

**16****R8C/20** Group, **R8C/21** Group  
Hardware ManualRENESAS 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.2.00  
Revision Date: Aug 27, 2008Renesas Technology  
[www.renesas.com](http://www.renesas.com)

## 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/20 Group and R8C/21 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/20 Group, R8C/21 Group Datasheet	REJ03B0120
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/20 Group, R8C/21 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

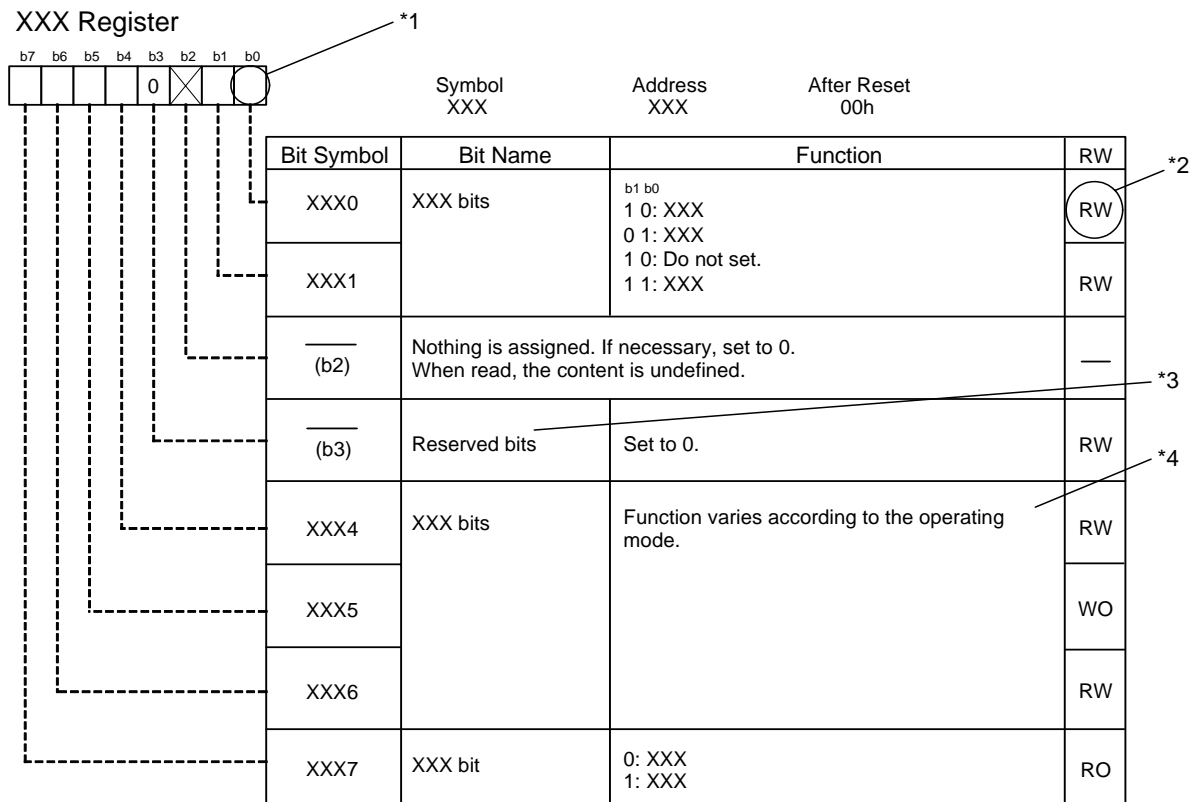
(2) Notation of Numbers

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 Applications .....	1
1.2 Performance Overview .....	2
1.3 Block Diagram .....	4
1.4 Product Information .....	5
1.5 Pin Assignments .....	7
1.6 Pin Functions .....	8
2. Central Processing Unit (CPU) .....	10
2.1 Data Registers (R0, R1, R2 and R3) .....	11
2.2 Address Registers (A0 and A1) .....	11
2.3 Frame Base Register (FB) .....	11
2.4 Interrupt Table Register (INTB) .....	11
2.5 Program Counter (PC) .....	11
2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP) .....	11
2.7 Static Base Register (SB) .....	11
2.8 Flag Register (FLG) .....	11
2.8.1 Carry Flag (C) .....	11
2.8.2 Debug Flag (D) .....	11
2.8.3 Zero Flag (Z) .....	11
2.8.4 Sign Flag (S) .....	11
2.8.5 Register Bank Select Flag (B) .....	11
2.8.6 Overflow Flag (O) .....	11
2.8.7 Interrupt Enable Flag (I) .....	12
2.8.8 Stack Pointer Select Flag (U) .....	12
2.8.9 Processor Interrupt Priority Level (IPL) .....	12
2.8.10 Reserved Bit .....	12
3. Memory .....	13
3.1 R8C/20 Group .....	13
3.2 R8C/21 Group .....	14
4. Special Function Registers (SFRs) .....	15
5. Resets .....	21
5.1 Hardware Reset .....	24
5.1.1 When Power Supply is Stable .....	24
5.1.2 Power On .....	24
5.2 Power-On Reset Function .....	26
5.3 Voltage Monitor 1 Reset .....	27
5.4 Voltage Monitor 2 Reset .....	27
5.5 Watchdog Timer Reset .....	27
5.6 Software Reset .....	27
6. Voltage Detection Circuit .....	28
6.1 VCC Input Voltage .....	34
6.1.1 Monitoring Vdet1 .....	34



6.1.2	Monitoring Vdet2 .....	34
6.2	Voltage Monitor 1 Reset .....	35
6.3	Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset .....	36
7.	Programmable I/O Ports .....	38
7.1	Functions of Programmable I/O Ports .....	38
7.2	Effect on Peripheral Functions .....	39
7.3	Pins Other than Programmable I/O Ports .....	39
7.4	Port Settings .....	50
7.5	Unassigned Pin Handling .....	61
8.	Processor Mode .....	62
8.1	Processor Modes .....	62
9.	Bus .....	63
10.	Clock Generation Circuit .....	64
10.1	XIN Clock .....	71
10.2	On-Chip Oscillator Clocks .....	72
10.2.1	Low-Speed On-Chip Oscillator Clock .....	72
10.2.2	High-Speed On-Chip Oscillator Clock .....	72
10.3	CPU Clock and Peripheral Function Clock .....	73
10.3.1	System Clock .....	73
10.3.2	CPU Clock .....	73
10.3.3	Peripheral Function Clock (f1, f2, f4, f8 and f32) .....	73
10.3.4	fOCO .....	73
10.3.5	fOCO40M .....	73
10.3.6	fOCO-F .....	73
10.3.7	fOCO-S .....	73
10.3.8	fOCO128 .....	73
10.4	Power Control .....	74
10.4.1	Standard Operating Mode .....	74
10.4.2	Wait Mode .....	75
10.4.3	Stop Mode .....	79
10.5	Oscillation Stop Detection Function .....	82
10.5.1	How to Use Oscillation Stop Detection Function .....	82
10.6	Notes on Clock Generation Circuit .....	85
10.6.1	Stop Mode .....	85
10.6.2	Wait Mode .....	85
10.6.3	Oscillation Stop Detection Function .....	85
10.6.4	Oscillation Circuit Constants .....	85
11.	Protection .....	86
12.	Interrupts .....	87
12.1	Interrupt Overview .....	87
12.1.1	Types of Interrupts .....	87
12.1.2	Software Interrupts .....	88
12.1.3	Special Interrupts .....	89

12.1.4	Peripheral Function Interrupt .....	89
12.1.5	Interrupts and Interrupt Vector .....	90
12.1.6	Interrupt Control .....	92
12.2	$\overline{\text{INT}}$ Interrupt .....	101
12.2.1	$\overline{\text{INT}}_i$ Interrupt (i = 0 to 3) .....	101
12.2.2	$\overline{\text{INT}}_i$ Input Filter (i = 0 to 3) .....	103
12.3	Key Input Interrupt .....	104
12.4	Address Match Interrupt .....	106
12.5	Timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupts and I <sup>2</sup> C bus Interface Interrupts (Interrupts with Multiple Interrupt Request Sources) .....	108
12.6	Notes on Interrupts .....	110
12.6.1	Reading Address 00000h .....	110
12.6.2	SP Setting .....	110
12.6.3	External Interrupt and Key Input Interrupt .....	110
12.6.4	Changing Interrupt Sources .....	111
12.6.5	Changing Interrupt Control Register Contents .....	112
13.	Watchdog Timer .....	113
13.1	Count Source Protection Mode Disabled .....	116
13.2	Count Source Protection Mode Enabled .....	117
14.	Timers .....	118
14.1	Timer RA .....	120
14.1.1	Timer Mode .....	124
14.1.2	Pulse Output Mode .....	126
14.1.3	Event Counter Mode .....	128
14.1.4	Pulse Width Measurement Mode .....	130
14.1.5	Pulse Period Measurement Mode .....	133
14.1.6	Notes on Timer RA .....	136
14.2	Timer RB .....	137
14.2.1	Timer Mode .....	141
14.2.2	Programmable Waveform Generation Mode .....	144
14.2.3	Programmable One-shot Generation Mode .....	147
14.2.4	Programmable Wait One-shot Generation Mode .....	151
14.2.5	Notes on Timer RB .....	155
14.3	Timer RD .....	159
14.3.1	Count Source .....	164
14.3.2	Buffer Operation .....	165
14.3.3	Synchronous Operation .....	167
14.3.4	Pulse Output Forced Cutoff .....	168
14.3.5	Input Capture Function .....	170
14.3.6	Output Compare Function .....	184
14.3.7	PWM Mode .....	200
14.3.8	Reset Synchronous PWM Mode .....	213
14.3.9	Complementary PWM Mode .....	223
14.3.10	PWM3 Mode .....	237
14.3.11	Timer RD Interrupt .....	249
14.3.12	Notes on Timer RD .....	251
14.4	Timer RE .....	257

14.4.1	Output Compare Mode .....	258
14.4.2	Notes on Timer RE .....	264
<b>15.</b>	<b>Serial Interface .....</b>	<b>265</b>
15.1	Clock Synchronous Serial I/O Mode .....	271
15.1.1	Polarity Select Function .....	274
15.1.2	LSB First/MSB First Select Function .....	274
15.1.3	Continuous Receive Mode .....	275
15.2	Clock Asynchronous Serial I/O (UART) Mode .....	276
15.2.1	Bit Rate .....	280
15.3	Notes on Serial Interface .....	281
<b>16.</b>	<b>Clock Synchronous Serial Interface .....</b>	<b>282</b>
16.1	Mode Selection .....	282
16.2	Clock Synchronous Serial I/O with Chip Select (SSU) .....	283
16.2.1	Transfer Clock .....	292
16.2.2	SS Shift Register (SSTRSR) .....	294
16.2.3	Interrupt Requests .....	295
16.2.4	Communication Modes and Pin Functions .....	296
16.2.5	Clock Synchronous Communication Mode .....	297
16.2.6	Operation in 4-Wire Bus Communication Mode .....	304
16.2.7	SCS Pin Control and Arbitration .....	310
16.2.8	Notes on Clock Synchronous Serial I/O with Chip Select .....	311
16.3	I <sup>2</sup> C Bus Interface .....	312
16.3.1	Transfer Clock .....	322
16.3.2	Interrupt Requests .....	323
16.3.3	I <sup>2</sup> C Bus Interface Mode .....	324
16.3.4	Clock Synchronous Serial Mode .....	335
16.3.5	Noise Canceller .....	338
16.3.6	Bit Synchronization Circuit .....	339
16.3.7	Examples of Register Setting .....	340
16.3.8	Notes on I <sup>2</sup> C Bus Interface .....	344
<b>17.</b>	<b>Hardware LIN .....</b>	<b>345</b>
17.1	Features .....	345
17.2	Input/Output Pins .....	346
17.3	Register Configuration .....	347
17.4	Functional Description .....	349
17.4.1	Master Mode .....	349
17.4.2	Slave Mode .....	352
17.4.3	Bus Collision Detection Function .....	356
17.4.4	Hardware LIN End Processing .....	357
17.5	Interrupt Requests .....	358
17.6	Notes on Hardware LIN .....	359
<b>18.</b>	<b>A/D Converter .....</b>	<b>360</b>
18.1	One-Shot Mode .....	364
18.2	Repeat Mode .....	367
18.3	Sample and Hold .....	370

18.4	A/D Conversion Cycles .....	371
18.5	Internal Equivalent Circuit of Analog Input .....	372
18.6	Output Impedance of Sensor Under A/D Conversion .....	373
18.7	Notes on A/D Converter .....	374
<b>19.</b>	<b>Flash Memory .....</b>	<b>375</b>
19.1	Overview .....	375
19.2	Memory Map .....	377
19.3	Functions to Prevent Rewriting of Flash Memory .....	380
19.3.1	ID Code Check Function .....	380
19.3.2	ROM Code Protect Function .....	381
19.4	CPU Rewrite Mode .....	382
19.4.1	EW0 Mode .....	383
19.4.2	EW1 Mode .....	383
19.4.3	Software Commands .....	392
19.4.4	Status Registers .....	397
19.4.5	Full Status Check .....	398
19.5	Standard Serial I/O Mode .....	400
19.5.1	ID Code Check Function .....	400
19.6	Parallel I/O Mode .....	404
19.6.1	ROM Code Protect Function .....	404
19.7	Notes on Flash Memory .....	405
19.7.1	CPU Rewrite Mode .....	405
<b>20.</b>	<b>Electrical Characteristics .....</b>	<b>408</b>
<b>21.</b>	<b>Usage Notes .....</b>	<b>428</b>
21.1	Notes on Clock Generation Circuit .....	428
21.1.1	Stop Mode .....	428
21.1.2	Wait Mode .....	428
21.1.3	Oscillation Stop Detection Function .....	428
21.1.4	Oscillation Circuit Constants .....	428
21.2	Notes on Interrupts .....	429
21.2.1	Reading Address 00000h .....	429
21.2.2	SP Setting .....	429
21.2.3	External Interrupt and Key Input Interrupt .....	429
21.2.4	Changing Interrupt Sources .....	430
21.2.5	Changing Interrupt Control Register Contents .....	431
21.3	Notes on Timers .....	432
21.3.1	Notes on Timer RA .....	432
21.3.2	Notes on Timer RB .....	433
21.3.3	Notes on Timer RD .....	437
21.3.4	Notes on Timer RE .....	443
21.4	Notes on Serial Interface .....	444
21.5	Clock Synchronous Serial Interface .....	445
21.5.1	Notes on Clock Synchronous Serial I/O with Chip Select .....	445
21.5.2	Notes on I <sup>2</sup> C Bus Interface .....	445
21.6	Notes on Hardware LIN .....	446
21.7	Notes on A/D Converter .....	447

21.8	Notes on Flash Memory .....	448
21.8.1	CPU Rewrite Mode .....	448
21.9	Notes on Noise .....	451
21.9.1	Inserting a Bypass Capacitor between VCC and VSS Pins as a Countermeasure against Noise and Latch-up .....	451
21.9.2	Countermeasures against Noise Error of Port Control Registers .....	451
22.	Notes on On-Chip Debugger .....	452
23.	Notes on Emulator Debugger .....	453
Appendix 1.	Package Dimensions .....	454
Appendix 2.	Connection Examples between Serial Writer and On-Chip Debugging Emulator .....	455
Appendix 3.	Example of Oscillation Evaluation Circuit .....	456
Index	.....	457

# SFR Page Reference

Address	Register	Symbol	Page
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	62
0005h	Processor Mode Register 1	PM1	62
0006h	System Clock Control Register 0	CM0	66
0007h	System Clock Control Register 1	CM1	67
0008h			
0009h			
000Ah	Protect Register	PRCR	86
000Bh			
000Ch	Oscillation Stop Detection Register	OCD	68
000Dh	Watchdog Timer Reset Register	WDTR	115
000Eh	Watchdog Timer Start Register	WDTS	115
000Fh	Watchdog Timer Control Register	WDC	114
0010h	Address Match Interrupt Register 0	RMAD0	107
0011h			
0012h			
0013h	Address Match Interrupt Enable Register	AIER	107
0014h	Address Match Interrupt Register 1	RMAD1	107
0015h			
0016h			
0017h			
0018h			
0019h			
001Ah			
001Bh			
001Ch	Count Source Protect Mode Register	CSPR	115
001Dh			
001Eh			
001Fh			
0020h			
0021h			
0022h			
0023h	High-Speed On-Chip Oscillator Control Register 0	FRA0	69
0024h	High-Speed On-Chip Oscillator Control Register 1	FRA1	69
0025h	High-Speed On-Chip Oscillator Control Register 2	FRA2	70
0026h			
0027h			
0028h			
0029h			
002Ah			
002Bh			
002Ch			
002Dh			
002Eh			
002Fh			
0030h			
0031h	Voltage Detection Register 1	VCA1	31
0032h	Voltage Detection Register 2	VCA2	31, 70
0033h			
0034h			
0035h			
0036h	Voltage Monitor 1 Circuit Control Register	VW1C	32
0037h	Voltage Monitor 2 Circuit Control Register	VW2C	33
0038h			
0039h			
003Ah			
003Bh			
003Ch			
003Dh			
003Eh			
003Fh			

Address	Register	Symbol	Page
0040h			
0041h			
0042h			
0043h			
0044h			
0045h			
0046h			
0047h			
0048h	Timer RD0 Interrupt Control Register	TRD0IC	93
0049h	Timer RD1 Interrupt Control Register	TRD1IC	93
004Ah	Timer RE Interrupt Control Register	TREIC	92
004Bh			
004Ch			
004Dh	Key Input Interrupt Control Register	KUPIC	92
004Eh	A/D Conversion Interrupt Control Register	ADIC	92
004Fh	SSU Interrupt Control Register/IIC Bus Interrupt Control Register	SSUIC/IICIC	93
0050h			
0051h	UART0 Transmit Interrupt Control Register	S0TIC	92
0052h	UART0 Receive Interrupt Control Register	S0RIC	92
0053h	UART1 Transmit Interrupt Control Register	S1TIC	92
0054h	UART1 Receive Interrupt Control Register	S1RIC	92
0055h	INT2 Interrupt Control Register	INT2IC	94
0056h	Timer RA Interrupt Control Register	TRAIC	92
0057h			
0058h	Timer RB Interrupt Control Register	TRBIC	92
0059h	INT1 Interrupt Control Register	INT1IC	94
005Ah	INT3 Interrupt Control Register	INT3IC	94
005Bh			
005Ch			
005Dh	INT0 Interrupt Control Register	INT0IC	94
005Eh			
005Fh			
0060h			
0061h			
0062h			
0063h			
0064h			
0065h			
0066h			
0067h			
0068h			
0069h			
006Ah			
006Bh			
006Ch			
006Dh			
006Eh			
006Fh			
0070h			
0071h			
0072h			
0073h			
0074h			
0075h			
0076h			
0077h			
0078h			
0079h			
007Ah			
007Bh			
007Ch			
007Dh			
007Eh			
007Fh			

NOTE:

- Blank columns are all reserved space. No access is allowed.

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	268
00A1h	UART0 Bit Rate Register	U0BRG	267
00A2h	UART0 Transmit Buffer Register	U0TB	267
00A3h			
00A4h	UART0 Transmit/Receive Control Register 0	U0C0	269
00A5h	UART0 Transmit/Receive Control Register 1	U0C1	269
00A6h	UART0 Receive Buffer Register	U0RB	267
00A7h			
00A8h	UART1 Transmit/Receive Mode Register	U1MR	268
00A9h	UART1 Bit Rate Register	U1BRG	267
00AAh	UART1 Transmit Buffer Register	U1TB	267
00ABh			
00ACh	UART1 Transmit/Receive Control Register 0	U1C0	269
00ADh	UART1 Transmit/Receive Control Register 1	U1C1	269
00AEh	UART1 Receive Buffer Register	U1RB	267
00AFh			
00B0h			
00B1h			
00B2h			
00B3h			
00B4h			
00B5h			
00B6h			
00B7h			
00B8h	SS Control Register H/IIC Bus Control Register 1	SSCRH/ICCR1	285, 315
00B9h	SS Control Register L/IIC Bus Control Register 2	SSCRL/ICCR2	286, 316
00BAh	SS Mode Register/IIC Bus Mode Register 1	SSMR/ICMR	287, 317
00BBh	SS Enable Register/IIC Bus Interrupt Enable Register	SSER/ICIER	288, 318
00BCh	SS Status Register/IIC Bus Status Register	SSSR/ICSR	289, 319
00BDh	SS Mode Register 2/Slave Address Register	SSMR2/SAR	290, 320
00BEh	SS Transmit Data Register/IIC Bus Transmit Data Register	SSTDR/ICDRT	291, 320
00BFh	SS Receive Data Register/IIC Bus Receive Data Register	SSRDR/ICDRR	291, 321

Address	Register	Symbol	Page
00C0h	A/D Register	AD	363
00C1h			
00C2h			
00C3h			
00C4h			
00C5h			
00C6h			
00C7h			
00C8h			
00C9h			
00CAh			
00CBh			
00CCh			
00CDh			
00CEh			
00CFh			
00D0h			
00D1h			
00D2h			
00D3h			
00D4h	A/D Control Register 2	ADCON2	363
00D5h			
00D6h	A/D Control Register 0	ADCON0	362, 365, 368
00D7h	A/D Control Register 1	ADCON1	363, 366, 369
00D8h			
00D9h			
00DAh			
00DBh			
00DCh			
00DDh			
00DEh			
00DFh			
00E0h	Port P0 Register	P0	48
00E1h	Port P1 Register	P1	48
00E2h	Port P0 Direction Register	PD0	48
00E3h	Port P1 Direction Register	PD1	48
00E4h	Port P2 Register	P2	48
00E5h	Port P3 Register	P3	48
00E6h	Port P2 Direction Register	PD2	48
00E7h	Port P3 Direction Register	PD3	48
00E8h	Port P4 Register	P4	48
00E9h			
00EAh	Port P4 Direction Register	PD4	48
00EBh			
00ECh	Port P6 Register	P6	48
00EDh			
00EEh	Port P6 Direction Register	PD6	48
00EFh			
00F0h			
00F1h			
00F2h			
00F3h			
00F4h			
00F5h	UART1 Function Select Register	U1SR	270
00F6h			
00F7h			
00F8h	Port Mode Register	PMR	49, 270, 291, 321
00F9h	External Input Enable Register	INTEN	101
00FAh	INT Input Filter Select Register	INTF	102
00FBh	Key Input Enable Register	KIEN	105
00FCh	Pull-Up Control Register 0	PUR0	49
00FDh	Pull-Up Control Register 1	PUR1	49
00FEh			
00FFh			

NOTE:

- Blank columns are all reserved space. No access is allowed.

Address	Register	Symbol	Page
0100h	Timer RA Control Register	TRACR	121
0101h	Timer RA I/O Control Register	TRAIOC	121, 124, 127, 129, 131, 134
0102h	Timer RA Mode Register	TRAMR	122
0103h	Timer RA Prescaler Register	TRAPRE	122
0104h	Timer RA Register	TRA	123
0105h			
0106h	LIN Control Register	LINCR	347
0107h	LIN Status Register	LINST	348
0108h	Timer RB Control Register	TRBCR	138
0109h	Timer RB One-Shot Control Register	TRBOCR	138
010Ah	Timer RB I/O Control Register	TRBIOC	139, 141, 145, 148, 153
010Bh	Timer RB Mode Register	TRBMR	139
010Ch	Timer RB Prescaler Register	TRBPRE	140
010Dh	Timer RB Secondary Register	TRBSC	140
010Eh	Timer RB Primary	TRBPR	140
010Fh			
0110h			
0111h			
0112h			
0113h			
0114h			
0115h			
0116h			
0117h			
0118h	Timer RE Counter Data Register	TRESEC	260
0119h	Timer RE Compare Data Register	TREMIN	260
011Ah			
011Bh			
011Ch	Timer RE Control Register 1	TRECR1	261
011Dh	Timer RE Control Register 2	TRECR2	261
011Eh	Timer RE Count Source Select Register	TRECSR	262
011Fh			

0130h			
0131h			
0132h			
0133h			
0134h			
0135h			
0136h			
0137h	Timer RD Start Register	TRDSTR	172, 186, 202, 215, 225, 239
0138h	Timer RD Mode Register	TRDMR	172, 186, 203, 216, 226, 240
0139h	Timer RD PWM Mode Register	TRDPMR	173, 187, 203
013Ah	Timer RD Function Control Register	TRDFCR	174, 188, 204, 216, 227, 240
013Bh	Timer RD Output Master Enable Register 1	TRDOER1	189, 205, 217, 228, 241
013Ch	Timer RD Output Master Enable Register 2	TRDOER2	189, 205, 217, 228, 241
013Dh	Timer RD Output Control Register	TRDOCR	190, 206, 242
013Eh	Timer RD Digital Filter Function Select Register 0	TRDDF0	175
013Fh	Timer RD Digital Filter Function Select Register 1	TRDDF1	175

Address	Register	Symbol	Page
0140h	Timer RD Control Register 0	TRDCR0	176, 191, 206, 218, 229, 243
0141h	Timer RD I/O Control Register A0	TRDIORA0	177, 192
0142h	Timer RD I/O Control Register C0	TRDIORC0	178, 193
0143h	Timer RD Status Register 0	TRDSR0	179, 194, 207, 219, 230, 244
0144h	Timer RD Interrupt Enable Register 0	TRDIER0	180, 195, 208, 220, 231, 245
0145h	Timer RD PWM Mode Output Level Control Register 0	TRDPOCR0	209
0146h	Timer RD Counter 0	TRD0	180, 195, 209, 220, 232, 245
0147h			
0148h	Timer RD General Register A0	TRDGRA0	181, 196, 210, 221, 232, 246
0149h			
014Ah	Timer RD General Register B0	TRDGRB0	181, 196, 210, 221, 232, 246
014Bh			
014Ch	Timer RD General Register C0	TRDGRC0	181, 196, 210, 221, 246
014Dh			
014Eh	Timer RD General Register D0	TRDGRD0	181, 196, 210, 221, 232, 246
014Fh			
0150h	Timer RD Control Register 1	TRDCR1	176, 191, 206, 229
0151h	Timer RD I/O Control Register A1	TRDIORA1	177, 192
0152h	Timer RD I/O Control Register C1	TRDIORC1	178, 193
0153h	Timer RD Status Register 1	TRDSR1	179, 194, 207, 219, 230, 244
0154h	Timer RD Interrupt Enable Register 1	TRDIER1	180, 195, 208, 220, 231, 245
0155h	Timer RD PWM Mode Output Level Control Register 1	TRDPOCR1	209
0156h	Timer RD Counter 1	TRD1	180, 195, 209, 232
0157h			
0158h	Timer RD General Register A1	TRDGRA1	181, 196, 210, 221, 232, 246
0159h			
015Ah	Timer RD General Register B1	TRDGRB1	181, 196, 210, 221, 232, 246
015Bh			
015Ch	Timer RD General Register C1	TRDGRC1	181, 196, 210, 221, 232, 246
015Dh			
015Eh	Timer RD General Register D1	TRDGRD1	181, 196, 210, 221, 232, 246
015Fh			

01B0h			
01B1h			
01B2h			
01B3h	Flash Memory Control Register 4	FMR4	388
01B4h			
01B5h	Flash Memory Control Register 1	FMR1	387
01B6h			
01B7h	Flash Memory Control Register 0	FMR0	386
01B8h			
01B9h			
01BAh			
01BBh			

FFFFh	Option Function Select Register	OFS	23, 114, 381
-------	---------------------------------	-----	--------------

NOTE:

- Blank columns are all reserved space. No access is allowed.



## 1. Overview

This MCU is built using the high-performance silicon gate CMOS process using the R8C CPU core and is packaged in a 48-pin plastic molded LQFP. This MCU operates using sophisticated instructions featuring a high level of instruction efficiency. With 1 Mbyte of address space, it is capable of executing instructions at high speed. This Furthermore, the data flash (1 KB x 2 blocks) is embedded in the R8C/21 Group.

The difference between R8C/20 and R8C/21 Groups is only the existence of the data flash. Their peripheral functions are the same.

### 1.1 Applications

Automotive, etc.

## 1.2 Performance Overview

Table 1.1 outlines the Functions and Specifications for R8C/20 Group and Table 1.2 outlines the Functions and Specifications for R8C/21 Group.

**Table 1.1 Functions and Specifications for R8C/20 Group**

	Item	Specification
CPU	Number of fundamental instructions	89 instructions
	Minimum instruction execution time	50 ns ( $f(XIN) = 20$ MHz, $VCC = 3.0$ to $5.5$ V) 100 ns ( $f(XIN) = 10$ MHz, $VCC = 2.7$ to $5.5$ V)
	Operating mode	Single-chip
	Address space	1 Mbyte
	Memory capacity	Refer to <b>Table 1.3 Product Information for R8C/20 Group</b>
Peripheral Function	Ports	I/O ports: 41 pins, Input port: 3 pins
	Timers	Timer RA: 8 bits x 1 channel, Timer RB: 8 bits x 1 channel (Each timer equipped with 8-bit prescaler) Timer RD: 16 bits x 2 channel (Circuits of input capture and output compare) Timer RE: With compare match function
	Serial interface	1 channel (UART0) Clock synchronous I/O, UART 1 channel (UART1) UART
	Clock synchronous serial interface	1 channel I <sup>2</sup> C bus interface <sup>(2)</sup> , Clock synchronous serial I/O with chip select
	LIN module	Hardware LIN: 1 channel (timer RA, UART0)
	A/D converter	10-bit A/D converter: 1 circuit, 12 channels
	Watchdog timer	15 bits x 1 channel (with prescaler) Reset start selectable
	Interrupt	Internal: 11 sources, External: 5 sources, Software: 4 sources, Priority level: 7 levels
	Clock generation circuits	2 circuits XIN clock generation circuit (with on-chip feedback resistor) On-chip oscillator (high speed, low speed) High-speed on-chip oscillator has frequency adjustment function.
	Oscillation stop detection function	Stop detection of XIN clock oscillation
	Voltage detection circuit	On-chip
	Power-on reset circuit include	On-chip
	Electric Characteristics	Supply voltage
Current consumption		Typ. 11.0 mA ( $VCC = 5$ V, $f(XIN) = 20$ MHz, High-speed on-chip oscillator stopping) Typ. 5.3 mA ( $VCC = 5$ V, $f(XIN) = 10$ MHz, High-speed on-chip oscillator stopping)
Flash Memory	Programming and erasure voltage	$VCC = 2.7$ to $5.5$ V
	Programming and erasure endurance	100 times
Operating Ambient Temperature		-40 to 85°C
		-40 to 125°C (option <sup>(1)</sup> )
Package		48-pin mold-plastic LQFP

### NOTES:

1. When using options, be sure to inquire about the specification.
2. I<sup>2</sup>C bus is a registered trademark of Koninklijke Philips Electronics N.V.

**Table 1.2 Functions and Specifications for R8C/21 Group**

Item		Specification
CPU	Number of fundamental instructions	89 instructions
	Minimum instruction execution time	50 ns (f(XIN) = 20 MHz, VCC = 3.0 to 5.5 V) 100 ns (f(XIN) = 10 MHz, VCC = 2.7 to 5.5 V)
	Operating mode	Single-chip
	Address space	1 Mbyte
	Memory capacity	Refer to <b>Table 1.4 Product Information for R8C/21 Group</b>
Peripheral Function	Ports	I/O ports: 41 pins, Input port: 3 pins
	Timers	Timer RA: 8 bits x 1 channel, Timer RB: 8 bits x 1 channel (Each timer equipped with 8-bit prescaler) Timer RD: 16 bits x 2 channel (Circuits of input capture and output compare) Timer RE: With compare match function
	Serial interface	1 channel (UART0) Clock synchronous I/O, UART 1 channel (UART1) UART
	Clock synchronous serial interface	1 channel I <sup>2</sup> C bus interface <sup>(2)</sup> , Clock synchronous serial I/O with chip select
	LIN module	Hardware LIN: 1 channel (Timer RA, UART0)
	A/D converter	10-bit A/D converter: 1 circuit, 12 channels
	Watchdog timer	15 bits x 1 channel (with prescaler) Reset start selectable
	Interrupts	Internal: 11 sources, External: 5 sources, Software: 4 sources, Priority level: 7 levels
	Clock generation circuits	2 circuits XIN clock generation circuit (with on-chip feedback resistor) On-chip oscillator (high speed, low speed) High-speed on-chip oscillator has frequency adjustment function.
	Oscillation stop detection function	Stop detection of XIN clock oscillation
	Voltage detection circuit	On-chip
	Power-on reset circuit include	On-chip
	Electric Characteristics	Supply voltage
Current consumption		Typ. 11.0 mA (VCC = 5 V, f(XIN) = 20 MHz, High-speed on-chip oscillator stopping) Typ. 5.3 mA (VCC = 5 V, f(XIN) = 10 MHz, High-speed on-chip oscillator stopping)
Flash Memory	Programming and erasure voltage	VCC = 2.7 to 5.5 V
	Programming and erasure endurance	10,000 times (data flash) 1,000 times (program ROM)
Operating Ambient Temperature		-40 to 85°C
		-40 to 125°C (option <sup>(1)</sup> )
Package		48-pin mold-plastic LQFP

## NOTES:

1. When using options, be sure to inquire about the specification.
2. I<sup>2</sup>C bus is a registered trademark of Koninklijke Philips Electronics N.V.

### 1.3 Block Diagram

Figure 1.1 shows a Block Diagram.

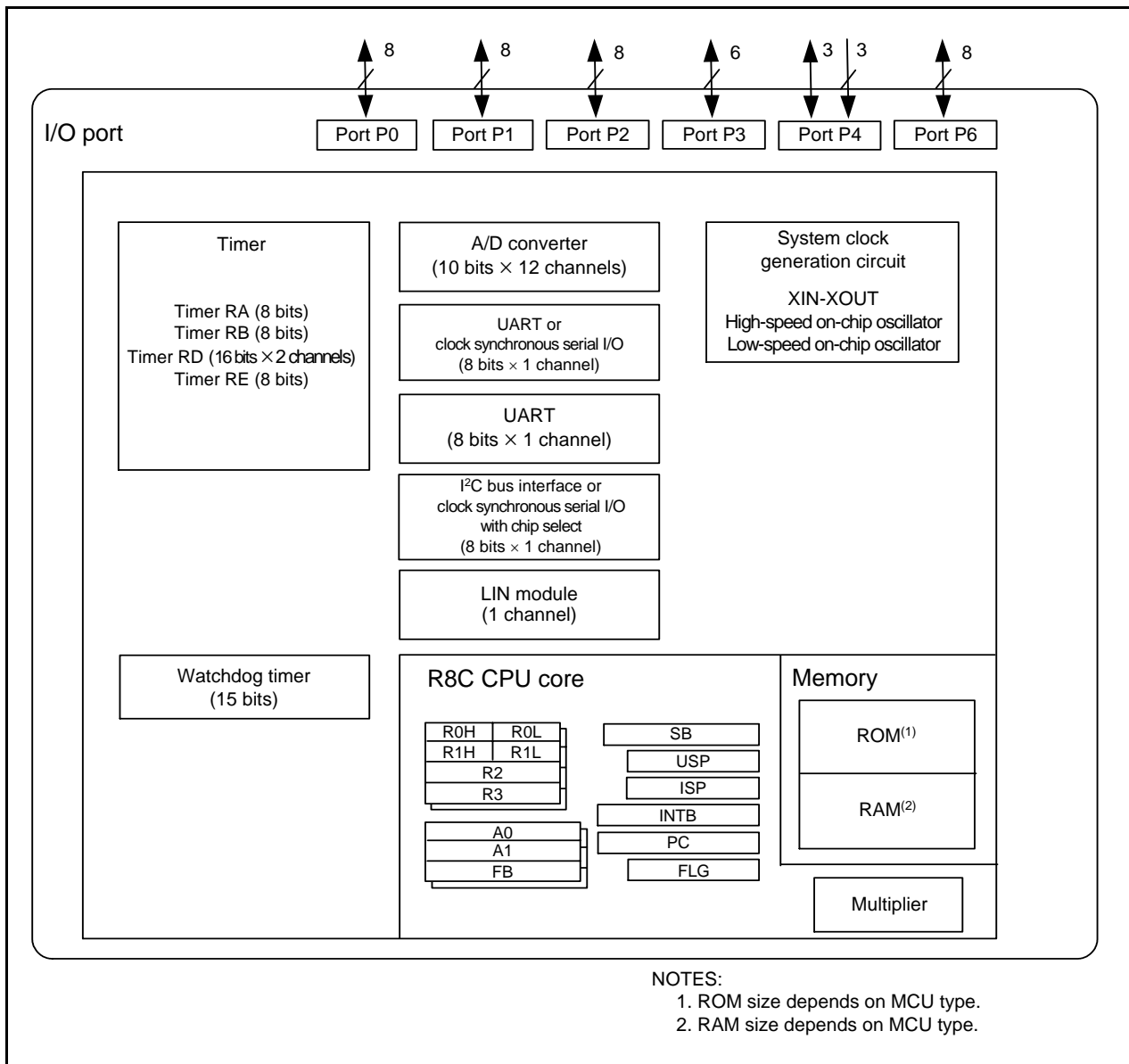


Figure 1.1 Block Diagram

## 1.4 Product Information

Table 1.3 lists Product Information for R8C/20 Group and Table 1.4 lists Product Information for R8C/21 Group.

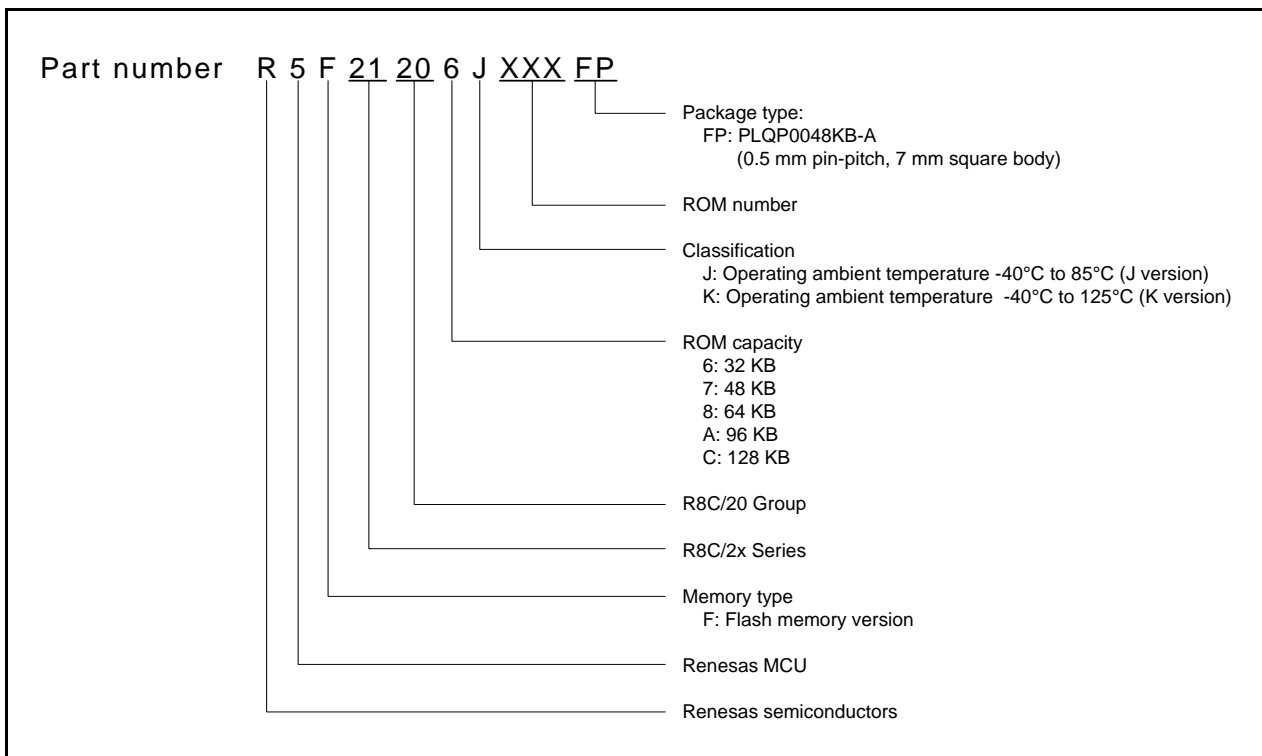
**Table 1.3 Product Information for R8C/20 Group**

**Current of Aug. 2008**

Type No.	ROM Capacity	RAM Capacity	Package Type	Remarks	
R5F21206JFP	32 Kbytes	2 Kbytes	PLQP0048KB-A	J version	Flash memory version
R5F21207JFP	48 Kbytes	2.5 Kbytes	PLQP0048KB-A		
R5F21208JFP	64 Kbytes	3 Kbytes	PLQP0048KB-A		
R5F2120AJFP	96 Kbytes	5 Kbytes	PLQP0048KB-A		
R5F2120CJFP	128 Kbytes <sup>(1)</sup>	6 Kbytes	PLQP0048KB-A		
R5F21206KFP	32 Kbytes	2 Kbytes	PLQP0048KB-A	K version	
R5F21207KFP	48 Kbytes	2.5 Kbytes	PLQP0048KB-A		
R5F21208KFP	64 Kbytes	3 Kbytes	PLQP0048KB-A		
R5F2120AKFP	96 Kbytes	5 Kbytes	PLQP0048KB-A		
R5F2120CKFP	128 Kbytes <sup>(1)</sup>	6 Kbytes	PLQP0048KB-A		

**NOTE:**

- Do not use addresses 20000h to 23FFFh because these areas are used for the emulator debugger. Refer to **23. Notes on Emulator Debugger**.



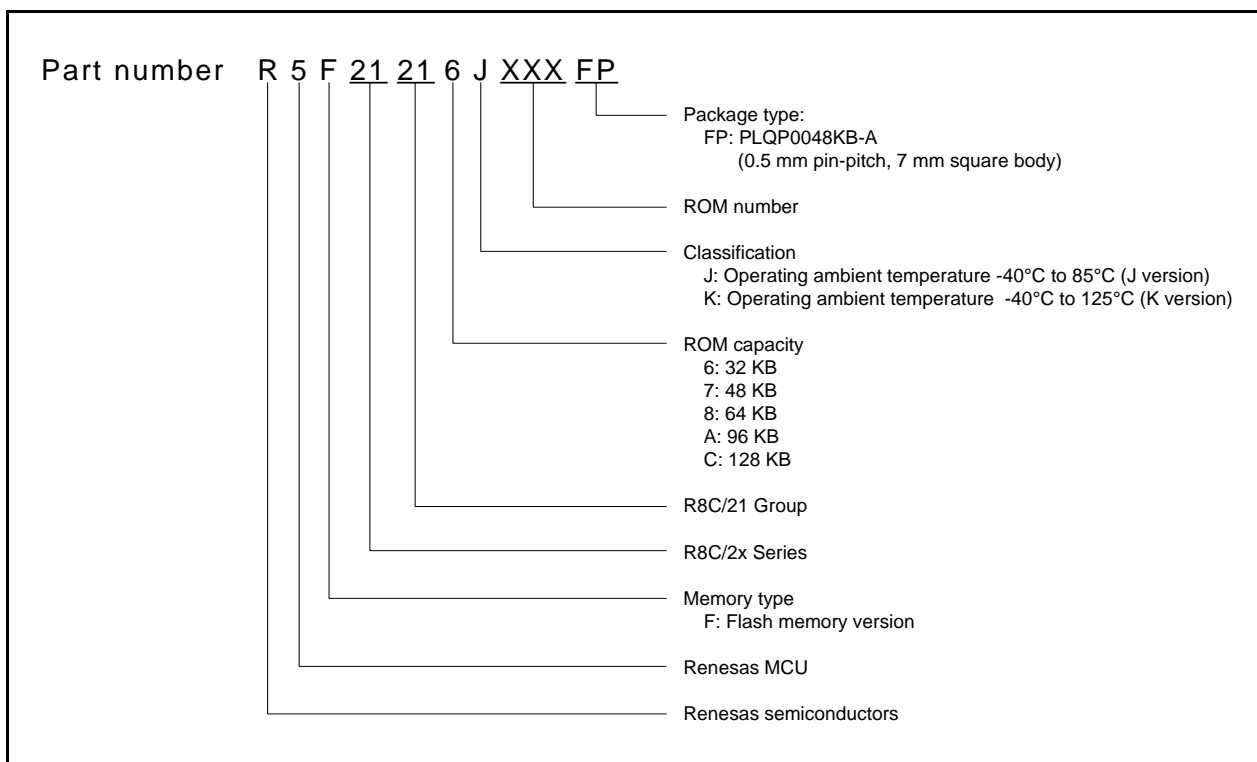
**Figure 1.2 Type Number, Memory Size, and Package of R8C/20 Group**

**Table 1.4 Product Information for R8C/21 Group** **Current of Aug. 2008**

Type No.	ROM Capacity		RAM Capacity	Package Type	Remarks	
	Program ROM	Data Flash				
R5F21216JFP	32 Kbytes	1 Kbyte X 2	2 Kbytes	PLQP0048KB-A	J version	Flash memory version
R5F21217JFP	48 Kbytes	1 Kbyte X 2	2.5 Kbytes	PLQP0048KB-A		
R5F21218JFP	64 Kbytes	1 Kbyte X 2	3 Kbytes	PLQP0048KB-A		
R5F2121AJFP	96 Kbytes	1 Kbyte X 2	5 Kbytes	PLQP0048KB-A		
R5F2121CJFP	128 Kbytes <sup>(1)</sup>	1 Kbyte X 2	6 Kbytes	PLQP0048KB-A		
R5F21216KFP	32 Kbytes	1 Kbyte X 2	2 Kbytes	PLQP0048KB-A	K version	
R5F21217KFP	48 Kbytes	1 Kbyte X 2	2.5 Kbytes	PLQP0048KB-A		
R5F21218KFP	64 Kbytes	1 Kbyte X 2	3 Kbytes	PLQP0048KB-A		
R5F2121AKFP	96 Kbytes	1 Kbyte X 2	5 Kbytes	PLQP0048KB-A		
R5F2121CKFP	128 Kbytes <sup>(1)</sup>	1 Kbyte X 2	6 Kbytes	PLQP0048KB-A		

**NOTE:**

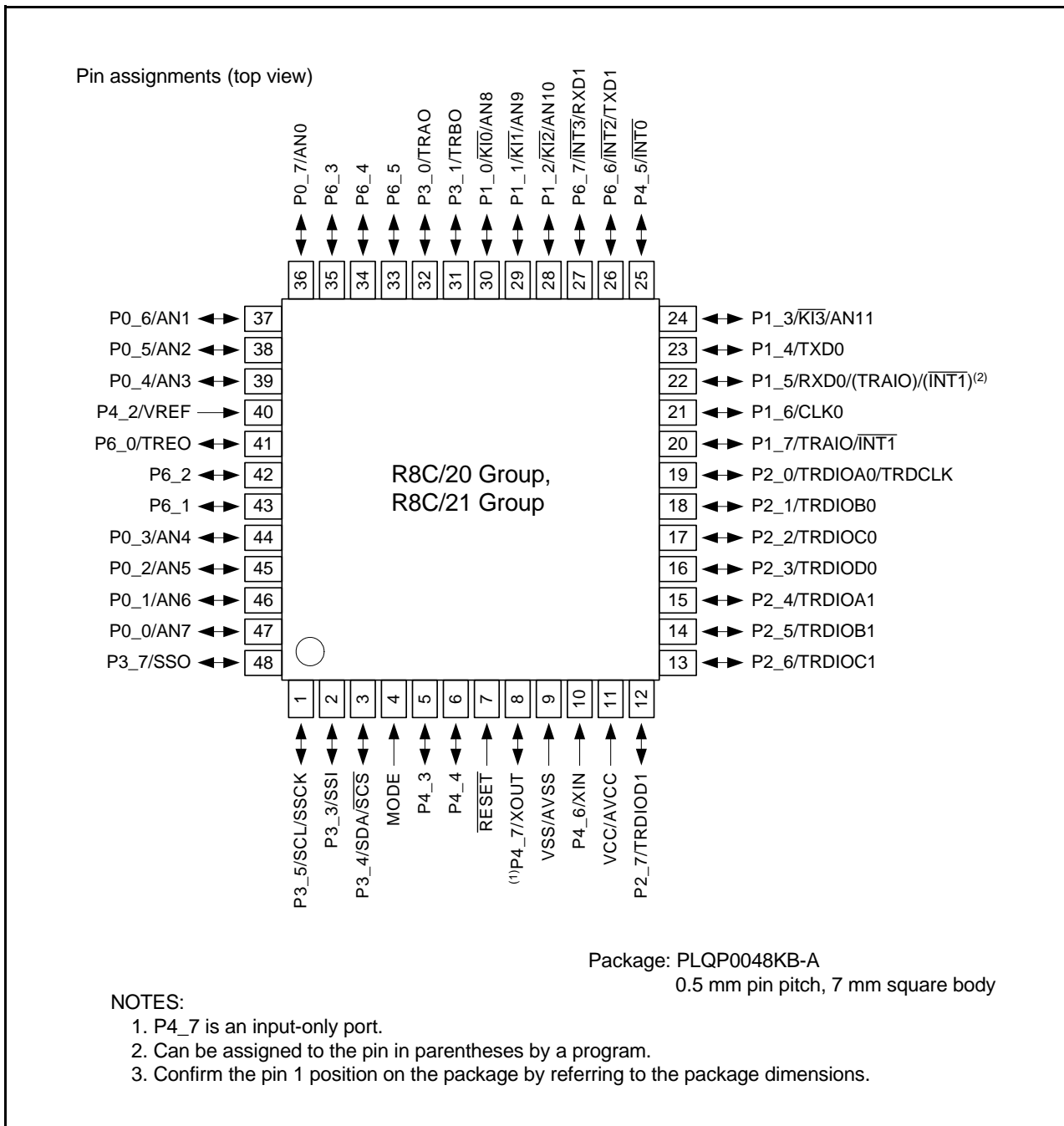
- Do not use addresses 20000h to 23FFFh because these areas are used for the emulator debugger. Refer to **23. Notes on Emulator Debugger**.



**Figure 1.3 Type Number, Memory Size, and Package of R8C/21 Group**

### 1.5 Pin Assignments

Figure 1.4 shows Pin Assignments (Top View).



**Figure 1.4 Pin Assignments (Top View)**

## 1.6 Pin Functions

Table 1.5 lists the Pin Functions and Table 1.6 lists the Pin Name Information by Pin Number.

**Table 1.5 Pin Functions**

Type	Symbol	I/O Type	Description
Power Supply Input	VCC VSS	I	Apply 2.7 V to 5.5 V to the VCC pin. Apply 0 V to the VSS pin.
Analog Power Supply Input	AVCC, AVSS	I	Applies the power supply for the A/D converter. Connect a capacitor between AVCC and AVSS.
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.
XIN Clock Input	XIN	I	These pins are provided for the XIN clock generation circuit I/O. Connect a ceramic resonator or a crystal oscillator between the XIN and XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open.
XIN Clock Output	XOUT	O	
$\overline{\text{INT}}$ Interrupt Input	$\overline{\text{INT0}}$ to $\overline{\text{INT3}}$	I	$\overline{\text{INT}}$ interrupt input pins. $\overline{\text{INT0}}$ Timer RD input pins. $\overline{\text{INT1}}$ Timer RA 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 RD	TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1	I/O	Timer RD I/O ports.
	TRDCLK	I	External clock input pin.
Timer RE	TREO	O	Divided clock output pin.
Serial Interface	CLK0	I/O	Transfer clock I/O pin.
	RXD0, RXD1	I	Serial data input pins.
	TXD0, TXD1	O	Serial data output pins.
I <sup>2</sup> C Bus Interface	SCL	I/O	Clock I/O pin.
	SDA	I/O	Data I/O pin.
Clock Synchronous Serial I/O with Chip Select	SSI	I/O	Data I/O pin.
	SCS	I/O	Chip-select signal I/O pin.
	SSCK	I/O	Clock I/O pin.
	SSO	I/O	Data I/O pin.
Reference Voltage Input	VREF	I	Reference voltage input pin to A/D converter.
A/D Converter	AN0 to AN11	I	Analog input pins to A/D converter.
I/O Port	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0, P3_1, P3_3 to P3_5, P3_7, P4_3 to P4_5, P6_0 to P6_7	I/O	CMOS I/O ports. Each port contains an input/output select direction register, allowing each pin in that port to be directed for input or output individually. Any port set to input can select whether to use a pull-up resistor or not by a program.
Input Port	P4_2, P4_6, P4_7	I	Input only ports.

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



Table 1.6 Pin Name Information by Pin Number

Pin Number	Control Pin	Port	I/O Pin Functions for of Peripheral Modules					
			Interrupt	Timer	Serial Interface	Clock Synchronous Serial I/O with Chip Select	I <sup>2</sup> C Bus Interface	A/D Converter
1		P3_5				SSCK	SCL	
2		P3_3				SSI		
3		P3_4				SCS	SDA	
4	MODE							
5		P4_3						
6		P4_4						
7	RESET							
8	XOUT	P4_7						
9	VSS/AVSS							
10	XIN	P4_6						
11	VCC/AVCC							
12		P2_7		TRDIOD1				
13		P2_6		TRDIOC1				
14		P2_5		TRDIQB1				
15		P2_4		TRDIOA1				
16		P2_3		TRDIOD0				
17		P2_2		TRDIQB0				
18		P2_1		TRDIQB0				
19		P2_0		TRDIOA0/TRDCLK				
20		P1_7	INT1	TRAIO				
21		P1_6			CLK0			
22		P1_5	(INT1) <sup>(1)</sup>	(TRAIO) <sup>(1)</sup>	RXD0			
23		P1_4			TXD0			
24		P1_3	KI3					AN11
25		P4_5	INT0	INT0				
26		P6_6	INT2		TXD1			
27		P6_7	INT3		RXD1			
28		P1_2	KI2					AN10
29		P1_1	KI1					AN9
30		P1_0	KI0					AN8
31		P3_1		TRBO				
32		P3_0		TRAQ				
33		P6_5						
34		P6_4						
35		P6_3						
36		P0_7						AN0
37		P0_6						AN1
38		P0_5						AN2
39		P0_4						AN3
40	VREF	P4_2						
41		P6_0		TREO				
42		P6_2						
43		P6_1						
44		P0_3						AN4
45		P0_2						AN5
46		P0_1						AN6
47		P0_0						AN7
48		P3_7				SSO		

NOTE:

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

## 2. Central Processing Unit (CPU)

Figure 2.1 shows the CPU Registers. The CPU contains 13 registers. Of these, R0, R1, R2, R3, A0, A1, and FB comprise a register bank. Two sets of register banks are provided.

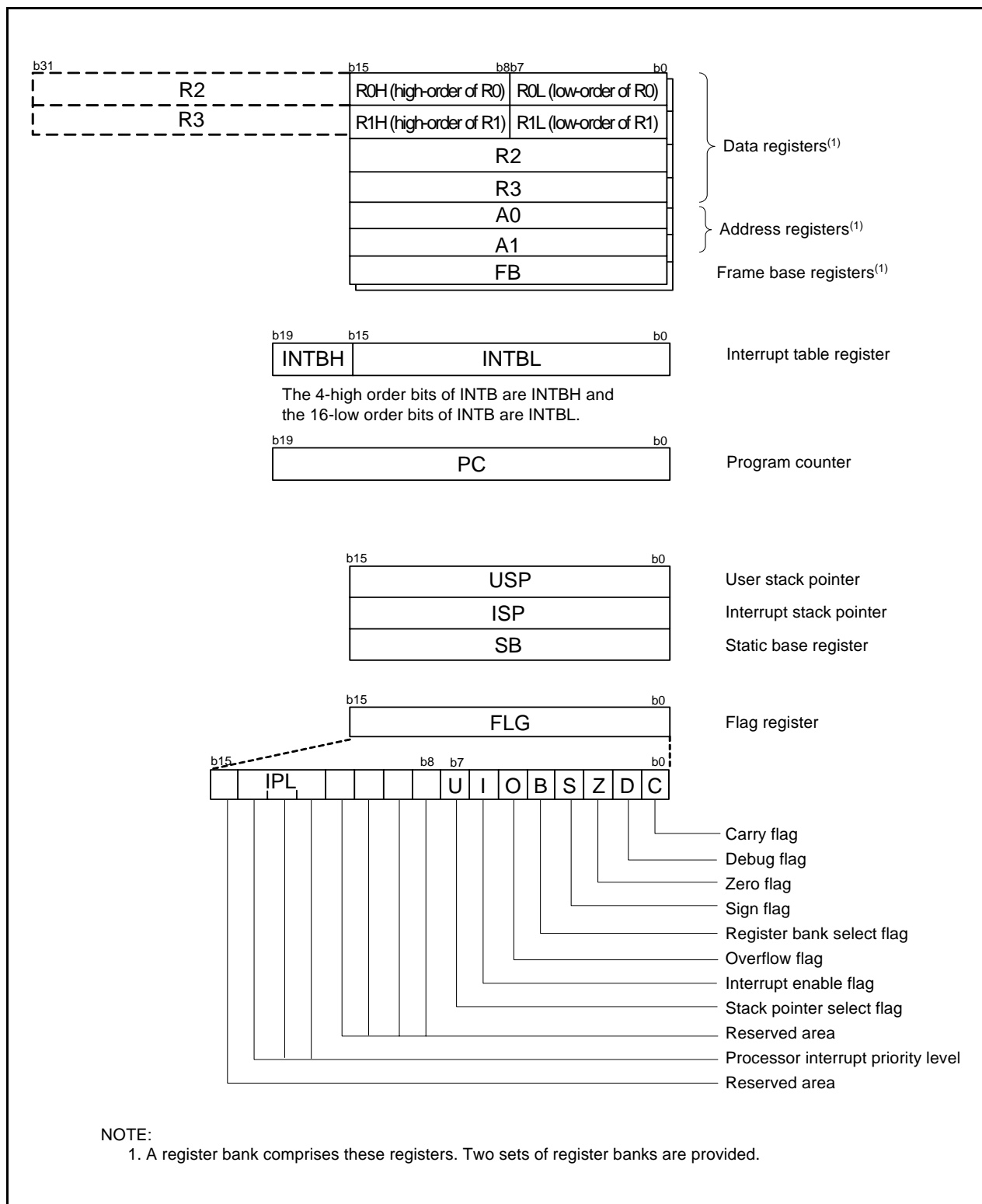


Figure 2.1 CPU Registers

## 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 bit (R0H) and low-order bit (R0L) to be used separately as 8-bit data registers. The same applies to R1H and R1L as R0H and R0L. R2 can be combined with R0 to be used as a 32-bit data register (R2R0). The same applies R3R1 as R2R0.

## 2.2 Address Registers (A0 and A1)

A0 is a 16-bit register for address register indirect addressing and address register relative addressing. They also are used for transfer, arithmetic and logic operations. The same applies to A1 as A0.

A1 can be combined with A0 to be used 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, a 20-bit register, indicates the start address of an interrupt vector table.

## 2.5 Program Counter (PC)

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

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

The stack pointer (SP), USP and ISP, are 16 bits wide each.

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 a 11-bit register indicating the CPU status.

### 2.8.1 Carry Flag (C)

The C flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic and logic unit.

### 2.8.2 Debug Flag (D)

The D flag is for debug only. Set to 0.

### 2.8.3 Zero Flag (Z)

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

### 2.8.4 Sign Flag (S)

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

### 2.8.5 Register Bank Select Flag (B)

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

### 2.8.6 Overflow Flag (O)

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

### 2.8.7 Interrupt Enable Flag (I)

The I flag enables a maskable interrupt.

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

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

### 2.8.10 Reserved Bit

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

## 3. Memory

### 3.1 R8C/20 Group

Figure 3.1 shows a Memory Map of R8C/20 Group. The R8C/20 Group has 1 Mbyte of address space from address 00000h to FFFFFFFh.

The internal ROM is allocated lower addresses, beginning with address 0FFFFh. For example, a 48-Kbyte internal ROM is allocated addresses 04000h 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 2.5-Kbyte internal RAM is allocated addresses 00400h to 00DFFh. 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 (SFR) 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 user and cannot be accessed by users.

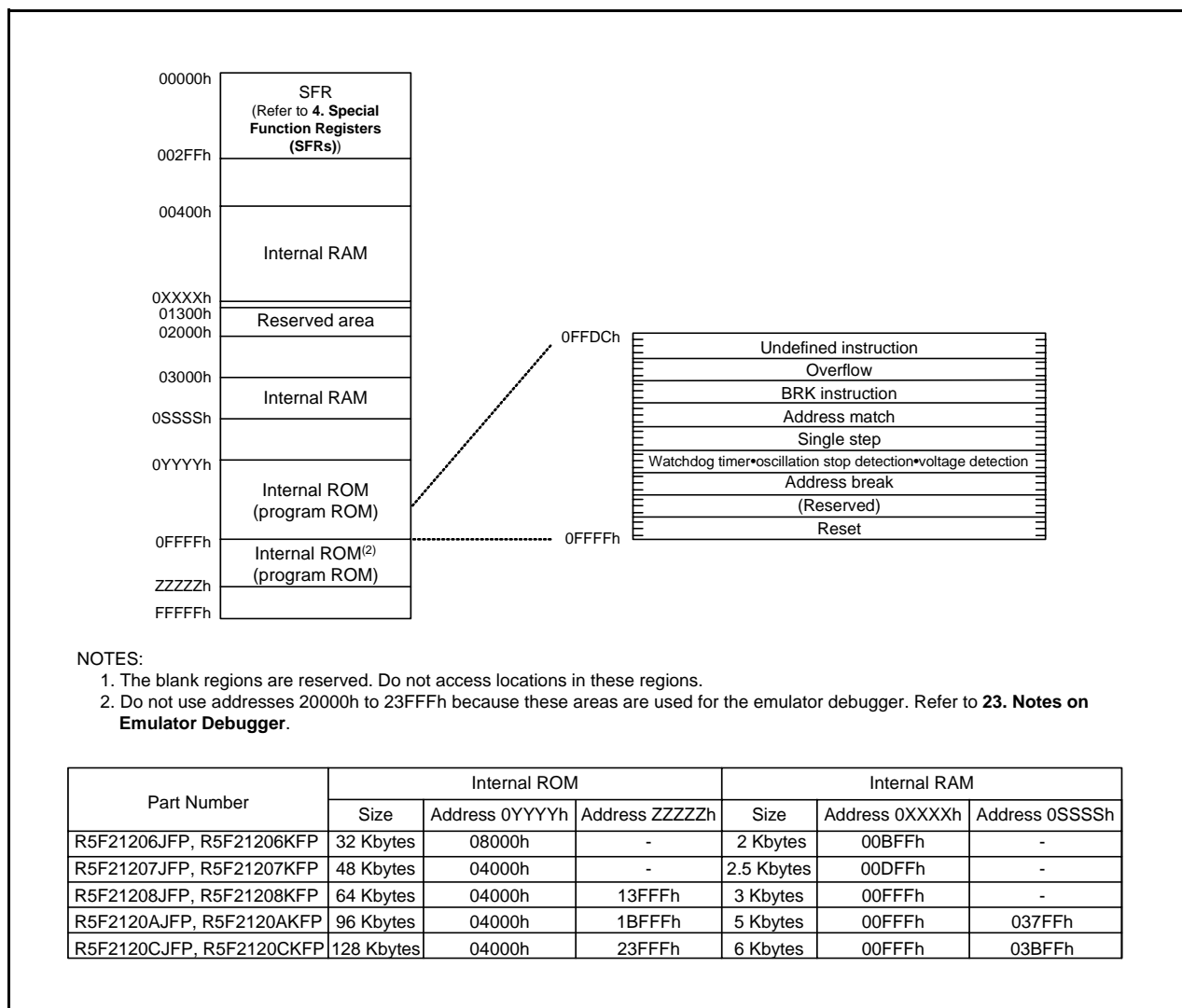


Figure 3.1 Memory Map of R8C/20 Group

### 3.2 R8C/21 Group

Figure 3.2 shows a Memory Map of R8C/21 Group. The R8C/21 Group has 1 Mbyte of address space from address 00000h to FFFFFFFh.

The internal ROM (program ROM) is allocated lower addresses, beginning with address 0FFFFh. For example, a 48-Kbyte internal ROM is allocated addresses 04000h to 0FFFFh.

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

The internal ROM (data flash) is allocated addresses 02400h to 02BFFh.

The internal RAM is allocated higher addresses, beginning with address 00400h. For example, a 2.5-Kbyte internal RAM is allocated addresses 00400h to 00DFFh. 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 (SFR) are allocated addresses 00000h to 002FFh. The peripheral function control registers are allocated them. All addresses within the SFR, which have nothing allocated are reserved for future use and cannot be accessed by users.

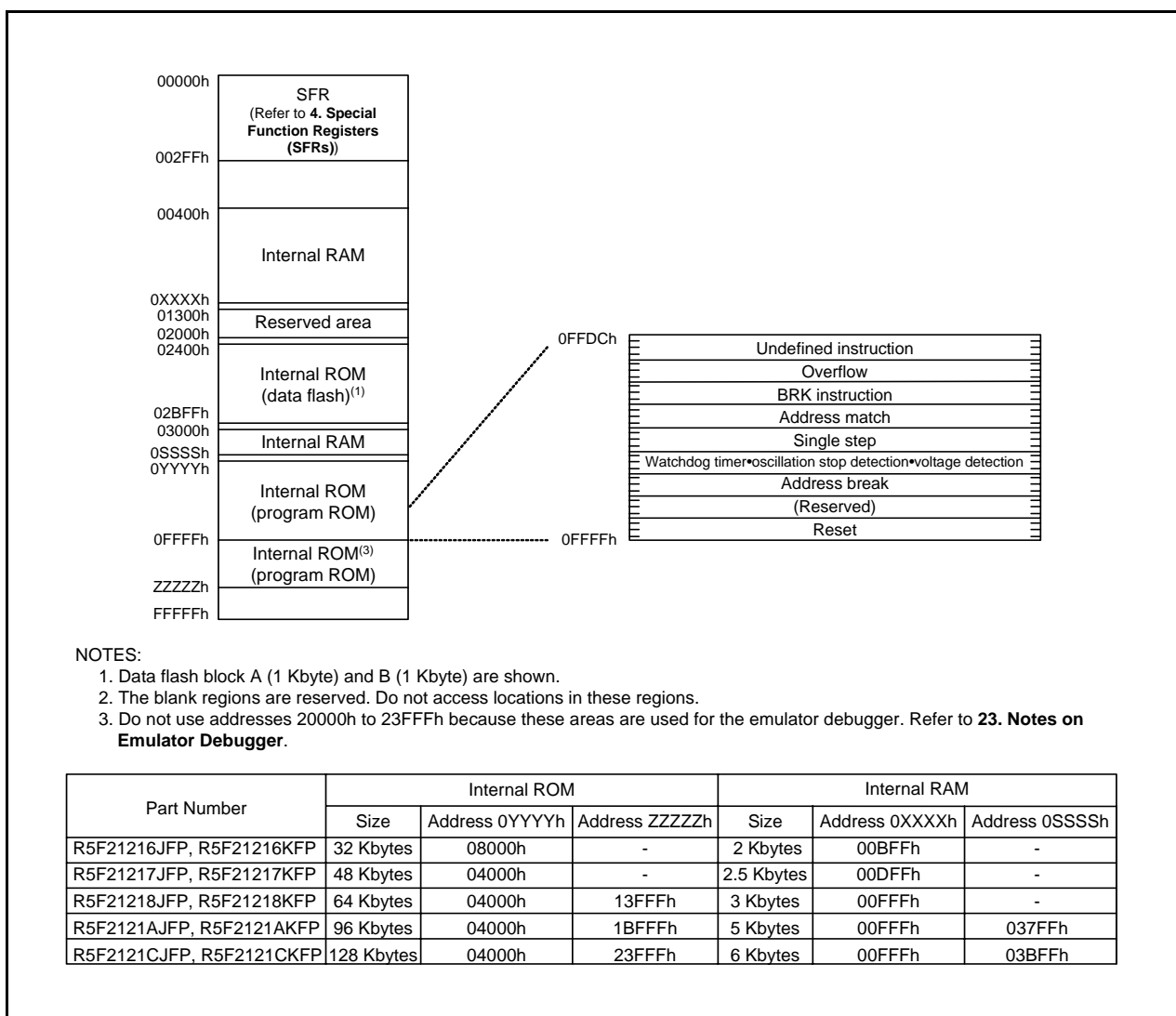


Figure 3.2 Memory Map of R8C/21 Group

## 4. Special Function Registers (SFRs)

An SFR (special function register) is a control register for a peripheral function. Table 4.1 to Table 4.6 list the SFR Information.

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

Address	Register	Symbol	After reset
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	00h
0005h	Processor Mode Register 1	PM1	00h
0006h	System Clock Control Register 0	CM0	01101000b
0007h	System Clock Control Register 1	CM1	00100000b
0008h			
0009h			
000Ah	Protect Register	PRCR	00h
000Bh			
000Ch	Oscillation Stop Detection Register	OCD	00000100b
000Dh	Watchdog Timer Reset Register	WDTR	XXh
000Eh	Watchdog Timer Start Register	WDTS	XXh
000Fh	Watchdog Timer Control Register	WDC	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			00h
0018h			
0019h			
001Ah			
001Bh			
001Ch	Count Source Protect Mode Register	CSPR	00h 10000000b <sup>(8)</sup>
001Dh			
001Eh			
001Fh			
0020h			
0021h			
0022h			
0023h	High-Speed On-Chip Oscillator Control Register 0	FRA0	00h
0024h	High-Speed On-Chip Oscillator Control Register 1	FRA1	When shipping
0025h	High-Speed On-Chip Oscillator Control Register 2	FRA2	00h
0026h			
0030h			
0031h	Voltage Detection Register 1 <sup>(2)</sup>	VCA1	00001000b
0032h	Voltage Detection Register 2 <sup>(6)</sup>	VCA2	00h <sup>(3)</sup> 01000000b <sup>(4)</sup>
0033h			
0034h			
0035h			
0036h	Voltage Monitor 1 Circuit Control Register <sup>(7)</sup>	VW1C	0000X000b <sup>(3)</sup> 0100X001b <sup>(4)</sup>
0037h	Voltage Monitor 2 Circuit Control Register <sup>(5)</sup>	VW2C	00h
0038h			
0039h			
003Fh			

X: Undefined

NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. Software reset, watchdog timer reset, and voltage monitor 2 reset do not affect this register.
3. The LVD0ON bit in the OFS register is set to 1.
4. Power-on reset, voltage monitor 1 reset or the LVD0ON bit in the OFS register is set to 0.
5. Software reset, watchdog timer reset, and voltage monitor 2 reset do not affect b2 and b3.
6. Software reset, watchdog timer reset, and voltage monitor 2 reset do not affect b7.
7. Software reset, the watchdog timer reset, and the voltage monitor 2 reset do not affect other than the b0 and b6.
8. The CSPROINI bit in the OFS register is 0.

**Table 4.2 SFR Information (2)<sup>(1)</sup>**

Address	Register	Symbol	After reset
0040h			
0041h			
0042h			
0043h			
0044h			
0045h			
0046h			
0047h			
0048h	Timer RD0 Interrupt Control Register	TRD0IC	XXXXX000b
0049h	Timer RD1 Interrupt Control Register	TRD1IC	XXXXX000b
004Ah	Timer RE Interrupt Control Register	TREIC	XXXXX000b
004Bh			
004Ch			
004Dh	Key Input Interrupt Control Register	KUPIC	XXXXX000b
004Eh	A/D Conversion Interrupt Control Register	ADIC	XXXXX000b
004Fh	SSU Interrupt Control Register/IIC Bus Interrupt Control Register <sup>(2)</sup>	SSUIC/IICIC	XXXXX000b
0050h			
0051h	UART0 Transmit Interrupt Control Register	S0TIC	XXXXX000b
0052h	UART0 Receive Interrupt Control Register	S0RIC	XXXXX000b
0053h	UART1 Transmit Interrupt Control Register	S1TIC	XXXXX000b
0054h	UART1 Receive Interrupt Control Register	S1RIC	XXXXX000b
0055h	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	INT3 Interrupt Control Register	INT3IC	XX00X000b
005Bh			
005Ch			
005Dh	INT0 Interrupt Control Register	INT0IC	XX00X000b
005Eh			
005Fh			
0060h			
0061h			
0062h			
0063h			
0064h			
0065h			
0066h			
0067h			
0068h			
0069h			
006Ah			
006Bh			
006Ch			
006Dh			
006Eh			
006Fh			
0070h			
0071h			
0072h			
0073h			
0074h			
0075h			
0076h			
0077h			
0078h			
0079h			
007Ah			
007Bh			
007Ch			
007Dh			
007Eh			
007Fh			

X: Undefined

## NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. Selected by the IICSEL bit in the PMR register.



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

Address	Register	Symbol	After reset
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	UART1 Transmit/Receive Mode Register	U1MR	00h
00A9h	UART1 Bit Rate Register	U1BRG	XXh
00AAh	UART1 Transmit Buffer Register	U1TB	XXh
00ABh			XXh
00ACh	UART1 Transmit/Receive Control Register 0	U1C0	00001000b
00ADh	UART1 Transmit/Receive Control Register 1	U1C1	00000010b
00AEh	UART1 Receive Buffer Register	U1RB	XXh
00AFh			XXh
00B0h			
00B1h			
00B2h			
00B3h			
00B4h			
00B5h			
00B6h			
00B7h			
00B8h	SS Control Register H/IIC Bus Control Register 1 <sup>(2)</sup>	SSCRH/ICCR1	00h
00B9h	SS Control Register L/IIC Bus Control Register 2 <sup>(2)</sup>	SSCRL/ICCR2	01111101b
00BAh	SS Mode Register/IIC Bus Mode Register 1 <sup>(2)</sup>	SSMR/ICMR	00011000b
00BBh	SS Enable Register/IIC Bus Interrupt Enable Register <sup>(2)</sup>	SSER/ICIER	00h
00BCh	SS Status Register/IIC Bus Status Register <sup>(2)</sup>	SSSR/ICSR	00h/0000X000b
00BDh	SS Mode Register 2/Slave Address Register <sup>(2)</sup>	SSMR2/SAR	00h
00BEh	SS Transmit Data Register/IIC Bus Transmit Data Register <sup>(2)</sup>	SSTDR/ICDRT	FFh
00BFh	SS Receive Data Register/IIC Bus Receive Data Register <sup>(2)</sup>	SSRDR/ICDRR	FFh

X: Undefined

NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. Selected by the IICSEL bit in the PMR register.

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

Address	Register	Symbol	After reset
00C0h	A/D Register	AD	XXh
00C1h			XXh
00C2h			
00C3h			
00C4h			
00C5h			
00C6h			
00C7h			
00C8h			
00C9h			
00CAh			
00CBh			
00CCh			
00CDh			
00CEh			
00CFh			
00D0h			
00D1h			
00D2h			
00D3h			
00D4h	A/D Control Register 2	ADCON2	00h
00D5h			
00D6h	A/D Control Register 0	ADCON0	00h
00D7h	A/D Control Register 1	ADCON1	00h
00D8h			
00D9h			
00DAh			
00DBh			
00DCh			
00DDh			
00DEh			
00DFh			
00E0h	Port P0 Register	P0	XXh
00E1h	Port P1 Register	P1	XXh
00E2h	Port P0 Direction Register	PD0	00h
00E3h	Port P1 Direction Register	PD1	00h
00E4h	Port P2 Register	P2	XXh
00E5h	Port P3 Register	P3	XXh
00E6h	Port P2 Direction Register	PD2	00h
00E7h	Port P3 Direction Register	PD3	00h
00E8h	Port P4 Register	P4	XXh
00E9h			
00EAh	Port P4 Direction Register	PD4	00h
00EBh			
00ECh	Port P6 Register	P6	XXh
00EDh			
00EEh	Port P6 Direction Register	PD6	00h
00EFh			
00F0h			
00F1h			
00F2h			
00F3h			
00F4h			
00F5h	UART1 Function Select Register	U1SR	XXh
00F6h			
00F7h			
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	XX00XX00b
00FEh			
00FFh			

X: Undefined

NOTE:

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

**Table 4.5 SFR Information (5)<sup>(1)</sup>**

Address	Register	Symbol	After reset
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	TRBPRES	FFh
010Dh	Timer RB Secondary Register	TRBSC	FFh
010Eh	Timer RB Primary	TRBPR	FFh
010Fh			
0110h			
0111h			
0112h			
0113h			
0114h			
0115h			
0116h			
0117h			
0118h	Timer RE Counter Data Register	TRESEC	00h
0119h	Timer RE Compare Data Register	TREMIN	00h
011Ah			
011Bh			
011Ch	Timer RE Control Register 1	TRECR1	00h
011Dh	Timer RE Control Register 2	TRECR2	00h
011Eh	Timer RE Count Source Select Register	TRECSR	00001000b
011Fh			
0120h			
0121h			
0122h			
0123h			
0124h			
0125h			
0126h			
0127h			
0128h			
0129h			
012Ah			
012Bh			
012Ch			
012Dh			
012Eh			
012Fh			
0130h			
0131h			
0132h			
0133h			
0134h			
0135h			
0136h			
0137h	Timer RD Start Register	TRDSTR	11111100b
0138h	Timer RD Mode Register	TRDMR	00001110b
0139h	Timer RD PWM Mode Register	TRDPMR	10001000b
013Ah	Timer RD Function Control Register	TRDFCR	10000000b
013Bh	Timer RD Output Master Enable Register 1	TRDOER1	FFh
013Ch	Timer RD Output Master Enable Register 2	TRDOER2	01111111b
013Dh	Timer RD Output Control Register	TRDOCR	00h
013Eh	Timer RD Digital Filter Function Select Register 0	TRDDF0	00h
013Fh	Timer RD Digital Filter Function Select Register 1	TRDDF1	00h

X: Undefined

NOTE:

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

**Table 4.6 SFR Information (6)<sup>(1)</sup>**

Address	Register	Symbol	After reset
0140h	Timer RD Control Register 0	TRDCR0	00h
0141h	Timer RD I/O Control Register A0	TRDIORA0	10001000b
0142h	Timer RD I/O Control Register C0	TRDIORC0	10001000b
0143h	Timer RD Status Register 0	TRDSR0	11100000b
0144h	Timer RD Interrupt Enable Register 0	TRDIER0	11100000b
0145h	Timer RD PWM Mode Output Level Control Register 0	TRDPOCR0	11111000b
0146h	Timer RD Counter 0	TRD0	00h
0147h			00h
0148h	Timer RD General Register A0	TRDGRA0	FFh
0149h			FFh
014Ah	Timer RD General Register B0	TRDGRB0	FFh
014Bh			FFh
014Ch	Timer RD General Register C0	TRDGRC0	FFh
014Dh			FFh
014Eh	Timer RD General Register D0	TRDGRD0	FFh
014Fh			FFh
0150h	Timer RD Control Register 1	TRDCR1	00h
0151h	Timer RD I/O Control Register A1	TRDIORA1	10001000b
0152h	Timer RD I/O Control Register C1	TRDIORC1	10001000b
0153h	Timer RD Status Register 1	TRDSR1	11000000b
0154h	Timer RD Interrupt Enable Register 1	TRDIER1	11100000b
0155h	Timer RD PWM Mode Output Level Control Register 1	TRDPOCR1	11111000b
0156h	Timer RD Counter 1	TRD1	00h
0157h			00h
0158h	Timer RD General Register A1	TRDGRA1	FFh
0159h			FFh
015Ah	Timer RD General Register B1	TRDGRB1	FFh
015Bh			FFh
015Ch	Timer RD General Register C1	TRDGRC1	FFh
015Dh			FFh
015Eh	Timer RD General Register D1	TRDGRD1	FFh
015Fh			FFh
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			
FFFFh	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

There are resets: hardware reset, power-on reset, voltage monitor 1 reset, voltage monitor 2 reset, watchdog timer reset, and software reset.

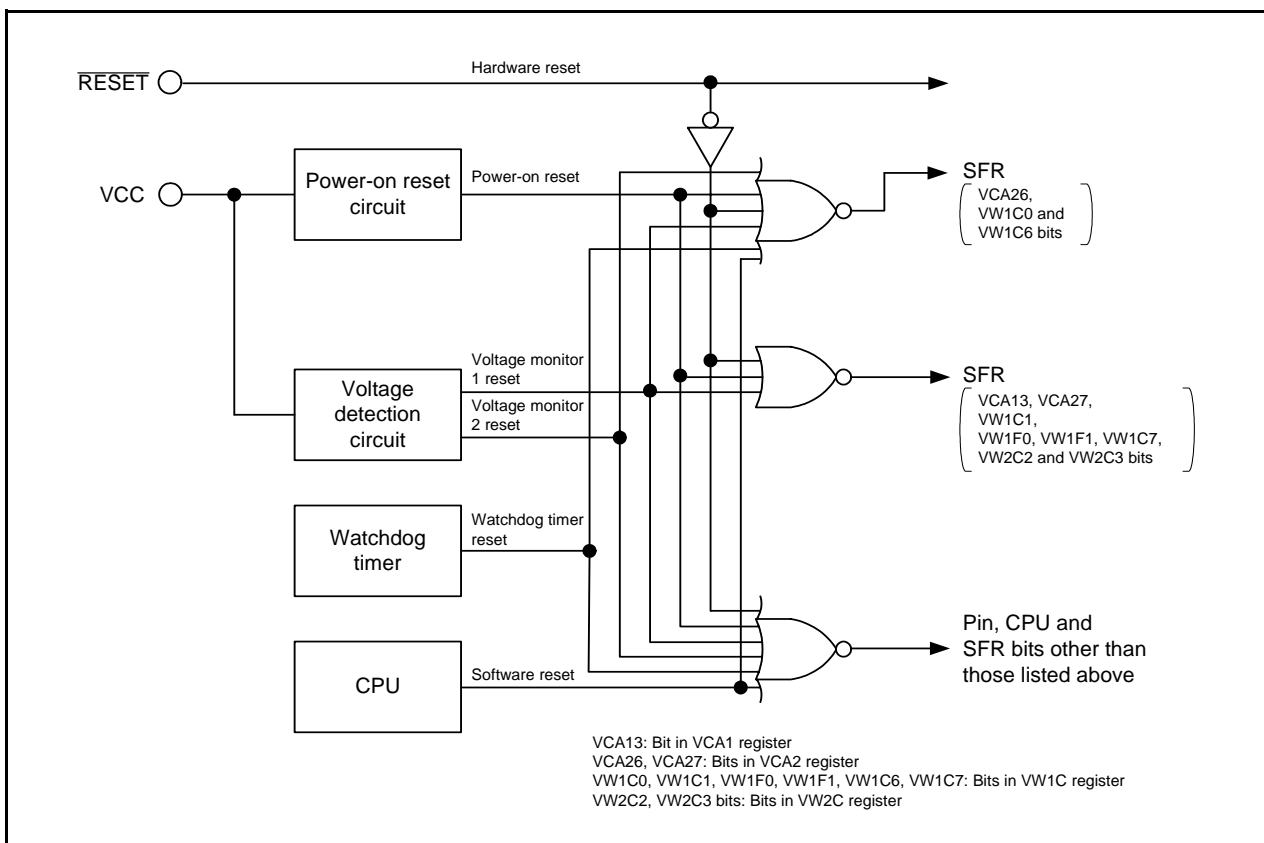
Table 5.1 lists the Reset Names and Sources.

**Table 5.1 Reset Names and Sources**

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

NOTE:

1. Because this product is under development, specifications may be changed.

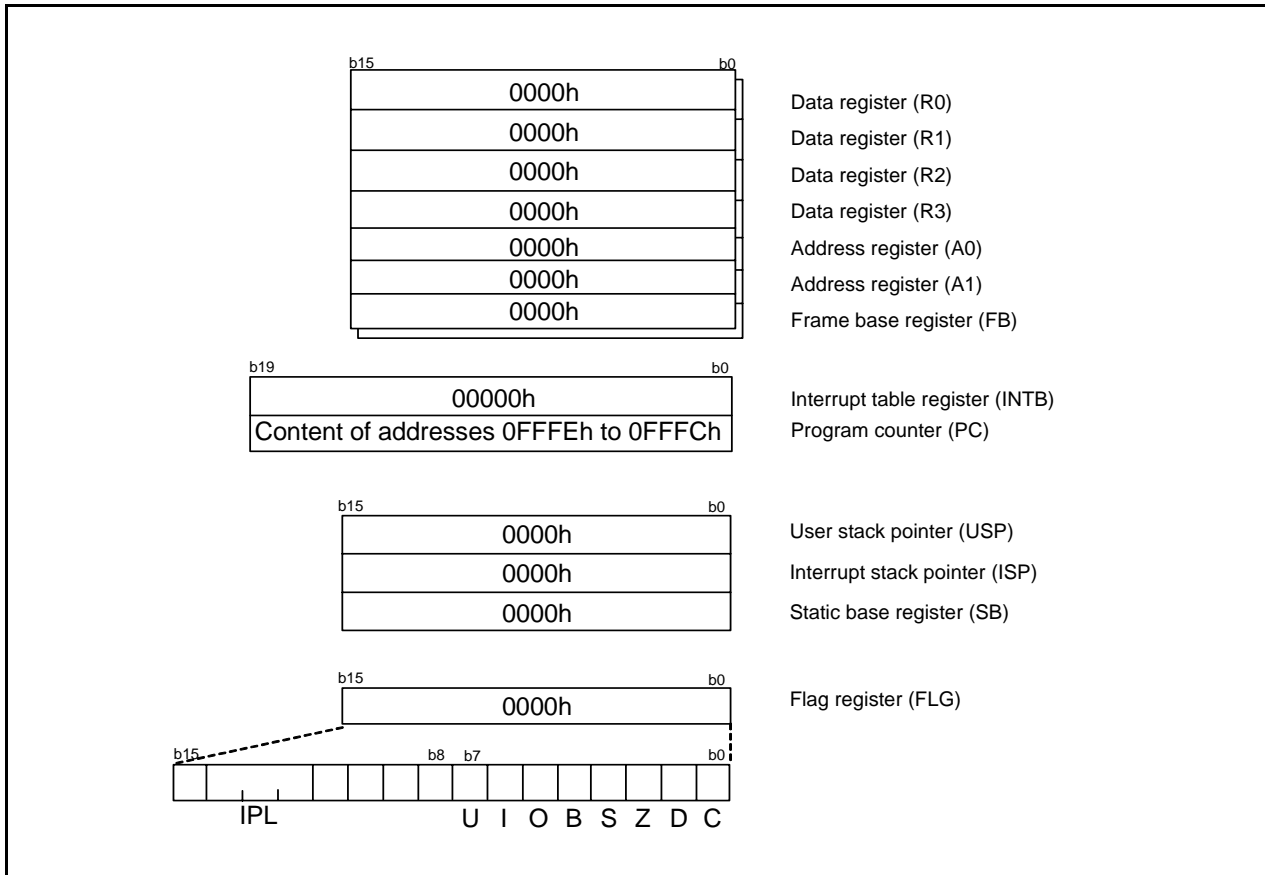


**Figure 5.1 Block Diagram of Reset Circuit**

Table 5.2 lists the Pin Functions after Reset, Figure 5.2 shows CPU Register Status after Reset, Figure 5.3 shows Reset Sequence, and Figure 5.4 shows the OFS Register.

**Table 5.2 Pin Functions after Reset**

Pin Name	Pin Functions
P0, P1, P2	Input port
P3_0, P3_1, P3_3 to P3_5, P3_7	Input port
P4_2 to P4_7	Input port
P6	Input port



**Figure 5.2 CPU Register Status after Reset**

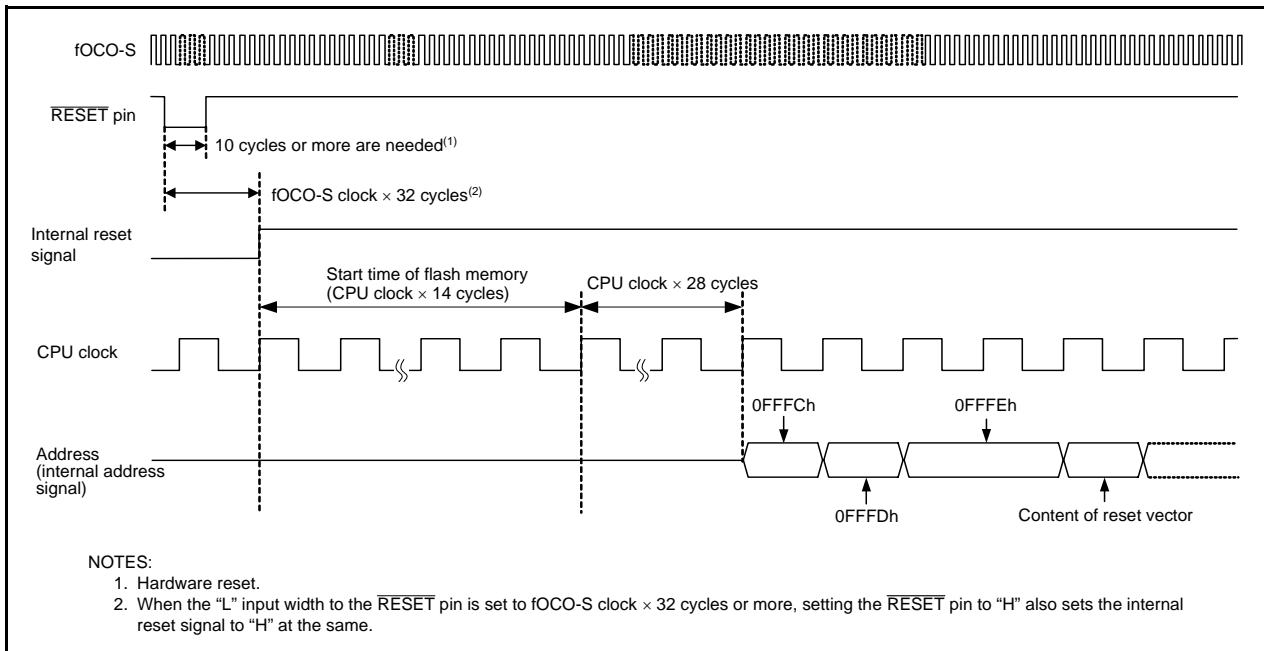


Figure 5.3 Reset Sequence

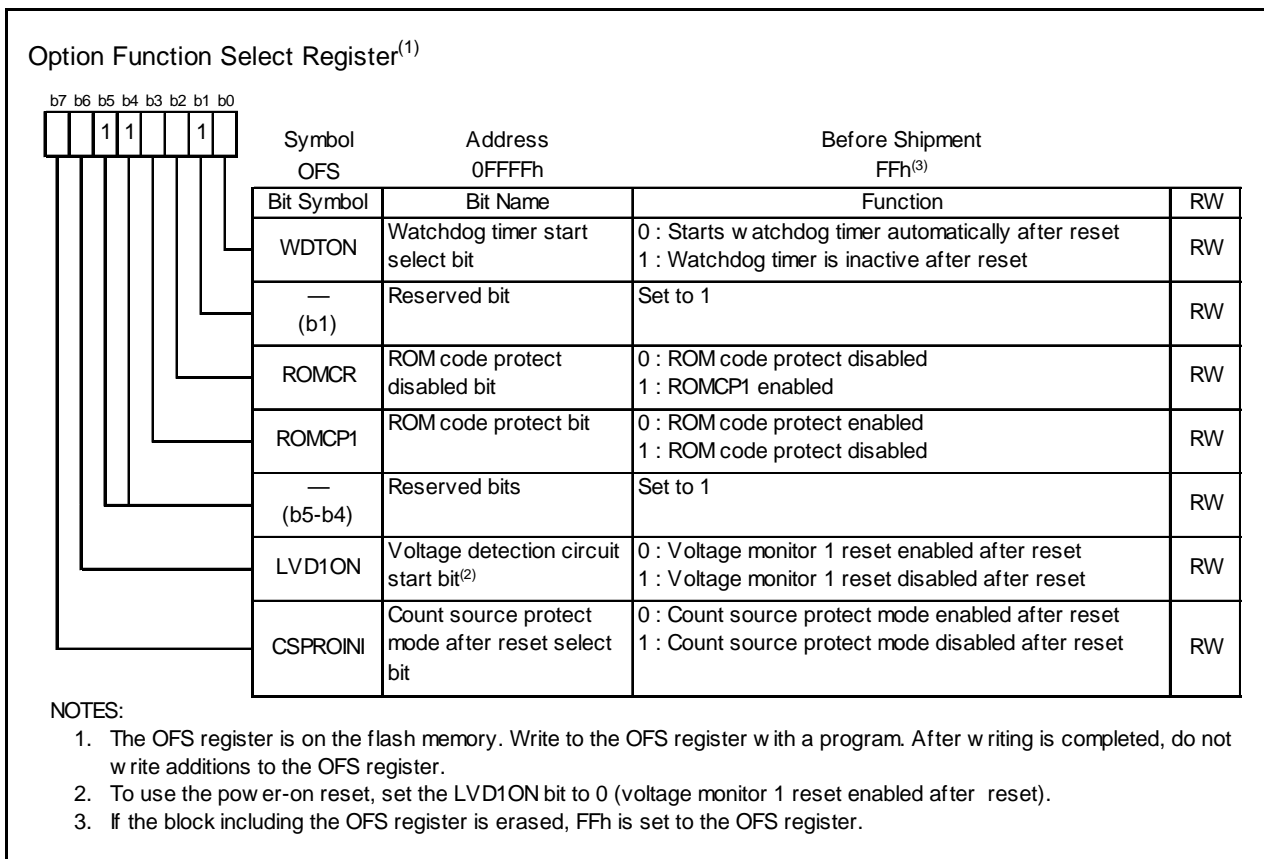


Figure 5.4 OFS Register

## 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 power supply voltage meets the recommended performance condition, the pins, CPU, and SFR are reset (refer to **Table 5.2 Pin Functions after Reset**). When the input level applied to the  $\overline{\text{RESET}}$  pin changes “L” to “H”, the 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 for the CPU clock.

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

The internal RAM is not reset. If the  $\overline{\text{RESET}}$  pin is pulled “L” during writing to the internal RAM, the internal RAM will be in indeterminate state.

Figure 5.5 shows the Example of Hardware Reset Circuit and Operation and Figure 5.6 shows the 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 $\mu$ s or more.
- (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 power supply voltage increase until it meets the recommended performance condition.
- (3) Wait for  $t_{d(P-R)}$  or more to allow the internal power supply to stabilize (refer to **20. Electrical Characteristics**).
- (4) Wait for 10 $\mu$ s or more.
- (5) Apply “H” to the  $\overline{\text{RESET}}$  pin.



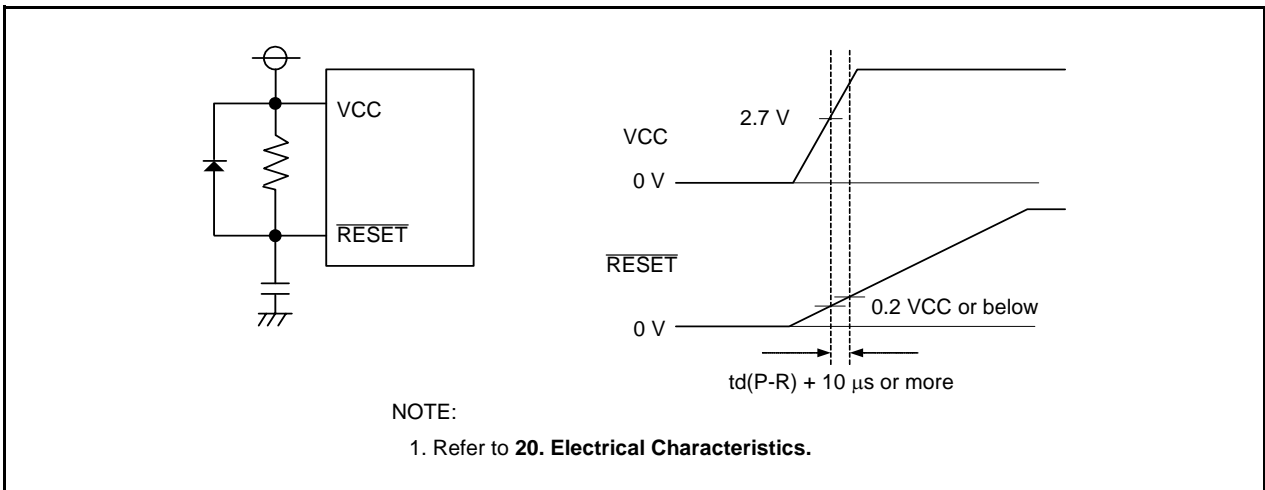


Figure 5.5 Example of Hardware Reset Circuit and Operation

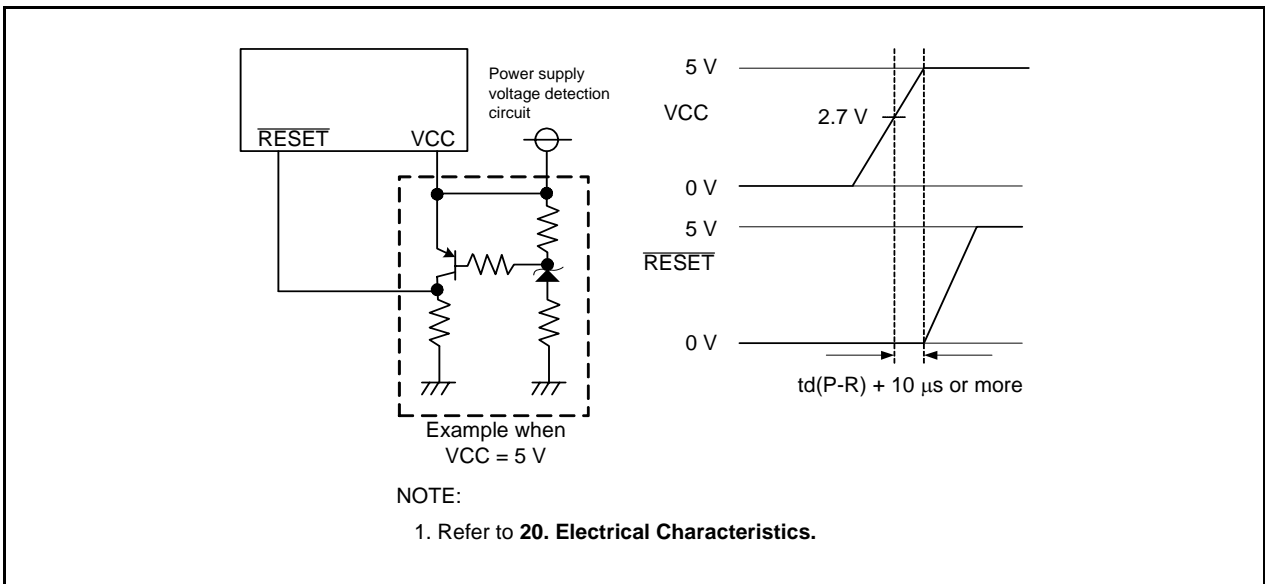


Figure 5.6 Example of Hardware Reset Circuit (Usage Example of External Supply Voltage Detection Circuit) and Operation

## 5.2 Power-On Reset Function(1)

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

When the input voltage to the VCC pin reaches to the Vdet0 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 divide-by-8 is automatically selected for the CPU after reset.

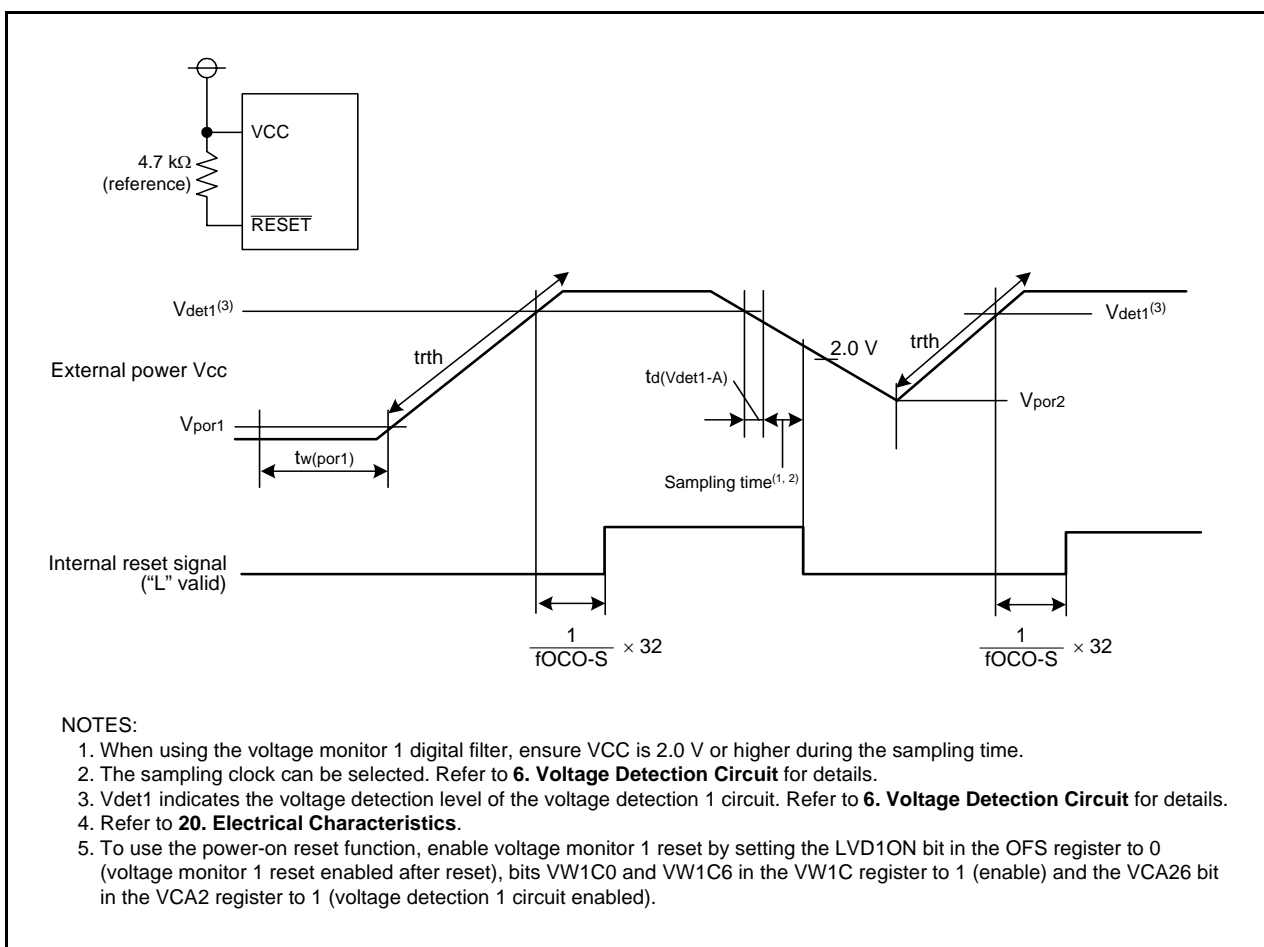
Refer to **4. Special Function Registers (SFRs)** for the status of the SFR after power-on reset.

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

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

### NOTE:

1. When using power-on reset function, set the LVD1ON bit to 0 (voltage monitor 1 reset enabled after reset).



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

### 5.3 Voltage Monitor 1 Reset

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

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

And when the input voltage to the VCC pin reaches to the Vdet1 level or above, count operation of the low-speed on-chip oscillator clock starts. When the operation counts the low-speed on-chip oscillator clock for 32 times, 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 divide-by-8 is automatically selected for the CPU after reset.

The LVD1ON bit in the OFS register can select to enable or disable voltage monitor 1 reset after a reset.

To use the power-on reset function, enable voltage monitor 1 reset by setting the LVD1ON bit in the OFS register to 0, bits VW1C0 and VW1C6 in the VW1C register to 1, the VCA bit in the VCA2 register to 1.

The LVD1ON bit cannot be changed by a program. When setting the LVD1ON bit, write 0 (voltage monitor 1 reset enabled after reset) or 1 (voltage monitor 1 reset disabled after reset) to the bit 6 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 1 reset.

The internal RAM is not reset. When the input voltage to the VCC pin reaches to the Vdet1 level or below during writing to the internal RAM, the internal RAM is in indeterminate state.

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

### 5.4 Voltage Monitor 2 Reset

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

When the input voltage to the VCC pin drops to the Vdet2 level or below, the pins, CPU, and SFR are reset and the program is executed beginning with the address indicated by the reset vector. After reset, the low-speed on-chip oscillator clock divide-by-8 is automatically selected for 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 to the Vdet2 level or below during writing to the internal RAM, the internal RAM is in indeterminate state.

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

### 5.5 Watchdog Timer Reset

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

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

### 5.6 Software Reset

When the PM03 bit in the PM0 register is set to 1 (MCU reset), the MCU resets its pins, CPU, and SFR. The program is executed beginning with the address indicated by the reset vector. After reset, the low-speed on-chip oscillator clock divide-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 is a circuit to monitor the input voltage to the VCC pin. This circuit monitors the VCC input voltage by the program. And the voltage monitor 1 reset, voltage monitor 2 interrupt and voltage monitor 2 reset can be used.

Table 6.1 lists the Specifications of Voltage Detection Circuit and Figures 6.1 to 6.3 show the Block Diagrams. Figures 6.4 to 6.6 show the Associated Registers.

**Table 6.1 Specifications of Voltage Detection Circuit**

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

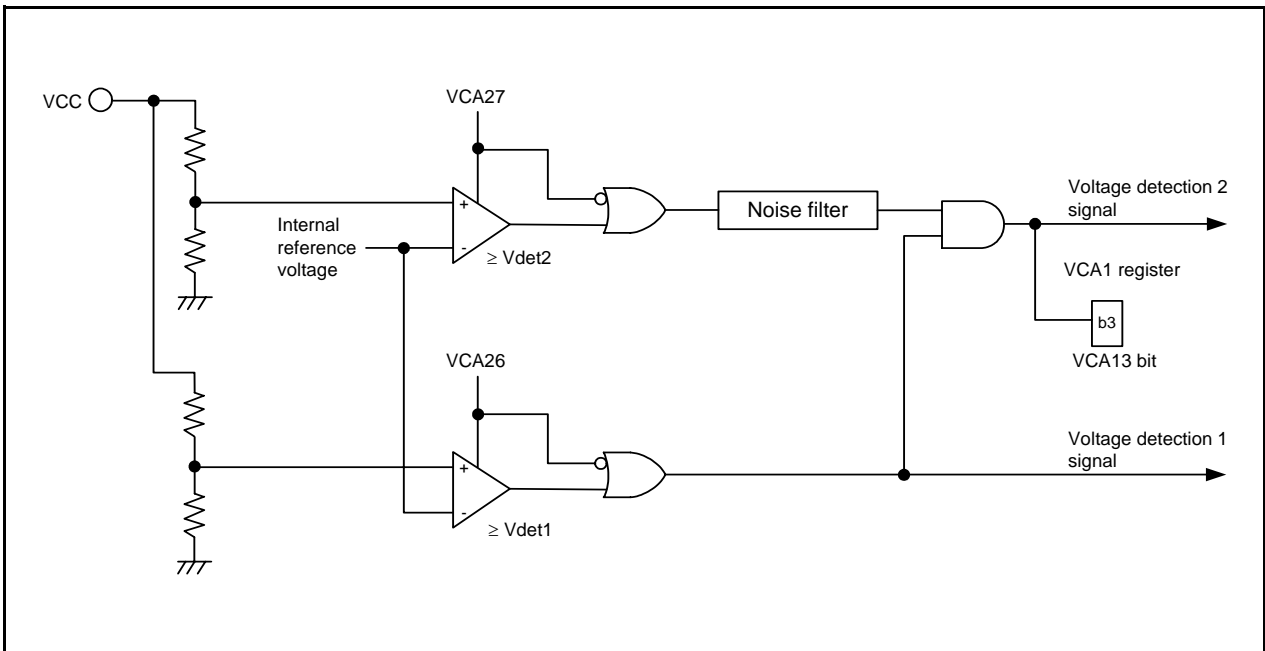


Figure 6.1 Block Diagram of Voltage Detection Circuit

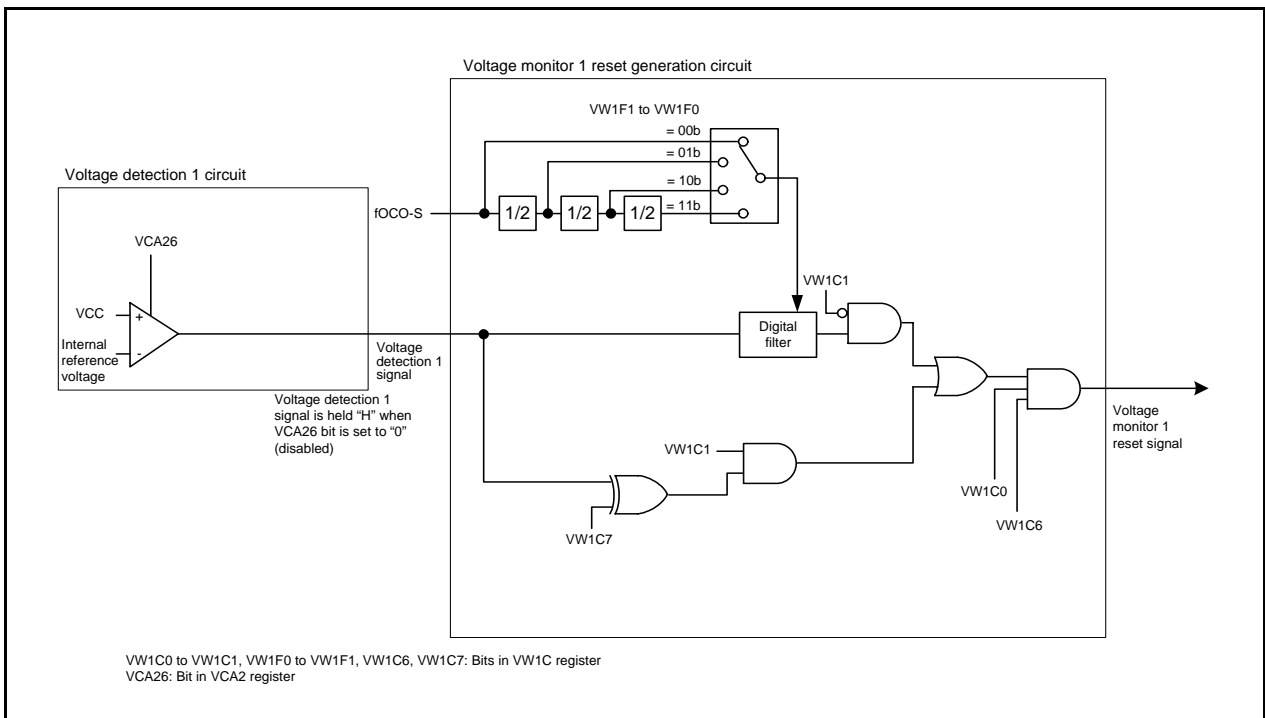
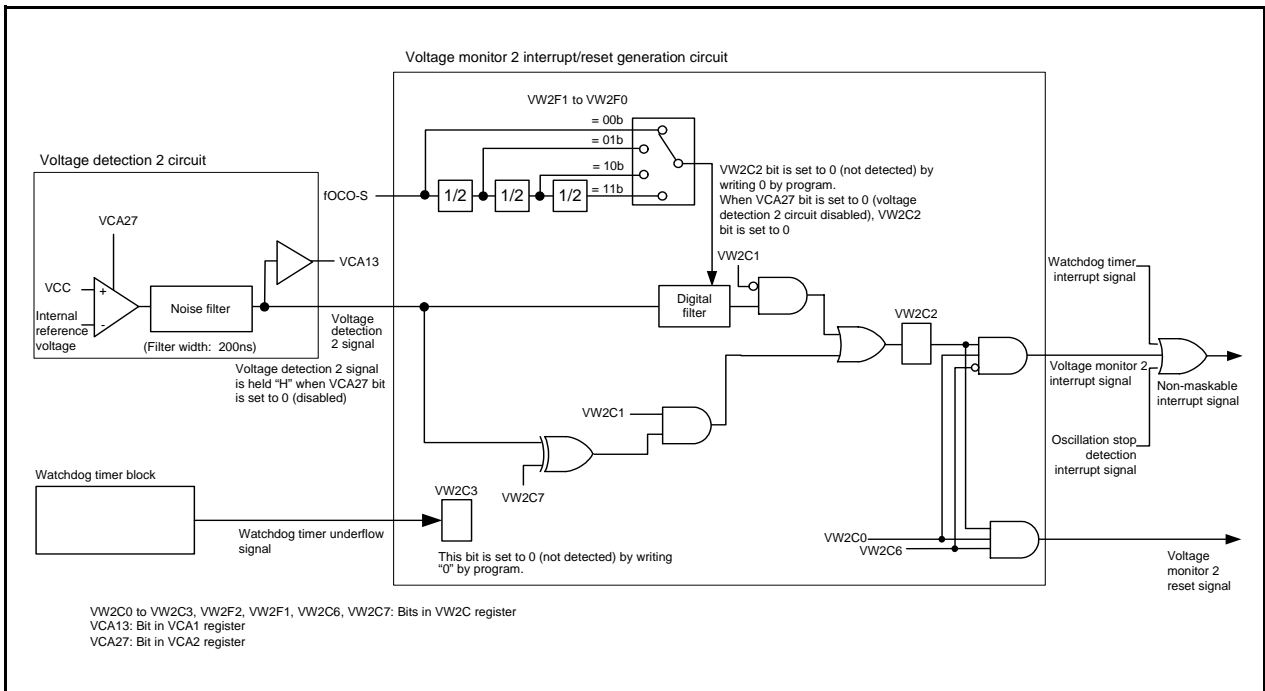


Figure 6.2 Block Diagram of Voltage Monitor 1 Reset Generation Circuit



**Figure 6.3 Block Diagram of Voltage Monitor 2 Interrupt/Reset Generation Circuit**

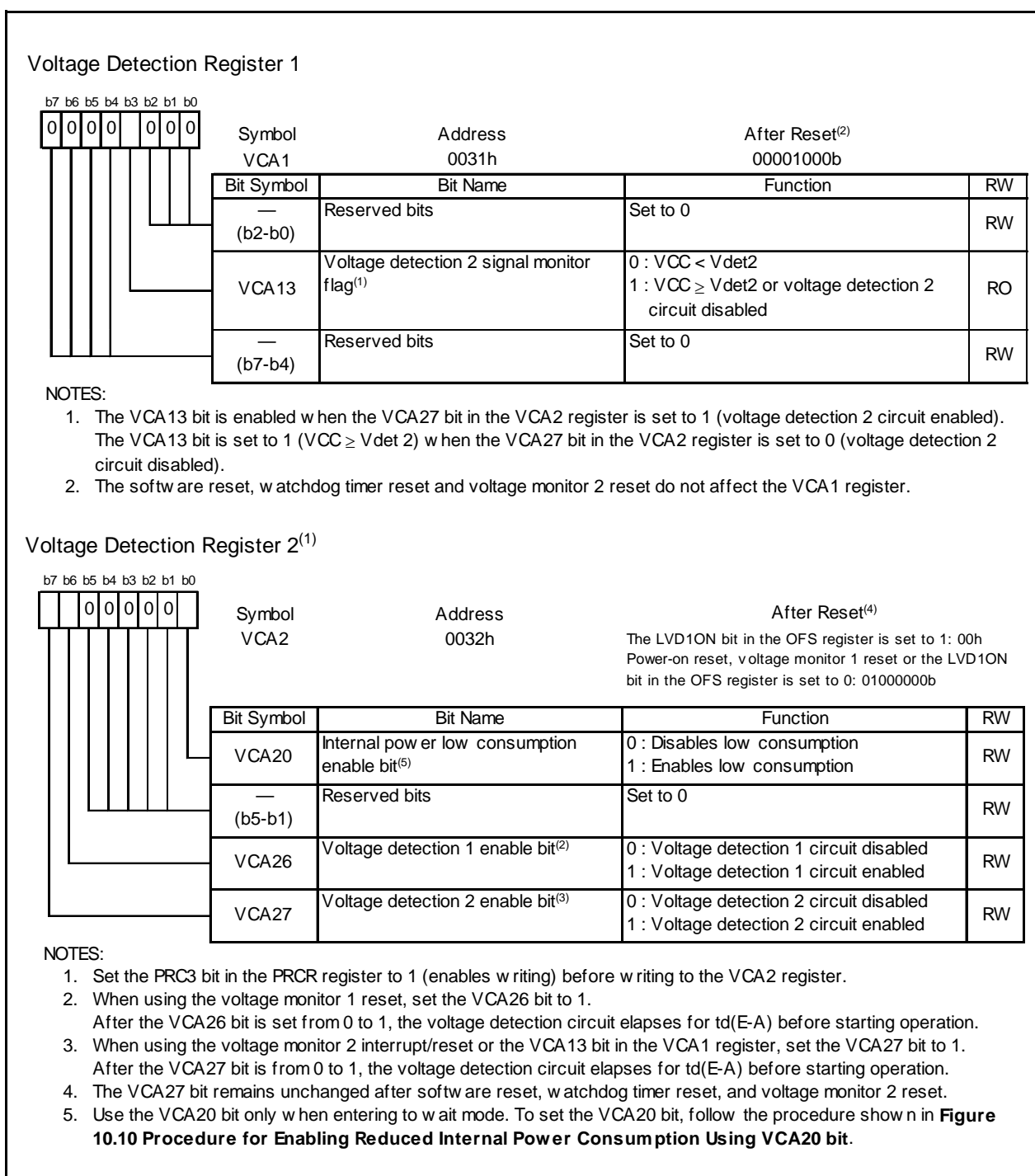


Figure 6.4 Registers VCA1 and VCA2

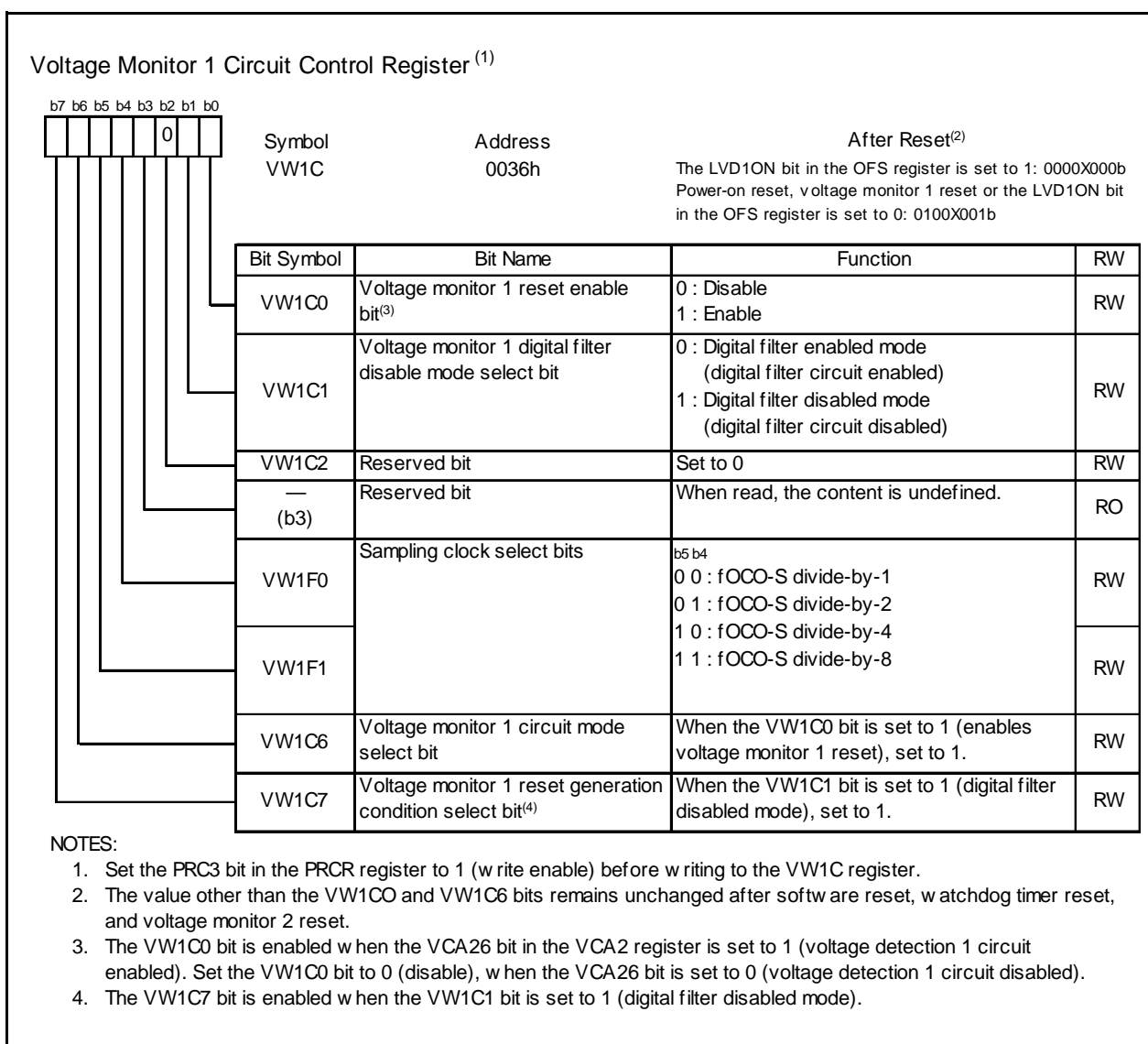


Figure 6.5 VW1C Register



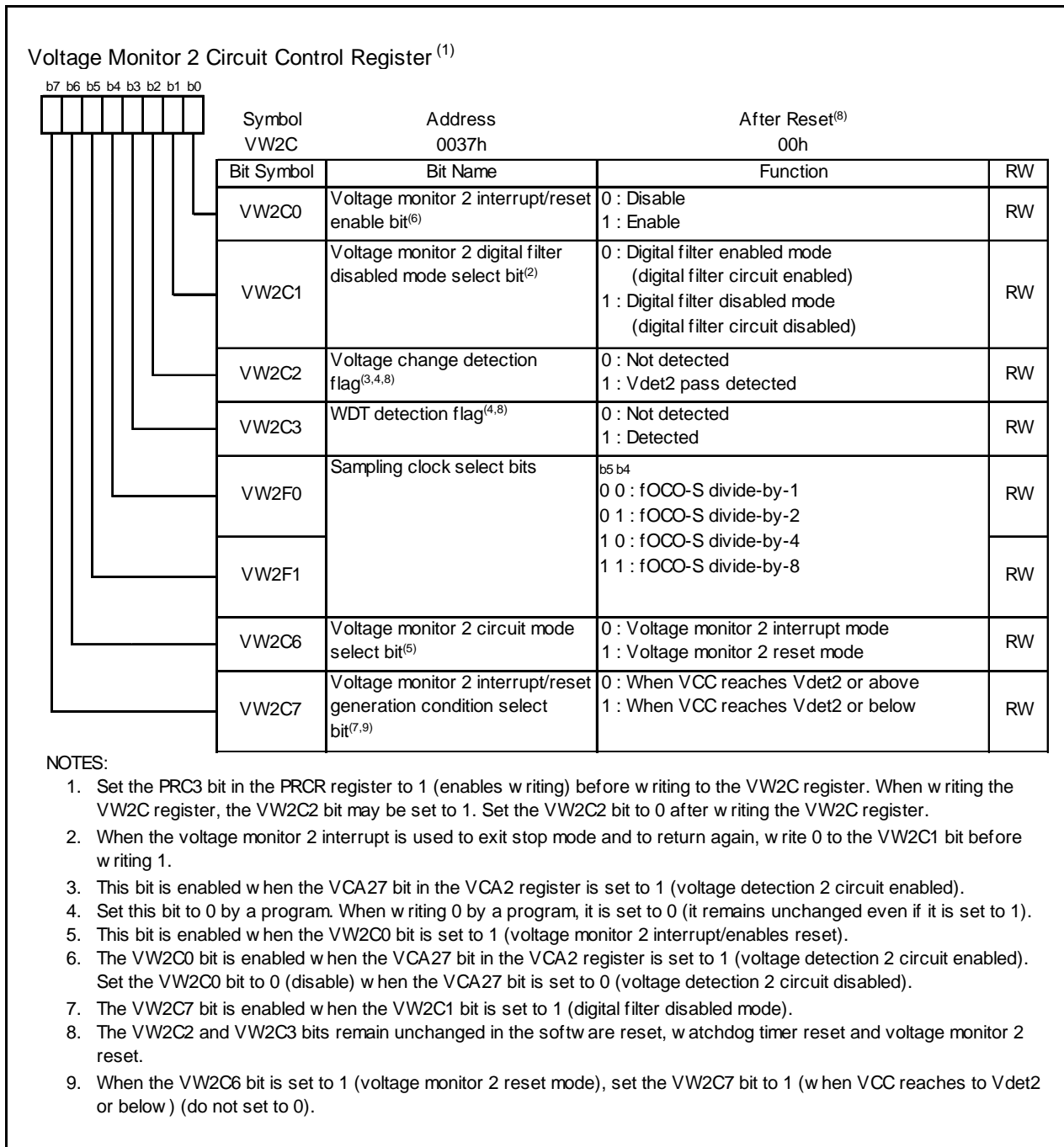


Figure 6.6 VW2C Register

## 6.1 VCC Input Voltage

### 6.1.1 Monitoring Vdet1

Vdet1 cannot be monitored.

### 6.1.2 Monitoring Vdet2

Set the VCA27 bit in the VCA2 register to 1 (voltage detection 2 circuit enabled). After  $t_d(E-A)$  has elapsed (refer to **20. Electrical Characteristics**), Vdet2 can be monitored by the VCA13 bit in the VCA1 register.

## 6.2 Voltage Monitor 1 Reset

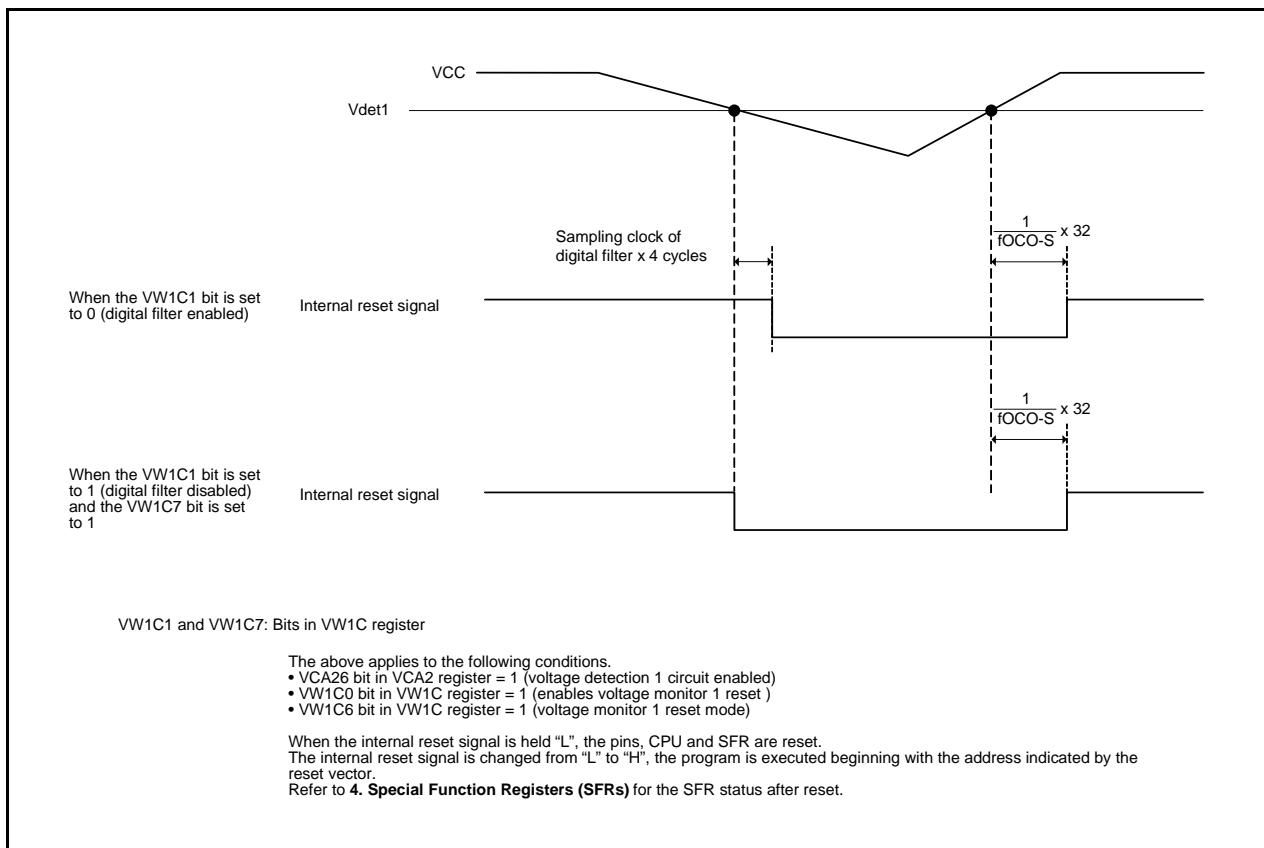
Table 6.2 lists the Procedure for Setting Bits Associated with Voltage Monitor 1 Reset and Figure 6.7 shows an Example of Voltage Monitor 1 Reset Operation. To use the voltage monitor 1 reset to exit stop mode, set the VW1C1 bit in the VW1C register to 1 (digital filter disabled).

**Table 6.2 Procedure for Setting Bits Associated with Voltage Monitor 1 Reset**

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

**NOTE:**

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



**Figure 6.7 Example of Voltage Monitor 1 Reset Operation**

### 6.3 Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset

Table 6.3 lists the Procedure for Setting Bits Associated with Voltage Monitor 2 Interrupt and Reset. Figure 6.8 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 6.3 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 VCA27 bit in the VCA2 register to 1 (voltage detection 2 circuit enabled)			
2	Wait for td(E-A)			
3	Select the sampling clock of the digital filter by the VW2F0 to VW2F1 bits in the VW2C register		Select the timing of the interrupt and reset request by the VW2C7 bit in the VW2C register <sup>(1)</sup>	
4 <sup>(2)</sup>	Set the VW2C1 bit in the VW2C register to 0 (digital filter enabled)		Set the VW2C1 bit in the VW2C register to 1 (digital filter disabled)	
5 <sup>(2)</sup>	Set the VW2C6 bit in the VW2C register to 0 (voltage monitor 2 interrupt mode)	Set the VW2C6 bit in the VW2C register to 1 (voltage monitor 2 reset mode)	Set the VW2C6 bit in the VW2C register to 0 (voltage monitor 2 interrupt mode)	Set the VW2C6 bit in the VW2C register to 1 (voltage monitor 2 reset mode)
6	Set the VW2C2 bit in the VW2C register to 0 (passing of Vdet2 is not detected)			
7	Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on)		-	
8	Wait for the sampling clock of the digital filter x 4 cycles		- (no wait time)	
9	Set the VW2C0 bit in the VW2C register to 1 (enables voltage monitor 2 interrupt/reset)			

**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 3, 4, and 5 can be executed simultaneously (with 1 instruction).

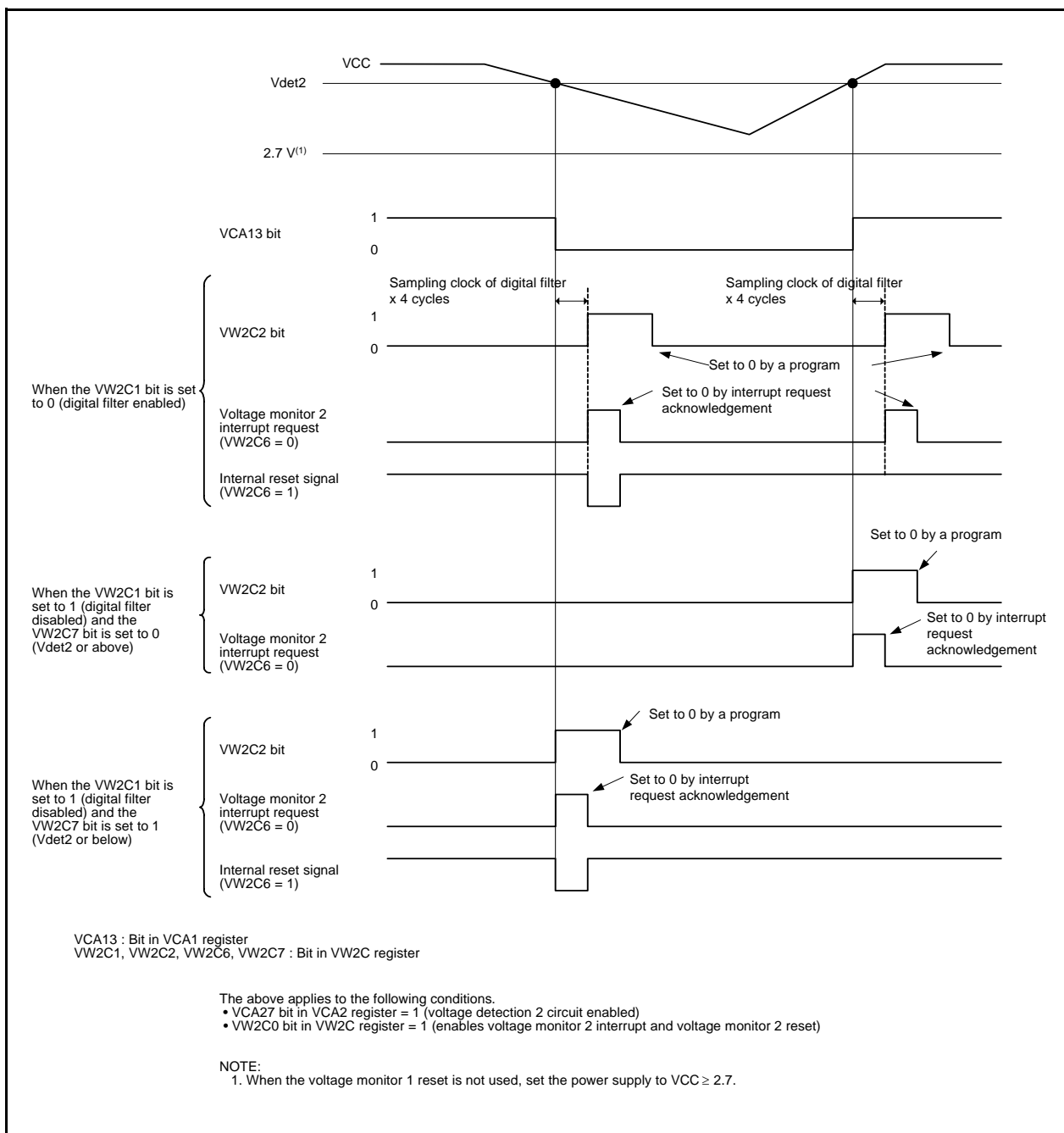


Figure 6.8 Example of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset Operation

## 7. Programmable I/O Ports

There are 41 programmable Input/Output ports (I/O ports) P0 to P2, P3\_0, P3\_1, P3\_3 to P3\_5, P3\_7, P4\_3 to P4\_5, and P6. Also, P4\_6 and P4\_7 can be used as input-only ports if the XIN clock oscillation circuit is not used, and the P4\_2 can be used as an input-only port if the A/D converter is not used.

**Table 7.1 Overview of Programmable I/O Ports**

Ports	I/O	Type of Output	I/O Setting	Internal Pull-Up Resistor
P0 to P2, P6	I/O	CMOS3 state	Set every bit	Set every 4 bits <sup>(1)</sup>
P3_0, P3_1, P3_3 to P3_5, P3_7	I/O	CMOS3 state	Set every bit	Set every 3 bits <sup>(1)</sup>
P4_3	I/O	CMOS3 state	Set every bit	Set every bit <sup>(1)</sup>
P4_4, P4_5	I/O	CMOS3 state	Set every bit	Set every 2 bits <sup>(1)</sup>
P4_2 <sup>(2)</sup> P4_6, P4_7 <sup>(3)</sup>	I	(No output function)	None	None

**NOTES:**

1. In input mode, whether the internal pull-up resistor is connected or not can be selected by the PUR0 and PUR1 registers.
2. When the A/D converter is not used, these ports can be used as the input port only.
3. When the XIN clock oscillation circuit is not used, these ports can be used as the input port only.

### 7.1 Functions of Programmable I/O Ports

The PDi<sub>j</sub> (i = 0 to 4, 6, j = 0 to 7) bit in the PDi register controls I/O of the ports P0 to P2, P3\_0, P3\_1, P3\_3 to P3\_5, P3\_7, P4\_3 to P4\_5, and P6. The Pi register consists of a port latch to hold output data and a circuit to read pin state.

Figures 7.1 to 7.7 show the Configurations of Programmable I/O Ports. Table 7.2 lists the Functions of Programmable I/O Ports. Also, Figure 7.9 shows the PDi (i = 0 to 4 and 6) Registers. Figure 7.10 shows the Pi (i = 0 to 4 and 6) Registers, Figure 7.11 shows the Registers PUR0 and PUR1 and Figure 7.12 shows the PMR Register.

**Table 7.2 Functions of Programmable I/O Ports**

Operation When Accessing Pi Register	Value of PDi <sub>j</sub> Bit in PDi Register <sup>(1)</sup>	
	When PDi <sub>j</sub> bit is set to 0 (input mode)	When PDi <sub>j</sub> 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 in the port latch, it is output from the pin.

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

**NOTE:**

1. Nothing is assigned to bits PD3\_2, PD3\_6, PD4\_0 to PD4\_2, PD4\_6, and PD4\_7.

## 7.2 Effect on Peripheral Functions

Programmable I/O ports function as I/O of peripheral functions (refer to **Table 1.6 Pin Name Information by Pin Number**).

Table 7.3 lists the Setting of PDi\_j Bit when Functioning as I/O Ports for Peripheral Functions (i = 0 to 4, 6 j = 0 to 7). Refer to descriptions of each function for how to set peripheral functions.

**Table 7.3 Setting of PDi\_j Bit when Functioning as I/O Ports for Peripheral Functions (i = 0 to 4, 6 j = 0 to 7)**

I/O of Peripheral Functions	PDi_j Bit Setting of Port shared with Pin
Input	Set this bit to 0 (input mode).
Output	This bit can be set to both 0 or 1 (output regardless of the port setting)

## 7.3 Pins Other than Programmable I/O Ports

Figure 7.8 shows the Configuration of I/O Pins.

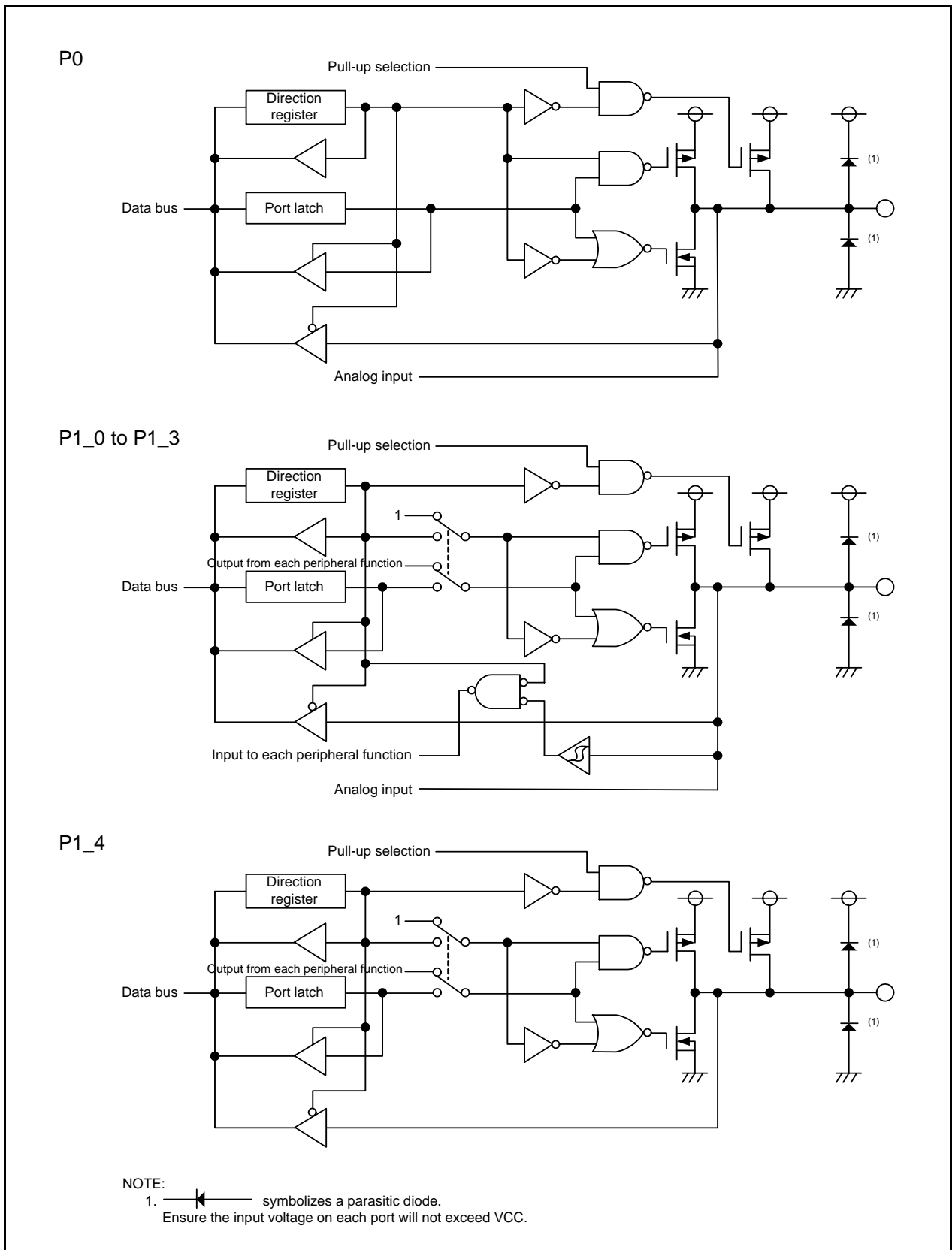


Figure 7.1 Configuration of Programmable I/O Ports (1)



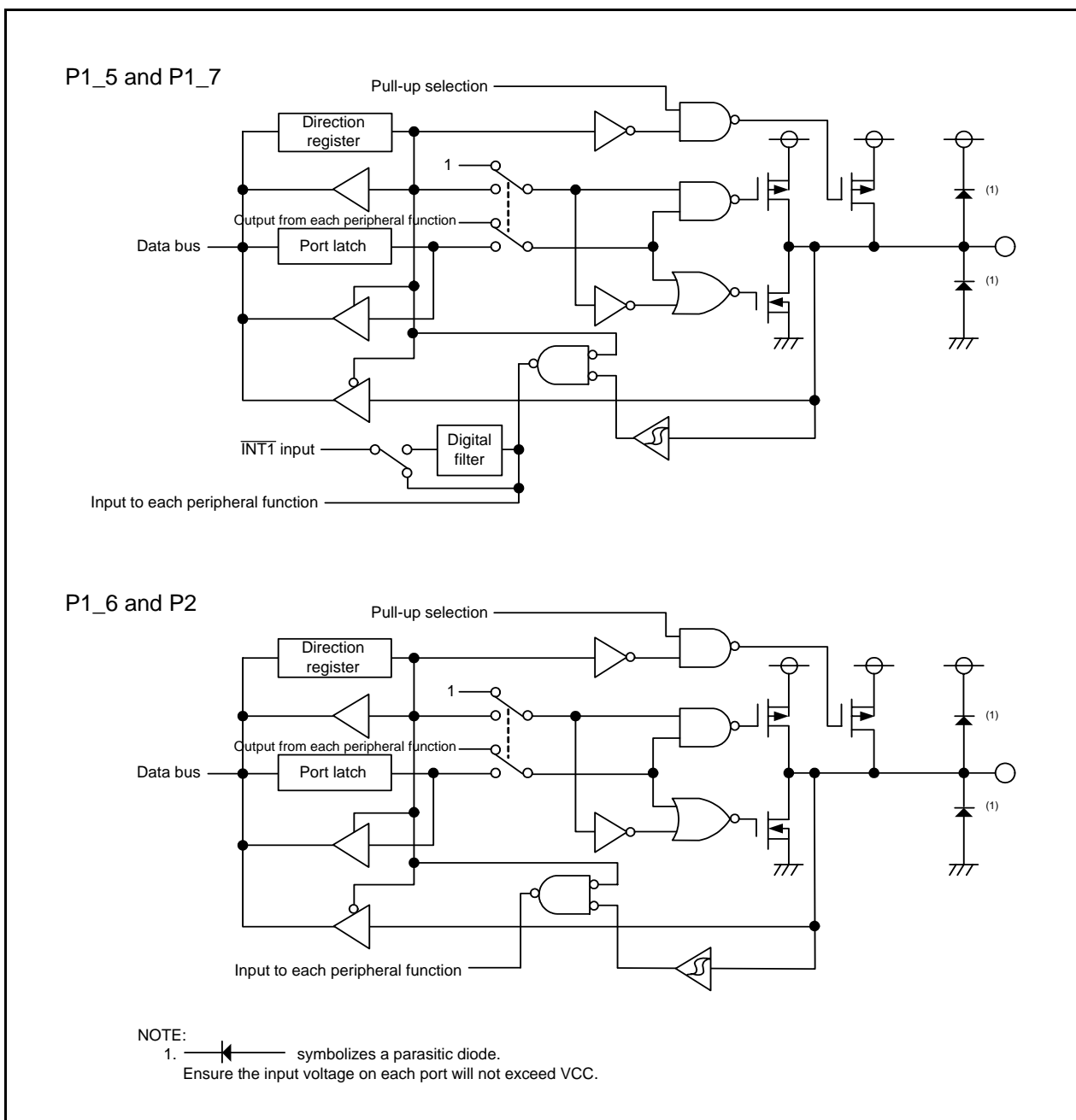


Figure 7.2 Configuration of Programmable I/O Ports (2)

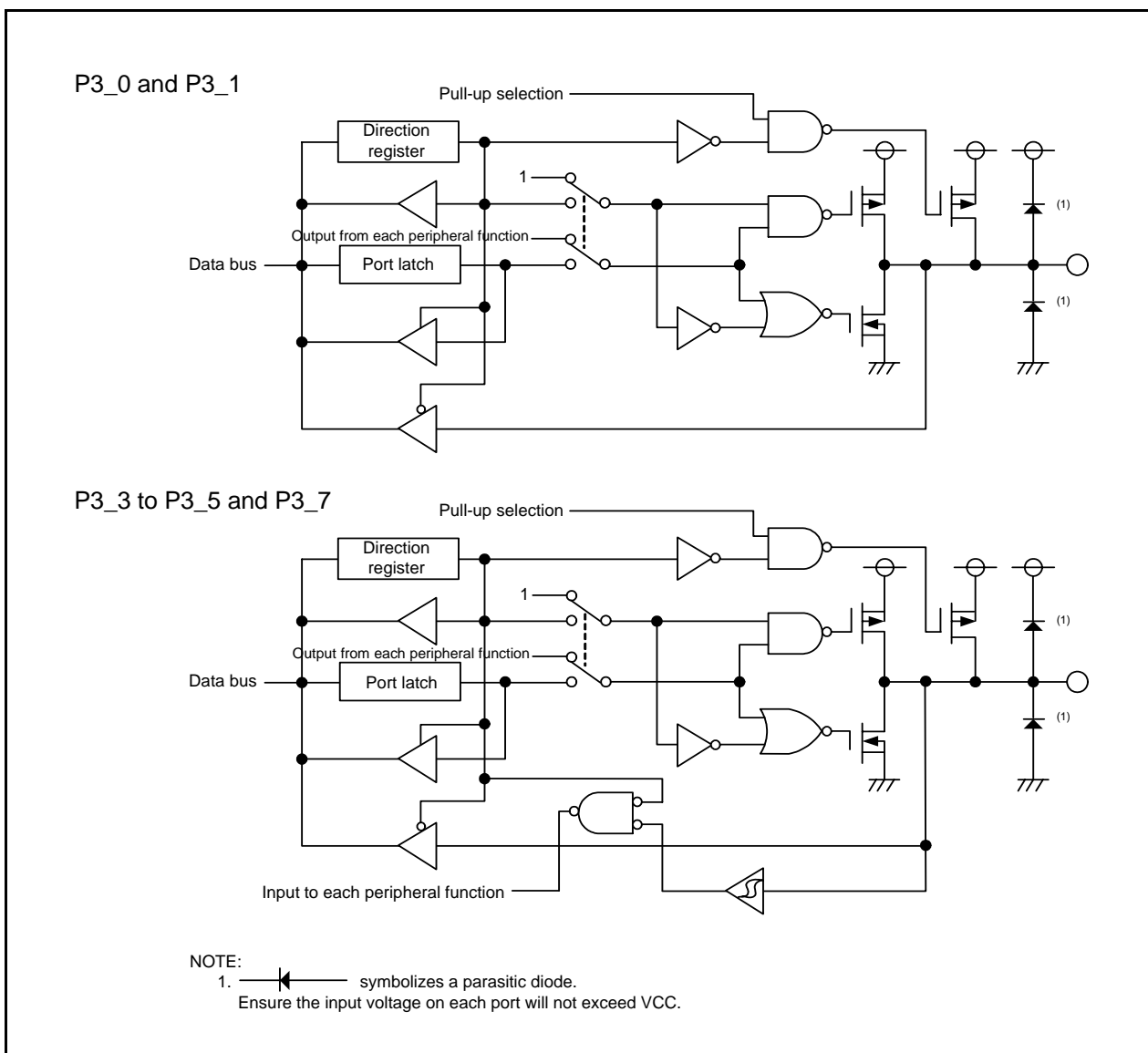


Figure 7.3 Configuration of Programmable I/O Ports (3)

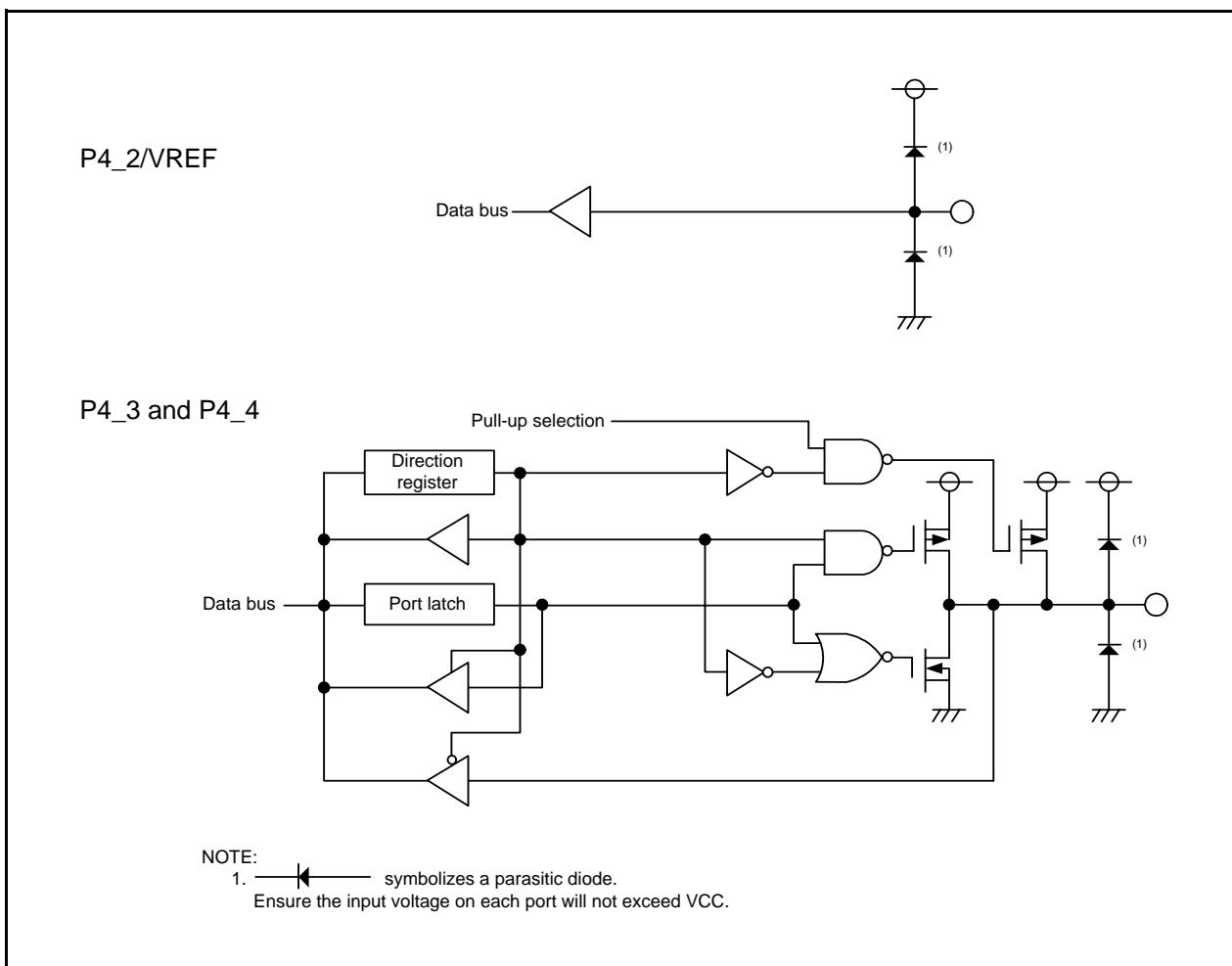


Figure 7.4 Configuration of Programmable I/O Ports (4)

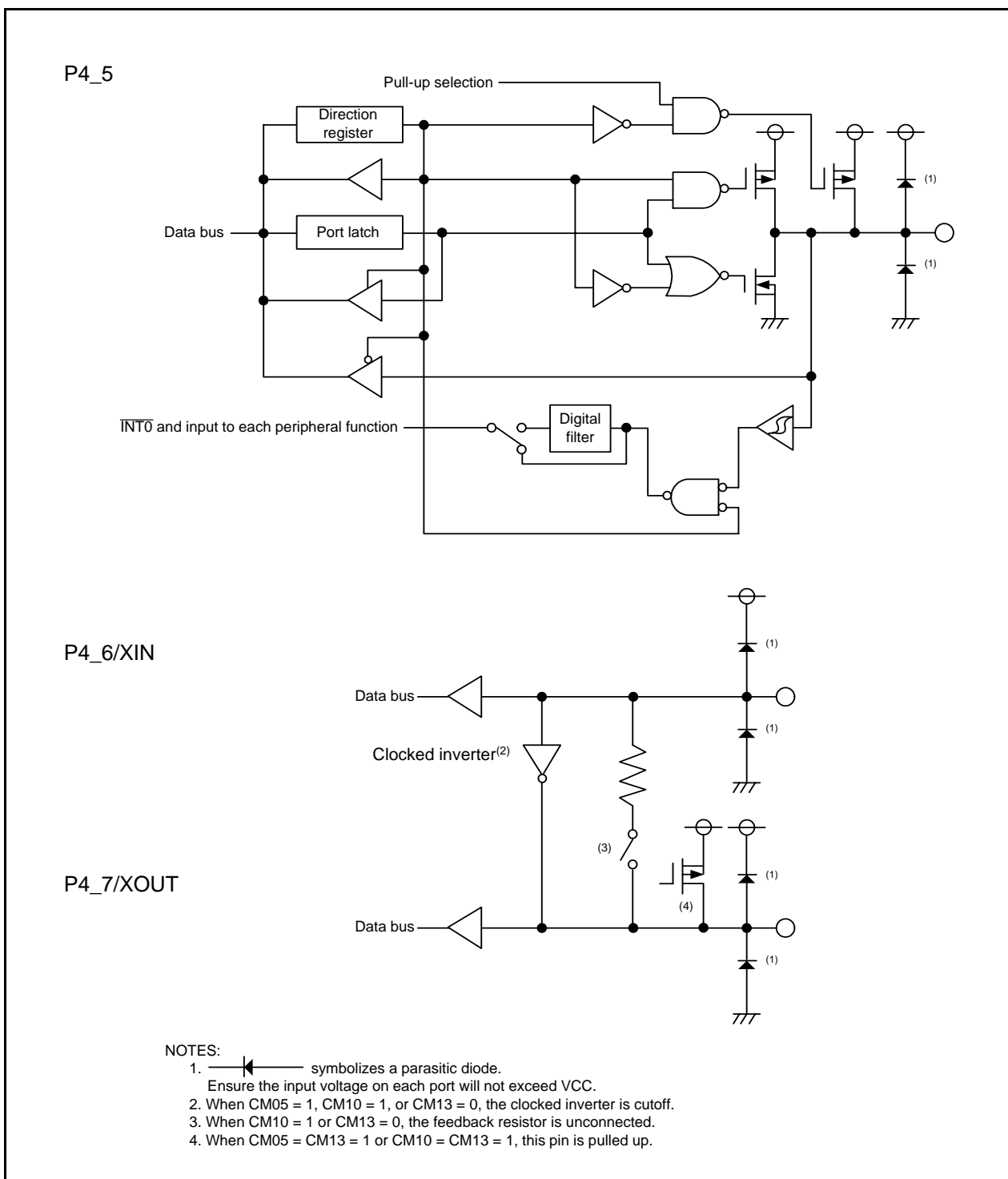


Figure 7.5 Configuration of Programmable I/O Ports (5)

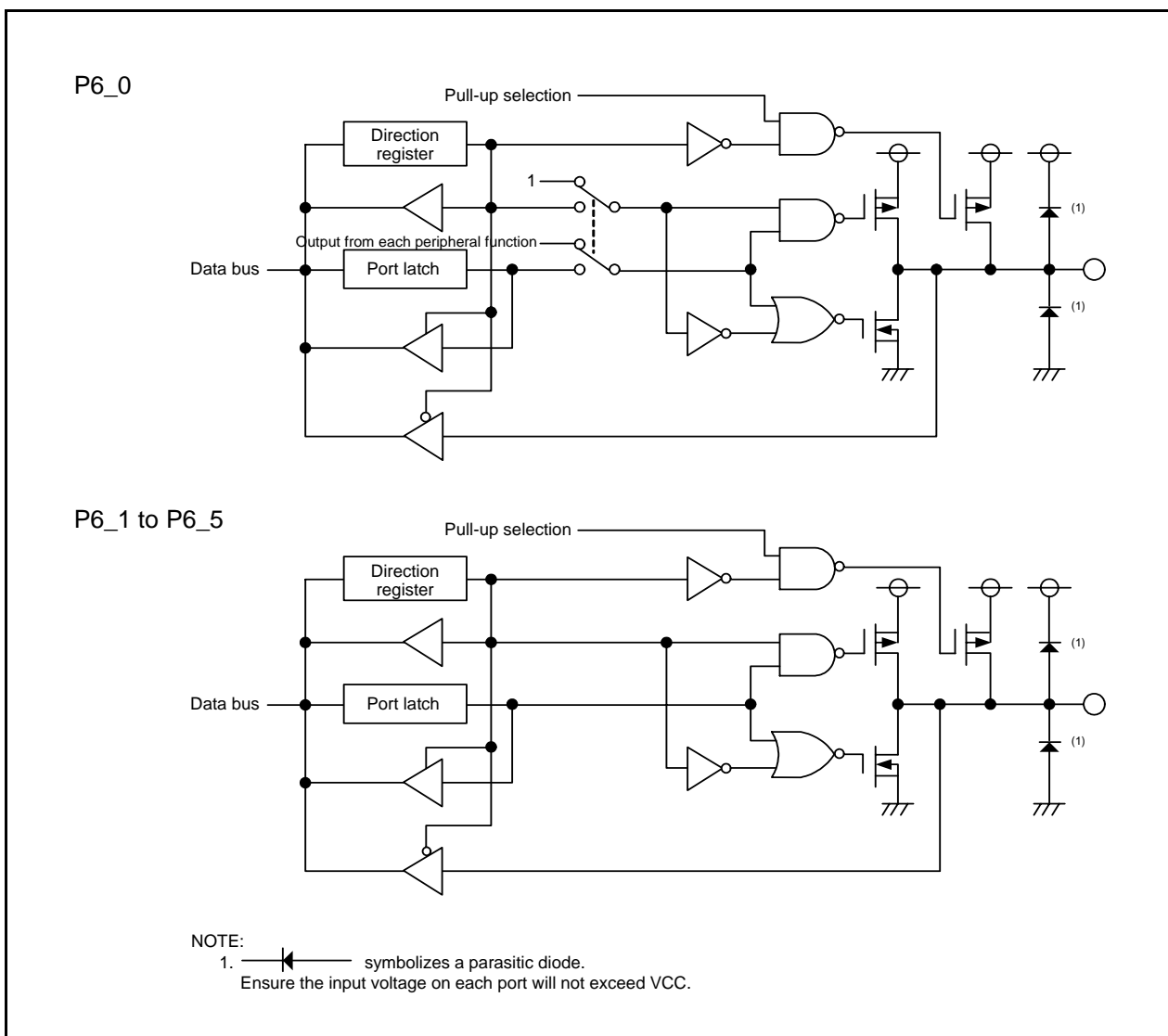


Figure 7.6 Configuration of Programmable I/O Ports (6)

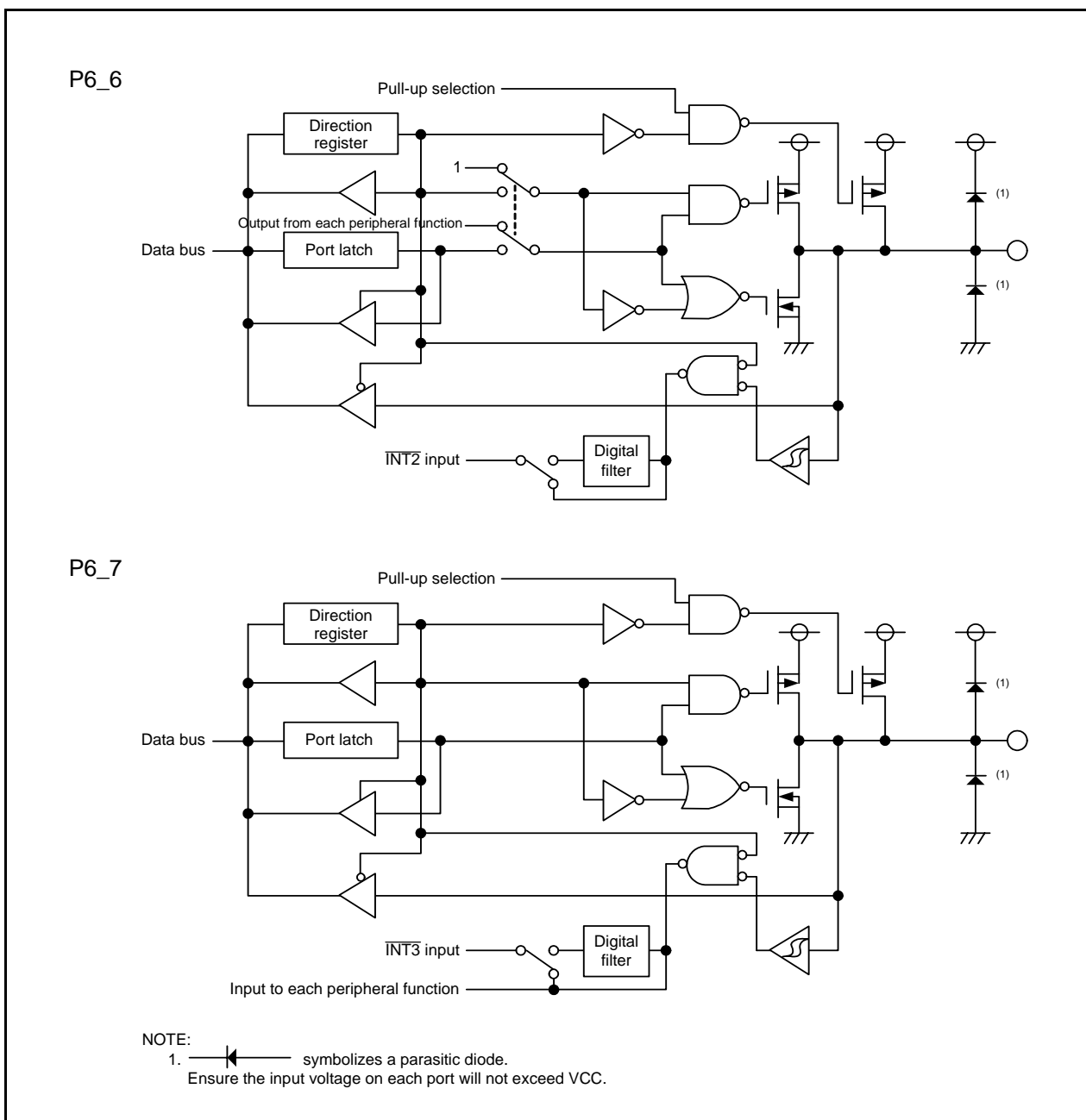


Figure 7.7 Configuration of Programmable I/O Ports (7)

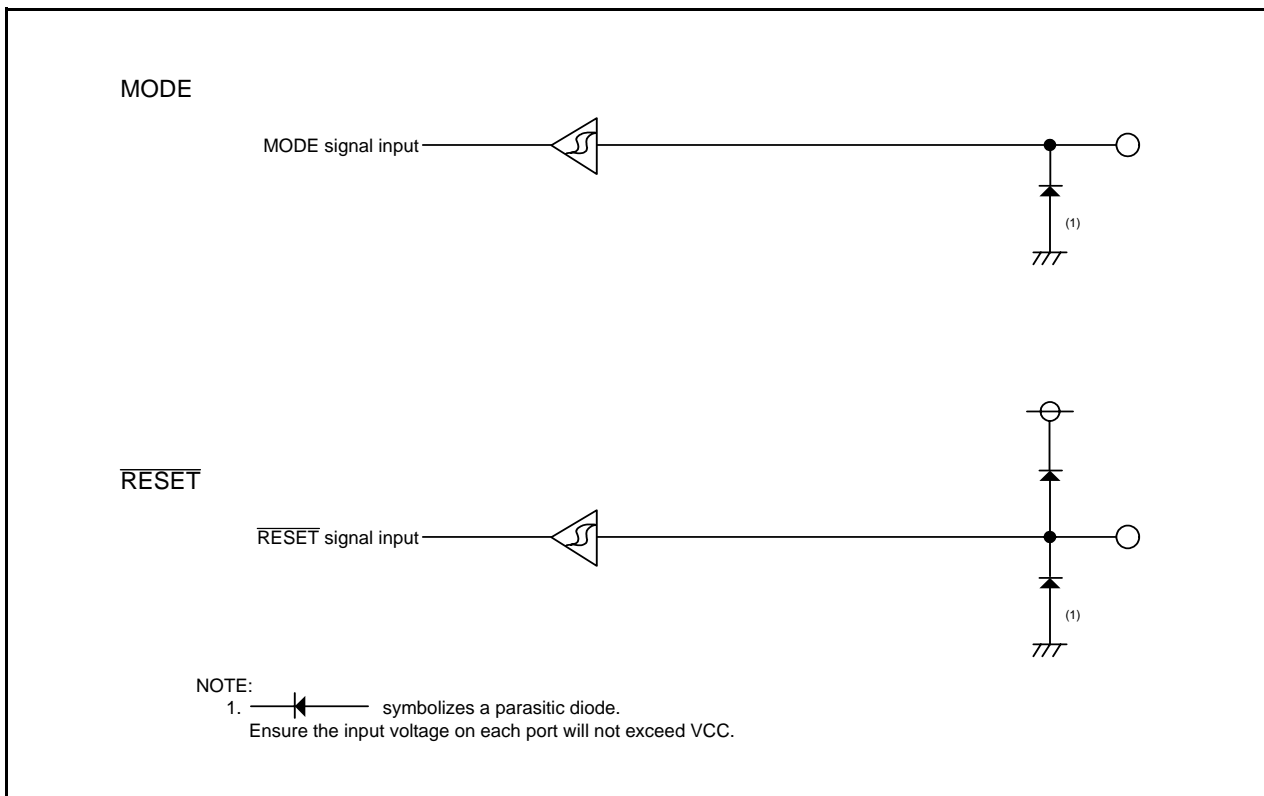


Figure 7.8 Configuration of I/O Pins

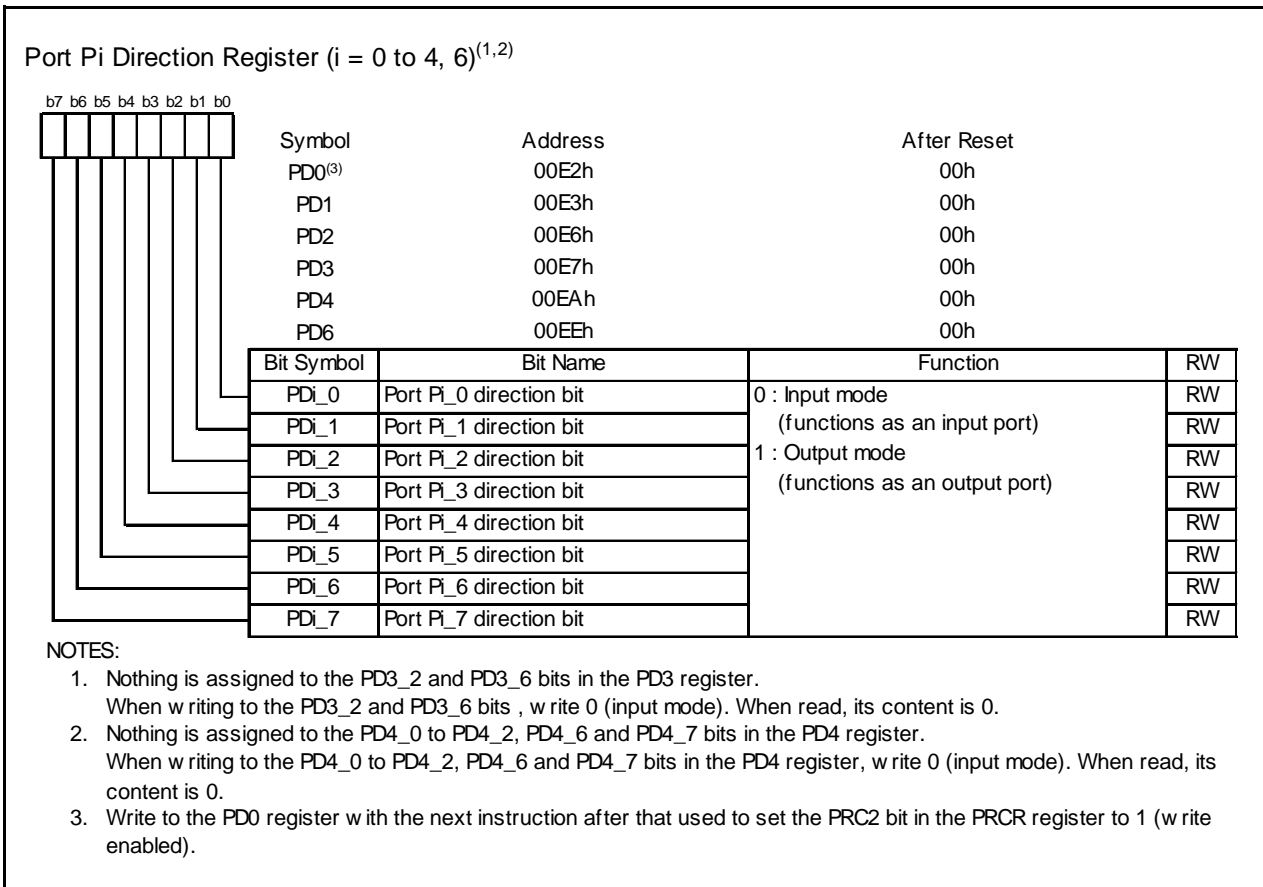


Figure 7.9 PDi (i = 0 to 4 and 6) Registers

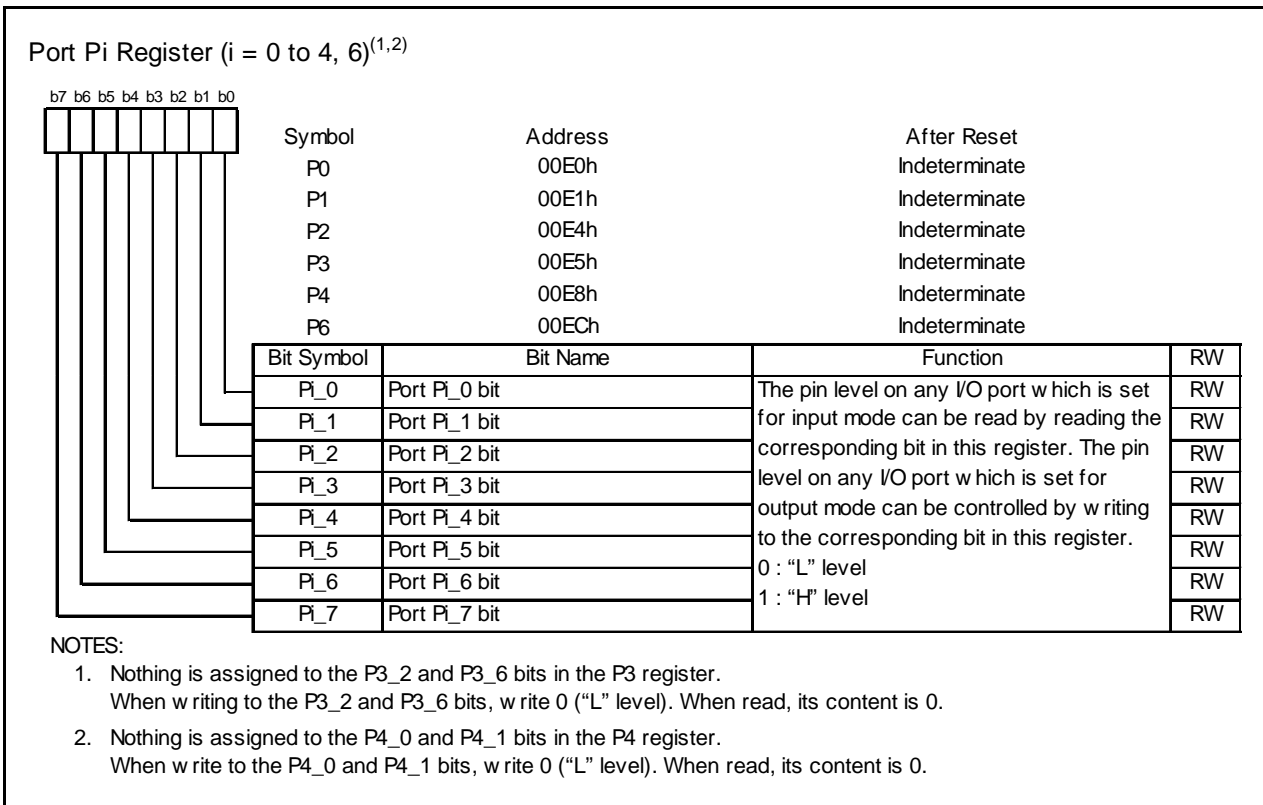
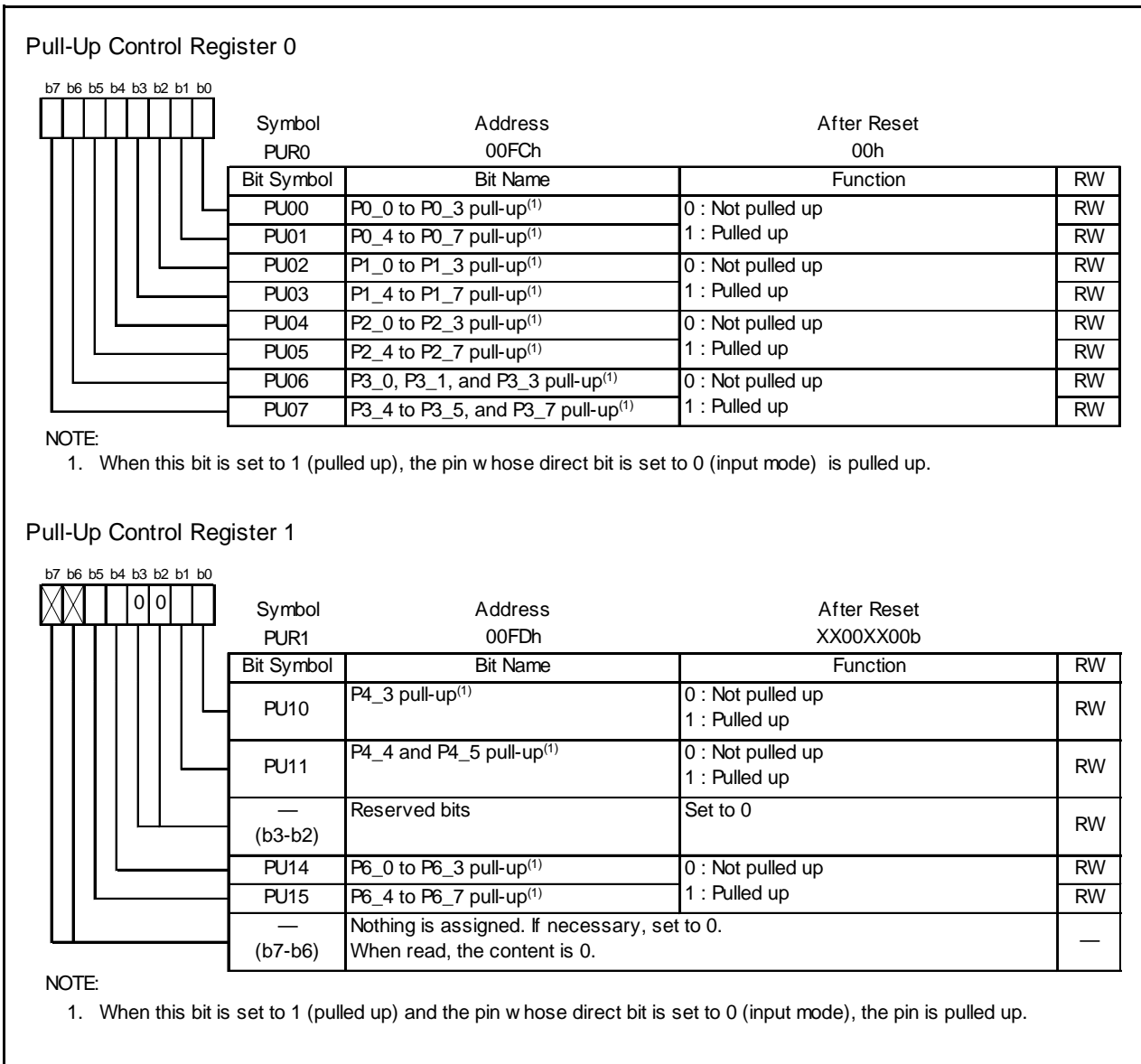
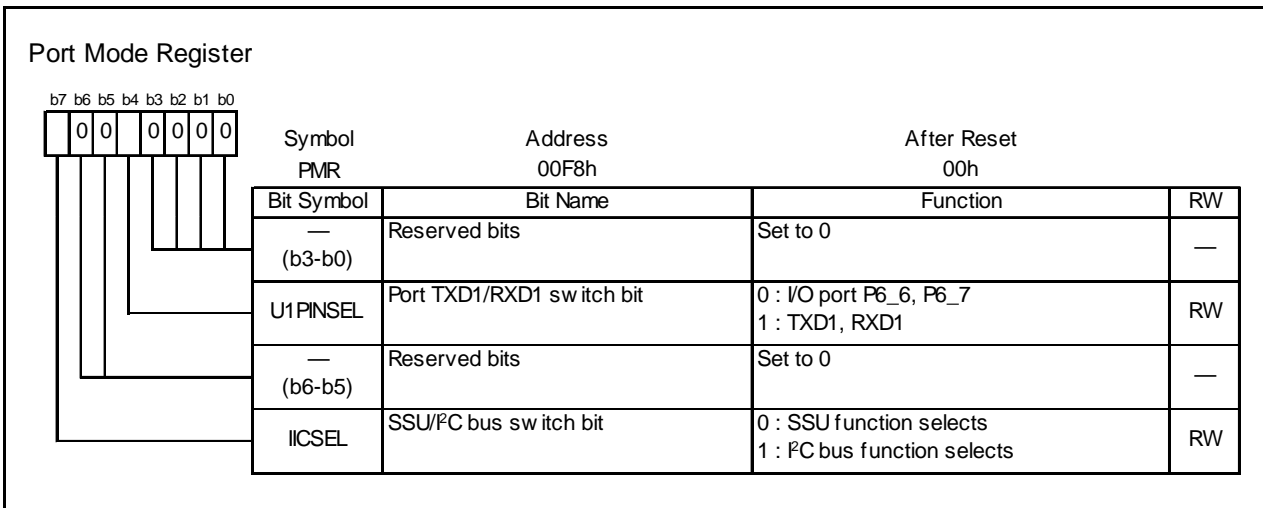


Figure 7.10 Pi (i = 0 to 4 and 6) Registers





**Figure 7.11 Registers PUR0 and PUR1**



## 7.4 Port Settings

Table 7.4 to Table 7.47 list the port settings.

**Table 7.4 Port P0\_0/AN7**

Register	PD0	ADCON0				Function
Bit	PD0_0	CH2	CH1	CH0	ADGSEL0	
Setting value	0	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	Output port
	0	1	1	1	0	A/D converter input (AN7)

X: 0 or 1

NOTE:

1. Pulled up by setting the PU00 bit in the PUR0 register to 1.

**Table 7.5 Port P0\_1/AN6**

Register	PD0	ADCON0				Function
Bit	PD0_1	CH2	CH1	CH0	ADGSEL0	
Setting value	0	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	Output port
	0	1	1	0	0	A/D converter input (AN6)

X: 0 or 1

NOTE:

1. Pulled up by setting the PU00 bit in the PUR0 register to 1.

**Table 7.6 Port P0\_2/AN5**

Register	PD0	ADCON0				Function
Bit	PD0_2	CH2	CH1	CH0	ADGSEL0	
Setting value	0	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	Output port
	0	1	0	1	0	A/D converter input (AN5)

X: 0 or 1

NOTE:

1. Pulled up by setting the PU00 bit in the PUR0 register to 1.

**Table 7.7 Port P0\_3/AN4**

Register	PD0	ADCON0				Function
Bit	PD0_3	CH2	CH1	CH0	ADGSEL0	
Setting value	0	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	Output port
	0	1	0	0	0	A/D converter input (AN4)

X: 0 or 1

NOTE:

1. Pulled up by setting the PU00 bit in the PUR0 register to 1.

**Table 7.8 Port P0\_4/AN3**

Register	PD0	ADCON0				Function
Bit	PD0_4	CH2	CH1	CH0	ADGSEL0	
Setting value	0	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	Output port
	0	0	1	1	0	A/D converter input (AN3)

X: 0 or 1

NOTE:

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

**Table 7.9 Port P0\_5/AN2**

Register	PD0	ADCON0				Function
Bit	PD0_5	CH2	CH1	CH0	ADGSEL0	
Setting value	0	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	Output port
	0	0	1	0	0	A/D converter input (AN2)

X: 0 or 1

NOTE:

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

**Table 7.10 Port P0\_6/AN1**

Register	PD0	ADCON0				Function
Bit	PD0_6	CH2	CH1	CH0	ADGSEL0	
Setting value	0	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	Output port
	0	0	0	1	0	A/D converter input (AN1)

X: 0 or 1

NOTE:

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

**Table 7.11 Port P0\_7/AN0**

Register	PD0	ADCON0				Function
Bit	PD0_7	CH2	CH1	CH0	ADGSEL0	
Setting value	0	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	Output port
	0	0	0	0	0	A/D converter input (AN0)

X: 0 or 1

NOTE:

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

**Table 7.12 Port P1\_0/KI0/AN8**

Register	PD1	KIEN	ADCON0				Function
Bit	PD1_0	KI0EN	CH2	CH1	CH0	ADGSEL0	
Setting value	0	X	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	X	Output port
	0	1	X	X	X	X	KI0 input
	0	X	1	0	0	1	A/D converter input (AN8)

X: 0 or 1

NOTE:

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

**Table 7.13 Port P1\_1/KI1/AN9**

Register	PD1	KIEN	ADCON0				Function
Bit	PD1_1	KI1EN	CH2	CH1	CH0	ADGSEL0	
Setting value	0	X	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	X	Output port
	0	1	X	X	X	X	KI1 input
	0	X	1	0	1	1	A/D converter input (AN9)

X: 0 or 1

NOTE:

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

**Table 7.14 Port P1\_2/KI2/AN10**

Register	PD1	KIEN	ADCON0				Function
Bit	PD1_2	KI2EN	CH2	CH1	CH0	ADGSEL0	
Setting value	0	X	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	X	Output port
	0	1	X	X	X	X	KI2 input
	0	X	1	1	0	1	A/D converter input (AN10)

X: 0 or 1

NOTE:

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

**Table 7.15 Port P1\_3/KI3/AN11**

Register	PD1	KIEN	ADCON0				Function
Bit	PD1_3	KI3EN	CH2	CH1	CH0	ADGSEL0	
Setting value	0	X	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	X	Output port
	0	1	X	X	X	X	KI3 input
	0	X	1	1	1	1	A/D converter input (AN11)

X: 0 or 1

NOTE:

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

**Table 7.16 Port P1\_4/TXD0**

Register	PD1	UOMR			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	X	0	0	1	TXD0 output <sup>(2)</sup>
			1	0	0	
			1	0	1	
			1	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 7.17 Port P1\_5/RXD0/(TRAIO)/(INT1)**

Register	PD1	TRAIOC		TRAMR			INTEN	Function	
Bit	PD1_5	TIOSEL	TOPCR	TMOD2	TMOD1	TMOD0	INT1EN		
Setting value	0	0	X	X	X	X	X	Input port <sup>(1)</sup>	
		X	1	X	X	X			
		X	X	Other than 001b					
	1	1	0	X	X	X	X	X	Output port
			X	1	X	X	X		
			X	X	Other than 001b				
	0	0	X	X	Other than 001b			X	RXD0 input
			0	X	0	0	1		
			1	X	Other than 001b				
			1	X	Other than 001b				
X	X	1	0	0	0	1	X	TRAIO pulse output	

X: 0 or 1

NOTE:

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

**Table 7.18 Port P1\_6/CLK0**

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

X: 0 or 1

NOTE:

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

**Table 7.19 Port P1\_7/TRAIO/ $\overline{\text{INT1}}$** 

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

X: 0 or 1

NOTE:

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

**Table 7.20 Port P2\_0/TRDIOA0/TRDCLK**

Register	PD2	TRDOER1	TRDFCR				TRDIORA0			Function
Bit	PD2_0	EA0	CMD1	CMD0	STCLK	PWM3	IOA2	IOA1	IOA0	
Setting value	0	1	X	X	X	X	X	X	X	Input port <sup>(1)</sup>
	1	1	X	X	X	X	X	X	X	Output port
	0	X	0	0	0	1	1	X	X	Timer mode (input capture function)
	0	X	X	X	1	1	0	0	0	External clock input (TRDCLK)
	X	0	0	0	0	0	0	X	X	PWM3 mode waveform output
	X	0	0	0	0	0	1	0	0	1
							0	1	X	

X: 0 or 1

NOTE:

1. Pulled up by setting the PU04 bit in the PUR0 register to 1.

**Table 7.21 Port P2\_1/TRDIOB0**

Register	PD2	TRDOER1	TRDFCR			TRDPMR	TRDIOA0			Function	
Bit	PD2_1	EB0	CMD1	CMD0	PWM3	PWMB0	IOB2	IOB1	IOB0		
Setting value	0	1	X	X	X	X	X	X	X	Input port <sup>(1)</sup>	
	1	1	X	X	X	X	X	X	X	Output port	
	0	X	0	0	1	0	1	X	X	Timer mode (input capture function)	
	X	0	1	0	X	X	X	X	X	X	Complementary PWM mode waveform output
			1	1							
	X	0	0	1	X	X	X	X	X	X	Reset synchronous PWM mode waveform output
	X	0	0	0	0	0	X	X	X	X	PWM3 mode waveform output
	X	0	0	0	1	1	X	X	X	X	PWM mode waveform output
X	0	0	0	1	0	0	0	1	X	Timer mode waveform output (output compare function)	
						0	1				

X: 0 or 1

NOTE:

1. Pulled up by setting the PU04 bit in the PUR0 register to 1.

**Table 7.22 Port P2\_2/TRDIOC0**

Register	PD2	TRDOER1	TRDFCR			TRDPMR	TRDIORC0			Function	
Bit	PD2_2	EC0	CMD1	CMD0	PWM3	PWMC0	IOC2	IOC1	IOC0		
Setting value	0	1	X	X	X	X	X	X	X	Input port <sup>(1)</sup>	
	1	1	X	X	X	X	X	X	X	Output port	
	0	X	0	0	1	0	1	X	X	Timer mode (input capture function)	
	X	0	1	0	X	X	X	X	X	X	Complementary PWM mode waveform output
			1	1							
	X	0	0	1	X	X	X	X	X	X	Reset synchronous PWM mode waveform output
	X	0	0	0	1	1	X	X	X	X	PWM mode waveform output
	X	0	0	0	1	0	0	0	1	X	Timer mode waveform output (output compare function)
0							1				

X: 0 or 1

NOTE:

1. Pulled up by setting the PU04 bit in the PUR0 register to 1.

**Table 7.23 Port P2\_3/TRDIOD0**

Register	PD2	TRDOER1	TRDFCR			TRDPMR	TRDIORC0			Function	
Bit	PD2_3	ED0	CMD1	CMD0	PWM3	PWMD0	IOD2	IOD1	IOD0		
Setting value	0	1	X	X	X	X	X	X	X	Input port <sup>(1)</sup>	
	1	1	X	X	X	X	X	X	X	Output port	
	0	X	0	0	1	0	1	X	X	Timer mode (input capture function)	
	X	0	1	0	X	X	X	X	X	X	Complementary PWM mode waveform output
			1	1							
	X	0	0	1	X	X	X	X	X	X	Reset synchronous PWM mode waveform output
	X	0	0	0	1	1	X	X	X	X	PWM mode waveform output
	X	0	0	0	1	0	0	0	1	X	Timer mode waveform output (output compare function)
0							1				

X: 0 or 1

NOTE:

1. Pulled up by setting the PU04 bit in the PUR0 register to 1.

**Table 7.24 Port P2\_4/TRDIOA1**

Register	PD2	TRDOER1	TRDFCR			TRDIOA1			Function	
Bit	PD2_4	EA1	CMD1	CMD0	PWM3	IOA2	IOA1	IOA0		
Setting value	0	1	X	X	X	X	X	X	Input port <sup>(1)</sup>	
	1	1	X	X	X	X	X	X	Output port	
	0	X	0	0	1	1	X	X	Timer mode (input capture function)	
	X	0	1	0	X	X	X	X	X	Complementary PWM mode waveform output
			1	1						
	X	0	0	1	X	X	X	X	Reset synchronous PWM mode waveform output	
X	0	0	0	1	0	0	1	X	Timer mode waveform output (output compare function)	
					0	1				

X: 0 or 1

NOTE:

1. Pulled up by setting the PU05 bit in the PUR0 register to 1.

**Table 7.25 Port P2\_5/TRDIOB1**

Register	PD2	TRDOER1	TRDFCR			TRDPMR	TRDIOA1			Function	
Bit	PD2_5	EB1	CMD1	CMD0	PWM3	PWMB1	IOB2	IOB1	IOB0		
Setting value	0	1	X	X	X	X	X	X	X	Input port <sup>(1)</sup>	
	1	1	X	X	X	X	X	X	X	Output port	
	0	X	0	0	1	0	1	X	X	Timer mode (input capture function)	
	X	0	1	0	X	X	X	X	X	X	Complementary PWM mode waveform output
			1	1							
	X	0	0	1	X	X	X	X	X	Reset synchronous PWM mode waveform output	
	X	0	0	0	1	1	X	X	X	PWM mode waveform output	
X	0	0	0	1	0	0	0	1	X	Timer mode waveform output (output compare function)	
						0	1				

X: 0 or 1

NOTE:

1. Pulled up by setting the PU05 bit in the PUR0 register to 1.

**Table 7.26 Port P2\_6/TRDIOC1**

Register	PD2	TRDOER1	TRDFCR			TRDPMR	TRDIOC1			Function	
Bit	PD2_6	EC1	CMD1	CMD0	PWM3	PWMC1	IOC2	IOC1	IOC0		
Setting value	0	1	X	X	X	X	X	X	X	Input port <sup>(1)</sup>	
	1	1	X	X	X	X	X	X	X	Output port	
	0	X	0	0	1	0	1	X	X	Timer mode (input capture function)	
	X	0	1	0	X	X	X	X	X	X	Complementary PWM mode waveform output
			1	1							
	X	0	0	1	X	X	X	X	X	Reset synchronous PWM mode waveform output	
	X	0	0	0	1	1	X	X	X	PWM mode waveform output	
X	0	0	0	1	0	0	0	1	X	Timer mode waveform output (output compare function)	
						0	1				

X: 0 or 1

NOTE:

1. Pulled up by setting the PU05 bit in the PUR0 register to 1.

**Table 7.27 Port P2\_7/TRDIOD1**

Register	PD2	TRDOER1	TRDFCR			TRDPMR	TRDIORC1			Function	
Bit	PD2_7	ED1	CMD1	CMD0	PWM3	PWMD1	IOD2	IOD1	IOD0		
Setting value	0	1	X	X	X	X	X	X	X	Input port <sup>(1)</sup>	
	1	1	X	X	X	X	X	X	X	Output port	
	0	X	0	0	1	0	1	X	X	Timer mode (input capture function)	
	X	0	1	0	X	X	X	X	X	X	Complementary PWM mode waveform output
			1	1							
	X	0	0	1	X	X	X	X	X	X	Reset synchronous PWM mode waveform output
	X	0	0	0	1	1	X	X	X	X	PWM mode waveform output
X	0	0	0	1	0	0	0	1	X	Timer mode waveform output (output compare function)	
						0	1				

X: 0 or 1

NOTE:

1. Pulled up by setting the PU05 bit in the PUR0 register to 1.

**Table 7.28 Port P3\_0/TRAO**

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

X: 0 or 1

NOTE:

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

**Table 7.29 Port P3\_1/TRBO**

Register	PD3	TRBMR		TRBIOC	Function
Bit	PD3_1	TMOD1	TMOD0	TOCNT	
Setting value	0	0	0	X	Input port <sup>(1)</sup>
	1	0	0	X	Output port
	X	01b		1	
	X	Other than 00b		0	TRBO output

X: 0 or 1

NOTE:

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

**Table 7.30 Port P3\_3/SSI**

Register	PD3	Clock Synchronous Serial I/O with Chip Select (Refer to <b>Table 16.4 Association between Communication Modes and I/O Pins.</b> )		PMR	Function	
Bit	PD3_3	SSI output control	SSI input control	IICSEL		
Setting value	0	0	0	0	Input port <sup>(1)</sup>	
	0	X	X	1		
	1	0	0	0	Output port <sup>(2)</sup>	
	1	X	X	1		
	X	0	1	0	0	SSI input
	X	1	1	0	0	SSI output <sup>(2)</sup>

X: 0 or 1

NOTES:

1. Pulled up by setting the PU06 bit in the PUR0 register to 1.
2. N-channel open drain output by setting the SOOS bit in the SSMR2 register to 1 when this pin functions as output.



**Table 7.31 Port P3\_4/SDA/SCS**

Register	PD3	SSMR2		PMR	ICCR1	Function	
Bit	PD3_4	CSS1	CSS0	IICSEL	ICE		
Setting value	0	0	0	0	X	Input port <sup>(1)</sup>	
	0	0	0	X	0		
	1	0	0	0	X	Output port <sup>(2)</sup>	
	1	0	0	X	0		
	X	0	1	0	X	SCS input	
	X	1	0	0	0	X	SCS output <sup>(2)</sup>
			1	1			
X	X	X	X	1	1	SDA input/output	

X: 0 or 1

NOTES:

1. Pulled up by setting the PU07 bit in the PUR0 register to 1.
2. N-channel open drain output by setting the CSOS bit in the SSMR2 register to 1 when this pin functions as output.

**Table 7.32 Port P3\_5/SCL/SSCK**

Register	PD3	Clock Synchronous Serial I/O with Chip Select (Refer to <b>Table 16.4 Association between Communication Modes and I/O Pins.</b> )		PMR	ICCR1	Function	
Bit	PD3_5	SSCK output control	SSCK input control	IICSEL	ICE		
Setting value	0	0	0	0	X	Input port <sup>(1)</sup>	
	0	0	0	X	0		
	1	0	0	0	X	Output port <sup>(2)</sup>	
	1	0	0	X	0		
	X	0	1	0	0	SSCK input	
	X	1	0	0	0	0	SSCK output <sup>(2)</sup>
	X	1	1	0	1	1	SCL input/output

X: 0 or 1

NOTES:

1. Pulled up by setting the PU07 bit in the PUR0 register to 1.
2. N-channel open drain output by setting the CSOS bit in the SSMR2 register to 1 when this pin functions as output.

**Table 7.33 Port P3\_7/SSO**

Register	PD3	Clock Synchronous Serial I/O with Chip Select (Refer to <b>Table 16.4 Association between Communication Modes and I/O Pins.</b> )		SSMR2	PMR	Function	
Bit	PD3_7	SSO output control	SSO input control	SOOS	IICSEL		
Setting value	0	0	0	X	0	Input port <sup>(1)</sup>	
	0	X	X	X	1		
	1	0	0	0	0	Output port	
	1	X	X	0	1		
	X	0	1	0	0	0	SSO input
	X	1	0	0	0	0	SSO output (CMOS output)
	X	1	1	0	1	0	SSO output (N-channel open-drain output)

X: 0 or 1

NOTE:

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

**Table 7.34 Port P4\_2/VREF**

Register	ADCON1	Function
Bit	VCUT	
Setting value	0	Input port
Setting value	1	Input port/VREF input

**Table 7.35 Port P4\_3**

Register	PD4	Function
Bit	PD4_3	
Setting value	0	Input port <sup>(1)</sup>
	1	Output port

NOTE:

1. Pulled up by setting the PU10 bit in the PUR0 register to 1.

**Table 7.36 Port P4\_4**

Register	PD4	Function
Bit	PD4_4	
Setting value	0	Input port <sup>(1)</sup>
	1	Output port

NOTE:

1. Pulled up by setting the PU11 bit in the PUR0 register to 1.

**Table 7.37 Port P4\_5/ $\overline{\text{INT0}}$** 

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

X: 0 or 1

NOTE:

1. Pulled up by setting the PU11 bit in the PUR0 register to 1.

**Table 7.38 Port P4\_6/XIN**

Register	CM1		CM0	Circuit specifications		Function
	CM13	CM10	CM05	Oscillation buffer	Feedback resistor	
Setting value	0	X	X	OFF	OFF	Input port
	1	0	0	ON	ON	XIN-XOUT oscillation
	1	0	1	OFF	ON	External XIN input
	1	1	0	OFF	OFF	XIN-XOUT oscillation stop
	1	1	1	OFF	OFF	XIN-XOUT oscillation stop

X: 0 or 1

**Table 7.39 Port P4\_7/XOUT**

Register	CM1		CM0	Circuit specifications		Function
	CM13	CM10	CM05	Oscillation buffer	Feedback resistor	
Setting value	0	X	X	OFF	OFF	Input port
	1	0	0	ON	ON	XIN-XOUT oscillation
	1	0	1	OFF	ON	XOUT is "H" pull-up
	1	1	0	OFF	OFF	XIN-XOUT oscillation stop
	1	1	1	OFF	OFF	XIN-XOUT oscillation stop

X: 0 or 1

**Table 7.40 Port P6\_0/TREO**

Register	PD6	TRECR1	Function
Bit	PD6_0	TOENA	
Setting value	0	0	Input port <sup>(1)</sup>
	1	0	Output port
	X	1	TREO output

X: 0 or 1

NOTE:

1. Pulled up by setting the PU14 bit in the PUR0 register to 1.

**Table 7.41 Port P6\_1**

Register	PD6	Function
Bit	PD6_1	
Setting value	0	Input port <sup>(1)</sup>
	1	Output port

X: 0 or 1

NOTE:

1. Pulled up by setting the PU14 bit in the PUR0 register to 1.

**Table 7.42 Port P6\_2**

Register	PD6	Function
Bit	PD6_2	
Setting value	0	Input port <sup>(1)</sup>
	1	Output port

X: 0 or 1

NOTE:

1. Pulled up by setting the PU14 bit in the PUR0 register to 1.

**Table 7.43 Port P6\_3**

Register	PD6	Function
Bit	PD6_3	
Setting value	0	Input port <sup>(1)</sup>
	1	Output port

NOTE:

1. Pulled up by setting the PU14 bit in the PUR0 register to 1.

**Table 7.44 Port P6\_4**

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

NOTE:

1. Pulled up by setting the PU15 bit in the PUR0 register to 1.

**Table 7.45 Port P6\_5**

Register	PD6	Function
Bit	PD6_5	
Setting value	0	Input port <sup>(1)</sup>
	1	Output port

NOTE:

1. Pulled up by setting the PU15 bit in the PUR0 register to 1.

**Table 7.46 Port P6\_6/ $\overline{\text{INT2}}$ /TXD1**

Register	PD6	PMR	U1MR			U1C0	INTEN	Function
Bit	PD6_6	U1PINSEL	SMD2	SMD1	SMD0	NCH	INT2EN	
Setting value	0	X	0	0	0	X	X	Input port <sup>(1)</sup>
		0	X	X	X			
	1	X	0	0	0	X	X	Output port
		0	X	X	X			
	0	X	X	X	X	X	1	$\overline{\text{INT2}}$ input
	X	1	0	0	1	0	X	TXD1 output (CMOS output)
			1	0	0			
			1	0	1			
			1	1	0			
	X	1	0	0	1	1	X	TXD1 output (N-channel open-drain output)
			1	0	0			
			1	0	1			
1			1	0				

X: 0 or 1

NOTE:

1. Pulled up by setting the PU15 bit in the PUR0 register to 1.

**Table 7.47 Port P6\_7/ $\overline{\text{INT3}}$ /RXD1**

Register	PD6	PMR	INTEN	Function
Bit	PD6_7	U1PINSEL	INT3EN	
Setting value	0	X	X	Input port <sup>(1)</sup>
	1	X	X	Output port
	0	X	1	$\overline{\text{INT3}}$ input
	0	1	X	RXD1 input

X: 0 or 1

NOTE:

1. Pulled up by setting the PU15 bit in the PUR0 register to 1.

## 7.5 Unassigned Pin Handling

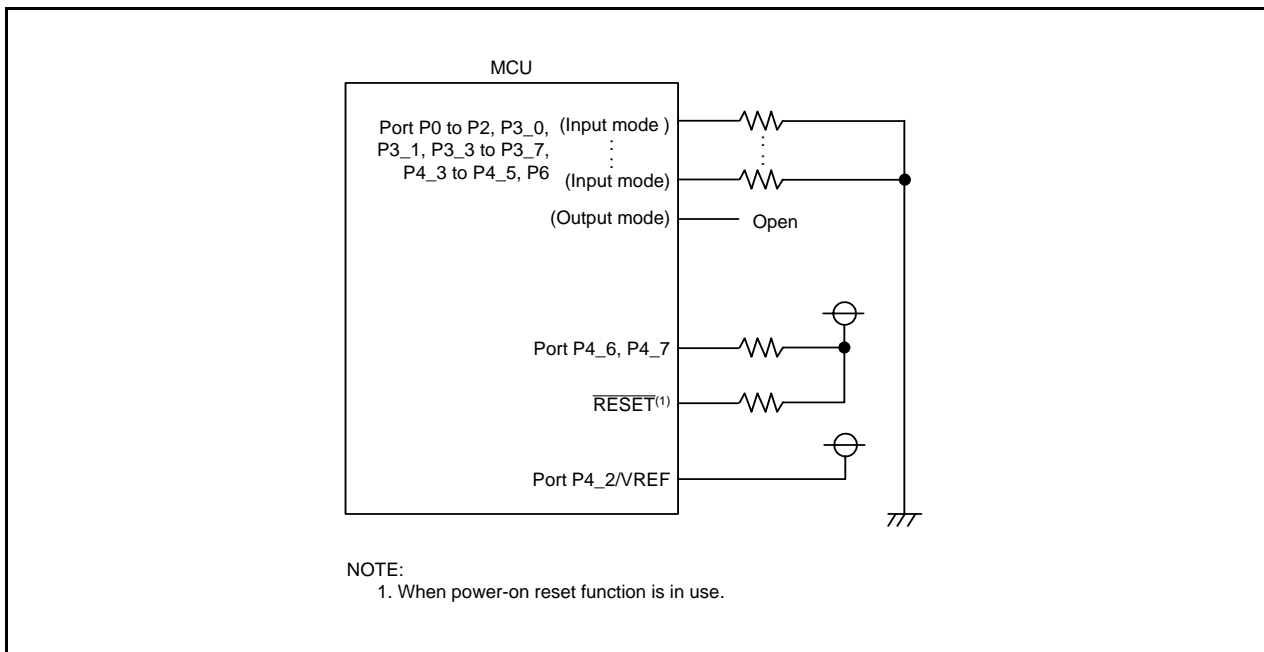
Table 7.48 lists Unassigned Pin Handling.

**Table 7.48 Unassigned Pin Handling**

Pin Name	Connection
Ports P0 to P2, P3_0, P3_1, P3_3 to P3_7, P4_3 to P4_5, P6	<ul style="list-style-type: none"> <li>• After setting to input mode, connect every pin to VSS via a resistor (pull-down) or connect every pin to VCC via a resistor (pull-up).(2)</li> <li>• After setting to output mode, leave these pins open.(1, 2)</li> </ul>
Ports P4_6, P4_7	Connect to VCC via a resistor (pull-up)(2)
Port P4_2/VREF	Connect to VCC
RESET <sup>(3)</sup>	Connect to VCC via a resistor (pull-up)(2)

### NOTES:

1. If these ports are set to output mode and left open, they remain 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 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) as possible.
3. When power-on reset function is in use.



**Figure 7.13 Unassigned Pin Handling**

## 8. Processor Mode

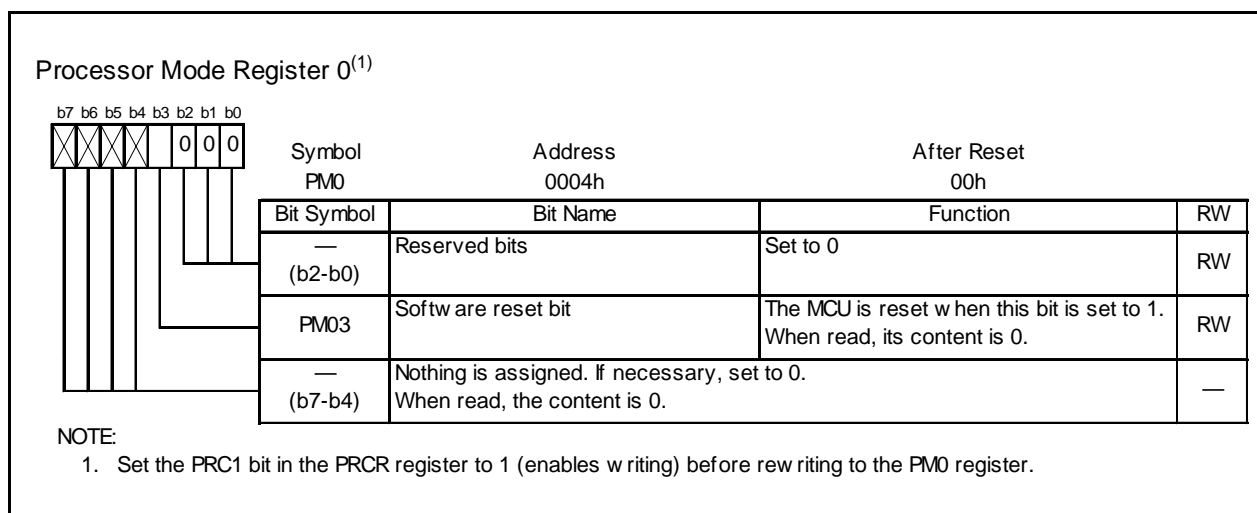
### 8.1 Processor Modes

Single-chip mode can be selected as processor mode.

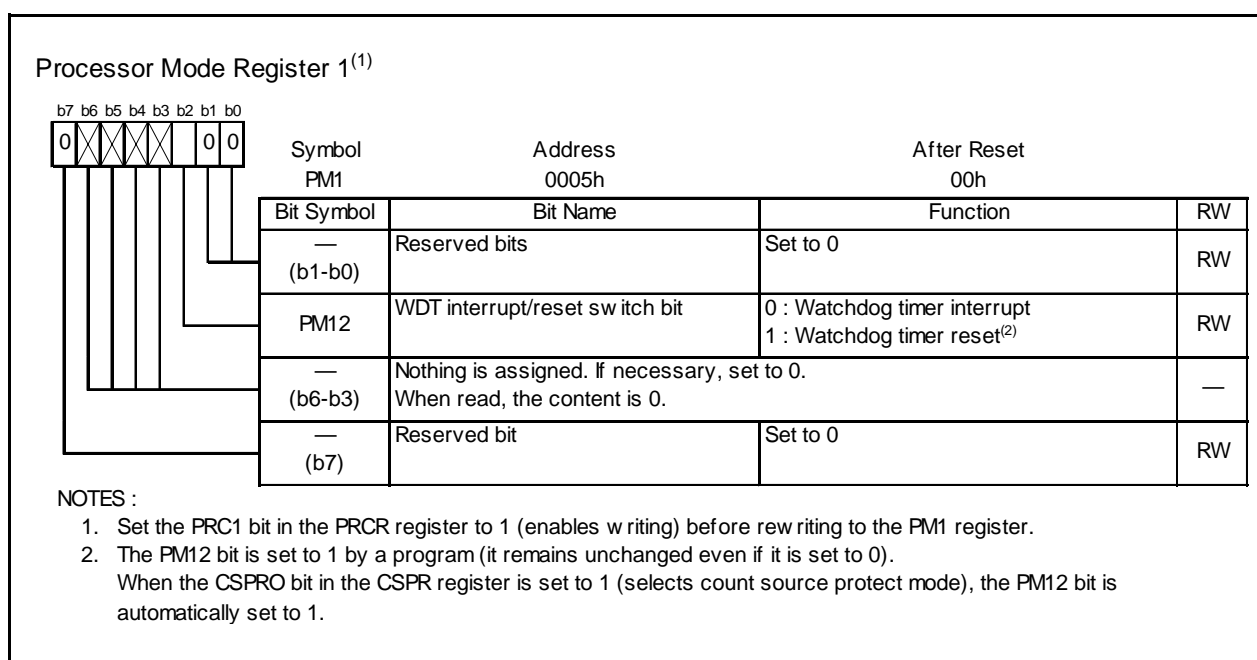
Table 8.1 lists Features of Processor Mode. Figure 8.1 shows the PM0 Register and Figure 8.2 shows the PM1 Register.

**Table 8.1 Features of Processor Mode**

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



**Figure 8.1 PM0 Register**



**Figure 8.2 PM1 Register**

## 9. Bus

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

Table 9.1 lists Bus Cycles by Access Space of the R8C/20 Group and Table 9.2 lists Bus Cycles by Access Space of the R8C/21 Group.

The ROM/RAM and SFR are connected to the CPU through an 8-bit bus. When accessing in word-(16 bits) unit, these area are accessed twice in 8-bit unit.

Table 9.3 lists Access Unit and Bus Operations.

**Table 9.1 Bus Cycles by Access Space of the R8C/20 Group**

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

**Table 9.2 Bus Cycles by Access Space of the R8C/21 Group**

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

**Table 9.3 Access Unit and Bus Operations**

Area	SFR, data flash	ROM (program ROM), RAM
Even address byte access		
Odd address byte access		
Even address word access		
Odd address word access		

However, only following SFRs are connected with the 16-bit bus:

Timer RD: registers TRDi (i = 0, 1), TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi

Therefore, they are accessed once in 16-bit units. The bus operation is the same as "Area: SFR, data flash, even address byte access" in Table 9.3 Access Unit and Bus Operations, and 16-bit data is accessed at a time.

## 10. Clock Generation Circuit

The clock generation circuit has:

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

Table 10.1 lists Specifications of Clock Generation Circuit. Figure 10.1 shows a Clock Generation Circuit. Figures 10.2 to 10.7 show clock-associated registers.

**Table 10.1 Specifications of Clock Generation Circuit**

Item	XIN Clock Oscillation Circuit	On-Chip Oscillator	
		High-Speed On-Chip Oscillator	Low-Speed On-Chip Oscillator
Use of Clock	<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> <li>• CPU and peripheral function clock sources when XIN clock stops oscillating</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> <li>• CPU and peripheral function clock sources when XIN clock stops oscillating</li> </ul>
Clock Frequency	0 to 20 MHz	Approx. 40 MHz <sup>(3)</sup>	Approx. 125 kHz
Connectable Oscillator	<ul style="list-style-type: none"> <li>• Ceramic resonator</li> <li>• Crystal oscillator</li> </ul>	–	–
Oscillator Connect Pins	XIN, XOUT <sup>(1)</sup>	– <sup>(1)</sup>	– <sup>(1)</sup>
Oscillation Stop, Restart Function	Usable	Usable	Usable
Oscillator Status After Reset	Stop	Stop	Oscillate
Others	Externally generated clock can be input <sup>(2)</sup>	–	–

**NOTES:**

1. These pins can be used as P4\_6 and P4\_7 when using the on-chip oscillator clock as the CPU clock while the XIN clock oscillation circuit is not used.
2. Set the CM05 bit in the CM0 register to 1 (main clock stops) and the CM13 bit in the CM1 register to 1 (XIN-XOUT pin) when the external clock is input.
3. The clock frequency is automatically set to up to 20 MHz by a driver when using the high-speed on-chip oscillator as the CPU clock source.



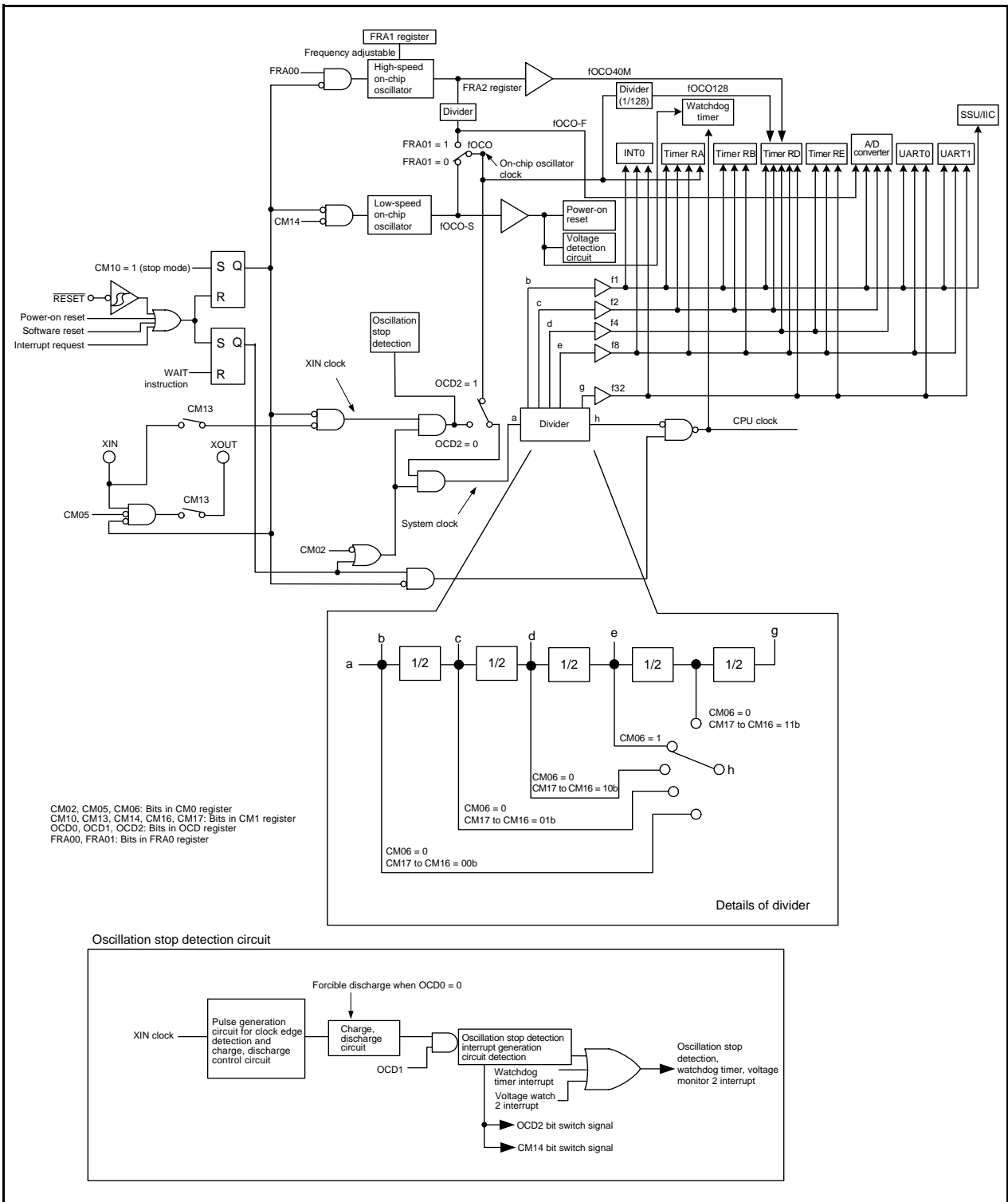


Figure 10.1 Clock Generation Circuit

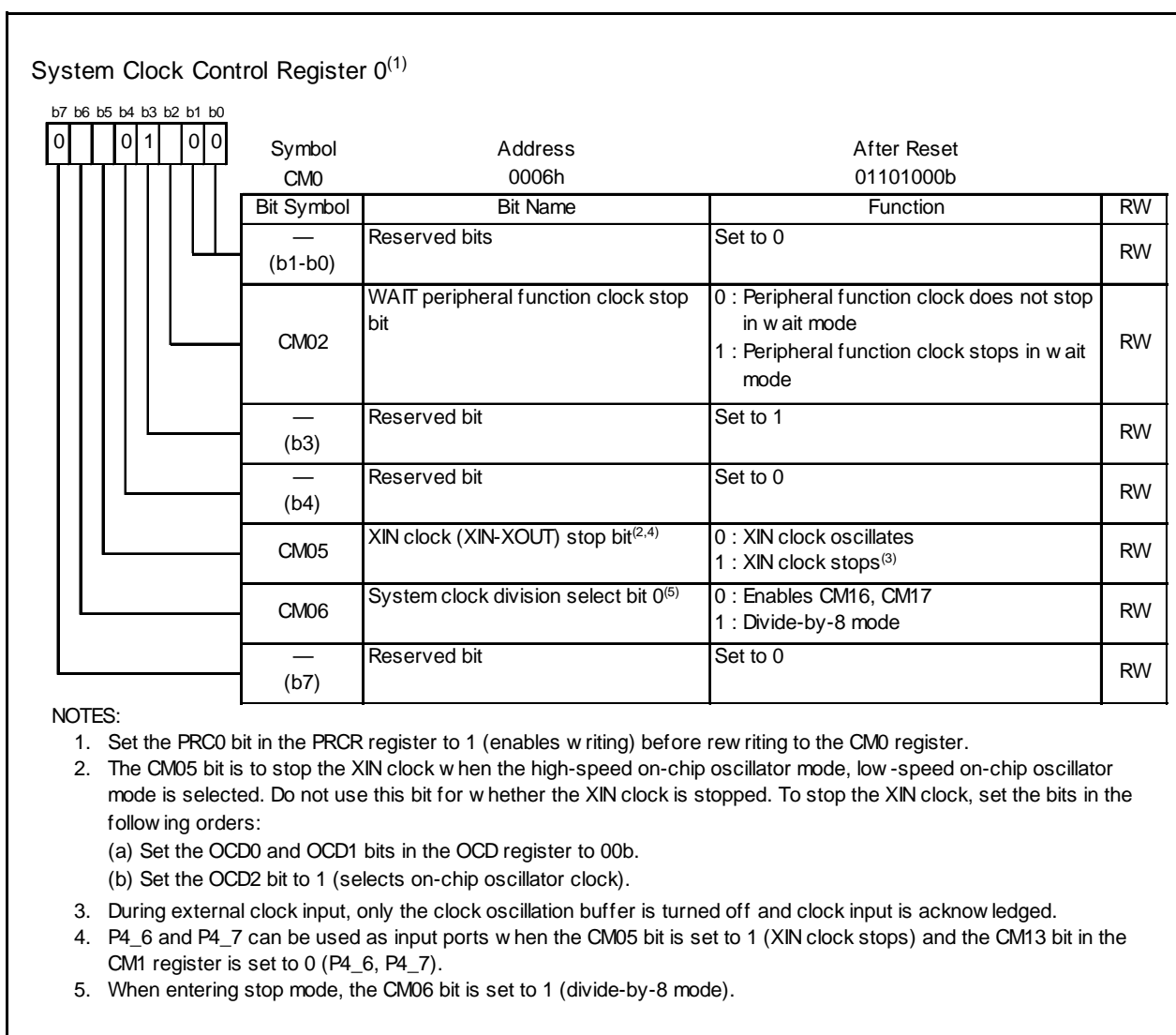


Figure 10.2 CM0 Register

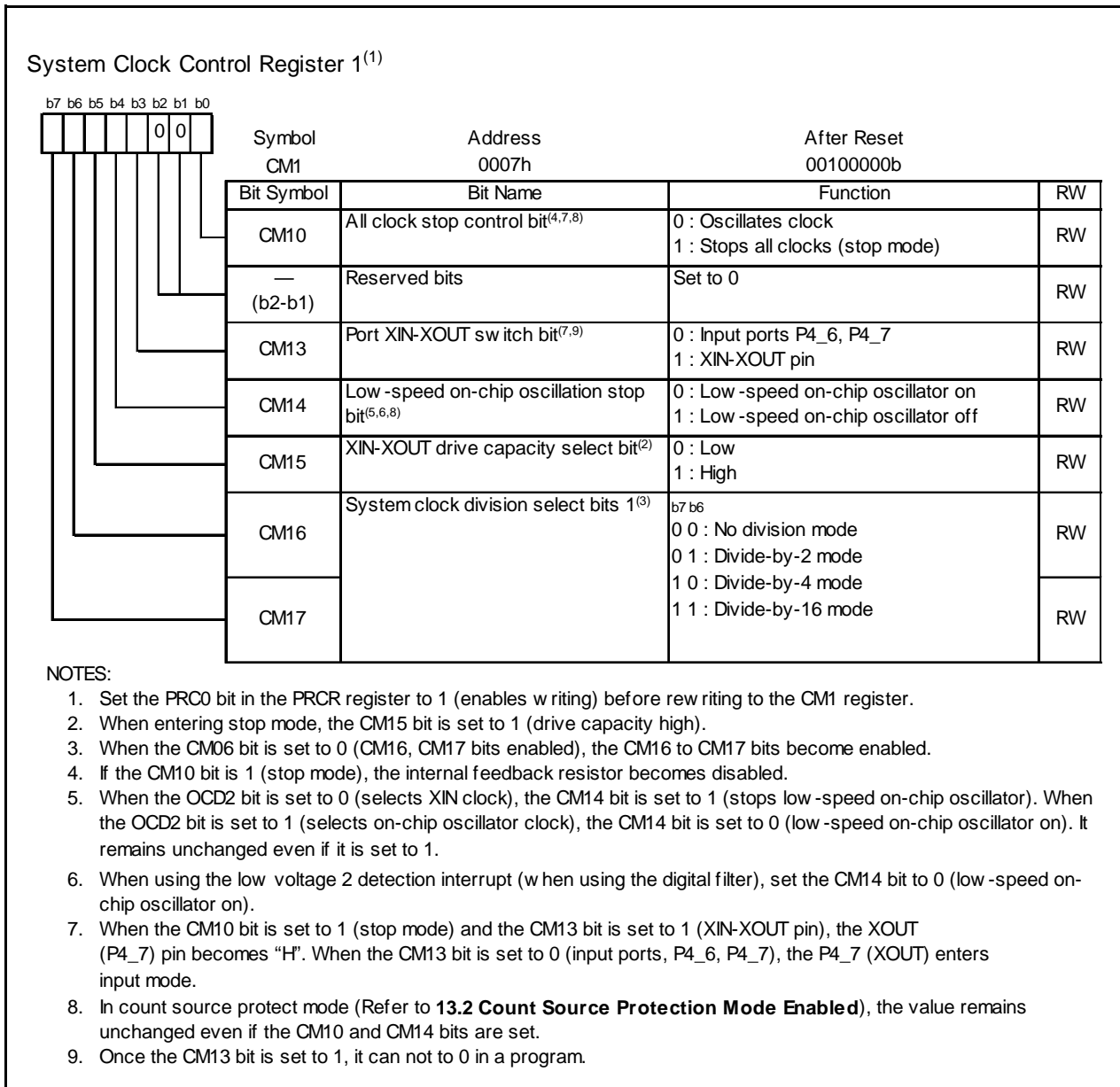


Figure 10.3 CM1 Register

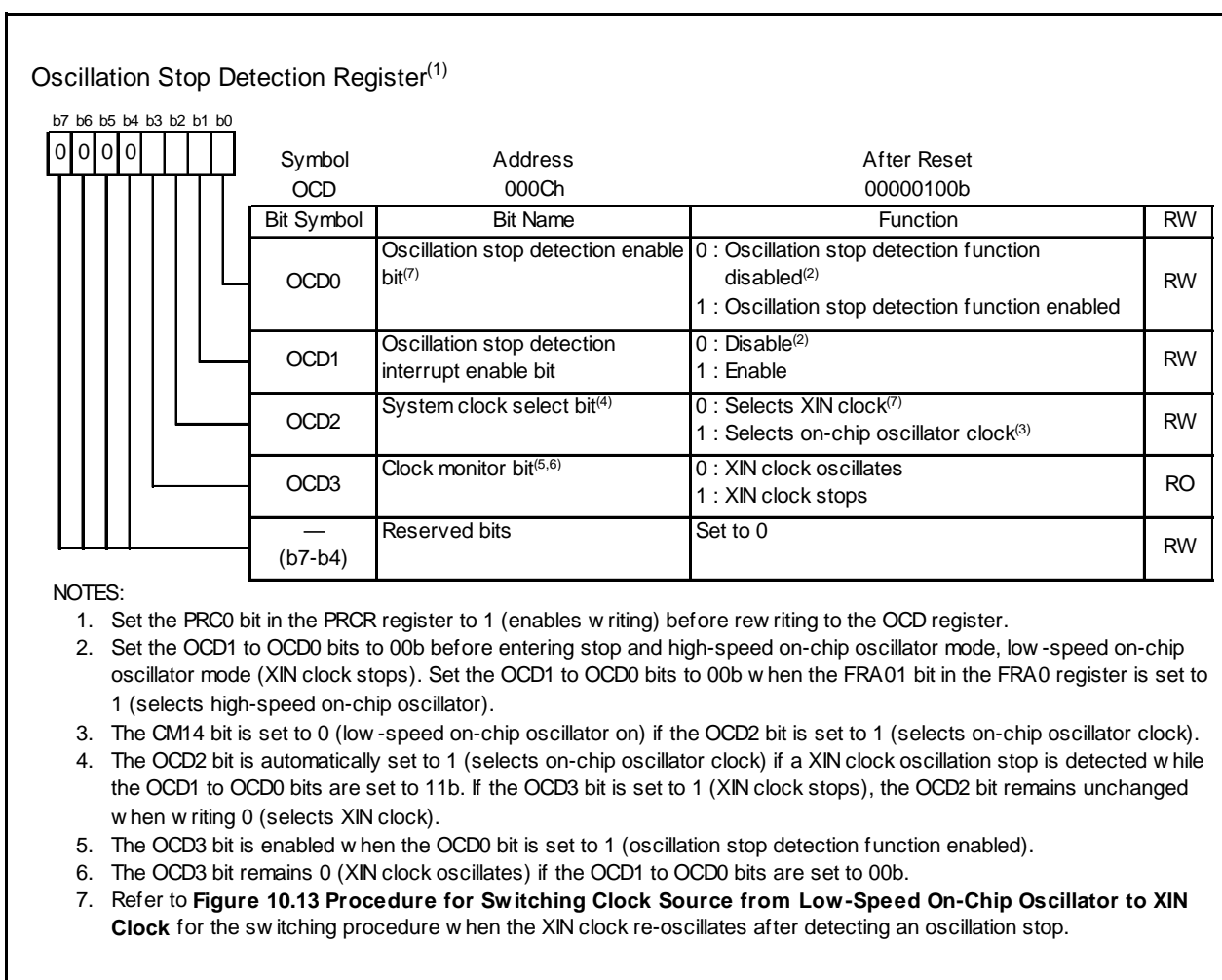


Figure 10.4 OCD Register

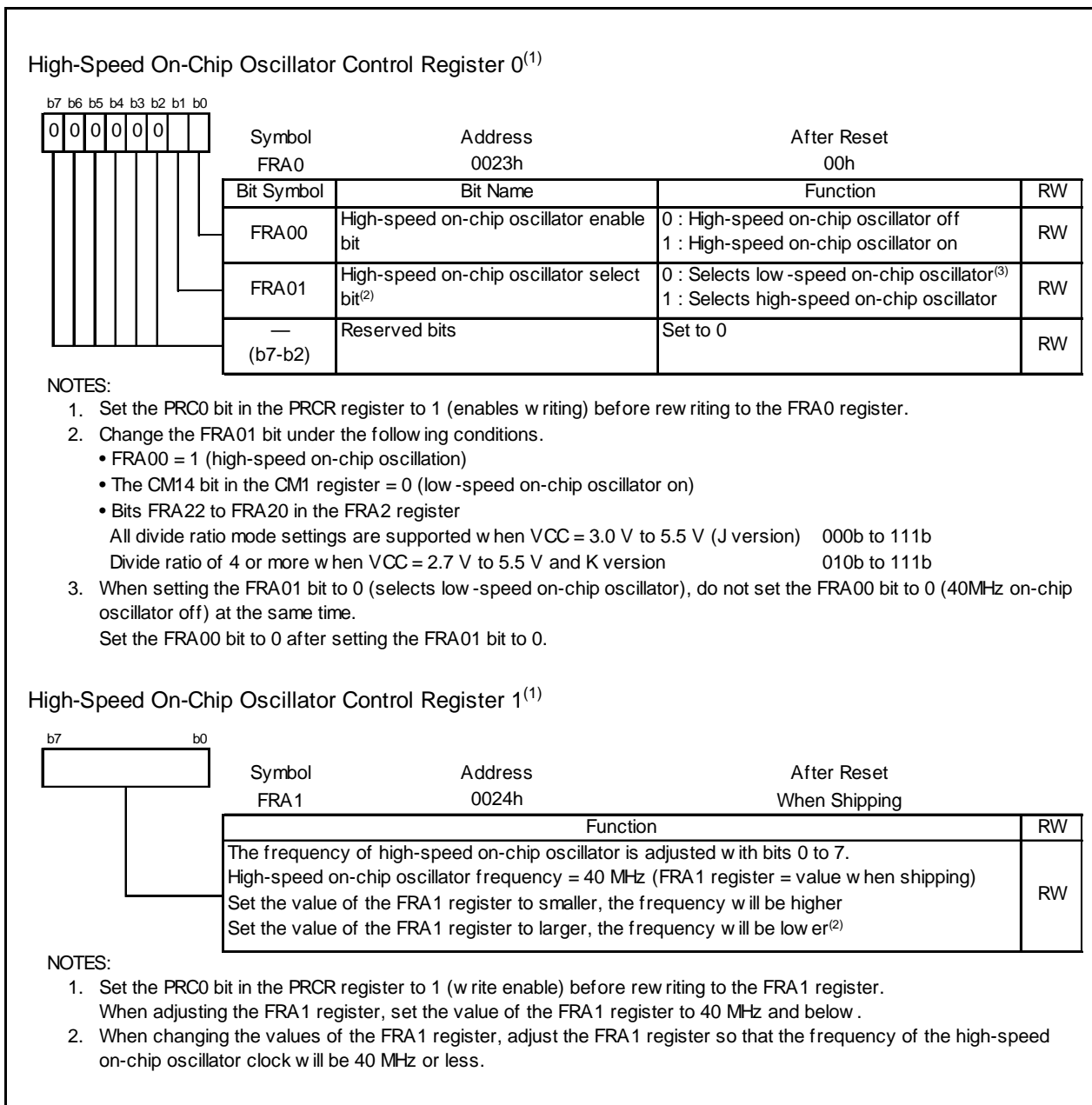


Figure 10.5 Registers FRA0 and FRA1

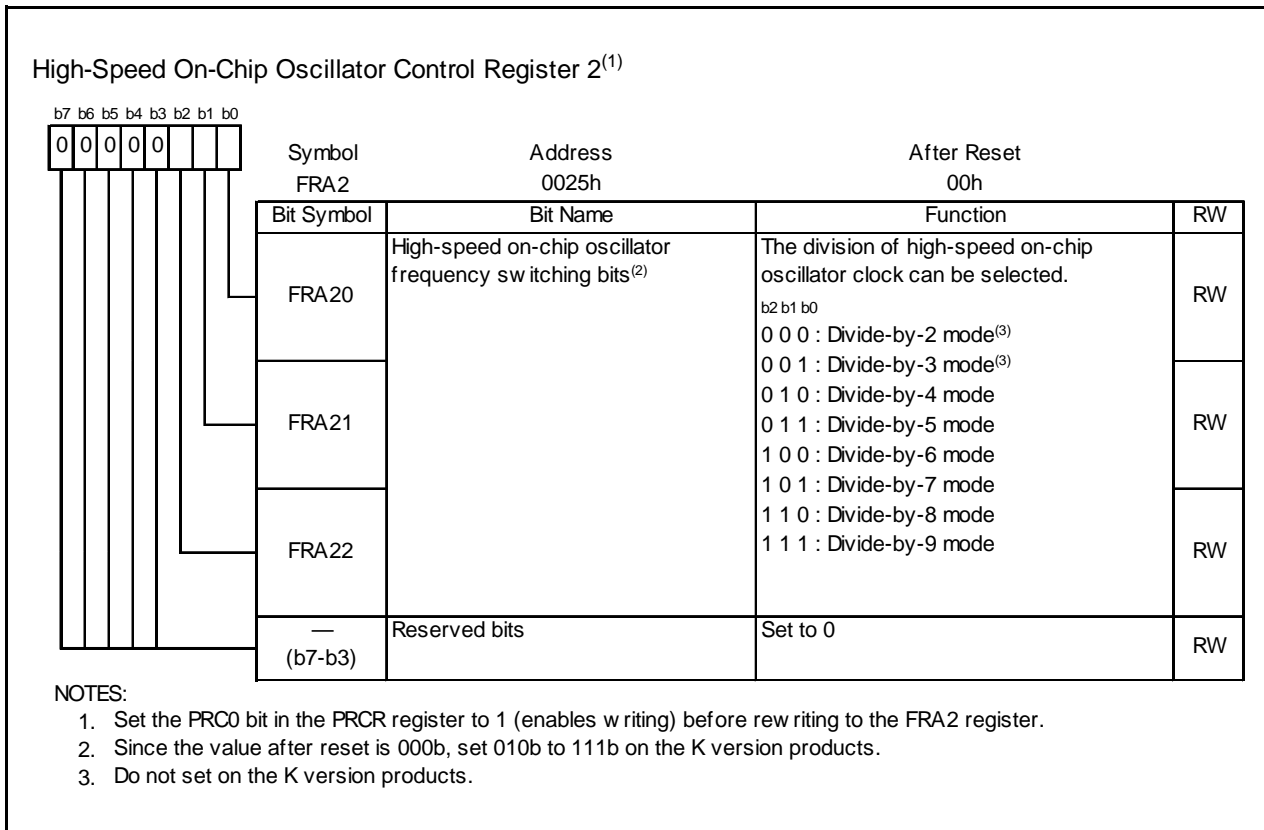


Figure 10.6 FRA2 Register

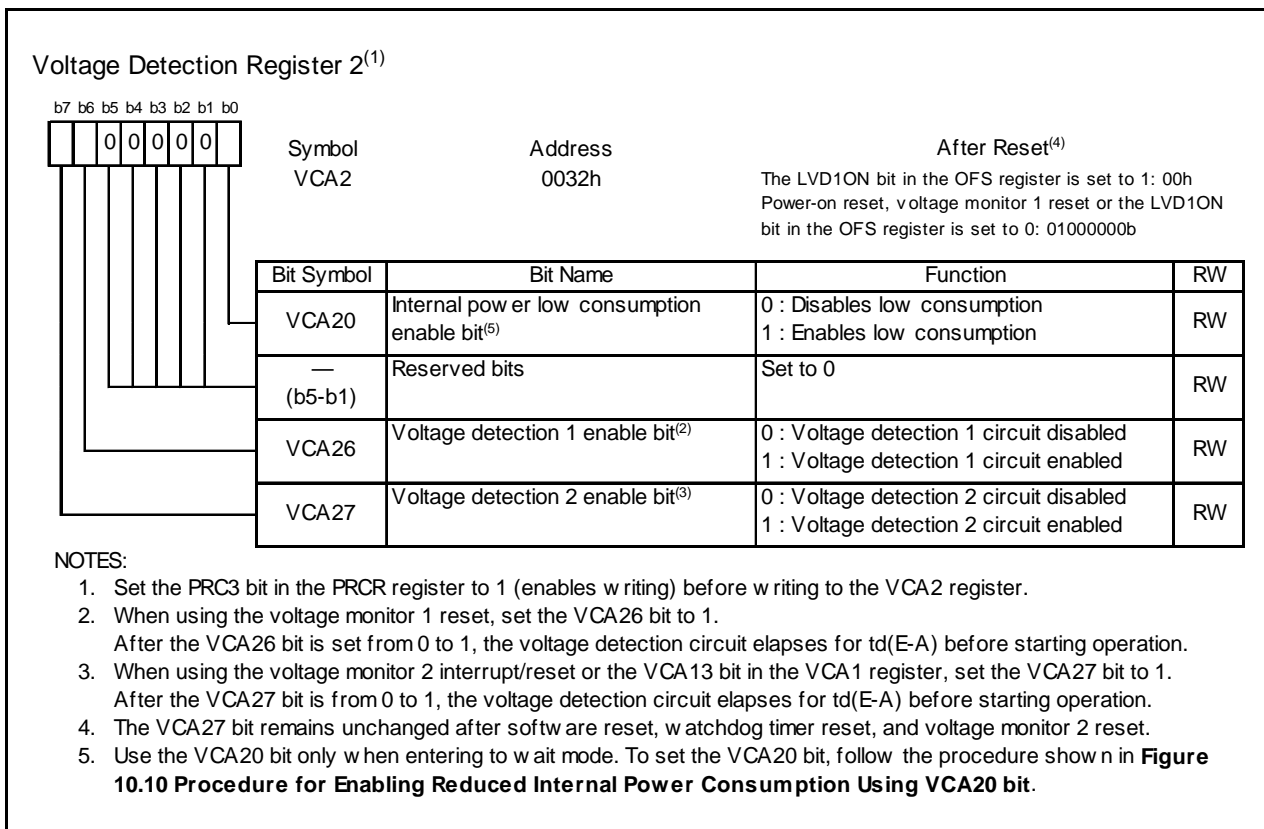


Figure 10.7 VCA2 Register

The following describes the clocks generated by the clock generation circuit.

### 10.1 XIN Clock

This clock is supplied by a XIN clock oscillation circuit. This clock is used as the clock source for the CPU and peripheral function clocks. The XIN clock oscillation circuit is configured by connecting a resonator between the XIN and XOUT pins. The XIN clock oscillation circuit contains 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 XIN clock oscillation circuit may also be configured by feeding an externally generated clock to the XIN pin.

Figure 10.8 shows Examples of XIN Clock Connection Circuit. During or after reset, the XIN clock stops.

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

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

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

When the clocks externally generated to the XIN pin are input, a XIN clock does not stop if setting the CM05 bit to 1. If necessary, use an external circuit to stop the clock.

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

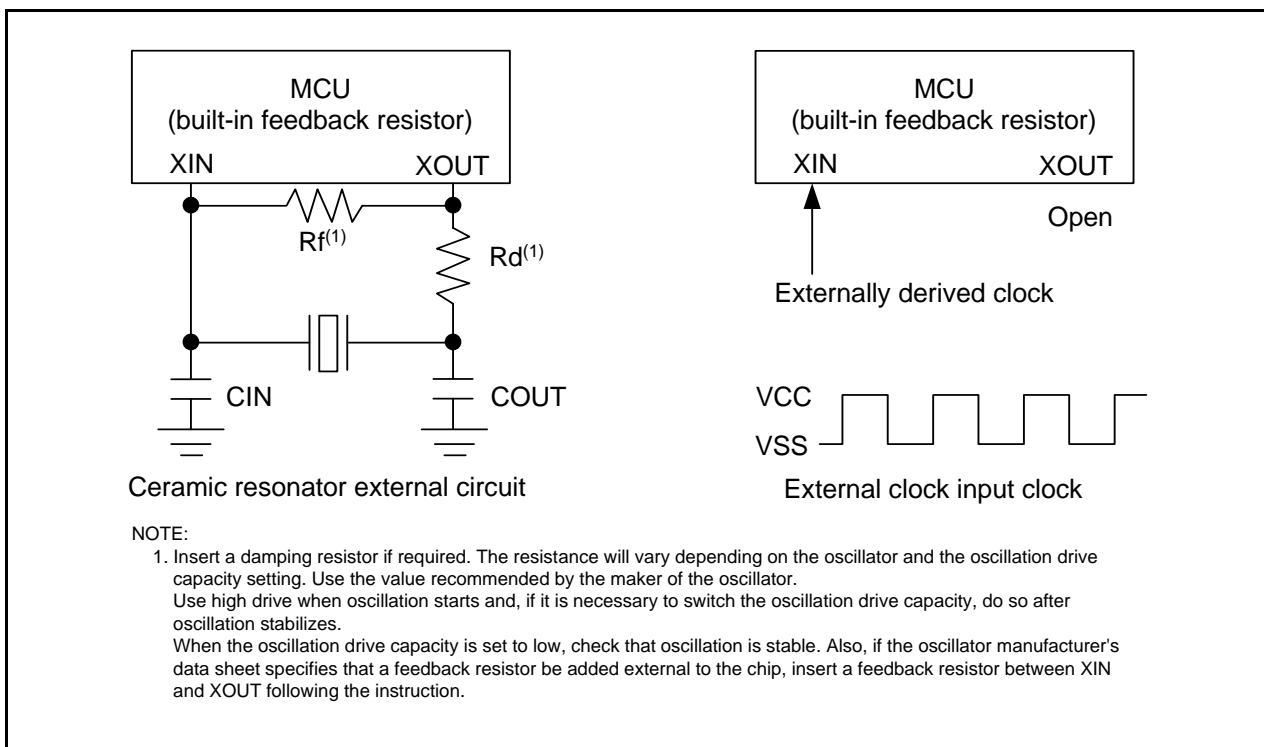


Figure 10.8 Examples of XIN Clock Connection Circuit

## 10.2 On-Chip Oscillator Clocks

This clock is supplied by an on-chip oscillator. The on-chip oscillator contains a high-speed on-chip oscillator and a low-speed on-chip oscillator. Either an on-chip oscillator clock is selected by the FRA01 bit in the FRA0 register.

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

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

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

If the XIN clock stops oscillating when the OCD1 to OCD0 bits in the OCD register are set to 11b, the low-speed on-chip oscillator automatically starts operating, supplying the necessary clock for the MCU.

The frequency of the low-speed on-chip oscillator varies depending on the supply voltage and the operating ambient temperature. The application products must be designed with sufficient margin to accommodate the frequency range.

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

The clock generated by the high-speed on-chip oscillator is used as the clock source for the CPU clock, peripheral function clock, fOCO, fOCO-F and fOCO40M.

To use the high-speed on-chip oscillator clock as the clock source of the CPU clock, peripheral clock, fOCO, and fOCO-F, set bits FRA20 to FRA22 in the FRA2 register as follows;

All divide ratio mode settings are supported when VCC = 3.0 V to 5.5 V (J version) 000b to 111b

Divide ratio of 4 or more when VCC = 2.7 V to 5.5 V and K version 010b to 111b

After reset, the on-chip oscillator clock generated by the high-speed on-chip oscillator stops. The oscillation starts by setting the FRA00 bit in the FRA0 register to 1 (high-speed on-chip oscillator on). The frequency can be adjusted by the FRA1 and FRA2 registers.

Since there are differences in the amount of frequency adjustment among the bits in the FRA1 register, make adjustments by changing the settings of individual bits.

Adjust the amount of high-speed on-chip oscillator frequency to 40 MHz and below by setting the FRA1 register.



### 10.3 CPU Clock and Peripheral Function Clock

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

#### 10.3.1 System Clock

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

#### 10.3.2 CPU Clock

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

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

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

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

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

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

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

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.

#### 10.3.4 fOCO

fOCO is operating clocks for the peripheral functions.

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

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

#### 10.3.5 fOCO40M

fOCO40M is used as the count source for the timer RD. The fOCO40M is generated by the high-speed on-chip oscillator and provided by setting the FRA00 bit to 1.

When the WAIT instruction is executed, the clock fOCO40M does not stop.

fOCO40M can be used with supply voltage  $VCC = 3.0$  to  $5.5V$ .

#### 10.3.6 fOCO-F

fOCO-F is used as the count source for the AD converter. The fOCO-F is generated by the high-speed on-chip oscillator and provided by setting the FRA00 bit to 1.

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

#### 10.3.7 fOCO-S

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

#### 10.3.8 fOCO128

fOCO128 is generated by fOCO divided-by-128. The clock fOCO128 is used for capture signal of timer RD (channel 0).

## 10.4 Power Control

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

### 10.4.1 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 after switching needs to be stabilized and oscillated. If the new clock source is the XIN clock, allow sufficient wait time in a program until an oscillation is stabilized before exiting.

**Table 10.2 Settings and Modes of Clock Associated Bits**

Modes		OCD Register	CM1 Register			CM0 Register		FRA0 Register	
		OCD2	CM17, CM16	CM14	CM13	CM06	CM05	FRA01	FRA00
High-speed clock mode	No division	0	00b	–	1	0	0	–	–
	Divide-by-2	0	01b	–	1	0	0	–	–
	Divide-by-4	0	10b	–	1	0	0	–	–
	Divide-by-8	0	–	–	1	1	0	–	–
	Divide-by-16	0	11b	–	1	0	0	–	–
High-speed 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	–

–: can be 0 or 1, no change in outcome.

#### 10.4.1.1 High-Speed Clock Mode

The XIN 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 FRA00 bit in the FRA0 register is set to 1 (high-speed on-chip oscillator on), the fOCO can be used for timers RA.

When the FRA00 bit is set to 1, fOCO40M can be used for timer RD.

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

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

The high-speed on-chip oscillator is used as the on-chip oscillator clock when the FRA00 bit in the FRA0 register is set to 1 (high-speed on-chip oscillator on) and the FRA01 bit in the FRA0 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) when transiting to high-speed clock mode.

If the FRA00 bit is set to 1, fOCO40M can be used for timer RD.

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

### 10.4.1.3 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 FRA01bit in the FRA0 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. Set the CM06 bit to 1 (divide-by-8 mode) when transiting to high-speed clock mode. When the FRA00 bit is set to 1, fOCO40M can be used for timer RD. When the CM14 bit is set to 0 (low-speed on-chip oscillator on), fOCO-S can be used for the watchdog timer and voltage detection circuit.

In this mode, stopping the XIN clock and 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.

When enabling reduced internal power consumption using the VCA20 bit, follow **Figure 10.10 Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit**.

### 10.4.2 Wait Mode

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

#### 10.4.2.1 Peripheral Function Clock Stop Function

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

#### 10.4.2.2 Entering Wait Mode

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

When the OCD2 bit in the OCD register is set to 1 (on-chip oscillator selected as system clock), set the OCD1bit in the OCD register to 0 (oscillation stop detection interrupt disabled) before executing the WAIT instruction.

If the MCU enters wait mode while the OCD1 bit is set to 1 (oscillation stop detection interrupt enabled), current consumption is not reduced because the CPU clock does not stop.

#### 10.4.2.3 Pin Status in Wait Mode

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

### 10.4.2.4 Exiting Wait Mode

The MCU exits wait mode by a hardware reset or peripheral function interrupt. When using a hardware reset to exit wait mode, set the ILVL2 to ILVL0 bits for the peripheral function interrupts to 000b (interrupts disabled) before executing the WAIT instruction.

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

Table 10.3 lists Interrupts to Exit Wait Mode and Usage Conditions.

**Table 10.3 Interrupts to Exit Wait Mode and Usage Conditions**

Interrupt	CM02 = 0	CM02 = 1
Serial Interface Interrupt	Usable when operating with internal or external clock	Usable when operating with external clock
Clock Synchronous Serial I/O with Chip Select Interrupt / I <sup>2</sup> C Bus Interface Interrupt	Usable in all modes	– (Do not use)
Key Input Interrupt	Usable	Usable
A/D Conversion Interrupt	Usable in one-shot mode	– (Do not use)
Timer RA Interrupt	Usable in all modes	Can be used if there is no filter in event counter mode. Usable by selecting fOCO as count source.
Timer RB Interrupt	Usable in all modes	– (Do not use)
Timer RD Interrupt	Usable in all modes	Usable by selecting fOCO40M as count source
Timer RE Interrupt	Usable in all modes	– (Do not use)
INT Interrupt	Usable	Usable (INT0 to INT3 can be used if there is no filter.)
Voltage Monitor 2 Interrupt	Usable	Usable
Oscillation Stop Detection Interrupt	Usable	– (Do not use)

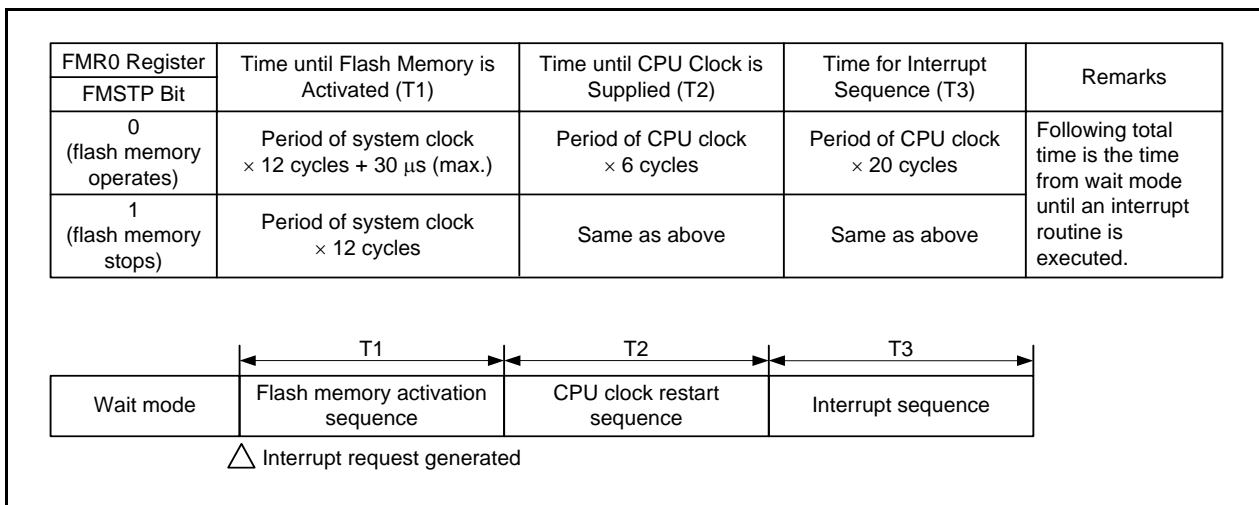
Figure 10.9 shows the Time from Wait Mode to Interrupt Routine Execution.

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 to the ILVL2 to ILVL0 bits in the interrupt control register of the peripheral function interrupts to use for exiting wait mode. Set the ILVL2 to ILVL0 bits of the peripheral function interrupts not to use for exiting wait mode to 000b (disables interrupt).
- (2) Set the I flag to 1.
- (3) Operate the peripheral function to use 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 10.9.

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.



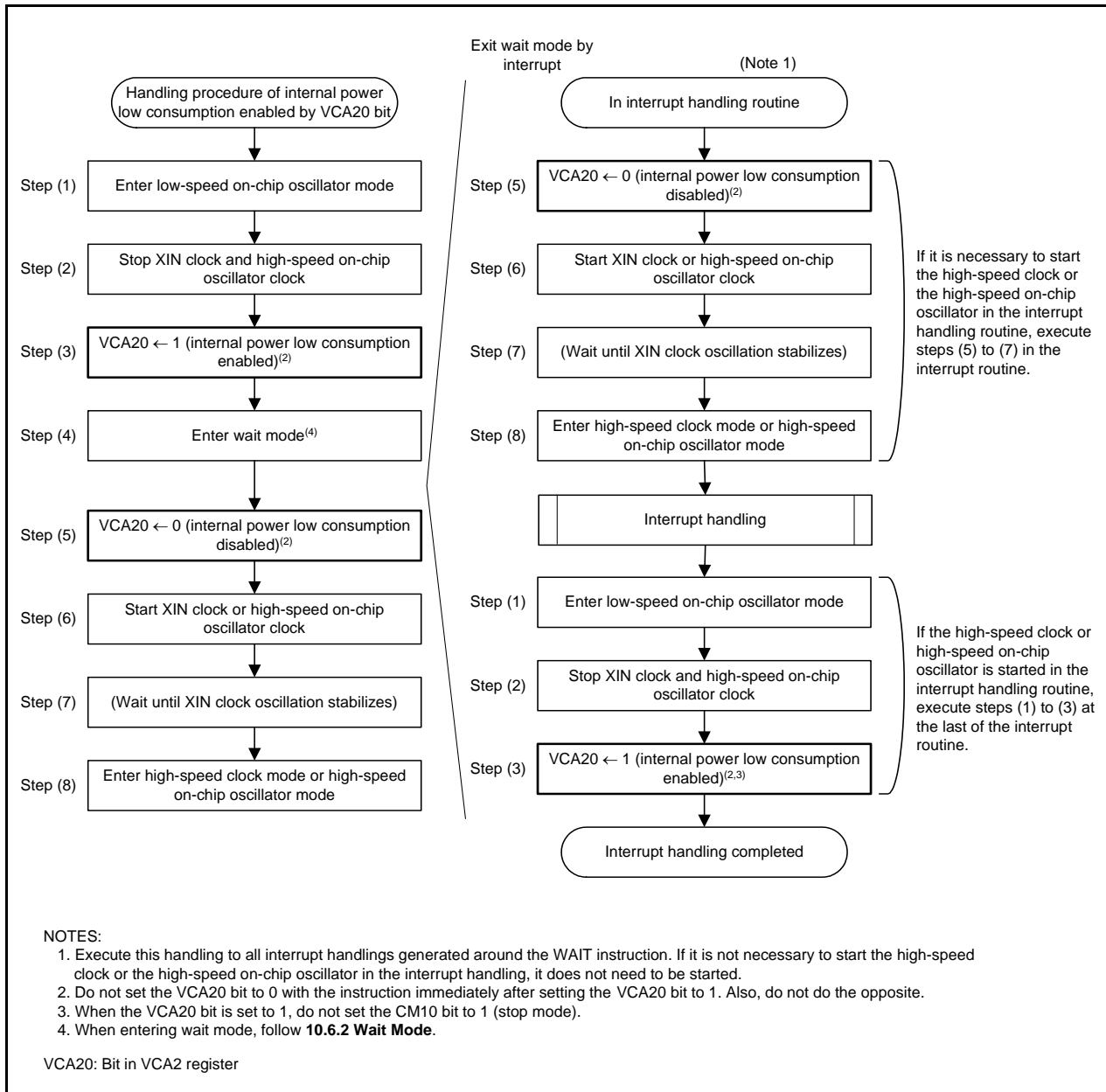
**Figure 10.9 Time from Wait Mode to Interrupt Routine Execution**

### 10.4.2.5 Reducing Internal Power Consumption

Internal power consumption can be reduced by using low-speed on-chip oscillator mode.

Figure 10.10 shows the Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit.

When enabling reduced internal power consumption using the VCA20 bit, follow **Figure 10.10 Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit**.



**Figure 10.10 Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit**

### 10.4.3 Stop Mode

Since the oscillator circuits stop in wait mode, the CPU clock and peripheral function clock stop and the CPU and peripheral functions clocked by 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 internal RAM is maintained.

The peripheral functions clocked by external signals maintain operating.

Table 10.4 lists Interrupts to Exit Stop Mode and Usage Conditions.

**Table 10.4 Interrupts to Exit Stop Mode and Usage Conditions**

Interrupt	Usage Conditions
Key Input Interrupt	–
$\overline{\text{INT0}}$ to $\overline{\text{INT3}}$ 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 2 Interrupt	Usable in digital filter disabled mode (VW2C1 bit in VW2C register is set to 1)

#### 10.4.3.1 Entering Stop Mode

The MCU enters stop mode by setting the CM10 bit in the CM1 register to 1 (all clocks stop). At the same time, the CM06 bit in the CM0 register is set to 1 (divide-by-8 mode) and the CM15 bit in the CM10 register is set to 1 (drive capability HIGH of XIN clock oscillator circuit).

When using stop mode, set the OCD1 to OCD0 bits to 00b before entering stop mode.

#### 10.4.3.2 Pin Status in Stop Mode

The status before entering wait mode is maintained.

However, when the CM13 bit in the CM1 register is set to 1 (XIN-XOUT pins), the XOUT(P4\_7) pin is held "H". When the CM13 bit is set to 0 (input port P4\_6 and P4\_7), the P4\_7(XOUT) is held in input status.

#### 10.4.3.3 Exiting Stop Mode

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

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

Figure 10.11 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 to the ILVL2 to ILVL0 bits of the peripheral function interrupts to use for exiting stop mode. Set the ILVL2 to ILVL0 bits of the peripheral function interrupts not to use for exiting stop mode to 000b (disables interrupt).
- (2) Set the I flag to 1.
- (3) Operates the peripheral function to use 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.

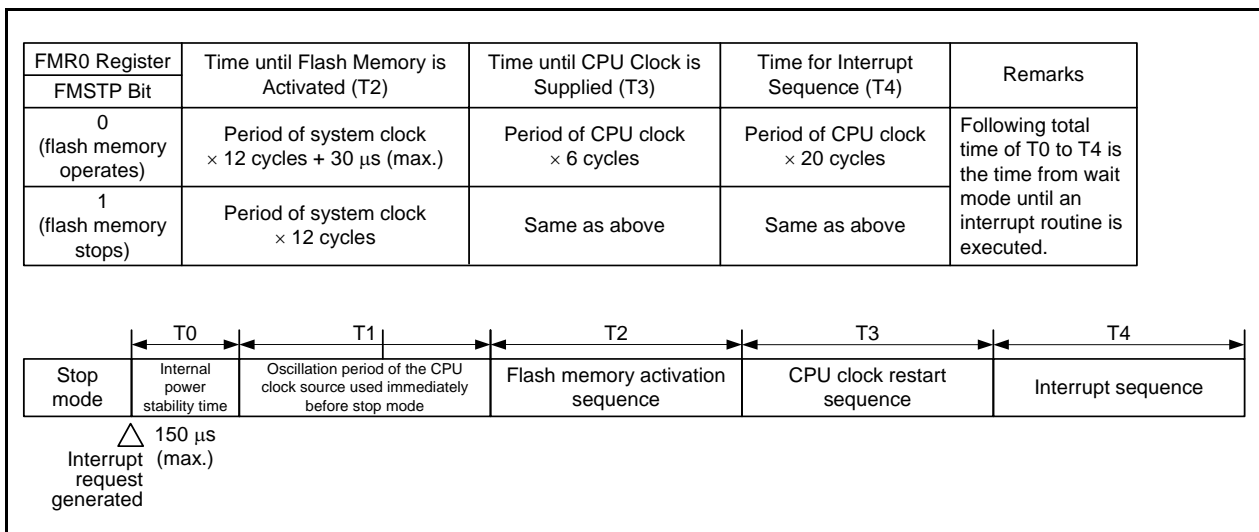


Figure 10.11 Time from Stop Mode to Interrupt Routine Execution



Figure 10.12 shows the State Transitions in Power Control Mode.

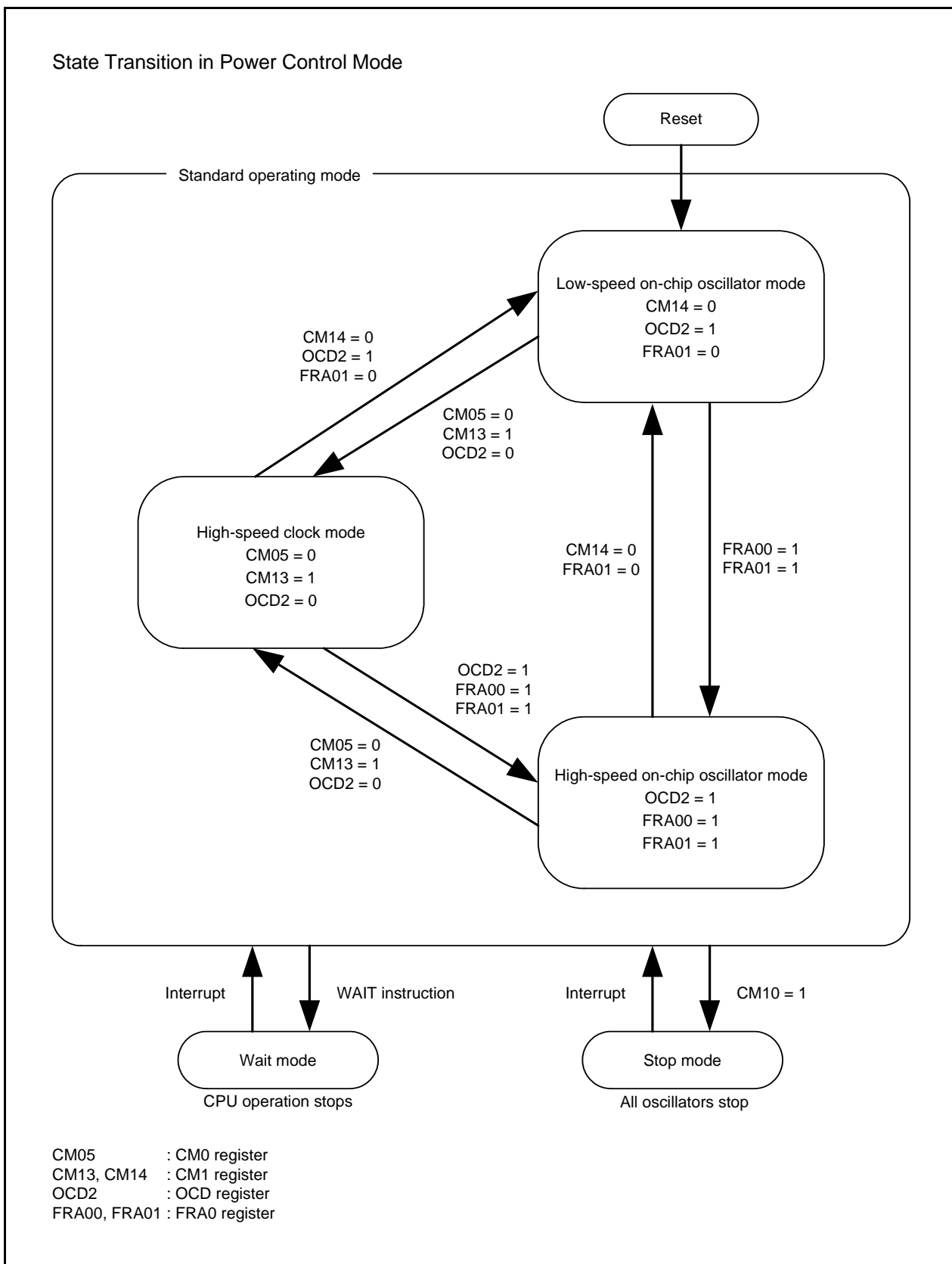


Figure 10.12 State Transitions in Power Control Mode

## 10.5 Oscillation Stop Detection Function

The oscillation stop detection function is a function to detect the stop of the XIN clock oscillating circuit. The oscillation stop detection function can be enabled and disabled by the OCD0 bit in the OCD register.

Table 10.5 lists the Specifications of Oscillation Stop Detection Function.

When the XIN clock is the CPU clock source and the OCD1 to OCD0 bits are set to 11b, the system is placed in the following state if the XIN clock stops.

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

**Table 10.5 Specifications of Oscillation Stop Detection Function**

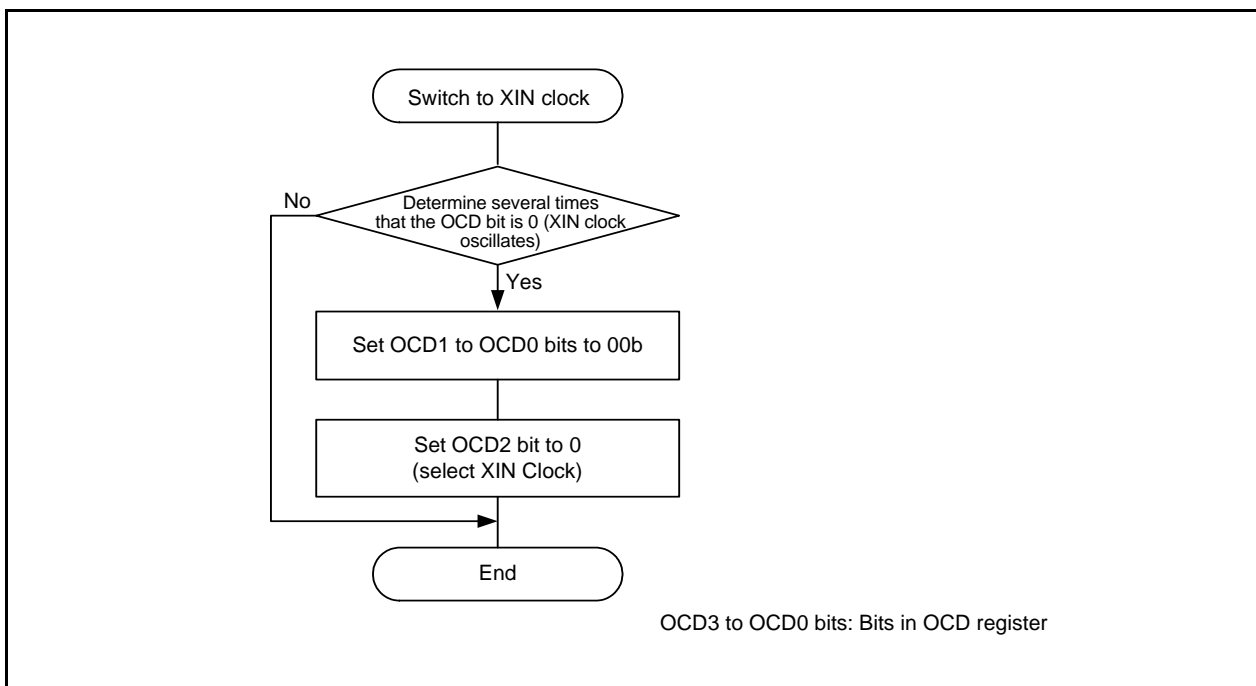
Item	Specification
Oscillation Stop Detection Clock and Frequency Bandwidth	$f(\text{XIN}) \geq 2 \text{ MHz}$
Enabled Condition for Oscillation Stop Detection Function	Set OCD1 to OCD0 bits to 11b
Operation at Oscillation Stop Detection	Oscillation stop detection interrupt is generated

### 10.5.1 How to Use Oscillation Stop Detection Function

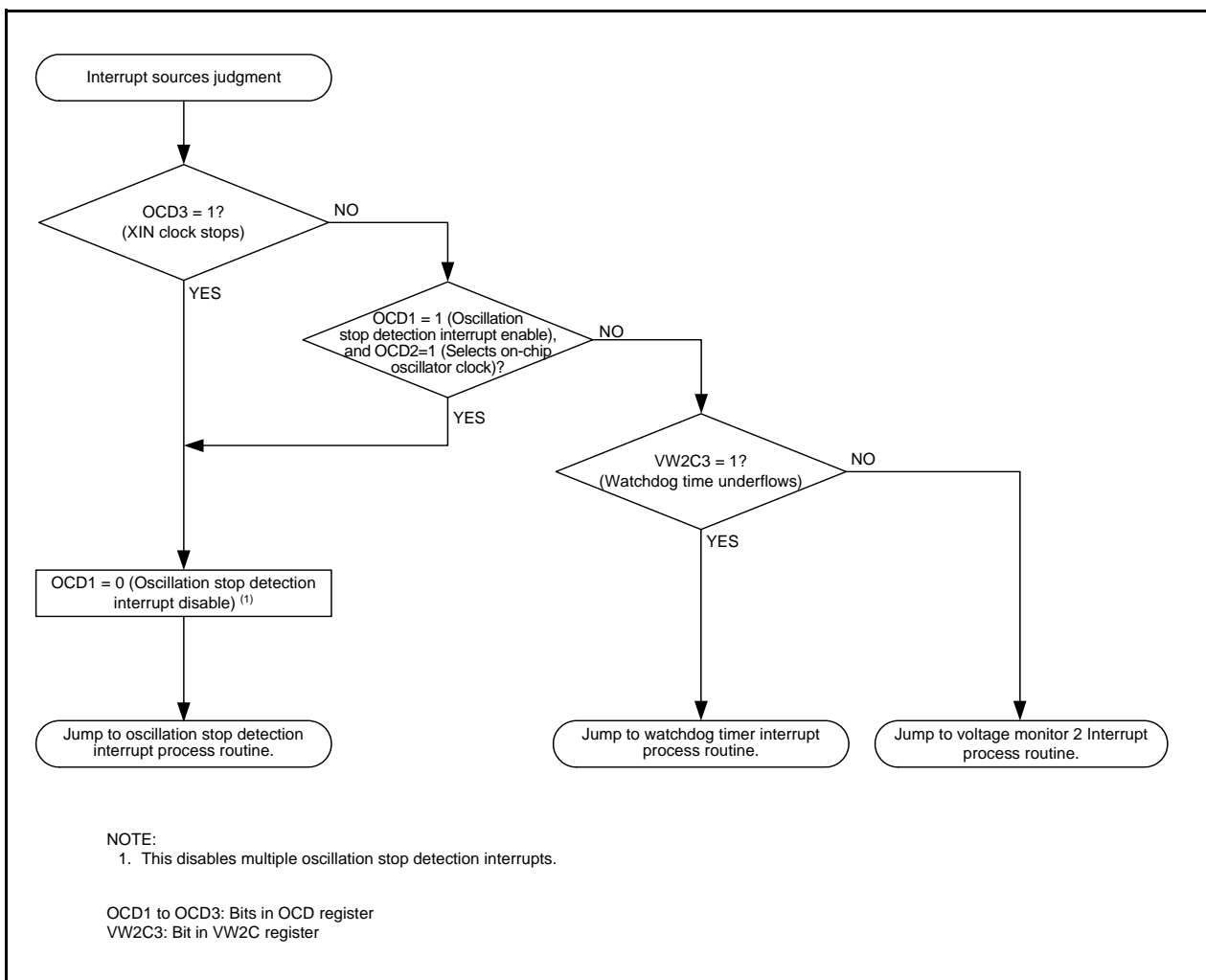
- The oscillation stop detection interrupt shares the vector with the voltage monitor 2 interrupt and the watchdog timer interrupt. When using the oscillation stop detection interrupt and watchdog timer interrupt, the interrupt cause needs to be determined.  
Table 10.6 lists the Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, Voltage Monitor 1, and Voltage Monitor 2 Interrupts.  
Figure 10.14 shows an Example of Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, Voltage Monitor 1, or Voltage Monitor 2 Interrupt.
- When the XIN clock is re-oscillated after oscillation stop, switch the XIN clock to the clock source of the CPU clock and peripheral functions by a program.  
Figure 10.13 shows the Procedure for Switching Clock Source from Low-Speed On-Chip Oscillator to XIN Clock.
- To enter wait mode while using the oscillation stop detection function, set the CM02 bit to 0 (peripheral function clock does not stop in wait mode).
- Since the oscillation stop detection function is a function preparing to stop the XIN clock by the external cause, set the OCD1 to OCD0 bits to 00b when the XIN clock stops or oscillates in the program, that is stop mode is selected or the CM05 bit is changed.
- This function cannot be used when the XIN clock frequency is less than 2 MHz. Set the OCD1 to OCD0 bits to 00b.
- When using the low-speed on-chip oscillator clock for the CPU clock and clock sources of peripheral functions after detecting the oscillation stop, set the FRA01 bit in the FRA0 register to 0 (low-speed on-chip oscillator selected) and the OCD1 to OCD0 bits to 11b.  
When using the high-speed on-chip oscillator clock for the CPU clock and clock sources of peripheral functions after detecting the oscillation stop, set the FRA00 bit to 1 (high-speed on-chip oscillator on) and the FRA01 bit to 1 (high-speed on-chip oscillator selected) and the OCD1 to OCD0 bits to 11b.

**Table 10.6 Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, Voltage Monitor 1, and Voltage Monitor 2 Interrupts**

Generated Interrupt Source	Bit Showing Interrupt Cause
Oscillation Stop Detection ((a) or (b))	(a) OCD3 bit in OCD register = 1
	(b) OCD1 to OCD0 bits in OCD register = 11b and the OCD2 bit = 1
Watchdog Timer	VW2C3 bit in VW2C register = 1
Voltage Monitor 2	VW2C2 bit in VW2C register = 1



**Figure 10.13 Procedure for Switching Clock Source from Low-Speed On-Chip Oscillator to XIN Clock**



**Figure 10.14 Example of Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, Voltage Monitor 1, or Voltage Monitor 2 Interrupt**

## 10.6 Notes on Clock Generation Circuit

### 10.6.1 Stop Mode

When entering stop mode, set the FMR01 bit to “0” (CPU rewrite mode disabled) and the CM10 bit to “1” (stop mode). An instruction queue pre-reads 4 bytes from the instruction which sets the CM10 bit in the CM1 register to “1” (stop mode) and the program stops. Insert at least 4 NOP instructions following the JMP.B instruction immediately after the instruction which sets the CM10 bit to “1”.

- 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

```

### 10.6.2 Wait Mode

When entering wait mode, set the FMR01 bit 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.

- Example to execute the WAIT instruction

```

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

```

### 10.6.3 Oscillation Stop Detection Function

Since the oscillation stop detection function cannot be used if the XIN clock frequency is less than 2 MHz, set the OCD1 to OCD0 bits to 00b.

### 10.6.4 Oscillation Circuit Constants

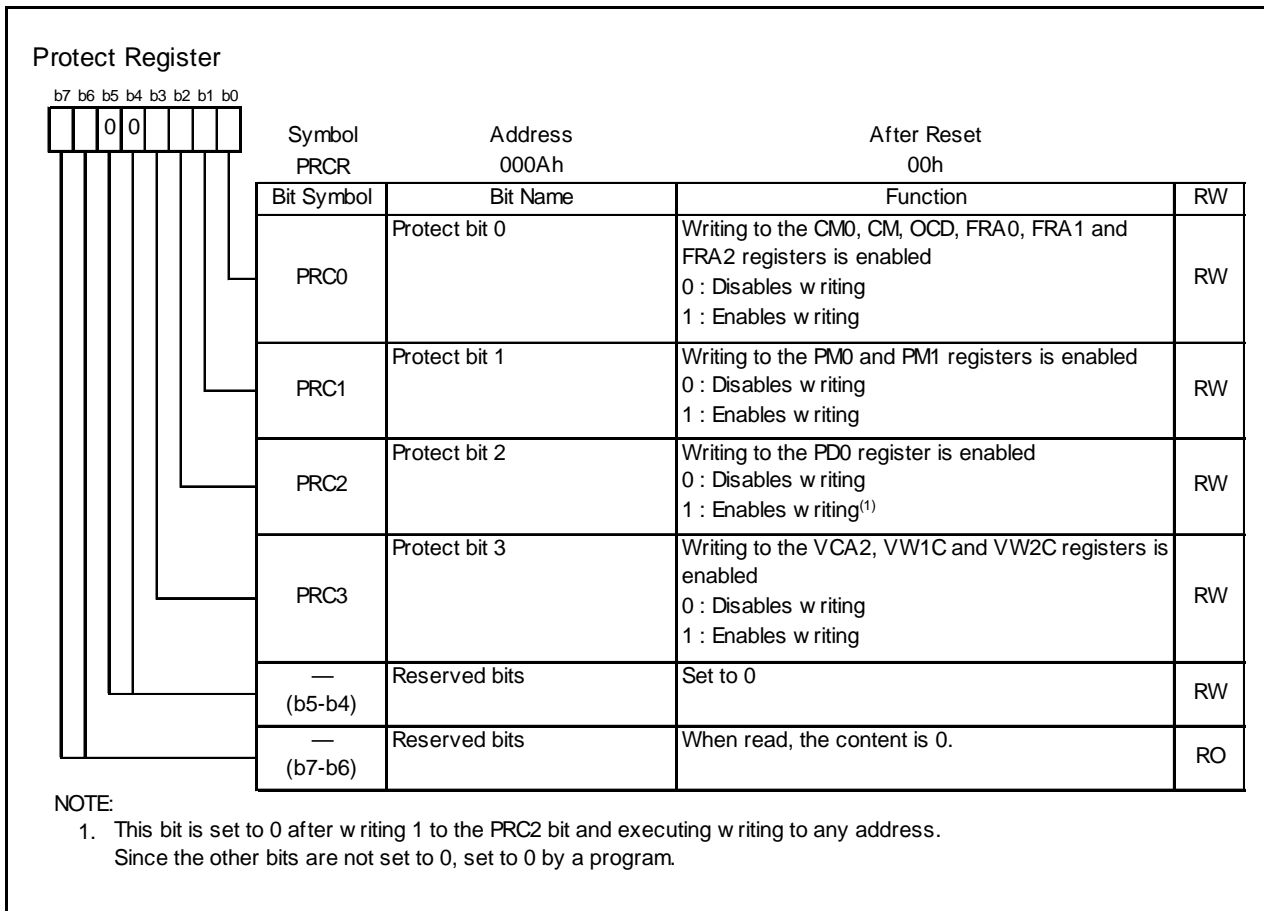
Ask the maker of the oscillator to specify the beat oscillation circuit constants on your system.

## 11. Protection

Protection function protects important registers from being easily overwritten when a program runs out of control.

Figure 11.1 shows the PRCR Register. The following lists the registers protected by the PRCR register.

- Registers protected by PRC0 bit: CM0, CM1, OCD, FRA0, FRA1 and FRA2 registers
- Registers protected by PRC1 bit: PM0 and PM1 registers
- Registers protected by PRC2 bit: PD0 register
- Registers protected by PRC3 bit: VCA2, VW1C and VW2C registers



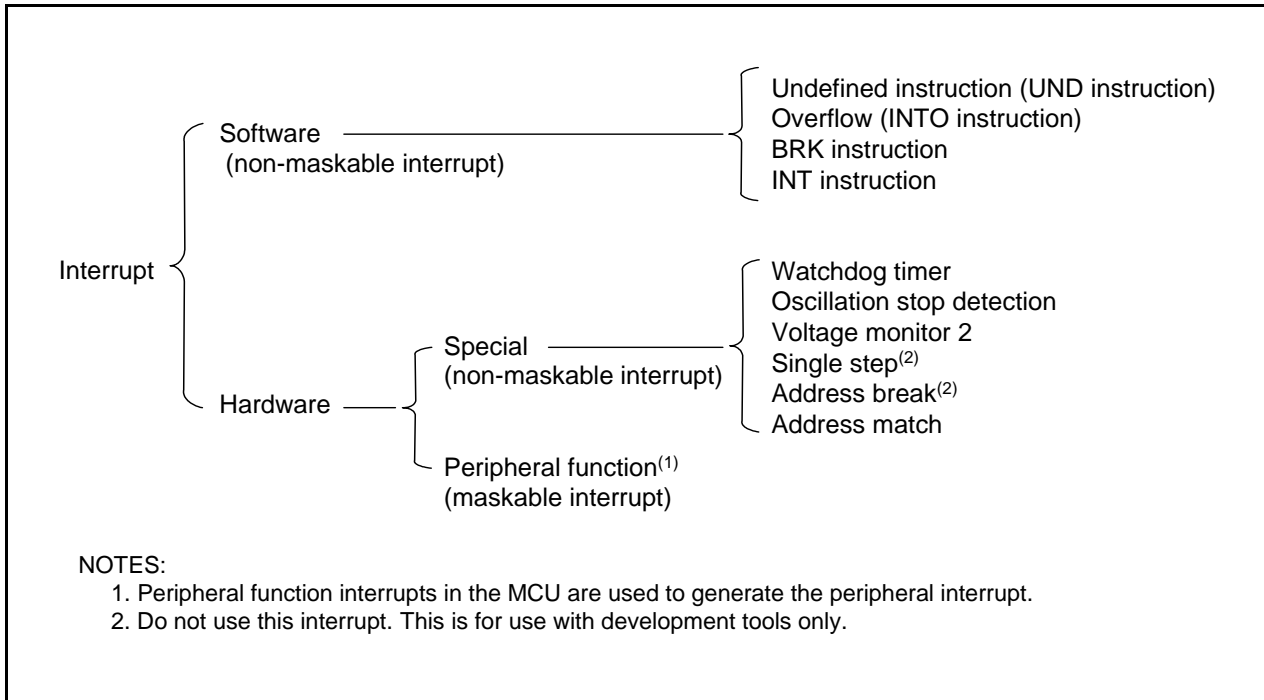
**Figure 11.1 PRCR Register**

## 12. Interrupts

### 12.1 Interrupt Overview

#### 12.1.1 Types of Interrupts

Figure 12.1 shows the Interrupts.



**Figure 12.1** Interrupts

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

## 12.1.2 Software Interrupts

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

### 12.1.2.1 Undefined Instruction Interrupt

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

### 12.1.2.2 Overflow Interrupt

The overflow interrupt is generated when the O flag is set to 1 (arithmetic operation overflow) and the INTO instruction is executed. Instructions to set the O flag are:

ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB

### 12.1.2.3 BRK Interrupt

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

### 12.1.2.4 INT Instruction Interrupt

An INT instruction interrupt is generated when the INT instruction is executed. The INT instruction can select software interrupt numbers 0 to 63. Software interrupt numbers 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. In software interrupt numbers 0 to 31, the U flag is saved to the stack during instruction execution and set the U flag to 0 (ISP selected) before executing an interrupt sequence. The U flag is restored from the stack when returning from the interrupt routine. In software interrupt numbers 32 to 63, the U flag does not change state during instruction execution, and the selected SP is used.



### 12.1.3 Special Interrupts

Special interrupts are non-maskable interrupts.

#### 12.1.3.1 Watchdog Timer Interrupt

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

#### 12.1.3.2 Oscillation Stop Detection Interrupt

Oscillation Stop Detection Interrupt is generated by the oscillation stop detection function. For details of the oscillation stop detection function, refer to **10. Clock Generation Circuit**.

#### 12.1.3.3 Voltage Monitor 2 Interrupt

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

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

Do not use the single-step interrupt. For development tools only.

#### 12.1.3.5 Address Match Interrupt

The address match interrupt is generated immediately before executing an instruction that is stored into an address indicated by the RMAD0 to RMAD1 registers when the AIER0 or AIER1 bit in the AIER register which is set to 1 (address match interrupt enable).

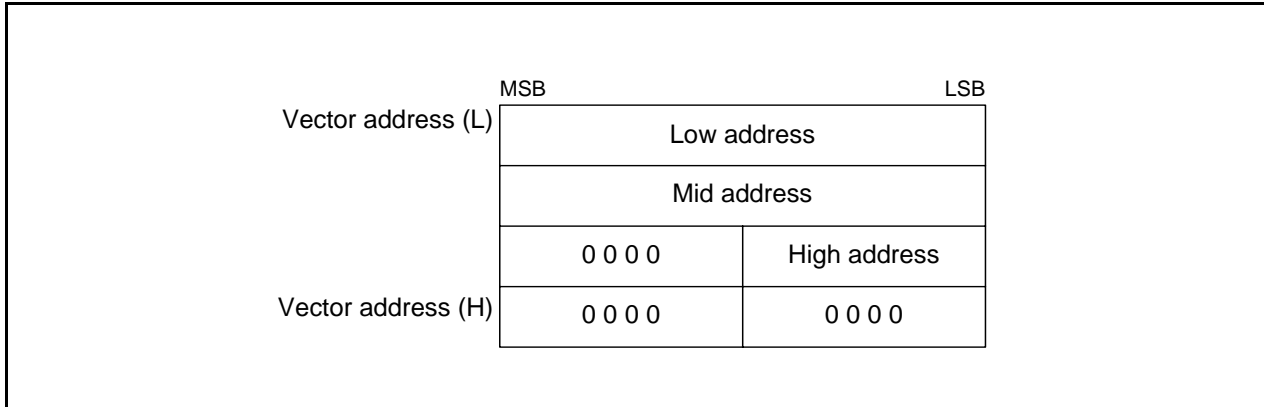
For details of the address match interrupt, refer to **12.4 Address Match Interrupt**.

### 12.1.4 Peripheral Function Interrupt

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

### 12.1.5 Interrupts and Interrupt Vector

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



**Figure 12.2** Interrupt Vector

#### 12.1.5.1 Fixed Vector Tables

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

Table 12.1 lists the Fixed Vector Tables. The vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to **19.3 Functions to Prevent Rewriting of Flash Memory**.

**Table 12.1** Fixed Vector Tables

Interrupt Source	Vector Addresses Address (L) to (H)	Remarks	Reference
Undefined Instruction	0FFDCh to 0FFDFh	Interrupt on UND instruction	R8C/Tiny Series software manual
Overflow	0FFE0h to 0FFE3h	Interrupt on INTO instruction	
BRK Instruction	0FFE4h to 0FFE7h	If the content of address 0FFE7h is FFh, program execution starts from the address shown by the vector in the relocatable vector table.	
Address Match	0FFE8h to 0FFEBh		12.4 Address Match Interrupt
Single Step <sup>(1)</sup>	0FFEC h to 0FFEFh		
Watchdog Timer Oscillation Stop Detection Voltage Monitor 2	0FFF0h to 0FFF3h		13. Watchdog Timer 10. Clock Generation Circuit 6. Voltage Detection Circuit
Address Break <sup>(1)</sup>	0FFF4h to 0FFF7h		
(Reserved)	0FFF8h to 0FFFBh		
Reset	0FFFCh to 0FFFFh		5. Resets

NOTE:

1. Do not use the single-step interrupt. For development tools only.

### 12.1.5.2 Relocatable Vector Tables

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

**Table 12.2 Relocatable Vector Tables**

Interrupt Source	Vector Address <sup>(1)</sup> Address (L) to Address (H)	Software Interrupt Number	Interrupt Control Register	Reference
BRK Instruction <sup>(3)</sup>	+0 to +3 (0000h to 0003h)	0	–	R8C/Tiny Series Software Manual
– (Reserved)		1 to 7	–	
Timer RD (Channel 0)	+32 to +35 (0020h to 0023h)	8	TRD0IC	14.3 Timer RD
Timer RD (Channel 1)	+36 to +39 (0024h to 0027h)	9	TRD1IC	
Timer RE	+40 to +43 (0028h to 002Bh)	10	TREIC	14.4 Timer RE
– (Reserved)		11 to 12	–	–
Key Input	+52 to +55 (0034h to 0037h)	13	KUPIC	12.3 Key Input Interrupt
A/D	+56 to +59 (0038h to 003Bh)	14	ADIC	18. A/D Converter
Clock Synchronous Serial I/O with Chip Select/I <sup>2</sup> C bus Interface <sup>(2)</sup>	+60 to +63 (003Ch to 003Fh)	15	SSUIC/IICIC	16.2 Clock Synchronous Serial I/O with Chip Select (SSU), 16.3 I <sup>2</sup> C Bus Interface
– (Reserved)		16	–	–
UART0 Transmit	+68 to +71 (0044h to 0047h)	17	S0TIC	15. Serial Interface
UART0 Receive	+72 to +75 (0048h to 004Bh)	18	S0RIC	
UART1 Transmit	+76 to +79 (004Ch to 004Fh)	19	S1TIC	
UART1 Receive	+80 to +83 (0050h to 0053h)	20	S1RIC	
$\overline{\text{INT}}_2$	+84 to +87 (0054h to 0057h)	21	INT2IC	12.2 $\overline{\text{INT}}_2$ Interrupt
Timer RA	+88 to +91 (0058h to 005Bh)	22	TRAIC	14.1 Timer RA
– (Reserved)		23	–	–
Timer RB	+96 to +99 (0060h to 0063h)	24	TRBIC	14.2 Timer RB
$\overline{\text{INT}}_1$	+100 to +103 (0064h to 0067h)	25	INT1IC	12.2 $\overline{\text{INT}}_1$ Interrupt
$\overline{\text{INT}}_3$	+104 to +107 (0068h to 006Bh)	26	INT3IC	
– (Reserved)		27	–	–
– (Reserved)		28	–	–
$\overline{\text{INT}}_0$	+116 to +119 (0074h to 0077h)	29	INT0IC	12.2 $\overline{\text{INT}}_0$ Interrupt
– (Reserved)		30	–	–
– (Reserved)		31	–	–
Software Interrupt <sup>(3)</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 IICSEL bit in the PMR register switches functions.
3. The I flag does not disable these interrupts.

### 12.1.6 Interrupt Control

The following describes enable/disable the maskable interrupts and set the priority order to acknowledge. The contents explained does not apply to the nonmaskable interrupts.

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

Figure 12.3 shows the Interrupt Control Register, Figure 12.4 shows Registers TRD0IC, TRD1IC, SSUIC, and IICIC and Figure 12.5 shows the Registers INT0IC to INT3IC.

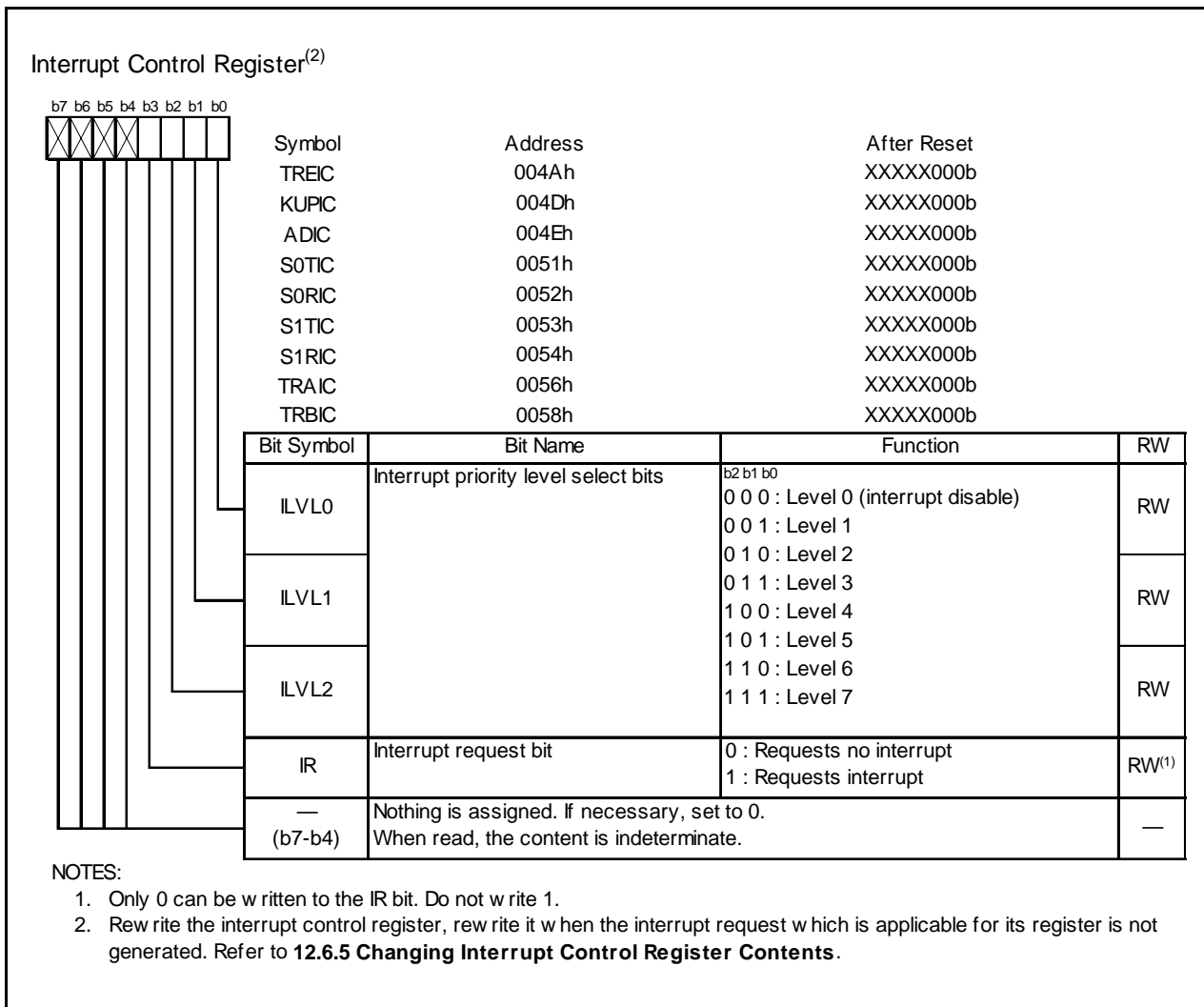


Figure 12.3 Interrupt Control Register

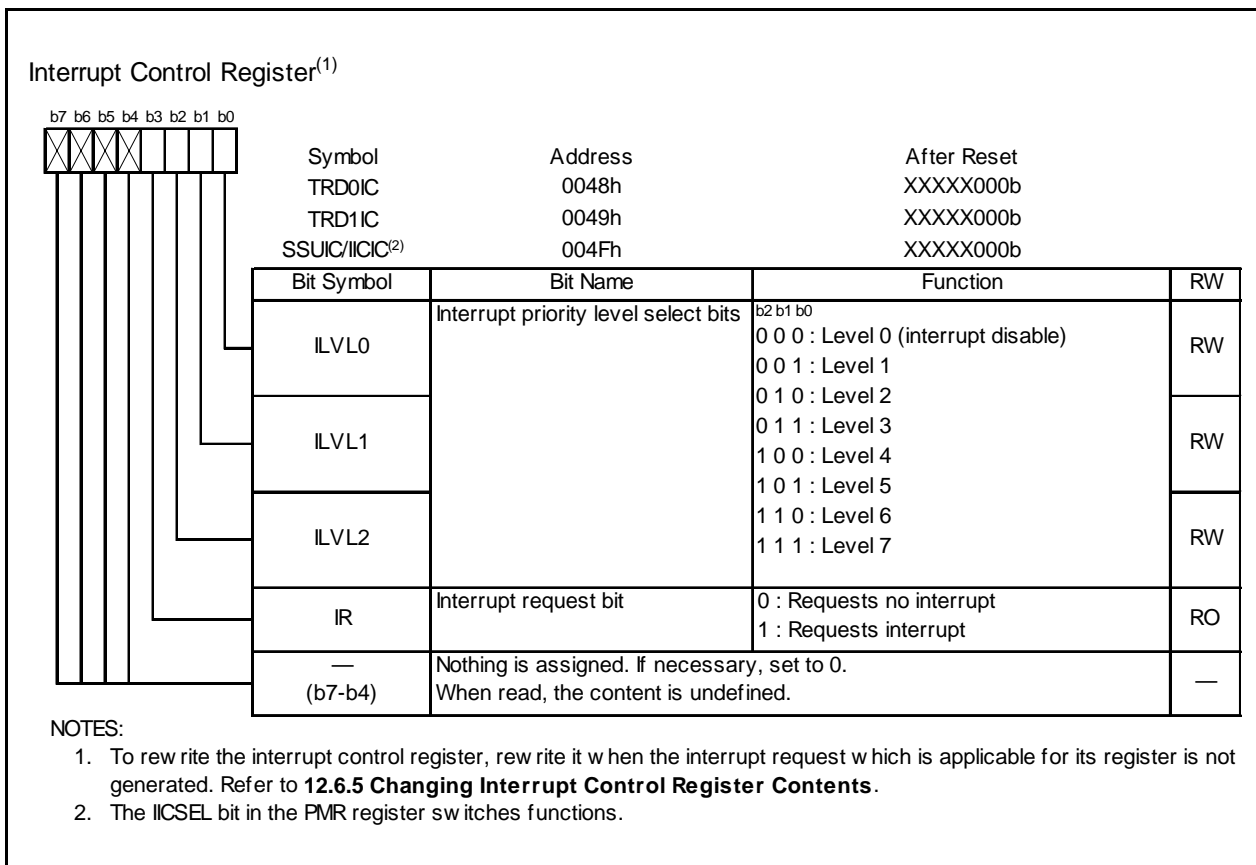


Figure 12.4 Registers TRD0IC, TRD1IC, SSUIC, and IICIC

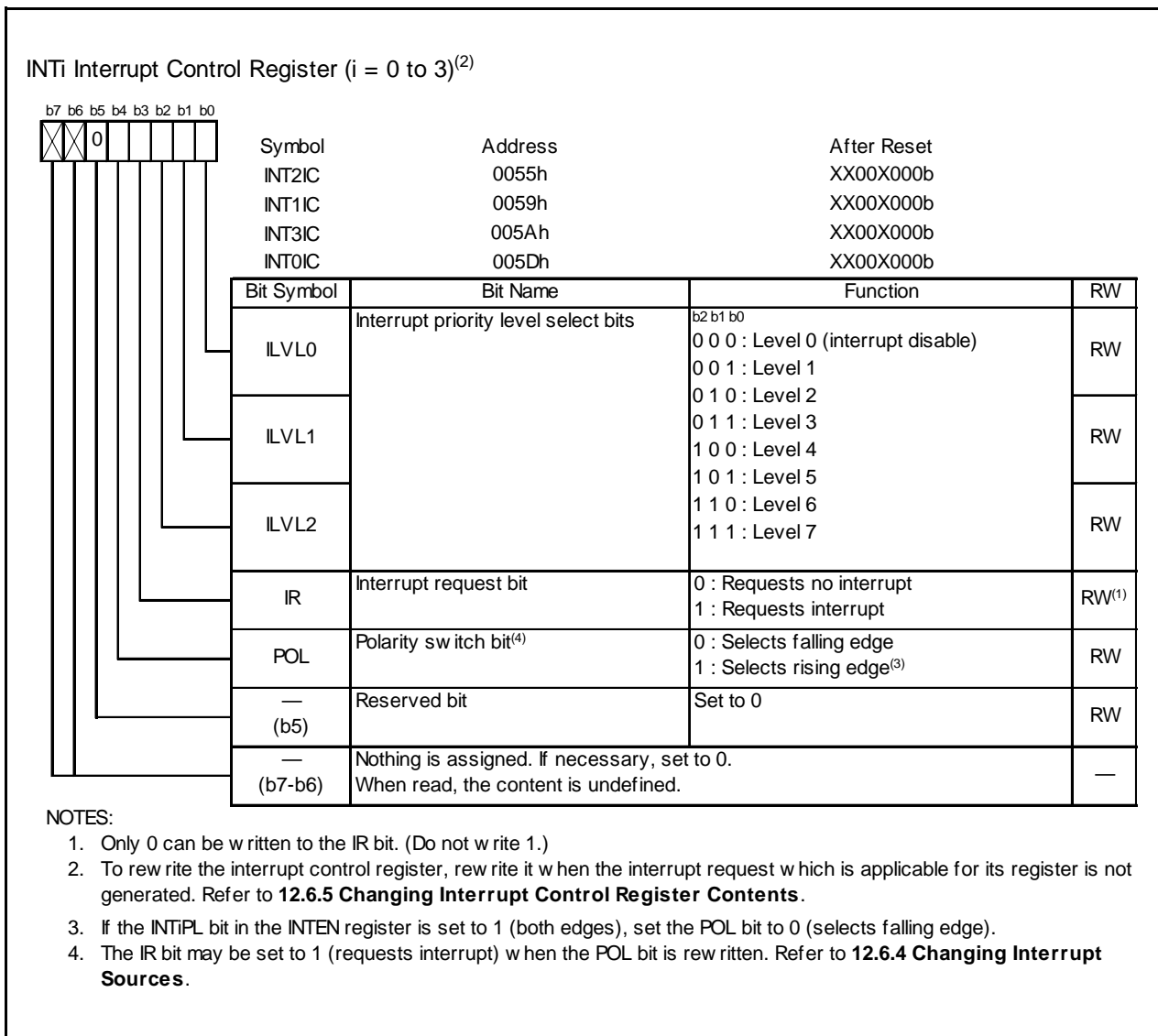


Figure 12.5 Registers INT0IC to INT3IC

### 12.1.6.1 I Flag

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

### 12.1.6.2 IR Bit

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

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

Operations of the IR bit vary by Timer RD interrupt, clock synchronous serial I/O interrupt with chip select or I<sup>2</sup>C bus interface interrupt.

For details, refer to **12.5 Timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupts and I<sup>2</sup>C bus Interface Interrupts (Interrupts with Multiple Interrupt Request Sources)**.

### 12.1.6.3 Bits ILVL2 to ILVL0 and IPL

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


Table 12.3 lists the Settings of Interrupt Priority Levels and Table 12.4 lists the Interrupt Priority Levels Enabled by IPL.

The following are conditions under which an interrupt is acknowledged:

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

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

**Table 12.3 Settings of Interrupt Priority Levels**

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

**Table 12.4 Interrupt Priority Levels Enabled by IPL**

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

### 12.1.6.4 Interrupt Sequence

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

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

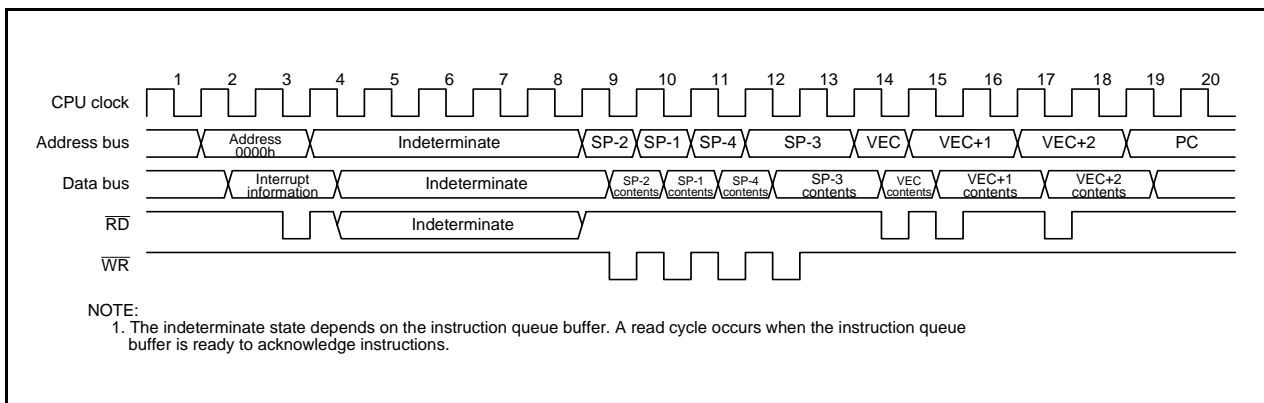
Figure 12.6 shows the Time Required for Executing Interrupt Sequence.

- (1) The CPU gets interrupt information (interrupt number and interrupt request level) by reading the address 00000h. The IR bit for the corresponding interrupt is set to 0 (interrupt not requested)<sup>(2)</sup>.
- (2) The FLG register immediately before entering the interrupt sequence is saved to the CPU internal temporary register<sup>(1)</sup>.
- (3) The I, D and U flags in the FLG register are set as follows:  
The I flag is set to 0 (disables interrupts).  
The D flag is set to 0 (disables single-step interrupt).  
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, the instructions are

NOTES:

1. This register cannot be used by user.



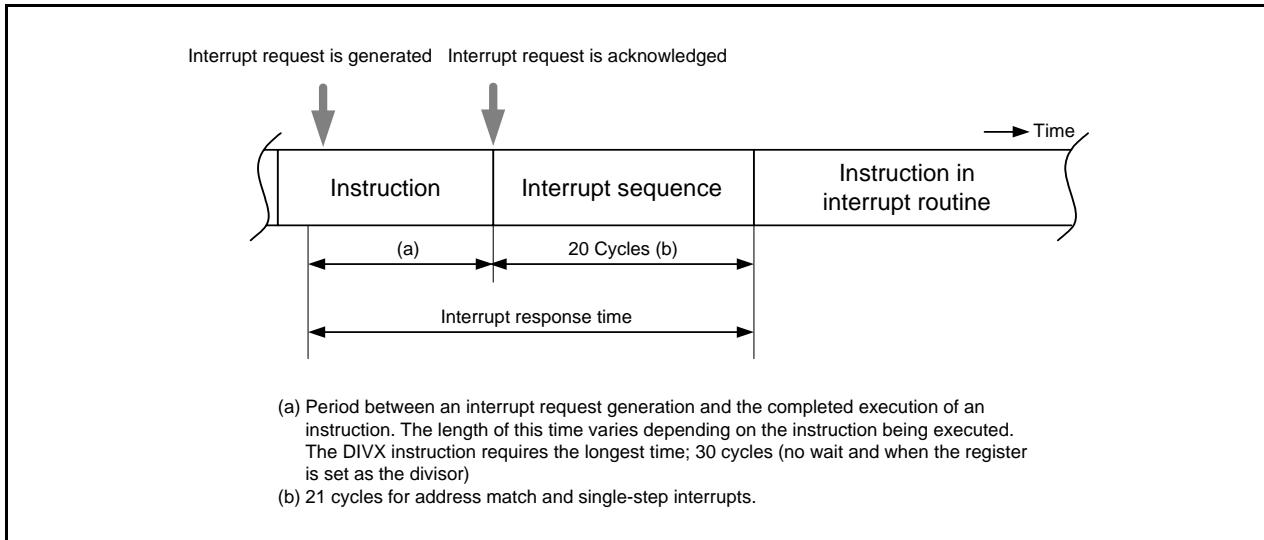
**Figure 12.6 Time Required for Executing Interrupt Sequence**

2. For operations of the IR bit, refer to **12.5 Timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupts and I<sup>2</sup>C bus Interface Interrupts (Interrupts with Multiple Interrupt Request Sources)**.



### 12.1.6.5 Interrupt Response Time

Figure 12.7 shows an Interrupt Response Time. The interrupt response time is the period between an interrupt request generation and the execution of the first instruction in an interrupt routine. An interrupt response time includes the period between an interrupt request generation and the completed execution of an instruction (refer to (a) in **Figure 12.7**) and the period required to perform an interrupt sequence (20 cycles, refer to (b) in **Figure 12.7**).



**Figure 12.7** Interrupt Response Time

### 12.1.6.6 IPL Change when Interrupt Request is Acknowledged

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

When a software interrupt and special interrupt request are acknowledged, the level listed in Table 12.5 is set to the IPL.

Table 12.5 lists the IPL Value When Software or Special Interrupt Is Acknowledged.

**Table 12.5** IPL Value When Software or Special Interrupt Is Acknowledged

Interrupt Sources	Value Set to IPL
Watchdog Timer, Oscillation Stop Detection, Voltage Monitor 2, Address Break	7
Software, Address Match, Single-Step	Not changed

### 12.1.6.7 Saving a Register

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

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

Figure 12.8 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 1 instruction.

NOTE:

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

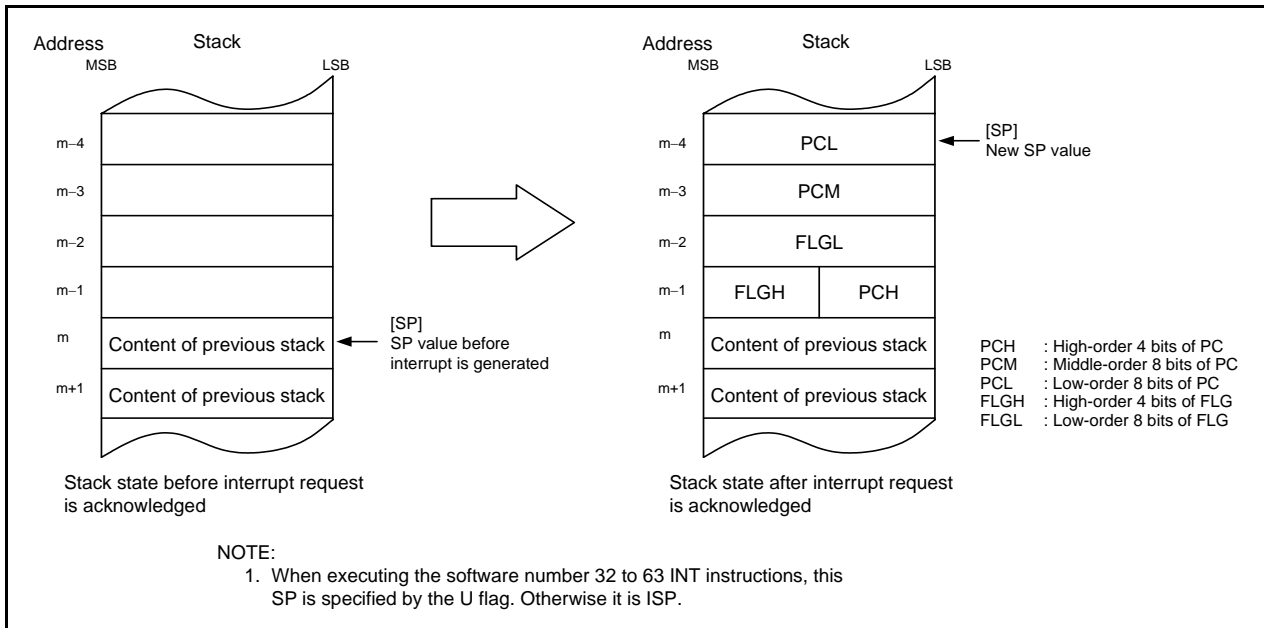


Figure 12.8 Stack State Before and After Acknowledgement of Interrupt Request

The register saving operation which is performed in the interrupt sequence is saved in 8 bits every 4 steps.

Figure 12.9 shows the Register Saving Operation.

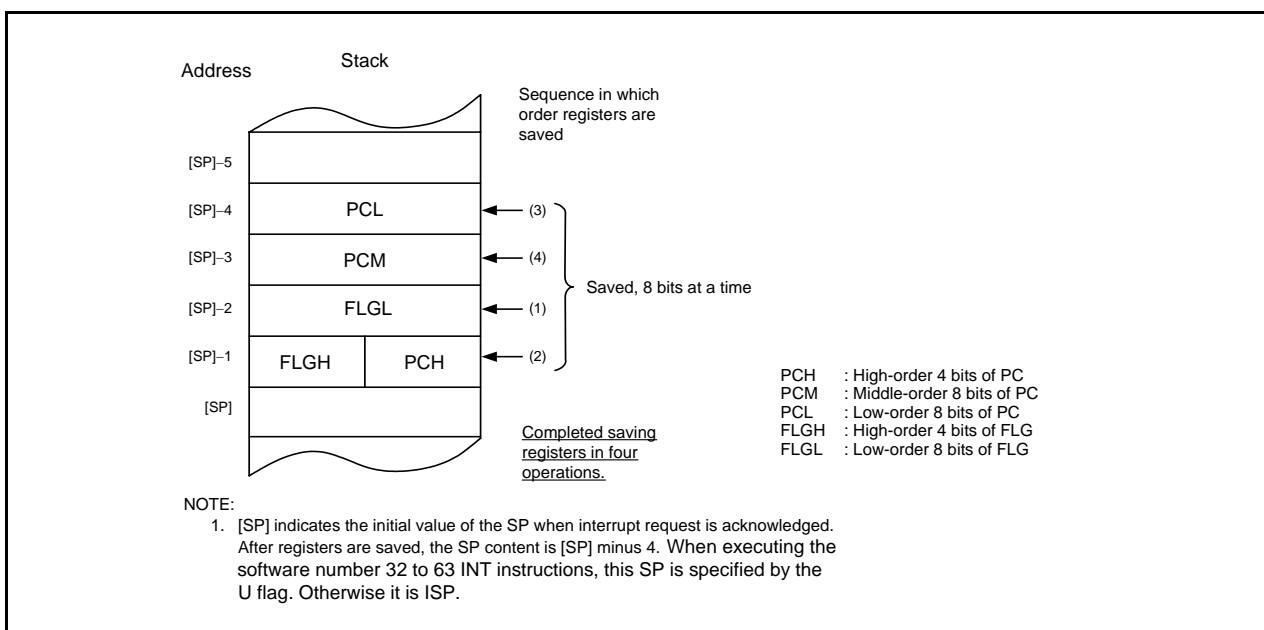


Figure 12.9 Register Saving Operation

### 12.1.6.8 Returning from an Interrupt Routine

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

Return the register saved by a program in an interrupt routine using the POPM instruction or others before the REIT instruction.

### 12.1.6.9 Interrupt Priority

If two or more interrupt requests are generated while executing one instruction, the interrupt with the higher priority is acknowledged.

Set the ILVL2 to ILVL0 bits 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, with the higher priority interrupt acknowledged in hardware.

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

Figure 12.10 shows the Priority Levels of Hardware Interrupts.

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

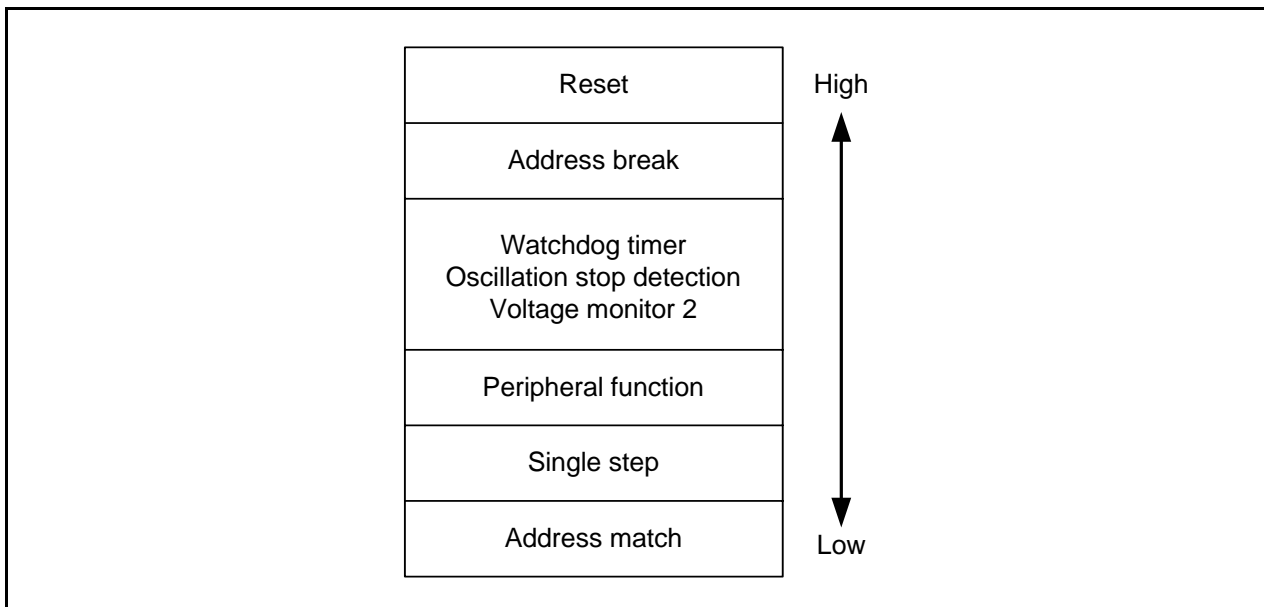


Figure 12.10 Priority Levels of Hardware Interrupts

### 12.1.6.10 Interrupt Priority Judgement Circuit

The interrupt priority judgement circuit selects the highest priority interrupt. Figure 12.11 shows the Interrupt Priority Level Judgement Circuit.

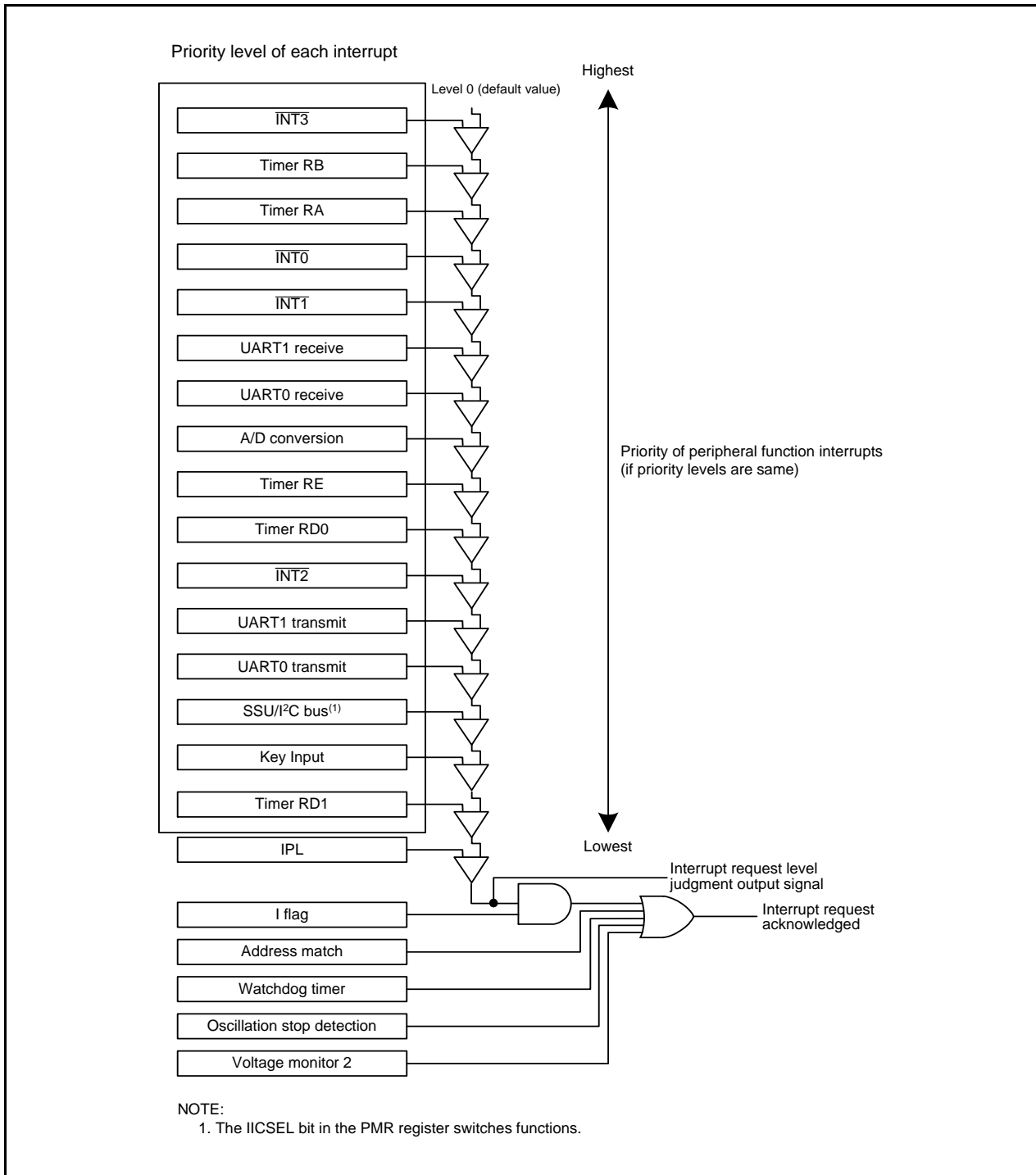


Figure 12.11 Interrupt Priority Level Judgement Circuit

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

### 12.2.1 $\overline{\text{INT}}_i$ Interrupt ( $i = 0$ to $3$ )

The  $\overline{\text{INT}}_i$  interrupt is generated by an  $\overline{\text{INT}}_i$  input. When using the  $\overline{\text{INT}}_i$  interrupt, the  $\text{INT}_i\text{EN}$  bit in the  $\text{INTEN}$  register is set to 1 (enable). The edge polarity is selected using the  $\text{INT}_i\text{PL}$  bit in the  $\text{INTEN}$  register and the  $\text{POL}$  bit in the  $\text{INTiIC}$  register.

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

The  $\overline{\text{INT}}_0$  pin is shared with the pulse output forced cutoff of timer RD and shared with the external trigger input pin of timer RB.

Figure 12.12 shows the  $\text{INTEN}$  Register. Figure 12.13 shows the  $\text{INTF}$  Register.

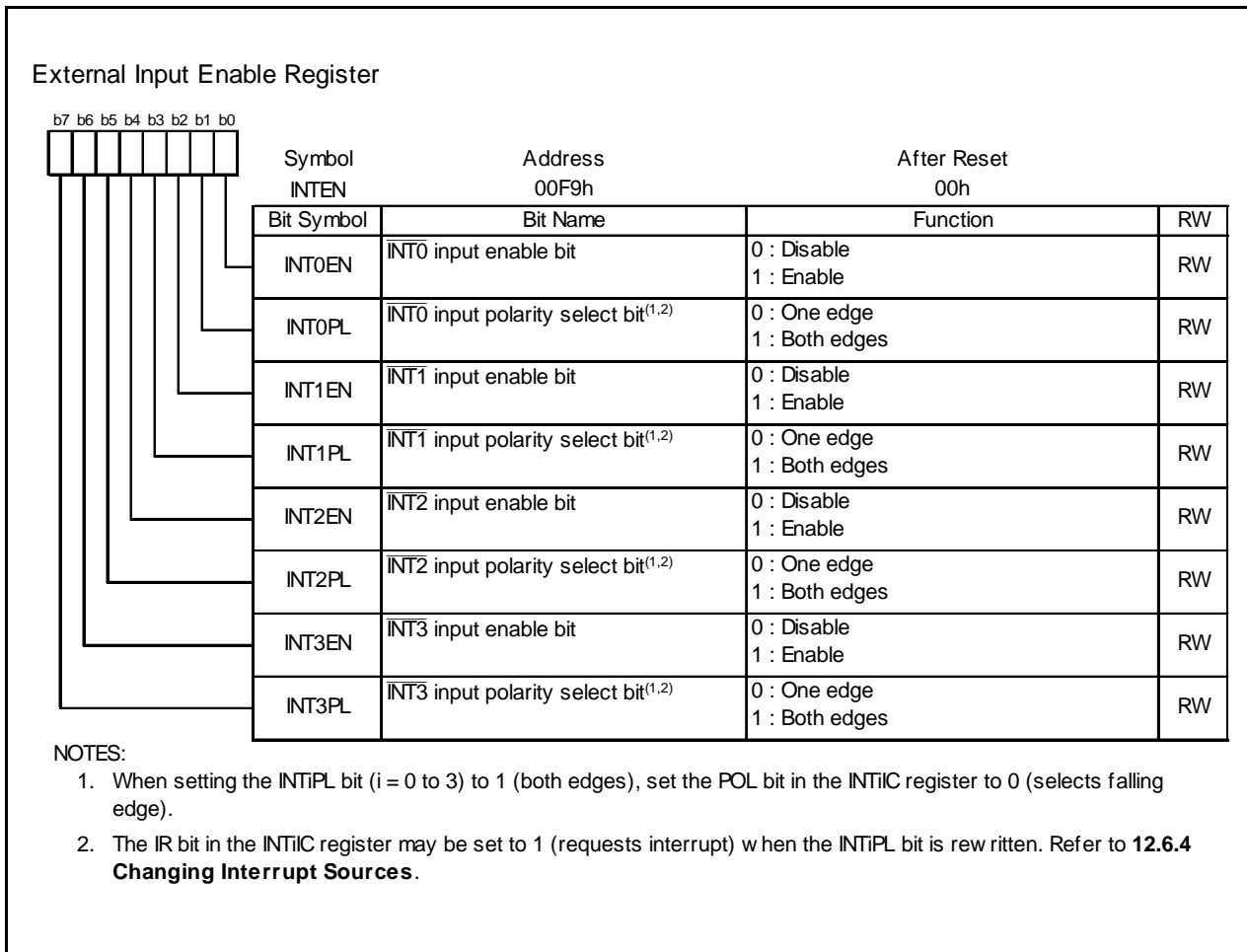


Figure 12.12  $\text{INTEN}$  Register

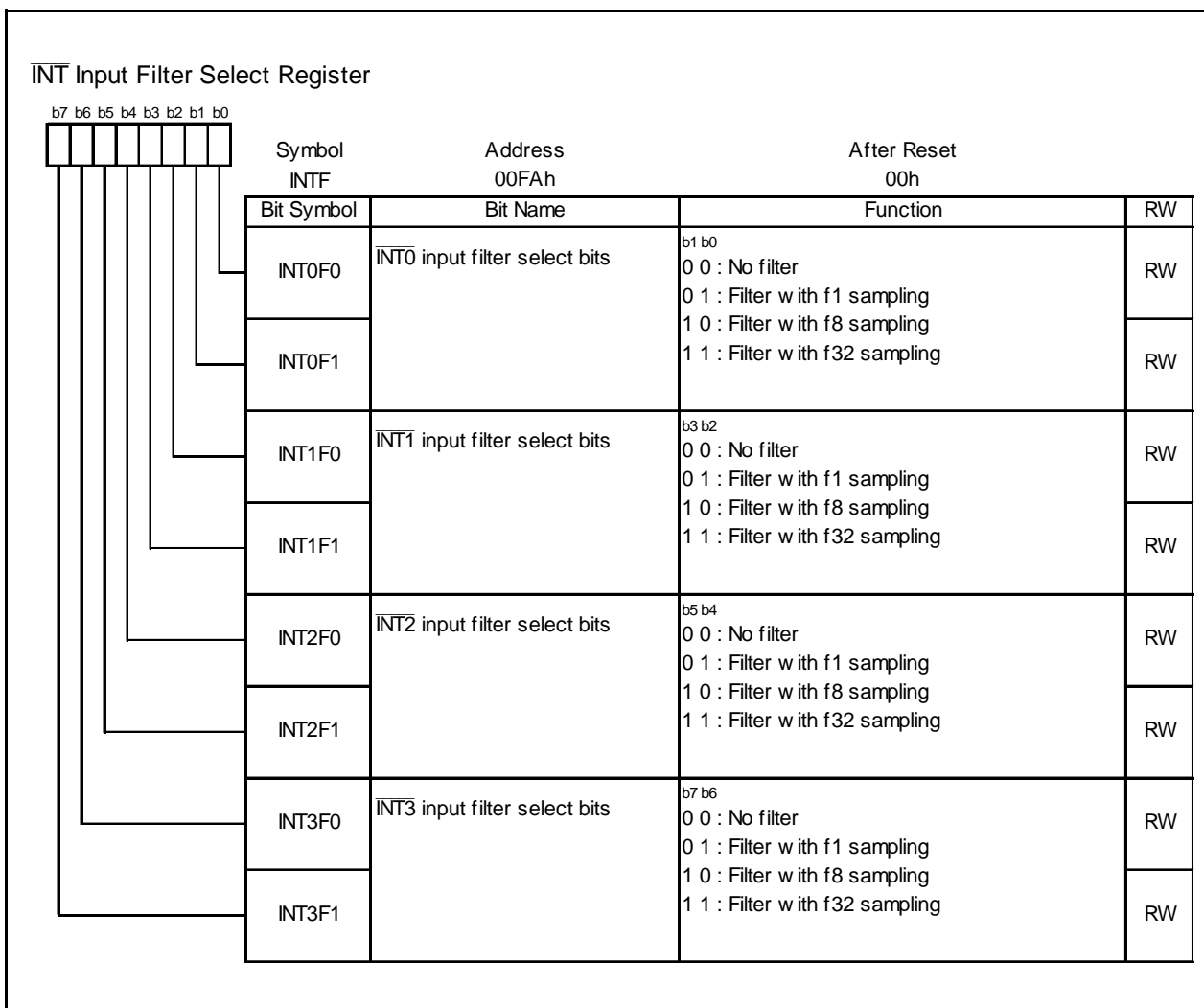


Figure 12.13 INTF Register

### 12.2.2 $\overline{\text{INT}}_i$ Input Filter (i = 0 to 3)

The  $\overline{\text{INT}}_i$  input contains a digital filter. The sampling clock is selected by the  $\text{INTiF1}$  to  $\text{INTiF0}$  bits in the  $\text{INTF}$  register. The  $\text{IR}$  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 12.14 shows the Configuration of  $\overline{\text{INT}}_i$  Input Filter. Figure 12.15 shows Operating Example of  $\overline{\text{INT}}_i$  Input Filter.

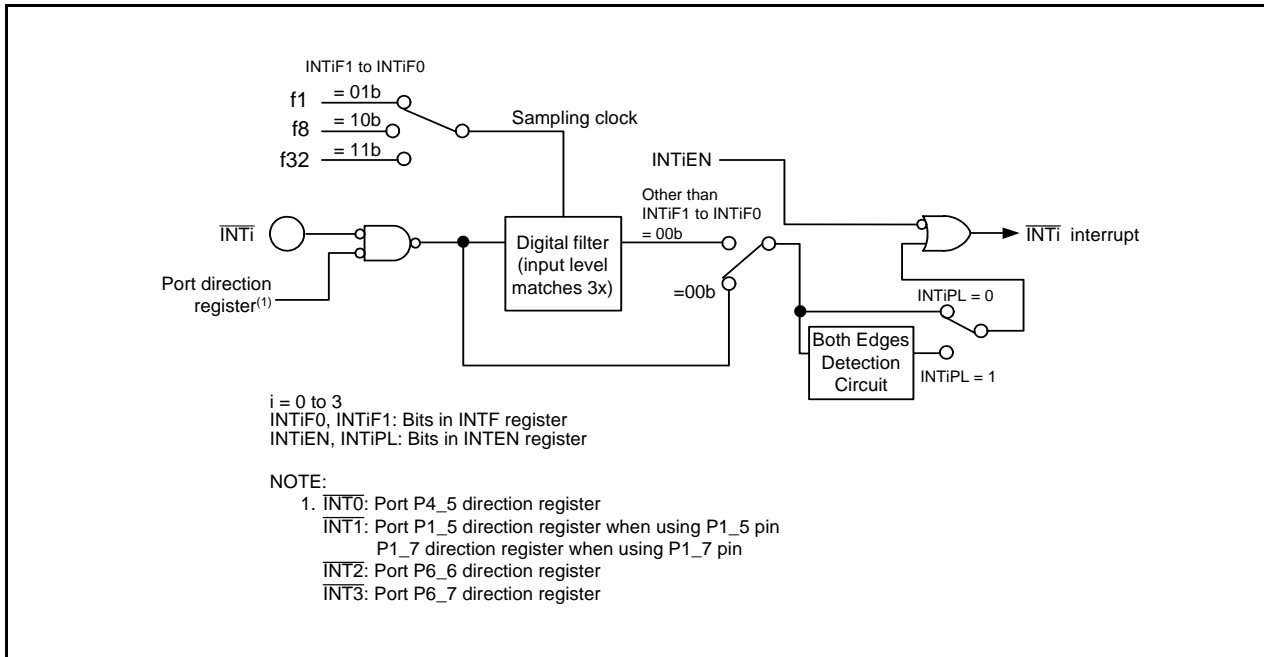


Figure 12.14 Configuration of  $\overline{\text{INT}}_i$  Input Filter

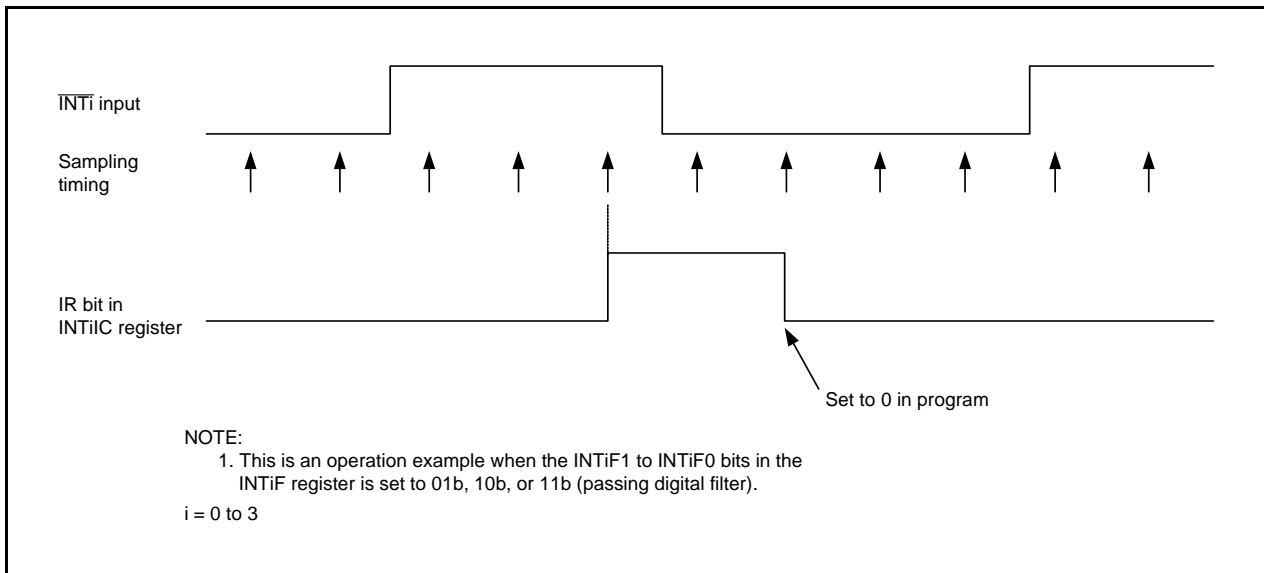


Figure 12.15 Operating Example of  $\overline{\text{INT}}_i$  Input Filter

### 12.3 Key Input Interrupt

A key input interrupt request is generated by one of the input edges of the  $\overline{K10}$  to  $\overline{K13}$  pins. 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  $\overline{K10}$  to  $\overline{K13}$  pins are 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  $\overline{K10}$  to  $\overline{K13}$  pins are not detected as interrupts.

Figure 12.16 shows a Block Diagram of Key Input Interrupt.

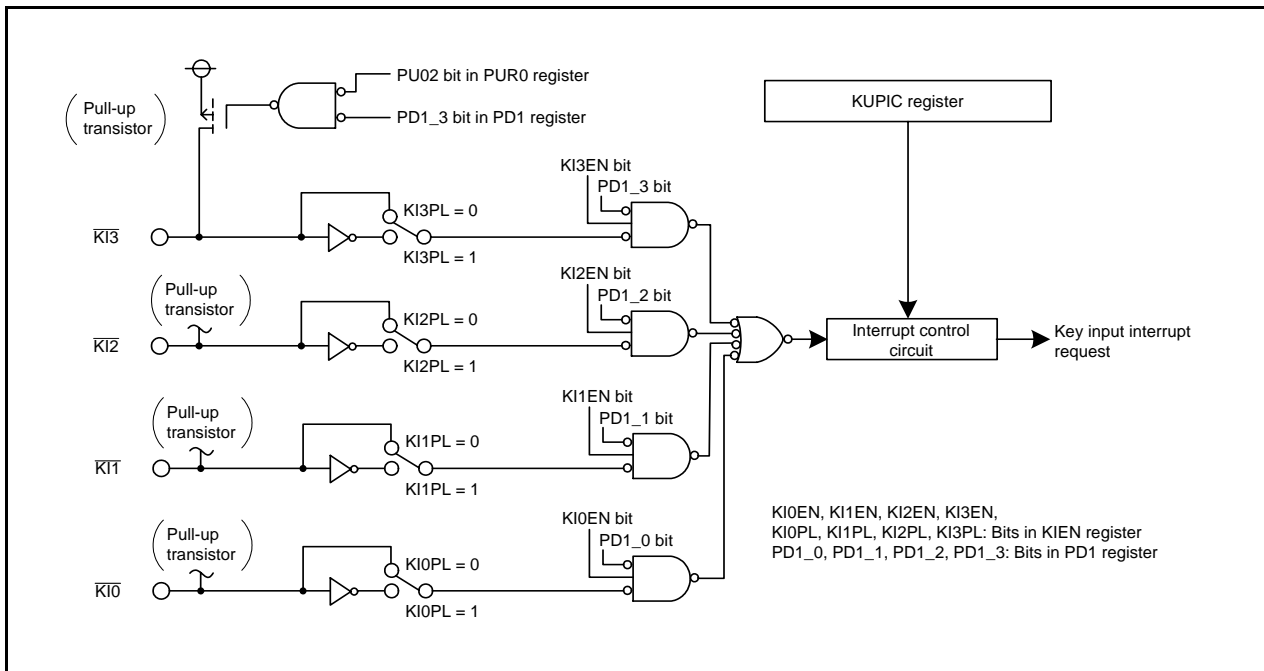


Figure 12.16 Block Diagram of Key Input Interrupt



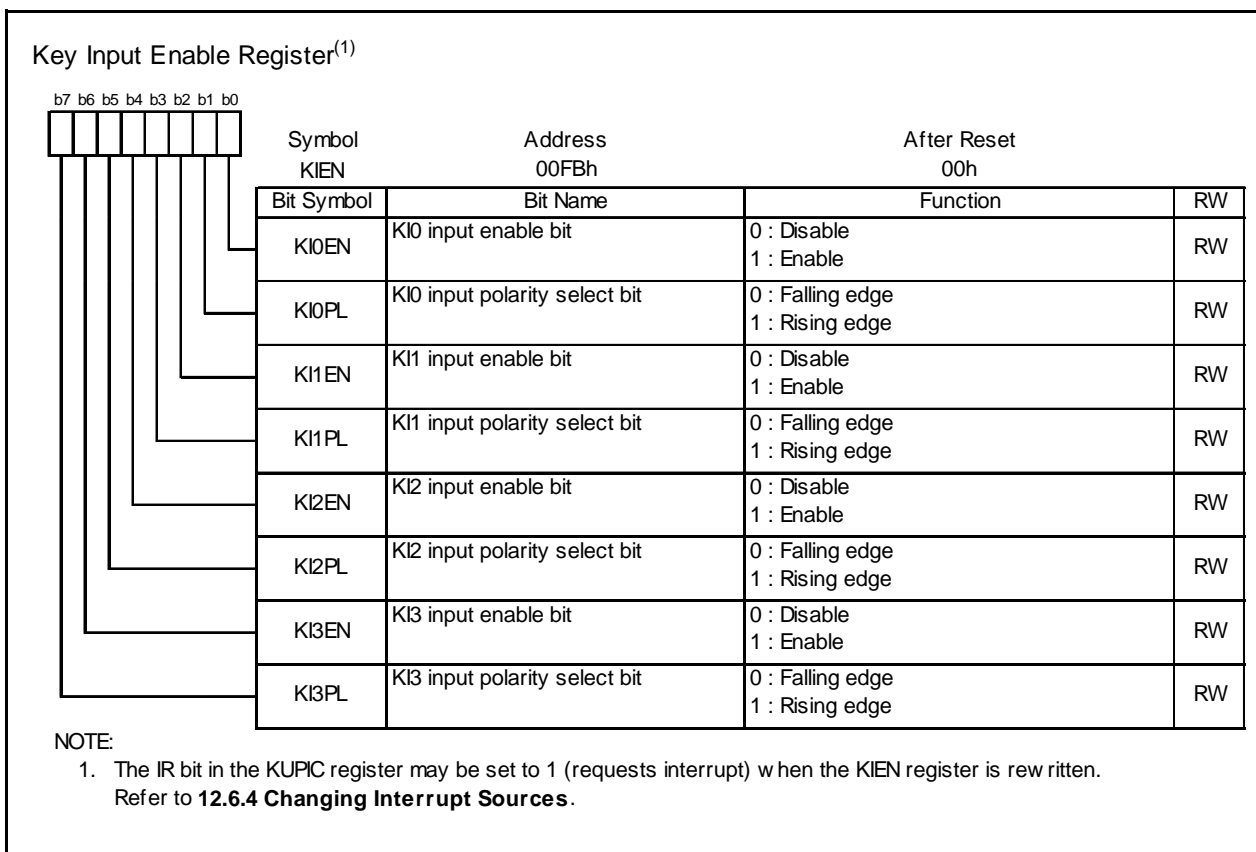


Figure 12.17 KIEN Register

## 12.4 Address Match Interrupt

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

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

- 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 an interrupt request was acknowledged. And then use a jump instruction.

Table 12.6 lists the Value of PC Saved to Stack when Address Match Interrupt is Acknowledged. Figure 12.18 shows the Registers AIER and RMAD0 to RMAD1.

**Table 12.6 Value 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 **12.1.6.7 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 12.7 Correspondence Between Address Match Interrupt Sources and Associated Registers**

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

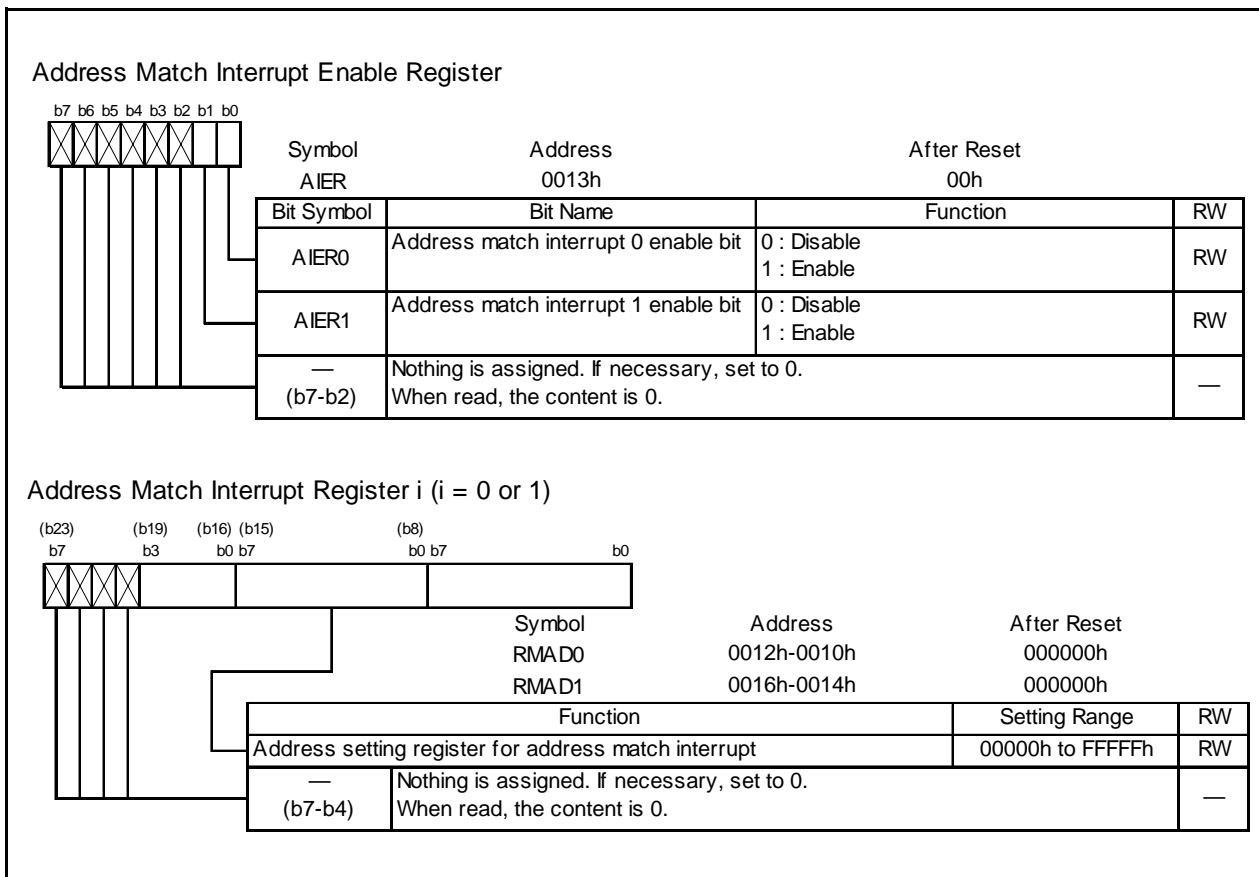


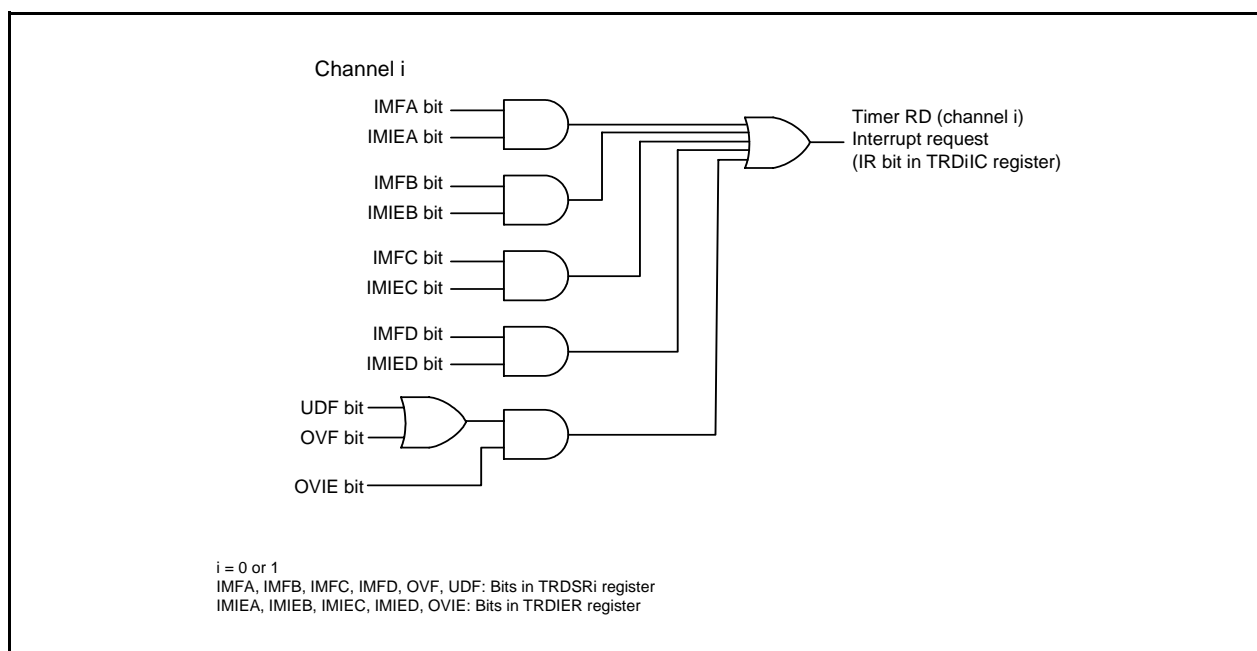
Figure 12.18 Registers AIER and RMAD0 to RMAD1

## 12.5 Timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupts and I<sup>2</sup>C bus Interface Interrupts (Interrupts with Multiple Interrupt Request Sources)

Timer RD (channel 0), timer RD (channel 1), clock synchronous serial I/O with chip select and I<sup>2</sup>C bus interface have several interrupt request sources and an interrupt request is generated by the logical OR of several interrupt request sources and the logical OR is reflected in the IR bit in the interrupt control register. Therefore, these peripheral functions which have the status register of its own interrupt request sources (status register) and the enable register of the interrupt request sources (enable register) control the generations of the interrupt request (change of the IR bit in the interrupt control register). Table 12.8 lists the Registers Associated with Timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupt, and I<sup>2</sup>C bus Interface Interrupt and Figure 12.19 shows the Block Diagram of Timer RD Interrupt.

**Table 12.8 Registers Associated with Timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupt, and I<sup>2</sup>C bus Interface Interrupt**

		Status Register of Interrupt Request Source	Enable Register of Interrupt Request Source	Interrupt Control Register
Timer RD	Channel 0	TRDSR0	TRDIER0	TRD0IC
	Channel 1	TRDSR1	TRDIER1	TRD1IC
Clock Synchronous Serial I/O with Chip Select		SSSR	SSER	SSUIC
I <sup>2</sup> C Bus Interface		ICSR	ICIER	IICIC



**Figure 12.19 Block Diagram of Timer RD Interrupt**

Controlling an interrupt with the I flag, IR bit, ILVL0 to ILVL2 bits and IPL by Timer RD (channel 0), Timer RD (channel 1), clock synchronous serial I/O with chip select and I<sup>2</sup>C bus interface is the same as that by other maskable interrupts. However, since an interrupt source is generated based on multiple interrupt request sources, there are the following differences from other maskable interrupts:

- When bits in the enable register corresponding to set bits in the status register to 1 are set to 1 (enable interrupt), the IR bit in the interrupt control register is set to 1 (interrupt requested).
- When either bits in the status register or bits in the enable register corresponding to bits in the status register, or both of them are set to 0, the IR bit is set to 0 (interrupt not requested). Basically, even though the interrupt is not acknowledged after the IR bit is set to 1, the interrupt request will not be maintained. Also, the IR bit is not set to 0 although 0 is written to the IR bit.
- Since each bit in the status register is not automatically set to 0 even if the interrupt is acknowledged. Therefore, the IR bit is not also automatically set to 0 when the interrupt is acknowledged. Set each bit in the status register to 0 in the interrupt routine. Refer to the status register figure how to set each bit in the status register to 0.
- When multiple bits in the enable register are set to 1 and other request sources are generated after the IR bit is set to 1, the IR bit remains 1.
- When multiple bits in the enable register are set to 1, determine by the status register which request source causes an interrupt.

Refer to chapters of each peripheral function (**14.3 Timer RD**, **16.2 Clock Synchronous Serial I/O with Chip Select (SSU)** and **16.3 I<sup>2</sup>C Bus Interface**) for the status register and enable register.

Refer to **12.1.6 Interrupt Control** for the interrupt control register.

## 12.6 Notes on Interrupts

### 12.6.1 Reading Address 00000h

Do not read the 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 the address 00000h is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is set to 0. This may cause a problem that the interrupt is canceled, or an unexpected interrupt is generated.

### 12.6.2 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 any value in the SP, the program may run out of control.

### 12.6.3 External Interrupt and Key Input Interrupt

Either an “L” level or an “H” level of width shown in the Electrical Characteristics is necessary for the signal input to the  $\overline{\text{INT0}}$  to  $\overline{\text{INT3}}$  pins and  $\overline{\text{KI0}}$  to  $\overline{\text{KI3}}$  pins regardless of the CPU clocks. For details, refer to **Table 20.19 External Interrupt  $\overline{\text{INTi}}$  (i = 0 to 3) Input**, **Table 20.25 External Interrupt  $\overline{\text{INTi}}$  (i = 0 to 3) Input**.

### 12.6.4 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, the changes of interrupt sources include all sources that change the interrupt sources assigned to individual software interrupt numbers, polarities, and timing. Therefore, when a mode change of the peripheral functions involves interrupt sources, edge polarities, and timing, Set the IR bit to 0 (no interrupt requested) after the change. Refer to each peripheral function for the interrupts caused by the peripheral functions.

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

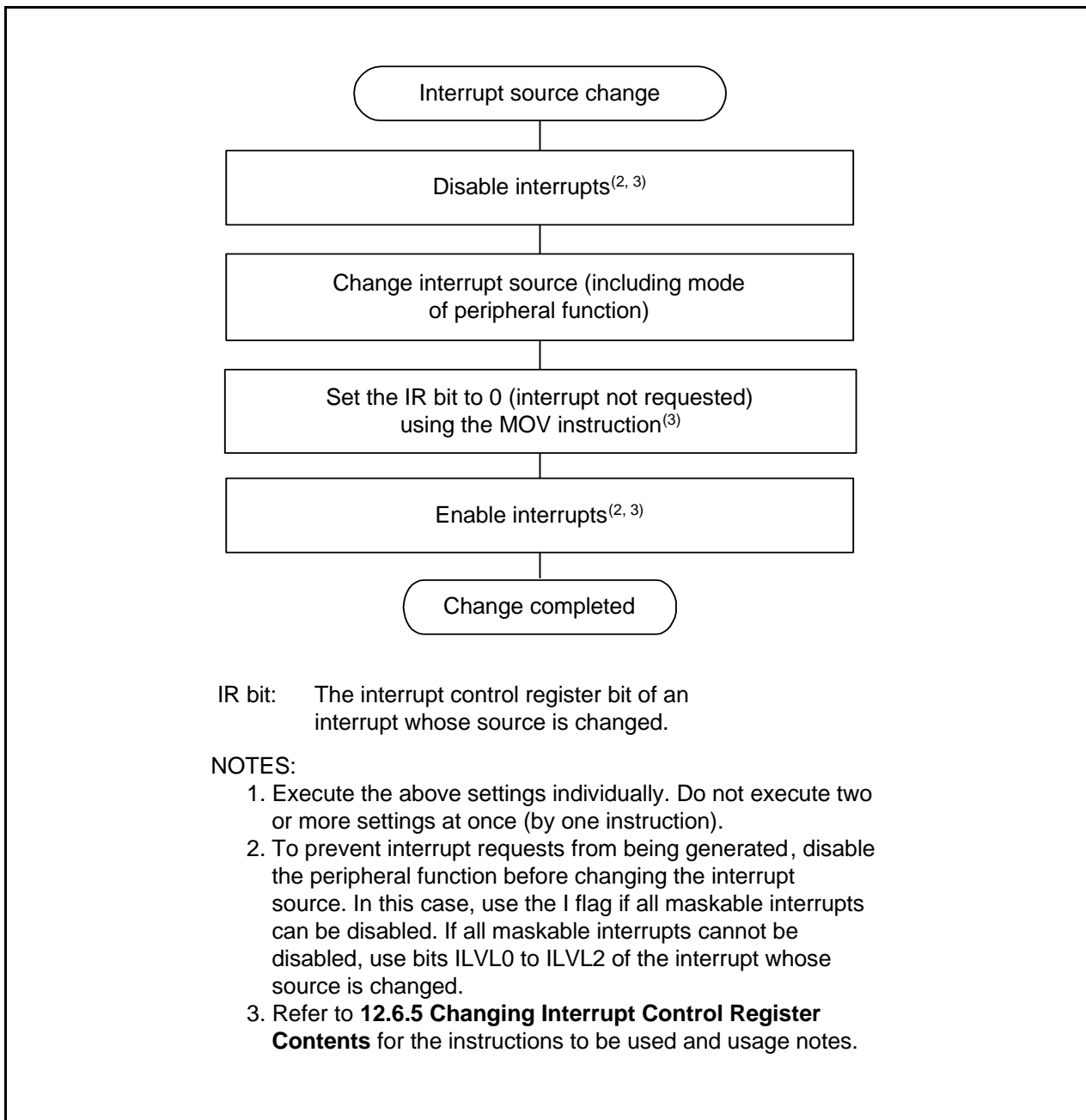


Figure 12.20 Example of Procedure for Changing Interrupt Sources

### 12.6.5 Changing Interrupt Control Register Contents

- (a) Each interrupt control register can only be changed while interrupt requests corresponding to that register are not generated. If interrupt requests may be generated, disable the interrupts before changing the interrupt control register.
- (b) When changing any interrupt control register after disabling interrupts, be careful with the instructions to be used.  
**When changing any bit other than IR bit**  
 If an interrupt request corresponding to that 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.  
 Instructions to use: AND, OR, BCLR, BSET  
**When 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 to be 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 according to the following sample programs. Refer to (b) for the change of interrupt control registers in the sample programs.

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

**Example 1: Use NOP instructions to prevent I flag 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 have FSET instruction wait**

```
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
```



## 13. Watchdog Timer

The watchdog timer is a function to detect when the program is out of control. To use the watchdog timer is recommend for improving reliability of a system. The watchdog timer contains a 15-bit counter and can select count source protection mode is enabled or disabled.

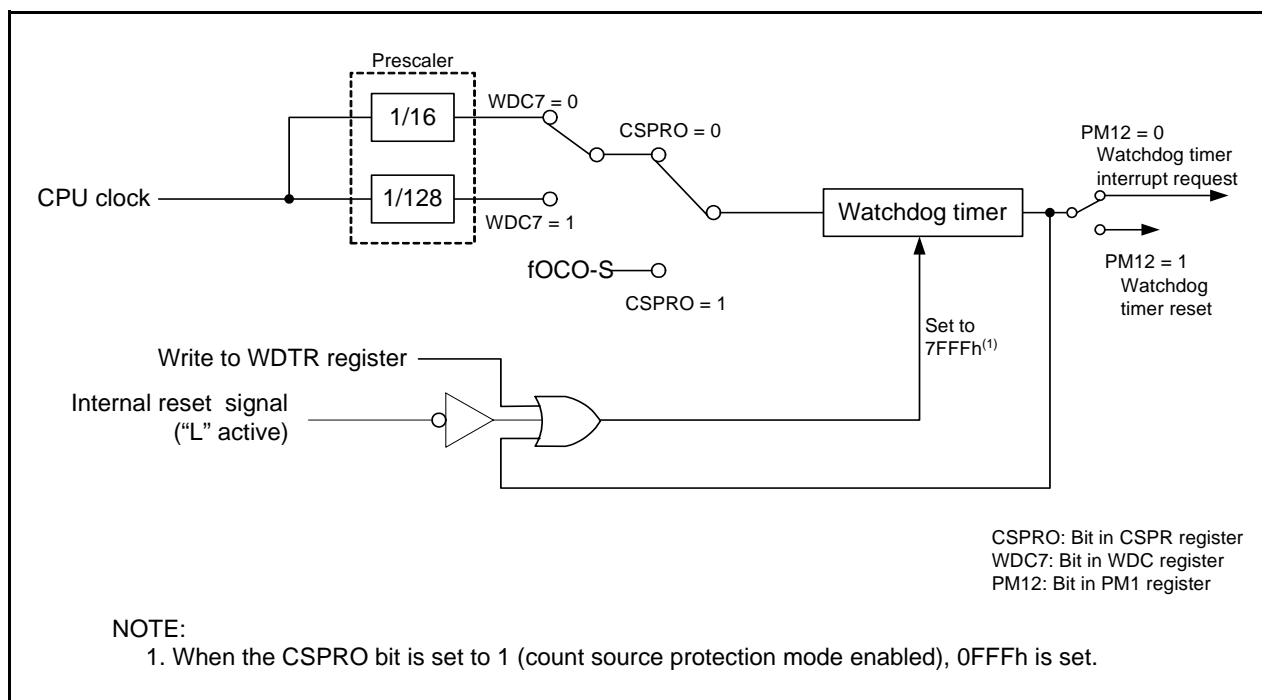
Table 13.1 lists the Count Source Protection Mode

Refer to **5.5 Watchdog Timer Reset** for details of the watchdog timer reset.

Figure 13.1 shows the Block Diagram of Watchdog Timer, Figure 13.2 shows Registers OFS and WDC and Figure 13.3 shows Registers WDTR, WDTS, and CSPR.

**Table 13.1 Count Source Protection Mode**

Item	Count Source Protection Mode Disabled	Count Source Protection Mode Enabled
Count Source	CPU clock	Low-speed on-chip oscillator clock
Count Operation	Decrement	
Count Start Condition	Either of 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	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



**Figure 13.1 Block Diagram of Watchdog Timer**

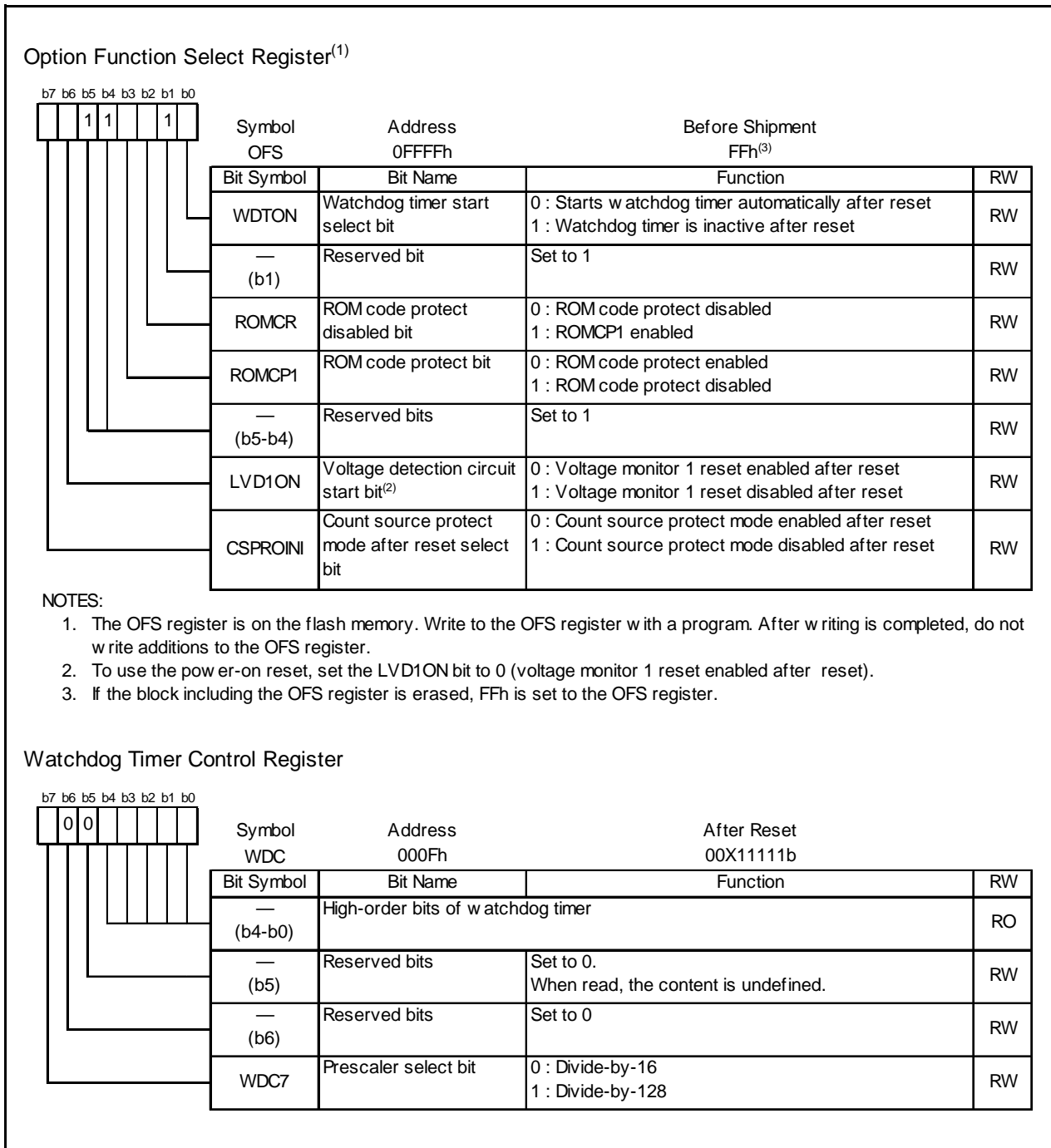


Figure 13.2 Registers OFS and WDC

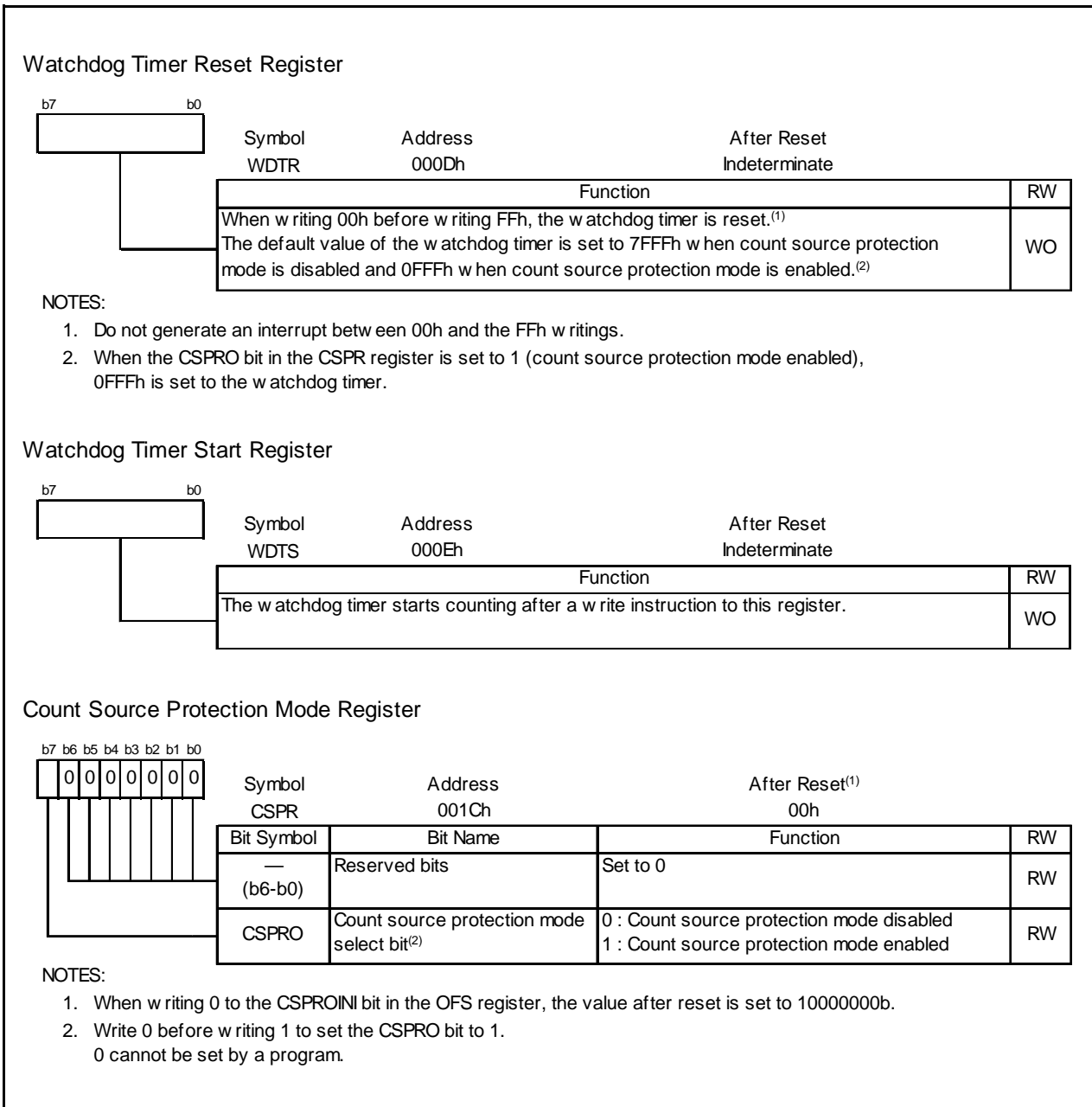


Figure 13.3 Registers WDTR, WDTS, and CSPR

### 13.1 Count Source Protection Mode Disabled

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

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

Item	Specification
Count Source	CPU clock
Count Operation	Decrement
Period	$\frac{\text{Division ratio of prescaler}(n) \times \text{count value of watchdog timer}(32768)^{(1)}}{\text{CPU clock}}$ n: 16 or 128 (selected by WDC7 bit in WDC register) e.g. When the CPU clock is 16 MHz and prescaler is divided by 16, the period is approximately 32.8 ms
Count Start Condition	The WDTON bit <sup>(2)</sup> in the OFS register (0FFFFh) selects the operation of watchdog timer after 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 reset and the count starts by writing to the WDTS register</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 reset</li> </ul>
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 Stop Condition	Stop and wait modes (inherit the count from the held value after exiting modes)
Operation at the 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>5.5 Watchdog Timer Reset</b>)</li> </ul>

**NOTES:**

1. The watchdog timer is reset when writing 00h to the WDTR register before writing FFh. The prescaler is reset after the MCU is reset. Some errors occur by the prescaler for the period of the watchdog timer.
2. The WDTON bit cannot be changed by a program. When setting the WDTON bit, write 0 to the bit 0 of the address 0FFFFh using a flash programmer.

## 13.2 Count Source Protection Mode Enabled

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

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

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

Item	Specification
Count Source	Low-speed on-chip oscillator clock
Count Operation	Decrement
Period	Count value of watchdog timer (4096) Low-speed on-chip oscillator clock e.g. Period is approximately 32.8 ms when the low-speed on-chip oscillator clock is 125 kHz
Count Start Condition	The WDTON bit <sup>(1)</sup> in the OFS register (0FFFFh) selects the operation of the watchdog timer after 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 reset and the count starts by writing to the WDTS register</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 reset</li> </ul>
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 Stop Condition	None (the count does not stop in wait mode after the count starts. The MCU does not enter stop mode)
Operation at the Time of Underflow	Watchdog timer reset (refer to <b>5.5 Watchdog Timer Reset</b> )
Register, Bit	<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 states are held in count source protection mode <ul style="list-style-type: none"> <li>- Writing to the CM10 bit in the CM1 register disables (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 disables (It remains unchanged even if it is set to 1. The low-speed on-chip oscillator does not stop)</li> </ul> </li> </ul>

### NOTES:

- The WDTON bit cannot be changed by a program. When setting the WDTON bit, write 0 to the bit 0 of the address 0FFFFh using a flash programmer.
- Even if writing 0 to the CSPROINI bit in the OFS register, the CSPRO bit is set to 1. The CSPROINI bit cannot be changed by a program. When setting the CSPROINI bit, write 0 to the bit 7 of the address 0FFFFh using a flash programmer.

## 14. Timers

The MCU contains two 8-bit timers with 8-bit prescaler, two 16-bit timers, and a timer with a 4-bit counter, and an 8-bit counter. The two 8-bit timers with the 8-bit prescaler contain timer RA and timer RB. These timers contain a reload register to memorize the default value of the counter. The 16-bit timer is timer RD which contains the input capture and output compare. The 4 and 8-bit counters are timer RE which contains the output compare. All these timers operate independently.

Table 14.1 lists Functional Comparison of Timers.

**Table 14.1 Functional Comparison of Timers**

Item		Timer RA	Timer RB	Timer RD	Timer RE
Configuration		8-bit timer with 8-bit prescaler (with reload register)	8-bit timer with 8-bit prescaler (with reload register)	16-bit free-run timer X 2 (with input capture and output compare)	4-bit counter 8-bit counter
Count		Decrement	Decrement	Increment / Decrement	Increment
Count Sources		<ul style="list-style-type: none"> <li>• f1</li> <li>• f2</li> <li>• f8</li> <li>• fOCO</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>• f1</li> <li>• f2</li> <li>• f4</li> <li>• f8</li> <li>• f32</li> <li>• fOCO40M</li> <li>• TRDIOA0</li> </ul>	<ul style="list-style-type: none"> <li>• f4</li> <li>• f8</li> <li>• f32</li> </ul>
Function	Timer mode	provided	provided	provided (input capture function, output compare function)	not provided
	Pulse output mode	provided	not provided	not provided	not provided
	Event counter mode	provided	not provided	not provided	not provided
	Pulse width measurement mode	provided	not provided	not provided	not provided
	Pulse period measurement mode	provided	not provided	not provided	not provided
	Programmable waveform generation mode	not provided	provided	not provided	not provided
	Programmable one-shot generation mode	not provided	provided	not provided	not provided
	Programmable wait one-shot generation mode	not provided	provided	not provided	not provided
	Input capture mode	not provided	not provided	provided	not provided
	Output compare mode	not provided	not provided	provided	provided
	PWM mode	not provided	not provided	provided	not provided
	Reset synchronized PWM mode	not provided	not provided	provided	not provided
	Complementary PWM mode	not provided	not provided	provided	not provided
PWM3 mode	not provided	not provided	provided	not provided	
Input Pin		TRAIO	INT0	INT0, TRDCLK, TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1	-
Output Pin		TRA0 TRAIO	TRBO	TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1	TRE0
Related Interrupt		Timer RA int INT1 int	Timer RB int INT0 int	Compare match / input capture A0 to D0 int Compare match / input capture A1 to D1 int Overflow int Underflow int <sup>(1)</sup> INT0 int	Timer RE int
Timer Stop		provided	provided	provided	provided

NOTE:

1. The underflow interrupt can be set to channel 1.

## 14.1 Timer RA

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

The prescaler and timer consist of the reload register and counter. The reload register and counter are allocated at the same address. When accessing the TRAPRE and TRA registers, the reload register and counter can be accessed (refer to **Table 14.2 to 14.6 the Specification of Each Modes**).

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

Figure 14.1 shows the Block Diagram of Timer RA. Figures 14.2 to 14.4 show the registers associated with Timer RA.

Timer RA contains five operation modes listed as follows:

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

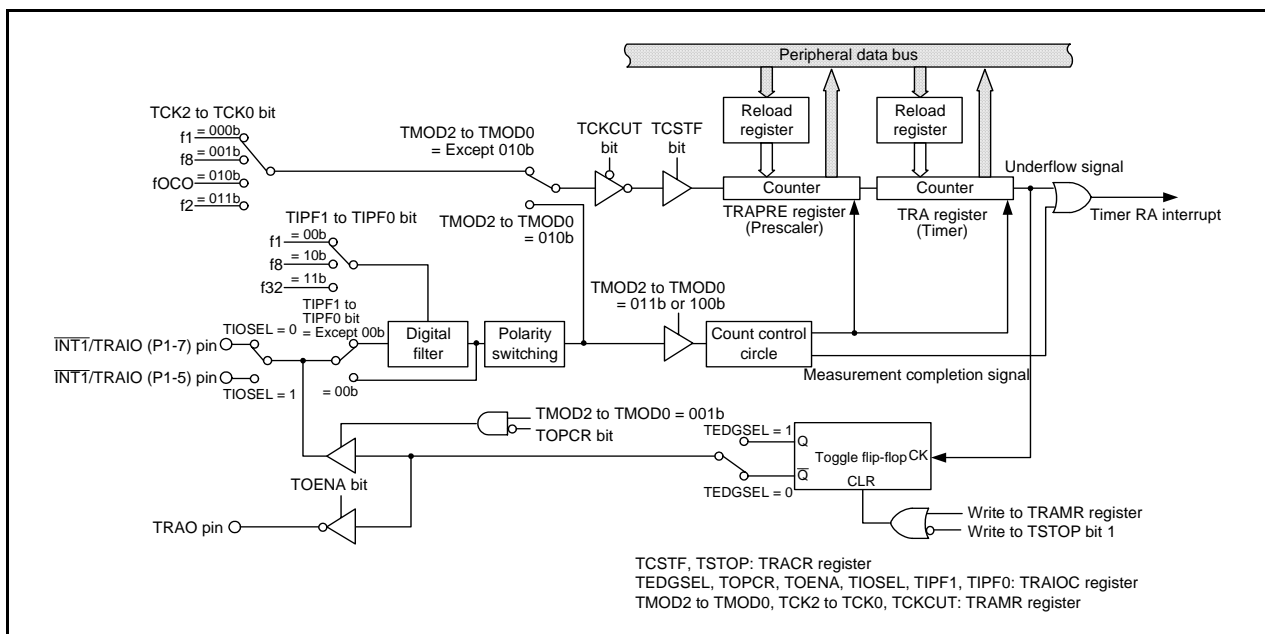


Figure 14.1 Block Diagram of Timer RA



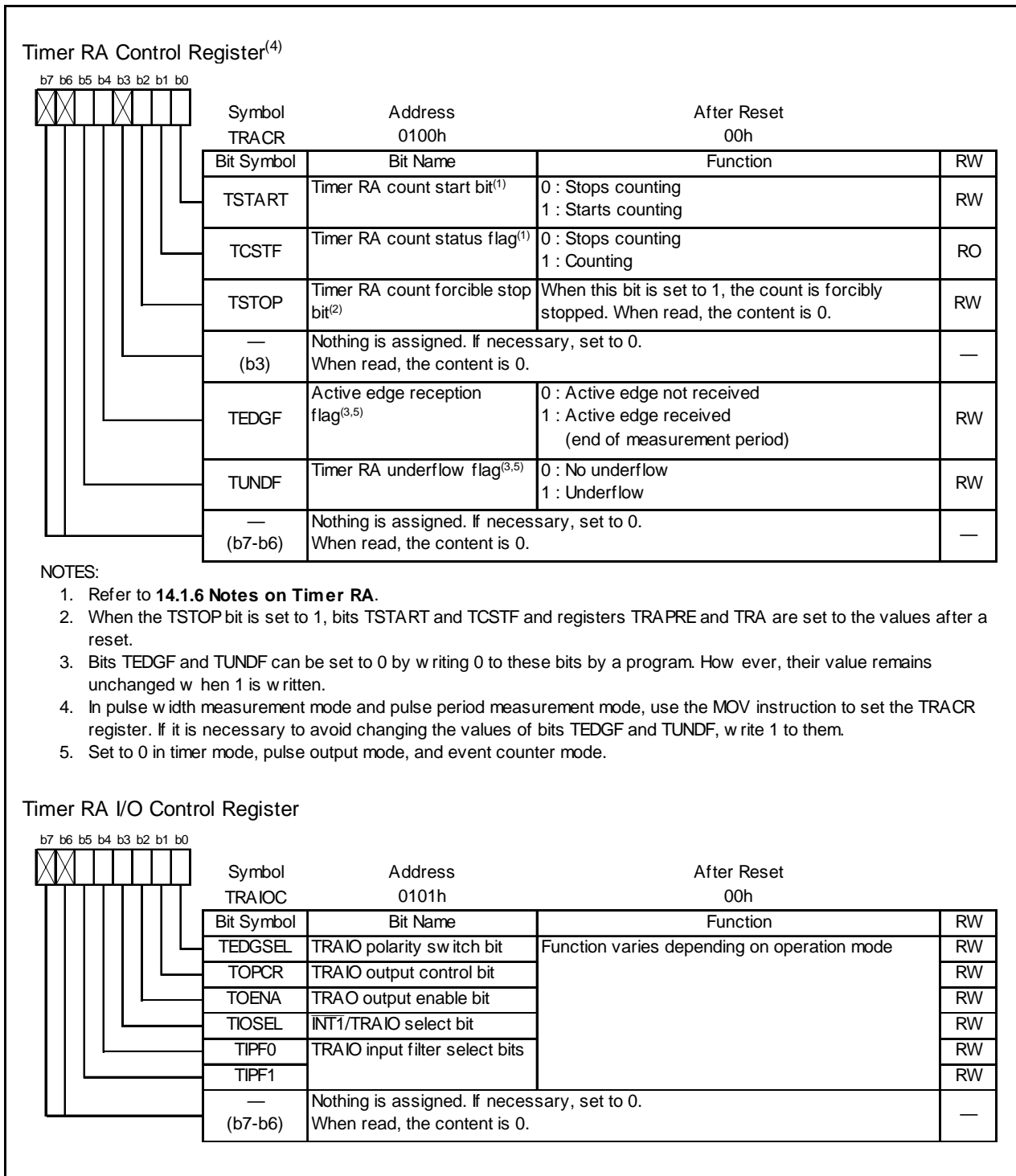
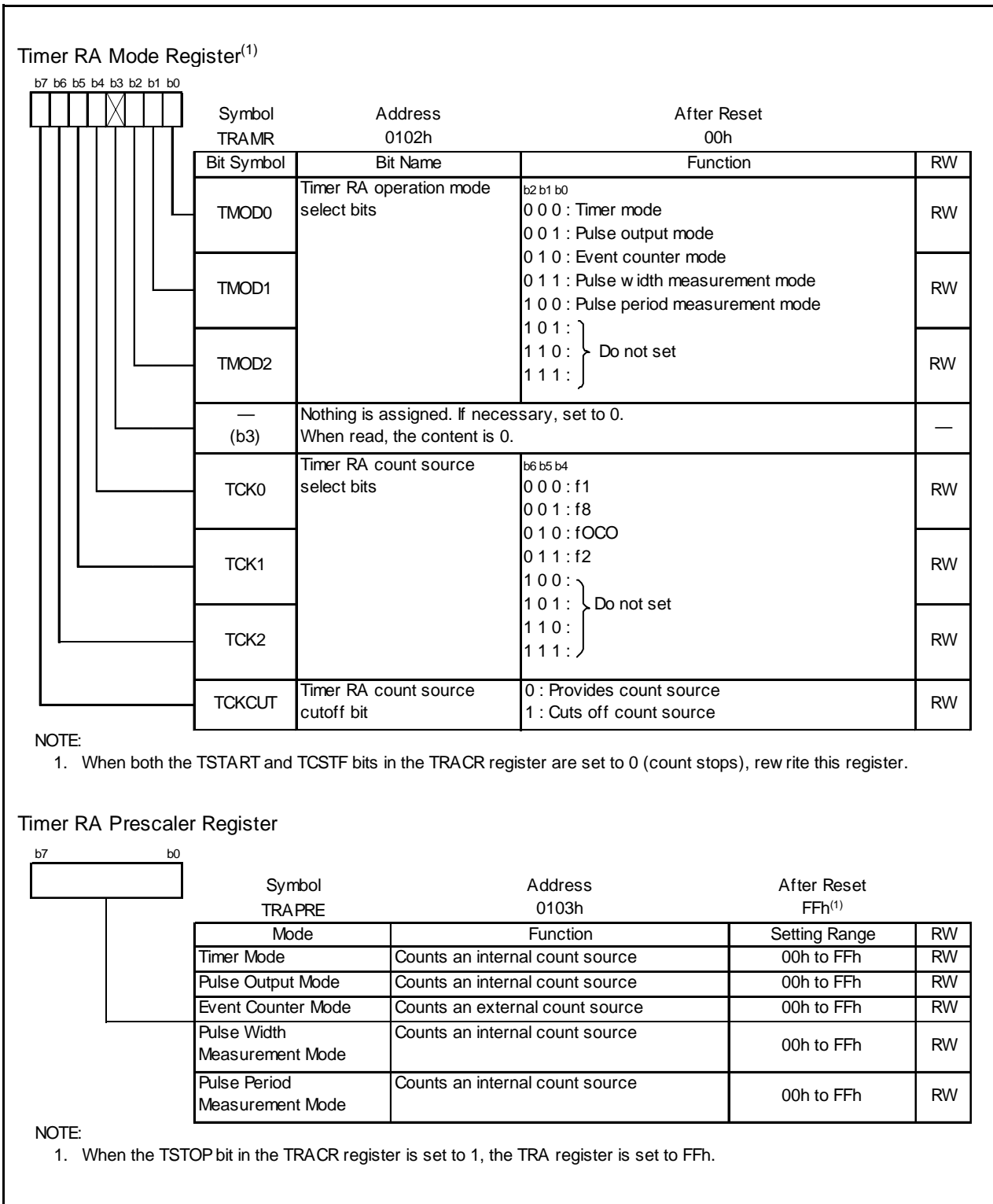


Figure 14.2 Registers TRACR and TRAIIOC



**Figure 14.3 Registers TRAMR and TRAPRE**

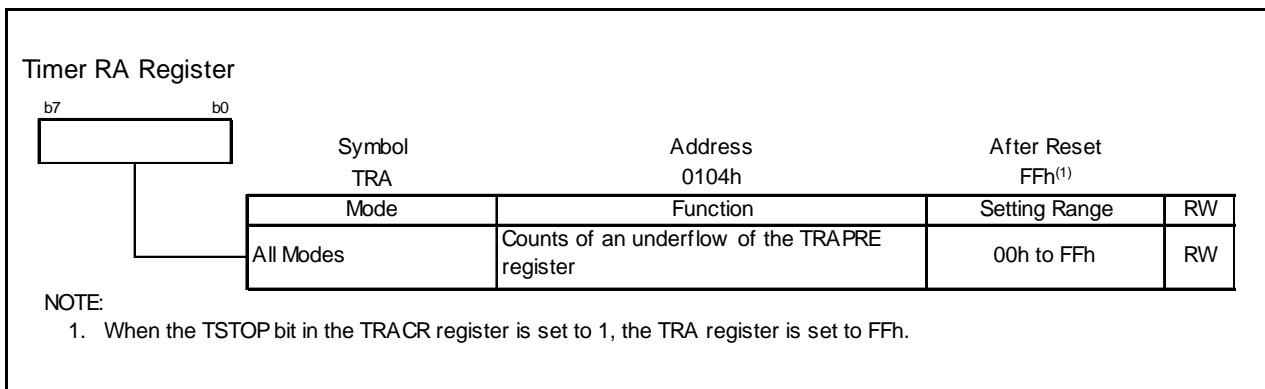


Figure 14.4 TRA Register

14.1.1 Timer Mode

In this mode, the timer counts an internally generated count source (see Table 14.2 Timer Mode Specifications).

Figure 14.5 shows the TRAIIOC Register in Timer Mode.

Table 14.2 Timer Mode Specifications

Item	Specification
Count Sources	f1, f2, f8, fOCO
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 inherited</li> </ul>
Divide Ratio	$1/(n+1)(m+1)$ n: setting value of TRAPRE register, m: setting value of TRA register
Count Start Condition	Write 1 (count starts) to the TSTART bit in the TRACR register
Count Stop Conditions	<ul style="list-style-type: none"> <li>Write 0 (count stops) to the TSTART bit in the TRACR register</li> <li>Write 1 (count forcibly stops) to the TSTOP bit in the TRACR register</li> </ul>
Interrupt Request Generation Timing	When Timer RA underflows [Timer RA interrupt]
INT1/TRAI/O Pin Function	Programmable I/O port or $\overline{\text{INT1}}$ interrupt input
TRAO Pin Function	Programmable I/O port
Read from Timer	The count value can be read by reading the TRA and TRAPRE registers
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 14.1.1.1 Timer Write Control during Count Operation).</li> </ul>

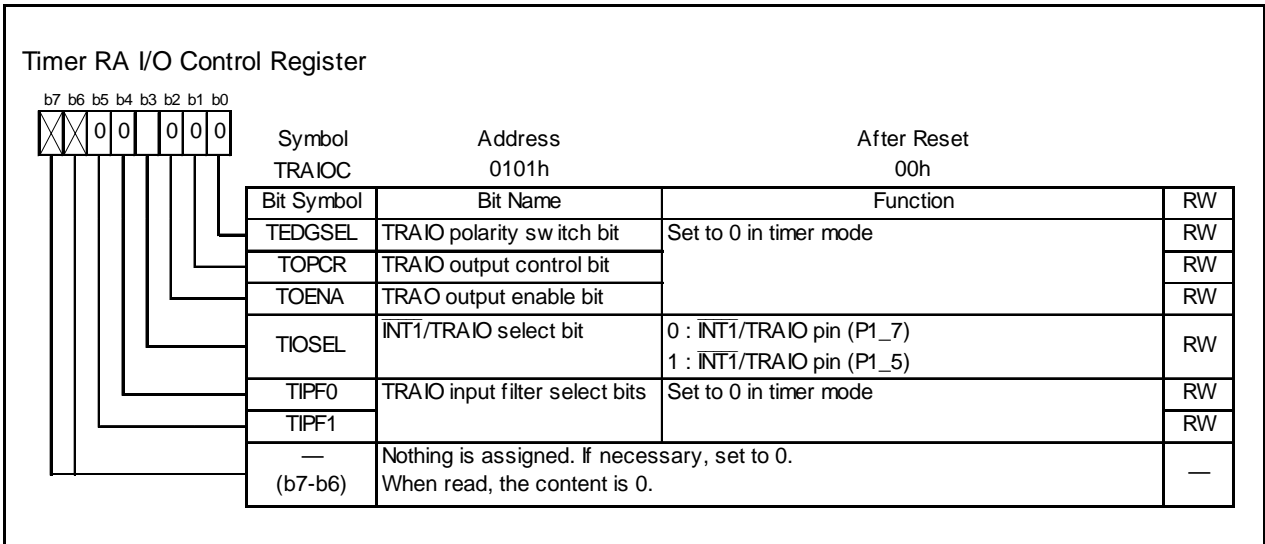
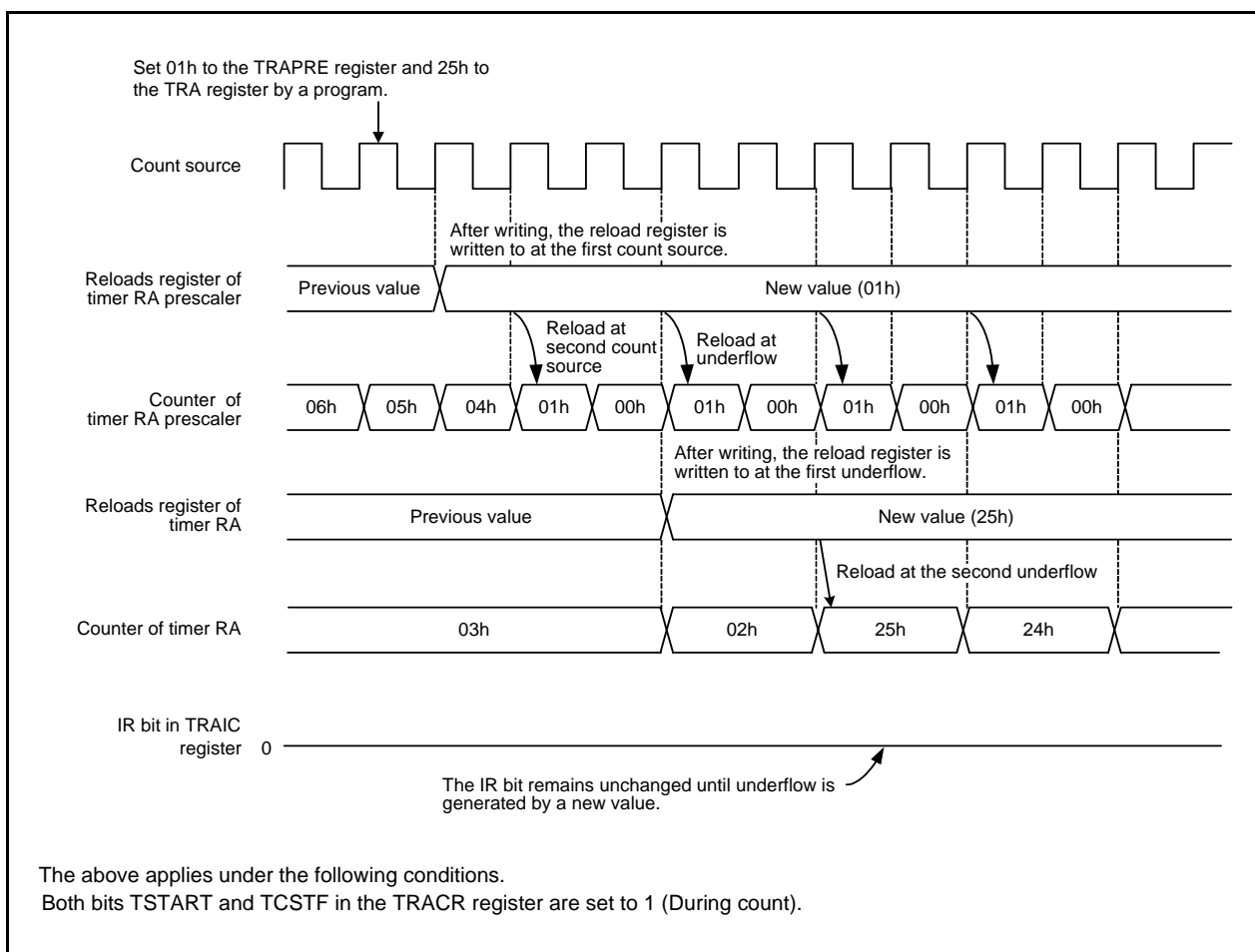


Figure 14.5 TRAIIOC Register in Timer Mode

### 14.1.1.1 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 14.6 shows an Operating Example of Timer RA when Counter Value is Rewritten during Count Operation.



**Figure 14.6** Operating Example of Timer RA when Counter Value is Rewritten during Count Operation

### 14.1.2 Pulse Output Mode

Pulse output mode is mode to count the count source internally generated and outputs the pulse which inverts the polarity from the TRAI0 pin each time the timer underflows (see **Table 14.3 Pulse Output Mode Specifications**).

Figure 14.7 shows the TRAI0C Register in Pulse Output Mode.

**Table 14.3 Pulse Output Mode Specifications**

Item	Specification
Count Sources	f1, f2, f8, fOCO
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 inherited</li> </ul>
Divide Ratio	$1/(n+1)(m+1)$ n: setting value of TRAPRE register, m: setting value of TRA register
Count Start Condition	Write 1 (count starts) to the TSTART bit in the TRACR register
Count Stop Conditions	<ul style="list-style-type: none"> <li>• Write 0 (count stops) to the TSTART bit in the TRACR register</li> <li>• Write 1 (count forcibly stops) 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>
TRAI0 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 the TRA and TRAPRE registers
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>14.1.1.1 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 can select the polarity level when the pulse output starts<sup>(1)</sup></li> <li>• Inverted pulse output function The pulse which inverts the polarity of the TRAI0 output can be output from the TRAI0 pin (selected by the TOENA bit in the TRAI0C register)</li> <li>• Pulse output stop function The pulse output from the TRAI0 pin can be stopped by the TOPCR bit.</li> <li>• INT1/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 output pulse turn into the level when the pulse output starts by writing the TRAMR register.

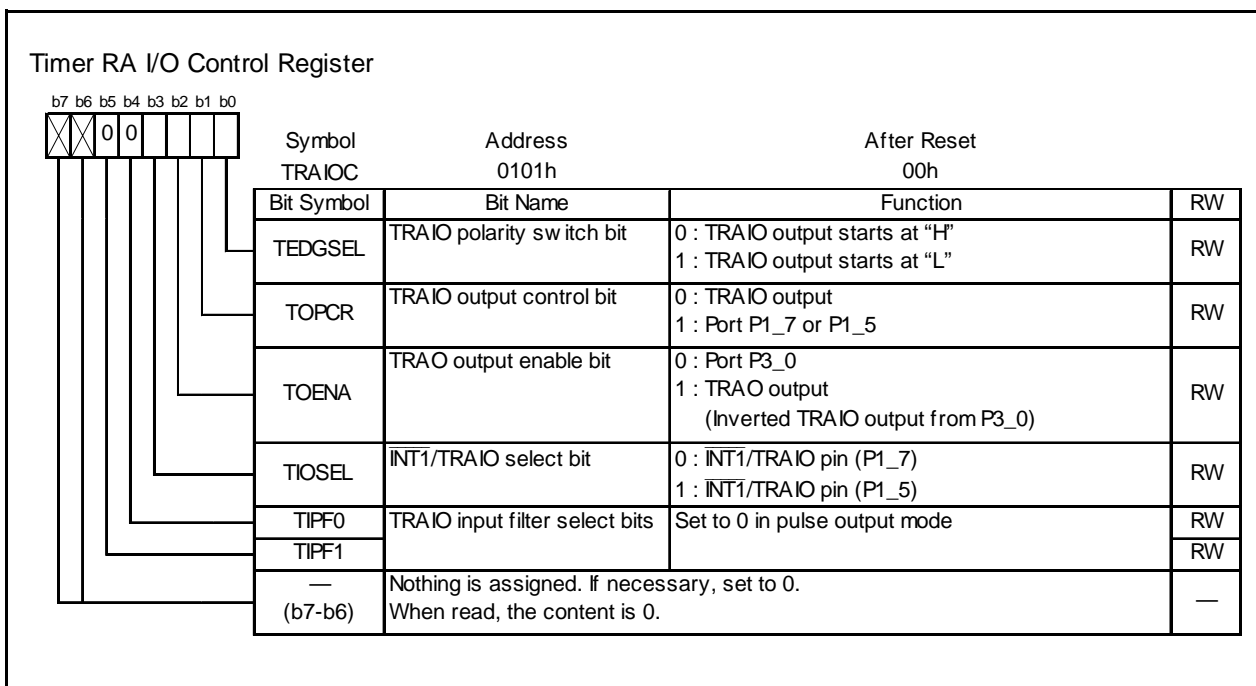


Figure 14.7 TRAI/O Register in Pulse Output Mode

### 14.1.3 Event Counter Mode

Event counter mode is mode to count an external signal which inputs from the  $\overline{\text{INT1}}$ /TRAIO pin (see **Table 14.4 Event Counter Mode Specifications**).

Figure 14.8 shows the TRAIIOC Register in Event Counter Mode.

**Table 14.4 Event Counter Mode Specifications**

Item	Specification
Count Source	External signal which is input to TRAIO pin (active edge is selectable by a program)
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 inherited</li> </ul>
Divide Ratio	$1/(n+1)(m+1)$ n: setting value of TRAPRE register, m: setting value of TRA register
Count Start Condition	Write 1 (count starts) to the TSTART bit in the TRACR register
Count Stop Conditions	<ul style="list-style-type: none"> <li>• Write 0 (count stops) to the TSTART bit in the TRACR register</li> <li>• Write 1 (count forcibly stops) to the TSTOP bit in the TRACR register</li> </ul>
Interrupt Request Generation Timing	When timer RA underflows [timer RA interrupt]
$\overline{\text{INT1}}$ /TRAIO Pin Function	Count source input ( $\overline{\text{INT1}}$ interrupt input)
TRAO Pin Function	Programmable I/O port <sup>(1)</sup>
Read from Timer	The count value can be read by reading the TRA and TRAPRE registers
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>14.1.1.1 Timer Write Control during Count Operation</b>).</li> </ul>
Select Functions	<ul style="list-style-type: none"> <li>• <math>\overline{\text{INT1}}</math> input polarity switch function The TEDGSEL bit in the TRAIIOC register can select 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 TRAIIOC register.</li> <li>• Pulse output function The pulse which inverts the polarity can be output from the TRAO pin each time the timer underflows. (selected by the TOENA bit in the TRAIIOC register)<sup>(1)</sup></li> <li>• Digital filter function Bits TIPF0 and TIPF1 in the TRAIIOC register enable or disable the digital filter and select the sampling frequency.</li> </ul>

**NOTE:**

1. The level of output pulse turn into the level when the pulse output starts by writing the TRAMR register.



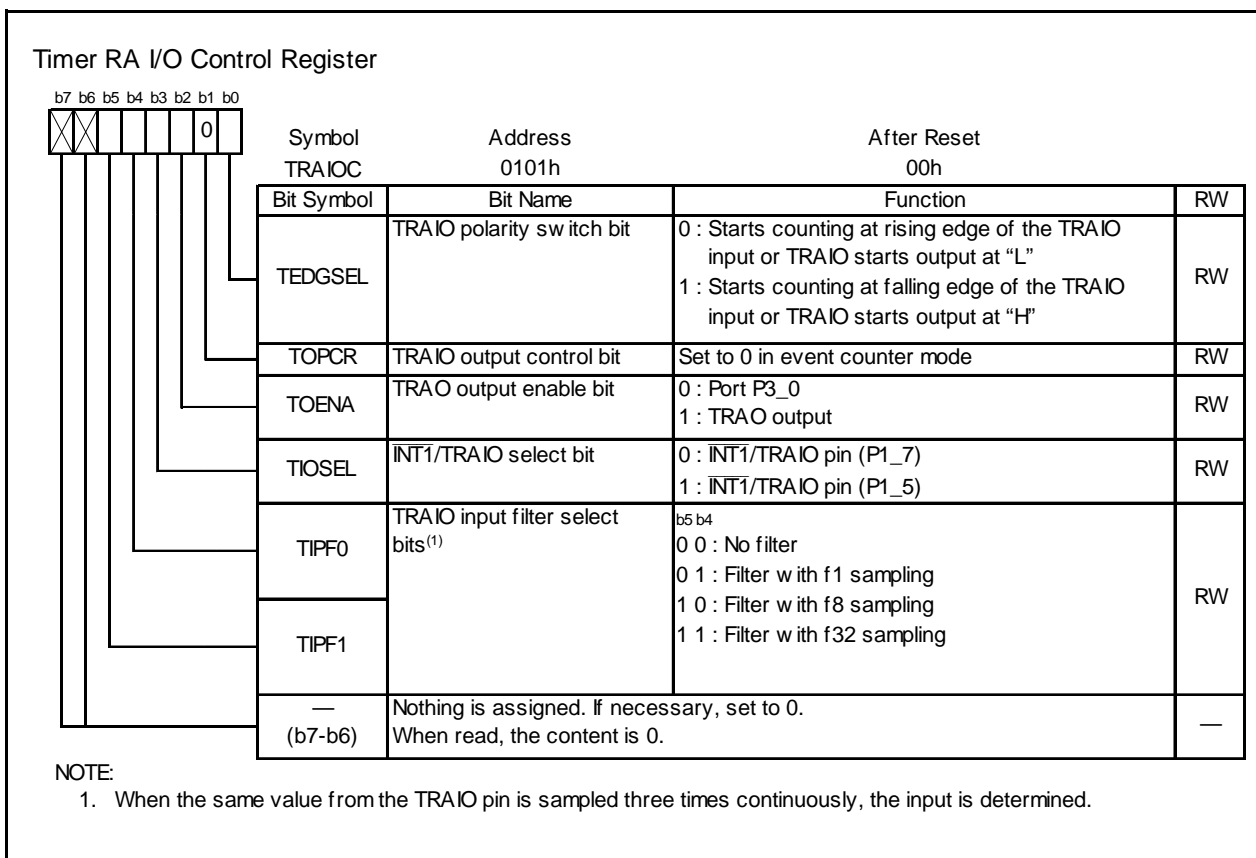


Figure 14.8 TRAI/O Register in Event Counter Mode

### 14.1.4 Pulse Width Measurement Mode

Pulse width measurement mode is mode to measure the pulse width of an external signal which inputs from the  $\overline{\text{INT1}}$ /TRAIO pin (see **Table 14.5 Pulse Width Measurement Mode Specifications**).

Figure 14.9 shows the TRAIOC Register in Pulse Width Measurement Mode and Figure 14.10 shows the Operating Example of Pulse Width Measurement Mode.

**Table 14.5 Pulse Width Measurement Mode Specifications**

Item	Specification
Count Sources	f1, f2, f8, fOCO
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 in the reload register is reloaded and the count is inherited</li> </ul>
Count Start Condition	Write 1 (count starts) to the TSTART bit in the TRACR register
Count Stop Conditions	<ul style="list-style-type: none"> <li>• Write 0 (count stops) to the TSTART bit in the TRACR register</li> <li>• Write 1 (count forcibly stops) 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	Measurement pulse input ( $\overline{\text{INT1}}$ interrupt input)
TRAIO Pin Function	Programmable I/O port
Read from Timer	The count value can be read by reading the TRA and TRAPRE registers.
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>14.1.1.1 Timer Write Control during Count Operation</b>).</li> </ul>
Select Functions	<ul style="list-style-type: none"> <li>• Measurement level select The TEDGSEL bit in the TRAIOC register can select during “H” or “L” level</li> <li>• Measurement 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>

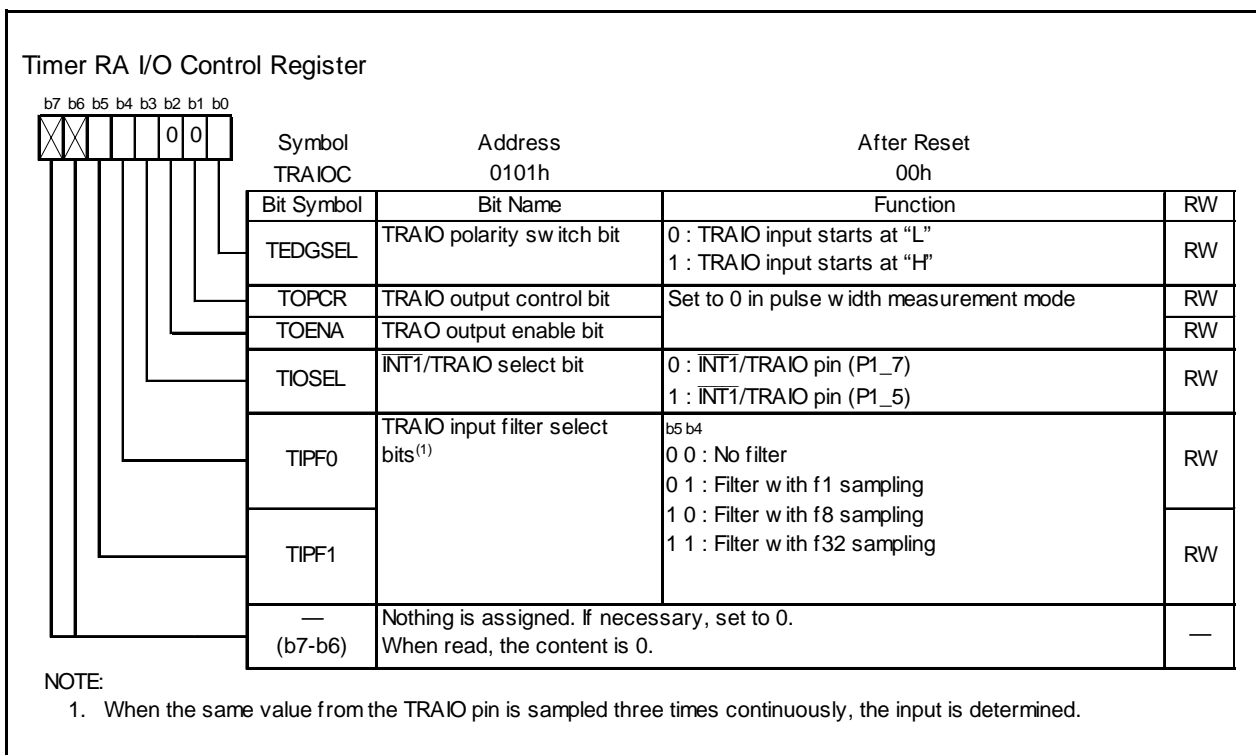


Figure 14.9 TRAI0C Register in Pulse Width Measurement Mode

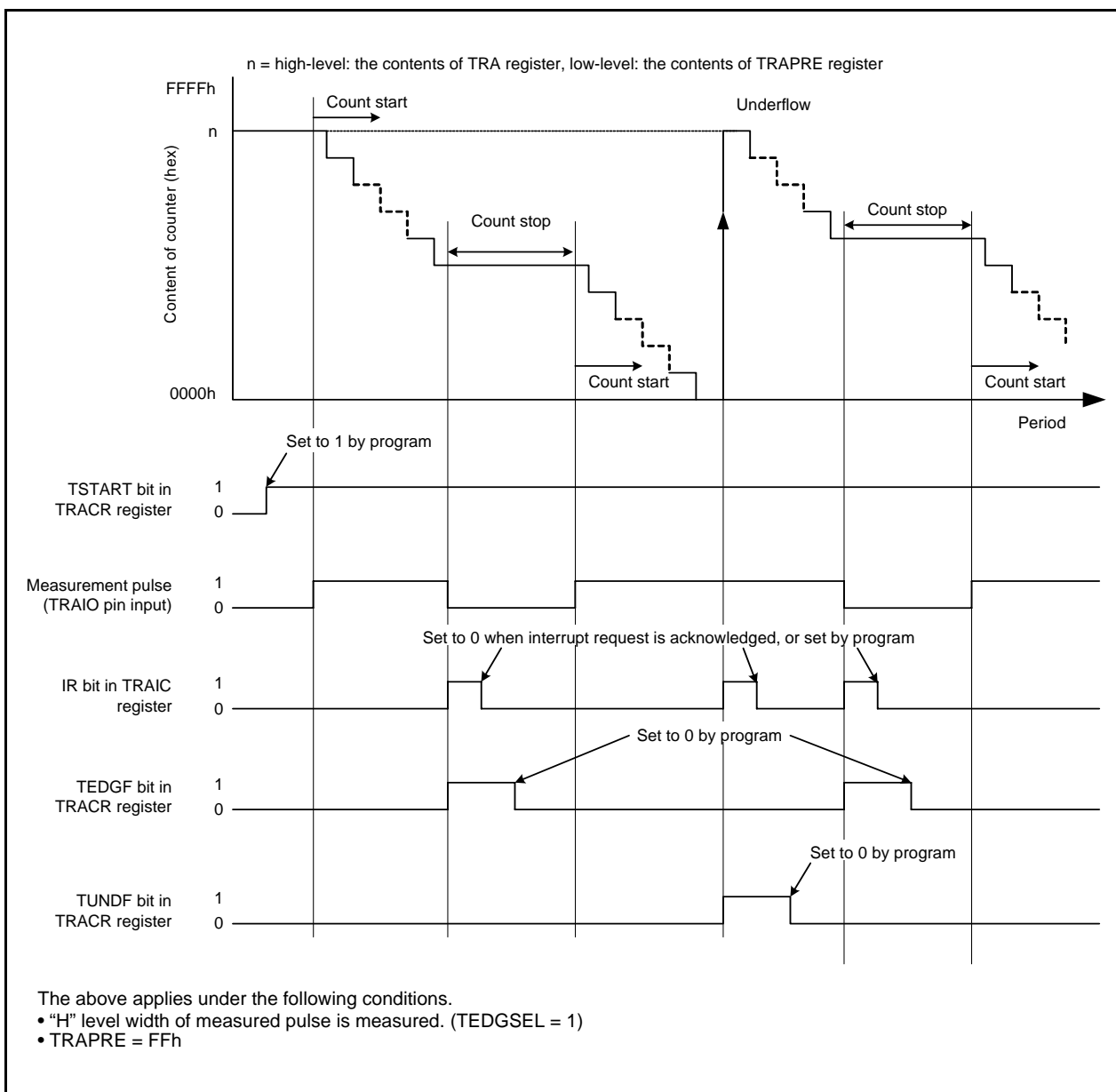


Figure 14.10 Operating Example of Pulse Width Measurement Mode

### 14.1.5 Pulse Period Measurement Mode

Pulse period measurement mode is mode to measure the pulse period of an external signal which inputs from the  $\overline{\text{INT1}}$ /TRAIO pin (see **Table 14.6 Pulse Period Measurement Mode Specifications**).

Figure 14.11 shows the TRAIOC Register in Pulse Period Measurement Mode and Figure 14.12 shows the Operating Example of Pulse Period Measurement Mode.

**Table 14.6 Pulse Period Measurement Mode Specifications**

Item	Specification
Count Sources	f1, f2, f8, fOCO
Count Operations	<ul style="list-style-type: none"> <li>• Decrement</li> <li>• After an active edge of measurement pulse is input, contents for the read-out buffer are retained at the first underflow of timer RA prescaler. Then timer RA reloads contents in the reload register at the second underflow of timer RA prescaler and continues counting.</li> </ul>
Count Start Condition	Write 1 (count start) to the TSTART bit in the TRACR register
Count Stop Conditions	<ul style="list-style-type: none"> <li>• Write 0 (count stop) to TSTART bit in the TRACR register</li> <li>• Write 1 (count forcibly stops) 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	Measurement 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 the TRA and TRAPRE registers.
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>14.1.1.1 Timer Write Control during Count Operation</b>).</li> </ul>
Select Functions	<ul style="list-style-type: none"> <li>• Measurement level select The TEDGSEL bit in the TRAIOC register can select the measurement period of input pulse.</li> <li>• Measurement 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 the pulse whose period is longer than twice of the timer RA prescaler period. Input the longer pulse for "H" width and "L" width than the timer RA prescaler period. If the shorter pulse than the period is input to the TRAIO pin, the input may be disabled.

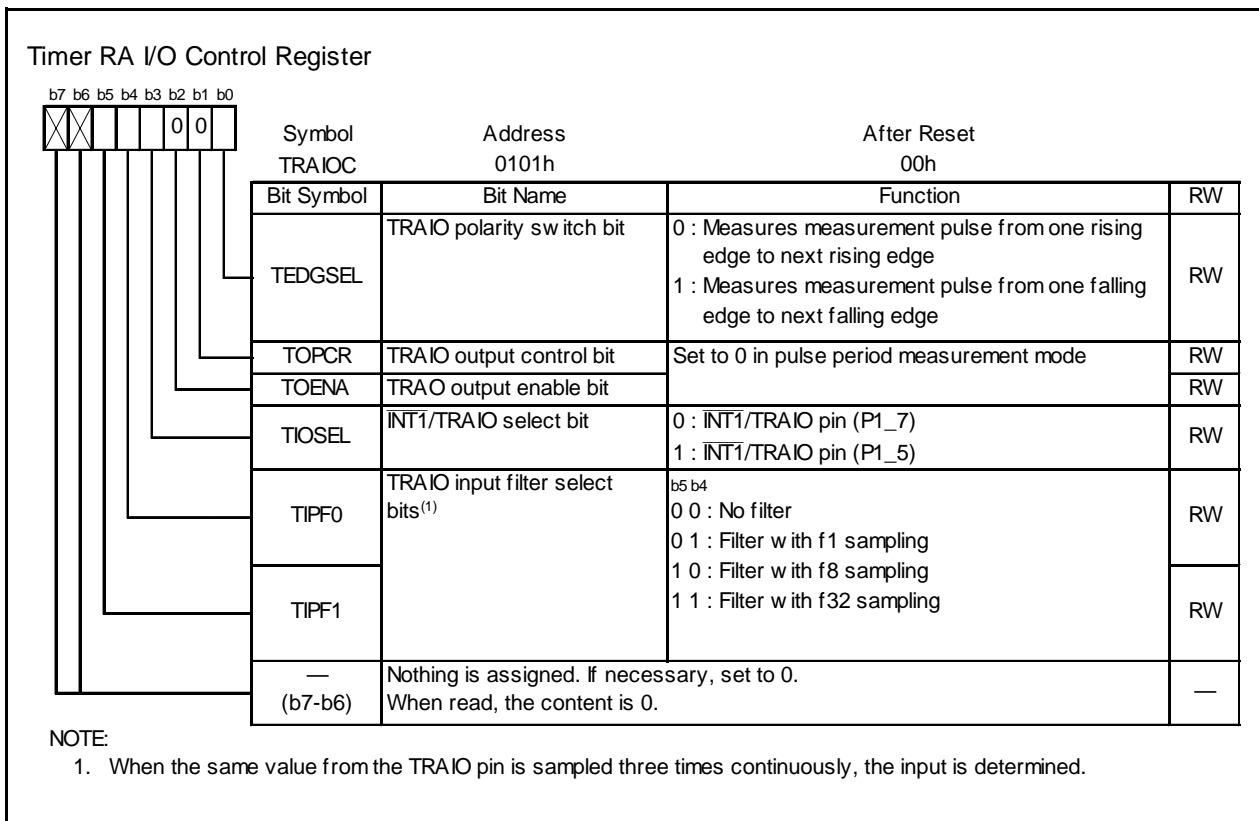
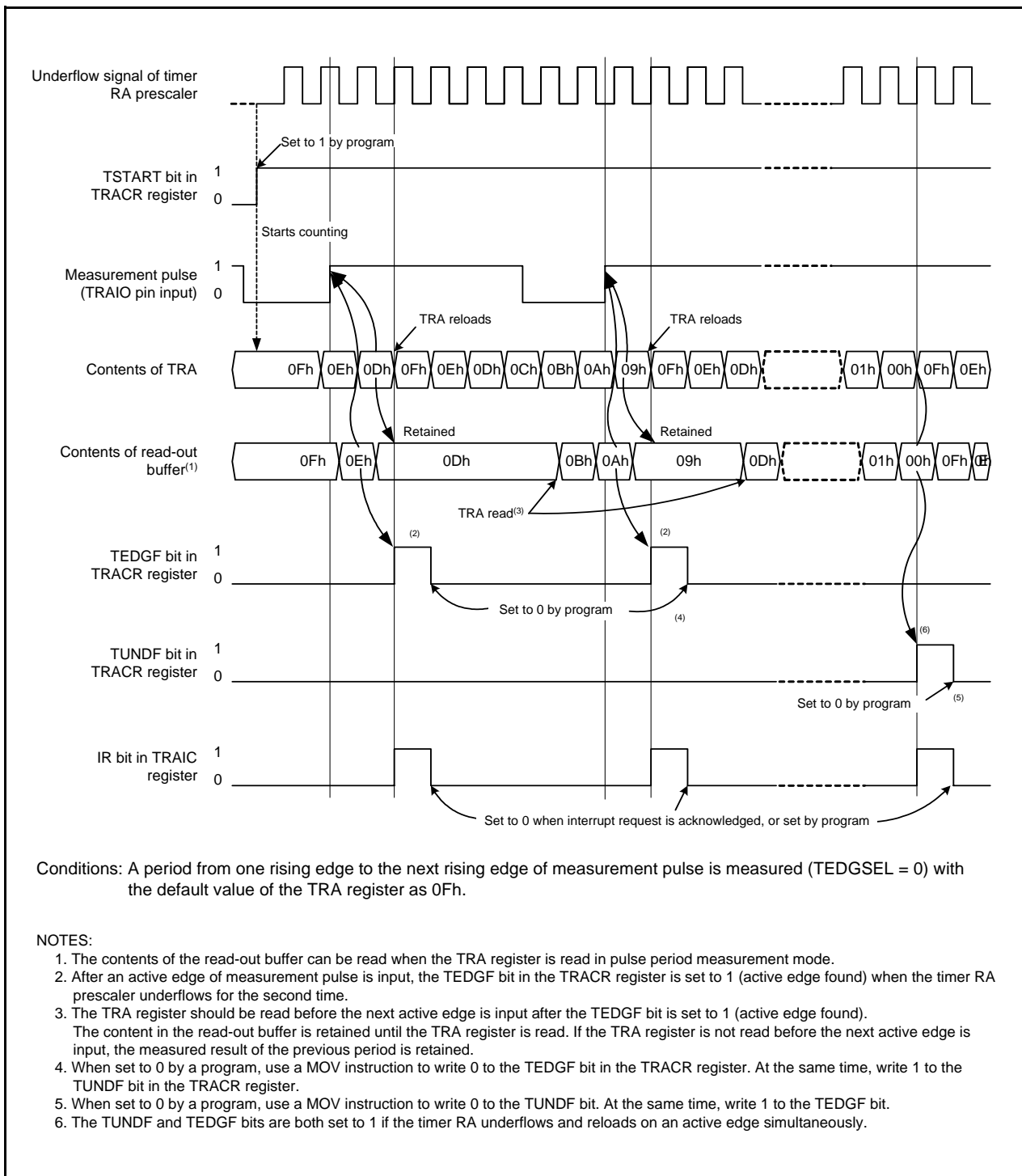


Figure 14.11 TRAI/O Register in Pulse Period Measurement Mode



**Figure 14.12 Operating Example of Pulse Period Measurement Mode**

### 14.1.6 Notes on Timer RA

- Timer RA stops counting after reset. Set the value to timer RA and timer RA prescaler before the count starts.
- Even if the prescaler and timer RA is read out in 16-bit units, these registers are read by 1 byte in the MCU. Consequently, the timer value may be updated during the period these two registers are being read.
- In pulse width measurement mode and pulse period measurement mode, the TEDGF and TUNDF bits in the TRACR register can be set to 0 by writing 0 to these bits by a program. However, these bits remain unchanged when 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 executed. At the time, 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 width measurement mode and pulse period measurement mode from other mode, the contents of the TEDGF and TUNDF bits are indeterminate. Write 0 to the TEDGF and TUNDF bits before the count starts.
- The TEDGF bit may be set to 1 by timer RA prescaler underflow which is generated for the first time since the count starts.
- When using the pulse period measurement mode, leave two periods or more of timer RA prescaler immediately after count starts, and 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 stops.  
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 retains 1 for 0 to 1 cycle of the count source after setting the TSTART bit to 0 (count stops) while the count is performing. 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, 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.



### 14.2 Timer RB

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

The prescaler and timer consist of the reload register and counter. (Refer to **Table 14.7 to 14.10 the Specification of Each Modes**).

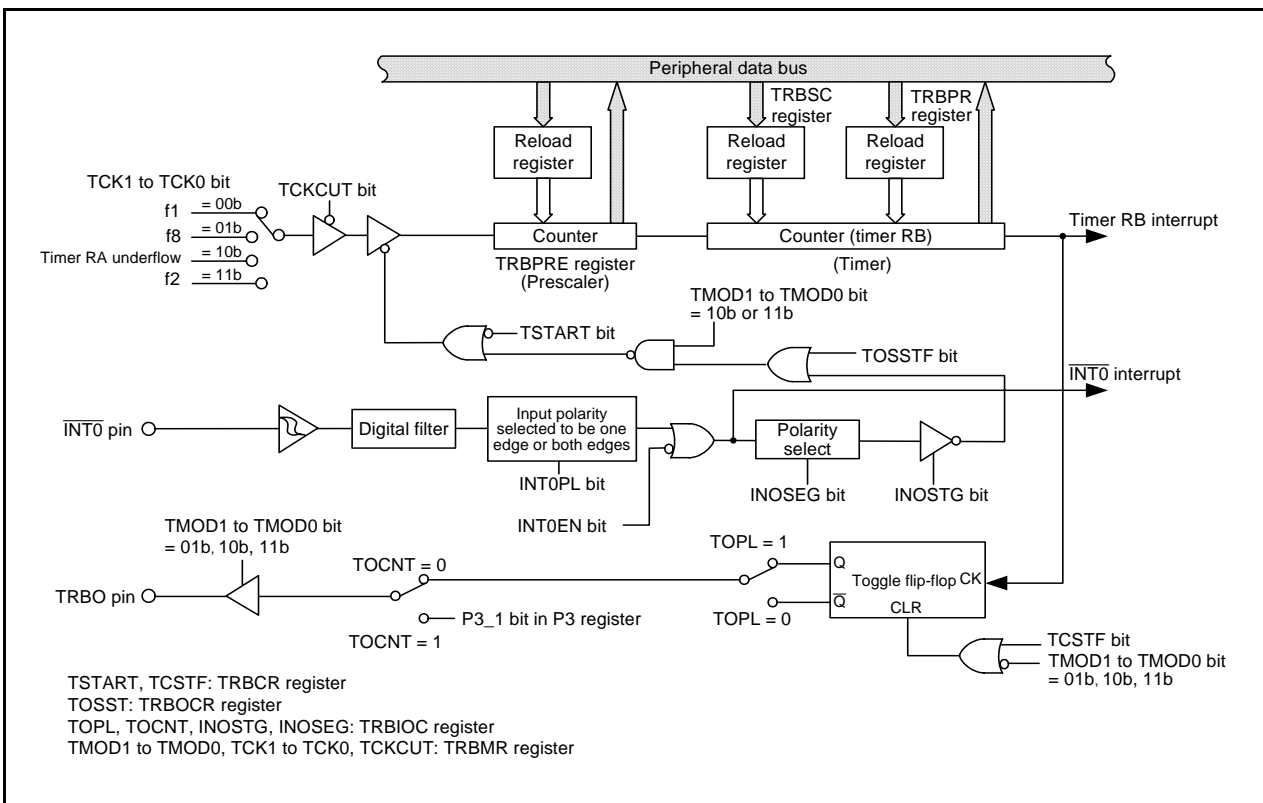
Timer RB contains the timer RB primary and timer RB secondary as the reload register.

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

Figure 14.13 shows the Block Diagram of Timer RB. Figures 14.14 and 14.16 show the registers associated with timer RB.

Timer RB contains 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 one-shot pulse.
- Programmable wait one-shot generation mode: The timer outputs delayed one-shot pulse.



**Figure 14.13 Block Diagram of Timer RB**

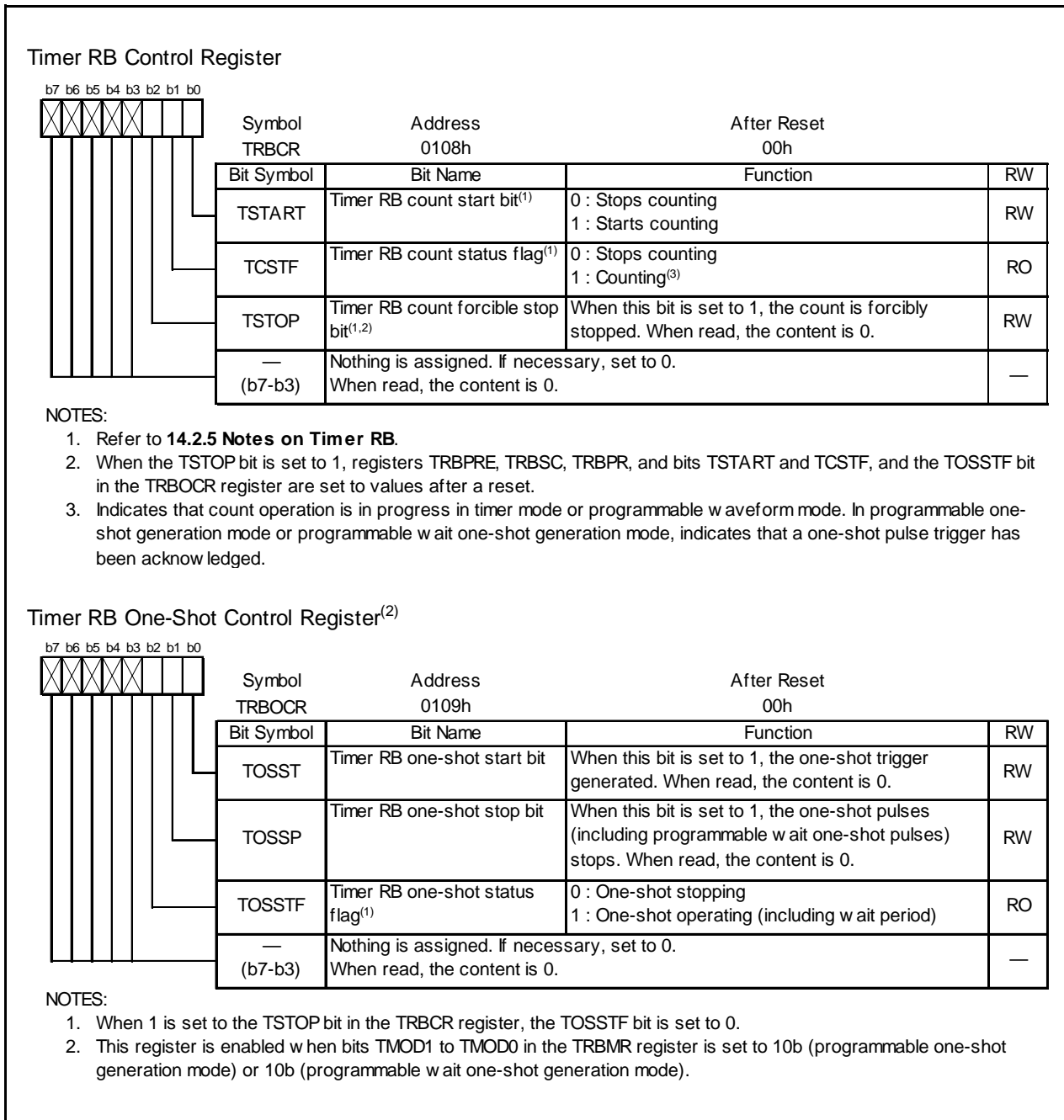


Figure 14.14 Registers TRBCR and TRBOCR

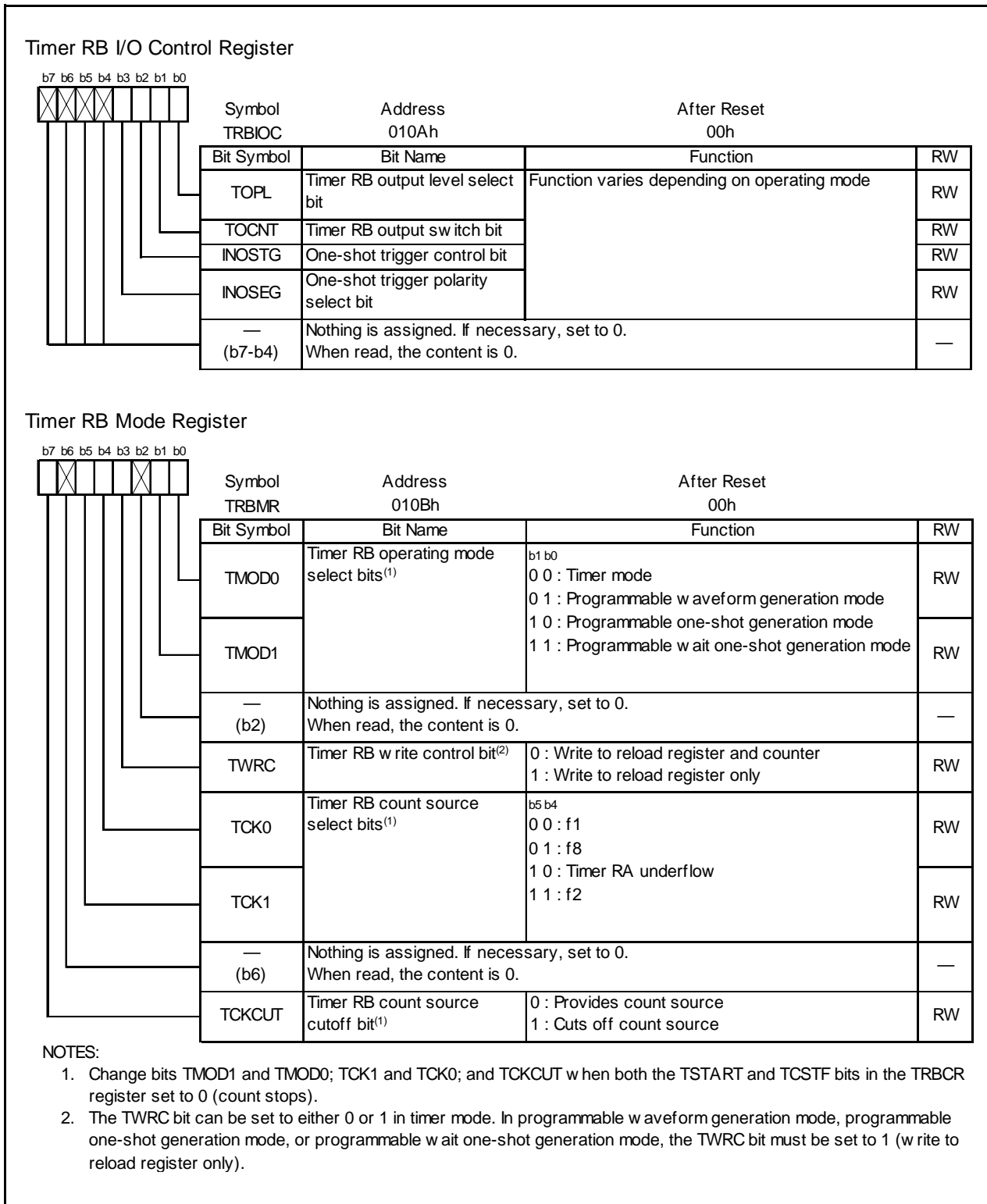


Figure 14.15 Registers TRBIOC and TRBMR

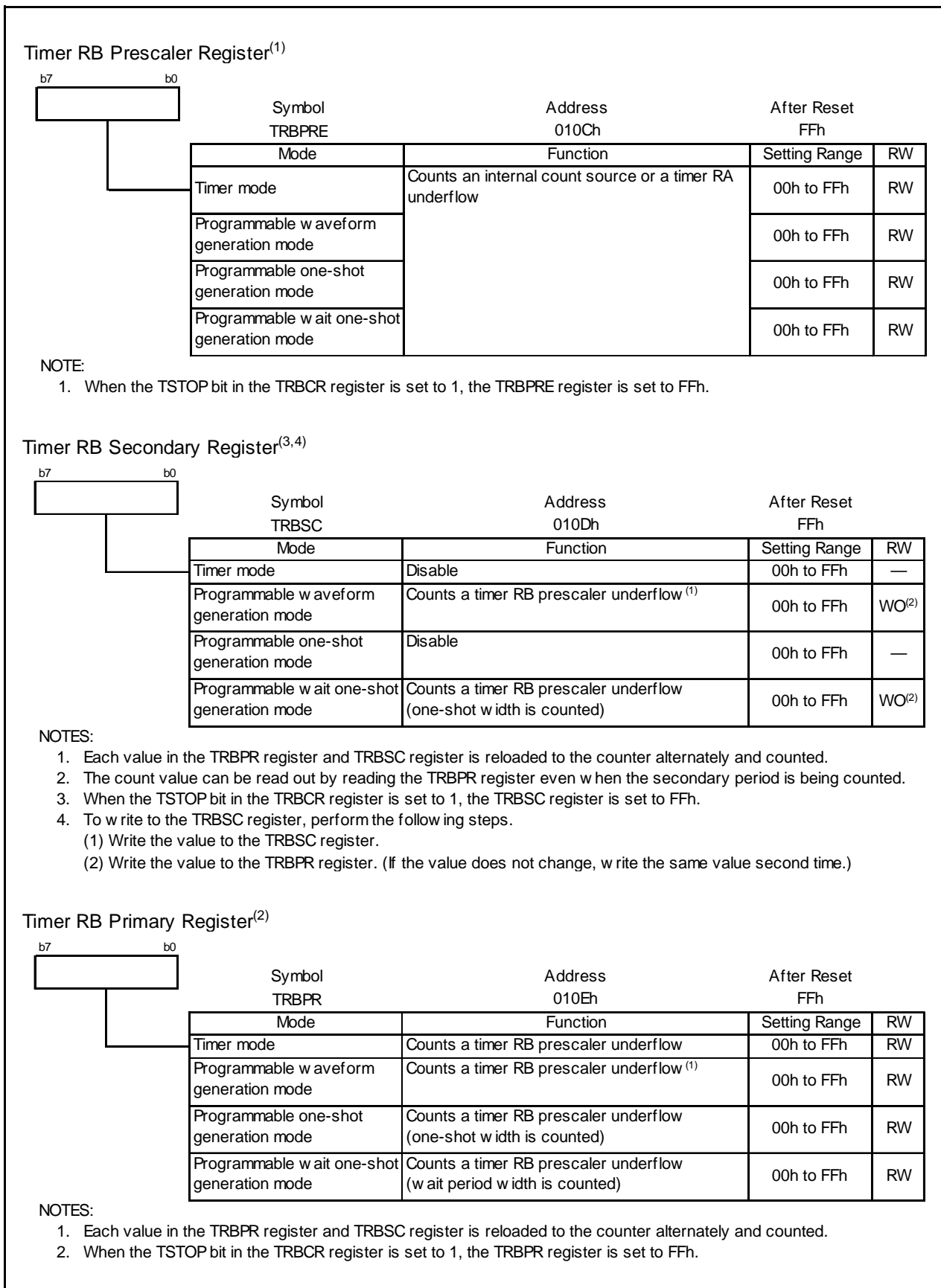


Figure 14.16 Registers TRBPRES, TRBSC, and TRBPR

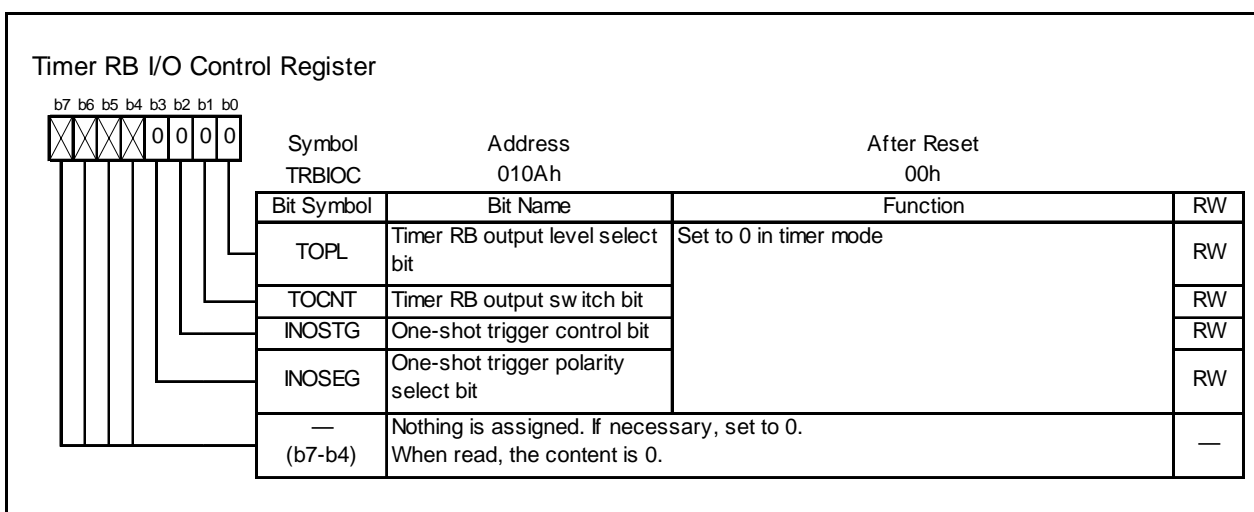
### 14.2.1 Timer Mode

Timer mode is mode to count a count source which is internally generated or timer RA underflow (see **Table 14.7 Timer Mode Specifications**). The TRBOCR and TRBSC registers are unused in timer mode.

Figure 14.17 shows the TRBIOC Register in Timer Mode.

**Table 14.7 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	Write 1 (count starts) to the TSTART bit in the TRBCR register
Count Stop Conditions	<ul style="list-style-type: none"> <li>Write 0 (count stops) to the TSTART bit in the TRBCR register</li> <li>Write 1 (count forcibly stop) 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 the TRBPR and TRBPRES registers
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>14.2.1.1 Timer Write Control during Count Operation.</b>)</li> </ul>



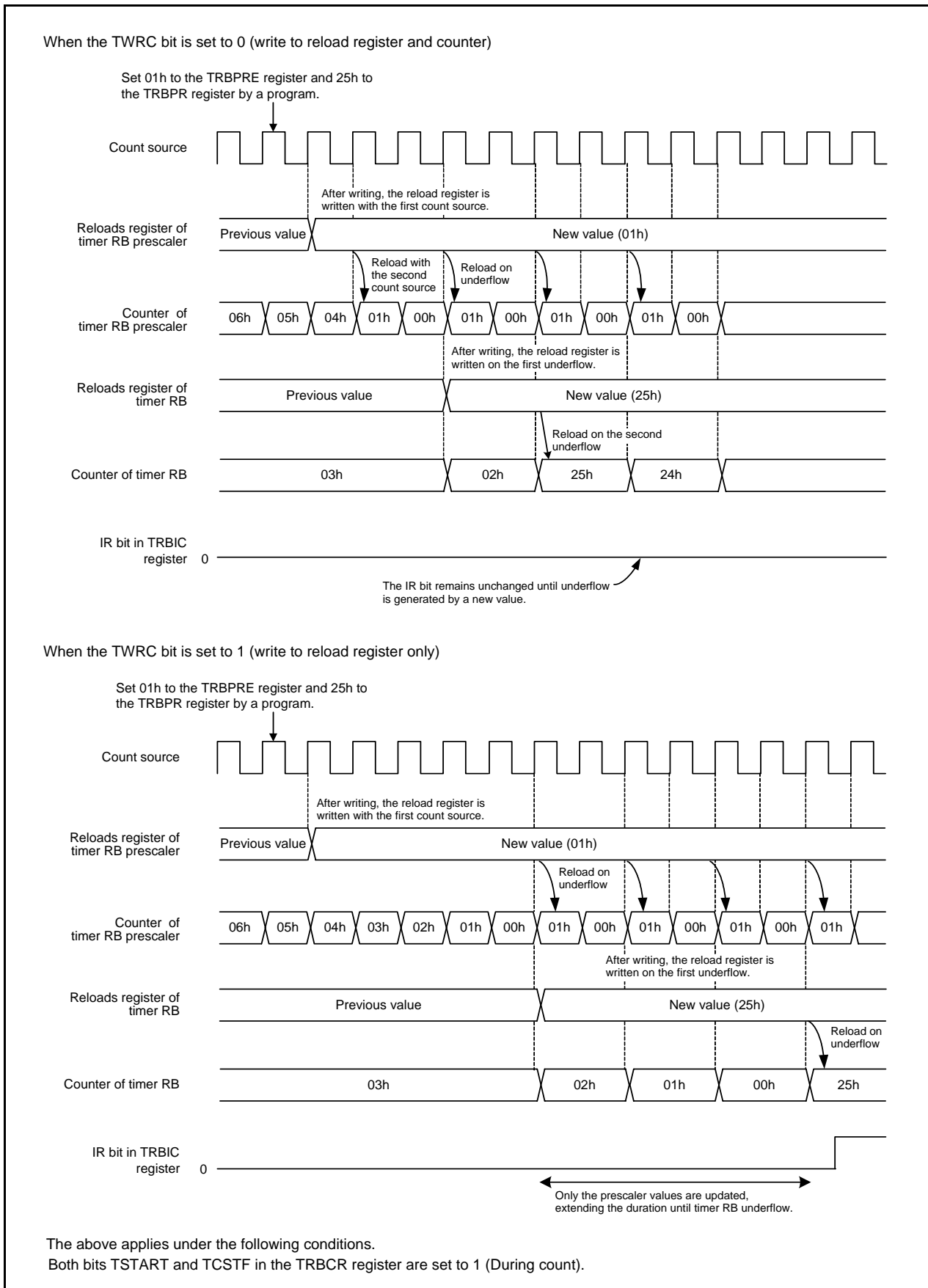
**Figure 14.17 TRBIOC Register in Timer Mode**

### 14.2.1.1 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 14.18 shows an Operating Example of Timer RB when Counter Value is Rewritten during Count Operation.



**Figure 14.18 Operating Example of Timer RB when Counter Value is Rewritten during Count Operation**

### 14.2.2 Programmable Waveform Generation Mode

Programmable waveform generation mode is mode to invert the signal output from the TRBO pin each time the counter underflows, while the values in the TRBPR and TRBSC registers are counted alternately (see **Table 14.8 Programmable Waveform Generation Mode Specifications**). A counting starts by counting the setting value in the TRBPR register. The TRBOCR register is unused in this mode.

Figure 14.19 shows the TRBIOC Register in Programmable Waveform Generation Mode. Figure 14.20 shows the Operation Example of Timer RB in Programmable Waveform Generation Mode.

**Table 14.8 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 $n$ : Setting value in TRBPRES register $m$ : Setting value in TRBPR register $p$ : Setting value in TRBSC register
Count Start Condition	Write 1 (count start) to the TSTART bit in the TRBCR register
Count Stop Conditions	<ul style="list-style-type: none"> <li>• Write 0 (count stop) to the TSTART bit in the TRBCR register</li> <li>• Write 1 (count forcibly stop) to the TSTOP bit in the TRBCR register</li> </ul>
Interrupt Request Generation Timing	In half of count source, after timer RB underflows during secondary period (at the same time as the TRBO output change) [timer RB interrupt]
TRBO Pin Function	Programmable output port or pulse output
INT0 Pin Function	Programmable I/O port or INT0 interrupt input
Read from Timer	The count value can be read out by reading the TRBPR and TRBPRES registers <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 can select the output level during primary and secondary periods.</li> <li>• TRBO pin output switch function Timer RB pulse output or P3_1 latch output is selected by the TOCNT bit in the TRBIOC register.<sup>(3)</sup></li> </ul>

**NOTES:**

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



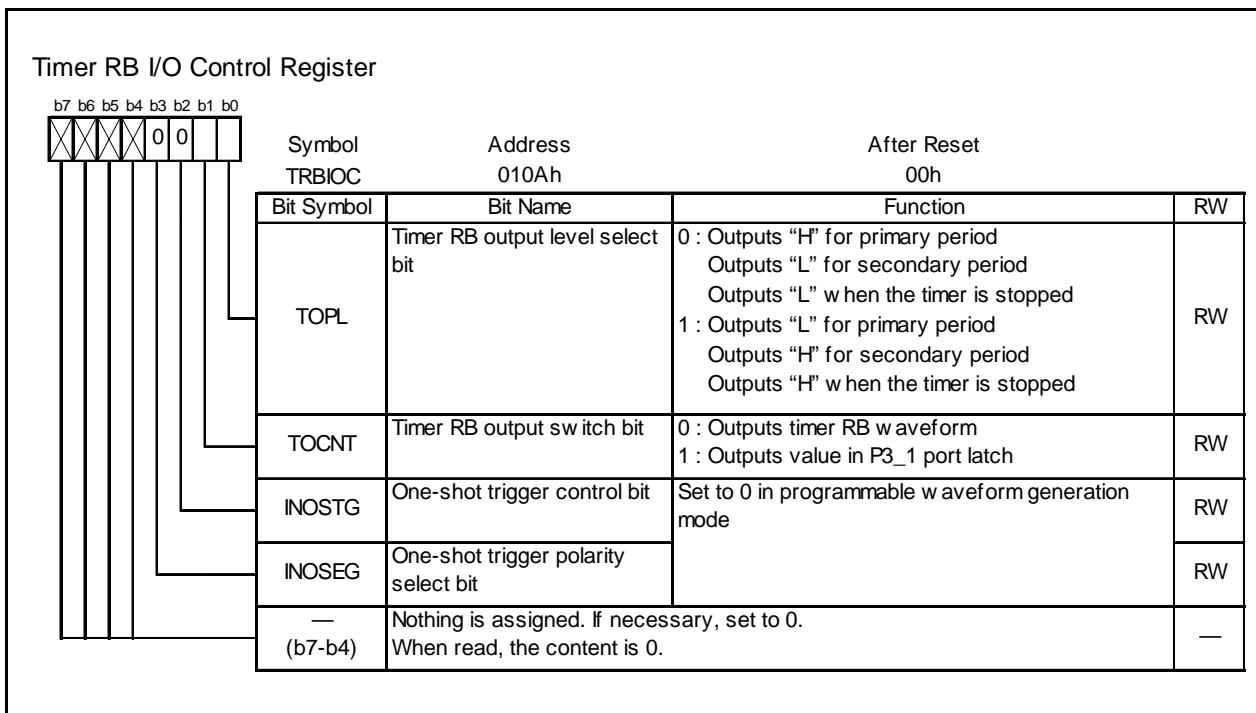


Figure 14.19 TRBIOC Register in Programmable Waveform Generation Mode

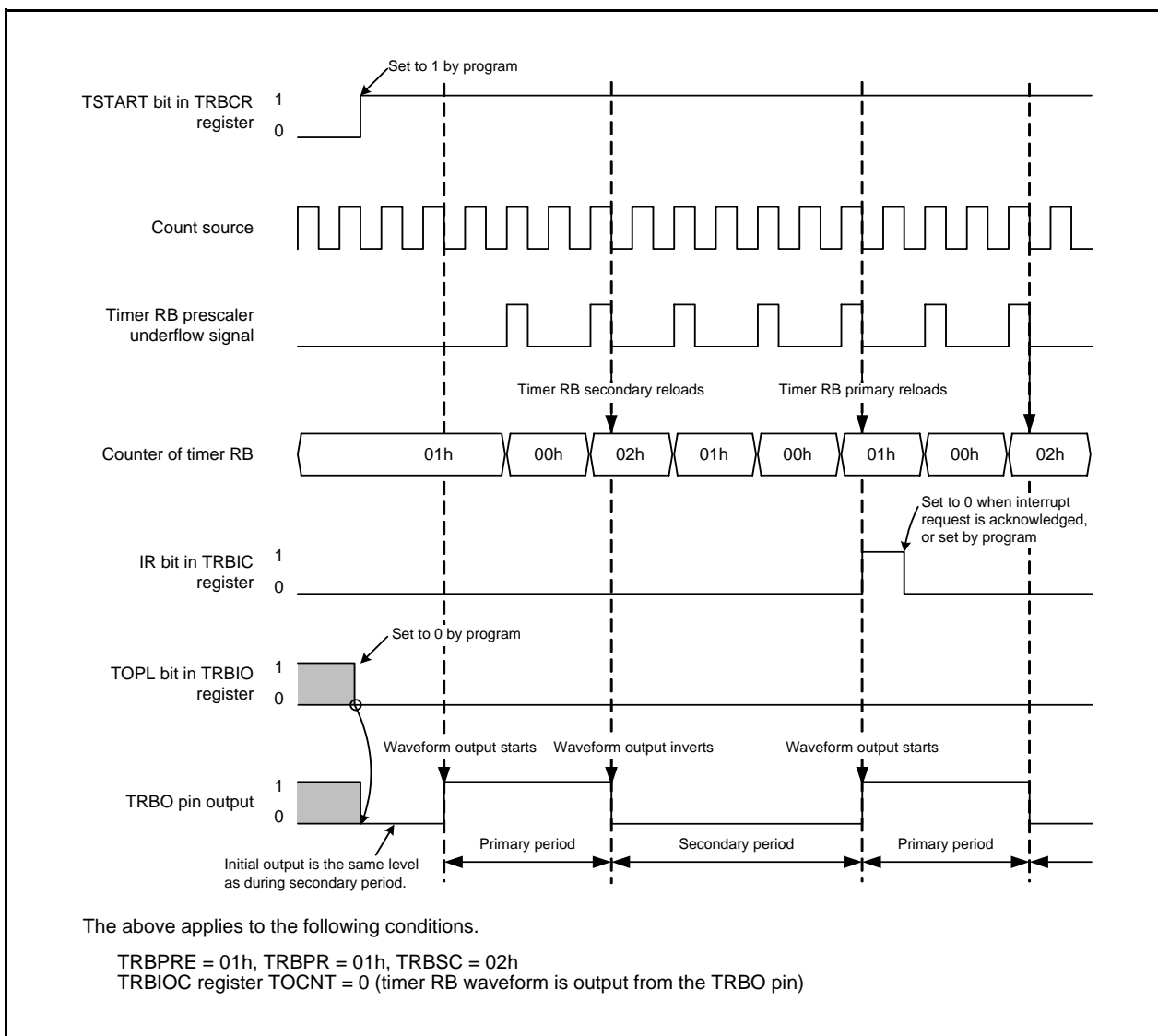


Figure 14.20 Operation Example of Timer RB in Programmable Waveform Generation Mode

### 14.2.3 Programmable One-shot Generation Mode

Programmable one-shot generation mode is mode to output the one-shot pulse from the TRBO pin by a program or an external trigger input (input to the  $\overline{\text{INT0}}$  pin) (see **Table 14.9 Programmable One-Shot Generation Mode Specifications**). When a trigger is generated, the timer starts operating from the point only once for a given period equal to the set value in the TRBPR register. The TRBSC register is unused in this mode.

Figure 14.21 shows the TRBIOC Register in Programmable One-Shot Generation Mode. Figure 14.22 shows the Operation Example of Programmable One-Shot Generation Mode.

**Table 14.9 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 a 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 cycles of 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 the TRBPR and TRBPRES registers.
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 can select the output level of the one-shot pulse waveform.</li> <li>• One-shot trigger select function Refer to <b>14.2.3.1 One-Shot Trigger Selection</b>.</li> </ul>

**NOTES:**

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

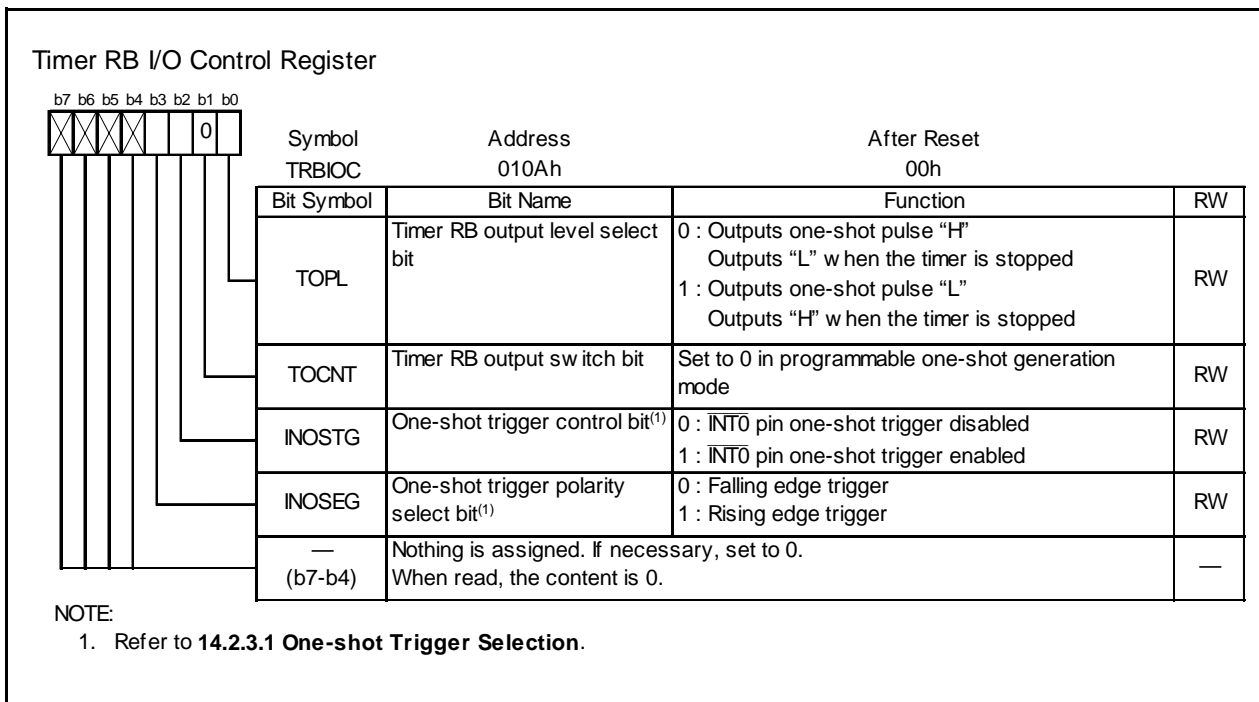


Figure 14.21 TRBIOC Register in Programmable One-Shot Generation Mode

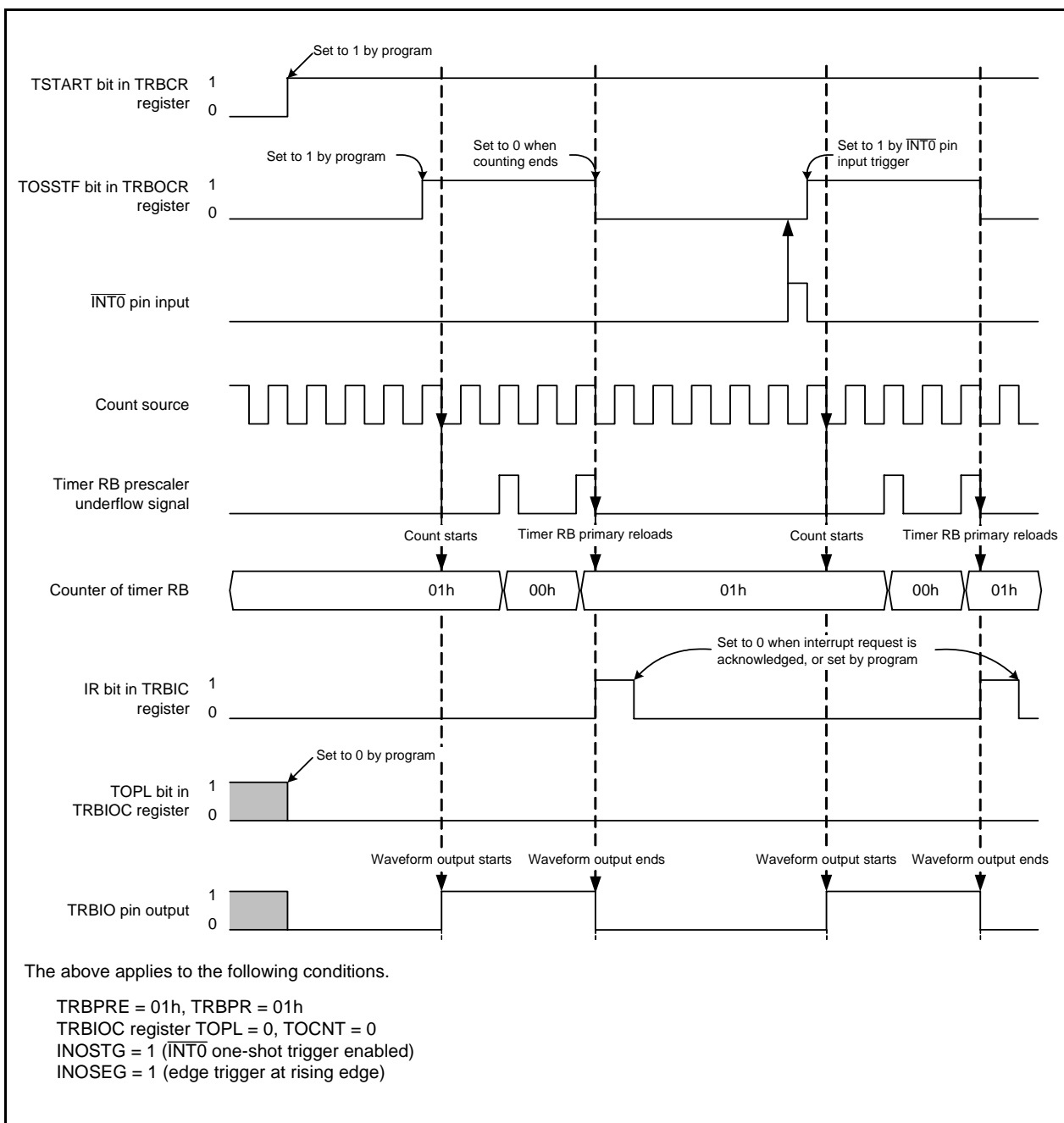


Figure 14.22 Operation Example of Programmable One-Shot Generation Mode

### 14.2.3.1 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 INTOEN 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 **12. 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 INTOIC register changes.

#### 14.2.4 Programmable Wait One-shot Generation Mode

Programmable wait one-shot generation mode is mode to output the one-shot pulse from the TRBO pin by a program or an external trigger input (input to the  $\overline{\text{INT0}}$  pin) (see **Table 14.10 Programmable Wait One-Shot Generation Mode Specifications**). When a trigger is generated from this point, the timer starts outputting pulses 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 14.23 shows the TRBIOC Register in Programmable Wait One-Shot Generation Mode. Figure 14.24 shows the Operation Example of Programmable Wait One-Shot Generation Mode.

**Table 14.10 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 setting value in timer RB primary</li> <li>• When a count of timer RB primary underflows, the timer reloads the contents of the timer RB secondary before the count continues.</li> <li>• When a count of timer RB secondary underflows, the timer reloads the contents of the timer RB primary before the count completes and the TOSSTF bit is set to 0 (one-shot stops).</li> <li>• When a 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 $n$ : Setting value in the TRBPRES register, $m$ : Setting value in the TRBPR register <sup>(2)</sup>
One-Shot Pulse Output Time	$(n+1)(p+1)/f_i$ $f_i$ : Count source frequency $n$ : Setting value in the TRBPRES register, $p$ : Setting value 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 INT0 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 0 (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 cycles 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
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 the TRBPR and TRBPRES registers.
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 TRBIO register can select the output level of the one-shot pulse waveform.</li> <li>• One-shot trigger select function Refer to <b>14.2.3.1 One-Shot Trigger Selection</b>.</li> </ul>

## NOTES:

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



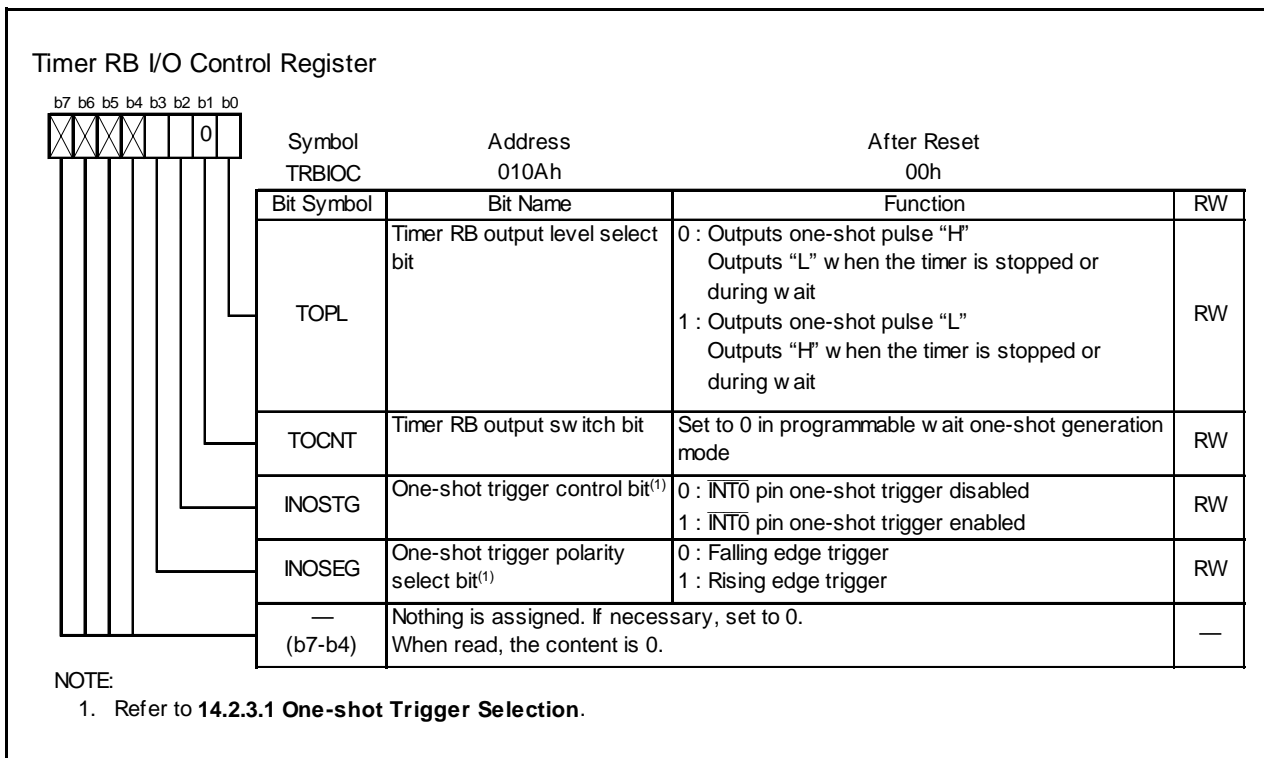


Figure 14.23 TRBIOC Register in Programmable Wait One-Shot Generation Mode

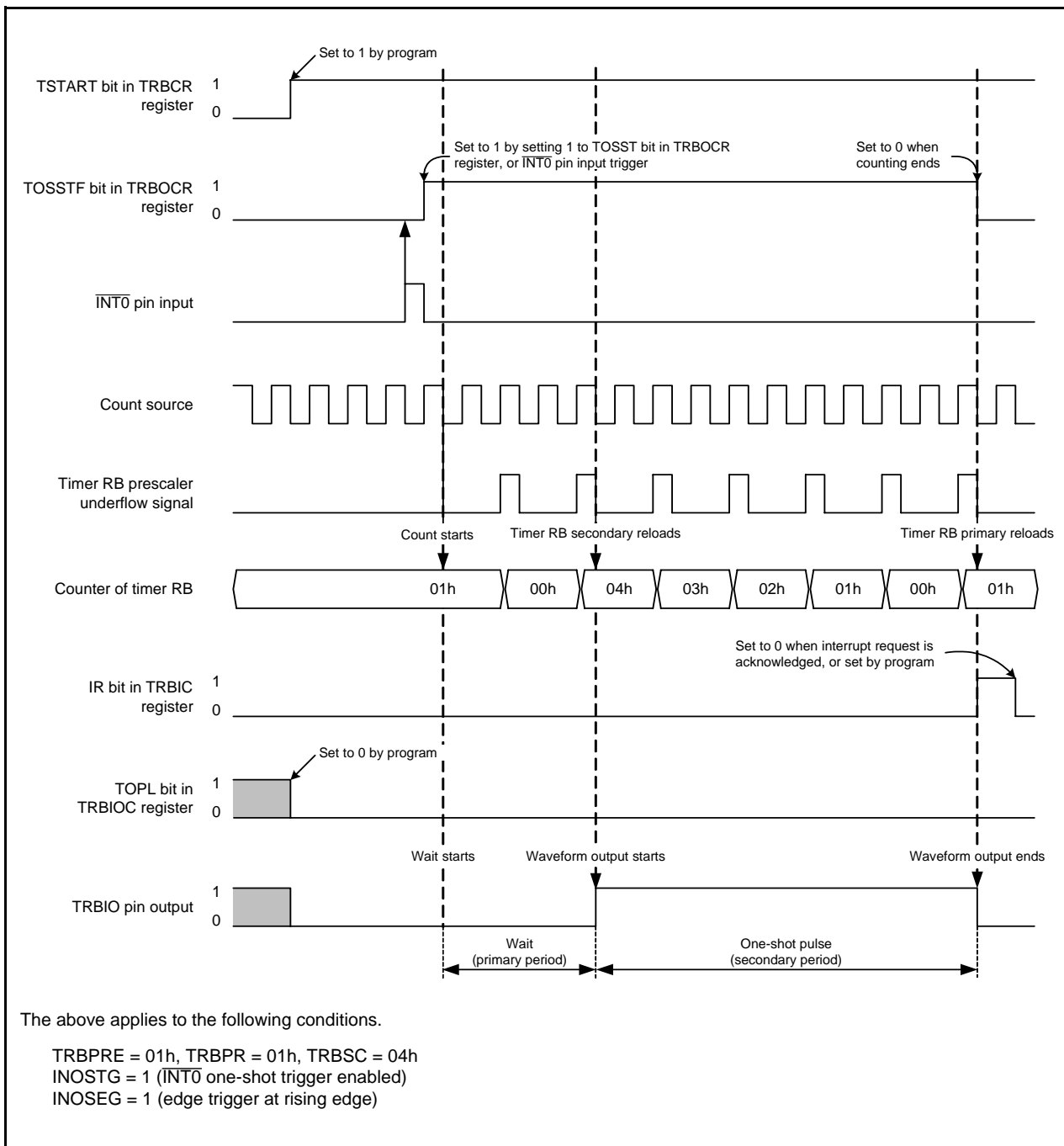


Figure 14.24 Operation Example of Programmable Wait One-Shot Generation Mode

### 14.2.5 Notes on Timer RB

- Timer RB stops counting after reset. Set the value to timer RB and timer RB prescaler before the count starts.
- Even if the prescaler and timer RB is read out in 16-bit units, these registers are read by 1 byte in the MCU. Consequently, the timer value may be updated during the period 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, read the timer count value in programmable one-shot generation mode and programmable wait one-shot generation mode before the timer stops.
- The TCSTF bit retains 0 (count stops) for 1 to 2 cycles of the count source after setting the TSTART bit to 1 (count starts) while the count stops.

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

The TCSTF bit retains 1 for 1 to 2 cycles of the count source after setting the TSTART bit to 0 (count stops) while the count is performing. 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, TRBPRE, TRBSC, 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.

#### 14.2.5.1 Timer mode

The following workaround should be performed in timer mode.

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.

### 14.2.5.2 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 14.25 and 14.26.

The following shows the detailed workaround examples.

- Workaround example (a):  
As shown in Figure 14.25, write to registers TRBSC and TRBPR in the timer RB interrupt routine. These write operations must be completed by the beginning of period A.

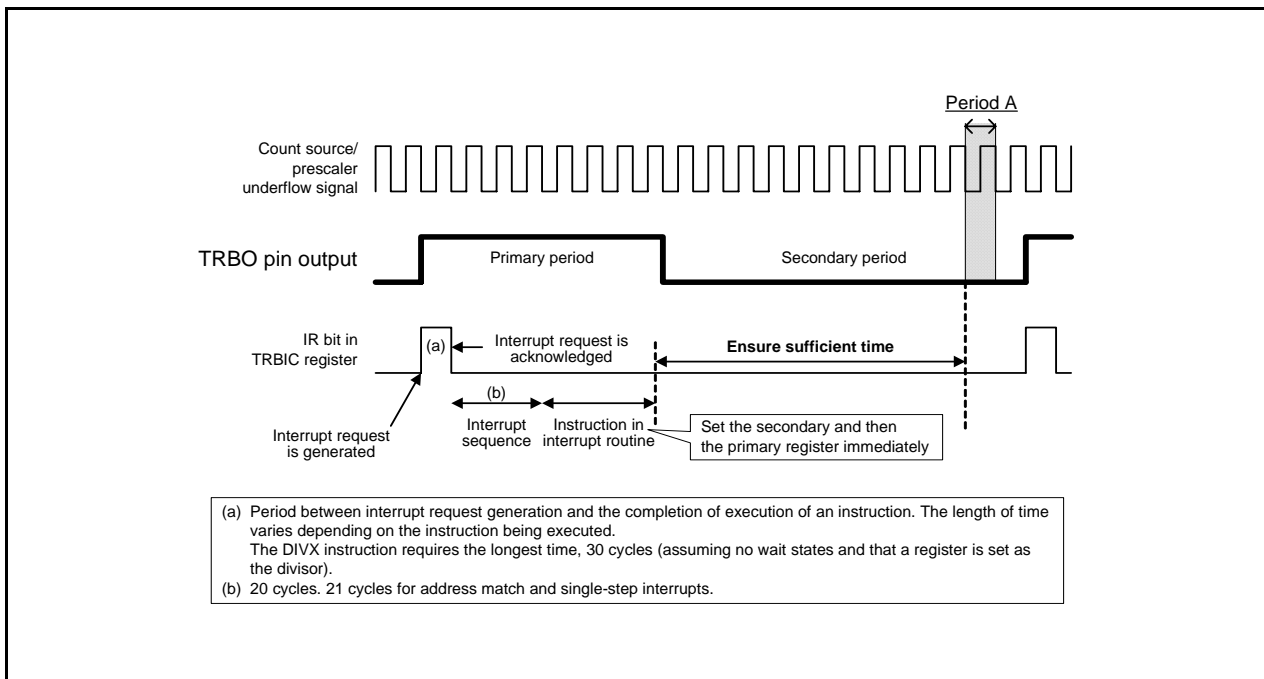
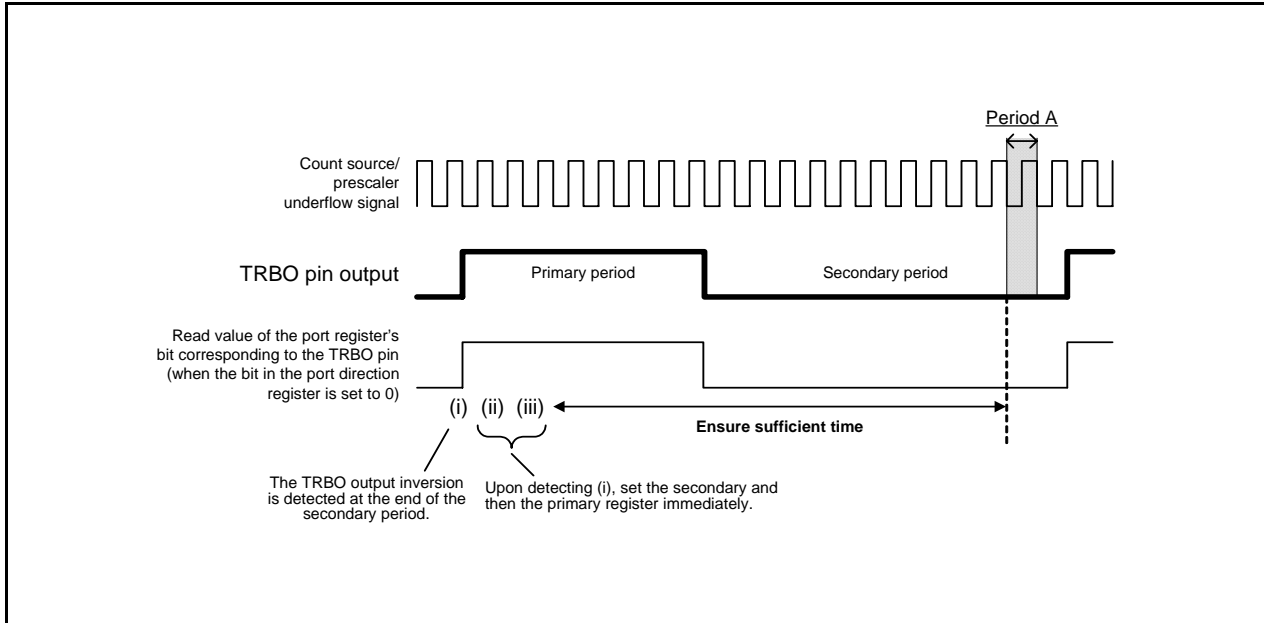


Figure 14.25 Workaround Example (a) When Timer RB Interrupt is Used

- Workaround example (b):

As shown in Figure 14.26 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 14.26 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.

### 14.2.5.3 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.

#### 14.2.5.4 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 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.

### 14.3 Timer RD

Timer RD has 2 16-bit timers (channels 0 and 1). Each channel has 4 I/O pins.

The operation clock of Timer RD is f1 or fOCO40M.

Table 14.11 lists the Timer RD Operation Clocks.

**Table 14.11 Timer RD Operation Clocks**

Condition	Operation Clock of Timer RD
The count source is f1, f2, f4, f8, f32 and TRDCLK input. (The TCK2 to TCK0 bits in the TRDCR0 and TRDCR1 registers are set to 000b to 101b.)	f1
The count source is fOCO40M. (The TCK2 to TCK0 bits in the TRDCR0 and TRDCR1 registers are set to 110b.)	fOCO40M

Figure 14.27 shows the Block Diagram of Timer RD. Timer RD has 5 modes:

- Timer mode
  - Input capture function                      Transfer the counter value to a register as a trigger of the external signal
  - Output compare function                      Detect the register value match with a counter  
(Pin output can be changed at detection)

The following 4 modes using the output compare function.

- PWM mode    Output any-wide pulse continuously
- Reset synchronous PWM mode                      Output three-phase waveforms (6) without sawtooth wave modulation and dead time
- Complementary PWM mode                      Output three-phase waveforms (6) with triangular wave modulation and dead time
- PWM3 mode    Output PWM waveform (2) with same period

In the input capture function, output compare function and PWM mode, Channels 0 and 1 have the equivalent functions, and functions or modes can be selected every pin. Also, a combination of these functions and modes can be used in 1 channel.

In reset synchronous PWM mode, complementary PWM mode and PWM3 mode, a waveform is output with a combination of counters and registers in Channels 0 and 1.

Tables 14.12 to 14.20 lists the Pin Functions of timer RD.

**Table 14.12 Pin Functions TRDIOA0/TRDCLK(P2\_0)**

Register	TRDOER1	TRDFCR			TRDIOA0		Function
Bit	EA0	PWM3	STCLK	CMD1, CMD0	IOA3	IOA2_IOA0	
Setting value	0	0	0	00b	X	XXXb	PWM3 mode waveform output
	0	1	0	00b	1	001b, 01Xb	Timer mode waveform output (output compare function)
	X	1	0	00b	X	1XXb	Timer mode trigger input (input capture function) <sup>(1)</sup>
		1	1	XXb	X	000b	External clock input (TRDCLK) <sup>(1)</sup>
	Other than above						

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2\_0 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function) and external clock input (TRDCLK).

**Table 14.13 Pin Functions TRDIOB0(P2\_1)**

Register	TRDOER1	TRDFCR		TRDPMR	TRDIOA0	Function
Bit	EB0	PWM3	CMD1, CMD0	PWMB0	IOB2_IOB0	
Setting value	0	X	1Xb	X	XXXb	Complementary PWM mode waveform output
	0	X	01b	X	XXXb	Reset synchronous PWM mode waveform output
	0	0	00b	X	XXXb	PWM3 mode waveform output
	0	1	00b	1	XXXb	PWM mode waveform output
	0	1	00b	0	001b, 01Xb	Timer mode waveform output (output compare function)
	X	1	00b	0	1XXb	Timer mode trigger input (input capture function) <sup>(1)</sup>
	Other than above					

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2\_1 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

**Table 14.14 Pin Functions TRDIOC0(P2\_2)**

Register	TRDOER1	TRDFCR		TRDPMR	TRDIOA0	Function
Bit	EC0	PWM3	CMD1, CMD0	PWMC0	IOC2_IOC0	
Setting value	0	X	1Xb	X	XXXb	Complementary PWM mode waveform output
	0	X	01b	X	XXXb	Reset synchronous PWM mode waveform output
	0	1	00b	1	XXXb	PWM mode waveform output
	0	1	00b	0	001b, 01Xb	Timer mode waveform output (output compare function)
	X	1	00b	0	1XXb	Timer mode trigger input (input capture function) <sup>(1)</sup>
	Other than above					

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2\_2 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).



**Table 14.15 Pin Functions TRDIOD0(P2\_3)**

Register	TRDOER1	TRDFCR		TRDPMR	TRDIORC0	Function
Bit	ED0	PWM3	CMD1, CMD0	PWMD0	IOD2_IOD0	
Setting value	0	X	1Xb	X	XXXb	Complementary PWM mode waveform output
	0	X	01b	X	XXXb	Reset synchronous PWM mode waveform output
	0	1	00b	1	XXXb	PWM mode waveform output
	0	1	00b	0	001b, 01Xb	Timer mode waveform output (output compare function)
	X	1	00b	0	1XXb	Timer mode trigger input (input capture function) <sup>(1)</sup>
Other than above						I/O port

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2\_3 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

**Table 14.16 Pin Functions TRDIOA1(P2\_4)**

Register	TRDOER1	TRDFCR		TRDIOA1	Function
Bit	EA1	PWM3	CMD1, CMD0	IOA2_IOA0	
Setting value	0	X	1Xb	XXXb	Complementary PWM mode waveform output
	0	X	01b	XXXb	Reset synchronous PWM mode waveform output
	0	1	00b	001b, 01Xb	Timer mode waveform output (output compare function)
	X	1	00b	1XXb	Timer mode trigger input (input capture function) <sup>(1)</sup>
Other than above					I/O port

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2\_4 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

**Table 14.17 Pin Functions TRDIOB1(P2\_5)**

Register	TRDOER1	TRDFCR		TRDPMR	TRDIOA1	Function
Bit	EB1	PWM3	CMD1, CMD0	PWMB1	IOB2_IOB0	
Setting value	0	X	1Xb	X	XXXb	Complementary PWM mode waveform output
	0	X	01b	X	XXXb	Reset synchronous PWM mode waveform output
	0	1	00b	1	XXXb	PWM mode waveform output
	0	1	00b	0	001b, 01Xb	Timer mode waveform output (output compare function)
	X	1	00b	0	1XXb	Timer mode trigger input (input capture function) <sup>(1)</sup>
Other than above						I/O port

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2\_5 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

**Table 14.18 Pin Functions TRDIOC1(P2\_6)**

Register	TRDOER1	TRDFCR		TRDPMR	TRDIORC1	Function
Bit	EC1	PWM3	CMD1, CMD0	PWMC1	IOC2_IOC0	
Setting value	0	X	1Xb	X	XXXb	Complementary PWM mode waveform output
	0	X	01b	X	XXXb	Reset synchronous PWM mode waveform output
	0	1	00b	1	XXXb	PWM mode waveform output
	0	1	00b	0	001b, 01Xb	Timer mode waveform output (output compare function)
	X	1	00b	0	1XXb	Timer mode trigger input (input capture function) <sup>(1)</sup>
Other than above						I/O port

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2\_6 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

**Table 14.19 Pin Functions TRDIOD1(P2\_7)**

Register	TRDOER1	TRDFCR		TRDPMR	TRDIORC1	Function
Bit	ED1	PWM3	CMD1, CMD0	PWMD1	IOD2_IOD0	
Setting value	0	X	1Xb	X	XXXb	Complementary PWM mode waveform output
	0	X	01b	X	XXXb	Reset synchronous PWM mode waveform output
	0	1	00b	1	XXXb	PWM mode waveform output
	0	1	00b	0	001b, 01Xb	Timer mode waveform output (output compare function)
	X	1	00b	0	1XXb	Timer mode trigger input (input capture function) <sup>(1)</sup>
Other than above						I/O port

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2\_7 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

**Table 14.20 Pin Functions  $\overline{\text{INT0}}$ (P4\_5)**

Register	TRDOER2	INTEN		PD4	Function
Bit	PTO	INT0PL	INT0EN	PD4_5	
Setting value	1	0	1	0	Pulse output forced cutoff signal input
Other than above					I/O port or $\overline{\text{INT0}}$ interrupt input

X: can be 0 or 1, no change in outcome

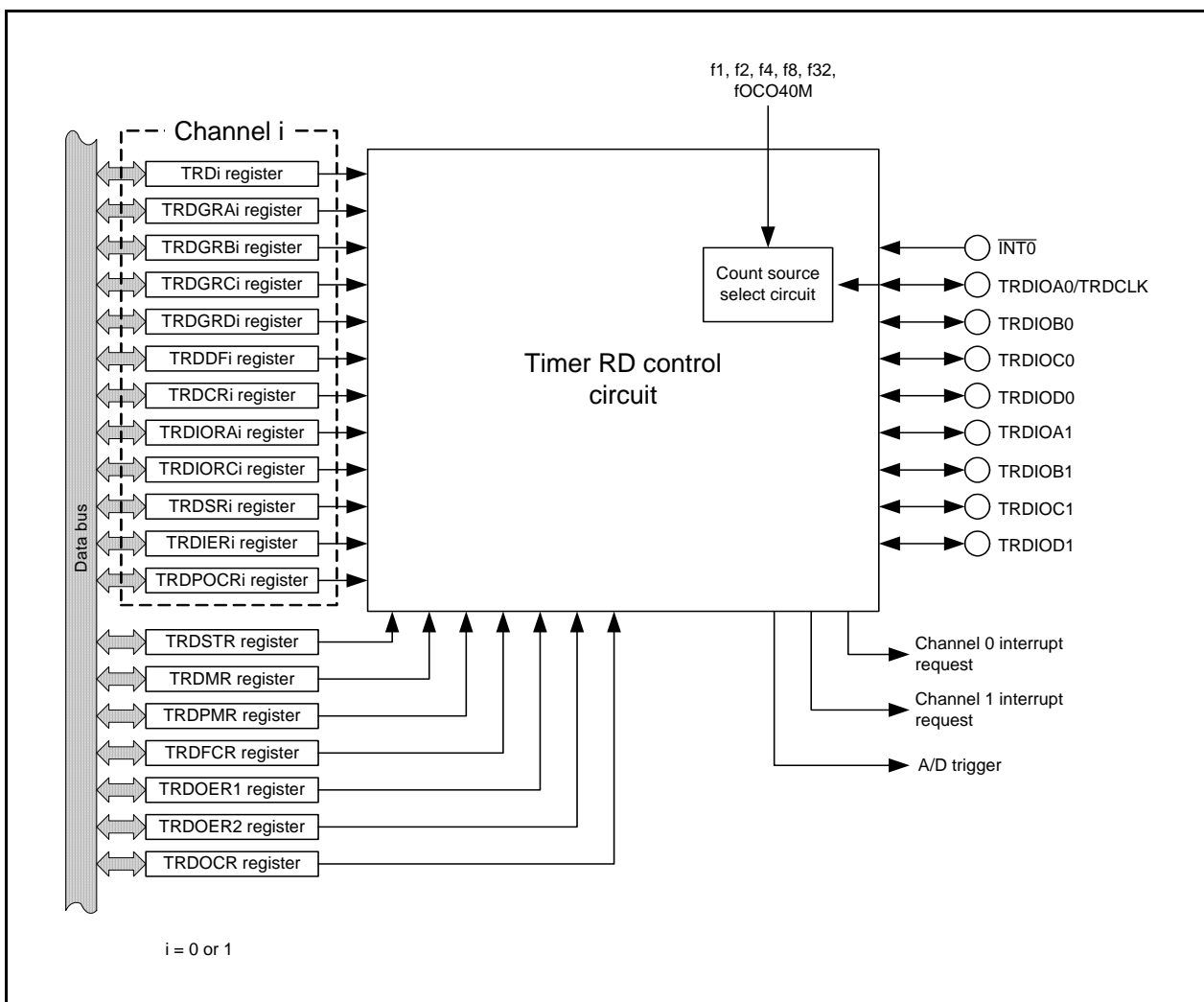


Figure 14.27 Block Diagram of Timer RD

### 14.3.1 Count Source

The count source selection can be used in all modes. However, in PWM3 mode, the external clock cannot be selected.

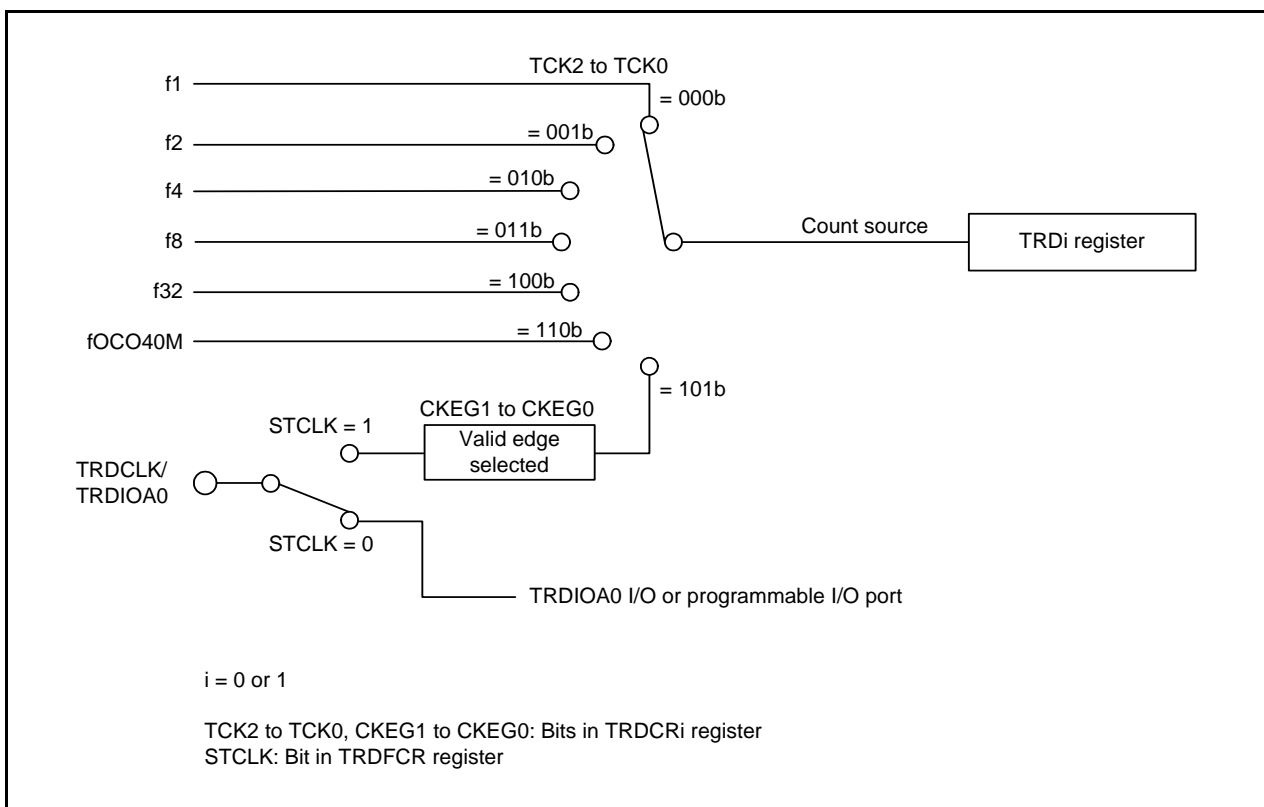
**Table 14.21 Count Source Selection**

Count Source	Selection
f1, f2, f4, f8, f32	The count source is selected by bits TCK2 to TCK0 in the TRDCR <sub>i</sub> register.
fOCO40M <sup>(1)</sup>	The FRA00 bit in the FRA0 register is set to 1 (high-speed on-chip oscillator frequency). Bits TCK2 to TCK0 in the TRDCR <sub>i</sub> register is set to 110b (fOCO40M).
External Signal Input to TRDCLK Pin	The STCLK bit in the TRDFCR register is set to 1 (external clock input enabled). The TCK2 to TCK0 bits in the TRDCR <sub>i</sub> register are set to 101b (count source: external clock). The valid edge selected by the CKEG1 to CKEG0 bits in the TRDCR <sub>i</sub> register. The PD2_0 bit in the PD2 register is set to 0 (input mode).

i = 0 or 1

**NOTE:**

- The count source fOCO40M can be used with VCC = 3.0 to 5.5 V.



**Figure 14.28 Block Diagram of Count Source**

Set the pulse width of the external clock which inputs to the TRDCLK pin to 3 cycles or above of the operation clock of Timer RD (refer to **Table 14.11 Timer RD Operation Clocks**).

When selecting fOCO40M for the count source, set the FRA00 bit in the FRA0 register to 1 (high-speed on-chip oscillator on) before setting the TCK2 to TCK0 bits in the TRDCR<sub>i</sub> register (i = 0 or 1) to 110b (fOCO40M).

### 14.3.2 Buffer Operation

The TRDGRC<sub>i</sub> register can be used as the buffer register of the TRDGRA<sub>i</sub> register, and the TRDGRD<sub>i</sub> register can be used as the buffer register of the TRDGRB<sub>i</sub> register by the BFC<sub>i</sub> and BFD<sub>i</sub> bits in the TRDMR register.

- TRDGRA<sub>i</sub> buffer register: TRDGRC<sub>i</sub> register
- TRDGRB<sub>i</sub> buffer register: TRDGRD<sub>i</sub> register

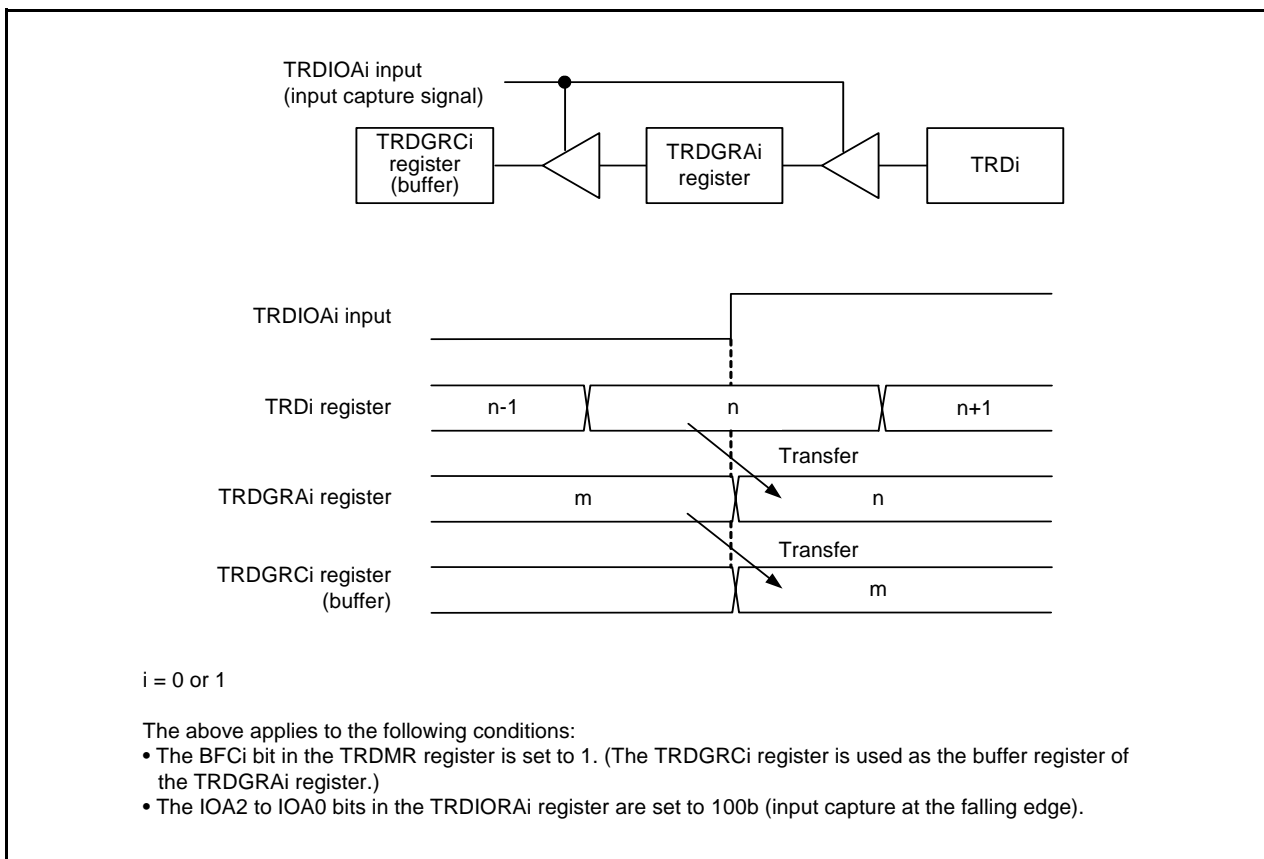
Buffer operation depends on modes. Table 14.22 lists the Buffer Operation in Each Mode.

Figure 14.29 shows the Buffer Operation in Input Capture Function, and Figure 14.30 shows the Buffer Operation in Output Compare Function.

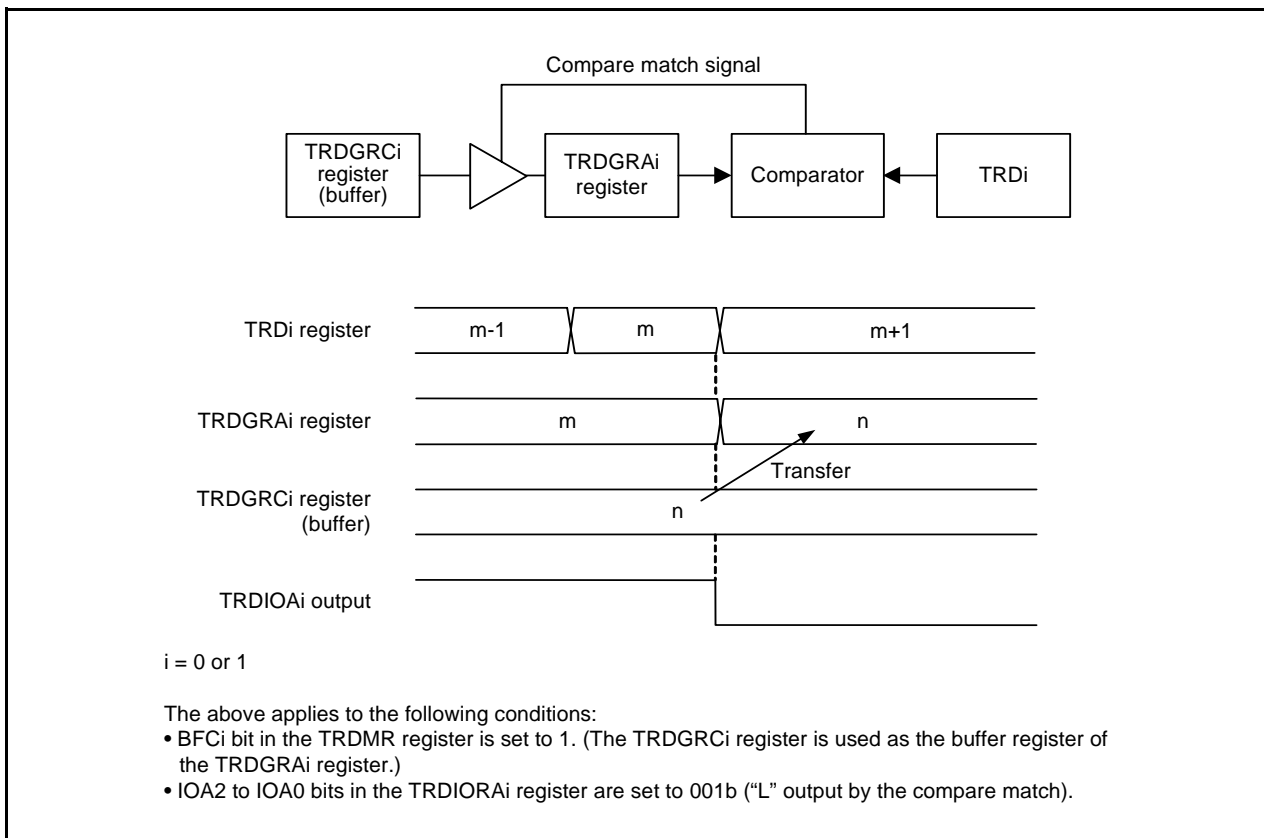
**Table 14.22 Buffer Operation in Each Mode**

Function and Mode	Transfer Timing	Transfer Register
Input Capture Function	Input capture signal input	Transfer content in TRDGRA <sub>i</sub> (TRDGRB <sub>i</sub> ) register to buffer register
Output Compare Function	Compare match with TRD <sub>i</sub> register and TRDGRA <sub>i</sub> (TRDGRB <sub>i</sub> ) register	Transfer content in buffer register to TRDGRA <sub>i</sub> (TRDGRB <sub>i</sub> ) register
PWM Mode	Compare match with TRD <sub>0</sub> register and TRDGRA <sub>0</sub> register	Transfer content in buffer register to TRDGRA <sub>i</sub> (TRDGRB <sub>i</sub> ) register
Reset Synchronous PWM Mode	Compare match with TRD <sub>0</sub> register and TRDGRA <sub>0</sub> register	Transfer content in buffer register to TRDGRA <sub>i</sub> (TRDGRB <sub>i</sub> ) register
Complementary PWM Mode	<ul style="list-style-type: none"> <li>• Compare match with TRD<sub>0</sub> register and TRDGRA<sub>0</sub> register</li> <li>• TRD<sub>1</sub> register underflow</li> </ul>	Transfer content in buffer register to TRDGRB <sub>0</sub> , TRDGRA <sub>1</sub> and TRDGRB <sub>1</sub> registers
PWM3 Mode	Compare match with TRD <sub>0</sub> register and TRDGRA <sub>0</sub> register	Transfer content in buffer register to TRDGRA <sub>0</sub> , TRDGRB <sub>0</sub> , TRDGRA <sub>1</sub> and TRDGRB <sub>1</sub> registers

$i = 0$  or  $1$



**Figure 14.29 Buffer Operation in Input Capture Function**



**Figure 14.30 Buffer Operation in Output Compare Function**

Perform the following for the timer mode (input capture and output compare functions).

When using the TRDGRCi ( $i = 0 \text{ or } 1$ ) register as the buffer register of the TRDGRAi register

- Set the IOC3 bit in the TRDIORCi register to 1 (general register or buffer register).
- Set the IOC2 bit in the TRDIORCi register to the same as the IOA2 bit in the TRDIORAi register.

When using the TRDGRDi register as the buffer register of the TRDGRBi register

- Set the IOD3 bit in the TRDIORDi register to 1 (general register or buffer register).
- Set the IOD2 bit in the TRDIORCi register to the same value as the IOB2 bit in the TRDIORAi register.

Bits IMFC and IMFD in the TRDSRi register are set to 1 at the input edge of the TRDIOCi pin when also using registers TRDGRCi and TRDGRDi as the buffer register in the input capture function.

When using the TRDGRCi and TRDGRDi registers for the buffer register in output compare function, reset synchronous PWM mode, complementary PWM mode and PWM3 mode, the IMFC and IMFD bits in the TRDSRi register are set to 1 by the compare match with the TRDi register.

### 14.3.3 Synchronous Operation

The TRD1 register is synchronized with the TRD0 register.

- Synchronous preset

When the SYNC bit in the TRDMR register is set to 1 (synchronous operation), the data is written to both the TRD0 and TRD1 registers after writing to the TRDi register.

- Synchronous clear

When the SYNC bit in the TRDMR register is set to 1 and the CCLR2 to CCLR0 bits in the TRDCRi register are set to 011b (synchronous clear), and the TRD0 register is set to 0000h at the same time as the TRD1 register is set to 0000h.

Also, when the SYNC bit in the TRDMR register is set to 1 and the CCLR2 to CCLR0 bits in the TRDCRi register are set to 011b (synchronous clear), and the TRD1 register is set to 0000h at the same time as the TRD0 register is set to 0000h.

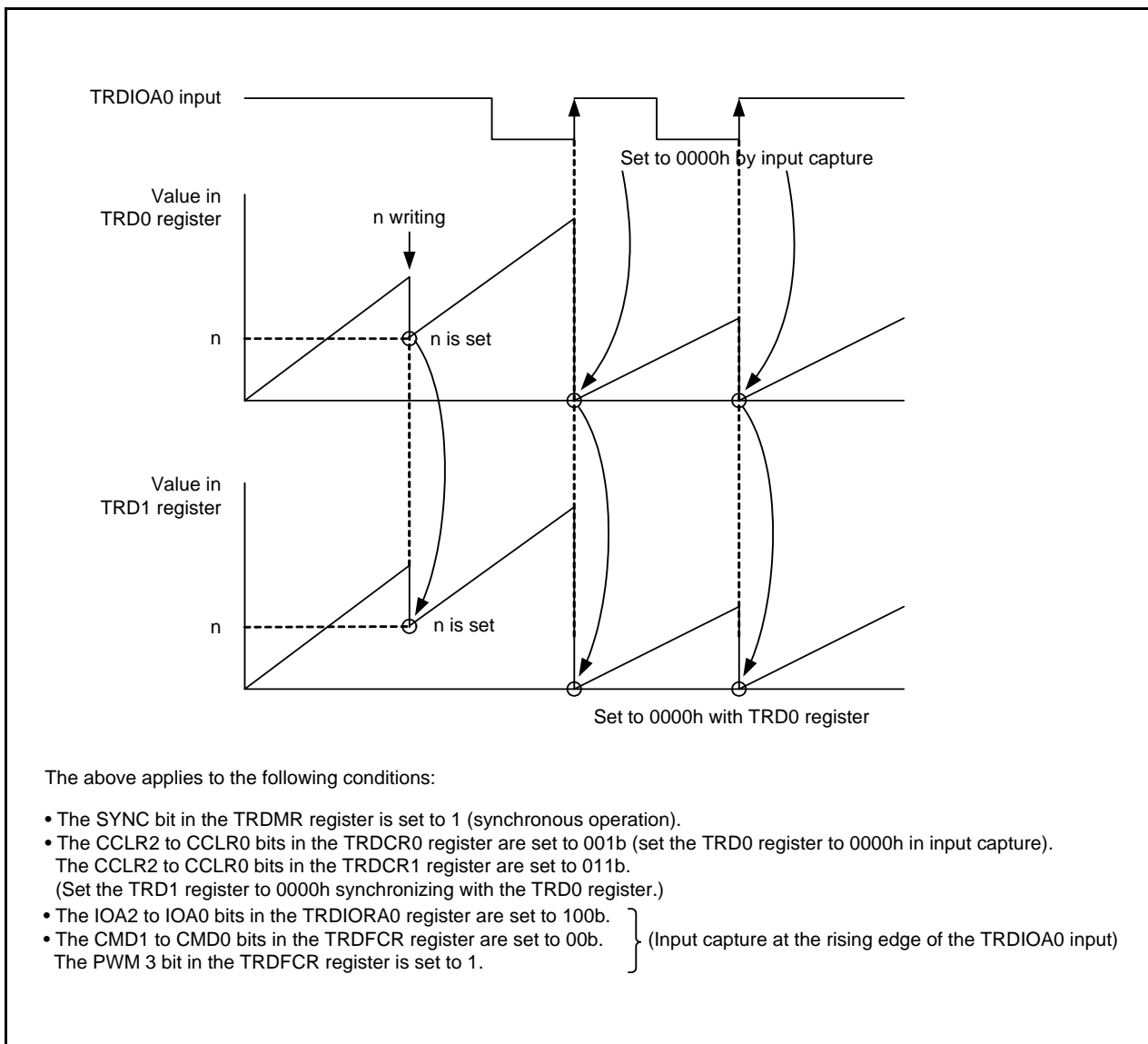


Figure 14.31 Synchronous Operation

### 14.3.4 Pulse Output Forced Cutoff

In the output compare function, PWM mode, reset synchronous PWM mode, complementary PWM mode and PWM3 mode, the TRDIO<sub>ji</sub> output pin can be forcibly set to the programmable I/O port by the  $\overline{\text{INT0}}$  pin input, and pulse output can be cut off.

The used pins for the output in these functions or modes can function as the output pin of Timer RD when setting the applicable bit in the TRDOER1 register to 0 (enable Timer RD output). When the PTO bit in the TRDOER2 register to 1 ( $\overline{\text{INT0}}$  of pulse output forced cutoff signal input enabled), all bits in the TRDOER1 register are set to 1 (disable Timer RD output, the TRDIO<sub>ji</sub> output pin is used as the programmable I/O port) after “L” is applied to the  $\overline{\text{INT0}}$  pin. The TRDIO<sub>ji</sub> output pin is set to the programmable I/O port after “L” is applied to the  $\overline{\text{INT0}}$  pin and waiting for 1 to 2 cycles of the Timer RD operation clock (refer to **Table 14.11 Timer RD Operation Clocks**).

Set as below when using this function:

- Set the pin status (high impedance, “L” or “H” output) with the pulse output forced cutoff by the P2 and PD2 registers.
- Set the INT0EN bit in the INTEN register to 1 (enable  $\overline{\text{INT0}}$  input) and the INT0PL bit to 0 (one edge).
- Set the PD4\_5 bit in the PD4 register to 0 (input mode).
- Set the  $\overline{\text{INT0}}$  digital filter by the INT0F1 to INT0F0 bits in the INTF register.
- Set the PTO bit in the TRDOER2 register to 1 (enable pulse output forced cutoff signal input  $\overline{\text{INT0}}$ ).

According to the selection of the POL bit in the INT0IC register and change of the  $\overline{\text{INT0}}$  pin input, the IR bit in the INT0IC register is set to 1 (interrupt request). Refer to **12. Interrupts** for details of interrupts.



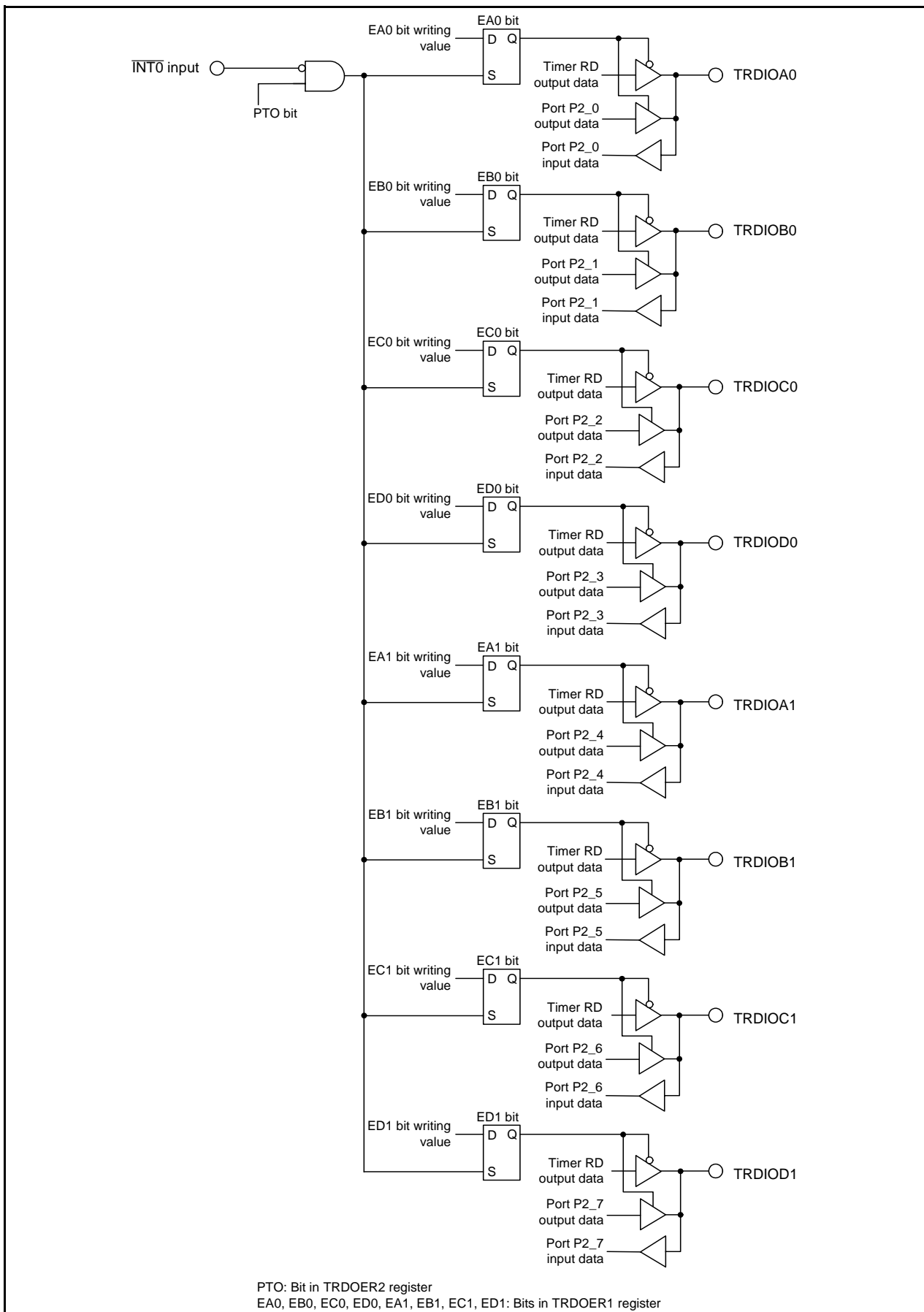


Figure 14.32 Pulse Output Forced Cutoff

### 14.3.5 Input Capture Function

The input capture function is to measure the external signal width and period. The content in the TRDi register (counter) is transferred to the TRDGRji register as a trigger of the TRDIOji ( $i = 0$  or  $1$ ,  $j =$  either A, B, C or D) pin external signal (input capture). Since this function is enabled with a combination of the TRDIOji pin and TRDGRji register, any of the input capture function, other modes or functions can be selected every pin.

The TRDGRA0 register can also select fOCO128 signal as input-capture trigger input.

Figure 14.33 shows the Block Diagram of Input Capture Function, Table 14.23 lists the Input Capture Function Specifications. Figures 14.34 to 14.44 show the Registers Associated with Input Capture Function and Figure 14.45 shows the Operating Example of Input Capture Function.

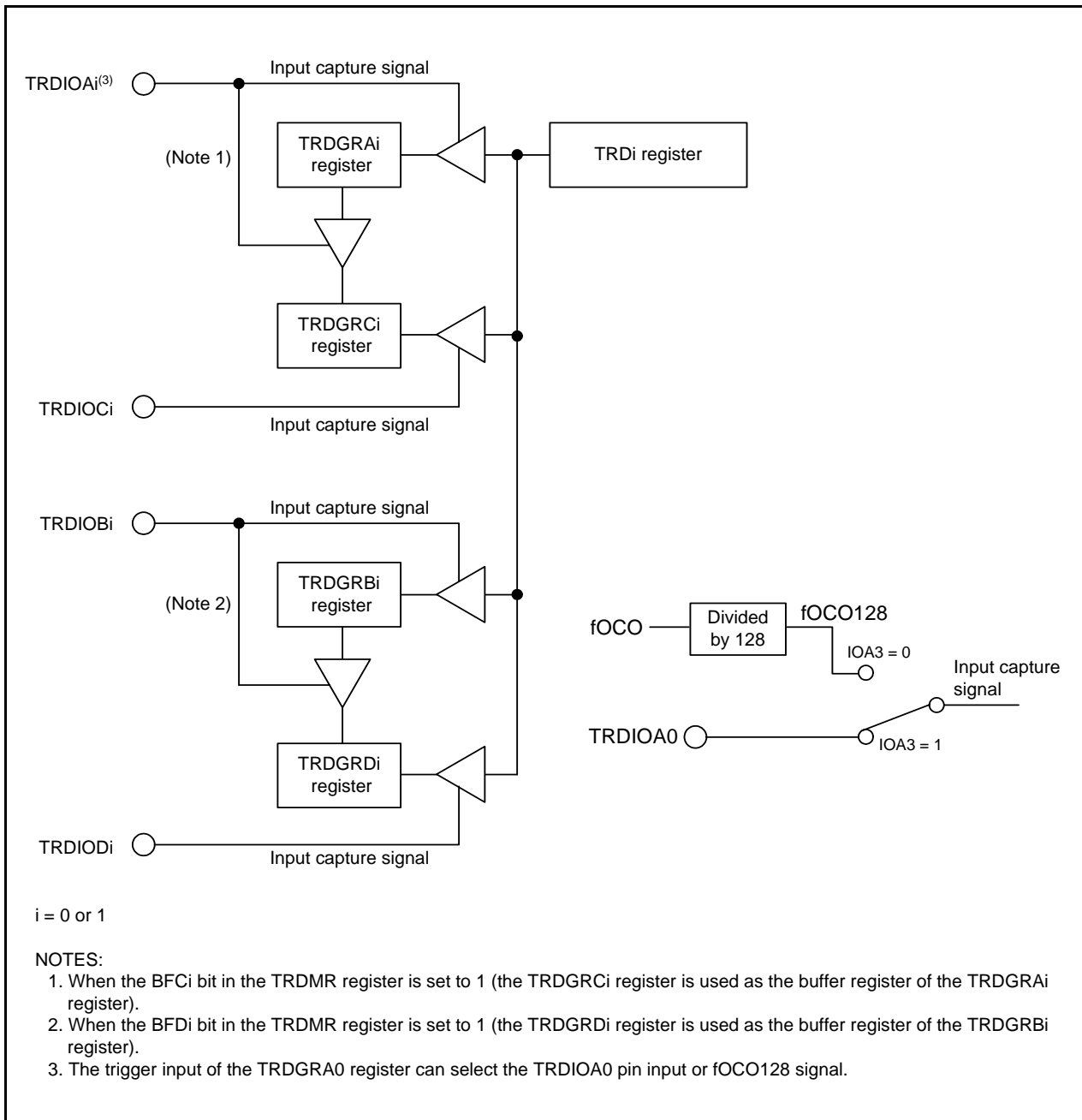


Figure 14.33 Block Diagram of Input Capture Function

**Table 14.23 Input Capture Function Specifications**

Item	Specification
Count Sources	f1, f2, f4, f8, f32, fOCO40M External signal input to the TRDCLK pin (valid edge selected by a program)
Count Operations	Increment
Count Period	When the CCLR2 to CCLR0 bits in the TRDCRi register are set to 000b (free-running operation). $1/fk \times 65536$ fk: Frequency of count source
Count Start Condition	Write 1 (count starts) to the TSTARTi bit in the TRDSTR register.
Count Stop Condition	Write 0 (count stops) to the TSTARTi bit in the TRDSTR register when the CSELi bit in the TRDSTR register is set to 1.
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>Input capture (valid edge of TRDIOji input or fOCO128 signal edge)</li> <li>TRDi register overflows</li> </ul>
TRDIOA0 Pin Function	Programmable I/O port, input-capture input, or TRDCLK (external clock) input
TRDIOB0, TRDIOC0, TRDIOD0, TRDIOA1 to TRDIOD1 Pin Functions	Programmable I/O port, or input-capture input (Select every pin)
INT0 Pin Function	Programmable I/O port or INT0 interrupt input
Read from Timer	The count value can be read by reading the TRDi register.
Write to Timer	<ul style="list-style-type: none"> <li>When the SYNC bit in the TRDMR register is set to 0 (channels 0 and 1 operate independently).</li> <li>Data can be written to the TRDi register.</li> <li>When the SYNC bit in the TRDMR register is set to 1 (channels 0 and 1 operate synchronously.)</li> <li>Data can be written to both the TRD0 and TRD1 registers by writing to the TRDi register.</li> </ul>
Selection Functions	<ul style="list-style-type: none"> <li>Input-capture input pin selected Either 1 pin or multiple pins of the TRDIOAi, TRDIOBi, TRDIOCi or TRDIODi pin.</li> <li>Input-capture input valid edge selected The rising edge, falling edge or both the rising and falling edges</li> <li>The timing when the TRDi register is set to 0000h At overflow or input capture</li> <li>Buffer operation (refer to <b>14.3.2 Buffer Operation</b>)</li> <li>Synchronous operation (refer to <b>14.3.3 Synchronous Operation</b>)</li> <li>Digital filter The TRDIOji input is sampled, and when the sampled input level match 3 times, its level is assumed as a determination.</li> <li>Input-capture trigger selected fOCO128 can be selected for input-capture trigger input of the TRDGRA0 register.</li> </ul>

i = 0 or 1, j = either A, B, C or D

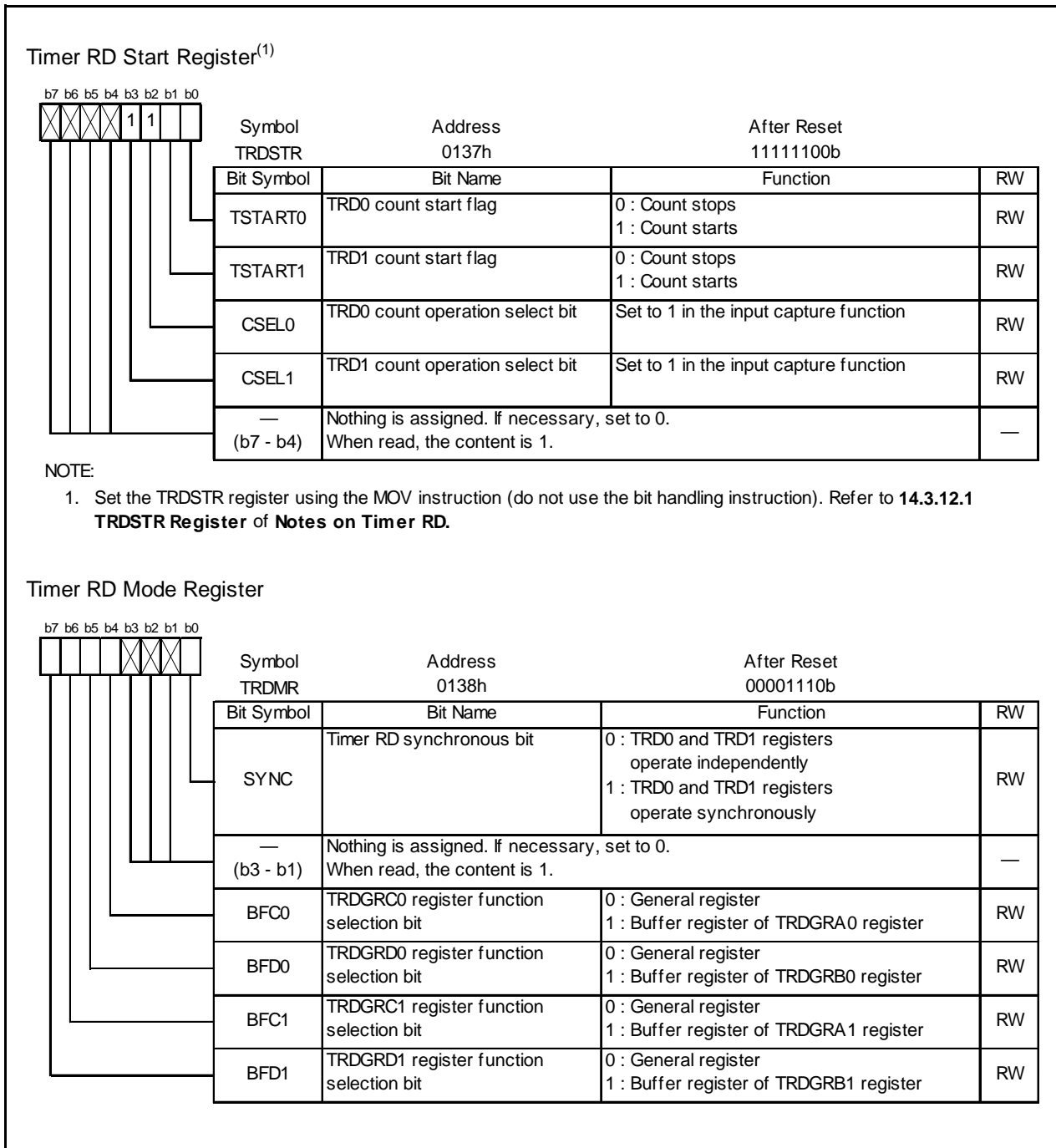


Figure 14.34 Registers TRDSTR and TRDMR in Input Capture Function

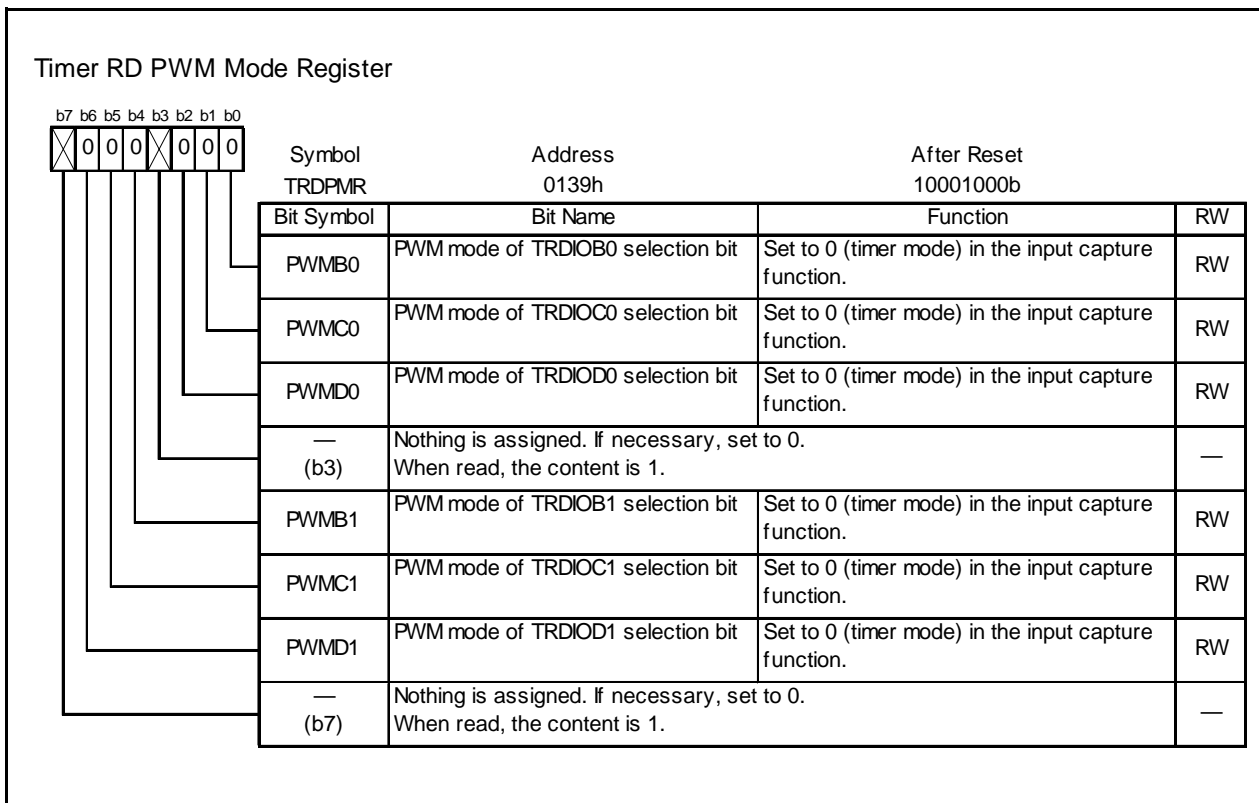


Figure 14.35 TRDPMR Register in Input Capture Function

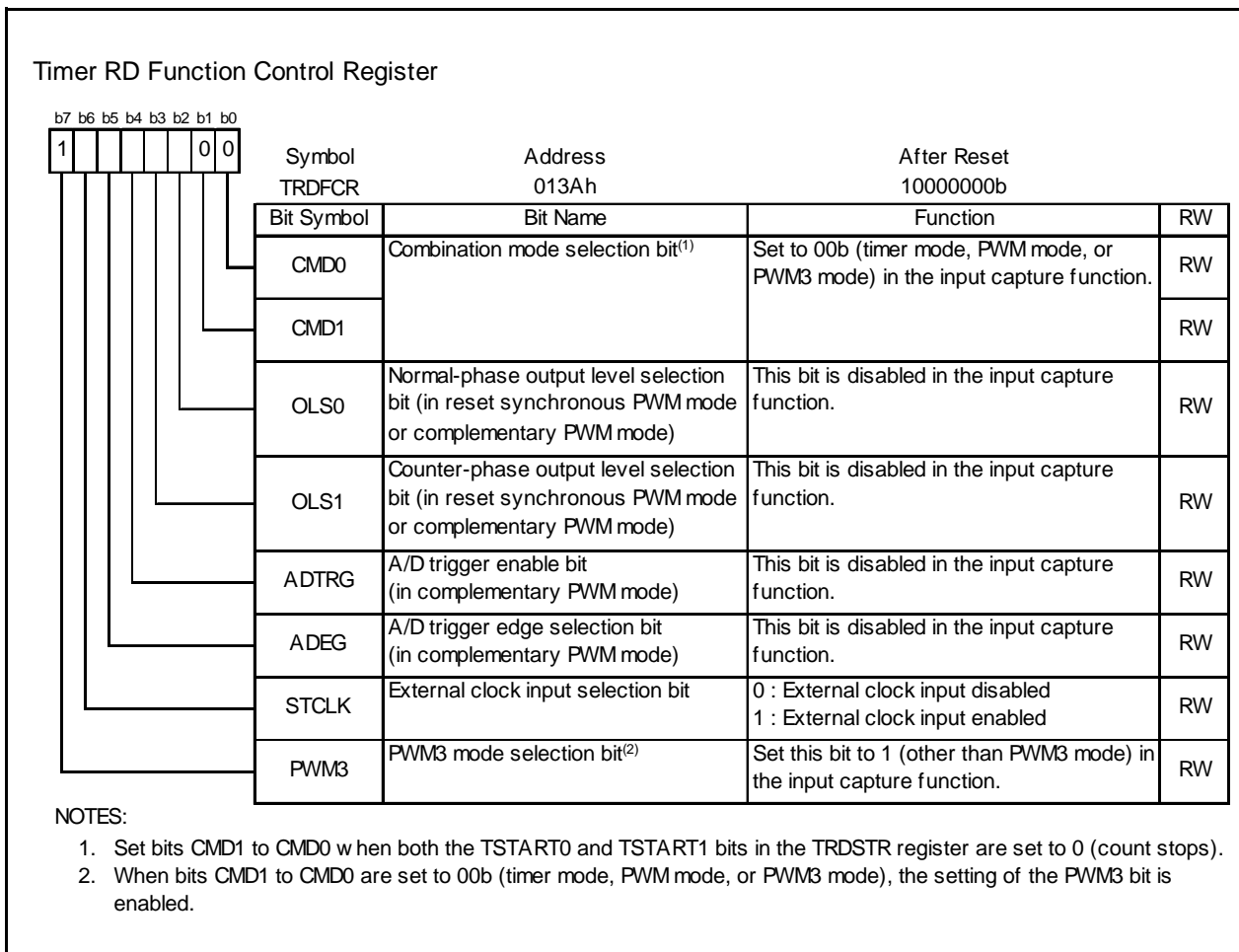


Figure 14.36 TRDFCR Register in Input Capture Function

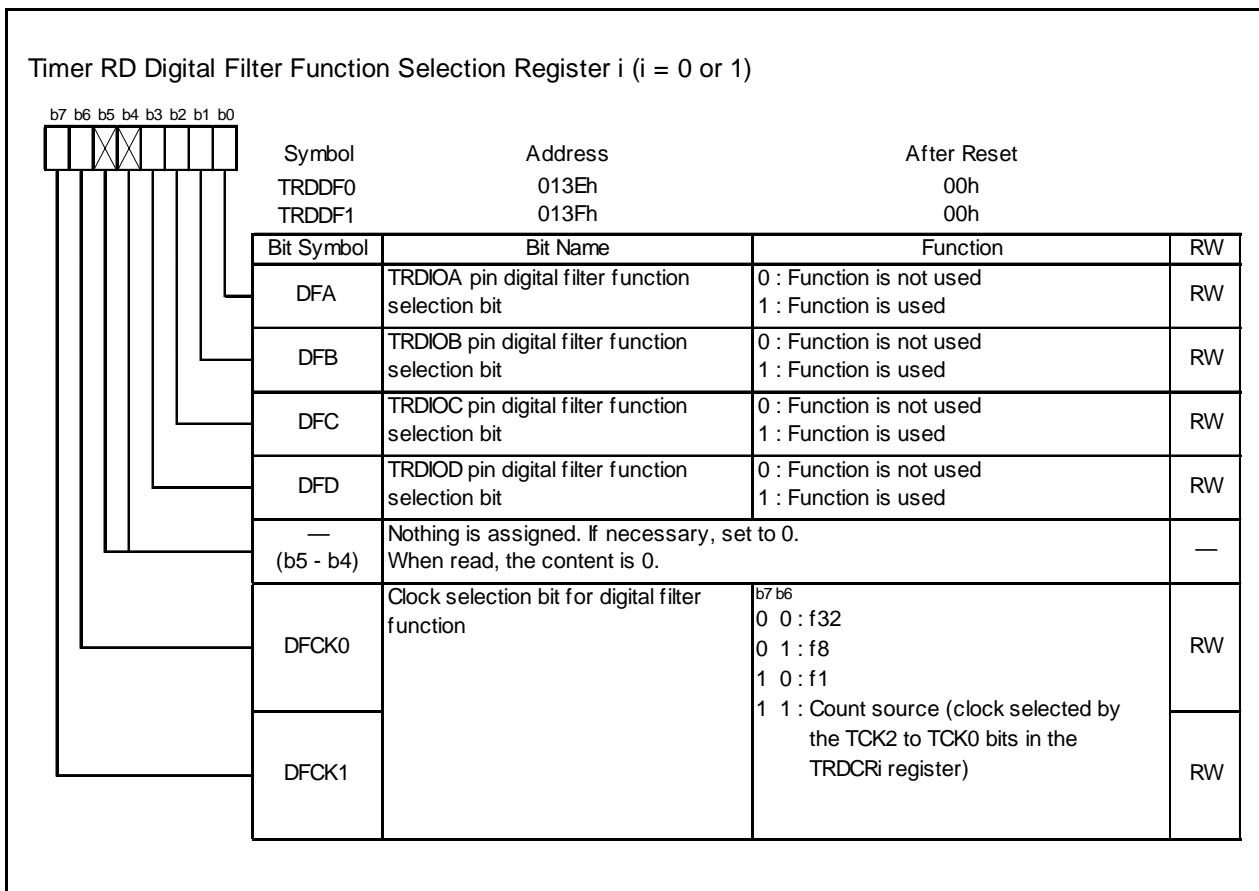


Figure 14.37 Registers TRDDF0 to TRDDF1 in Input Capture Function

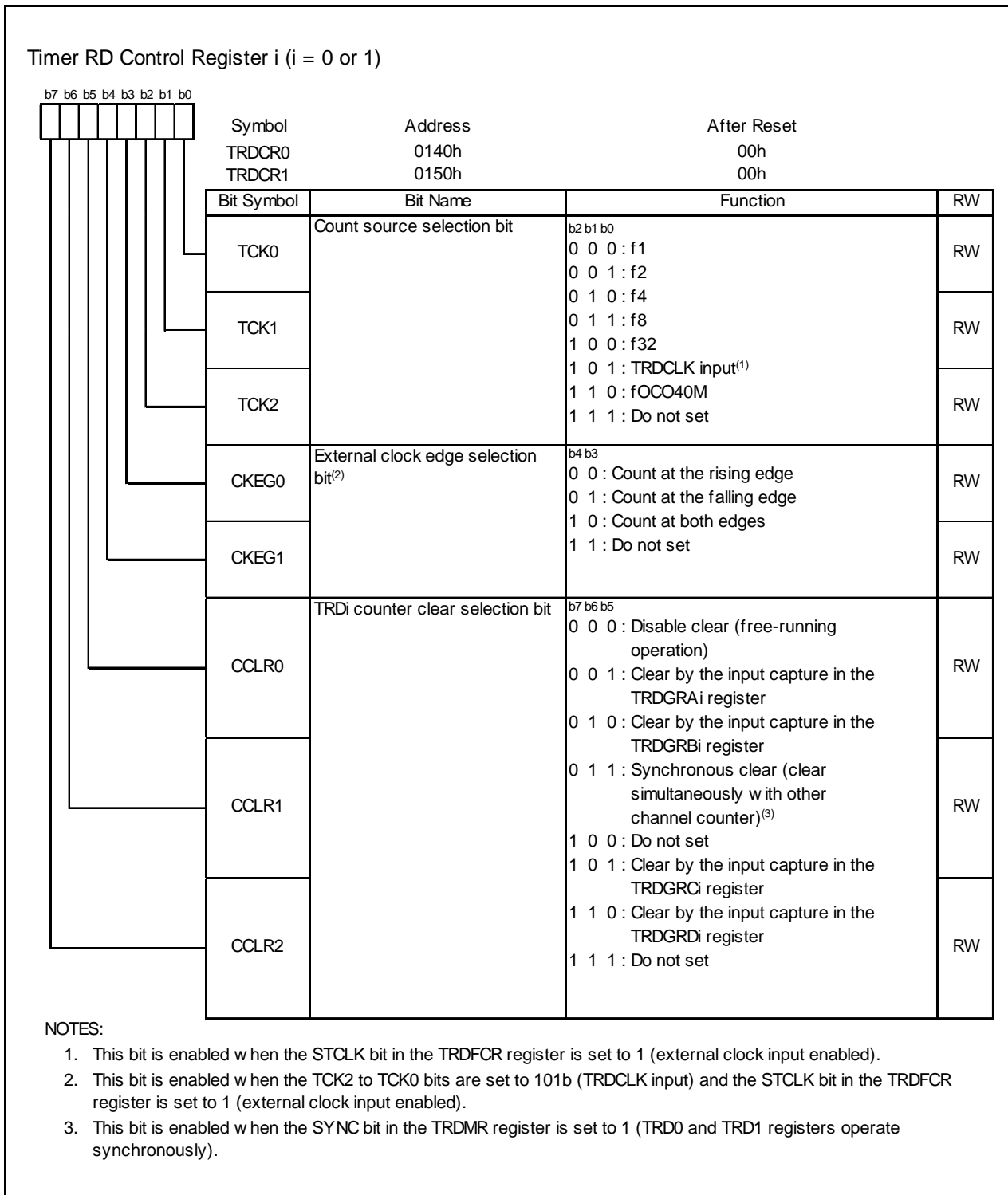


Figure 14.38 Registers TRDCR0 to TRDCR1 in Input Capture Function



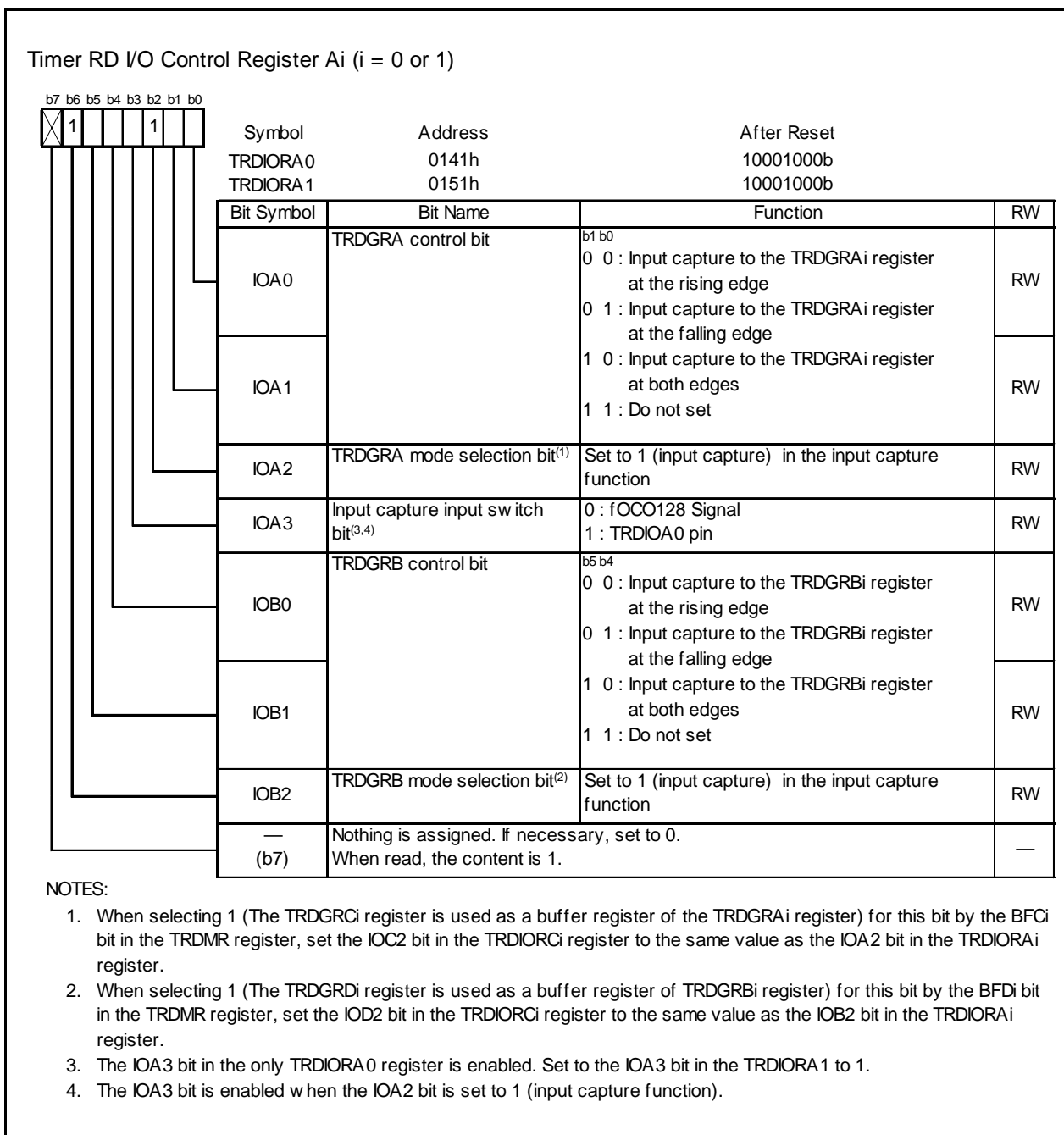


Figure 14.39 Registers TRDIORA0 to TRDIORA1 in Input Capture Function

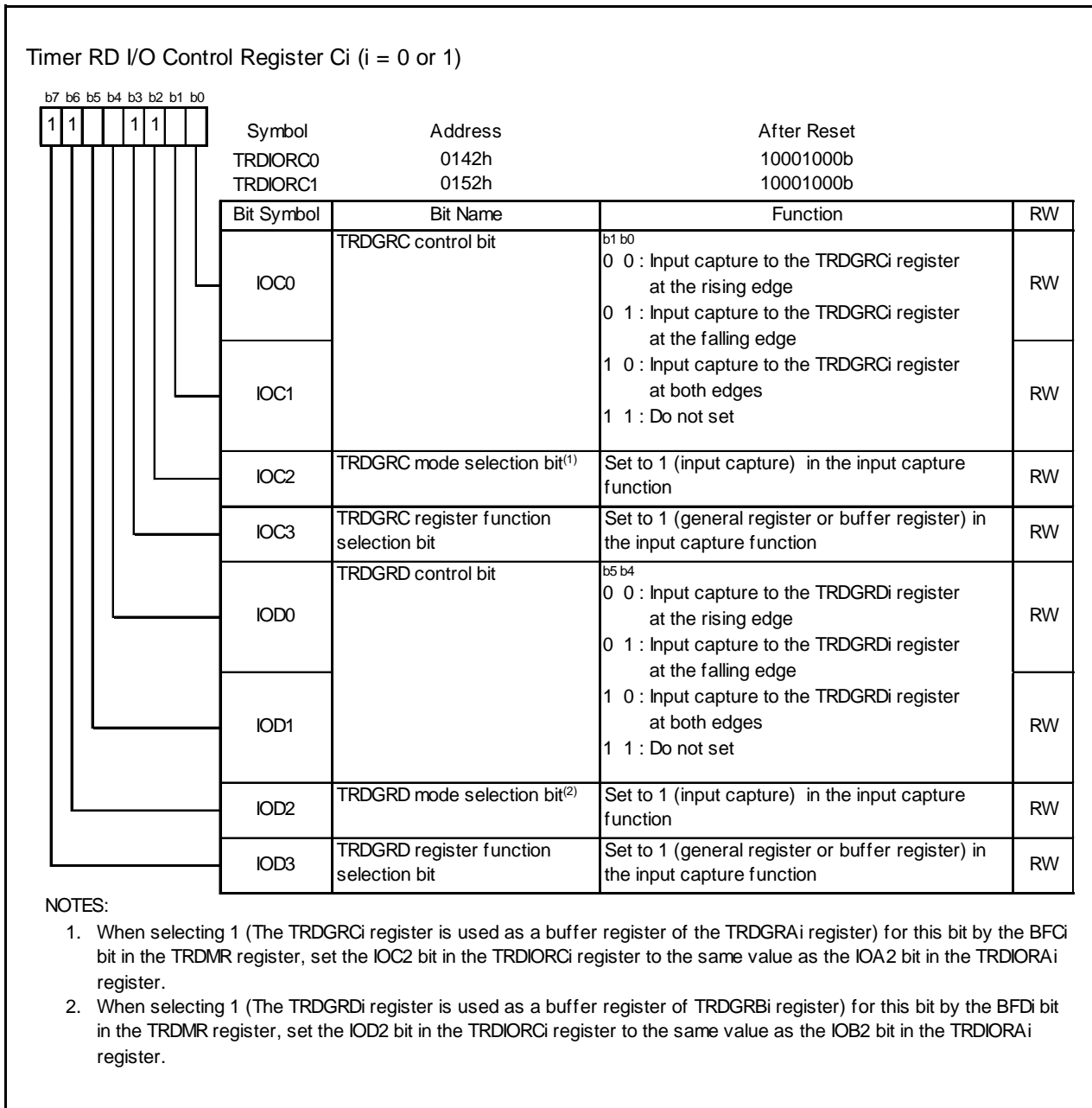


Figure 14.40 Registers TRDIORC0 to TRDIORC1 in Input Capture Function

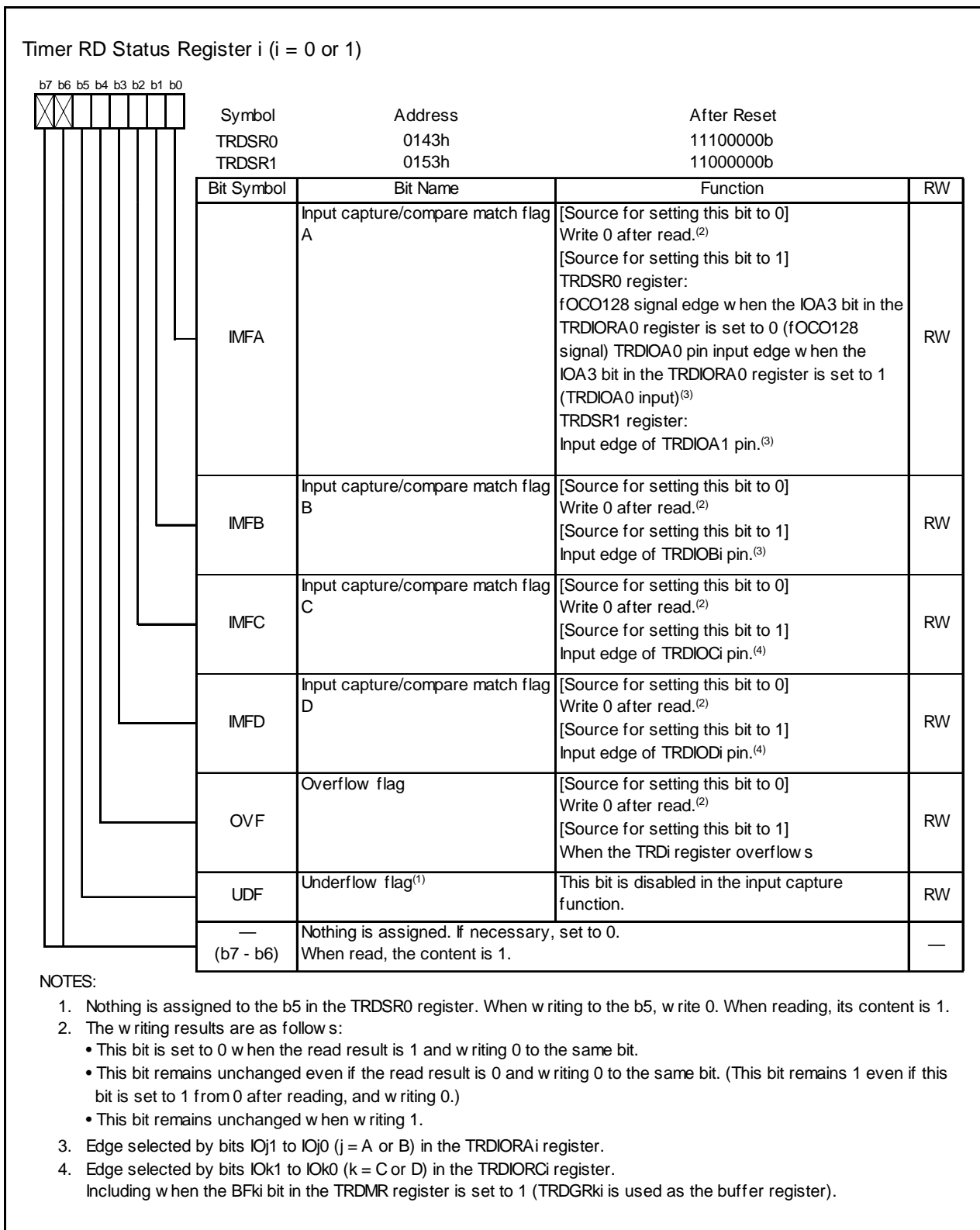


Figure 14.41 Registers TRDSR0 to TRDSR1 in Input Capture Function

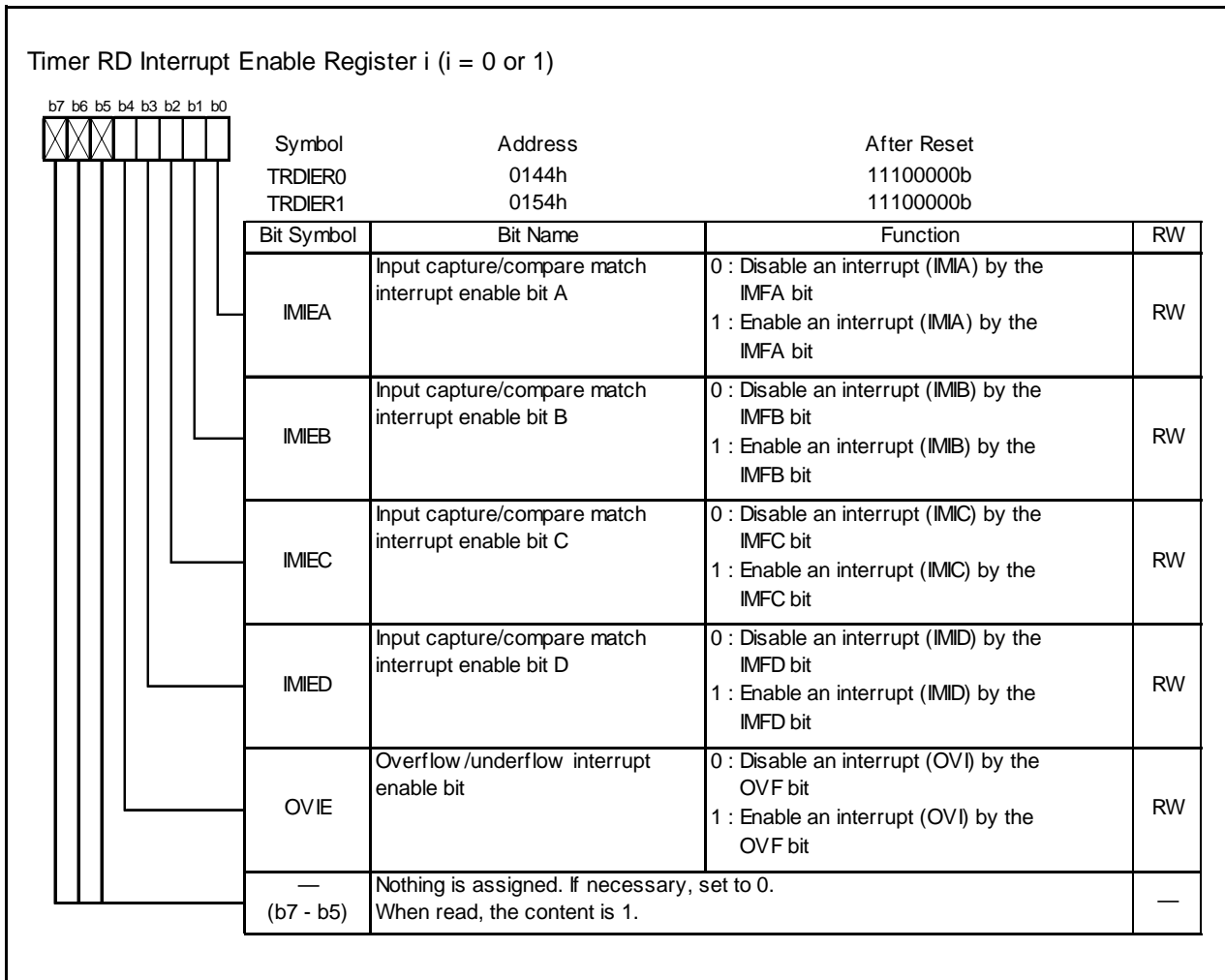


Figure 14.42 Registers TRDIER0 to TRDIER1 in Input Capture Function

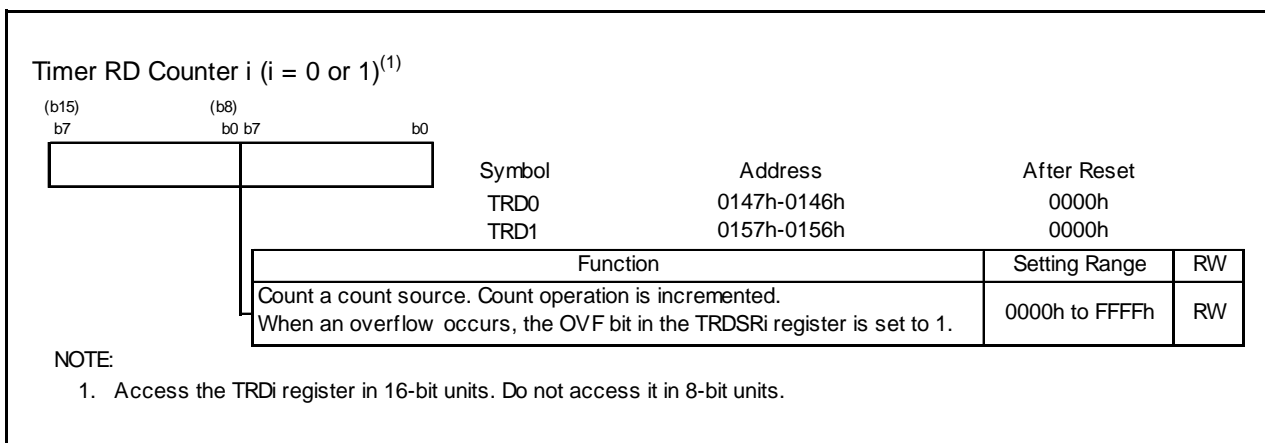


Figure 14.43 Registers TRD0 to TRD1 in Input Capture Function

Timer RD General Register Ai, Bi, Ci and Di (i = 0 or 1)<sup>(1)</sup>

(b15) b7	(b8) b0 b7			
		Symbol	Address	After Reset
		TRDGRA0	0149h-0148h	FFFFh
		TRDGRB0	014Bh-014Ah	FFFFh
		TRDGRC0	014Dh-014Ch	FFFFh
		TRDGRD0	014Fh-014Eh	FFFFh
		TRDGRA1	0159h-0158h	FFFFh
		TRDGRB1	015Bh-015Ah	FFFFh
		TRDGRC1	015Dh-015Ch	FFFFh
		TRDGRD1	015Fh-015Eh	FFFFh
Function				RW
Refer to Table 14.24 TRDGRji Register Functions in Input Capture Function				RW

NOTE:  
1. Access the TRDGRAi to TRDGRDi registers in 16-bit units. Do not access them in 8-bit units.

**Figure 14.44 Registers TRDGRAi, TRDGRBi, TRDGRCi and TRDGRDi in Input Capture Function**

The following registers are disabled in the input capture function:  
TRDOER1, TRDOER2, TRDOCR, TRDPOCR0 and TRDPOCR1

**Table 14.24 TRDGRji Register Functions in Input Capture Function**

Register	Setting	Register Function	Input-Capture Input Pin
TRDGRAi	–	General register The value in the TRDi register can be read at the input capture.	TRDIOAi
TRDGRBi			TRDIOBi
TRDGRCi	BFCi = 0	General register The value in the TRDi register can be read at the input capture.	TRDIOCi
TRDGRDi	BFDi = 0		TRDIODi
TRDGRCi	BFCi = 1	Buffer register The value in the TRDi register can be read at the input capture. (Refer to 14.3.2 Buffer Operation)	TRDIOAi
TRDGRDi	BFDi = 1		TRDIOBi

i = 0 or 1, j = either A, B, C or D  
BFCi, BFDi: Bits in TRDMR Register

Set the pulse width of the input capture signal applied to the TRDIOji pin to 3 cycles or more of the Timer RD operation clock (refer to Table 14.11 Timer RD Operation Clocks) for “no digital filter” (the DFj bit in the TRDDFi register is set to 0).

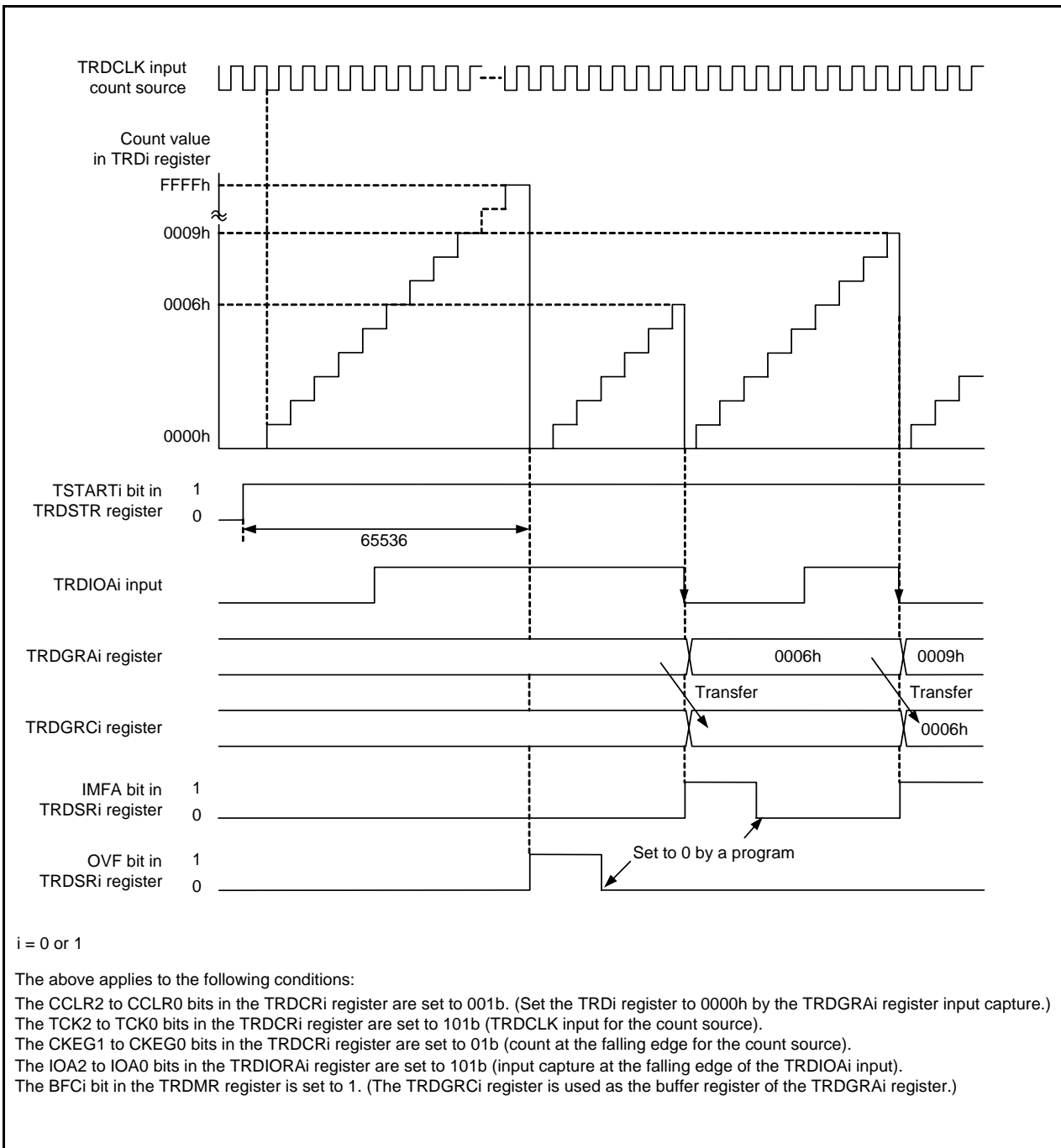


Figure 14.45 Operating Example of Input Capture Function

### 14.3.5.1 Digital Filter

The TRDIO<sub>j</sub> input is sampled, and when the sampled input level matches 3 times, its level is assumed as a determination. Select the digital filter function and sampling clock by the TRDDF<sub>i</sub> register.

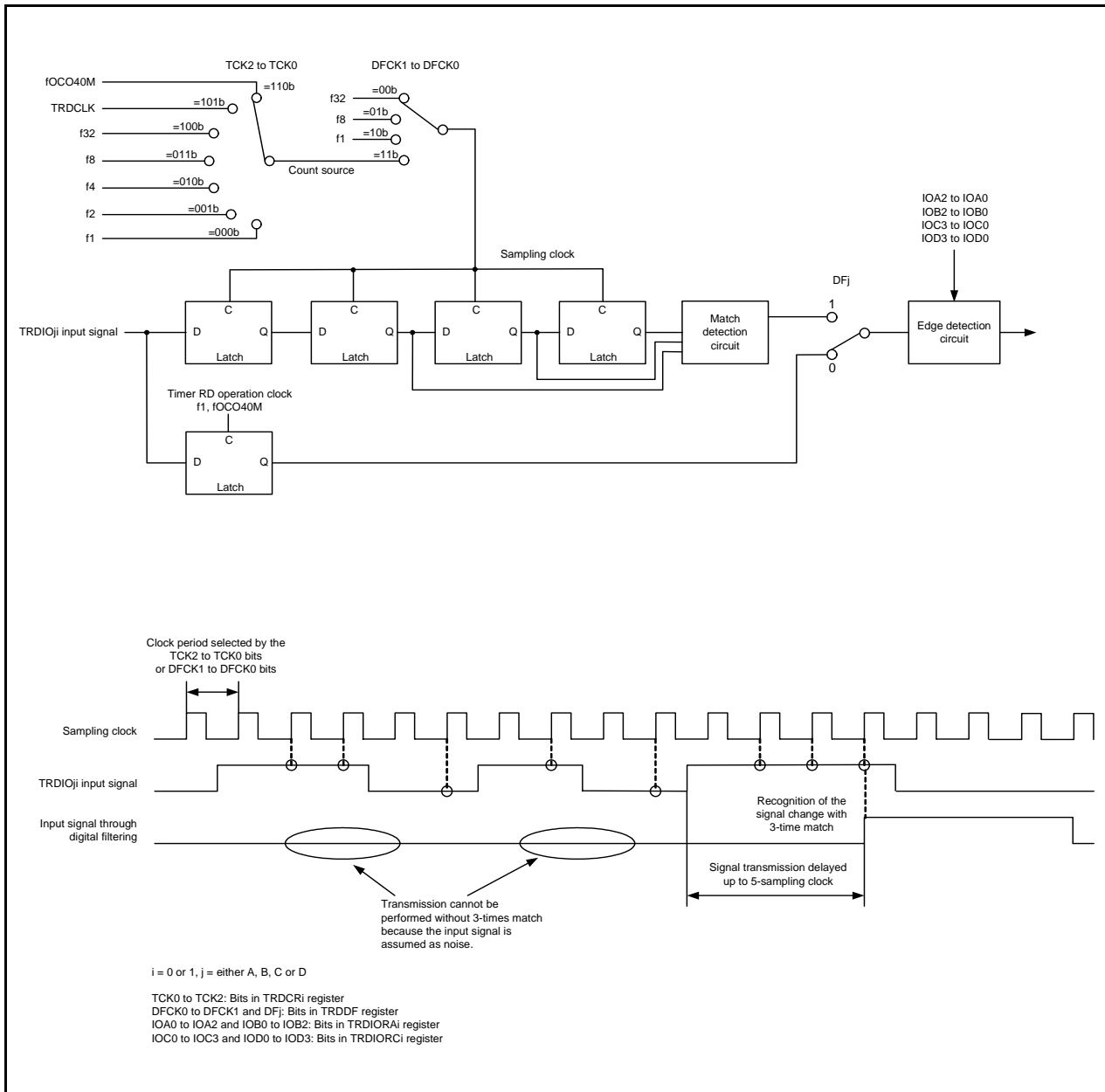


Figure 14.46 Block Diagram of Digital Filter

### 14.3.6 Output Compare Function

This function is to detect the match (compare match) of the content in the TRDGR<sub>j</sub> (j = either A, B, C and D) register with the content in the TRD<sub>i</sub> (i = 0 or 1) register. When the content matches, any level is output from the TRDIO<sub>j</sub> pin. Since this function is enabled with a combination of the TRDIO<sub>j</sub> pin and TRDGR<sub>j</sub> register, any of the output compare function, other modes or functions can be selected every pin.

Figure 14.47 shows the Block Diagram of Output Compare Function, Table 14.25 lists the Output Compare Function Specifications. Figures 14.48 to 14.59 list the Registers Associated with Output Compare Function and Figure 14.60 shows the Operating Example of Output Compare Function.

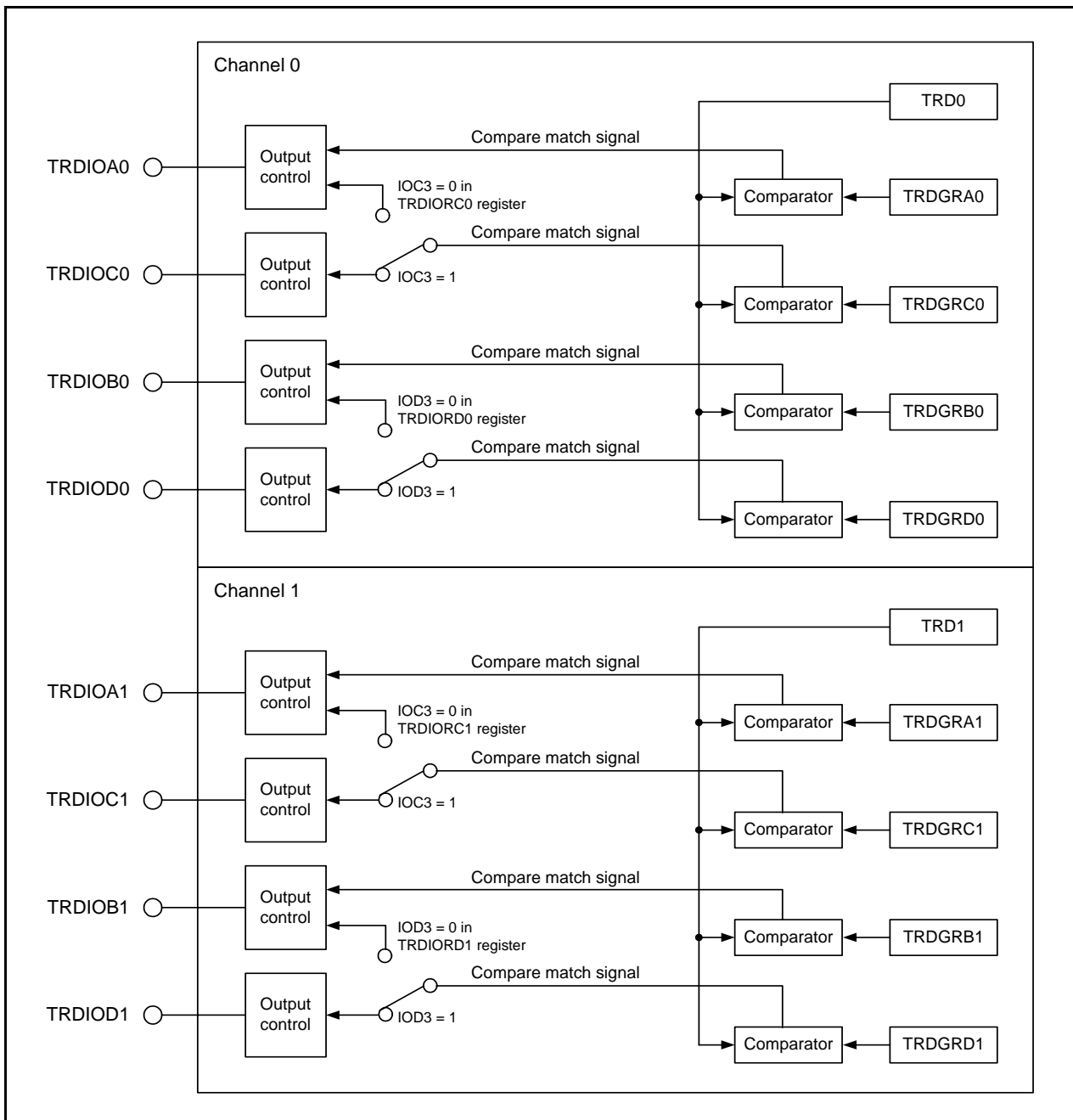


Figure 14.47 Block Diagram of Output Compare Function



**Table 14.25 Output Compare Function Specifications**

Item	Specification
Count Sources	f1, f2, f4, f8, f32, fOCO40M External signal input to the TRDCLK pin (valid edge selected by a program)
Count Operations	Increment
Count Period	<ul style="list-style-type: none"> <li>When the CCLR2 to CCLR0 bits in the TRDCRi register are set to 000b (free-running operation) 1/fk × 65536 fk: Frequency of count source</li> <li>The CCLR1 to CCLR0 bits in the TRDCRi register are set to 01b or 10b (set the TRDi register to 0000h at the compare match in the TRDGRji register.) Frequency of count source × (n + 1) n: Setting value in the TRDGRji register</li> </ul>
Waveform Output Timing	Compare match
Count Start Condition	Write 1 (count starts) to the TSTARTi bit in the TRDSTR register.
Count Stop Conditions	<ul style="list-style-type: none"> <li>Write 0 (count stops) to the TSTARTi bit in the TRDSTR register when the CSELi bit in the TRDSTR register is set to 1. The output compare output pin holds output level before the count stops.</li> <li>When the CSELi bit in the TRDSTR register is set to 0, the count stops at the compare match in the TRDGRAi register. The output compare output pin holds level after output change by the compare match.</li> </ul>
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>Compare match (the content in the TRDi register matches with the content in the TRDGRji register.)</li> <li>TRDi register overflows</li> </ul>
TRDIOA0 Pin Function	Programmable I/O port, output-compare output or TRDCLK (external clock) input
TRDIOB0, TRDIOC0, TRDIOD0, TRDIOA1 to TRDIOD1 Pin Functions	Programmable I/O port or output-compare output (select every pin)
INT0 Pin Function	Programmable I/O port, pulse output forced cutoff signal input or INT0 interrupt input
Read from Timer	The count value can be read by reading the TRDi register.
Write to Timer	<ul style="list-style-type: none"> <li>When the SYNC bit in the TRDMR register is set to 0 (channels 0 and 1 operate independently) Data can be written to the TRDi register.</li> <li>When the SYNC bit in the TRDMR register is set to 1 (channels 0 and 1 operate synchronously). Data can be written to both the TRD0 and TRD1 registers by writing to the TRDi register.</li> </ul>
Selection Functions	<ul style="list-style-type: none"> <li>Output-compare output pin selected Either 1 pin or multiple pins of the TRDIOAi, TRDIOBi, TRDIOCi or TRDIODi pin.</li> <li>Output level at the compare match selected “L” output, “H” output or output level inversed</li> <li>Initial output level selected Set the level at period from the count start to the compare match.</li> <li>Timing to set the TRDi register to 0000h Overflow or compare match in the TRDGRAi register</li> <li>Buffer operation (refer to <b>14.3.2 Buffer Operation</b>)</li> <li>Synchronous operation (refer to <b>14.3.3 Synchronous Operation</b>)</li> <li>Output pin in the TRDGRCi and TRDGRDi registers changed The TRDGRCi register can be used as output control of the TRDIOAi pin and the TRDGRDi register can be used as output control of the TRDIOBi pin.</li> <li>Pulse output forced cutoff signal input (refer to <b>14.3.4 Pulse Output Forced Cutoff</b>)</li> <li>Timer RD can be used as the internal timer without output.</li> </ul>

i = 0 or 1, j = either A, B, C or D

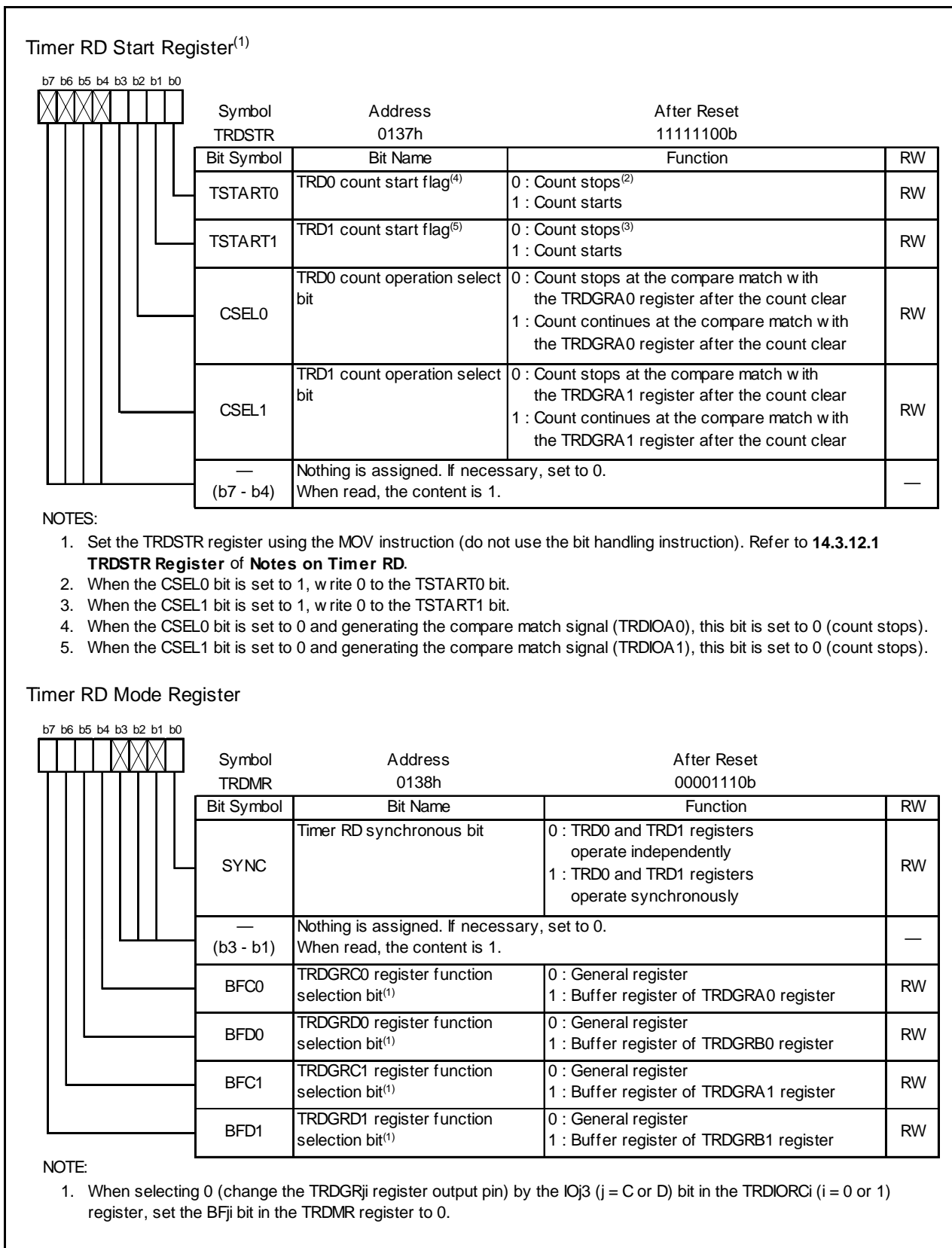


Figure 14.48 Registers TRDSTR and TRDMR in Output Compare Function

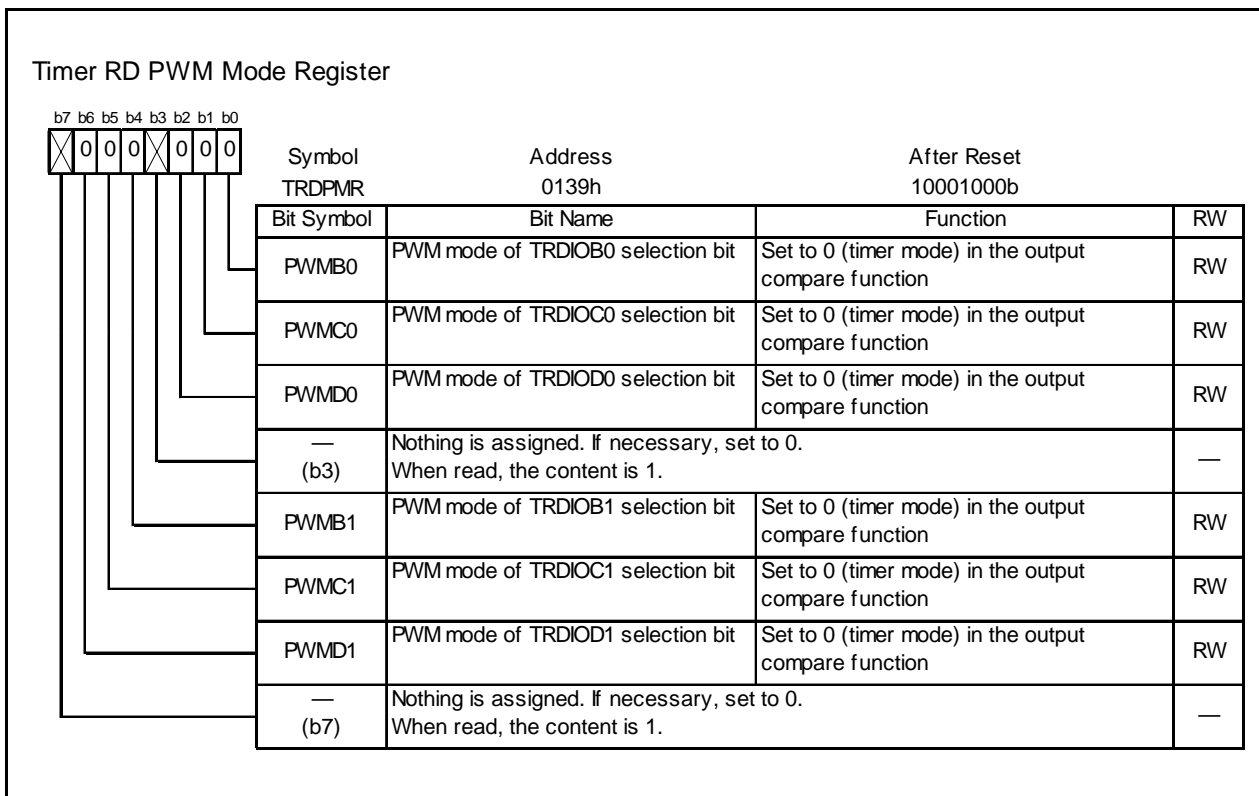


Figure 14.49 TRDPMR Register in Output Compare Function

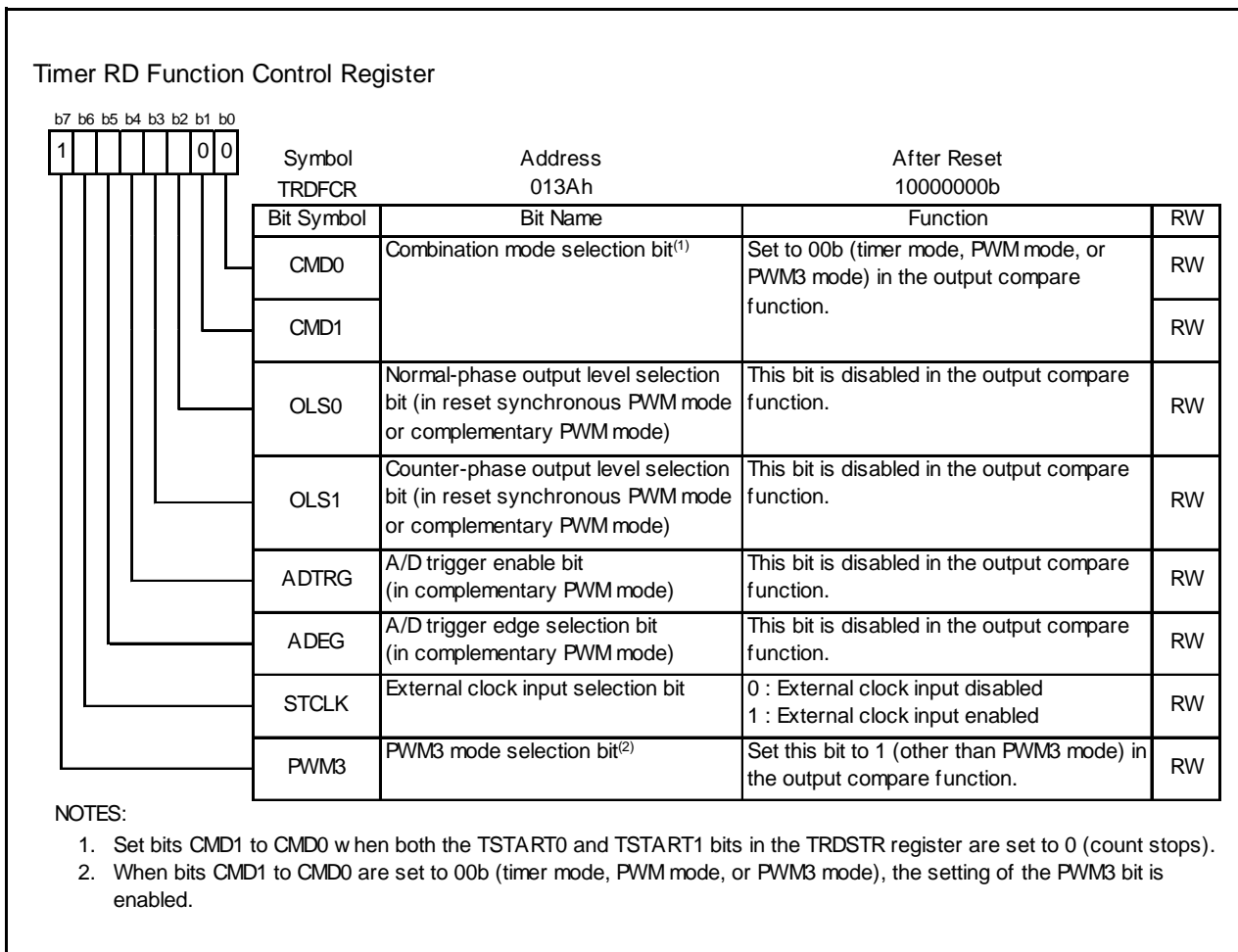


Figure 14.50 TRDFCR Register in Output Compare Function

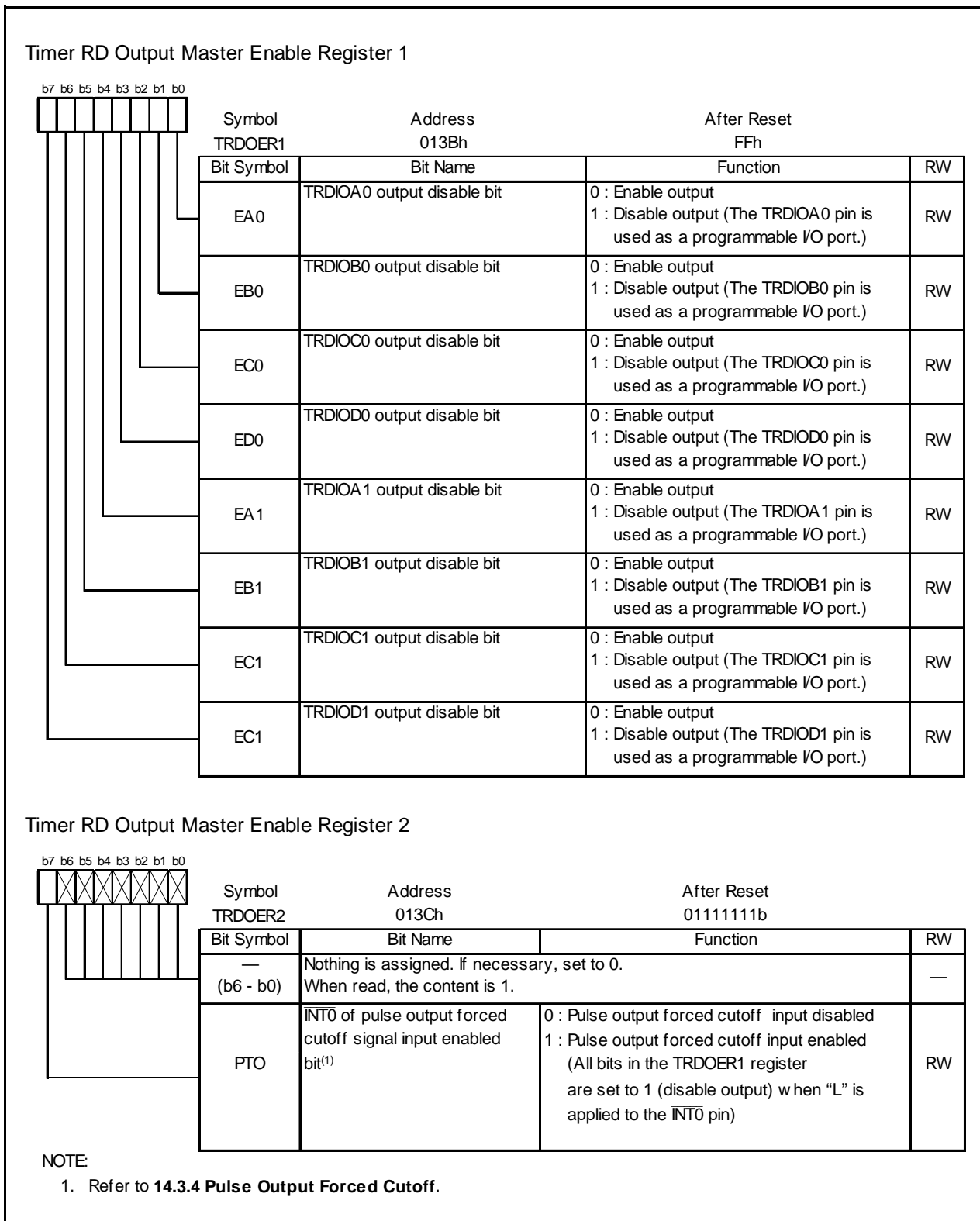


Figure 14.51 Registers TRDOER1 to TRDOER2 in Output Compare Function

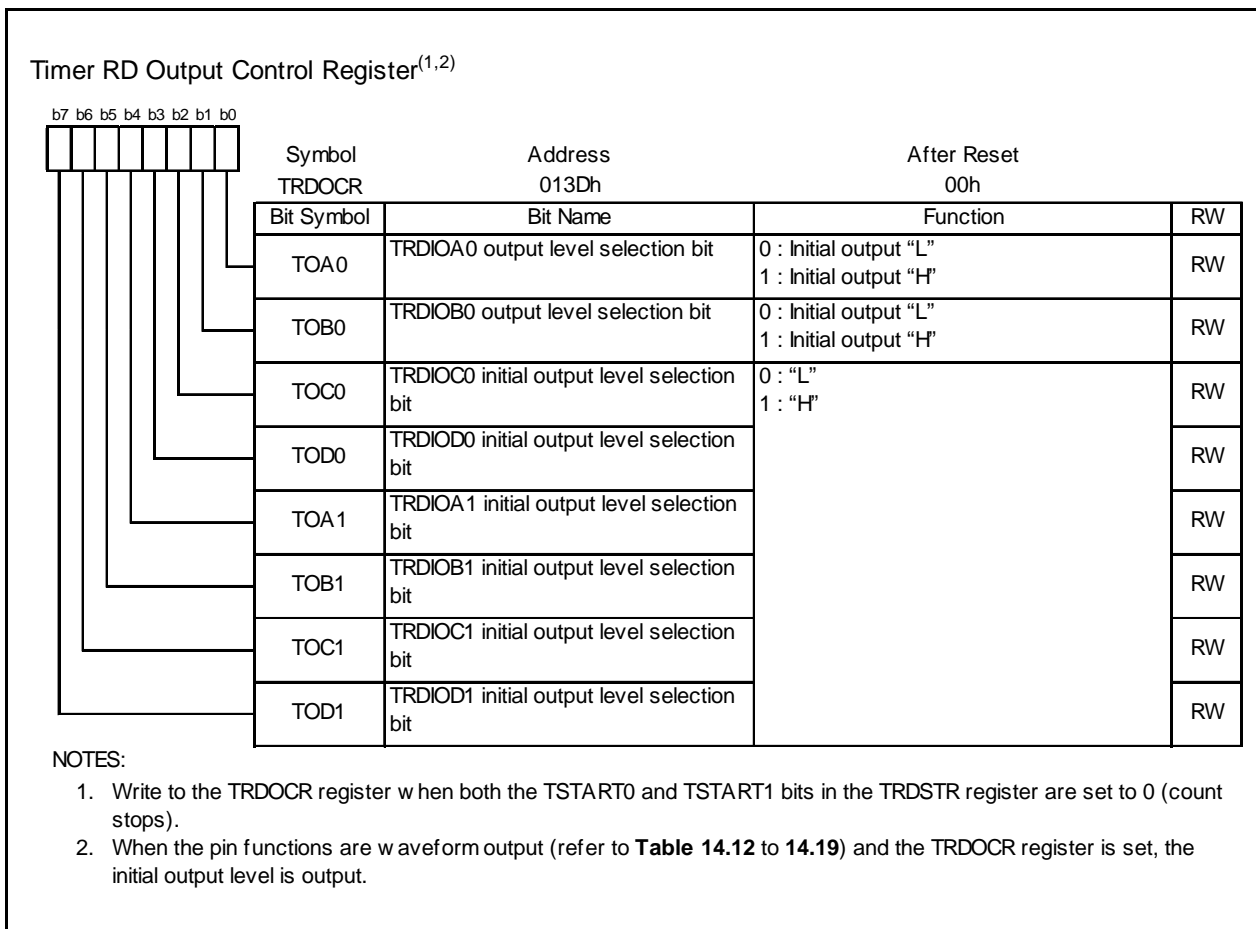


Figure 14.52 TRDOCR Register in Output Compare Function

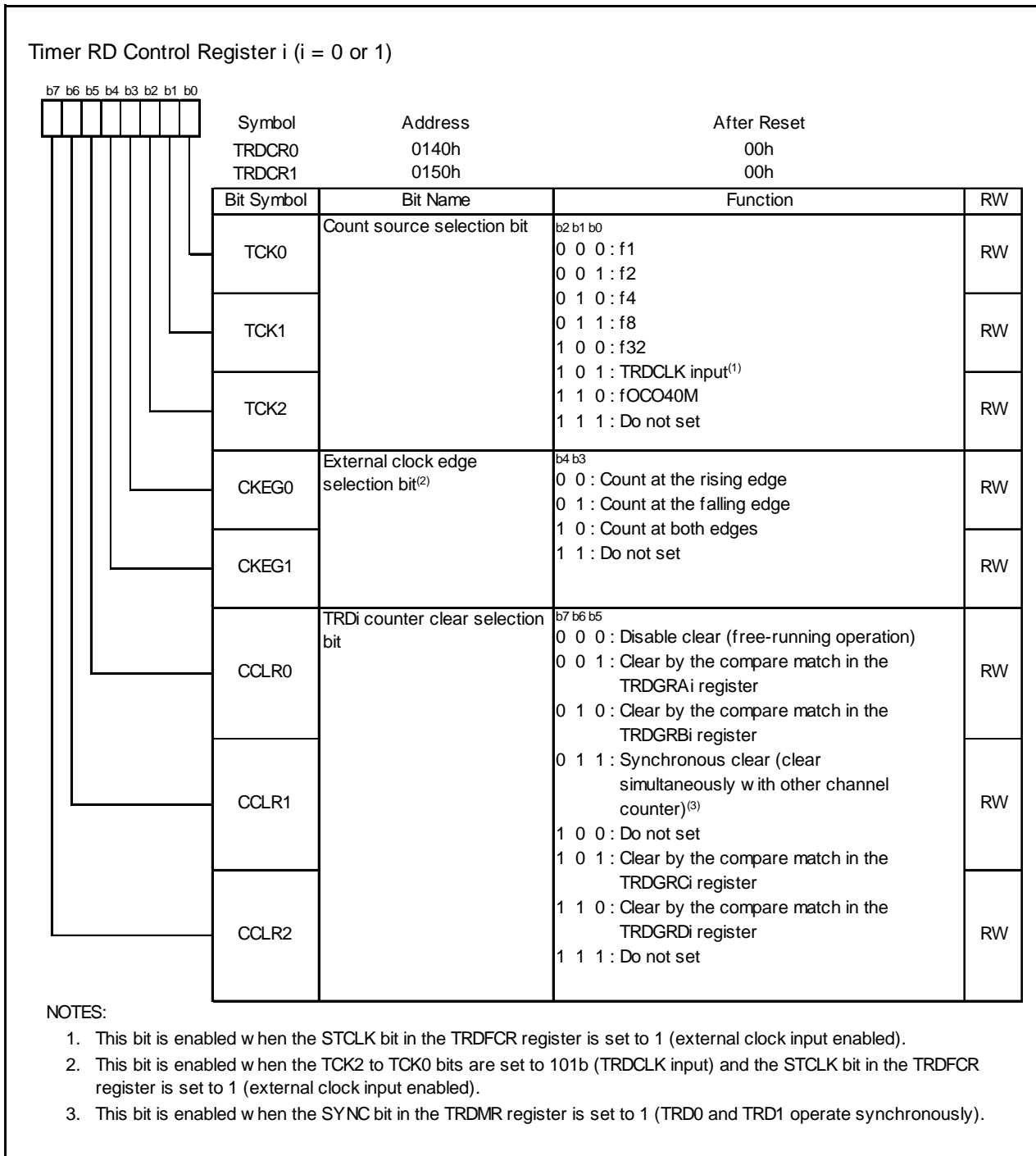
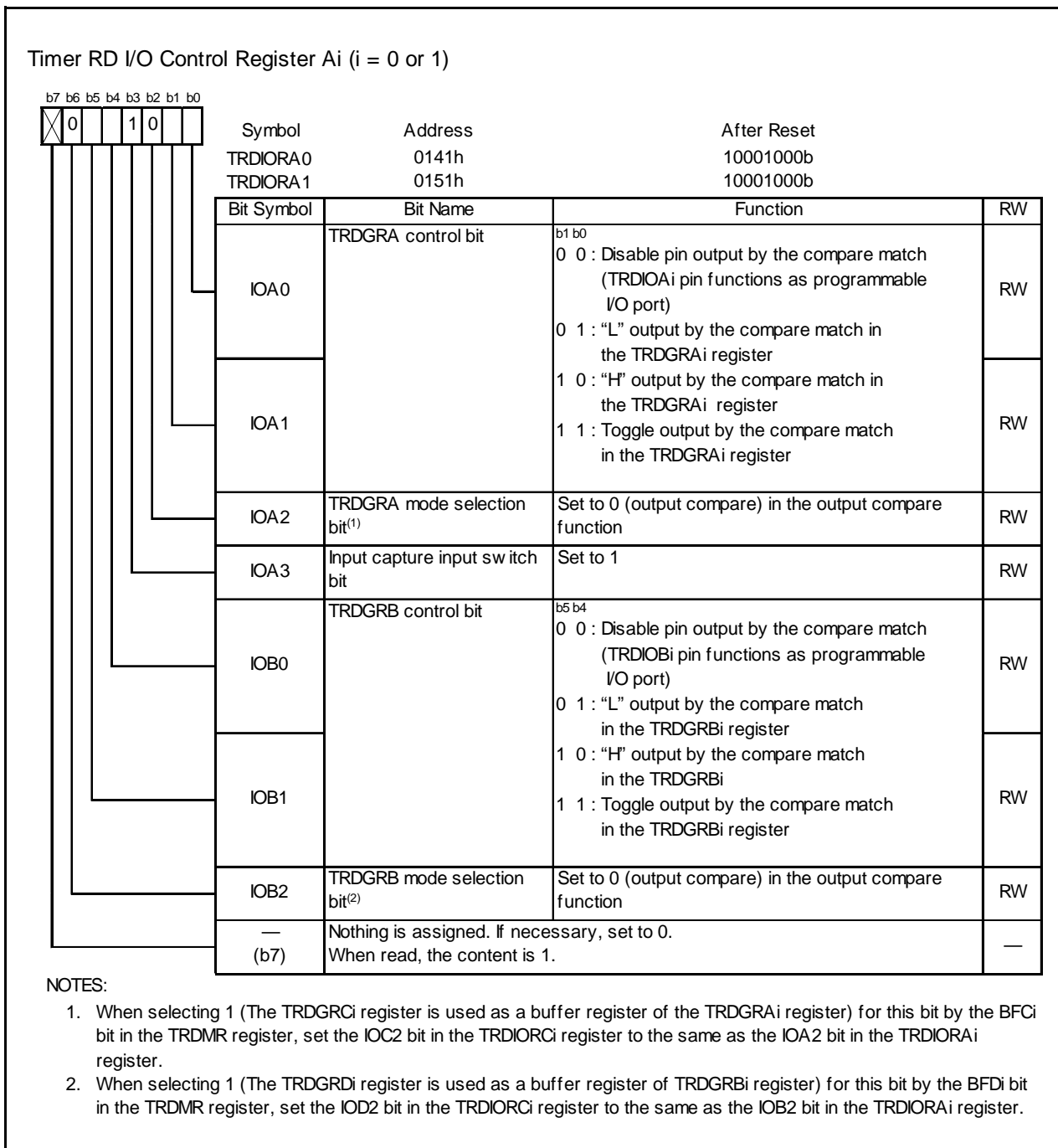


Figure 14.53 Registers TRDCR0 to TRDCR1 in Output Compare Function



**Figure 14.54** Registers TRDIORA0 to TRDIORA1 in Output Compare Function



Timer RD I/O Control Register Ci (i = 0 or 1)

Bit	Symbol	Address	After Reset
b7	0		
b6			
b5			
b4			
b3			
b2	0		
b1			
b0			
	Symbol	Address	After Reset
	TRDIORC0	0142h	10001000b
	TRDIORC1	0152h	10001000b

Bit Symbol	Bit Name	Function	RW
IOC0	TRDGRC control bit	b1 b0 0 0 : Disable pin output by the compare match	RW
		0 1 : "L" output by the compare match in the TRDGRCi register	
IOC1	TRDGRC control bit	1 0 : "H" output by the compare match in the TRDGRCi register	RW
		1 1 : Toggle output by the compare match in the TRDGRCi register	
IOC2	TRDGRC mode selection bit <sup>(1)</sup>	Set to 0 (output compare) in the output compare function	RW
IOC3	TRDGRC register function selection bit <sup>(1)</sup>	0 : TRDIOA output register (Refer to <b>14.3.6.1 Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi</b> ) 1 : General register or buffer register	RW
IOD0	TRDGRD control bit	b5 b4 0 0 : Disable pin output by the compare match	RW
		0 1 : "L" output by the compare match in the TRDGRDi register	
IOD1	TRDGRD control bit	1 0 : "H" output by the compare match in the TRDGRDi register	RW
		1 1 : Toggle output by the compare match in the TRDGRDi register	
IOD2	TRDGRD mode selection bit <sup>(2)</sup>	Set to 0 (output compare) in the output compare function	RW
IOD3	TRDGRD register function selection bit	0 : TRDIOB output register (Refer to <b>14.3.6.1 Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi</b> ) 1 : General register or buffer register	RW

NOTES:

- When selecting 1 (The TRDGRCi register is used as a buffer register of the TRDGRAi register) for this bit by the BFCi bit in the TRDMR register, set the IOC2 bit in the TRDIORCi register to the same as the IOA2 bit in the TRDIORAi register.
- When selecting 1 (The TRDGRDi register is used as a buffer register of TRDGRBi register) for this bit by the BFDi bit in the TRDMR register, set the IOD2 bit in the TRDIORCi register to the same as the IOB2 bit in the TRDIORAi register.

Figure 14.55 Registers TRDIORC0 to TRDIORC1 in Output Compare Function

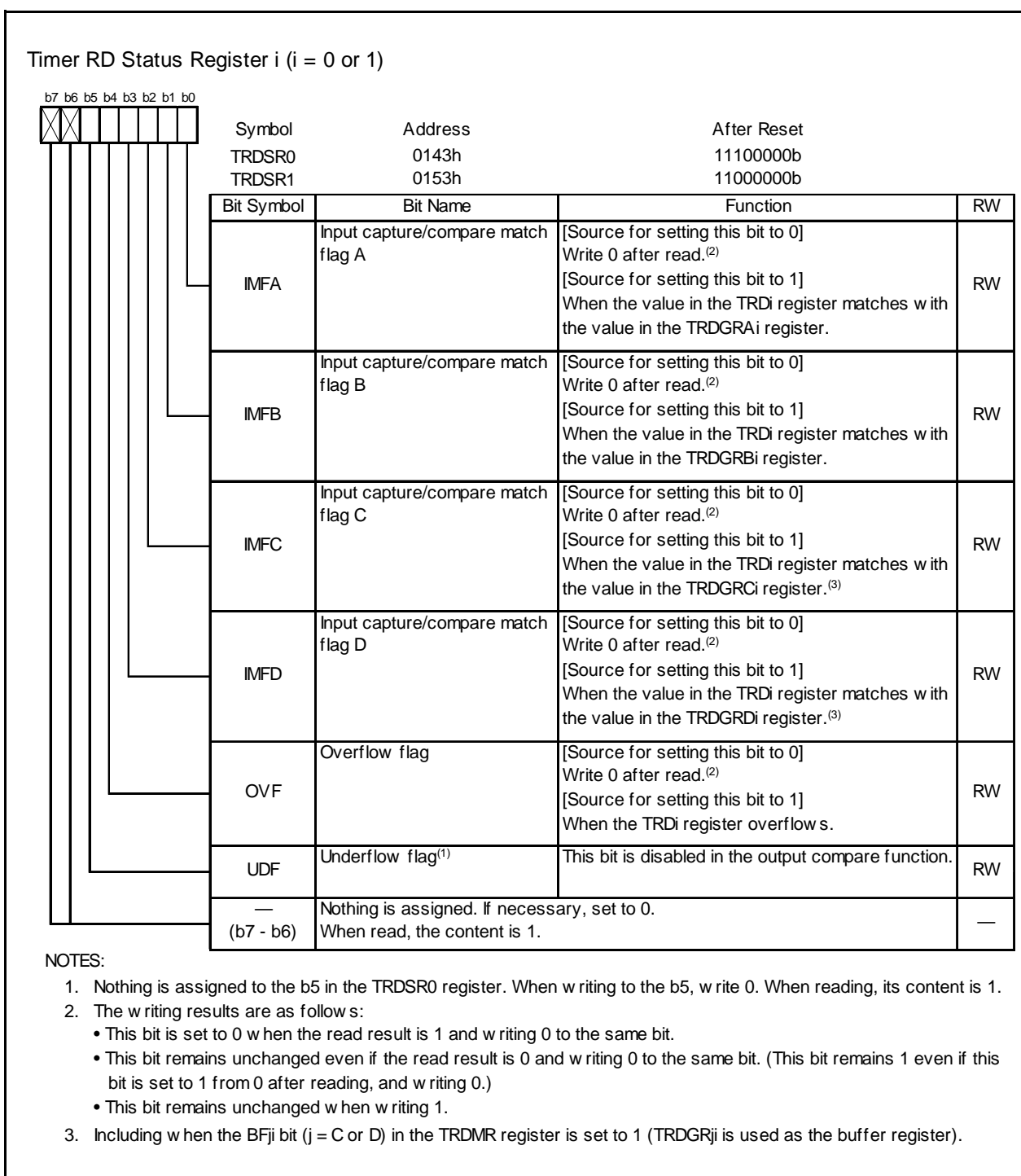


Figure 14.56 Registers TRDSR0 to TRDSR1 in Output Compare Function

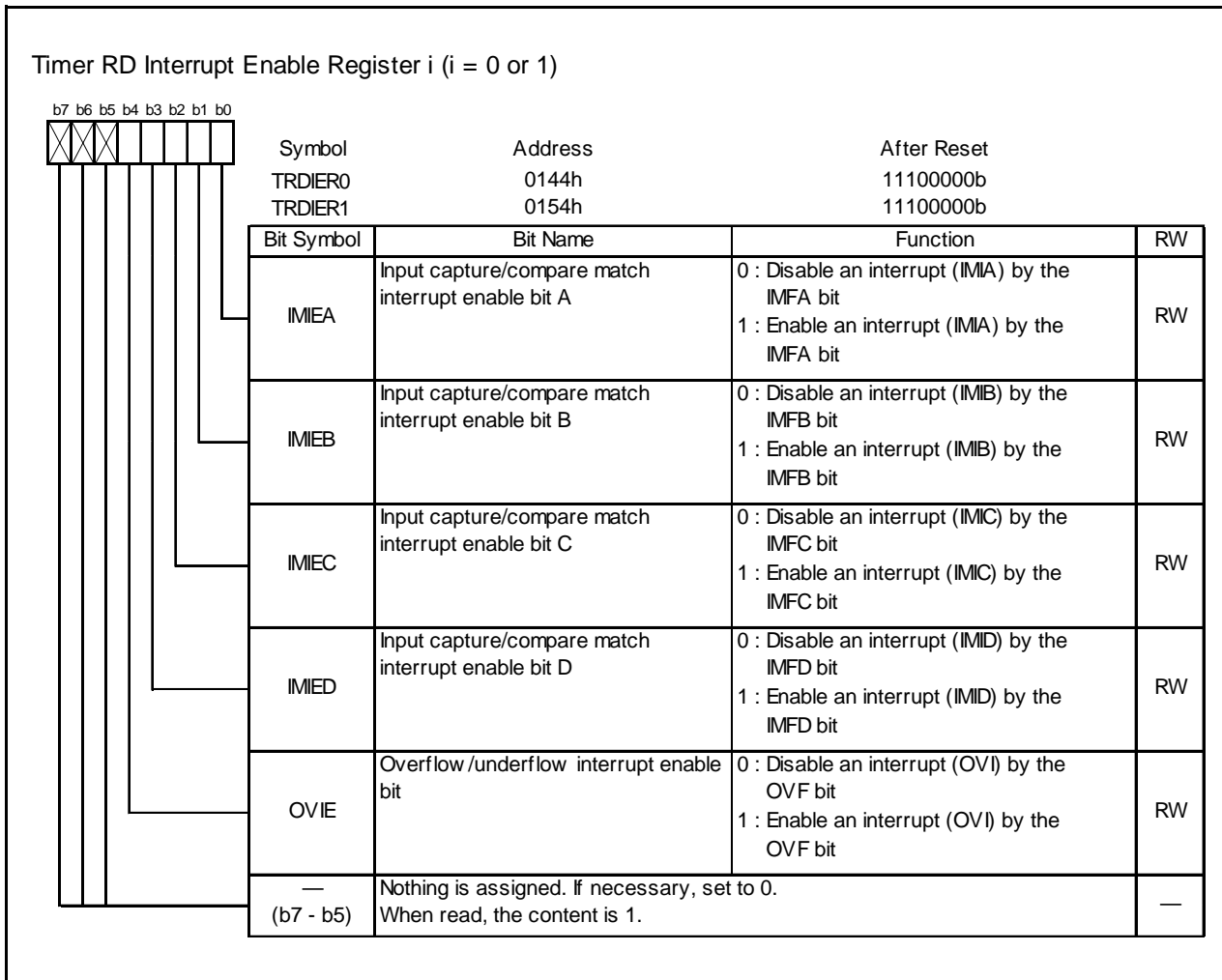


Figure 14.57 Registers TRDIER0 to TRDIER1 in Output Compare Function

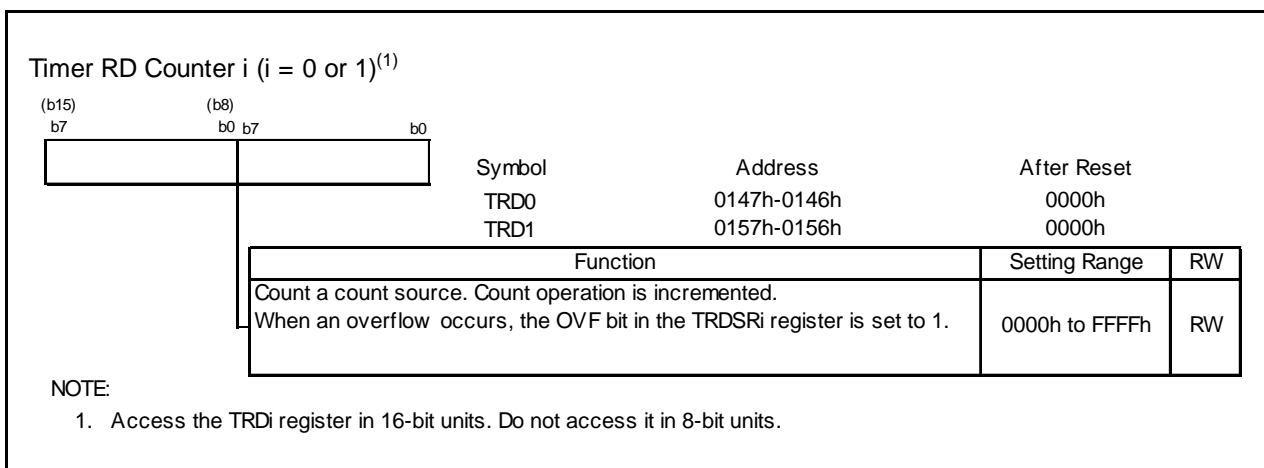


Figure 14.58 Registers TRD0 to TRD1 in Output Compare Function

Timer RD General Register Ai, Bi, Ci and Di (i = 0 or 1)<sup>(1)</sup>

(b15) b7	(b8) b0 b7	b0	Symbol	Address	After Reset
			TRDGRA0	0149h-0148h	FFFFh
			TRDGRB0	014Bh-014Ah	FFFFh
			TRDGRC0	014Dh-014Ch	FFFFh
			TRDGRD0	014Fh-014Eh	FFFFh
			TRDGRA1	0159h-0158h	FFFFh
			TRDGRB1	015Bh-015Ah	FFFFh
			TRDGRC1	015Dh-015Ch	FFFFh
			TRDGRD1	015Fh-015Eh	FFFFh
Function					RW
Refer to <b>Table 14.26 TRDGRji Register Functions in Output Compare Function</b>					RW

NOTE:

- Access the TRDGRAi to TRDGRDi registers in 16-bit units. Do not access them in 8-bit units.

**Figure 14.59 Registers TRDGRAi, TRDGRBi, TRDGRCi and TRDGRDi in Output Compare Function**

The following registers are disabled in the output compare function:  
TRDDF0, TRDDF1, TRDPOCR0 and TRDPOCR1

**Table 14.26 TRDGRji Register Functions in Output Compare Function**

Register	Setting		Register Function	Output-Compare Output Pin
	BFji	IOj3		
TRDGRAi	–	–	General register. Write the compare value.	TRDIOAi
TRDGRBi				TRDIOBi
TRDGRCi	0	1	General register. Write the compare value.	TRDIOCi
TRDGRDi				TRDIODi
TRDGRCi	1	1	Buffer register. Write the next compare value (refer to <b>14.3.2 Buffer Operation.</b> )	TRDIOAi
TRDGRDi				TRDIOBi
TRDGRCi	0	0	TRDIOAi output control (refer to <b>14.3.6.1 Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi.</b> )	TRDIOAi
TRDGRDi				TRDIOBi

i = 0 or 1, j = either A, B, C or D

BFji: Bit in TRDMR register

IOj3: Bit in TRDIORCi register

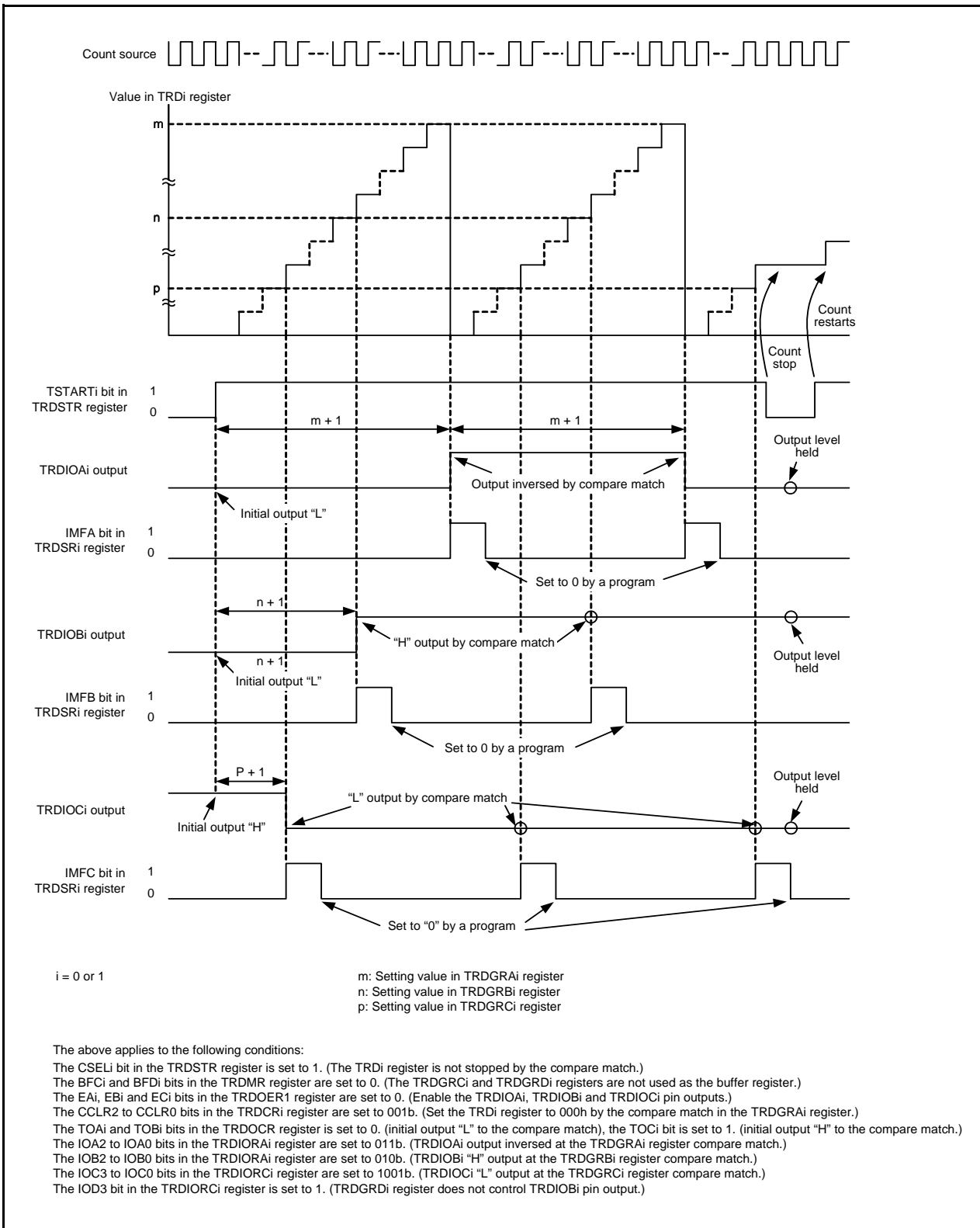


Figure 14.60 Operating Example of Output Compare Function

### 14.3.6.1 Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi

The TRDGRCi register can be used as output control of the TRDIOAi pin and the TRDGRDi register can be used as output control of the TRDIOBi pin. Therefore, each pin output can be controlled as follows:

- TRDIOAi output is controlled by the values in the TRDGRAi and TRDGRCi registers.
- TRDIOBi output is controlled by the values in the TRDGRBi and TRDGRDi registers.

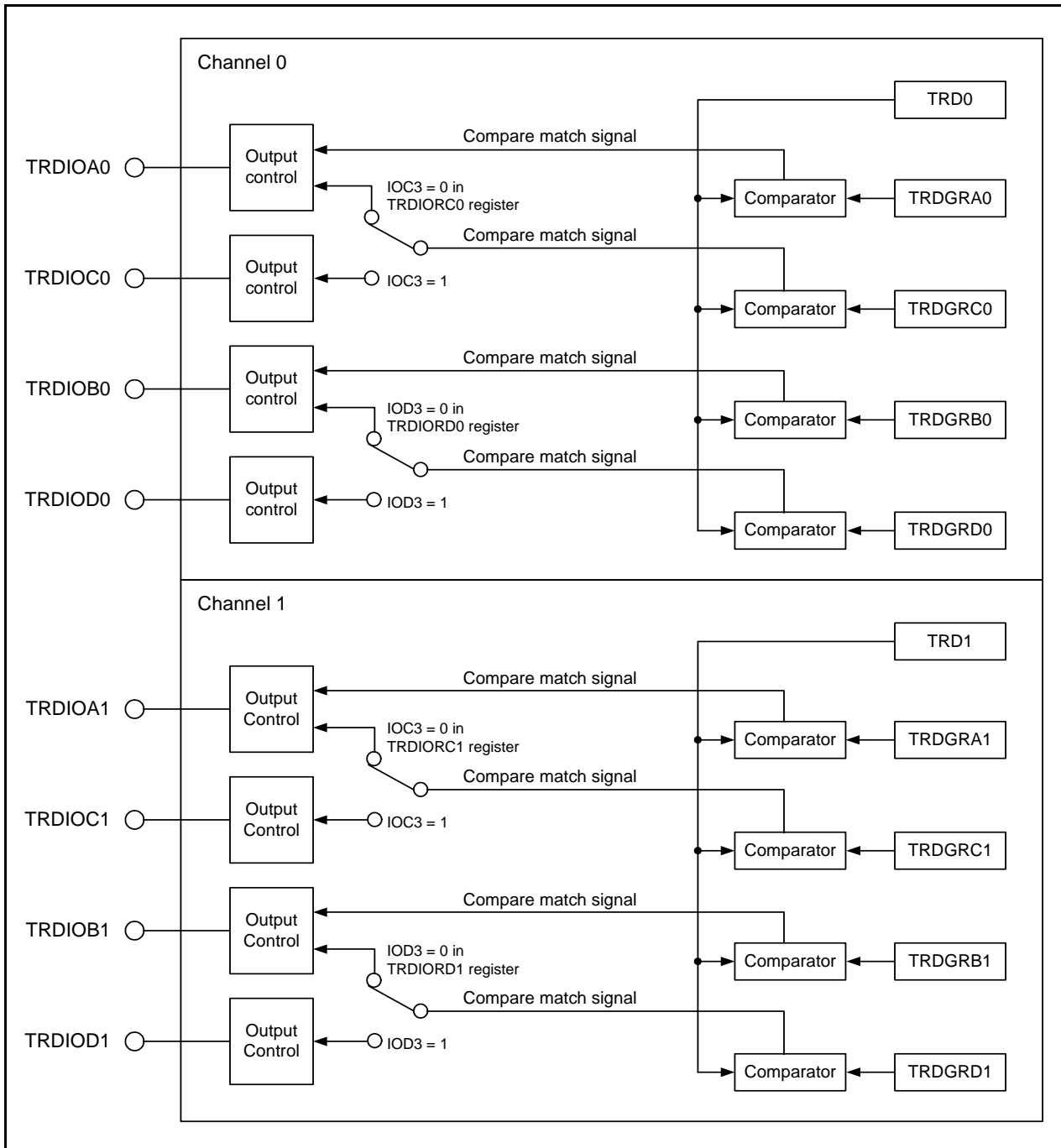


Figure 14.61 Changing Output Pins in Registers TRDGRCi and TRDGRDi

Change output pins in the TRDGRCi and TRDGRDi registers as below:

- Select 0 (change TRDGRji register output pin) by the IOj3 (j = C or D) bit in the TRDIORCi register.
- Set the BFji bit in the TRDMR register to 0 (general register).
- Set the different value in the TRDGRCi register and the TRDGRAi register. Also, set the different value in the TRDGRDi register and the TRDGRBi register.

Figure 14.62 lists the Operating Example When TRDGRCi Register is Used for Output Control of TRDIOAi Pin and TRDGRDi Register is Used for Output Control of TRDIOBi Pin.

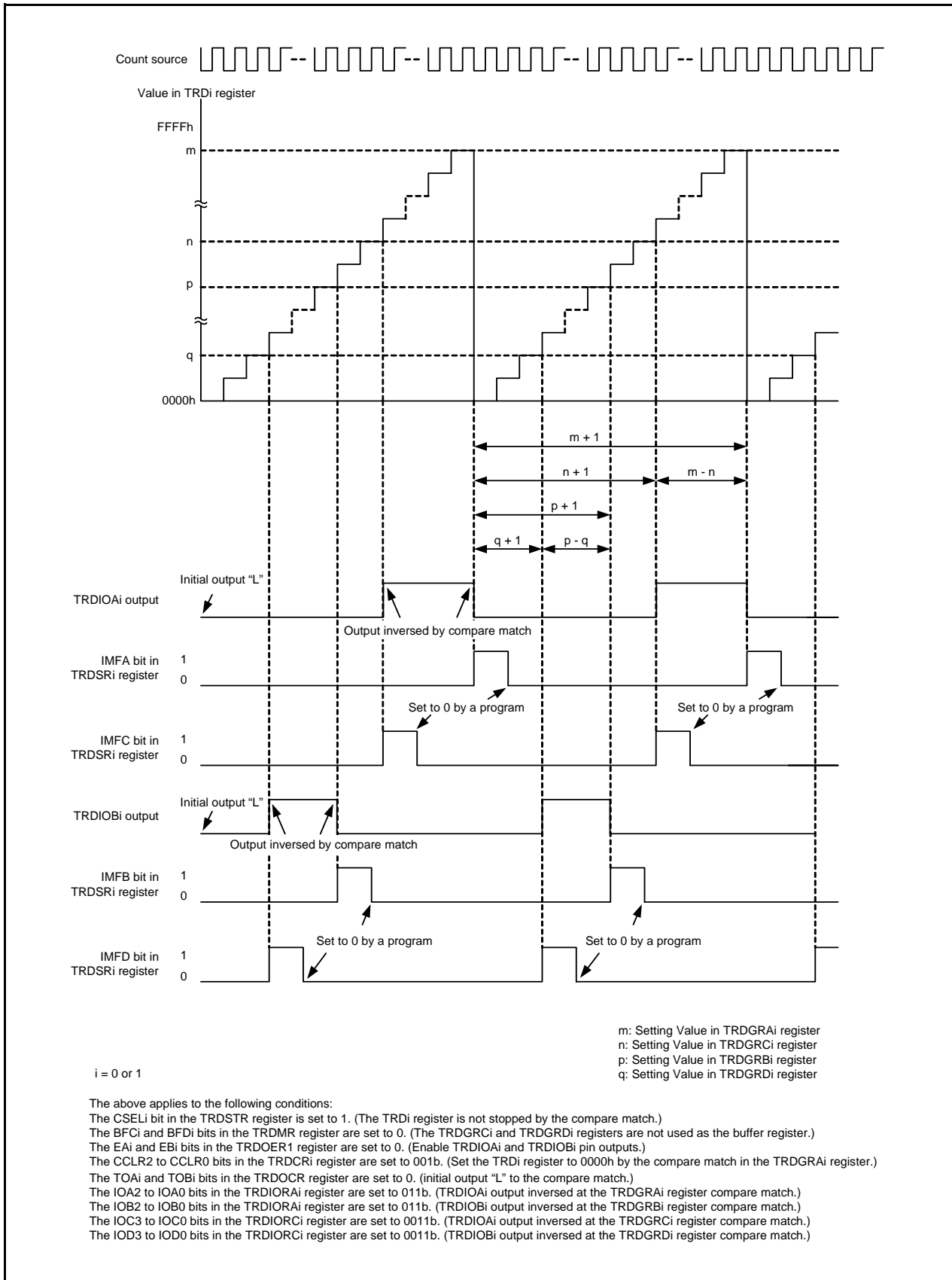


Figure 14.62 Operating Example When TRDGRCi Register is Used for Output Control of TRDIOAi Pin and TRDGRDi Register is Used for Output Control of TRDIOBi Pin

### 14.3.7 PWM Mode

PWM mode is to output a PWM waveform. Up to 3 PWM waveforms with the same period can be output by 1 channel. Also, Up to 6 PWM waveforms with the same period can be output by synchronizing Channels 0 and 1. Since this mode functions by a combination of the TRDIO<sub>ji</sub> ( $i = 0$  or  $1$ ,  $j = B, C$  or  $D$ ) pin and TRDGR<sub>ji</sub> register, any of PWM mode, other modes or functions can be selected every pin. (However, since the TRDGRA<sub>i</sub> register is used when using any pin for PWM mode, the TRDGRA<sub>i</sub> register cannot be used for other modes.)

Figure 14.63 shows the Block Diagram of PWM Mode, Table 14.27 lists the PWM Mode Specifications. Figures 14.64 to 14.73 show the Registers Associated with PWM Mode and Figures 14.74 to 14.75 show the Operations of PWM Mode.

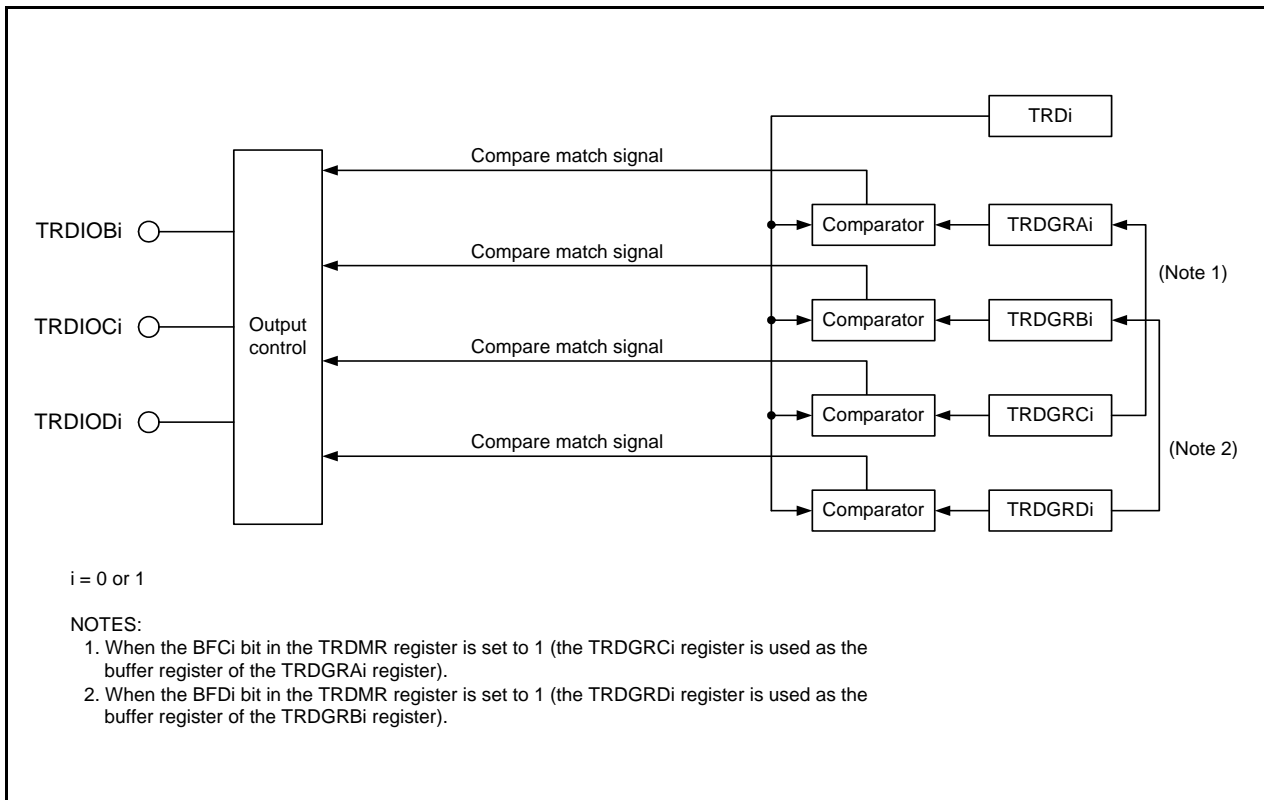
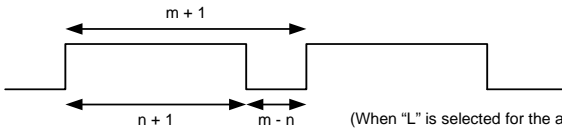


Figure 14.63 Block Diagram of PWM Mode



**Table 14.27 PWM Mode Specifications**

Item	Specification
Count Sources	f1, f2, f4, f8, f32, fOCO40M External signal input to the TRDCLK pin (valid edge selected by a program)
Count Operations	Increment
PWM Waveform	<p>PWM period: <math>1/f_k \times (m+1)</math>  Active level width: <math>1/f_k \times (m-n)</math>  Inactive level width: <math>1/f_k \times (n+1)</math>  <math>f_k</math>: Frequency of count source  <math>m</math>: Setting value in the TRDGRA<math>_i</math> register  <math>n</math>: Setting value in the TRDGR<math>_j</math> register</p>  <p>(When "L" is selected for the active level)</p>
Count Start Condition	Write 1 (count starts) to the TSTART $_i$ bit in the TRDSTR register.
Count Stop Conditions	<ul style="list-style-type: none"> <li>• Write 0 (count stops) to the TSTART<math>_i</math> bit in the TRDSTR register when the CSEL<math>_i</math> bit in the TRDSTR register is set to 1.</li> <li>• The PWM output pin holds output level before the count stops.</li> <li>• When the CSEL<math>_i</math> bit in the TRDSTR register is set to 0, the count stops at the compare match in the TRDGRA<math>_i</math> register.</li> <li>• The PWM output pin holds level after output change by the compare match.</li> </ul>
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>• Compare match (the content in the TRD<math>_i</math> register matches with the content in the TRDGR<math>_h</math> register.)</li> <li>• TRD<math>_i</math> register overflows</li> </ul>
TRDIOA0 Pin Function	Programmable I/O port or TRDCLK (external clock) input
TRDIOA1 Pin Function	Programmable I/O port
TRDIOB0, TRDIOC0, TRDIOD0, TRDIOB1, TRDIOC1, TRDIOD1 Pin Functions	Programmable I/O port or pulse output (select every pin)
INT0 Pin Function	Programmable I/O port, pulse output forced cutoff signal input or $\overline{\text{INT0}}$ interrupt input
Read from Timer	The count value can be read by reading the TRD $_i$ register.
Write to Timer	The value can be written to the TRD $_i$ register.
Selection Functions	<ul style="list-style-type: none"> <li>• 1 to 3 PWM output pins selected per 1 channel</li> <li>• Either 1 pin or multiple pins of the TRDIOB<math>_i</math>, TRDIOC<math>_i</math> or TRDIOD<math>_i</math> pin.</li> <li>• The active level selected every pin.</li> <li>• Initial output level selected every pin.</li> <li>• Synchronous operation (refer to <b>14.3.3 Synchronous Operation.</b>)</li> <li>• Buffer operation (refer to <b>14.3.2 Buffer Operation.</b>)</li> <li>• Pulse output forced cutoff signal input (refer to <b>14.3.4 Pulse Output Forced Cutoff.</b>)</li> </ul>

$i = 0$  or  $1$ ,  $j =$  either B, C or D,  
 $h =$  either A, B, C or D

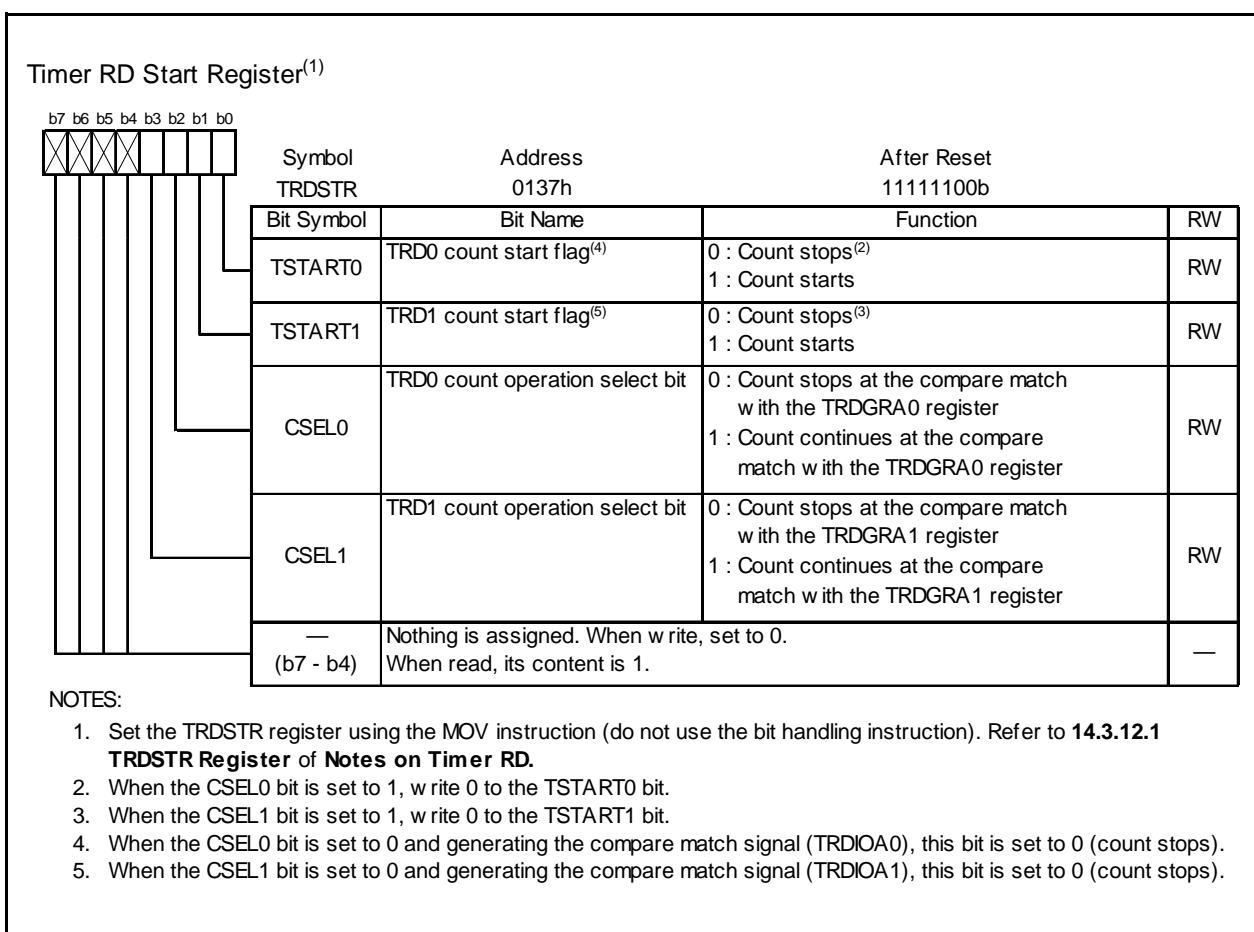


Figure 14.64 TRDSTR Register in PWM Mode

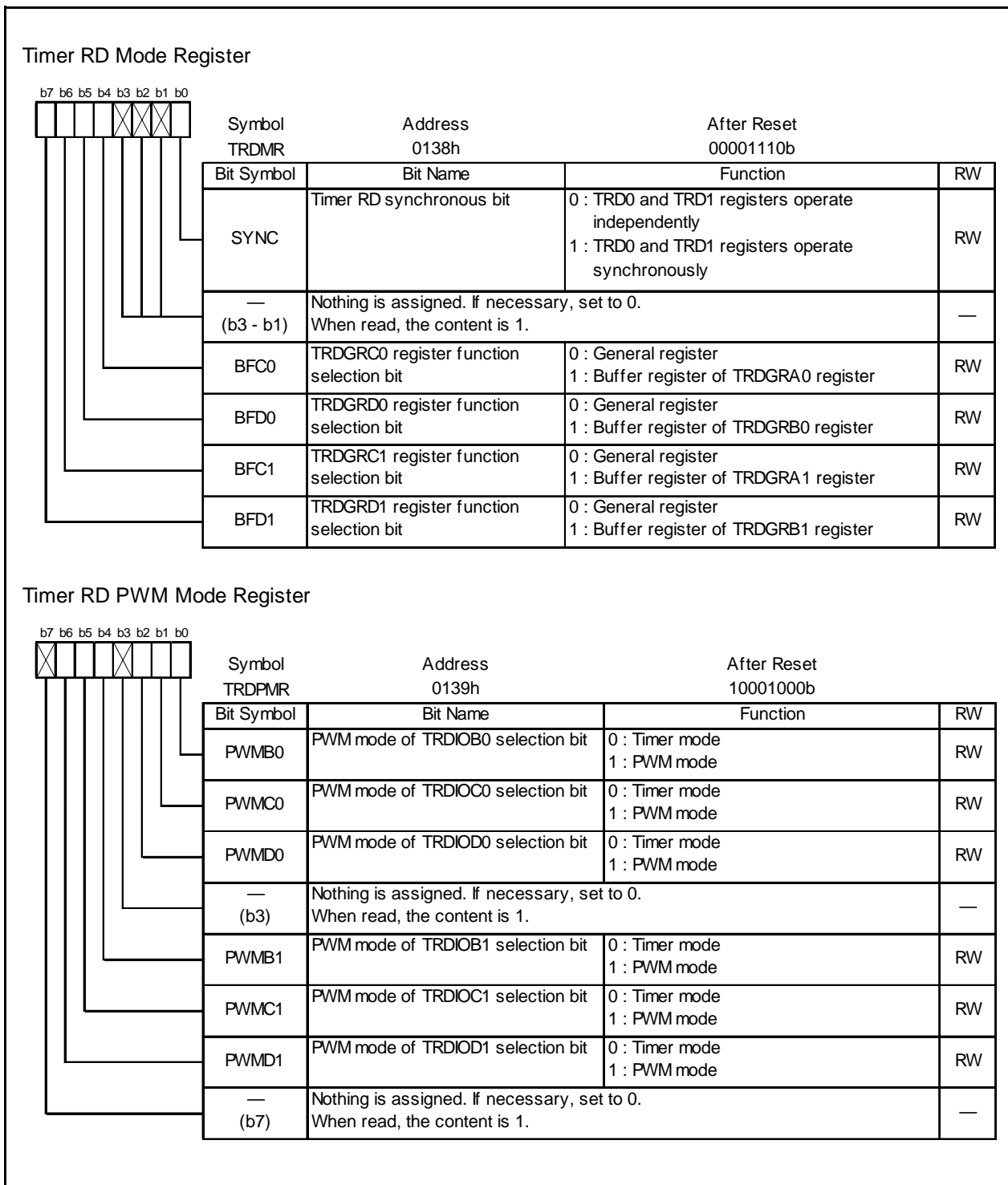


Figure 14.65 Registers TRDMR and TRDPMR in PWM Mode

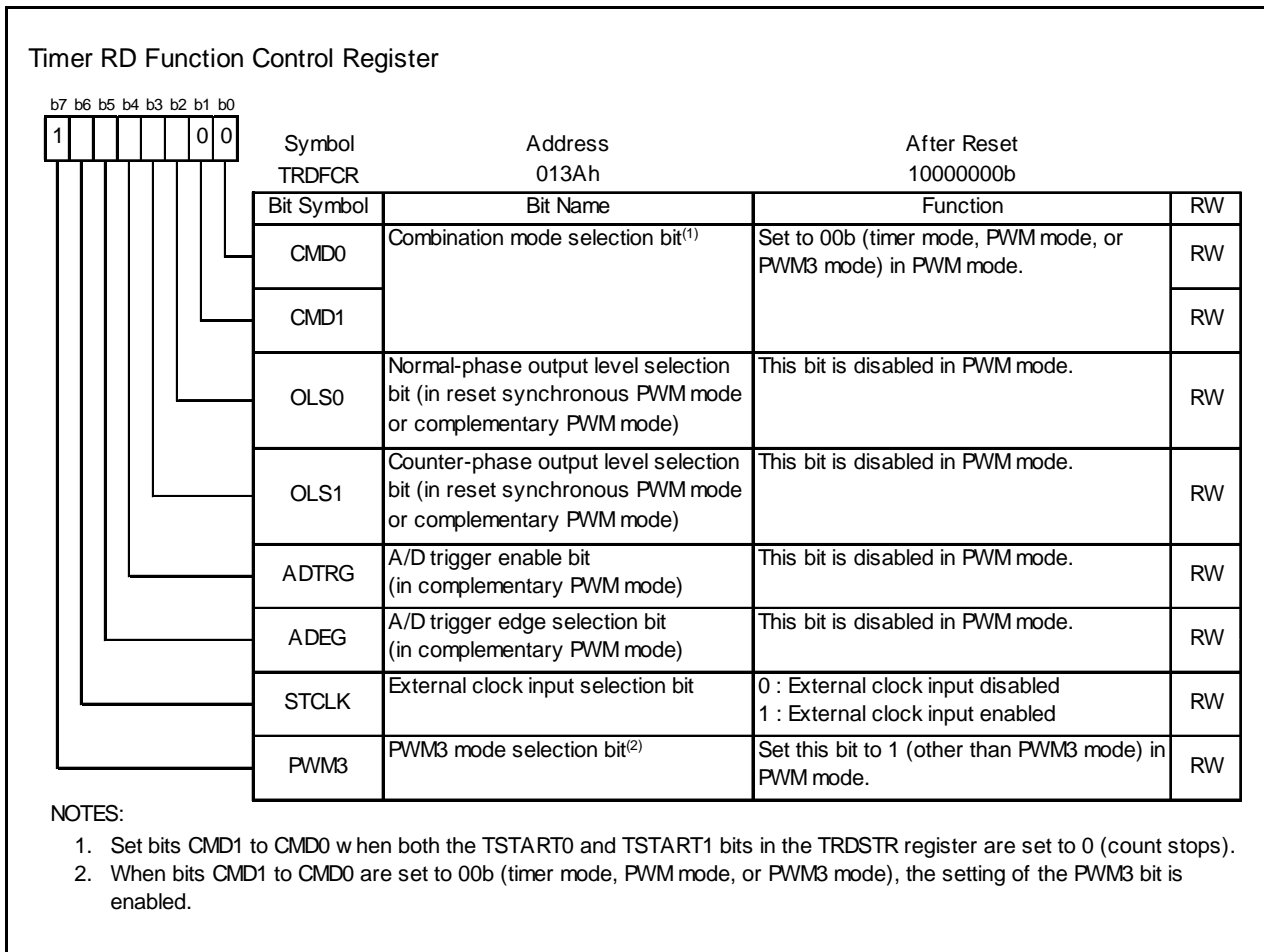


Figure 14.66 TRDFCR Register in PWM Mode

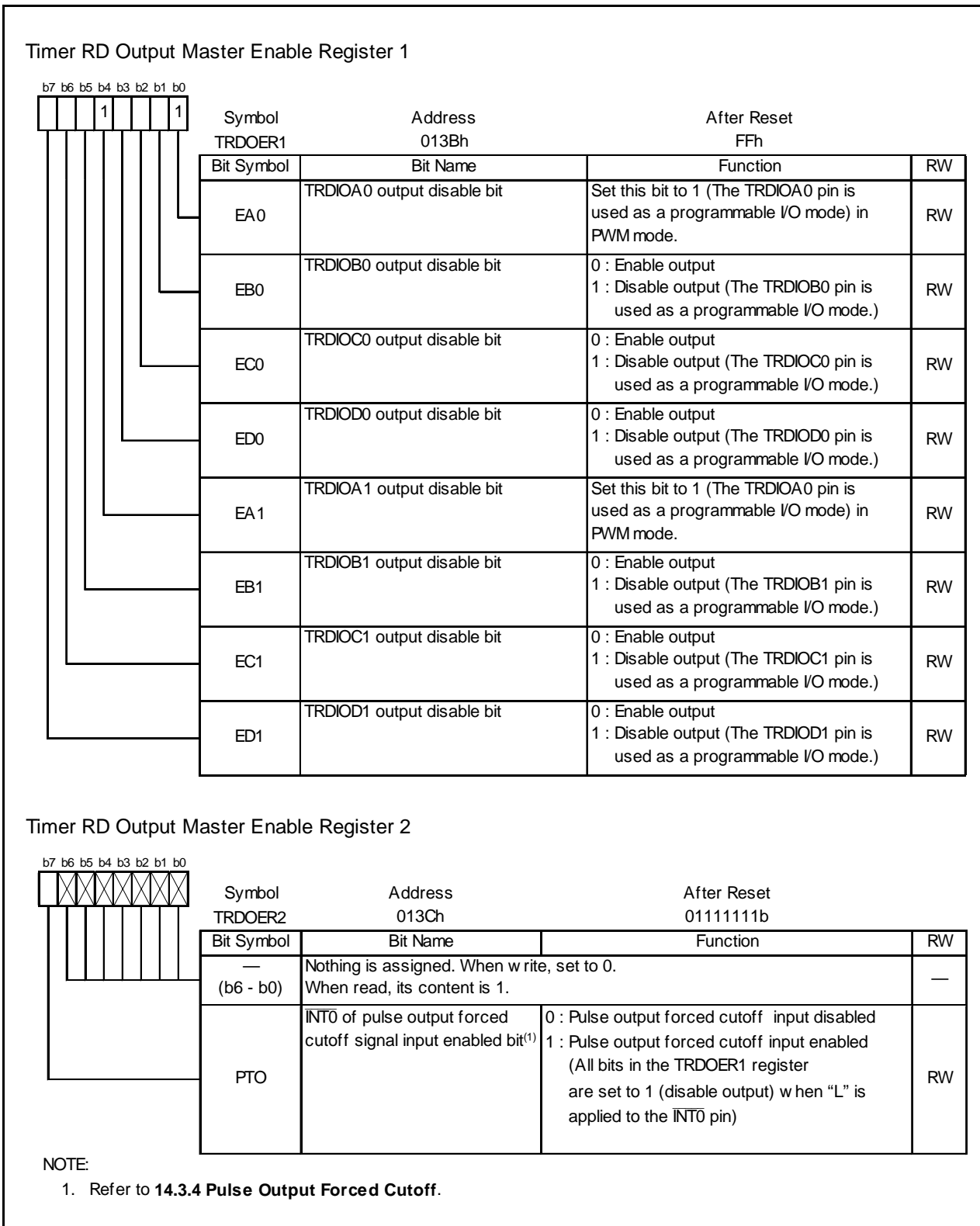


Figure 14.67 Registers TRDOER1 to TRDOER2 in PWM Mode

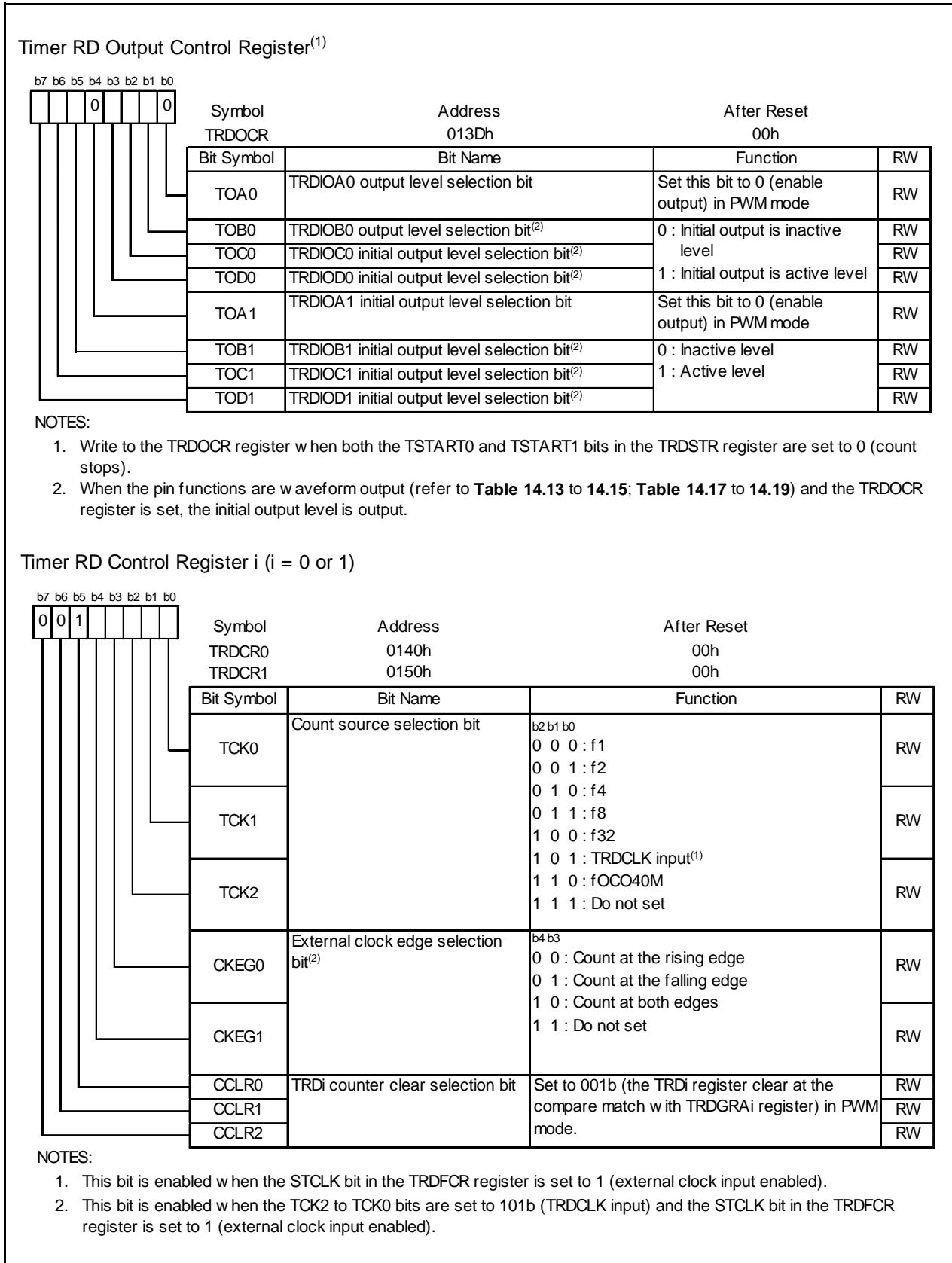


Figure 14.68 Registers TRDOCR and TRDCR0 to TRDCR1 in PWM Mode

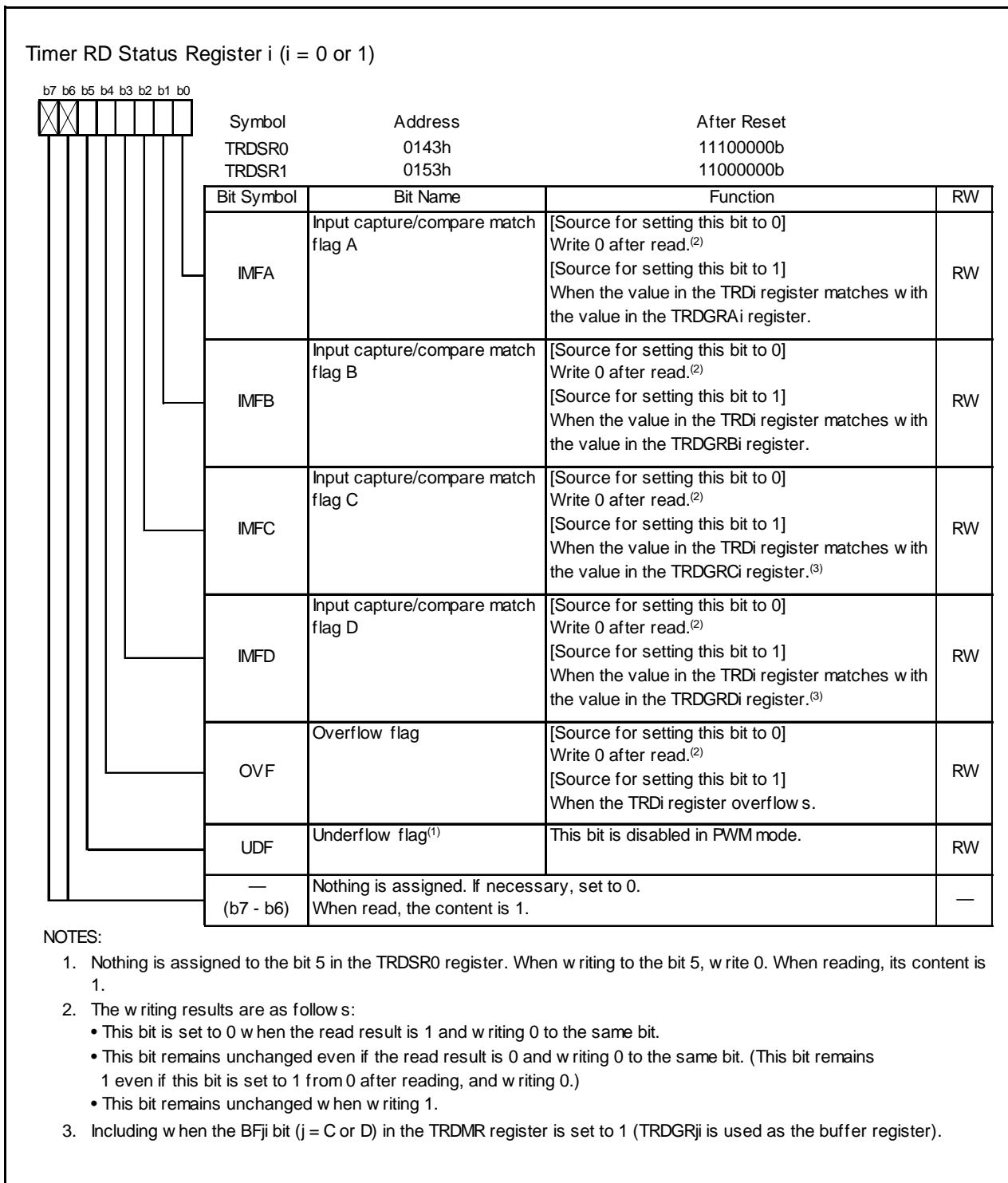


Figure 14.69 Registers TRDSR0 to TRDSR1 in PWM Mode

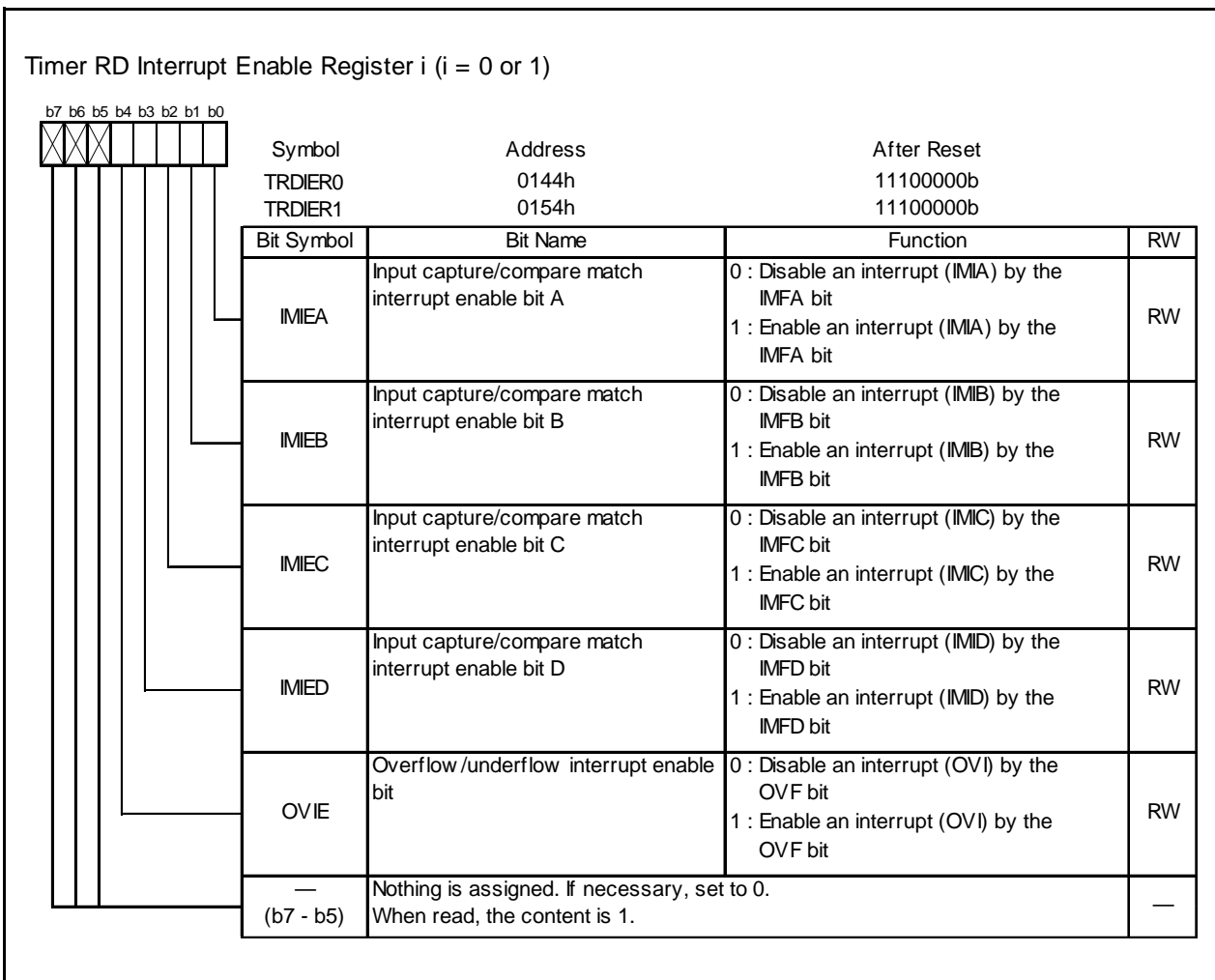


Figure 14.70 Registers TRDIER0 to TRDIER1 in PWM Mode



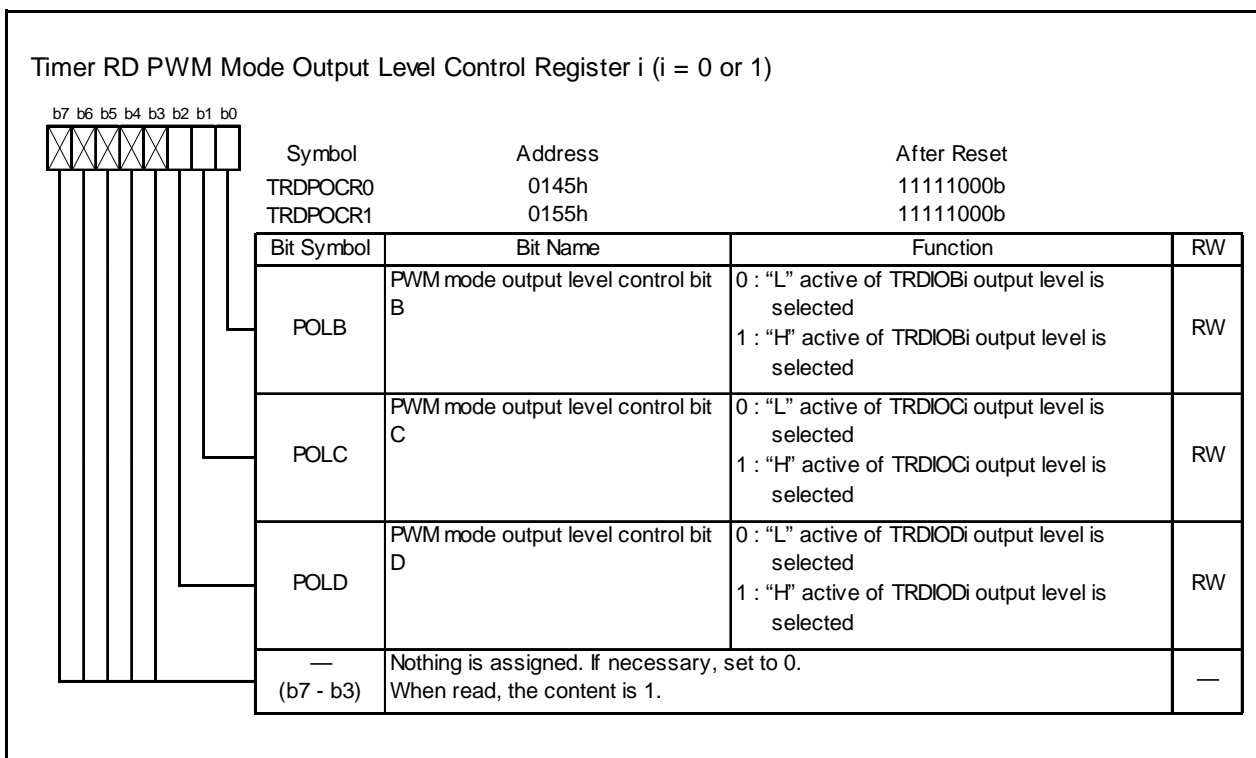


Figure 14.71 Registers TRDPOCR0 to TRDPOCR1 in PWM Mode

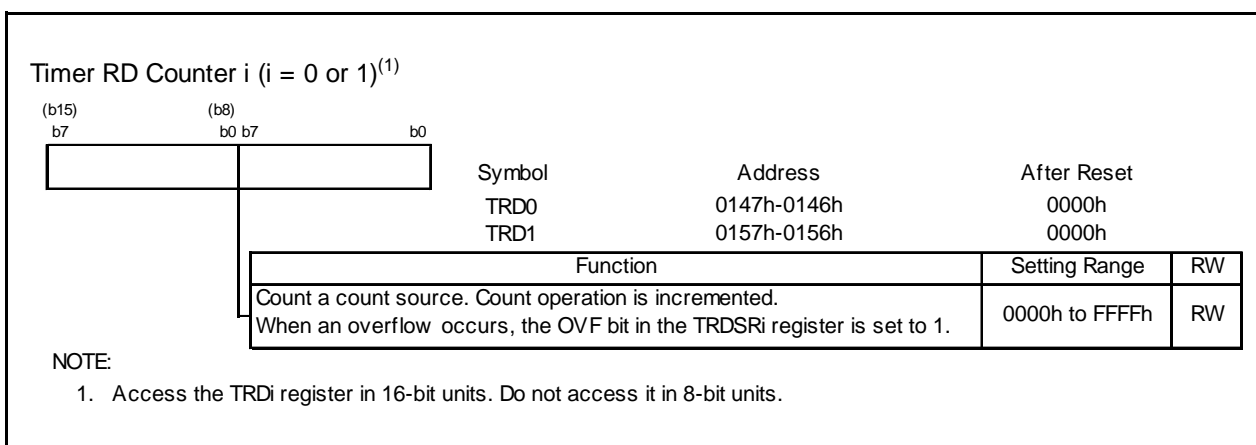


Figure 14.72 Registers TRD0 to TRD1 in PWM Mode

Timer RD General Register Ai, Bi, Ci and Di (i = 0 or 1)<sup>(1)</sup>

(b15) b7	(b8) b0 b7	b0	Symbol	Address	After Reset
			TRDGRA0	0149h-0148h	FFFFh
			TRDGRB0	014Bh-014Ah	FFFFh
			TRDGRC0	014Dh-014Ch	FFFFh
			TRDGRD0	014Fh-014Eh	FFFFh
			TRDGRA1	0159h-0158h	FFFFh
			TRDGRB1	015Bh-015Ah	FFFFh
			TRDGRC1	015Dh-015Ch	FFFFh
			TRDGRD1	015Fh-015Eh	FFFFh
Function					RW
Refer to Table 14.28 TRDGRji Register Functions in PWM Mode					RW

NOTE:  
1. Access the TRDGRAi to TRDGRDi registers in 16-bit units. Do not access them in 8-bit units.

**Figure 14.73 Registers TRDGRAi, TRDGRBi, TRDGRCi and TRDGRDi in PWM Mode**

The following registers are disabled in the PWM mode:

TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDIORA1 and TRDIORC1

**Table 14.28 TRDGRji Register Functions in PWM Mode**

Register	Setting	Register Function	PWM Output Pin
TRDGRAi	–	General register. Set the PWM period.	–
TRDGRBi	–	General register. Set the changing point of PWM output	TRDIOBi
TRDGRCi	BFCi = 0	General register. Set the changing point of PWM output	TRDIOCi
TRDGRDi	BFDi = 0		TRDIODi
TRDGRCi	BFCi = 1	Buffer register. Set the next PWM period (refer to <b>14.3.2 Buffer Operation.</b> )	–
TRDGRDi	BFDi = 1	Buffer register. Set the changing point of the next PWM output (refer to <b>14.3.2 Buffer Operation.</b> )	TRDIOBi

i = 0 or 1

BFCi, BFDi: Bits in TRDMR register

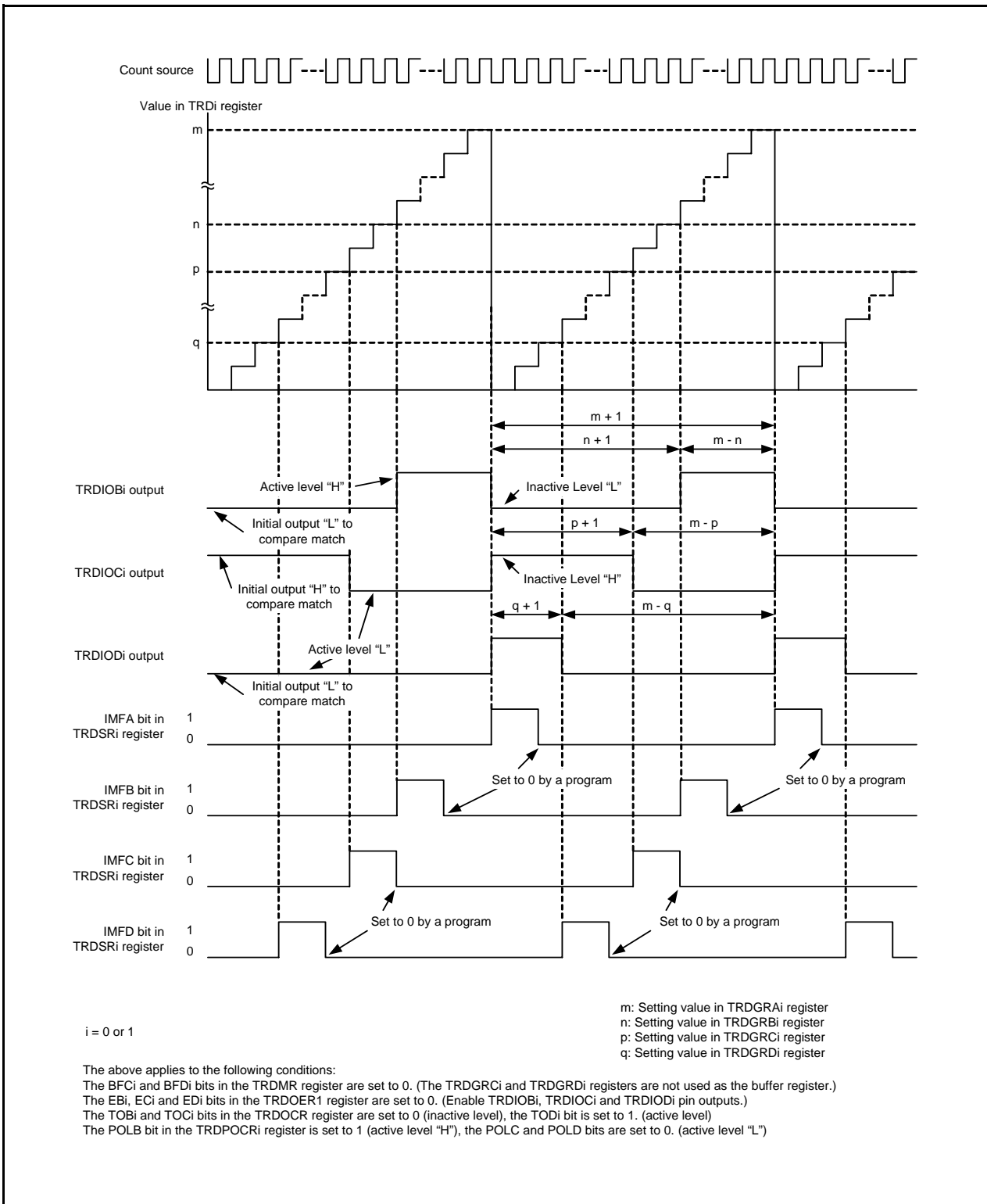


Figure 14.74 Operating Example of PWM Mode

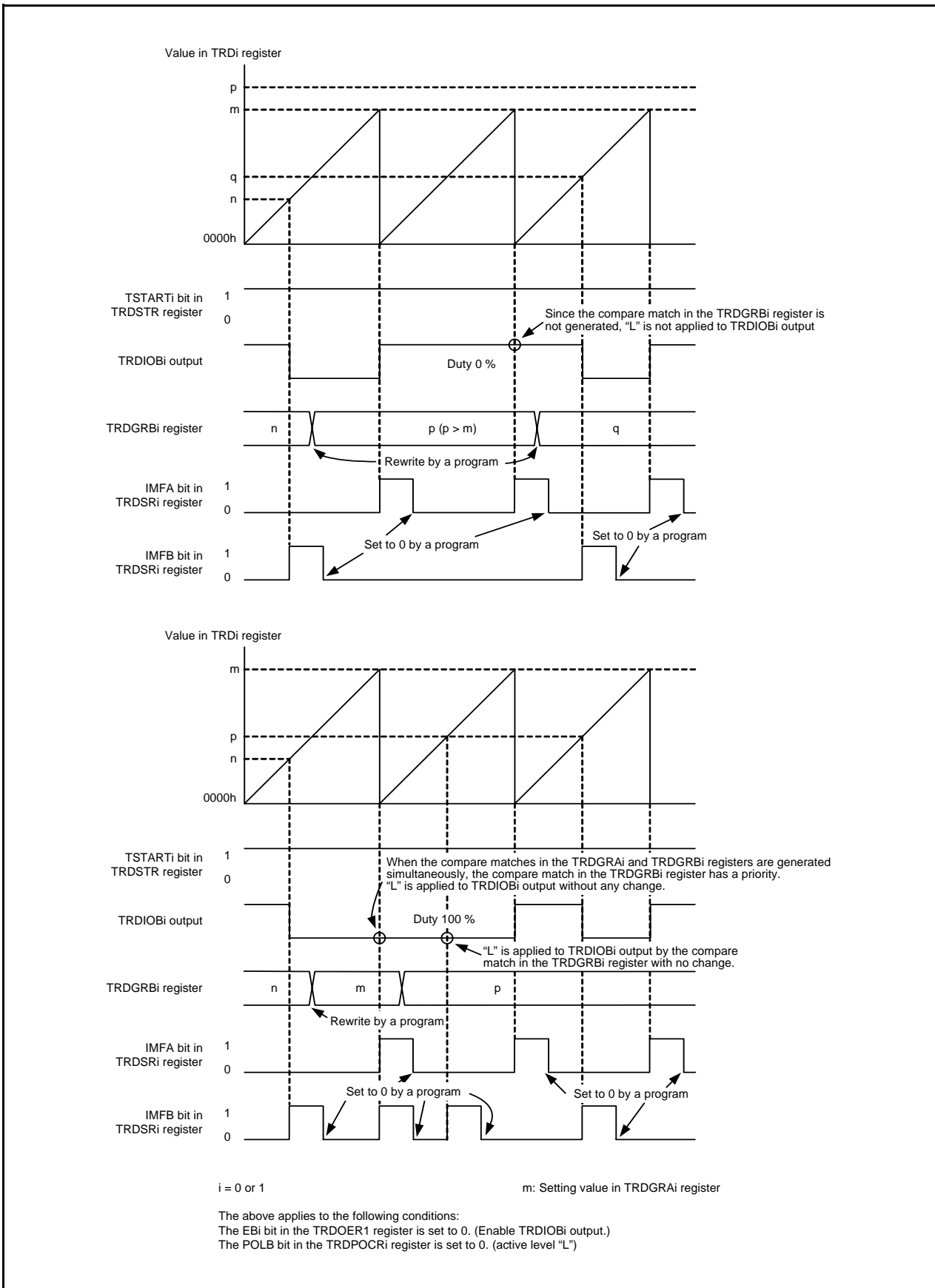


Figure 14.75 Operating Example of PWM Mode (Duty 0%, Duty 100%)

### 14.3.8 Reset Synchronous PWM Mode

Output 3 normal-phases and 3 counter-phases of the PWM waveform with the same period (no three-phase, sawtooth wave modulation and dead time).

Figure 14.76 shows the Block Diagram of Reset Synchronous PWM Mode, Table 14.29 lists the Reset Synchronous PWM Mode Specifications. Figures 14.77 to 14.84 show the Registers Associated with Reset Synchronous PWM Mode and Figure 14.85 shows the Operating Example of Reset Synchronous PWM Mode. Refer to **Figure 14.75 Operating Example of PWM Mode (Duty 0%, Duty 100%)** for the operation example in PWM Mode of duty 0% and duty 100%.

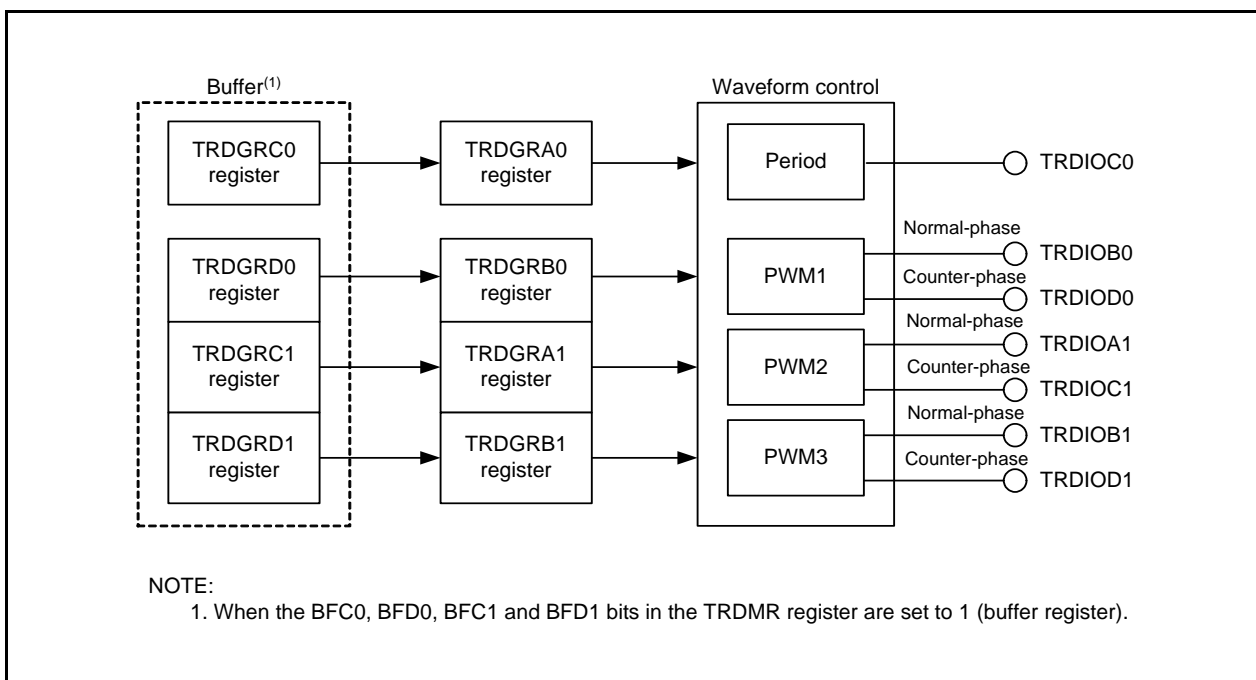


Figure 14.76 Block Diagram of Reset Synchronous PWM Mode

**Table 14.29 Reset Synchronous PWM Mode Specifications**

Item	Specification
Count Sources	f1, f2, f4, f8, f32, fOCO40M External signal input to the TRDCLK pin (valid edge selected by a program)
Count Operations	The TRD0 register is incremented (The TRD1 register is not used.)
PWM Waveform	<p>PWM period: <math>1/f_k \times (m + 1)</math>  Active level width of normal-phase: <math>1/f_k \times (m - n)</math>  Active level width of counter-phase: <math>1/f_k \times (n + 1)</math>  <math>f_k</math>: Frequency of count source  <math>m</math>: Setting value in the TRDGRA0 register  <math>n</math>: Setting value in the TRDGRB0 register (PWM output 1),  Setting value in the TRDGRA1 register (PWM output 2),  Setting value in the TRDGRB1 register (PWM output 3)</p> <p>(When "L" is selected for the active level)</p>
Count Start Condition	Write 1 (count starts) to the TSTART0 bit in the TRDSTR register.
Count Stop Conditions	<ul style="list-style-type: none"> <li>Write 0 (count stops) to the TSTART0 bit in the TRDSTR register when the CSEL0 bit in the TRDSTR register is set to 1.</li> <li>The PWM output pin holds output level before the count stops</li> <li>When the CSEL0 bit in the TRDSTR register is set to 0, the count stops at the compare match in the TRDGRA0 register.</li> <li>The PWM output pin holds level after output change by the compare match.</li> </ul>
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>Compare match (the content in the TRD0 register matches with the content in the TRDGRj0, TRDGRA1 and TRDGRB1 registers.)</li> <li>The TRD0 register overflows</li> </ul>
TRDIOA0 Pin Function	Programmable I/O port or TRDCLK (external clock) input
TRDIOB0 Pin Function	PWM output 1 normal-phase output
TRDIOD0 Pin Function	PWM output 1 counter-phase output
TRDIOA1 Pin Function	PWM output 2 normal-phase output
TRDIOC1 Pin Function	PWM output 2 counter-phase output
TRDIOB1 Pin Function	PWM output 3 normal-phase output
TRDIOD1 Pin Function	PWM output 3 counter-phase output
TRDIOC0 Pin Function	Output inverted every period of PWM
INT0 Pin Function	Programmable I/O port, pulse output forced cutoff signal input or INT0 interrupt input
Read from Timer	The count value can be read by reading the TRD0 register.
Write to Timer	The value can be written to the TRD0 register.
Selection Functions	<ul style="list-style-type: none"> <li>The active level of normal-phase and counter-phase and initial output level selected individually.</li> <li>Buffer operation (refer to <b>14.3.2 Buffer Operation.</b>)</li> <li>Pulse output forced cutoff signal input (refer to <b>14.3.4 Pulse Output Forced Cutoff.</b>)</li> </ul>

j = either A, B, C or D

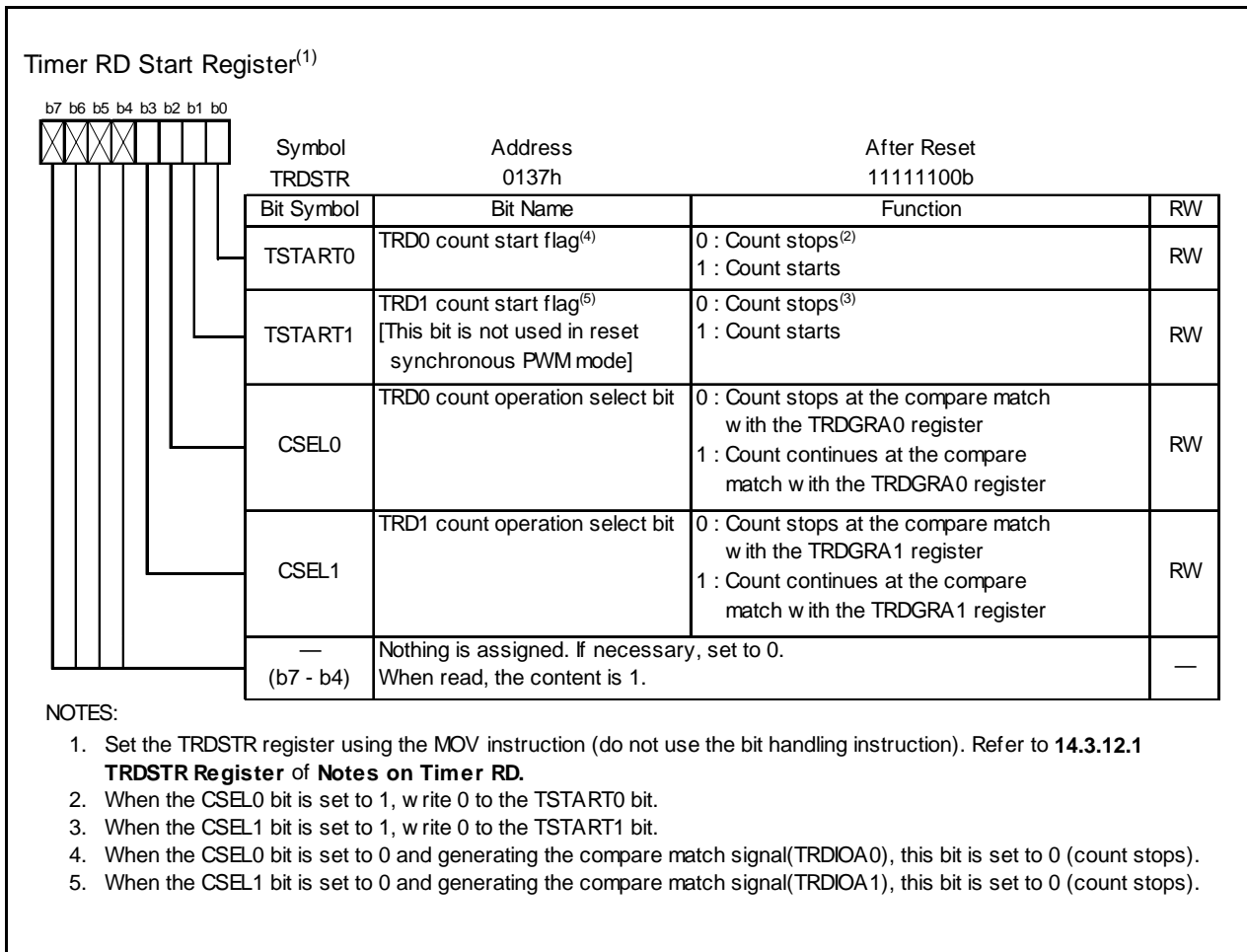


Figure 14.77 TRDSTR Register in Reset Synchronous PWM Mode

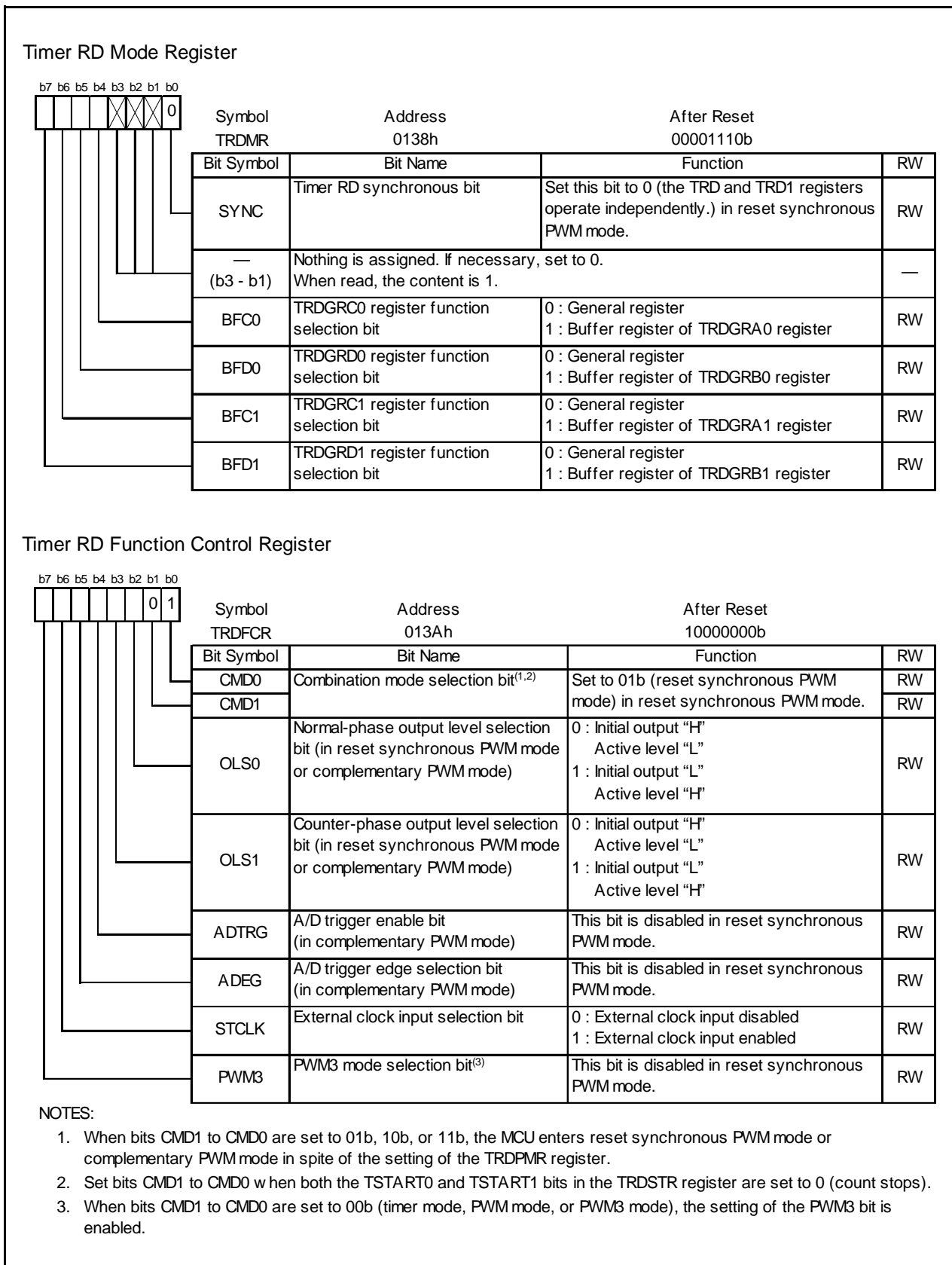


Figure 14.78 Registers TRDMR and TRDFCR in Reset Synchronous PWM Mode



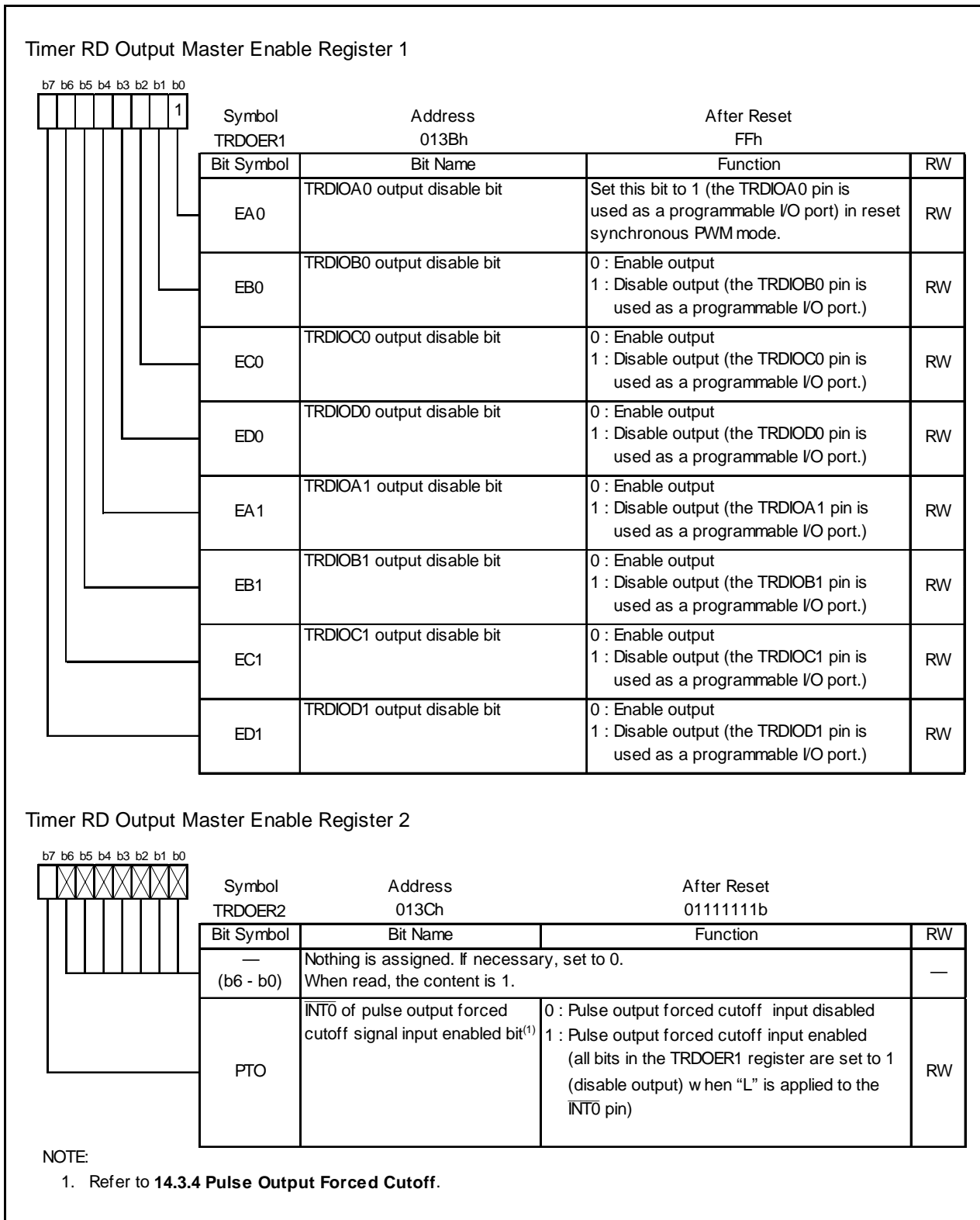


Figure 14.79 Registers TRDOER1 to TRDOER2 in Reset Synchronous PWM Mode

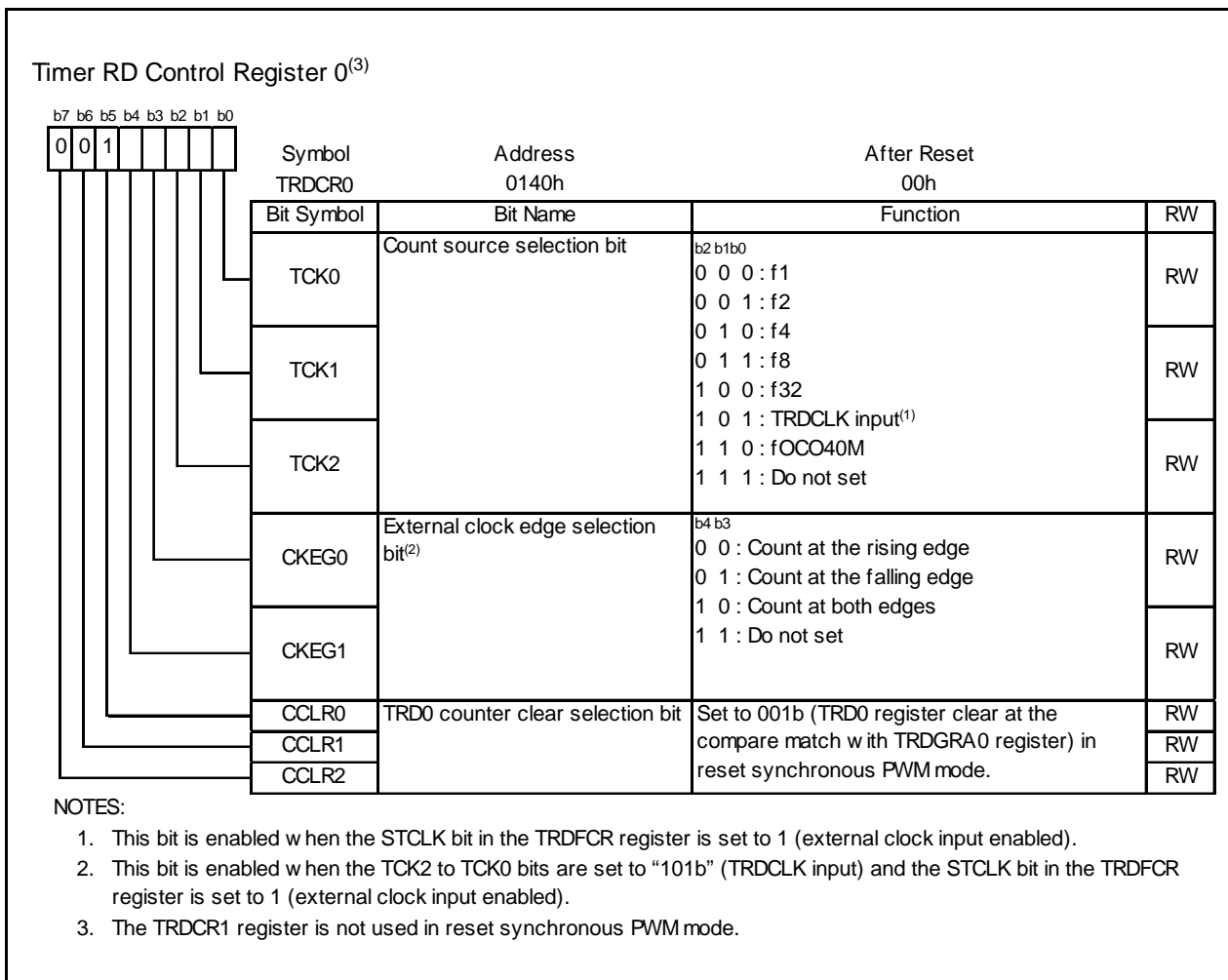


Figure 14.80 TRDCR0 Register in Reset Synchronous PWM Mode

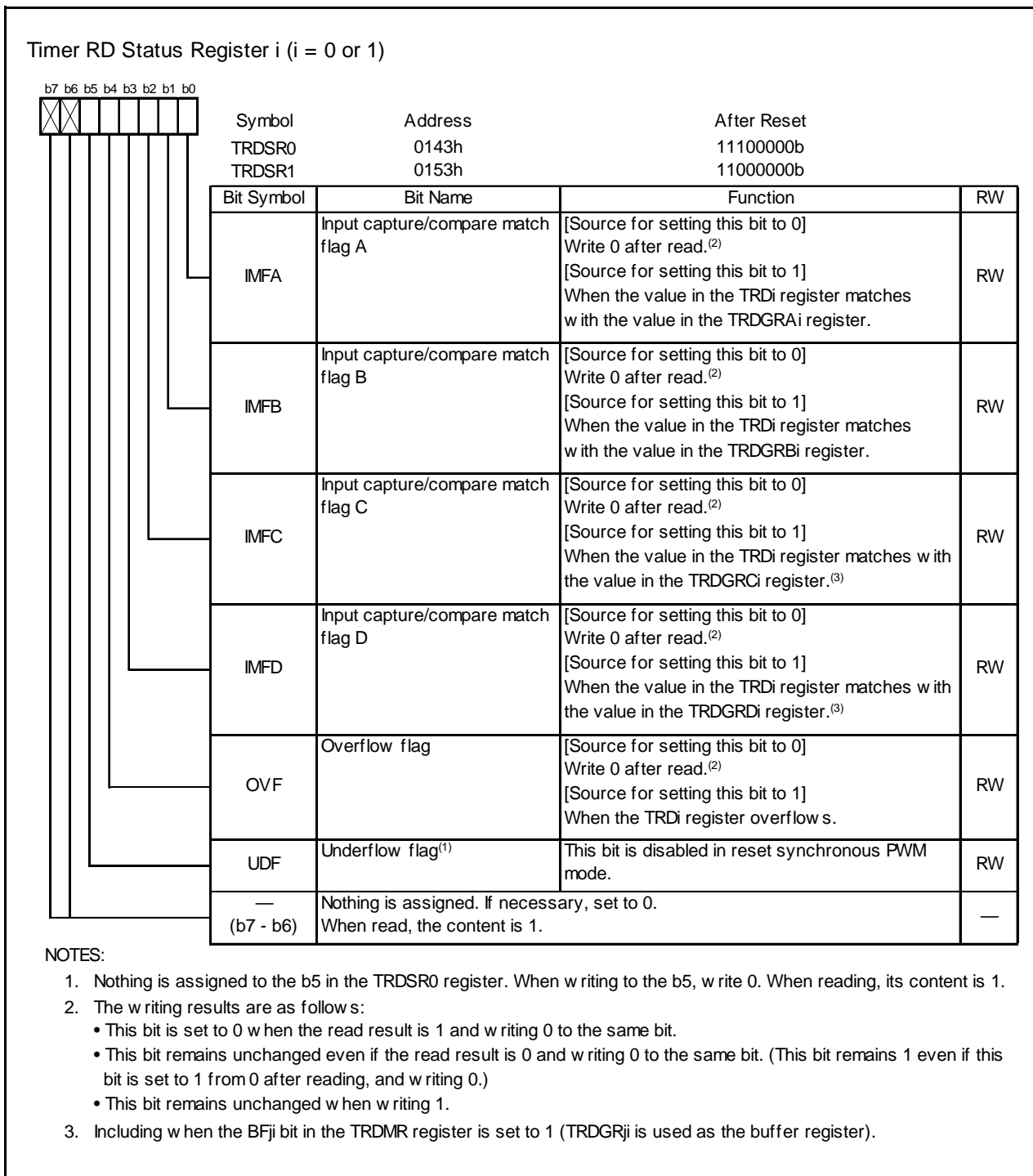


Figure 14.81 Registers TRDSR0 to TRDSR1 in Reset Synchronous PWM Mode

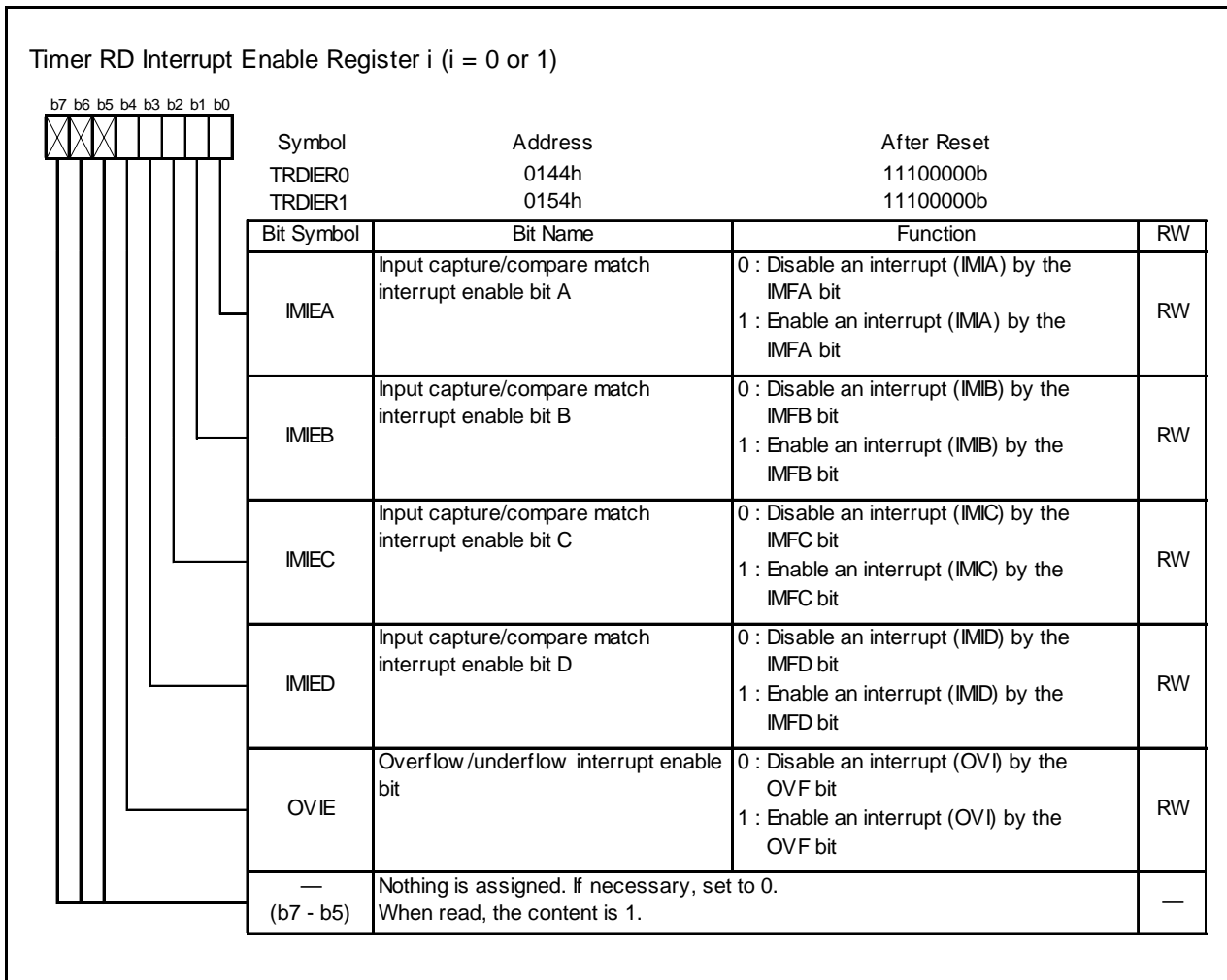


Figure 14.82 Registers TRDIER0 to TRDIER1 in Reset Synchronous PWM Mode

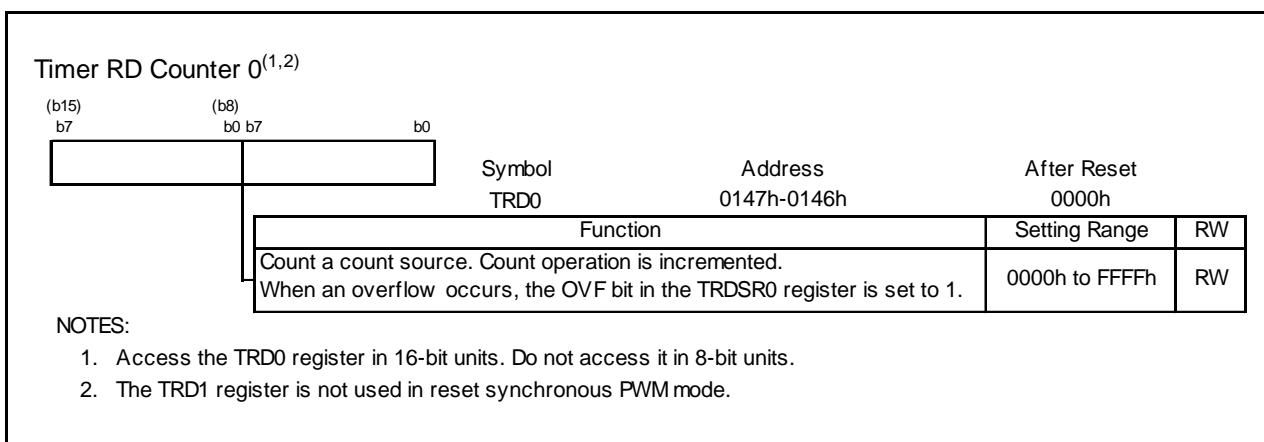


Figure 14.83 TRD0 Registrar in Reset Synchronous PWM Mode

Timer RD General Register Ai, Bi, Ci and Di (i = 0 or 1) <sup>(1)</sup>		Symbol	Address	After Reset
(b15) b7	(b8) b0 b7			
		TRDGRA0	0149h-0148h	FFFFh
		TRDGRB0	014Bh-014Ah	FFFFh
		TRDGRC0	014Dh-014Ch	FFFFh
		TRDGRD0	014Fh-014Eh	FFFFh
		TRDGRA1	0159h-0158h	FFFFh
		TRDGRB1	015Bh-015Ah	FFFFh
		TRDGRC1	015Dh-015Ch	FFFFh
		TRDGRD1	015Fh-015Eh	FFFFh
Function				RW
Refer to <b>Table 14.30 TRDGRji Register Functions in Reset Synchronous PWM Mode</b>				RW

NOTE:

1. Access the TRDGRAi to TRDGRDi registers in 16-bit units. Do not access them in 8-bit units.

**Figure 14.84 Registers TRDGRAi, TRDGRBi, TRDGRCi and TRDGRDi in Reset Synchronous PWM Mode**

The following registers are disabled in the reset synchronous PWM mode:

TRDPMR, TRDOCR, TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDPOCR0, TRDIORA1, TRDIORC1 and TRDPOCR1

**Table 14.30 TRDGRji Register Functions in Reset Synchronous PWM Mode**

Register	Setting	Register Function	PWM Output Pin
TRDGRA0	–	General register. Set the PWM period.	(Output inverted every period of TRDIOC0 and PWM pins)
TRDGRB0	–	General register. Set the changing point of PWM1 output.	TRDIOB0 TRDIOD0
TRDGRC0	BFC0 = 0	(These registers are not used in reset synchronous PWM mode.)	–
TRDGRD0	BFD0 = 0		
TRDGRA1	–	General register. Set the changing point of PWM2 output.	TRDIOA1 TRDIOC1
TRDGRB1	–	General register. Set the changing point of PWM3 output.	TRDIOB1 TRDIOD1
TRDGRC1	BFC1 = 0	(These points are not used in reset synchronous PWM mode.)	–
TRDGRD1	BFD1 = 0		
TRDGRC0	BFC0 = 1	Buffer register. Set the next PWM period. (Refer to <b>14.3.2 Buffer Operation</b> )	(Output inverted every period of TRDIOC0 and PWM pins)
TRDGRD0	BFD0 = 1	Buffer register. Set the changing point of the next PWM1 output. (Refer to <b>14.3.2 Buffer Operation</b> )	TRDIOB0 TRDIOD0
TRDGRC1	BFC1 = 1	Buffer register. Set the changing point of the next PWM2 output. (Refer to <b>14.3.2 Buffer Operation</b> )	TRDIOA1 TRDIOC1
TRDGRD1	BFD1 = 1	Buffer register. Set the changing point of the next PWM3 output. (Refer to <b>14.3.2 Buffer Operation</b> )	TRDIOB1 TRDIOD1

BFC0, BFD0, BFC1, BFD1: Bits in TRDMR Register

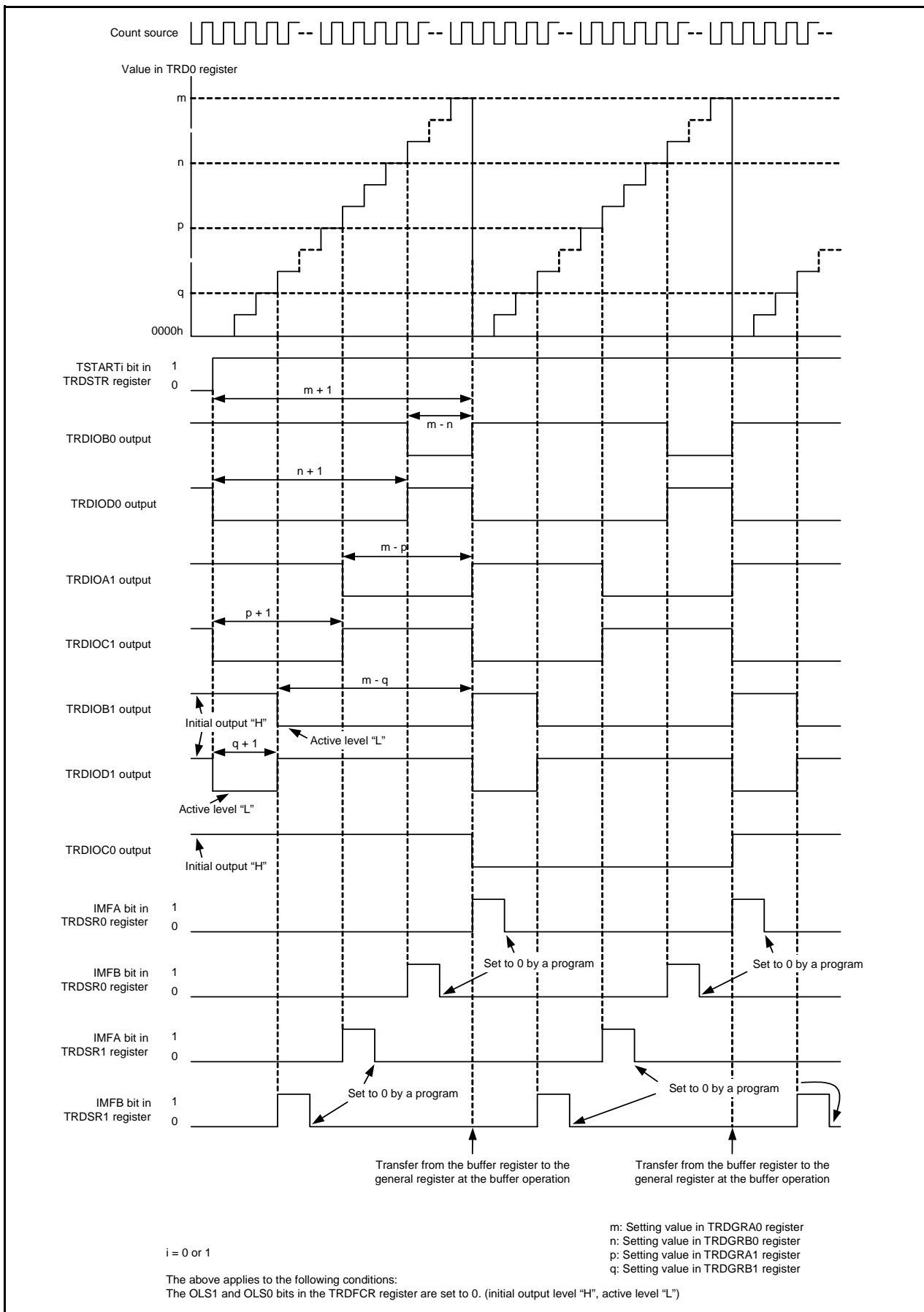


Figure 14.85 Operating Example of Reset Synchronous PWM Mode

### 14.3.9 Complementary PWM Mode

Output 3 normal-phases and 3 counter-phases of the PWM waveform with the same period (with three-phase, triangular wave modulation and dead time).

Figure 14.86 shows the Block Diagram of Complementary PWM Mode, Table 14.31 lists the Complementary PWM Mode Specifications. Figures 14.87 to 14.95 show the Registers Associated with Complementary PWM Mode, Figure 14.96 shows the Output Model of Complementary PWM Mode and Figure 14.97 shows the Operating Example of Complementary PWM Mode.

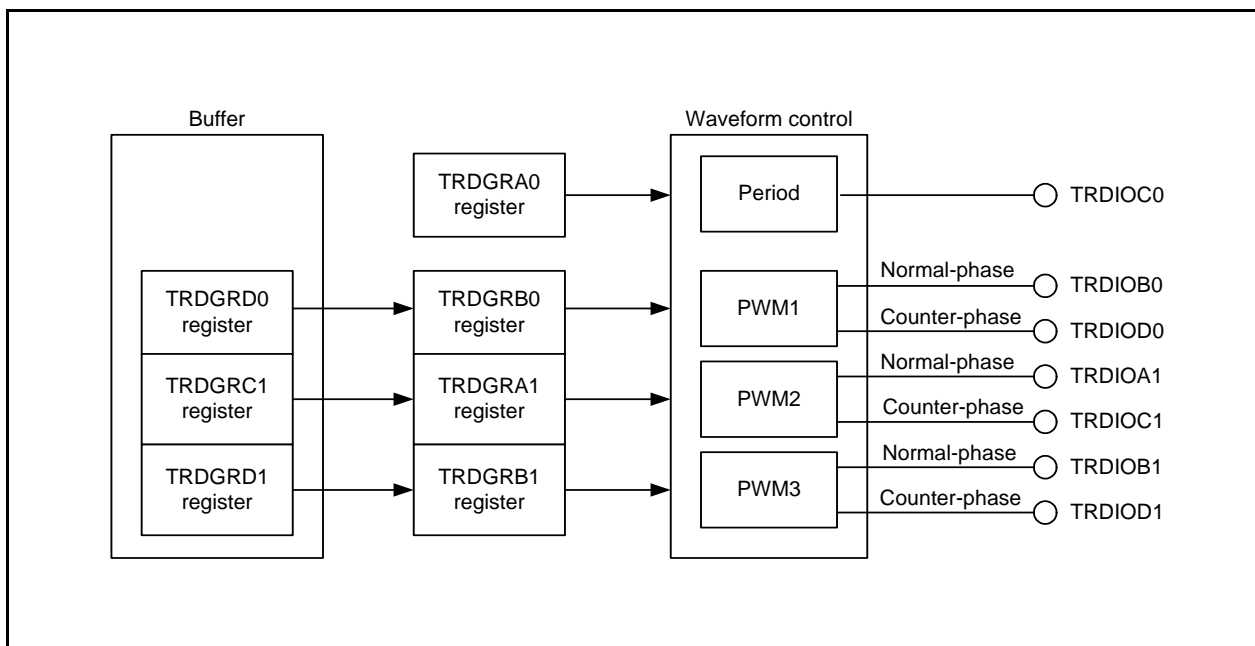


Figure 14.86 Block Diagram of Complementary PWM Mode

**Table 14.31 Complementary PWM Mode Specifications**

Item	Specification
Count Sources	f1, f2, f4, f8, f32, fOCO40M External signal input to the TRDCLK pin (valid edge selected by a program) Set the TCK2 to TCK0 bits in the TRDCR1 register to the same value (same count source) as the TCK2 to TCK0 bits in the TRDCR0 register.
Count Operations	Increment or decrement The TRD0 and TRD1 registers are decremented with the compare match in the TRD0 and TRDGRA0 registers during increment. The TRD1 register is set from 0000h to FFFFh during decrement, the TRD0 and TRD1 registers are incremented.
PWM Operations	PWM period: $1/fk \times (m + 2 - p) \times 2^{(1)}$ Dead time: p Active level width of normal-phase: $1/fk \times (m - n - p + 1) \times 2$ Active level width of counter-phase: $1/fk \times (n + 1 - p) \times 2$ fk: Frequency of count source m: Setting value in the TRDGRA0 register n: Setting value in the TRDGRB0 register (PWM output 1) Setting value in the TRDGRA1 register (PWM output 2) Setting value in the TRDGRB1 register (PWM output 3) p: Setting value in the TRD0 register 
Count Start Condition	Write 1 (count starts) to the TSTART0 and TSTART1 bits in the TRDSTR register.
Count Stop Conditions	Write 0 (count stops) to the TSTART0 and TSTART1 bits in the TRDSTR register when the CSEL0 bit in the TRDSTR register is set to 1. (The PWM output pin holds output level before the count stops.)
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>Compare match (the content in the TRDi register matches with the content in the TRDGRji register.)</li> <li>The TRD1 register undeflows</li> </ul>
TRDIOA0 Pin Function	Programmable I/O port or TRDCLK (external clock) input
TRDIOB0 Pin Function	PWM output 1 normal-phase output
TRDIOD0 Pin Function	PWM output 1 counter-phase output
TRDIOA1 Pin Function	PWM output 2 normal-phase output
TRDIOC1 Pin Function	PWM output 2 counter-phase output
TRDIOB1 Pin Function	PWM output 3 normal-phase output
TRDIOD1 Pin Function	PWM output 3 counter-phase output
TRDIOC0 Pin Function	Output inversed every 1/2 period of PWM
INT0 Pin Function	Programmable I/O port, pulse output forced cutoff signal input or INT0 interrupt input
Read from Timer	The count value can be read by reading the TRDi register.
Write to Timer	The value can be written to the TRDi register.
Selection Functions	<ul style="list-style-type: none"> <li>Pulse output forced cutoff signal input (refer to <b>14.3.4 Pulse Output Forced Cutoff</b>)</li> <li>The active level of normal-phase and counter-phase and initial output level selected individually.</li> <li>Transfer timing from the buffer register selected</li> <li>A/D trigger generated</li> </ul>

i = 0 or 1, j = either A, B, C or D

NOTE:

- After a count starts, the PWM period is stable.



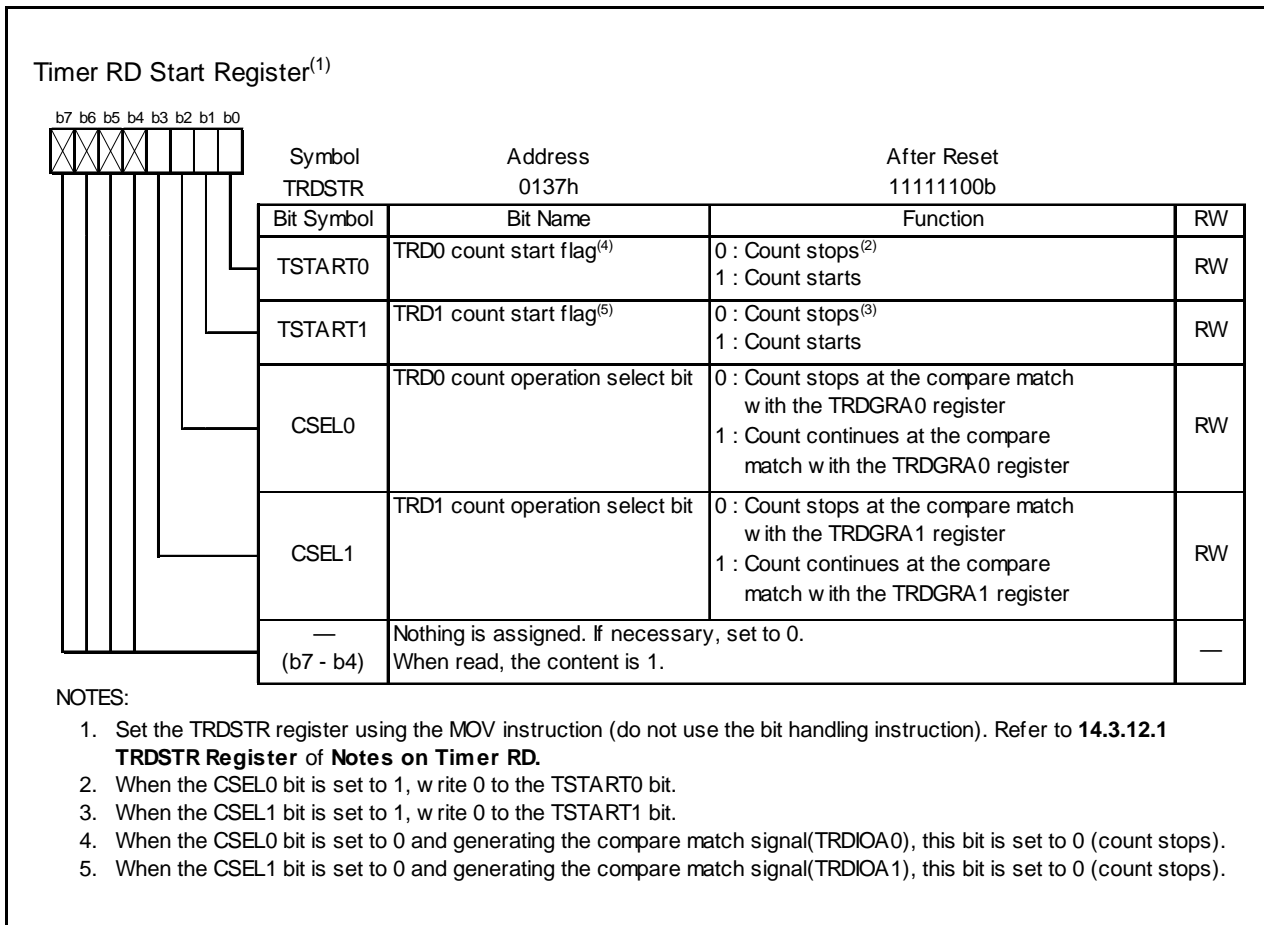


Figure 14.87 TRDSTR Register in Complementary PWM Mode

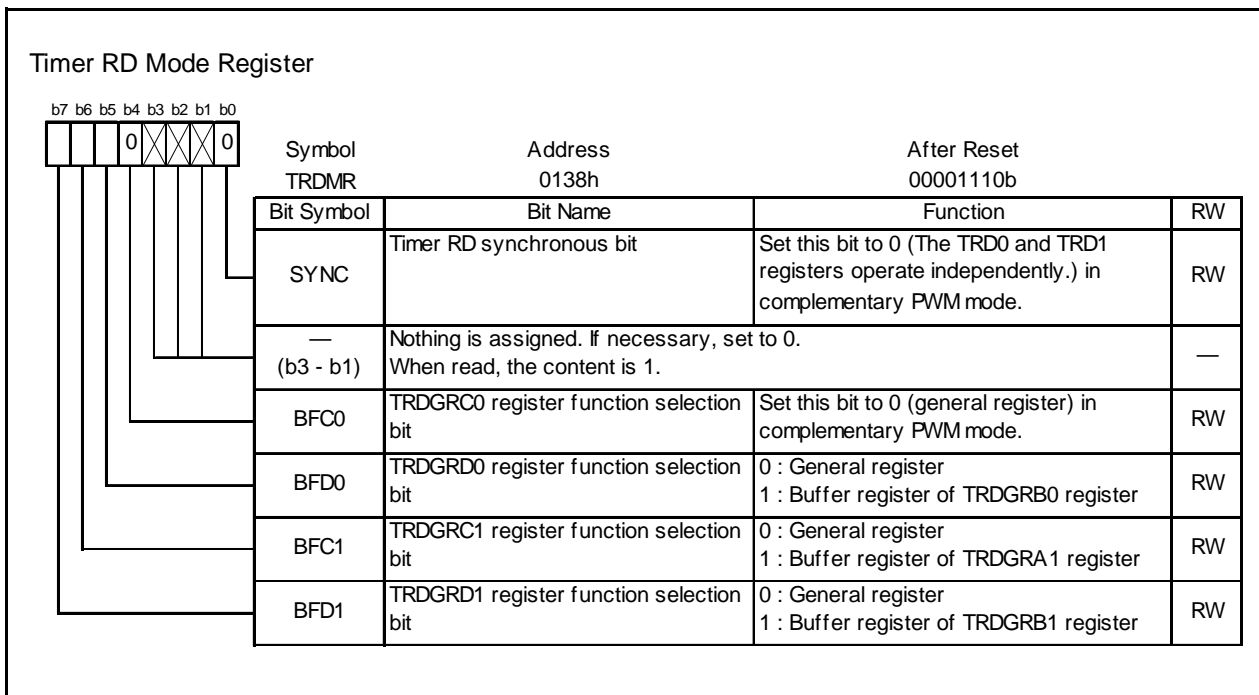


Figure 14.88 TRDMR Register in Complementary PWM Mode

Timer RD Function Control Register

Symbol	Address	After Reset	
TRDFCR	013Ah	1000000b	
Bit Symbol	Bit Name	Function	RW
CMD0	Combination mode selection bit <sup>(1,2)</sup>	b1 b0 1 0 : Complementary PWM mode (transfer from the buffer register to the general register at the underflow in the TRD1 register.)  1 1 : Complementary PWM mode (transfer from the buffer register to the general register at the compare match with the TRD0 and TRDGRA0 registers.)  Other than above : Do not set	RW
			RW
OLS0	Normal-phase output level selection bit (in reset synchronous PWM mode or complementary PWM mode)	0 : Initial output "H" Active level "L" 1 : Initial output "L" Active level "H"	RW
OLS1	Counter-phase output level selection bit (in reset synchronous PWM mode or complementary PWM mode)	0 : Initial output "H" Active level "L" 1 : Initial output "L" Active level "H"	RW
ADTRG	A/D trigger enable bit (in complementary PWM mode)	0 : Disable A/D trigger 1 : Enable A/D trigger <sup>(3)</sup>	RW
ADEG	A/D trigger edge selection bit (in complementary PWM mode)	0 : A/D trigger is generated at the compare match in the TRD0 and TRDGRA0 register 1 : A/D trigger is generated at the underflow in the TRD1 register	RW
STCLK	External clock input selection bit	0 : External clock input disabled 1 : External clock input enabled	RW
PWM3	PWM3 mode selection bit <sup>(4)</sup>	This bit is disabled in complementary PWM mode.	RW

NOTES:

- When setting bits CMD1 to CMD0 to 10b or 11b, the MCU enters complementary PWM mode in spite of the setting of the TRDPMR register.
- Set bits CMD1 to CMD0 when both the TSTART0 and TSTART1 bits in the TRDSTR register are set to 0 (count stops).
- Set the ADCAP bit in the ADCON0 register to 1 (starts by timer RD).
- When bits CMD1 to CMD0 are set to 00b (timer mode, PWM mode, or PWM3 mode), the setting of the PWM3 bit is enabled.

Figure 14.89 TRDFCR Register in Complementary PWM Mode

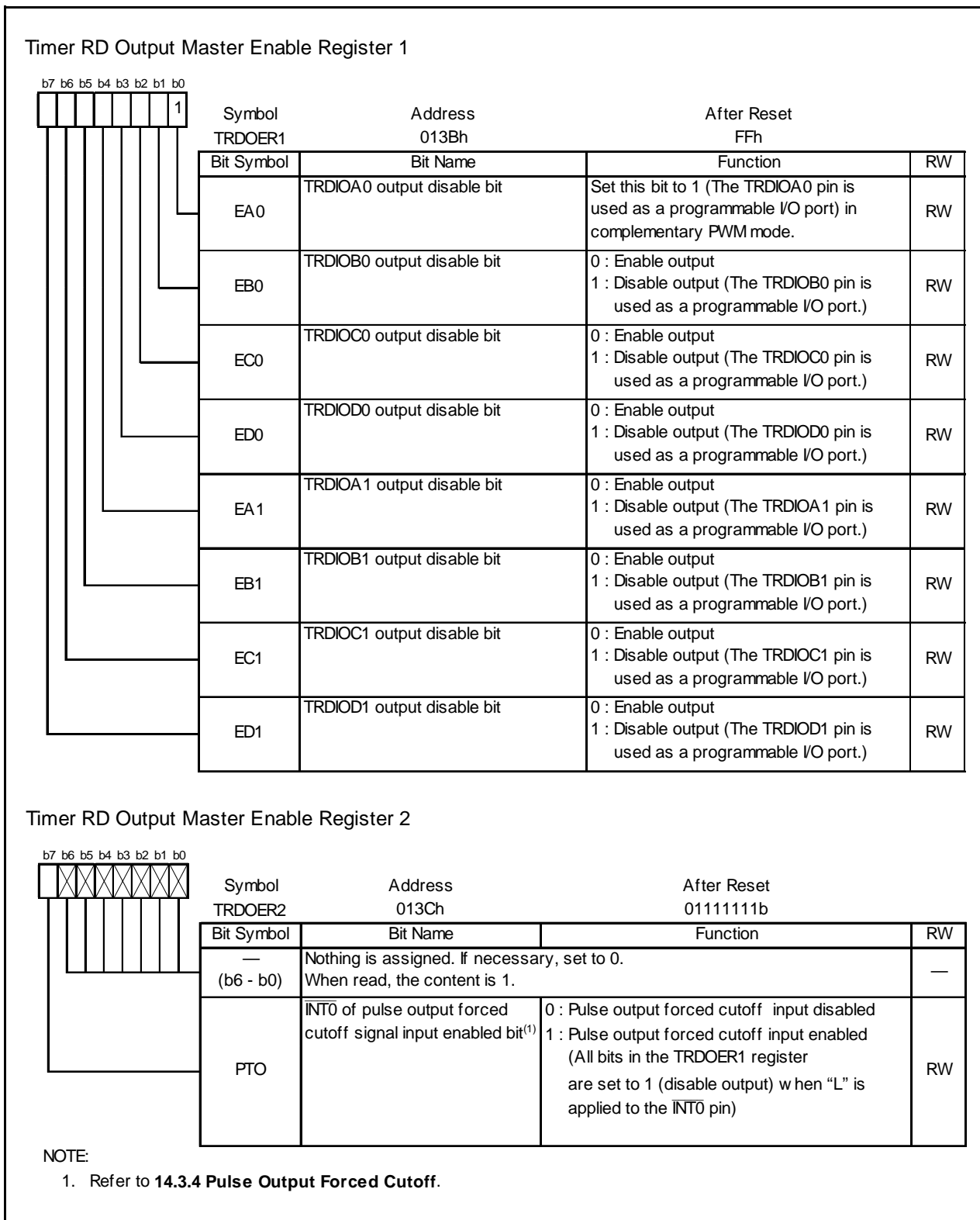


Figure 14.90 Registers TRDOER1 to TRDOER2 in Complementary PWM Mode

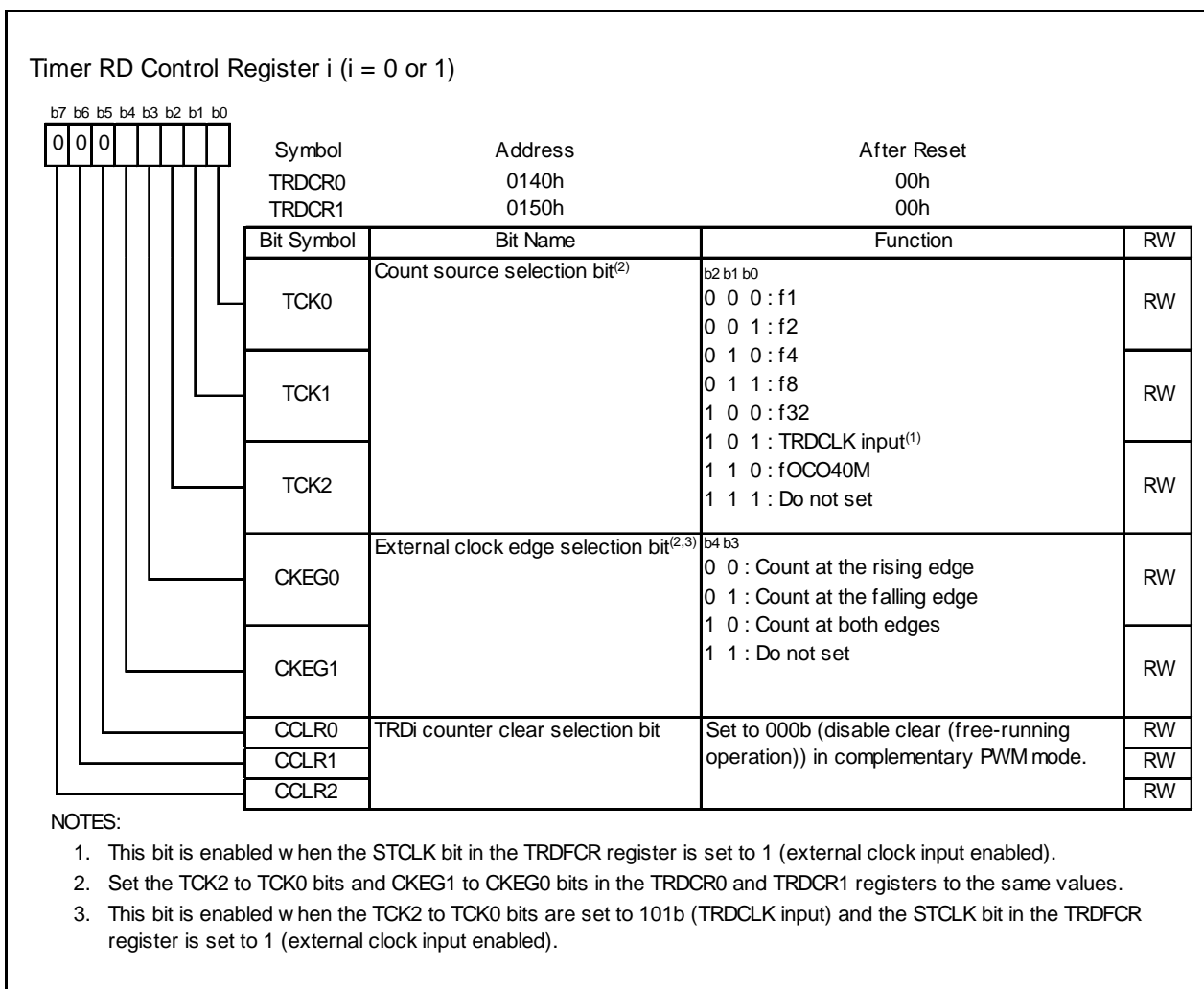


Figure 14.91 Registers TRDCR0 to TRDCR1 in Complementary PWM Mode

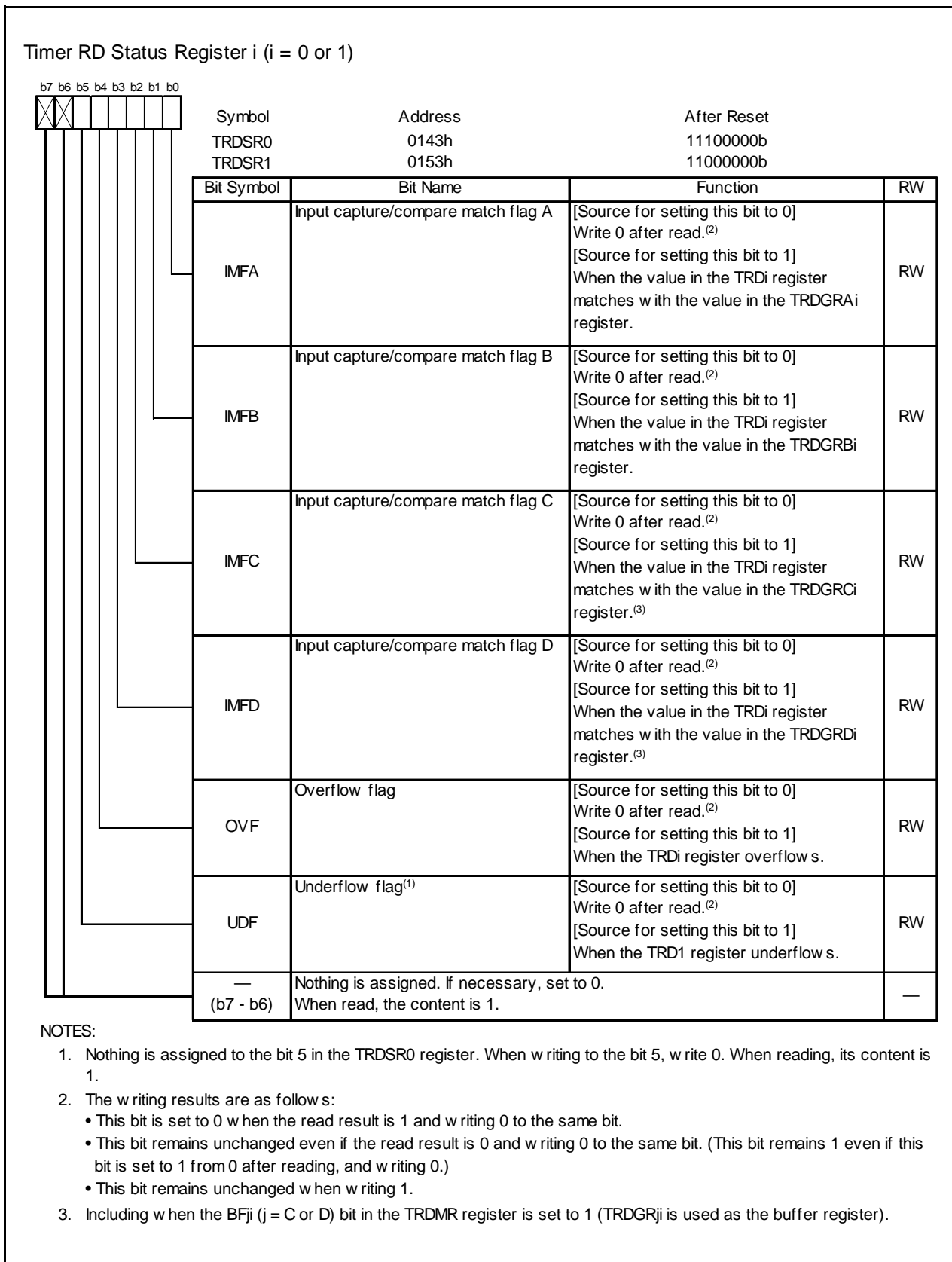


Figure 14.92 Registers TRDSR0 to TRDSR1 in Complementary PWM Mode

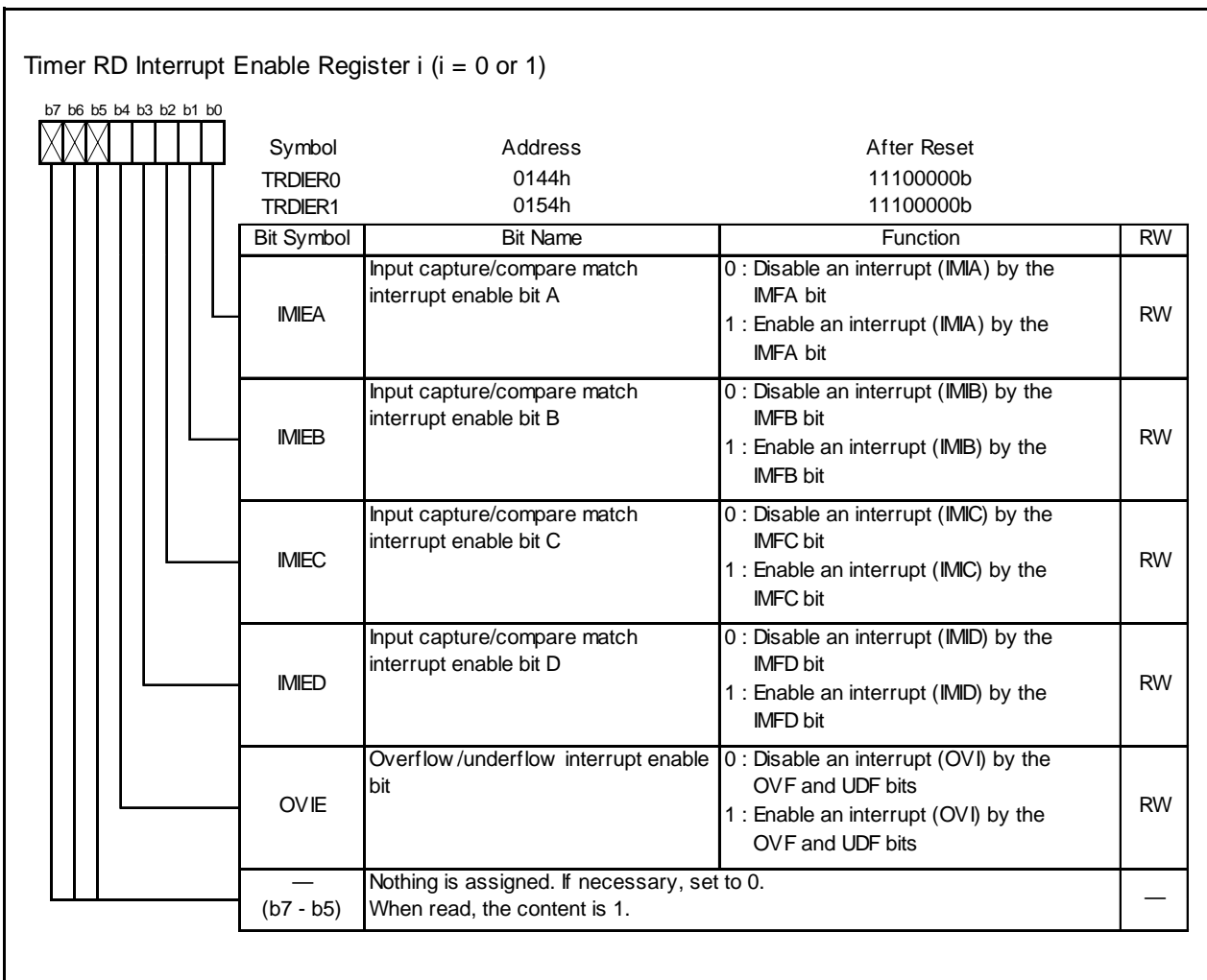
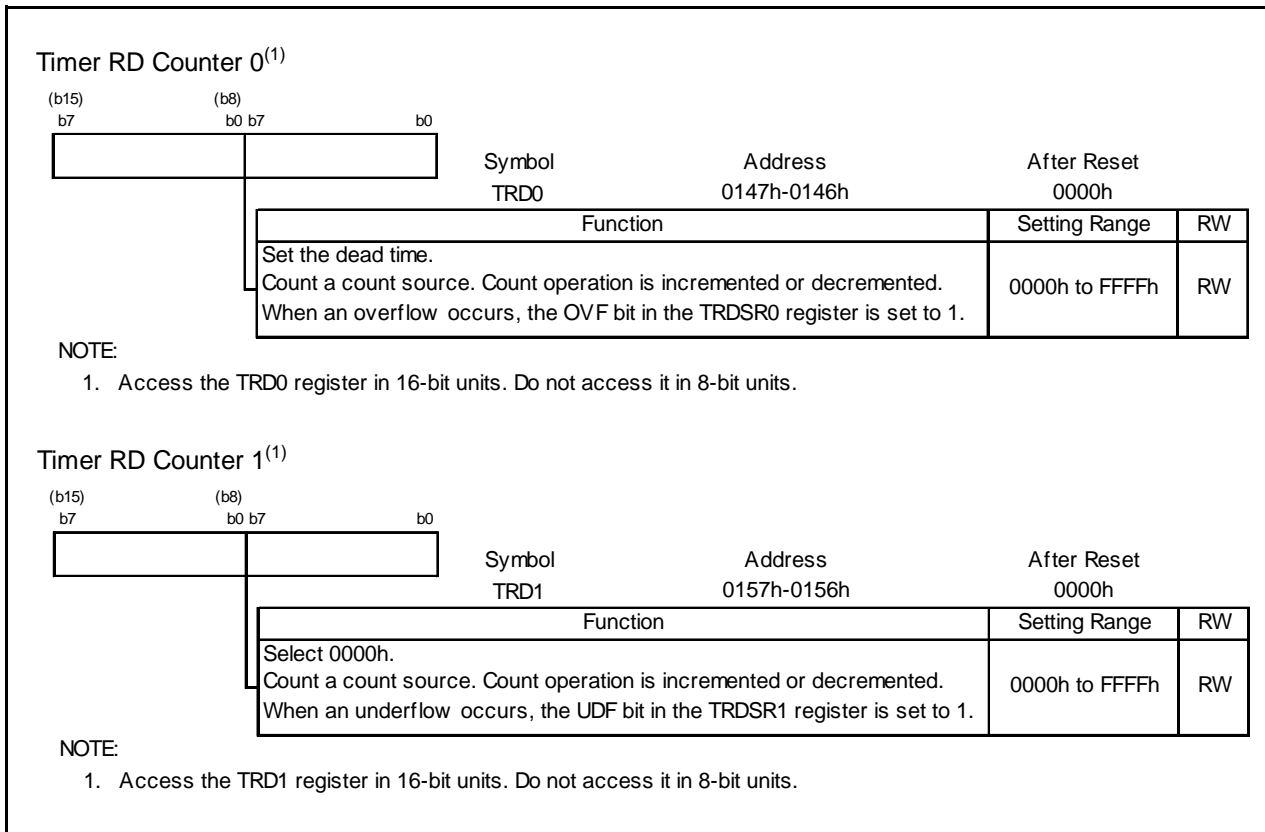
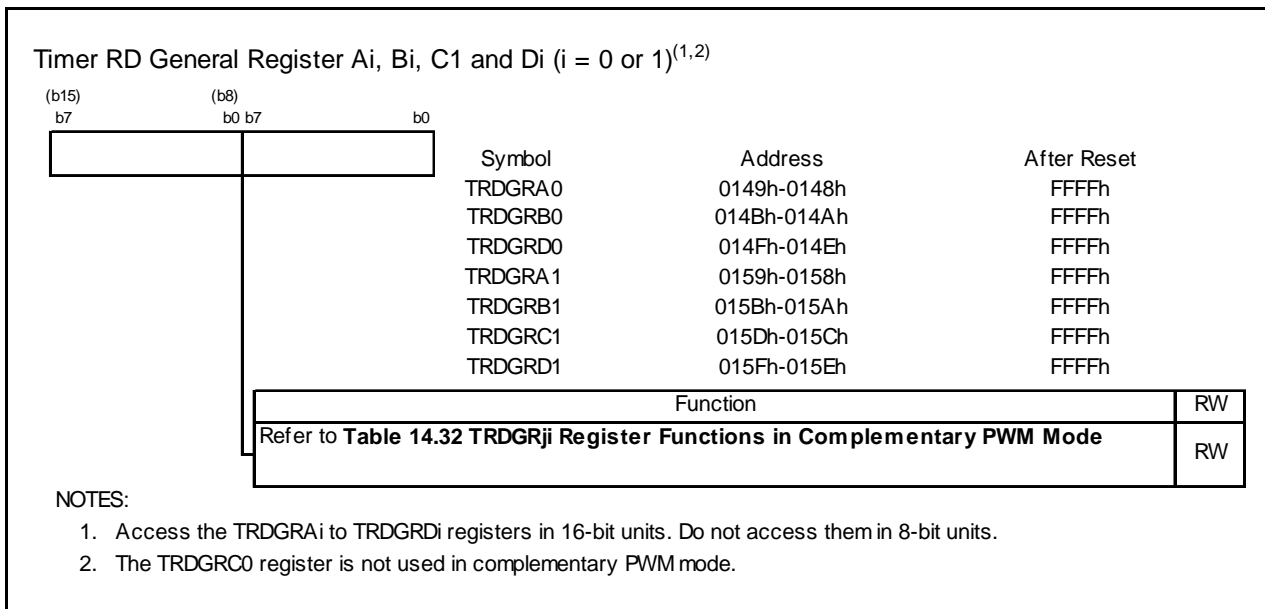


Figure 14.93 Registers TRDIER0 to TRDIER1 in Complementary PWM Mode



**Figure 14.94 Registers TRD0 to TRD1 in Complementary PWM Mode**



**Figure 14.95 Registers TRDGRAi, TRDGRBi, TRDGRC1 and TRDGRDi in Complementary PWM Mode**

The following registers are disabled in the complementary PWM mode:

TRDPMR, TRDOCR, TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDPOCR0, TRDIORA1, TRDIORC1 and TRDPOCR1

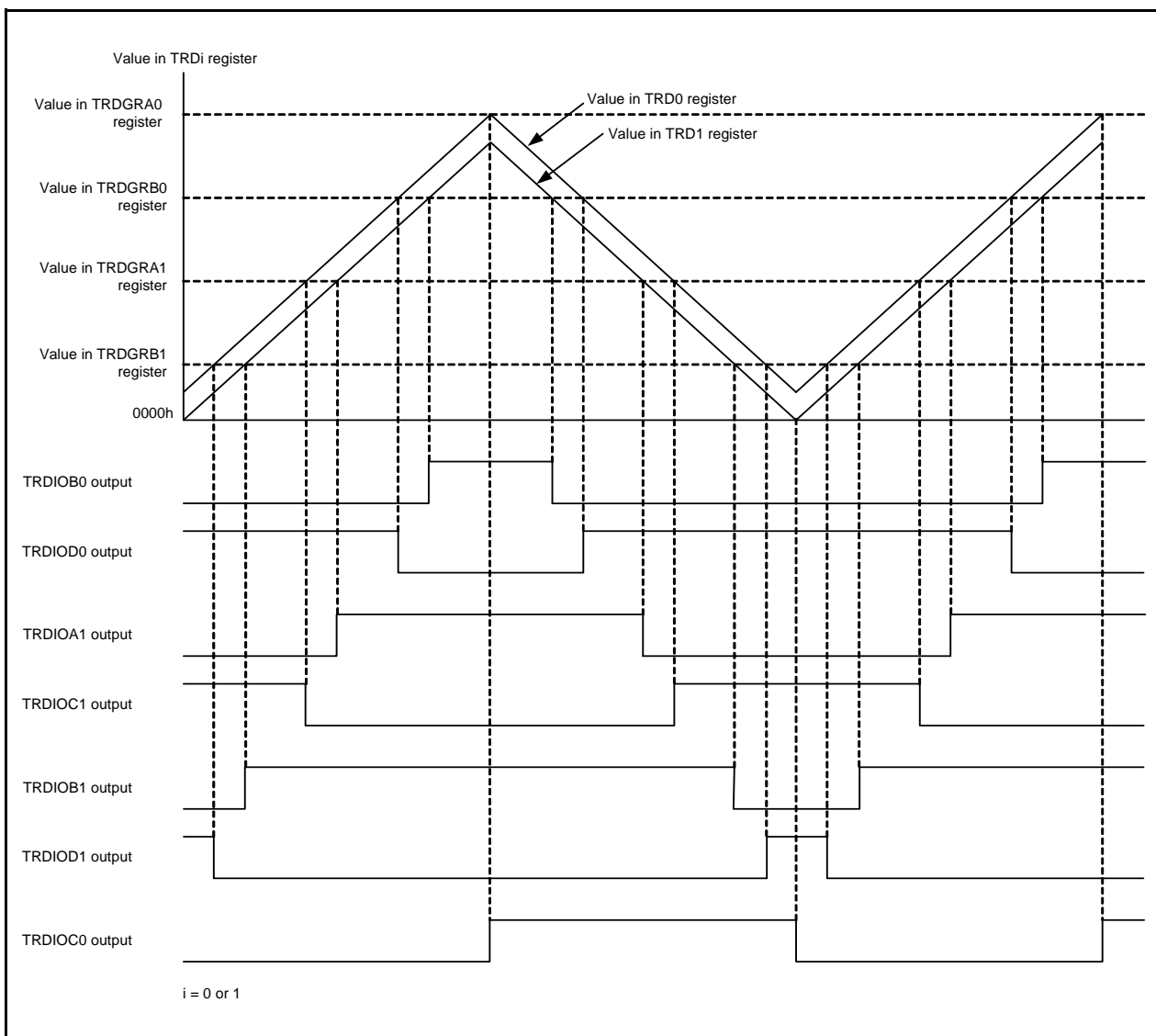


**Table 14.32 TRDGR<sub>ji</sub> Register Functions in Complementary PWM Mode**

Register	Setting	Register Function	PWM Output Pin
TRDGRA0	–	General register. Set the PWM period at initialization. Setting range: Setting value or above in TRD0 register FFFFh - TRD0 register setting value or below Do not write when the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts).	(Output inverted every half period of TRDIOC0 pin)
TRDGRB0	–	General register. Set the changing point of PWM1 output at initialization. Setting range: Setting value or above in TRD0 register TRDGRA0 register - TRD0 register setting value or below Do not write when the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts).	TRDIOB0 TRDIOD0
TRDGRA1	–	General register. Set the changing point of PWM2 output at initialization. Setting range: Setting value or above in TRD0 register TRDGRA0 register - TRD0 register setting value or below Do not write when the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts).	TRDIOA1 TRDIOC1
TRDGRB1	–	General register. Set the changing point of PWM3 output at initialization. Setting range: Setting value or above in TRD0 register TRDGRA0 register - TRD0 register setting value or below Do not write when the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts).	TRDIOB1 TRDIOD1
TRDGRC0	–	These registers not used in complementary PWM mode.	–
TRDGRD0	BFD0 = 1	Buffer register. Set the changing point of next PWM1 output. (Refer to <b>14.3.2 Buffer Operation</b> ) Setting range: Setting value or above in TRD0 register TRDGRA0 register - TRD0 register setting value or below Set this register to the same value as the TRDGRB0 register for the initialization.	TRDIOB0 TRDIOD0
TRDGRC1	BFC1 = 1	Buffer register. Set the changing point of next PWM2 output. (Refer to <b>14.3.2 Buffer Operation</b> ) Setting range: Setting value or above in TRD0 register TRDGRA0 register - TRD0 register setting value or below Set this register to the same value as the TRDGRA1 register for the initialization.	TRDIOA1 TRDIOC1
TRDGRD1	BFD1 = 1	Buffer register. Set the changing point of next PWM3 output. (Refer to <b>14.3.2 Buffer Operation</b> ) Setting range: Setting value or above in TRD0 register TRDGRA0 register - TRD0 register setting value or below Set this register to the same value as the TRDGRB1 register for the initialization.	TRDIOB1 TRDIOD1

BFC0, BFD0, BFC1, BFD1: Bits in TRDMR register

Since values cannot be written to the TRDGRB0, TRDGRA1, or TRDGRB1 register directly after count operation starts (prohibited item), use the TRDGRD0, TRDGRC1, or TRDGRD1 register as a buffer register. However, to write data to the TRDGRD0, TRDGRC1, or TRDGRD1 register, set bits BFD0, BFC1, and BFD1 to 0 (general register). After this, bits BFD0, BFC1, and BFD1 may be set to 1 (buffer register).



**Figure 14.96 Output Model of Complementary PWM Mode**

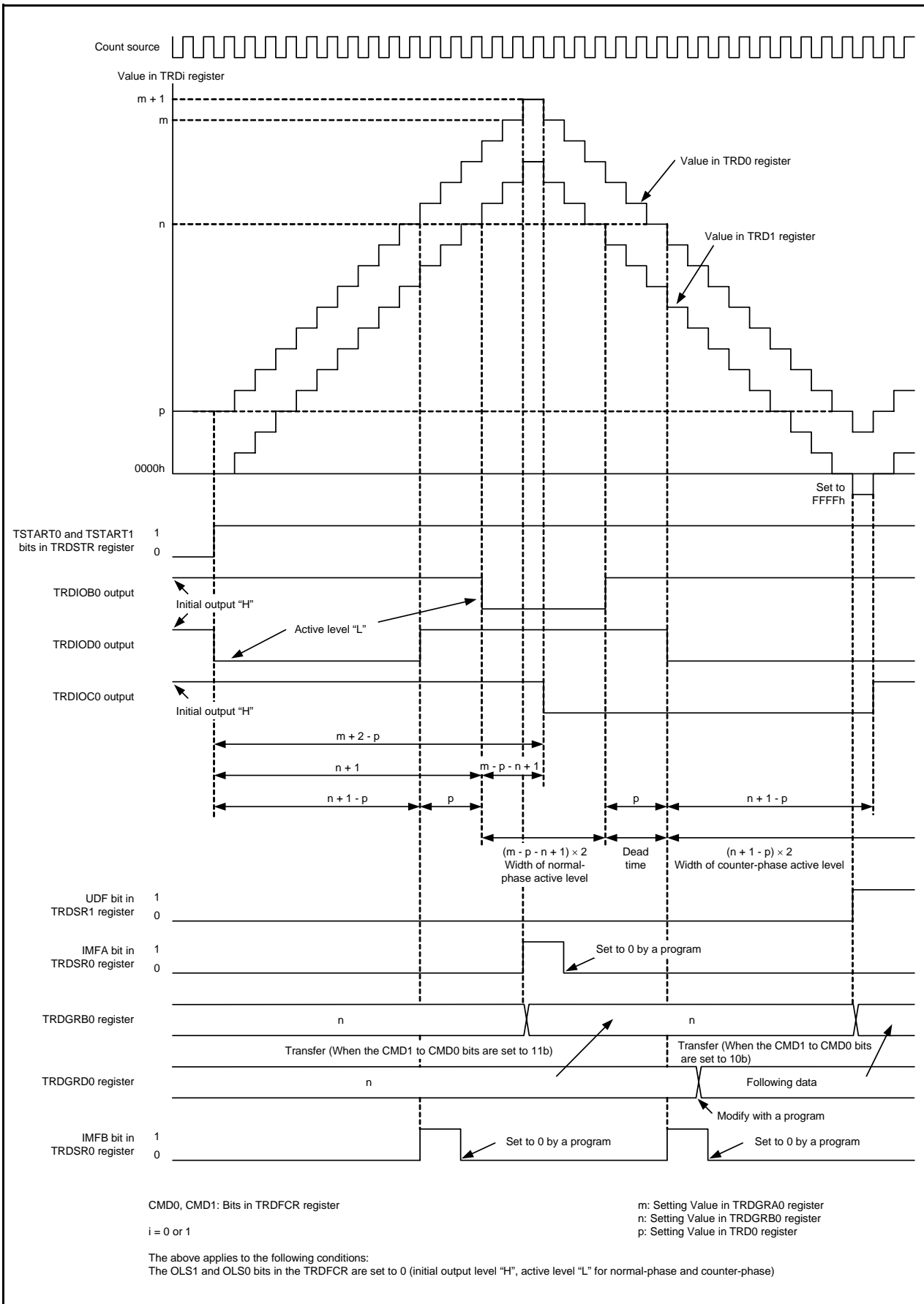


Figure 14.97 Operating Example of Complementary PWM Mode

### 14.3.9.1 Transfer Timing from Buffer Register

- Transfer from the TRDGRD0, TRDGRC1 and TRDGRD1 registers to the TRDGRB0, TRDGRA1 and TRDGRB1 registers

When the CMD1 to CMD0 bits in the TRDFCR register are set to 10b, the content is transferred when the TRD1 register underflows.

When the CMD1 to CMD0 bits are set to 11b, the content is transferred at the compare match in the TRD0 and TRDGRA0 registers.

### 14.3.9.2 A/D Trigger Generation

The compare match in the TRD0 and TRDGRA0 registers and the TRD1 underflow can be used as a conversion start trigger of the A/D converter. It can be selected by the ADEG and ADTRG bits in the TRDFCR register.

Also, set the ADCAP bit in the ADCON0 register to 1 (starts in Timer RD).

### 14.3.10 PWM3 Mode

Output 2 PWM waveforms with the same period.

Figure 14.98 shows the Block Diagram of PWM3 Mode, Table 14.33 lists the PWM3 Mode Specifications. Figures 14.99 to 14.107 show the Registers Associated with PWM3 Mode and Figure 14.108 shows the Operating Example of PWM3 Mode.

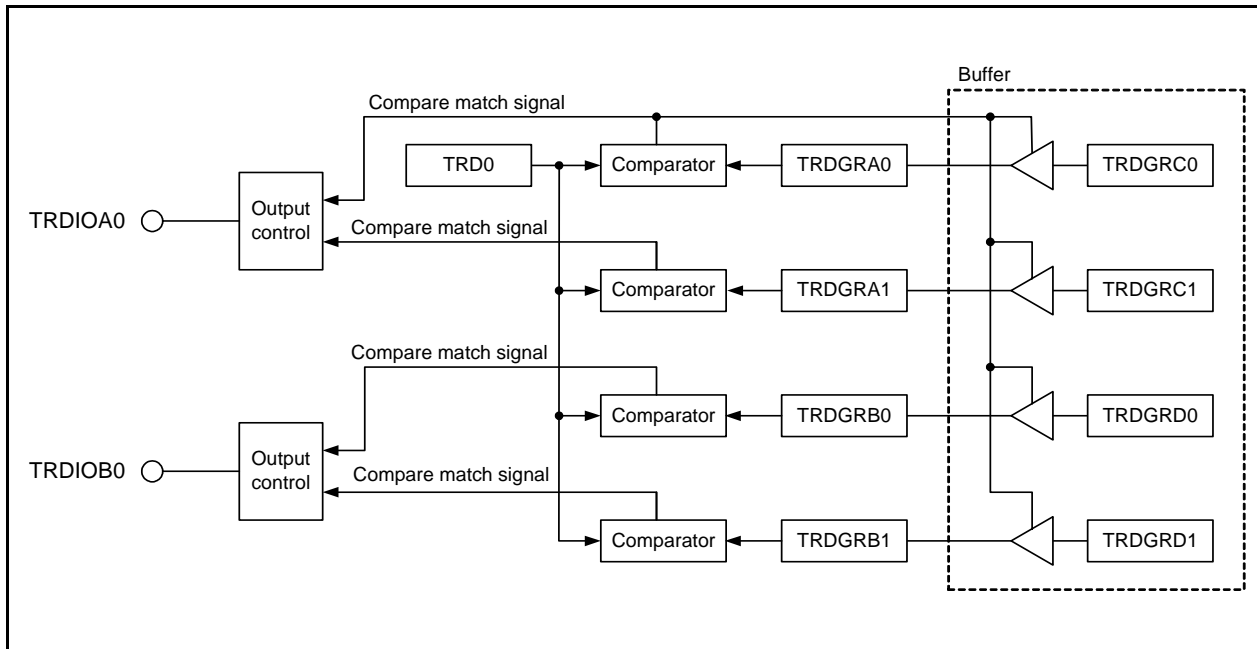


Figure 14.98 Block Diagram of PWM3 Mode

**Table 14.33 PWM3 Mode Specifications**

Item	Specification
Count Sources	f1, f2, f4, f8, f32, fOCO40M
Count Operations	The TRD0 register is incremented. (The TRD1 is not used.)
PWM Waveform	<p>PWM period: <math>1/f_k \times (m + 1)</math>  Active level width of TRDIOA0 output: <math>1/f_k \times (m - n)</math>  Active level width of TRDIOB0 output: <math>1/f_k \times (p - q)</math>  <math>f_k</math>: Frequency of count source  <math>m</math>: Setting value in the TRDGRA0 register  <math>n</math>: Setting value in the TRDGRA1 register  <math>p</math>: Setting value in the TRDGRB0 register  <math>q</math>: Setting value in the TRDGRB1 register</p> <p>(When "H" is selected for the active level)</p>
Count Start Condition	Write 1 (count starts) to the TSTART0 bit in the TRDSTR register.
Count Stop Conditions	<ul style="list-style-type: none"> <li>Write 0 (count stops) to the TSTART0 bit in the TRDSTR register when the CSEL0 bit in the TRDSTR register is set to 1.</li> <li>The PWM output pin holds output level before the count stops</li> <li>When the CSEL0 bit in the TRDSTR register is set to 0, the count stops at the compare match in the TRDGRA0 register.</li> <li>The PWM output pin holds level after output change by the compare match.</li> </ul>
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>Compare match (the content in the TRDi register matches with the content in the TRDGRji register.)</li> <li>The TRD0 register overflows</li> </ul>
TRDIOA0, TRDIOB0 Pin Functions	PWM output
TRDIOC0, TRDIOD0, TRDIOA1 to TRDIOD1 Pin Functions	Programmable I/O port
INT0 Pin Function	Programmable I/O port, pulse output forced cutoff signal input or INT0 interrupt input
Read from Timer	The count value can be read by reading the TRD0 register.
Write to Timer	The value can be written to the TRD0 register.
Selection Functions	<ul style="list-style-type: none"> <li>Pulse output forced cutoff signal input (refer to <b>14.3.4 Pulse Output Forced Cutoff</b>)</li> <li>Select the active level every pin.</li> <li>Buffer operation (refer to <b>14.3.2 Buffer Operation</b>)</li> </ul>

$i = 0$  or  $1$ ,  $j =$  either A, B, C or D

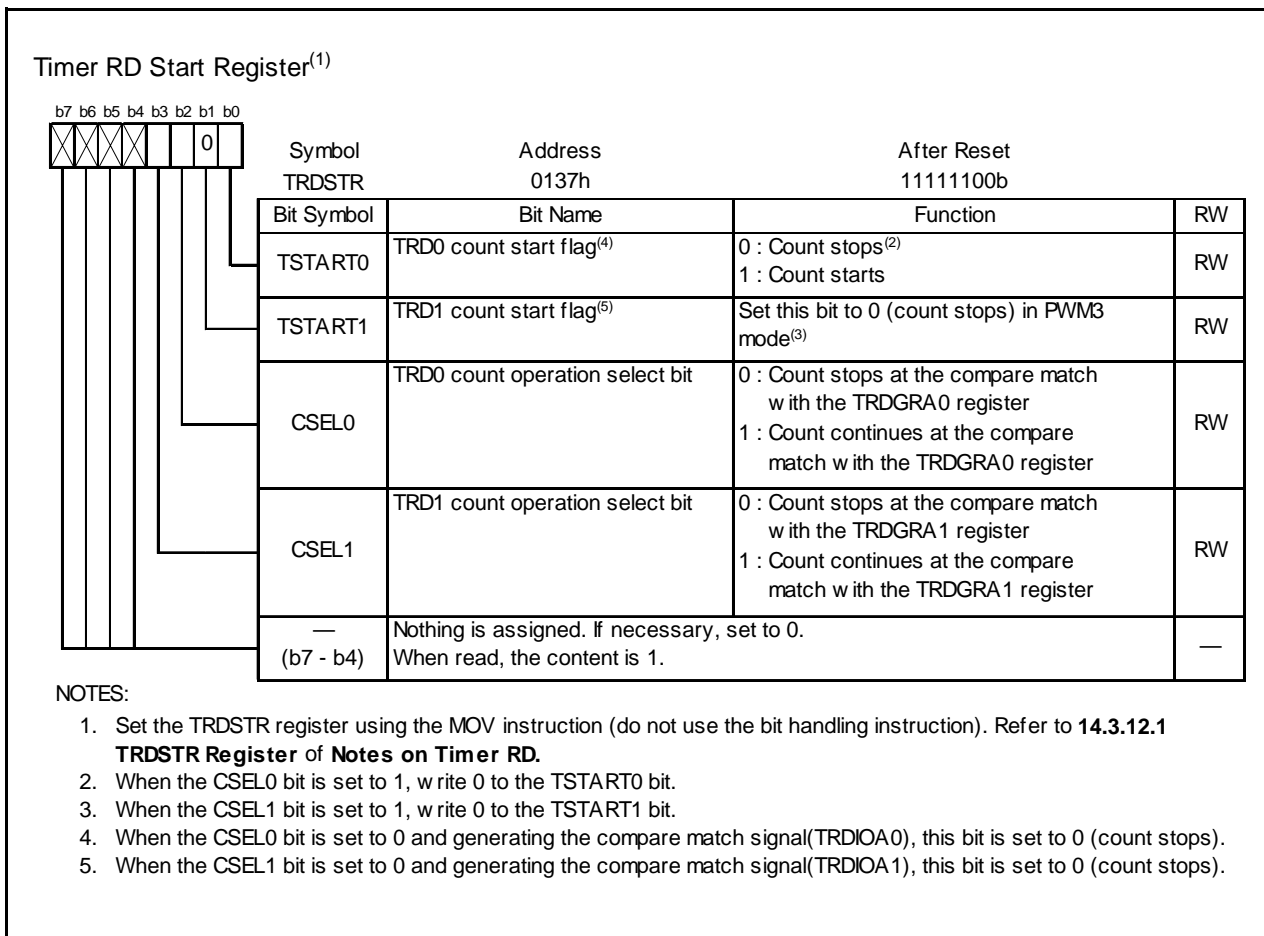


Figure 14.99 TRDSTR Register in PWM3 Mode

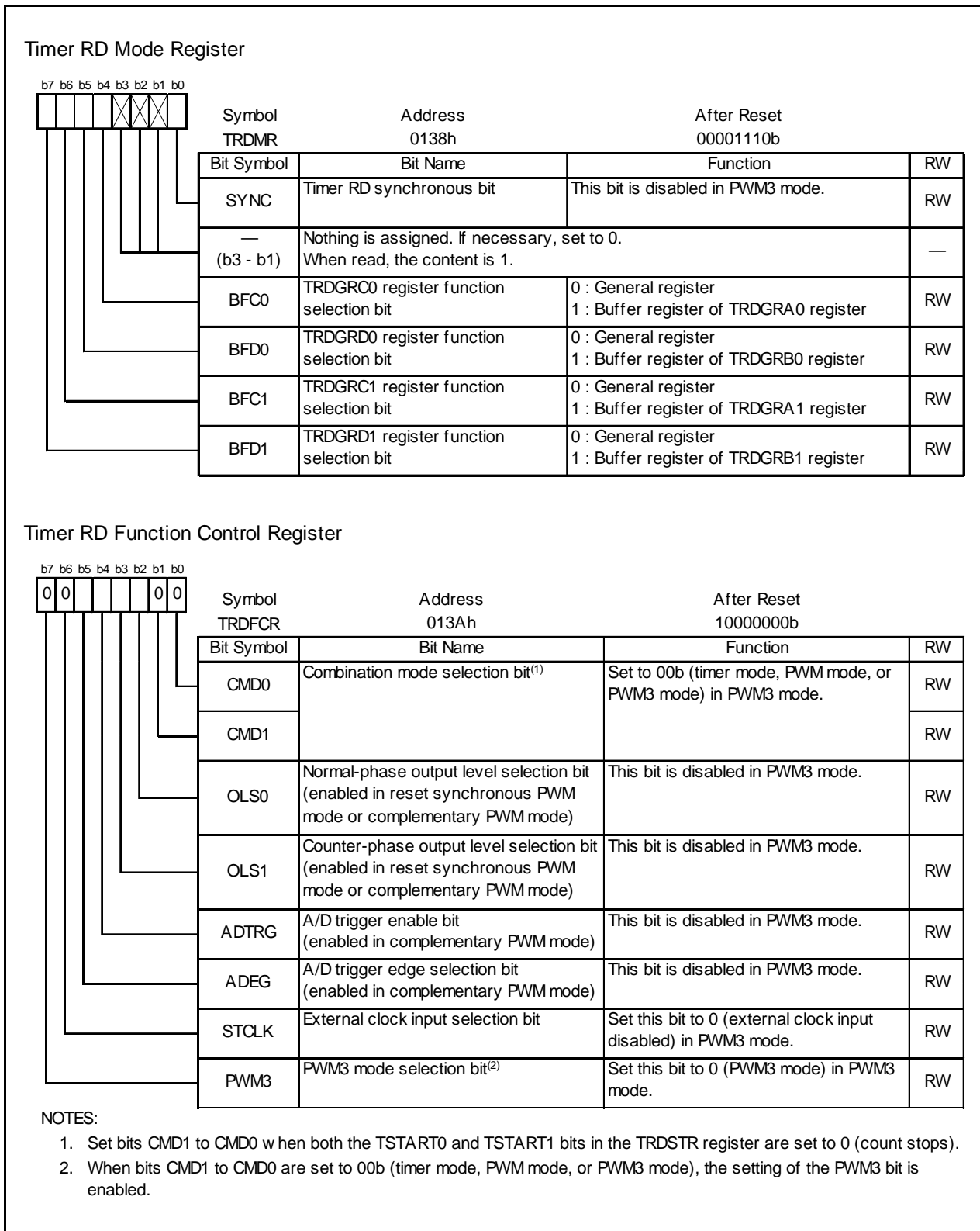


Figure 14.100 Registers TRDMR and TRDFCR in PWM3 Mode



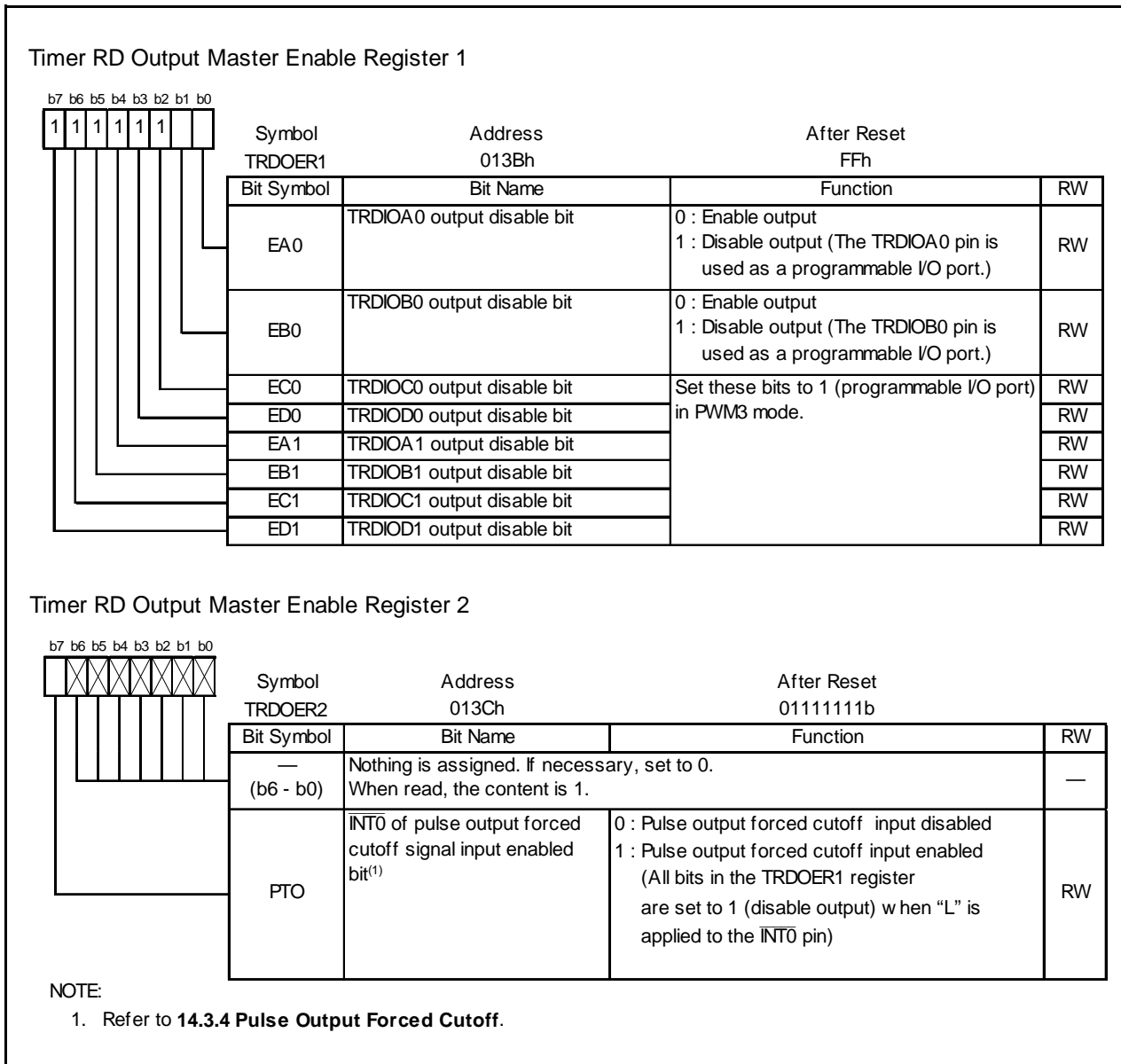


Figure 14.101 Registers TRDOER1 to TRDOER2 in PWM3 Mode

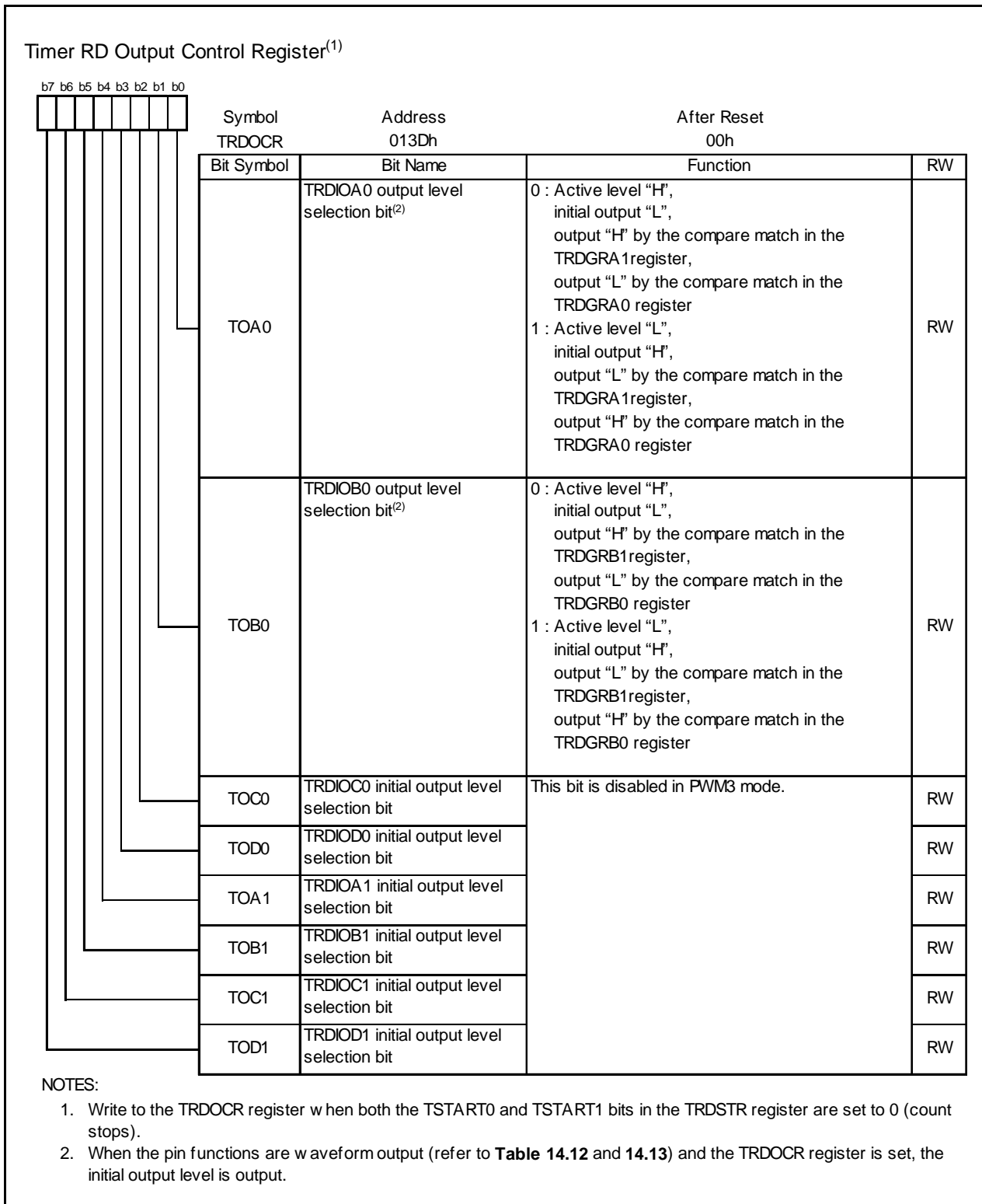


Figure 14.102 TRDOCR Register in PWM3 Mode

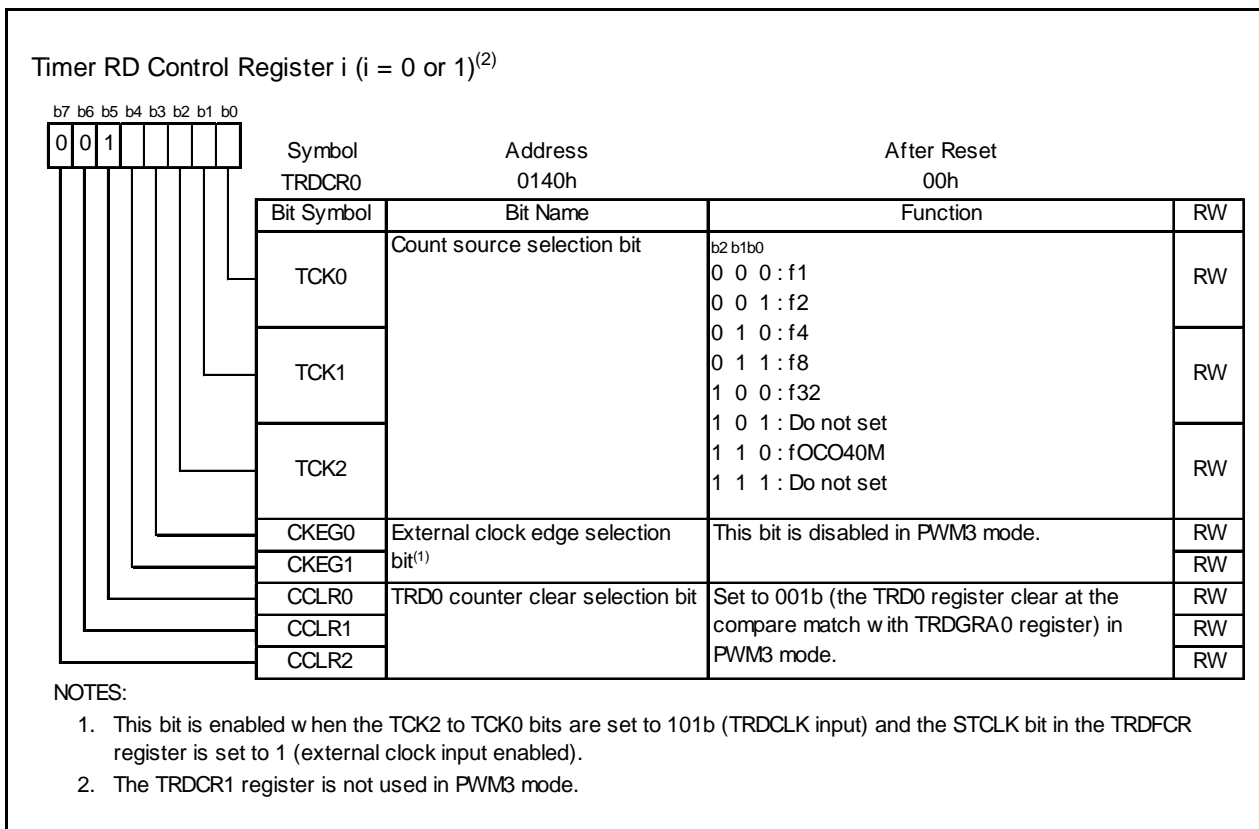


Figure 14.103 TRDCR0 Register in PWM3 Mode

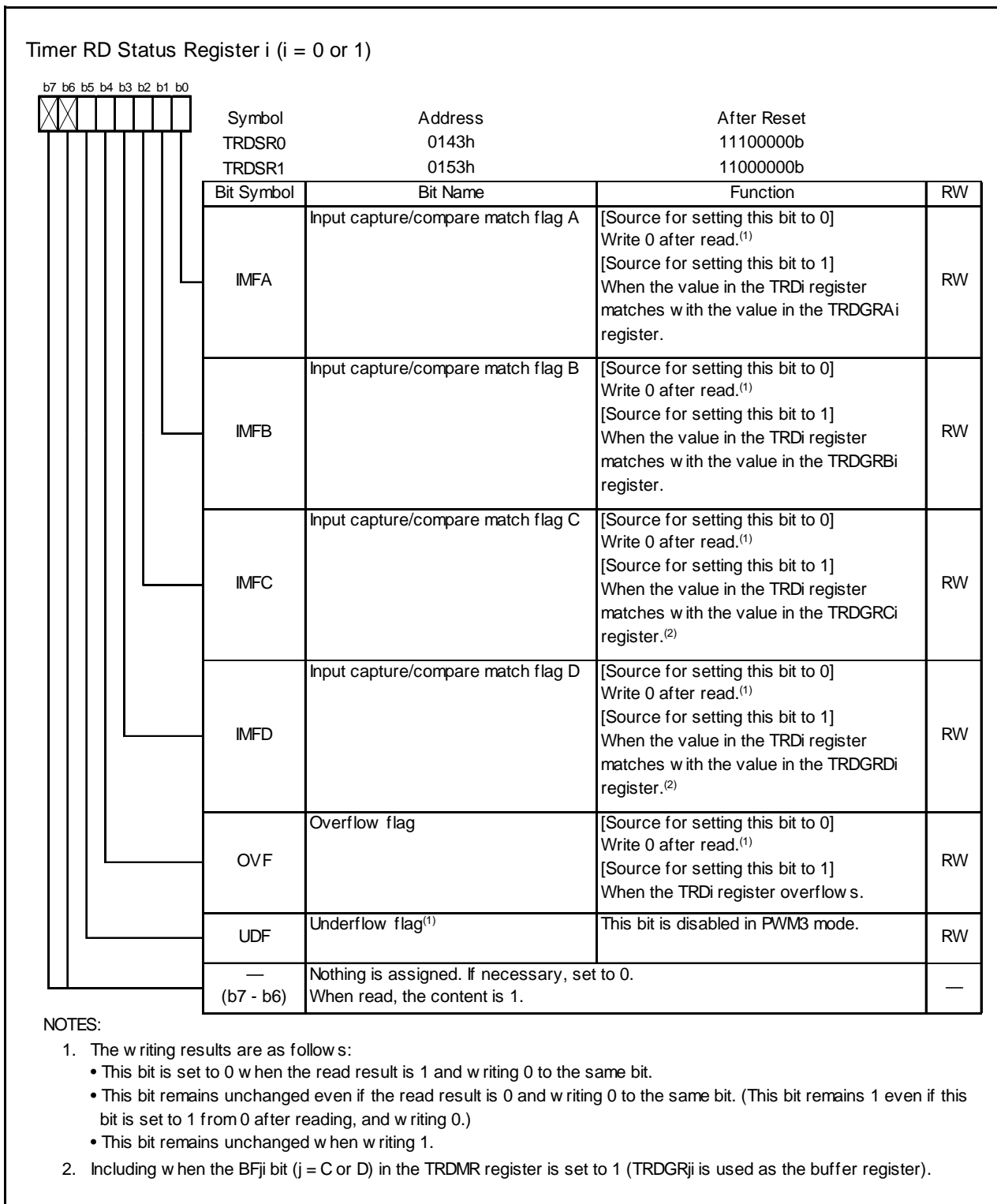


Figure 14.104 Registers TRDSR0 and TRDSR1 in PWM3 Mode

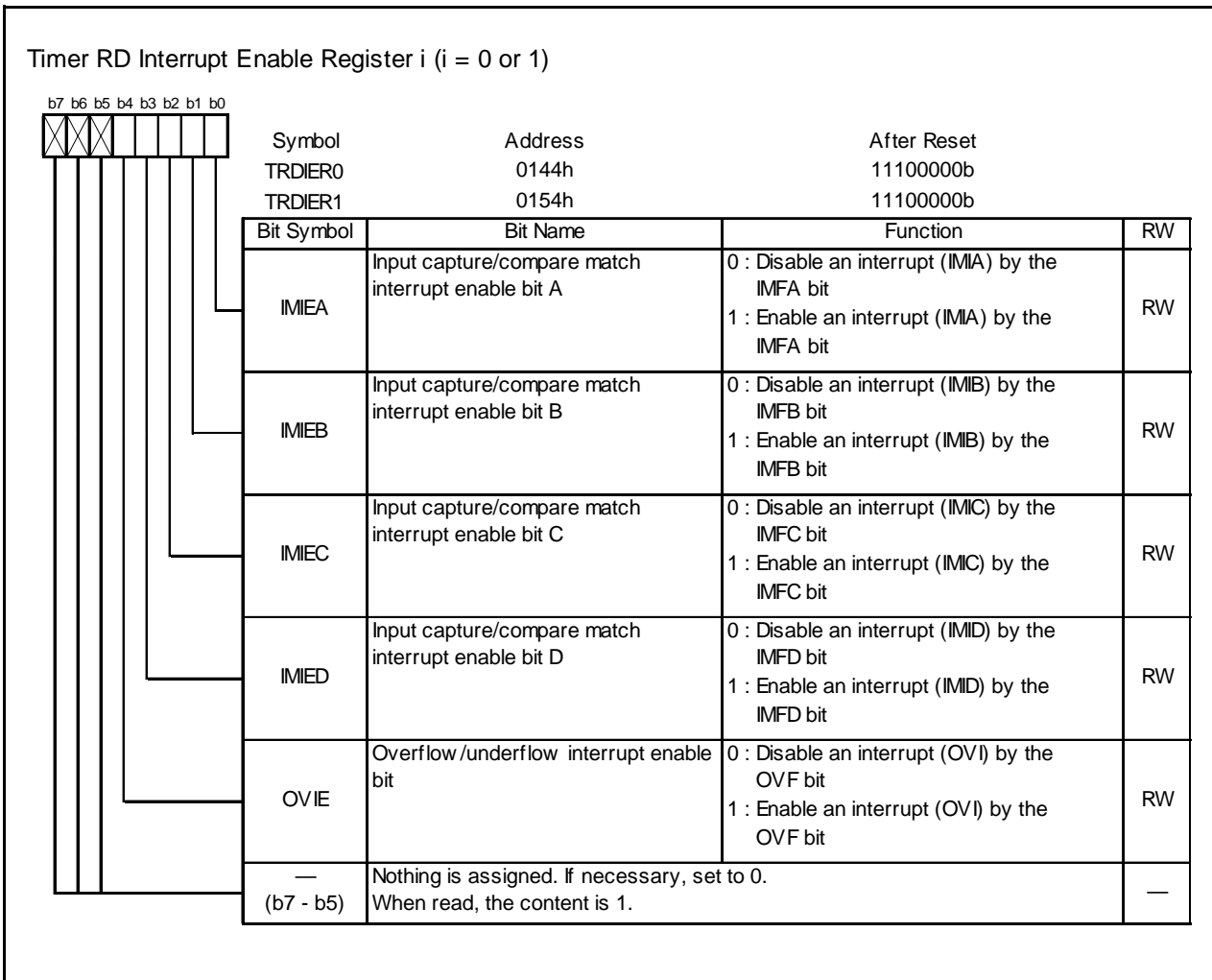


Figure 14.105 Registers TRDIER0 and TRDIER1 in PWM3 Mode

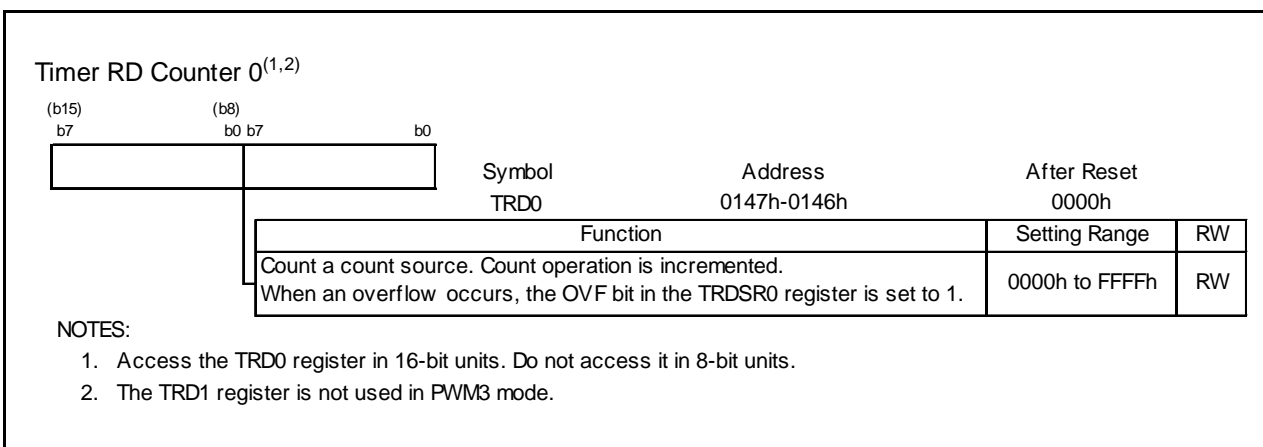
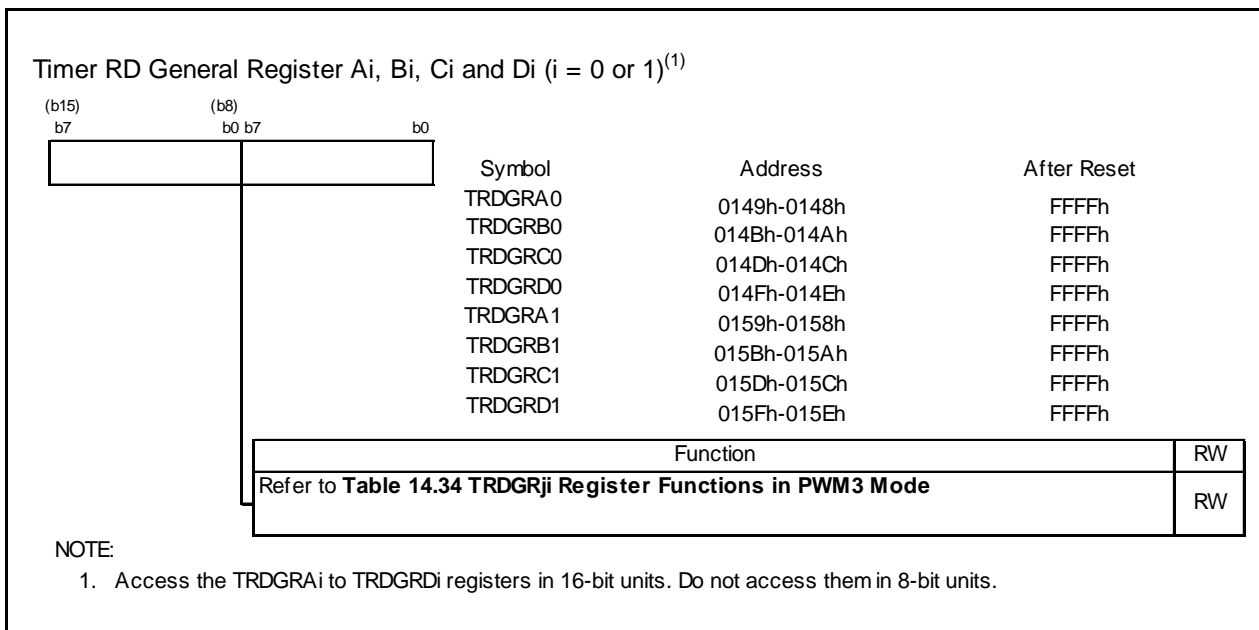


Figure 14.106 TRD0 Register in PWM3 Mode



**Figure 14.107 Registers TRDGRAi, TRDGRBi, TRDGRCi and TRDGRDi in PWM3 Mode**

The following registers are disabled in PWM3 mode:

TRDPMR, TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDPOCR0, TRDIORA1, TRDIORC1 and TRDPOCR1

**Table 14.34 TRDGR<sub>ji</sub> Register Functions in PWM3 Mode**

Register	Setting	Register Function	PWM Output Pin
TRDGRA0	–	General register. Set the PWM period. Setting range: Value set in TRDGRA1 register or above	TRDIOA0
TRDGRA1		General register. Set the changing point (the active level timing) of PWM output. Setting range: Value set in TRDGRA0 register or below	
TRDGRB0		General register. Set the changing point (the timing that returns to initial output level) of PWM output. Setting range: Value set in TRDGRB1 register or above Value set in TRDGRA0 register or below	TRDIOB0
TRDGRB1		General register. Set the changing point (active level timing) of PWM output. Setting range: Value set in TRDGRB0 register or below	
TRDGRC0	BFC0 = 0	(These registers are not used in PWM3 mode)	–
TRDGRC1	BFC1 = 0		
TRDGRD0	BFD0 = 0		
TRDGRD1	BFD1 = 0		
TRDGRC0	BFC0 = 1	Buffer register. Set the next PWM period. (Refer to <b>14.3.2 Buffer Operation</b> .) Setting range: Value set in TRDGRC1 register or above	TRDIOA0
TRDGRC1	BFC1 = 1	Buffer register. Set the changing point of next PWM output. (Refer to <b>14.3.2 Buffer Operation</b> .) Setting range: Value set in TRDGRC0 register or below	
TRDGRD0	BFD0 = 1	Buffer register. Set the changing point of next PWM output. (Refer to <b>14.3.2 Buffer Operation</b> .) Setting range: Value set in TRDGRD1 register or above, setting value or below in TRDGRC0 register.	TRDIOB0
TRDGRD1	BFD1 = 1	Buffer register. Set the changing point of next PWM output. (Refer to <b>14.3.2 Buffer Operation</b> .) Setting range: Value set in TRDGRD0 register or below	

BFC0, BFD0, BFC1, BFD1: Bits in TRDMR Register

Registers TRDGRC0, TRDGRC1, TRDGRD0, and TRDGRD1 are not used in PWM3 mode. To use them as buffer registers, set bits BFC0, BFC1, BFD0, and BFD1 to 0 (general register) and write a value to the TRDGRC0, TRDGRC1, TRDGRD0, or TRDGRD1 register. After this, bits BFC0, BFC1, BFD0, and BFD1 may be set to 1 (buffer register).

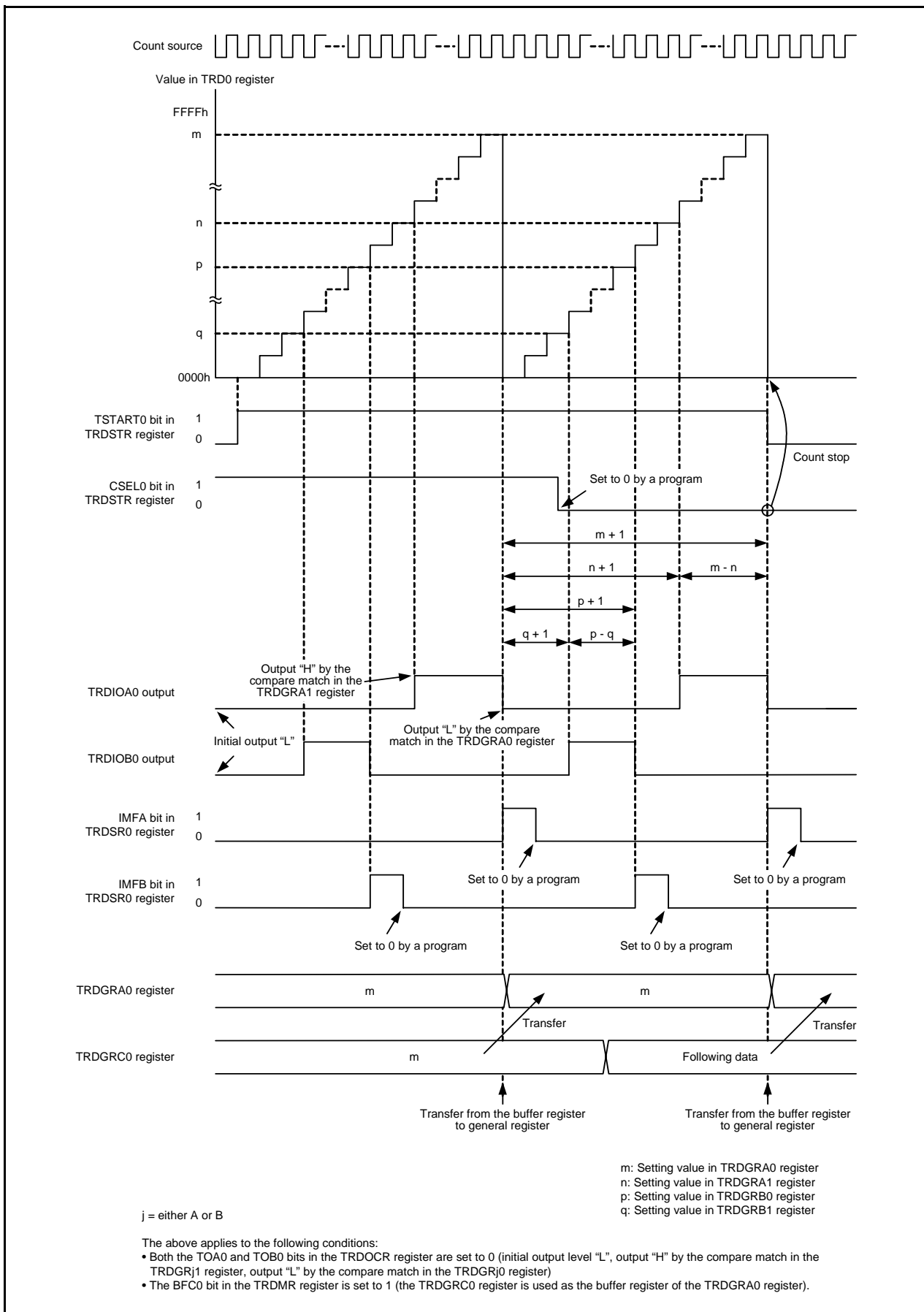


Figure 14.108 Operating Example of PWM3 Mode



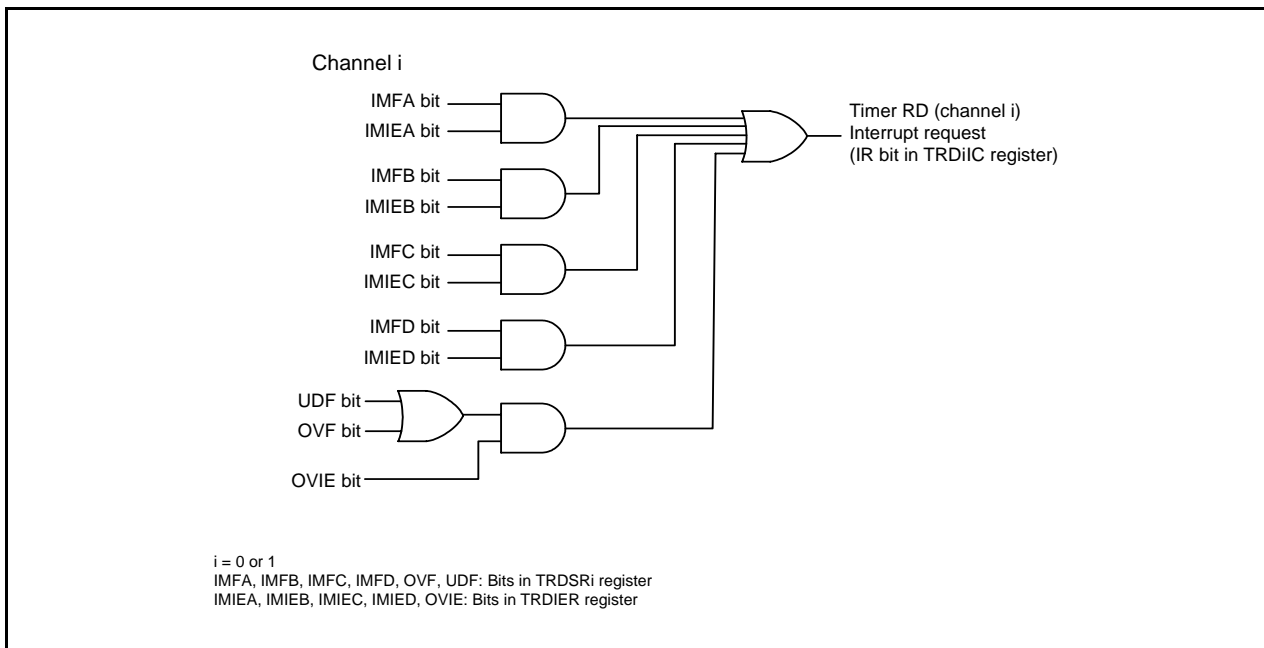
### 14.3.11 Timer RD Interrupt

Timer RD generates the Timer RD interrupt request based on 6 sources every channel. The Timer RD interrupt has 1 TRDiIC register (IR bit, ILVL0 to ILVL2 bits) every channel, and 1 vector.

Table 14.35 lists the Registers Associated with Timer RD Interrupt and Figure 14.109 shows the Block Diagram of Timer RD Interrupt.

**Table 14.35 Registers Associated with Timer RD Interrupt**

	Timer RD Status Register	Timer RD Interrupt Enable Register	Timer RD Interrupt Control Register
Channel 0	TRDSR0	TRDIER0	TRD0IC
Channel 1	TRDSR1	TRDIER1	TRD1IC



**Figure 14.109 Block Diagram of Timer RD Interrupt**

As with other maskable interrupts, the timer RD interrupt is controlled by the combination of the I flag, IR bit, bits ILVL0 to ILVL2, and IPL. However, since the interrupt source (timer RD interrupt) is generated by a combination of multiple interrupt request sources, the following differences from other maskable interrupts apply:

- When bits in the TRDSR $i$  register corresponding to bits set to 1 in the TRDIER $i$  register are set to 1 (enable interrupt), the IR bit in the TRDiIC register is set to 1 (interrupt requested).
- When either bits in the TRDSR $i$  register or bits in the TRDIER $i$  register corresponding to bits in the TRDSR $i$  register, or both of them, are set to 0, the IR bit is set to 0 (interrupt not requested). Therefore, even though the interrupt is not acknowledged after the IR bit is set to 1, the interrupt request will not be maintained.
- When the conditions of other request sources are met, the IR bit remains 1.
- When multiple bits in the TRDIER $i$  register are set to 1, which request source causes an interrupt is determined by the TRDSR $i$  register.
- Since each bit in the TRDSR $i$  register is not automatically set to 0 even if the interrupt is acknowledged, set each bit to 0 in the interrupt routine. For information on how to set these bits to 0, refer to the descriptions of the registers used in the different modes (**Figures 14.41, 14.56, 14.69, 14.81, 14.92 and 14.104**).

Refer to **TRDSR0 to TRDSR1 Registers in each mode (Figures 14.41, 14.56, 14.69, 14.81, 14.92 and 14.104)** for the TRDSRi register. Refer to **TRDIER0 to TRDIER1 Registers in each mode (Figures 14.42, 14.57, 14.70, 14.82, 14.93 and 14.105)** for the TRDIERi register.

Refer to **12.1.6 Interrupt Control** for the TRDiC register and **12.1.5.2 Relocatable Vector Tables** for the interrupt vector.

## 14.3.12 Notes on Timer RD

### 14.3.12.1 TRDSTR Register

- Set the TRDSTR register using the MOV instruction.
- When the CSELi (i = 0 or 1) is set to 0 (the count stops at compare match of registers TRDi and TRDGRAi), the count does not stop and the TSTARTi bit remains unchanged even if 0 (count stops) is written to the TSTARTi bit.  
Therefore, set the TSTARTi bit to 0 to change other bits without changing the TSTARTi bit when the CSELi bit is set to 0.  
To stop counting by a program, set the TSTARTi bit to 0 after setting the CSELi bit to 1. Although the CSELi bit is set to 1 and the TSTARTi bit is set to 0 at the same time (with 1 instruction), the count cannot be stopped.
- Table 14.36 lists the TRDIOj (j = A, B, C, or D) Pin Output Level when Count Stops to use the TRDIOj pin with the timer RD output.

**Table 14.36 TRDIOj (j = A, B, C, or D) Pin Output Level when Count Stops**

Count Stop	TRDIOj Pin Output when Count Stops
When the CSELi bit is set to 1, set the TSTARTi bit to 0 and the count stops.	Hold the output level immediately before the count stops.
When the CSELi bit is set to 0, the count stops at compare match of registers TRDi and TRDGRAi.	Hold the output level after output changes by compare match.

### 14.3.12.2 TRDi Register (i = 0 or 1)

- When writing the value to the TRDi register by a program while the TSTARTi bit in the TRDSTR register is set to 1 (count starts), avoid to overlap with the timing to set the TRDi register to 0000h, and then write. When the timing to set the TRDi register to 0000h overlaps with the timing to write the value to the TRDi register, the value is not written and the TRDi register is set to 0000h.  
These precautions are applicable when selecting the following by the CCLR2 to CCLR0 bits in the TRDCRi register.
  - 001b (clear by the TRDi register at the compare match with the TRDGRAi register)
  - 010b (clear by the TRDi register at the compare match with the TRDGRBi register.)
  - 011b (synchronous clear)
  - 101b (clear by the TRDi register at the compare match with the TRDGRCi register.)
  - 110b (clear by the TRDi register at the compare match with the TRDGRDi register.)
- When writing the value to the TRDi register and continuously reading the same register, the value before writing may be read. In this case, execute the JMP.B instruction between the writing and reading.
 

```

Program Example      MOV.W      #XXXXh, TRD0      ;Writing
                    JMP.B      L1              ;JMP.B
                    L1:      MOV.W      TRD0,DATA    ;Reading
      
```

### 14.3.12.3 TRDSRi Register (i = 0 or 1)

When writing the value to the TRDSRi register and continuously reading the same register, the value before writing may be read. In this case, execute the JMP.B instruction between the writing and reading.

```

Program Example      MOV.B      #XXh, TRDSR0      ;Writing
                    JMP.B      L1              ;JMP.B
                    L1:      MOV.B      TRDSR0,DATA    ;Reading
      
```

#### 14.3.12.4 Count Source Switch

- When switching the count source, switch it after the count stops.

Change procedure

- (1) Set the TSTART<sub>i</sub> (i = 0 or 1) bit in the TRDSTR register to 0 (count stops).
- (2) Change the TCK2 to TCK0 bits in the TRDCR<sub>i</sub> register.

- When changing the count source from fOCO40M to the other and stopping fOCO40M, wait 2 cycles or more of f1 after setting the clock switch, and then stop fOCO40M.

Change procedure

- (1) Set the TSTART<sub>i</sub> (i = 0 or 1) bit in the TRDSTR register to 0 (count stops).
- (2) Change the TCK2 to TCK0 bits in the TRDCR<sub>i</sub> register.
- (3) Wait 2 cycles or more of f1.
- (4) Set the FRA00 bit in the FRA0 register to 0 (high-speed on-chip oscillator stops).

#### 14.3.12.5 Input Capture Function

- Set the pulse width of input capture signal to 3 cycles or more of the Timer RD operation clock. (Refer to **Table 14.11 Timer RD Operation Clocks.**)
- The value in the TRD<sub>i</sub> register is transferred to the TRDGR<sub>ji</sub> register after 2 to 3 cycles of the Timer RD operation clock since the input capture signal is applied to the TRDIO<sub>ji</sub> pin (i = 0 or 1, j = either A, B, C or D) (no digital filter).

#### 14.3.12.6 Reset Synchronous PWM Mode

- When reset synchronous PWM mode is used for motor control, use it with OLS0 = OLS1.
- Set to reset synchronous PWM mode in the following procedure:

Change procedure

- (1) Set the TSTART0 bit in the TRDSTR register to 0 (count stops).
- (2) Set the CMD1 to CMD0 bits in the TRDFCR register to 00b (timer mode, PWM mode, and PWM3 mode).
- (3) Set the CMD1 to CMD0 bits to 01b (reset synchronous PWM mode).
- (4) Set the registers associated with other Timer RD again.

#### 14.3.12.7 Complementary PWM Mode

- When complementary PWM mode is used for motor control, use it with OLS0 = OLS1.
- Change the CMD1 to CMD0 bits in the TRDFCR register in the following procedure.

Change procedure: When setting to complementary PWM mode (including re-set), or changing the transfer timing from the buffer register to the general register in complementary PWM mode.

- (1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
- (2) Set the CMD1 to CMD0 bits in the TRDFCR register to 00b (timer mode, PWM mode, and PWM3 mode)
- (3) Set the DMD1 to CMD0 bits to 10b or 11b (complementary PWM mode).
- (4) Set the registers associated with other Timer RD again.

Change procedure: When stopping complementary PWM mode

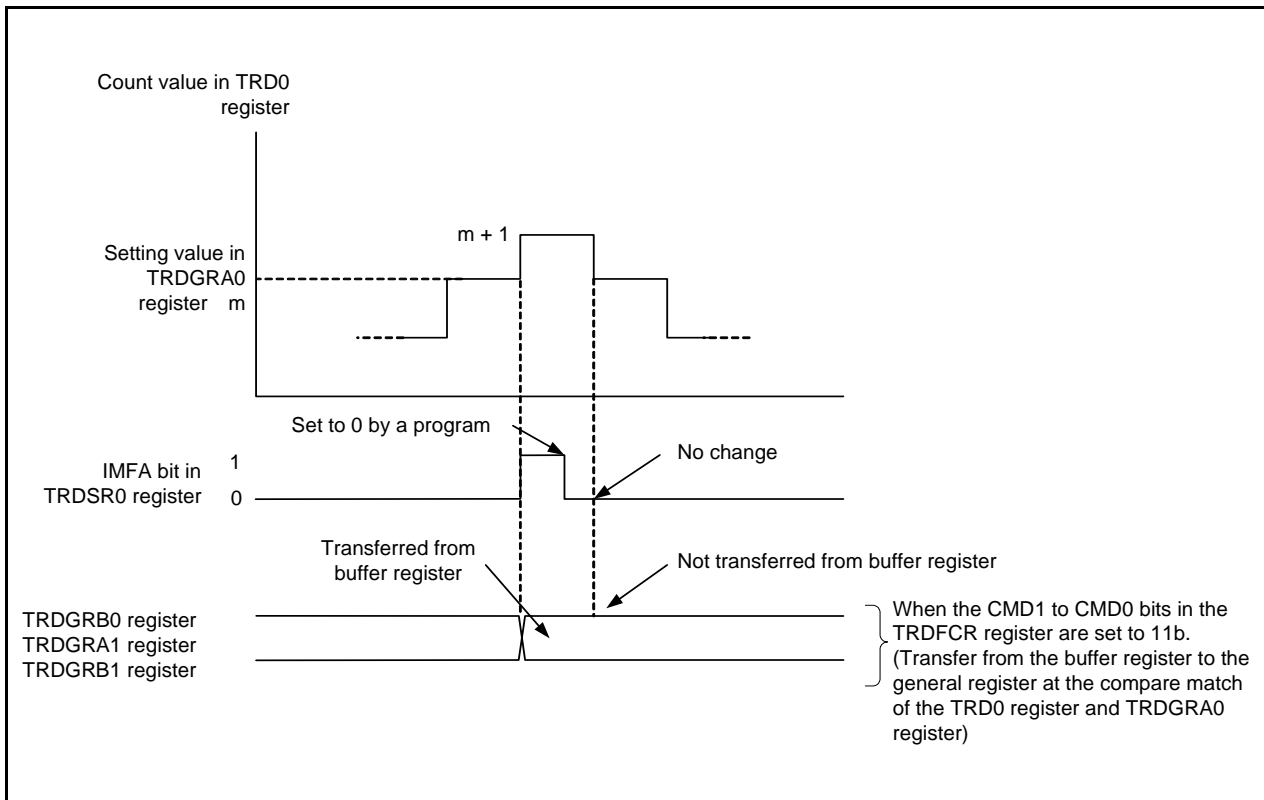
- (1) Set both the TSTART0 and CSEL1 bits in the TRDSTR register to 0 (count stops).
- (2) Set the CMD1 to CMD bits to 00b (other than reset synchronous PWM mode, complementary PWM mode)

- Do not write to the TRDGRA0, TRDGRB0, TRDGRA1 and TRDGRB1 registers during operation. When changing the PWM waveform, transfer the value written to the TRDGRD0, TRDGRC1 and TRDGRD1 registers to the TRDGRB0, TRDGRA1 and TRDGRB1 registers using the buffer operation. However, to write data to the TRDGRD0, TRDGRC1, or TRDGRD1 register, set bits BFD0, BFC1, and BFD1 to 0 (general register). After this, bits BFD0, BFC1, and BFD1 may be set to 1 (buffer register). The PWM period cannot be changed.

- When the value in the TRDGRA0 register is assumed as  $m$ , the TRD0 register counts order of  $m - 1$ ,  $m$ ,  $m + 1$ ,  $m$ ,  $m - 1$  when changing from increment to decrement.

When changing from  $m$  to  $m + 1$ , the IMFA bit is set to 1. Also, the CMD1 to CMD0 bits in the TRDFCR register are set to 11b (complementary PWM mode, buffer data transferred by the compare match in the TRD0 and TRDGRA0 registers), the content in the buffer register (TRDGRD0, TRDGRC1, TRDGRD1) is transferred to the general register (TRDGRB0, TRDGRA1, TRDGRB1).

For the order of  $m + 1$ ,  $m$ ,  $m - 1$  operation, the IMFA bit remains unchanged and data are not transferred to the register such as the TRDGRA0 register.



**Figure 14.110 Operation at Compare Match between Registers TRD0 and TRDGRA0 in Complementary PWM Mode**

- The TRD1 register counts the order of 1, 0, FFFFh, 0, 1 when changing from decrement to increment. The UDF bit is set to 1 by the order of 1, 0, FFFFh operation. Also, when the CMD1 to CMD0 bits in the TRDFCR register are set to 10b (complementary PWM mode, buffer data transferred by the underflow in the TRD1 register), the content in the buffer register (TRDGRD0, TRDGRC1, TRDGRD1) is transferred to the general register (TRDGRB0, TRDGRA1, TRDGRB1). For the order of FFFFh, 0, 1 operation, data are not transferred to the register such as the TRDGRB0 register. Also, at this time, the OVF bit remains unchanged.

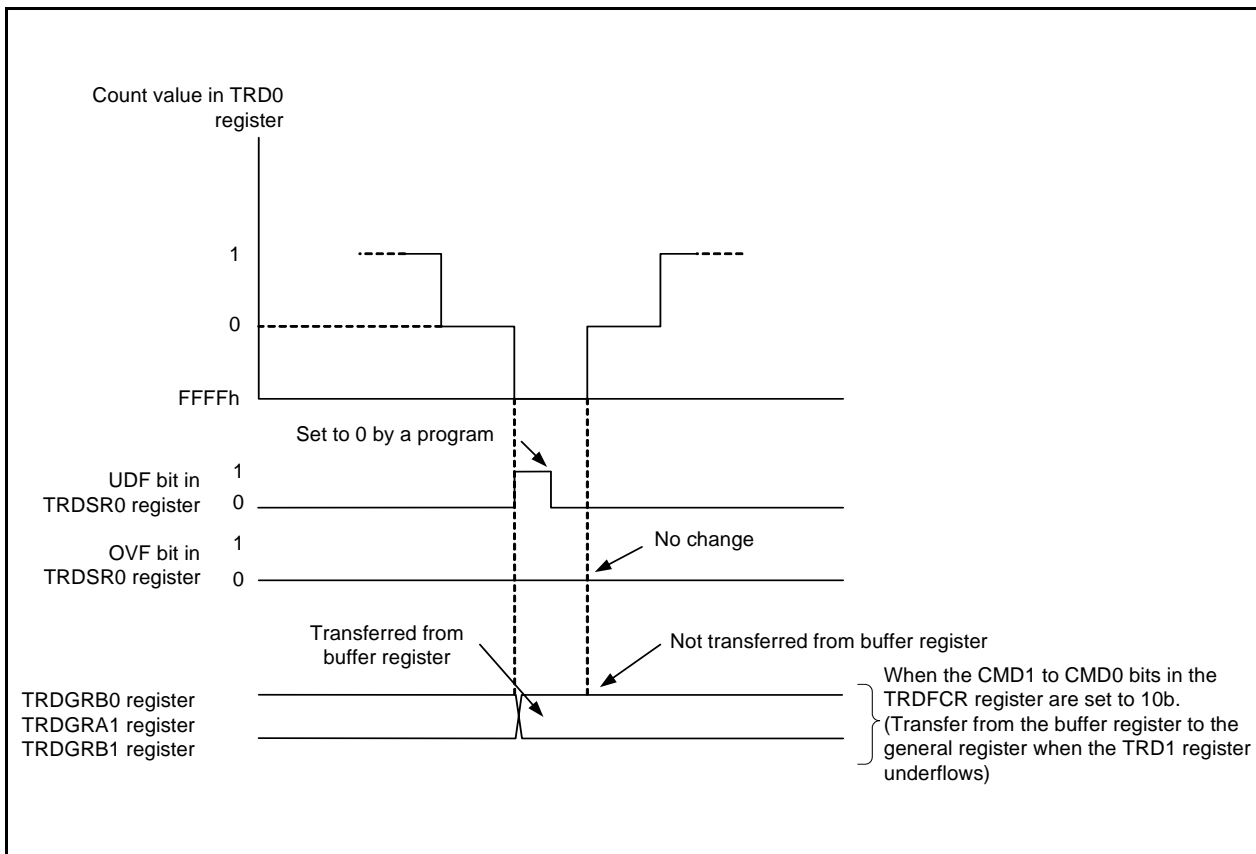


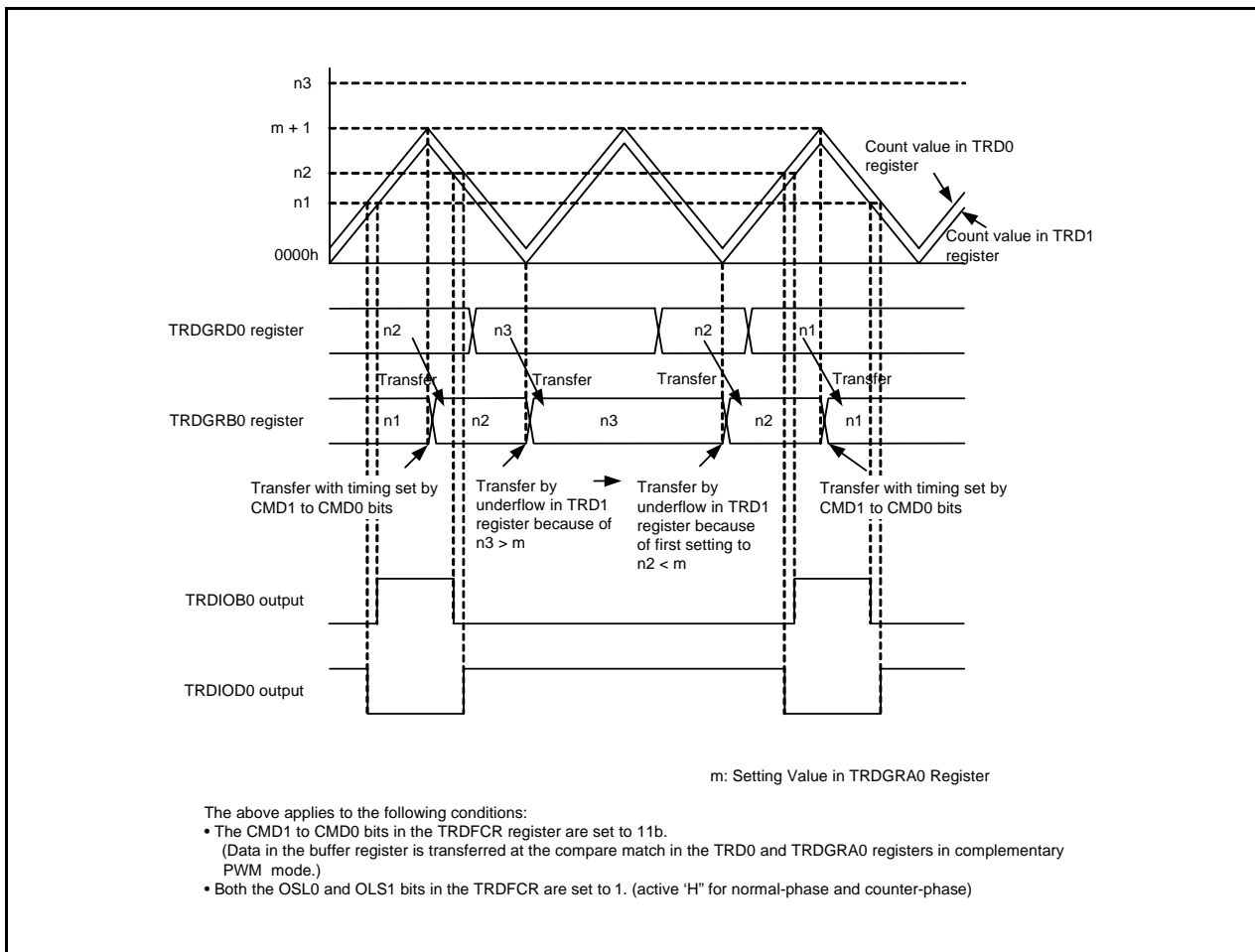
Figure 14.111 Operation When TRD1 Register Underflows in Complementary PWM Mode

- Select with the CMD1 to CMD0 bits for the data transfer timing from the buffer register to the general register. However, transfer with the following timing in spite of the value of the CMD1 to CMD0 bits for the following cases:

Value in buffer register  $\geq$  Value in TRDGRA0 register:

Transfer at the underflow in the TRD1 register.

And then, when setting the buffer register to 0001h or above and the smaller value than the one in the TRDGRA0 register, and the TRD1 register underflows in the first time after setting, the value is transferred to the general register. After that, transfer the value with the timing selected by the CMD1 to CMD0 bits.



**Figure 14.112 Operation When Value in Buffer Register  $\geq$  Value in TRDGRA0 Register in Complementary PWM Mode**

When the value in the buffer register is set to 0000h:

Transfer by the compare match in the TRD0 and TRDGRA0 registers.

And then, when setting the buffer register to 0001h or above and the smaller value than the one in the TRDGRA0 register, and the compare match in the TRD0 and TRDGRA0 registers in the first time after setting, the value is transferred to the general register. After that, transfer the value with the timing selected by the CMD1 to CMD0 bits.

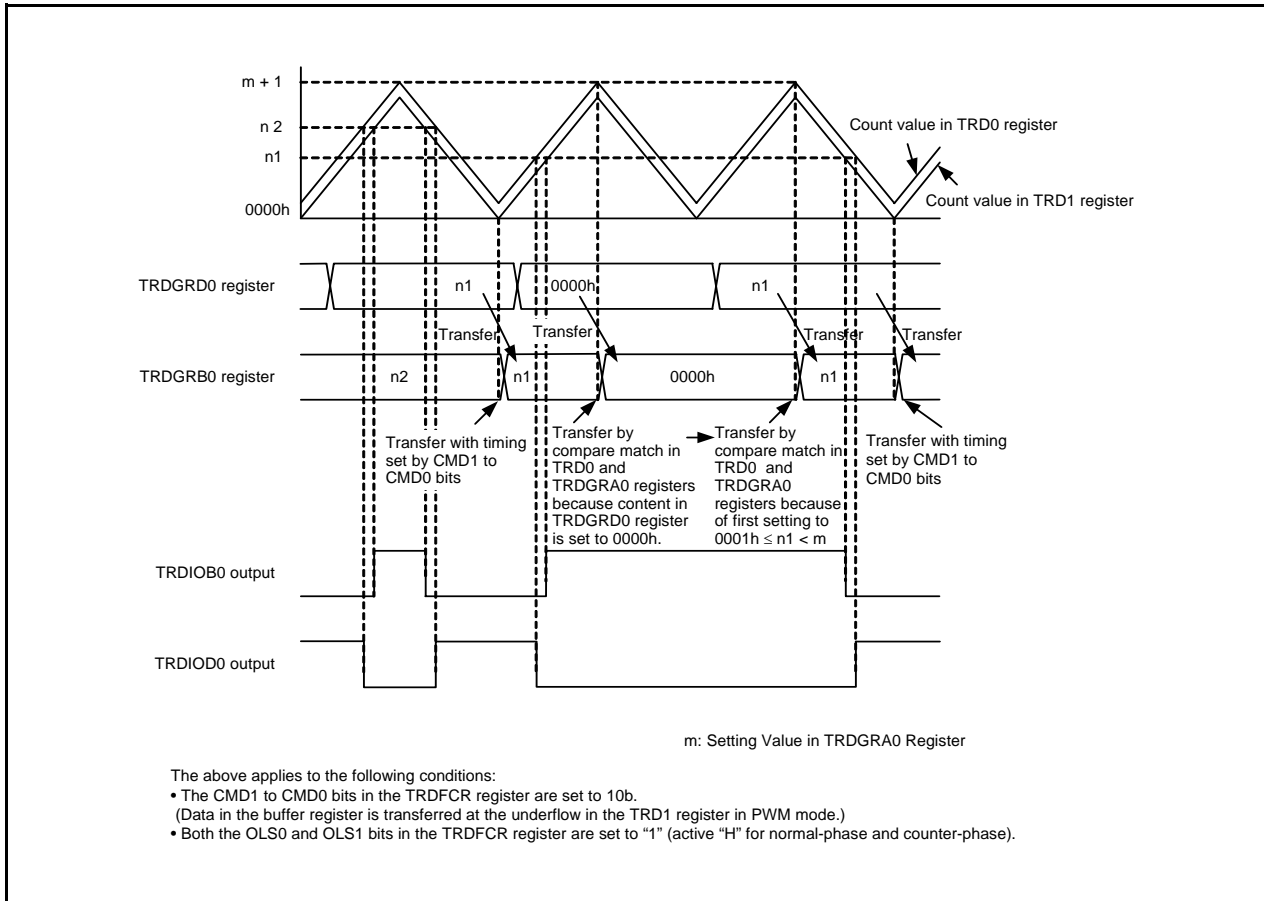


Figure 14.113 Operation When Value in Buffer Register Is Set to 0000h in Complementary PWM Mode

### 14.3.12.8 Count Source fOCO40M

The count source fOCO40M can be used with supply voltage  $VCC = 3.0$  to  $5.5$  V. For supply voltage other than that, do not set bits TCK2 to TCK0 in registers TRDCR0 and TRDCR to 110b (select fOCO40M as the count source).



## 14.4 Timer RE

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

- Output compare mode    Count a count source and detect the compare match

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

### 14.4.1 Output Compare Mode

The output compare mode is to count the internal count source divided-by-2 using the 4-bit or 8-bit counter and detect the compare value match with the 8-bit counter.

Figure 14.114 shows the Block Diagram of Output Compare Mode and Table 14.37 lists the Output Compare Mode Specifications. Figures 14.115 to 14.119 show the Registers Associated with Output Compare Mode and Figure 14.120 shows the Operation in Output Compare Mode.

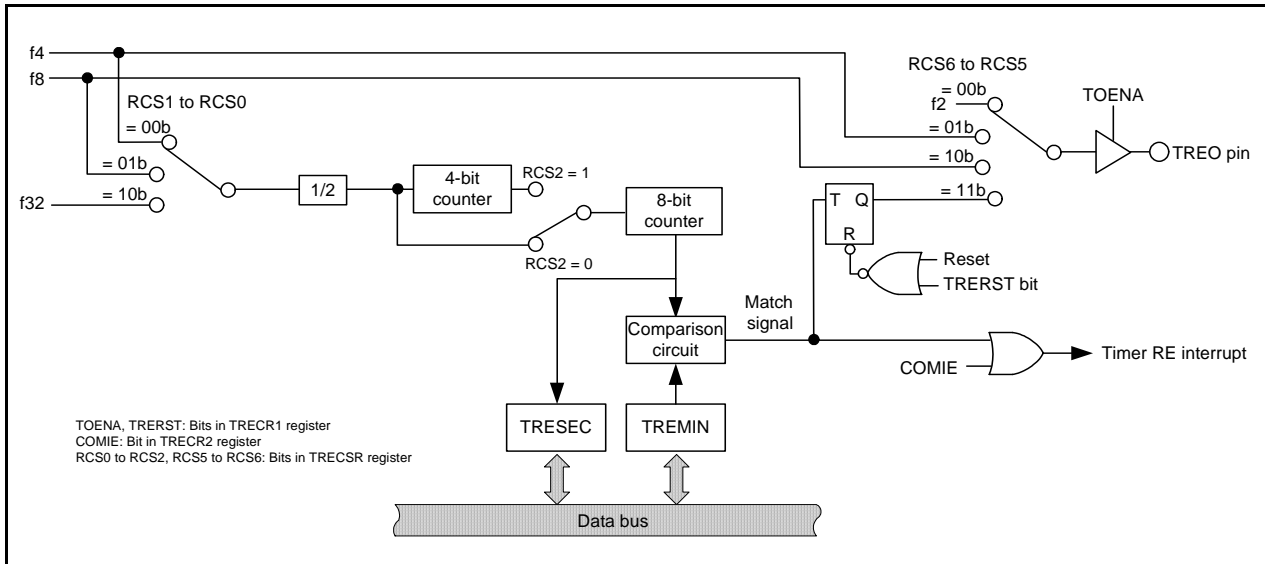


Figure 14.114 Block Diagram of Output Compare Mode

**Table 14.37 Output Compare Mode Specifications**

Item	Specification
Count Source	f4, f8, f32
Count Operation	<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	Write 1 (count starts) to the TSTART bit in the TRECR1 register
Count Stop Condition	Write 0 (count stops) 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 followings: <ul style="list-style-type: none"> <li>• Programmable I/O ports</li> <li>• Output any one of f2, f4 and 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 the TSTART and TCSTF bits in the TRECR1 register are set to 0 (timer stops), writing to the TREMIN register is enabled.
Select Functions	<ul style="list-style-type: none"> <li>• Select use of 4-bit counter</li> <li>• Compare output function</li> </ul> Every time the 8-bit counter value matches with the TREMIN register value, TREO output polarity is reversed. The TREO pin outputs "L" after reset is deasserted and the Timer RE reset by the TRERST bit in the TRECR1 register. Output level is held by setting the TSTART bit to 0 (count stops).

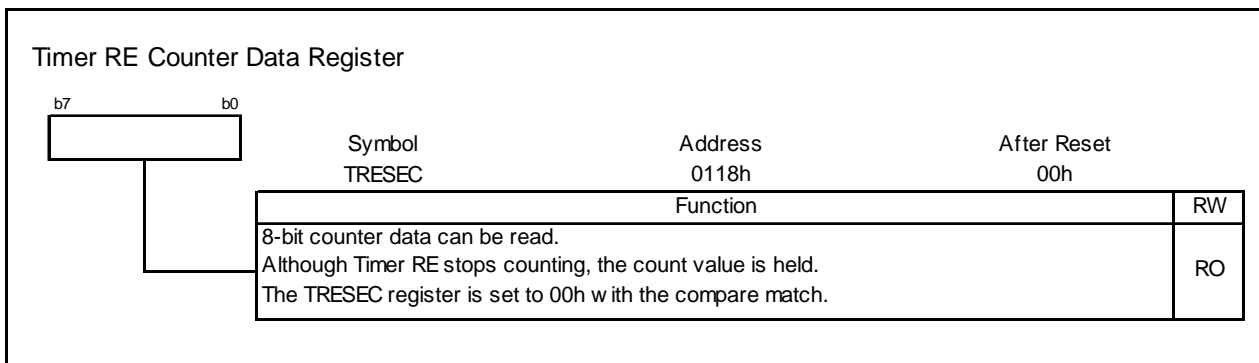


Figure 14.115 TRESEC Register in Output Compare Mode

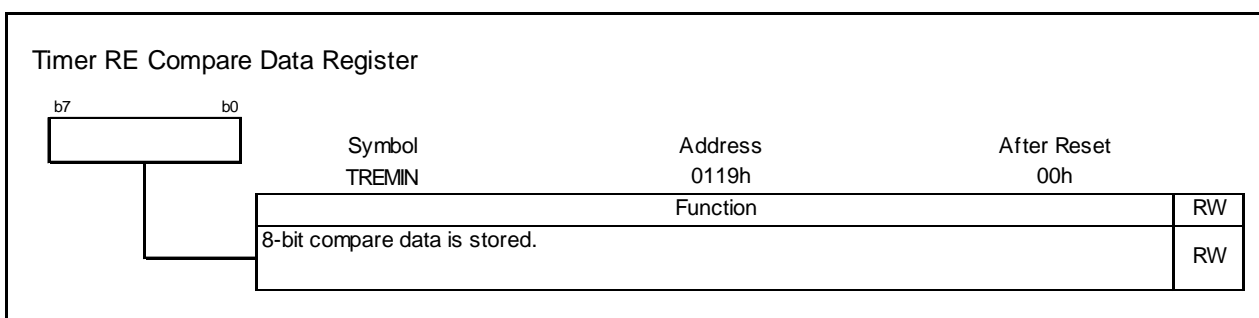


Figure 14.116 TREMIN Register in Output Compare Mode

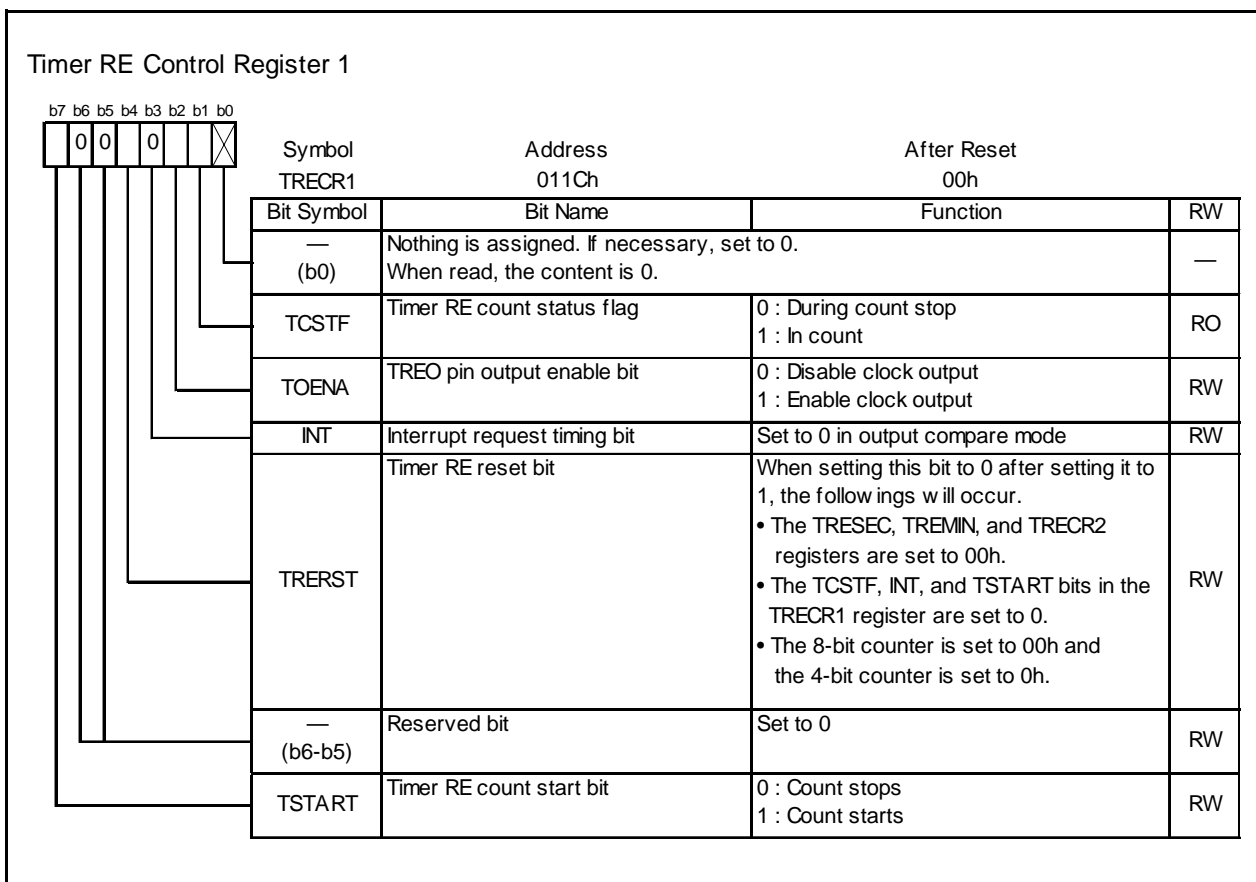


Figure 14.117 TREC1 Register in Output Compare Mode

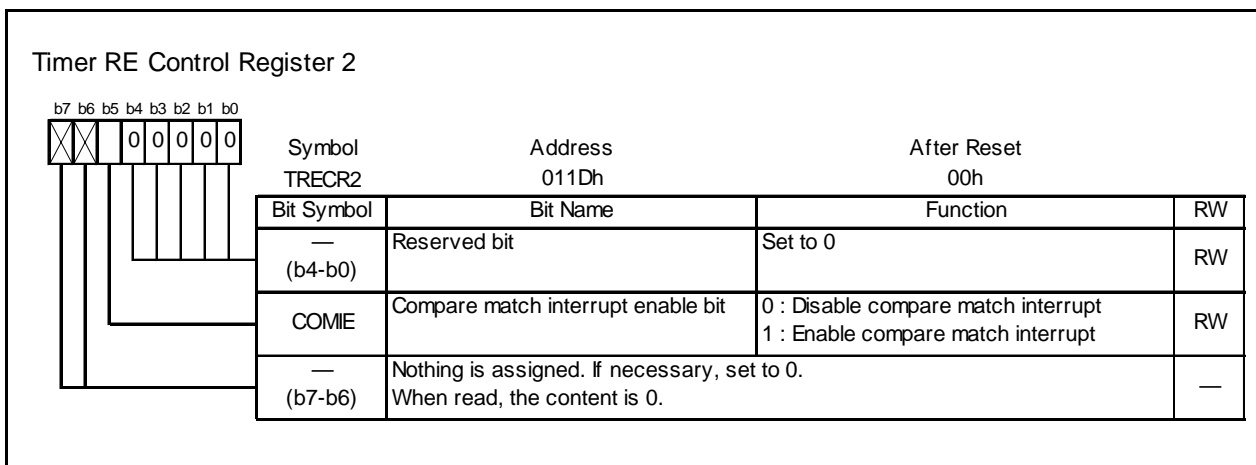


Figure 14.118 TREC2 Register in Output Compare Mode

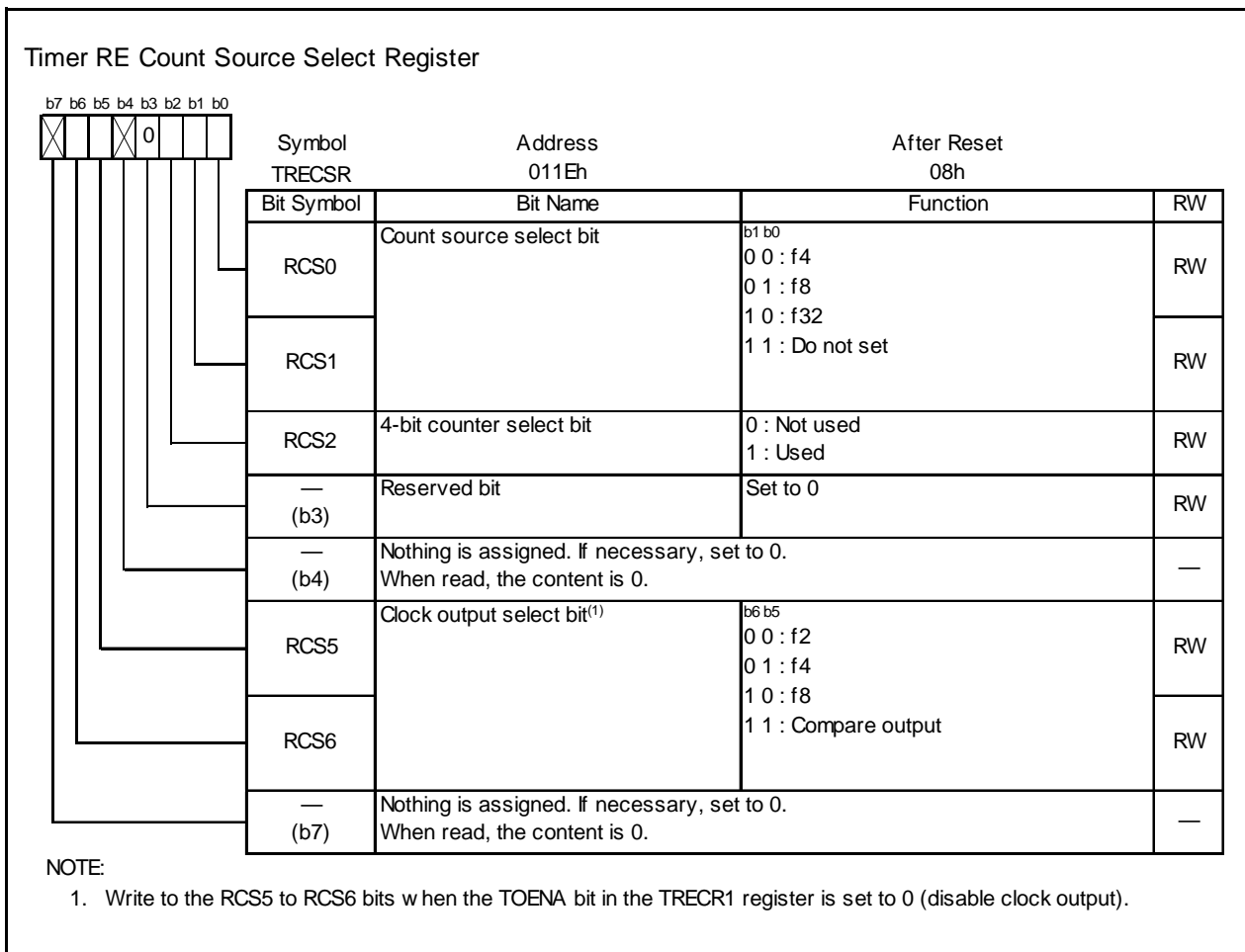


Figure 14.119 TRECSR Register in Output Compare Mode

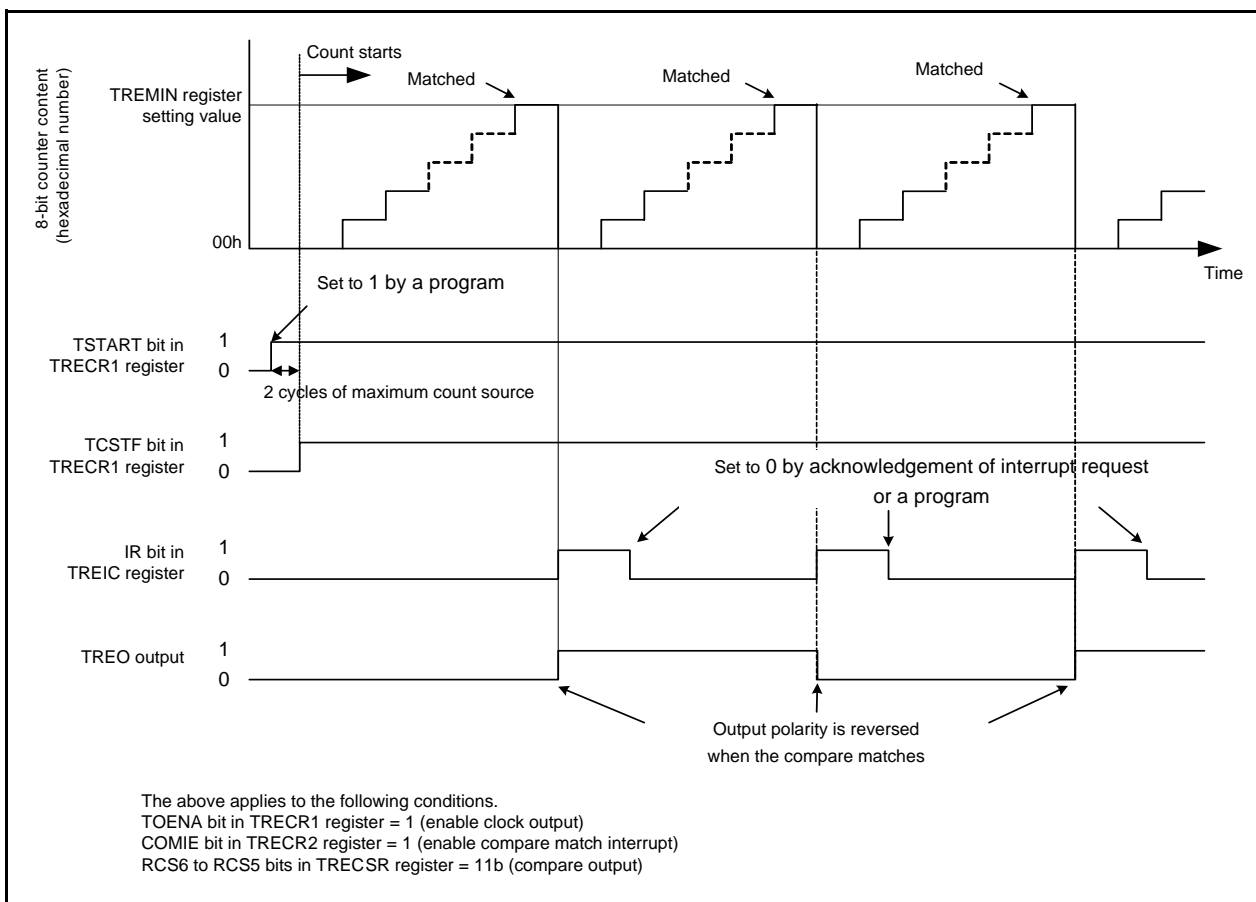


Figure 14.120 Operation in Output Compare Mode

## 14.4.2 Notes on Timer RE

### 14.4.2.1 Starting and Stopping Count

Timer RE has the TSTART bit for instructing count start or stop, and the TCSTF bit which indicates count start or stop. The TSTART and TCSTF bits are in the TREC1 register.

Timer RE starts counting when setting the TSTART bit to 1 (count starts) and the TCSTF bit is set to 1 (count starts). It takes the time for 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, TREC1, TREC2, TRECSR

### 14.4.2.2 Register Setting

Write to the following registers or bits while timer RE stops.

- TRESEC and TREC2 registers
- The INT bit in TREC1 register
- RCS0 to RCS2 bits in TRECSR register

The state while Timer RE stops is indicated as the state where the TSTART and TCSTF bits in the TREC1 register are set to 0 (timer RE stops).

Also, set all above-mentioned registers and bits (immediately before timer RE count starts) before setting the TREC2 register.



## 15. Serial Interface

Serial Interface is configured with two channels: UART0 and UART1. Each UART0 and Uart1 has an exclusive timer to generate a transfer clock and they operate independently.

Figure 15.1 shows the UARTi (i = 0 or 1) Block Diagram. Figure 15.2 shows the UARTi (i = 0 or 1) Transmit/Receive Unit.

UART0 has two modes: clock synchronous serial I/O mode, and clock asynchronous serial I/O mode (UART mode). UART1 has only one mode: clock asynchronous serial interface mode (UART mode).

Figures 15.3 to 15.6 show the Registers Associated with UARTi.

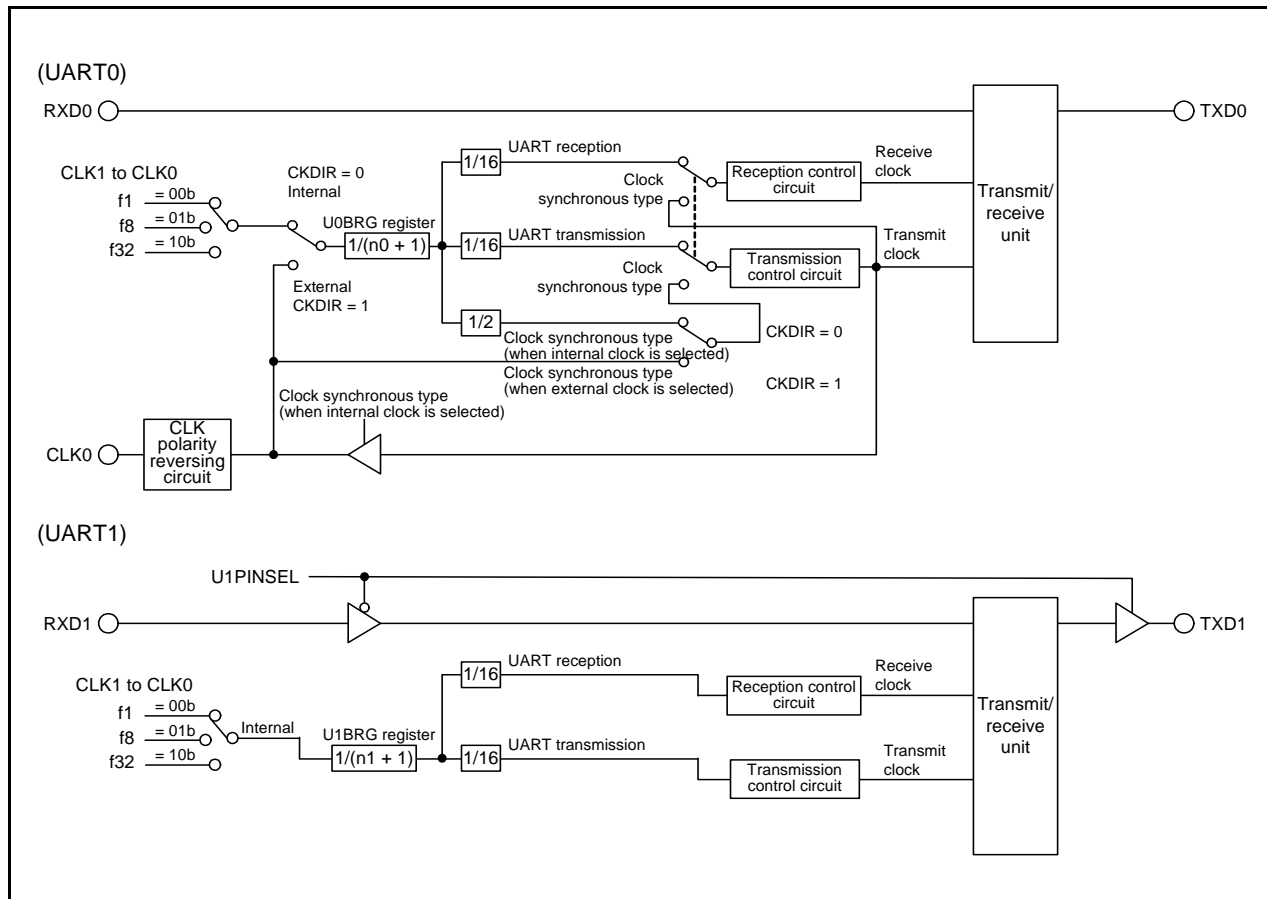


Figure 15.1 UARTi (i = 0 or 1) Block Diagram

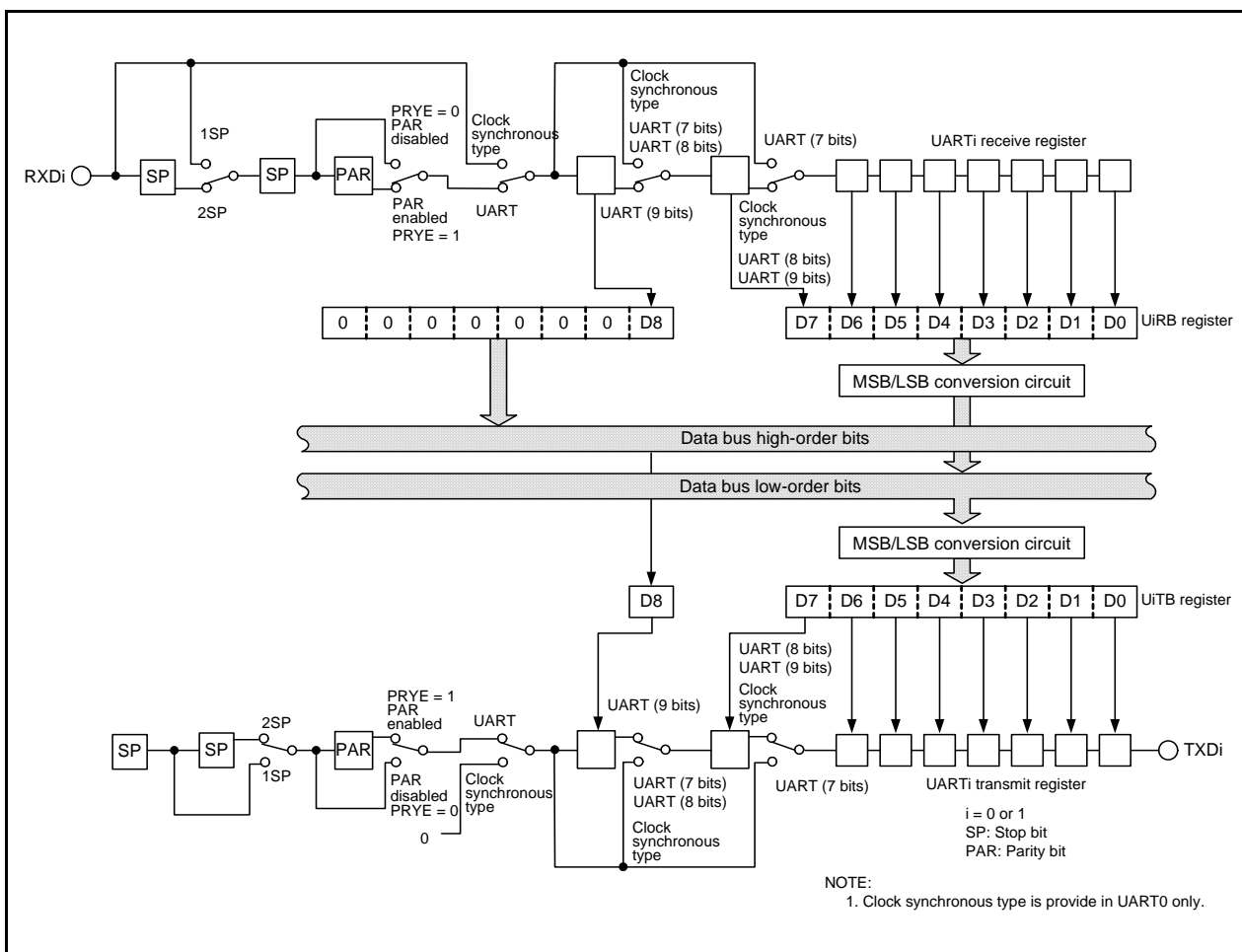


Figure 15.2 UARTi (i = 0 or 1) Transmit/Receive Unit

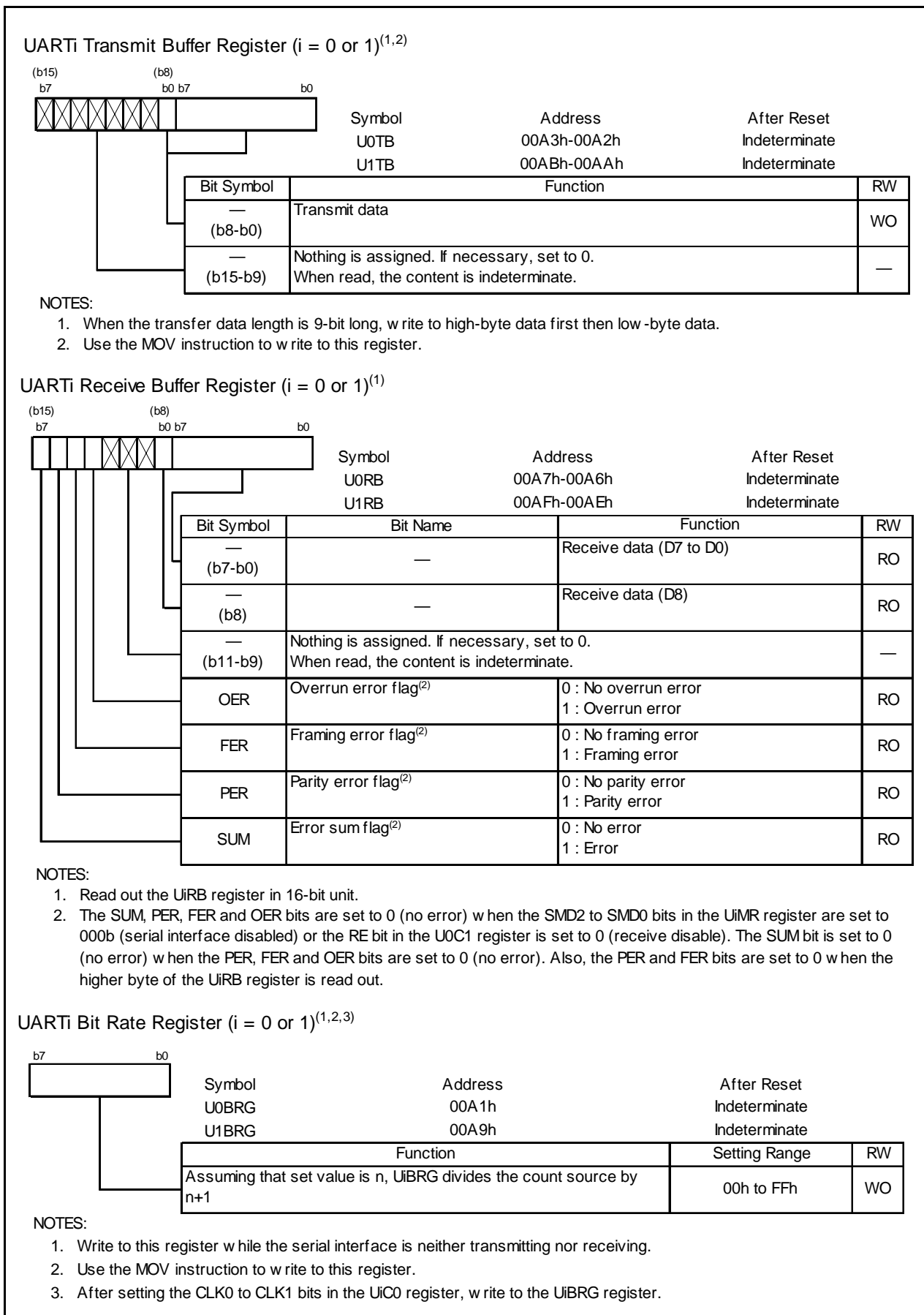


Figure 15.3 Registers UiTB, U1RB, and UiBRG (i = 0 or 1)

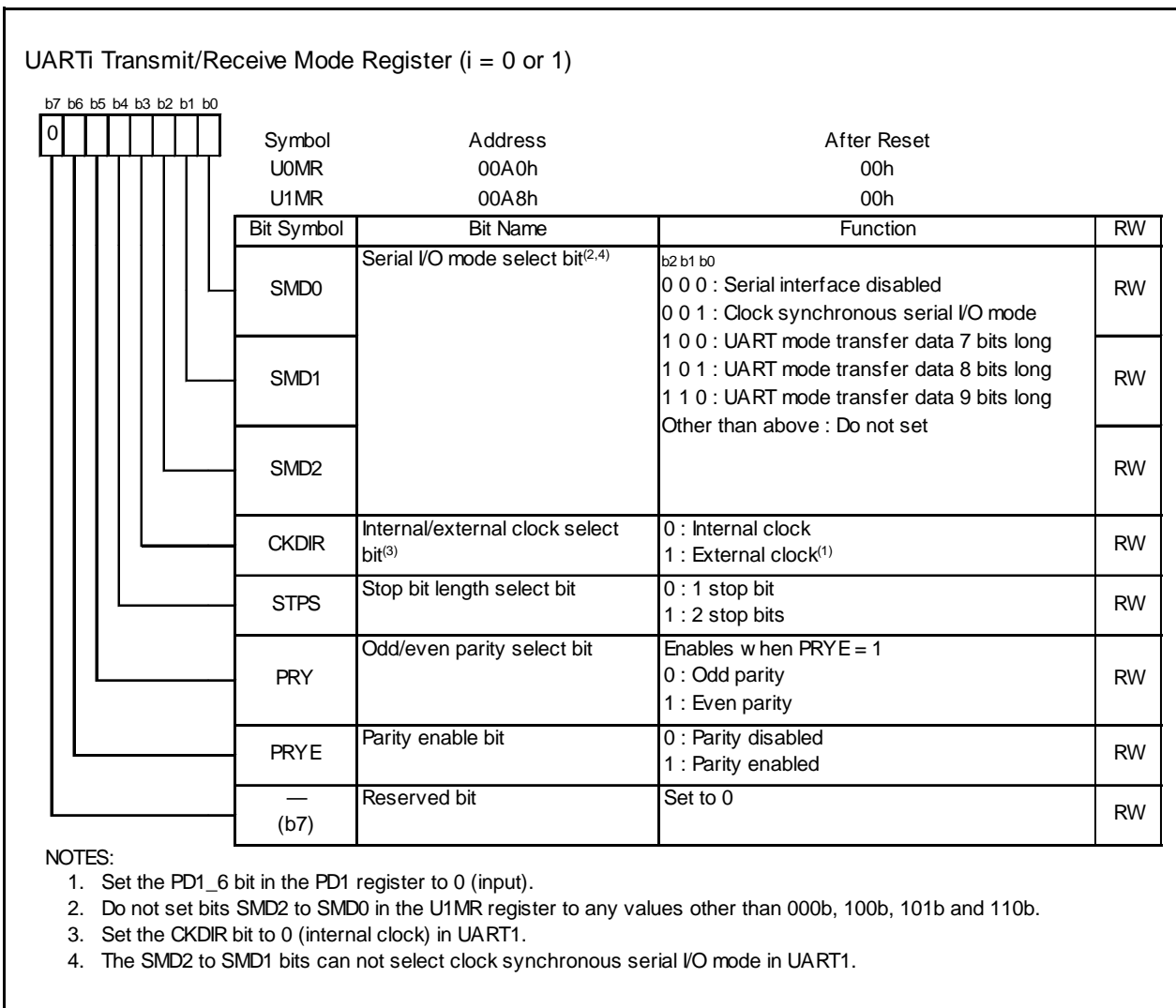


Figure 15.4 UiMR Register (i = 0 or 1)

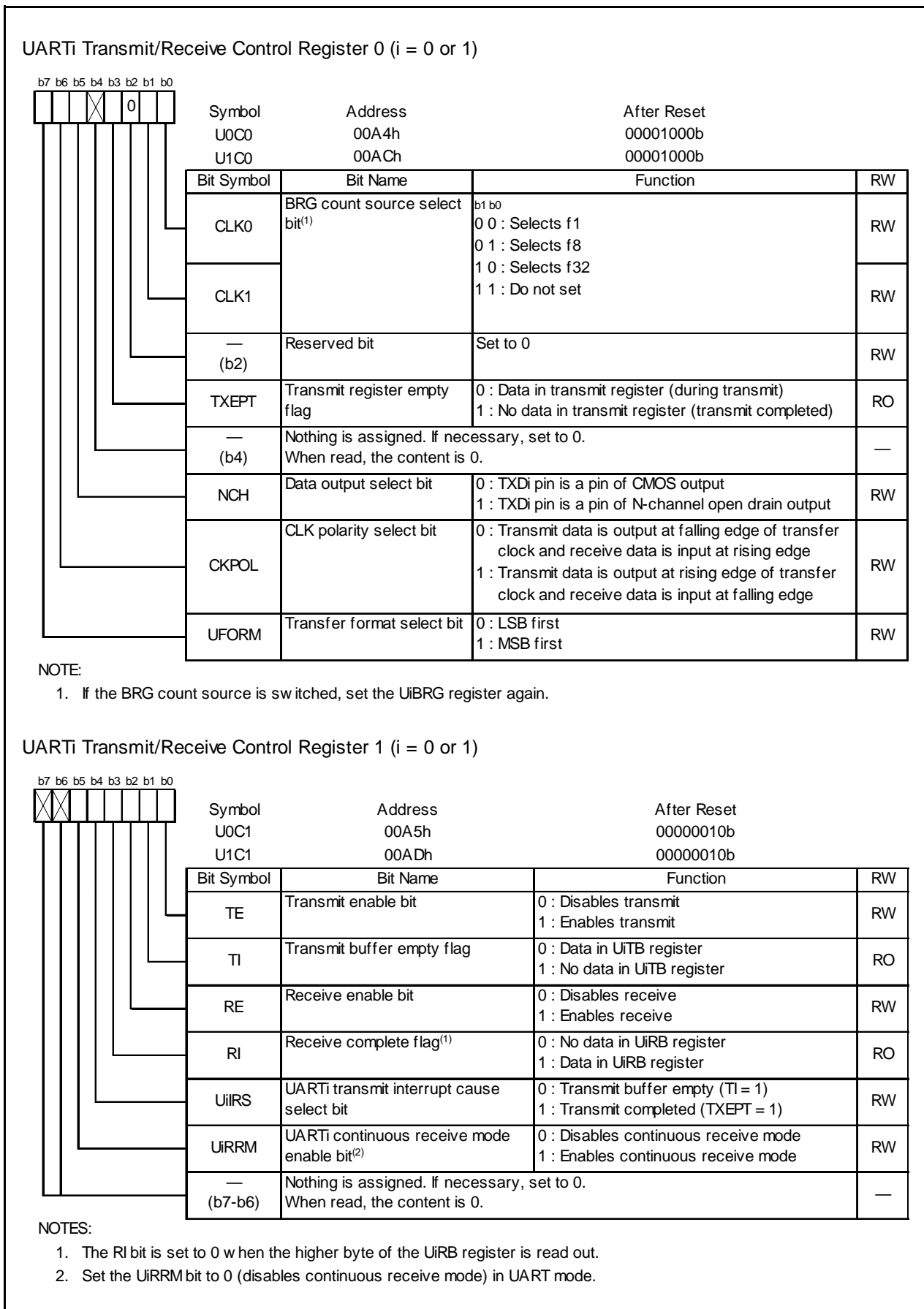


Figure 15.5 Registers UiC0 and UiC1 (i = 0 or 1)

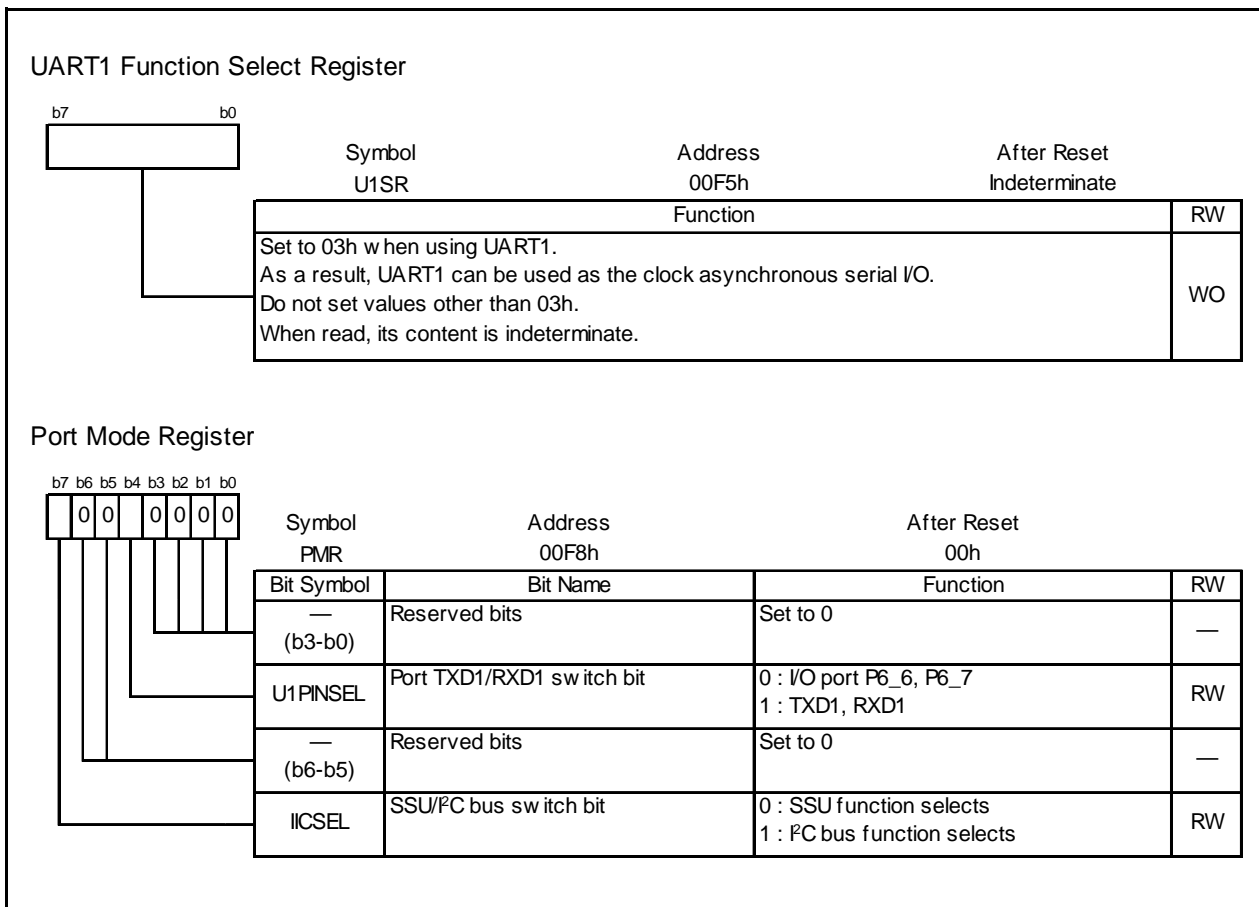


Figure 15.6 Registers U1SR and PMR

## 15.1 Clock Synchronous Serial I/O Mode

The clock synchronous serial I/O mode is mode to transmit and receive data using a transfer clock. This mode is selected in UART0 only.

Table 15.1 lists the Clock Synchronous Serial I/O Mode Specifications. Table 15.2 lists the Registers Used and Settings in Clock Synchronous Serial I/O Mode<sup>(1)</sup>.

**Table 15.1 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 U0MR 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> setting value in U0BRG register: 00h to FFh</li> <li>The CKDIR bit is set to 1 (external clock): input from CLK0 pin</li> </ul>
Transmit Start Conditions	<ul style="list-style-type: none"> <li>Before transmit starts, the following requirements are required<sup>(1)</sup> <ul style="list-style-type: none"> <li>The TE bit in the U0C1 register is set to 1 (transmit enabled)</li> <li>The TI bit in the U0C1 register is set to 0 (data in the U0TB register)</li> </ul> </li> </ul>
Receive Start Conditions	<ul style="list-style-type: none"> <li>Before receive starts, the following requirements are required<sup>(1)</sup> <ul style="list-style-type: none"> <li>The RE bit in the U0C1 register is set to 1 (receive enabled)</li> <li>The TE bit in the U0C1 register is set to 1 (transmit enabled)</li> <li>The TI bit in the U0C1 register is set to 0 (data in the U0TB register)</li> </ul> </li> </ul>
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>When transmit, one of the following conditions can be selected <ul style="list-style-type: none"> <li>The U0IRS bit is set to 0 (transmit buffer empty): when transferring data from the U0TB register to UART0 transmit register (when transmit starts)</li> <li>The U0IRS bit is set to 1 (transmit completes): when completing transmit data from U0Ti transmit register</li> </ul> </li> <li>When receive When transferring data from the UART0 receive register to the U0RB register (when receive completes)</li> </ul>
Error Detection	<ul style="list-style-type: none"> <li>Overrun error<sup>(2)</sup> This error occurs if serial interface starts receiving the following data before reading the U0RB register and receives the 7th bit of the following 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 beginning with the bit 0 or beginning with the bit 7 can be selected</li> <li>Continuous receive mode selection Receive is enabled immediately by reading the U0RB register</li> </ul>

### NOTES:

- When an external clock is selected, meet the conditions while the CKPOL bit in the U0C0 register is set to 0 (transmit data output at the falling edge and the receive data input at the rising edge of the transfer clock), the external clock is held "H"; if the CKPOL bit in the U0C0 register is set to 1 (transmit data output at the rising edge and the receive data input at the falling edge of the transfer clock), the external clock is held "L".
- If an overrun error occurs, the receive data (b0 to b8) of the U0RB register will be undefined. The IR bit in the S0RIC register remains unchanged.

**Table 15.2 Registers Used and Settings in Clock Synchronous Serial I/O Mode(1)**

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

**NOTE:**

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

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

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

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



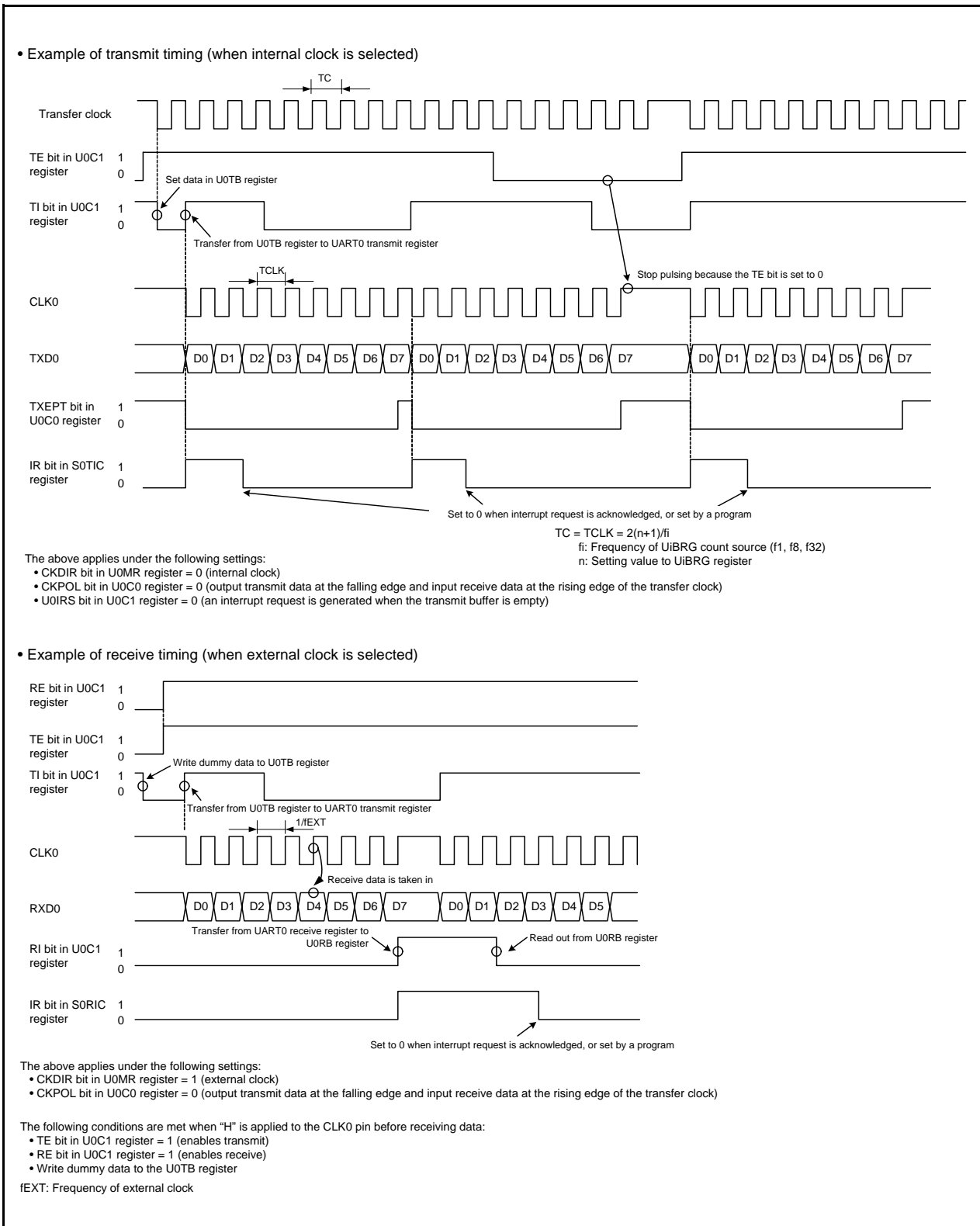


Figure 15.7 Transmit and Receive Timing Example in Clock Synchronous Serial I/O Mode

### 15.1.1 Polarity Select Function

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

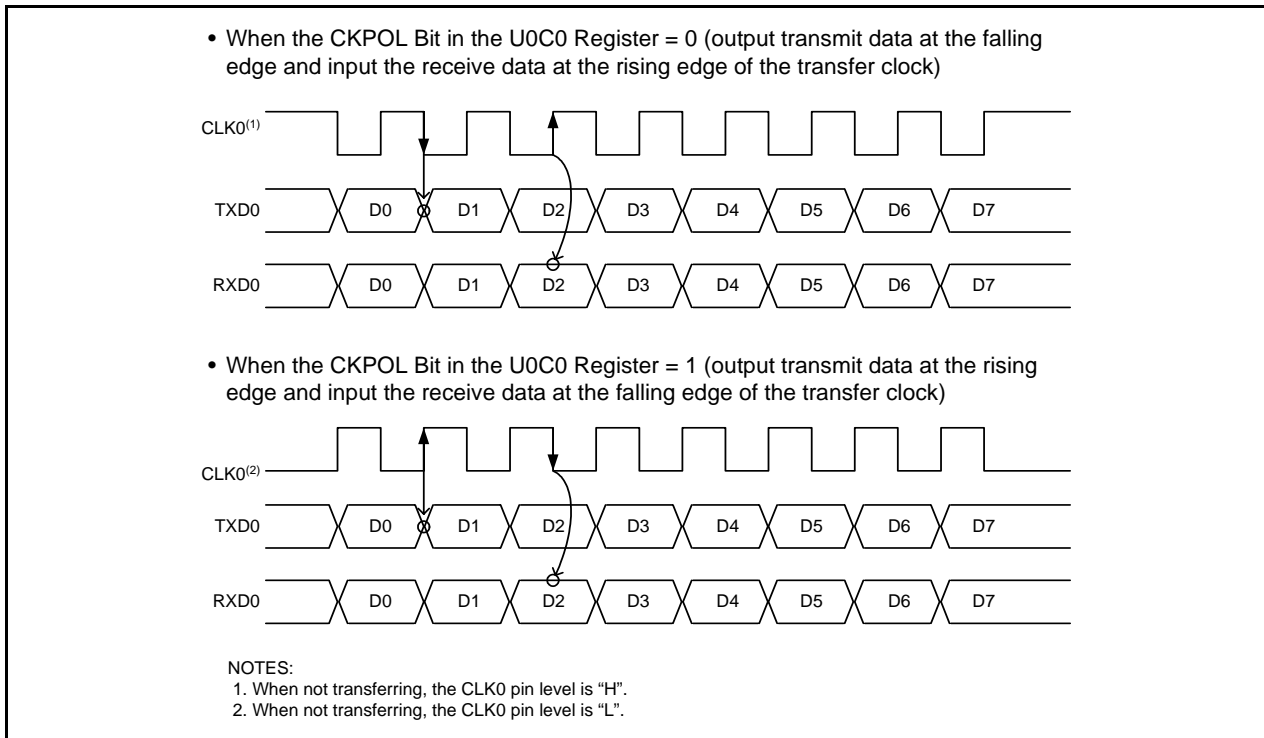


Figure 15.8 Transfer Clock Polarity

### 15.1.2 LSB First/MSB First Select Function

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

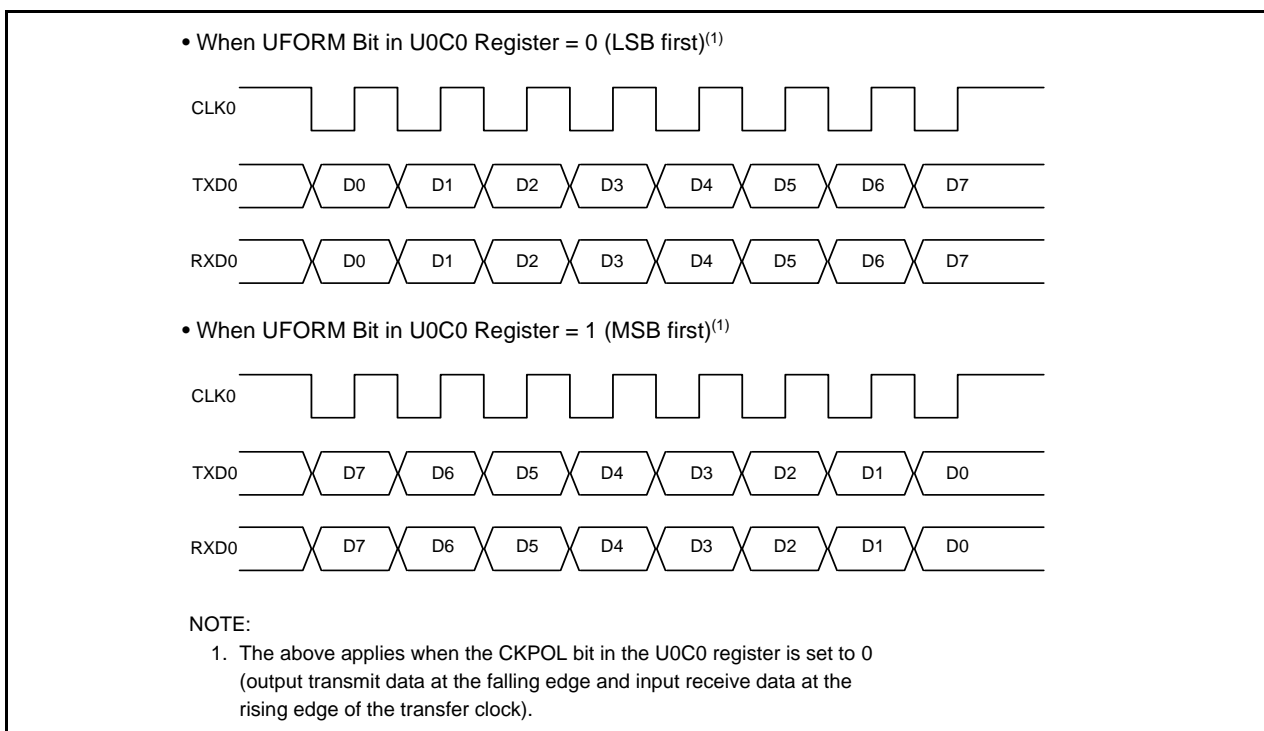


Figure 15.9 Transfer Format

### 15.1.3 Continuous Receive Mode

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

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

The UART mode allows transmit and receive data after setting the desired bit rate and transfer data format.

Table 15.4 lists the UART Mode Specifications. Table 15.5 lists the Registers Used and Settings for UART Mode.

**Table 15.4 UART Mode Specifications**

Item	Specification
Transfer Data Formats	<ul style="list-style-type: none"> <li>• Character bit (transfer data): selectable from 7, 8 or 9 bits</li> <li>• Start bit: 1 bit</li> <li>• Parity bit: selectable from odd, even, or none</li> <li>• Stop bit: selectable from 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> setting value in U0BRG 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 CLK0 pin <math>n =</math> setting value in UiBRG register: 00h to FFh</li> </ul>
Transmit Start Conditions	<ul style="list-style-type: none"> <li>• Before transmit starts, the following are required <ul style="list-style-type: none"> <li>- TE bit in UiC1 register is set to 1 (transmit 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 receive starts, the following are required <ul style="list-style-type: none"> <li>- RE bit in UiC1 register is set to 1 (receive enabled)</li> <li>- Detects start bit</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 transmit 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 receive ends)</li> </ul>
Error Detection	<ul style="list-style-type: none"> <li>• Overrun error<sup>(1)</sup> This error occurs if serial interface starts receiving the following data before reading the UirB register and receiving the bit one before the last stop bit of the following data</li> <li>• Framing error This error occurs when the number of stop bits set are not detected</li> <li>• Parity error This error occurs when parity is enabled, 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 any of the overrun, framing, and parity errors is generated</li> </ul>

$i = 0$  or  $1$

NOTE:

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

**Table 15.5 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-bit long Set to 101b when transfer data is 8-bit long Set to 110b when transfer data is 9-bit long
	CKDIR	Select the internal clock or external clock <sup>(3)</sup>
	STPS	Select the stop bit
	PRY, PRYE	Select whether parity is included and 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-bit long. Set to 0 when transfer data is 7- or 9-bit 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 UARTi transmit interrupt source
	UiRRM	Set to 0

i = 0 or 1

**NOTES:**

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

Table 15.6 lists the I/O Pin Functions in UART Mode. After the UARTi (i = 0 or 1) operating mode is selected, the TXDi pin outputs "H" level (if the NCH bit is set to 1 (N-channel open-drain outputs), this pin is in a high-impedance state) until transfer starts.

**Table 15.6 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 receive only)
RXD0(P1_5)	Input serial data	The PD1_5 bit in the PD1 register = 0 (P1_5 can be used as an input port when performing transmit only)
CLK0(P1_6)	Programmable I/O port	The CKDIR bit in the U0MR register = 0
	Input transfer clock	The CKDIR bit in the U0MR register = 1 The PD1_6 bit in the PD1 register = 0
TXD1(P6_6)	Output serial data	U1PINSEL bit in PMR register = 1 (Cannot be used as a port when performing receive only)
RXD1(P6_7)	Input serial data	U1PINSEL bit in PMR register = 1 The PD6_7bit in the PD6 register = 0 (P6_7 can be used as an input port when performing transmit only)

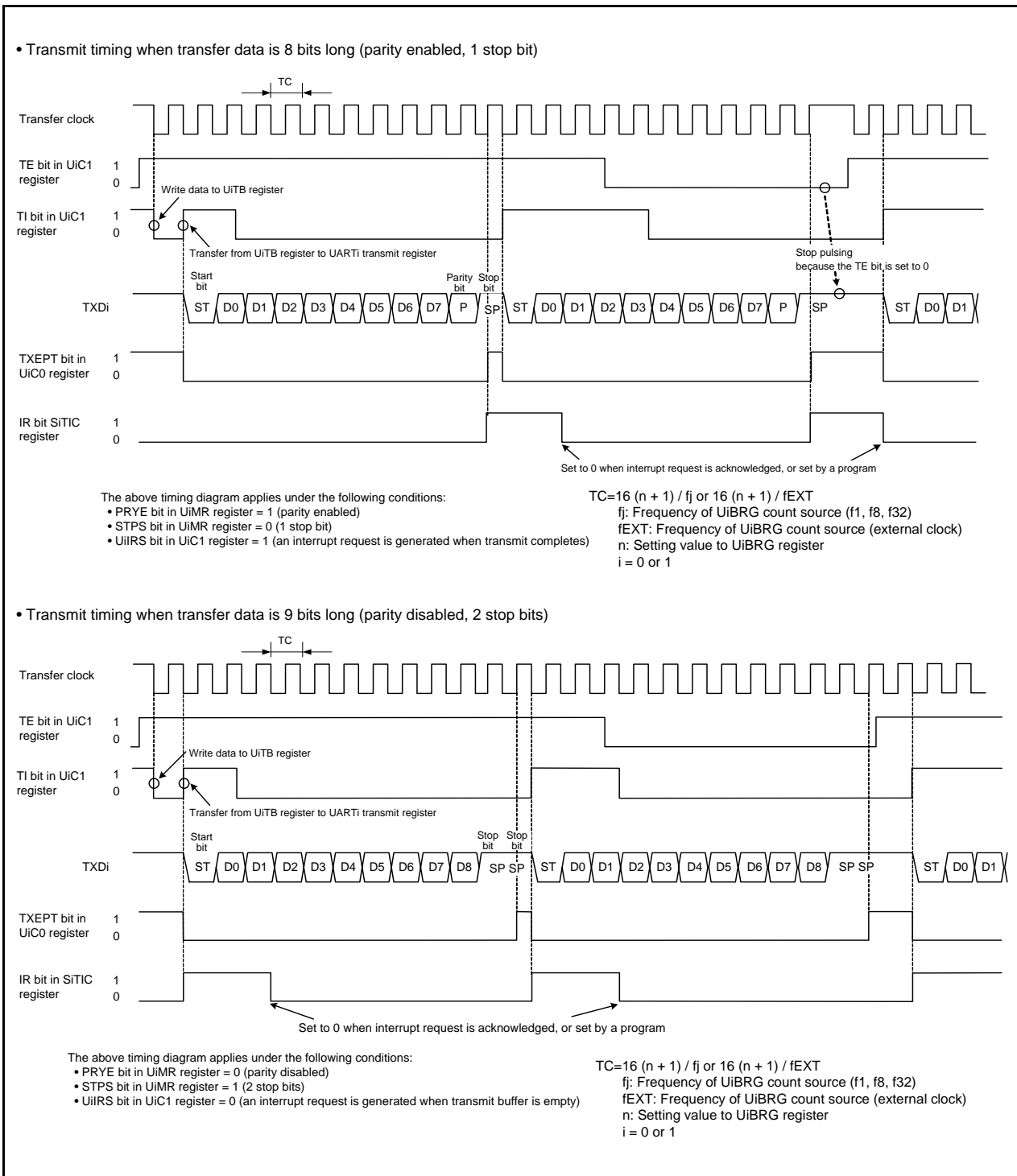


Figure 15.10 Transmit Timing in UART Mode

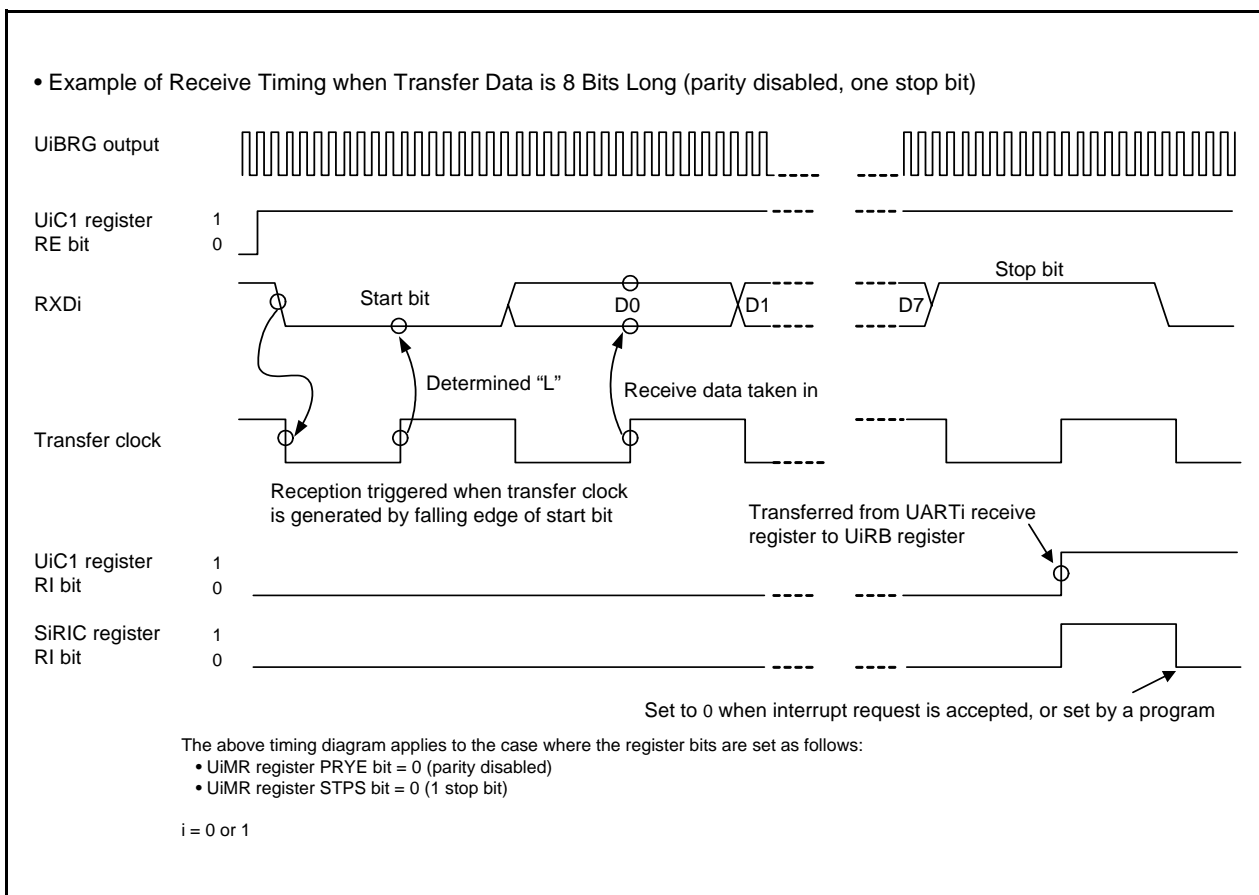


Figure 15.11 Receive Timing Example in UART Mode

### 15.2.1 Bit Rate

Divided-by-16 of frequency by the UiBRG (i = 0 or 1) register in UART mode is a bit rate.

<UART Mode>	
• When selecting internal clock	Setting value to the UiBRG register = $\frac{f_j}{\text{Bit Rate} \times 16} - 1$
	f <sub>j</sub> : Count source frequency of the UiBRG register (f1, f8 and f32)
• When selecting external clock	Setting value to the UiBRG register = $\frac{f_{EXT}}{\text{Bit Rate} \times 16} - 1$
	f <sub>EXT</sub> : Count source frequency of the UiBRG register (external clock) i = 0 or 1

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

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

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

i = 0 or 1



### 15.3 Notes on Serial Interface

- When reading data from the UiRB (i = 0 or 1) register even in the clock asynchronous serial I/O mode or in the clock synchronous serial I/O mode. Ensure to read data in 16-bit unit. When the high-order byte of the UiRB register is read, the PER and FER bits 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 high-order byte first, then low-order byte in 8-bit units.

Example (when reading transmit buffer register):

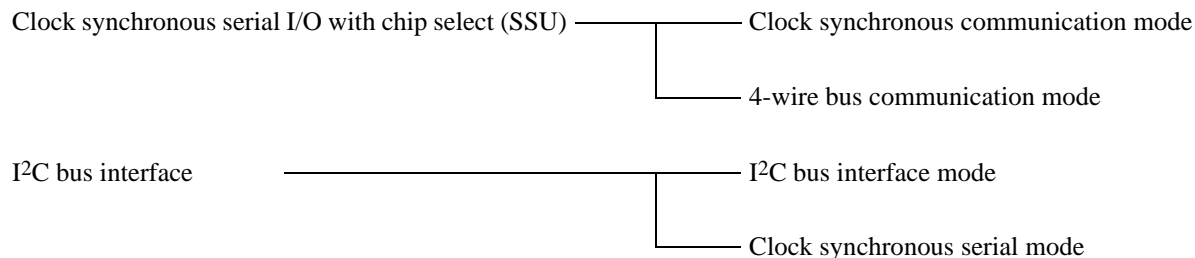
```
MOV.B #XXH,00A3H ; Write the high-order byte of UOTB register
```

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

## 16. Clock Synchronous Serial Interface

The clock synchronous serial interface is configured as follows.

### Clock Synchronous Serial Interface



The clock synchronous serial interface uses the registers of addresses 00B8h to 00BFh. Registers, bits, symbols and functions vary even in the same addresses depending on the modes. Refer to registers of each function for details. Also, the differences between clock synchronous communication mode and clock synchronous serial mode are the options of the transfer clock, clock output format and data output format.

### 16.1 Mode Selection

The clock synchronous serial interface contains 4 modes.

Table 16.1 lists the Mode Selections. Refer to **16.2 Clock Synchronous Serial I/O with Chip Select (SSU)** or after for details of each mode.

**Table 16.1 Mode Selections**

IICSEL Bit in PMR Register	Bit 7 in 00B8h (ICE Bit in ICCR1 Register)	Bit 0 in 00BDh (SSUMS Bit in SSMR2 Register, FS Bit in SAR Register)	Function	Mode
0	0	0	Clock synchronous serial I/O with chip select	Clock synchronous communication mode
0	0	1		4-wire bus communication mode
1	1	0	I <sup>2</sup> C bus interface	I <sup>2</sup> C bus interface mode
1	1	1		Clock synchronous serial mode

## 16.2 Clock Synchronous Serial I/O with Chip Select (SSU)

The serial data of the clock synchronous can communicate for the clock synchronous serial I/O with chip select. Table 16.2 lists the Clock Synchronous Serial I/O with Chip Select Specifications and Figure 16.1 shows a Block Diagram of Clock Synchronous Serial I/O with Chip Select.

Figures 16.2 to 16.9 show Clock Synchronous Serial I/O with Chip Select Associated Registers.

**Table 16.2 Clock Synchronous Serial I/O with Chip Select Specifications**

Item	Specification
Transfer Data Format	<ul style="list-style-type: none"> <li>Transfer-data length 8 bits</li> </ul> Continuous transmit and receive of serial data are enabled since both transmitter and receiver have buffer structure.
Operating Mode	<ul style="list-style-type: none"> <li>Clock synchronous communication mode</li> <li>4-wire bus communication mode (including bidirectional communication)</li> </ul>
Master / Slave Device	Selectable
I/O Pin	SSCK (I/O): Clock I/O pin SSI (I/O): Data I/O pin SSO (I/O): Data I/O pin $\overline{\text{SCS}}$ (I/O): Chip-select I/O pin
Transfer Clock	<ul style="list-style-type: none"> <li>When the MSS bit in the SSCRH register is set to 0 (operates as slave device), external clock can be selected.</li> <li>When the MSS bit in the SSCRH register is set to 1 (operates as master device), internal clock (selects from f1/256, f1/128, f1/64, f1/32, f1/16, f1/8 and f1/4 and outputs from SSCK pin) can be selected.</li> <li>Clock polarity and phase of SSCK can be selected.</li> </ul>
Receive Error Detection	<ul style="list-style-type: none"> <li>Overrun error</li> </ul> Overrun error occurs during receive and completes by error. While the RDRF bit in the SSSR register is set to 1 (data in the SSRDR register) and completing the next serial data receive, the ORER bit is set to 1.
Multimaster Error Detection	<ul style="list-style-type: none"> <li>Conflict error</li> </ul> While the SSUMS bit in the SSMR2 register is set to 1 (4-wire bus communication mode) and the MSS bit in the SSCRH register is set to 1 (operates as master device) and when starting a serial communication, the CE bit in the SSSR register is set to 1 if "L" applies to the $\overline{\text{SCS}}$ pin input. When the SSUMS bit in the SSMR2 register is set to 1 (4-wire bus communication mode), the MSS bit in the SSCRH register is set to 0 (operates as slave device) and the $\overline{\text{SCS}}$ pin input changes state from "L" to "H", the CE bit in the SSSR register is set to 1.
Interrupt Request	5 interrupt requests (transmit-end, transmit-data-empty, receive-data-full, overrun error and conflict error).(1)
Select Function	<ul style="list-style-type: none"> <li>Data transfer direction</li> <li>Selects MSB-first or LSB-first</li> <li>SSCK clock polarity</li> <li>Selects "L" or "H" level when clock stops</li> <li>SSCK clock phase</li> <li>Selects edge of data change and data download</li> </ul>

NOTE:

1. The interrupt vector table is one of the clock synchronous serial I/O with chip select specification.

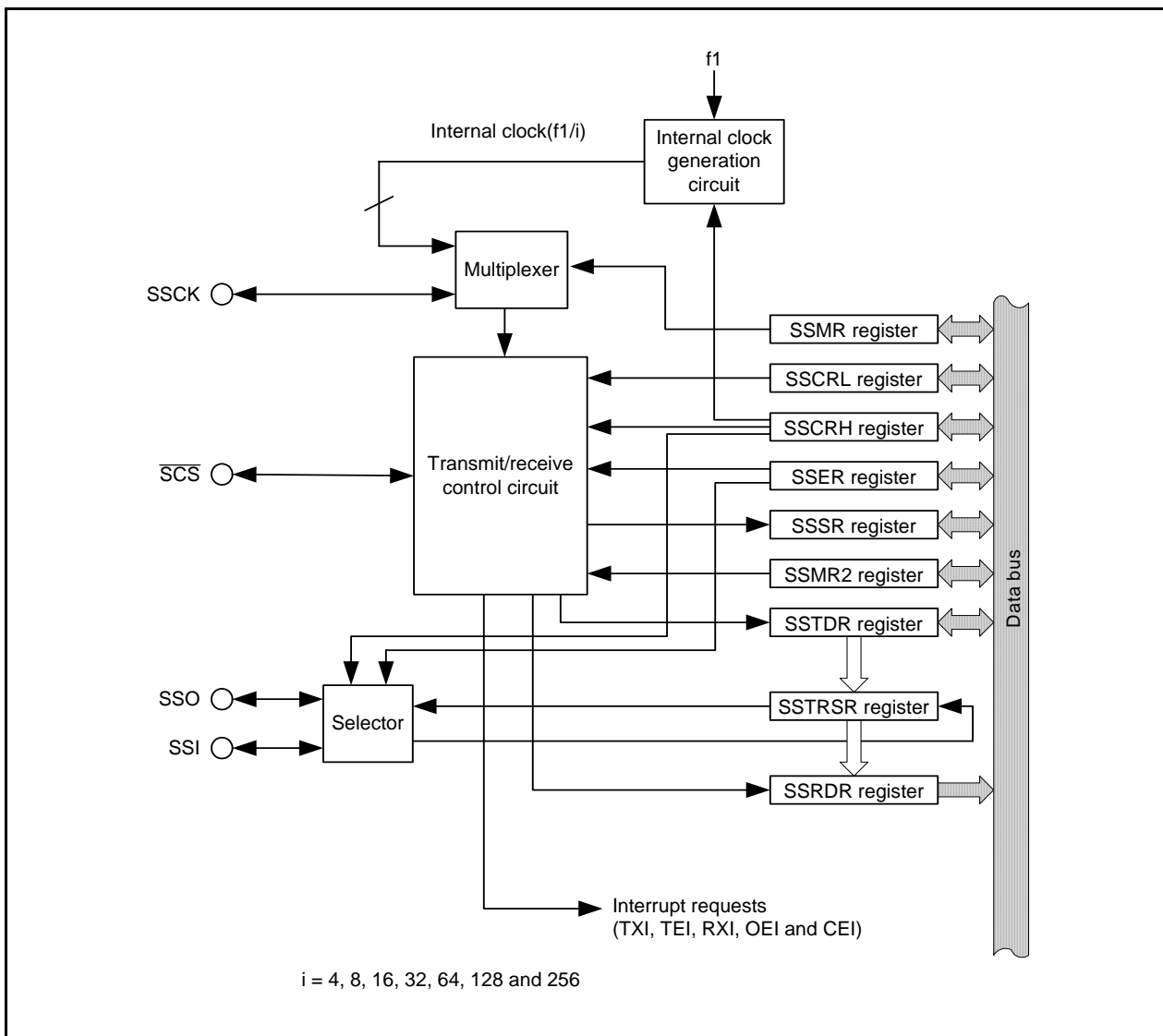
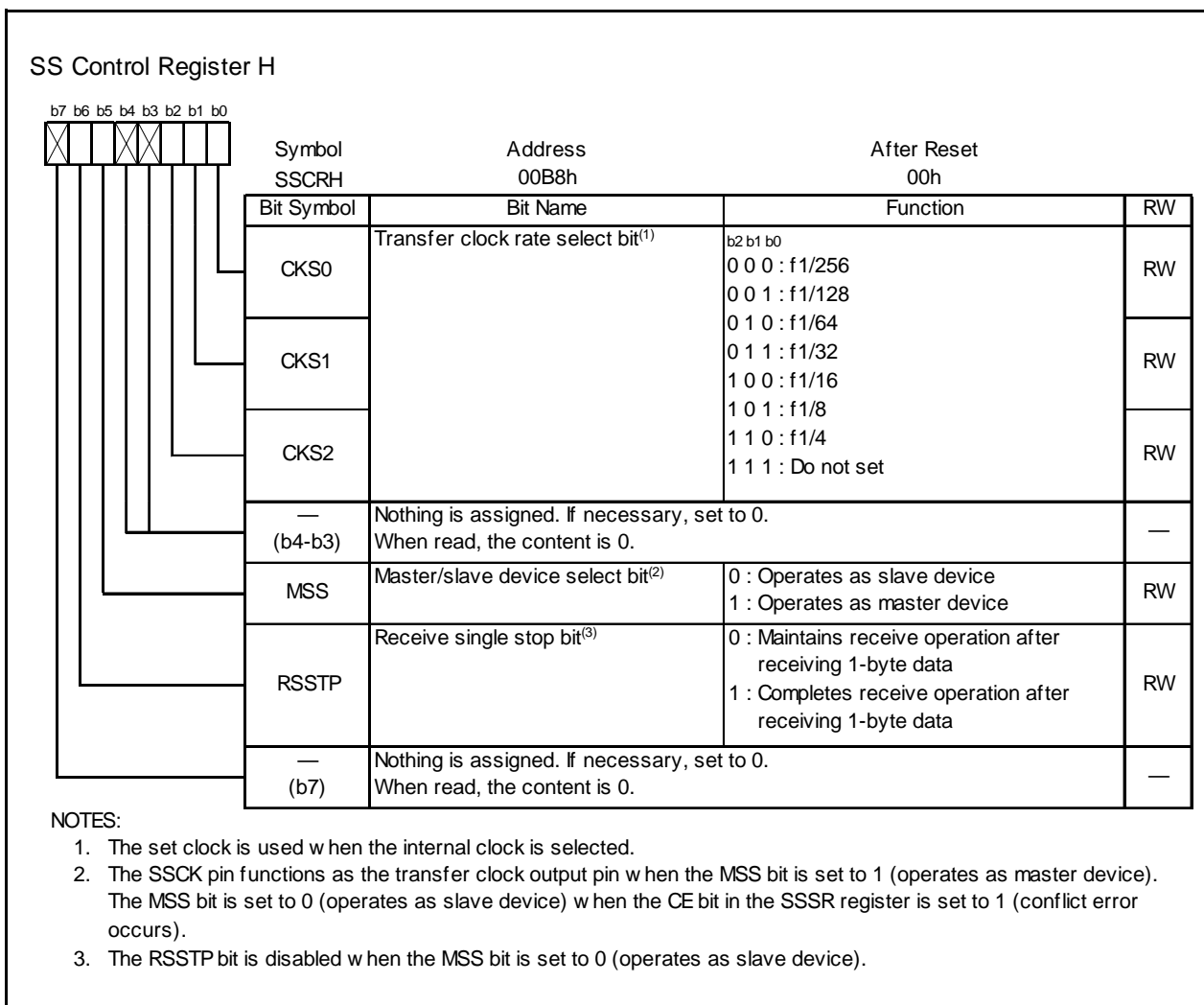
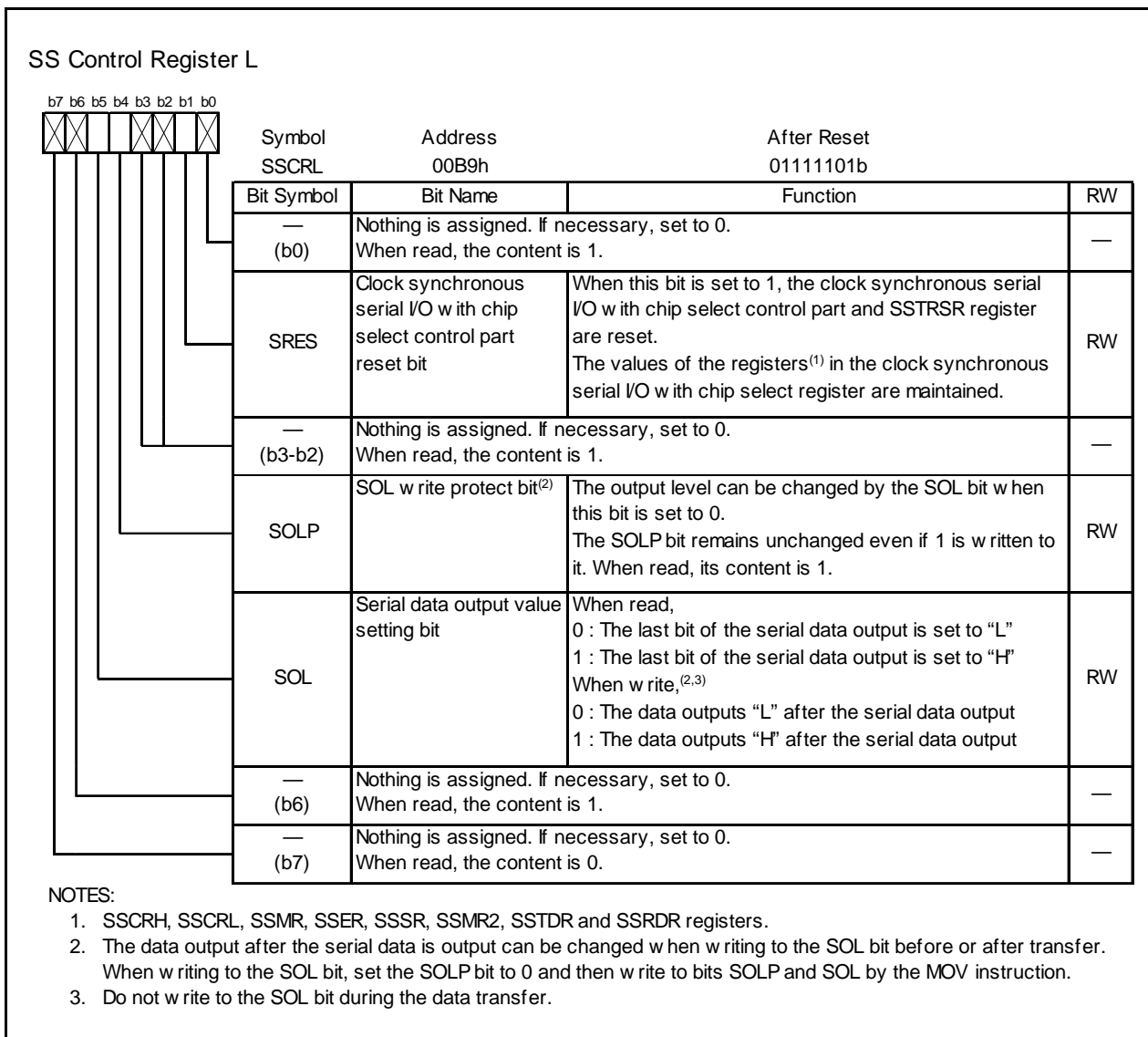


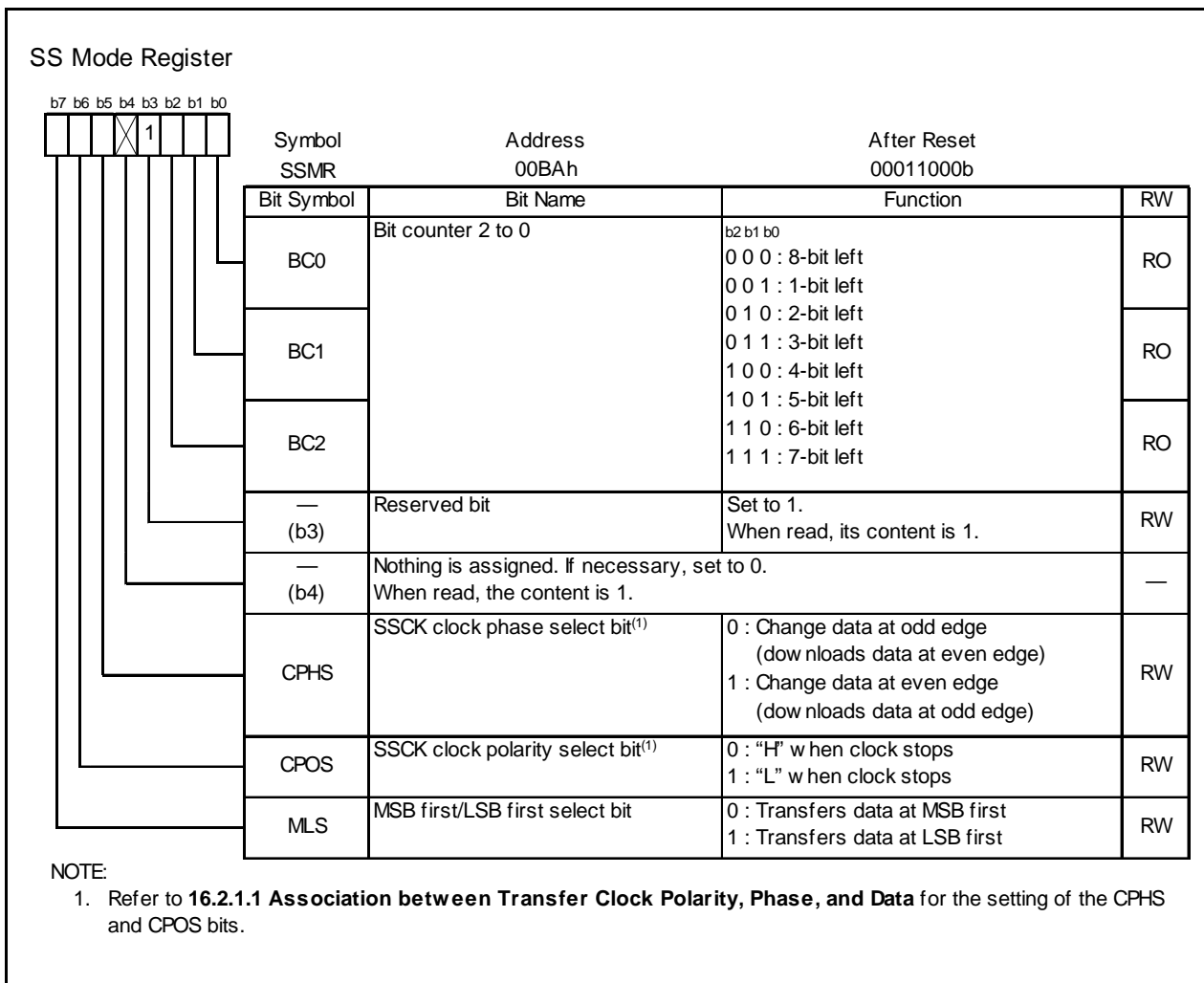
Figure 16.1 Block Diagram of Clock Synchronous Serial I/O with Chip Select



**Figure 16.2 SSCRH Register**



**Figure 16.3 SSCRL Register**



**Figure 16.4 SSMR Register**

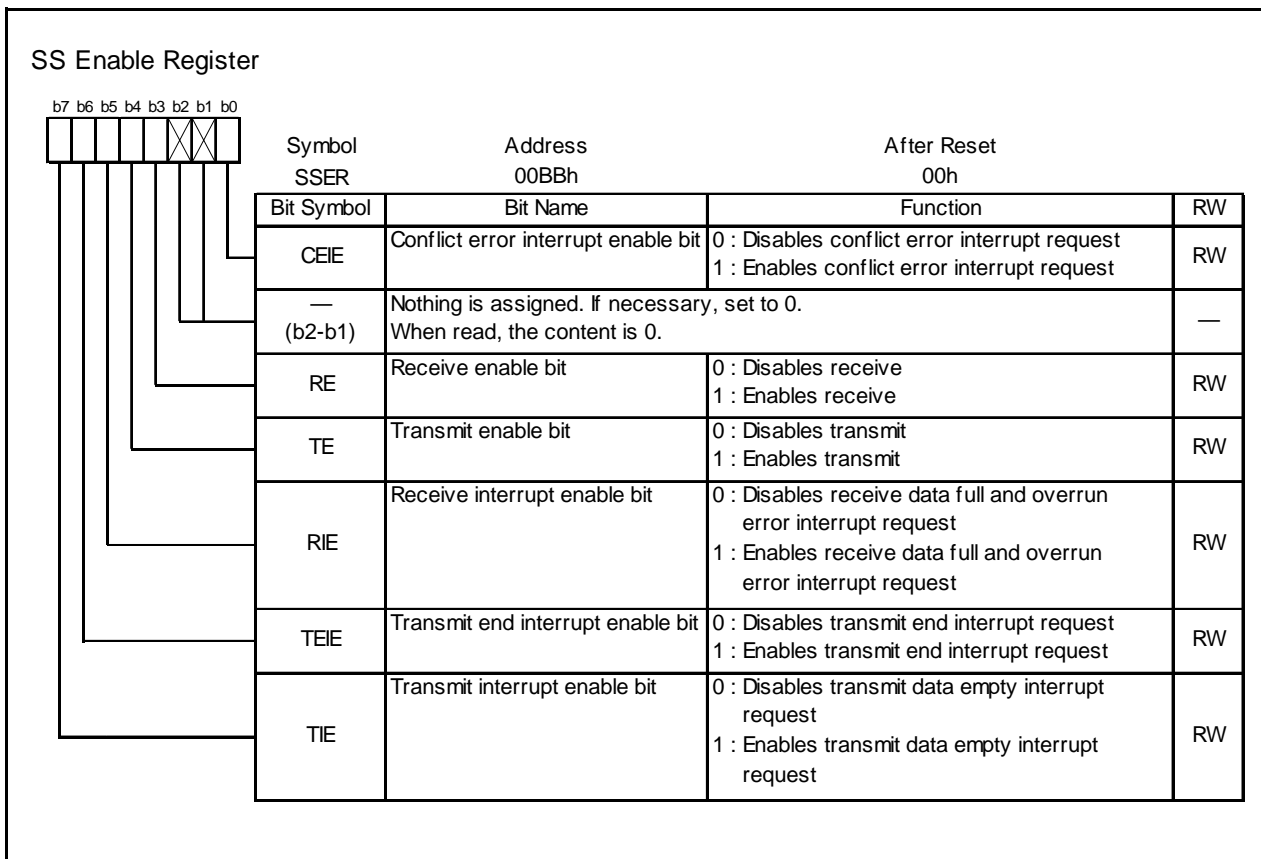


Figure 16.5 SSER Register



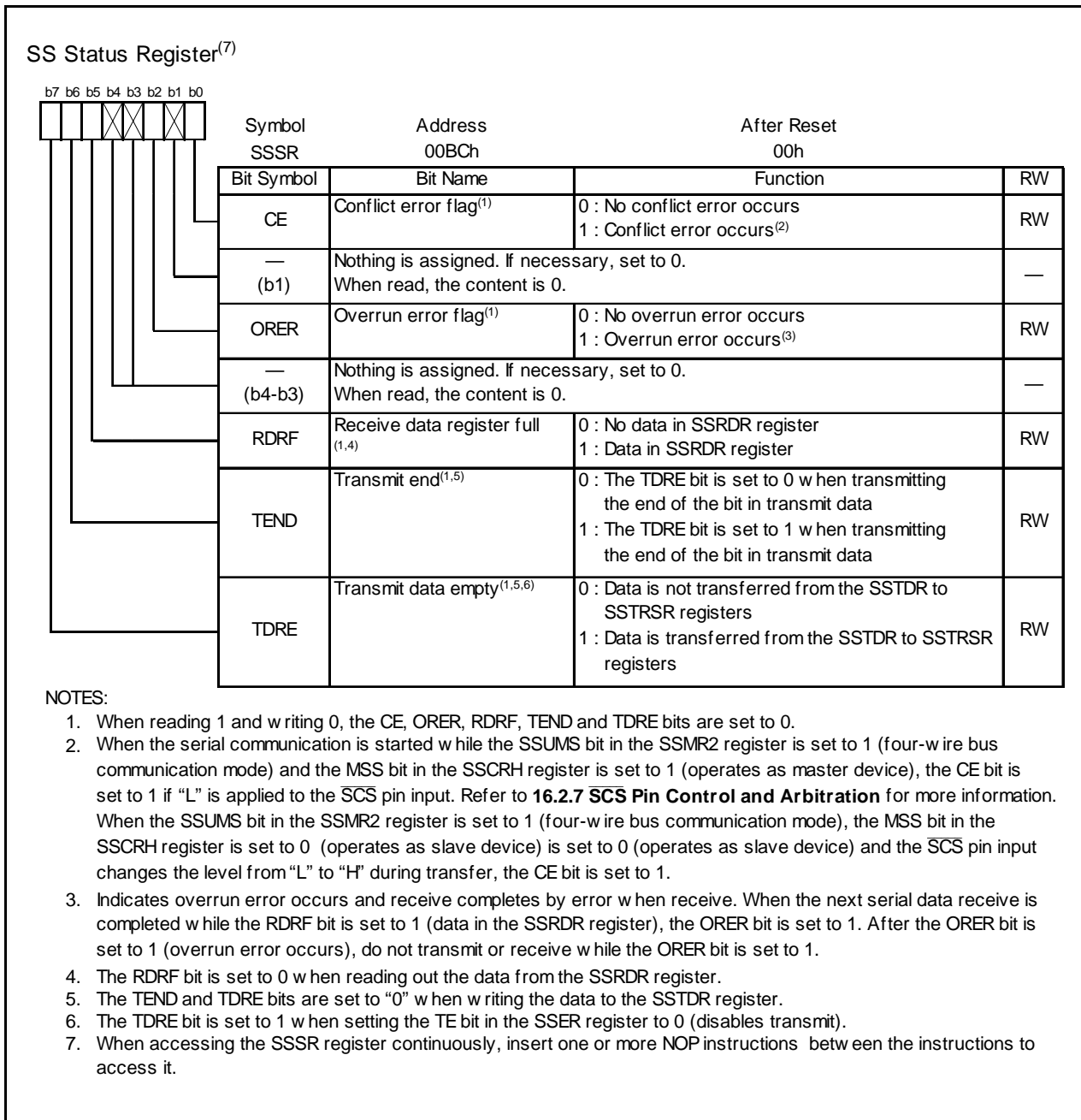
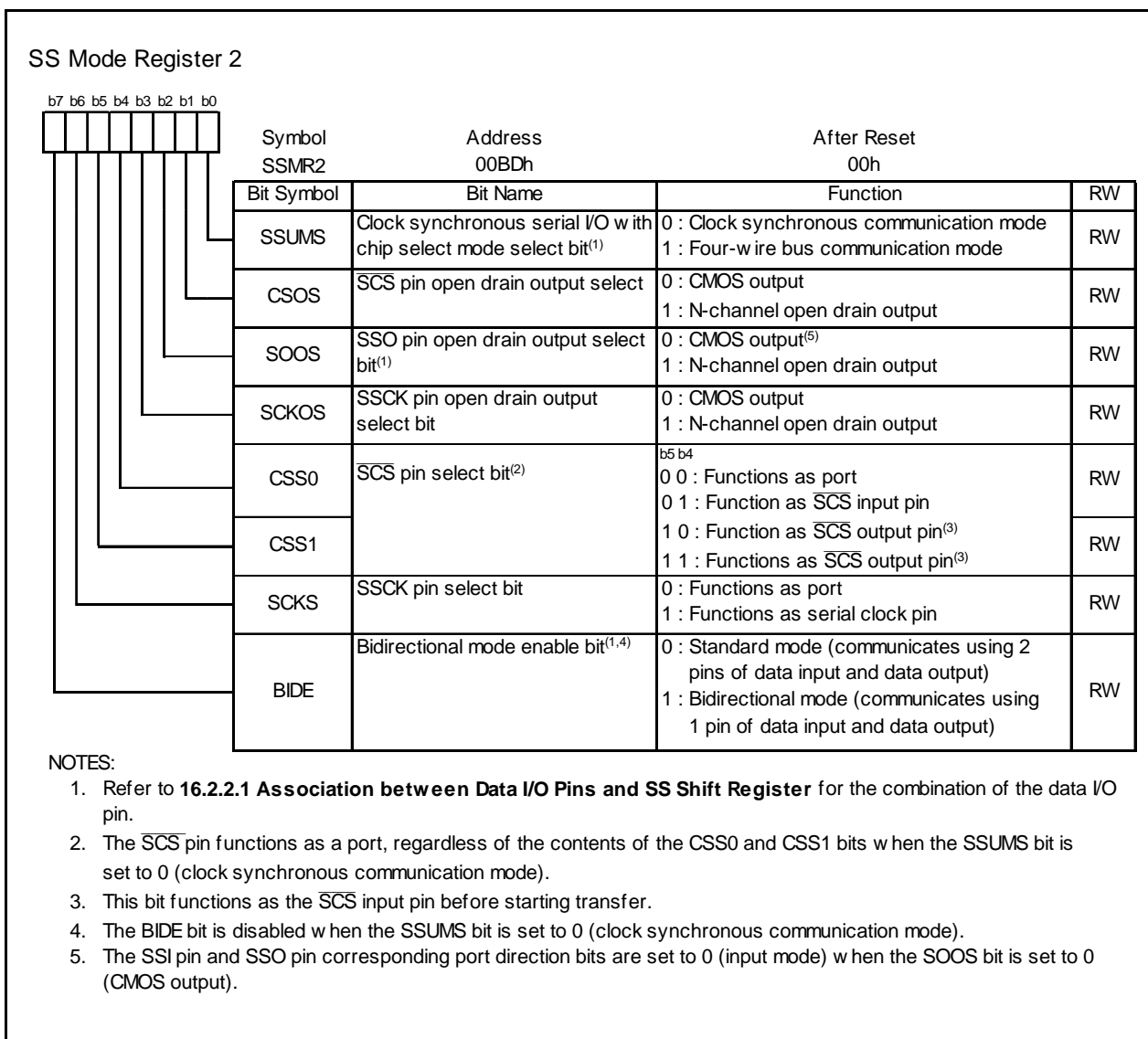


Figure 16.6 SSSR Register



**Figure 16.7 SSMR2 Register**

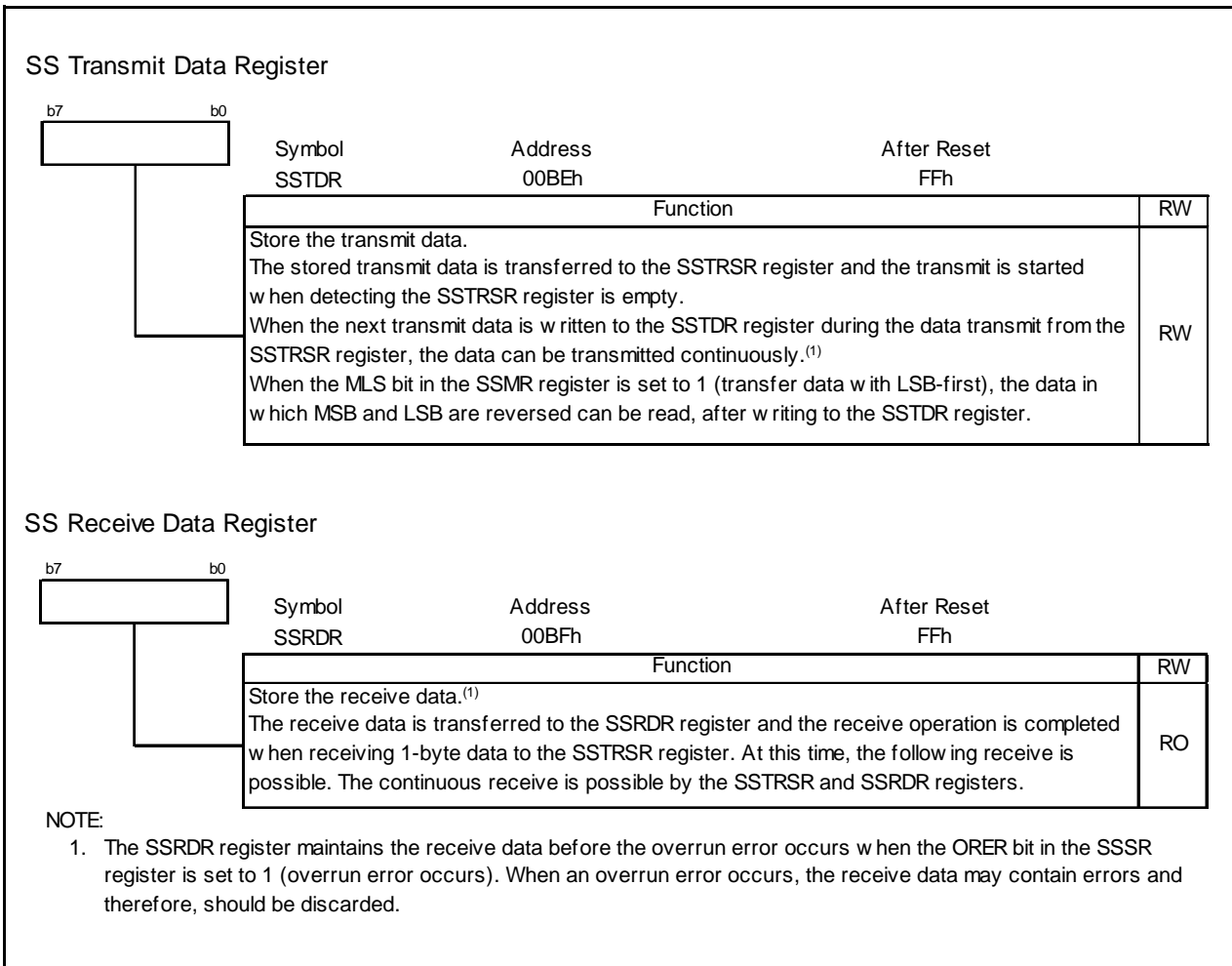


Figure 16.8 Registers SSTDR and SSRDR

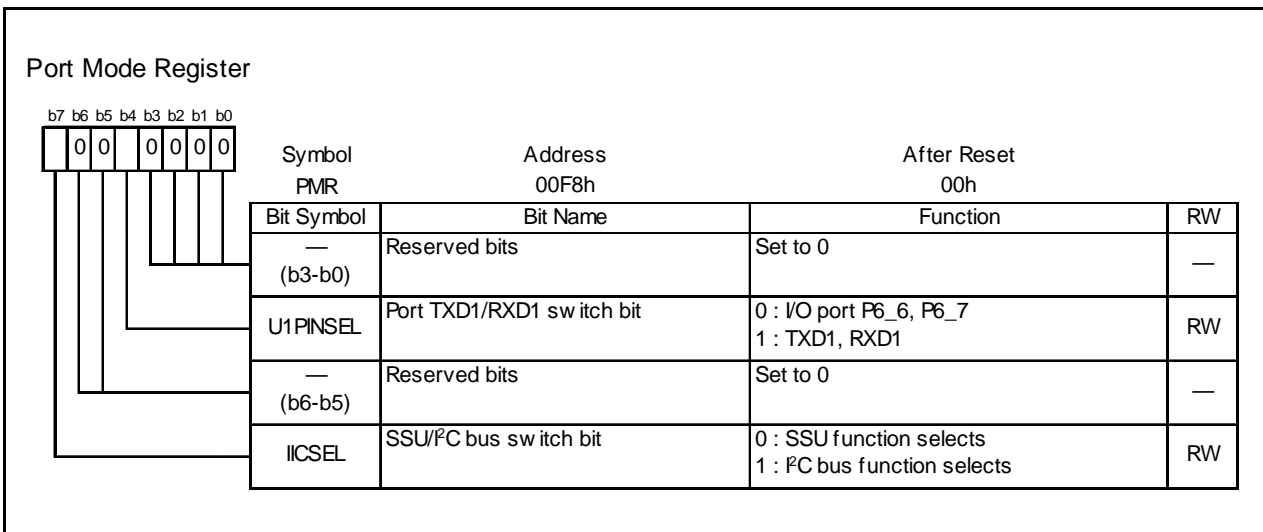


Figure 16.9 PMR Register

## 16.2.1 Transfer Clock

A transfer clock can be selected from 7 internal clocks ( $f_1/256$ ,  $f_1/128$ ,  $f_1/64$ ,  $f_1/32$ ,  $f_1/16$ ,  $f_1/8$  and  $f_1/4$ ) and an external clock.

When using the clock synchronous serial I/O with chip select, set the SCKS bit in the SSMR2 register to 1 and select the SSCK pin as the serial clock pin.

When the MSS bit in the SSCRH register is set to 1 (operates as master device), an internal clock can be selected and the SSCK pin functions as output. When transfer is started, the SSCK pin outputs clocks of the transfer rate selected in the CKS0 to CKS2 bits in the SSCRH register.

When the MSS bit in the SSCRH register is set to 0 (operates as slave device), an external clock can be selected and the SSCK pin functions as input.

### 16.2.1.1 Association between Transfer Clock Polarity, Phase, and Data

Association between transfer clock polarity, phase and data changes according to a combination of the SSUMS bit in the SSMR2 register and the CPHS and CPOS bits in the SSMR register.

Figure 16.10 shows the Association between Transfer Clock Polarity, Phase, and Transfer Data.

Also, the MSB-first transfer or LSB-first transfer can be selected by setting the MLS bit in the SSMR register. When the MLS bit is set to 1, transfer is started from the LSB to MSB. When the MLS bit is set to 0, transfer is started from the MSB to LSB.

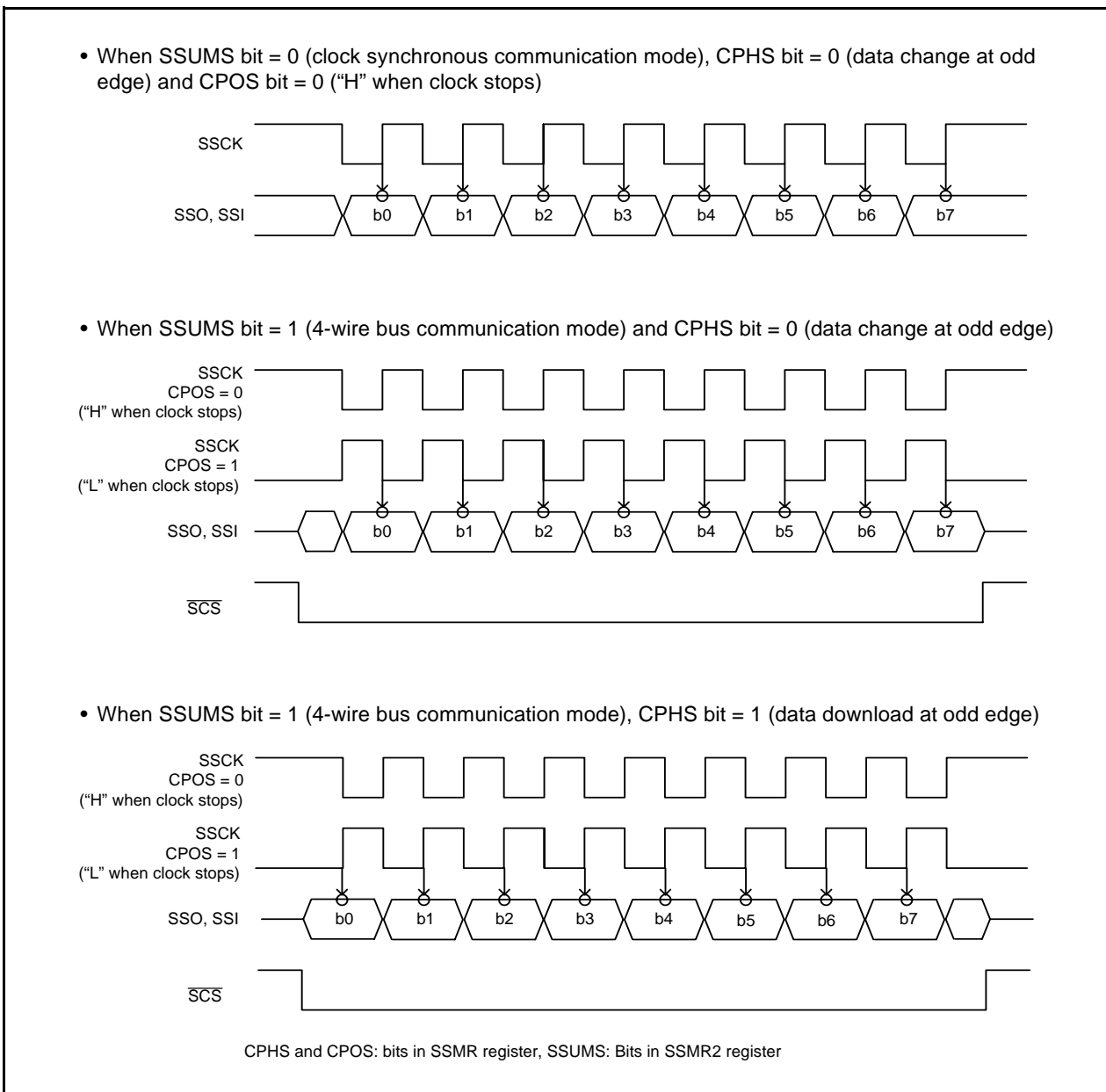


Figure 16.10 Association between Transfer Clock Polarity, Phase, and Transfer Data

## 16.2.2 SS Shift Register (SSTRSR)

The SSTRSR register is the shift register to transmit and receive the serial data.

When the transmit data is transferred from the SSTDR register to the SSTRSR register and the MLS bit in the SSMR register is set to 0 (MSB-first), the bit 0 in the SSTDR register is transferred to the bit 0 in the SSTRSR register. When the MLS bit is set to 1 (LSB-first), the bit 7 in the SSTDR register is transferred to the bit 0 in the SSTRSR register.

### 16.2.2.1 Association between Data I/O Pins and SS Shift Register

Connecting association between the data I/O pin and SSTRSR register (SS shift register) changes according to a combination of the MSS bit in the SSCRH register and the SSUMS bit in the SSMR2 register. Also, connecting association changes according to the BIDE bit in the SSMR2 register.

Figure 16.11 shows an Association between Data I/O Pins and SSTRSR Register.

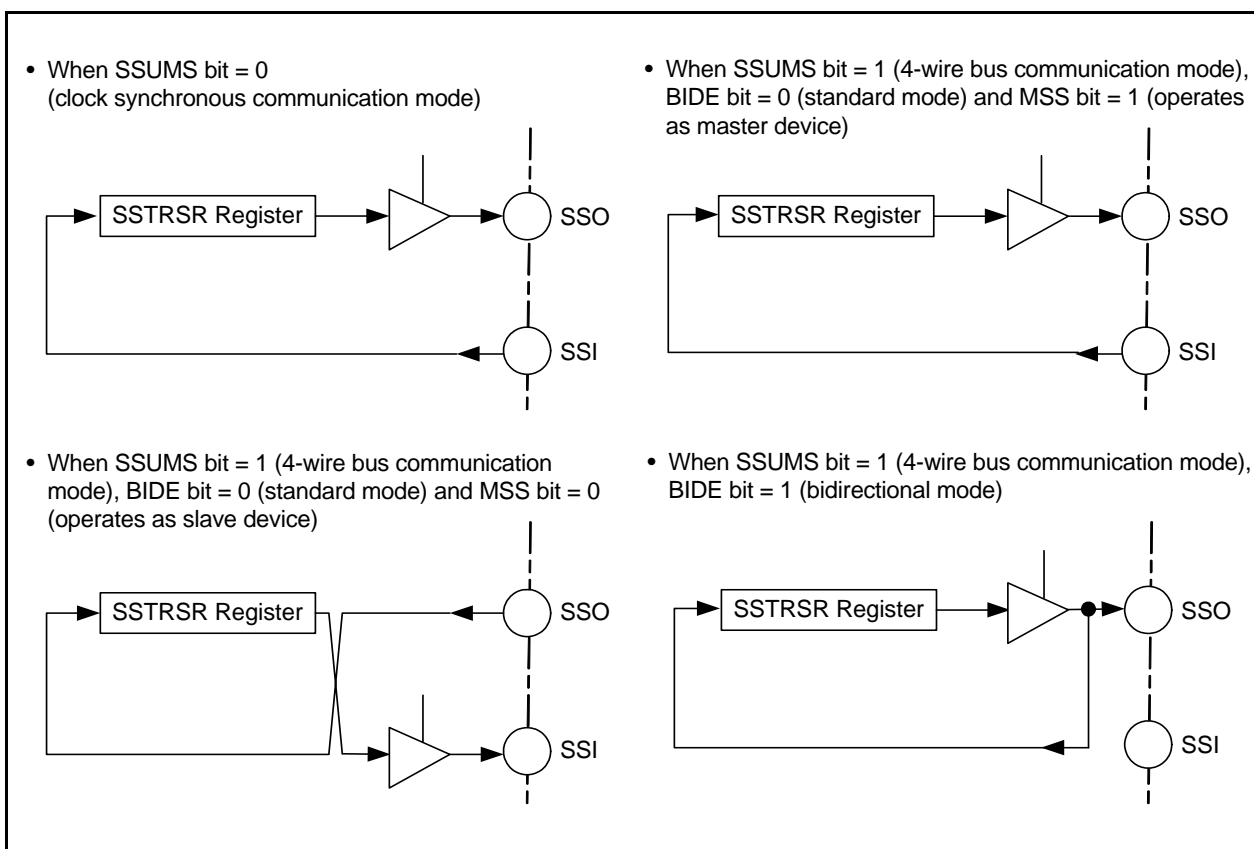


Figure 16.11 Association between Data I/O Pins and SSTRSR Register

### 16.2.3 Interrupt Requests

Clock synchronous serial I/O with chip select has five interrupt requests: transmit data empty, transmit end, receive data full, overrun error and conflict error. Since these interrupt requests are assigned to the clock synchronous serial I/O with chip select interrupt vector table, determining interrupt sources by flags is required. Table 16.3 shows the Clock Synchronous Serial I/O with Chip Select Interrupt Requests.

**Table 16.3 Clock Synchronous Serial I/O with Chip Select Interrupt Requests**

Interrupt Request	Abbreviation	Generation Condition
Transmit Data Empty	TXI	TIE = 1, TDRE = 1
Transmit End	TEI	TEIE = 1, TEND = 1
Receive Data Full	RXI	RIE = 1, RDRF = 1
Overrun Error	OEI	RIE = 1, ORER = 1
Conflict Error	CEI	CEIE = 1, CE = 1

CEIE, RIE, TEIE and TIE: Bits in SSER register

ORER, RDRF, TEND and TDRE: Bits in SSSR register

Generation conditions of Table 16.3 are met, a clock synchronous serial I/O with chip select interrupt request is generated. Set the each interrupt source to 0 by a clock synchronous serial I/O with chip select interrupt routine.

However, the TDRE and TEND bits are automatically set to 0 by writing the transmit data to the SSTDR register and the RDRF bit is automatically set to 0 by reading the SSRDR register. When writing the transmit data to the SSTDR register, at the same time the TDRE bit is set to 1 (data is transmitted from the SSTDR to SSTRSR registers) again and when setting the TDRE bit to 0 (data is not transmitted from the SSTDR to SSTRSR registers), additional 1-byte data may be transmitted.

### 16.2.4 Communication Modes and Pin Functions

Clock synchronous serial I/O with chip select switches functions of the I/O pin in each communication mode according to the setting of the MSS bit in the SSCRH register and the RE and TE bits in the SSER register.

Table 16.4 shows the Association between Communication Modes and I/O Pins.

**Table 16.4 Association between Communication Modes and I/O Pins**

Communication Mode	Bit Setting					Pin State		
	SSUMS	BIDE	MSS	TE	RE	SSI	SSO	SSCK
Clock Synchronous Communication Mode	0	Disabled	0	0	1	Input	–(1)	Input
				1	0	–(1)	Output	Input
				1	1	Input	Output	Input
			1	0	1	Input	–(1)	Output
				1	0	–(1)	Output	Output
				1	1	Input	Output	Output
4-Wire Bus Communication Mode	1	0	0	0	1	–(1)	Input	Input
				1	0	Output	–(1)	Input
				1	1	Output	Input	Input
			1	0	1	Input	–(1)	Output
				1	0	–(1)	Output	Output
				1	1	Input	Output	Output
4-Wire Bus (Bidirectional) Communication Mode <sup>(2)</sup>	1	1	0	0	1	–(1)	Input	Input
				1	0	–(1)	Output	Input
			1	0	1	–(1)	Input	Output
				1	0	–(1)	Output	Output

**NOTES:**

1. This pin can be used as programmable I/O port.
2. Do not set both the TE and RE bits to 1 in 4-wire bus (bidirectional) communication mode.

SSUMS and BIDE: Bits in SSMR2 register

MSS: Bit in SSCRH register

TE and RE: Bits in SSER register



## 16.2.5 Clock Synchronous Communication Mode

### 16.2.5.1 Initialization in Clock Synchronous Communication Mode

Figure 16.12 shows an Initialization in Clock Synchronous Communication Mode. Set the TE bit in the SSER register to 0 (disables transmit) and the RE bit to 0 (disables receive) before data transmit / receive as an initialization.

When communication mode and format are changed, set the TE bit to 0 and the RE bit to 0 before changing. Setting the RE bit to 0 does not change the contents of the RDRF and ORER flags, and the contents of the SSRDR register.

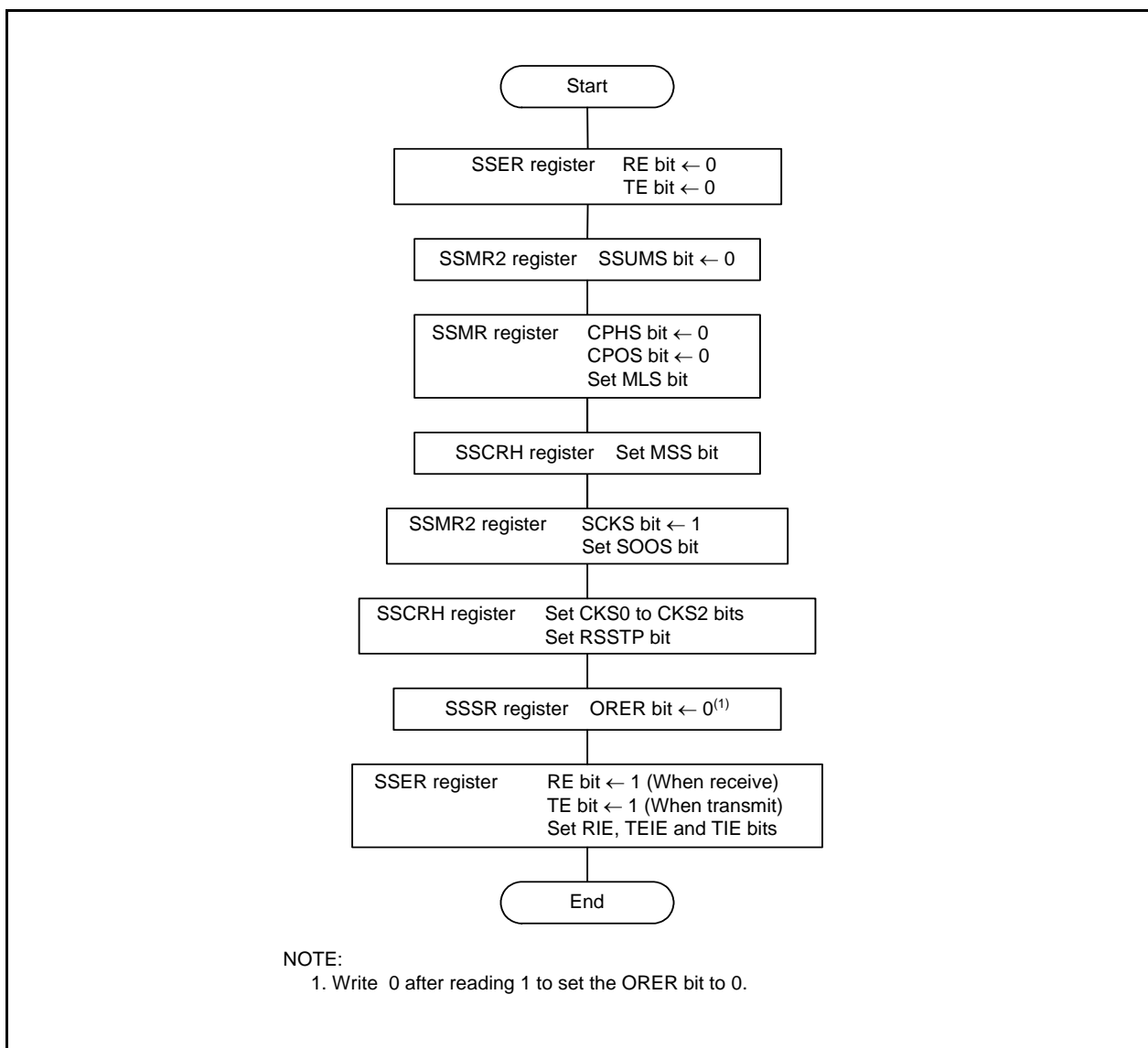


Figure 16.12 Initialization in Clock Synchronous Communication Mode

### 16.2.5.2 Data Transmission

Figure 16.13 shows an Example of Clock Synchronous Serial I/O with Chip Select Operation for Data Transmission (Clock Synchronous Communication Mode). During the data transmit, the clock synchronous serial I/O with chip select operates as described below.

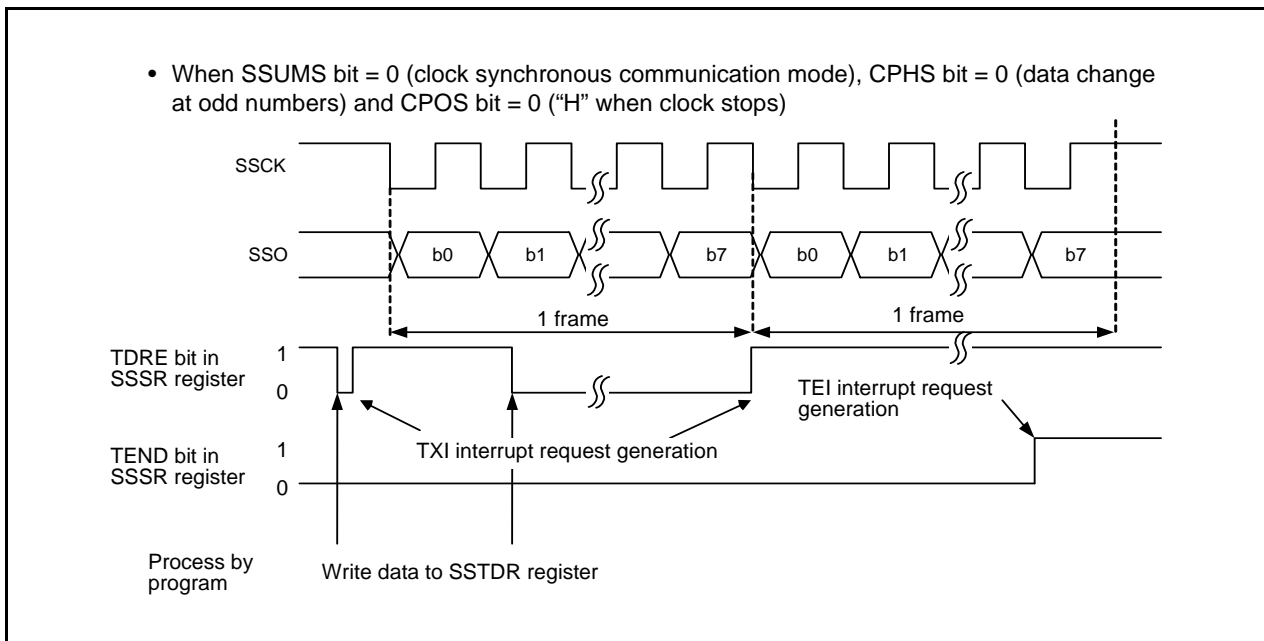
When the clock synchronous serial I/O with chip select is set as a master device, it outputs a synchronous clock and data.

When the clock synchronous serial I/O with chip select is set as a slave device, it outputs data synchronized with the input clock. When setting the TE bit to 1 (enables transmit) before writing the transmit data to the SSTDR register, the TDRE bit is automatically set to 0 (data is not transferred from the SSTDR to SSTRSR registers) and the data is transferred from the SSTDR to SSTRSR registers.

After the TDRE bit is set to 1 (data is transferred from the SSTDR to SSTRSR registers), a transmit is started. When the TIE bit in the SSER register is set to 1, the TXI interrupt request is generated. When one frame of data is transferred while the TDRE bit is set to 0, data is transferred from the SSTDR to SSTRSR registers and a transmit of the next frame is started. If the 8th bit is transmitted while the TDRE bit is set to 1, the TEND bit in the SSSR register is set to 1 (the TDRE bit is set to 1 when the last bit of the transmit data is transmitted) and the state is retained. The TEI interrupt request is generated when the TEIE bit in the SSER register is set to 1 (enables transmit-end interrupt request). The SSCK pin is retained "H" after transmit-end.

Transmit can not be performed while the ORER bit in the SSSR register is set to 1 (overrun error occurs). Confirm that the ORER bit is set to 0 before transmit.

Figure 16.14 shows a Sample Flowchart of Data Transmission (Clock Synchronous Communication Mode).



**Figure 16.13 Example of Clock Synchronous Serial I/O with Chip Select Operation for Data Transmission (Clock Synchronous Communication Mode)**

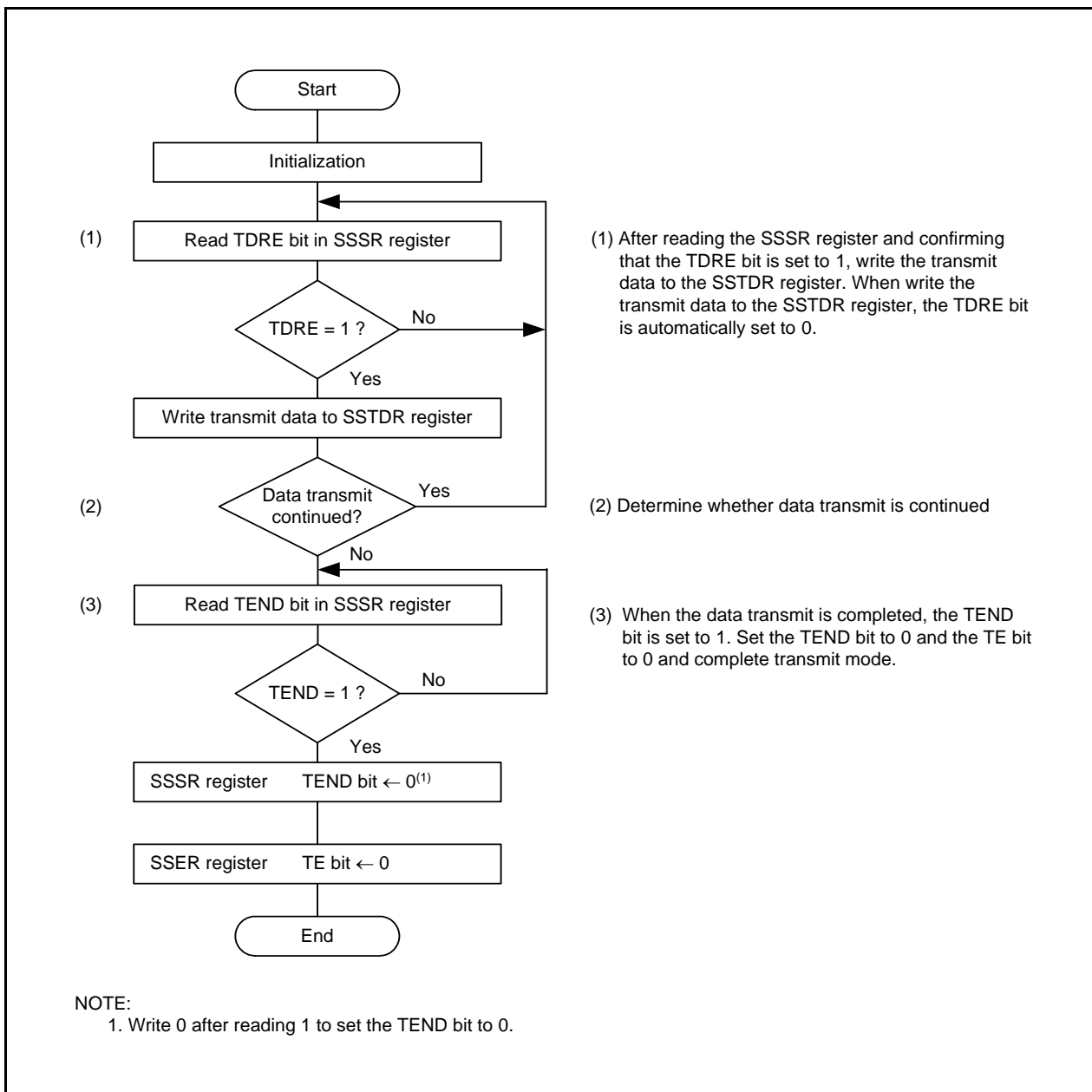


Figure 16.14 Sample Flowchart of Data Transmission (Clock Synchronous Communication Mode)

### 16.2.5.3 Data Reception

Figure 16.15 shows an Example of Clock Synchronous Serial I/O with Chip Select Operation for Data Reception (Clock Synchronous Communication Mode).

During the data receive, the clock synchronous serial I/O with chip select operates as described below. When the clock synchronous serial I/O with chip select is set as a master device, it outputs a synchronous clock and inputs data.

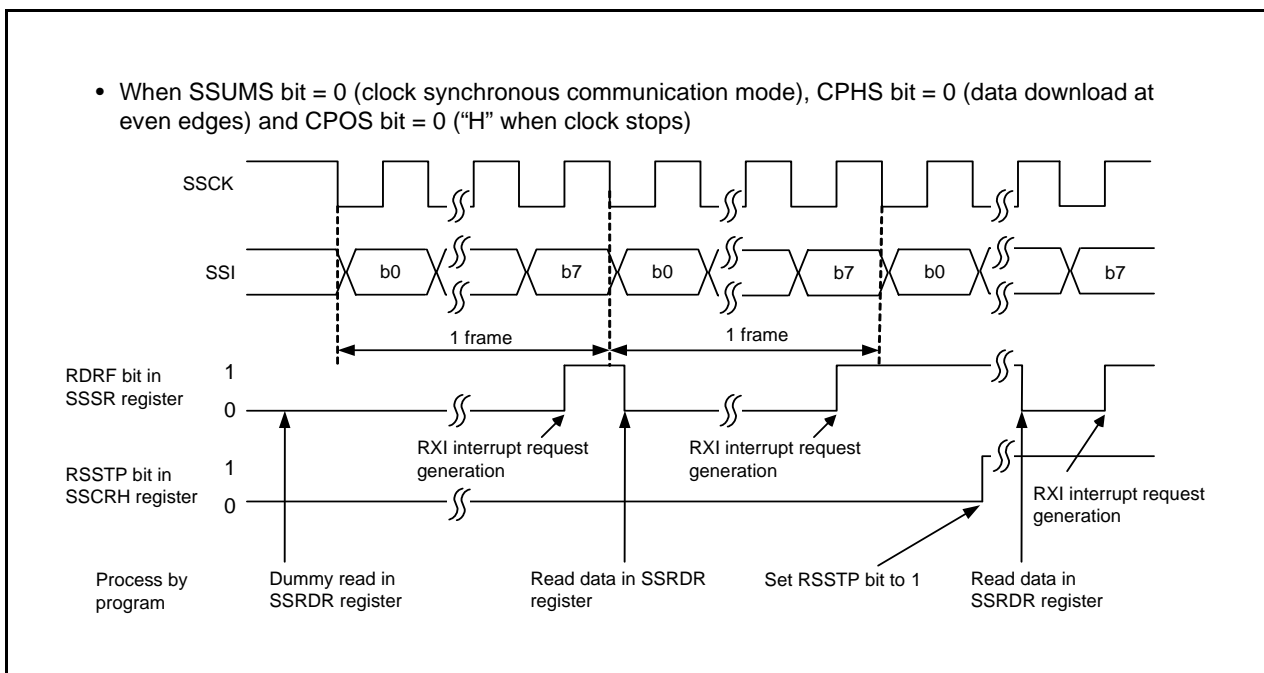
When the clock synchronous serial I/O with chip select is set as a slave device, it outputs data synchronized with the input clock. When the clock synchronous serial I/O with chip select is set as a master device, it outputs a receive clock and starts receiving by performing dummy read on the SSRDR register.

After the 8-bit data is received, the RDRF bit in the SSSR register is set to 1 (data in the SSRDR register) and receive data is stored in the SSRDR register. When the RIE bit in the SSER register is set to 1 (enables RXI and OEI interrupt request), the RXI interrupt request is generated. If the SSRDR register is read, the RDRF bit is automatically set to 0 (no data in the SSRDR register).

Read the receive data after setting the RSSTP bit in the SSCRH register to 1 (after receiving 1-byte data, the receive operation is completed). The clock synchronous serial I/O with chip select outputs a clock for receiving 8-bit data and stops. After that, set the RE bit in the SSER register to 0 (disables receive) and the RSSTP bit to 0 (receive operation is continued after receiving the 1-byte data) and read the receive data. If the SSRDR register is read while the RE bit is set to 1 (enables receive), a receive clock is output again.

When the 8th clock rises while the RDRF bit is set to 1, the ORER bit in the SSSR register is set to 1 (overflow error occurs: OEI) and the operation is stopped. When the ORER bit is set to 1, receive can not be performed. Confirm that the ORER bit is set to 0 before restarting receive.

Figure 16.16 shows a Sample Flowchart of Data Reception (MSS = 1) (Clock Synchronous Communication Mode).



**Figure 16.15 Example of Clock Synchronous Serial I/O with Chip Select Operation for Data Reception (Clock Synchronous Communication Mode)**

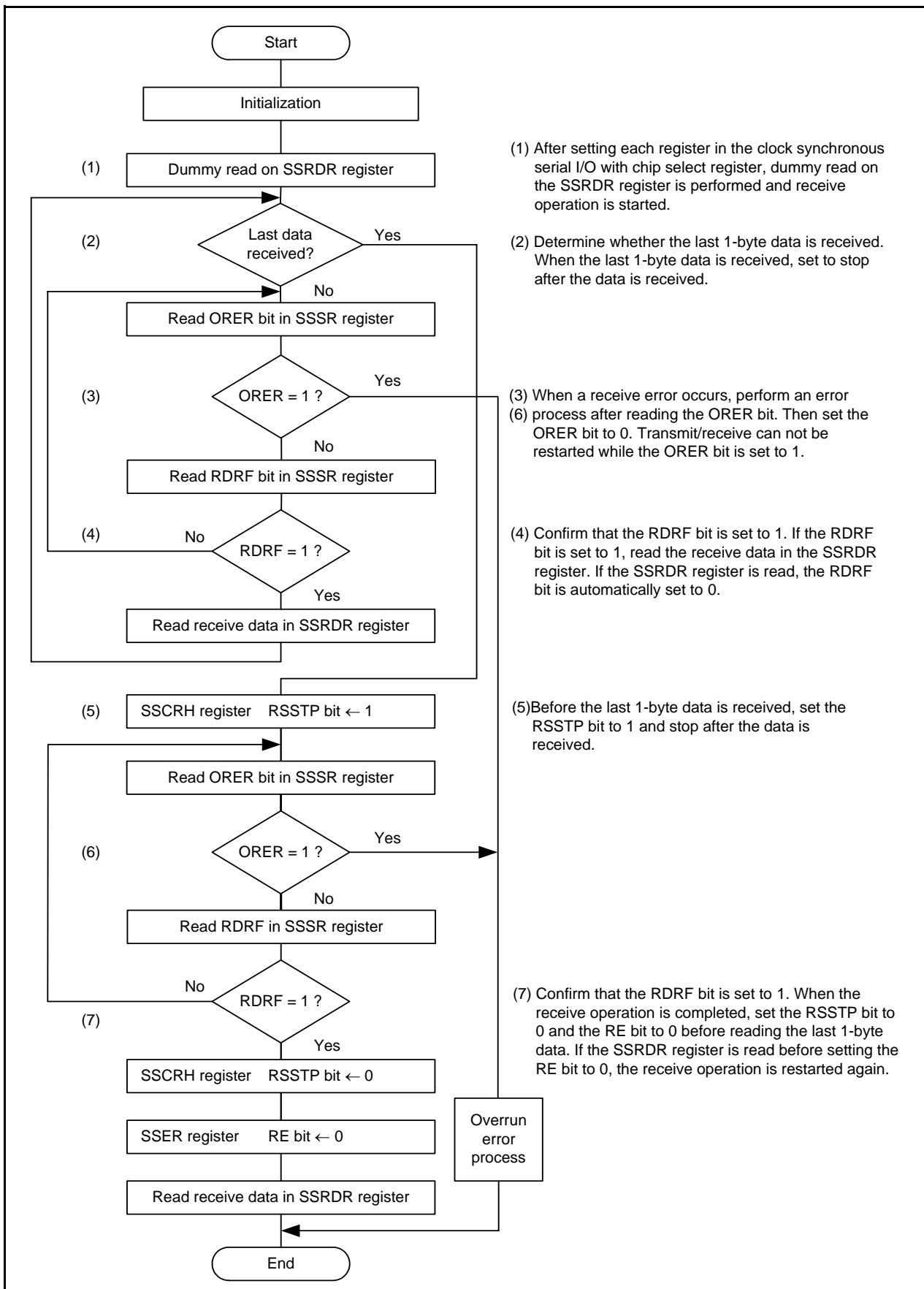


Figure 16.16 Sample Flowchart of Data Reception (MSS = 1) (Clock Synchronous Communication Mode)

#### 16.2.5.4 Data Transmission/Reception

Data transmit/receive is a combined operation of data transmit and receive which are described before. Transmit/receive is started by writing data in the SSTDR register.

When the 8th clock rises or the ORER bit is set to 1 (overrun error occurs) while the TDRE bit is set to 1 (data is transferred from the SSTDR to SSTRSR registers), the transmit/receive operation is stopped.

When switching from transmit mode (TE = 1) or receive mode (RE = 1) to transmit/receive mode (TE = RE = 1), set the TE bit to 0 and RE bit to 0 before switching. After confirming that the TEND bit is set to 0 (the TDRE bit is set to 0 when the last bit of the transmit data is transmitted), the RERF bit is set to 0 (no data in the SSRDR register) and the ORER bit is set to 0 (no overrun error), set the TE and RE bits to 1.

Figure 16.17 shows a Sample Flowchart of Data Transmission/Reception (Clock Synchronous Communication Mode).

When exiting transmit/receive mode after this mode is used (TE = RE = 1), a clock may be output if transmit/receive mode is exited after reading the SSRDR register. To avoid any clock outputs, perform either of the following:

- First set the RE bit to 0, and then set the TE bit to 0.
- Set bits TE and RE at the same time.

When subsequently switching to receive mode (TE = 0 and RE = 1), first set the SRES bit to 1, and set this bit to 0 to reset the clock synchronous serial interface control unit and the SSTRSR register. Then, set the RE bit to 1.

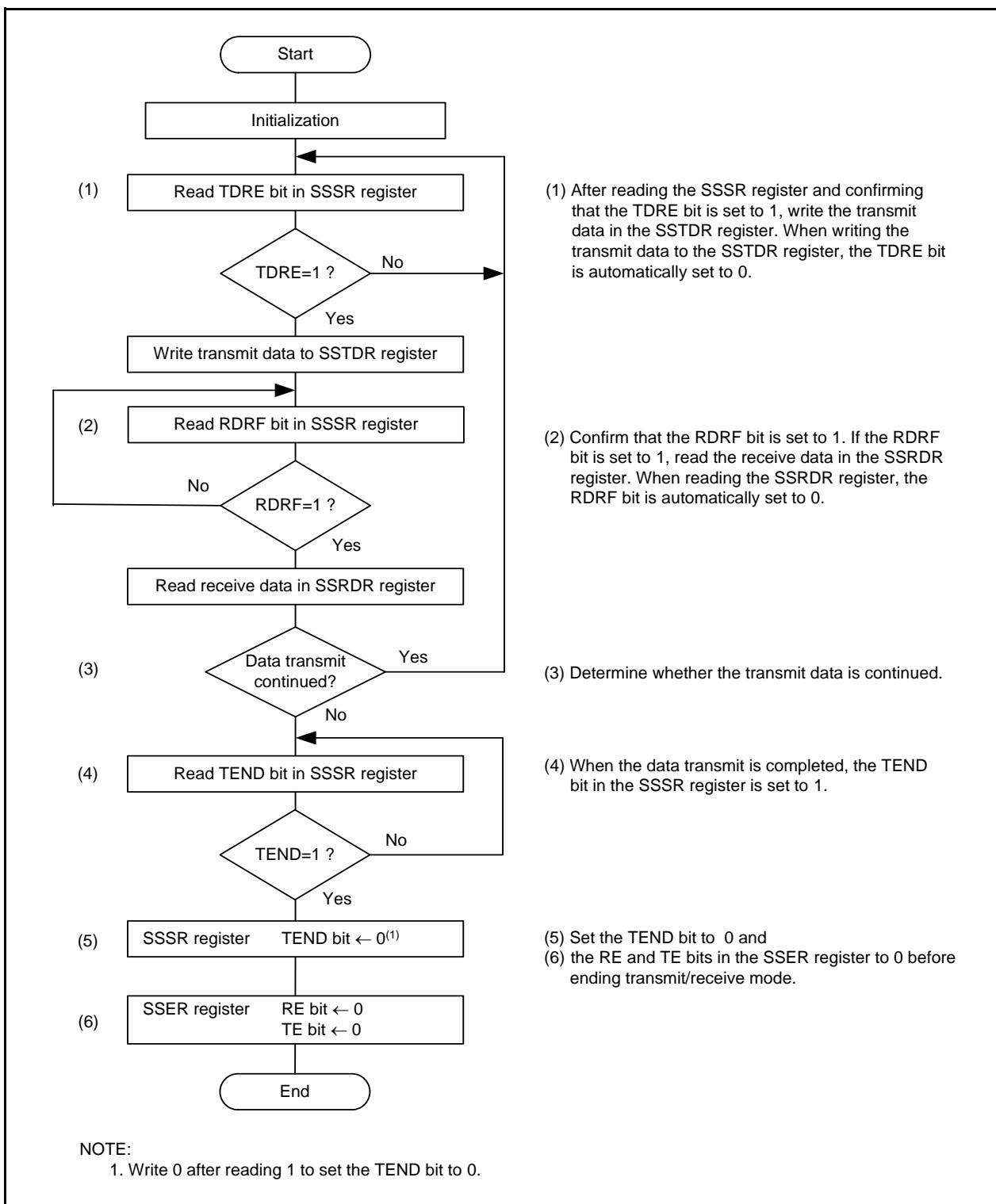


Figure 16.17 Sample Flowchart of Data Transmission/Reception (Clock Synchronous Communication Mode)

### 16.2.6 Operation in 4-Wire Bus Communication Mode

4-wire bus communication mode is a mode which communicates with the 4-wire bus; a clock line, data input line, data output line and chip select line. This mode includes bidirectional mode in which the data input line and data output line function as a single pin.

The data input line and output line are changed according to the setting of the MSS bit in the SSCRH register and the BIDE bit in the SSMR2 register. For details, refer to **16.2.2.1 Association between Data I/O Pins and SS Shift Register**. In this mode, association between the clock polarity, phase and data can be set by the CPOS and CPHS bits in the SSMR register. For details, refer to **16.2.1.1 Association between Transfer Clock Polarity, Phase, and Data**.

When the clock synchronous serial I/O with chip select is set as a master device, the chip select line controls output. When the clock synchronous serial I/O with chip select is set as a slave device, the chip select line controls input. When the clock synchronous serial I/O with chip select is set as master device, the chip select line controls output of the  $\overline{\text{SCS}}$  pin or controls output of a general port by setting the CSS1 bit in the SSMR2 register. When the clock synchronous serial I/O with chip select is set as a slave device, the chip select line set the  $\overline{\text{SCS}}$  pin as an input pin by setting the CSS1 and CSS0 bits in the SSMR2 register to 01b.

In 4-wire bus communication mode, the MLS bit in the SSMR register is set to 0 and communication is performed using the MSB-first.

#### 16.2.6.1 Initialization in 4-Wire Bus Communication Mode

Figure 16.18 shows an Initialization in 4-Wire Bus Communication Mode. Before the data transit/receive, set the TE bit in the SSER register to 0 (disables transmit) and the RE bit in the SSER register to 0 (disables receive) and initialize the clock synchronous serial I/O with chip select.

When communication mode and format are changed, set the TE bit to 0 and the RE bit to 0 before changing.

Setting the RE bit to 0 does not change the contents of the RDRF and ORER flags, and the contents of the SSRDR register.



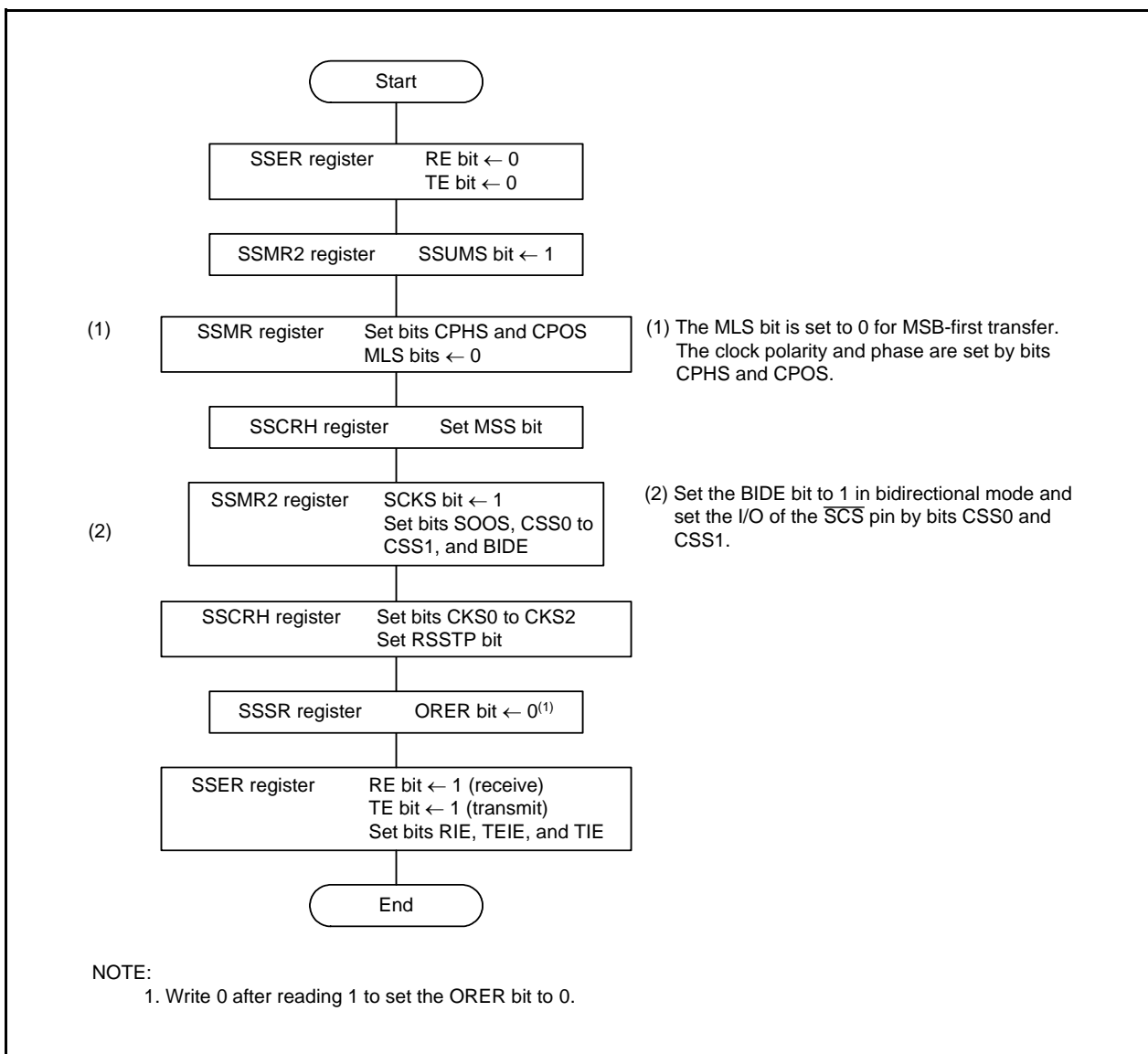


Figure 16.18 Initialization in 4-Wire Bus Communication Mode

### 16.2.6.2 Data Transmission

Figure 16.19 shows an Example of Clock Synchronous Serial I/O with Chip Select Operation during Data Transmission (4-Wire Bus Communication Mode). During the data transmit, the clock synchronous serial I/O with chip select operates as described below.

When the clock synchronous serial I/O with chip select is set as a master device, it outputs a synchronous clock and data. When the UUSA is set as a slave device, it outputs data in synchronized with the input clock while “L” applies to the  $\overline{\text{SCS}}$  pin.

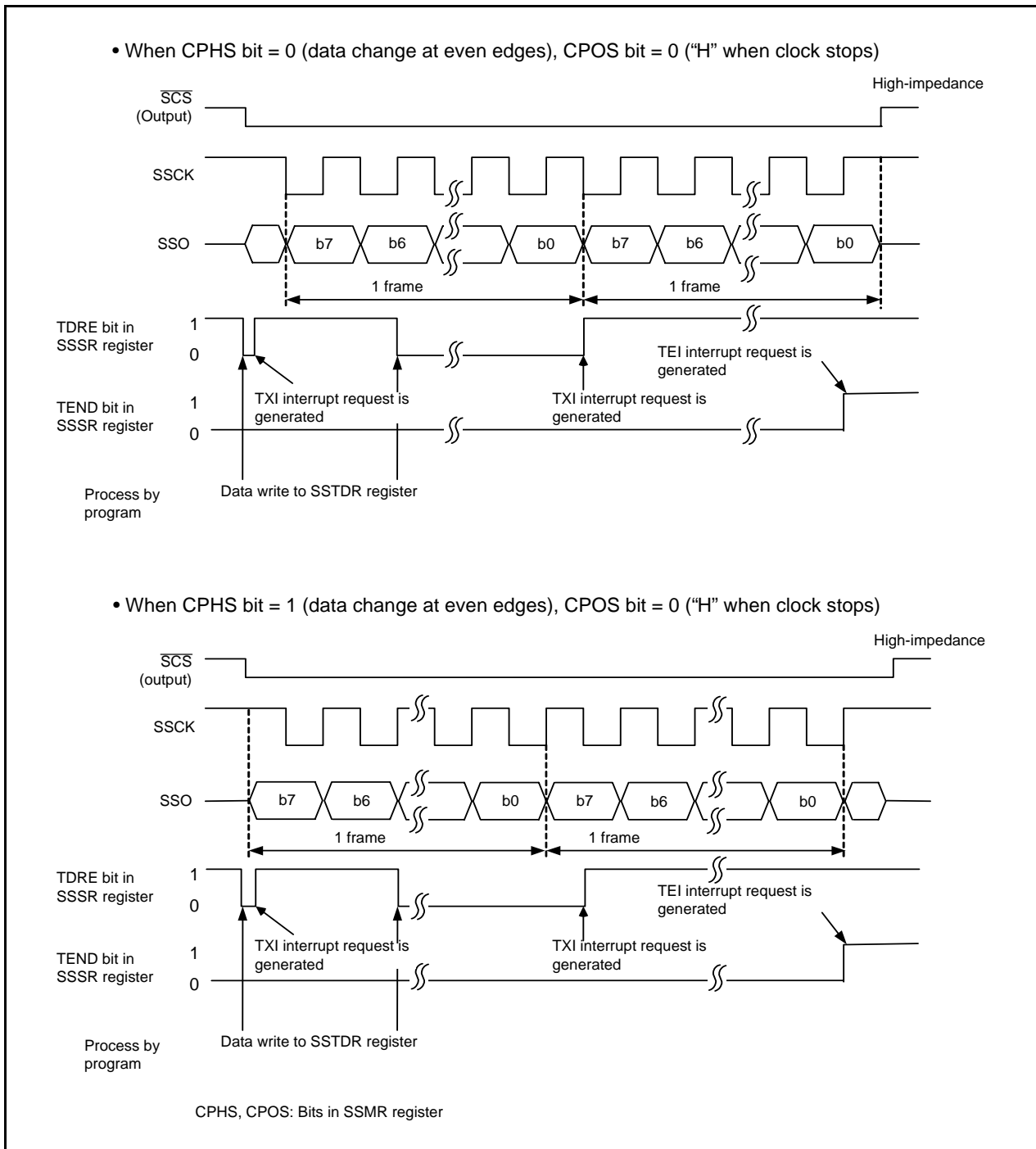
When writing the transmit data to the SSTDR register after setting the TE bit to 1 (enables transmit), the TDRE bit is automatically set to 0 (data is not transferred from the SSTDR to SSTRSR registers) and the data is transferred from the SSTDR to SSTRSR registers. After the TDRE bit is set to 1 (data is transferred from the SSTDR to SSTRSR registers), a transmit is started. When the TIE bit in the SSER register is set to 1, the TXI interrupt request is generated.

When the 1-frame data is transferred while the TDRE bit is set to 0, the data is transferred from the SSTDR to SSTRSR registers and the next frame transmit is started. If the 8th bit is transmitted while the TDRE is set to 1, the TEND in the SSSR register is set to 1 (when the last bit of the transmit data is transmitted, the TDRE bit is set to 1) and the state is retained. If the TEIE bit in the SSER register is set to 1 (enables transmit-end interrupt request), the TEI interrupt request is generated. The SSCK pin is retained “H” after transmit-end and the  $\overline{\text{SCS}}$  pin is held “H”. When the  $\overline{\text{SCS}}$  pin is transmitted When transmitting continuously while the  $\overline{\text{SCS}}$  pin is held “L”, write the next transmit data to the SSTDR register before transmitting the 8th bit.

Transmit can not be performed while the ORER bit in the SSSR register is set to 1 (overrun error occurs). Confirm that the ORER bit is set to 0 before transmit.

The difference from the clock synchronous communication mode is that the SSO pin is placed in high-impedance state while the  $\overline{\text{SCS}}$  pin is placed in high-impedance state when operating as a master device and the SSI pin is placed in high-impedance state while the  $\overline{\text{SCS}}$  pin is placed in “H” input state when operating as a slave device.

A sample flowchart is the same as the clock synchronous communication mode (refer to **Figure 16.14 Sample Flowchart of Data Transmission (Clock Synchronous Communication Mode)**).



**Figure 16.19 Example of Clock Synchronous Serial I/O with Chip Select Operation during Data Transmission (4-Wire Bus Communication Mode)**

### 16.2.6.3 Data Reception

Figure 16.20 shows an Example of Clock Synchronous Serial I/O with Chip Select Operation during Data Reception (4-Wire Bus Communication Mode). During the data receive, the clock synchronous serial I/O with chip select operates as described below.

When the clock synchronous serial I/O with chip select is set as a master device, it outputs a synchronous clock and inputs data. When the clock synchronous serial I/O with chip select is set as a slave device, it outputs data synchronized with the input clock while the  $\overline{SCS}$  pin is held "L" input. When the clock synchronous serial I/O with chip select is set as a master device, it outputs a receive clock and starts receiving by performing dummy read on the SSRDR register.

After the 8-bit data is received, the RDRF bit in the SSSR register is set to 1 (data in the SSRDR register) and receive data is stored in the SSRDR register. When the RIE bit in the SSER register is set to 1 (enables RXI and OEI interrupt request), the RXI interrupt request is generated. If the SSRDR register is read, the RDRF bit is automatically set to 0 (no data in the SSRDR register).

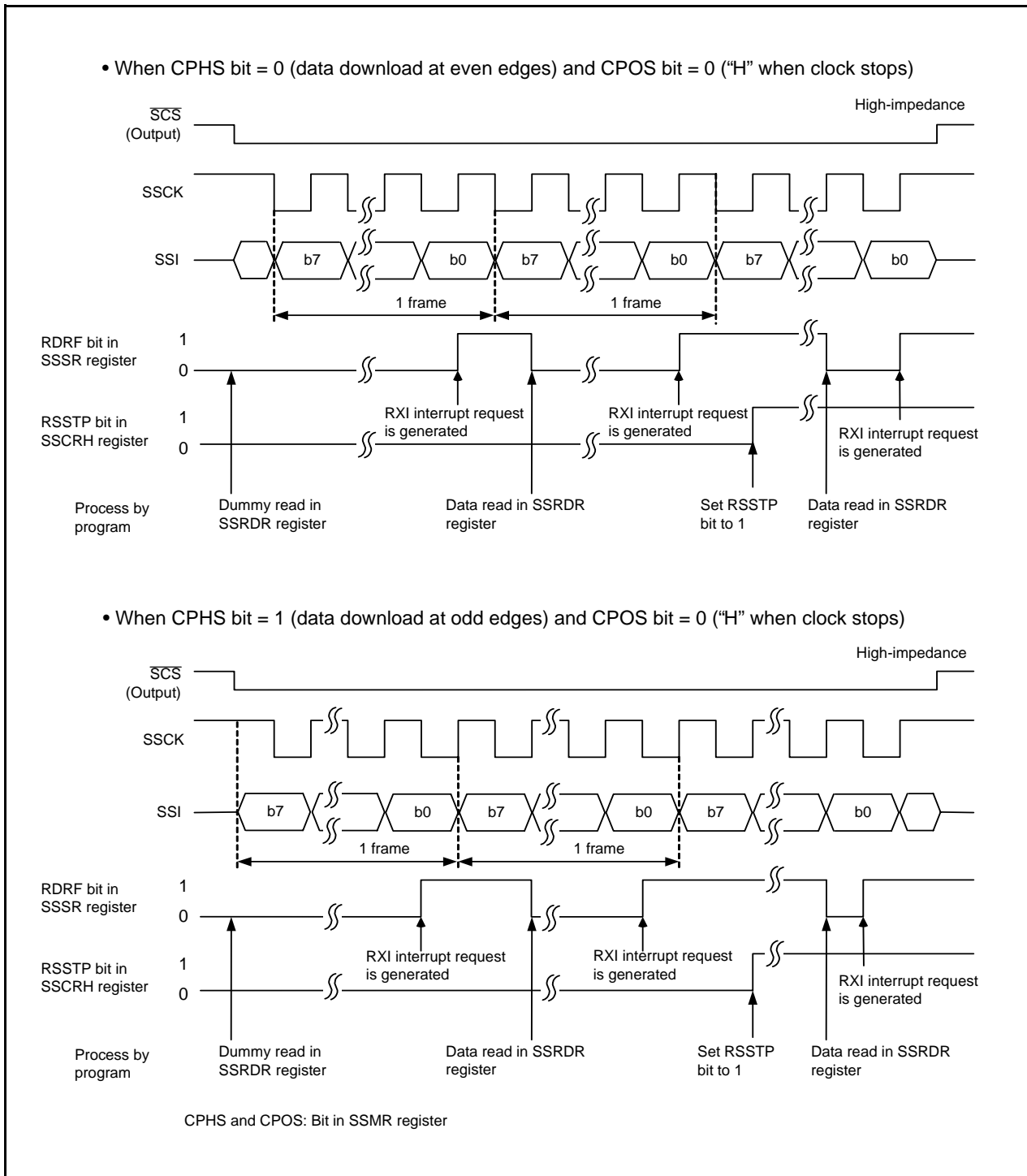
Read the receive data after setting the RSSTP bit in the SSCRH register to 1 (after receiving 1-byte data, the receive operation is completed). The clock synchronous serial I/O with chip select outputs a clock for receiving 8-bit data and stops. After that, set the RE bit in the SSER register to 0 (disables receive) and the RSSTP bit to 0 (receive operation is continued after receiving 1-byte data) and read the receive data. If the SSRDR register is read while the RE bit is set to 1 (enables receive), a receive clock is output again.

When the 8th clock rises while the RDRF bit is set to 1, the ORER bit in the SSSR register is set to 1 (overrun error occurs: OEI) and the operation is stopped. When the ORER bit is set to 1, receive can not be performed. Confirm that the ORER bit is set to 0 before restarting receive.

When the RDRF and ORER bits are set to 1, it varies depending on setting the CPHS bit in the SSMR register. Figure 16.20 shows when the RDRF and ORER bits are set to 1.

When the CPHS bit is set to 1 (data download at the odd edges), the RDRF and ORER bits are set to 1 at one point of a frame.

A sample flowchart is the same as the clock synchronous communication mode (refer to **Figure 16.16 Sample Flowchart of Data Reception (MSS = 1) (Clock Synchronous Communication Mode)**).



**Figure 16.20 Example of Clock Synchronous Serial I/O with Chip Select Operation during Data Reception (4-Wire Bus Communication Mode)**

### 16.2.7 $\overline{\text{SCS}}$ Pin Control and Arbitration

When setting the SSUMS bit in the SSMR2 register to 1 (4-wire bus communication mode), and the CSS1 bit in the SSMR2 register to 1 (functions as  $\overline{\text{SCS}}$  output pin), Set the MSS bit in the SSCRH register to 1 (operates as a master device) and check the arbitration of the  $\overline{\text{SCS}}$  pin before starting serial transfer. If the clock synchronous serial I/O with chip select detects that the synchronized internal  $\overline{\text{SCS}}$  signal is held "L" in this period, the CE bit in the SSSR register to 1 (a conflict error occurs) and the MSS bit is automatically set to 0 (operates as a slave device).

Figure 16.21 shows an Arbitration Check Timing.

A future transmit operation is not performed while the CE bit is set to 1. Set the CE bit to 0 (a conflict error does not occur) before a transmit is started.

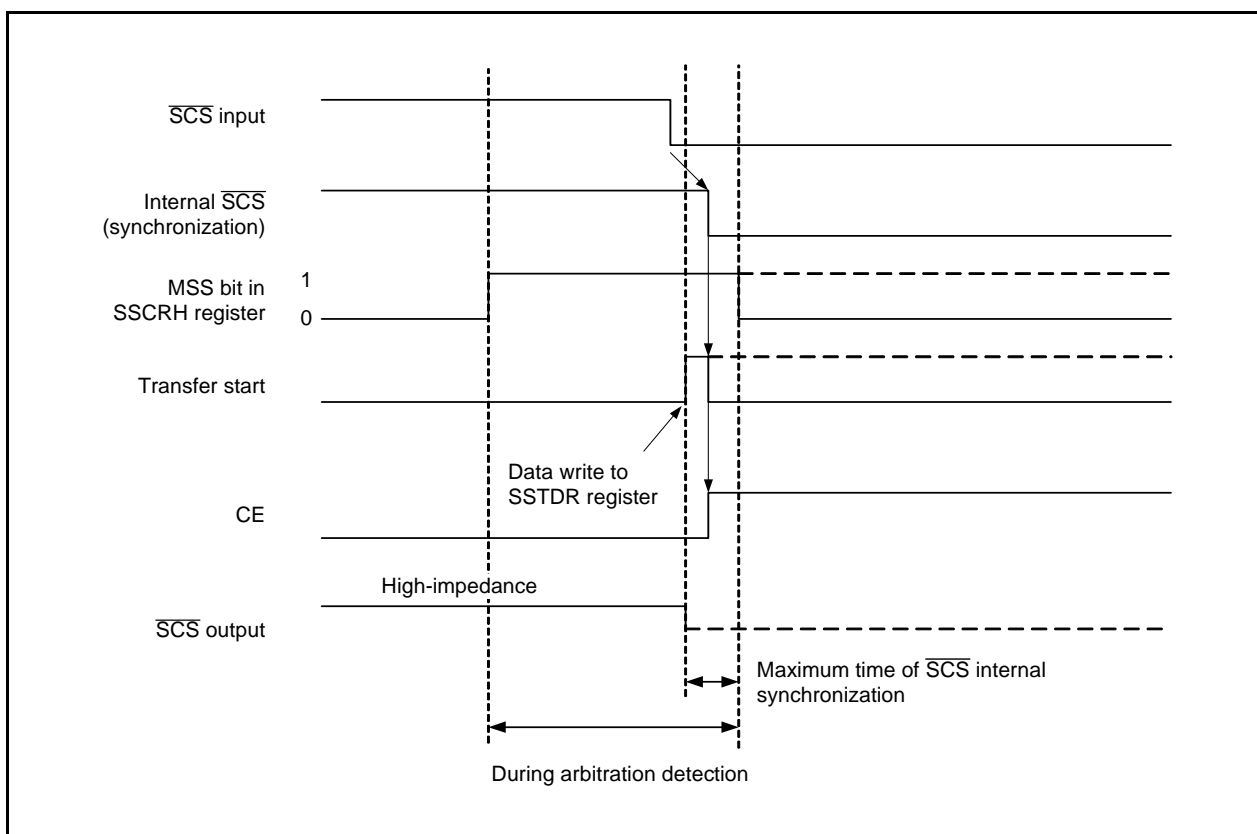


Figure 16.21 Arbitration Check Timing

### 16.2.8 Notes on Clock Synchronous Serial I/O with Chip Select

Set the IICSEL bit in the PMR register to 0 (select clock synchronous serial I/O with chip select function) to use the clock synchronous serial I/O with chip select.

### 16.3 I<sup>2</sup>C Bus Interface

The I<sup>2</sup>C bus interface is the circuit which is used for a serial communication based on the data transfer format of the Philips I<sup>2</sup>C bus.

Table 16.5 lists a I<sup>2</sup>C bus Interface Specifications, Figure 16.22 shows a Block Diagram of I<sup>2</sup>C bus Interface and Figure 16.23 shows the External Circuit Connection Example of Pins SCL and SDA. Figures 16.24 to 16.31 show the registers associated with the I<sup>2</sup>C bus interface.

\* I<sup>2</sup>C bus is a trademark of Koninklijke Philips Electronics N. V.

**Table 16.5 I<sup>2</sup>C Bus Interface Specifications**

Item	Specification
Communication Formats	<ul style="list-style-type: none"> <li>• I<sup>2</sup>C bus format               <ul style="list-style-type: none"> <li>- Selectable for master / slave device</li> <li>- Continuous transmit / receive (since the shift register, transmit data register and receive data register are independent)</li> <li>- Start / stop conditions are automatically generated in master mode</li> <li>- Automatic loading of acknowledge bit when transmit</li> <li>- Bit synchronization / wait function (in master mode, the state of the SCL signal is monitored per bit and the timing is synchronized automatically. If the transfer is not possible yet, stand by to set the SCL signal to "L".</li> <li>- Direct drive of the SCL and SDA pins (N-channel open drain output) is enabled</li> </ul> </li> <li>• Clock synchronous serial format               <ul style="list-style-type: none"> <li>- Continuous transmit / receive (since the shift register, transmit data register and receive data register are independent)</li> </ul> </li> </ul>
I/O Pins	SCL (I/O): Serial clock I/O pin SDA (I/O): Serial data I/O pin
Transfer Clocks	<ul style="list-style-type: none"> <li>• When the MST bit in the ICCR1 register is set to 0 The external clock (input from the SCL pin)</li> <li>• When the MST bit in the ICCR1 register is set to 1 The internal clock selected by the CKS0 to CKS3 bits in the ICCR1 register (output from the SCL pin)</li> </ul>
Receive Error Detection	<ul style="list-style-type: none"> <li>• Detects overrun error (clock synchronous serial format) An overrun error occurs during receive. When the last bit of the following data is received while the RDRF bit in the ICSR register is set to 1 (data in the ICDRR register), the AL bit is set to 1.</li> </ul>
Interrupt Sources	<ul style="list-style-type: none"> <li>• I<sup>2</sup>C bus format ..... 6 types<sup>(1)</sup> Transmit data empty (including when slave address matches), transmit ends, receive data full (including when slave address matches), arbitration lost, NACK detection and stop condition detection.</li> <li>• Clock synchronous serial format ..... 4 types<sup>(1)</sup> Transmit data empty, transmit ends, receive data full and overrun error</li> </ul>
Select Functions	<ul style="list-style-type: none"> <li>• I<sup>2</sup>C bus format               <ul style="list-style-type: none"> <li>- Selectable for the output level of the acknowledge signal when receive</li> </ul> </li> <li>• Clock synchronous serial format               <ul style="list-style-type: none"> <li>- Selectable for the MSB-first or LSB-first to the data transfer direction</li> </ul> </li> </ul>

NOTE:

1. The interrupt sources can use the only I<sup>2</sup>C bus interface interrupt vector table.



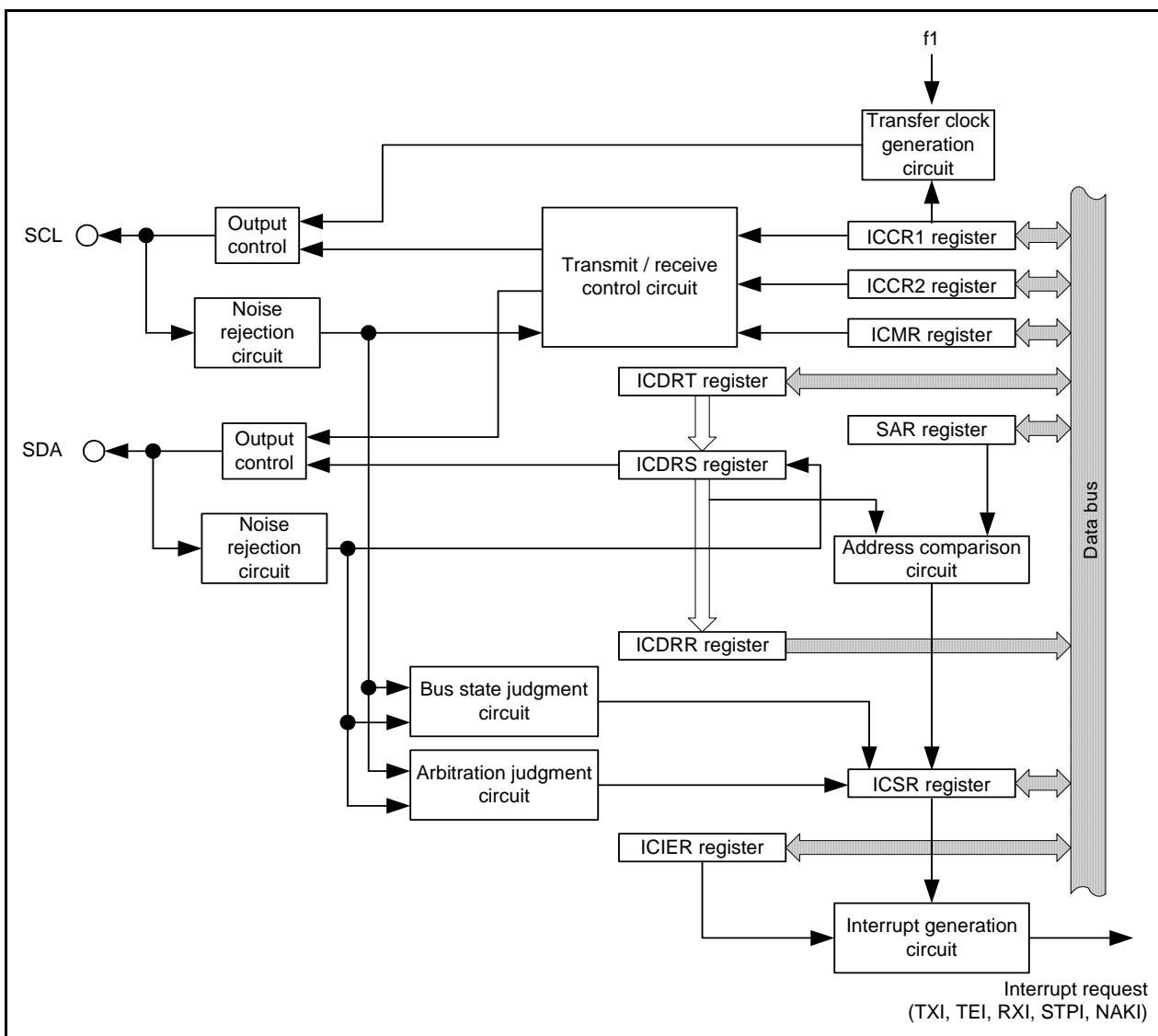


Figure 16.22 Block Diagram of I<sup>2</sup>C Bus Interface

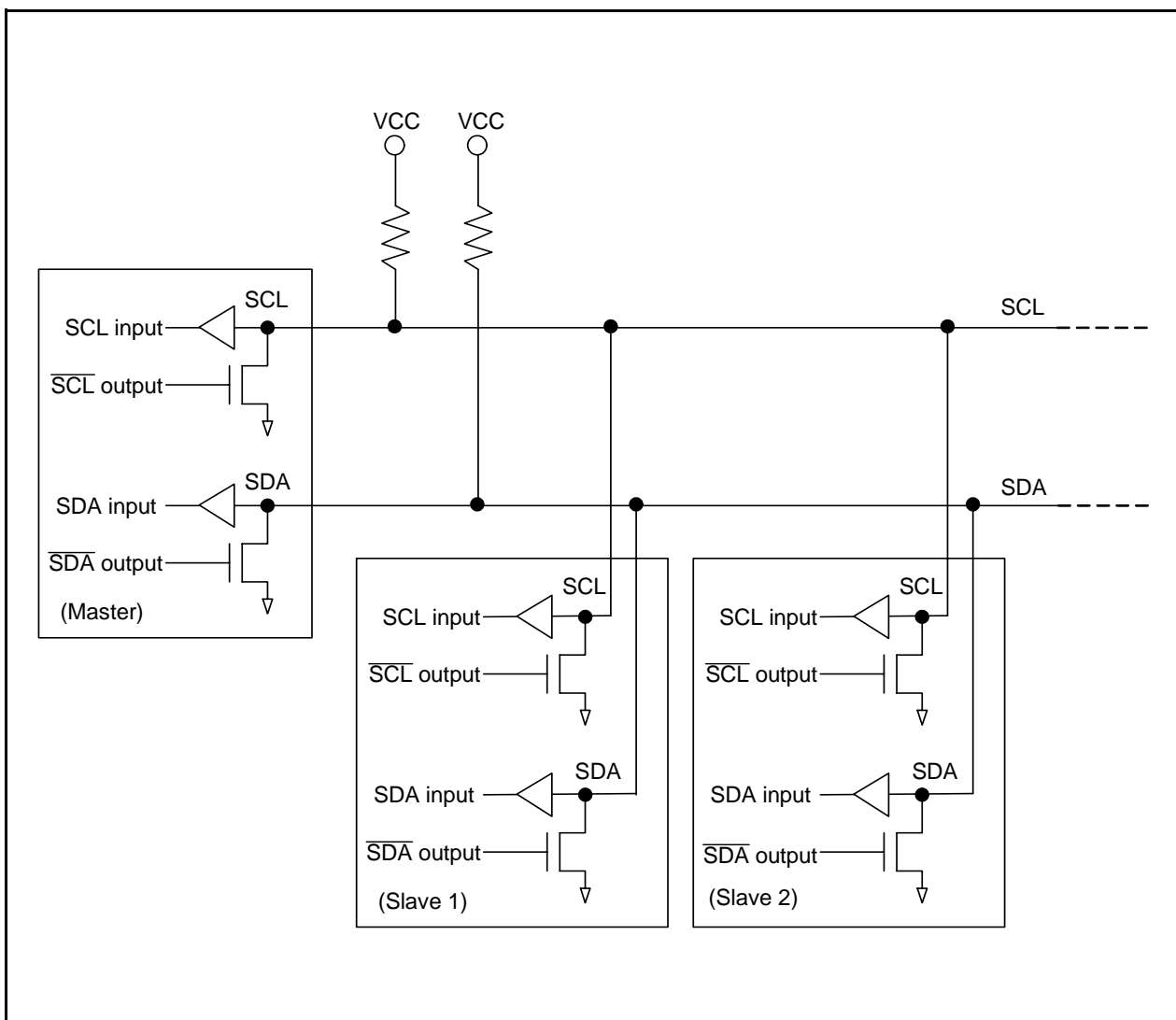


Figure 16.23 External Circuit Connection Example of Pins SCL and SDA

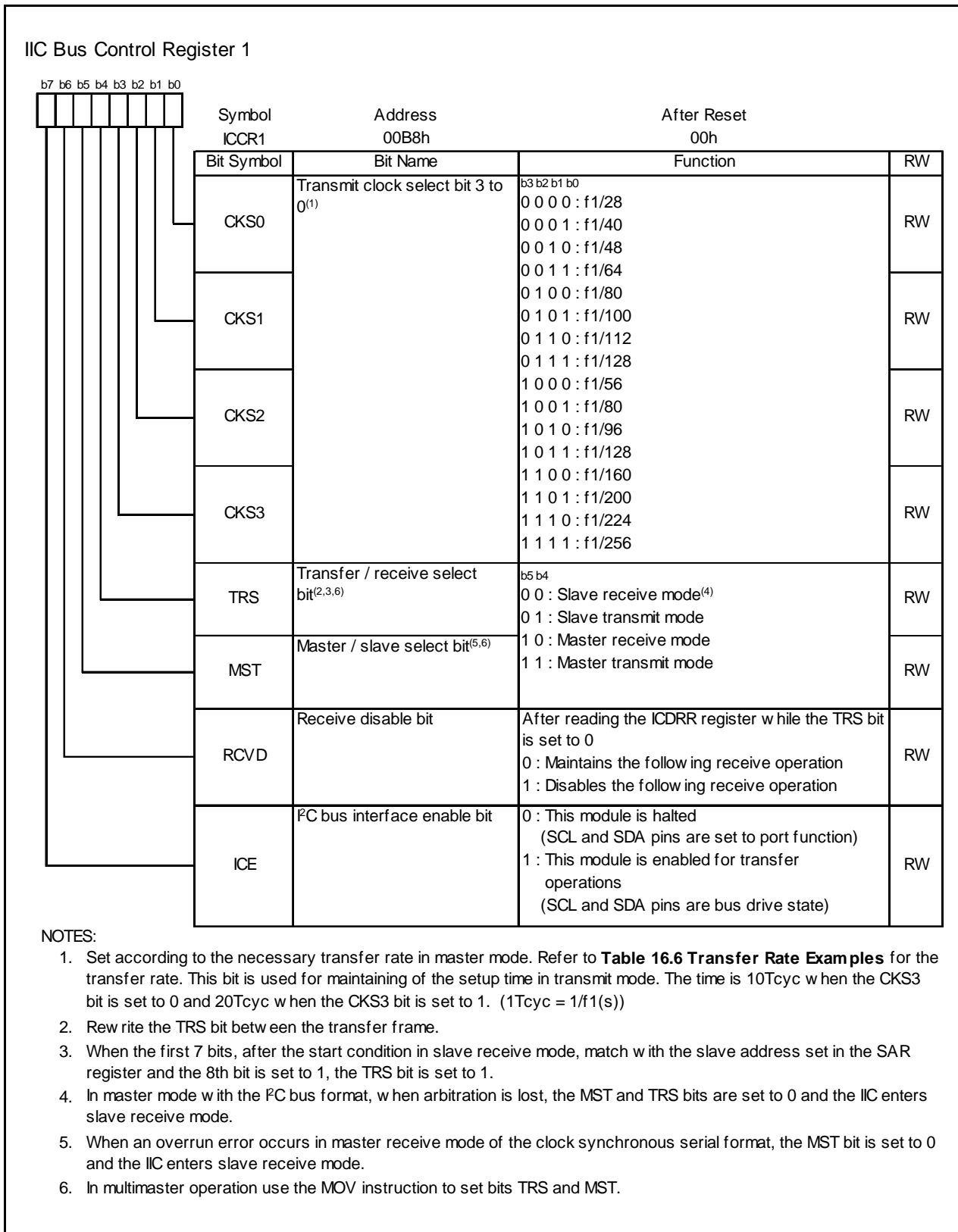
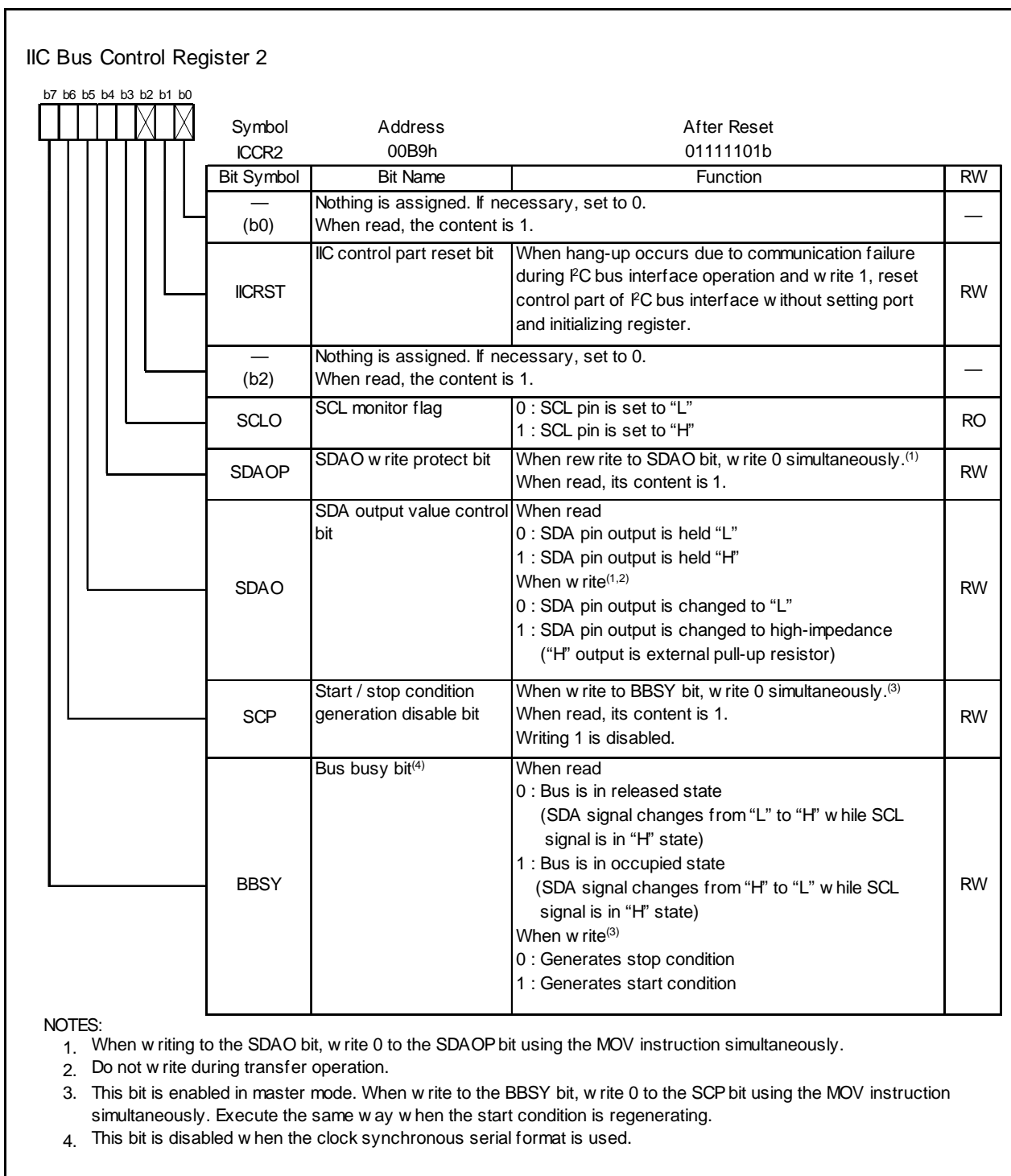


Figure 16.24 ICCR1 Register



**Figure 16.25 ICCR2 Register**

IIC Bus Mode Register		Symbol	Address	After Reset
		ICMR	00BAh	00011000b
Bit Symbol	Bit Name	Function		RW
BC0	Bit counter 2 to 0	I <sup>2</sup> C bus format (remaining transfer bit numbers when read out and data bit numbers of transfer to the next when write) <sup>(1,2)</sup>		RW
		b2 b1 b0		
		0 0 0 : 9 bits <sup>(3)</sup> 0 0 1 : 2 bits 0 1 0 : 3 bits 0 1 1 : 4 bits 1 0 0 : 5 bits		
BC1	Bit counter 1 to 0	Clock synchronous serial format (when read, read the remaining transfer bit numbers and when write, write 000b.)		RW
		b2 b1 b0		
		1 0 1 : 6 bits 1 1 0 : 7 bits 1 1 1 : 8 bits		
BC2	Bit counter 2 to 0	Clock synchronous serial format (when read, read the remaining transfer bit numbers and when write, write 000b.)		RW
		b2 b1 b0		
		0 0 0 : 8 bits 0 0 1 : 1 bit 0 1 0 : 2 bits 0 1 1 : 3 bits 1 0 0 : 4 bits 1 0 1 : 5 bits 1 1 0 : 6 bits 1 1 1 : 7 bits		
BCWP	BC write protect bit	When write to the BC0 to BC2 bits, write 0 simultaneously. <sup>(2,4)</sup> When read, its content is 1.		RW
— (b4)	Nothing is assigned. If necessary, set to 0. When read, the content is 1.			—
— (b5)	Reserved bit	Set to 0		RW
WAIT	Wait insertion bit <sup>(6)</sup>	0 : No wait (Transfer data and acknowledge bit consecutively) 1 : Wait (After the falling of the clock for the final data bit, "L" period is extended for two transfer clocks)		RW
MLS	MSB-first / LSB-first select	0 : Data transfer by MSB-first <sup>(6)</sup> 1 : Data transfer by LSB-first		RW

NOTES:

1. Rewrite between transfer frames. When write values other than 000b, write when the SCL signal is "L".
2. When write to the BC0 to BC2 bits, write 0 to the BCWP bit using the MOV instruction.
3. After data including the acknowledge bit is transferred, bits b2 to b0 are automatically set to 000b. When the start condition is detected, these bits are automatically set to 000b.
4. Do not rewrite when the clock synchronous serial format is used.
5. The setting value is enabled in master mode of the I<sup>2</sup>C bus format. It is disabled in slave mode of the I<sup>2</sup>C bus format or when the clock synchronous serial format is used.
6. Set to 0 when the I<sup>2</sup>C bus format is used.

Figure 16.26 ICMR Register

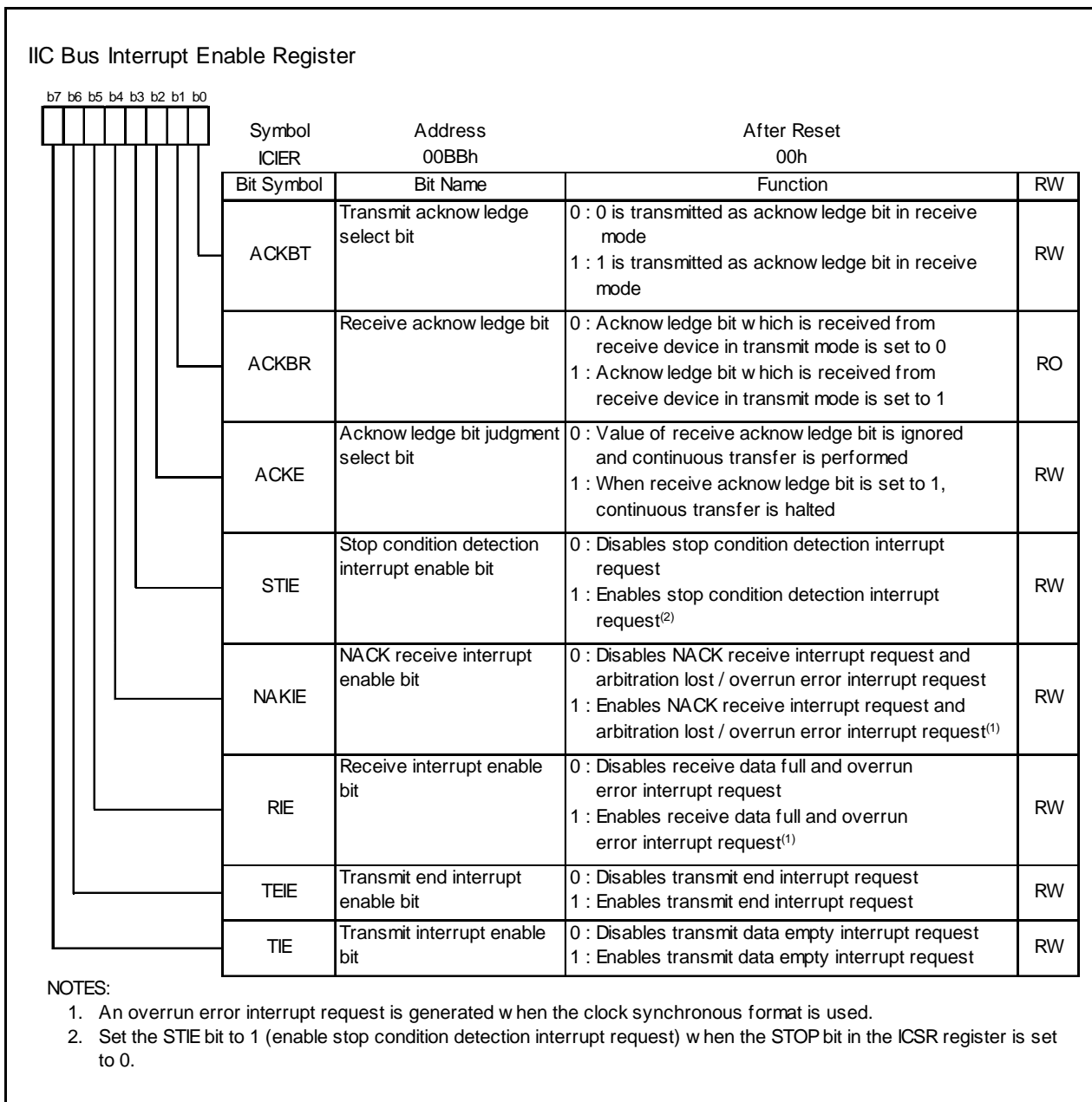


Figure 16.27 ICIER Register

IIC Bus Status Register <sup>(7)</sup>			
b7 b6 b5 b4 b3 b2 b1 b0			
Symbol	Address	After Reset	
ICSR	00BCh	0000X000b	
Bit Symbol	Bit Name	Function	RW
ADZ	General call address recognition flag <sup>(1,2)</sup>	When detecting the general call address, this flag is set to 1.	RW
AAS	Slave address recognition flag <sup>(1)</sup>	This flag is set to 1 when the first frame following start condition matches the SVA0 to SVA6 bits in the SAR register in slave receive mode. (Detect the slave address and generate call address)	RW
AL	Arbitration lost flag / overrun error flag <sup>(1)</sup>	When the I <sup>2</sup> C bus format is used, this flag indicates that arbitration is lost in master mode. In the following case, this flag is set to 1. <sup>(3)</sup> <ul style="list-style-type: none"> <li>When the internal SDA signal and SDA pin level do not match at the rise of the SCL signal in master transmit mode</li> <li>When the start condition is detected and the SDA pin is held "H" in master transmit / receive mode</li> </ul> <p>This flag indicates that an overrun error occurs when the clock synchronous format is used. In the following case, this flag is set to 1.</p> <ul style="list-style-type: none"> <li>When the last bit of the following data is received while the RDRF bit is set to 1</li> </ul>	RW
STOP	Stop condition detection flag <sup>(1)</sup>	When the stop condition is detected after the frame is transferred, this flag is set to 1	RW
NACKF	No acknowledge detection flag <sup>(1,4)</sup>	When no acknowledge is detected from receive device when transmit, this flag is set to 1	RW
RDRF	Receive data register full <sup>(1,5)</sup>	When receive data is transferred from ICDRS to ICDRR registers, this flag is set to 1	RW
TEND	Transmit end <sup>(1,6)</sup>	When the 9th clock of the SCL signal with the I <sup>2</sup> C bus format while the TDRE bit is set to 1, this flag is set to 1 This flag is set to 1 when the final bit of the transmit frame is transmitted with the clock synchronous format	RW
TDRE	Transmit data empty <sup>(1,6)</sup>	In the following cases, this flag is set to 1 <ul style="list-style-type: none"> <li>Data is transferred from ICRT to ICDRS registers and ICRT register is empty</li> <li>When setting the TRS bit in the ICCR1 register to 1 (transmit mode)</li> <li>When generating the start condition (including retransmit)</li> <li>When changing from slave receive mode to slave transmit mode</li> </ul>	RW

NOTES:

- Each bit is set to 0 when reading 1 before writing 0.
- This flag is enabled in slave receive mode of the I<sup>2</sup>C bus format.
- When two or more master devices attempt to occupy the bus at nearly the same time, if the I<sup>2</sup>C bus Interface monitors the SDA pin and the data which the IIC transmits is different, the AL flag is set to 1 and the bus is occupied by the other masters.
- The NACKF bit is enabled when the ACKE bit in the ICIER register is set to 1 (when the receive acknowledge bit is set to 1, transfer is halted)
- The RDRF bit is set to 0 when reading data from the ICDRR register.
- The TEND and TDRE bits are set to 0 when writing data to the ICRT register.
- When accessing the ICSR register continuously, insert one or more NOP instructions between the instructions to access it.

Figure 16.28 ICSR Register

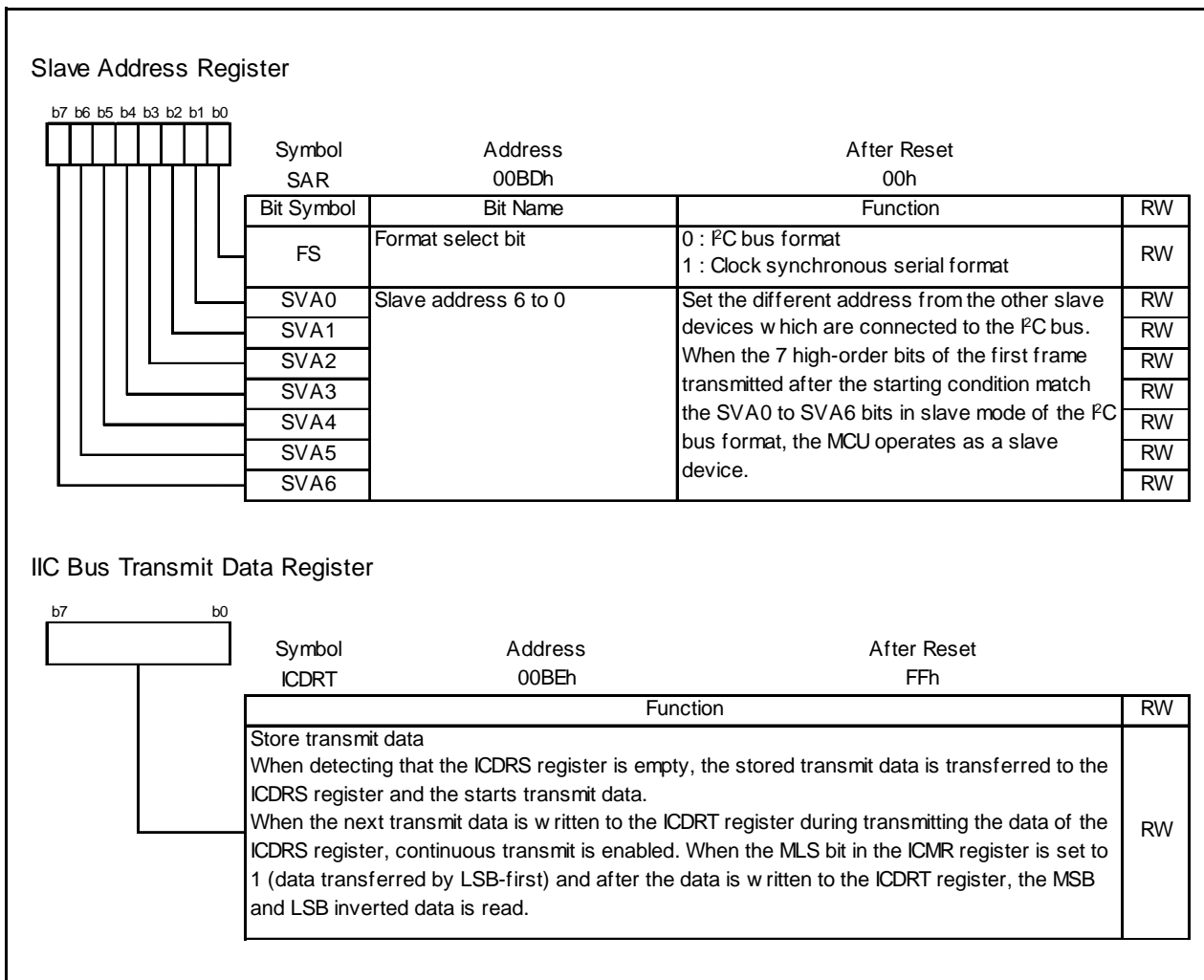


Figure 16.29 Registers SAR and ICDRT



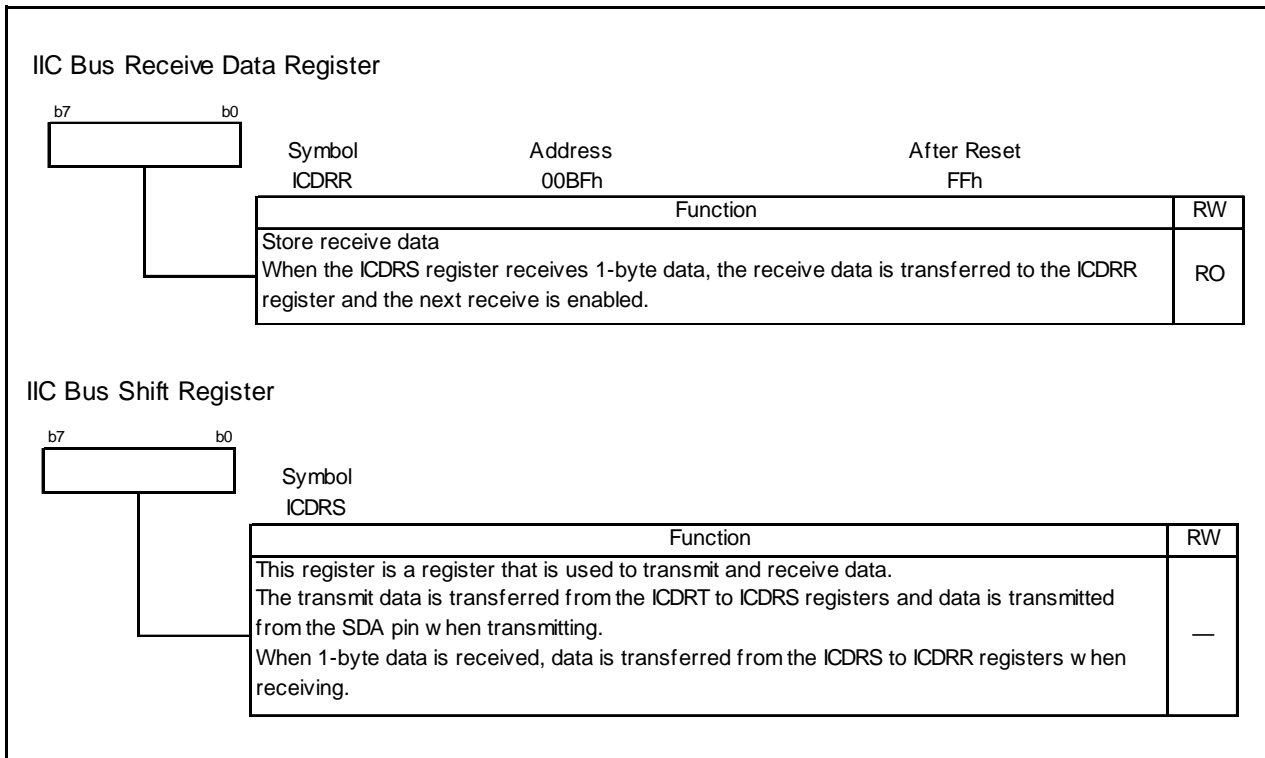


Figure 16.30 Registers ICDRR and ICDRS

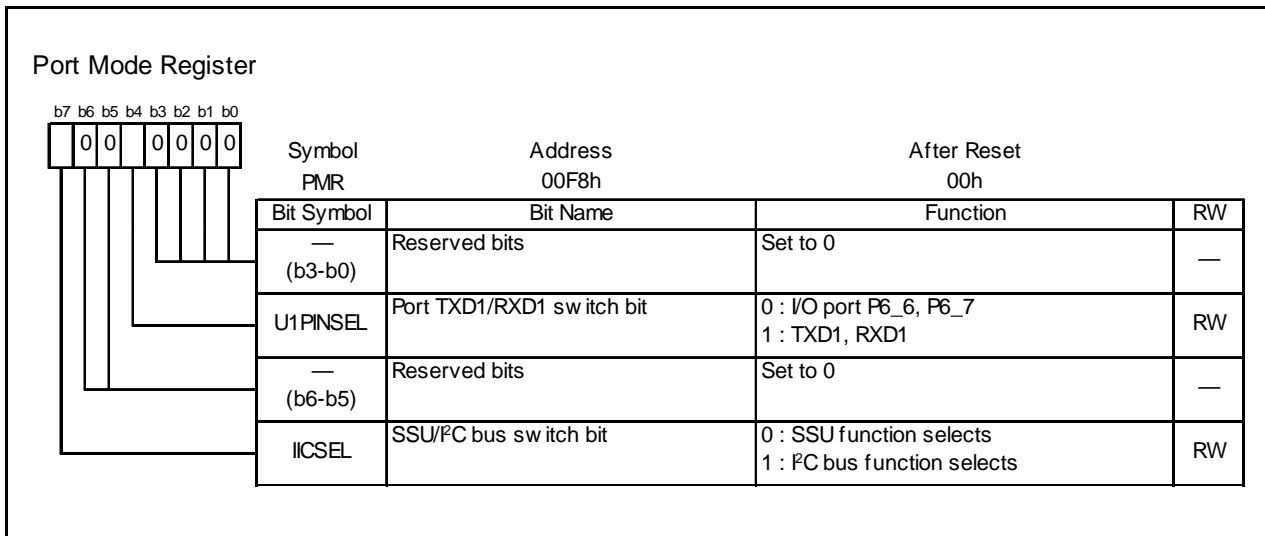


Figure 16.31 PMR Register

### 16.3.1 Transfer Clock

When the MST bit in the ICCR1 register is set to 0, the transfer clock is the external clock input from the SCL pin. When the MST bit in the ICCR1 register is set to 1, the transfer clock is the internal clock selected by the CKS0 to CKS3 bits in the ICCR1 register and the transfer clock is output from the SCL pin.

Table 16.6 lists the Transfer Rate Examples.

**Table 16.6 Transfer Rate Examples**

ICCR1 Register				Transfer Clock	Transfer Rate				
CKS3	CKS2	CKS1	CKS0		f1 = 5 MHz	f1 = 8 MHz	f1 = 10 MHz	f1 = 16 MHz	f1 = 20 MHz
0	0	0	0	f1/28	179 kHz	286 kHz	357 kHz	571 kHz	714 kHz
			1	f1/40	125 kHz	200 kHz	250 kHz	400 kHz	500 kHz
		1	0	f1/48	104 kHz	167 kHz	208 kHz	333 kHz	417 kHz
			1	f1/64	78.1 kHz	125 kHz	156 kHz	250 kHz	313 kHz
	1	0	0	f1/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz
			1	f1/100	50.0 kHz	80.0 kHz	100 kHz	160 kHz	200 kHz
		1	0	f1/112	44.6 kHz	71.4 kHz	89.3 kHz	143 kHz	179 kHz
			1	f1/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	0	0	0	f1/56	89.3 kHz	143 kHz	179 kHz	286 kHz	357 kHz
			1	f1/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz
		1	0	f1/96	52.1 kHz	83.3 kHz	104 kHz	167 kHz	208 kHz
			1	f1/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
	1	0	0	f1/160	31.3 kHz	50.0 kHz	62.5 kHz	100 kHz	125 kHz
			1	f1/200	25.0 kHz	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz
		1	0	f1/224	22.3 kHz	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz
			1	f1/256	19.5 kHz	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz

### 16.3.2 Interrupt Requests

The interrupt request of the I<sup>2</sup>C bus interface contains 6 types when the I<sup>2</sup>C bus format is used and 4 types when the clock synchronous serial format is used.

Table 16.7 lists the Interrupt Requests of I<sup>2</sup>C Bus Interface.

Since these interrupt requests are allocated at the I<sup>2</sup>C bus interface interrupt vector table, determining the source by each bit is necessary.

**Table 16.7 Interrupt Requests of I<sup>2</sup>C Bus Interface**

Interrupt Request		Generation Condition	Format	
			I <sup>2</sup> C bus	Clock Synchronous Serial
Transmit Data Empty	TXI	TIE = 1 and TDRE = 1	Enabled	Enabled
Transmit Ends	TEI	TEIE = 1 and TEND = 1	Enabled	Enabled
Receive Data Full	RXI	RIE = 1 and RDRF = 1	Enabled	Enabled
Stop Condition Detection	STPI	STIE = 1 and STOP = 1	Enabled	Disabled
NACK Detection	NAKI	NAKIE = 1 and AL = 1 (or	Enabled	Disabled
Arbitration Lost / Overrun Error		NAKIE = 1 and NACKF = 1)	Enabled	Enabled

STIE, NAKIE, RIE, TEIE, TIE: Bits in ICIER register

AL, STOP, NACKF, RDRF, TEND, TDRE: Bits in ICSR register

When the generation conditions on the Table 16.7 are met, the I<sup>2</sup>C bus interface interrupt request is generated. Set the interrupt generation conditions to 0 by the I<sup>2</sup>C bus interface interrupt routine. However, the TDRE and TEND bits are automatically set to 0 by writing transmit data to the ICDRT register and the RDRF bit is automatically set to 0 by reading the ICDRR register. When writing transmit data to the ICDRT register, the TDRE bit is set to 0. When data is transferred from the ICDRT to ICDRS registers, the TDRE bit is set to 1 and when further setting the TDRE bit to 0, extra 1 byte may be transmitted.

Also, set the STIE bit to 1 (enable stop condition detection interrupt request) when the STOP bit is set to 0.

### 16.3.3 I<sup>2</sup>C Bus Interface Mode

#### 16.3.3.1 I<sup>2</sup>C Bus Format

Setting the FS bit in the SAR register to 0 communicates in I<sup>2</sup>C bus format.

Figure 16.32 shows the I<sup>2</sup>C Bus Format and Bus Timing. The 1st frame following start condition consists of 8 bits.

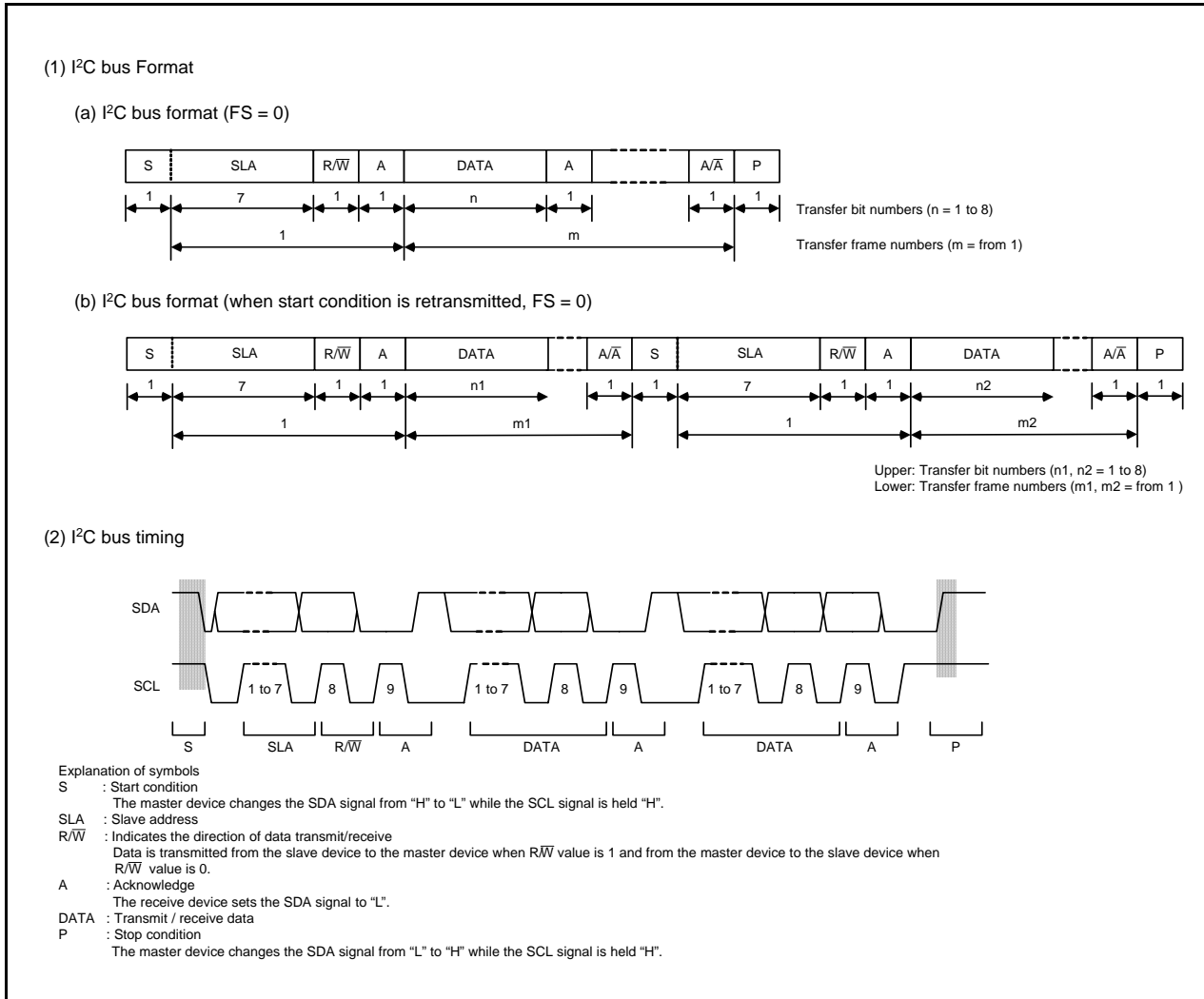


Figure 16.32 I<sup>2</sup>C Bus Format and Bus Timing

### 16.3.3.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and data, and the slave device returns an acknowledge signal.

Figure 16.33 and Figure 16.34 show the Operation Timing in Master Transmit Mode (I<sup>2</sup>C Bus Interface Mode).

The transmit procedure and operation in master transmit mode are shown below.

- (1) Set the STOP bit in the ICSR register to 0 to reset it. And then set the ICE bit in the ICCR1 register to 1 (transfer operation enabled). Set the WAIT and MLS bits in the ICMR register and set the CKS0 to CKS3 bits in the ICCR1 register (initial setting).
- (2) Read the BBSY bit in the ICCR2 register to confirm that the bus is free. Set the TRS and MST bits in the ICCR1 register to master transmit mode. The start condition is generated by writing 1 to the BBSY bit and 0 to the SCP bit by the MOV instruction.
- (3) After confirming that the TDRE bit in the ICSR register is set to 1 (data is transferred from the ICDRT to ICDRS registers), write transmit data to the ICDRT register (data in which a slave address and R/W are shown at the 1st byte). At this time, the TDRE bit is automatically set to 0 and data is transferred from the ICDRT to ICDRS registers, the TDRE bit is set to 1 again.
- (4) When the transmit of 1-byte data is completed while the TDRE bit is set to 1, the TEND bit in the ICSR register is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in the ICIER register, and confirm that the slave is selected. Write the 2nd-byte data to the ICDRT register. Since the slave device is not acknowledged when the ACKBR bit is set to 1, generate the stop condition. The stop condition is generated by the writing 0 to the BBSY bit and 0 to the SCP bit by the MOV instruction. The SCL signal is held “L” until data is available and the stop condition is generated.
- (5) Write the transmit data after the 2nd byte to the ICDRT register every time the TDRE bit is set to 1.
- (6) When writing the number of bytes to be transmitted to the ICDRT register, wait until the TEND bit is set to 1 while the TDRE bit is set to 1. Or wait for NACK (the NACKF bit in the ICSR register is set to 1) from the receive device while the ACKE bit in the ICIER register is set to 1 (when the receive acknowledge bit is set to 1, transfer is halted). And generate the stop condition before setting the TEND and NACKF bits to 0.
- (7) When the STOP bit in the ICSR register is set to 1, return to slave receive mode.

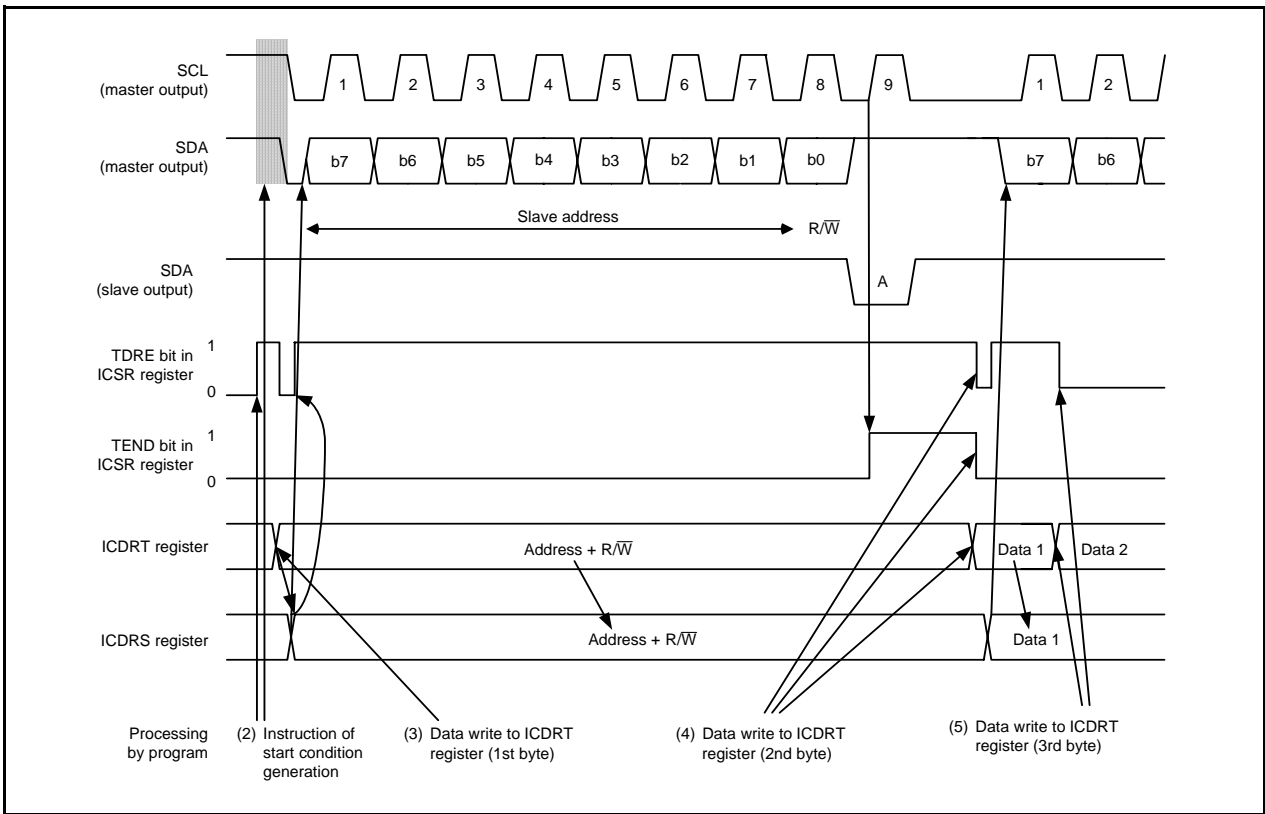


Figure 16.33 Operating Timing in Master Transmit Mode (I<sup>2</sup>C Bus Interface Mode) (1)

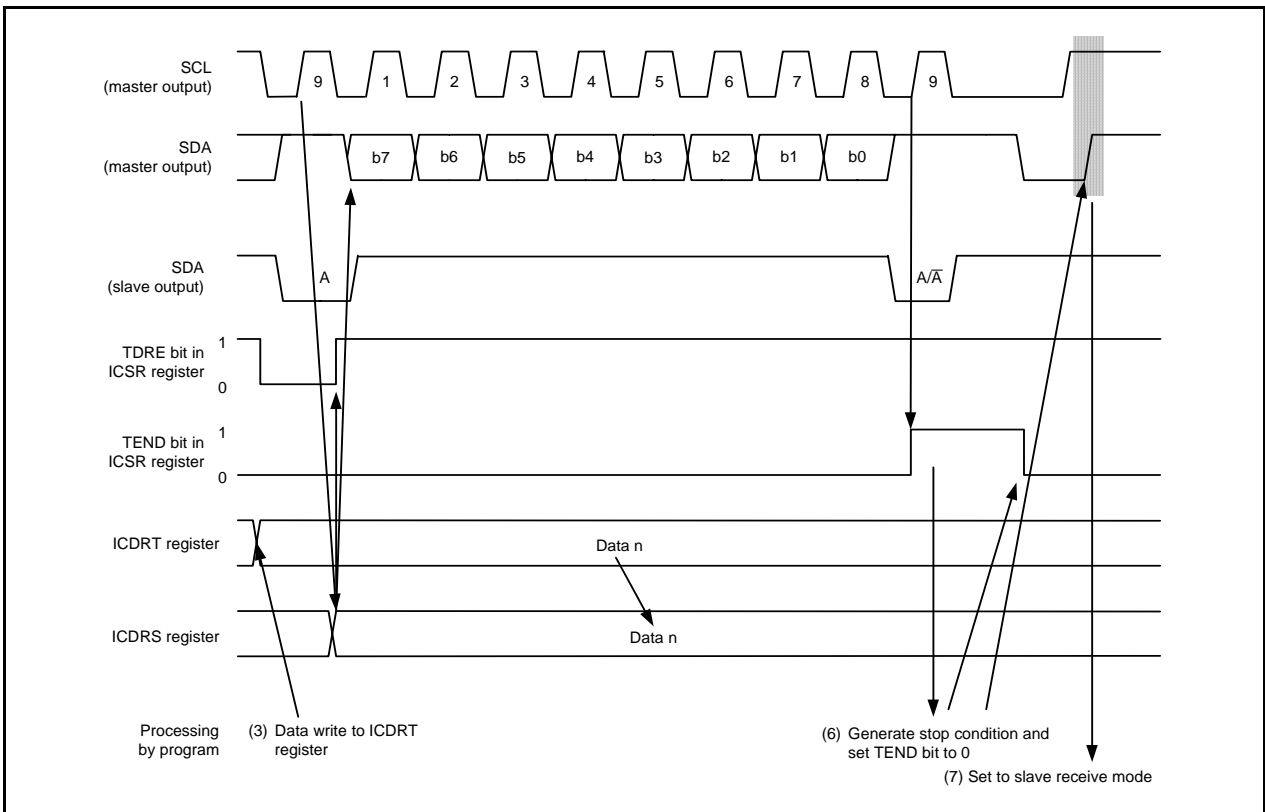


Figure 16.34 Operating Timing in Master Transmit Mode (I<sup>2</sup>C Bus Interface Mode) (2)

### 16.3.3.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal.

Figure 16.35 and Figure 16.36 show the Operation Timing in Master Receive Mode (I<sup>2</sup>C Bus Interface Mode).

The receive procedure and operation in master receive mode are shown below.

- (1) After setting the TEND bit in the ICSR register to 0, switch from master transmit mode to master receive mode by setting the TRS bit in the ICCR1 register. And set the TDRE bit in the ICSR register to 0.
- (2) When performing the dummy-read of the ICDRR register and starting receive, output the receive clock synchronizing with the internal clock and receive data. The master device outputs the level set by the ACKBT bit in the ICIER register to the SDA pin at the 9th clock of the receive clock.
- (3) The 1-frame data receive is completed and the RDRF bit in the ICSR register is set to 1 at the rise of the 9th clock. At this time, when reading the ICDRR register, the received data can be read and the RDRF bit is set to 0 simultaneously.
- (4) The continuous receive is enabled by reading the ICDRR register every time the RDRF bit is set to 1. If the 8th clock falls after reading the ICDRR register by the other processes while the RDRF bit is set to 1, the SCL signal is fixed "L" until the ICDRR register is read.
- (5) If the following frame is the last receive frame and the RCVD bit in the ICCR1 register is set to 1 (disables the next receive operation) before reading the ICDRR register, the stop condition generation is enabled after the following receive.
- (6) When the RDRF bit is set to 1 at the rise of the 9th clock of the receive clock, generate the stop condition.
- (7) When the STOP bit in the ICSR register is set to 1, read the ICDRR register. And set the RCVD bit to 0 (maintain the following receive operation).
- (8) Return to slave receive mode.

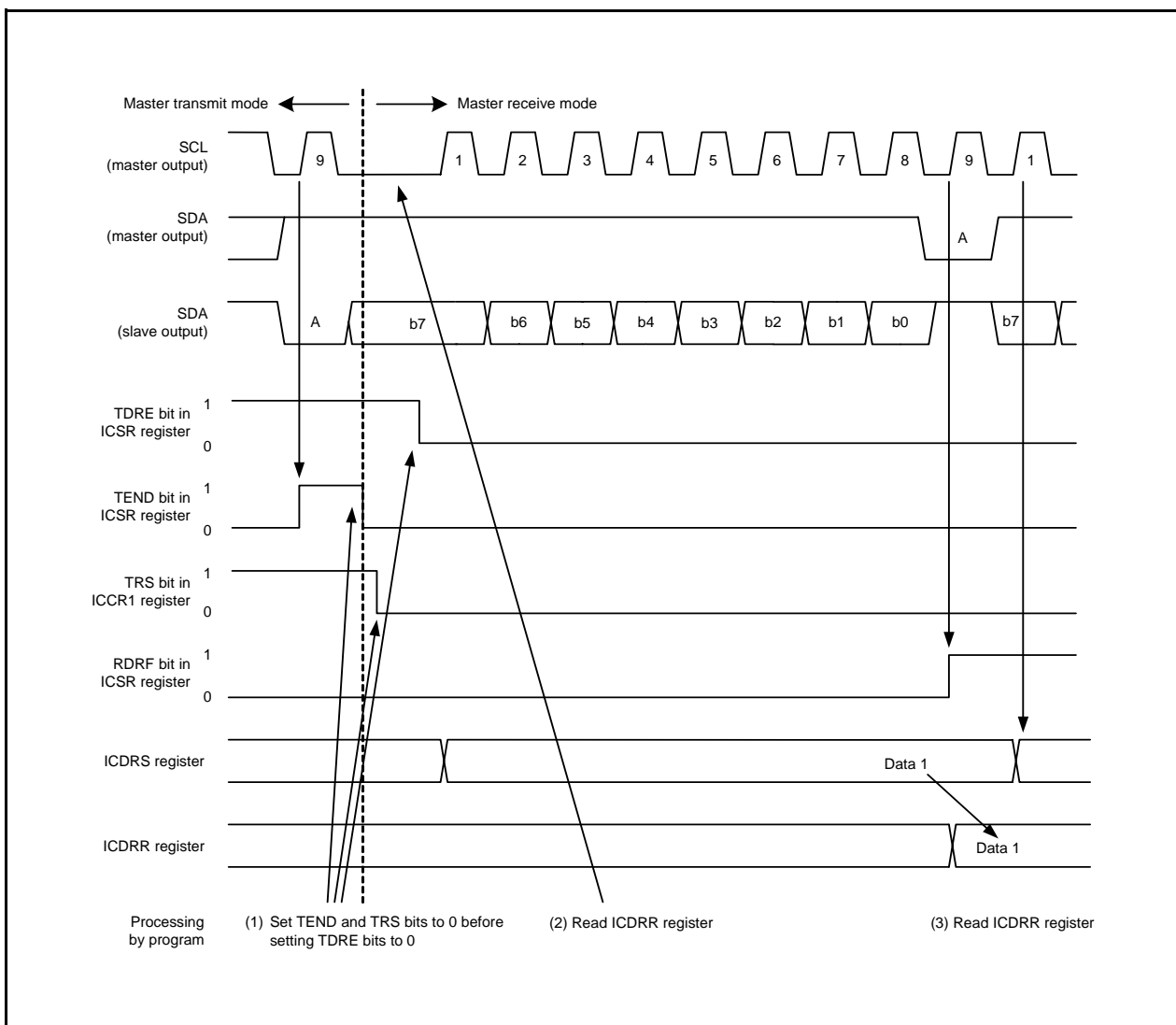


Figure 16.35 Operating Timing in Master Receive Mode (I<sup>2</sup>C Bus Interface Mode) (1)



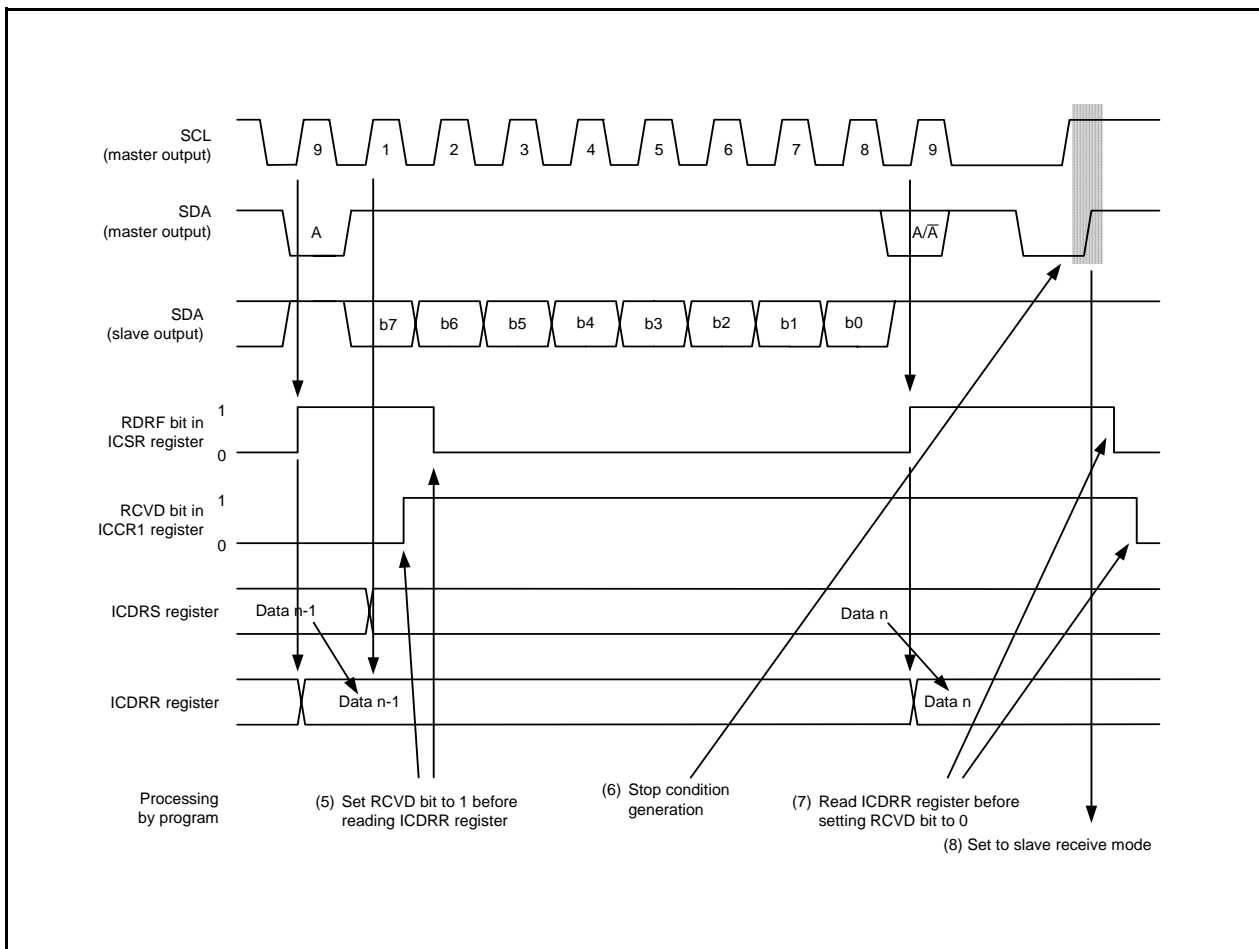


Figure 16.36 Operating Timing in Master Receive Mode (I<sup>2</sup>C Bus Interface Mode) (2)

### 16.3.3.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data while the master device outputs the receive clock and returns an acknowledge signal.

Figure 16.37 and Figure 16.38 show the Operation Timing in Slave Transmit Mode (I<sup>2</sup>C Bus Interface Mode).

The transmit procedure and operation in slave transmit mode are shown below.

- (1) Set the ICE bit in the ICCR1 register to 1 (transfer operation enabled). Set the WAIT and MLS bits in the ICMR register and CKS0 to CKS3 bits in the ICCR1 register (initial setting). Set the TRS and MST bits in the ICCR1 register to 0 and wait until the slave address matches in slave receive mode.
- (2) When the slave address matches at the 1st frame after detecting the start condition, the slave device outputs the level set by the ACKBT bit in the ICIER register to the SDA pin at the rise of the 9th clock. At this time, if the 8-bit data ( $\overline{R/\overline{W}}$ ) is set to 1, the TRS and TDRE bit in the ICSR register are set to 1, the mode is switched to slave transmit mode automatically. When writing transmit data to the ICDRT register every time the TDRE bit is set to 1, the continuous transmit is enabled.
- (3) When the TDRE bit in the ICDRT register is set to 1 after writing the last transmit data to the ICDRT register, wait until the TEND bit in the ICSR register is set to 1 while the TDRE bit is set to 1. When the TEND bit is set to 1, set the TEND bit to 0.
- (4) The SCL signal is released by setting the TRS bit to 0 and performing the dummy-read of the ICDRR register for the end process.
- (5) Set the TDRE bit to 0.

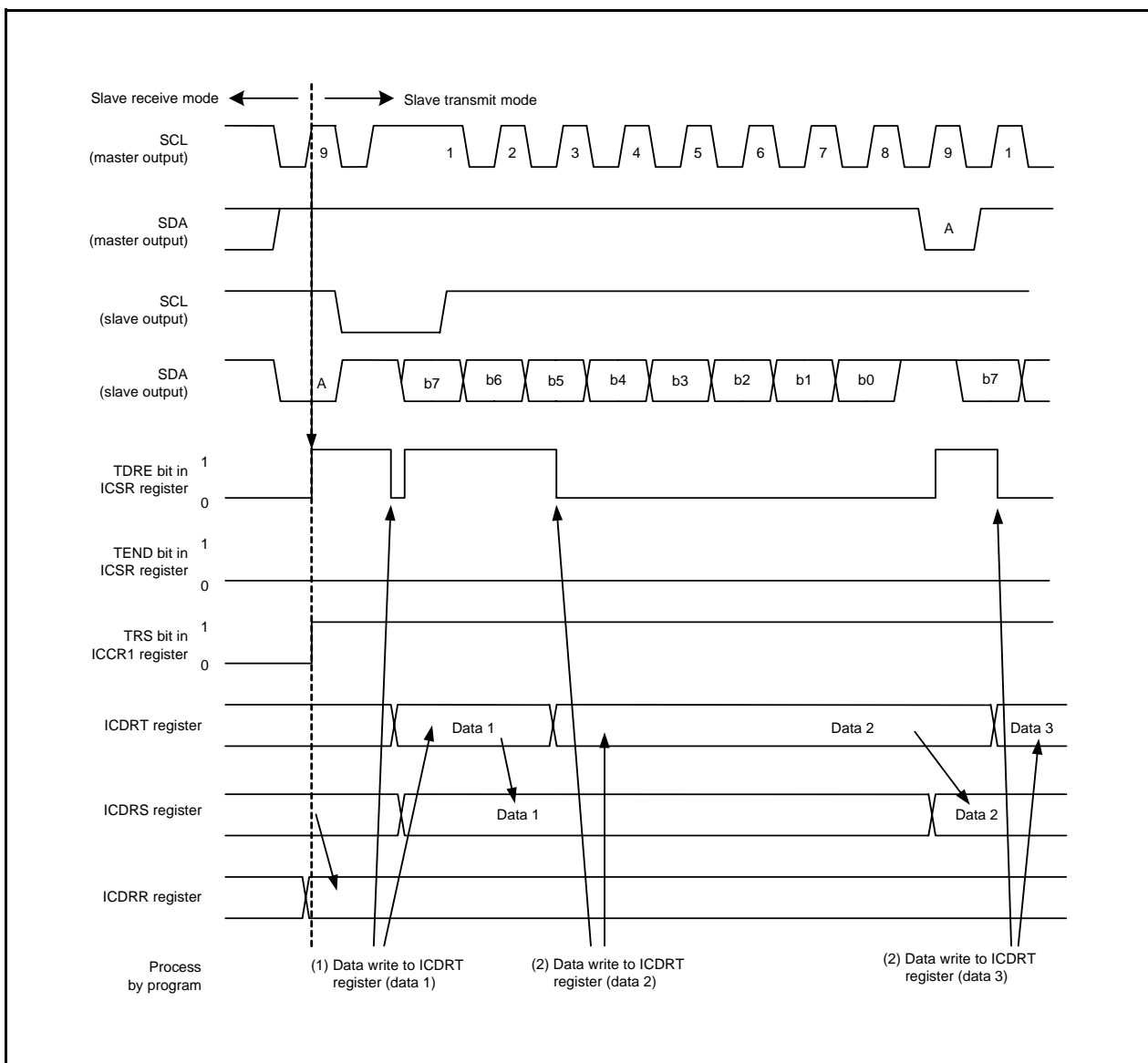


Figure 16.37 Operating Timing in Slave Transmit Mode (I<sup>2</sup>C Bus Interface Mode) (1)

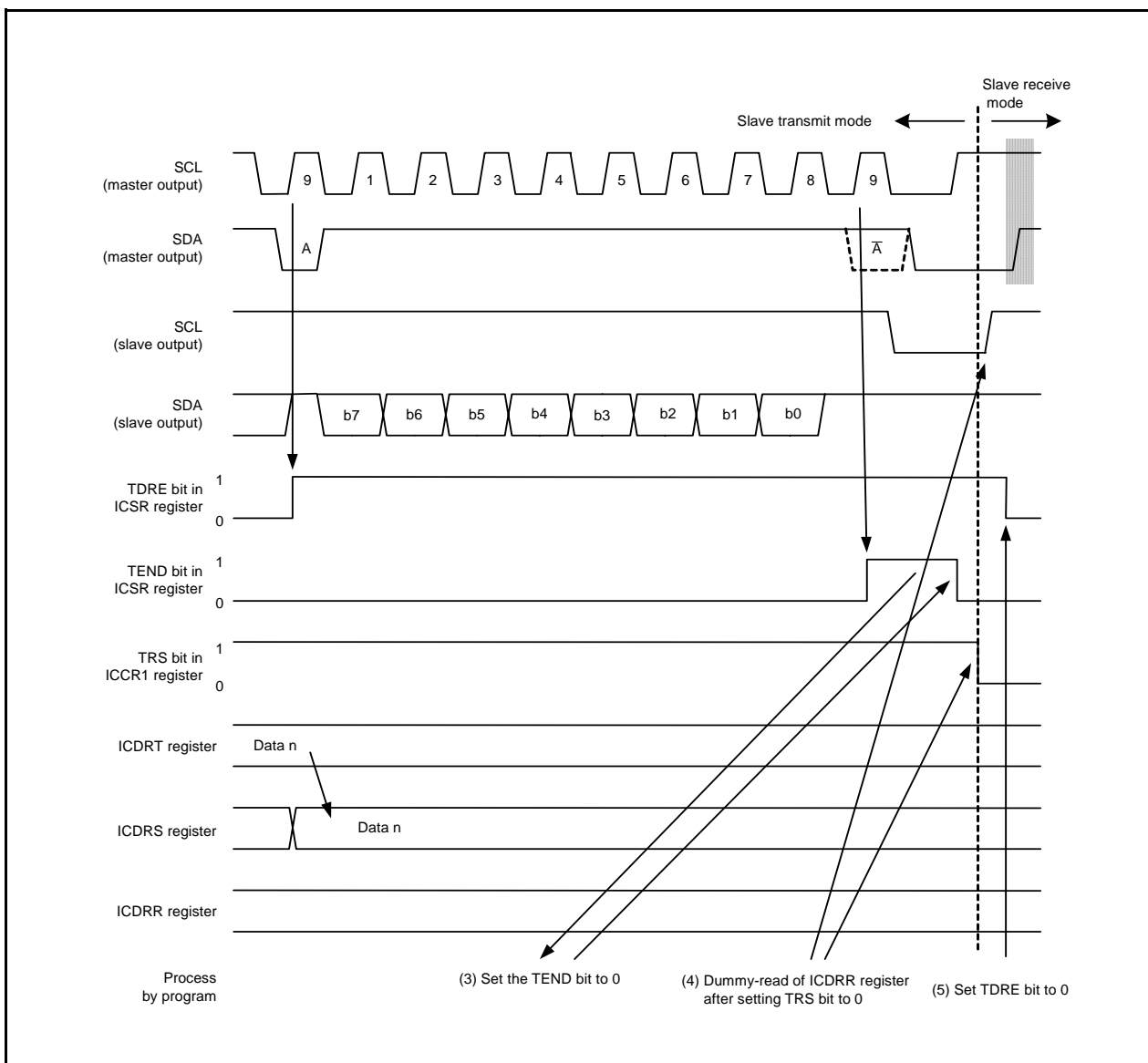


Figure 16.38 Operating Timing in Slave Transmit Mode (I<sup>2</sup>C Bus Interface Mode) (2)

### 16.3.3.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and data, and the slave device returns an acknowledge signal.

Figure 16.39 and Figure 16.40 show the Operation Timing in Slave Receive Mode (I<sup>2</sup>C Bus Interface Mode).

The receive procedure and operation in slave receive mode are shown below.

- (1) Set the ICE bit in the ICCR1 register to 1 (transfer operation enabled). Set the WAIT and MLS bits in the ICMR register and CKS0 to CKS3 bits in the ICCR1 register (initial setting). Set the TRS and MST bits in the ICCR1 register to 0 and wait until the slave address matches in slave receive mode.
- (2) When the slave address matches at the 1st frame after detecting the start condition, the slave device outputs the level set in the ACKBT bit in the ICIER register to the SDA pin at the rise of the 9th clock. Since the RDRF bit in the ICSR register is set to 1 simultaneously, perform the dummy-read (the read data is unnecessary because of showing slave address and  $\overline{R/\overline{W}}$ ).
- (3) Read the ICDRR register every time the RDRF bit is set to 1. If the 8th clock falls while the RDRF bit is set to 1, the SCL signal is fixed “L” until the ICDRR register is read. The setting change of the acknowledge signal which returns to master device before reading the ICDRR register reflects the following transfer frame.
- (4) Reading the last byte is performed by reading the ICDRR register as well.

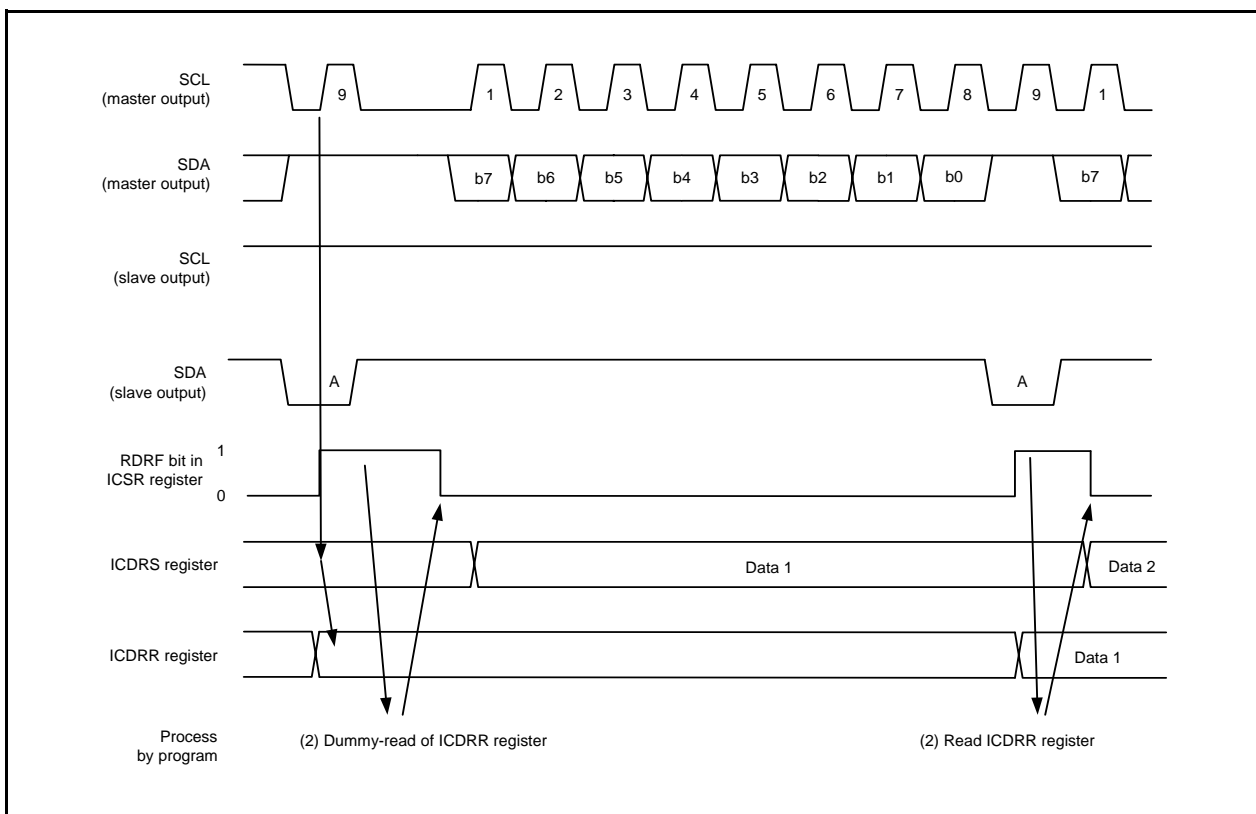


Figure 16.39 Operating Timing in Slave Receive Mode (I<sup>2</sup>C Bus Interface Mode) (1)

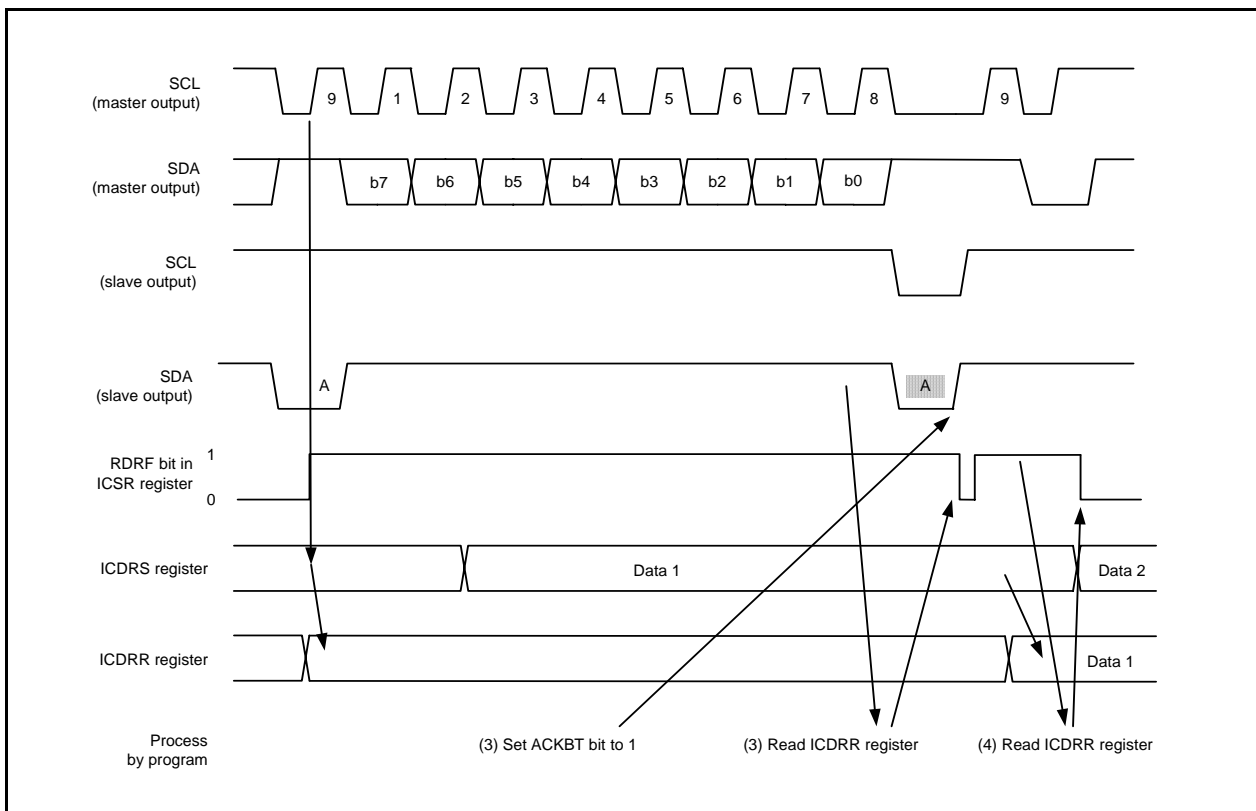


Figure 16.40 Operating Timing in Slave Receive Mode (I<sup>2</sup>C Bus Interface Mode) (2)

### 16.3.4 Clock Synchronous Serial Mode

#### 16.3.4.1 Clock Synchronous Serial Format

When setting the FS bit in the SAR register to 1, the clock synchronous serial format is used to communicate. Figure 16.41 shows the Transfer Format of Clock Synchronous Serial Format.

When the MST bit in the ICCR1 register is set to 1, the transfer clock is output from the SCL pin and when the MST bit is set to 0, the external clock is input.

The transfer data is output between the fall and the following fall of the SCL clock, and data is determined by the rise of the SCL clock. The MSB-first or LSB-first can be selected for the order of the data transfer by setting the MLS bit in the ICMR register. The SDA output level can be changed by the SDAO bit in the ICCR2 register during the transfer standby.

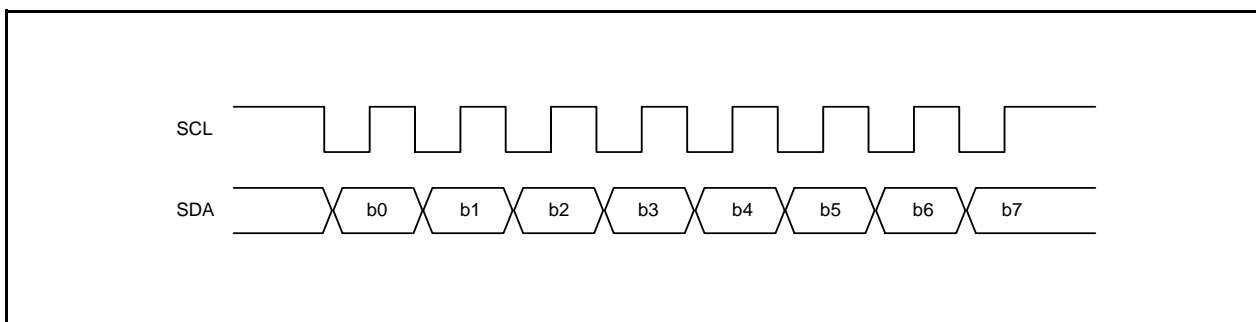


Figure 16.41 Transfer Format of Clock Synchronous Serial Format

### 16.3.4.2 Transmit Operation

In transmit mode, transmit data is output from the SDA pin synchronizing with the fall of the transfer clock. The transfer clock is output when the MST bit in the ICCR1 register is set to 1 and input when the MST bit is set to 0.

Figure 16.42 shows the Operating Timing in Transmit Mode (Clock Synchronous Serial Mode).

The transmit procedure and operation in transmit mode are shown below.

- (1) Set the ICE bit in the ICCR1 register to 1 (transfer operation enabled). Set the CKS0 to CKS3 bits in the ICCR1 register and set the MST bit (initial setting).
- (2) The TDRE bit in the ICSR register is set to 1 by selecting transmit mode after setting the TRS bit in the ICCR1 register to 1.
- (3) Data is transferred from the ICDRT to ICDRS registers and the TDRE bit is automatically set to 1 by writing transmit data to the ICDRT register after confirming that the TDRE bit is set to 1. When writing data to the ICDRT register every time the TDRE bit is set to 1, the continuous transmit is enabled. When switching from transmit to receive modes, set the TRS bit to 0 while the TDRE bit is set to 1.

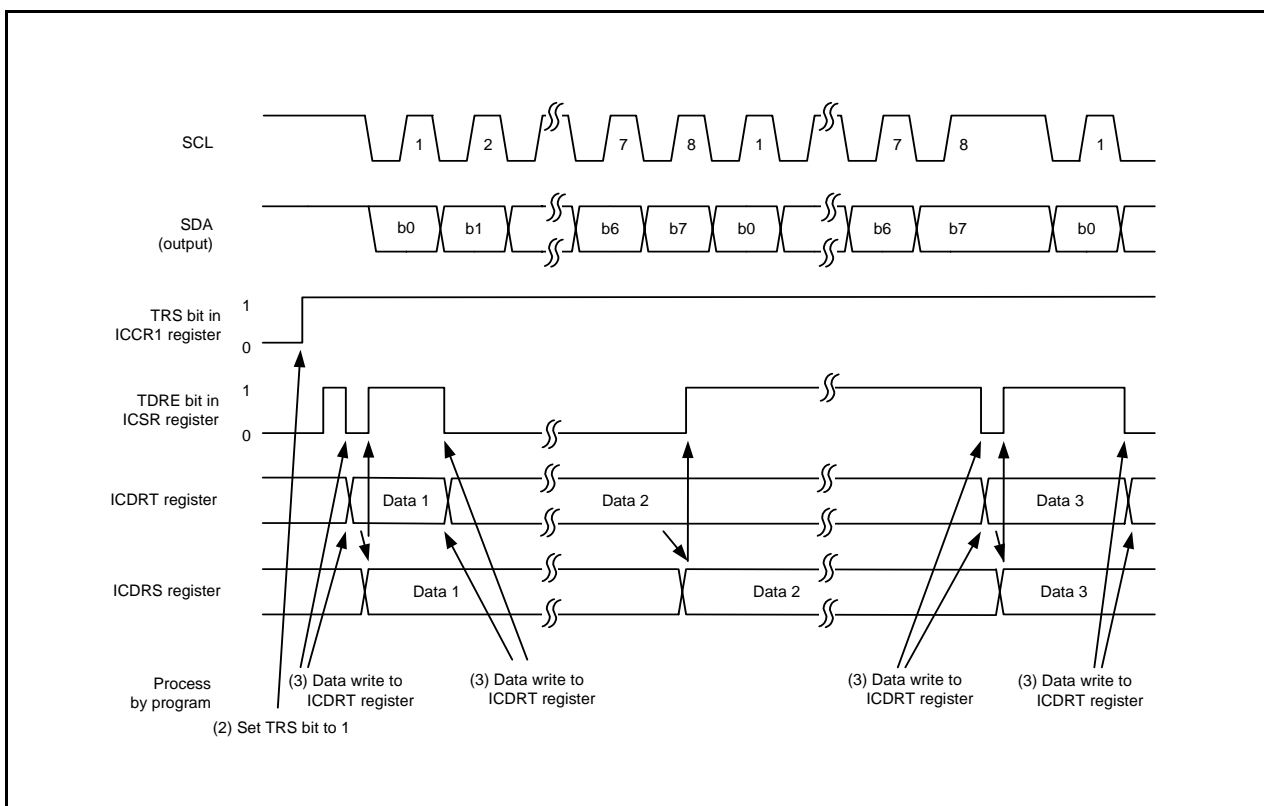


Figure 16.42 Operating Timing in Transmit Mode (Clock Synchronous Serial Mode)



### 16.3.4.3 Receive Operation

In receive mode, data is latched at the rise of the transfer clock. The transfer clock is output when the MST bit in the ICCR1 register is set to 1 and input when the MST bit is set to 0.

Figure 16.43 shows the Operating Timing in Receive Mode (Clock Synchronous Serial Mode).

The receive procedure and operation in receive mode are shown below.

- (1) Set the ICE bit in the ICCR1 register to 1 (transfer operation enabled). Set the CKS0 to CKS3 bits in the ICCR1 register and set the MST bit (initial setting).
- (2) The output of the receive clock starts by setting the MST bit to 1 when the transfer clock is output.
- (3) Data is transferred from the ICDRS to ICDRR registers and the RDRF bit in the ICSR register is set to 1, when the receive is completed. Since the following-byte data is enabled to receive when the MST bit is set to 1, the continuous clock is output. The continuous receive is enabled by reading the ICDRR register every time the RDRF bit is set to 1. An overrun is detected at the rise of the 8th clock while the RDRF bit is set to 1, the AL bit in the ICSR register is set to 1. At this time, the former receive data is retained in the ICDRR register.
- (4) When the MST bit is set to 1, set the RCVD bit in the ICCR1 register to 1 (disables the following receive operation) and read the ICDRR register. The SCL signal is fixed “H” after the receive of the following-byte data is completed.

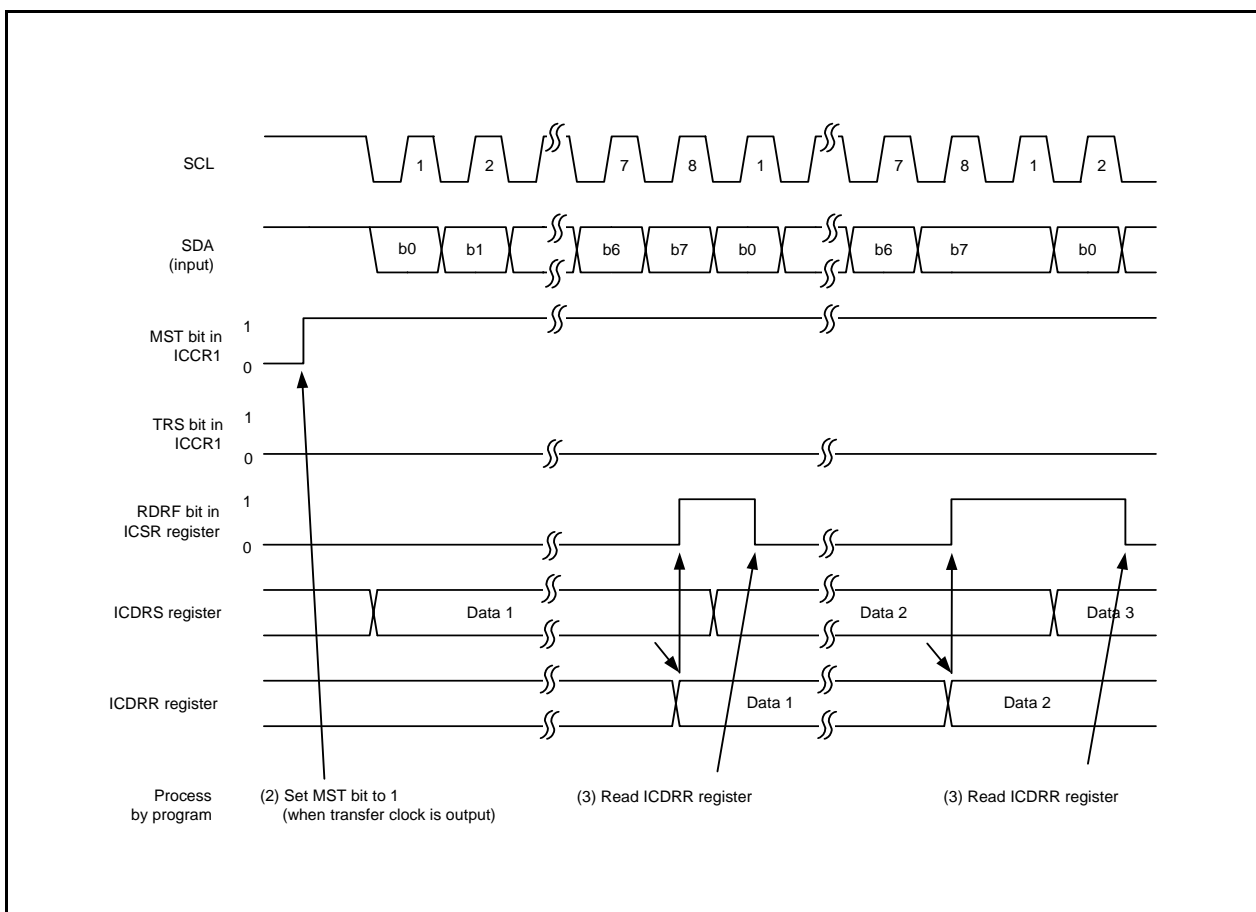


Figure 16.43 Operating Timing in Receive Mode (Clock Synchronous Serial Mode)

### 16.3.5 Noise Canceller

The state of the SCL and SDA pins are routed through the noise rejection circuit before being latched internally. Figure 16.44 shows the Block Diagram of Noise Canceller.

The noise rejection circuit consists of two cascaded latch and match detector circuits. When the SCL pin input signal (or SDA pin input signal) is sampled on  $f_1$  and 2 latch outputs match, the level is passed forward to the next circuit. When they do not match, the former value is retained.

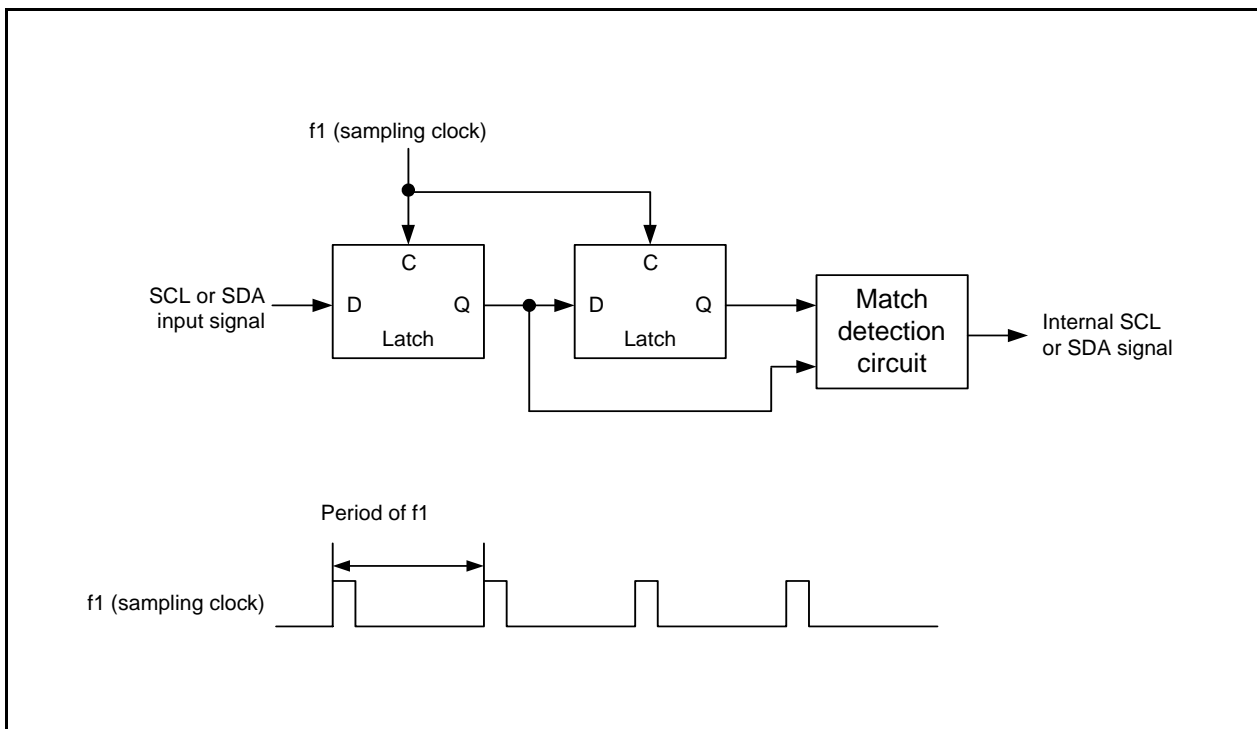


Figure 16.44 Block Diagram of Noise Canceller

### 16.3.6 Bit Synchronization Circuit

When setting the I<sup>2</sup>C bus interface in master mode.

- When the SCL signal is driven to “L” by the slave device.
- Since the “H” period may become shorter while the SCL signal is driven to “L” by the slave device and the rising speed of the SCL signal is lowered by the load (load capacity and pull-up resistor) of the SCL line, the SCL signal is monitored and the communication synchronizes per bit.

Figure 16.45 shows the Timing of Bit Synchronous Circuit and Table 16.8 lists the Time between Changing SCL Signal from “L” Output to High-Impedance and Monitoring of SCL Signal.

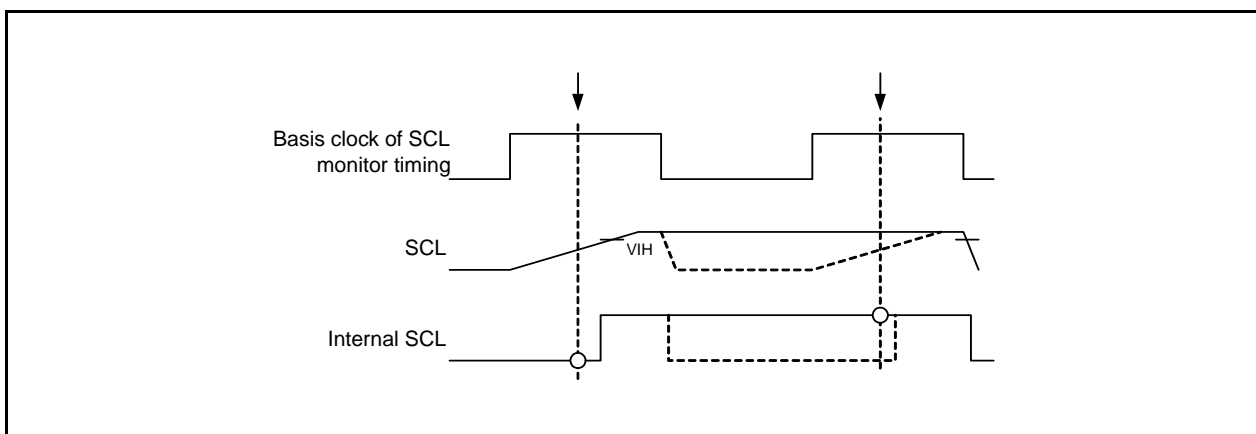


Figure 16.45 Timing of Bit Synchronous Circuit

Table 16.8 Time between Changing SCL Signal from “L” Output to High-Impedance and Monitoring of SCL Signal

ICCR1 Register		Time for Monitoring SCL
CKS3	CKS2	
0	0	7.5 Tcyc
	1	19.5 Tcyc
1	0	17.5 Tcyc
	1	41.5 Tcyc

1 Tcyc = 1/f1(s)

### 16.3.7 Examples of Register Setting

Figure 16.46 to Figure 16.49 show the Examples of Register Setting When Using I<sup>2</sup>C Bus Interface.

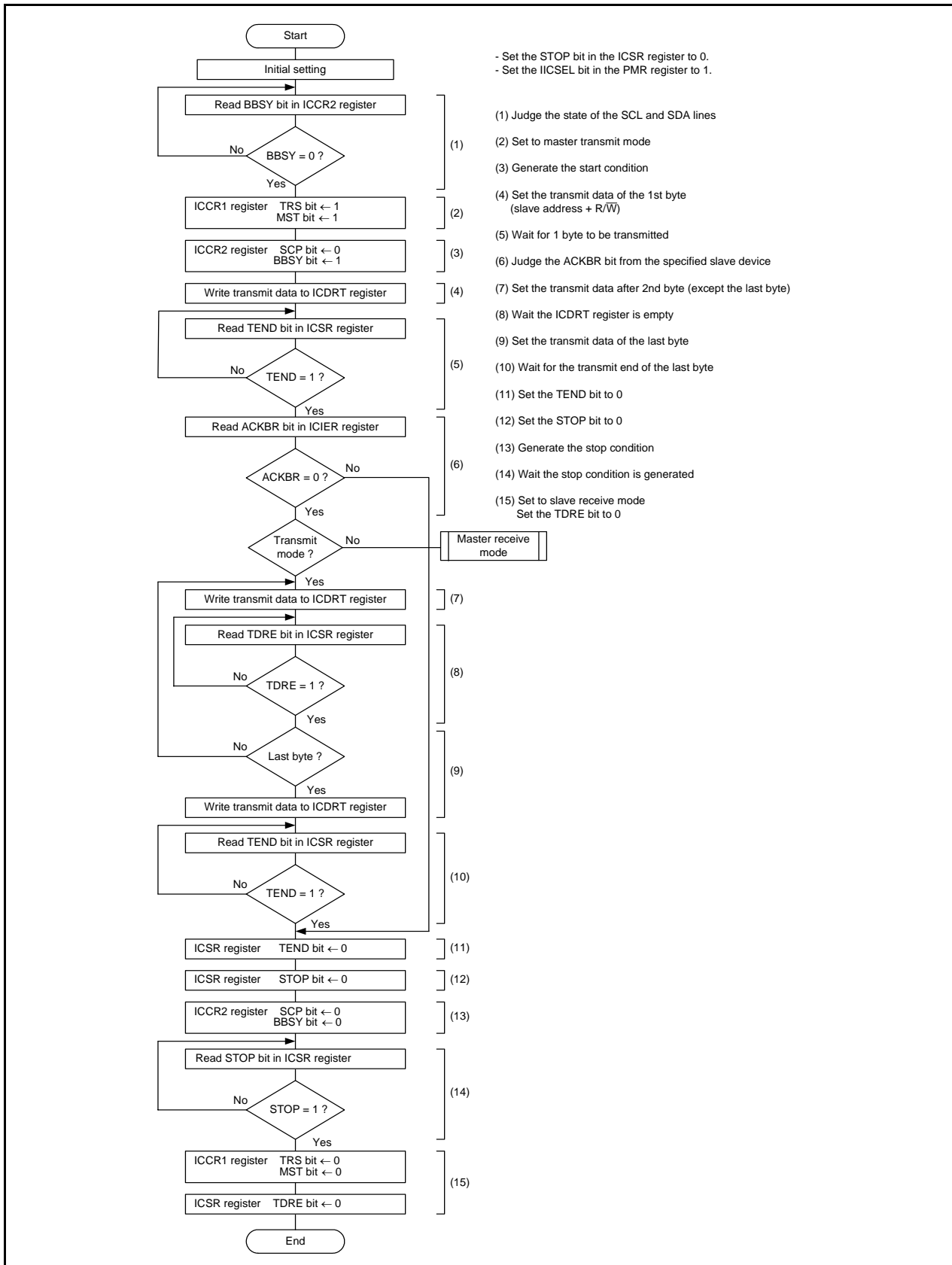


Figure 16.46 Example of Register Setting in Master Transmit Mode (I<sup>2</sup>C Bus Interface Mode)

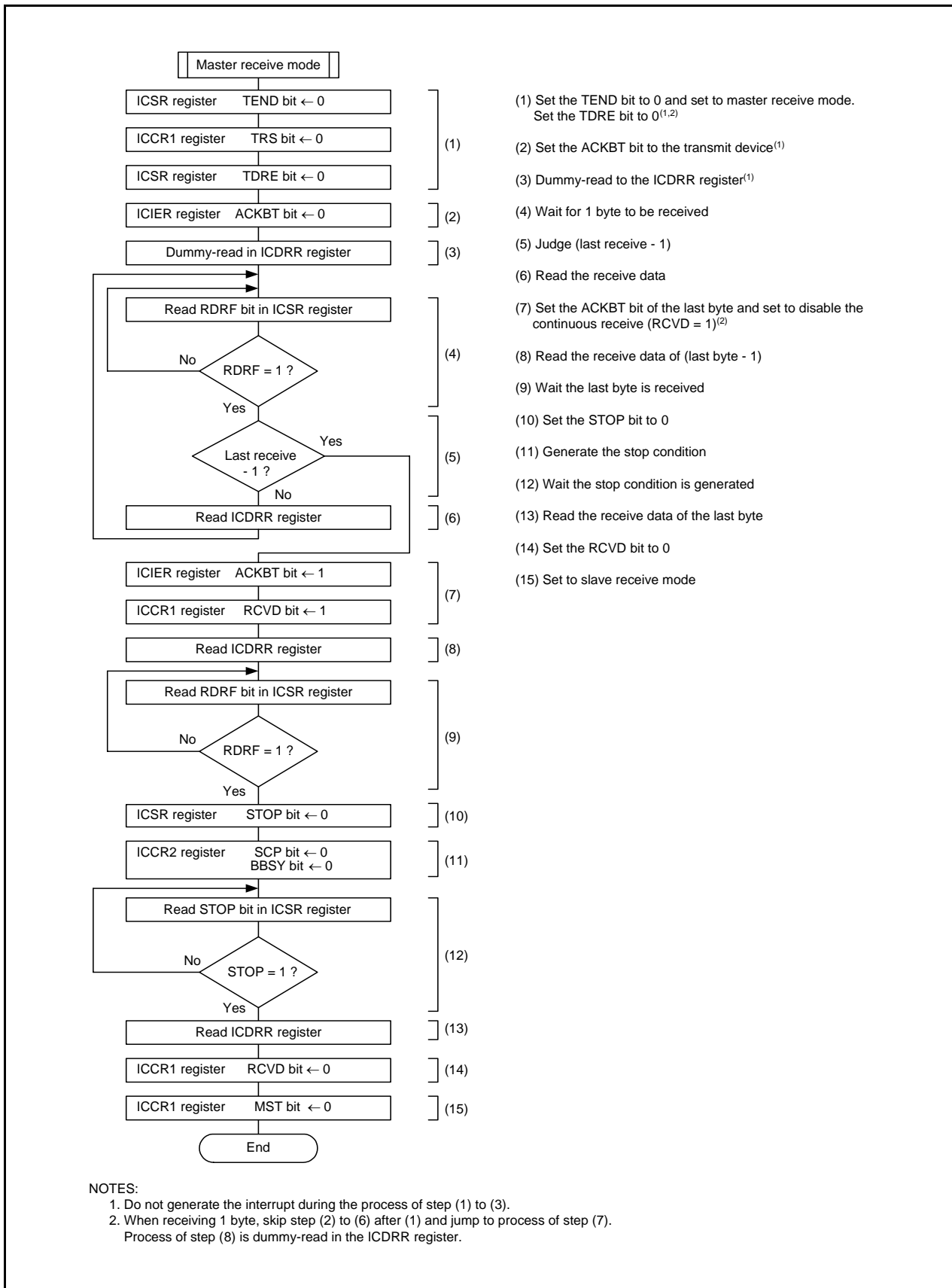


Figure 16.47 Example of Register Setting in Master Receive Mode (I<sup>2</sup>C Bus Interface Mode)

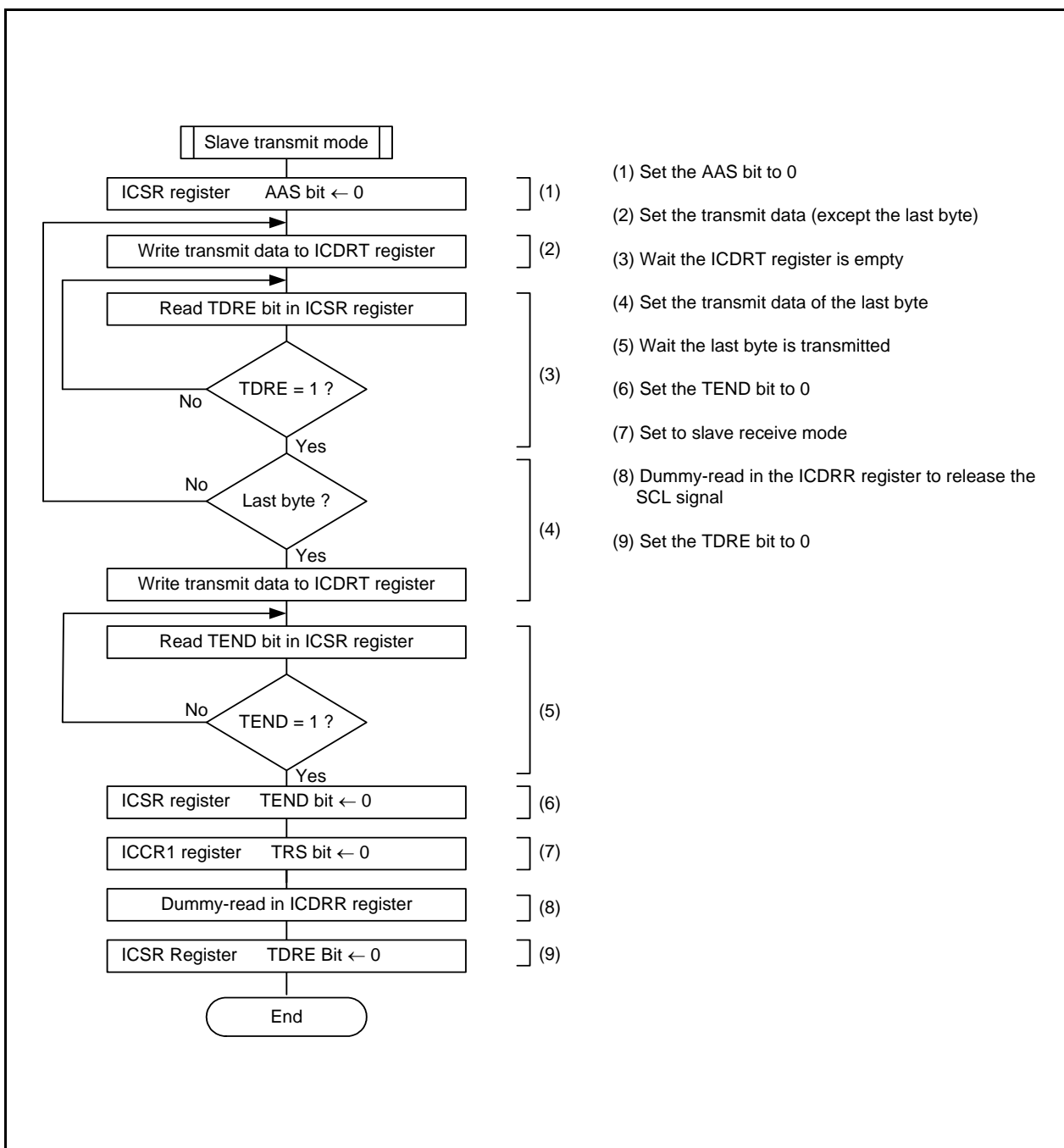


Figure 16.48 Example of Register Setting in Slave Transmit Mode (I<sup>2</sup>C Bus Interface Mode)

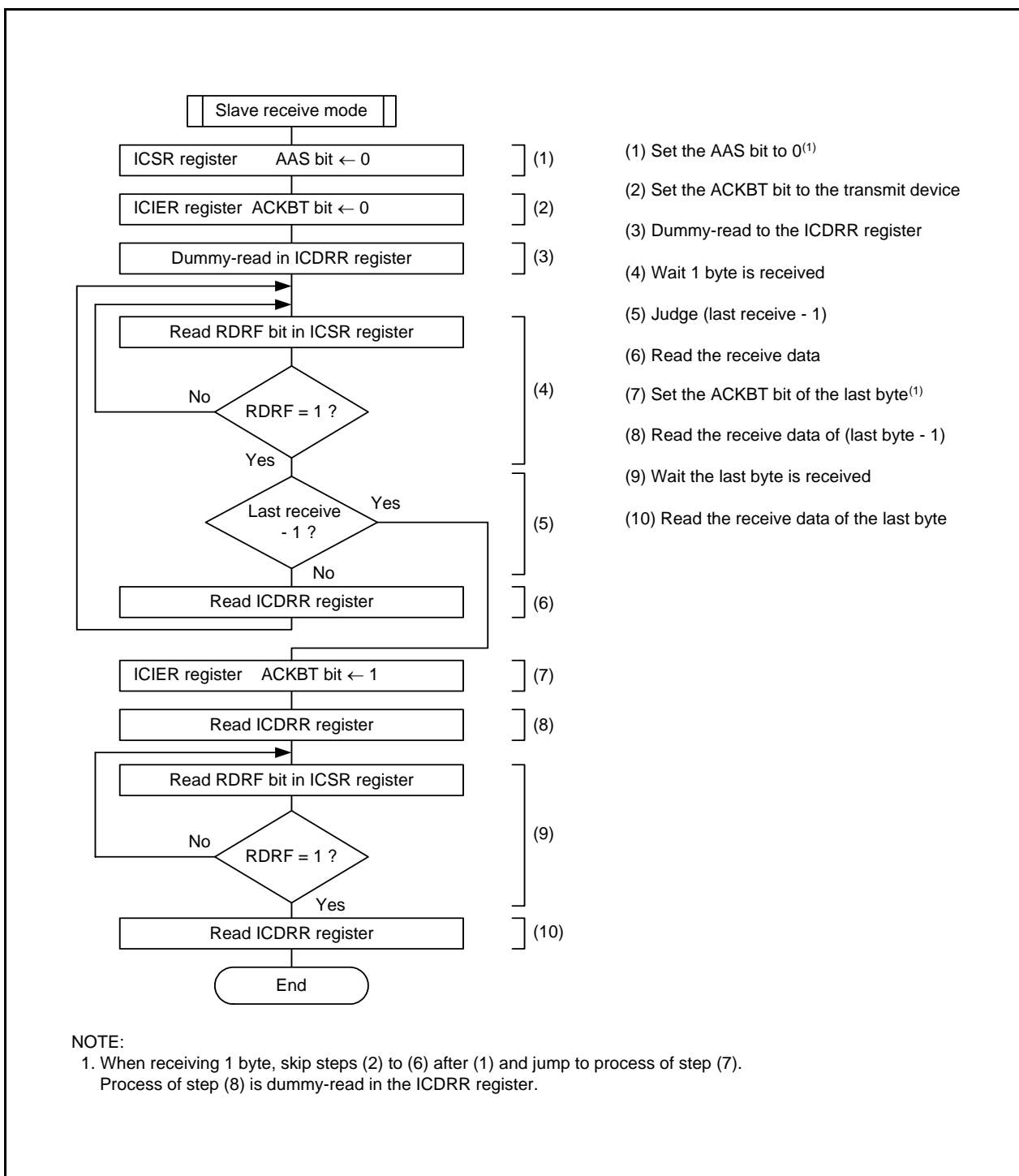


Figure 16.49 Example of Register Setting in Slave Receive Mode (I<sup>2</sup>C Bus Interface Mode)

### 16.3.8 Notes on I<sup>2</sup>C Bus Interface

Set the IICSEL bit in the PMR register to 1 (select I<sup>2</sup>C bus interface function) to use I<sup>2</sup>C bus interface.

#### 16.3.8.1 Multimaster Operation

The following actions must be performed to use the I<sup>2</sup>C bus interface in multimaster operation.

- Transfer rate  
Set the transfer rate by 1/1.8 or faster than the fastest rate of the other masters. For example, if the fastest transfer rate of the other masters is set to 400 kbps, the I<sup>2</sup>C-bus transfer rate in this MCU should be set to 223 kbps (= 400/1.8) or more.
- Bits MST and TRS in the ICCR1 register setting
  - (a) Use the MOV instruction to set bits MST and TRS.
  - (b) When arbitration is lost, confirm the contents of bits MST and TRS. If the contents are other than the MST bit set to 0 and the TRS bit set to 0 (slave receive mode), set the MST bit to 0 and the TRS bit to 0 again.

#### 16.3.8.2 Master Receive Mode

Either of the following actions must be performed to use the I<sup>2</sup>C bus interface in master receive mode.

- (a) In master receive mode while the RDRF bit in the ICSR register is set to 1, read the ICDRR register before the rising edge of the 8th clock.
- (b) In master receive mode, set the RCVD bit in the ICCR1 register to 1 (disables the next receive operation) to perform 1-byte communications.



## 17. Hardware LIN

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

### 17.1 Features

The hardware LIN has the following features.

Figure 17.1 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.

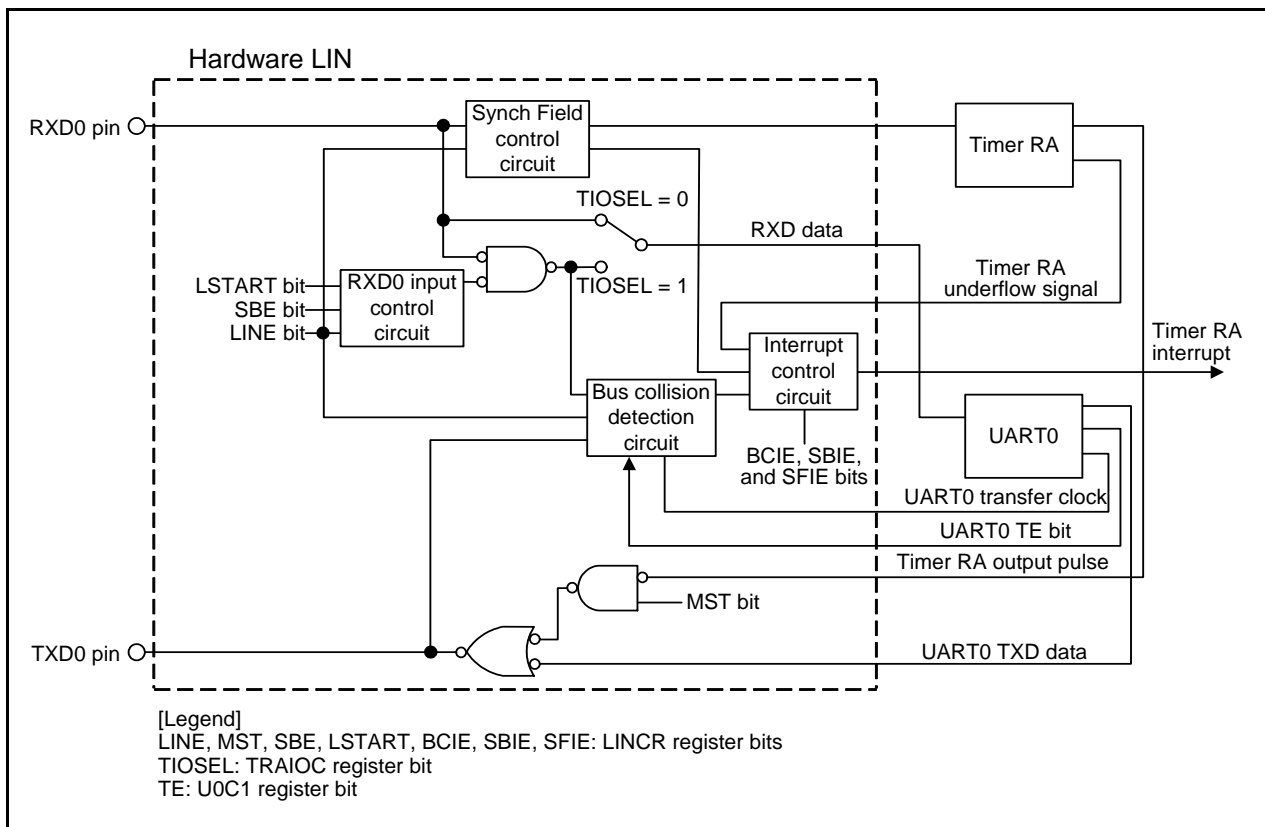


Figure 17.1 Block Diagram of Hardware LIN

## 17.2 Input/Output Pins

Table 17.1 lists the Pin Configuration of the hardware LIN.

**Table 17.1 Pin Configuration**

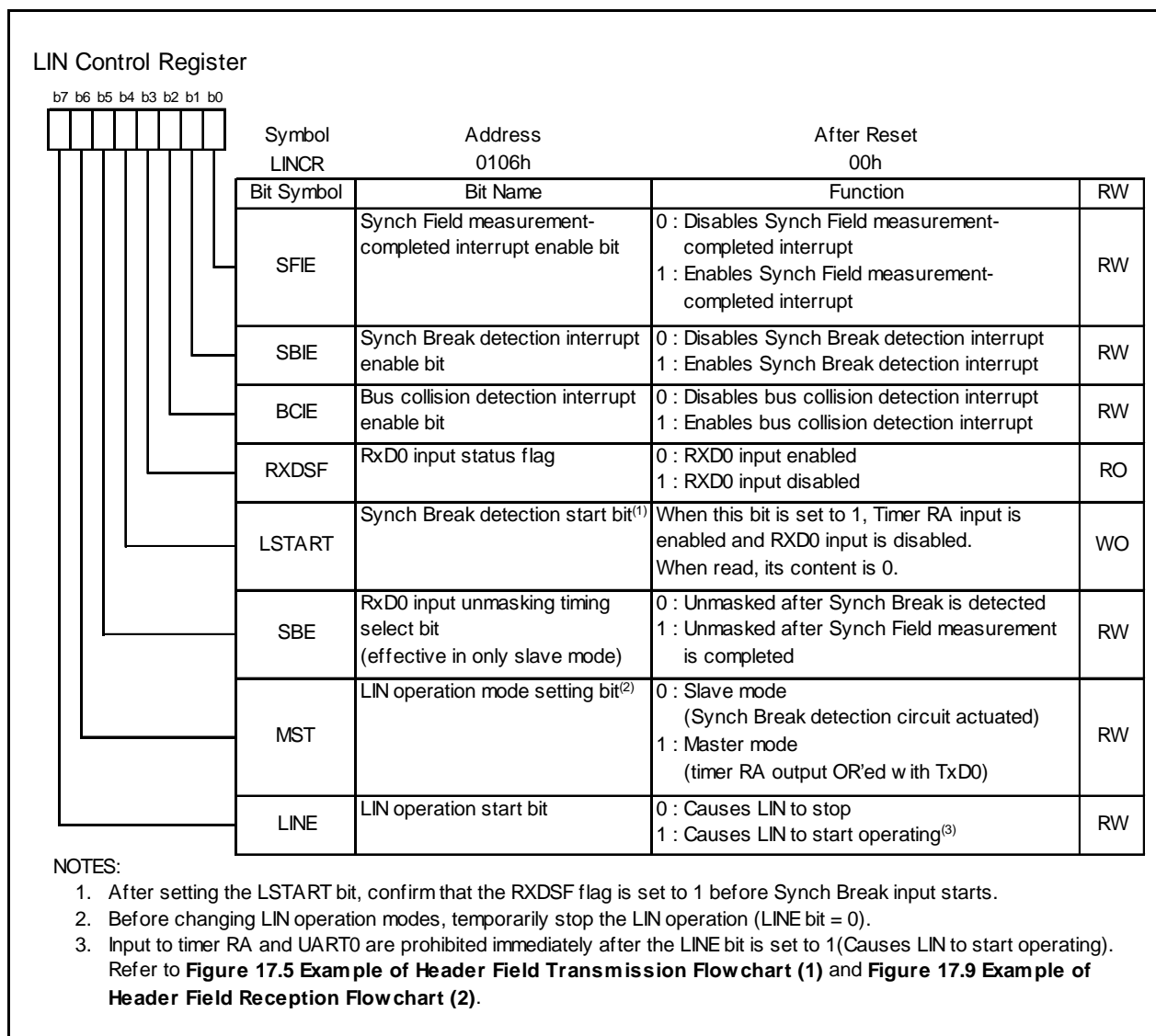
Name	Abbreviation	Input/Output	Function
Receive Data Input	RXD0	Input	Receive data input pin of the hardware LIN
Transmit Data Output	TXD0	Output	Transmit data output pin of the hardware LIN

### 17.3 Register Configuration

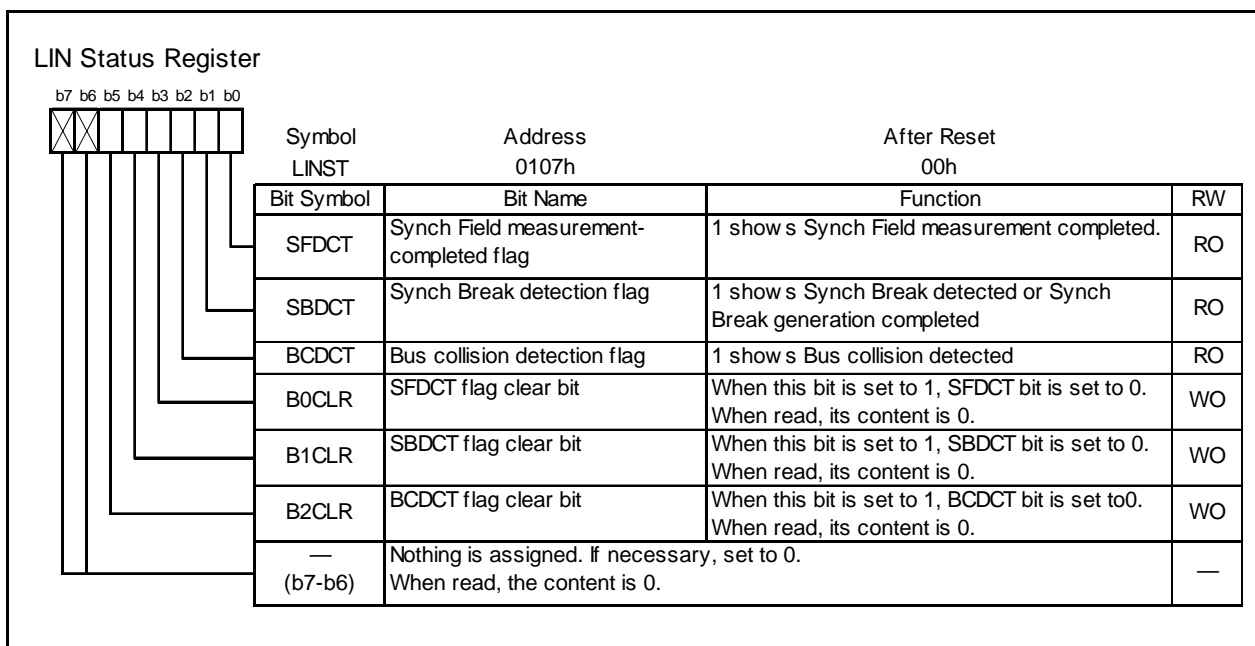
The hardware LIN contains the following registers.

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

Figure 17.2 and Figure 17.3 show the LINCRC and LINST Registers.



**Figure 17.2 LINCRC Register**



**Figure 17.3 LINST Register**

## 17.4 Functional Description

### 17.4.1 Master Mode

Figure 17.4 shows a Typical Operation when Sending a Header Field. Figure 17.5 and Figure 17.6 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 a program, the hardware LIN outputs a low-level signal from the TXD0 pin for the period that is set in the TRAPRE and TRA registers 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 finished sending 55h.
- (5) The hardware LIN performs communication for a response field after it finished sending the ID field.

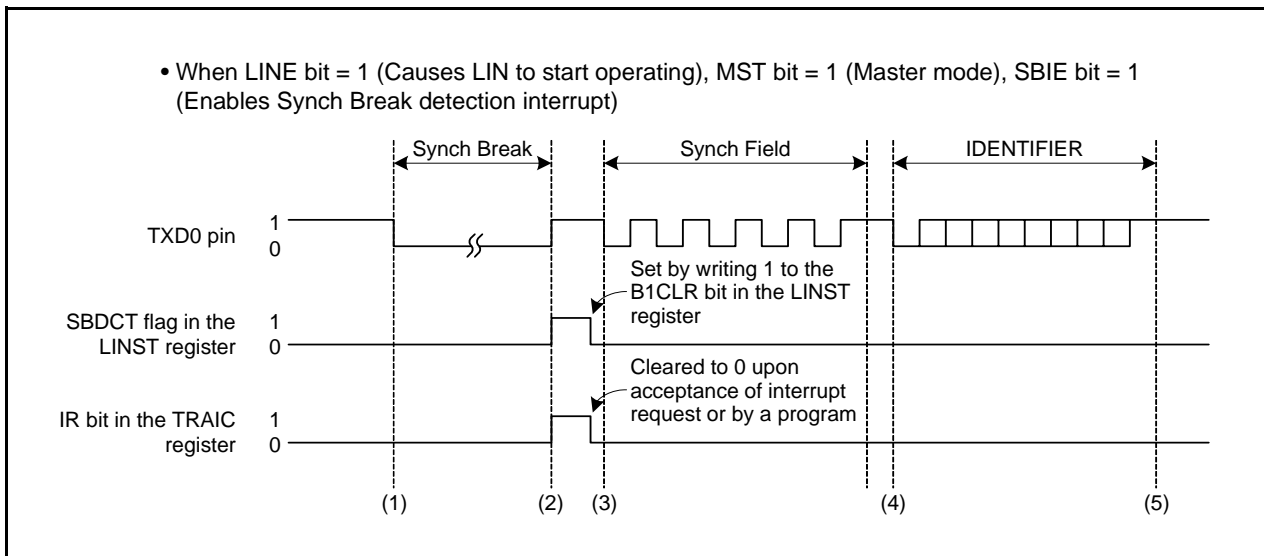


Figure 17.4 Typical Operation when Sending a Header Field

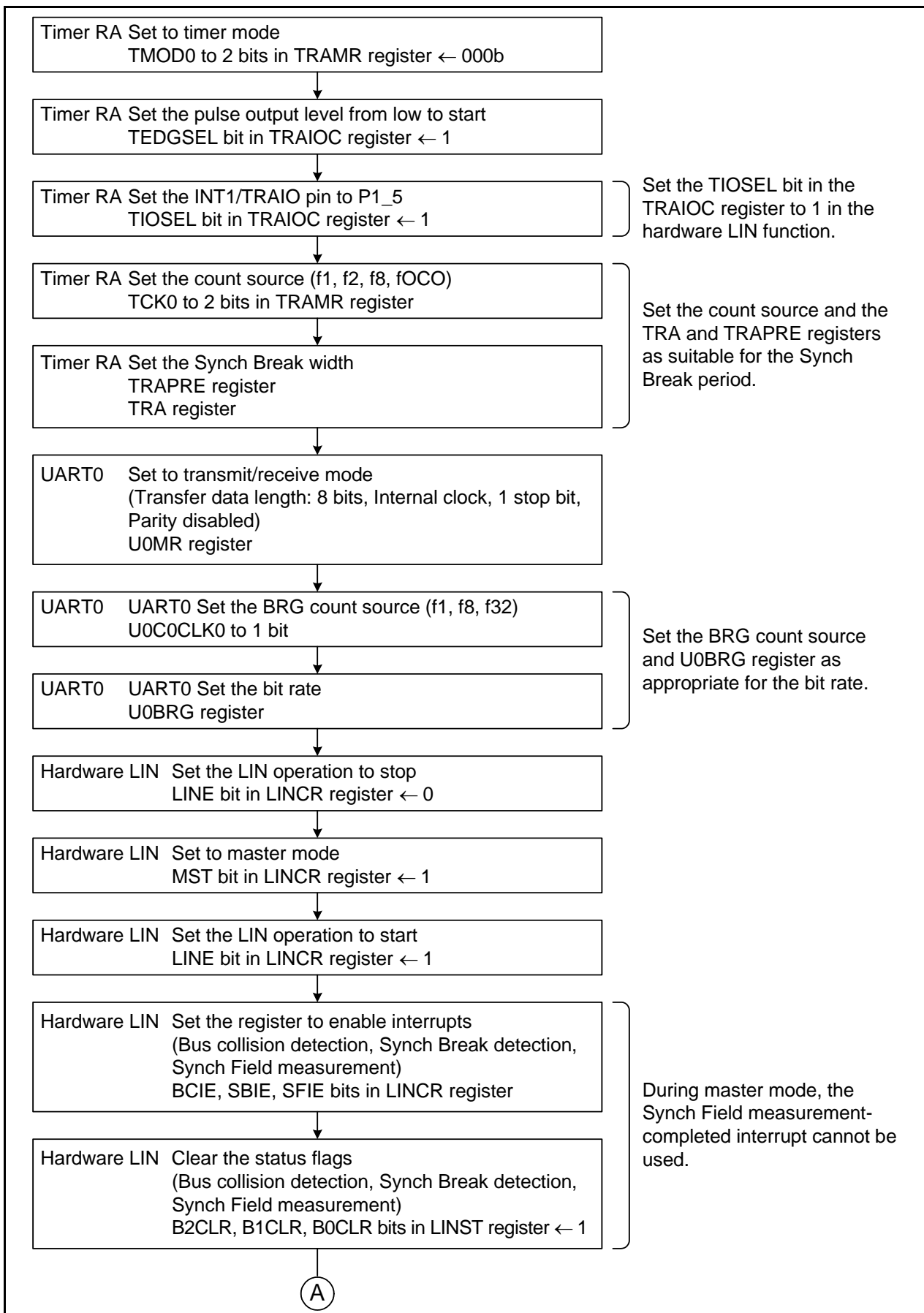


Figure 17.5 Example of Header Field Transmission Flowchart (1)

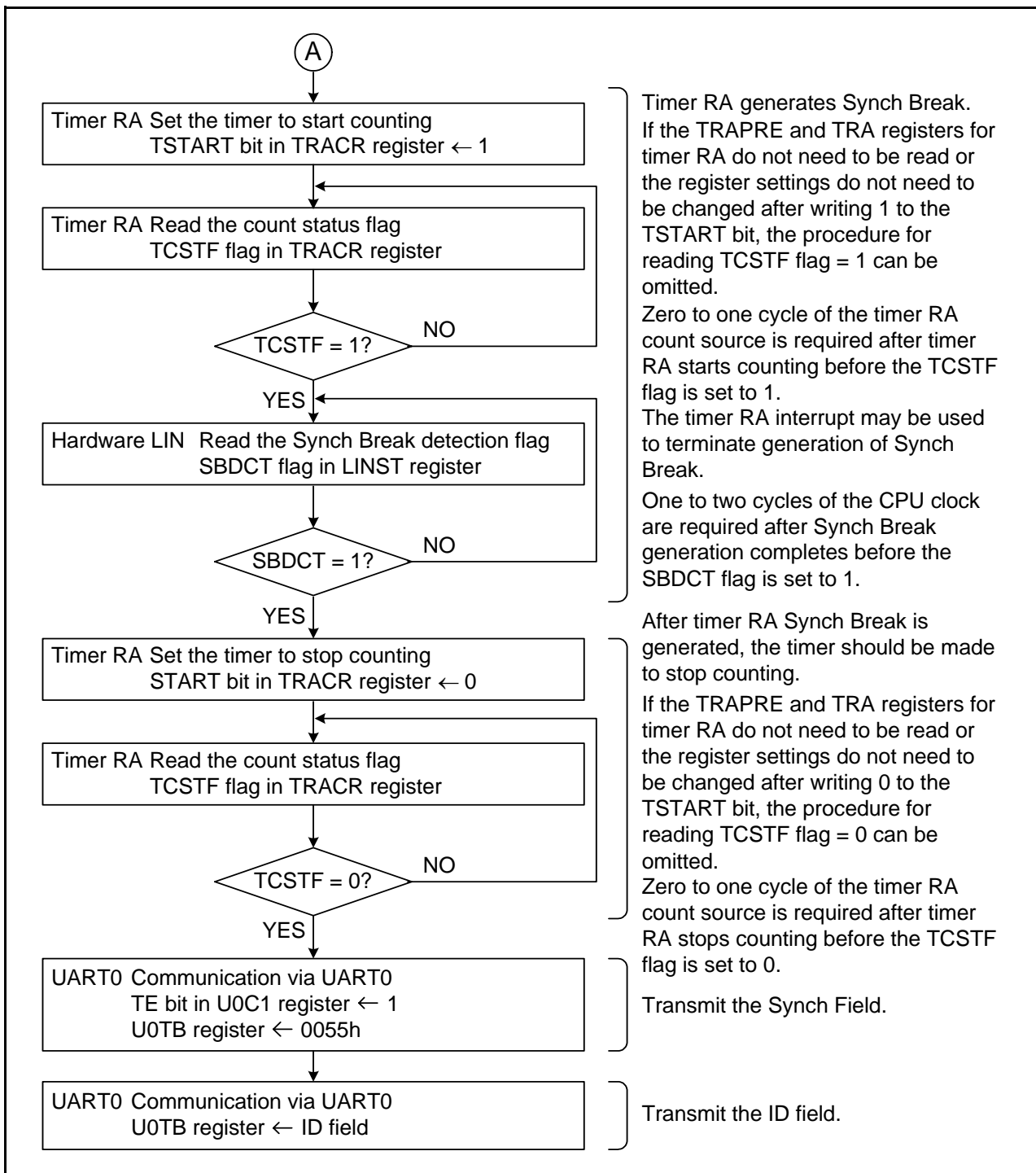


Figure 17.6 Example of Header Field Transmission Flowchart (2)

### 17.4.2 Slave Mode

Figure 17.7 shows a Typical Operation when Receiving a Header Field. Figure 17.8 through Figure 17.10 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 a low-level signal 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 finished 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 finished measuring the Synch Field, the hardware LIN calculates a transfer rate from the count value of timer RA and sets the result in UART0 and sets the TRAPRE and TRA registers of the timer RA back again. Then it receives an ID field via UART0.
- (6) The hardware LIN performs communication for a response field after it finished receiving the ID field.

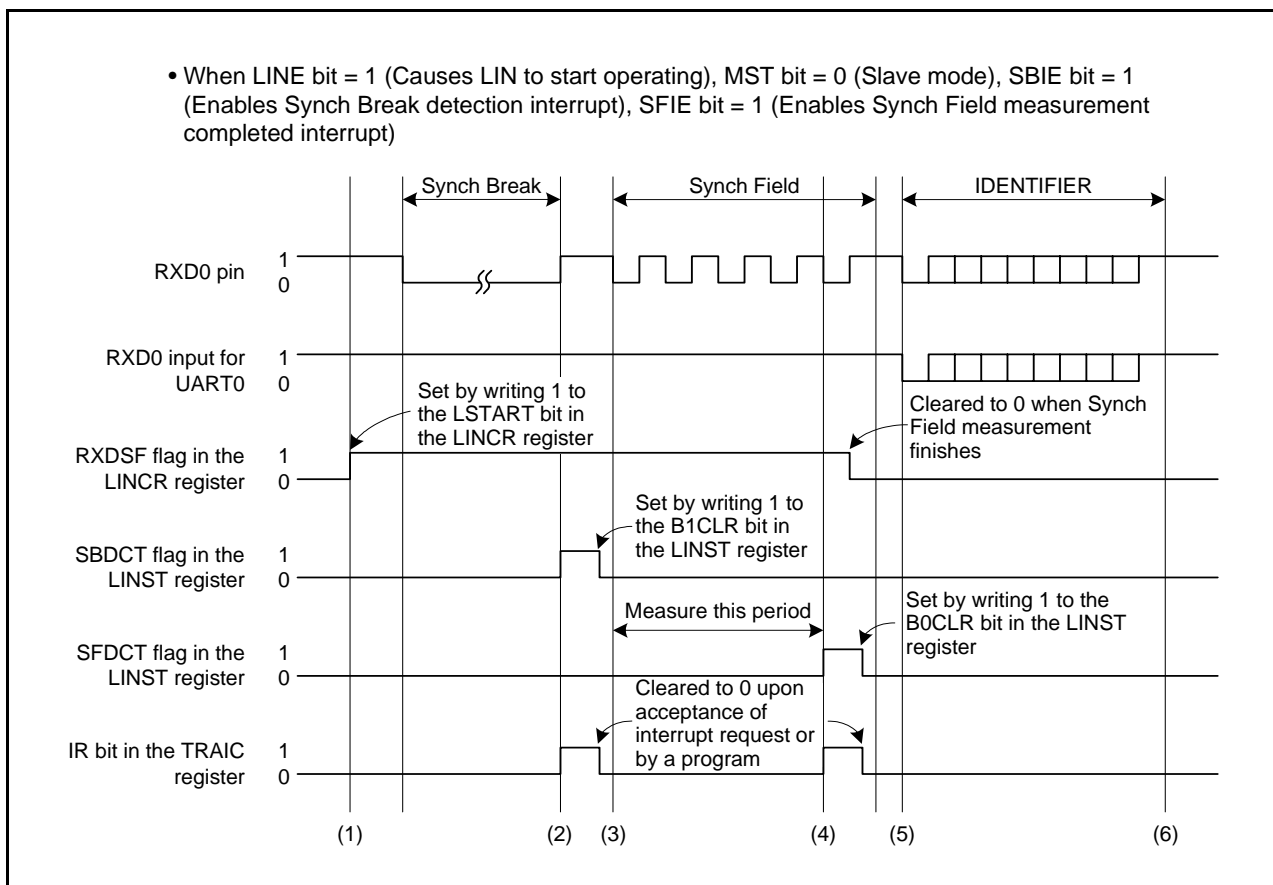


Figure 17.7 Typical Operation when Receiving a Header Field



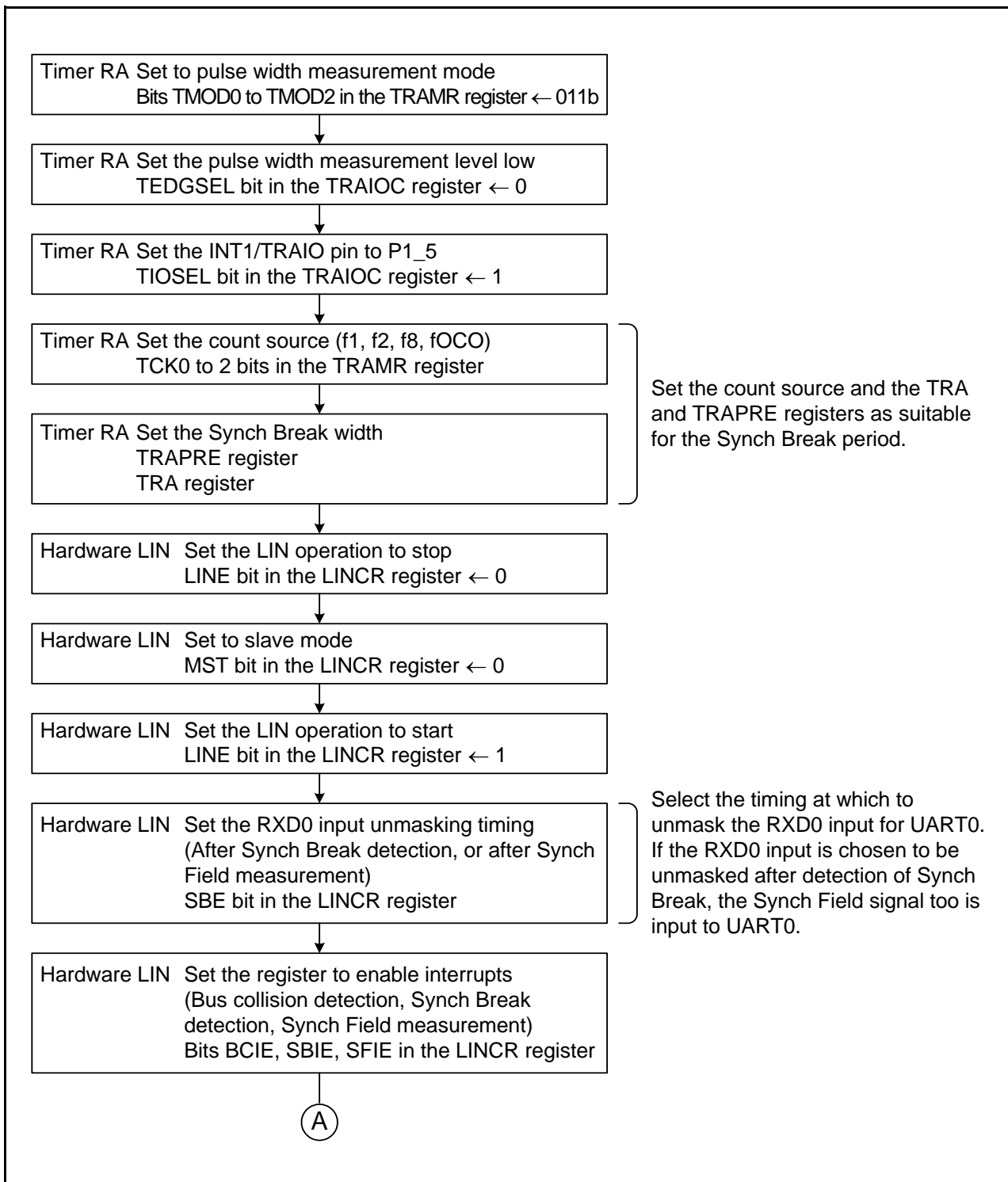


Figure 17.8 Example of Header Field Reception Flowchart (1)

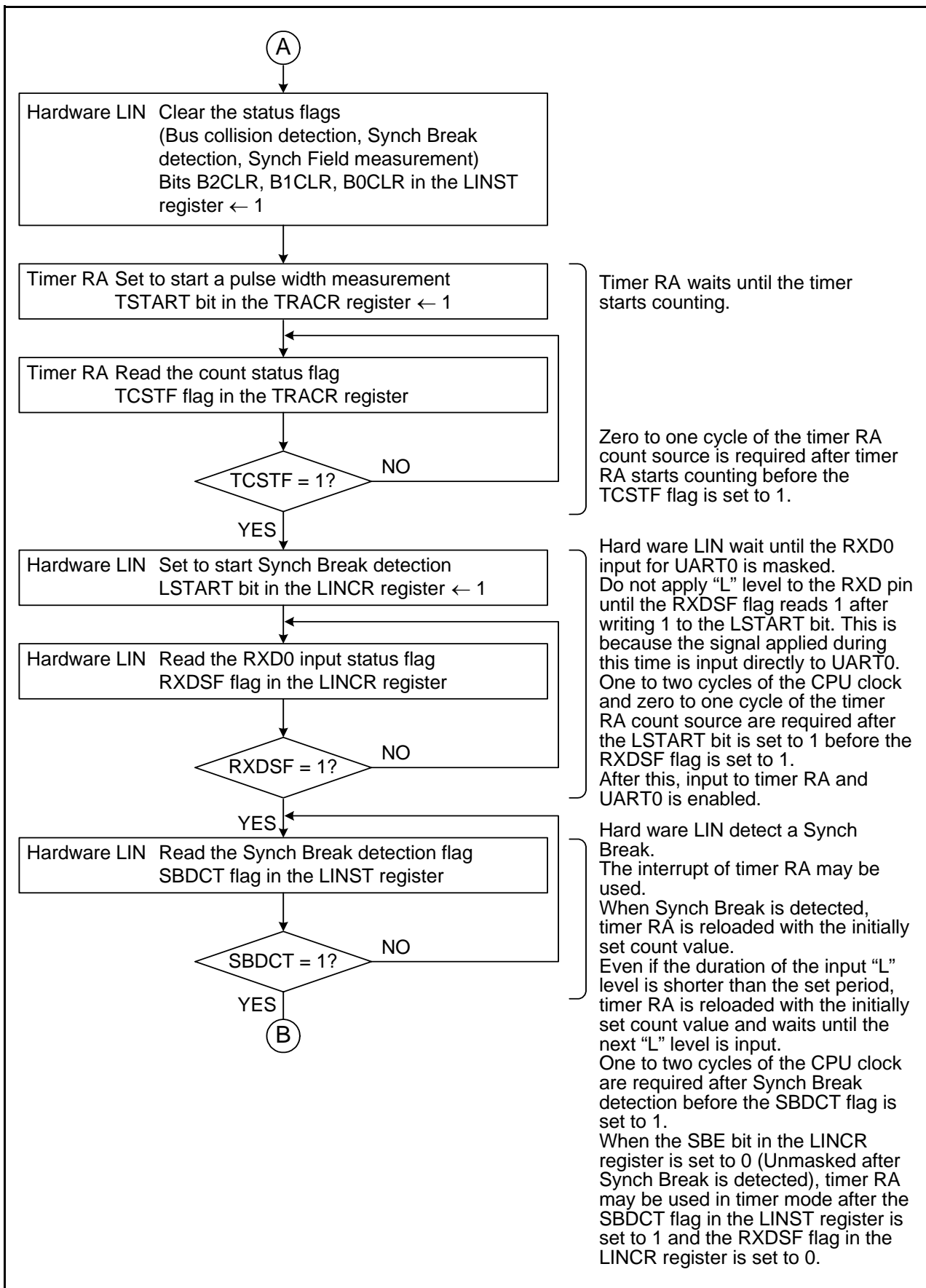


Figure 17.9 Example of Header Field Reception Flowchart (2)

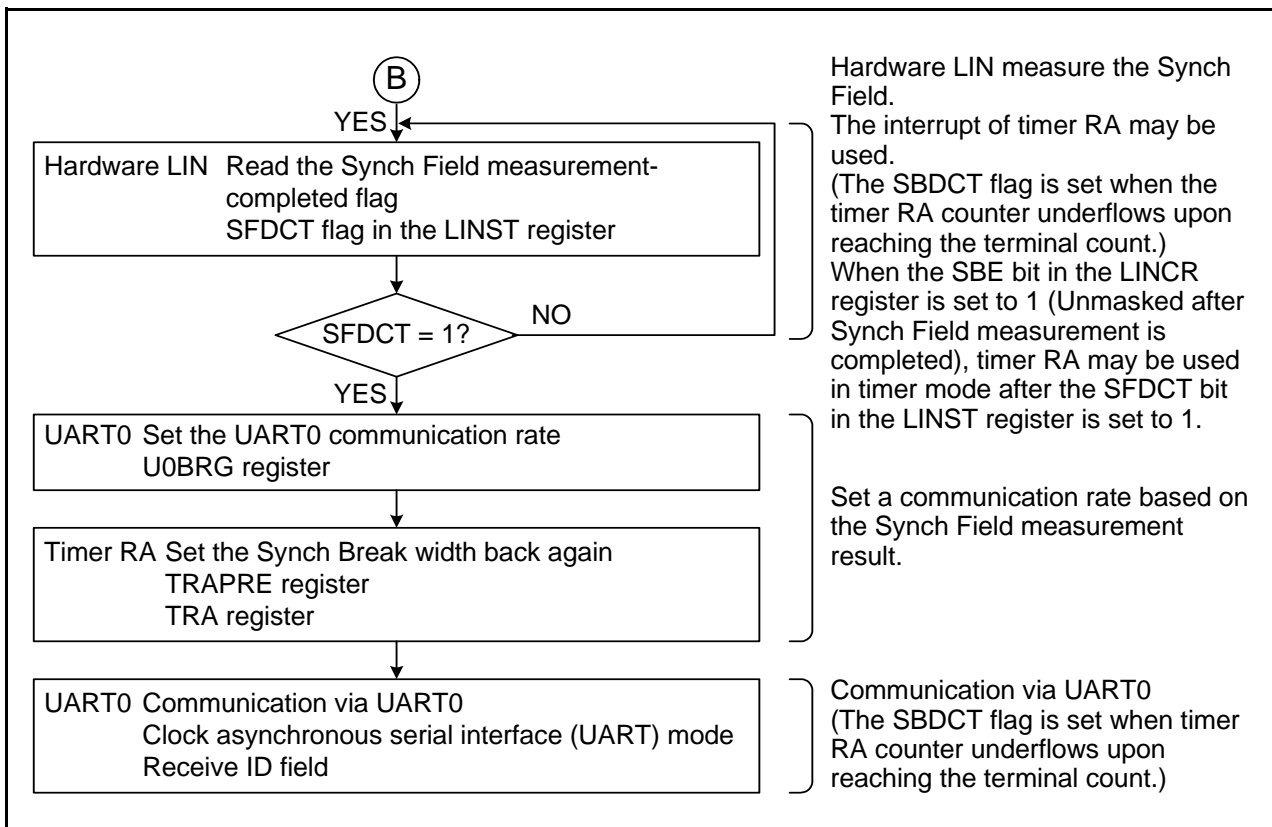


Figure 17.10 Example of Header Field Reception Flowchart (3)

### 17.4.3 Bus Collision Detection Function

The bus collision detection function can be used when UART0 is enabled for transmission (TE bit in the UOC1 register = 1).

Figure 17.11 shows a Typical Operation when a Bus Collision is Detected.

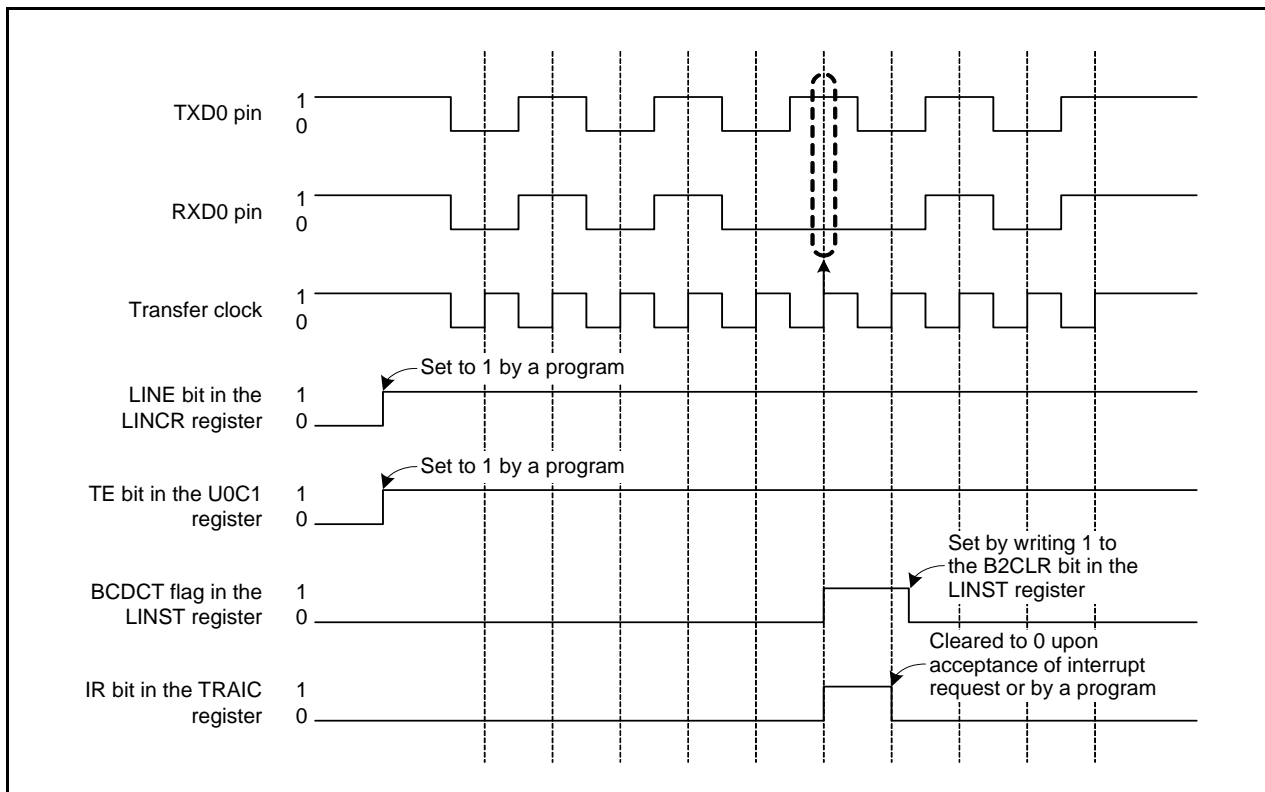


Figure 17.11 Typical Operation when a Bus Collision is Detected

### 17.4.4 Hardware LIN End Processing

Figure 17.12 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.

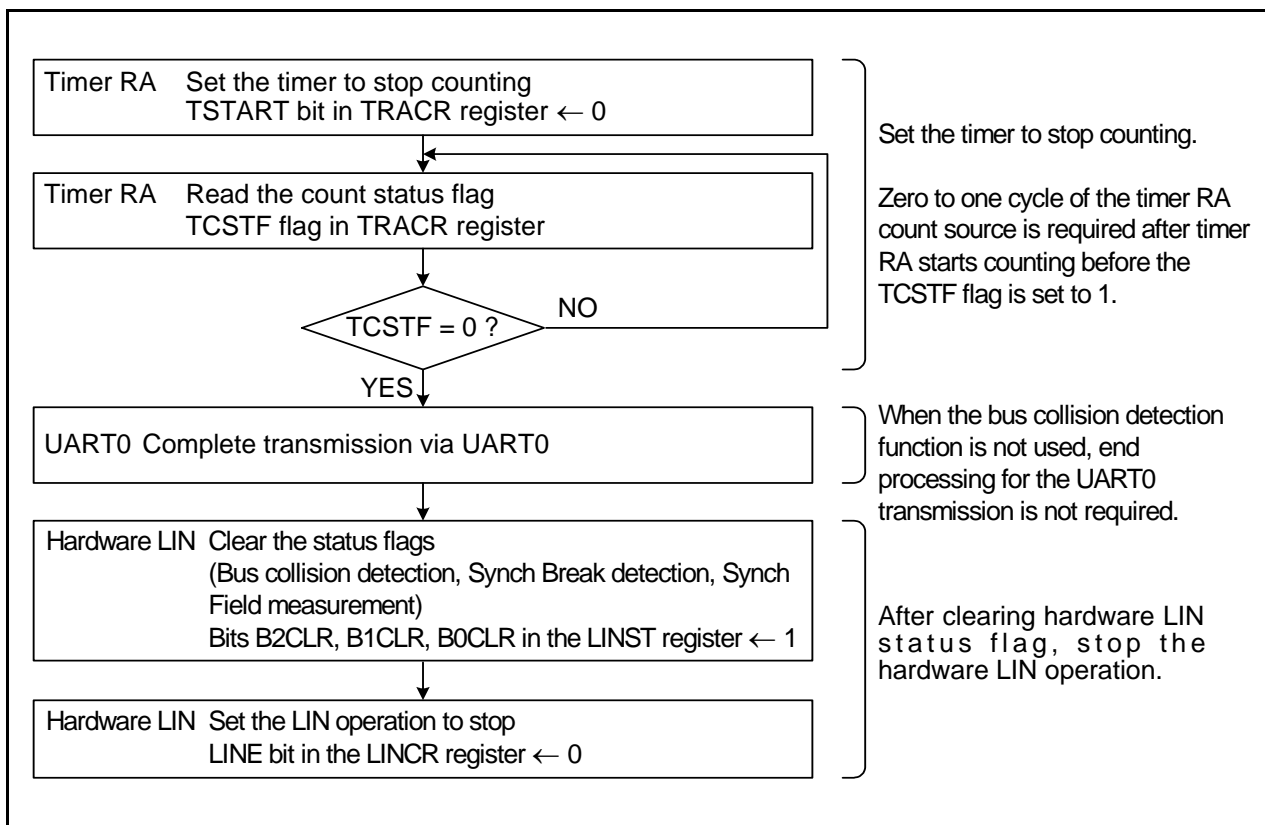


Figure 17.12 Example of Hardware LIN Communication Completion Flowchart

## 17.5 Interrupt Requests

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

Table 17.2 lists the Interrupt Requests of Hardware LIN.

**Table 17.2 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 low level duration of RXD0 input, or when a low-level signal is input for a duration longer than the Synch Break period during communication.
Synch Break Generation Completed		Generated when timer RA has completed outputting a low-level signal to TXD0 for set period.
Synch Field Measurement	SFDCT	Generated when measurement for 8 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.

## 17.6 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 respect to a Synch Break detection interrupt as the starting point.

## 18. A/D Converter

The A/D converter consists of one 10-bit successive approximation A/D converter circuit with a capacitive coupling amplifier. The analog input shares the pins with P0\_0 to P0\_7, P1\_0 to P1\_3. Therefore, when using these pins, ensure the corresponding port direction bits are set to 0 (input mode).

When not using the A/D converter, set the VCUT bit in the ADCON1 register to 0 (Vref unconnected), so that no current will flow from the VREF pin into the resistor ladder, helping to reduce the power consumption of the chip.

The result of A/D conversion is stored in the AD register.

Table 18.1 lists the Performance of A/D Converter. Figure 18.1 shows the Block Diagram of A/D Converter. Figure 18.2 and Figure 18.3 show the A/D converter-related registers.

**Table 18.1 Performance of A/D Converter**

Item	Performance
A/D Conversion Method	Successive approximation (with capacitive coupling amplifier)
Analog Input Voltage <sup>(1)</sup>	0 V to AVCC
Operating Clock $\phi_{AD}$ <sup>(2)</sup>	4.2 V $\leq$ AVCC $\leq$ 5.5 V f1, f2, f4, fOCO-F 2.7 V $\leq$ AVCC < 4.2 V f2, f4, fOCO-F
Resolution	8 bit or 10 bit is selectable
Absolute Accuracy	AVCC = Vref = 5 V, $\phi_{AD}$ = 10MHz <ul style="list-style-type: none"> <li>• 8-bit resolution <math>\pm 2</math> LSB</li> <li>• 10-bit resolution <math>\pm 3</math> LSB</li> </ul> AVCC = Vref = 3.3 V, $\phi_{AD}$ = 10MHz <ul style="list-style-type: none"> <li>• 8-bit resolution <math>\pm 2</math> LSB</li> <li>• 10-bit resolution <math>\pm 5</math> LSB</li> </ul>
Operating Mode	One-shot and repeat modes <sup>(3)</sup>
Analog Input Pin	12 pins (AN0 to AN11)
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>• Software trigger Set the ADST bit in the ADCON0 register to 1 (A/D conversion starts)</li> <li>• Capture Timer RD interrupt request is generated while the ADST bit is set to 1</li> </ul>
Conversion Rate Per Pin	<ul style="list-style-type: none"> <li>• Without sample and hold function 8-bit resolution: 49<math>\phi_{AD}</math> cycles, 10-bit resolution: 59<math>\phi_{AD}</math> cycles</li> <li>• With sample and hold function 8-bit resolution: 28<math>\phi_{AD}</math> cycles, 10-bit resolution: 33<math>\phi_{AD}</math> cycles</li> </ul>

### NOTES:

1. Analog input voltage does not depend on use of sample and hold function.  
When analog input voltage exceeds reference voltage, A/D conversion result is 3FFh in 10-bit mode, FFh in 8-bit mode.
2. The frequency of  $\phi_{AD}$  must be 10 MHz or below.  
Without sample and hold function, the  $\phi_{AD}$  frequency should be 250 kHz or above.  
With the sample and hold function, the  $\phi_{AD}$  frequency should be 1 MHz or above.
3. In repeat mode, only 8-bit mode can be used.



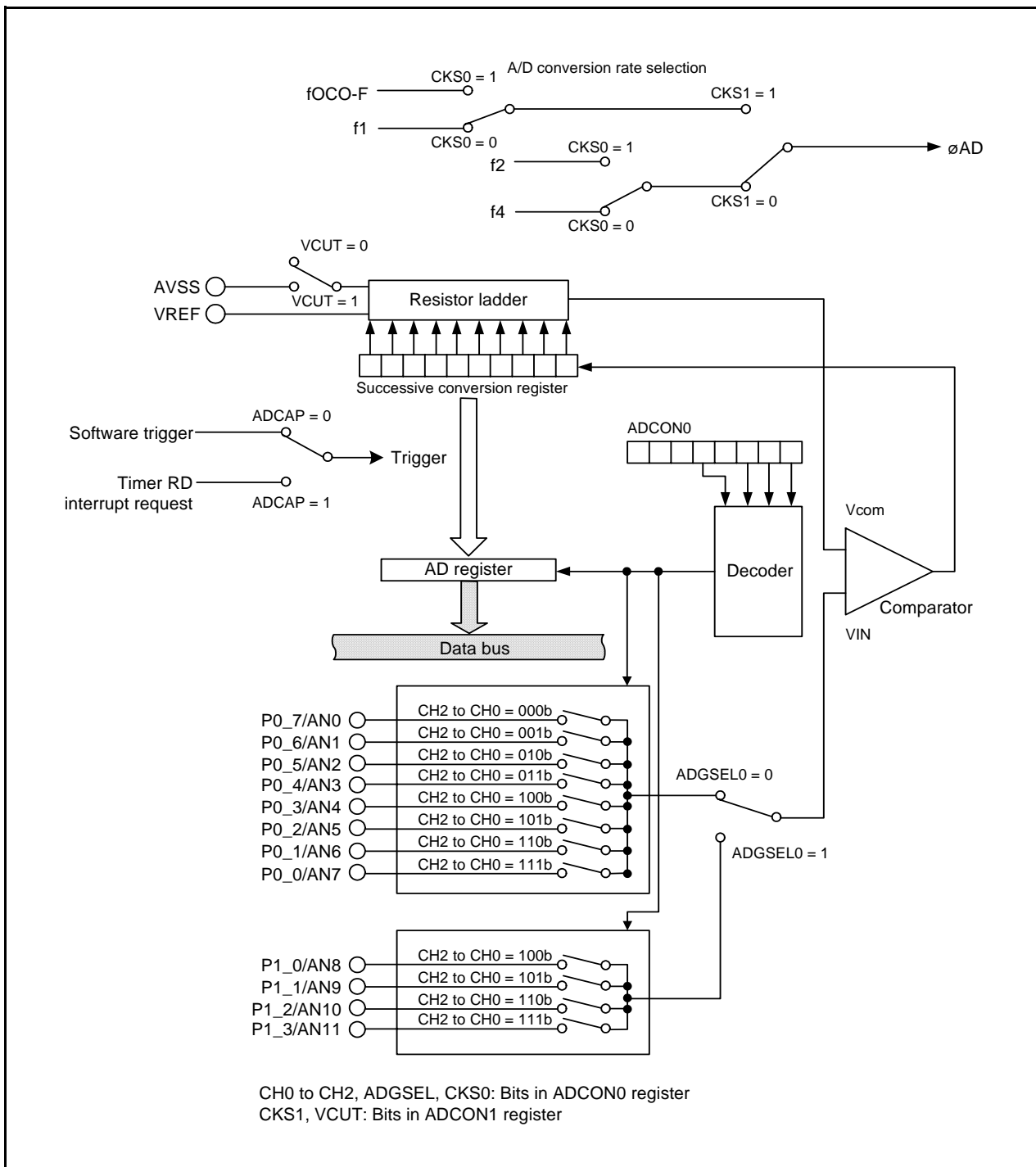


Figure 18.1 Block Diagram of A/D Converter

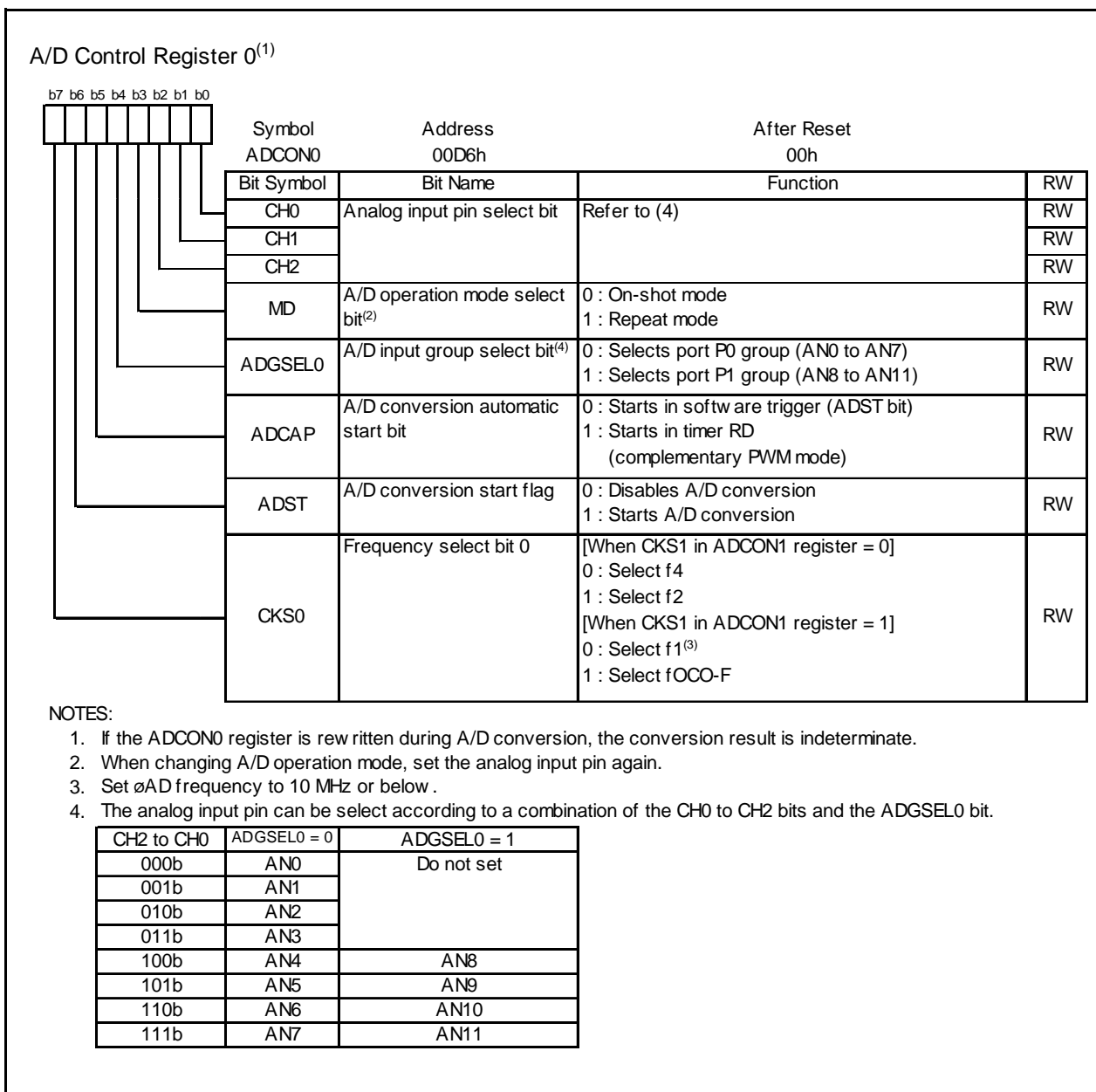


Figure 18.2 ADCON0 Register

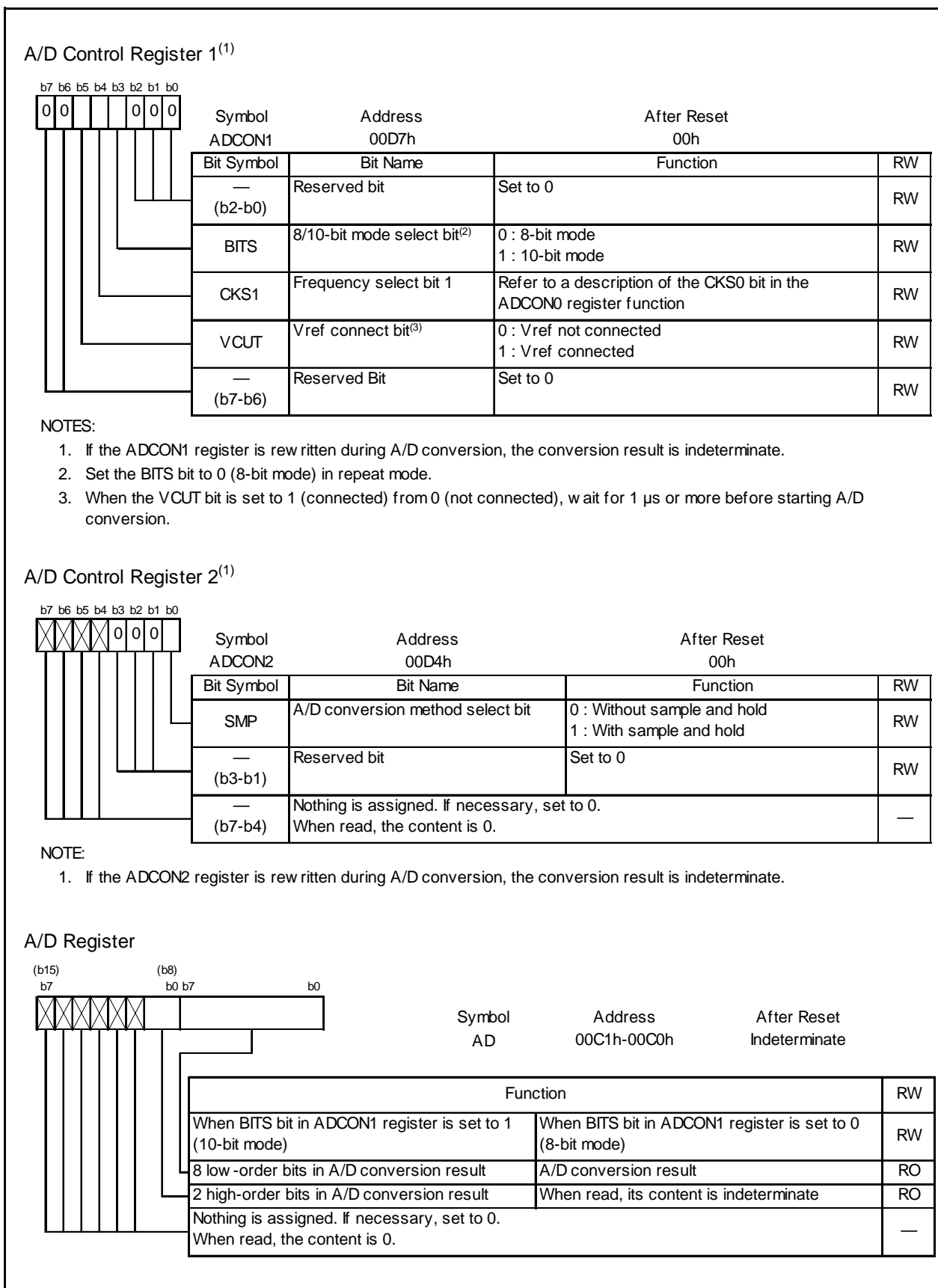


Figure 18.3 Registers ADCON1, ADCON2, and AD

## 18.1 One-Shot Mode

In one-shot mode, the input voltage on one selected pin is A/D converted once.

Table 18.2 lists the One-Shot Mode Specifications. Figure 18.4 shows the ADCON0 Register in One-Shot Mode and Figure 18.5 shows the ADCON1 Register in One-Shot Mode.

**Table 18.2 One-Shot Mode Specifications**

Item	Specification
Function	The input voltage on one selected pin by bits CH2 to CH0 and ADGSEL0 is A/D converted once
Start Condition	<ul style="list-style-type: none"> <li>• When the ADCAP bit is set to 0 (software trigger), Set the ADST bit to 1 (A/D conversion starts)</li> <li>• When the ADCAP bit is set to 1 (starts in timer RD (complementary PWM mode)), The compare match in the TRD0 and TRDGRA0 registers or the TRD1 underflow is generated while the ADST bit is set to 1</li> </ul>
Stop Condition	<ul style="list-style-type: none"> <li>• A/D conversion completes (When the ADCAP bit is set to 0 (software trigger), the ADST bit is set to 0)</li> <li>• Set the ADST bit to 0</li> </ul>
Interrupt Request Generation Timing	A/D conversion completes
Input Pin	Select one of AN0 to AN11
Reading of A/D Conversion Result	Read the AD register

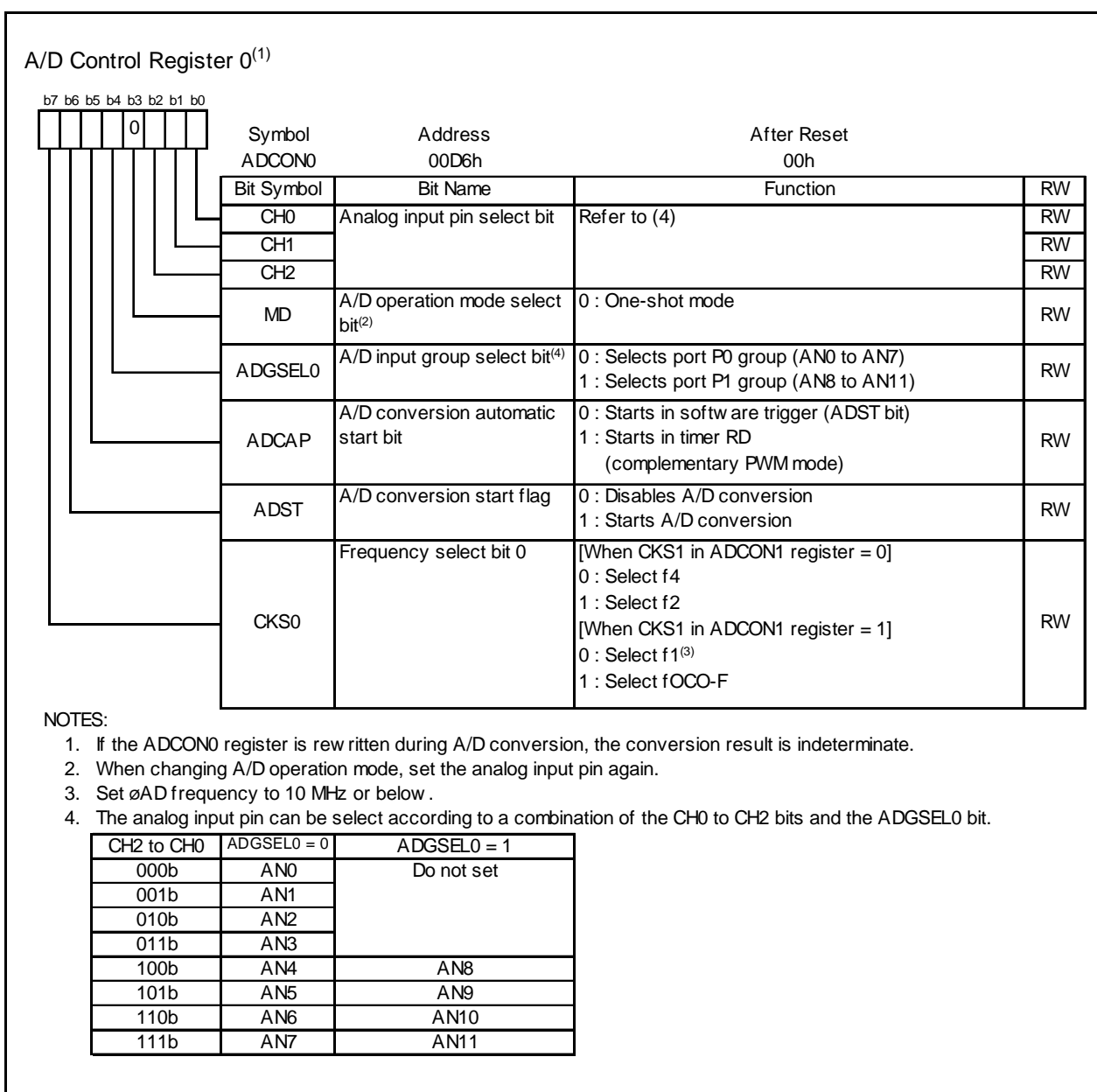


Figure 18.4 ADCON0 Register in One-Shot Mode

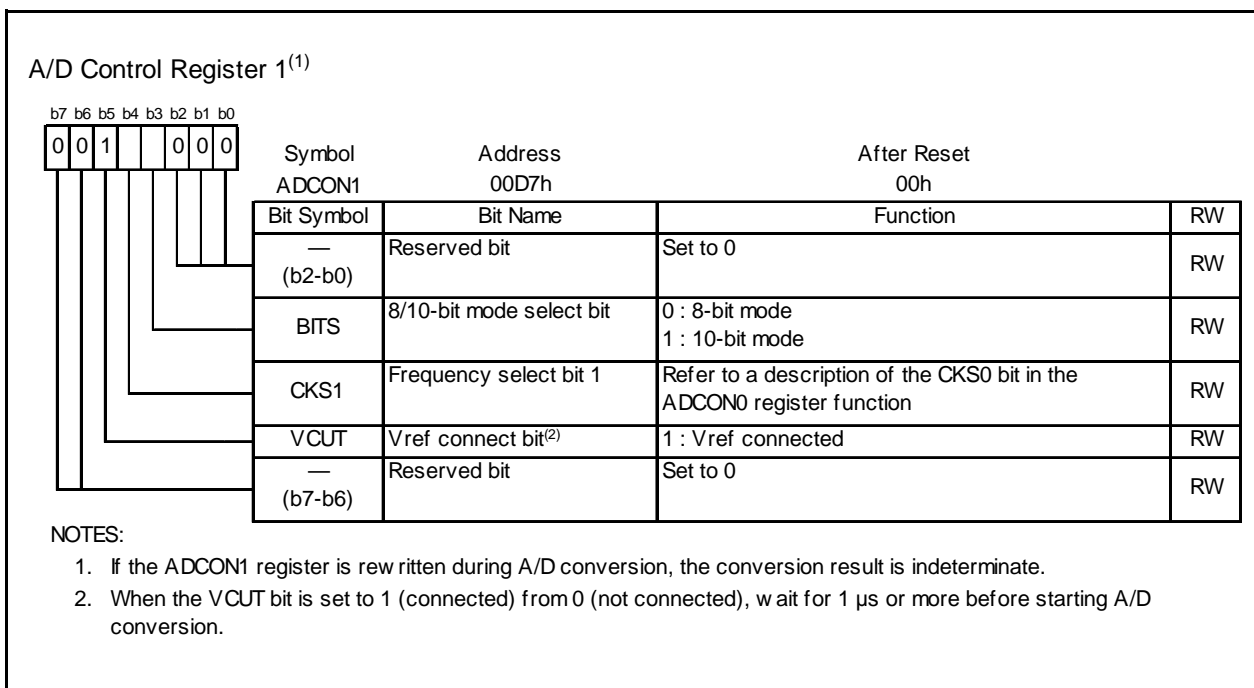


Figure 18.5 ADCON1 Register in One-Shot Mode

## 18.2 Repeat Mode

In repeat mode, the input voltage on one selected pin is A/D converted repeatedly.

Table 18.3 lists the Repeat Mode Specifications. Figure 18.6 shows the ADCON0 Register in Repeat Mode and Figure 18.7 shows the ADCON1 Register in Repeat Mode.

**Table 18.3 Repeat Mode Specifications**

Item	Specification
Function	The Input voltage on one pin selected by CH2 to CH0 and ADGSEL0 bits is A/D converted repeatedly
Start Condition	<ul style="list-style-type: none"> <li>• When the ADCAP bit is set to 0 (software trigger) Set the ADST bit to 1 (A/D conversion starts)</li> <li>• When the ADCAP bit is set to 1 (starts in timer RD (complementary PWM mode)), The compare match in the TRD0 and TRDGRA0 registers or the TRD1 underflow is generated while the ADST bit is set to 1</li> </ul>
Stop Condition	Set the ADST bit to 0
Interrupt Request Generation Timing	Not generated
Input Pin	Select one of AN0 to AN11
Reading of Result of A/D Converter	Read the AD register

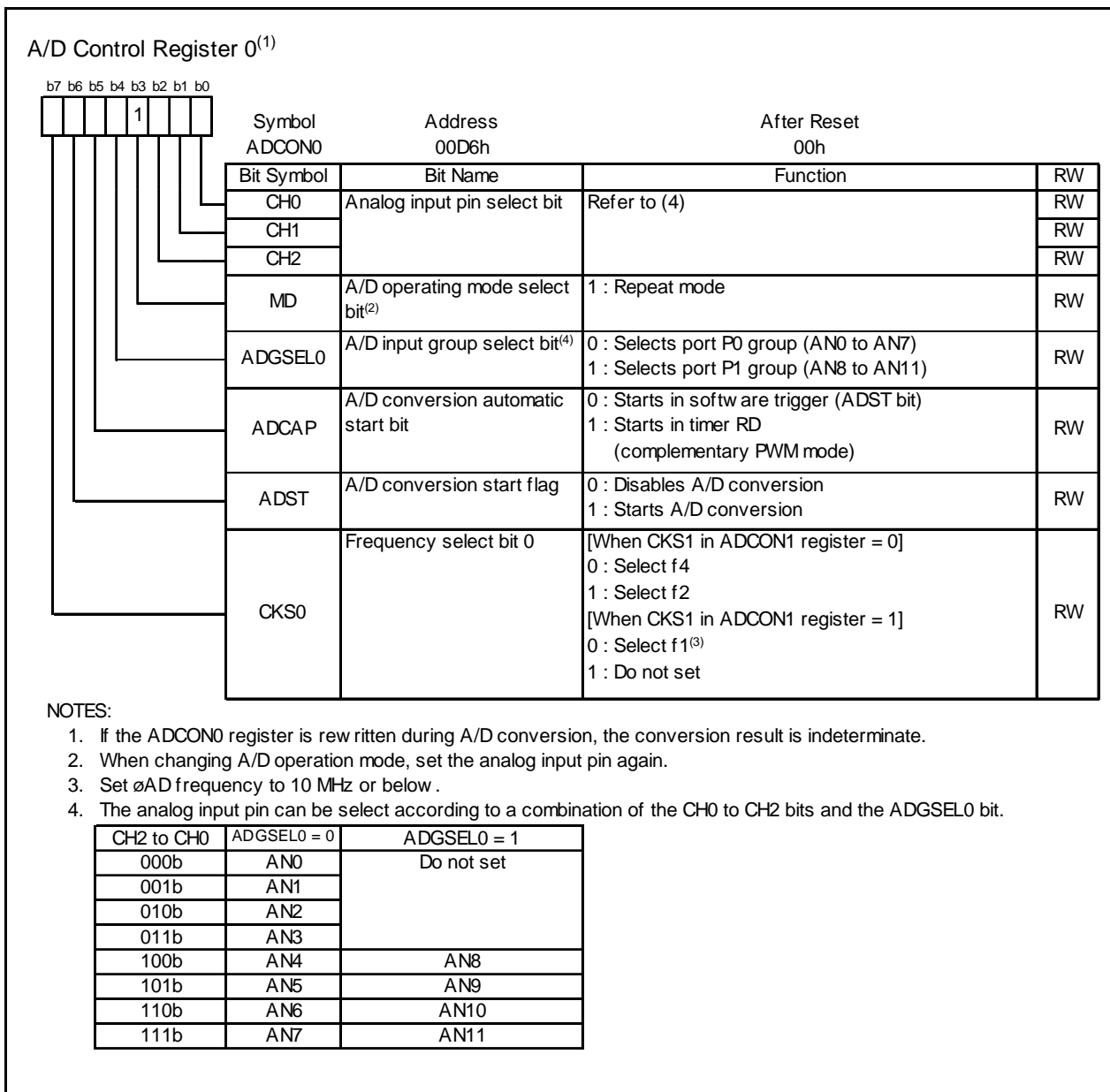
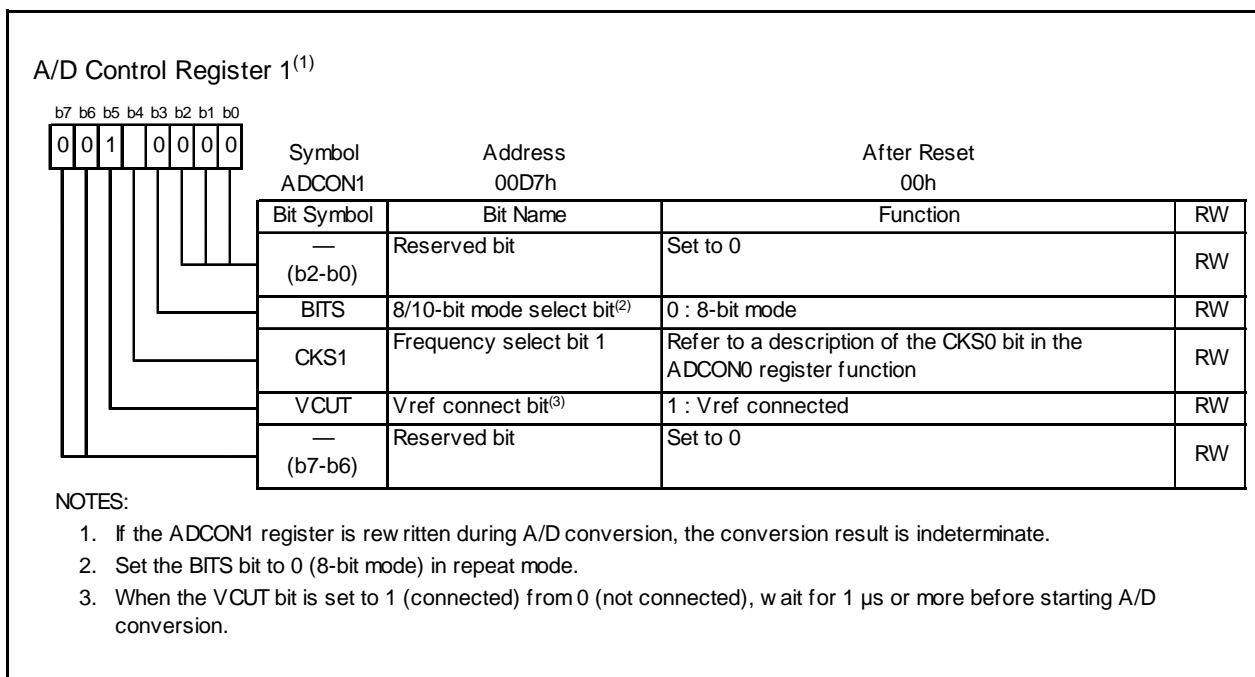


Figure 18.6 ADCON0 Register in Repeat Mode





**Figure 18.7 ADCON1 Register in Repeat Mode**

### 18.3 Sample and Hold

When the SMP bit in the ADCON2 register is set to 1 (with sample and hold function), A/D conversion rate per pin increases. The sample and hold function is available in all operating modes. Start the A/D conversion after selecting whether the sample and hold circuit is to be used or not.

Figure 18.8 shows the Timing Diagram of A/D Conversion.

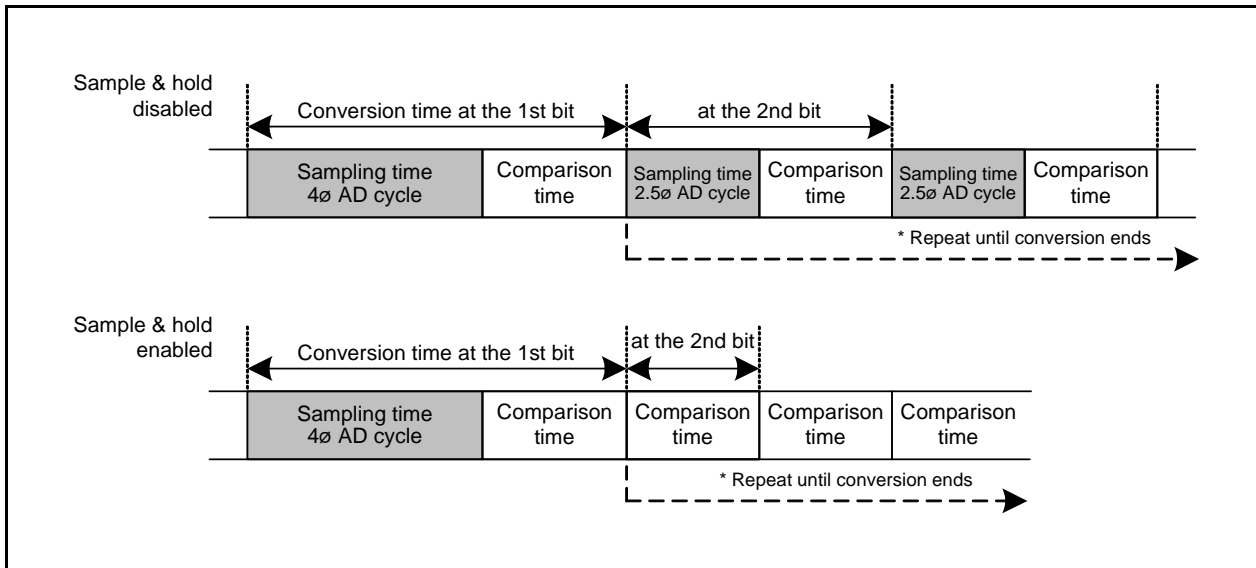


Figure 18.8 Timing Diagram of A/D Conversion

### 18.4 A/D Conversion Cycles

Figure 18.9 shows the A/D Conversion Cycles.

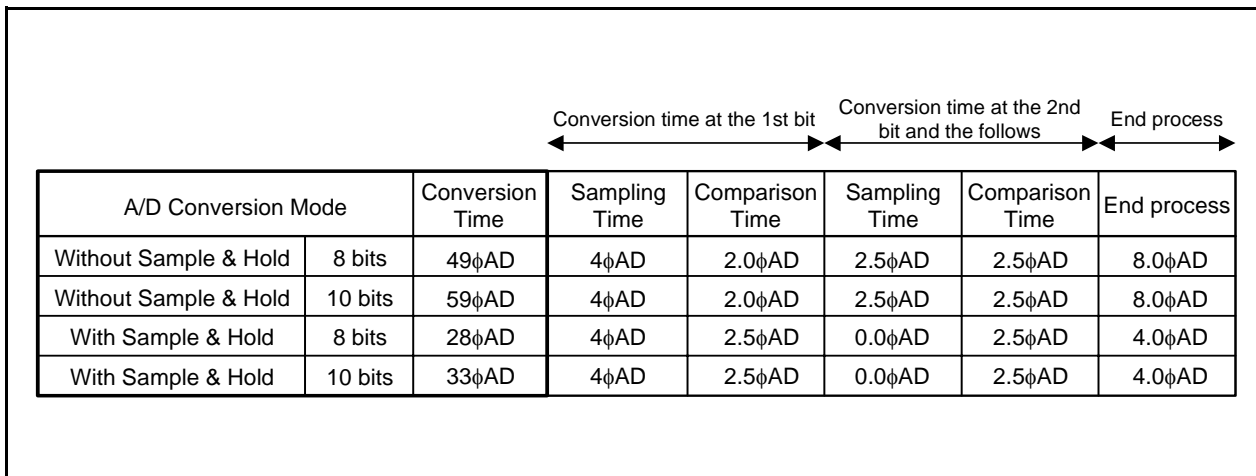


Figure 18.9 A/D Conversion Cycles

### 18.5 Internal Equivalent Circuit of Analog Input

Figure 18.10 shows the Internal Equivalent Circuit of Analog Input.

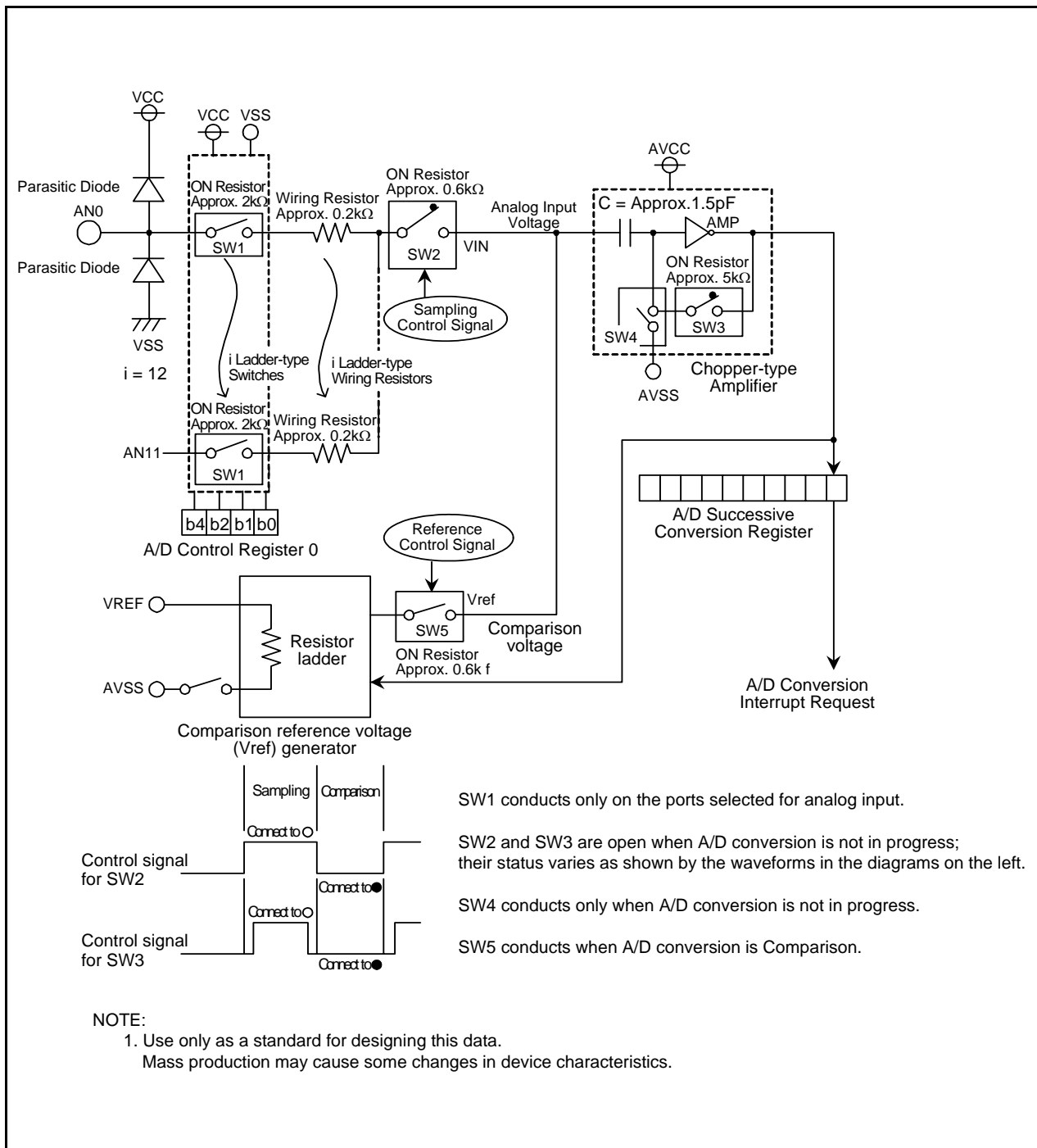


Figure 18.10 Internal Equivalent Circuit of Analog Input

## 18.6 Output Impedance of Sensor Under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor  $C$  shown in Figure 18.11 has to be completed within a specified period of time.  $T$  (sampling time) as the specified time. Let output impedance of sensor equivalent circuit be  $R_0$ , internal resistance of microcomputer be  $R$ , precision (error) of the A/D converter be  $X$ , and the resolution of A/D converter be  $Y$  ( $Y$  is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

$$VC \text{ is generally } VC = VIN \left\{ 1 - e^{-\frac{1}{C(R_0 + R)}t} \right\}$$

$$\text{And when } t = T, VC = VIN - \frac{X}{Y}VIN = VIN \left( 1 - \frac{X}{Y} \right)$$

$$e^{-\frac{1}{C(R_0 + R)}T} = \frac{X}{Y}$$

$$-\frac{1}{C(R_0 + R)}T = \ln \frac{X}{Y}$$

$$\text{Hence, } R_0 = -\frac{T}{C \cdot \ln \frac{X}{Y}} - R$$

Figure 18.11 shows Analog Input Pin and External Sensor Equivalent Circuit. When the difference between  $VIN$  and  $VC$  becomes 0.1LSB, we find impedance  $R_0$  when voltage between pins  $VC$  changes from 0 to  $VIN$  (0.1/1024)  $VIN$  in time  $T$ . (0.1/1024) means that A/D precision drop due to insufficient capacitor charge is held to 0.1LSB at time of A/D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1LSB.

When  $f(XIN) = 10 \text{ MHz}$ ,  $T = 0.25 \mu\text{s}$  in the A/D conversion mode without sample & hold. Output impedance  $R_0$  for sufficiently charging capacitor  $C$  within time  $T$  is determined as follows.

$T = 0.25 \mu\text{s}$ ,  $R = 2.8 \text{ k}\Omega$ ,  $C = 6.0 \text{ pF}$ ,  $X = 0.1$ , and  $Y = 1024$ . Hence,

$$R_0 = -\frac{0.25 \times 10^{-6}}{6.0 \times 10^{-12} \cdot \ln \frac{0.1}{1024}} - 2.8 \times 10^3 \approx 1.7 \times 10^3$$

Thus, the allowable output impedance of the sensor equivalent circuit, making the precision (error) 0.1LSB or less, is approximately 1.7 k $\Omega$  maximum.

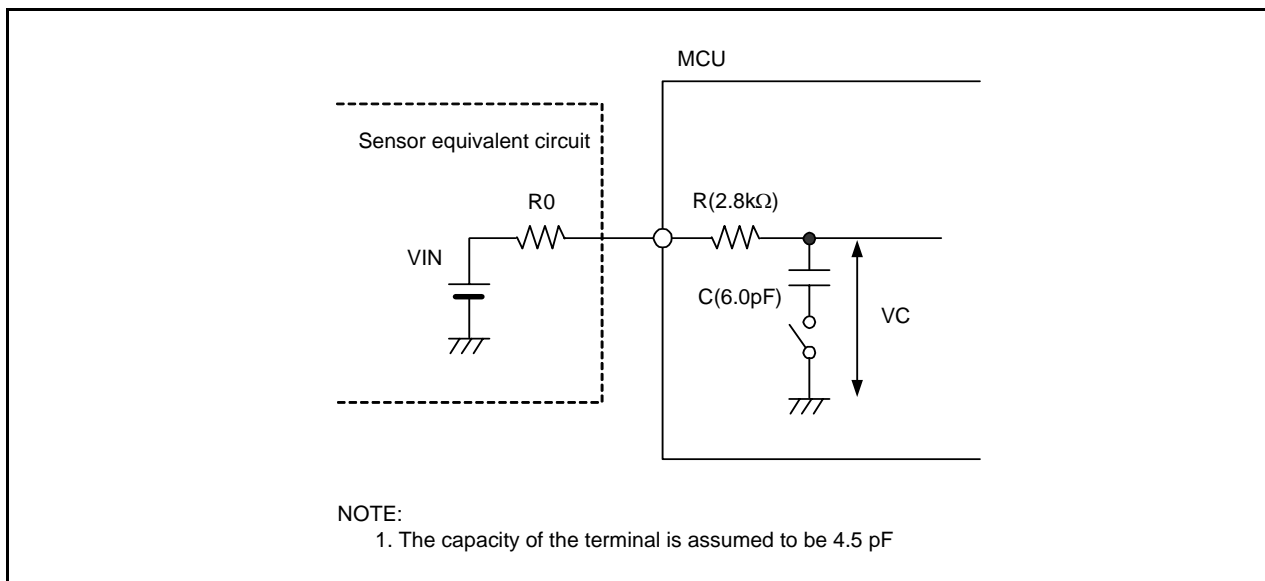


Figure 18.11 Analog Input Pin and External Sensor Equivalent Circuit

## 18.7 Notes on A/D Converter

- Write to each bit (other than bit 6) in the ADCON0 register, each bit in the ADCON1 register, or the SMP bit in the ADCON2 register when the A/D conversion stops (before a trigger occurs).  
When the VCUT bit in the ADCON1 register is changed from 0 (VREF not connected) to 1 (VREF connected), wait for at least 1  $\mu$ s or longer before the A/D conversion starts.
- When changing A/D operating mode, select an analog input pin again.
- When using in one-shot mode. Ensure that the A/D conversion is completed and read the AD register. The IR bit in the ADIC register or the ADST bit in the ADCON0 register can determine whether the A/D conversion is completed.
- When using the repeat mode, select the frequency of the A/D converter operating clock  $\phi$ AD or more for the CPU clock during A/D conversion.  
Do not select the fOCO-F for the  $\phi$ AD.
- If setting the ADST bit in the ADCON0 register to 0 (A/D conversion stops) by a program and the A/D conversion is forcibly terminated during the A/D conversion operation, the conversion result of the A/D converter will be indeterminate. If the ADST bit is set to 0 by a program, do not use the value of AD register.
- Connect 0.1  $\mu$ F capacitor between the P4\_2/VREF pin and AVSS pin.
- Do not enter stop mode during A/D conversion.
- Do not enter wait mode when the CM02 bit in the CM0 register is set to 1 (peripheral function clock stops in wait mode) during A/D conversion.

## 19. Flash Memory

### 19.1 Overview

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

Table 19.1 lists the Flash Memory Performance (see **Table 1.1** and **Table 1.2 Performance** for the items not listed on Table 19.1).

**Table 19.1 Flash Memory Performance**

Item		Specification
Flash Memory Operating Mode		3 modes (CPU rewrite, standard serial I/O, and parallel I/O mode)
Division of Erase Block		See <b>Figure 19.1</b> and <b>Figure 19.2</b>
Program Method		Byte unit
Erase Method		Block erase
Program, Erase Control Method		Program and erase control by software command
Rewrite Control Method		Rewrite control for blocks 0 and 1 by FMR02 bit in FMR0 register
		Rewrite control for block 0 by FMR16 bit and block 1 by FMR16 bit
Number of Commands		5 commands
Programming and erase endurance <sup>(1)</sup>	Blocks 0 and 1 (Program ROM)	R8C/20 Group: 100 times; R8C/21 Group: 1,000 times
	Blocks A and B (Data Flash) <sup>(2)</sup>	10,000 times
ID Code Check Function		Standard serial I/O mode supported
ROM Code Protect		For parallel I/O mode supported

NOTES:

1. Definition of programming and erasure endurance

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

If the programming and erasure endurance is n (n = 100 or 10,000), each block can be erased n times. For example, if 1,024 1-byte writes are performed to different addresses in block A, a 1-Kbyte block, and then the block is erased, the programming/erasure endurance still stands at one. When performing 100 or more rewrites, the actual erasure endurance 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 erasure endurance of each block and limit the number of erase operations to a certain number.

2. Blocks A and B are embedded only in the R8C/21 Group.

**Table 19.2 Flash Memory Rewrite Modes**

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



## 19.2 Memory Map

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

Figure 19.1 shows the Flash Memory Block Diagram for R8C/20 Group. Figure 19.2 shows the Flash Memory Block Diagram for R8C/21 Group.

The user ROM area of R8C/21 Group contains an area which stores a MCU operating program (program ROM) and the 1-Kbyte block A and B (data flash).

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

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

When rewriting the block 2 and block 3 in CPU rewrite mode, set the FMR02 bit in the FMR0 register to 1 (rewrite enables).

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

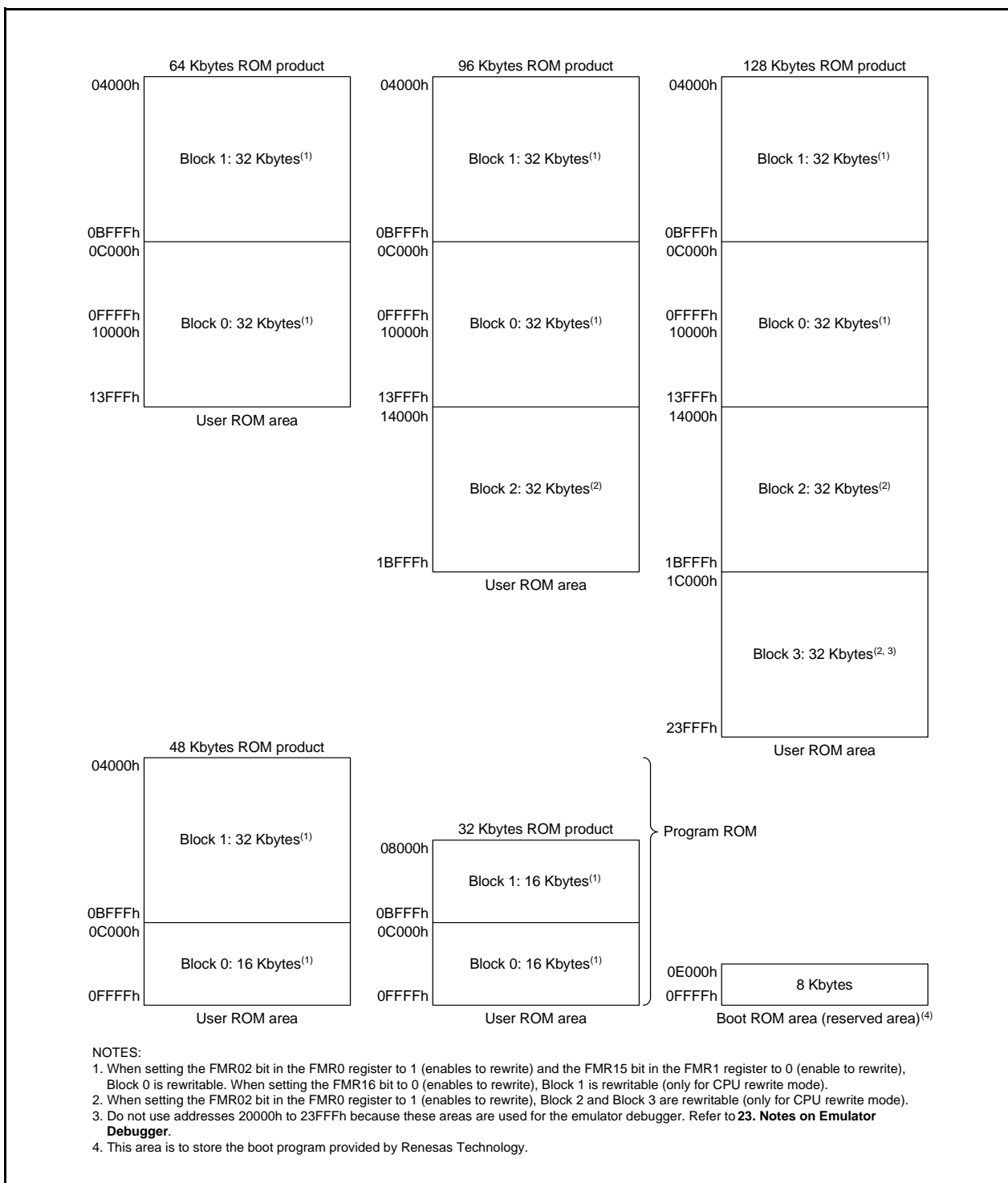


Figure 19.1 Flash Memory Block Diagram for R8C/20 Group

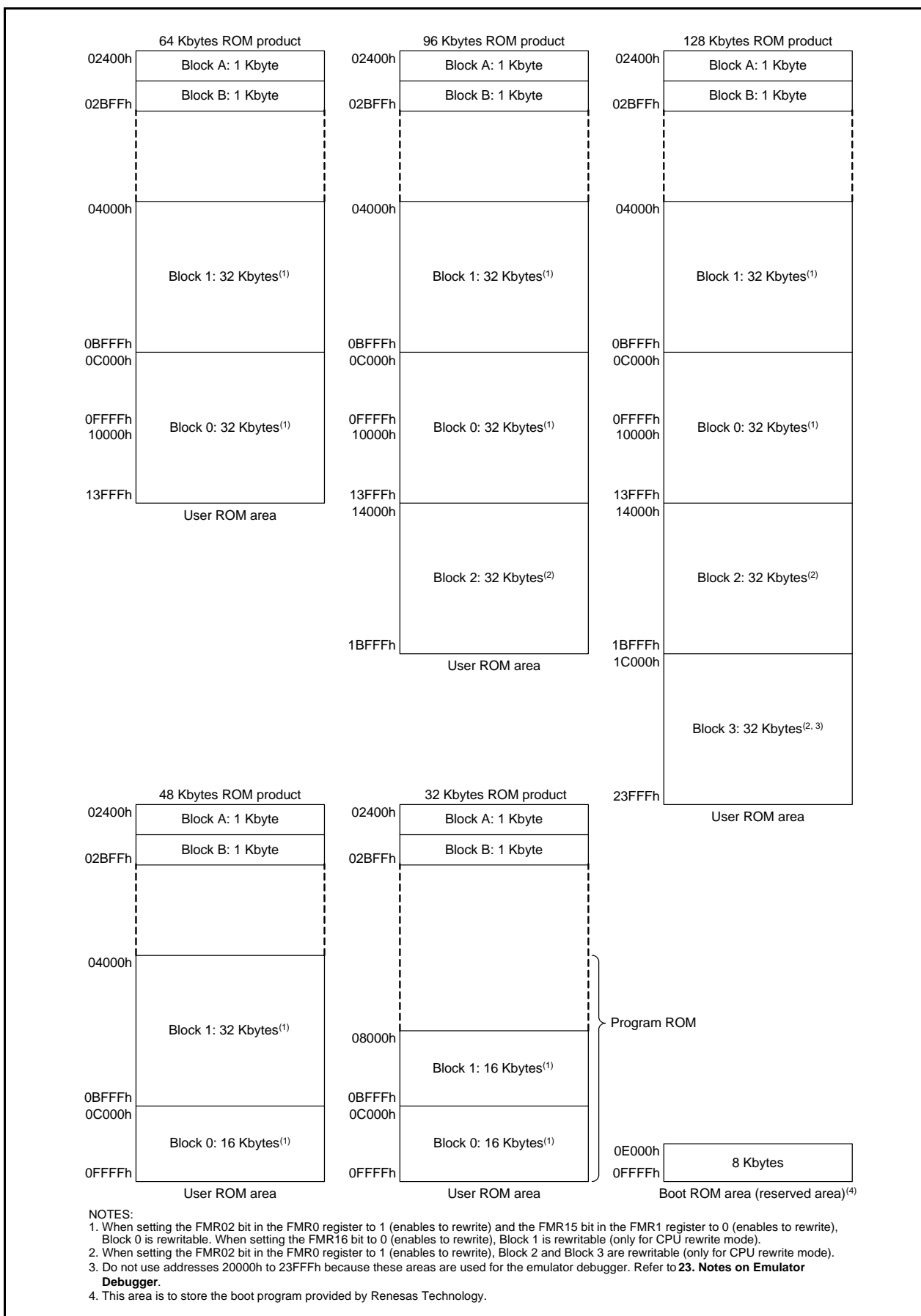


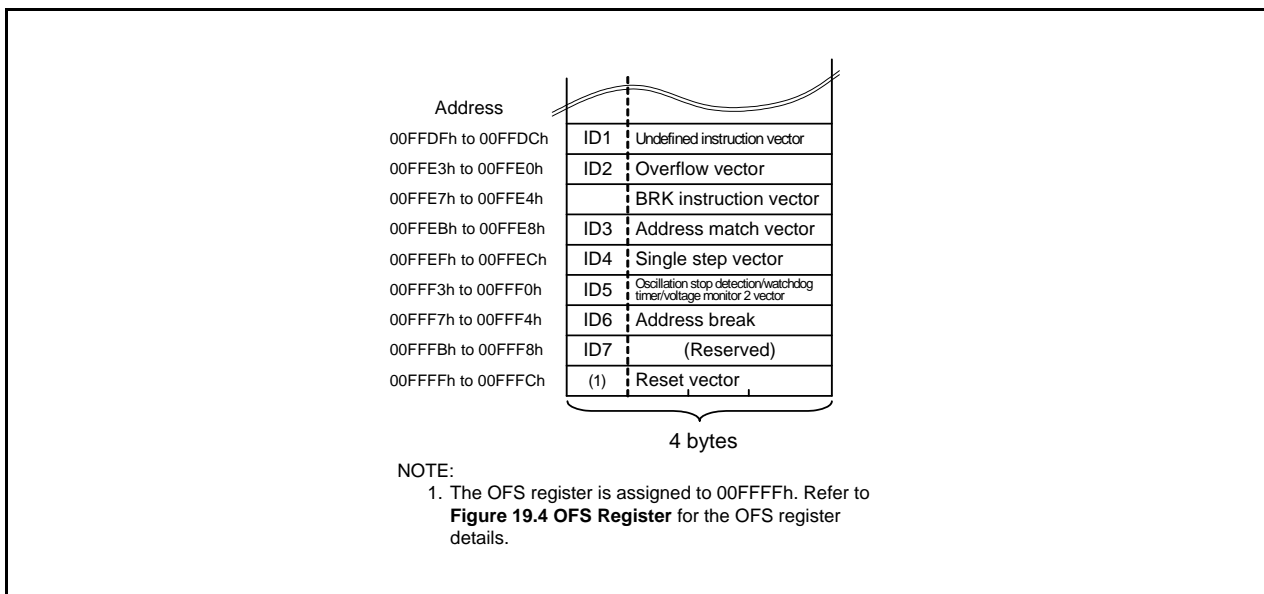
Figure 19.2 Flash Memory Block Diagram for R8C/21 Group

## 19.3 Functions to Prevent Rewriting of Flash Memory

Standard serial I/O mode contains an ID code check function, and the parallel I/O mode contains a ROM code protect function to prevent the flash memory from reading or rewriting easily.

### 19.3.1 ID Code Check Function

Use this function in standard serial I/O mode. Unless the flash memory is blank, the ID codes sent from the programmer and the ID codes written in the flash memory are determined whether they match. If the ID codes do not match, the commands sent from the programmer are not acknowledged. The ID code consists of 8-bit data, the areas of which, beginning with the first byte, are 00FFDFh, 00FFE3h, 00FFEBh, 00FFEFh, 00FFF3h, 00FFF7h, and 00FFFBh. Write a program in which the ID codes are set at these addresses and write them in the flash memory.



**Figure 19.3** Address for Stored ID Code

### 19.3.2 ROM Code Protect Function

The ROM code protect function disables to read and change the internal flash memory by the OFS register in parallel I/O mode.

Figure 19.4 shows the OFS Register.

The ROM code protect function is enabled by writing 0 to the ROMCP1 bit and 1 to the ROMCR bit and disables to read and change the internal flash memory.

Once the ROM code protect is enabled, the content in the internal flash memory cannot be rewritten in parallel I/O mode. When the ROM code protect is disabled, erase the block including the OFS register with CPU rewrite mode or standard serial I/O mode.

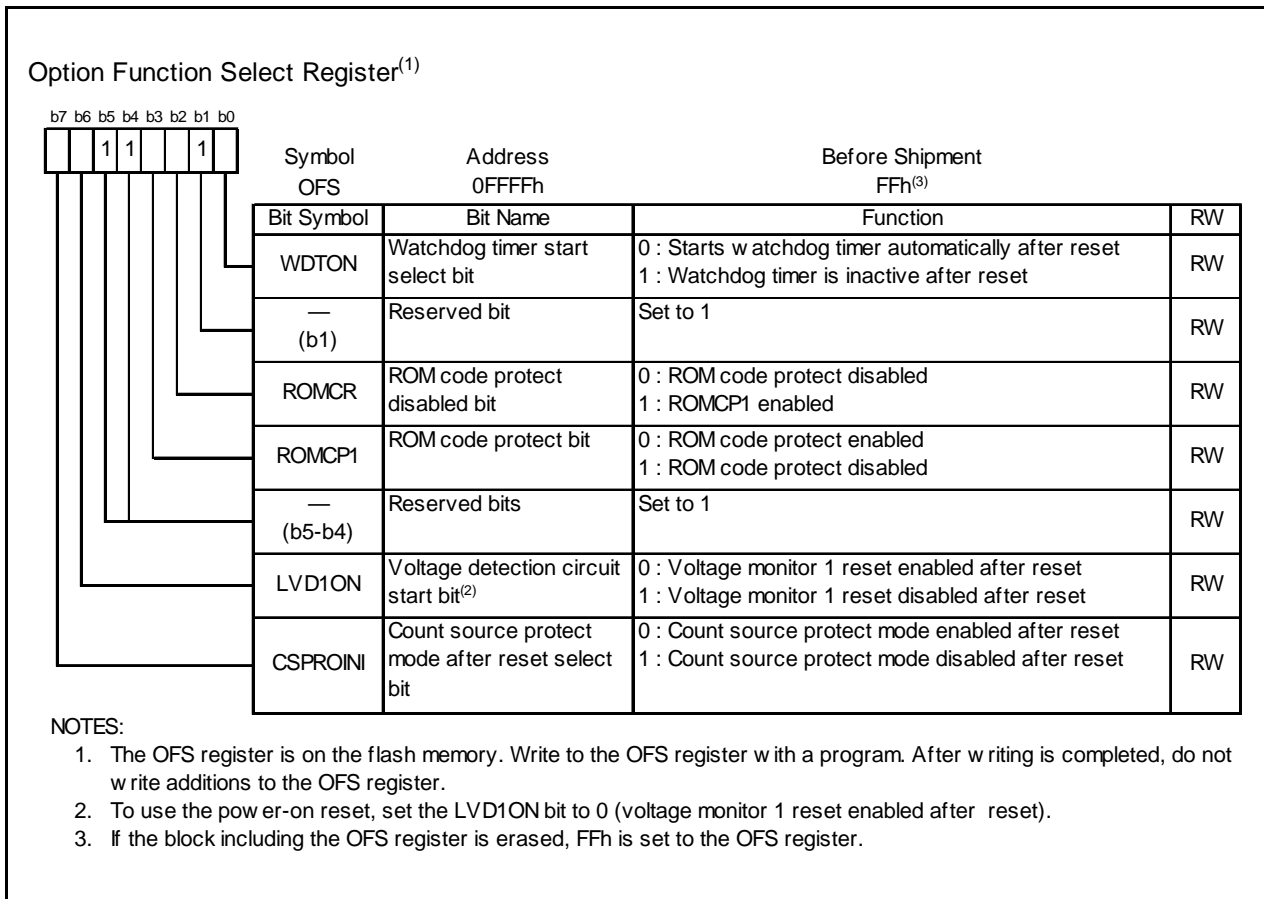


Figure 19.4 OFS Register

## 19.4 CPU Rewrite Mode

In CPU rewrite mode, 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 such as a ROM programmer. Execute the program and block erase commands only to each block in user ROM area.

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

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

CPU rewrite mode contains erase write 0 mode (EW0 mode) and erase write 1 mode (EW1 mode).

Table 19.3 lists the Differences between EW0 Mode and EW1 Mode.

**Table 19.3 Differences between EW0 Mode and EW1 Mode**

Item	EW0 Mode	EW1 Mode
Operating Mode	Single chip mode	Single chip mode
Areas in which a Rewrite Control Program Can be Located	User ROM area	User ROM area
Areas in which a Rewrite Control Program can be Executed	Necessary to transfer to any areas other than the flash memory (e.g., RAM) before executing	Executing directly in user ROM or RAM area possible
Areas which can be Rewritten	User ROM area	User ROM area However, other than the blocks which contain a rewrite control program <sup>(1)</sup>
Software Command Restriction	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 the FMR00, FMR06, and FMR07 bits in the FMR0 register by a program</li> <li>Execute the read status register command and read the SR7, SR5, and SR4 bits in the status register.</li> </ul>	Read the FMR00, FMR06, and FMR07 bits in the FMR0 register by a program
Conditions for Transition to Erase-suspend	Set the FMR40 and FMR41 bits in the FMR4 register to 1 by a program.	The FMR40 bit in the FMR4 register is set to 1 and the interrupt request of the enabled maskable interrupt is generated
Conditions for Transitions to Program-suspend	Set the FMR40 and FMR42 bits in the FMR4 register to 1 by a program.	The FMR40 bit in the FMR4 register is set to 1 and the interrupt request of the enabled maskable interrupt is generated
CPU Clock	5 MHz or below	No restriction to the following (clock frequency to be used)

**NOTE:**

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

### 19.4.1 EW0 Mode

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

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

When entering an erase-suspend, set the FMR40 bit to 1 (enables suspend) and the FMR41 bit to 1 (requests erase-suspend). Wait for  $td(SR-SUS)$  and ensure that the FMR46 bit is set to 1 (enables reading) before accessing the user ROM area. The auto-erase operation restarts by setting the FMR41 bit to 0 (erase restarts).

When entering a program-suspend during the auto-program, set the FMR40 bit to 1 (enables suspend) and the FMR42 bit to 1 (requests program-suspend). Wait for  $td(SR-SUS)$  and ensure that the FMR46 bit is set to 1 (enables reading) before accessing the user ROM area. The auto-program operation restarts by setting the FMR42 bit to 0 (program restarts).

### 19.4.2 EW1 Mode

The MCU enters 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 determine status when program and erase operation complete. Do not execute commands of the read status register in EW1 mode.

To enable the erase-suspend function during the auto-erase, execute the block erase command after setting the FMR40 bit to 1 (enables suspend). The interrupt to enter an erase-suspend should be in interrupt enabled status. After passing  $td(SR-SUS)$  since the block erase command is executed, an interrupt request is acknowledged.

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

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

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

Figure 19.5 shows the FMR0 Register, Figure 19.6 shows the FMR1 Register and Figure 19.7 shows the FMR4 Register.

#### 19.4.2.1 FMR00 Bit

This bit indicates the operating status of the flash memory. The bit is 0 during programming, erasing (including suspend periods), or erase-suspend mode; otherwise, the bit is 1.

#### 19.4.2.2 FMR01 Bit

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

#### 19.4.2.3 FMR02 Bit

The block0, block1, block2 and block3 do not accept the program and block erase commands if the FMR02 bit is set to 0 (rewrite disabled).

The block0 and block1 are controlled rewriting in the FMR15 and FMR16 bits if the FMR02 bit is set to 1 (rewrite enabled).

#### 19.4.2.4 FMSTP Bit

This bit is provided for initializing the flash memory control circuits, as well as for reducing the amount of current consumed in the flash memory. The flash memory is disabled against access 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))
- When entering high-speed on-chip oscillator mode, low-speed on-chip oscillator mode (XIN clock stop)

Figure 19.11 shows a Process to Reduce Power Consumption in High-Speed On-Chip Oscillator Mode, Low-Speed On-Chip Oscillator Mode (XIN Clock Stops) and Low-Speed Clock Mode (XIN Clock Stops). 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.

#### 19.4.2.5 FMR06 Bit

This is a read-only bit indicating the status of auto program operation. The bit is set to 1 when a program error occurs; otherwise, it is cleared to 0. For details, refer to the description of the **19.4.5 Full Status Check**.

#### 19.4.2.6 FMR07 Bit

This is a read-only bit indicating the status of auto erase operation. The bit is set to 1 when an erase error occurs; otherwise, it is set to 0. Refer to **19.4.5 Full Status Check** for the details.

#### 19.4.2.7 FMR11 Bit

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

#### 19.4.2.8 FMR15 Bit

When the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit is set to 0 (rewrite enabled), the block0 accepts the program command and block erase command.

#### 19.4.2.9 FMR16 Bit

When the FMR02 bit is set to 1 (rewrite enabled) and the FMR16 bit is set to 0 (rewrite enabled), the block1 accepts the program command and block erase command.

#### 19.4.2.10 FMR40 Bit

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



#### 19.4.2.11 FMR41 Bit

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

Set the FMR41 bit to 0 (erase restart) when the auto-erase operation restarts.

#### 19.4.2.12 FMR42 Bit

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

Set the FMR42 bit to 0 (program restarts) when the auto-program operation restarts.

#### 19.4.2.13 FMR43 Bit

When the auto-erase operation starts, the FMR43 bit is set to 1 (during erase execution). The FMR43 bit remains 1 (during erase execution) during erase-suspend operation.

When the auto-erase operation ends, the FMR43 bit is set to 0 (erase not executed).

#### 19.4.2.14 FMR44 Bit

When the auto-program starts, the FMR44 bit is set to 1 (during program execution). The FMR44 bit remains 1 (during program execution) during program-suspend operation.

When the auto-program operation ends, the FMR44 bit is set to 0 (program not executed).

#### 19.4.2.15 FMR46 Bit

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

#### 19.4.2.16 FMR47 Bit

Power consumption when reading the flash memory can be reduced by setting the FMR47 bit to 1 (enabled) in low-speed on-chip oscillator mode (XIN clock stops).

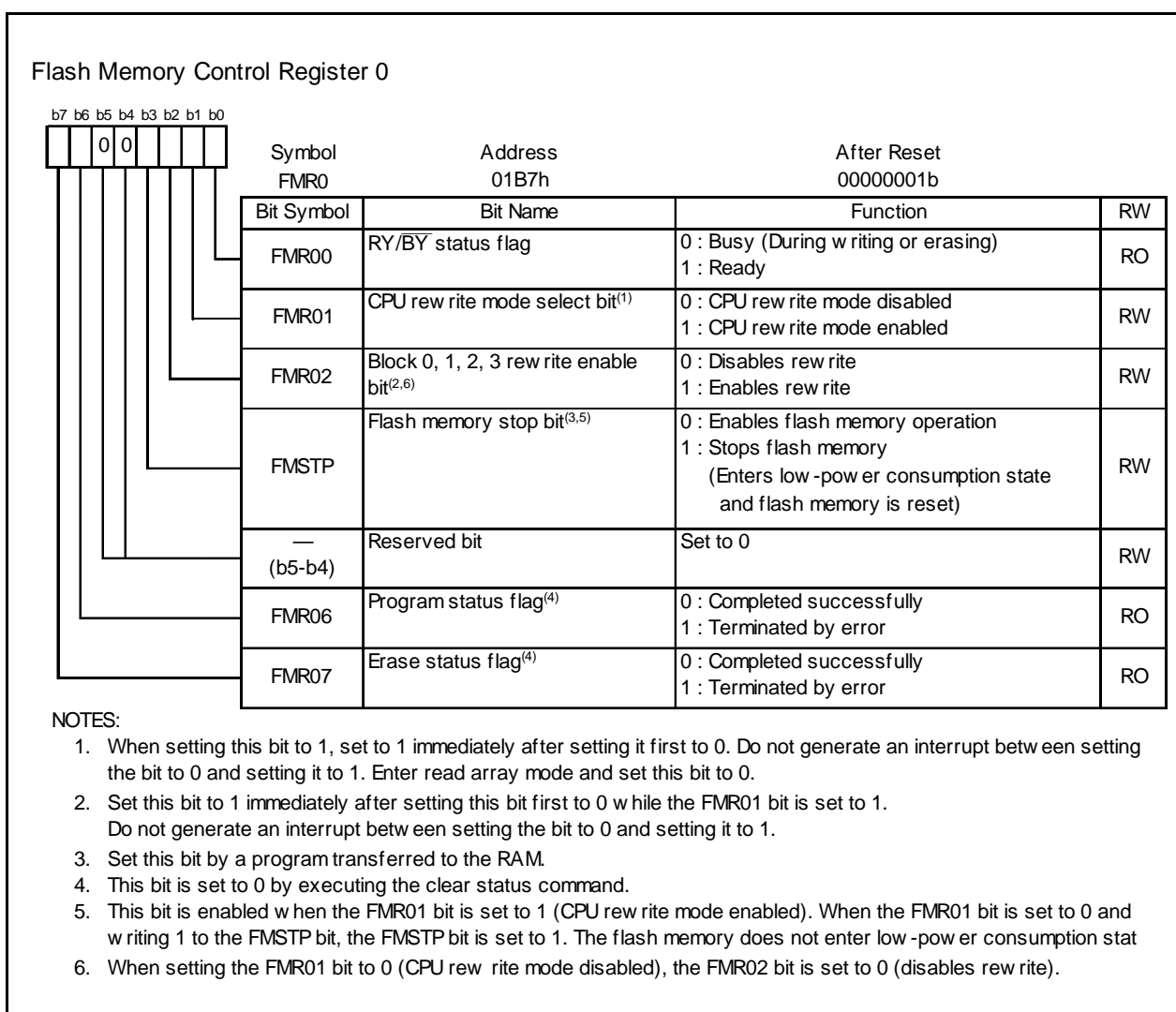


Figure 19.5 FMR0 Register

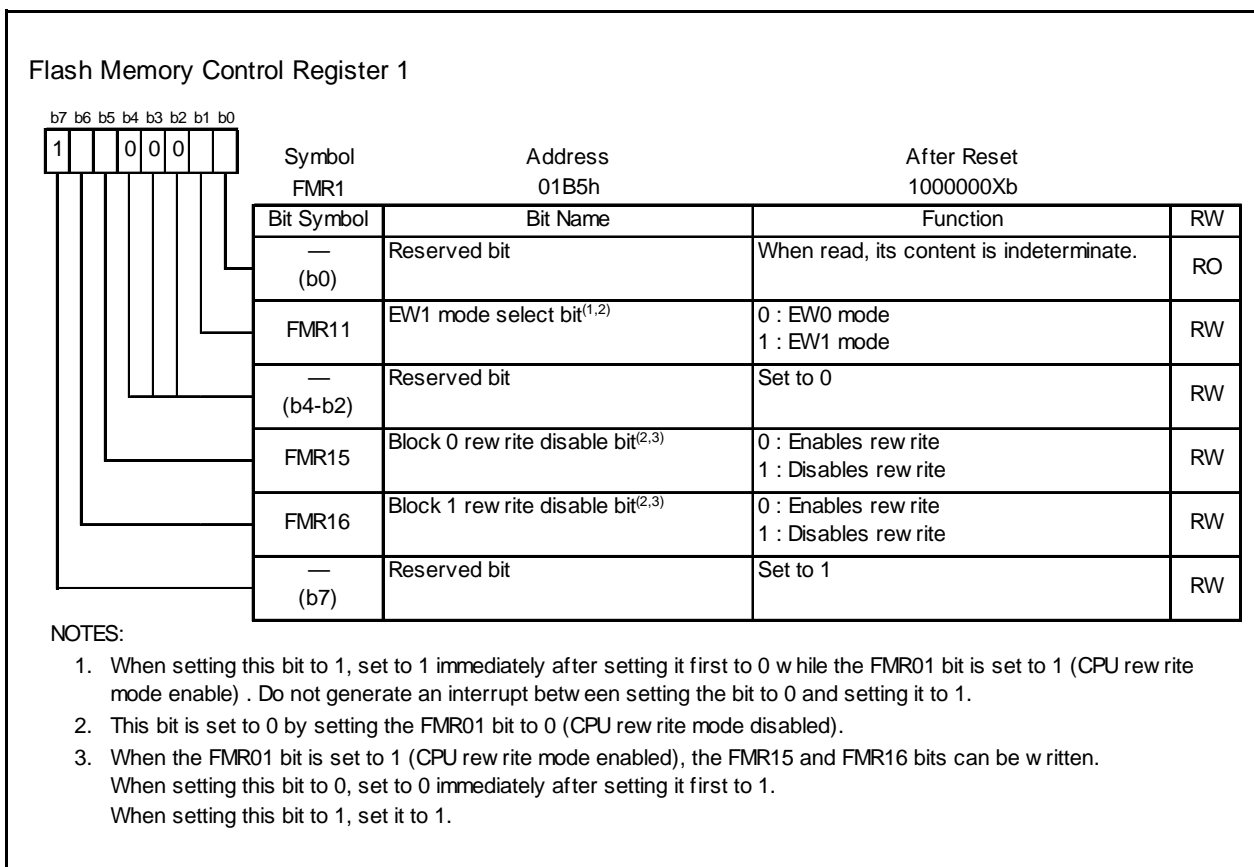


Figure 19.6 FMR1 Register

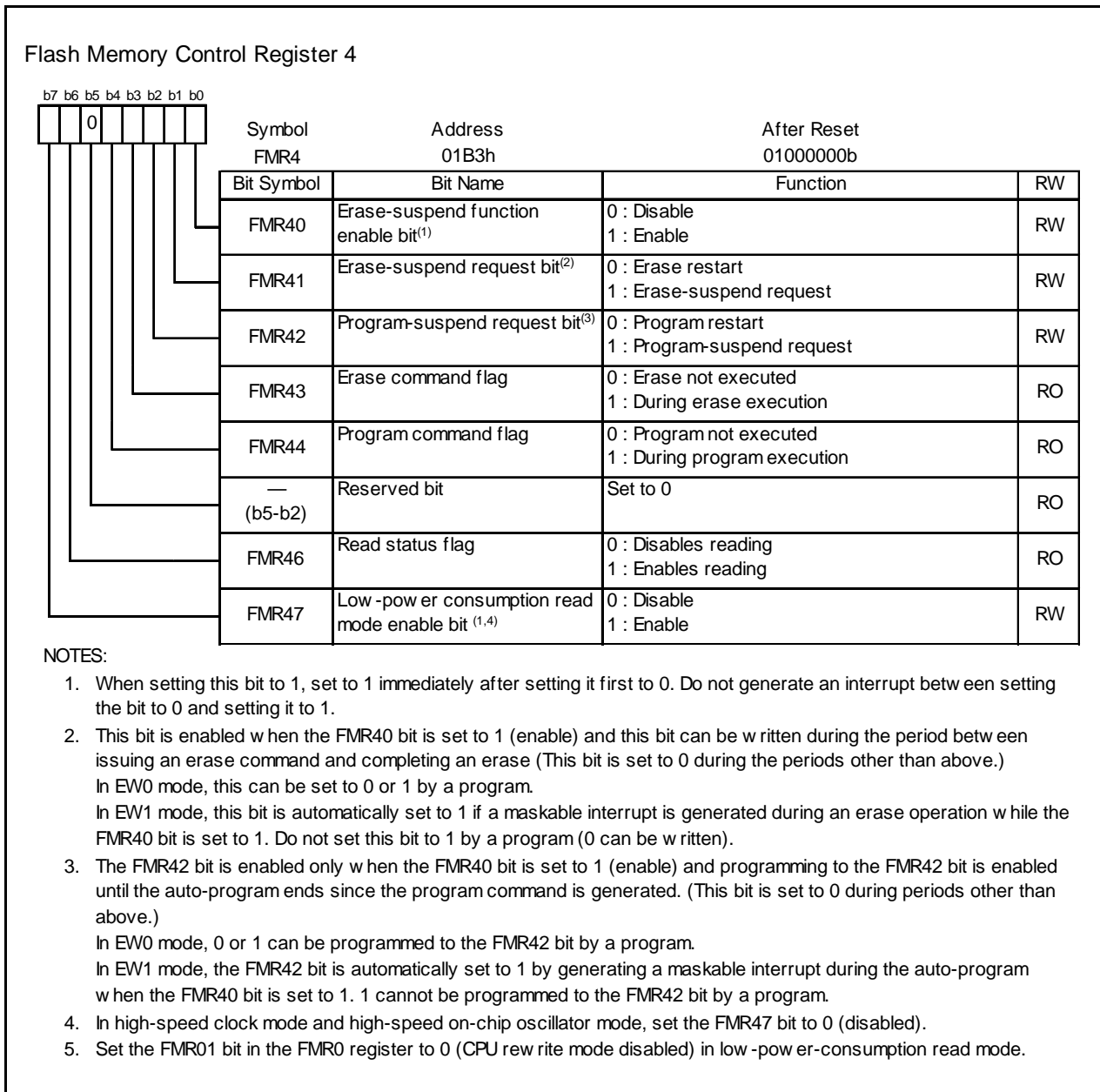


Figure 19.7 FMR4 Register

Figure 19.8 shows the Timing of Suspend Operation.

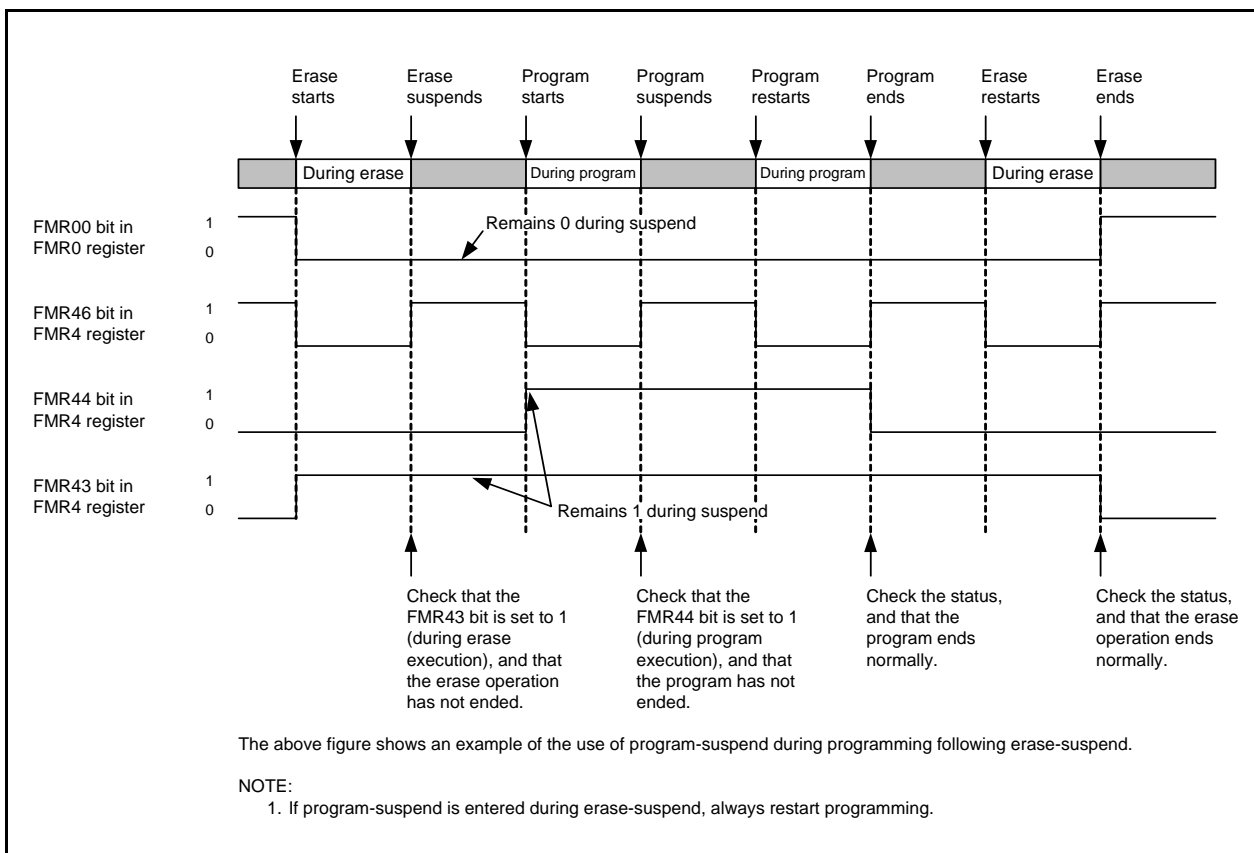
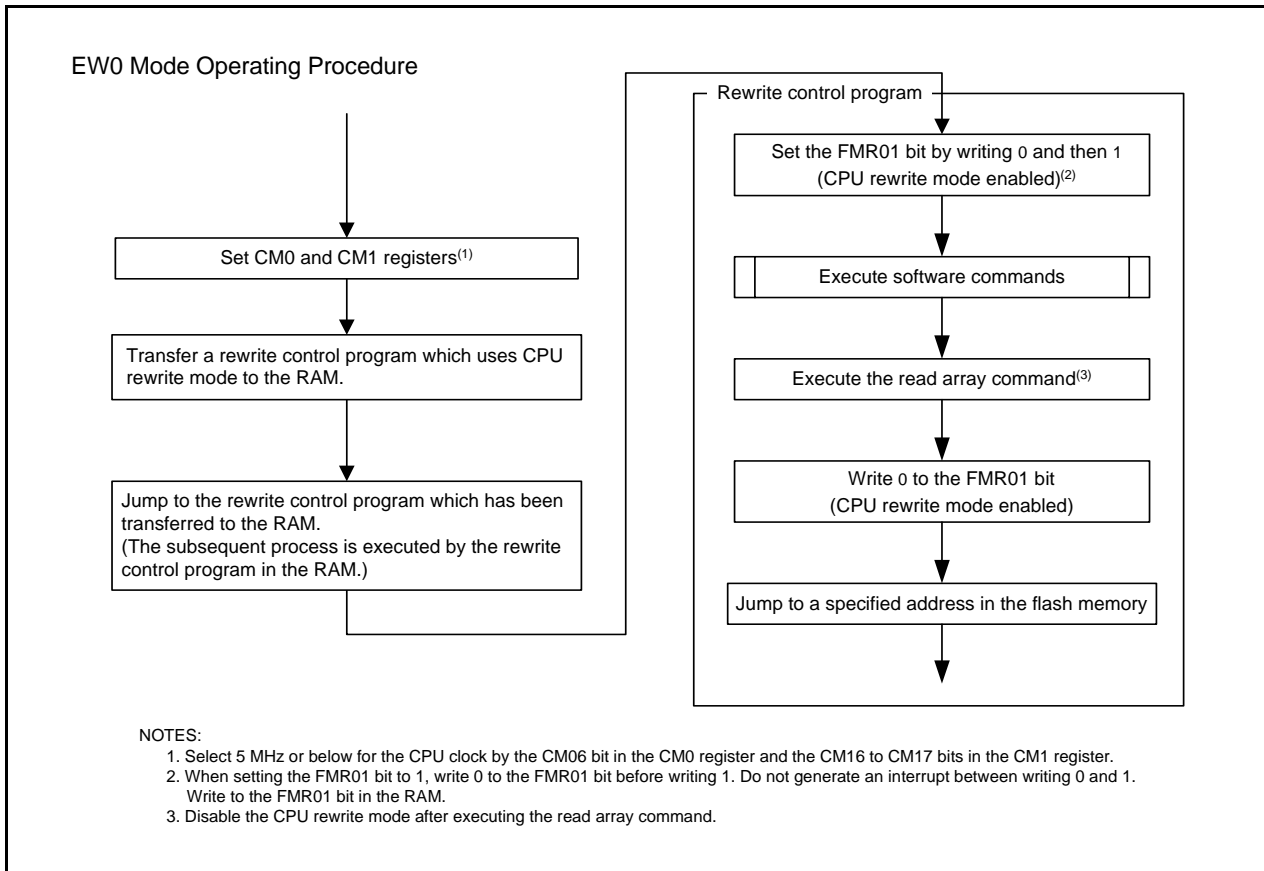
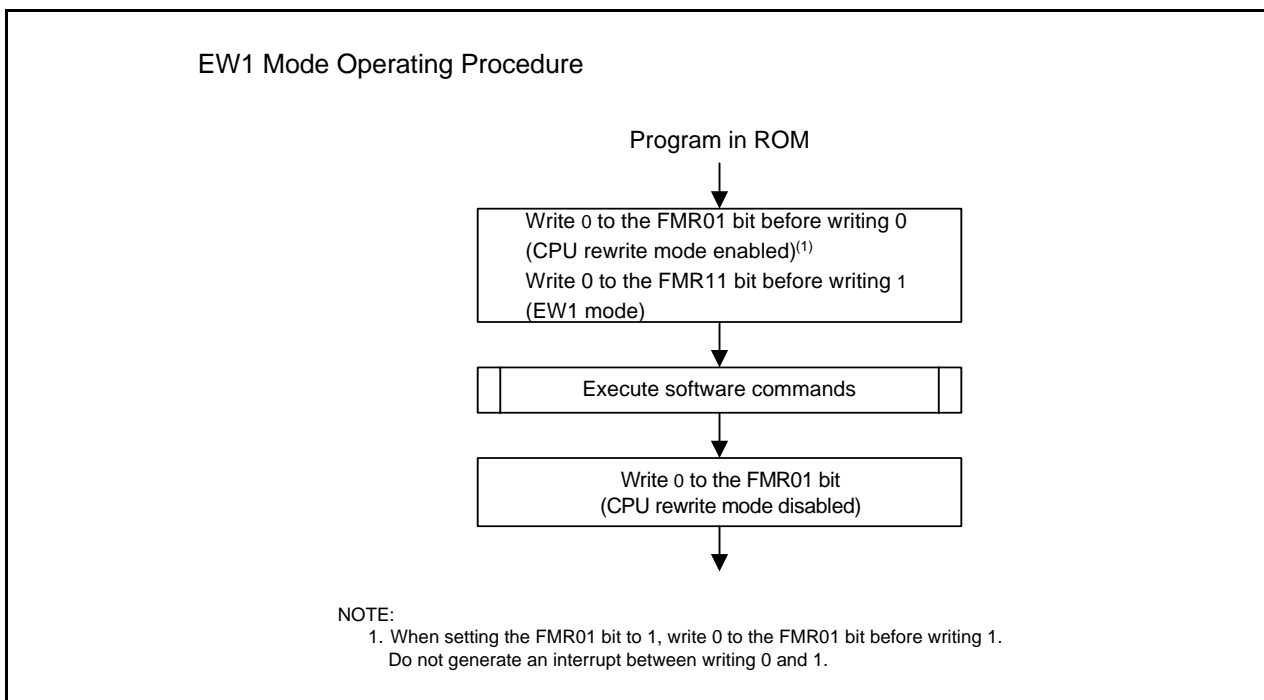


Figure 19.8 Timing of Suspend Operation

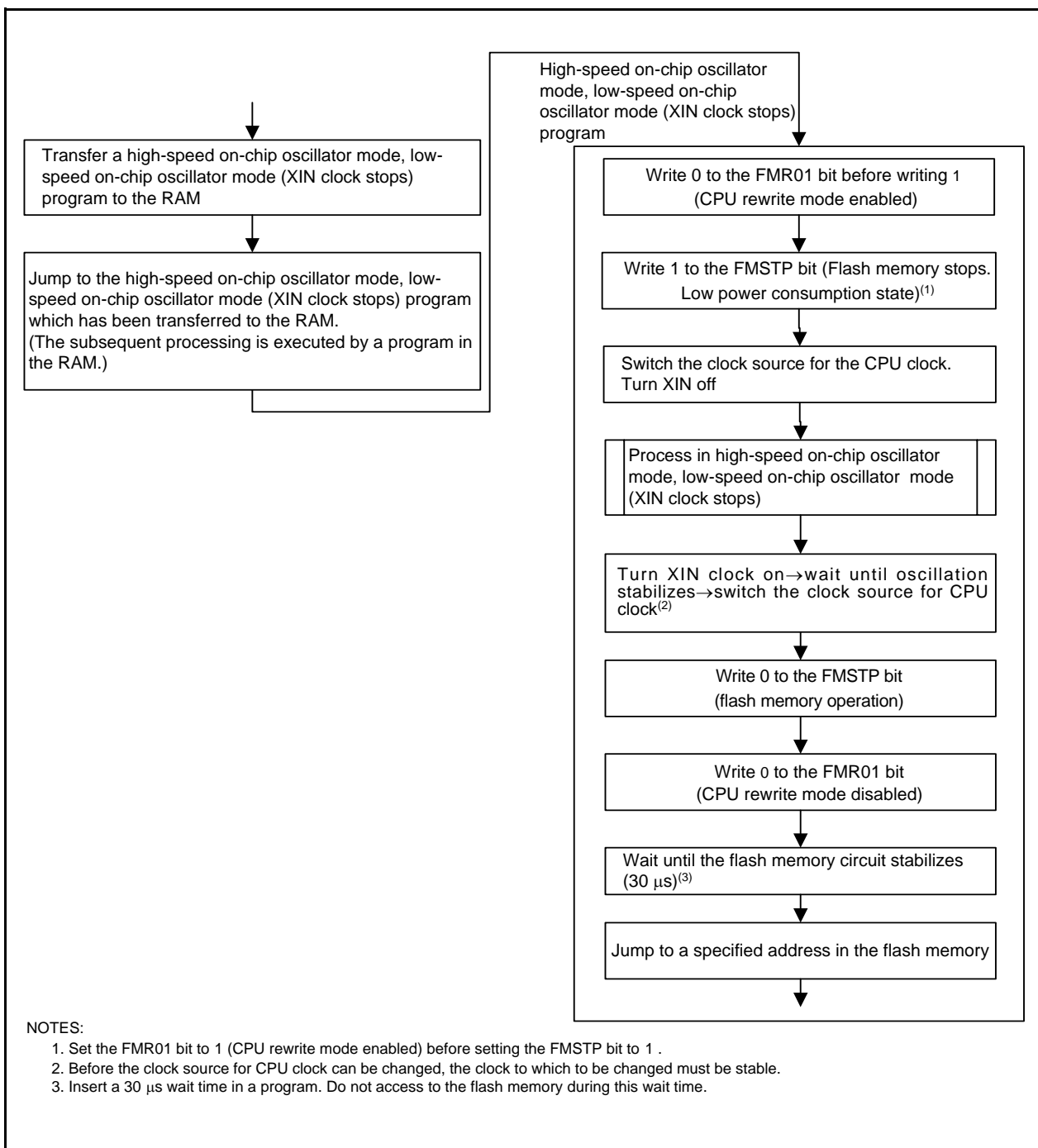
Figure 19.9 shows the How to Set and Exit EW0 Mode. Figure 19.10 shows the How to Set and Exit EW1 Mode.



**Figure 19.9 How to Set and Exit EW0 Mode**



**Figure 19.10 How to Set and Exit EW1 Mode**



**Figure 19.11 Process to Reduce Power Consumption in High-Speed On-Chip Oscillator Mode, Low-Speed On-Chip Oscillator Mode (XIN Clock Stops) and Low-Speed Clock Mode (XIN Clock Stops)**

### 19.4.3 Software Commands

Software commands are described below. Read or write commands and data from or to in 8-bit units.

**Table 19.4 Software Commands**

Command	First Bus Cycle			Second Bus Cycle		
	Mode	Address	Data (D7 to D0)	Mode	Address	Data (D7 to D0)
Read Array	Write	x	FFh			
Read Status Register	Write	x	70h	Read	x	SRD
Clear Status Register	Write	x	50h			
Program	Write	WA	40h	Write	WA	WD
Block Erase	Write	x	20h	Write	BA	D0h

SRD: Status register data (D7 to D0)

WA: Write address (Ensure the address specified in the first bus cycle is the same address as the write address specified in the second bus cycle.)

WD: Write data (8 bits)

BA: Given block address

x: Any specified address in the user ROM area

#### 19.4.3.1 Read Array Command

The read array command reads the flash memory.

The MCU enters read array mode by writing FFh in the first bus cycle. If entering the read address after the following bus cycles, the content of the specified address can be read in 8-bit units.

Since the MCU remains in read array mode until another command is written, the contents of multiple addresses can be read continuously.

In addition, the MCU enters read array mode after a reset.

#### 19.4.3.2 Read Status Register Command

The read status register command reads the status register.

If writing 70h in the first bus cycle, the status register can be read in the second bus cycle. (Refer to **19.4.4 Status Registers**) When reading the status register, specify an address in the user ROM area.

Do not execute this command in EW1 mode.

The MCU remains in read status register mode until the next read array command is written.

#### 19.4.3.3 Clear Status Register Command

The clear status register command sets the status register to 0.

If writing 50h in the first bus cycle, the FMR06 to FMR07 bits in the FMR0 register and SR4 to SR5 in the status register will be set to 0.



### 19.4.3.4 Program Command

The program command writes data to the flash memory in 1-byte units.

By writing 40h in the first bus cycle and data in the second bus cycle to the write address, and an auto program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same address as the write address specified in the second bus cycle.

The FMR00 bit in the FMR0 register can determine whether auto programming has completed.

When suspend function disabled, the FMR00 bit is set to 0 during auto-programming and set to 1 when auto-programming completes.

When suspend function enabled, the FMR44 bit is set to 1 during auto-programming and set to 0 when auto-programming completes.

The FMR06 bit in the FMR0 register can determine the result of auto programming after it has been finished. (Refer to **19.4.5 Full Status Check**)

When the FMR02 bit in the FMR0 register is set to 0 (disable rewriting), program commands targeting block 0 to 3 are not acknowledged. When the FMR02 bit is set to 1 (rewrite enables) and the FMR15 bit in the FMR1 register is set to 1 (disable rewriting), program commands targeting block 0 are not acknowledged. When the FMR16 bit is set to 1 (disable rewriting), program commands targeting block 1 are not acknowledged.

Figure 19.12 shows the Program Command (When Suspend Function Disabled). Figure 19.14 shows the Program Command (When Suspend Function Enabled).

In EW1 mode, do not execute this command on any address at which the rewrite control program is allocated.

In EW0 mode, the MCU enters read status register mode at the same time auto programming starts and the status register can be read. The status register bit 7 (SR7) is set to 0 at the same time auto programming starts and set back to 1 when auto programming completes. In this case, the MCU remains in read status register mode until a read array command is written next. Reading the status register can determine the result of auto programming after auto programming has completed.

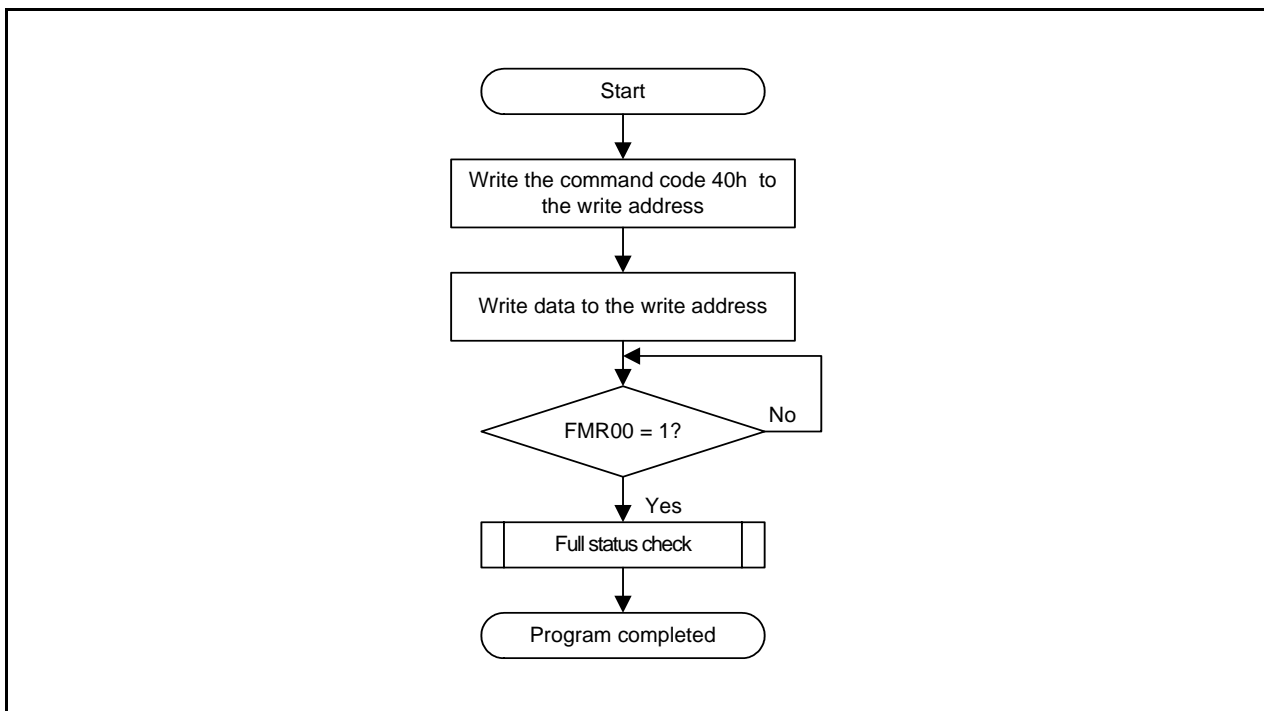


Figure 19.12 Program Command (When Suspend Function Disabled)

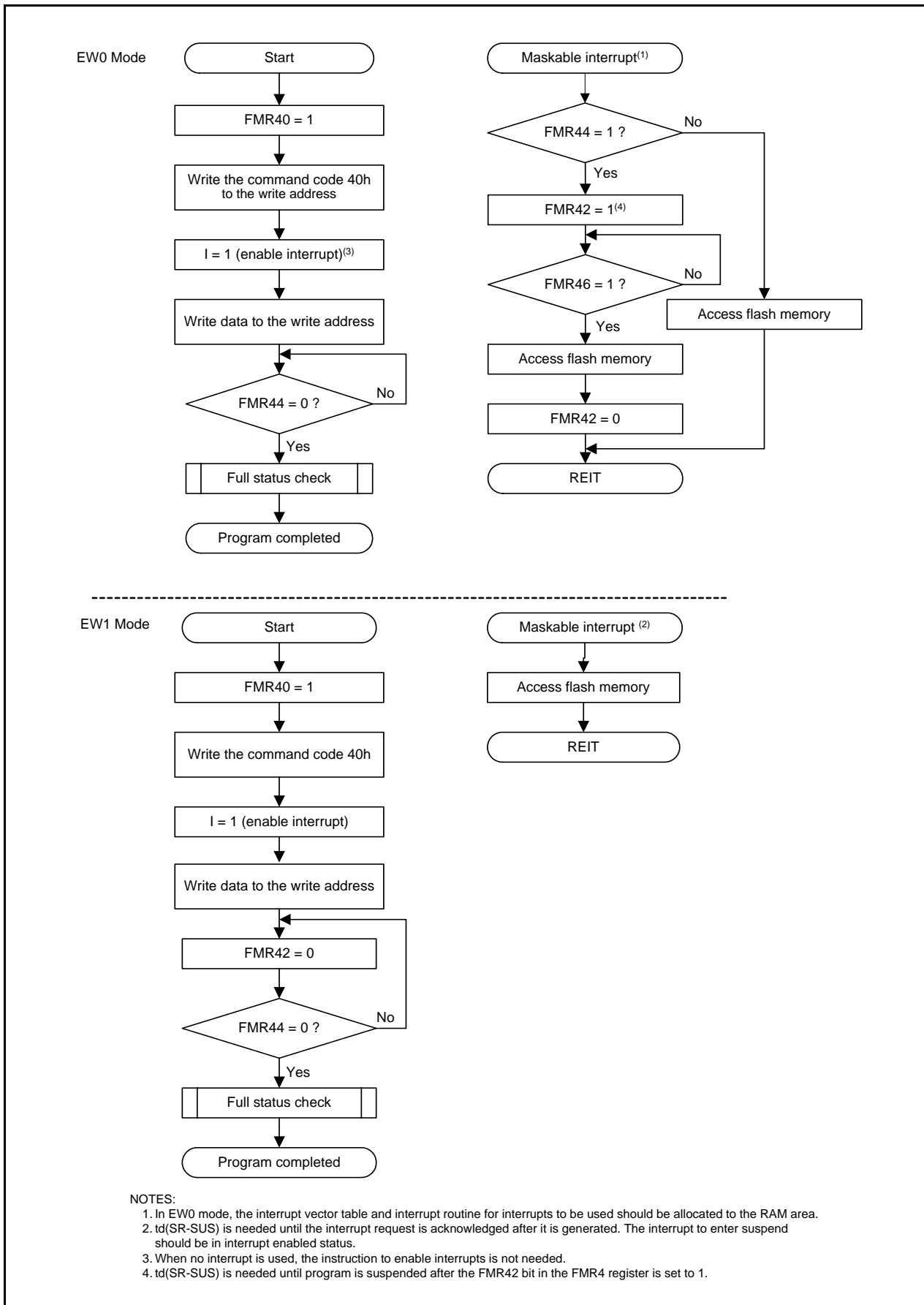


Figure 19.13 Program Command (When Suspend Function Enabled)

### 19.4.3.5 Block Erase

If writing 20h in the first bus cycle and D0h to the given address of a block in the second bus cycle, and an auto erase operation (erase and verify) will start.

The FMR00 bit in the FMR0 register can determine whether auto erasing has completed.

The FMR00 bit is set to 0 during auto erasing and set to 1 when auto erasing completes.

The FMR07 bit in the FMR0 register can determine the result of auto erasing after auto erasing has completed. (Refer to **19.4.5 Full Status Check**)

When the FMR02 bit in the FMR0 register is set to 0 (disable rewriting) or the FMR02 bit is set to 1 (rewrite enables) and the FMR15 bit in the FMR1 register is set to 1 (disable rewriting), the block erase command on block 0 is not acknowledged. When the FMR16 bit is set to 1 (disable rewriting), the block erase command on block 1 is not acknowledged.

Do not use the block erase command during program-suspend.

Figure 19.14 shows the Block Erase Command (When Erase-Suspend Function Disabled). Figure 19.15 shows the Block Erase Command (When Erase-Suspend Function Enabled).

In EW1 mode, do not execute this command on any address at which the rewrite control program is allocated.

In EW0 mode, the MCU enters read status register mode at the same time auto erasing starts and the status register can be read. The status register bit 7 (SR7) is set to 0 at the same time auto erasing starts and set back to 1 when auto erasing completes. In this case, the MCU remains in read status register mode until the read array command is written next.

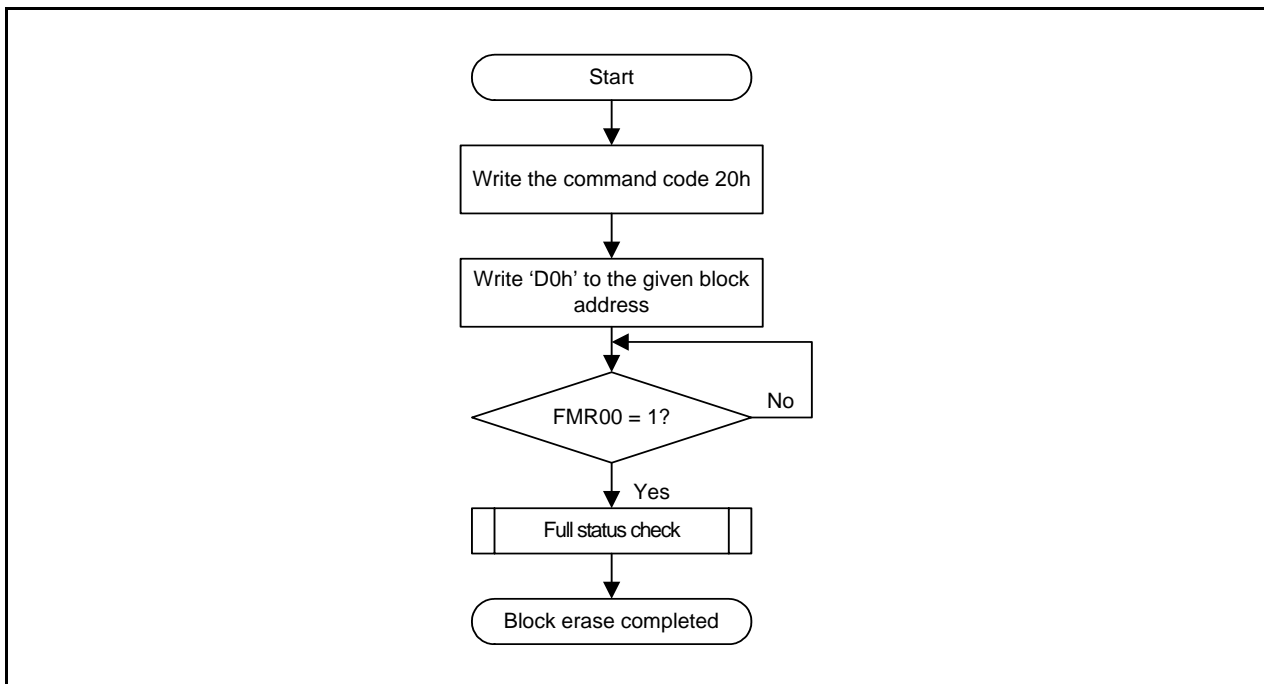


Figure 19.14 Block Erase Command (When Erase-Suspend Function Disabled)

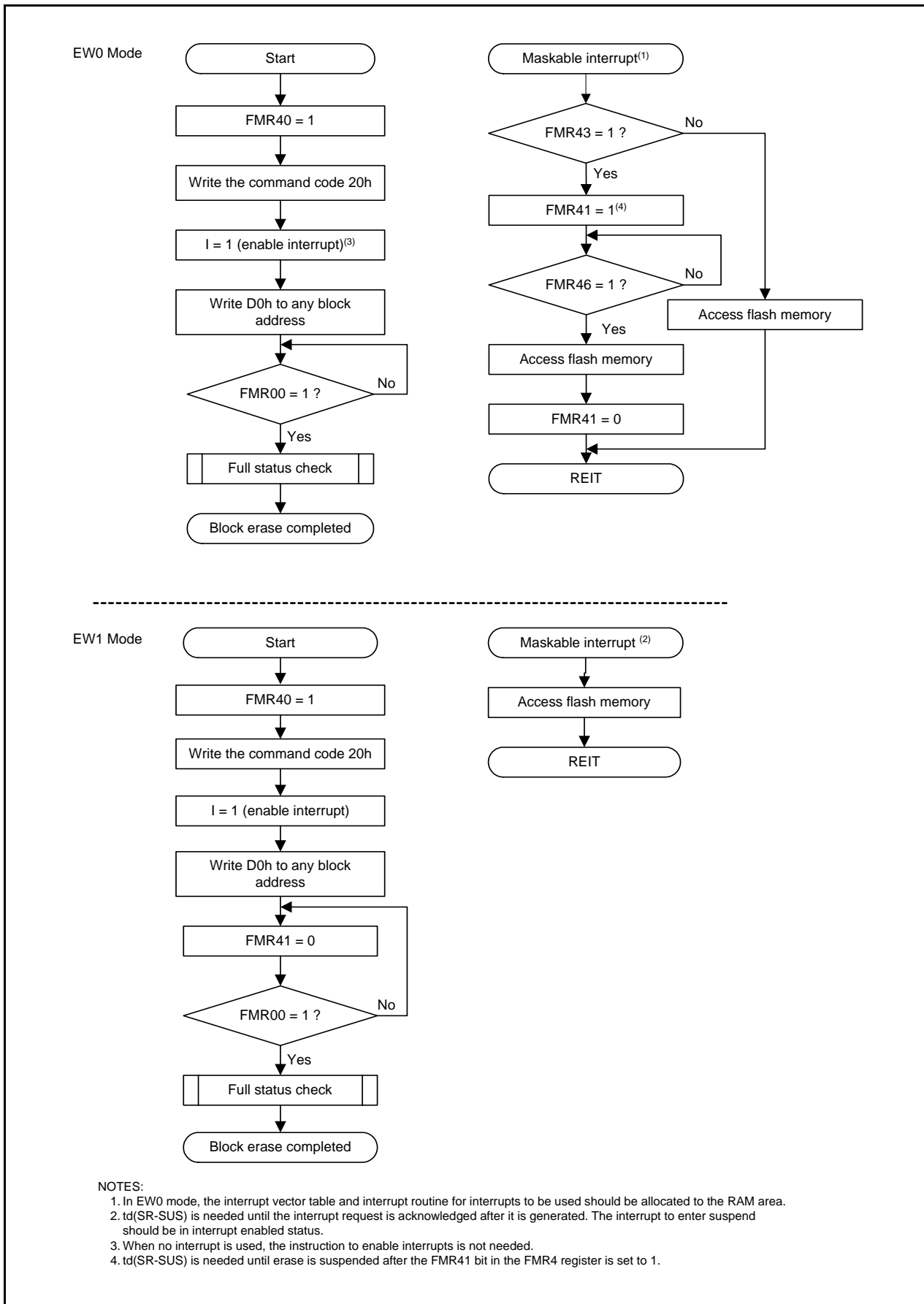


Figure 19.15 Block Erase Command (When Erase-Suspend Function Enabled)

### 19.4.4 Status Registers

The status register indicates the operating status of the flash memory and whether an erasing or programming operation completes normally or in error. Status of the status register can be read by the FMR00, FMR06, and FMR07 bits in the FMR0 register.

Table 19.5 lists the Status Register Bits.

In EW0 mode, the status register can be read in the following cases:

- When a given address in the user ROM area is read after writing the read status register command
- When a given address in the user ROM area is read after executing the program or block erase command but before executing the read array command.

#### 19.4.4.1 Sequencer Status (Bits SR7 and FMR00)

The sequencer status indicates operating status of the flash memory. SR7 = 0 (busy) during auto programming and auto erasing, and is set to 1 (ready) at the same time the operation completes.

#### 19.4.4.2 Erase Status (Bits SR5 and FMR07)

Refer to 19.4.5 Full Status Check.

#### 19.4.4.3 Program Status (Bits SR4 and FMR06)

Refer to 19.4.5 Full Status Check.

**Table 19.5 Status Register Bits**

Status Register Bit	FMR0 Register Bit	Status Name	Description		Value after Reset
			0	1	
SR0 (D0)	–	Reserved	–	–	–
SR1 (D1)	–	Reserved	–	–	–
SR2 (D2)	–	Reserved	–	–	–
SR3 (D3)	–	Reserved	–	–	–
SR4 (D4)	FMR06	Program status	Completed normally	Error	0
SR5 (D5)	FMR07	Erase status	Completed normally	Error	0
SR6 (D6)	–	Reserved	–	–	–
SR7 (D7)	FMR00	Sequencer status	Busy	Ready	1

- D0 to D7: Indicates the data bus which is read when the read status register command is executed.
- The FMR07 (SR5) to FMR06 bits (SR4) are set to 0 by executing the clear status register command.
- When the FMR07 bit (SR5) or FMR06 bit (SR4) is set to 1, the program and block erase command cannot be accepted.

### 19.4.5 Full Status Check

When an error occurs, the FMR06 to FMR07 bits in the FMR0 register are set to 1, indicating occurrence of each specific error. Therefore, checking these status bits (full status check) can determine the executed result.

Table 19.6 lists the Errors and FMR0 Register Status. Figure 19.16 shows the Full Status Check and Handling Procedure for Individual Errors.

**Table 19.6 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 any command is not written correctly</li> <li>• When invalid data other than those that can be written in the second bus cycle of the block erase command is written (i.e., other than D0h or FFh)<sup>(1)</sup></li> <li>• When executing the program command or block erase command while rewriting is disabled using the FMR02 bit in the FMR0 register, the FMR15 or FMR16 bit in the FMR1 register.</li> <li>• When inputting and erasing the address in which the Flash memory is not allocated during the erase command input</li> <li>• When executing to erase the block which disables rewriting during the erase command input.</li> <li>• When inputting and writing the address in which the Flash memory is not allocated during the write command input.</li> <li>• When executing to write the block which disables rewriting during the write command input.</li> </ul>
1	0	Erase error	<ul style="list-style-type: none"> <li>• When the block erase command is executed but not automatically erased correctly</li> </ul>
0	1	Program error	<ul style="list-style-type: none"> <li>• When the program command is executed but not automatically programmed correctly.</li> </ul>

**NOTE:**

1. The MCU enters read array mode by writing FFh in the second bus cycle of these commands, at the same time the command code written in the first bus cycle will disabled.

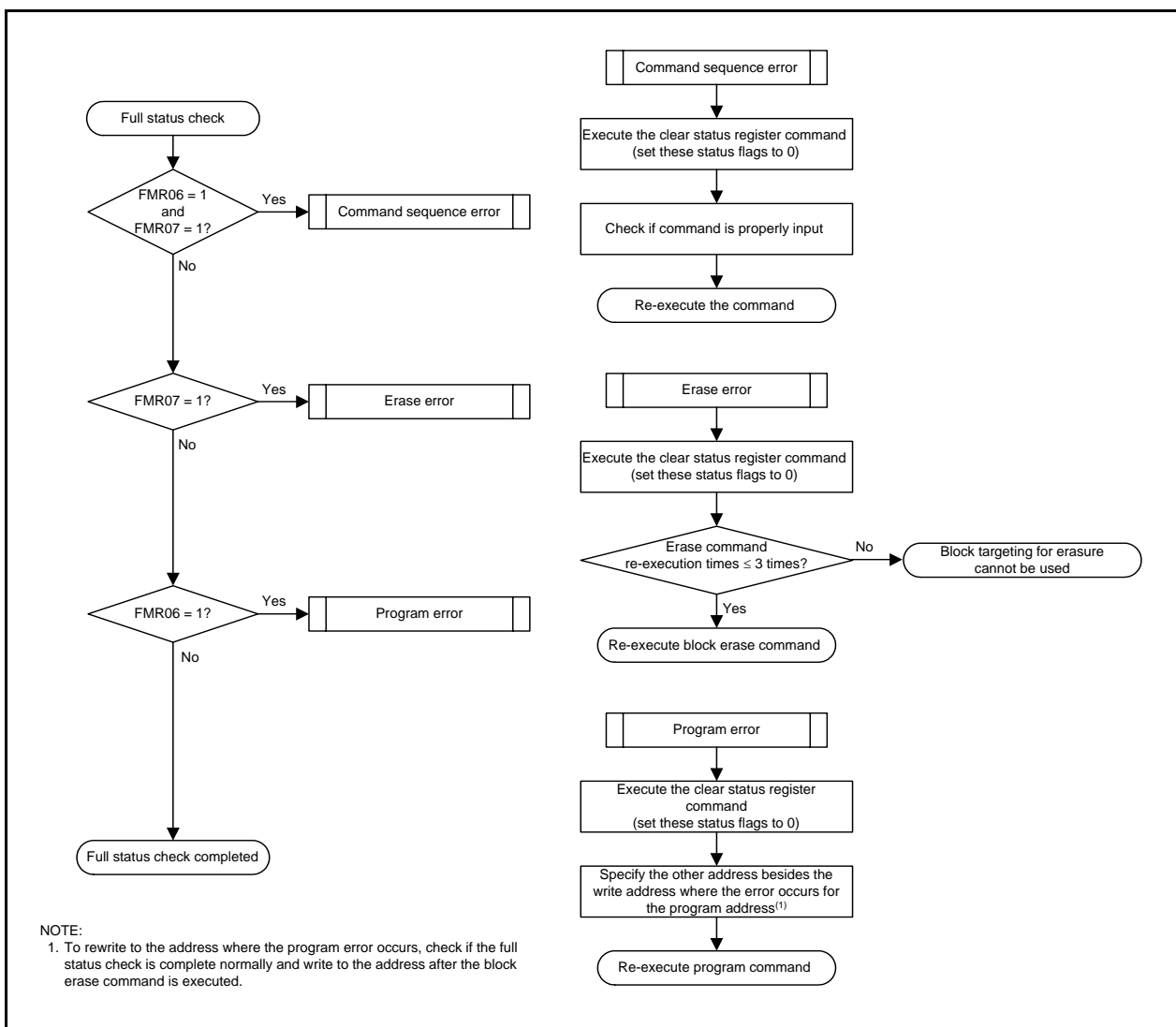


Figure 19.16 Full Status Check and Handling Procedure for Individual Errors

## 19.5 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 applicable 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 2 and Standard serial I/O mode 3.

Refer to **Appendix 2. Connection Examples between Serial Writer and On-Chip Debugging Emulator.** Contact the manufacturer of your serial programmer for serial programmer. Refer to the user's manual of your serial programmer for details on how to use it.

Table 19.7 lists the Pin Functions (Flash Memory Standard Serial I/O Mode 2), Table 19.8 lists the Pin Functions (Flash Memory Standard Serial I/O Mode 3), Figure 19.17 shows Pin Connections for Standard Serial I/O Mode 3. After processing the pins shown in Table 19.8 and rewriting a flash memory using a writer, apply "H" to the MODE pin and reset a hardware if a program is operated on the flash memory in single-chip mode.

### 19.5.1 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 **19.3 Functions to Prevent Rewriting of Flash Memory**).

**Table 19.7 Pin Functions (Flash Memory Standard Serial I/O Mode 2)**

Pin	Name	I/O	Description
VCC,VSS	Power input		Apply the voltage guaranteed for programming and erasure to the VCC pin and 0 V to the VSS pin.
RESET	Reset input	I	Reset input pin.
P4_6/XIN	P4_6 input/clock input	I	Connect a ceramic resonator or crystal oscillator between the XIN and XOUT pins.
P4_7/XOUT	P4_7 input/clock output	I/O	
P0_0 to P0_7	Input port P0	I	Input "H" or "L" level signal or leave the pin open.
P1_0 to P1_7	Input port P1	I	Input "H" or "L" level signal or leave the pin open.
P2_0 to P2_7	Input port P2	I	Input "H" or "L" level signal or leave the pin open.
P3_0, P3_1, P3_3 to P3_5, P3_7	Input port P3	I	Input "H" or "L" level signal or leave the pin open.
P4_2, P4_5	Input port P4	I	Input "H" or "L" level signal or leave the pin open.
P6_0 to P6_5	Input port P6	I	Input "H" or "L" level signal or leave the pin open.
MODE	MODE	I	Input "L".
P6_6	TXD output	O	Serial data input pin.
P6_7	RXD input	I	Serial data output pin.



**Table 19.8 Pin Functions (Flash Memory Standard Serial I/O Mode 3)**

Pin	Name	I/O	Description
VCC,VSS	Power input		Apply the voltage guaranteed for programming and erasure to the VCC pin and 0 V to the VSS pin.
RESET	Reset input	I	Reset input pin.
P4_6/XIN	P4_6 input/clock input	I	Connect ceramic resonator or crystal oscillator between XIN and XOUT pins when connecting external oscillator. Apply "H" and "L" or leave the pin open when using as input port
P4_7/XOUT	P4_7 input/clock output	I/O	
P0_0 to P0_7	Input port P0	I	Input "H" or "L" level signal or leave the pin open.
P1_0 to P1_7	Input port P1	I	Input "H" or "L" level signal or leave the pin open.
P2_0 to P2_7	Input port P2	I	Input "H" or "L" level signal or leave the pin open.
P3_0, P3_1, P3_3 to P3_5, P3_7	Input port P3	I	Input "H" or "L" level signal or leave the pin open.
P4_2 to P4_5	Input port P4	I	Input "H" or "L" level signal or leave the pin open.
P6_0 to P6_7	Input port P6	I	Input "H" or "L" level signal or leave the pin open.
MODE	MODE	I/O	Serial data I/O pin. connect to the flash programmer.

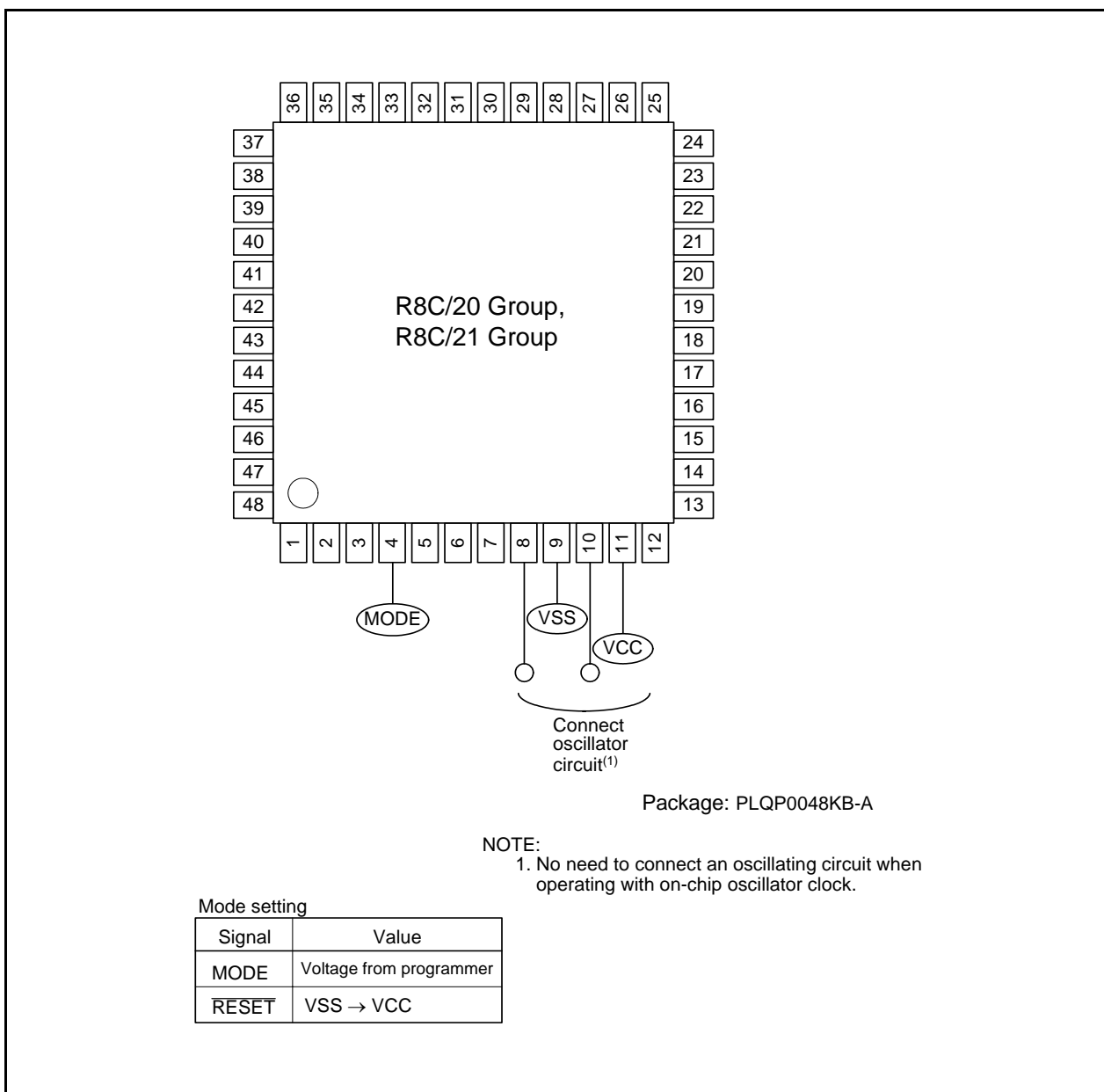


Figure 19.17 Pin Connections for Standard Serial I/O Mode 3

### 19.5.1.1 Example of Circuit Application in the Standard Serial I/O Mode

Figure 19.18 shows an example of Pin Processing in Standard Serial I/O Mode 2 and Figure 19.19 shows an example of Pin Processing in Standard Serial I/O Mode 3. Since the controlled pins vary depending on the programmer, refer to the manual of your serial programmer.

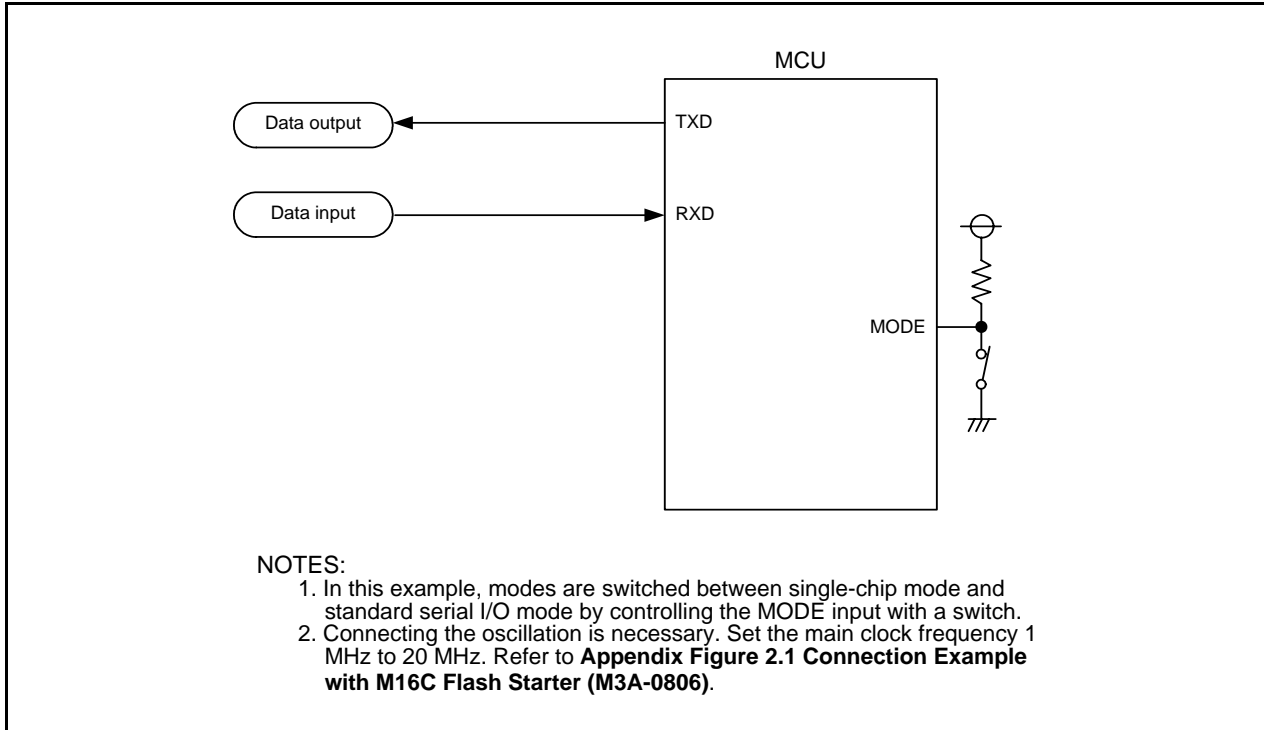


Figure 19.18 Pin Processing in Standard Serial I/O Mode 2

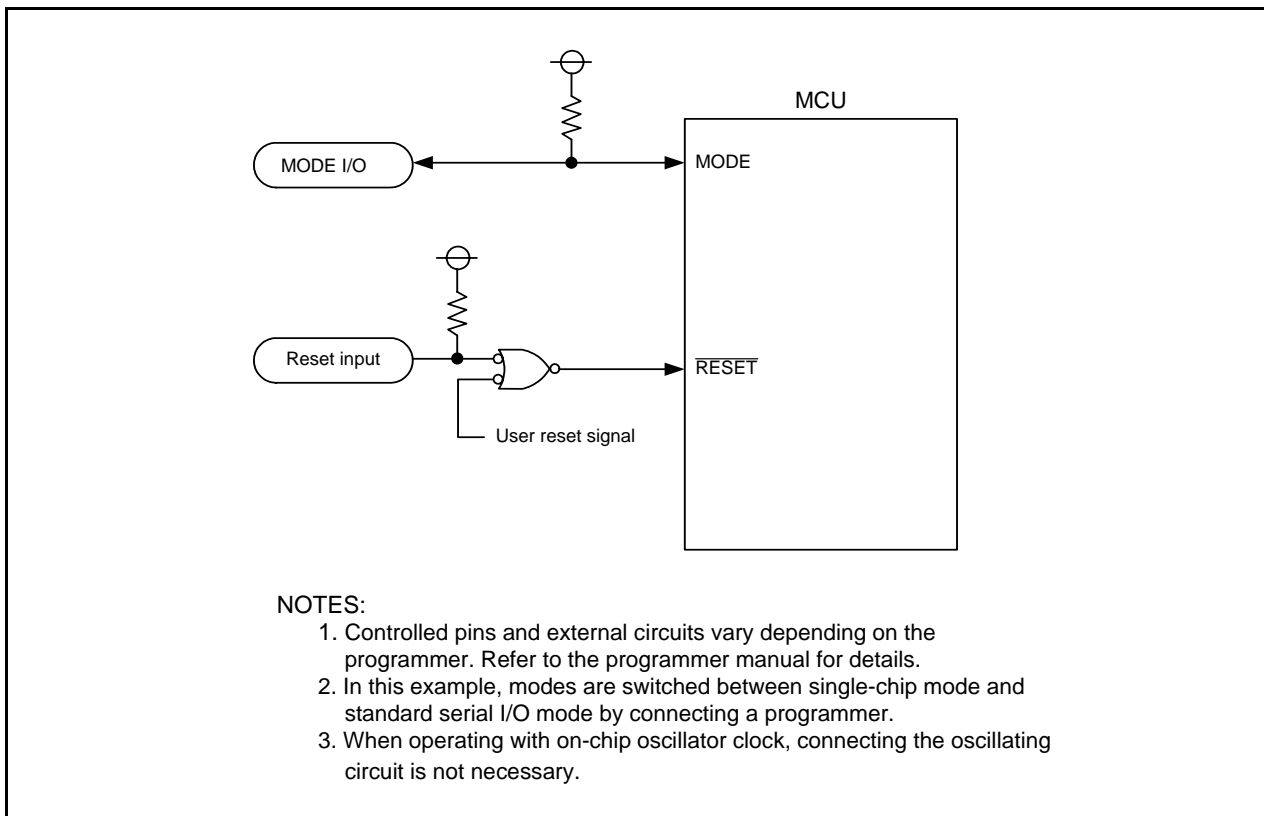


Figure 19.19 Pin Processing in Standard Serial I/O Mode 3

## 19.6 Parallel I/O Mode

Parallel I/O mode is used to input and output the required software command, address and data parallel to controls (read, program and erase) for internal flash memory. Use a parallel programmer which supports this MCU. Contact the manufacturer of your parallel programmer about the parallel programmer and refer to the user's manual of your parallel programmer for details on how to use it.

User ROM area can be rewritten shown in Figure 19.1 and Figure 19.2 in parallel I/O mode.

### 19.6.1 ROM Code Protect Function

The ROM code protect function disables to read and rewrite the flash memory. (Refer to **19.3 Functions to Prevent Rewriting of Flash Memory.**)

## 19.7 Notes on Flash Memory

### 19.7.1 CPU Rewrite Mode

#### 19.7.1.1 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 the CM16 to CM17 bits in the CM1 register. This usage note is not needed for EW1 mode.

#### 19.7.1.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because the flash memory internal data is referenced: UND, INTO, and BRK instructions.

#### 19.7.1.3 Interrupts

Table 19.9 lists the EW0 Mode Interrupts and Table 19.10 lists the EW1 Mode Interrupts.

Table 19.9 EW0 Mode Interrupts

Mode	Status	When Maskable Interrupt Request is Acknowledged	When Watchdog Timer, Oscillation Stop Detection and Voltage Monitor 2 Interrupt Request are Acknowledged
EW0	During automatic erasing	Any interrupt can be used by allocating a vector to RAM	Once an interrupt request is acknowledged, the auto-programming or auto-erasing is forcibly stopped immediately and resets the flash memory. An interrupt process starts after the fixed period and the flash memory restarts. Since the block during the auto-erasing or the address during the auto-programming is forcibly stopped, the normal value may not be read. Execute the auto-erasing again and ensure the auto-erasing is completed normally. Since the watchdog timer does not stop during the command operation, the interrupt request may be generated. Reset the watchdog timer regularly.
	Automatic writing		

NOTES:

1. Do not use the address match interrupt while the command is executed because the vector of the address match interrupt is allocated on ROM.
2. Do not use the non-maskable interrupt while block 0 is automatically erased because the fixed vector is allocated block 0.

Table 19.10 EW1 Mode Interrupts

Mode	Status	When Maskable Interrupt Request is Acknowledged	When Watchdog Timer, Oscillation Stop Detection and Voltage Monitor 2 Interrupt Request are Acknowledged
EW1	During automatic erasing (erase- suspend function is enabled)	The auto-erasing is suspended after td(SR-SUS) and the interrupt process is executed. The auto-erasing can be restarted by setting the FMR41 bit in the FMR4 register to 0 (erase restart) after the interrupt process completes.	Once an interrupt request is acknowledged, the auto-programming or auto-erasing is forcibly stopped immediately and resets the flash memory. An interrupt process starts after the fixed period and the flash memory restarts. Since the block during the auto-erasing or the address during the auto-programming is forcibly stopped, the normal value may not be read. Execute the auto-erasing again and ensure the auto-erasing is completed normally.  Since the watchdog timer does not stop during the command operation, the interrupt request may be generated. Reset the watchdog timer regularly using the erase-suspend function.
	During automatic erasing (erase- suspend function is disabled)	The auto-erasing has a priority and the interrupt request acknowledgement is waited. The interrupt process is executed after the auto-erasing completes.	
	During automatic programming (program suspend function enabled)	The auto-programming is suspended after td(SR-SUS) and the interrupt process is executed. The auto-programming can be restarted by setting the FMR42 bit in the FMR4 register to 0 (program restart) after the interrupt process completes.	
	Auto programming (program suspend function disabled)	The auto-programming has a priority and the interrupt request acknowledgement is waited. The interrupt process is executed after the auto-programming completes.	

## NOTES:

1. Do not use the address match interrupt while the command is executed because the vector of the address match interrupt is allocated on ROM.
2. Do not use the non-maskable interrupt while block 0 is automatically erased because the fixed vector is allocated block 0.

**19.7.1.4 How to Access**

Write 0 to the corresponding bits before writing 1 when setting the FMR01, FMR02, or FMR11 bit to 1. Do not generate an interrupt between writing 0 and 1.

**19.7.1.5 Rewriting User ROM Area**

In EW0 mode, if the power supply voltage drops while rewriting any block in which the rewrite control program is stored, the flash memory may not be able to be rewritten because the rewrite control program cannot be rewritten correctly. In this case, use standard serial I/O mode.

**19.7.1.6 Program**

Do not write additions to the already programmed address.

### **19.7.1.7 Entering Stop Mode or Wait Mode**

Do not enter stop mode or wait mode during erase-suspend.

## 20. Electrical Characteristics

**Table 20.1 Absolute Maximum Ratings**

Symbol	Parameter	Condition	Rated value	Unit
V <sub>CC</sub> /AV <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	-40°C ≤ Topr ≤ 85°C	300	mW
		85°C < Topr ≤ 125°C	125	mW
Topr	Operating ambient temperature		-40 to 85 (J version) / -40 to 125 (K version)	°C
T <sub>stg</sub>	Storage temperature		-65 to 150	°C

**Table 20.2 Recommended Operating Conditions**

Symbol	Parameter		Conditions	Standard			Unit	
				Min.	Typ.	Max.		
V <sub>CC</sub> /AV <sub>CC</sub>	Supply voltage			2.7	–	5.5	V	
V <sub>SS</sub> /AV <sub>CC</sub>	Supply voltage			–	0	–	V	
V <sub>IH</sub>	Input “H” voltage			0.8V <sub>CC</sub>	–	V <sub>CC</sub>	V	
V <sub>IL</sub>	Input “L” voltage			0	–	0.2V <sub>CC</sub>	V	
I <sub>OH(sum)</sub>	Peak sum output “H” current	Sum of all Pins I <sub>OH</sub> (peak)		–	–	-60	mA	
I <sub>OH(peak)</sub>	Peak output “H” current			–	–	-10	mA	
I <sub>OH(avg)</sub>	Average output “H” current			–	–	-5	mA	
I <sub>OL(sum)</sub>	Peak sum output “L” currents	Sum of all Pins I <sub>OL</sub> (peak)		–	–	60	mA	
I <sub>OL(peak)</sub>	Peak output “L” currents			–	–	10	mA	
I <sub>OL(avg)</sub>	Average output “L” current			–	–	5	mA	
f <sub>(XIN)</sub>	XIN clock input oscillation frequency		3.0 V ≤ V <sub>CC</sub> ≤ 5.5 V -40°C ≤ Topr ≤ 85°C	0	–	20	MHz	
			3.0 V ≤ V <sub>CC</sub> ≤ 5.5 V -40°C ≤ Topr ≤ 125°C	0	–	16	MHz	
			2.7 V ≤ V <sub>CC</sub> < 3.0 V	0	–	10	MHz	
–	System clock	OCD2 = 0 When XIN clock is selected.	3.0 V ≤ V <sub>CC</sub> ≤ 5.5 V -40°C ≤ Topr ≤ 85°C	0	–	20	MHz	
			3.0 V ≤ V <sub>CC</sub> ≤ 5.5 V -40°C ≤ Topr ≤ 125°C	0	–	16	MHz	
			2.7 V ≤ V <sub>CC</sub> < 3.0 V	0	–	10	MHz	
		OCD2 = 1 When on-chip oscillator clock is selected.	FRA01 = 0 When low-speed on-chip oscillator clock is selected.	–	125	–	–	kHz
			FRA01 = 1 When high-speed on-chip oscillator clock is selected. 3.0 V ≤ V <sub>CC</sub> ≤ 5.5 V -40°C ≤ Topr ≤ 85°C	–	–	–	20	MHz
			FRA01 = 1 When high-speed on-chip oscillator clock is selected.	–	–	–	10	MHz

**NOTES:**

- V<sub>CC</sub> = 2.7 to 5.5 V at Topr = -40 to 85°C (J version) / -40 to 125°C (K version), unless otherwise specified.
- The average output current indicates the average value of current measured during 100 ms.

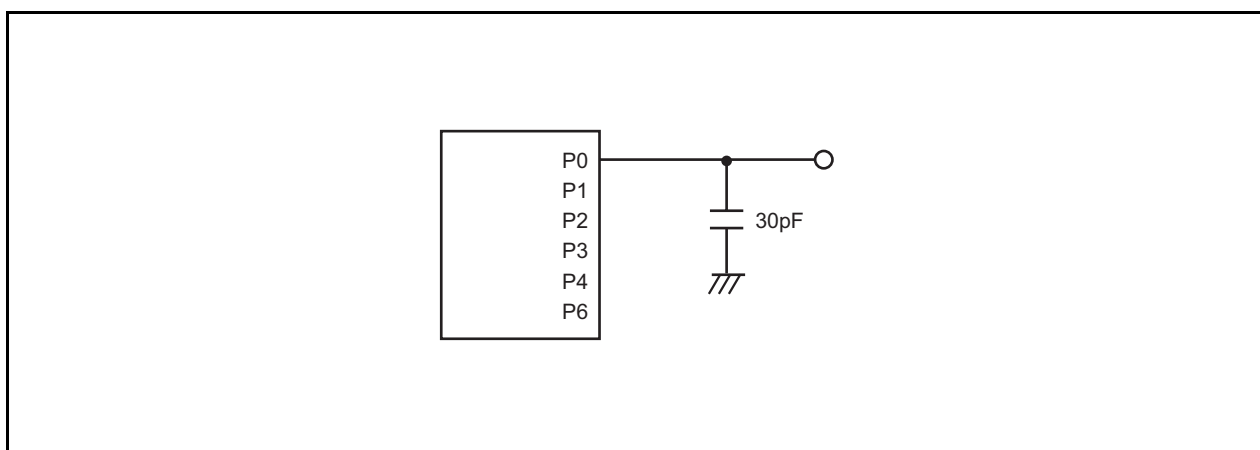


**Table 20.3 A/D Converter Characteristics**

Symbol	Parameter		Conditions	Standard			Unit
				Min.	Typ.	Max.	
–	Resolution		$V_{ref} = AV_{CC}$	–	–	10	Bits
–	Absolute Accuracy	10-bit mode	$\phi_{AD} = 10 \text{ MHz}, V_{ref} = AV_{CC} = 5.0 \text{ V}$	–	–	$\pm 3$	LSB
		8-bit mode	$\phi_{AD} = 10 \text{ MHz}, V_{ref} = AV_{CC} = 5.0 \text{ V}$	–	–	$\pm 2$	LSB
		10-bit mode	$\phi_{AD} = 10 \text{ MHz}, V_{ref} = AV_{CC} = 3.3 \text{ V}$	–	–	$\pm 5$	LSB
		8-bit mode	$\phi_{AD} = 10 \text{ MHz}, V_{ref} = AV_{CC} = 3.3 \text{ V}$	–	–	$\pm 2$	LSB
$R_{ladder}$	Resistor ladder		$V_{ref} = AV_{CC}$	10	–	40	$k\Omega$
$t_{conv}$	Conversion time	10-bit mode	$\phi_{AD} = 10 \text{ MHz}, V_{ref} = AV_{CC} = 5.0 \text{ V}$	3.3	–	–	$\mu\text{s}$
		8-bit mode	$\phi_{AD} = 10 \text{ MHz}, V_{ref} = AV_{CC} = 5.0 \text{ V}$	2.8	–	–	$\mu\text{s}$
$V_{ref}$	Reference voltage			2.7	–	$AV_{CC}$	V
$V_{IA}$	Analog input voltage <sup>(2)</sup>			0	–	$AV_{CC}$	V
–	A/D operating clock frequency	Without sample & hold		0.25	–	10	MHz
		With sample & hold		1	–	10	MHz

## NOTES:

- $V_{CC} = AV_{CC} = 2.7$  to  $5.5 \text{ V}$  at  $T_{opr} = -40$  to  $85^\circ\text{C}$  (J version) /  $-40$  to  $125^\circ\text{C}$  (K version), unless otherwise specified.
- When analog input voltage exceeds reference voltage, A/D conversion result is 3FFh in 10-bit mode, FFh in 8-bit mode.

**Figure 20.1 Ports P0 to P4, P6 Timing Measurement Circuit**

**Table 20.4 Flash Memory (Program ROM) Electrical Characteristics**

Symbol	Parameter	Conditions	Standard			Unit
			Min.	Typ.	Max.	
–	Program/erase endurance <sup>(2)</sup>	R8C/20 Group	100 <sup>(3)</sup>	–	–	times
		R8C/21 Group	1,000 <sup>(3)</sup>	–	–	times
–	Byte program time		–	50	400	μs
–	Block erase time		–	0.4	9	s
t <sub>d</sub> (SR-SUS)	Time delay from suspend request until erase suspend		–	–	97 + CPU clock × 6 cycle	μs
–	Interval from erase start/restart until following suspend request		650	–	–	μs
–	Interval from program start/restart until following suspend request		0	–	–	ns
–	Time from suspend until program/erase restart		–	–	3 + CPU clock × 4 cycle	μs
–	Program, erase voltage		2.7	–	5.5	V
–	Read voltage		2.7	–	5.5	V
–	Program, erase temperature		0	–	60	°C
–	Data hold time <sup>(7)</sup>	Ambient temperature = 55°C	20	–	–	year

## NOTES:

- V<sub>CC</sub> = 2.7 to 5.5 V at Topr = -40 to 85°C (J version) / -40 to 125°C (K version), unless otherwise specified.
- 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 1,000), each block can be erased n times.  
For example, if 1,024 1-byte writes are performed to different addresses in 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).
- Endurance to guarantee all electrical characteristics after program and erase (1 to Min. value can be guaranteed).
- In a system that executes multiple programming operations, the actual erasure endurance 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 erasure endurance of each block and limit the number of erase operations to a certain number.
- If error occurs during block erase, attempt to execute the clear status register command, then the block erase command at least three times until the erase error does not occur.
- Customers desiring program/erase failure rate information should contact their Renesas technical support representative.
- The data hold time includes time that the power supply is off or the clock is not supplied.

**Table 20.5 Flash Memory (Data Flash Block A, Block B) Electrical Characteristics<sup>(4)</sup>**

Symbol	Parameter	Conditions	Standard			Unit
			Min.	Typ.	Max.	
–	Program/erase endurance <sup>(2)</sup>		10,000 <sup>(3)</sup>	–	–	times
–	Byte program time (Program/erase endurance ≤ 1,000 times)		–	50	400	μs
–	Byte program time (Program/erase endurance > 1,000 times)		–	65	–	μs
–	Block erase time (Program/erase endurance ≤ 1,000 times)		–	0.2	9	s
–	Block erase time (Program/erase endurance > 1,000 times)		–	0.3	–	s
t <sub>d(SR-SUS)</sub>	Time delay from suspend request until erase suspend		–	–	97 + CPU clock × 6 cycle	μs
–	Interval from erase start/restart until following suspend request		650	–	–	μs
–	Interval from program start/restart until following suspend request		0	–	–	ns
–	Time from suspend until program/erase restart		–	–	3 + CPU clock × 4 cycle	μs
–	Program, erase voltage		2.7	–	5.5	V
–	Read voltage		2.7	–	5.5	V
–	Program, erase temperature		-40	–	85 <sup>(8)</sup>	°C
–	Data hold time <sup>(9)</sup>	Ambient temperature = 55°C	20	–	–	year

**NOTES:**

- V<sub>CC</sub> = 2.7 to 5.5 V at Topr = -40 to 85°C (J version) / -40 to 125°C (K version), unless otherwise specified.
- 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 = 10,000), each block can be erased n times.  
For example, if 1,024 1-byte writes are performed to different addresses in 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).
- Minimum endurance to guarantee all electrical characteristics after program and erase (1 to Min. value can be guaranteed).
- Standard of block A and block B when program and erase endurance exceeds 1,000 times. Byte program time to 1,000 times are the same as that in program ROM.
- In a system that executes multiple programming operations, the actual erasure endurance can be reduced by writing to sequential addresses in turn so that as much of the block as possible is used up before performing an erase operation. For example, when programming groups of 16 bytes, the effective number of rewrites can be minimized by programming up to 128 groups before erasing them all in one operation. In addition, averaging the erasure endurance between blocks A and B can further reduce the actual erasure endurance. It is also advisable to retain data on the erasure endurance of each block and limit the number of erase operations to a certain number.
- If error occurs during block erase, attempt to execute the clear status register command, then the block erase command at least three times until the erase error does not occur.
- Customers desiring program/erase failure rate information should contact their Renesas technical support representative.
- 125°C for K version.
- The data hold time includes time that the power supply is off or the clock is not supplied.

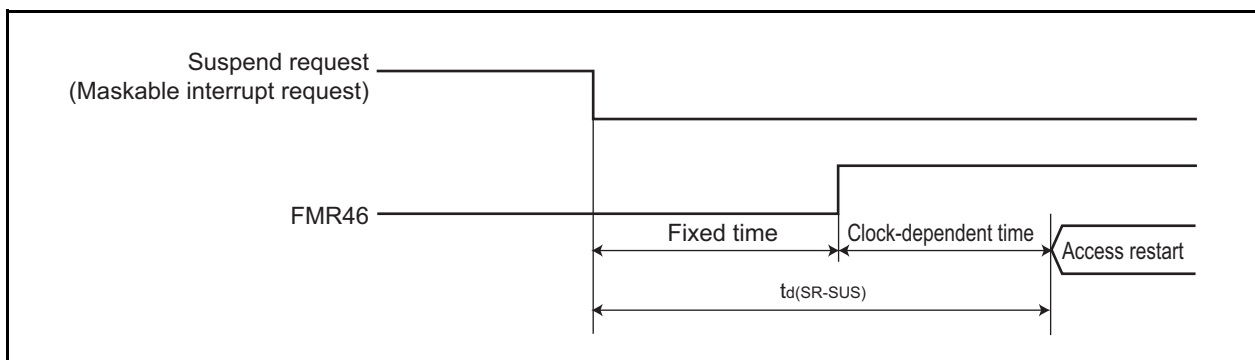


Figure 20.2 Time delay until Suspend

Table 20.6 Voltage Detection 1 Circuit Electrical Characteristics

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V <sub>det1</sub>	Voltage detection level <sup>(3, 4)</sup>		2.70	2.85	3.00	V
t <sub>d</sub> (V <sub>det1</sub> -A)	Voltage monitor 1 reset generation time <sup>(5)</sup>		–	40	200	μs
–	Voltage detection circuit self power consumption	VCA26 = 1, V <sub>CC</sub> = 5.0 V	–	0.6	–	μA
t <sub>d</sub> (E-A)	Waiting time until voltage detection circuit operation starts <sup>(2)</sup>		–	–	100	μs
V <sub>CCmin</sub>	MCU operating voltage minimum value		2.70	–	–	V

## NOTES:

1. The measurement condition is V<sub>CC</sub> = 2.7 V to 5.5 V and Topr = -40°C to 85°C (J version) / -40°C to 125°C (K version).
2. Necessary time until the voltage detection circuit operates when setting to 1 again after setting the VCA26 bit in the VCA2 register to 0.
3. Hold V<sub>det2</sub> > V<sub>det1</sub>.
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.
5. Time until the voltage monitor 1 reset is generated after the voltage passes V<sub>det1</sub> when V<sub>CC</sub> falls. When using the digital filter, its sampling time is added to t<sub>d</sub>(V<sub>det1</sub>-A). When using the voltage monitor 1 reset, maintain this time until V<sub>CC</sub> = 2.0 V after the voltage passes V<sub>det1</sub> when the power supply falls.

Table 20.7 Voltage Detection 2 Circuit Electrical Characteristics

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V <sub>det2</sub>	Voltage detection level <sup>(4)</sup>		3.3	3.6	3.9	V
t <sub>d</sub> (V <sub>det2</sub> -A)	Voltage monitor 2 reset/interrupt request generation time <sup>(2, 5)</sup>		–	40	200	μs
–	Voltage detection circuit self power consumption	VCA27 = 1, V <sub>CC</sub> = 5.0V	–	0.6	–	μA
t <sub>d</sub> (E-A)	Waiting time until voltage detection circuit operation starts <sup>(3)</sup>		–	–	100	μs

## NOTES:

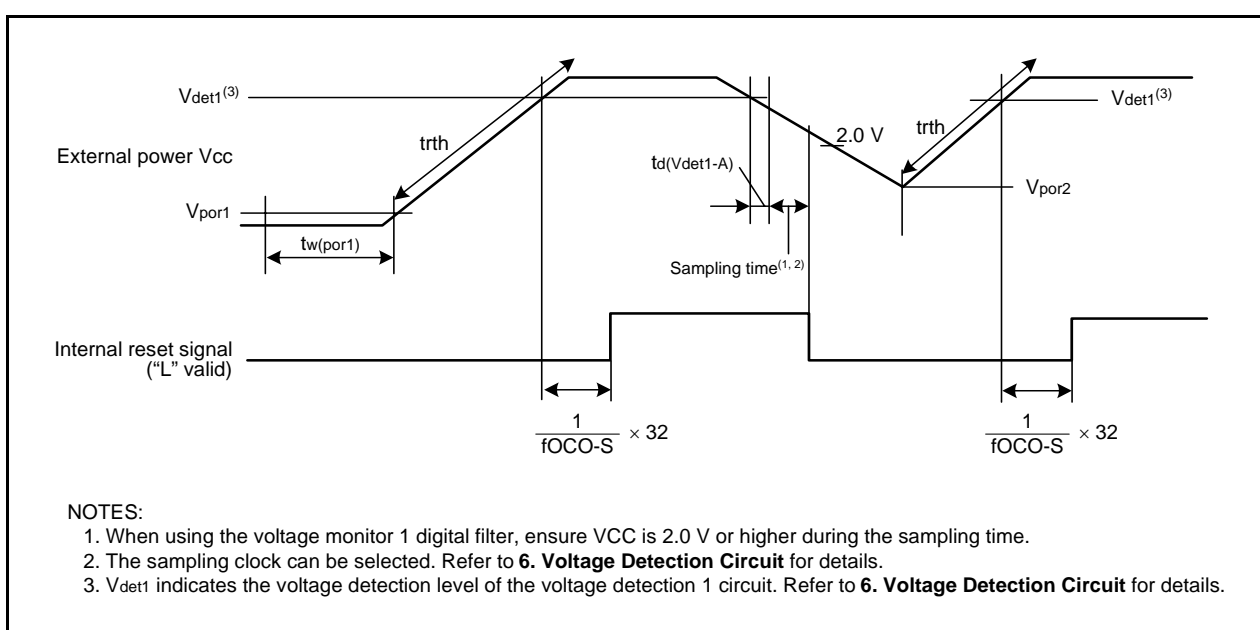
1. The measurement condition is V<sub>CC</sub> = 2.7 V to 5.5 V and Topr = -40°C to 85°C (J version) / -40°C to 125°C (K version).
2. Time until the voltage monitor 2 reset/interrupt request is generated since the voltage passes V<sub>det2</sub>.
3. Necessary time until the voltage detection circuit operates when setting to 1 again after setting the VCA27 bit in the VCA2 register to 0.
4. Hold V<sub>det2</sub> > V<sub>det1</sub>.
5. When using the digital filter, its sampling time is added to t<sub>d</sub>(V<sub>det2</sub>-A). When using the voltage monitor 2 reset, maintain this time until V<sub>CC</sub> = 2.0 V after the voltage passes V<sub>det2</sub> when the power supply falls.

**Table 20.8 Power-on Reset Circuit, Voltage Monitor 1 Reset Circuit Electrical Characteristics<sup>(3)</sup>**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
$V_{por1}$	Power-on reset valid voltage <sup>(4)</sup>		–	–	0.1	V
$V_{por2}$	Power-on reset or voltage monitor 1 valid voltage		0	–	$V_{det1}$	V
trth	External power Vcc rise gradient	$V_{cc} \leq 3.6$ V	20 <sup>(2)</sup>	–	–	mV/msec
		$V_{cc} > 3.6$ V	20 <sup>(2)</sup>	–	2,000	mV/msec

## NOTES:

1.  $T_{opr} = -40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$  (J version) /  $-40^{\circ}\text{C}$  to  $125^{\circ}\text{C}$  (K version), unless otherwise specified.
2. This condition (the minimum value of external power Vcc rise gradient) does not apply if  $V_{por2} \geq 1.0$  V.
3. To use the power-on reset function, enable voltage monitor 1 reset by setting the LVD1ON bit in the OFS register to 0, the VW1C0 and VW1C6 bits in the VW1C register to 1 respectively, and the VCA26 bit in the VCA2 register to 1.
4.  $t_{w(por1)}$  indicates the duration the external power Vcc must be held below the effective voltage ( $V_{por1}$ ) to enable a power on reset. When turning on the power for the first time, maintain  $t_{w(por1)}$  for 30s or more if  $-20^{\circ}\text{C} \leq T_{opr} \leq 125^{\circ}\text{C}$ , maintain  $t_{w(por1)}$  for 3,000s or more if  $-40^{\circ}\text{C} \leq T_{opr} < -20^{\circ}\text{C}$ .

**Figure 20.3 Power-on Reset Circuit Electrical Characteristics**

**Table 20.9 High-Speed On-Chip Oscillator Circuit Electrical Characteristics**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
fOCO40M	High-speed on-chip oscillator frequency temperature • supply voltage dependence	$V_{CC} = 4.75 \text{ V to } 5.25 \text{ V}$ , $0^{\circ}\text{C} \leq T_{opr} \leq 60^{\circ}\text{C}^{(2)}$	39.2	40	40.8	MHz
		$V_{CC} = 3.0 \text{ V to } 5.25 \text{ V}$ , $-20^{\circ}\text{C} \leq T_{opr} \leq 85^{\circ}\text{C}^{(2)}$	38.8	40	41.2	MHz
		$V_{CC} = 3.0 \text{ V to } 5.5 \text{ V}$ , $-40^{\circ}\text{C} \leq T_{opr} \leq 85^{\circ}\text{C}^{(2)}$	38.4	40	41.6	MHz
		$V_{CC} = 3.0 \text{ V to } 5.5 \text{ V}$ , $-40^{\circ}\text{C} \leq T_{opr} \leq 125^{\circ}\text{C}^{(2)}$	38.0	40	42.0	MHz
		$V_{CC} = 2.7 \text{ V to } 5.5 \text{ V}$ , $-40^{\circ}\text{C} \leq T_{opr} \leq 125^{\circ}\text{C}^{(2)}$	37.6	40	42.4	MHz
–	The value of the FRA1 register when the reset is deasserted		08h	40	F7h	–
–	High-speed on-chip oscillator adjustment range	Adjust the FRA1 register to -1 bit (the value when the reset is deasserted)	–	+ 0.3	–	MHz
–	Oscillation stability time		–	10	100	$\mu\text{s}$
–	Self power consumption when high-speed on-chip oscillator oscillating	$V_{CC} = 5.0 \text{ V}$ , $T_{opr} = 25^{\circ}\text{C}$	–	600	–	$\mu\text{A}$

## NOTES:

- $V_{CC} = 2.7 \text{ V to } 5.5 \text{ V}$ ,  $T_{opr} = -40^{\circ}\text{C to } 85^{\circ}\text{C}$  (J version) /  $-40^{\circ}\text{C to } 125^{\circ}\text{C}$  (K version), unless otherwise specified.
- The standard value shows when the reset is deasserted for the FRA1 register.

**Table 20.10 Low-Speed On-Chip Oscillator Circuit Electrical Characteristics**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
fOCO-S	Low-speed on-chip oscillator frequency		40	125	250	kHz
–	Oscillation stability time		–	10	100	$\mu\text{s}$
–	Self power consumption when low-speed on-chip oscillator oscillating	$V_{CC} = 5.0 \text{ V}$ , $T_{opr} = 25^{\circ}\text{C}$	–	15	–	$\mu\text{A}$

## NOTE:

- $V_{CC} = 2.7 \text{ V to } 5.5 \text{ V}$ ,  $T_{opr} = -40^{\circ}\text{C to } 85^{\circ}\text{C}$  (J version) /  $-40^{\circ}\text{C to } 125^{\circ}\text{C}$  (K version), unless otherwise specified.

**Table 20.11 Power Supply Circuit Timing Characteristics**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
$t_{d(P-R)}$	Time for internal power supply stabilization during power-on <sup>(2)</sup>		1	–	2000	$\mu\text{s}$
$t_{d(R-S)}$	STOP exit time <sup>(3)</sup>		–	–	150	$\mu\text{s}$

## NOTES:

- The measurement condition is  $V_{CC} = 2.7 \text{ to } 5.5 \text{ V}$  and  $T_{opr} = -40^{\circ}\text{C to } 85^{\circ}\text{C}$  (J version) /  $-40^{\circ}\text{C to } 125^{\circ}\text{C}$  (K version), unless otherwise specified.
- Waiting time until the internal power supply generation circuit stabilizes during power-on.
- Time until CPU clock supply starts since the interrupt is acknowledged to exit stop mode.

**Table 20.12 Timing Requirements of Clock Synchronous Serial I/O with Chip Select<sup>(1)</sup>**

Symbol	Parameter	Conditions	Standard			Unit
			Min.	Typ.	Max.	
tsUCYC	SSCK clock cycle time		4	–	–	tcyc <sup>(2)</sup>
tHI	SSCK clock "H" width		0.4	–	0.6	tsUCYC
tLO	SSCK clock "L" width		0.4	–	0.6	tsUCYC
tRISE	SSCK clock rising time	Master	–	–	1	tcyc <sup>(2)</sup>
		Slave	–	–	1	μs
tFALL	SSCK clock falling time	Master	–	–	1	tcyc <sup>(2)</sup>
		Slave	–	–	1	μs
tsU	SSO, SSI data input setup time		100	–	–	ns
tH	SSO, SSI data input hold time		1	–	–	tcyc <sup>(2)</sup>
tLEAD	$\overline{\text{SCS}}$ setup time	Slave	1tcyc + 50	–	–	ns
tLAG	$\overline{\text{SCS}}$ hold time	Slave	1tcyc + 50	–	–	ns
tOD	SSO, SSI data output delay time		–	–	1	tcyc <sup>(2)</sup>
tSA	SSI slave access time		–	–	1tcyc + 100	ns
tOR	SSI slave out open time		–	–	1tcyc + 100	ns

## NOTES:

1. Vcc = 2.7 to 5.5 V, Vss = 0 V at Topr = -40 to 85°C (J version) / -40 to 125°C (K version), unless otherwise specified.
2. 1tcyc = 1/f1(s)

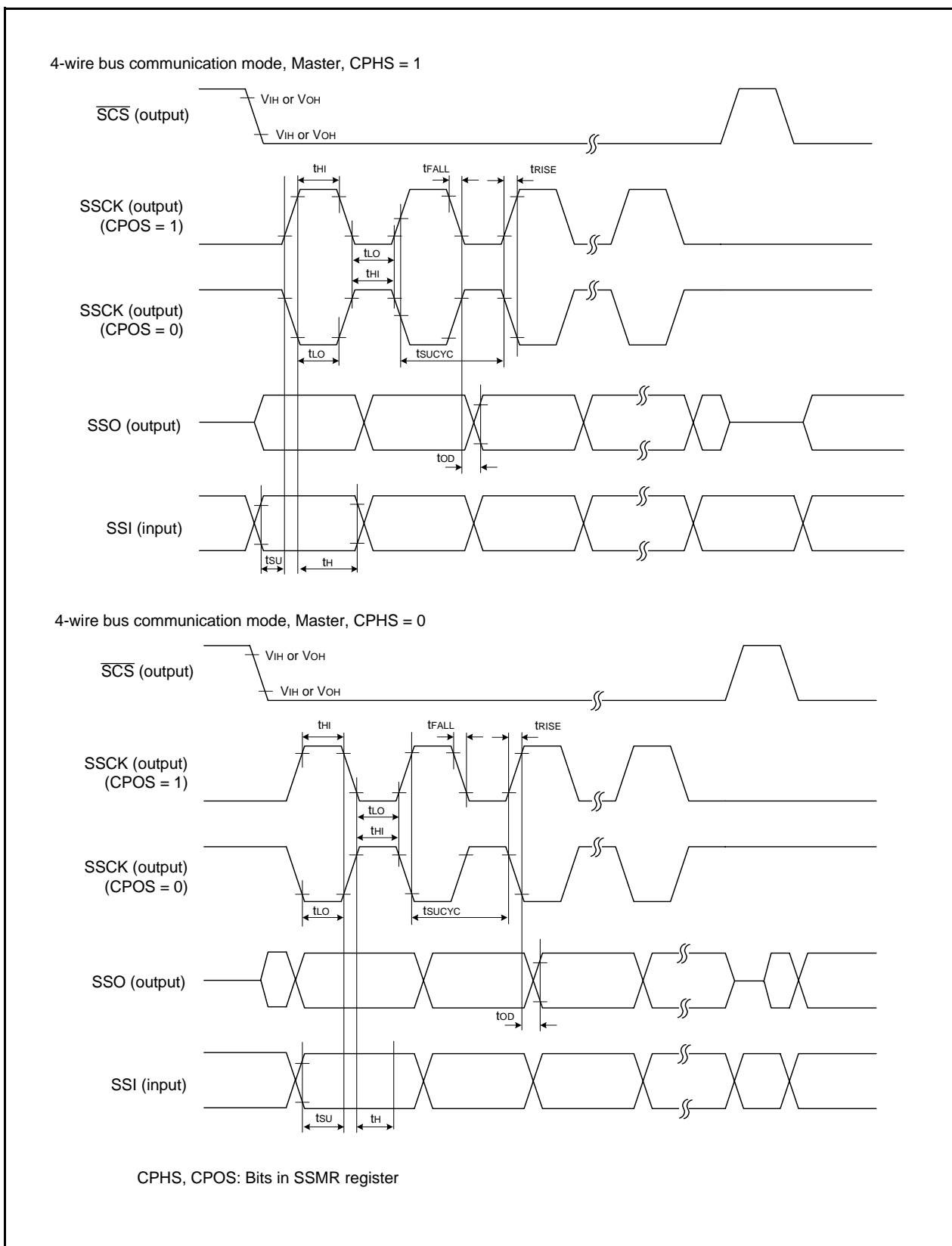


Figure 20.4 I/O Timing of Clock Synchronous Serial I/O with Chip Select (Master)



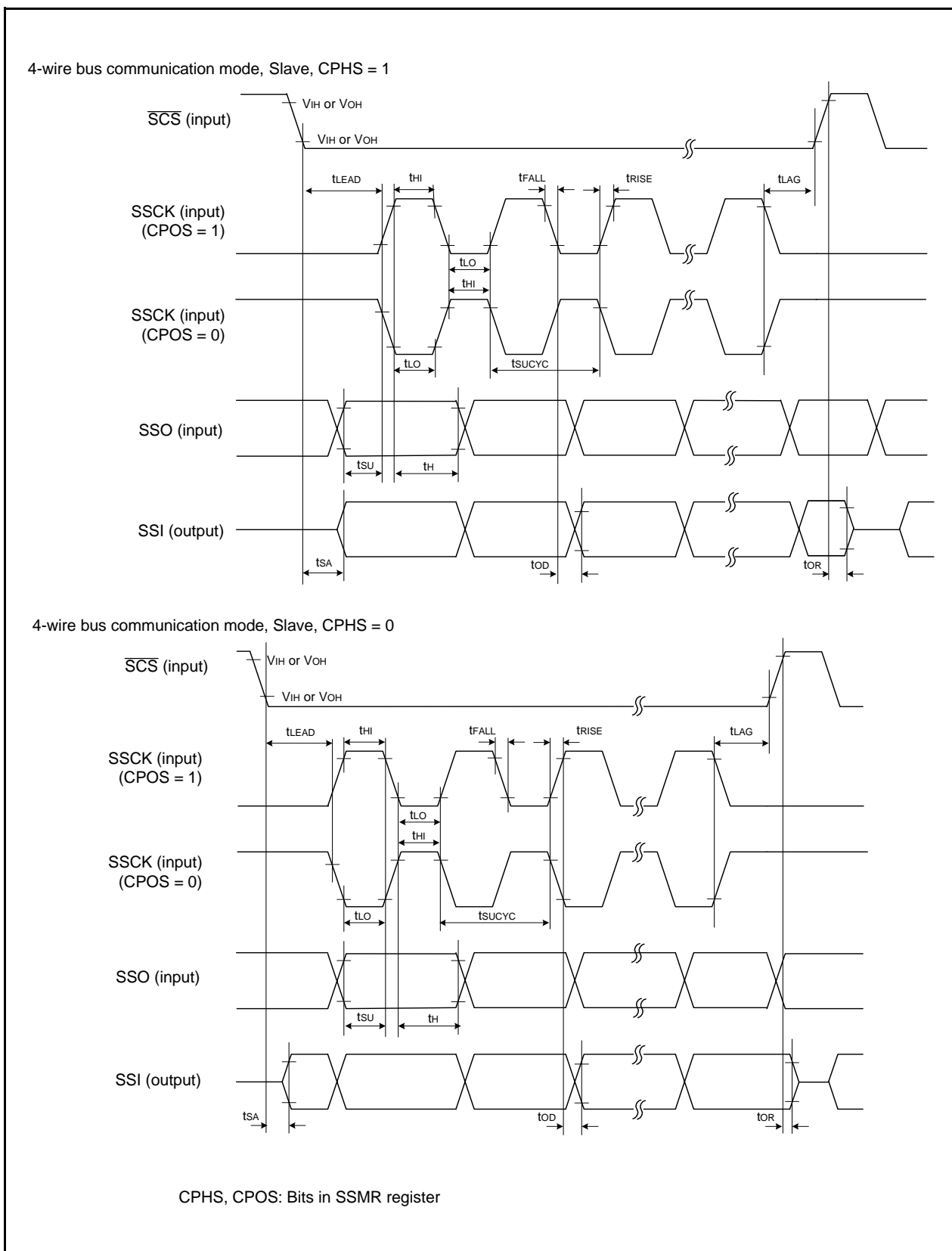
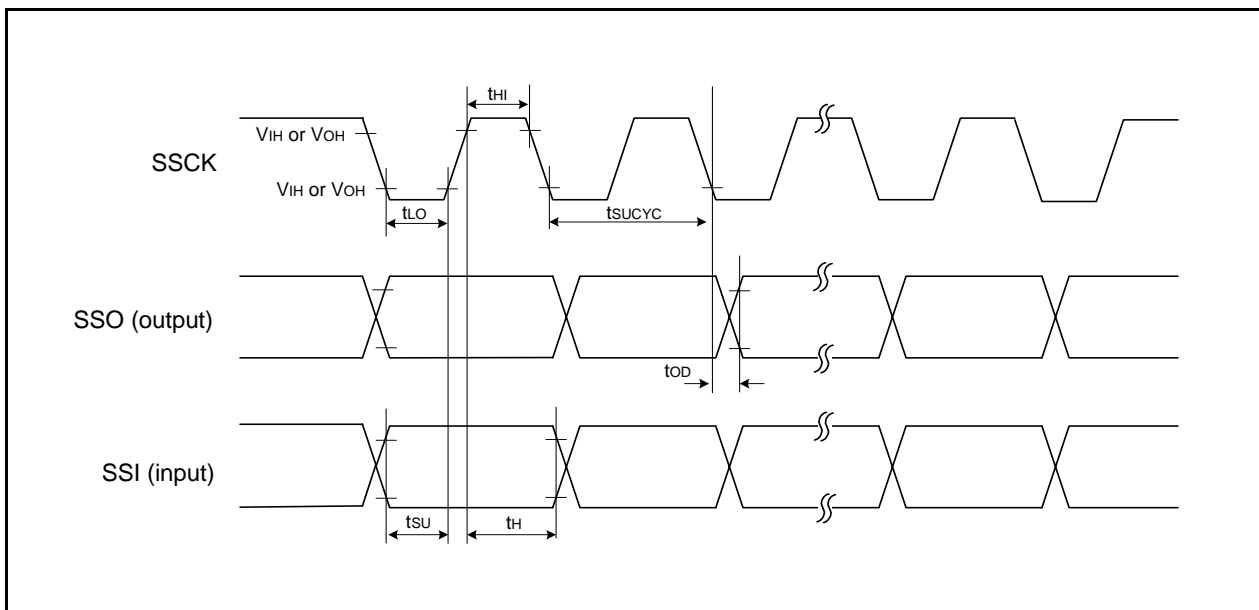


Figure 20.5 I/O Timing of Clock Synchronous Serial I/O with Chip Select (Slave)



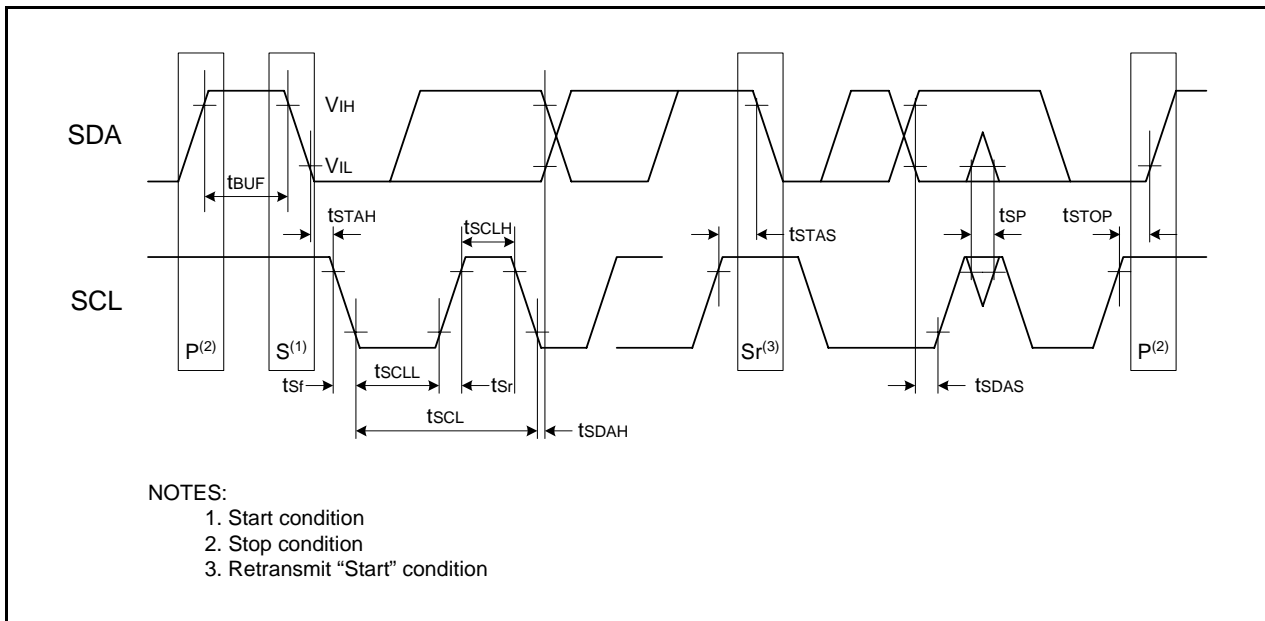
**Figure 20.6 I/O Timing of Clock Synchronous Serial I/O with Chip Select (Clock Synchronous Communication Mode)**

**Table 20.13 Timing Requirements of I<sup>2</sup>C Bus Interface(1)**

Symbol	Parameter	Conditions	Standard			Unit
			Min.	Typ.	Max.	
t <sub>SCL</sub>	SCL input cycle time		12t <sub>CYC</sub> + 600 <sup>(2)</sup>	–	–	ns
t <sub>SCLH</sub>	SCL input “H” width		3t <sub>CYC</sub> + 300 <sup>(2)</sup>	–	–	ns
t <sub>SCLL</sub>	SCL input “L” width		5t <sub>CYC</sub> + 300 <sup>(2)</sup>	–	–	ns
t <sub>sf</sub>	SCL, SDA input falling time		–	–	300	ns
t <sub>SP</sub>	SCL, SDA input spike pulse rejection time		–	–	1t <sub>CYC</sub> <sup>(2)</sup>	ns
t <sub>BUF</sub>	SDA input bus-free time		5t <sub>CYC</sub> <sup>(2)</sup>	–	–	ns
t <sub>STAH</sub>	Start condition input hole time		3t <sub>CYC</sub> <sup>(2)</sup>	–	–	ns
t <sub>STAS</sub>	Retransmit start condition input setup time		3t <sub>CYC</sub> <sup>(2)</sup>	–	–	ns
t <sub>STOP</sub>	Stop condition input setup time		3t <sub>CYC</sub> <sup>(2)</sup>	–	–	ns
t <sub>SOAS</sub>	Data input setup time		1t <sub>CYC</sub> + 20 <sup>(2)</sup>	–	–	ns
t <sub>SDAH</sub>	Data input hold time		0	–	–	ns

NOTES:

- V<sub>CC</sub> = 2.7 to 5.5 V, V<sub>SS</sub> = 0V at Topr = -40 to 85°C (J version) / -40 to 125°C (K version), unless otherwise specified.
- 1t<sub>CYC</sub> = 1/f<sub>1</sub>(s)



**Figure 20.7 I/O Timing of I<sup>2</sup>C Bus Interface**

**Table 20.14 Electrical Characteristics (1) [Vcc = 5 V]**

Symbol	Parameter		Condition		Standard			Unit
					Min.	Typ.	Max.	
VOH	Output "H" Voltage	Except XOUT	IOH = -5 mA		Vcc - 2.0	-	Vcc	V
			IOH = -200 μA		Vcc - 0.3	-	Vcc	V
		XOUT	Drive capacity HIGH	IOH = -1 mA	Vcc - 2.0	-	Vcc	V
			Drive capacity LOW	IOH = -500 μA	Vcc - 2.0	-	Vcc	V
VOL	Output "L" Voltage	Except XOUT	IOL = 5 mA		-	-	2.0	V
			IOL = 200 μA		-	-	0.45	V
		XOUT	Drive capacity HIGH	IOL = 1 mA	-	-	2.0	V
			Drive capacity LOW	IOL = 500 μA	-	-	2.0	V
VT+ - VT-	Hysteresis	INT0, INT1, INT2, INT3, KI0, KI1, KI2, KI3, TRAIO, RXD0, RXD1, CLK0, SSI, SCL, SDA, SSO			0.1	0.5	-	V
		RESET			0.1	1.0	-	V
IiH	Input "H" current		VI = 5 V, Vcc = 5 V		-	-	5.0	μA
IiL	Input "L" current		VI = 0 V, Vcc = 5 V		-	-	-5.0	μA
RPULLUP	Pull-Up Resistance		VI = 0 V, Vcc = 5 V		30	50	167	kΩ
RfXIN	Feedback Resistance	XIN			-	1.0	-	MΩ
VRAM	RAM Hold Voltage		During stop mode		2.0	-	-	V

## NOTE:

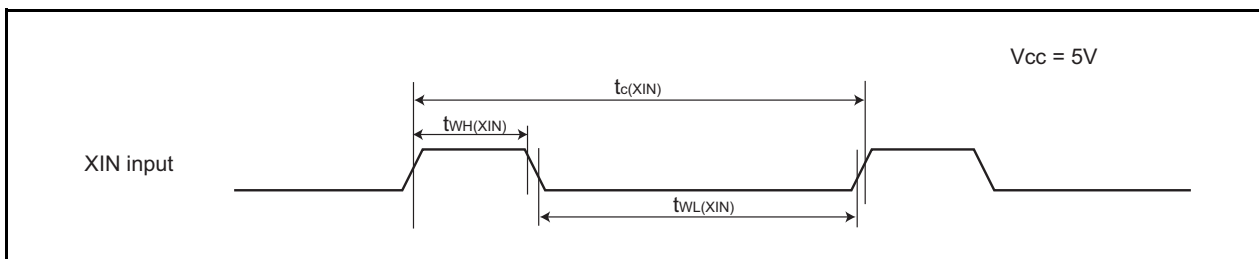
- Vcc = 4.2 to 5.5 V at Topr = -40 to 85°C (J version) / -40 to 125°C (K version), f(XIN) = 20 MHz, unless otherwise specified.

**Table 20.15 Electrical Characteristics (2) [V<sub>CC</sub> = 5 V]  
(Topr = -40 to 85°C (J version) / -40 to 125°C (K version), Unless Otherwise Specified.)**

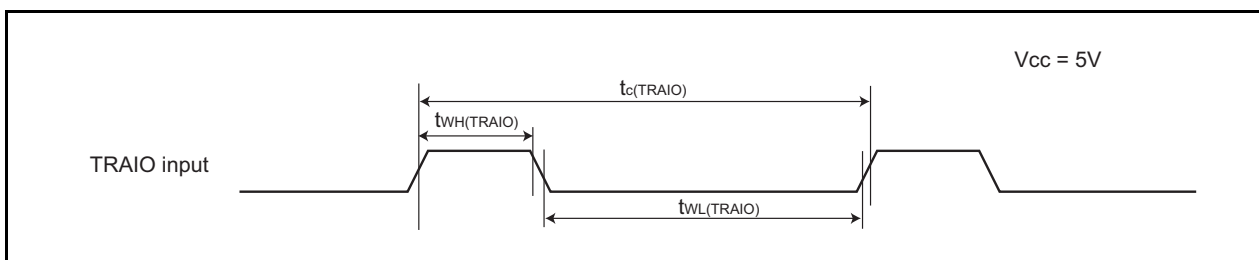
Symbol	Parameter	Condition	Standard			Unit	
			Min.	Typ.	Max.		
I <sub>CC</sub>	Power supply current (V <sub>CC</sub> = 3.3 to 5.5 V) In single-chip mode, the output pins are open and other pins are V <sub>SS</sub>	High-clock mode	XIN = 20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	11.0	22.0	mA
			XIN = 16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	8.8	17.6	mA
			XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	5.8	–	mA
			XIN = 20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	5.0	–	mA
			XIN = 16MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	3.8	–	mA
			XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	2.8	–	mA
		High-speed on-chip oscillator mode	XIN clock off High-speed on-chip oscillator on f <sub>OCO</sub> = 10 MHz Low-speed on-chip oscillator on = 125 kHz No division	–	5.8	11.6	mA
			XIN clock off High-speed on-chip oscillator on f <sub>OCO</sub> = 10 MHz Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	2.5	–	mA
		Low-speed on-chip oscillator mode	XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8 FMR47 = 1	–	143	286	μA
		Wait mode	XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock operation VCA20 = 0 VCA26 = VCA27 = 0	–	53	106	μA
			XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock off VCA20 = 0 VCA26 = VCA27 = 0	–	38	76	μA
		Stop mode Topr = 25°C	XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA26 = VCA27 = 0	–	0.8	3.0	μA
		Stop mode Topr = 85°C	XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA26 = VCA27 = 0	–	1.2	–	μA
		Stop mode Topr = 125°C	XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA26 = VCA27 = 0	–	4.0	–	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 20.16 XIN Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(XIN)}$	XIN input cycle time	50	–	ns
$t_{WH(XIN)}$	XIN input "H" width	25	–	ns
$t_{WL(XIN)}$	XIN input "L" width	25	–	ns

**Figure 20.8 XIN Input Timing Diagram when  $V_{CC} = 5\text{ V}$** **Table 20.17 TRAIO Input**

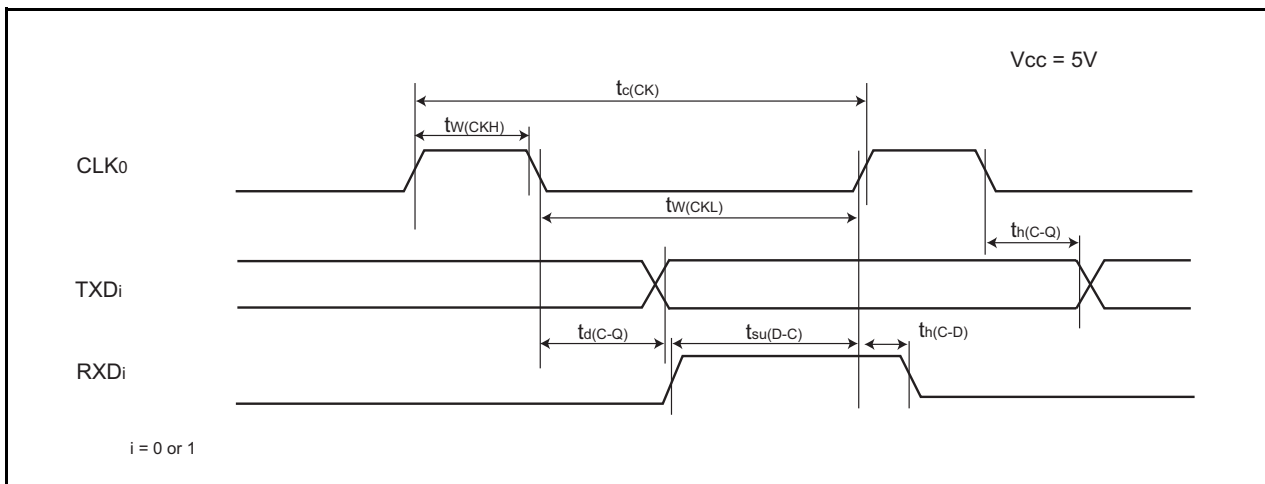
Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TRAIO)}$	TRAIO input cycle time	100	–	ns
$t_{WH(TRAIO)}$	TRAIO input "H" width	40	–	ns
$t_{WL(TRAIO)}$	TRAIO input "L" width	40	–	ns

**Figure 20.9 TRAIO Input Timing Diagram when  $V_{CC} = 5\text{ V}$**

**Table 20.18 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLK0 input cycle time	200	–	ns
$t_{w(CKH)}$	CLK0 input “H” width	100	–	ns
$t_{w(CKL)}$	CLK0 input “L” width	100	–	ns
$t_{d(C-Q)}$	TXDi output delay time	–	50	ns
$t_{h(C-Q)}$	TXDi hold time	0	–	ns
$t_{su(D-C)}$	RXDi input setup time	50	–	ns
$t_{h(C-D)}$	RXDi input hold time	90	–	ns

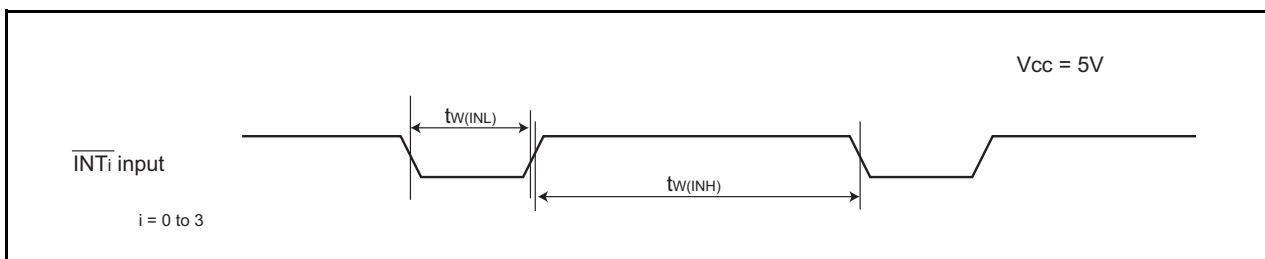
i = 0 or 1

**Figure 20.10 Serial Interface Timing Diagram when Vcc = 5 V****Table 20.19 External Interrupt  $\overline{INTi}$  (i = 0 to 3) 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:

- When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use the  $\overline{INTi}$  input HIGH width to the greater value, either (1/digital filter clock frequency x 3) or the minimum value of standard.
- When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use the  $\overline{INTi}$  input LOW width to the greater value, either (1/digital filter clock frequency x 3) or the minimum value of standard.

**Figure 20.11 External Interrupt  $\overline{INTi}$  Input Timing Diagram when Vcc = 5 V (i = 0 to 3)**

**Table 20.20 Electrical Characteristics (3) [Vcc = 3 V]**

Symbol	Parameter		Condition		Standard			Unit
					Min.	Typ.	Max.	
VOH	Output "H" voltage	Except XOUT	IOH = -1 mA		Vcc - 0.5	-	Vcc	V
		XOUT	Drive capacity HIGH	IOH = -0.1 mA	Vcc - 0.5	-	Vcc	V
			Drive capacity LOW	IOH = -50 μA	Vcc - 0.5	-	Vcc	V
VOL	Output "L" voltage	Except XOUT	IOL = 1 mA		-	-	0.5	V
		XOUT	Drive capacity HIGH	IOL = 0.1 mA	-	-	0.5	V
			Drive capacity LOW	IOL = 50 μA	-	-	0.5	V
VT+-VT-	Hysteresis	$\overline{\text{INT0}}, \overline{\text{INT1}}, \overline{\text{INT2}}, \overline{\text{INT3}}, \text{KI0}, \text{KI1}, \text{KI2}, \text{KI3}, \text{TRAIO}, \text{RXD0}, \text{RXD1}, \text{CLK0}, \text{SSI}, \text{SCL}, \text{SDA}, \text{SSO}$			0.1	0.3	-	V
		$\overline{\text{RESET}}$			0.1	0.4	-	V
I <sub>IH</sub>	Input "H" current		VI = 3 V, Vcc = 3 V		-	-	4.0	μA
I <sub>IL</sub>	Input "L" current		VI = 0 V, Vcc = 3 V		-	-	-4.0	μA
R <sub>PULLUP</sub>	Pull-up resistance		VI = 0 V, Vcc = 3 V		66	160	500	kΩ
R <sub>IXIN</sub>	Feedback resistance	XIN			-	3.0	-	MΩ
V <sub>RAM</sub>	RAM hold voltage		During stop mode		2.0	-	-	V

## NOTE:

- Vcc = 2.7 to 3.3 V at Topr = -40 to 85°C (J version) / -40 to 125°C (K version), f(XIN) = 10 MHz, unless otherwise specified.

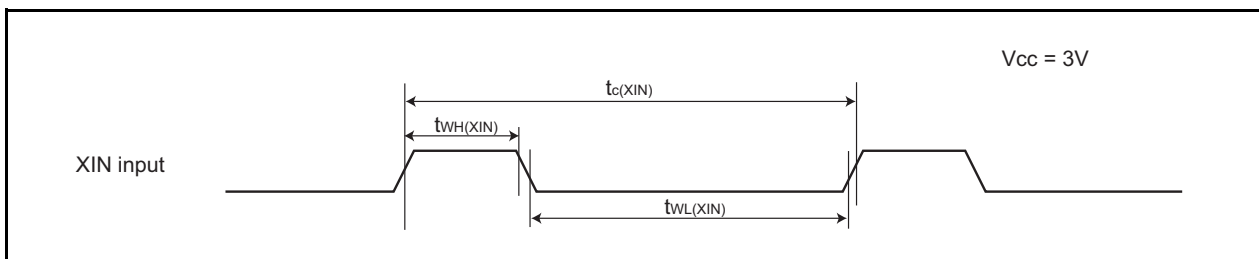


**Table 20.21 Electrical Characteristics (4) [V<sub>CC</sub> = 3 V]  
(Topr = -40 to 85°C (J version) / -40 to 125°C (K version), Unless Otherwise Specified.)**

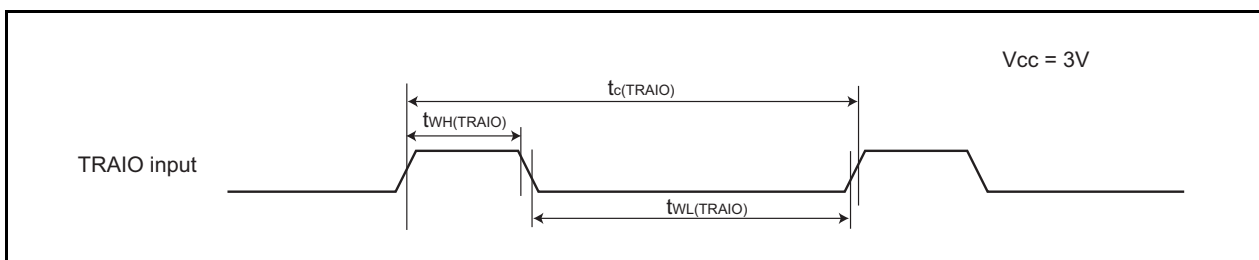
Symbol	Parameter	Condition	Standard			Unit	
			Min.	Typ.	Max.		
I <sub>CC</sub>	Power supply current (V <sub>CC</sub> = 2.7 to 3.3 V) In single-chip mode, the output pins are open and other pins are V <sub>SS</sub>	High-clock mode	XIN = 20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	10.5	21.0	mA
			XIN = 16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	8.3	16.6	mA
			XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	5.3	10.6	mA
			XIN = 20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	4.5	–	mA
			XIN = 16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	3.3	–	mA
			XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	2.3	–	mA
		High-speed on-chip oscillator mode	XIN clock off High-speed on-chip oscillator on f <sub>OCO</sub> = 10 MHz Low-speed on-chip oscillator on = 125 kHz No division	–	5.6	11.2	mA
			XIN clock off High-speed on-chip oscillator on f <sub>OCO</sub> = 10 MHz Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	2.4	–	mA
		Low-speed on-chip oscillator mode	XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8 FMR47 = 1	–	138	276	μA
		Wait mode	XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock operation VCA20 = 0 VCA26 = VCA27 = 0	–	48	96	μA
			XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock off VCA20 = 0 VCA26 = VCA27 = 0	–	35	70	μA
		Stop mode Topr = 25°C	XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA26 = VCA27 = 0	–	0.7	3.0	μA
		Stop mode Topr = 85°C	XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA26 = VCA27 = 0	–	1.1	–	μA
		Stop mode Topr = 125°C	XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA26 = VCA27 = 0	–	3.8	–	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 20.22 XIN Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(XIN)}$	XIN input cycle time	100	–	ns
$t_{WH(XIN)}$	XIN input "H" width	40	–	ns
$t_{WL(XIN)}$	XIN input "L" width	40	–	ns

**Figure 20.12 XIN Input Timing Diagram when  $V_{CC} = 3\text{ V}$** **Table 20.23 TRAIO Input**

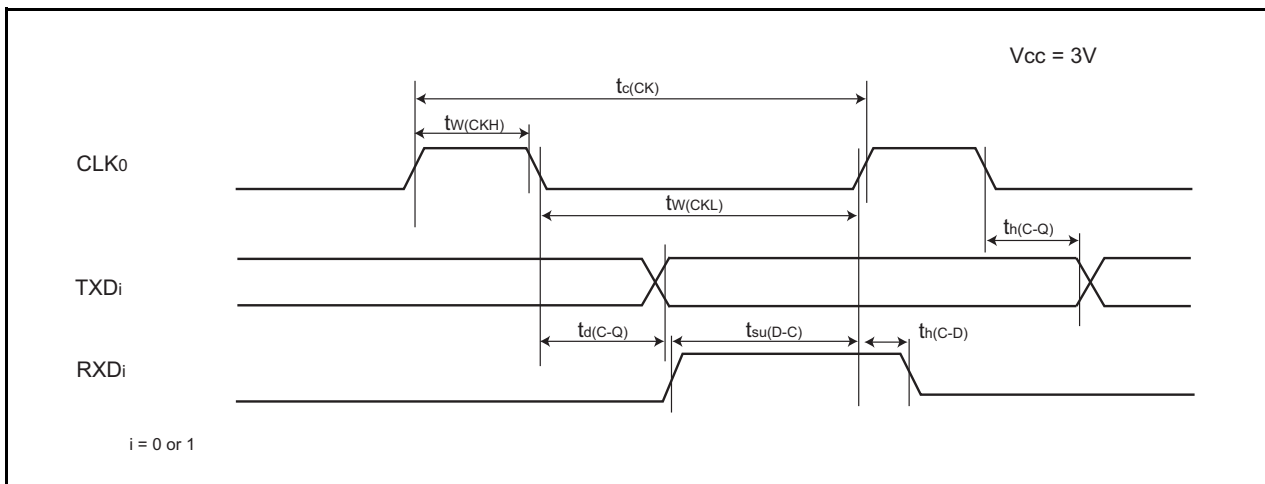
Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TRAIO)}$	TRAIO input Cycle time	300	–	ns
$t_{WH(TRAIO)}$	TRAIO input "H" width	120	–	ns
$t_{WL(TRAIO)}$	TRAIO input "L" width	120	–	ns

**Figure 20.13 TRAIO Input Timing Diagram when  $V_{CC} = 3\text{ V}$**

**Table 20.24 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLK0 input cycle time	300	–	ns
$t_{w(CKH)}$	CLK0 input “H” width	150	–	ns
$t_{w(CKL)}$	CLK0 input “L” width	150	–	ns
$t_{d(C-Q)}$	TXDi output delay time	–	80	ns
$t_{h(C-Q)}$	TXDi hold time	0	–	ns
$t_{su(D-C)}$	RXDi input setup time	70	–	ns
$t_{h(C-D)}$	RXDi input hold time	90	–	ns

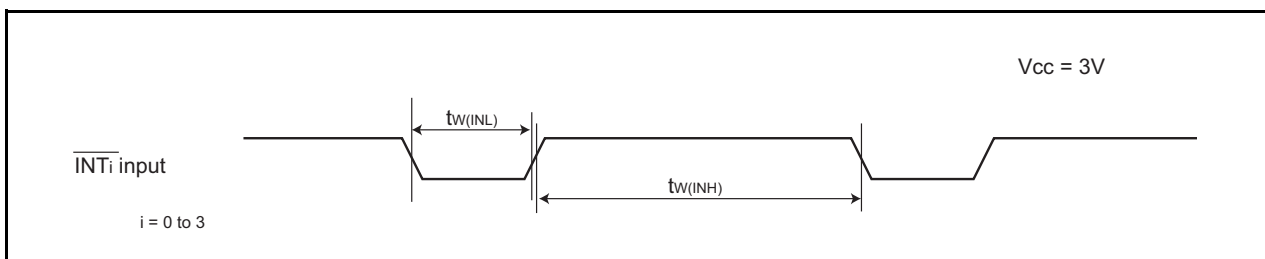
i = 0 or 1

**Figure 20.14 Serial Interface Timing Diagram when Vcc = 3 V****Table 20.25 External Interrupt  $\overline{INTi}$  (i = 0 to 3) 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:

- When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use the  $\overline{INTi}$  input HIGH width to the greater value, either (1/digital filter clock frequency x 3) or the minimum value of standard.
- When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use the  $\overline{INTi}$  input LOW width to the greater value, either (1/digital filter clock frequency x 3) or the minimum value of standard.

**Figure 20.15 External Interrupt  $\overline{INTi}$  Input Timing Diagram when Vcc = 3 V (i = 0 to 3)**

## 21. Usage Notes

### 21.1 Notes on Clock Generation Circuit

#### 21.1.1 Stop Mode

When entering stop mode, set the FMR01 bit to “0” (CPU rewrite mode disabled) and the CM10 bit to “1” (stop mode). An instruction queue pre-reads 4 bytes from the instruction which sets the CM10 bit in the CM1 register to “1” (stop mode) and the program stops. Insert at least 4 NOP instructions following the JMP.B instruction immediately after the instruction which sets the CM10 bit to “1”.

- 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

```

#### 21.1.2 Wait Mode

When entering wait mode, set the FMR01 bit 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.

- Example to execute the WAIT instruction

```

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

```

#### 21.1.3 Oscillation Stop Detection Function

Since the oscillation stop detection function cannot be used if the XIN clock frequency is less than 2 MHz, set the OCD1 to OCD0 bits to 00b.

#### 21.1.4 Oscillation Circuit Constants

Ask the maker of the oscillator to specify the beat oscillation circuit constants on your system.

## 21.2 Notes on Interrupts

### 21.2.1 Reading Address 00000h

Do not read the 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 the address 00000h is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is set to 0. This may cause a problem that the interrupt is canceled, or an unexpected interrupt is generated.

### 21.2.2 SP Setting

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

### 21.2.3 External Interrupt and Key Input Interrupt

Either an “L” level or an “H” level of width shown in the Electrical Characteristics is necessary for the signal input to the  $\overline{\text{INT0}}$  to  $\overline{\text{INT3}}$  pins and  $\overline{\text{KI0}}$  to  $\overline{\text{KI3}}$  pins regardless of the CPU clocks. For details, refer to **Table 20.19 External Interrupt  $\overline{\text{INTi}}$  (i = 0 to 3) Input**, **Table 20.25 External Interrupt  $\overline{\text{INTi}}$  (i = 0 to 3) Input**.

### 21.2.4 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, the changes of interrupt sources include all sources that change the interrupt sources assigned to individual software interrupt numbers, polarities, and timing. Therefore, when a mode change of the peripheral functions involves interrupt sources, edge polarities, and timing, Set the IR bit to 0 (no interrupt requested) after the change. Refer to each peripheral function for the interrupts caused by the peripheral functions.

Figure 21.1 shows an Example of Procedure for Changing Interrupt Sources.

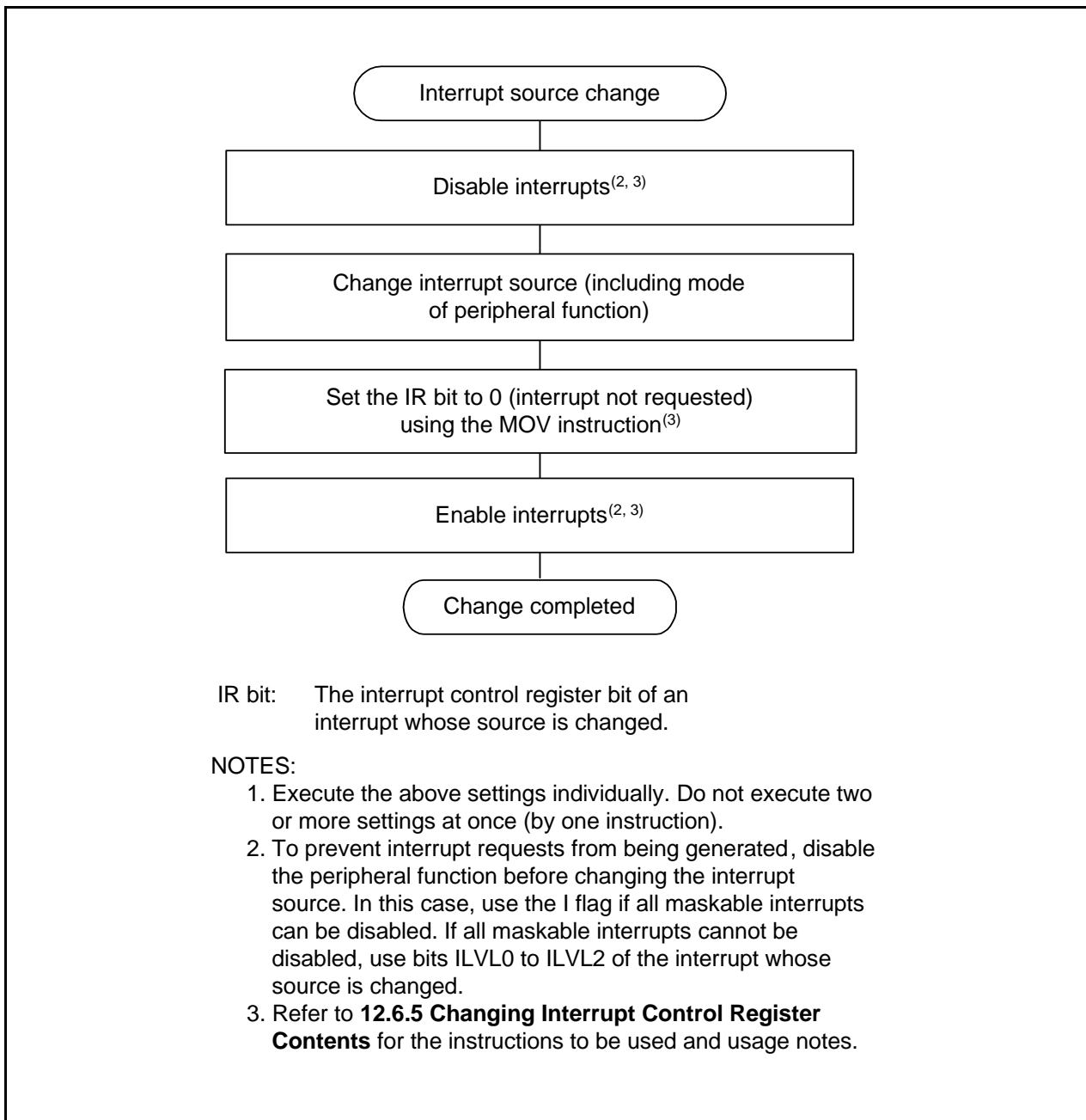


Figure 21.1 Example of Procedure for Changing Interrupt Sources

### 21.2.5 Changing Interrupt Control Register Contents

- (a) Each interrupt control register can only be changed while interrupt requests corresponding to that register are not generated. If interrupt requests may be generated, disable the interrupts before changing the interrupt control register.
- (b) When changing any interrupt control register after disabling interrupts, be careful with the instructions to be used.  
**When changing any bit other than IR bit**  
 If an interrupt request corresponding to that 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.  
 Instructions to use: AND, OR, BCLR, BSET  
**When 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 to be 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 according to the following sample programs. Refer to (b) for the change of interrupt control registers in the sample programs.

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

**Example 1: Use NOP instructions to prevent I flag 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 have FSET instruction wait**

```
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
```

## 21.3 Notes on Timers

### 21.3.1 Notes on Timer RA

- Timer RA stops counting after reset. Set the value to timer RA and timer RA prescaler before the count starts.
- Even if the prescaler and timer RA is read out in 16-bit units, these registers are read by 1 byte in the MCU. Consequently, the timer value may be updated during the period these two registers are being read.
- In pulse width measurement mode and pulse period measurement mode, the TEDGF and TUNDF bits in the TRACR register can be set to 0 by writing 0 to these bits by a program. However, these bits remain unchanged when 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 executed. At the time, 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 width measurement mode and pulse period measurement mode from other mode, the contents of the TEDGF and TUNDF bits are indeterminate. Write 0 to the TEDGF and TUNDF bits before the count starts.
- The TEDGF bit may be set to 1 by timer RA prescaler underflow which is generated for the first time since the count starts.
- When using the pulse period measurement mode, leave two periods or more of timer RA prescaler immediately after count starts, and 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 stops.

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 retains 1 for 0 to 1 cycle of the count source after setting the TSTART bit to 0 (count stops) while the count is performing. 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, 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.



### 21.3.2 Notes on Timer RB

- Timer RB stops counting after reset. Set the value to timer RB and timer RB prescaler before the count starts.
- Even if the prescaler and timer RB is read out in 16-bit units, these registers are read by 1 byte in the MCU. Consequently, the timer value may be updated during the period 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, read the timer count value in programmable one-shot generation mode and programmable wait one-shot generation mode before the timer stops.
- The TCSTF bit retains 0 (count stops) for 1 to 2 cycles of the count source after setting the TSTART bit to 1 (count starts) while the count stops.

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

The TCSTF bit retains 1 for 1 to 2 cycles of the count source after setting the TSTART bit to 0 (count stops) while the count is performing. 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, TRBPRE, TRBSC, 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.

#### 21.3.2.1 Timer mode

The following workaround should be performed in timer mode.

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.

### 21.3.2.2 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 21.2 and 21.3.

The following shows the detailed workaround examples.

- Workaround example (a):  
As shown in Figure 21.2, write to registers TRBSC and TRBPR in the timer RB interrupt routine. These write operations must be completed by the beginning of period A.

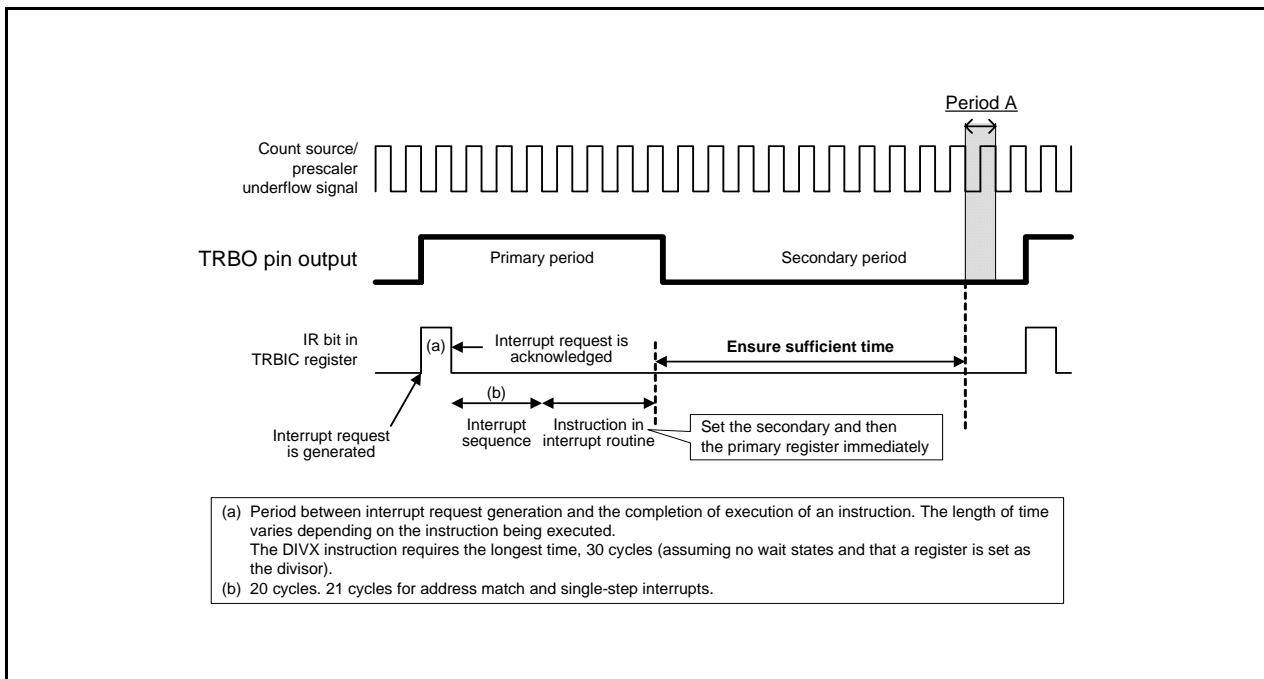
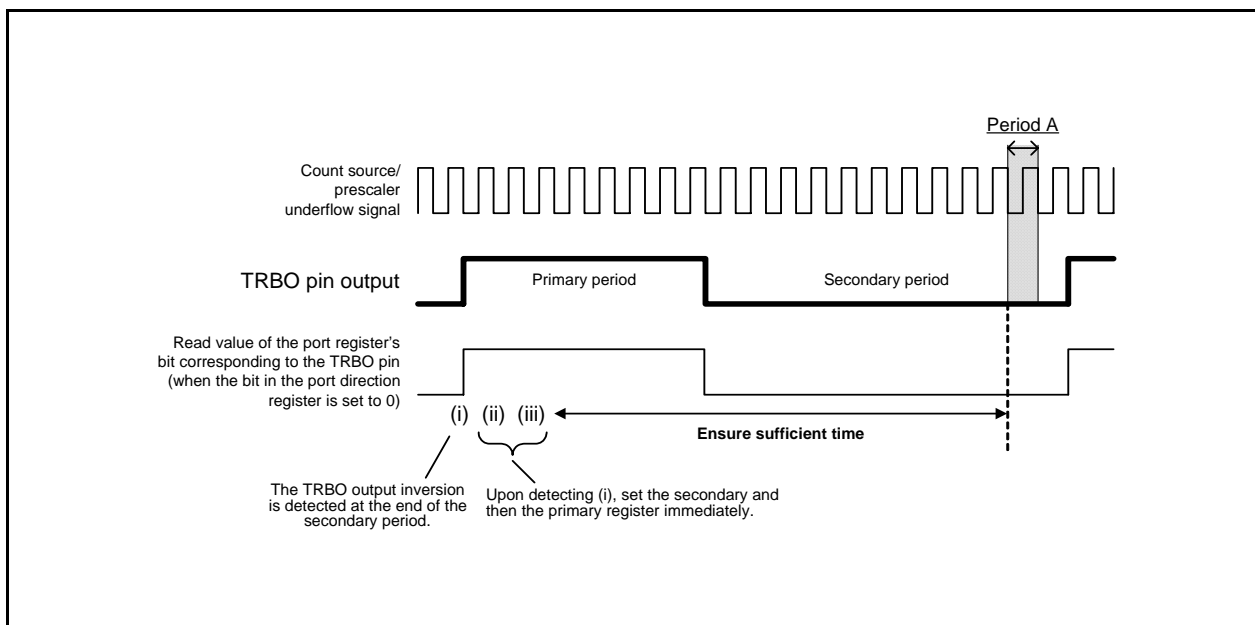


Figure 21.2 Workaround Example (a) When Timer RB Interrupt is Used

- Workaround example (b):

As shown in Figure 21.3 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 21.3 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.

### 21.3.2.3 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.

### 21.3.2.4 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.

### 21.3.3 Notes on Timer RD

#### 21.3.3.1 TRDSTR Register

- Set the TRDSTR register using the MOV instruction.
- When the CSELi (i = 0 or 1) is set to 0 (the count stops at compare match of registers TRDi and TRDGRAi), the count does not stop and the TSTARTi bit remains unchanged even if 0 (count stops) is written to the TSTARTi bit.  
Therefore, set the TSTARTi bit to 0 to change other bits without changing the TSTARTi bit when the CSELi bit is set to 0.  
To stop counting by a program, set the TSTARTi bit to 0 after setting the CSELi bit to 1. Although the CSELi bit is set to 1 and the TSTARTi bit is set to 0 at the same time (with 1 instruction), the count cannot be stopped.
- Table 21.1 lists the TRDIOj (j = A, B, C, or D) Pin Output Level when Count Stops to use the TRDIOj pin with the timer RD output.

**Table 21.1 TRDIOj (j = A, B, C, or D) Pin Output Level when Count Stops**

Count Stop	TRDIOj Pin Output when Count Stops
When the CSELi bit is set to 1, set the TSTARTi bit to 0 and the count stops.	Hold the output level immediately before the count stops.
When the CSELi bit is set to 0, the count stops at compare match of registers TRDi and TRDGRAi.	Hold the output level after output changes by compare match.

#### 21.3.3.2 TRDi Register (i = 0 or 1)

- When writing the value to the TRDi register by a program while the TSTARTi bit in the TRDSTR register is set to 1 (count starts), avoid to overlap with the timing to set the TRDi register to 0000h, and then write. When the timing to set the TRDi register to 0000h overlaps with the timing to write the value to the TRDi register, the value is not written and the TRDi register is set to 0000h.  
These precautions are applicable when selecting the following by the CCLR2 to CCLR0 bits in the TRDCRi register.
  - 001b (clear by the TRDi register at the compare match with the TRDGRAi register)
  - 010b (clear by the TRDi register at the compare match with the TRDGRBi register.)
  - 011b (synchronous clear)
  - 101b (clear by the TRDi register at the compare match with the TRDGRCi register.)
  - 110b (clear by the TRDi register at the compare match with the TRDGRDi register.)
- When writing the value to the TRDi register and continuously reading the same register, the value before writing may be read. In this case, execute the JMP.B instruction between the writing and reading.
 

```

Program Example      MOV.W      #XXXXh, TRD0      ;Writing
                    JMP.B      L1              ;JMP.B
                    L1:      MOV.W      TRD0,DATA    ;Reading
      
```

#### 21.3.3.3 TRDSRi Register (i = 0 or 1)

When writing the value to the TRDSRi register and continuously reading the same register, the value before writing may be read. In this case, execute the JMP.B instruction between the writing and reading.

```

Program Example      MOV.B      #XXh, TRDSR0    ;Writing
                    JMP.B      L1              ;JMP.B
                    L1:      MOV.B      TRDSR0,DATA  ;Reading
      
```

### 21.3.3.4 Count Source Switch

- When switching the count source, switch it after the count stops.

Change procedure

- (1) Set the TSTART<sub>i</sub> (i = 0 or 1) bit in the TRDSTR register to 0 (count stops).
- (2) Change the TCK2 to TCK0 bits in the TRDCR<sub>i</sub> register.

- When changing the count source from fOCO40M to the other and stopping fOCO40M, wait 2 cycles or more of f1 after setting the clock switch, and then stop fOCO40M.

Change procedure

- (1) Set the TSTART<sub>i</sub> (i = 0 or 1) bit in the TRDSTR register to 0 (count stops).
- (2) Change the TCK2 to TCK0 bits in the TRDCR<sub>i</sub> register.
- (3) Wait 2 cycles or more of f1.
- (4) Set the FRA00 bit in the FRA0 register to 0 (high-speed on-chip oscillator stops).

### 21.3.3.5 Input Capture Function

- Set the pulse width of input capture signal to 3 cycles or more of the Timer RD operation clock. (Refer to **Table 14.11 Timer RD Operation Clocks**.)
- The value in the TRD<sub>i</sub> register is transferred to the TRDGR<sub>ji</sub> register after 2 to 3 cycles of the Timer RD operation clock since the input capture signal is applied to the TRDIO<sub>ji</sub> pin (i = 0 or 1, j = either A, B, C or D) (no digital filter).

### 21.3.3.6 Reset Synchronous PWM Mode

- When reset synchronous PWM mode is used for motor control, use it with OLS0 = OLS1.
- Set to reset synchronous PWM mode in the following procedure:

Change procedure

- (1) Set the TSTART0 bit in the TRDSTR register to 0 (count stops).
- (2) Set the CMD1 to CMD0 bits in the TRDFCR register to 00b (timer mode, PWM mode, and PWM3 mode).
- (3) Set the CMD1 to CMD0 bits to 01b (reset synchronous PWM mode).
- (4) Set the registers associated with other Timer RD again.

### 21.3.3.7 Complementary PWM Mode

- When complementary PWM mode is used for motor control, use it with OLS0 = OLS1.
- Change the CMD1 to CMD0 bits in the TRDFCR register in the following procedure.

Change procedure: When setting to complementary PWM mode (including re-set), or changing the transfer timing from the buffer register to the general register in complementary PWM mode.

- (1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
- (2) Set the CMD1 to CMD0 bits in the TRDFCR register to 00b (timer mode, PWM mode, and PWM3 mode)
- (3) Set the DMD1 to CMD0 bits to 10b or 11b (complementary PWM mode).
- (4) Set the registers associated with other Timer RD again.

Change procedure: When stopping complementary PWM mode

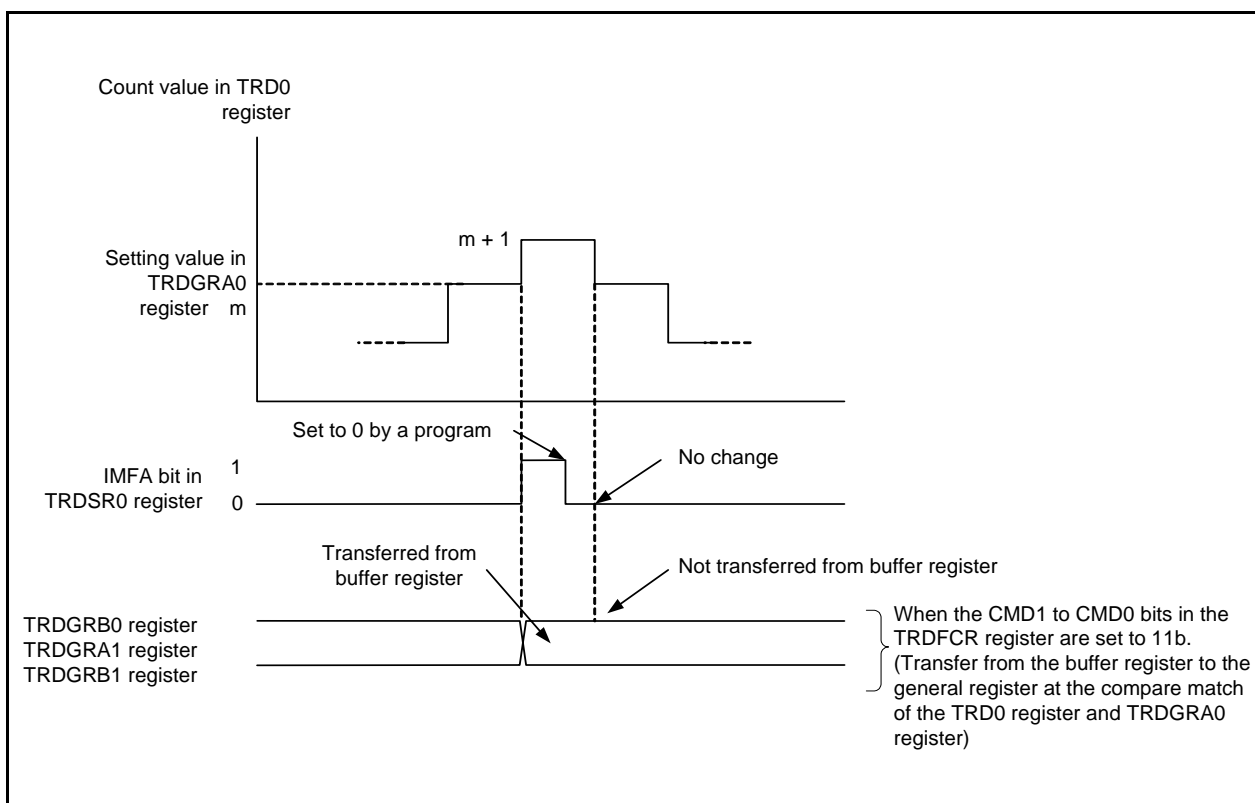
- (1) Set both the TSTART0 and CSEL1 bits in the TRDSTR register to 0 (count stops).
- (2) Set the CMD1 to CMD bits to 00b (other than reset synchronous PWM mode, complementary PWM mode)

- Do not write to the TRDGRA0, TRDGRB0, TRDGRA1 and TRDGRB1 registers during operation. When changing the PWM waveform, transfer the value written to the TRDGRD0, TRDGRC1 and TRDGRD1 registers to the TRDGRB0, TRDGRA1 and TRDGRB1 registers using the buffer operation. However, to write data to the TRDGRD0, TRDGRC1, or TRDGRD1 register, set bits BFD0, BFC1, and BFD1 to 0 (general register). After this, bits BFD0, BFC1, and BFD1 may be set to 1 (buffer register). The PWM period cannot be changed.

- When the value in the TRDGRA0 register is assumed as  $m$ , the TRD0 register counts order of  $m - 1$ ,  $m$ ,  $m + 1$ ,  $m$ ,  $m - 1$  when changing from increment to decrement.

When changing from  $m$  to  $m + 1$ , the IMFA bit is set to 1. Also, the CMD1 to CMD0 bits in the TRDFCR register are set to 11b (complementary PWM mode, buffer data transferred by the compare match in the TRD0 and TRDGRA0 registers), the content in the buffer register (TRDGRD0, TRDGRC1, TRDGRD1) is transferred to the general register (TRDGRB0, TRDGRA1, TRDGRB1).

For the order of  $m + 1$ ,  $m$ ,  $m - 1$  operation, the IMFA bit remains unchanged and data are not transferred to the register such as the TRDGRA0 register.



**Figure 21.4 Operation at Compare Match between Registers TRD0 and TRDGRA0 in Complementary PWM Mode**

- The TRD1 register counts the order of 1, 0, FFFFh, 0, 1 when changing from decrement to increment. The UDF bit is set to 1 by the order of 1, 0, FFFFh operation. Also, when the CMD1 to CMD0 bits in the TRDFCR register are set to 10b (complementary PWM mode, buffer data transferred by the underflow in the TRD1 register), the content in the buffer register (TRDGRD0, TRDGRC1, TRDGRD1) is transferred to the general register (TRDGRB0, TRDGRA1, TRDGRB1). For the order of FFFFh, 0, 1 operation, data are not transferred to the register such as the TRDGRB0 register. Also, at this time, the OVF bit remains unchanged.

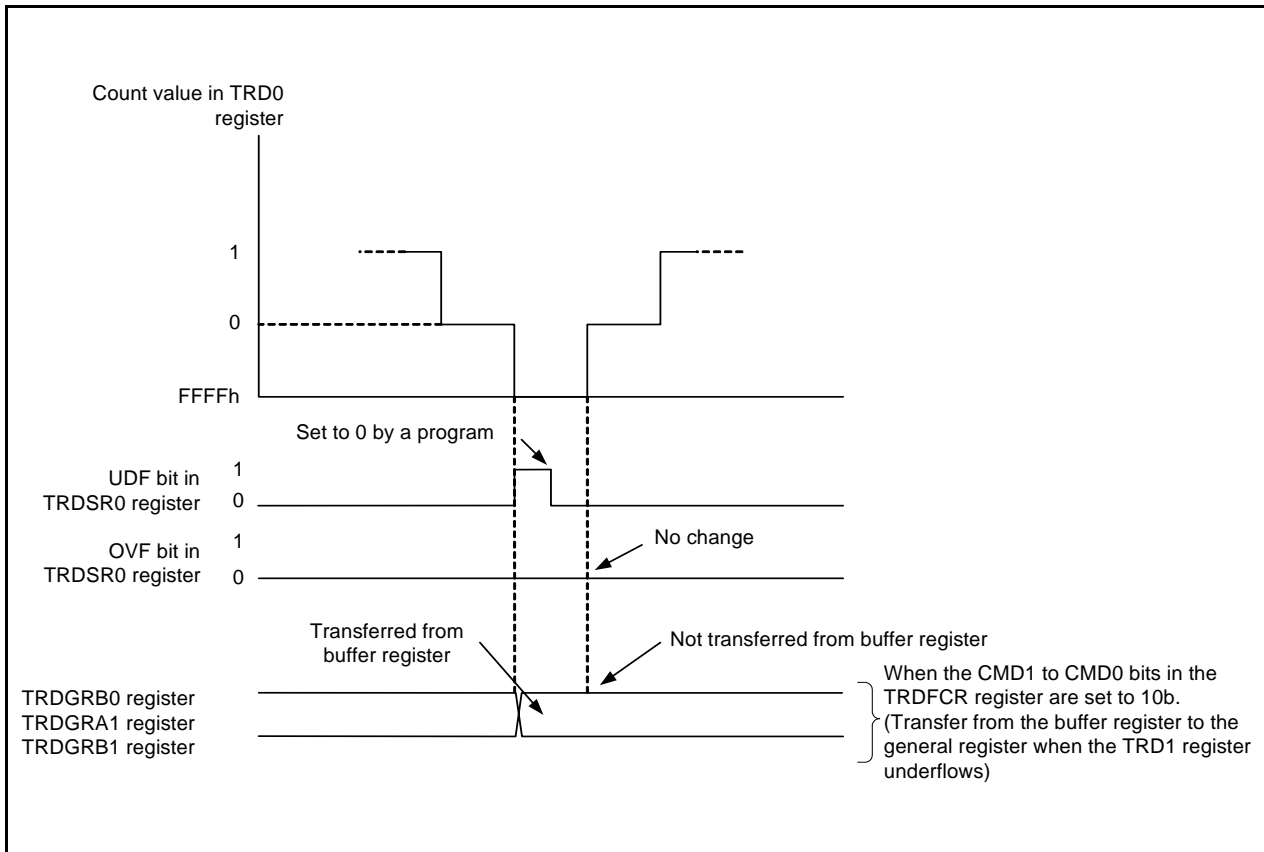


Figure 21.5 Operation When TRD1 Register Underflows in Complementary PWM Mode

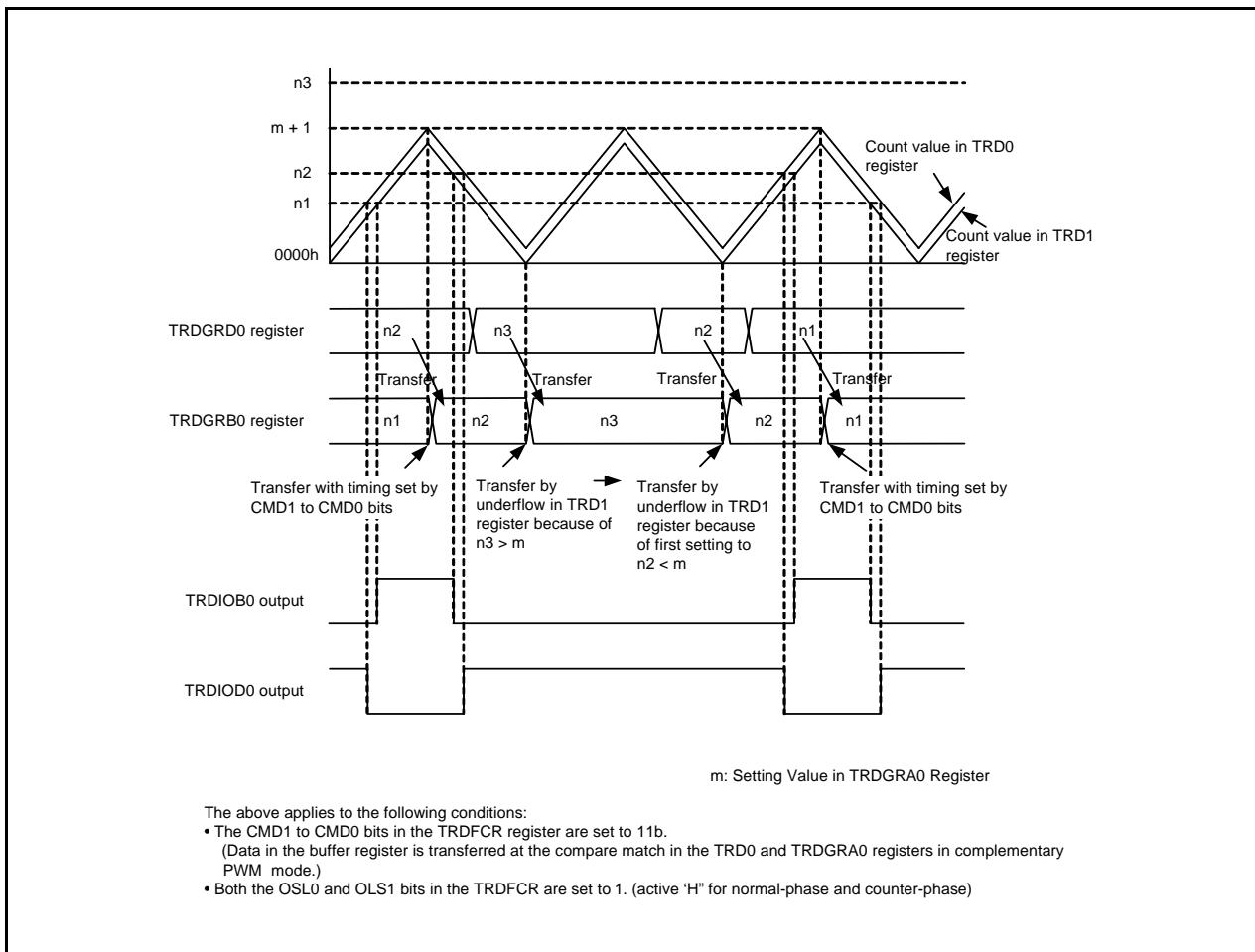


- Select with the CMD1 to CMD0 bits for the data transfer timing from the buffer register to the general register. However, transfer with the following timing in spite of the value of the CMD1 to CMD0 bits for the following cases:

Value in buffer register  $\geq$  Value in TRDGRA0 register:

Transfer at the underflow in the TRD1 register.

And then, when setting the buffer register to 0001h or above and the smaller value than the one in the TRDGRA0 register, and the TRD1 register underflows in the first time after setting, the value is transferred to the general register. After that, transfer the value with the timing selected by the CMD1 to CMD0 bits.

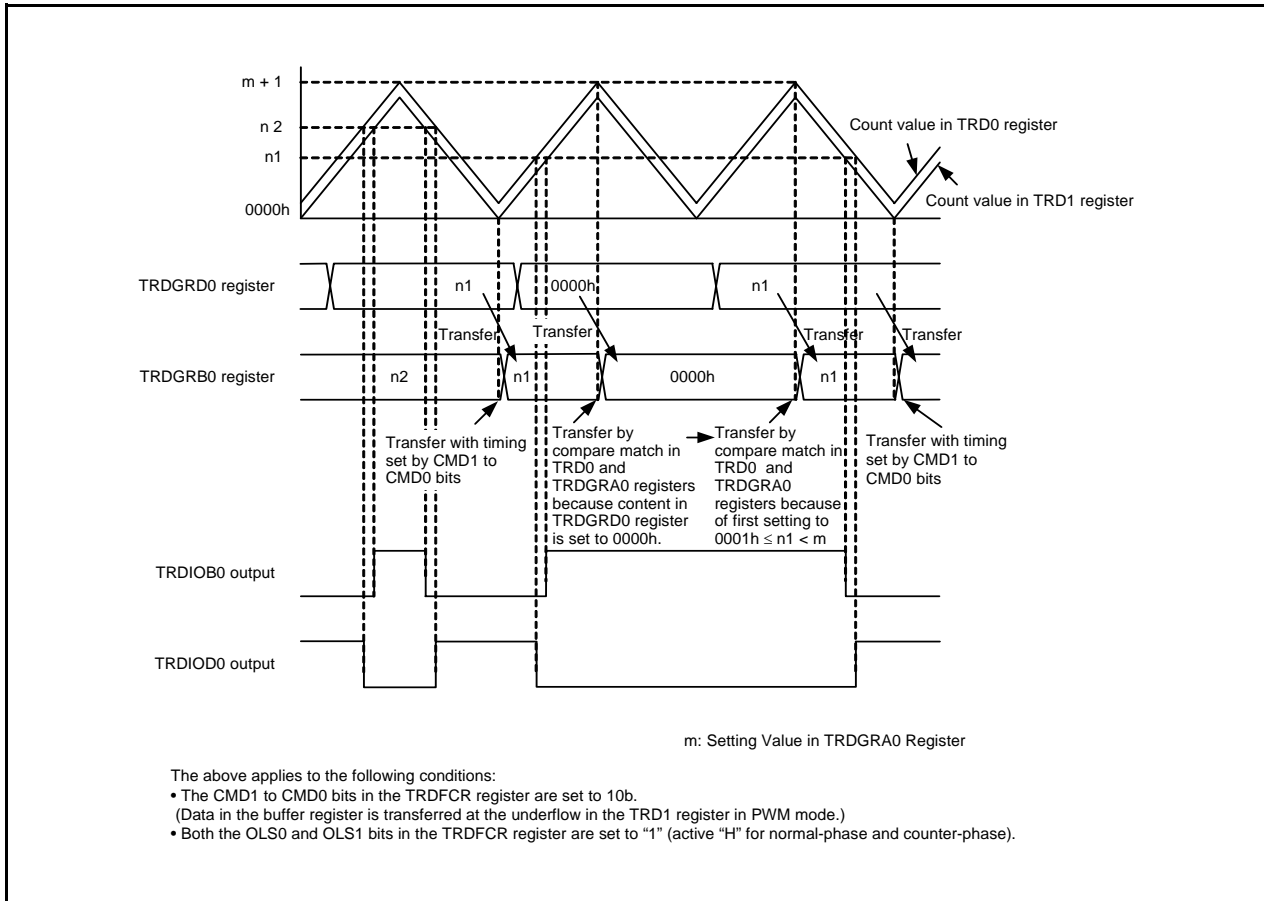


**Figure 21.6** Operation When Value in Buffer Register  $\geq$  Value in TRDGRA0 Register in Complementary PWM Mode

When the value in the buffer register is set to 0000h:

Transfer by the compare match in the TRD0 and TRDGRA0 registers.

And then, when setting the buffer register to 0001h or above and the smaller value than the one in the TRDGRA0 register, and the compare match in the TRD0 and TRDGRA0 registers in the first time after setting, the value is transferred to the general register. After that, transfer the value with the timing selected by the CMD1 to CMD0 bits.



**Figure 21.7 Operation When Value in Buffer Register Is Set to 0000h in Complementary PWM Mode**

### 21.3.3.8 Count Source fOCO40M

The count source fOCO40M can be used with supply voltage  $VCC = 3.0$  to  $5.5$  V. For supply voltage other than that, do not set bits TCK2 to TCK0 in registers TRDCR0 and TRDCR to 110b (select fOCO40M as the count source).

## 21.3.4 Notes on Timer RE

### 21.3.4.1 Starting and Stopping Count

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

Timer RE starts counting when setting the TSTART bit to 1 (count starts) and the TCSTF bit is set to 1 (count starts). It takes the time for 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, TRECRC1, TRECRC2, TRECSR

### 21.3.4.2 Register Setting

Write to the following registers or bits while timer RE stops.

- TRESEC and TRECRC2 registers
- The INT bit in TRECRC1 register
- RCS0 to RCS2 bits in TRECSR register

The state while Timer RE stops is indicated as the state where the TSTART and TCSTF bits in the TRECRC1 register are set to 0 (timer RE stops).

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

## 21.4 Notes on Serial Interface

- When reading data from the UiRB (i = 0 or 1) register even in the clock asynchronous serial I/O mode or in the clock synchronous serial I/O mode. Ensure to read data in 16-bit unit. When the high-order byte of the UiRB register is read, the PER and FER bits 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 high-order byte first, then 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
```

## 21.5 Clock Synchronous Serial Interface

### 21.5.1 Notes on Clock Synchronous Serial I/O with Chip Select

Set the IICSEL bit in the PMR register to 0 (select clock synchronous serial I/O with chip select function) to use the clock synchronous serial I/O with chip select.

### 21.5.2 Notes on I<sup>2</sup>C Bus Interface

Set the IICSEL bit in the PMR register to 1 (select I<sup>2</sup>C bus interface function) to use I<sup>2</sup>C bus interface.

#### 21.5.2.1 Multimaster Operation

The following actions must be performed to use the I<sup>2</sup>C bus interface in multimaster operation.

- Transfer rate  
Set the transfer rate by 1/1.8 or faster than the fastest rate of the other masters. For example, if the fastest transfer rate of the other masters is set to 400 kbps, the I<sup>2</sup>C-bus transfer rate in this MCU should be set to 223 kbps (= 400/1.8) or more.
- Bits MST and TRS in the ICCR1 register setting
  - (a) Use the MOV instruction to set bits MST and TRS.
  - (b) When arbitration is lost, confirm the contents of bits MST and TRS. If the contents are other than the MST bit set to 0 and the TRS bit set to 0 (slave receive mode), set the MST bit to 0 and the TRS bit to 0 again.

#### 21.5.2.2 Master Receive Mode

Either of the following actions must be performed to use the I<sup>2</sup>C bus interface in master receive mode.

- (a) In master receive mode while the RDRF bit in the ICSR register is set to 1, read the ICDRR register before the rising edge of the 8th clock.
- (b) In master receive mode, set the RCVD bit in the ICCR1 register to 1 (disables the next receive operation) to perform 1-byte communications.

## 21.6 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 respect to a Synch Break detection interrupt as the starting point.

## 21.7 Notes on A/D Converter

- Write to each bit (other than bit 6) in the ADCON0 register, each bit in the ADCON1 register, or the SMP bit in the ADCON2 register when the A/D conversion stops (before a trigger occurs).  
When the VCUT bit in the ADCON1 register is changed from 0 (VREF not connected) to 1 (VREF connected), wait for at least 1  $\mu$ s or longer before the A/D conversion starts.
- When changing A/D operating mode, select an analog input pin again.
- When using in one-shot mode. Ensure that the A/D conversion is completed and read the AD register. The IR bit in the ADIC register or the ADST bit in the ADCON0 register can determine whether the A/D conversion is completed.
- When using the repeat mode, select the frequency of the A/D converter operating clock  $\phi$ AD or more for the CPU clock during A/D conversion.  
Do not select the fOCO-F for the  $\phi$ AD.
- If setting the ADST bit in the ADCON0 register to 0 (A/D conversion stops) by a program and the A/D conversion is forcibly terminated during the A/D conversion operation, the conversion result of the A/D converter will be indeterminate. If the ADST bit is set to 0 by a program, do not use the value of AD register.
- Connect 0.1  $\mu$ F capacitor between the P4\_2/VREF pin and AVSS pin.
- Do not enter stop mode during A/D conversion.
- Do not enter wait mode when the CM02 bit in the CM0 register is set to 1 (peripheral function clock stops in wait mode) during A/D conversion.

## 21.8 Notes on Flash Memory

### 21.8.1 CPU Rewrite Mode

#### 21.8.1.1 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 the CM16 to CM17 bits in the CM1 register. This usage note is not needed for EW1 mode.

#### 21.8.1.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because the flash memory internal data is referenced: UND, INTO, and BRK instructions.

#### 21.8.1.3 Interrupts

Table 21.2 lists the EW0 Mode Interrupts and Table 21.3 lists the EW1 Mode Interrupts.

Table 21.2 EW0 Mode Interrupts

Mode	Status	When Maskable Interrupt Request is Acknowledged	When Watchdog Timer, Oscillation Stop Detection and Voltage Monitor 2 Interrupt Request are Acknowledged
EW0	During automatic erasing	Any interrupt can be used by allocating a vector to RAM	Once an interrupt request is acknowledged, the auto-programming or auto-erasing is forcibly stopped immediately and resets the flash memory. An interrupt process starts after the fixed period and the flash memory restarts. Since the block during the auto-erasing or the address during the auto-programming is forcibly stopped, the normal value may not be read. Execute the auto-erasing again and ensure the auto-erasing is completed normally. Since the watchdog timer does not stop during the command operation, the interrupt request may be generated. Reset the watchdog timer regularly.
	Automatic writing		

NOTES:

1. Do not use the address match interrupt while the command is executed because the vector of the address match interrupt is allocated on ROM.
2. Do not use the non-maskable interrupt while block 0 is automatically erased because the fixed vector is allocated block 0.



Table 21.3 EW1 Mode Interrupts

Mode	Status	When Maskable Interrupt Request is Acknowledged	When Watchdog Timer, Oscillation Stop Detection and Voltage Monitor 2 Interrupt Request are Acknowledged
EW1	During automatic erasing (erase- suspend function is enabled)	The auto-erasing is suspended after td(SR-SUS) and the interrupt process is executed. The auto-erasing can be restarted by setting the FMR41 bit in the FMR4 register to 0 (erase restart) after the interrupt process completes.	Once an interrupt request is acknowledged, the auto-programming or auto-erasing is forcibly stopped immediately and resets the flash memory. An interrupt process starts after the fixed period and the flash memory restarts. Since the block during the auto-erasing or the address during the auto-programming is forcibly stopped, the normal value may not be read. Execute the auto-erasing again and ensure the auto-erasing is completed normally.  Since the watchdog timer does not stop during the command operation, the interrupt request may be generated. Reset the watchdog timer regularly using the erase-suspend function.
	During automatic erasing (erase- suspend function is disabled)	The auto-erasing has a priority and the interrupt request acknowledgement is waited. The interrupt process is executed after the auto-erasing completes.	
	During automatic programming (program suspend function enabled)	The auto-programming is suspended after td(SR-SUS) and the interrupt process is executed. The auto-programming can be restarted by setting the FMR42 bit in the FMR4 register to 0 (program restart) after the interrupt process completes.	
	Auto programming (program suspend function disabled)	The auto-programming has a priority and the interrupt request acknowledgement is waited. The interrupt process is executed after the auto-programming completes.	

## NOTES:

1. Do not use the address match interrupt while the command is executed because the vector of the address match interrupt is allocated on ROM.
2. Do not use the non-maskable interrupt while block 0 is automatically erased because the fixed vector is allocated block 0.

**21.8.1.4 How to Access**

Write 0 to the corresponding bits before writing 1 when setting the FMR01, FMR02, or FMR11 bit to 1. Do not generate an interrupt between writing 0 and 1.

**21.8.1.5 Rewriting User ROM Area**

In EW0 mode, if the power supply voltage drops while rewriting any block in which the rewrite control program is stored, the flash memory may not be able to be rewritten because the rewrite control program cannot be rewritten correctly. In this case, use standard serial I/O mode.

**21.8.1.6 Program**

Do not write additions to the already programmed address.

### **21.8.1.7 Entering Stop Mode or Wait Mode**

Do not enter stop mode or wait mode during erase-suspend.

## 21.9 Notes on Noise

### 21.9.1 Inserting a Bypass Capacitor between VCC and VSS Pins as a Countermeasure against Noise and Latch-up

Connect the bypass capacitor (at least 0.1  $\mu$ F) using the shortest and thickest as possible.

### 21.9.2 Countermeasures against Noise Error of Port Control Registers

During severe noise testing, mainly power supply system noise, and introduction of external noise, the data of port related registers may be changed.

As a firmware countermeasure, it is recommended to periodically reset the port registers, port direction registers and pull-up control registers. However, examine fully before introducing the reset routine as conflicts may be created between this reset routine and interrupt routines.

## 22. Notes on On-Chip Debugger

When using the on-chip debugger to develop the R8C/20 and R8C/21 Groups program and debug, pay the following attention.

- (1) Do not access the related UART1 registers.
- (2) 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.
- (3) Do not set the address match interrupt (the registers of AIER, RMAD0, RMAD1 and the fixed vector tables) in a user system.
- (4) Do not use the BRK instruction in a user system.

Connecting and using the on-chip debugger has some peculiar restrictions. Refer to each on-chip debugger manual for on-chip debugger details.

## 23. Notes on Emulator Debugger

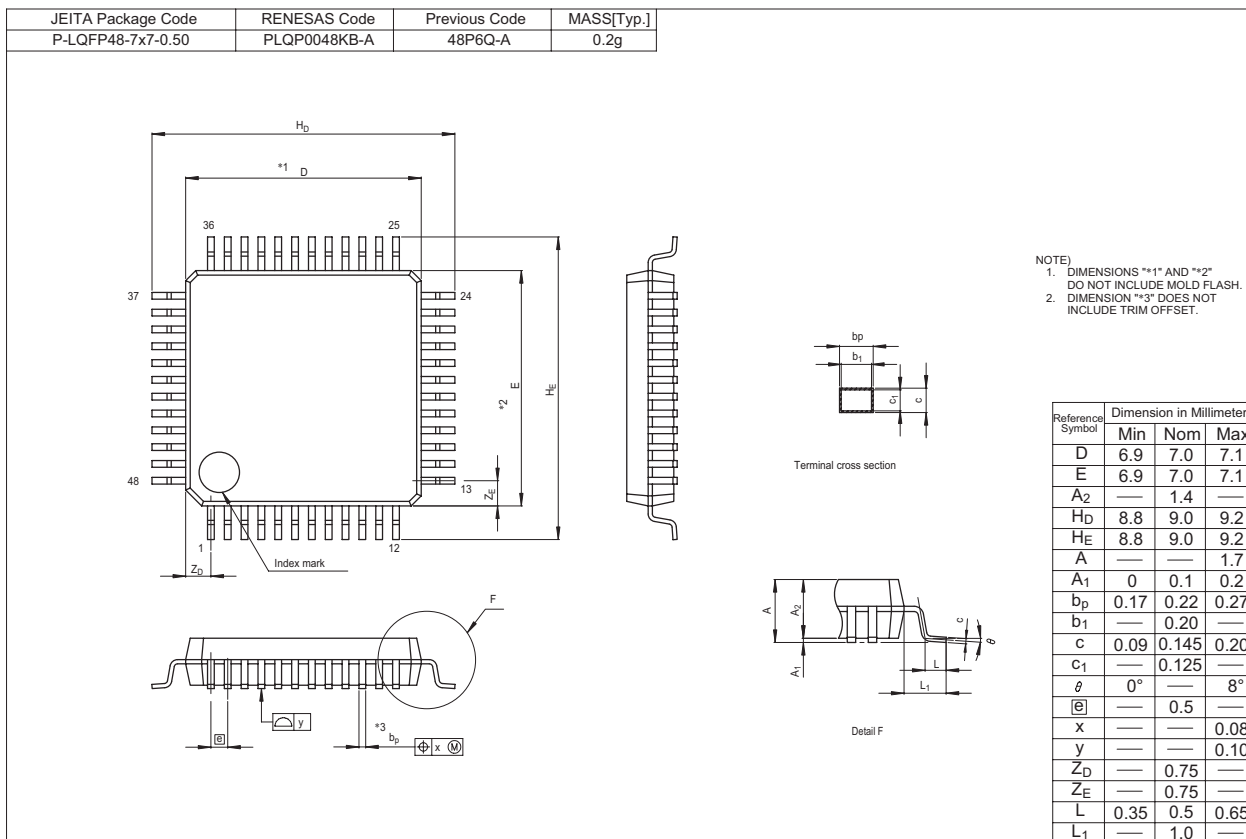
When using the emulator debugger to develop the R8C/20 and R8C/21 Groups program and debug, pay the following attention.

- (1) Do not use the following flash memory areas because these areas are used for the emulator debugger. When debugging of these areas, intensive evaluation on the real chip is required.  
ROM 128 KB Product (R5F2120CJFP, R5F2120CKFP, R5F2121CJFP, R5F2121CKFP) addresses 20000h to 23FFFh

Connecting and using the emulator debugger has some peculiar restrictions. Refer to each emulator debugger manual for emulator debugger details.

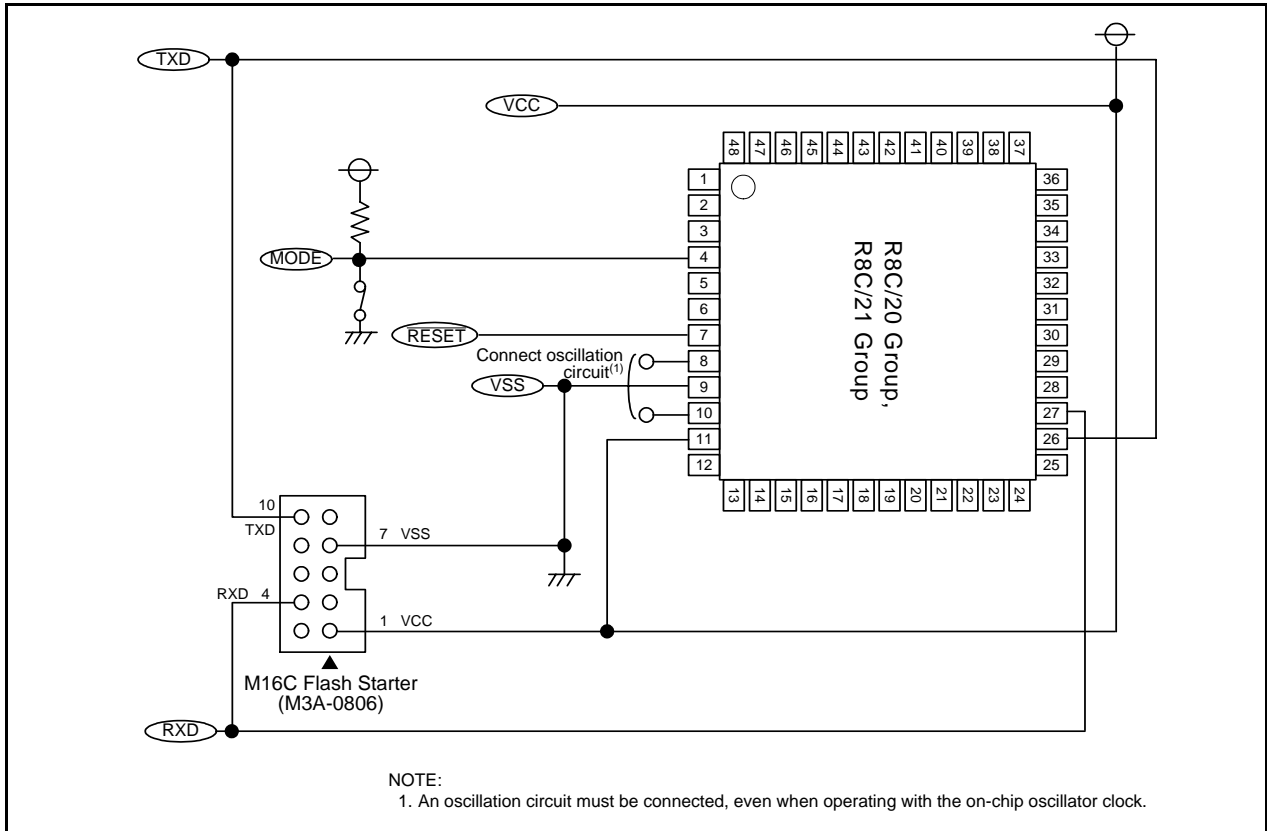
## Appendix 1. Package Dimensions

Diagrams showing the latest package dimensions and mounting information are available in the “Packages” section of the Renesas Technology website.

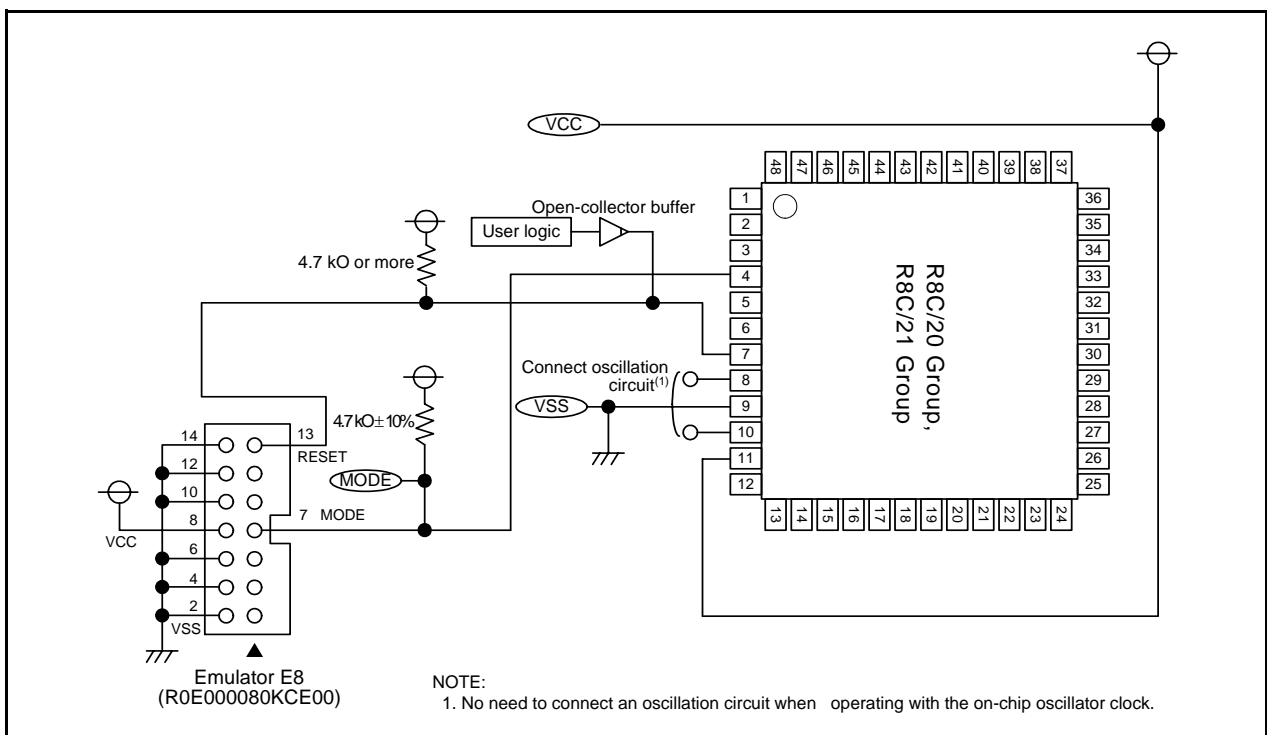


## Appendix 2. Connection Examples between Serial Writer and On-Chip Debugging Emulator

Appendix Figure 2.1 shows the Connection Example with M16C Flash Starter (M3A-0806) and Appendix Figure 2.2 shows the Connection Example with E8 Emulator (R0E000080KCE00).



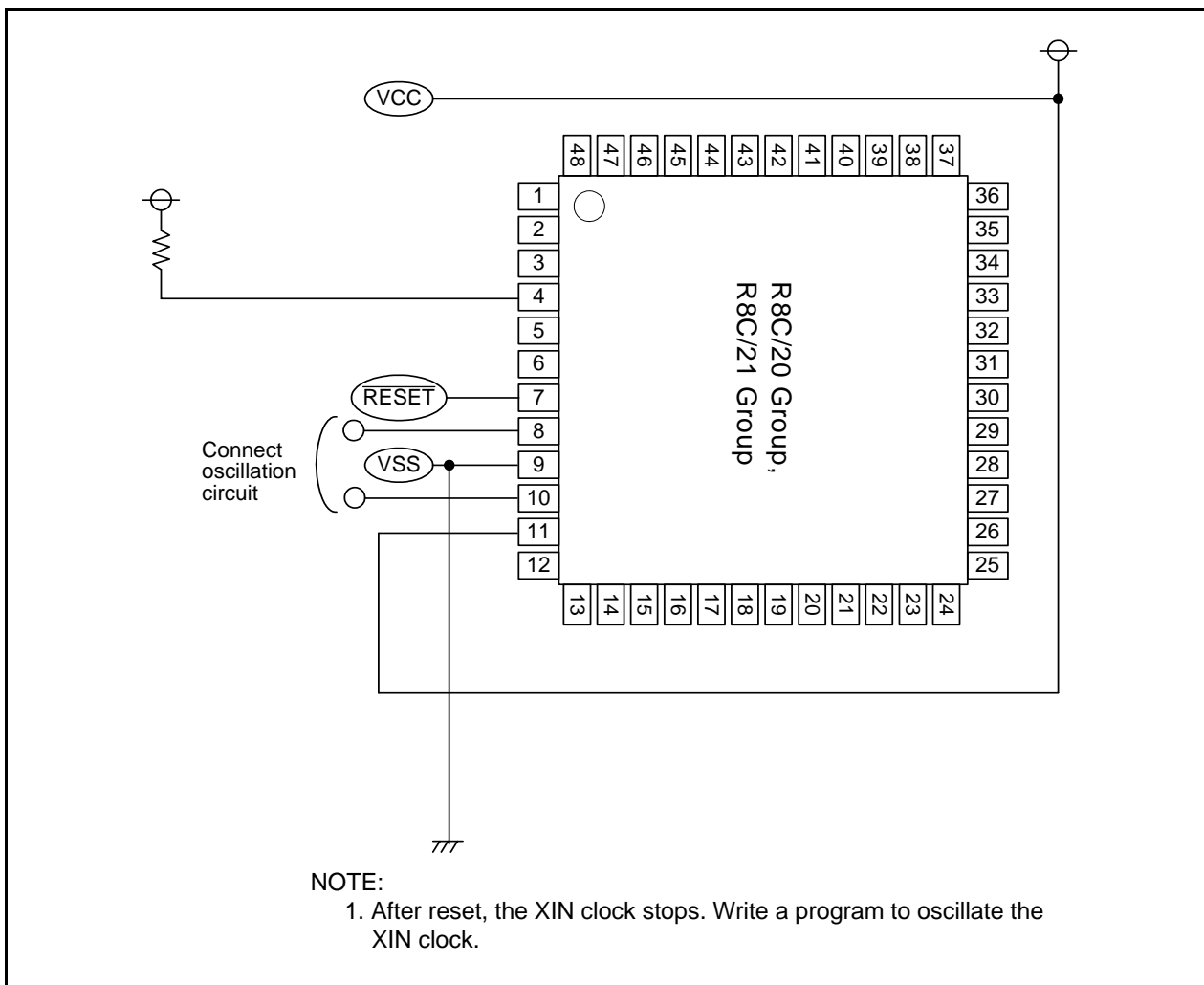
Appendix Figure 2.1 Connection Example with M16C Flash Starter (M3A-0806)



Appendix Figure 2.2 Connection Example with E8 Emulator (R0E000080KCE00)

### Appendix 3. Example of Oscillation Evaluation Circuit

Appendix Figure 3.1 shows the Example of Oscillation Evaluation Circuit.



Appendix Figure 3.1 Example of Oscillation Evaluation Circuit



## Index

[ A ]		[ R ]	
AD .....	363	RMAD0 .....	107
ADCON0 .....	362, 365, 368	RMAD1 .....	107
ADCON1 .....	363, 366, 369		
ADCON2 .....	363	[ S ]	
ADIC .....	92	S0RIC .....	92
AIER .....	107	S0TIC .....	92
		S1RIC .....	92
[ C ]		S1TIC .....	92
CM0 .....	66	SAR .....	320
CM1 .....	67	SSCRH .....	285
CSPR .....	115	SSCRL .....	286
		SSER .....	288
[ F ]		SSMR .....	287
FMR0 .....	386	SSMR2 .....	290
FMR1 .....	387	SSRDR .....	291
FMR4 .....	388	SSSR .....	289
FRA0 .....	69	SSTDR .....	291
FRA1 .....	69	SSUIC .....	93
FRA2 .....	70		
		[ T ]	
[ I ]		TRA .....	123
ICCR1 .....	315	TRACR .....	121
ICCR2 .....	316	TRAIC .....	92
ICDRR .....	321	TRAI0C .....	121, 124, 127, 129, 131, 134
ICDRS .....	321	TRAMR .....	122
ICDRT .....	320	TRAPRE .....	122
ICIER .....	318	TRBCR .....	138
ICMR .....	317	TRBIC .....	92
ICSR .....	319	TRBIOC .....	139, 141, 145, 148, 153
IICIC .....	93	TRBMR .....	139
INT0IC .....	94	TRBOCR .....	138
INT1IC .....	94	TRBPR .....	140
INT2IC .....	94	TRBPRE .....	140
INT3IC .....	94	TRBSC .....	140
INTEN .....	101	TRD0 .....	180, 195, 209, 220, 232, 245
INTF .....	102	TRD0IC .....	93
		TRD1 .....	180, 195, 209, 232
[ K ]		TRD1IC .....	93
KIEN .....	105	TRDCR0 .....	176, 191, 206, 218, 229, 243
KUPIC .....	92	TRDCR1 .....	176, 191, 206, 229
		TRDDF0 .....	175
[ L ]		TRDDF1 .....	175
LINCR .....	347	TRDFCR .....	174, 188, 204, 216, 227, 240
LINST .....	348	TRDGRAi (i = 0 or 1) .....	181, 196, 210, 221, 232, 246
		TRDGRBi (i = 0 or 1) .....	181, 196, 210, 221, 232, 246
[ O ]		TRDGRCi (i = 0 or 1) .....	181, 196, 210, 221, 232, 246
OCD .....	68	TRDGRDi (i = 0 or 1) .....	181, 196, 210, 221, 232, 246
OFS .....	23, 114, 381	TRDIER0 .....	180, 195, 208, 220, 231, 245
		TRDIER1 .....	180, 195, 208, 220, 231, 245
[ P ]		TRDIORA0 .....	177, 192
PDi (i = 0 to 4, 6) .....	48	TRDIORA1 .....	177, 192
Pi (i = 0 to 4, 6) .....	48	TRDIORC0 .....	178, 193
PM0 .....	62	TRDIORC1 .....	178, 193
PM1 .....	62	TRDMR .....	172, 186, 203, 216, 226, 240
PMR .....	49, 270, 291, 321	TRDOCR .....	190, 206, 242
PRCR .....	86	TRDOER1 .....	189, 205, 217, 228, 241
PUR0 .....	49	TRDOER2 .....	189, 205, 217, 228, 241
PUR1 .....	49	TRDPMR .....	173, 187, 203
		TRDPOCR0 .....	209
		TRDPOCR1 .....	209
		TRDSR0 .....	179, 194, 207, 219, 230, 244
		TRDSR1 .....	179, 194, 207, 219, 230, 244
		TRDSTR .....	172, 186, 202, 215, 225, 239
		TRECR1 .....	261

TRECR2 .....	261
TRECSR .....	262
TREIC .....	92
TREMIN .....	260
TRESEC .....	260

## [ U ]

U1SR .....	270
UiBRG (i = 0 or 1) .....	267
UiC0 (i = 0 or 1) .....	269
UiC1 (i = 0 or 1) .....	269
UiMR (i = 0 or 1) .....	268
UiRB (i = 0 or 1) .....	267
UiTB (i = 0 or 1) .....	267

## [ V ]

VCA1 .....	31
VCA2 .....	31, 70
VW1C .....	32
VW2C .....	33

## [ W ]

WDC .....	114
WDTR .....	115
WDTS .....	115

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.10	Sep 29, 2005	–	First Edition issued
0.20	Jun 28, 2006	All pages	<ul style="list-style-type: none"> <li>• Symbol name revised. “SSUAIC” → “SSUIC”, “IIC2AIC” → “IICIC”, “TRDMDR” → “TRDMR”</li> <li>• Bit Symbol name revised. “TSTOP0” → “CSEL0” in the TRDSTR register “TSTOP1” → “CSEL1” in the TRDSTR register “TPSC0” → “TCK0” in the TRDCRi register (i=0 or 1) “TPSC1” → “TCK1” in the TRDCRi register (i=0 or 1) “TPSC2” → “TCK2” in the TRDCRi register (i=0 or 1)</li> <li>• Pin name revised. “TCLK” → “TRDCLK”</li> <li>• Register name revised. “Timer RE Comparison Reigster” → “Timer RE Compare Data Register”</li> </ul> <p>1 1. Overview, on the 5th and 6th lines; “data flash” added.</p> <p>2 Table 1.1 Functions and Specifications for R8C/20 Group revised</p> <p>3 Table 1.2 Functions and Specifications for R8C/21 Group revised</p> <p>4 Figure 1.1 Block Diagram; “System Clock Generation” → “System clock generation circuit” revised</p> <p>5 Table 1.3 Product Information of R8C/20 Group revised. Figure 1.2 Type Number, Memory Size, and Package of R8C/20 Group revised.</p> <p>6 Table 1.4 Product Information of R8C/21 Group revised Figure 1.3 Type Number, Memory Size, and Package of R8C/21 Group revised.</p> <p>7 Figure 1.4 Pin Assignments (Top View); “TCLK” → “TRDCLK” revised</p> <p>8 Table 1.5 Pin Functions; “Analog Power Supply Input” revised</p> <p>9 Table 1.6 Pin Name Information by Pin Number revised. NOTE added.</p> <p>11 2.8.1 Carry Flag (C) “2.8.1 Carry Flag (C Flag)” → “2.8.1 Carry Flag (C)” revised. 2.8.2 Debug Flag (D) “2.8.2 Debug Flag (D Flag)” → “2.8.2 Debug Flag (D)” revised. 2.8.3 Zero Flag (Z) “2.8.3 Zero Flag (Z Flag)” → “2.8.3 Zero Flag (Z)” revised. 2.8.4 Sign Flag (S) “2.8.4 Sign Flag (S Flag)” → “2.8.4 Sign Flag (S)” revised. 2.8.5 Register Bank Select Flag (B) “2.8.5 Rgister Bank Select Flag (B Flag)” → “2.8.5 Register Bank Select Flag (B)” revised. 2.8.6 Overflow Flag (O) “2.8.6 Overflow Flag (O Flag)” → “2.8.6 Overflow Flag (O)” revised.</p>

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.20	Jun 28, 2006	12	2.8.7 Interrupt Enable Flag (I) "2.8.7 Interrupt Enable Flag (I Flag)" → "2.8.7 Interrupt Enable Flag (I)" revised. 2.8.8 Stack Pointer Select Flag (U) "2.8.8 Stack Pointer Select Flag (U Flag)" → "2.8.8 Stack Pointer Select Flag (U)" revised. 2.8.10 Reserved Bit "2.8.10 Reserved Area" → "2.8.10 Reserved Bit" revised.
		13	Figure 3.1 Memory Map of R8C/20 Group; "Internal ROM" → "Internal ROM (program ROM)" revised Address "1ZZZZh" added. NOTE revised. Part Number revised.
		14	Figure 3.2 Memory Map of R8C/21 Group; "Internal ROM" → "Internal ROM (program ROM)" revised. "Data area" → "Data flash" "program area" → "program ROM" revised Address "1ZZZZh" added. NOTE2 added. Part Number revised.
		15	Table 4.1 SFR Information (1); 001Ch: 00h → 00h, 1000000b 0024h: TBD → Value when shipping NOTES revised.
		24	Figure 5.4 OFS Register, Function of the LVD1ON bit; "~ after Hardware reset" → "~ after reset" revised. NOTES revised.
		25	5.1.1 When Power Supply is Stable (2) revised. 5.1.2 Power On (4) revised.
		26	Figure 5.5 Example of Hardware Reset Circuit and Operation and Figure 5.6 Example of Hardware Reset Circuit (Usage Example of External Supply Voltage Detection Circuit) and Operation revised.
		27	5.2 Power-On Reset Function, on the 2nd line; "When a capacitor is ~ or more." added. Figure 5.7 Example of Power-On Reset Circuit and Operation revised. NOTES revised.
		28	5.3 Voltage Monitor 1 Reset(1); on the 8th line; The LVD1ON bit in the OFS register can select to~ "after a reset" added.
		29 to 62	"6. Programmable I/O Ports" → "6. Voltage Detection Circuit" and "7. Voltage Detection Circuit" → "7. Programmable I/O Ports" revised.
		32	Figure 6.4 Registers VCA1 and VCA2; VCA2 register revised.
33	Figure 6.5 VW1C Register revised.		

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.20	Jun 28, 2006	35	6.1 VCC Input Voltage; "6.1 Monitoring VCC Input Voltage" → "6.1 VCC Input Voltage" revised.
		42	Figure 7.2 Configuration of Programmable I/O Ports (2) revised.
		43	Figure 7.3 Configuration of Programmable I/O Ports (3) revised.
		45	Figure 7.5 Configuration of Programmable I/O Ports (5) revised.
		47	Figure 7.7 Configuration of Programmable I/O Ports (7) revised.
		49	Figure 7.9 PDi (i = 0 to 4 and 6) Registers; NOTE3 added.
			Figure 7.10 Pi (i = 0 to 4 and 6) Registers; P6 Address "00EEh" → "00ECh" corrected.
		51 to 61	7.4 Port Settings added.
		63	8.1 Processor Modes "8.1 Type of Processor Mode" → "8.1 Processor Modes" revised.
		64	9. Bus revised; Table 9.2 Bus Cycles by Access Space of the R8C/21 Group added. Table 9.3 Access Unit and Bus Operations; "SFR" → "SFR, data flash" "ROM/RAM" → "ROM (program ROM), RAM" below the Table.9.3 "However, only following ~ at a time." added.
		67	Figure 10.2 CM0 Register; NOTE6 deleted.
		69	Figure 10.4 OCD Register; "System clock select bet(3)" → "System clock select bet(4)" "1:Selects on-chip oscillator clock(4)" → "1:Selects on-chip oscillator clock(3)" corrected.
		70	Figure 10.5 Registers FRA0 and FRA1; NOTE2 in the FRA0 register revised.
		71	Figure 10.7 VCA2 Register added.
		72	Figure 10.8 Examples of XIN Clock Connection Circuit; NOTE revised.
73	10.2.2 High-Speed On-Chip Oscillator Clock, on the 3rd and 8th lines; "To use the high-speed ~ (divide-by-4 mode or more)." added. "Since the difference ~ each bit" → "Since there are ~ individual bits." revised.		
74	10.3.5 fOCO40M; "fOCO40M can be used with supply voltage VCC = 3.0 to 5.5V." added.		
75	Table 10.2 Settings and Modes of Clock Associated Bits; "-: can be 0 or 1, no change in outcome." added.		

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.20	Jun 28, 2006	76	10.4.1.3 Low-Speed On-Chip Oscillator Mode, on the 8th line; "In this mode, ~ consumption operation." added.
			10.4.2.2 Entering Wait Mode revised.
			10.4.2.3 Pin Status in Wait Mode revised.
		77	10.4.2.4 Exiting Wait Mode revised.
			Table 10.3 Interrupts to Exit Wait Mode and Usage Conditions; CM02 = 1 of Timer RA Interrupt revised. CM02 = 1 of Timer RD Interrupt revised.
		78	Figure 10.9 Time from Wait Mode to Interrupt Routine Execution added.
		82	Figure 10.11 Procedure for Switching Clock Source from Low-Speed On-Chip Oscillator to XIN Clock revised.
		83	Figure 10.12 Example of Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, Voltage Monitor 1, or Voltage Monitor 2 Interrupt revised.
		84	10.6 Notes on Clock Generation Circuit revised.
		86	Figure 12.1 Interrupts; Address break (2) added.
		90	Table 12.2 Relocatable Vector Tables; "AORIC" → "S0RIC" corrected.
		96	Table 12.5 IPL Value When Software or Special Interrupt Is Acknowledged; "Address break" added.
		98	Figure 12.10 Priority Levels of Hardware Interrupts; "Address break" added.
		109	12.6 Notes on Interrupts; "12.6 Precautions on Interrupts" → "12.6 Notes on Interrupts" revised.
		112	Figure 13.1 Block Diagram of Watchdog Timer; "("L") active", "PM12: Bit in PM1 register" added.
		113	Figure 13.2 Registers OFS and WDC revised.
		117	14. Timers; "The count source for ~ counting and reloading" deleted.
118	Table 14.1 Functional Comparison of Timers; Count source of Timer RD, "TRCIOA0" → "TRDIOA0" corrected.		
119	14.1 Timer RA, the 5th line; "The count source for ~ counting and reloading" added. Figure 14.1 Block Diagram of Timer RA revised.		
120	Figure 14.2 Registers TRACR and TRAI0C; The TRAI0C register revised.		
139	14.1.6 Notes on Timer RA; "14.1.6 Precautions on Timer RA" → "14.1.6 Notes on Timer RA" revised.		
140	14.2 Timer RB, on the 5th line; "The count source for timer RB ~ counting and reloading" added. Figure 14.17 Block Diagram of Timer RB revised.		

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.20	Jun 28, 2006	146	Table 14.8 Programmable Waveform Generation Mode Specifications, "Write to Timer" in the item; "TRAPRE" → "TRBPRES" corrected.
		149	Table 14.9 Programmable One-Shot Generation Mode Specifications, "Write to Timer" in the item; "TRAPRE" → "TRBPRES" corrected.
		153	Table 14.10 Programmable Wait One-Shot Generation Mode Specifications, "Write to Timer" in the item; "TRAPRE" → "TRBPRES" corrected.
		157	Table 14.11 Timer RD Operation Clocks; "TPSC2" → "TCK2" and "TRSC0" → "TCK0" revised. On the 5th line below the Table 14.11; "(Pin output ~ detection)" added.
		158 to 160	Table 14.12 Pin Functions TRDIOA0/TRDCLK(P2_0) Table 14.13 Pin Functions TRDIOB0(P2_1) Table 14.14 Pin Functions TRDIOC0(P2_2) Table 14.15 Pin Functions TRDIOD0(P2_3) Table 14.16 Pin Functions TRDIOA1(P2_4) Table 14.17 Pin Functions TRDIOB1(P2_5) Table 14.18 Pin Functions TRDIOC1(P2_6) Table 14.19 Pin Functions TRDIOD1(P2_7) Table 14.20 Pin Functions INT0(P4_5) added.
		162	14.3.1 Mode Selection deleted Table 14.21 Count Source Selection; Selection of f1, f2, f4, f8, f32, and fOCO40M revised. Figure 14.29 Block Diagram of Count Source; "TPSC2 to TPSC0" → "TCK2 to TCK0" revised.
		163	Figure 14.30 Buffer Operation in Input Capture Function revised.
		164	Figure 14.31 Buffer Operation in Output Compare Function revised. On the 4th line below the Figure 14.31; "IOC2 to IOC0 bits" → "IOC2 bit" and "IOA2 to IOA0 bits" → "IOA2 bit" revised. On the 7th line below the Figure 14.31; "IOD2 to IOD0 bits" → "IOD2 bit" and "IOB2 to IOB0 bits" → "IOB2 bit" revised. On the 8th line below the Figure 14.31; "Bits IMFC ~ capture function" added.
		165	Below the Figure 14.32 Synchronous Operation; "For the synchronous ~ register=110b)" deleted.
		166	14.3.4 Pulse Output Forced Cutoff, on the 13th line; "P2D" → "PD2" corrected. On the 15th line; "P4_5 bit in the P4D register" → "PD4_5 bit in the PD4 register" corrected. On the 2nd line from the bottom; "According to ~ of interrupts" added.

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.20	Jun 28, 2006	168	14.3.5 Input Capture Function, on the 5th line; “The TRDGRA0 register ~ trigger input.” added. Figure 14.34 Block Diagram of Input Capture Function; NOTES revised.
		169	Table 14.23 Input Capture Function Specifications revised.
		170	Figure 14.35 Registers TRDSTR and TRDMR in Input Capture Function; The TRDSTR register revised.
		171	Figure 14.36 TRDPMR Register in Input Capture Function revised.
		172	Figure 14.37 TRDFCR Register in Input Capture Function revised.
		174	Figure 14.39 Registers TRDCR0 to TRDCR1 in Input Capture Function revised.
		175	Figure 14.40 Registers TRDIORA0 to TRDIORA1 in Input Capture Function revised.
		176	Figure 14.41 Registers TRDIORC0 to TRDIORC1 in Input Capture Function revised.
		177	Figure 14.42 Registers TRDSR0 to TRDSR1 in Input Capture Function revised.
		183	Table 14.25 Output Compare Function Specifications, on the 5 to 6th lines from the bottom; “TRCIOAi” → “TRDIOAi” and “TRCIOBi” → “TRDIOBi” corrected.
		184	Figure 14.49 Registers TRDSTR and TRDMR in Output Compare Function revised.
		185	Figure 14.50 TRDPMR Register in Output Compare Function revised.
		186	Figure 14.51 TRDFCR Register in Output Compare Function revised.
		187	Figure 14.52 Registers TRDOER1 to TRDOER2 in Output Compare Function; NOTE in the TRDOER2 register added.
		189	Figure 14.54 Registers TRDCR0 to TRDCR1 in Output Compare Function revised.
		190	Figure 14.55 Registers TRDIORA0 to TRDIORA1 in Output Compare Function revised.
		191	Figure 14.56 Registers TRDIORC0 to TRDIORC1 in Output Compare Function revised.
		192	Figure 14.57 Registers TRDSR0 to TRDSR1 in Output Compare Function revised.
		200	Figure 14.65 TRDSTR Register in PWM Mode revised
		201	Figure 14.66 Registers TRDMR and TRDPMR in PWM Mode revised
202	Figure 14.67 TRDFCR Register in PWM Mode revised		
203	Figure 14.68 Registers TRDOER1 to TRDOER2 in PWM Mode; NOTE in the TRDOER2 register added.		
204	Figure 14.69 Registers TRDOCR and TRDCR0 to TRDCR1 in PWM Mode revised.		



REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.20	Jun 28, 2006	205	Figure 14.70 Registers TRDSR0 to TRDSR1 in PWM Mode revised.
		212	Table 14.29 Reset Synchronous PWM Mode Specifications revised.
		213	Figure 14.78 TRDSTR Register in Reset Synchronous PWM Mode revised.
		214	Figure 14.79 Registers TRDMR and TRDFCR in Reset Synchronous PWM Mode revised.
		215	Figure 14.80 Registers TRDOER1 to TRDOER2 in Reset Synchronous PWM Mode; NOTE in the TRDOER2 register added.
		216	Figure 14.81 TRDCR0 Register in Reset Synchronous PWM Mode revised.
		217	Figure 14.82 Registers TRDSR0 to TRDSR1 in Reset Synchronous PWM Mode revised.
		223	Figure 14.88 TRDSTR Register in Complementary PWM Mode revised.
		224	Figure 14.89 TRDMR Register in Complementary PWM Mode revised.
		225	Figure 14.90 TRDFCR Register in Complementary PWM Mode revised.
		226	Figure 14.91 Registers TRDOER1 to TRDOER2 in Complementary PWM Mode; NOTE in the TRDOER2 register added.
		227	Figure 14.92 Registers TRDCR0 to TRDCR1 in Complementary PWM Mode revised.
		228	Figure 14.93 Registers TRDSR0 to TRDSR1 in Complementary PWM Mode revised.
		231	Below the Table 14.32; "Since values ~ (buffer register)." added.
		235	Figure 14.99 Block Diagram of PWM3 Mode; "Buffer" added.
		236	Table 14.33 PWM3 Mode Specifications revised.
		237	Figure 14.100 TRDSTR Register in PWM3 Mode revised.
		238	Figure 14.101 Registers TRDMR and TRDFCR in PWM3 Mode revised.
		239	Figure 14.102 Registers TRDOER1 to TRDOER2 in PWM3 Mode; NOTE in the TRDOER2 register added.
		241	Figure 14.104 TRDCR0 Register in PWM3 Mode revised.
242	Figure 14.105 Registers TRDSR0 and TRDSR1 in PWM3 Mode added.		
243	Figure 14.106 Registers TRDIER0 and TRDIER1 in PWM3 Mode revised.		
245	Table 14.34 TRDGRji Register Functions in PWM3 Mode revised. On the 4th line from the bottom; "Registers TRDGRC0, ~ (buffer register)." added.		
246	Figure 14.109 Operating Example of PWM3 Mode revised.		

## REVISION HISTORY

## R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.20	Jun 28, 2006	249	14.3.12 Notes on Timer RD; "14.3.13 Precautions on Timer RD" → "14.3.12 Notes on Timer RD" revised.
			14.3.12.1 TRDSTR Register (i = 0 or 1) added.
		250	14.3.12.6 Reset Synchronous PWM Mode revised. 14.3.12.7 Complementary PWM Mode revised.
		254	14.3.13.7 PWM3 mode deleted. 14.3.12.8 Count Source fOCO40M added.
		255	14.4 Timer RE, on the 3rd line; "The count source ~ timer operations." added.
		262	14.4.2 Notes on Timer RE; "14.4.2 Precautions on Timer RE" → "14.4.2 Notes on Timer RE" revised.
		267	Figure 15.5 Registers UiC0 and UiC1 (i = 0 or 1); The UiC0 register (i=0 or 1) revised.
		275	Table 15.5 Registers Used and Settings for UART Mode revised. Table 15.6 I/O Pin Functions in UART Mode revised.
		276	Figure 15.10 Transmit Timing in UART Mode revised.
		277	Figure 15.11 Receive Timing Example in UART Mode revised.
		279	15.3 Notes on Serial Interface; "15.3 Precautions on Serial Interface" → "15.3 Notes on Serial Interface" revised.
		280	16. Clock Synchronous Serial Interface, on the 3rd line; "(SSU)" added.
		281	16.2 Clock Synchronous Serial I/O with Chip Select (SSU); "(SSU)" added. Table 16.2 Clock Synchronous Serial I/O with Chip Select Specifications; NOTE2 deleted.
		285	Figure 16.4 SSMR Register revised.
		288	Figure 16.7 SSMR2 Register revised.
		289	Figure 16.8 Registers SSTDR and SSRDR; NOTE in the SSTDR register revised.
		290	16.2.1 Transfer Clock; "φ" → "f1" revised.
		296	16.2.5.2 Data Transmission; "16.2.5.2 Data Transmit" → "16.2.5.2 Data Transmission" revised. 16.2.5.2 Data Transmission, on the 4th line from the bottom; "When setting the ~ transmit is enabled." deleted.
		297	Figure 16.14 Sample Flowchart of Data Transmission (Clock Synchronous Communication Mode); NOTE revised.
		298	16.2.5.3 Data Reception; "16.2.5.3 Data Receive" → "16.2.5.3 Data Reception" revised.

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.20	Jun 28, 2006	300	16.2.5.4 Data Transmission/Reception; "16.2.5.4 Data Transmit/Receive" → "16.2.5.4 Data Transmission/Reception" revised. 16.2.5.4 Data Transmission/Reception, on the 5th line from the bottom; "When setting the ~ transmit is enabled." deleted.
		301	Figure 16.17 Sample Flowchart of Data Transmission/Reception (Clock Synchronous Communication Mode) revised.
		304	16.2.6.2 Data Transmission; "16.2.6.2 Data Transmit" → "16.2.6.2 Data Transmission" revised. 16.2.6.2 Data Transmission, on the 9th line from the bottom; "When setting the ~ transmit is enabled." deleted.
		306	16.2.6.3 Data Reception; "16.2.6.3 Data Receive" → "16.2.6.3 Data Reception" revised.
		309	16.2.8 Notes on Clock Synchronous Serial I/O with Chip Select; "16.2.8 Precautions on Clock Synchronous Serial I/O with Chip Select" → "16.2.8 Notes on Clock Synchronous Serial I/O with Chip Select" revised.
		338	Figure 16.46 Example of Register Setting in Master Transmit Mode (I <sup>2</sup> C Bus Interface Mode); "Figure 16.46 Example of Register Setting in Master Transmit Mode (Clock Synchronous Serial)" → "Figure 16.46 Example of Register Setting in Master Transmit Mode (I <sup>2</sup> C Bus Interface Mode)" revised.
		339	Figure 16.47 Example of Register Setting in Master Receive Mode (I <sup>2</sup> C Bus Interface Mode); "Figure 16.47 Example of Register Setting in Master Receive Mode (Clock Synchronous Serial)" → "Figure 16.47 Example of Register Setting in Master Receive Mode (I <sup>2</sup> C Bus Interface Mode)" revised.
		340	Figure 16.48 Example of Register Setting in Slave Transmit Mode (I <sup>2</sup> C Bus Interface Mode); "Figure 16.48 Example of Register Setting in Slave Transmit Mode (Clock Synchronous Serial)" → "Figure 16.48 Example of Register Setting in Slave Transmit Mode (I <sup>2</sup> C Bus Interface Mode)" revised.
		341	Figure 16.49 Example of Register Setting in Slave Receive Mode (I <sup>2</sup> C Bus Interface Mode); "Figure 16.49 Example of Register Setting in Slave Receive Mode (Clock Synchronous Serial)" → "Figure 16.49 Example of Register Setting in Slave Receive Mode (I <sup>2</sup> C Bus Interface Mode)" revised.
		342	16.3.8 Notes on I <sup>2</sup> O Bus Interface; "16.3.8 Precautions on I <sup>2</sup> O Bus Interface" → "16.3.8 Notes on I <sup>2</sup> O Bus Interface" revised.
		343 to 385	17. Hardware LIN; "Sync" → "Synch" revised.
		345	Figure 17.2 LINCR Register revised.
		346	Figure 17.3 LINST Register revised.

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.20	Jun 28, 2006	347	Figure 17.4 Typical Operation when Sending a Header Field; "RAIC" → "TRAIC" corrected.
		348	Figure 17.5 Example of Header Field Transmission Flowchart (1) revised.
		349	Figure 17.6 Example of Header Field Transmission Flowchart (2) revised.
		350	Figure 17.7 Typical Operation when Receiving a Header Field; "RAIC" → "TRAIC" corrected.
		351	Figure 17.8 Example of Header Field Reception Flowchart (1) revised.
		352	Figure 17.9 Example of Header Field Reception Flowchart (2) revised.
		353	Figure 17.10 Example of Header Field Reception Flowchart (3) revised.
		354	Figure 17.11 Typical Operation when a Bus Collision is Detected; "RAIC" → "TRAIC" corrected.
		355	17.5 Interrupt Requests, on the 2nd line; "Synch Break generation competed" added.
		357	Table 18.1 Performance of A/D converter revised.
		358	Figure 18.1 Block Diagram of A/D Converter; "ADGSEL" → "ADGSEL0" corrected.
		361	Table 18.2 One-Shot Mode Specifications, Input pin; "AN8" → "AN0" corrected.
		367	18.3 Sample and Hold, on the 2nd and 5th lines; "to 28 φ AD cycles ~ 10-bit resolution." deleted. "When performing ~ the microcomputer." deleted.
		368	18.4 A/D Conversion Cycles added.
		369	18.5 Internal Equivalent Circuit of Analog Input added.
		370	18.6 Output Impedance of Sensor Under A/D Conversion added.
		371	18.7 Notes on A/D Converter; "18.7 Precautions on A/D Converter" → "18.7 Notes on A/D Converter" revised.
		372	19. Flash Memory; "19. Flash Memory Version" → "19. Flash Memory" revised. Table 19.1 Flash Memory Performance, Program and Erase Endurance; "Program Area" → "Program ROM" "Data Area" → "Data ROM" revised.
		374	19.2 Memory Map, on the 4th and 5th lines; "(program ROM)" and "(data flash)" added. Figure 19.1 Flash Memory Block Diagram for R8C/20 Group revised.
		375	Figure 19.2 Flash Memory Block Diagram for R8C/21 Group revised.
		376	19.3 Functions to Prevent Rewriting of Flash Memory; "19.3 Functions to prevent Flash Memory from Rewriting" → "19.3 Functions to Prevent Rewriting of Flash Memory" revised. 19.3.2 ROM Code Protect Function, on the 5th and 7th lines; "The ROM code ~ flash memory." deleted. "write 0 to the ROMCR bit" → "erase the block including the OFS register" revised.

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.20	Jun 28, 2006	377	Figure 19.4 OFS Register revised.
		379	19.4.2 EW1 Mode, on the 3rd line; “Do not execute software command ~” → “Do not execute command ~” revised.
		380	19.4.2.1 FMR00 Bit, on the 1st line; “(including suspend periods)” added.
		381	19.4.2.16 FMR47 Bit revised.
		382	Figure 19.5 FMR0 Register; NOTE6 added.
		384	Figure 19.7 FMR4 Register; NOTES revised.
		385	Figure 19.8 Timing of Suspend Operation revised.
		389	19.4.3.4 Program Command, on the 5th line; “The FMR00 bit is ~ completes.” → “When suspend function ~ auto-programming completes.” revised.
		390	Figure 19.13 Program Command (When Suspend Function Enabled) added.
		391	19.4.3.5 Block Erase, on the 11th line; “The block erase ~ program suspend.” → “Do not use ~ program-suspend” revised.
		392	Figure 19.15 Block Erase Command (When Erase-Suspend Function Enabled) revised.
		393	Table 19.5 Status Register Bits, Value after Reset of SR7 (D7) “0” → “1” corrected.
		396	19.5 Standard Serial I/O Mode, on the 3rd line; “Standard serial I/O ~ interface” → “There are three ~ serial I/O mode 3.” revised. Table 19.7 Pin Functions (Flash Memory Standard Serial I/O Mode 2) revised.
		398	Figure 19.17 Pin Connections for Standard Serial I/O Mode 3; “Figure 19.17 Pin Connections for Standard Serial I/O Mode” → “Figure 19.17 Pin Connections for Standard Serial I/O Mode 3” revised.
		399	Figure 19.18 Pin Processing in Standard Serial I/O Mode 2 added. Figure 19.19 Pin Processing in Standard Serial I/O Mode 3; “Figure 19.19 Pin Processing in Standard Serial I/O Mode” → “Figure 19.19 Pin Processing in Standard Serial I/O Mode 3” revised.
401	19.7 Notes on Flash Memory; “19.7 Precautions on Flash Memory Version” → “19.7 Notes on Flash Memory” revised.		
404 to 423	20. Electrical Characteristics revised.		

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.20	Jun 28, 2006	424	21. Usage Notes; "21. Precautions" → "21. Usage Notes" revised.
		430	21.1.1 Stop Mode and Wait Mode revised.
		431	21.1.3 Oscillation Circuit Constants revised.
		430	21.3.3.1 TRDSTR Register (i = 0 or 1) added.
		431	21.3.3.6 Reset Synchronous PWM Mode; (2) revised.
		431	21.3.3.7 Complementary PWM Mode; (2) revised.
		431	On the 3rd line from the bottom; "However, to write data ~ to 1 (buffer register)." added.
		435	21.3.3.7 PWM3 Mode deleted.
		435	21.3.3.8 Count Source fOCO40M added.
		445	22. Notes on On-Chip Debugger; "22. Precaution for On-chip Debugger" → "22. Notes on On-Chip Debugger" revised.
		445	(2) and (3) added.
		446	Appendix 1. Package Dimensions; "Diagrams shows ~ website." added.
		447	Appendix Figure 2.1 Connection Example with M16C Flash Starter (M3A-0806); NOTES revised.
		448	Appendix Figure 3.1 Example of Oscillation Evaluation Circuit revised.
1.00	Nov 15, 2006	All pages	"Preliminary" and "Under development" deleted
		2	Table 1.1 Functions and Specifications for R8C/20 Group revised. NOTE1 deleted.
		3	Table 1.2 Functions and Specifications for R8C/21 Group revised. NOTE1 deleted.
		5	Table 1.3 Product Information for R8C/20 Group; "R5F2120AJFP (D)", "R5F2120CJFP (D)", "R5F2120AKFP (D)", "R5F2120CKFP (D)", and NOTE added. Figure 1.2 Type Number, Memory Size, and Package of R8C/20 Group; "A: 96 KB" and "C: 128 KB" added.
		6	Table 1.4 Product Information for R8C/21 Group; "R5F2121AJFP (D)", "R5F2121CJFP (D)", "R5F2121AKFP (D)", "R5F2121CKFP (D)", and NOTE added. Figure 1.3 Type Number, Memory Size, and Package of R8C/21 Group; "A: 96 KB" and "C: 128 KB" added.
		13	Figure 3.1 Memory Map of R8C/20 Group revised.
		14	Figure 3.2 Memory Map of R8C/21 Group revised.
		15	Table 4.1 SFR Information (1) <sup>(1)</sup> ; NOTE8; "The CSPROINI bit in the OFS register is set to 0." → "The CSPROINI bit in the OFS register is 0." revised.
		22	Table 5.2 Title of Table revised

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Nov 15, 2006	23	Figure 5.4 OFS Register; NOTE2; "LVD00N" → "LVD10N" revised.
		26	5.2 Power-On Reset Function <sup>(1)</sup> ; NOTE1 deleted. NOTE2 revised. Figure 5.7 Example of Power-On Reset Circuit and Operation revised.
		27	5.3 Voltage Monitor 1 Reset, on the 9th line; "To use the power-on reset function, enable voltage monitor 1 reset by setting the LVD10N bit in the OFS register to 0, bits VW1C0 and VW1C6 in the VW1C register to 1, the VCA bit in the VCA2 register to 1." added. NOTE1 deleted.
			5.4 Voltage Monitor 2 Reset; NOTE1 deleted.
		28	6. Voltage Detection Circuit; NOTE1 deleted.
		31	Figure 6.4 Registers VCA1 and VCA2; Voltage Detection Register 2 <sup>(1)</sup> revised. NOTE5 added.
		40 to 46	Figure 7.1 to Figure 7.3 Configuration of Programmable I/O Ports; NOTE1 added
		48	Figure 7.9 PDi (i = 0 to 4 and 6) Registers; Bit Names revised. Figure 7.10 Pi (i = 0 to 4 and 6) Registers; Bit Names revised.
		64	Table 10.1 Specifications of Clock Generation Circuit; NOTE3; "10 MHz" → "20 MHz" revised.
		68	Figure 10.4 OCD Register; NOTE7; "Figure 10.12" → "Figure 10.14" corrected.
		69	Figure 10.5 Registers FRA0 and FRA1; High-Speed On-Chip Oscillator Control Register 0 <sup>(1)</sup> ; NOTE2 revised. High-Speed On-Chip Oscillator Control Register 1 <sup>(1)</sup> ; NOTE revised.
		70	Figure 10.6 FRA2 Register; High-Speed On-Chip Oscillator Control Register 2 <sup>(1)</sup> revised. NOTE3 added. Figure 10.7 VCA2 Register; Voltage Detection Register 2 <sup>(1)</sup> revised. NOTE5 added.
		71	Figure 10.8 Examples of XIN Clock Connection Circuit; Ceramic resonator external circuit revised.
		72	10.2.2 High-Speed On-Chip Oscillator Clock; On the 4th line revised. On the 2nd line from the bottom; "Adjust the amount of high-speed on-chip oscillator frequency to 40 MHz and below by setting the FRA1 register." added.

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Nov 15, 2006	75	10.4.1.3 Low-Speed On-Chip Oscillator Mode; On the 2nd line from the bottom; "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." added.
		76	10.4.2.4 Exiting Wait Mode; On the 13th line from the bottom; Figure 10.9 shows the Time from Wait Mode to Interrupt Routine Execution. added.
		77	Figure 10.9 Time from Wait Mode to Interrupt Routine Execution revised.
		78	10.4.2.5 Reducing Internal Power Consumption and Figure 10.10 Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit added
		79	10.4.3.3 Exiting Stop Mode, on the 4th line; "Figure 10.11 shows the Time from Stop Mode to Interrupt Routine Execution." added.
		80	Figure 10.11 Time from Stop Mode to Interrupt Routine Execution added.
		81	"Figure 10.10 State Transitions in Power Control Mode" → "Figure 10.12 State Transitions in Power Control Mode" corrected.
		82	10.5.1 How to Use Oscillation Stop Detection Function, on the 6th line; "Figure 10.12" → "Figure 10.14" corrected. On the 10th line; "Figure 10.11" → "Figure 10.13" corrected.
		83	"Figure 10.11 ~" → "Figure 10.13 ~" corrected.
		84	"Figure 10.12 ~" → "Figure 10.15 ~" corrected.
		85	"10.6 Notes on Clock Generation Circuit" revised.
		94	Figure 12.5 Registers INTOIC to INT3IC; NOTE3; "INTOPL" → "INTIPL" corrected.
		102	Figure 12.13 INTF Register revised
		114	Figure 13.2 Registers OFS and WDC; Option Function Select Register <sup>(1)</sup> ; NOTE2 revised. Watchdog Timer Control Register revised.
		117	Table 13.3 Watchdog Timer Specifications (with Count Source Protection Mode Enabled); NOTE2; "CSPRO" → "CSPROINI" corrected.
		120	Figure 14.1 Block Diagram of Timer RA revised.
		121	Figure 14.2 Registers TRACR and TRAI OC revised.
		122	Figure 14.3 Registers TRAMR and TRAPRE Timer RA Mode Register <sup>(1)</sup> ; NOTE added. Timer RA Prescaler Register; NOTE1 revised.
		123	Figure 14.4 TRA Register; NOTE1 revised.
		124	Table 14.2 Timer Mode Specifications; "Write to Timer" revised. Figure 14.5 TRAI OC Register in Timer Mode; NOTES deleted.



REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Nov 15, 2006	125	Figure 14.6 Registers TRAIOC and TRAMR in Timer Mode deleted. 14.1.1.1 Timer Write Control during Count Operation and Figure 14.6 Operating Example of Timer RA when Counter Value is Rewritten during Count Operation added.
		126	Table 14.3 Pulse Output Mode Specifications revised.
		127	Figure 14.7 Register TRACR and TRAIOC in Pulse Output Mode → Figure 14.7 TRAIOC Register in Pulse Output Mode replaced. Timer RA Control Register deleted.
		128	Figure 14.8 TRAMR Register in Pulse Output Mode deleted. Table 14.4 Event Counter Mode Specifications revised.
		129	Figure 14.9 Registers TRACR and TRAIOC in Event Counter Mode → Figure 14.8 TRAIOC Register in Event Counter Mode replaced. Timer RA Control Register deleted. Figure 14.10 TRAMR Register in Event Counter Mode deleted.
		130	14.1.4 Pulse Width Measurement Mode, on the 3rd line; Table 14.5 Pulse Width Measurement Mode Specifications revised.
		131	Figure 14.11 Registers TRACR and TRAIOC in Pulse Width Measurement Mode → Figure 14.9 TRAIOC Register in Pulse Width Measurement Mode replaced. Timer RA Control Register <sup>(4)</sup> deleted. Figure 14.12 TRAMR Register in Pulse Width Measurement Mode deleted.
		132	Figure 14.10 Operating Example of Pulse Width Measurement Mode revised.
		133	Table 14.6 Pulse Period Measurement Mode Specifications revised.
		134	Figure 14.14 Registers TRACR and TRAIOC in Pulse Period Measurement Mode → Figure 14.11 TRAIOC Register in Pulse Period Measurement Mode replaced. Timer RA Control Register <sup>(4)</sup> deleted. Figure 14.15 TRAMR Register in Pulse Period Measurement Mode deleted.
		135	Figure 14.16 Operating Example of Pulse Period Measurement Mode → Figure 14.12 Operating Example of Pulse Period Measurement Mode replaced. NOTE6 revised.
		138	Figure 14.18 Registers TRBCR and TRBOCR → Figure 14.14 Registers TRBCR and TRBOCR replaced. Timer RB Control Register; NOTES revised. Timer RB One-Shot Control Register; Function and NOTES revised. NOTE deleted.
		139	Figure 14.19 Registers TRBIOC and TRBMR → Figure 14.15 Registers TRBIOC and TRBMR replaced. Timer RB Mode Register; TWRC bit and NOTES revised.
		140	Figure 14.20 Registers TRBPRES, TRBSC, and TRBPR → Figure 14.16 Registers TRBPRES, TRBSC, and TRBPR replaced. NOTES revised.

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Nov 15, 2006	141	Table 14.7 Timer Mode Specifications; "Write to Timer" revised. Figure 14.21 Registers TRBIOC and TRBMR in Timer Mode → Figure 14.17 TRBIOC Register in Timer Mode replaced. Timer RB Mode Register deleted.
		142	14.2.1.1 Timer Write Control during Count Operation added.
		143	Figure 14.18 TRBIOC Register in Timer Mode added.
		144	Table 14.8 Programmable Waveform Generation Mode Specifications revised. NOTE2 and NOTE3 revised.
		145	Figure 14.22 Registers TRBIOC and TRBMR in Programmable Waveform Generation Mode → Figure 14.19 TRBIOC Register in Programmable Waveform Generation Mode replaced. TOCNT bit revised. NOTE deleted. Timer RB Mode Register deleted.
		146	Figure 14.23 Operation Example of Timer RB in Programmable Waveform Generation Mode → Figure 14.20 Operation Example of Timer RB in Programmable Waveform Generation Mode replaced. Figure 14.20 revised.
		147	Table 14.9 Programmable One-Shot Generation Mode Specifications revised. NOTE added.
		148	Figure 14.24 Registers TRBIOC and TRBMR in Programmable One-Shot Generation Mode → Figure 14.21 TRBIOC Register in Programmable One-Shot Generation Mode replaced. NOTE revised. Timer RB Mode Register deleted.
		150	14.2.3.1 One-Shot Trigger Selection added.
		152	Table 14.10 Programmable Wait One-Shot Generation Mode Specifications revised. NOTE1 revised.
		153	Figure 14.26 Registers TRBIOC and TRBMR in Programmable Wait One-Shot Generation Mode → Figure 14.23 TRBIOC Register in Programmable Wait One-Shot Generation Mode replaced. TOPL bit and NOTE revised. Timer RB Mode Register deleted.
		154	Figure 14.27 Operation Example of Programmable Wait One-Shot Generation Mode → Figure 14.24 Operation Example of Programmable Wait One-Shot Generation Mode replaced. Figure 14.24 revised.
		155	14.2.5 Notes on Timer RB revised.
		161	Table 14.21 Count Source Selection; NOTE1 added.

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Nov 15, 2006	169	Figure 14.35 Registers TRDSTR and TRDMR in Input Capture Function → Figure 14.32 Registers TRDSTR and TRDMR in Input Capture Function replaced. Timer RD Start Register <sup>(1)</sup> ; “TRD0 count start bit” → “TRD0 count start flag” revised. “TRD1 count start bit” → “TRD1 count start flag” revised.
		183	Figure 14.49 Registers TRDSTR and TRDMR in Output Compare Function → Figure 14.46 Registers TRDSTR and TRDMR in Output Compare Function replaced. Timer RD Start Register <sup>(1)</sup> ; “TRD0 count start bit <sup>(4)</sup> ” → “TRD0 count start flag <sup>(4)</sup> ” revised. “TRD1 count start bit <sup>(5)</sup> ” → “TRD1 count start flag <sup>(5)</sup> ” revised.
		186	Figure 14.52 Registers TRDOER1 to TRDOER2 in Output Compare Function → Figure 14.49 Registers TRDOER1 to TRDOER2 in Output Compare Function replaced. Timer RD Output Master Enable Register 1 revised.
		187	Figure 14.50 TRDOCR Register in Output Compare Function → Figure 14.50 TRDOCR Register in Output Compare Function replaced. NOTE2 added.
		196	On the first line; “Figure 14.63 ~” → “Figure 14.60 lists ~” corrected.
		199	Figure 14.65 TRDSTR Register in PWM Mode → Figure 14.62 TRDSTR Register in PWM Mode replaced. Timer RD Star Register <sup>(1)</sup> ; “TRD0 count start bit <sup>(4)</sup> ” → “TRD0 count start flag <sup>(4)</sup> ” corrected. “TRD1 count start bit <sup>(5)</sup> ” → “TRD1 count start flag <sup>(5)</sup> ” corrected.
		202	Figure 14.68 Registers TRDOER1 to TRDOER2 in PWM Mode → Figure 14.65 Registers TRDOER1 to TRDOER2 in PWM Mode replaced.
		203	Figure 14.69 Registers TRDOCR and TRDCR0 to TRDCR1 in PWM Mode → Figure 14.66 Registers TRDOCR and TRDCR0 to TRDCR1 in PWM Mode replaced. Timer RD Output Control Register <sup>(1)</sup> ; NOTE2 added.
		208	Figure 14.75 Operating Example of PWM Mode → Figure 14.72 Operating Example of PWM Mode replaced. Figure 14.72 revised.
		212	Figure 14.78 TRDSTR Register in Reset Synchronous PWM Mode → Figure 14.75 TRDSTR Register in Reset Synchronous PWM Mode replaced. Timer RD Start Register <sup>(1)</sup> ; “TRD0 count start bit <sup>(4)</sup> ” → “TRD0 count start flag <sup>(4)</sup> ” corrected. “TRDi count start bit <sup>(5)</sup> ” → “TRD1 count start flag <sup>(5)</sup> ” corrected.
		218	Table 14.30 TRDGRji Register Functions in Reset Synchronous PWM Mode; TRDGRA0 register; “(Output inversed every half period of TRDIOC0 pin)” → “(Output inversed every period of TRDIOC0 and PWM pins)”, TRDGRC0 register; “(Output inversed every half period of TRDIOC0 pin)” → “(Output inversed every period of TRDIOC0 and PWM pins)” revised.

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Nov 15, 2006	221	Table 14.31 Complementary PWM Mode Specifications, on the 3rd line from the bottom; "i = 0 to 2, j = either A, B, C or D" → "i = 0 or 1, j = either A, B, C or D" corrected.
		222	Figure 14.85 TRDSTR Register in Complementary PWM Mode → Figure 14.85 TRDSTR Register in Complementary PWM Mode replaced. Timer RD Start Register <sup>(1)</sup> ; TRD0 count start bit <sup>(4)</sup> → "TRD0 count start flag <sup>(4)</sup> " corrected. "TRD1 count start bit <sup>(5)</sup> " → "TRD1 count start flag <sup>(5)</sup> " corrected.
		232	Figure 14.98 Operating Example of Complementary PWM Mode → Figure 14.95 Operating Example of Complementary PWM Mode replaced.
		235	Table 14.33 PWM3 Mode Specifications, on the bottom line; "j = either A, B, C or D" → "i = 0 or 1, j = either A, B, C or D" corrected.
		236	Figure 14.100 TRDSTR Register in PWM3 Mode → Figure 14.97 TRDSTR Register in PWM3 Mode replaced. Timer RD Start Register <sup>(1)</sup> ; "TRD0 count start bit <sup>(4)</sup> " → "TRD0 count start flag <sup>(4)</sup> " corrected. "TRD1 count start bit <sup>(5)</sup> " → "TRD1 count start flag <sup>(5)</sup> " corrected.
		239	Figure 14.103 TRDOCR Register in PWM3 Mode → Figure 14.100 TRDOCR Register in PWM3 Mode replaced. NOTE2 added.
		249	14.3.12.4 Count Source Switch; "count clock source" → "count source" corrected. 14.3.12.7 Complementary PWM Mode, on the bottom line; "Do not use the TRDGRC0 register in complementary PWM mode." deleted.
		265	Figure 15.4 UiMR Register (i = 0 or 1); "Serial Interface mode select bit <sup>(2,4)</sup> " → "Serial I/O mode select bit <sup>(2,4)</sup> " corrected.
		266	Figure 15.5 Registers UiC0 and UiC1 (i = 0 or 1); UARTi Transmit/Receive Control Register 1 (i = 0 or 1) revised. NOTE2 added.
		279	Table 16.1 Mode Selections revised.
		283	Figure 16.3 SSCRL Register; NOTE2 revised
		311	Figure 16.23 External Circuit Connection Example of Pins SCL and SDA revised.
		338	Figure 16.47 Example of Register Setting in Master Receive Mode (I <sup>2</sup> C Bus Interface Mode); (1) "Set the ~ master receive mode ~" → "Set the ~ master transmit mode ~" corrected.
		347	Figure 17.5 Example of Header Field Transmission Flowchart (1); Hard ware LIN Clear the status flags; "~ in LINST register" → "~ in LINST register: 0" corrected.

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Nov 15, 2006	351	Figure 17.9 Example of Header Field Reception Flowchart (2); “When the SBE bit in the LINCR register is 0(Unmasked after Synch Break is detected), timer RA is usable in timer mode after the SBDCT flag in the LINST register is set to 1.” added.
		352	Figure 17.10 Example of Header Field Reception Flowchart (3); “When the SBE bit in the LINCR register is 1 (Unmasked after Synch Field measurement is completed), timer RA is usable in timer mode after the SFDCT flag in the LINST register is set to 1.” added.
		354	17.4.4 Hardware LIN End Processing and Figure 17.12 Example of Hardware LIN Communication Completion Flowchart added.
		365	Figure 18.6 ADCON0 Register in Repeat Mode, in the Function of Frequency select bit 0; “1: Select fOCO-F” → “Do not set” revised.
		369	Figure 18.10 Internal Equivalent Circuit of Analog Input; “i = 4” → “i = 12” corrected.
		371	18.7 Notes on A/D Converter, on the 5th line from the bottom; “Do not select the fOCO-F for the φAD.” added.
		374	19.2 Memory Map, on the 4th line from the bottom; “When rewriting the block 2 and block 3 in CPU rewrite mode, set the FMR02 bit in the FMR0 register to 1 (rewrite enables).” added.
		375	Figure 19.1 Flash Memory Block Diagram for R8C/20 Group revised.
		376	Figure 19.2 Flash Memory Block Diagram for R8C/21 Group revised.
		378	Figure 19.4 OFS Register; NOTE2; “LVD0ON” → “LVD1ON” and “(voltage monitor 0 reset enabled after reset)” → “voltage monitor 0 reset enabled after reset” corrected.
		379	Table 19.3 Differences between EW0 Mode and EW1 Mode; Modes After Read Status Register added.
		381	19.4.2.3 FMR02 Bit; “The block 1 and block 0 do not ~” → “The block0 block1 block2, and block3 do not ~” corrected.
		389	19.4.3.1 Read Array Command, on the bottom line; “In addition, the MCU enters read array mode after a reset.” added. 19.4.3.2 Read Status Register Command, on the bottom line; “The MCU remains in read status register mode until the next read array command is written.” added.
		390	Figure 19.13 Program Command (When Suspend Function Enabled) revised. NOTE3 added.
		393	Figure 19.15 Block Erase Command (When Erase-Suspend Function Enabled) revised. NOTE3 added.
		396	Figure 19.16 Full Status Check and Handling Procedure for Individual Errors; “FMR07 = 0?” → “FMR07 = 1?” and “FMR06 = 0?” → “FMR06 = 1?” corrected.

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Nov 15, 2006	398	Table 19.8 Pin Functions (Flash Memory Standard Serial I/O Mode 3); P4_2/VREF deleted. P4_3 or P4_5 → P4_2 to P4_5 corrected.
		404	19.7.1.7 Reset Flash Memory deleted. 19.7.1.8 Entering Stop Mode or Wait Mode → 19.7.1.7 Entering Stop Mode or Wait Mode corrected.
		405	Table 20.1 Absolute Maximum Ratings; Power dissipation revised. Table 20.2 Recommended Operating Conditions; System clock revised.
		410	Table 20.8 Voltage Monitor 1 Reset Circuit Electrical Characteristics → Table 20.8 Power-on Reset Circuit, Voltage Monitor 1 Reset Circuit Electrical Characteristics <sup>(1)</sup> replaced. Table 20.8 revised. NOTE3 added. Table 20.9 Power-on Reset Circuit Electrical Characteristics deleted. Figure 20.3 Power-on Reset Circuit Electrical Characteristics revised.
		411	Table 20.10 High-Speed On-Chip Oscillator Circuit Electrical Characteristics → Table 20.9 High-Speed On-Chip Oscillator Circuit Electrical Characteristics revised.
		417	Table 20.15 Electrical Characteristics (1) [VCC = 5 V] → Table 20.14 Electrical Characteristics (1) [VCC = 5 V] revised. RAM Hold Voltage, Min.; "1.8" → "2.0" corrected.
		418	Table 20.16 Electrical Characteristics (2) [Vcc = 5 V] → Table 20.15 Electrical Characteristics (2) [Vcc = 5 V] revised. Wait mode revised.
		421	Table 20.21 Electrical Characteristics (3) [VCC = 3 V] → Table 20.20 Electrical Characteristics (3) [VCC = 3 V] revised. RAM hold voltage, Min.; "1.8" → "2.0" corrected.
		422	Table 20.22 Electrical Characteristics (4) [Vcc = 3 V] → Table 20.21 Electrical Characteristics (4) [Vcc = 3 V] revised. Wait mode revised.
		425	21.1.1 Stop Mode and Wait Mode → 21.1.1 Stop Mode revised. 21.1.2 Wait Mode added.
		430	21.3.2 Notes on Timer RB; "Timer RB starts counting at the first valid edge of the count source after The TCSTF bit is set to 1 (during count)." deleted. On the 8th line from the bottom; "- If the TSTOP bit ~ stops immediately." added. On the 6th line from the bottom; "- If 1 is written to the TOSST or TOSSP bit ~ either 0 or 1." added.
		432	21.3.3.4 Count Source Switch; "count clock source" → "count source." corrected.
		442	21.7 Notes on A/D Converter; On the 6th line from the bottom; "Do not select the fOCO-F for the φAD." added.
		445	21.8.1.7 Reset Flash Memory deleted.

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Nov 15, 2006	447	22. Notes on On-Chip Debugger, (2); "ROM 128 KB Product (R5F2120CJFP, R5F2120CKFP, R5F2121CJFP, R5F2121CKFP) addresses 23800h to 23FFFh" added.
		448	(3); ROM 128 KB Product (R5F2120CJFP, R5F2120CKFP, R5F2121CJFP, R5F2121CKFP) addresses 03B00h to 03BFFh added.
1.10	Oct 31, 2007	5	23. Notes on Emulator Debugger added.
		6	Table 1.3; Development status updated.
		7	Table 1.4; Development status updated.
		15	Figure 1.4; NOTE 3 added.
		23	Table 4.1; • 000Ah: "00XXX000b" → "00h" • 000Fh: "00011111b" → "00X11111b"
		24	Table 4.1; Figure 5.3 and Figure 5.4 NOTE1 revised.
		25	5.1.1 (2) and 5.1.2 (4) revised.
		26	Figure 5.5 and Figure 5.6 revised.
		31	5.2 and Figure 5.7 revised.
		52	Figure 6.4; VCA2 register NOTE5 revised.
		53	Table 7.17 revised.
		56	Table 7.19 revised.
		57	Table 7.29 and Table 7.30 revised.
		66	Table 7.33 revised.
		69	Figure 10.2; NOTE4 revised.
		70	Figure 10.5; FRA0 register NOTE2 revised and FRA1 register NOTE2 added.
		73	Figure 10.7; VCA2 register NOTE5 revised.
		75	10.3.2 revised.
		76	10.4.1.3 revised.
		78	Table 10.3; Watchdog Timer Interrupt deleted.
80	10.4.2.5 and Figure 10.10 revised.		
82	Figure 10.11; Remarks revised.		
85	10.5.1; the second line from the bottom revised.		
86	10.6.1; Program example and 10.6.2 revised.		
89	Figure 11.1; After Reset of PRCR register: "00XXX000b" → "00h"		
101	12.1.3.1 revised.		
106	12.2.1 revised.		
110	Table 12.6 revised and NOTE2 added.		
111	12.6.3 revised and Watchdog Timer Interrupt deleted.		
			Figure 12.20; NOTE2 revised.

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.10	Oct 31, 2007	114	Figure 13.2; • OFS register NOTE1 revised. • After Reset of WDC register: "000xxxxb" → "00X11111b"
		125	Figure 14.6 Comment; "0 (During count)" → "1 (During count)"
		136	14.1.6 revised.
		137	14.2; the second line from the top revised.
		140	Figure 14.16; TRBSC register NOTE3: "TRBPRES" → "TRBSC" TRBPR register NOTE2: "TRBPRES" → "TRBPR"
		143	Figure 14.18 Comment; "0 (During count)" → "1 (During count)"
		147	Table 14.9; NOTE2 added.
		152	Table 14.10; NOTE2 added.
		155	14.2.5 revised and 14.2.5.1 added.
		156	14.2.5.2 added.
		157	14.2.5.3 added.
		158	14.2.5.4 added.
		170	Figure 14.33; "input capture signal" added.
		183	Figure 14.46 revised.
		185	Table 14.25; Count Stop Conditions revised.
		201	Table 14.27; Count Stop Conditions revised.
		214	Table 14.29; Count Stop Conditions revised.
		221	Figure 14.84; NOTE1 revised.
		238	Table 14.33; Count Stop Conditions revised.
		251	14.3.12.1 and Table 14.36 revised.
		267	Figure 15.3; Registers U0BRG and U1BRG: "U0BRG" → "UiBRG"
		271	Table 15.1; NOTE2 revised.
		276	Table 15.4; NOTE1 revised.
		277	Table 15.5; NOTE2 added.
		278	Figure 15.10 revised.
		281	15.3; the fourth line from the top added.
		287	Figure 16.4; NOTE2 deleted.
		288	Figure 16.5; NOTE1 deleted.
		289	Figure 16.6; NOTE2 and NOTE7 revised.
		290	Figure 16.7; NOTE5 revised.
		291	Figure 16.8; SSTDR NOTE1 and SSRDR NOTE2 deleted.
		311	16.2.8.1 deleted.
		315	Figure 16.24; NOTE6 revised.



REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.10	Oct 31, 2007	316	Figure 16.25; NOTE5 deleted.
		317	Figure 16.26; NOTE3 revised and NOTE7 deleted.
		318	Figure 16.27; NOTE3 deleted.
		319	Figure 16.28; NOTE7 revised.
		324	Figure 16.32 revised.
		326	Figure 16.33 and Figure 16.34 revised.
		328	Figure 16.35 revised.
		329	Figure 16.36 revised.
		344	16.3.8.1 replaced and 16.3.8.2 added.
		350	Figure 17.5; Procedure of Hardware LIN Clear the status flags: "LINST register ← 0" → "LINST register ← 1"
		352	Figure 17.7 revised.
		353	Figure 17.8; Bit name in the TRAMR register: "MOD0 to 2 bits" → "Bits TMOD0 to TMOD2"
		354	Figure 17.9; Procedure of Hardware LIN Clear the status flags: "LINST register ← 0" → "LINST register ← 1"
		356	Figure 17.11; Bit name in the LINST register: "SCDCT flag" → "BCDCT flag"
		357	Figure 17.12; Procedure of Hardware LIN Clear the status flags: "LINST register ← 0" → "LINST register ← 1"
		362	Figure 18.2; NOTE4 revised.
		364	Table 18.2; Stop Condition revised.
		365	Figure 18.4; NOTE4 revised.
		368	Figure 18.6; NOTE4 revised.
		372	Figure 18.10 revised. SW5 added.
		373	18.6; the six line from the bottom: "A/D conversion mode with" → "A/D conversion mode without"
		374	18.7 revised.
		376	Table 19.2; Function of CPU Rewrite Mode: "any area other than the flash memory" → "the RAM"
		381	Figure 19.4; NOTE1 revised.
		382	Table 19.3; EW1 Mode: "ROM area" → "ROM or RAM area"
		383	19.4.1 and 19.4.2; "td(SR-ES)" → "td(SR-SUS)"
		384	19.4.2.4; the third line from the top: "in other than the flash memory" → "transferred to the RAM"
		385	19.4.2.15 revised.
		386	Figure 19.5; NOTE3 and NOTE5 revised.
		388	Figure 19.7; NOTE5 revised.

**REVISION HISTORY**
**R8C/20 Group, R8C/21 Group Hardware Manual**

Rev.	Date	Description	
		Page	Summary
1.10	Oct 31, 2007	390	Figure 19.9; “any area other than the flash memory” → “the RAM”
		391	Figure 19.11; • “any area other than the flash memory” → “the RAM” • “15us” → “30us” • NOTE4 deleted.
		393	19.4.3.4 revised.
		394	Figure 19.13 revised and NOTE4 added.
		396	Figure 19.15 revised and NOTE4 added.
		400	Table 19.7; MODE pin revised.
		412	Table 20.6; NOTE4 added.
		414	Table 20.11; NOTE1 revised.
		423	Table 20.19 and Figure 20.11; “(i = 0,2,3)” → “(i = 0 to 3)”
		427	Table 20.25 and Figure 20.15; “(i = 0,2,3)” → “(i = 0 to 3)”
		428	21.1.1; Program example and 21.1.2 revised.
		429	21.2.3 revised and Watchdog Timer Interrupt deleted.
		430	Figure 21.1; NOTE2 revised.
		432	21.3.1 revised.
		433	21.3.2 revised and 21.3.2.1 added.
		434	21.3.2.2 added.
		435	21.3.2.3 added.
		436	21.3.2.4 added.
		437	21.3.3.1 and Table 21.1 revised.
		444	21.4; the fourth line from the top added.
445	21.5.2.1 replaced and 21.5.2.2 added.		
447	21.7 revised.		
452	22 revised.		
455	Appendix Figure 2.2 revised.		
456	Appendix Figure 3.1 NOTE1 revised.		
2.00	Aug 27, 2008	–	“RENESAS TECHNICAL UPDATE” reflected: TN-16C-A172A/E
		5, 6	Table 1.3, Table 1.4 revised Figure 1.2, Figure 1.3; ROM number “XXX” added
		13, 14	Figure 3.1, Figure 3.2; “Expanding area” deleted
		26	Figure 5.7 revised
		120	Figure 14.1 “TSTART” → “TCSTF”, “TCKCUT bit” revised
		137	Figure 14.13 “TCSTF” → “TSTART” revised and added
		147	Table 14.9 “TRBP pin function” → “TRBO pin function”

REVISION HISTORY

R8C/20 Group, R8C/21 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
2.00	Aug 27, 2008	172, 186, 225, 239	Figure 14.34, Figure 14.48, Figure 14.87, Figure 14.99; "0137Dh" → "0137h"
		199	Figure 14.62 revised
		201	Table 14.27 revised
		244	Figure 14.104 "Timer RD Status Register i (i = 0 or 1) <sup>(2)</sup> " NOTE2 deleted
		268	Figure 15.4; NOTE3: "... 1 (internal clock) ..." → "... 0 (internal clock) ..."
		286	Figure 16.3 revised
		302	16.2.5.4 added
		305	Figure 16.18 revised
		351	Figure 17.6 revised
		354	Figure 17.9 revised
		375	Table 19.1; NOTE1 revised
		398	Table 19.6 "FRM0 Register" → "FMR0 Register"
		408	Table 20.2; NOTE2 revised
		410	Table 20.4; NOTE2 and NOTE4 revised
		411	Table 20.5; NOTE2 and NOTE5 revised
		412	Table 20.6; "td(Vdet1-A)" added, NOTE5 added Table 20.7; "td(Vdet2-A)" and NOTE2 revised, NOTE5 added
		413	Table 20.8; "trth" and NOTE2 revised Figure 20.3 revised

---

R8C/20 Group, R8C/21 Group Hardware Manual

Publication Data: Rev.0.10 Sep 29, 2005  
Rev.2.00 Aug 27, 2008

Published by: Sales Strategic Planning Div.  
Renesas Technology Corp.

---

© 2008. Renesas Technology Corp., All rights reserved. Printed in Japan

# R8C/20 Group, R8C/21 Group Hardware Manual



Renesas Technology Corp.  
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan