

**16**

# R8C/2G Group

## Hardware Manual

RENESAS MCU  
R8C FAMILY / R8C/2x 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 Technology Corp. without notice. Please review the latest information published by Renesas Technology Corp. through various means, including the Renesas Technology Corp. website (<http://www.renesas.com>).

Rev.1.00  
Revision Date: Apr 04, 2008

Renesas Technology  
[www.renesas.com](http://www.renesas.com)

Hardware Manual

## Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com> )
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
  - (1) artificial life support devices or systems
  - (2) surgical implantations
  - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
  - (4) any other purposes that pose a direct threat to human lifeRenesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

## General Precautions in the Handling of MPU/MCU Products

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

### 1. Handling of Unused Pins

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

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

### 2. Processing at Power-on

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

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

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

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

### 3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

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

### 4. Clock Signals

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

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

### 5. Differences between Products

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

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

# How to Use This Manual

## 1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual.

The manual comprises an overview of the product; descriptions of the CPU, system control functions, peripheral functions, and electrical characteristics; and usage notes.

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

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

The following documents apply to the R8C/2G Group. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Technology Web site.

Document Type	Description	Document Title	Document No.
Datasheet	Hardware overview and electrical characteristics	R8C/2G Group Datasheet	REJ03B0223
Hardware manual	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	R8C/2G Group Hardware Manual	This hardware manual
Software manual	Description of CPU instruction set	R8C/Tiny Series Software Manual	REJ09B0001
Application note	Information on using peripheral functions and application examples Sample programs Information on writing programs in assembly language and C	Available from Renesas Technology Web site.	
Renesas technical update	Product specifications, updates on documents, etc.		

## 2. Notation of Numbers and Symbols

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

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

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

Examples the PM03 bit in the PM0 register  
P3\_5 pin, VCC pin

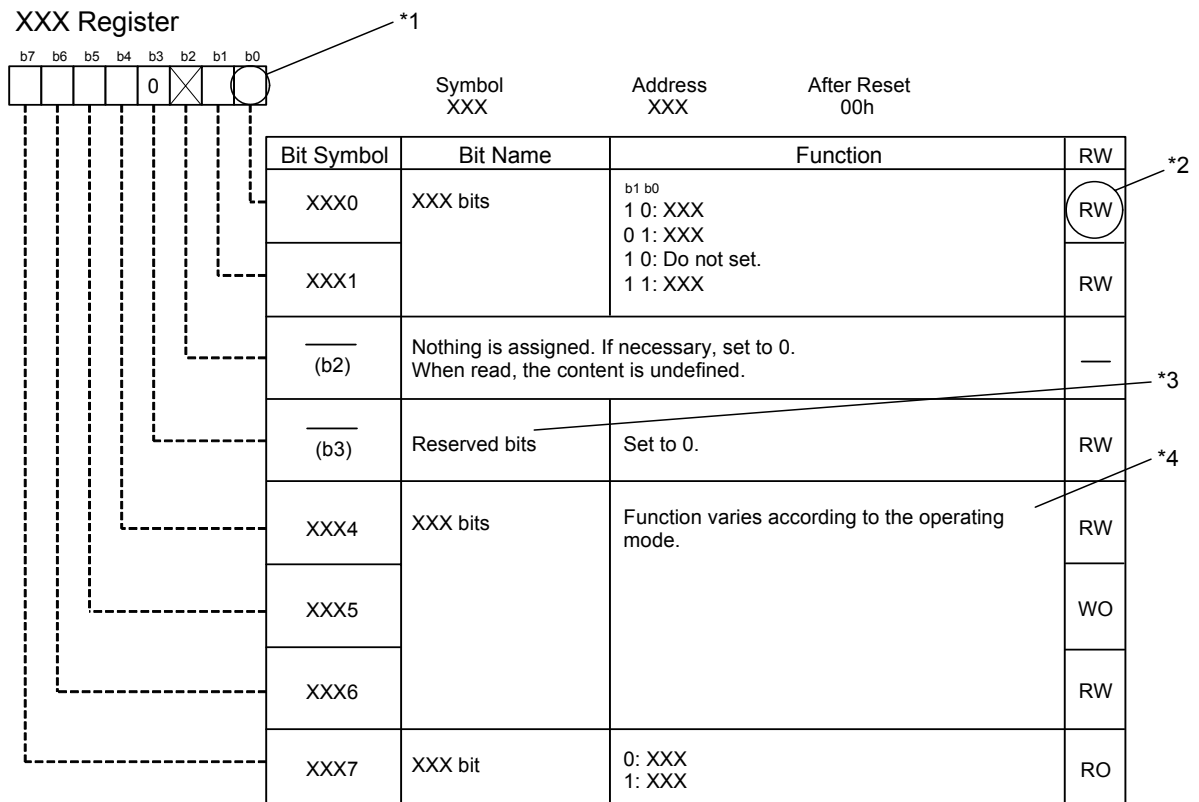
(2) Notation of Numbers

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

Examples Binary: 11b  
Hexadecimal: EFA0h  
Decimal: 1234

### 3. Register Notation

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



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

\*2  
RW: Read and write.  
RO: Read only.  
WO: Write only.  
—: Nothing is assigned.

\*3  
• Reserved bit  
Reserved bit. Set to specified value.

\*4  
• Nothing is assigned  
Nothing is assigned to the bit. As the bit may be used for future functions, if necessary, set to 0.  
• Do not set to a value  
Operation is not guaranteed when a value is set.  
• Function varies according to the operating mode.  
The function of the bit varies with the peripheral function mode. Refer to the register diagram for information on the individual modes.

#### 4. List of Abbreviations and Acronyms

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

All trademarks and registered trademarks are the property of their respective owners.

# Table of Contents

SFR Page Reference .....	B - 1
1. Overview .....	1
1.1 Features .....	1
1.1.1 Applications .....	1
1.1.2 Specifications .....	1
1.2 Product List .....	3
1.3 Block Diagram .....	4
1.4 Pin Assignment .....	5
1.5 Pin Functions .....	7
2. Central Processing Unit (CPU) .....	8
2.1 Data Registers (R0, R1, R2, and R3) .....	9
2.2 Address Registers (A0 and A1) .....	9
2.3 Frame Base Register (FB) .....	9
2.4 Interrupt Table Register (INTB) .....	9
2.5 Program Counter (PC) .....	9
2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP) .....	9
2.7 Static Base Register (SB) .....	9
2.8 Flag Register (FLG) .....	9
2.8.1 Carry Flag (C) .....	9
2.8.2 Debug Flag (D) .....	9
2.8.3 Zero Flag (Z) .....	9
2.8.4 Sign Flag (S) .....	9
2.8.5 Register Bank Select Flag (B) .....	9
2.8.6 Overflow Flag (O) .....	9
2.8.7 Interrupt Enable Flag (I) .....	10
2.8.8 Stack Pointer Select Flag (U) .....	10
2.8.9 Processor Interrupt Priority Level (IPL) .....	10
2.8.10 Reserved Bit .....	10
3. Memory .....	11
4. Special Function Registers (SFRs) .....	12
5. Resets .....	24
5.1 Hardware Reset .....	27
5.1.1 When Power Supply is Stable .....	27
5.1.2 Power On .....	27
5.2 Power-On Reset Function .....	29
5.3 Voltage Monitor 0 Reset .....	30
5.4 Voltage Monitor 1 Reset .....	30
5.5 Voltage Monitor 2 Reset .....	30
5.6 Watchdog Timer Reset .....	31
5.7 Software Reset .....	31
6. Voltage Detection Circuit .....	32
6.1 VCC Input Voltage .....	40
6.1.1 Monitoring Vdet0 .....	40



6.1.2	Monitoring Vdet1 .....	40
6.1.3	Monitoring Vdet2 .....	40
6.2	Voltage Monitor 0 Reset .....	41
6.3	Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset .....	42
6.4	Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset .....	44
<b>7.</b>	<b>Comparator .....</b>	<b>46</b>
7.1	Overview .....	46
7.2	Register Description .....	48
7.3	Monitoring Comparison Results .....	55
7.3.1	Monitoring Comparator 1 .....	55
7.3.2	Monitoring Comparator 2 .....	55
7.4	Functional Description .....	56
7.4.1	Comparator 1 .....	56
7.4.2	Comparator 2 .....	59
7.5	Comparator 1 and Comparator 2 Interrupts .....	62
7.5.1	Non-Maskable Interrupts .....	62
7.5.2	Maskable Interrupts .....	62
7.6	Adjusting Internal Reference Voltage (Vref) .....	63
<b>8.</b>	<b>I/O Ports .....</b>	<b>65</b>
8.1	Functions of I/O Ports .....	65
8.2	Effect on Peripheral Functions .....	66
8.3	Pins Other than Programmable I/O Ports .....	66
8.4	Port Setting .....	75
8.5	Unassigned Pin Handling .....	83
8.6	Notes on I/O Ports .....	84
8.6.1	Port P4_3, P4_4 .....	84
<b>9.</b>	<b>Processor Mode .....</b>	<b>85</b>
9.1	Processor Modes .....	85
<b>10.</b>	<b>Bus .....</b>	<b>86</b>
<b>11.</b>	<b>Clock Generation Circuit .....</b>	<b>87</b>
11.1	On-Chip Oscillator Clocks .....	96
11.1.1	Low-Speed On-Chip Oscillator Clock .....	96
11.1.2	High-Speed On-Chip Oscillator Clock .....	96
11.2	XCIN Clock .....	97
11.3	CPU Clock and Peripheral Function Clock .....	98
11.3.1	System Clock .....	98
11.3.2	CPU Clock .....	98
11.3.3	Peripheral Function Clock (f1, f2, f4, f8, and f32) .....	98
11.3.4	fOCO .....	98
11.3.5	fOCO-F .....	98
11.3.6	fOCO-S .....	98
11.3.7	fC4 and fC32 .....	98
11.4	Power Control .....	99
11.4.1	Standard Operating Mode .....	99

11.4.2	Wait Mode .....	101
11.4.3	Stop Mode .....	103
11.5	Notes on Clock Generation Circuit .....	106
11.5.1	Stop Mode .....	106
11.5.2	Wait Mode .....	106
11.5.3	Oscillation Circuit Constants .....	106
12.	Protection .....	107
13.	Interrupts .....	108
13.1	Interrupt Overview .....	108
13.1.1	Types of Interrupts .....	108
13.1.2	Software Interrupts .....	109
13.1.3	Special Interrupts .....	110
13.1.4	Peripheral Function Interrupt .....	110
13.1.5	Interrupts and Interrupt Vectors .....	111
13.1.6	Interrupt Control .....	113
13.2	$\overline{\text{INT}}$ Interrupt .....	121
13.2.1	$\overline{\text{INT}}_i$ Interrupt (i = 0, 1, 2, 4) .....	121
13.2.2	$\overline{\text{INT}}_i$ Input Filter (i = 0, 1, 2, 4) .....	124
13.3	Key Input Interrupt .....	125
13.4	Address Match Interrupt .....	127
13.5	Notes on Interrupts .....	129
13.5.1	Reading Address 00000h .....	129
13.5.2	SP Setting .....	129
13.5.3	External Interrupt and Key Input Interrupt .....	129
13.5.4	Changing Interrupt Sources .....	130
13.5.5	Changing Interrupt Control Register Contents .....	131
14.	ID Code Areas .....	132
14.1	Overview .....	132
14.2	Functions .....	132
14.3	Notes on ID Code Areas .....	133
14.3.1	Setting Example of ID Code Areas .....	133
15.	Option Function Select Area .....	134
15.1	Overview .....	134
15.2	OFS Register .....	135
15.3	Notes on Option Function Select Area .....	136
15.3.1	Setting Example of Option Function Select Area .....	136
16.	Watchdog Timer .....	137
16.1	Count Source Protection Mode Disabled .....	141
16.2	Count Source Protection Mode Enabled .....	142
17.	Timers .....	143
17.1	Timer RA .....	145
17.1.1	Timer Mode .....	148
17.1.2	Pulse Output Mode .....	150

17.1.3	Event Counter Mode .....	152
17.1.4	Pulse Width Measurement Mode .....	154
17.1.5	Pulse Period Measurement Mode .....	157
17.1.6	Notes on Timer RA .....	160
17.2	Timer RB .....	161
17.2.1	Timer Mode .....	165
17.2.2	Programmable Waveform Generation Mode .....	168
17.2.3	Programmable One-shot Generation Mode .....	171
17.2.4	Programmable Wait One-Shot Generation Mode .....	175
17.2.5	Notes on Timer RB .....	178
17.3	Timer RE .....	182
17.3.1	Real-Time Clock Mode .....	183
17.3.2	Output Compare Mode .....	191
17.3.3	Notes on Timer RE .....	197
17.4	Timer RF .....	200
17.4.1	Input Capture Mode .....	205
17.4.2	Output Compare Mode .....	208
17.4.3	Notes on Timer RF .....	212
<b>18.</b>	<b>Serial Interface .....</b>	<b>213</b>
18.1	Clock Synchronous Serial I/O Mode .....	218
18.1.1	Polarity Select Function .....	221
18.1.2	LSB First/MSB First Select Function .....	221
18.1.3	Continuous Receive Mode .....	222
18.2	Clock Asynchronous Serial I/O (UART) Mode .....	223
18.2.1	Bit Rate .....	227
18.3	Notes on Serial Interface .....	228
<b>19.</b>	<b>Hardware LIN .....</b>	<b>229</b>
19.1	Features .....	229
19.2	Input/Output Pins .....	230
19.3	Register Configuration .....	231
19.4	Functional Description .....	233
19.4.1	Master Mode .....	233
19.4.2	Slave Mode .....	236
19.4.3	Bus Collision Detection Function .....	240
19.4.4	Hardware LIN End Processing .....	241
19.5	Interrupt Requests .....	242
19.6	Notes on Hardware LIN .....	243
<b>20.</b>	<b>Flash Memory .....</b>	<b>244</b>
20.1	Overview .....	244
20.2	Memory Map .....	245
20.3	Functions to Prevent Rewriting of Flash Memory .....	246
20.3.1	ID Code Check Function .....	246
20.3.2	ROM Code Protect Function .....	247
20.4	CPU Rewrite Mode .....	248
20.4.1	Register Description .....	249
20.4.2	Status Check Procedure .....	254

20.4.3	EW0 Mode .....	255
20.4.4	EW1 Mode .....	261
20.5	Standard Serial I/O Mode .....	266
20.5.1	ID Code Check Function .....	266
20.6	Parallel I/O Mode .....	268
20.6.1	ROM Code Protect Function .....	268
20.7	Notes on Flash Memory .....	269
20.7.1	CPU Rewrite Mode .....	269
<b>21.</b>	<b>Reducing Power Consumption .....</b>	<b>271</b>
21.1	Overview .....	271
21.2	Key Points and Processing Methods for Reducing Power Consumption .....	271
21.2.1	Voltage Detection Circuit .....	271
21.2.2	Ports .....	271
21.2.3	Clocks .....	271
21.2.4	Selecting Oscillation Drive Capacity .....	271
21.2.5	Wait Mode, Stop Mode .....	271
21.2.6	Stopping Peripheral Function Clocks .....	271
21.2.7	Timers .....	271
21.2.8	Reducing Internal Power Consumption .....	272
21.2.9	Stopping Flash Memory .....	273
21.2.10	Low-Current-Consumption Read Mode .....	274
<b>22.</b>	<b>Electrical Characteristics .....</b>	<b>275</b>
<b>23.</b>	<b>Usage Notes .....</b>	<b>292</b>
23.1	Notes on I/O Ports .....	292
23.1.1	Port P4_3, P4_4 .....	292
23.2	Notes on Clock Generation Circuit .....	293
23.2.1	Stop Mode .....	293
23.2.2	Wait Mode .....	293
23.2.3	Oscillation Circuit Constants .....	293
23.3	Notes on Interrupts .....	294
23.3.1	Reading Address 00000h .....	294
23.3.2	SP Setting .....	294
23.3.3	External Interrupt and Key Input Interrupt .....	294
23.3.4	Changing Interrupt Sources .....	295
23.3.5	Changing Interrupt Control Register Contents .....	296
23.4	Notes on ID Code Areas .....	297
23.4.1	Setting Example of ID Code Areas .....	297
23.5	Notes on Option Function Select Area .....	298
23.5.1	Setting Example of Option Function Select Area .....	298
23.6	Notes on Timers .....	299
23.6.1	Notes on Timer RA .....	299
23.6.2	Notes on Timer RB .....	300
23.6.3	Notes on Timer RE .....	304
23.6.4	Notes on Timer RF .....	307
23.7	Notes on Serial Interface .....	308
23.8	Notes on Hardware LIN .....	309

23.9	Notes on Flash Memory .....	310
23.9.1	CPU Rewrite Mode .....	310
23.10	Notes on Noise .....	312
23.10.1	Inserting a Bypass Capacitor between VCC and VSS Pins as a Countermeasure against Noise and Latch-up .....	312
23.10.2	Countermeasures against Noise Error of Port Control Registers .....	312
24.	Notes for On-Chip Debugger .....	313
	Appendix 1. Package Dimensions .....	314
	Appendix 2. Connection Examples with On-Chip Debugging Emulator .....	315
	Appendix 3. Example of Oscillation Evaluation Circuit .....	316
	Index .....	317

# SFR Page Reference

Address	Register	Symbol	Page
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	85
0005h	Processor Mode Register 1	PM1	85
0006h	System Clock Control Register 0	CM0	89
0007h	System Clock Control Register 1	CM1	90
0008h			
0009h			
000Ah	Protect Register	PRCR	107
000Bh			
000Ch	System Clock Select Register	OCD	91
000Dh	Watchdog Timer Reset Register	WDTR	139
000Eh	Watchdog Timer Start Register	WDTS	139
000Fh	Watchdog Timer Control Register	WDC	139
0010h	Address Match Interrupt Register 0	RMAD0	128
0011h			
0012h			
0013h	Address Match Interrupt Enable Register	AIER	128
0014h	Address Match Interrupt Register 1	RMAD1	128
0015h			
0016h			
0017h			
0018h			
0019h			
001Ah			
001Bh			
001Ch	Count Source Protection Mode Register	CSPR	140
001Dh			
001Eh			
001Fh			
0020h	High-Speed On-Chip Oscillator Control Register 0	HRA0	92
0021h	High-Speed On-Chip Oscillator Control Register 1	HRA1	92
0022h	High-Speed On-Chip Oscillator Control Register 2	HRA2	92
0023h			
0024h			
0025h			
0026h			
0027h			
0028h	Clock Prescaler Reset Flag	CPSRF	93
0029h	High-Speed On-Chip Oscillator Control Register 4	FRA4	93
002Ah			
002Bh	High-Speed On-Chip Oscillator Control Register 6	FRA6	93
002Ch			
002Dh			
002Eh	BGR Trimming Auxiliary Register A	BGRTRMA	48
002Fh	BGR Trimming Auxiliary Register B	BGRTRMB	48
0030h			
0031h	Voltage Detection Register 1	VCA1	35, 49
0032h	Voltage Detection Register 2	VCA2	35, 49, 94
0033h			
0034h			
0035h			
0036h	Voltage Monitor 1 Circuit Control Register	VW1C	37, 50
0037h	Voltage Monitor 2 Circuit Control Register	VW2C	38, 51
0038h	Voltage Monitor 0 Circuit Control Register	VW0C	36
0039h			
003Ah			
003Bh	Voltage Detection Circuit External Input Control Register	VCAB	52
003Ch	Comparator Mode Register	ALCMR	52
003Dh	Voltage Monitor Circuit Edge Select Register	VCAC	39, 53
003Eh	BGR Control Register	BGRCR	53
003Fh	BGR Trimming Register	BGRTRM	54

Address	Register	Symbol	Page
0040h			
0041h	Comparator 1 Interrupt Control Register	VCMP1IC	113
0042h	Comparator 2 Interrupt Control Register	VCMP2IC	113
0043h			
0044h			
0045h			
0046h			
0047h			
0048h			
0049h			
004Ah	Timer RE Interrupt Control Register	TREIC	113
004Bh	UART2 Transmit Interrupt Control Register	S2TIC	113
004Ch	UART2 Receive Interrupt Control Register	S2RIC	113
004Dh	Key Input Interrupt Control Register	KUPIC	113
004Eh			
004Fh			
0050h	Compare 1 Interrupt Control Register	CMP1IC	113
0051h	UART0 Transmit Interrupt Control Register	S0TIC	113
0052h	UART0 Receive Interrupt Control Register	S0RIC	113
0053h			
0054h			
0055h	INT2 Interrupt Control Register	INT2IC	114
0056h	Timer RA Interrupt Control Register	TRAIC	113
0057h			
0058h	Timer RB Interrupt Control Register	TRBIC	113
0059h	INT1 Interrupt Control Register	INT1IC	114
005Ah			
005Bh	Timer RF Interrupt Control Register	TRFIC	113
005Ch	Compare 0 Interrupt Control Register	CMP0IC	113
005Dh	INT0 Interrupt Control Register	INT0IC	114
005Eh	INT4 Interrupt Control Register	INT4IC	114
005Fh	Capture Interrupt Control Register	CAPIC	113
0060h			
0061h			
0062h			
0063h			
0064h			
0065h			
0066h			
0067h			
0068h			
0069h			
006Ah			
006Bh			
006Ch			
006Dh			
006Eh			
006Fh			
0070h			
0071h			
0072h			
0073h			
0074h			
0075h			
0076h			
0077h			
0078h			
0079h			
007Ah			
007Bh			
007Ch			
007Dh			
007Eh			
007Fh			

**NOTE:**

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

Address	Register	Symbol	Page
0080h			
0081h			
0082h			
0083h			
0084h			
0085h			
0086h			
0087h			
0088h			
0089h			
008Ah			
008Bh			
008Ch			
008Dh			
008Eh			
008Fh			
0090h			
0091h			
0092h			
0093h			
0094h			
0095h			
0096h			
0097h			
0098h			
0099h			
009Ah			
009Bh			
009Ch			
009Dh			
009Eh			
009Fh			
00A0h	UART0 Transmit/Receive Mode Register	U0MR	215
00A1h	UART0 Bit Rate Register	U0BRG	215
00A2h	UART0 Transmit Buffer Register	U0TB	216
00A3h			
00A4h	UART0 Transmit/Receive Control Register 0	U0C0	216
00A5h	UART0 Transmit/Receive Control Register 1	U0C1	217
00A6h	UART0 Receive Buffer Register	U0RB	217
00A7h			
00A8h			
00A9h			
00AAh			
00ABh			
00ACh			
00ADh			
00AEh			
00AFh			
00B0h			
00B1h			
00B2h			
00B3h			
00B4h			
00B5h			
00B6h			
00B7h			
00B8h			
00B9h			
00BAh			
00BBh			
00BCh			
00BDh			
00BEh			
00BFh			

Address	Register	Symbol	Page
00C0h			
00C1h			
00C2h			
00C3h			
00C4h			
00C5h			
00C6h			
00C7h			
00C8h			
00C9h			
00CAh			
00CBh			
00CCh			
00CDh			
00CEh			
00CFh			
00D0h			
00D1h			
00D2h			
00D3h			
00D4h			
00D5h			
00D6h			
00D7h			
00D8h			
00D9h			
00DAh			
00DBh			
00DCh			
00DDh			
00DEh			
00DFh			
00E0h	Port P0 Register	P0	72
00E1h	Port P1 Register	P1	72
00E2h	Port P0 Direction Register	PD0	71
00E3h	Port P1 Direction Register	PD1	71
00E4h			
00E5h	Port P3 Register	P3	72
00E6h			
00E7h	Port P3 Direction Register	PD3	71
00E8h	Port P4 Register	P4	72
00E9h			
00EAh	Port P4 Direction Register	PD4	71
00EBh			
00ECh	Port P6 Register	P6	72
00EDh			
00EEh	Port P6 Direction Register	PD6	71
00EFh			
00F0h			
00F1h			
00F2h			
00F3h			
00F4h			
00F5h			
00F6h	Pin Select Register 2	PINSR2	73
00F7h	Pin Select Register 3	PINSR3	73
00F8h	Port Mode Register	PMR	74
00F9h	External Input Enable Register	INTEN	121
00FAh	INT Input Filter Select Register	INTF	122
00FBh	Key Input Enable Register	KIEN	126
00FCh	Pull-Up Control Register 0	PUR0	74
00FDh	Pull-Up Control Register 1	PUR1	74
00FEh			
00FFh			

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

Address	Register	Symbol	Page
0100h	Timer RA Control Register	TRACR	146
0101h	Timer RA I/O Control Register	TRAIOC	146, 148, 151, 153, 155, 158
0102h	Timer RA Mode Register	TRAMR	147
0103h	Timer RA Prescaler Register	TRAPRE	147
0104h	Timer RA Register	TRA	147
0105h			
0106h	LIN Control Register	LINCR	231
0107h	LIN Status Register	LINST	232
0108h	Timer RB Control Register	TRBCR	162
0109h	Timer RB One-Shot Control Register	TRBOCR	162
010Ah	Timer RB I/O Control Register	TRBIOC	163, 165, 169, 172, 176
010Bh	Timer RB Mode Register	TRBMR	163
010Ch	Timer RB Prescaler Register	TRBPRES	164
010Dh	Timer RB Secondary Register	TRBSC	164
010Eh	Timer RB Primary Register	TRBPR	164
010Fh			
0110h			
0111h			
0112h			
0113h			
0114h			
0115h			
0116h			
0117h			
0118h	Timer RE Second Data Register / Counter Data Register	TRESEC	185, 193
0119h	Timer RE Minute Data Register / Compare Data Register	TREMIN	185, 193
011Ah	Timer RE Hour Data Register	TREHR	186
011Bh	Timer RE Day of Week Data Register	TREWK	186
011Ch	Timer RE Control Register 1	TRECR1	187, 194
011Dh	Timer RE Control Register 2	TRECR2	188, 194
011Eh	Timer RE Count Source Select Register	TRECSR	189, 195
011Fh	Timer RE Real-Time Clock Precision Adjust Register	TREOPR	189
0120h			
0121h			
0122h			
0123h			
0124h			
0125h			
0126h			
0127h			
0128h			
0129h			
012Ah			
012Bh			
012Ch			
012Dh			
012Eh			
012Fh			

Address	Register	Symbol	Page
0130h			
0131h			
0132h			
0133h			
0134h			
0135h			
0136h			
0137h			
0138h			
0139h			
013Ah			
013Bh			
013Ch			
013Dh			
013Eh			
013Fh			
0140h			
0141h			
0142h			
0143h			
0144h			
0145h			
0146h			
0147h			
0148h			
0149h			
014Ah			
014Bh			
014Ch			
014Dh			
014Eh			
014Fh			
0150h			
0151h			
0152h			
0153h			
0154h			
0155h			
0156h			
0157h			
0158h			
0159h			
015Ah			
015Bh			
015Ch			
015Dh			
015Eh			
015Fh			

NOTE:

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



Address	Register	Symbol	Page
0160h	UART2 Transmit/Receive Mode Register	U2MR	215
0161h	UART2 Bit Rate Register	U2BRG	215
0162h	UART2 Transmit Buffer Register	U2TB	216
0163h			
0164h	UART2 Transmit/Receive Control Register 0	U2C0	216
0165h	UART2 Transmit/Receive Control Register 1	U2C1	217
0166h	UART2 Receive Buffer Register	U2RB	217
0167h			
0168h			
0169h			
016Ah			
016Bh			
016Ch			
016Dh			
016Eh			
016Fh			
0170h			
0171h			
0172h			
0173h			
0174h			
0175h			
0176h			
0177h			
0178h			
0179h			
017Ah			
017Bh			
017Ch			
017Dh			
017Eh			
017Fh			
0180h			
0181h			
0182h			
0183h			
0184h			
0185h			
0186h			
0187h			
0188h			
0189h			
018Ah			
018Bh			
018Ch			
018Dh			
018Eh			
018Fh			
0190h			
0191h			
0192h			
0193h			
0194h			
0195h			
0196h			
0197h			
0198h			
0199h			
019Ah			
019Bh			
019Ch			
019Dh			
019Eh			
019Fh			

Address	Register	Symbol	Page
01A0h			
01A1h			
01A2h			
01A3h			
01A4h			
01A5h			
01A6h			
01A7h			
01A8h			
01A9h			
01AAh			
01ABh			
01ACh			
01ADh			
01AEh			
01AFh			
01B0h			
01B1h			
01B2h			
01B3h	Flash Memory Control Register 4	FMR4	253
01B4h			
01B5h	Flash Memory Control Register 1	FMR1	252
01B6h			
01B7h	Flash Memory Control Register 0	FMR0	249
01B8h			
01B9h			
01BAh			
01BBh			
01BCh			
01BDh			
01BEh			
01C0h			
01C1h			
01C2h			
01C3h			
01C4h			
01C5h			
01C6h			
01C7h			
01C8h			
01C9h			
01CAh			
01CBh			
01CCh			
01CDh			
01CEh			
01CFh			
01D0h			
01D1h			
01D2h			
01D3h			
01D4h			
01D5h			
01D6h			
01D7h			
01D8h			
01D9h			
01DAh			
01DBh			
01DCh			
01DDh			
01DEh			
01DFh			

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

Address	Register	Symbol	Page
01E0h			
01E1h			
01E2h			
01E3h			
01E4h			
01E5h			
01E6h			
01E7h			
01E8h			
01E9h			
01EAh			
01EBh			
01ECh			
01EDh			
01EEh			
01EFh			
01F0h			
01F1h			
01F2h			
01F3h			
01F4h			
01F5h			
01F6h			
01F7h			
01F8h			
01F9h			
01FAh			
01FBh			
01FCh			
01FDh			
01FEh			
01FFh			
0200h			
0201h			
0202h			
0203h			
0204h			
0205h			
0206h			
0207h			
0208h			
0209h			
020Ah			
020Bh			
020Ch			
020Dh			
020Eh			
020Fh			
0210h			
0211h			
0212h			
0213h			
0214h			
0215h			
0216h			
0217h			
0218h			
0219h			
021Ah			
021Bh			
021Ch			
021Dh			
021Eh			
021Fh			

Address	Register	Symbol	Page
0220h			
0221h			
0222h			
0223h			
0224h			
0225h			
0226h			
0227h			
0228h			
0229h			
022Ah			
022Bh			
022Ch			
022Dh			
022Eh			
022Fh			
0230h			
0231h			
0232h			
0233h			
0234h			
0235h			
0236h			
0237h			
0238h			
0239h			
023Ah			
023Bh			
023Ch			
023Dh			
023Eh			
023Fh			
0240h			
0241h			
0242h			
0243h			
0244h			
0245h			
0246h			
0247h			
0248h			
0249h			
024Ah			
024Bh			
024Ch			
024Dh			
024Eh			
024Fh			
0250h			
0251h			
0252h			
0253h			
0254h			
0255h			
0256h			
0257h			
0258h			
0259h			
025Ah			
025Bh			
025Ch			
025Dh			
025Eh			
025Fh			

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

Address	Register	Symbol	Page
0260h			
0261h			
0262h			
0263h			
0264h			
0265h			
0266h			
0267h			
0268h			
0269h			
026Ah			
026Bh			
026Ch			
026Dh			
026Eh			
026Fh			
0270h			
0271h			
0272h			
0273h			
0274h			
0275h			
0276h			
0277h			
0278h			
0279h			
027Ah			
027Bh			
027Ch			
027Dh			
027Eh			
027Fh			
0280h			
0281h			
0282h			
0283h			
0284h			
0285h			
0286h			
0287h			
0288h			
0289h			
028Ah			
028Bh			
028Ch			
028Dh			
028Eh			
028Fh			
0290h	Timer RF Register	TRF	202
0291h			
0292h			
0293h			
0294h			
0295h			
0296h			
0297h			
0298h			
0299h	Timer RF Control Register 2	TRFCR2	203
029Ah	Timer RF Control Register 0	TRFCR0	203
029Bh	Timer RF Control Register 1	TRFCR1	204
029Ch	Capture and Compare 0 Register	TRFM0	202
029Dh			
029Eh	Compare 1 Register	TRFM1	202
029Fh			

Address	Register	Symbol	Page
02A0h			
02A1h			
02A2h			
02A3h			
02A4h			
02A5h			
02A6h			
02A7h			
02A8h			
02A9h			
02AAh			
02ABh			
02ACh			
02ADh			
02AEh			
02AFh			
02B0h			
02B1h			
02B2h			
02B3h			
02B4h			
02B5h			
02B6h			
02B7h			
02B8h			
02B9h			
02BAh			
02BBh			
02BCh			
02BDh			
02BEh			
02BFh			
02C0h			
02C1h			
02C2h			
02C3h			
02C4h			
02C5h			
02C6h			
02C7h			
02C8h			
02C9h			
02CAh			
02CBh			
02CCh			
02CDh			
02CEh			
02CFh			
02D0h			
02D1h			
02D2h			
02D3h			
02D4h			
02D5h			
02D6h			
02D7h			
02D8h			
02D9h			
02DAh			
02DBh			
02DCh			
02DDh			
02DEh			
02DFh			

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

Address	Register	Symbol	Page
02E0h			
02E1h			
02E2h			
02E3h			
02E4h			
02E5h			
02E6h			
02E7h			
02E8h			
02E9h			
02EAh			
02EBh			
02ECh			
02EDh			
02EEh			
02EFh			
02F0h			
02F1h			
02F2h			
02F3h			
02F4h			
02F5h			
02F6h			
02F7h			
02F8h			
02F9h			
02FAh			
02FBh	Pin Select Register 4	PINSR4	39, 54, 73
02FCh			
02FDh	External Input Enable Register 2	INTEN2	122
02FEh	INT Input Filter Select Register 2	INTF2	123
02FFh	Timer RF Output Control Register	TRFOUT	204
FFFh	Option Function Select Register	OFS	26, 135, 140, 247

NOTE:

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

## **1. Overview**

### **1.1 Features**

The R8C/2G Group of single-chip MCUs incorporates the R8C/Tiny Series CPU core, employing sophisticated instructions for a high level of efficiency. With 1 Mbyte of address space, and it is capable of executing instructions at high speed. In addition, the CPU core boasts a multiplier for high-speed operation processing.

Power consumption is low, and the supported operating modes allow additional power control. These MCUs also use an anti-noise configuration to reduce emissions of electromagnetic noise and are designed to withstand EMI.

Integration of many peripheral functions, including multifunction timer and serial interface, reduces the number of system components.

#### **1.1.1 Applications**

Electric power meters, electronic household appliances, office equipment, audio equipment, consumer equipment, etc.

#### **1.1.2 Specifications**

Table 1.1 outlines the Specifications for R8C/2G Group.

**Table 1.1 Specifications for R8C/2G Group**

Item	Function	Specification
CPU	Central processing unit	R8C/Tiny series core <ul style="list-style-type: none"> <li>Number of fundamental instructions: 89</li> <li>Minimum instruction execution time: <ul style="list-style-type: none"> <li>125 ns (System clock = 8 MHz, VCC = 2.7 to 5.5 V)</li> <li>250 ns (System clock = 4 MHz, VCC = 2.2 to 5.5 V)</li> </ul> </li> <li>Multiplier: 16 bits × 16 bits → 32 bits</li> <li>Multiply-accumulate instruction: 16 bits × 16 bits + 32 bits → 32 bits</li> <li>Operation mode: Single-chip mode (address space: 1 Mbyte)</li> </ul>
Memory	ROM, RAM	Refer to <b>Table 1.2 Product List for R8C/2G Group</b> .
Power Supply Voltage Detection	Voltage detection circuit	<ul style="list-style-type: none"> <li>Power-on reset</li> <li>Voltage detection 3</li> </ul>
Comparator		<ul style="list-style-type: none"> <li>2 circuits (shared with voltage monitor 1 and voltage monitor 2)</li> <li>External reference voltage input is available</li> </ul>
I/O Ports		<ul style="list-style-type: none"> <li>Output-only: 1</li> <li>CMOS I/O ports: 27, selectable pull-up resistor</li> </ul>
Clock	Clock generation circuits	<ul style="list-style-type: none"> <li>2 circuits: On-chip oscillator (high-speed, low-speed) (high-speed on-chip oscillator has a frequency adjustment function), XCIN clock oscillation circuit (32 kHz)</li> <li>Frequency divider circuit: Dividing selectable 1, 2, 4, 8, and 16</li> <li>Low power consumption modes: <ul style="list-style-type: none"> <li>Standard operating mode (low-speed clock, high-speed on-chip oscillator, low-speed on-chip oscillator), wait mode, stop mode</li> </ul> </li> </ul>
		Real-time clock (timer RE)
Interrupts		<ul style="list-style-type: none"> <li>External: 5 sources, Internal: 17 sources, Software: 4 sources</li> <li>Priority levels: 7 levels</li> </ul>
Watchdog Timer		15 bits × 1 (with prescaler), reset start selectable
Timer	Timer RA	8 bits × 1 (with 8-bit prescaler) Timer mode (period timer), pulse output mode (output level inverted every period), event counter mode, pulse width measurement mode, pulse period measurement mode
	Timer RB	8 bits × 1 (with 8-bit prescaler) Timer mode (period timer), programmable waveform generation mode (PWM output), programmable one-shot generation mode, programmable wait one-shot generation mode
	Timer RE	8 bits × 1 Real-time clock mode (count seconds, minutes, hours, days of week), output compare mode
	Timer RF	16 bits × 1 (with capture/compare register pin and compare register pin) Input capture mode, output compare mode
Serial Interface	UART0, UART2	Clock synchronous serial I/O/UART × 2
LIN Module		Hardware LIN: 1 (timer RA, UART0)
Flash Memory		<ul style="list-style-type: none"> <li>Programming and erasure voltage: VCC = 2.7 to 5.5 V</li> <li>Programming and erasure endurance: 100 times</li> <li>Program security: ROM code protect, ID code check</li> <li>Debug functions: On-chip debug, on-board flash rewrite function</li> </ul>
Operating Frequency/Supply Voltage		System clock = 8 MHz (VCC = 2.7 to 5.5 V) System clock = 4 MHz (VCC = 2.2 to 5.5 V)
Current consumption		5 mA (VCC = 5 V, system clock = 8 MHz) 23 μA (VCC = 3 V, wait mode (low-speed on-chip oscillator on)) 0.7 μA (VCC = 3 V, stop mode, BGR trimming circuit disabled)
Operating Ambient Temperature		-20 to 85°C (N version) -40 to 85°C (D version) <sup>(1)</sup>
Package		32-pin LQFP Package code: PLQP0032GB-A (previous code: 32P6U-A)

NOTE:

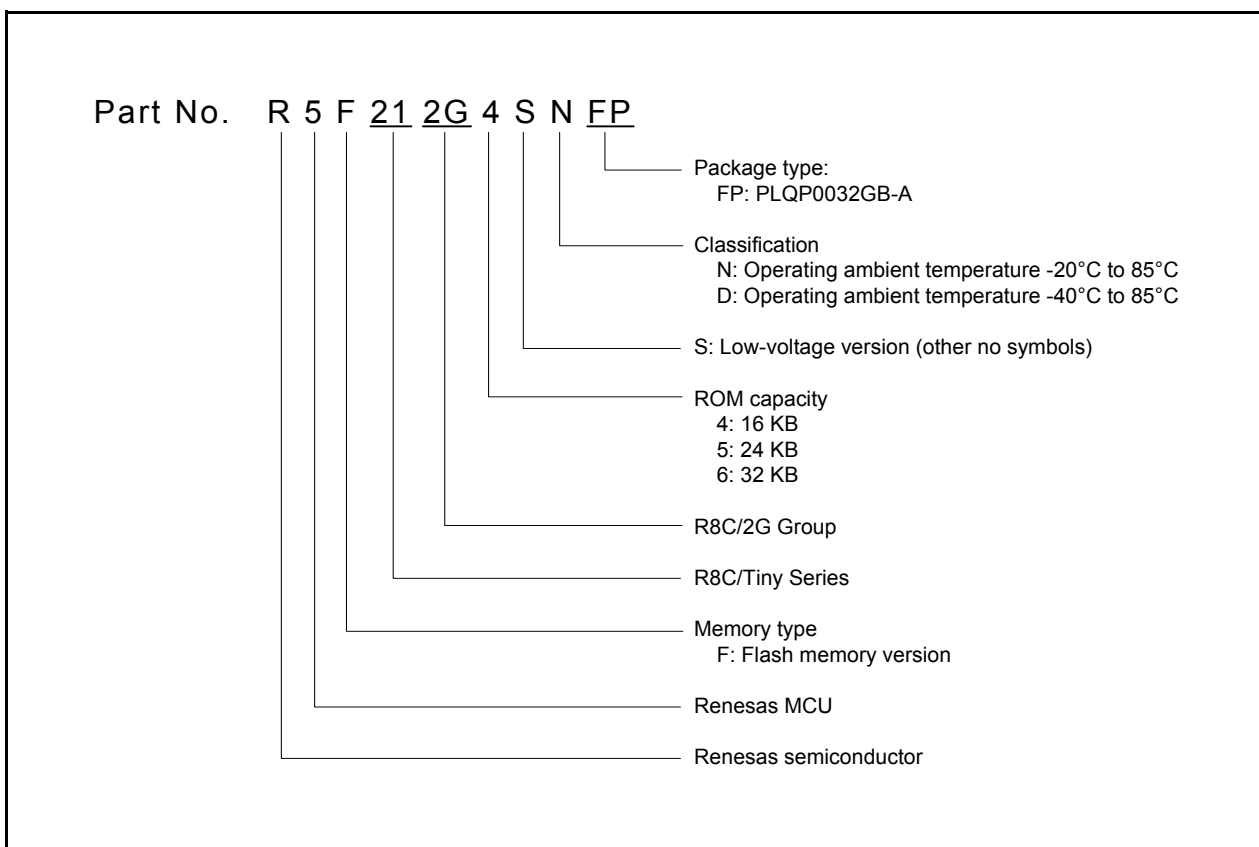
- Specify the D version if D version functions are to be used.

## 1.2 Product List

Table 1.2 lists Product List for R8C/2G Group, Figure 1.1 shows a Part Number, Memory Size, and Package of R8C/2G Group.

**Table 1.2 Product List for R8C/2G Group** **Current of Apr. 2008**

Part No.	ROM Capacity	RAM Capacity	Package Type	Remarks
R5F212G4SNFP	16 Kbytes	512 bytes	PLQP0032GB-A	N version
R5F212G5SNFP	24 Kbytes	1 Kbytes	PLQP0032GB-A	
R5F212G6SNFP	32 Kbytes	1 Kbytes	PLQP0032GB-A	
R5F212G4SDFP	16 Kbytes	512 bytes	PLQP0032GB-A	D version
R5F212G5SDFP	24 Kbytes	1 Kbytes	PLQP0032GB-A	
R5F212G6SDFP	32 Kbytes	1 Kbytes	PLQP0032GB-A	



**Figure 1.1 Part Number, Memory Size, and Package of R8C/2G Group**

### 1.3 Block Diagram

Figure 1.2 shows a Block Diagram.

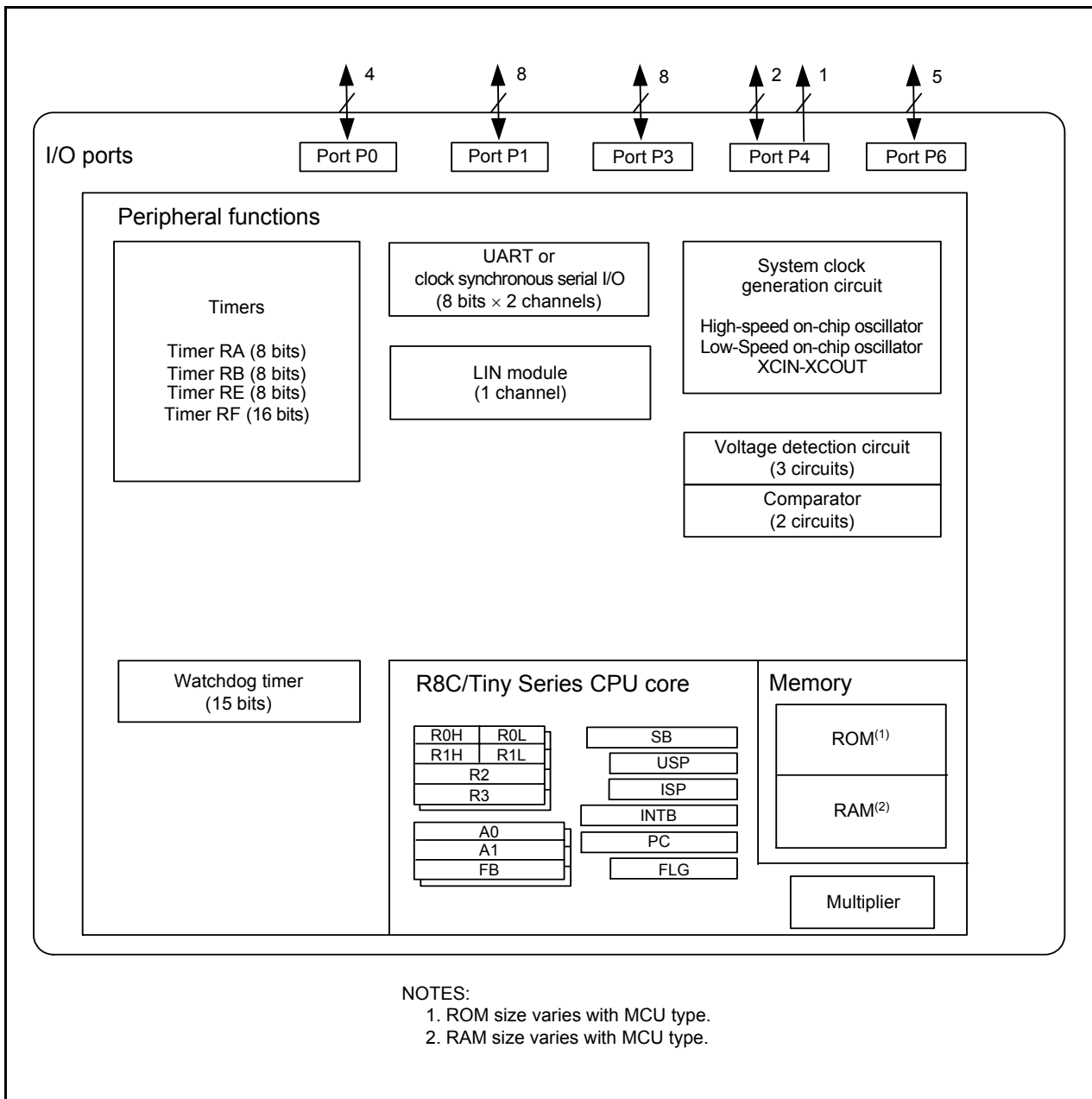


Figure 1.2 Block Diagram



### 1.4 Pin Assignment

Figure 1.3 shows Pin Assignment (Top View). Table 1.3 outlines the Pin Name Information by Pin Number.

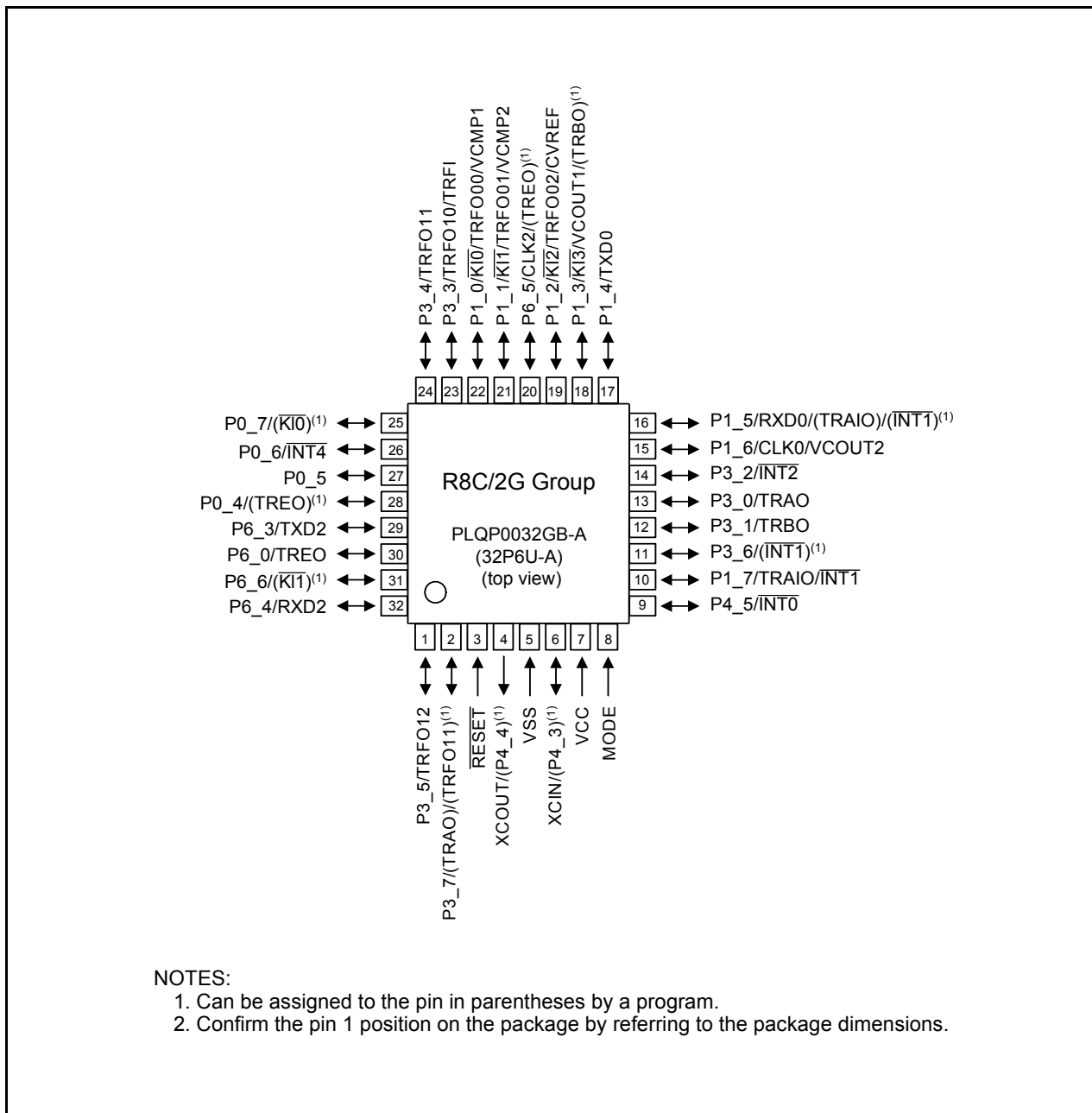


Figure 1.3 Pin Assignment (Top View)

**Table 1.3 Pin Name Information by Pin Number**

Pin Number	Control Pin	Port	I/O Pin Functions for of Peripheral Modules			
			Interrupt	Timer	Serial Interface	Comparator
1		P3_5		TRFO12		
2		P3_7		(TRAO)/(TRFO11) <sup>(1)</sup>		
3	RESET					
4	XCOU	(P4_4)				
5	VSS					
6	XCIN	(P4_3)				
7	VCC					
8	MODE					
9		P4_5	INT0			
10		P1_7	INT1	TRAIO		
11		P3_6	(INT1) <sup>(1)</sup>			
12		P3_1		TRBO		
13		P3_0		TRAO		
14		P3_2	INT2			
15		P1_6			CLK0	VCOUT2
16		P1_5	(INT1) <sup>(1)</sup>	(TRAIO) <sup>(1)</sup>	RXD0	
17		P1_4			TXD0	
18		P1_3	KI3	(TRBO) <sup>(1)</sup>		VCOUT1
19		P1_2	KI2	TRFO02		CVREF
20		P6_5		(TREQ) <sup>(1)</sup>	CLK2	
21		P1_1	KI1	TRFO01		VCMP2
22		P1_0	KI0	TRFO00		VCMP1
23		P3_3		TRFO10/TRFI		
24		P3_4		TRFO11		
25		P0_7	(KI0) <sup>(1)</sup>			
26		P0_6	INT4			
27		P0_5				
28		P0_4		(TREQ) <sup>(1)</sup>		
29		P6_3			TXD2	
30		P6_0		TREQ		
31		P6_6	(KI1) <sup>(1)</sup>			
32		P6_4			RXD2	

NOTE:

1. Can be assigned to the pin in parentheses by a program.

## 1.5 Pin Functions

Table 1.4 lists Pin Functions.

**Table 1.4 Pin Functions**

Type	Symbol	I/O Type	Description
Power supply input	VCC, VSS	–	Apply 2.2 V to 5.5 V to the VCC pin. Apply 0 V to the VSS pin.
Reset input	$\overline{\text{RESET}}$	I	Input “L” on this pin resets the MCU.
MODE	MODE	I	Connect this pin to VCC via a resistor.
XCIN clock input	XCIN	I	These pins are provided for XCIN clock generation circuit I/O. Connect a crystal oscillator between the XCIN and XCOU pins. <sup>(1)</sup> To use an external clock, input it to the XCIN pin and leave the XCOU pin open.
XCIN clock output	XCOU	O	
$\overline{\text{INT}}$ interrupt input	$\overline{\text{INT0}}$ to $\overline{\text{INT2}}$ , $\overline{\text{INT4}}$	I	$\overline{\text{INT}}$ interrupt input pins
Key input interrupt	$\overline{\text{KI0}}$ to $\overline{\text{KI3}}$	I	Key input interrupt input pins
Timer RA	TRAIO	I/O	Timer RA I/O pin
	TRAO	O	Timer RA output pin
Timer RB	TRBO	O	Timer RB output pin
Timer RE	TREO	O	Divided clock output pin
Timer RF	TRFI	I	Timer RF input pin
	TRFO00 to TRFO02, TRFO10 to TRFO12	O	Timer RF output pins
Serial interface	CLK0, CLK2	I/O	Clock I/O pin
	RXD0, RXD2	I	Serial data input pin
	TXD0, TXD2	O	Serial data output pin
Comparator	VCMP1, VCMP2	I	Analog input pins to comparator
	CVREF	I	Reference voltage input pin to comparator
	VCOUT1, VCOUT2	O	Comparator output pins
I/O port	P0_4 to P0_7, P1_0 to P1_7, P3_0 to P3_7, P4_3, P4_5, P6_0, P6_3 to P6_6	I/O	CMOS I/O ports. Each port has an I/O select direction register, allowing each pin in the port to be directed for input or output individually. Any port set to input can be set to use a pull-up resistor or not by a program.
Output port	P4_4	O	Output-only port

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

NOTE:

1. Refer to the oscillator manufacturer for oscillation characteristics.

## 2. Central Processing Unit (CPU)

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

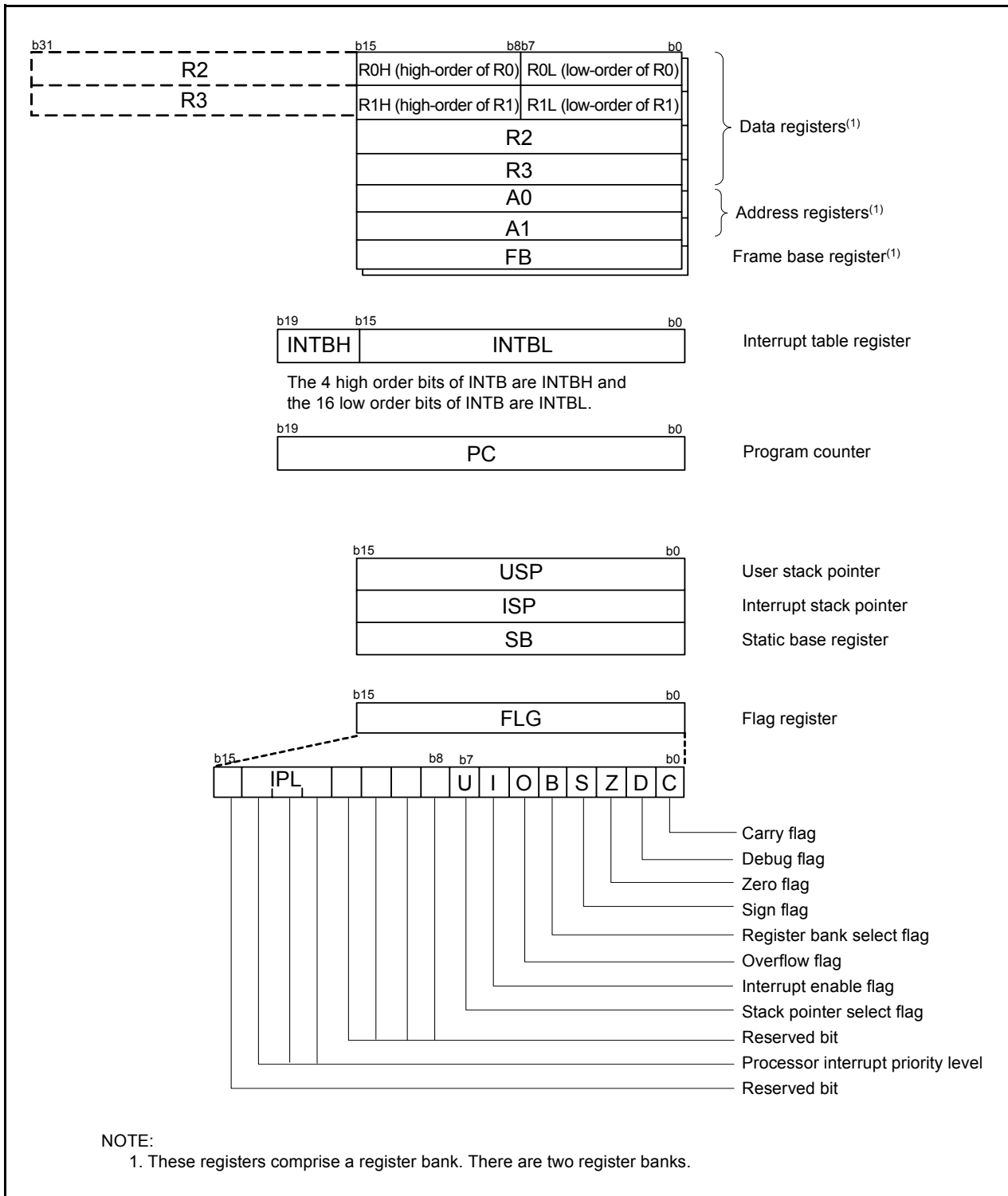


Figure 2.1 CPU Registers

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

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

## 2.2 Address Registers (A0 and A1)

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

## 2.3 Frame Base Register (FB)

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

## 2.4 Interrupt Table Register (INTB)

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

## 2.5 Program Counter (PC)

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

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

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

## 2.7 Static Base Register (SB)

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

## 2.8 Flag Register (FLG)

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

### 2.8.1 Carry Flag (C)

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

### 2.8.2 Debug Flag (D)

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

### 2.8.3 Zero Flag (Z)

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

### 2.8.4 Sign Flag (S)

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

### 2.8.5 Register Bank Select Flag (B)

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

### 2.8.6 Overflow Flag (O)

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

### 2.8.7 Interrupt Enable Flag (I)

The I flag enables maskable interrupts.

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

### 2.8.8 Stack Pointer Select Flag (U)

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

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

### 2.8.9 Processor Interrupt Priority Level (IPL)

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

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

### 2.8.10 Reserved Bit

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

### 3. Memory

Figure 3.1 is a Memory Map of R8C/2G Group. The R8C/2G group has 1 Mbyte of address space from addresses 00000h to FFFFFh.

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

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

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

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

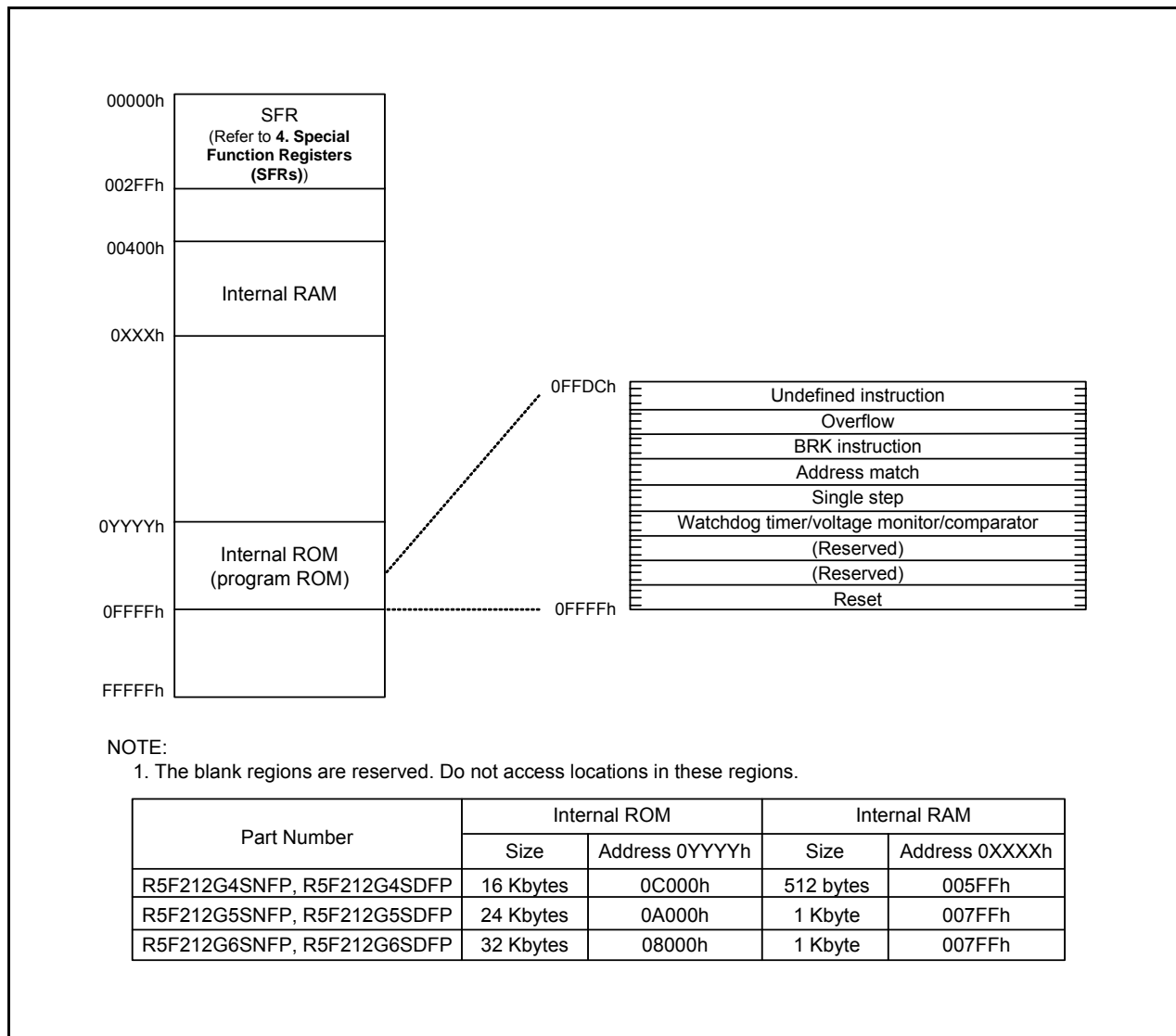


Figure 3.1 Memory Map of R8C/2G Group

## 4. Special Function Registers (SFRs)

An SFR (special function register) is a control register for a peripheral function. Tables 4.1 to 4.12 list the special function registers.

**Table 4.1 SFR Information (1)(1)**

Address	Register	Symbol	After reset
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	00h
0005h	Processor Mode Register 1	PM1	00h
0006h	System Clock Control Register 0	CM0	01011000b
0007h	System Clock Control Register 1	CM1	00h
0008h			
0009h			
000Ah	Protect Register	PRCR	00h
000Bh			
000Ch	System Clock Select Register	OCD	00000100b
000Dh	Watchdog Timer Reset Register	WDTR	XXh
000Eh	Watchdog Timer Start Register	WDTS	XXh
000Fh	Watchdog Timer Control Register	WDC	00X11111b
0010h	Address Match Interrupt Register 0	RMAD0	00h
0011h			00h
0012h			00h
0013h	Address Match Interrupt Enable Register	AIER	00h
0014h	Address Match Interrupt Register 1	RMAD1	00h
0015h			00h
0016h			00h
0017h			
0018h			
0019h			
001Ah			
001Bh			
001Ch	Count Source Protection Mode Register	CSPR	00h 10000000b(2)
001Dh			
001Eh			
001Fh			
0020h	High-Speed On-Chip Oscillator Control Register 0	HRA0	00h
0021h	High-Speed On-Chip Oscillator Control Register 1	HRA1	When Shipping
0022h	High-Speed On-Chip Oscillator Control Register 2	HRA2	00h
0023h			
0024h			
0025h			
0026h			
0027h			
0028h	Clock Prescaler Reset Flag	CPSRF	00h
0029h	High-Speed On-Chip Oscillator Control Register 4	FRA4	When Shipping
002Ah			
002Bh	High-Speed On-Chip Oscillator Control Register 6	FRA6	When Shipping
002Ch			
002Dh			
002Eh	BGR Trimming Auxiliary Register A	BGRTRMA	When Shipping
002Fh	BGR Trimming Auxiliary Register B	BGRTRMB	When Shipping

X: Undefined

NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. The CSPROINI bit in the OFS register is set to 0.



**Table 4.2 SFR Information (2)(1)**

Address	Register	Symbol	After reset
0030h			
0031h	Voltage Detection Register 1(2)	VCA1	00001000b
0032h	Voltage Detection Register 2(2)	VCA2	00h(3) 00100000b(4)
0033h			
0034h			
0035h			
0036h	Voltage Monitor 1 Circuit Control Register(5)	VW1C	00001010b
0037h	Voltage Monitor 2 Circuit Control Register(5)	VW2C	00000010b
0038h	Voltage Monitor 0 Circuit Control Register(2)	VW0C	1000X010b(3) 1100X011b(4)
0039h			
003Ah			
003Bh	Voltage Detection Circuit External Input Control Register	VCAB	00h
003Ch	Comparator Mode Register	ALCMR	00h
003Dh	Voltage Monitor Circuit Edge Select Register	VCAC	00h
003Eh	BGR Control Register	BGRCR	00h
003Fh	BGR Trimming Register	BGRTRM	When Shipping
0040h			
0041h	Comparator 1 Interrupt Control Register	VCMP1IC	XXXXX000b
0042h	Comparator 2 Interrupt Control Register	VCMP2IC	XXXXX000b
0043h			
0044h			
0045h			
0046h			
0047h			
0048h			
0049h			
004Ah	Timer RE Interrupt Control Register	TREIC	XXXXX000b
004Bh	UART2 Transmit Interrupt Control Register	S2TIC	XXXXX000b
004Ch	UART2 Receive Interrupt Control Register	S2RIC	XXXXX000b
004Dh	Key Input Interrupt Control Register	KUPIC	XXXXX000b
004Eh			
004Fh			
0050h	Compare 1 Interrupt Control Register	CMP1IC	XXXXX000b
0051h	UART0 Transmit Interrupt Control Register	S0TIC	XXXXX000b
0052h	UART0 Receive Interrupt Control Register	S0RIC	XXXXX000b
0053h			
0054h			
0055h	INT2 Interrupt Control Register	INT2IC	XX00X000b
0056h	Timer RA Interrupt Control Register	TRAIC	XXXXX000b
0057h			
0058h	Timer RB Interrupt Control Register	TRBIC	XXXXX000b
0059h	INT1 Interrupt Control Register	INT1IC	XX00X000b
005Ah			
005Bh	Timer RF Interrupt Control Register	TRFIC	XXXXX000b
005Ch	Compare 0 Interrupt Control Register	CMP0IC	XXXXX000b
005Dh	INT0 Interrupt Control Register	INT0IC	XX00X000b
005Eh	INT4 Interrupt Control Register	INT4IC	XX00X000b
005Fh	Capture Interrupt Control Register	CAPIC	XXXXX000b
0060h			
0061h			
0062h			
0063h			
0064h			
0065h			
0066h			
0067h			
0068h			
0069h			
006Ah			
006Bh			
006Ch			
006Dh			
006Eh			
006Fh			

X: Undefined

NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. Software reset, watchdog timer reset, voltage monitor 1 reset, or voltage monitor 2 reset do not affect this register.
3. The LVD0ON bit in the OFS register is set to 1 and hardware reset.
4. Power-on reset, voltage monitor 0 reset, or the LVD0ON bit in the OFS register is set to 0 and hardware reset.
5. Software reset, watchdog timer reset, voltage monitor 1 reset, or voltage monitor 2 reset do not affect b2 and b3.

**Table 4.3 SFR Information (3)(1)**

Address	Register	Symbol	After reset
0070h			
0071h			
0072h			
0073h			
0074h			
0075h			
0076h			
0077h			
0078h			
0079h			
007Ah			
007Bh			
007Ch			
007Dh			
007Eh			
007Fh			
0080h			
0081h			
0082h			
0083h			
0084h			
0085h			
0086h			
0087h			
0088h			
0089h			
008Ah			
008Bh			
008Ch			
008Dh			
008Eh			
008Fh			
0090h			
0091h			
0092h			
0093h			
0094h			
0095h			
0096h			
0097h			
0098h			
0099h			
009Ah			
009Bh			
009Ch			
009Dh			
009Eh			
009Fh			
00A0h	UART0 Transmit/Receive Mode Register	U0MR	00h
00A1h	UART0 Bit Rate Register	U0BRG	XXh
00A2h	UART0 Transmit Buffer Register	U0TB	XXh
00A3h			XXh
00A4h	UART0 Transmit/Receive Control Register 0	U0C0	00001000b
00A5h	UART0 Transmit/Receive Control Register 1	U0C1	00000010b
00A6h	UART0 Receive Buffer Register	U0RB	XXh
00A7h			XXh
00A8h			
00A9h			
00AAh			
00ABh			
00ACh			
00ADh			
00AEh			
00AFh			

X: Undefined

NOTE:

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

**Table 4.4 SFR Information (4)<sup>(1)</sup>**

Address	Register	Symbol	After reset
00B0h			
00B1h			
00B2h			
00B3h			
00B4h			
00B5h			
00B6h			
00B7h			
00B8h			
00B9h			
00BAh			
00BBh			
00BCh			
00BDh			
00BEh			
00BFh			
00C0h			
00C1h			
00C2h			
00C3h			
00C4h			
00C5h			
00C6h			
00C7h			
00C8h			
00C9h			
00CAh			
00CBh			
00CCh			
00CDh			
00CEh			
00CFh			
00D0h			
00D1h			
00D2h			
00D3h			
00D4h			
00D5h			
00D6h			
00D7h			
00D8h			
00D9h			
00DAh			
00DBh			
00DCh			
00DDh			
00DEh			
00DFh			
00E0h	Port P0 Register	P0	00h
00E1h	Port P1 Register	P1	00h
00E2h	Port P0 Direction Register	PD0	00h
00E3h	Port P1 Direction Register	PD1	00h
00E4h			
00E5h	Port P3 Register	P3	00h
00E6h			
00E7h	Port P3 Direction Register	PD3	00h
00E8h	Port P4 Register	P4	00h
00E9h			
00EAh	Port P4 Direction Register	PD4	00h
00EBh			
00ECh	Port P6 Register	P6	00h
00EDh			
00EEh	Port P6 Direction Register	PD6	00h
00EFh			

X: Undefined

NOTE:

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

**Table 4.5 SFR Information (5)(1)**

Address	Register	Symbol	After reset
00F0h			
00F1h			
00F2h			
00F3h			
00F4h			
00F5h			
00F6h	Pin Select Register 2	PINSR2	00h
00F7h	Pin Select Register 3	PINSR3	00h
00F8h	Port Mode Register	PMR	00h
00F9h	External Input Enable Register	INTEN	00h
00FAh	INT Input Filter Select Register	INTF	00h
00FBh	Key Input Enable Register	KIEN	00h
00FCh	Pull-Up Control Register 0	PUR0	00h
00FDh	Pull-Up Control Register 1	PUR1	00h
00FEh			
00FFh			
0100h	Timer RA Control Register	TRACR	00h
0101h	Timer RA I/O Control Register	TRAIOC	00h
0102h	Timer RA Mode Register	TRAMR	00h
0103h	Timer RA Prescaler Register	TRAPRE	FFh
0104h	Timer RA Register	TRA	FFh
0105h			
0106h	LIN Control Register	LINCR	00h
0107h	LIN Status Register	LINST	00h
0108h	Timer RB Control Register	TRBCR	00h
0109h	Timer RB One-Shot Control Register	TRBOCR	00h
010Ah	Timer RB I/O Control Register	TRBIOC	00h
010Bh	Timer RB Mode Register	TRBMR	00h
010Ch	Timer RB Prescaler Register	TRBPRE	FFh
010Dh	Timer RB Secondary Register	TRBSC	FFh
010Eh	Timer RB Primary Register	TRBPR	FFh
010Fh			
0110h			
0111h			
0112h			
0113h			
0114h			
0115h			
0116h			
0117h			
0118h	Timer RE Second Data Register / Counter Data Register	TRESEC	XXh
0119h	Timer RE Minute Data Register / Compare Data Register	TREMIN	XXh
011Ah	Timer RE Hour Data Register	TREHR	X0XXXXXXb
011Bh	Timer RE Day of Week Data Register	TREWK	X0000XXXb
011Ch	Timer RE Control Register 1	TRECR1	XXX0X0X0b
011Dh	Timer RE Control Register 2	TRECR2	00XXXXXXb
011Eh	Timer RE Count Source Select Register	TRECSR	00001000b
011Fh	Timer RE Real-Time Clock Precision Adjust Register	TREOPR	00h
0120h			
0121h			
0122h			
0123h			
0124h			
0125h			
0126h			
0127h			
0128h			
0129h			
012Ah			
012Bh			
012Ch			
012Dh			
012Eh			
012Fh			

X: Undefined

NOTE:

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

**Table 4.6 SFR Information (6)(1)**

Address	Register	Symbol	After reset
0130h			
0131h			
0132h			
0133h			
0134h			
0135h			
0136h			
0137h			
0138h			
0139h			
013Ah			
013Bh			
013Ch			
013Dh			
013Eh			
013Fh			
0140h			
0141h			
0142h			
0143h			
0144h			
0145h			
0146h			
0147h			
0148h			
0149h			
014Ah			
014Bh			
014Ch			
014Dh			
014Eh			
014Fh			
0150h			
0151h			
0152h			
0153h			
0154h			
0155h			
0156h			
0157h			
0158h			
0159h			
015Ah			
015Bh			
015Ch			
015Dh			
015Eh			
015Fh			
0160h	UART2 Transmit/Receive Mode Register	U2MR	00h
0161h	UART2 Bit Rate Register	U2BRG	XXh
0162h	UART2 Transmit Buffer Register	U2TB	XXh
0163h			XXh
0164h	UART2 Transmit/Receive Control Register 0	U2C0	00001000b
0165h	UART2 Transmit/Receive Control Register 1	U2C1	00000010b
0166h	UART2 Receive Buffer Register	U2RB	XXh
0167h			XXh
0168h			
0169h			
016Ah			
016Bh			
016Ch			
016Dh			
016Eh			
016Fh			

X: Undefined

NOTE:

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

**Table 4.7 SFR Information (7)(1)**

Address	Register	Symbol	After reset
0170h			
0171h			
0172h			
0173h			
0174h			
0175h			
0176h			
0177h			
0178h			
0179h			
017Ah			
017Bh			
017Ch			
017Dh			
017Eh			
017Fh			
0180h			
0181h			
0182h			
0183h			
0184h			
0185h			
0186h			
0187h			
0188h			
0189h			
018Ah			
018Bh			
018Ch			
018Dh			
018Eh			
018Fh			
0190h			
0191h			
0192h			
0193h			
0194h			
0195h			
0196h			
0197h			
0198h			
0199h			
019Ah			
019Bh			
019Ch			
019Dh			
019Eh			
019Fh			
01A0h			
01A1h			
01A2h			
01A3h			
01A4h			
01A5h			
01A6h			
01A7h			
01A8h			
01A9h			
01AAh			
01ABh			
01ACh			
01ADh			
01AEh			
01AFh			

X: Undefined

NOTE:

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

**Table 4.8 SFR Information (8)(1)**

Address	Register	Symbol	After reset
01B0h			
01B1h			
01B2h			
01B3h	Flash Memory Control Register 4	FMR4	01000000b
01B4h			
01B5h	Flash Memory Control Register 1	FMR1	1000000Xb
01B6h			
01B7h	Flash Memory Control Register 0	FMR0	00000001b
01B8h			
01B9h			
01BAh			
01BBh			
01BCh			
01BDh			
01BEh			
01BFh			
01C0h			
01C1h			
01C2h			
01C3h			
01C4h			
01C5h			
01C6h			
01C7h			
01C8h			
01C9h			
01CAh			
01CBh			
01CCh			
01CDh			
01CEh			
01CFh			
01D0h			
01D1h			
01D2h			
01D3h			
01D4h			
01D5h			
01D6h			
01D7h			
01D8h			
01D9h			
01DAh			
01DBh			
01DCh			
01DDh			
01DEh			
01DFh			
01E0h			
01E1h			
01E2h			
01E3h			
01E4h			
01E5h			
01E6h			
01E7h			
01E8h			
01E9h			
01EAh			
01EBh			
01ECh			
01EDh			
01EEh			
01EFh			

X: Undefined

NOTE:

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

**Table 4.9 SFR Information (9)<sup>(1)</sup>**

Address	Register	Symbol	After reset
01F0h			
01F1h			
01F2h			
01F3h			
01F4h			
01F5h			
01F6h			
01F7h			
01F8h			
01F9h			
01FAh			
01FBh			
01FCh			
01FDh			
01FEh			
01FFh			
0200h			
0201h			
0202h			
0203h			
0204h			
0205h			
0206h			
0207h			
0208h			
0209h			
020Ah			
020Bh			
020Ch			
020Dh			
020Eh			
020Fh			
0210h			
0211h			
0212h			
0213h			
0214h			
0215h			
0216h			
0217h			
0218h			
0219h			
021Ah			
021Bh			
021Ch			
021Dh			
021Eh			
021Fh			
0220h			
0221h			
0222h			
0223h			
0224h			
0225h			
0226h			
0227h			
0228h			
0229h			
022Ah			
022Bh			
022Ch			
022Dh			
022Eh			
022Fh			

X: Undefined

NOTE:

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



**Table 4.10 SFR Information (10)(1)**

Address	Register	Symbol	After reset
0230h			
0231h			
0232h			
0233h			
0234h			
0235h			
0236h			
0237h			
0238h			
0239h			
023Ah			
023Bh			
023Ch			
023Dh			
023Eh			
023Fh			
0240h			
0241h			
0242h			
0243h			
0244h			
0245h			
0246h			
0247h			
0248h			
0249h			
024Ah			
024Bh			
024Ch			
024Dh			
024Eh			
024Fh			
0250h			
0251h			
0252h			
0253h			
0254h			
0255h			
0256h			
0257h			
0258h			
0259h			
025Ah			
025Bh			
025Ch			
025Dh			
025Eh			
025Fh			
0260h			
0261h			
0262h			
0263h			
0264h			
0265h			
0266h			
0267h			
0268h			
0269h			
026Ah			
026Bh			
026Ch			
026Dh			
026Eh			
026Fh			

X: Undefined

NOTE:

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

**Table 4.11 SFR Information (11)(1)**

Address	Register	Symbol	After reset
0270h			
0271h			
0272h			
0273h			
0274h			
0275h			
0276h			
0277h			
0278h			
0279h			
027Ah			
027Bh			
027Ch			
027Dh			
027Eh			
027Fh			
0280h			
0281h			
0282h			
0283h			
0284h			
0285h			
0286h			
0287h			
0288h			
0289h			
028Ah			
028Bh			
028Ch			
028Dh			
028Eh			
028Fh			
0290h	Timer RF Register	TRF	00h
0291h			00h
0292h			
0293h			
0294h			
0295h			
0296h			
0297h			
0298h			
0299h	Timer RF Control Register 2	TRFCR2	00h
029Ah	Timer RF Control Register 0	TRFCR0	00h
029Bh	Timer RF Control Register 1	TRFCR1	00h
029Ch	Capture and Compare 0 Register	TRFM0	0000h <sup>(2)</sup>
029Dh			FFFFh <sup>(3)</sup>
029Eh	Compare 1 Register	TRFM1	FFh
029Fh			FFh
02A0h			
02A1h			
02A2h			
02A3h			
02A4h			
02A5h			
02A6h			
02A7h			
02A8h			
02A9h			
02AAh			
02ABh			
02ACh			
02ADh			
02AEh			
02AFh			

X: Undefined

## NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. After input capture mode.
3. After output compare mode.

Table 4.12 SFR Information (12)(1)

Address	Register	Symbol	After reset
02B0h			
02B1h			
02B2h			
02B3h			
02B4h			
02B5h			
02B6h			
02B7h			
02B8h			
02B9h			
02BAh			
02BBh			
02BCh			
02BDh			
02BEh			
02BFh			
02C0h			
02C1h			
02C2h			
02C3h			
02C4h			
02C5h			
02C6h			
02C7h			
02C8h			
02C9h			
02CAh			
02CBh			
02CCh			
02CDh			
02CEh			
02CFh			
02D0h			
02D1h			
02D2h			
02D3h			
02D4h			
02D5h			
02D6h			
02D7h			
02D8h			
02D9h			
02DAh			
02DBh			
02DCh			
02DDh			
02DEh			
02DFh			
02E0h			
02EFh			
02F0h			
02F1h			
02F2h			
02F3h			
02F4h			
02F5h			
02F6h			
02F7h			
02F8h			
02F9h			
02FAh			
02FBh	Pin Select Register 4	PINSR4	00h
02FCh			
02FDh	External Input Enable Register 2	INTEN2	00h
02FEh	INT Input Filter Select Register 2	INTF2	00h
02FFh	Timer RF Output Control Register	TRFOUT	00h
FFFh	Option Function Select Register	OFS	(Note 2)

X: Undefined

NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. The OFS register cannot be changed by a program. Use a flash programmer to write to it.

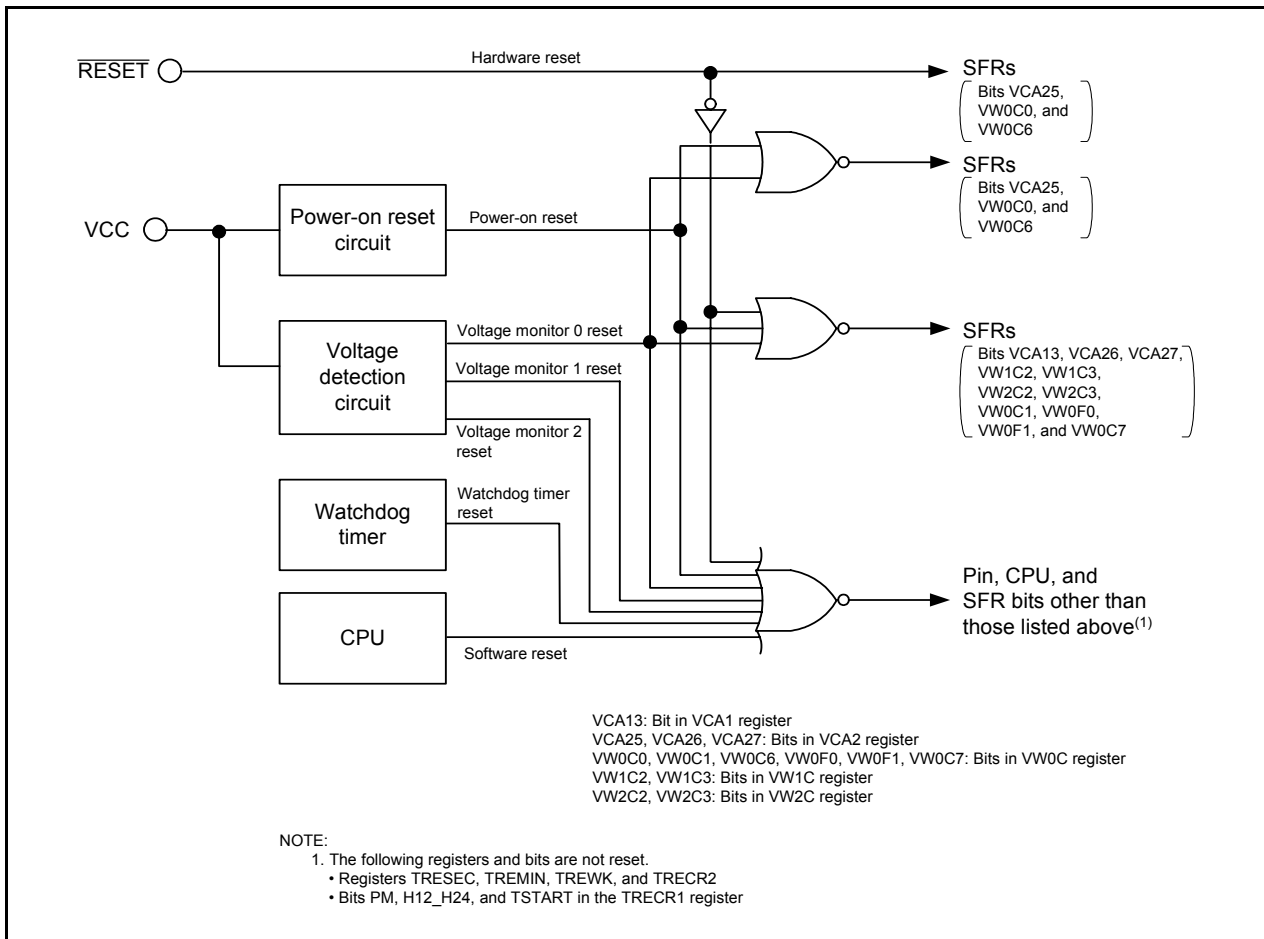
## 5. Resets

The following resets are implemented: hardware reset, power-on reset, voltage monitor 0 reset, voltage monitor 1 reset, voltage monitor 2 reset, watchdog timer reset, and software reset.

Table 5.1 lists the Reset Names and Sources. Figure 5.1 lists the Block Diagram of Reset Circuit.

**Table 5.1 Reset Names and Sources**

Reset Name	Source
Hardware reset	Input voltage of $\overline{\text{RESET}}$ pin is held "L"
Power-on reset	VCC rises
Voltage monitor 0 reset	VCC falls (monitor voltage: Vdet0)
Voltage monitor 1 reset	VCC falls (monitor voltage: Vdet1)
Voltage monitor 2 reset	VCC falls (monitor voltage: Vdet2)
Watchdog timer reset	Underflow of watchdog timer
Software reset	Write 1 to PM03 bit in PM0 register

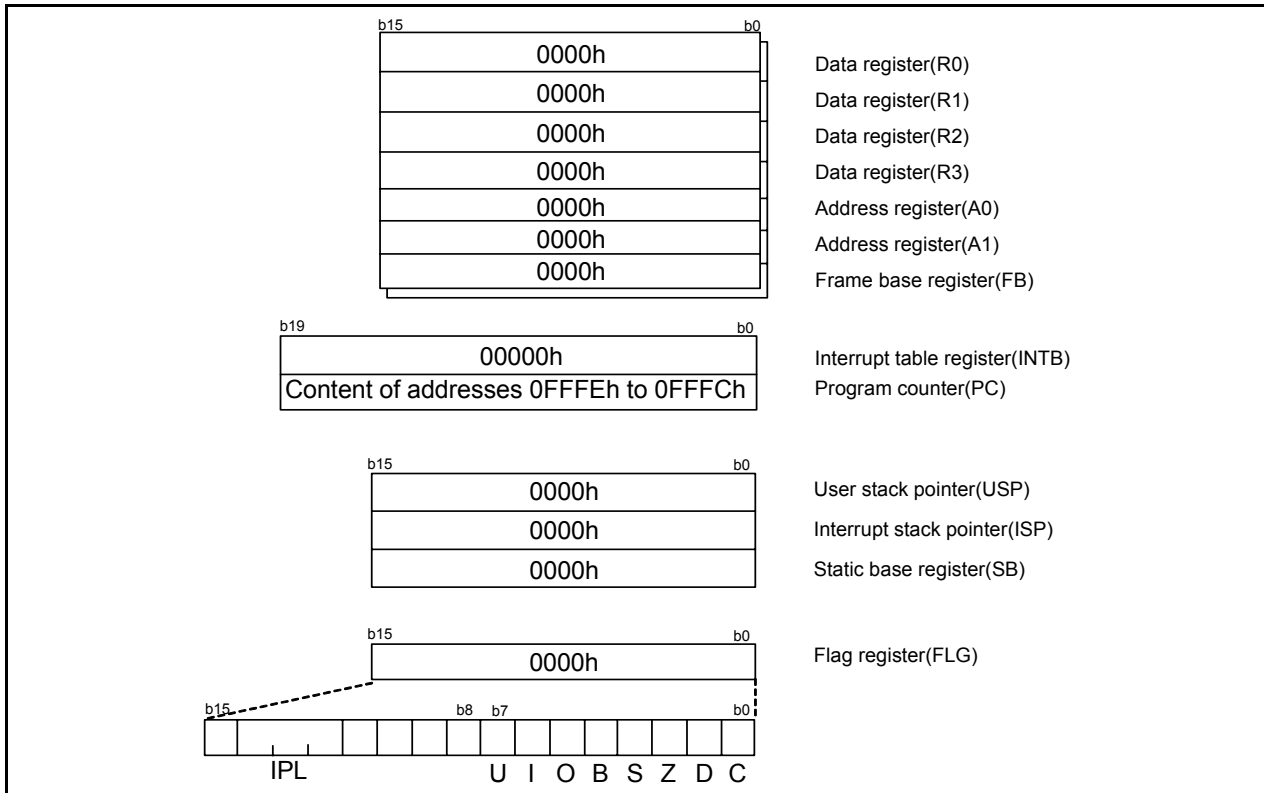


**Figure 5.1 Block Diagram of Reset Circuit**

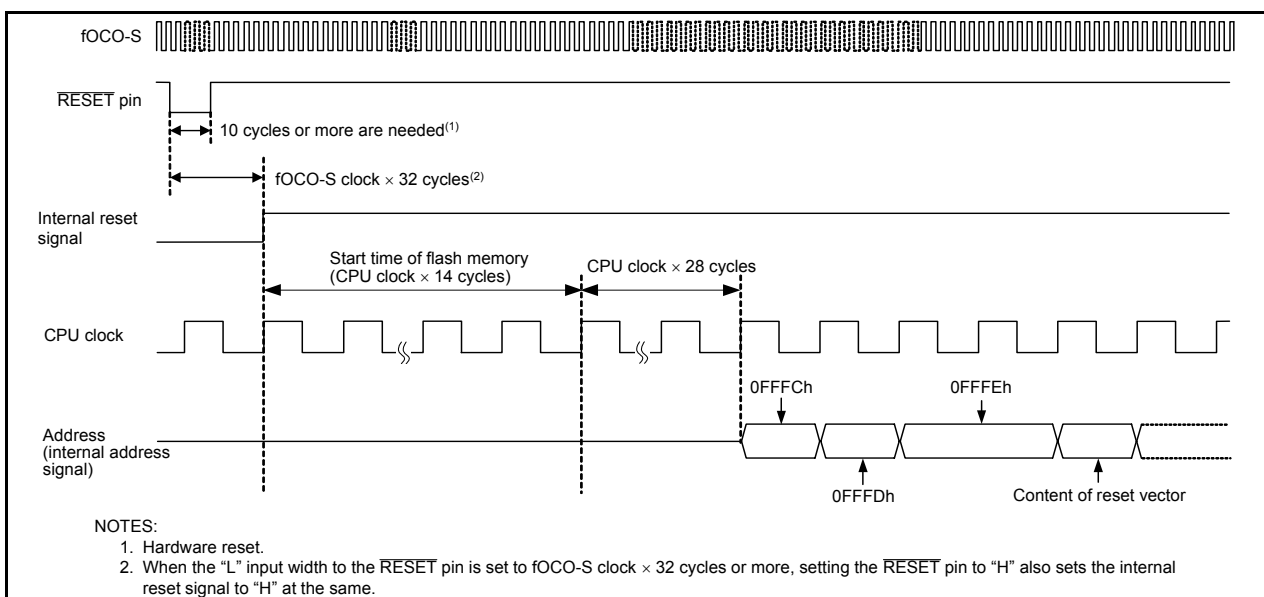
Table 5.2 shows the Pin Functions while  $\overline{\text{RESET}}$  Pin Level is “L”, Figure 5.2 shows the CPU Register Status after Reset, Figure 5.3 shows the Reset Sequence, and Figure 5.4 shows the OFS Register.

**Table 5.2 Pin Functions while  $\overline{\text{RESET}}$  Pin Level is “L”**

Pin Name	Pin Functions
P0_4 to P0_7	Input port
P1, P3	Input port
P4_3, P4_5	Input port
P4_4	Output port
P6_0, P6_3 to P6_6	Input port



**Figure 5.2 CPU Register Status after Reset**



**Figure 5.3 Reset Sequence**

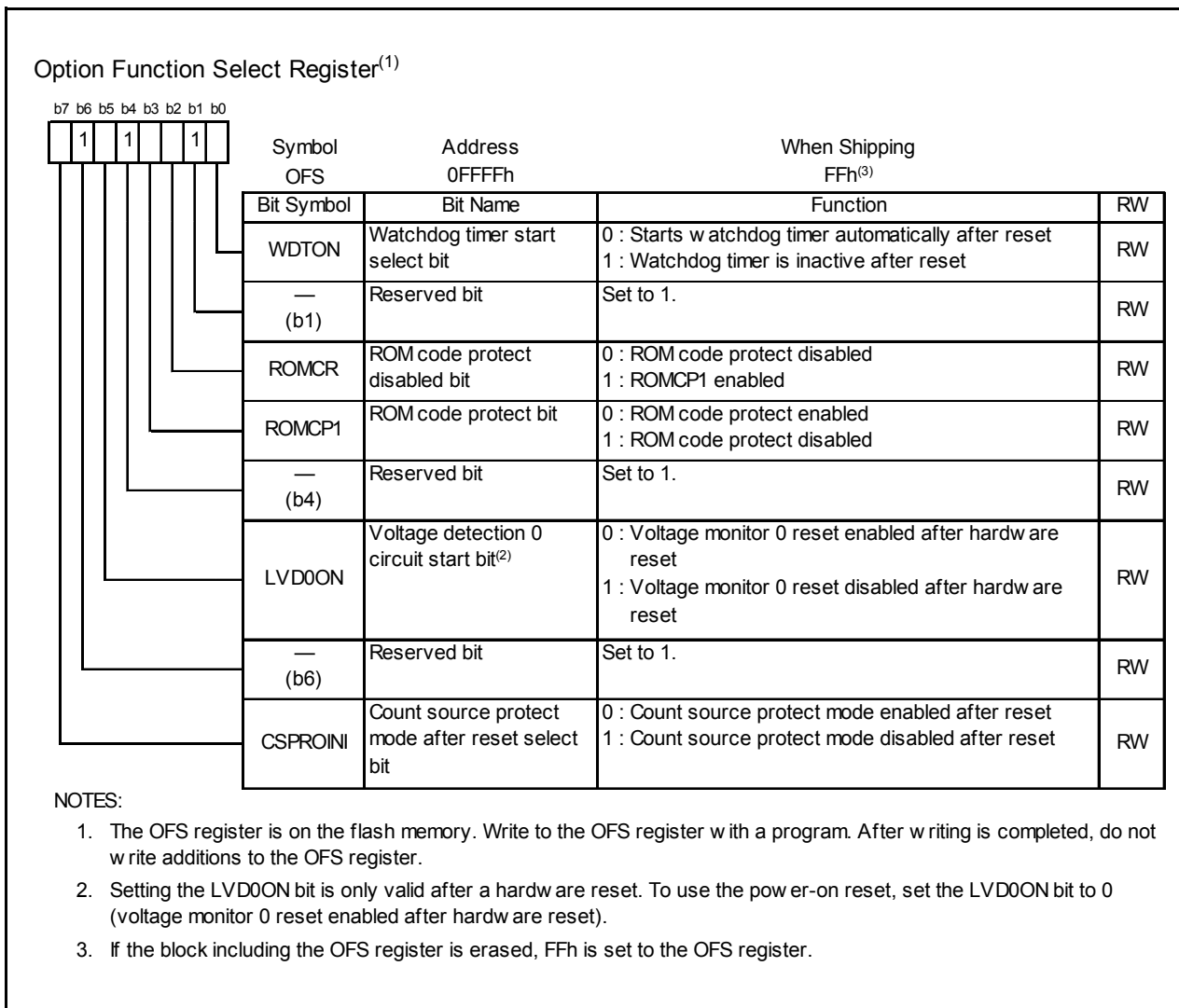


Figure 5.4 OFS Register

## 5.1 Hardware Reset

A reset is applied using the  $\overline{\text{RESET}}$  pin. When an “L” signal is applied to the  $\overline{\text{RESET}}$  pin while the supply voltage meets the recommended operating conditions, pins, CPU, and SFRs are all reset (refer to **Table 5.2 Pin Functions while RESET Pin Level is “L”**). When the input level applied to the  $\overline{\text{RESET}}$  pin changes from “L” to “H”, a program is executed beginning with the address indicated by the reset vector. After reset, the low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock.

Refer to **4. Special Function Registers (SFRs)** for the state of the SFRs after reset.

The internal RAM is not reset. If the  $\overline{\text{RESET}}$  pin is pulled “L” while writing to the internal RAM is in progress, the contents of internal RAM will be undefined.

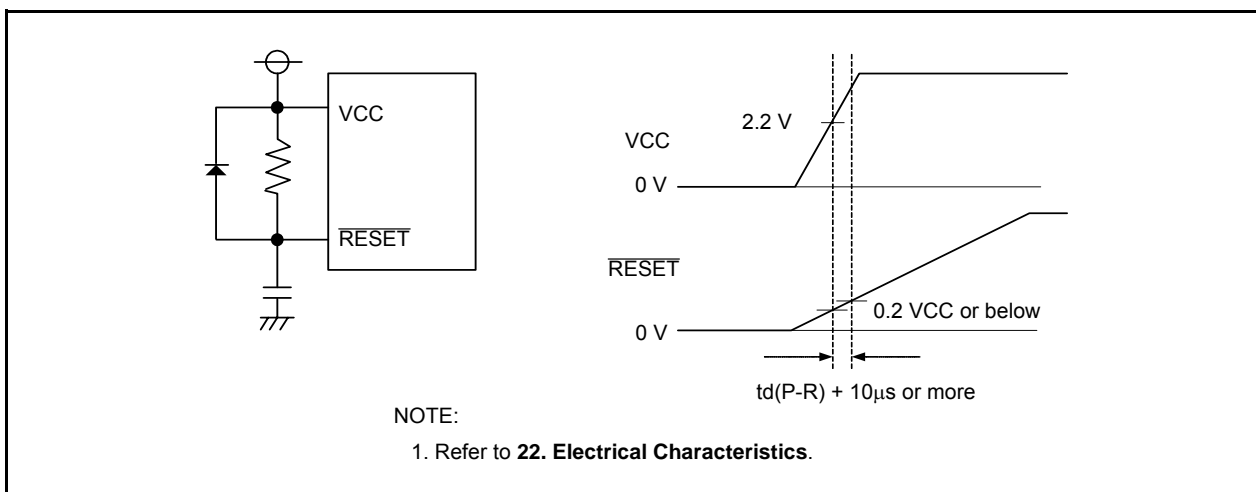
Figure 5.5 shows an Example of Hardware Reset Circuit and Operation and Figure 5.6 shows an Example of Hardware Reset Circuit (Usage Example of External Supply Voltage Detection Circuit) and Operation.

### 5.1.1 When Power Supply is Stable

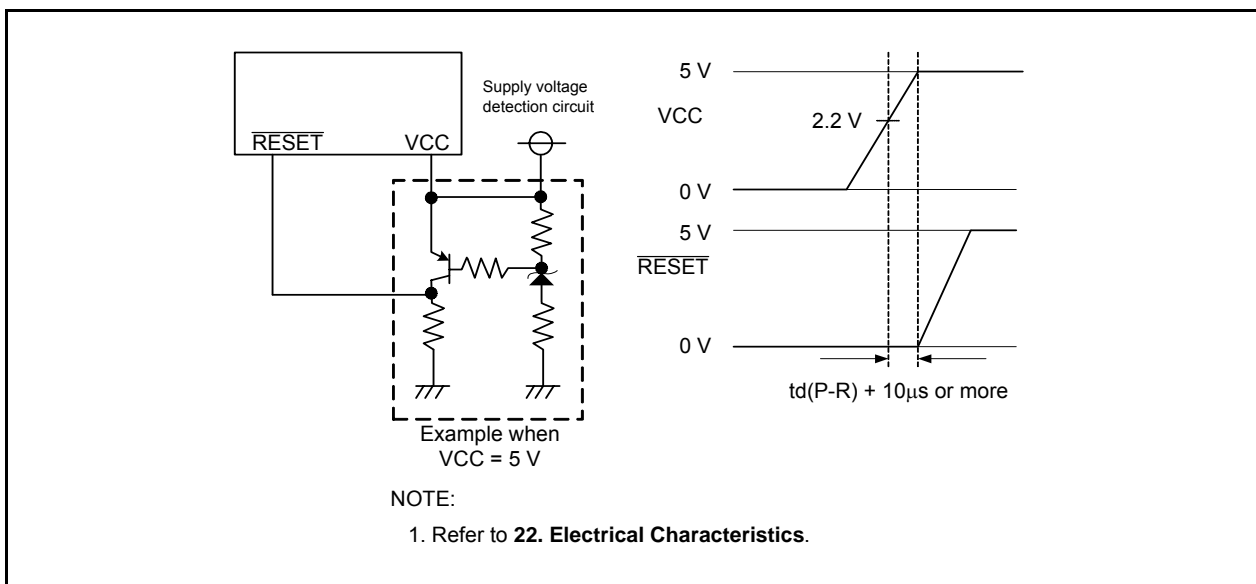
- (1) Apply “L” to the  $\overline{\text{RESET}}$  pin.
- (2) Wait for 10  $\mu\text{s}$ .
- (3) Apply “H” to the  $\overline{\text{RESET}}$  pin.

### 5.1.2 Power On

- (1) Apply “L” to the  $\overline{\text{RESET}}$  pin.
- (2) Let the supply voltage increase until it meets the recommended operating conditions.
- (3) Wait for  $t_{d(P-R)}$  or more to allow the internal power supply to stabilize (refer to **22. Electrical Characteristics**).
- (4) Wait for 10  $\mu\text{s}$ .
- (5) Apply “H” to the  $\overline{\text{RESET}}$  pin.



**Figure 5.5 Example of Hardware Reset Circuit and Operation**



**Figure 5.6 Example of Hardware Reset Circuit (Usage Example of External Supply Voltage Detection Circuit) and Operation**



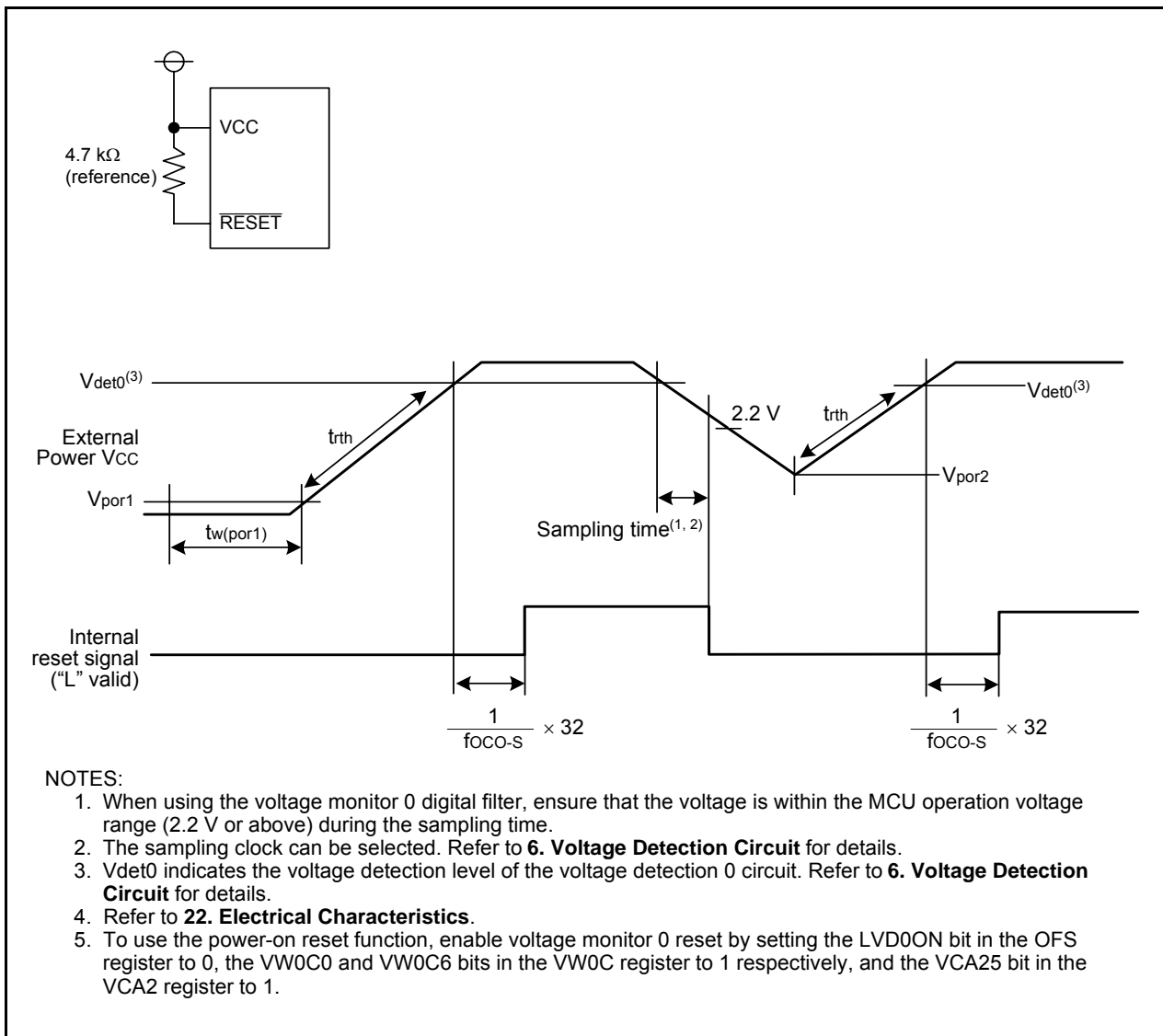
## 5.2 Power-On Reset Function

When the  $\overline{\text{RESET}}$  pin is connected to the VCC pin via a pull-up resistor, and the VCC pin voltage level rises while the rise gradient is  $t_{\text{rth}}$  or more, the power-on reset function is enabled and the MCU resets its pins, CPU, and SFR. When a capacitor is connected to the  $\overline{\text{RESET}}$  pin, too, always keep the voltage to the  $\overline{\text{RESET}}$  pin  $0.8V_{\text{CC}}$  or more. When the input voltage to the VCC pin reaches the  $V_{\text{det0}}$  level or above, the low-speed on-chip oscillator clock starts counting. When the low-speed on-chip oscillator clock count reaches 32, the internal reset signal is held “H” and the MCU enters the reset sequence (refer to **Figure 5.3**). The low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock after reset.

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

The voltage monitor 0 reset is enabled after power-on reset.

Figure 5.7 shows an Example of Power-On Reset Circuit and Operation.



**Figure 5.7** Example of Power-On Reset Circuit and Operation

### 5.3 Voltage Monitor 0 Reset

A reset is applied using the on-chip voltage detection 0 circuit. The voltage detection 0 circuit monitors the input voltage to the VCC pin. The voltage to monitor is Vdet0.

When the input voltage to the VCC pin reaches the Vdet0 level or below, the pins, CPU, and SFR are reset.

When the input voltage to the VCC pin reaches the Vdet0 level or above, the low-speed on-chip oscillator clock start counting. When the low-speed on-chip oscillator clock count reaches 32, the internal reset signal is held “H” and the MCU enters the reset sequence (refer to **Figure 5.3**). The low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock after reset.

The LVD0ON bit in the OFS register can be used to enable or disable voltage monitor 0 reset after a hardware reset. Setting the LVD0ON bit is only valid after a hardware reset.

To use the power-on reset function, enable voltage monitor 0 reset by setting the LVD0ON bit in the OFS register to 0, the VW0C0 and VW0C6 bits in the VW0C register to 1 respectively, and the VCA25 bit in the VCA2 register to 1.

The LVD0ON bit cannot be changed by a program. To set the LVD0ON bit, write 0 (voltage monitor 0 reset enabled after hardware reset) or 1 (voltage monitor 0 reset disabled after hardware reset) to bit 5 of address 0FFFFh using a flash programmer.

Refer to **Figure 5.4 OFS Register** for details of the OFS register.

Refer to **4. Special Function Registers (SFRs)** for the status of the SFR after voltage monitor 0 reset.

The internal RAM is not reset. When the input voltage to the VCC pin reaches the Vdet0 level or below while writing to the internal RAM is in progress, the contents of internal RAM are undefined.

Refer to **6. Voltage Detection Circuit** for details of voltage monitor 0 reset.

### 5.4 Voltage Monitor 1 Reset

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

When the input voltage to the VCC pin reaches the Vdet1 level or below, the pins, CPU, and SFR are reset and a program is executed beginning with the address indicated by the reset vector. After reset, the low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock.

The voltage monitor 1 does not reset some portions of the SFR. Refer to **4. Special Function Registers (SFRs)** for details.

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

Refer to **6. Voltage Detection Circuit** for details of voltage monitor 1 reset.

### 5.5 Voltage Monitor 2 Reset

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

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

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

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

Refer to **6. Voltage Detection Circuit** for details of voltage monitor 2 reset.

## 5.6 Watchdog Timer Reset

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

The watchdog timer reset does not reset some SFRs. Refer to **4. Special Function Registers (SFRs)** for details. The internal RAM is not reset. When the watchdog timer underflows, the contents of internal RAM are undefined. Refer to **16. Watchdog Timer** for details of the watchdog timer.

## 5.7 Software Reset

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

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

## 6. Voltage Detection Circuit

The voltage detection circuit monitors the input voltage to the VCC pin. This circuit can be used to monitor the VCC input voltage by a program. Alternately, voltage monitor 0 reset, voltage monitor 1 interrupt, voltage monitor 1 reset, voltage monitor 2 interrupt, and voltage monitor 2 reset can also be used.

Note that voltage monitor 1 and voltage monitor 2 share the voltage detection circuit with comparator 1 and comparator 2. Either voltage monitor 1 and voltage monitor 2 or comparator 1 and comparator 2 can be selected.

Table 6.1 lists the Specifications of Voltage Detection Circuit and Figures 6.1 to 6.4 show the Block Diagrams. Figures 6.5 to 6.10 show the Associated Registers.

**Table 6.1 Specifications of Voltage Detection Circuit**

	Item	Voltage Detection 0	Voltage Detection 1	Voltage Detection 2
VCC Monitor	Voltage to monitor	Vdet0	Vdet1	Vdet2
	Detection target	Whether passing through Vdet0 by falling	Passing through Vdet1 by rising or falling	Passing through Vdet2 by rising or falling
	Monitor	None	VW1C3 bit in VW1C register Whether VCC is higher or lower than Vdet1	VCA13 bit in VCA1 register Whether VCC is higher or lower than Vdet2
Process When Voltage is Detected	Reset	Voltage monitor 0 reset Reset at Vdet0 > VCC; restart CPU operation at VCC > Vdet0	Voltage monitor 1 reset Reset at Vdet1 > VCC; restart CPU operation after a specified time	Voltage monitor 2 reset Reset at Vdet2 > VCC; restart CPU operation after a specified time
	Interrupt	None	Voltage monitor 1 interrupt Interrupt request at both or either of Vdet1 > VCC and VCC > Vdet1	Voltage monitor 2 interrupt Interrupt request at both or either of Vdet2 > VCC and VCC > Vdet2
Digital Filter	Switch enabled/disabled	Available	Available	Available
	Sampling time	(Divide-by-n of fOCO-S) × 4 n: 1, 2, 4, and 8	(Divide-by-n of fOCO-S) × 2 n: 1, 2, 4, and 8	(Divide-by-n of fOCO-S) × 2 n: 1, 2, 4, and 8

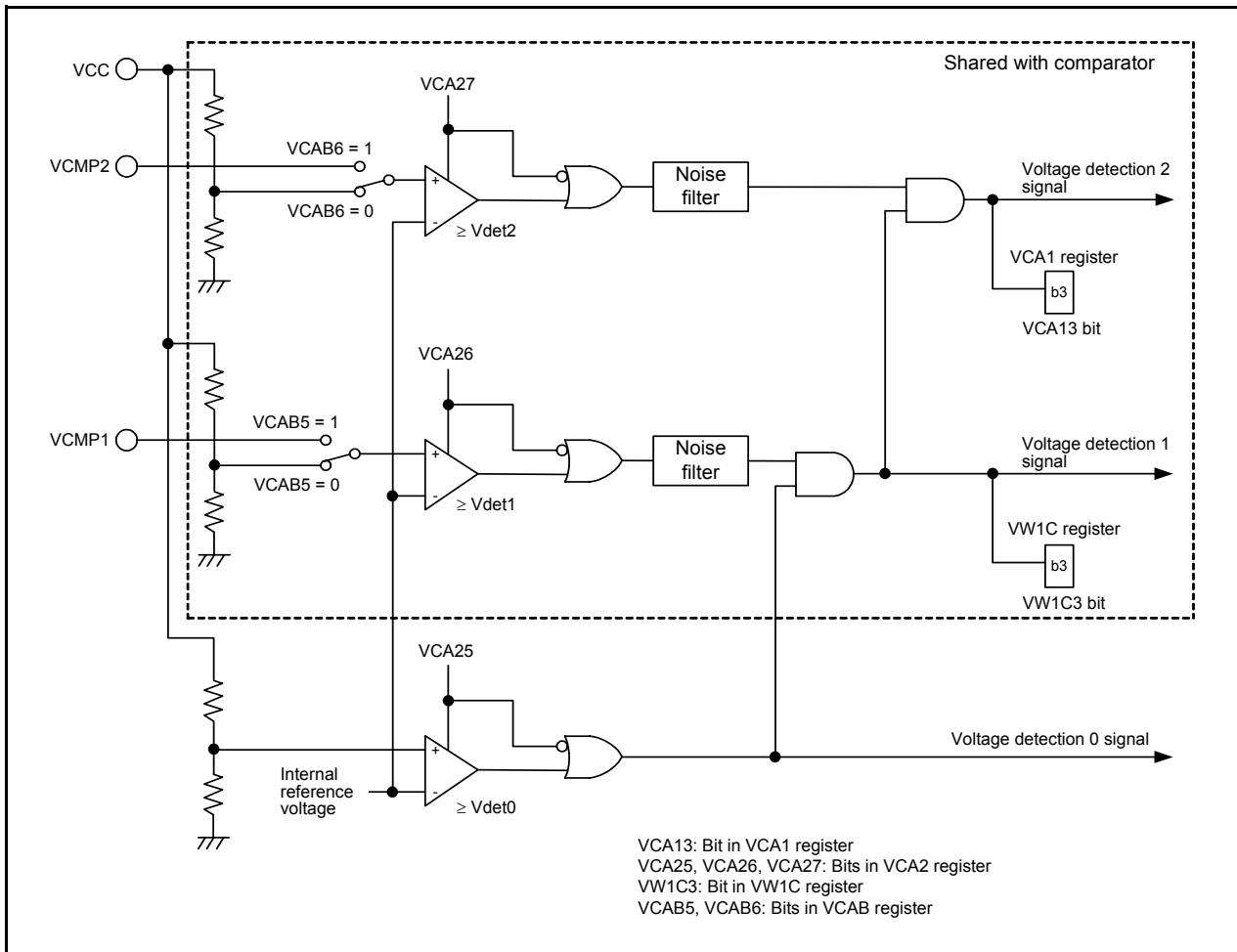


Figure 6.1 Block Diagram of Voltage Detection Circuit

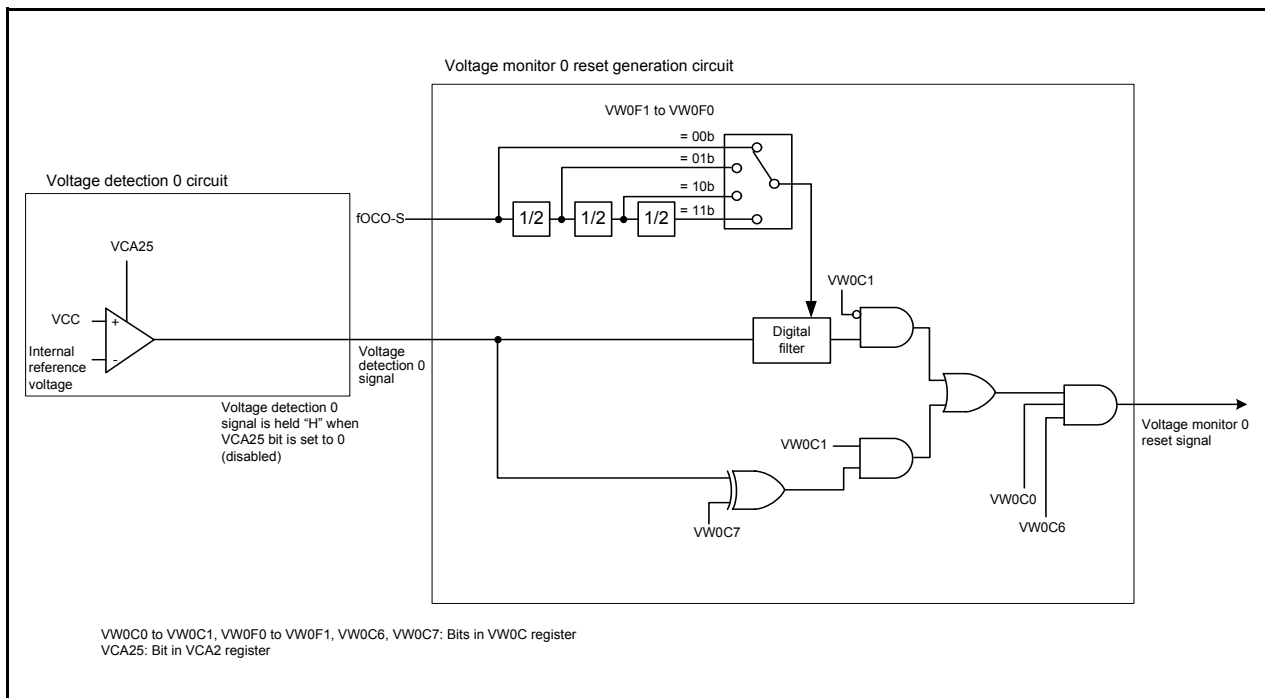


Figure 6.2 Block Diagram of Voltage Monitor 0 Reset Generation Circuit

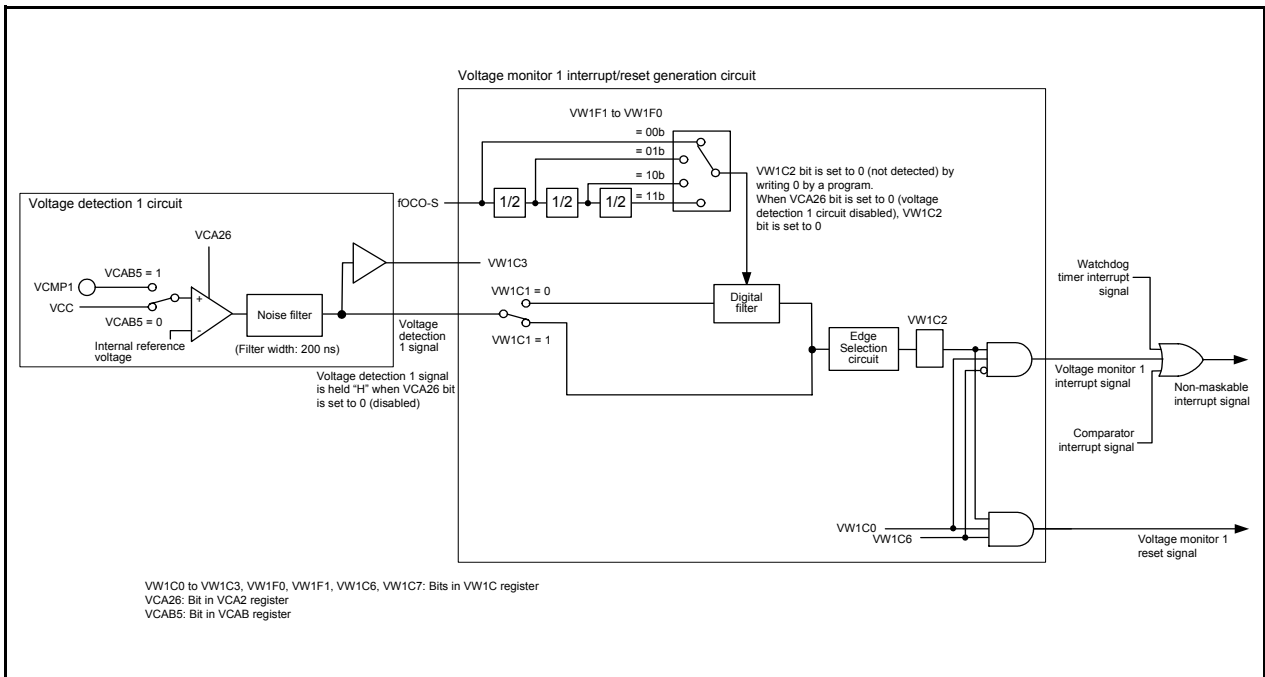


Figure 6.3 Block Diagram of Voltage Monitor 1 Interrupt/Reset Generation Circuit

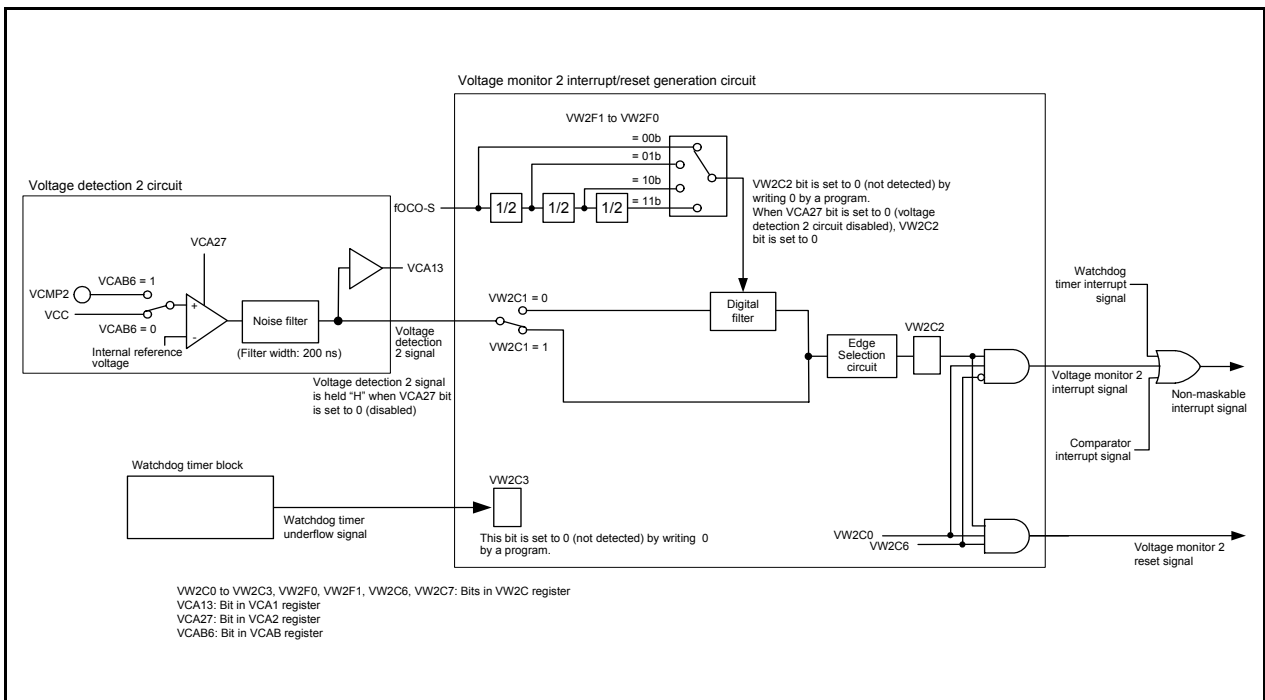


Figure 6.4 Block Diagram of Voltage Monitor 2 Interrupt/Reset Generation Circuit

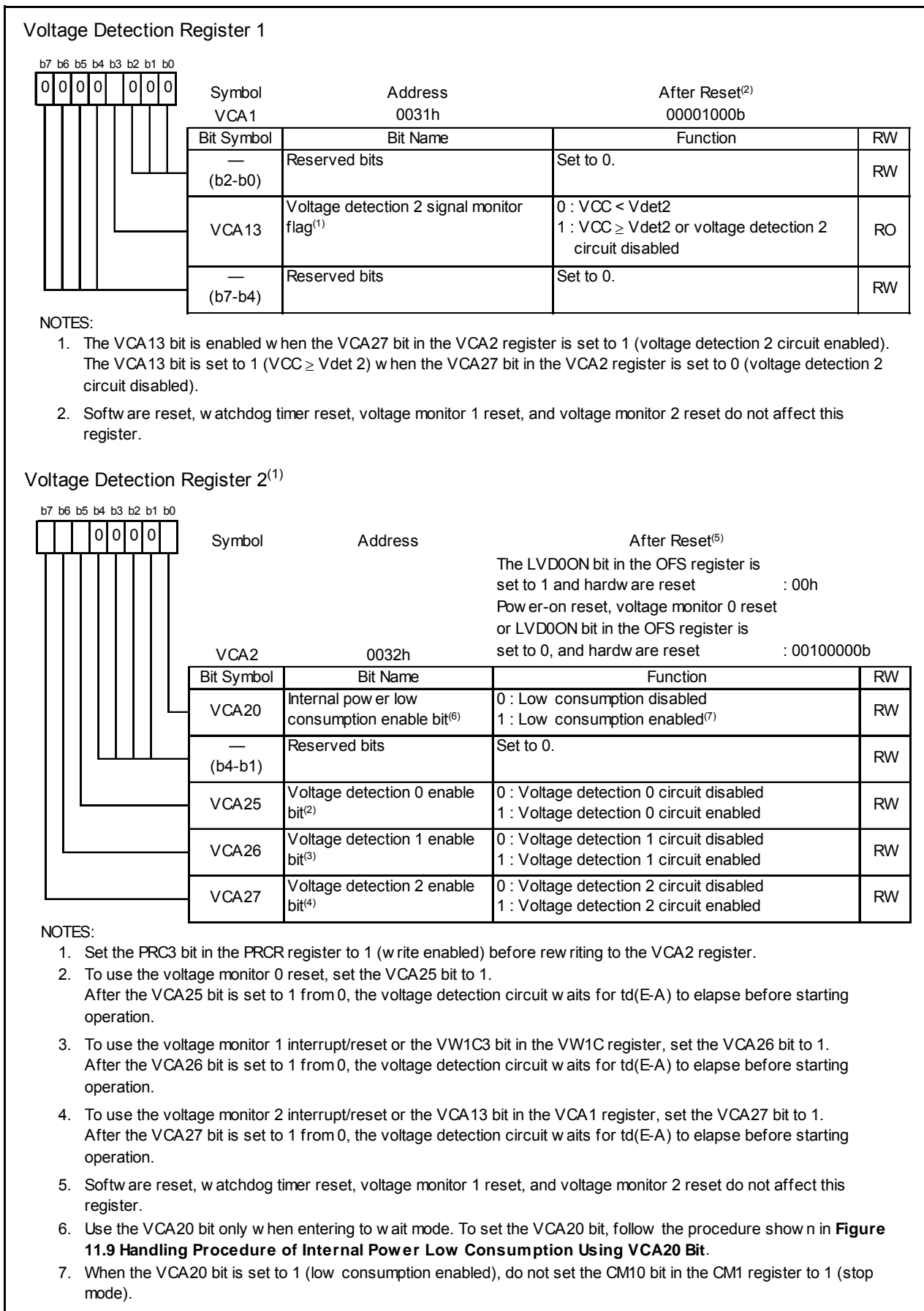


Figure 6.5 Registers VCA1 and VCA2

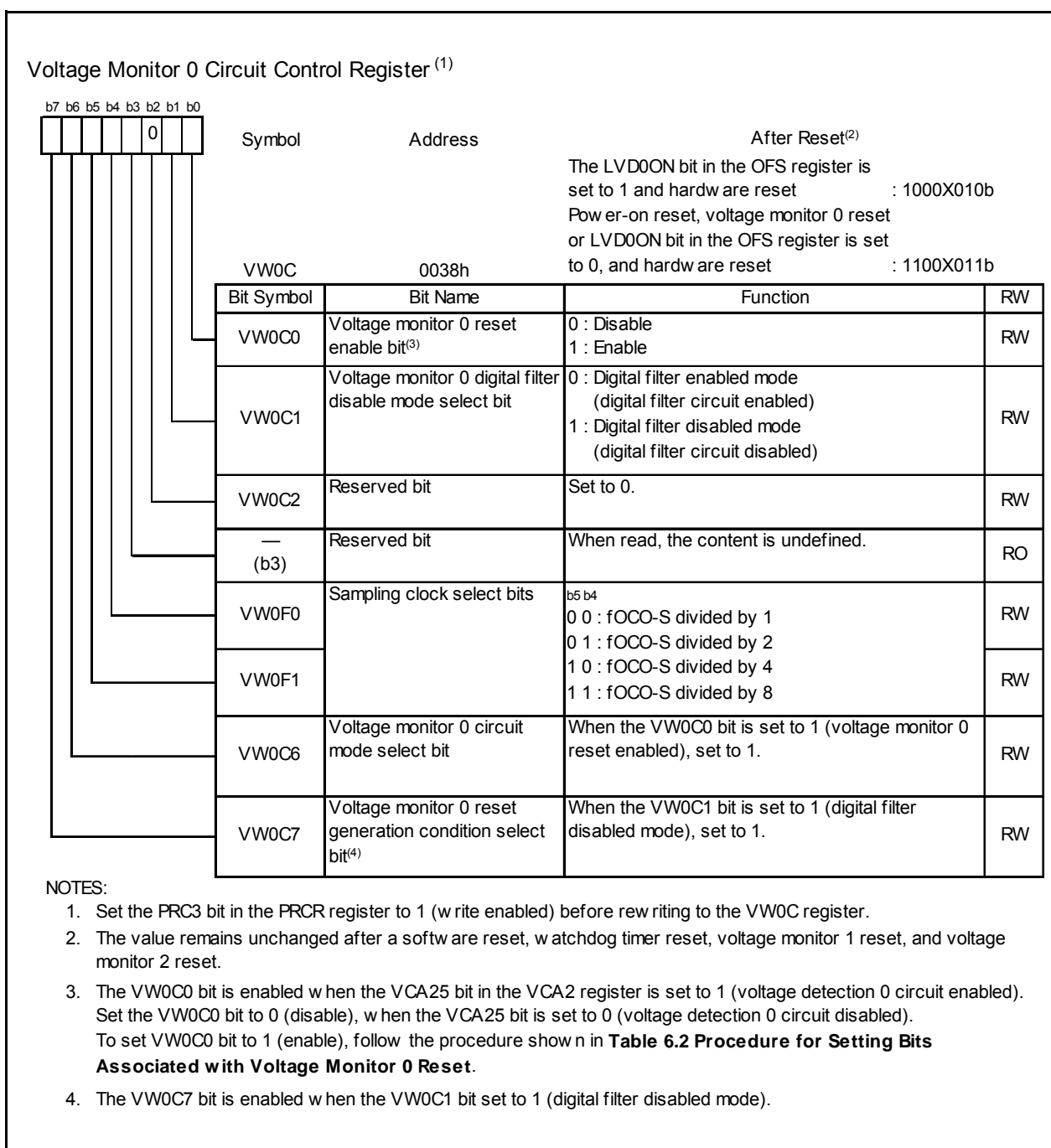


Figure 6.6 VW0C Register



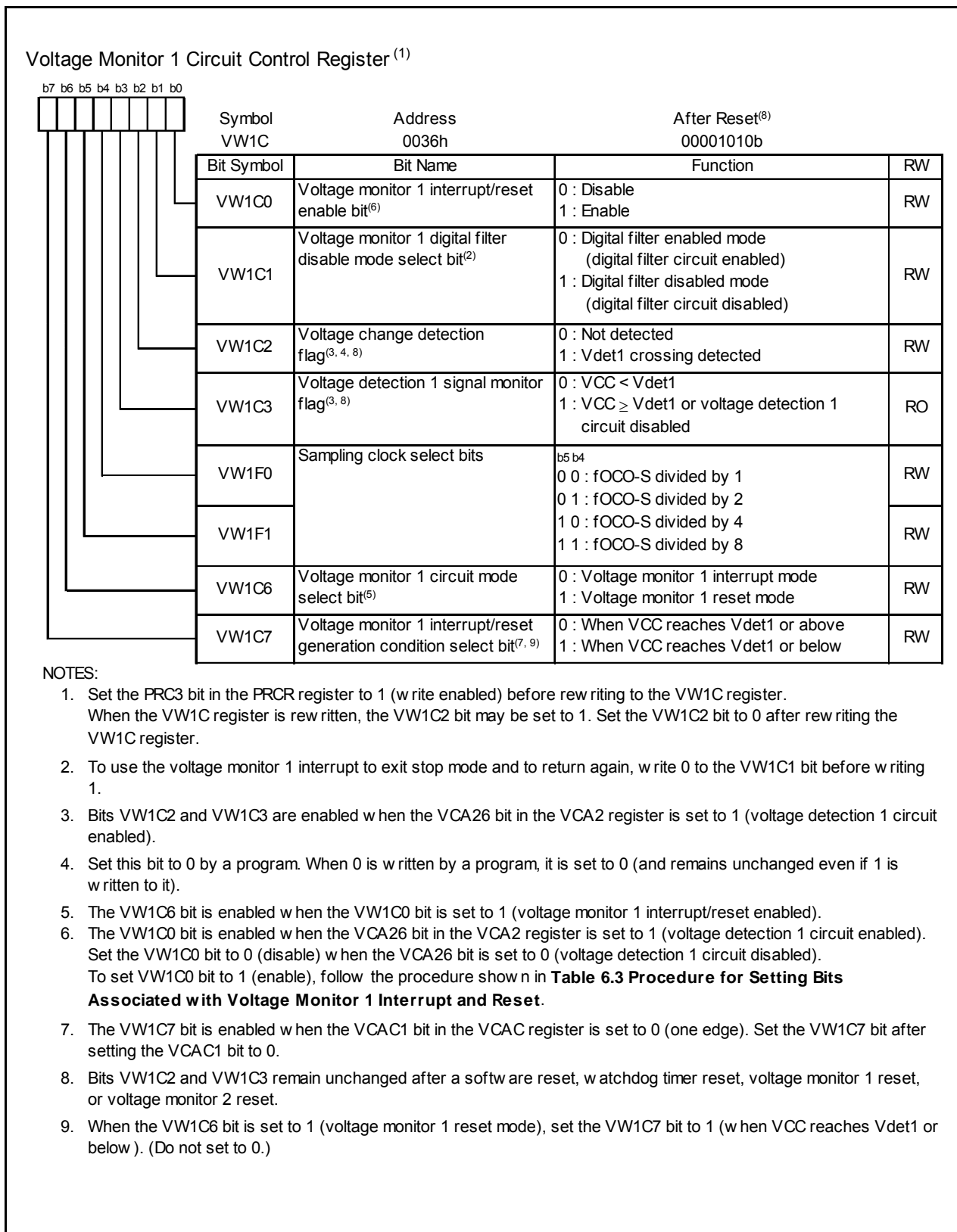


Figure 6.7 VW1C Register

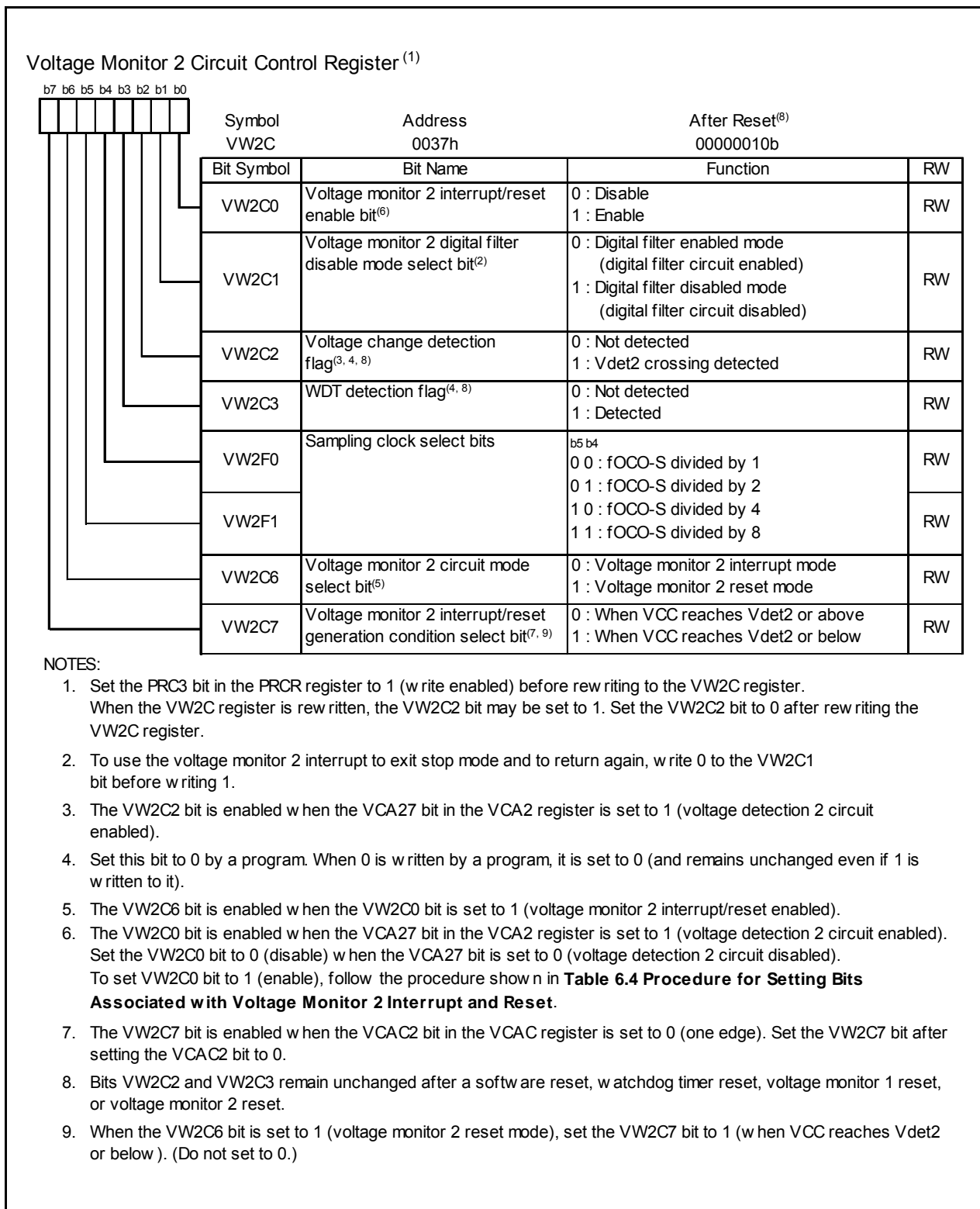


Figure 6.8 VW2C Register

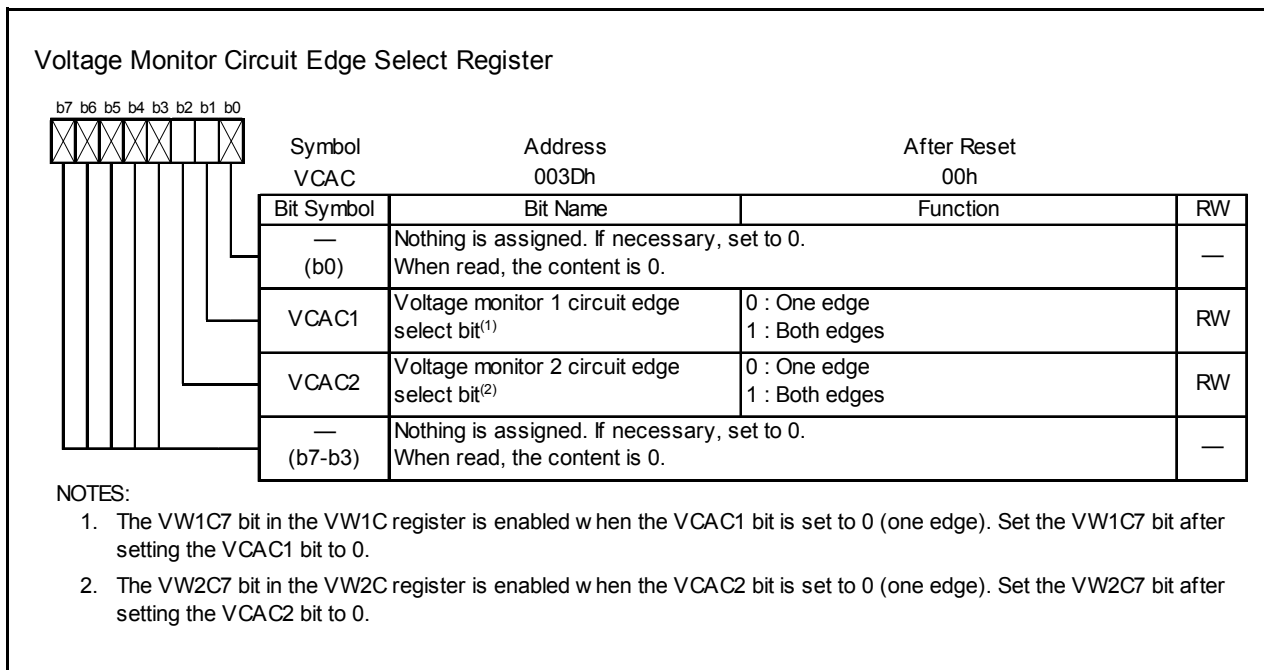


Figure 6.9 VCAC Register

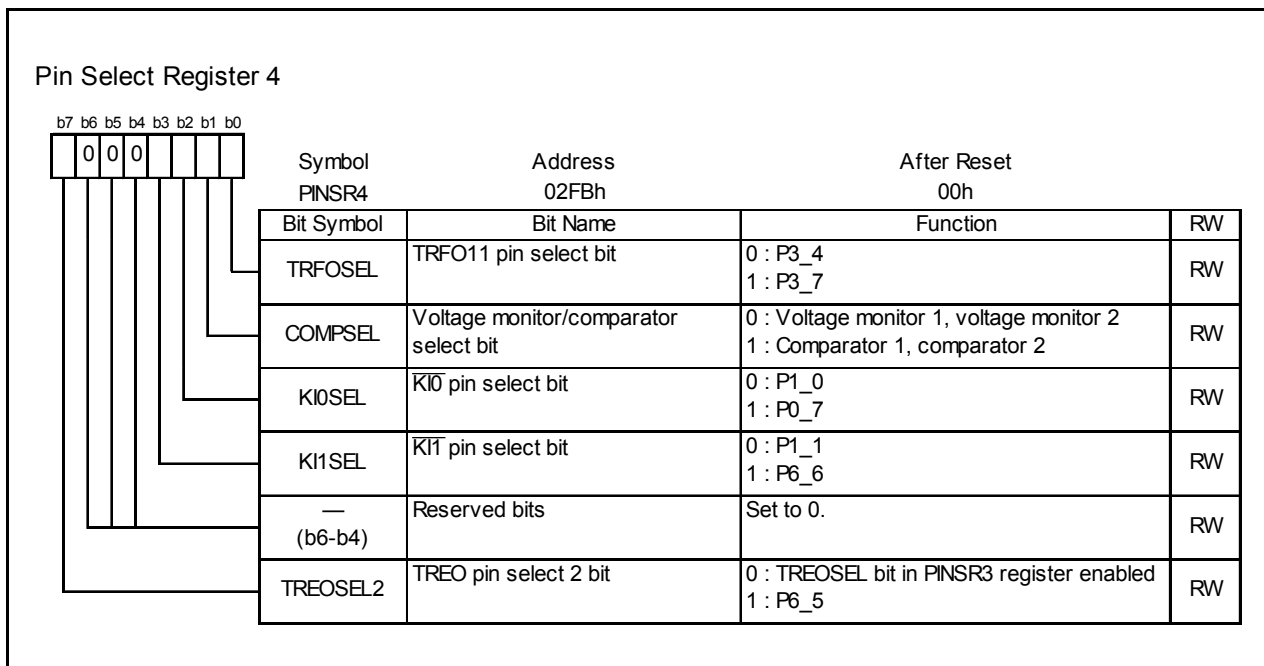


Figure 6.10 PINSR4 Register

## 6.1 VCC Input Voltage

### 6.1.1 Monitoring Vdet0

Vdet0 cannot be monitored.

### 6.1.2 Monitoring Vdet1

Set the VCA26 bit in the VCA2 register to 1 (voltage detection 1 circuit enabled). After td(E-A) has elapsed (refer to **22. Electrical Characteristics**), Vdet1 can be monitored by the VW1C3 bit in the VW1C register.

### 6.1.3 Monitoring Vdet2

Set the VCA27 bit in the VCA2 register to 1 (voltage detection 2 circuit enabled). After td(E-A) has elapsed (refer to **22. Electrical Characteristics**), Vdet2 can be monitored by the VCA13 bit in the VCA1 register.

## 6.2 Voltage Monitor 0 Reset

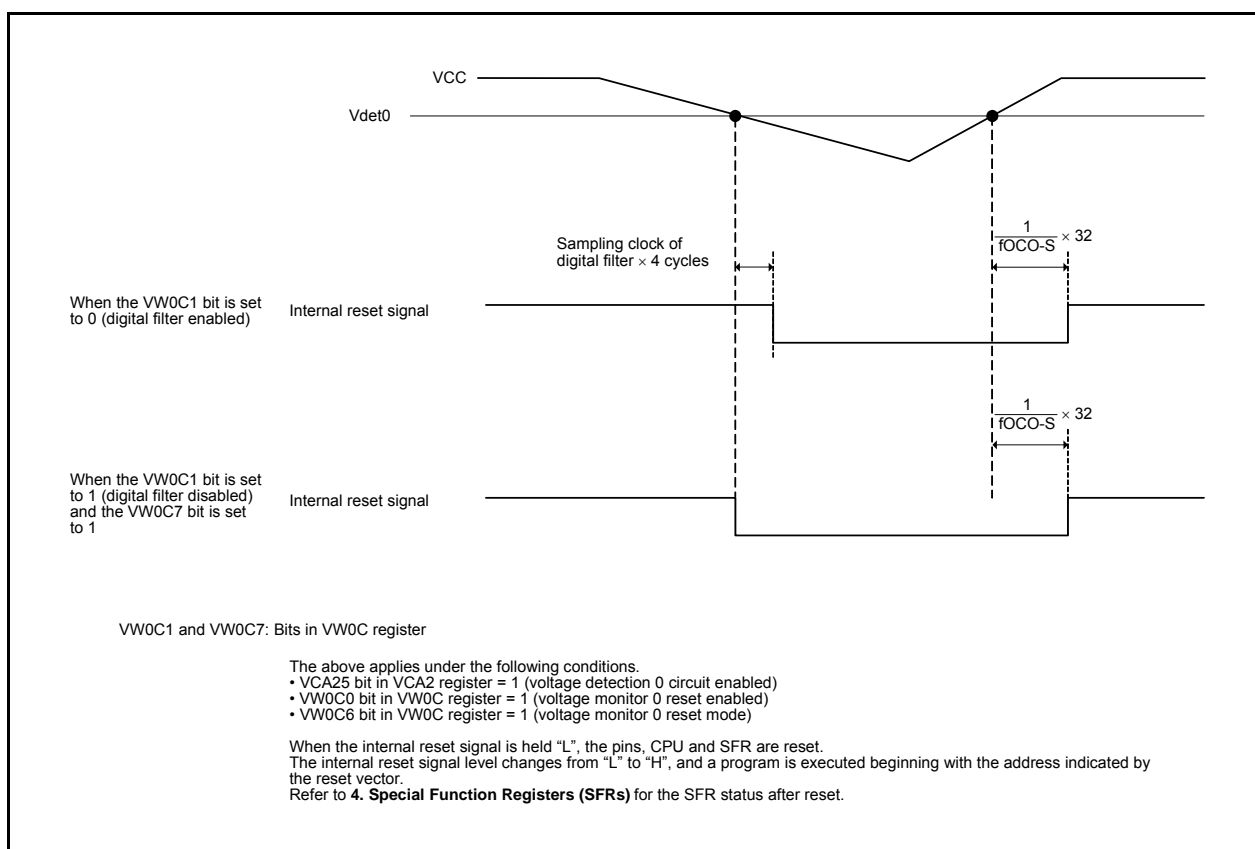
Table 6.2 lists the Procedure for Setting Bits Associated with Voltage Monitor 0 Reset and Figure 6.11 shows an Example of Voltage Monitor 0 Reset Operation. To use the voltage monitor 0 reset to exit stop mode, set the VW0C1 bit in the VW0C register to 1 (digital filter disabled).

**Table 6.2 Procedure for Setting Bits Associated with Voltage Monitor 0 Reset**

Step	When Using Digital Filter	When Not Using Digital Filter
1	Set the VCA25 bit in the VCA2 register to 1 (voltage detection 0 circuit enabled)	
2	Wait for $t_d(E-A)$	
3	Select the sampling clock of the digital filter by the VW0F0 to VW0F1 bits in the VW0C register	Set the VW0C7 bit in the VW0C register to 1
4(1)	Set the VW0C1 bit in the VW0C register to 0 (digital filter enabled)	Set the VW0C1 bit in the VW0C register to 1 (digital filter disabled)
5(1)	Set the VW0C6 bit in the VW0C register to 1 (voltage monitor 0 reset mode)	
6	Set the VW0C2 bit in the VW0C register to 0	
7	Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on)	–
8	Wait for 4 cycles of the sampling clock of the digital filter	– (No wait time required)
9	Set the VW0C0 bit in the VW0C register to 1 (voltage monitor 0 reset enabled)	

**NOTE:**

- When the VW0C0 bit is set to 0, steps 3, 4, and 5 can be executed simultaneously (with 1 instruction).



**Figure 6.11 Example of Voltage Monitor 0 Reset Operation**

### 6.3 Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset

Table 6.3 lists the Procedure for Setting Bits Associated with Voltage Monitor 1 Interrupt and Reset. Figure 6.12 shows an Example of Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset Operation. To use the voltage monitor 1 interrupt or voltage monitor 1 reset to exit stop mode, set the VW1C1 bit in the VW1C register to 1 (digital filter disabled).

**Table 6.3 Procedure for Setting Bits Associated with Voltage Monitor 1 Interrupt and Reset**

Step	When Using Digital Filter		When Not Using Digital Filter	
	Voltage Monitor 1 Interrupt	Voltage Monitor 1 Reset	Voltage Monitor 1 Interrupt	Voltage Monitor 1 Reset
1	Set the COMPSEL bit in the PINSR4 register to 0 (voltage monitor 1, voltage monitor 2)			
2	Set the VCA26 bit in the VCA2 register to 1 (voltage detection 1 circuit enabled)			
3	Wait for $t_d(E-A)$			
4	Select the sampling clock of the digital filter by the VW1F0 to VW1F1 bits in the VW1C register		Set the VW1C1 bit in the VW1C register to 1 (digital filter disabled)	
5 <sup>(2)</sup>	Set the VW1C1 bit in the VW1C register to 0 (digital filter enabled)		-	
6	Select the timing of the interrupt and reset request by the VCAC1 bit in the VCAC register and the VW1C7 bit in the VW1C register <sup>(1)</sup>		Select the timing of the interrupt and reset request by the VCAC1 bit in the VCAC register and the VW1C7 bit in the VW1C register <sup>(1)</sup>	
7	Set the VW1C6 bit in the VW1C register to 0 (voltage monitor 1 interrupt mode)	Set the VW1C6 bit in the VW1C register to 1 (voltage monitor 1 reset mode)	Set the VW1C6 bit in the VW1C register to 0 (voltage monitor 1 interrupt mode)	Set the VW1C6 bit in the VW1C register to 1 (voltage monitor 1 reset mode)
8	Set the VW1C2 bit in the VW1C register to 0 (Vdet1 crossing is not detected)			
9	Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on)		-	
10	Wait for 2 cycles of the sampling clock of the digital filter		- (No wait time required)	
11	Set the VW1C0 bit in the VW1C register to 1 (voltage monitor 1 interrupt/reset enabled)			

**NOTES:**

1. Set the VW1C7 bit to 1 (when VCC reaches Vdet1 or below) for the voltage monitor 1 reset.
2. When the VW1C0 bit is set to 0, steps 4 and 5 can be executed simultaneously (with 1 instruction).

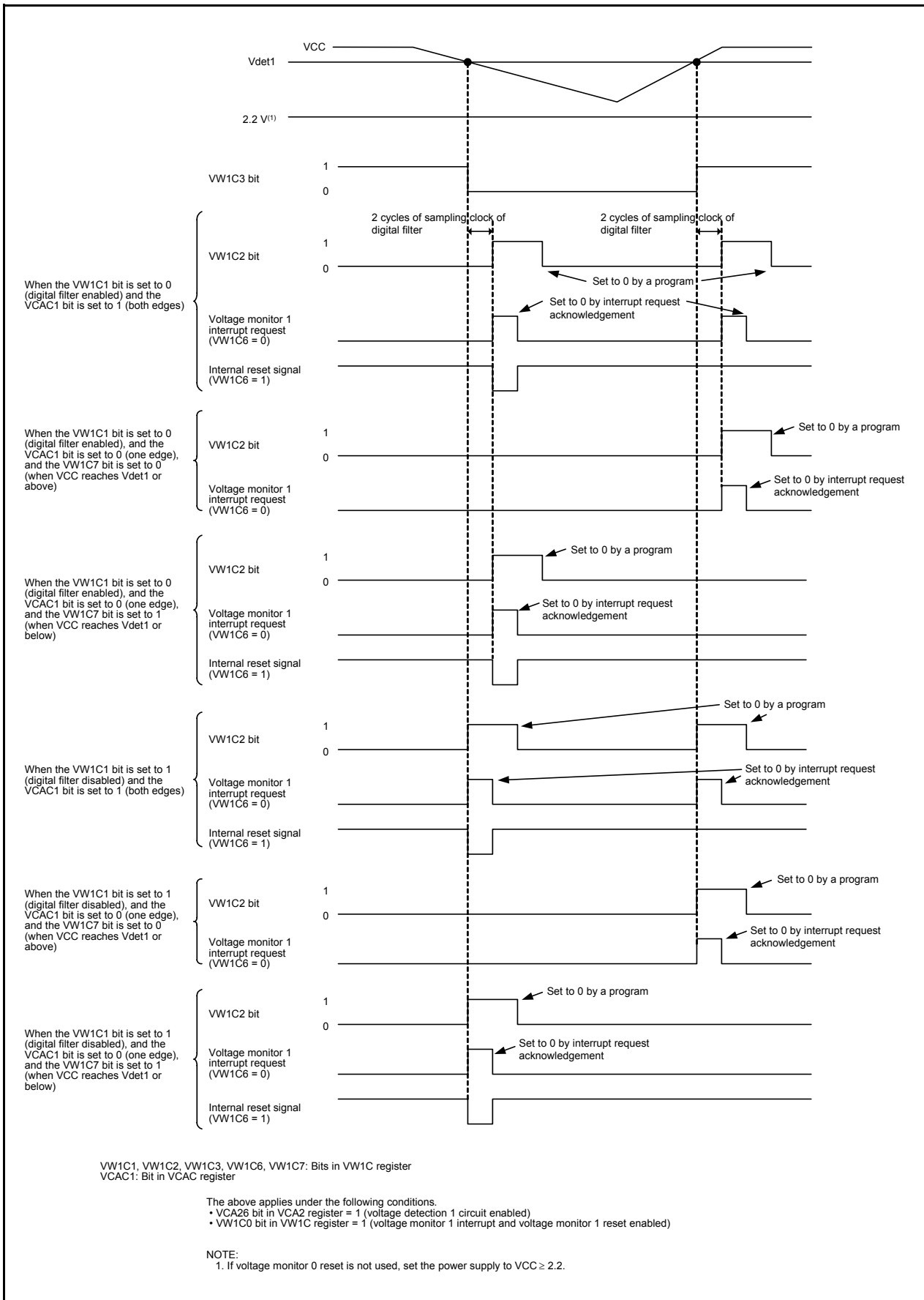


Figure 6.12 Example of Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset Operation

## 6.4 Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset

Table 6.4 lists the Procedure for Setting Bits Associated with Voltage Monitor 2 Interrupt and Reset. Figure 6.13 shows an Example of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset Operation. To use the voltage monitor 2 interrupt or voltage monitor 2 reset to exit stop mode, set the VW2C1 bit in the VW2C register to 1 (digital filter disabled).

**Table 6.4 Procedure for Setting Bits Associated with Voltage Monitor 2 Interrupt and Reset**

Step	When Using Digital Filter		When Not Using Digital Filter	
	Voltage Monitor 2 Interrupt	Voltage Monitor 2 Reset	Voltage Monitor 2 Interrupt	Voltage Monitor 2 Reset
1	Set the COMPSEL bit in the PINSR4 register to 0 (voltage monitor 1, voltage monitor 2)			
2	Set the VCA27 bit in the VCA2 register to 1 (voltage detection 2 circuit enabled)			
3	Wait for td(E-A)			
4	Select the sampling clock of the digital filter by the VW2F0 to VW2F1 bits in the VW2C register		Set the VW2C1 bit in the VW2C register to 1 (digital filter disabled)	
5 <sup>(2)</sup>	Set the VW2C1 bit in the VW2C register to 0 (digital filter enabled)		–	
6	Select the timing of the interrupt and reset request by the VCAC2 bit in the VCAC register and the VW2C7 bit in the VW2C register <sup>(1)</sup>		Select the timing of the interrupt and reset request by the VCAC2 bit in the VCAC register and the VW2C7 bit in the VW2C register <sup>(1)</sup>	
7	Set the VW2C6 bit in the VW2C register to 0 (voltage monitor 2 interrupt mode)	Set the VW2C6 bit in the VW2C register to 1 (voltage monitor 2 reset mode)	Set the VW2C6 bit in the VW2C register to 0 (voltage monitor 2 interrupt mode)	Set the VW2C6 bit in the VW2C register to 1 (voltage monitor 2 reset mode)
8	Set the VW2C2 bit in the VW2C register to 0 (Vdet2 crossing is not detected)			
9	Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on)		–	
10	Wait for 2 cycles of the sampling clock of the digital filter		– (No wait time required)	
11	Set the VW2C0 bit in the VW2C register to 1 (voltage monitor 2 interrupt/reset enabled)			

**NOTES:**

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



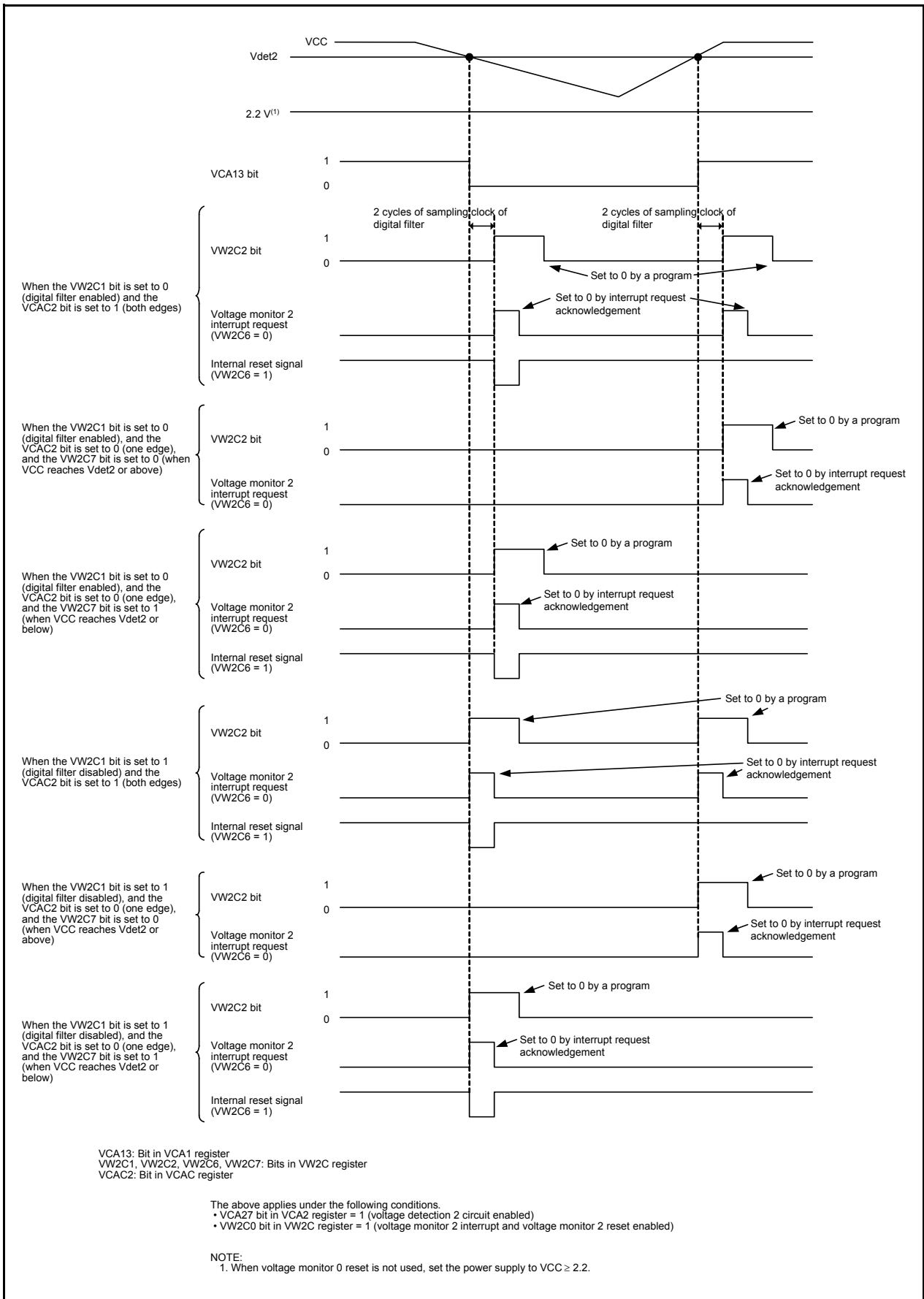


Figure 6.13 Example of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset Operation

## 7. Comparator

The comparators compare a reference input voltage and an analog input voltage. Comparator 1 and comparator 2 are independent of each other. Note that comparator 1 and comparator 2 share the voltage detection circuit with voltage monitor 1 and voltage monitor 2. Either comparator 1 and comparator 2 or voltage monitor 1 and voltage monitor 2 can be selected to use the voltage detection circuit.

### 7.1 Overview

The comparison result of the reference input voltage and analog input voltage can be read by software. The result also can be output from the VCOUT<sub>i</sub> (i = 1 or 2) pin. An internal reference voltage or input voltage to the CVREF pin can be selected as the reference input voltage. The comparator 1 interrupt and comparator 2 interrupt also can be used.

Table 7.1 lists the Specifications of Comparator, Figure 7.1 shows the Block Diagram of Comparator, and Table 7.2 lists the Pin Configuration of Comparator.

**Table 7.1 Specifications of Comparator**

Item		Comparator 1	Comparator 2
Analog input voltage		Input voltage to VCMP1 pin	Input voltage to VCMP2 pin
Reference input voltage		Internal reference voltage or input voltage to CVREF pin	
Comparison target		Whether passing through reference input voltage by rising or falling	
Comparison result monitor		VW1C3 bit in VW1C register	VCA13 bit in VCA1 register
		Whether higher or lower than reference input voltage	
Interrupt		Comparator 1 interrupt (non-makable or maskable selectable)	Comparator 2 interrupt (non-makable or maskable selectable)
		Interrupt request at both or either of reference input voltage > input voltage to VCMP1 pin and input voltage to VCMP1 pin > reference input voltage	Interrupt request at both or either of reference input voltage > input voltage to VCMP2 pin and input voltage to VCMP2 pin > reference input voltage
Digital Filter	Switch enabled/disabled	Available	
	Sampling time	(fOCO-S divided by n) × 2 n: 1, 2, 4, 8	
Comparison result output		Output from VCOUT1 pin (Whether the comparison result output is inverted or not can be selected)	Output from VCOUT2 pin (Whether the comparison result output is inverted or not can be selected)

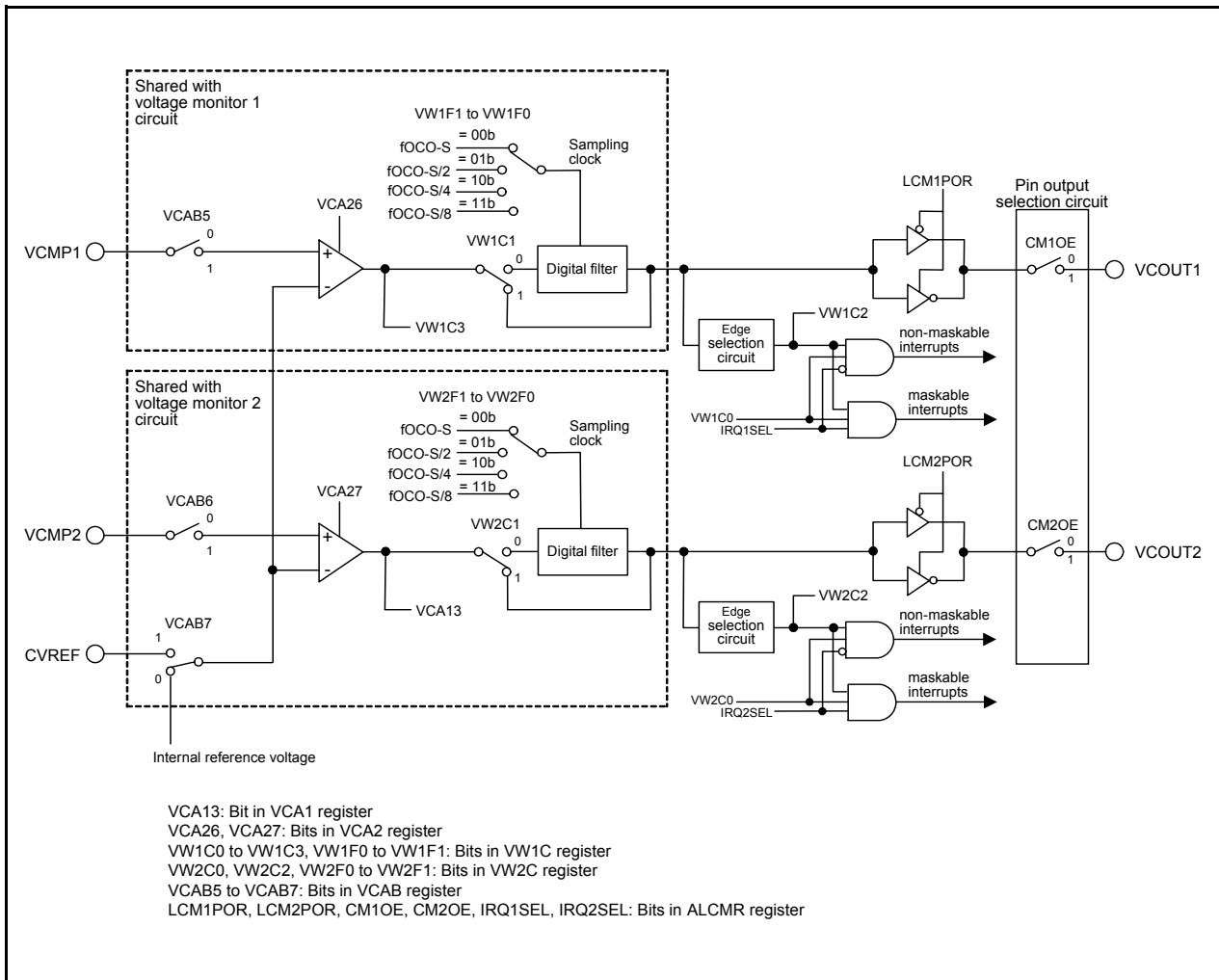


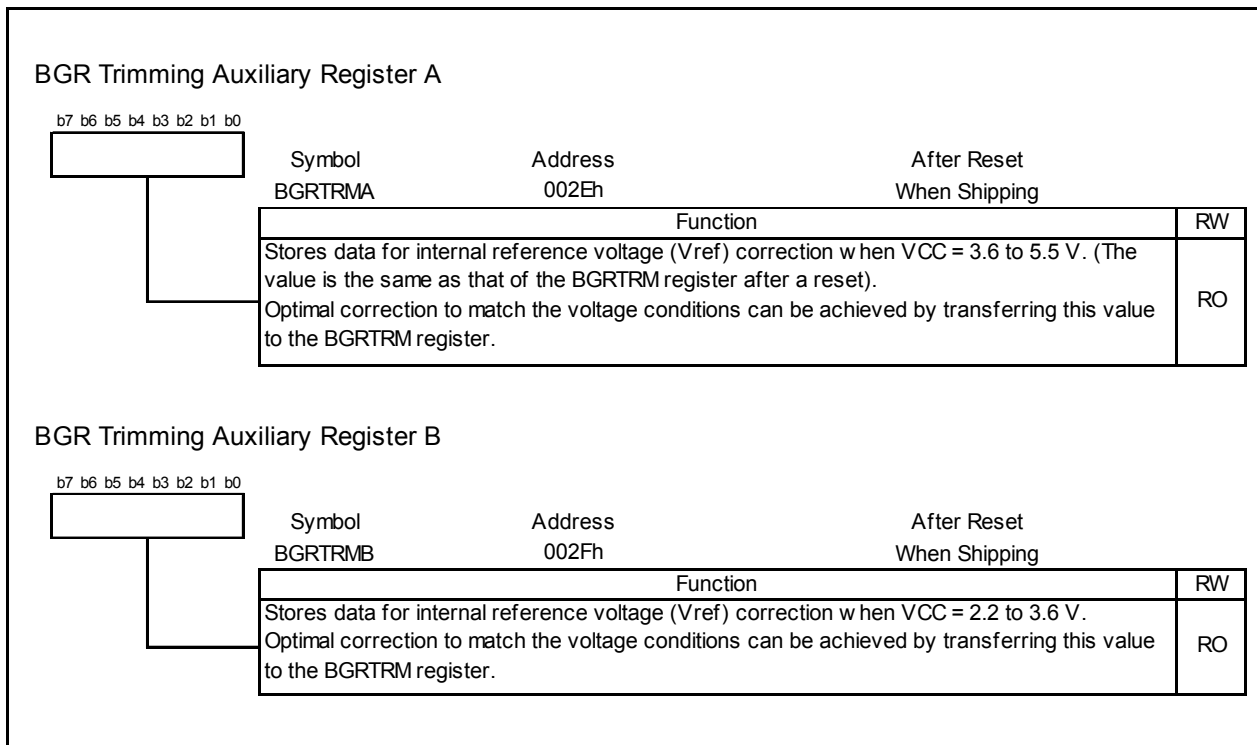
Figure 7.1 Block Diagram of Comparator

Table 7.2 Pin Configuration of Comparator

Pin Name	I/O	Function
VCMP1	Input	Comparator 1 analog pin
VCOU1	Output	Comparator 1 comparison result output pin
VCMP2	Input	Comparator 2 analog pin
VCOU2	Output	Comparator 2 comparison result output pin
CVREF	Input	Comparator reference voltage pin

## 7.2 Register Description

Figures 7.2 to 7.11 show the registers associated with the comparator when comparator 1 or comparator 2 is selected.



**Figure 7.2 Registers BGRTRMA and BGRTRMB**

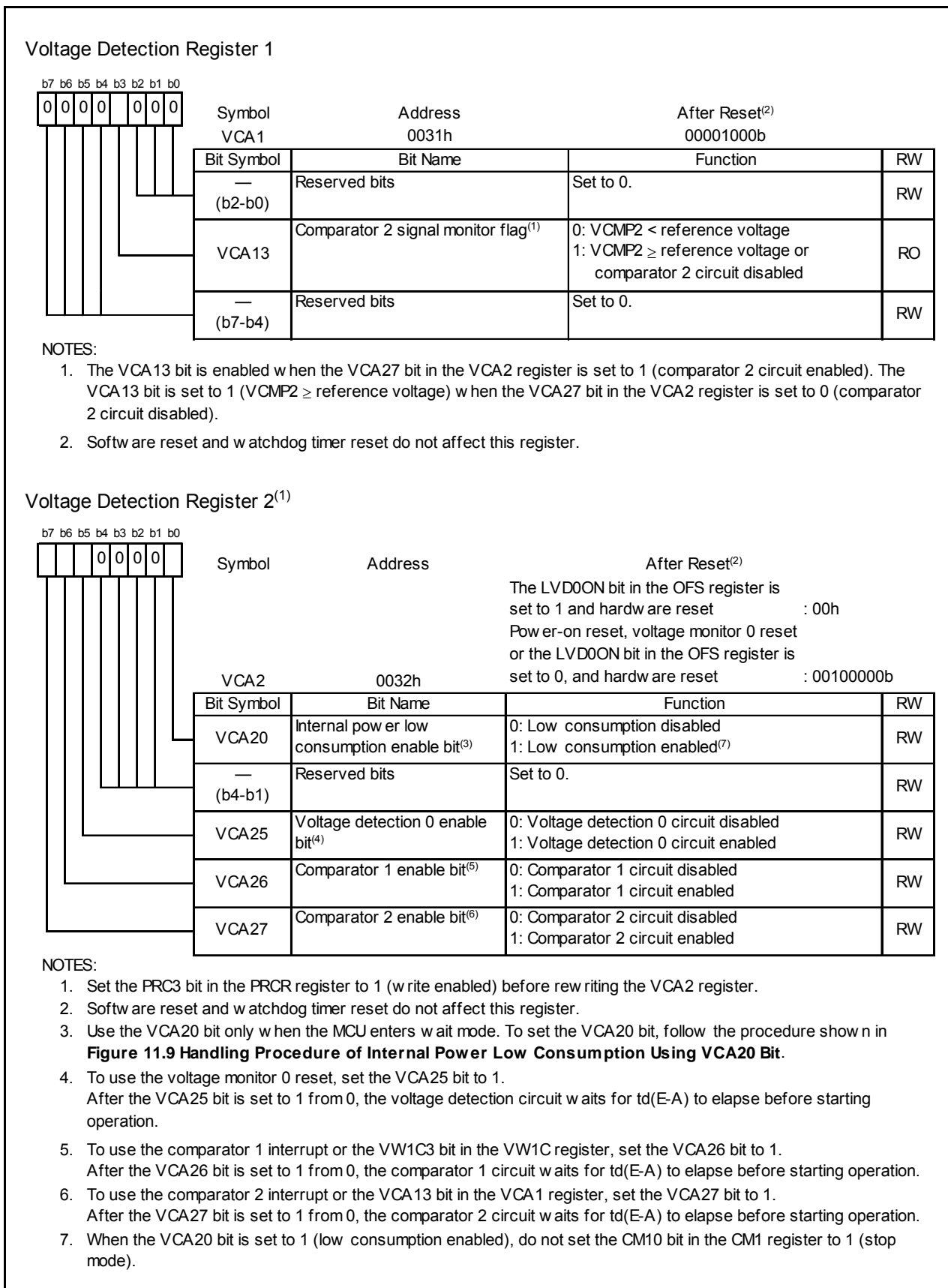


Figure 7.3 Registers VCA1 and VCA2

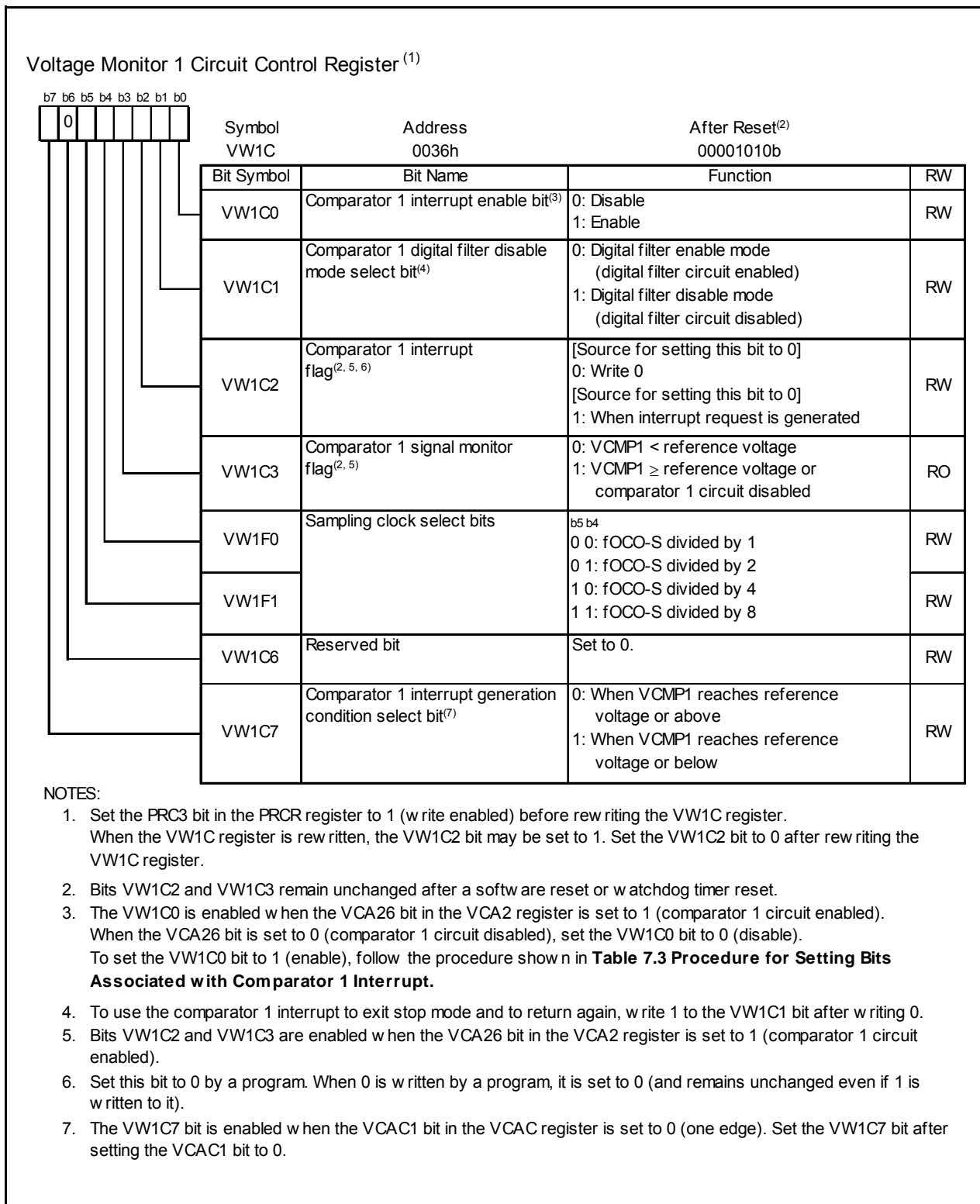


Figure 7.4 VW1C Register

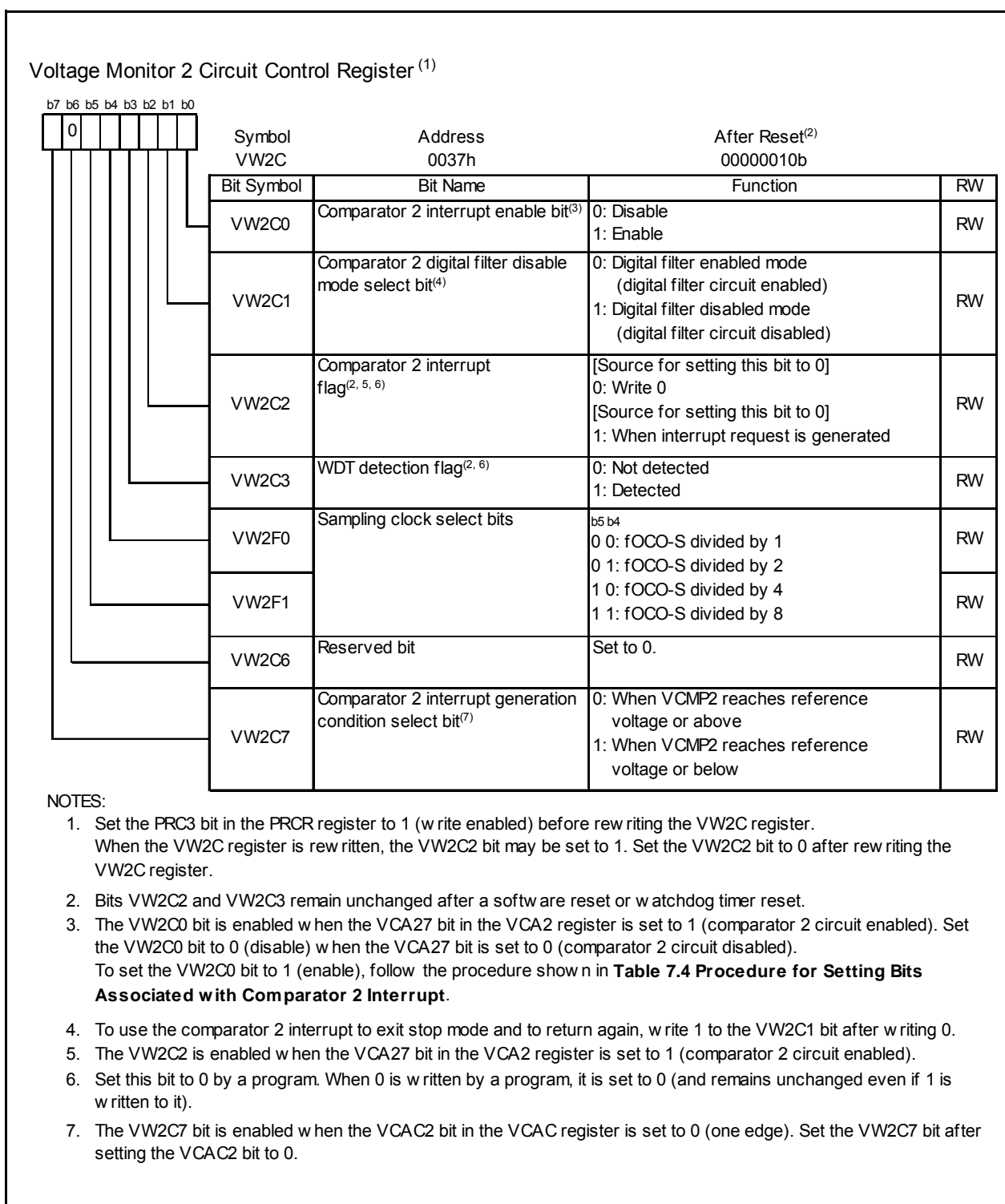


Figure 7.5 VW2C Register

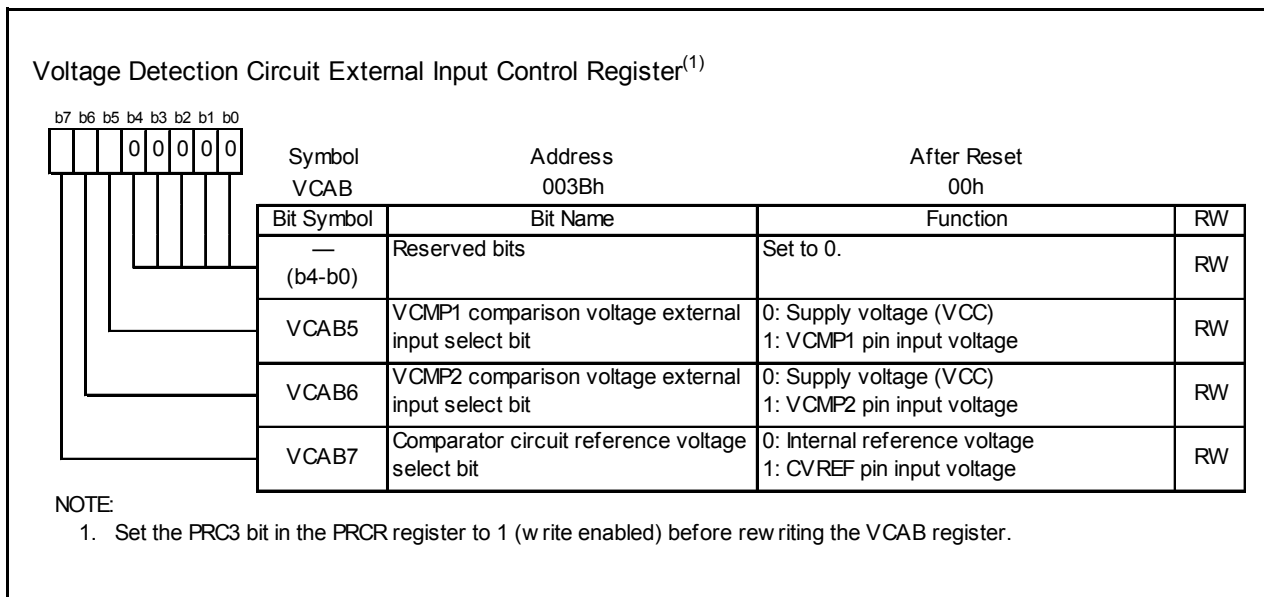


Figure 7.6 VCAB Register

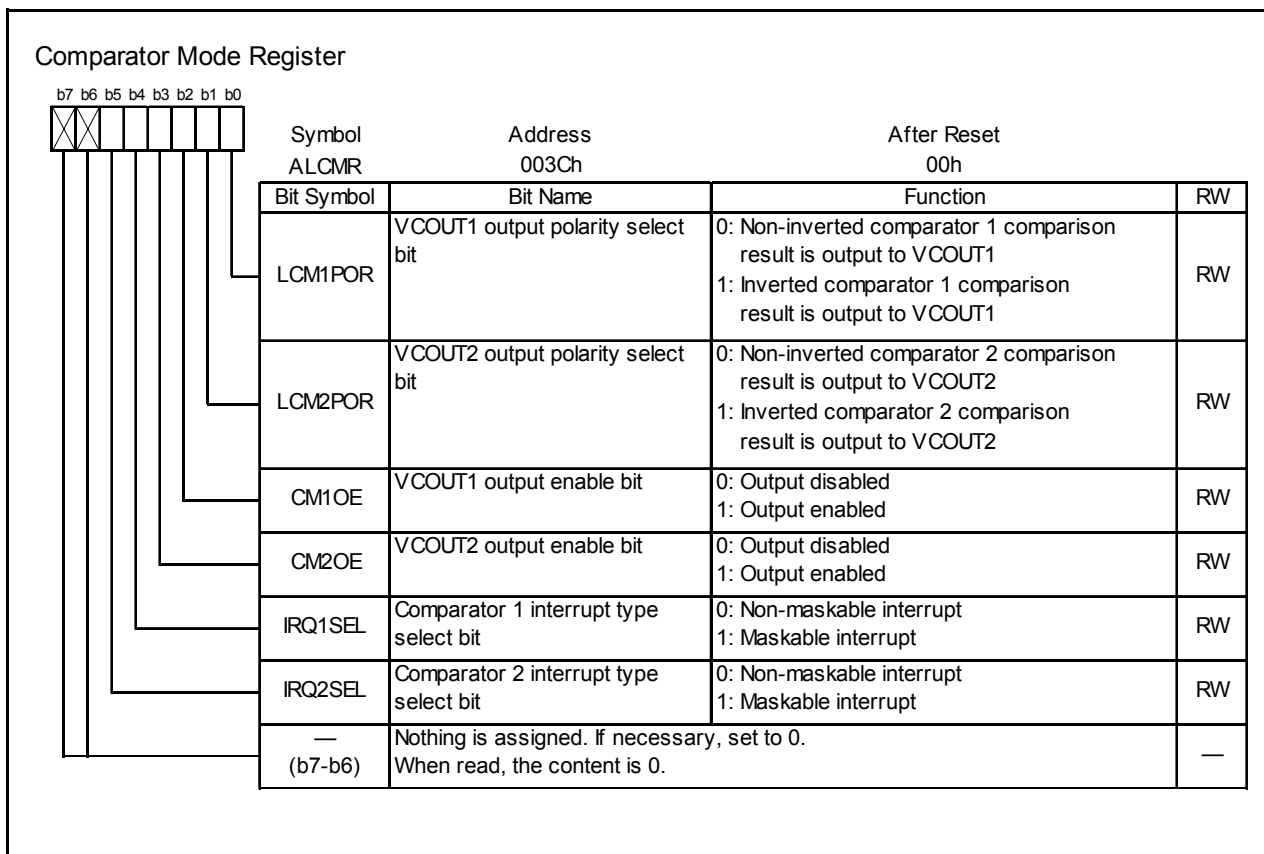


Figure 7.7 ALCMR Register



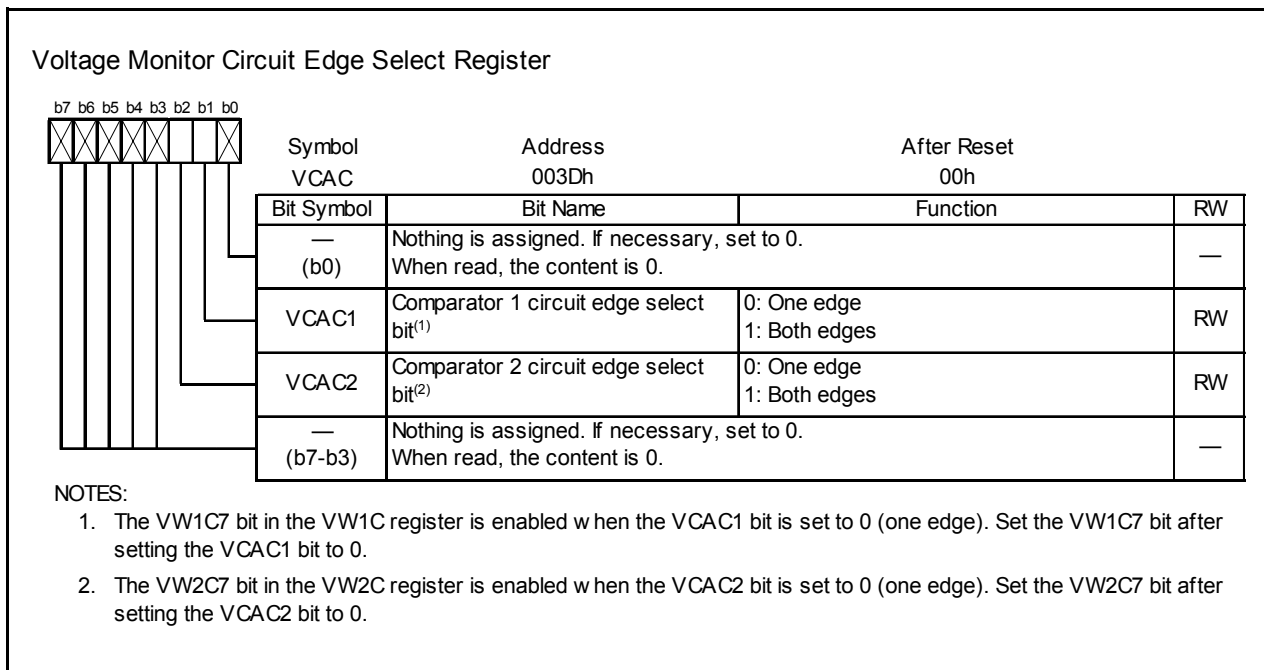


Figure 7.8 VCAC Register

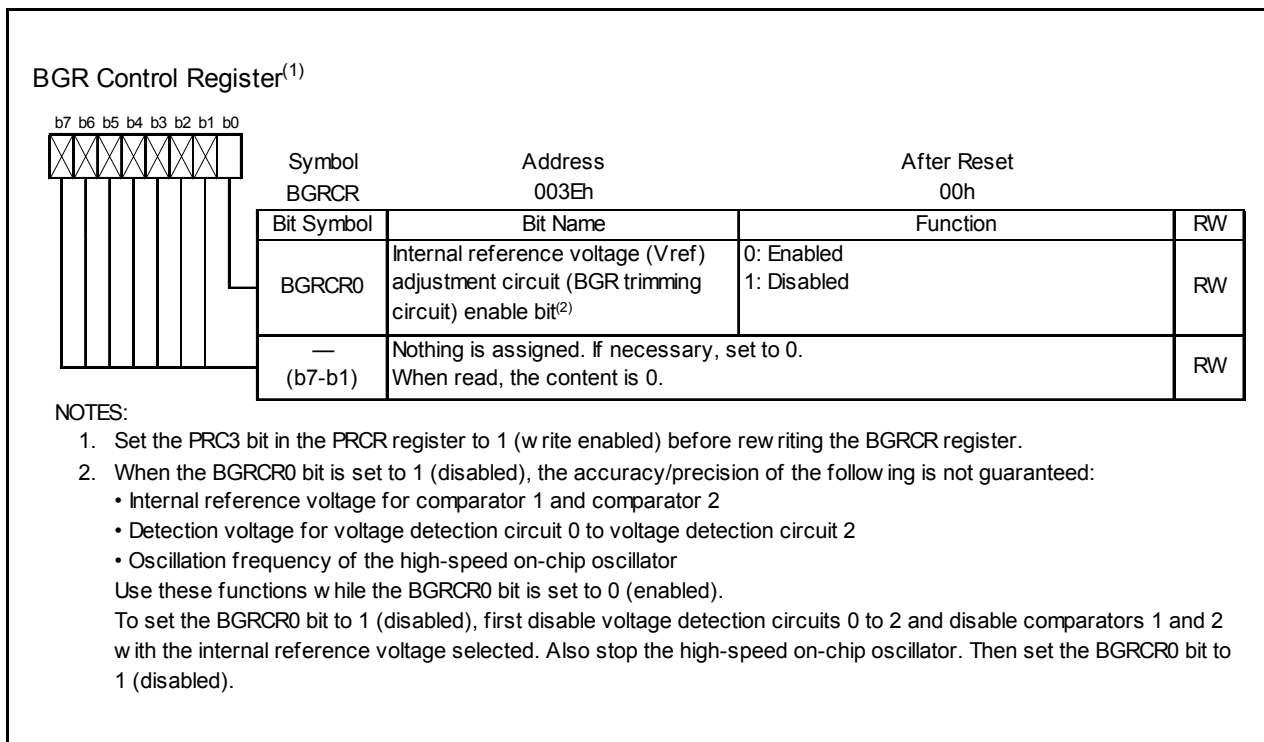


Figure 7.9 BGRCR Register

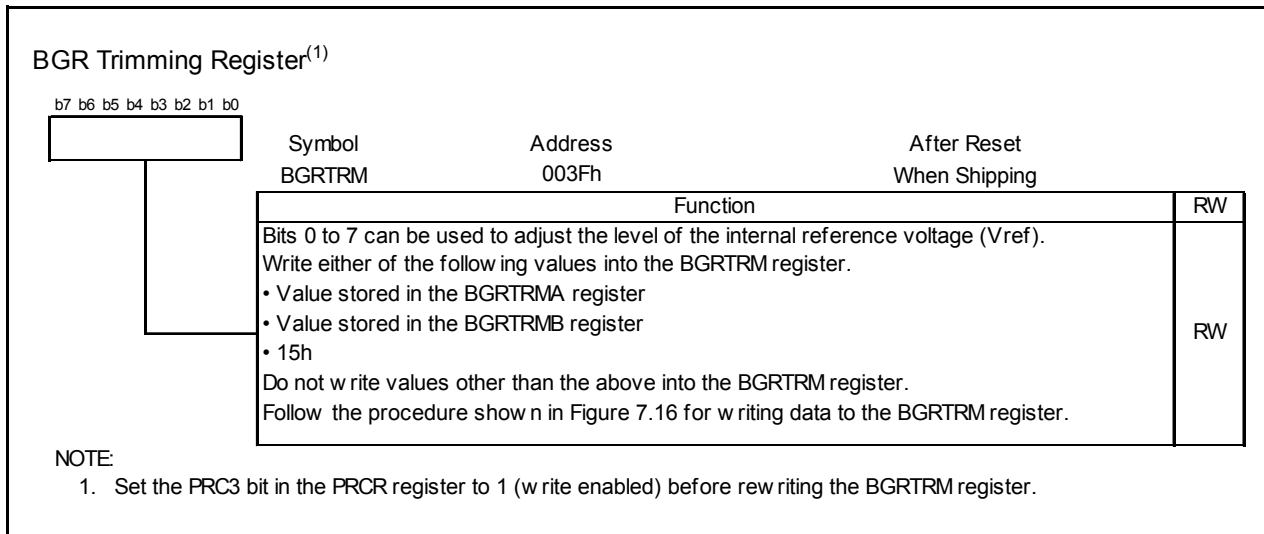


Figure 7.10 BGRTRM Register

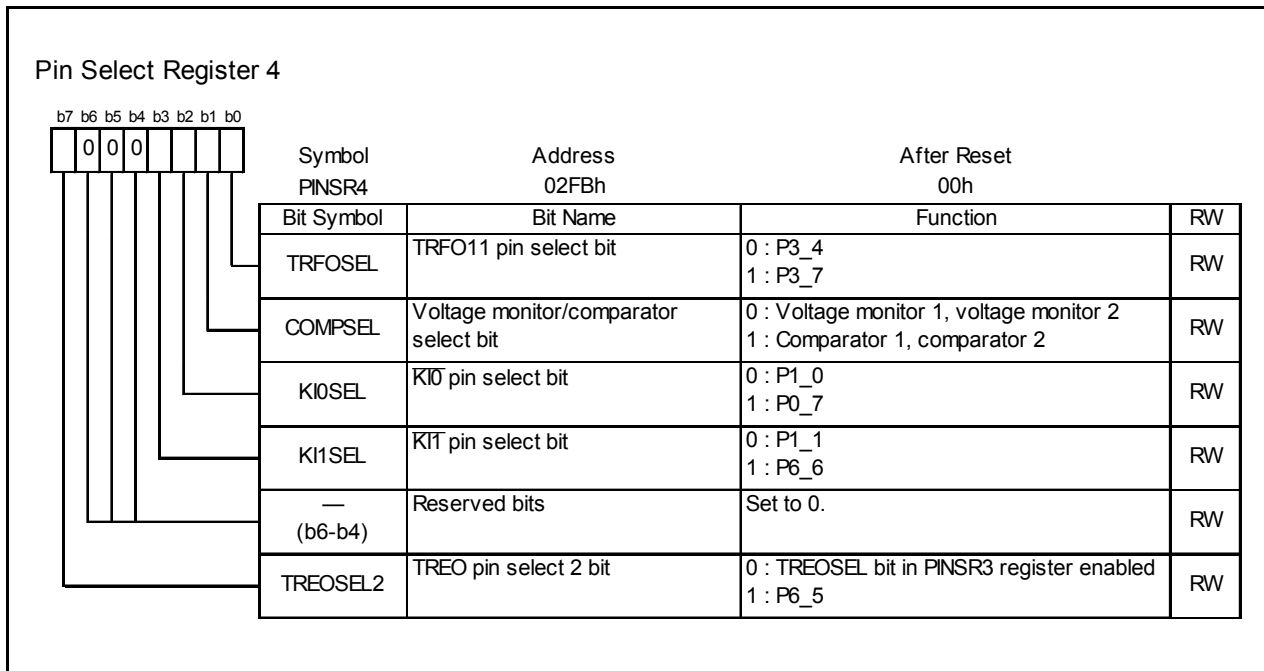


Figure 7.11 PINSR4 Register

## 7.3 Monitoring Comparison Results

### 7.3.1 Monitoring Comparator 1

After the following settings are made, the comparison result of comparator 1 can be monitored by the VW1C3 bit in the VW1C register after  $t_d(E-A)$  has elapsed (refer to **22. Electrical Characteristics**).

- (1) Set the COMPSEL bit in the PINSR4 register is set to 1 (comparator 1, comparator 2).
- (2) Set the VCAB5 bit in the VCAB register to 1 (VCMP1 pin input voltage).
- (3) Set the VCA26 bit in the VCA2 register to 1 (comparator 1 circuit enabled).

### 7.3.2 Monitoring Comparator 2

After the following settings are made, the comparison result of comparator 2 can be monitored by the VCA13 bit in the VCA1 register after  $t_d(E-A)$  has elapsed (refer to **22. Electrical Characteristics**).

- (1) Set the COMPSEL bit in the PINSR4 register to 1 (comparator 1, comparator 2).
- (2) Set the VCAB6 bit in the VCAB register to 1 (VCMP2 pin input voltage).
- (3) Set the VCA27 bit in the VCA2 register to 1 (comparator 2 circuit enabled).

## 7.4 Functional Description

Comparator 1 and comparator 2 operate independently.

The comparison result of the reference input voltage and analog input voltage can be read by software. The result can also be output from the VCOU<sub>Ti</sub> (i = 1 or 2) pin. An internal reference voltage or input voltage to the CVREF pin can be selected as the reference input voltage. The comparator 1 interrupt or the comparator 2 interrupt also can be used by selecting non-maskable or maskable for each interrupt.

### 7.4.1 Comparator 1

Table 7.3 lists the Procedure for Setting Bits Associated with Comparator 1 Interrupt, Figure 7.12 shows an Operating Example of Comparator 1 (When Digital Filter Enabled), and Figure 7.13 shows an Operating Example of Comparator 1 (When Digital Filter Disabled).

**Table 7.3 Procedure for Setting Bits Associated with Comparator 1 Interrupt**

Step	When Using Digital Filter	When Not Using Digital Filter
1	Set the COMPSEL bit in the PINSR4 register to 1 (comparator 1, comparator 2)	
2	Set the VCAB5 bit in the VCAB register to 1 (VCMP1 pin input voltage)	
3	Set the VCA26 bit in the VCA2 register to 1 (comparator 1 circuit enabled)	
4	Wait for $t_d(E-A)$	
5	Select the interrupt type by the IRQ1SEL bit in the ALCMR register	
6	Select the sampling clock by bits VW1F0 and VW1F1 in the VW1C register	Set the VW1C1 bit in the VW1C register to 1 (digital filter disabled)
7 <sup>(1)</sup>	Set the VW1C1 bit in the VW1C register to 0 (digital filter enabled)	–
8	Select the interrupt request timing by the VCAC1 bit in the VCAC register and the VW1C7 bit in the VW1C register	
9	Set the VW1C2 bit in the VW1C register to 0	
10	Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on)	–
11	Wait for 2 cycles of the sampling clock of the digital filter.	– (No wait time required)
12	Set the VW1C0 bit in the VW1C register to 1 (comparator 1 interrupt enabled)	

NOTE:

1. When the VW1C0 bit is set to 0, steps 6 and 7 can be executed at the same time (with one instruction)

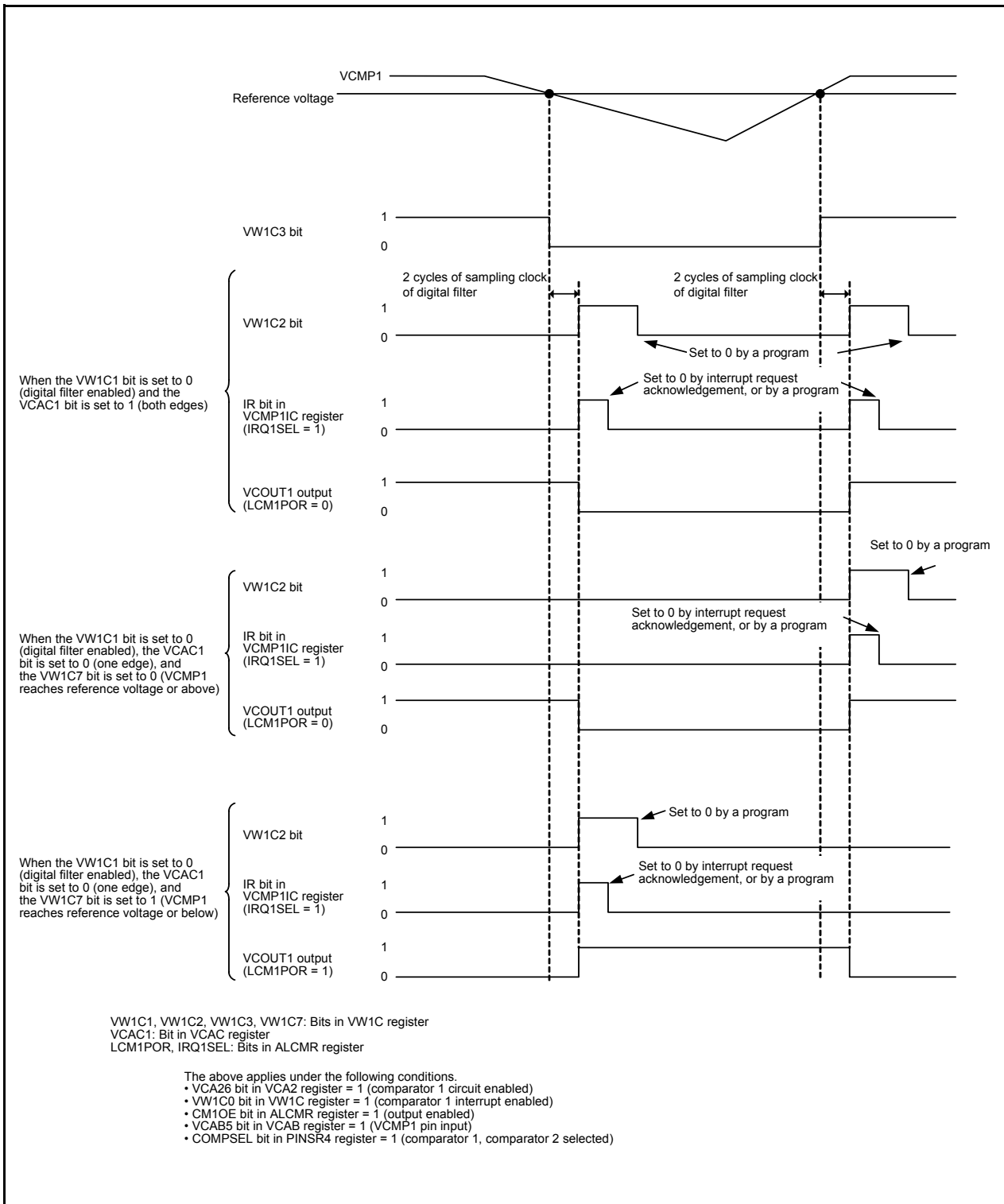


Figure 7.12 Operating Example of Comparator 1 (When Digital Filter Enabled)

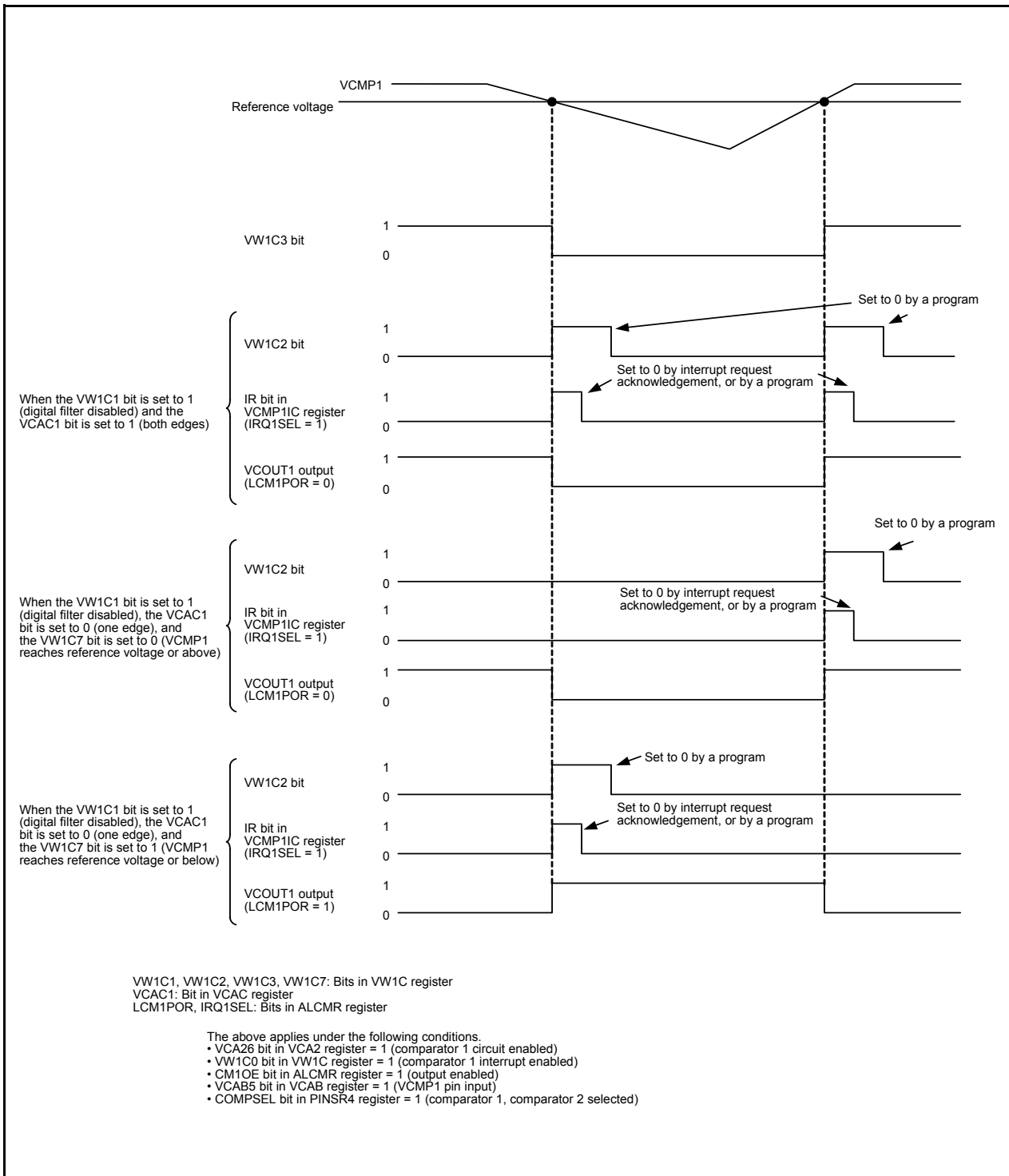


Figure 7.13 Operating Example of Comparator 1 (When Digital Filter Disabled)

## 7.4.2 Comparator 2

Table 7.4 lists the Procedure for Setting Bits Associated with Comparator 2 Interrupt, Figure 7.14 shows an Operating Example of Comparator 2 (When Digital Filter Enabled), and Figure 7.15 shows an Operating Example of Comparator 2 (When Digital Filter Disabled).

**Table 7.4 Procedure for Setting Bits Associated with Comparator 2 Interrupt**

Step	When Using Digital Filter	When Not Using Digital Filter
1	Set the COMPSEL bit in the PINSR4 register to 1 (comparator 1, comparator 2)	
2	Set the VCAB6 bit in the VCAB register to 1 (VCMP2 pin input voltage)	
3	Set the VCA27 bit in the VCA2 register to 1 (comparator 2 circuit enabled)	
4	Wait for $t_d(E-A)$	
5	Select the interrupt type by the IRQ2SEL bit in the ALCMR register	
6	Select the sampling clock by bits VW2F0 and VW2F1 in the VW2C register	Set the VW2C1 bit in the VW2C register to 1 (digital filter disabled)
7 <sup>(1)</sup>	Set the VW2C1 bit in the VW2C register to 0 (digital filter enabled)	–
8	Select the interrupt request timing by the VCAC2 bit in the VCAC register and the VW2C7 bit in the VW2C register	
9	Set the VW2C2 bit in the VW2C register to 0	
10	Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on)	–
11	Wait for 2 cycles of the sampling clock of the digital filter.	– (No wait time required)
12	Set the VW2C0 bit in the VW2C register to 1 (comparator 2 interrupt enabled)	

**NOTE:**

1. When the VW2C0 bit is set to 0, steps 6 and 7 can be executed at the same time (with one instruction).

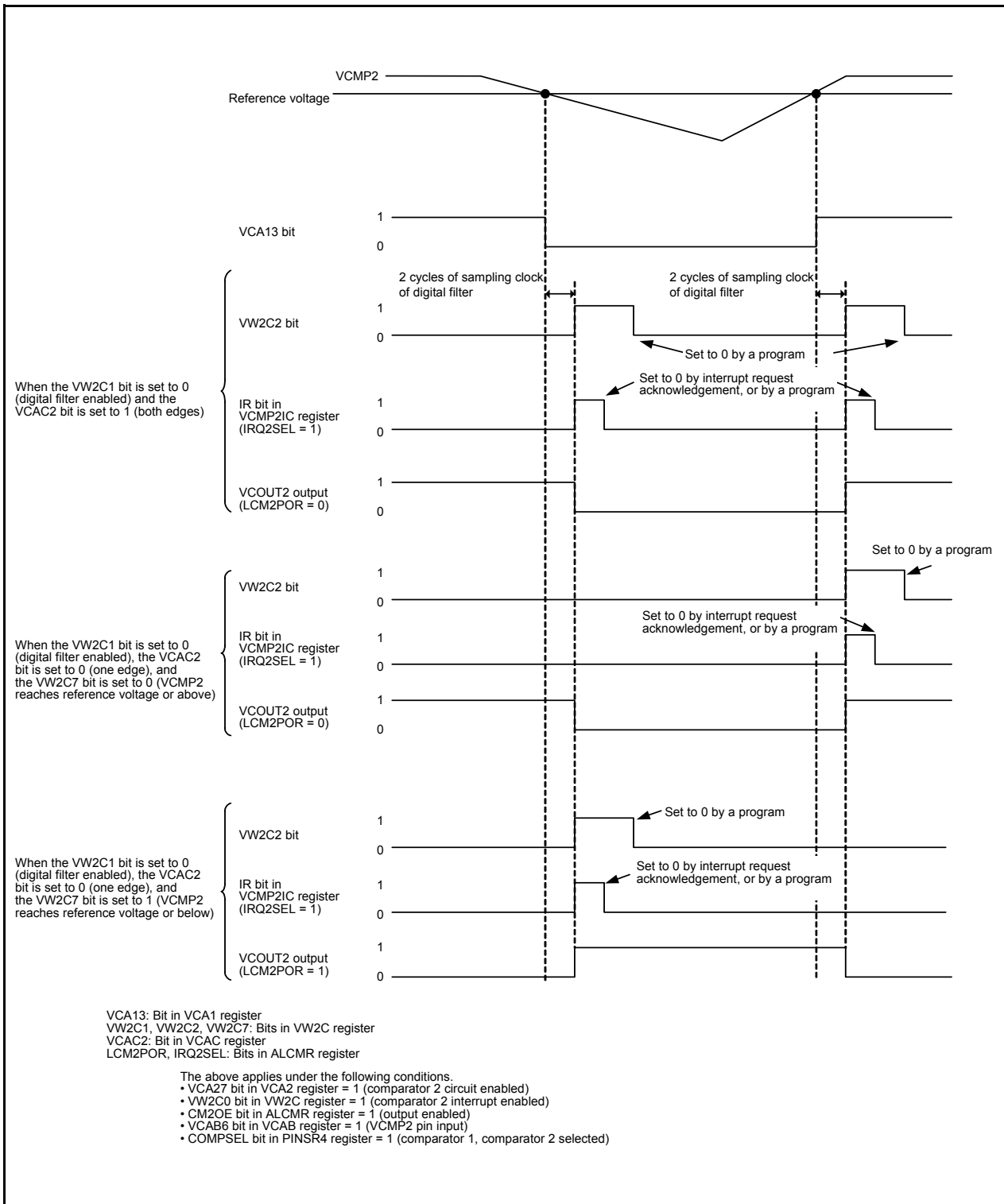


Figure 7.14 Operating Example of Comparator 2 (When Digital Filter Enabled)



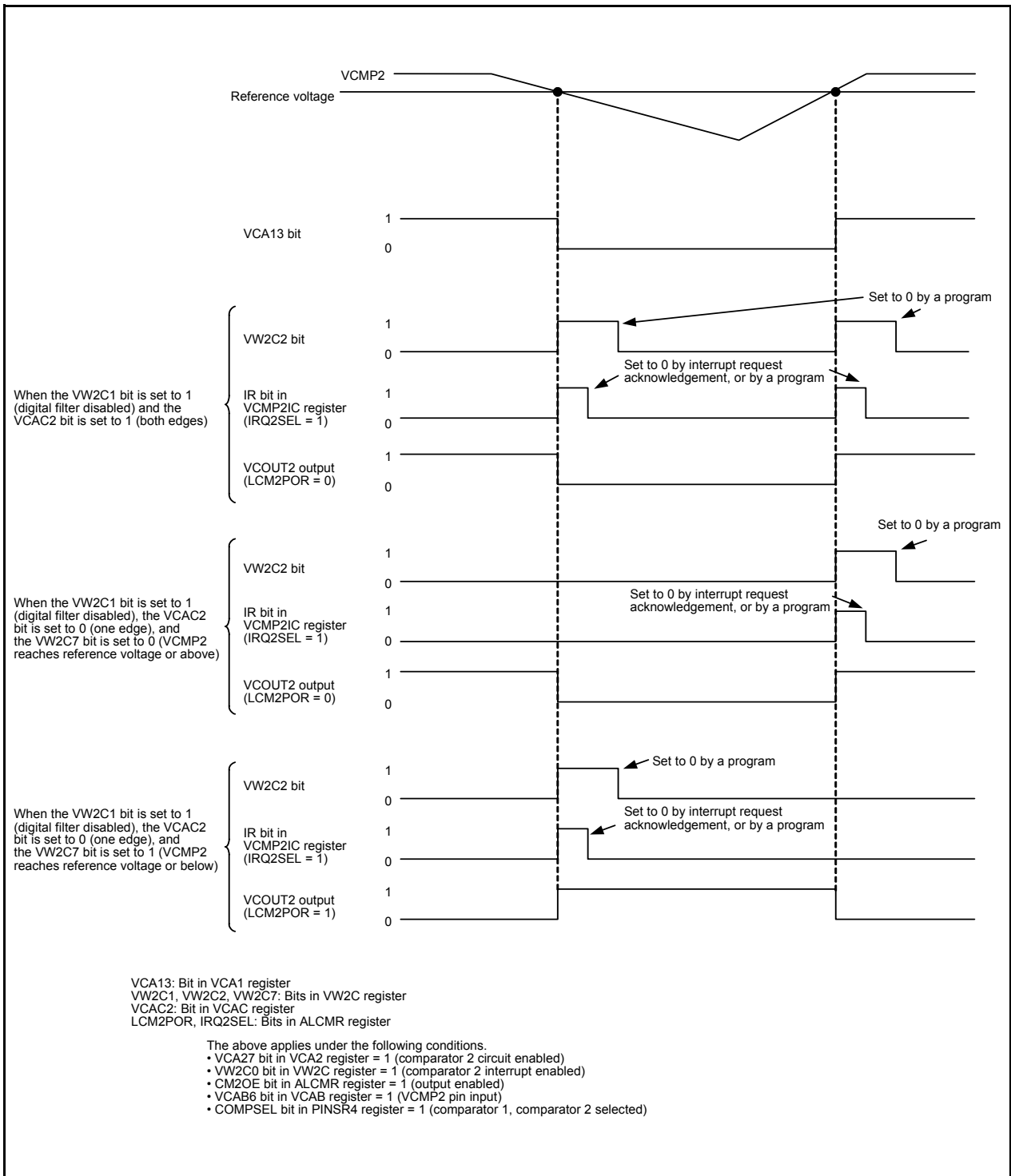


Figure 7.15 Operating Example of Comparator 2 (When Digital Filter Disabled)

## 7.5 Comparator 1 and Comparator 2 Interrupts

Two interrupt requests are generated, one each for comparator 1 and comparator 2. Non-maskable or maskable can be selected for each interrupt type. Refer to **13. Interrupts** for interrupts.

### 7.5.1 Non-Maskable Interrupts

When IRQiSEL (i = 1 or 2) bit in the ALCMR register is set to 0, the comparator i interrupt functions as a non-maskable interrupt. When the selected interrupt request timing occurs, the VWiC2 bit in the VWiC register is set to 1. At this time, a non-maskable interrupt request for comparator i is generated.

### 7.5.2 Maskable Interrupts

When the IRQiSEL (i = 1 or 2) bit in the ALCMR register is set to 1, the comparator i interrupt functions as a maskable interrupt. The comparator i interrupt uses the single VCMPiIC register (bits IR and ILVL0 to ILVL2) and a single vector. When the selected interrupt request timing occurs, the VWiC2 bit in the VWiC register is set to 1. At this time, the IR bit in the VCMPiIC register is set to 1 (interrupt requested).

Refer to **13.1.6 Interrupt Control** for the VCMPiIC register and **13.1.5.2 Relocatable Vector Tables** for interrupt vectors.

## 7.6 Adjusting Internal Reference Voltage (Vref)

The level of the internal reference voltage (Vref) can be adjusted with the value of the BGRTRM register. The values for correcting the Vref are stored in registers BGRTRMA and BGRTRMB before shipping the MCU. The value of the BGRTRMA register is the same as that of the BGRTRM register after reset.

To use separate correction values to match the supply voltage ranges, transfer them from registers BGRTRMA and BGRTRMB to the BGRTRM register. Figure 7.16 shows the Procedure for Adjusting Internal Reference Voltage (Vref).

When the BGRCR0 bit in the BGRCCR register to 1 (disabled), the internal reference voltage (Vref) adjustment circuit (BGR trimming circuit) is disabled and the value of the BGRTRM register is also disabled.

When the BGR trimming circuit is disabled, the accuracy of the internal reference voltage (Vref) is not guaranteed. Disable voltage detection circuits 0 to 2 and disable comparators 1 and 2 with the internal reference voltage selected. The high-speed on-chip oscillator should also be stopped as necessary because the precision of its oscillation frequency is not also guaranteed.

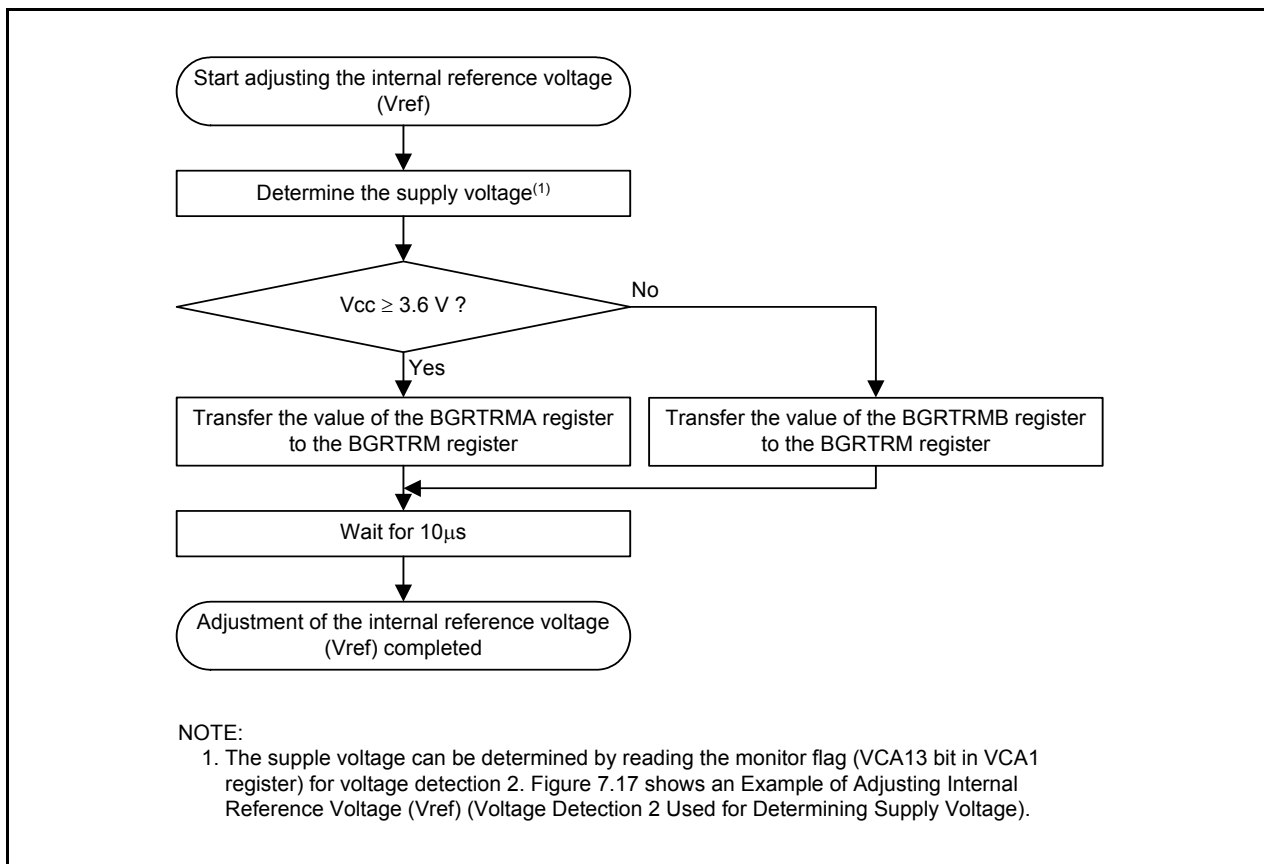
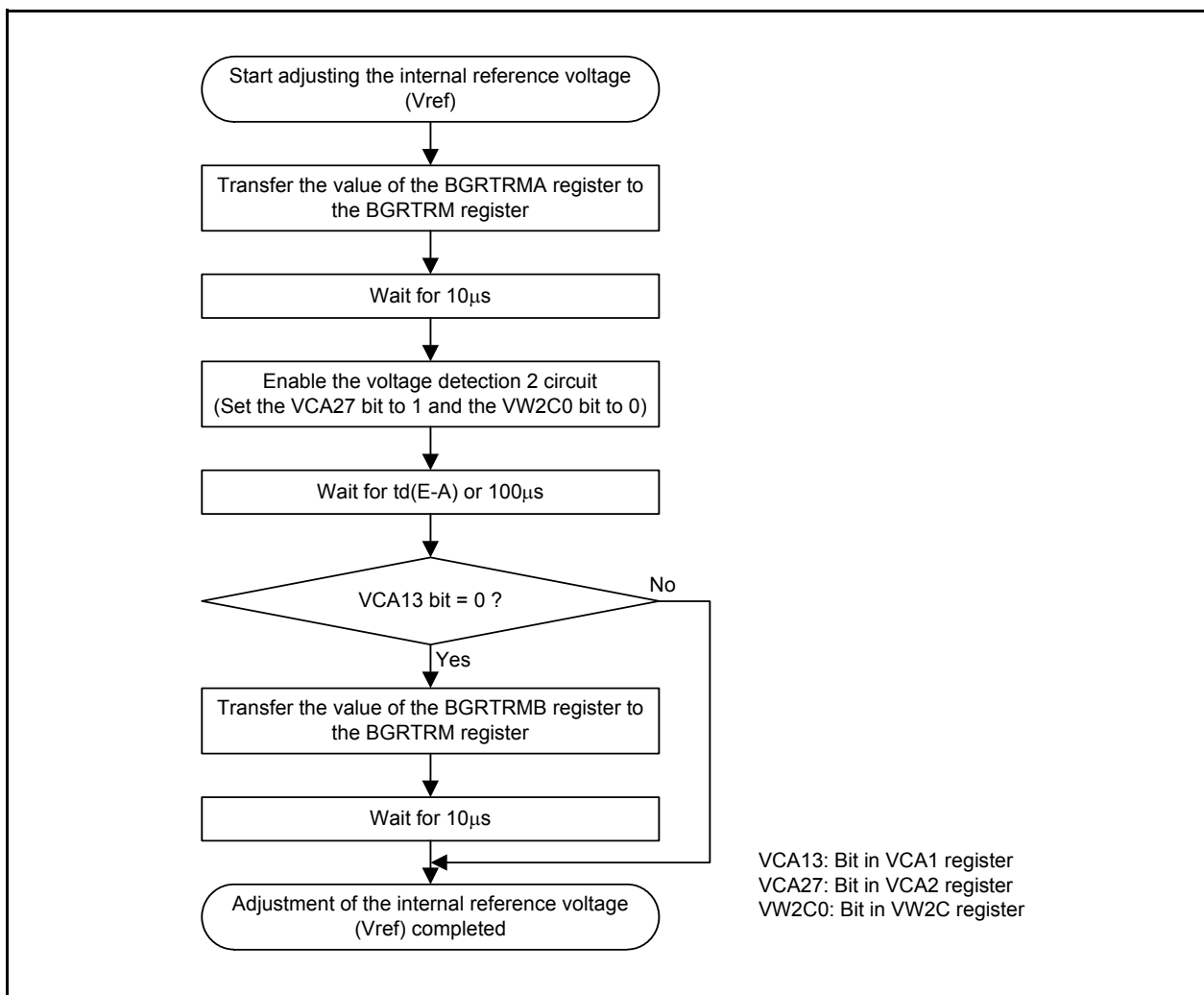


Figure 7.16 Procedure for Adjusting Internal Reference Voltage (Vref)



**Figure 7.17 Example of Adjusting Internal Reference Voltage (Vref) (Voltage Detection 2 Used for Determining Supply Voltage)**

## 8. I/O Ports

There are 27 I/O ports P0\_4 to P0\_7, P1, P3, P4\_3, P4\_5, P6\_0, P6\_3 to P6\_6.

When the XCIN clock oscillation circuit is not used, P4\_3 can be used as an I/O port and P4\_4 can be used as an output port.

Table 8.1 lists an Overview of I/O Ports.

**Table 8.1 Overview of I/O Ports**

Ports	I/O	Type of Output	I/O Setting	Internal Pull-Up Resistor
P0_4 to P0_7, P1, P3	I/O	CMOS3 State	Set per bit	Set every 4 bits <sup>(1)</sup>
P4_3	I/O	CMOS3 State	Set per bit	Set every bit <sup>(2)</sup>
P4_4	Output	CMOS3 State	Set per bit <sup>(3)</sup>	None
P4_5	I/O	CMOS3 State	Set per bit	Set every bit <sup>(2)</sup>
P6_0, P6_3	I/O	CMOS3 State	Set per bit	Set every 2 bits <sup>(2)</sup>
P6_4 to P6_6	I/O	CMOS3 State	Set per bit	Set every 3 bits <sup>(2)</sup>

NOTES:

1. In input mode, whether an internal pull-up resistor is connected or not can be selected by PUR0 register.
2. In input mode, whether an internal pull-up resistor is connected or not can be selected by PUR1 register.
3. Do not use port P4\_4 as an input port (input mode).

### 8.1 Functions of I/O Ports

The PDi\_j (j = 0 to 7) bit in the PDi (i = 0, 1, 3, 4, 6) register controls I/O of the ports P0\_4 to P0\_7, P1, P3, P4\_3 to P4\_5, P6\_0, P6\_3 to P6\_6. The Pi register consists of a port latch to hold output data and a circuit to read pin states.

Figures 8.1 to 8.3 show the Configurations of I/O Ports. Table 8.2 lists the Functions of I/O Ports. Also, Figure 8.5 shows the PDi (i = 0, 1, 3, 4, or 6) Register. Figure 8.6 shows the Pi (i = 0, 1, 3, 4, or 6) Register, Figure 8.7 shows Registers PINSR2, PINSR3, and PINSR4, Figure 8.8 shows the PMR Register, Figure 8.9 shows Registers PUR0 and PUR1.

**Table 8.2 Functions of I/O Ports**

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

i = 0, 1, 3, 4, 6, j = 0 to 7

NOTE:

1. Nothing is assigned to bits PD0\_0 to PD0\_3, PD4\_0 to PD4\_2, PD4\_6, PD4\_7, PD6\_1, PD6\_2, PD6\_7.

## 8.2 Effect on Peripheral Functions

I/O ports function as I/O ports for peripheral functions (refer to **Table 1.3 Pin Name Information by Pin Number**).

Table 8.3 lists the Setting of PDi\_j Bit when Functioning as I/O Ports for Peripheral Functions (i = 0, 1, 3, 4, 6, j = 0 to 7). Refer to the description of each function for information on how to set peripheral functions.

**Table 8.3 Setting of PDi\_j Bit when Functioning as I/O Ports for Peripheral Functions (i = 0, 1, 3, 4, 6, j = 0 to 7)**

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

NOTE:

- Nothing is assigned to bits PD0\_0 to PD0\_3, PD4\_0 to PD4\_2, PD4\_6, PD4\_7, PD6\_1, PD6\_2, PD6\_7.

## 8.3 Pins Other than Programmable I/O Ports

Figure 8.4 shows the Configuration of I/O Pins.

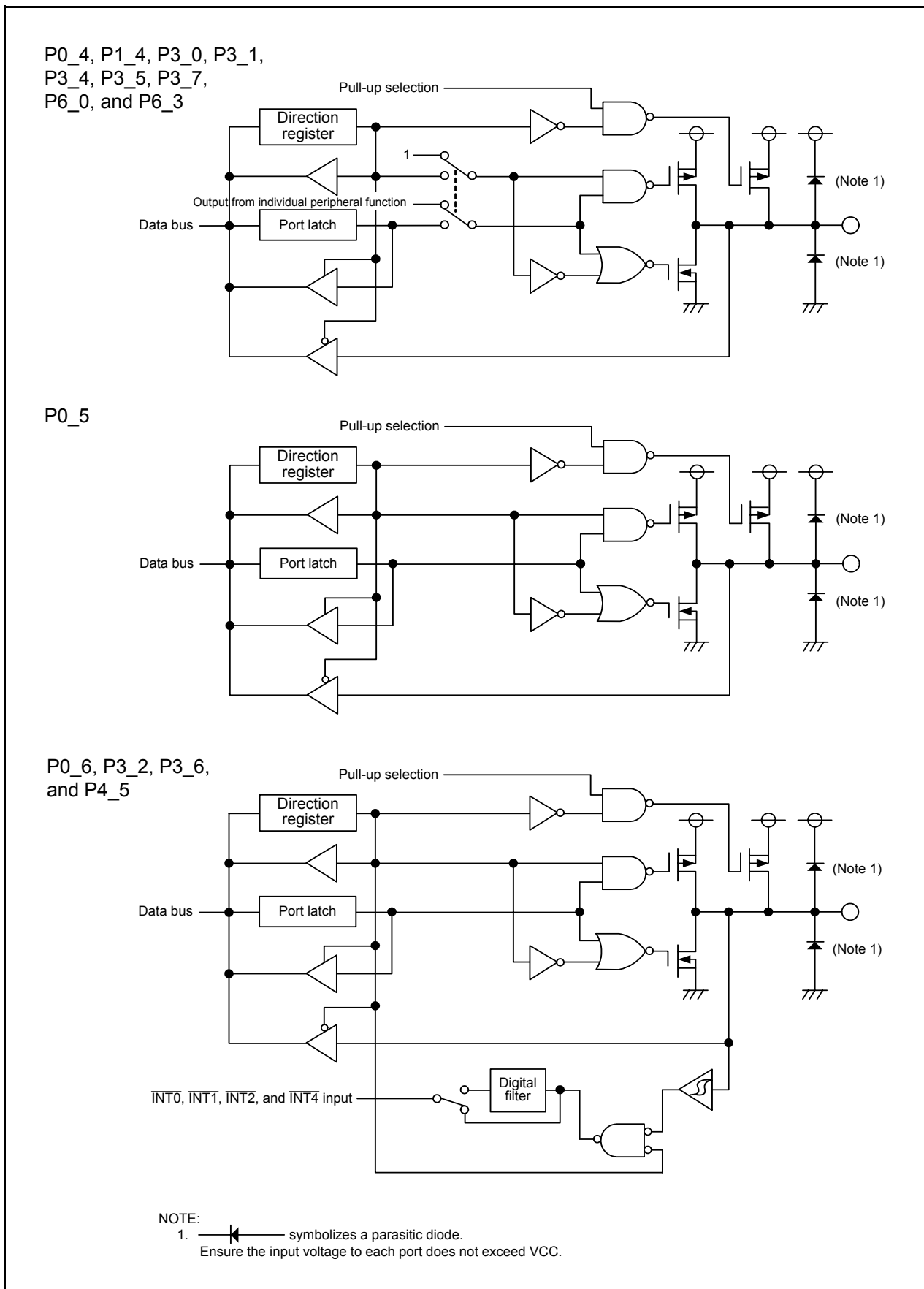


Figure 8.1 Configuration of I/O Ports (1)

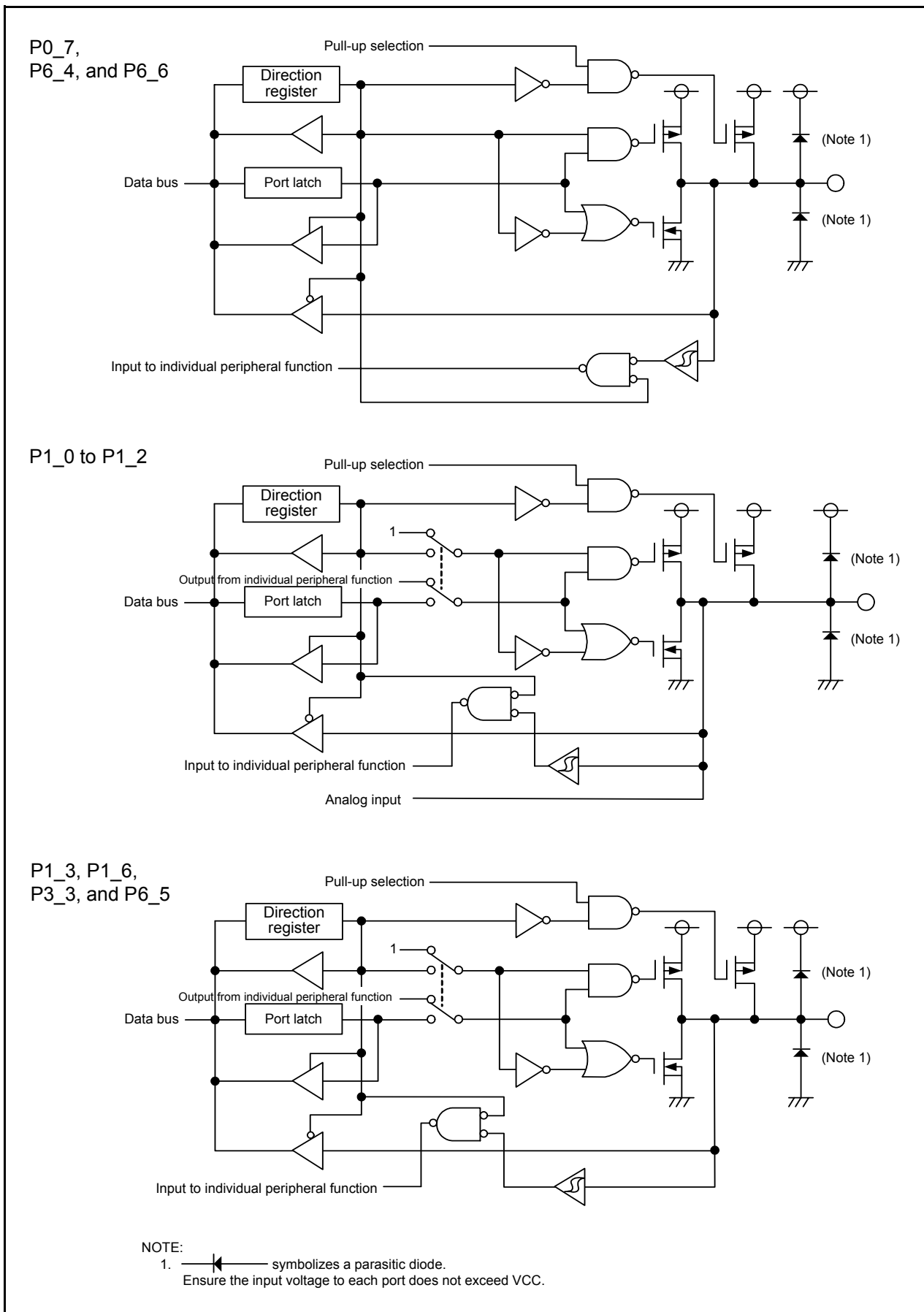


Figure 8.2 Configuration of I/O Ports (2)



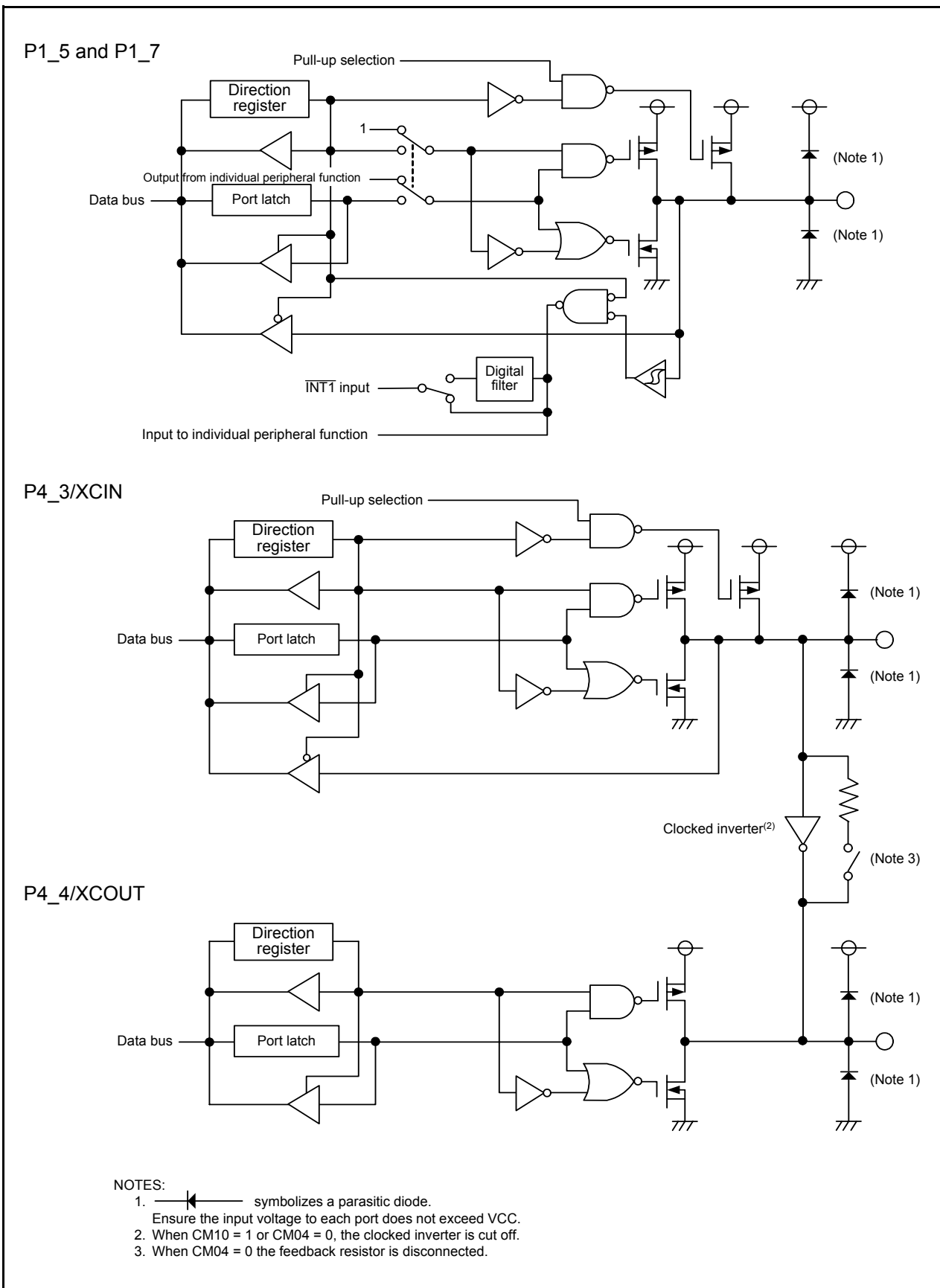


Figure 8.3 Configuration of I/O Ports (3)

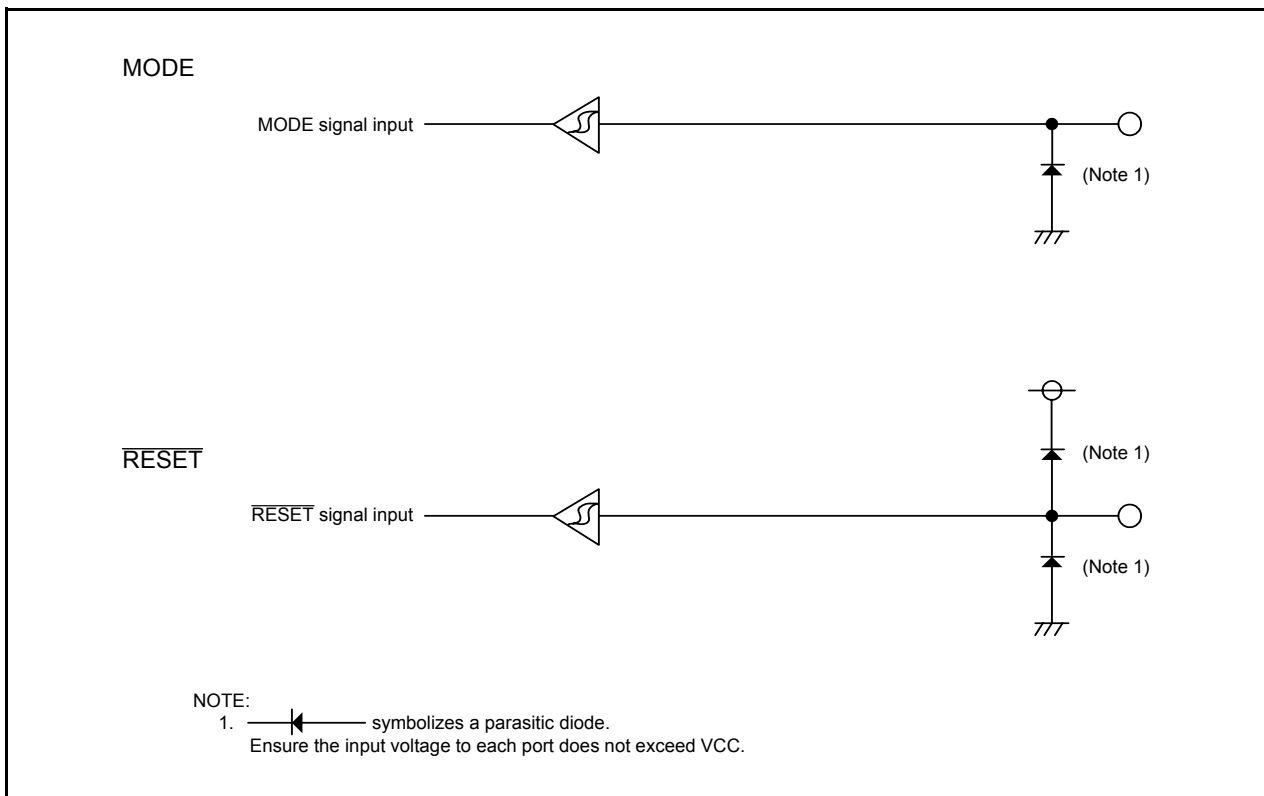


Figure 8.4 Configuration of I/O Pins

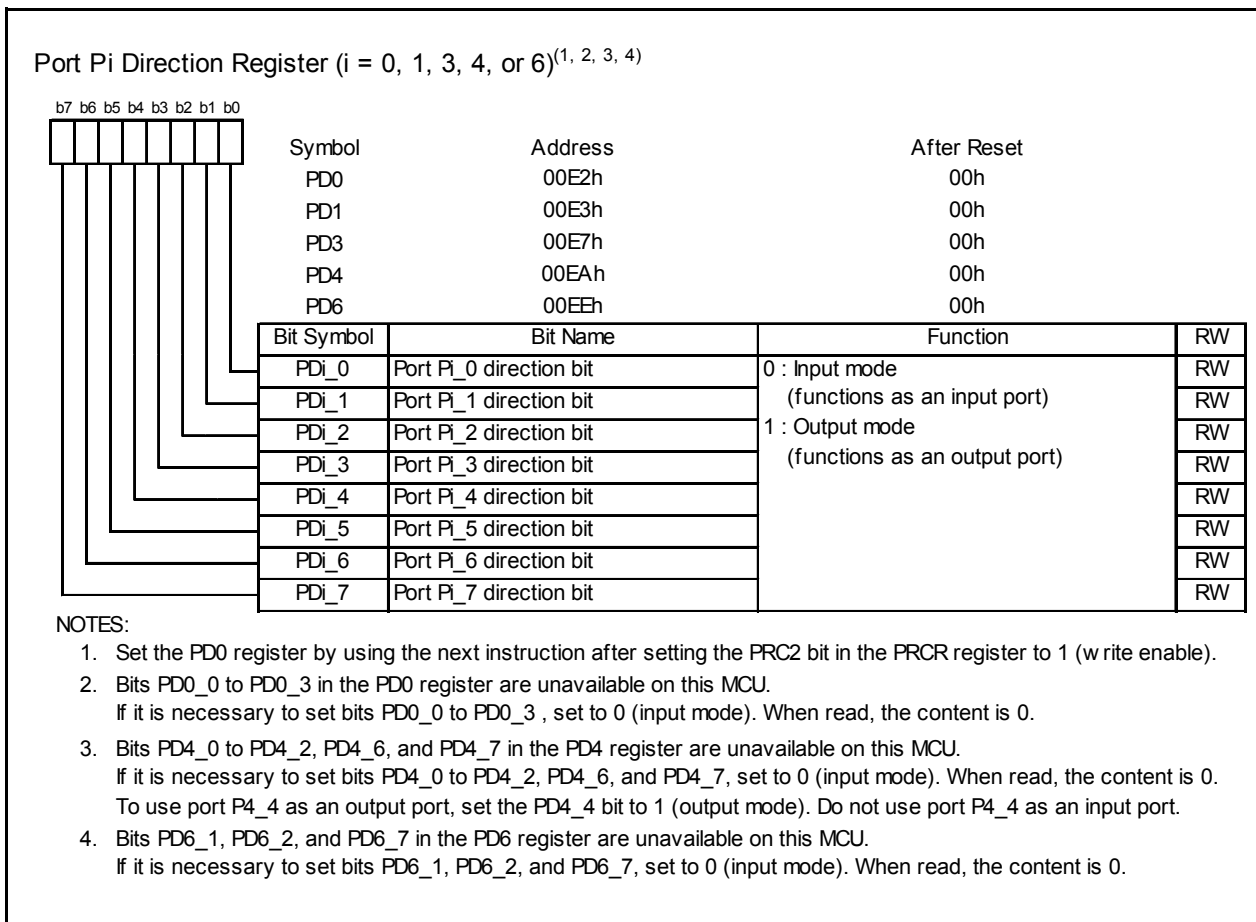
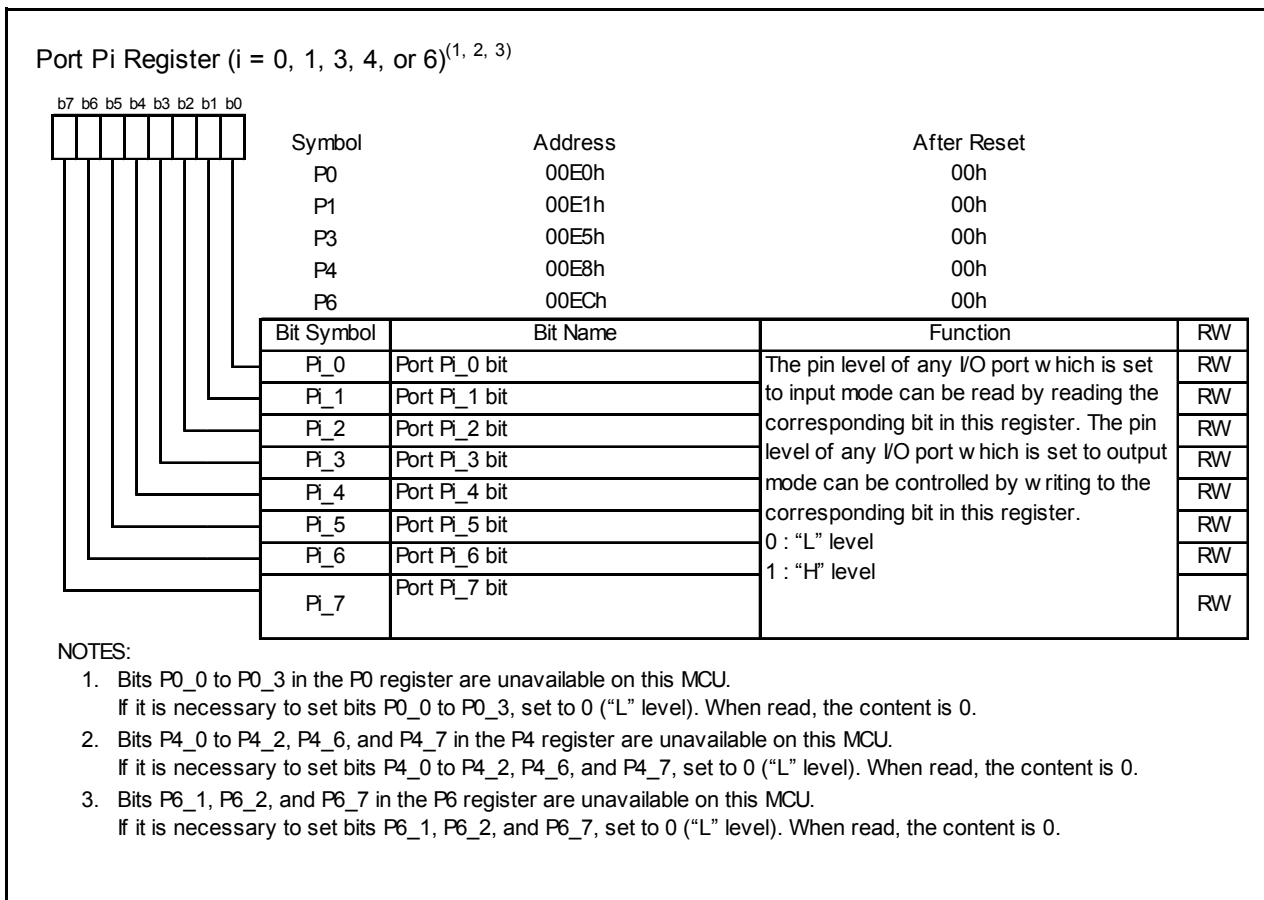


Figure 8.5 PDi (i = 0, 1, 3, 4, or 6) Register



**Figure 8.6 Pi (i = 0, 1, 3, 4, or 6) Register**

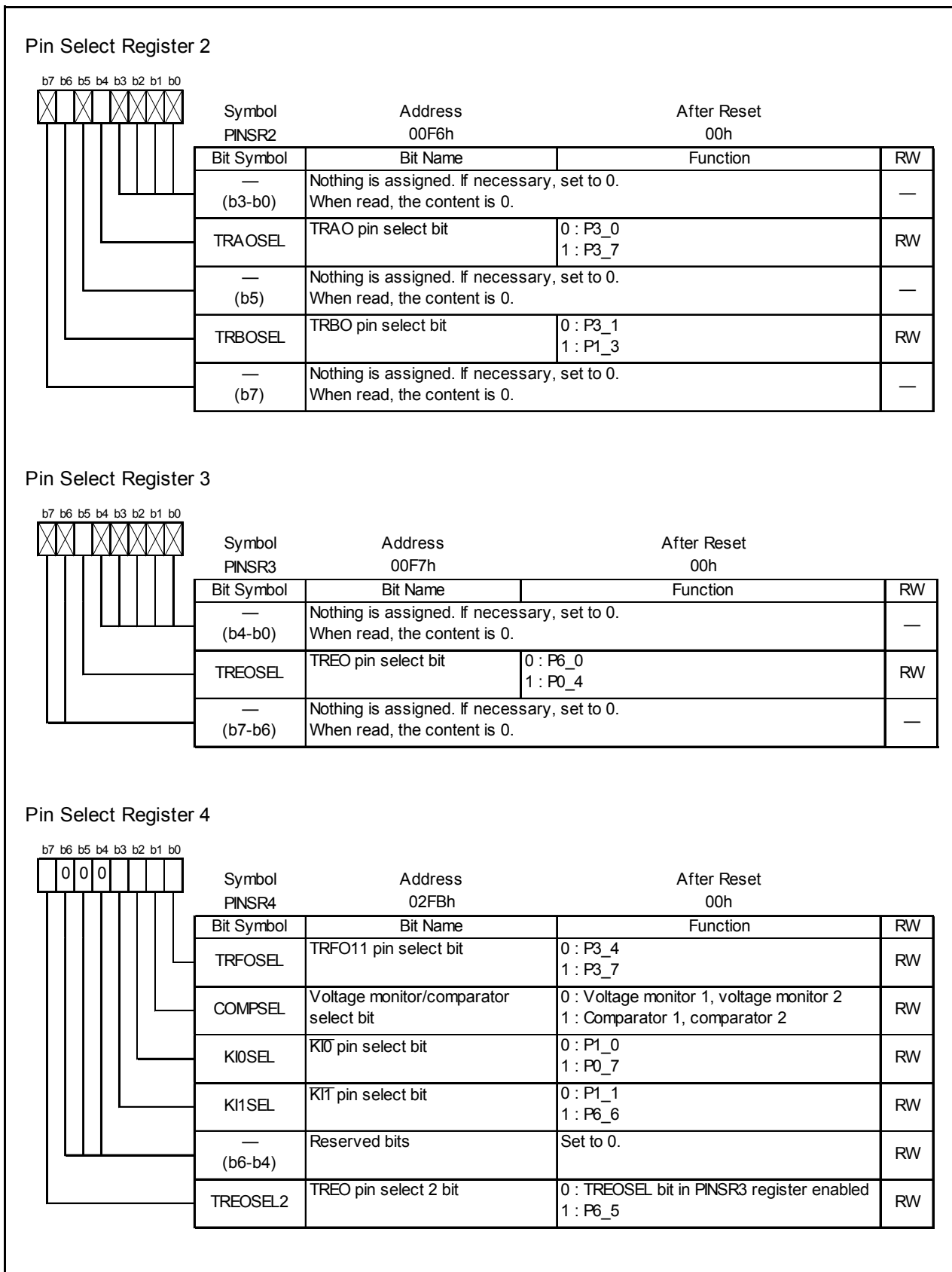


Figure 8.7 Registers PINSR2, PINSR3, and PINSR4

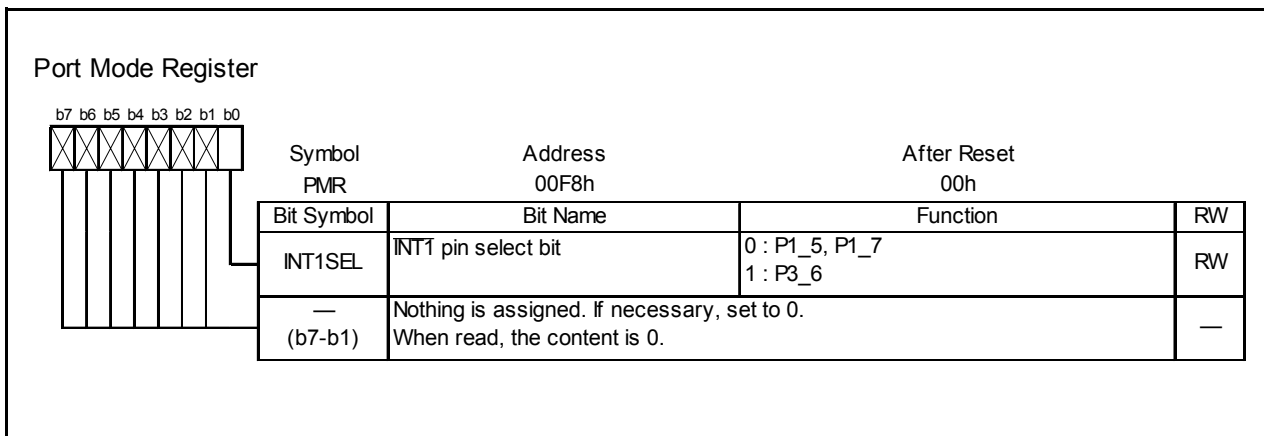


Figure 8.8 PMR Register

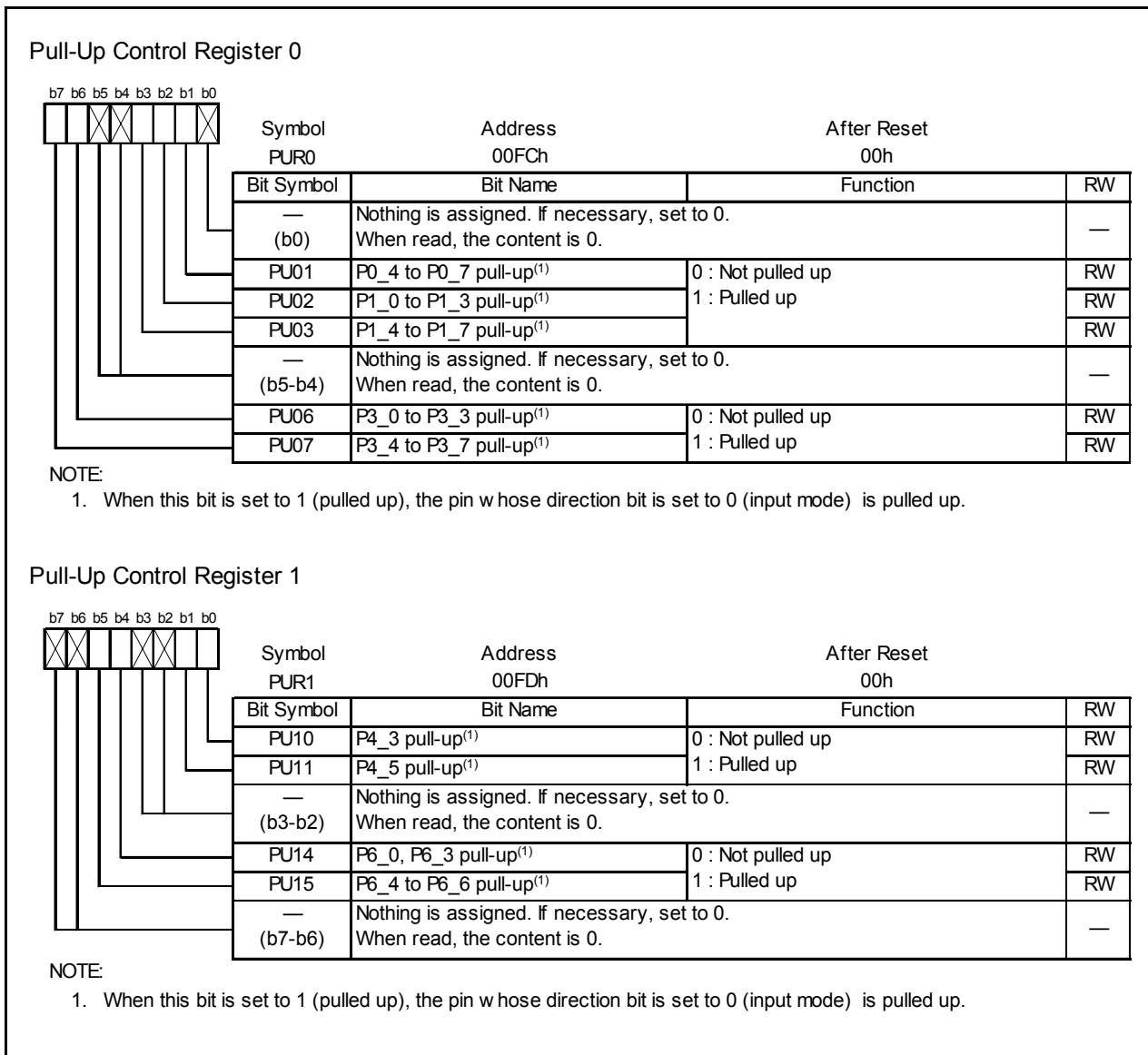


Figure 8.9 Registers PUR0 and PUR1

## 8.4 Port Setting

Table 8.4 to Table 8.33 list the port setting.

**Table 8.4 Port P0\_4/(TREQ)**

Register	PD0	PINSR4	PINSR3	TRECR1	Function
Bit	PD0_4	TREOSEL2	TREOSEL	TOENA	
Setting value	0	Other than 011b			Input port <sup>(1)</sup>
	1	Other than 011b			Output port
	X	0	1	1	TREQ output

X: 0 or 1

NOTE:

1. Pulled up by setting the PU01 bit in the PUR0 register to 1.

**Table 8.5 Port P0\_5**

Register	PD0	Function
Bit	PD0_5	
Setting value	0	Input port <sup>(1)</sup>
	1	Output port

NOTE:

1. Pulled up by setting the PU01 bit in the PUR0 register to 1.

**Table 8.6 Port P0\_6/ $\overline{\text{INT4}}$**

Register	PD0	INTEN2	Function
Bit	PD0_6	INT4EN	
Setting value	0	0	Input port <sup>(1)</sup>
	1	0	Output port
	0	1	$\overline{\text{INT4}}$ input <sup>(1)</sup>

NOTE:

1. Pulled up by setting the PU01 bit in the PUR0 register to 1.

**Table 8.7 Port P0\_7/( $\overline{\text{KI0}}$ )**

Register	PD0	PINSR4	KIEN	Function
Bit	PD0_7	KI0SEL	KI0EN	
Setting value	0	X	0	Input port <sup>(1)</sup>
	1	X	0	Output port
	0	1	1	$\overline{\text{KI0}}$ input <sup>(1)</sup>

X: 0 or 1

NOTE:

1. Pulled up by setting the PU01 bit in the PUR0 register to 1.

**Table 8.8 Port P1\_0/ $\overline{\text{KI0}}$ /TRFO00/VCMP1**

Register	PD1	TRFOUT	PINSR4	KIEN	VCAB	Function
Bit	PD1_0	TRFOUT0	KI0SEL	KI0EN	VCAB5	
Setting value	0	0	X	0	0	Input port <sup>(1)</sup>
	1	0	X	0	0	Output port
	X	1	X	0	0	TRFO00 output
	0	0	0	1	0	$\overline{\text{KI0}}$ input <sup>(1)</sup>
	0	0	X	0	1	VCMP1 input <sup>(1)</sup>

X: 0 or 1

NOTE:

1. Pulled up by setting the PU02 bit in the PUR0 register to 1.

**Table 8.9 Port P1\_1/ $\overline{\text{KI1}}$ /TRFO01/VCMP2**

Register	PD1	TRFOUT	PINSR4	KIEN	VCAB	Function
Bit	PD1_1	TRFOUT1	KI1SEL	KI1EN	VCAB6	
Setting value	0	0	X	0	0	Input port <sup>(1)</sup>
	1	0	X	0	0	Output port
	X	1	X	0	0	TRFO01 output
	0	0	0	1	0	$\overline{\text{KI1}}$ input <sup>(1)</sup>
	0	0	X	0	1	VCMP2 input <sup>(1)</sup>

X: 0 or 1

NOTE:

1. Pulled up by setting the PU02 bit in the PUR0 register to 1.

**Table 8.10 Port P1\_2/ $\overline{\text{KI2}}$ /TRFO02/CVREF**

Register	PD1	TRFOUT	KIEN	VCAB	Function
Bit	PD1_2	TRFOUT2	KI2EN	VCAB7	
Setting value	0	0	0	0	Input port <sup>(1)</sup>
	1	0	0	0	Output port
	X	1	0	0	TRFO02 output
	0	0	1	0	$\overline{\text{KI2}}$ input <sup>(1)</sup>
	0	0	0	0	1

X: 0 or 1

NOTE:

1. Pulled up by setting the PU02 bit in the PUR0 register to 1.



**Table 8.11 Port P1\_3/KI3/VCOUT1/(TRBO)**

Register	PD1	Timer RB Setting	KIEN	ALCMR	Function
Bit	PD1_3	–	KI3EN	CM1OE	
Setting value	0	Other than TRBO usage conditions	0	0	Input port <sup>(1)</sup>
	1	Other than TRBO usage conditions	0	0	Output port
	0	Other than TRBO usage conditions	1	0	KI3 input <sup>(1)</sup>
	X	Refer to <b>Table 8.12 TRBO Pin Setting</b>	0	0	TRBO output
	X	Other than TRBO usage conditions	0	1	VCOUT1 output

X: 0 or 1

NOTE:

1. Pulled up by setting the PU02 bit in the PUR0 register to 1.

**Table 8.12 TRBO Pin Setting**

Register	PINSR2	TRBIOC	TRBMR		Function
Bit	TRBOSEL	TOCNT <sup>(1)</sup>	TMOD1	TMOD0	
Setting value	1	0	0	1	Programmable waveform generation mode
	1	0	1	0	Programmable one-shot generation mode
	1	0	1	1	Programmable wait one-shot generation mode
	1	1	0	1	P1_3 output port
	Other than above				Other than TRBO usage conditions

NOTE:

1. Set the TOCNT bit in the TRBIOC register to 0 in modes except for programmable waveform generation mode.

**Table 8.13 Port P1\_4/TXD0**

Register	PD1	U0MR			Function	
Bit	PD1_4	SMD2	SMD1	SMD0		
Setting value	0	0	0	0	Input port <sup>(1)</sup>	
	1	0	0	0	Output port	
	X	1	0	0	1	TXD0 output <sup>(2)</sup>
					0	
					1	
					0	

X: 0 or 1

NOTES:

1. Pulled up by setting the PU03 bit in the PUR0 register to 1.
2. N-channel open-drain output by setting the NCH bit in the U0C0 register to 1.

**Table 8.14 Port P1\_5/RXD0/(TRAIO)/(INT1)**

Register	PD1	TRAIOC		TRAMR			PMR	INTEN	Function
Bit	PD1_5	TIOSEL	TOPCR <sup>(2)</sup>	TMOD2	TMOD1	TMOD0	INT1SEL	INT1EN	
Setting value	0	0	X	X	X	X	X	0	Input port <sup>(1)</sup>
		1	1	0	0	1			
		0	0						
	1	0	X	X	X	X	X	0	Output port
		1	0	0	0	0			
	0	0	X	X	X	X	X	0	RXD0 input <sup>(1)</sup>
		1	0	Other than 001b					
	0	1	0	0	0	0	0	0	1
1			1						
0	1	0	Other than 000b, 001b			X	X	TRAIO input <sup>(1)</sup>	
0	1	0	Other than 000b, 001b			0	1	TRAIO input/ $\overline{\text{INT1}}$ input <sup>(1)</sup>	
X	1	0	0	0	1	X	X	TRAIO output	

X: 0 or 1

NOTES:

1. Pulled up by setting the PU03 bit in the PUR0 register to 1.
2. Set the TOPCR bit in the TRAIOC register to 0 in modes except for pulse output mode.

**Table 8.15 Port P1\_6/CLK0/VCOUT2**

Register	PD1	ALCMR	UOMR				Function
Bit	PD1_6	CM2OE	CKDIR	SMD2	SMD1	SMD0	
Setting value	0	0	0	Other than 001b			Input port <sup>(1)</sup>
			1	X	X	X	
	1	0	X	Other than 001b			Output port
	X	0	0	0	0	1	CLK0 output
	0	0	1	X	X	X	CLK0 input <sup>(1)</sup>
X	1	X	X	X	X	VCOUT2 output	

X: 0 or 1

NOTE:

1. Pulled up by setting the PU03 bit in the PUR0 register to 1.

**Table 8.16 Port P1\_7/TRAIO/INT1**

Register	PD1	TRAIOC		TRAMR			PMR	INTEN	Function	
Bit	PD1_7	TIOSEL	TOPCR <sup>(2)</sup>	TMOD2	TMOD1	TMOD0	INT1SEL	INT1EN		
Setting value	0	1	X	X	X	X	X	0	Input port <sup>(1)</sup>	
		0	1	0	0	1				
		0	0							
	1	1	X	X	X	X	X	0	Output port	
		0	0	0	0	0				
	0	0	0	0	0	0	0	0	1	$\overline{\text{INT1}}$ input <sup>(1)</sup>
			1							
	0	0	0	Other than 000b, 001b			X	X	TRAIO input <sup>(1)</sup>	
0	0	0	Other than 000b, 001b			0	1	TRAIO input/ $\overline{\text{INT1}}$ input <sup>(1)</sup>		
X	0	0	0	0	1	X	X	TRAIO output		

X: 0 or 1

NOTES:

1. Pulled up by setting the PU03 bit in the PUR0 register to 1.
2. Set the TOPCR bit in the TRAIOC register to 0 in modes except for pulse output mode.

**Table 8.17 Port P3\_0/TRAO**

Register	PD3	PINSR2	TRAIOC	Function
Bit	PD3_0	TRAOSEL	TOENA	
Setting value	0	X	0	Input port <sup>(1)</sup>
	1	X	0	Output port
	X	0	1	TRAO output

X: 0 or 1

NOTE:

1. Pulled up by setting the PU06 bit in the PUR0 register to 1.

**Table 8.18 Port P3\_1/TRBO**

Register	PD3	Timer RB Setting	Function
Bit	PD3_1	–	
Setting value	0	Other than TRBO usage conditions	Input port <sup>(1)</sup>
	1	Other than TRBO usage conditions	Output port
	X	Refer to <b>Table 8.19 TRBO Pin Setting</b>	TRBO output

X: 0 or 1

NOTE:

1. Pulled up by setting the PU06 bit in the PUR0 register to 1.

**Table 8.19 TRBO Pin Setting**

Register	PINSR2	TRBIOC	TRBMR		Function
Bit	TRBOSEL	TOCNT <sup>(1)</sup>	TMOD1	TMOD0	
Setting value	0	0	0	1	Programmable waveform generation mode
	0	0	1	0	Programmable one-shot generation mode
	0	0	1	1	Programmable wait one-shot generation mode
	0	1	0	1	P3_1 output port
	Other than above				Other than TRBO usage conditions

NOTE:

1. Set the TOCNT bit in the TRBIOC register to 0 in modes except for programmable waveform generation mode.

**Table 8.20 Port P3\_2/ $\overline{\text{INT2}}$** 

Register	PD3	INTEN	Function
Bit	PD3_2	INT2EN	
Setting value	0	0	Input port <sup>(1)</sup>
	1	0	Output port
	0	1	$\overline{\text{INT2}}$ input

NOTE:

1. Pulled up by setting the PU06 bit in the PUR0 register to 1.

**Table 8.21 Port P3\_3/TRFO10/TRFI**

Register	PD3	TRFOUT	Function
Bit	PD3_3	TRFOUT3	
Setting value	0	0	Input port <sup>(1)</sup>
	1	0	Output port
	X	1	TRFO10 output
	0	0	TRFI input <sup>(1)</sup>

X: 0 or 1

NOTE:

1. Pulled up by setting the PU06 bit in the PUR0 register to 1.

**Table 8.22 Port P3\_4/TRFO11**

Register	PD3	PINSR4	TRFOUT	Function
Bit	PD3_4	TRFOSEL	TRFOUT4	
Setting value	0	X	0	Input port <sup>(1)</sup>
	1	X	0	Output port
	X	0	1	TRFO11 output

X: 0 or 1

NOTE:

1. Pulled up by setting the PU07 bit in the PUR0 register to 1.

**Table 8.23 Port P3\_5/TRFO12**

Register	PD3	TRFOUT	Function
Bit	PD3_5	TRFOUT5	
Setting value	0	0	Input port <sup>(1)</sup>
	1	0	Output port
	X	1	TRFO12 output

X: 0 or 1

NOTE:

1. Pulled up by setting the PU07 bit in the PUR0 register to 1.

**Table 8.24 Port P3\_6/ $\overline{\text{INT1}}$** 

Register	PD3	PMR	INTEN	Function
Bit	PD3_6	INT1SEL	INT1EN	
Setting value	0	X	0	Input port <sup>(1)</sup>
	1	X	0	Output port
	0	1	1	$\overline{\text{INT1}}$ input <sup>(1)</sup>

X: 0 or 1

NOTE:

1. Pulled up by setting the PU07 bit in the PUR0 register to 1.

**Table 8.25 Port P3\_7/(TRAO)/(TRFO11)**

Register	PD3	PINSR2	TRAI OC	PINSR4	TRFOUT	Function
Bit	PD3_7	TRAOSEL	TOENA	TRFOSEL	TRFOUT4	
Setting value	0	X	0	X	0	Input port <sup>(1)</sup>
	1	X	0	X	0	Output port
	X	1	1	X	0	TRAO output
	X	X	0	1	1	TRFO11 output

X: 0 or 1

NOTE:

1. Pulled up by setting the PU07 bit in the PUR0 register to 1.

**Table 8.26 Port P4\_3/(XCIN)**

Register	PD4	CM0	CM1		Circuit specifications		Function
Bit	PD4_3	CM04	CM10	CM12	Oscillation buffer	Feedback resistor	
Setting value	0	0	X	X	OFF	OFF	Input port <sup>(1, 2)</sup>
	1	0	X	X	OFF	OFF	Output port <sup>(2)</sup>
	X	1	0	0	ON	ON	XCIN clock oscillation (on-chip feedback resistor enabled)
	X	1	0	1	ON	OFF	XCIN clock oscillation (on-chip feedback resistor disabled)
	X	1	1	0	OFF	ON	XCIN clock oscillation stop
				1	OFF	OFF	
X	1	0	0	ON	ON	External XCIN clock input	
			1	ON	OFF		

X: 0 or 1

NOTES:

1. Pulled up by setting the PU10 bit in the PUR1 register to 1.
2. Refer to 8.6.1 Port P4\_3, P4\_4.

**Table 8.27 Port P4\_4/(XCOUT)**

Register	PD4	CM0	CM1		Circuit specifications		Function
Bit	PD4_4	CM04	CM10	CM12	Oscillation buffer	Feedback resistor	
Setting value	1	0	X	X	OFF	OFF	Output port <sup>(1)</sup>
	X	1	0	0	ON	ON	XCIN clock oscillation (on-chip feedback resistor enabled)
	X	1	0	1	ON	OFF	XCIN clock oscillation (on-chip feedback resistor disabled)
	X	1	1	0	OFF	ON	XCIN clock oscillation stop
				1	OFF	OFF	
	X	1	0	0	ON	ON	External XCOUT clock output (inverted output of XCIN clock)
			1	ON	OFF		

X: 0 or 1

NOTE:

1. Refer to 8.6.1 Port P4\_3, P4\_4.

**Table 8.28 Port P4\_5/ $\overline{\text{INT0}}$** 

Register	PD4	INTEN	Function
Bit	PD4_5	INT0EN	
Setting value	0	0	Input port <sup>(1)</sup>
	1	0	Output port
	0	1	$\overline{\text{INT0}}$ input

NOTE:

1. Pulled up by setting the PU11 bit in the PUR1 register to 1.

**Table 8.29 Port P6\_0/TREO**

Register	PD6	PINSR4	PINSR3	TRECR1	Function
Bit	PD6_0	TREOSEL2	TREOSEL	TOENA	
Setting value	0	Other than 001b			Input port <sup>(1)</sup>
	1	Other than 001b			Output port
	X	0	0	1	TREO output

X: 0 or 1

NOTE:

1. Pulled up by setting the PU14 bit in the PUR1 register to 1.

**Table 8.30 Port P6\_3/TXD2**

Register	PD6	U2MR			Function	
Bit	PD6_3	SMD2	SMD1	SMD0		
Setting value	0	0	0	0	Input port <sup>(1)</sup>	
	1	0	0	0	Output port	
	X	1	0	0	1	TXD2 output <sup>(2)</sup>
			0		0	
			1	1		
			0	0		

X: 0 or 1

NOTES:

1. Pulled up by setting the PU14 bit in the PUR1 register to 1.
2. N-channel open-drain output by setting the NCH bit in the U2C0 register to 1.

**Table 8.31 Port P6\_4/RXD2**

Register	PD6	Function
Bit	PD6_4	
Setting value	0	Input port <sup>(1)</sup>
	1	Output port
	0	RXD2 input <sup>(1)</sup>

NOTE:

1. Pulled up by setting the PU15 bit in the PUR1 register to 1.

**Table 8.32 Port P6\_5/CLK2/(TREO)**

Register	PD6	PINSR4	TRECR1	U2MR				Function
Bit	PD6_5	TREOSEL2	TOENA	CKDIR	SMD2	SMD1	SMD0	
Setting value	0	0	X	0	Other than 001b			Input port <sup>(1)</sup>
				1	X	X	X	
	1	0	X	X	Other than 001b			Output port
	X	0	X	0	0	0	1	CLK2 output
	0	0	X	1	X	X	X	CLK2 input <sup>(1)</sup>
X	1	1	X	X	X	X	X	TREO output

X: 0 or 1

NOTE:

1. Pulled up by setting the PU15 bit in the PUR1 register to 1.

**Table 8.33 Port P6\_6/(KI1)**

Register	PD6	PINSR4	INTEN	Function
Bit	PD6_6	KI1SEL	INT0EN	
Setting value	0	X	0	Input port <sup>(1)</sup>
	1	X	0	Output port
	0	1	1	KI1 input <sup>(1)</sup>

X: 0 or 1

NOTE:

1. Pulled up by setting the PU15 bit in the PUR1 register to 1.

## 8.5 Unassigned Pin Handling

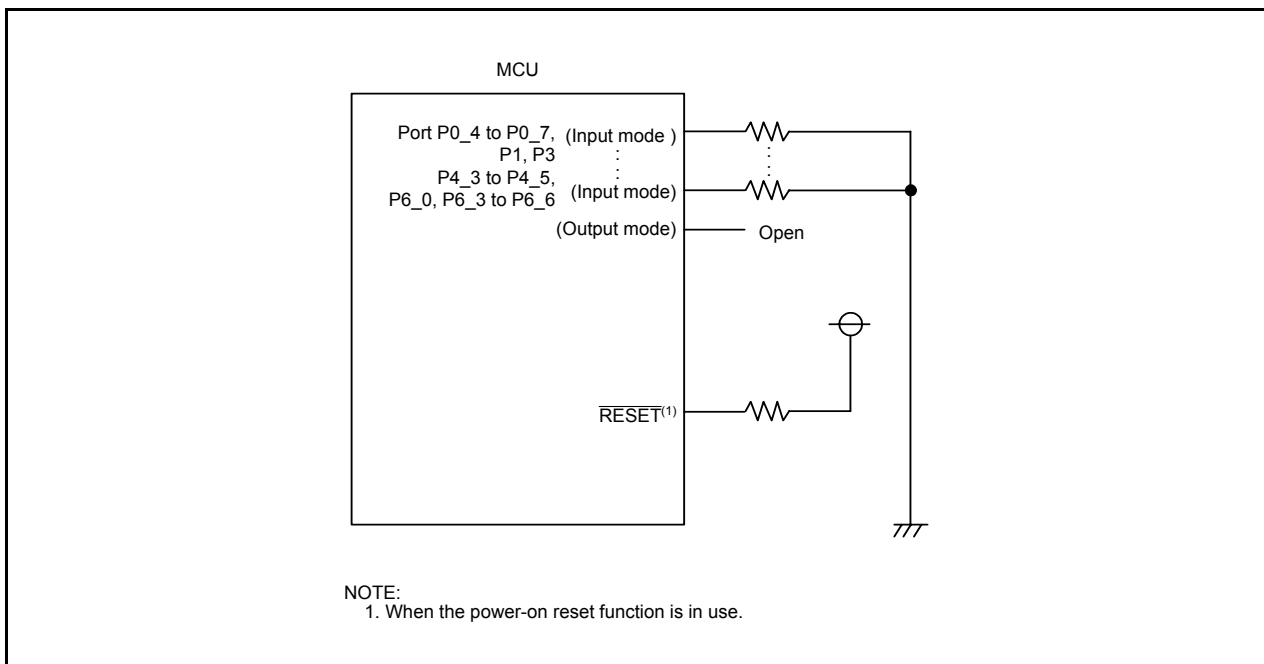
Table 8.34 lists Unassigned Pin Handling.

**Table 8.34 Unassigned Pin Handling**

Pin Name	Connection
Ports P0_4 to P0_7, P1, P3, P4_3 to P4_5, P6_0, P6_3 to P6_6	<ul style="list-style-type: none"> <li>• After setting to input mode, connect each pin to VSS via a resistor (pull-down) or connect each pin to VCC via a resistor (pull-up).(2)</li> <li>• After setting to output mode, leave these pins open.(1, 2)</li> </ul>
RESET <sup>(3)</sup>	Connect to VCC via a pull-up resistor(2)

### NOTES:

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



**Figure 8.10 Unassigned Pin Handling**

## 8.6 Notes on I/O Ports

### 8.6.1 Port P4\_3, P4\_4

Ports P4\_3 and P4\_4 are also used as the XCIN function and the XCOU function, respectively. During a reset period and after a reset release, these ports are set to the XCIN and XCOU functions. Pins P4\_3 and P4\_4 can be switched to the port functions by setting the CM04 bit in the CM0 register to 0 (ports P4\_3 and P4\_4) by a program.

To use ports P4\_3 and P4\_4 as ports, note the following:

- Port P4\_3

After a reset until the CM04 bit is set to 0 (ports P4\_3 and P4\_4) by a program, a typical 10 MΩ impedance is connected between the P4\_3 pin and the MCU power supply or GND. If the XCIN is set to intermediate-level input or left floating, a shoot-through current flows into the oscillation driver.

- Port P4\_4

Use port P4\_4 as an output port by setting the PD4\_4 bit in the PD4 register to 1 (output mode). After a reset until the CM04 bit is set to 0 (ports P4\_3 and P4\_4) by a program, the P4\_4 pin may output an intermediate potential of about 2.0 V.



## 9. Processor Mode

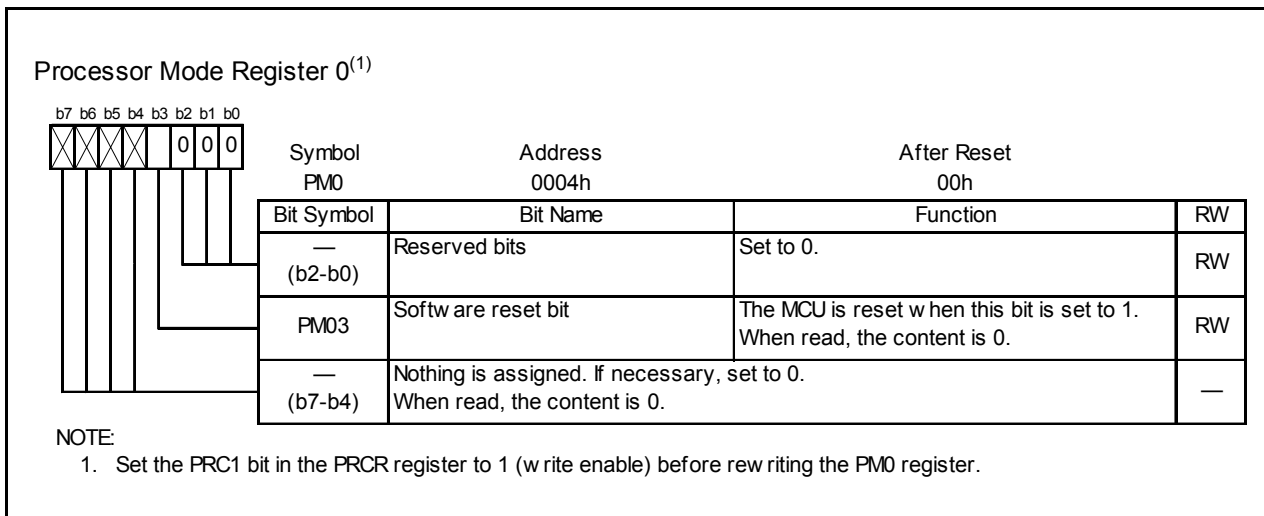
### 9.1 Processor Modes

Single-chip mode can be selected as the processor mode.

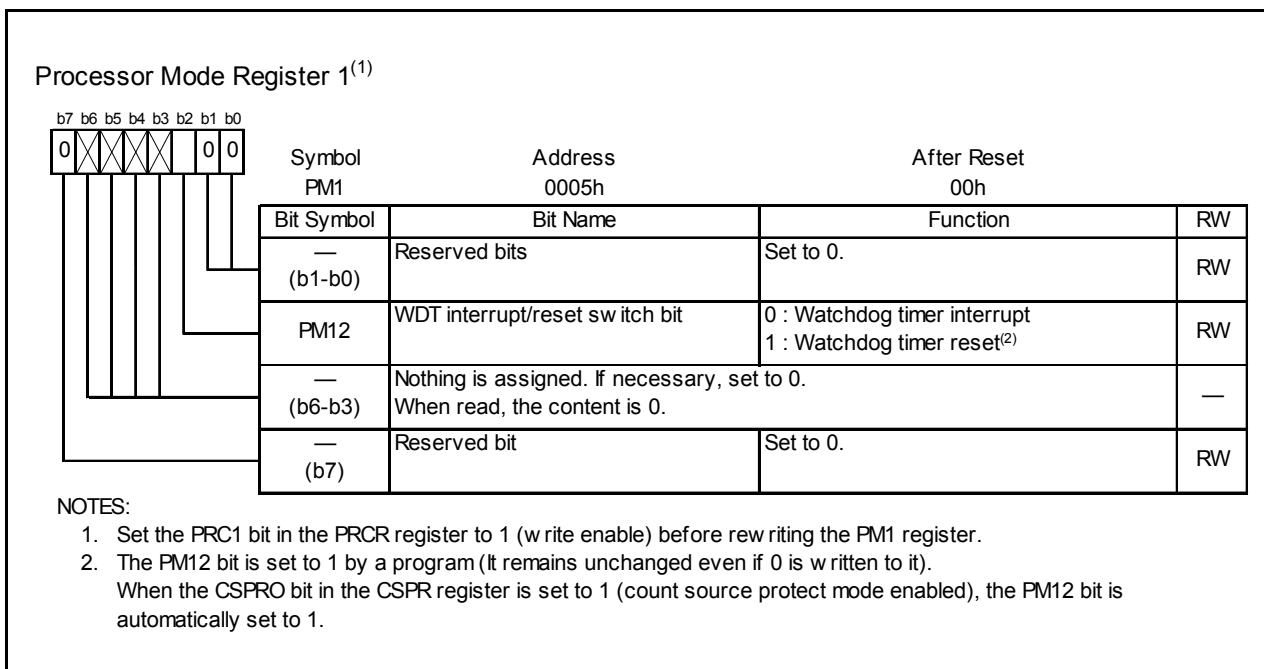
Table 9.1 lists Features of Processor Mode. Figure 9.1 shows the PM0 Register and Figure 9.2 shows the PM1 Register.

**Table 9.1 Features of Processor Mode**

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



**Figure 9.1 PM0 Register**



**Figure 9.2 PM1 Register**

## 10. Bus

The bus cycles differ when accessing ROM/RAM, and when accessing SFR.

Table 10.1 lists Bus Cycles by Access Space of the R8C/2G Group.

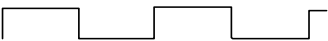
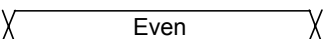
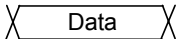

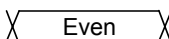


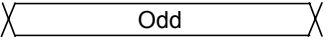
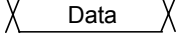

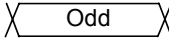


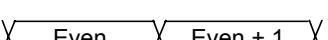


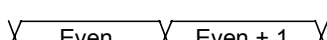


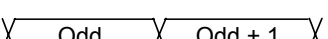

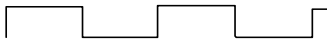
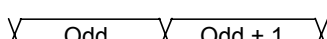

ROM/RAM and SFR are connected to the CPU by an 8-bit bus. When accessing in word (16-bit) units, these areas are accessed twice in 8-bit units.

Table 10.2 lists Access Units and Bus Operations.

**Table 10.1 Bus Cycles by Access Space of the R8C/2G Group**

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

**Table 10.2 Access Units and Bus Operations**

Area	SFR	ROM, RAM
Even address Byte access	CPU clock  Address  Data 	CPU clock  Address  Data 
Odd address Byte access	CPU clock  Address  Data 	CPU clock  Address  Data 
Even address Word access	CPU clock  Address  Data 	CPU clock  Address  Data 
Odd address Word access	CPU clock  Address  Data 	CPU clock  Address  Data 

## 11. Clock Generation Circuit

The clock generation circuit has:

- XCIN clock oscillation circuit
- Low-speed on-chip oscillator
- High-speed on-chip oscillator

Table 11.1 lists Specifications of Clock Generation Circuit. Figure 11.1 shows a Clock Generation Circuit. Figures 11.2 to 11.8 show clock associated registers. Figure 11.9 shows a Handling Procedure of Internal Power Low Consumption Using VCA20 Bit.

**Table 11.1 Specifications of Clock Generation Circuit**

Item	XCIN Clock Oscillation Circuit	On-Chip Oscillator	
		High-Speed On-Chip Oscillator	Low-Speed On-Chip Oscillator
Applications	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> </ul>
Clock frequency	32.768 kHz	Approx. 8 MHz	Approx. 125 kHz
Connectable oscillator	<ul style="list-style-type: none"> <li>• Crystal oscillator</li> </ul>	–	–
Oscillator connect pins	XCIN, XCOUT <sup>(1)</sup>	– <sup>(1)</sup>	– <sup>(1)</sup>
Oscillation stop, restart function	Usable	Usable	Usable
Oscillator status after reset	Oscillate	Stop	Oscillate
Others	<ul style="list-style-type: none"> <li>• Externally generated clock can be input<sup>(2)</sup></li> <li>• On-chip feedback resistor RfXCIN (connected/ not connected, selectable)</li> </ul>	–	–

NOTES:

1. These pins can be used as P4\_3 or P4\_4 when using the on-chip oscillator clock as the CPU clock while the XCIN clock oscillation circuit is not used.
2. Set the CM04 bit in the CM0 register to 1 (XCIN-XCOUT pin) when an external clock is input.

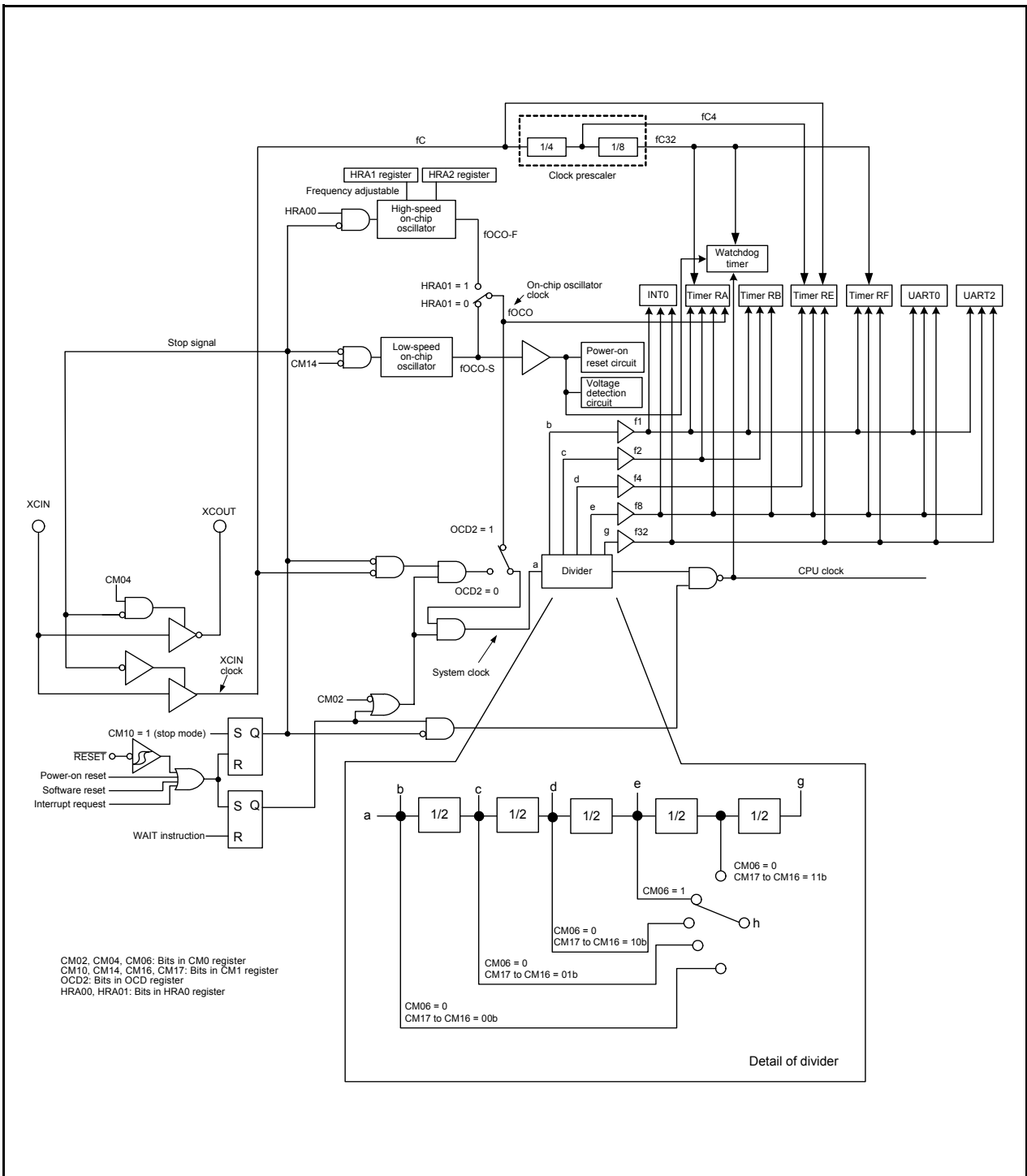


Figure 11.1 Clock Generation Circuit

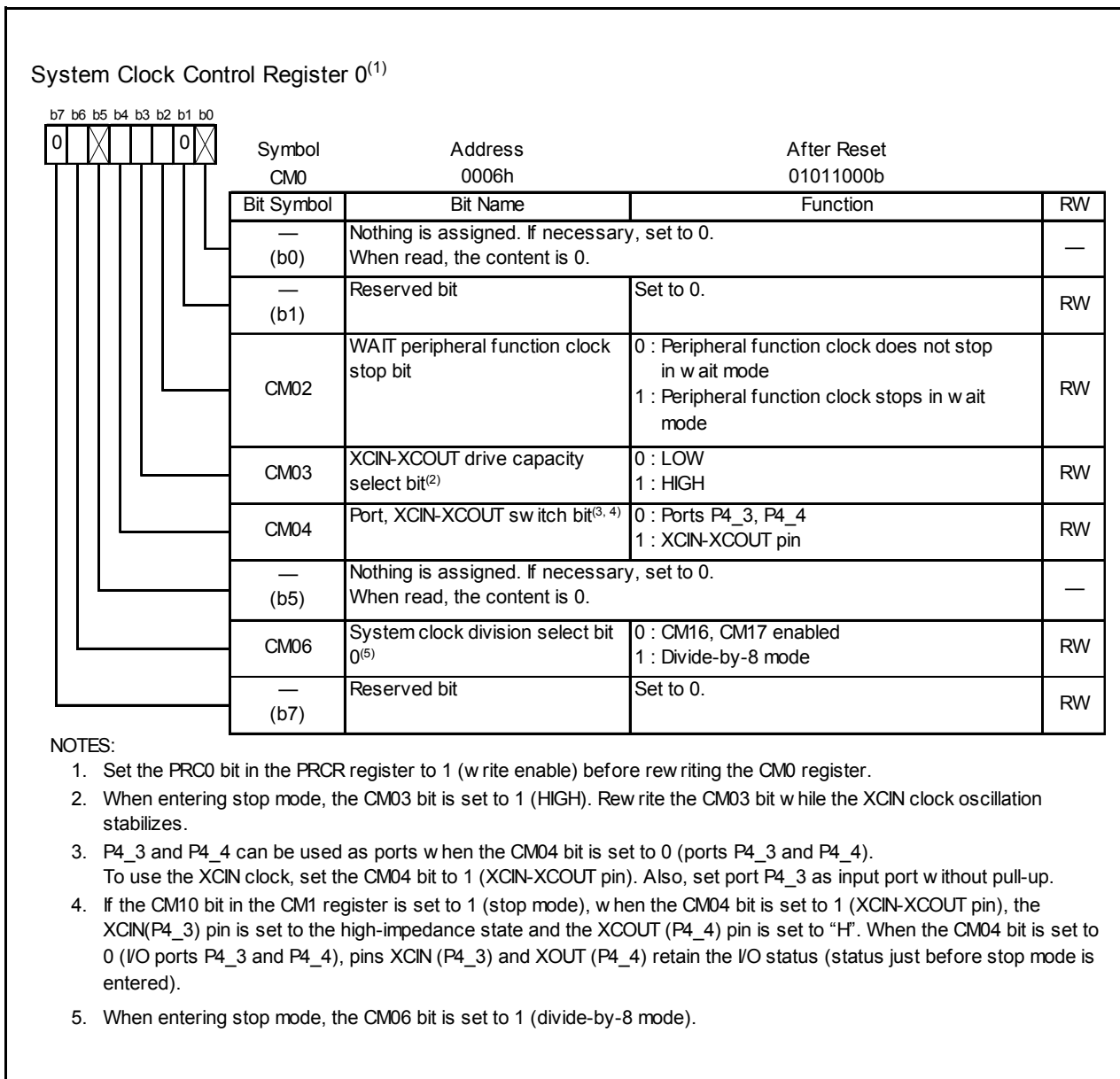


Figure 11.2 CM0 Register

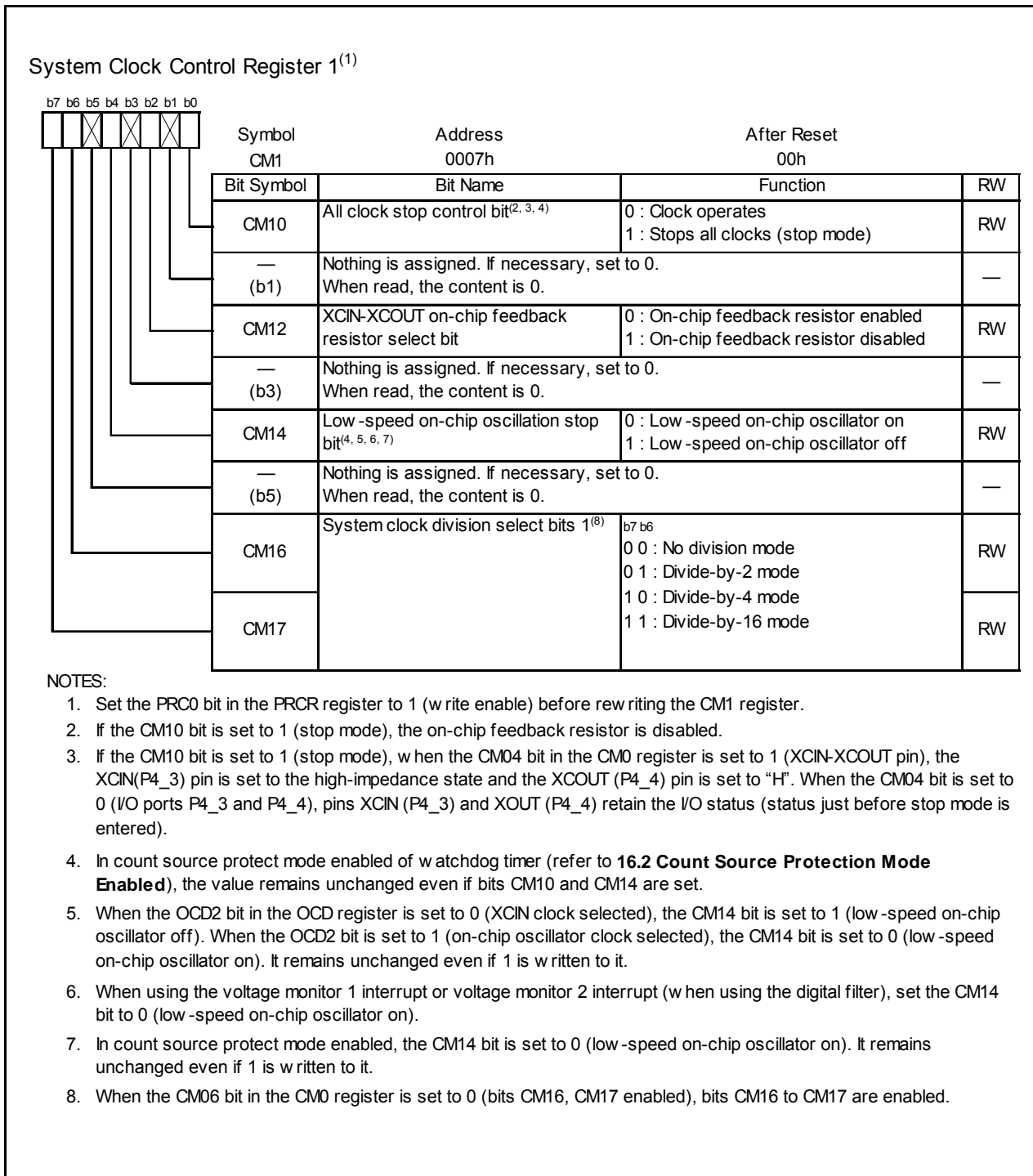
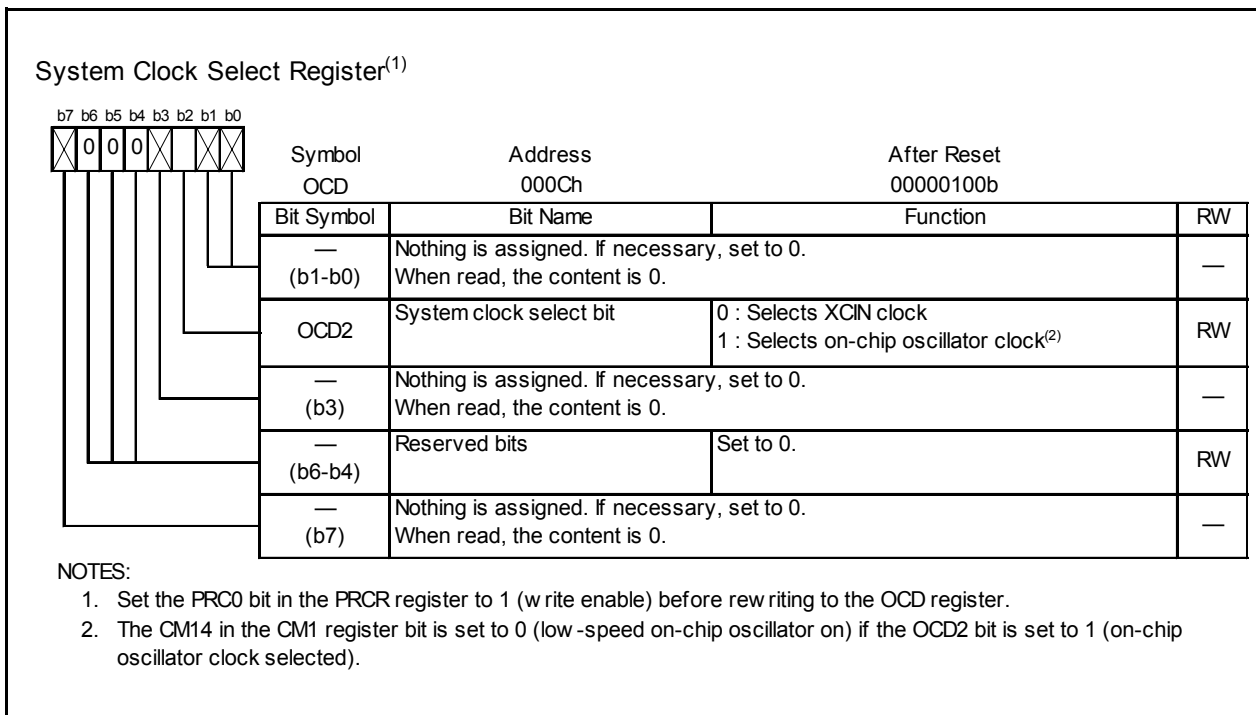


Figure 11.3 CM1 Register



**Figure 11.4** OCD Register

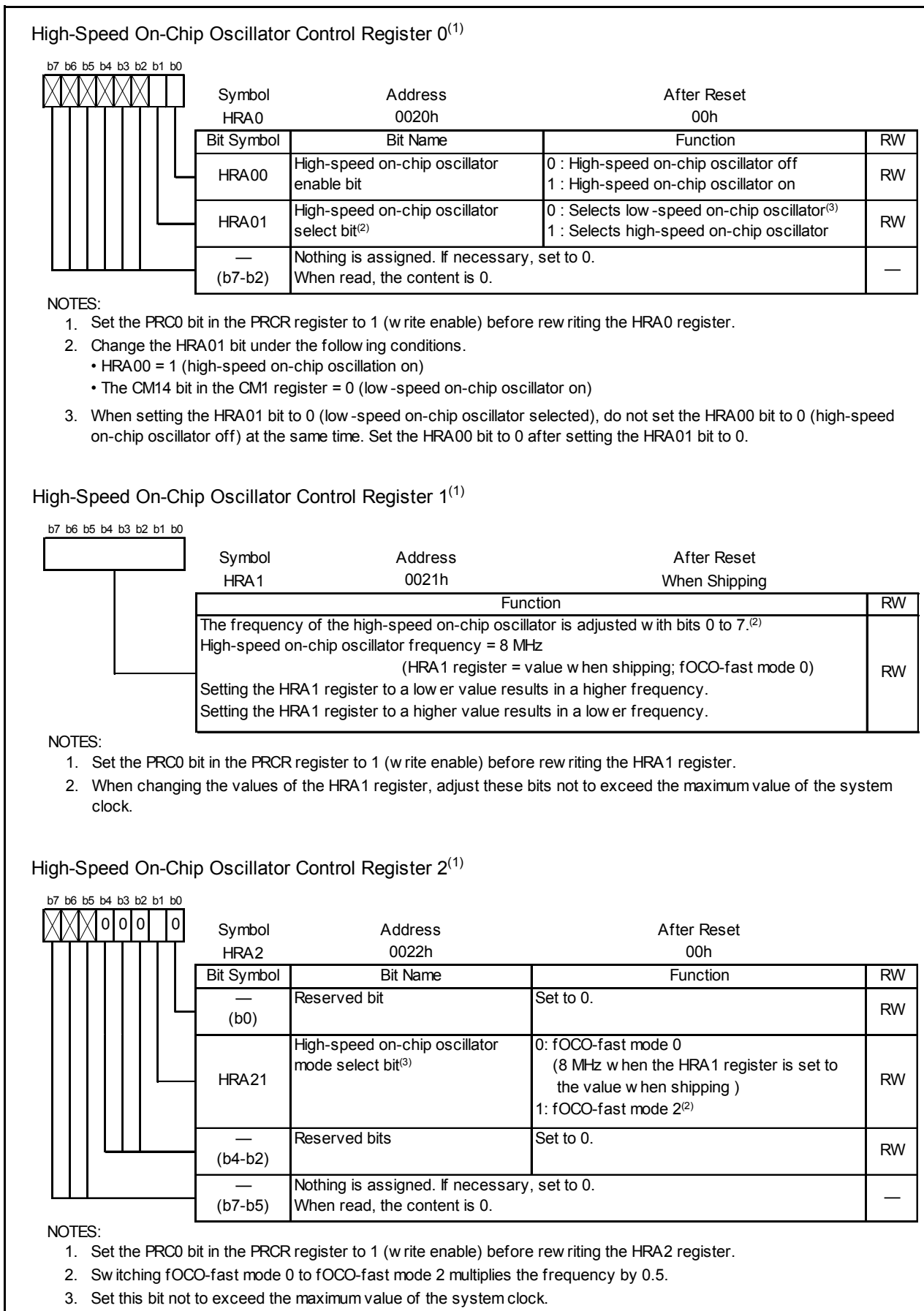
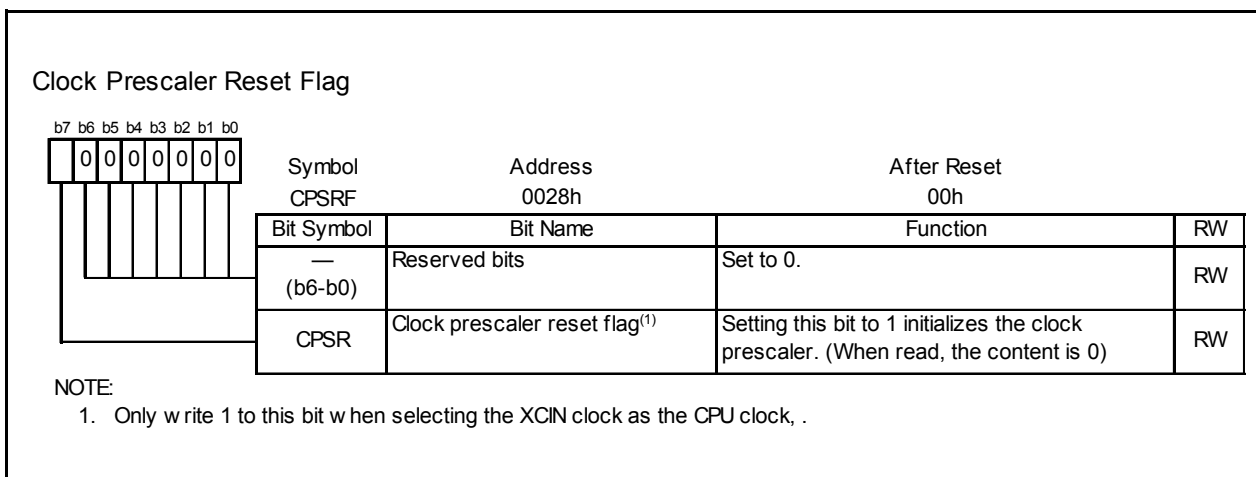
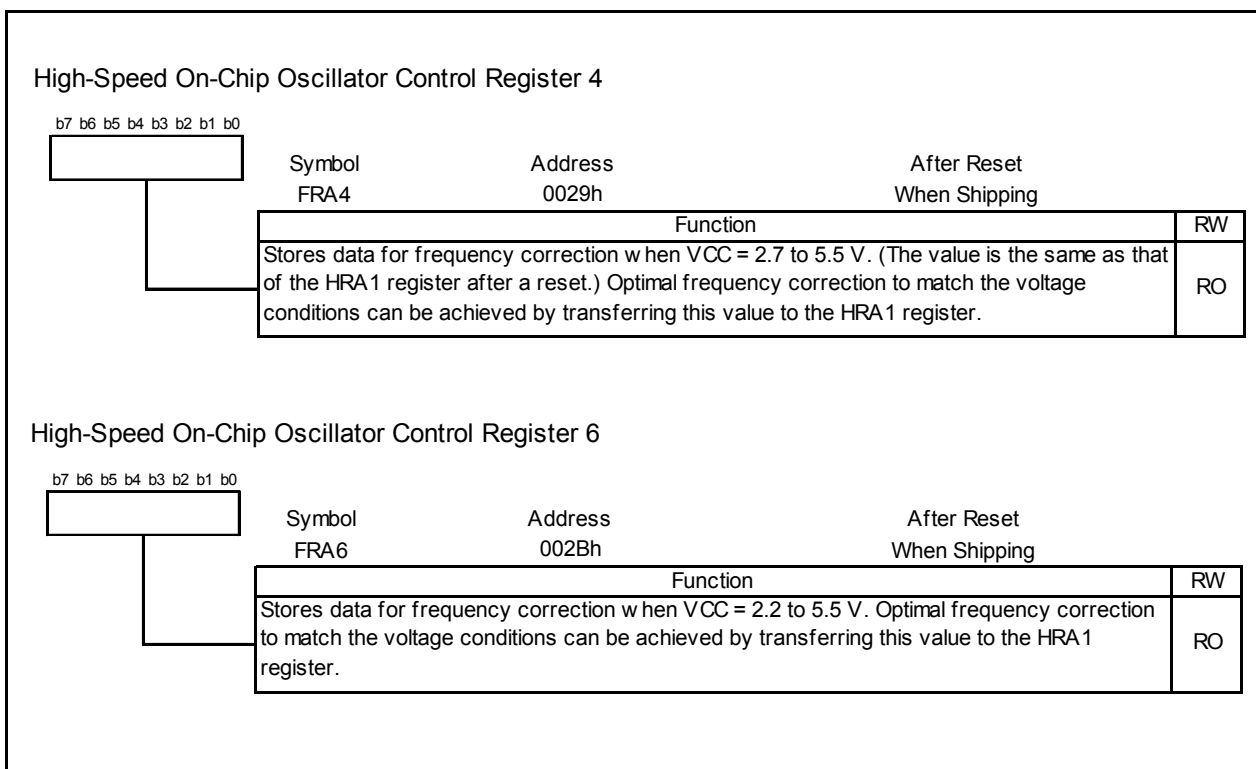


Figure 11.5 Registers HRA0, HRA1, and HRA2





**Figure 11.6 CPSRF Register**



**Figure 11.7 Registers FRA4 and FRA6**

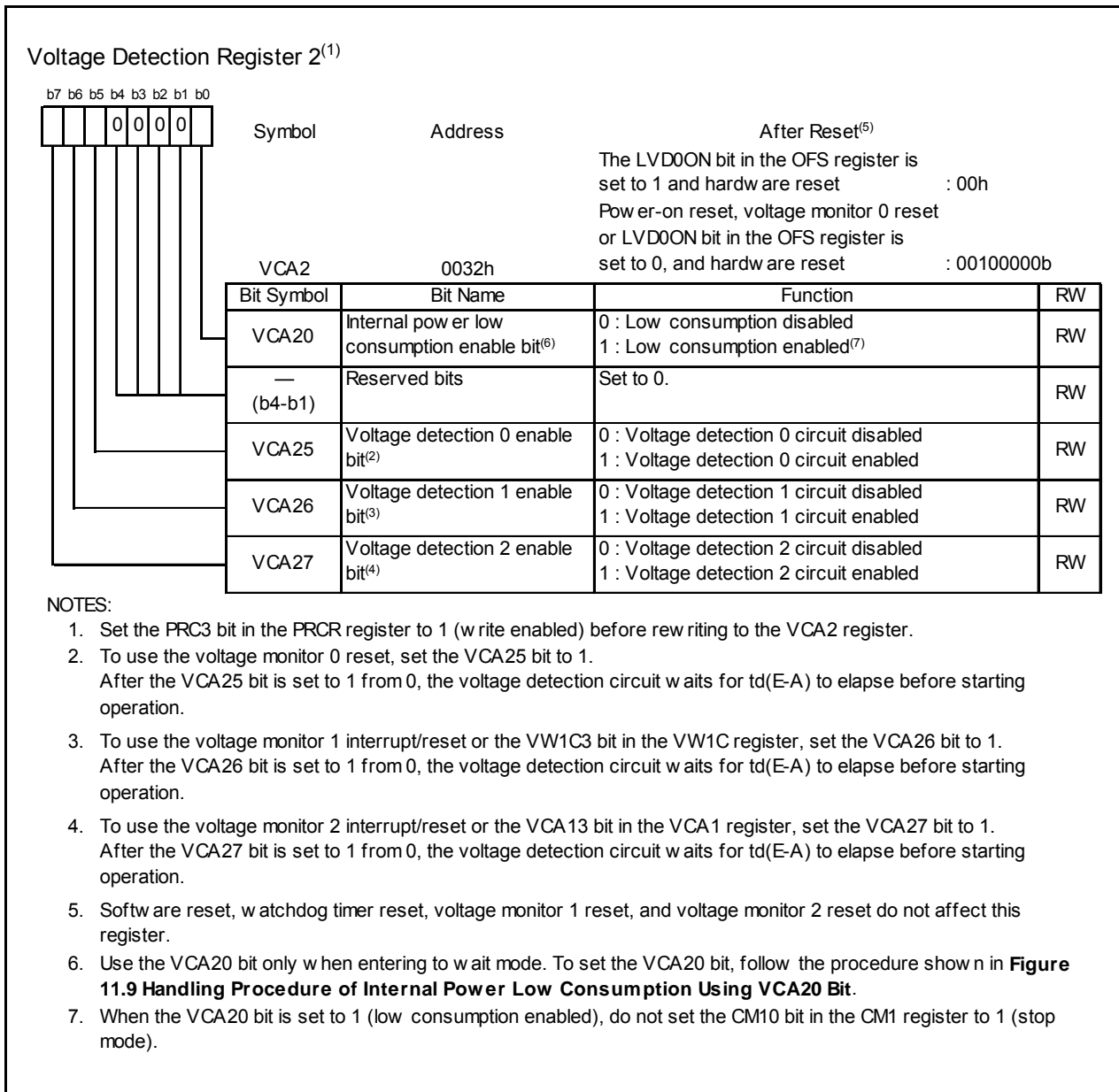
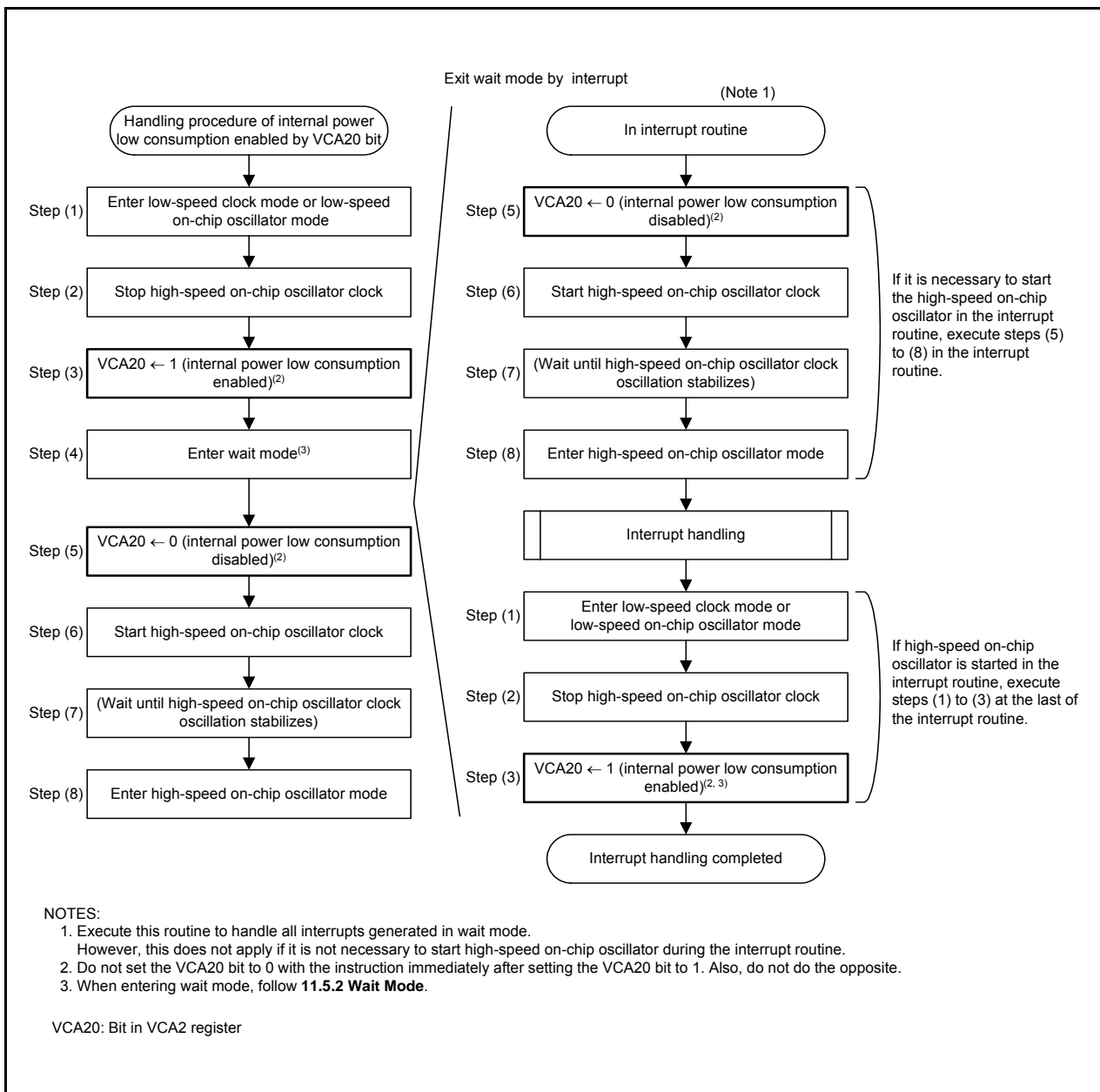


Figure 11.8 VCA2 Register



**Figure 11.9 Handling Procedure of Internal Power Low Consumption Using VCA20 Bit**

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

## 11.1 On-Chip Oscillator Clocks

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

### 11.1.1 Low-Speed On-Chip Oscillator Clock

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

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

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

### 11.1.2 High-Speed On-Chip Oscillator Clock

The clock generated by the high-speed on-chip oscillator is used as the clock source for the CPU clock, peripheral function clock, fOCO, and fOCO-F.

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

Furthermore, frequency correction data corresponding to the supply voltage ranges listed below is stored in registers FRA4 and FRA6. To use separate correction values to match these voltage ranges, transfer them from register FRA4 or FRA6 to the HRA1 register.

- FRA4 register: Stores data for frequency correction corresponding to  $VCC = 2.7\text{ V}$  to  $5.5\text{ V}$ .  
(The value is the same as that of the HRA1 register after a reset.)
- FRA6 register: Stores data for frequency correction corresponding to  $VCC = 2.2\text{ V}$  to  $5.5\text{ V}$ .

Since there are differences in the amount of frequency adjustment among the bits in the HRA1 register, make adjustments by changing the settings of individual bits. Adjust the HRA1 register so that the frequency of the high-speed on-chip oscillator clock does not exceed the maximum value of the system clock.

## 11.2 XCIN Clock

This clock is supplied by the XCIN clock oscillation circuit. This clock is used as the clock source for the CPU clock, peripheral function clock. The XCIN clock oscillation circuit is configured by connecting a resonator between the XCIN and XCOOUT pins. The XCIN clock oscillation circuit includes an on-chip a feedback resistor, which is disconnected from the oscillation circuit in stop mode in order to reduce the amount of power consumed in the chip. The XCIN clock oscillation circuit may also be configured by feeding an externally generated clock to the XCIN pin.

Figure 11.10 shows Examples of XCIN Clock Connection Circuits.

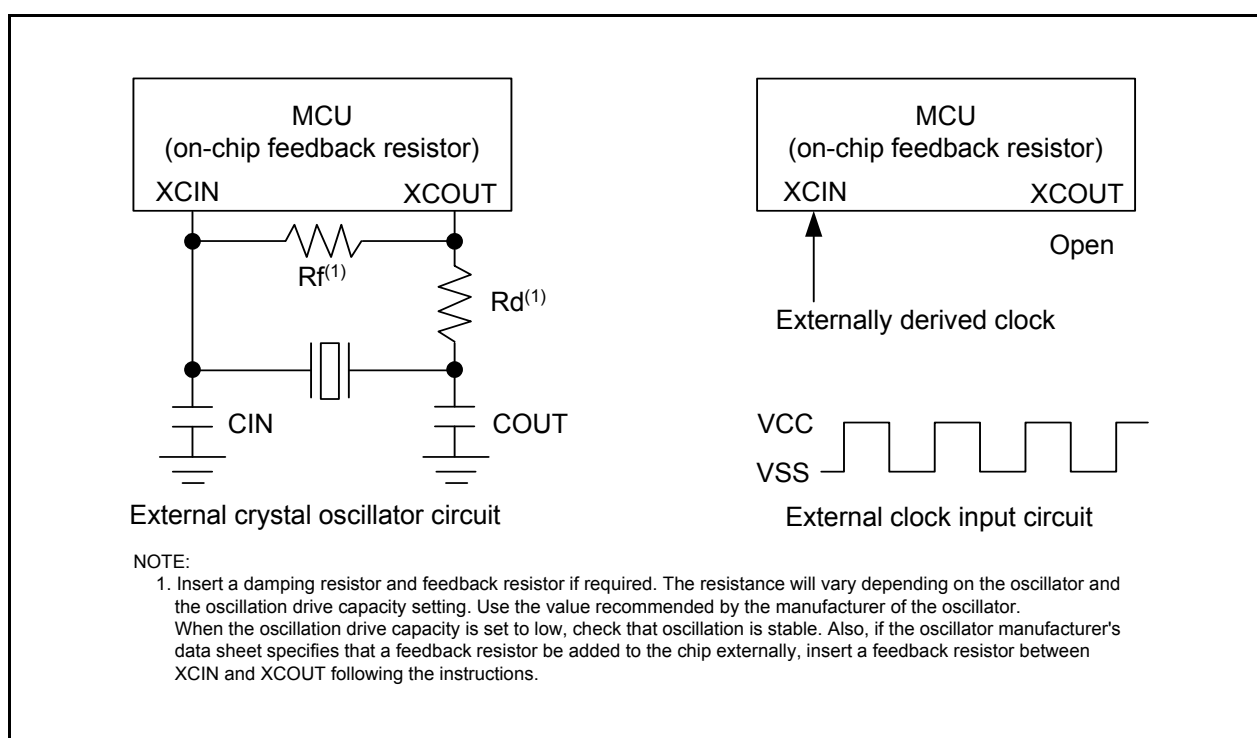
During and after reset, the XCIN clock oscillates.

The XCIN clock starts oscillating when the CM04 bit in the CM0 register is set to 1 (XCIN-XCOOUT pin).

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

This MCU has an on-chip feedback resistor and on-chip resistor disable/enable switching is possible by the CM12 bit in the CM1 register.

In stop mode, all clocks including the XCIN clock stop. Refer to **11.4 Power Control** for details.



**Figure 11.10** Examples of XCIN Clock Connection Circuits

## 11.3 CPU Clock and Peripheral Function Clock

There are a CPU clock to operate the CPU and a peripheral function clock to operate the peripheral functions. Refer to **Figure 11.1 Clock Generation Circuit**.

### 11.3.1 System Clock

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

### 11.3.2 CPU Clock

The CPU clock is an operating clock for the CPU and watchdog timer.

The system clock can be divided by 1 (no division), 2, 4, 8, or 16 to produce the CPU clock. Use the CM06 bit in the CM0 register and bits CM16 to CM17 in the CM1 register to select the value of the division.

Use the XCIN clock while the XCIN clock oscillation stabilizes.

After reset, the low-speed on-chip oscillator clock divided by 8 provides the CPU clock.

When entering stop mode from high-speed clock mode, the CM06 bit is set to 1 (divide-by-8 mode).

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

The peripheral function clock is the operating clock for the peripheral functions.

The clock  $f_i$  ( $i = 1, 2, 4, 8,$  and  $32$ ) is generated by the system clock divided by  $i$ . The clock  $f_i$  is used for timers RA, RB, RE, and RF, and the serial interface.

When the WAIT instruction is executed after setting the CM02 bit in the CM0 register to 1 (peripheral function clock stops in wait mode), the clock  $f_i$  stop.

### 11.3.4 fOCO

fOCO is an operating clock for the peripheral functions.

fOCO runs at the same frequency as the on-chip oscillator clock and can be used as the source for timer RA.

When the WAIT instruction is executed, the clocks fOCO does not stop.

### 11.3.5 fOCO-F

fOCO-F is generated by the high-speed on-chip oscillator and supplied by setting the HRA00 bit to 1.

When the WAIT instruction is executed, the clock fOCO-F does not stop.

### 11.3.6 fOCO-S

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

### 11.3.7 fC4 and fC32

The clock fC4 is used for timer RE and the clock fC32 is used for timer RA, timer RF, and watchdog timer.

Use fC4 and fC32 while the XCIN clock oscillation stabilizes.

## 11.4 Power Control

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

### 11.4.1 Standard Operating Mode

Standard operating mode is further separated into three modes.

In standard operating mode, the CPU clock and the peripheral function clock are supplied to operate the CPU and the peripheral function clocks. Power consumption control is enabled by controlling the CPU clock frequency. The higher the CPU clock frequency, the more processing power increases. The lower the CPU clock frequency, the more power consumption decreases. When unnecessary oscillator circuits stop, power consumption is further reduced.

Before the clock sources for the CPU clock can be switched over, the new clock source needs to be oscillating and stable. If the new clock source is the XCIN clock, allow sufficient wait time in a program until oscillation is stabilized before exiting.

**Table 11.2 Settings and Modes of Clock Associated Bits**

Modes		OCD Register	CM1 Register		CM0 Register		HRA0 Register	
		OCD2	CM17, CM16	CM14	CM06	CM04	HRA01	HRA00
High-speed on-chip oscillator mode	No division	1	00b	–	0	–	1	1
	Divide-by-2	1	01b	–	0	–	1	1
	Divide-by-4	1	10b	–	0	–	1	1
	Divide-by-8	1	–	–	1	–	1	1
	Divide-by-16	1	11b	–	0	–	1	1
Low-speed on-chip oscillator mode	No division	1	00b	0	0	–	0	–
	Divide-by-2	1	01b	0	0	–	0	–
	Divide-by-4	1	10b	0	0	–	0	–
	Divide-by-8	1	–	0	1	–	0	–
	Divide-by-16	1	11b	0	0	–	0	–
Low-speed clock mode	No division	0	00b	–	0	1	–	–
	Divide-by-2	0	01b	–	0	1	–	–
	Divide-by-4	0	10b	–	0	1	–	–
	Divide-by-8	0	–	–	1	1	–	–
	Divide-by-16	0	11b	–	0	1	–	–

–: Can be 0 or 1, no change in outcome

### 11.4.1.1 High-Speed On-Chip Oscillator Mode

The high-speed on-chip oscillator is used as the on-chip oscillator clock when the HRA00 bit in the HRA0 register is set to 1 (high-speed on-chip oscillator on) and the HRA01 bit in the HRA0 register is set to 1. The on-chip oscillator divided by 1 (no division), 2, 4, 8, or 16 provides the CPU clock. Set the CM06 bit to 1 (divide-by-8 mode) when transiting to high-speed clock mode.

When the CM14 bit is set to 0 (low-speed on-chip oscillator on), fOCO-S can be used as the watchdog timer and voltage detection circuit.

### 11.4.1.2 Low-Speed On-Chip Oscillator Mode

If the CM14 bit in the CM1 register is set to 0 (low-speed on-chip oscillator on) or the HRA01 bit in the HRA0 register is set to 0, the low-speed on-chip oscillator provides the on-chip oscillator clock.

The on-chip oscillator clock divided by 1 (no division), 2, 4, 8 or 16 provides the CPU clock. The on-chip oscillator clock is also the clock source for the peripheral function clocks.

When the CM14 bit is set to 0 (low-speed on-chip oscillator on), fOCO-S can be used as the watchdog timer and voltage detection circuit.

In this mode, stopping the high-speed on-chip oscillator, and setting the FMR47 bit in the FMR4 register to 1 (flash memory low consumption current read mode enabled) enables low consumption operation.

To enter wait mode from low-speed on-chip oscillator mode, setting the VCA20 bit in the VCA2 register to 1 (internal power low consumption enabled) enables lower consumption current in wait mode.

Refer to **21. Reducing Power Consumption** for how to reduce the power consumption.

### 11.4.1.3 Low-Speed Clock Mode

The XCIN clock divided by 1 (no division), 2, 4, 8, or 16 provides the CPU clock. Set the CM06 bit to 1 (divide-by-8 mode) when transiting to high-speed on-chip oscillator mode, low-speed on-chip oscillator mode. If the CM14 bit is set to 0 (low-speed on-chip oscillator on) or the HRA00 bit in the HRA0 register is set to 1 (high speed on-chip oscillator on), fOCO can be used as timer RA.

When the CM14 bit is set to 0 (low-speed on-chip oscillator on), fOCO-S can be used as the watchdog timer and voltage detection circuit.

In this mode, stopping the high-speed on-chip oscillator, and setting the FMR47 bit in the FMR4 register to 1 (flash memory low consumption current read mode enabled) enables low consumption operation.

To enter wait mode from low-speed clock mode, setting the VCA20 bit in the VCA2 register to 1 (internal power low consumption enabled) enables lower consumption current in wait mode.

Refer to **21. Reducing Power Consumption** for how to reduce the power consumption.



## 11.4.2 Wait Mode

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

### 11.4.2.1 Peripheral Function Clock Stop Function

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

### 11.4.2.2 Entering Wait Mode

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

### 11.4.2.3 Pin Status in Wait Mode

The I/O port is the status before wait mode was entered is maintained.

### 11.4.2.4 Exiting Wait Mode

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

The peripheral function interrupts are affected by the CM02 bit. When the CM02 bit is set to 0 (peripheral function clock does not stop in wait mode), all peripheral function interrupts can be used to exit wait mode. When the CM02 bit is set to 1 (peripheral function clock stops in wait mode), the peripheral functions using the peripheral function clock stop operating and the peripheral functions operated by external signals or on-chip oscillator clock can be used to exit wait mode.

Table 11.3 lists Interrupts to Exit Wait Mode and Usage Conditions.

**Table 11.3 Interrupts to Exit Wait Mode and Usage Conditions**

Interrupt	CM02 = 0	CM02 = 1
Serial interface interrupt	Usable when operating with internal or external clock	Usable when operating with external clock
Key input interrupt	Usable	Usable
Timer RA interrupt	Usable in all modes	Can be used if there is no filter in event counter mode. Usable by selecting fOCO or fC32 as count source.
Timer RB interrupt	Usable in all modes	(Do not use)
Timer RE interrupt	Usable in all modes	Usable when operating in real time clock mode
Timer RF interrupt	Usable in all modes	(Do not use)
$\overline{\text{INT0}}$ , $\overline{\text{INT1}}$ , $\overline{\text{INT2}}$ , $\overline{\text{INT4}}$ interrupt	Usable	Can be used if there is no filter
Voltage monitor 1 interrupt	Usable	Usable
Voltage monitor 2 interrupt	Usable	Usable

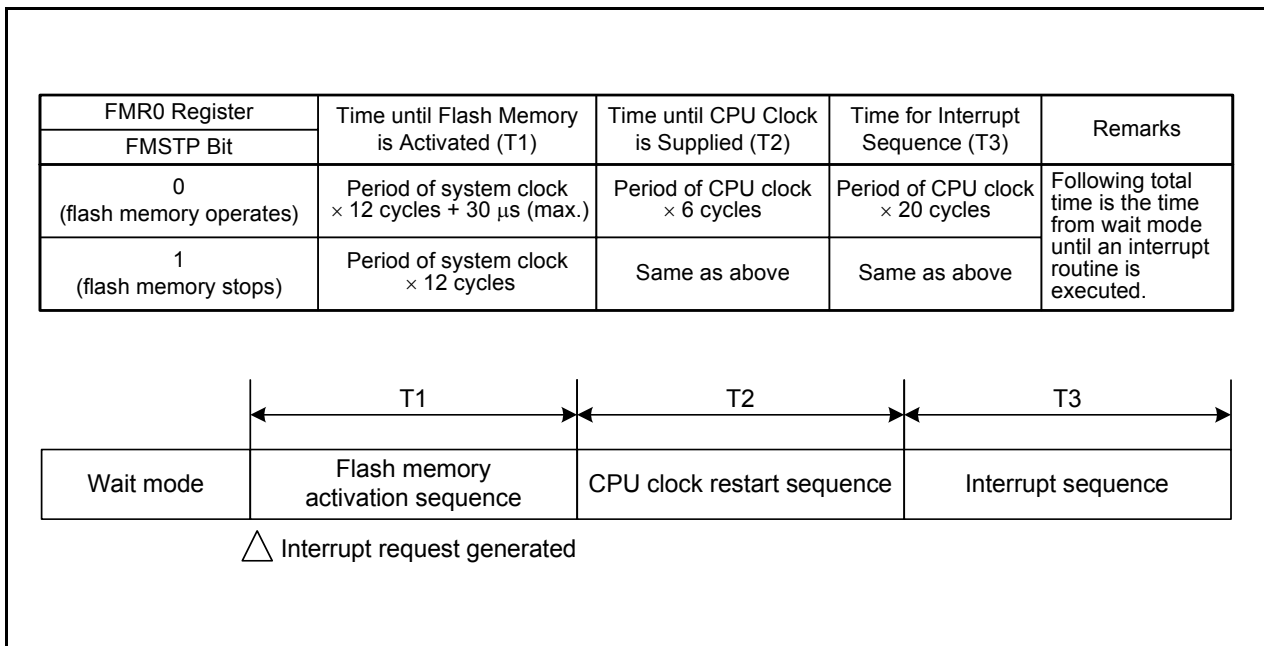
Figure 11.11 shows the Time from Wait Mode to Interrupt Routine Execution.

When using a peripheral function interrupt to exit wait mode, set up the following before executing the WAIT instruction.

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

When exiting by a peripheral function interrupt, the time (number of cycles) between interrupt request generation and interrupt routine execution is determined by the settings of the FMSTP bit in the FMR0 register, as described in Figure 11.11.

The CPU clock, when exiting wait mode by a peripheral function interrupt, is the same clock as the CPU clock when the WAIT instruction is executed.



**Figure 11.11 Time from Wait Mode to Interrupt Routine Execution**

### 11.4.3 Stop Mode

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

The peripheral functions clocked by external signals continue operating.

Table 11.4 lists Interrupts to Exit Stop Mode and Usage Conditions.

**Table 11.4 Interrupts to Exit Stop Mode and Usage Conditions**

Interrupt	Usage Conditions
Key input interrupt	–
$\overline{\text{INT0}}$ , $\overline{\text{INT1}}$ , $\overline{\text{INT2}}$ , $\overline{\text{INT4}}$ interrupt	Can be used if there is no filter
Timer RA interrupt	When there is no filter and external pulse is counted in event counter mode
Serial interface interrupt	When external clock is selected
Voltage monitor 1 interrupt	Usable in digital filter disabled mode (VW1C1 bit in VW1C register is set to 1)
Voltage monitor 2 interrupt	Usable in digital filter disabled mode (VW2C1 bit in VW2C register is set to 1)

#### 11.4.3.1 Entering Stop Mode

The MCU enters stop mode when the CM10 bit in the CM1 register is set to 1 (all clocks stop). At the same time, the CM06 bit in the CM0 register is set to 1 (divide-by-8 mode), the CM03 bit in the CM0 register is set to 1 (XCIN clock oscillator circuit drive capacity high).

#### 11.4.3.2 Pin Status in Stop Mode

The status before wait mode was entered is maintained.

When the CM04 bit in the CM0 register is set to 1 (XCIN-XCOUT pin), the XCIN(P4\_3) pin is set to the high-impedance state and the XCOUT (P4\_4) pin is set to “H”. When the CM04 bit is set to 0 (I/O ports P4\_3 and P4\_4), pins XCIN (P4\_3) and XOUT (P4\_4) retain the I/O status (status just before stop mode is entered).

### 11.4.3.3 Exiting Stop Mode

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

Figure 11.12 shows the Time from Stop Mode to Interrupt Routine Execution.

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

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

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

If the clock used immediately before stop mode is a system clock and stop mode is exited by a peripheral function interrupt, the CPU clock becomes the previous system clock divided by 8.

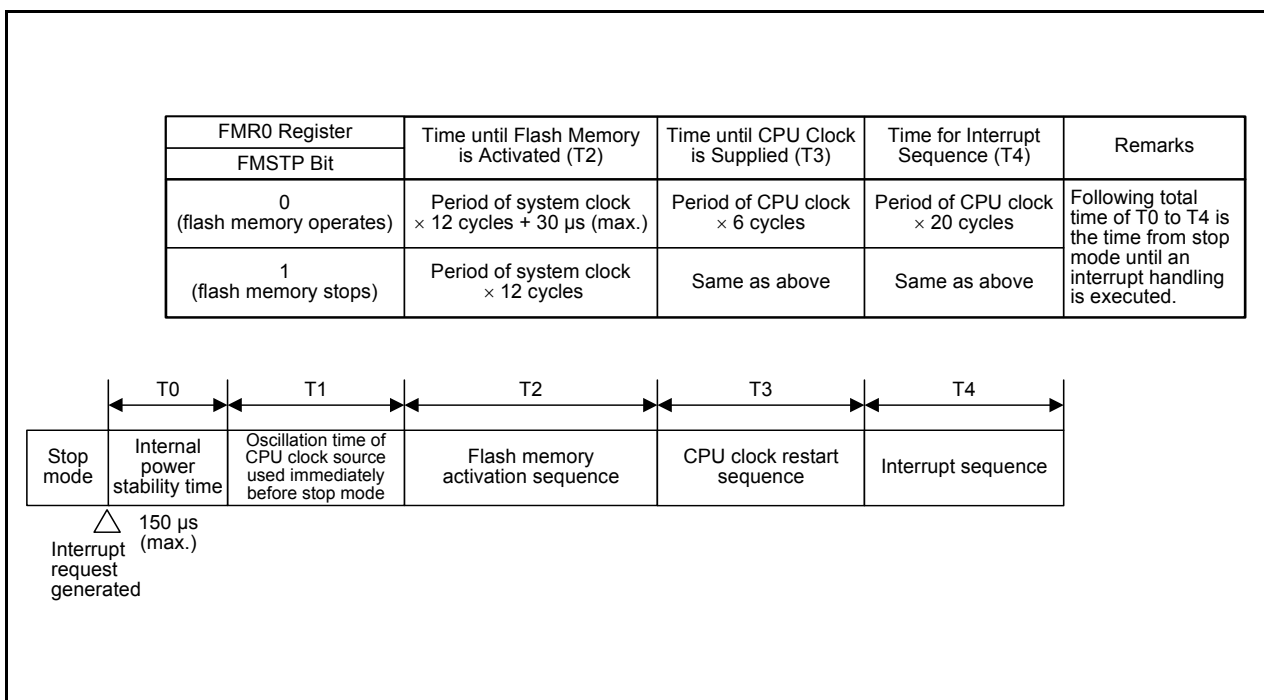


Figure 11.12 Time from Stop Mode to Interrupt Routine Execution

Figure 11.13 shows the State Transitions in Power Control Mode.

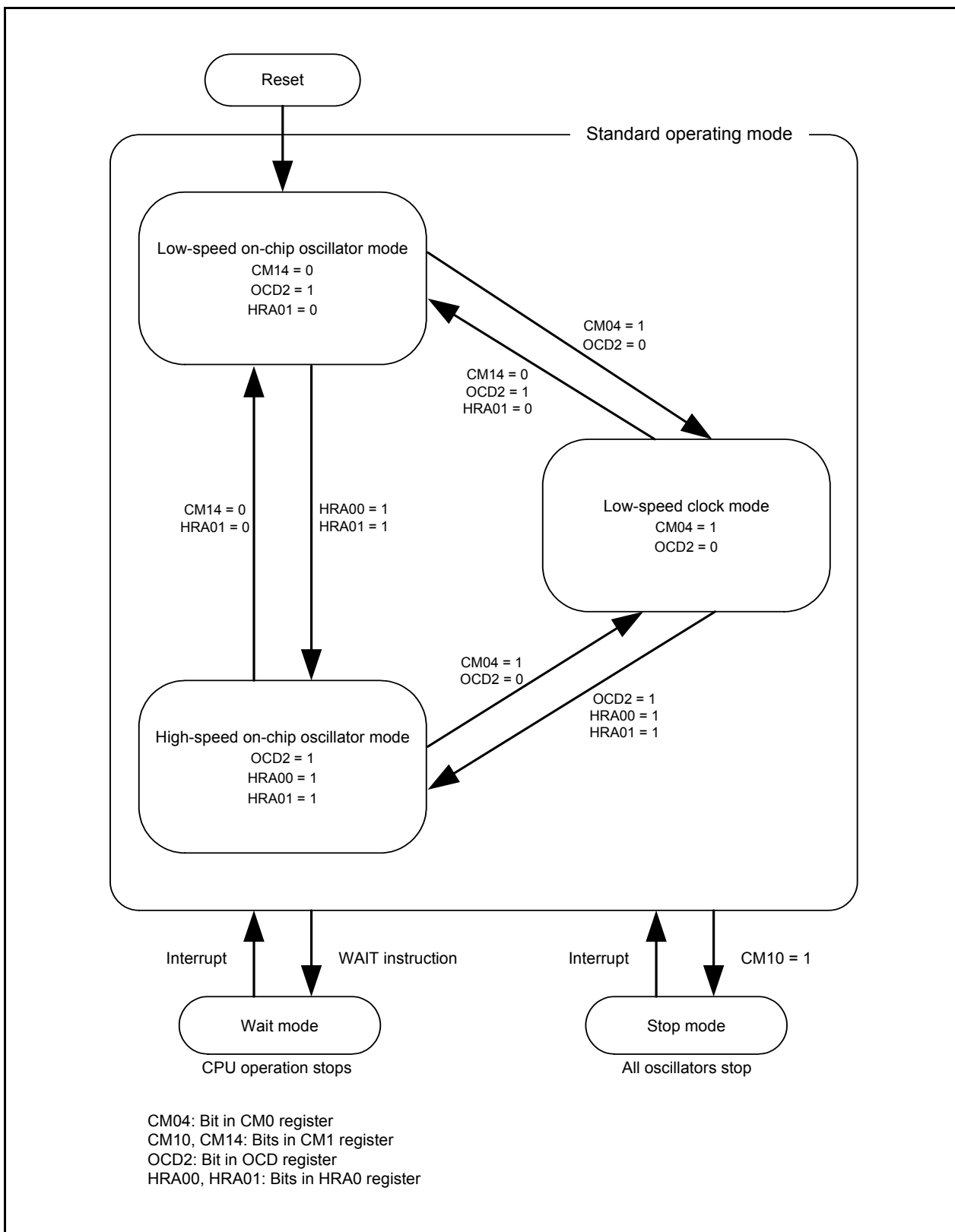


Figure 11.13 State Transitions in Power Control Mode

## 11.5 Notes on Clock Generation Circuit

### 11.5.1 Stop Mode

When entering stop mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) and the CM10 bit in the CM1 register to 1 (stop mode). An instruction queue pre-reads 4 bytes from the instruction which sets the CM10 bit to 1 (stop mode) and the program stops.

Insert at least 4 NOP instructions following the JMP.B instruction after the instruction which sets the CM10 bit to 1.

- Program example to enter stop mode

```

        BCLR    1,FMR0        ; CPU rewrite mode disabled
        BSET    0,PRCR       ; Protect disabled
        FSET    I            ; Enable interrupt
        BSET    0,CM1        ; Stop mode
        JMP.B   LABEL_001
LABEL_001 :
        NOP
        NOP
        NOP
        NOP

```

### 11.5.2 Wait Mode

When entering wait mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) and execute the WAIT instruction. An instruction queue pre-reads 4 bytes from the WAIT instruction and the program stops. Insert at least 4 NOP instructions after the WAIT instruction.

- Program example to execute the WAIT instruction

```

        BCLR    1,FMR0        ; CPU rewrite mode disabled
        FSET    I            ; Enable interrupt
        WAIT                    ; Wait mode
        NOP
        NOP
        NOP
        NOP

```

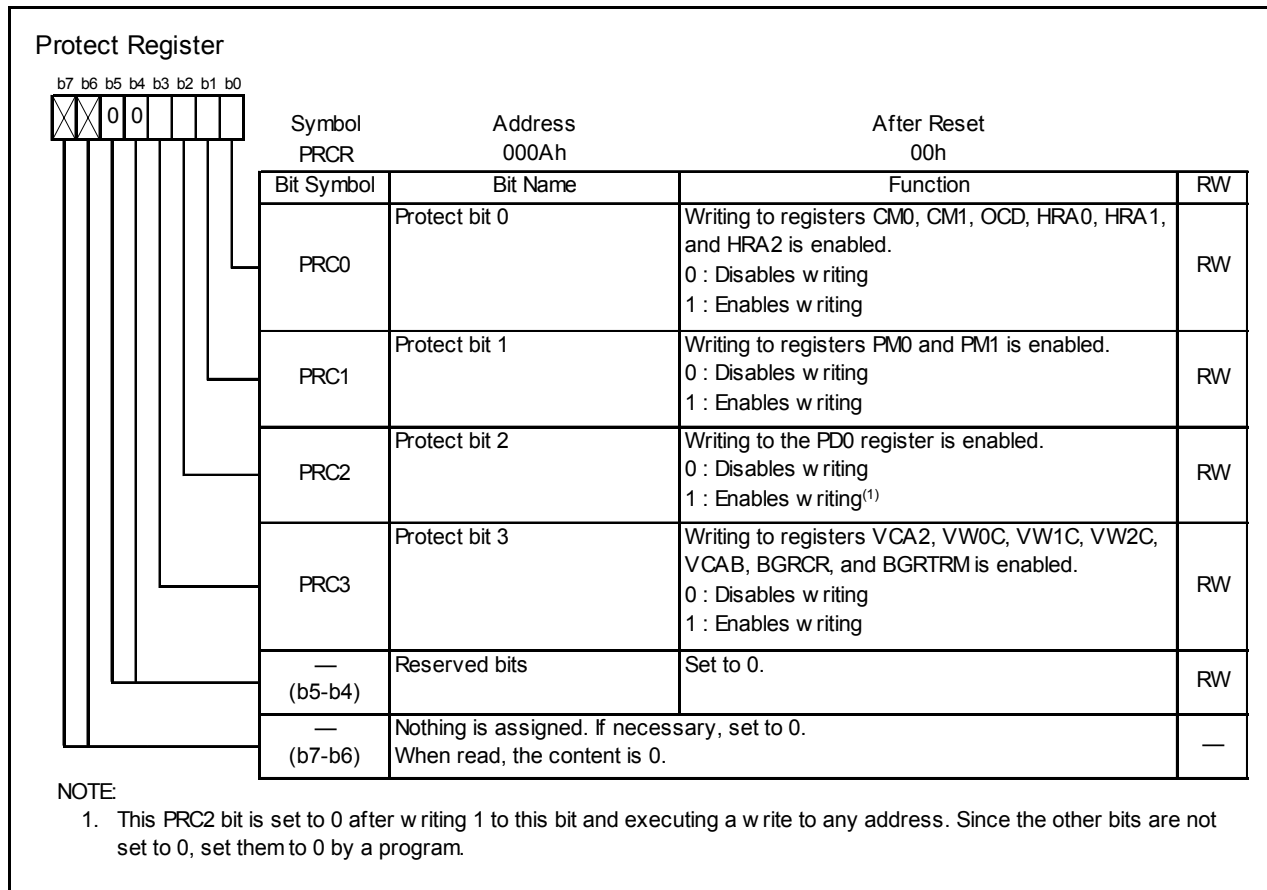
### 11.5.3 Oscillation Circuit Constants

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

## 12. Protection

The protection function protects important registers from being easily overwritten when a program runs out of control. Figure 12.1 shows the PRCR Register. The registers protected by the PRCR register are listed below.

- Registers protected by PRC0 bit: Registers CM0, CM1, OCD, HRA0, HRA1, and HRA2
- Registers protected by PRC1 bit: Registers PM0 and PM1
- Registers protected by PRC2 bit: PD0 register
- Registers protected by PRC3 bit: Registers VCA2, VW0C, VW1C, VW2C, VCAB, BGRCCR, and BGRTRM



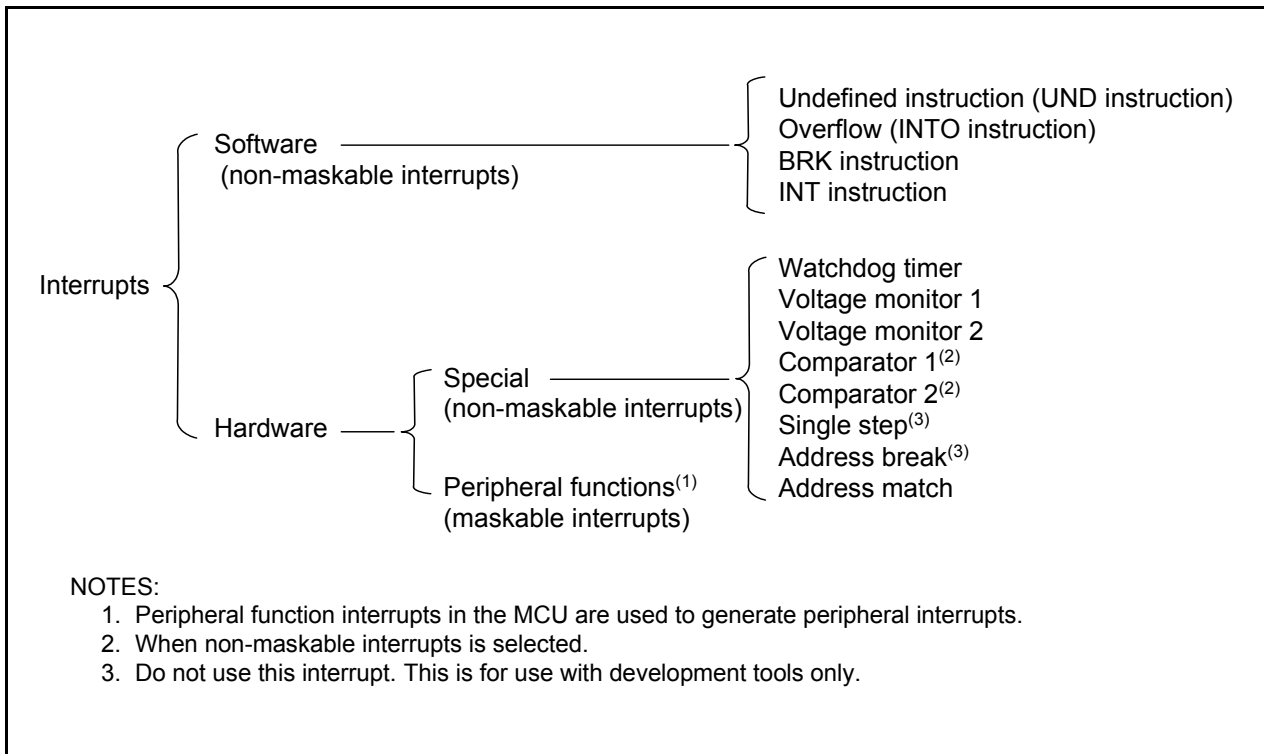
**Figure 12.1 PRCR Register**

## 13. Interrupts

### 13.1 Interrupt Overview

#### 13.1.1 Types of Interrupts

Figure 13.1 shows the Types of Interrupts.



**Figure 13.1** Types of Interrupts

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



## 13.1.2 Software Interrupts

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

### 13.1.2.1 Undefined Instruction Interrupt

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

### 13.1.2.2 Overflow Interrupt

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

### 13.1.2.3 BRK Interrupt

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

### 13.1.2.4 INT Instruction Interrupt

An INT instruction interrupt is generated when the INT instruction is executed. The INT instruction can select software interrupt numbers 0 to 63. Software interrupt numbers 3 to 31 are assigned to the peripheral function interrupt. Therefore, the MCU executes the same interrupt routine when the INT instruction is executed as when a peripheral function interrupt is generated. For software interrupt numbers 0 to 31, the U flag is saved to the stack during instruction execution and the U flag is set to 0 (ISP selected) before the interrupt sequence is executed. The U flag is restored from the stack when returning from the interrupt routine. For software interrupt numbers 32 to 63, the U flag does not change state during instruction execution, and the selected SP is used.

### 13.1.3 Special Interrupts

Special interrupts are non-maskable. However, the comparator 1 and comparator 2 can select maskable interrupts, too.

#### 13.1.3.1 Watchdog Timer Interrupt

The watchdog timer interrupt is generated by the watchdog timer. For details of the watchdog timer, refer to **16. Watchdog Timer**.

#### 13.1.3.2 Voltage Monitor 1 Interrupt

The voltage monitor 1 interrupt is generated by the voltage monitor 1 circuit. For details of the voltage monitor 1 circuit, refer to **6. Voltage Detection Circuit**.

#### 13.1.3.3 Voltage Monitor 2 Interrupt

The voltage monitor 2 interrupt is generated by the voltage monitor 2 circuit. For details of the voltage monitor 2, refer to **6. Voltage Detection Circuit**.

#### 13.1.3.4 Comparator 1 Interrupt

The comparator 1 interrupt is generated by the comparator 1. The non-maskable interrupt or maskable interrupt can be selected. For details of the comparator 1 interrupt, refer to **7. Comparator**.

#### 13.1.3.5 Comparator 2 Interrupt

The comparator 2 interrupt is generated by the comparator 2. The non-maskable interrupt or maskable interrupt can be selected. For details of the comparator 2 interrupt, refer to **7. Comparator**.

#### 13.1.3.6 Single-Step Interrupt, and Address Break Interrupt

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

#### 13.1.3.7 Address Match Interrupt

The address match interrupt is generated immediately before executing an instruction that is stored at an address indicated by registers RMAD0 to RMAD1 when the AIER0 or AIER1 bit in the AIER register is set to 1 (address match interrupt enable). For details of the address match interrupt, refer to **13.4 Address Match Interrupt**.

### 13.1.4 Peripheral Function Interrupt

The peripheral function interrupt is generated by the internal peripheral function of the MCU and is a maskable interrupt. Refer to **Table 13.2 Relocatable Vector Tables** for sources of the peripheral function interrupt. For details of peripheral functions, refer to the descriptions of individual peripheral functions.

### 13.1.5 Interrupts and Interrupt Vectors

There are 4 bytes in each vector. Set the starting address of an interrupt routine in each interrupt vector. When an interrupt request is acknowledged, the CPU branches to the address set in the corresponding interrupt vector. Figure 13.2 shows an Interrupt Vector.

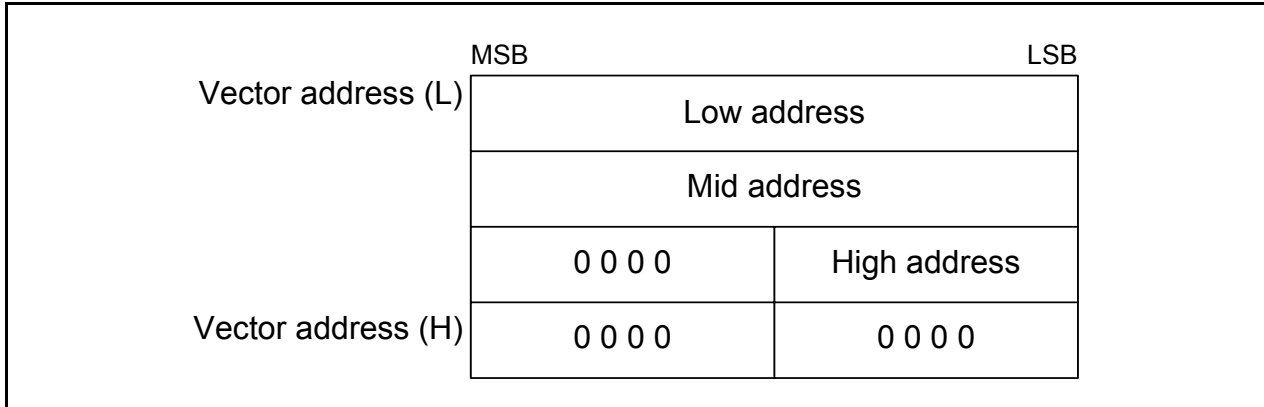


Figure 13.2 Interrupt Vector

#### 13.1.5.1 Fixed Vector Tables

The fixed vector tables are allocated addresses 0FFDCh to 0FFFFh.

Table 13.1 lists the Fixed Vector Tables. The vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to **20.3 Functions to Prevent Rewriting of Flash Memory**.

Table 13.1 Fixed Vector Tables

Interrupt Source	Vector Addresses Address (L) to (H)	Remarks	Reference
Undefined instruction	0FFDCh to 0FFDFh	Interrupt on UND instruction	R8C/Tiny Series Software Manual
Overflow	0FFE0h to 0FFE3h	Interrupt on INTO instruction	
BRK instruction	0FFE4h to 0FFE7h	If the content of address 0FFE7h is FFh, program execution starts from the address shown by the vector in the relocatable vector table.	
Address match	0FFE8h to 0FFEBh		13.4 Address Match Interrupt
Single step <sup>(1)</sup>	0FFEC h to 0FFEFh		
Watchdog timer, Voltage monitor 1, Voltage monitor 2, Comparator 1, Comparator 2	0FFF0h to 0FFF3h		16. Watchdog Timer 6. Voltage Detection Circuit 7. Comparator
Address break <sup>(1)</sup>	0FFF4h to 0FFF7h		
(Reserved)	0FFF8h to 0FFFBh		
Reset	0FFFCh to 0FFFFh		5. Resets

NOTE:

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

### 13.1.5.2 Relocatable Vector Tables

The relocatable vector tables occupy 256 bytes beginning from the starting address set in the INTB register. Table 13.2 lists the Relocatable Vector Tables.

**Table 13.2 Relocatable Vector Tables**

Interrupt Source	Vector Addresses <sup>(1)</sup> Address (L) to Address (H)	Software Interrupt Number	Interrupt Control Register	Reference
BRK instruction <sup>(2)</sup>	+0 to +3(0000h to 0003h)	0	–	R8C/Tiny Series Software Manual
Comparator 1	+4 to +7(0004h to 0007h)	1	VCMP1IC	7. Comparator
Comparator 2	+8 to +11(0008h to 000Bh)	2	VCMP2IC	
(Reserved)		3 to 9	–	–
Timer RE	+40 to +43(0028h to 002Bh)	10	TREIC	17.3 Timer RE
UART2 transmit	+44 to +47(002Ch to 002Fh)	11	S2TIC	18. Serial Interface
UART2 receive	+48 to +51(0030h to 0033h)	12	S2RIC	
Key input	+52 to +55(0034h to 0037h)	13	KUPIC	13.3 Key Input Interrupt
(Reserved)		14	–	–
(Reserved)		15	–	–
Compare 1	+64 to +67(0040h to 0043h)	16	CMP1IC	17.4 Timer RF
UART0 transmit	+68 to +71(0044h to 0047h)	17	S0TIC	18. Serial Interface
UART0 receive	+72 to +75(0048h to 004Bh)	18	S0RIC	
(Reserved)		19	–	–
(Reserved)		20	–	–
$\overline{\text{INT}}_2$	+84 to +87(0054h to 0057h)	21	INT2IC	13.2 $\overline{\text{INT}}$ Interrupt
Timer RA	+88 to +91(0058h to 005Bh)	22	TRAIC	17.1 Timer RA
(Reserved)		23	–	–
Timer RB	+96 to +99(0060h to 0063h)	24	TRBIC	17.2 Timer RB
$\overline{\text{INT}}_1$	+100 to +103(0064h to 0067h)	25	INT1IC	13.2 $\overline{\text{INT}}$ Interrupt
(Reserved)		26	–	–
Timer RF	+108 to +111(006Ch to 006Fh)	27	TRFIC	17.4 Timer RF
Compare 0	+112 to +115(0070h to 0073h)	28	CMP0IC	
$\overline{\text{INT}}_0$	+116 to +119(0074h to 0077h)	29	INT0IC	13.2 $\overline{\text{INT}}$ Interrupt
$\overline{\text{INT}}_4$	+120 to +123(0078h to 007Bh)	30	INT4IC	
Capture	+124 to +127(007Ch to 007Fh)	31	CAPIC	17.4 Timer RF
Software interrupt <sup>(2)</sup>	+128 to +131(0080h to 0083h) to +252 to +255(00FCh to 00FFh)	32 to 63	–	R8C/Tiny Series Software Manual

NOTES:

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

### 13.1.6 Interrupt Control

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

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

Figure 13.3 shows the Interrupt Control Register and Figure 13.4 shows the INTiIC Register (i=0, 1, 2, 4).

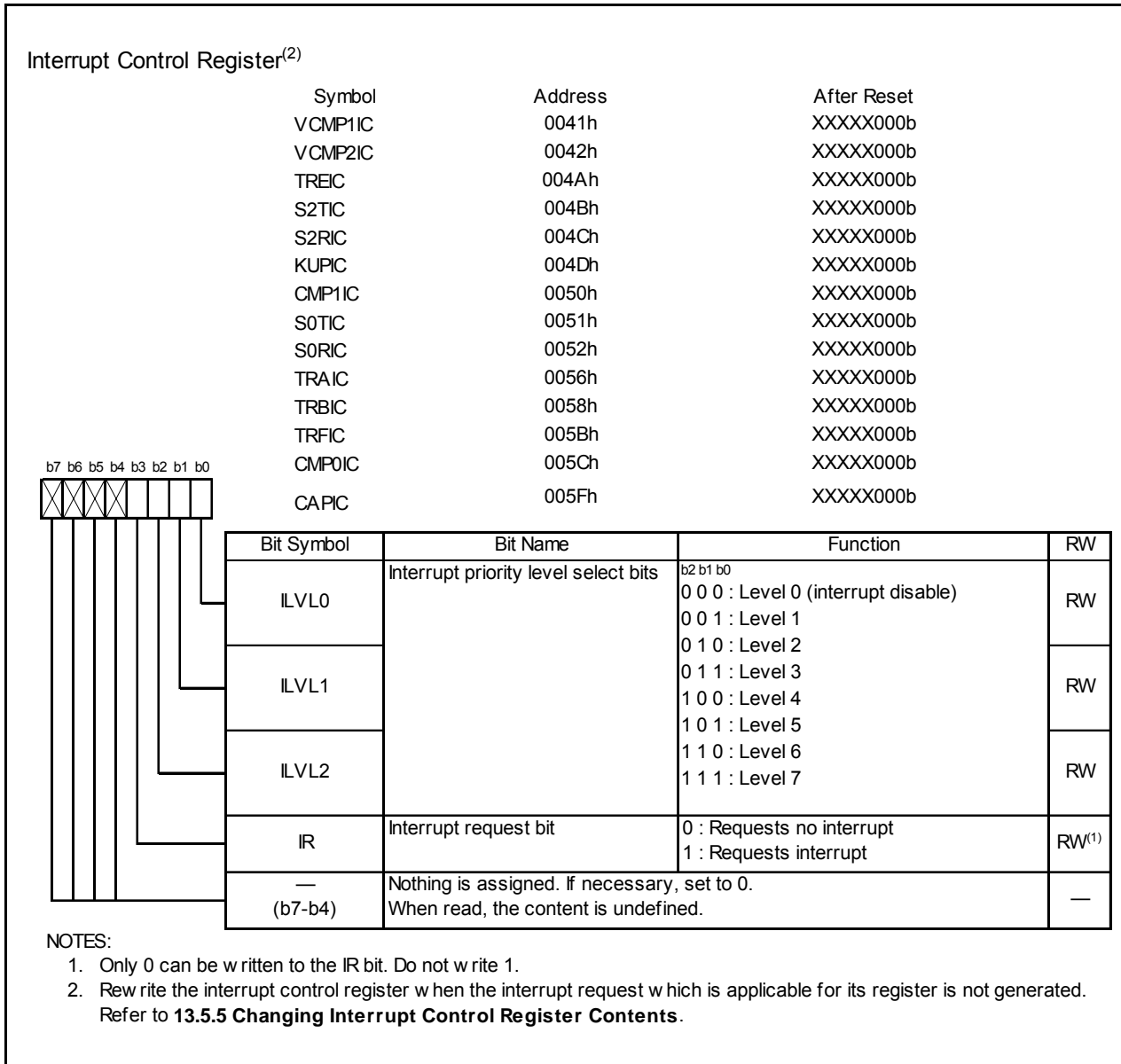


Figure 13.3 Interrupt Control Register

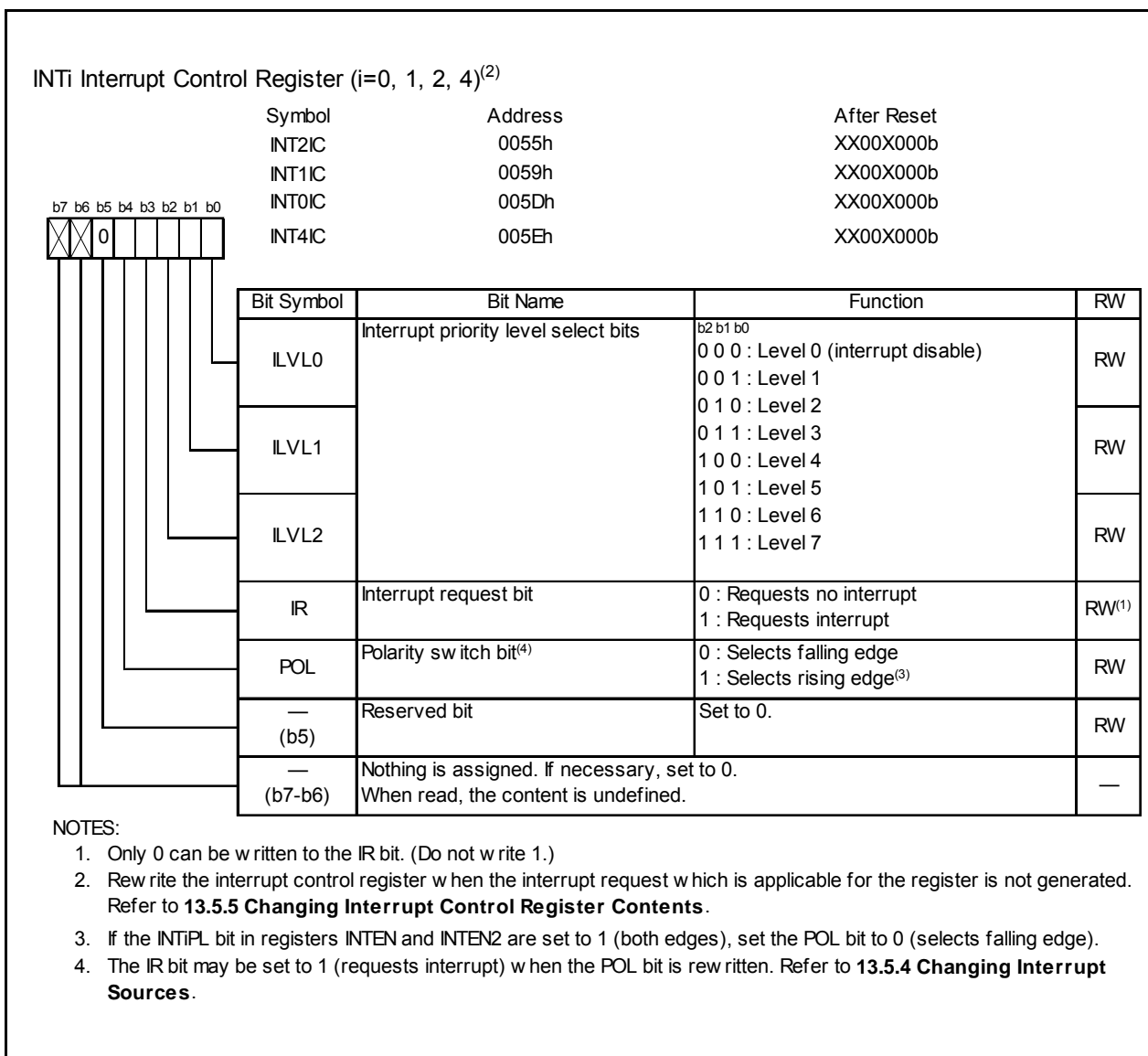


Figure 13.4 INT<sub>i</sub>IC Register (i=0, 1, 2, 4)

### 13.1.6.1 I Flag

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

### 13.1.6.2 IR Bit

The IR bit is set to 1 (interrupt requested) when an interrupt request is generated. Then, when the interrupt request is acknowledged and the CPU branches to the corresponding interrupt vector, the IR bit is set to 0 (= interrupt not requested).

The IR bit can be set to 0 by a program. Do not write 1 to this bit.

### 13.1.6.3 ILVL2 to ILVL0 Bits and IPL

Interrupt priority levels can be set using bits ILVL2 to ILVL0.


Table 13.3 lists the Settings of Interrupt Priority Levels and Table 13.4 lists the Interrupt Priority Levels Enabled by IPL.

The following are conditions under which an interrupt is acknowledged:

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

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

**Table 13.3 Settings of Interrupt Priority Levels**

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

**Table 13.4 Interrupt Priority Levels Enabled by IPL**

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

### 13.1.6.4 Interrupt Sequence

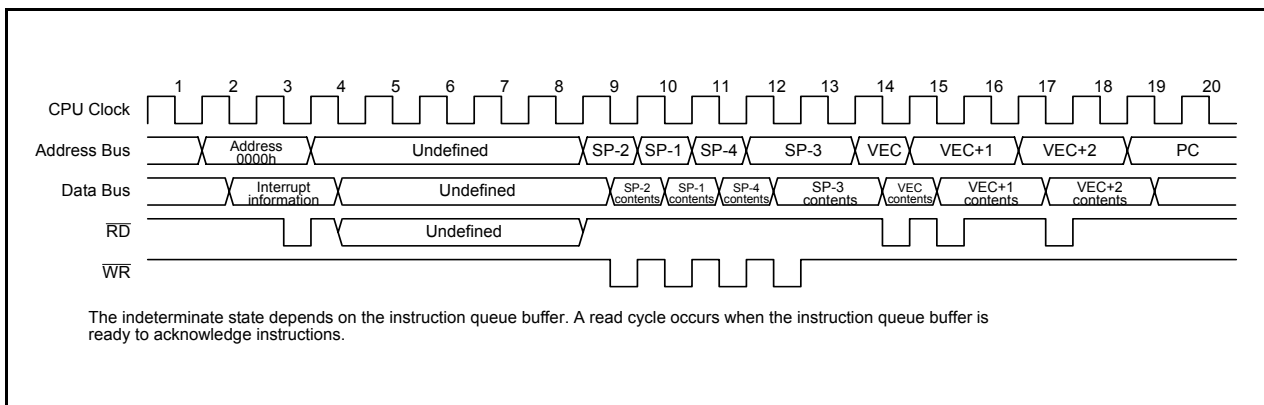
An interrupt sequence is performed between an interrupt request acknowledgement and interrupt routine execution.

When an interrupt request is generated while an instruction is being executed, the CPU determines its interrupt priority level after the instruction is completed. The CPU starts the interrupt sequence from the following cycle. However, for the SMOVB, SMOVF, SSTR, or RMPA instructions, if an interrupt request is generated while the instruction is being executed, the MCU suspends the instruction to start the interrupt sequence. The interrupt sequence is performed as indicated below.

Figure 13.5 shows the Time Required for Executing Interrupt Sequence.

- (1) The CPU gets interrupt information (interrupt number and interrupt request level) by reading address 0000h. The IR bit for the corresponding interrupt is set to 0 (interrupt not requested).
- (2) The FLG register is saved to a temporary register<sup>(1)</sup> in the CPU immediately before entering the interrupt sequence.
- (3) The I, D and U flags in the FLG register are set as follows:  
The I flag is set to 0 (interrupts disabled).  
The D flag is set to 0 (single-step interrupt disabled).  
The U flag is set to 0 (ISP selected).  
However, the U flag does not change state if an INT instruction for software interrupt number 32 to 63 is executed.
- (4) The CPU's internal temporary register<sup>(1)</sup> is saved to the stack.
- (5) The PC is saved to the stack.
- (6) The interrupt priority level of the acknowledged interrupt is set in the IPL.
- (7) The starting address of the interrupt routine set in the interrupt vector is stored in the PC.

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



**Figure 13.5 Time Required for Executing Interrupt Sequence**

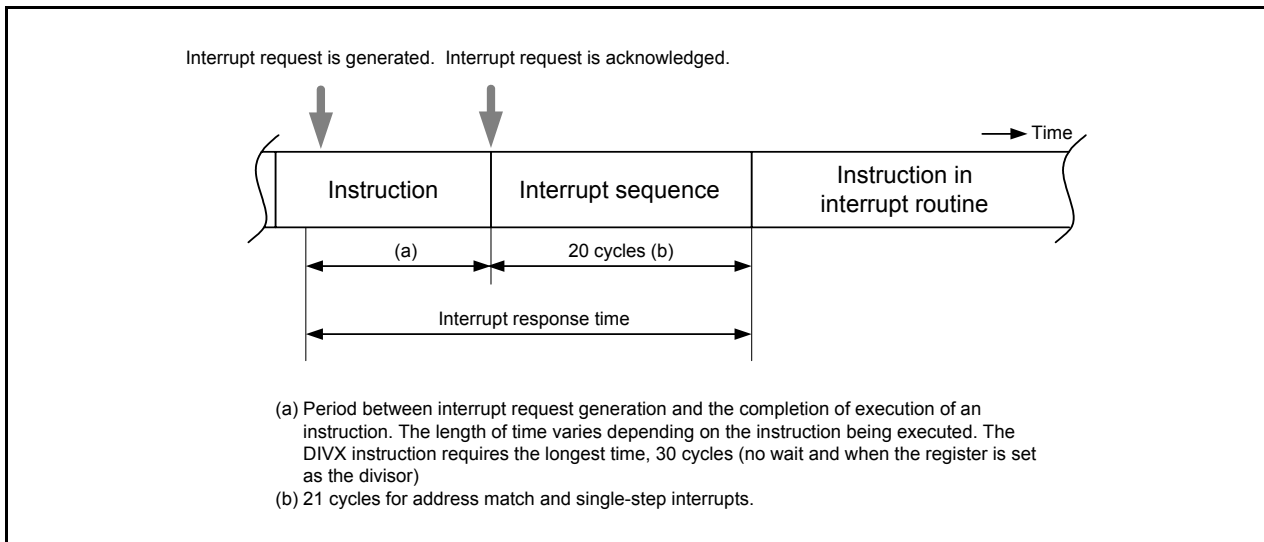
NOTE:

1. This register cannot be used by user.



### 13.1.6.5 Interrupt Response Time

Figure 13.6 shows the Interrupt Response Time. The interrupt response time is the period between an interrupt request generation and the execution of the first instruction in the interrupt routine. The interrupt response time includes the period between interrupt request generation and the completion of execution of the instruction (refer to (a) in **Figure 13.6**) and the period required to perform the interrupt sequence (20 cycles, refer to (b) in **Figure 13.6**).



**Figure 13.6** Interrupt Response Time

### 13.1.6.6 IPL Change when Interrupt Request is Acknowledged

When an interrupt request of a maskable interrupt is acknowledged, the interrupt priority level of the acknowledged interrupt is set in the IPL.

When a software interrupt or special interrupt request is acknowledged, the level listed in Table 13.5 is set in the IPL.

Table 13.5 lists the IPL Value When Software or Special Interrupt Is Acknowledged.

**Table 13.5** IPL Value When Software or Special Interrupt Is Acknowledged

Interrupt Source	Value Set in IPL
Watchdog timer, voltage monitor 1, voltage monitor 2, comparator 1 <sup>(1)</sup> , comparator 2 <sup>(1)</sup> , address break	7
Software, address match, single-step	Not changed

NOTE:

1. When non-maskable interrupts is selected.

### 13.1.6.7 Saving a Register

In the interrupt sequence, the FLG register and PC are saved to the stack.

After an extended 16 bits, 4 high-order bits in the PC and 4 high-order (IPL) and 8 low-order bits in the FLG register, are saved to the stack, the 16 low-order bits in the PC are saved.

Figure 13.7 shows the Stack State Before and After Acknowledgement of Interrupt Request.

The other necessary registers are saved by a program at the beginning of the interrupt routine. The PUSHM instruction can save several registers in the register bank being currently used<sup>(1)</sup> with a single instruction.

NOTE:

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

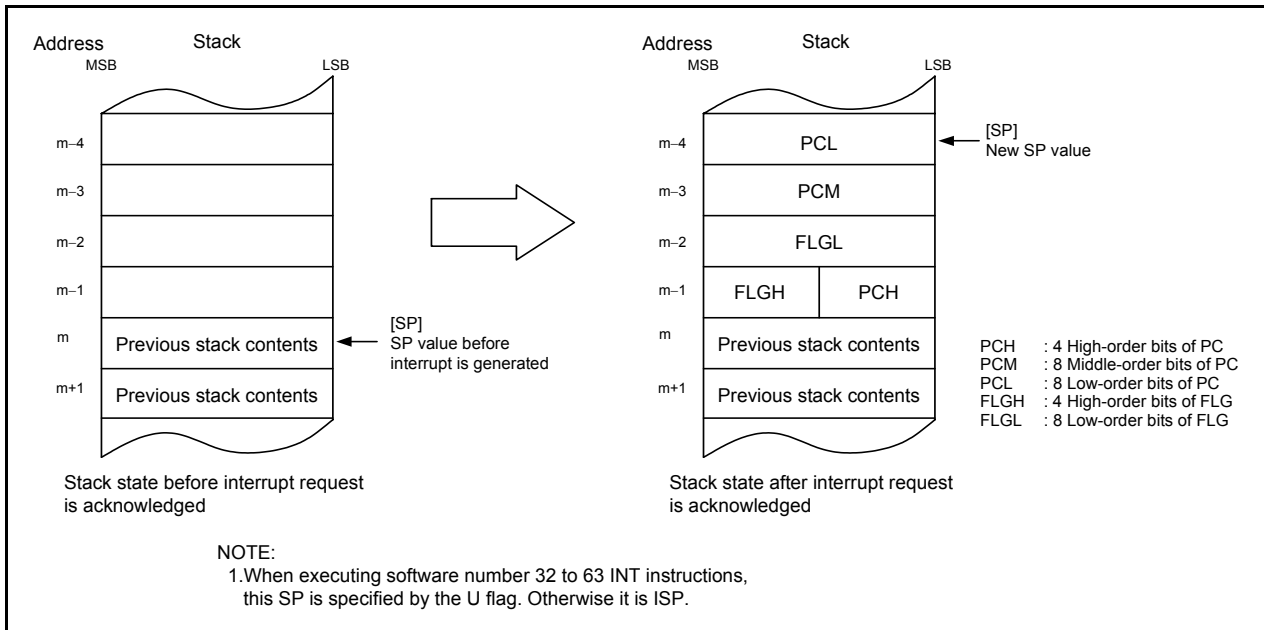


Figure 13.7 Stack State Before and After Acknowledgement of Interrupt Request

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

Figure 13.8 shows the Register Saving Operation.

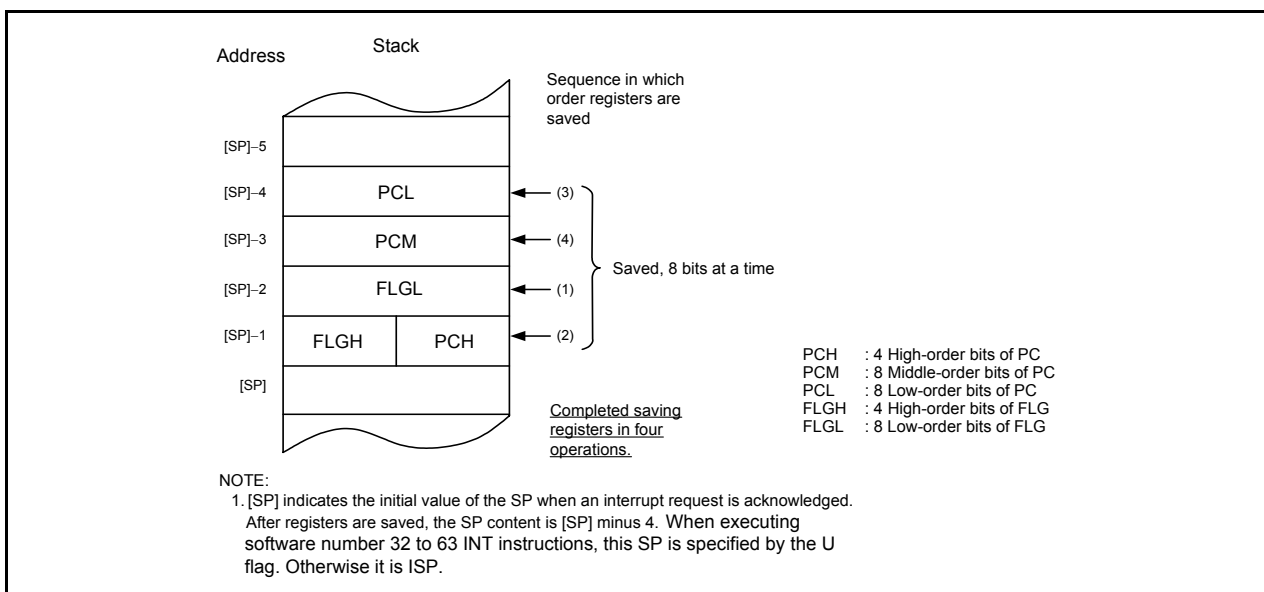


Figure 13.8 Register Saving Operation

### 13.1.6.8 Returning from an Interrupt Routine

When the REIT instruction is executed at the end of an interrupt routine, the FLG register and PC, which have been saved to the stack, are automatically restored. The program, that was running before the interrupt request was acknowledged, starts running again.

Restore registers saved by a program in an interrupt routine using the POPM instruction or others before executing the REIT instruction.

### 13.1.6.9 Interrupt Priority

If two or more interrupt requests are generated while a single instruction is being executed, the interrupt with the higher priority is acknowledged.

Set bits ILVL2 to ILVL0 to select the desired priority level for maskable interrupts (peripheral functions). However, if two or more maskable interrupts have the same priority level, their interrupt priority is resolved by hardware, and the higher priority interrupts acknowledged.

The priority levels of special interrupts, such as reset (reset has the highest priority) and watchdog timer, are set by hardware.

Figure 13.9 shows the Priority Levels of Hardware Interrupts.

The interrupt priority does not affect software interrupts. The MCU jumps to the interrupt routine when the instruction is executed.

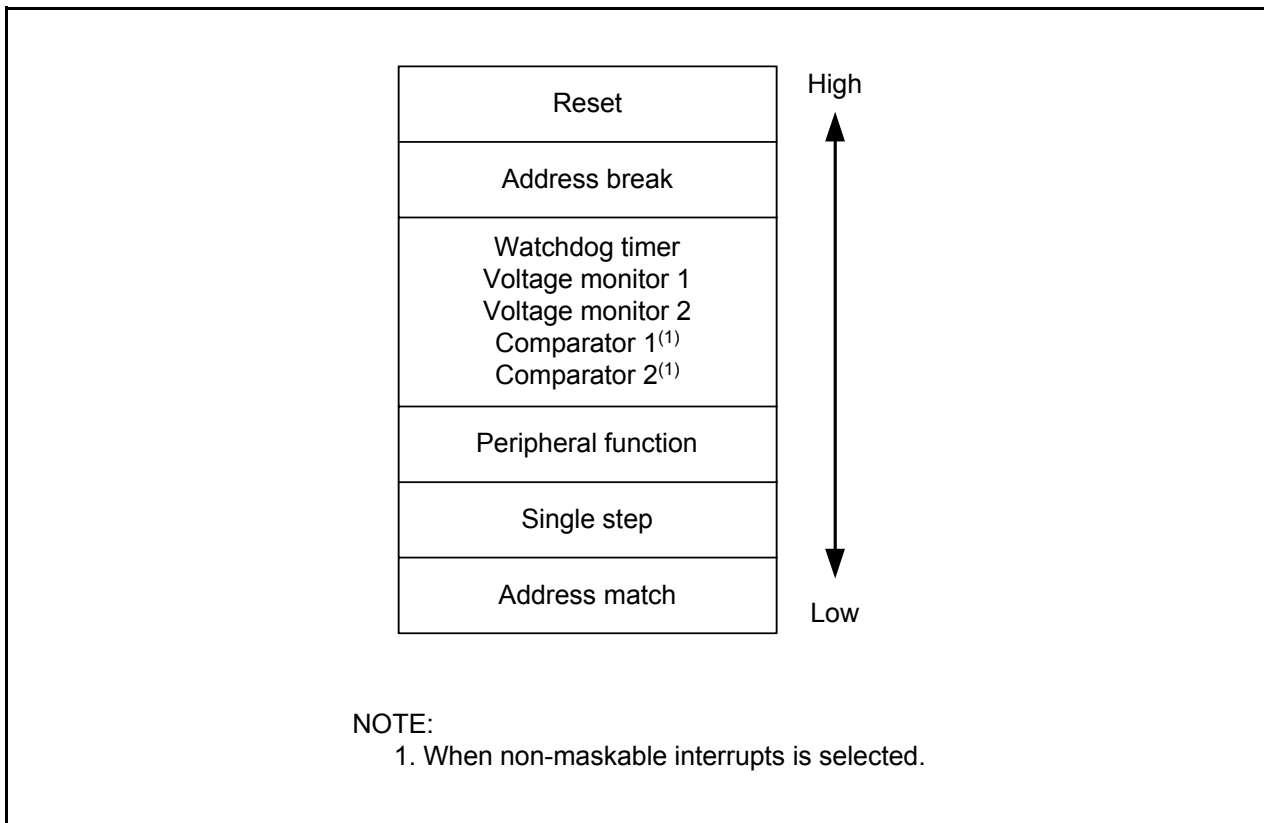


Figure 13.9 Priority Levels of Hardware Interrupts

### 13.1.6.10 Interrupt Priority Judgement Circuit

The interrupt priority judgement circuit selects the highest priority interrupt, as shown in Figure 13.10.

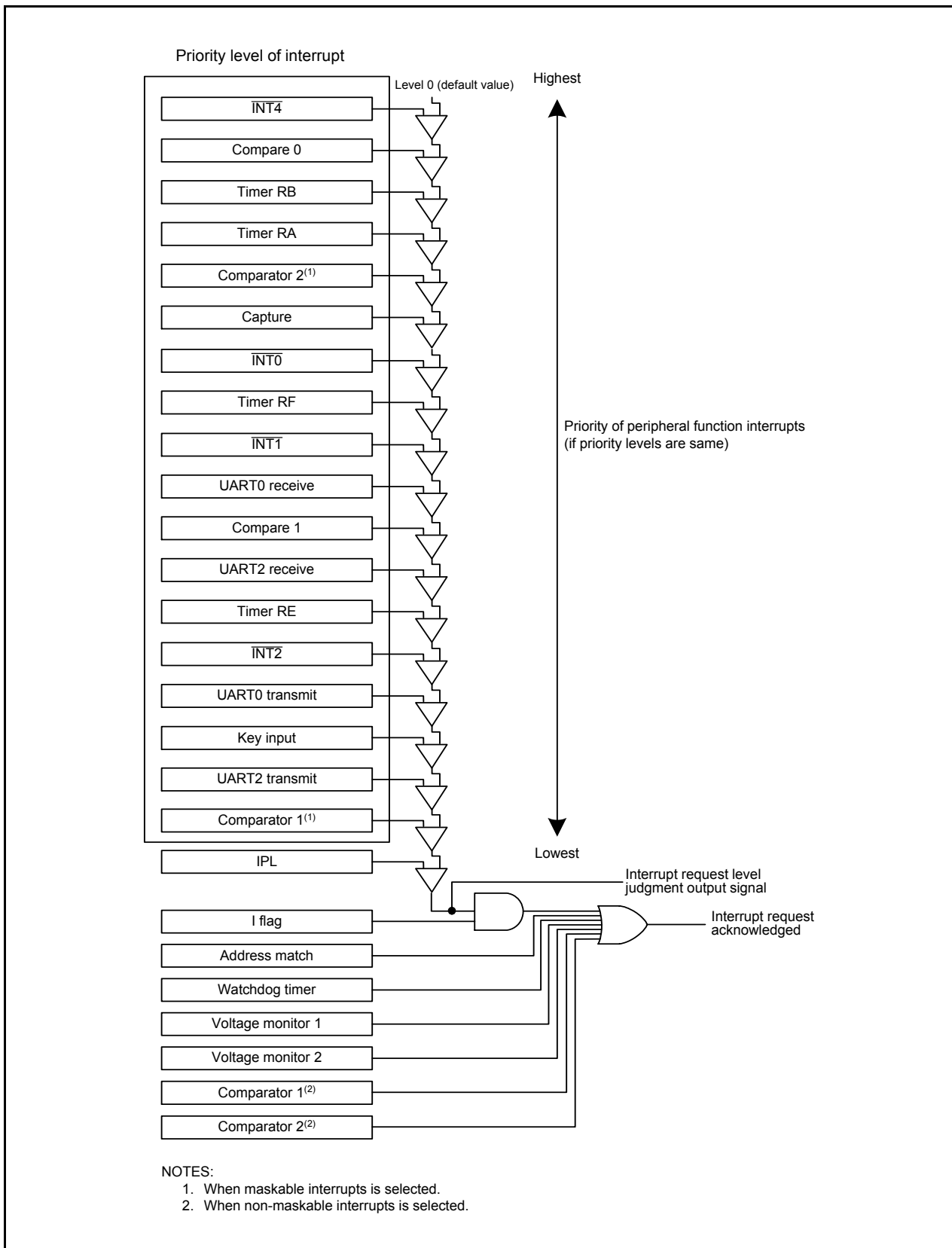


Figure 13.10 Interrupt Priority Level Judgement Circuit

## 13.2 $\overline{\text{INT}}$ Interrupt

### 13.2.1 $\overline{\text{INT}}_i$ Interrupt ( $i = 0, 1, 2, 4$ )

The  $\overline{\text{INT}}_i$  interrupt is generated by an  $\overline{\text{INT}}_i$  input. Table 13.6 lists the Pin Configuration of  $\overline{\text{INT}}$  Interrupt. When using the  $\overline{\text{INT}}_i$  interrupt, the  $\text{INT}_i\text{EN}$  bit in registers  $\text{INTEN}$  and  $\text{INTEN2}$  are set to 1 (enable). The edge polarity is selected using the  $\text{INT}_i\text{PL}$  bit in registers  $\text{INTEN}$  and  $\text{INTEN2}$ , and the  $\text{POL}$  bit in the  $\text{INT}_i\text{IC}$  register.

Inputs can be passed through a digital filter with three different sampling clocks.

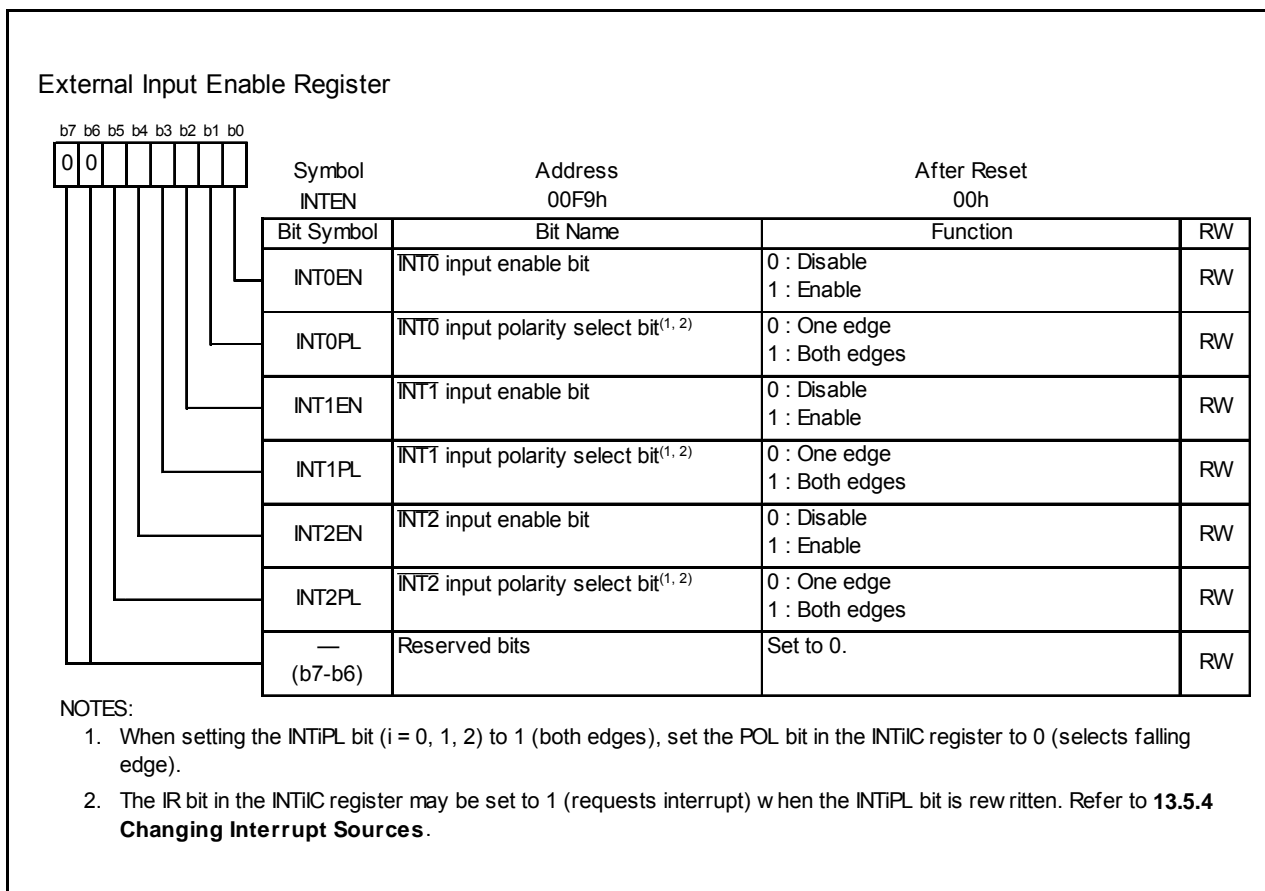
Figure 13.11 shows the  $\text{INTEN}$  Register. Figure 13.12 shows the  $\text{INTF}$  Register. Figure 13.13 shows the  $\text{INTEN2}$  Register. Figure 13.14 shows the  $\text{INTF2}$  Register.

**Table 13.6 Pin Configuration of  $\overline{\text{INT}}$  Interrupt**

Pin name	Input/Output	Function
$\overline{\text{INT}}_0$ (P4_5)	Input	$\overline{\text{INT}}_0$ interrupt input, Timer RB external trigger input
$\overline{\text{INT}}_1$ (P1_5, P1_7, or P3_6) <sup>(1)</sup>	Input	$\overline{\text{INT}}_1$ interrupt input
$\overline{\text{INT}}_2$ (P3_2)	Input	$\overline{\text{INT}}_2$ interrupt input
$\overline{\text{INT}}_4$ (P0_6)	Input	$\overline{\text{INT}}_4$ interrupt input

NOTE:

- The  $\overline{\text{INT}}_1$  pin is selected by the  $\text{INT1SEL}$  bit in the  $\text{PMR}$  register and the  $\text{TIOSEL}$  bit in the  $\text{TRAIOC}$  register. Refer to **8. I/O Ports** for details.



**Figure 13.11 INTEN Register**

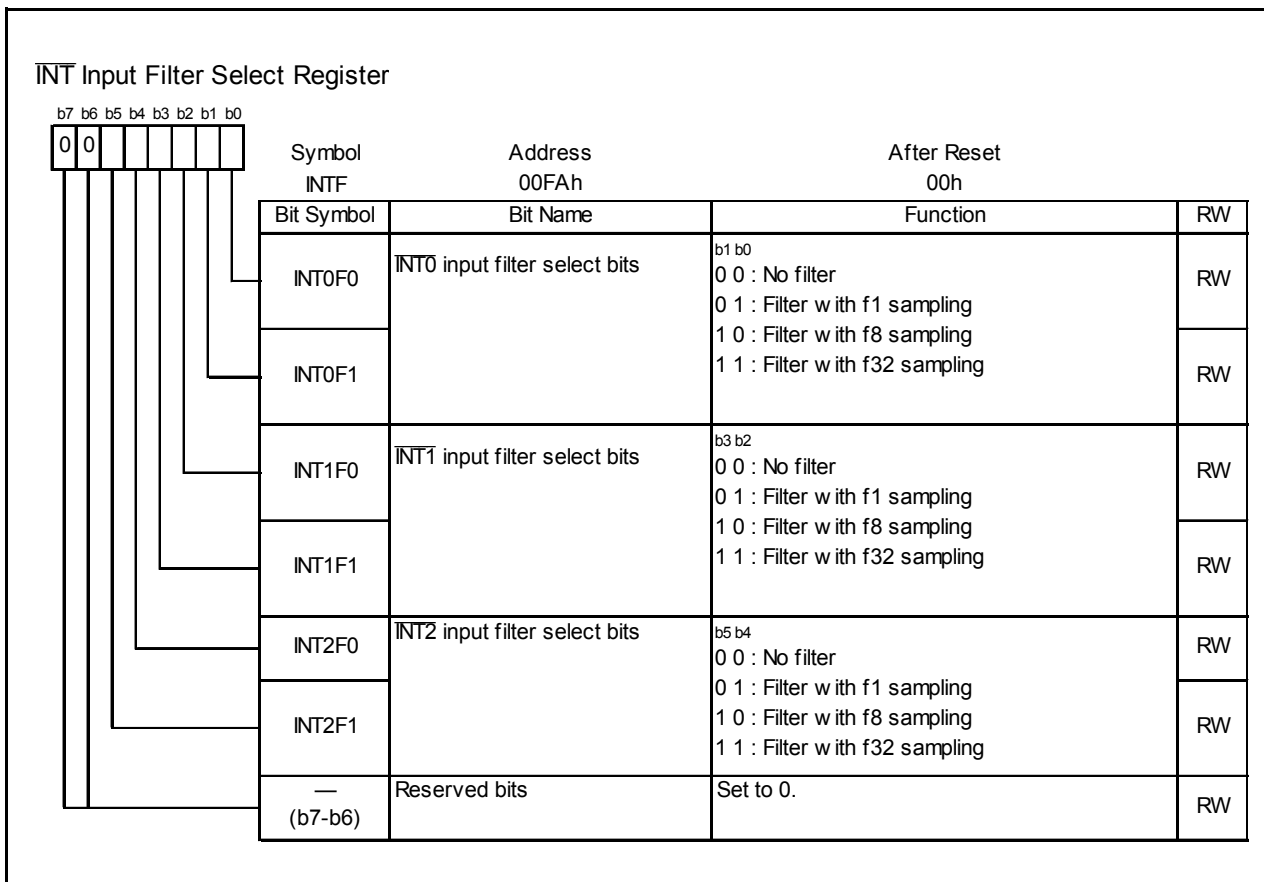


Figure 13.12 INTF Register

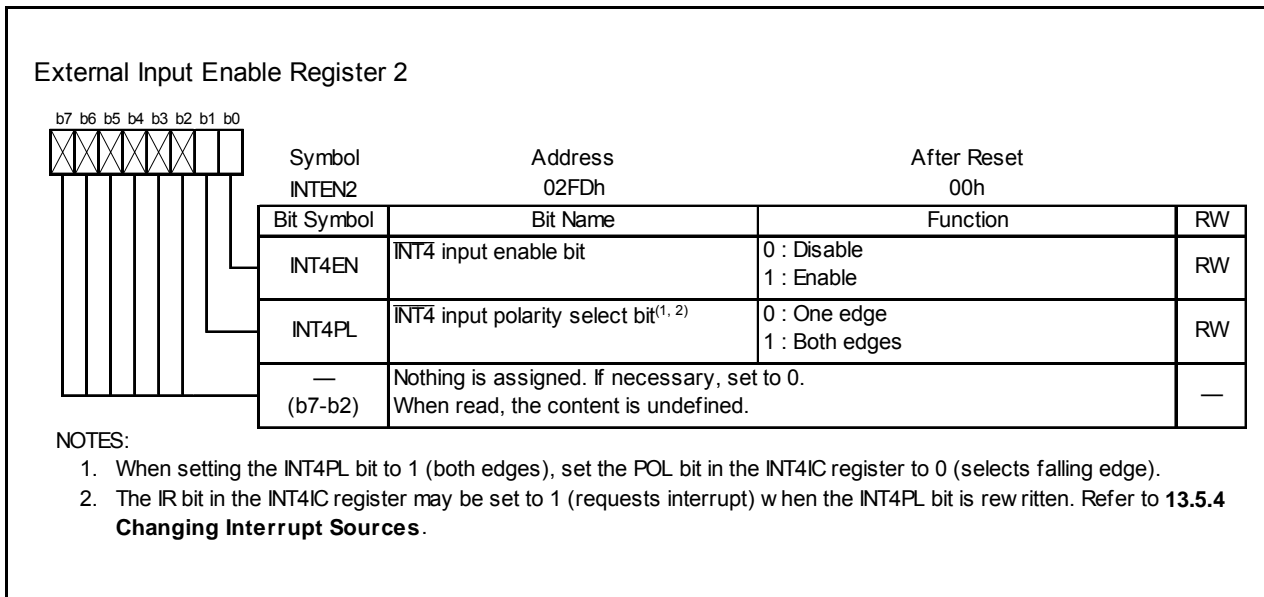
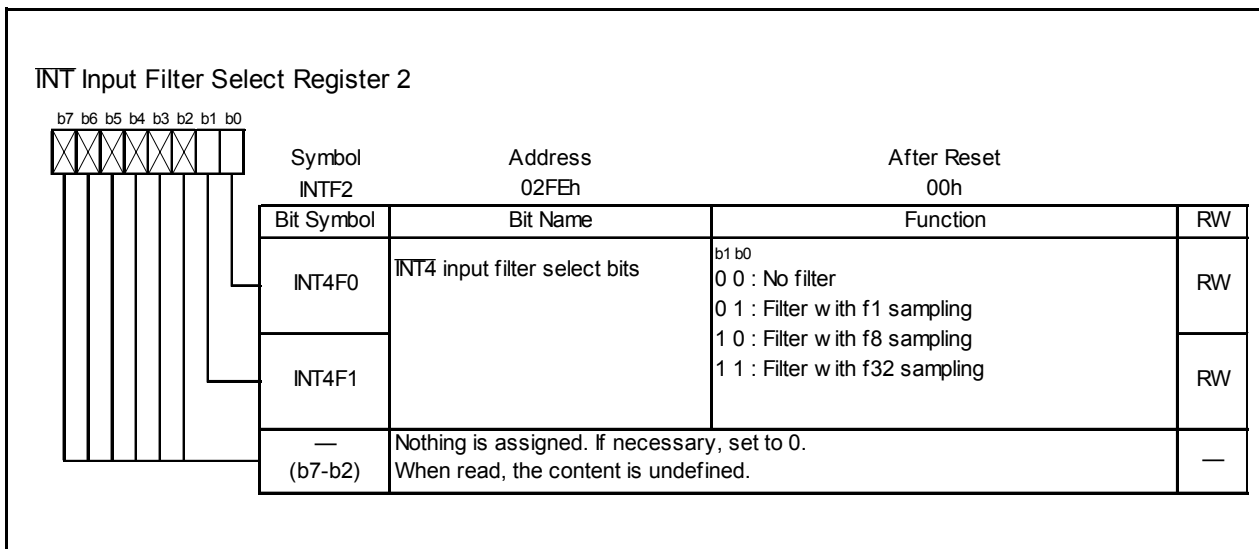


Figure 13.13 INTEN2 Register



**Figure 13.14 INTF2 Register**

### 13.2.2 $\overline{\text{INT}}_i$ Input Filter (i = 0, 1, 2, 4)

The  $\overline{\text{INT}}_i$  input contains a digital filter. The sampling clock is selected by bits  $\text{INTiF1}$  to  $\text{INTiF0}$  in registers  $\text{INTF}$  and  $\text{INTF2}$ . The IR bit in the  $\text{INTiIC}$  register is set to 1 (interrupt requested) when the  $\overline{\text{INT}}_i$  level is sampled for every sampling clock and the sampled input level matches three times.

Figure 13.15 shows the Configuration of  $\overline{\text{INT}}_i$  Input Filter. Figure 13.16 shows an Operating Example of  $\overline{\text{INT}}_i$  Input Filter.

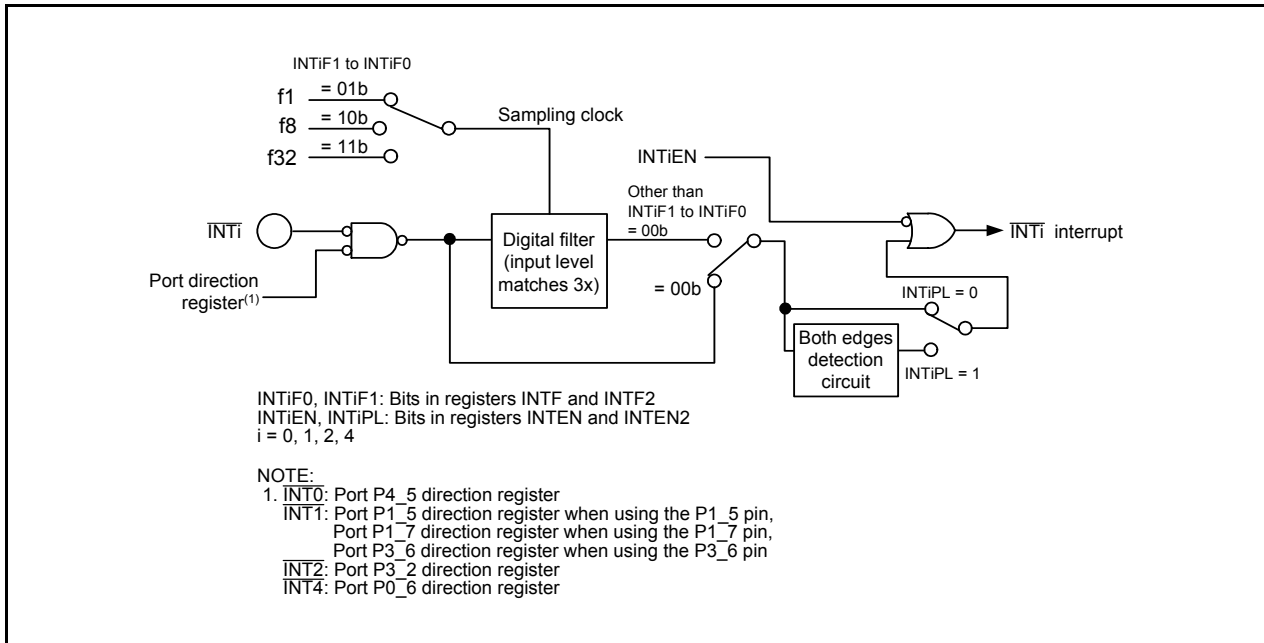


Figure 13.15 Configuration of  $\text{INT}_i$  Input Filter

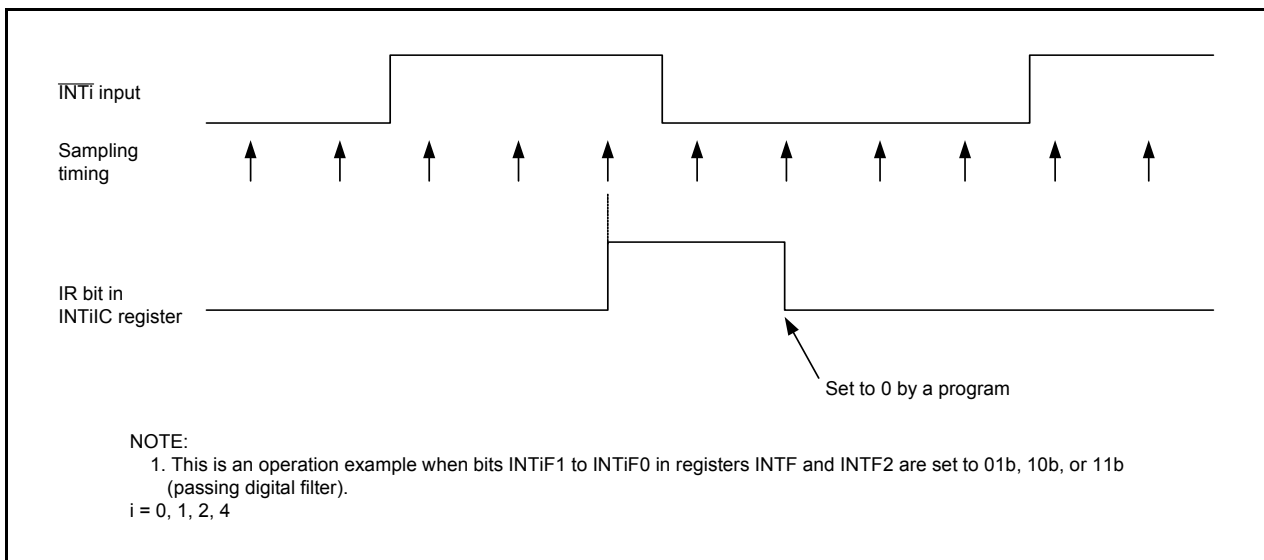


Figure 13.16 Operating Example of  $\text{INT}_i$  Input Filter



### 13.3 Key Input Interrupt

A key input interrupt request is generated by one of the input edges of the  $\overline{KI0}$  to  $\overline{KI3}$  pins. Table 13.7 lists the Pin Configuration of Key Input Interrupt. The key input interrupt can be used as a key-on wake-up function to exit wait or stop mode.

The  $KIiEN$  ( $i = 0$  to  $3$ ) bit in the  $KIEN$  register can select whether the pins are used as  $\overline{KIi}$  input. The  $KIiPL$  bit in the  $KIEN$  register can select the input polarity.

When inputting “L” to the  $\overline{KIi}$  pin which sets the  $KIiPL$  bit to 0 (falling edge), the input of the other pins  $\overline{KI0}$  to  $\overline{KI3}$  is not detected as interrupts. Also, when inputting “H” to the  $\overline{KIi}$  pin, which sets the  $KIiPL$  bit to 1 (rising edge), the input of the other pins  $\overline{KI0}$  to  $\overline{KI3}$  is not detected as interrupts.

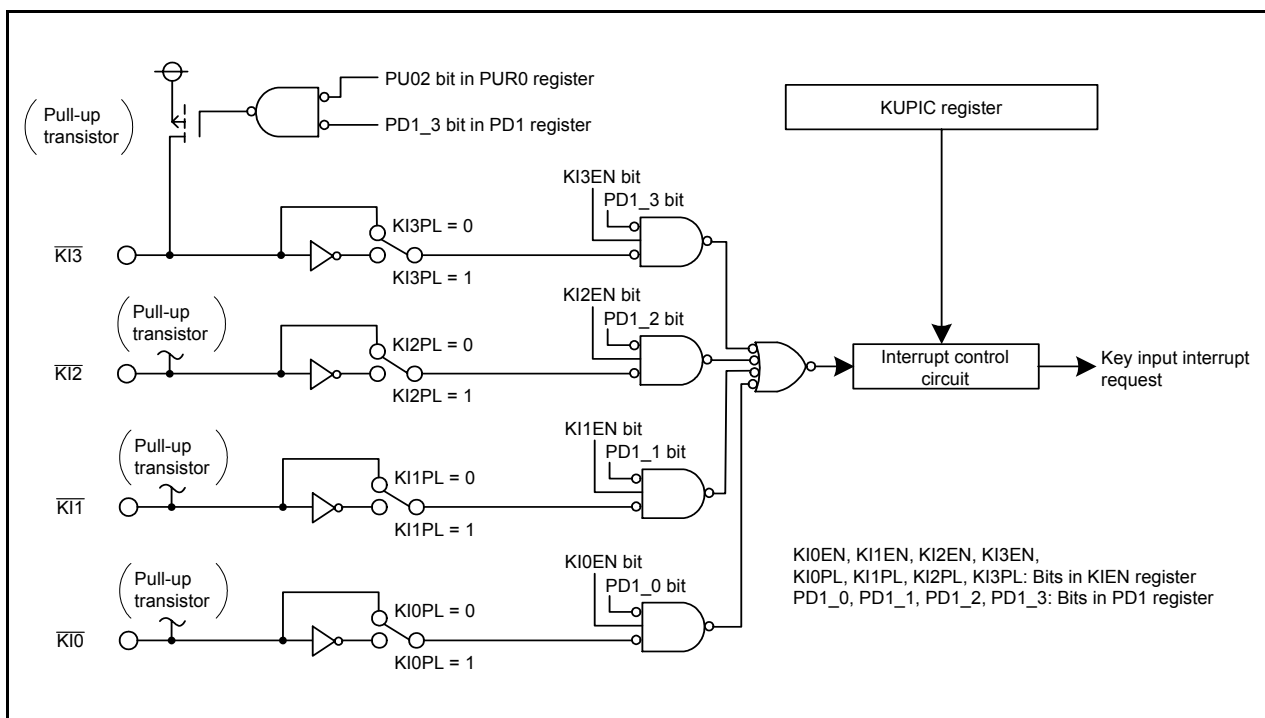
Figure 13.17 shows a Block Diagram of Key Input Interrupt and Figure 13.18 shows the  $KIEN$  Register.

**Table 13.7 Pin Configuration of Key Input Interrupt**

Pin name	Input/Output	Function
$\overline{KI0}$ (P0_7 or P1_0 <sup>(1)</sup> )	Input	$\overline{KI0}$ input
$\overline{KI1}$ (P1_1 or P6_6 <sup>(2)</sup> )	Input	$\overline{KI1}$ input
$\overline{KI2}$ (P1_2)	Input	$\overline{KI2}$ input
$\overline{KI3}$ (P1_3)	Input	$\overline{KI3}$ input

**NOTES:**

1. The  $\overline{KI0}$  pin is selected by the  $KI0SEL$  bit in the  $PINSR4$  register. Refer to **8. I/O Ports** for details.
2. The  $\overline{KI1}$  pin is selected by the  $KI1SEL$  bit in the  $PINSR4$  register. Refer to **8. I/O Ports** for details.



**Figure 13.17 Block Diagram of Key Input Interrupt**

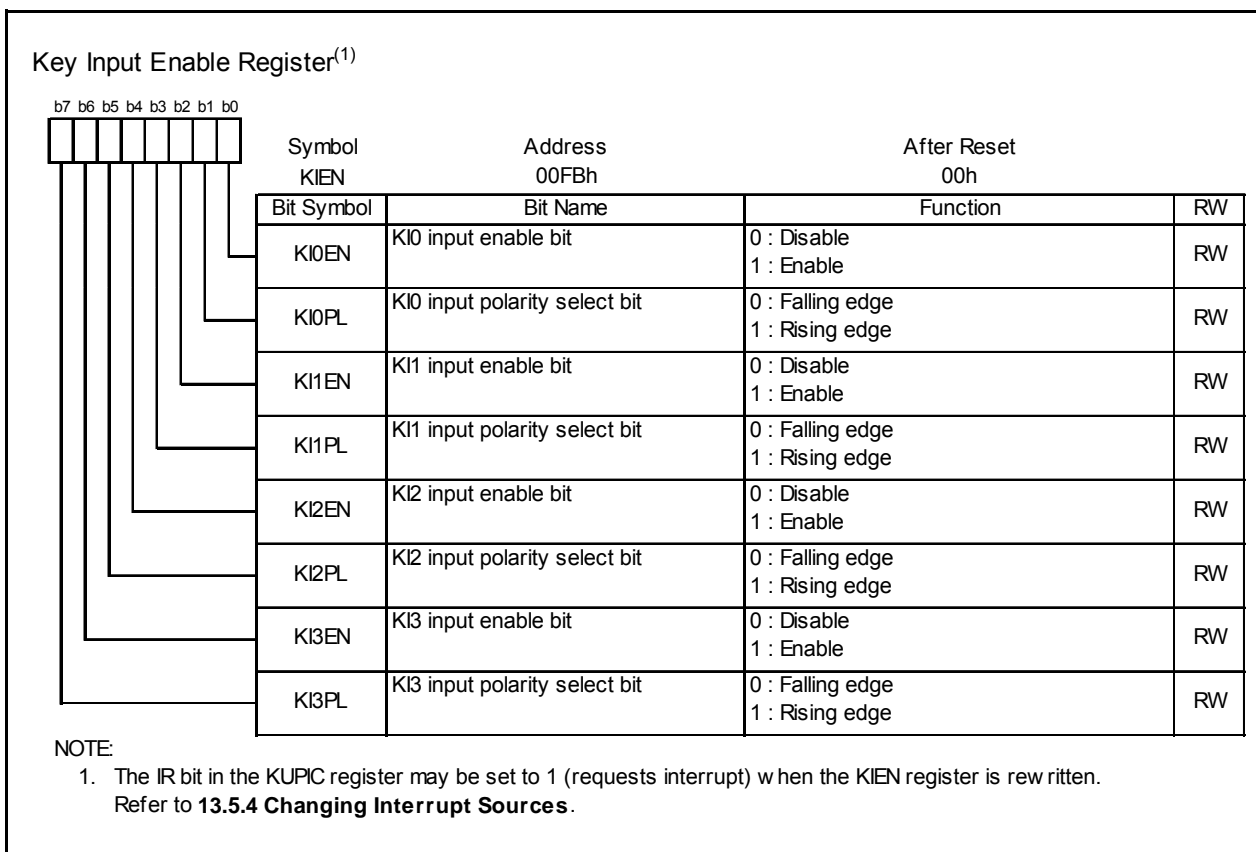


Figure 13.18 KIEN Register

### 13.4 Address Match Interrupt

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

Set the starting address of any instruction in the RMADi register. Bits AIER0 and AIER1 in the AIER0 register can be used to select enable or disable of the interrupt. The I flag and IPL do not affect the address match interrupt.

The value of the PC (refer to **13.1.6.7 Saving a Register** for the value of the PC) which is saved to the stack when an address match interrupt is acknowledged varies depending on the instruction at the address indicated by the RMADi register. (The appropriate return address is not saved on the stack.) When returning from the address match interrupt, return by one of the following means:

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

Table 13.8 lists the Values of PC Saved to Stack when Address Match Interrupt is Acknowledged and Table 13.9 lists the Correspondence Between Address Match Interrupt Sources and Associated Registers.

Figure 13.19 shows Registers AIER and RMAD0 to RMAD1.

**Table 13.8 Values of PC Saved to Stack when Address Match Interrupt is Acknowledged**

Address Indicated by RMADi Register (i = 0 or 1)	PC Value Saved <sup>(1)</sup>
<ul style="list-style-type: none"> <li>• Instruction with 2-byte operation code<sup>(2)</sup></li> <li>• Instruction with 1-byte operation code<sup>(2)</sup></li> </ul> ADD.B:S #IMM8,dest SUB.B:S #IMM8,dest AND.B:S #IMM8,dest OR.B:S #IMM8,dest MOV.B:S #IMM8,dest STZ #IMM8,dest STNZ #IMM8,dest STZX #IMM81,#IMM82,dest CMP.B:S #IMM8,dest PUSHM src POPM dest JMPS #IMM8 JSRS #IMM8 MOV.B:S #IMM,dest (however, dest = A0 or A1)	Address indicated by RMADi register + 2
Instructions other than the above	Address indicated by RMADi register + 1

NOTES:

1. Refer to the **13.1.6.7 Saving a Register** for the PC value saved.
2. Operation code: Refer to the **R8C/Tiny Series Software Manual (REJ09B0001)**.

**Chapter 4. Instruction Code/Number of Cycles** contains diagrams showing operation code below each syntax. Operation code is shown in the bold frame in the diagrams.

**Table 13.9 Correspondence Between Address Match Interrupt Sources and Associated Registers**

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

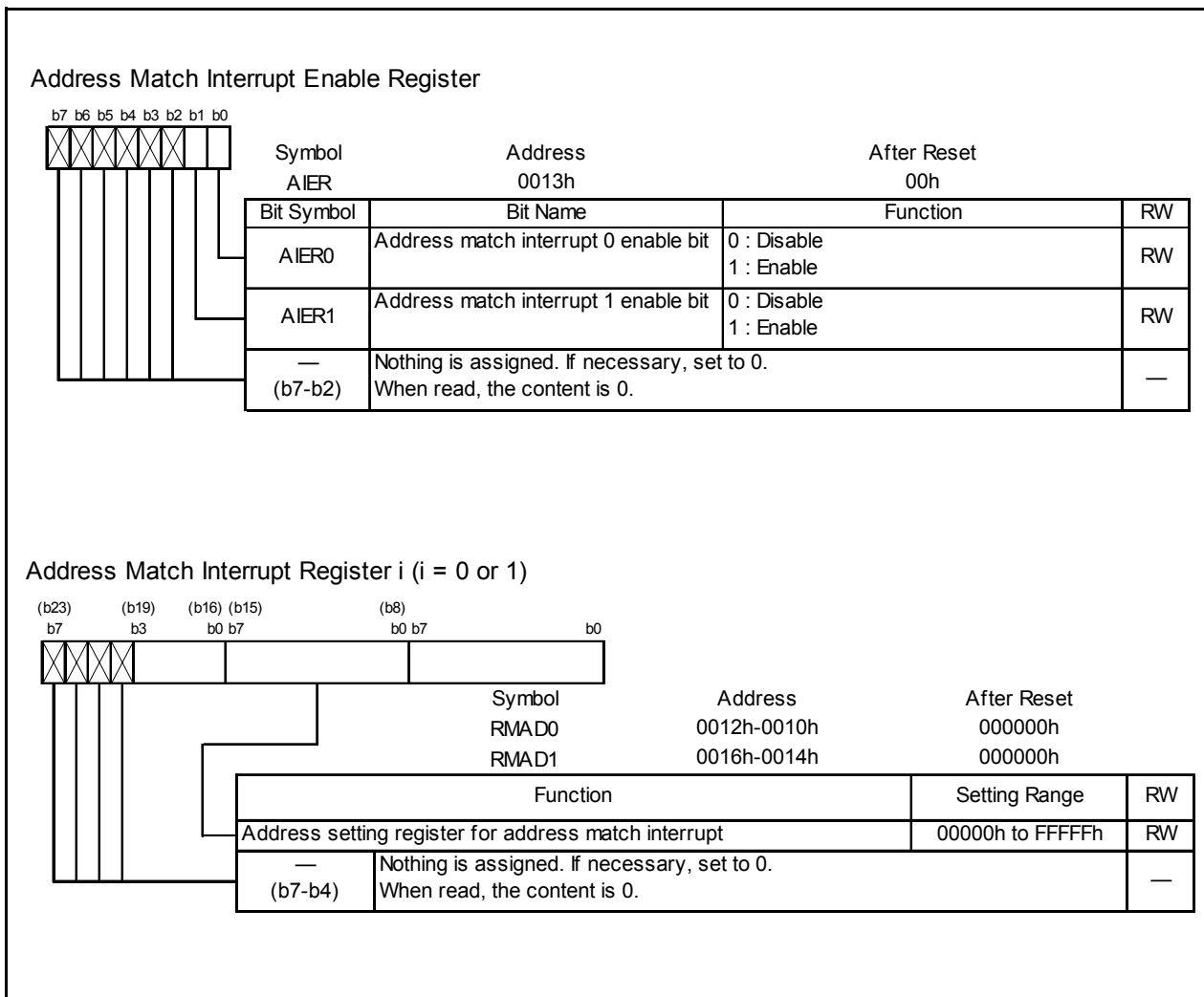


Figure 13.19 Registers AIER and RMAD0 to RMAD1

## 13.5 Notes on Interrupts

### 13.5.1 Reading Address 00000h

Do not read address 00000h by a program. When a maskable interrupt request is acknowledged, the CPU reads interrupt information (interrupt number and interrupt request level) from 00000h in the interrupt sequence. At this time, the acknowledged interrupt IR bit is set to 0.

If address 00000h is read by a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is set to 0. This may cause the interrupt to be canceled, or an unexpected interrupt to be generated.

### 13.5.2 SP Setting

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

### 13.5.3 External Interrupt and Key Input Interrupt

Either “L” level or an “H” level of width shown in the Electrical Characteristics is necessary for the signal input to pins  $\overline{\text{INT0}}$ ,  $\overline{\text{INT1}}$ ,  $\overline{\text{INT2}}$ ,  $\overline{\text{INT4}}$  and pins  $\overline{\text{KI0}}$  to  $\overline{\text{KI3}}$ , regardless of the CPU clock.

For details, refer to **Table 22.17** (VCC = 5V), **Table 22.23** (VCC = 3V), and **Table 22.29** (VCC = 2.2V) **External Interrupt INT<sub>i</sub> (i = 0, 1, 2, 4) Input**.

### 13.5.4 Changing Interrupt Sources

The IR bit in the interrupt control register may be set to 1 (interrupt requested) when the interrupt source changes. When using an interrupt, set the IR bit to 0 (no interrupt requested) after changing the interrupt source. In addition, changes of interrupt sources include all factors that change the interrupt sources assigned to individual software interrupt numbers, polarities, and timing. Therefore, if a mode change of a peripheral function involves interrupt sources, edge polarities, and timing, set the IR bit to 0 (no interrupt requested) after the change. Refer to the individual peripheral function for its related interrupts.

Figure 13.20 shows an Example of Procedure for Changing Interrupt Sources.

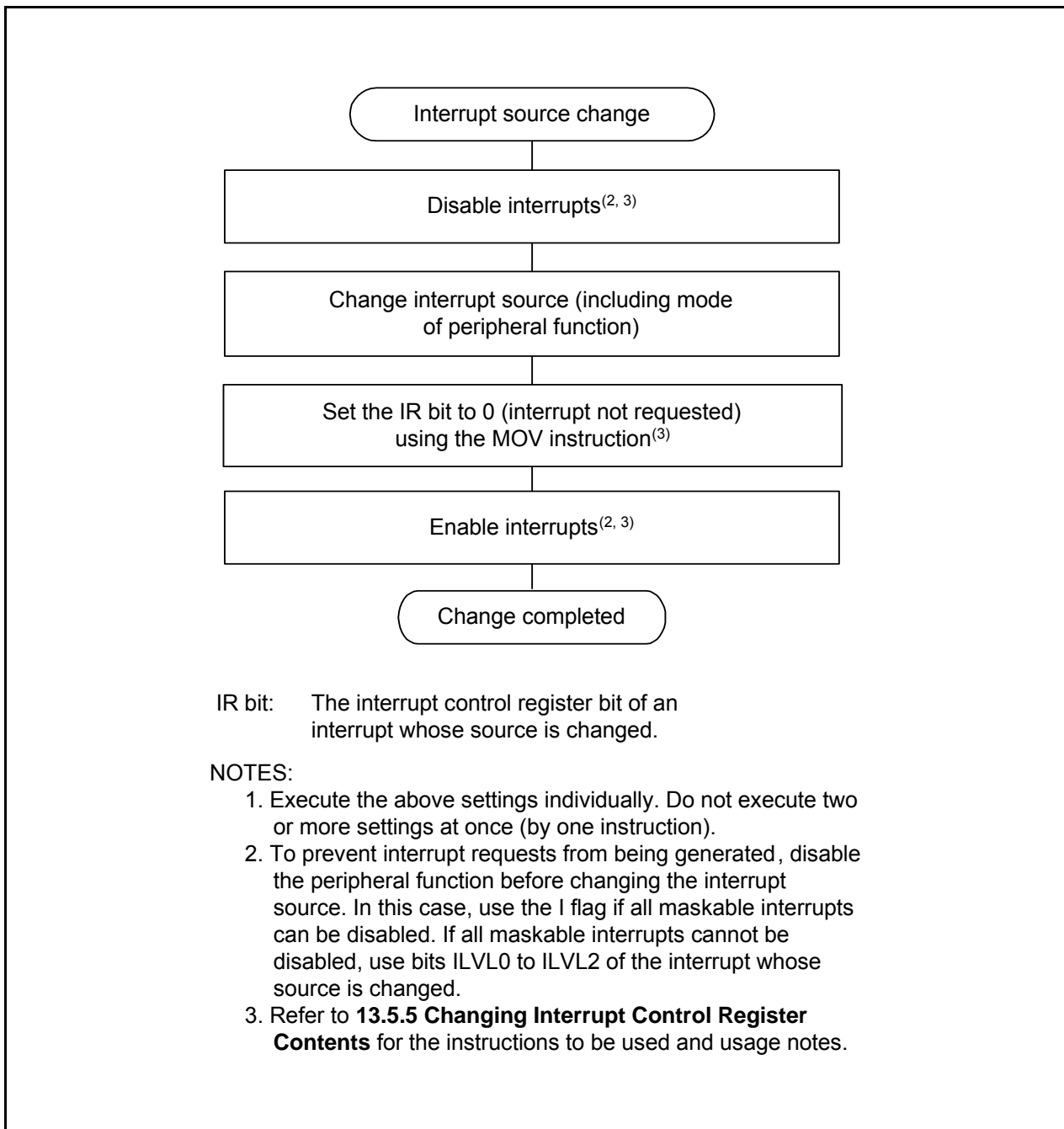


Figure 13.20 Example of Procedure for Changing Interrupt Sources

### 13.5.5 Changing Interrupt Control Register Contents

- (a) The contents of an interrupt control register can only be changed while no interrupt requests corresponding to that register are generated. If interrupt requests may be generated, disable interrupts before changing the interrupt control register contents.
- (b) When changing the contents of an interrupt control register after disabling interrupts, be careful to choose appropriate instructions.

#### Changing any bit other than IR bit

If an interrupt request corresponding to a register is generated while executing the instruction, the IR bit may not be set to 1 (interrupt requested), and the interrupt request may be ignored. If this causes a problem, use the following instructions to change the register: AND, OR, BCLR, BSET

#### Changing IR bit

If the IR bit is set to 0 (interrupt not requested), it may not be set to 0 depending on the instruction used. Therefore, use the MOV instruction to set the IR bit to 0.

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

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

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

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

#### Example 2: Use dummy read to delay FSET instruction

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

#### Example 3: Use POPC instruction to change I flag

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

## 14. ID Code Areas

### 14.1 Overview

The ID code areas are used to implement a function that prevents the flash memory from being rewritten in standard serial I/O mode. This function prevents the flash memory from read, rewritten, or erased.

The ID code areas are assigned to 0FFDFh, 0FFE3h, 0FFEBh, 0FFE7h, 0FFF3h, 0FFF7h, and 0FFFBh of the respective vector highest-order addresses of the fixed vector table. Figure 14.1 shows the ID Code Areas.

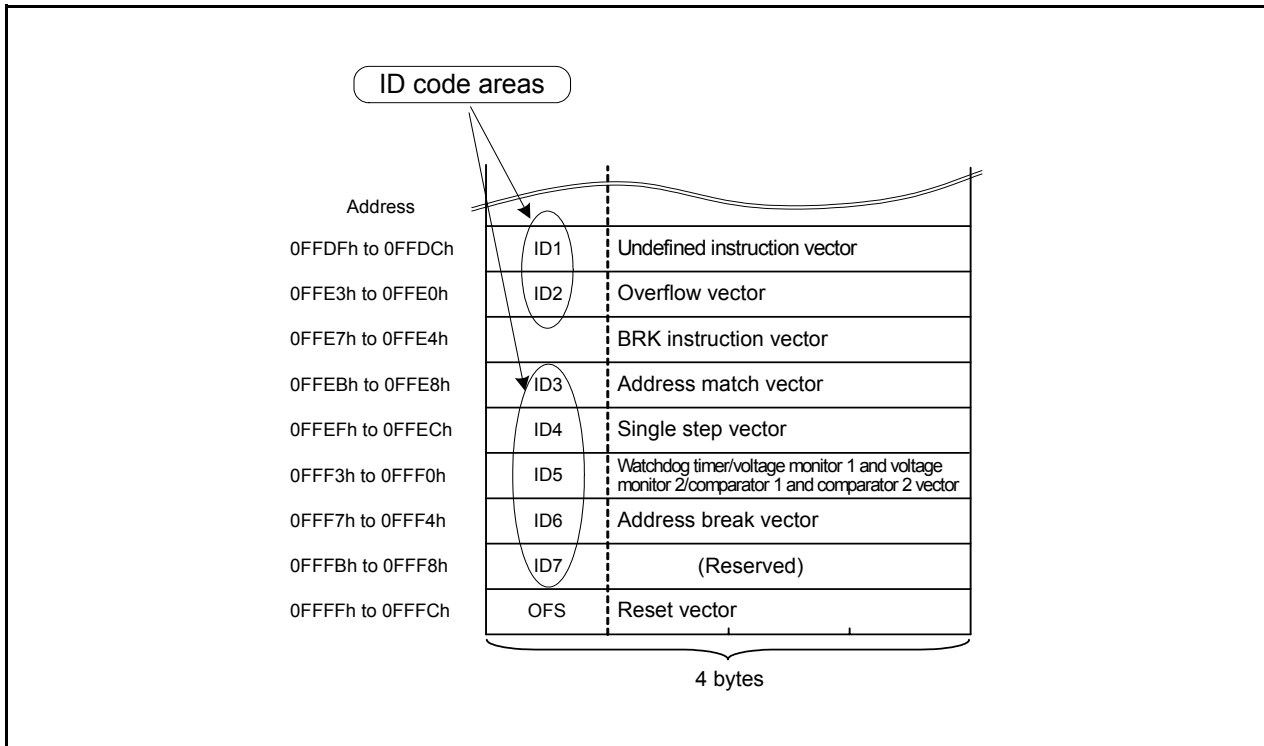


Figure 14.1 ID Code Areas

### 14.2 Functions

The ID code areas are used in standard serial I/O mode. Unless 3 bytes (addresses from 0FFFCh to 0FFFEh) of the reset vector are set to FFFFFFFh, the ID codes stored in the ID code areas and the ID codes sent from the serial programmer or the on-chip debugging emulator are checked to see if they match. If the ID codes match, the commands sent from the serial programmer or the on-chip debugging emulator are acknowledged. If the ID codes do not match, the commands are not acknowledged. To use the serial programmer or the on-chip debugging simulator, first write predetermined ID codes to the ID code areas.

As the ID code areas are allocated in the flash memory (not in the SFRs), they cannot be rewritten by executing an instruction. Write appropriate values when creating a program.



## 14.3 Notes on ID Code Areas

### 14.3.1 Setting Example of ID Code Areas

As the ID code areas are allocated in the flash memory (not in the SFRs), they cannot be rewritten by executing an instruction. Write appropriate values when creating a program. The following shows a setting example.

- To set 55h in all of the ID code areas

```
.org 00FFDCH
.word dummy | (55000000h) ; UND
.word dummy | (55000000h) ; INTO
.word dummy ; BREAK
.word dummy | (55000000h) ; ADDRESS MATCH
.word dummy | (55000000h) ; SET SINGLE STEP
.word dummy | (55000000h) ; WDT
.word dummy | (55000000h) ; ADDRESS BREAK
.word dummy | (55000000h) ; RESERVE
```

(Programming formats vary depending on the compiler. Check the compiler manual.)

## 15. Option Function Select Area

### 15.1 Overview

The option function select area is used to select the MCU state after reset or the function to prevent rewriting in parallel I/O mode. The reset vector highest-order-address, 0FFFFh, is assigned as the option function select area. Figure 15.1 shows the Option Function Select Area.

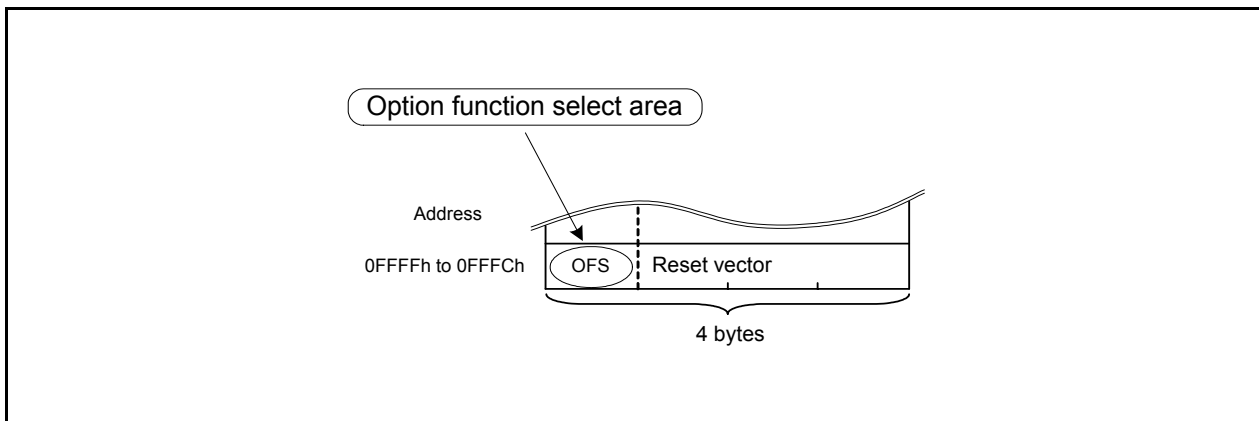


Figure 15.1 Option Function Select Area

## 15.2 OFS Register

The OFS register is used to select the MCU state after reset or the function to prevent rewriting in parallel I/O mode. Figure 15.2 shows the OFS Register.

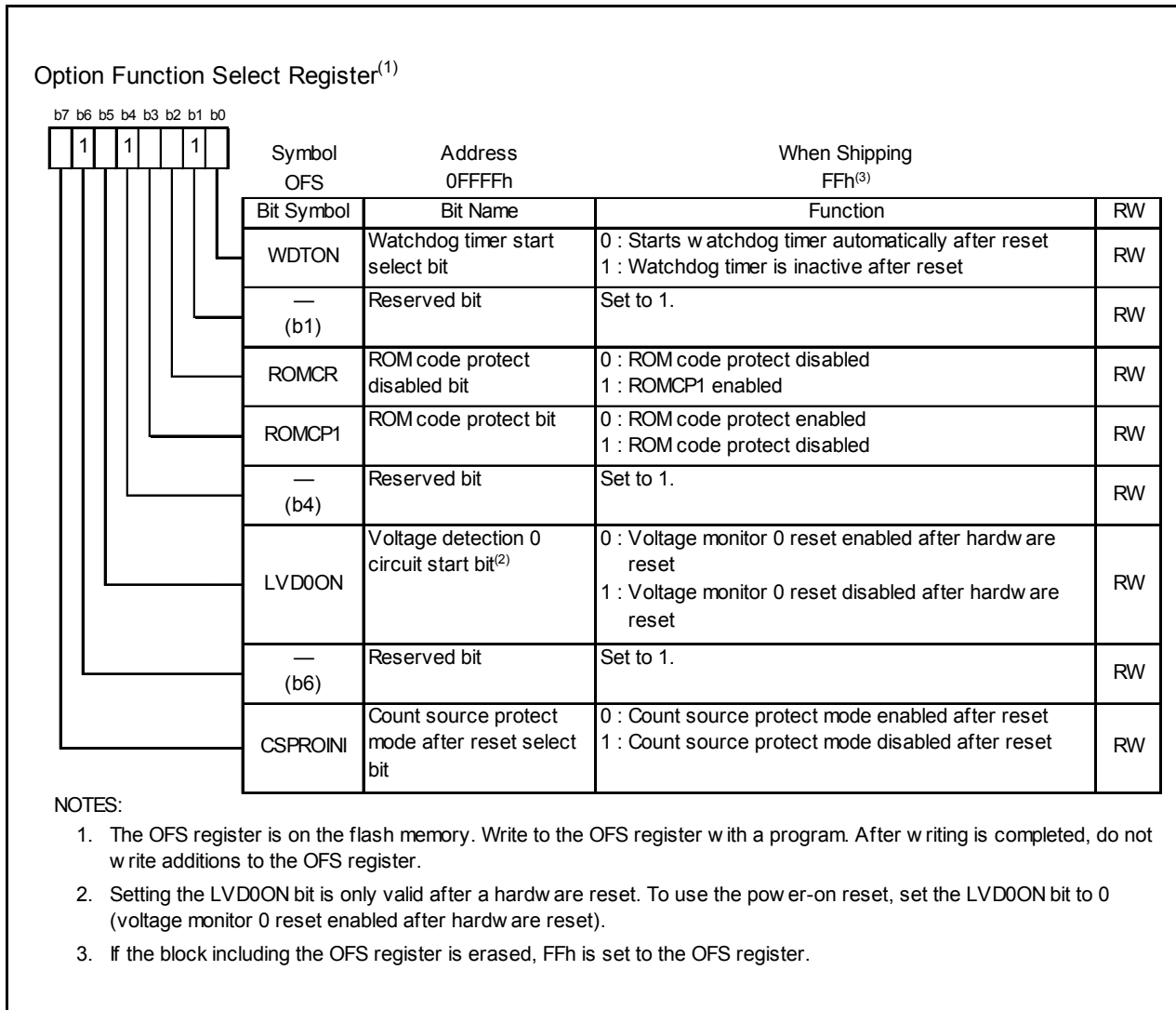


Figure 15.2 OFS Register

## 15.3 Notes on Option Function Select Area

### 15.3.1 Setting Example of Option Function Select Area

As the option function select area is allocated in the flash memory (not in the SFRs), they cannot be rewritten by executing an instruction. Write appropriate values when creating a program. The following shows a setting example.

- To set FFh in the OFS register

```
.org 00FFFCH
```

```
.lword reset | (0FF00000h) ; RESET
```

(Programming formats vary depending on the compiler. Check the compiler manual.)

## 16. Watchdog Timer

The watchdog timer is a function that detects when a program is out of control. Use of the watchdog timer is recommended to improve the reliability of the system. The watchdog timer contains a 15-bit counter and allows selection of count source protection mode enable or disable.

Table 16.1 lists information on the Watchdog Timer Specifications.

Refer to **5.6 Watchdog Timer Reset** for details on the watchdog timer.

Figure 16.1 shows the Block Diagram of Watchdog Timer. Figure 16.2 shows the Registers WDTR, WDTS, and WDC. Figure 16.3 shows the Registers CSPR and OFS.

**Table 16.1 Watchdog Timer Specifications**

Item	Count Source Protection Mode Disabled		Count Source Protection Mode Enabled
	Count source	CPU clock	XCIN clock divided by 32 (fC32)
Count operation	Decrement		
Count start condition	Either of the following can be selected <ul style="list-style-type: none"> <li>• After reset, count starts automatically</li> <li>• Count starts by writing to WDTS register</li> </ul>		
Count stop condition	Stop mode, wait mode	Stop mode	None
Reset condition of watchdog timer	<ul style="list-style-type: none"> <li>• Reset</li> <li>• Write 00h to the WDTR register before writing FFh</li> <li>• Underflow</li> </ul>		
Operation at the time of underflow	Watchdog timer interrupt or watchdog timer reset		Watchdog timer reset
Select functions	<ul style="list-style-type: none"> <li>• Division ratio of prescaler (when select the CPU clock as the count source) Selected by the WDC7 bit in the WDC register</li> <li>• The default value of the watchdog timer (when select fC32 as the count source) Selected by bits CVS0 to CVS1 in the CSPR register</li> <li>• Count source protection mode Whether count source protection mode is enabled or disabled after a reset can be selected by the CSPROINI bit in the OFS register (flash memory). If count source protection mode is disabled after a reset, it can be enabled or disabled by the CSPRO bit in the CSPR register (program).</li> <li>• Starts or stops of the watchdog timer after a reset Selected by the WDTON bit in the OFS register (flash memory).</li> </ul>		

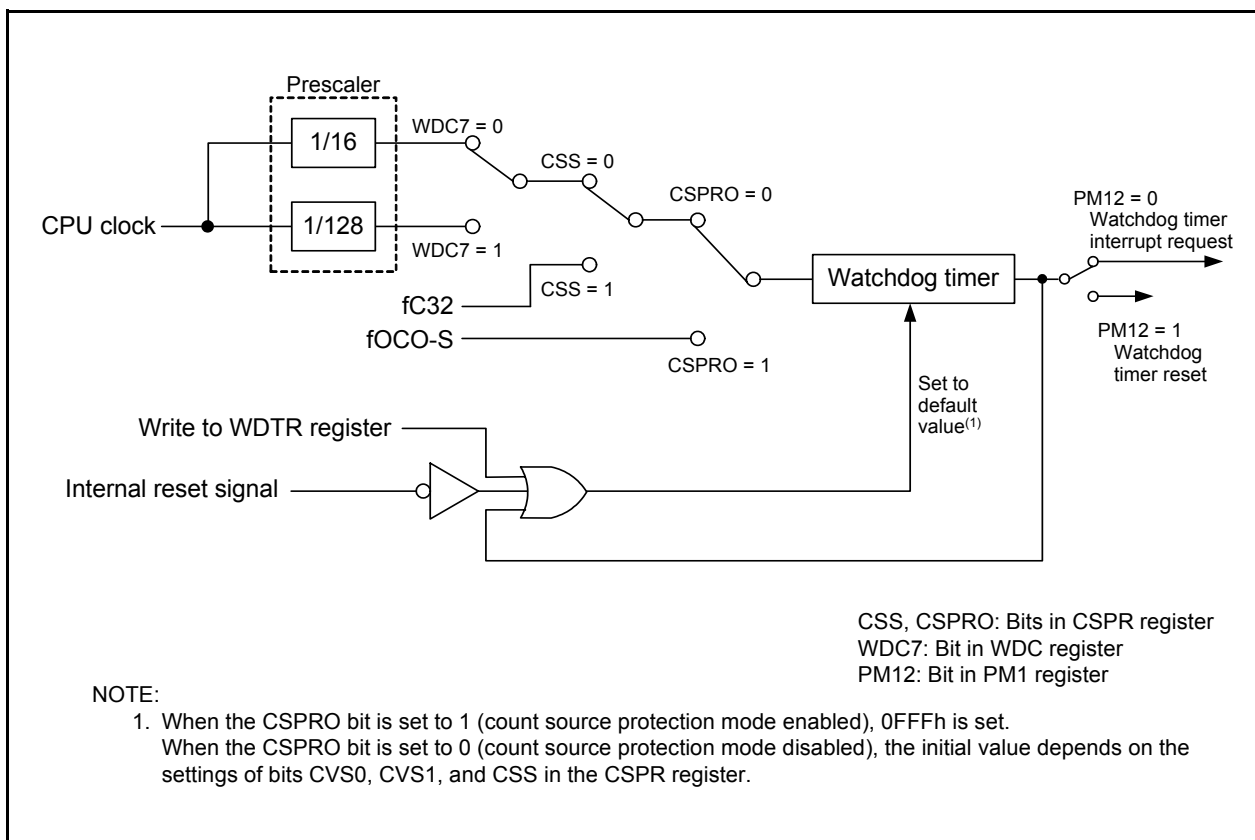


Figure 16.1 Block Diagram of Watchdog Timer

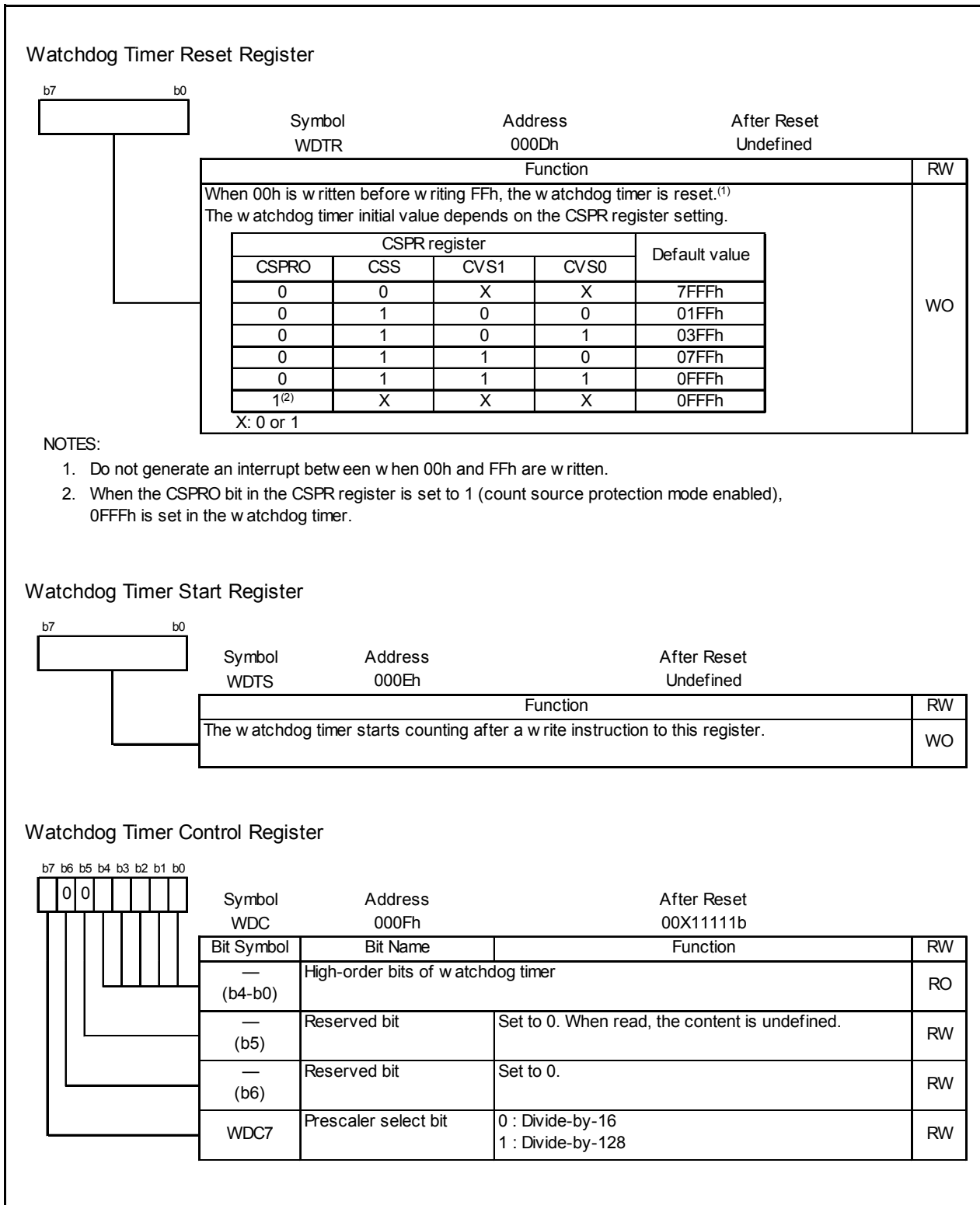


Figure 16.2 Registers WDTR, WDTS, and WDC

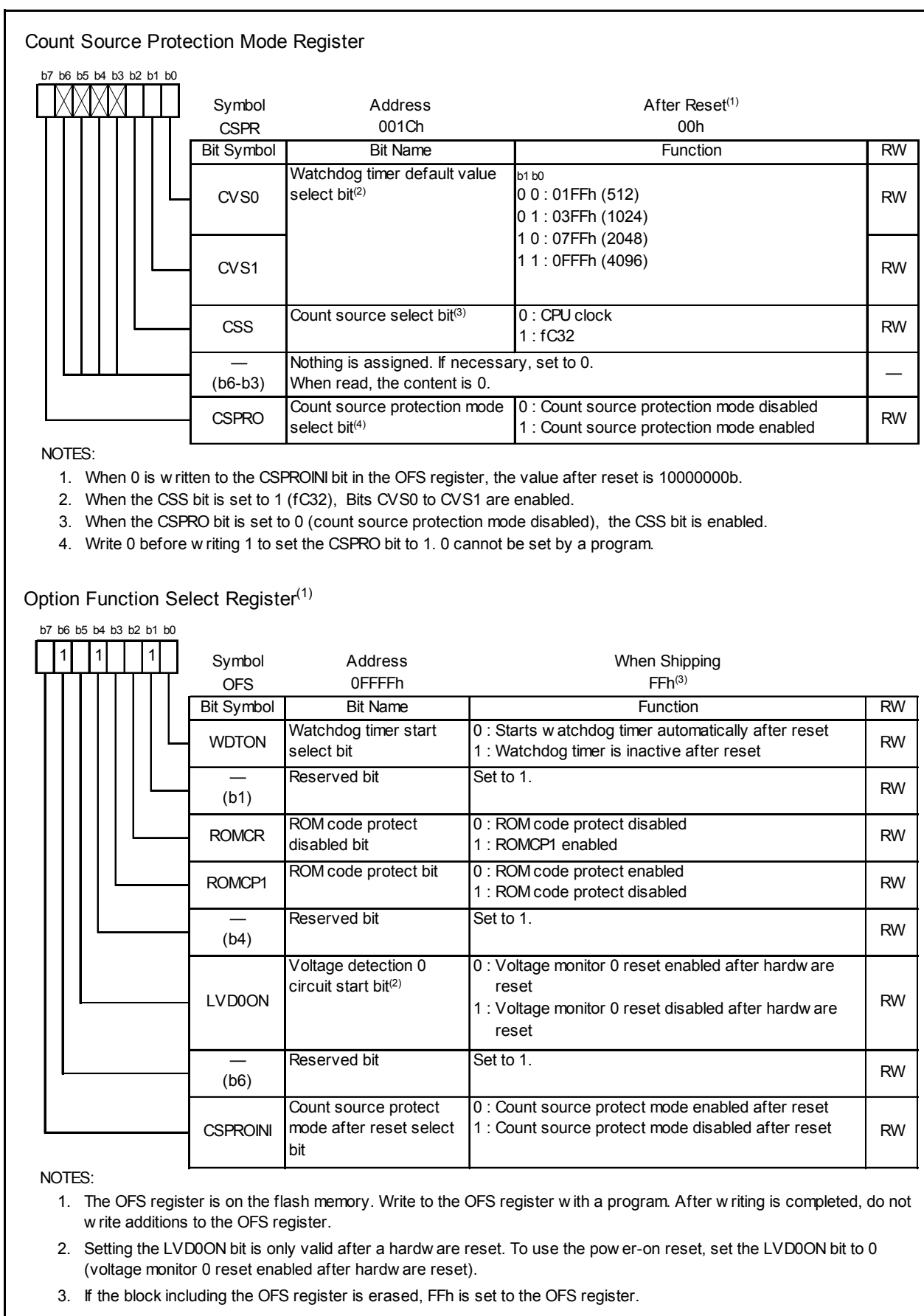


Figure 16.3 Registers CSPR and OFS



## 16.1 Count Source Protection Mode Disabled

The count source of the watchdog timer is either the CPU clock or the XCIN clock divided by 32 (fC32) can be selected when count source protection mode is disabled. fC32 does not stop in wait mode, the watchdog timer to count continues.

Table 16.2 lists the Watchdog Timer Specifications (with Count Source Protection Mode Disabled).

**Table 16.2 Watchdog Timer Specifications (with Count Source Protection Mode Disabled)**

Item	Specification	
Count source	CPU clock	XCIN clock divided by 32 (fC32)
Count operation	Decrement	
Period	$\frac{\text{Division ratio of prescaler (n)}}{\text{CPU clock}} \times \text{count value of watchdog timer (32768)}^{(1, 2)}$ n: 16 or 128 (selected by WDC7 bit in WDC register) Example: When the CPU clock frequency is 8 MHz and prescaler divided by 16, the period is approximately 65.5 ms	$\frac{32}{\text{XCIN clock}} \times \text{count value of watchdog timer (m)}^{(1)}$ m: 512, 1024, 2048 or 4096 (selected by bits CVS0 to CVS1 in the CSPR register) Example: When the XCIN clock frequency is 32.768 kHz and the count value by 512, the period is 0.5 s
Reset condition of watchdog timer	<ul style="list-style-type: none"> <li>Reset</li> <li>Write 00h to the WDTR register before writing FFh</li> <li>Underflow</li> </ul>	
Count start condition	The WDTON bit <sup>(3)</sup> in the OFS register (0FFFFh) selects the operation of the watchdog timer after a reset <ul style="list-style-type: none"> <li>When the WDTON bit is set to 1 (watchdog timer is in stop state after reset) The watchdog timer and prescaler stop after a reset and the count starts when the WDTS register is written to</li> <li>When the WDTON bit is set to 0 (watchdog timer starts automatically after exiting) The watchdog timer and prescaler start counting automatically after a reset</li> </ul>	
Count stop condition	Stop and wait modes (inherit the count from the held value after exiting modes)	Stop mode (inherit the count from the held value after exiting modes)
Operation at time of underflow	<ul style="list-style-type: none"> <li>When the PM12 bit in the PM1 register is set to 0 Watchdog timer interrupt</li> <li>When the PM12 bit in the PM1 register is set to 1 Watchdog timer reset (refer to <b>5.6 Watchdog Timer Reset</b>)</li> </ul>	

### NOTES:

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

## 16.2 Count Source Protection Mode Enabled

The count source of the watchdog timer is the low-speed on-chip oscillator clock when count source protection mode is enabled. If the CPU clock stops when a program is out of control, the clock can still be supplied to the watchdog timer.

Table 16.3 lists the Watchdog Timer Specifications (with Count Source Protection Mode Enabled).

**Table 16.3 Watchdog Timer Specifications (with Count Source Protection Mode Enabled)**

Item	Specification
Count source	Low-speed on-chip oscillator clock
Count operation	Decrement
Period	Count value of watchdog timer (4096) Low-speed on-chip oscillator clock Example: Period is approximately 32.8 ms when the low-speed on-chip oscillator clock frequency is 125 kHz
Reset condition of watchdog timer	<ul style="list-style-type: none"> <li>• Reset</li> <li>• Write 00h to the WDTR register before writing FFh</li> <li>• Underflow</li> </ul>
Count start condition	<p>The WDTON bit<sup>(1)</sup> in the OFS register (0FFFFh) selects the operation of the watchdog timer after a reset.</p> <ul style="list-style-type: none"> <li>• When the WDTON bit is set to 1 (watchdog timer is in stop state after reset) The watchdog timer and prescaler stop after a reset and the count starts when the WDTS register is written to</li> <li>• When the WDTON bit is set to 0 (watchdog timer starts automatically after reset) The watchdog timer and prescaler start counting automatically after a reset</li> </ul>
Count stop condition	None (The count does not stop in wait mode after the count starts. The MCU does not enter stop mode.)
Operation at time of underflow	Watchdog timer reset (refer to <b>5.6 Watchdog Timer Reset</b> )
Registers, bits	<ul style="list-style-type: none"> <li>• When setting the CSPPRO bit in the CSPR register to 1 (count source protection mode is enabled)<sup>(2)</sup>, the following are set automatically <ul style="list-style-type: none"> <li>- Set 0FFFh to the watchdog timer</li> <li>- Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on)</li> <li>- Set the PM12 bit in the PM1 register to 1 (The watchdog timer is reset when watchdog timer underflows)</li> </ul> </li> <li>• The following conditions apply in count source protection mode <ul style="list-style-type: none"> <li>- Writing to the CM10 bit in the CM1 register is disabled (It remains unchanged even if it is set to 1. The MCU does not enter stop mode.)</li> <li>- Writing to the CM14 bit in the CM1 register is disabled (It remains unchanged even if it is set to 1. The low-speed on-chip oscillator does not stop.)</li> </ul> </li> </ul>

### NOTES:

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

## 17. Timers

The MCU has two 8-bit timers with 8-bit prescalers, one 16-bit timer, and a timer with a 4-bit counter and an 8-bit counter. The two 8-bit timers with 8-bit prescalers are timer RA and timer RB. These timers contain a reload register to store the default value of the counter. The one 16-bit timer is timer RF and have input capture and output compare functions. The 4-bit and 8-bit counters are timer RE, and has an output compare function. All the timers operate independently.

Table 17.1 lists Functional Comparison of Timers.

**Table 17.1 Functional Comparison of Timers**

Item		Timer RA	Timer RB	Timer RE	Timer RF
Configuration		8-bit timer with 8-bit prescaler (with reload register)	8-bit timer with 8-bit prescaler (with reload register)	4-bit counter 8-bit counter	16-bit timer (with input capture and output compare)
Count		Decrement	Decrement	Increment	Increment
Count sources		<ul style="list-style-type: none"> <li>• f1</li> <li>• f2</li> <li>• f8</li> <li>• fOCO</li> <li>• fC32</li> </ul>	<ul style="list-style-type: none"> <li>• f1</li> <li>• f2</li> <li>• f8</li> <li>• Timer RA underflow</li> </ul>	<ul style="list-style-type: none"> <li>• f4</li> <li>• f8</li> <li>• f32</li> <li>• fC4</li> </ul>	<ul style="list-style-type: none"> <li>• f1</li> <li>• f8</li> <li>• f32</li> </ul>
Function	Count of the internal count source	Timer mode	Timer mode	—	Output compare mode
	Count of the external count source	Event counter mode	—	—	—
	External pulse width/period measurement	Pulse width measurement mode, pulse period measurement mode	—	—	Input capture mode
	PWM output	Pulse output mode <sup>(1)</sup> , Event counter mode <sup>(1)</sup>	Programmable waveform generation mode	Output compare mode <sup>(1)</sup>	Output compare mode
	One-shot waveform output	—	Programmable one-shot generation mode, Programmable wait one-shot generation mode	—	—
	Timer	Timer mode (only fC32 count)	—	Real-time clock mode	—
Input pin		TRAIO	$\overline{\text{INT0}}$	—	TRFI
Output pin		TRAO TRAIO	TRBO	TREO	TRFO00 to TRFO02, TRFO10 to TRFO12
Related interrupt		Timer RA interrupt, $\overline{\text{INT1}}$ interrupt	Timer RB interrupt, $\overline{\text{INT0}}$ interrupt	Timer RE interrupt	Timer RF interrupt, Compare 0 interrupt, Compare 1 interrupt, Capture interrupt
Timer stop		Provided	Provided	Provided	Provided

## NOTE:

1. Rectangular waves are output in these modes. Since the waves are inverted at each overflow, the “H” and “L” level widths of the pulses are the same.

## 17.1 Timer RA

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

The prescaler and timer each consist of a reload register and counter. The reload register and counter are allocated at the same address, and can be accessed when accessing registers TRAPRE and TRA (refer to **Tables 17.2 to 17.6 the Specifications of Each Mode**).

The count source for timer RA is the operating clock that regulates the timing of timer operations such as counting and reloading.

Figure 17.1 shows a Block Diagram of Timer RA. Figures 17.2 and 17.3 show the registers associated with timer RA.

Timer RA has the following five operating modes:

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

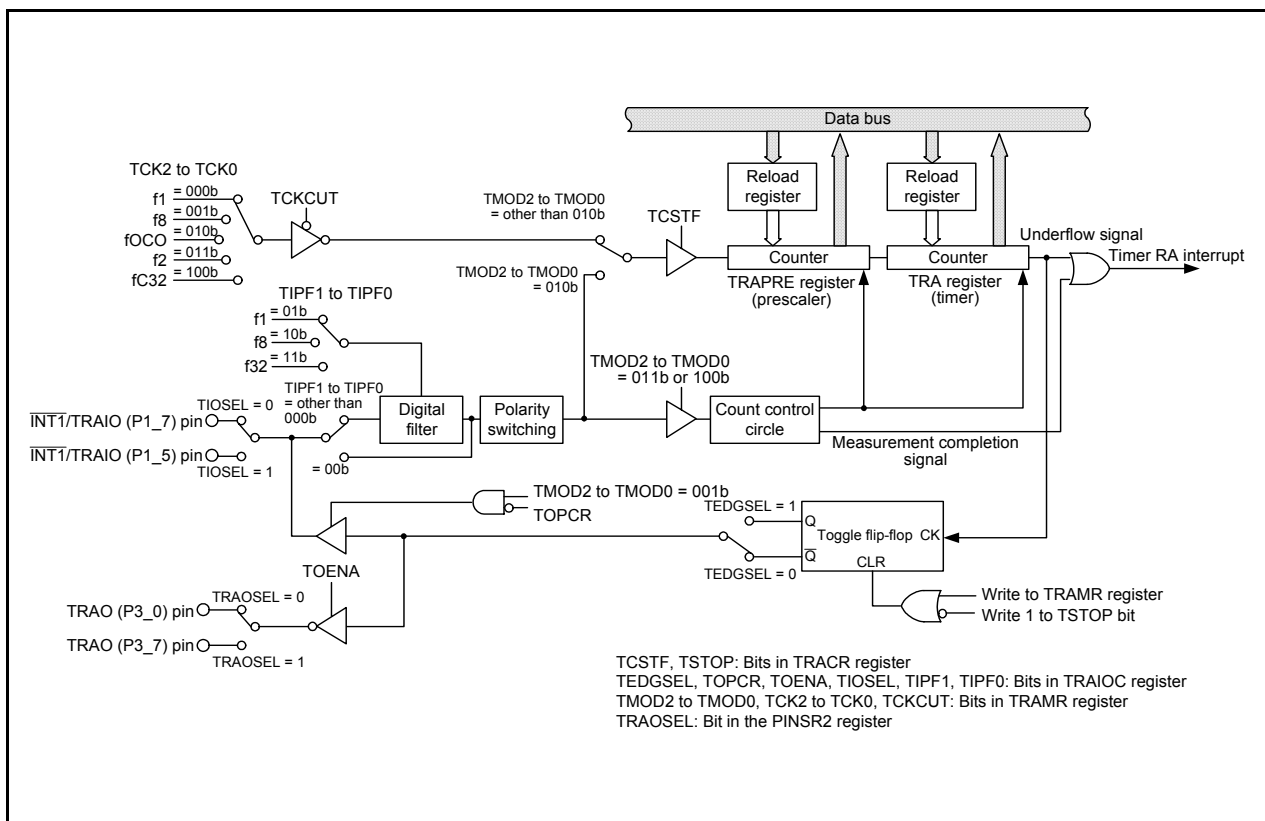


Figure 17.1 Block Diagram of Timer RA

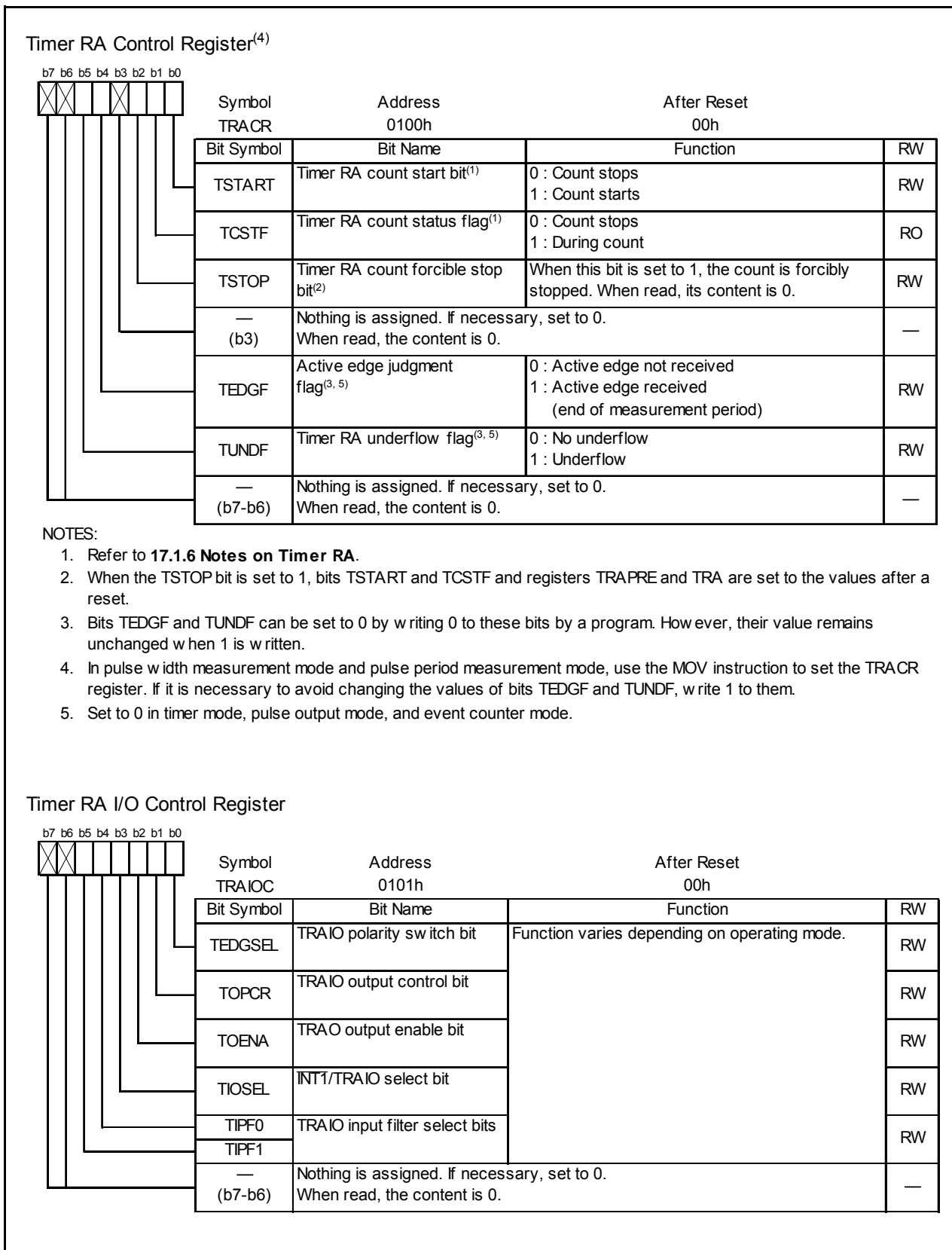


Figure 17.2 Registers TRACR and TRATIOC

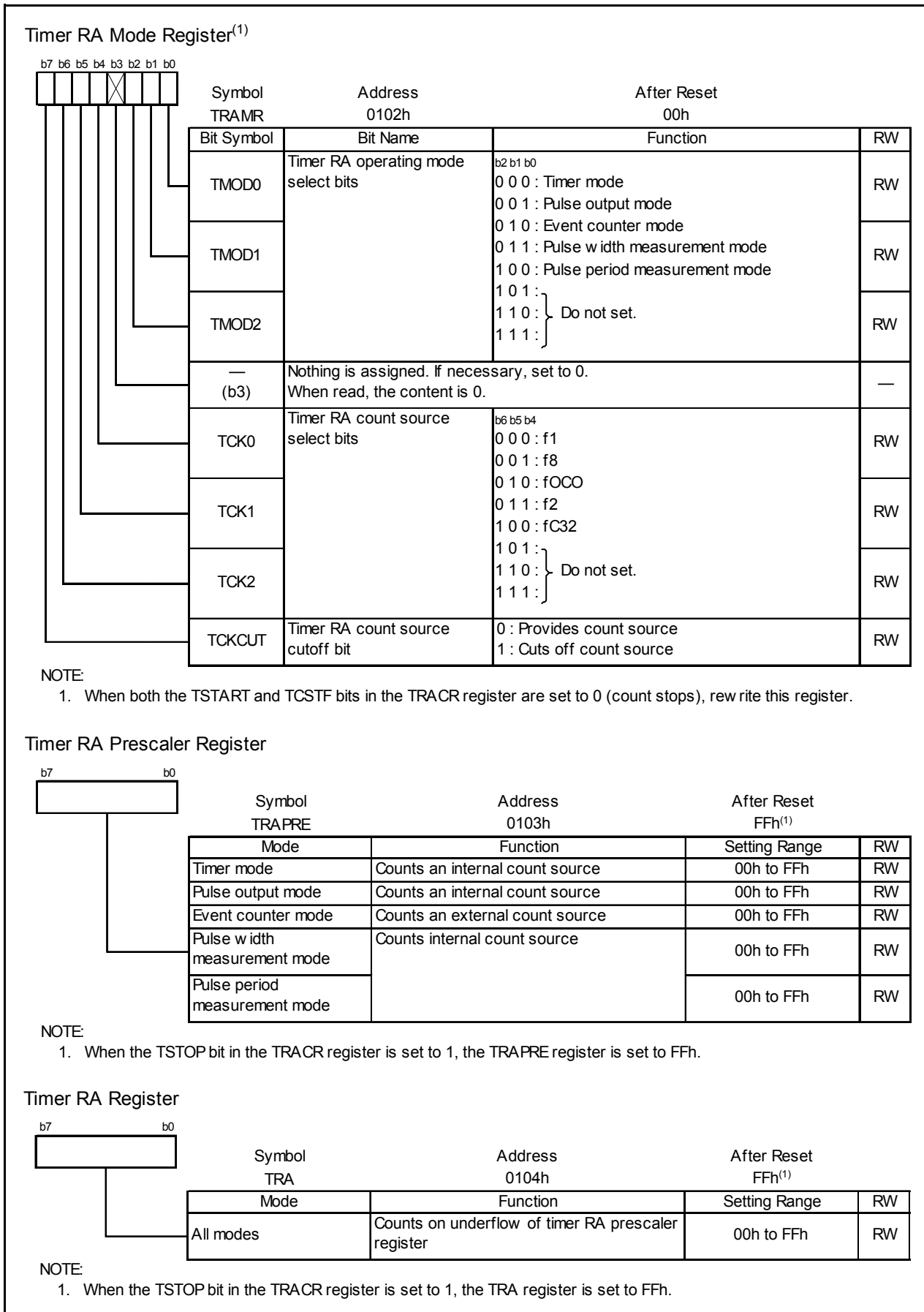


Figure 17.3 Registers TRAMR, TRAPRE, and TRA

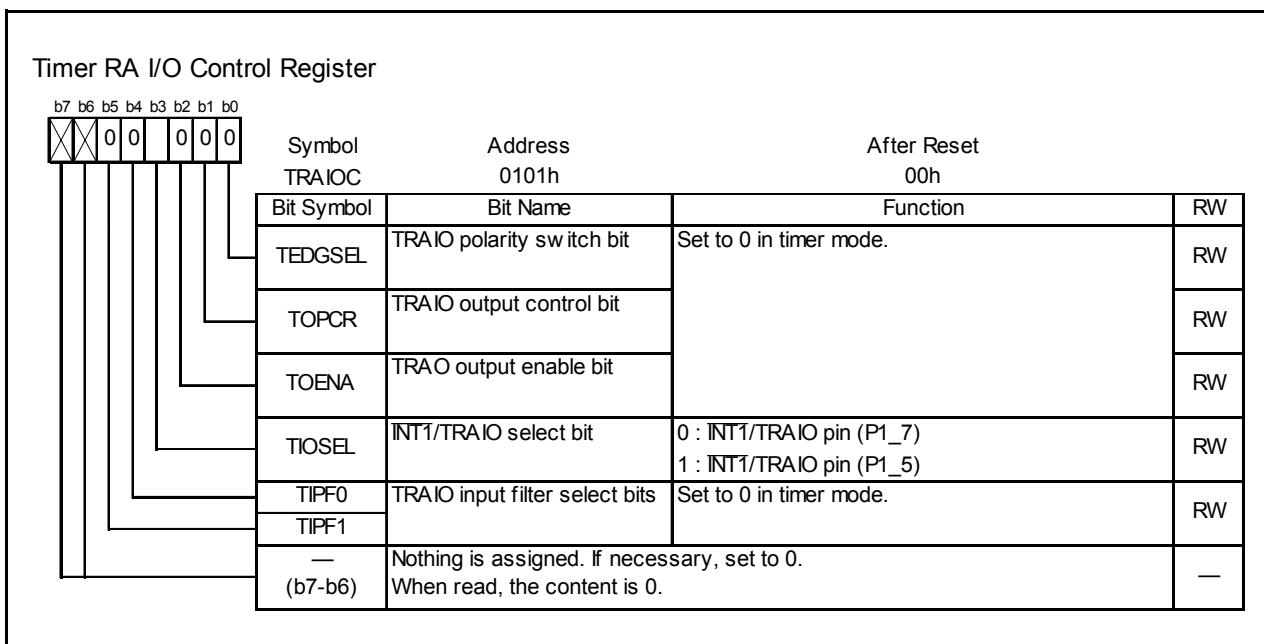
### 17.1.1 Timer Mode

In this mode, the timer counts an internally generated count source (refer to **Table 17.2 Timer Mode Specifications**).

Figure 17.4 shows TRAIOC Register in Timer Mode.

**Table 17.2 Timer Mode Specifications**

Item	Specification
Count sources	f1, f2, f8, fOCO, fC32
Count operations	<ul style="list-style-type: none"> <li>Decrement</li> <li>When the timer underflows, the contents of the reload register are reloaded and the count is continued.</li> </ul>
Divide ratio	$1/(n+1)(m+1)$ n: Value set in TRAPRE register, m: Value set in TRA register
Count start condition	1 (count starts) is written to the TSTART bit in the TRACR register.
Count stop conditions	<ul style="list-style-type: none"> <li>0 (count stops) is written to the TSTART bit in the TRACR register.</li> <li>1 (count forcibly stops) is written to the TSTOP bit in the TRACR register.</li> </ul>
Interrupt request generation timing	When timer RA underflows [timer RA interrupt].
INT1/TRAI0 pin function	Programmable I/O port, or INT1 interrupt input
TRAO pin function	Programmable I/O port
Read from timer	The count value can be read by reading registers TRA and TRAPRE.
Write to timer	<ul style="list-style-type: none"> <li>When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter.</li> <li>When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter (refer to <b>17.1.1.1 Timer Write Control during Count Operation</b>).</li> </ul>



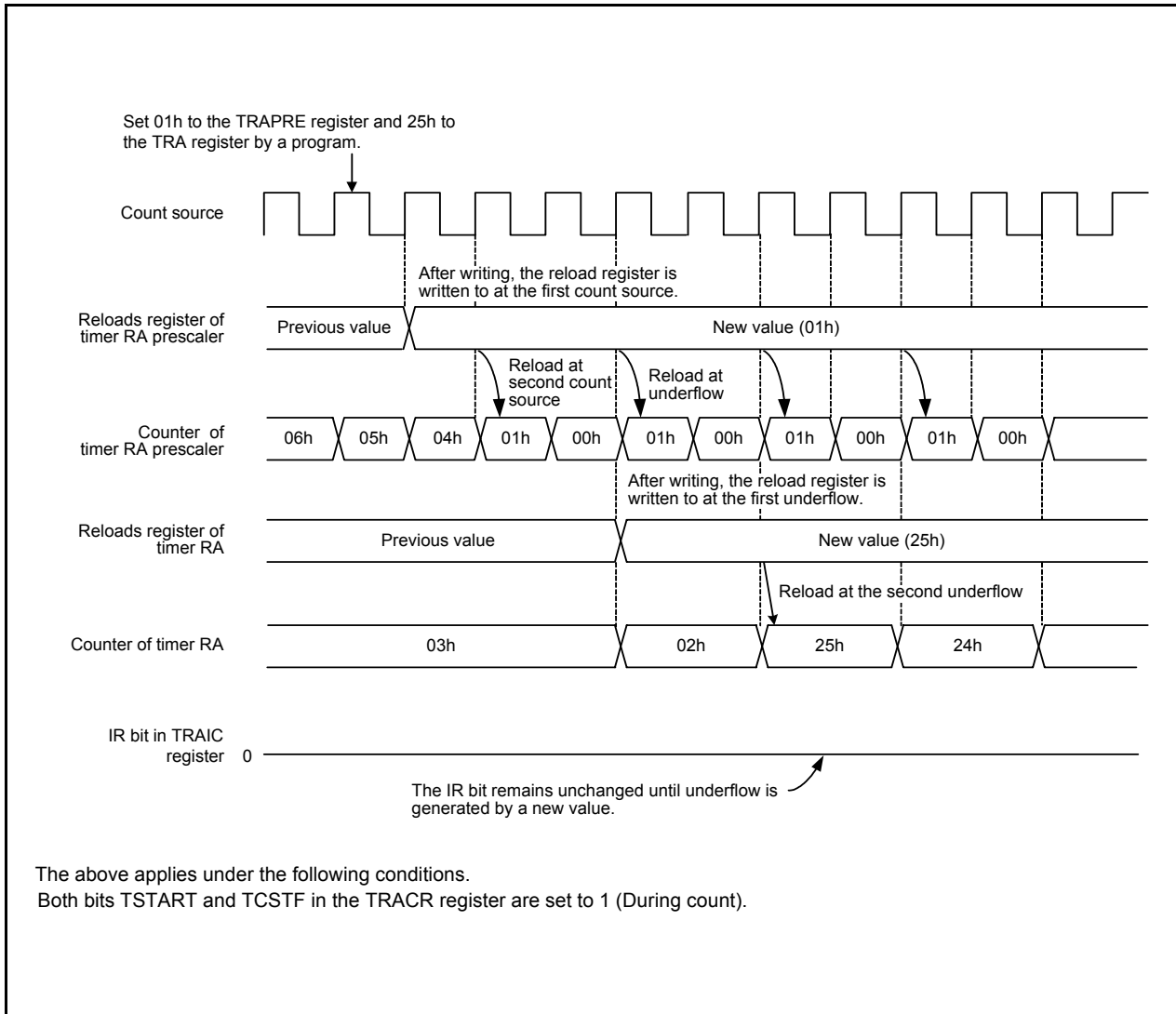
**Figure 17.4 TRAIOC Register in Timer Mode**



### 17.1.1.1 Timer Write Control during Count Operation

Timer RA has a prescaler and a timer (which counts the prescaler underflows). The prescaler and timer each consist of a reload register and a counter. When writing to the prescaler or timer, values are written to both the reload register and counter.

However, values are transferred from the reload register to the counter of the prescaler in synchronization with the count source. In addition, values are transferred from the reload register to the counter of the timer in synchronization with prescaler underflows. Therefore, if the prescaler or timer is written to when count operation is in progress, the counter value is not updated immediately after the WRITE instruction is executed. Figure 17.5 shows an Operating Example of Timer RA when Counter Value is Rewritten during Count Operation.



**Figure 17.5 Operating Example of Timer RA when Counter Value is Rewritten during Count Operation**

### 17.1.2 Pulse Output Mode

In pulse output mode, the internally generated count source is counted, and a pulse with inverted polarity is output from the TRAI0 pin each time the timer underflows (refer to **Table 17.3 Pulse Output Mode Specifications**).

Figure 17.6 shows TRAI0C Register in Pulse Output Mode.

**Table 17.3 Pulse Output Mode Specifications**

Item	Specification
Count sources	f1, f2, f8, fOCO, fC32
Count operations	<ul style="list-style-type: none"> <li>• Decrement</li> <li>• When the timer underflows, the contents in the reload register is reloaded and the count is continued.</li> </ul>
Divide ratio	$1/(n+1)(m+1)$ n: Value set in TRAPRE register, m: Value set in TRA register
Count start condition	1 (count starts) is written to the TSTART bit in the TRACR register.
Count stop conditions	<ul style="list-style-type: none"> <li>• 0 (count stops) is written to the TSTART bit in the TRACR register.</li> <li>• 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register.</li> </ul>
Interrupt request generation timing	When timer RA underflows [timer RA interrupt].
INT1/TRAI0 pin function	Pulse output, programmable output port, or $\overline{\text{INT1}}$ interrupt <sup>(1)</sup>
TRAO pin function	Programmable I/O port or inverted output of TRAI0 <sup>(1)</sup>
Read from timer	The count value can be read by reading registers TRA and TRAPRE.
Write to timer	<ul style="list-style-type: none"> <li>• When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter.</li> <li>• When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter (refer to <b>17.1.1.1 Timer Write Control during Count Operation</b>).</li> </ul>
Select functions	<ul style="list-style-type: none"> <li>• TRAI0 output polarity switch function The TEDGSEL bit in the TRAI0C register selects the level at the start of pulse output.<sup>(1)</sup></li> <li>• TRAO output function Pulses inverted from the TRAI0 output polarity can be output from the TRAO pin (selectable by the TOENA bit in the TRAI0C register).</li> <li>• TRAO pin select function P3_0 or P3_7 is selected by the TRAOSEL bit in the PINSR2 register.</li> <li>• Pulse output stop function Output from the TRAI0 pin is stopped by the TOPCR bit in the TRAI0C register.</li> <li>• <math>\overline{\text{INT1}}</math>/TRAI0 pin select function P1_7 or P1_5 is selected by the TIOSEL bit in the TRAI0C register.</li> </ul>

**NOTE:**

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

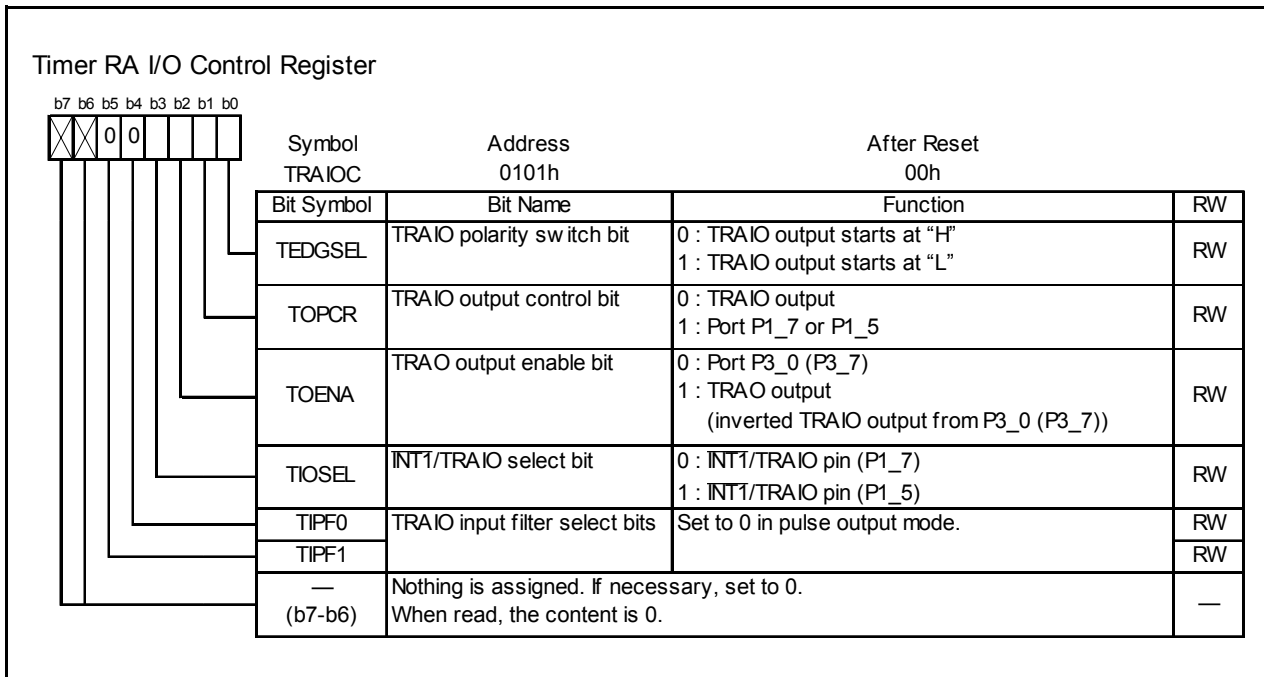


Figure 17.6 TRAI0C Register in Pulse Output Mode

### 17.1.3 Event Counter Mode

In event counter mode, external signal inputs to the  $\overline{\text{INT1}}$ /TRAIO pin are counted (refer to **Table 17.4 Event Counter Mode Specifications**).

Figure 17.7 shows TRAIOC Register in Event Counter Mode.

**Table 17.4 Event Counter Mode Specifications**

Item	Specification
Count source	External signal which is input to TRAIO pin (active edge selectable by a program)
Count operations	<ul style="list-style-type: none"> <li>• Decrement</li> <li>• When the timer underflows, the contents of the reload register are reloaded and the count is continued.</li> </ul>
Divide ratio	$1/(n+1)(m+1)$ n: setting value of TRAPRE register, m: setting value of TRA register
Count start condition	1 (count starts) is written to the TSTART bit in the TRACR register.
Count stop conditions	<ul style="list-style-type: none"> <li>• 0 (count stops) is written to the TSTART bit in the TRACR register.</li> <li>• 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register.</li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When timer RA underflows [timer RA interrupt].</li> </ul>
$\overline{\text{INT1}}$ /TRAIO pin function	Count source input ( $\overline{\text{INT1}}$ interrupt input)
TRAO pin function	Programmable I/O port or pulse output <sup>(1)</sup>
Read from timer	The count value can be read by reading registers TRA and TRAPRE.
Write to timer	<ul style="list-style-type: none"> <li>• When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter.</li> <li>• When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter (refer to <b>17.1.1.1 Timer Write Control during Count Operation</b>).</li> </ul>
Select functions	<ul style="list-style-type: none"> <li>• <math>\overline{\text{NT1}}</math> input polarity switch function The TEDGSEL bit in the TRAIOC register selects the active edge of the count source.</li> <li>• Count source input pin select function P1_7 or P1_5 is selected by the TIOSEL bit in the TRAIOC register.</li> <li>• Pulse output function Pulses of inverted polarity can be output from the TRAO pin each time the timer underflows (selectable by the TOENA bit in the TRAIOC register).<sup>(1)</sup></li> <li>• TRAO pin select function P3_0 or P3_7 is selected by the TRAOSEL bit in the PINSR2 register.</li> <li>• Digital filter function Bits TIPF0 and TIPF1 in the TRAIOC register enable or disable the digital filter and select the sampling frequency.</li> </ul>

NOTE:

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

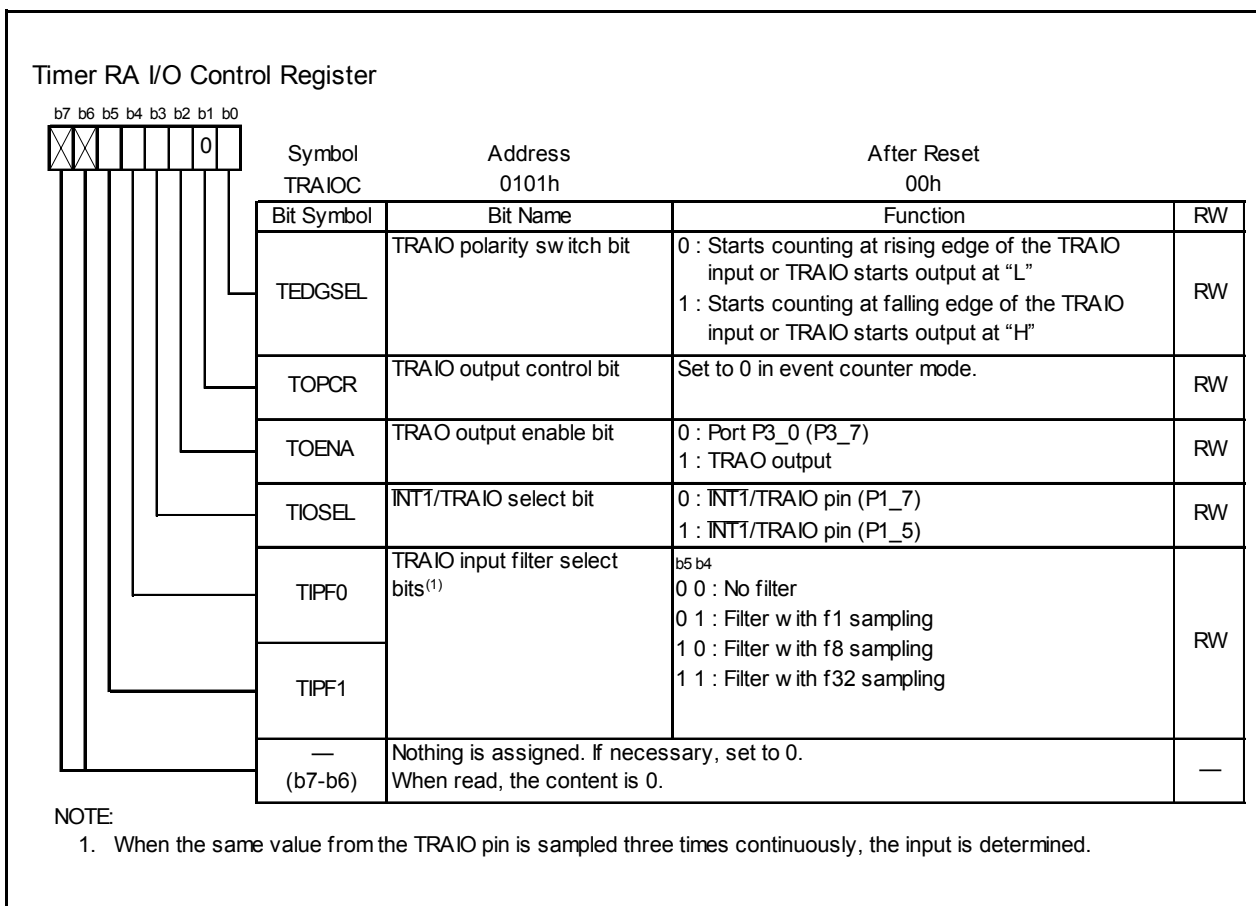


Figure 17.7 TRAI0C Register in Event Counter Mode

### 17.1.4 Pulse Width Measurement Mode

In pulse width measurement mode, the pulse width of an external signal input to the  $\overline{\text{INT1}}$ /TRAIO pin is measured (refer to **Table 17.5 Pulse Width Measurement Mode Specifications**).

Figure 17.8 shows TRAI0C Register in Pulse Width Measurement Mode and Figure 17.9 shows an Operating Example of Pulse Width Measurement Mode.

**Table 17.5 Pulse Width Measurement Mode Specifications**

Item	Specification
Count sources	f1, f2, f8, fOCO, fC32
Count operations	<ul style="list-style-type: none"> <li>• Decrement</li> <li>• Continuously counts the selected signal only when measurement pulse is “H” level, or conversely only “L” level.</li> <li>• When the timer underflows, the contents of the reload register are reloaded and the count is continued.</li> </ul>
Count start condition	1 (count starts) is written to the TSTART bit in the TRACR register.
Count stop conditions	<ul style="list-style-type: none"> <li>• 0 (count stops) is written to the TSTART bit in the TRACR register.</li> <li>• 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register.</li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When timer RA underflows [timer RA interrupt].</li> <li>• Rising or falling of the TRAIO input (end of measurement period) [timer RA interrupt]</li> </ul>
$\overline{\text{INT1}}$ /TRAIO pin function	Measured pulse input ( $\overline{\text{INT1}}$ interrupt input)
TRAO pin function	Programmable I/O port
Read from timer	The count value can be read by reading registers TRA and TRAPRE.
Write to timer	<ul style="list-style-type: none"> <li>• When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter.</li> <li>• When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter (refer to <b>17.1.1.1 Timer Write Control during Count Operation</b>).</li> </ul>
Select functions	<ul style="list-style-type: none"> <li>• Measurement level select</li> <li>• The TEDGSEL bit in the TRAI0C register selects the “H” or “L” level period.</li> <li>• Measured pulse input pin select function P1_7 or P1_5 is selected by the TIOSEL bit in the TRAI0C register.</li> <li>• Digital filter function Bits TIPF0 and TIPF1 in the TRAI0C register enable or disable the digital filter and select the sampling frequency.</li> </ul>

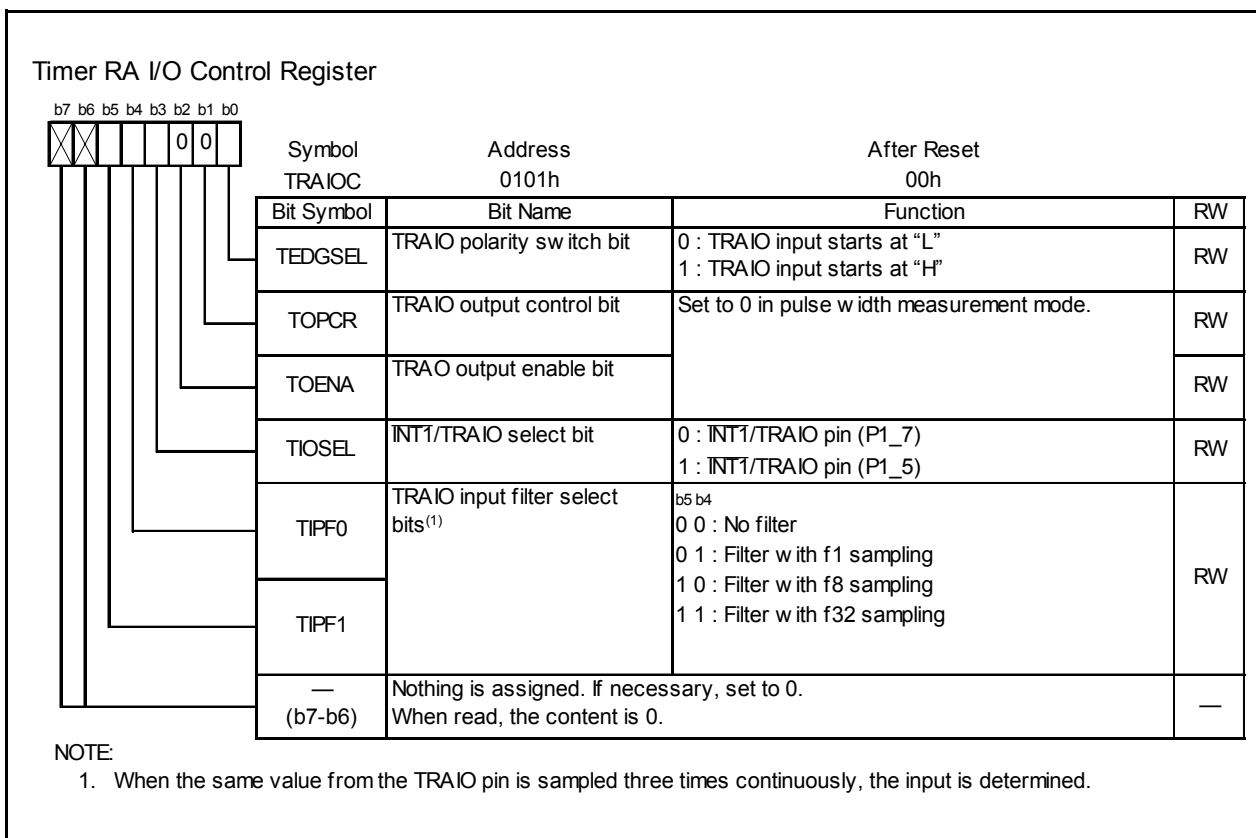


Figure 17.8 TRAI0C Register in Pulse Width Measurement Mode

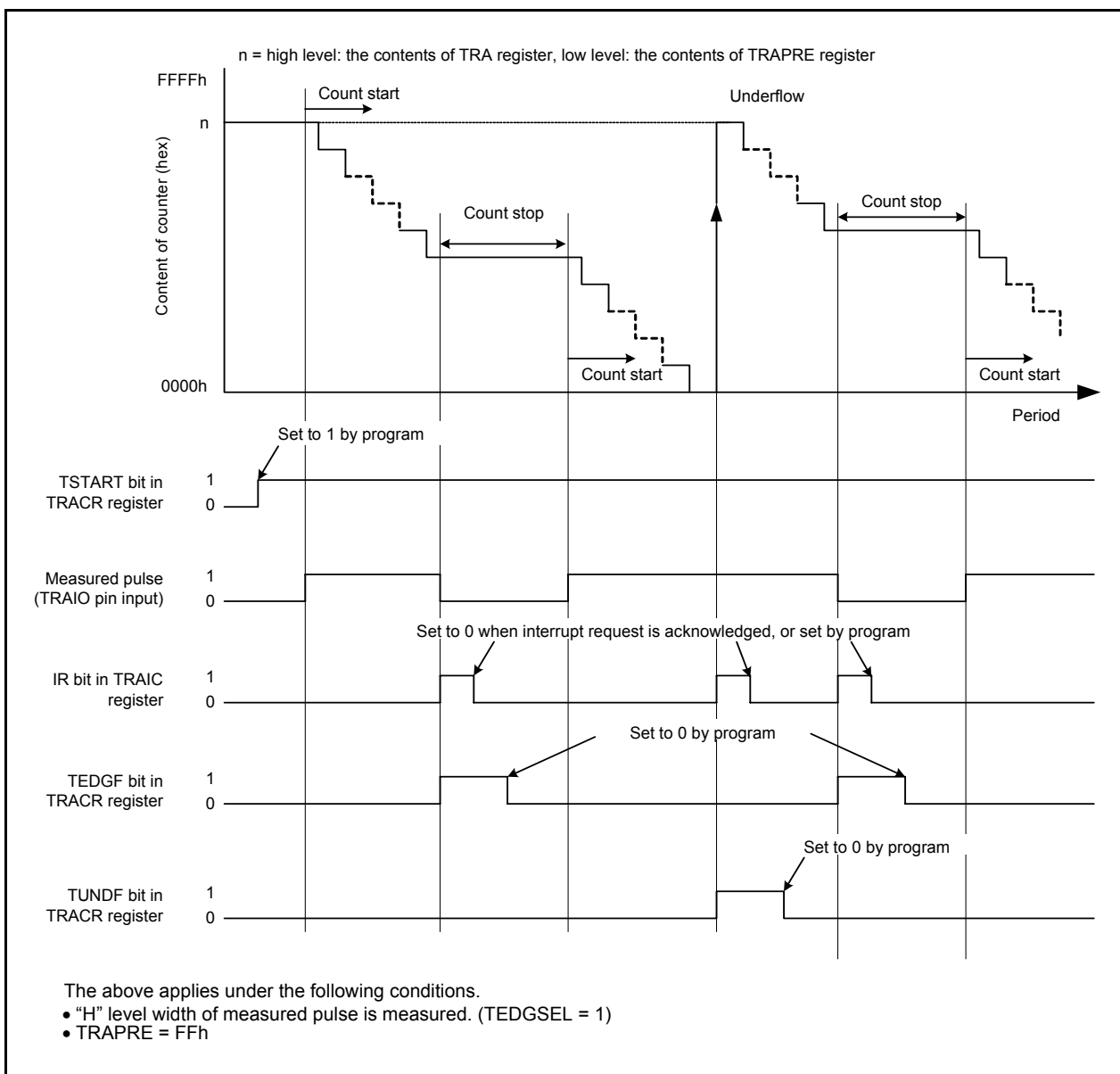


Figure 17.9 Operating Example of Pulse Width Measurement Mode



### 17.1.5 Pulse Period Measurement Mode

In pulse period measurement mode, the pulse period of an external signal input to the  $\overline{\text{INT1}}$ /TRAIO pin is measured (refer to **Table 17.6 Pulse Period Measurement Mode Specifications**).

Figure 17.10 shows TRAIOC Register in Pulse Period Measurement Mode and Figure 17.11 shows an Operating Example of Pulse Period Measurement Mode.

**Table 17.6 Pulse Period Measurement Mode Specifications**

Item	Specification
Count sources	f1, f2, f8, fOCO, fC32
Count operations	<ul style="list-style-type: none"> <li>• Decrement</li> <li>• After the active edge of the measured pulse is input, the contents of the read-out buffer are retained at the first underflow of timer RA prescaler. Then timer RA reloads the contents in the reload register at the second underflow of timer RA prescaler and continues counting.</li> </ul>
Count start condition	1 (count starts) is written to the TSTART bit in the TRACR register.
Count stop conditions	<ul style="list-style-type: none"> <li>• 0 (count stops) is written to TSTART bit in the TRACR register.</li> <li>• 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register.</li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When timer RA underflows or reloads [timer RA interrupt].</li> <li>• Rising or falling of the TRAIO input (end of measurement period) [timer RA interrupt]</li> </ul>
$\overline{\text{INT1}}$ /TRAIO pin function	Measured pulse input <sup>(1)</sup> ( $\overline{\text{INT1}}$ interrupt input)
TRAIO pin function	Programmable I/O port
Read from timer	The count value can be read by reading registers TRA and TRAPRE.
Write to timer	<ul style="list-style-type: none"> <li>• When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter.</li> <li>• When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter (refer to <b>17.1.1.1 Timer Write Control during Count Operation</b>).</li> </ul>
Select functions	<ul style="list-style-type: none"> <li>• Measurement period select The TEDGSEL bit in the TRAIOC register selects the measurement period of the input pulse.</li> <li>• Measured pulse input pin select function P1_7 or P1_5 is selected by the TIOSEL bit in the TRAIOC register.</li> <li>• Digital filter function Bits TIPF0 and TIPF1 in the TRAIOC register enable or disable the digital filter and select the sampling frequency.</li> </ul>

**NOTE:**

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

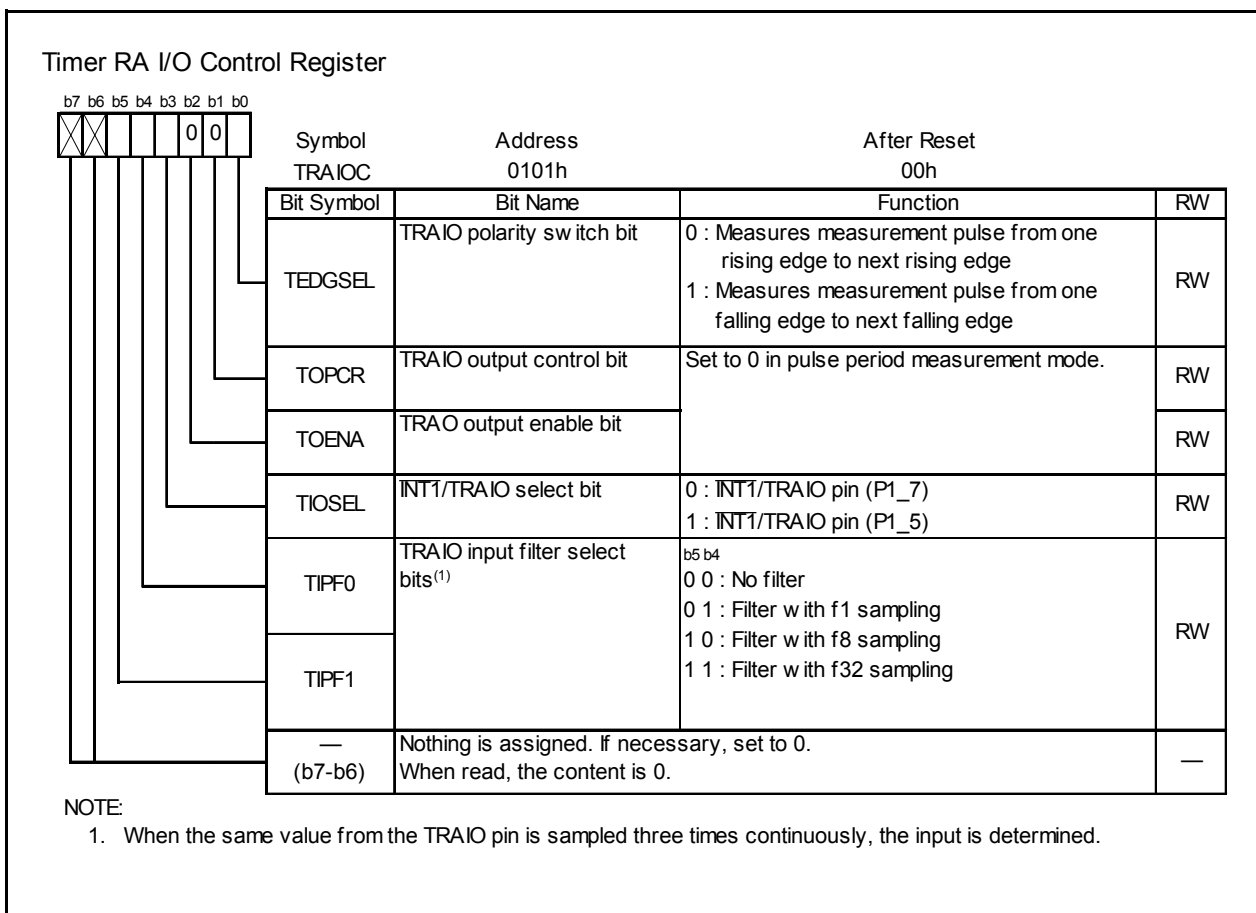


Figure 17.10 TRAI OC Register in Pulse Period Measurement Mode

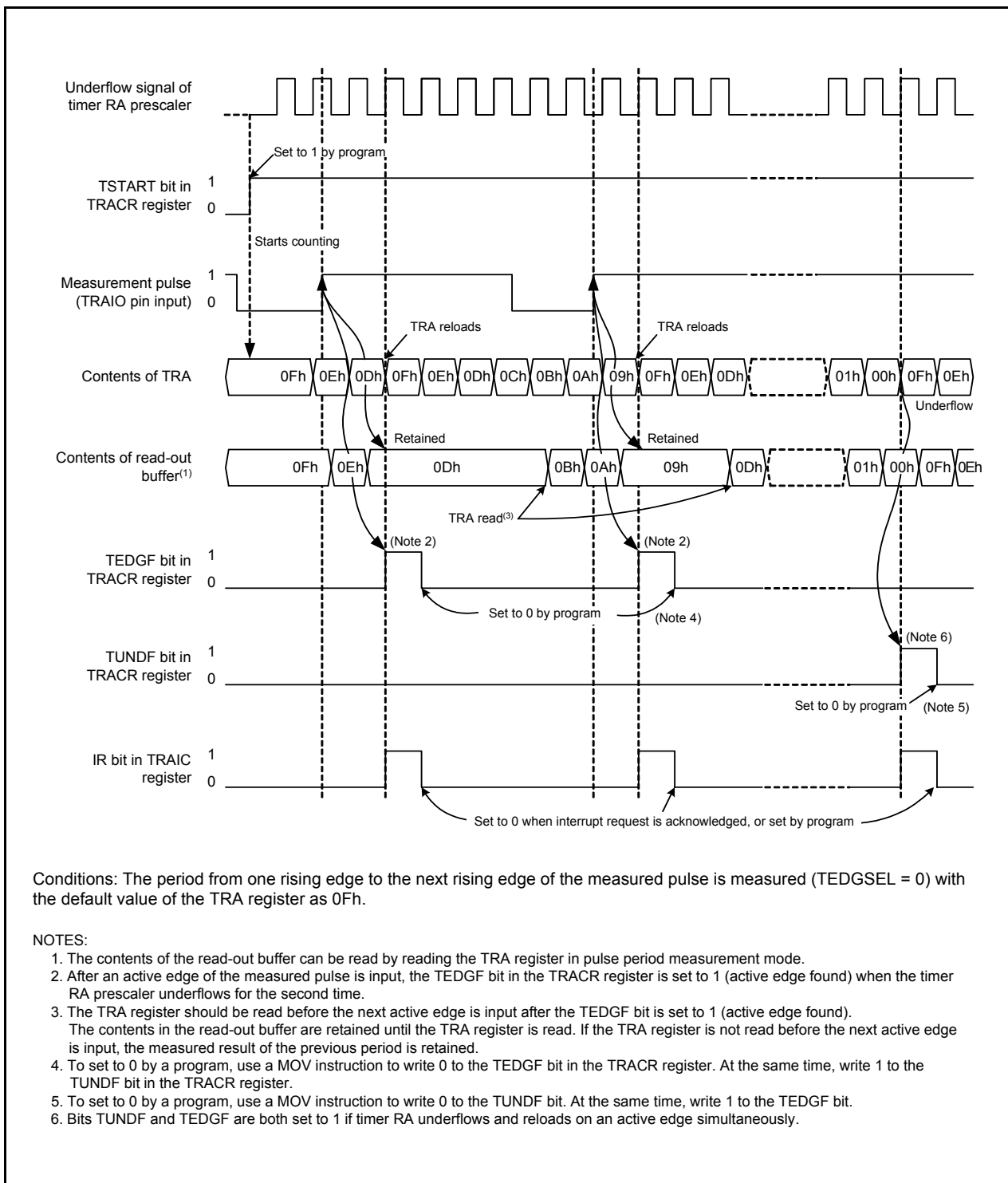


Figure 17.11 Operating Example of Pulse Period Measurement Mode

### 17.1.6 Notes on Timer RA

- Timer RA stops counting after a reset. Set the values in the timer RA and timer RA prescalers before the count starts.
- Even if the prescaler and timer RA are read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- In pulse period measurement mode, bits TEDGF and TUNDF in the TRACR register can be set to 0 by writing 0 to these bits by a program. However, these bits remain unchanged if 1 is written. When using the READ-MODIFY-WRITE instruction for the TRACR register, the TEDGF or TUNDF bit may be set to 0 although these bits are set to 1 while the instruction is being executed. In this case, write 1 to the TEDGF or TUNDF bit which is not supposed to be set to 0 with the MOV instruction.
- When changing to pulse period measurement mode from another mode, the contents of bits TEDGF and TUNDF are undefined. Write 0 to bits TEDGF and TUNDF before the count starts.
- The TEDGF bit may be set to 1 by the first timer RA prescaler underflow generated after the count starts.
- When using the pulse period measurement mode, leave two or more periods of the timer RA prescaler immediately after the count starts, then set the TEDGF bit to 0.
- The TCSTF bit retains 0 (count stops) for 0 to 1 cycle of the count source after setting the TSTART bit to 1 (count starts) while the count is stopped.

During this time, do not access registers associated with timer RA<sup>(1)</sup> other than the TCSTF bit. Timer RA starts counting at the first valid edge of the count source after The TCSTF bit is set to 1 (during count). The TCSTF bit remains 1 for 0 to 1 cycle of the count source after setting the TSTART bit to 0 (count stops) while the count is in progress. Timer RA counting is stopped when the TCSTF bit is set to 0. During this time, do not access registers associated with timer RA<sup>(1)</sup> other than the TCSTF bit.

NOTE:

1. Registers associated with timer RA: TRACR, TRAIOC, TRAMR, TRAPRE, and TRA.

- When the TRAPRE register is continuously written during count operation (TCSTF bit is set to 1), allow three or more cycles of the count source clock for each write interval.
- When the TRA register is continuously written during count operation (TCSTF bit is set to 1), allow three or more cycles of the prescaler underflow for each write interval.

17.2 Timer RB

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

The prescaler and timer each consist of a reload register and counter (refer to **Tables 17.7 to 17.10 the Specifications of Each Mode**).

Timer RB has timer RB primary and timer RB secondary as reload registers.

The count source for timer RB is the operating clock that regulates the timing of timer operations such as counting and reloading.

Figure 17.12 shows a Block Diagram of Timer RB. Figures 17.13 to 17.15 show the registers associated with timer RB.

Timer RB has four operation modes listed as follows:

- Timer mode: The timer counts an internal count source (peripheral function clock or timer RA underflows).
- Programmable waveform generation mode: The timer outputs pulses of a given width successively.
- Programmable one-shot generation mode: The timer outputs a one-shot pulse.
- Programmable wait one-shot generation mode: The timer outputs a delayed one-shot pulse.

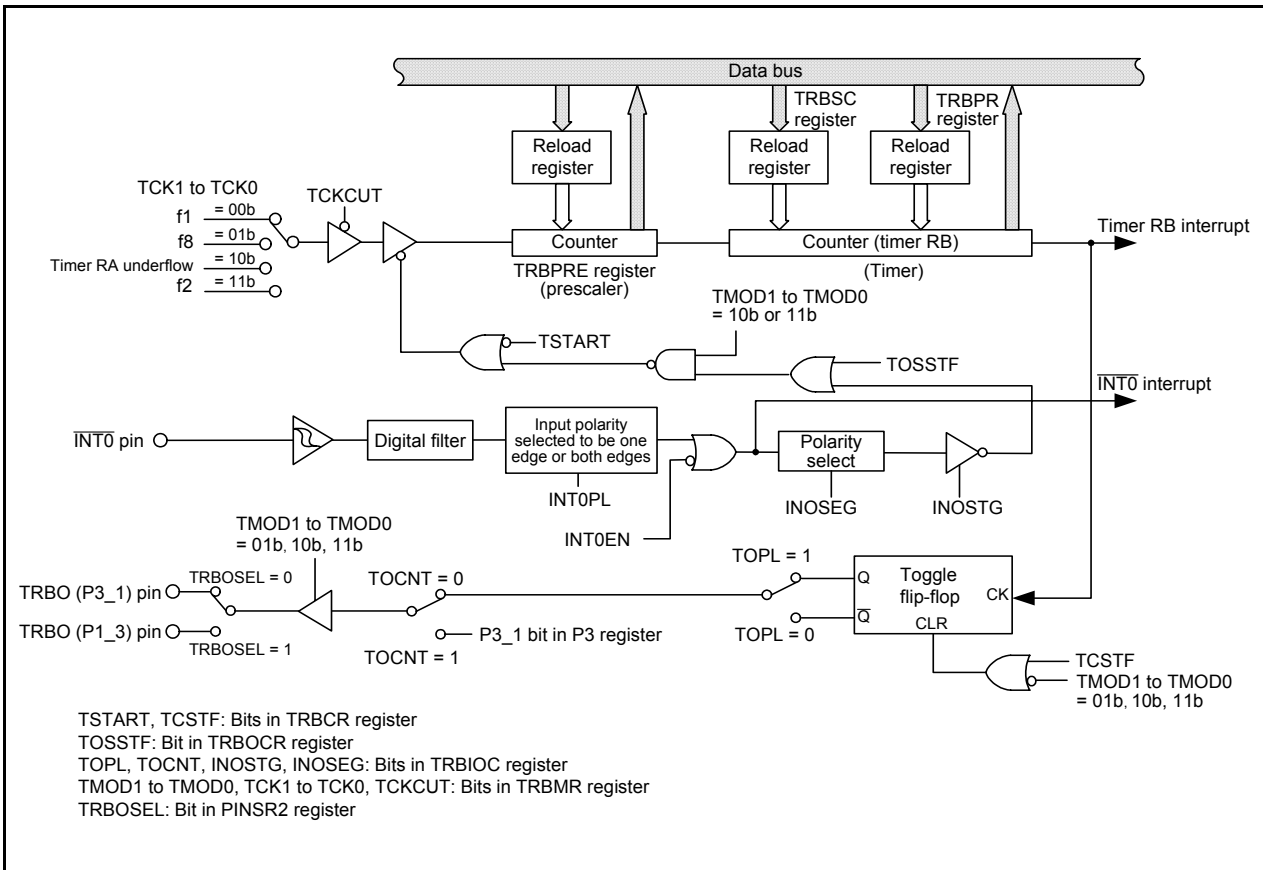


Figure 17.12 Block Diagram of Timer RB

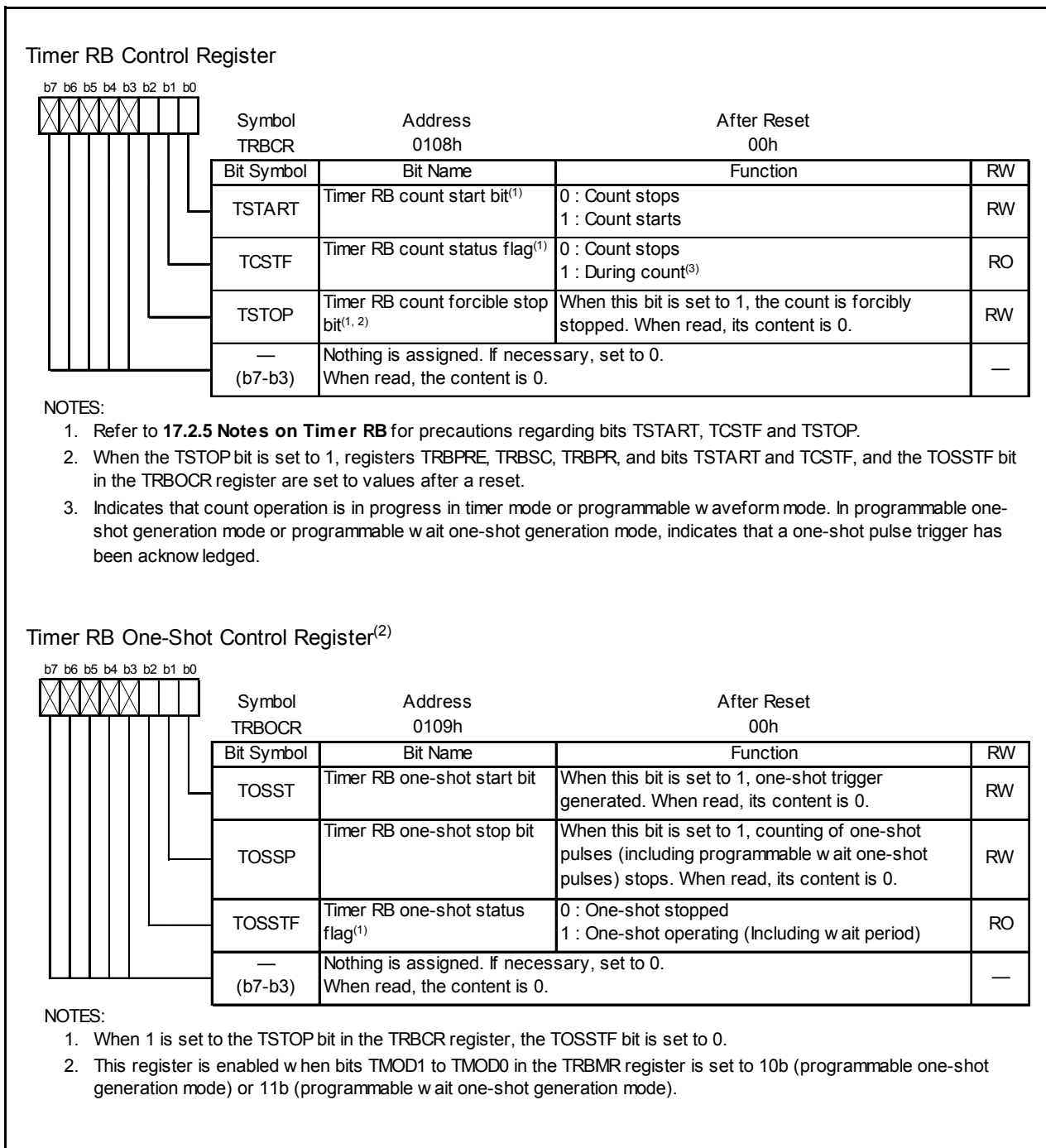


Figure 17.13 Registers TRBCR and TRBOCR

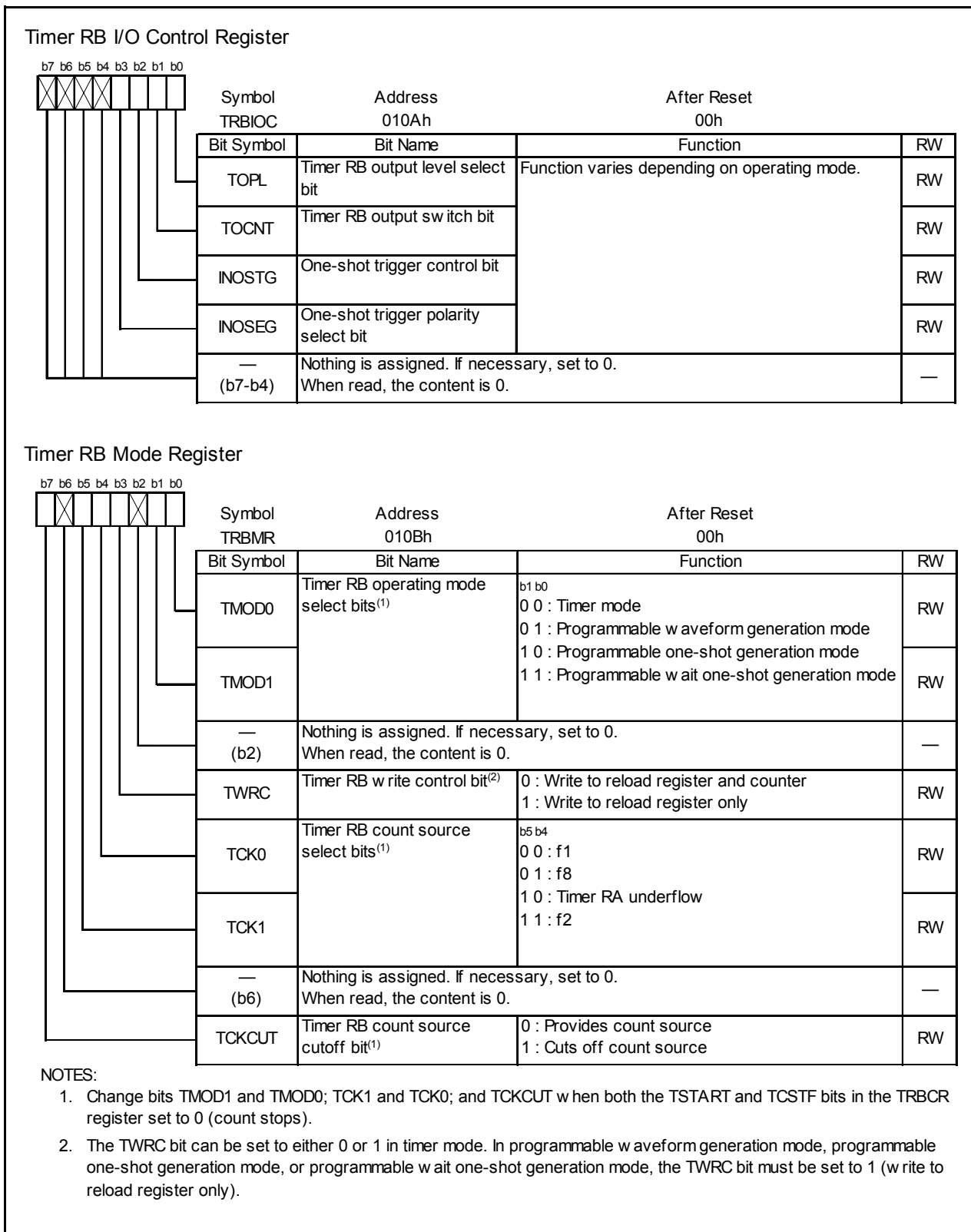


Figure 17.14 Registers TRBIOC and TRBMR

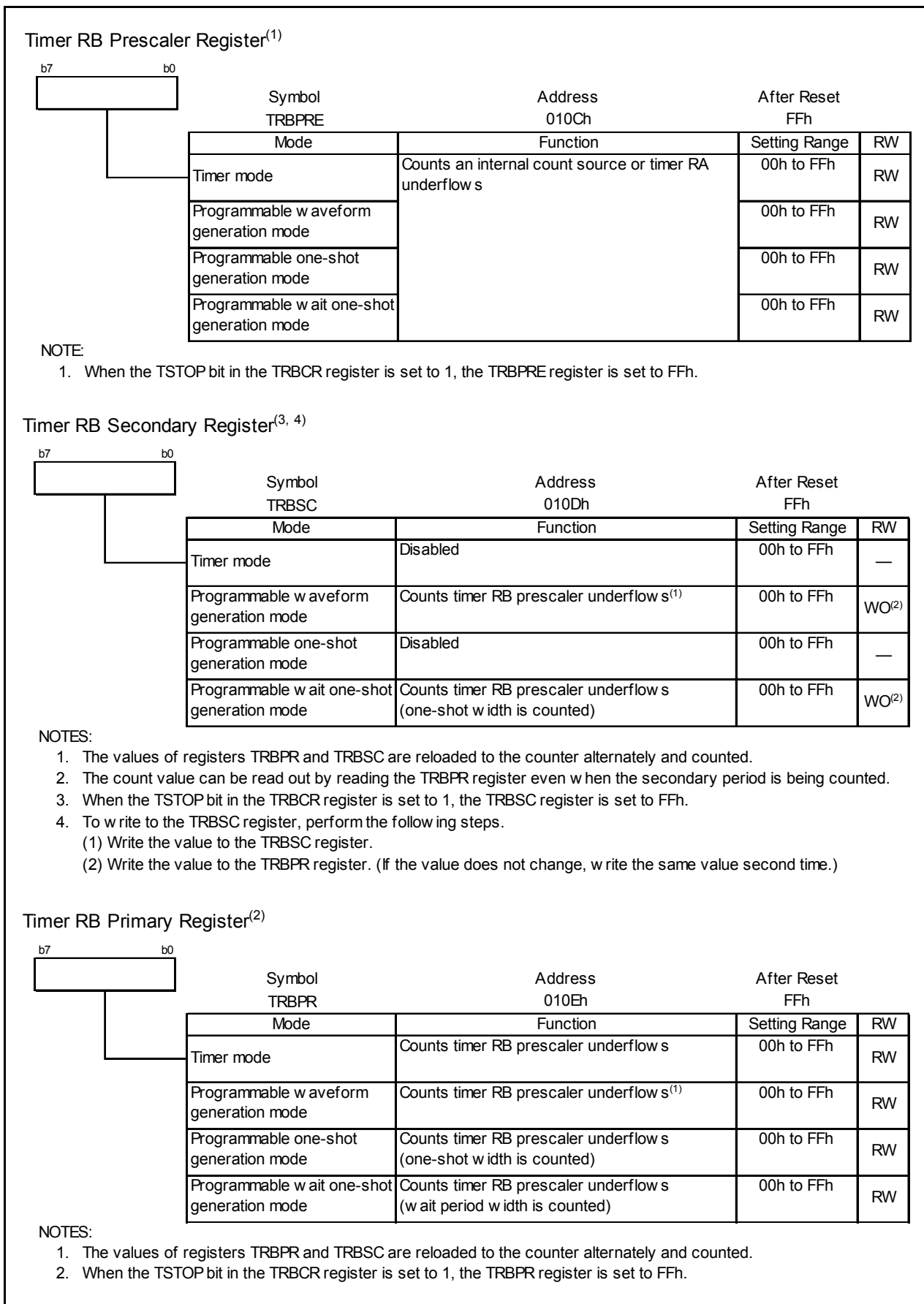


Figure 17.15 Registers TRBPRES, TRBSC, and TRBPR



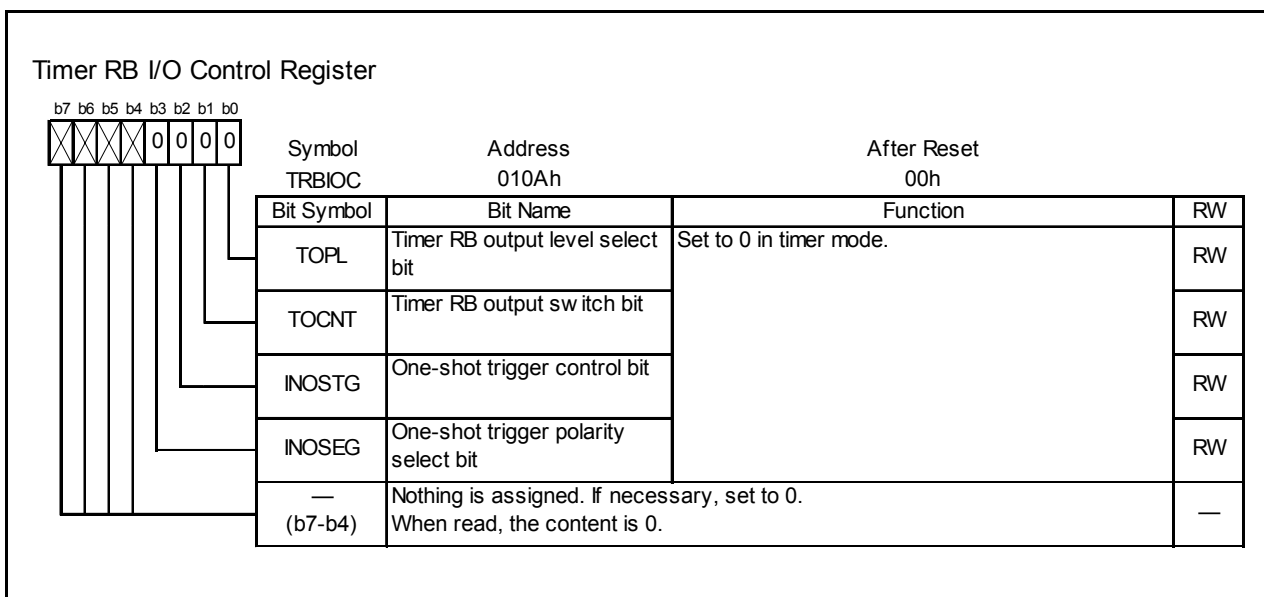
### 17.2.1 Timer Mode

In timer mode, a count source which is internally generated or timer RA underflows are counted (refer to **Table 17.7 Timer Mode Specifications**). Registers TRBOCR and TRBSC are not used in timer mode.

Figure 17.16 shows TRBIOC Register in Timer Mode.

**Table 17.7 Timer Mode Specifications**

Item	Specification
Count sources	f1, f2, f8, timer RA underflow
Count operations	<ul style="list-style-type: none"> <li>Decrement</li> <li>When the timer underflows, it reloads the reload register contents before the count continues (when timer RB underflows, the contents of timer RB primary reload register is reloaded).</li> </ul>
Divide ratio	$1/(n+1)(m+1)$ n: setting value in TRBPRES register, m: setting value in TRBPR register
Count start condition	1 (count starts) is written to the TSTART bit in the TRBCR register.
Count stop conditions	<ul style="list-style-type: none"> <li>0 (count stops) is written to the TSTART bit in the TRBCR register.</li> <li>1 (count forcibly stop) is written to the TSTOP bit in the TRBCR register.</li> </ul>
Interrupt request generation timing	When timer RB underflows [timer RB interrupt].
TRBO pin function	Programmable I/O port
INT0 pin function	Programmable I/O port or $\overline{\text{INT0}}$ interrupt input
Read from timer	The count value can be read out by reading registers TRBPR and TRBPRES.
Write to timer	<ul style="list-style-type: none"> <li>When registers TRBPRES and TRBPR are written while the count is stopped, values are written to both the reload register and counter.</li> <li>When registers TRBPRES and TRBPR are written to while count operation is in progress:                If the TWRC bit in the TRBMR register is set to 0, the value is written to both the reload register and the counter.                If the TWRC bit is set to 1, the value is written to the reload register only.                (Refer to <b>17.2.1.1 Timer Write Control during Count Operation</b>.)</li> </ul>

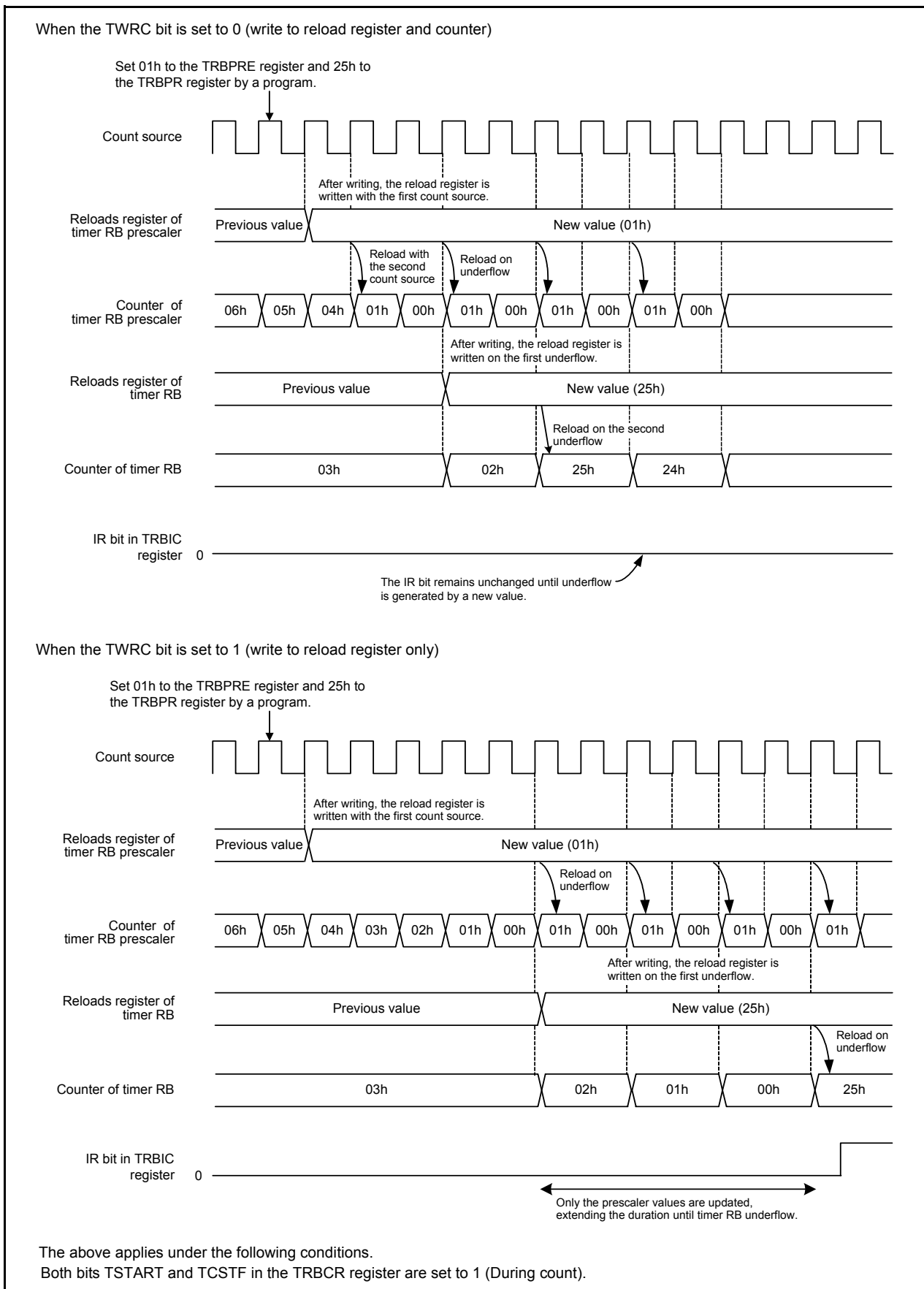


**Figure 17.16 TRBIOC Register in Timer Mode**

### 17.2.1.1 Timer Write Control during Count Operation

Timer RB has a prescaler and a timer (which counts the prescaler underflows). The prescaler and timer each consist of a reload register and a counter. In timer mode, the TWRC bit in the TRBMR register can be used to select whether writing to the prescaler or timer during count operation is performed to both the reload register and counter or only to the reload register.

However, values are transferred from the reload register to the counter of the prescaler in synchronization with the count source. In addition, values are transferred from the reload register to the counter of the timer in synchronization with prescaler underflows. Therefore, even if the TWRC bit is set for writing to both the reload register and counter, the counter value is not updated immediately after the WRITE instruction is executed. In addition, if the TWRC bit is set for writing to the reload register only, the synchronization of the writing will be shifted if the prescaler value changes. Figure 17.17 shows an Operating Example of Timer RB when Counter Value is Rewritten during Count Operation.



**Figure 17.17 Operating Example of Timer RB when Counter Value is Rewritten during Count Operation**

## 17.2.2 Programmable Waveform Generation Mode

In programmable waveform generation mode, the signal output from the TRBO pin is inverted each time the counter underflows, while the values in registers TRBPR and TRBSC are counted alternately (refer to **Table 17.8 Programmable Waveform Generation Mode Specifications**). Counting starts by counting the setting value in the TRBPR register. The TRBOCR register is unused in this mode.

Figure 17.18 shows TRBIOC Register in Programmable Waveform Generation Mode. Figure 17.19 shows an Operating Example of Timer RB in Programmable Waveform Generation Mode.

**Table 17.8 Programmable Waveform Generation Mode Specifications**

Item	Specification
Count sources	f1, f2, f8, timer RA underflow
Count operations	<ul style="list-style-type: none"> <li>Decrement</li> <li>When the timer underflows, it reloads the contents of the primary reload and secondary reload registers alternately before the count continues.</li> </ul>
Width and period of output waveform	Primary period: $(n+1)(m+1)/f_i$ Secondary period: $(n+1)(p+1)/f_i$ Period: $(n+1)\{(m+1)+(p+1)\}/f_i$ $f_i$ : Count source frequency $n$ : Value set in TRBPRES register $m$ : Value set in TRBPR register $p$ : Value set in TRBSC register
Count start condition	1 (count start) is written to the TSTART bit in the TRBCR register.
Count stop conditions	<ul style="list-style-type: none"> <li>0 (count stop) is written to the TSTART bit in the TRBCR register.</li> <li>1 (count forcibly stop) is written to the TSTOP bit in the TRBCR register.</li> </ul>
Interrupt request generation timing	In half a cycle of the count source, after timer RB underflows during the secondary period (at the same time as the TRBO output change) [timer RB interrupt]
TRBO pin function	Programmable output port or pulse output
$\overline{\text{INT0}}$ pin function	Programmable I/O port or $\overline{\text{INT0}}$ interrupt input
Read from timer	The count value can be read out by reading registers TRBPR and TRBPRES. <sup>(1)</sup>
Write to timer	<ul style="list-style-type: none"> <li>When registers TRBPRES, TRBSC, and TRBPR are written while the count is stopped, values are written to both the reload register and counter.</li> <li>When registers TRBPRES, TRBSC, and TRBPR are written to during count operation, values are written to the reload registers only.<sup>(2)</sup></li> </ul>
Select functions	<ul style="list-style-type: none"> <li>Output level select function The TOPL bit in the TRBIOC register selects the output level during primary and secondary periods.</li> <li>TRBO pin output switch function Timer RB pulse output or P3_1 (P1_3) latch output is selected by the TOCNT bit in the TRBIOC register.<sup>(3)</sup></li> <li>TRBO pin select function P3_1 or P1_3 is selected by the TRBOSEL bit in the PINSR2 register.</li> </ul>

NOTES:

- Even when counting the secondary period, the TRBPR register may be read.
- The set values are reflected in the waveform output beginning with the following primary period after writing to the TRBPR register.
- The value written to the TOCNT bit is enabled by the following.
  - When counting starts.
  - When a timer RB interrupt request is generated.  
The contents after the TOCNT bit is changed are reflected from the output of the following primary period.

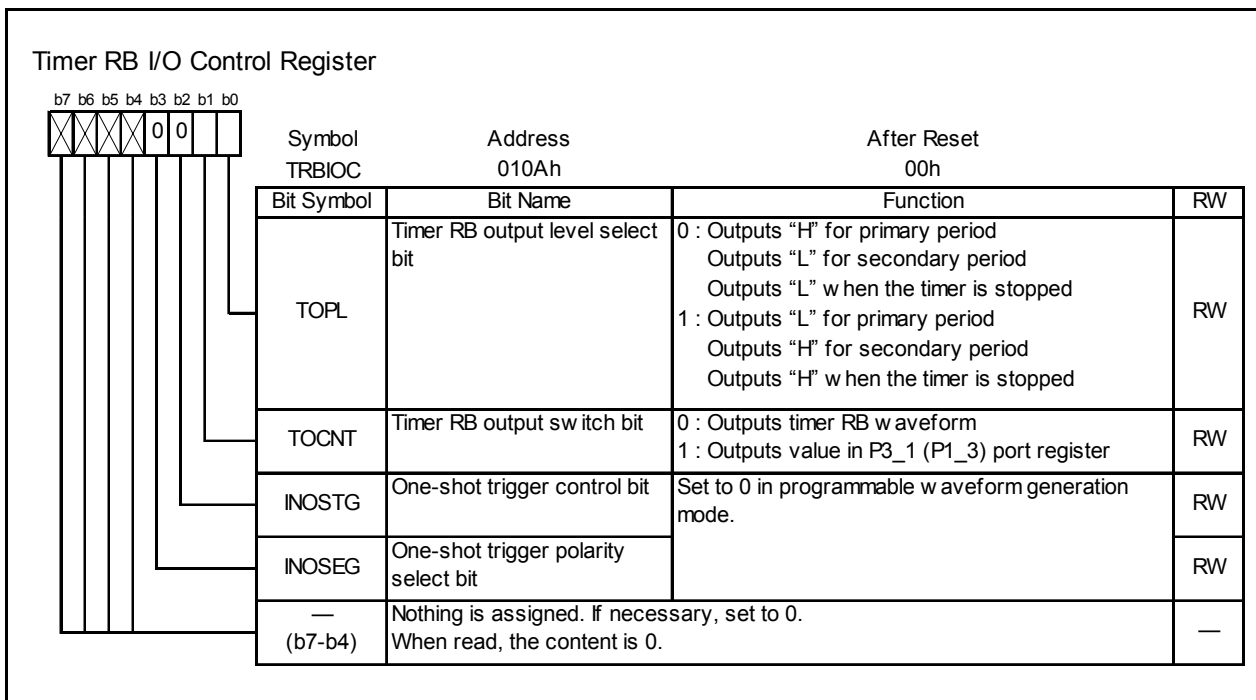


Figure 17.18 TRBIOC Register in Programmable Waveform Generation Mode

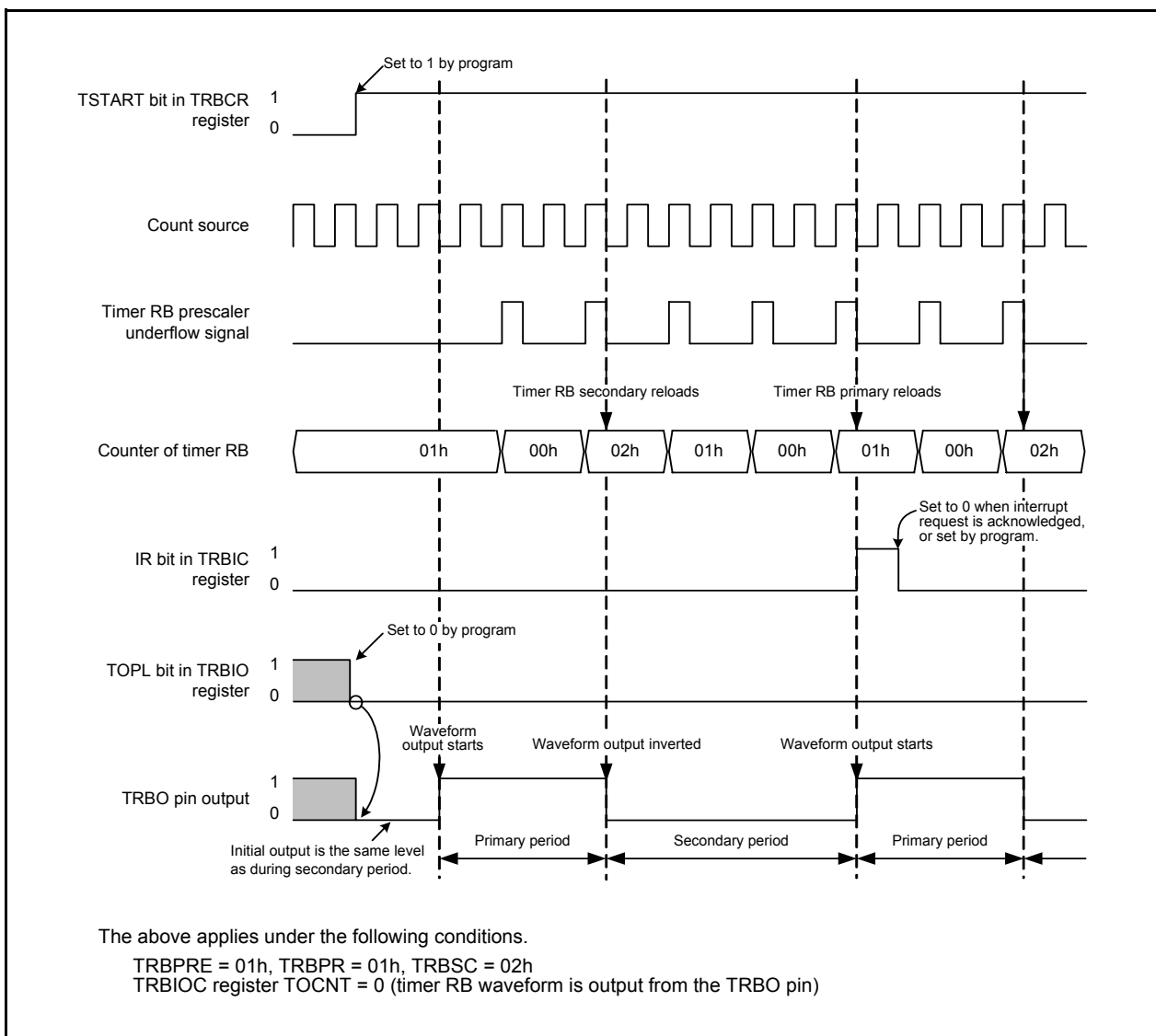


Figure 17.19 Operating Example of Timer RB in Programmable Waveform Generation Mode

### 17.2.3 Programmable One-shot Generation Mode

In programmable one-shot generation mode, a one-shot pulse is output from the TRBO pin by a program or an external trigger input (input to the  $\overline{\text{INT0}}$  pin) (refer to **Table 17.9 Programmable One-Shot Generation Mode Specifications**). When a trigger is generated, the timer starts operating from the point only once for a given period equal to the set value in the TRBPR register. The TRBSC register is not used in this mode.

Figure 17.20 shows TRBIOC Register in Programmable One-Shot Generation Mode. Figure 17.21 shows an Operating Example of Programmable One-Shot Generation Mode.

**Table 17.9 Programmable One-Shot Generation Mode Specifications**

Item	Specification
Count sources	f1, f2, f8, timer RA underflow
Count operations	<ul style="list-style-type: none"> <li>Decrement the setting value in the TRBPR register</li> <li>When the timer underflows, it reloads the contents of the reload register before the count completes and the TOSSTF bit is set to 0 (one-shot stops).</li> <li>When the count stops, the timer reloads the contents of the reload register before it stops.</li> </ul>
One-shot pulse output time	$(n+1)(m+1)/f_i$ $f_i$ : Count source frequency, $n$ : Setting value in TRBPRES register, $m$ : Setting value in TRBPR register <sup>(2)</sup>
Count start conditions	<ul style="list-style-type: none"> <li>The TSTART bit in the TRBCR register is set to 1 (count starts) and the next trigger is generated</li> <li>Set the TOSST bit in the TRBOCR register to 1 (one-shot starts)</li> <li>Input trigger to the <math>\overline{\text{INT0}}</math> pin</li> </ul>
Count stop conditions	<ul style="list-style-type: none"> <li>When reloading completes after timer RB underflows during primary period</li> <li>When the TOSSP bit in the TRBOCR register is set to 1 (one-shot stops)</li> <li>When the TSTART bit in the TRBCR register is set to 0 (stops counting)</li> <li>When the TSTOP bit in the TRBCR register is set to 1 (forcibly stops counting)</li> </ul>
Interrupt request generation timing	In half a cycle of the count source, after the timer underflows (at the same time as the TRBO output ends) [timer RB interrupt]
TRBO pin function	Pulse output
$\overline{\text{INT0}}$ pin functions	<ul style="list-style-type: none"> <li>When the INOSTG bit in the TRBIOC register is set to 0 (<math>\overline{\text{INT0}}</math> one-shot trigger disabled): programmable I/O port or <math>\overline{\text{INT0}}</math> interrupt input</li> <li>When the INOSTG bit in the TRBIOC register is set to 1 (<math>\overline{\text{INT0}}</math> one-shot trigger enabled): external trigger (<math>\overline{\text{INT0}}</math> interrupt input)</li> </ul>
Read from timer	The count value can be read out by reading registers TRBPR and TRBPRES.
Write to timer	<ul style="list-style-type: none"> <li>When registers TRBPRES and TRBPR are written while the count is stopped, values are written to both the reload register and counter.</li> <li>When registers TRBPRES and TRBPR are written during the count, values are written to the reload register only (the data is transferred to the counter at the following reload).<sup>(1)</sup></li> </ul>
Select functions	<ul style="list-style-type: none"> <li>Output level select function The TOPL bit in the TRBIOC register selects the output level of the one-shot pulse waveform.</li> <li>One-shot trigger select function Refer to <b>17.2.3.1 One-Shot Trigger Selection</b>.</li> <li>TRBO pin select function P3_1 or P1_3 is selected by the TRBOSEL bit in the PINSR2 register.</li> </ul>

**NOTES:**

- The set value is reflected at the following one-shot pulse after writing to the TRBPR register.
- Do not set both the TRBPRES and TRBPR registers to 00h.

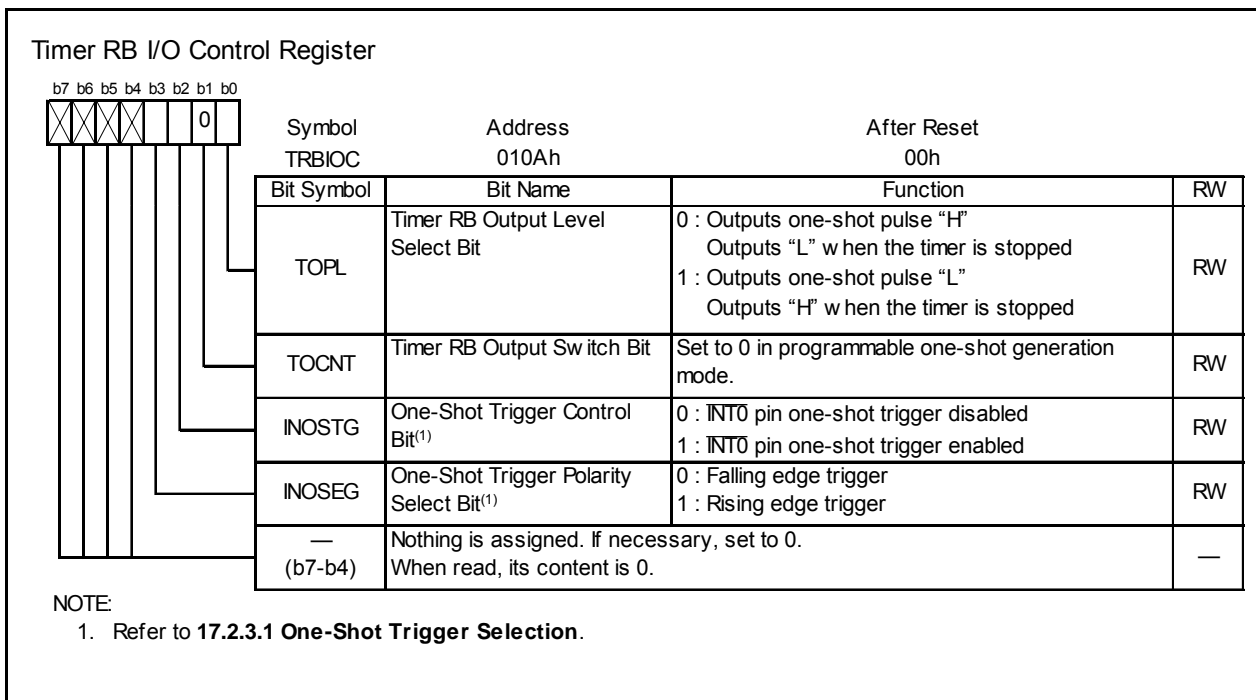


Figure 17.20 TRBIOC Register in Programmable One-Shot Generation Mode



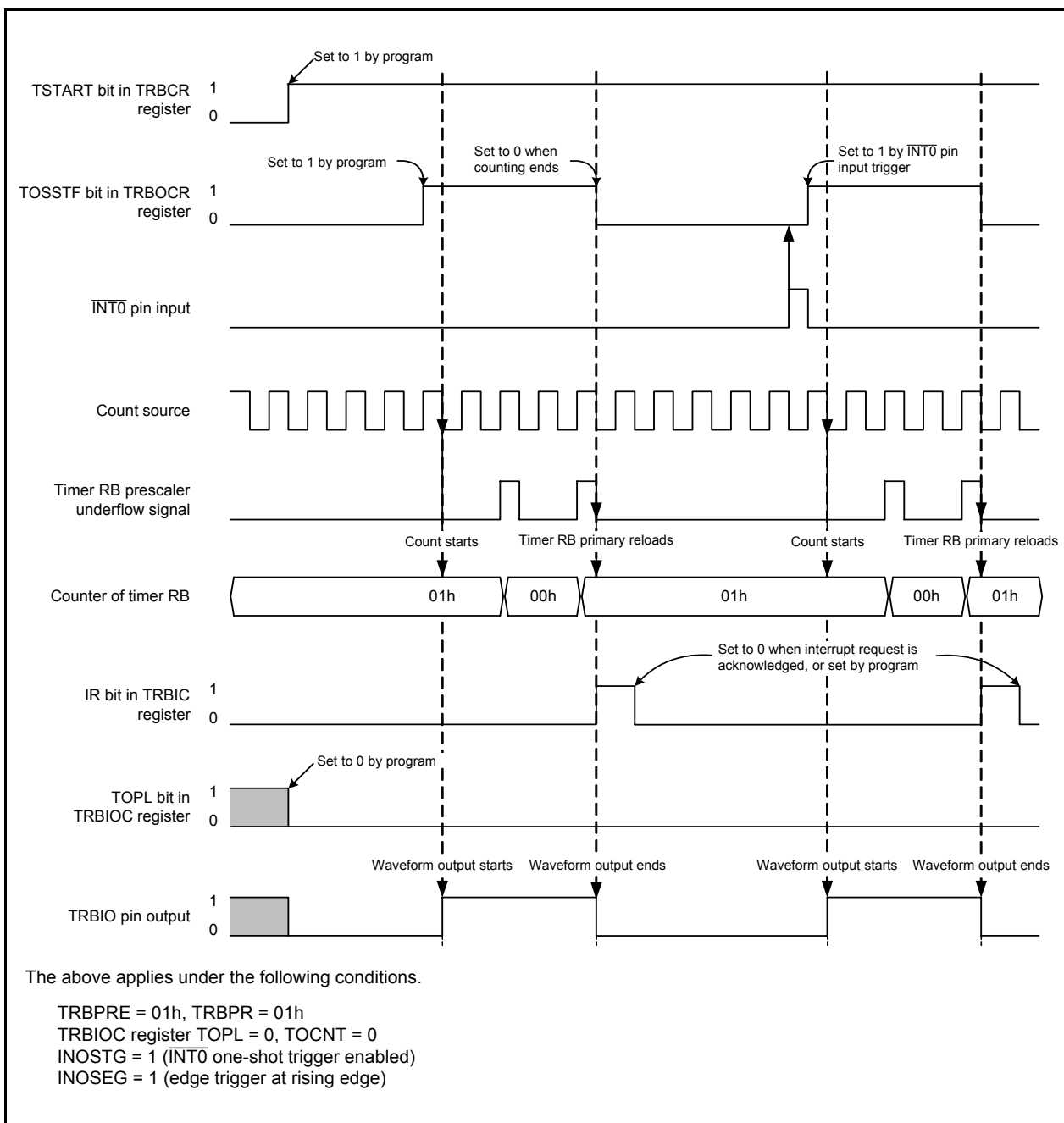


Figure 17.21 Operating Example of Programmable One-Shot Generation Mode

### 17.2.3.1 One-Shot Trigger Selection

In programmable one-shot generation mode and programmable wait one-shot generation mode, operation starts when a one-shot trigger is generated while the TCSTF bit in the TRBCR register is set to 1 (count starts).

A one-shot trigger can be generated by either of the following causes:

- 1 is written to the TOSST bit in the TRBOCR register by a program.
- Trigger input from the  $\overline{\text{INT0}}$  pin.

When a one-shot trigger occurs, the TOSSTF bit in the TRBOCR register is set to 1 (one-shot operation in progress) after one or two cycles of the count source have elapsed. Then, in programmable one-shot generation mode, count operation begins and one-shot waveform output starts. (In programmable wait one-shot generation mode, count operation starts for the wait period.) If a one-shot trigger occurs while the TOSSTF bit is set to 1, no retriggering occurs.

To use trigger input from the  $\overline{\text{INT0}}$  pin, input the trigger after making the following settings:

- Set the PD4\_5 bit in the PD4 register to 0 (input port).
- Select the  $\overline{\text{INT0}}$  digital filter with bits INT0F1 and INT0F0 in the INTF register.
- Select both edges or one edge with the INT0PL bit in INTEN register. If one edge is selected, further select falling or rising edge with the INOSEG bit in TRBIOC register.
- Set the INT0EN bit in the INTEN register to 0 (enabled).
- After completing the above, set the INOSTG bit in the TRBIOC register to 1 ( $\overline{\text{INT}}$  pin one-shot trigger enabled).

Note the following points with regard to generating interrupt requests by trigger input from the  $\overline{\text{INT0}}$  pin.

- Processing to handle the interrupts is required. Refer to **13. Interrupts**, for details.
- If one edge is selected, use the POL bit in the INT0IC register to select falling or rising edge. (The INOSEG bit in the TRBIOC register does not affect  $\overline{\text{INT0}}$  interrupts).
- If a one-shot trigger occurs while the TOSSTF bit is set to 1, timer RB operation is not affected, but the value of the IR bit in the INT0IC register changes.

### 17.2.4 Programmable Wait One-Shot Generation Mode

In programmable wait one-shot generation mode, a one-shot pulse is output from the TRBO pin by a program or an external trigger input (input to the  $\overline{\text{INT0}}$  pin) (refer to **Table 17.10 Programmable Wait One-Shot Generation Mode Specifications**). When a trigger is generated from that point, the timer outputs a pulse only once for a given length of time equal to the setting value in the TRBSC register after waiting for a given length of time equal to the setting value in the TRBPR register.

Figure 17.22 shows TRBIOC Register in Programmable Wait One-Shot Generation Mode. Figure 17.23 shows an Operating Example of Programmable Wait One-Shot Generation Mode.

**Table 17.10 Programmable Wait One-Shot Generation Mode Specifications**

Item	Specification
Count sources	f1, f2, f8, timer RA underflow
Count operations	<ul style="list-style-type: none"> <li>Decrement the timer RB primary setting value.</li> <li>When a count of the timer RB primary underflows, the timer reloads the contents of timer RB secondary before the count continues.</li> <li>When a count of the timer RB secondary underflows, the timer reloads the contents of timer RB primary before the count completes and the TOSSTF bit is set to 0 (one-shot stops).</li> <li>When the count stops, the timer reloads the contents of the reload register before it stops.</li> </ul>
Wait time	$(n+1)(m+1)/f_i$ $f_i$ : Count source frequency $n$ : Value set in the TRBPRES register, $m$ : Value set in the TRBPR register <sup>(2)</sup>
One-shot pulse output time	$(n+1)(p+1)/f_i$ $f_i$ : Count source frequency $n$ : Value set in the TRBPRES register, $p$ : Value set in the TRBSC register
Count start conditions	<ul style="list-style-type: none"> <li>The TSTART bit in the TRBCR register is set to 1 (count starts) and the next trigger is generated.</li> <li>Set the TOSST bit in the TRBOCR register to 1 (one-shot starts).</li> <li>Input trigger to the <math>\overline{\text{INT0}}</math> pin</li> </ul>
Count stop conditions	<ul style="list-style-type: none"> <li>When reloading completes after timer RB underflows during secondary period.</li> <li>When the TOSSP bit in the TRBOCR register is set to 1 (one-shot stops).</li> <li>When the TSTART bit in the TRBCR register is set to 0 (starts counting).</li> <li>When the TSTOP bit in the TRBCR register is set to 1 (forcibly stops counting).</li> </ul>
Interrupt request generation timing	In half a cycle of the count source after timer RB underflows during secondary period (complete at the same time as waveform output from the TRBO pin) [timer RB interrupt].
TRBO pin function	Pulse output
$\overline{\text{INT0}}$ pin functions	<ul style="list-style-type: none"> <li>When the INOSTG bit in the TRBIOC register is set to 0 (<math>\overline{\text{INT0}}</math> one-shot trigger disabled): programmable I/O port or <math>\overline{\text{INT0}}</math> interrupt input</li> <li>When the INOSTG bit in the TRBIOC register is set to 1 (<math>\overline{\text{INT0}}</math> one-shot trigger enabled): external trigger (<math>\overline{\text{INT0}}</math> interrupt input)</li> </ul>
Read from timer	The count value can be read out by reading registers TRBPR and TRBPRES.
Write to timer	<ul style="list-style-type: none"> <li>When registers TRBPRES, TRBSC, and TRBPR are written while the count stops, values are written to both the reload register and counter.</li> <li>When registers TRBPRES, TRBSC, and TRBPR are written to during count operation, values are written to the reload registers only.<sup>(1)</sup></li> </ul>
Select functions	<ul style="list-style-type: none"> <li>Output level select function The TOPL bit in the TRBIOC register selects the output level of the one-shot pulse waveform.</li> <li>One-shot trigger select function Refer to <b>17.2.3.1 One-Shot Trigger Selection</b>.</li> <li>TRBO pin select function P3_1 or P1_3 is selected by the TRBOSEL bit in the PINSR2 register.</li> </ul>

NOTES:

- The set value is reflected at the following one-shot pulse after writing to registers TRBSC and TRBPR.
- Do not set both the TRBPRES and TRBPR registers to 00h.

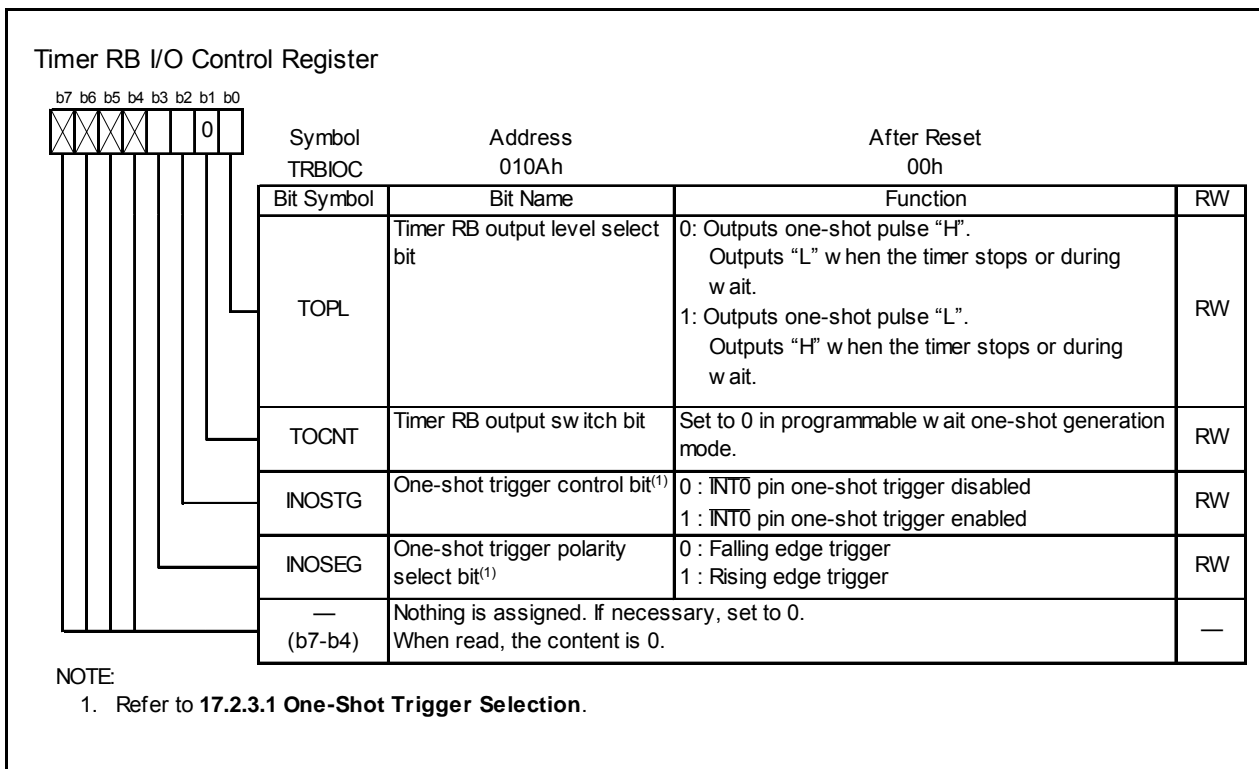


Figure 17.22 TRBIOC Register in Programmable Wait One-Shot Generation Mode

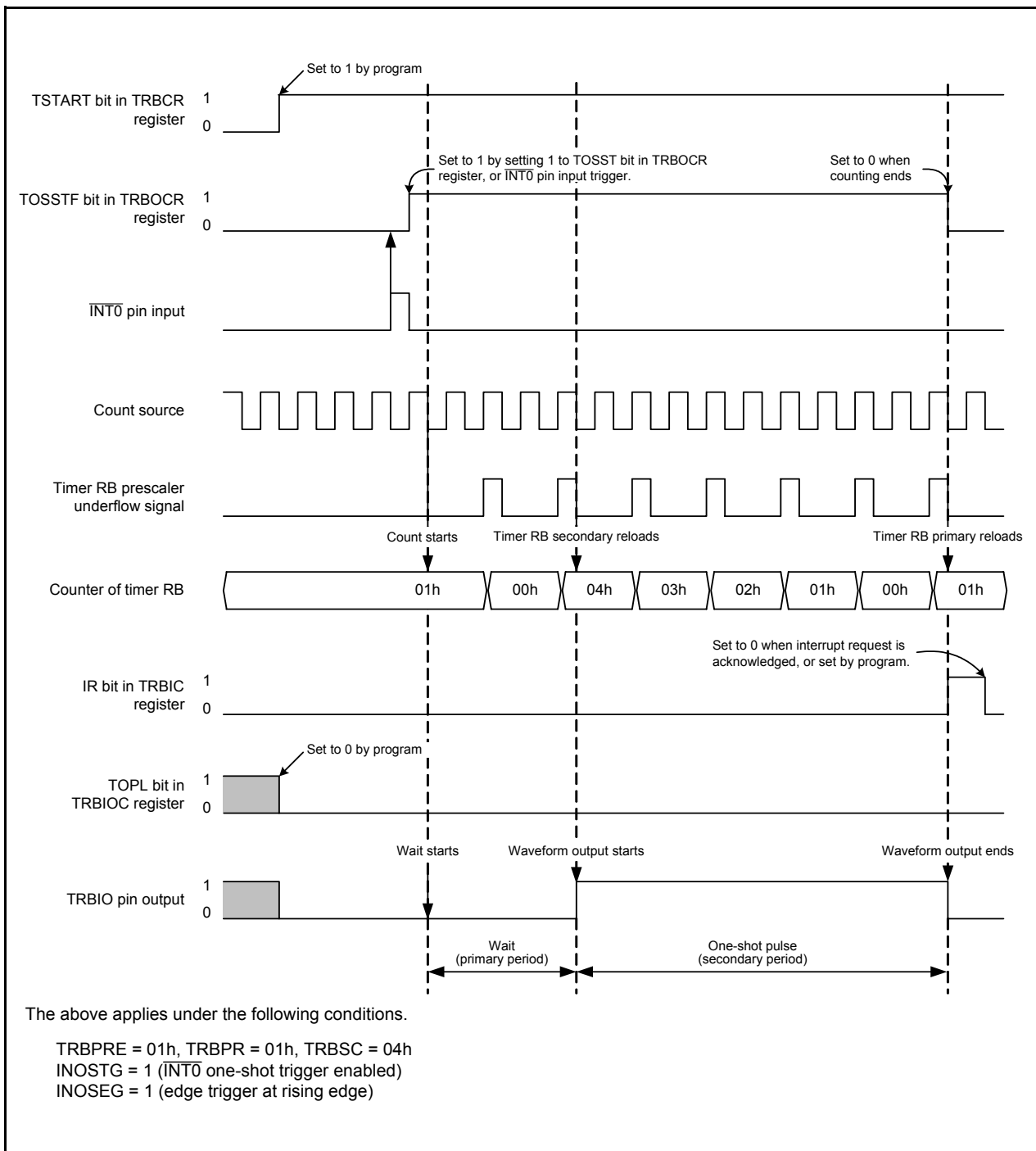


Figure 17.23 Operating Example of Programmable Wait One-Shot Generation Mode

## 17.2.5 Notes on Timer RB

- Timer RB stops counting after a reset. Set the values in the timer RB and timer RB prescalers before the count starts.
- Even if the prescaler and timer RB is read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- In programmable one-shot generation mode and programmable wait one-shot generation mode, when setting the TSTART bit in the TRBCR register to 0, 0 (stops counting) or setting the TOSSP bit in the TRBOCR register to 1 (stops one-shot), the timer reloads the value of reload register and stops. Therefore, in programmable one-shot generation mode and programmable wait one-shot generation mode, read the timer count value before the timer stops.
- The TCSTF bit remains 0 (count stops) for 1 to 2 cycles of the count source after setting the TSTART bit to 1 (count starts) while the count is stopped.

During this time, do not access registers associated with timer RB<sup>(1)</sup> other than the TCSTF bit. Timer RB starts counting at the first valid edge of the count source after the TCSTF bit is set to 1 (during count).

The TCSTF bit remains 1 for 1 to 2 cycles of the count source after setting the TSTART bit to 0 (count stops) while the count is in progress. Timer RB counting is stopped when the TCSTF bit is set to 0.

During this time, do not access registers associated with timer RB<sup>(1)</sup> other than the TCSTF bit.

### NOTE:

1. Registers associated with timer RB: TRBCR, TRBOCR, TRBIOC, TRBMR, TRBPRES, TRBSC, and TRBPR.

- If the TSTOP bit in the TRBCR register is set to 1 during timer operation, timer RB stops immediately.
- If 1 is written to the TOSST or TOSSP bit in the TRBOCR register, the value of the TOSSTF bit changes after one or two cycles of the count source have elapsed. If the TOSSP bit is written to 1 during the period between when the TOSST bit is written to 1 and when the TOSSTF bit is set to 1, the TOSSTF bit may be set to either 0 or 1 depending on the content state. Likewise, if the TOSST bit is written to 1 during the period between when the TOSSP bit is written to 1 and when the TOSSTF bit is set to 0, the TOSSTF bit may be set to either 0 or 1.

### 17.2.5.1 Timer mode

The following workaround should be performed in timer mode.

To write to registers TRBPRES and TRBPR during count operation (TCSTF bit is set to 1), note the following points:

- When the TRBPRES register is written continuously, allow three or more cycles of the count source for each write interval.
- When the TRBPR register is written continuously, allow three or more cycles of the prescaler underflow for each write interval.

### 17.2.5.2 Programmable waveform generation mode

The following three workarounds should be performed in programmable waveform generation mode.

- (1) To write to registers TRBPRE and TRBPR during count operation (TCSTF bit is set to 1), note the following points:
  - When the TRBPRE register is written continuously, allow three or more cycles of the count source for each write interval.
  - When the TRBPR register is written continuously, allow three or more cycles of the prescaler underflow for each write interval.
- (2) To change registers TRBPRE and TRBPR during count operation (TCSTF bit is set to 1), synchronize the TRBO output cycle using a timer RB interrupt, etc. This operation should be performed only once in the same output cycle. Also, make sure that writing to the TRBPR register does not occur during period A shown in Figures 17.24 and 17.25.

The following shows the detailed workaround examples.

- Workaround example (a):  
As shown in Figure 17.24, write to registers TRBSC and TRBPR in the timer RB interrupt routine. These write operations must be completed by the beginning of period A.

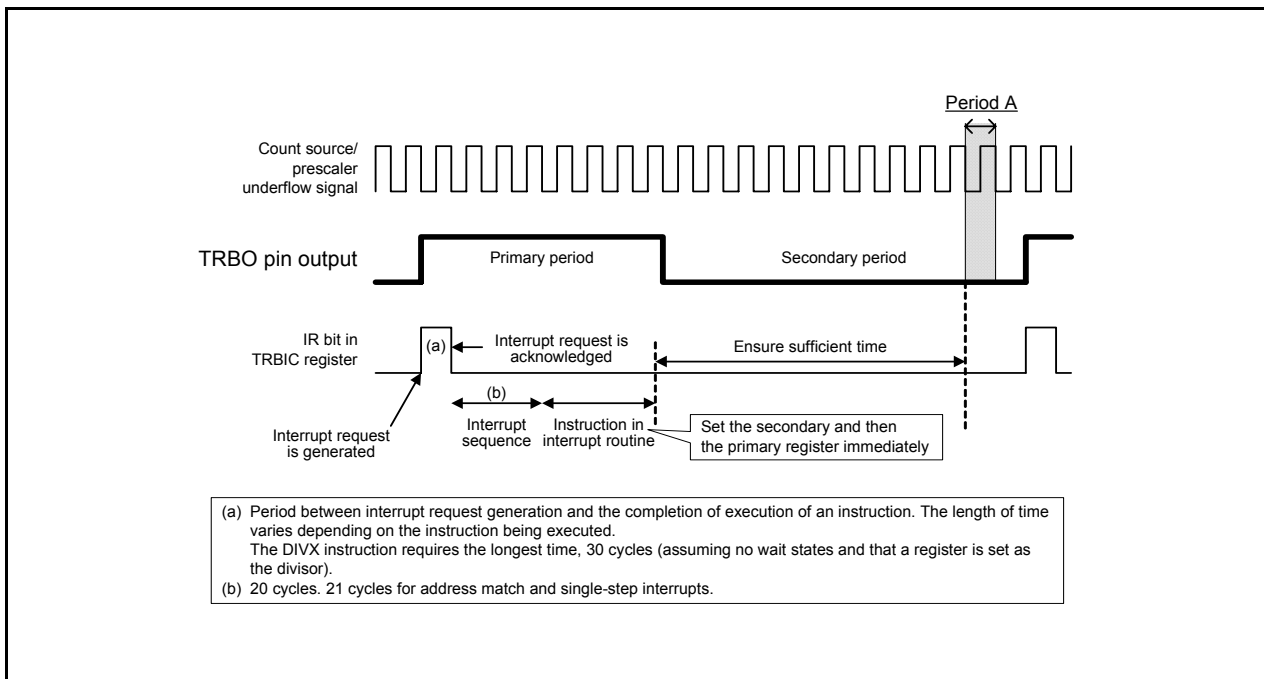
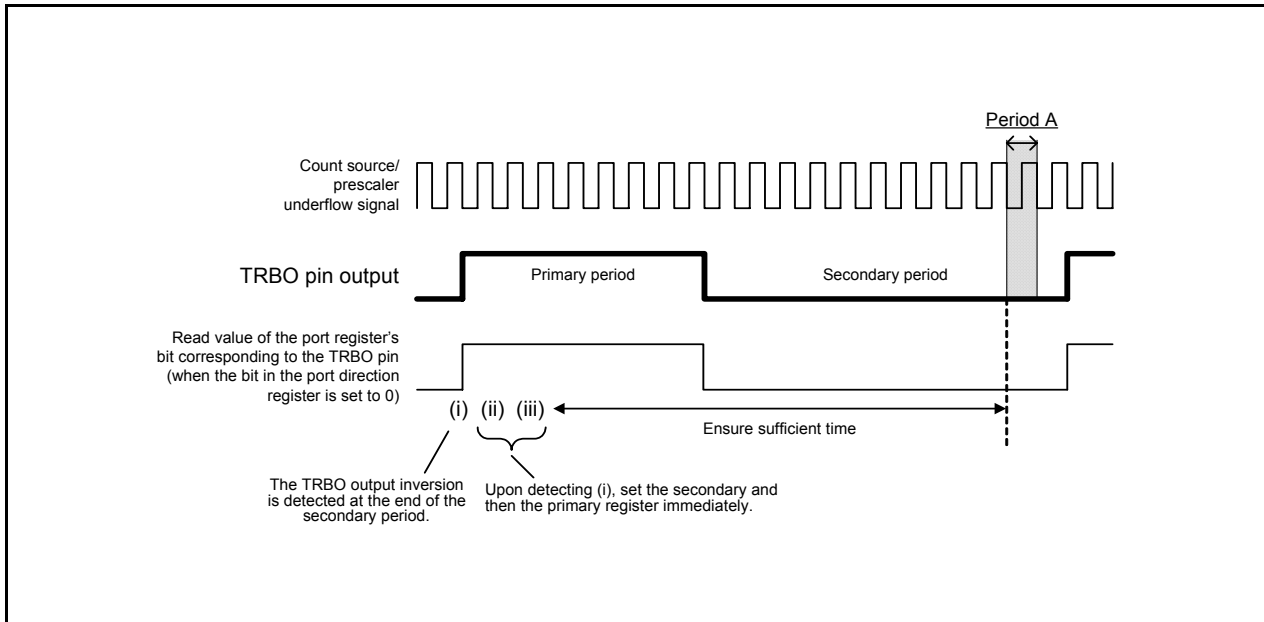


Figure 17.24 Workaround Example (a) When Timer RB interrupt is Used

- Workaround example (b):

As shown in Figure 17.25 detect the start of the primary period by the TRBO pin output level and write to registers TRBSC and TRBPR. These write operations must be completed by the beginning of period A.

If the port register's bit value is read after the port direction register's bit corresponding to the TRBO pin is set to 0 (input mode), the read value indicates the TRBO pin output value.



**Figure 17.25 Workaround Example (b) When TRBO Pin Output Value is Read**

- (3) To stop the timer counting in the primary period, use the TSTOP bit in the TRBCR register. In this case, registers TRBPRES and TRBPR are initialized and their values are set to the values after reset.

### 17.2.5.3 Programmable one-shot generation mode

The following two workarounds should be performed in programmable one-shot generation mode.

- (1) To write to registers TRBPRES and TRBPR during count operation (TCSTF bit is set to 1), note the following points:
  - When the TRBPRES register is written continuously during count operation (TCSTF bit is set to 1), allow three or more cycles of the count source for each write interval.
  - When the TRBPR register is written continuously during count operation (TCSTF bit is set to 1), allow three or more cycles of the prescaler underflow for each write interval.
- (2) Do not set both the TRBPRES and TRBPR registers to 00h.



#### 17.2.5.4 Programmable wait one-shot generation mode

The following three workarounds should be performed in programmable wait one-shot generation mode.

- (1) To write to registers TRBPRE and TRBPR during count operation (TCSTF bit is set to 1), note the following points:
  - When the TRBPRE register is written continuously, allow three or more cycles of the count source for each write interval.
  - When the TRBPR register is written continuously, allow three or more cycles of the prescaler underflow for each write interval.
- (2) Do not set both the TRBPRE and TRBPR registers to 00h.
- (3) Set registers TRBSC and TRBPR using the following procedure.
  - (a) To use “ $\overline{\text{INT0}}$  pin one-shot trigger enabled” as the count start condition  
Set the TRBSC register and then the TRBPR register. At this time, after writing to the TRBPR register, allow an interval of 0.5 or more cycles of the count source before trigger input from the  $\overline{\text{INT0}}$  pin.
  - (b) To use “writing 1 to TOSST bit” as the start condition  
Set the TRBSC register, the TRBPR register, and then TOSST bit. At this time, after writing to the TRBPR register, allow an interval of 0.5 or more cycles of the count source before writing to the TOSST bit.

### 17.3 Timer RE

Timer RE has the 4-bit counter and 8-bit counter. Timer RE has the following 2 modes:

- Real-time clock mode      Generate 1-second signal from fC4 and count seconds, minutes, hours, and days of the week.
- Output compare mode      Count a count source and detect compare matches.

The count source for timer RE is the operating clock that regulates the timing of timer operations.

### 17.3.1 Real-Time Clock Mode

In real-time clock mode, a 1-second signal is generated from fC4 using a divide-by-2 frequency divider, 4-bit counter, and 8-bit counter and used to count seconds, minutes, hours, and days of the week. Figure 17.26 shows a Block Diagram of Real-Time Clock Mode and Table 17.11 lists the Real-Time Clock Mode Specifications. Figures 17.27 to 17.31 and 17.33 to 17.35 show the Registers Associated with Real-Time Clock Mode. Table 17.12 lists the Interrupt Sources, Figure 17.32 shows the Definition of Time Representation and Figure 17.36 shows the Operating Example in Real-Time Clock Mode.

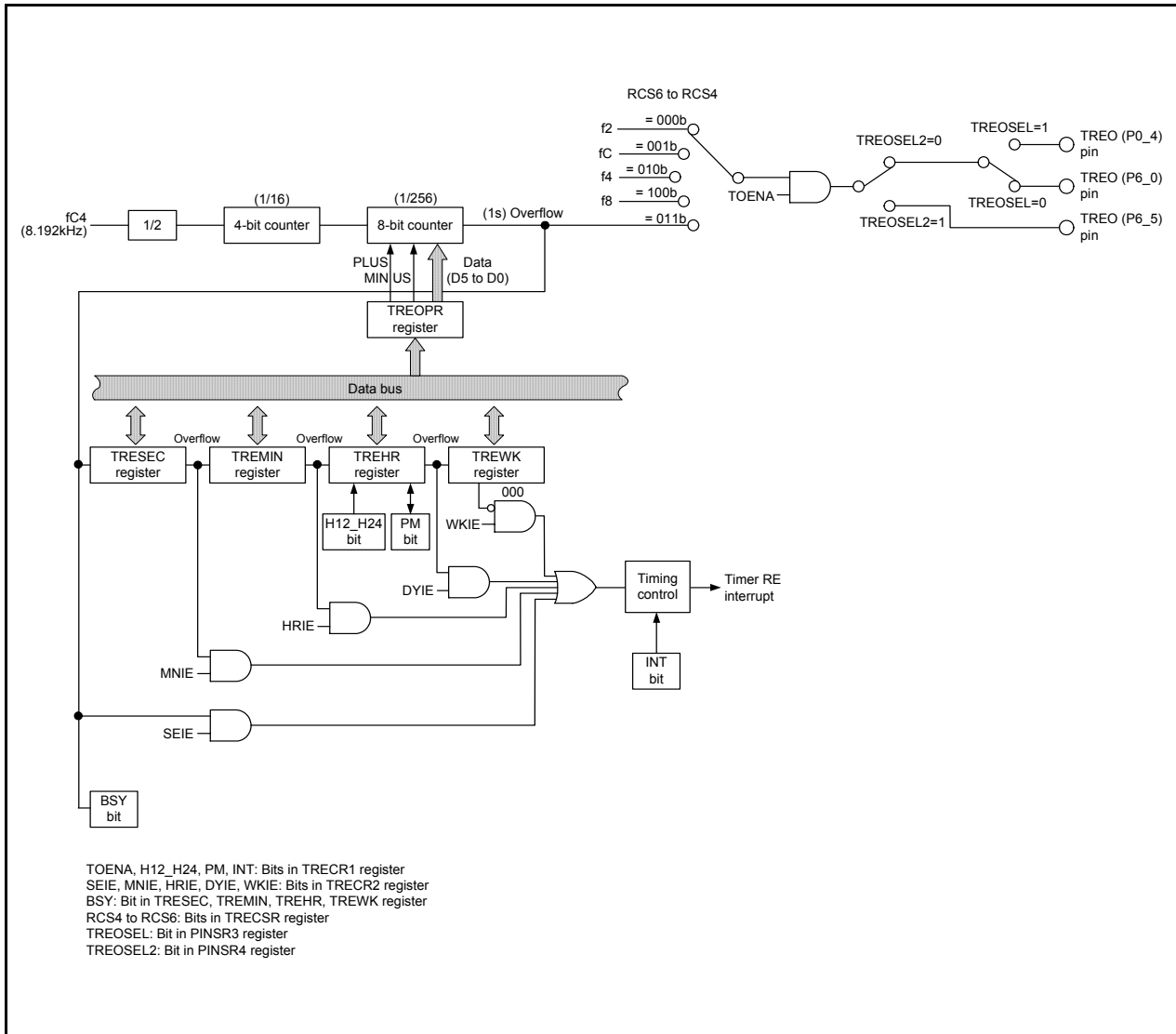


Figure 17.26 Block Diagram of Real-Time Clock Mode

**Table 17.11 Real-Time Clock Mode Specifications**

Item	Specification
Count source	fC4
Count operation	Increment
Count start condition	1 (count starts) is written to TSTART bit in TRECR1 register
Count stop condition	0 (count stops) is written to TSTART bit in TRECR1 register
Interrupt request generation timing	Select any one of the following: <ul style="list-style-type: none"> <li>• Update second data</li> <li>• Update minute data</li> <li>• Update hour data</li> <li>• Update day of week data</li> <li>• When day of week data is set to 000b (Sunday)</li> </ul>
TREO pin function	Programmable I/O ports or output of f2, fC, f4, f8 or, 1Hz
Read from timer	When reading TRESEC, TREMIN, TREHR, or TREWK register, the count value can be read. The values read from registers TRESEC, TREMIN, and TREHR are represented by the BCD code.
Write to timer	When bits TSTART and TCSTF in the TRECR1 register are set to 0 (timer stops), the value can be written to registers TRESEC, TREMIN, TREHR, and TREWK. The values written to registers TRESEC, TREMIN, and TREHR are represented by the BCD codes.
Selectable functions	<ul style="list-style-type: none"> <li>• 12-hour mode/24-hour mode switch function</li> <li>• Counter precision adjustment function</li> <li>• TREO pin select function</li> </ul> P0_4, P6_0, or P6_5 is selected by the TREOSEL bit in the PINSR3 register and the TREOSEL2 bit in the PINSR4 register.

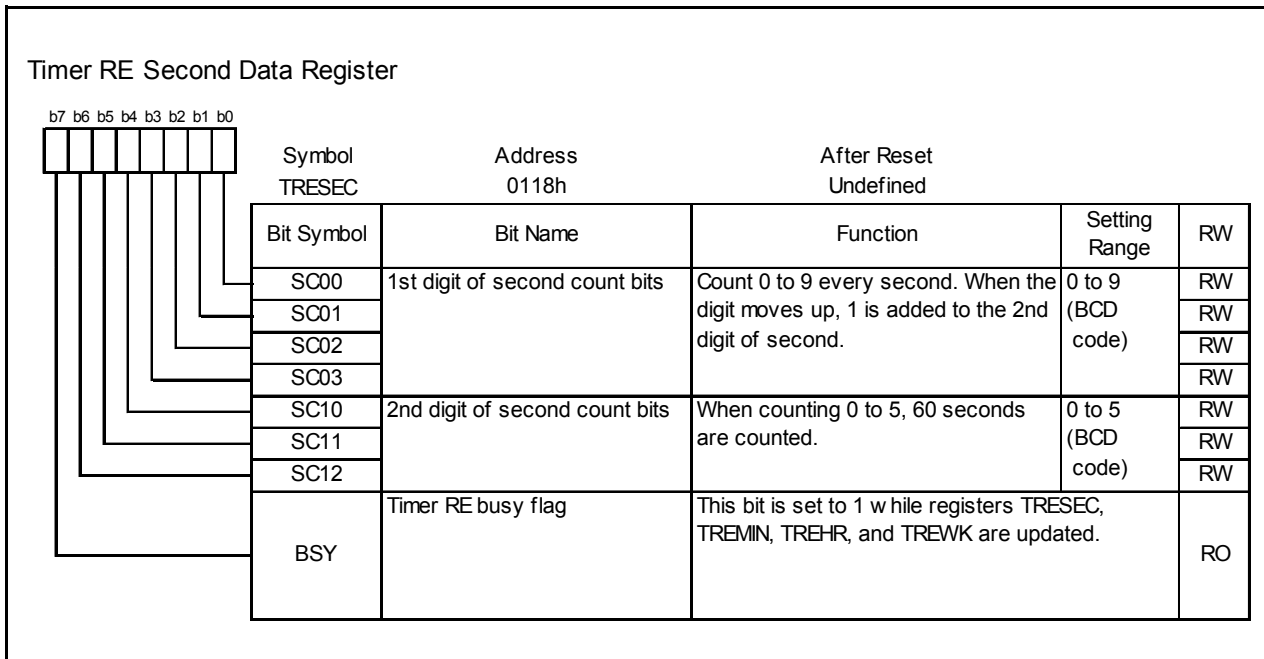


Figure 17.27 TRESEC Register in Real-Time Clock Mode

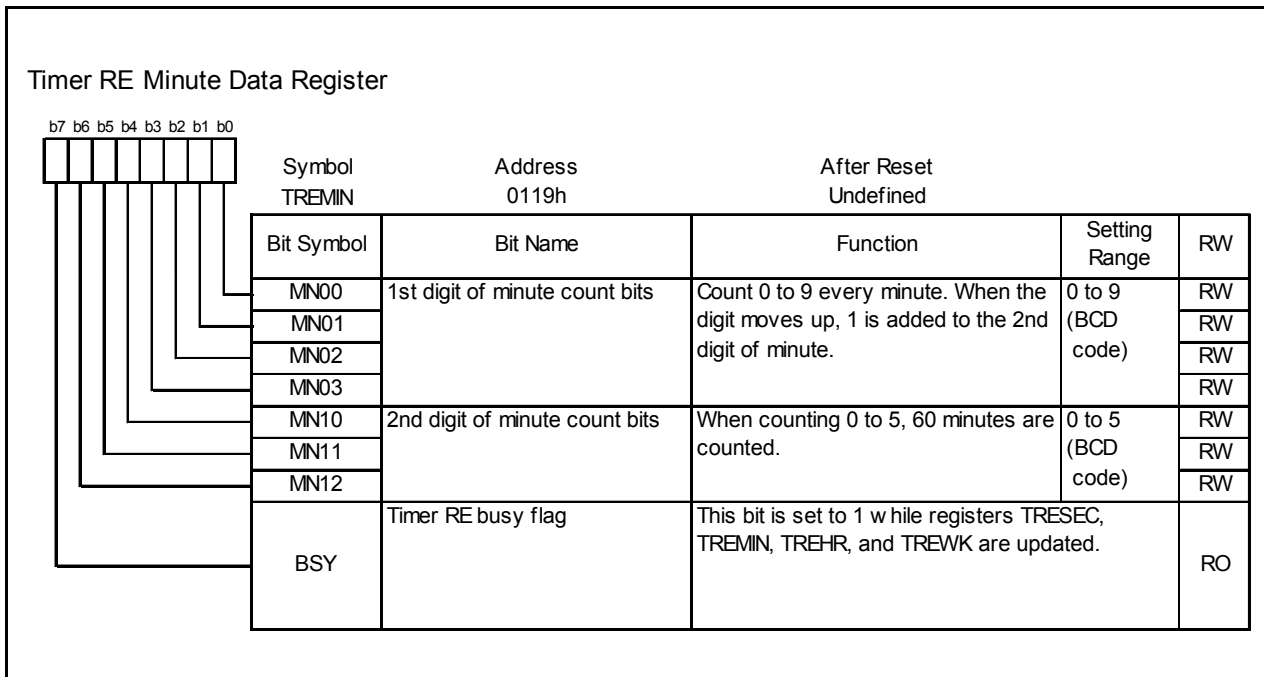


Figure 17.28 TREMIN Register in Real-Time Clock Mode

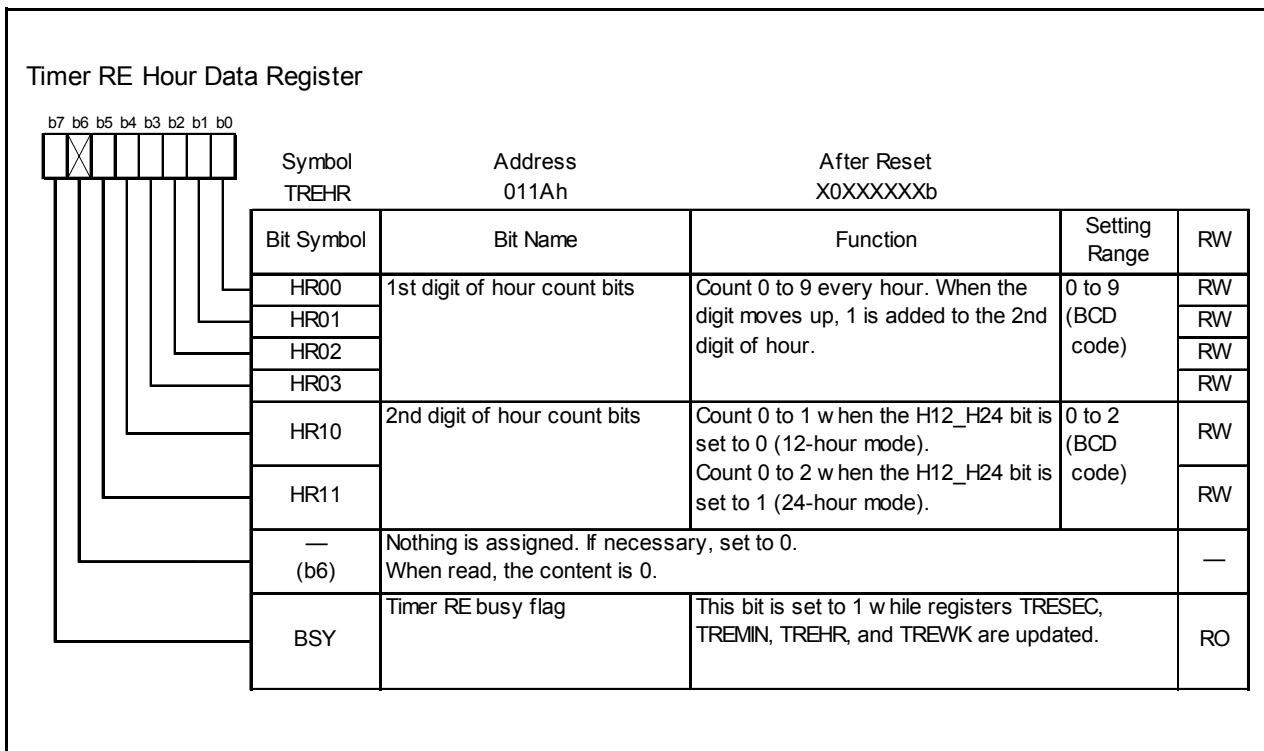


Figure 17.29 TREHR Register in Real-Time Clock Mode

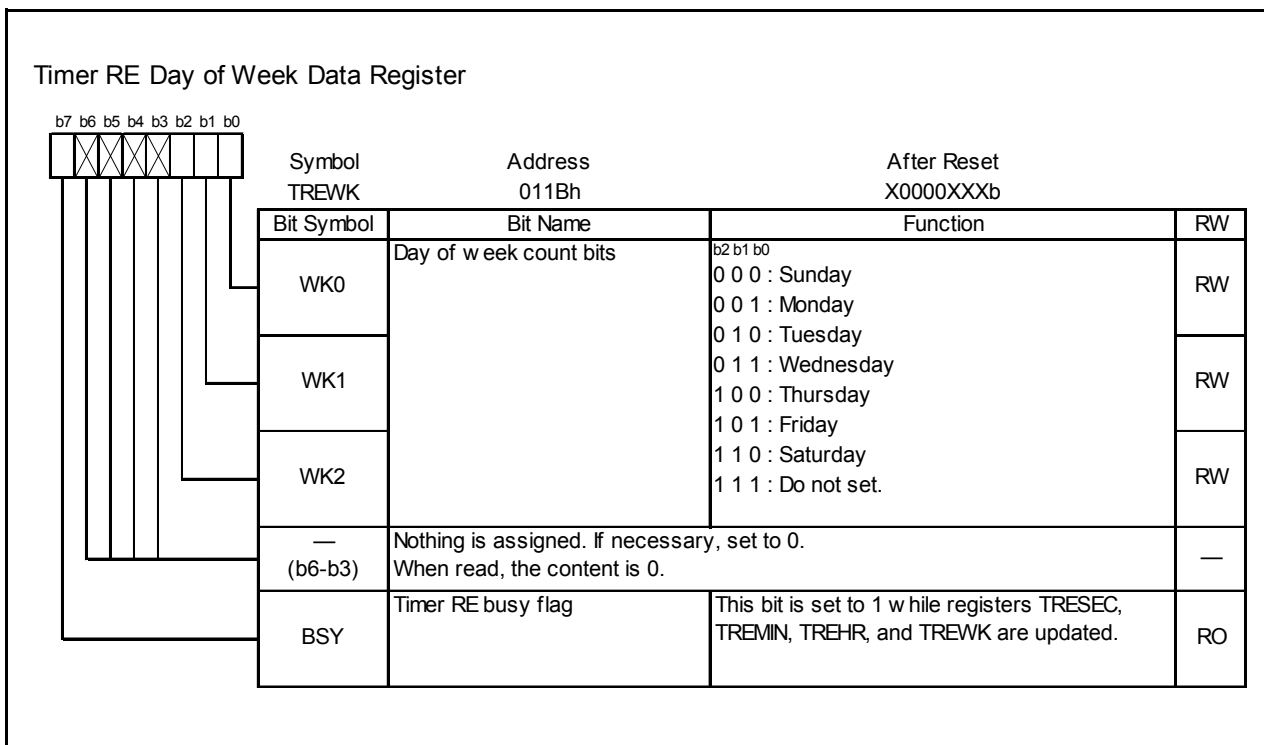


Figure 17.30 TREWK Register in Real-Time Clock Mode

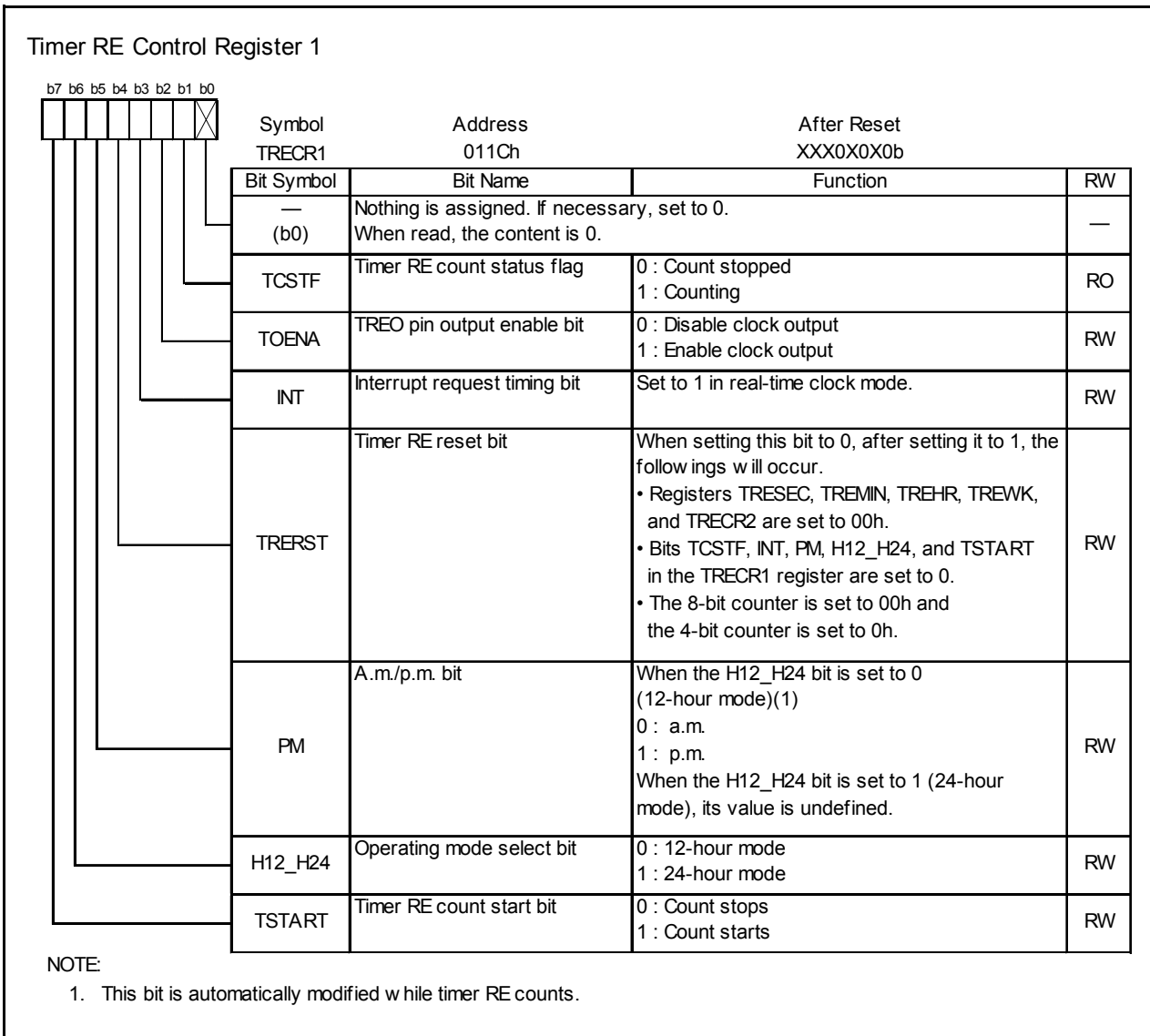


Figure 17.31 TRECR1 Register in Real-Time Clock Mode

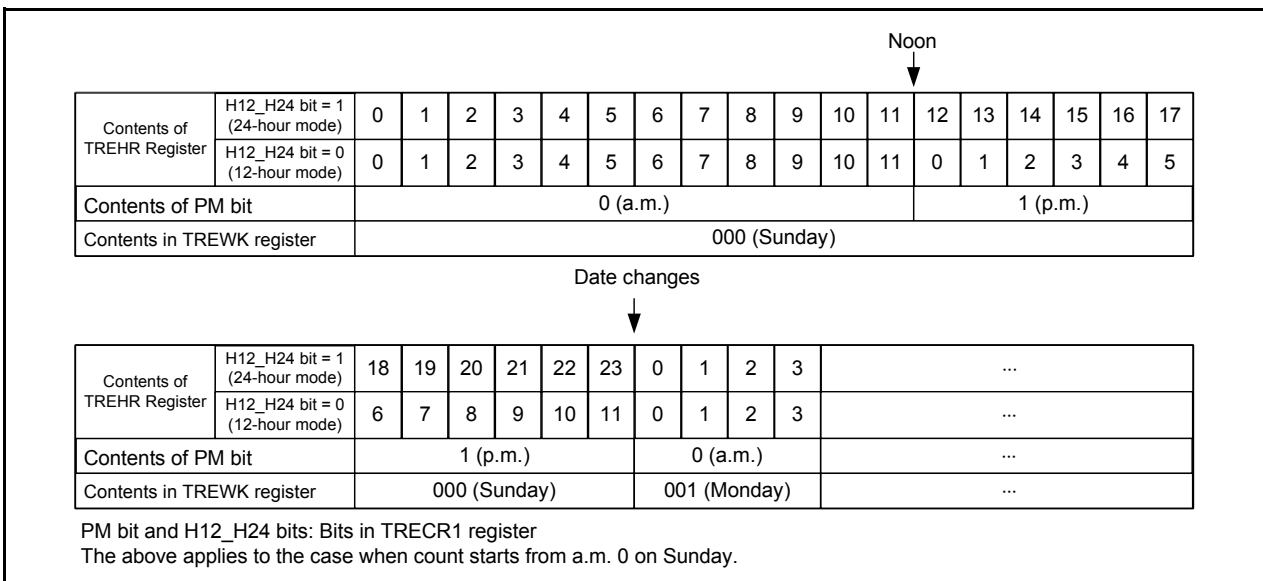


Figure 17.32 Definition of Time Representation

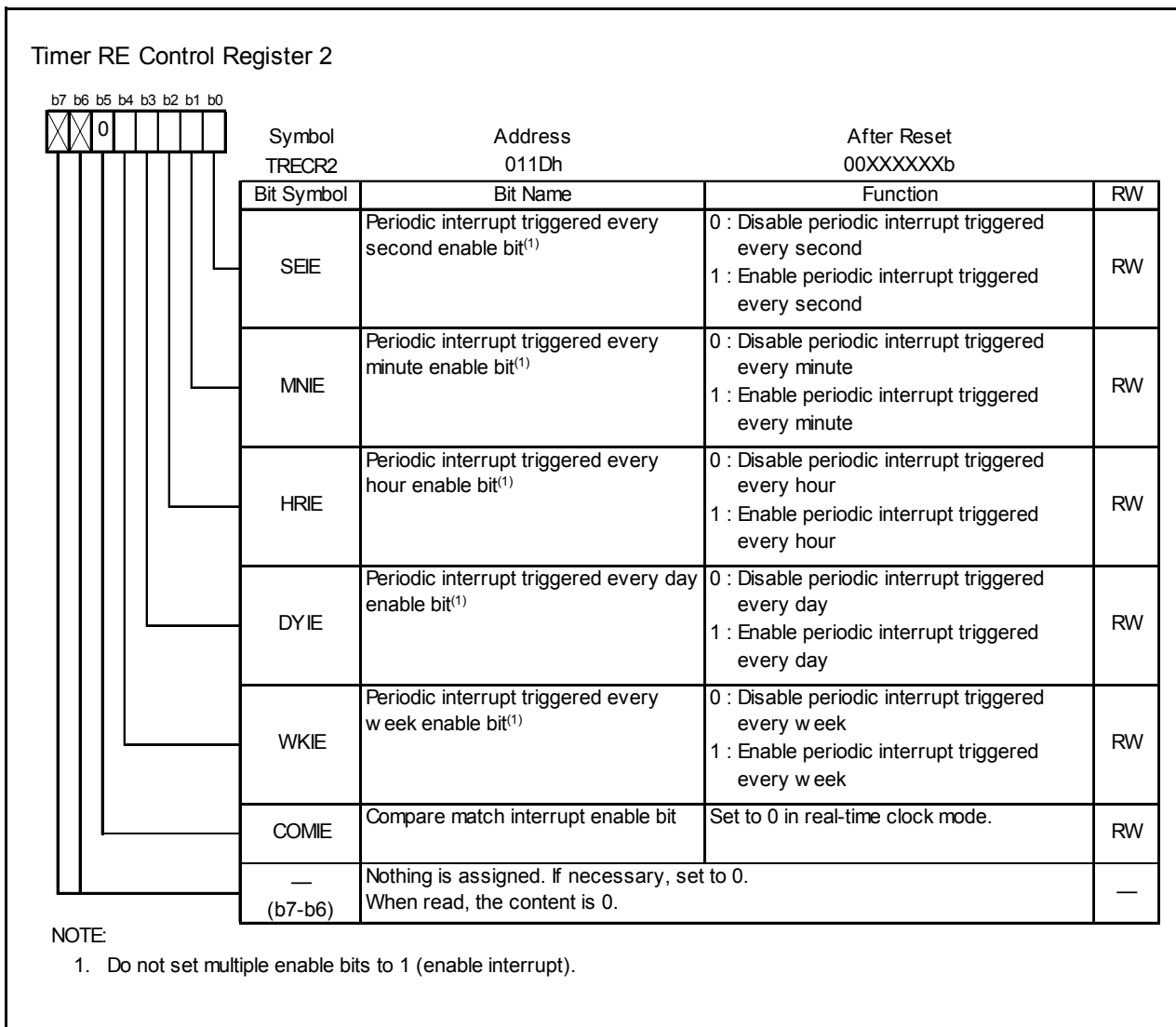
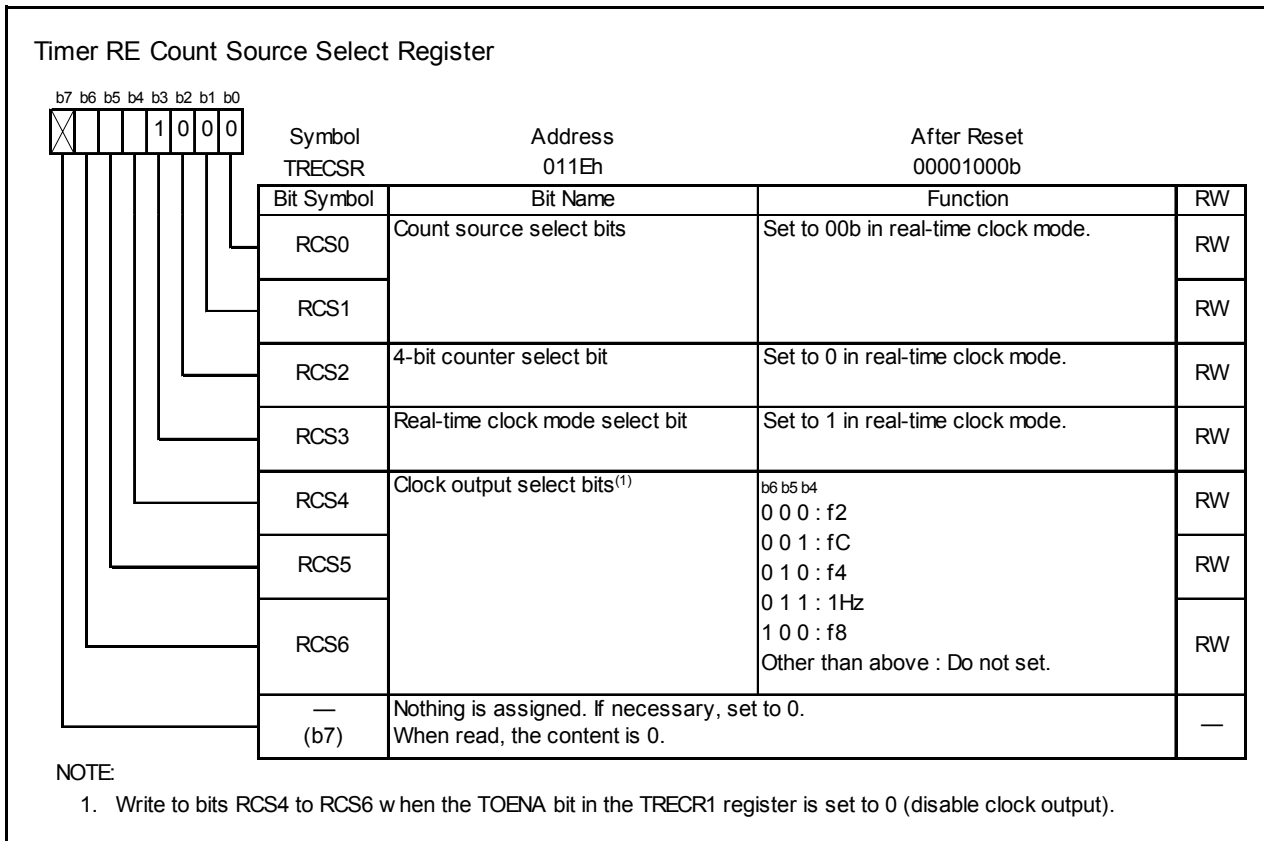


Figure 17.33 TREC2 Register in Real-Time Clock Mode

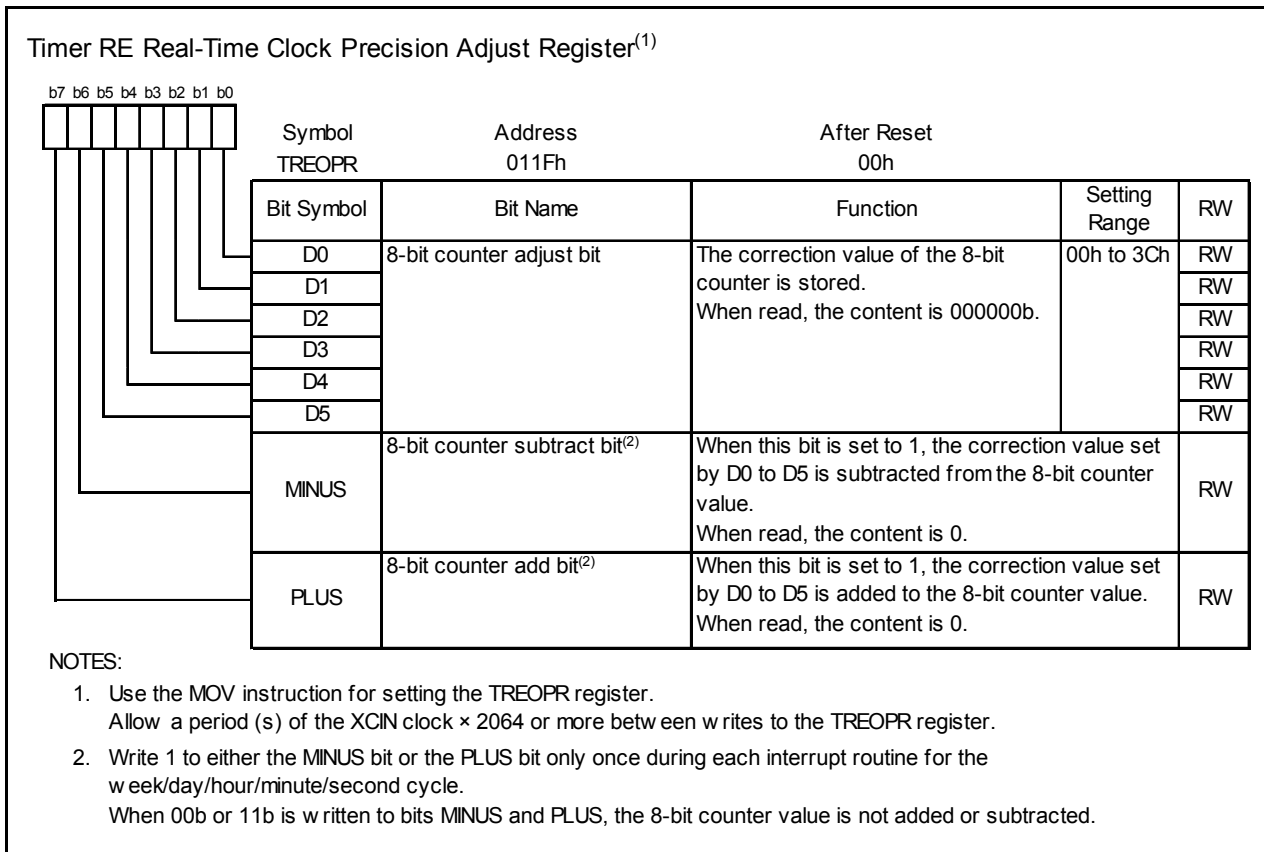
Table 17.12 Interrupt Sources

Factor	Interrupt Source	Interrupt Enable Bit
Periodic interrupt triggered every week	Value in TREWK register is set to 000b (Sunday) (1-week period)	WKIE
Periodic interrupt triggered every day	TREWK register is updated (1-day period)	DYIE
Periodic interrupt triggered every hour	TREHR register is updated (1-hour period)	HRIE
Periodic interrupt triggered every minute	TREMIN register is updated (1-minute period)	MNIE
Periodic interrupt triggered every second	TRESEC register is updated (1-second period)	SEIE





**Figure 17.34 TRECSR Register in Real-Time Clock Mode**



**Figure 17.35 TREOPR Register in Real-Time Clock Mode**

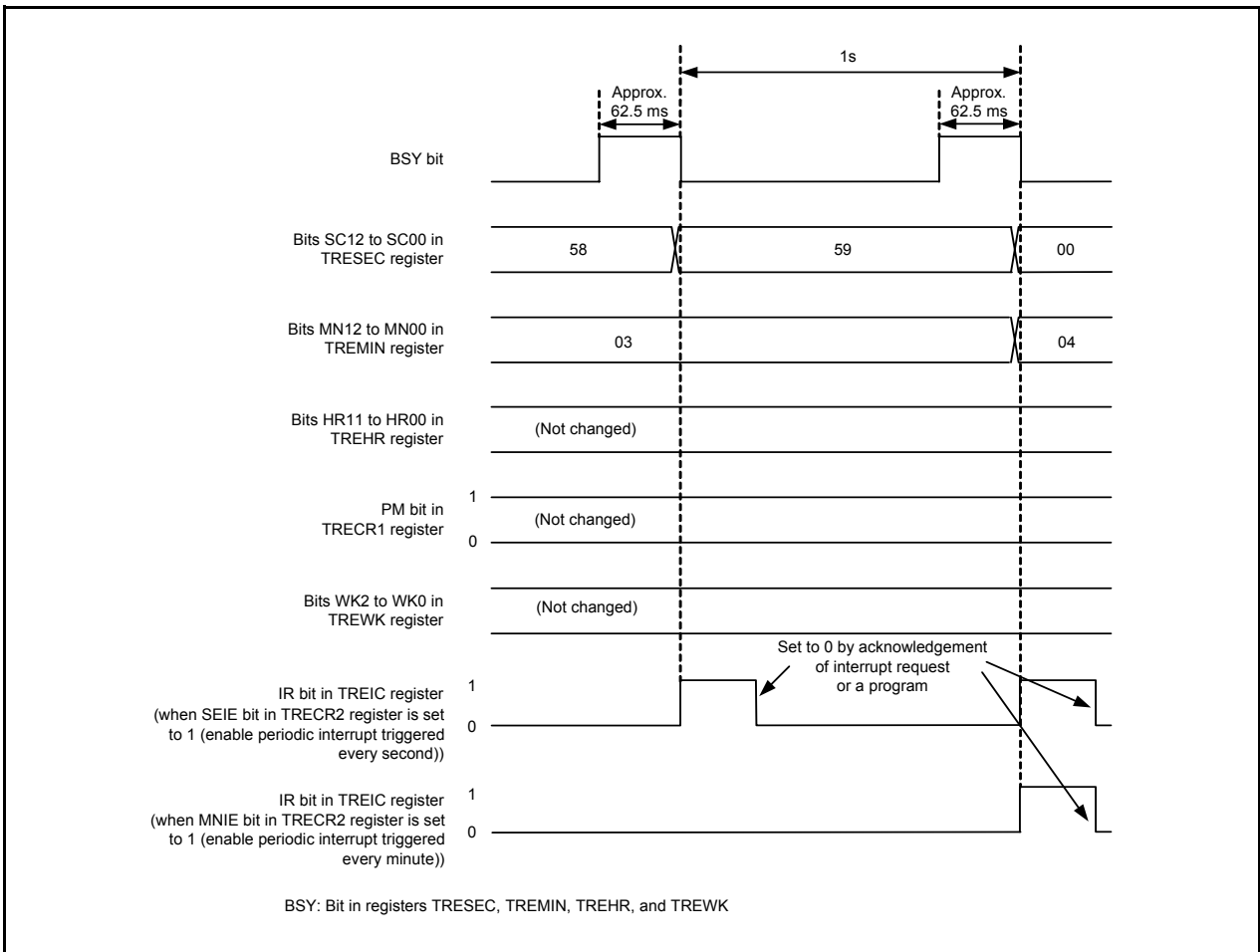


Figure 17.36 Operating Example in Real-Time Clock Mode

### 17.3.2 Output Compare Mode

In output compare mode, the internal count source divided by 2 is counted using the 4-bit or 8-bit counter and compare value match is detected with the 8-bit counter. Figure 17.37 shows a Block Diagram of Output Compare Mode and Table 17.13 lists the Output Compare Mode Specifications. Figures 17.38 to 17.42 show the Registers Associated with Output Compare Mode, and Figure 17.43 shows the Operating Example in Output Compare Mode.

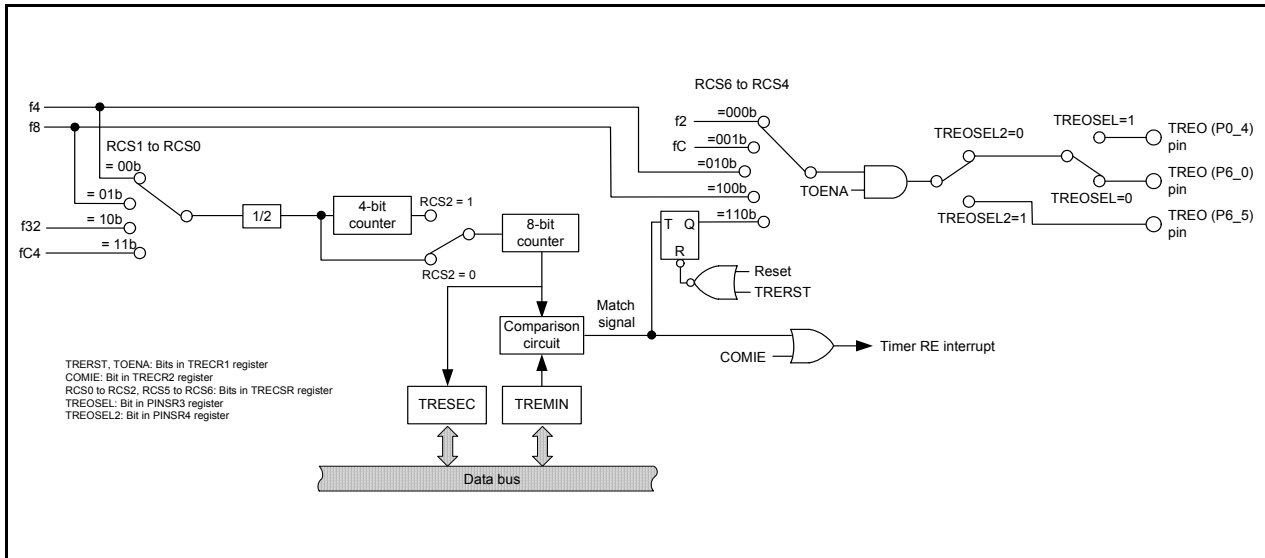


Figure 17.37 Block Diagram of Output Compare Mode

**Table 17.13 Output Compare Mode Specifications**

Item	Specification
Count sources	f4, f8, f32, fC4
Count operations	<ul style="list-style-type: none"> <li>• Increment</li> <li>• When the 8-bit counter content matches with the TREMIN register content, the value returns to 00h and count continues. The count value is held while count stops.</li> </ul>
Count period	<ul style="list-style-type: none"> <li>• When RCS2 = 0 (4-bit counter is not used)  <math>1/f_i \times 2 \times (n+1)</math></li> <li>• When RCS2 = 1 (4-bit counter is used)  <math>1/f_i \times 32 \times (n+1)</math></li> </ul> f <sub>i</sub> : Frequency of count source n: Setting value of TREMIN register
Count start condition	1 (count starts) is written to the TSTART bit in the TRECR1 register
Count stop condition	0 (count stops) is written to the TSTART bit in the TRECR1 register
Interrupt request generation timing	When the 8-bit counter content matches with the TREMIN register content
TREO pin function	Select any one of the following: <ul style="list-style-type: none"> <li>• Programmable I/O ports</li> <li>• Output f2, fC, f4, or f8</li> <li>• Compare output</li> </ul>
Read from timer	When reading the TRESEC register, the 8-bit counter value can be read. When reading the TREMIN register, the compare value can be read.
Write to timer	Writing to the TRESEC register is disabled. When bits TSTART and TCSTF in the TRECR1 register are set to 0 (timer stops), writing to the TREMIN register is enabled.
Selectable functions	<ul style="list-style-type: none"> <li>• Select use of 4-bit counter</li> <li>• Compare output function            Every time the 8-bit counter value matches the TREMIN register value, TREO output polarity is reversed. The TREO pin outputs "L" after reset is deasserted and the timer RE is reset by the TRERST bit in the TRECR1 register. Output level is held by setting the TSTART bit to 0 (count stops).</li> <li>• TREO pin select function            P0_4, P6_0, or P6_5 is selected by the TREOSEL bit in the PINSR3 register and the TREOSEL2 bit in the PINSR4 register.</li> </ul>

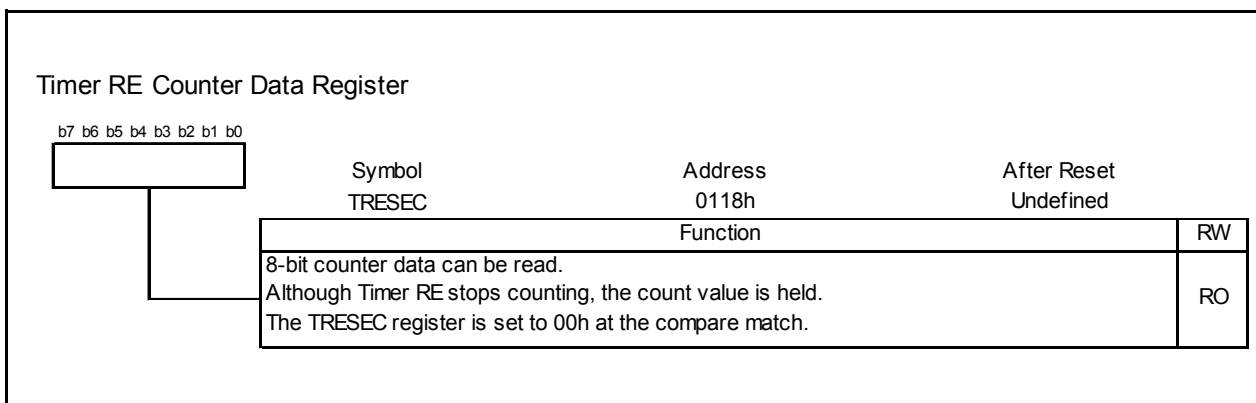


Figure 17.38 TRESEC Register in Output Compare Mode

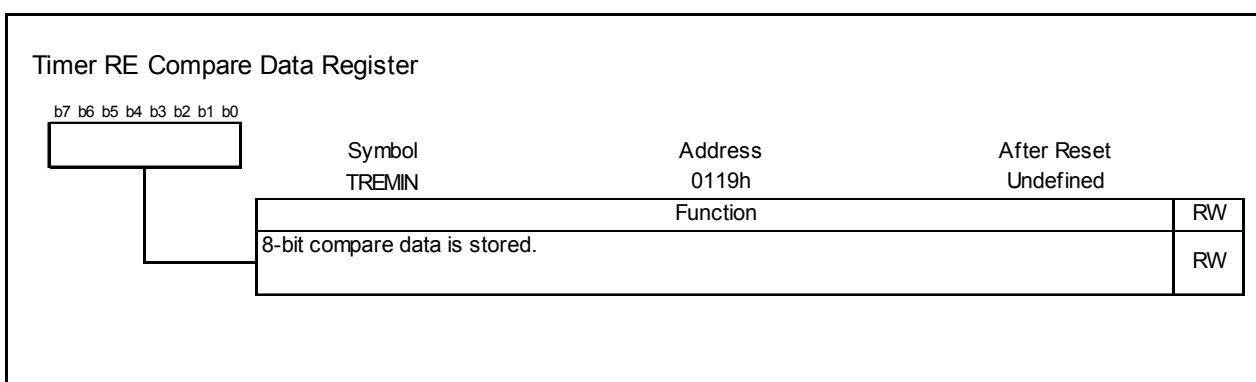


Figure 17.39 TREMINT Register in Output Compare Mode

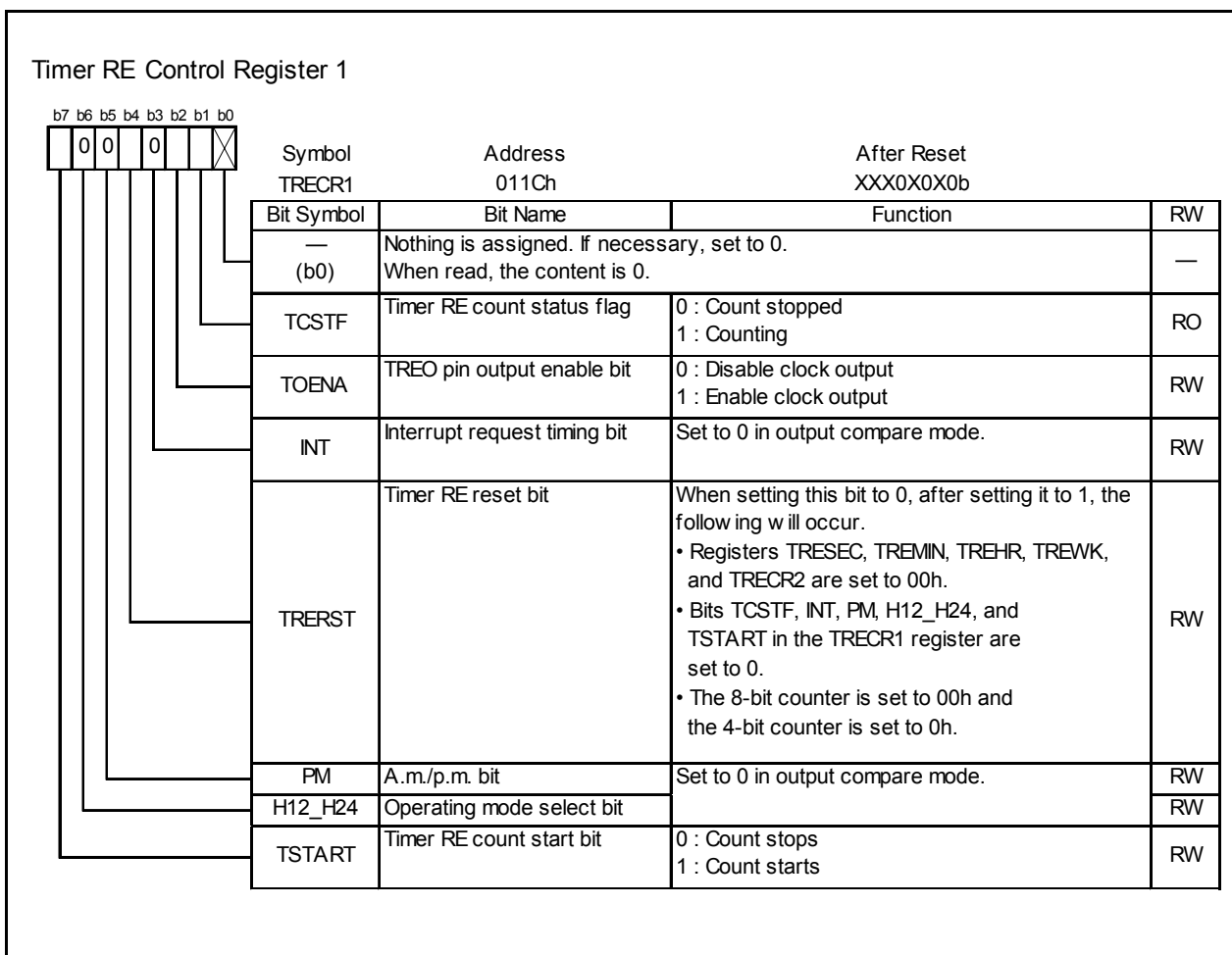


Figure 17.40 TREC1 Register in Output Compare Mode

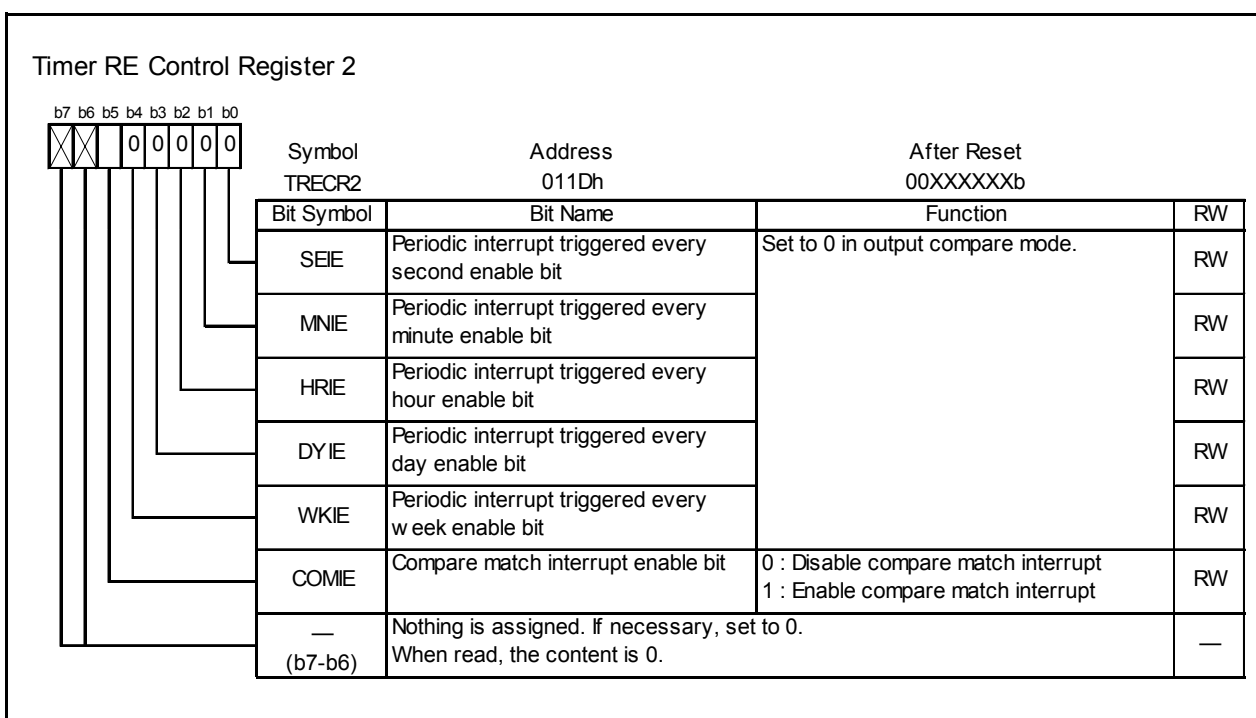


Figure 17.41 TREC2 Register in Output Compare Mode

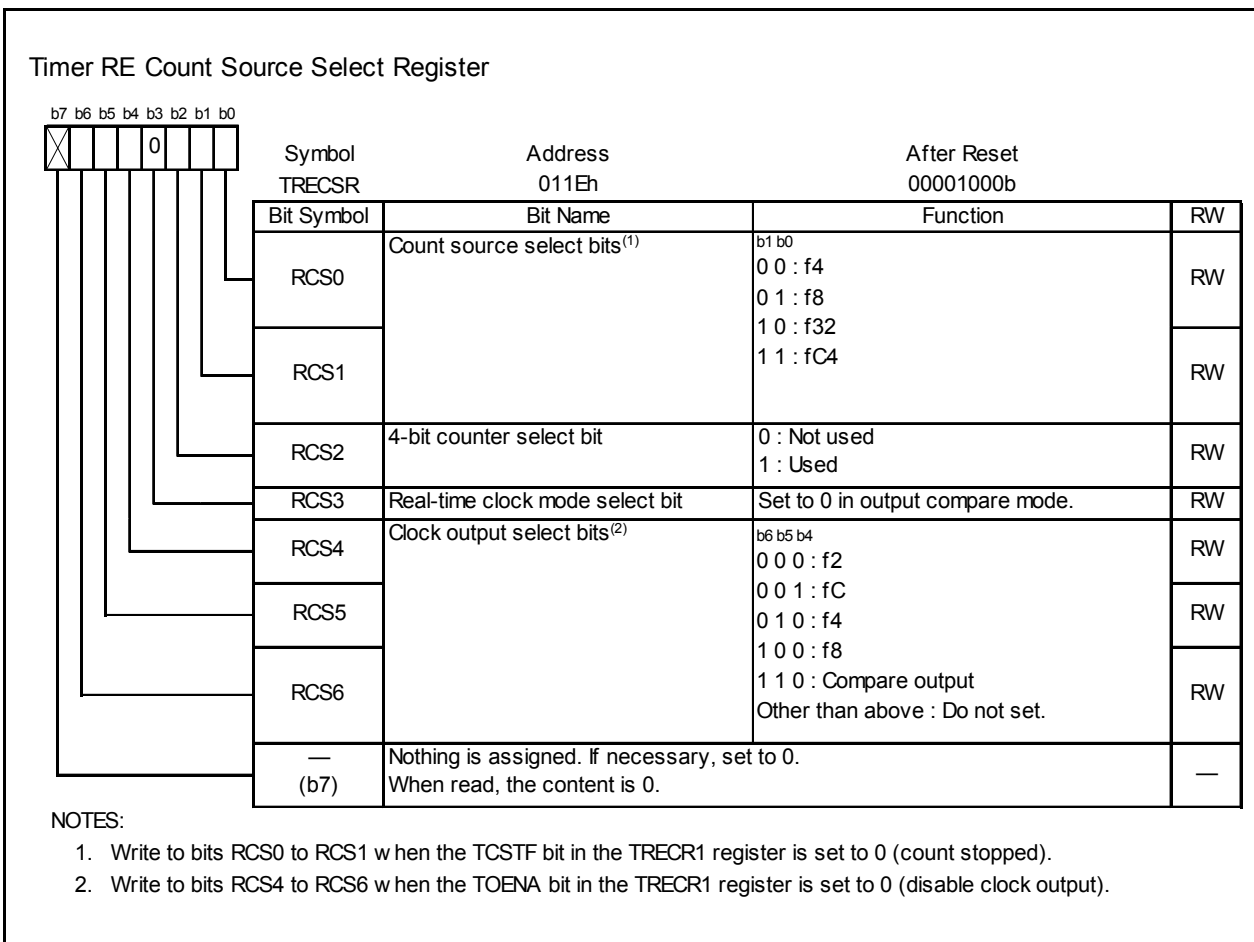


Figure 17.42 TRECSR Register in Output Compare Mode

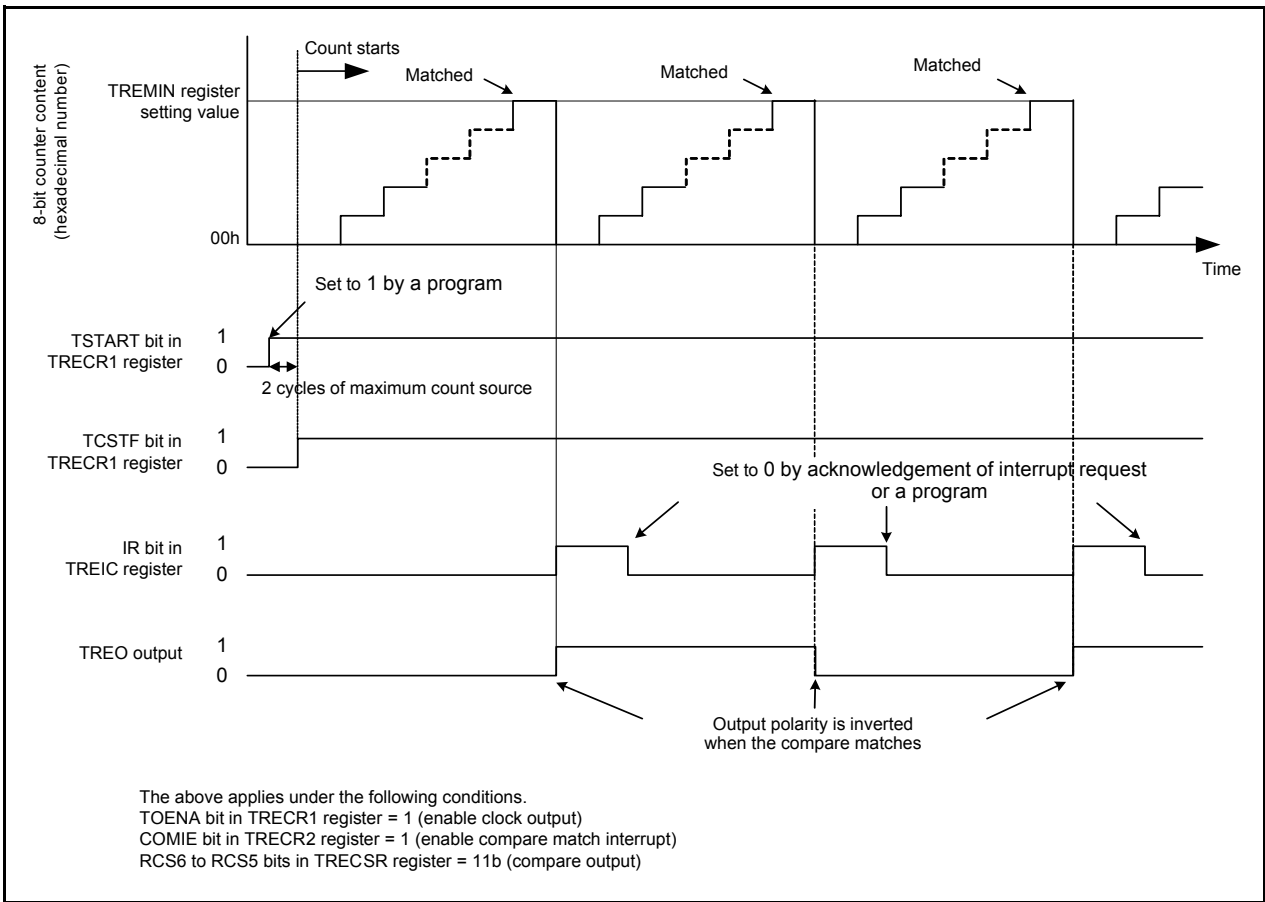


Figure 17.43 Operating Example in Output Compare Mode



### 17.3.3 Notes on Timer RE

#### 17.3.3.1 Starting and Stopping Count

Timer RE has the TSTART bit for instructing the count to start or stop, and the TCSTF bit, which indicates count start or stop. Bits TSTART and TCSTF are in the TRECRC1 register.

Timer RE starts counting and the TCSTF bit is set to 1 (count starts) when the TSTART bit is set to 1 (count starts). It takes up to 2 cycles of the count source until the TCSTF bit is set to 1 after setting the TSTART bit to 1. During this time, do not access registers associated with timer RE<sup>(1)</sup> other than the TCSTF bit.

Also, timer RE stops counting when setting the TSTART bit to 0 (count stops) and the TCSTF bit is set to 0 (count stops). It takes the time for up to 2 cycles of the count source until the TCSTF bit is set to 0 after setting the TSTART bit to 0. During this time, do not access registers associated with timer RE other than the TCSTF bit.

NOTE:

1. Registers associated with timer RE: TRESEC, TREMIN, TREHR, TREWK, TRECRC1, TRECRC2, TRECSCR, and TRECOPR.

#### 17.3.3.2 Register Setting

Write to the following registers or bits when timer RE is stopped.

- Registers TRESEC, TREMIN, TREHR, TREWK, and TRECRC2
- Bits H12\_H24, PM, and INT in TRECRC1 register
- Bits RCS0 to RCS3 in TRECSCR register

Timer RE is stopped when bits TSTART and TCSTF in the TRECRC1 register are set to 0 (timer RE stopped).

Also, set all above-mentioned registers and bits (immediately before timer RE count starts) before setting the TRECRC2 register.

Figure 17.44 shows a Setting Example in Real-Time Clock Mode.

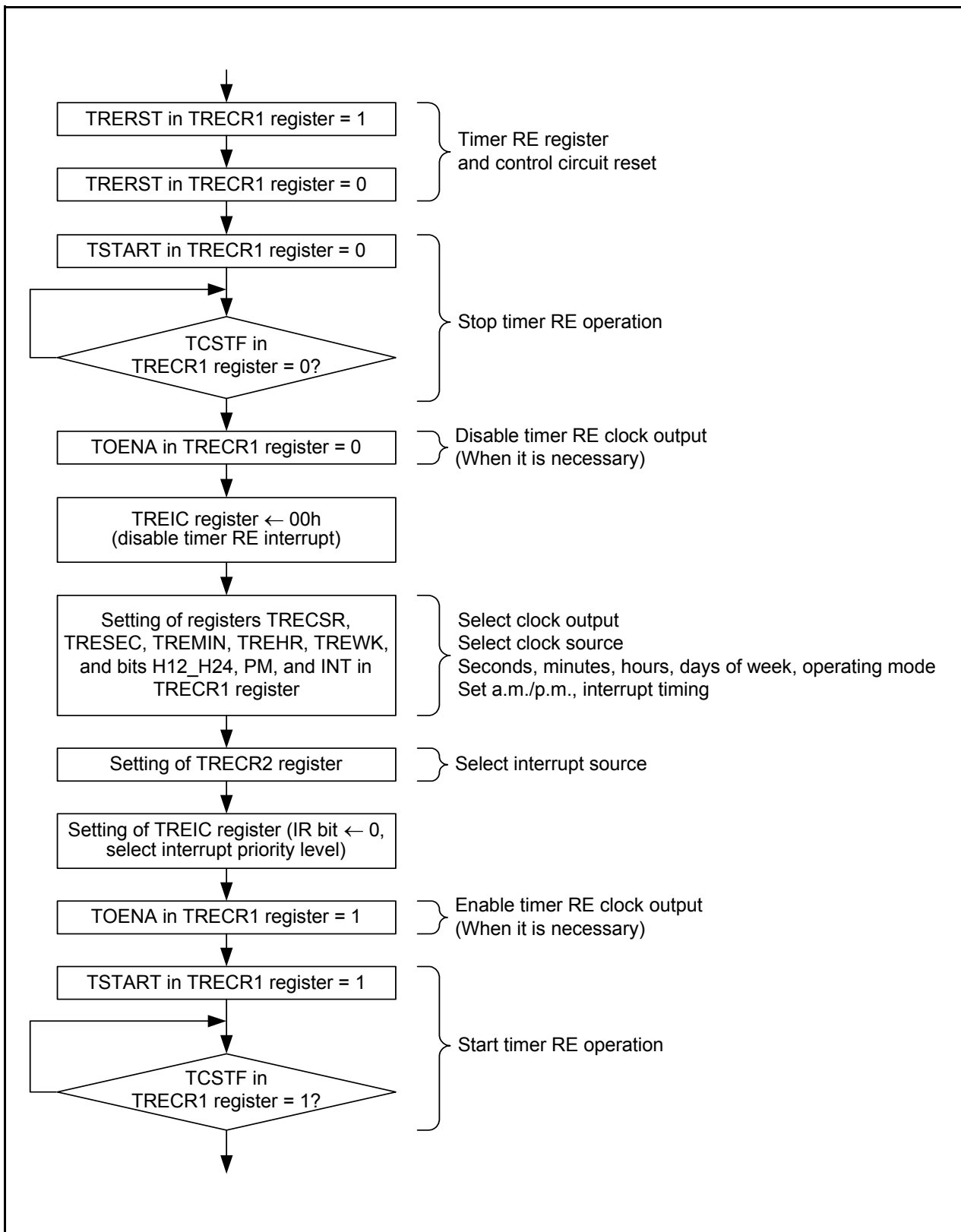


Figure 17.44 Setting Example in Real-Time Clock Mode

### 17.3.3.3 Time Reading Procedure of Real-Time Clock Mode

In real-time clock mode, read registers TRESEC, TREMIN, TREHR, and TREWK when time data is updated and read the PM bit in the TRECRC1 register when the BSY bit is set to 0 (not while data is updated).

Also, when reading several registers, an incorrect time will be read if data is updated before another register is read after reading any register.

In order to prevent this, use the reading procedure shown below.

- Using an interrupt  
Read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECRC1 register in the timer RE interrupt routine.
- Monitoring with a program 1  
Monitor the IR bit in the TREIC register with a program and read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECRC1 register after the IR bit in the TREIC register is set to 1 (timer RE interrupt request generated).
- Monitoring with a program 2
  - (1) Monitor the BSY bit.
  - (2) Monitor until the BSY bit is set to 0 after the BSY bit is set to 1 (approximately 62.5 ms while the BSY bit is set to 1).
  - (3) Read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECRC1 register after the BSY bit is set to 0.
- Using read results if they are the same value twice
  - (1) Read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECRC1 register.
  - (2) Read the same register as (1) and compare the contents.
  - (3) Recognize as the correct value if the contents match. If the contents do not match, repeat until the read contents match with the previous contents.

Also, when reading several registers, read them as continuously as possible.

### 17.4 Timer RF

Timer RF is a 16-bit timer. The count source for timer RF is the operating clock that regulates the timing of timer operations. Figure 17.45 shows a Block Diagram of Timer RF. Figure 17.46 shows a Block Diagram of CMP Waveform Generation Unit. Figure 17.47 shows a Block Diagram of CMP Waveform Output Unit. Timer RF has two modes: input capture mode and output compare mode. Figures 17.48 to 17.51 show the timer RF associated registers.

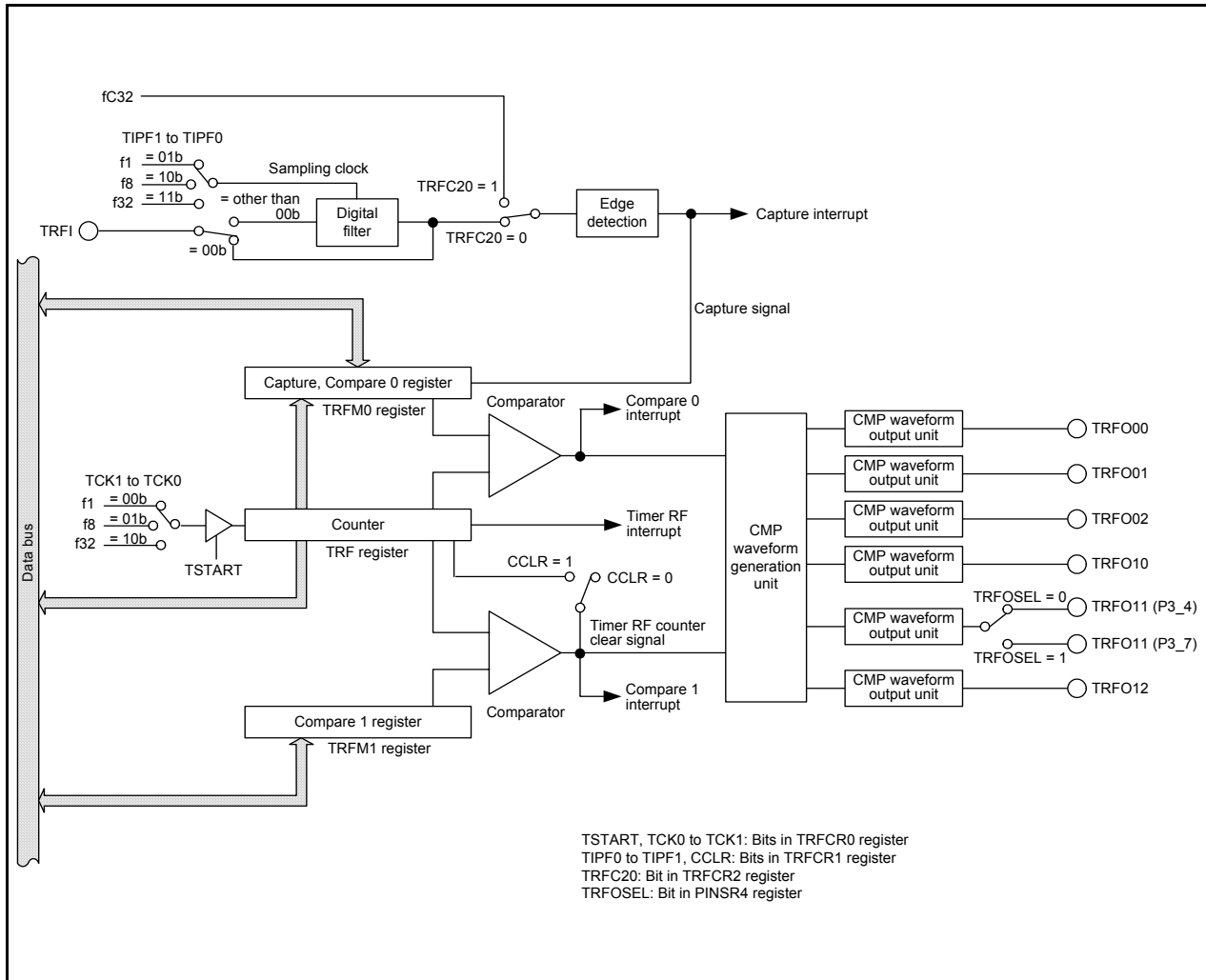


Figure 17.45 Block Diagram of Timer RF

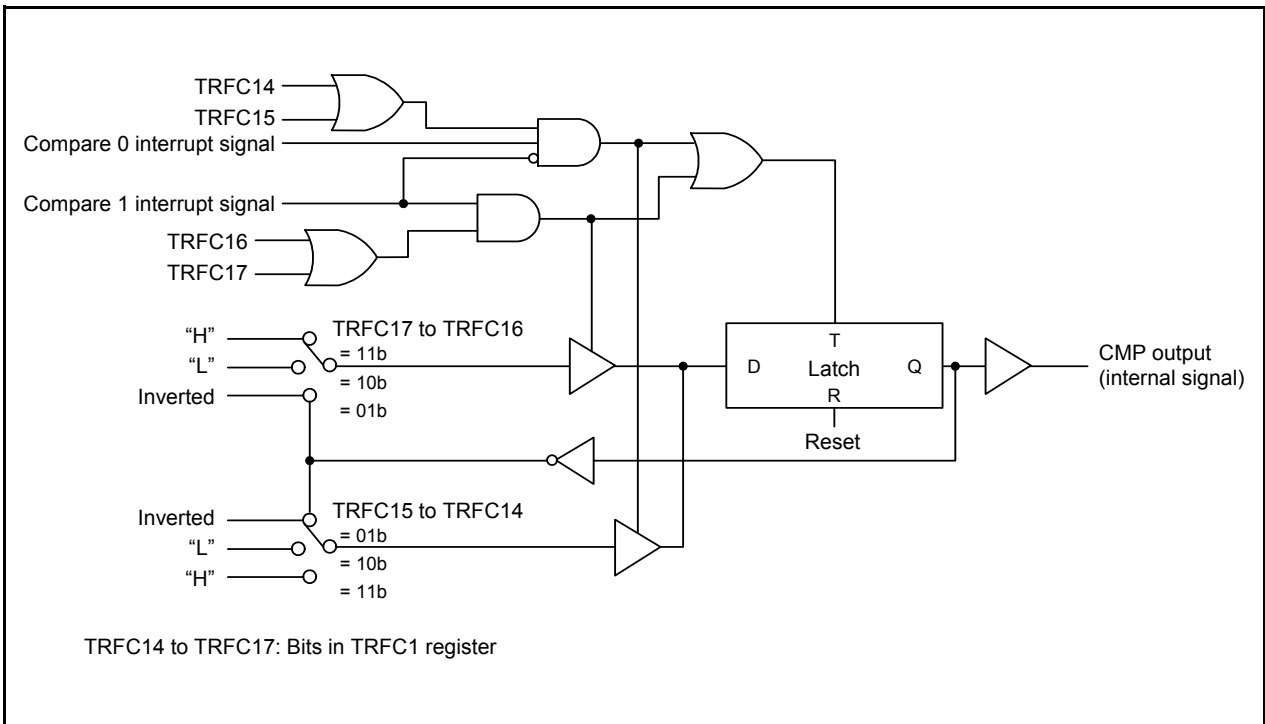


Figure 17.46 Block Diagram of CMP Waveform Generation Unit

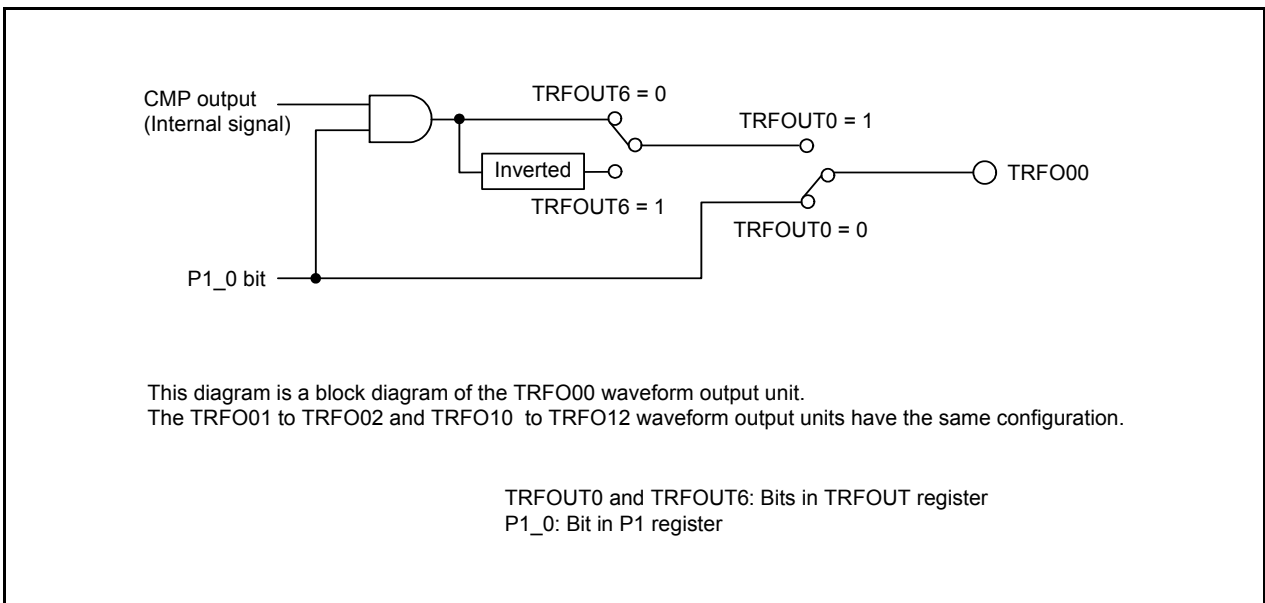
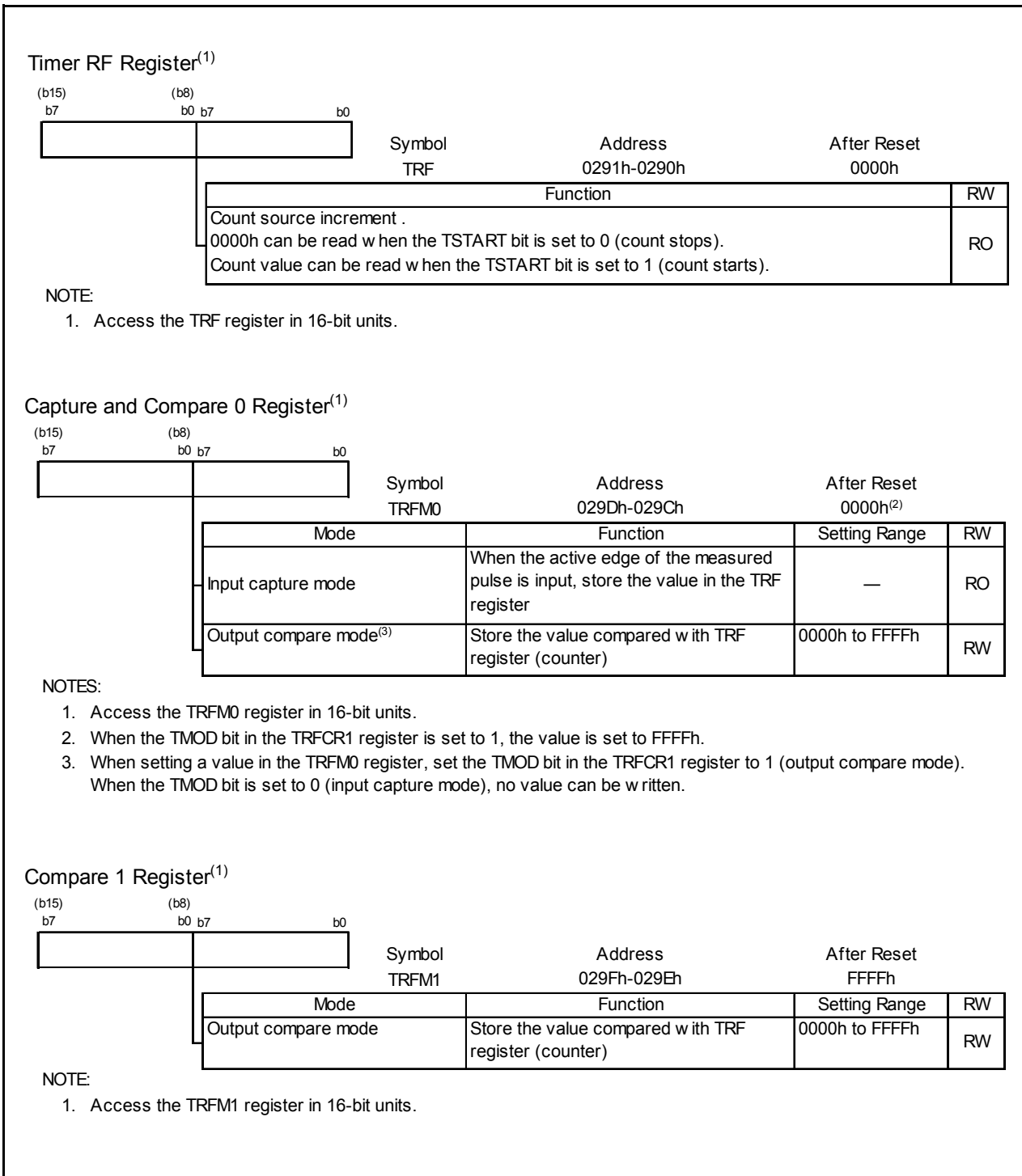


Figure 17.47 Block Diagram of CMP Waveform Output Unit



**Figure 17.48 Registers TRF, TRFM0, and TRFM1**

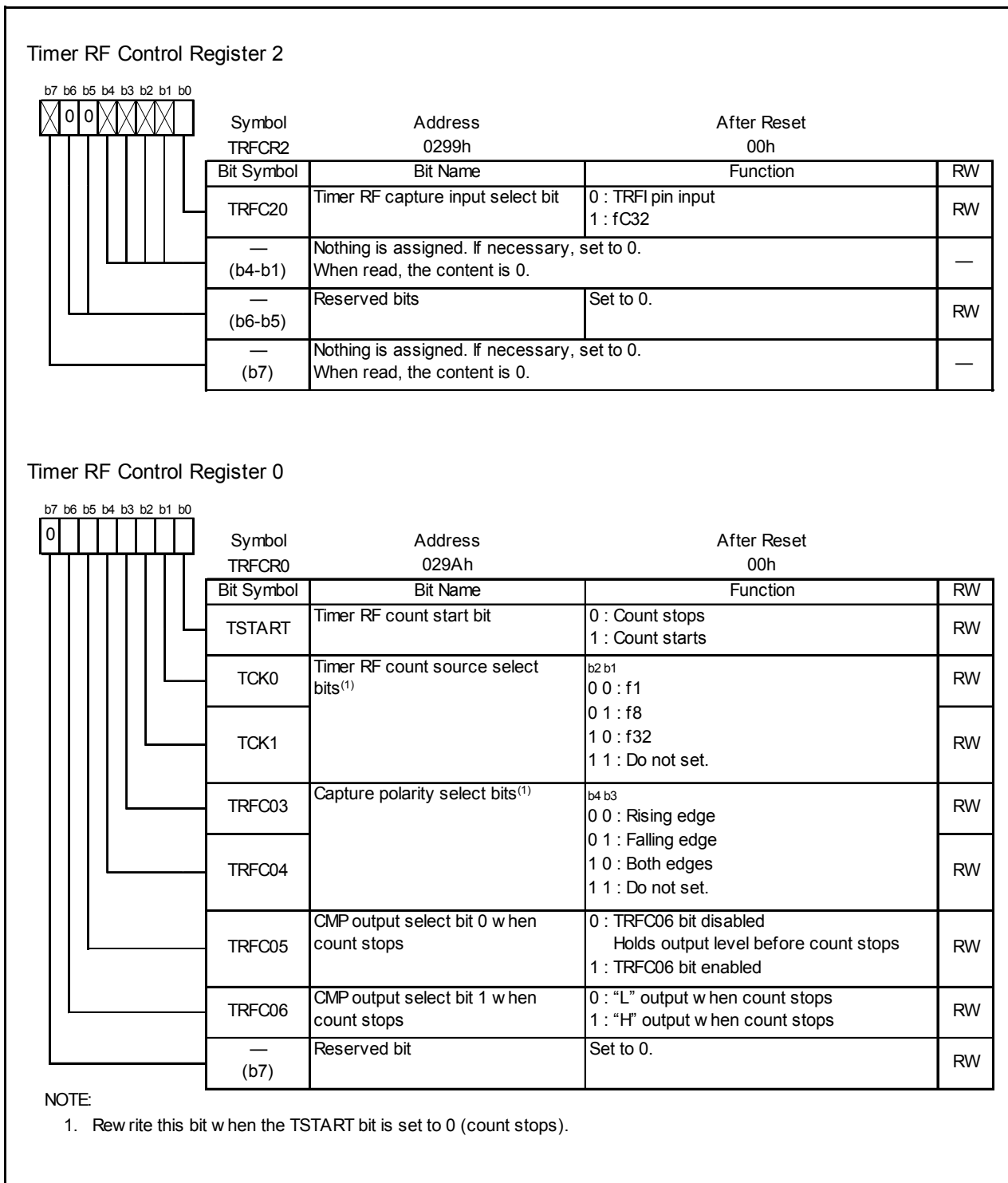
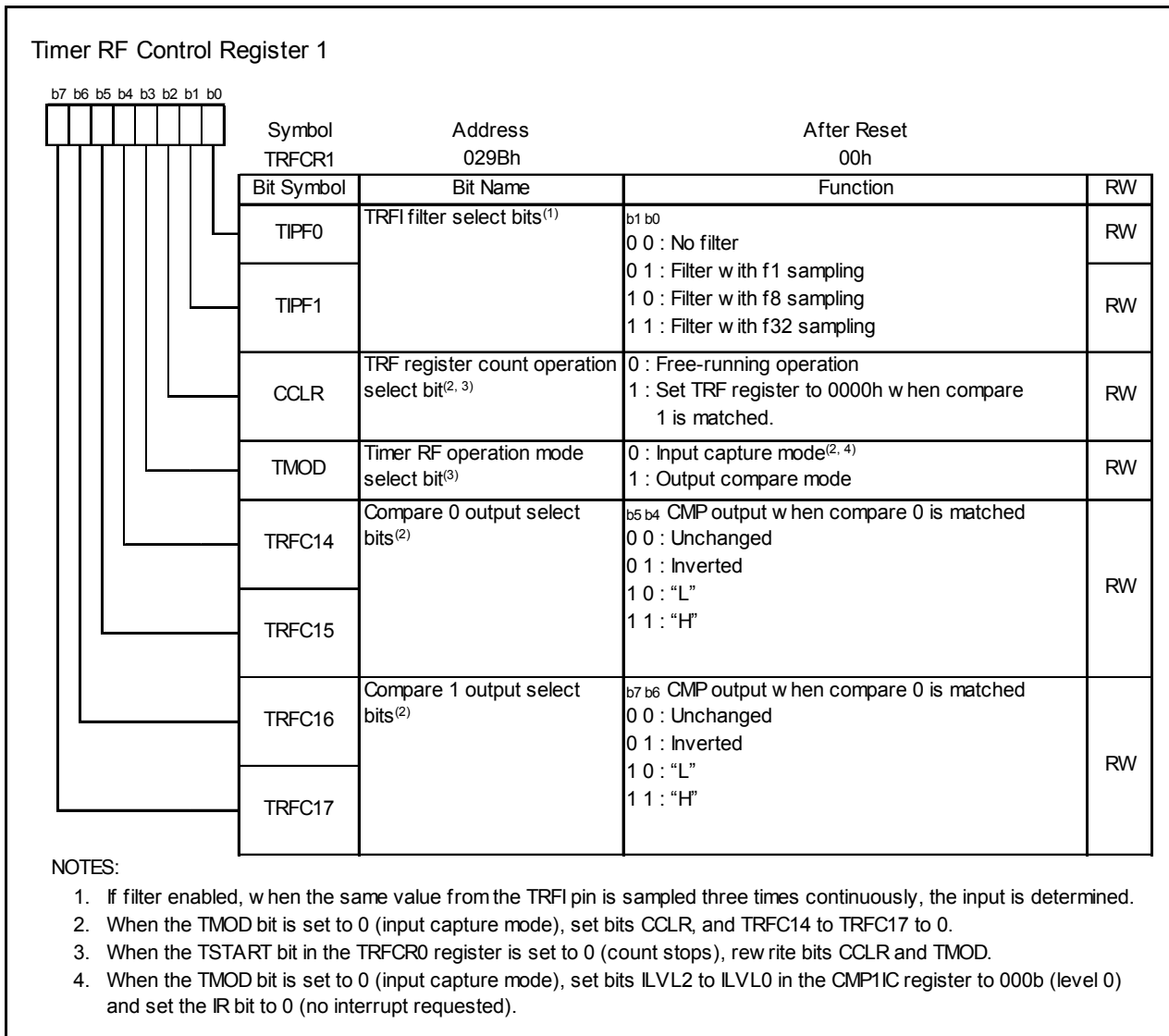
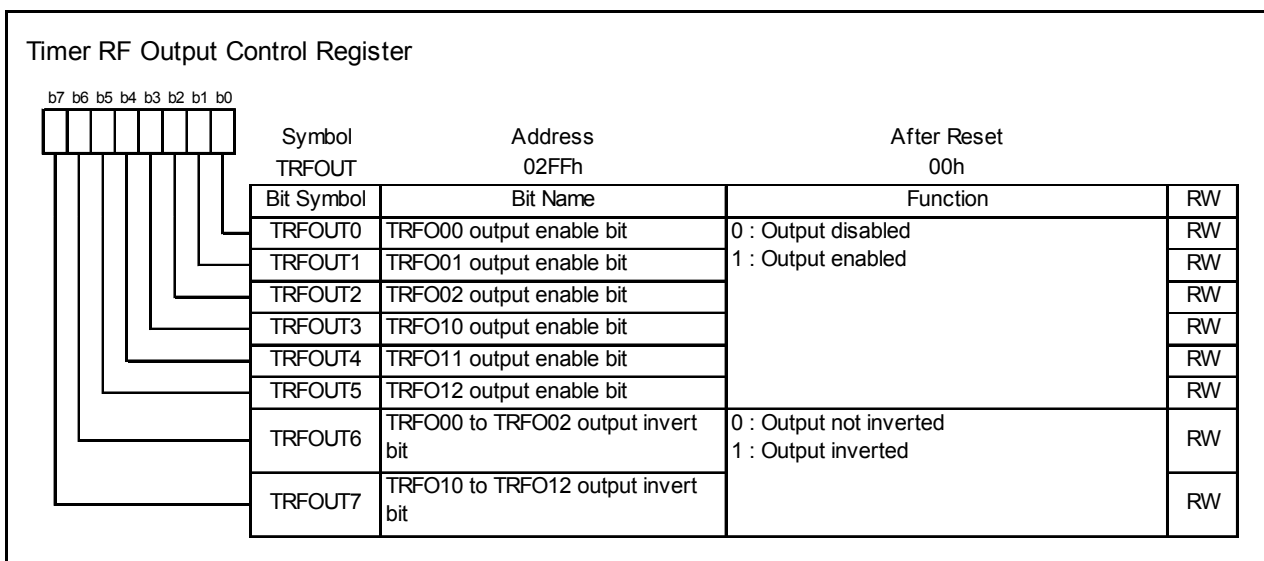


Figure 17.49 Registers TRFCR2 and TRFCR0



**Figure 17.50 TRFCR1 Register**



**Figure 17.51 TRFOUT Register**



### 17.4.1 Input Capture Mode

In input capture mode, the edge of the TRFI pin input signal or fC32 is used as a trigger to latch the timer value and the width or the period of external signal is measured. The TRFI input is equipped with a digital filter, and this prevents errors caused by noise or the like from occurring. Table 17.14 shows the Input Capture Mode Specifications. Figure 17.52 shows an Operating Example in Input Capture Mode.

**Table 17.14 Input Capture Mode Specifications**

Item	Specification
Count sources	f1, f8, f32
Count operations	<ul style="list-style-type: none"> <li>• Increment</li> <li>• Transfer the value in the TRF register to the TRFM0 register at the valid edge of the measured pulse.</li> </ul>
Count period	$1/fk \times 65536$ fk: Frequency of count source
Count start condition	The TSTART bit in the TRFCR0 register is set to 1 (count starts).
Count stop condition	The TSTART bit in the TRFCR0 register is set to 0 (count stops).
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• The valid edge of TRFI input or fC32 [capture interrupt]</li> <li>• When timer RF overflows [timer RF interrupt]</li> </ul>
TRFI pin function	Measured pulse input
TRFO00 to TRFO02, TRFO11 to TRFO12 pin functions	Programmable I/O port
Counter value reset timing	In the following cases, the value in the TRF register is set to 0000h. <ul style="list-style-type: none"> <li>• When the TSTART bit in the TRFCR0 register is set to 0 (count stops).</li> </ul>
Read from timer	<ul style="list-style-type: none"> <li>• The count value can be read out by reading the TRF register.</li> <li>• The count value at the measured pulse valid edge input can be read out by reading the TRFM0 register.</li> </ul>
Write to timer	Write to the TRF and TRFM0 registers is disabled.
Select functions	<ul style="list-style-type: none"> <li>• TRFI or fC32 polarity selected Selects the valid edge of the measured pulse. (Bits TRFC03 to TRFC04 in the TRFCR0 register.)</li> <li>• Digital filter function The TRFI input is sampled, and when the sampled input level matches as three times, the level is determined. Selects the sampling clock of the digital filter. (Bits TIPF0 to TIPF1 in the TRFCR1 register.)</li> </ul>

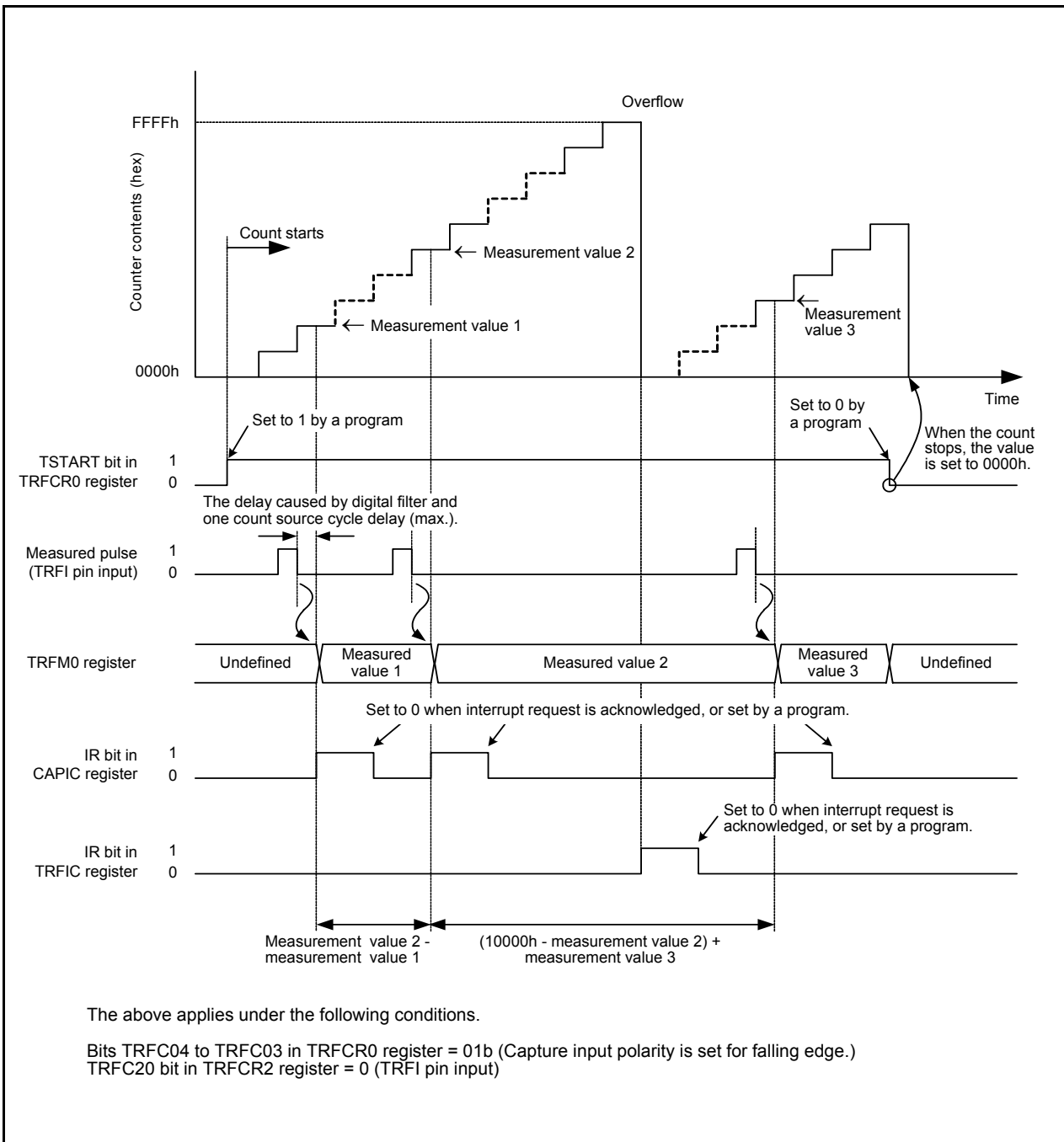


Figure 17.52 Operating Example in Input Capture Mode

### 17.4.1.1 Digital Filter

The TRFI input is sampled, and when the sampled input level matches three times, its level is determined. Select the digital filter function and sampling clock by the TRFCR1 register.

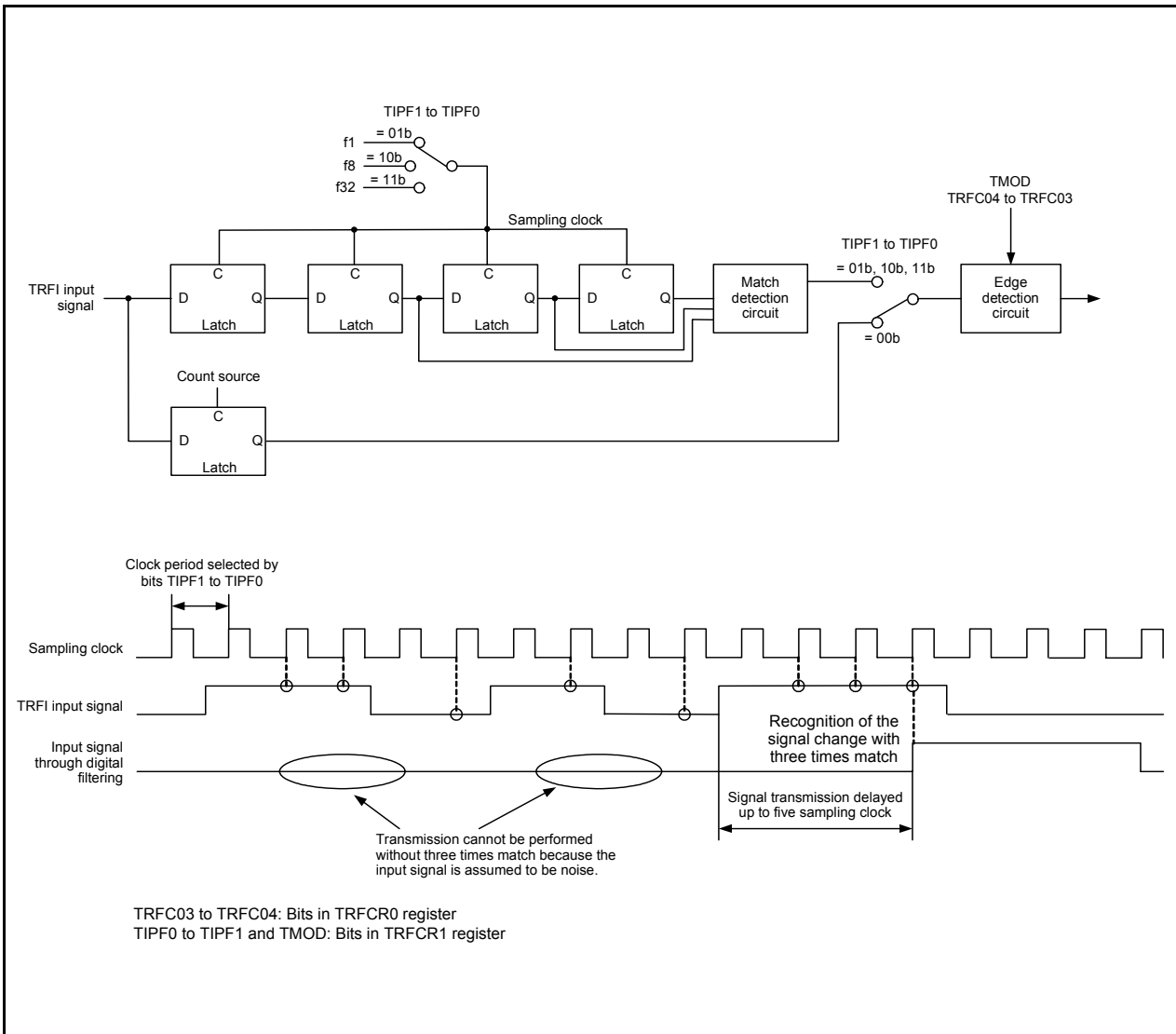


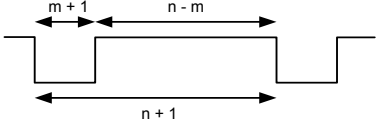
Figure 17.53 Block Diagram of Digital Filter

### 17.4.2 Output Compare Mode

In output compare mode, when the value of the TRF register matches the value of the TRFM0 (compare 0 match) or TRFM1 (compare 1 match) register, a user-set level is output mode from the output-compare output pin.

Table 17.15 shows the Output Compare Mode Specifications. Table 17.16 shows the Output in Output Compare Mode (Example of TRFO00 Pin). Figure 17.54 shows an Operating Example in Output Compare Mode. Figure 17.55 shows an Operating Example in Output Compare Mode (“L” and “H” Held Output in Count Stops).

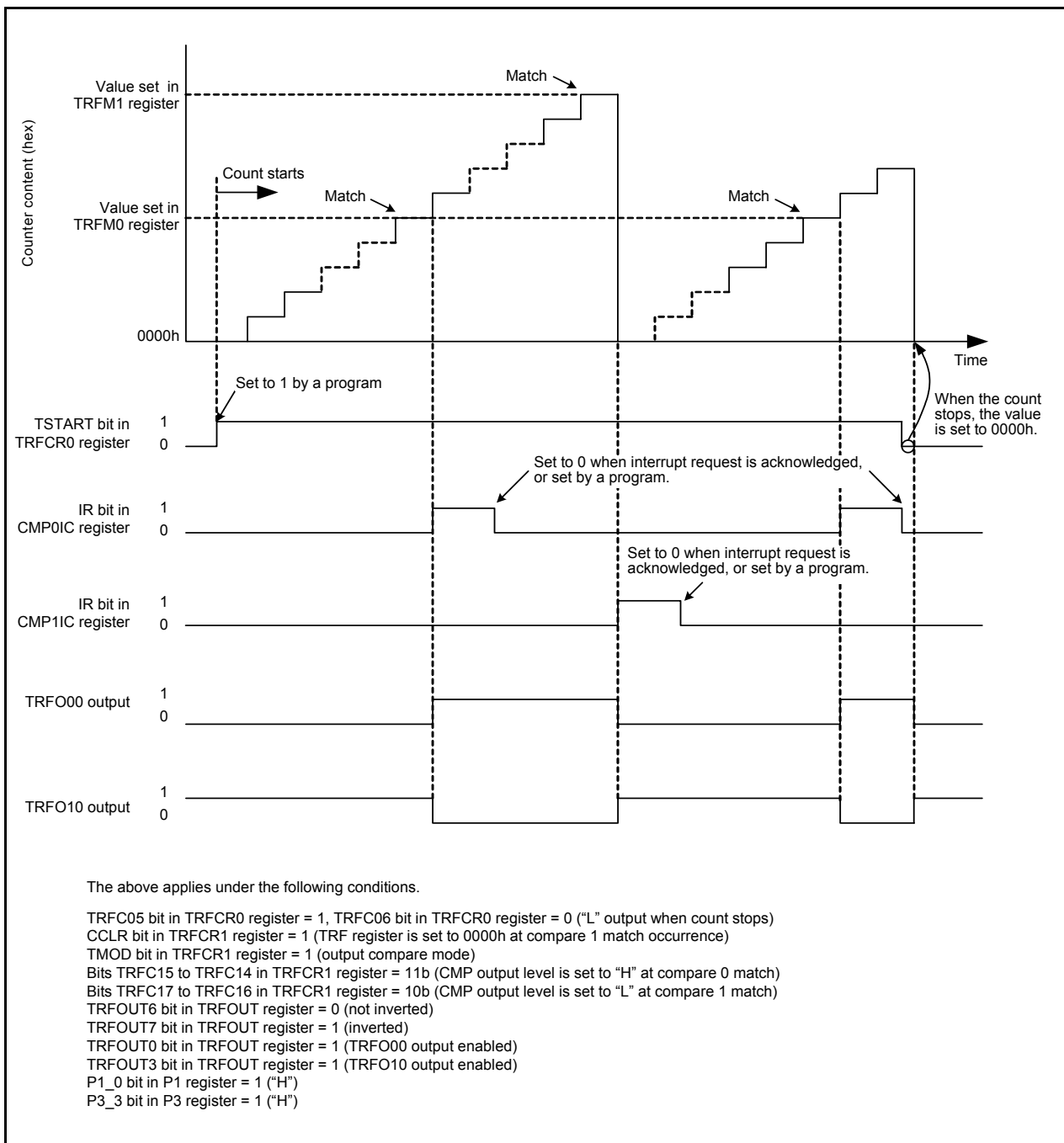
**Table 17.15 Output Compare Mode Specifications**

Item	Specification
Count sources	f1, f8, f32
Count operations	Increment
PWM waveform	<p>PWM period: <math>1/fk \times (n + 1)</math>  “L” level width: <math>1/fk \times (m + 1)</math>  “H” level width: <math>1/fk \times (n - m)</math>  fk: Frequency of count source  m: Value set in the TRFM0 register  n: Value set in the TRFM1 register</p>  <p>It applies under the following conditions.</p> <ul style="list-style-type: none"> <li>• CMP output “H” when compare 0 is matched</li> <li>• CMP output “L” when compare 1 is matched</li> <li>• CMP output not inverted</li> </ul>
Count start condition	The TSTART bit in the TRFCR0 register is set to 1 (count starts).
Count stop condition	The TSTART bit in the TRFCR0 register is set to 0 (count stops).
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When compare 0 match is generated [compare 0 interrupt]</li> <li>• When compare 1 match is generated [compare 1 interrupt]</li> <li>• When time RF overflows [timer RF interrupt].</li> </ul>
TRFO00 to TRFO12 pin functions	Programmable I/O port or output-compare output
Counter value reset timing	<p>In the following cases, the value in the TRF register is set to 0000h.</p> <ul style="list-style-type: none"> <li>• When the TSTART bit in the TRFCR0 register is set to 0 (count stops).</li> <li>• The CCLR bit in the TRFCR1 register is set to 1 (the TRF register is set to 0000h at compare 1 match) in the compare 1 matches.</li> </ul>
Read from timer	<ul style="list-style-type: none"> <li>• The count value can be read out by reading the TRF register.</li> <li>• The value in the compare register can be read out by reading registers TRFM0 and TRFM1.</li> </ul>
Write to timer	Write to the TRF register is disabled
Select functions	<ul style="list-style-type: none"> <li>• Output-compare output pin selected  Either 1 pin or multiple pins among TRFO00 to TRFO02, or TRFO10 to TRFO12 (bits TRFOOUT0 to TRFOOUT5 in the TRFOOUT register).</li> <li>• Output level at the compare match  Selects “H”, “L”, inverted, or unchanged (bits TRFC14 to TRFC17 in the TRFCR1 register).</li> <li>• Output level inverted  Selects output level inverted or not inverted (bits TRFOOUT6 to TRFOOUT7 in the TRFOOUT register).</li> <li>• Output level at the count stops  Selects “H”, “L”, or unchanged (bits TRFC05 to TRFC06 in the TRFCR0 register).</li> <li>• Timing to set the TRF register to 0000h  Overflow or compare 1 match in the TRFM1 register (the CCLR bit in the TRFCR1 register).</li> <li>• TRFO11 pin select function  P3_4 or P3_7 is selected by the TRFOSEL bit in the PINSR4 register.</li> </ul>

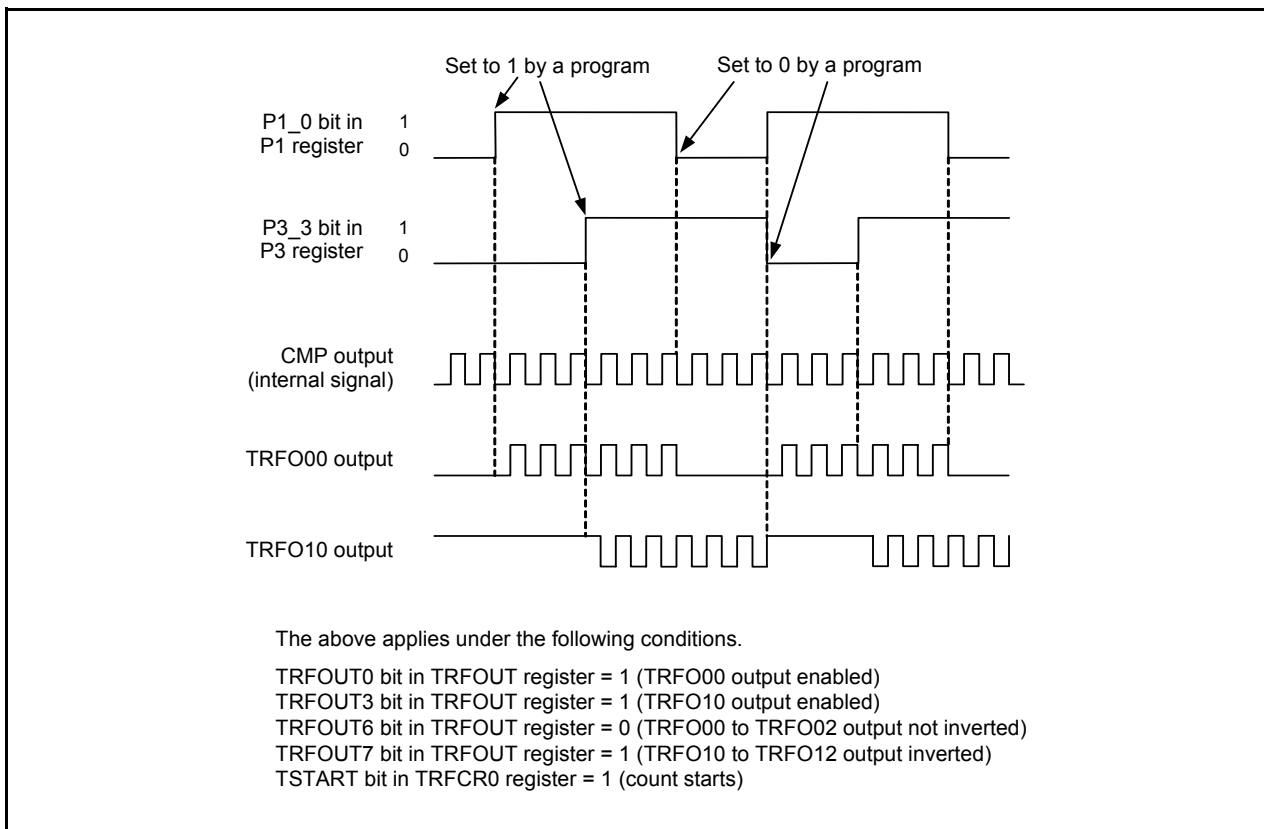
**Table 17.16 Output in Output Compare Mode (Example of TRFO00 Pin)**

TRFO00 Output		Bit Setting Value					
		TRFCR0 Register			TRFOUT Register		P1 Register
		TRFC06	TRFC05	TSTART	TRFOUT6	TRFOUT0	P1_0
Counting	CMP output	X	X	1	0	1	1
	Inverted output of CMP output	X	X	1	1	1	1
	"L" output	X	X	1	0	1	0
	"H" output	X	X	1	1	1	0
Count stops	Holds output level before count stops	X	0	0	X	1	1
	"L" output	0	1	0	X	1	1
	"H" output	1	1	0	X	1	1

X: 0 or 1



**Figure 17.54 Operating Example in Output Compare Mode**



**Figure 17.55 Operating Example in Output Compare Mode (“L” and “H” Held Output in Count Stops)**

In output compare mode, the same PWM waveform is output from all of pins TRFO00 to TRFO02 and TRFO10 to TRFO12 during count operation. Note that the output waveform can be inverted for pins TRFO00 to TRFO02 or for pins TRFO10 to TRFO12. The output can also be fixed at “L” or “H” for individual pins for a given period.

The behavior when count operation stops can be selected from the following two options: the output level before the count stops is maintained, or output is fixed at “L” or “H”.

The values in the compare *i* register can be read by reading the TRFM<sub>*i*</sub> (*i* = 0 or 1) register. Writing to the TRFM<sub>*i*</sub> register causes the values to be stored in the compare *i* register in the following timing:

- If the TSTART bit is set to 0 (count stops)
  - Values are stored simultaneously with the write to the TRFM<sub>*i*</sub> register.
- If the TSTART bit is set to 1 (count starts) and the CCLR bit in the TRFCR1 register is set to 0 (free running)
  - Values are stored when the TRF register (counter) overflows.
- If the TSTART bit is set to 1 and the CCLR bit is set to 1 (TRF register set to 0000h at compare 1 match)
  - Values are stored when the compare 1 and TRF register (counter) values match.

### 17.4.3 Notes on Timer RF

- Access registers TRF, TRFM0, and TRFM1 in 16-bit units.

Example of reading timer RF:

```
MOV.W    0290H,R0    ; Read out timer RF
```

- In input capture mode, a capture interrupt request is generated by inputting an edge selected by bits TRFC03 and TRFC04 in the TRFCR0 register even when the TSTART bit in the TRFCR0 register is set to 0 (count stops).



## 18. Serial Interface

The serial interface consists of two channels (UART0 or UART2). Each UART<sub>i</sub> ( $i = 0$  or  $2$ ) has an exclusive timer to generate the transfer clock and operates independently.

Figure 18.1 shows a UART<sub>i</sub> ( $i = 0$  or  $2$ ) Block Diagram. Figure 18.2 shows a UART<sub>i</sub> Transmit/Receive Unit.

UART<sub>i</sub> has two modes: clock synchronous serial I/O mode and clock asynchronous serial I/O mode (UART mode).

Figures 18.3 to 18.5 show the Registers Associated with UART<sub>i</sub>.

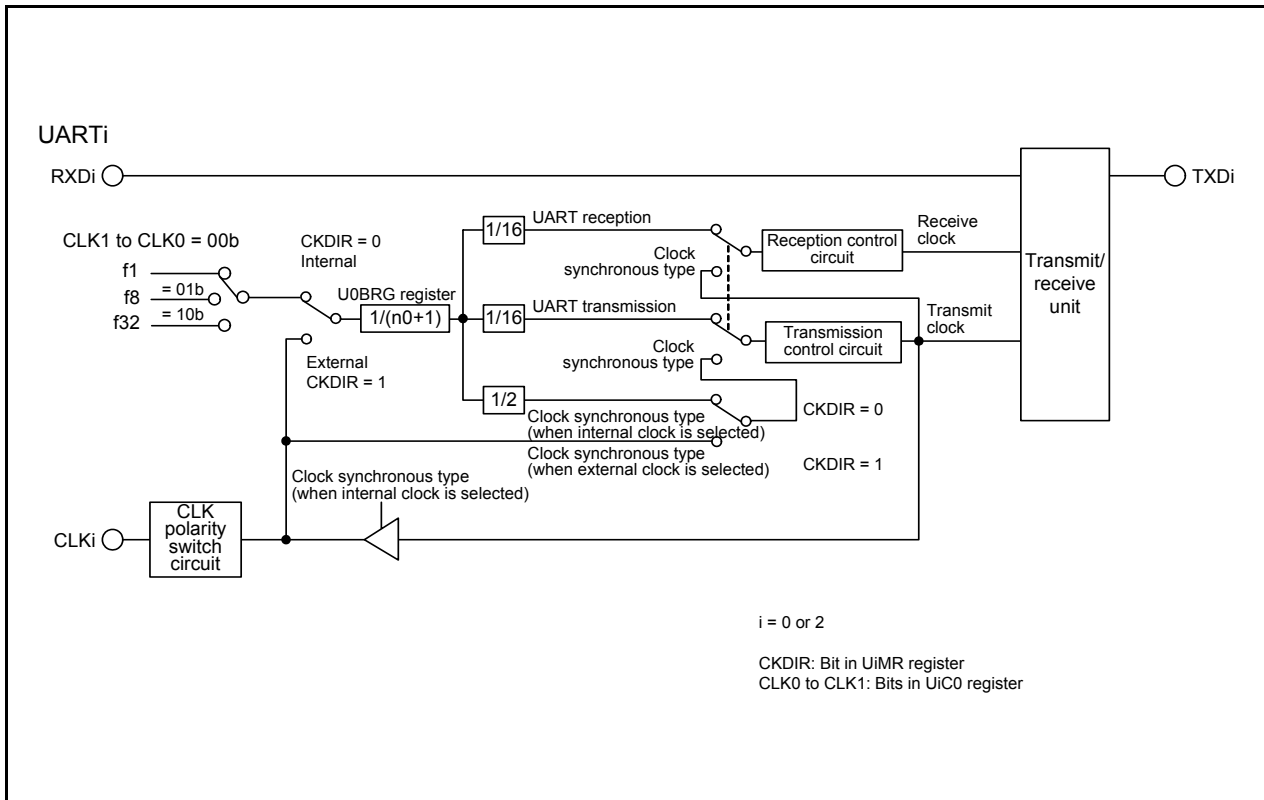


Figure 18.1 UART<sub>i</sub> ( $i = 0$  or  $2$ ) Block Diagram

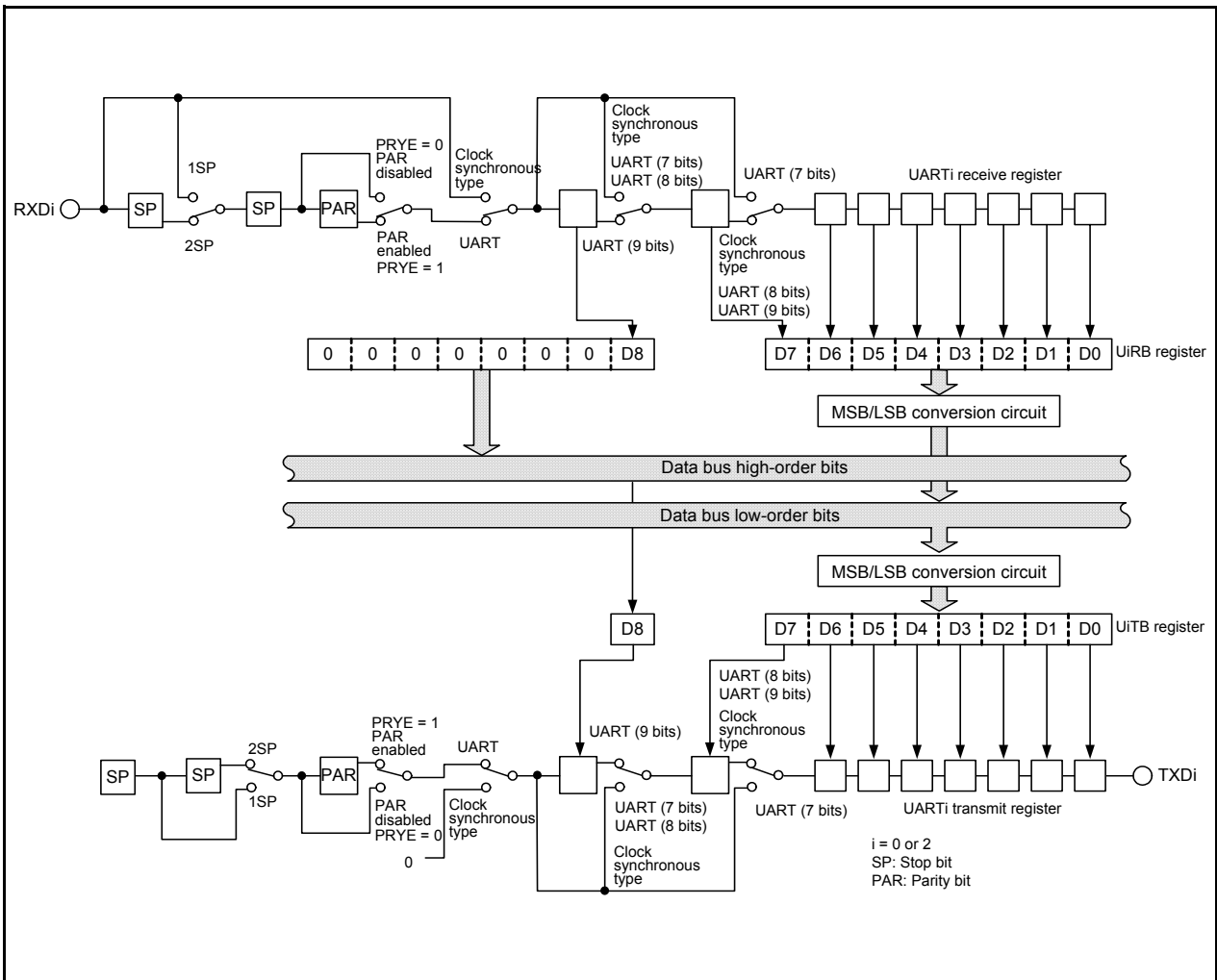


Figure 18.2 UARTi Transmit/Receive Unit

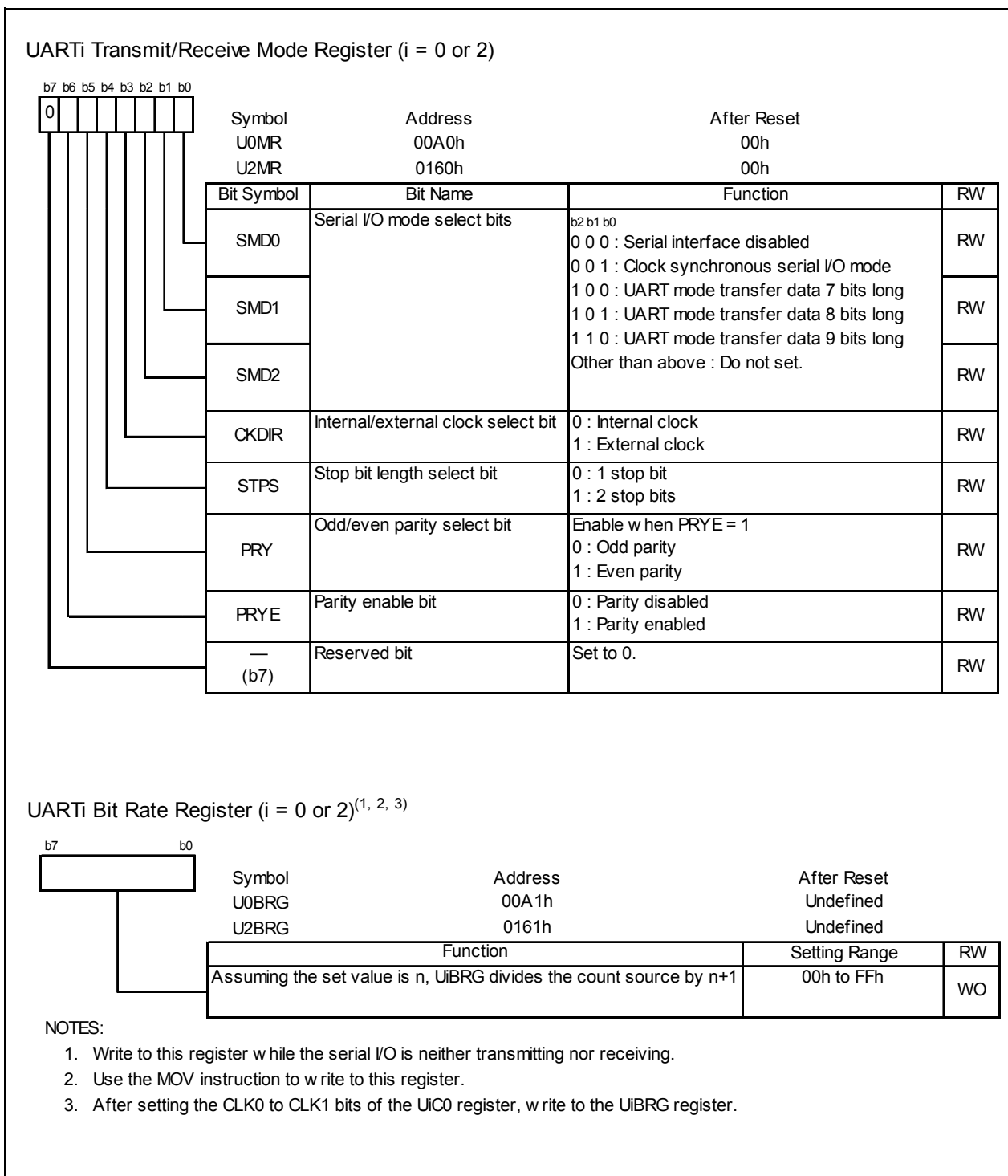


Figure 18.3 Registers U0MR, U2MR and U0BRG, U2BRG

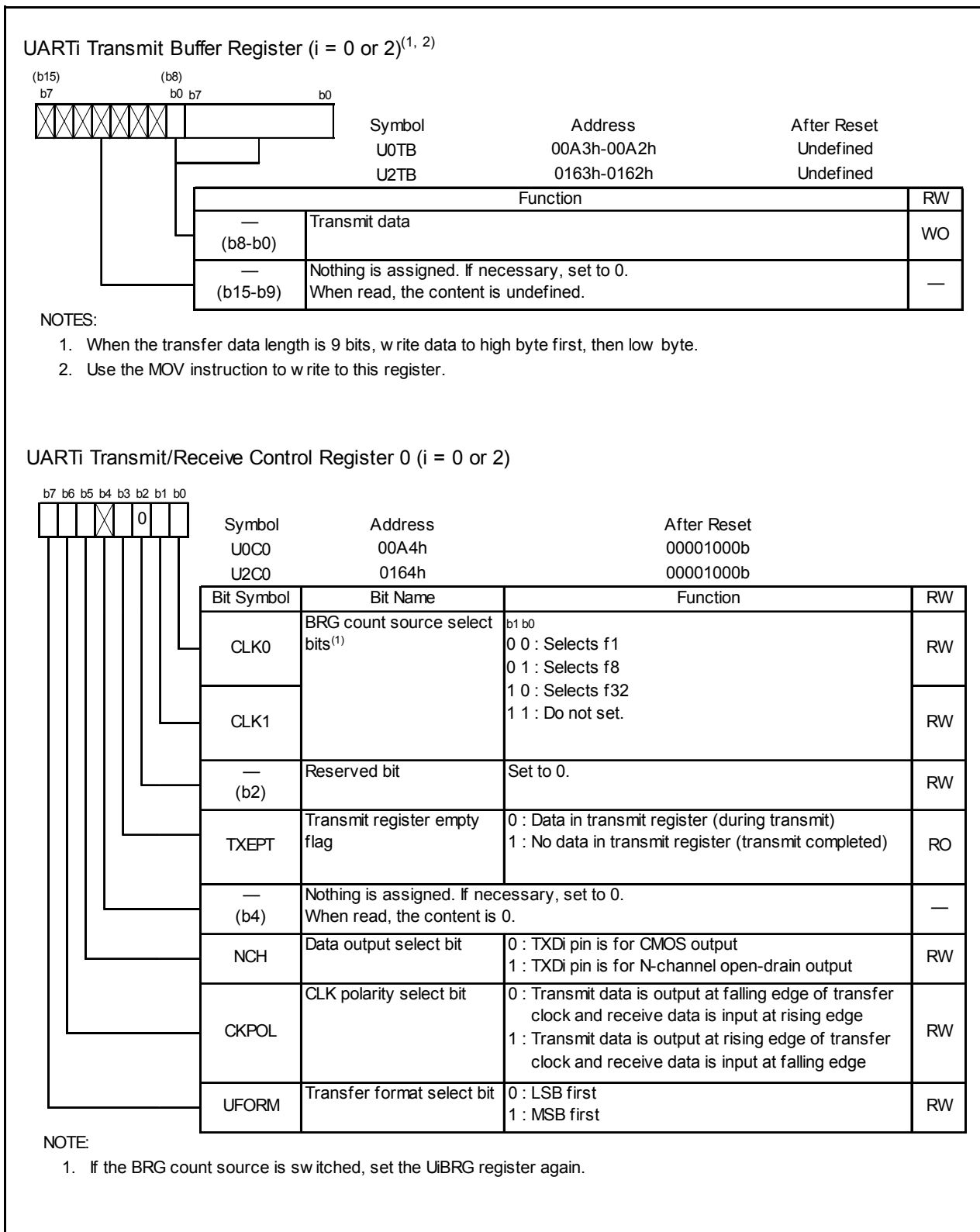


Figure 18.4 Registers U0TB, U2TB and U0C0, U2C0

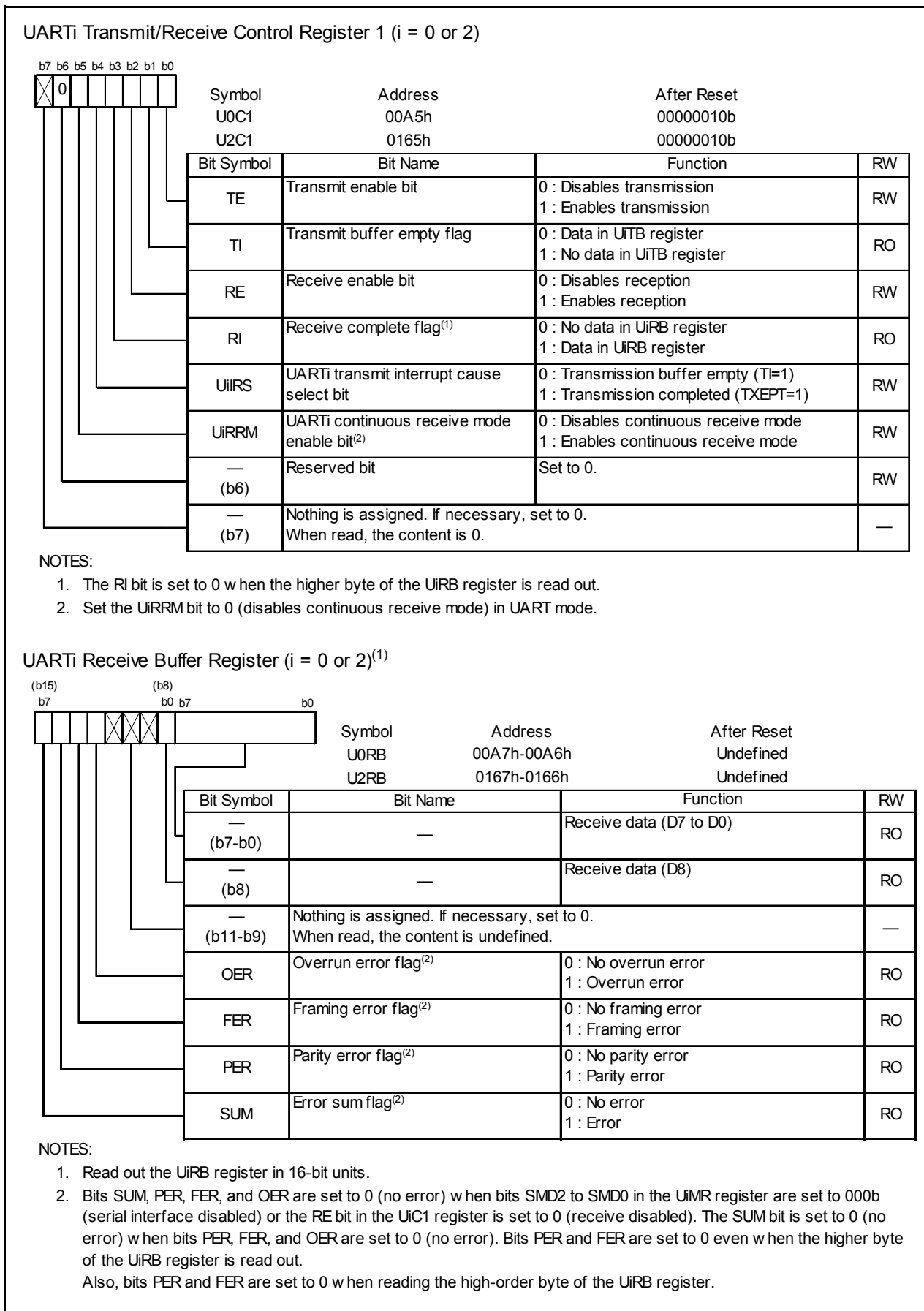


Figure 18.5 Registers U0C1, U2C1 and U0RB, U2RB

## 18.1 Clock Synchronous Serial I/O Mode

In clock synchronous serial I/O mode, data is transmitted and received using a transfer clock.

Table 18.1 lists the Clock Synchronous Serial I/O Mode Specifications. Table 18.2 lists the Registers Used and Settings in Clock Synchronous Serial I/O Mode<sup>(1)</sup>.

**Table 18.1 Clock Synchronous Serial I/O Mode Specifications**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data length: 8 bits</li> </ul>
Transfer clocks	<ul style="list-style-type: none"> <li>CKDIR bit in UiMR register is set to 0 (internal clock): <math>f_i/(2(n+1))</math>  <math>f_i = f_1, f_8, f_{32}</math> <math>n =</math> value set in UiBRG register: 00h to FFh</li> <li>The CKDIR bit is set to 1 (external clock): input from CLKi pin</li> </ul>
Transmit start conditions	<ul style="list-style-type: none"> <li>Before transmission starts, the following requirements must be met<sup>(1)</sup> <ul style="list-style-type: none"> <li>The TE bit in the UiC1 register is set to 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register is set to 0 (data in the UiTB register)</li> </ul> </li> </ul>
Receive start conditions	<ul style="list-style-type: none"> <li>Before reception starts, the following requirements must be met<sup>(1)</sup> <ul style="list-style-type: none"> <li>The RE bit in the UiC1 register is set to 1 (reception enabled)</li> <li>The TE bit in the UiC1 register is set to 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register is set to 0 (data in the UiTB register)</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>When transmitting, one of the following conditions can be selected <ul style="list-style-type: none"> <li>The UiIRS bit is set to 0 (transmit buffer empty): When transferring data from the UiTB register to UARTi transmit register (when transmission starts).</li> <li>The UiIRS bit is set to 1 (transmission completes): When completing data transmission from UARTi transmit register.</li> </ul> </li> <li>When receiving When data transfer from the UARTi receive register to the UiRB register (when reception completes).</li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Overrun error<sup>(2)</sup> This error occurs if the serial interface starts receiving the next data item before reading the UiRB register and receives the 7th bit of the next data.</li> </ul>
Select functions	<ul style="list-style-type: none"> <li>CLK polarity selection Transfer data input/output can be selected to occur synchronously with the rising or the falling edge of the transfer clock.</li> <li>LSB first, MSB first selection Whether transmitting or receiving data begins with bit 0 or begins with bit 7 can be selected.</li> <li>Continuous receive mode selection Receive is enabled immediately by reading the UiRB register.</li> </ul>

i = 0 or 2

### NOTES:

- If an external clock is selected, ensure that the external clock is "H" when the CKPOL bit in the UiC0 register is set to 0 (transmit data output at falling edge and receive data input at rising edge of transfer clock), and that the external clock is "L" when the CKPOL bit is set to 1 (transmit data output at rising edge and receive data input at falling edge of transfer clock).
- If an overrun error occurs, the receive data (b0 to b8) of the UiRB register will be undefined. The IR bit in the SiRIC register remains unchanged.

**Table 18.2 Registers Used and Settings in Clock Synchronous Serial I/O Mode<sup>(1)</sup>**

Register	Bit	Function
UiTB	0 to 7	Set data transmission
UiRB	0 to 7	Data reception can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set bit rate
UiMR	SMD2 to SMD0	Set to 001b
	CKDIR	Select the internal clock or external clock
UiC0	CLK1 to CLK0	Select the count source in the UiBRG register
	TXEPT	Transmit register empty flag
	NCH	Select TXDi pin output mode
	CKPOL	Select the transfer clock polarity
	UFORM	Select the LSB first or MSB first
UiC1	TE	Set this bit to 1 to enable transmission/reception
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	UiIRS	Select the UARTi transmit interrupt source
	UiRRM	Set this bit to 1 to use continuous receive mode

i = 0 or 2

**NOTE:**

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

Table 18.3 lists the I/O Pin Functions in Clock Synchronous Serial I/O Mode. The TXDi pin outputs “H” level between the operating mode selection of UARTi (i = 0 or 2) and transfer start. (If the NCH bit is set to 1 (N-channel open-drain output), this pin is in a high-impedance state.)

**Table 18.3 I/O Pin Functions in Clock Synchronous Serial I/O Mode**

Pin Name	Function	Selection Method
TXD0 (P1_4)	Output serial data	(Outputs dummy data when performing reception only)
RXD0 (P1_5)	Input serial data	PD1_5 bit in PD1 register = 0 (P1_5 can be used as an input port when performing transmission only)
CLK0 (P1_6)	Output transfer clock	CKDIR bit in U0MR register = 0
	Input transfer clock	CKDIR bit in U0MR register = 1 PD1_6 bit in PD1 register = 0
TXD2 (P6_3)	Output serial data	(Outputs dummy data when performing reception only)
RXD2 (P6_4)	Input serial data	PD6_4 bit in PD6 register = 0 (P6_4 can be used as an input port when performing transmission only)
CLK2 (P6_5)	Output transfer clock	CKDIR bit in U2MR register = 0
	Input transfer clock	CKDIR bit in U2MR register = 1 PD6_5 bit in PD6 register = 0

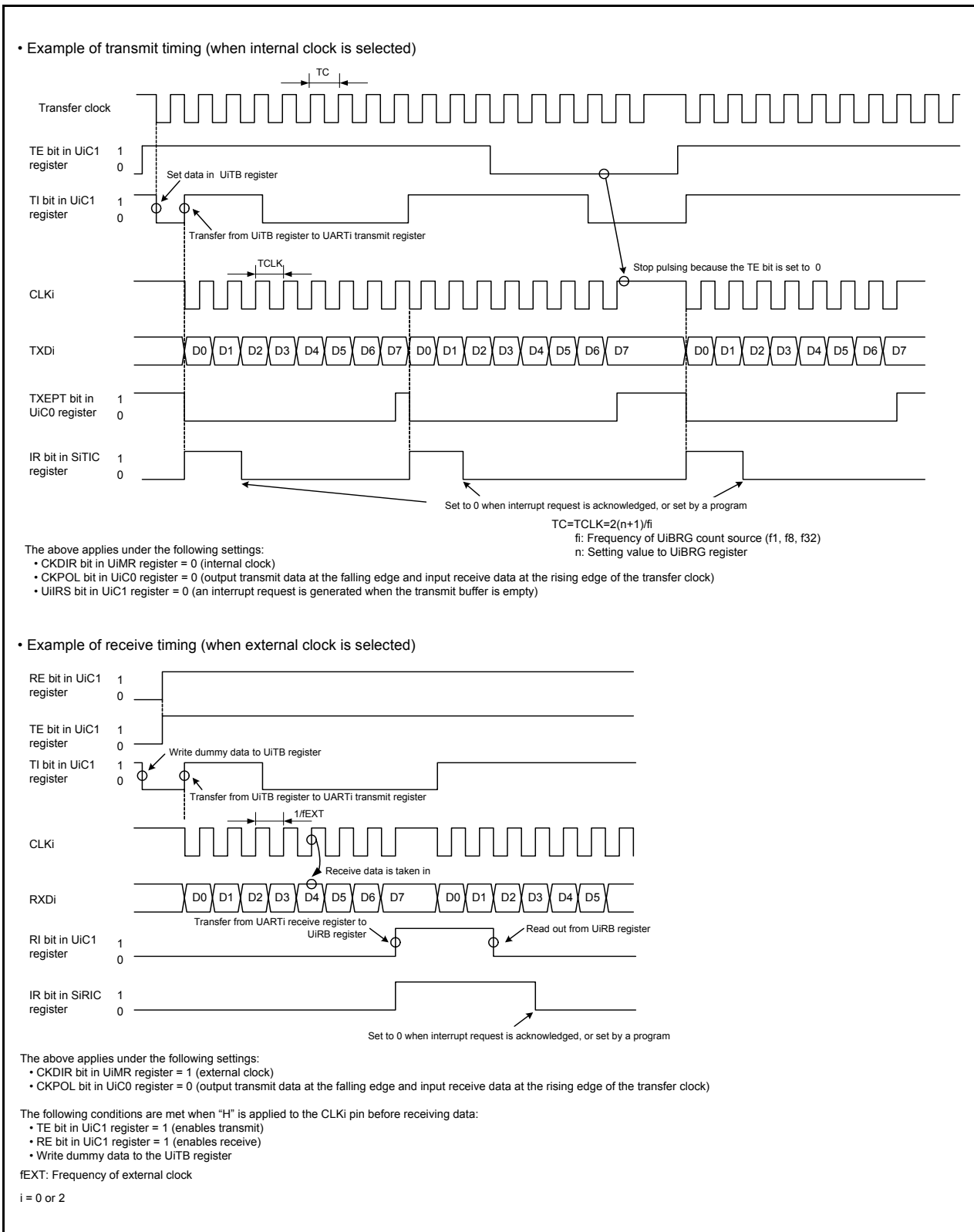


Figure 18.6 Transmit and Receive Timing Example in Clock Synchronous Serial I/O Mode



### 18.1.1 Polarity Select Function

Figure 18.7 shows the Transfer Clock Polarity. Use the CKPOL bit in the UiC0 (i = 0 or 2) register to select the transfer clock polarity.

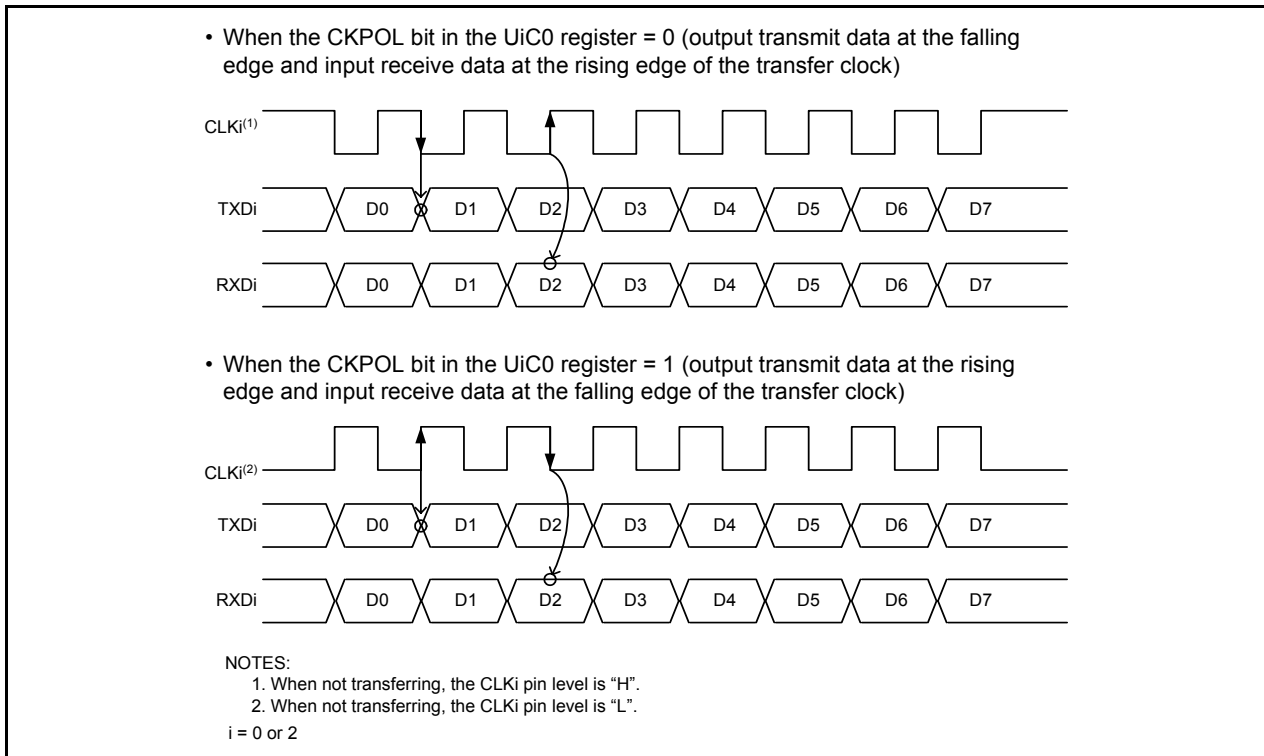


Figure 18.7 Transfer Clock Polarity

### 18.1.2 LSB First/MSB First Select Function

Figure 18.8 shows the Transfer Format. Use the UFORM bit in the UiC0 (i = 0 or 2) register to select the transfer format.

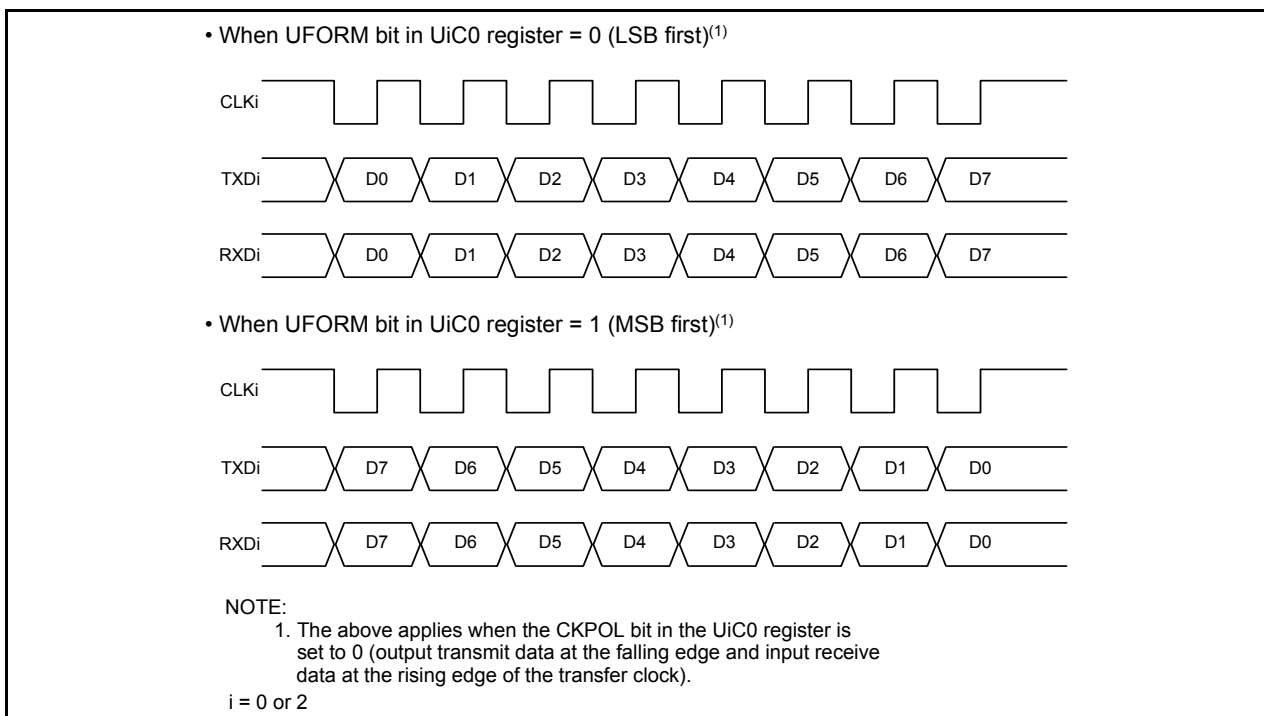


Figure 18.8 Transfer Format

### 18.1.3 Continuous Receive Mode

Continuous receive mode is selected by setting the UiRRM (i = 0 or 2) bit in the UiC1 register to 1 (enables continuous receive mode). In this mode, reading the UiRB register sets the TI bit in the UiC1 register to 0 (data in the UiTB register). When the UiRRM bit is set to 1, do not write dummy data to the UiTB register by a program.

## 18.2 Clock Asynchronous Serial I/O (UART) Mode

The UART mode allows data transmission and reception after setting the desired bit rate and transfer data format. Table 18.4 lists the UART Mode Specifications. Table 18.5 lists the Registers Used and Settings for UART Mode.

**Table 18.4 UART Mode Specifications**

Item	Specification
Transfer data formats	<ul style="list-style-type: none"> <li>• Character bit (transfer data): Selectable among 7, 8 or 9 bits</li> <li>• Start bit: 1 bit</li> <li>• Parity bit: Selectable among odd, even, or none</li> <li>• Stop bit: Selectable among 1 or 2 bits</li> </ul>
Transfer clocks	<ul style="list-style-type: none"> <li>• CKDIR bit in UiMR register is set to 0 (internal clock): <math>f_j/(16(n+1))</math>  <math>f_j = f_1, f_8, f_{32}</math> <math>n =</math> value set in UiBRG register: 00h to FFh</li> <li>• CKDIR bit is set to 1 (external clock): <math>f_{EXT}/(16(n+1))</math>  <math>f_{EXT}</math>: Input from CLKi pin, <math>n =</math> value set in UiBRG register: 00h to FFh</li> </ul>
Transmit start conditions	<ul style="list-style-type: none"> <li>• Before transmission starts, the following are required               <ul style="list-style-type: none"> <li>- TE bit in UiC1 register is set to 1 (transmission enabled)</li> <li>- TI bit in UiC1 register is set to 0 (data in UiTB register)</li> </ul> </li> </ul>
Receive start conditions	<ul style="list-style-type: none"> <li>• Before reception starts, the following are required               <ul style="list-style-type: none"> <li>- RE bit in UiC1 register is set to 1 (reception enabled)</li> <li>- Start bit detected</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When transmitting, one of the following conditions can be selected               <ul style="list-style-type: none"> <li>- UiIRS bit is set to 0 (transmit buffer empty):                    When transferring data from the UiTB register to UARTi transmit register (when transmission starts).</li> <li>- UiIRS bit is set to 1 (transfer ends):                    When serial interface completes transmitting data from the UARTi transmit register</li> </ul> </li> <li>• When receiving                When transferring data from the UARTi receive register to UiRB register (when reception ends).</li> </ul>
Error detection	<ul style="list-style-type: none"> <li>• Overrun error<sup>(1)</sup>                This error occurs if the serial interface starts receiving the next data item before reading the UiRB register and receive the bit preceding the final stop bit of the next data item.</li> <li>• Framing error                This error occurs when the set number of stop bits is not detected.</li> <li>• Parity error                This error occurs when parity is enabled, and the number of 1's in parity and character bits do not match the number of 1's set.</li> <li>• Error sum flag                This flag is set is set to 1 when an overrun, framing, or parity error is generated.</li> </ul>

i = 0 or 2

NOTE:

1. If an overrun error occurs, the receive data (b0 to b8) of the UiRB register will be undefined. The IR bit in the SiRIC register remains unchanged.

**Table 18.5 Registers Used and Settings for UART Mode**

Register	Bit	Function
UiTB	0 to 8	Set transmit data <sup>(1)</sup>
UiRB	0 to 8	Receive data can be read <sup>(1, 2)</sup>
	OER, FER, PER, SUM	Error flag
UiBRG	0 to 7	Set a bit rate
UiMR	SMD2 to SMD0	Set to 100b when transfer data is 7 bits long Set to 101b when transfer data is 8 bits long Set to 110b when transfer data is 9 bits long
	CKDIR	Select the internal clock or external clock
	STPS	Select the stop bit
	PRY, PRYE	Select whether parity is included and whether odd or even
UiC0	CLK0, CLK1	Select the count source for the UiBRG register
	TXEPT	Transmit register empty flag
	NCH	Select TXDi pin output mode
	CKPOL	Set to 0
	UFORM	LSB first or MSB first can be selected when transfer data is 8 bits long. Set to 0 when transfer data is 7 or 9 bits long.
UiC1	TE	Set to 1 to enable transmit
	TI	Transmit buffer empty flag
	RE	Set to 1 to enable receive
	RI	Receive complete flag
	UiIRS	Select the source of UARTi transmit interrupt
	UiRRM	Set to 0

i = 0 or 2

## NOTES:

- The bits used for transmit/receive data are as follows: Bits 0 to 6 when transfer data is 7 bits long; bits 0 to 7 when transfer data is 8 bits long; bits 0 to 8 when transfer data is 9 bits long.
- The following bits are undefined: Bits 7 and 8 when transfer data is 7 bits long; bit 8 when transfer data is 8 bits long.

Table 18.6 lists the I/O Pin Functions in UART Mode. After the UARTi (i = 0 or 2) operating mode is selected, the TXDi pin outputs “H” level. (If the NCH bit is set to 1 (N-channel open-drain output), this pin is in a high-impedance state) until transfer starts.)

**Table 18.6 I/O Pin Functions in UART Mode**

Pin name	Function	Selection Method
TXD0 (P1_4)	Output serial data	(Cannot be used as a port when performing reception only)
RXD0 (P1_5)	Input serial data	PD1_5 bit in PD1 register = 0 (P1_5 can be used as an input port when performing transmission only)
CLK0 (P1_6)	Programmable I/O Port	CKDIR bit in U0MR register = 0
	Input transfer clock	CKDIR bit in U0MR register = 1 PD1_6 bit in PD1 register = 0
TXD2 (P6_3)	Output serial data	(Cannot be used as a port when performing reception only)
RXD2 (P6_4)	Input serial data	PD6_4 bit in PD6 register = 0 (P6_4 can be used as an input port when performing transmission only)
CLK2 (P6_5)	Programmable I/O Port	CKDIR bit in U2MR register = 0
	Input transfer clock	CKDIR bit in U2MR register = 1 PD6_5 bit in PD6 register = 0

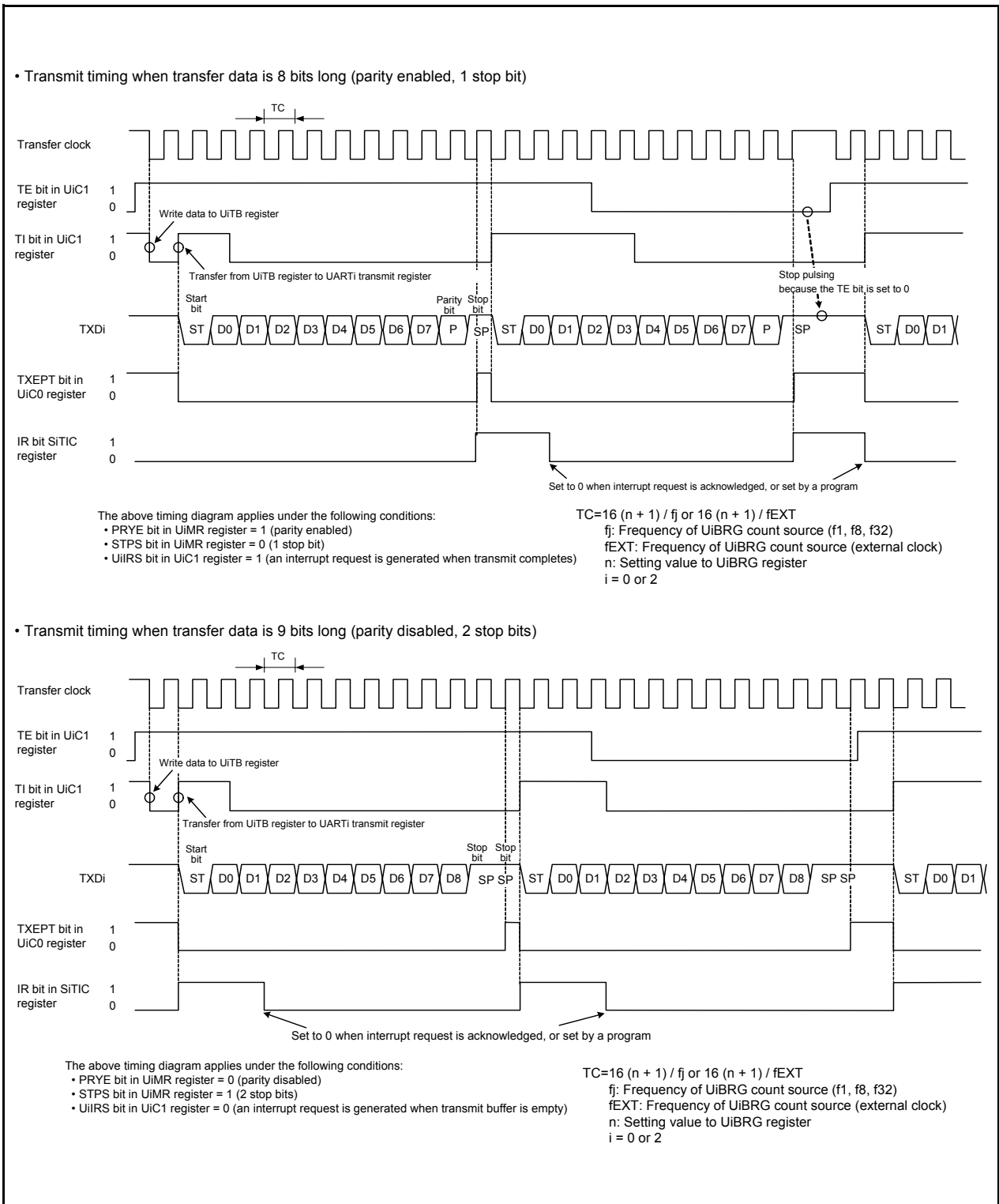


Figure 18.9 Transmit Timing in UART Mode

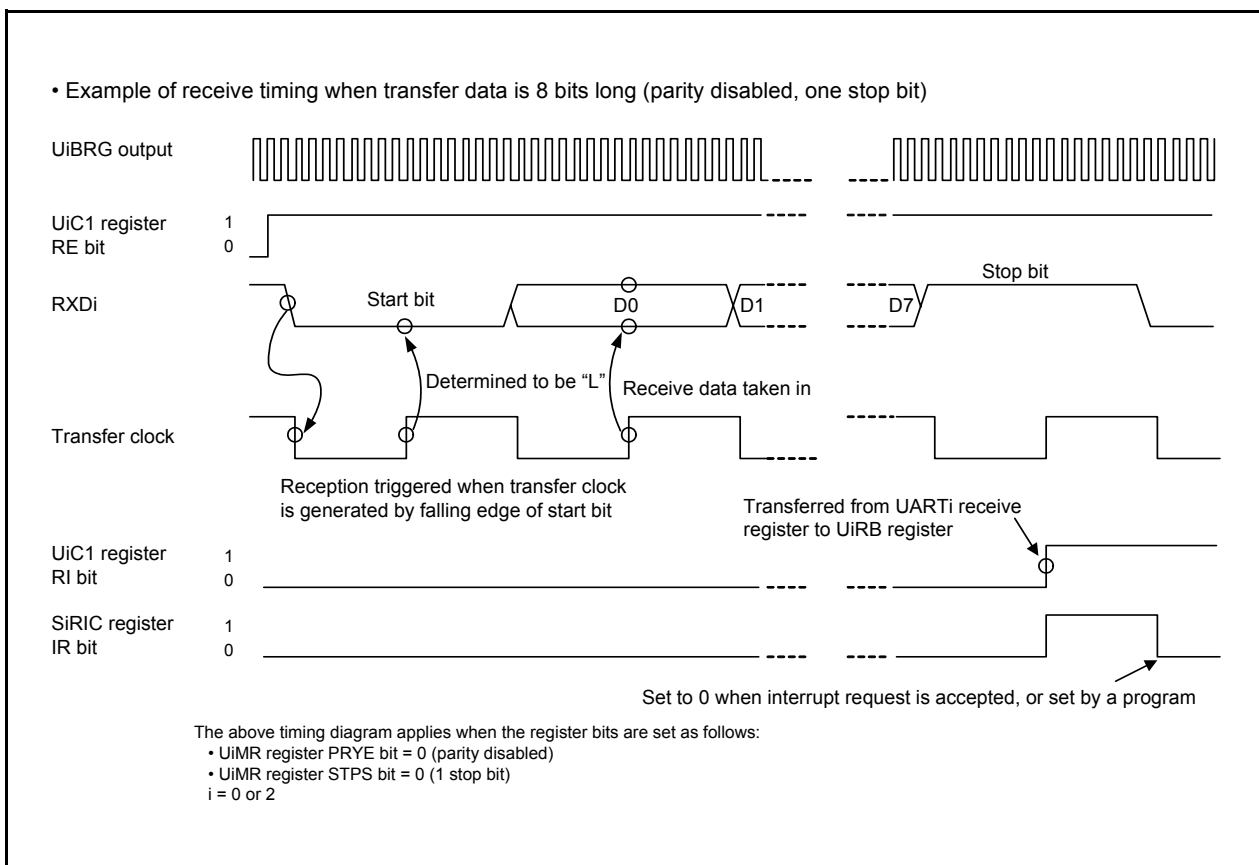


Figure 18.10 Receive Timing Example in UART Mode

### 18.2.1 Bit Rate

In UART mode, the bit rate is the frequency divided by the UiBRG (i = 0 or 2) register.

<p>UART mode</p> <ul style="list-style-type: none"> <li>• Internal clock selected</li> </ul> $\text{UiBRG register setting value} = \frac{f_j}{\text{Bit Rate} \times 16} - 1$ <p style="text-align: center;">Fj: Count source frequency of the UiBRG register (f1, f8, or f32)</p> <ul style="list-style-type: none"> <li>• External clock selected</li> </ul> $\text{UiBRG register setting value} = \frac{f_{\text{EXT}}}{\text{Bit Rate} \times 16} - 1$ <p style="text-align: center;">fEXT: Count source frequency of the UiBRG register (external clock)</p> <p style="text-align: center;">i = 0 or 2</p>
---

**Figure 18.11 Calculation Formula of UiBRG (i = 0 or 2) Register Setting Value**

**Table 18.7 Bit Rate Setting Example in UART Mode (Internal Clock Selected)**

Bit Rate (bps)	BRG Count Source	System Clock = 8 MHz		
		UiBRG Setting Value	Actual Time (bps)	Error (%)
1200	f8	51 (33h)	1201.92	0.16
2400	f8	25 (19h)	2403.85	0.16
4800	f8	12 (0Ch)	4807.69	0.16
9600	f1	51 (33h)	9615.38	0.16
14400	f1	34 (22h)	14285.71	-0.79
19200	f1	25 (19h)	19230.77	0.16
28800	f1	16 (10h)	29411.76	2.12
31250	f1	15 (0Fh)	31250.00	0.00
38400	f1	12 (0Ch)	38461.54	0.16
51200	f1	9 (09h)	50000.00	-2.34

### 18.3 Notes on Serial Interface

- When reading data from the UiRB (i = 0 or 2) register either in the clock synchronous serial I/O mode or in the clock asynchronous serial I/O mode. Ensure the data is read in 16-bit units. When the high-order byte of the UiRB register is read, bits PER and FER in the UiRB register and the RI bit in the UiC1 register are set to 0. To check receive errors, read the UiRB register and then use the read data.

Example (when reading receive buffer register):

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

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

Example (when reading transmit buffer register):

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



## 19. Hardware LIN

The hardware LIN performs LIN communication in cooperation with timer RA and UART0.

### 19.1 Features

The hardware LIN has the features listed below.

Figure 19.1 shows a Block Diagram of Hardware LIN.

Master mode

- Generates Synch Break
- Detects bus collision

Slave mode

- Detects Synch Break
- Measures Synch Field
- Controls Synch Break and Synch Field signal inputs to UART0
- Detects bus collision

NOTE:

1. The WakeUp function is detected by INT1.

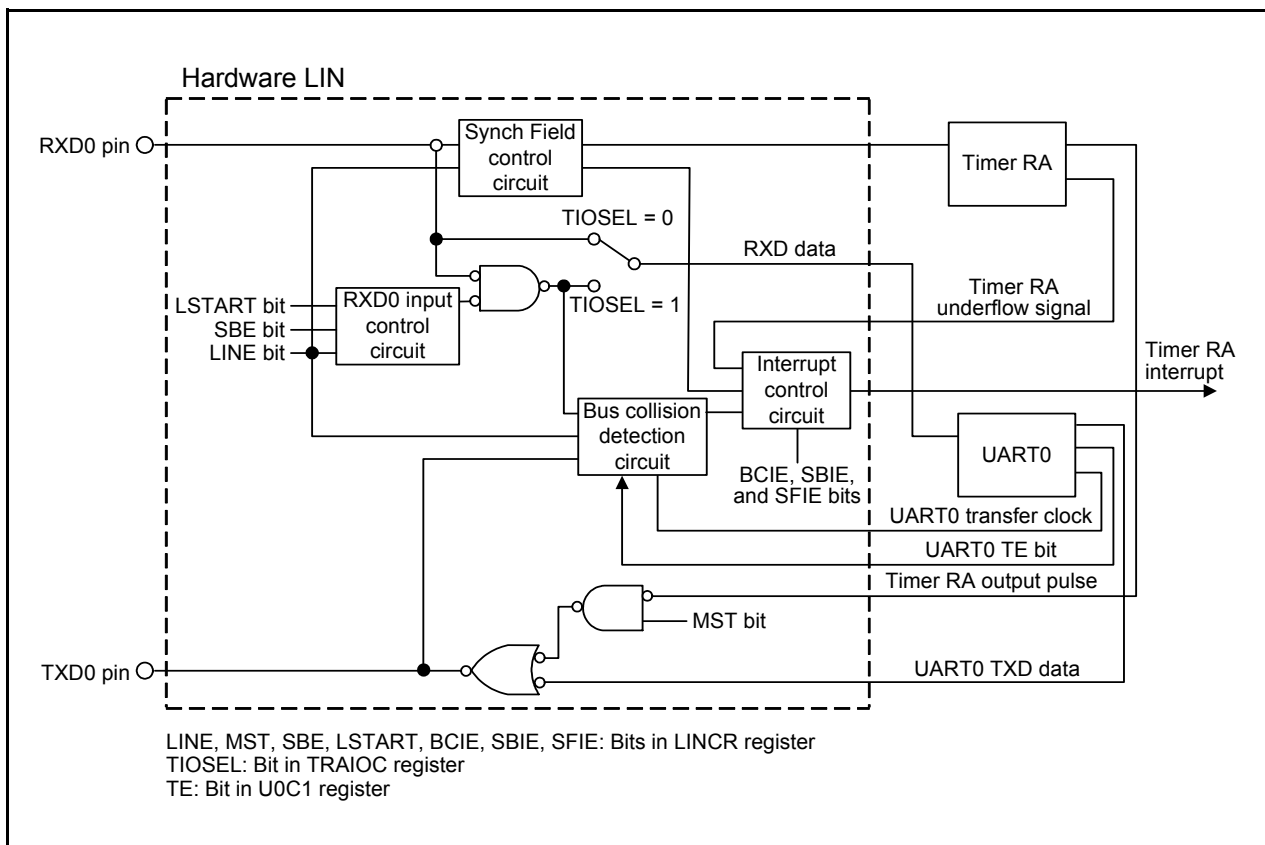


Figure 19.1 Block Diagram of Hardware LIN

## 19.2 Input/Output Pins

The pin configuration of the hardware LIN is listed in Table 19.1.

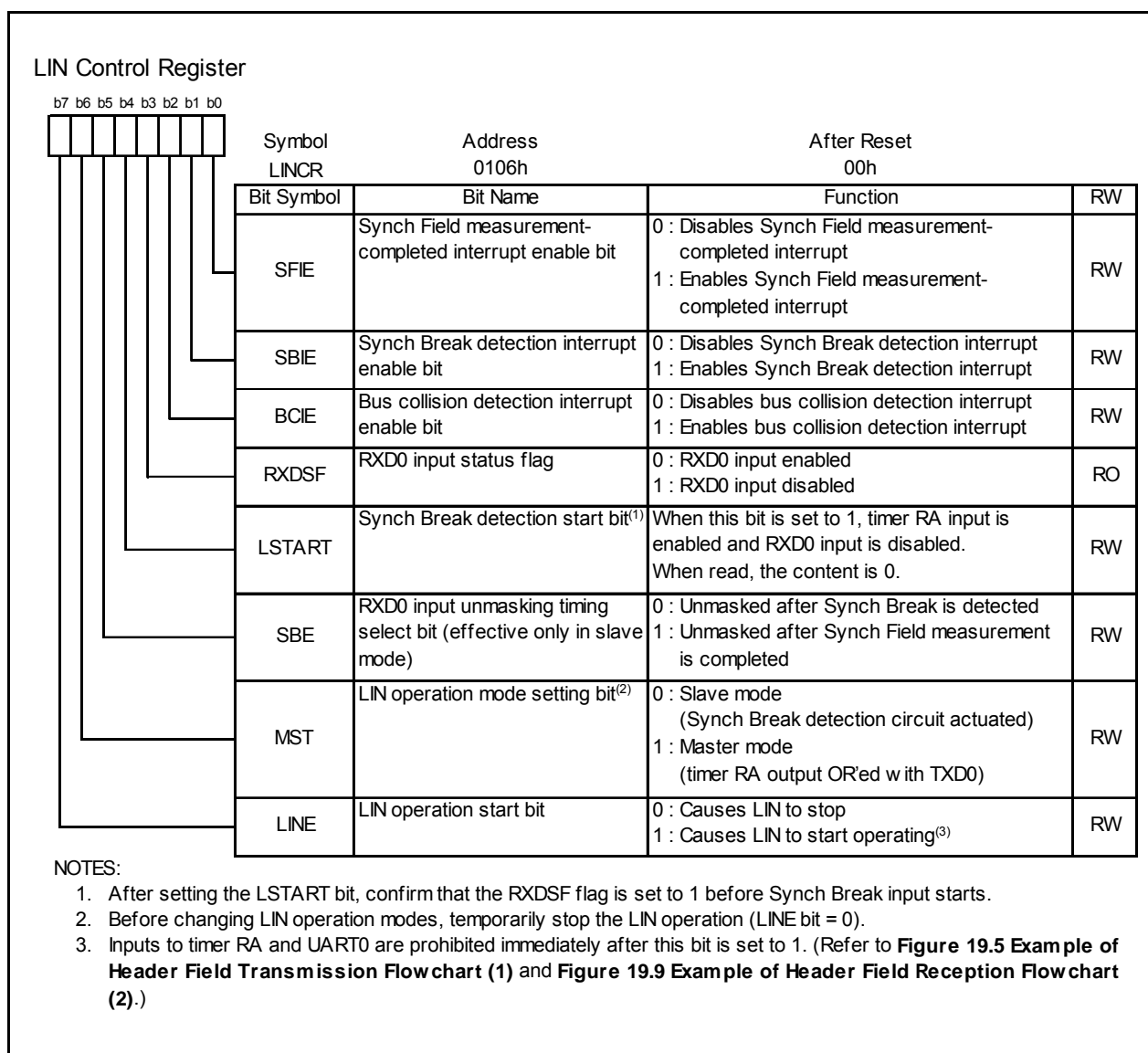
**Table 19.1 Pin Configuration**

Name	Abbreviation	Input/Output	Function
Receive data input	RXD0	Input	Receive data input pin of the hardware LIN
Transmit data output	TXD0	Output	Transmit data output pin of the hardware LIN

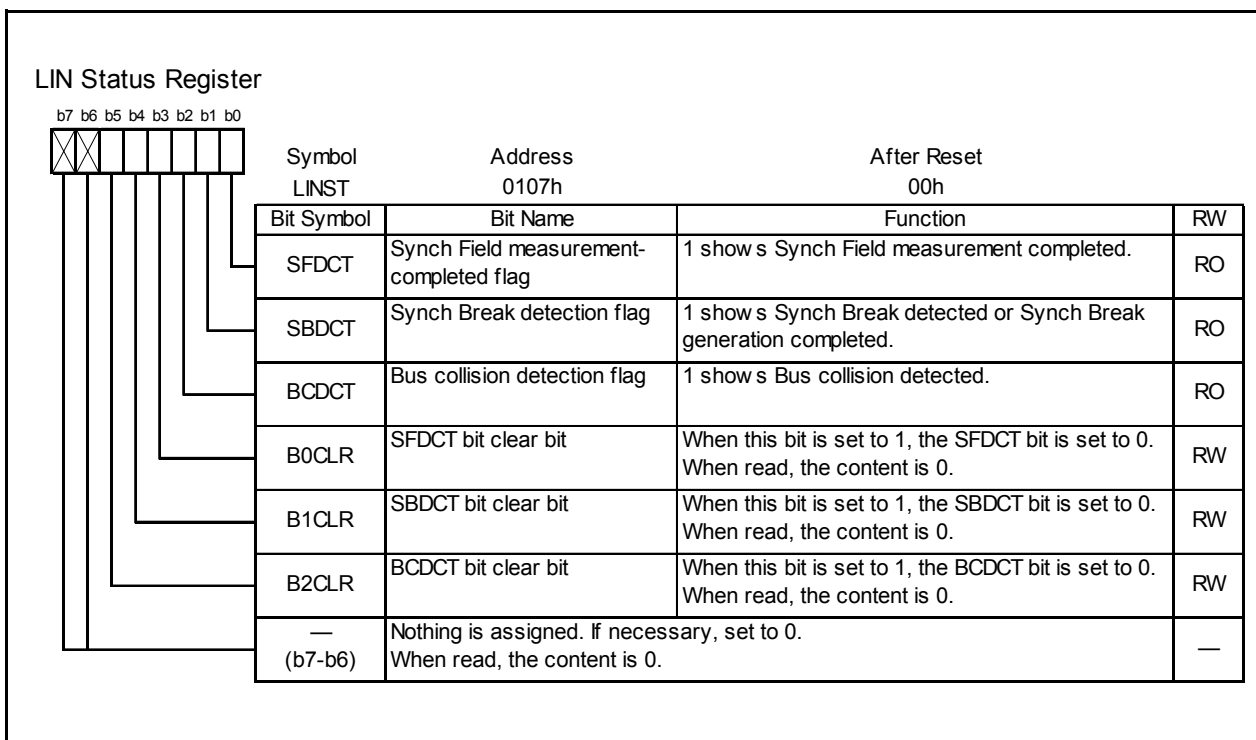
### 19.3 Register Configuration

The hardware LIN contains the registers listed below.  
These registers are detailed in Figures 19.2 and 19.3.

- LIN Control Register (LINCR)
- LIN Status Register (LINST)



**Figure 19.2 LINCR Register**



**Figure 19.3 LINST Register**

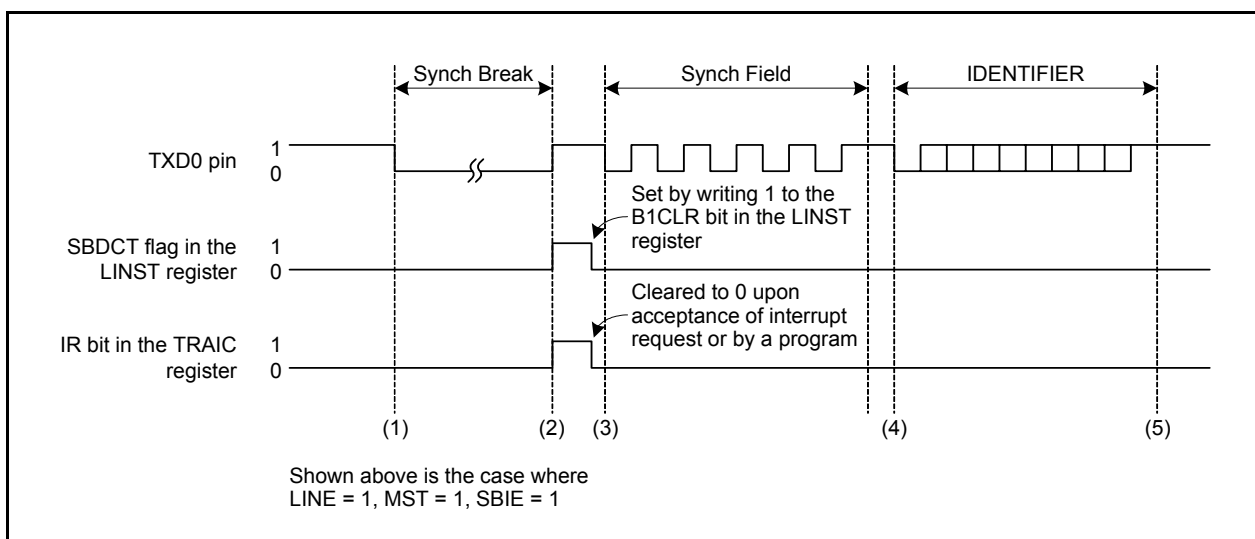
## 19.4 Functional Description

### 19.4.1 Master Mode

Figure 19.4 shows typical operation of the hardware LIN when transmitting a header field in master mode. Figures 19.5 and 19.6 show an Example of Header Field Transmission Flowchart.

When transmitting a header field, the hardware LIN operates as described below.

- (1) When the TSTART bit in the TRACR register for timer RA is set by writing 1 in software, the hardware LIN outputs “L” level from the TXD0 pin for the period that is set in registers TRAPRE and TRA for timer RA.
- (2) When timer RA underflows upon reaching the terminal count, the hardware LIN reverses the output of the TXD0 pin and sets the SBDCT flag in the LINST register to 1. Furthermore, if the SBIE bit in the LINCR register is set to 1, it generates a timer RA interrupt.
- (3) The hardware LIN transmits 55h via UART0.
- (4) The hardware LIN transmits an ID field via UART0 after it finishes sending 55h.
- (5) The hardware LIN performs communication for a response field after it finishes sending the ID field.



**Figure 19.4** Typical Operation when Sending a Header Field

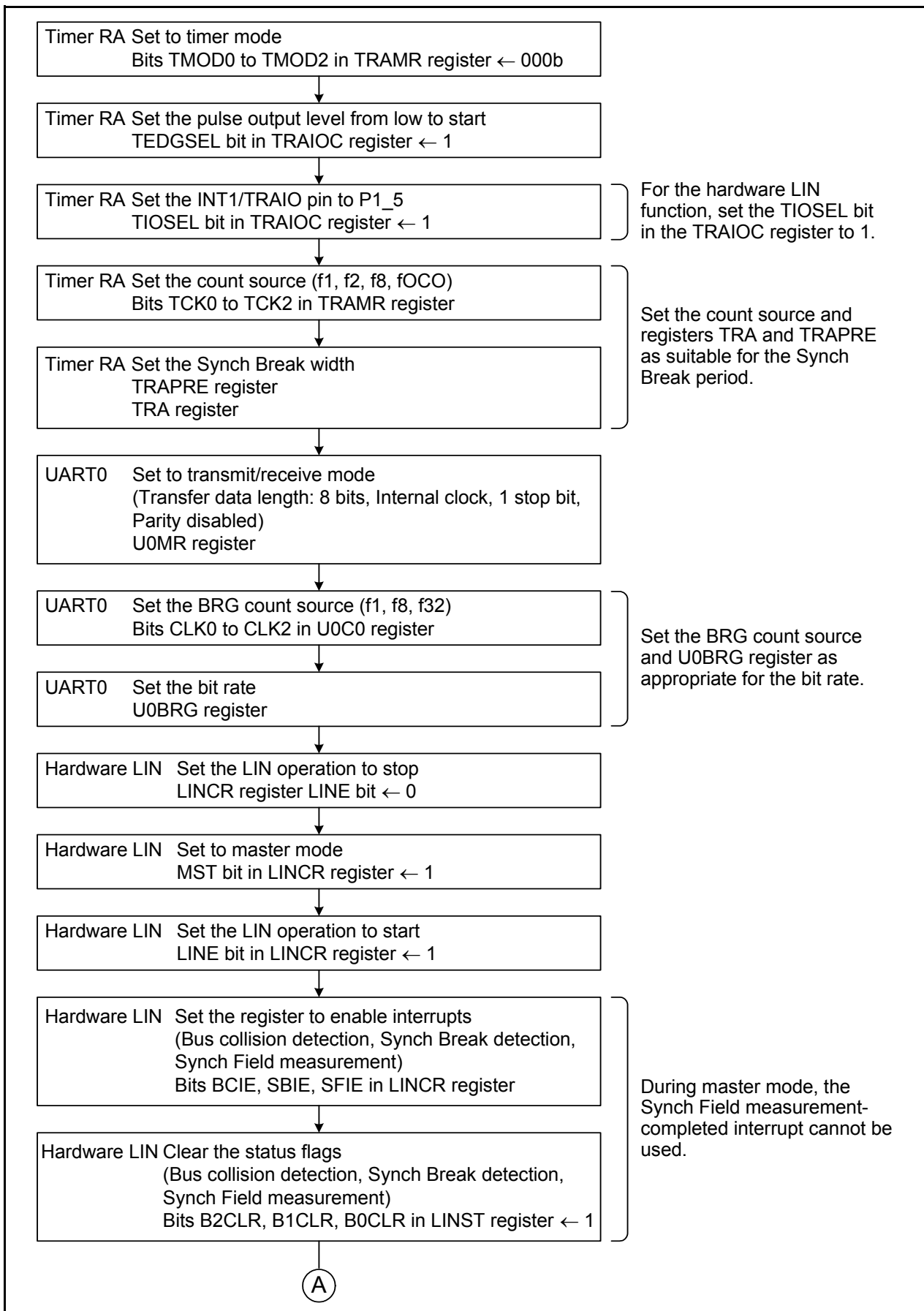


Figure 19.5 Example of Header Field Transmission Flowchart (1)

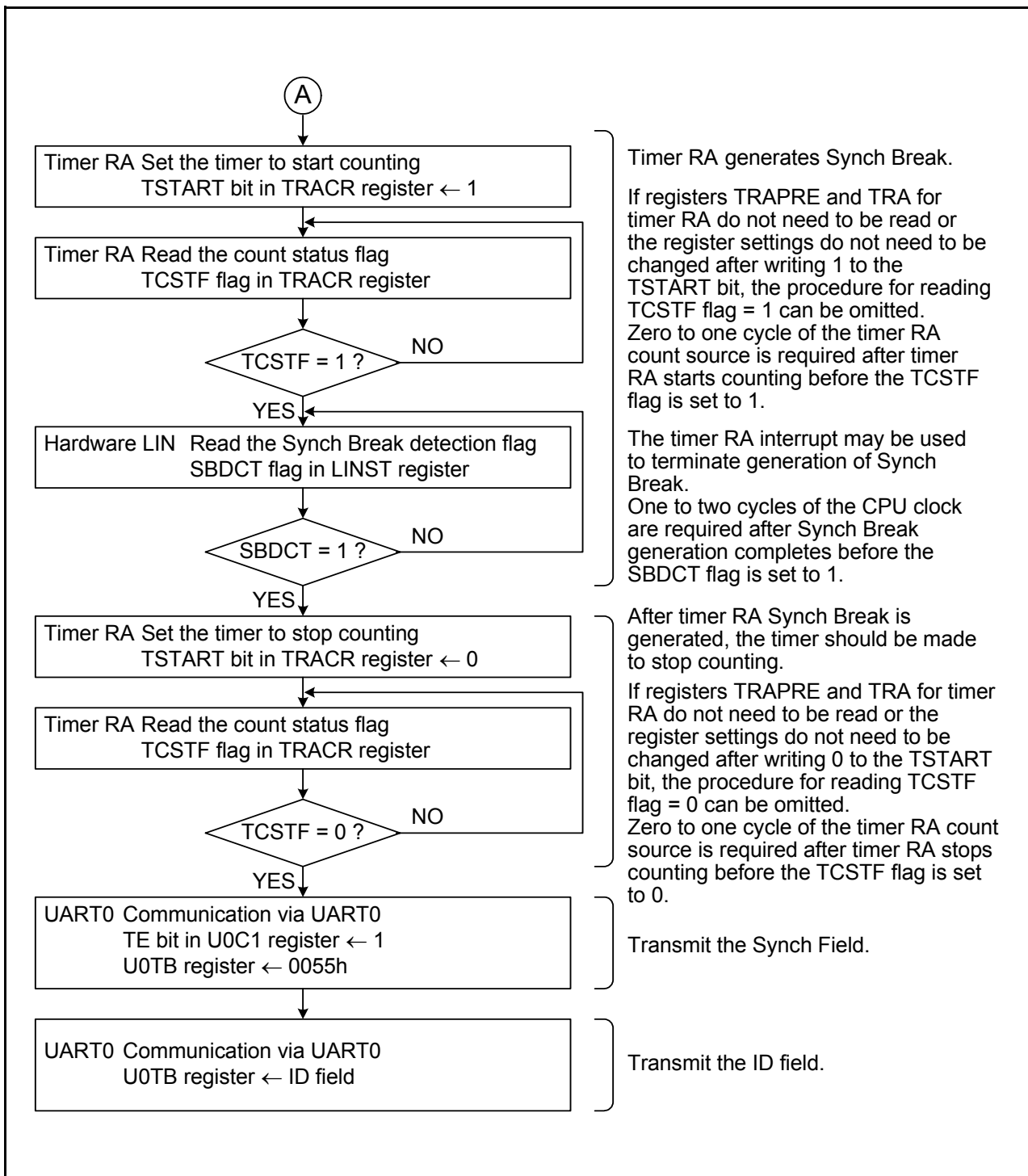


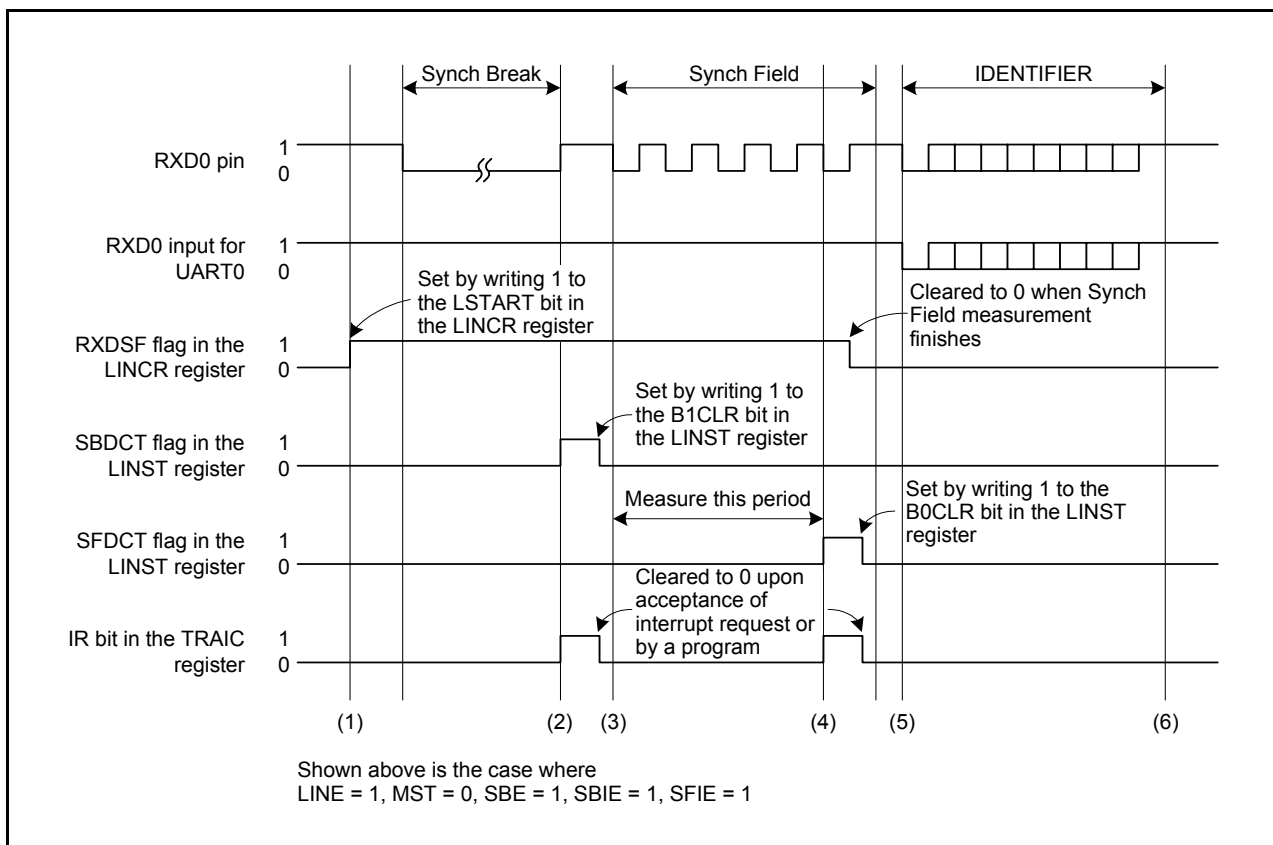
Figure 19.6 Example of Header Field Transmission Flowchart (2)

### 19.4.2 Slave Mode

Figure 19.7 shows typical operation of the hardware LIN when receiving a header field in slave mode. Figure 19.8 through Figure 19.10 show an Example of Header Field Reception Flowchart.

When receiving a header field, the hardware LIN operates as described below.

- (1) Synch Break detection is enabled by writing 1 to the LSTART bit in the LINCR register of the hardware LIN.
- (2) When “L” level is input for a duration equal to or greater than the period set in timer RA, the hardware LIN detects it as Synch Break. At this time, the SBDCT flag in the LINST register is set to 1. Furthermore, if the SBIE bit in the LINCR register is set to 1, the hardware LIN generates a timer RA interrupt. Then it goes to Synch Field measurement.
- (3) The hardware LIN receives a Synch Field (55h). At this time, it measures the period of the start bit and bits 0 to 6 by using timer RA. In this case, it is possible to select whether to input the Synch Field signal to RXD0 of UART0 by setting the SBE bit in the LINCR register accordingly.
- (4) The hardware LIN sets the SFDCT flag in the LINST register to 1 when it finishes measuring the Synch Field. Furthermore, if the SFIE bit in the LINCR register is set to 1, it generates a timer RA interrupt.
- (5) After it finishes measuring the Synch Field, calculate a transfer rate from the count value of timer RA and set to UART0 and registers TRAPRE and TRA of timer RA again. Then it receives an ID field via UART0.
- (6) The hardware LIN performs communication for a response field after it finishes receiving the ID field.



**Figure 19.7 Typical Operation when Receiving a Header Field**



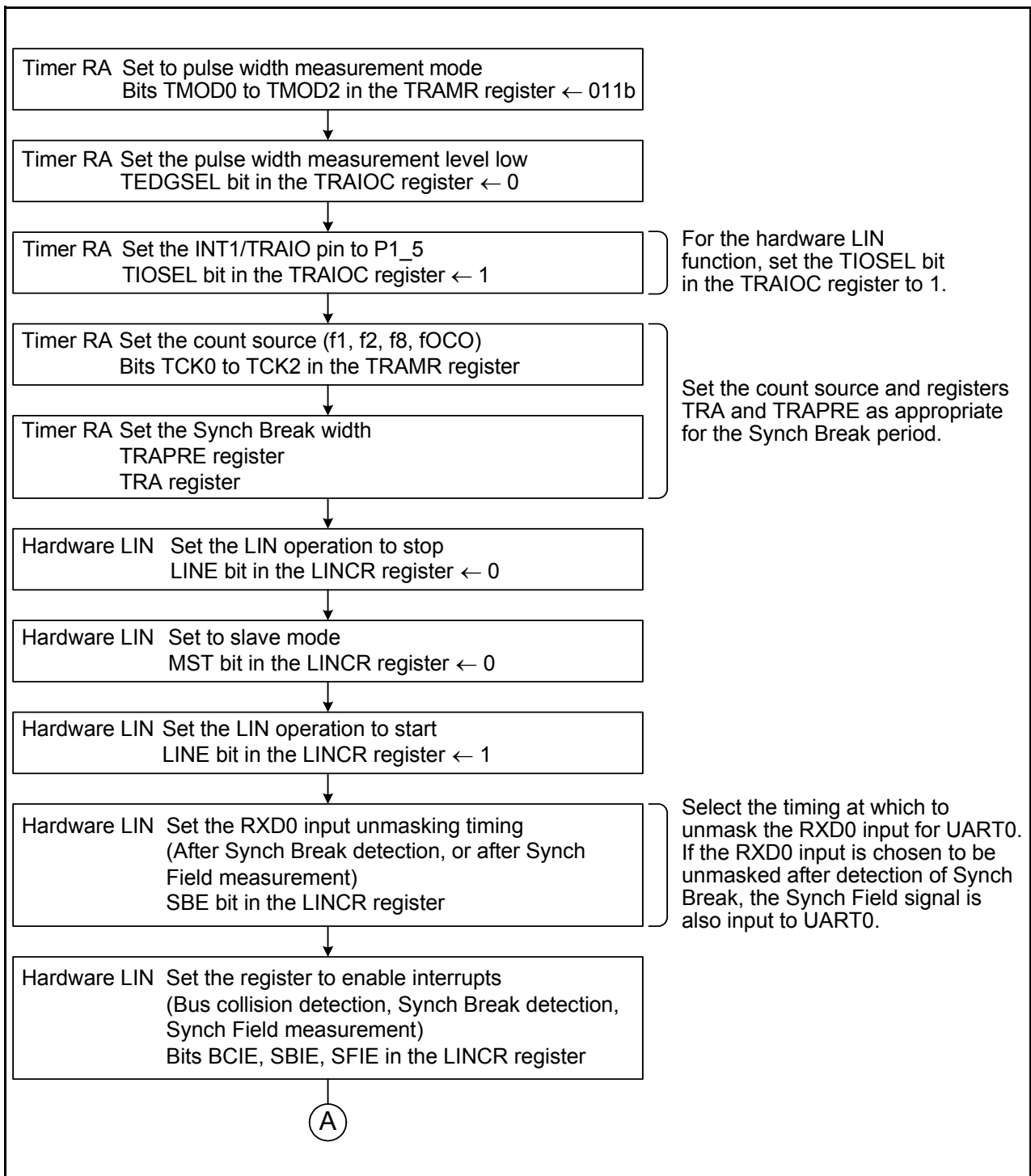


Figure 19.8 Example of Header Field Reception Flowchart (1)

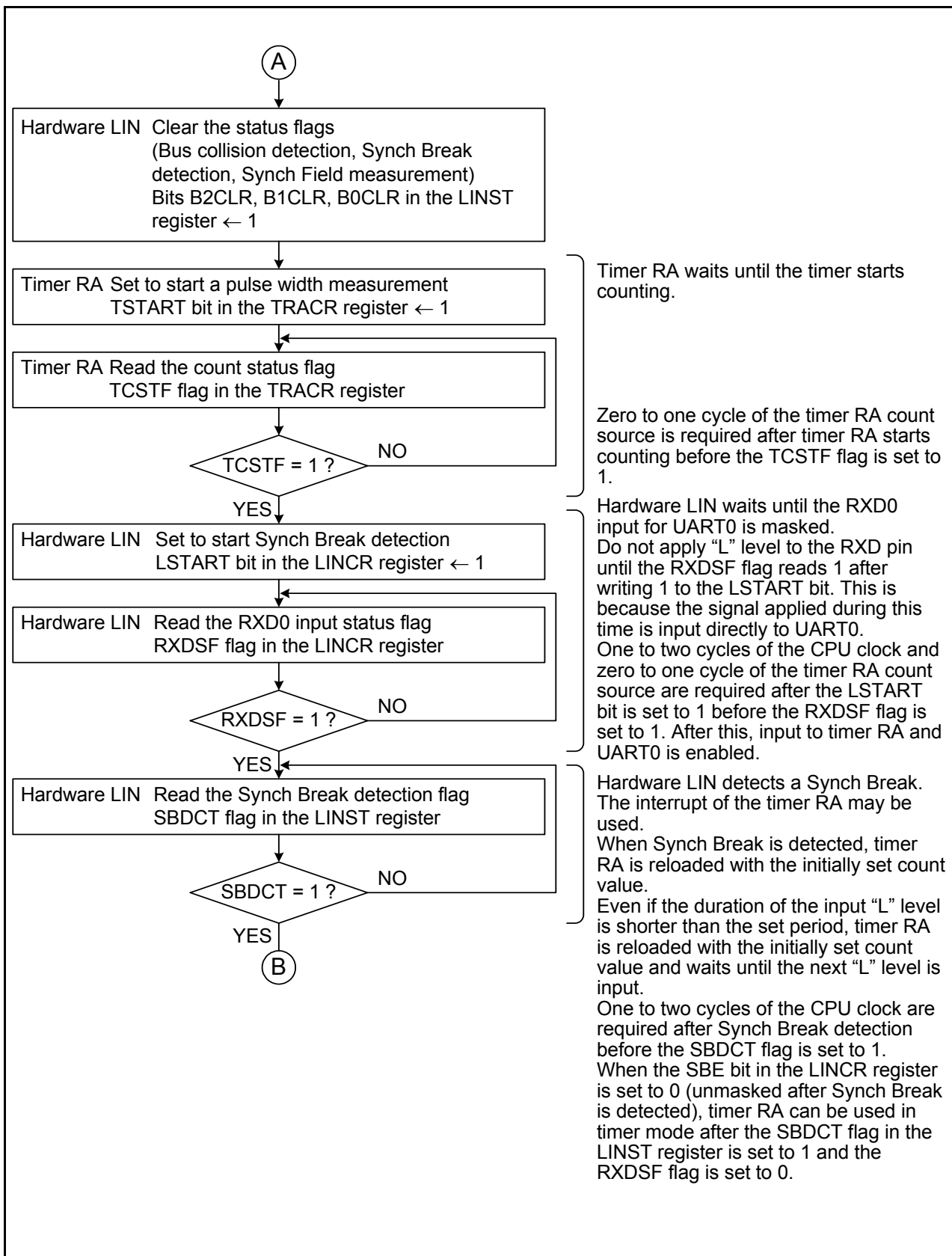


Figure 19.9 Example of Header Field Reception Flowchart (2)

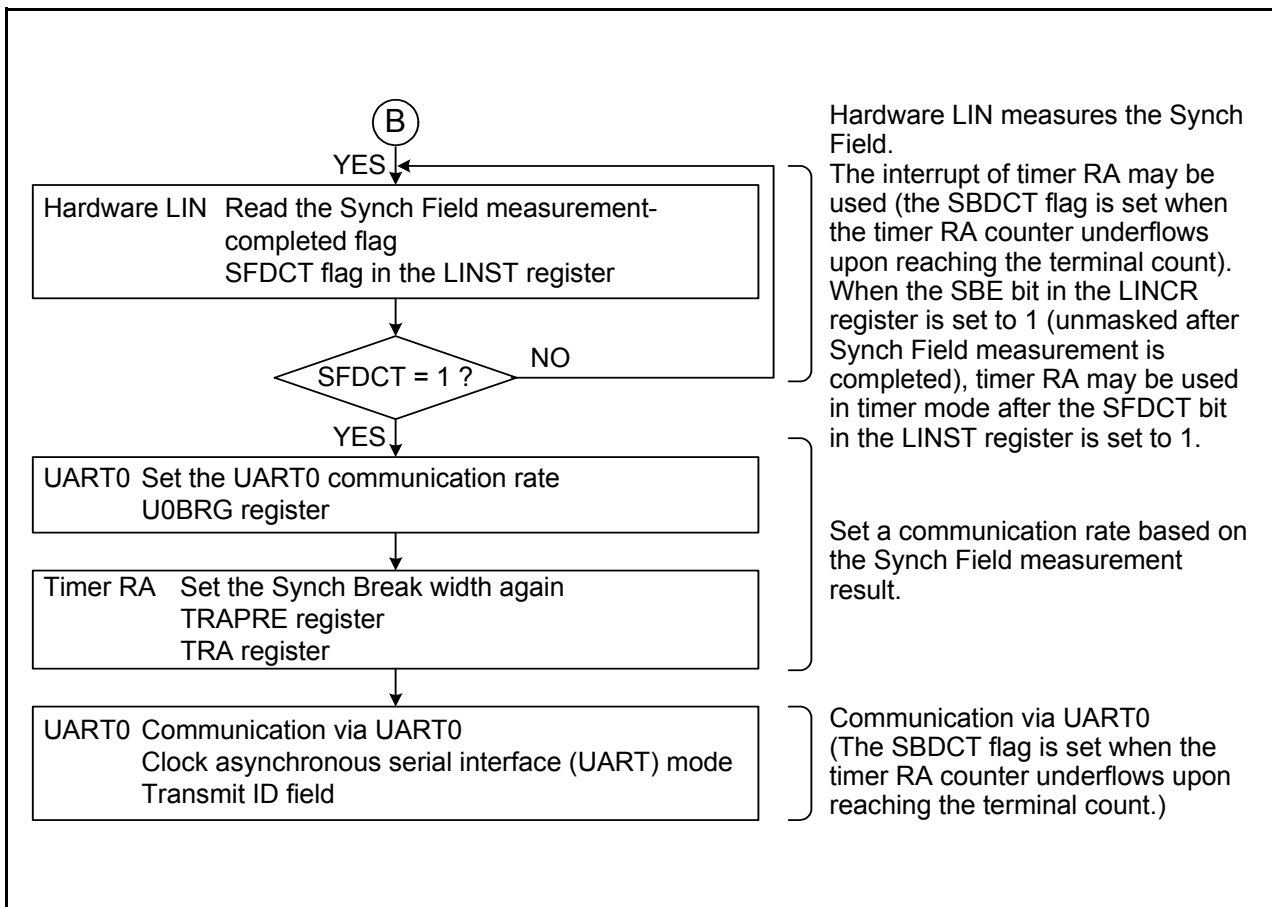


Figure 19.10 Example of Header Field Reception Flowchart (3)

### 19.4.3 Bus Collision Detection Function

The bus collision detection function can be used when UART0 is enabled for transmission (TE bit in the U0C1 register = 1).

Figure 19.11 shows the Typical Operation when a Bus Collision is Detected.

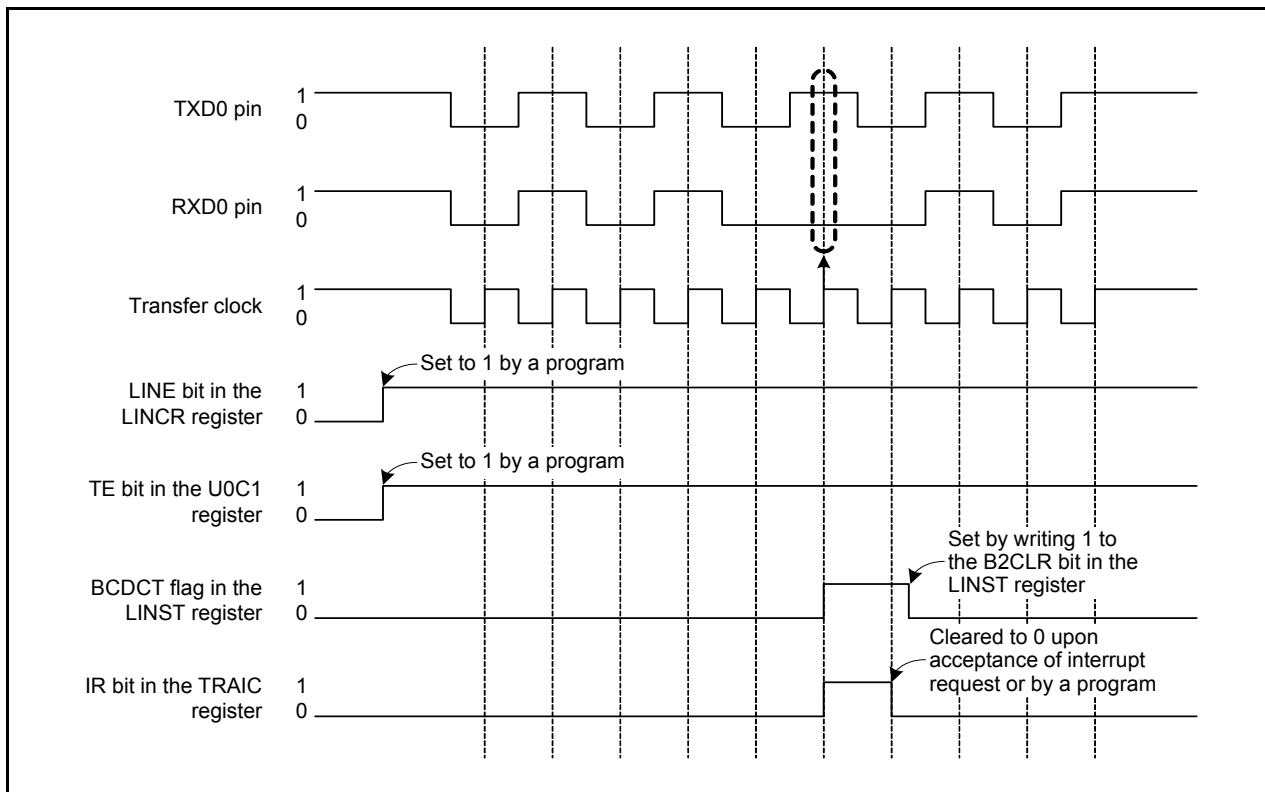


Figure 19.11 Typical Operation when a Bus Collision is Detected

### 19.4.4 Hardware LIN End Processing

Figure 19.12 shows an Example of Hardware LIN Communication Completion Flowchart.

Use the following timing for hardware LIN end processing:

- If the hardware bus collision detection function is used  
Perform hardware LIN end processing after checksum transmission completes.
- If the bus collision detection function is not used  
Perform hardware LIN end processing after header field transmission and reception complete.

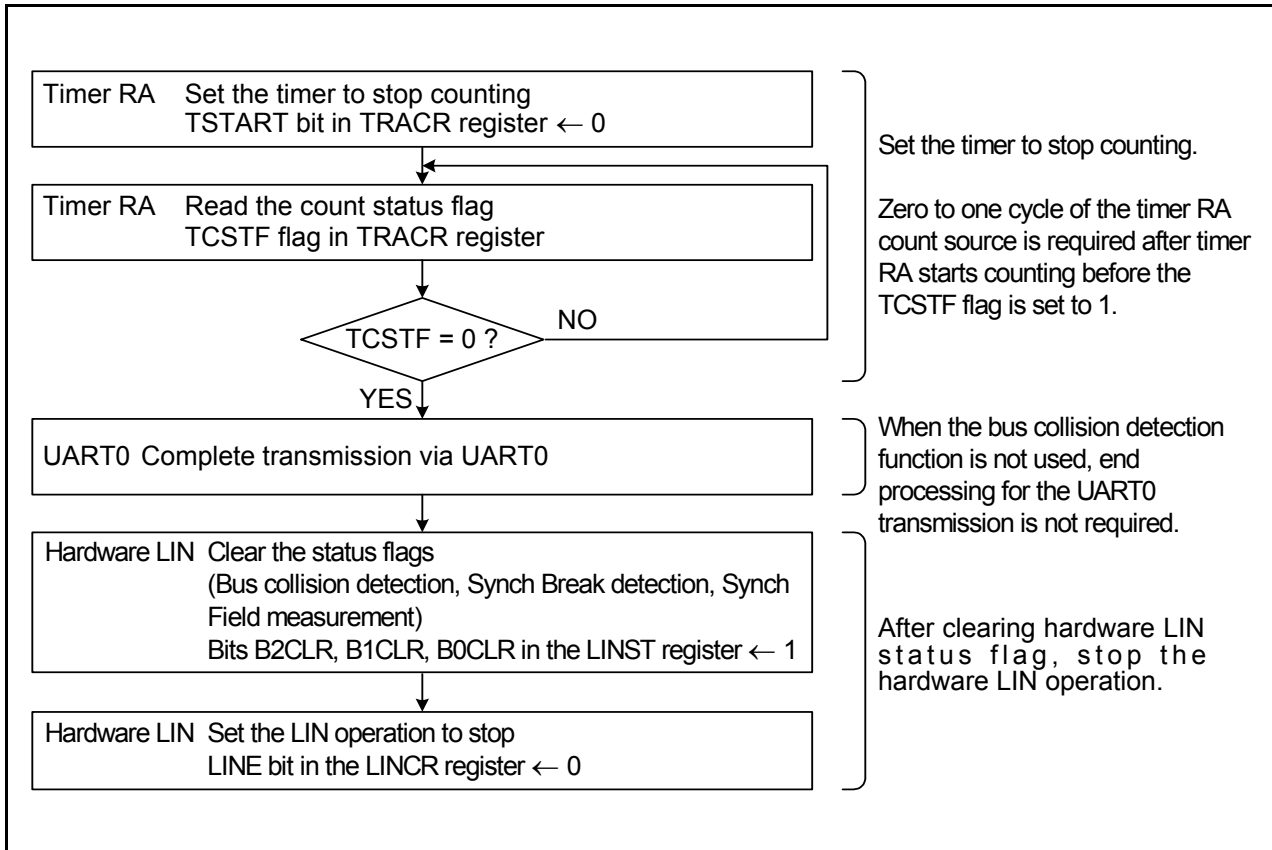


Figure 19.12 Example of Hardware LIN Communication Completion Flowchart

## 19.5 Interrupt Requests

There are four interrupt requests that are generated by the hardware LIN: Synch Break detection, Synch Break generation completed, Synch Field measurement completed, and bus collision detection. These interrupts are shared with timer RA.

Table 19.2 lists the Interrupt Requests of Hardware LIN.

**Table 19.2 Interrupt Requests of Hardware LIN**

Interrupt Request	Status Flag	Cause of Interrupt
Synch Break detection	SBDCT	Generated when timer RA has underflowed after measuring the "L" level duration of RXD0 input, or when a "L" level is input for a duration longer than the Synch Break period during communication.
Synch Break generation completed		Generated when "L" level output to TXD0 for the duration set by timer RA completes.
Synch Field measurement completed	SFDCT	Generated when measurement for 6 bits of the Synch Field by timer RA is completed.
Bus collision detection	BCDCT	Generated when the RXD0 input and TXD0 output values differed at data latch timing while UART0 is enabled for transmission.

## 19.6 Notes on Hardware LIN

For the time-out processing of the header and response fields, use another timer to measure the duration of time with a Synch Break detection interrupt as the starting point.

## 20. Flash Memory

### 20.1 Overview

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

Table 20.1 lists the Flash Memory Performance (refer to **Table 1.1 Specifications for R8C/2G Group** for items not listed in **Table 20.1**).

**Table 20.1 Flash Memory Performance**

Item		Specification
Flash memory operating mode		3 modes (CPU rewrite, standard serial I/O, and parallel I/O)
Division of erase block		Refer to <b>Figure 20.1</b>
Programming method		Byte unit
Erase method		Block erase
Programming and erasure control method		Program and erase control by software command
Protection method		Program ROM protection by FMR0 register
Number of commands		5 commands
Programming and erasure endurance <sup>(1)</sup>	Blocks 0 and 1 (program ROM)	100 times
Programming and erasure voltage		VCC = 2.7 to 5.5 V
ID code check function		Standard serial I/O mode supported
ROM code protect		Parallel I/O mode supported

NOTE:

1. Definition of programming and erasure endurance.  
The programming and erasure endurance is defined on a per-block basis.

**Table 20.2 Flash Memory Rewrite Modes**

Flash Memory Rewrite Mode	CPU Rewrite Mode	Standard Serial I/O Mode	Parallel I/O Mode
Function	User ROM area is rewritten by executing software commands from the CPU.	User ROM area is rewritten by a dedicated serial programmer.	User ROM area is rewritten by a dedicated parallel programmer.
Areas which can be rewritten	User ROM area	User ROM area	User ROM area
Rewrite Program	User program	Standard boot program	–



## 20.2 Memory Map

The flash memory contains a user ROM area and a boot ROM area (reserved area).

Figure 20.1 shows the Flash Memory Block Diagram for R8C/2G Group.

The user ROM area contains program ROM.

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

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

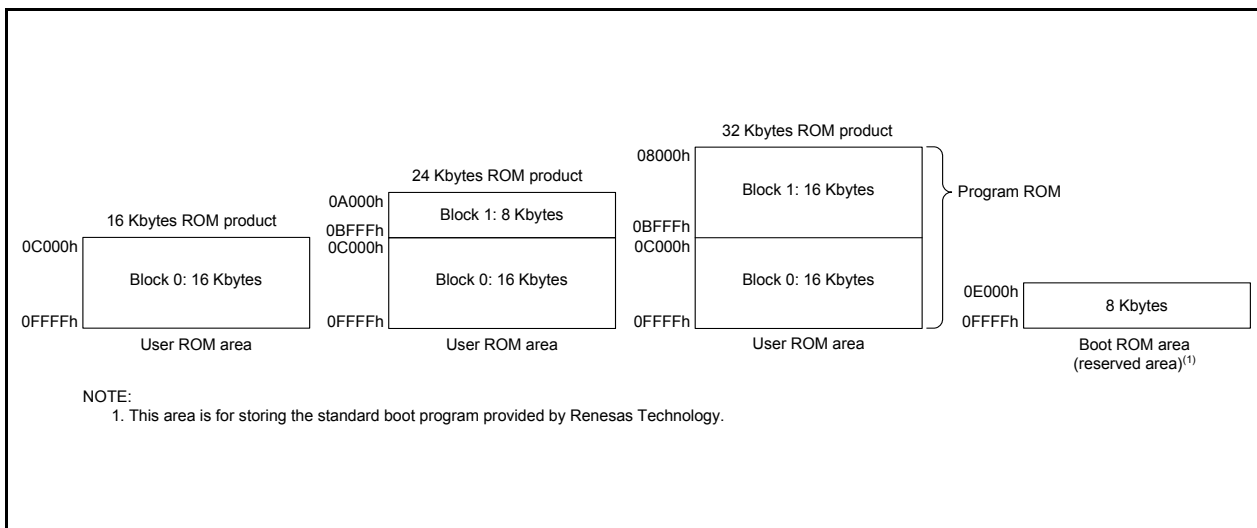


Figure 20.1 Flash Memory Block Diagram for R8C/2G Group

## 20.3 Functions to Prevent Rewriting of Flash Memory

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

### 20.3.1 ID Code Check Function

The ID code check function is used in standard serial I/O mode. Unless 3 bytes (addresses from 0FFFCh to 0FFFEh) of the reset vector are set to FFFFFFFh, the ID codes sent from the serial programmer or the on-chip debugging emulator and the 7-byte ID codes written in the flash memory are checked to see if they match. If the ID codes do not match, the commands sent from the serial programmer or the on-chip debugging emulator are not acknowledged. For details of the ID code check function, refer to **14. ID Code Areas**.

### 20.3.2 ROM Code Protect Function

The ROM protect function prevents the contents of the flash memory from being read, rewritten, or erased by means of the OFS register when parallel I/O mode is used.

Figure 20.2 shows the OFS Register. Refer to **15. Option Function Select Area** for details of the OFS register.

The ROM code protect function is enabled by writing 0 to the ROMCP1 bit and 1 to the ROMCR bit. It disables reading or changing the contents of the on-chip flash memory.

Once ROM code protect is enabled, the content in the internal flash memory cannot be rewritten in parallel I/O mode. To disable ROM code protect, erase the block including the OFS register with CPU rewrite mode or standard serial I/O mode.

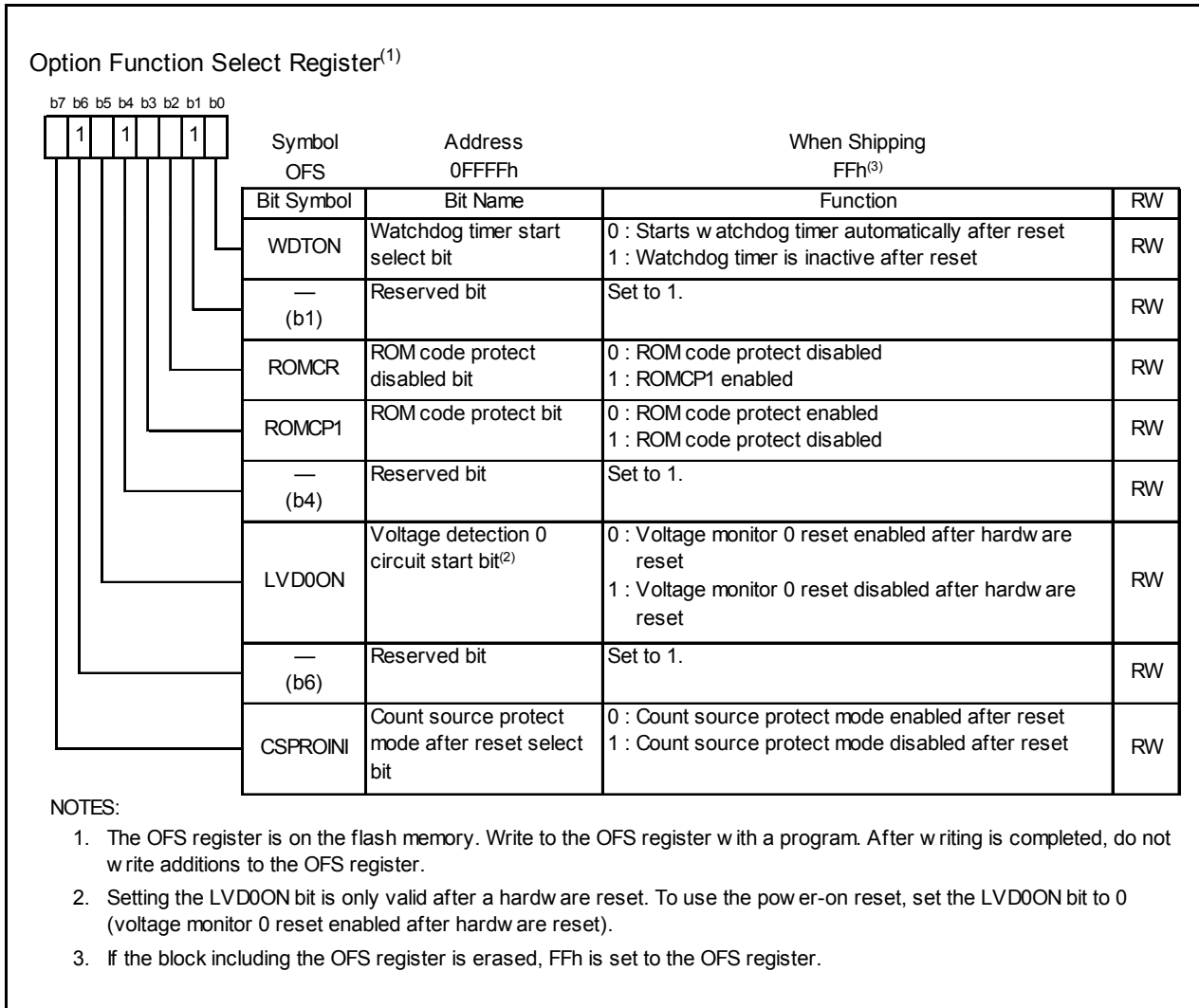


Figure 20.2 OFS Register

## 20.4 CPU Rewrite Mode

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

Table 20.3 lists the Differences between EW0 Mode and EW1 Mode.

**Table 20.3 Differences between EW0 Mode and EW1 Mode**

Item	EW0 Mode	EW1 Mode
Operating mode	Single-chip mode	Single-chip mode
Areas in which a rewrite control program can be executed	RAM (Rewrite control program is executed after being transferred)	User ROM or RAM
Areas which can be rewritten	User ROM	User ROM However, blocks which contain a rewrite control program are excluded
Software command restrictions	None	<ul style="list-style-type: none"> <li>• Program and block erase commands Cannot be run on any block which contains a rewrite control program</li> <li>• Read status register command Cannot be executed</li> </ul>
Modes after program or erase	Read status register mode	Read array mode
Modes after read status register	Read status register mode	Do not execute this command
CPU status during auto-write and auto-erase	Operating	Hold state (I/O ports hold state before the command is executed)
Flash memory status detection	<ul style="list-style-type: none"> <li>• Read bits FMR00, FMR06, and FMR07 in the FMR0 register by a program</li> <li>• Execute the read status register command and read bits SR7, SR5, and SR4 in the status register.</li> </ul>	Read bits FMR00, FMR06, and FMR07 in the FMR0 register by a program
CPU clock	5 MHz or below	No restriction (on clock frequency to be used)

## 20.4.1 Register Description

The registers used in CPU rewrite mode are described.

### 20.4.1.1 FMR0 Register (FMR0)

Figure 20.3 shows the FMR0 Register.

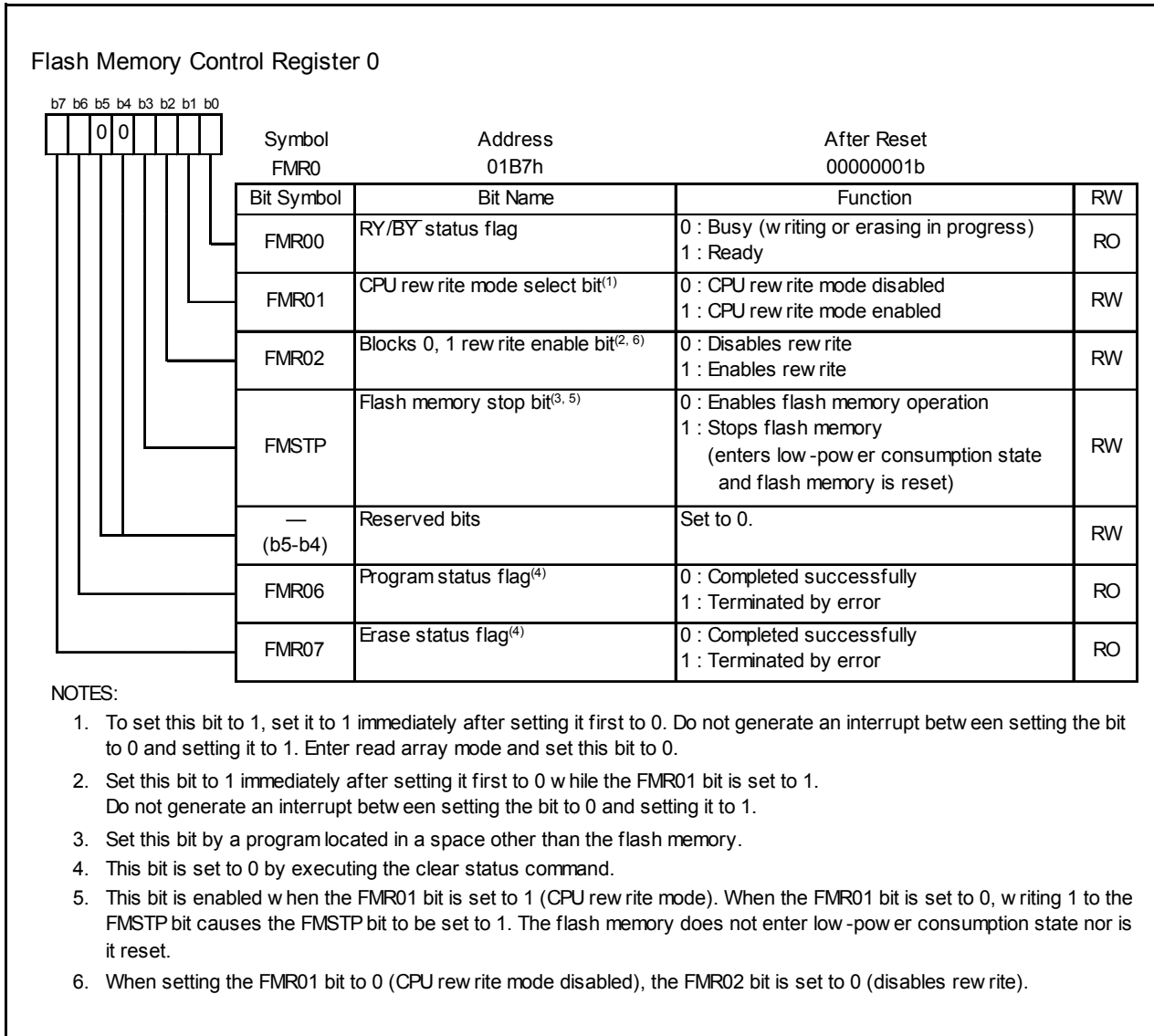


Figure 20.3 FMR0 Register

- **FMR00 Bit**  
This bit indicates the operating status of the flash memory. The bit's value is 0 during programming, erasure, or erase-suspend mode; otherwise, it is 1.
- **FMR01 Bit**  
The MCU is made ready to accept commands by setting the FMR01 bit to 1 (CPU rewrite mode).
- **FMR02 Bit**  
Rewriting of blocks 0 and 1 does not accept program or block erase commands if the FMR02 bit is set to 0 (rewrite disabled).  
Rewriting of blocks 0 and 1 is controlled by bits FMR15 and FMR16 if the FMR02 bit is set to 1 (rewrite enabled).
- **FMSTP Bit**  
This bit is used to initialize the flash memory control circuits, and also to reduce the amount of current consumed by the flash memory. Access to the flash memory is disabled by setting the FMSTP bit to 1. Therefore, the FMSTP bit must be written to by a program transferred to the RAM.  
In the following cases, set the FMSTP bit to 1:
  - When flash memory access resulted in an error while erasing or programming in EW0 mode (FMR00 bit not reset to 1 (ready))
  - To provide lower consumption in low-speed on-chip oscillator mode and low-speed clock mode.

Note that when going to stop or wait mode while the CPU rewrite mode is disabled, the FMR0 register does not need to be set because the power for the flash memory is automatically turned off and is turned back on again after returning from stop or wait mode.
- **FMR06 Bit**  
This is a read-only bit indicating the status of an auto-program operation. The bit is set to 1 when a program error occurs; otherwise, it is set to 0. For details, refer to the description in **Table 20.4 Errors and FMR0 Register Status**.
- **FMR07 Bit**  
This is a read-only bit indicating the status of an auto-erase operation. The bit is set to 1 when an erase error occurs; otherwise, it is set to 0. Refer to **Table 20.4 Errors and FMR0 Register Status** for details.

**Table 20.4 Errors and FMR0 Register Status**

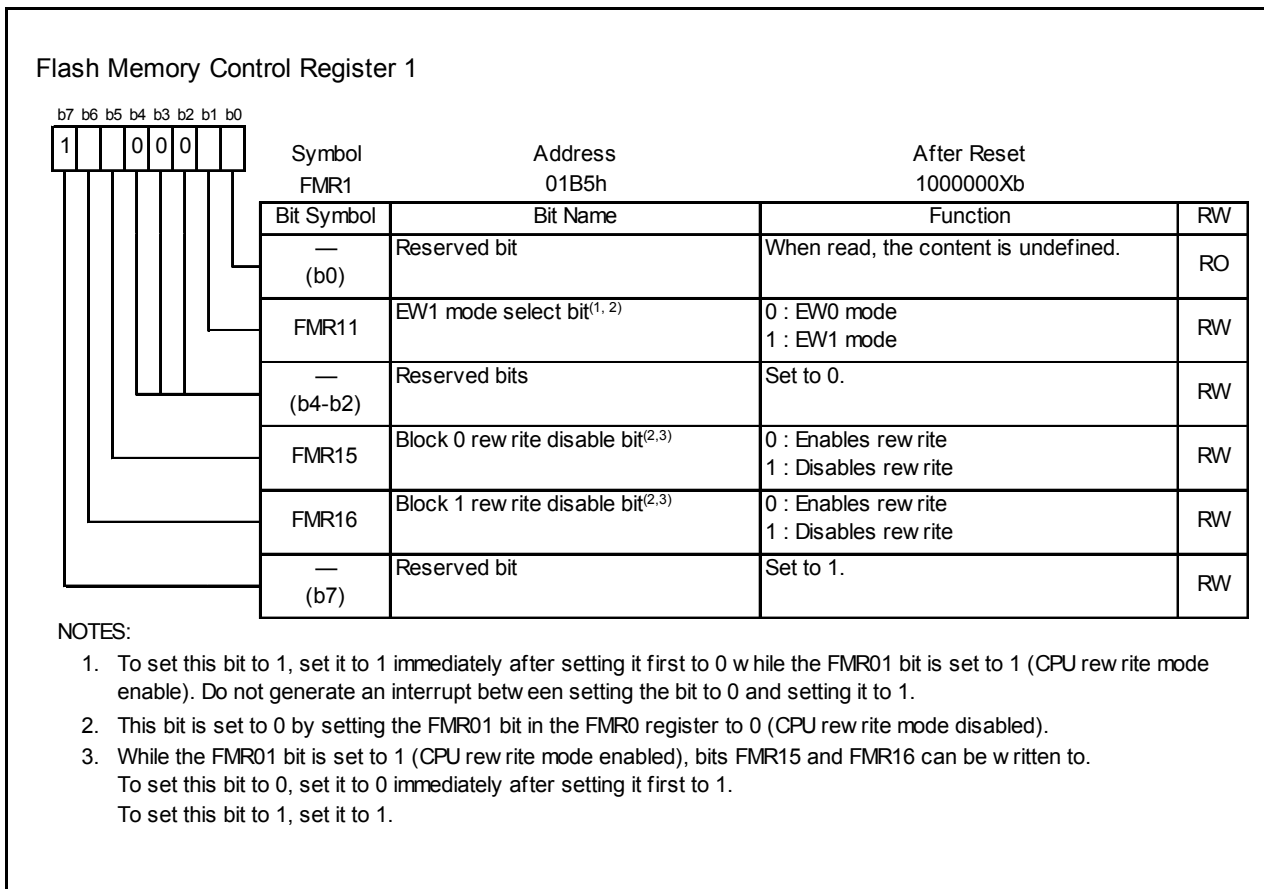
FMR0 Register (Status Register) Status		Error	Error Occurrence Condition
FMR07(SR5)	FMR06(SR4)		
1	1	Command sequence error	<ul style="list-style-type: none"> <li>• When a command is not written correctly.</li> <li>• When D0h or FFh is not written in the 2nd byte of the block erase command.<sup>(1)</sup></li> <li>• When the program command or block erase command is executed while rewriting is disabled by the FMR02 bit in the FMR0 register, or the FMR15 or FMR16 bit in the FMR1 register.</li> <li>• When an address not allocated in flash memory is input during erase command input</li> <li>• When attempting to erase the block for which rewriting is disabled during erase command input.</li> <li>• When an address not allocated in flash memory is input during write command input.</li> <li>• When attempting to write to a block for which rewriting is disabled during write command input.</li> </ul>
1	0	Erase error	<ul style="list-style-type: none"> <li>• When the block erase command is executed but auto-erasure does not complete correctly</li> </ul>
0	1	Program error	<ul style="list-style-type: none"> <li>• When the program command is executed but not auto-programming does not complete.</li> </ul>
0	0	Completed successfully	–

## NOTE:

1. When FFh is written in the 2nd byte of the block erase command, the MCU enters read array mode, and the command code written in the 1st byte is disabled.

### 20.4.1.2 FMR1 Register (FMR1)

Figure 20.4 shows the FMR1 Register.



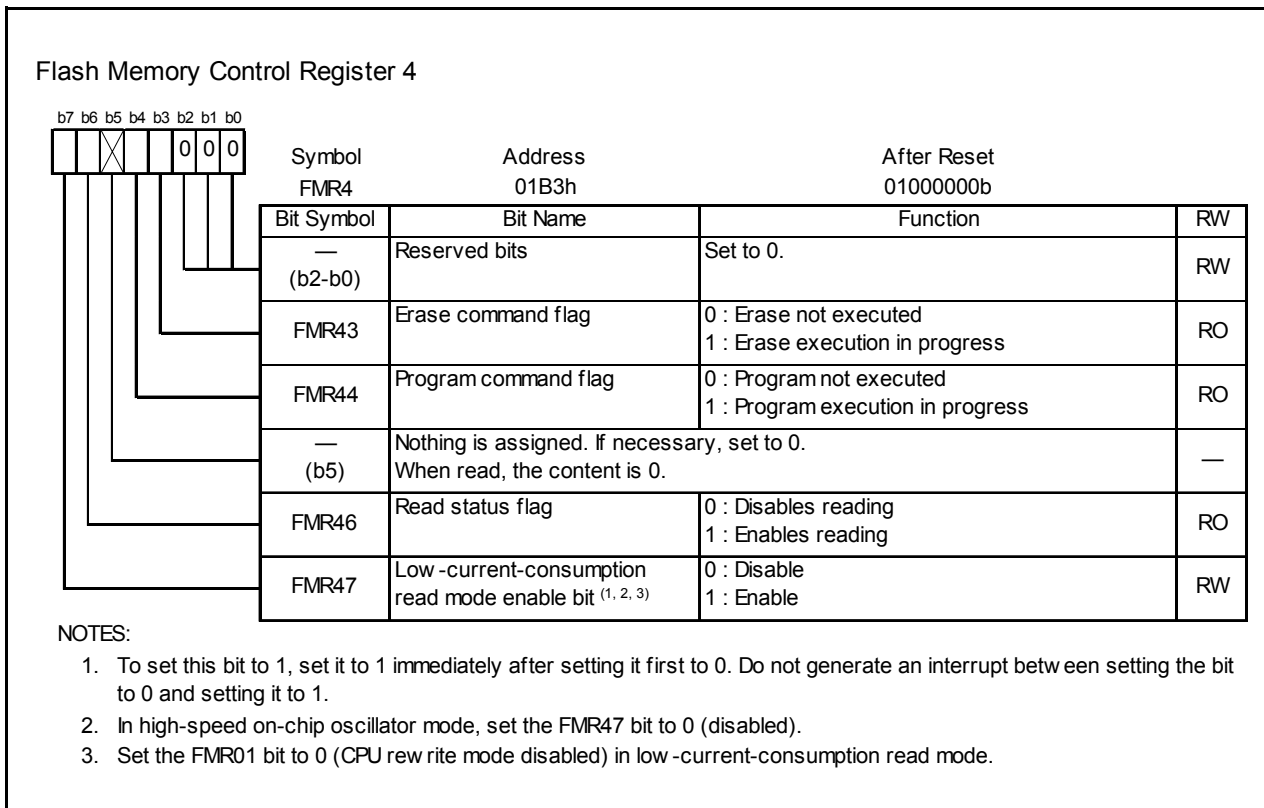
**Figure 20.4 FMR1 Register**

- **FMR11 Bit**  
Setting this bit to 1 (EW1 mode) places the MCU in EW1 mode.
- **FMR15 Bit**  
When the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit is set to 0 (rewrite enabled), block 0 accepts program and block erase commands.
- **FMR16 Bit**  
When the FMR02 bit is set to 1 (rewrite enabled) and the FMR16 bit is set to 0 (rewrite enabled), block 1 accepts program and block erase commands.



### 20.4.1.3 FMR4 Register (FMR4)

Figure 20.5 shows the FMR4 Register.



**Figure 20.5 FMR4 Register**

- FMR43 Bit**  
 When the auto-erase operation starts, the FMR43 bit is set to 1 (erase execution in progress).  
 When the auto-erase operation ends, the FMR43 bit is set to 0 (erase not executed).
- FMR44 Bit**  
 When the auto-program operation starts, the FMR44 bit is set to 1 (program execution in progress).  
 When the auto-program operation ends, the FMR44 bit is set to 0 (program not executed).
- FMR46 Bit**  
 The FMR46 bit is set to 0 (reading disabled) during auto-program or auto-erase execution. Do not access the flash memory while this bit is set to 0.
- FMR47 Bit**  
 Current consumption when reading the flash memory can be reduced by setting the FMR47 bit to 1 (enabled) in low-speed clock mode and low-speed on-chip oscillator mode.  
 Refer to **21.2.10 Low-Current-Consumption Read Mode** for details of the handling procedure.

### 20.4.2 Status Check Procedure

When an error occurs, bits FMR06 to FMR07 in the FMR0 register are set to 1, indicating the occurrence of an error. Therefore, checking these status bits (full status check) can be used to determine the execution result. Figure 20.6 shows the Full Status Check and Handling Procedure for Individual Errors.

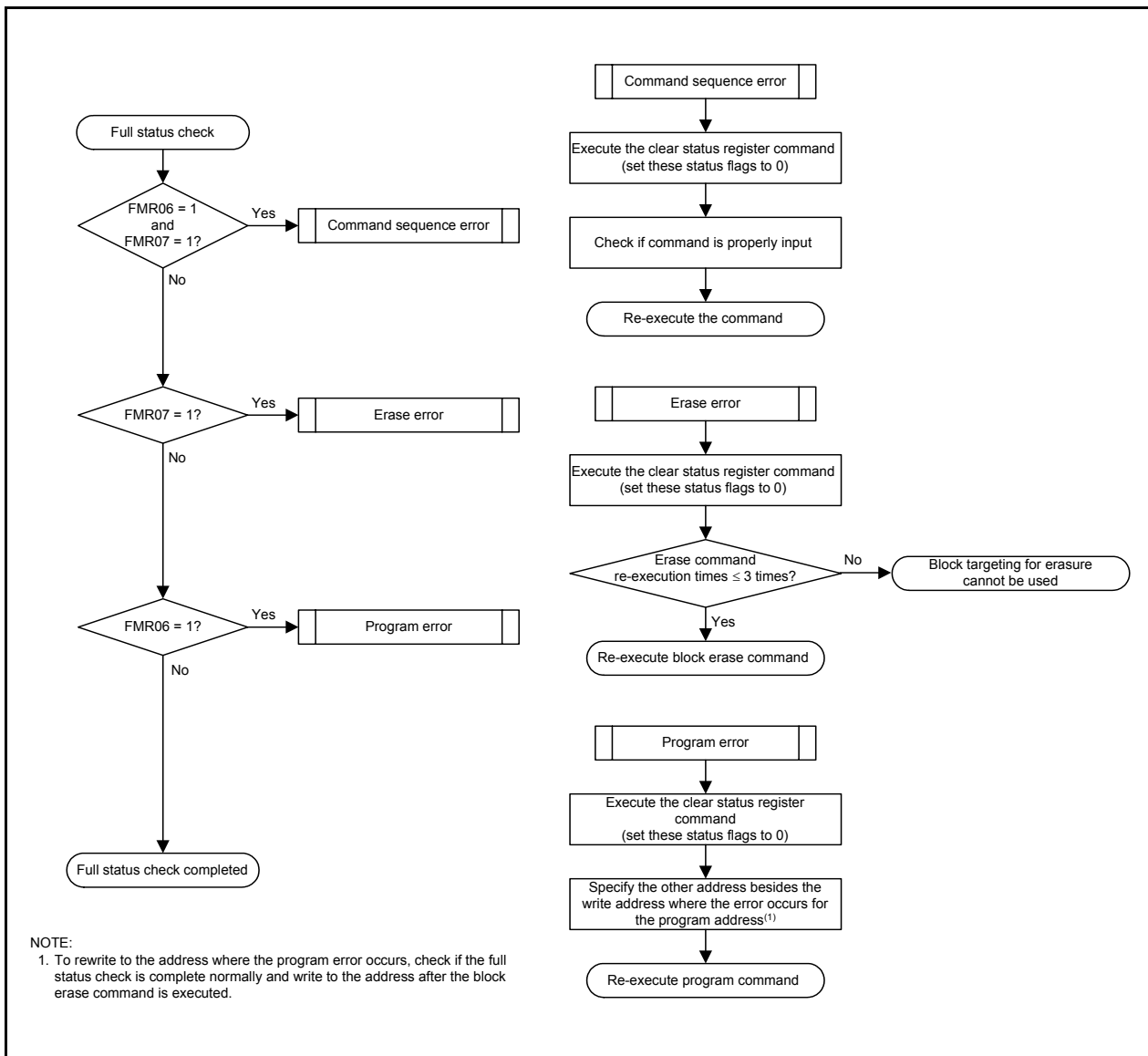


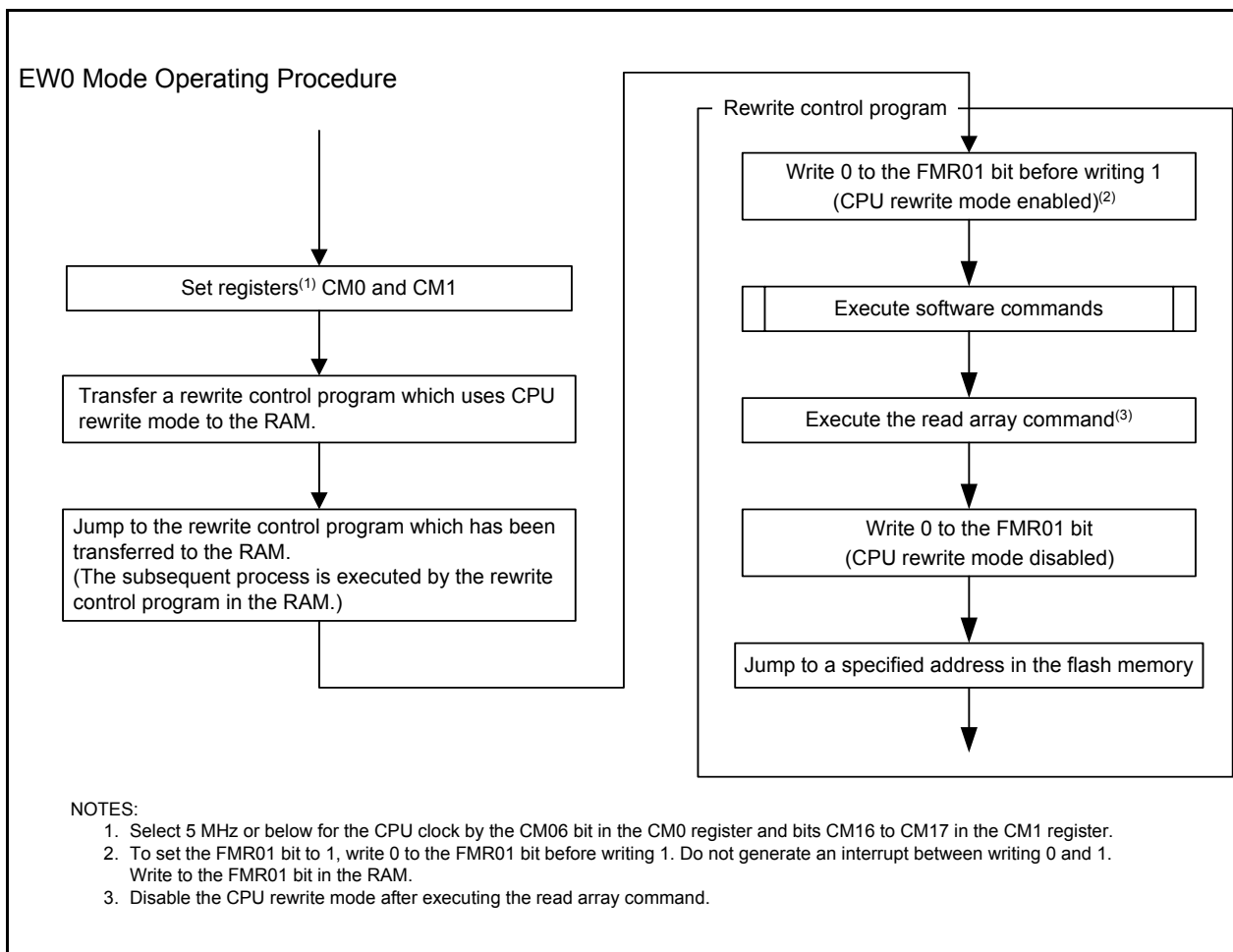
Figure 20.6 Full Status Check and Handling Procedure for Individual Errors

### 20.4.3 EW0 Mode

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

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

Figure 20.7 shows How to Set and Exit EW0 Mode.



**Figure 20.7 How to Set and Exit EW0 Mode**

### 20.4.3.1 Software Commands

There are five types of software commands:

- Read array
- Read status register
- Clear status register
- Program
- Block erase

Figure 20.8 shows Software Command Status Transition Diagram in EW0 Mode.

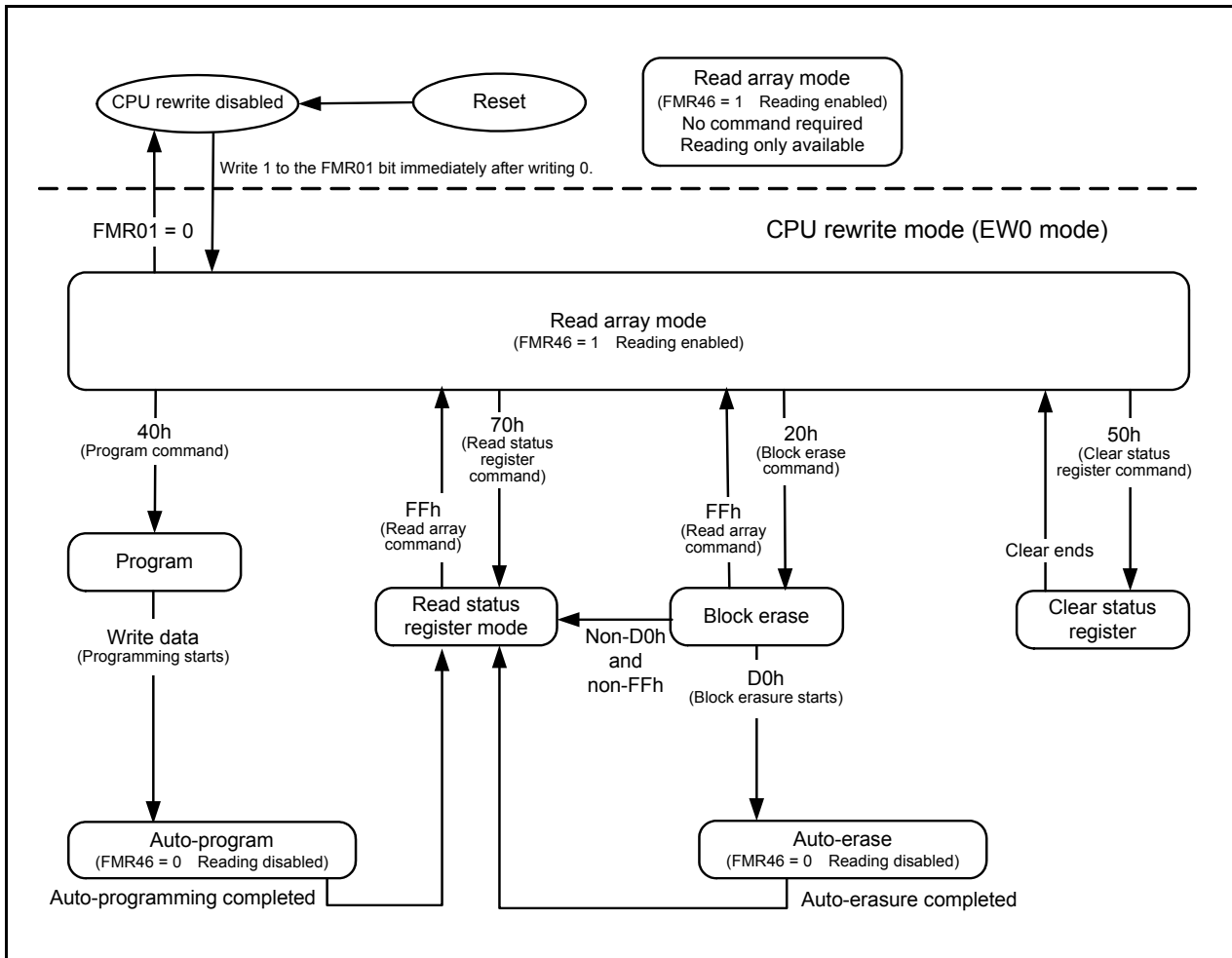


Figure 20.8 Software Command Status Transition Diagram in EW0 Mode

- Read Array Command

The read array command reads the flash memory.

When FFh is written to an address in the user ROM area, the MCU enters read array mode. In this mode, the contents of the specified address can be read.

Read array mode continues until other commands are written. The MCU enters this mode after a reset is deasserted.

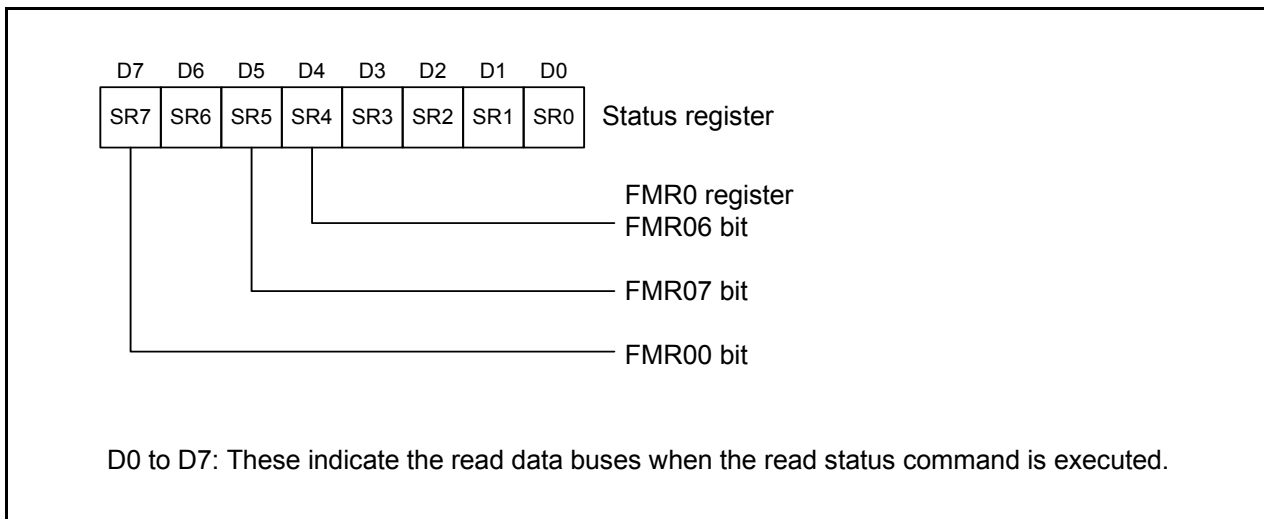
- Read Status Register Command

The read status register command is used to read the status register. Figure 20.9 shows Status Register.

The status register indicates the operating status of the flash memory and whether an erase or program operation has completed normally or in error (refer to **Table 20.4 Errors and FMR0 Register Status**). When 70h is written to an address in the user ROM area, the MCU enters read status register mode. When the address in the user ROM area is read subsequently, the status register can be read.

The MCU remains in read status register mode until the next read array command is written.

The status of the status register can be determined by reading bits FMR00, FMR06, and FMR07 in the FMR0 register.



**Figure 20.9 Status Register**

- Clear Status Register Command

The clear status register command sets the status register to 0.

When 50h is written to an address in the user ROM area, bits FMR07 and FMR06 in the FMR0 register and bits SR5 and SR4 in the status register are set to 00b.

- Program Command

The program command writes data to the flash memory in 1-byte units.

When 40h is written and then data is written to the write address, an auto-program operation (data program and verify) starts.

The FMR00 bit in the FMR0 register can be used to determine whether auto-programming has completed. The FMR00 bit is set to 0 during auto-programming and set to 1 when auto-programming completes.

The FMR06 bit in the FMR0 register can be used to determine the result of auto-programming after it has been finished (refer to **20.4.2 Status Check Procedure**).

Do not write additions to the already programmed addresses.

Also, when the FMR02 bit in the FMR0 register is set to 0 (rewrite disabled), or the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register is set to 1 (rewrite disabled), program commands targeting block 0 are not acknowledged. When the FMR 16 bit is set to 1 (rewrite disabled), program commands targeting block 1 are not acknowledged.

Figure 20.10 shows the Program Command in EW0 Mode.

In EW0 mode, the MCU enters read status register mode at the same time auto-programming starts and the status register can be read. In this case, the MCU remains in read status register mode until the next read array command is written.

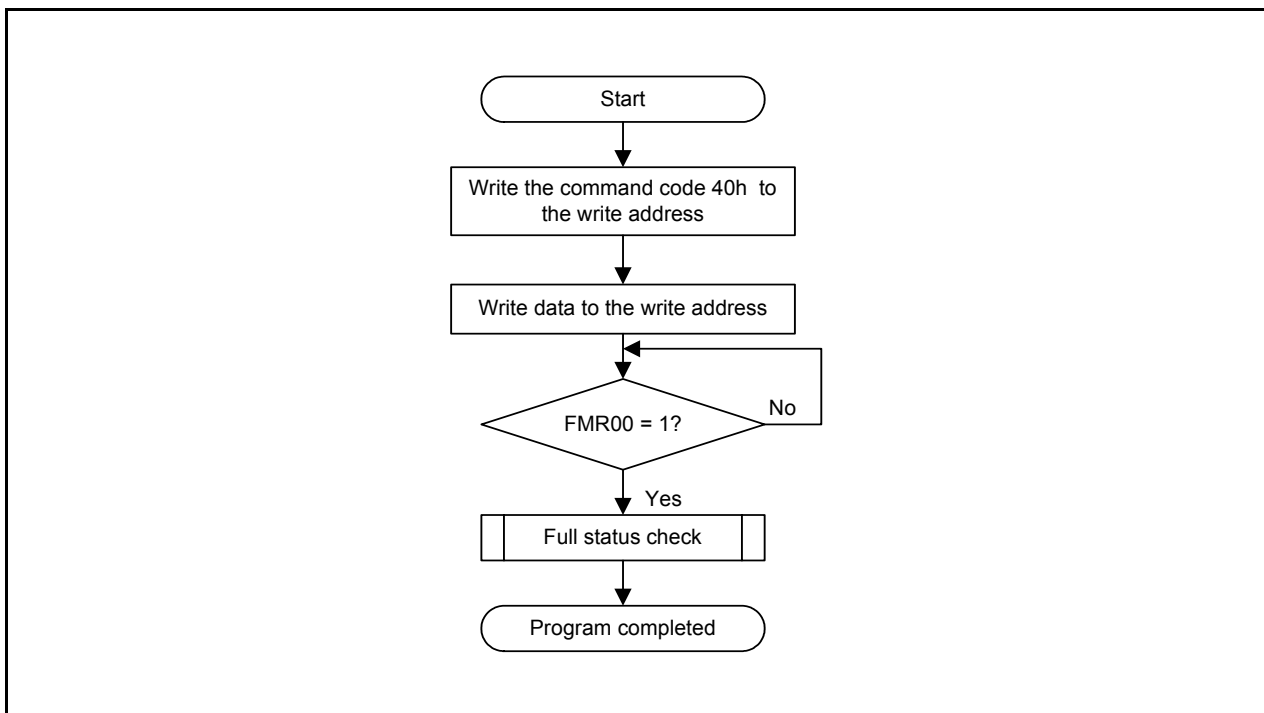


Figure 20.10 Program Command in EW0 Mode

- Block Erase

When 20h is first written and then D0h is written to a given block address, an auto-erase operation (erase and verify) of the specified block starts.

The FMR00 bit in the FMR0 register can be used to determine whether auto-erasure has completed.

The FMR00 bit is set to 0 during auto-erasure and set to 1 when auto-erasure completes.

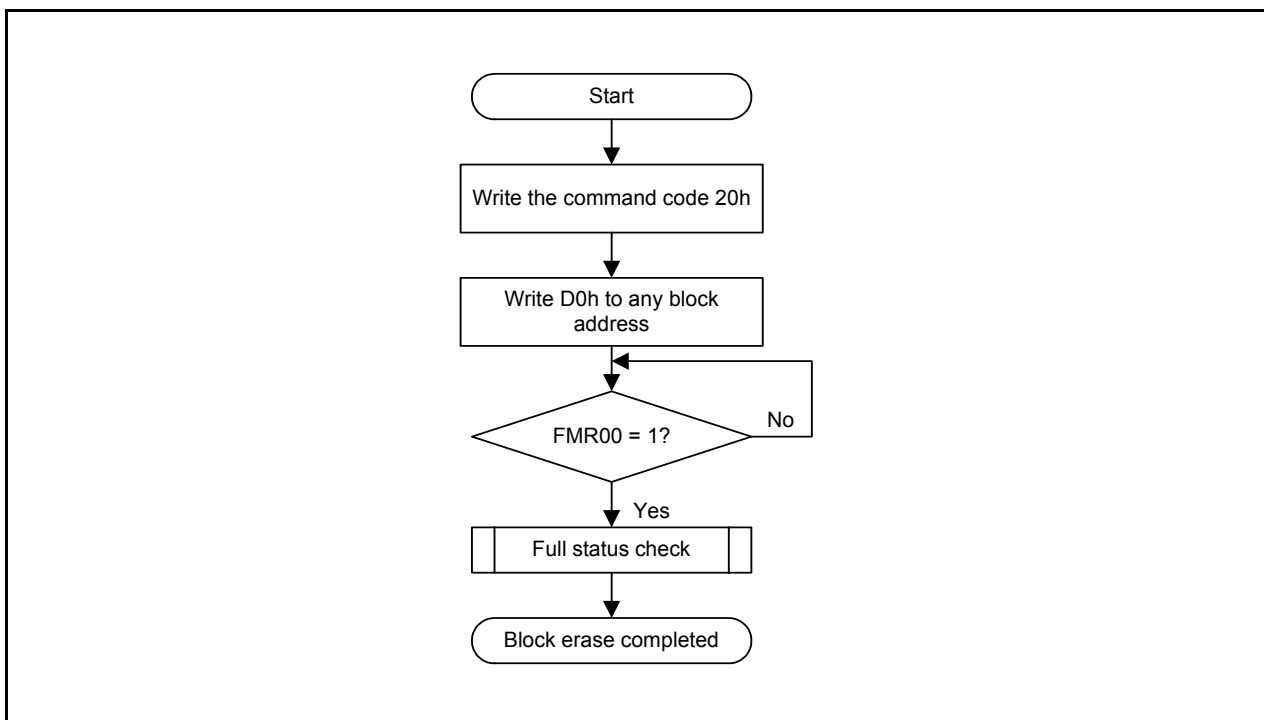
The FMR07 bit in the FMR0 register can be used to determine the result of auto-erasure after auto-erasure has completed (refer to **20.4.2 Status Check Procedure**).

Also, when the FMR02 bit in the FMR0 register is set to 0 (rewrite disabled), or the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register is set to 1 (rewrite disabled), block erase commands targeting block 0 are not acknowledged. When the FMR16 bit is set to 1 (rewrite disabled), block erase commands targeting block 1 are not acknowledged.

In EW0 mode, the MCU enters read status register mode at the same time auto-erasure starts and the status register can be read. In this case, the MCU remains in read status register mode until the next read array command is written.

Figure 20.11 shows the Block Erase Command in EW0 Mode.

If the programming and erasure endurance is  $n$  ( $n = 100, 1000, \text{ or } 10,000$ ), each block can be erased  $n$  times. For example, if 1,024 1-byte writes are performed to block A, a 1-Kbyte block, and then the block is erased, the erase count stands at one. When performing 100 or more rewrites, the actual erase count can be reduced by executing programming operations in such a way that all blank areas are used before performing an erase operation. Avoid rewriting only particular blocks and try to average out the programming and erasure endurance of the blocks. It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.



**Figure 20.11** Block Erase Command in EW0 Mode

### 20.4.3.2 EW0 Mode Interrupts

In EW0 mode, maskable interrupts can be used by allocating a vector in RAM. Table 20.5 lists the EW0 Mode Interrupts. Refer to **20.7.1.3 Non-Maskable Interrupts** for details of the non-maskable interrupt.

**Table 20.5 EW0 Mode Interrupts**

Status	When Maskable Interrupt Request is Acknowledged
During auto-erasure	Interrupt handling is executed.
Auto-programming	



### 20.4.4 EW1 Mode

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

The FMR0 register can be used to determine when program and erase operations complete. Figure 20.12 shows How to Set and Exit EW1 Mode.

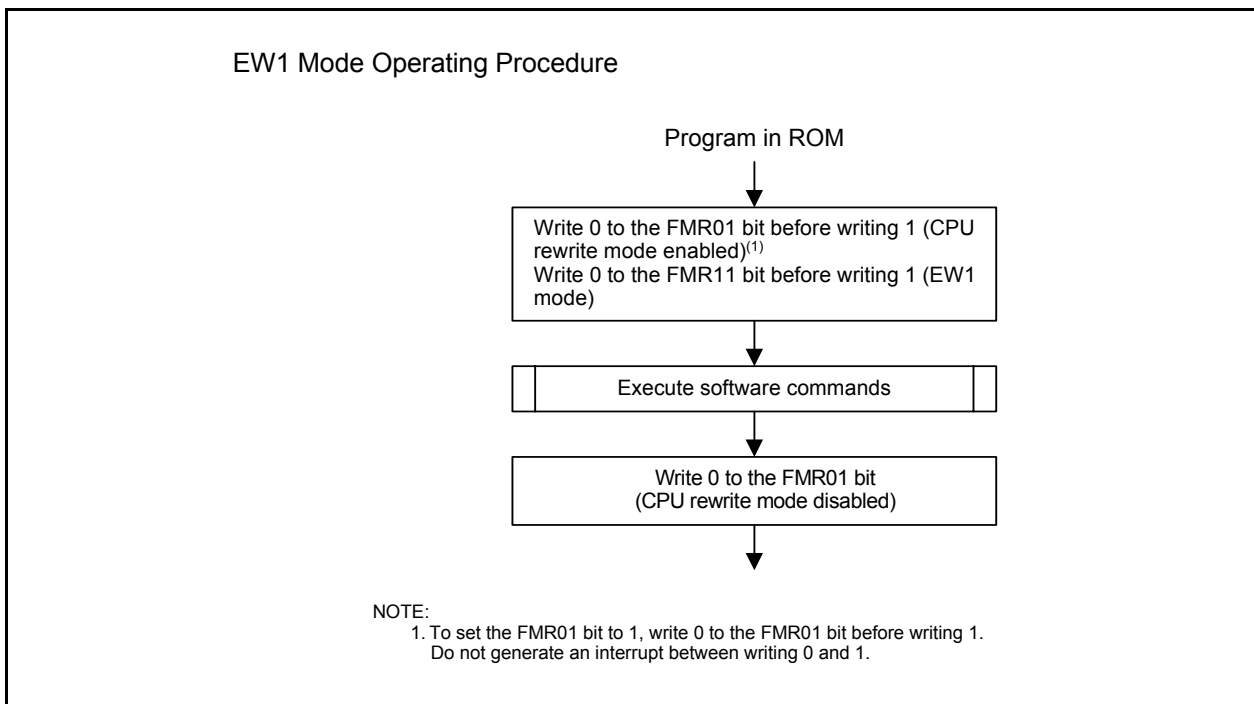


Figure 20.12 How to Set and Exit EW1 Mode

### 20.4.4.1 Software Commands

There are four types of software commands:

- Read array
- Clear status register
- Program
- Block erase

Do not execute read status register command in EW1 mode.

Figure 20.13 shows Software Command Status Transition Diagram in EW1 Mode.

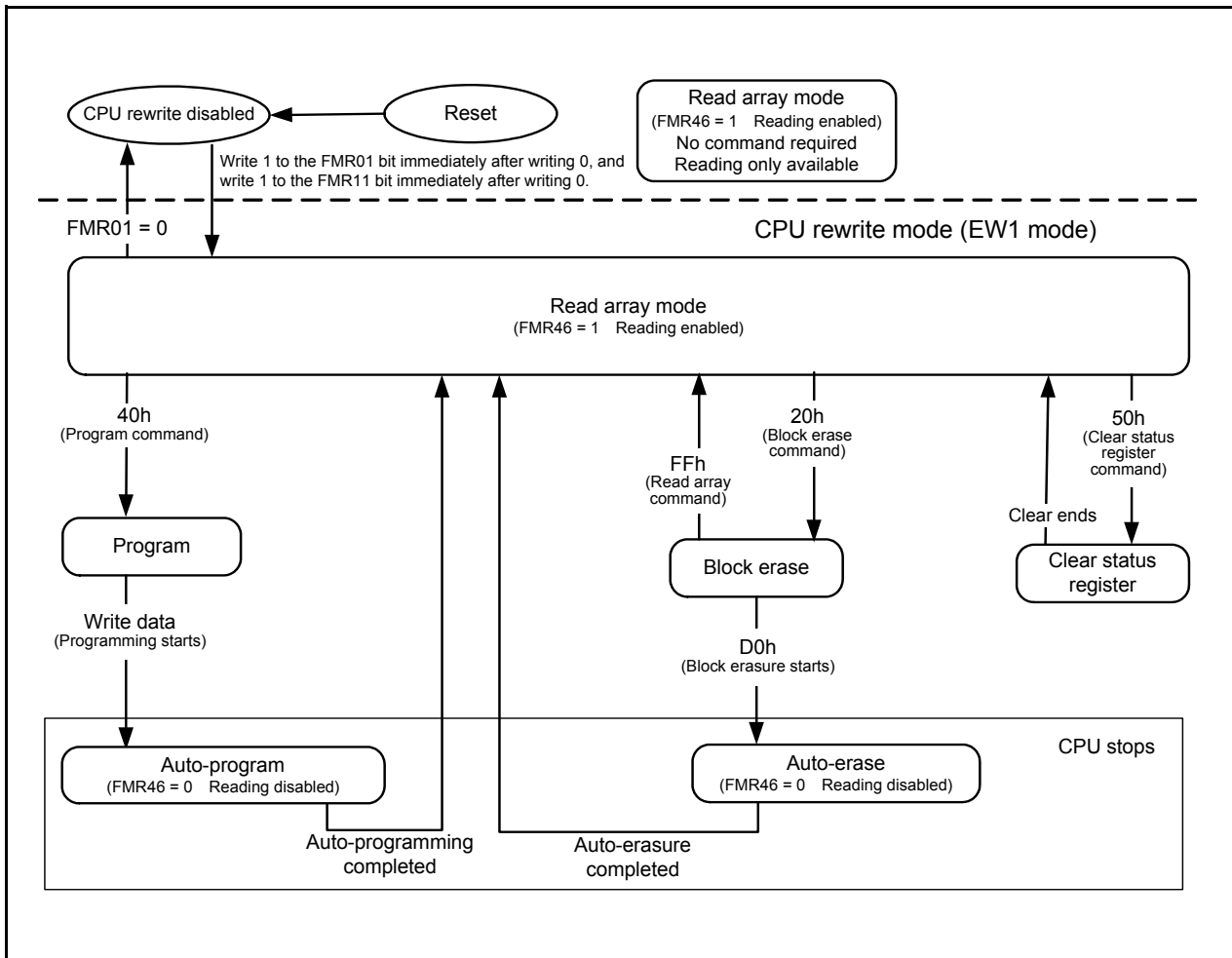


Figure 20.13 Software Command Status Transition Diagram in EW1 Mode

- Read Array Command  
The read array command reads the flash memory.  
When FFh is written to an address in the user ROM area, the MCU enters read array mode. In this mode, the contents of the specified address can be read.  
Read array mode continues until other commands are written. The MCU enters this mode after a reset is deasserted.
- Clear Status Register Command  
The clear status register command sets the status register to 0.  
When 50h is written to an address in the user ROM area, bits FMR07 and FMR06 in the FMR0 register and bits SR5 and SR4 in the status register are set to 00b.

- Program Command

The program command writes data to the flash memory in 1-byte units.

When 40h is written and then data is written to the write address, an auto-program operation (data program and verify) starts.

The FMR00 bit in the FMR0 register can be used to determine whether auto-programming has completed.

The FMR00 bit is set to 0 during auto-programming and set to 1 when auto-programming completes.

The FMR06 bit in the FMR0 register can be used to determine the result of auto-programming after it has been finished (refer to **20.4.2 Status Check Procedure**).

Do not write additions to the already programmed addresses.

Also, when the FMR02 bit in the FMR0 register is set to 0 (rewrite disabled), or the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register is set to 1 (rewrite disabled), program commands targeting block 0 are not acknowledged. When the FMR 16 bit is set to 1 (rewrite disabled), program commands targeting block 1 are not acknowledged.

In EW1 mode, do not execute this command for any address which a rewrite control program is allocated.

Figure 20.14 shows the Program Command in EW1 Mode.

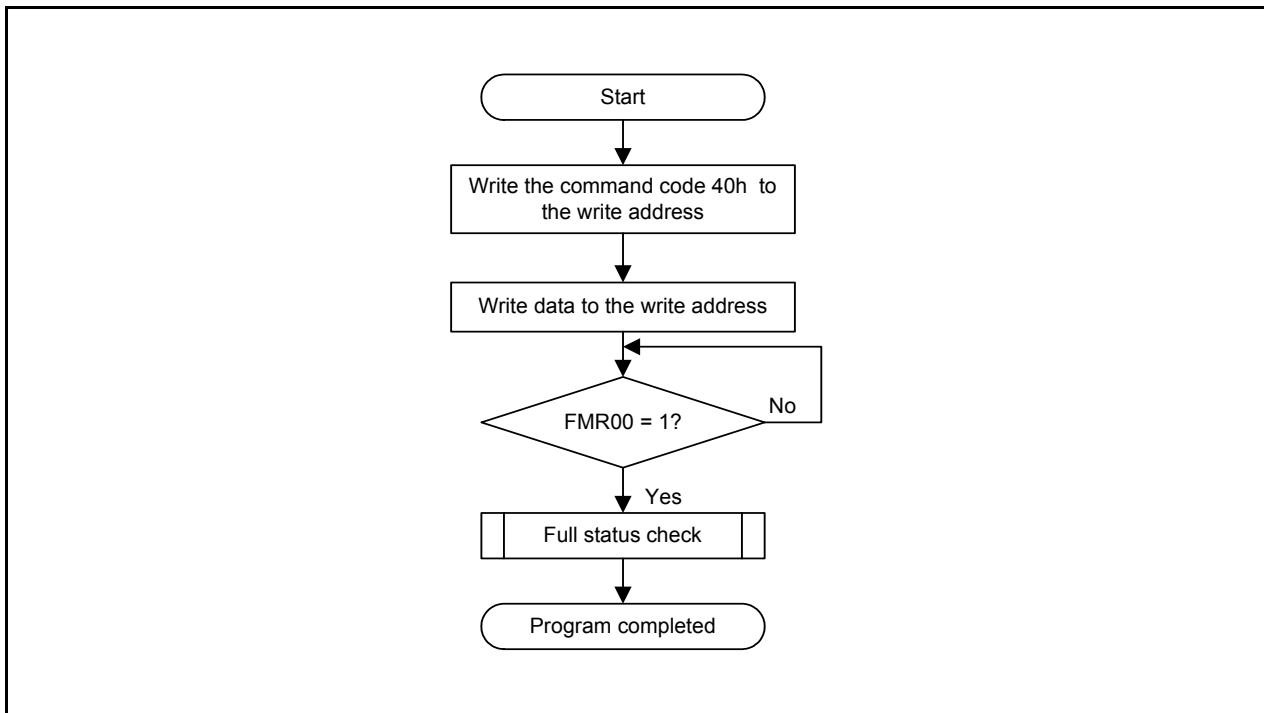


Figure 20.14 Program Command in EW1 Mode

- Block Erase

When 20h is first written and then D0h is written to a given block address, an auto-erase operation (erase and verify) of the specified block starts.

The FMR00 bit in the FMR0 register can be used to determine whether auto-erasure has completed.

The FMR00 bit is set to 0 during auto-erasure and set to 1 when auto-erasure completes.

The FMR07 bit in the FMR0 register can be used to determine the result of auto-erasure after auto-erasure has completed (refer to **20.4.2 Status Check Procedure**).

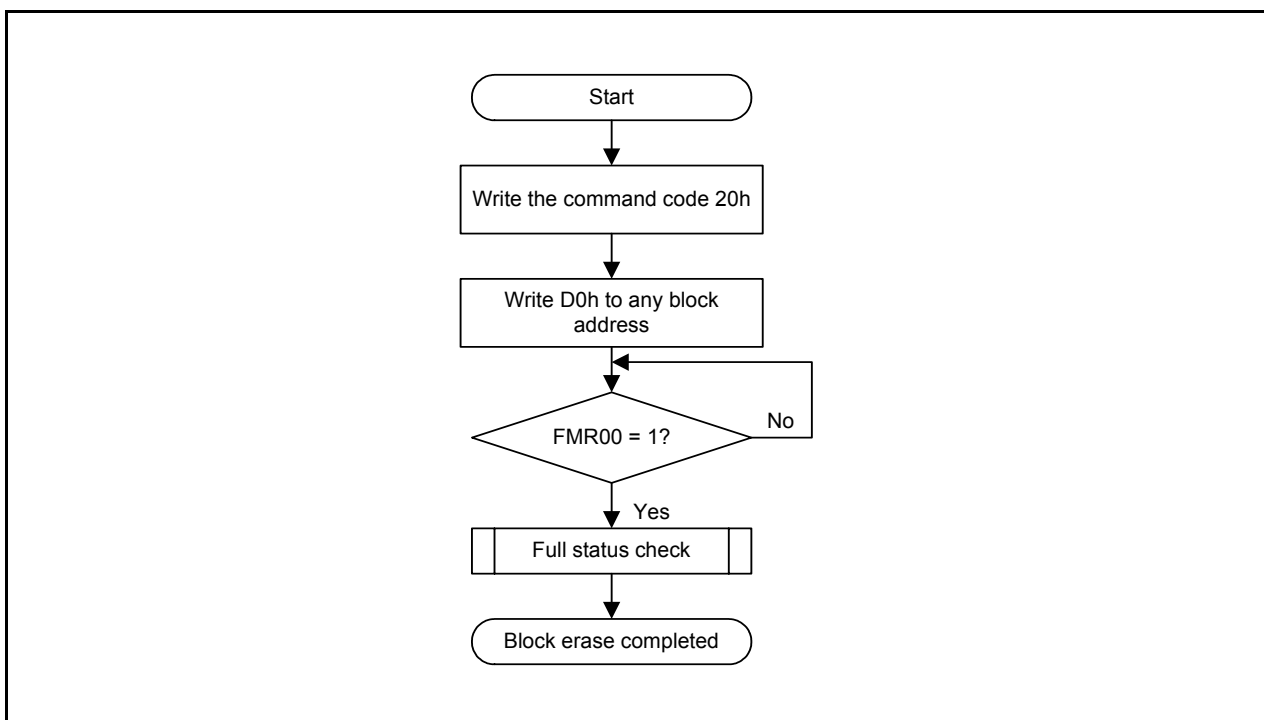
Also, when the FMR02 bit in the FMR0 register is set to 0 (rewrite disabled), or the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register is set to 1 (rewrite disabled), block erase commands targeting block 0 are not acknowledged. When the FMR16 bit is set to 1 (rewrite disabled), block erase commands targeting block 1 are not acknowledged.

Do not execute this command for any address to which a rewrite control program is allocated.

Figure 20.15 shows the Block Erase Command in EW1 Mode.

If the programming and erasure endurance is  $n$  ( $n = 100, 1000, \text{ or } 10,000$ ), each block can be erased  $n$  times. For example, if 1,024 1-byte writes are performed to block A, a 1-Kbyte block, and then the block is erased, the erase count stands at one. When performing 100 or more rewrites, the actual erase count can be reduced by executing programming operations in such a way that all blank areas are used before performing an erase operation. Avoid rewriting only particular blocks and try to average out the programming and erasure endurance of the blocks.

It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.



**Figure 20.15** Block Erase Command in EW1 Mode

### 20.4.4.2 EW1 Mode Interrupts

In EW1 mode, maskable interrupts can be used.

Table 20.6 lists the EW1 Mode Interrupts. Refer to **20.7.1.3 Non-Maskable Interrupts** for details of the non-maskable interrupt.

**Table 20.6 EW1 Mode Interrupts**

Status	When Maskable Interrupt Request is Acknowledged
During auto-erasure	Auto-erasure has priority and the interrupt request acknowledgement is put on standby. Interrupt handling is executed after auto-erasure completes.
During auto-programming	Auto-programming has priority and the interrupt request acknowledgement is put on standby. Interrupt handling is executed after auto-programming completes.

## 20.5 Standard Serial I/O Mode

In standard serial I/O mode, the user ROM area can be rewritten while the MCU is mounted on-board by using a serial programmer which is suitable for the MCU.

There are three types of standard serial I/O modes:

- Standard serial I/O mode 1 .....Clock synchronous serial I/O used to connect with a serial programmer
- Standard serial I/O mode 2 .....Clock asynchronous serial I/O used to connect with a serial programmer
- Standard serial I/O mode 3 .....Special clock asynchronous serial I/O used to connect with a serial programmer

This MCU uses Standard serial I/O mode 3.

Refer to **Appendix 2. Connection Examples with On-Chip Debugging Emulator**. Contact the manufacturer of your serial programmer for details. Refer to the user's manual of your serial programmer for instructions on how to use it.

Table 20.7 lists the Pin Functions (Flash Memory Standard Serial I/O Mode 3), and Figure 20.16 shows an Example of Pin Processing in Standard Serial I/O Mode 3.

After processing the pins shown in Table 20.7 and rewriting the flash memory using the programmer, apply "H" to the MODE pin and reset the hardware to run a program in the flash memory in single-chip mode.

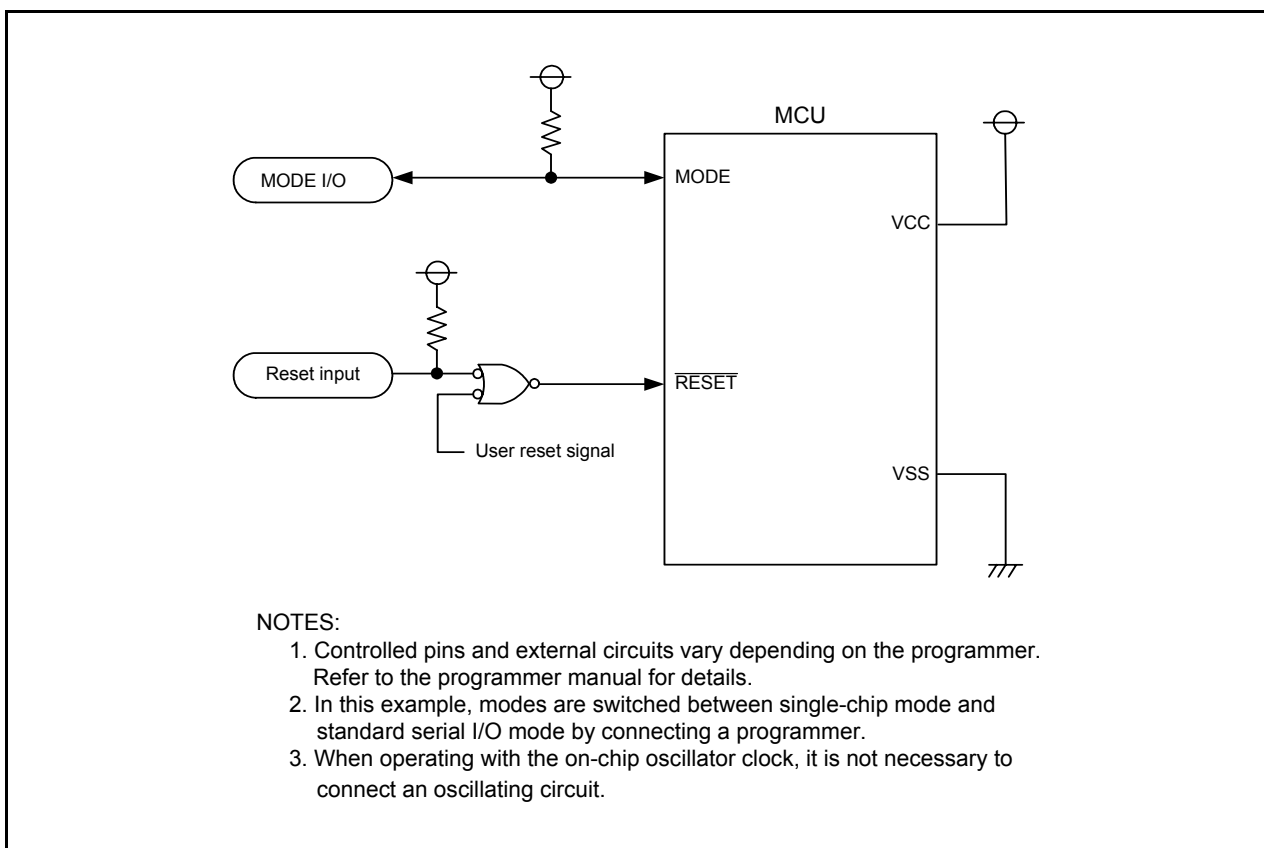
### 20.5.1 ID Code Check Function

The ID code check function determines whether the ID codes sent from the serial programmer and those written in the flash memory match.

Refer to **14. ID Code Areas** for details of the ID code check.

**Table 20.7 Pin Functions (Flash Memory Standard Serial I/O Mode 3)**

Pin	Name	I/O	Description
VCC,VSS	Power input		Apply the voltage guaranteed for programming and erasure to the VCC pin and 0 V to the VSS pin.
RESET	Reset input	I	Reset input pin.
P4_3/XCIN	P4_3 input/clock input	I	Connect crystal oscillator between pins XCIN and XCOOUT when connecting external oscillator.
P4_4/XCOOUT	P4_4 output/clock output	O	To use P4_3 as an input port, input a "H" or "L" level signal or leave the pin open. To use P4_4 as an output port, leave the pin open.
P0_4 to P0_7	Input port P0	I	Input a "H" or "L" level signal or leave the pin open.
P1_0 to P1_7	Input port P1	I	
P3_0 to P3_7	Input port P3	I	
P4_5	Input port P4	I	
P6_0, P6_3 to P6_6	Input port P6	I	
MODE	MODE	I/O	

**Figure 20.16 Example of Pin Processing in Standard Serial I/O Mode 3**

## 20.6 Parallel I/O Mode

Parallel I/O mode is used to input and output software commands, addresses and data necessary to control (read, program, and erase) the on-chip flash memory. Use a parallel programmer which supports this MCU. Contact the manufacturer of the parallel programmer for more information, and refer to the user's manual of the parallel programmer for details on how to use it.

ROM areas shown in Figure 20.1 can be rewritten in parallel I/O mode.

### 20.6.1 ROM Code Protect Function

The ROM code protect function disables the reading and rewriting of the flash memory. (Refer to **20.3.2 ROM Code Protect Function**.)



## 20.7 Notes on Flash Memory

### 20.7.1 CPU Rewrite Mode

#### 20.7.1.1 Operating Speed

Before entering CPU rewrite mode (EW0 mode), select 5 MHz or below for the CPU clock using the CM06 bit in the CM0 register and bits CM16 to CM17 in the CM1 register. This does not apply to EW1 mode.

#### 20.7.1.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because they reference data in the flash memory: UND, INTO, and BRK.

#### 20.7.1.3 Non-Maskable Interrupts

- EW0 Mode

Once a watchdog timer, voltage monitor1, voltage monitor 2, comparator 1, or comparator 2 interrupt request is acknowledged, auto-erasure or auto-programming is forcibly stopped immediately and the flash memory is reset. Interrupt handling starts after a fixed period and the flash memory restarts.

As the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be readable. Execute auto-erasure again and ensure it completes normally.

The watchdog timer does not stop during command operation, so that interrupt requests may be generated. Initialize the watchdog timer regularly.

Do not use the address match interrupt while a command is being executed because the vector of the address match interrupt is allocated in ROM.

Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

- EW1 Mode

Once a watchdog timer, voltage monitor1, voltage monitor 2, comparator 1, or comparator 2 interrupt request is acknowledged, auto-erasure or auto-programming is forcibly stopped immediately and the flash memory is reset. Interrupt handling starts after a fixed period and the flash memory restarts.

As the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be readable. Execute auto-erasure again and ensure it completes normally.

The watchdog timer does not stop even during command operation, so that interrupt requests may be generated. Initialize the watchdog timer by using the erase-suspend function.

Do not use the address match interrupt while a command is being executed because the vector of the address match interrupt is allocated in ROM.

Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

#### 20.7.1.4 How to Access

Write 0 before writing 1 when setting Bits FMR01, FMR02 in the FMR0 register, or FMR11 bit in the FMR1 register to 1. Do not generate an interrupt between writing 0 and 1.

#### 20.7.1.5 Rewriting User ROM Area

In EW0 Mode, if the supply voltage drops while rewriting any block in which a rewrite control program is stored, it may not be possible to rewrite the flash memory because the rewrite control program cannot be rewritten correctly. In this case, use standard serial I/O mode.

#### 20.7.1.6 Program

Do not write additions to the already programmed address.

#### 20.7.1.7 Program and Erase Voltage for Flash Memory

To perform programming and erasure, use  $VCC = 2.7\text{ V}$  to  $5.5\text{ V}$  as the supply voltage. Do not perform programming and erasure at less than  $2.7\text{ V}$ .

## 21. Reducing Power Consumption

### 21.1 Overview

This chapter describes key points and processing methods for reducing power consumption.

### 21.2 Key Points and Processing Methods for Reducing Power Consumption

Key points for reducing power consumption are shown below. They should be referred to when designing a system or creating a program.

#### 21.2.1 Voltage Detection Circuit

When voltage monitor 1 and comparator 1 are not used, set the VCA26 bit in the VCA2 register to 0 (voltage detection 1 circuit disabled). When voltage monitor 2 and comparator 2 are not used, set the VCA27 bit in the VCA2 register to 0 (voltage detection 2 circuit disabled).

If the power-on reset and voltage monitor 0 reset are not used, set the VCA25 bit in the VCA2 register to 0 (voltage detection 0 circuit disabled).

#### 21.2.2 Ports

Even after the MCU enters wait mode or stop mode, the states of the I/O ports are retained. Current flows into the output ports in the active state, and shoot-through current flows into the input ports in the high-impedance state. Unnecessary ports should be set to input and fixed to a stable electric potential before the MCU enters wait mode or stop mode.

#### 21.2.3 Clocks

Power consumption generally depends on the number of the operating clocks and their frequencies. The fewer the number of operating clocks or the lower their frequencies, the more power consumption decreases. Unnecessary clocks should be stopped accordingly.

Stopping low-speed on-chip oscillator oscillation: CM14 bit in CM1 register

Stopping high-speed on-chip oscillator oscillation: HRA00 bit in HRA0 register

#### 21.2.4 Selecting Oscillation Drive Capacity

Set the drive capacity of the XCIN clock oscillation circuit to "LOW". Confirm that the circuit oscillates stably while it is in the "LOW" state.

Selecting XCIN-XCOUT drive capacity: CM03 bit in CM0 register

#### 21.2.5 Wait Mode, Stop Mode

Power consumption can be reduced in wait mode and stop mode. Refer to **11.4 Power Control** for details.

#### 21.2.6 Stopping Peripheral Function Clocks

If the peripheral function f1, f2, f4, f8, and f32 clocks are not necessary in wait mode, set the CM02 bit in the CM0 register to 1 (peripheral function clock stops in wait mode). This will stop the f1, f2, f4, f8, and f32 clocks in wait mode.

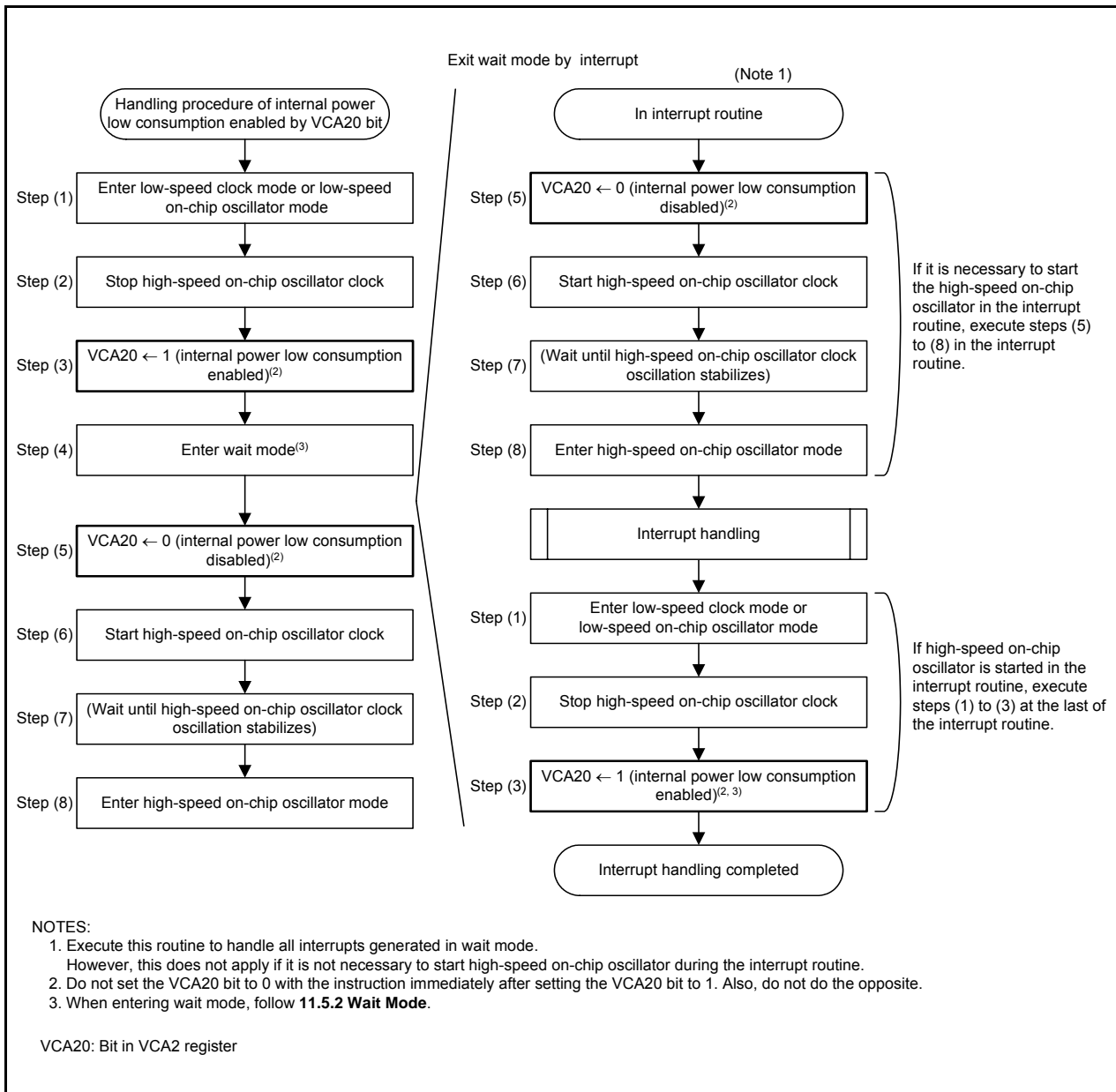
#### 21.2.7 Timers

If timer RA is not used, set the TCKCUT bit in the TRAMR register to 1 (count source cutoff).

If timer RB is not used, set the TCKCUT bit in the TRBMR register to 1 (count source cutoff).

### 21.2.8 Reducing Internal Power Consumption

When the MCU enters wait mode using low-speed clock mode or low-speed on-chip oscillator mode, internal power consumption can be reduced by using the VCA20 bit in the VCA2 register. Figure 21.1 shows the Handling Procedure of Internal Power Low Consumption Using VCA20 Bit. To enable internal power low consumption by the VCA20 bit, follow Figure 21.1 Handling Procedure of Internal Power Low Consumption Using VCA20 Bit.



**Figure 21.1 Handling Procedure of Internal Power Low Consumption Using VCA20 Bit**

### 21.2.9 Stopping Flash Memory

In low-speed on-chip oscillator mode and low-speed clock mode, power consumption can be further reduced by stopping the flash memory using the FMSTP bit in the FMR0 register.

Access to the flash memory is disabled by setting the FMSTP bit to 1 (flash memory stops). The FMSTP bit must be written to by a program transferred to RAM.

When the MCU enters stop mode or wait mode while CPU rewrite mode is disabled, the power for the flash memory is automatically turned off. It is turned back on again after the MCU exits stop mode or wait mode. This eliminates the need to set the FMR0 register.

Figure 21.2 shows the Handling Procedure Example of Low Power Consumption Using FMSTP Bit.

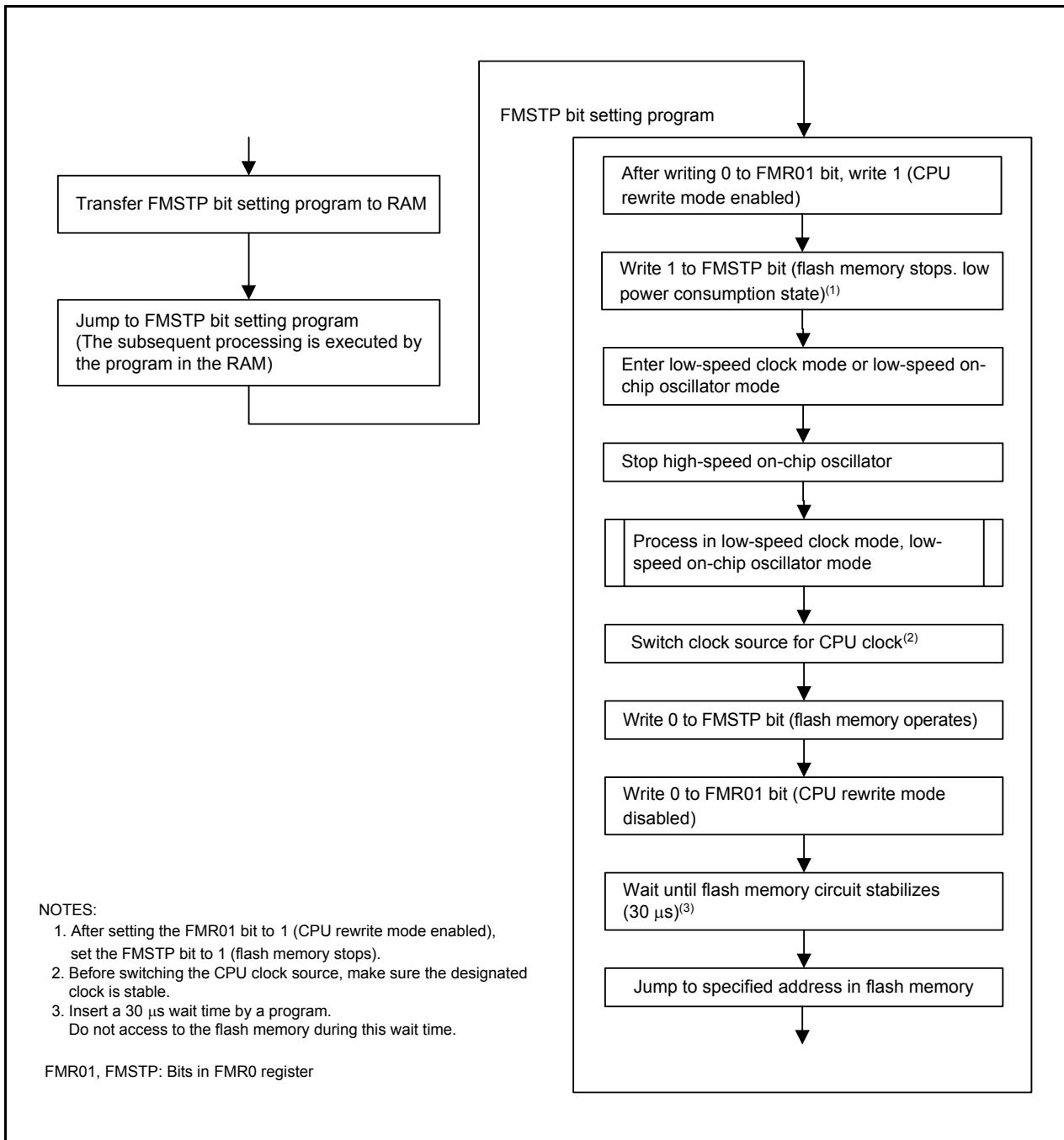


Figure 21.2 Handling Procedure Example of Low Power Consumption Using FMSTP Bit

### 21.2.10 Low-Current-Consumption Read Mode

In low-speed clock mode and low-speed on-chip oscillator mode, the current consumption when reading the flash memory can be reduced by setting the FMR47 bit in the FMR4 register to 1 (enabled).

Figure 21.3 shows the Handling Procedure Example of Low-Current-Consumption Read Mode.

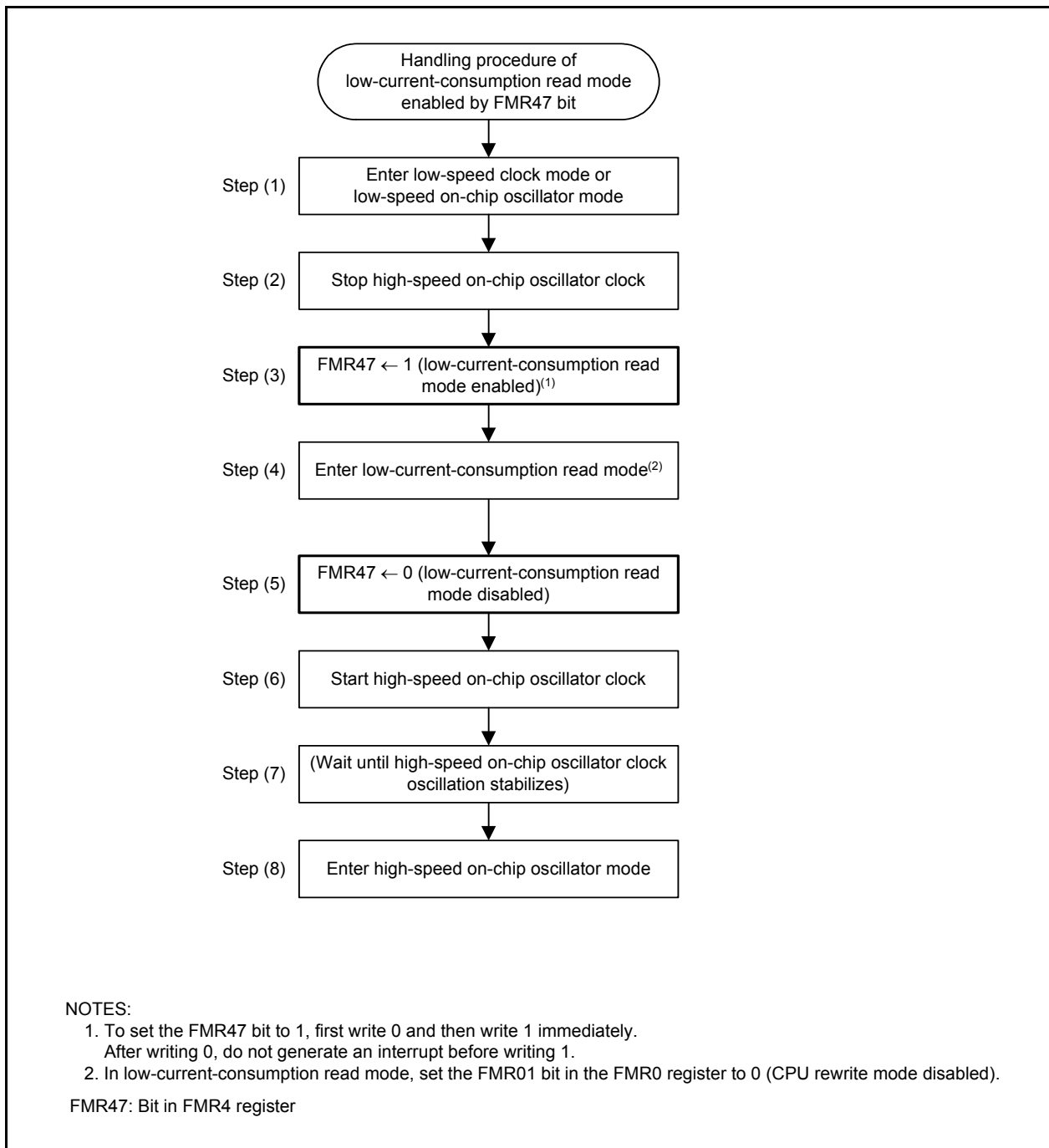


Figure 21.3 Handling Procedure Example of Low-Current-Consumption Read Mode

## 22. Electrical Characteristics

**Table 22.1 Absolute Maximum Ratings**

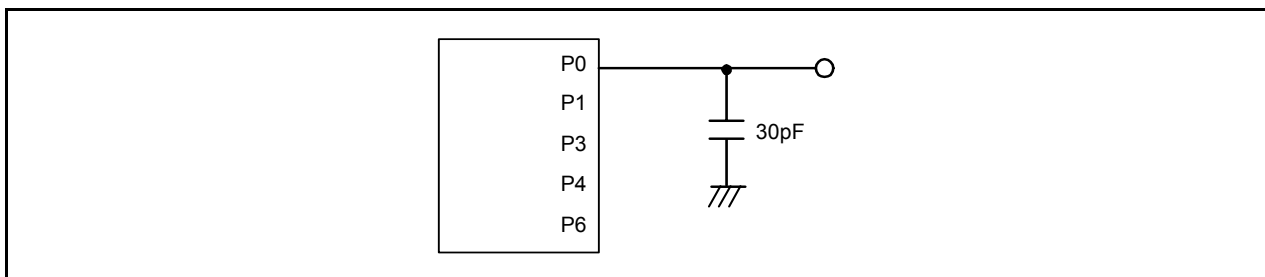
Symbol	Parameter	Condition	Rated Value	Unit
V <sub>CC</sub>	Supply voltage		-0.3 to 6.5	V
V <sub>I</sub>	Input voltage		-0.3 to V <sub>CC</sub> + 0.3	V
V <sub>O</sub>	Output voltage		-0.3 to V <sub>CC</sub> + 0.3	V
P <sub>d</sub>	Power dissipation	T <sub>opr</sub> = 25°C	500	mW
T <sub>opr</sub>	Operating ambient temperature		-20 to 85 (N version) / -40 to 85 (D version)	°C
T <sub>stg</sub>	Storage temperature		-65 to 150	°C

**Table 22.2 Recommended Operating Conditions**

Symbol	Parameter		Conditions	Standard			Unit
				Min.	Typ.	Max.	
V <sub>CC</sub>	Supply voltage			2.2	-	5.5	V
V <sub>SS</sub>	Supply voltage			-	0	-	V
V <sub>IH</sub>	Input "H" voltage			0.8 V <sub>CC</sub>	-	V <sub>CC</sub>	V
V <sub>IL</sub>	Input "L" voltage			0	-	0.2 V <sub>CC</sub>	V
I <sub>OH(sum)</sub>	Peak sum output "H" current	Sum of all pins I <sub>OH(peak)</sub>		-	-	-160	mA
I <sub>OH(sum)</sub>	Average sum output "H" current	Sum of all pins I <sub>OH(avg)</sub>		-	-	-80	mA
I <sub>OH(peak)</sub>	Peak output "H" current	All pins		-	-	-10	mA
I <sub>OH(avg)</sub>	Average output "H" current	All pins		-	-	-5	mA
I <sub>OL(sum)</sub>	Peak sum output "L" currents	Sum of all pins I <sub>OL(peak)</sub>		-	-	160	mA
I <sub>OL(sum)</sub>	Average sum output "L" currents	Sum of all pins I <sub>OL(avg)</sub>		-	-	80	mA
I <sub>OL(peak)</sub>	Peak output "L" currents	All pins		-	-	10	mA
I <sub>OL(avg)</sub>	Average output "L" current	All pins		-	-	5	mA
f <sub>(XCIN)</sub>	XCIN clock input oscillation frequency		2.2 V ≤ V <sub>CC</sub> ≤ 5.5 V	0	-	70	kHz
-	System clock	OCD2 = 0 XCIN clock selected	2.2 V ≤ V <sub>CC</sub> ≤ 5.5 V	0	-	70	kHz
-		OCD2 = 1 On-chip oscillator clock selected	HRA01 = 0 Low-speed on-chip oscillator selected	-	125	-	kHz
-			HRA01 = 1 High-speed on-chip oscillator selected 2.7 V ≤ V <sub>CC</sub> ≤ 5.5 V	-	-	8	MHz
-			HRA01 = 1 High-speed on-chip oscillator selected 2.2 V ≤ V <sub>CC</sub> ≤ 5.5 V	-	-	4	MHz

**NOTES:**

- V<sub>CC</sub> = 2.2 to 5.5 V at T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.
- The average output current indicates the average value of current measured during 100 ms.


**Figure 22.1 Ports P0, P1, P3, P4, and P6 Timing Measurement Circuit**

**Table 22.3 Flash Memory (Program ROM) Electrical Characteristics**

Symbol	Parameter	Conditions	Standard			Unit
			Min.	Typ.	Max.	
–	Program/erase endurance <sup>(2)</sup>		100 <sup>(3)</sup>	–	–	times
–	Byte program time		–	50	400	μs
–	Block erase time		–	0.4	9	s
–	Program, erase voltage		2.7	–	5.5	V
–	Read voltage		2.2	–	5.5	V
–	Program, erase temperature		0	–	60	°C
–	Data hold time <sup>(7)</sup>	Ambient temperature = 55°C	20	–	–	year

## NOTES:

1. V<sub>CC</sub> = 2.7 to 5.5 V at T<sub>opr</sub> = 0 to 60°C, unless otherwise specified.
2. Definition of programming/erasure endurance  
The programming and erasure endurance is defined on a per-block basis.  
If the programming and erasure endurance is n (n = 100 or 10,000), each block can be erased n times. For example, if 1,024 1-byte writes are performed to block A, a 1 Kbyte block, and then the block is erased, the programming/erasure endurance still stands at one.  
However, the same address must not be programmed more than once per erase operation (overwriting prohibited).
3. Endurance to guarantee all electrical characteristics after program and erase. (1 to Min. value can be guaranteed).
4. In a system that executes multiple programming operations, the actual erasure count can be reduced by writing to sequential addresses in turn so that as much of the block as possible is used up before performing an erase operation. For example, when programming groups of 16 bytes, the effective number of rewrites can be minimized by programming up to 128 groups before erasing them all in one operation. It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.
5. If an error occurs during block erase, attempt to execute the clear status register command, then execute the block erase command at least three times until the erase error does not occur.
6. Customers desiring program/erase failure rate information should contact their Renesas technical support representative.
7. The data hold time includes time that the power supply is off or the clock is not supplied.



**Table 22.4 Voltage Detection 0 Circuit Electrical Characteristics**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V <sub>det0</sub>	Voltage detection level		2.2	2.3	2.4	V
–	Voltage detection circuit self power consumption	VCA25 = 1, V <sub>CC</sub> = 5.0 V	–	0.9	–	μA
t <sub>d(E-A)</sub>	Waiting time until voltage detection circuit operation starts <sup>(2)</sup>		–	–	300	μs
V <sub>ccmin</sub>	MCU operating voltage minimum value		2.2	–	–	V

## NOTES:

1. The measurement condition is V<sub>CC</sub> = 2.2 to 5.5 V and T<sub>opr</sub> = –20 to 85°C (N version) / –40 to 85°C (D version).
2. Necessary time until the voltage detection circuit operates when setting to 1 again after setting the VCA25 bit in the VCA2 register to 0.

**Table 22.5 Voltage Detection 1 Circuit Electrical Characteristics**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V <sub>det1</sub>	Voltage detection level <sup>(4)</sup>		2.70	2.85	3.00	V
–	Voltage monitor 1 interrupt request generation time <sup>(2)</sup>		–	40	–	μs
–	Voltage detection circuit self power consumption	VCA26 = 1, V <sub>CC</sub> = 5.0 V	–	0.6	–	μA
t <sub>d(E-A)</sub>	Waiting time until voltage detection circuit operation starts <sup>(3)</sup>		–	–	100	μs

## NOTES:

1. The measurement condition is V<sub>CC</sub> = 2.2 to 5.5 V and T<sub>opr</sub> = –20 to 85°C (N version) / –40 to 85°C (D version).
2. Time until the voltage monitor 1 interrupt request is generated after the voltage passes V<sub>det1</sub>.
3. Necessary time until the voltage detection circuit operates when setting to 1 again after setting the VCA26 bit in the VCA2 register to 0.
4. This parameter shows the voltage detection level when the power supply drops.  
The voltage detection level when the power supply rises is higher than the voltage detection level when the power supply drops by approximately 0.1 V.

**Table 22.6 Voltage Detection 2 Circuit Electrical Characteristics**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V <sub>det2</sub>	Voltage detection level		3.3	3.6	3.9	V
–	Voltage monitor 2 interrupt request generation time <sup>(2)</sup>		–	40	–	μs
–	Voltage detection circuit self power consumption	VCA27 = 1, V <sub>CC</sub> = 5.0 V	–	0.6	–	μA
t <sub>d(E-A)</sub>	Waiting time until voltage detection circuit operation starts <sup>(3)</sup>		–	–	100	μs

## NOTES:

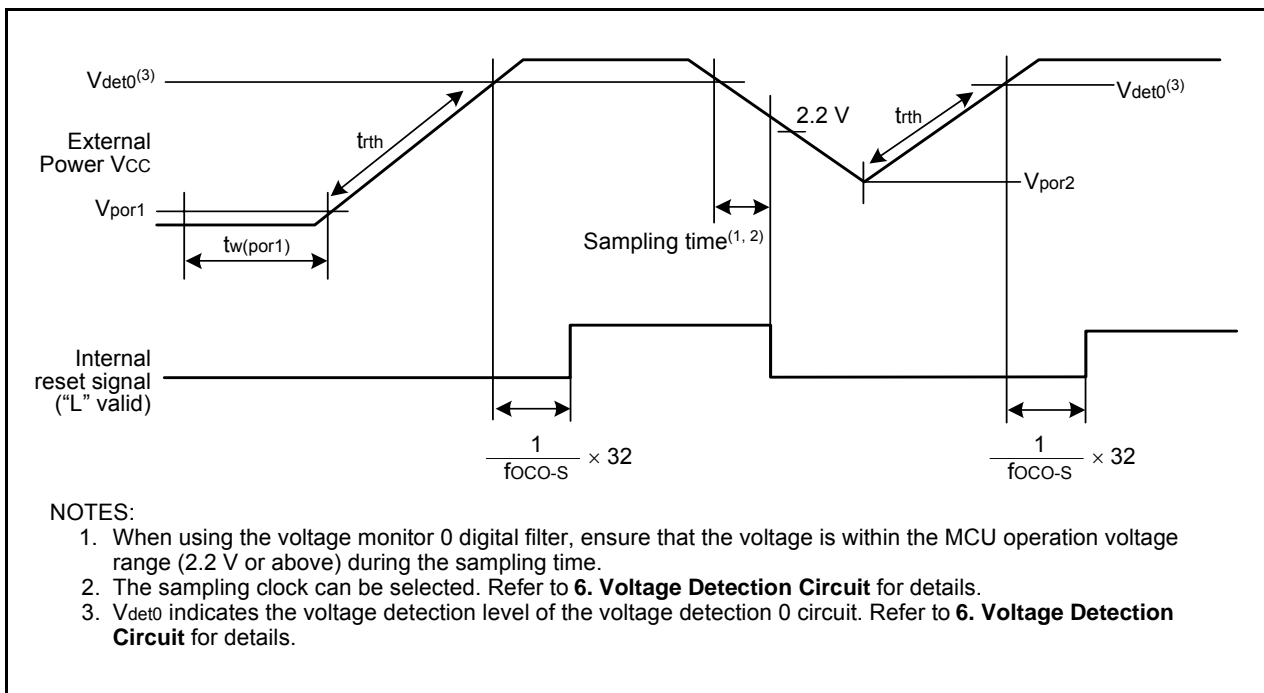
1. The measurement condition is V<sub>CC</sub> = 2.2 to 5.5 V and T<sub>opr</sub> = –20 to 85°C (N version) / –40 to 85°C (D version).
2. Time until the voltage monitor 2 interrupt request is generated after the voltage passes V<sub>det2</sub>.
3. Necessary time until the voltage detection circuit operates after setting to 1 again after setting the VCA27 bit in the VCA2 register to 0.

**Table 22.7 Power-on Reset Circuit, Voltage Monitor 0 Reset Electrical Characteristics<sup>(3)</sup>**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V <sub>por1</sub>	Power-on reset valid voltage <sup>(4)</sup>		–	–	0.1	V
V <sub>por2</sub>	Power-on reset or voltage monitor 0 reset valid voltage		0	–	V <sub>det0</sub>	V
t <sub>trth</sub>	External power V <sub>CC</sub> rise gradient <sup>(2)</sup>		20	–	–	mV/msec

**NOTES:**

- The measurement condition is T<sub>opr</sub> = –20 to 85°C (N version) / –40 to 85°C (D version), unless otherwise specified.
- This condition (external power V<sub>CC</sub> rise gradient) does not apply if V<sub>CC</sub> ≥ 1.0 V.
- To use the power-on reset function, enable voltage monitor 0 reset by setting the LVD0ON bit in the OFS register to 0, the VW0C0 and VW0C6 bits in the VW0C register to 1 respectively, and the VCA25 bit in the VCA2 register to 1.
- t<sub>w(por1)</sub> indicates the duration the external power V<sub>CC</sub> must be held below the effective voltage (V<sub>por1</sub>) to enable a power on reset. When turning on the power for the first time, maintain t<sub>w(por1)</sub> for 30 s or more if –20°C ≤ T<sub>opr</sub> ≤ 85°C, maintain t<sub>w(por1)</sub> for 3,000 s or more if –40°C ≤ T<sub>opr</sub> < –20°C.

**Figure 22.2 Reset Circuit Electrical Characteristics**

**Table 22.8 Comparator Electrical Characteristics**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
Vref	Internal reference voltage	V <sub>CC</sub> = 2.2 V to 5.5 V, T <sub>opr</sub> = 25°C	1.15	1.25	1.35	V
		V <sub>CC</sub> = 2.2 V to 5.5 V, T <sub>opr</sub> = -40 to 85°C	–	1.25	–	V
Vcref	External input reference voltage	V <sub>CC</sub> = 2.2 V to 4.0 V	0.5	–	V <sub>CC</sub> – 1.1	V
		V <sub>CC</sub> = 4.0 V to 5.5 V	0.5	–	V <sub>CC</sub> – 1.5	
Vcin	External comparison voltage input range		-0.3	–	V <sub>CC</sub> + 0.3	V
Vofs	Input offset voltage		–	20	120	mV
Tcrsp	Response time		–	4	–	μs

NOTE:

1. The measurement condition is T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.

**Table 22.9 High-speed On-Chip Oscillator Circuit Electrical Characteristics**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
fOCO-F	High-speed on-chip oscillator frequency temperature • supply voltage dependence	V <sub>CC</sub> = 4.75 V to 5.25 V T <sub>opr</sub> = 0 to 60°C <sup>(2)</sup>	7.76	8	8.24	MHz
		V <sub>CC</sub> = 2.7 V to 5.5 V T <sub>opr</sub> = -20 to 85°C <sup>(2)</sup>	7.68	8	8.32	MHz
		V <sub>CC</sub> = 2.7 V to 5.5 V T <sub>opr</sub> = -40 to 85°C <sup>(2)</sup>	7.44	8	8.32	MHz
		V <sub>CC</sub> = 2.2 V to 5.5 V T <sub>opr</sub> = -20 to 85°C <sup>(3)</sup>	7.04	8	8.96	MHz
		V <sub>CC</sub> = 2.2 V to 5.5 V T <sub>opr</sub> = -40 to 85°C <sup>(3)</sup>	6.8	8	9.2	MHz

NOTES:

1. The measurement condition is T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.
2. These standard values show when the HRA1 register is set to the value before shipment and the HRA2 register is set to 00h.
3. These standard values show when the correction value in the FRA6 register is written into the HRA1 register.

**Table 22.10 Low-speed On-Chip Oscillator Circuit Electrical Characteristics**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
fOCO-S	Low-speed on-chip oscillator frequency		30	125	250	kHz
–	Oscillation stability time		–	10	100	μs
–	Self power consumption at oscillation	V <sub>CC</sub> = 5.0 V, T <sub>opr</sub> = 25°C	–	15	–	μA

NOTE:

1. V<sub>CC</sub> = 2.2 to 5.5 V, T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.

**Table 22.11 Power Supply Circuit Timing Characteristics**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
t <sub>d</sub> (P-R)	Time for internal power supply stabilization during power-on <sup>(2)</sup>		1	–	2000	μs
t <sub>d</sub> (R-S)	STOP exit time <sup>(3)</sup>		–	–	150	μs

NOTES:

1. The measurement condition is V<sub>CC</sub> = 2.2 to 5.5 V and T<sub>opr</sub> = 25°C.
2. Waiting time until the internal power supply generation circuit stabilizes during power-on.
3. Time until system clock supply starts after the interrupt is acknowledged to exit stop mode.

**Table 22.12 Electrical Characteristics (1) [V<sub>CC</sub> = 5 V]**

Symbol	Parameter		Condition	Standard			Unit
				Min.	Typ.	Max.	
V <sub>OH</sub>	Output "H" voltage		I <sub>OH</sub> = -5 mA	V <sub>CC</sub> - 2.0	-	V <sub>CC</sub>	V
			I <sub>OH</sub> = -200 μA	V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
V <sub>OL</sub>	Output "L" voltage		I <sub>OL</sub> = 5 mA	-	-	2.0	V
			I <sub>OL</sub> = 200 μA	-	-	0.45	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	$\overline{\text{INT0}}, \overline{\text{INT1}}, \overline{\text{INT2}}, \overline{\text{INT4}},$ $\overline{\text{KI0}}, \overline{\text{KI1}}, \overline{\text{KI2}}, \overline{\text{KI3}},$ $\overline{\text{RXD0}}, \overline{\text{RXD2}},$ $\overline{\text{CLK0}}, \overline{\text{CLK2}}$		0.1	0.5	-	V
		$\overline{\text{RESET}}$		0.1	1.0	-	V
I <sub>IH</sub>	Input "H" current		V <sub>I</sub> = 5 V, V <sub>CC</sub> = 5 V	-	-	5.0	μA
I <sub>IL</sub>	Input "L" current		V <sub>I</sub> = 0 V, V <sub>CC</sub> = 5 V	-	-	-5.0	μA
R <sub>PULLUP</sub>	Pull-up resistance		V <sub>I</sub> = 0 V, V <sub>CC</sub> = 5 V	30	50	167	kΩ
R <sub>XCIN</sub>	Feedback resistance	XCIN		-	18	-	MΩ
V <sub>RAM</sub>	RAM hold voltage		During stop mode	2.0	-	-	V

## NOTE:

- V<sub>CC</sub> = 4.2 to 5.5 V at T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.

**Table 22.13 Electrical Characteristics (2) [Vcc = 5 V]**  
**(T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.)**

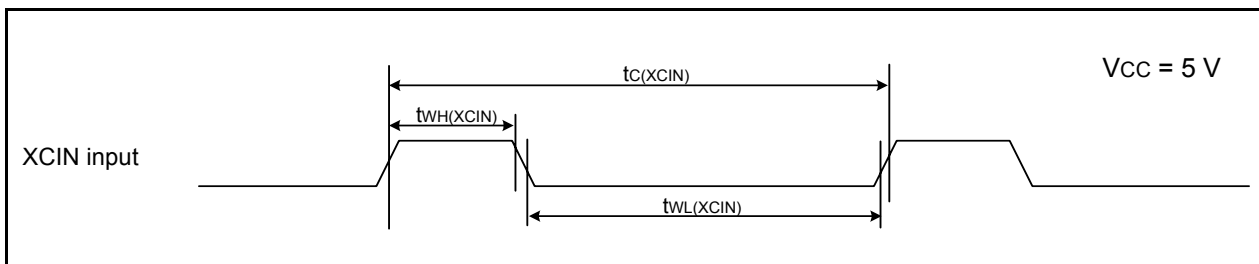
Symbol	Parameter	Condition	Standard			Unit	
			Min.	Typ.	Max.		
I <sub>cc</sub>	Power supply current (V <sub>cc</sub> = 3.3 to 5.5 V) Single-chip mode, output pins are open, other pins are V <sub>ss</sub>	High-speed on-chip oscillator mode	High-speed on-chip oscillator on = 8 MHz Low-speed on-chip oscillator on = 125 kHz No division	–	5	8	mA
			High-speed on-chip oscillator on = 8 MHz Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	2	–	mA
		Low-speed on-chip oscillator mode	High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8, FMR47 = 1	–	130	300	μA
			Low-speed clock mode	High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) FMR47 = 1	–	130	300
		High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) Program operation on RAM Flash memory off, FMSTP = 1		–	30	–	μA
		Wait mode	High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock operation VCA27 = VCA26 = VCA25 = 0 VCA20 = 1	–	25	75	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock off VCA27 = VCA26 = VCA25 = 0 VCA20 = 1	–	23	60	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (high drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 BGR trimming circuit disabled (BGRCR0 = 1)	–	4	–	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 BGR trimming circuit disabled (BGRCR0 = 1)	–	2.2	–	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (high drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 BGR trimming circuit enabled (BGRCR0 = 0)	–	8	–	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 BGR trimming circuit enabled (BGRCR0 = 0)	–	6	–	μA
			Stop mode	XCIN clock off, T <sub>opr</sub> = 25°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 BGR trimming circuit disabled (BGRCR0 = 1)	–	0.8	3
		XCIN clock off, T <sub>opr</sub> = 85°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 BGR trimming circuit disabled (BGRCR0 = 1)		–	1.2	–	μA
		XCIN clock off, T <sub>opr</sub> = 25°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 BGR trimming circuit enabled (BGRCR0 = 0)		–	5	8	μA
		XCIN clock off, T <sub>opr</sub> = 85°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 BGR trimming circuit enabled (BGRCR0 = 0)		–	5.5	–	μA

**Timing Requirements**

(Unless Otherwise Specified:  $V_{CC} = 5\text{ V}$ ,  $V_{SS} = 0\text{ V}$  at  $T_{opr} = 25^\circ\text{C}$ ) [ $V_{CC} = 5\text{ V}$ ]

**Table 22.14 XCIN Input**

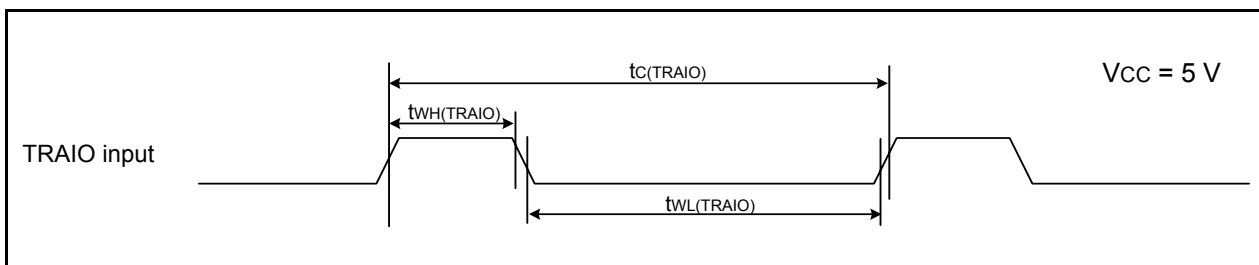
Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(XCIN)}$	XCIN input cycle time	14	–	$\mu\text{s}$
$t_{WH(XCIN)}$	XCIN input "H" width	7	–	$\mu\text{s}$
$t_{WL(XCIN)}$	XCIN input "L" width	7	–	$\mu\text{s}$



**Figure 22.3 XCIN Input Timing Diagram when  $V_{CC} = 5\text{ V}$**

**Table 22.15 TRAI0 Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TRAIO)}$	TRAIO input cycle time	100	–	ns
$t_{WH(TRAIO)}$	TRAIO input "H" width	40	–	ns
$t_{WL(TRAIO)}$	TRAIO input "L" width	40	–	ns

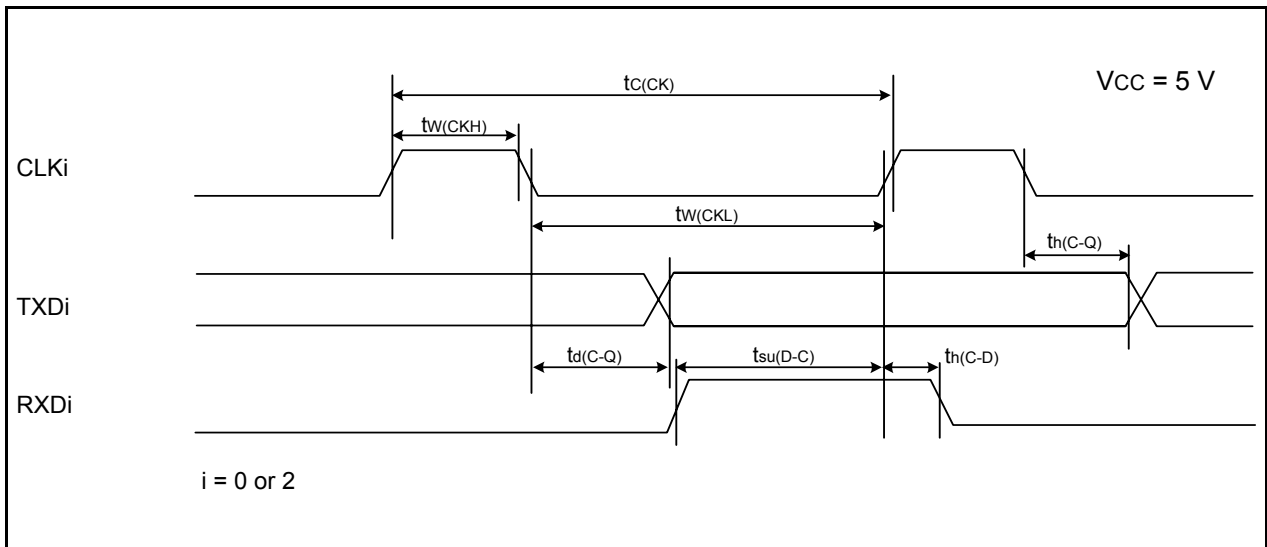


**Figure 22.4 TRAI0 Input Timing Diagram when  $V_{CC} = 5\text{ V}$**

**Table 22.16 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200	–	ns
$t_{w(CKH)}$	CLKi input “H” width	100	–	ns
$t_{w(CKL)}$	CLKi input “L” width	100	–	ns
$t_{d(C-Q)}$	TXDi output delay time	–	50	ns
$t_{h(C-Q)}$	TXDi hold time	0	–	ns
$t_{su(D-C)}$	RXDi input setup time	50	–	ns
$t_{h(C-D)}$	RXDi input hold time	90	–	ns

i = 0 or 2



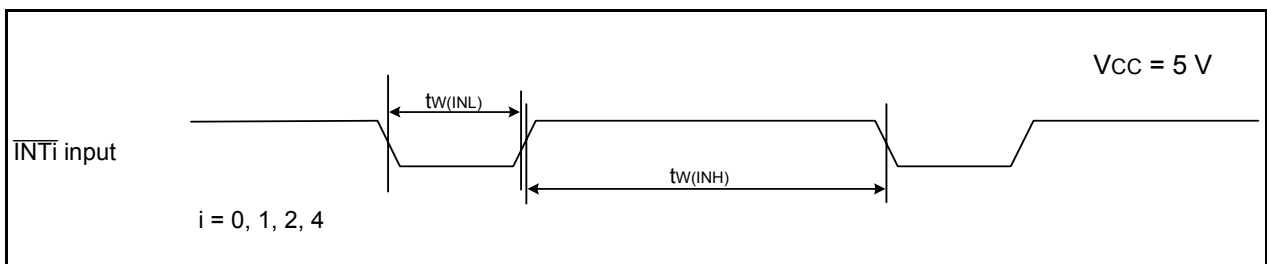
**Figure 22.5 Serial Interface Timing Diagram when Vcc = 5 V**

**Table 22.17 External Interrupt  $\overline{INTi}$  (i = 0, 1, 2, 4) Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input “H” width	250 <sup>(1)</sup>	–	ns
$t_{w(INL)}$	$\overline{INTi}$ input “L” width	250 <sup>(2)</sup>	–	ns

NOTES:

1. When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use an  $\overline{INTi}$  input HIGH width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.
2. When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use an  $\overline{INTi}$  input LOW width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.



**Figure 22.6 External Interrupt  $\overline{INTi}$  Input Timing Diagram when Vcc = 5 V**

**Table 22.18 Electrical Characteristics (3) [V<sub>CC</sub> = 3 V]**

Symbol	Parameter		Condition	Standard			Unit
				Min.	Typ.	Max.	
V <sub>OH</sub>	Output "H" voltage		I <sub>OH</sub> = -1 mA	V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
V <sub>OL</sub>	Output "L" voltage		I <sub>OL</sub> = 1 mA	-	-	0.5	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	INT0, INT1, INT2, INT4, KI0, KI1, KI2, KI3, RXD0, RXD2, CLK0, CLK2		0.1	0.3	-	V
		RESET		0.1	0.4	-	V
I <sub>IH</sub>	Input "H" current		V <sub>I</sub> = 3 V, V <sub>CC</sub> = 3 V	-	-	4.0	μA
I <sub>IL</sub>	Input "L" current		V <sub>I</sub> = 0 V, V <sub>CC</sub> = 3 V	-	-	-4.0	μA
R <sub>PULLUP</sub>	Pull-up resistance		V <sub>I</sub> = 0 V, V <sub>CC</sub> = 3 V	66	160	500	kΩ
R <sub>XCIN</sub>	Feedback resistance	XCIN		-	18	-	MΩ
V <sub>RAM</sub>	RAM hold voltage		During stop mode	1.8	-	-	V

## NOTE:

- V<sub>CC</sub> = 2.7 to 3.3 V at T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.



**Table 22.19 Electrical Characteristics (4) [V<sub>CC</sub> = 3 V]**  
**(T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.)**

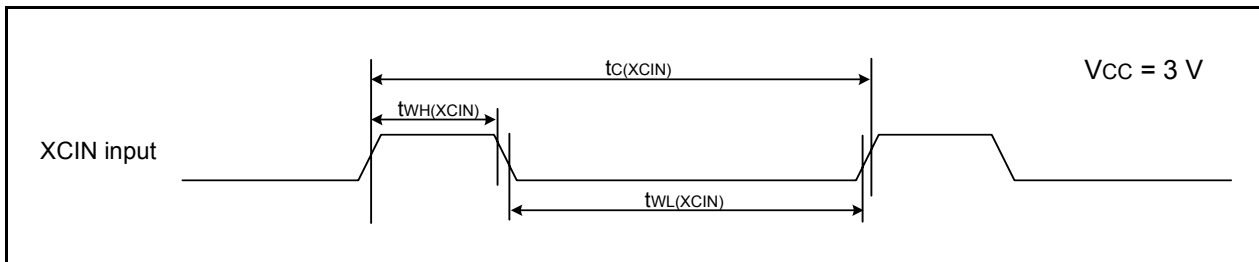
Symbol	Parameter	Condition	Standard			Unit	
			Min.	Typ.	Max.		
I <sub>CC</sub>	Power supply current (V <sub>CC</sub> = 2.7 to 3.3 V) Single-chip mode, output pins are open, other pins are V <sub>SS</sub>	High-speed on-chip oscillator mode	High-speed on-chip oscillator on = 8 MHz Low-speed on-chip oscillator on = 125 kHz No division	-	5	-	mA
			High-speed on-chip oscillator on = 8 MHz Low-speed on-chip oscillator on = 125 kHz Divide-by-8	-	2	-	mA
		Low-speed on-chip oscillator mode	High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8, FMR47 = 1	-	130	300	μA
			Low-speed clock mode	High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) FMR47 = 1	-	130	300
		High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) Program operation on RAM Flash memory off, FMSTP = 1		-	30	-	μA
		Wait mode	High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock operation VCA27 = VCA26 = VCA25 = 0 VCA20 = 1	-	25	70	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock off VCA27 = VCA26 = VCA25 = 0 VCA20 = 1	-	23	55	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (high drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 BGR trimming circuit disabled (BGRCR0 = 1)	-	3.8	-	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 BGR trimming circuit disabled (BGRCR0 = 1)	-	2	-	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (high drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 BGR trimming circuit enabled (BGRCR0 = 0)	-	8	-	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 BGR trimming circuit enabled (BGRCR0 = 0)	-	6	-	μA
			Stop mode	XCIN clock off, T <sub>opr</sub> = 25°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 BGR trimming circuit disabled (BGRCR0 = 1)	-	0.7	3
		XCIN clock off, T <sub>opr</sub> = 85°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 BGR trimming circuit disabled (BGRCR0 = 1)		-	1.1	-	μA
		XCIN clock off, T <sub>opr</sub> = 25°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 BGR trimming circuit enabled (BGRCR0 = 0)		-	5	7	μA
		XCIN clock off, T <sub>opr</sub> = 85°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 BGR trimming circuit enabled (BGRCR0 = 0)		-	5.5	-	μA

**Timing requirements**

(Unless Otherwise Specified:  $V_{CC} = 3\text{ V}$ ,  $V_{SS} = 0\text{ V}$  at  $T_{opr} = 25^\circ\text{C}$ ) [ $V_{CC} = 3\text{ V}$ ]

**Table 22.20 XCIN Input**

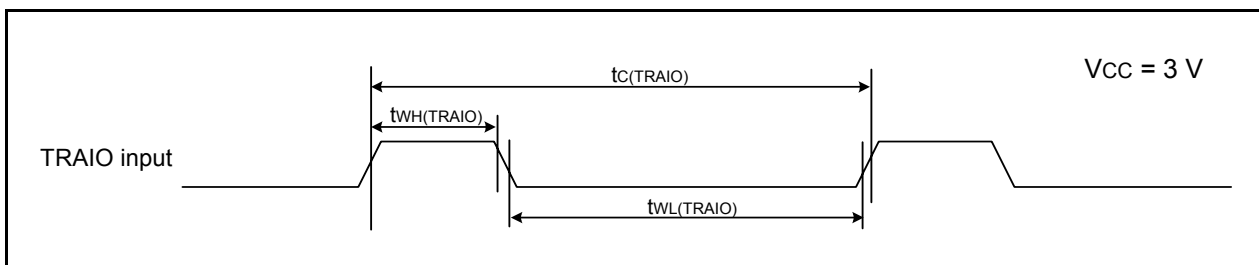
Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(XCIN)}$	XCIN input cycle time	14	–	$\mu\text{s}$
$t_{WH(XCIN)}$	XCIN input "H" width	7	–	$\mu\text{s}$
$t_{WL(XCIN)}$	XCIN input "L" width	7	–	$\mu\text{s}$



**Figure 22.7 XCIN Input Timing Diagram when  $V_{CC} = 3\text{ V}$**

**Table 22.21 TRAI0 Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TRAIO)}$	TRAIO input cycle time	300	–	ns
$t_{WH(TRAIO)}$	TRAIO input "H" width	120	–	ns
$t_{WL(TRAIO)}$	TRAIO input "L" width	120	–	ns

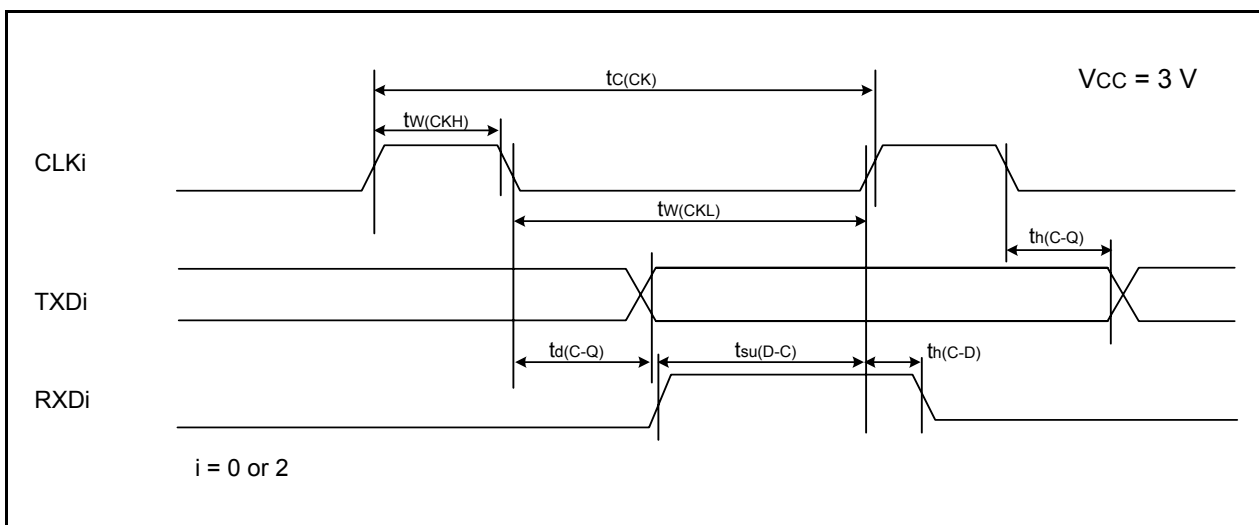


**Figure 22.8 TRAI0 Input Timing Diagram when  $V_{CC} = 3\text{ V}$**

**Table 22.22 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300	–	ns
$t_{w(CKH)}$	CLKi input “H” width	150	–	ns
$t_{w(CKL)}$	CLKi Input “L” width	150	–	ns
$t_{d(C-Q)}$	TXDi output delay time	–	80	ns
$t_{h(C-Q)}$	TXDi hold time	0	–	ns
$t_{su(D-C)}$	RXDi input setup time	70	–	ns
$t_{h(C-D)}$	RXDi input hold time	90	–	ns

i = 0 or 2



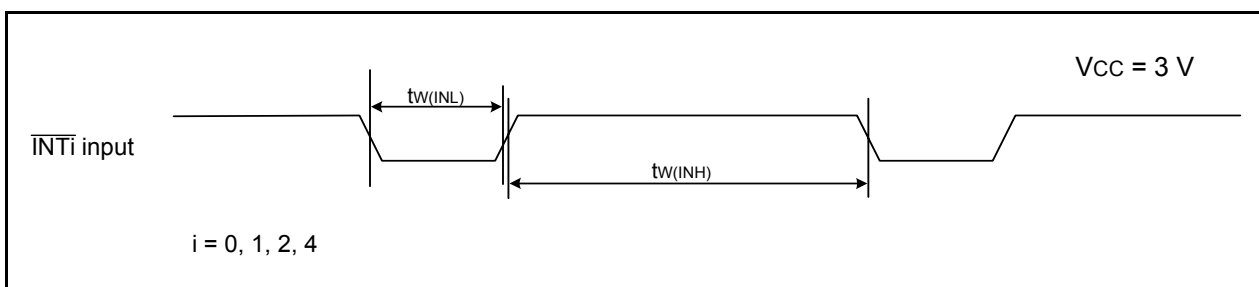
**Figure 22.9 Serial Interface Timing Diagram when Vcc = 3 V**

**Table 22.23 External Interrupt  $\overline{INTi}$  (i = 0, 1, 2, 4) Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input “H” width	380 <sup>(1)</sup>	–	ns
$t_{w(INL)}$	$\overline{INTi}$ input “L” width	380 <sup>(2)</sup>	–	ns

NOTES:

1. When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use an  $\overline{INTi}$  input HIGH width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.
2. When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use an  $\overline{INTi}$  input LOW width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.



**Figure 22.10 External Interrupt  $\overline{INTi}$  Input Timing Diagram when Vcc = 3 V**

**Table 22.24 Electrical Characteristics (5) [V<sub>CC</sub> = 2.2 V]**

Symbol	Parameter		Condition	Standard			Unit
				Min.	Typ.	Max.	
V <sub>OH</sub>	Output "H" voltage		I <sub>OH</sub> = -1 mA	V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
V <sub>OL</sub>	Output "L" voltage		I <sub>OL</sub> = 1 mA	-	-	0.5	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	INT0, INT1, INT2, INT4, K10, K11, K12, K13, RXD0, RXD2, CLK0, CLK2		0.05	0.3	-	V
		RESET		0.05	0.15	-	V
I <sub>IH</sub>	Input "H" current		V <sub>I</sub> = 2.2 V	-	-	4.0	μA
I <sub>IL</sub>	Input "L" current		V <sub>I</sub> = 0 V	-	-	-4.0	μA
R <sub>PULLUP</sub>	Pull-up resistance		V <sub>I</sub> = 0 V	100	200	600	kΩ
R <sub>FXCIN</sub>	Feedback resistance	XCIN		-	35	-	MΩ
V <sub>RAM</sub>	RAM hold voltage		During stop mode	1.8	-	-	V

## NOTE:

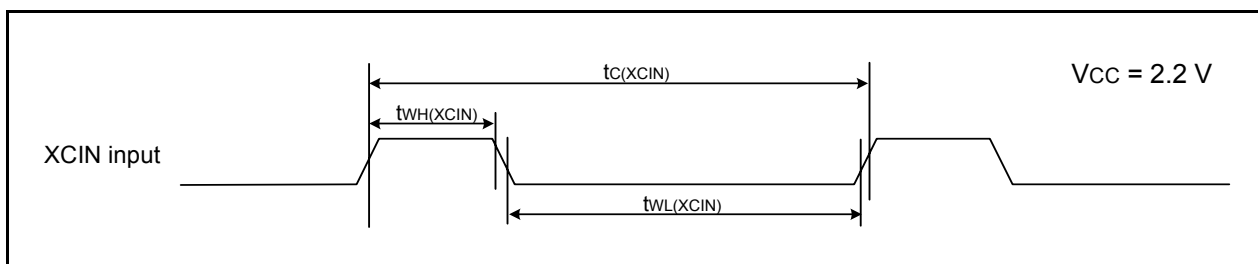
- V<sub>CC</sub> = 2.2 V at T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.

**Table 22.25 Electrical Characteristics (6) [V<sub>CC</sub> = 2.2 V]**  
**(T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.)**

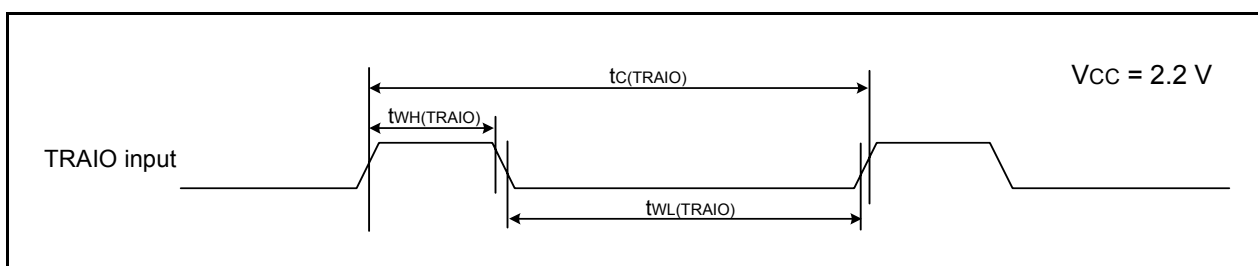
Symbol	Parameter	Condition	Standard			Unit	
			Min.	Typ.	Max.		
I <sub>CC</sub>	Power supply current (V <sub>CC</sub> = 2.2 to 2.7 V) Single-chip mode, output pins are open, other pins are V <sub>SS</sub>	High-speed on-chip oscillator mode	High-speed on-chip oscillator on = 4 MHz Low-speed on-chip oscillator on = 125 kHz No division	–	3.5	–	mA
			High-speed on-chip oscillator on = 4 MHz Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	1.5	–	mA
		Low-speed on-chip oscillator mode	High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8, FMR47 = 1	–	100	230	μA
			Low-speed clock mode	High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) FMR47 = 1	–	100	230
		High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) Program operation on RAM Flash memory off, FMSTP = 1		–	25	–	μA
		Wait mode	High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock operation VCA27 = VCA26 = VCA25 = 0 VCA20 = 1	–	22	60	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock off VCA27 = VCA26 = VCA25 = 0 VCA20 = 1	–	20	55	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (high drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 BGR trimming circuit disabled (BGRCR0 = 1)	–	3	–	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 BGR trimming circuit disabled (BGRCR0 = 1)	–	1.8	–	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (high drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 BGR trimming circuit enabled (BGRCR0 = 0)	–	7	–	μA
			High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 BGR trimming circuit enabled (BGRCR0 = 0)	–	6	–	μA
			Stop mode	XCIN clock off, T <sub>opr</sub> = 25°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 BGR trimming circuit disabled (BGRCR0 = 1)	–	0.7	3
		XCIN clock off, T <sub>opr</sub> = 85°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 BGR trimming circuit disabled (BGRCR0 = 1)		–	1.1	–	μA
		XCIN clock off, T <sub>opr</sub> = 25°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 BGR trimming circuit enabled (BGRCR0 = 0)		–	5	7	μA
		XCIN clock off, T <sub>opr</sub> = 85°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 BGR trimming circuit enabled (BGRCR0 = 0)		–	5.5	–	μA

**Timing requirements****(Unless Otherwise Specified:  $V_{CC} = 2.2\text{ V}$ ,  $V_{SS} = 0\text{ V}$  at  $T_{opr} = 25^{\circ}\text{C}$ ) [ $V_{CC} = 2.2\text{ V}$ ]****Table 22.26 XCIN Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(XCIN)}$	XCIN input cycle time	14	–	$\mu\text{s}$
$t_{WH(XCIN)}$	XCIN input "H" width	7	–	$\mu\text{s}$
$t_{WL(XCIN)}$	XCIN input "L" width	7	–	$\mu\text{s}$

**Figure 22.11 XCIN Input Timing Diagram when  $V_{CC} = 2.2\text{ V}$** **Table 22.27 TRAI0 Input**

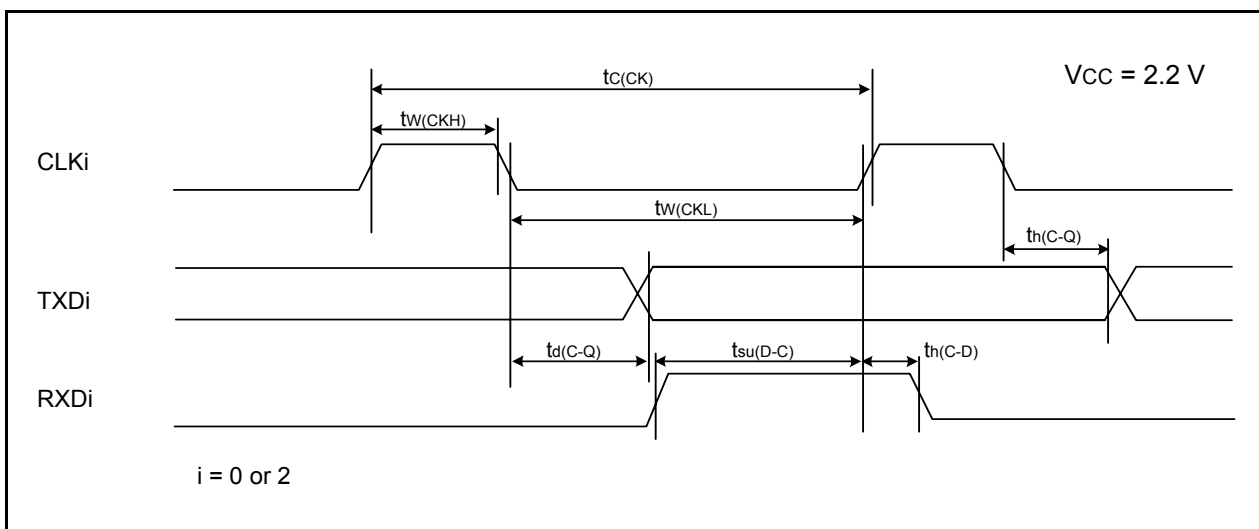
Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TRAIO)}$	TRAIO input cycle time	500	–	ns
$t_{WH(TRAIO)}$	TRAIO input "H" width	200	–	ns
$t_{WL(TRAIO)}$	TRAIO input "L" width	200	–	ns

**Figure 22.12 TRAI0 Input Timing Diagram when  $V_{CC} = 2.2\text{ V}$**

**Table 22.28 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	800	–	ns
$t_{w(CKH)}$	CLKi input “H” width	400	–	ns
$t_{w(CKL)}$	CLKi input “L” width	400	–	ns
$t_{d(C-Q)}$	TXDi output delay time	–	200	ns
$t_{h(C-Q)}$	TXDi hold time	0	–	ns
$t_{su(D-C)}$	RXDi input setup time	150	–	ns
$t_{h(C-D)}$	RXDi input hold time	90	–	ns

$i = 0$  or  $2$



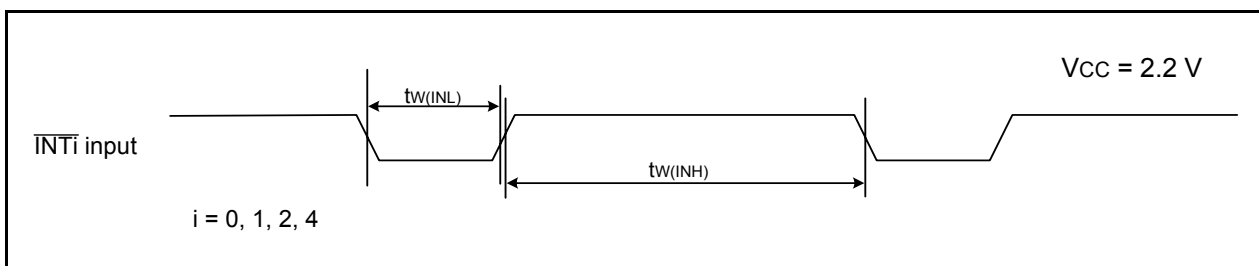
**Figure 22.13 Serial Interface Timing Diagram when  $V_{cc} = 2.2$  V**

**Table 22.29 External Interrupt  $\overline{INTi}$  ( $i = 0, 1, 2, 4$ ) Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input “H” width	1000 <sup>(1)</sup>	–	ns
$t_{w(INL)}$	$\overline{INTi}$ input “L” width	1000 <sup>(2)</sup>	–	ns

NOTES:

1. When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use an  $\overline{INTi}$  input HIGH width of either (1/digital filter clock frequency  $\times$  3) or the minimum value of standard, whichever is greater.
2. When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use an  $\overline{INTi}$  input LOW width of either (1/digital filter clock frequency  $\times$  3) or the minimum value of standard, whichever is greater.



**Figure 22.14 External Interrupt  $\overline{INTi}$  Input Timing Diagram when  $V_{cc} = 2.2$  V**

## 23. Usage Notes

### 23.1 Notes on I/O Ports

#### 23.1.1 Port P4\_3, P4\_4

Ports P4\_3 and P4\_4 are also used as the XCIN function and the XCOU function, respectively. During a reset period and after a reset release, these ports are set to the XCIN and XCOU functions. Pins P4\_3 and P4\_4 can be switched to the port functions by setting the CM04 bit in the CM0 register to 0 (ports P4\_3 and P4\_4) by a program.

To use ports P4\_3 and P4\_4 as ports, note the following:

- Port P4\_3

After a reset until the CM04 bit is set to 0 (ports P4\_3 and P4\_4) by a program, a typical 10 M $\Omega$  impedance is connected between the P4\_3 pin and the MCU power supply or GND. If the XCIN is set to intermediate-level input or left floating, a shoot-through current flows into the oscillation driver.

- Port P4\_4

Use port P4\_4 as an output port by setting the PD4\_4 bit in the PD4 register to 1 (output mode). After a reset until the CM04 bit is set to 0 (ports P4\_3 and P4\_4) by a program, the P4\_4 pin may output an intermediate potential of about 2.0 V.



## 23.2 Notes on Clock Generation Circuit

### 23.2.1 Stop Mode

When entering stop mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) and the CM10 bit in the CM1 register to 1 (stop mode). An instruction queue pre-reads 4 bytes from the instruction which sets the CM10 bit to 1 (stop mode) and the program stops.

Insert at least 4 NOP instructions following the JMP.B instruction after the instruction which sets the CM10 bit to 1.

- Program example to enter stop mode

```

BCLR    1,FMR0        ; CPU rewrite mode disabled
BSET    0,PRCR        ; Protect disabled
FSET    I             ; Enable interrupt
BSET    0,CM1         ; Stop mode
JMP.B   LABEL_001
LABEL_001 :
NOP
NOP
NOP
NOP

```

### 23.2.2 Wait Mode

When entering wait mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) and execute the WAIT instruction. An instruction queue pre-reads 4 bytes from the WAIT instruction and the program stops. Insert at least 4 NOP instructions after the WAIT instruction.

- Program example to execute the WAIT instruction

```

BCLR    1,FMR0        ; CPU rewrite mode disabled
FSET    I             ; Enable interrupt
WAIT                    ; Wait mode
NOP
NOP
NOP
NOP

```

### 23.2.3 Oscillation Circuit Constants

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

## 23.3 Notes on Interrupts

### 23.3.1 Reading Address 00000h

Do not read address 00000h by a program. When a maskable interrupt request is acknowledged, the CPU reads interrupt information (interrupt number and interrupt request level) from 00000h in the interrupt sequence. At this time, the acknowledged interrupt IR bit is set to 0.

If address 00000h is read by a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is set to 0. This may cause the interrupt to be canceled, or an unexpected interrupt to be generated.

### 23.3.2 SP Setting

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

### 23.3.3 External Interrupt and Key Input Interrupt

Either “L” level or an “H” level of width shown in the Electrical Characteristics is necessary for the signal input to pins  $\overline{\text{INT0}}$ ,  $\overline{\text{INT1}}$ ,  $\overline{\text{INT2}}$ ,  $\overline{\text{INT4}}$  and pins  $\overline{\text{KI0}}$  to  $\overline{\text{KI3}}$ , regardless of the CPU clock.

For details, refer to **Table 22.17** (VCC = 5V), **Table 22.23** (VCC = 3V), and **Table 22.29** (VCC = 2.2V) **External Interrupt INT<sub>i</sub> (i = 0, 1, 2, 4) Input**.

### 23.3.4 Changing Interrupt Sources

The IR bit in the interrupt control register may be set to 1 (interrupt requested) when the interrupt source changes. When using an interrupt, set the IR bit to 0 (no interrupt requested) after changing the interrupt source. In addition, changes of interrupt sources include all factors that change the interrupt sources assigned to individual software interrupt numbers, polarities, and timing. Therefore, if a mode change of a peripheral function involves interrupt sources, edge polarities, and timing, set the IR bit to 0 (no interrupt requested) after the change. Refer to the individual peripheral function for its related interrupts.

Figure 23.1 shows an Example of Procedure for Changing Interrupt Sources.

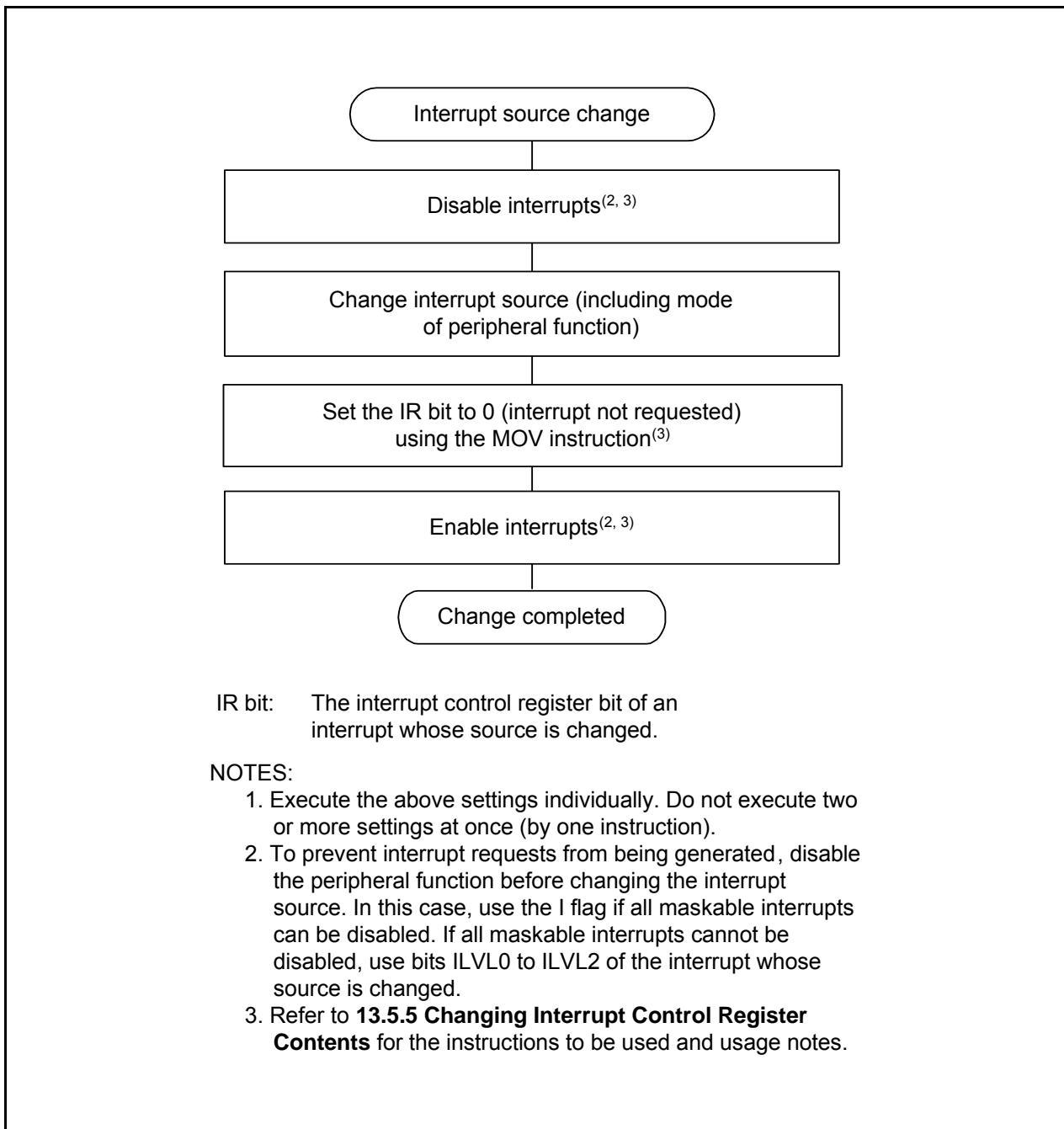


Figure 23.1 Example of Procedure for Changing Interrupt Sources

### 23.3.5 Changing Interrupt Control Register Contents

- (a) The contents of an interrupt control register can only be changed while no interrupt requests corresponding to that register are generated. If interrupt requests may be generated, disable interrupts before changing the interrupt control register contents.
- (b) When changing the contents of an interrupt control register after disabling interrupts, be careful to choose appropriate instructions.

#### Changing any bit other than IR bit

If an interrupt request corresponding to a register is generated while executing the instruction, the IR bit may not be set to 1 (interrupt requested), and the interrupt request may be ignored. If this causes a problem, use the following instructions to change the register: AND, OR, BCLR, BSET

#### Changing IR bit

If the IR bit is set to 0 (interrupt not requested), it may not be set to 1 depending on the instruction used. Therefore, use the MOV instruction to set the IR bit to 1.

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

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

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

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

#### Example 2: Use dummy read to delay FSET instruction

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

#### Example 3: Use POPC instruction to change I flag

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

## 23.4 Notes on ID Code Areas

### 23.4.1 Setting Example of ID Code Areas

As the ID code areas are allocated in the flash memory (not in the SFRs), they cannot be rewritten by executing an instruction. Write appropriate values when creating a program. The following shows a setting example.

- To set 55h in all of the ID code areas

```
.org 00FFDCH
.word dummy | (55000000h) ; UND
.word dummy | (55000000h) ; INTO
.word dummy ; BREAK
.word dummy | (55000000h) ; ADDRESS MATCH
.word dummy | (55000000h) ; SET SINGLE STEP
.word dummy | (55000000h) ; WDT
.word dummy | (55000000h) ; ADDRESS BREAK
.word dummy | (55000000h) ; RESERVE
```

(Programming formats vary depending on the compiler. Check the compiler manual.)

## 23.5 Notes on Option Function Select Area

### 23.5.1 Setting Example of Option Function Select Area

As the option function select area is allocated in the flash memory (not in the SFRs), they cannot be rewritten by executing an instruction. Write appropriate values when creating a program. The following shows a setting example.

- To set FFh in the OFS register

```
.org 00FFFCH
```

```
.lword reset | (0FF00000h) ; RESET
```

(Programming formats vary depending on the compiler. Check the compiler manual.)

## 23.6 Notes on Timers

### 23.6.1 Notes on Timer RA

- Timer RA stops counting after a reset. Set the values in the timer RA and timer RA prescalers before the count starts.
- Even if the prescaler and timer RA are read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- In pulse period measurement mode, bits TEDGF and TUNDF in the TRACR register can be set to 0 by writing 0 to these bits by a program. However, these bits remain unchanged if 1 is written. When using the READ-MODIFY-WRITE instruction for the TRACR register, the TEDGF or TUNDF bit may be set to 0 although these bits are set to 1 while the instruction is being executed. In this case, write 1 to the TEDGF or TUNDF bit which is not supposed to be set to 0 with the MOV instruction.
- When changing to pulse period measurement mode from another mode, the contents of bits TEDGF and TUNDF are undefined. Write 0 to bits TEDGF and TUNDF before the count starts.
- The TEDGF bit may be set to 1 by the first timer RA prescaler underflow generated after the count starts.
- When using the pulse period measurement mode, leave two or more periods of the timer RA prescaler immediately after the count starts, then set the TEDGF bit to 0.
- The TCSTF bit retains 0 (count stops) for 0 to 1 cycle of the count source after setting the TSTART bit to 1 (count starts) while the count is stopped.

During this time, do not access registers associated with timer RA<sup>(1)</sup> other than the TCSTF bit. Timer RA starts counting at the first valid edge of the count source after The TCSTF bit is set to 1 (during count).

The TCSTF bit remains 1 for 0 to 1 cycle of the count source after setting the TSTART bit to 0 (count stops) while the count is in progress. Timer RA counting is stopped when the TCSTF bit is set to 0.

During this time, do not access registers associated with timer RA<sup>(1)</sup> other than the TCSTF bit.

NOTE:

1. Registers associated with timer RA: TRACR, TRAIOC, TRAMR, TRAPRE, and TRA.

- When the TRAPRE register is continuously written during count operation (TCSTF bit is set to 1), allow three or more cycles of the count source clock for each write interval.
- When the TRA register is continuously written during count operation (TCSTF bit is set to 1), allow three or more cycles of the prescaler underflow for each write interval.

## 23.6.2 Notes on Timer RB

- Timer RB stops counting after a reset. Set the values in the timer RB and timer RB prescalers before the count starts.
- Even if the prescaler and timer RB is read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- In programmable one-shot generation mode and programmable wait one-shot generation mode, when setting the TSTART bit in the TRBCR register to 0, 0 (stops counting) or setting the TOSSP bit in the TRBOCR register to 1 (stops one-shot), the timer reloads the value of reload register and stops. Therefore, in programmable one-shot generation mode and programmable wait one-shot generation mode, read the timer count value before the timer stops.
- The TCSTF bit remains 0 (count stops) for 1 to 2 cycles of the count source after setting the TSTART bit to 1 (count starts) while the count is stopped.

During this time, do not access registers associated with timer RB<sup>(1)</sup> other than the TCSTF bit. Timer RB starts counting at the first valid edge of the count source after the TCSTF bit is set to 1 (during count).

The TCSTF bit remains 1 for 1 to 2 cycles of the count source after setting the TSTART bit to 0 (count stops) while the count is in progress. Timer RB counting is stopped when the TCSTF bit is set to 0.

During this time, do not access registers associated with timer RB<sup>(1)</sup> other than the TCSTF bit.

### NOTE:

1. Registers associated with timer RB: TRBCR, TRBOCR, TRBIOC, TRBMR, TRBPRES, TRBSC, and TRBPR.

- If the TSTOP bit in the TRBCR register is set to 1 during timer operation, timer RB stops immediately.
- If 1 is written to the TOSST or TOSSP bit in the TRBOCR register, the value of the TOSSTF bit changes after one or two cycles of the count source have elapsed. If the TOSSP bit is written to 1 during the period between when the TOSST bit is written to 1 and when the TOSSTF bit is set to 1, the TOSSTF bit may be set to either 0 or 1 depending on the content state. Likewise, if the TOSST bit is written to 1 during the period between when the TOSSP bit is written to 1 and when the TOSSTF bit is set to 0, the TOSSTF bit may be set to either 0 or 1.

### 23.6.2.1 Timer mode

The following workaround should be performed in timer mode.

To write to registers TRBPRES and TRBPR during count operation (TCSTF bit is set to 1), note the following points:

- When the TRBPRES register is written continuously, allow three or more cycles of the count source for each write interval.
- When the TRBPR register is written continuously, allow three or more cycles of the prescaler underflow for each write interval.



### 23.6.2.2 Programmable waveform generation mode

The following three workarounds should be performed in programmable waveform generation mode.

- (1) To write to registers TRBPRE and TRBPR during count operation (TCSTF bit is set to 1), note the following points:
  - When the TRBPRE register is written continuously, allow three or more cycles of the count source for each write interval.
  - When the TRBPR register is written continuously, allow three or more cycles of the prescaler underflow for each write interval.
- (2) To change registers TRBPRE and TRBPR during count operation (TCSTF bit is set to 1), synchronize the TRBO output cycle using a timer RB interrupt, etc. This operation should be performed only once in the same output cycle. Also, make sure that writing to the TRBPR register does not occur during period A shown in Figures 23.2 and 23.3.

The following shows the detailed workaround examples.

- Workaround example (a):  
As shown in Figure 23.2, write to registers TRBSC and TRBPR in the timer RB interrupt routine. These write operations must be completed by the beginning of period A.

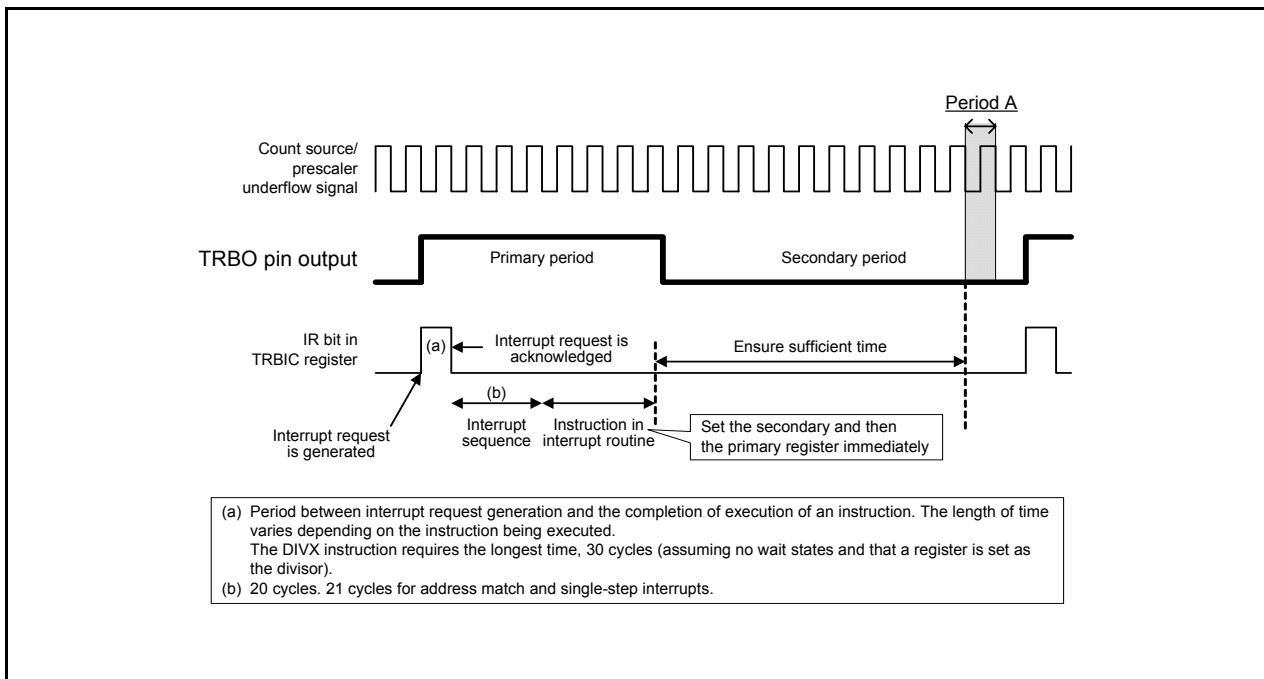
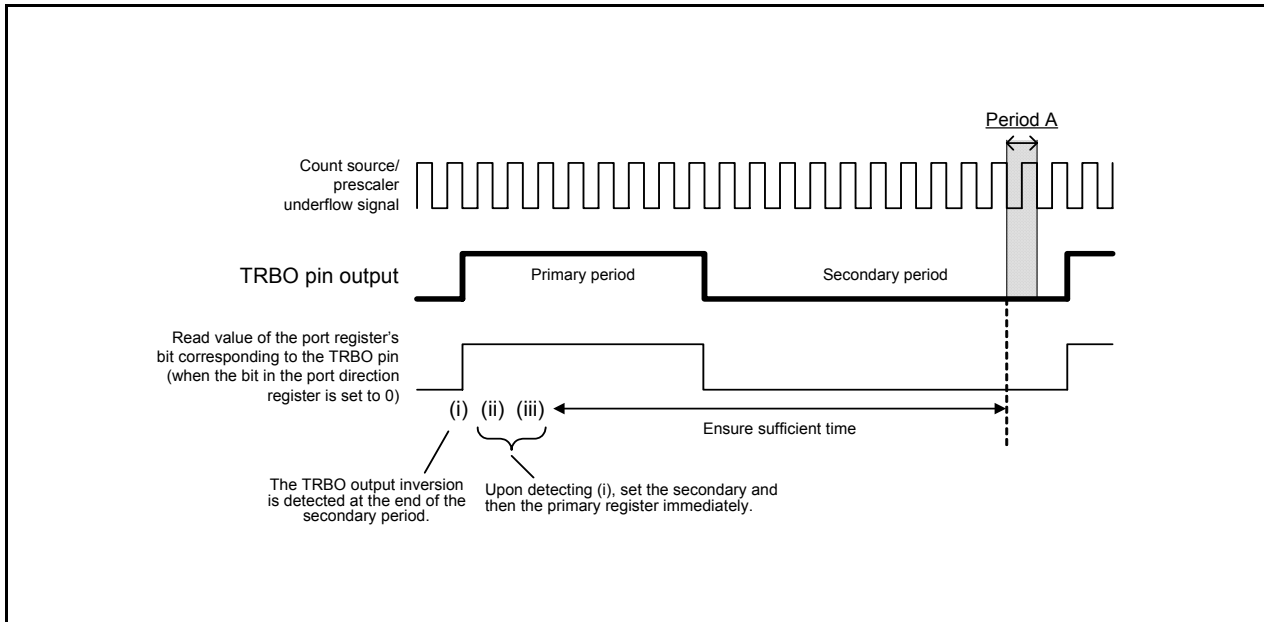


Figure 23.2 Workaround Example (a) When Timer RB interrupt is Used

- Workaround example (b):

As shown in Figure 23.3 detect the start of the primary period by the TRBO pin output level and write to registers TRBSC and TRBPR. These write operations must be completed by the beginning of period A.

If the port register's bit value is read after the port direction register's bit corresponding to the TRBO pin is set to 0 (input mode), the read value indicates the TRBO pin output value.



**Figure 23.3 Workaround Example (b) When TRBO Pin Output Value is Read**

- (3) To stop the timer counting in the primary period, use the TSTOP bit in the TRBCR register. In this case, registers TRBPRES and TRBPR are initialized and their values are set to the values after reset.

### 23.6.2.3 Programmable one-shot generation mode

The following two workarounds should be performed in programmable one-shot generation mode.

- (1) To write to registers TRBPRES and TRBPR during count operation (TCSTF bit is set to 1), note the following points:
  - When the TRBPRES register is written continuously during count operation (TCSTF bit is set to 1), allow three or more cycles of the count source for each write interval.
  - When the TRBPR register is written continuously during count operation (TCSTF bit is set to 1), allow three or more cycles of the prescaler underflow for each write interval.
- (2) Do not set both the TRBPRES and TRBPR registers to 00h.

### 23.6.2.4 Programmable wait one-shot generation mode

The following three workarounds should be performed in programmable wait one-shot generation mode.

- (1) To write to registers TRBPRES and TRBPR during count operation (TCSTF bit is set to 1), note the following points:
  - When the TRBPRES register is written continuously, allow three or more cycles of the count source for each write interval.
  - When the TRBPR register is written continuously, allow three or more cycles of the prescaler underflow for each write interval.
- (2) Do not set both the TRBPRES and TRBPR registers to 00h.
- (3) Set registers TRBSC and TRBPR using the following procedure.
  - (a) To use “ $\overline{\text{INT0}}$  pin one-shot trigger enabled” as the count start condition  
Set the TRBSC register and then the TRBPR register. At this time, after writing to the TRBPR register, allow an interval of 0.5 or more cycles of the count source before trigger input from the  $\overline{\text{INT0}}$  pin.
  - (b) To use “writing 1 to TOSST bit” as the start condition  
Set the TRBSC register, the TRBPR register, and then TOSST bit. At this time, after writing to the TRBPR register, allow an interval of 0.5 or more cycles of the count source before writing to the TOSST bit.

### 23.6.3 Notes on Timer RE

#### 23.6.3.1 Starting and Stopping Count

Timer RE has the TSTART bit for instructing the count to start or stop, and the TCSTF bit, which indicates count start or stop. Bits TSTART and TCSTF are in the TRECRC1 register.

Timer RE starts counting and the TCSTF bit is set to 1 (count starts) when the TSTART bit is set to 1 (count starts). It takes up to 2 cycles of the count source until the TCSTF bit is set to 1 after setting the TSTART bit to 1. During this time, do not access registers associated with timer RE<sup>(1)</sup> other than the TCSTF bit.

Also, timer RE stops counting when setting the TSTART bit to 0 (count stops) and the TCSTF bit is set to 0 (count stops). It takes the time for up to 2 cycles of the count source until the TCSTF bit is set to 0 after setting the TSTART bit to 0. During this time, do not access registers associated with timer RE other than the TCSTF bit.

NOTE:

1. Registers associated with timer RE: TRESEC, TREMIN, TREHR, TREWK, TRECRC1, TRECRC2, TRECSR, and TREOPR.

#### 23.6.3.2 Register Setting

Write to the following registers or bits when timer RE is stopped.

- Registers TRESEC, TREMIN, TREHR, TREWK, and TRECRC2
- Bits H12\_H24, PM, and INT in TRECRC1 register
- Bits RCS0 to RCS3 in TRECSR register

Timer RE is stopped when bits TSTART and TCSTF in the TRECRC1 register are set to 0 (timer RE stopped).

Also, set all above-mentioned registers and bits (immediately before timer RE count starts) before setting the TRECRC2 register.

Figure 23.4 shows a Setting Example in Real-Time Clock Mode.

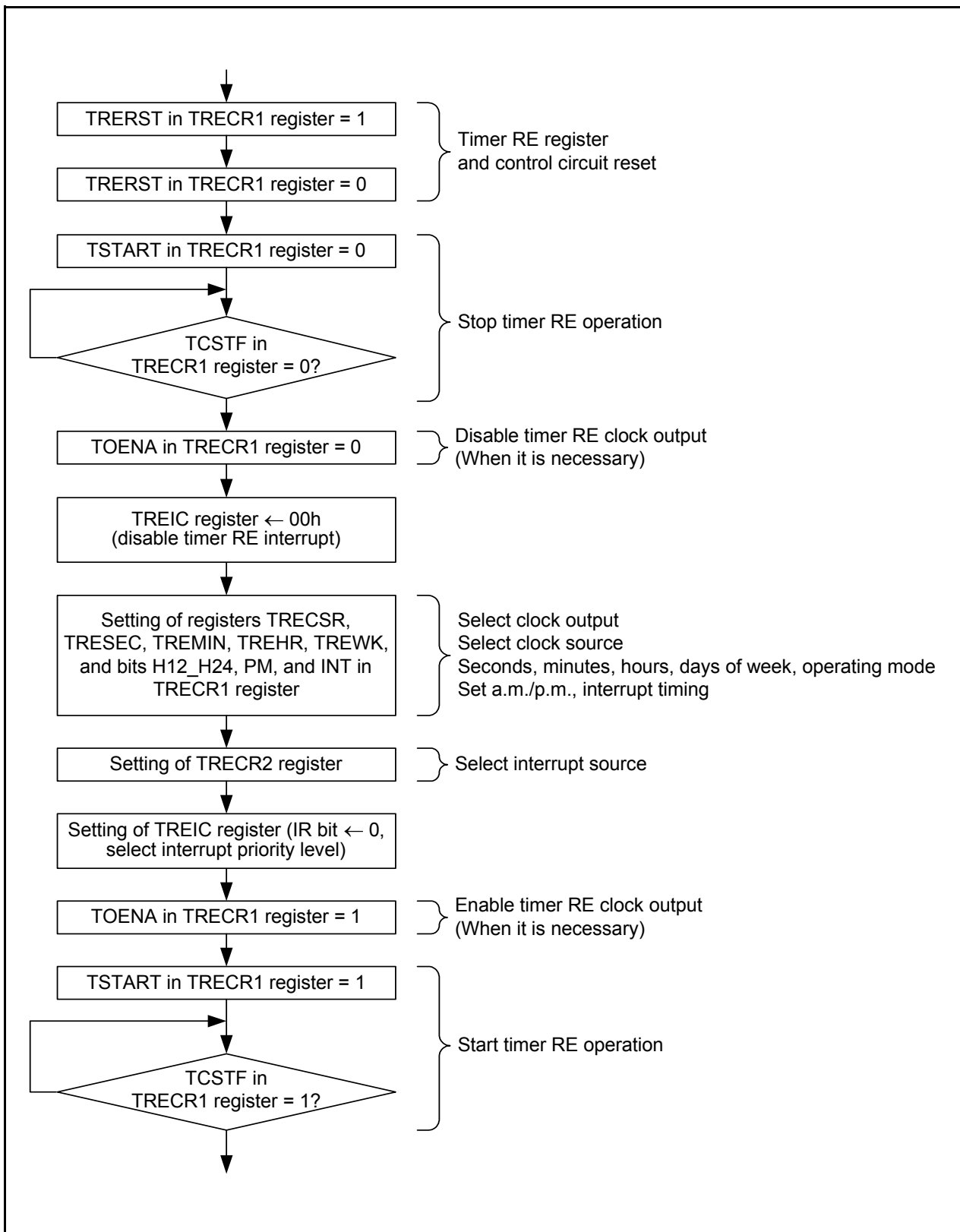


Figure 23.4 Setting Example in Real-Time Clock Mode

### 23.6.3.3 Time Reading Procedure of Real-Time Clock Mode

In real-time clock mode, read registers TRESEC, TREMIN, TREHR, and TREWK when time data is updated and read the PM bit in the TRECR1 register when the BSY bit is set to 0 (not while data is updated).

Also, when reading several registers, an incorrect time will be read if data is updated before another register is read after reading any register.

In order to prevent this, use the reading procedure shown below.

- Using an interrupt

Read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECR1 register in the timer RE interrupt routine.

- Monitoring with a program 1

Monitor the IR bit in the TREIC register with a program and read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECR1 register after the IR bit in the TREIC register is set to 1 (timer RE interrupt request generated).

- Monitoring with a program 2

- (1) Monitor the BSY bit.

- (2) Monitor until the BSY bit is set to 0 after the BSY bit is set to 1 (approximately 62.5 ms while the BSY bit is set to 1).

- (3) Read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECR1 register after the BSY bit is set to 0.

- Using read results if they are the same value twice

- (1) Read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECR1 register.

- (2) Read the same register as (1) and compare the contents.

- (3) Recognize as the correct value if the contents match. If the contents do not match, repeat until the read contents match with the previous contents.

Also, when reading several registers, read them as continuously as possible.

### 23.6.4 Notes on Timer RF

- Access registers TRF, TRFM0, and TRFM1 in 16-bit units.

Example of reading timer RF:

```
MOV.W    0290H,R0    ; Read out timer RF
```

- In input capture mode, a capture interrupt request is generated by inputting an edge selected by bits TRFC03 and TRFC04 in the TRFCR0 register even when the TSTART bit in the TRFCR0 register is set to 0 (count stops).

## 23.7 Notes on Serial Interface

- When reading data from the UiRB (i = 0 or 2) register either in the clock synchronous serial I/O mode or in the clock asynchronous serial I/O mode. Ensure the data is read in 16-bit units. When the high-order byte of the UiRB register is read, bits PER and FER in the UiRB register and the RI bit in the UiC1 register are set to 0. To check receive errors, read the UiRB register and then use the read data.

Example (when reading receive buffer register):

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

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

Example (when reading transmit buffer register):

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



### 23.8 Notes on Hardware LIN

For the time-out processing of the header and response fields, use another timer to measure the duration of time with a Synch Break detection interrupt as the starting point.

## 23.9 Notes on Flash Memory

### 23.9.1 CPU Rewrite Mode

#### 23.9.1.1 Operating Speed

Before entering CPU rewrite mode (EW0 mode), select 5 MHz or below for the CPU clock using the CM06 bit in the CM0 register and bits CM16 to CM17 in the CM1 register. This does not apply to EW1 mode.

#### 23.9.1.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because they reference data in the flash memory: UND, INTO, and BRK.

#### 23.9.1.3 Non-Maskable Interrupts

- EW0 Mode

Once a watchdog timer, voltage monitor1, voltage monitor 2, comparator 1, or comparator 2 interrupt request is acknowledged, auto-erasure or auto-programming is forcibly stopped immediately and the flash memory is reset. Interrupt handling starts after a fixed period and the flash memory restarts.

As the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be readable. Execute auto-erasure again and ensure it completes normally.

The watchdog timer does not stop during command operation, so that interrupt requests may be generated. Initialize the watchdog timer regularly.

Do not use the address match interrupt while a command is being executed because the vector of the address match interrupt is allocated in ROM.

Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

- EW1 Mode

Once a watchdog timer, voltage monitor1, voltage monitor 2, comparator 1, or comparator 2 interrupt request is acknowledged, auto-erasure or auto-programming is forcibly stopped immediately and the flash memory is reset. Interrupt handling starts after a fixed period and the flash memory restarts.

As the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be readable. Execute auto-erasure again and ensure it completes normally.

The watchdog timer does not stop even during command operation, so that interrupt requests may be generated. Initialize the watchdog timer by using the erase-suspend function.

Do not use the address match interrupt while a command is being executed because the vector of the address match interrupt is allocated in ROM.

Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

#### **23.9.1.4 How to Access**

Write 0 before writing 1 when setting Bits FMR01, FMR02 in the FMR0 register, or FMR11 bit in the FMR1 register to 1. Do not generate an interrupt between writing 0 and 1.

#### **23.9.1.5 Rewriting User ROM Area**

In EW0 Mode, if the supply voltage drops while rewriting any block in which a rewrite control program is stored, it may not be possible to rewrite the flash memory because the rewrite control program cannot be rewritten correctly. In this case, use standard serial I/O mode.

#### **23.9.1.6 Program**

Do not write additions to the already programmed address.

#### **23.9.1.7 Program and Erase Voltage for Flash Memory**

To perform programming and erasure, use  $VCC = 2.7\text{ V}$  to  $5.5\text{ V}$  as the supply voltage. Do not perform programming and erasure at less than  $2.7\text{ V}$ .

## 23.10 Notes on Noise

### 23.10.1 Inserting a Bypass Capacitor between VCC and VSS Pins as a Countermeasure against Noise and Latch-up

Connect a bypass capacitor (at least 0.1  $\mu\text{F}$ ) using the shortest and thickest wire possible.

### 23.10.2 Countermeasures against Noise Error of Port Control Registers

During rigorous noise testing or the like, external noise (mainly power supply system noise) can exceed the capacity of the MCU's internal noise control circuitry. In such cases the contents of the port related registers may be changed.

As a firmware countermeasure, it is recommended that the port registers, port direction registers, and pull-up control registers be reset periodically. However, examine the control processing fully before introducing the reset routine as conflicts may be created between the reset routine and interrupt routines.

## 24. Notes for On-Chip Debugger

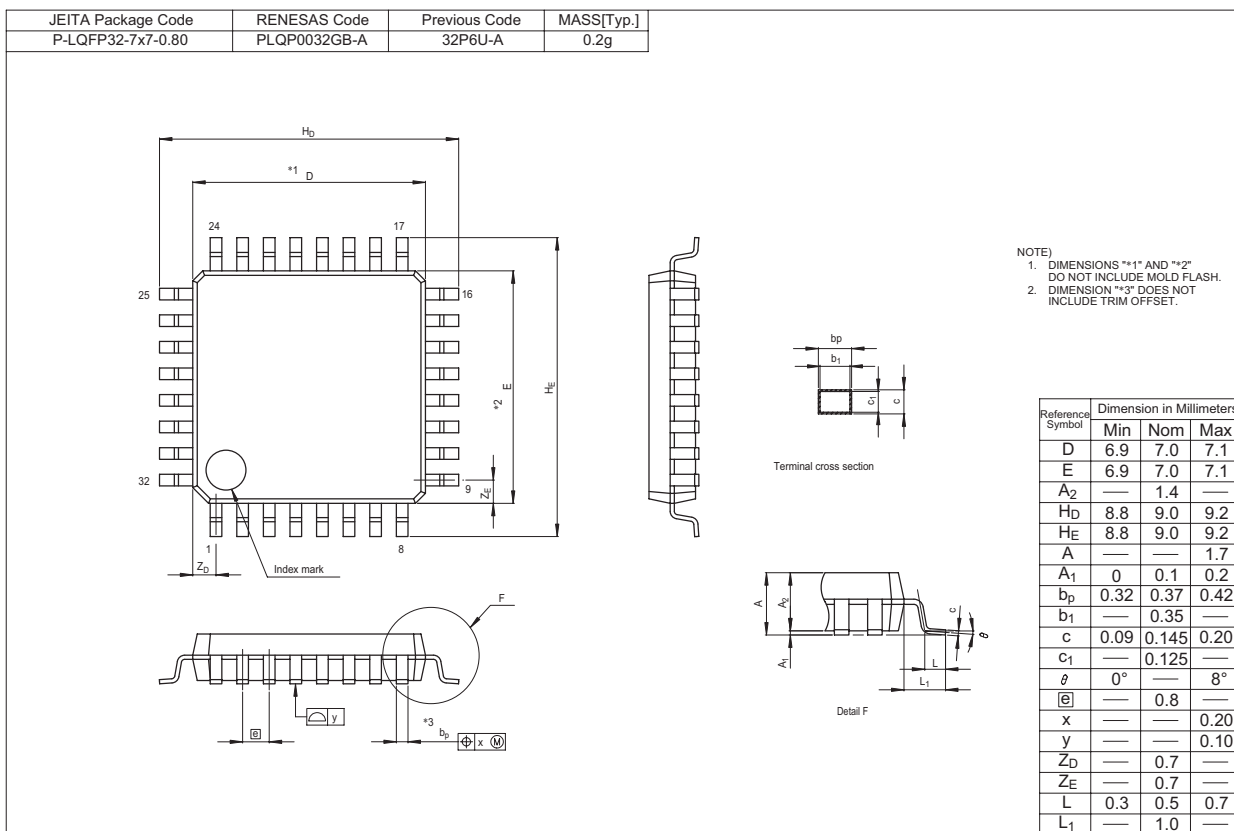
When using the on-chip debugger to develop and debug programs for the R8C/2G Group take note of the following.

- (1) Some of the user flash memory and RAM areas are used by the on-ship debugger. These areas cannot be accessed by the user.  
Refer to the on-chip debugger manual for which areas are used.
- (2) Do not set the address match interrupt (registers AIER, RMAD0, and RMAD1 and fixed vector tables) in a user system.
- (3) Do not use the BRK instruction in a user system.
- (4) Debugging is available under the condition of supply voltage  $VCC = 2.7$  to  $5.5$  V. Debugging with the on-chip debugger under less than  $2.7$  V is not allowed.

Connecting and using the on-chip debugger has some special restrictions. Refer to the on-chip debugger manual for details.

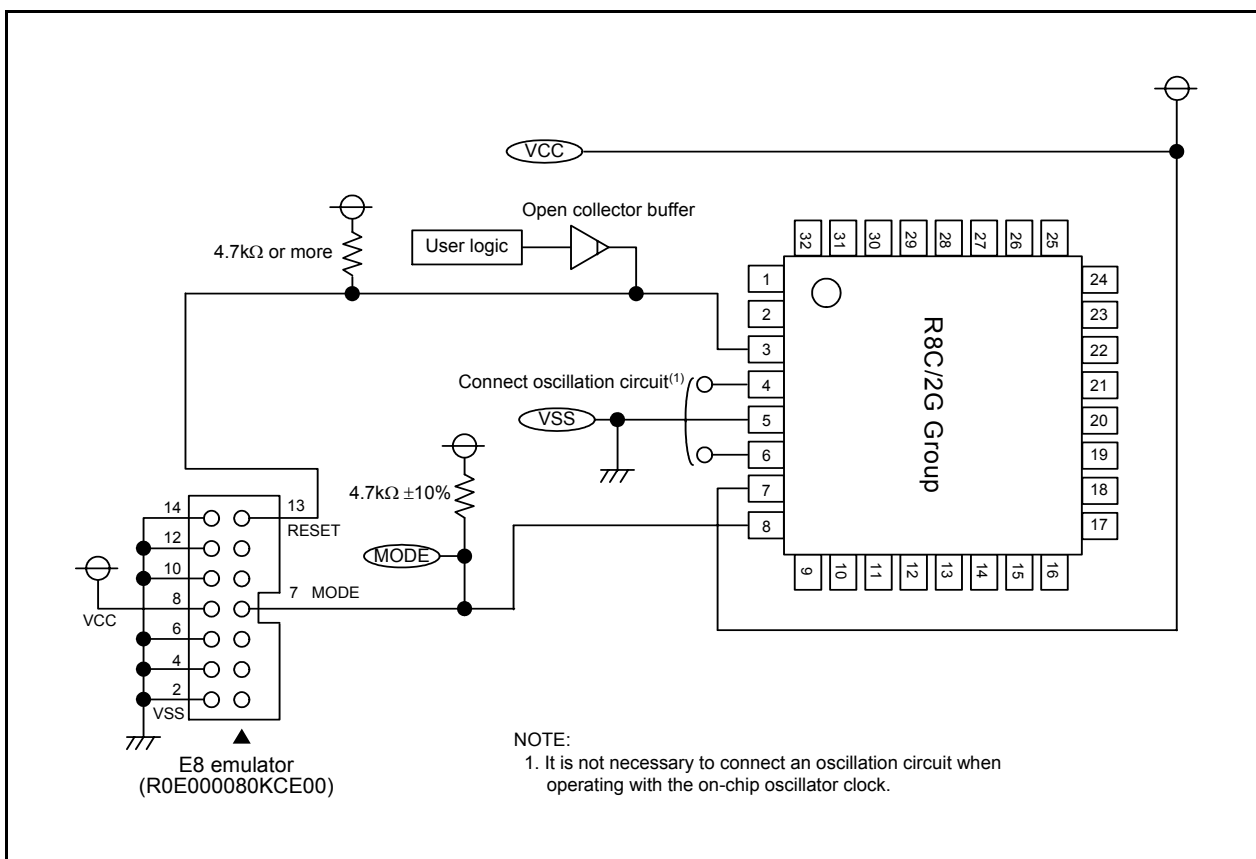
## Appendix 1. Package Dimensions

Diagrams showing the latest package dimensions and mounting information are available in the “Packages” section of the Renesas Technology website.



## Appendix 2. Connection Examples with On-Chip Debugging Emulator

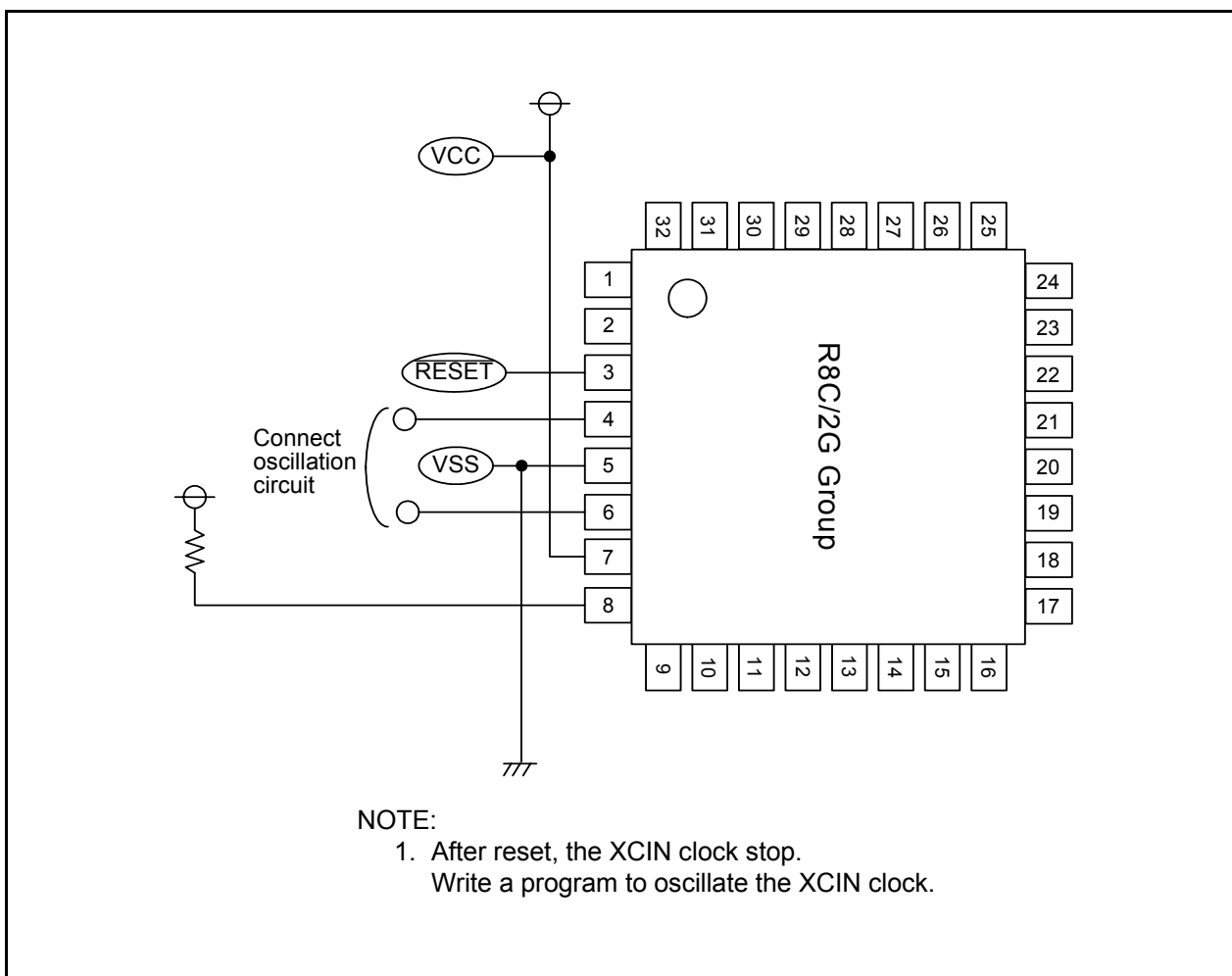
Appendix Figure 2.1 shows a Connection Example with E8 Emulator (R0E000080KCE00).



Appendix Figure 2.1 Connection Example with E8 Emulator (R0E000080KCE00)

## Appendix 3. Example of Oscillation Evaluation Circuit

Appendix Figure 3.1 shows an Example of Oscillation Evaluation Circuit.



Appendix Figure 3.1 Example of Oscillation Evaluation Circuit



## Index

[ A ]		PM0 .....	85
AIER .....	128	PM1 .....	85
ALCMR .....	52	PMR .....	74
		PRCR .....	107
[ B ]		PUR0 .....	74
BGRCR .....	53	PUR1 .....	74
BGRTRM .....	54		
BGRTRMA .....	48	[ R ]	
BGRTRMB .....	48	RMAD0 .....	128
		RMAD1 .....	128
[ C ]			
CAPIC .....	113	[ S ]	
CM0 .....	89	S0RIC .....	113
CM1 .....	90	S0TIC .....	113
CMP0IC .....	113	S2RIC .....	113
CMP1IC .....	113	S2TIC .....	113
CPSRF .....	93		
CSPR .....	140	[ T ]	
		TRA .....	147
[ F ]		TRACR .....	146
FMR0 .....	249	TRAIC .....	113
FMR1 .....	252	TRAI0C .....	146, 148, 151, 153, 155, 158
FMR4 .....	253	TRAMR .....	147
FRA4 .....	93	TRAPRE .....	147
FRA6 .....	93	TRBCR .....	162
		TRBIC .....	113
[ H ]		TRBIOC .....	163, 165, 169, 172, 176
HRA0 .....	92	TRBMR .....	163
HRA1 .....	92	TRBOCR .....	162
HRA2 .....	92	TRBPR .....	164
		TRBPPE .....	164
[ I ]		TRBSC .....	164
INT0IC .....	114	TRECR1 .....	187, 194
INT1IC .....	114	TRECR2 .....	188, 194
INT2IC .....	114	TRECSR .....	189, 195
INT4IC .....	114	TREHR .....	186
INTEN .....	121	TREIC .....	113
INTEN2 .....	122	TREMIN .....	185, 193
INTF .....	122	TREOPR .....	189
INTF2 .....	123	TRESEC .....	185, 193
		TREWK .....	186
[ K ]		TRF .....	202
KIEN .....	126	TRFCR0 .....	203
KUPIC .....	113	TRFCR1 .....	204
		TRFCR2 .....	203
[ L ]		TRFIC .....	113
LINCR .....	231	TRFM0 .....	202
LINST .....	232	TRFM1 .....	202
		TRFOUT .....	204
[ O ]			
OCD .....	91	[ U ]	
OFS .....	26, 135, 140, 247	U0BRG .....	215
		U0C0 .....	216
[ P ]		U0C1 .....	217
PD <sub>i</sub> (i = 0, 1, 3, 4, or 6) .....	71	U0MR .....	215
Pi (i = 0, 1, 3, 4, or 6) .....	72	U0RB .....	217
PINSR2 .....	73	U0TB .....	216
PINSR3 .....	73	U2BRG .....	215
PINSR4 .....	39, 54, 73	U2C0 .....	216
		U2C1 .....	217
		U2MR .....	215
		U2RB .....	217
		U2TB .....	216

## [ V ]

VCA1 .....	35, 49
VCA2 .....	35, 49, 94
VCAB .....	52
VCAC .....	39, 53
VCMP1IC .....	113
VCMP2IC .....	113
VW0C .....	36
VW1C .....	37, 50
VW2C .....	38, 51

## [ W ]

WDC .....	139
WDTR .....	139
WDTS .....	139

REVISION HISTORY

R8C/2G Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.01	Mar 30, 2007	–	First Edition issued
0.10	Jul 20, 2007	–	“RENESAS TECHNICAL UPDATE” reflected: TN-16C-A164A/E, TN-16C-A167A/E
		–	Register/bit symbols revised: “CM1POR” → “LCM1POR”, “CM2POR” → “LCM2POR”, “ACMR” → “ALCMR”
		2	Table 1.1: Clock; “Real-time clock (timer RE)” added
		5	Figure 1.3 “P4_4/(XCOUT) <sup>(1)</sup> ” → “P4_4/XCOUT”, “P4_3/(XCIN) <sup>(1)</sup> ” → “P4_3/XCIN”
		6	Table 1.3 “(XCOUT) <sup>(1)</sup> ” → “XCOUT”, “(XCIN) <sup>(1)</sup> ” → “XCIN”
		13, 36	Table 4.2, Figure 6.6: 0038h After reset; “0000X010b” → “1000X010b”, “0100X011b” → “1100X011b”
		25	Figure 5.3 revised
		26, 130, 135, 242	Figure 5.4, Figure 15.2, Figure 16.3, Figure 20.2: OFS Register; NOTE1 revised
		83	Figure 11.1 revised
		139	Table 17.1: Timer RE; “• fC32” deleted
		144	Figure 17.5 revised
		156	17.2 “The reload register .... same address” added
		159	Figure 17.15 “Programmable one-shot mode” → “Programmable one-shot generation mode”
		162	Figure 17.17 revised
		173	NOTE: “TRBIOC” added
		229	Figure 19.5 revised
		230	Figure 19.6 revised
		231	Figure 19.7: SFDCT flag in the LINST register; “Set by ....the B1CLR bit in the LINST register” → “Set by ....the B0CLR bit in the LINST register”
		233	Figure 19.9 revised
		236	Figure 19.12 revised
268	Figure 20.23: Title is revised		
270	Figure 20.24: Title is revised		
276	Figure 21.2 NOTE4 deleted		
282	Table 22.9 Parameter: “High-speed on-chip oscillator temperature supply voltage dependence” → “High-speed on-chip oscillator frequency temperature • supply voltage dependence”		
317	Appendix Figure 2.1 revised		
0.20	Nov 12, 2007	2	Table 1.1 I/O Ports: “• Output-only: 1” added “• CMOS I/O ports: 28” → “• CMOS I/O ports: 27”
		4	Figure 1.2 revised

Rev.	Date	Description	
		Page	Summary
0.20	Nov 12, 2007	5	Figure 1.3 revised
		6	Table 1.3 Pin Number: 4, 6, 20 revised
		7	Table 1.4 I/O port: "P4_3 to P4_5" → "P4_3, P4_5" Output port added
		12	Table 4.1 0006h "01001000b" → "01011000b"
		16	Table 4.5 0118h to 011Dh: After reset revised 011Fh "Timer RE Real-Time Clock Precision Adjust Register" added
		45	Figure 6.13 revised
		61	8. "There are 28 I/O ports ..... oscillation circuit is not used." → "There are 27 I/O ports ..... used as an output port." Table 8.1 revised, NOTE3 added
		65	Figure 8.3 revised
		67	Figure 8.5 NOTE3 "To use port P4_4 as ... an input port." added
		69	Figure 8.7 b7 revised
		70	Figure 8.9 PUR1: b1 revised
		71	Table 8.4 revised
		77	Table 8.26 NOTE2 added, Table 8.27 revised
		78	Table 8.29, Table 8.32 revised
		80	8.6 added
		83	Table 11.1 Oscillator status after reset: XCIN Clock Oscillation Circuit "Stop" → "Oscillate"
		85	Figure 11.2 revised
		93	11.2 "During and after reset, the XCIN clock stops." → "During and after reset, the XCIN clock oscillates."
		141	Figure 17.1 "TSTART" → "TCSTF"
		179	Figure 17.26 revised
		180	Table 17.11 Select function revised
		181	Figure 17.27, Figure 17.28 After Reset "00h" → "Undefined"
		182	Figure 17.29 After Reset "00h" → "X0XXXXXXb" Figure 17.30 After Reset "00h" → "X0000XXXb"
		183	Figure 17.31 After Reset "00h" → "XXX0X0X0b"
		184	Figure 17.33 After Reset "00h" → "00XXXXXXb"
		185	Figure 17.35 added
		187	Figure 17.37 revised
		188	Table 17.13 Select functions: Specification revised
		189	Figure 17.38, Figure 17.39 After Reset "00h" → "Undefined"
		190	Figure 17.40 After Reset "00h" → "XXX0X0X0b" Figure 17.41 After Reset "00h" → "00XXXXXXb"
193	17.3.3.1 NOTE revised		

REVISION HISTORY

R8C/2G Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.20	Nov 12, 2007	194	Figure 17.44 revised
		200	Figure 17.50 NOTE4 added
		234	Figure 19.9 revised
		279	Table 22.2 NOTE2 revised
		309	Figure 23.4 revised
1.00	Apr 04, 2008	All pages	"Under development" deleted
		2	Table 1.1 revised
		3	Table 1.2 "(D): Under development" deleted
		11	Figure 3.1 "Expanded area" deleted
		12	Table 4.1 "002Eh" "002Fh" revised
		13	Table 4.2 "003Eh" "003Fh" revised
		24	Figure 5.1 NOTE1 added
		25	Table 5.2 revised
		38, 51	Figure 6.8, Figure 7.5; "7. The VW2C7 ... 1." → "7. The VW2C7 ... 0."
		48	Figure 7.2 added
		53, 54	Figure 7.9, Figure 7.10 added
		63, 64	7.6, Figure 7.16, Figure 7.17 added
		107	12, Figure 12.1; "BGRCCR, and BGRTRM" added
		144	Table 17.1 Timer RF "Capture interrupt" added
		161	Figure 17.12 "TSTRAT" → "TSTART"
		171	Table 17.9 "TRBP pin function" → "TRBO pin function"
		235	Figure 19.6 "Three to five ..." → "One to two ..."
		238	Figure 19.9 revised
		244	Table 20.1 "Suspend function" deleted
		248	20.4 "The flash module ... (EW1 mode)." deleted Table 20.3 "... to erase-suspend" "... to program-suspend" deleted
		250	• FMR00 Bit "(including suspend periods)" deleted
		251	Table 20.4 "FRM0 Register ..." → "FMR0 Register ..."
		253	Figure 20.5 revised • FMR40 Bit, • FMR41 Bit, • FMR42 Bit, • FMR43 Bit, • FMR44 Bit, • FMR46 Bit; deleted
256	Figure 20.8 revised		
258	• Program Command; revised Old Figure 20.11 deleted		
259	• Block Erase; revised Old Figure 20.13, Old 20.4.3.2, Old Figure 20.14, Old Figure 20.15; deleted		
262	Figure 20.13 revised		
263	• Program Command; revised Old Figure 20.19 deleted		

REVISION HISTORY

R8C/2G Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Apr 04, 2008	264	• Block Erase; revised Old Figure 20.21, Old 20.4.4.2, Old Figure 20.22, Old Figure 20.23; deleted
		265	Table 20.6 revised
		267	Table 20.7 "P4_4 input/clock output" → "P4_4 output/clock output"
		270	Old 20.7.1.7, Old 20.7.1.8 deleted
		276	Table 22.3 revised Figure 22.2 deleted
		279	Table 22.8, Table 22.11 revised Table 22.9 revised, NOTE3 added
		281	Table 22.13 revised
		285	Table 22.19 revised
		289	Table 22.25 revised
		311	Old 23.9.1.7, Old 23.9.1.8 deleted

---

R8C/2G Group Hardware Manual

Publication Date: Rev.0.01 Mar 30, 2007  
Rev.1.00 Apr 04, 2008

Published by: Sales Strategic Planning Div.  
Renesas Technology Corp.

---

© 2008. Renesas Technology Corp., All rights reserved. Printed in Japan

# R8C/2G Group Hardware Manual



Renesas Technology Corp.  
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan