package only) (1)**

Pin Name	Setting
Ports P1, P6 to P10 (excluding ports P8_5 and P9_1) (2, 3)	Configure as input ports so that each pin is connected to VSS via its own resistor; (4) or configure as output ports to leave the pins open
P9_1	Connect the pin to VSS via a resistor (4)
$\overline{BC0}$ to $\overline{BC1}$ , $\overline{WR0}$ to $\overline{WR1}$ , ALE, $\overline{HLDA}$ , XOUT (5), BCLK	Leave the pins open
$\overline{HOLD}$ , $\overline{RDY}$	Connect the pins to VCC2 via a resistor (4)
$\overline{NMI}$ (P8_5)	Connect the pin to VCC1 via a resistor (4)
AVCC	Connect the pin to VCC1
AVSS, VREF	Connect the pins to VSS
NSD	Connect the pin to VCC1 via a resistor of 1 to 4.7 k $\Omega$

## Notes:

- Unused pins should be wired within 2 cm of the MCU.
- When configuring as an output port to release the pin open, it remains as an input port until it is set as an output port after a reset is released. Therefore, while it remains as an input port, the power supply current may increase due to the undefined voltage level of the pin. In addition, the direction register value may change due to noise or program runaway caused by the noise. To avoid these situations, reconfigure the direction register regularly by software, which may achieve higher program reliability.
- Ports P7\_0 and P7\_1 are N-channel open drain output. To use these ports as output ports, a low signal should be output.
- Select a resistance value that is appropriate for the system. A range from 10 to 100 k $\Omega$  is recommended.
- This setting is applicable when an external clock is applied to the XIN pin.

## Pull-up/pull-down resistors



The figure shows the equivalent circuit of an input pin.

The equivalent input resistors ( $R_P$  and  $R_N$ ) are calculated using input power current ( $I_{IL}$  and  $I_{IH}$ ).

Example: When  $V_{CC} = 5.0\text{ V}$ ,  $I_{IH} = I_{IL} = 5\ \mu\text{A}$ ,

$$R_P = R_N = \frac{5.0}{5 \times 10^{-6}} = 1\ \text{M}\Omega$$

Since the voltage ( $V_{IH}$ ) defined as high is more than 0.8 VCC, the resistance value R should satisfy the following expression:

$$R/R_P : R_N = 0.2 : 0.8$$

That is,

$$R = \frac{2R_P R_N}{8R_P - 2R_N}$$

Specifically,

Example: When  $V_{CC} = 5.0\text{ V}$ ,  $I_{IH} = I_{IL} = 5\ \mu\text{A}$ ,

$$R = \frac{2 \times 10^6 \times 10^6}{8 \times 10^6 - 2 \times 10^6} = 333333$$

The maximum pull-up resistor R is approximately 330 k $\Omega$ .

The actual resistance value is the calculated value with some margins.

**Figure 24.28 Pull-up/Pull-down Resistors**

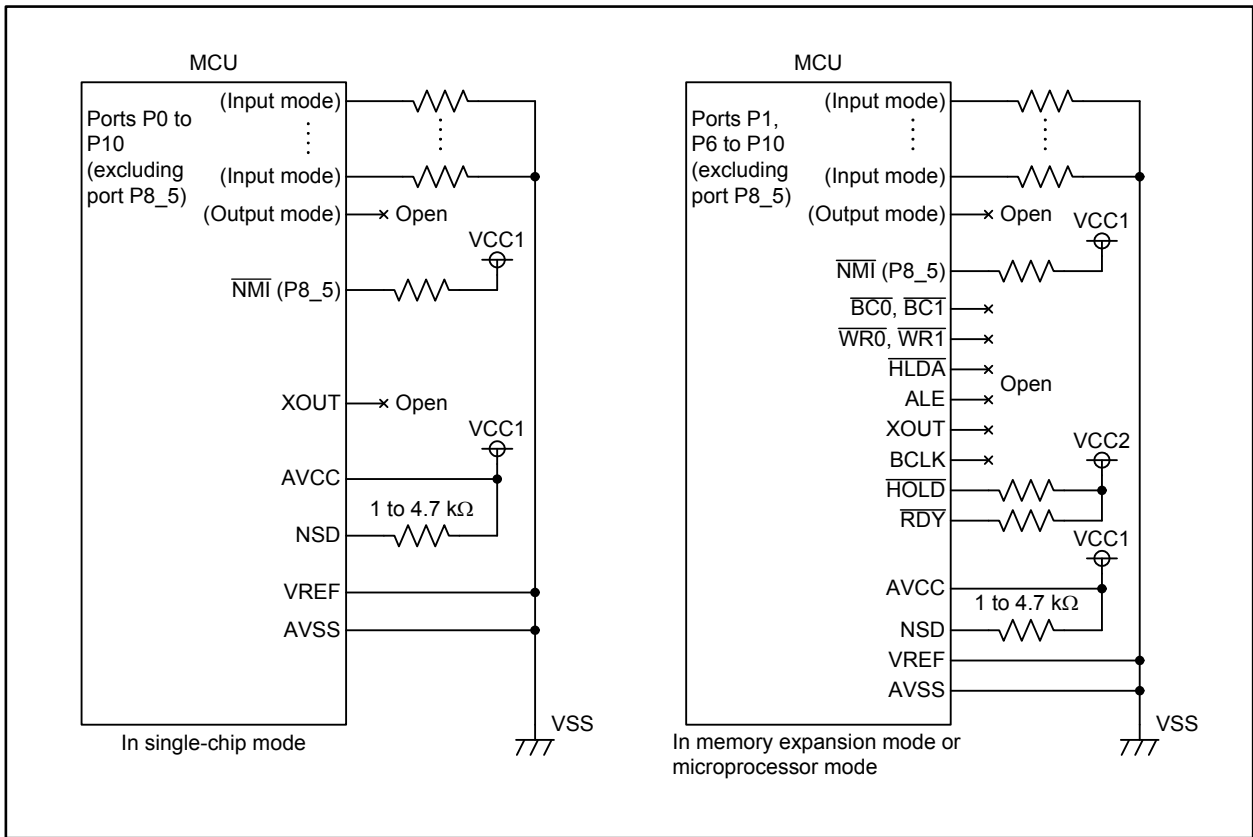


Figure 24.29 Unused Pin Configuration

## 25. Flash Memory

### 25.1 Overview

The flash memory can be programmed in the following three modes: CPU rewrite mode, standard serial I/O mode, and parallel I/O mode.

Table 25.1 lists specifications of the flash memory and Table 25.2 shows the overview of each rewrite mode.

**Table 25.1 Flash Memory Specifications**

Item	Specification
Rewrite modes	CPU rewrite mode, standard serial I/O mode, parallel I/O mode
Structure	Block architecture. Refer to Figure 25.1
Program operation	8-byte basis
Erase operation	1-block basis
Program and erase control method	Software commands
Protection types	Lock bit protect, ROM code protect, ID code protect
Software commands	9

**Table 25.2 Flash Memory Rewrite Mode Overview**

Rewrite Mode	CPU Rewrite Mode	Standard Serial I/O Mode	Parallel I/O Mode
Function	CPU executes a software command to rewrite the flash memory EW0 mode: Rewritable in areas other than the on-chip flash memory EW1 mode: Rewritable in areas other than specified blocks to be rewritten	A dedicated serial programmer rewrites the flash memory Standard serial I/O mode 1: Synchronous serial I/O selected Standard serial I/O mode 2: UART selected	A dedicated parallel programmer rewrites the flash memory
CPU operating mode	Single-chip mode, Memory expansion mode (EW0 mode)	Standard serial I/O mode	Parallel I/O mode
Programmer	—	Serial programmer	Parallel programmer
On-board programming	Supported	Supported	Not supported

Figure 25.1 shows the on-chip flash memory structure.

The on-chip flash memory contains program area to store user programs, and data area/data flash to store the result of user programs. The program area consists of blocks 0 to 9, and data area/data flash consists of blocks A and B.

Each block can be individually protected (locked) from programming or erasing by setting the lock bit.

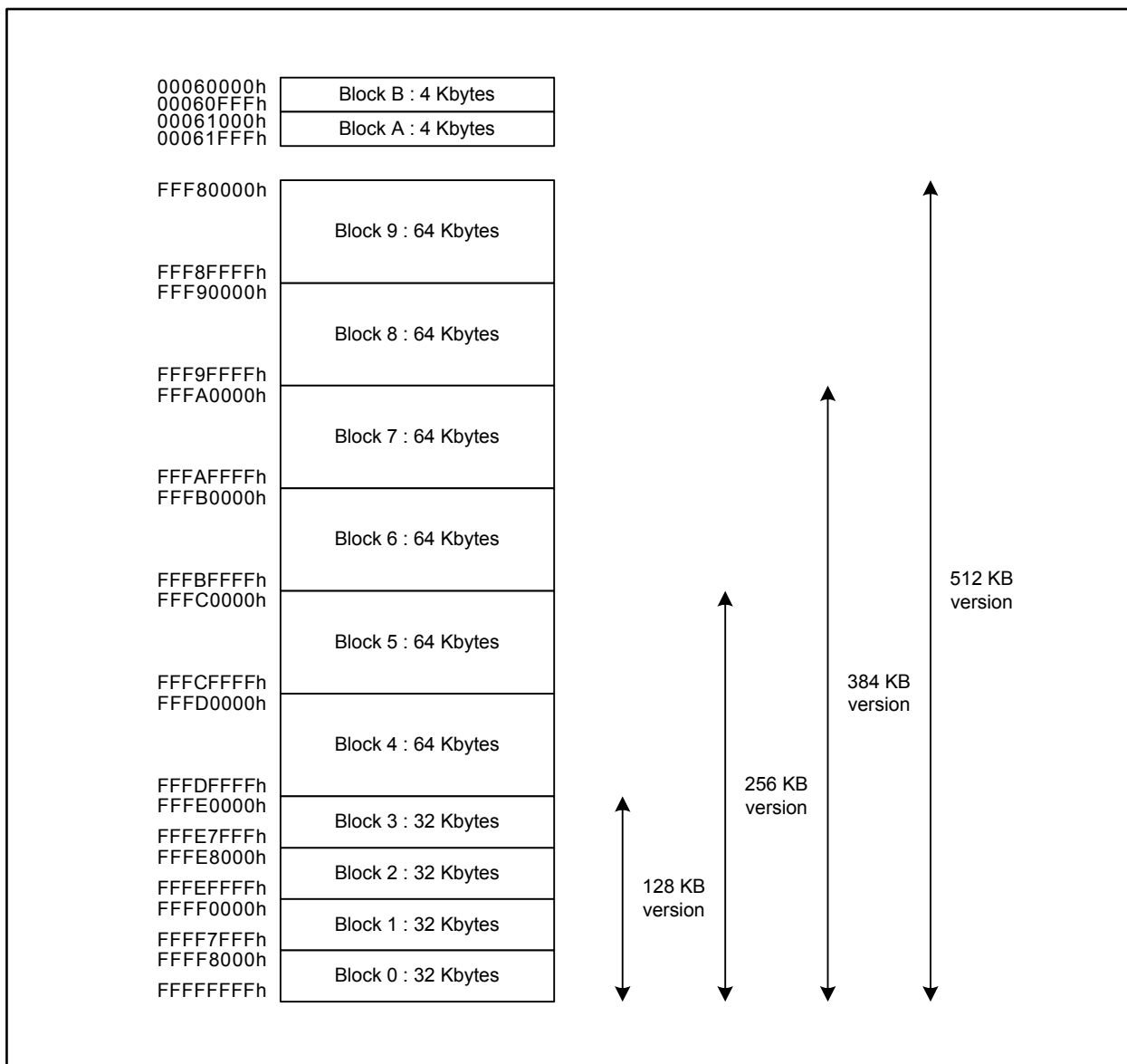


Figure 25.1 On-chip Flash Memory Block Diagram

## 25.2 Flash Memory Protection

There are three types of protection as shown in Table 25.3. Lock bit protection is intended to prevent accidental write or erase by program runaway. ROM code protection and ID code protection are intended to prevent read or write by a third party.

**Table 25.3 Protection Types and Characteristics**

Protection Type	Lock Bit Protection	ROM Code Protection	ID Code Protection
Protected operations	Erase, write	Read, write	Read, erase, write
Protection available in	CPU rewrite mode Standard serial I/O mode Parallel I/O mode	Parallel I/O mode	Standard serial I/O mode
Protection available for	Individual blocks	Entire flash memory	Entire flash memory
Protection settings	Setting 0 to the lock bit of block to be protected	Setting the protect bit of any block to 0	Writing the program which has set an ID code to specified address
Protection disabled by	Setting the LBD bit in the FMR register to 1 (lock bit protection disabled), or by erasing the blocks whose lock bits are set to 0 to permanently disable the protection	Erasing all blocks whose protect bits are set to 0	Sending a proper ID code from the serial programmer

### 25.2.1 Lock Bit Protection

This protection can be used in all three rewrite modes. When the lock bit protection is enabled, all blocks whose lock bits are set to 0 (locked) are protected against programming and erasing.

To set the lock bit to 0, the lock bit program command must be issued.

To temporarily disable the protection of all protected blocks, disable the lock bit protection itself by setting the LBD bit in the FMR1 register to 1 (lock bit protection disabled). The protection of a protected block is disabled permanently and its lock bit becomes 1 (unlocked) if the block is erased.

### 25.2.2 ROM Code Protection

This protection can only be used in parallel I/O mode. When the ROM code protection is enabled, the entire flash memory is protected against reading and writing.

To disable the protection, erase all the blocks whose protect bits are set to 0 (protected).

Each block has two protect bits. Setting any protect bit to 0 by a software command enables the protection for the entire flash memory. Table 25.4 lists protect bit addresses.

**Table 25.4 Protect Bit Addresses**

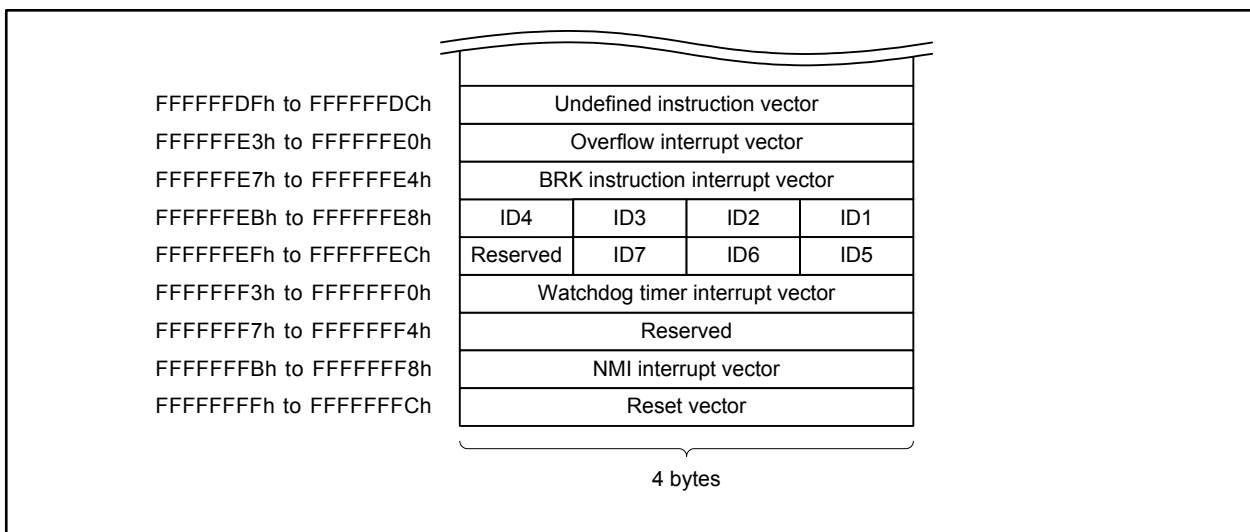
Block	Protect Bit 0	Protect Bit 1
Block B	00060100h	00060300h
Block A	00061100h	00061300h
Block 9	FFF80100h	FFF80300h
Block 8	FFF90100h	FFF90300h
Block 7	FFFA0100h	FFFA0300h
Block 6	FFFB0100h	FFFB0300h
Block 5	FFFC0100h	FFFC0300h
Block 4	FFFD0100h	FFFD0300h
Block 3	FFFE0100h	FFFE0300h
Block 2	FFFE8100h	FFFE8300h
Block 1	FFFF0100h	FFFF0300h
Block 0	FFFF8100h	FFFF8300h

### 25.2.3 ID Code Protection

This protection can only be used in standard serial I/O mode. A command from the serial programmer is to be accepted when the 7-byte ID code sent from the serial programmer matches the ID code programmed in the flash memory. However, when the reset vector is FFFFFFFFh, the ID code check is skipped because the flash memory is considered to be blank. When the reset vector is FFFFFFFFh and the ROM code protection is enabled, only the block erase command is accepted.

The ID codes sent from the serial programmer are consecutively numbered as ID1, ID2, ..., and ID7. ID codes programmed in the flash memory, also numbered as ID1, ID2, ..., and ID7, are assigned to addresses FFFFFFFE8h, FFFFFFFE9h, ..., and FFFFFFFEeh as shown in Figure 25.2. The ID code protection is enabled when a program which has an ID code set in the corresponding address is written to the flash memory.

In the 64-pin package, the following two ASCII code combinations are specified as reserved ID codes: "ALeRASE" and "Protect". Refer to Table 25.5, 25.2.4 "Forcible Erase Function", and 25.2.5 "Standard Serial I/O Mode Disable Function" for details.

**Figure 25.2 Addresses for ID Code Stored**



**Table 25.5 Reserved ID Codes**

ID Code		ID1	ID2	ID3	ID4	ID5	ID6	ID7
ALeRASE	Glyph	A	L	e	R	A	S	E
	ASCII code	41h	4Ch	65h	52h	41h	53h	45h
Protect	Glyph	P	r	o	t	e	c	t
	ASCII code	50h	72h	6Fh	74h	65h	63h	74h

### 25.2.4 Forcible Erase Function

The forcible erase function is available in standard serial I/O mode in the 64-pin package and is optional in the 100-pin package. Contact a Renesas Electronics sales office for details. With this function, all blocks of the flash memory are forcibly erased when ID codes sent from the serial programmer matches the ASCII code corresponding to the following sequential ASCII-glyphs: “A”, “L”, “e”, “R”, “A”, “S”, and “E”. However, the function is ignored when the ROM code protection is activated and ID codes other than “ALeRASE” are programmed in the flash memory.

**Table 25.6 Operational Conditions for Forcible Erase Function**

ID Codes Sent From the Serial Programmer	ID Codes Programmed in the Flash Memory	ROM Code Protection	Function
“ALeRASE”	“ALeRASE”	—	Erase all blocks of the flash memory
	Any codes other than “ALeRASE” or “Protect”	Inactivated	
		Activated	Check ID codes (resulted in unmatched codes)
Any codes other than “ALeRASE”	“ALeRASE”	—	Check ID codes (resulted in unmatched codes)
	Any codes other than “ALeRASE” or “Protect”	—	Check ID codes

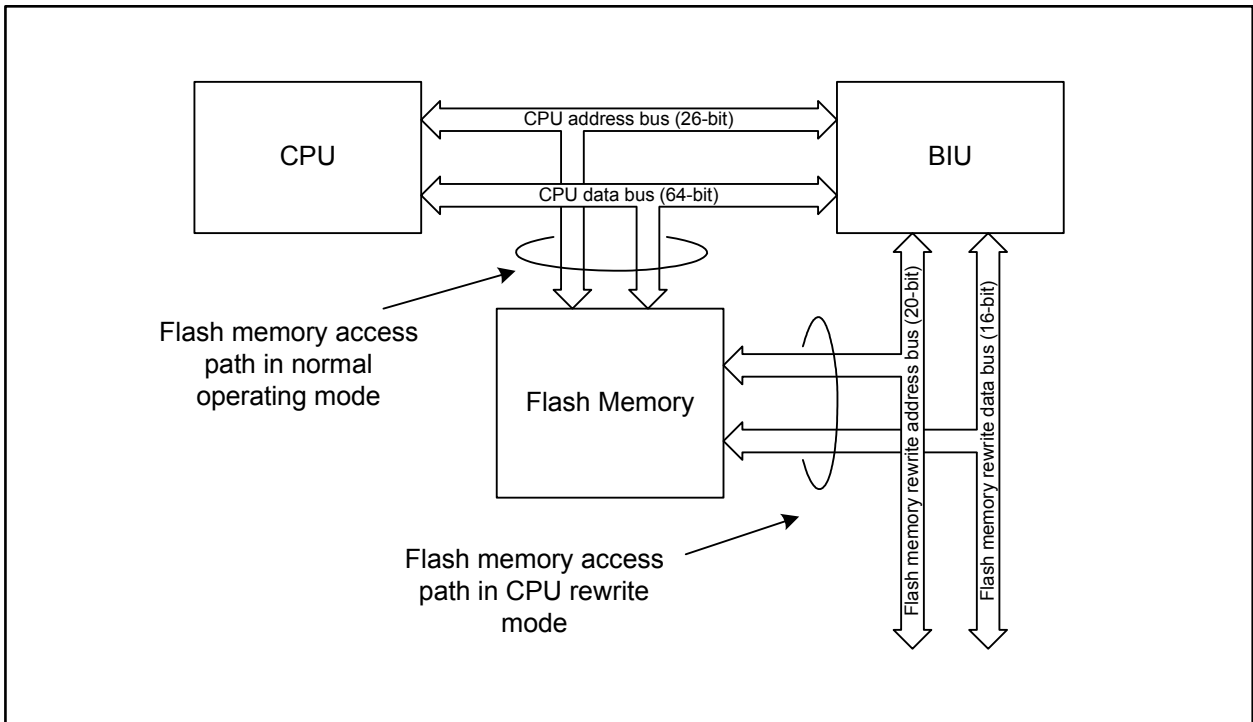
### 25.2.5 Standard Serial I/O Mode Disable Function

The standard serial I/O mode disable function is available in the 64-pin package and is optional in the 100-pin package. Contact a Renesas Electronics sales office for details. With the standard serial I/O mode disable function, the flash memory in standard serial I/O mode is inaccessible from the CPU when ID code programmed in the flash memory are ASCII codes corresponding to the following sequential ASCII-glyphs: “P”, “r”, “o”, “t”, “e”, “c”, and “t”.

When the ROM code protection is activated and ID codes corresponding to “Protect” are programmed, the serial programmer cannot deactivate the ROM code protection. In this case, the flash memory is not accessible from the outside of MCU, except that the parallel programmer can delete the flash memory.

### 25.3 CPU Rewrite Mode

In CPU rewrite mode, the CPU executes software commands to rewrite the flash memory. The CPU accesses the flash memory not via the CPU buses, but via the dedicated flash memory rewrite buses (refer to Figure 25.3).



**Figure 25.3 Flash Memory Access Path in CPU Rewrite Mode**

Bus setting for flash memory rewrite should be performed by registers FEBC0 and FEBC3. Refer to 25.3.2 “Flash Memory Rewrite Bus Timing” and 26. “Electrical Characteristics” for the appropriate bus setting. Note that registers FEBC0 and FEBC3 share respective addresses with registers EBC0 and EBC3. That is, a rewrite of these registers affects the external bus setting. Set registers EBC0 and EBC3 again after rewriting the registers FEBC0 and FEBC3.

The CPU rewrite mode contains modes EW0 and EW1 as shown in Table 25.7.

**Table 25.7 EW0 and EW1 Modes**

Item	EW0 Mode	EW1 Mode
CPU operating modes	Single-chip mode Memory expansion mode <sup>(1)</sup>	Single-chip mode
Rewrite program executable spaces	Spaces other than the on-chip flash memory	Internal spaces other than specified blocks to be rewritten, internal RAM
Restrictions on software commands	None	<ul style="list-style-type: none"> <li>• Do not execute either the program command or the block erase command for blocks where the rewrite control programs are written to</li> <li>• Do not execute the enter read status register mode command</li> <li>• Execute the enter read lock bit status mode command in RAM</li> <li>• Execute the enter read protect bit status mode command in RAM</li> </ul>
Mode after program/erase operation	Read status register mode	Read array mode
CPU state during program/erase operation	Operating	In a hold state (I/O ports maintain the state before the command was executed)
Flash memory state detection by	<ul style="list-style-type: none"> <li>• Reading the FMSR0 register by a program</li> <li>• Executing the enter read status register mode command to read data</li> </ul>	<ul style="list-style-type: none"> <li>• Reading the FMSR0 register by a program</li> </ul>
Other restrictions	None	<ul style="list-style-type: none"> <li>• Disable interrupts (except NMI) and DMA transfer during program/erase operation</li> </ul>

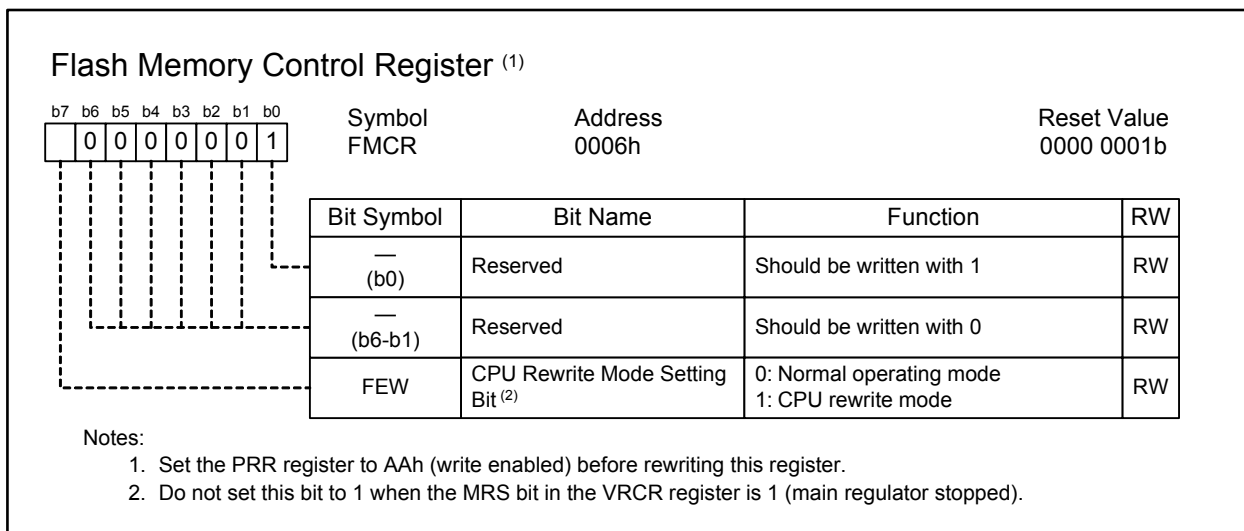
Note:

1. The  $\overline{CS0}$  space and  $\overline{CS3}$  space have limited availability in memory expansion mode. Refer to 25.3.1 "CPU Operating Mode and Flash Memory Rewrite" for details.

To select CPU rewrite mode, the FEW bit in the FMCR register should be set to 1. Then, EW0 mode/EW1 mode can be selected by setting the EWM bit in the FMR0 register.

Registers FMCR and FMR0 are protected by registers PRR and FPR0, respectively.

Figures 25.4 to 25.11 show associated registers.

**Figure 25.4 FMCR Register**

**Flash Memory Rewrite Bus Control Register i (i = 0, 3)<sup>(1)</sup>**

Symbol  
FEBC0, FEBC3

Address  
001Dh-001Ch, 0011h-0010h

Reset Value  
0000h

Bit Symbol	Bit Name	Function	RW
FWR0	RD Pulse Width Setting Bit	b3 b2 b1 b0 0 0 0 0 : wr = 1 0 0 0 1 : wr = 2 0 1 0 1 : wr = 3 0 1 1 0 : wr = 4 1 0 1 0 : wr = 5 1 0 1 1 : wr = 6 1 1 1 1 : wr = 7 Only use the combinations listed above	RW
FWR1			RW
FWR2			RW
FWR3			RW
FWR4	RD Pulse Width Extension Select Bit	0: No pulse width extension 1: Pulse width extension selected	RW
— (b5)	Reserved	Should be written with 0	RW
MPY0	Multiplied Cycle Setting Bit	b7 b6 0 0 : Do not use this combination 0 1 : Do not use this combination 1 0 : mpy = 3 1 1 : mpy = 4	RW
MPY1			RW
FSUW0	Address Setup Before WR Setting Bit	b9 b8 0 0 : suw = 0 0 1 : suw = 1 1 0 : suw = 2 1 1 : suw = 3	RW
FSUW1			
FWW0	WR Pulse Width Setting Bit	b11 b10 0 0 : ww = 1 0 1 : ww = 2 1 0 : ww = 3 1 1 : ww = 4	RW
FWW1			
— (b12)	Reserved	Should be written with 1	RW
— (b13)	Reserved	Should be written with 0	RW
— (b14)	Reserved	Should be written with 1	RW
— (b15)	Reserved	Should be written with 0	RW

Note:

- Set the PRR register to AAh (write enabled) before rewriting this register.

**Figure 25.5 Registers FEBC0 and FEBC3**

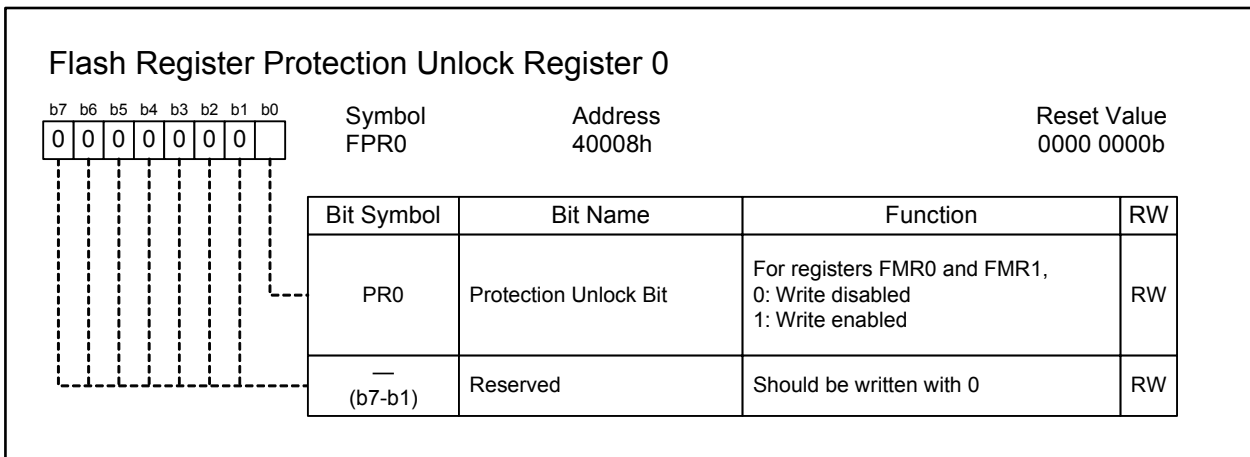


Figure 25.6 FPR0 Register

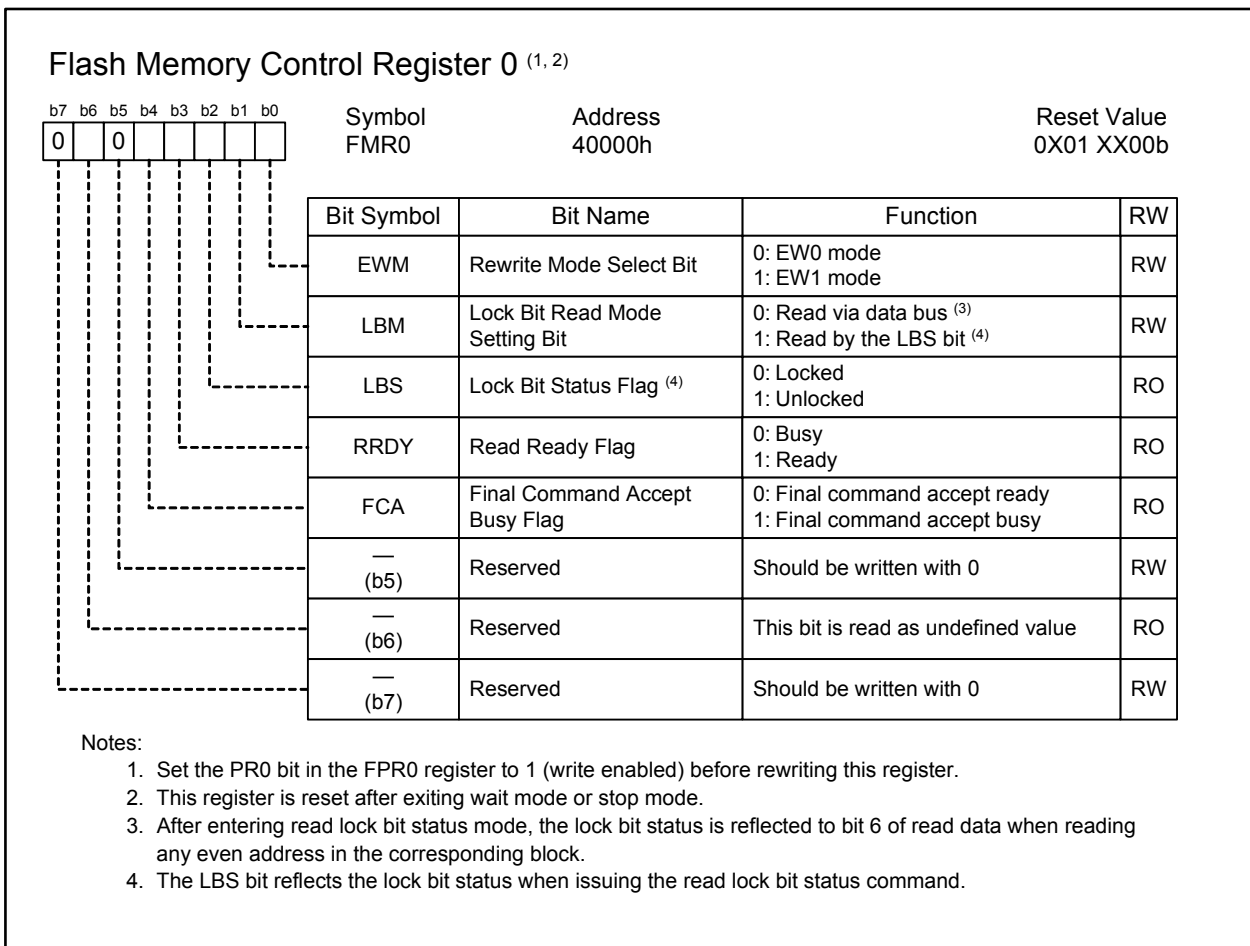


Figure 25.7 FMR0 Register

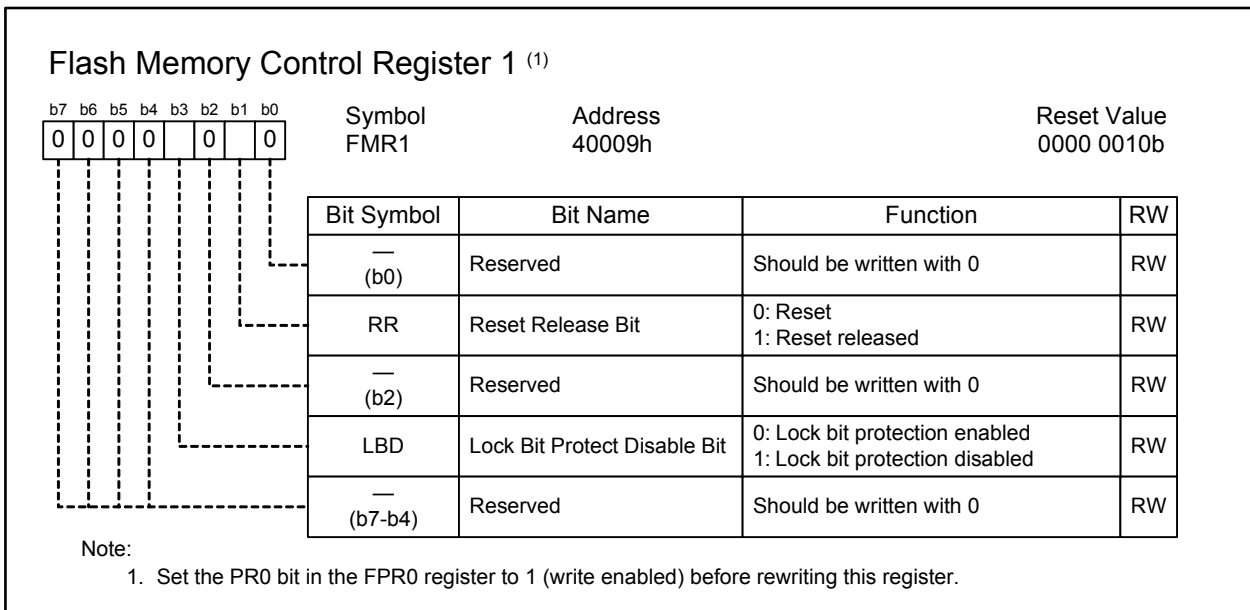


Figure 25.8 FMR1 Register

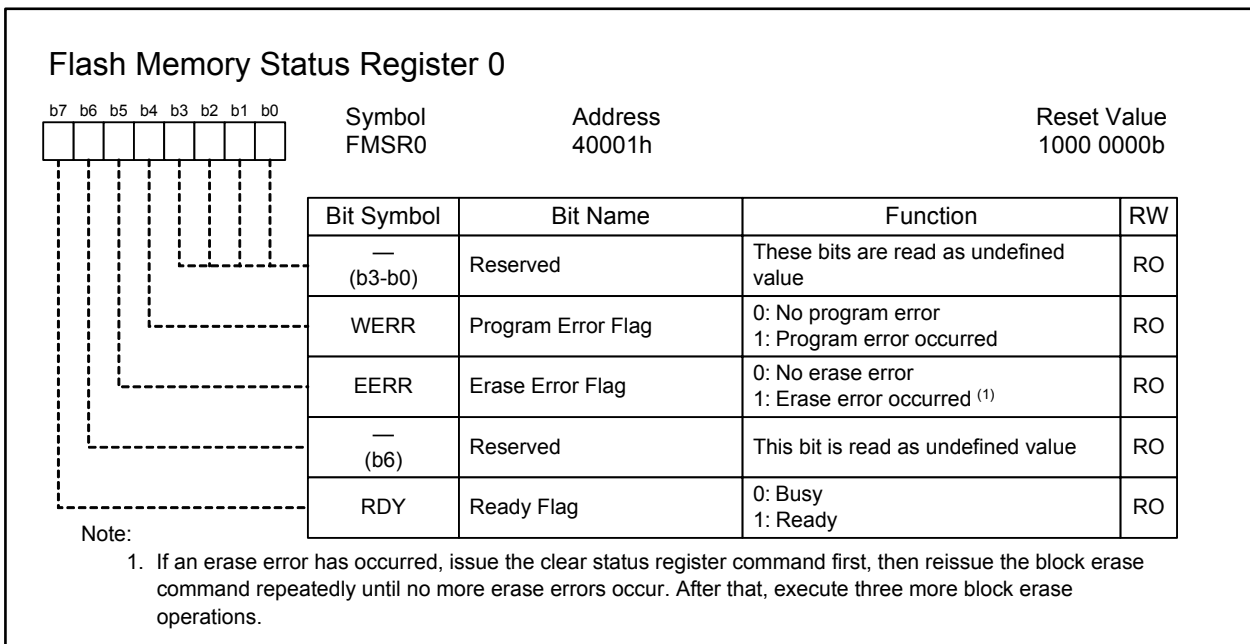


Figure 25.9 FMSR0 Register

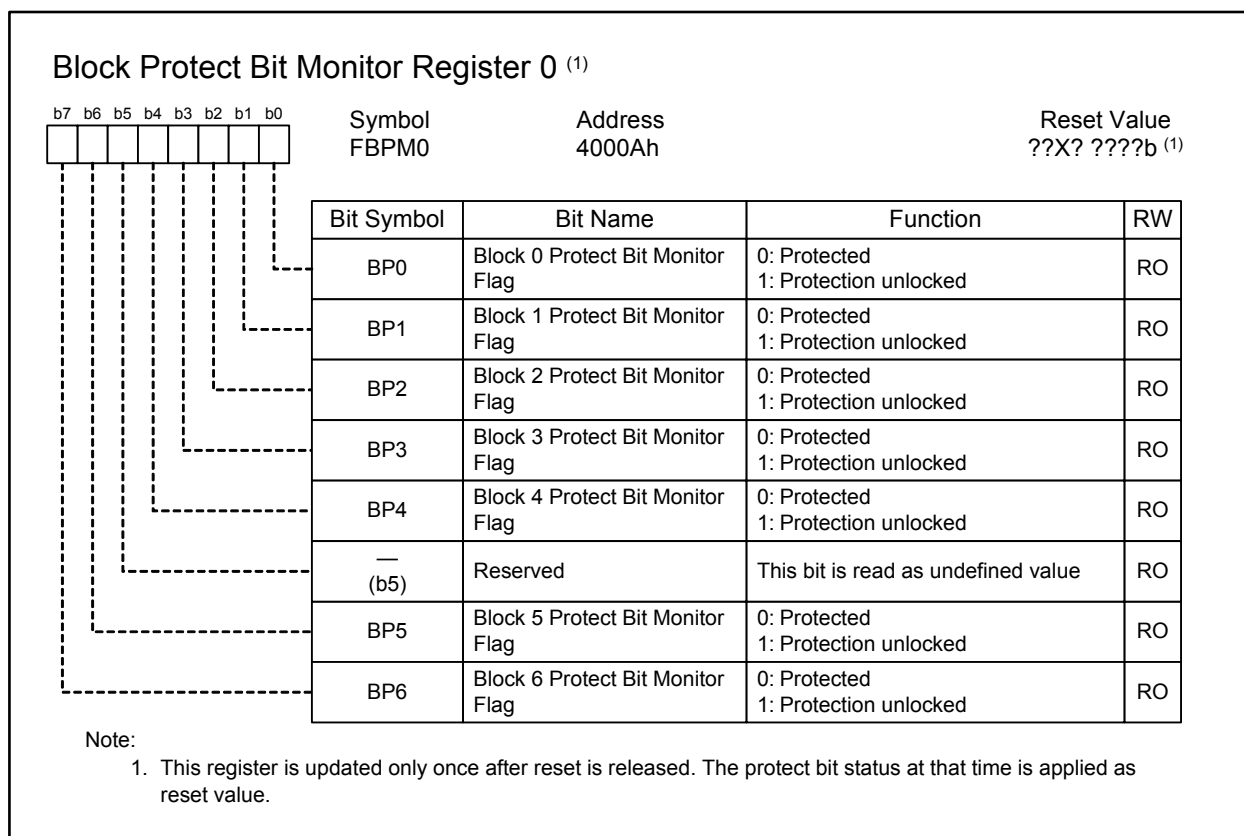


Figure 25.10 FBPM0 Register

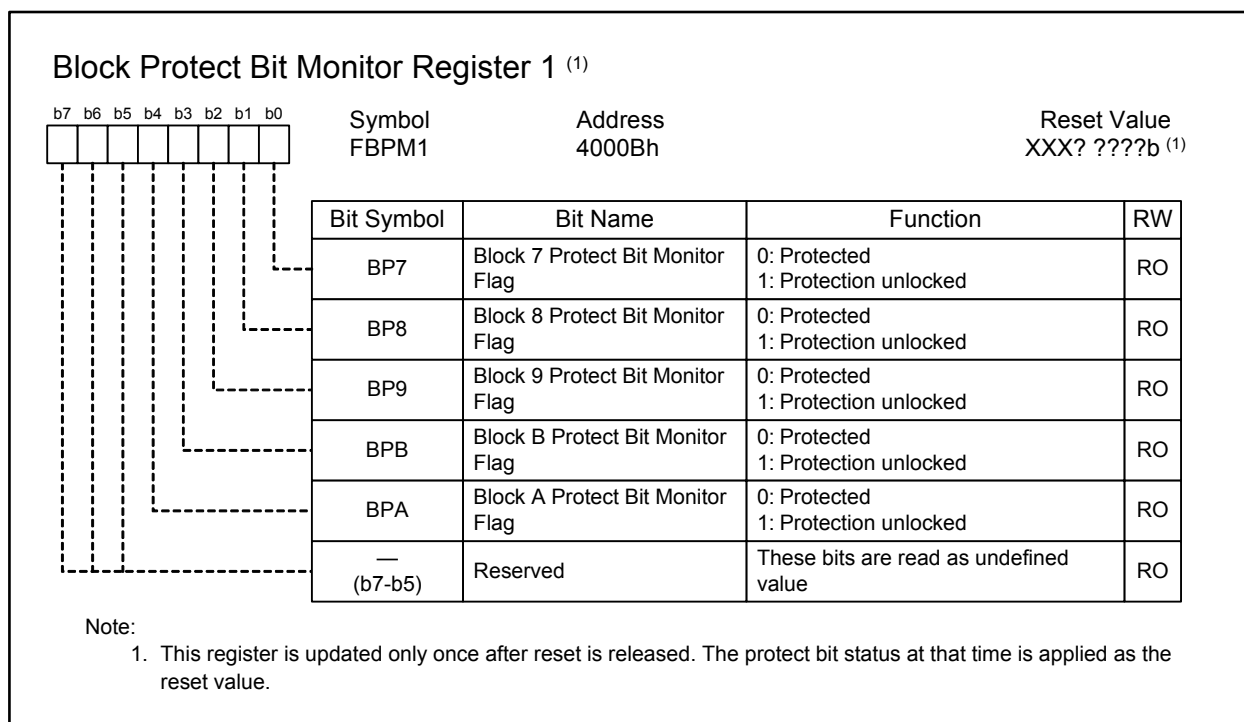


Figure 25.11 FBPM1 Register



### 25.3.1 CPU Operating Mode and Flash Memory Rewrite

Registers used to set the bus timing of rewriting the flash memory vary with the CPU operating modes. Do not change the 00h reset value of registers CB01, CB12, and CB23 when using single-chip mode. The bus setting for both the program area and data area can be performed using the FEBC0 register. In cases other than the above, when the CPU operation is performed in memory expansion mode more than once, set registers CB01, CB12, and CB23 according to each setting range as shown in Table 25.8. The bus setting for program area and data area can be performed by the FEBC0 register and FEBC3 register, respectively.

Note that registers FEBC0 and FEBC3 in memory expansion mode share respective addresses with registers EBC0 and EBC3. That is, when the FEBC<sub>i</sub> register (i = 0, 3) is set for the flash memory rewrite, the setting value for the EBC<sub>i</sub> register is accordingly changed. This may cause external devices allocated to the  $\overline{CS0}$  space and/or  $\overline{CS3}$  space in CPU rewrite mode to become inaccessible.

Table 25.8 lists the details of bus setting for the flash memory rewrite in each CPU operating mode.

**Table 25.8 CPU Operating Mode and Flash Memory Rewrite**

Item	CPU Operating Mode	
	Single-chip mode	Memory expansion mode
CB01 register	Hold the reset value 00h	Setting range: 02h to F8h Set a value equal to or greater than that of the CB12 register
CB12 register	Hold the reset value 00h	Setting range: 02h to F8h Set a value equal to or greater than that of the CB23 register and equal to or less than that of the CB01 register
CB23 register	Hold the reset value 00h	Setting range: 02h to F8h Set a value equal to or less than that of the CB12 register
Bus setting for program area	FEBC0 register	FEBC0 register
Bus setting for data area	FEBC0 register	FEBC3 register
State of $\overline{CS0}$ space and $\overline{CS3}$ space after the FEBC <sub>i</sub> register is set	N/A	<ul style="list-style-type: none"> <li>• Separate bus format</li> <li>• 16-bit bus width</li> <li>• <math>\overline{RDY}</math> ignored</li> </ul>
Restrictions for the use of $\overline{CS0}$ space and $\overline{CS3}$ space	None	<ul style="list-style-type: none"> <li>• <math>\overline{HOLD}</math> is ignored</li> <li>• In CPU rewrite mode, external devices become inaccessible to data with the bus format set for <math>\overline{CS0}</math> space and/or <math>\overline{CS3}</math> space as multiplexed bus</li> <li>• The change in bus timing may cause external devices in the <math>\overline{CS0}</math> space and/or <math>\overline{CS3}</math> space to become inaccessible</li> </ul>

### 25.3.2 Flash Memory Rewrite Bus Timing

As mentioned in 25.3.1, the bus setting for the flash memory rewrite is performed by setting the FEBC0 and/or FEBC3 registers. This section specifically describes the setting of registers FEBC0 and FEBC3. The reference clock is the base clock set with bits BCD1 and BCD0 in the CCR register. Time duration including  $t_{su}$ ,  $t_w$ ,  $t_c$ , and  $t_h$  are specified by the number of base clock cycles.

Tables 25.9 to 25.11 show the correlation of the read cycle and setting of bits MPY1, MPY0, and FWR4 to FWR0, according to peripheral bus clock divide ratios. Tables 25.12 to 25.14 show the correlation of the write cycle and setting of bits MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0. Associated read/write timings are illustrated in Figures 25.12 and 25.13, respectively.

Read/write cycle timing is selected from the tables below to meet the timing requirements in the CPU rewrite mode described in the electrical characteristics.

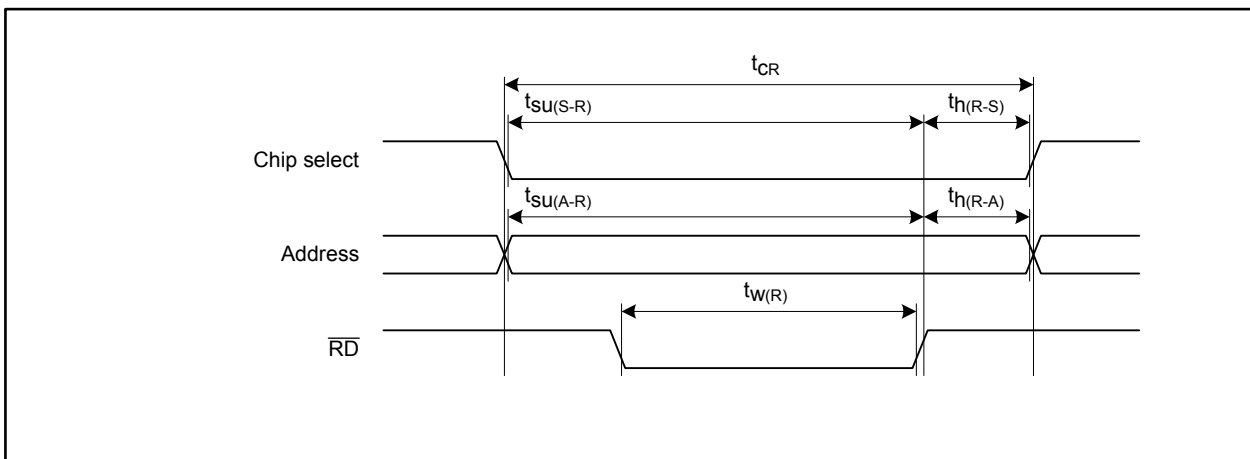


Figure 25.12 Read Timing

Table 25.9 Read Cycle and Bit Settings: MPY1, MPY0, and FWR4 to FWR0, When Peripheral Bus Clock is Divided by 2 (unit: cycles)

FWR3 to FWR0 Bit Settings		FWR4 Bit Settings	MPY1 and MPY0 Bit Settings							
			10b				11b			
			$mpy = 3$				$mpy = 4$			
			$t_{su}(S-R),$ $t_{su}(A-R)$	$t_w(R)$	$t_{CR}$	$t_h(R-S),$ $t_h(R-A)$	$t_{su}(S-R),$ $t_{su}(A-R)$	$t_w(R)$	$t_{CR}$	$t_h(R-S),$ $t_h(R-A)$
0000b	$wr = 1$	0	4	3	4	0	6	5	6	0
		1	6	5	6	0	6	5	6	0
0001b	$wr = 2$	0	8	7	8	0	10	9	10	0
		1	8	7	8	0	10	9	10	0
0101b	$wr = 3$	0	10	9	10	0	14	13	14	0
		1	12	11	12	0	14	13	14	0
0110b	$wr = 4$	0	14	13	14	0	18	17	18	0
		1	14	13	14	0	18	17	18	0
1010b	$wr = 5$	0	16	15	16	0	22	21	22	0
		1	18	17	18	0	22	21	22	0
1011b	$wr = 6$	0	20	19	20	0	26	25	26	0
		1	20	19	20	0	26	25	26	0
1111b	$wr = 7$	0	22	21	22	0	30	29	30	0
		1	24	23	24	0	30	29	30	0

**Table 25.10 Read Cycle and Bit Settings: MPY1, MPY0, and FWR4 to FWR0, When Peripheral Bus Clock is Divided by 3 (unit: cycles)**

FWR3 to FWR0 Bit Settings		FWR4 Bit Settings	MPY1 and MPY0 Bit Settings							
			10b				11b			
			<i>mpy = 3</i>				<i>mpy = 4</i>			
			tsu(S-R), tsu(A-R)	tw(R)	tCR	th(R-S), th(R-A)	tsu(S-R), tsu(A-R)	tw(R)	tCR	th(R-S), th(R-A)
0000b	<i>wr = 1</i>	0	6	4.5	6	0	6	4.5	6	0
		1	6	4.5	6	0	6	4.5	6	0
0001b	<i>wr = 2</i>	0	9	7.5	9	0	9	7.5	9	0
		1	9	7.5	9	0	12	10.5	12	0
0101b	<i>wr = 3</i>	0	12	10.5	12	0	15	13.5	15	0
		1	12	10.5	12	0	15	13.5	15	0
0110b	<i>wr = 4</i>	0	15	13.5	15	0	18	16.5	18	0
		1	15	13.5	15	0	18	16.5	18	0
1010b	<i>wr = 5</i>	0	18	16.5	18	0	21	19.5	21	0
		1	18	16.5	18	0	24	22.5	24	0
1011b	<i>wr = 6</i>	0	21	19.5	21	0	27	25.5	27	0
		1	21	19.5	21	0	27	25.5	27	0
1111b	<i>wr = 7</i>	0	24	22.5	24	0	30	28.5	30	0
		1	24	22.5	24	0	30	28.5	30	0

**Table 25.11 Read Cycle and Bit Settings: MPY1, MPY0, and FWR4 to FWR0, When Peripheral Bus Clock is Divided by 4 (unit: cycles)**

FWR3 to FWR0 Bit Settings		FWR4 Bit Settings	MPY1 and MPY0 Bit Settings							
			10b				11b			
			<i>mpy = 3</i>				<i>mpy = 4</i>			
			tsu(S-R), tsu(A-R)	tw(R)	tCR	th(R-S), th(R-A)	tsu(S-R), tsu(A-R)	tw(R)	tCR	th(R-S), th(R-A)
0000b	<i>wr = 1</i>	0	4	2	4	0	8	6	8	0
		1	8	6	8	0	8	6	8	0
0001b	<i>wr = 2</i>	0	8	6	8	0	12	10	12	0
		1	8	6	8	0	12	10	12	0
0101b	<i>wr = 3</i>	0	12	10	12	0	16	14	16	0
		1	12	10	12	0	16	14	16	0
0110b	<i>wr = 4</i>	0	16	14	16	0	20	18	20	0
		1	16	14	16	0	20	18	20	0
1010b	<i>wr = 5</i>	0	16	14	16	0	24	22	24	0
		1	20	18	20	0	24	22	24	0
1011b	<i>wr = 6</i>	0	20	18	20	0	28	26	28	0
		1	20	18	20	0	28	26	28	0
1111b	<i>wr = 7</i>	0	24	22	24	0	32	30	32	0
		1	24	22	24	0	32	30	32	0

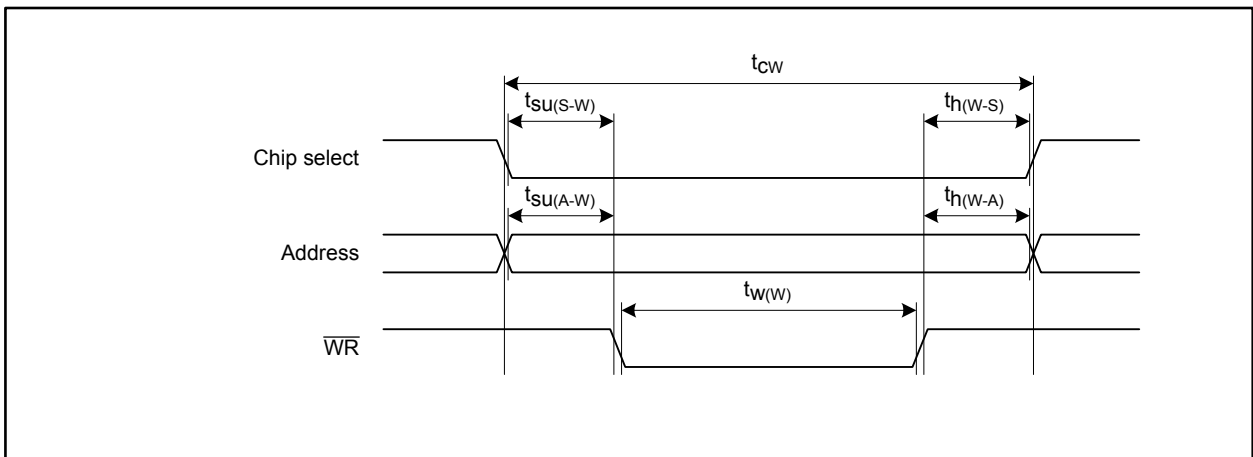


Figure 25.13 Write Timing

Table 25.12 Write Cycle and Bit Settings: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0, When Peripheral Bus Clock is Divided by 2 (unit: cycles)

FSUW1 and FSUW0 Bit Settings	FWW1 and FWW0 Bit Settings	MPY1 and MPY0 Bit Settings									
		10b					11b				
		<i>mpy</i> = 3					<i>mpy</i> = 4				
		tsu(S-W), tsu(A-W)	tw(W)	tcw	th(W-S), th(W-A)	tsu(S-W), tsu(A-W)	tw(W)	tcw	th(W-S), th(W-A)		
00b	<i>suw</i> = 0	00b	<i>ww</i> = 1	1	3	6	2	1	4	6	1
		01b	<i>ww</i> = 2	1	6	8	1	1	8	10	1
		10b	<i>ww</i> = 3	1	9	12	2	1	12	14	1
		11b	<i>ww</i> = 4	1	12	14	1	1	16	18	1
01b	<i>suw</i> = 1	00b	<i>ww</i> = 1	4	3	8	1	5	4	10	1
		01b	<i>ww</i> = 2	4	6	12	2	5	8	14	1
		10b	<i>ww</i> = 3	4	9	14	1	5	12	18	1
		11b	<i>ww</i> = 4	4	12	18	2	5	16	22	1
10b	<i>suw</i> = 2	00b	<i>ww</i> = 1	7	3	12	2	9	4	14	1
		01b	<i>ww</i> = 2	7	6	14	1	9	8	18	1
		10b	<i>ww</i> = 3	7	9	18	2	9	12	22	1
		11b	<i>ww</i> = 4	7	12	20	1	9	16	26	1
11b	<i>suw</i> = 3	00b	<i>ww</i> = 1	10	3	14	1	13	4	18	1
		01b	<i>ww</i> = 2	10	6	18	2	13	8	22	1
		10b	<i>ww</i> = 3	10	9	20	1	13	12	26	1
		11b	<i>ww</i> = 4	10	12	24	2	13	16	30	1

**Table 25.13 Write Cycle and Bit Settings: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0, When Peripheral Bus Clock is Divided by 3 (unit: cycles)**

FSUW1 and FSUW0 Bit Settings		FWW1 and FWW0 Bit Settings		MPY1 and MPY0 Bit Settings							
				10b				11b			
				<i>mpy = 3</i>				<i>mpy = 4</i>			
				tsu(S-W), tsu(A-W)	tw(W)	tcw	th(W-S), th(W-A)	tsu(S-W), tsu(A-W)	tw(W)	tcw	th(W-S), th(W-A)
00b	<i>suw = 0</i>	00b	<i>ww = 1</i>	1	3	6	2	1	4	6	1
		01b	<i>ww = 2</i>	1	6	9	2	1	8	12	3
		10b	<i>ww = 3</i>	1	9	12	2	1	12	15	2
		11b	<i>ww = 4</i>	1	12	15	2	1	16	18	1
01b	<i>suw = 1</i>	00b	<i>ww = 1</i>	4	3	9	2	6	3	12	3
		01b	<i>ww = 2</i>	4	6	12	2	6	7	15	2
		10b	<i>ww = 3</i>	4	9	15	2	6	11	18	1
		11b	<i>ww = 4</i>	4	12	18	2	6	15	24	3
10b	<i>suw = 2</i>	00b	<i>ww = 1</i>	7	3	12	2	9	4	15	2
		01b	<i>ww = 2</i>	7	6	15	2	9	8	18	1
		10b	<i>ww = 3</i>	7	9	18	2	9	12	24	3
		11b	<i>ww = 4</i>	7	12	21	2	9	16	27	2
11b	<i>suw = 3</i>	00b	<i>ww = 1</i>	10	3	15	2	13	4	18	1
		01b	<i>ww = 2</i>	10	6	18	2	13	8	24	3
		10b	<i>ww = 3</i>	10	9	21	2	13	12	27	2
		11b	<i>ww = 4</i>	10	12	24	2	13	16	30	1

**Table 25.14 Write Cycle and Bit Settings: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0, When Peripheral Bus Clock is Divided by 4 (unit: cycles)**

FSUW1 and FSUW0 Bit Settings		FWW1 and FWW0 Bit Settings		MPY1 and MPY0 Bit Settings							
				10b				11b			
				<i>mpy = 3</i>				<i>mpy = 4</i>			
				tsu(S-W), tsu(A-W)	tw(W)	tcw	th(W-S), th(W-A)	tsu(S-W), tsu(A-W)	tw(W)	tcw	th(W-S), th(W-A)
00b	<i>suw = 0</i>	00b	<i>ww = 1</i>	1	3	8	4	1	4	8	3
		01b	<i>ww = 2</i>	1	6	8	1	1	8	12	3
		10b	<i>ww = 3</i>	1	9	12	2	1	12	16	3
		11b	<i>ww = 4</i>	1	12	16	3	1	16	20	3
01b	<i>suw = 1</i>	00b	<i>ww = 1</i>	4	3	8	1	5	4	12	3
		01b	<i>ww = 2</i>	4	6	12	2	5	8	16	3
		10b	<i>ww = 3</i>	4	9	16	3	5	12	20	3
		11b	<i>ww = 4</i>	4	12	20	4	5	16	24	3
10b	<i>suw = 2</i>	00b	<i>ww = 1</i>	8	2	12	2	9	4	16	3
		01b	<i>ww = 2</i>	8	5	16	3	9	8	20	3
		10b	<i>ww = 3</i>	8	8	20	4	9	12	24	3
		11b	<i>ww = 4</i>	8	11	20	1	9	16	28	3
11b	<i>suw = 3</i>	00b	<i>ww = 1</i>	10	3	16	3	13	4	20	3
		01b	<i>ww = 2</i>	10	6	20	4	13	8	24	3
		10b	<i>ww = 3</i>	10	9	20	1	13	12	28	3
		11b	<i>ww = 4</i>	10	12	24	2	13	16	32	3

### 25.3.3 Software Commands

In CPU rewrite mode, software commands enable program and erase operations for the flash memory. Writing commands and reading/writing data should be performed in 16-bit units.

Table 25.15 lists the software commands.

**Table 25.15 Software Commands**

Command	First Command Cycle		Second Command Cycle	
	Address	Data	Address	Data
Enter read array mode	FFFFFF800h	00FFh	—	—
Enter read status register mode <sup>(1)</sup>	FFFFFF800h	0070h	—	—
Clear status register	FFFFFF800h	0050h	—	—
Program <sup>(2)</sup>	FFFFFF800h	0043h	WA	WD
Block erase	FFFFFF800h	0020h	BA	00D0h
Lock bit program	FFFFFF800h	0077h	BA	00D0h
Read lock bit status	FFFFFF800h	0071h	BA	00D0h
Enter read lock bit status mode <sup>(3)</sup>	FFFFFF800h	0071h	—	—
Protect bit program	FFFFFF800h	0067h	PBA	00D0h
Enter read protect bit status mode <sup>(3)</sup>	FFFFFF800h	0061h	—	—

WA: Even address to be written

WD: 16-bit data to be written

BA: Even address within a specific block

PBA: Protect bit address (refer to Table 25.4)

**Notes:**

1. This command cannot be executed in EW1 mode.
2. The program is performed in 64-bit (4-word) units. A sequence of commands consists of commands from the second to fifth. The upper 29 bits of the address WA should be fixed and the lower 3 bits of respective commands from the second to fifth should be set to 000b, 010b, 100b, and 110b for the addresses 0h, 2h, 4h, and 6h, or 8h, Ah, Ch, and Eh.
3. This command should be executed in RAM.

### 25.3.4 Mode Transition

CPU rewrite mode supports four flash memory operating modes:

- Read array mode
- Read status register mode
- Read lock bit status mode
- Read protect bit status mode

When reading the flash memory in these modes, the memory data, the status register value, the state of the lock bit in the read block, and the state of the protect bit are individually read. Details are listed in Tables 25.16 to 25.18.

**Table 25.16 Status Register**

Bit	Bit Symbol	Bit Name	Definition	
			0	1
b15-b8	—	Disabled bit	—	—
b7	SR7	Sequencer status	BUSY	READY
b6	—	Reserved bit	—	—
b5	SR5	Erase status	Successfully completed	Error
b4	SR4	Program status	Successfully completed	Error
b3	—	Reserved bit	—	—
b2	—	Reserved bit	—	—
b1	—	Reserved bit	—	—
b0	—	Reserved bit	—	—

**Table 25.17 Lock Bit Status**

Bit	Bit Symbol	Bit Name	Definition	
			0	1
b15-b7	—	Disabled bit	—	—
b6	LBS	Lock bit status	Locked	Unlocked
b5-b0	—	Disabled bit	—	—

**Table 25.18 Protect Bit Status**

Bit	Bit Symbol	Bit Name	Definition	
			0	1
b15-b7	—	Disabled bit	—	—
b6	PBS	Protect bit status	Protected	Unprotected
b5-b0	—	Disabled bit	—	—

In these operating modes, program or erase operation can be performed by software commands. After an operation is completed, the flash memory module automatically enters read array mode (in EW1 mode) or read status register mode (in EW0 mode).

### 25.3.5 Issuing Software Commands

This section describes how to issue software commands.

These commands should be issued while the RDY bit in the FMSR0 register is 1 (ready).

#### 25.3.5.1 Enter Read Array Mode Command

Execute this command to enter read array mode.

When 00FFh is written to address FFFF800h, the flash memory enters read array mode. In this mode, the value stored to a given address in memory can be read.

In EW1 mode, the flash memory is always in read array mode.

#### 25.3.5.2 Enter Read Status Register Mode

Execute this command to enter read status register mode.

When 0070h is written to address FFFF800h, the status register value is read in any address of the flash memory.

Do not issue this command in EW1 mode.

#### 25.3.5.3 Clear Status Register

Execute this command to reset the status register in the flash memory.

When 0050h is written to address FFFF800h, bits SR5 and SR4 in the status register become 0 (successfully completed) (refer to Table 25.16). Consequently, bits EERR and WERR in the FMSR0 register become 0 (no errors).



### 25.3.5.4 Program Command

Execute this command to program the flash memory in 8-byte (4-word) units.

To start automatic programming (program and program-verify operations), write 0043h to address FFFF800h, then write data to addresses  $8n + 0$  to  $8n + 6$ . Verify that the FCA bit in the FMR0 register is 0 just before executing the final command.

To monitor the automatic program operation, read the RDY bit in the FMSR0 register. This bit becomes 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.

The operation result can be verified by the WERR bit in the FMSR0 register (refer to 25.3.6 “Status Check”).

Do not write additional data to an address that is already programmed.

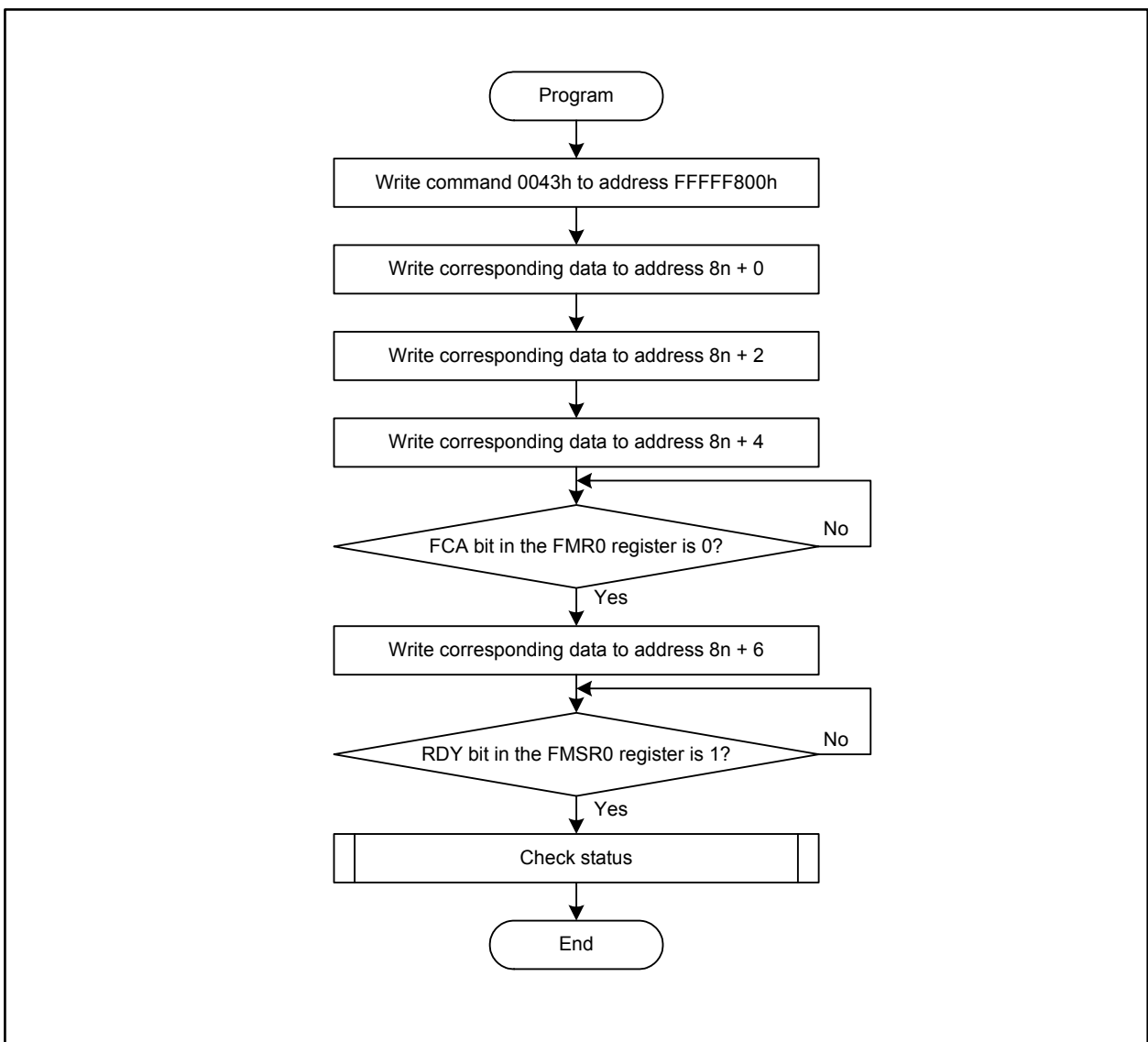


Figure 25.14 Program Command Execution Flowchart

### 25.3.5.5 Block Erase Command

Execute this command to erase a specified block in the flash memory.

To start automatic erasing of a specified block (erase and erase-verify operations), write 0020h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, then write 00D0h to an even address in the corresponding block.

To monitor the automatic erase operation, read the RDY bit in the FMSR0 register. This bit becomes 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.

The operation result can be verified by the EERR bit in the FMSR0 register (refer to 25.3.6 “Status Check”).

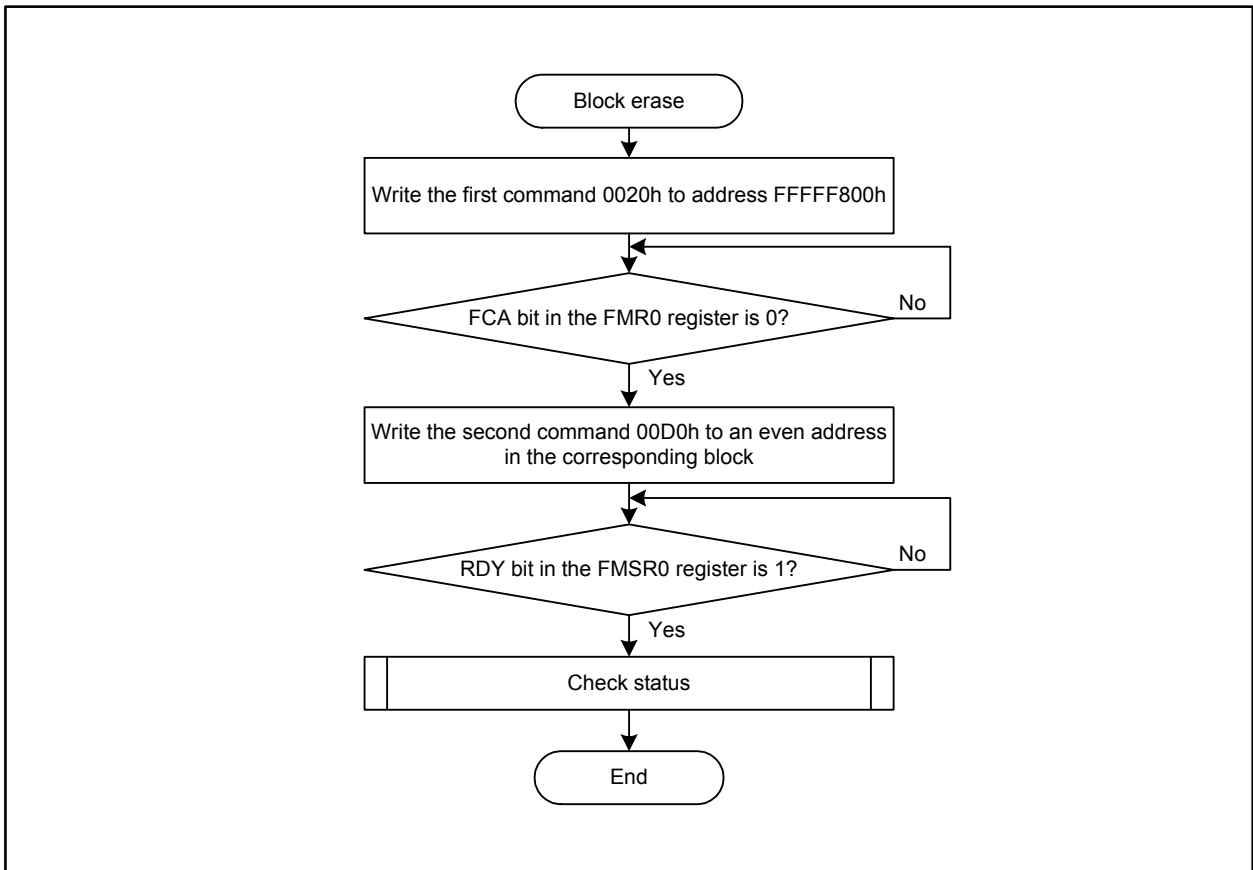


Figure 25.15 Block Erase Command Execution Flowchart

### 25.3.5.6 Lock Bit Program Command

Execute this command to lock a specified block in the flash memory.

To lock the block, write 0077h to address FFFF800h, verify that the FCA bit in the FMR0 register is 0, then write 00D0h to an even address in the corresponding block. Then the lock bit of the block becomes 0 (locked).

To monitor the lock bit program, read the RDY bit in the FMSR0 register. This bit becomes 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.

The state of the lock bit can be verified by the read lock bit status command if the LBM bit in the FMR0 register is 1 (read by the LBS bit) (refer to 25.3.5.7 “Read Lock Bit Status Command”). If the LBM bit is 0 (read via data bus), enter read lock bit status mode (refer to 25.3.5.8 “Enter Read Lock Bit Status Mode Command”).

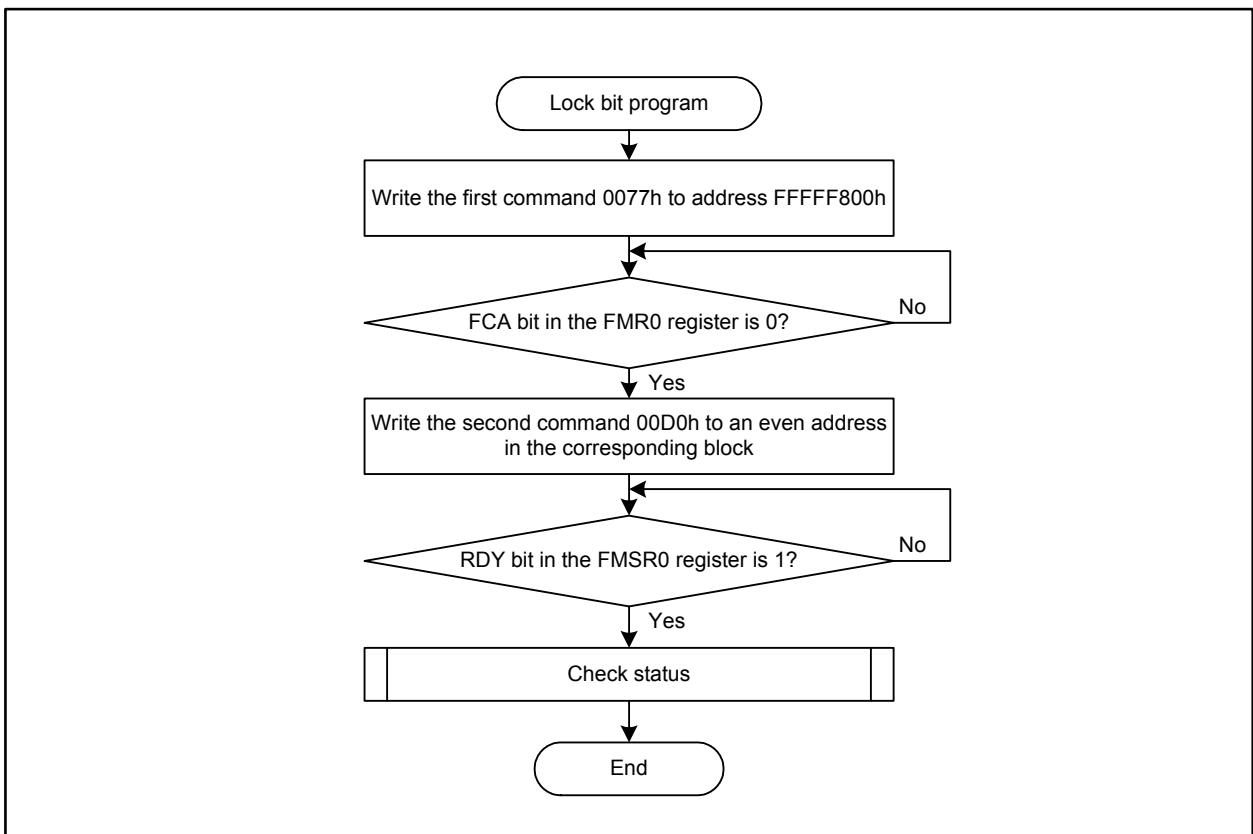


Figure 25.16 Lock Bit Program Command Execution Flowchart

### 25.3.5.7 Read Lock Bit Status Command

Execute this command to verify if a specified block in the flash memory is locked. This command can be used when the LBM bit in the FMR0 register is 1 (read by the LBS bit).

The LBS bit in the FMSR0 register reflects the lock bit status of the specified block when the following is performed: first write 0071h to address FFFFF800h and verify that the FCA bit in the FMR0 register becomes 0. Then write 00D0h to an even address of the corresponding block.

Read the LBS bit after the RDY bit in the FMSR0 register becomes 1 (ready).

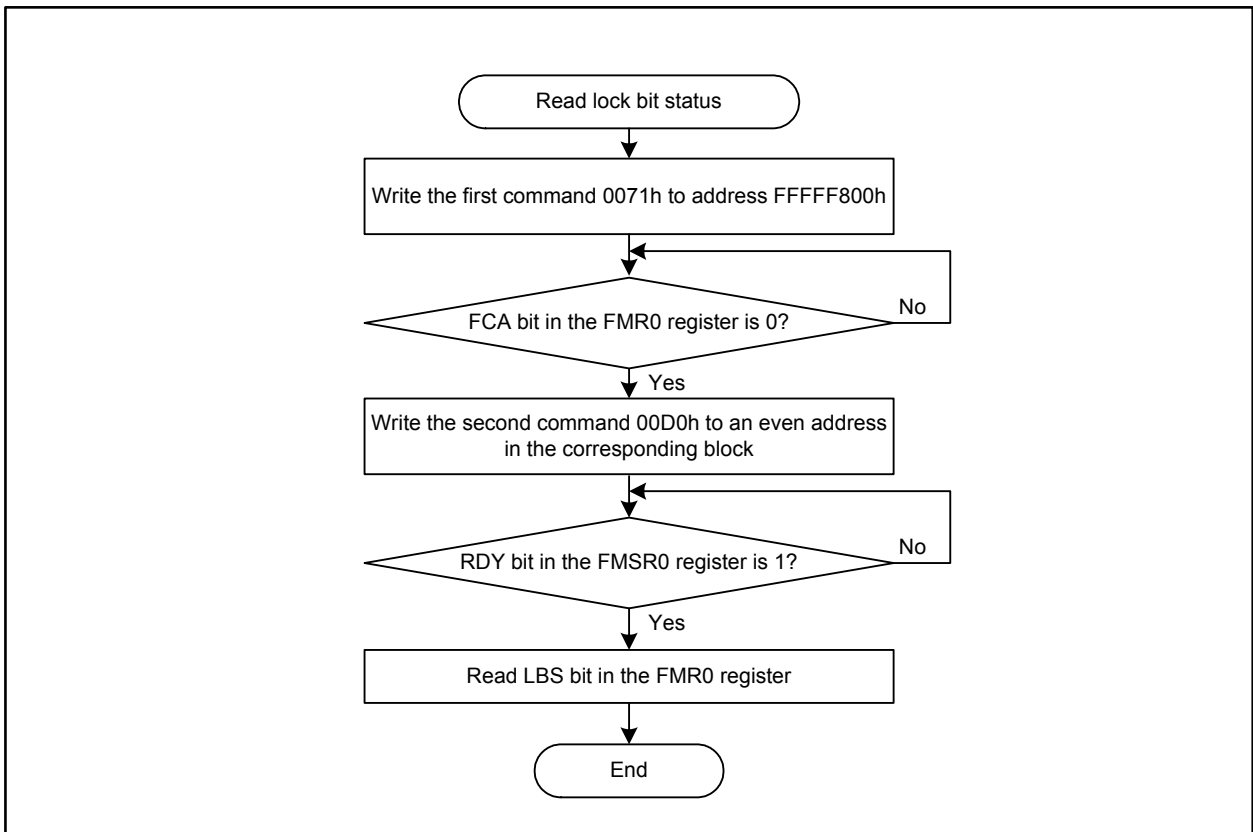


Figure 25.17 Read Lock Bit Status Command Execution Flowchart

### 25.3.5.8 Enter Read Lock Bit Status Mode Command

Execute this command to enter read lock bit status mode. This command is enabled when the LBM bit in the FMR0 register is 0 (read via data bus).

To read the lock bit status of the read block, write 0071h to address FFFFF800h (refer to Table 25.17).

The status is read in any address of the flash memory.

Execute this command in RAM.

### 25.3.5.9 Protect Bit Program Command

Execute this command to protect a specific block in the flash memory. ROM code protection is enabled by setting one of the protect bits of the block to 0.

To set the protect bit of the designated block to 0 (protected), write 0067h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, and then write 00D0h to the protect bit of the corresponding block (refer to Table 25.4).

To monitor the protect bit program, read the RDY bit in the FMSR0 register. This bit becomes 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.

To verify the state of protect bit, enter read protect bit status mode (refer to 25.3.5.10 “Enter Read Protect Bit Status Mode Command”), then read the flash memory.

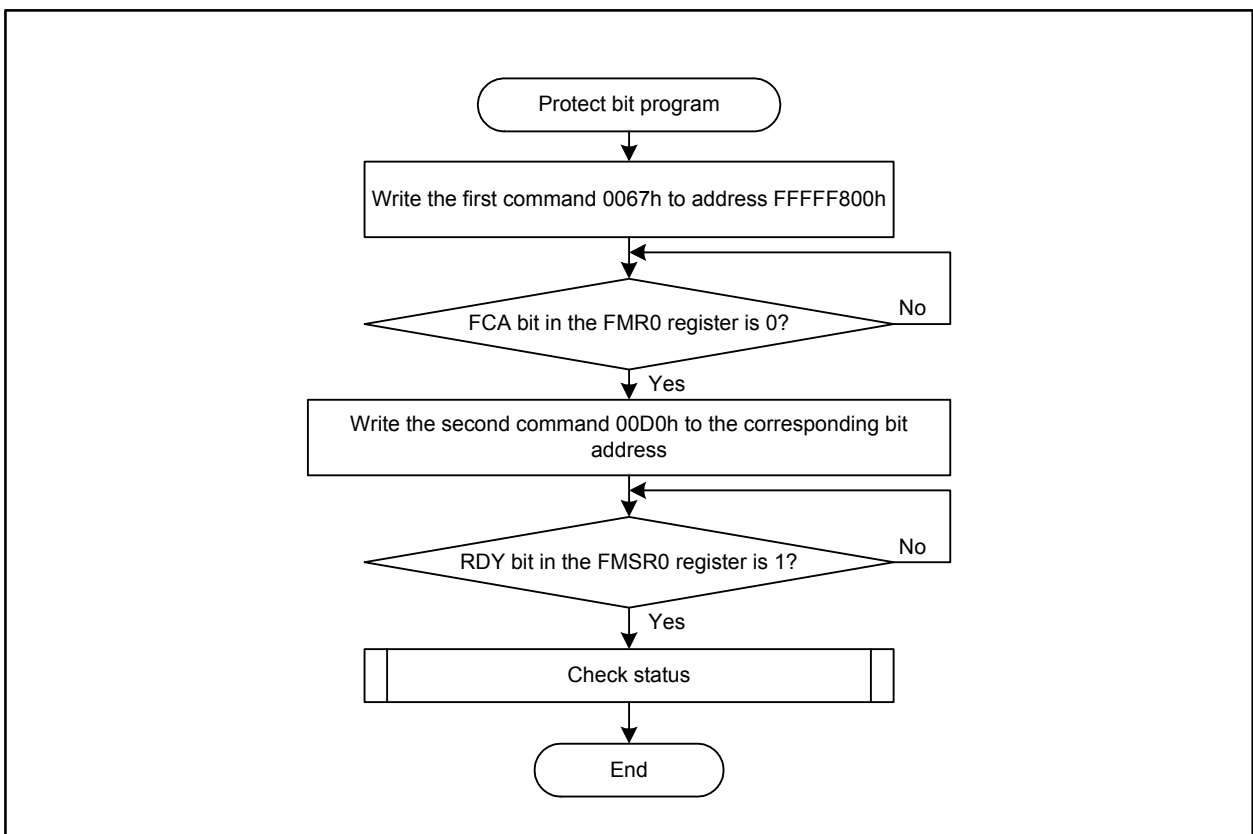


Figure 25.18 Protect Bit Program Command Execution Flowchart

### 25.3.5.10 Enter Read Protect Bit Status Mode Command

Execute this command to enter read protect bit status mode.

To read the protect bit status of the read block, write 0061h to address FFFFF800h (refer to Table 25.18). The status is read from any address in the flash memory.

Execute this command in RAM.

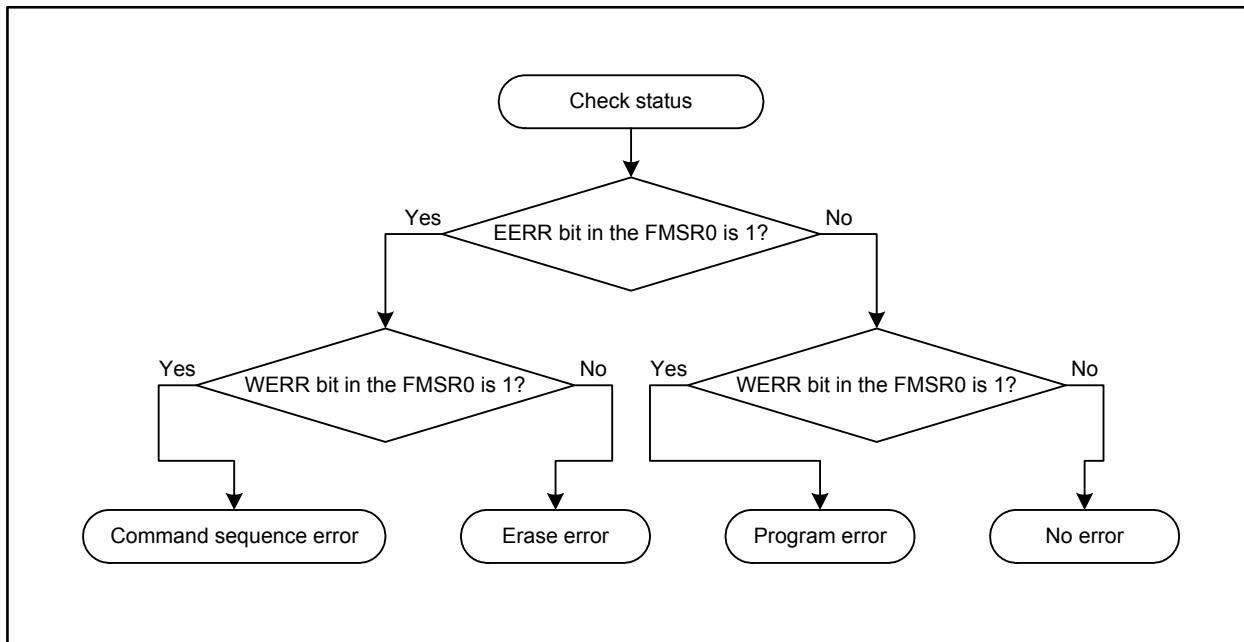
### 25.3.6 Status Check

To verify if a software command is successfully executed, read the EERR or WERR bit in the FMSR0 register, or the SR5 bit or SR4 bit in the status register.

Table 25.19 lists status and errors indicated by these bits and Figure 25.19 shows the flowchart of the status check.

**Table 25.19 Status and Errors**

FMSR0 Register (Status Register)		Error	Source of Error
EERR bit (SR5 bit)	WERR bit (SR4 bit)		
1	1	Command sequence error	<ul style="list-style-type: none"> <li>Data other than 00D0h or 00FFh (command to cancel) was written as the last command of two commands</li> <li>An unavailable address was specified by an address specifying command</li> </ul>
1	0	Erase error	<ul style="list-style-type: none"> <li>Attempted to erase a locked block</li> <li>Corresponding block was not erased properly</li> </ul>
0	1	Program error	<ul style="list-style-type: none"> <li>Attempted to program a locked block</li> <li>Data was not programmed properly</li> <li>Lock bit was not programmed properly</li> <li>Protect bit was not programmed properly</li> </ul>
0	0	No error	



**Figure 25.19 Status Check Flowchart**

When an error occurs, execute the clear status register command and then handle the error.

If erase errors or program errors occur frequently even though the program is correct, the corresponding block may be disabled.

## 25.4 Standard Serial I/O Mode

In standard serial I/O mode, an R32C/111 Group compatible serial programmer can be used to rewrite the flash memory while the MCU is mounted on a board.

For further information on the serial programmer, contact your serial programmer manufacturer and refer to the user's manual included with the serial programmer for instructions.

As shown in Table 25.20, this mode provides two types of transmit/receive mode: Standard serial I/O mode 1 which uses a synchronous serial interface, and standard serial I/O mode 2 which uses UART.

**Table 25.20 Standard Serial I/O Mode Specifications**

Item		Standard Serial I/O Mode 1	Standard Serial I/O Mode 2
Transmit/receive mode		Synchronous serial I/O	UART
Transmit/receive bit rate		High	Low
Serial interface to be used		UART1	UART1
Pin settings	CNVSS	High	High
	CE (P5_0) (1)	High	High
	EPM (P5_5) (2)	Low	Low
	SCLK (P6_5)	In reset: Low In transmission/reception: Transmit/receive clock	In reset: Low In transmission/reception: Unused
Pin functions	BUSY (P6_4)	BUSY signal	Monitor to check program operation
	RXD (P6_6)	Serial data input	Serial data input
	TXD (P6_7)	Serial data output	Serial data output

Notes:

1. This signal is assigned to port P8\_0 in the 64-pin package.
2. This signal is assigned to port P8\_1 in the 64-pin package.

Tables 25.21 and 25.22 list the pin definitions and functions in standard serial I/O mode. Figures 25.20 and 25.21, and Figures 25.22 and 25.23 show examples of a circuit application in standard serial I/O modes 1 and 2, respectively. Refer to the serial programmer user manual to handle pins controlled by the serial programmer.

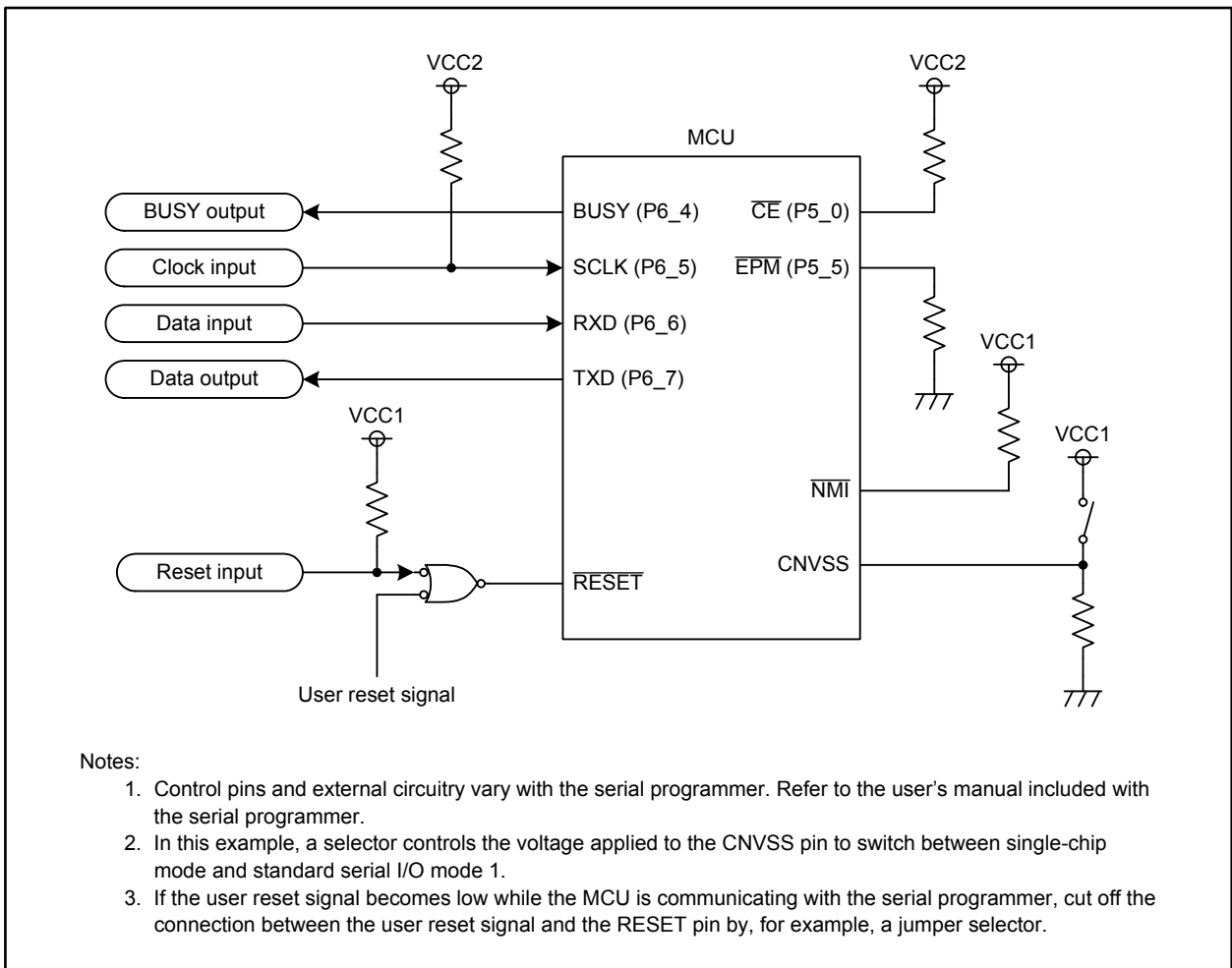
**Table 25.21 Pin Definitions and Functions in Standard Serial I/O Mode (100-pin package)**

Symbol	Function	I/O	Power Supply	Description
VCC1, VCC2, VSS	Power supply input	I	—	Applicable as follows: VCC1 and VCC2 = guaranteed voltage for program/erase operations, VSS = 0 V
VDC1, VDC0	Connecting pins for decoupling capacitor	—	—	A decoupling capacitor for internal voltage should be connected between VDC0 and VDC1
CNVSS	CNVSS	I	VCC1	This pin should be connected to VCC1 via a resistor
RESET	Reset input	I	VCC1	Reset input pin. While the RESET pin is driven low, at least 20 clock cycles should be input at the XIN pin
XIN	Main clock input	I	VCC1	A ceramic resonator or a crystal oscillator should be connected between pins XIN and XOUT. An external clock should be input at XIN while leaving XOUT open
XOUT	Main clock output	O	VCC1	
NSD	Debug port	I/O	VCC1	This pin should be connected to VCC1 via a resistor of 1 to 4.7 k $\Omega$
AVCC, AVSS	Analog power supply	I	—	AVCC and AVSS should be connected to VCC1 and VSS, respectively
VREF	Reference voltage input	I	—	Reference voltage input for the A/D converter and D/A converter
P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7	Input port	I	VCC2	High or low should be input, or the ports should be left open
P5_0	$\overline{CE}$ input	I	VCC2	High should be input
P5_1 to P5_4	Input port	I	VCC2	High or low should be input, or the ports should be left open
P5_5	$\overline{EPM}$ input	I	VCC2	Low should be input
P5_6, P5_7	Input port	I	VCC2	High or low should be input, or the ports should be left open
P6_0 to P6_3	Input port	I	VCC1	High or low should be input, or the ports should be left open
P6_4	BUSY output	O	VCC1	Standard serial I/O mode 1: BUSY output pin Standard serial I/O mode 2: Program operation monitor
P6_5	SCLK input	I	VCC1	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Low should be input
P6_6	Data input RXD	I	VCC1	Serial data input pin
P6_7	Data output TXD	O	VCC1	Serial data output pin
P7_0 to P7_7, P8_0 to P8_4	Input port	I	VCC1	High or low should be input, or the ports should be left open
P8_5	NMI input	I	VCC1	This pin should be connected to VCC1 via a resistor
P8_6, P8_7, P9_1, P9_3 to P9_7, P10_0 to P10_7	Input port	I	VCC1	High or low should be input, or the ports should be left open

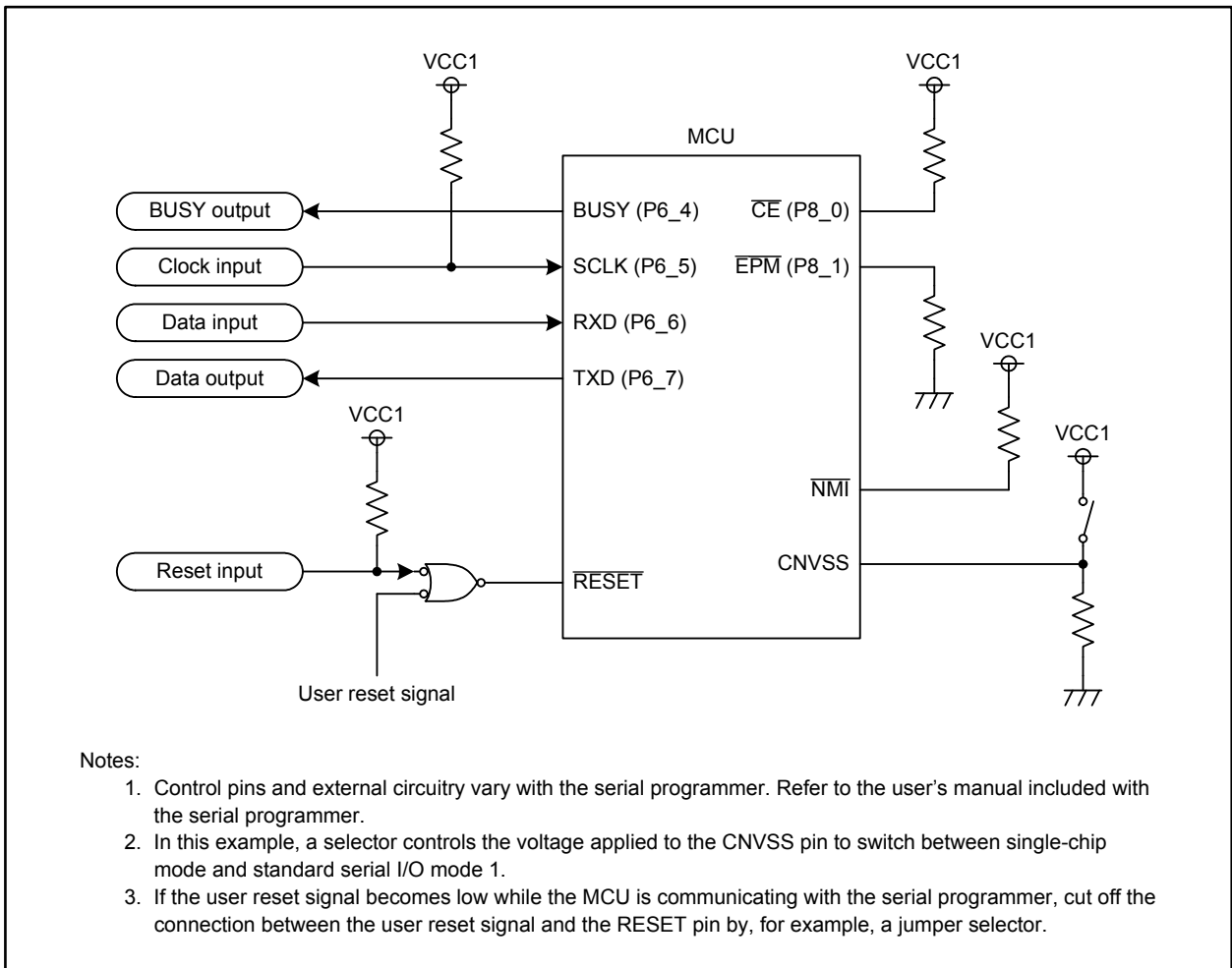


**Table 25.22 Pin Definitions and Functions in Standard Serial I/O Mode (64-pin package)**

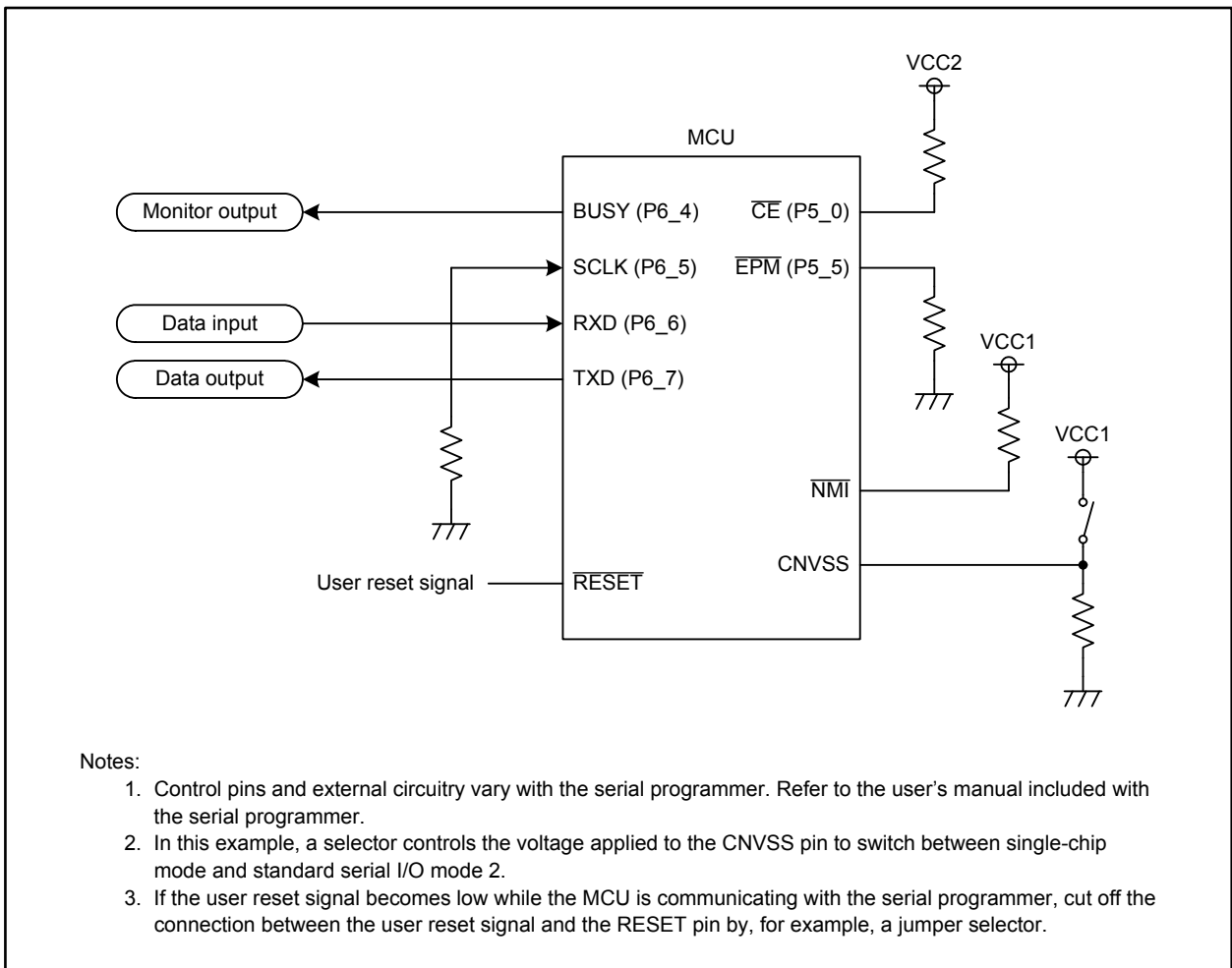
Symbol	Function	I/O	Power Supply	Description
VCC1, VSS	Power supply input	I	—	Applicable as follows: VCC1 = guaranteed voltage for program/erase operations, VSS = 0 V
VDC1, VDC0	Connecting pins for decoupling capacitor	—	—	A decoupling capacitor for internal voltage should be connected between VDC0 and VDC1
CNVSS	CNVSS	I	VCC1	This pin should be connected to VCC1 via a resistor
RESET	Reset input	I	VCC1	Reset input pin. While the RESET pin is driven low, at least 20 clock cycles should be input at the XIN pin
XIN	Main clock input	I	VCC1	A ceramic resonator or a crystal oscillator should be connected between pins XIN and XOUT. An external clock should be input at XIN while leaving XOUT open
XOUT	Main clock output	O	VCC1	
NSD	Debug port	I/O	VCC1	This pin should be connected to VCC1 via a resistor of 1 to 4.7 k $\Omega$
AVCC, AVSS	Analog power supply	I	—	AVCC and AVSS should be connected to VCC1 and VSS, respectively
VREF	Reference voltage input	I	—	Reference voltage input for the A/D converter and D/A converter
P0_0 to P0_3, P1_5 to P1_7, P2_0 to P2_7, P3_0 to P3_3	I/O port	I	VCC1	High or low should be input, or the ports should be left open
P6_0 to P6_3	Input port	I	VCC1	High or low should be input, or the ports should be left open
P6_4	BUSY output	O	VCC1	Standard serial I/O mode 1: BUSY output pin Standard serial I/O mode 2: Program operation monitor
P6_5	SCLK input	I	VCC1	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Low should be input
P6_6	Data input RXD	I	VCC1	Serial data input pin
P6_7	Data output TXD	O	VCC1	Serial data output pin
P7_0 to P7_7	Input port	I	VCC1	High or low should be input, or the ports should be left open
P8_0	CE input	I	VCC1	High should be input
P8_1	EPM input	I	VCC1	Low should be input
P8_2 to P8_4	Input port	I	VCC1	High or low should be input, or the ports should be left open
P8_5	NMI input	I	VCC1	This pin should be connected to VCC1 via a resistor
P8_6, P8_7, P9_3, P10_0 to P10_7	Input port	I	VCC1	High or low should be input, or the ports should be left open



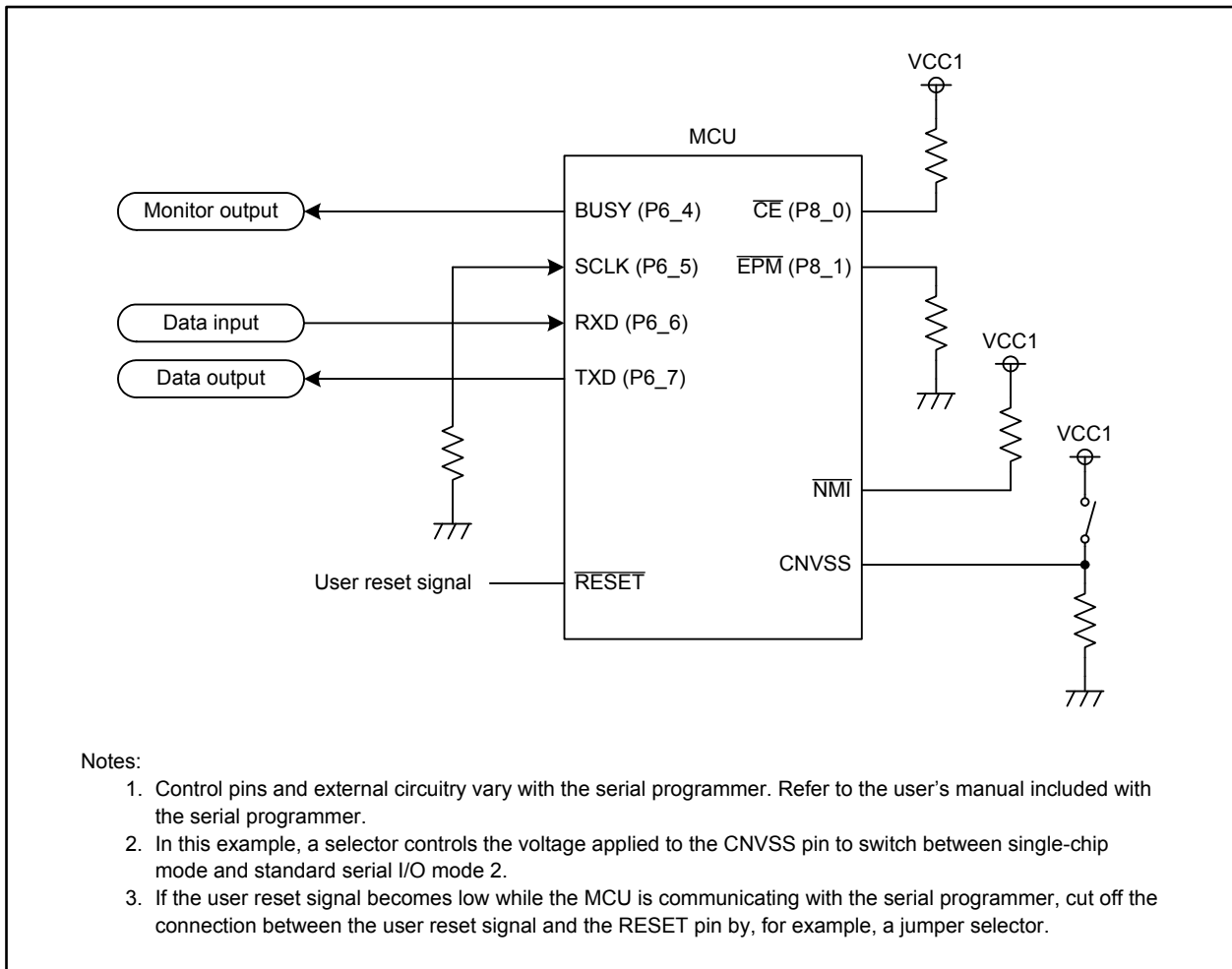
**Figure 25.20 Circuit Application in Standard Serial I/O Mode 1 (100-pin package)**



**Figure 25.21 Circuit Application in Standard Serial I/O Mode 1 (64-pin package)**



**Figure 25.22 Circuit Application in Standard Serial I/O Mode 2 (100-pin package)**



**Figure 25.23 Circuit Application in Standard Serial I/O Mode 2 (64-pin package)**

## 25.5 Parallel I/O mode

In parallel I/O mode, an R32C/111 Group compatible parallel programmer can be used to rewrite the flash memory.

For further information on the parallel programmer, contact your parallel programmer manufacturer and refer to the user's manual included with your parallel programmer for instructions.

## 25.6 Notes on Flash Memory Rewriting

### 25.6.1 Note on Power Supply

- Keep the supply voltage constant within the range specified in the electrical characteristics while a rewrite operation on the flash memory is in progress. If the supply voltage goes beyond the guaranteed value, the device cannot be guaranteed.

### 25.6.2 Note on Hardware Reset

- Do not perform a hardware reset while a rewrite operation on the flash memory is in progress.

### 25.6.3 Note on Flash Memory Protection

- If an ID code written in an assigned address has an error, any read/write operation on the flash memory in standard serial I/O mode is disabled.

### 25.6.4 Notes on Programming

- Do not set the FEW bit in the FMCR register to 1 (CPU rewrite mode) in low speed mode or low power mode.
- The program, block erase, lock bit program, and protect bit program are interrupted by an NMI, a watchdog timer interrupt, an oscillator stop detection interrupt, or a low voltage detection interrupt. If any of the software commands above are interrupted, erase the corresponding block and then execute the same command again. If the block erase command is interrupted, the lock bit and protect bit values become undefined. Therefore, disable the lock bit, and then execute the block erase command again.

### 25.6.5 Notes on Interrupts

- EW0 mode
  - To use interrupts assigned to the relocatable vector table, the vector table should be addressed in RAM space.
  - When an NMI, watchdog timer interrupt, oscillator stop detection interrupt, or low voltage detection interrupt occurs, the flash memory module automatically enters read array mode. Therefore, these interrupts are enabled even during a rewrite operation. However, the rewrite operation in progress is aborted by the interrupts and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.
  - Instructions BRK, INTO, and UND, which refer to data on the flash memory, cannot be used in this mode.
- EW1 mode
  - Interrupts assigned to the relocatable vector table should not be accepted during program or block erase operation.
  - The watchdog timer interrupt should not be generated.
  - When an NMI, watchdog timer interrupt, oscillator stop detection interrupt, or low voltage detection interrupt occurs, the flash memory module automatically enters read array mode. Therefore, these interrupts are enabled even during a rewrite operation. However, the rewrite operation in progress is aborted by the interrupts and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the EWM bit in the FMR0 register to 1 (EW1 mode) and the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.

### 25.6.6 Notes on Rewrite Control Program

- EW0 mode
  - If the supply voltage drops during the rewrite operation of blocks having the rewrite control program, the rewrite control program may not be successfully rewritten, and the rewrite operation itself may not be performed. In this case, perform the rewrite operation by serial programmer or parallel programmer.
- EW1 mode
  - Do not rewrite blocks having the rewrite control program.

### 25.6.7 Notes on Number of Program/Erase Cycles and Software Command Execution Time

- The time to execute software commands (program, block erase, lock bit program, and protect bit program) increases as the number of program/erase cycles increases. If the number of program/erase cycles exceeds the endurance value specified in the electrical characteristics, it may take an unpredictable amount of time to execute the software commands. The wait time for executing software commands should be set much longer than the execution time specified in the electrical characteristics.

### 25.6.8 Other Notes

- The minimum values of program/erase cycles specified in the electrical characteristics are the maximum values that can guarantee the initial performance of the flash memory. The program/erase operation may still be performed even if the number of program/erase cycles exceeds the guaranteed values.
- Chips repeatedly programmed and erased for debugging should not be used for commercial products.

## 26. Electrical Characteristics

**Table 26.1 Absolute Maximum Ratings (1)**

Symbol	Characteristic		Condition	Value (2)	Unit
$V_{CC1}, V_{CC2}$	Supply voltage		$V_{CC1} = AV_{CC}$	-0.3 to 6.0	V
$V_{CC2}$	Supply voltage		—	-0.3 to $V_{CC1}$	V
$AV_{CC}$	Analog supply voltage		$V_{CC1} = AV_{CC}$	-0.3 to 6.0	V
$V_I$	Input voltage	XIN, RESET, CNVSS, NSD, $V_{REF}$ , P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_7, P9_1, P9_3 to P9_7, P10_0 to P10_7 (3)		-0.3 to $V_{CC1} + 0.3$	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7 (3)		-0.3 to $V_{CC2} + 0.3$	V
		P7_0, P7_1		-0.3 to 6.0	V
$V_O$	Output voltage	XOUT, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3 to P9_7, P10_0 to P10_7 (3)		-0.3 to $V_{CC1} + 0.3$	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7 (3)		-0.3 to $V_{CC2} + 0.3$	V
		P7_0, P7_1		-0.3 to 6.0	V
$P_d$	Power consumption		$T_a = 25^\circ\text{C}$	500	mW
—	Operating temperature range			-40 to 85	$^\circ\text{C}$
$T_{stg}$	Storage temperature range			-65 to 150	$^\circ\text{C}$

**Notes:**

- Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
- The  $V_{CC2}$  pin is available in the 100-pin package only. It should be considered as  $V_{CC1}$  in the 64-pin package.
- Ports P0\_4 to P0\_7, P1\_0 to P1\_4, P3\_4 to P3\_7, P4, P5, P9\_1, and P9\_4 to P9\_7 are available in the 100-pin package only.



**Table 26.2 Operating Conditions (1/5) (1)**

Symbol	Characteristic		Value (2)			Unit	
			Min.	Typ.	Max.		
$V_{CC1}$ , $V_{CC2}$	Digital supply voltage ( $V_{CC1} \geq V_{CC2}$ )		3.0	5.0	5.5	V	
$AV_{CC}$	Analog supply voltage			$V_{CC1}$		V	
$V_{REF}$	Reference voltage		3.0		$V_{CC1}$	V	
$V_{SS}$	Digital ground voltage			0		V	
$AV_{SS}$	Analog ground voltage			0		V	
$dV_{CC1}/dt$	$V_{CC1}$ ramp up rate ( $V_{CC1} < 2.0$ V)		0.05			V/ms	
$V_{IH}$	High level input voltage	P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7 (4)	$0.8 \times V_{CC2}$		$V_{CC2}$	V	
		XIN, $\overline{RESET}$ , CNVSS, NSD, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_7 (3), P9_1, P9_3 to P9_7, P10_0 to P10_7 (4)	$0.8 \times V_{CC1}$		$V_{CC1}$	V	
		P7_0, P7_1	$0.8 \times V_{CC1}$		6.0	V	
		P0_0 to P0_7, P1_0 to P1_7 (4)	in single-chip mode	$0.8 \times V_{CC2}$		$V_{CC2}$	V
			in memory expansion mode or microprocessor mode (5)	$0.5 \times V_{CC2}$		$V_{CC2}$	V
$V_{IL}$	Low level input voltage	P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7 (4)	0		$0.2 \times V_{CC2}$	V	
		XIN, $\overline{RESET}$ , CNVSS, NSD, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7 (3), P9_1, P9_3 to P9_7, P10_0 to P10_7 (4)	0		$0.2 \times V_{CC1}$	V	
		P0_0 to P0_7, P1_0 to P1_7 (4)	in single-chip mode	0		$0.2 \times V_{CC2}$	V
			in memory expansion mode or microprocessor mode (5)	0		$0.16 \times V_{CC2}$	V
$T_{opr}$	Operating temperature range	N version	-20		85	°C	
		D version	-40		85	°C	

## Notes:

1. The device is operationally guaranteed under these operating conditions.
2. The  $V_{CC2}$  pin is available in the 100-pin package only. It should be considered as  $V_{CC1}$  in the 64-pin package.
3.  $V_{IH}$  and  $V_{IL}$  for P8\_7 are specified for P8\_7 as a programmable port. These values are not applicable for P8\_7 as XCIN.
4. Ports P0\_4 to P0\_7, P1\_0 to P1\_4, P3\_4 to P3\_7, P4, P5, P9\_1, and P9\_4 to P9\_7 are available in the 100-pin package only.
5. Memory expansion mode and microprocessor mode are available in the 100-pin package only.

**Table 26.3 Operating Conditions (2/5)****( $V_{CC1} = V_{CC2} = 3.0$  to  $5.5$  V,  $V_{SS} = 0$  V, and  $T_a = T_{opr}$ , unless otherwise noted) (1)**

Symbol	Characteristic		Value (2)			Unit
			Min.	Typ.	Max.	
$C_{VDC}$	Decoupling capacitance for voltage regulator	Inter-pin voltage: 1.5 V	2.4		10.0	$\mu$ F

## Notes:

1. The device is operationally guaranteed under these operating conditions.
2. This value should be met with due consideration to the following conditions: operating temperature, DC bias, aging, etc.

**Table 26.4 Operating Conditions (3/5)****( $V_{CC1} = V_{CC2} = 3.0$  to  $5.5$  V,  $V_{SS} = 0$  V, and  $T_a = T_{opr}$ , unless otherwise noted) (1)**

Symbol	Characteristic		Value			Unit
			Min.	Typ.	Max.	
$I_{OH(peak)}$	High level peak output current (2)	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3 to P9_7, P10_0 to P10_7 (3)			-10.0	mA
$I_{OH(avg)}$	High level average output current (4)	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3 to P9_7, P10_0 to P10_7 (3)			-5.0	mA
$I_{OL(peak)}$	Low level peak output current (2)	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3 to P9_7, P10_0 to P10_7 (3)			10.0	mA
$I_{OL(avg)}$	Low level average output current (4)	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3 to P9_7, P10_0 to P10_7 (3)			5.0	mA

## Notes:

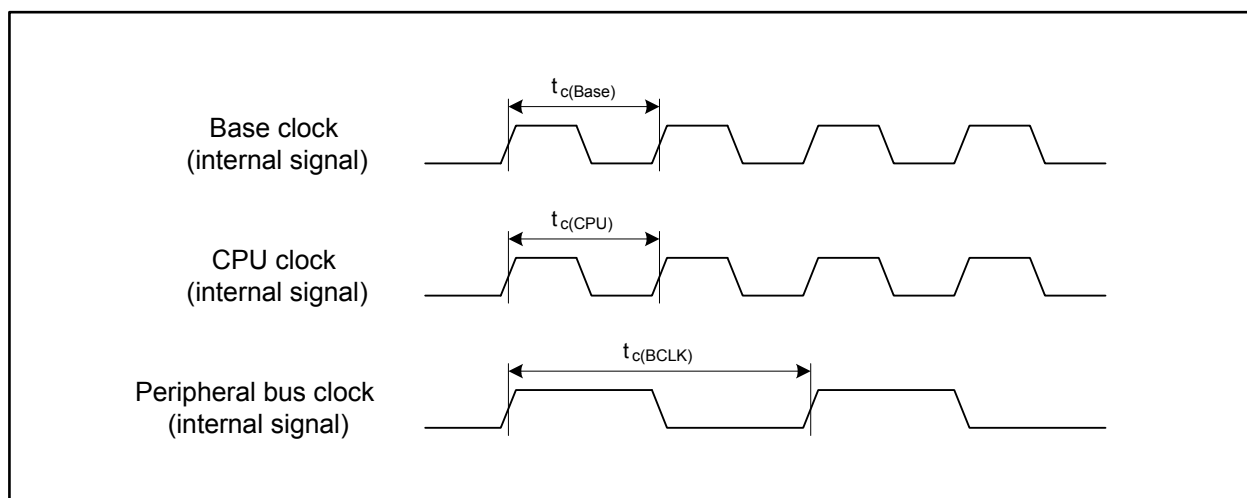
- The device is operationally guaranteed under these operating conditions.
- The following conditions should be satisfied:
  - The sum of  $I_{OL(peak)}$  of ports P0, P1, P2, P8\_6, P8\_7, P9, and P10 is 80 mA or less.
  - The sum of  $I_{OL(peak)}$  of ports P3, P4, P5, P6, P7, and P8\_0 to P8\_4 is 80 mA or less.
  - The sum of  $I_{OH(peak)}$  of ports P0, P1, and P2 is -40 mA or less.
  - The sum of  $I_{OH(peak)}$  of ports P8\_6, P8\_7, P9, and P10 is -40 mA or less.
  - The sum of  $I_{OH(peak)}$  of ports P3, P4, and P5 is -40 mA or less.
  - The sum of  $I_{OH(peak)}$  of ports P6, P7, and P8\_0 to P8\_4 is -40 mA or less.
- Ports P0\_4 to P0\_7, P1\_0 to P1\_4, P3\_4 to P3\_7, P4, P5, and P9\_4 to P9\_7 are available in the 100-pin package only.
- Average value within 100 ms.

**Table 26.5 Operating Conditions (4/5)****( $V_{CC1} = V_{CC2} = 3.0$  to  $5.5$  V,  $V_{SS} = 0$  V, and  $T_a = T_{opr}$ , unless otherwise noted) (1)**

Symbol	Characteristic	Value			Unit
		Min.	Typ.	Max.	
$f_{(XIN)}$	Main clock oscillator frequency	4		16	MHz
$f_{(XRef)}$	Reference clock frequency	2		4	MHz
$f_{(PLL)}$	PLL clock oscillator frequency	96		128	MHz
$f_{(Base)}$	Base clock frequency			50	MHz
$t_{c(Base)}$	Base clock cycle time	20			ns
$f_{(CPU)}$	CPU operating frequency			50	MHz
$t_{c(CPU)}$	CPU clock cycle time	20			ns
$f_{(BCLK)}$	Peripheral bus clock operating frequency			25	MHz
$t_{c(BCLK)}$	Peripheral bus clock cycle time	40			ns
$f_{(PER)}$	Peripheral clock source frequency			32	MHz
$f_{(XCIN)}$	Sub clock oscillator frequency		32.768	62.5	kHz

Note:

1. The device is operationally guaranteed under these operating conditions.

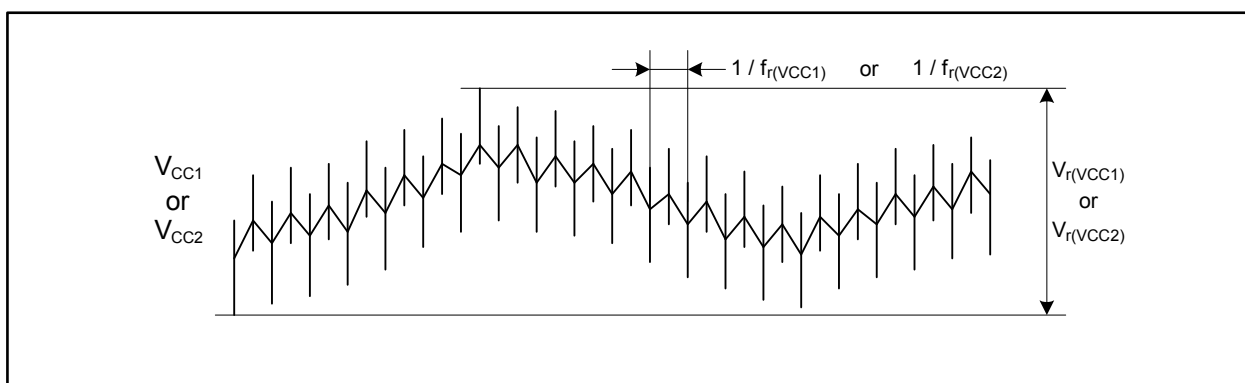
**Figure 26.1 Clock Cycle Time**

**Table 26.6 Operating Conditions (5/5)****( $V_{CC1} = V_{CC2} = 3.0$  to  $5.5$  V,  $V_{SS} = 0$  V, and  $T_a = T_{opr}$ , unless otherwise noted) (1)**

Symbol	Characteristic	Value			Unit
		Min.	Typ.	Max.	
$V_{r(VCC1)}$	Allowable ripple voltage	$V_{CC1} = 5.0$ V		0.5	Vp-p
		$V_{CC1} = 3.0$ V		0.3	Vp-p
$V_{r(VCC2)}$	Allowable ripple voltage	$V_{CC2} = 5.0$ V		0.5	Vp-p
		$V_{CC2} = 3.0$ V		0.3	Vp-p
$dV_{r(VCC1)}/dt$	Ripple voltage gradient	$V_{CC1} = 5.0$ V		$\pm 0.3$	V/ms
		$V_{CC1} = 3.0$ V		$\pm 0.3$	V/ms
$dV_{r(VCC2)}/dt$	Ripple voltage gradient	$V_{CC2} = 5.0$ V		$\pm 0.3$	V/ms
		$V_{CC2} = 3.0$ V		$\pm 0.3$	V/ms
$f_{r(VCC1)}$	Allowable ripple frequency			10	kHz
$f_{r(VCC2)}$	Allowable ripple frequency			10	kHz

Note:

- The device is operationally guaranteed under these operating conditions.

**Figure 26.2 Ripple Waveform**

**Table 26.7 Electrical Characteristics of RAM****( $V_{CC1} = V_{CC2} = 3.0$  to  $5.5$  V,  $V_{SS} = 0$  V, and  $T_a = T_{opr}$ , unless otherwise noted)**

Symbol	Characteristic	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
$V_{RDR}$	RAM data retention voltage <sup>(1)</sup>	In stop mode	2.0			V

Note:

1. The value listed in the table is the minimum  $V_{CC1}$  to retain RAM data.

**Table 26.8 Electrical Characteristics of Flash Memory****( $V_{CC1} = V_{CC2} = 3.0$  to  $5.5$  V,  $V_{SS} = 0$  V, and  $T_a = T_{opr}$ , unless otherwise noted)**

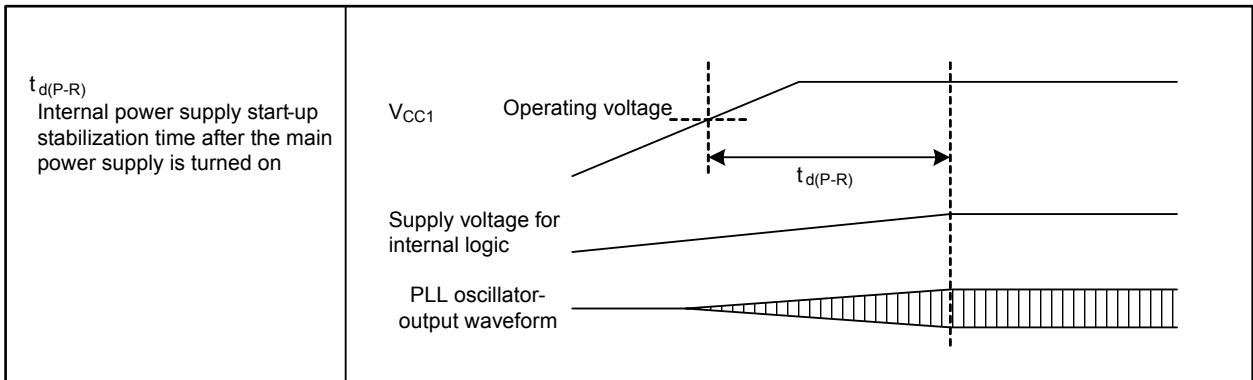
Symbol	Characteristic		Value			Unit
			Min.	Typ.	Max.	
—	Program/erase cycles <sup>(1)</sup>	Program area	1000			Cycles
		Data area	10000			Cycles
—	4-word program time	Program area		150	900	$\mu$ s
		Data area		300	1700	$\mu$ s
—	Lock bit program time	Program area		70	500	$\mu$ s
		Data area		140	1000	$\mu$ s
—	Block erasure time	4-Kbyte block		0.12	3.0	s
		32-Kbyte block		0.17	3.0	s
		64-Kbyte block		0.20	3.0	s
—	Data retention <sup>(2)</sup>	$T_a = 55^\circ\text{C}$ <sup>(3)</sup>	10			Years

Notes:

1. Program/erase definition  
This value represents the number of erasures per block.  
When the number of program/erase cycles is n, each block can be erased n times.  
For example, if a 4-word write is performed in 512 different addresses in the 4-Kbyte block A and then the block is erased, this is counted as a single program/erase operation.  
However, the same address cannot be written to more than once per erasure (overwrite disabled).
2. Data retention includes periods when no supply voltage is applied and no clock is provided.
3. Contact a Renesas Electronics sales office for data retention times other than the above condition.

**Table 26.9 Power Supply Circuit Timing Characteristics**  
 ( $V_{CC1} = V_{CC2} = 3.0$  to  $5.5$  V,  $V_{SS} = 0$  V, and  $T_a = T_{opr}$ , unless otherwise noted)

Symbol	Characteristic	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
$t_{d(P-R)}$	Internal power supply start-up stabilization time after the main power supply is turned on				2	ms



**Figure 26.3 Power Supply Circuit Timing**

**Table 26.10 Electrical Characteristics of Voltage Regulator for Internal Logic**  
 ( $V_{CC1} = V_{CC2} = 3.0$  to  $5.5$  V,  $V_{SS} = 0$  V, and  $T_a = T_{opr}$ , unless otherwise noted)

Symbol	Characteristics	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
$V_{VDC1}$	Output voltage			1.5		V

**Table 26.11 Electrical Characteristics of Low Voltage Detector**  
 ( $V_{CC1} = V_{CC2} = 4.2$  to  $5.5$  V,  $V_{SS} = 0$  V, and  $T_a = T_{opr}$ , unless otherwise noted)

Symbol	Characteristics	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
$\Delta V_{det}$	Detected voltage error				$\pm 0.3$	V
$V_{det(R)} - V_{det(F)}$	Hysteresis width		0			V
—	Self-consuming current	$V_{CC1} = 5.0$ V, low voltage detector enabled		4		$\mu A$
$t_{d(E-A)}$	Operation start time of low voltage detector				150	$\mu s$

**Table 26.12 Electrical Characteristics of Oscillator** $(V_{CC1} = V_{CC2} = 3.0 \text{ to } 5.5 \text{ V}, V_{SS} = 0 \text{ V}, \text{ and } T_a = T_{opr}, \text{ unless otherwise noted})$ 

Symbol	Characteristics	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
$f_{SO(PLL)}$	PLL clock self-oscillation frequency		35	55	80	MHz
$t_{LOCK(PLL)}$	PLL lock time (1)				1	ms
$t_{jitter(p-p)}$	PLL jitter period (p-p)				2.0	ns
$f_{(OCO)}$	On-chip oscillator frequency		62.5	125	250	kHz

Note:

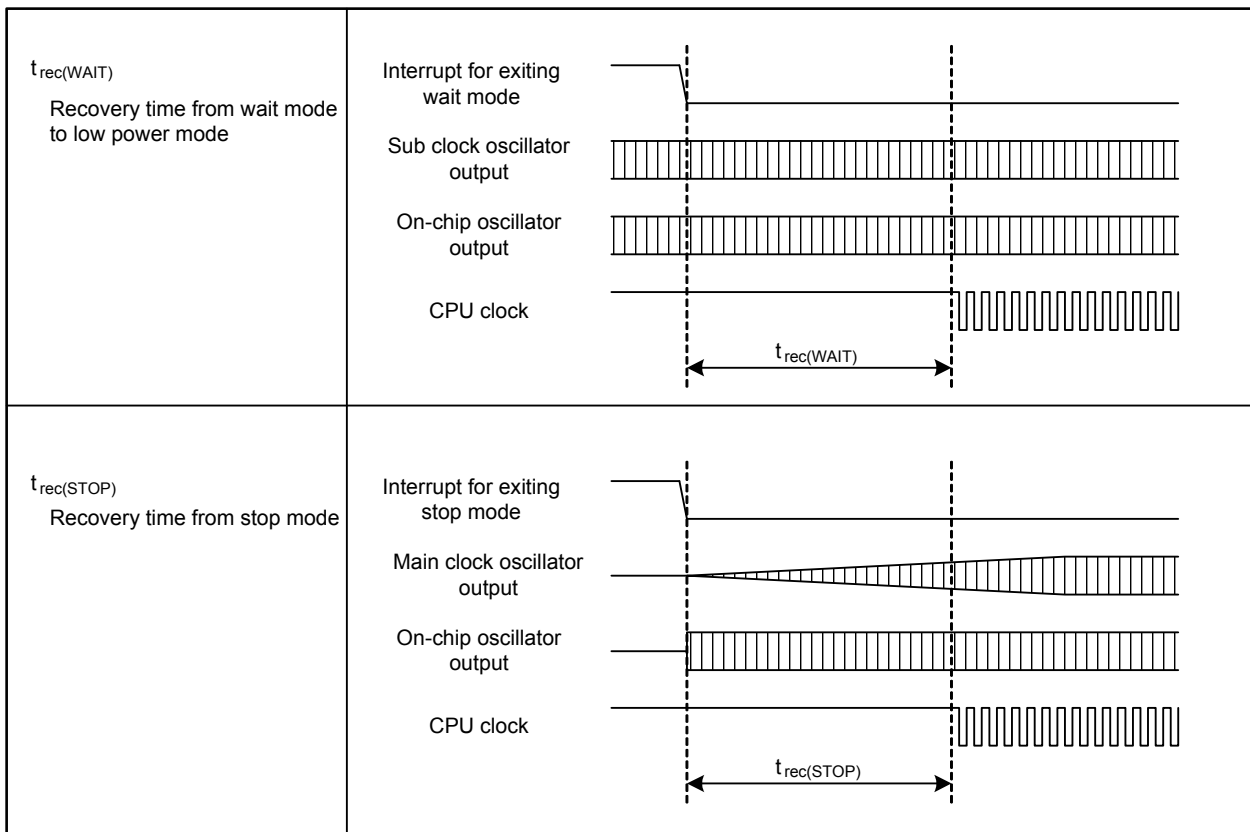
1. This value is applicable only when the main clock oscillation is stable.

**Table 26.13 Electrical Characteristics of Clock Circuitry** $(V_{CC1} = V_{CC2} = 3.0 \text{ to } 5.5 \text{ V}, V_{SS} = 0 \text{ V}, \text{ and } T_a = T_{opr}, \text{ unless otherwise noted})$ 

Symbol	Characteristics	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
$t_{rec(WAIT)}$	Recovery time from wait mode to low power mode				225	$\mu\text{s}$
$t_{rec(STOP)}$	Recovery time from stop mode (1)				225	$\mu\text{s}$

Note:

1. The recovery time from stop mode does not include the main clock oscillation stabilization time. The CPU starts operating before the oscillator is stabilized.

**Figure 26.4 Clock Circuit Timing**



Timing Requirements ( $V_{CC1} = V_{CC2} = 3.0$  to  $5.5$  V,  $V_{SS} = 0$  V, and  $T_a = T_{opr}$ , unless otherwise noted)

Table 26.14 Flash Memory CPU Rewrite Mode Timing

Symbol	Characteristics	Value		Unit
		Min.	Max.	
$t_{cR}$	Read cycle time	200		ns
$t_{su(S-R)}$	Chip-select setup time before read	200		ns
$t_{h(R-S)}$	Chip-select hold time after read	0		ns
$t_{su(A-R)}$	Address setup time before read	200		ns
$t_{h(R-A)}$	Address hold time after read	0		ns
$t_{w(R)}$	Read pulse width	100		ns
$t_{cW}$	Write cycle time	200		ns
$t_{su(S-W)}$	Chip-select setup time before write	0		ns
$t_{h(W-S)}$	Chip-select hold time after write	30		ns
$t_{su(A-W)}$	Address setup time before write	0		ns
$t_{h(W-A)}$	Address hold time after write	30		ns
$t_{w(W)}$	Write pulse width	50		ns

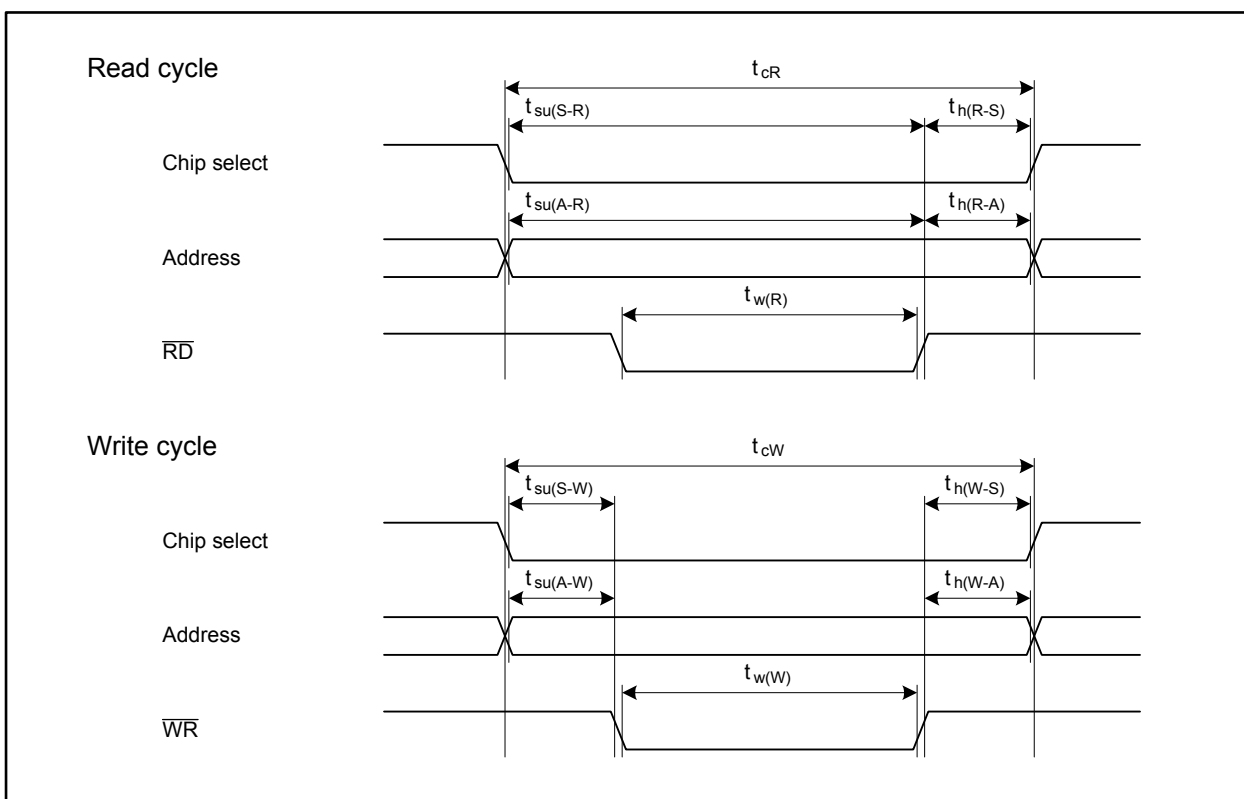


Figure 26.5 Flash Memory CPU Rewrite Mode Timing

$$V_{CC1} = V_{CC2} = 5 \text{ V}$$

**Table 26.15 Electrical Characteristics (1/3)**

( $V_{CC1} = V_{CC2} = 4.2$  to  $5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ ,  $T_a = T_{opr}$ , and  $f_{(CPU)} = 50 \text{ MHz}$ , unless otherwise noted)

Symbol	Characteristic		Measurement Condition	Value (2)			Unit
				Min.	Typ.	Max.	
$V_{OH}$	High level output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7 (1)	$I_{OH} = -5 \text{ mA}$	$V_{CC2} - 2.0$		$V_{CC2}$	V
		P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3 to P9_7, P10_0 to P10_7 (1)	$I_{OH} = -5 \text{ mA}$	$V_{CC1} - 2.0$		$V_{CC1}$	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7 (1)	$I_{OH} = -200 \mu\text{A}$	$V_{CC2} - 0.3$		$V_{CC2}$	V
		P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3 to P9_7, P10_0 to P10_7 (1)	$I_{OH} = -200 \mu\text{A}$	$V_{CC1} - 0.3$		$V_{CC1}$	V
$V_{OL}$	Low level output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3 to P9_7, P10_0 to P10_7 (1)	$I_{OL} = 5 \text{ mA}$			2.0	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3 to P9_7, P10_0 to P10_7 (1)	$I_{OL} = 200 \mu\text{A}$			0.45	V

## Notes:

- Ports P0\_4 to P0\_7, P1\_0 to P1\_4, P3\_4 to P3\_7, P4, P5, and P9\_4 to P9\_7 are available in the 100-pin package only.
- The  $V_{CC2}$  pin is available in the 100-pin package only. It should be considered as  $V_{CC1}$  in the 64-pin package.

$$V_{CC1} = V_{CC2} = 5 \text{ V}$$

**Table 26.16 Electrical Characteristics (2/3)**

( $V_{CC1} = V_{CC2} = 4.2$  to  $5.5$  V,  $V_{SS} = 0$  V,  $T_a = T_{opr}$ , and  $f_{(CPU)} = 50$  MHz, unless otherwise noted)

Symbol	Characteristic		Measurement Condition	Value			Unit
				Min.	Typ.	Max.	
$V_{T+} - V_{T-}$	Hysteresis	HOLD, RDY, NMI, INT0 to INT5, KI0 to KI3, TA0IN to TA4IN, TA0OUT to TA4OUT, TB0IN to TB5IN, CTS0 to CTS8, CLK0 to CLK8, RXD0 to RXD8, SCL0 to SCL6, SDA0 to SDA6, SS0 to SS6, SRXD0 to SRXD6, ADTRG, IIO0_0 to IIO0_7, IIO1_0 to IIO1_7, UD0A, UD0B, UD1A, UD1B, ISCLK2, ISRXD2, IEIN (1)		0.2		1.0	V
		RESET		0.2		1.8	V
$I_{IH}$	High level input current	XIN, RESET, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_1, P9_3 to P9_7, P10_0 to P10_7 (2)	$V_I = 5 \text{ V}$			5.0	$\mu\text{A}$
$I_{IL}$	Low level input current	XIN, RESET, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_1, P9_3 to P9_7, P10_0 to P10_7 (2)	$V_I = 0 \text{ V}$			-5.0	$\mu\text{A}$
$R_{PULLUP}$	Pull-up resistor	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_1, P9_3 to P9_7, P10_0 to P10_7 (2)	$V_I = 0 \text{ V}$	30	50	170	$\text{k}\Omega$
$R_{fXIN}$	Feedback resistor	XIN			1.5		$\text{M}\Omega$
$R_{fXCIN}$	Feedback resistor	XCIN			15		$\text{M}\Omega$

**Notes:**

1. Pins TB4IN, CTS4, CLK4, RXD4, SCL4, SDA4, SS4, SRXD4, and UART6, and UART7 are available in the 100-pin package only.
2. Ports P0\_4 to P0\_7, P1\_0 to P1\_4, P3\_4 to P3\_7, P4, P5, P9\_1, and P9\_4 to P9\_7 are available in the 100-pin package only.

$$V_{CC1} = V_{CC2} = 5 \text{ V}$$

**Table 26.17 Electrical Characteristics (3/3)**

( $V_{CC1} = V_{CC2} = 4.2$  to  $5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

Symbol	Characteristic	Measurement Condition	Value			Unit	
			Min.	Typ.	Max.		
$I_{CC}$	Power supply current	In single-chip mode, output pins are left open and others are connected to $V_{SS}$	$f_{(CPU)} = 50 \text{ MHz}$ , $f_{(BCLK)} = 25 \text{ MHz}$ , $f_{(XIN)} = 8 \text{ MHz}$ , Active: XIN, PLL, Stopped: XCIN, OCO		32	45	mA
			$f_{(CPU)} = f_{SO(PLL)}/24 \text{ MHz}$ , Active: PLL (self-oscillation), Stopped: XIN, XCIN, OCO		10		mA
			$f_{(CPU)} = f_{(BCLK)} = f_{(XIN)}/256 \text{ MHz}$ , $f_{(XIN)} = 8 \text{ MHz}$ , Active: XIN, Stopped: PLL, XCIN, OCO		1.2		mA
			$f_{(CPU)} = f_{(BCLK)} = 32.768 \text{ kHz}$ , Active: XCIN, Stopped: XIN, PLL, OCO, Main regulator: shutdown		220		$\mu\text{A}$
			$f_{(CPU)} = f_{(BCLK)} = f_{(OCO)}/4 \text{ kHz}$ , Active: OCO, Stopped: XIN, PLL, XCIN, Main regulator: shutdown		230		$\mu\text{A}$
			$f_{(CPU)} = f_{(BCLK)} = f_{(XIN)}/256 \text{ MHz}$ , $f_{(XIN)} = 8 \text{ MHz}$ , Active: XIN, Stopped: PLL, XCIN, OCO, $T_a = 25^\circ\text{C}$ , Wait mode		960	1600	$\mu\text{A}$
			$f_{(CPU)} = f_{(BCLK)} = 32.768 \text{ kHz}$ , Active: XCIN, Stopped: XIN, PLL, OCO, Main regulator: shutdown, $T_a = 25^\circ\text{C}$ , Wait mode		8	140	$\mu\text{A}$
			$f_{(CPU)} = f_{(BCLK)} = f_{(OCO)}/4 \text{ kHz}$ , Active: OCO, Stopped: XIN, PLL, XCIN, Main regulator: shutdown, $T_a = 25^\circ\text{C}$ , Wait mode		10	150	$\mu\text{A}$
			Stopped: all clocks, Main regulator: shutdown, $T_a = 25^\circ\text{C}$		5	70	$\mu\text{A}$

$$V_{CC1} = V_{CC2} = 5 \text{ V}$$

**Table 26.18 A/D Conversion Characteristics ( $V_{CC1} = V_{CC2} = AV_{CC} = V_{REF} = 4.2$  to  $5.5 \text{ V}$ ,  
 $V_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = T_{opr}$ , and  $f_{(BCLK)} = 25 \text{ MHz}$ , unless otherwise noted)**

Symbol	Characteristic	Measurement Condition	Value			Unit	
			Min.	Typ.	Max.		
—	Resolution	$V_{REF} = V_{CC1}$			10	Bits	
—	Absolute error	$V_{REF} = V_{CC1} = V_{CC2} = 5 \text{ V}$	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0, ANEX1 (1)			±3	LSB
			External op-amp connection mode			±7	LSB
INL	Integral non-linearity error	$V_{REF} = V_{CC1} = V_{CC2} = 5 \text{ V}$	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0, ANEX1 (1)			±3	LSB
			External op-amp connection mode			±7	LSB
DNL	Differential non-linearity error				±1	LSB	
—	Offset error				±3	LSB	
—	Gain error				±3	LSB	
$R_{LADDER}$	Resistor ladder	$V_{REF} = V_{CC1}$	4		20	kΩ	
$t_{CONV}$	Conversion time (10 bits)	$\phi_{AD} = 16 \text{ MHz}$ , with sample and hold function	2.06			μs	
		$\phi_{AD} = 16 \text{ MHz}$ , without sample and hold function	3.69			μs	
$t_{CONV}$	Conversion time (8 bits)	$\phi_{AD} = 16 \text{ MHz}$ , with sample and hold function	1.75			μs	
		$\phi_{AD} = 16 \text{ MHz}$ , without sample and hold function	3.06			μs	
$t_{SAMP}$	Sampling time	$\phi_{AD} = 16 \text{ MHz}$	0.188			μs	
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V	
$\phi_{AD}$	Operating clock frequency	Without sample and hold function	0.25		16	MHz	
		With sample and hold function	1		16	MHz	

Note:

1. Pins AN0\_4 to AN0\_7, ANEX0, and ANEX1 are available in the 100-pin package only.

$$V_{CC1} = V_{CC2} = 5 \text{ V}$$

**Table 26.19 D/A Conversion Characteristics ( $V_{CC1} = V_{CC2} = AV_{CC} = V_{REF} = 4.2$  to  $5.5 \text{ V}$ ,  
 $V_{SS} = AV_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)**

Symbol	Characteristic	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
—	Resolution				8	Bits
—	Absolute precision				1.0	%
$t_s$	Settling time				3	$\mu\text{s}$
$R_O$	Output resistance		4	10	20	$\text{k}\Omega$
$I_{VREF}$	Reference input current	See Note 1			1.5	mA

Note:

- One D/A converter is used. The DAi register ( $i = 0, 1$ ) of the other unused converter is set to 00h. The resistor ladder for the A/D converter is not considered.  
Even when the VCUT bit in the AD0CON1 register is set to 0 ( $V_{REF}$  disconnected),  $I_{VREF}$  is supplied.

$$V_{CC1} = V_{CC2} = 5 \text{ V}$$

Timing Requirements ( $V_{CC1} = V_{CC2} = 4.2$  to  $5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.20 External Clock Input**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{C(X)}$	External clock input period	62.5	250	ns
$t_{W(XH)}$	External clock input high level pulse width	25		ns
$t_{W(XL)}$	External clock input low level pulse width	25		ns
$t_{R(X)}$	External clock input rise time		5	ns
$t_{F(X)}$	External clock input fall time		5	ns
$t_W / t_C$	External clock input duty	40	60	%

**Table 26.21 External Bus Timing**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{SU(D-R)}$	Data setup time before read	40		ns
$t_{H(R-D)}$	Data hold time after read	0		ns
$t_{DIS(R-D)}$	Data disable time after read		$0.5 \times t_{C(\text{Base})} + 10$	ns

$$V_{CC1} = V_{CC2} = 5 \text{ V}$$

Timing Requirements ( $V_{CC1} = V_{CC2} = 4.2$  to  $5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.22 Timer A Input (counting input in event counter mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{C(TA)}$	TAiIN input clock cycle time	200		ns
$t_{W(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{W(TAL)}$	TAiIN input low level pulse width	80		ns

**Table 26.23 Timer A Input (gating input in timer mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{C(TA)}$	TAiIN input clock cycle time	400		ns
$t_{W(TAH)}$	TAiIN input high level pulse width	180		ns
$t_{W(TAL)}$	TAiIN input low level pulse width	180		ns

**Table 26.24 Timer A Input (external trigger input in one-shot timer mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{C(TA)}$	TAiIN input clock cycle time	200		ns
$t_{W(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{W(TAL)}$	TAiIN input low level pulse width	80		ns

**Table 26.25 Timer A Input (external trigger input in pulse-width modulation mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{W(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{W(TAL)}$	TAiIN input low level pulse width	80		ns

**Table 26.26 Timer A Input (increment/decrement switching input in event counter mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{C(UP)}$	TAiOUT input clock cycle time	2000		ns
$t_{W(UPH)}$	TAiOUT input high level pulse width	1000		ns
$t_{W(UPL)}$	TAiOUT input low level pulse width	1000		ns
$t_{Su(UP-TIN)}$	TAiOUT input setup time	400		ns
$t_h(TIN-UP)$	TAiOUT input hold time	400		ns



$$V_{CC1} = V_{CC2} = 5 \text{ V}$$

Timing Requirements ( $V_{CC1} = V_{CC2} = 4.2$  to  $5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.27 Timer B Input (counting input in event counter mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time (one edge counting)	200		ns
$t_{W(TBH)}$	TBiIN input high level pulse width (one edge counting)	80		ns
$t_{W(TBL)}$	TBiIN input low level pulse width (one edge counting)	80		ns
$t_{c(TB)}$	TBiIN input clock cycle time (both edges counting)	200		ns
$t_{W(TBH)}$	TBiIN input high level pulse width (both edges counting)	80		ns
$t_{W(TBL)}$	TBiIN input low level pulse width (both edges counting)	80		ns

**Table 26.28 Timer B Input (pulse period measure mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time	400		ns
$t_{W(TBH)}$	TBiIN input high level pulse width	180		ns
$t_{W(TBL)}$	TBiIN input low level pulse width	180		ns

**Table 26.29 Timer B Input (pulse-width measure mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time	400		ns
$t_{W(TBH)}$	TBiIN input high level pulse width	180		ns
$t_{W(TBL)}$	TBiIN input low level pulse width	180		ns

$$V_{CC1} = V_{CC2} = 5 \text{ V}$$

Timing Requirements ( $V_{CC1} = V_{CC2} = 4.2$  to  $5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.30 Serial Interface**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input clock cycle time	200		ns
$t_{W(CKH)}$	CLKi input high level pulse width	80		ns
$t_{W(CKL)}$	CLKi input low level pulse width	80		ns
$t_{su(D-C)}$	RXD <sub>i</sub> input setup time	80		ns
$t_{h(C-D)}$	RXD <sub>i</sub> input hold time	90		ns

**Table 26.31 A/D Trigger Input**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{W(ADH)}$	ADTRG input high level pulse width Hardware trigger input high level pulse width	$\frac{3}{\phi_{AD}}$		ns
$t_{W(ADL)}$	ADTRG input low level pulse width Hardware trigger input high level pulse width	125		ns

**Table 26.32 External Interrupt  $\overline{INT}_i$  Input**

Symbol	Characteristic		Value		Unit
			Min.	Max.	
$t_{W(INH)}$	$\overline{INT}_i$ input high level pulse width	Edge sensitive	250		ns
		Level sensitive	$t_{c(CPU)} + 200$		ns
$t_{W(INL)}$	$\overline{INT}_i$ input low level pulse width	Edge sensitive	250		ns
		Level sensitive	$t_{c(CPU)} + 200$		ns

**Table 26.33 Intelligent I/O**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(ISCLK2)}$	ISCLK2 input clock cycle time	600		ns
$t_{W(ISCLK2H)}$	ISCLK2 input high level pulse width	270		ns
$t_{W(ISCLK2L)}$	ISCLK2 input low level pulse width	270		ns
$t_{su(RXD-ISCLK2)}$	ISRXD2 input setup time	150		ns
$t_{h(ISCLK2-RXD)}$	ISRXD2 input hold time	100		ns

$$V_{CC1} = V_{CC2} = 5 \text{ V}$$

Switching Characteristics ( $V_{CC1} = V_{CC2} = 4.2 \text{ to } 5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.34 External Bus Timing (separate bus)**

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{su(S-R)}$	Chip-select setup time before read	Refer to Figure 26.6	(1)		ns
$t_{h(R-S)}$	Chip-select hold time after read		$t_{c(Base)} - 10$		ns
$t_{su(A-R)}$	Address setup time before read		(1)		ns
$t_{h(R-A)}$	Address hold time after read		$t_{c(Base)} - 10$		ns
$t_{w(R)}$	Read pulse width		(1)		ns
$t_{su(S-W)}$	Chip-select setup time before write		(1)		ns
$t_{h(W-S)}$	Chip-select hold time after write		$1.5 \times t_{c(Base)} - 10$		ns
$t_{su(A-W)}$	Address setup time before write		(1)		ns
$t_{h(W-A)}$	Address hold time after write		$1.5 \times t_{c(Base)} - 10$		ns
$t_{w(W)}$	Write pulse width		(1)		ns
$t_{su(D-W)}$	Data setup time before write		(1)		ns
$t_{h(W-D)}$	Data hold time after write		0		ns

Note:

- The value is calculated using the formulas below based on the base clock cycles ( $t_{c(Base)}$ ) and respective cycles of  $T_{su(A-R)}$ ,  $T_w(R)$ ,  $T_{su(A-W)}$ , and  $T_w(W)$  set by registers EBC0 to EBC3. If the calculation results in a negative value, modify the value to be set. For details on how to set values, refer to 9.3.5 "External Bus Timing".

$$t_{su(S-R)} = t_{su(A-R)} = T_{su(A-R)} \times t_{c(Base)} - 15 \text{ [ns]}$$

$$t_{w(R)} = T_w(R) \times t_{c(Base)} - 10 \text{ [ns]}$$

$$t_{su(S-W)} = t_{su(A-W)} = T_{su(A-W)} \times t_{c(Base)} - 15 \text{ [ns]}$$

$$t_{w(W)} = t_{su(D-W)} = T_w(W) \times t_{c(Base)} - 10 \text{ [ns]}$$

$$V_{CC1} = V_{CC2} = 5 \text{ V}$$

Switching Characteristics ( $V_{CC1} = V_{CC2} = 4.2$  to  $5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.35 External Bus Timing (multiplexed bus)**

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{su(S-ALE)}$	Chip-select setup time before ALE	Refer to Figure 26.6	(1)		ns
$t_{h(R-S)}$	Chip-select hold time after read		$1.5 \times t_{c(Base)} - 10$		ns
$t_{su(A-ALE)}$	Address setup time before ALE		(1)		ns
$t_{h(ALE-A)}$	Address hold time after ALE		$0.5 \times t_{c(Base)} - 5$		ns
$t_{h(R-A)}$	Address hold time after read		$1.5 \times t_{c(Base)} - 10$		ns
$t_{d(ALE-R)}$	ALE-read delay time		$0.5 \times t_{c(Base)} - 5$	$0.5 \times t_{c(Base)} + 10$	ns
$t_{w(ALE)}$	ALE pulse width		(1)		ns
$t_{dis(R-A)}$	Address disable time after read			8	ns
$t_{w(R)}$	Read pulse width		(1)		ns
$t_{h(W-S)}$	Chip-select hold time after write		$1.5 \times t_{c(Base)} - 10$		ns
$t_{h(W-A)}$	Address hold time after write		$1.5 \times t_{c(Base)} - 10$		ns
$t_{d(ALE-W)}$	ALE-write delay time		$0.5 \times t_{c(Base)} - 5$	$0.5 \times t_{c(Base)} + 10$	ns
$t_{w(W)}$	Write pulse width		(1)		ns
$t_{su(D-W)}$	Data setup time before write		(1)		ns
$t_{h(W-D)}$	Data hold time after write		$0.5 \times t_{c(Base)}$		ns

Note:

- The value is calculated using the formulas below based on the base clock cycles ( $t_{c(Base)}$ ) and respective cycles of  $T_{su(A-R)}$ ,  $T_{w(R)}$ ,  $T_{su(A-W)}$ , and  $T_{w(W)}$  set by registers EBC0 to EBC3. If the calculation results in a negative value, modify the value to be set. For details on how to set values, refer to 9.3.5 "External Bus Timing".

$$t_{su(S-ALE)} = t_{su(A-ALE)} = t_{w(ALE)} = (T_{su(A-R)} - 0.5) \times t_{c(Base)} - 15 \text{ [ns]}$$

$$t_{w(R)} = T_{w(R)} \times t_{c(Base)} - 10 \text{ [ns]}$$

$$t_{w(W)} = t_{su(D-W)} = T_{w(W)} \times t_{c(Base)} - 10 \text{ [ns]}$$

$$V_{CC1} = V_{CC2} = 5 \text{ V}$$

Switching Characteristics ( $V_{CC1} = V_{CC2} = 4.2$  to  $5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.36 Serial Interface**

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{d(C-Q)}$	TXDi output delay time	Refer to Figure 26.6		80	ns
$t_{h(C-Q)}$	TXDi output hold time		0		ns

**Table 26.37 Intelligent I/O**

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{d(ISCLK2-TXD)}$	ISTXD2 output delay time	Refer to Figure 26.6		180	ns
$t_{h(ISCLK2-RXD)}$	ISTXD2 output hold time		0		ns

$$V_{CC1} = V_{CC2} = 3.3 \text{ V}$$

**Table 26.38 Electrical Characteristics (1/3) ( $V_{CC1} = V_{CC2} = 3.0$  to  $3.6$  V,  $V_{SS} = 0$  V,  $T_a = T_{opr}$ , and  $f_{(CPU)} = 50$  MHz, unless otherwise noted)**

Symbol	Characteristic		Measurement Condition	Value (2)			Unit
				Min.	Typ.	Max.	
V <sub>OH</sub>	High level output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7 (1)	I <sub>OH</sub> = -1 mA	V <sub>CC2</sub> - 0.6		V <sub>CC2</sub>	V
		P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3 to P9_7, P10_0 to P10_7 (1)	I <sub>OH</sub> = -1 mA	V <sub>CC1</sub> - 0.6		V <sub>CC1</sub>	V
V <sub>OL</sub>	Low level output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_3 to P9_7, P10_0 to P10_7 (1)	I <sub>OL</sub> = 1 mA			0.5	V

## Notes:

- Ports P0\_4 to P0\_7, P1\_0 to P1\_4, P3\_4 to P3\_7, P4, P5, and P9\_4 to P9\_7 are available in the 100-pin package only.
- The V<sub>CC2</sub> pin is available in the 100-pin package only. It should be considered as V<sub>CC1</sub> in the 64-pin package.

$$V_{CC1} = V_{CC2} = 3.3 \text{ V}$$

**Table 26.39 Electrical Characteristics (2/3) ( $V_{CC1} = V_{CC2} = 3.0$  to  $3.6 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ ,  $T_a = T_{opr}$ , and  $f_{(CPU)} = 50 \text{ MHz}$ , unless otherwise noted)**

Symbol	Characteristic	Measurement Condition	Value			Unit	
			Min.	Typ.	Max.		
$V_{T+} - V_{T-}$	Hysteresis	HOLD, RDY, NMI, INT0 to INT5, KI0 to KI3, TA0IN to TA4IN, TA0OUT to TA4OUT, TB0IN to TB5IN, CTS0 to CTS8, CLK0 to CLK8, RXD0 to RXD8, SCL0 to SCL6, SDA0 to SDA6, SS0 to SS6, SRXD0 to SRXD6, ADTRG, IIO0_0 to IIO0_7, IIO1_0 to IIO1_7, UD0A, UD0B, UD1A, UD1B, ISCLK2, ISRXD2, IEIN (1)	0.2		1.0	V	
							RESET
$I_{IH}$	High level input current	XIN, RESET, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_1, P9_3 to P9_7, P10_0 to P10_7 (2)	$V_I = 3.3 \text{ V}$			4.0	$\mu\text{A}$
$I_{IL}$	Low level input current	XIN, RESET, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_1, P9_3 to P9_7, P10_0 to P10_7 (2)	$V_I = 0 \text{ V}$			-4.0	$\mu\text{A}$
$R_{PULLUP}$	Pull-up resistor	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_1, P9_3 to P9_7, P10_0 to P10_7 (2)	$V_I = 0 \text{ V}$	50	100	500	$\text{k}\Omega$
$R_{fXIN}$	Feedback resistor	XIN			3		$\text{M}\Omega$
$R_{fXCIN}$	Feedback resistor	XCIN			25		$\text{M}\Omega$

## Notes:

1. Pins TB4IN, CTS4, CLK4, RXD4, SCL4, SDA4, SS4, SRXD4, and UART6, and UART7 are available in the 100-pin package only.
2. Ports P0\_4 to P0\_7, P1\_0 to P1\_4, P3\_4 to P3\_7, P4, P5, P9\_1, and P9\_4 to P9\_7 are available in the 100-pin package only.

$$V_{CC1} = V_{CC2} = 3.3 \text{ V}$$

**Table 26.40 Electrical Characteristics (3/3)**

( $V_{CC1} = V_{CC2} = 3.0$  to  $3.6 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

Symbol	Characteristic	Measurement Condition	Value			Unit	
			Min.	Typ.	Max.		
$I_{CC}$	Power supply current	In single-chip mode, output pins are left open and others are connected to $V_{SS}$	$f_{(CPU)} = 50 \text{ MHz}$ , $f_{(BCLK)} = 25 \text{ MHz}$ , $f_{(XIN)} = 8 \text{ MHz}$ , Active: XIN, PLL, Stopped: XCIN, OCO		28	40	mA
		XIN-XOUT Drive strength: low	$f_{(CPU)} = f_{SO(PLL)}/24 \text{ MHz}$ , Active: PLL (self-oscillation), Stopped: XIN, XCIN, OCO		7		mA
		XCIN-XCOUT Drive strength: low	$f_{(CPU)} = f_{(BCLK)} = f_{(XIN)}/256 \text{ MHz}$ , $f_{(XIN)} = 8 \text{ MHz}$ , Active: XIN, Stopped: PLL, XCIN, OCO		670		$\mu\text{A}$
			$f_{(CPU)} = f_{(BCLK)} = 32.768 \text{ kHz}$ , Active: XCIN, Stopped: XIN, PLL, OCO, Main regulator: shutdown		180		$\mu\text{A}$
			$f_{(CPU)} = f_{(BCLK)} = f_{(OCO)}/4 \text{ kHz}$ , Active: OCO, Stopped: XIN, PLL, XCIN, Main regulator: shutdown		190		$\mu\text{A}$
			$f_{(CPU)} = f_{(BCLK)} = f_{(XIN)}/256 \text{ MHz}$ , $f_{(XIN)} = 8 \text{ MHz}$ , Active: XIN, Stopped: PLL, XCIN, OCO, $T_a = 25^\circ\text{C}$ , Wait mode		500	900	$\mu\text{A}$
			$f_{(CPU)} = f_{(BCLK)} = 32.768 \text{ kHz}$ , Active: XCIN, Stopped: XIN, PLL, OCO, Main regulator: shutdown, $T_a = 25^\circ\text{C}$ , Wait mode		8	140	$\mu\text{A}$
			$f_{(CPU)} = f_{(BCLK)} = f_{(OCO)}/4 \text{ kHz}$ , Active: OCO, Stopped: XIN, PLL, XCIN, Main regulator: shutdown, $T_a = 25^\circ\text{C}$ , Wait mode		10	150	$\mu\text{A}$
			Stopped: all clocks, Main regulator: shutdown, $T_a = 25^\circ\text{C}$		5	70	$\mu\text{A}$



$$V_{CC1} = V_{CC2} = 3.3 \text{ V}$$

**Table 26.41 A/D Conversion Characteristics ( $V_{CC1} = V_{CC2} = AV_{CC} = V_{REF} = 3.0$  to  $3.6 \text{ V}$ ,  
 $V_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = T_{opr}$ , and  $f_{(BCLK)} = 25 \text{ MHz}$ , unless otherwise noted)**

Symbol	Characteristic	Measurement Condition	Value			Unit	
			Min.	Typ.	Max.		
—	Resolution	$V_{REF} = V_{CC1}$			10	Bits	
—	Absolute error	$V_{REF} = V_{CC1} = V_{CC2} = 3.3 \text{ V}$	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0, ANEX1 (1)			±5	LSB
			External op-amp connection mode			±7	LSB
INL	Integral non-linearity error	$V_{REF} = V_{CC1} = V_{CC2} = 3.3 \text{ V}$	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0, ANEX1 (1)			±5	LSB
			External op-amp connection mode			±7	LSB
DNL	Differential non- linearity error	$V_{REF} = V_{CC1} = V_{CC2} = 3.3 \text{ V}$			±1	LSB	
—	Offset error				±3	LSB	
—	Gain error				±3	LSB	
$R_{LADDER}$	Resistor ladder	$V_{REF} = V_{CC1}$	4		20	k $\Omega$	
$t_{CONV}$	Conversion time (10 bits)	$\phi_{AD} = 10 \text{ MHz}$ , with sample and hold function	3.3			$\mu\text{s}$	
$t_{CONV}$	Conversion time (8 bits)	$\phi_{AD} = 10 \text{ MHz}$ , with sample and hold function	2.8			$\mu\text{s}$	
$t_{SAMP}$	Sampling time	$\phi_{AD} = 10 \text{ MHz}$	0.3			$\mu\text{s}$	
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V	
$\phi_{AD}$	Operating clock frequency	Without sample and hold function	0.25		10	MHz	
		With sample and hold function	1		10	MHz	

Note:

1. Pins AN0\_4 to AN0\_7, ANEX0, and ANEX1 are available in the 100-pin package only.

$$V_{CC1} = V_{CC2} = 3.3 \text{ V}$$

**Table 26.42 D/A Conversion Characteristics ( $V_{CC1} = V_{CC2} = AV_{CC} = V_{REF} = 3.0$  to  $3.6 \text{ V}$ ,  
 $V_{SS} = AV_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)**

Symbol	Characteristic	Measurement Condition	Value			Unit
			Min.	Typ.	Max.	
—	Resolution				8	Bits
—	Absolute precision				1.0	%
$t_s$	Settling time				3	$\mu\text{s}$
$R_O$	Output resistance		4	10	20	$\text{k}\Omega$
$I_{VREF}$	Reference input current	See Note 1			1.0	mA

Note:

- One D/A converter is used. The DAi register ( $i = 0, 1$ ) of the other unused converter is set to 00h. The resistor ladder for the A/D converter is not considered.  
Even when the VCUT bit in the AD0CON1 register is set to 0 ( $V_{REF}$  disconnected),  $I_{VREF}$  is supplied.

$$V_{CC1} = V_{CC2} = 3.3 \text{ V}$$

Timing Requirements ( $V_{CC1} = V_{CC2} = 3.0$  to  $3.6 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.43 External Clock Input**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{C(X)}$	External clock input period	62.5	250	ns
$t_{W(XH)}$	External clock input high level pulse width	25		ns
$t_{W(XL)}$	External clock input low level pulse width	25		ns
$t_{r(X)}$	External clock input rise time		5	ns
$t_{f(X)}$	External clock input fall time		5	ns
$t_W / t_C$	External clock input duty	40	60	%

**Table 26.44 External Bus Timing**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{su(D-R)}$	Data setup time before read	40		ns
$t_{h(R-D)}$	Data hold time after read	0		ns
$t_{dis(R-D)}$	Data disable time after read		$0.5 \times t_{C(Base)} + 10$	ns

$$V_{CC1} = V_{CC2} = 3.3 \text{ V}$$

Timing Requirements ( $V_{CC1} = V_{CC2} = 3.0$  to  $3.6 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.45 Timer A Input (counting input in event counter mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input clock cycle time	200		ns
$t_{w(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	80		ns

**Table 26.46 Timer A Input (gating input in timer mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input clock cycle time	400		ns
$t_{w(TAH)}$	TAiIN input high level pulse width	180		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	180		ns

**Table 26.47 Timer A Input (external trigger input in one-shot timer mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input clock cycle time	200		ns
$t_{w(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	80		ns

**Table 26.48 Timer A Input (external trigger input in pulse-width modulation mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	80		ns

**Table 26.49 Timer A Input (increment/decrement switching input in event counter mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input clock cycle time	2000		ns
$t_{w(UPH)}$	TAiOUT input high level pulse width	1000		ns
$t_{w(UPL)}$	TAiOUT input low level pulse width	1000		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	400		ns
$t_h(TIN-UP)$	TAiOUT input hold time	400		ns

$$V_{CC1} = V_{CC2} = 3.3 \text{ V}$$

Timing Requirements ( $V_{CC1} = V_{CC2} = 3.0$  to  $3.6 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.50 Timer B Input (counting input in event counter mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time (one edge counting)	200		ns
$t_{w(TBH)}$	TBiIN input high level pulse width (one edge counting)	80		ns
$t_{w(TBL)}$	TBiIN input low level pulse width (one edge counting)	80		ns
$t_{c(TB)}$	TBiIN input clock cycle time (both edges counting)	200		ns
$t_{w(TBH)}$	TBiIN input high level pulse width (both edges counting)	80		ns
$t_{w(TBL)}$	TBiIN input low level pulse width (both edges counting)	80		ns

**Table 26.51 Timer B Input (pulse period measure mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time	400		ns
$t_{w(TBH)}$	TBiIN input high level pulse width	180		ns
$t_{w(TBL)}$	TBiIN input low level pulse width	180		ns

**Table 26.52 Timer B Input (pulse-width measure mode)**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time	400		ns
$t_{w(TBH)}$	TBiIN input high level pulse width	180		ns
$t_{w(TBL)}$	TBiIN input low level pulse width	180		ns

$$V_{CC1} = V_{CC2} = 3.3 \text{ V}$$

Timing Requirements ( $V_{CC1} = V_{CC2} = 3.0$  to  $3.6 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.53 Serial Interface**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input clock cycle time	200		ns
$t_{w(CKH)}$	CLKi input high level pulse width	80		ns
$t_{w(CKL)}$	CLKi input low level pulse width	80		ns
$t_{su(D-C)}$	RXDi input setup time	80		ns
$t_{h(C-D)}$	RXDi input hold time	90		ns

**Table 26.54 A/D Trigger Input**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{w(ADH)}$	ADTRG input high level pulse width Hardware trigger input high level pulse width	$\frac{3}{\phi_{AD}}$		ns
$t_{w(ADL)}$	ADTRG input low level pulse width Hardware trigger input high level pulse width	125		ns

**Table 26.55 External Interrupt  $\overline{INTi}$  Input**

Symbol	Characteristic		Value		Unit
			Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input high level pulse width	Edge sensitive	250		ns
		Level sensitive	$t_{c(CPU)} + 200$		ns
$t_{w(INL)}$	$\overline{INTi}$ input low level pulse width	Edge sensitive	250		ns
		Level sensitive	$t_{c(CPU)} + 200$		ns

**Table 26.56 Intelligent I/O**

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(ISCLK2)}$	ISCLK2 input clock cycle time	600		ns
$t_{w(ISCLK2H)}$	ISCLK2 input high level pulse width	270		ns
$t_{w(ISCLK2L)}$	ISCLK2 input low level pulse width	270		ns
$t_{su(RXD-ISCLK2)}$	ISRXD2 input setup time	150		ns
$t_{h(ISCLK2-RXD)}$	ISRXD2 input hold time	100		ns

$$V_{CC1} = V_{CC2} = 3.3 \text{ V}$$

Switching Characteristics ( $V_{CC1} = V_{CC2} = 3.0$  to  $3.6 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.57 External Bus Timing (separate bus)**

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{su(S-R)}$	Chip-select setup time before read	Refer to Figure 26.6	(1)		ns
$t_{h(R-S)}$	Chip-select hold time after read		$t_{c(Base)} - 10$		ns
$t_{su(A-R)}$	Address setup time before read		(1)		ns
$t_{h(R-A)}$	Address hold time after read		$t_{c(Base)} - 10$		ns
$t_{w(R)}$	Read pulse width		(1)		ns
$t_{su(S-W)}$	Chip-select setup time before write		(1)		ns
$t_{h(W-S)}$	Chip-select hold time after write		$1.5 \times t_{c(Base)} - 10$		ns
$t_{su(A-W)}$	Address setup time before write		(1)		ns
$t_{h(W-A)}$	Address hold time after write		$1.5 \times t_{c(Base)} - 10$		ns
$t_{w(W)}$	Write pulse width		(1)		ns
$t_{su(D-W)}$	Data setup time before write		(1)		ns
$t_{h(W-D)}$	Data hold time after write		0		ns

Note:

1. The value is calculated using the formulas below based on the base clock cycles ( $t_{c(Base)}$ ) and respective cycles of  $T_{su(A-R)}$ ,  $T_{w(R)}$ ,  $T_{su(A-W)}$ , and  $T_{w(W)}$  set by registers EBC0 to EBC3. If the calculation results in a negative value, modify the value to be set. For details on how to set values, refer to 9.3.5 "External Bus Timing".

$$t_{su(S-R)} = t_{su(A-R)} = T_{su(A-R)} \times t_{c(Base)} - 15 \text{ [ns]}$$

$$t_{w(R)} = T_{w(R)} \times t_{c(Base)} - 10 \text{ [ns]}$$

$$t_{su(S-W)} = t_{su(A-W)} = T_{su(A-W)} \times t_{c(Base)} - 15 \text{ [ns]}$$

$$t_{w(W)} = t_{su(D-W)} = T_{w(W)} \times t_{c(Base)} - 10 \text{ [ns]}$$

$$V_{CC1} = V_{CC2} = 3.3 \text{ V}$$

Switching Characteristics ( $V_{CC1} = V_{CC2} = 3.0$  to  $3.6 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.58 External Bus Timing (multiplexed bus)**

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{su(S-ALE)}$	Chip-select setup time before ALE	Refer to Figure 26.6	(1)		ns
$t_{h(R-S)}$	Chip-select hold time after read		$1.5 \times t_{c(Base)} - 10$		ns
$t_{su(A-ALE)}$	Address setup time before ALE		(1)		ns
$t_{h(ALE-A)}$	Address hold time after ALE		$0.5 \times t_{c(Base)} - 5$		ns
$t_{h(R-A)}$	Address hold time after read		$1.5 \times t_{c(Base)} - 10$		ns
$t_{d(ALE-R)}$	ALE-read delay time		$0.5 \times t_{c(Base)} - 5$	$0.5 \times t_{c(Base)} + 10$	ns
$t_{w(ALE)}$	ALE pulse width		(1)		ns
$t_{dis(R-A)}$	Address disable time after read			8	ns
$t_{w(R)}$	Read pulse width		(1)		ns
$t_{h(W-S)}$	Chip-select hold time after write		$1.5 \times t_{c(Base)} - 10$		ns
$t_{h(W-A)}$	Address hold time after write		$1.5 \times t_{c(Base)} - 10$		ns
$t_{d(ALE-W)}$	ALE-write delay time		$0.5 \times t_{c(Base)} - 5$	$0.5 \times t_{c(Base)} + 10$	ns
$t_{w(W)}$	Write pulse width		(1)		ns
$t_{su(D-W)}$	Data setup time before write		(1)		ns
$t_{h(W-D)}$	Data hold time after write		$0.5 \times t_{c(Base)}$		ns

Note:

- The value is calculated using the formulas below based on the base clock cycles ( $t_{c(Base)}$ ) and respective cycles of  $T_{su(A-R)}$ ,  $T_{w(R)}$ ,  $T_{su(A-W)}$ , and  $T_{w(W)}$  set by registers EBC0 to EBC3. If the calculation results in a negative value, modify the value to be set. For details on how to set values, refer to 9.3.5 "External Bus Timing".

$$t_{su(S-ALE)} = t_{su(A-ALE)} = t_{w(ALE)} = (T_{su(A-R)} - 0.5) \times t_{c(Base)} - 15 \text{ [ns]}$$

$$t_{w(R)} = T_{w(R)} \times t_{c(Base)} - 10 \text{ [ns]}$$

$$t_{w(W)} = t_{su(D-W)} = T_{w(W)} \times t_{c(Base)} - 10 \text{ [ns]}$$



$$V_{CC1} = V_{CC2} = 3.3 \text{ V}$$

Switching Characteristics ( $V_{CC1} = V_{CC2} = 3.0$  to  $3.6 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ , and  $T_a = T_{opr}$ , unless otherwise noted)

**Table 26.59 Serial Interface**

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{d(C-Q)}$	TXDi output delay time	Refer to Figure 26.6		80	ns
$t_{h(C-Q)}$	TXDi output hold time		0		ns

**Table 26.60 Intelligent I/O**

Symbol	Characteristic	Measurement Condition	Value		Unit
			Min.	Max.	
$t_{d(ISCLK2-TXD)}$	ISTXD2 output delay time	Refer to Figure 26.6		180	ns
$t_{h(ISCLK2-RXD)}$	ISTXD2 output hold time		0		ns

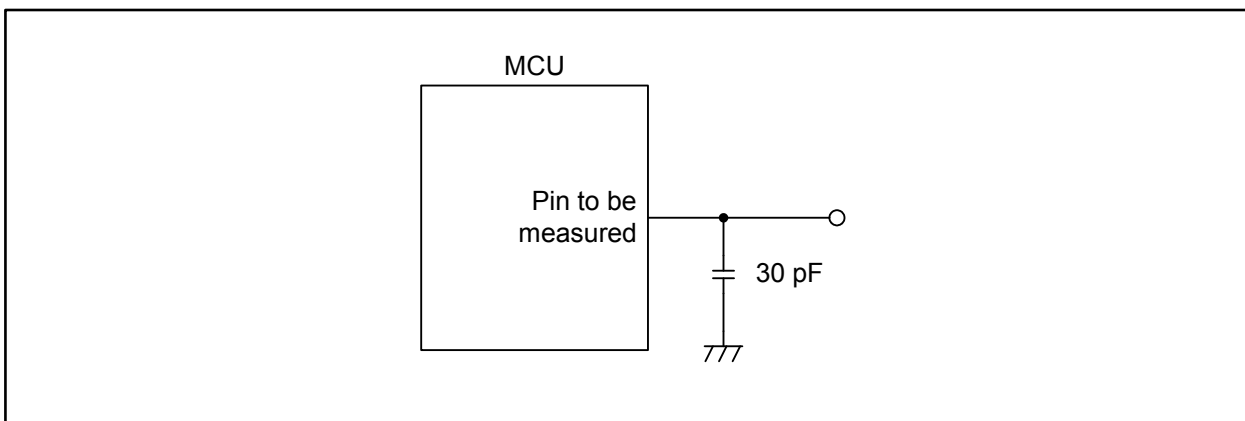


Figure 26.6 Switching Characteristic Measurement Circuit

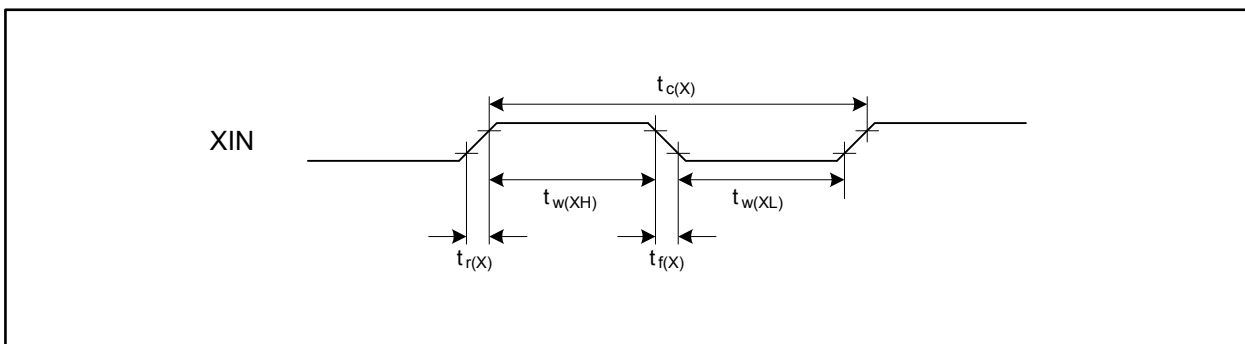


Figure 26.7 External Clock Input Timing

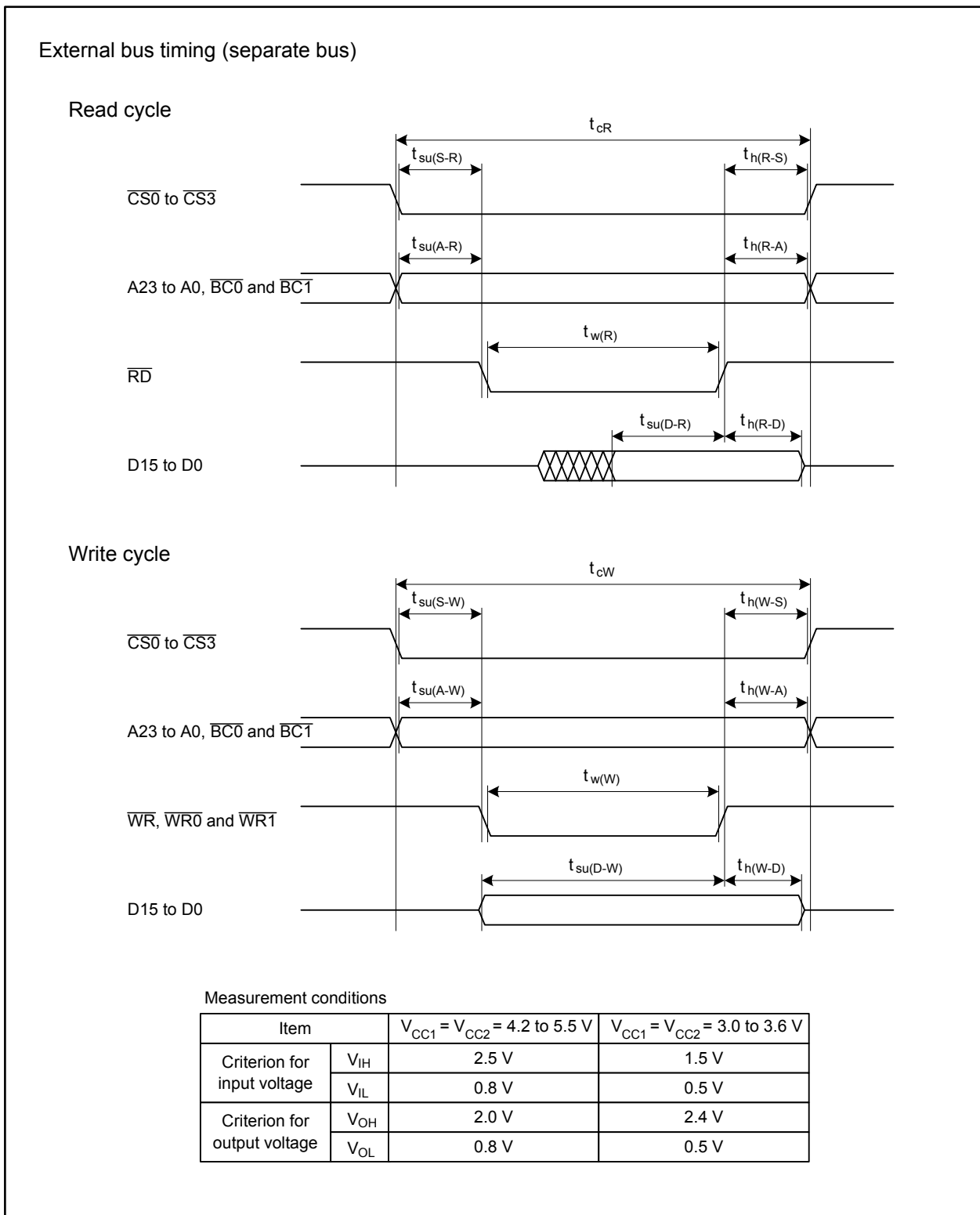


Figure 26.8 External Bus Timing for Separate Bus

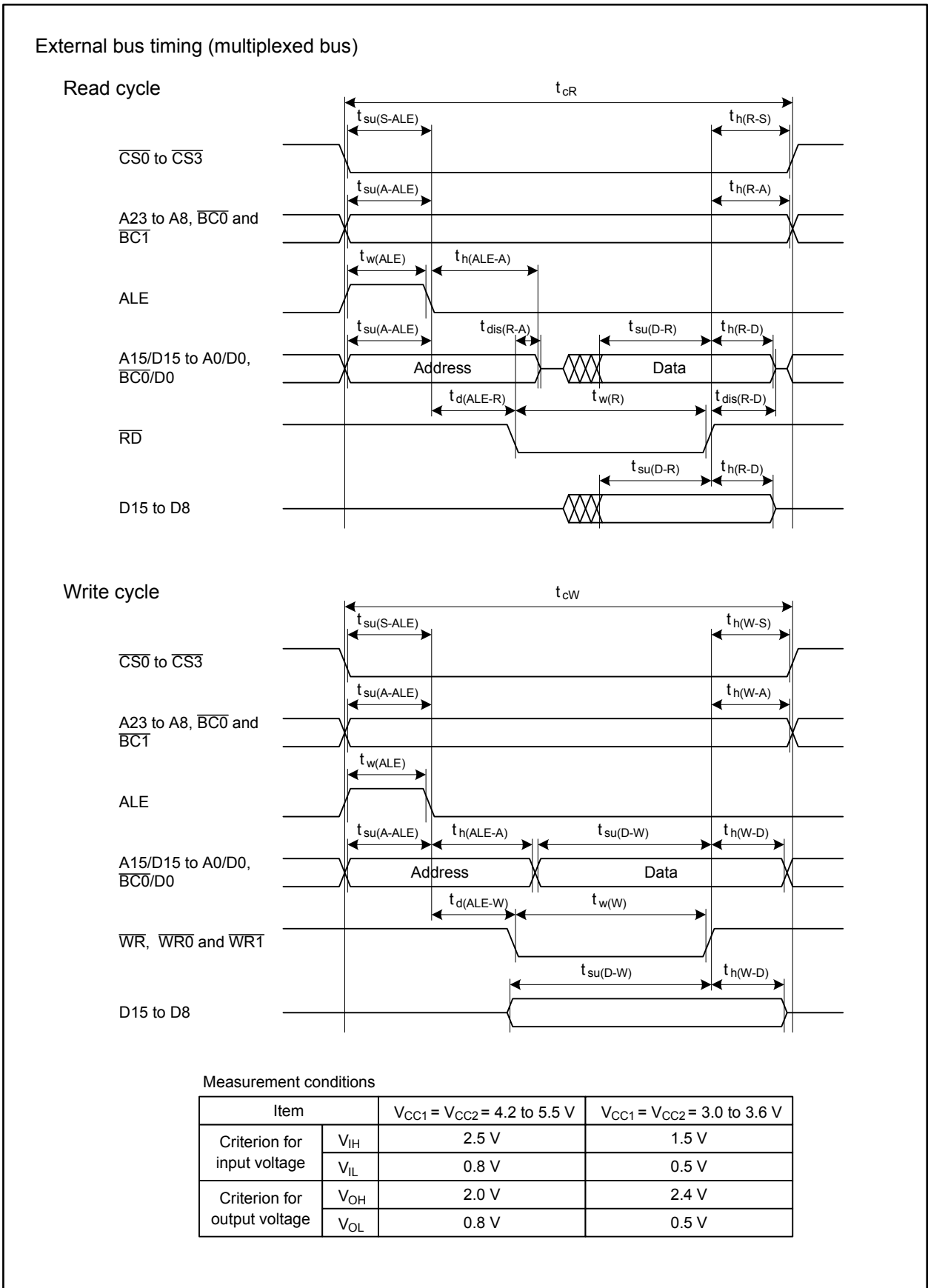


Figure 26.9 External Bus Timing for Multiplexed Bus

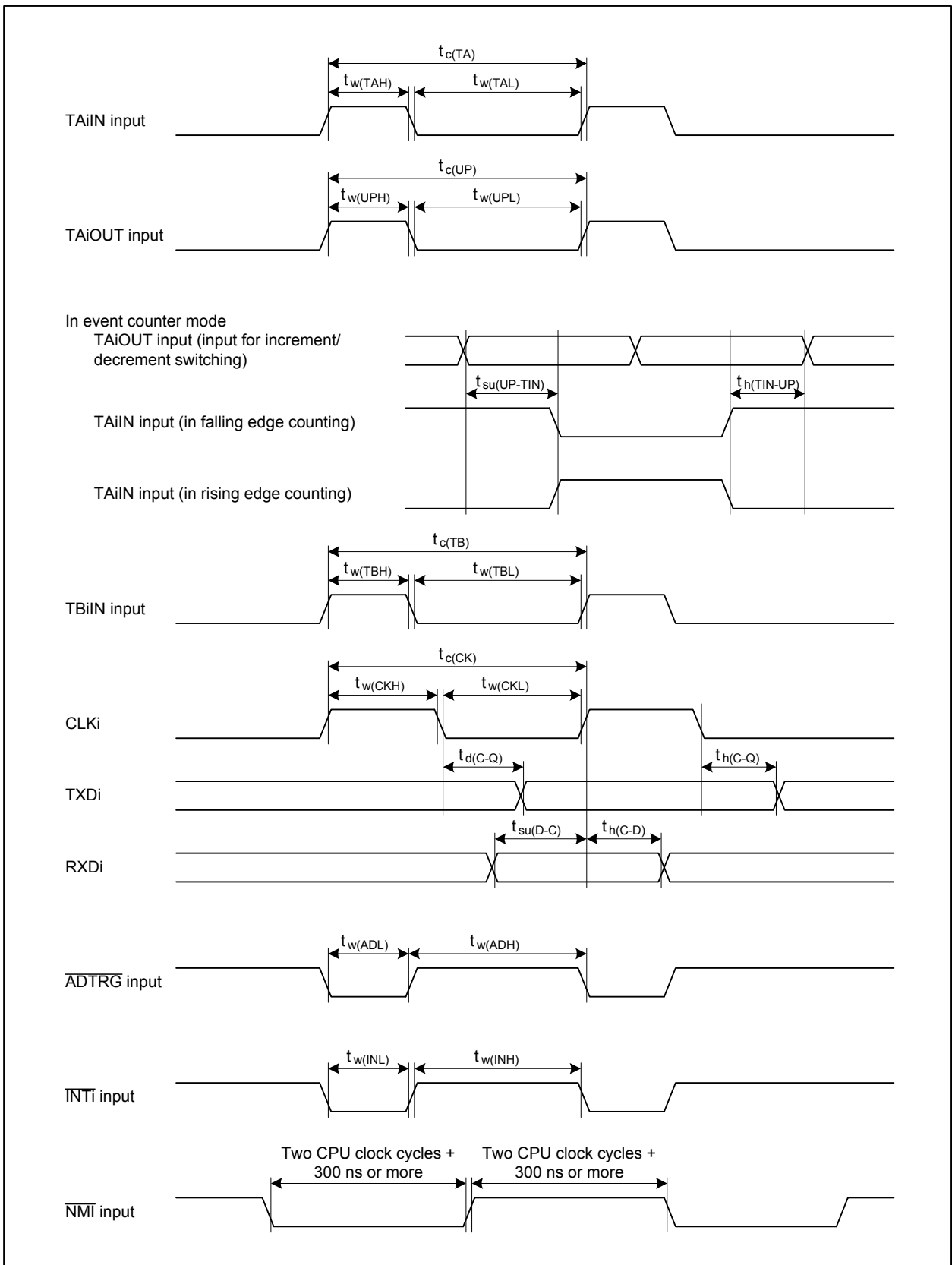


Figure 26.10 Timing of Peripherals

## 27. Usage Notes

### 27.1 Notes on Board Designing

#### 27.1.1 Power Supply Pins

The board should be designed so there is no potential difference between pins with the same name. Note the following point:

- Connect all VSS pins to the same GND. Traces for the pins should be as wide as physically possible so the same voltage can be applied to every VSS pin.

Insert a capacitor between VCC1 pin and the VSS pin, and VCC2 pin and the VSS pin to prevent operation errors due to noise. The capacitor should be beneficially effective at high and low frequencies and should have a capacitance of approximately 0.1  $\mu$ F. The traces for the capacitor and the power supply pins should be as short and wide as physically possible.

#### 27.1.2 Supply Voltage

The device is operationally guaranteed under operating conditions specified in electrical characteristics.

Drive the  $\overline{\text{RESET}}$  pin low before the supply voltage becomes lower than the operating voltage.

## 27.2 Notes on Register Setting

### 27.2.1 Registers with Write-only Bits

Read-modify-write instructions cannot be used when setting a register containing write-only bits. Read-modify-write instructions read a value of an address, modify the value, and write the modified value to the same address. Table 27.1 lists read-modify-write instructions, and Table 27.2 lists registers containing write-only bits. To set a new value by modifying the previous one, write the previous value into RAM as well as to the register, change the contents of the RAM and then transfer the new value to the register by the MOV instruction.

**Table 27.1 Read-modify-write Instructions**

Function	Mnemonic
Transfer	MOV <i>Dir</i>
Bit processing	BCLR, BMC <i>nd</i> , BNOT, BSET, BTSTC, and BTSTS
Shifting	ROLC, RORC, ROT, SHA, and SHL
Arithmetic operation	ABS, ADC, ADCF, ADD, ADSF, DEC, DIV, DIVU, DIVX, EXTS, EXTZ, INC, MUL, MULU, NEG, SBB, and SUB
Decimal operation	DADC, DADD, DSBB, and DSUB
Floating-point operation	ADDF, DIVF, MULF, and SUBF
Logical operation	AND, NOT, OR, and XOR

**Table 27.2 Registers with Write-only Bits**

Module	Register	Symbol	Address
Watchdog timer	Watchdog timer start register	WDTS	04404Eh
Timer A	Timer A0 register <sup>(1)</sup>	TA0	0347h-0346h
	Timer A1 register <sup>(1)</sup>	TA1	0349h-0348h
	Timer A2 register <sup>(1)</sup>	TA2	034Bh-034Ah
	Timer A3 register <sup>(1)</sup>	TA3	034Dh-034Ch
	Timer A4 register <sup>(1)</sup>	TA4	034Fh-034Eh
	Increment/decrement select register	UDF	0344h
Three-phase motor control timers	Timer B2 interrupt generating frequency set counter	ICTB2	030Dh
	Timer A1-1 register	TA11	0303h-0302h
	Timer A2-1 register	TA21	0305h-0304h
	Timer A4-1 register	TA41	0307h-0306h
	Dead time timer	DTT	030Ch
Serial interface	UART0 bit rate register	U0BRG	0369h
	UART1 bit rate register	U1BRG	02E9h
	UART2 bit rate register	U2BRG	0339h
	UART3 bit rate register	U3BRG	0329h
	UART4 bit rate register	U4BRG	02F9h
	UART5 bit rate register	U5BRG	01C9h
	UART6 bit rate register	U6BRG	01D9h
	UART7 bit rate register	U7BRG	01E1h
	UART8 bit rate register	U8BRG	01E9h
	UART0 transmit buffer register	U0TB	036Bh-036Ah
	UART1 transmit buffer register	U1TB	02EBh-02EAh
	UART2 transmit buffer register	U2TB	033Bh-033Ah
	UART3 transmit buffer register	U3TB	032Bh-032Ah
	UART4 transmit buffer register	U4TB	02FBh-02FAh
	UART5 transmit buffer register	U5TB	01CBh-01CAh
	UART6 transmit buffer register	U6TB	01DBh-01DAh
	UART7 transmit buffer register	U7TB	01E3h-01E2h
	UART8 transmit buffer register	U8TB	01EBh-01EAh
	Intelligent I/O	Group 2 SIO transmit buffer register	G2TB

**Note:**

1. The register has write-only bits in one-shot timer mode and pulse-width modulation mode.



## 27.3 Notes on Clock Generator

### 27.3.1 Sub Clock

#### 27.3.1.1 Oscillator Constant Matching

The constant matching of the sub clock oscillator should be evaluated in both cases when the drive strength is high and low.

Contact the oscillator manufacturer for details on the oscillation circuit constant matching.

### 27.3.2 Power Control

Do not switch the base clock source until the oscillation of the clock to be used has stabilized. However, this does not apply to the on-chip oscillator since it starts running immediately after the CM31 bit in the CM3 register is set to 1.

To switch the base clock source from the PLL clock to a low speed clock, use the MOV.L or OR.L instruction to set the BCS bit in the CCR register to 1.

- Program example in assembly language

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

- Program example in C language

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

#### 27.3.2.1 Stop Mode

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

#### 27.3.2.2 Suggestions for Power Saving

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

- I/O pins:

If inputs are floating, both transistors may be conducting. Set unassigned pins to input mode and connect each of them to VSS via a resistor, or set them to output mode and leave them open.

- A/D converter:

When not performing the A/D conversion, set the VCUT bit in the AD0CON1 register to 0 (VREF disconnected). To perform the A/D conversion, set the VCUT bit to 1 (VREF connected) and wait at least 1  $\mu\text{s}$  before starting conversion.

- D/A converter:

When not performing the D/A conversion, set the DAiE bit in the DACON register ( $i = 0, 1$ ) to 0 (output disabled) and the DAi register to 00h.

- Peripheral clock stop:

When entering wait mode, power consumption can be reduced by setting the CM02 bit in the CM0 register to 1 to stop the peripheral clock source. However, this setting does not stop the fC32.

## 27.4 Notes on Bus

### 27.4.1 Notes on Designing a System

When a flash memory rewrite is performed in CPU rewrite mode using memory expansion mode, the use of  $\overline{CS0}$  space and  $\overline{CS3}$  space has the following restrictions:

- If the FEBC0 and/or FEBC3 registers are set in CPU rewrite mode, the bus format for the corresponding space functions as separate bus. Any external devices connected in multiplexed bus format become inaccessible.
- If the FEBC0 and/or FEBC3 registers are set in CPU rewrite mode, the bus timing for the corresponding space changes. This may cause external devices to become inaccessible depending on the register settings.

Devices required to be accessed in CPU rewrite mode should be allocated in  $\overline{CS1}$  space and/or  $\overline{CS2}$  space.

### 27.4.2 Notes on Register Settings

#### 27.4.2.1 Chip Select Boundary Select Registers

When not using memory expansion mode, do not change values after a reset for registers CB01, CB12, and CB23.

When using memory expansion mode, set all of these registers to a value within the specified range whether or not each chip select space is used.

#### 27.4.2.2 External Bus Control Registers

Registers EBC0 and EBC3 share respective addresses with registers FEBC0 and FEBC3. If the FEBC0 and/or FEBC3 registers are set while the flash memory is being rewritten, set the EBC0 and/or EBC3 registers again after rewriting the flash memory.

## 27.5 Notes on Interrupts

### 27.5.1 ISP Setting

The interrupt stack pointer (ISP) is initialized to 00000000h after a reset. Set a value to the ISP before an interrupt is accepted, otherwise the program may go out of control. A multiple of 4 should be set to the ISP, which enables faster interrupt sequence due to less memory access.

When using NMI, in particular, since this interrupt cannot be disabled, set the PM24 bit in the PM2 register to 1 (NMI enabled) after setting the ISP at the beginning of the program.

### 27.5.2 NMI

- NMI cannot be disabled once the PM24 bit in the PM2 register is set to 1 (NMI enabled). This bit setting should be done only when using NMI.
- When the PM24 bit in the PM2 register is 1 (NMI enabled), the P8\_5 bit in the P8 register is enabled just for monitoring the  $\overline{\text{NMI}}$  pin state. It is not enabled as a general port.

### 27.5.3 External Interrupts

- The input signal to the  $\overline{\text{INTi}}$  pin requires the pulse width specified in the electrical characteristics ( $i = 0$  to 5). If the pulse width is narrower than the specification, an external interrupt may not be accepted.
- When the effective level or edge of the  $\overline{\text{INTi}}$  pin ( $i = 0$  to 5) is changed by the following bits: bits POL, LVS in the INTiIC register, the IFSR0i bit ( $i = 0$  to 5) in the IFSR0 register, the corresponding IR bit may become 1 (interrupt requested). When setting the above mentioned bits, preset bits ILVL2 to ILVL0 in the INTiIC register to 000b (interrupt disabled). After setting the above mentioned bits, set the corresponding IR bit to 0 (no interrupt requested), then rewrite bits ILVL2 to ILVL0.

## 27.6 Notes on DMAC

### 27.6.1 DMAC-associated Register Settings

- Set DMAC-associated registers while bits MDi1 and MDi0 in the DMDi register are 00b (DMA transfer disabled) (i = 0 to 3). Then, set bits MDi1 and MDi0 to 01b (single transfer) or 11b (repeat transfer) at the end of the setup procedure. This procedure also applies when rewriting bits UDAi, USAi, and BWi1 and BWi0 in the DMDi register.
- When rewriting the DMAC-associated registers while DMA transfer is enabled, stop the peripherals that can be DMA triggers so that no DMA transfer request is generated, then set bits MDi1 and MDi0 in the DMDi register of the corresponding channel to 00b (DMA transfer disabled).
- Once a DMA transfer request is accepted, DMA transfer cannot be disabled even if setting bits MDi1 and MDi0 in the DMDi register to 00b (DMA transfer disabled). Do not change the settings of any DMAC-associated registers other than bits MDi1 and MDi0 until the DMA transfer is completed.
- After setting registers DMiSL and DMiSL2, wait at least six peripheral bus clocks to set bits MDi1 and MDi0 in the DMDi register to 01b (single transfer) or 11b (repeat transfer).

### 27.6.2 Reading DMAC-associated Registers

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

## 27.7 Notes on Timers

### 27.7.1 Timer A and Timer B

All timers are stopped after a reset. To restart timers, configure parameters such as operating mode, count source, and counter value, then set the TAI<sub>S</sub> bit or TB<sub>J</sub>S bit in the TABSR or TBSR register to 1 (count starts) (i = 0 to 4; j = 0 to 5).

The following registers and bits should be set while the TAI<sub>S</sub> bit or TB<sub>J</sub>S bit is 0 (count stops):

- Registers TAI<sub>MR</sub> and TB<sub>J</sub>MR
- UDF register
- Bits TAZIE, TA0TGL, and TA0TGH in the ONSF register
- TRGSR register

### 27.7.2 Timer A

#### 27.7.2.1 Timer Mode

- While the timer counter is running, the TAI register indicates a counter value at any given time. However, FFFFh is read while reloading is in progress. A set value is read if the TAI register is set while the timer counter is stopped.

#### 27.7.2.2 Event Counter Mode

- While the timer counter is running, the TAI register indicates a counter value at any given time. However, FFFFh is read if the timer counter underflows or 0000h if overflows while reloading is in progress. A set value is read if the TAI register is set while the timer counter is stopped.

#### 27.7.2.3 One-shot Timer Mode

- If the TAI<sub>S</sub> bit in the TABSR register is set to 0 (count stops) while the timer counter is running, the following operations are performed:
  - The timer counter stops and the setting value of the TAI register is reloaded.
  - A low signal is output at the TAI<sub>OUT</sub> pin.
  - The IR bit in the TAI<sub>IC</sub> register becomes 1 (interrupts requested) after one CPU clock cycle.
- The one-shot timer is operated by an internal count source. When the trigger is an input to the TAI<sub>IN</sub> pin, the signal is output with a maximum one count source clock delay after a trigger input to the TAI<sub>IN</sub> pin.
- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt, set the IR bit to 0 after one of the settings below is done:
  - Select one-shot timer mode after a reset.
  - Switch operating modes from timer mode to one-shot timer mode.
  - Switch operating modes from event counter mode to one-shot timer mode.
- If a retrigger occurs while counting, the timer counter decrements by one, reloads the setting value of the TAI register, and then continues counting. To generate a retrigger while counting, wait at least one count source cycle after the last trigger is generated.
- When an external trigger input is selected to start counting in timer A one-shot mode, do not provide an external retrigger for 300 ns before the timer counter reaches 0000h. Otherwise, it may stop counting.

#### 27.7.2.4 Pulse-width Modulation Mode

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt, set the IR bit to 0 after one of the settings below is done (i = 0 to 4):
  - Select pulse-width modulation mode after a reset.
  - Switch operating modes from timer mode to pulse-width modulation mode.
  - Switch operating modes from event counter mode to pulse-width modulation mode.
  
- If the TAI<sub>S</sub> bit in the TABSR register is set to 0 (count stops) while PWM pulse is output, the following operations are performed:
  - The timer counter stops.
  - The output level at the TAI<sub>OUT</sub> pin changes from high to low. The IR bit becomes 1.
  - When a low signal is output at the TAI<sub>OUT</sub> pin, it does not change. The IR bit does not change, either.

### 27.7.3 Timer B

#### 27.7.3.1 Restriction for the Use of 64-pin Package's Products

- TB4IN pin is not available in the 64-pin package. Timer B4 is available in timer mode only.

#### 27.7.3.2 Timer Mode and Event Counter Mode

- While the timer counter is running, the TBJ register indicates a counter value at any given time ( $j = 0$  to  $5$ ). However, FFFFh is read while reloading is in progress. When a value is set to the TBJ register while the timer counter is stopped, if the TBJ register is read before the count starts, the set value is read.

#### 27.7.3.3 Pulse Period/Pulse-width Measure Mode

- To set the MR3 bit in the TBJMR register to 0 (no overflow), write the TBJMR register after the MR3 bit becomes 1 (overflow) and at least one count source cycle has elapsed while the TBJS bit in the TABSR or TBSR register is 1 (start counter).
- Use the IR bit in the TBJIC register to detect overflow. The MR3 bit is used only to determine an interrupt request source within the interrupt handler.
- The counter value is undefined when the timer counter starts. Therefore, the timer counter may overflow before a measured pulse is applied on the initial valid edge and cause a timer B $j$  interrupt request to be generated.
- When the measured pulse is applied on the initial valid edge after the timer counter starts, an undefined value is transferred to the reload register. At this time, a timer B $j$  interrupt request is not generated.
- The IR bit may become 1 (interrupt requested) by changing bits MR1 and MR0 in the TBJMR register after the timer counter starts. However, if the same value is rewritten to bits MR1 and MR0, the IR bit does not change.
- Pulse width is continuously measured in pulse-width measure mode. Whether the measurement result is high-level width or not is determined by a program.
- When an overflow occurs at the same time a pulse is applied on the valid edge, this pulse is not recognized since an interrupt request is generated only once. Do not let an overflow occur in pulse period measure mode.
- In pulse-width measure mode, determine whether an interrupt source is a pulse applied on the valid edge or an overflow by reading the port level in the timer B $j$  interrupt handler.

## 27.8 Notes on Three-phase Motor Control Timers

### 27.8.1 Shutdown

- When a low signal is applied to the  $\overline{\text{NMI}}$  pin with the following bit settings, pins TA1OUT, TA2OUT, and TA4OUT become high-impedance: the PM24 bit in the PM2 register is 1 (NMI enabled), the INV02 bit in the INVC0 register is 1 (three-phase motor control timers used), and the INV03 bit is 1 (three-phase motor control timer output enabled).

### 27.8.2 Register Setting

- Do not write to the TAI1 register before and after timer B2 underflows ( $i = 1, 2, 4$ ). Before writing to the TAI1 register, read the TB2 register to verify that sufficient time remains until timer B2 underflows. Then, immediately write to the TAI1 register so no interrupt handling is performed during this write procedure. If the TB2 register indicates little time remains until the underflow, write to the TAI1 register after timer B2 underflows.

### 27.8.3 Restriction for the Use of 64-pin Package's Product

Port P3 is available only for output pins V and  $\overline{\text{V}}$  in the 64-pin package. Use ports P7 and P8 for other output pins U,  $\overline{\text{U}}$ , W, and  $\overline{\text{W}}$ .



## 27.9 Notes on Serial Interface

### 27.9.1 Changing the UiBRG Register (i = 0 to 8)

- Set the UiBRG register after setting bits CLK1 and CLK0 in the UiC0 register. When these bits are changed, the UiBRG register must be set again.
- When a clock is input immediately after the UiBRG register is set to 00h, the counter may become FFh. In this case, it requires extra 256 clocks to reload 00h to the register. Once 00h is reloaded, the counter performs the operation without dividing the count source according to the setting.

### 27.9.2 Restriction for the Use of 64-pin Package's Products

In the 64-pin package, all signals of UART4, UART6, and UART7 are not connected to respective external pins. Use UART0 to UART3, UART5, or UART8.

### 27.9.3 Synchronous Serial Interface Mode

#### 27.9.3.1 Selecting an External Clock

- If an external clock is selected, the following conditions must be met while the external clock is held high when the CKPOL bit in the UiC0 register is 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge), or while the external clock is held low when the CKPOL bit is 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge) (i = 0 to 8):
  - The TE bit in the UiC1 register is 1 (transmission enabled).
  - The RE bit in the UiC1 register is 1 (reception enabled). This bit setting is not required when only transmitting.
  - The TI bit in the UiC1 register is 0 (data held in the UiTB register).

#### 27.9.3.2 Receive Operation

- In synchronous serial interface mode, the transmit/receive clock is controlled by the transmit control circuit. Set UARTi-associated registers for a transmit operation, even if the MCU is used only for receive operation (i = 0 to 8). Dummy data is output from the TXDi pin while receiving when the TXDi pin is set to output mode.
- When data is received continuously, an overrun error occurs when the RI bit in the UiC1 register is 1 (data held in the UiRB register) and the seventh bit of the next data is received in the UARTi receive shift register. Then, the OER bit in the UiRB register becomes 1 (overrun error occurred). In this case, the UiRB register becomes undefined. If an overrun error occurs, the IR bit in the SiRIC register does not change to 1.

### 27.9.4 Special Mode 1 (I<sup>2</sup>C Mode)

- To generate a START condition, STOP condition, or repeated START condition, set the STSPSEL bit in the UiSMR4 register to 0 (i = 0 to 6). Then, wait at least a half clock cycle of the transmit/receive clock to change the condition generate bits (STAREQ, RSTAREQ, or STPREQ bit) from 0 to 1.

### 27.9.5 Reset Procedure or Suspend/Resume Procedure

Operations which result in communication errors such as rewriting function select registers during transmission/reception should not be performed. Follow the procedure below to reset the internal circuit once the communication error occurs in the following cases: when the operation above is performed by a receiver or transmitter or when a bit slip is caused by noise.

Also follow the procedure below when suspending and resuming communication in an emergency.

#### A. Synchronous Serial Interface Mode

- (1) Set the TE bit in the UiC1 register to 0 (transmission disabled) and the RE bit to 0 (reception disabled) (i = 0 to 8).
- (2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
- (3) Set bits SMD2 to SMD0 in the UiMR register to 001b (synchronous serial interface mode).
- (4) Set the TE bit in the UiC1 register to 1 (transmission enabled) and the RE bit to 1 (reception enabled) if necessary.

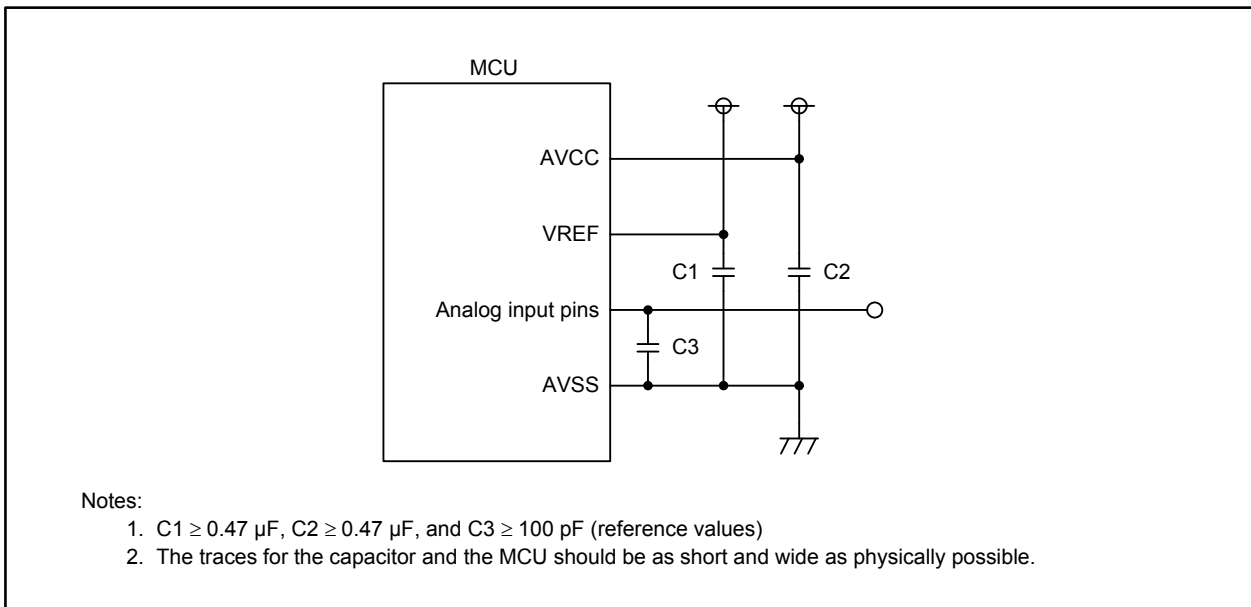
#### B. UART Mode

- (1) Set the TE bit in the UiC1 register to 0 (transmission disabled) and the RE bit to 0 (reception disabled).
- (2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
- (3) Set bits SMD2 to SMD0 in the UiMR register to 100b (UART mode, 7-bit character length), 101b (UART mode, 8-bit character length), or 110b (UART mode, 9-bit character length).
- (4) Set the TE bit in the UiC1 register to 1 (transmission enabled) and the RE bit to 1 (reception enabled) if necessary.

## 27.10 Notes on A/D Converter

### 27.10.1 Notes on Designing Boards

- Three capacitors should be placed between the AVSS pin and pins such as AVCC, VREF, and analog inputs (AN\_0 to AN\_7, AN0\_0 to AN0\_7, and AN2\_0 to AN2\_7) to avoid erroneous operations caused by noise or latchup, and to reduce conversion errors. Figure 27.1 shows an example of pin configuration for A/D converter.



**Figure 27.1 Pin Configuration for the A/D Converter**

- Do not use AN\_4 to AN\_7 for analog input if the key input interrupt is to be used. Otherwise, a key input interrupt request occurs when the A/D input voltage becomes VIL or lower.
- In the 100-pin package, when  $AVCC = VREF = VCC1 \geq VCC2$ , A/D input voltage for pins AN\_0 to AN\_7, ANEX0, and ANEX1 should be VCC1 or lower, and A/D input voltage for pins AN0\_0 to AN0\_7, and AN2\_0 to AN2\_7 should be VCC2 or lower. In the 64-pin package, under the same condition as that in the 100-pin package, all A/D input voltages should be VCC1 or lower.

### 27.10.2 Notes on Programming

- The following registers should be written while A/D conversion is stopped. That is, before a trigger occurs: AD0CON0 (except the ADST bit), AD0CON1, AD0CON2, AD0CON3, and AD0CON4.
- When the VCUT bit in the AD0CON1 register is changed from 0 (VREF connected) to 1 (VREF disconnected), wait for at least 1  $\mu$ s before starting A/D conversion. When not performing A/D conversion, set the VCUT bit to 0 to reduce power consumption.
- Set the port direction bit for the pin to be used as an analog input pin to 0 (input). Set the ASEL bit of the corresponding port function select register to 1 (port is used as A/D input).
- When the TRG bit in the AD0CON0 register is 1 (external trigger or hardware trigger), set the corresponding port direction bit (PD9\_7 bit) for the  $\overline{\text{ADTRG}}$  pin to 0 (input).
- The  $\phi_{\text{AD}}$  frequency should be 16 MHz or lower when VCC1 is 4.2 to 5.5 V, and 10 MHz or lower when VCC1 is 3.0 to 4.2 V. It should be 1 MHz or higher when the sample and hold function is enabled. If not, it should be 250 kHz or higher.
- When A/D operating mode (bits MD1 and MD0 in the AD0CON0 register or the MD2 bit in the AD0CON1 register) has been changed, reselect analog input pins by setting bits CH2 to CH0 in the AD0CON0 register or bits SCAN1 and SCAN0 in the AD0CON1 register.
- If the AD0i register is read when the A/D converted result is stored to the register, the stored value may have an error ( $i = 0$  to 7). Read the AD0i register after A/D conversion is completed. In one-shot mode or single sweep mode, read the AD0i register after the IR bit in the AD0IC register becomes 1 (interrupt requested). In repeat mode, repeat sweep mode 0, or repeat sweep mode 1, an interrupt request can be generated each time A/D conversion is completed when the DUS bit in the AD0CON3 register is 1 (DMAC operating mode enabled). Similar to the other modes above, read the AD00 register after the IR bit in the AD0IC register becomes 1 (interrupt requested).
- When an A/D conversion is halted by setting the ADST bit in the AD0CON0 register to 0, the converted result is undefined. In addition, the unconverted AD0i register may also become undefined. Consequently, the AD0i register should not be used just after A/D conversion is halted.
- External triggers cannot be used in DMAC operating mode. When the DMAC is configured to transfer converted results, do not read the AD00 register by a program.
- While in single sweep mode, if A/D conversion is halted by setting the ADST bit in the AD0CON0 register to 0 (A/D conversion is stopped), an interrupt request may be generated even though the sweep is not completed. To halt A/D conversion, disable interrupts before setting the ADST bit to 0.

## 27.11 Notes on Flash Memory Rewriting

### 27.11.1 Note on Power Supply

- Keep the supply voltage constant within the range specified in the electrical characteristics while a rewrite operation on the flash memory is in progress. If the supply voltage goes beyond the guaranteed value, the device cannot be guaranteed.

### 27.11.2 Note on Hardware Reset

- Do not perform a hardware reset while a rewrite operation on the flash memory is in progress.

### 27.11.3 Note on Flash Memory Protection

- If an ID code written in an assigned address has an error, any read/write operation on the flash memory in standard serial I/O mode is disabled.

### 27.11.4 Notes on Programming

- Do not set the FEW bit in the FMCR register to 1 (CPU rewrite mode) in low speed mode or low power mode.
- The program, block erase, lock bit program, and protect bit program are interrupted by an NMI, a watchdog timer interrupt, an oscillator stop detection interrupt, or a low voltage detection interrupt. If any of the software commands above are interrupted, erase the corresponding block and then execute the same command again. If the block erase command is interrupted, the lock bit and protect bit values become undefined. Therefore, disable the lock bit, and then execute the block erase command again.

### 27.11.5 Notes on Interrupts

- EW0 mode
  - To use interrupts assigned to the relocatable vector table, the vector table should be addressed in RAM space.
  - When an NMI, watchdog timer interrupt, oscillator stop detection interrupt, or low voltage detection interrupt occurs, the flash memory module automatically enters read array mode. Therefore, these interrupts are enabled even during a rewrite operation. However, the rewrite operation in progress is aborted by the interrupts and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.
  - Instructions BRK, INTO, and UND, which refer to data on the flash memory, cannot be used in this mode.
- EW1 mode
  - Interrupts assigned to the relocatable vector table should not be accepted during program or block erase operation.
  - The watchdog timer interrupt should not be generated.
  - When an NMI, watchdog timer interrupt, oscillator stop detection interrupt, or low voltage detection interrupt occurs, the flash memory module automatically enters read array mode. Therefore, these interrupts are enabled even during a rewrite operation. However, the rewrite operation in progress is aborted by the interrupts and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the EWM bit in the FMR0 register to 1 (EW1 mode) and the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.

### 27.11.6 Notes on Rewrite Control Program

- EW0 mode
  - If the supply voltage drops during the rewrite operation of blocks having the rewrite control program, the rewrite control program may not be successfully rewritten, and the rewrite operation itself may not be performed. In this case, perform the rewrite operation by serial programmer or parallel programmer.
- EW1 mode
  - Do not rewrite blocks having the rewrite control program.

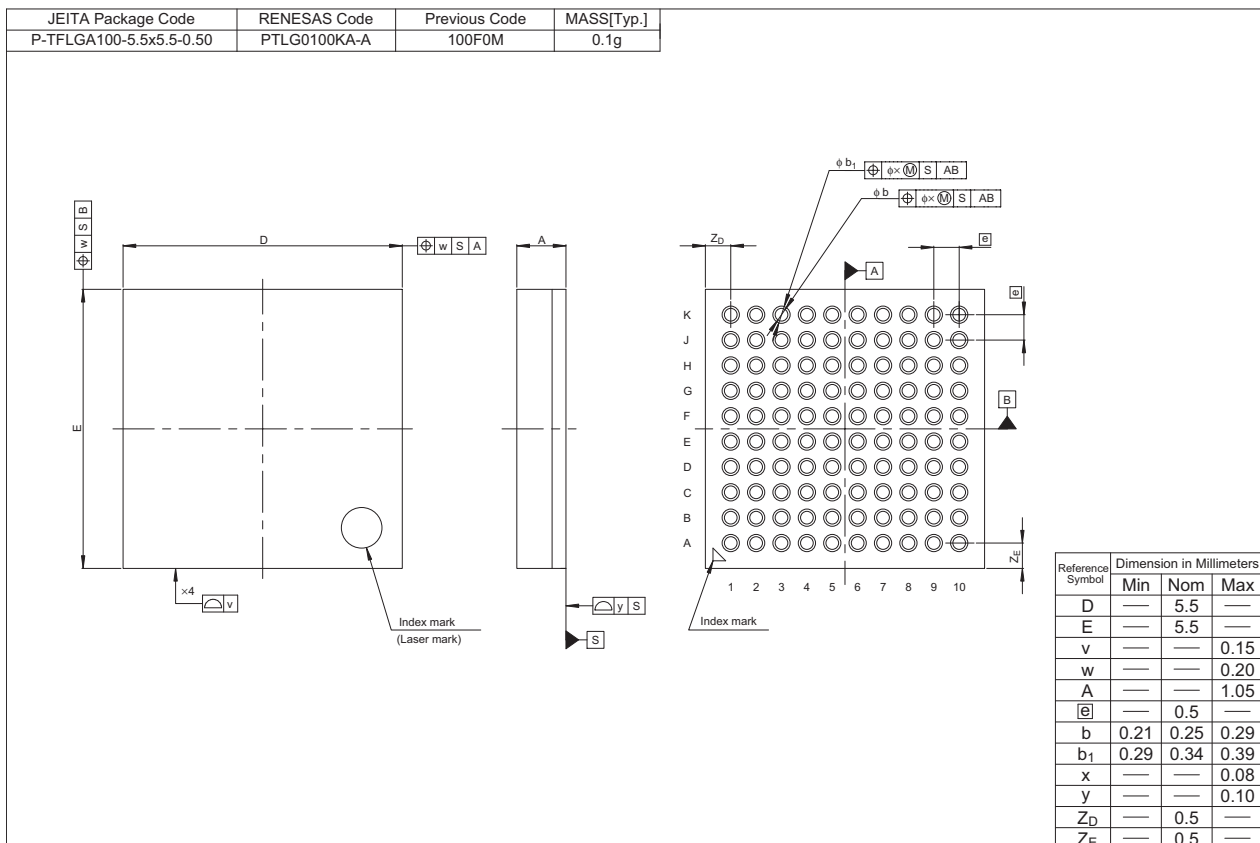
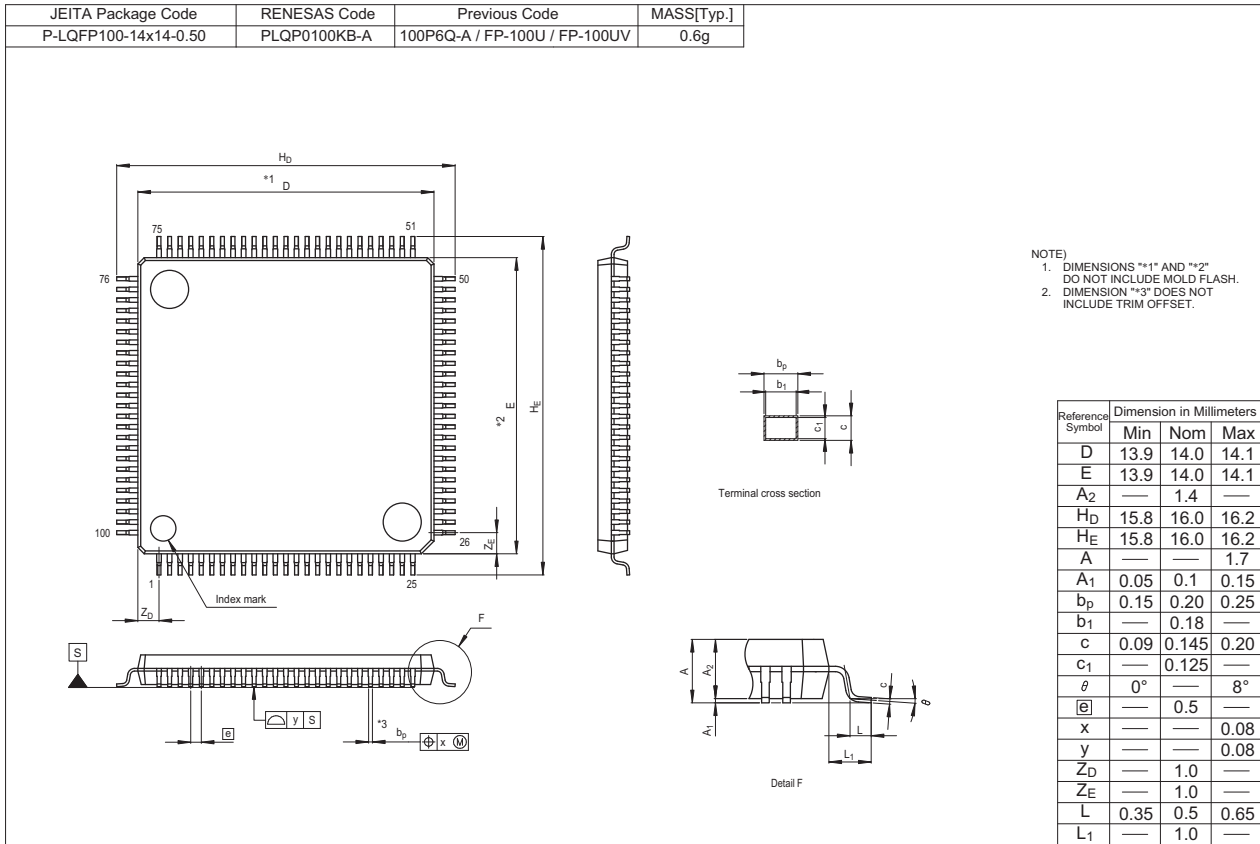
### 27.11.7 Notes on Number of Program/Erase Cycles and Software Command Execution Time

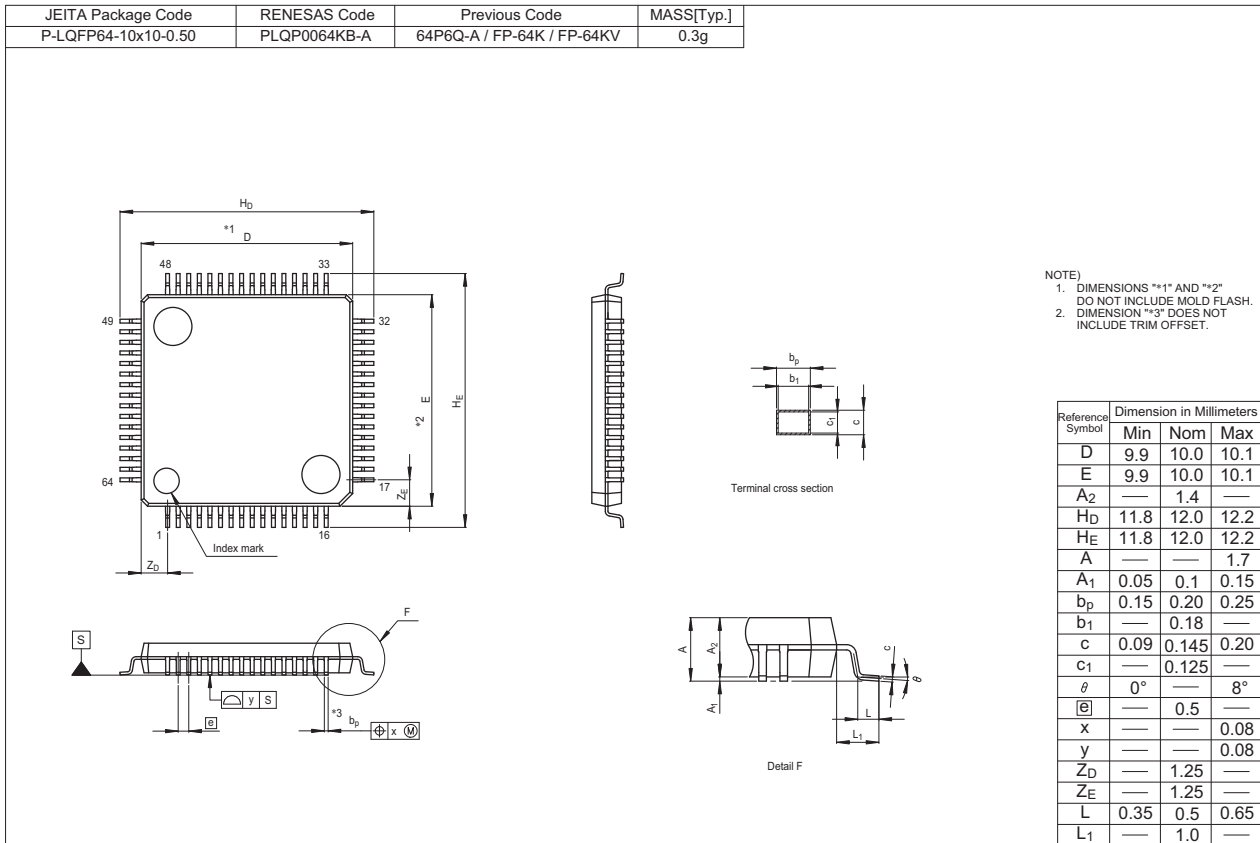
- The time to execute software commands (program, block erase, lock bit program, and protect bit program) increases as the number of program/erase cycles increases. If the number of program/erase cycles exceeds the endurance value specified in the electrical characteristics, it may take an unpredictable amount of time to execute the software commands. The wait time for executing software commands should be set much longer than the execution time specified in the electrical characteristics.

### 27.11.8 Other Notes

- The minimum values of program/erase cycles specified in the electrical characteristics are the maximum values that can guarantee the initial performance of the flash memory. The program/erase operation may still be performed even if the number of program/erase cycles exceeds the guaranteed values.
- Chips repeatedly programmed and erased for debugging should not be used for commercial products.

# Appendix 1. Package Dimensions







## INDEX

### A

A0	26
A1	26
A2	26
A3	26
AD00 to AD07	292
AD0CON0	288
AD0CON1	289
AD0CON2	290
AD0CON3	291
AD0CON4	292
AD0IC	137
Address Register	26

### B

B Flag	27
BCN0IC to BCN6IC	137
BRK Instruction Interrupt	128
BRK2 Instruction Interrupt	128
BTSR	323

### C

C Flag	26
Carry Flag	26
CB01	105
CB12	105
CB23	106
CCR	71
CM0	72
CM1	73
CM2	74
CM3	74
CPSRF	75
CRCD	308
CRCIN	309
CSOP0	104
CSOP1	104

### D

D Flag	26
DA0, DA1	307
DACON	307
Data Register	26
DCR0 to DCR3	28, 162
DCT0 to DCT3	28, 161
DDA0 to DDA3	28, 163

DDR0 to DDR3	28, 163
Debug Flag	26
DM0IC to DM3IC	137
DM0SL to DM3SL	157
DM0SL2 to DM3SL2	158
DMA Destination Address Register	28
DMA Destination Address Reload Register	28
DMA Mode Register	28
DMA Source Address Register	28
DMA Source Address Reload Register	28
DMA Terminal Count Register	28
DMA Terminal Count Reload Register	28
DMD0 to DMD3	28, 161
DP Bit	27
DSA0 to DSA3	28, 162
DSR0 to DSR3	28, 163
DTT	229
DVCR	63

### E

EBC0 to EBC3	109
--------------	-----

### F

Fast Interrupt	130
FB	26
FBPM0	400
FBPM1	400
FEBC0, FEBC3	397
Fixed-point Designation Bit	27
Flag Register	26
FLG	26
Floating-point Overflow Flag	27
Floating-point Rounding Mode	27
Floating-point Underflow Flag	27
FMCR	396
FMR0	398
FMR1	399
FMSR0	399
FO Flag	27
FPR0	398
Frame Base Register	26
FU Flag	27

### G

G0BCR0 to G2BCR0	320
G0BCR1, G1BCR1	321
G0BT to G2BT	319
G0FE to G2FE	329
G0FS, G1FS	328
G0PO0 to G0PO7	328

G0POCR0 to G0POCR7 .....	326
G0TM0 to G0TM7 .....	325
G0TMCR0 to G0TMCR7 .....	324
G0TPR6, G0TPR7 .....	324
G1PO0 to G1PO7 .....	328
G1POCR0 to G1POCR7 .....	326
G1TM0 to G1TM7 .....	325
G1TMCR0 to G1TMCR7 .....	324
G1TPR6, G1TPR7 .....	324
G2BCR1 .....	322
G2CR .....	356
G2MR .....	355
G2PO0 to G2PO7 .....	328
G2POCR0 to G2POCR7 .....	327
G2RB .....	355
G2RTP .....	329
G2TB .....	354

**H**

Hardware Interrupt .....	129
--------------------------	-----

**I**

I Flag .....	27
ICTB2 .....	221
IDB0, IDB1 .....	220
IEAR .....	357
IECR .....	357
IERIF .....	358
IETIF .....	358
IFS0 .....	378
IFS2 .....	379
IFS3 .....	381
IFS7 .....	381
IFSR0 .....	146, 248
IFSR1 .....	147, 249
IIO0IC to IIO11IC .....	137
IIO0IE to IIO11IE .....	151
IIO0IR to IIO11IR .....	150
INT Instruction Interrupt .....	128
INT0IC to INT5IC .....	138
INTB .....	26
Interrupt Control Register .....	137
Interrupt Enable Flag .....	27
Interrupt request level .....	138
Interrupt Response Time .....	142
Interrupt Sequence .....	141
Interrupt Stack Pointer .....	26
Interrupt Types .....	127
Interrupt Vector Table Base Register .....	26
INVC0 .....	217
INVC1 .....	218

IOBC .....	219
IPL .....	27, 136
ISP .....	26

**K**

KUPIC .....	137
-------------	-----

**L**

Low Voltage Detection Interrupt .....	64, 129
Low Voltage Detector .....	61
LVDC .....	62

**M**

Maskable Interrupt .....	127
MOD .....	174

**N**

NMI (Non Maskable Interrupt) .....	129
Non-maskable Interrupt .....	127

**O**

O Flag .....	27
ONSF .....	188
Oscillator Stop Detection Interrupt .....	129
Overflow Flag .....	27
Overflow Interrupt .....	128

**P**

P0 to P10 .....	181
P0_0S to P0_7S .....	365
P10_0S to P10_7S .....	377
P1_0S to P1_7S .....	366
P2_0S to P2_7S .....	367
P3_0S to P3_7S .....	369
P4_0S to P4_7S .....	371
P5_0S to P5_7S .....	372
P6_0S to P6_7S .....	373
P7_0S to P7_7S .....	374
P8_0S to P8_4S, P8_6S, P8_7S .....	375
P9_3S to P9_7S .....	376
PBC .....	101
PC .....	26
PCR .....	385
PD0 to PD10 .....	363
Peripheral Interrupt .....	129
PLC0 .....	80

PLC1 .....	81
PM0 .....	67
PM2 .....	76
PM3 .....	77
PRCR .....	124
PRCR2 .....	125
PRCR3 .....	125
Processor Interrupt Priority Level .....	27, 136
Program Counter .....	26
PRR .....	126
PUR0 .....	382
PUR1 .....	383
PUR2 .....	383
PUR3 .....	384

**R**

R2R0 .....	26
R3R1 .....	26
R6R4 .....	26
R7R5 .....	26
Register Bank Select Flag .....	27
Register Saving .....	143
RIPL1, RIPL2 .....	140, 171
RND .....	27

**S**

S Flag .....	26
S0RIC to S8RIC .....	137
S0TIC to S8TIC .....	137
Save Flag Register .....	28
Save PC Register .....	28
SB .....	26
Sign Flag .....	26
Single-step Interrupt .....	129
Software Interrupt .....	128
SP .....	26
Special Interrupt .....	129
Stack Pointer .....	26
Stack Pointer Select Flag .....	27
Static Base Register .....	26
SVF .....	28
SVP .....	28

**T**

TA0 to TA4 .....	185
TA0IC to TA4IC .....	137
TA0MR to TA4MR ..	186, 192, 195, 198, 200
TA1, TA2, TA4, TA11, TA21, TA41 .....	225
TA1MR, TA2MR, TA4MR .....	226
TABSR .....	186, 204, 227

TB0 to TB5 .....	203
TB0IC to TB5IC .....	137
TB0MR to TB5MR .....	203, 206, 208, 210
TB2 .....	223
TB2MR .....	223
TB2SC .....	224
TBSR .....	204
TCSPR .....	75, 190
TRGSR .....	189, 227

**U**

U Flag .....	27
U0BRG to U8BRG .....	245
U0C0 to U6C0 .....	238
U0C1 to U6C1 .....	240
U0MR to U6MR .....	236
U0RB to U6RB .....	246
U0SMR to U6SMR .....	242
U0SMR2 to U6SMR2 .....	243
U0SMR3 to U6SMR3 .....	244
U0SMR4 to U6SMR4 .....	245
U0TB to U8TB .....	246
U78CON .....	241
U7C0, U8C0 .....	239
U7C1, U8C1 .....	240
U7MR, U8MR .....	237
U7RB, U8RB .....	247
UDF .....	187
Undefined Instruction Interrupt .....	128
User Stack Pointer .....	26
USP .....	26

**V**

VCT .....	28
Vector Register .....	28
VRCR .....	59

**W**

Watchdog Timer Interrupt .....	129
WDC .....	154
WDTS .....	154

**X**

X0R to X15R .....	311
XYC .....	311

**Y**

Y0R to Y15R ..... 312

**Z**

Z Flag ..... 26

Zero Flag ..... 26

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
1.00	Dec 12, 2008	—	Initial release
1.10	Oct 21, 2009	—	Second edition released
		—	<p>The manual in general</p> <ul style="list-style-type: none"> <li>• Added 100-pin plastic molded LGA and 80- and 64-pin plastic molded LQFP packages</li> <li>• When new tables/figures are added for 80-/64-pin packages, add the following description: “(for/in the 100-pin package)” to the title of corresponding current tables/figures</li> <li>• Changed the following expressions: “drive capacity”, to “drive power” (under Chapters 8 and 27); “start/stop condition”, to “start condition/ stop condition” (under Chapters 4, 11, and 18)</li> <li>• Modified the following descriptions: “Do not use any combination other than the above”, to “Only use the combination listed above” (under Chapters 6, 18, and 25); “This register should be rewritten after (the xxx bit in) the xxx register is set to 1/AAh (write enabled).”, to “Set (the xxx bit in) the xxx register to 1/AAh (write enabled) before rewriting this register.” (under Chapters 6, 8, 9, 18, and 25)</li> </ul>
		—	<p><b>About This Manual</b></p> <ul style="list-style-type: none"> <li>• Changed “26” chapters” in line 4 of <b>1. Purpose and Target User</b>, to “27 chapters”</li> <li>• Revised the table in <b>1. Purpose and Target User</b></li> <li>• Made major text modifications to <b>2. Numbers and Symbols</b></li> <li>• Revised the illustration in <b>3. Registers</b></li> </ul>
		1	<p><b>Chapter 1. Overview</b></p> <ul style="list-style-type: none"> <li>• Added description for 100-pin LGA and 80-/64-pin packages to lines 12 and 13 of <b>1.1</b>; Added description “a maximum of” to “nine channels of serial interface”; Deleted the whole description of “Notes to users”</li> </ul>
		2	<ul style="list-style-type: none"> <li>• Changed minimum RAM “40 Kbytes” in <b>Table 1.1</b>, to “32 Kbytes”</li> <li>• Modified description for “External Bus Expansion” in <b>Table 1.1</b>; Moved this unit below “Clock”</li> </ul>
		3	<ul style="list-style-type: none"> <li>• Modified description for “Flash memory” in <b>Table 1.2</b></li> <li>• Added “100-pin plastic molded TFLGA (PTLG0100KA-A)” to <b>Table 1.2</b></li> </ul>
		4-7	<ul style="list-style-type: none"> <li>• Added <b>Tables 1.3 to 1.6</b> to provide specifications for 80-/64-pin packages</li> </ul>
		8	<ul style="list-style-type: none"> <li>• Completed “under development” phase of part numbers R5F64110DFB, R5F64111DFB, R5F64114DFB, and R5F64115DFB in <b>Table 1.7</b></li> </ul>
		8, 9	<ul style="list-style-type: none"> <li>• Added product information for 100-pin LGA and 80-/64-pin packages to <b>Table 1.7 and Figure 1.1</b></li> </ul>
		11, 12, 14, 18, 21	<ul style="list-style-type: none"> <li>• Added <b>Figures 1.3, 1.4, and 1.6 to 1.8</b> to provide block diagrams and pin assignment for 100-pin LGA and 80-/64-pin packages</li> </ul>
		13	<ul style="list-style-type: none"> <li>• Corrected a typo “R5_3” for pin No. 41 in <b>Figure 1.5</b>, to “P5_3”</li> </ul>
		15-17	<ul style="list-style-type: none"> <li>• Added pin No. for LGA package to <b>Tables 1.8 to 1.10</b></li> </ul>
		19, 20, 22, 23	<ul style="list-style-type: none"> <li>• Added <b>Tables 1.11 to 1.14</b> to provide pin characteristics for 80-/64-pin packages</li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		26	<ul style="list-style-type: none"> <li>• Deleted the following description: “assigned for P7_0” from “Description” for “TXD0 to TXD8” in <b>Table 1.17</b></li> </ul>
		28-30	<ul style="list-style-type: none"> <li>• Added <b>Tables 1.19 to 1.21</b> to provide pin definitions and functions for 80-/64-pin packages</li> </ul>
		—	<p><b>Chapter 2. CPU</b></p> <ul style="list-style-type: none"> <li>• Made major text modifications to this chapter</li> </ul>
		35	<p><b>Chapter 3. Memory</b></p> <ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> <li>• Changed RAM size “40” in line 8 of this chapter, to “63”, and address “0000A3FFh” in line 8, to “0000FFFFh”</li> <li>• Added descriptions for 32 Kbyte RAM and 128 Kbyte ROM to <b>Figure 3.1</b></li> </ul>
		36	<p><b>Chapter 4. SFRs</b></p> <ul style="list-style-type: none"> <li>• Changed hexadecimal format of reset values for registers CCR and FMCR in <b>Table 4.1</b>, to binary</li> <li>• Modified the following register name in <b>Table 4.1</b>: “Chip-select Boundary (between n and n + 1) Setting Register”, to “Chip-select n and n + 1 Boundary Setting Register”</li> </ul>
		45	<ul style="list-style-type: none"> <li>• Modified reset values “XXXX XXXXb” and “XXXX 000Xb” for registers U7RB and U8RB in <b>Table 4.10</b>, to “XXXXh”</li> </ul>
		46	<ul style="list-style-type: none"> <li>• Changed expression of register name “Xi Register Yi Register” (i = 0 to 15) and register symbol “XiR, YiR” in <b>Table 4.11</b>, to “Xi Register/Yi Register” and “XiR/YiR”, respectively</li> </ul>
		54	<ul style="list-style-type: none"> <li>• Modified Note 1 for <b>Table 4.19</b></li> </ul>
		55	<ul style="list-style-type: none"> <li>• Modified reset values for IFS0 and IFS2 in <b>Table 4.20</b>; Added IFS7 register and Notes 1 to 3 for 80-/64-pin packages</li> </ul>
		55-57	<ul style="list-style-type: none"> <li>• Modified the following register name in <b>Tables 4.20 to 4.22</b>: “Port Pi_j Port Function Select Register”, to “Port Pi_j Function Select Register”</li> </ul>
		59	<ul style="list-style-type: none"> <li>• Modified register name “DMAi Request Source Select Register 1” in <b>Table 4.24</b>, to “DMAi Request Source Select Register”</li> </ul>
		—	<p><b>Chapter 5. Resets</b></p> <ul style="list-style-type: none"> <li>• Made major text modifications to this chapter</li> </ul>
		60	<ul style="list-style-type: none"> <li>• Modified Note 1 for <b>Figure 5.1</b> to be applied to 100-pin package only</li> </ul>
		61	<ul style="list-style-type: none"> <li>• Deleted the arrow to indicate the period from BCLK oscillation to address output, from <b>Figure 5.2</b></li> </ul>
		62	<ul style="list-style-type: none"> <li>• Modified Notes 1 and 4 for <b>Table 5.1</b> descriptively; Added Notes 2 and 3 for 80-/64-pin packages</li> </ul>
		64	<ul style="list-style-type: none"> <li>• Modified <b>Figure 5.4</b></li> </ul>
		—	<p><b>Chapter 6. Power Management</b></p> <ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> </ul>
		67	<ul style="list-style-type: none"> <li>• Corrected typos “LVDEN”, “LVEIEN”, and “LVDCR” in <b>Figure 6.3</b>, to “VDEN”, “LVDIEN”, and “LVDC”, respectively</li> </ul>
		68	<ul style="list-style-type: none"> <li>• Modified register name “Low voltage detection circuit control register” for LVDC register in <b>Figure 6.4</b>, to “Low voltage detector control register”; Modified Notes 2 and 4 descriptively</li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
			<b>Chapter 7. Processor Mode</b>
		—	<ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> </ul>
		72	<ul style="list-style-type: none"> <li>• Added description for 80-/64-pin packages to <b>7.1</b></li> </ul>
			<b>Chapter 8. Clock Generator</b>
		—	<ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> </ul>
		75	<ul style="list-style-type: none"> <li>• Modified description "peripheral clock" as a clock source, to "peripheral clock source"</li> </ul>
		76	<ul style="list-style-type: none"> <li>• Corrected a typo "Figure 8.3" in line 7 of <b>8.1</b>, to "Figure 8.2"</li> </ul>
		76	<ul style="list-style-type: none"> <li>• Modified the following items in <b>Figure 8.1</b> to show bits indicated in the diagram: "CM06 to CM00", to "CM00 to CM02, CM04, and CM05"; "CM27 to CM20", to "CM20"; "CM31 and CM30", to "CM30 and CM31"; "PM27 to PM20", to "PM26"</li> </ul>
		78	<ul style="list-style-type: none"> <li>• Modified Note 3 for CM0 register in <b>Figure 8.3</b> descriptively; Added description "f2n whose clock source is the main clock", to Note 4</li> </ul>
		86	<ul style="list-style-type: none"> <li>• Added "in the PLC1 register" to "SEO bit" in <b>Figure 8.13</b></li> </ul>
		86, 87	<ul style="list-style-type: none"> <li>• Added description to Note 1 for registers PLC0 and PLC1 in <b>Figures 8.14 and 8.15</b>, respectively</li> </ul>
		87, 90, 91	<ul style="list-style-type: none"> <li>• Deleted description associated with frequency from line 14 below <b>Figure 8.15</b>, line 2 of <b>8.3</b>, and line 2 of <b>8.4</b></li> </ul>
		87	<ul style="list-style-type: none"> <li>• Modified "t<sub>OCS(PLL)</sub>" in line 20 below <b>Figure 8.15</b>, to "t<sub>LOCK(PLL)</sub>"</li> </ul>
		90	<ul style="list-style-type: none"> <li>• Corrected a typo "fc" in line 4 of <b>8.3</b>, to "fC"</li> </ul>
		91	<ul style="list-style-type: none"> <li>• Modified the first paragraph of <b>8.5</b> descriptively; Specified f2n in line 5 of (1) by adding "whose clock source is the peripheral clock source"</li> </ul>
		92	<ul style="list-style-type: none"> <li>• Added "in the 100-pin package" to line 1 of <b>8.6</b></li> </ul>
		94-96	<ul style="list-style-type: none"> <li>• Added description for bit symbols to <b>Figures 8.17 to 8.19</b></li> </ul>
		98	<ul style="list-style-type: none"> <li>• Added description for procedure (6) to <b>8.7.2.2</b></li> </ul>
		100	<ul style="list-style-type: none"> <li>• Modified "the peripheral clock", to "f1, f8, f32, f2n (when the clock source is the peripheral clock source), or fAD" in lines 8 and 9 of <b>8.7.2.4</b></li> </ul>
		101	<ul style="list-style-type: none"> <li>• Changed the following description: "Before executing STOP instruction" in line 5 of <b>8.7.3.1</b>, to "Before entering stop mode"</li> </ul>
		102	<ul style="list-style-type: none"> <li>• Moved previous Table 8.7 with one sentence above the table to 8.7.3.3 as <b>Table 8.8</b></li> </ul>
		104	<ul style="list-style-type: none"> <li>• Deleted whole description for wait mode from <b>8.9.2.1</b>; Deleted description of the first bullet for stop mode from <b>8.9.2.1</b></li> </ul>
			<b>Chapter 9. Bus</b>
		—	<ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> </ul>
		105	<ul style="list-style-type: none"> <li>• Modified expression "memory-expanded mode" in this chapter, to "memory expansion mode"</li> </ul>
		106	<ul style="list-style-type: none"> <li>• Deleted description for frequency and Note 1 in <b>Figure 9.1</b>; Modified description for peripheral data bus "16-bit", to "16-/32-bit"</li> </ul>
		106, 107	<ul style="list-style-type: none"> <li>• Modified peripheral bus width in line 1 of <b>9.2</b>, from "16-bit width" to "16-/32-bit width"</li> </ul>
		106, 107	<ul style="list-style-type: none"> <li>• Added the reference for the theoretical value and maximum frequency to lines 1 and 2 of <b>9.2 and 9.3</b></li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		109	<ul style="list-style-type: none"> <li>• Modified the following register names in <b>Figures 9.4 and 9.5</b>: “CS Output Pin Setting REGISTER i”, to “Chip Select Output Pin Setting Register i”</li> <li>• Modified description for setting the P5_7B bit to 0 in <b>Figure 9.5</b>: “Output RDY from P5_7”, to “RDY input pin”</li> </ul>
		112, 113	<ul style="list-style-type: none"> <li>• Modified description “CBxy” in <b>Figures 9.9 and 9.10</b>, to “CBxx”</li> </ul>
		114	<ul style="list-style-type: none"> <li>• Added “1”s, to bit fields b4 and b12 of EBCi in <b>Figure 9.11</b></li> </ul>
		118	<ul style="list-style-type: none"> <li>• Corrected a typo “WR0” in <b>Table 9.4</b>, to “WR1”</li> <li>• Corrected a typo “WR0” in <b>Table 9.5</b>, to “BC0”</li> </ul>
		121, 122, 126	<ul style="list-style-type: none"> <li>• Added “(i = 0 to 3)” to the title of <b>Figures 9.12, 9.13, and 9.16</b></li> </ul>
		122, 126	<ul style="list-style-type: none"> <li>• Modified “address/data (write)” width in <b>Figures 9.13 and 9.16</b></li> </ul>
		125	<ul style="list-style-type: none"> <li>• Modified the title of <b>Table 9.10</b> descriptively; Added “b”s to respective EBCi setting value in the table; Added description “X: given value”, to below the table</li> </ul>
		128	<ul style="list-style-type: none"> <li>• Changed “the FEBCi register (i = 0 to 3) is”, to “the FEBC0 and/or FEBC3 registers are” in the first and second bullet points of <b>9.5.1</b></li> </ul>
			<b>Chapter 10. Protection</b>
		129	<ul style="list-style-type: none"> <li>• Changed the order of registers for PRC1 and PRC2 in <b>Table 10.1 and Figure 10.1</b></li> </ul>
		130	<ul style="list-style-type: none"> <li>• Deleted “(i = 0 to 7)” from the title of <b>Table 10.2</b></li> </ul>
			<b>Chapter 11. Interrupts</b>
		—	<ul style="list-style-type: none"> <li>• Made major text modifications to this chapter</li> </ul>
		133	<ul style="list-style-type: none"> <li>• Added “MULX” to (2) of <b>11.2</b></li> </ul>
		134	<ul style="list-style-type: none"> <li>• Modified “seven” in line 1 of <b>11.3.1</b>, to “five”</li> </ul>
		137, 138, 140	<ul style="list-style-type: none"> <li>• Added details to Reference in <b>Tables 11.2, 11.3, and 11.5</b></li> </ul>
		142, 143	<ul style="list-style-type: none"> <li>• Changed IR bit name in <b>Figures 11.3 and 11.4</b> from “Interrupt Request Bit” to “Interrupt Request Flag”</li> </ul>
		147	<ul style="list-style-type: none"> <li>• Modified execution time of interrupt sequence in <b>Table 11.7</b></li> </ul>
		149	<ul style="list-style-type: none"> <li>• Moved the priority order of “NMI” in <b>11.8</b> to lower than that of the following interrupts: “Watchdog timer”, “Oscillator stop detection”, and “Low voltage detection”</li> </ul>
		150	<ul style="list-style-type: none"> <li>• Modified “Bits RLVL02 to RLVL00” and “Bits RLVL12 to RLVL10” in <b>Figure 11.8</b>, to “Bits RLVL2 to RLVL0 in the RIPL1 register” and “Bits RLVL2 to RLVL0 in the RIPL2 register”, respectively</li> </ul>
		152	<ul style="list-style-type: none"> <li>• Modified Note 1 for <b>11.11</b> descriptively</li> </ul>
		154	<ul style="list-style-type: none"> <li>• Moved “(i = 0 to 11)” in <b>Figure 11.12</b> to the title</li> </ul>
		155, 156	<ul style="list-style-type: none"> <li>• Modified the following register names: “Intelligent I/O Interrupt Request Register” in <b>Figure 11.13</b>, and “Intelligent I/O Interrupt Enable Register” in <b>Figure 11.14</b>, to “Intelligent I/O Interrupt Request Register i (i = 0 to 11)”, and “Intelligent I/O Interrupt Enable Register i (i = 0 to 11)”, respectively</li> <li>• Changed variables “i”s, “j”s, and “k”s for description of bits in <b>Figures 11.13 and 11.14</b>, to “x”s, “y”s, and “z”s, respectively; Added expression “channel”, to descriptions for TMxyR, POxyR, TMxyE, and POxyE</li> </ul>



Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		157	<ul style="list-style-type: none"> <li>• Corrected a typo “PR2 register” in lines 4 to 5 of <b>11.14.1</b>, to “PM2 register”</li> </ul>
		—	<p><b>Chapter 12. Watchdog Timer</b></p> <ul style="list-style-type: none"> <li>• Made major text modifications to this chapter</li> <li>• Modified description “bus clock”s, to “peripheral bus clock”s</li> </ul>
		158	<ul style="list-style-type: none"> <li>• Modified one of conditions to stop the watchdog timer “hold state” in line 9 of this chapter, to “when the <math>\overline{\text{HOLD}}</math> is driven low”; Changed sample calculation values in lines 14 to 15 of this chapter</li> </ul>
		—	<p><b>Chapter 13. DMAC</b></p> <ul style="list-style-type: none"> <li>• Changed the following principle expressions: “transfer data” to “transfer type”, “transfer data unit” to “transfer size”, “destination address” to “addressing mode”, “fixed” to “non-incrementing addressing”, “forward” to “incrementing addressing”</li> </ul>
		162	<ul style="list-style-type: none"> <li>• Modified the following description: “registers DMiSL and DMiSL2” in line 3 of a paragraph above <b>Figure 13.2</b>, to “the DMiSL register, and in bits DSEL24 to DSEL20 in the DMiSL2 register”</li> </ul>
		163	<ul style="list-style-type: none"> <li>• Modified the following descriptions for DMiSL2 register in <b>Figure 13.3</b>: “DMiSL Register 2” for bits DSEL20 to DSEL24, to “DMiSL2 Register”; “DMA request” for DSR bit, to “DMA transfer request”</li> <li>• Corrected typos “DSEL4” and “DSEL0” in Note 1 for <b>Figure 13.3</b>, to “DESL24” and “DSEL20”, respectively</li> </ul>
		164	<ul style="list-style-type: none"> <li>• Changed the order of Notes in <b>Table 13.2</b></li> </ul>
		166	<ul style="list-style-type: none"> <li>• Changed bit names USAi and UDAi for DMDi register in <b>Figure 13.4</b> and their function descriptively</li> <li>• Deleted the second sentence of Note 2 for DMDi register in <b>Figure 13.4</b>; Added Note 3</li> </ul>
		166, 167	<ul style="list-style-type: none"> <li>• Added “0”s, to bit fields b24 to b31 of registers DCTi and DCRi in <b>Figure 13.5 and 13.6</b>, respectively</li> </ul>
		166	<ul style="list-style-type: none"> <li>• Modified Note 2 for <b>Figure 13.5</b> descriptively; Deleted Note 3</li> </ul>
		169	<ul style="list-style-type: none"> <li>• Added description “(i = 0 to 3)”, to line 3 of <b>13.1</b></li> </ul>
		170	<ul style="list-style-type: none"> <li>• Changed description of “Access Cycles” for “External bus” in <b>Table 13.5</b></li> </ul>
		174	<ul style="list-style-type: none"> <li>• Modified description “channel i” in line 1 of the first bullet point of <b>13.4.1</b>, to “the DMDi register”; Added one sentence to the same bullet point; Deleted whole description of the second and third bullet points; Added two new paragraphs</li> </ul>
		—	<p><b>Chapter 14. DMAC II</b></p> <ul style="list-style-type: none"> <li>• Revised this chapter entirely</li> <li>Changed the following principle expressions: “transfer data” to “transfer type”, “transfer data unit” to “transfer size”, “transfer space” to “transfer memory space”, “transfer direction” to “addressing mode”, “fixed address” to “non-incrementing/constant address”, “forward address” to “incrementing address”, and “end-of-transfer interrupt” to “DMA II transfer complete interrupt”</li> </ul>
		175	<ul style="list-style-type: none"> <li>• Corrected a typo “64 Kbyte-space” in <b>Table 14.1</b>, to “64 Mbyte-space”</li> </ul>
		177	<ul style="list-style-type: none"> <li>• Corrected a typo “Base + 6” in <b>Figure 14.2</b>, to “Base + 8”</li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		180	<ul style="list-style-type: none"> <li>• Corrected a typo “FE0000000h” in the first bullet point of <b>14.3</b>, to “FE000000h”</li> </ul>
		185	<p><b>Chapter 15. Programmable I/O Ports</b></p> <ul style="list-style-type: none"> <li>• Added descriptions for 80-/64pin packages to the first paragraph of this chapter</li> <li>• Modified description “A pin” in <b>Figure 15.1</b>, to “Pin”</li> </ul>
		186	<ul style="list-style-type: none"> <li>• Added “(in the 100-pin package only)” to the title of <b>Figure 15.3</b></li> </ul>
		187	<ul style="list-style-type: none"> <li>• Added Notes 2, 3, and 6 for 80-/64-pin packages to <b>Figure 15.4</b>;</li> <li>Changed description for Note 5 for 100-pin package</li> </ul>
		—	<p><b>Chapter 16. Timers</b></p> <ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> </ul>
		188	<ul style="list-style-type: none"> <li>• Corrected the following typos: “TTA0TGL” in <b>Figure 16.1</b>, to “TA0TGL”, “TAiGH” and “TAiGL”, to “TAiTGH” and “TAiTGL”, respectively</li> </ul>
		190	<ul style="list-style-type: none"> <li>• Corrected a typo “TBiS bit”, to “TAiS” in <b>Figure 16.3</b>; Added description “Event/trigger selection”</li> </ul>
		205	<ul style="list-style-type: none"> <li>• Corrected a typo “FEh” as value of m for “8-bit PWM” in <b>Table 16.5</b>, to “FFh”</li> </ul>
		206	<ul style="list-style-type: none"> <li>• Modified reset value for TAiMR register in <b>Figure 16.16</b> from “0000 000b” to “0000 0000b”</li> </ul>
		208	<ul style="list-style-type: none"> <li>• Changed expression “TBiS bit” in <b>Figure 16.19</b>, to “TBiS”</li> </ul>
		209	<ul style="list-style-type: none"> <li>• Corrected “TM5MR” in the title of <b>Figure 16.21</b>, to “TB5MR”; Changed description for Note 1</li> </ul>
		214	<ul style="list-style-type: none"> <li>• Deleted “0” from bit field b4 for TBiMR register in <b>Figure 16.25</b></li> </ul>
		216	<ul style="list-style-type: none"> <li>• Moved note symbol “(1)” in “Function” for TBiMR register in <b>Figure 16.26</b>, to respective bit name “Measure Mode Select Bit”</li> </ul>
		218, 219	<ul style="list-style-type: none"> <li>• Deleted “(i = 0 to 4)” from line 1 of the first bullet point of <b>16.3.2.3</b> and the second bullet point of <b>16.3.2.4</b></li> </ul>
		219	<ul style="list-style-type: none"> <li>• Added “(i = 0 to 4)” to line 1 of the first bullet point of <b>16.3.2.4</b></li> </ul>
		220	<ul style="list-style-type: none"> <li>• Added description for 80-/64-pin packages to <b>16.3.3.1</b></li> <li>• Changed the variable “i” to “j” in <b>16.3.3</b>; Added “(j = 0 to 5)” to line 1 of <b>16.3.3.2</b></li> </ul>
		—	<p><b>Chapter 17. Three-phase Motor Control Timers</b></p> <ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> </ul>
		223	<ul style="list-style-type: none"> <li>• Modified expression “timer B2 counter” to “timer B2”</li> <li>• Modified “RW”, to “RO” for INV05 bit in <b>Figure 17.2</b>; Modified Notes 6 to 8 for INVC0 register descriptively</li> </ul>
		224	<ul style="list-style-type: none"> <li>• Added “0” to bit field b7 for INVC1 register in <b>Figure 17.3</b></li> </ul>
		226	<ul style="list-style-type: none"> <li>• Changed function of b7 and b6 for IDBi register in <b>Figure 17.5</b> from “Reserved”, to “No register bits”; Modified reset value</li> </ul>
		229, 230, 233	<ul style="list-style-type: none"> <li>• Modified figure title for registers TB2, TB2MR, TB2SC, and TABSR in <b>Figures 17.7 to 17.9 and 17.13</b>, respectively</li> </ul>
		229	<ul style="list-style-type: none"> <li>• Modified reset value for TB2MR register in <b>Figure 17.8</b></li> </ul>
		230	<ul style="list-style-type: none"> <li>• Modified description “odd numbered output of timer A” for PWCON bit in <b>Figure 17.9</b>, to “odd cycle of timer A output”</li> </ul>
		234	<ul style="list-style-type: none"> <li>• Modified <b>17.4</b> descriptively</li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		234, 235	<ul style="list-style-type: none"> <li>• Modified the following expression “high side- and low side- transistors” in line 2 of <b>17.4 and Figure 17.15</b>, to “high-side and low-side transistors”</li> </ul>
		237	<ul style="list-style-type: none"> <li>• Corrected a typo “TA4-1 register” in <b>Figure 17.17</b>, to “TA41 register”</li> </ul>
		238	<ul style="list-style-type: none"> <li>• Modified description “Refer to Figure 16.1” in Note 1 in <b>Figure 17.17</b>, to “Refer to the block diagram of three-phase motor control timers”</li> </ul>
		239	<ul style="list-style-type: none"> <li>• Modified description “The three motor control timer is applicable” in <b>Figure 17.18</b>, to “This bit setting is applicable”</li> </ul>
		239	<ul style="list-style-type: none"> <li>• Added description for 80-/64-pin packages to <b>17.6.3</b></li> </ul>
		—	<b>Chapter 18. Serial Interface</b>
		240	<ul style="list-style-type: none"> <li>• Made major text modifications to this chapter</li> </ul>
		240	<ul style="list-style-type: none"> <li>• Changed the following description: “UART mode” in <b>Table 18.1</b>, to “Asynchronous serial interface mode (UART mode)”</li> </ul>
		241, 242	<ul style="list-style-type: none"> <li>• Deleted “CRS” from the list of bits in the UiC0 register in <b>Figures 18.1 and 18.2</b></li> </ul>
		244	<ul style="list-style-type: none"> <li>• Modified expression “7 (, 8, and 9)-bit transfer data” for “Function” of UiMR register in <b>Figure 18.4</b>, to “7(, 8, and 9)-bit character length”</li> </ul>
		246	<ul style="list-style-type: none"> <li>• Corrected a typo “firs” for UFORM bit in <b>Figure 18.6</b>, to “first”</li> </ul>
		250	<ul style="list-style-type: none"> <li>• Added a “0”, to bit field b7 for UiSMR2 register in <b>Figure 18.11</b>; Modified descriptions for bits SWC and SWC2; Modified the following pin names: “SCL”, to “SCLi”; “SDA”, to “SDAi”</li> </ul>
		251	<ul style="list-style-type: none"> <li>• Modified description “To set the SS” in Note 2 for UiSMR3 register in <b>Figure 18.12</b>, to “To use the SS function”; Corrected a typo “UiCO register” in Note 2, to “UiC0 register”</li> </ul>
		252	<ul style="list-style-type: none"> <li>• Modified description for SWC9 bit in <b>Figure 18.13</b></li> </ul>
		253	<ul style="list-style-type: none"> <li>• Deleted description “and read as undefined value” from “Function” of b15-b9 for UiTB register in <b>Figure 18.15</b></li> </ul>
		258	<ul style="list-style-type: none"> <li>• Corrected the following typos: “000h” for bits 7 to 4 of UiSMR3 register in <b>Table 18.3</b>, to “0000b”; “000h” for bits 2 to 0, to “000b”</li> </ul>
		262, 263	<ul style="list-style-type: none"> <li>• Changed the following register names in <b>Figures 18.22 and 18.23</b>: “UjLCH”, to “UiLCH”; “UjC1”, to “UiC1”; Deleted “(j = 0 to 6)” from Notes 2 and 4 in <b>Figure 18.22</b> and Notes 1 and 2 in <b>Figure 18.23</b></li> </ul>
		265	<ul style="list-style-type: none"> <li>• Modified description for “Overrun error” of “Error detection” in Table 18.5</li> </ul>
		274	<ul style="list-style-type: none"> <li>• Deleted the following description: “signals are” from “Transmit/receive clock” of <b>Table 18.9</b>; Modified description for Note 1</li> </ul>
		275	<ul style="list-style-type: none"> <li>• Added bits ACKC and ACKD to <b>Figure 18.31</b></li> </ul>
		276	<ul style="list-style-type: none"> <li>• Added description for UiIRS bit to UiC1 register in <b>Table 18.10</b>; Modified bits “7 to 4” in the UiC1 register, to “7 to 5”</li> </ul>
		277	<ul style="list-style-type: none"> <li>• Modified the following descriptions in <b>Table 18.11</b>: “UART transmit/ UART receive interrupt”, “Pi_jS register (i, j = 0 to 7)”, and “I/O port is selected”, to “Transmit/receive interrupt”, “Port Pi register (i = 0 to 7)”, and “I/O port is selected by output function select registers”, respectively</li> </ul>
		282	<ul style="list-style-type: none"> <li>• Modified description for SCLi line in <b>Figure 18.35</b></li> </ul>
		287	<ul style="list-style-type: none"> <li>• Corrected a typo “SS pin” in title of <b>Figure 18.37</b>, to “SSi pin”</li> </ul>
		290	<ul style="list-style-type: none"> <li>• Added description for 80-/64-pin packages to <b>18.5.2</b></li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		291	<ul style="list-style-type: none"> <li>• Deleted whole description from the third bullet point of <b>18.5.3.2</b></li> </ul>
		—	<b>Chapter 19. A/D Converter</b>
		293	<ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> <li>• Changed Note 3 of <b>Table 19.1</b> for 80-/64-pin packages; Added Notes 4 and 5</li> </ul>
		294	<ul style="list-style-type: none"> <li>• Added Note 2 to <b>Figure 19.1</b> for 64-pin package</li> </ul>
		299	<ul style="list-style-type: none"> <li>• Added boxes for bit symbols for AD0i registers in <b>Figure 19.7</b>;</li> <li>Changed description for b15-b10</li> </ul>
		300-306	<ul style="list-style-type: none"> <li>• Corrected a typo "A/D00"s in <b>Tables 19.2 to 19.8</b>, to "AD00"</li> </ul>
		307	<ul style="list-style-type: none"> <li>• Modified description "AD0j register (j = 0 to 7)" in line 2 of <b>19.2.1</b>, to "AD0i register (i = 0 to 7)"</li> </ul>
		308	<ul style="list-style-type: none"> <li>• Added "(refer to Table 19.10)" to line 2 of <b>19.2.5</b></li> </ul>
		311	<ul style="list-style-type: none"> <li>• Changed description for the third bullet of <b>19.3.1</b> to add 80-/64-pin packages</li> </ul>
		312	<ul style="list-style-type: none"> <li>• Modified description "the respective AD0i register" in the seventh bullet point of <b>19.3.2</b>, to "the AD00 register"</li> <li>• Deleted "(i = 0 to 7)" from the eighth bullet point of <b>19.3.2</b></li> </ul>
		313	<b>Chapter 20. D/A Converter</b>
		314	<ul style="list-style-type: none"> <li>• Added description for 80-/64-pin packages to the first paragraph of this chapter</li> <li>• Modified description for Note 2 in <b>Figure 20.4</b></li> </ul>
		—	<b>Chapter 21. CRC Calculator</b>
		317	<ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> <li>• Modified <b>Figure 21.4</b> visually</li> </ul>
		—	<b>Chapter 22. X-Y Conversion</b>
		—	<ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> </ul>
		—	<b>Chapter 23. Intelligent I/O</b>
		323-325	<ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> <li>• Moved "(j = 0 to 7)" in <b>Figures 23.1 to 23.3</b> to the titles</li> </ul>
		323	<ul style="list-style-type: none"> <li>• Added description for BT0R to <b>Figure 23.1</b></li> </ul>
		324	<ul style="list-style-type: none"> <li>• Added description for BT1R to <b>Figure 23.2</b></li> </ul>
		325	<ul style="list-style-type: none"> <li>• Added description for bits BT2R, PO2jR, IE0R to IE2R, SIO2TR, and SIO2RR to <b>Figure 23.3</b>; Deleted note symbol "(3)"</li> </ul>
		328, 330	<ul style="list-style-type: none"> <li>• Added a "0" to bit field b7 for GiBCR1 register in <b>Figure 23.6</b> and bit field b3 for BTSR register in <b>Figure 23.8</b></li> </ul>
		329	<ul style="list-style-type: none"> <li>• Deleted description "(Refer to Figure 23.14 for details on the G2P00 register)" from Note 1 for G2BCR1 register in <b>Figure 23.7</b></li> </ul>
		343	<ul style="list-style-type: none"> <li>• Changed "IIOi_j pin function" in <b>Table 23.4</b>, to "IIOi_j input pin function"; Moved "(j = 0 to 7)" for "Trigger input polarity", to the table title</li> </ul>
		344	<ul style="list-style-type: none"> <li>• Moved "j = 0 to 7; k = 6, 7" below <b>Table 23.5</b> to the title</li> </ul>
		344, 345	<ul style="list-style-type: none"> <li>• Moved "j = 0 to 7" in <b>Figures 23.22 and 23.23</b> to the titles</li> </ul>
		346	<ul style="list-style-type: none"> <li>• Moved "j = 6, 7" in <b>Figure 23.24</b> to the title</li> </ul>
		347	<ul style="list-style-type: none"> <li>• Modified the following description: "The GiPOCRj register (i = 0, 1) does not have it" in Note 1 for <b>Table 23.6</b>, to "Neither the G0POCRj nor G1POCRj register has it"</li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		347	<ul style="list-style-type: none"> <li>• Moved “j = 0 to 7” below <b>Table 23.6</b> to the title</li> </ul>
		348, 350, 352	<ul style="list-style-type: none"> <li>• Added description “(or OUTC2_j pin for Group 2)” to line 1 of <b>23.3.1, 23.3.2, and 23.3.3</b></li> </ul>
		348, 350, 353	<ul style="list-style-type: none"> <li>• Modified “IIOi_j pin function” in <b>Table 23.7</b>, and “IIOi_j pin function (output)” in <b>Tables 23.8 and 23.9</b>, to “IIOi_j output pin (or OUTC2_j pin for Group 2) function”; Added description “(or OUTC2_j pin for Group 2)” after “IIOi_j output pin” in “Selectable functions”</li> </ul>
		355, 356 357, 359	<ul style="list-style-type: none"> <li>• Modified the variable “i” to “2” in <b>23.3.4, Tables 23.10, 23.12, 23.13, and Figure 23.28</b></li> </ul>
		358	<ul style="list-style-type: none"> <li>• Added description “in the G2RTP register” to “RTPj bit” in <b>Figure 23.30</b></li> </ul>
		360	<ul style="list-style-type: none"> <li>• Corrected typos “PO2_xR” in <b>Figure 23.32</b>, to “PO2xR”</li> </ul>
			<b>Chapter 24. I/O Pins</b>
		—	<ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> </ul>
		370	<ul style="list-style-type: none"> <li>• Added Notes 2 and 3 of PDi register for 80-/64-pin packages to <b>Figure 24.2</b></li> </ul>
		371	<ul style="list-style-type: none"> <li>• Added “b” to binary form in <b>Table 24.1</b></li> </ul>
		372, 373, 375, 377-379, 383	<ul style="list-style-type: none"> <li>• Added description for 80-/64-pin packages to respective paragraph below <b>Figures 24.3, 24.4, 24.6, 24.8, 24.9, 24.10, and 24.14</b></li> </ul>
		375, 377, 385, 387-389	<ul style="list-style-type: none"> <li>• Added <b>Figures 24.6, 24.8, 24.17, 24.19, 24.21, 24.22</b> to provide registers P2_iS, P3_iS, IFS0, IFS2, and IFS7 for 80-/64-pin packages, respectively</li> </ul>
		385, 386	<ul style="list-style-type: none"> <li>• Modified configuration and reset value of IFS0 register for 100-pin package in <b>Figure 24.16</b> and IFS2 register for 100-pin package in <b>Figure 24.18</b></li> </ul>
		385	<ul style="list-style-type: none"> <li>• Modified expression “TAiIN input” in <b>Figure 24.16</b>, to “TAiIN”</li> </ul>
		391	<ul style="list-style-type: none"> <li>• Modified description “P4_4 to P4_7” in Note 1 for PUR1 register in <b>Figure 24.24</b>, to “P4_0 to P4_3”</li> </ul>
		394	<ul style="list-style-type: none"> <li>• Added the following description: “(in the 100-pin package)” to the P9_1 pin in <b>Table 24.2</b>; Changed expression “negated” in Note 2, to “released”; Added Notes 4 and 5 for 80-/64-pin packages</li> </ul>
		395	<ul style="list-style-type: none"> <li>• Added the following description: “(in the 100-pin package only)” to the table title of <b>Table 24.3</b>; Changed expression “negated” in Note 2, to “released”</li> </ul>
			<b>Chapter 25. Flash Memory</b>
		—	<ul style="list-style-type: none"> <li>• Made major text modifications to this chapter</li> </ul>
		397, 399	<ul style="list-style-type: none"> <li>• Changed expressions “write” and “rewrite”, to “program” when this word is used in combination with “erase”</li> </ul>
		398	<ul style="list-style-type: none"> <li>• Modified <b>Tables 25.1 and 25.3</b> descriptively</li> </ul>
		400	<ul style="list-style-type: none"> <li>• Changed <b>Figure 25.1</b> to add 128 KB version</li> </ul>
		401	<ul style="list-style-type: none"> <li>• Added new function of ID code protection for 80-/64-pin packages to <b>25.2.3</b></li> </ul>
		401	<ul style="list-style-type: none"> <li>• Added <b>Tables 25.5 and 25.6, 25.2.4, and 25.2.5</b> for reserved ID codes, forcible erase function, and standard serial I/O mode disable function</li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		402	<ul style="list-style-type: none"> <li>• Modified descriptions “Flash rewrite address bus” and “Flash rewrite data bus” in <b>Figure 25.3</b>, to “Flash memory rewrite address bus” and “Flash memory rewrite data bus”, respectively</li> </ul>
		403	<ul style="list-style-type: none"> <li>• Modified <b>Table 25.7</b> descriptively</li> </ul>
		403, 415, 430	<ul style="list-style-type: none"> <li>• Corrected a typo “read allay mode” in <b>Table 25.7</b>, in line 2 of <b>25.3.4</b>, in line 2 below <b>Table 25.18</b>, and in line 5 and 17 of <b>25.6.5</b> to “read array mode”</li> </ul>
		403	<ul style="list-style-type: none"> <li>• Modified description “Figure 25.10” in line 4 below <b>Table 25.7</b>, to “Figure 25.11”</li> </ul>
		404	<ul style="list-style-type: none"> <li>• Added a note symbol “(2)” to FEW bit in <b>Figure 25.4</b></li> </ul>
		405	<ul style="list-style-type: none"> <li>• Modified register name “Flash Memory Control Register” in <b>Figure 25.5</b>, to “Flash Memory Control Register i”</li> </ul>
		406	<ul style="list-style-type: none"> <li>• Modified “Bit name” and “Function” for FCA bit in <b>Figure 25.7</b></li> </ul>
		407	<ul style="list-style-type: none"> <li>• Corrected a typo “(b7-4)” for FMR1 register in <b>Figure 25.8</b>, to “(b7-b4)”</li> </ul>
		414	<ul style="list-style-type: none"> <li>• Added a note symbol “(3)” to the command “Enter read protect bit status mode” in <b>Table 25.15</b></li> </ul>
		415	<ul style="list-style-type: none"> <li>• Corrected typos “Read lock bit mode” and “Read protect bit mode” in lines 4 and 5 of <b>25.3.4</b>, to “Read lock bit status mode” and “Read protect bit status mode”, respectively</li> </ul>
		417	<ul style="list-style-type: none"> <li>• Corrected a typo “b5-0” in <b>Tables 25.17 and 25.18</b>, to “b5-b0”</li> <li>• Added description “Verify that the FCA bit in the FMR0 register is 0 just before executing the final command” to <b>25.3.5.4</b></li> <li>• Modified the position of flowchart symbol “decision making and branching” for FCA bit in <b>Figure 25.14</b> and its description, to “FCA bit in the FMR0 register is 0”</li> </ul>
		418	<ul style="list-style-type: none"> <li>• Corrected a typo “FFFF800h” in line 3 of <b>25.3.5.5</b>, to “FFFFFF800h”</li> </ul>
		418-421	<ul style="list-style-type: none"> <li>• Modified description of flowchart symbol “decision making and branching” for FCA bit in <b>Figures 25.15 to 25.18</b>, to “FCA bit in the FMR0 register is 0”</li> </ul>
		418, 420, 421	<ul style="list-style-type: none"> <li>• Added description “verify that the FCA bit in the FMR0 register is 0” to <b>25.3.5.5, 25.3.5.7, and 25.3.5.9</b></li> </ul>
		419	<ul style="list-style-type: none"> <li>• Added description “write 0077h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, then” to <b>25.3.5.6</b></li> </ul>
		423	<ul style="list-style-type: none"> <li>• Added Notes 1 and 2 for <b>Table 25.20</b></li> </ul>
		424	<ul style="list-style-type: none"> <li>• Modified description “I/O port” for “P0_0 to...P4_7” in <b>Table 25.21</b>, to “Input port”</li> </ul>
		425	<ul style="list-style-type: none"> <li>• Added <b>Table 25.22</b> to provide pin definitions and functions for 80-/64-pin packages</li> </ul>
		427, 429	<ul style="list-style-type: none"> <li>• Added <b>Figures 25.21 and 25.23</b> to provide circuit applications for 80-64-pin packages</li> </ul>
		430	<ul style="list-style-type: none"> <li>• Modified description “bits EWM and LBD in the FMR1 register” in the third bullet point of EW1 Mode in <b>25.6.5</b>, to “the EWM bit in the FMR0 register” and “the LBD bit in the FMR1 register”</li> </ul>
			<b>Chapter 26. Electrical Characteristics</b>
		432	<ul style="list-style-type: none"> <li>• Added Notes 2 and 3 for 80-/64-pin packages to <b>Table 26.1</b></li> </ul>
		433	<ul style="list-style-type: none"> <li>• Added Notes 2, 4, and 5 for 80-/64-pin packages to <b>Table 26.2</b></li> </ul>
		434	<ul style="list-style-type: none"> <li>• Added Note 2 for <b>Table 26.3</b></li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		435	• Added Note 3 for 80-/64-pin packages to <b>Table 26.4</b>
		437	• Corrected a typo “ $dV_{r(VCC)2}/dt$ ” in <b>Table 26.6</b> , to “ $dV_{r(VCC)2}/dt$ ”
		438	• Modified the following “Characteristic” name: “RAM data retention”, to “RAM data retention voltage” in <b>Table 26.7</b> ; Deleted “voltage” from the description of Note 1
		439	• Deleted measurement condition for power supply circuit timing characteristics in <b>Table 26.9</b>
			• Modified self-consuming current “ $V_{CC}$ ” in <b>Table 26.11</b> , to “ $V_{CC1}$ ”
		440	• Changed typical value and maximum value for $f_{SO(PLL)}$ in <b>Table 26.12</b> , to “55” and “80”, respectively
			• Changed description order of $t_{rec(STOP)}$ and $t_{rec(WAIT)}$ in <b>Table 26.13 and Figure 26.4</b>
		442, 443, 454, 455	• Added Notes 1 and 2 for 80-/64-pin packages to <b>Tables 26.15, 26.16, 26.38, and 26.39</b>
		444, 456	• Added “XIN” as “Active” to first, third, and sixth rows of <b>Tables 26.17 and 26.40</b>
		445, 457	• Added Note 1 for 80-/64-pin packages to <b>Tables 26.18 and 26.41</b>
		450, 462	• Changed maximum value for $t_{h(C-D)}$ “30” in <b>Tables 26.30 and 26.53</b> , to “80”
			• Modified minimum value for $t_{w(ADH)}$ in <b>Tables 26.31 and 26.54</b> , to “ $\frac{3}{\Phi_{AD}}$ ”
		452, 464	• Deleted a “(1)” from “Value” in <b>Tables 26.35 and 26.58</b>
			• Modified “Characteristics” for $t_{su(S-ALE)}$ in <b>Tables 26.35 and 26.58</b> , from “Chip-select hold time for ALE” to “Chip-select setup time for ALE”
			• Changed minimum value of $t_{h(W-D)}$ “0” in <b>Tables 26.35 and 26.58</b> , to “ $0.5 \times t_{c(Base)}$ ”
		453, 465	• Modified “Characteristics” for $t_{h(C-Q)}$ in <b>Tables 26.36 and 26.59</b> , from “TXDi hold time” to “TXDi output hold time”
		457	• Added a skipped word “error” after “Differential non-linearity” in <b>Table 26.41</b>
		459	• Corrected typos “ $t_{w(H)}$ ”, “ $t_{w(L)}$ ”, “ $t_r$ ”, and “ $t_f$ ” in <b>Table 26.43</b> , to “ $t_{w(XH)}$ ”, “ $t_{w(XL)}$ ”, “ $t_r(X)$ ”, and “ $t_f(X)$ ”, respectively
			<b>Chapter 27. Usage Notes</b>
		—	• Made minor text modifications to this chapter
		470	• Moved description for <b>27.2</b> to <b>27.2.1</b>
		474	• Deleted whole description for wait mode from <b>27.3.2.1</b> ; Deleted description of the first bullet for stop mode from <b>27.3.2.1</b>
		475	• Changed “the FEBCi register (i = 0 to 3)”, to “the FEBC0 and/or FEBC3 registers” in the first and second bullet points in <b>27.4.1</b>
		476	• Corrected a typo “PR2 register” in lines 4 to 5 of <b>27.5.1</b> , to “PM2 register”

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		477	<ul style="list-style-type: none"> <li>• Modified description “channel i” in line 1 of the first bullet point of <b>27.6.1</b>, to “the DMDi register”; Added one sentence to the same bullet point; Deleted whole description of the second and third bullet points; Added two new paragraphs</li> </ul>
		478, 479	<ul style="list-style-type: none"> <li>• Deleted “(i = 0 to 4)” from line 1 of the first bullet point of <b>27.7.2.3</b> and the second bullet point of <b>27.7.2.4</b></li> </ul>
		479	<ul style="list-style-type: none"> <li>• Added “(i = 0 to 4)”, to line 1 of the first bullet point of <b>27.7.2.4</b></li> </ul>
		480	<ul style="list-style-type: none"> <li>• Added description for 80-/64-pin packages to <b>27.7.3.1</b></li> <li>• Changed the variable “i” to “j” in <b>27.7.3</b>; Added “(j = 0 to 5)” to line 1 of <b>27.7.3.2</b></li> </ul>
		481	<ul style="list-style-type: none"> <li>• Added description for 80-/64-pin packages to <b>27.8.3</b></li> </ul>
		482	<ul style="list-style-type: none"> <li>• Added description for 80-/64-pin packages to <b>27.9.2</b></li> </ul>
		483	<ul style="list-style-type: none"> <li>• Deleted whole description from the third bullet point of <b>27.9.3.2</b></li> </ul>
		484	<ul style="list-style-type: none"> <li>• Changed description for the third bullet of <b>27.10.1</b> to add 80-/64-pin packages</li> </ul>
		485	<ul style="list-style-type: none"> <li>• Modified description “the respective AD0i register” in the seventh bullet point of <b>27.10.2</b>, to “the AD00 register”</li> <li>• Deleted “(i = 0 to 7)” from the eighth bullet point of <b>27.10.2</b></li> </ul>
		486	<ul style="list-style-type: none"> <li>• Modified description “bits EWM and LBD in the FMR1 register” in the third bullet point of EW1 Mode in <b>27.11.5</b>, to “the EWM bit in the FMR0 register” and “the LBD bit in the FMR1 register”</li> <li>• Corrected a typo “read allay mode” in <b>27.11.5</b>, “read array mode”</li> </ul>
			<b>Appendix 1</b>
		488, 489	<ul style="list-style-type: none"> <li>• Added figures for 100-pin plastic molded LGA, and 80-/64-pin plastic molded LQFP packages</li> </ul>
1.20	Sep 26, 2011	—	Ninth edition released
		—	The manual in general <ul style="list-style-type: none"> <li>• Applied new Renesas templates and formats to the manual</li> <li>• Changed company name to “Renesas Electronics Corporation” and changed related descriptions due to business merger of Renesas Technology Corporation and NEC Electronics Corporation</li> <li>• Modified expressions “version N” and “version D” to “N version” and “D version”, respectively (under Chapters 1 and 26)</li> </ul>
		—	<b>Chapter 1. Overview</b>
		2, 4, 6	<ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> <li>• Modified the following expressions in <b>Tables 1.1, 1.3, and 1.5</b>: “Main clock oscillator stop/re-oscillation detection” to “Main clock oscillator stop/restart detection”, and “inputs/outputs” to “I/O ports”</li> </ul>
		3	<ul style="list-style-type: none"> <li>• Deleted Note 1 from <b>Table 1.2</b></li> </ul>
		4, 6	<ul style="list-style-type: none"> <li>• Deleted memory expansion mode and microprocessor mode from the operating mode of the CPU in <b>Tables 1.3 and 1.5</b></li> </ul>
		5, 7	<ul style="list-style-type: none"> <li>• Deleted Note 2 from <b>Tables 1.4 and 1.6</b></li> </ul>
		8	<ul style="list-style-type: none"> <li>• Completed “under development” phase of R5F6411EDFN in <b>Table 1.7</b></li> </ul>
		10-12	<ul style="list-style-type: none"> <li>• Deleted Note 1 from <b>Figures 1.2 to 1.4</b></li> </ul>
		13, 18, 21	<ul style="list-style-type: none"> <li>• Changed order of signals in <b>Figures 1.5, 1.7, and 1.8</b></li> </ul>



Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		15, 19, 22 24	<ul style="list-style-type: none"> <li>• Changed order of timer pins “TB5IN/TA0IN” in <b>Tables 1.8, 1.11, and 1.13</b> to “TA0IN/TB5IN”</li> <li>• Modified expression “fC” in <b>Table 1.15</b> to “low speed clocks”</li> </ul>
		—	<p><b>Chapter 2. CPU</b></p> <ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> </ul>
		32	<ul style="list-style-type: none"> <li>• Corrected a typo “R3R0” in line 3 of <b>2.1.1</b> to “R3R1”</li> </ul>
		—	<p><b>Chapter 3. Memory</b></p> <ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> </ul>
		—	<p><b>Chapter 4. SFRs</b></p> <ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> </ul>
		41, 42, 44 41, 44	<ul style="list-style-type: none"> <li>• Changed hexadecimal format of reset values for registers G1BCR0, G2BCR0, and G0BCR0 in <b>Tables 4.6, 4.7, and 4.9</b> to binary</li> <li>• Changed register name “Group i Timer Measurement Prescaler Register” in <b>Tables 4.6 and 4.9</b> to “Group i Time Measurement Prescaler Register”</li> </ul>
		43	<ul style="list-style-type: none"> <li>• Modified expression “IE Bus” in <b>Table 4.8</b> to “IEBus”</li> </ul>
		46	<ul style="list-style-type: none"> <li>• Modified expression “XY Control Register” in <b>Table 4.11</b> to “X-Y Control Register”</li> </ul>
		48	<ul style="list-style-type: none"> <li>• Changed register name “UART2 Transmission/Receive Mode Register” and “Increment/Decrement Counting Select Register” in <b>Table 4.13</b> to “UART2 Transmit/Receive Mode Register” and “Increment/Decrement Select Register”, respectively; Changed hexadecimal format of reset values for registers TABSR, ONSF, and TRGSR to binary</li> </ul>
		50	<ul style="list-style-type: none"> <li>• Changed reset value “X00X X000b” for AD0CON2 register in <b>Table 4.15</b> to “XX0X X000b”</li> </ul>
		59	<ul style="list-style-type: none"> <li>• Changed register name “External Interrupt Source Select Register i” in <b>Table 4.24</b> to “External Interrupt Request Source Select Register i”</li> </ul>
		—	<p><b>Chapter 5. Resets</b></p> <ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> </ul>
		60	<ul style="list-style-type: none"> <li>• Changed expression “operating level” in (2) of B. in <b>5.1</b> to “operating voltage”</li> </ul>
		—	<p><b>Chapter 6. Power Management</b></p> <ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> </ul>
		65	<ul style="list-style-type: none"> <li>• Modified descriptions “main clock oscillator active” and “PLL clock oscillator active” in Note 2 of <b>Figure 6.2</b> to “main clock oscillator enabled” and “PLL oscillator enabled”</li> </ul>
		68	<ul style="list-style-type: none"> <li>• Modified bit name of VDEN bit in <b>Figure 6.4</b> to “Low Voltage Detector Enable Bit”; Modified its function descriptions “low voltage detection disabled” and “low voltage detection enabled” to “low voltage detector disabled” and “low voltage detector enabled”, respectively</li> </ul>
		—	<p><b>Chapter 7. Processor Mode</b></p> <ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> </ul>
		73	<ul style="list-style-type: none"> <li>• Corrected a typo “44044h” in <b>Figure 7.1</b> to “40044h”</li> </ul>
		74	<ul style="list-style-type: none"> <li>• Deleted addresses “00008000h” and “FFF80000h” from <b>Figure 7.2</b></li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
			<b>Chapter 8. Clock Generator</b>
		—	<ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> </ul>
		76	<ul style="list-style-type: none"> <li>• Modified expression “fC” in <b>Figure 8.1</b> to “Low speed clock”; Modified “low speed clock” associated items; Added BCS bit</li> </ul>
		77	<ul style="list-style-type: none"> <li>• Deleted the last sentence from Note 2 in <b>Figure 8.2</b>; Added description of BCS bit setting to Note 6</li> </ul>
		78	<ul style="list-style-type: none"> <li>• Modified expression “fC” in <b>Figure 8.3</b> to “a low speed clock”; Modified function descriptions of the CM05 bit to “main clock oscillator enabled” and “main clock oscillator disabled”; Added Note 8</li> </ul>
		79	<ul style="list-style-type: none"> <li>• Modified CM10 bit name “PLL Clock Oscillator Stop Bit” in <b>Figure 8.4</b> to “PLL Oscillator Stop Bit”; Modified its function descriptions to “PLL oscillator enabled” and “PLL oscillator disabled”; Modified description of Note 2; Added Note 4</li> </ul>
		80	<ul style="list-style-type: none"> <li>• Modified function description of CM20 bit in <b>Figure 8.5</b> to “Disable oscillator stop detection” when it is 0 and “Enable oscillator stop detection” when it is 1; Corrected “CM02 bit” in Note 3 to “CM20 bit”</li> </ul>
		82	<ul style="list-style-type: none"> <li>• Modified description of Note 1 in <b>Figure 8.6</b></li> <li>• Modified the following function descriptions in Note 3 of <b>Figure 8.9</b>: “start main clock oscillator running” for CM05 bit to “main clock oscillator enabled/disabled” and “start PLL clock oscillator running” for CM10 bit to “PLL oscillator enabled/disabled”; Added Note 5</li> </ul>
		83	<ul style="list-style-type: none"> <li>• Added description to Note 1 in <b>Figure 8.10</b></li> </ul>
		87	<ul style="list-style-type: none"> <li>• Modified SEO bit name and its function description in <b>Figure 8.15</b></li> </ul>
		90	<ul style="list-style-type: none"> <li>• Deleted the last line in parenthesis in <b>8.2</b></li> </ul>
		92, 100, 103	<ul style="list-style-type: none"> <li>• Added description of stopping main clock to lines 6 to 7 of <b>8.2.1</b></li> <li>• Modified expression “fC” in line 1 of <b>8.6</b> to “Low speed clocks” and “fC” in <b>Tables 8.3 to 8.5 and 8.7</b> to “a low speed clock”</li> </ul>
		93	<ul style="list-style-type: none"> <li>• Revised the entire paragraph of <b>8.7</b></li> </ul>
		94	<ul style="list-style-type: none"> <li>• Revised <b>8.7.1</b> entirely</li> </ul>
		96-98	<ul style="list-style-type: none"> <li>• Moved <b>Figures 8.17 to 8.19</b> into <b>8.7.1</b> and revised them entirely</li> </ul>
		96	<ul style="list-style-type: none"> <li>• Corrected a typo “f(XPLL)” in the third row of <b>Figure 8.17</b> to “f(PLL)”; Deleted Note 4</li> </ul>
		97	<ul style="list-style-type: none"> <li>• Corrected a typo “CM0 = 1” in the fifth row of <b>Figure 8.18</b> to “CM05 = 1”; Deleted Note 3</li> </ul>
		98	<ul style="list-style-type: none"> <li>• Corrected “CM31 = 1” in the first row and “CM10 = 0” for “Low speed mode” in the second row of <b>Figure 8.19</b> to “CM31 = 0” and “CM10 = 1”, respectively; Deleted Note 3</li> </ul>
		99	<ul style="list-style-type: none"> <li>• Modified description in lines 2 to 3 in <b>8.7.2</b></li> <li>• Added description “in wait mode” to line 3 of <b>8.7.2.1</b></li> <li>• Changed expression “Before executing WAIT instruction” in <b>8.7.2.2</b> to “Steps before entering wait mode”; Updated procedures for entering wait mode based on the technical update “TN-16C-A182A/E”</li> </ul>
		101	<ul style="list-style-type: none"> <li>• Revised <b>8.7.2.4</b></li> </ul>
		102	<ul style="list-style-type: none"> <li>• Modified description of line 1 in <b>8.7.3</b></li> <li>• Updated procedures for entering stop mode in <b>8.7.3.1</b> based on the technical update “TN-16C-A182A/E”</li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		103	<ul style="list-style-type: none"> <li>• Added interrupt numbers 0 to 63 to line 2 of <b>8.7.3.3</b> as factors of exiting stop mode</li> </ul>
		—	<p><b>Chapter 9. Bus</b></p> <ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> </ul>
		108	<ul style="list-style-type: none"> <li>• Changed mathematical symbol “&lt;” in formulas in <b>9.3.1</b> to “≤”</li> </ul>
		110	<ul style="list-style-type: none"> <li>• Deleted Note 2 from <b>Figures 9.4 and 9.5</b></li> </ul>
		111	<ul style="list-style-type: none"> <li>• Changed minimum value for registers CB01 and CB12 in <b>Figures 9.6 and 9.7</b> to “02h”</li> </ul>
		111, 112	<ul style="list-style-type: none"> <li>• Changed maximum value for registers CB12 and CB23 in <b>Figures 9.7 and 9.8</b> to “F8h” in memory expansion mode and “FFh” in microprocessor mode</li> </ul>
		114	<ul style="list-style-type: none"> <li>• Modified description of Note 2 in <b>Figures 9.6 to 9.8</b></li> </ul>
		114	<ul style="list-style-type: none"> <li>• Added description “(except for the <math>\overline{CS0}</math> signal)” to Note 1 in <b>Figure 9.10</b></li> </ul>
		115	<ul style="list-style-type: none"> <li>• Modified bit names of bits ESUR1 and ESUR0, bits ESUW1 and ESUW0, bits EWR1 and EWR0, and bits EWW1 and EWW0 in <b>Figure 9.11</b> to “Address Setup Cycles Before Read Setting Bit”, “Address Setup Cycles Before Write Setting Bit”, “Read Pulse Width Setting Bit”, and “Write Pulse Width Setting Bit”, respectively</li> </ul>
		117	<ul style="list-style-type: none"> <li>• Added EXMPX bit value to column of each bus format in <b>Table 9.2</b>; Modified function of P4_0 to P4_3 for memory expansion mode to I/O ports only</li> </ul>
		120	<ul style="list-style-type: none"> <li>• Modified descriptions “(address setup before <math>\overline{RD}</math>)”, “(address setup before <math>\overline{WR}</math>)”, “(<math>\overline{RD}</math> pulse width)”, and “(<math>\overline{WR}</math> pulse width)” in the second paragraph of <b>9.3.5</b> to “(address setup cycles before read)”, “(address setup cycles before write)”, “(read pulse width)”, and “(write pulse width)”, respectively</li> </ul>
		124	<ul style="list-style-type: none"> <li>• Added period of address becoming undefined to “(1) 8-bit data bus” in <b>Figure 9.14</b>; Added Note 2</li> </ul>
		—	<p><b>Chapter 10. Protection</b></p> <ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> <li>• Changed subchapter titles in this chapter</li> </ul>
		—	<p><b>Chapter 11. Interrupt</b></p> <ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> </ul>
		133	<ul style="list-style-type: none"> <li>• Modified description of Note 1 in <b>Figure 11.1</b></li> </ul>
		134	<ul style="list-style-type: none"> <li>• Modified descriptions in the second paragraph in (5) of <b>11.2</b></li> </ul>
		136, 137, 144	<ul style="list-style-type: none"> <li>• Modified description of the jump operation in <b>11.5, Table 11.1, and below Figure 11.4</b></li> </ul>
		147	<ul style="list-style-type: none"> <li>• Moved description of Note 1 to (2) and (4) of <b>11.6.4</b></li> </ul>
		148	<ul style="list-style-type: none"> <li>• Modified Note 1 of <b>Table 11.7</b></li> </ul>
		151	<ul style="list-style-type: none"> <li>• Changed expression “DMAC II” in <b>Figure 11.8</b> to “DMA II transfer complete”; Deleted “Bits RLVL2 to RLVL0 in the RIPL2 register” and associated signal lines</li> </ul>
		153	<ul style="list-style-type: none"> <li>• Deleted Note 1 from <b>Figure 11.10</b></li> </ul>
		155	<ul style="list-style-type: none"> <li>• Corrected register symbol “IIOiE” in <b>11.13</b> to “IIOiIE”</li> </ul>
		156	<ul style="list-style-type: none"> <li>• Changed function description of b0 in <b>Figure 11.13</b>; Modified Note 3</li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		—	<b>Chapter 12. Watchdog Timer</b>
		159	• Modified wording and enhanced description in this chapter
		160	• Modified description in lines 3 to 4 in <b>12. Watchdog Timer</b>
			• Added Note 1 to <b>Figure 12.2</b>
		—	<b>Chapter 13. DMAC</b>
		162	• Modified wording and enhanced description in this chapter
			• Modified description of timer-associated interrupt request in “DMA request sources” in <b>Table 13.1</b> ; Modified description “more than 00000001h” in “DMA transfer start-up” to “other than 00000000h”
		170	• Modified description in lines 1 to 2 of <b>13.1</b>
		—	<b>Chapter 14. DMAC II</b>
		176, 182	• Modified wording and enhanced description in this chapter
			• Changed expression “calculation transfer” in <b>Table 14.1, 14.3.3, and 14.4.3</b> to “calculation result transfer”
		176	• Corrected source address “FFFFFFFh” in Note 1 of <b>Table 14.1</b> to “FFFFFFFh”
		180	• Corrected bit name “IIRLT” in the fifth bullet point of <b>14.1</b> to “IRLT”
		181	• Changed bit names of OPER bit and bits CNT2 to CNT0 in <b>Figure 14.3</b> to “Calculation Result Transfer Select Bit” and “Number of Transfers Setting Bit”, respectively
			• Modified addresses “001FFFFFFh”, “00200000h”, and “00000000h” in the second paragraph of <b>14.3.1</b> to “01FFFFFFh”, “02000000h”, and “FE000000h”, respectively
		—	<b>Chapter 15. Programmable I/O Ports</b>
		185	• Modified wording and enhanced description in this chapter
			• Added “ports P6_P10 (excluding P8_5, P9_0 to P9_2, and P9_4 to P9_7)” to lines 2 to 3 of <b>15. Programmable I/O Ports</b>
		187	• Modified expression “bus control pins” in line 7 of <b>15.1</b> and Note 1 of <b>Figure 15.4</b> to “bus control signals”
		—	<b>Chapter 16. Timers</b>
		189	• Modified wording and enhanced description in this chapter
			• Separated signal for overflow or underflow from interrupt signal in <b>Figure 16.2</b>
		193	• Deleted “Counting” from UDF register name and bit names of bits TA4UD to TA0UD in <b>Figure 16.7</b>
		201	• Changed the MR2 bit name “Increment/Decrement Count Switching Source Select Bit” in <b>Figure 16.12</b> to “Increment/Decrement Switching Source Select Bit”; Corrected bit symbols “TAiTGH and TAiTGL” in Note 5 to “TAjTGH and TAjTGL”
		203	• Corrected a typo “TA4NR” in line 3 of <b>16.1.3</b> to “TA4MR”
		204, 206	• Modified function descriptions of MR2 bit in <b>Figures 16.15 and 16.16</b>
		220	• Modified description in line 2 to 4 of <b>16.3.3.2</b>
			• Modified expression “Tbj interrupt handler” in eight bullet point of <b>16.3.3.3</b> to “timer Bj interrupt handler”
		—	<b>Chapter 17. Three-phase Motor Control Timers</b>
			• Modified wording and enhanced description in this chapter

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		222	• Added "P3_2 to P3_7" to <b>Figure 17.1</b> as ports not shown in the diagram
		224	• Changed function descriptions "Timer A reload control signal is 0" and "Timer A reload control signal is 1" for INV13 bit in <b>Figure 17.3</b> to "Timer A1 reload control signal is 0" and "Timer A1 reload control signal is 1", respectively; Modified Note 1
		229	• Changed function descriptions of bits MR3 and MR2 in <b>Figure 17.8</b>
		230	• Changed function description of PWCON bit in <b>Figure 17.9</b>
		231	• Deleted description in lines 8 to 9 of <b>17.3</b>
		238	• Modified "overflow" in <b>17.6.2</b> to "underflow"
		<b>Chapter 18. Serial Interface</b>	
		—	• Modified wording and enhanced description in this chapter
		243	• Deleted "I <sup>2</sup> C mode" from function of bits SMD2 to SMD0 in <b>Figure 18.4</b>
		244, 245	• Modified CRD bit name in <b>Figures 18.5 and 18.6</b> to "CTS Function Disable Bit"; Modified its function descriptions
		246	• Modified bit name of UiLCH bit in <b>Figure 18.7</b> to "Logic Inversion Select Bit"; Added 0 to b7 of the bit box
		248	• Added "0" to b7 in <b>Figure 18.10</b>
		249	• Modified bit name of CSC bit in <b>Figure 18.11</b> to "Clock Synchronization Bit"
		249, 251	• Deleted "of the SCLi" from function descriptions of bits SWC and SWC9 in <b>Figures 18.11 and 18.13</b>
		250	• Corrected description "UiBRG count source" in function description of bits DL2 to DL0 in <b>Figure 18.12</b> to "baud rate generator count source"
		251, 279	• Corrected a typo "STARREQ" in Note 3 of <b>Figure 18.13</b> and line 1 of <b>18.3.2</b> to "STAREQ"
		255	• Deleted Note 1 from <b>Figure 18.19</b>
		256	• Modified i value "(i = 0 to 6)" for "Transmit/receive clock" in <b>Table 18.2</b> to "(i= 0 to 8)"
		259	• Modified "TXEPT flag" in <b>Figure 18.20</b> to "TXEPT bit"; Corrected bit symbol "UiRS" in the fourth dash to "UiIRS"
		261, 270	• Deleted "(i = 0 to 8)" from B of <b>18.1.1 and 18.2.2</b>
		266	• Corrected bit symbol "SUM0" for UiRB register in <b>Table 18.7</b> to "SUM"
		267, 268	• Corrected bit functions of UiIRS bit in the fourth dash in <b>Figures 18.25 and 18.26</b>
		271, 272	• Changed expression "Transmit/receive clock" in <b>Figures 18.29 and 18.30</b> to "CLKi"
		276, 277	• Divided Table 18.11 into <b>Tables 18.11 and 18.12</b>
		290	• Moved description in the fourth dash in <b>18.5.3.1</b> to the second dash
		291	• Added new paragraph for "Reset Procedure on Communication Error" as <b>18.5.5</b>
		<b>Chapter 19. A/D Converter</b>	
		—	• Modified wording and enhanced description in this chapter
			• Changed expression "A/D conversion result" to "A/D converted result"

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		297	<ul style="list-style-type: none"> <li>• Modified expression “sample &amp; hold function” in <b>Figure 19.4</b> to “sample and hold function”</li> </ul>
		298	<ul style="list-style-type: none"> <li>• Changed description of Note 5 in <b>Figure 19.5</b></li> </ul>
		299	<ul style="list-style-type: none"> <li>• Corrected the position of footnote (2) in function column of bits MPS11 and MPS10 in <b>Figure 19.6</b> to their bit name column; Modified “(Note 2)” in function column to “(Note 3)”</li> </ul>
		300-306	<ul style="list-style-type: none"> <li>• Modified Notes 1, 3, and 4 in <b>Figure 19.7</b></li> <li>• Added description “Set the TRG0 bit in the AD0CON2 register to select external trigger request source” to “Start conditions” of <b>Tables 19.2 to 19.8</b>; Described details of the start conditions by TRG0 bit value</li> </ul>
		301, 303, 304	<ul style="list-style-type: none"> <li>• Added the second bullet point to “Specification” of “Reading A/D converted result” in <b>Tables 19.3, 19.5, and 19.6</b></li> </ul>
		304	<ul style="list-style-type: none"> <li>• Modified description of the number of prioritized pins in line 1 of <b>19.1.5</b> and “Function” in <b>Table 19.6</b></li> </ul>
		308	<ul style="list-style-type: none"> <li>• Deleted description “(AN0 to AN7, ANEX0, ANEX1 as analog input port)” from line 10 of <b>19.2.6</b></li> </ul>
		312	<ul style="list-style-type: none"> <li>• Added description to the ninth bullet point of <b>19.3.2</b></li> </ul>
			<b>Chapter 20. D/A Converter</b>
		—	<ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> </ul>
			<b>Chapter 21. CRC Calculator</b>
		—	<ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> </ul>
		315	<ul style="list-style-type: none"> <li>• Corrected a typo “CRC_CCITT” in line 2 of <b>21. CRC Calculator</b> to “CRC-CCITT”</li> <li>• Modified <b>Figure 21.1</b></li> </ul>
			<b>Chapter 22. X-Y Conversion</b>
		—	<ul style="list-style-type: none"> <li>• Made minor text modifications to this chapter</li> </ul>
		318, 319	<ul style="list-style-type: none"> <li>• Changed preposition “to” in between addresses to “-” in <b>Figures 22.2 and 22.3</b>; Changed figure titles “XiR Register” and “YiR Register” to “Registers X0R to X15R” and “Registers Y0R to Y15R”, respectively</li> </ul>
			<b>Chapter 23. Intelligent I/O</b>
		—	<ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> </ul>
		323, 324	<ul style="list-style-type: none"> <li>• Changed “the <math>\overline{\text{INT0}}</math> pin” in <b>Figure 23.1</b> and “the <math>\overline{\text{INT1}}</math> pin” in <b>Figure 23.2</b> to “the <math>\overline{\text{INT0}}</math> pin or the <math>\overline{\text{INT1}}</math> pin”</li> </ul>
		325	<ul style="list-style-type: none"> <li>• Corrected pin names “IE_IN/ISRxD2” and “ISTxD2/IE_OUT” in <b>Figure 23.3</b> to “IEIN/ISRxD2” and “ISTXD2/IEOUT”, respectively</li> </ul>
		328	<ul style="list-style-type: none"> <li>• Changed “<math>\overline{\text{INTi}}</math> pin” in <b>Figure 23.6</b> to “<math>\overline{\text{INT0}}/\overline{\text{INT1}}</math>”; Modified bit name for bits UD1 and UD0 to “Increment/Decrement Control Bit”; Deleted sentence in parenthesis in Note 2; Modified Note 3</li> </ul>
		330	<ul style="list-style-type: none"> <li>• Corrected bit symbol “BT3S” in (2) of Note 1 in <b>Figure 23.8</b> to “BT2S”</li> </ul>
		331	<ul style="list-style-type: none"> <li>• Deleted Note 3 from <b>Figure 23.9</b></li> </ul>
		334	<ul style="list-style-type: none"> <li>• Corrected pin name “ISTxD2” in Note 3 of <b>Figure 23.13</b> to “ISTXD2”</li> </ul>
		338	<ul style="list-style-type: none"> <li>• Modified descriptions of “Reset conditions” in <b>Table 23.2</b>; Changed the timing of the timer counter to start decrementing in the first bullet point</li> </ul>
		339	<ul style="list-style-type: none"> <li>• Changed “<math>\overline{\text{INTi}}</math> pin” in <b>Figure 23.18</b> to “<math>\overline{\text{INT0}}/\overline{\text{INT1}}</math> pin”; Moved “i = 0 to 2” to its figure title</li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		340, 341 346	<ul style="list-style-type: none"> <li>• Moved "(i = 0, 1)" in <b>Figures 23.19 and 23.20</b> to their figure titles</li> <li>• Moved "k = 4, 5" in <b>Figure 23.24</b> to its figure title</li> </ul>
		349, 351, 354	<ul style="list-style-type: none"> <li>• Modified "input to the IIOi_j pin" in <b>Figures 23.25 to 23.27</b> to "IIOi_j pin"</li> </ul>
		352, 353 355	<ul style="list-style-type: none"> <li>• Divided Table 23.9 into <b>Tables 23.9 and 23.10</b></li> <li>• Modified register name "G2PO0" in <b>Table 23.11</b> to "G2POj"; Modified setting value (10 lower bits) of G2POj register to "000h to 3FFh"</li> </ul>
		356	<ul style="list-style-type: none"> <li>• Modified "fBTi" and "G2POCR" in <b>Figure 23.28</b> to "fBT2" and "G2POCRj", respectively</li> </ul>
		356, 358	<ul style="list-style-type: none"> <li>• Corrected "input to the OUTC2_j pin" in <b>Figures 23.28 and 23.30</b> to "OUTC2_j pin"</li> </ul>
		358	<ul style="list-style-type: none"> <li>• Deleted the second dash of Case 1 in <b>Figure 23.30</b></li> </ul>
		361	<ul style="list-style-type: none"> <li>• Deleted Note 1 from <b>23.4</b></li> </ul>
		363	<ul style="list-style-type: none"> <li>• Modified "ISTxD" and "ISRxD" in <b>Figure 23.36</b> to "ISTXD2" and "ISRXD2", respectively</li> </ul>
		367	<ul style="list-style-type: none"> <li>• Corrected pin name "ISRX2" in <b>Table 23.16</b> to "ISRXD2"</li> </ul>
			<b>Chapter 24. I/O Pins</b>
		—	<ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> </ul>
		369	<ul style="list-style-type: none"> <li>• Deleted ASEL from a factor of pull-up resistor being separated from peripheral functions in <b>24. I/O Pins and Figure 24.1</b></li> </ul>
		370	<ul style="list-style-type: none"> <li>• Modified expression "bus control pins" in line 4 of <b>24.1</b> and Note 1 of <b>Figure 24.2</b> to "bus control signals"</li> </ul>
		377	<ul style="list-style-type: none"> <li>• Corrected description "PD3_i register" in lines 4 to 5 below <b>Figure 24.8</b> to "PD3_i bit"</li> </ul>
		380	<ul style="list-style-type: none"> <li>• Corrected pin name "OUTC_1" in <b>Figure 24.11</b> to "OUTC2_1"</li> </ul>
		383	<ul style="list-style-type: none"> <li>• Corrected a typo "PD_9i" in line 8 below <b>Figure 24.14</b> to "PD9_i"</li> </ul>
			<b>Chapter 25. Flash Memory</b>
		—	<ul style="list-style-type: none"> <li>• Modified wording and enhanced description in this chapter</li> <li>• Added question marks to the sentences in diamond-shaped boxes of flows</li> </ul>
		399	<ul style="list-style-type: none"> <li>• Deleted descriptions "erase" and "by using the serial programmer" from "ROM Code Protection" in <b>Table 25.3</b>; Added "erase" to "ID Code Protection"; Corrected a typo "diactivate" to "deactivate"</li> <li>• Deleted description "use the serial programmer to" from line 3 of <b>25.2.2</b></li> </ul>
		400	<ul style="list-style-type: none"> <li>• Corrected address "FFFFFFE8h" in line 9 of <b>25.2.3</b> to "FFFFFFE8h"</li> </ul>
		401	<ul style="list-style-type: none"> <li>• Modified description of last line in <b>25.2.5</b></li> </ul>
		403	<ul style="list-style-type: none"> <li>• Changed the following expressions in <b>Table 25.7</b>: "the program or the block erase command" to "the program command or the block erase command" and "the read status register command" and "the ready status register command" to "the enter read status register mode command"</li> </ul>
		409	<ul style="list-style-type: none"> <li>• Changed minimum value for registers CB01 and CB12 in <b>Table 25.8</b> to "02h" and maximum value for registers CB12 and CB23 to "F8h"; Modified descriptions of setting range</li> </ul>
		410, 412	<ul style="list-style-type: none"> <li>• Changed expressions "<math>\overline{CS0}</math>" and "A23 to A0, <math>\overline{BC3}</math> to <math>\overline{BC0}</math>" in <b>Figures 25.12 and 25.13</b> to "Chip select" and "Address", respectively</li> </ul>

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		415	• Changed expression “program/erase” in <b>25.3.4</b> to “program or erase”
		424	• Corrected power supply “VCC2” for ports P6_0 to P6_7 in <b>Table 25.21</b> to “VCC1”
		431	• Modified description of the first bullet point in <b>25.6.8</b>
		—	<b>Chapter 26. Electrical Characteristics</b>
			• Modified wording and enhanced description in this chapter
		433	• Changed expression “clock period” to “clock cycle time”
		438	• Changed description format for ports P0 and P1 in <b>Table 26.2</b>
			• Changed expression “Programming and erasure endurance” in <b>Table 26.8</b> to “Program and erase cycles”; Changed its unit “times” in the table and Note 1 to “Cycles”
		441	• Changed expressions “CS0” and “A23 to A0, BC0 to BC3” in <b>Figure 26.5</b> to “Chip select” and “Address”, respectively
		450, 462	• Corrected “INTi” in title of <b>Tables 26.32 and 26.55</b> to “INTi”
		453, 465	• Added measurement condition to <b>Tables 26.37 and 26.60</b>
		—	<b>Chapter 27. Usage Notes</b>
			• Modified wording and enhanced description in this chapter
		470	• Corrected a typo “trances” in the first bullet point of <b>27.1.1</b> to “traces”
		471, 472	• Changed order of <b>Tables 27.1 and 27.2</b>
		472	• Deleted expression “counting” from UDF register name in <b>Table 27.2</b>
		479	• Modified description in line 2 to 4 of <b>27.7.3.2</b>
			• Modified expression “Tbj interrupt handler” in eight bullet point of <b>27.7.3.3</b> to “timer Bj interrupt handler”
		480	• Modified “overflow” in <b>27.8.2</b> to “underflow”
		482	• Moved description in the fourth dash in <b>27.9.3.1</b> to the second dash
		483	• Added new paragraph for “Reset Procedure on Communication Error” as <b>27.9.5</b>
		485	• Added description to the ninth bullet point of <b>27.10.2</b>
		487	• Modified description of the first bullet point in <b>27.11.8</b>
		—	<b>Appendix 1. Package Dimensions</b>
		488-489	• Added a seating plane to the drawing of package dimension
1.30	Mar 13, 2014	—	10th edition released
		—	• Deleted description for the 80-pin package
		—	<b>Chapter 1. Overview</b>
			• Modified wording and enhanced description in this chapter
		1	• Modified expression “I <sup>2</sup> C” in line 9 of <b>1.1</b> to “I <sup>2</sup> C-bus interface”
		2, 4	• Modified expressions “calculation transfer” and “chained transfer” in <b>Tables 1.1 and 1.3</b> to “calculation result transfer” and “chain transfer”, respectively
		5	• Deleted N version from the Operating Temperature row in <b>Table 1.4</b>
		6	• Deleted products on planning phase from <b>Table 1.5</b>
		—	<b>Chapter 2. CPU</b>
			• Modified wording and enhanced description in this chapter
		—	<b>Chapter 5. Resets</b>
			• Modified wording and enhanced description in this chapter



Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
			<b>Chapter 6. Power Management</b>
		—	• Modified wording and enhanced description in this chapter
		62	• Modified description in Note 4 of <b>Figure 6.4</b>
		64	• Modified description “the voltage rises above Vdet(R)” in line 7 of <b>6.2.1</b> to “the voltage rises to or above Vdet(R)”
			<b>Chapter 7. Processor Mode</b>
		67	• Modified wording and enhanced description in Notes 1, 4, and 5 of <b>Figure 7.1</b>
			<b>Chapter 8. Clock Generator</b>
		—	• Modified wording and enhanced description in this chapter
		72	• Modified CM03 bit name “XCIN-XCOUT Drive Power Select Bit” in <b>Figure 8.3</b> to “XCIN-XCOUT Drive Strength Select Bit”
		73	• Modified bit name of the CM15 bit “XIN-XOUT Drive Power Select Bit” in <b>Figure 8.4</b> to “XIN-XOUT Drive Strength Select Bit”
		99	• Modified expression “drive power” in line 2 of <b>8.9.1.1</b> to “drive strength”
			<b>Chapter 9. Bus</b>
		—	• Modified wording and enhanced description in this chapter
		113	• Typo “ $\overline{WR0}$ ” in <b>Figure 9.5</b> is corrected to “ $\overline{WR}$ ”
			<b>Chapter 10. Protection</b>
		—	• Modified wording and enhanced description in this chapter
		124	• Deleted description of bits PRC0 and PRC1 from Note 1 in <b>Figure 10.1</b>
			<b>Chapter 11. Interrupts</b>
		—	• Modified wording and enhanced description in this chapter
		129	• Added description “Peripheral interrupts are maskable.” to <b>11.3.2</b>
		144	• Modified descriptions in lines 7 to 8 of <b>11.8</b>
			<b>Chapter 12. Watchdog Timer</b>
		—	• Modified wording and enhanced description in this chapter
		153	• Modified description in line 6 of <b>12</b>
			• Added description “Depending on the timing of when a value is written to the WDTS register” to line 13 of <b>12</b>
			<b>Chapter 13. DMAC</b>
		—	• Modified wording and enhanced description in this chapter
		156	• Moved “(i = 0 to 3)” in <b>Table 13.1</b> to the title
		165	• Corrected address “00060000h” for the external bus in <b>Table 13.5</b> to “0080000h”
			<b>Chapter 14. DMAC II</b>
		—	• Modified wording and enhanced description in this chapter
		170	• Modified expression “chained transfer” to “chain transfer”
		172, 173	• Modified description of chain transfer in <b>Table 14.1</b>
			• Changed expression “DMA II transfer complete interrupt vector address” in lines 3 to 4 and the seventh bullet point of <b>14.1.2</b> and <b>Figure 14.2</b> to “jump address for the DMA II transfer complete interrupt handler”
		172, 175	• Modified expression “interrupt vector” in <b>Figure 14.2</b> and line 1 of <b>14.1.4</b> to “interrupt vector space”

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		173	• Changed expression “jump address” in the seventh bullet point of <b>14.1.2</b> to “start address”
		178	• Modified descriptions in <b>Figure 14.5</b>
		<b>Chapter 16. Timers</b>	
		—	• Modified wording and enhanced description in this chapter
		184	• Modified description for one-shot timer mode in <b>16.1</b>
		189	• Corrected typos “b2 b3”, “b4 b5”, and “b6 b7” in <b>Figure 16.9</b> to “b3 b2”, “b5 b4”, and “b7 b6”, respectively
		196	• Corrected pin name “INT2” in <b>Figures 16.13 and 16.14</b> to “ $\overline{\text{INT2}}$ ”
		200	• Corrected bit symbol “TAiS” in the Function column for the MR2 bit in <b>Figure 16.16</b> to “TAiOS”
		<b>Chapter 17. Three-phase Motor Control Timers</b>	
		—	• Modified wording and enhanced description in this chapter
		230	• Corrected register symbol “INV1” in Note 2 of <b>Figure 17.17</b> to “INVC1”
		<b>Chapter 18. Serial Interface</b>	
		—	• Modified wording and enhanced description in this chapter
		—	• Modified “SS function” to “slave select function”
		—	• Modified “transmit register” and “receive register” to “transmit shift register” and “receive shift register”, respectively
		—	• Modified “restart condition” to “repeated START condition”
		234	• Modified the position of “010” for bits SMD2 to SMD0 in <b>Figure 18.1</b>
		240	• Changed “UiTB register” in Function of the UiIRS bit in <b>Figure 18.7</b> to “Transmit buffer”
		244	• Modified “the baud rate generator count source” in <b>Figure 18.12</b> to “count source for the UiBRG register”; Modified description in note 2
		251, 252, 259, 260, 269, 279	• Modified description in Function of the UiBRG register in <b>Tables 18.3, 18.4, 18.6, 18.7, 18.10, and 18.15</b>
		252	• Added description for (b7 to b4) to registers UiC1 and U78CON in <b>Table 18.4</b>
		253	• Modified waveform of the IR bit in <b>Figure 18.20</b>
		255, 264	• Modified descriptions in <b>18.1.1 and 18.2.2</b>
		257	• Added description “after the last bit is transmitted” to line 3 of <b>18.1.6</b>
		260	• Added description for (b7) to the UiMR register in <b>Table 18.7</b> ; Added description for (b7 to b4) to registers UiC1 and U78CON
		261, 262	• Corrected “Internal transmit/receive clock” in <b>Figures 18.25 and 18.26</b> to “Internal transmit clock”
		263	• Corrected “Transmit/receive clock” in <b>Figure 18.27</b> to “Internal receive clock”; Modified waveform of the RI bit
		—	• Modified “BRG” in <b>Table 18.8</b> to “UiBRG”
		267	• Modified the Interrupt request generating timing row in <b>Table 18.9</b>
		269	• Added description for (b6 to b4) to the UiMR register in <b>Table 18.10</b>
		284	• Added description for suspending and resuming communication to <b>18.5.5</b>
		<b>Chapter 19. A/D Converter</b>	
		—	• Modified wording and enhanced description in this chapter

Revision History	R32C/111 Group User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
		—	<b>Chapter 20. D/A Converter</b> • Modified wording and enhanced description in this chapter
		—	<b>Chapter 22. X-Y Conversion</b> • Modified wording and enhanced description in this chapter
		—	<b>Chapter 23. Intelligent I/O</b> • Modified wording and enhanced description in this chapter
		—	<b>Chapter 24. I/O Pins</b> • Modified wording and enhanced description in this chapter
		362	• Modified descriptions for pull-up registers in lines 5 and 6 in <b>24</b>
		386	• Modified descriptions in lines 10 to 12 below <b>Figure 24.1</b>
		386	• Modified “addresses: 03CAh, 03CBh” in Note 5 of <b>Table 24.2</b> to “registers PD4 and PD5”
		386, 387	• Modified descriptions in Note 2 of <b>Tables 24.2 and 24.3</b>
		—	<b>Chapter 25. Flash Memory</b> • Modified wording and enhanced description in this chapter
		400	• Deleted description “When the ID code protection is activated,” from <b>25.2.3</b>
		—	<b>Chapter 26. Electrical Characteristics</b> • Modified wording and enhanced description in this chapter
		435, 447	• Deleted TXD4, STXD4, and $\overline{\text{RTS4}}$ from Note 1 of <b>Tables 26.16 and 26.39</b>
		436, 448	• Modified description “Drive power” in <b>Tables 26.17 and 26.40</b> to “Drive strength”
		—	<b>Chapter 27. Usage Notes</b> • Modified wording and enhanced description in this chapter
		465	• Modified expression “drive power” in line 2 of <b>27.3.1.1</b> to “drive strength”
		473	• Modified expression “restart condition” in line 1 of <b>27.9.4</b> to “repeated START condition”
		474	• Added description for suspending and resuming communication to <b>27.9.5</b>

---

R32C/111 Group User's Manual: Hardware

Publication Date: Rev.1.00 Dec 12, 2008  
Rev.1.30 Mar 13, 2014

Published by: Renesas Electronics Corporation

---

**SALES OFFICES****Renesas Electronics Corporation**<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.**Renesas Electronics America Inc.**2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130**Renesas Electronics Canada Limited**1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada  
Tel: +1-905-898-5441, Fax: +1-905-898-3220**Renesas Electronics Europe Limited**Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K  
Tel: +44-1628-585-100, Fax: +44-1628-585-900**Renesas Electronics Europe GmbH**Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-6503-0, Fax: +49-211-6503-1327**Renesas Electronics (China) Co., Ltd.**Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679**Renesas Electronics (Shanghai) Co., Ltd.**Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333  
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999**Renesas Electronics Hong Kong Limited**Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2265-6688, Fax: +852 2686-9022/9044**Renesas Electronics Taiwan Co., Ltd.**13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670**Renesas Electronics Singapore Pte. Ltd.**80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949  
Tel: +65-6213-0200, Fax: +65-6213-0300**Renesas Electronics Malaysia Sdn.Bhd.**Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510**Renesas Electronics Korea Co., Ltd.**12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

R32C/111 Group



Renesas Electronics Corporation

R01UH0209EJ0130