

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

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

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

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

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

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

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
  - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
  - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

# R8C/2K Group, R8C/2L Group

Hardware Manual

RENESAS MCU

R8C FAMILY / R8C/2x SERIES

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

## 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/2K Group, R8C/2L 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/2K Group, R8C/2L Group Group Datasheet	REJ03B0219
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/2K Group, R8C/2L 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

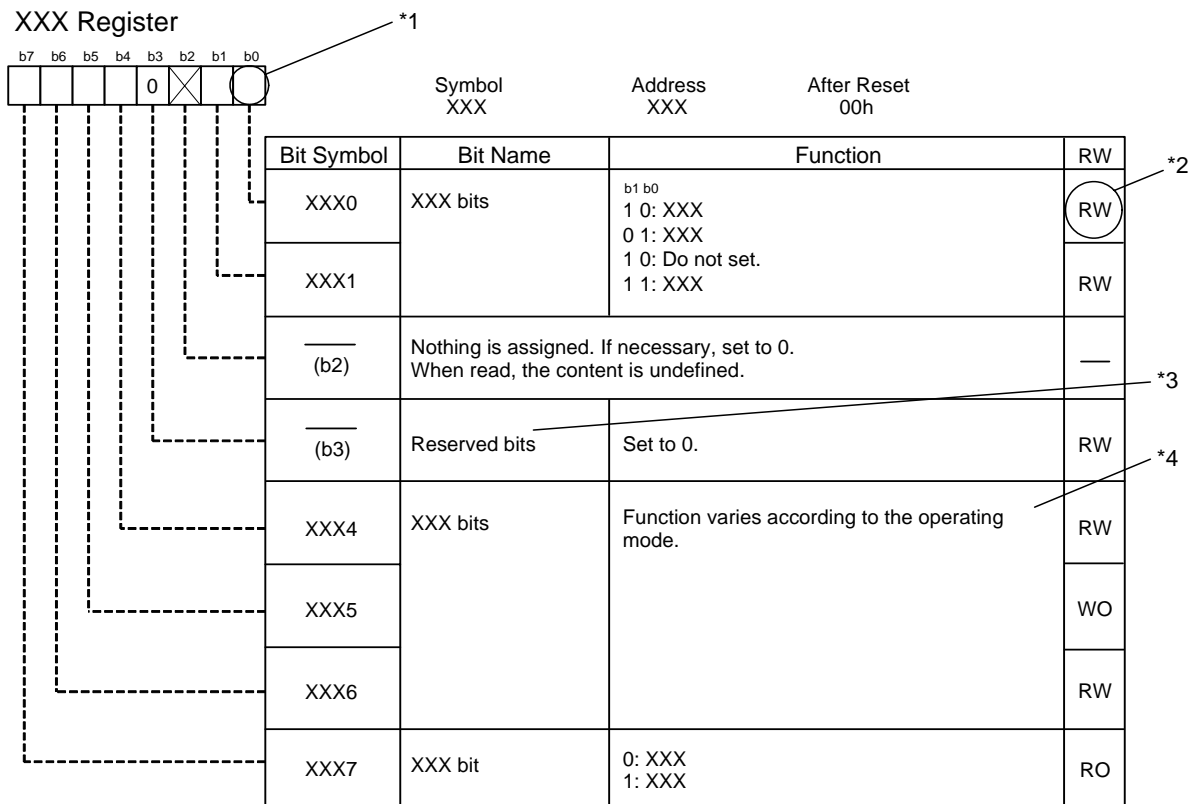
(2) Notation of Numbers

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-Connection
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SFR	Special Function Registers
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

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

# Table of Contents

SFR Page Reference .....	B - 1
1. Overview .....	1
1.1 Features .....	1
1.1.1 Applications .....	1
1.1.2 Specifications .....	2
1.2 Product List .....	6
1.3 Block Diagram .....	8
1.4 Pin Assignment .....	9
1.5 Pin Functions .....	11
2. Central Processing Unit (CPU) .....	12
2.1 Data Registers (R0, R1, R2, and R3) .....	13
2.2 Address Registers (A0 and A1) .....	13
2.3 Frame Base Register (FB) .....	13
2.4 Interrupt Table Register (INTB) .....	13
2.5 Program Counter (PC) .....	13
2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP) .....	13
2.7 Static Base Register (SB) .....	13
2.8 Flag Register (FLG) .....	13
2.8.1 Carry Flag (C) .....	13
2.8.2 Debug Flag (D) .....	13
2.8.3 Zero Flag (Z) .....	13
2.8.4 Sign Flag (S) .....	13
2.8.5 Register Bank Select Flag (B) .....	13
2.8.6 Overflow Flag (O) .....	13
2.8.7 Interrupt Enable Flag (I) .....	14
2.8.8 Stack Pointer Select Flag (U) .....	14
2.8.9 Processor Interrupt Priority Level (IPL) .....	14
2.8.10 Reserved Bit .....	14
3. Memory .....	15
3.1 R8C/2K Group .....	15
3.2 R8C/2L Group .....	16
4. Special Function Registers (SFRs) .....	17
5. Resets .....	24
5.1 Hardware Reset .....	27
5.1.1 When Power Supply is Stable .....	27
5.1.2 Power On .....	27
5.2 Power-On Reset Function .....	29
5.3 Voltage Monitor 0 Reset .....	30
5.4 Voltage Monitor 1 Reset .....	30
5.5 Voltage Monitor 2 Reset .....	30
5.6 Watchdog Timer Reset .....	31
5.7 Software Reset .....	31

6.	Voltage Detection Circuit .....	32
6.1	VCC Input Voltage .....	39
6.1.1	Monitoring Vdet0 .....	39
6.1.2	Monitoring Vdet1 .....	39
6.1.3	Monitoring Vdet2 .....	39
6.2	Voltage Monitor 0 Reset .....	40
6.3	Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset .....	41
6.4	Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset .....	43
7.	Programmable I/O Ports .....	45
7.1	Functions of Programmable I/O Ports .....	45
7.2	Effect on Peripheral Functions .....	46
7.3	Pins Other than Programmable I/O Ports .....	46
7.4	Port settings .....	58
7.5	Unassigned Pin Handling .....	68
8.	Processor Mode .....	69
8.1	Processor Modes .....	69
9.	Bus .....	70
10.	Clock Generation Circuit .....	71
10.1	XIN Clock .....	81
10.2	On-Chip Oscillator Clocks .....	82
10.2.1	Low-Speed On-Chip Oscillator Clock .....	82
10.2.2	High-Speed On-Chip Oscillator Clock .....	82
10.3	CPU Clock and Peripheral Function Clock .....	83
10.3.1	System Clock .....	83
10.3.2	CPU Clock .....	83
10.3.3	Peripheral Function Clock (f1, f2, f4, f8, and f32) .....	83
10.3.4	fOCO .....	83
10.3.5	fOCO40M .....	83
10.3.6	fOCO-F .....	83
10.3.7	fOCO-S .....	83
10.3.8	fOCO128 .....	83
10.4	Power Control .....	84
10.4.1	Standard Operating Mode .....	84
10.4.2	Wait Mode .....	86
10.4.3	Stop Mode .....	89
10.5	Oscillation Stop Detection Function .....	92
10.5.1	How to Use Oscillation Stop Detection Function .....	92
10.6	Notes on Clock Generation Circuit .....	95
10.6.1	Stop Mode .....	95
10.6.2	Wait Mode .....	95
10.6.3	Oscillation Stop Detection Function .....	95
10.6.4	Oscillation Circuit Constants .....	95

11.	Protection .....	96
12.	Interrupts .....	97
12.1	Interrupt Overview .....	97
12.1.1	Types of Interrupts .....	97
12.1.2	Software Interrupts .....	98
12.1.3	Special Interrupts .....	99
12.1.4	Peripheral Function Interrupt .....	99
12.1.5	Interrupts and Interrupt Vectors .....	100
12.1.6	Interrupt Control .....	102
12.2	$\overline{\text{INT}}$ Interrupt .....	111
12.2.1	$\overline{\text{INT}}_i$ Interrupt (i = 0, 1, 3) .....	111
12.2.2	$\overline{\text{INT}}_i$ Input Filter (i = 0, 1, 3) .....	113
12.3	Key Input Interrupt .....	114
12.4	Address Match Interrupt .....	116
12.5	Timer RC Interrupt, Timer RD Interrupt (Interrupts with Multiple Interrupt Request Sources) .....	118
12.6	Notes on Interrupts .....	120
12.6.1	Reading Address 00000h .....	120
12.6.2	SP Setting .....	120
12.6.3	External Interrupt and Key Input Interrupt .....	120
12.6.4	Changing Interrupt Sources .....	121
12.6.5	Changing Interrupt Control Register Contents .....	122
13.	ID Code Areas .....	123
13.1	Overview .....	123
13.2	Functions .....	123
13.3	Notes on ID Code Areas .....	124
13.3.1	Setting Example of ID Code Areas .....	124
14.	Option Function Select Area .....	125
14.1	Overview .....	125
14.2	OFS Register .....	126
14.3	Notes on Option Function Select Area .....	127
14.3.1	Setting Example of Option Function Select Area .....	127
15.	Watchdog Timer .....	128
15.1	Count Source Protection Mode Disabled .....	132
15.2	Count Source Protection Mode Enabled .....	133
16.	Timers .....	134
16.1	Timer RA .....	136
16.1.1	Timer Mode .....	139
16.1.2	Pulse Output Mode .....	141
16.1.3	Event Counter Mode .....	143
16.1.4	Pulse Width Measurement Mode .....	145
16.1.5	Pulse Period Measurement Mode .....	148
16.1.6	Notes on Timer RA .....	151
16.2	Timer RB .....	152
16.2.1	Timer Mode .....	157

16.2.2	Programmable Waveform Generation Mode .....	160
16.2.3	Programmable One-shot Generation Mode .....	163
16.2.4	Programmable Wait One-Shot Generation Mode .....	167
16.2.5	Notes on Timer RB .....	170
16.3	Timer RC .....	174
16.3.1	Overview .....	174
16.3.2	Registers Associated with Timer RC .....	176
16.3.3	Common Items for Multiple Modes .....	186
16.3.4	Timer Mode (Input Capture Function) .....	192
16.3.5	Timer Mode (Output Compare Function) .....	197
16.3.6	PWM Mode .....	203
16.3.7	PWM2 Mode .....	208
16.3.8	Timer RC Interrupt .....	214
16.3.9	Notes on Timer RC .....	215
16.4	Timer RD .....	216
16.4.1	Count Sources .....	221
16.4.2	Buffer Operation .....	222
16.4.3	Synchronous Operation .....	224
16.4.4	Pulse Output Forced Cutoff .....	225
16.4.5	Input Capture Function .....	227
16.4.6	Output Compare Function .....	241
16.4.7	PWM Mode .....	258
16.4.8	Reset Synchronous PWM Mode .....	271
16.4.9	Complementary PWM Mode .....	281
16.4.10	PWM3 Mode .....	295
16.4.11	Timer RD Interrupt .....	308
16.4.12	Notes on Timer RD .....	310
17.	Serial Interface .....	316
17.1	Clock Synchronous Serial I/O Mode .....	322
17.1.1	Polarity Select Function .....	325
17.1.2	LSB First/MSB First Select Function .....	325
17.1.3	Continuous Receive Mode .....	326
17.2	Clock Asynchronous Serial I/O (UART) Mode .....	327
17.2.1	Bit Rate .....	331
17.3	Notes on Serial Interface .....	332
18.	Hardware LIN .....	333
18.1	Features .....	333
18.2	Input/Output Pins .....	334
18.3	Register Configuration .....	335
18.4	Functional Description .....	337
18.4.1	Master Mode .....	337
18.4.2	Slave Mode .....	340
18.4.3	Bus Collision Detection Function .....	344
18.4.4	Hardware LIN End Processing .....	345
18.5	Interrupt Requests .....	346
18.6	Notes on Hardware LIN .....	347

19.	A/D Converter .....	348
19.1	One-Shot Mode .....	352
19.2	Repeat Mode .....	355
19.3	Sample and Hold .....	358
19.4	A/D Conversion Cycles .....	358
19.5	Internal Equivalent Circuit of Analog Input .....	359
19.6	Output Impedance of Sensor under A/D Conversion .....	360
19.7	Notes on A/D Converter .....	361
20.	Flash Memory .....	362
20.1	Overview .....	362
20.2	Memory Map .....	363
20.3	Functions to Prevent Rewriting of Flash Memory .....	364
20.3.1	ID Code Check Function .....	364
20.3.2	ROM Code Protect Function .....	365
20.4	CPU Rewrite Mode .....	366
20.4.1	Register Description .....	367
20.4.2	Status Check Procedure .....	373
20.4.3	EW0 Mode .....	374
20.4.4	EW1 Mode .....	384
20.5	Standard Serial I/O Mode .....	392
20.5.1	ID Code Check Function .....	392
20.6	Parallel I/O Mode .....	395
20.6.1	ROM Code Protect Function .....	395
20.7	Notes on Flash Memory .....	396
20.7.1	CPU Rewrite Mode .....	396
21.	Reducing Power Consumption .....	398
21.1	Overview .....	398
21.2	Key Points and Processing Methods for Reducing Power Consumption .....	398
21.2.1	Voltage Detection Circuit .....	398
21.2.2	Ports .....	398
21.2.3	Clocks .....	398
21.2.4	Wait Mode, Stop Mode .....	398
21.2.5	Stopping Peripheral Function Clocks .....	398
21.2.6	Timers .....	398
21.2.7	A/D Converter .....	398
21.2.8	Reducing Internal Power Consumption .....	399
21.2.9	Stopping Flash Memory .....	400
21.2.10	Low-Current-Consumption Read Mode .....	401
22.	Electrical Characteristics .....	402
23.	Usage Notes .....	423
23.1	Notes on Clock Generation Circuit .....	423
23.1.1	Stop Mode .....	423
23.1.2	Wait Mode .....	423
23.1.3	Oscillation Stop Detection Function .....	423
23.1.4	Oscillation Circuit Constants .....	423

23.2	Notes on Interrupts .....	424
23.2.1	Reading Address 00000h .....	424
23.2.2	SP Setting .....	424
23.2.3	External Interrupt and Key Input Interrupt .....	424
23.2.4	Changing Interrupt Sources .....	425
23.2.5	Changing Interrupt Control Register Contents .....	426
23.3	Notes on Timers .....	427
23.3.1	Notes on Timer RA .....	427
23.3.2	Notes on Timer RB .....	428
23.3.3	Notes on Timer RC .....	432
23.3.4	Notes on Timer RD .....	433
23.4	Notes on Serial Interface .....	439
23.5	Notes on Hardware LIN .....	440
23.6	Notes on A/D Converter .....	441
23.7	Notes on Flash Memory .....	442
23.7.1	CPU Rewrite Mode .....	442
23.8	Notes on Noise .....	444
23.8.1	Inserting a Bypass Capacitor between VCC and VSS Pins as a Countermeasure against Noise and Latch-up .....	444
23.8.2	Countermeasures against Noise Error of Port Control Registers .....	444
24.	Notes for On-Chip Debugger .....	445
	Appendix 1. Package Dimensions .....	446
	Appendix 2. Connection Examples between Serial Writer and On-Chip Debugging Emulator .....	447
	Appendix 3. Example of Oscillation Evaluation Circuit .....	448
	Index .....	449

# SFR Page Reference

Address	Register	Symbol	Page
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	69
0005h	Processor Mode Register 1	PM1	69
0006h	System Clock Control Register 0	CM0	74
0007h	System Clock Control Register 1	CM1	75
0008h			
0009h			
000Ah	Protect Register	PRCR	96
000Bh			
000Ch	Oscillation Stop Detection Register	OCD	76
000Dh	Watchdog Timer Reset Register	WDTR	130
000Eh	Watchdog Timer Start Register	WDTS	130
000Fh	Watchdog Timer Control Register	WDC	130
0010h	Address Match Interrupt Register 0	RMAD0	117
0011h			
0012h			
0013h	Address Match Interrupt Enable Register	AIER	117
0014h	Address Match Interrupt Register 1	RMAD1	117
0015h			
0016h			
0017h			
0018h			
0019h			
001Ah			
001Bh			
001Ch	Count Source Protection Mode Register	CSPR	131
001Dh			
001Eh			
001Fh			
0020h			
0021h			
0022h			
0023h	High-Speed On-Chip Oscillator Control Register 0	FRA0	77
0024h	High-Speed On-Chip Oscillator Control Register 1	FRA1	77
0025h	High-Speed On-Chip Oscillator Control Register 2	FRA2	78
0026h			
0027h			
0028h			
0029h			
002Ah			
002Bh	High-Speed On-Chip Oscillator Control Register 6	FRA6	78
002Ch	High-Speed On-Chip Oscillator Control Register 7	FRA7	78
002Dh			
002Eh			
002Fh			
0030h			
0031h	Voltage Detection Register 1	VCA1	35
0032h	Voltage Detection Register 2	VCA2	35, 79
0033h			
0034h			
0035h			
0036h	Voltage Monitor 1 Circuit Control Register	VW1C	37
0037h	Voltage Monitor 2 Circuit Control Register	VW2C	38
0038h	Voltage Monitor 0 Circuit Control Register	VW0C	36
0039h			
003Ah			
003Bh			
003Ch			
003Dh			
003Eh			
003Fh			

Address	Register	Symbol	Page
0040h			
0041h			
0042h			
0043h			
0044h			
0045h			
0046h			
0047h	Timer RC Interrupt Control Register	TRCIC	103
0048h	Timer RD0 Interrupt Control Register	TRD0IC	103
0049h	Timer RD1 Interrupt Control Register	TRD1IC	103
004Ah			
004Bh	UART2 Transmit Interrupt Control Register	S2TIC	102
004Ch	UART2 Receive Interrupt Control Register	S2RIC	102
004Dh	Key Input Interrupt Control Register	KUPIC	102
004Eh	A/D Conversion Interrupt Control Register	ADIC	102
004Fh			
0050h			
0051h	UART0 Transmit Interrupt Control Register	S0TIC	102
0052h	UART0 Receive Interrupt Control Register	S0RIC	102
0053h			
0054h			
0055h			
0056h	Timer RA Interrupt Control Register	TRAIC	102
0057h			
0058h	Timer RB Interrupt Control Register	TRBIC	102
0059h	INT1 Interrupt Control Register	INT1IC	104
005Ah	INT3 Interrupt Control Register	INT3IC	104
005Bh			
005Ch			
005Dh	INT0 Interrupt Control Register	INT0IC	104
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:**

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



Address	Register	Symbol	Page
0080h			
0081h			
0082h			
0083h			
0084h			
0085h			
0086h			
0087h			
0088h			
0089h			
008Ah			
008Bh			
008Ch			
008Dh			
008Eh			
008Fh			
0090h			
0091h			
0092h			
0093h			
0094h			
0095h			
0096h			
0097h			
0098h			
0099h			
009Ah			
009Bh			
009Ch			
009Dh			
009Eh			
009Fh			
00A0h	UART0 Transmit/Receive Mode Register	U0MR	318
00A1h	UART0 Bit Rate Register	U0BRG	318
00A2h	UART0 Transmit Buffer Register	U0TB	319
00A3h			
00A4h	UART0 Transmit / Receive Control Register 0	U0C0	319
00A5h	UART0 Transmit / Receive Control Register 1	U0C1	320
00A6h	UART0 Receive Buffer Register	U0RB	320
00A7h			
00A8h			
00A9h			
00AAh			
00ABh			
00ACh			
00ADh			
00AEh			
00AFh			
00B0h			
00B1h			
00B2h			
00B3h			
00B4h			
00B5h			
00B6h			
00B7h			
00B8h			
00B9h			
00BAh			
00BBh			
00BCh			
00BDh			
00BEh			
00BFh			

Address	Register	Symbol	Page
00C0h	A/D Register	AD	351
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	351
00D5h			
00D6h	A/D Control Register 0	ADCON0	350
00D7h	A/D Control Register 1	ADCON1	351
00D8h			
00D9h			
00DAh			
00DBh			
00DCh			
00DDh			
00DEh			
00DFh			
00E0h	Port P0 Register	P0	55
00E1h	Port P1 Register	P1	55
00E2h	Port P0 Direction Register	PD0	54
00E3h	Port P1 Direction Register	PD1	54
00E4h	Port P2 Register	P2	55
00E5h	Port P3 Register	P3	55
00E6h	Port P2 Direction Register	PD2	54
00E7h	Port P3 Direction Register	PD3	54
00E8h	Port P4 Register	P4	55
00E9h			
00EAh	Port P4 Direction Register	PD4	54
00EBh			
00ECh			
00EDh			
00EEh			
00EFh			
00F0h			
00F1h			
00F2h			
00F3h			
00F4h	Port P2 Drive Capacity Control Register	P2DRR	55
00F5h	Pin Select Register 1	PINSR1	56
00F6h	Pin Select Register 2	PINSR2	56
00F7h	Pin Select Register 3	PINSR3	56
00F8h	Port Mode Register	PMR	56
00F9h	External Input Enable Register	INTEN	111
00FAh	INT Input Filter Select Register	INTF	112
00FBh	Key Input Enable Register	KIEN	115
00FCh	Pull-Up Control Register 0	PUR0	57
00FDh	Pull-Up Control Register 1	PUR1	57
00FEh			
00FFh			

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

Address	Register	Symbol	Page
0100h	Timer RA Control Register	TRACR	137
0101h	Timer RA I/O Control Register	TRAIOC	137, 139, 142, 144, 146, 149
0102h	Timer RA Mode Register	TRAMR	138
0103h	Timer RA Prescaler Register	TRAPRE	138
0104h	Timer RA Register	TRA	138
0105h	LIN Control Register 2	LINCR2	335
0106h	LIN Control Register	LINCR	335
0107h	LIN Status Register	LINST	336
0108h	Timer RB Control Register	TRBCR	154
0109h	Timer RB One-Shot Control Register	TRBOCR	154
010Ah	Timer RB I/O Control Register	TRBIOC	155, 157, 161, 164, 168
010Bh	Timer RB Mode Register	TRBMR	155
010Ch	Timer RB Prescaler Register	TRBPRE	156
010Dh	Timer RB Secondary Register	TRBSC	156
010Eh	Timer RB Primary Register	TRBPR	156
010Fh			
0110h			
0111h			
0112h			
0113h			
0114h			
0115h			
0116h			
0117h			
0118h			
0119h			
011Ah			
011Bh			
011Ch			
011Dh			
011Eh			
011Fh			
0120h	Timer RC Mode Register	TRCMR	177
0121h	Timer RC Control Register 1	TRCCR1	178, 201, 205, 210
0122h	Timer RC Interrupt Enable Register	TRCIER	179
0123h	Timer RC Status Register	TRCSR	180
0124h	Timer RC I/O Control Register 0	TRCIOR0	185, 194, 199
0125h	Timer RC I/O Control Register 1	TRCIOR1	185, 195, 200
0126h	Timer RC Counter	TRC	181
0127h			
0128h	Timer RC General Register A	TRCGRA	181
0129h			
012Ah	Timer RC General Register B	TRCGRB	181
012Bh			
012Ch	Timer RC General Register C	TRCGRC	181
012Dh			
012Eh	Timer RC General Register D	TRCGRD	181
012Fh			

NOTE:

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

Address	Register	Symbol	Page
0130h	Timer RC Control Register 2	TRCCR2	182
0131h	Timer RC Digital Filter Function Select Register	TRCDF	183
0132h	Timer RC Output Master Enable Register	TRCOER	184
0133h			
0134h			
0135h			
0136h			
0137h	Timer RD Start Register	TRDSTR	229, 243, 260, 273, 283, 297
0138h	Timer RD Mode Register	TRDMR	229, 243, 260, 273, 284, 298
0139h	Timer RD PWM Mode Register	TRDPMR	230, 244, 261
013Ah	Timer RD Function Control Register	TRDFCR	231, 245, 262, 274, 285, 299
013Bh	Timer RD Output Master Enable Register 1	TRDOER1	246, 263, 275, 286, 300
013Ch	Timer RD Output Master Enable Register 2	TRDOER2	246, 263, 275, 286, 300
013Dh	Timer RD Output Control Register	TRDOCR	247, 264, 301
013Eh	Timer RD Digital Filter Function Select Register 0	TRDDF0	232
013Fh	Timer RD Digital Filter Function Select Register 1	TRDDF1	232
0140h	Timer RD Control Register 0	TRDCR0	233, 248, 264, 276, 287, 302
0141h	Timer RD I/O Control Register A0	TRDIORA0	234, 249
0142h	Timer RD I/O Control Register C0	TRDIORC0	235, 250
0143h	Timer RD Status Register 0	TRDSR0	236, 251, 265, 277, 288, 303
0144h	Timer RD Interrupt Enable Register 0	TRDIER0	237, 252, 266, 278, 289, 304
0145h	Timer RD PWM Mode Output Level Control Register 0	TRDPOCR0	267
0146h	Timer RD Counter 0	TRD0	237, 253, 267, 278, 290, 304
0147h			
0148h	Timer RD General Register A0	TRDGRA0	238, 253, 268, 279, 290, 305
0149h			
014Ah	Timer RD General Register B0	TRDGRB0	238, 253, 268, 279, 290, 305
014Bh			
014Ch	Timer RD General Register C0	TRDGRC0	238, 253, 268, 279, 290, 305
014Dh			
014Eh	Timer RD General Register D0	TRDGRD0	238, 253, 268, 279, 290, 305
014Fh			
0150h	Timer RD Control Register 1	TRDCR1	233, 248, 264, 287
0151h	Timer RD I/O Control Register A1	TRDIORA1	234, 249
0152h	Timer RD I/O Control Register C1	TRDIORC1	235, 250
0153h	Timer RD Status Register 1	TRDSR1	236, 251, 265, 277, 288, 303
0154h	Timer RD Interrupt Enable Register 1	TRDIER1	237, 252, 266, 278, 289, 304
0155h	Timer RD PWM Mode Output Level Control Register 1	TRDPOCR1	267
0156h	Timer RD Counter 1	TRD1	237, 253, 267, 290
0157h			
0158h	Timer RD General Register A1	TRDGRA1	238, 253, 268, 279, 290, 305
0159h			
015Ah	Timer RD General Register B1	TRDGRB1	238, 253, 268, 279, 290, 305
015Bh			
015Ch	Timer RD General Register C1	TRDGRC1	238, 253, 268, 279, 290, 305
015Dh			
015Eh	Timer RD General Register D1	TRDGRD1	238, 253, 268, 279, 290, 305
015Fh			

Address	Register	Symbol	Page
0160h	UART2 Transmit/Receive Mode Register	U2MR	318
0161h	UART2 Bit Rate Register	U2BRG	318
0162h	UART2 Transmit Buffer Register	U2TB	319
0163h			
0164h	UART2 Transmit/Receive Control Register 0	U2C0	319
0165h	UART2 Transmit/Receive Control Register 1	U2C1	320
0166h	UART2 Receive Buffer Register	U2RB	320
0167h			
0168h			
0169h			
016Ah			
016Bh			
016Ch			
016Dh			
016Eh			
016Fh			
0170h			
0171h			
0172h			
0173h			
0174h			
0175h			
0176h			
0177h			
0178h			
0179h			
017Ah			
017Bh			
017Ch			
017Dh			
017Eh			
017Fh			
0180h			
0181h			
0182h			
0183h			
0184h			
0185h			
0186h			
0187h			
0188h			
0189h			
018Ah			
018Bh			
018Ch			
018Dh			
018Eh			
018Fh			
0190h			
0191h			
0192h			
0193h			
0194h			
0195h			
0196h			
0197h			
0198h			
0199h			
019Ah			
019Bh			
019Ch			
019Dh			
019Eh			
019Fh			

Address	Register	Symbol	Page
01A0h			
01A1h			
01A2h			
01A3h			
01A4h			
01A5h			
01A6h			
01A7h			
01A8h			
01A9h			
01AAh			
01ABh			
01ACh			
01ADh			
01AEh			
01AFh			
01B0h			
01B1h			
01B2h			
01B3h	Flash Memory Control Register 4	FMR4	371
01B4h			
01B5h	Flash Memory Control Register 1	FMR1	370
01B6h			
01B7h	Flash Memory Control Register 0	FMR0	367
01B8h			
01B9h			
01BAh			
01BBh			
01BCh			
01BDh			
01BEh			
01BFh			
FFFFh	Option Function Select Register	OFS	26, 126, 131, 365

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

## **1. Overview**

### **1.1 Features**

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

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

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

Furthermore, the R8C/2L Group has on-chip data flash (1 KB × 2 blocks).

The difference between the R8C/2K Group and R8C/2L Group is only the presence or absence of data flash. Their peripheral functions are the same.

#### **1.1.1 Applications**

Electronic household appliances, office equipment, audio equipment, consumer equipment, etc.

## 1.1.2 Specifications

Tables 1.1 and 1.2 outlines the Specifications for R8C/2K Group and Tables 1.3 and 1.4 outlines the Specifications for R8C/2L Group.

**Table 1.1 Specifications for R8C/2K Group (1)**

Item	Function	Specification
CPU	Central processing unit	R8C/Tiny series core <ul style="list-style-type: none"> <li>• Number of fundamental instructions: 89</li> <li>• Minimum instruction execution time:               <ul style="list-style-type: none"> <li>50 ns (f(XIN) = 20 MHz, VCC = 3.0 to 5.5 V)</li> <li>100 ns (f(XIN) = 10 MHz, VCC = 2.7 to 5.5 V)</li> <li>200 ns (f(XIN) = 5 MHz, VCC = 2.2 to 5.5 V)</li> </ul> </li> <li>• Multiplier: 16 bits × 16 bits → 32 bits</li> <li>• Multiply-accumulate instruction: 16 bits × 16 bits + 32 bits → 32 bits</li> <li>• Operation mode: Single-chip mode (address space: 1 Mbyte)</li> </ul>
Memory	ROM, RAM	Refer to <b>Table 1.5 Product List for R8C/2K Group</b> .
Power Supply Voltage Detection	Voltage detection circuit	<ul style="list-style-type: none"> <li>• Power-on reset</li> <li>• Voltage detection 3</li> </ul>
I/O Ports	Programmable I/O ports	<ul style="list-style-type: none"> <li>• Input-only: 3 pins</li> <li>• CMOS I/O ports: 25, selectable pull-up resistor</li> <li>• High current drive ports: 8</li> </ul>
Clock	Clock generation circuits	2 circuits: XIN clock oscillation circuit (with on-chip feedback resistor), On-chip oscillator (high-speed, low-speed) (high-speed on-chip oscillator has a frequency adjustment function) <ul style="list-style-type: none"> <li>• Oscillation stop detection: XIN clock oscillation stop detection function</li> <li>• Frequency divider circuit: Dividing selectable 1, 2, 4, 8, and 16</li> <li>• Low power consumption modes:               <ul style="list-style-type: none"> <li>Standard operating mode (high-speed clock, high-speed on-chip oscillator, low-speed on-chip oscillator), wait mode, stop mode</li> </ul> </li> </ul>
Interrupts		<ul style="list-style-type: none"> <li>• External: 4 sources, Internal: 15 sources, Software: 4 sources</li> <li>• Priority levels: 7 levels</li> </ul>
Watchdog Timer		15 bits × 1 (with prescaler), reset start selectable
Timer	Timer RA	8 bits × 1 (with 8-bit prescaler) Timer mode (period timer), pulse output mode (output level inverted every period), event counter mode, pulse width measurement mode, pulse period measurement mode
	Timer RB	8 bits × 1 (with 8-bit prescaler) Timer mode (period timer), programmable waveform generation mode (PWM output), programmable one-shot generation mode, programmable wait one-shot generation mode
	Timer RC	16 bits × 1 (with 4 capture/compare registers) Timer mode (input capture function, output compare function), PWM mode (output 3 pins), PWM2 mode (PWM output pin)
	Timer RD	16 bits × 2 (with 4 capture/compare registers) Timer mode (input capture function, output compare function), PWM mode (output 6 pins), reset synchronous PWM mode (output three-phase waveforms (6 pins), sawtooth wave modulation), complementary PWM mode (output three-phase waveforms (6 pins), triangular wave modulation), PWM3 mode (PWM output 2 pins with fixed period)

**Table 1.2 Specifications for R8C/2K Group (2)**

Item	Function	Specification
Serial Interface	UART0, UART2	Clock synchronous serial I/O/UART x 2
LIN Module		Hardware LIN: 1 (timer RA, UART0)
A/D Converter		10-bit resolution x 9 channels, includes sample and hold function
Flash Memory		<ul style="list-style-type: none"> <li>• Programming and erasure voltage: VCC = 2.7 to 5.5 V</li> <li>• Programming and erasure endurance: 100 times</li> <li>• Program security: ROM code protect, ID code check</li> <li>• Debug functions: On-chip debug, on-board flash rewrite function</li> </ul>
Operating Frequency/Supply Voltage		f(XIN) = 20 MHz (VCC = 3.0 to 5.5 V) f(XIN) = 10 MHz (VCC = 2.7 to 5.5 V) f(XIN) = 5 MHz (VCC = 2.2 to 5.5 V) (VCC = 2.7 to 5.5 V for A/D converter only)
Current consumption		Typ. 10 mA (VCC = 5.0 V, f(XIN) = 20 MHz) Typ. 6 mA (VCC = 3.0 V, f(XIN) = 10 MHz) Typ. 23 $\mu$ A (VCC = 3.0 V, wait mode, low-speed on-chip oscillator used) Typ. 0.7 $\mu$ A (VCC = 3.0 V, stop mode)
Operating Ambient Temperature		-20 to 85°C (N version) -40 to 85°C (D version) <sup>(1)</sup> -20 to 105°C (Y version) <sup>(2)</sup>
Package		32-pin LQFP <ul style="list-style-type: none"> <li>• Package code: PLQP0032GB-A (previous code: 32P6U-A)</li> </ul>

## NOTES:

1. Specify the D version if D version functions are to be used.
2. Please contact Renesas Technology sales offices for the Y version.

**Table 1.3 Specifications for R8C/2L Group (1)**

Item	Function	Specification
CPU	Central processing unit	R8C/Tiny series core <ul style="list-style-type: none"> <li>• Number of fundamental instructions: 89</li> <li>• Minimum instruction execution time: <ul style="list-style-type: none"> <li>50 ns (f(XIN) = 20 MHz, VCC = 3.0 to 5.5 V)</li> <li>100 ns (f(XIN) = 10 MHz, VCC = 2.7 to 5.5 V)</li> <li>200 ns (f(XIN) = 5 MHz, VCC = 2.2 to 5.5 V)</li> </ul> </li> <li>• Multiplier: 16 bits × 16 bits → 32 bits</li> <li>• Multiply-accumulate instruction: 16 bits × 16 bits + 32 bits → 32 bits</li> <li>• Operation mode: Single-chip mode (address space: 1 Mbyte)</li> </ul>
Memory	ROM, RAM	Refer to <b>Table 1.6 Product List for R8C/2L Group</b> .
Power Supply Voltage Detection	Voltage detection circuit	<ul style="list-style-type: none"> <li>• Power-on reset</li> <li>• Voltage detection 3</li> </ul>
I/O Ports	Programmable I/O ports	<ul style="list-style-type: none"> <li>• Input-only: 3 pins</li> <li>• CMOS I/O ports: 25, selectable pull-up resistor</li> <li>• High current drive ports: 8</li> </ul>
Clock	Clock generation circuits	<p>2 circuits: XIN clock oscillation circuit (with on-chip feedback resistor), On-chip oscillator (high-speed, low-speed) (high-speed on-chip oscillator has a frequency adjustment function)</p> <ul style="list-style-type: none"> <li>• Oscillation stop detection: XIN clock oscillation stop detection function</li> <li>• Frequency divider circuit: Dividing selectable 1, 2, 4, 8, and 16</li> <li>• Low power consumption modes: <ul style="list-style-type: none"> <li>Standard operating mode (high-speed clock, high-speed on-chip oscillator, low-speed on-chip oscillator), wait mode, stop mode</li> </ul> </li> </ul>
Interrupts		<ul style="list-style-type: none"> <li>• External: 4 sources, Internal: 15 sources, Software: 4 sources</li> <li>• Priority levels: 7 levels</li> </ul>
Watchdog Timer		15 bits × 1 (with prescaler), reset start selectable
Timer	Timer RA	8 bits × 1 (with 8-bit prescaler) Timer mode (period timer), pulse output mode (output level inverted every period), event counter mode, pulse width measurement mode, pulse period measurement mode
	Timer RB	8 bits × 1 (with 8-bit prescaler) Timer mode (period timer), programmable waveform generation mode (PWM output), programmable one-shot generation mode, programmable wait one-shot generation mode
	Timer RC	16 bits × 1 (with 4 capture/compare registers) Timer mode (input capture function, output compare function), PWM mode (output 3 pins), PWM2 mode (PWM output pin)
	Timer RD	16 bits × 2 (with 4 capture/compare registers) Timer mode (input capture function, output compare function), PWM mode (output 6 pins), reset synchronous PWM mode (output three-phase waveforms (6 pins), sawtooth wave modulation), complementary PWM mode (output three-phase waveforms (6 pins), triangular wave modulation), PWM3 mode (PWM output 2 pins with fixed period)

**Table 1.4 Specifications for R8C/2L Group (2)**

Item	Function	Specification
Serial Interface	UART0, UART2	Clock synchronous serial I/O/UART x 2
LIN Module		Hardware LIN: 1 (timer RA, UART0)
A/D Converter		10-bit resolution x 9 channels, includes sample and hold function
Flash Memory		<ul style="list-style-type: none"> <li>• Programming and erasure voltage: VCC = 2.7 to 5.5 V</li> <li>• Programming and erasure endurance: 10,000 times (data flash) 1,000 times (program ROM)</li> <li>• Program security: ROM code protect, ID code check</li> <li>• Debug functions: On-chip debug, on-board flash rewrite function</li> </ul>
Operating Frequency/Supply Voltage		f(XIN) = 20 MHz (VCC = 3.0 to 5.5 V) f(XIN) = 10 MHz (VCC = 2.7 to 5.5 V) f(XIN) = 5 MHz (VCC = 2.2 to 5.5 V) (VCC = 2.7 to 5.5 V for A/D converter only)
Current consumption		Typ. 10 mA (VCC = 5.0 V, f(XIN) = 20 MHz) Typ. 6 mA (VCC = 3.0 V, f(XIN) = 10 MHz) Typ. 23 $\mu$ A (VCC = 3.0 V, wait mode, low-speed on-chip oscillator used) Typ. 0.7 $\mu$ A (VCC = 3.0 V, stop mode)
Operating Ambient Temperature		-20 to 85°C (N version) -40 to 85°C (D version) <sup>(1)</sup> -20 to 105°C (Y version) <sup>(2)</sup>
Package		32-pin LQFP • Package code: PLQP0032GB-A (previous code: 32P6U-A)

## NOTES:

1. Specify the D version if D version functions are to be used.
2. Please contact Renesas Technology sales offices for the Y version.



## 1.2 Product List

Table 1.5 lists the Product List for R8C/2K Group, Figure 1.1 shows a Part Number, Memory Size, and Package of R8C/2K Group, Table 1.6 lists the Product List for R8C/2L Group, and Figure 1.2 shows a Part Number, Memory Size, and Package of R8C/2L Group.

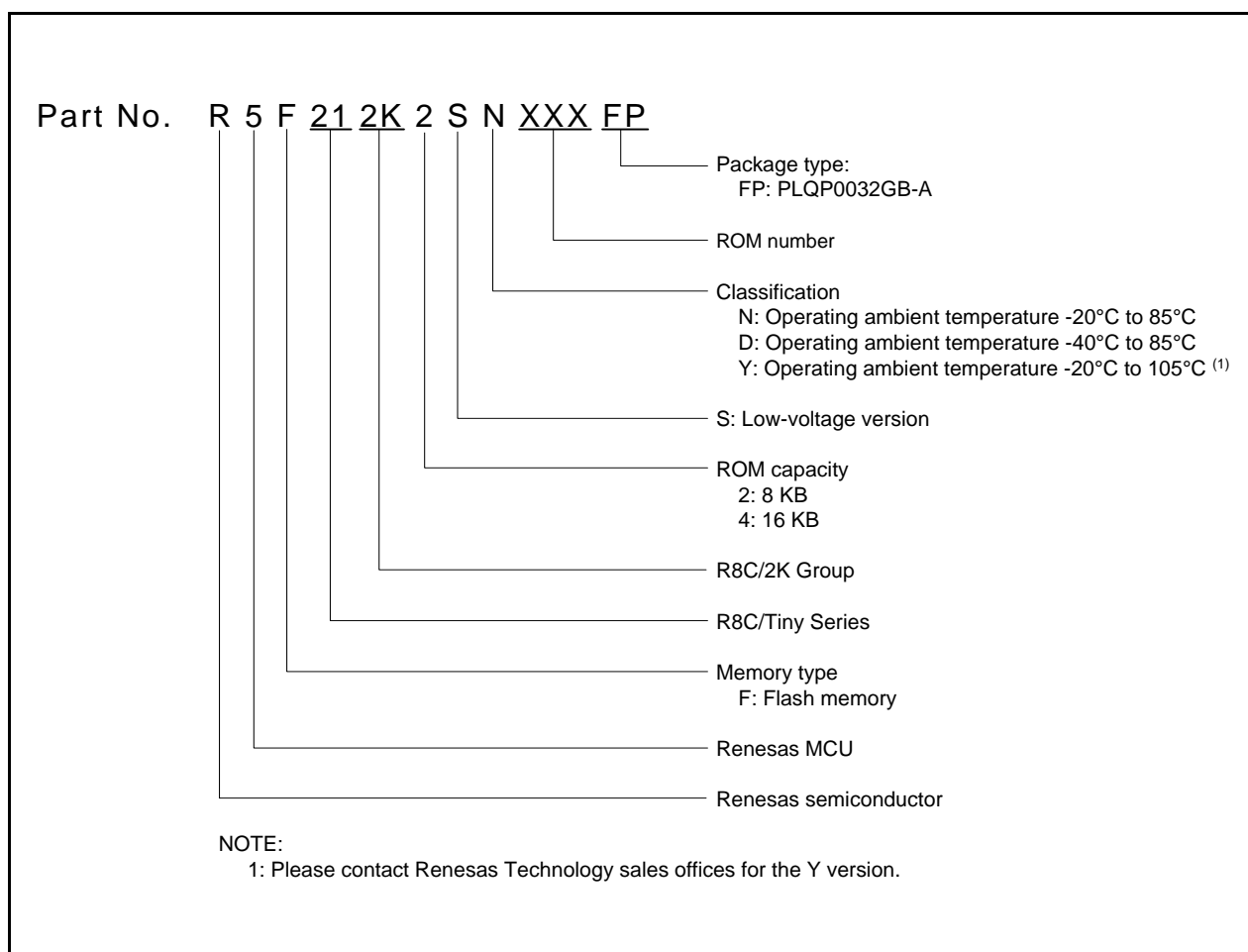
**Table 1.5 Product List for R8C/2K Group** **Current of Dec. 2007**

Part No.	ROM Capacity	RAM Capacity	Package Type	Remarks
R5F212K2SNFP	8 Kbytes	1 Kbyte	PLQP0032GB-A	N version
R5F212K4SNFP	16 Kbytes	1.5 Kbytes	PLQP0032GB-A	
R5F212K2SDFP	8 Kbytes	1 Kbyte	PLQP0032GB-A	D version
R5F212K4SDFP	16 Kbytes	1.5 Kbytes	PLQP0032GB-A	
R5F212K2SNXXXFP (D)	8 Kbytes	1 Kbyte	PLQP0032GB-A	N version Factory programming product <sup>(1)</sup>
R5F212K4SNXXXFP (D)	16 Kbytes	1.5 Kbytes	PLQP0032GB-A	
R5F212K2SDXXXFP (D)	8 Kbytes	1 Kbyte	PLQP0032GB-A	D version Factory programming product <sup>(1)</sup>
R5F212K4SDXXXFP (D)	16 Kbytes	1.5 Kbytes	PLQP0032GB-A	

(D): Under development

NOTE:

- The user ROM is programmed before shipment.



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

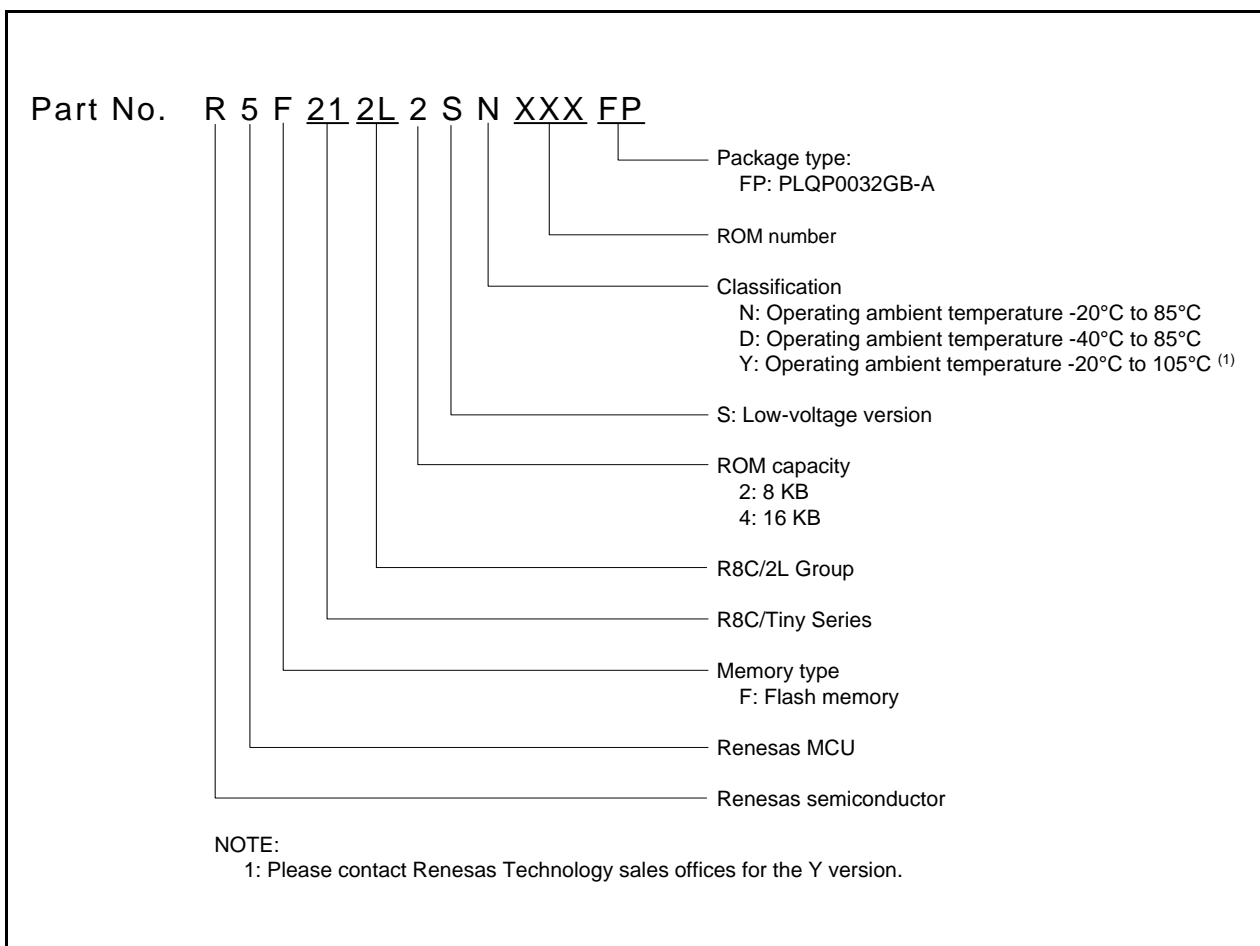
**Table 1.6 Product List for R8C/2L Group** **Current of Dec. 2007**

Part No.	ROM Capacity		RAM Capacity	Package Type	Remarks
	Program ROM	Data flash			
R5F212L2SNFP	8 Kbytes	1 Kbyte × 2	1 Kbyte	PLQP0032GB-A	N version
R5F212L4SNFP	16 Kbytes	1 Kbyte × 2	1.5 Kbytes	PLQP0032GB-A	
R5F212L2SDFP	8 Kbytes	1 Kbyte × 2	1 Kbyte	PLQP0032GB-A	D version
R5F212L4SDFP	16 Kbytes	1 Kbyte × 2	1.5 Kbytes	PLQP0032GB-A	
R5F212L2SNXXXFP (D)	8 Kbytes	1 Kbyte × 2	1 Kbyte	PLQP0032GB-A	N version Factory programming product <sup>(1)</sup>
R5F212L4SNXXXFP (D)	16 Kbytes	1 Kbyte × 2	1.5 Kbytes	PLQP0032GB-A	
R5F212L2SDXXXFP (D)	8 Kbytes	1 Kbyte × 2	1 Kbyte	PLQP0032GB-A	D version Factory programming product <sup>(1)</sup>
R5F212L4SDXXXFP (D)	16 Kbytes	1 Kbyte × 2	1.5 Kbytes	PLQP0032GB-A	

(D): Under development

NOTE:

1. The user ROM is programmed before shipment.



**Figure 1.2 Part Number, Memory Size, and Package of R8C/2L Group**

### 1.3 Block Diagram

Figure 1.3 shows a Block Diagram.

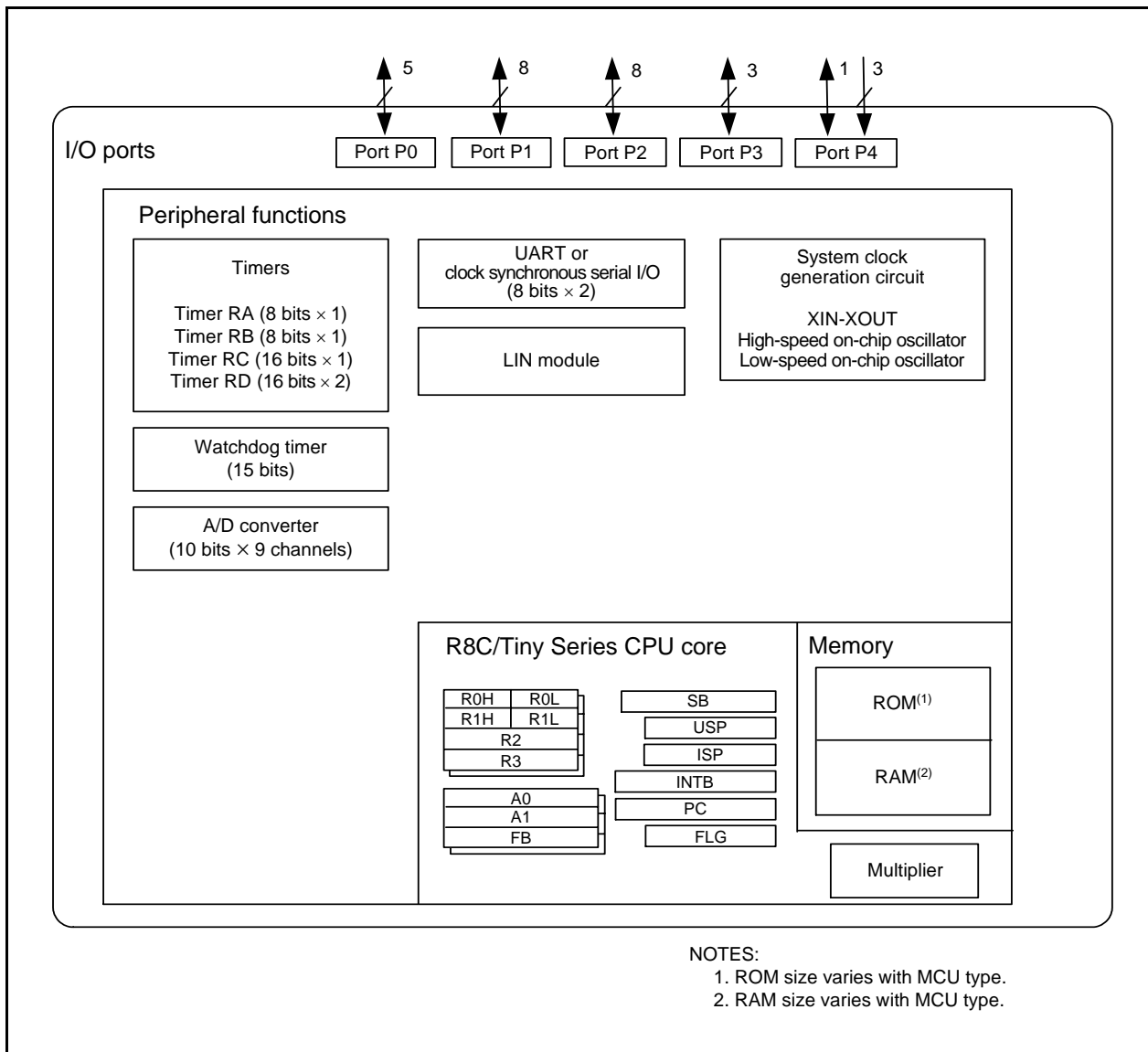
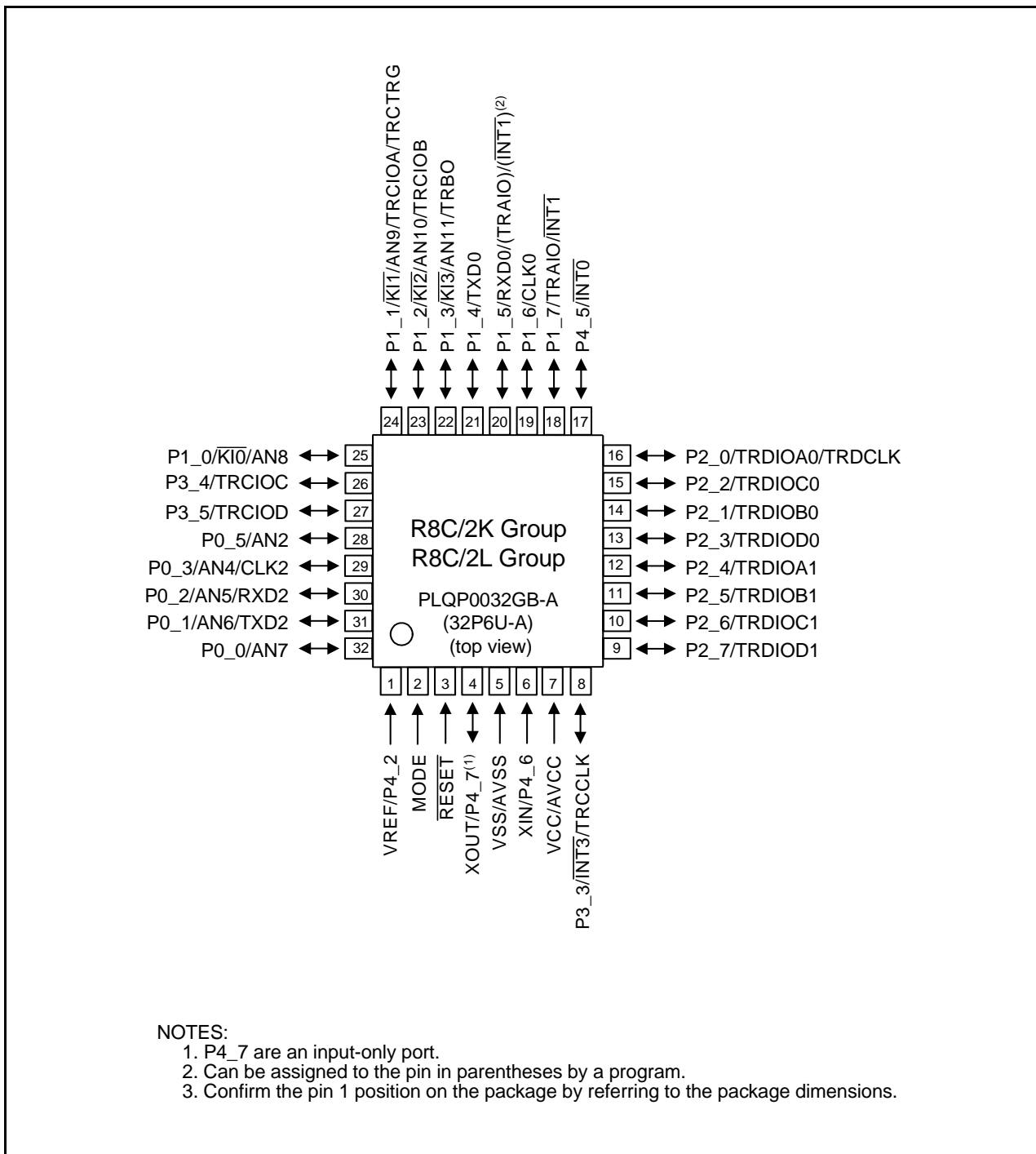


Figure 1.3 Block Diagram

### 1.4 Pin Assignment

Figure 1.4 shows the Pin Assignment (Top View). Table 1.7 outlines the Pin Name Information by Pin Number.



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

**Table 1.7 Pin Name Information by Pin Number**

Pin Number	Control Pin	Port	I/O Pin Functions for of Peripheral Modules			
			Interrupt	Timer	Serial Interface	A/D Converter
1	VREF	P4_2				
2	MODE					
3	RESET					
4	XOUT	P4_7				
5	VSS/AVSS					
6	XIN	P4_6				
7	VCC/AVCC					
8		P3_3	$\overline{\text{INT3}}$	TRCCLK		
9		P2_7		TRDIOD1		
10		P2_6		TRDIOC1		
11		P2_5		TRDIOB1		
12		P2_4		TRDIOA1		
13		P2_3		TRDIOD0		
14		P2_1		TRDIOB0		
15		P2_2		TRDIOC0		
16		P2_0		TRDIOA0/TRDCLK		
17		P4_5	$\overline{\text{INT0}}$			
18		P1_7	$\overline{\text{INT1}}$	TRAIO		
19		P1_6			CLK0	
20		P1_5	$(\overline{\text{INT1}})^{(1)}$	$(\text{TRAIO})^{(1)}$	RXD0	
21		P1_4			TXD0	
22		P1_3	$\overline{\text{KI3}}$	TRBO		AN11
23		P1_2	$\overline{\text{KI2}}$	TRCIOB		AN10
24		P1_1	$\overline{\text{KI1}}$	TRCIOA/TRCTRG		AN9
25		P1_0	$\overline{\text{KI0}}$			AN8
26		P3_4		TRCIOC		
27		P3_5		TRCIOD		
28		P0_5				AN2
29		P0_3			CLK2	AN4
30		P0_2			RXD2	AN5
31		P0_1			TXD2	AN6
32		P0_0				AN7

NOTE:

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

## 1.5 Pin Functions

Table 1.8 lists Pin Functions.

**Table 1.8 Pin Functions**

Item	Pin Name	I/O Type	Description
Power supply input	VCC, VSS	–	Apply 2.2 V to 5.5 V to the VCC pin. Apply 0 V to the VSS pin.
Analog power supply input	AVCC, AVSS	–	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 XIN clock generation circuit I/O. Connect a ceramic resonator or a crystal oscillator between the XIN and XOUT pins <sup>(1)</sup> . To use an external clock, input it to the XIN pin and leave the XOUT pin open.
XIN clock output	XOUT	O	
$\overline{\text{INT}}$ interrupt input	$\overline{\text{INT0}}$ , $\overline{\text{INT1}}$ , $\overline{\text{INT3}}$	I	$\overline{\text{INT}}$ interrupt input pins. $\overline{\text{INT0}}$ is timer RB, timer RC and timer RD 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
Timer RB	TRBO	O	Timer RB output pin
Timer RC	TRCLK	I	External clock input pin
	TRCTRG	I	External trigger input pin
	TRCIOA, TRCIOB, TRCIOC, TRCIOD	I/O	Timer RC I/O pins
Timer RD	TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1	I/O	Timer RD I/O pins
	TRDCLK	I	External clock input pin
Serial interface	CLK0, CLK2	I/O	Transfer clock I/O pins
	RXD0, RXD2	I	Serial data input pins
	TXD0, TXD2	O	Serial data output pins
Reference voltage input	VREF	I	Reference voltage input pin to A/D converter
A/D converter	AN2, AN4 to AN11	I	Analog input pins to A/D converter
I/O port	P0_0 to P0_3, P0_5, P1_0 to P1_7, P2_0 to P2_7, P3_3 to P3_5, P4_5,	I/O	CMOS I/O ports. Each port has an I/O select direction register, allowing each pin in the port to be directed for input or output individually. Any port set to input can be set to use a pull-up resistor or not by a program. P2_0 to P2_7 also function as LED drive ports.
Input port	P4_2, P4_6, P4_7	I	Input-only ports

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

NOTE:

1. Refer to the oscillator manufacturer for oscillation characteristics.

## 2. Central Processing Unit (CPU)

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

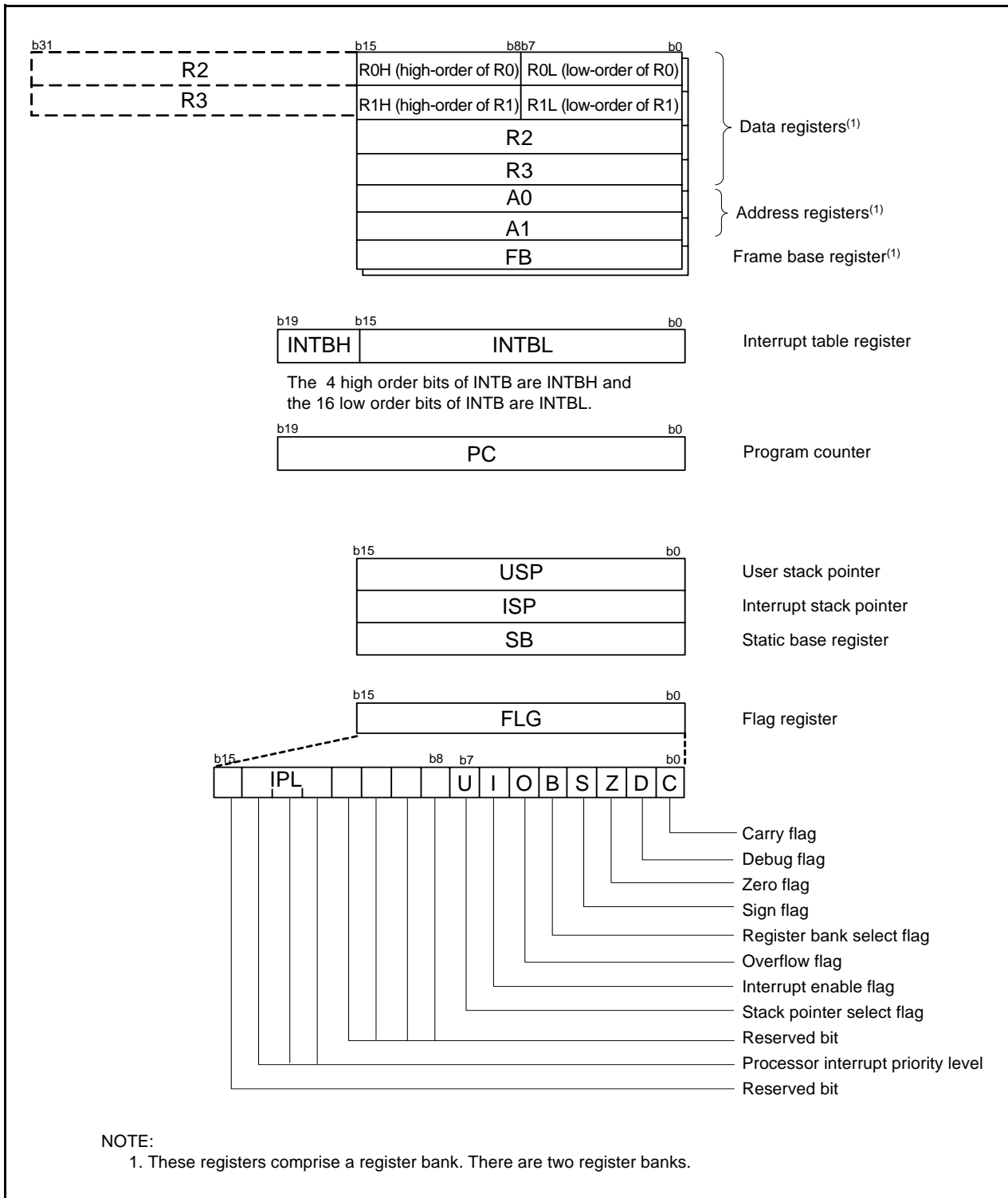


Figure 2.1 CPU Registers

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

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

## 2.2 Address Registers (A0 and A1)

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

## 2.3 Frame Base Register (FB)

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

## 2.4 Interrupt Table Register (INTB)

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

## 2.5 Program Counter (PC)

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

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

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

## 2.7 Static Base Register (SB)

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

## 2.8 Flag Register (FLG)

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

### 2.8.1 Carry Flag (C)

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

### 2.8.2 Debug Flag (D)

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

### 2.8.3 Zero Flag (Z)

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

### 2.8.4 Sign Flag (S)

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

### 2.8.5 Register Bank Select Flag (B)

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

### 2.8.6 Overflow Flag (O)

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



### **2.8.7 Interrupt Enable Flag (I)**

The I flag enables maskable interrupts.

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

### **2.8.8 Stack Pointer Select Flag (U)**

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

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

### **2.8.9 Processor Interrupt Priority Level (IPL)**

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

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

### **2.8.10 Reserved Bit**

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

### 3. Memory

#### 3.1 R8C/2K Group

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

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

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

The internal RAM is allocated higher addresses beginning with address 00400h. For example, a 1.5-Kbyte internal RAM area is allocated addresses 00400h to 009FFh. The internal RAM is used not only for storing data but also for calling subroutines and as stacks when interrupt requests are acknowledged.

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

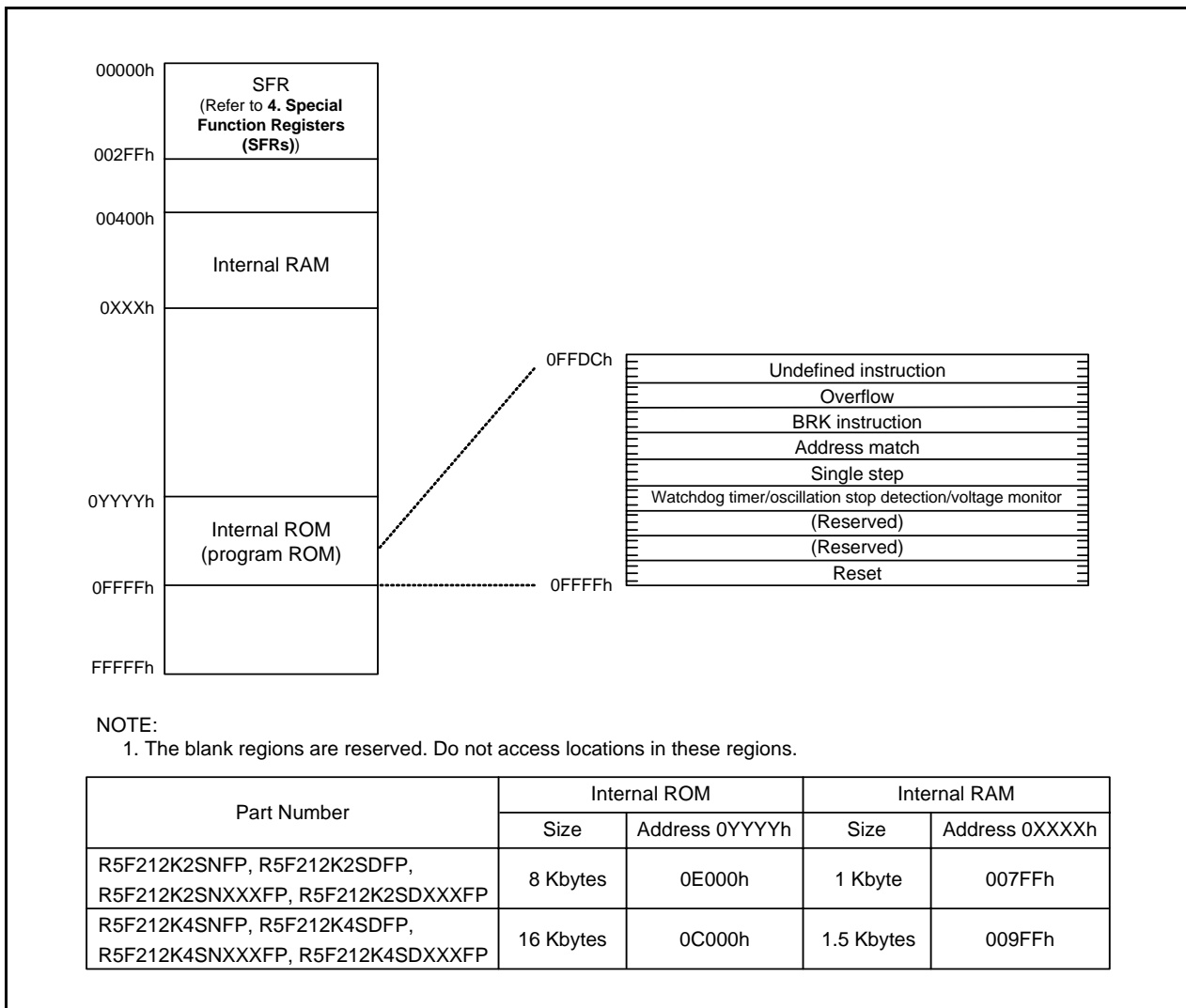


Figure 3.1 Memory Map of R8C/2K Group

### 3.2 R8C/2L Group

Figure 3.2 is a Memory Map of R8C/2L Group. The R8C/2L Group has 1 Mbyte of address space from addresses 00000h to FFFFFh.

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

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

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

The internal RAM area is allocated higher addresses, beginning with address 00400h. For example, a 1.5-Kbyte internal RAM is allocated addresses 00400h to 009FFh. The internal RAM is used not only for storing data but also for calling subroutines and as stacks when interrupt requests are acknowledged.

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

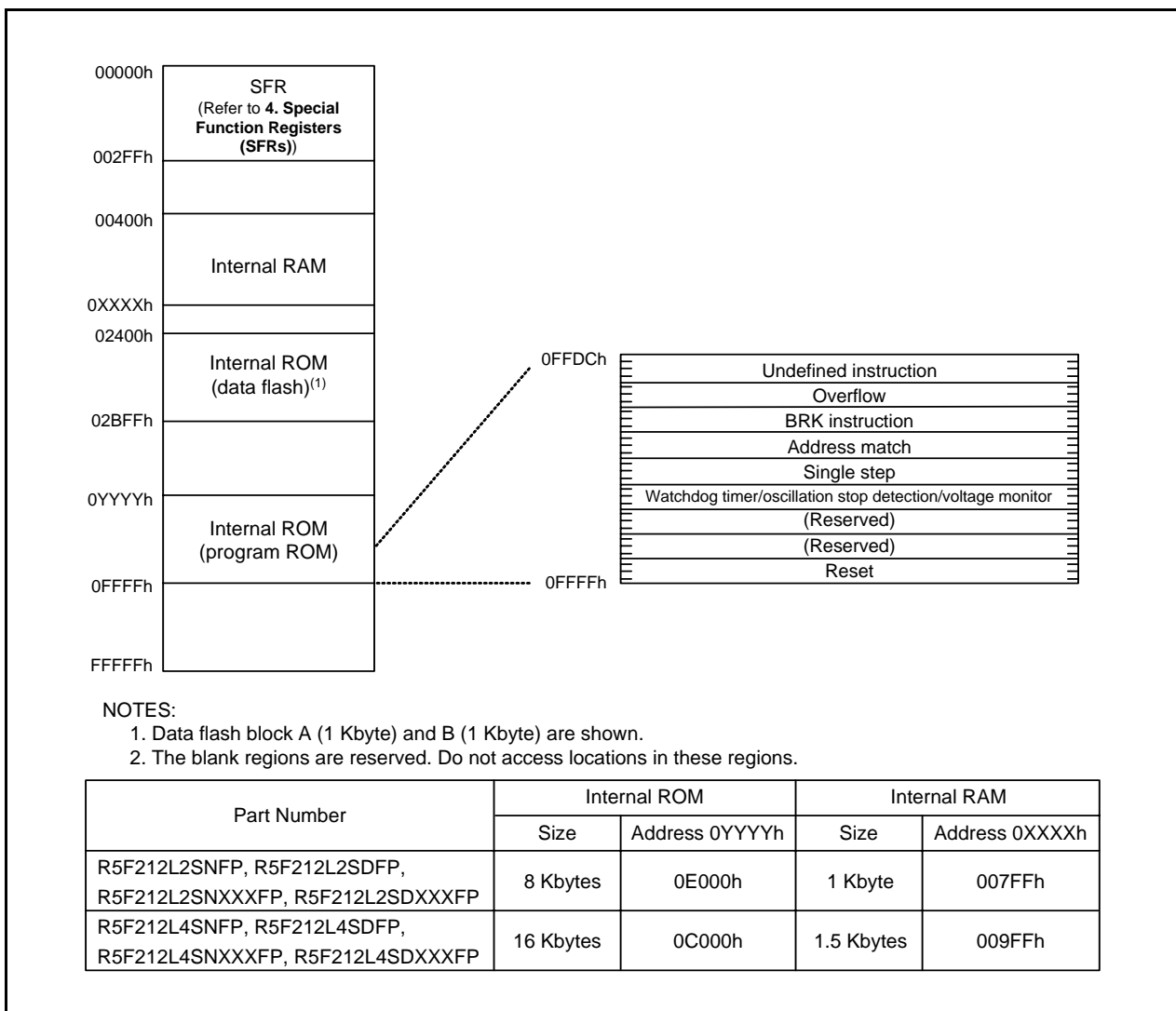


Figure 3.2 Memory Map of R8C/2L Group

## 4. Special Function Registers (SFRs)

An SFR (special function register) is a control register for a peripheral function. Tables 4.1 to 4.7 list the special function registers.

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

Address	Register	Symbol	After reset
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	00h
0005h	Processor Mode Register 1	PM1	00h
0006h	System Clock Control Register 0	CM0	01101000b
0007h	System Clock Control Register 1	CM1	00100000b
0008h			
0009h			
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			
0018h			
0019h			
001Ah			
001Bh			
001Ch	Count Source Protection Mode Register	CSPR	00h 10000000b <sup>(6)</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			
0027h			
0028h			
0029h			
002Ah			
002Bh	High-Speed On-Chip Oscillator Control Register 6	FRA6	When Shipping
002Ch	High-Speed On-Chip Oscillator Control Register 7	FRA7	When Shipping
0030h			
0031h	Voltage Detection Register 1 <sup>(2)</sup>	VCA1	00001000b
0032h	Voltage Detection Register 2 <sup>(2)</sup>	VCA2	00h <sup>(3)</sup> 00100000b <sup>(4)</sup>
0033h			
0034h			
0035h			
0036h	Voltage Monitor 1 Circuit Control Register <sup>(5)</sup>	VW1C	00001000b
0037h	Voltage Monitor 2 Circuit Control Register <sup>(5)</sup>	VW2C	00h
0038h	Voltage Monitor 0 Circuit Control Register <sup>(2)</sup>	VW0C	0000X000b <sup>(3)</sup> 0100X001b <sup>(4)</sup>
0039h			
003Ah			
003Eh			
003Fh			

X: Undefined

NOTES:

- The blank regions are reserved. Do not access locations in these regions.
- Software reset, watchdog timer reset, voltage monitor 1 reset, or voltage monitor 2 reset do not affect this register.
- The LVD00N bit in the OFS register is set to 1 and hardware reset.
- Power-on reset, voltage monitor 0 reset, or the LVD00N bit in the OFS register is set to 0 and hardware reset.
- Software reset, watchdog timer reset, voltage monitor 1 reset, or voltage monitor 2 reset do not affect b2 and b3.
- The CSPROINI bit in the OFS register is set to 0.

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

Address	Register	Symbol	After reset
0040h			
0041h			
0042h			
0043h			
0044h			
0045h			
0046h			
0047h	Timer RC Interrupt Control Register	TRCIC	XXXXX000b
0048h	Timer RD0 Interrupt Control Register	TRD0IC	XXXXX000b
0049h	Timer RD1 Interrupt Control Register	TRD1IC	XXXXX000b
004Ah			
004Bh	UART2 Transmit Interrupt Control Register	S2TIC	XXXXX000b
004Ch	UART2 Receive Interrupt Control Register	S2RIC	XXXXX000b
004Dh	Key Input Interrupt Control Register	KUPIC	XXXXX000b
004Eh	A/D Conversion Interrupt Control Register	ADIC	XXXXX000b
004Fh			
0050h			
0051h	UART0 Transmit Interrupt Control Register	S0TIC	XXXXX000b
0052h	UART0 Receive Interrupt Control Register	S0RIC	XXXXX000b
0053h			
0054h			
0055h			
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

NOTE:

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

**Table 4.3 SFR Information (3)<sup>(1)</sup>**

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			
00A9h			
00AAh			
00ABh			
00ACh			
00ADh			
00AEh			
00AFh			
00B0h			
00B1h			
00B2h			
00B3h			
00B4h			
00B5h			
00B6h			
00B7h			
00B8h			
00B9h			
00BAh			
00BBh			
00BCh			
00BDh			
00BEh			
00BFh			

X: Undefined

NOTE:

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

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

Address	Register	Symbol	After reset
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			
00EDh			
00EEh			
00EFh			
00F0h			
00F1h			
00F2h			
00F3h			
00F4h	Port P2 Drive Capacity Control Register	P2DRR	00h
00F5h	Pin Select Register 1	PINSR1	XXh
00F6h	Pin Select Register 2	PINSR2	XXh
00F7h	Pin Select Register 3	PINSR3	XXh
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	XX000000b
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	LIN Control Register 2	LINCR2	00h
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 Register	TRBPR	FFh
010Fh			
0110h			
0111h			
0112h			
0113h			
0114h			
0115h			
0116h			
0117h			
0118h			
0119h			
011Ah			
011Bh			
011Ch			
011Dh			
011Eh			
011Fh			
0120h	Timer RC Mode Register	TRCMR	01001000b
0121h	Timer RC Control Register 1	TRCCR1	00h
0122h	Timer RC Interrupt Enable Register	TRCIER	01110000b
0123h	Timer RC Status Register	TRCSR	01110000b
0124h	Timer RC I/O Control Register 0	TRCIOR0	10001000b
0125h	Timer RC I/O Control Register 1	TRCIOR1	10001000b
0126h	Timer RC Counter	TRC	00h
0127h			00h
0128h	Timer RC General Register A	TRCGRA	FFh
0129h			FFh
012Ah	Timer RC General Register B	TRCGRB	FFh
012Bh			FFh
012Ch	Timer RC General Register C	TRCGRC	FFh
012Dh			FFh
012Eh	Timer RC General Register D	TRCGRD	FFh
012Fh			FFh
0130h	Timer RC Control Register 2	TRCCR2	00011111b
0131h	Timer RC Digital Filter Function Select Register	TRCDF	00h
0132h	Timer RC Output Master Enable Register	TRCOER	01111111b
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

## 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
0160h	UART2 Transmit/Receive Mode Register	U2MR	00h
0161h	UART2 Bit Rate Register	U2BRG	XXh
0162h	UART2 Transmit Buffer Register	U2TB	XXh
0163h			XXh
0164h	UART2 Transmit/Receive Control Register 0	U2C0	00001000b
0165h	UART2 Transmit/Receive Control Register 1	U2C1	00000010b
0166h	UART2 Receive Buffer Register	U2RB	XXh
0167h			XXh
0168h			
0169h			
016Ah			
016Bh			
016Ch			
016Dh			
016Eh			
016Fh			
0170h			
0171h			
0172h			
0173h			
0174h			
0175h			
0176h			
0177h			
0178h			
0179h			
017Ah			
017Bh			
017Ch			
017Dh			
017Eh			
017Fh			

X: Undefined

NOTE:

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

**Table 4.7 SFR Information (7)<sup>(1)</sup>**

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

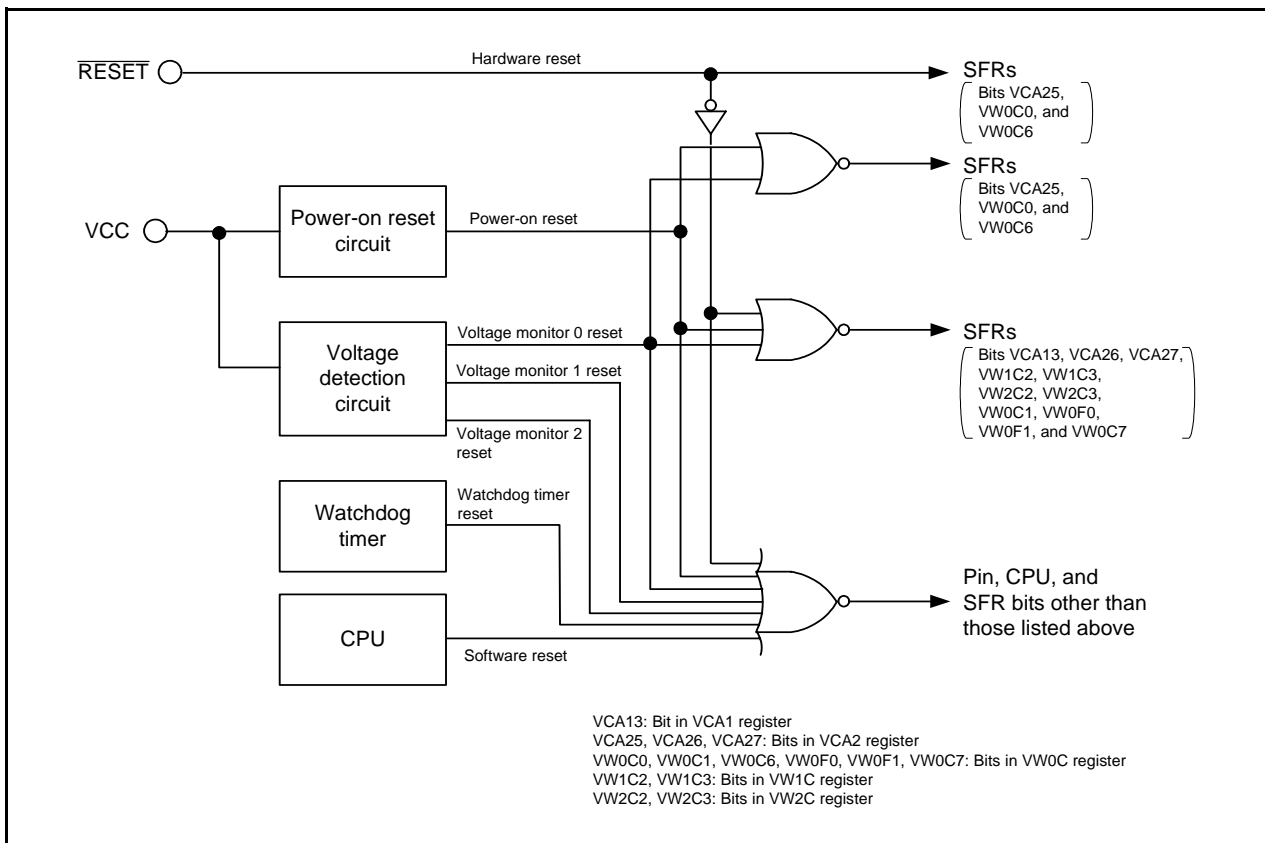
## 5. Resets

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

Table 5.1 lists the Reset Names and Sources.

**Table 5.1 Reset Names and Sources**

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

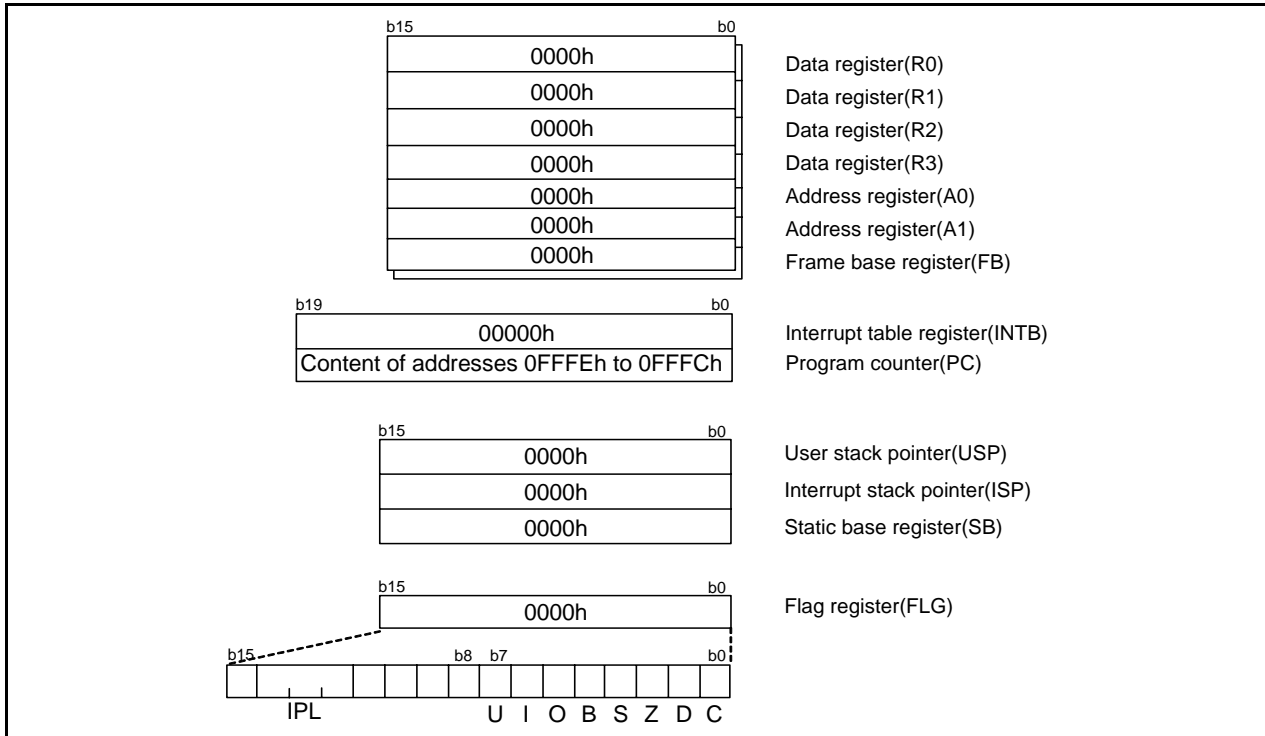


**Figure 5.1 Block Diagram of Reset Circuit**

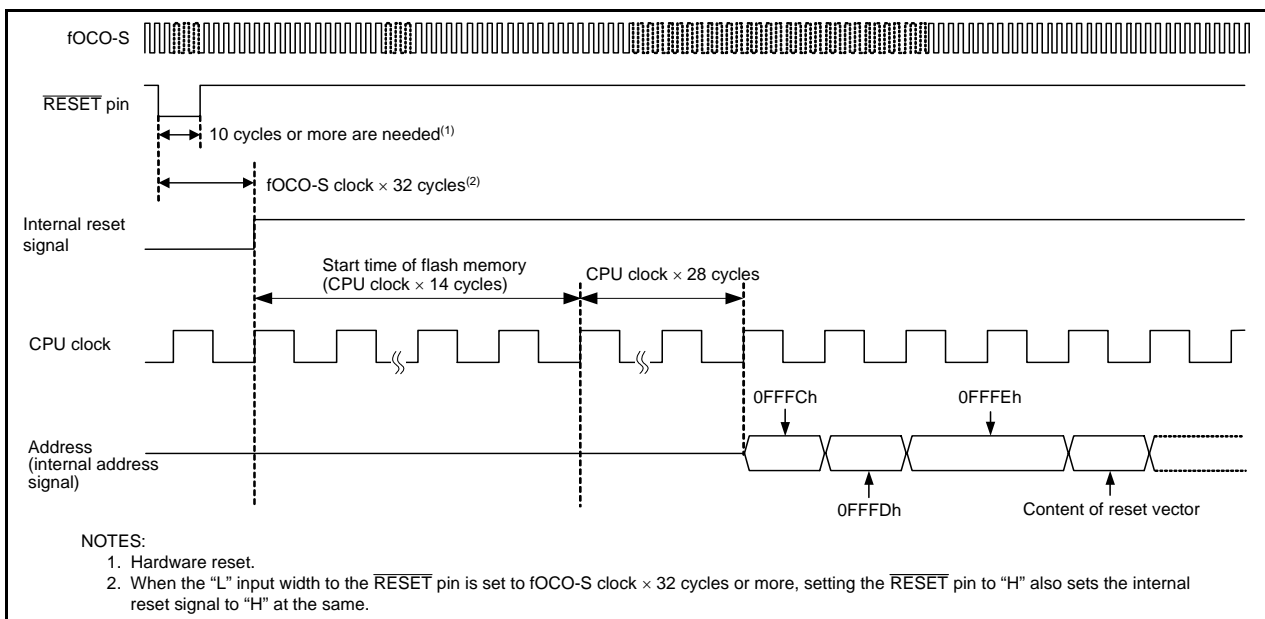
Table 5.2 shows the Pin Functions while  $\overline{\text{RESET}}$  Pin Level is “L”, Figure 5.2 shows the CPU Register Status after Reset, Figure 5.3 shows the Reset Sequence, and Figure 5.4 shows the OFS Register.

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

Pin Name	Pin Functions
P0_0 to P0_3, P0_5	Input port
P1, P2	Input port
P3_3 to P3_5	Input port
P4_2, P4_5 to P4_7	Input port



**Figure 5.2 CPU Register Status after Reset**



**Figure 5.3 Reset Sequence**

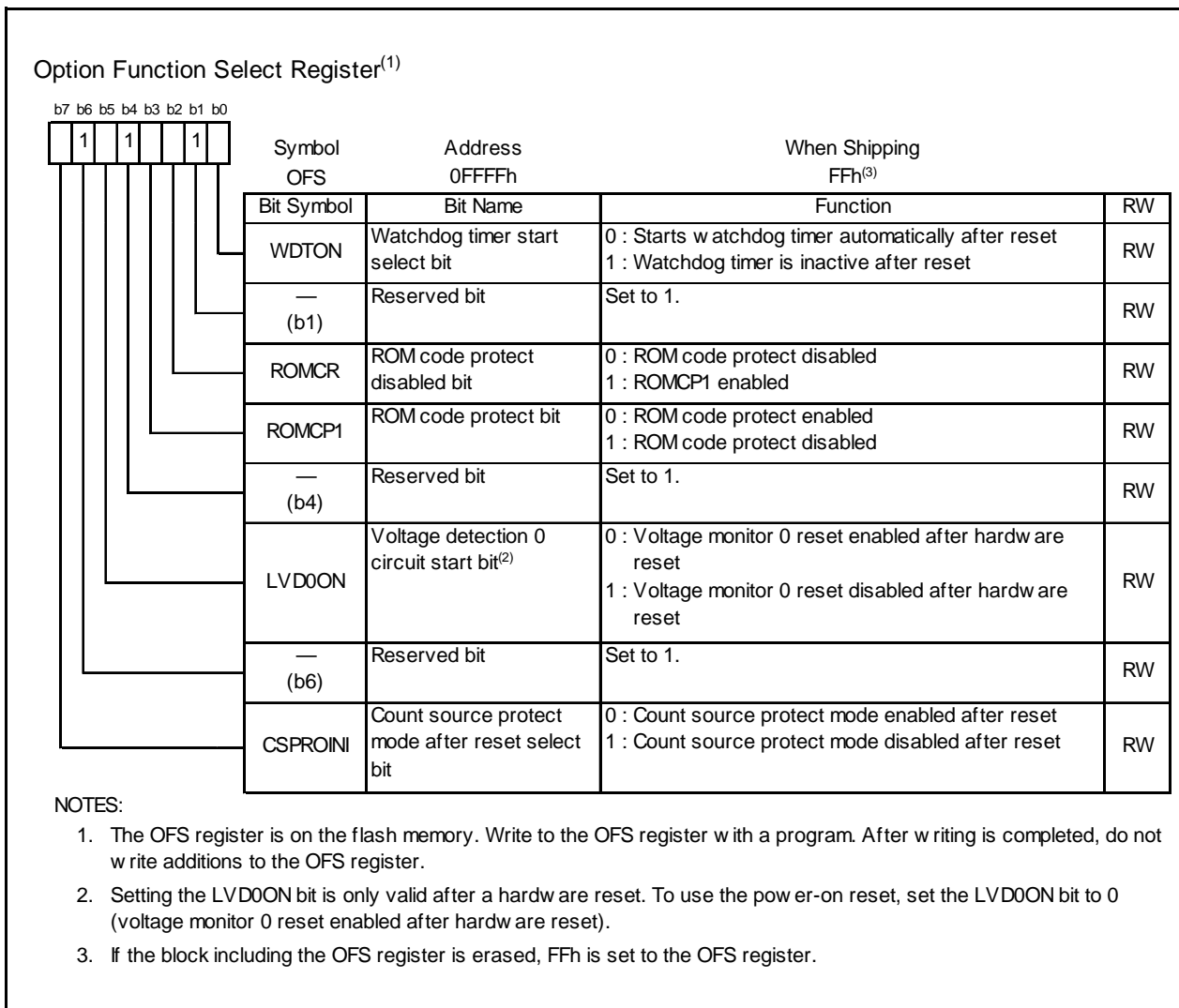


Figure 5.4 OFS Register

## 5.1 Hardware Reset

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

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

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

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

### 5.1.1 When Power Supply is Stable

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

### 5.1.2 Power On

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

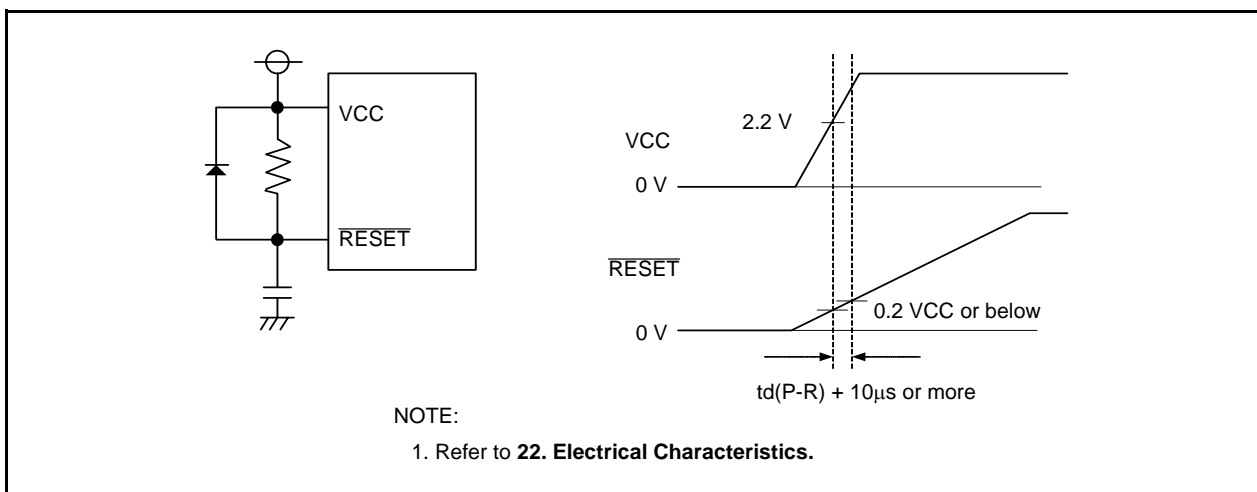


Figure 5.5 Example of Hardware Reset Circuit and Operation

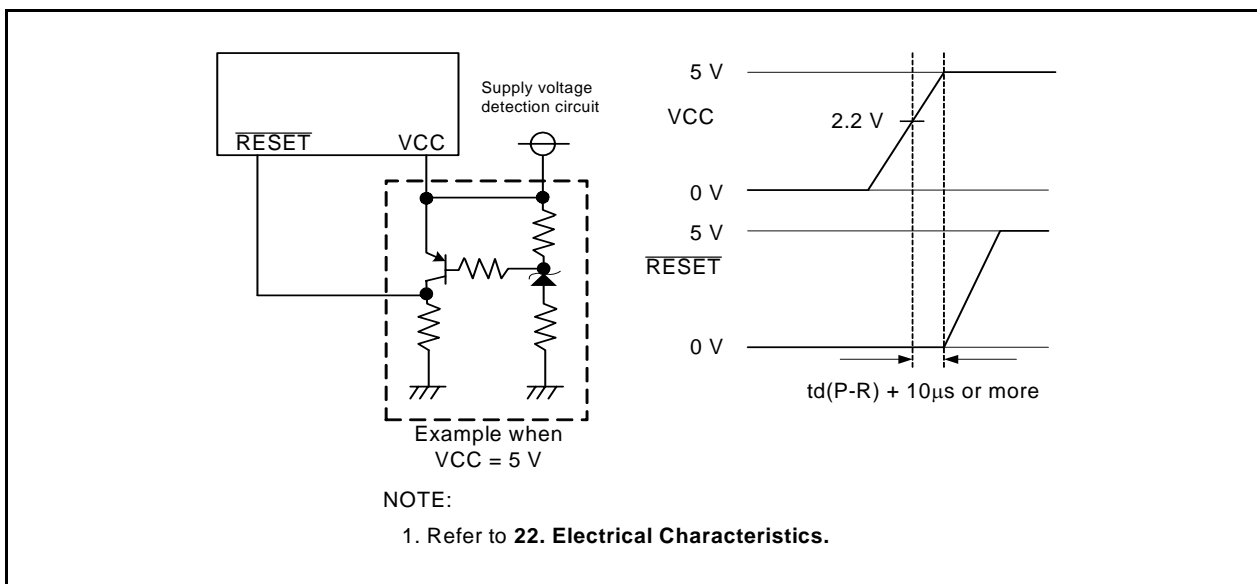


Figure 5.6 Example of Hardware Reset Circuit (Usage Example of External Supply Voltage Detection Circuit) and Operation

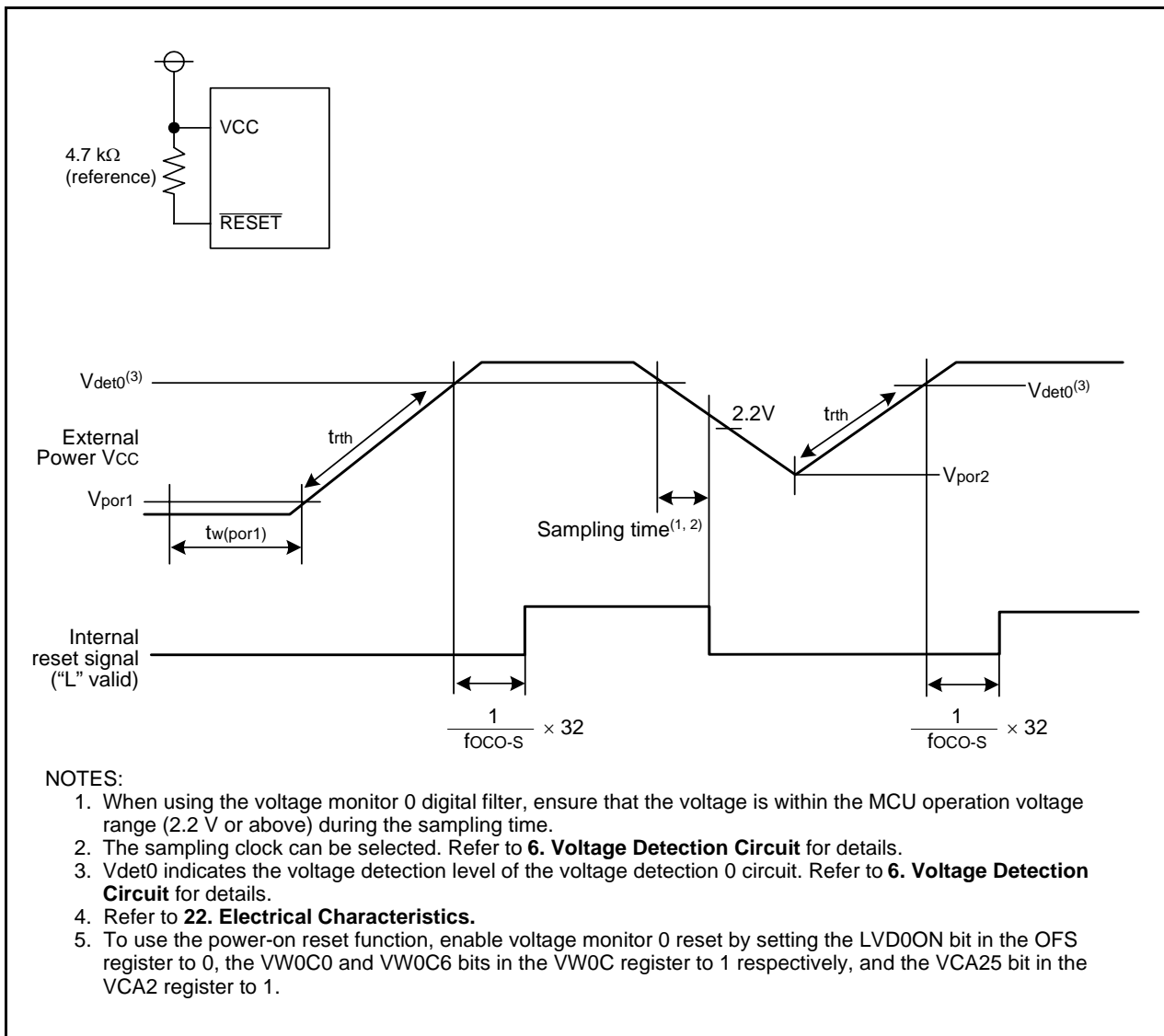
## 5.2 Power-On Reset Function

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

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

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

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



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



### 5.3 Voltage Monitor 0 Reset

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

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

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

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

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

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

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

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

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

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

### 5.4 Voltage Monitor 1 Reset

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

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

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

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

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

### 5.5 Voltage Monitor 2 Reset

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

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

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

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

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

## 5.6 Watchdog Timer Reset

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

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

The internal RAM is not reset. When the watchdog timer underflows, the contents of internal RAM are undefined. Refer to **15. Watchdog Timer** for details of the watchdog timer.

## 5.7 Software Reset

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

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

The internal RAM is not reset.

## 6. Voltage Detection Circuit

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

Table 6.1 lists the Specifications of Voltage Detection Circuit and Figures 6.1 to 6.4 show the Block Diagrams. Figures 6.5 to 6.8 show the Associated Registers.

**Table 6.1 Specifications of Voltage Detection Circuit**

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

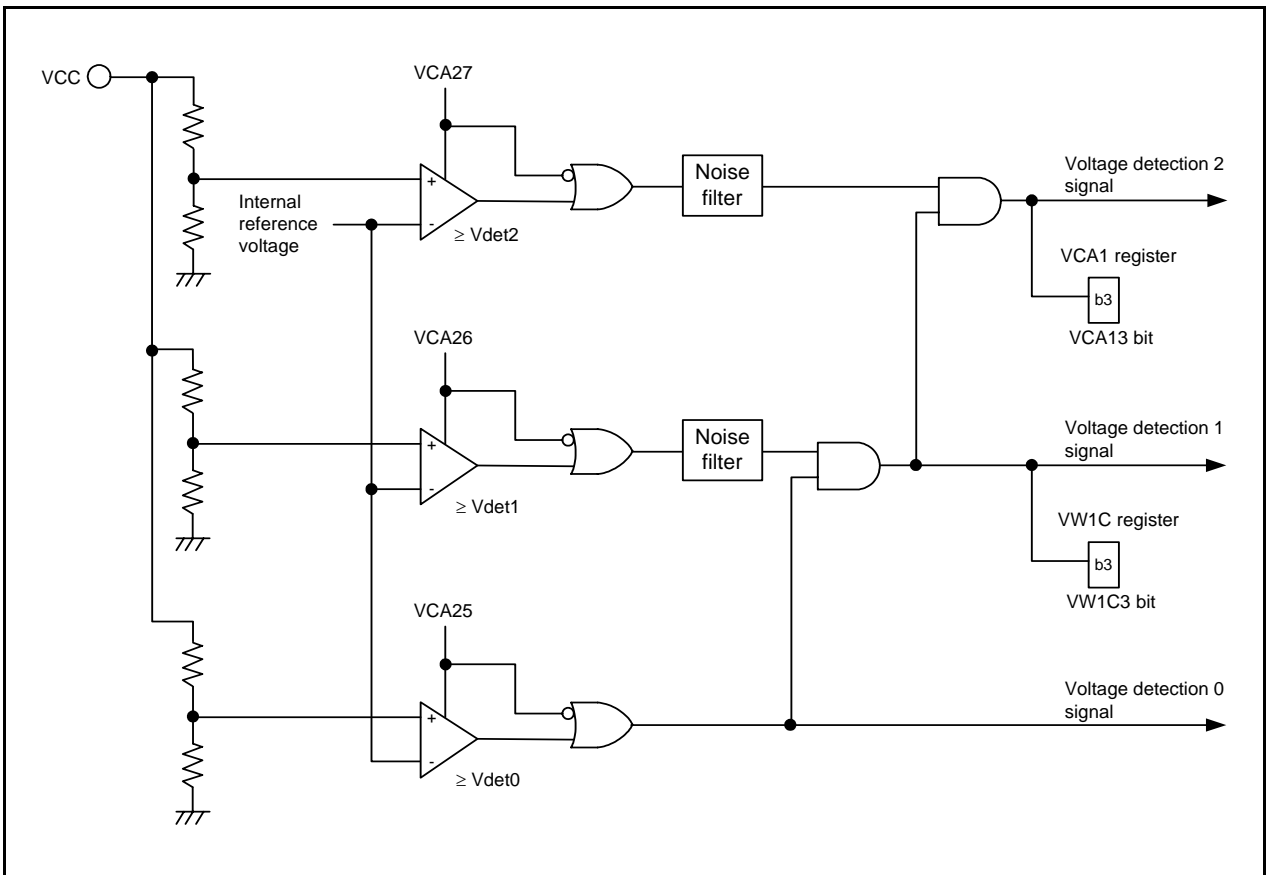


Figure 6.1 Block Diagram of Voltage Detection Circuit

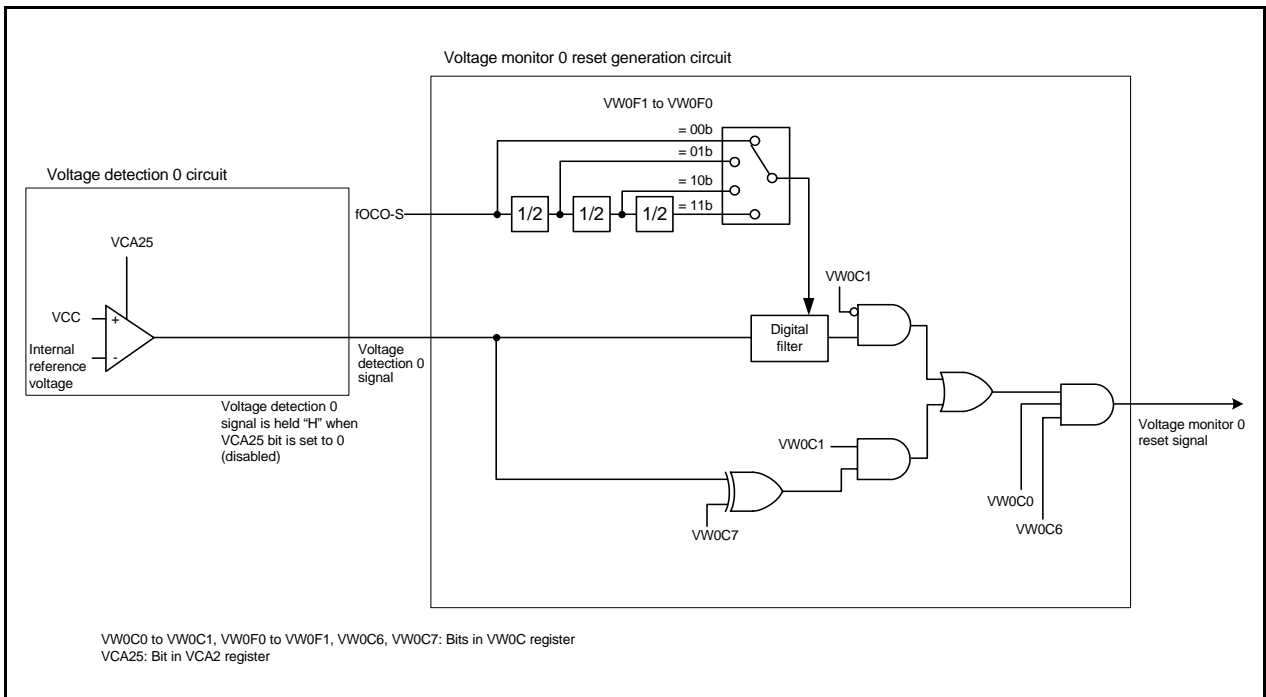


Figure 6.2 Block Diagram of Voltage Monitor 0 Reset Generation Circuit

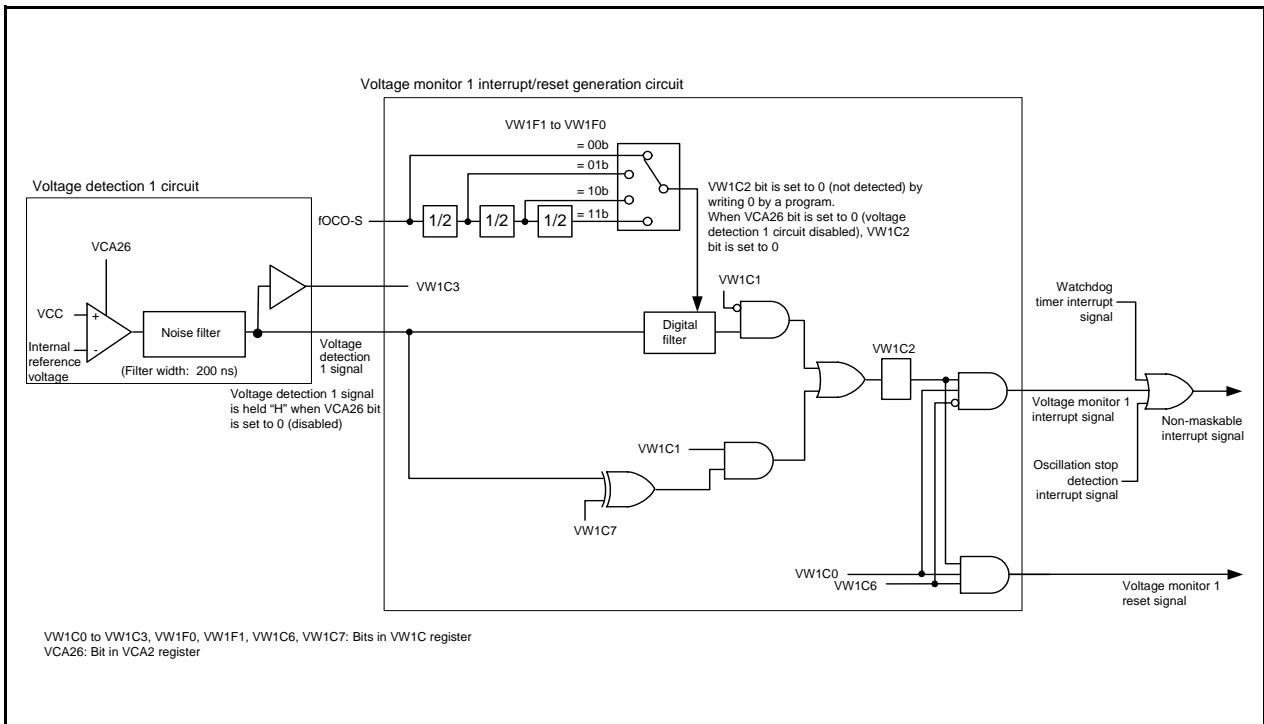


Figure 6.3 Block Diagram of Voltage Monitor 1 Interrupt/Reset Generation Circuit

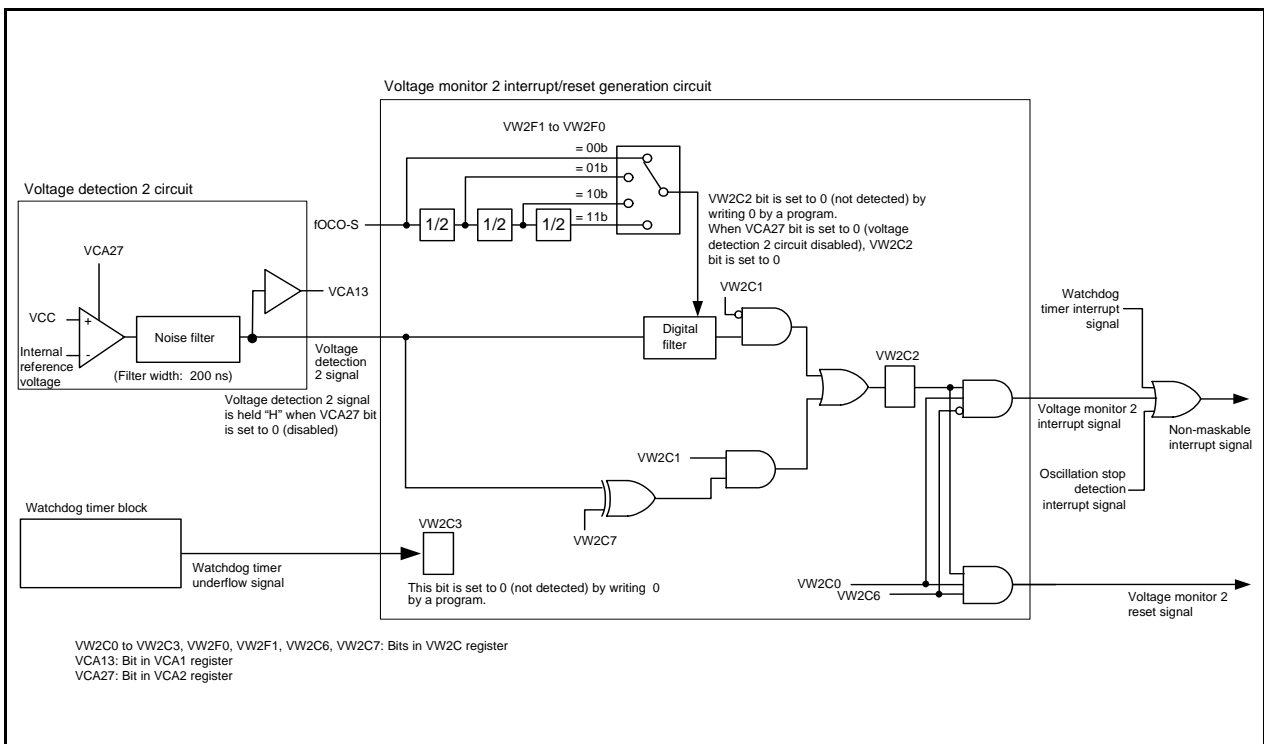


Figure 6.4 Block Diagram of Voltage Monitor 2 Interrupt/Reset Generation Circuit

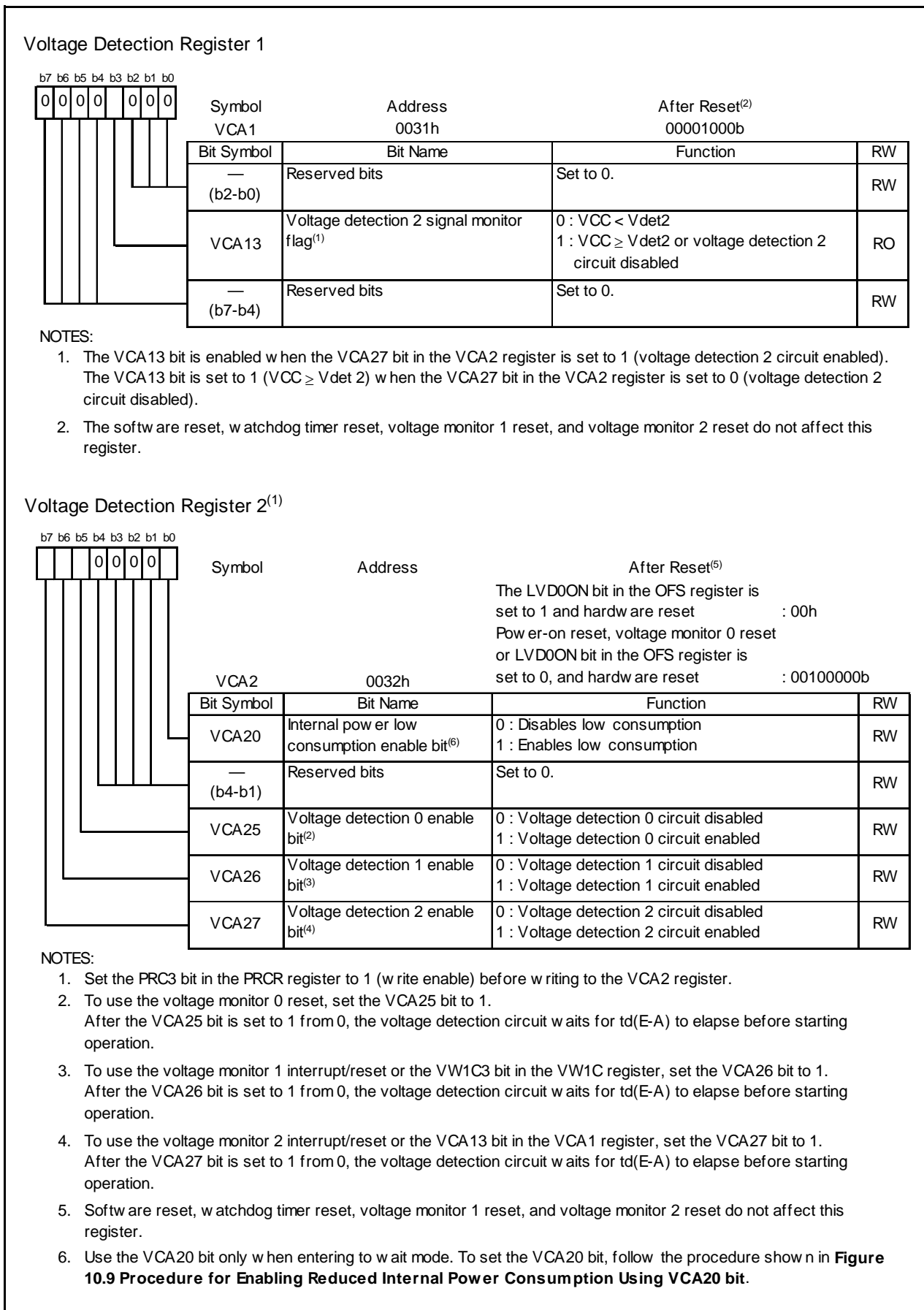


Figure 6.5 Registers VCA1 and VCA2

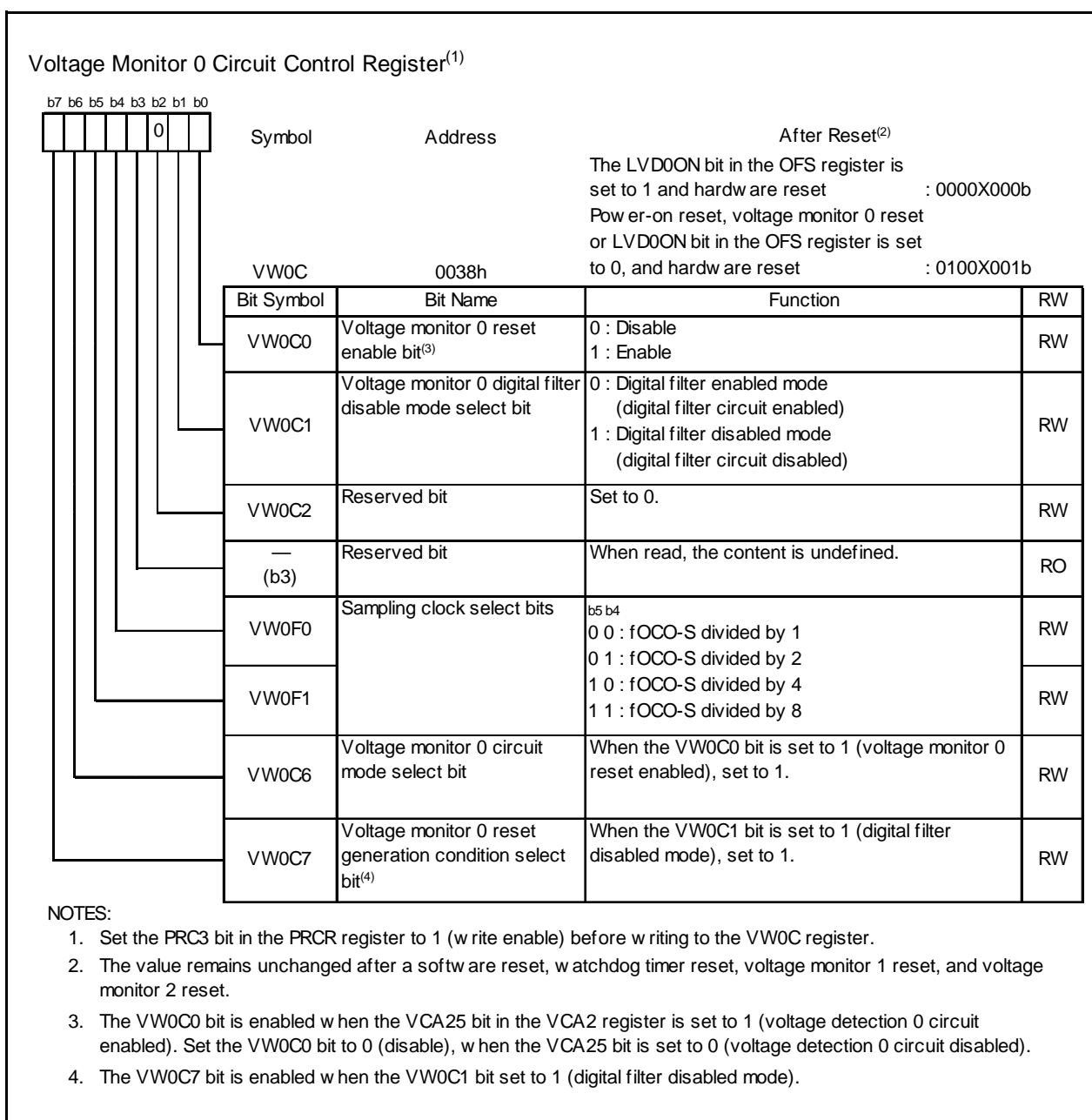


Figure 6.6 VW0C Register

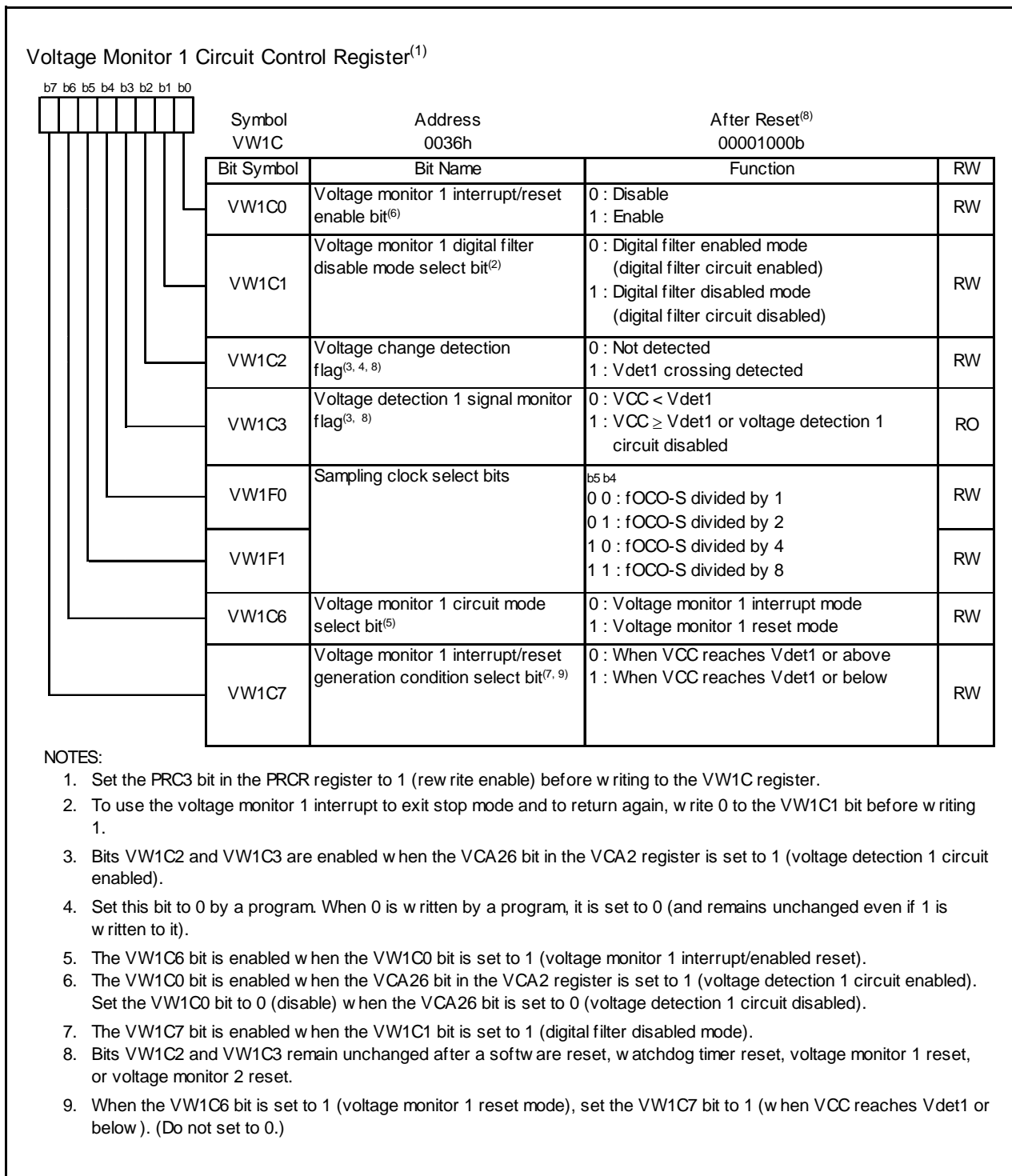


Figure 6.7 VW1C Register



Voltage Monitor 2 Circuit Control Register<sup>(1)</sup>

Bit	Symbol	Address	After Reset <sup>(8)</sup>	RW
b7	VW2C7	0037h	00h	RW
b6	VW2C6			
b5	VW2F1			
b4	VW2F0			
b3	VW2C3			
b2	VW2C2			
b1	VW2C1			
b0	VW2C0			
Bit Symbol	Bit Name	Function	RW	
VW2C0	Voltage monitor 2 interrupt/reset enable bit <sup>(6)</sup>	0 : Disable 1 : Enable	RW	
VW2C1	Voltage monitor 2 digital filter disable mode select bit <sup>(2)</sup>	0 : Digital filter enabled mode (digital filter circuit enabled) 1 : Digital filter disabled mode (digital filter circuit disabled)	RW	
VW2C2	Voltage change detection flag <sup>(3, 4, 8)</sup>	0 : Not detected 1 : VCC has crossed Vdet2	RW	
VW2C3	WDT detection flag <sup>(4, 8)</sup>	0 : Not detected 1 : Detected	RW	
VW2F0	Sampling clock select bits	b5 b4 0 0 : fOCO-S divided by 1 0 1 : fOCO-S divided by 2 1 0 : fOCO-S divided by 4 1 1 : fOCO-S divided by 8	RW	
VW2F1			RW	
VW2C6	Voltage monitor 2 circuit mode select bit <sup>(5)</sup>	0 : Voltage monitor 2 interrupt mode 1 : Voltage monitor 2 reset mode	RW	
VW2C7	Voltage monitor 2 interrupt/reset generation condition select bit <sup>(7, 9)</sup>	0 : When VCC reaches Vdet2 or above 1 : When VCC reaches Vdet2 or below	RW	

NOTES:

- Set the PRC3 bit in the PRCR register to 1 (write enable) before writing to the VW2C register.
- To use the voltage monitor 2 interrupt to exit stop mode and to return again, write 0 to the VW2C1 bit before writing 1.
- The VW2C2 bit is enabled when the VCA27 bit in the VCA2 register is set to 1 (voltage detection 2 circuit enabled).
- Set this bit to 0 by a program. When 0 is written by a program, it is set to 0 (and remains unchanged even if 1 is written to it).
- The VW2C6 bit is enabled when the VW2C0 bit is set to 1 (voltage monitor 2 interrupt/enables reset).
- The VW2C0 bit is enabled when the VCA27 bit in the VCA2 register is set to 1 (voltage detection 2 circuit enabled). Set the VW2C0 bit to 0 (disable) when the VCA27 bit is set to 0 (voltage detection 2 circuit disabled).
- The VW2C7 bit is enabled when the VW2C1 bit is set to 1 (digital filter disabled mode).
- Bits VW2C2 and VW2C3 remain unchanged after a software reset, watchdog timer reset, voltage monitor 1 reset, or voltage monitor 2 reset.
- When the VW2C6 bit is set to 1 (voltage monitor 2 reset mode), set the VW2C7 bit to 1 (when VCC reaches Vdet2 or below). (Do not set to 0.)

Figure 6.8 VW2C Register

## 6.1 VCC Input Voltage

### 6.1.1 Monitoring Vdet0

Vdet0 cannot be monitored.

### 6.1.2 Monitoring Vdet1

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

### 6.1.3 Monitoring Vdet2

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

## 6.2 Voltage Monitor 0 Reset

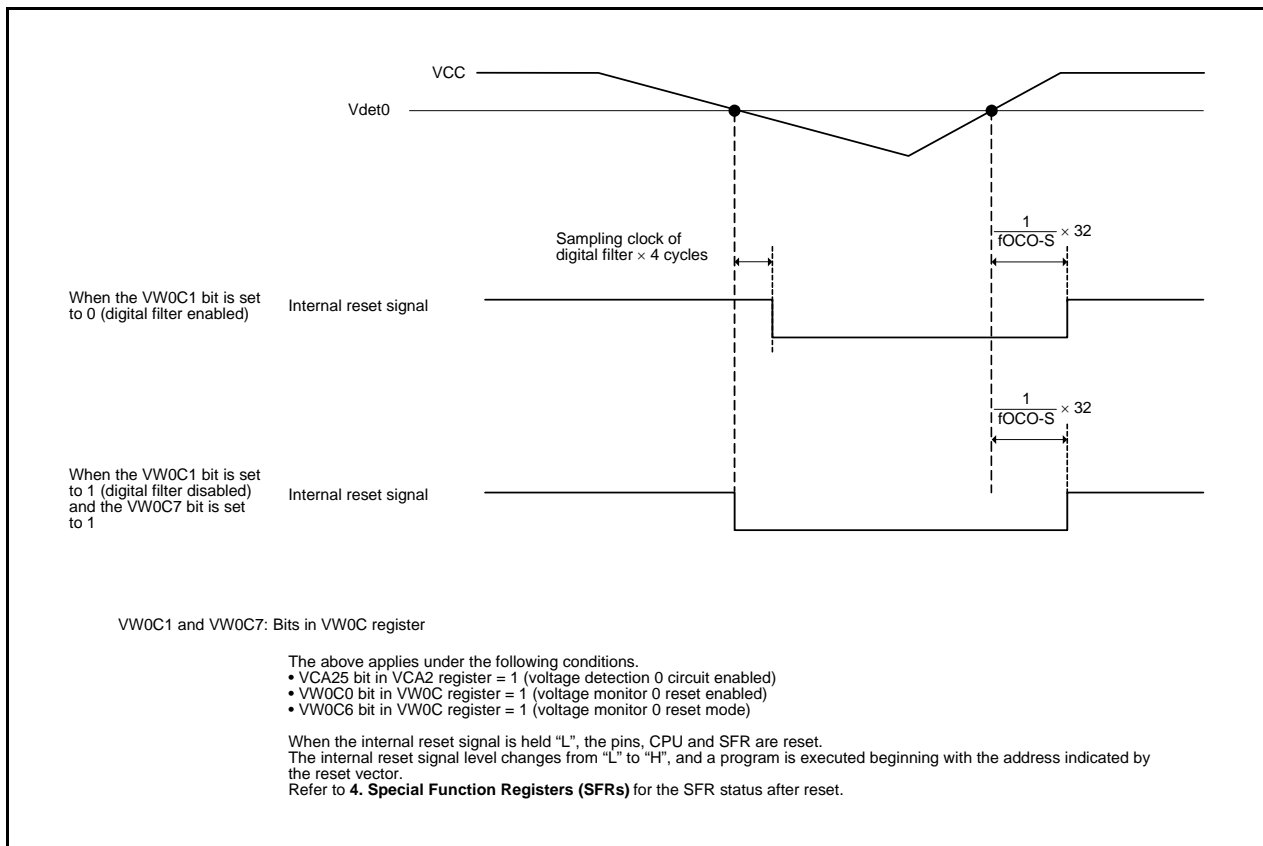
Table 6.2 lists the Procedure for Setting Bits Associated with Voltage Monitor Reset and Figure 6.9 shows an Example of Voltage Monitor 0 Reset Operation. To use the voltage monitor 0 reset to exit stop mode, set the VW0C1 bit in the VW0C register to 1 (digital filter disabled).

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

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

**NOTE:**

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



**Figure 6.9 Example of Voltage Monitor 0 Reset Operation**

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

Table 6.3 lists the Procedure for Setting Bits Associated with Voltage Monitor 1 Interrupt and Reset. Figure 6.10 shows an Example of Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset Operation. To use the voltage monitor 1 interrupt or voltage monitor 1 reset to exit stop mode, set the VW1C1 bit in the VW1C register to 1 (digital filter disabled).

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

Step	When Using Digital Filter		When Not Using Digital Filter	
	Voltage Monitor 1 Interrupt	Voltage Monitor 1 Reset	Voltage Monitor 1 Interrupt	Voltage Monitor 1 Reset
1	Set the 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		Select the timing of the interrupt and reset request by the VW1C7 bit in the VW1C register <sup>(1)</sup>	
4 <sup>(2)</sup>	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 <sup>(2)</sup>	Set the VW1C6 bit in the VW1C register to 0 (voltage monitor 1 interrupt mode)	Set the VW1C6 bit in the VW1C register to 1 (voltage monitor 1 reset mode)	Set the VW1C6 bit in the VW1C register to 0 (voltage monitor 1 interrupt mode)	Set the VW1C6 bit in the VW1C register to 1 (voltage monitor 1 reset mode)
6	Set the VW1C2 bit in the VW1C register to 0 (passing of $V_{det1}$ is not detected)			
7	Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on)		–	
8	Wait for 4 cycles of the sampling clock of the digital filter		– (No wait time required)	
9	Set the VW1C0 bit in the VW1C register to 1 (voltage monitor 1 interrupt/reset enabled)			

**NOTES:**

1. Set the VW1C7 bit to 1 (when VCC reaches  $V_{det1}$  or below) for the voltage monitor 1 reset.
2. When the VW1C0 bit is set to 0, steps 3, 4, and 5 can be executed simultaneously (with 1 instruction).

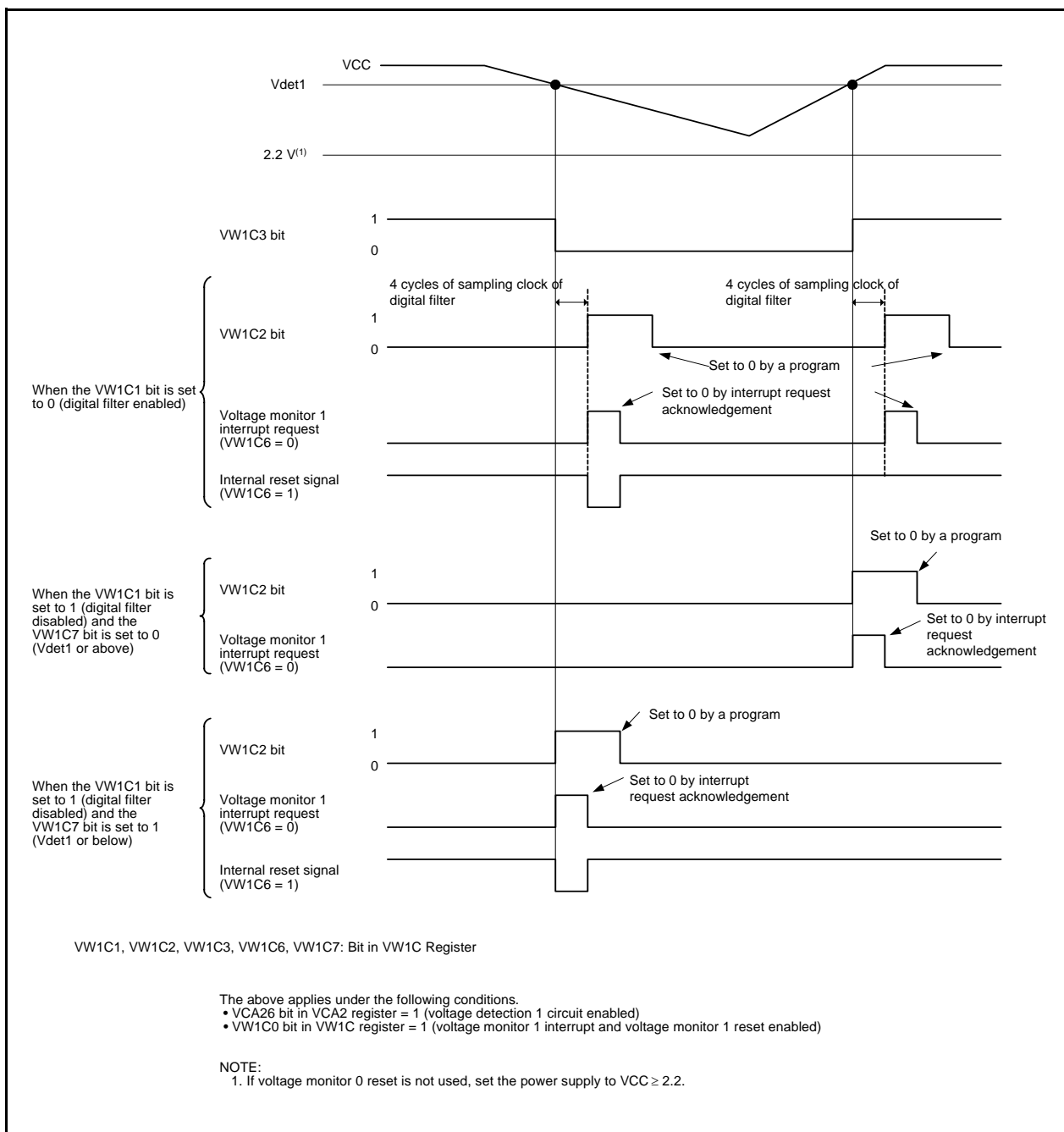


Figure 6.10 Example of Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset Operation

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

Table 6.4 lists the Procedure for Setting Bits Associated with Voltage Monitor 2 Interrupt and Reset. Figure 6.11 shows an Example of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset Operation. To use the voltage monitor 2 interrupt or voltage monitor 2 reset to exit stop mode, set the VW2C1 bit in the VW2C register to 1 (digital filter disabled).

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

Step	When Using Digital Filter		When Not Using Digital Filter	
	Voltage Monitor 2 Interrupt	Voltage Monitor 2 Reset	Voltage Monitor 2 Interrupt	Voltage Monitor 2 Reset
1	Set the 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	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 4 cycles of the sampling clock of the digital filter		– (No wait time required)	
9	Set the VW2C0 bit in the VW2C register to 1 (voltage monitor 2 interrupt/reset enabled)			

### NOTES:

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

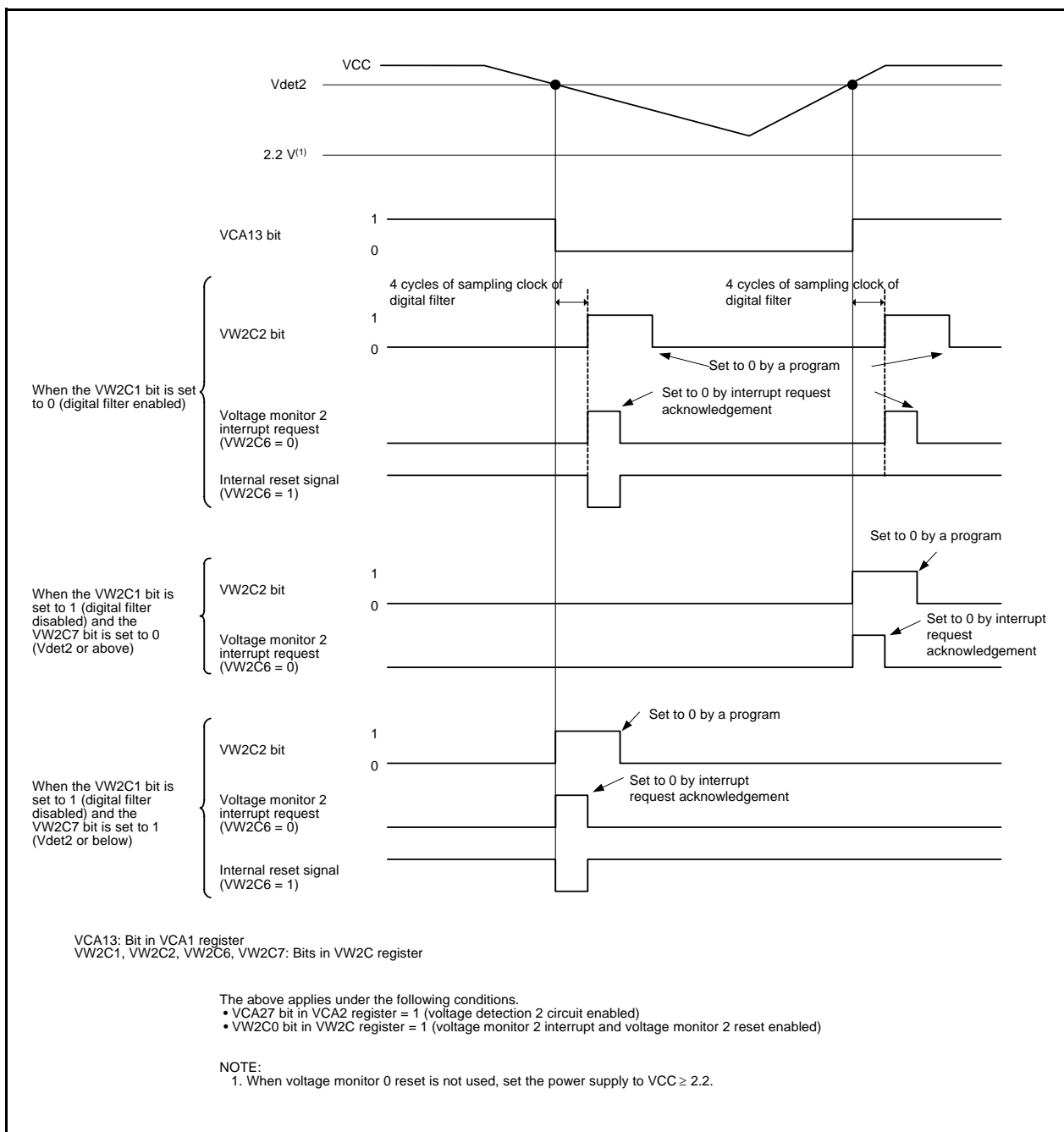


Figure 6.11 Example of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset Operation

## 7. Programmable I/O Ports

There are 25 programmable Input/Output ports (I/O ports) P0\_0 to P0\_3, P0\_5, P1, P2, P3\_3 to P3\_5, P4\_5. Also, if the XIN clock oscillation circuit is not used, P4\_6 and P4\_7 can be used as input-only ports. If the A/D converter is not used, P4\_2 can be used as an input-only port.

Table 7.1 lists an Overview of Programmable I/O Ports.

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

Ports	I/O	Type of Output	I/O Setting	Internal Pull-Up Resistor
P0_0 to P0_3, P1, P2	I/O	CMOS3 State	Set per bit	Set every 4 bits <sup>(1)</sup>
P3_4, P3_5	I/O	CMOS3 State	Set per bit	Set every 2 bits <sup>(1)</sup>
P0_5, P3_3, P4_5	I/O	CMOS3 State	Set per bit	Set every bit <sup>(1)</sup>
P4_2, P4_6 <sup>(2)</sup> , P4_7 <sup>(2)</sup>	I	(No output function)	None	None

**NOTES:**

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

### 7.1 Functions of Programmable I/O Ports

The PDi<sub>j</sub> (j = 0 to 7) bit in the PDi (i = 0 to 4) register controls I/O of the ports P0\_0 to P0\_3, P0\_5, P1, P2, P3\_3 to P3\_5, P4\_5. The Pi register consists of a port latch to hold output data and a circuit to read pin states.

Figures 7.1 to 7.6 show the Configurations of Programmable I/O Ports. Table 7.2 lists the Functions of Programmable I/O Ports. Also, Figure 7.8 shows the PDi (i = 0 to 4) Registers. Figure 7.9 shows the Pi (i = 0 to 4) Registers, Figure 7.10 shows the P2DRR Register, Figure 7.11 shows Registers PINSR1, PINSR2, PINSR3, and PMR, Figure 7.12 shows Registers PUR0, and PUR1.

**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 to the port latch is output from the pin.

i = 0 to 4, j = 0 to 7

**NOTE:**

1. Nothing is assigned to the following bits:  
PD0\_4, PD0\_6, PD0\_7, PD3\_0 to PD3\_2, PD3\_6, PD3\_7, PD4\_0 to PD4\_4, PD4\_6, PD4\_7



## 7.2 Effect on Peripheral Functions

Programmable I/O ports function as I/O ports for peripheral functions (Refer to **Table 1.7 Pin Name Information by Pin Number**).

Table 7.3 lists the Setting of PDi<sub>j</sub> Bit when Functioning as I/O Ports for Peripheral Functions (i = 0 to 4, j = 0 to 7).

Refer to the description of each function for information on how to set peripheral functions.

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

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

## 7.3 Pins Other than Programmable I/O Ports

Figure 7.7 shows the Configuration of I/O Pins.

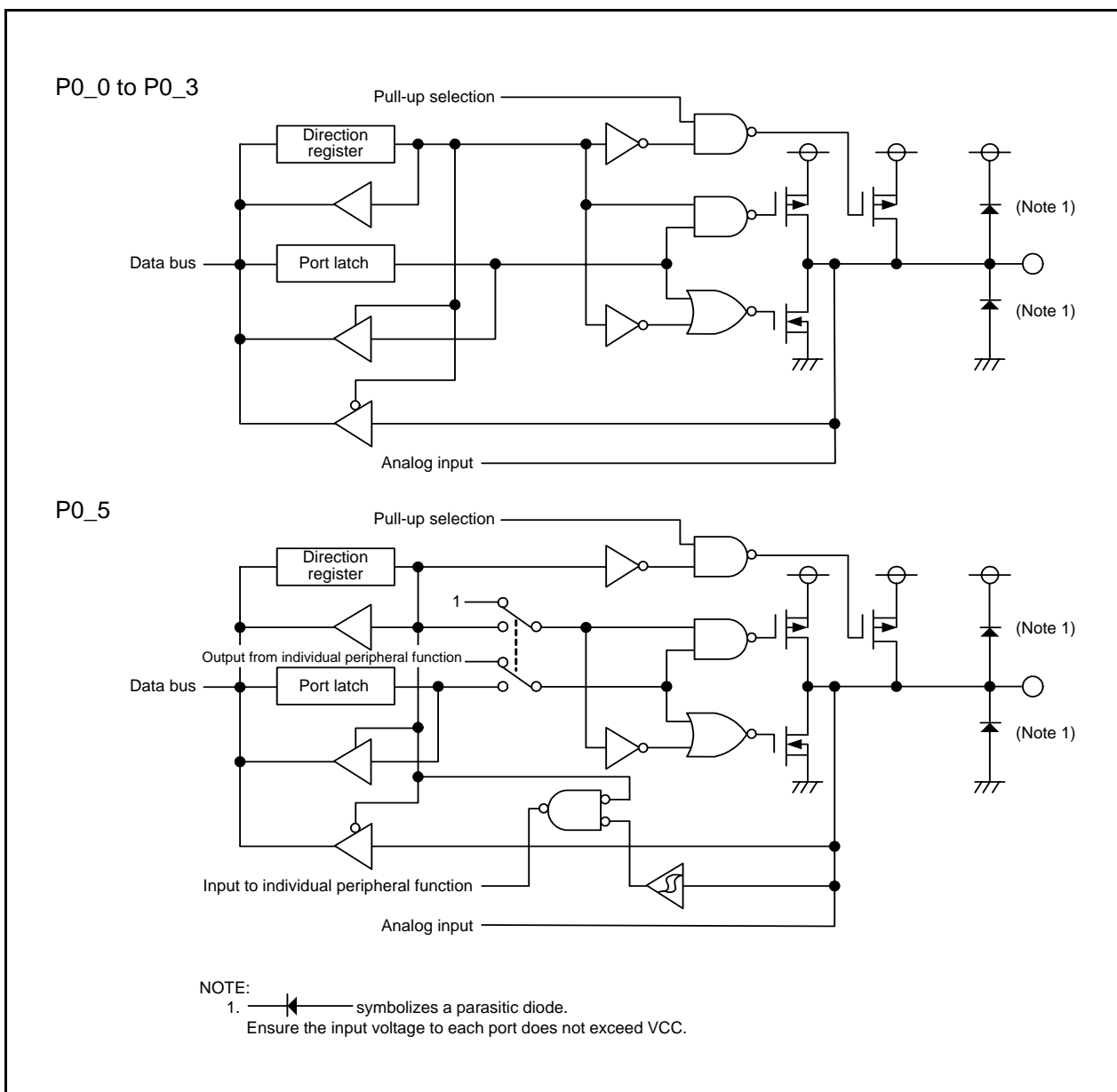


Figure 7.1 Configuration of Programmable I/O Ports (1)

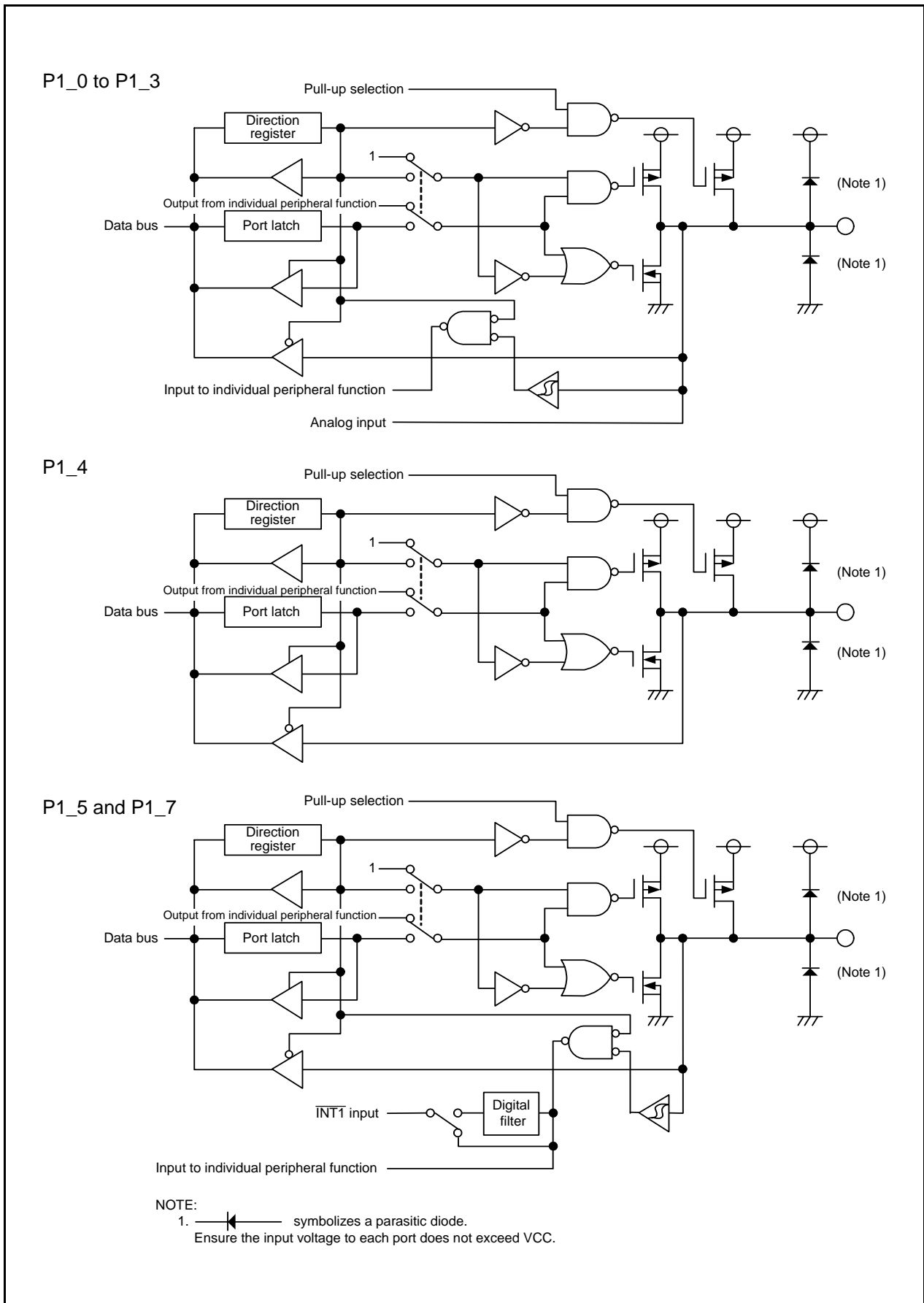


Figure 7.2 Configuration of Programmable I/O Ports (2)

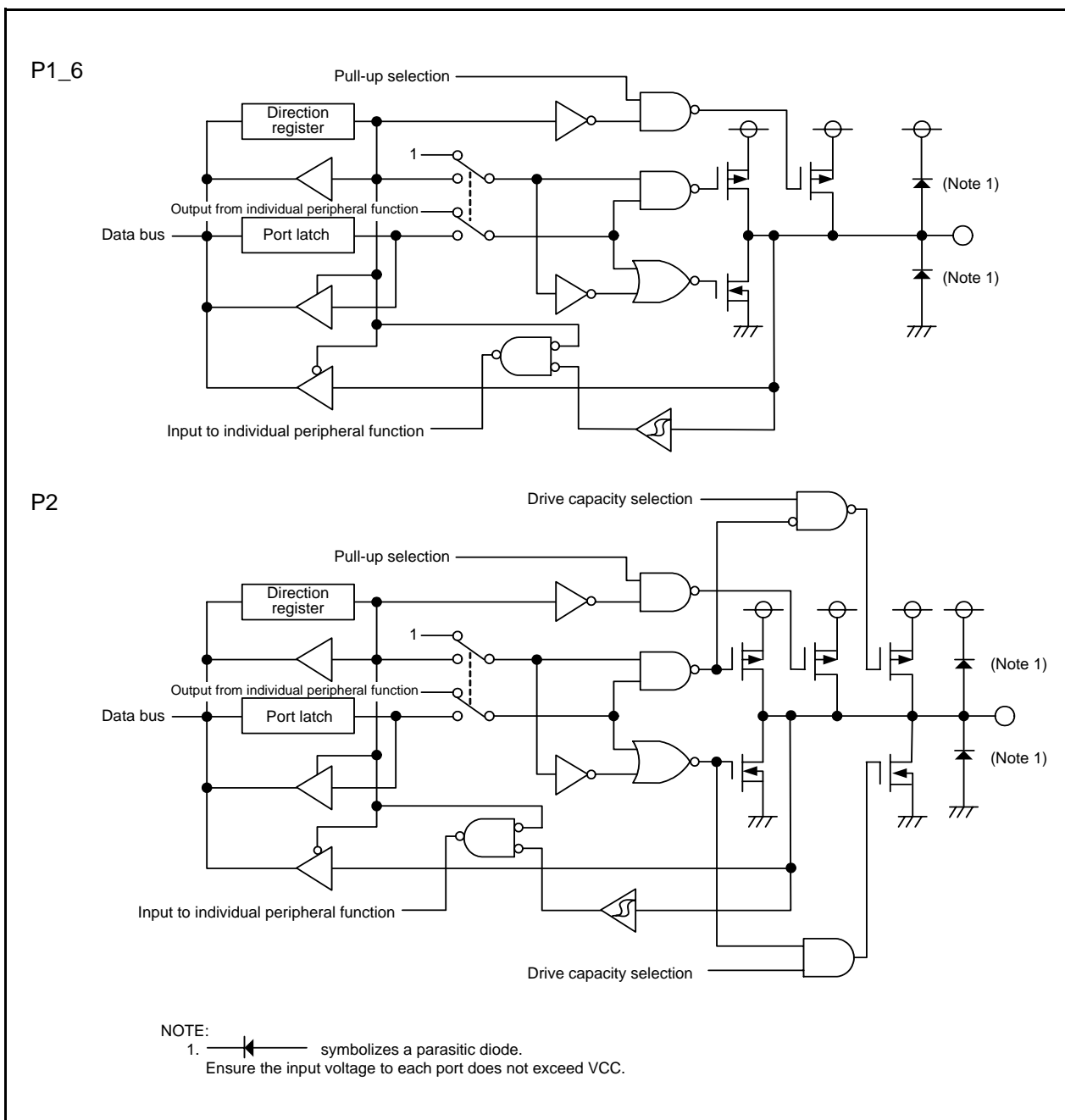


Figure 7.3 Configuration of Programmable I/O Ports (3)

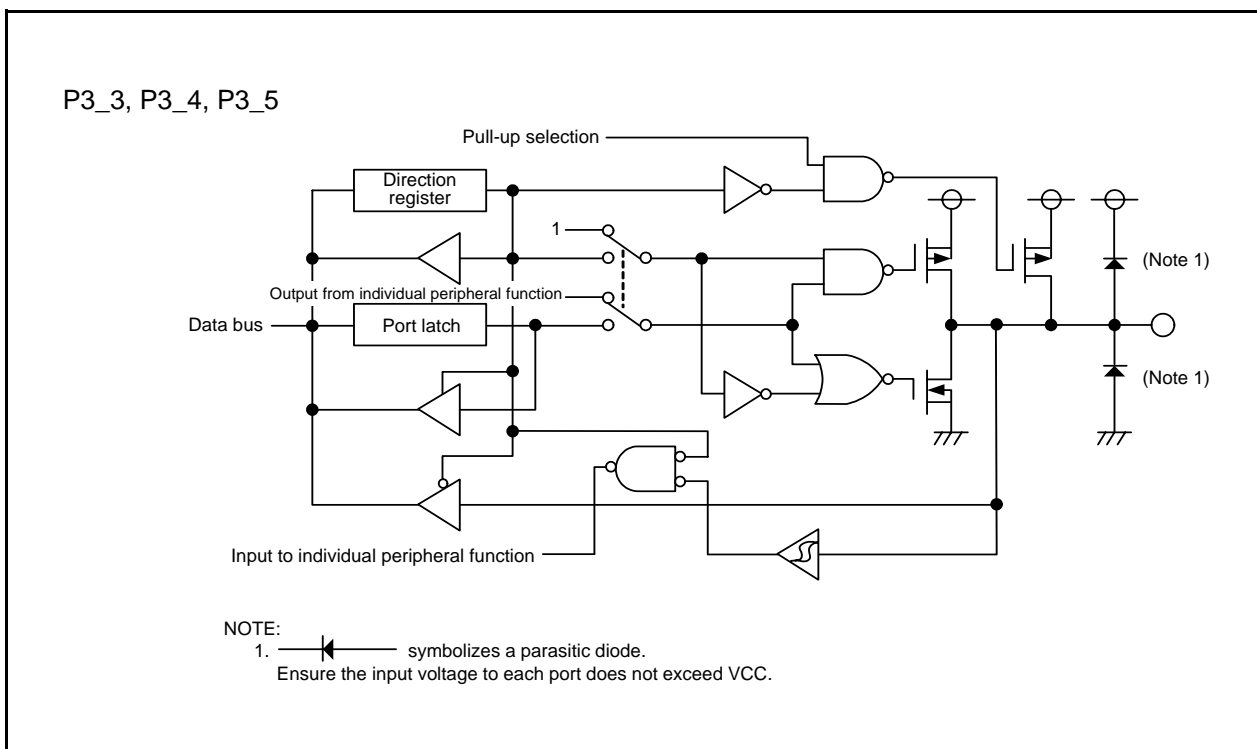


Figure 7.4 Configuration of Programmable I/O Ports (4)

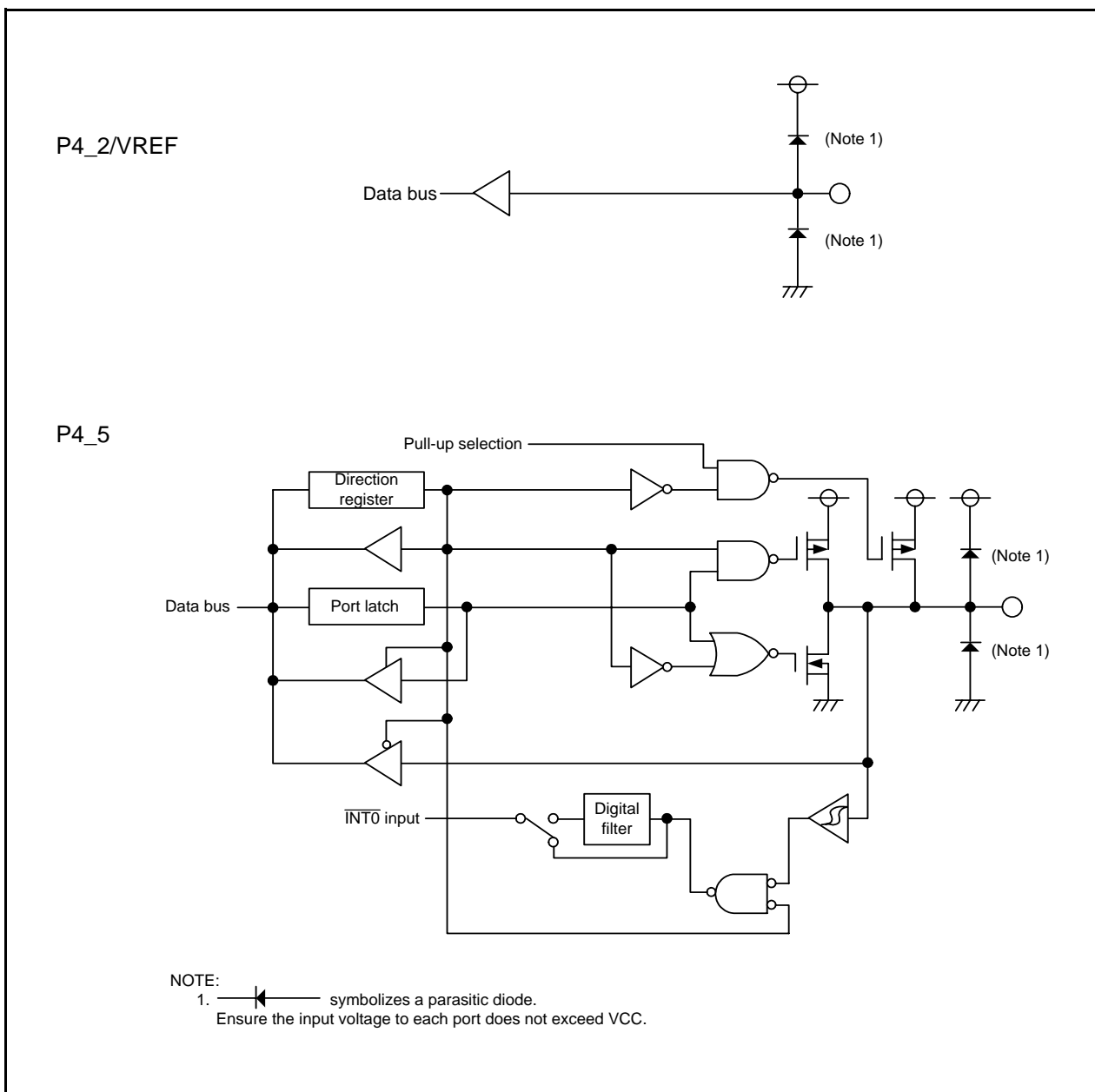


Figure 7.5 Configuration of Programmable I/O Ports (5)

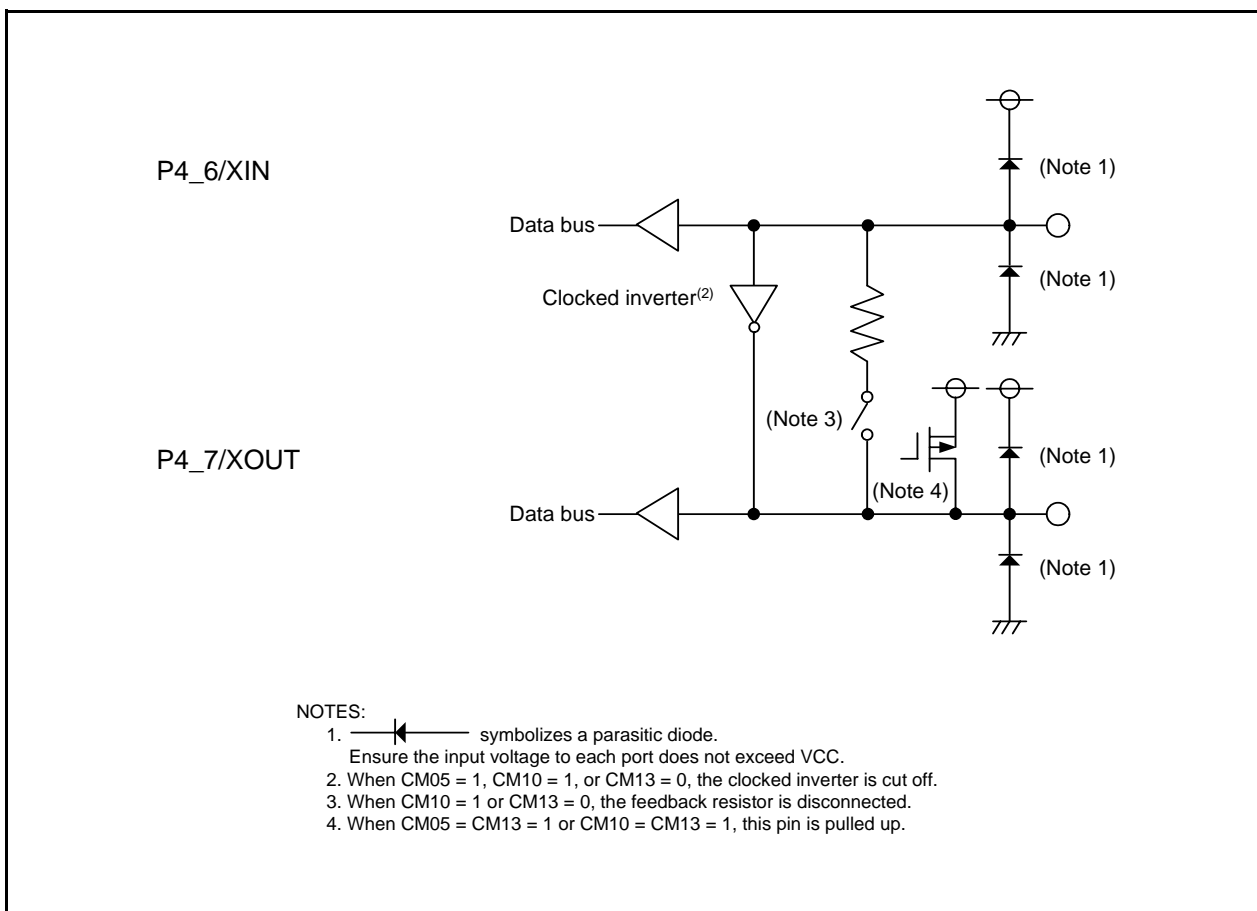
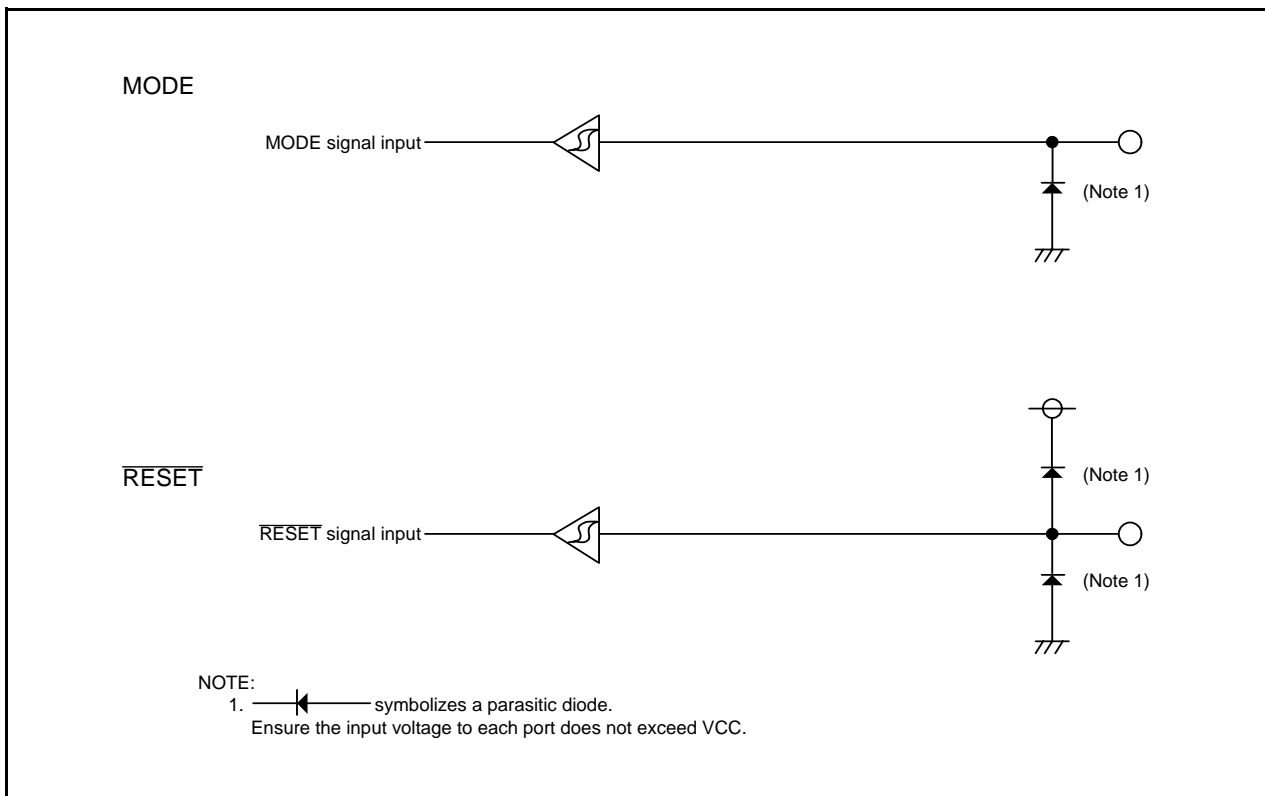


Figure 7.6 Configuration of Programmable I/O Ports (6)

**Figure 7.7 Configuration of I/O Pins**



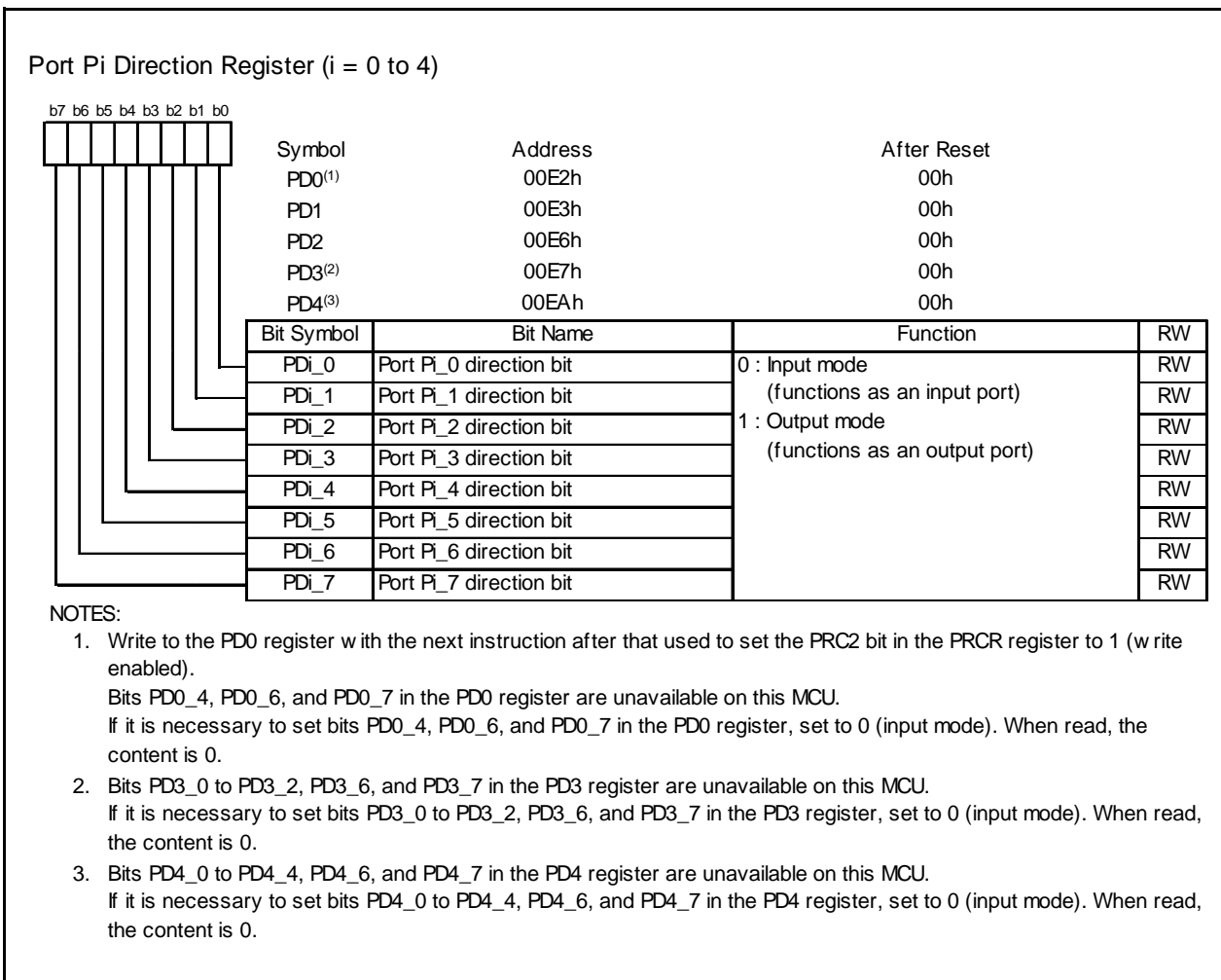
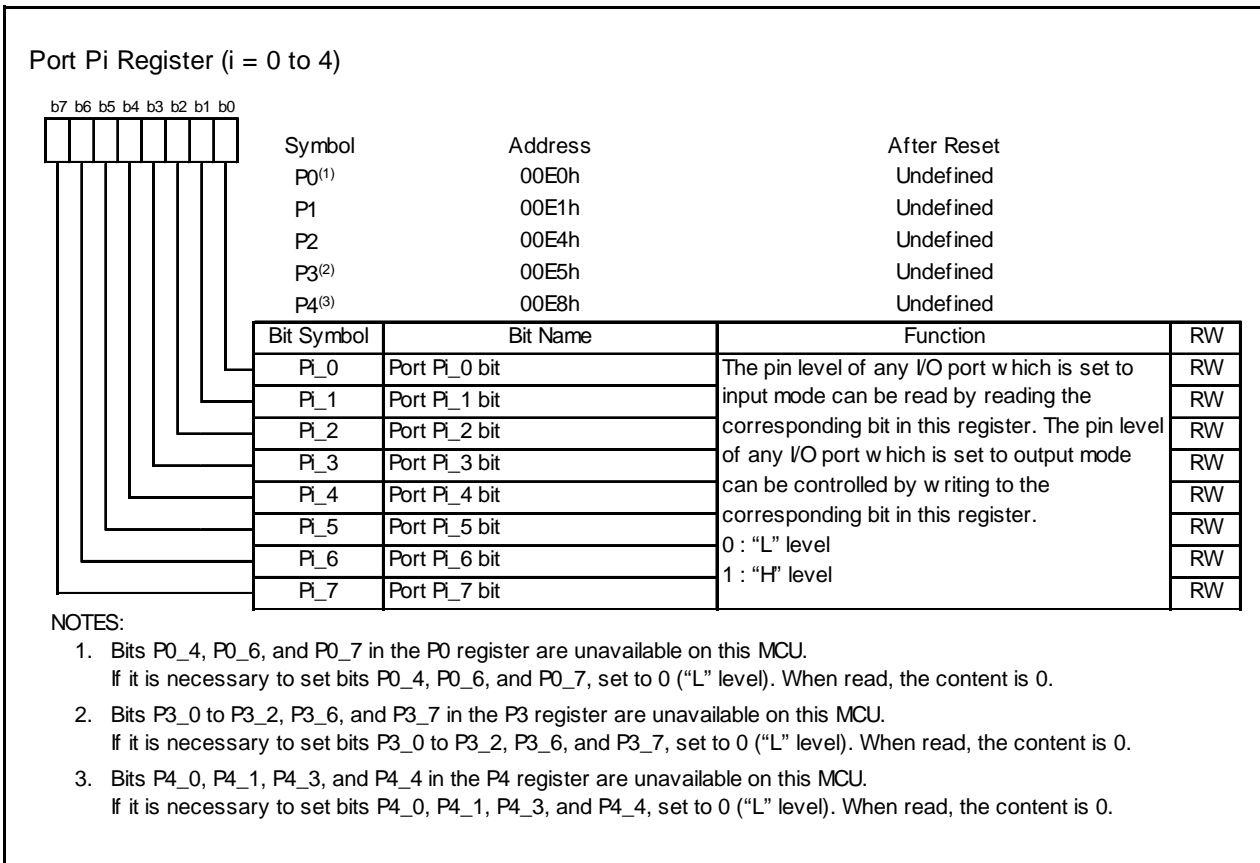
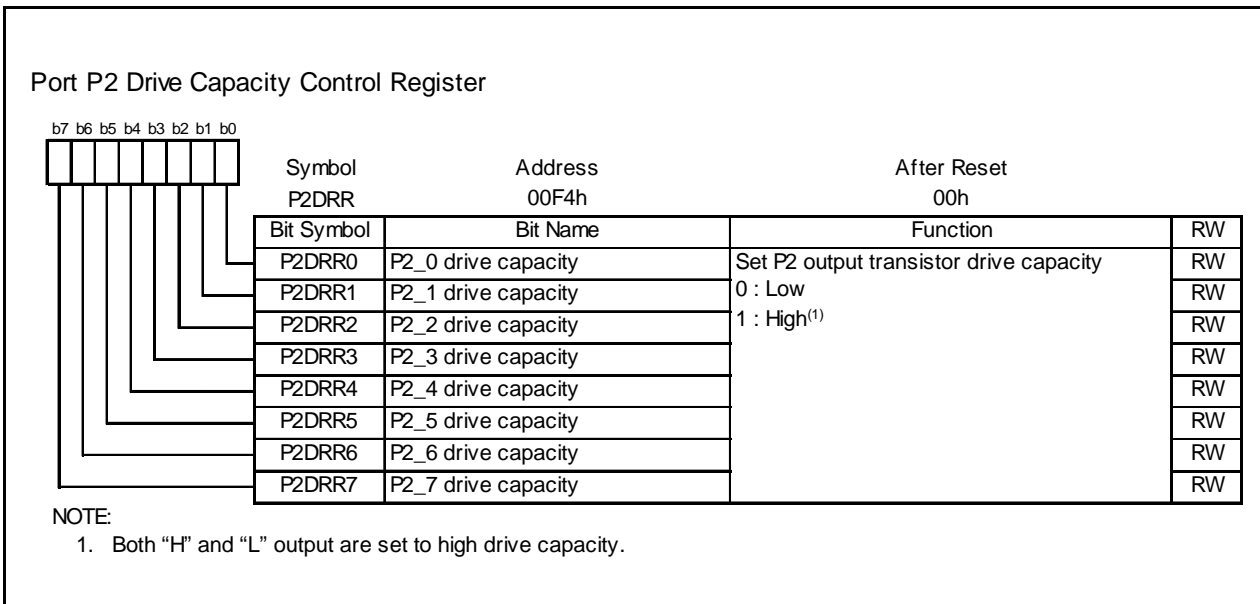


Figure 7.8 PDi (i = 0 to 4) Registers



**Figure 7.9 Pi (i = 0 to 4) Registers**



**Figure 7.10 P2DRR Register**

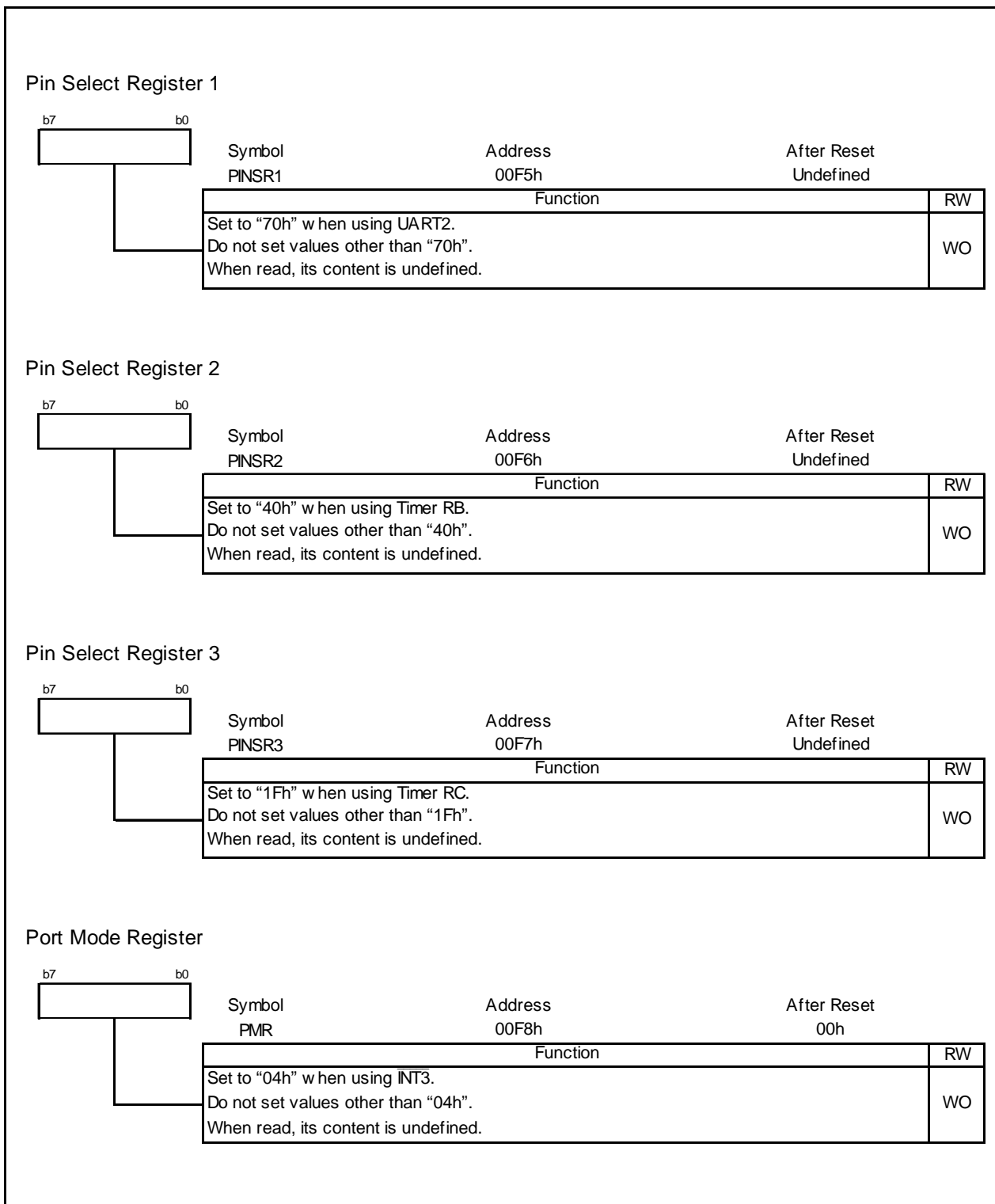
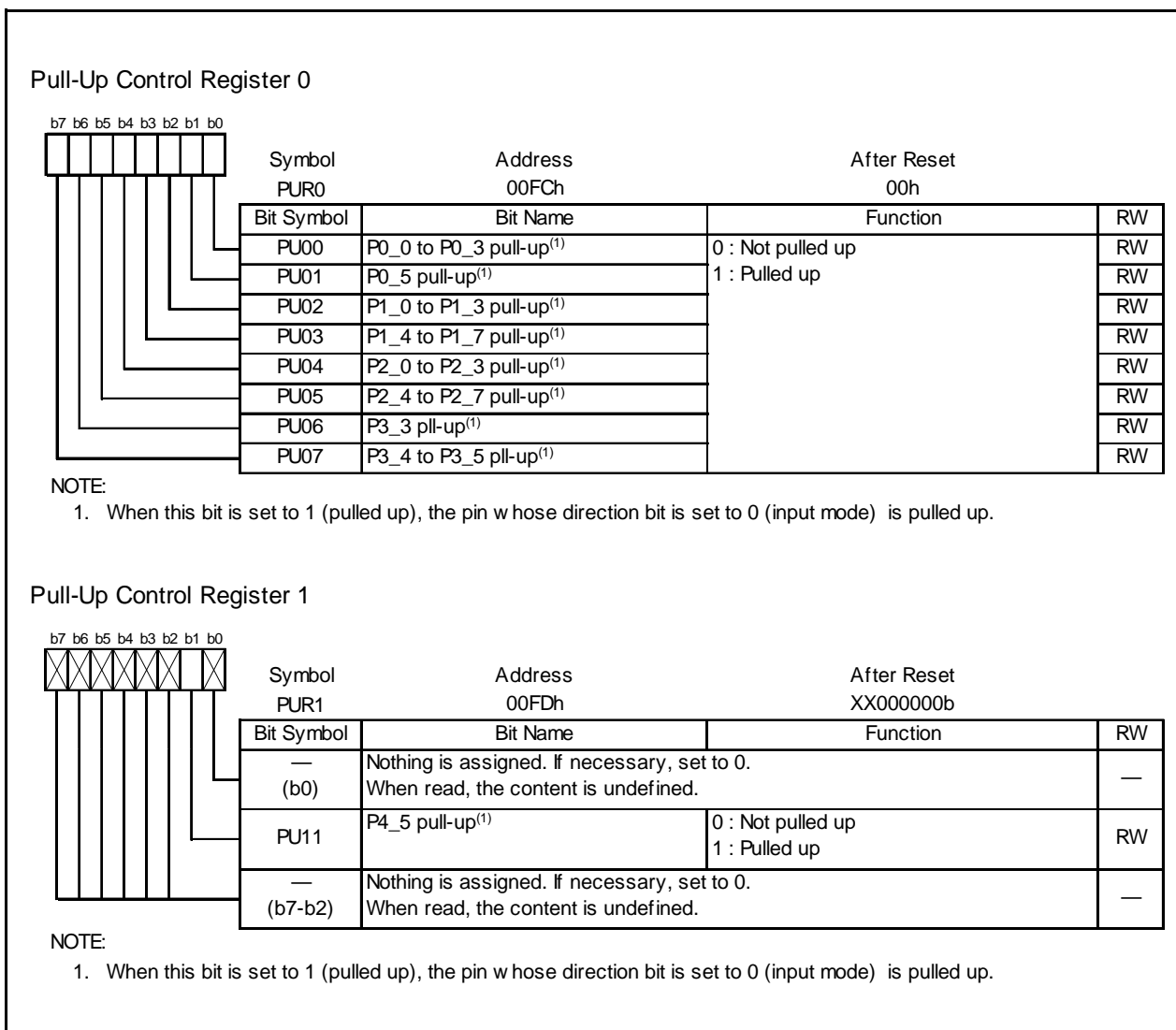


Figure 7.11 Registers PINSR1, PINSR2, PINSR3, and PMR



**Figure 7.12 Registers PUR0, and PUR1**

## 7.4 Port settings

Tables 7.4 to 7.36 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/TXD2**

Register	PD0	ADCON0				U2MR			Function	
Bit	PD0_1	CH2	CH1	CH0	ADGSEL0	SMD2	SMD1	SMD0		
Setting Value	0	X	X	X	X	X	X	X	Input port <sup>(1)</sup>	
	1	X	X	X	X	X	X	X	Output port	
	X	X	X	X	X	X	0	0	1	TXD2 output <sup>(2, 3)</sup>
							1		0	
								1	0	
	0	1	1	0	0	0	X	X	X	A/D converter input (AN6)

X: 0 or 1

NOTES:

1. Pulled up by setting the PU00 bit in the PUR0 register to 1.
2. N-channel open drain output by setting the NCH bit in the U2C0 register to 1.
3. To use the UART2, set the PINSR1 register to "70h".

**Table 7.6 Port P0\_2/AN5/RXD2**

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)
	0	X	X	X	X	RXD2 output <sup>(2)</sup>

X: 0 or 1

NOTES:

1. Pulled up by setting the PU00 bit in the PUR0 register to 1.
2. To use the UART2, set the PINSR1 register to "70h".

**Table 7.7 Port P0\_3/AN4/CLK2**

Register	PD0	ADCON0				U2MR				Function
Bit	PD0_3	CH2	CH1	CH0	ADGSEL0	SMD2	SMD1	SMD0	CKDIR	
Setting Value	0	X	X	X	X	X	X	X	X	Input port <sup>(1)</sup>
	1	X	X	X	X	Other than 001b			X	Output port
	0	X	X	X	X	X	X	X	1	CLK2 (external clock) input <sup>(2)</sup>
	X	X	X	X	X	0	0	1	0	CLK2 (internal clock) output <sup>(2)</sup>
	0	1	0	0	0	0	X	X	X	A/D converter input (AN4)

X: 0 or 1

NOTES:

1. Pulled up by setting the PU00 bit in the PUR0 register to 1.
2. To use the UART2, set the PINSR1 register to "70h".

**Table 7.8 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 7.9 Port P1\_0/AN8/KI0**

Register	PD0	KIEN	ADCON0				Function
Bit	PD0_3	KI0EN	CH2	CH1	CH0	ADGSEL0	
Setting Value	0	0	X	X	X	X	Input port <sup>(1)</sup>
	1	0	X	X	X	X	Output port
	0	1	X	X	X	X	$\overline{KI0}$ input
	0	0	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 7.10 Port P1\_1/AN9/KI1/TRCIOA/TRCTR**

Register	PD1	KIEN	Timer RC Setting	ADCON0				Function
Bit	PD1_1	KI1EN	–	CH2	CH1	CH0	ADGSEL0	
Setting value	0	0	Other than TRCIOA usage conditions	X	X	X	X	Input port <sup>(1)</sup>
	1	0	Other than TRCIOA usage conditions	X	X	X	X	Output port
	0	0	Other than TRCIOA usage conditions	1	0	1	1	A/D converter input (AN9)
	0	1	Other than TRCIOA usage conditions	X	X	X	X	$\overline{KI1}$ input <sup>(1)</sup>
	X	0	Refer to <b>Table 7.11 TRCIOA Pin Setting</b>	X	X	X	X	TRCIOA output <sup>(2)</sup>
	0	0	Refer to <b>Table 7.11 TRCIOA Pin Setting</b>	X	X	X	X	TRCIOA input <sup>(1, 2)</sup>

X: 0 or 1

NOTES:

1. Pulled up by setting the PU02 bit in the PUR0 register to 1.
2. To use the Timer RC, set the PINSR3 register to "1Fh".

**Table 7.11 TRCIOA Pin Setting**

Register	TRCOER	TRCMR	TRCIOR0			TRCCR2		Function
Bit	EA	PWM2	IOA2	IOA1	IOA0	TCEG1	TCEG0	
Setting value	0	1	0	0	1	X	X	Timer waveform output (output compare function)
			0	1	X	X	X	
	1	1	1	X	X	X	X	Timer mode (input capture function)
	1	0	X	X	X	0	1	PWM2 mode TRCTR input
						1	X	
Other than above								Other than TRCIOA usage conditions

X: 0 or 1

**Table 7.12 Port P1\_2/AN10/KI2/TRCIOB**

Register	PD1	KIEN	Timer RC Setting	ADCON0				Function
Bit	PD1_2	KI2EN	–	CH2	CH1	CH0	ADGSEL0	
Setting value	0	0	Other than TRCIOB usage conditions	X	X	X	X	Input port <sup>(1)</sup>
	1	0	Other than TRCIOB usage conditions	X	X	X	X	Output port
	0	0	Other than TRCIOB usage conditions	1	1	0	1	A/D converter input (AN10)
	0	1	Other than TRCIOB usage conditions	X	X	X	X	KI2 input <sup>(1)</sup>
	X	0	Refer to <b>Table 7.13 TRCIOB Pin Setting</b>	X	X	X	X	TRCIOB output <sup>(2)</sup>
	0	0	Refer to <b>Table 7.13 TRCIOB Pin Setting</b>	X	X	X	X	TRCIOB input <sup>(1, 2)</sup>

X: 0 or 1

NOTES:

1. Pulled up by setting the PU02 bit in the PUR0 register to 1.
2. To use the Timer RC, set the PINSR3 register to "1Fh".

**Table 7.13 TRCIOB Pin Setting**

Register	TRCOER	TRCMR		TRCIOR0			Function
Bit	EB	PWM2	PWMB	IOB2	IOB1	IOB0	
Setting value	0	0	X	X	X	X	PWM2 mode waveform output
	0	1	1	X	X	X	PWM mode waveform output
	0	1	0	0	0	1	Timer waveform output (output compare function)
				0	1	X	
	0	1	0	1	X	X	Timer mode (input capture function)
	1						
Other than above							Other than TRCIOB usage conditions

X: 0 or 1

**Table 7.14 Port P1\_3/AN11/KI3/TRBO**

Register	PD1	KIEN	Timer RB Setting	ADCON0				Function
Bit	PD1_3	KI3EN	–	CH2	CH1	CH0	ADGSEL0	
Setting value	0	0	Other than TRBO usage conditions	X	X	X	X	Input port <sup>(1)</sup>
	1	0	Other than TRBO usage conditions	X	X	X	X	Output port
	0	0	Other than TRBO usage conditions	1	1	1	1	A/D converter input (AN11)
	0	1	Other than TRBO usage conditions	X	X	X	X	KI3 input <sup>(1)</sup>
	X	0	Refer to <b>Table 7.15 TRBO Pin Setting</b>	X	X	X	X	TRBO output <sup>(2)</sup>

X: 0 or 1

NOTES:

1. Pulled up by setting the PU02 bit in the PUR0 register to 1.
2. To use the Timer RB, set the PINSR2 register to "40h".

**Table 7.15 TRBO Pin Setting**

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

NOTE:

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

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

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

X: 0 or 1

NOTES:

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

**Table 7.17 Port P1\_5/RXD0/(TRAIO)/(INT1)**

Register	PD1	TRAI0C		TRAMR			INTEN	Function
Bit	PD1_5	TIOSEL	TOPCR <sup>(2)</sup>	TMOD2	TMOD1	TMOD0	INT1EN	
Setting value	0	0	X	X	X	X	X	Input port <sup>(1)</sup>
		1	1	0	0	1	0	
		1	0	0	0	0	0	
	1	0	X	X	X	X	X	Output port
		1	0	0	0	0	X	
	0	0	X	X	X	X	X	RXD0 input <sup>(1)</sup>
		1	0	Other than 001b			0	TRAIO input <sup>(1)</sup>
		1	0	Other than 000b, 001b			0	TRAIO input <sup>(1)</sup>
		1	0	0	0	0	1	$\overline{\text{INT1}}$
		1	1	0	0	1	1	TRAIO input/ $\overline{\text{INT1}}$ <sup>(1)</sup>
X	1	0	Other than 000b, 001b			1	TRAIO pulse output	

X: 0 or 1

NOTES:

1. Pulled up by setting the PU03 bit in the PUR0 register to 1.
2. Set the TOPCR in the TRAI0C register to 0 in modes except for pulse output mode.

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

Register	PD1	U2MR				Function	
Bit	PD1_6	SMD2	SMD1	SMD0	CKDIR		
Setting Value	0	X	X	X	X	Input port <sup>(1)</sup>	
	1	Other than 001b				X	Output port
	X	0	0	1	0	CLK0 output	
	0	X	X	X	1	CLK0 input <sup>(1)</sup>	

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 <sup>(2)</sup>	TMOD2	TMOD1	TMOD0	INT1EN		
Setting value	0	1	X	X	X	X	X	Input port <sup>(1)</sup>	
		0	1	0	0	1	0		
		0	0	0	0	0	0		
	1	1	X	X	X	X	X	Output port	
		0	0	0	0	0	X		
	0	0	0	0	Other than 000b, 001b			0	TRAIO input <sup>(1)</sup>
			0	0	0	0	0	1	$\overline{\text{INT1}}$
			0	1	0	0	1	1	TRAIO input/ $\overline{\text{INT1}}$ <sup>(1)</sup>
0			0	Other than 000b, 001b			1		
X	0	0	0	0	1	X	TRAIO pulse output		

X: 0 or 1

NOTES:

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

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

Register	PD2	TRDOER1	TRDFCR				TRDIOA0			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 <sup>(2)</sup>
	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 <sup>(2)</sup>
	X	0	0	0	0	0	1	0	0	1

X: 0 or 1

NOTES:

1. Pulled up by setting the PU04 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR0 bit in the P2DRR 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 <sup>(2)</sup>	
	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 <sup>(2)</sup>
	X	0	0	0	0	1	1	X	X	X	PWM mode waveform output <sup>(2)</sup>
X	0	0	0	1	0	0	0	0	1	Timer mode waveform output (output compare function) <sup>(2)</sup>	
							0	1	X		

X: 0 or 1

NOTES:

1. Pulled up by setting the PU04 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR1 bit in the P2DRR 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 <sup>(2)</sup>	
	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 <sup>(2)</sup>
			1	1							
	X	0	0	1	X	X	X	X	X	Reset synchronous PWM mode waveform output <sup>(2)</sup>	
	X	0	0	0	1	1	X	X	X	PWM mode waveform output <sup>(2)</sup>	
X	0	0	0	1	0	0	0	1	X	Timer mode waveform output (output compare function) <sup>(2)</sup>	
						0	1				

X: 0 or 1

NOTES:

1. Pulled up by setting the PU04 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR2 bit in the P2DRR 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 <sup>(2)</sup>	
	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 <sup>(2)</sup>
			1	1							
	X	0	0	1	X	X	X	X	X	Reset synchronous PWM mode waveform output <sup>(2)</sup>	
	X	0	0	0	1	1	X	X	X	PWM mode waveform output <sup>(2)</sup>	
X	0	0	0	1	0	0	0	1	X	Timer mode waveform output (output compare function) <sup>(2)</sup>	
						0	1				

X: 0 or 1

NOTES:

1. Pulled up by setting the PU04 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR3 bit in the P2DRR 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 <sup>(2)</sup>	
	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 <sup>(2)</sup>
			1	1						
	X	0	0	1	X	X	X	X	Reset synchronous PWM mode waveform output <sup>(2)</sup>	
X	0	0	0	1	0	0	1	X	Timer mode waveform output (output compare function) <sup>(2)</sup>	
					0	1				

X: 0 or 1

NOTES:

1. Pulled up by setting the PU05 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR4 bit in the P2DRR 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 <sup>(2)</sup>	
	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 <sup>(2)</sup>
			1	1							
	X	0	0	1	X	X	X	X	X	Reset synchronous PWM mode waveform output <sup>(2)</sup>	
	X	0	0	0	1	1	X	X	X	PWM mode waveform output <sup>(2)</sup>	
X	0	0	0	1	0	0	0	1	Timer mode waveform output (output compare function) <sup>(2)</sup>		
						0	1	X			

X: 0 or 1

NOTES:

1. Pulled up by setting the PU05 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR5 bit in the P2DRR register to 1.

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

Register	PD2	TRDOER1	TRDFCR			TRDPMR	TRDIORC1			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 <sup>(2)</sup>	
	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 <sup>(2)</sup>
			1	1							
	X	0	0	1	X	X	X	X	X	Reset synchronous PWM mode waveform output <sup>(2)</sup>	
	X	0	0	0	1	1	X	X	X	PWM mode waveform output <sup>(2)</sup>	
X	0	0	0	1	0	0	0	1	Timer mode waveform output (output compare function) <sup>(2)</sup>		
						0	1	X			

X: 0 or 1

NOTES:

1. Pulled up by setting the PU05 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR6 bit in the P2DRR 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 <sup>(2)</sup>	
	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 <sup>(2)</sup>
			1	1							
	X	0	0	1	X	X	X	X	X	Reset synchronous PWM mode waveform output <sup>(2)</sup>	
	X	0	0	0	1	1	X	X	X	PWM mode waveform output <sup>(2)</sup>	
X	0	0	0	1	0	0	0	1	Timer mode waveform output (output compare function) <sup>(2)</sup>		
						0	1	X			

X: 0 or 1

NOTES:

1. Pulled up by setting the PU05 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR7 bit in the P2DRR register to 1.

**Table 7.28 Port P3\_3/ $\overline{\text{INT3}}$ / TRCCLK**

Register	PD3	TRCCR1			INTEN	Function
Bit	PD3_3	TCK2	TCK1	TCK0	INT3EN	
Setting Value	0	Other than 101b			0	Input port <sup>(1)</sup>
	1	Other than 101b			0	Output port
	0	Other than 101b			1	$\overline{\text{INT3}}$ input <sup>(1, 2)</sup>
	0	1	0	1	0	TRCCLK input <sup>(1, 3)</sup>

NOTES:

1. Pulled up by setting the PU06 bit in the PUR0 register to 1.
2. To use the INT3, set the PMR register to "04h".
3. To use the Timer RC, set the PINSR3 register to "1Fh".

**Table 7.29 Port P3\_4/TRCIOC**

Register	PD3	Timer RC Setting	Function
Bit	PD3_3	–	
Setting Value	0	Other than TRCIOC usage conditions	Input port <sup>(1)</sup>
	1	Other than TRCIOC usage conditions	Output port
	X	Refer to <b>Table 7.30 TRCIOC Pin Setting</b>	TRCIOC output <sup>(2)</sup>
	0	Refer to <b>Table 7.30 TRCIOC Pin Setting</b>	TRCIOC input <sup>(1, 2)</sup>

X: 0 or 1

NOTES:

1. Pulled up by setting the PU07 bit in the PUR0 register to 1.
2. To use the Timer RC, set the PINSR3 register to "1Fh".

**Table 7.30 TRCIOC Pin Setting**

Register	TRCOER	TRCMR		TRCIOR1			Function
Bit	EC	PWM2	PWMC	IOC2	IOC1	IOC0	
Setting value	0	1	1	X	X	X	PWM mode waveform output
	0	1	0	0	0	1	Timer waveform output (output compare function)
				0	1	X	
	1	1	0	1	X	X	Timer mode (input capture function)
				1	X	X	
Other than above							Other than TRCIOC usage conditions

X: 0 or 1

**Table 7.31 Port P3\_5/TRCIOD**

Register	PD3	Timer RC Setting	Function
Bit	PD3_5	–	
Setting Value	0	Other than TRCIOD usage conditions	Input port <sup>(1)</sup>
	1	Other than TRCIOD usage conditions	Output port
	X	Refer to <b>Table 7.32 TRCIOD Pin Setting</b>	TRCIOD output <sup>(2)</sup>
	0	Refer to <b>Table 7.32 TRCIOD Pin Setting</b>	TRCIOD input <sup>(1, 2)</sup>

X: 0 or 1

NOTES:

1. Pulled up by setting the PU07 bit in the PUR0 register to 1.
2. To use the Timer RC, set the PINSR3 register to "1Fh".

**Table 7.32 TRCIOD Pin Setting**

Register	TRCOER	TRCMR		TRCIOR1			Function
Bit	EC	PWM2	PWMD	IOC2	IOC1	IOC0	
Setting value	0	1	1	X	X	X	PWM mode waveform output
	0	1	0	0	0	1	Timer waveform output (output compare function)
					1	X	
	0	1	0	1	X	X	Timer mode (input capture function)
	1				X	X	
Other than above							Other than TRCIOD usage conditions

X: 0 or 1

**Table 7.33 Port P4\_2/VREF**

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

**Table 7.34 Port P4\_5/ $\overline{\text{INT0}}$** 

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

NOTE:

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

**Table 7.35 Port P4\_6/XIN**

Register	CM0	CM1			Circuit specifications		Function
Bit	CM5	CM13	CM11	CM10	Oscillation buffer	Feedback resistor	
Setting Value	1	0	X	0	OFF	–	Input port
	0	1	0	0	ON	ON	XIN clock oscillation (on-chip feedback resistor enabled)
	0	1	1	0	ON	OFF	XIN clock oscillation (on-chip feedback resistor disabled)
	1	1	0	0	OFF	ON	External clock input
	1	1	0	0	OFF	ON	XIN clock oscillation stop (on-chip feedback resistor enabled)
	1	1	1	0	OFF	OFF	XIN clock oscillation stop (on-chip feedback resistor disabled)
	1	1	1	1	OFF	OFF	XIN clock oscillation stop (stop mode)

X: 0 or 1

**Table 7.36 Port P4\_7/XOUT**

Register	CM0	CM1			Circuit specifications		Function
Bit	CM5	CM13	CM11	CM10	Oscillation buffer	Feedback resistor	
Setting Value	1	0	X	0	OFF	–	Input port
	0	1	0	0	ON	ON	XIN clock oscillation (on-chip feedback resistor enabled)
	0	1	1	0	ON	OFF	XIN clock oscillation (on-chip feedback resistor disabled)
	1	1	0	0	OFF	ON	External clock input
	1	1	0	0	OFF	ON	XIN clock oscillation stop (on-chip feedback resistor enabled)
	1	1	1	0	OFF	OFF	XIN clock oscillation stop (on-chip feedback resistor disabled)
	1	1	1	1	OFF	OFF	XIN clock oscillation stop (stop mode)

X: 0 or 1

## 7.5 Unassigned Pin Handling

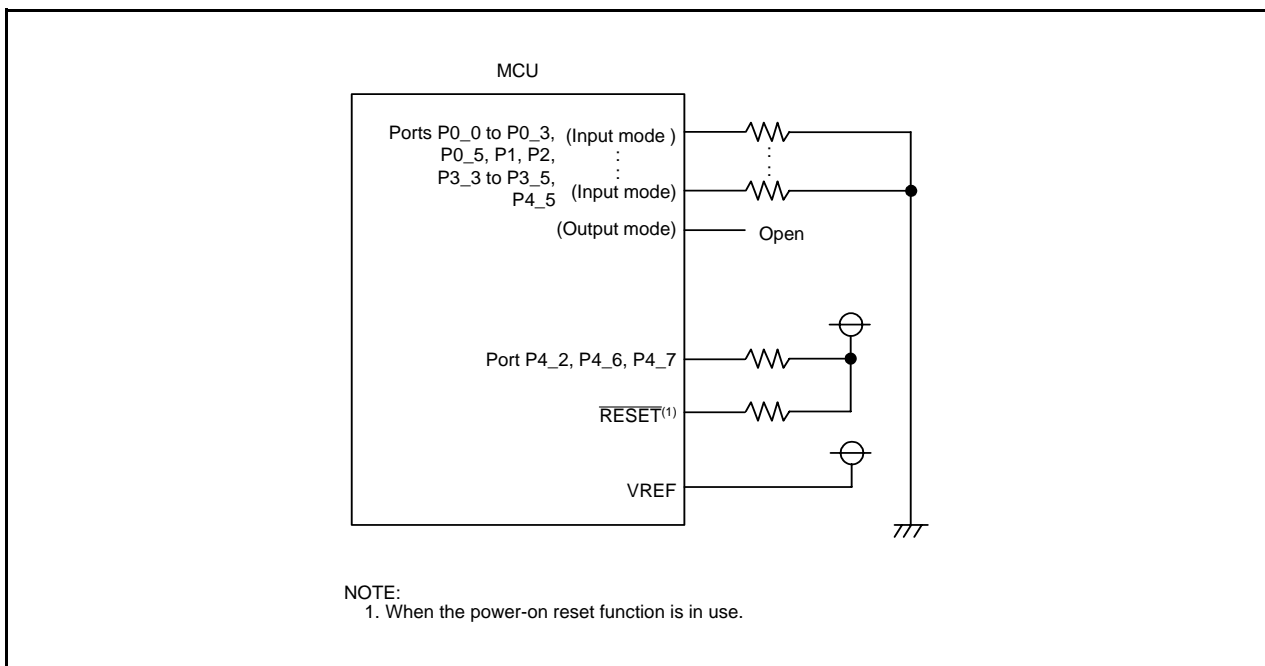
Table 7.37 lists the Unassigned Pin Handling.

**Table 7.37 Unassigned Pin Handling**

Pin Name	Connection
Ports P0_0 to P0_3, P0_5, P1, P2, P3_3 to P3_5, P4_5	<ul style="list-style-type: none"> <li>• After setting to input mode, connect each pin to VSS via a resistor (pull-down) or connect each pin to VCC via a resistor (pull-up).(2)</li> <li>• After setting to output mode, leave these pins open.(1,2)</li> </ul>
Ports P4_2, P4_6, P4_7	Connect to VCC via a pull-up resistor(2)
VREF	Connect to VCC
RESET <sup>(3)</sup>	Connect to VCC via a pull-up resistor(2)

### NOTES:

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



**Figure 7.13 Unassigned Pin Handling**

## 8. Processor Mode

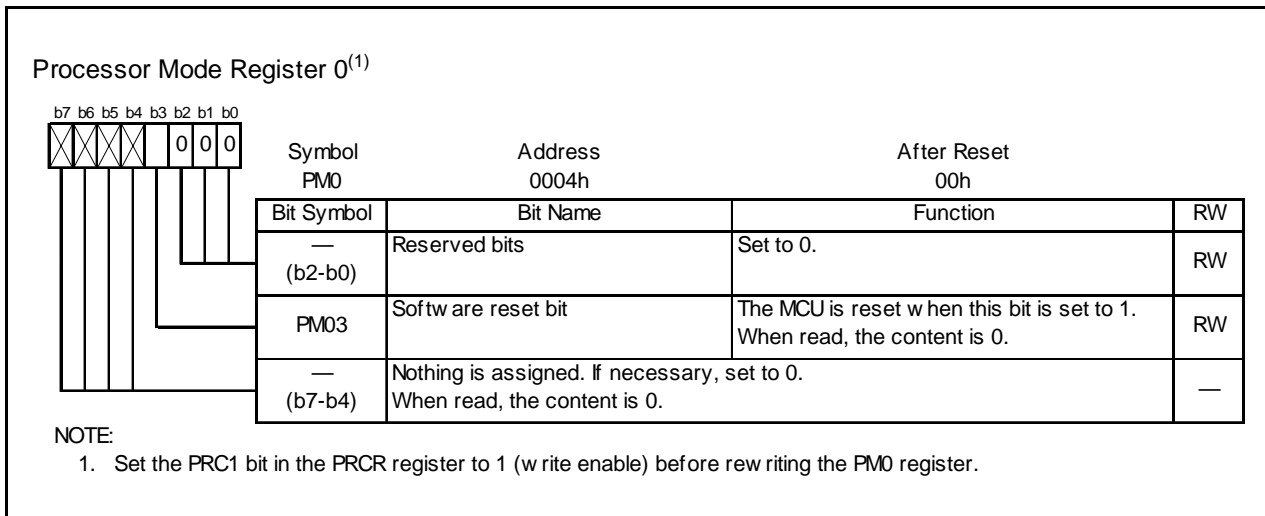
### 8.1 Processor Modes

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

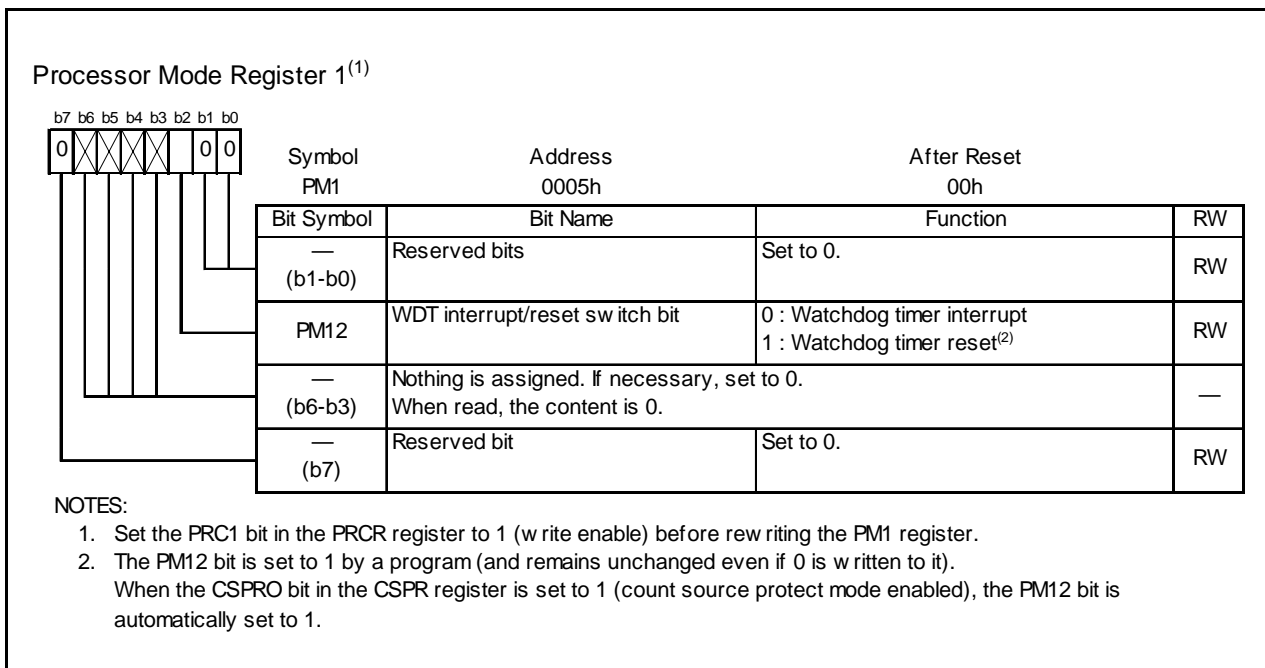
Table 8.1 lists Features of Processor Mode. Figure 8.1 shows the PM0 Register and Figure 8.2 shows the PM1 Register.

**Table 8.1 Features of Processor Mode**

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



**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/2K Group and Table 9.2 lists Bus Cycles by Access Space of the R8C/2L Group.

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

Table 9.3 lists Access Units and Bus Operations.

**Table 9.1 Bus Cycles by Access Space of the R8C/2K 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/2L Group**

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

**Table 9.3 Access Units and Bus Operations**

Area	SFR, data flash	ROM (program ROM), RAM
Even address Byte access		
Odd address Byte access		
Even address Word access		
Odd address Word access		

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

Timer RC: registers TRC, TRCGRA, TRCGRB, TRCGRC, and TRCGRD

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

Therefore, when accessing in word (16-bit) unit, 16-bit data is accessed at a time. The bus operation is the same as “Area: SFR, data flash, even address byte access” in Table 9.3 Access Units 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 the Specifications of Clock Generation Circuit. Figure 10.1 shows a Clock Generation Circuit. Figure 10.2 shows a Peripheral Function Clock. Figures 10.3 to 10.8 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
Applications	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> <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	<ul style="list-style-type: none"> <li>• Externally generated clock can be input<sup>(2)</sup></li> <li>• On-chip feedback resistor RfXIN (connected/ not connected, selectable)</li> </ul>	–	–

NOTES:

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

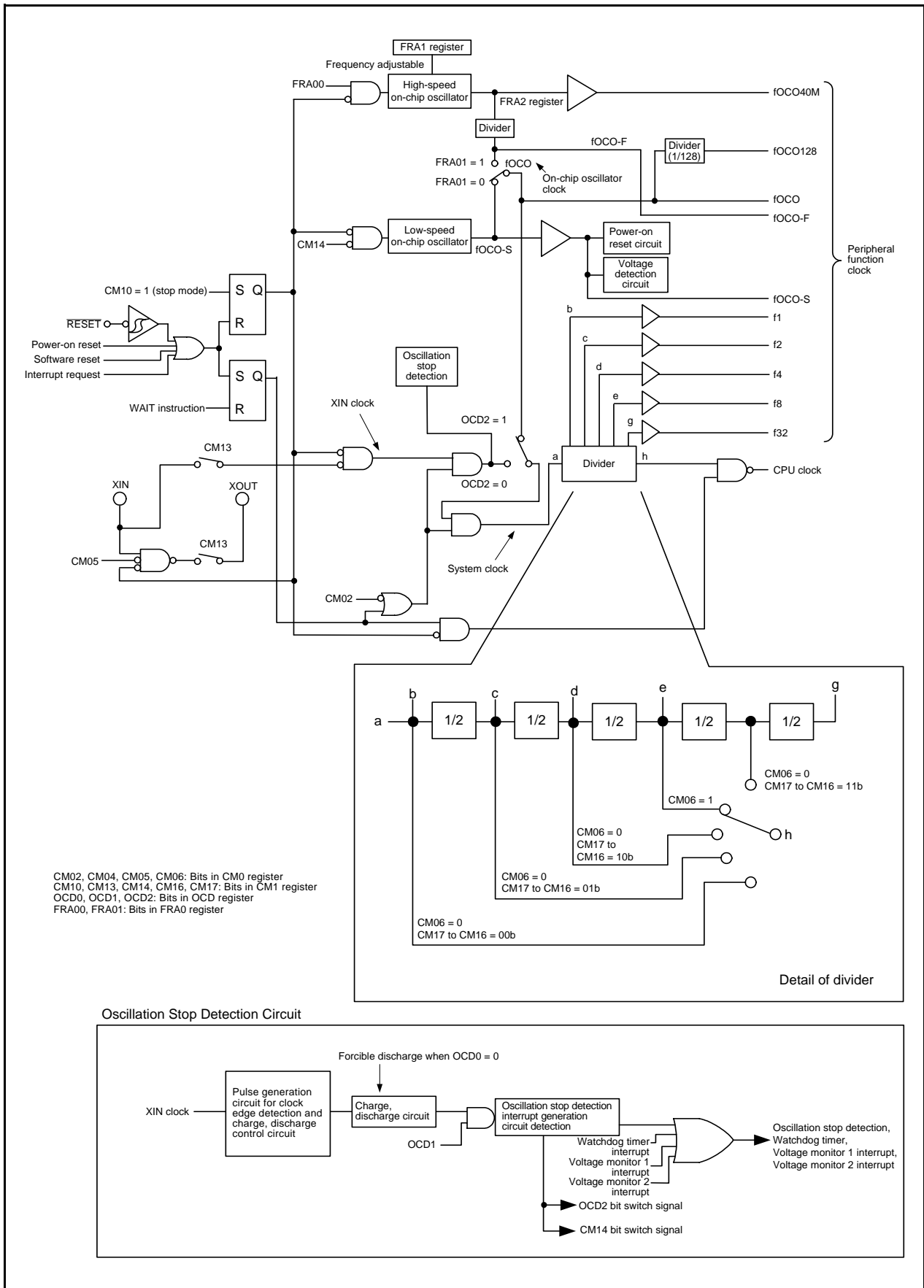


Figure 10.1 Clock Generation Circuit

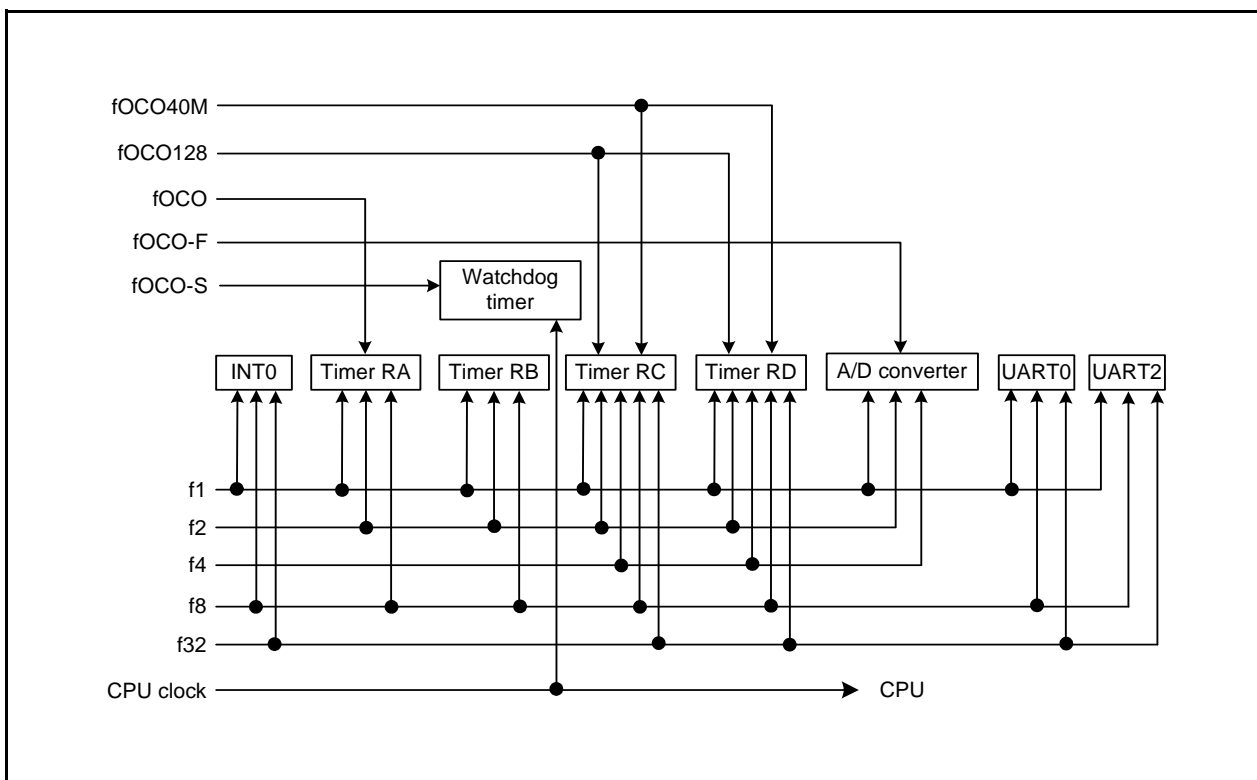


Figure 10.2 Peripheral Function Clock

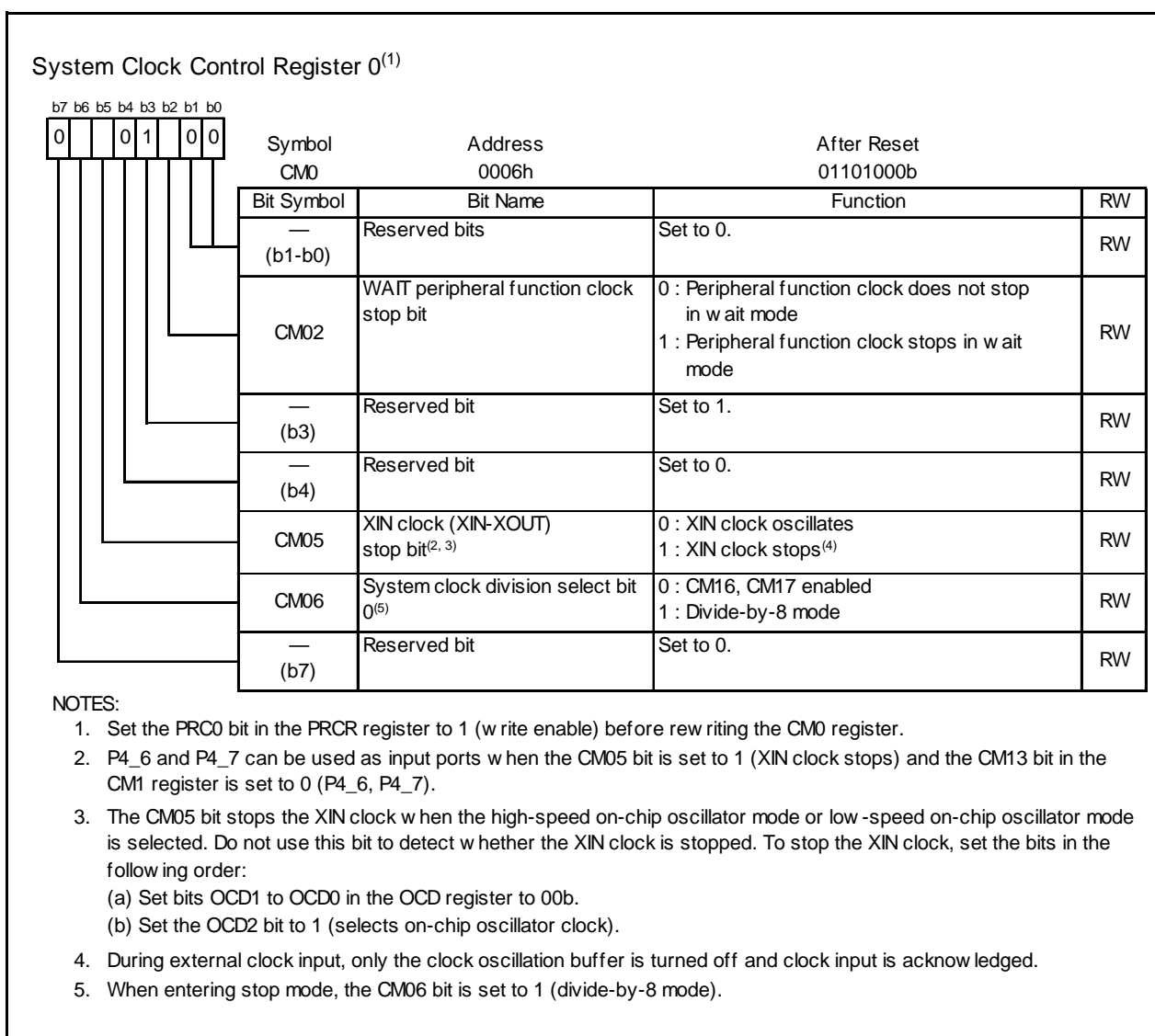


Figure 10.3 CM0 Register

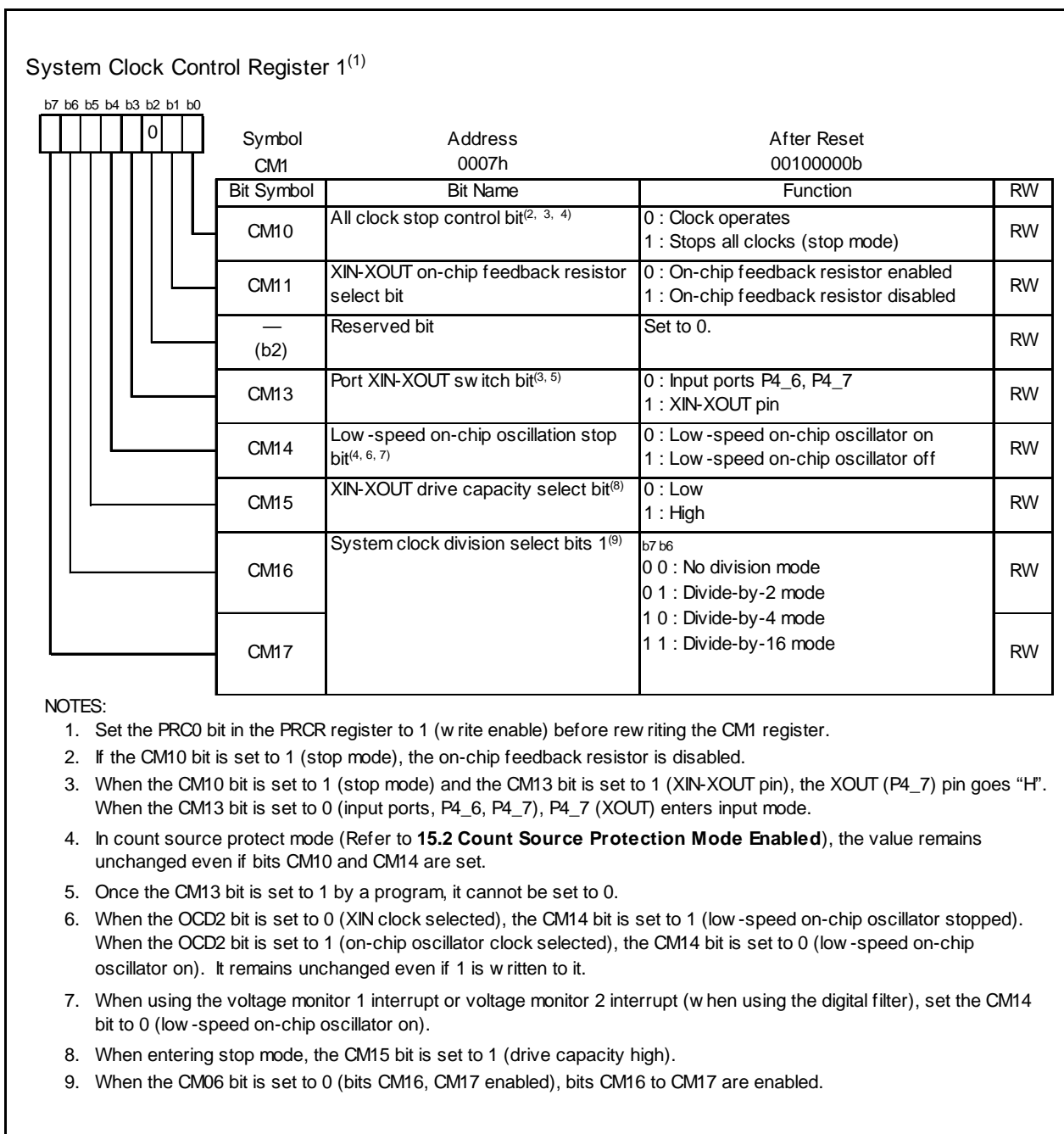


Figure 10.4 CM1 Register

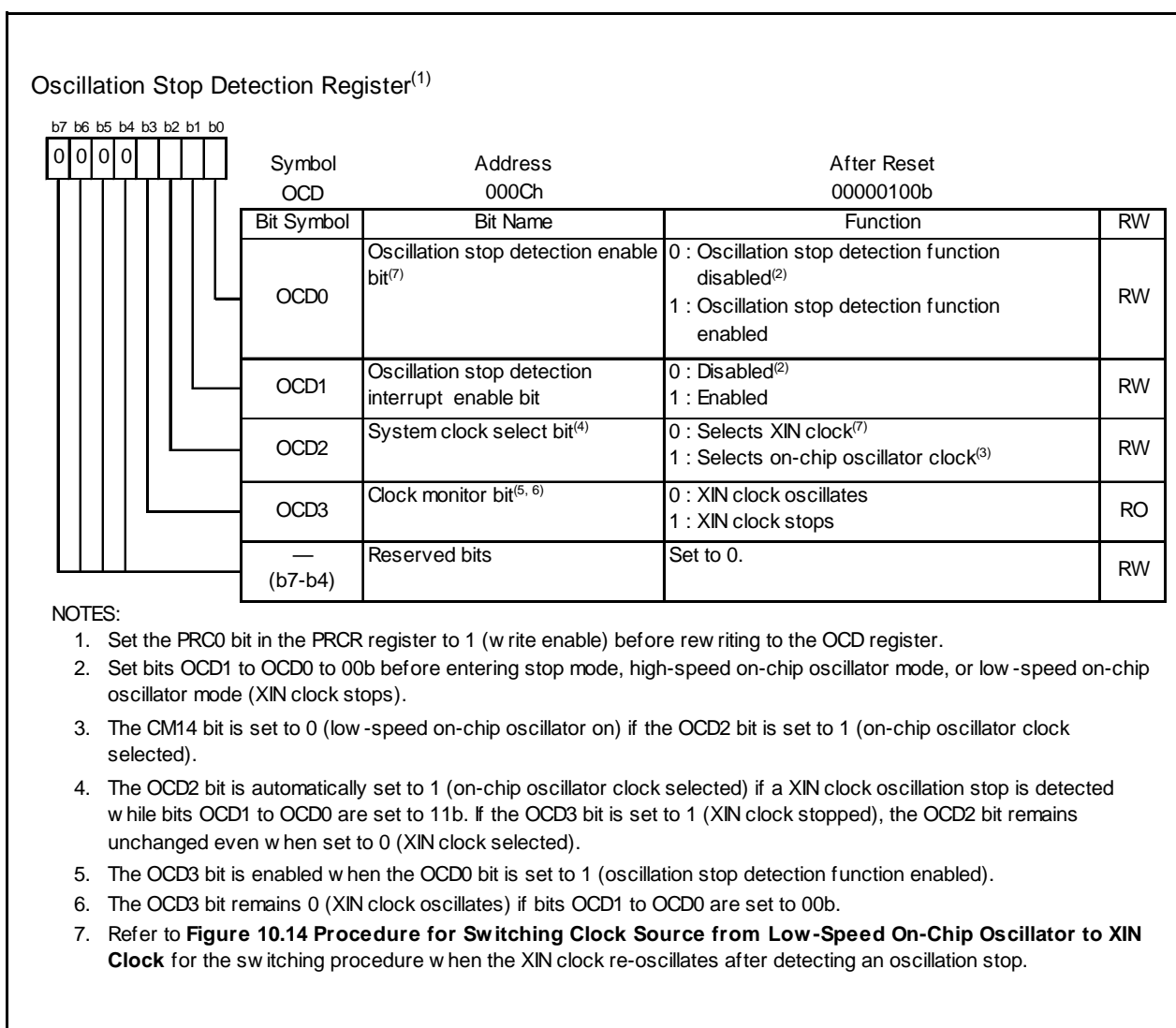


Figure 10.5 OCD Register

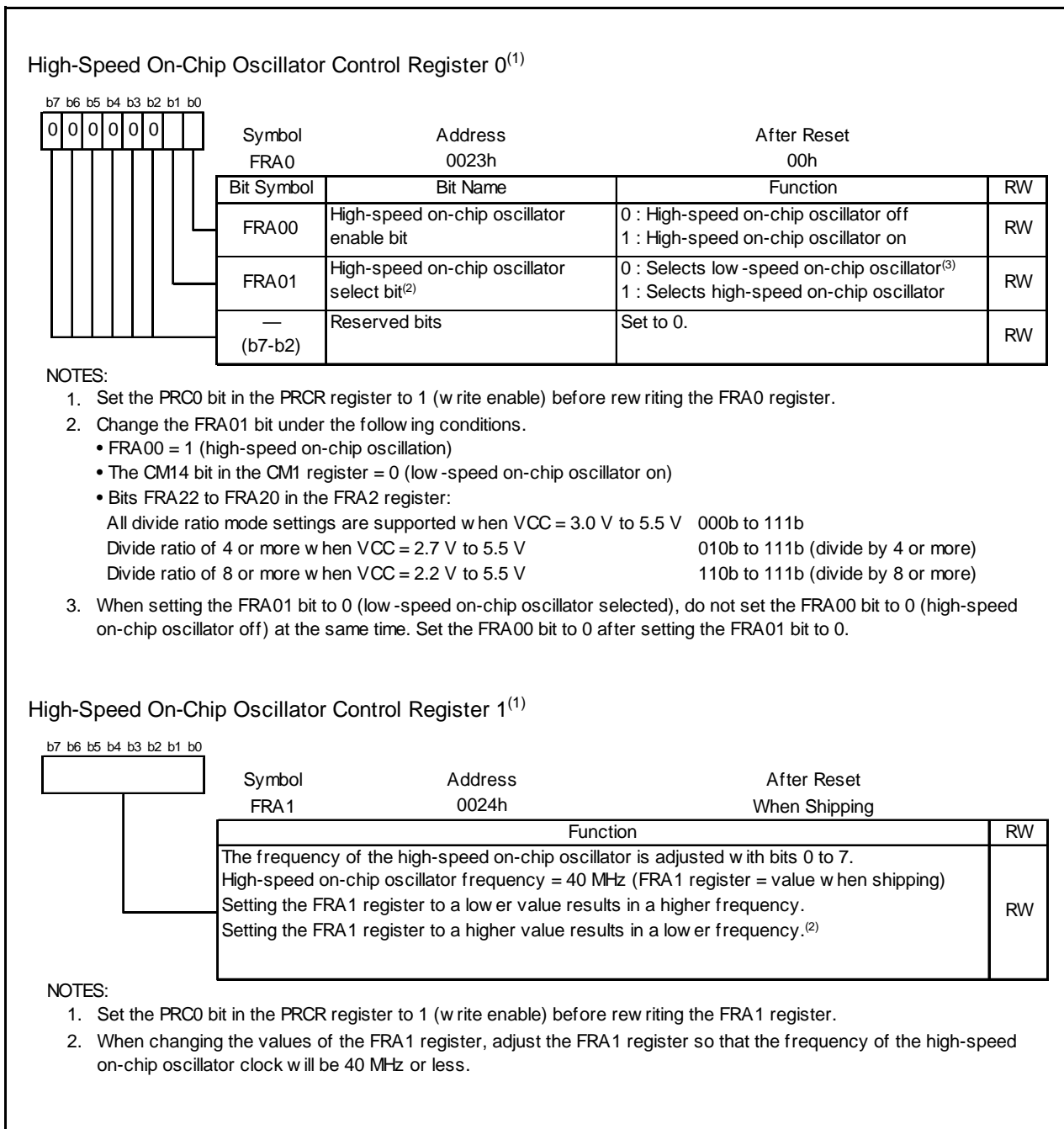


Figure 10.6 Registers FRA0 and FRA1



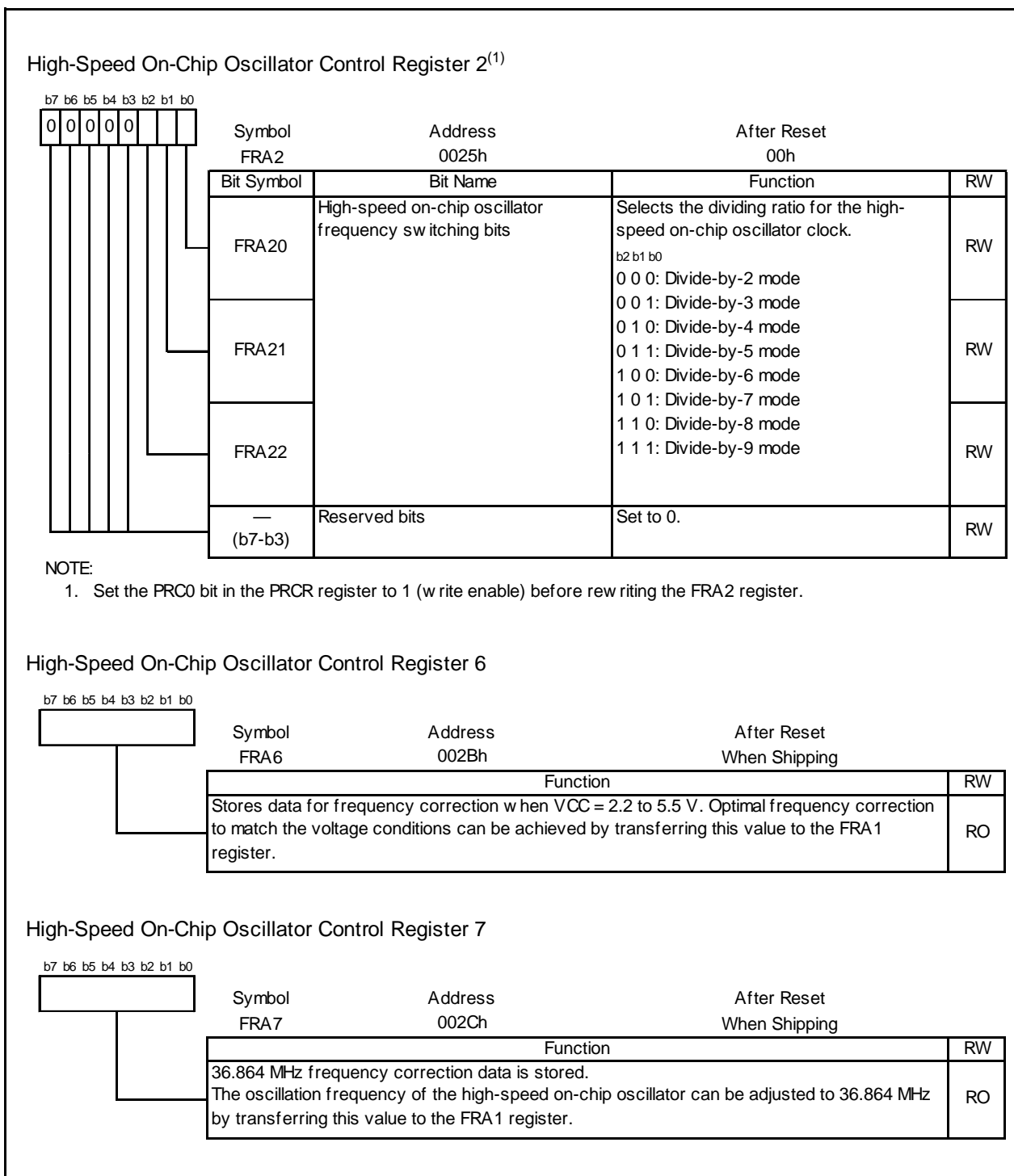


Figure 10.7 Registers FRA2, FRA6 and FRA7

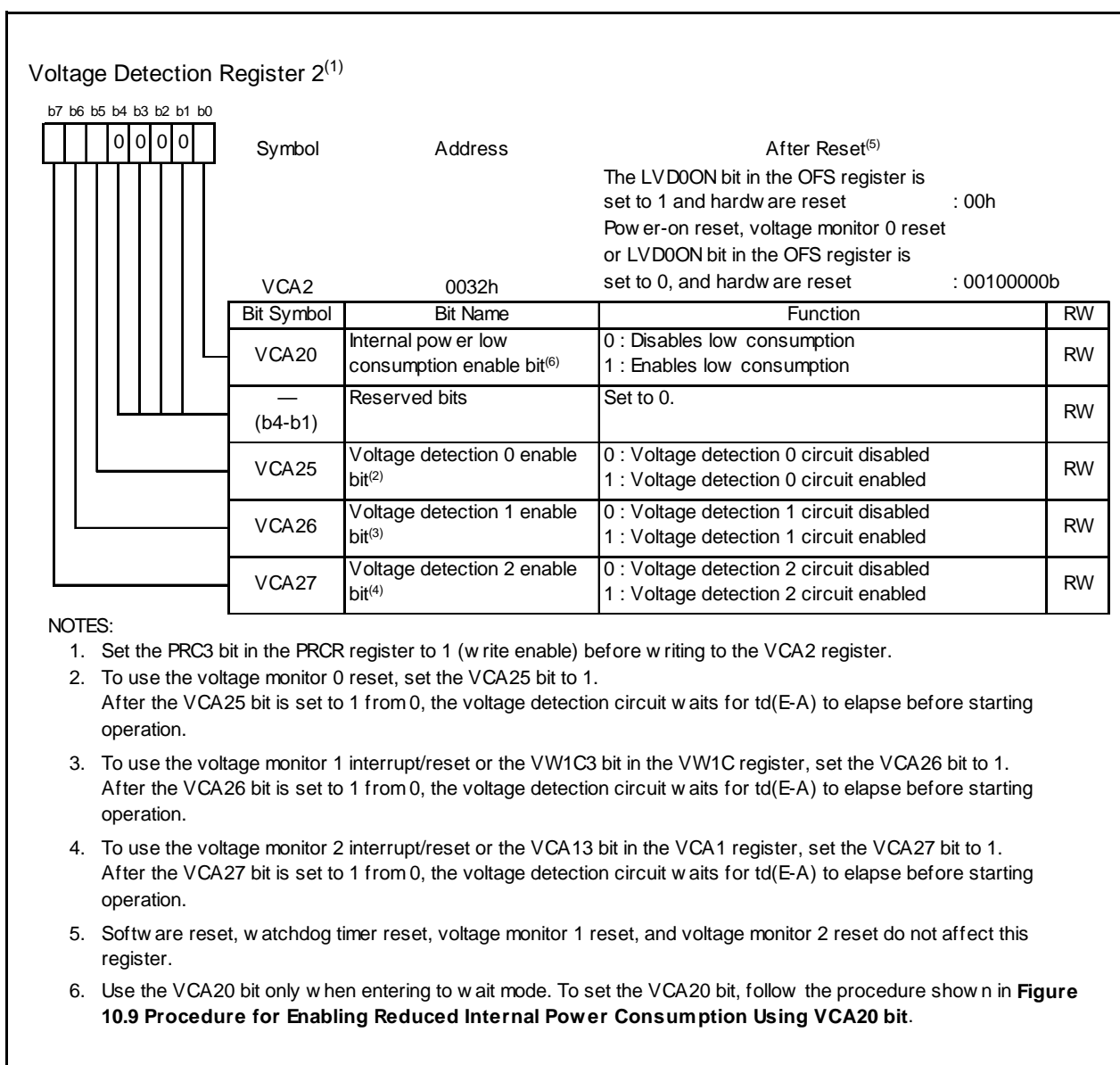
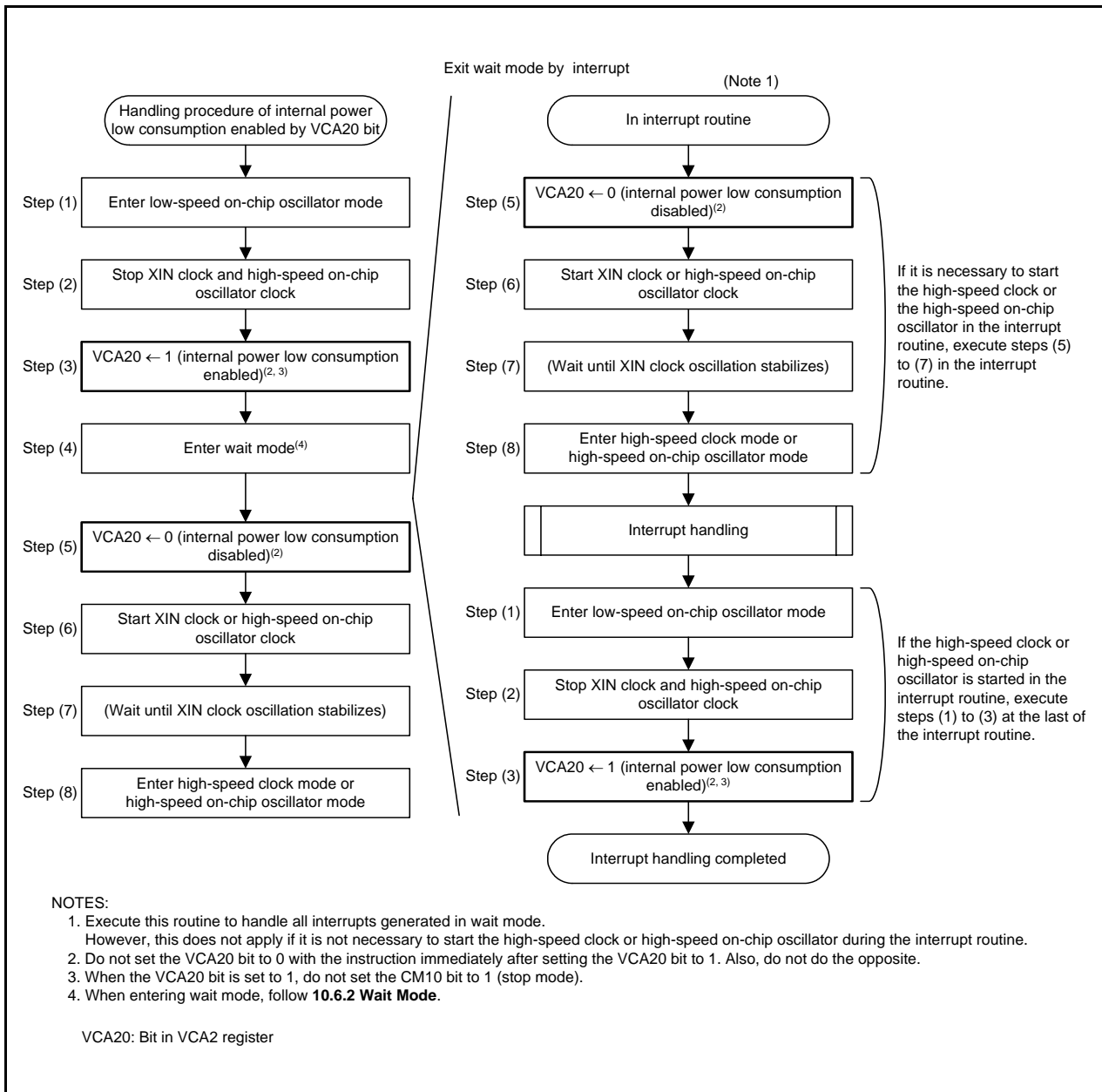


Figure 10.8 VCA2 Register



**Figure 10.9 Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit**

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

## 10.1 XIN Clock

This clock is supplied by the 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 includes an on-chip feedback resistor, which is disconnected from the oscillation circuit in stop mode in order to reduce the amount of power consumed by the chip. The XIN clock oscillation circuit may also be configured by feeding an externally generated clock to the XIN pin.

Figure 10.10 shows Examples of XIN Clock Connection Circuit.

In reset and after reset, the XIN clock stops.

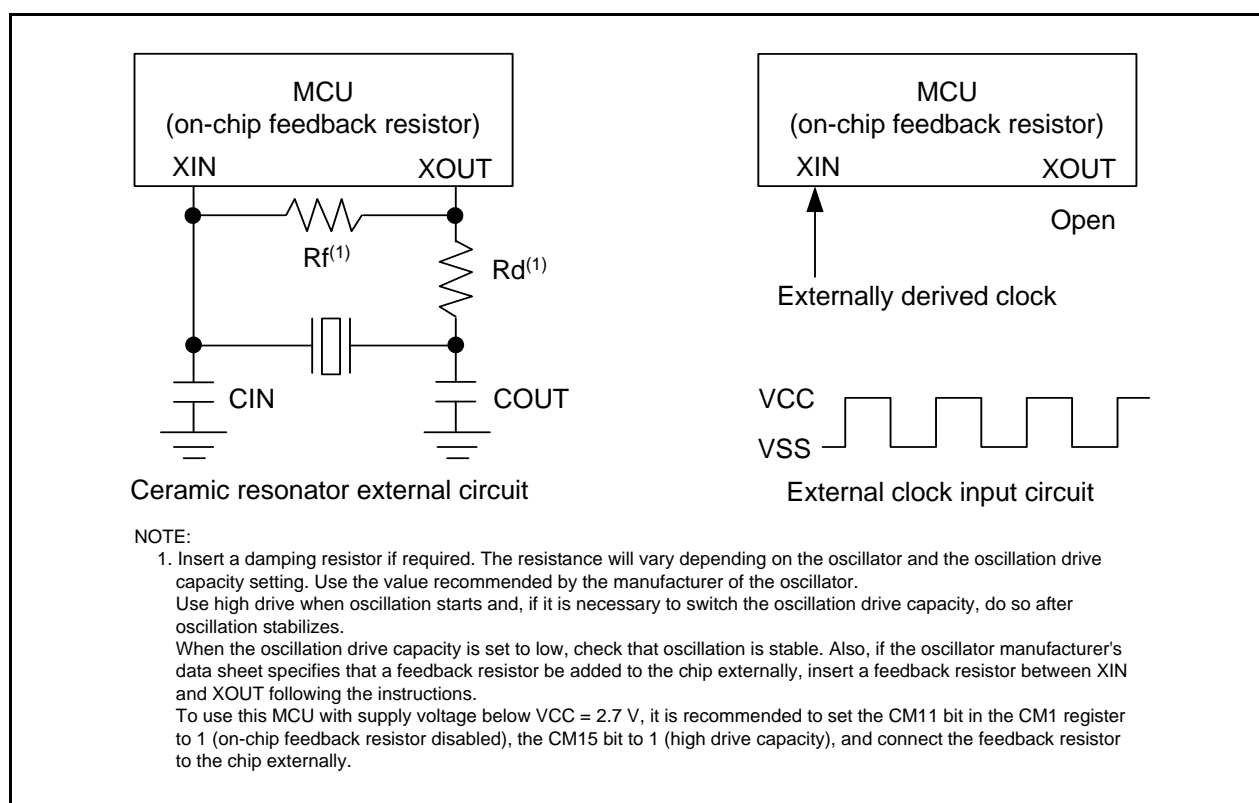
The XIN clock starts oscillating when the CM05 bit in the CM0 register is set to 0 (XIN clock oscillates) after setting the CM13 bit in the CM1 register to 1 (XIN- XOUT pin). To use the XIN clock for the CPU clock source, set the OCD2 bit in the OCD register to 0 (select XIN clock) after the XIN clock is oscillating stably.

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

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

This MCU has an on-chip feedback resistor and on-chip resistor disable/enable switching is possible by the CM11 bit in the CM1 register.

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



**Figure 10.10** Examples of XIN Clock Connection Circuit

## 10.2 On-Chip Oscillator Clocks

These clocks are supplied by the on-chip oscillators (high-speed on-chip oscillator and a low-speed on-chip oscillator). The on-chip oscillator clock is selected by the 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 divided by 8 is selected as the CPU clock.

If the XIN clock stops oscillating when bits OCD1 to OCD0 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. Application products must be designed with sufficient margin to allow for frequency changes.

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

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

To use the high-speed on-chip oscillator clock as the clock source for 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 000b to 111b
- Divide ratio of 4 or more when VCC = 2.7 V to 5.5 V 010b to 111b (divide by 4 or more)
- Divide ratio of 8 or more when VCC = 2.2 V to 5.5 V 110b to 111b (divide by 8 or more)

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

Furthermore, frequency correction data corresponding to the supply voltage ranges VCC = 2.2 V to 5.5 V is stored in FRA6 register. To use separate correction values to match this voltage ranges, transfer them from the FRA6 register to the FRA1 register.

The frequency correction data of 36.864 MHz is stored in the FRA7 register. To set the frequency of the high-speed on-chip oscillator to 36.864 MHz, transfer the correction value in the FRA7 register to the FRA1 register before use. This enables the setting errors of bit rates such as 9600 bps and 38400 bps to be 0% when the serial interface is used in UART mode (refer to **Table 17.7 Bit Rate Setting Example in UART Mode**).

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 FRA1 register so that the frequency of the high-speed on-chip oscillator clock will be 40 MHz or less.

### 10.3 CPU Clock and Peripheral Function Clock

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

#### 10.3.1 System Clock

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

#### 10.3.2 CPU Clock

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

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

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

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

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

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

The clock  $f_i$  ( $i = 1, 2, 4, 8,$  and  $32$ ) is generated by the system clock divided by  $i$ . The clock  $f_i$  is used for timers RA, RB, RC, and RD, the serial interface and the 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 an operating clock for the peripheral functions.

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

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

#### 10.3.5 fOCO40M

fOCO40M is used as the count source for timer RC and timer RD. fOCO40M is generated by the high-speed on-chip oscillator and supplied 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  $V_{CC} = 3.0$  to  $5.5$  V.

#### 10.3.6 fOCO-F

fOCO-F is used as the count source for the A/D converter. fOCO-F is generated by the high-speed on-chip oscillator and supplied 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. fOCO-S is supplied by setting the CM14 bit to 0 (low-speed on-chip oscillator on) and uses the clock generated by the low-speed on-chip oscillator. When the WAIT instruction is executed or in count source protect mode of the watchdog timer, fOCO-S does not stop.

#### 10.3.8 fOCO128

fOCO128 is generated by fOCO divided by 128.

The clock fOCO128 is used for capture signal of timer RC's TRCGRA register and timer RD (channel 0).

## 10.4 Power Control

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

### 10.4.1 Standard Operating Mode

Standard operating mode is further separated into four modes.

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

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

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

Modes		OCD Register	CM1 Register			CM0 Register		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), fOCO can be used as timer RA. When the FRA00 bit is set to 1, fOCO40M can be used as timer RC and timer RD.

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

### 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 mode) when transiting to high-speed clock mode. If the FRA00 bit is set to 1, fOCO40M can be used as timer RC and timer RD.

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

### 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 FRA01 bit 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 as timer RC and timer RD.

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

In this mode, stopping the 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 on-chip oscillator mode, setting the VCA20 bit in the VCA2 register to 1 (internal power low consumption enabled) enables lower consumption current in wait mode.



## 10.4.2 Wait Mode

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

### 10.4.2.1 Peripheral Function Clock Stop Function

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

### 10.4.2.2 Entering Wait Mode

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

When the OCD2 bit in the OCD register is set to 1 (on-chip oscillator selected as system clock), set the OCD1 bit 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 reset or a peripheral function interrupt.

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

Table 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
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 or fC32 as count source.
Timer RB interrupt	Usable in all modes	(Do not use)
Timer RC interrupt	Usable in all modes	(Do not use)
Timer RD interrupt	Usable in all modes	Usable by selecting fOCO40M as count source.
INT̄ interrupt	Usable	Usable (INT0, INT1, INT3 can be used if there is no filter.)
Voltage monitor 1 interrupt	Usable	Usable
Voltage monitor 2 interrupt	Usable	Usable
Oscillation stop detection interrupt	Usable	(Do not use)

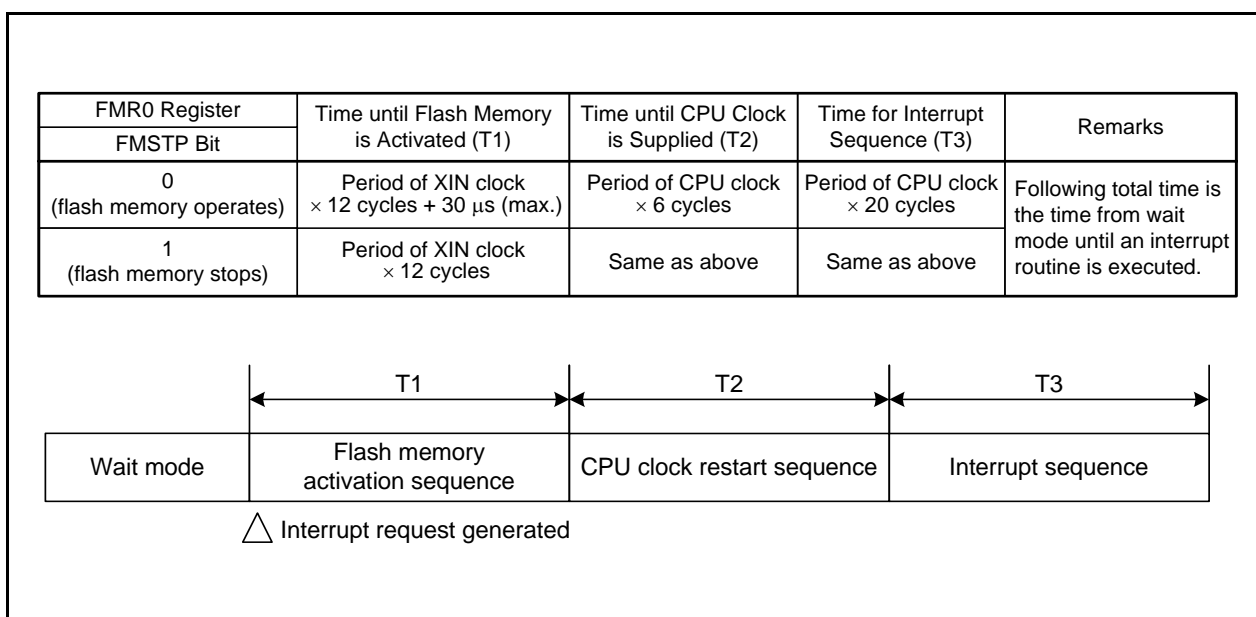
Figure 10.11 shows the Time from Wait Mode to Interrupt Routine Execution.

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

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

When exiting by a peripheral function interrupt, the time (number of cycles) between interrupt request generation and interrupt routine execution is determined by the settings of the FMSTP bit in the FMR0 register, as described in Figure 10.11.

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



**Figure 10.11 Time from Wait Mode to Interrupt Routine Execution**

### 10.4.3 Stop Mode

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

The peripheral functions clocked by external signals continue operating.

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

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

Interrupt	Usage Conditions
Key input interrupt	–
$\overline{\text{INT0}}$ , $\overline{\text{INT1}}$ , $\overline{\text{INT3}}$ 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 1 interrupt	Usable in digital filter disabled mode (VW1C1 bit in VW1C register is set to 1)
Voltage monitor 2 interrupt	Usable in digital filter disabled mode (VW2C1 bit in VW2C register is set to 1)

#### 10.4.3.1 Entering Stop Mode

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

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

#### 10.4.3.2 Pin Status in Stop Mode

The status before wait mode was entered is maintained.

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

### 10.4.3.3 Exiting Stop Mode

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

Figure 10.12 shows the Time from Stop Mode to Interrupt Routine Execution.

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

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

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

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

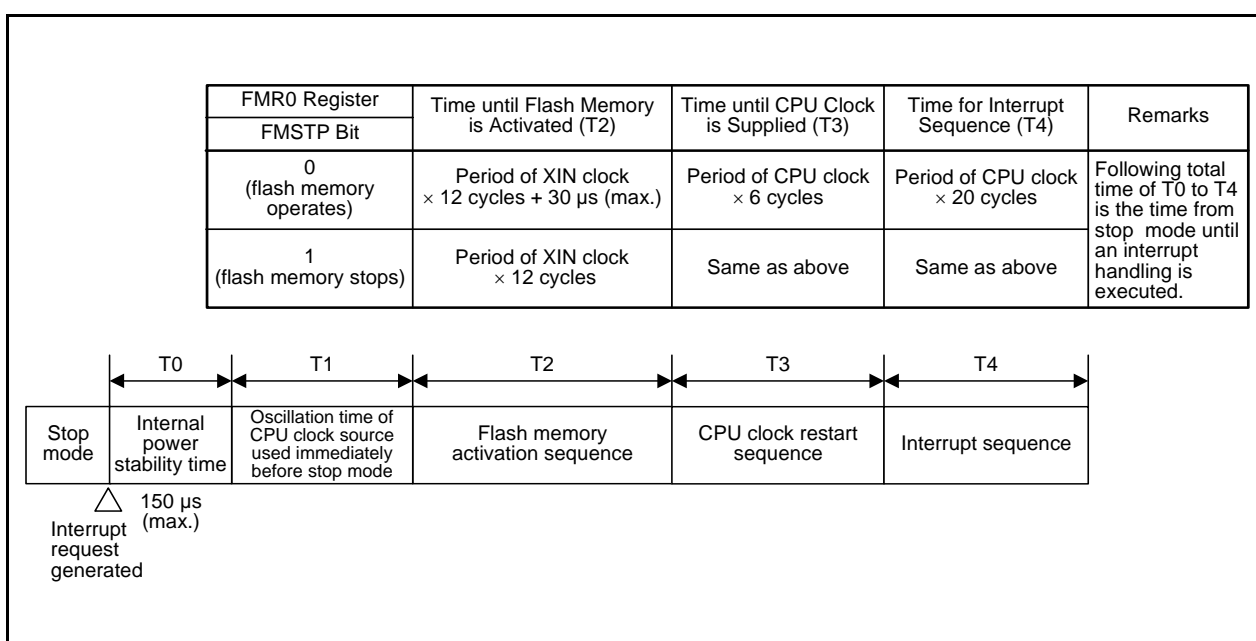


Figure 10.12 Time from Stop Mode to Interrupt Routine Execution

Figure 10.13 shows the State Transitions in Power Control Mode.

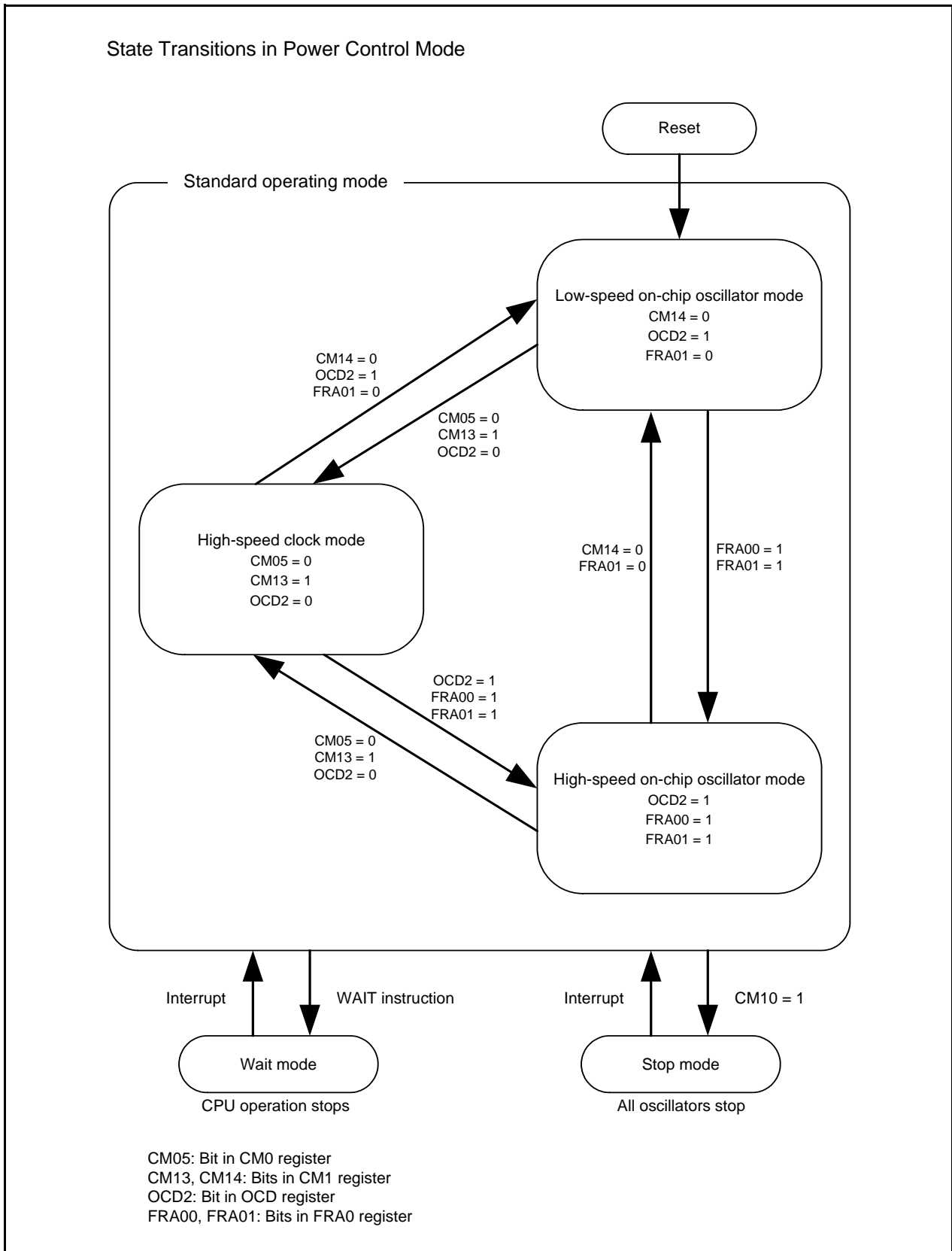


Figure 10.13 State Transitions in Power Control Mode

## 10.5 Oscillation Stop Detection Function

The oscillation stop detection function detects 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 bits OCD1 to OCD0 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**

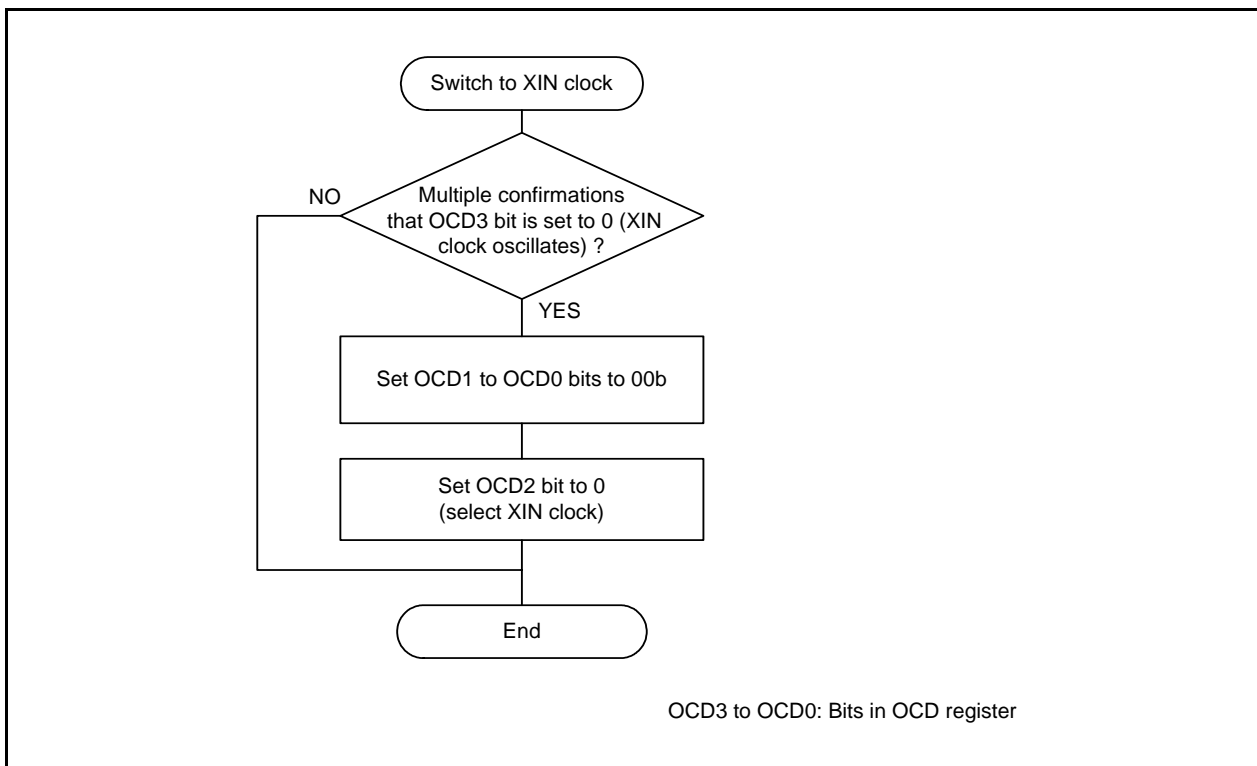
Item	Specification
Oscillation stop detection clock and frequency bandwidth	$f(\text{XIN}) \geq 2 \text{ MHz}$
Enabled condition for oscillation stop detection function	Set bits OCD1 to OCD0 to 11b
Operation at oscillation stop detection	Oscillation stop detection interrupt is generated

### 10.5.1 How to Use Oscillation Stop Detection Function

- The oscillation stop detection interrupt shares a vector with the voltage monitor 1 interrupt, the voltage monitor 2 interrupt, and the watchdog timer interrupt. When using the oscillation stop detection interrupt and watchdog timer interrupt, the interrupt source needs to be determined.  
Table 10.6 lists the Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, Voltage Monitor 1, and Voltage Monitor 2 Interrupts. Figure 10.15 shows the Example of Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, Voltage Monitor 1, or Voltage Monitor 2 Interrupt.
- When the XIN clock restarts after oscillation stop, switch the XIN clock to the clock source of the CPU clock and peripheral functions by a program.  
Figure 10.14 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 for cases where the XIN clock is stopped by an external cause, set bits OCD1 to OCD0 to 00b when the XIN clock stops or is started by a program, (stop mode is selected or the CM05 bit is changed).
- This function cannot be used when the XIN clock frequency is 2 MHz or below. In this case, set bits OCD1 to OCD0 to 00b.
- To use the low-speed on-chip oscillator clock for the CPU clock and clock sources of peripheral functions after detecting the oscillation stop, set the FRA01 bit in the FRA0 register to 0 (low-speed on-chip oscillator selected) and bits OCD1 to OCD0 to 11b.  
To use the high-speed on-chip oscillator clock for the CPU clock and clock sources of peripheral functions after detecting the oscillation stop, set the FRA00 bit to 1 (high-speed on-chip oscillator on) and the FRA01 bit to 1 (high-speed on-chip oscillator selected) and then set bits OCD1 to OCD0 to 11b.

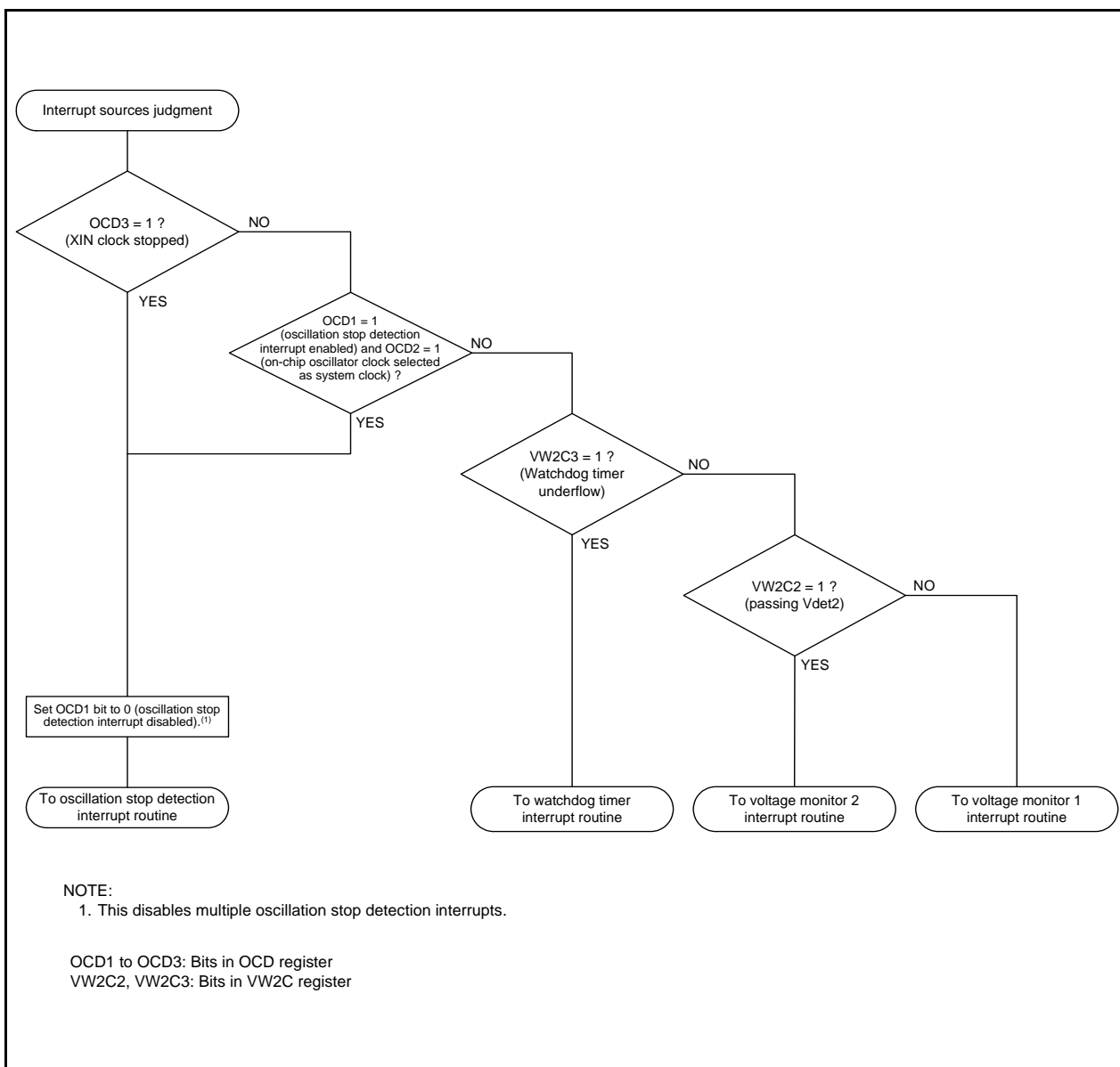
**Table 10.6 Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, Voltage Monitor 1, and Voltage Monitor 2 Interrupts**

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 OCD2 bit = 1
Watchdog timer	VW2C3 bit in VW2C register = 1
Voltage monitor 1	VW1C2 bit in VW1C register = 1
Voltage monitor 2	VW2C2 bit in VW2C register = 1



**Figure 10.14 Procedure for Switching Clock Source from Low-Speed On-Chip Oscillator to XIN Clock**





**Figure 10.15 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 in the FMR0 register to 0 (CPU rewrite mode disabled) and the CM10 bit in the CM1 register to 1 (stop mode). An instruction queue pre-reads 4 bytes from the instruction which sets the CM10 bit to 1 (stop mode) and the program stops.

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

- Program example to enter stop mode

```

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

```

### 10.6.2 Wait Mode

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

- Program example to execute the WAIT instruction

```

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

```

### 10.6.3 Oscillation Stop Detection Function

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

### 10.6.4 Oscillation Circuit Constants

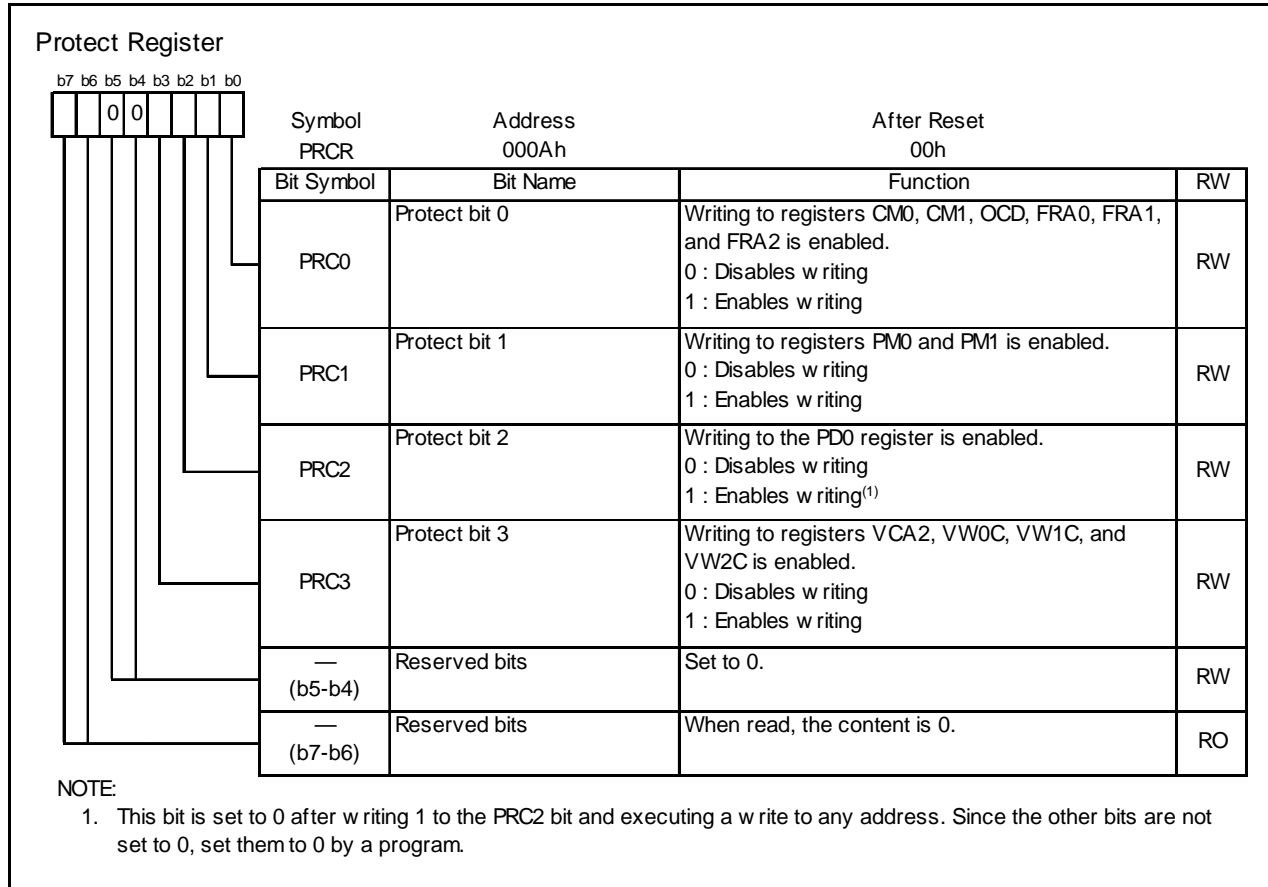
Ask the manufacturer of the oscillator to specify the best oscillation circuit constants for your system.

To use this MCU with supply voltage below  $VCC = 2.7$  V, it is recommended to set the CM11 bit in the CM1 register to 1 (on-chip feedback resistor disabled), the CM15 bit to 1 (high drive capacity), and connect the feedback resistor to the chip externally.

## 11. Protection

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

- Registers protected by PRC0 bit: Registers CM0, CM1, OCD, FRA0, FRA1, and FRA2
- Registers protected by PRC1 bit: Registers PM0 and PM1
- Registers protected by PRC2 bit: PD0 register
- Registers protected by PRC3 bit: Registers VCA2, VW0C, VW1C, and VW2C



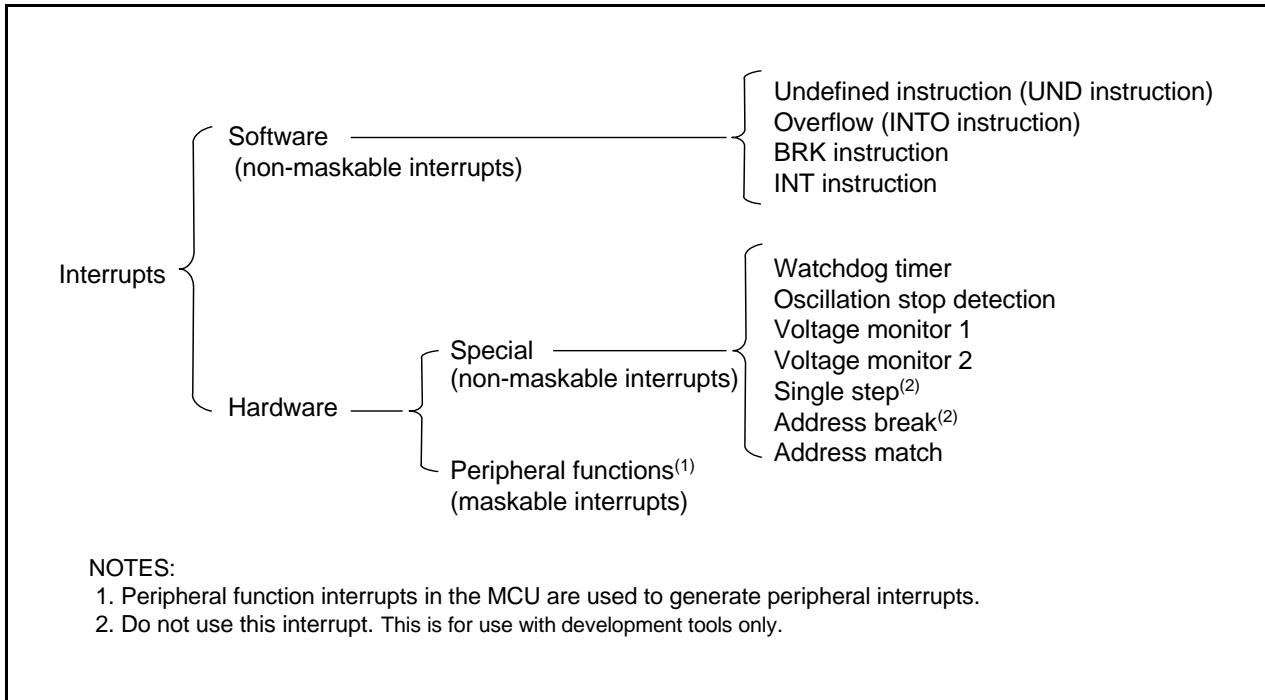
**Figure 11.1 PRCR Register**

## 12. Interrupts

### 12.1 Interrupt Overview

#### 12.1.1 Types of Interrupts

Figure 12.1 shows the types of Interrupts.



**Figure 12.1** Interrupts

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

## 12.1.2 Software Interrupts

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

### 12.1.2.1 Undefined Instruction Interrupt

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

### 12.1.2.2 Overflow Interrupt

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

### 12.1.2.3 BRK Interrupt

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

### 12.1.2.4 INT Instruction Interrupt

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

### 12.1.3 Special Interrupts

Special interrupts are non-maskable.

#### 12.1.3.1 Watchdog Timer Interrupt

The watchdog timer interrupt is generated by the watchdog timer. For details, refer to **15. Watchdog Timer**.

#### 12.1.3.2 Oscillation Stop Detection Interrupt

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

#### 12.1.3.3 Voltage Monitor 1 Interrupt

The voltage monitor 1 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 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**.

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

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

#### 12.1.3.6 Address Match Interrupt

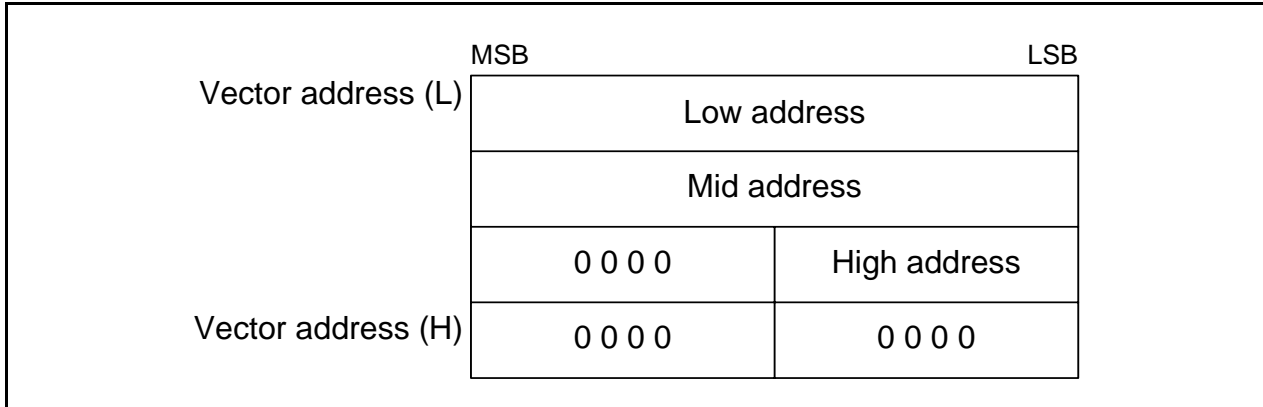
The address match interrupt is generated immediately before executing an instruction that is stored at an address indicated by registers RMAD0 to RMAD1 when the AIER0 or AIER1 bit in the AIER register is set to 1 (address match interrupt enable). For details of the address match interrupt, refer to **12.4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

### 12.1.4 Peripheral Function Interrupt

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

### 12.1.5 Interrupts and Interrupt Vectors

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



**Figure 12.2** Interrupt Vector

#### 12.1.5.1 Fixed Vector Tables

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

Table 12.1 lists the Fixed Vector Tables. The vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to **20.3 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 1, Voltage monitor 2	0FFF0h to 0FFF3h		15. 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 these interrupts. They are for use by development tools only.

### 12.1.5.2 Relocatable Vector Tables

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

**Table 12.2 Relocatable Vector Tables**

Interrupt Source	Vector Addresses <sup>(1)</sup> Address (L) to Address (H)	Software Interrupt Number	Interrupt Control Register	Reference
BRK instruction <sup>(2)</sup>	+0 to +3 (0000h to 0003h)	0	–	R8C/Tiny Series Software Manual
(Reserved)		1 to 6	–	–
Timer RC	+28 to +31 (001Ch to 001Fh)	7	TRCIC	16.3 Timer RC
Timer RD (channel 0)	+32 to +35 (0020h to 0023h)	8	TRD0IC	16.4 Timer RD
Timer RD (channel 1)	+36 to +39 (0024h to 0027h)	9	TRD1IC	
(Reserved)		10	–	–
UART2 transmit	+44 to +47 (002Ch to 002Fh)	11	S2TIC	17. Serial Interface
UART2 receive	+48 to +51 (0030h to 0033h)	12	S2RIC	
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	19. A/D Converter
(Reserved)		15	–	–
(Reserved)		16	–	–
UART0 transmit	+68 to +71 (0044h to 0047h)	17	S0TIC	17. Serial Interface
UART0 receive	+72 to +75 (0048h to 004Bh)	18	S0RIC	
(Reserved)		19	–	–
(Reserved)		20	–	–
(Reserved)		21	–	–
Timer RA	+88 to +91 (0058h to 005Bh)	22	TRAIC	16.1 Timer RA
(Reserved)		23	–	–
Timer RB	+96 to +99 (0060h to 0063h)	24	TRBIC	16.2 Timer RB
$\overline{\text{INT}}1$	+100 to +103 (0064h to 0067h)	25	INT1IC	12.2 $\overline{\text{INT}}$ 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 $\overline{\text{INT}}$ Interrupt
(Reserved)		30	–	–
(Reserved)		31	–	–
Software interrupt <sup>(2)</sup>	+128 to +131 (0080h to 0083h) to +252 to +255 (00FCh to 00FFh)	32 to 63	–	R8C/Tiny Series Software Manual

NOTES:

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



### 12.1.6 Interrupt Control

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

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

Figure 12.3 shows the Interrupt Control Register, Figure 12.4 shows Registers TRCIC, TRD0IC, and TRD1IC and Figure 12.5 shows the INTiIC Register.

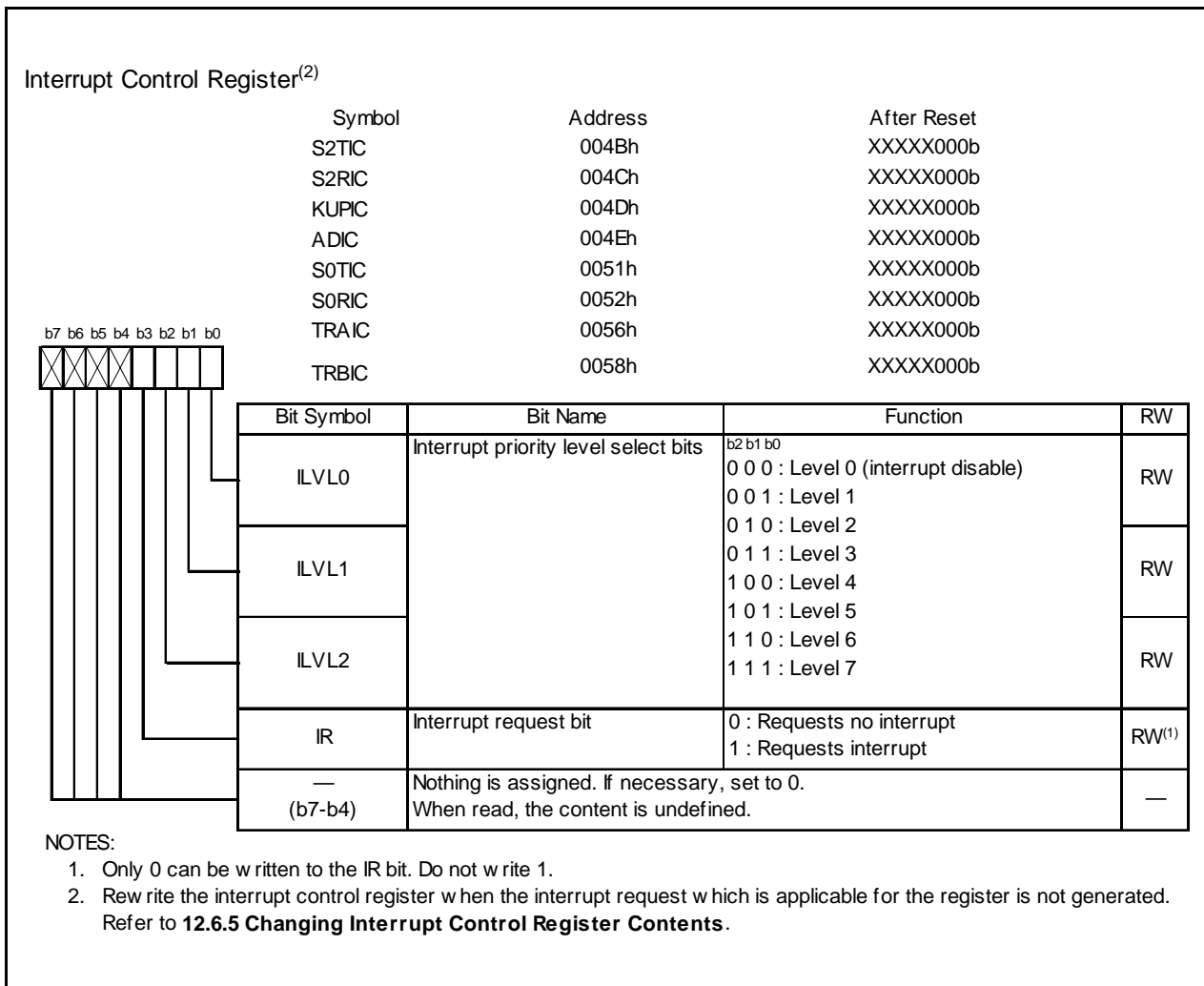


Figure 12.3 Interrupt Control Register

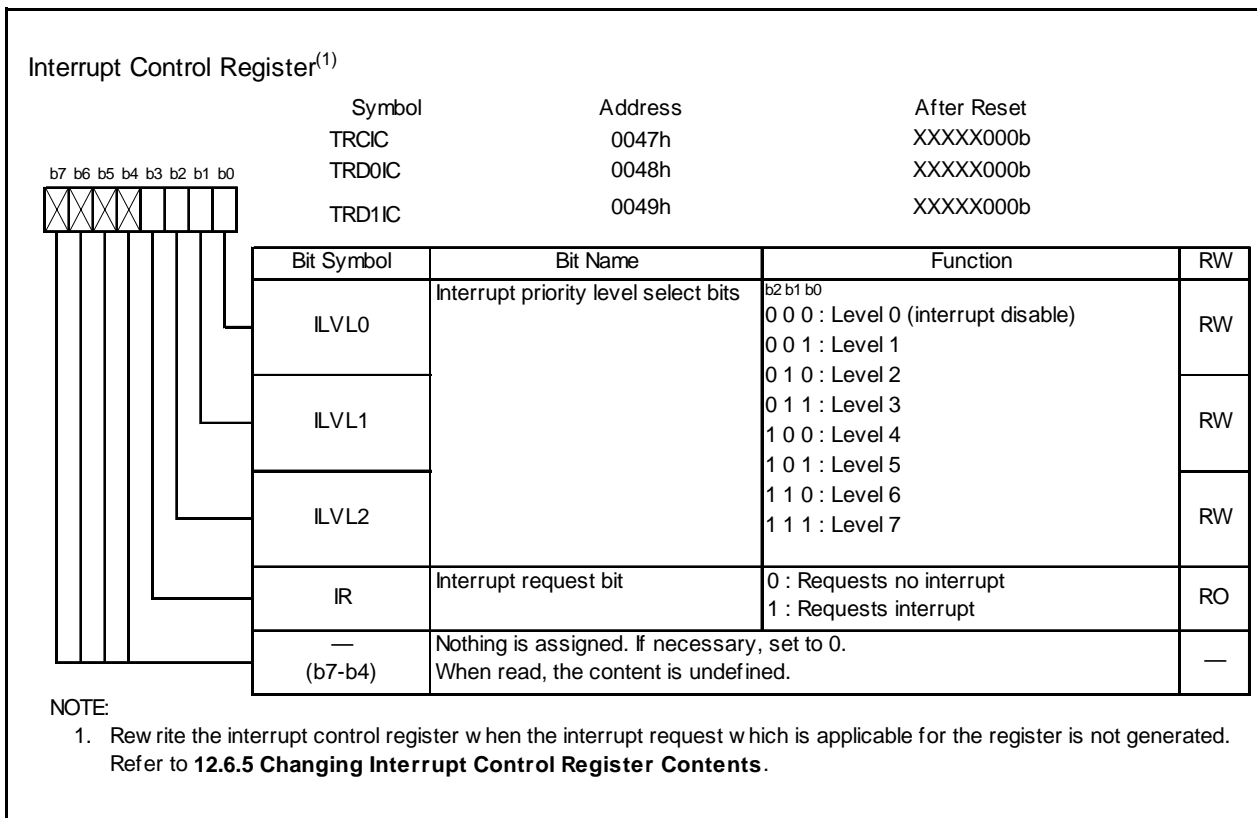
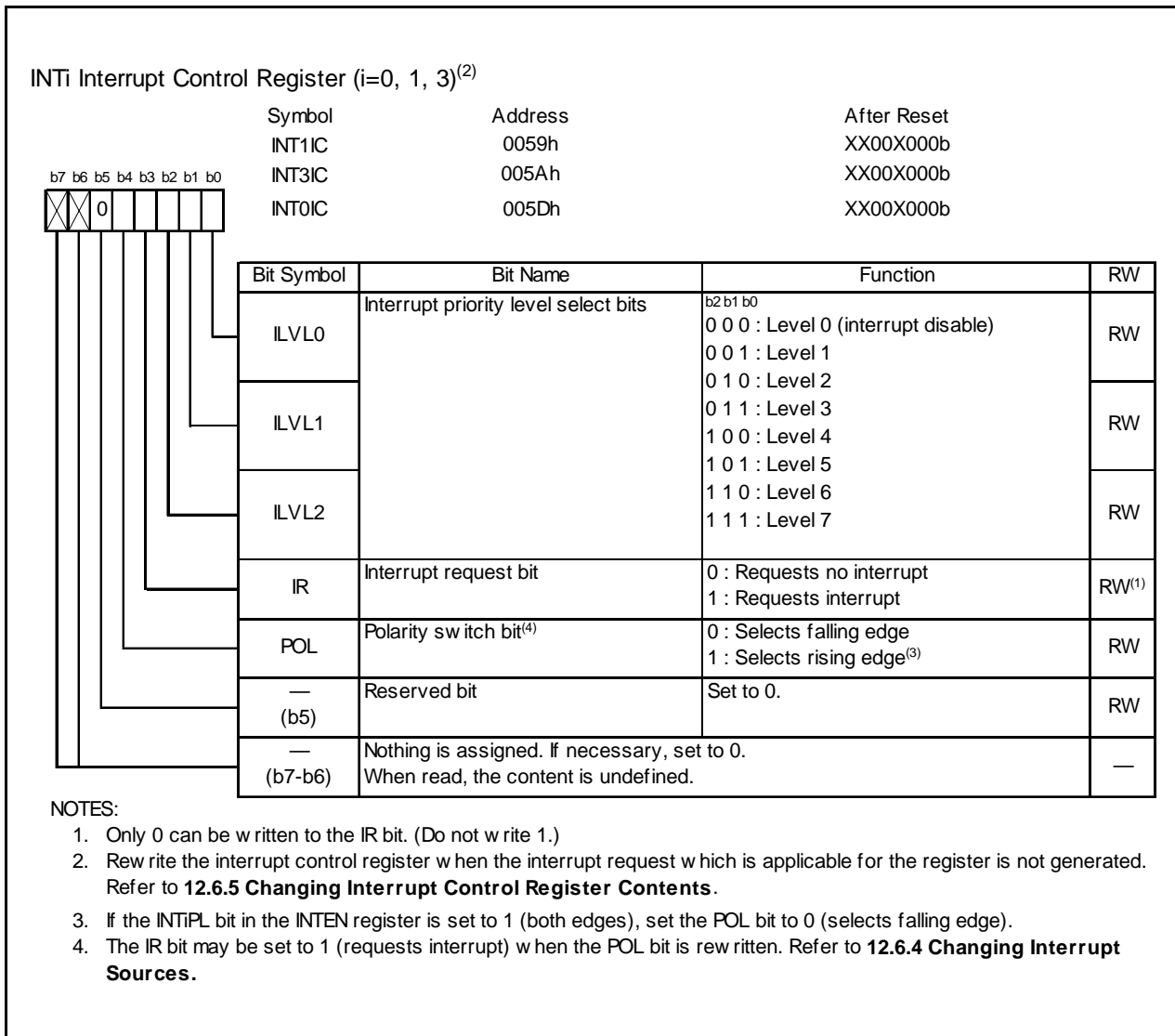


Figure 12.4 Registers TRCIC, TRD0IC, and TRD1IC

Figure 12.5 INT<sub>i</sub>IC Register

### 12.1.6.1 I Flag

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

### 12.1.6.2 IR Bit

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

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

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

### 12.1.6.3 Bits ILVL2 to ILVL0 and IPL

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


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

The following are conditions under which an interrupt is acknowledged:

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

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

**Table 12.3 Settings of Interrupt Priority Levels**

ILVL2 to ILVL0 Bits	Interrupt Priority Level	Priority Order
000b	Level 0 (interrupt disabled)	–
001b	Level 1	Low  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

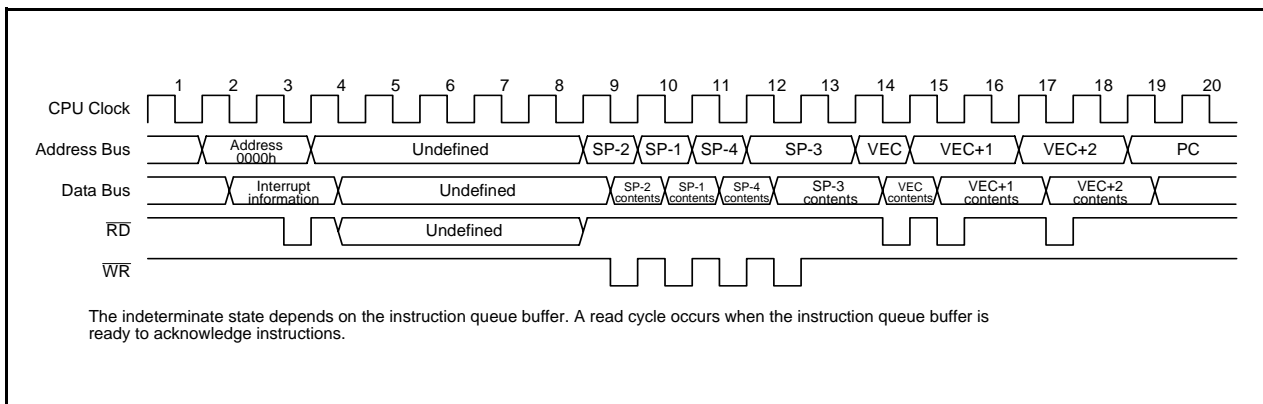
An interrupt sequence is performed between an interrupt request acknowledgement and interrupt routine execution.

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

Figure 12.6 shows the Time Required for Executing Interrupt Sequence.

- (1) The CPU gets interrupt information (interrupt number and interrupt request level) by reading address 00000h. The IR bit for the corresponding interrupt is set to 0 (interrupt not requested).<sup>(2)</sup>
- (2) The FLG register is saved to a temporary register<sup>(1)</sup> in the CPU immediately before entering the interrupt sequence.
- (3) The I, D and U flags in the FLG register are set as follows:  
The I flag is set to 0 (interrupts disabled).  
The D flag is set to 0 (single-step interrupt disabled).  
The U flag is set to 0 (ISP selected).  
However, the U flag does not change state if an INT instruction for software interrupt number 32 to 63 is executed.
- (4) The CPU's internal temporary register<sup>(1)</sup> is saved to the stack.
- (5) The PC is saved to the stack.
- (6) The interrupt priority level of the acknowledged interrupt is set in the IPL.
- (7) The starting address of the interrupt routine set in the interrupt vector is stored in the PC.

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



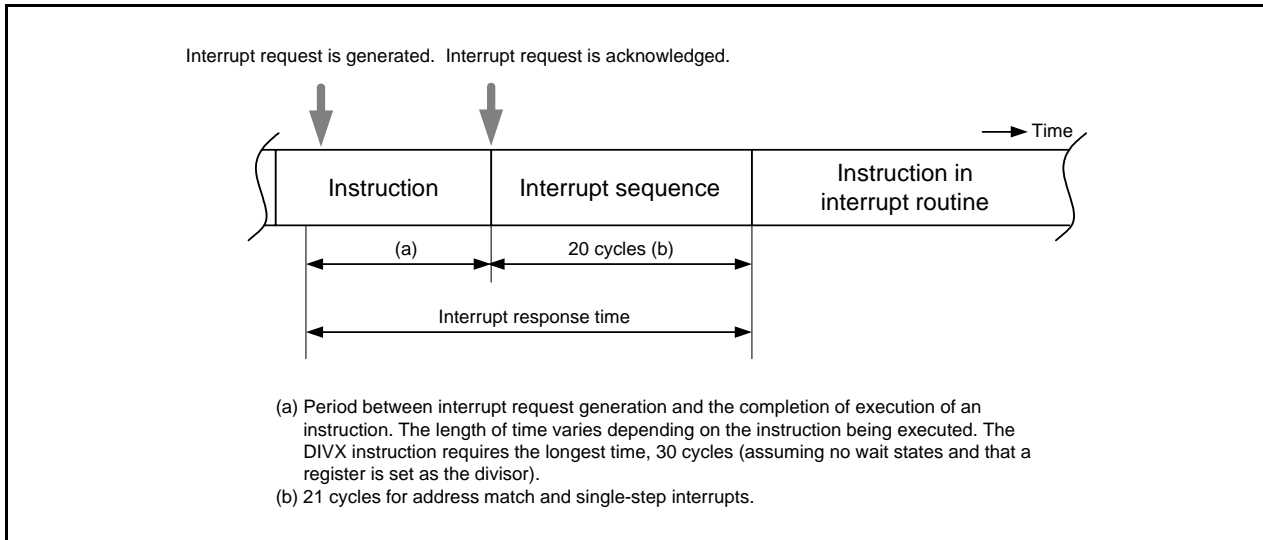
**Figure 12.6 Time Required for Executing Interrupt Sequence**

#### NOTES:

1. This register cannot be accessed by the user.
2. Refer to **12.5 Timer RC Interrupt, Timer RD Interrupt (Interrupts with Multiple Interrupt Request Sources)** for the IR bit operations of the timer RC Interrupt, timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupt, and the I<sup>2</sup>C bus Interface Interrupt.

### 12.1.6.5 Interrupt Response Time

Figure 12.7 shows the Interrupt Response Time. The interrupt response time is the period between an interrupt request generation and the execution of the first instruction in the interrupt routine. The interrupt response time includes the period between interrupt request generation and the completion of execution of the instruction (refer to (a) in **Figure 12.7**) and the period required to perform the 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 or special interrupt request is acknowledged, the level listed in Table 12.5 is set in the IPL.

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

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

Interrupt Source	Value Set in IPL
Watchdog timer, oscillation stop detection, voltage monitor 1, 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 an extended 16 bits, 4 high-order bits in the PC and 4 high-order (IPL) and 8 low-order bits in the FLG register, are saved to the stack, the 16 low-order bits in the PC are saved.

Figure 12.8 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 a single instruction.

NOTE:

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

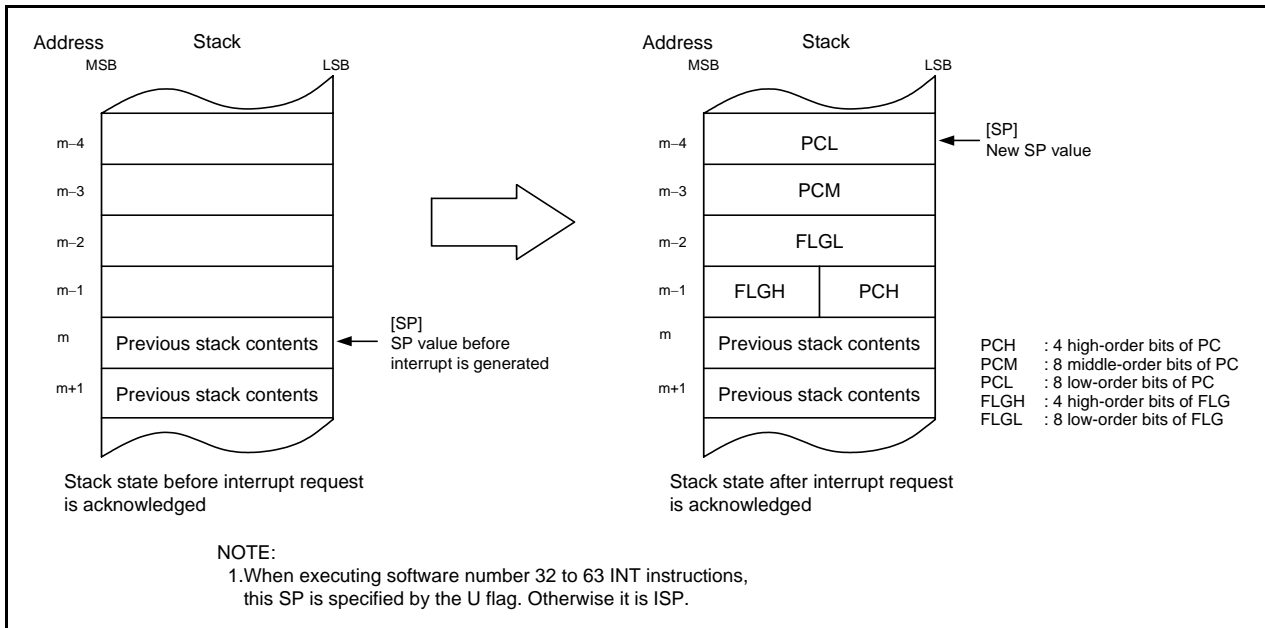


Figure 12.8 Stack State Before and After Acknowledgement of Interrupt Request

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

Figure 12.9 shows the Register Saving Operation.

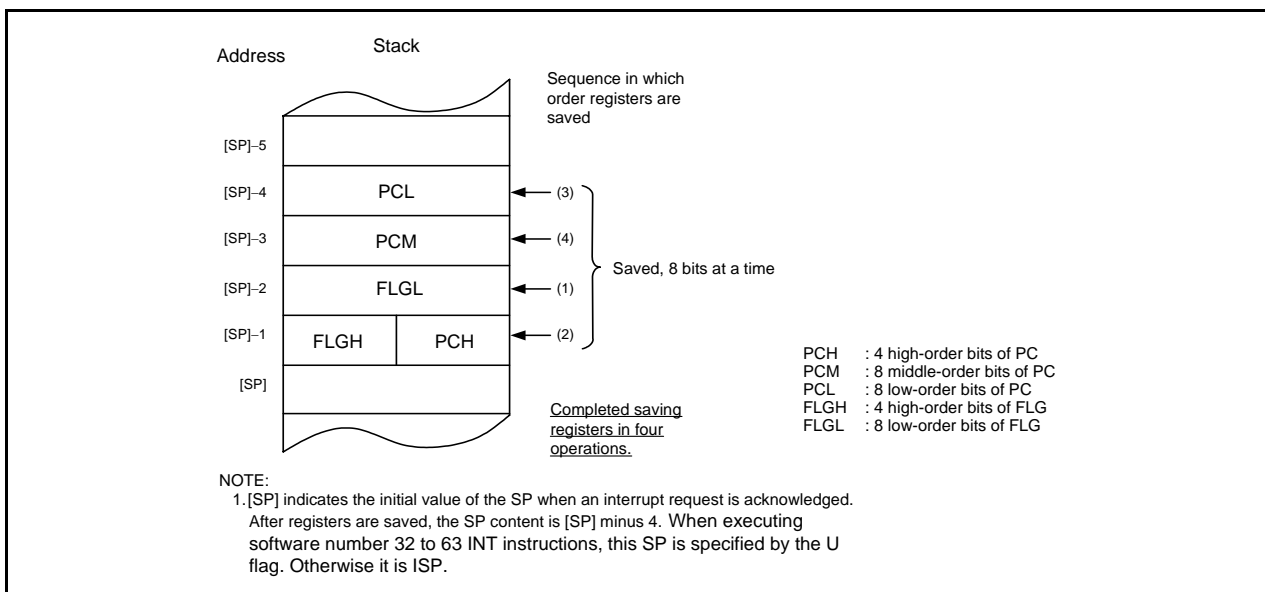


Figure 12.9 Register Saving Operation

### 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 restored. The program, that was running before the interrupt request was acknowledged, starts running again.

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

### 12.1.6.9 Interrupt Priority

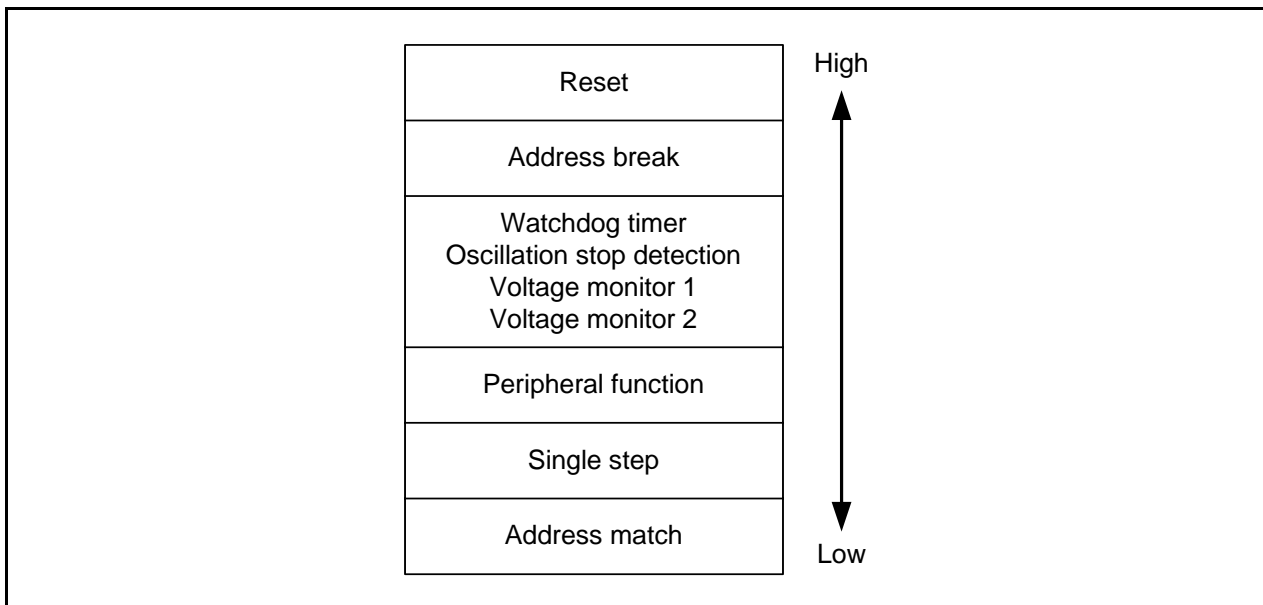
If two or more interrupt requests are generated while a single instruction is being executed, the interrupt with the higher priority is acknowledged.

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

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

Figure 12.10 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, as shown in Figure 12.11.

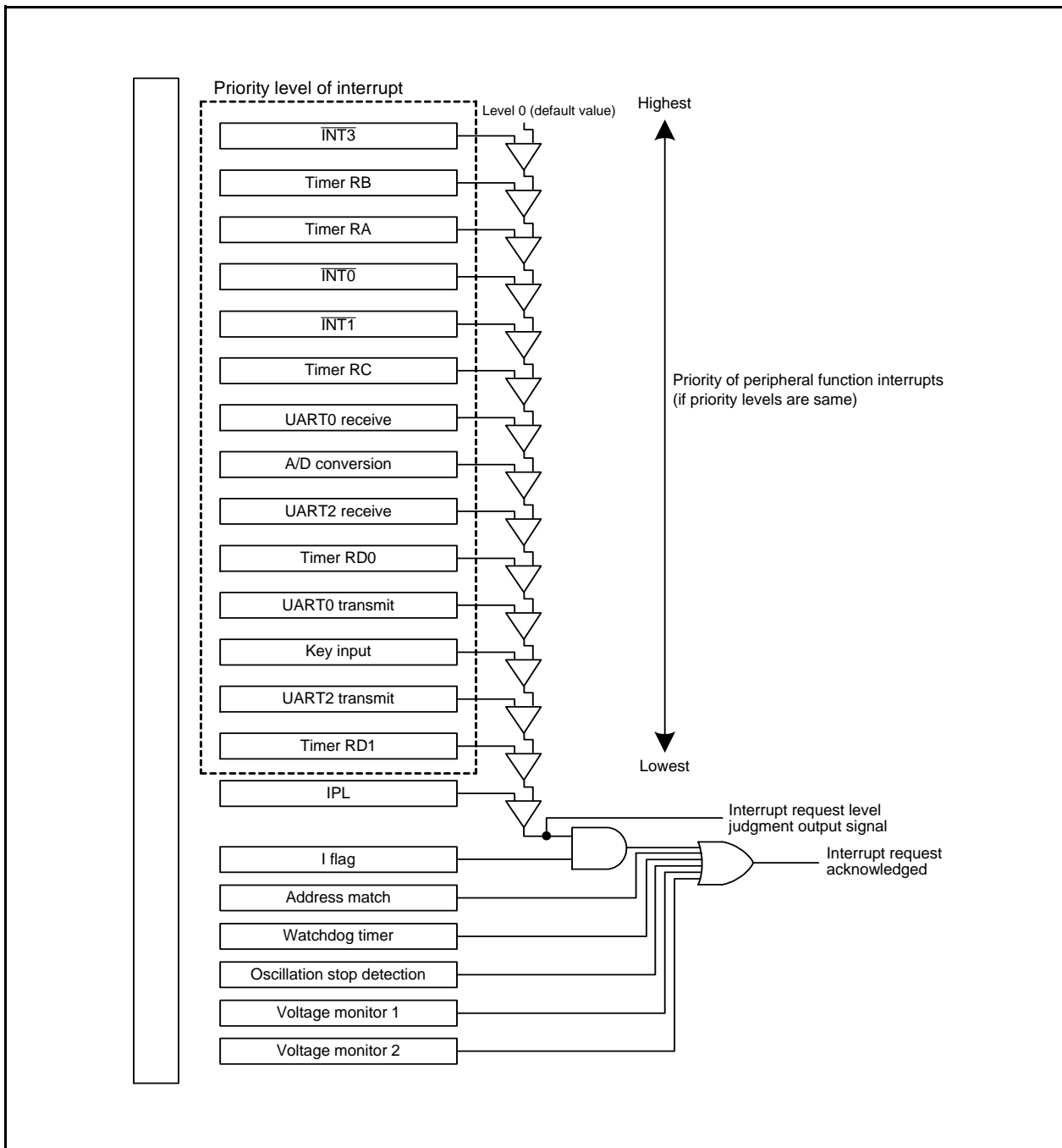


Figure 12.11 Interrupt Priority Level Judgement Circuit

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

### 12.2.1 $\overline{\text{INT}}_i$ Interrupt ( $i = 0, 1, 3$ )

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 RC and timer RD, and the external trigger input of timer RB.

Figure 12.12 shows the  $\text{PMR}$  Register, Figure 12.13 shows the  $\text{INTEN}$  Register, Figure 12.14 shows the  $\text{INTF}$  Register, and Figure 12.15 shows the  $\text{TRAIOC}$  Register.

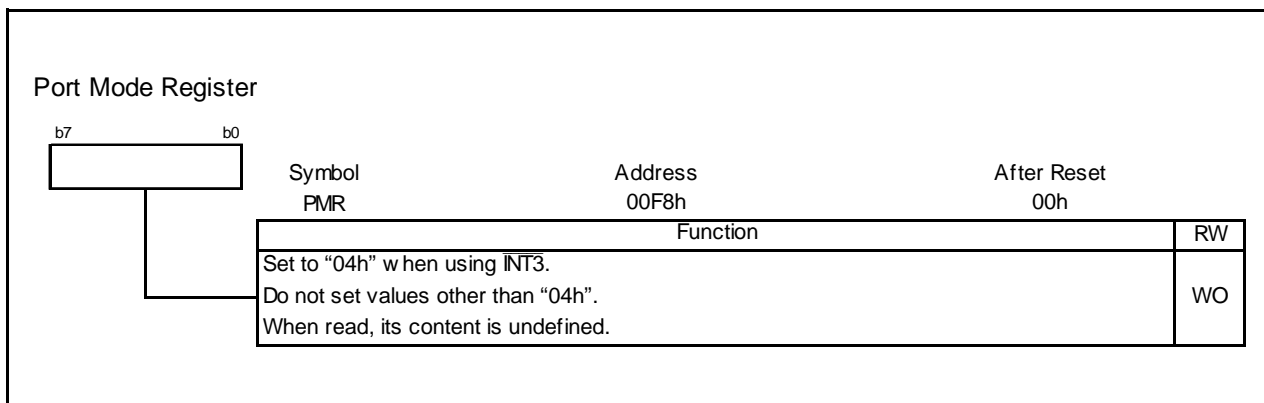


Figure 12.12 PMR Register

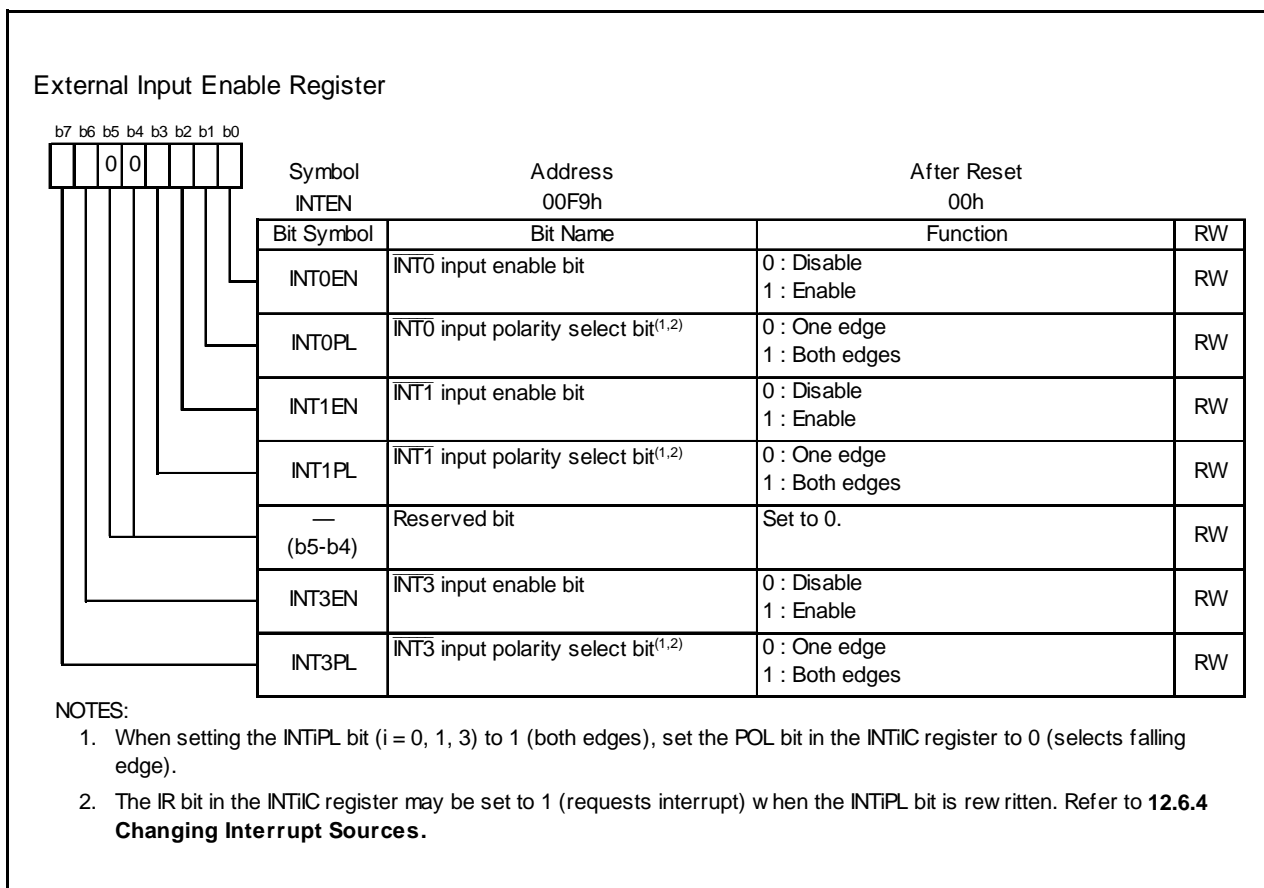


Figure 12.13 INTEN Register

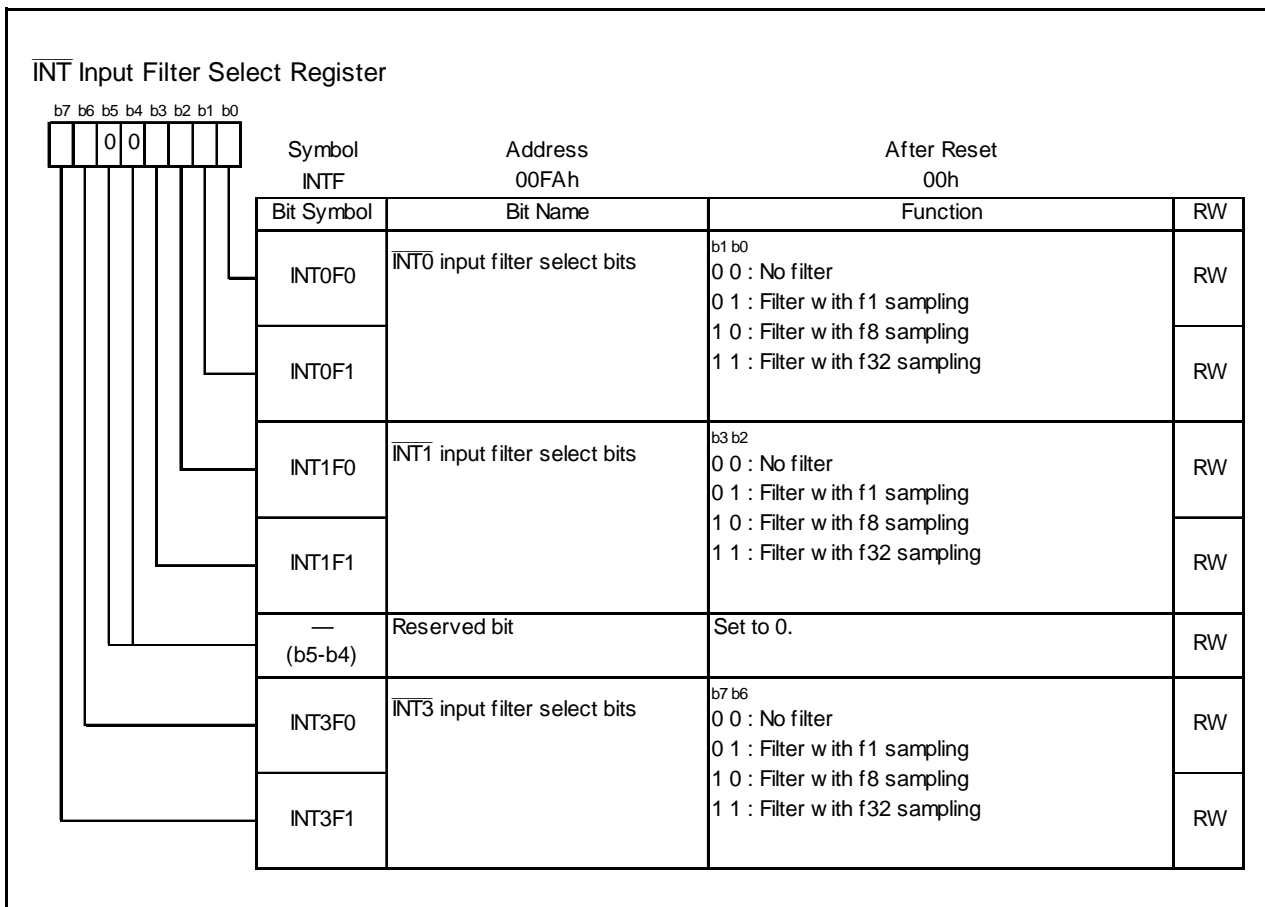


Figure 12.14 INTF Register

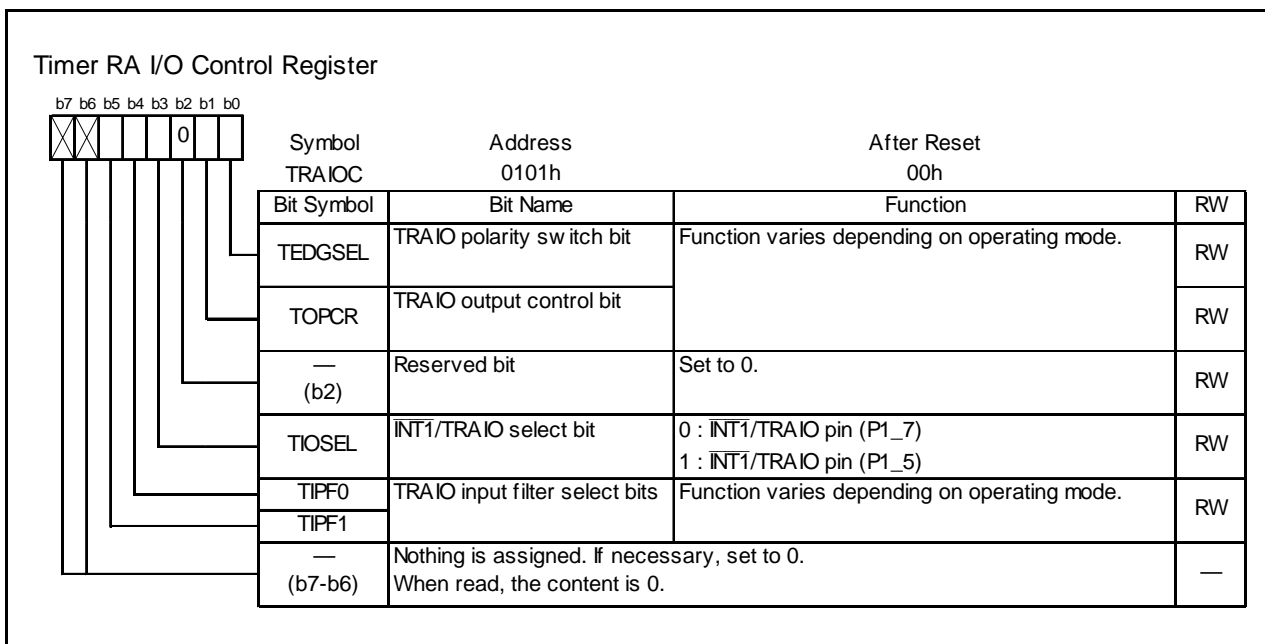


Figure 12.15 TRAI0C Register

### 12.2.2 $\overline{\text{INT}}_i$ Input Filter (i = 0, 1, 3)

The  $\overline{\text{INT}}_i$  input contains a digital filter. The sampling clock is selected by bits  $\text{INTiF1}$  to  $\text{INTiF0}$  in the  $\text{INTF}$  register.

The  $\overline{\text{INT}}_i$  level is sampled every sampling clock cycle and if the sampled input level matches three times, the IR bit in the  $\text{INTiC}$  register is set to 1 (interrupt requested).

Figure 12.16 shows the Configuration of  $\overline{\text{INT}}_i$  Input Filter. Figure 12.17 shows an Operating Example of  $\overline{\text{INT}}_i$  Input Filter.

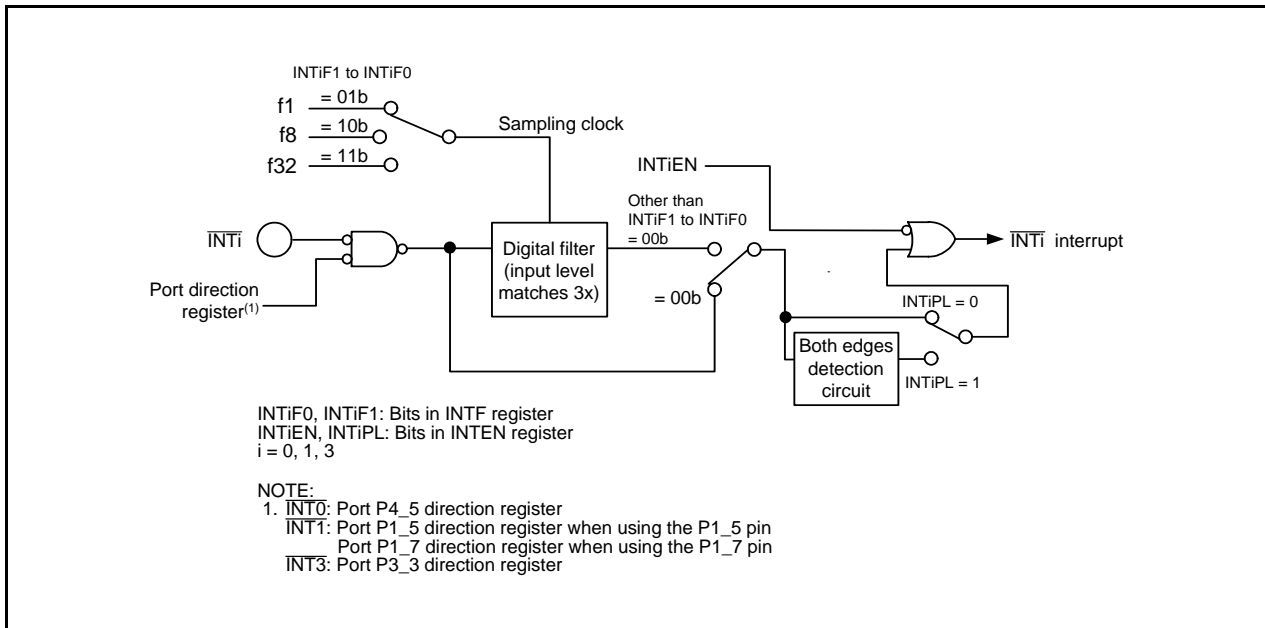


Figure 12.16 Configuration of  $\overline{\text{INT}}_i$  Input Filter

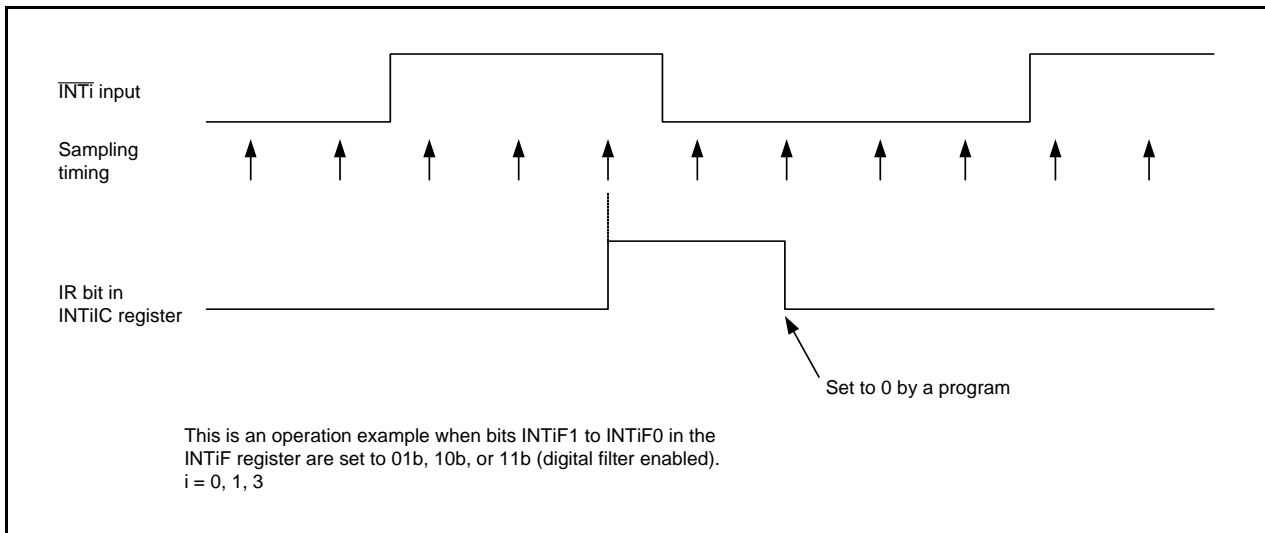


Figure 12.17 Operating Example of  $\overline{\text{INT}}_i$  Input Filter

### 12.3 Key Input Interrupt

A key input interrupt request is generated by one of the input edges of pins  $\overline{K10}$  to  $\overline{K13}$ . The key input interrupt can be used as a key-on wake-up function to exit wait or stop mode.

The  $KIiEN$  ( $i = 0$  to  $3$ ) bit in the  $KIEN$  register can select whether or not the pins are used as  $\overline{KIi}$  input. The  $KIiPL$  bit in the  $KIEN$  register can select the input polarity.

When inputting “L” to the  $\overline{KIi}$  pin which sets the  $KIiPL$  bit to 0 (falling edge), the input of the other pins  $\overline{K10}$  to  $\overline{K13}$  is not detected as interrupts. Also, when inputting “H” to the  $\overline{KIi}$  pin, which sets the  $KIiPL$  bit to 1 (rising edge), the input of the other pins  $\overline{K10}$  to  $\overline{K13}$  is not detected as interrupts.

Figure 12.18 shows a Block Diagram of Key Input Interrupt.

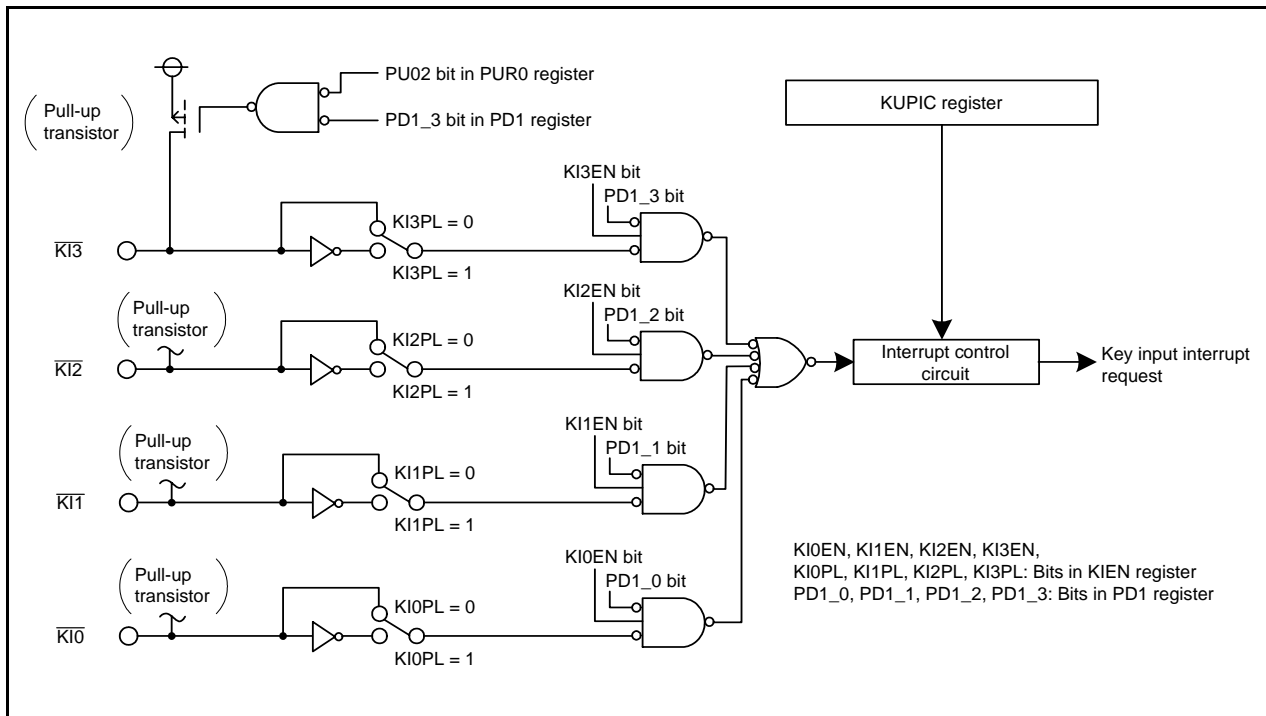


Figure 12.18 Block Diagram of Key Input Interrupt

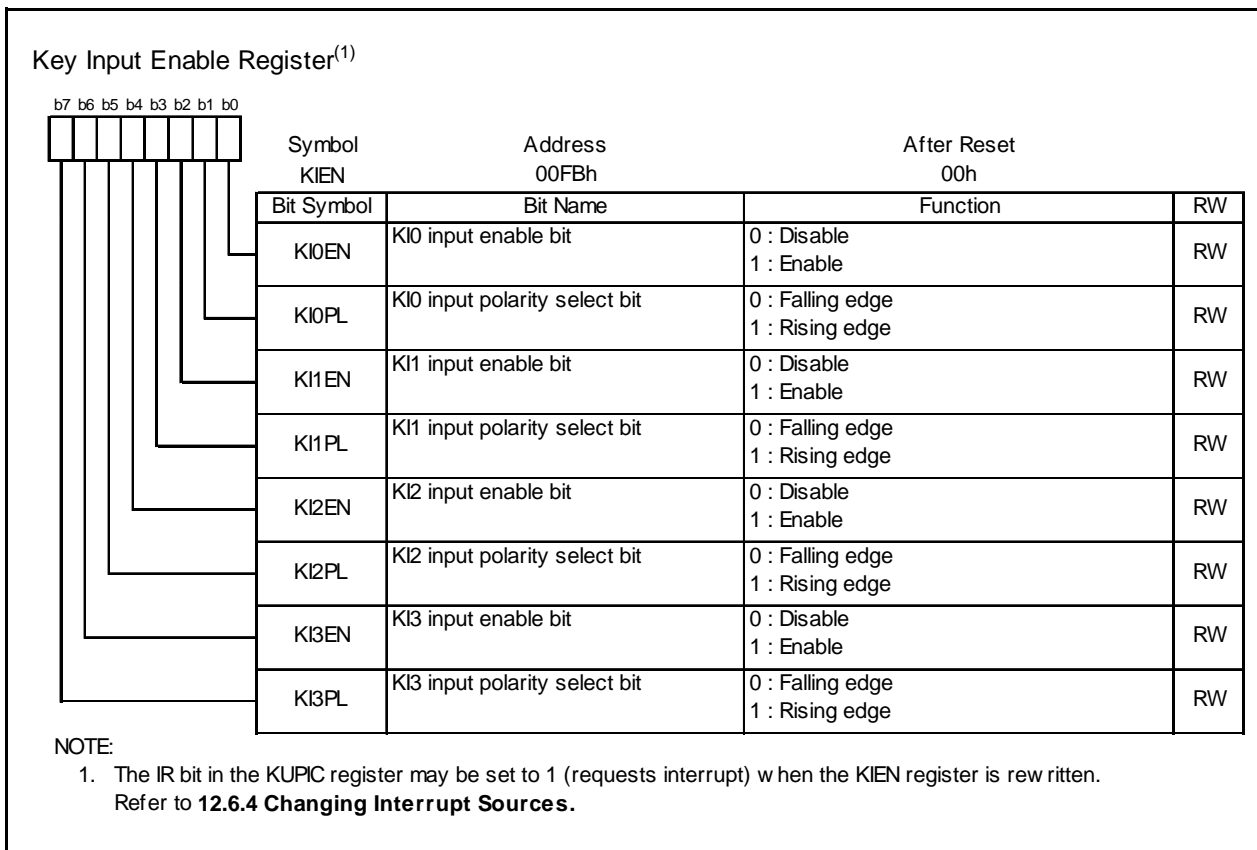


Figure 12.19 KIEN Register

## 12.4 Address Match Interrupt

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

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

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

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

Figure 12.20 shows Registers AIER and RMAD0 to RMAD1.

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

Address Indicated by RMADi Register (i = 0 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
<ul style="list-style-type: none"> <li>• Instructions other than the above</li> </ul>	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

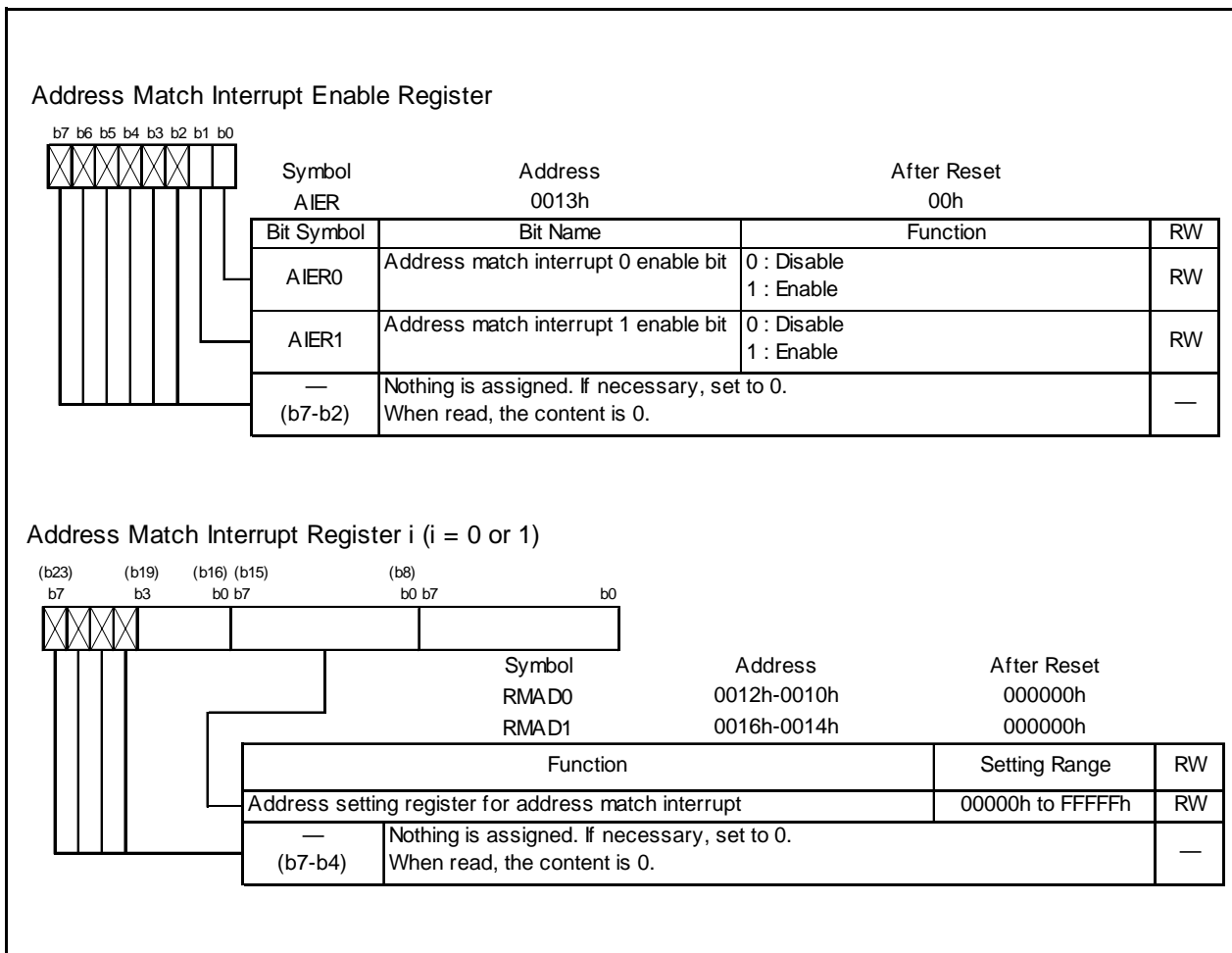


Figure 12.20 Registers AIER and RMAD0 to RMAD1

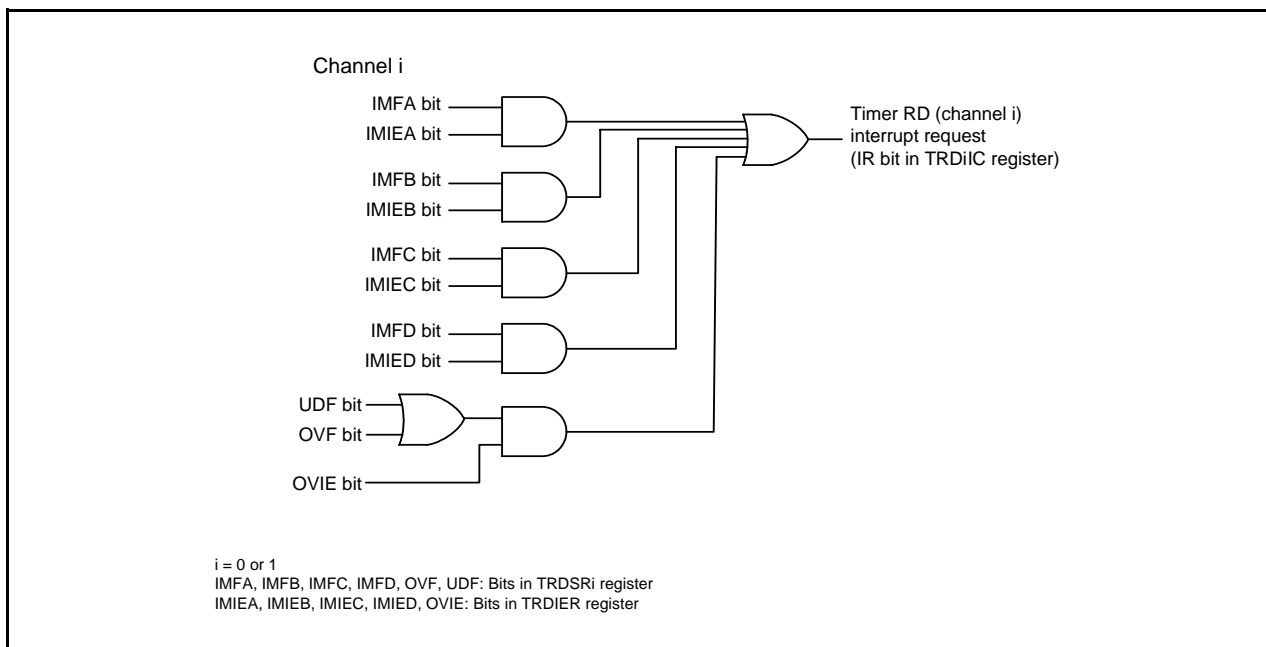


## 12.5 Timer RC Interrupt, Timer RD Interrupt (Interrupts with Multiple Interrupt Request Sources)

The timer RC interrupt, timer RD (channel 0) interrupt, and timer RD (channel 1) interrupt each have multiple interrupt request sources. An interrupt request is generated by the logical OR of several interrupt request factors and is reflected in the IR bit in the corresponding interrupt control register. Therefore, each of these peripheral functions has its own interrupt request source status register (status register) and interrupt request source enable register (enable register) to control the generation of interrupt requests (change the IR bit in the interrupt control register). Table 12.8 lists the Registers Associated with Timer RC Interrupt and Timer RD Interrupt and Figure 12.21 shows a Block Diagram of Timer RD Interrupt.

**Table 12.8 Registers Associated with Timer RC Interrupt and Timer RD Interrupt**

Peripheral Function Name		Status Register of Interrupt Request Source	Enable Register of Interrupt Request Source	Interrupt Control Register
Timer RC		TRCSR	TRCIER	TRCIC
Timer RD	Channel 0	TRDSR0	TRDIER0	TRD0IC
	Channel 1	TRDSR1	TRDIER1	TRD1IC



**Figure 12.21 Block Diagram of Timer RD Interrupt**

As with other maskable interrupts, the timer RC interrupt, timer RD (channel 0) interrupt, and timer RD (channel 1) interrupt are controlled by the combination of the I flag, IR bit, bits ILVL0 to ILVL2, and IPL. However, since each interrupt source is generated by a combination of multiple interrupt request sources, the following differences from other maskable interrupts apply:

- When bits in the enable register corresponding to bits set to 1 in the status register 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, 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 even if 0 is written to the IR bit.
- Individual bits in the status register are not automatically set to 0 even if the interrupt is acknowledged. Therefore, the IR bit is also not 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 for how to set individual bits 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 the individual peripheral functions (**16.3 Timer RC** and **16.4 Timer RD**) 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 address 00000h by a program. When a maskable interrupt request is acknowledged, the CPU reads interrupt information (interrupt number and interrupt request level) from 00000h in the interrupt sequence. At this time, the acknowledged interrupt IR bit is set to 0.

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

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

### 12.6.3 External Interrupt and Key Input Interrupt

Either “L” level or an “H” level of width shown in the Electrical Characteristics is necessary for the signal input to pins  $\overline{\text{INT0}}$ ,  $\overline{\text{INT1}}$ ,  $\overline{\text{INT3}}$  and pins  $\overline{\text{KI0}}$  to  $\overline{\text{KI3}}$ , regardless of the CPU clock.

For details, refer to **Table 22.19** (VCC = 5V), **Table 22.25** (VCC = 3V), **Table 22.31** (VCC = 2.2V) **External Interrupt INT<sub>i</sub> (i = 0, 1, 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, changes of interrupt sources include all factors that change the interrupt sources assigned to individual software interrupt numbers, polarities, and timing. Therefore, if a mode change of a peripheral function involves interrupt sources, edge polarities, and timing, set the IR bit to 0 (no interrupt requested) after the change. Refer to the individual peripheral function for its related interrupts.

Figure 12.22 shows an Example of Procedure for Changing Interrupt Sources.

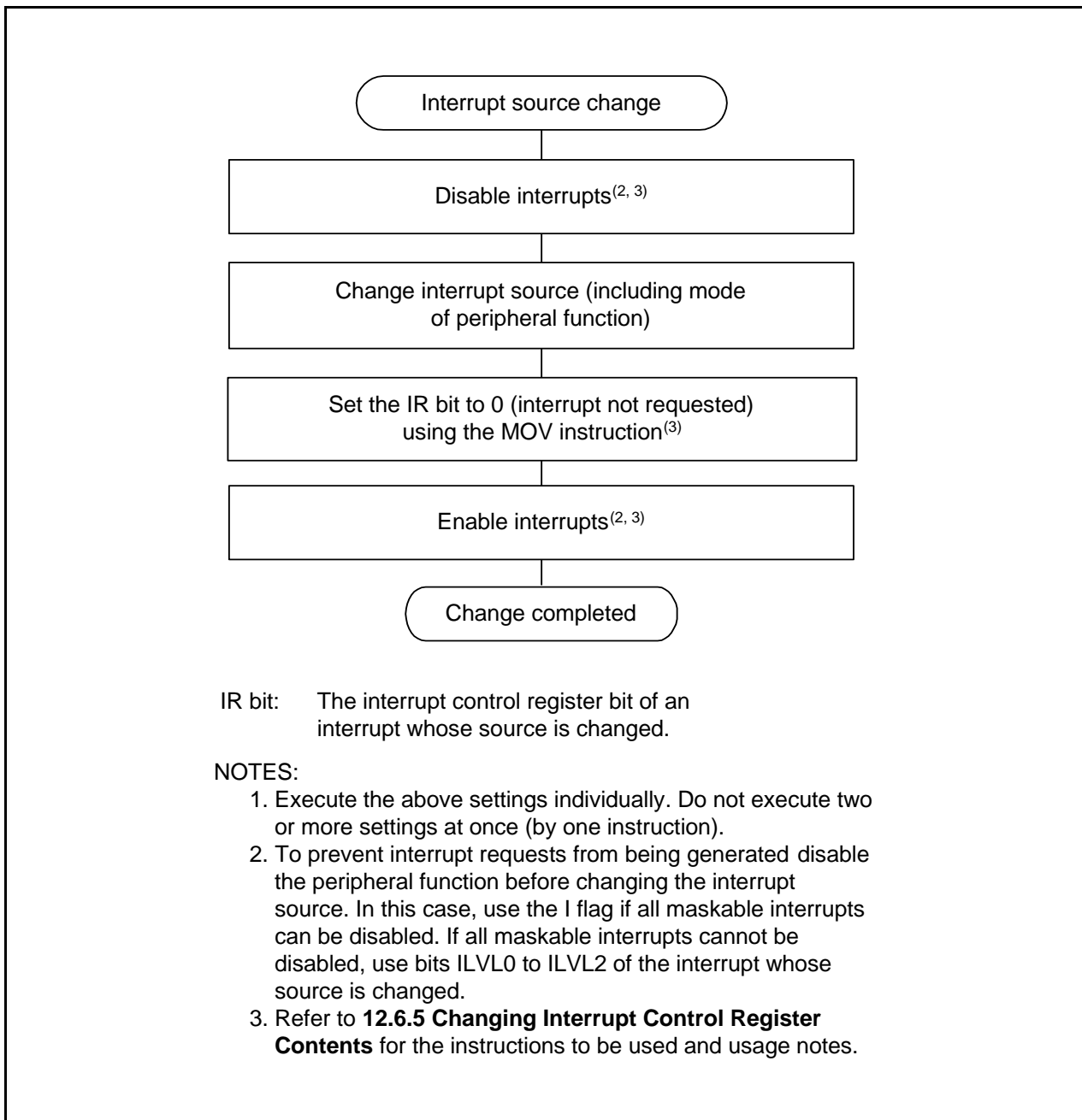


Figure 12.22 Example of Procedure for Changing Interrupt Sources

### 12.6.5 Changing Interrupt Control Register Contents

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

#### Changing any bit other than IR bit

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

#### Changing IR bit

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

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

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

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

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

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

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

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

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

## 13. ID Code Areas

### 13.1 Overview

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

The ID code areas are assigned to 0FFDFh, 0FFE3h, 0FFE8h, 0FEFh, 0FFF3h, 0FFF7h, and 0FFFBh of the respective vector highest-order addresses of the fixed vector table. Figure 13.1 shows the ID Code Areas.

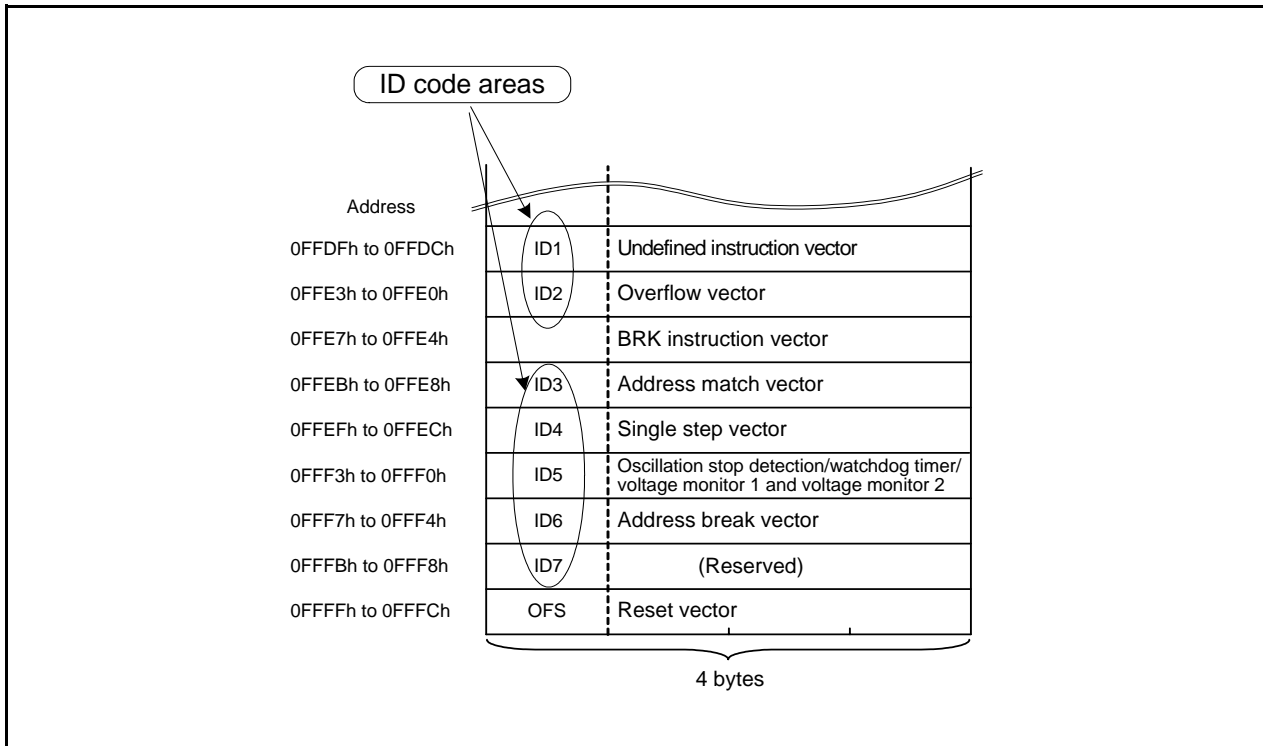


Figure 13.1 ID Code Areas

### 13.2 Functions

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

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

## 13.3 Notes on ID Code Areas

### 13.3.1 Setting Example of ID Code Areas

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

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

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

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

## 14. Option Function Select Area

### 14.1 Overview

The option function select area is used to select the MCU state after reset or the function to prevent rewriting in parallel I/O mode. The reset vector highest-order-address, 0FFFFh, is assigned as the option function select area. Figure 14.1 shows the Option Function Select Area.

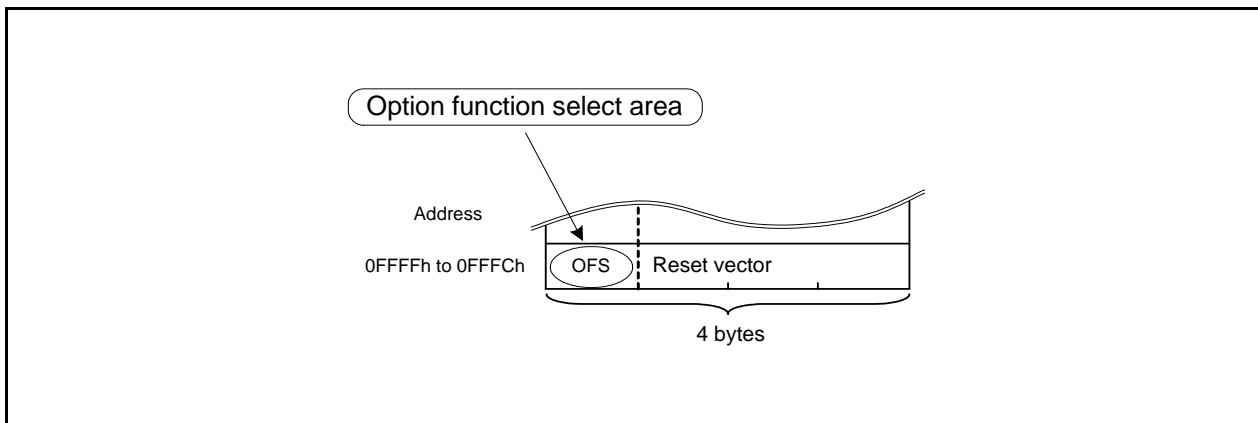


Figure 14.1 Option Function Select Area



## 14.2 OFS Register

The OFS register is used to select the MCU state after reset or the function to prevent rewriting in parallel I/O mode. Figure 14.2 shows the OFS Register.

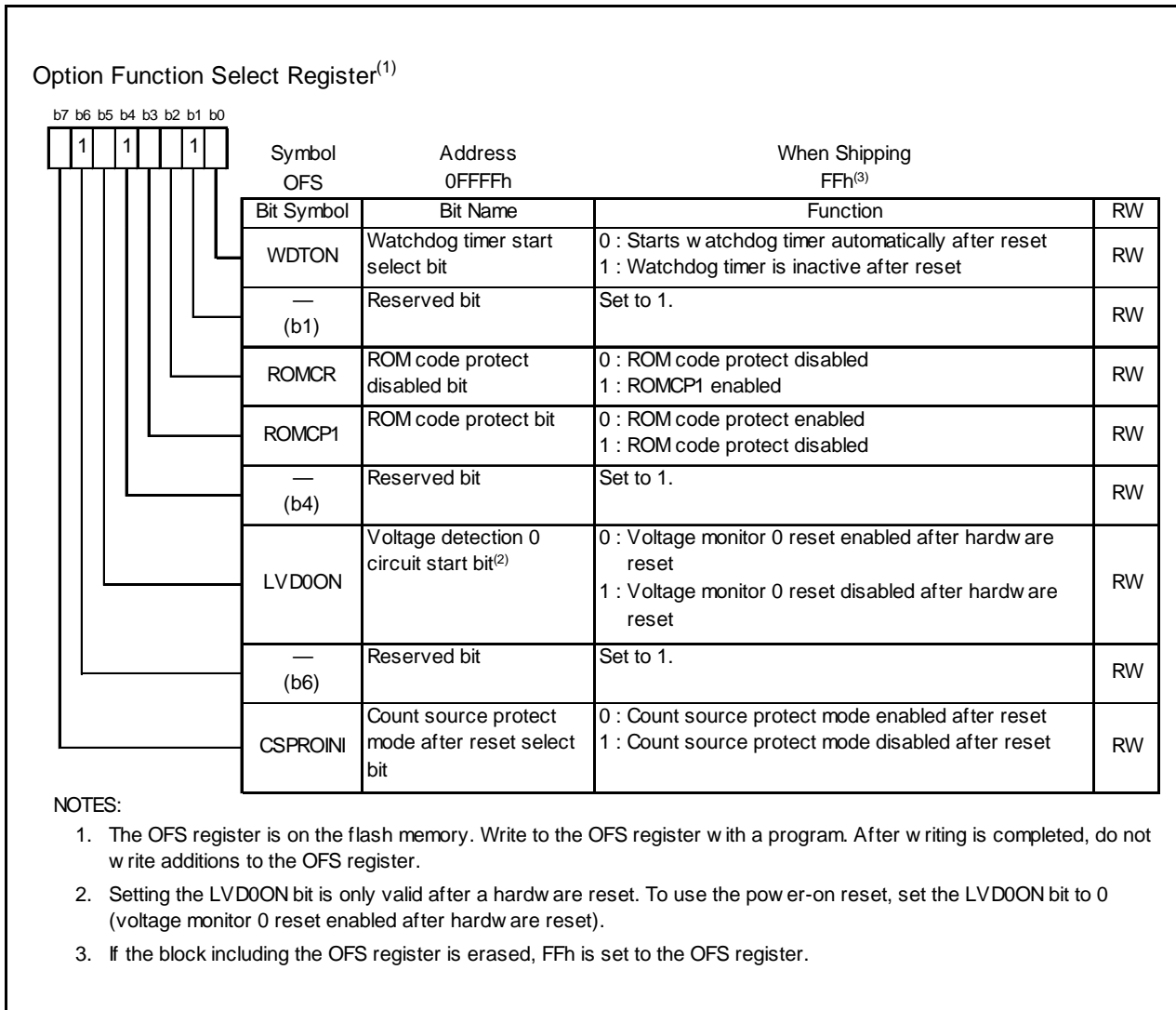


Figure 14.2 OFS Register

## 14.3 Notes on Option Function Select Area

### 14.3.1 Setting Example of Option Function Select Area

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

- To set FFh in the OFS register

```
.org 00FFFCH
```

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

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

## 15. Watchdog Timer

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

Table 15.1 lists information on the Watchdog Timer Specifications.

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

Figure 15.1 shows the Block Diagram of Watchdog Timer. Figure 15.2 shows the Registers WDTR, WDTS, and WDC. Figure 15.3 shows the Registers CSPR and OFS.

**Table 15.1 Watchdog Timer Specifications**

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 the following can be selected <ul style="list-style-type: none"> <li>• After reset, count starts automatically</li> <li>• Count starts by writing to WDTS register</li> </ul>	
Count stop condition	Stop mode, wait mode	None
Reset condition of watchdog timer	<ul style="list-style-type: none"> <li>• Reset</li> <li>• Write 00h to the WDTR register before writing FFh</li> <li>• Underflow</li> </ul>	
Operation at the time of underflow	Watchdog timer interrupt or watchdog timer reset	Watchdog timer reset
Select functions	<ul style="list-style-type: none"> <li>• Division ratio of prescaler Selected by the WDC7 bit in the WDC register</li> <li>• Count source protection mode Whether count source protection mode is enabled or disabled after a reset can be selected by the CSPROINI bit in the OFS register (flash memory). If count source protection mode is disabled after a reset, it can be enabled or disabled by the CSPRO bit in the CSPR register (program).</li> <li>• Starts or stops of the watchdog timer after a reset Selected by the WDTON bit in the OFS register (flash memory).</li> </ul>	

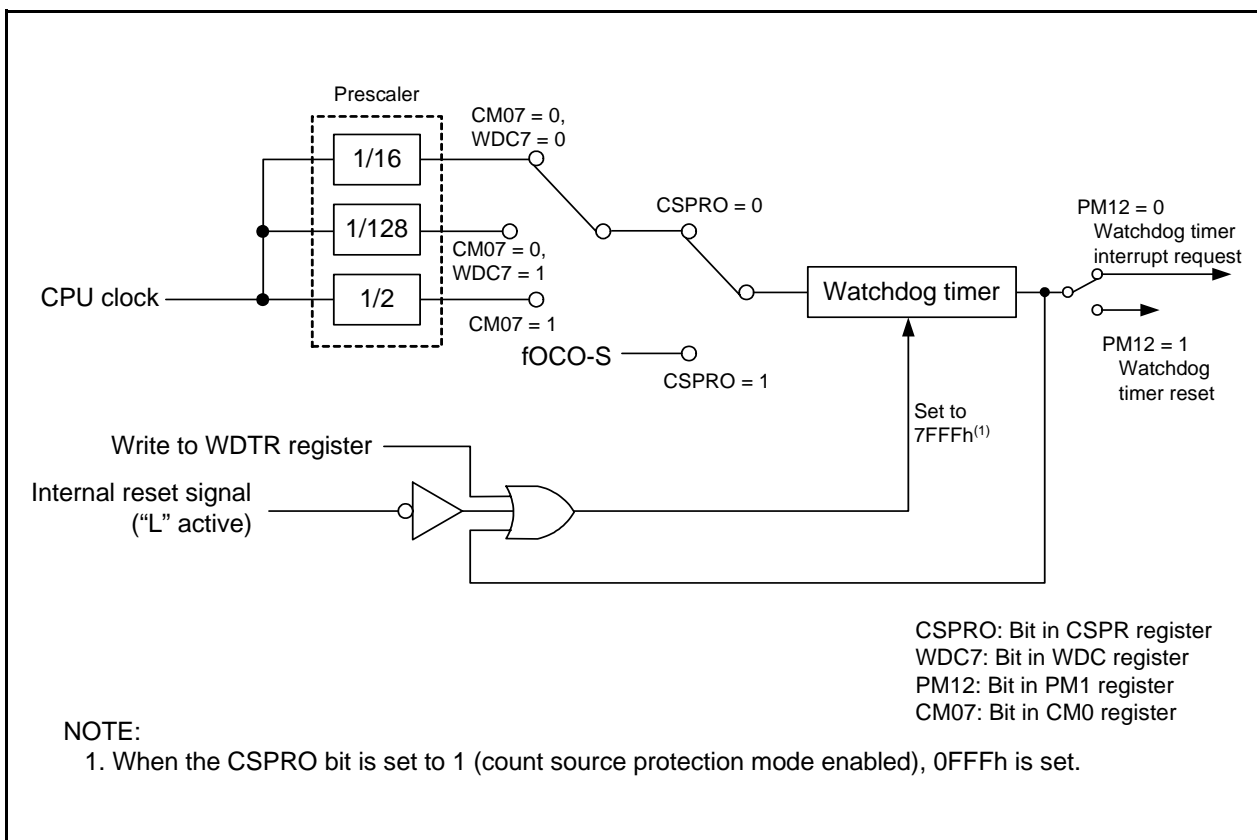


Figure 15.1 Block Diagram of Watchdog Timer

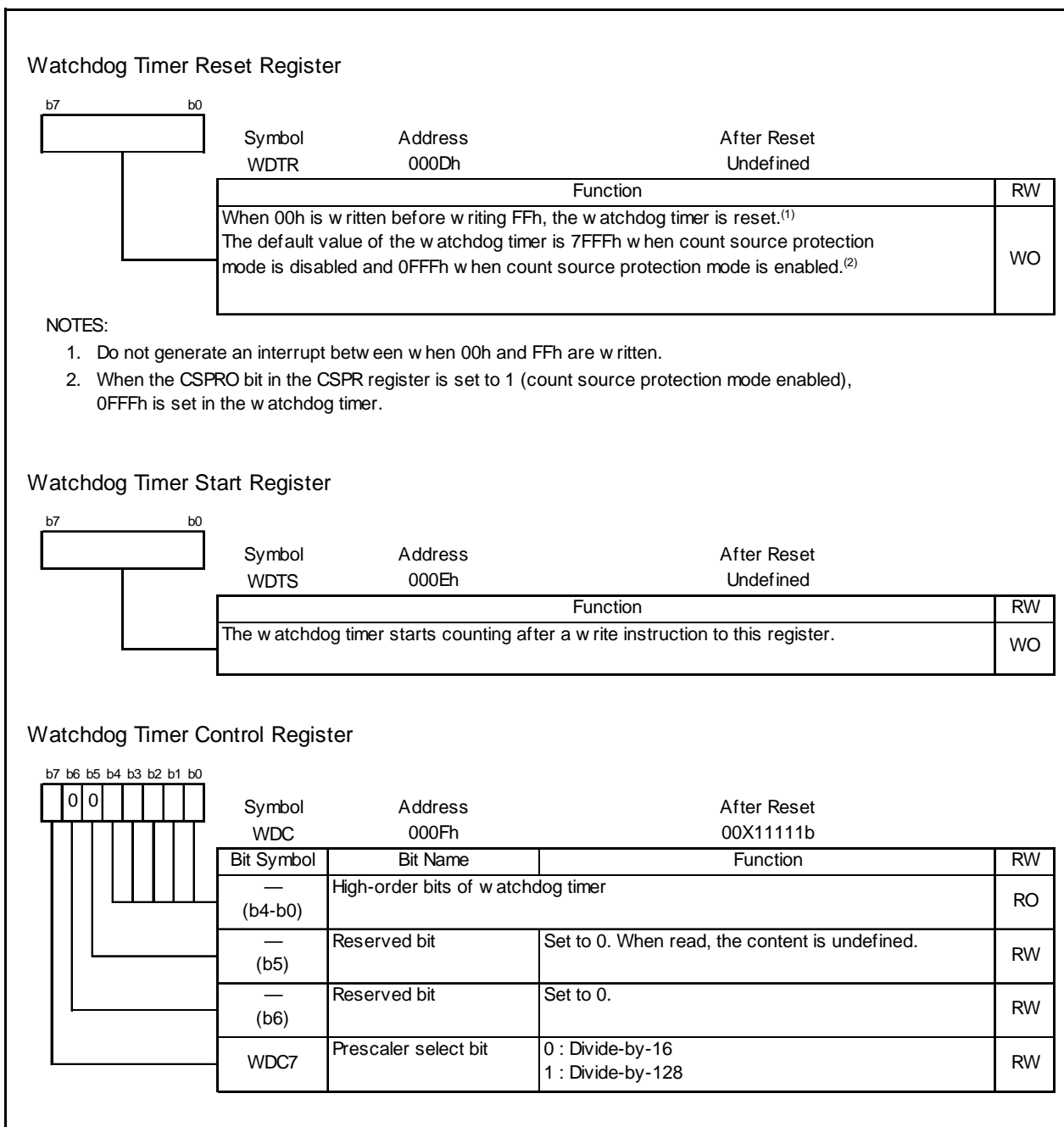


Figure 15.2 Registers WDTR, WDTS, and WDC

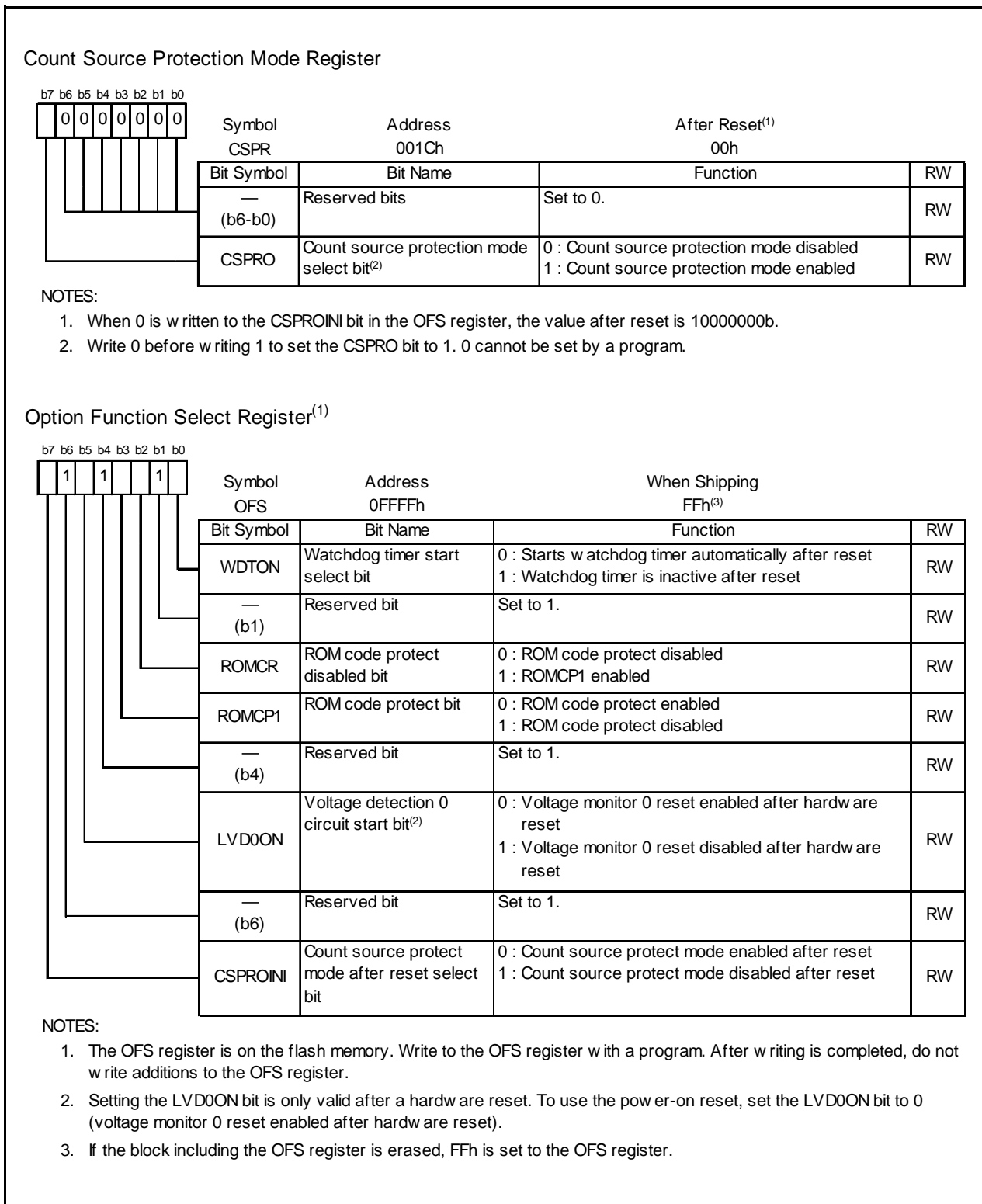


Figure 15.3 Registers CSPR and OFS

## 15.1 Count Source Protection Mode Disabled

The count source of the watchdog timer is the CPU clock when count source protection mode is disabled. Table 15.2 lists the Watchdog Timer Specifications (with Count Source Protection Mode Disabled).

**Table 15.2 Watchdog Timer Specifications (with Count Source Protection Mode Disabled)**

Item	Specification
Count source	CPU clock
Count operation	Decrement
Period	$\text{Division ratio of prescaler (n) } \times \text{ count value of watchdog timer (32768)}^{(1)}$ CPU clock n: 16 or 128 (selected by WDC7 bit in WDC register) Example: When the CPU clock frequency is 16 MHz and prescaler divided by 16, the period is approximately 32.8 ms
Reset condition of watchdog timer	<ul style="list-style-type: none"> <li>• Reset</li> <li>• Write 00h to the WDTR register before writing FFh</li> <li>• Underflow</li> </ul>
Count start condition	The WDTON bit <sup>(2)</sup> in the OFS register (0FFFFh) selects the operation of the watchdog timer after a reset <ul style="list-style-type: none"> <li>• When the WDTON bit is set to 1 (watchdog timer is in stop state after reset) The watchdog timer and prescaler stop after a reset and the count starts when the WDTS register is written to</li> <li>• When the WDTON bit is set to 0 (watchdog timer starts automatically after exiting) The watchdog timer and prescaler start counting automatically after a reset</li> </ul>
Count stop condition	Stop and wait modes (inherit the count from the held value after exiting modes)
Operation at time of underflow	<ul style="list-style-type: none"> <li>• When the PM12 bit in the PM1 register is set to 0 Watchdog timer interrupt</li> <li>• When the PM12 bit in the PM1 register is set to 1 Watchdog timer reset (Refer to <b>5.6 Watchdog Timer Reset.</b>)</li> </ul>

### NOTES:

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

## 15.2 Count Source Protection Mode Enabled

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

Table 15.3 lists the Watchdog Timer Specifications (with Count Source Protection Mode Enabled).

**Table 15.3 Watchdog Timer Specifications (with Count Source Protection Mode Enabled)**

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

### NOTES:

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



## 16. Timers

The MCU has two 8-bit timers with 8-bit prescalers and two 16-bit timers. The two 8-bit timers with 8-bit prescalers are timer RA and timer RB. These timers contain a reload register to store the default value of the counter. The two 16-bit timers are timer RC, timer RD, and have input capture and output compare functions. All the timers operate independently.

Tables 16.1 lists Functional Comparison of Timers.

**Table 16.1 Functional Comparison of Timers**

Item		Timer RA	Timer RB	Timer RC	Timer RD
Configuration		8-bit timer with 8-bit prescaler (with reload register)	8-bit timer with 8-bit prescaler (with reload register)	16-bit timer (with input capture and output compare)	16-bit timer × 2 (with input capture and output compare)
Count		Decrement	Decrement	Increment	Increment/Decrement
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>• TRCCLK</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>
Function	Count of the internal count source	Timer mode	Timer mode	Timer mode (output compare function)	Timer mode (output compare function)
	Count of the external count source	Event counter mode	—	Timer mode (output compare function)	Timer mode (output compare function)
	External pulse width/period measurement	Pulse width measurement mode, pulse period measurement mode	—	Timer mode (input capture function; 4 pins)	Timer mode (input capture function; 2 channels × 4 pins)
	PWM output	Pulse output mode <sup>(1)</sup> , Event counter mode <sup>(1)</sup>	Programmable waveform generation mode	Timer mode (output compare function; 4 pins) <sup>(1)</sup> , PWM mode (3 pins), PWM2 mode (1 pin)	Timer mode (output compare function; 2 channels × 4 pins) <sup>(1)</sup> , PWM mode (2 channels × 3 pins), PWM3 mode (2 channels × 2 pins)
	One-shot waveform output	—	Programmable one-shot generation mode, Programmable wait one-shot generation mode	PWM mode (3 pins)	PWM mode (2 channels × 3 pins)
	Three-phase waveforms output	—	—	—	Reset synchronous PWM mode (2 channels × 3 pins, Sawtooth wave modulation), Complementary PWM mode (2 channels × 3 pins, triangular wave modulation, dead time)
Input pin		TRAIO	INT0	INT0, TRCCLK, TRCTRG, TRCIOA, TRCIOB, TRCIOC, TRCIOD	INT0, TRDCLK, TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1
Output pin		TRAO TRAIO	TRBO	TRCIOA, TRCIOB, TRCIOC, TRCIOD	TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1
Related interrupt		<u>Timer RA</u> interrupt, INT1 interrupt	<u>Timer RB</u> interrupt, INT0 interrupt	Compare match/input capture A to D interrupt, <u>Overflow</u> interrupt, INT0 interrupt	Compare match/input capture A0 to D0 interrupt, Compare match/input capture A1 to D1 interrupt, <u>Overflow</u> interrupt, <u>Underflow</u> interrupt <sup>(2)</sup> , INT0 interrupt
Timer stop		Provided	Provided	Provided	Provided

## NOTES:

1. Rectangular waves are output in these modes. Since the waves are inverted at each overflow, the “H” and “L” level widths of the pulses are the same.
2. The underflow interrupt can be set to channel 1.

## 16.1 Timer RA

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

The prescaler and timer each consist of a reload register and counter. The reload register and counter are allocated at the same address, and can be accessed when accessing registers TRAPRE and TRA (refer to **Tables 16.2 to 16.6 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 16.1 shows a Block Diagram of Timer RA. Figures 16.2 and 16.3 show the registers associated with Timer RA.

Timer RA contains the following five operating modes:

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

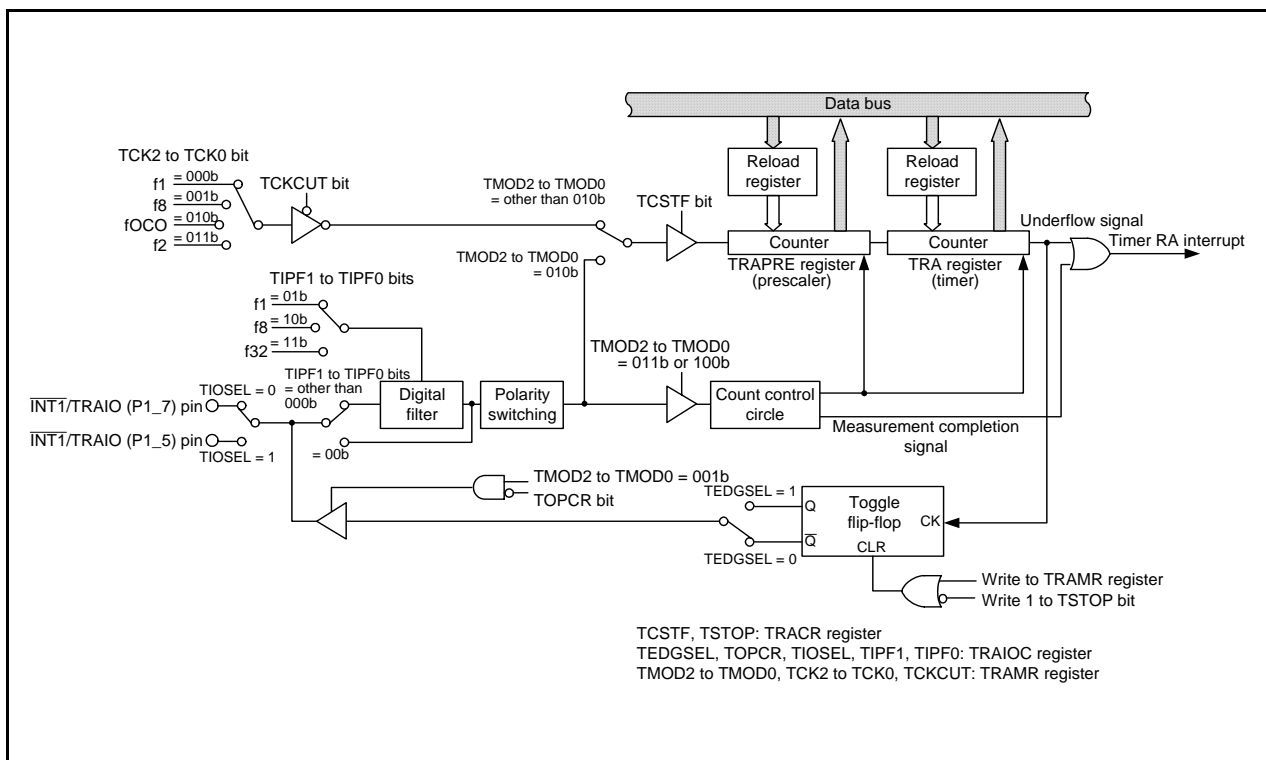


Figure 16.1 Block Diagram of Timer RA

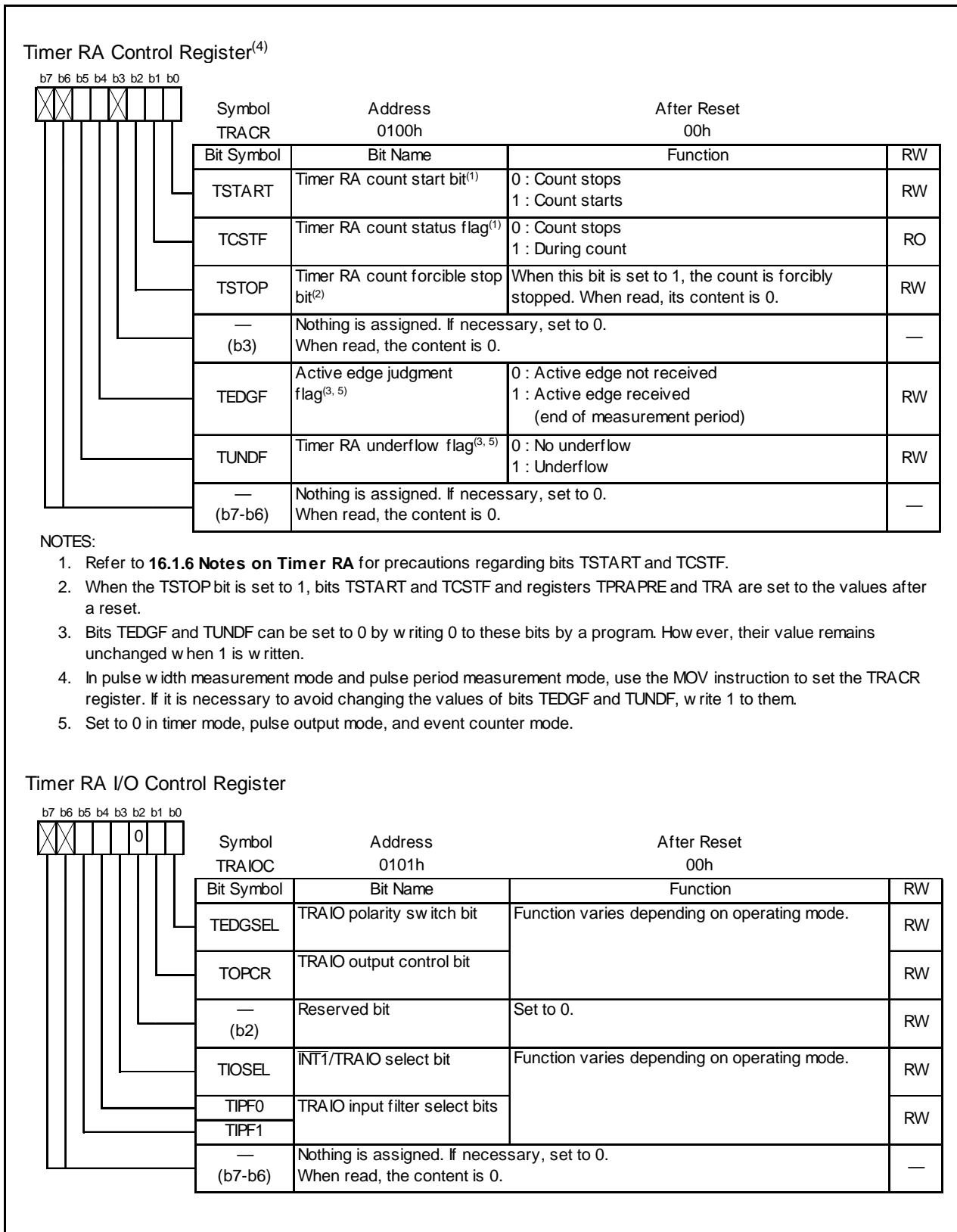


Figure 16.2 Registers TRACR and TRAIIOC

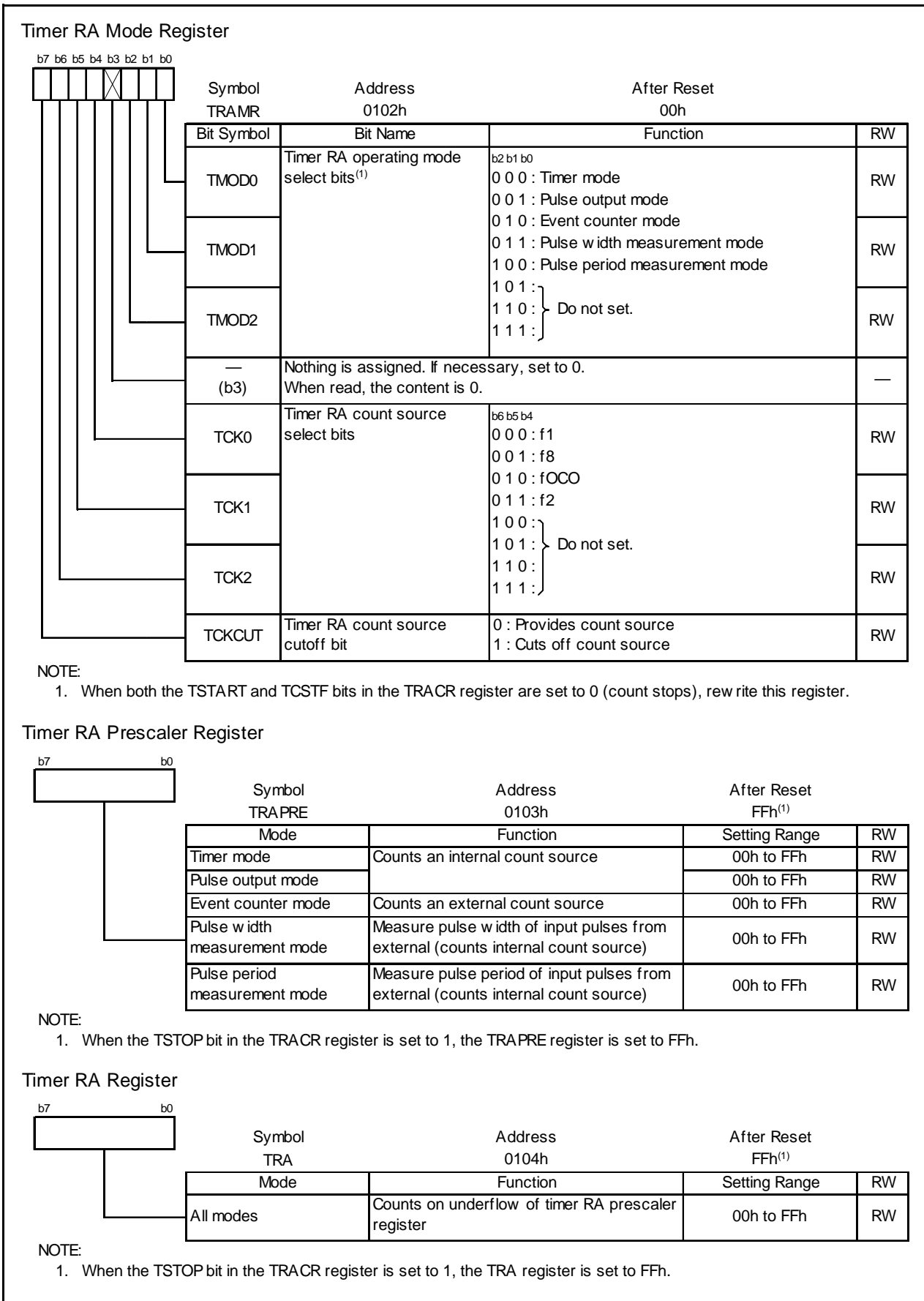


Figure 16.3 Registers TRAMR, TRAPRE, and TRA

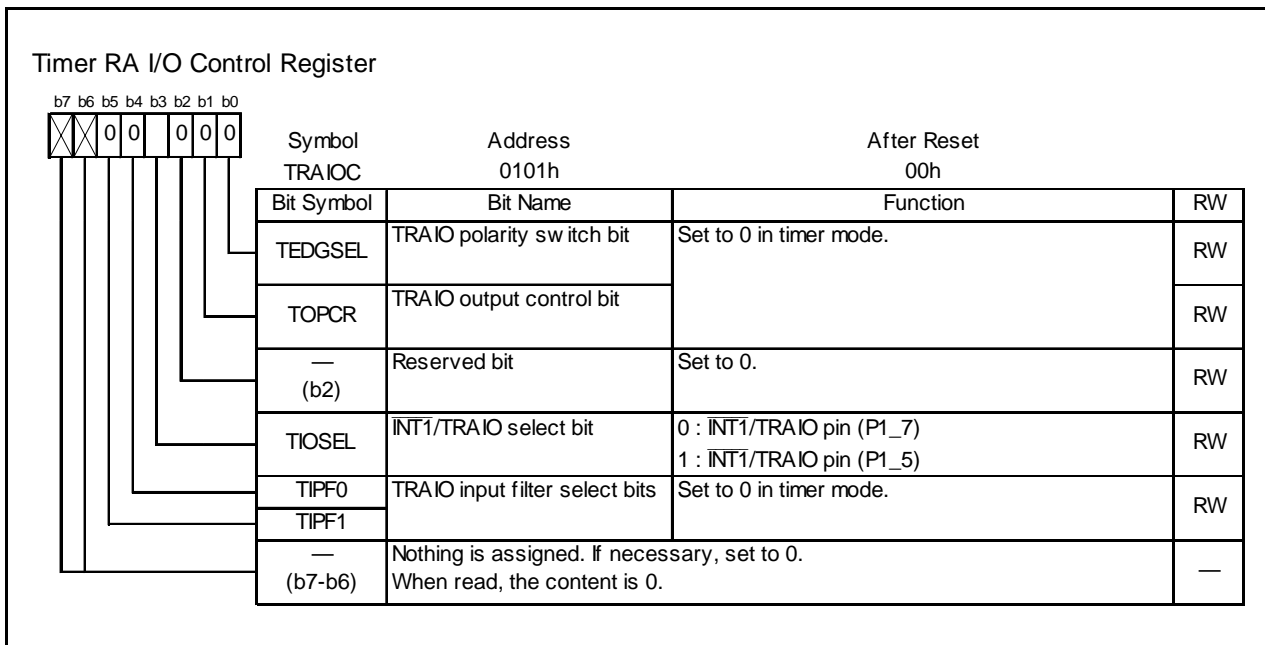
### 16.1.1 Timer Mode

In this mode, the timer counts an internally generated count source (refer to **Table 16.2 Timer Mode Specifications**).

Figure 16.4 shows the TRAIOC Register in Timer Mode.

**Table 16.2 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 of the reload register are reloaded and the count is continued.</li> </ul>
Divide ratio	$1/(n+1)(m+1)$ n: Value set in TRAPRE register, m: Value set in TRA register
Count start condition	1 (count starts) is written to the TSTART bit in the TRACR register.
Count stop conditions	<ul style="list-style-type: none"> <li>0 (count stops) is written to the TSTART bit in the TRACR register.</li> <li>1 (count forcibly stops) is written to the TSTOP bit in the TRACR register.</li> </ul>
Interrupt request generation timing	When timer RA underflows [timer RA interrupt].
INT1/TRAI0 pin function	Programmable I/O port, or INT1 interrupt input
Read from timer	The count value can be read by reading registers TRA and TRAPRE.
Write to timer	<ul style="list-style-type: none"> <li>When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter.</li> <li>When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter. (Refer to <b>16.1.1.1 Timer Write Control during Count Operation</b>.)</li> </ul>

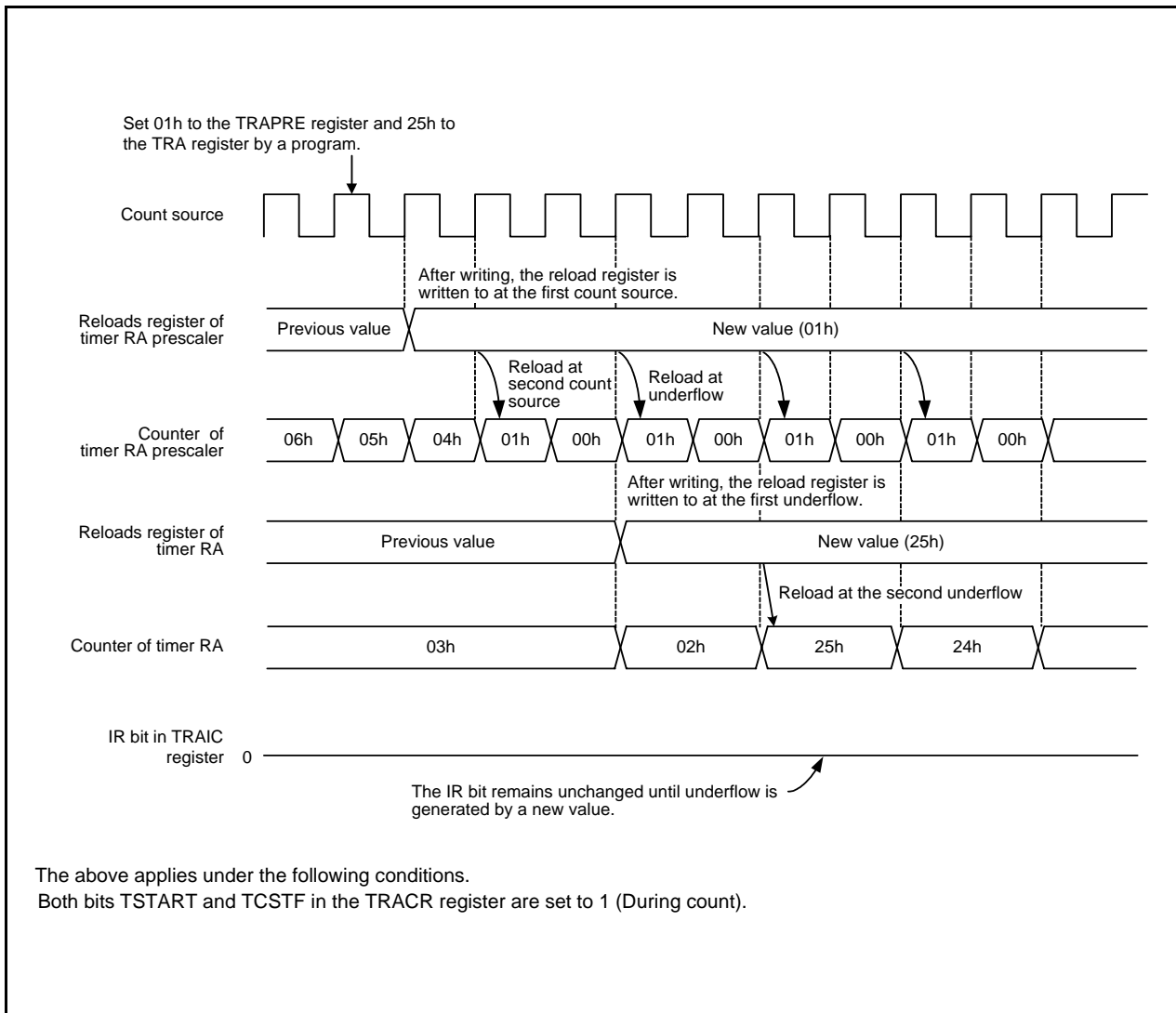


**Figure 16.4 TRAI0C Register in Timer Mode**

### 16.1.1.1 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 16.5 shows an Operating Example of Timer RA when Counter Value is Rewritten during Count Operation.



**Figure 16.5** Operating Example of Timer RA when Counter Value is Rewritten during Count Operation

### 16.1.2 Pulse Output Mode

In pulse output mode, the internally generated count source is counted, and a pulse with inverted polarity is output from the TRAI0 pin each time the timer underflows (refer to **Table 16.3 Pulse Output Mode Specifications**).

Figure 16.6 shows the TRAI0C Register in Pulse Output Mode.

**Table 16.3 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 continued.</li> </ul>
Divide ratio	$1/(n+1)(m+1)$ n: Value set in TRAPRE register, m: Value set in TRA register
Count start condition	1 (count starts) is written to the TSTART bit in the TRACR register.
Count stop conditions	<ul style="list-style-type: none"> <li>• 0 (count stops) is written to the TSTART bit in the TRACR register.</li> <li>• 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register.</li> </ul>
Interrupt request generation timing	When timer RA underflows [timer RA interrupt].
INT1/TRAI0 pin function	Pulse output, programmable output port, or $\overline{\text{INT1}}$ interrupt <sup>(1)</sup>
Read from timer	The count value can be read by reading registers TRA and TRAPRE.
Write to timer	<ul style="list-style-type: none"> <li>• When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter.</li> <li>• When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter. (Refer to <b>16.1.1.1 Timer Write Control during Count Operation</b>.)</li> </ul>
Select functions	<ul style="list-style-type: none"> <li>• TRAI0 signal polarity switch function The TEDGSEL bit in the TRAI0C register selects the level at the start of pulse output.<sup>(1)</sup></li> <li>• Pulse output stop function Output from the TRAI0 pin is stopped by the TOPCR bit in the TRAI0C register.</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 the output pulse becomes the level when the pulse output starts when the TRAMR register is written to.



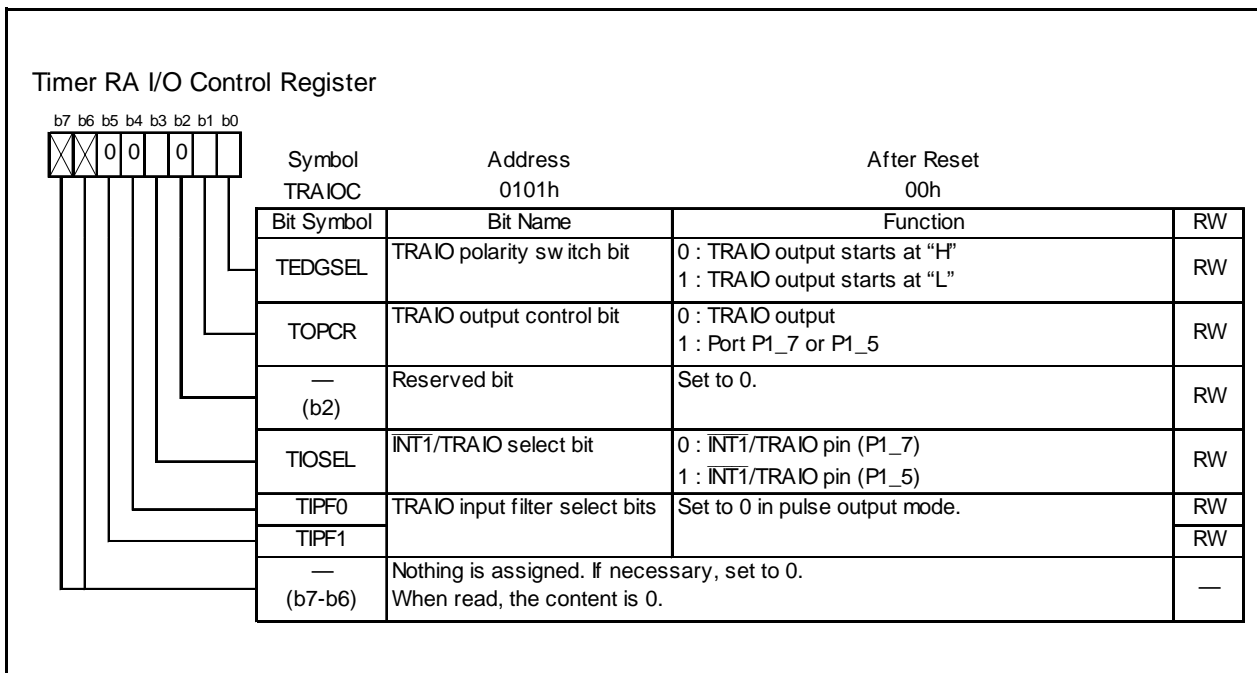


Figure 16.6 TRATIO Register in Pulse Output Mode

### 16.1.3 Event Counter Mode

In event counter mode, external signal inputs to the  $\overline{\text{INT1}}$ /TRAIO pin are counted (refer to **Table 16.4 Event Counter Mode Specifications**).

Figure 16.7 shows the TRAIOC Register in Event Counter Mode.

**Table 16.4 Event Counter Mode Specifications**

Item	Specification
Count source	External signal which is input to TRAIO pin (active edge selectable by a program)
Count operations	<ul style="list-style-type: none"> <li>• Decrement</li> <li>• When the timer underflows, the contents of the reload register are reloaded and the count is continued.</li> </ul>
Divide ratio	$1/(n+1)(m+1)$ n: setting value of TRAPRE register, m: setting value of TRA register
Count start condition	1 (count starts) is written to the TSTART bit in the TRACR register.
Count stop conditions	<ul style="list-style-type: none"> <li>• 0 (count stops) is written to the TSTART bit in the TRACR register.</li> <li>• 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register.</li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When timer RA underflows [timer RA interrupt].</li> </ul>
$\overline{\text{INT1}}$ /TRAIO pin function	Count source input ( $\overline{\text{INT1}}$ interrupt input)
Read from timer	The count value can be read by reading registers TRA and TRAPRE.
Write to timer	<ul style="list-style-type: none"> <li>• When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter.</li> <li>• When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter. (Refer to <b>16.1.1.1 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 TRAIOC register selects the active edge of the count source.</li> <li>• Count source input pin select function P1_7 or P1_5 is selected by the TIOSEL bit in the TRAIOC register.</li> <li>• Digital filter function Bits TIPF0 and TIPF1 in the TRAIOC register enable or disable the digital filter and select the sampling frequency.</li> </ul>

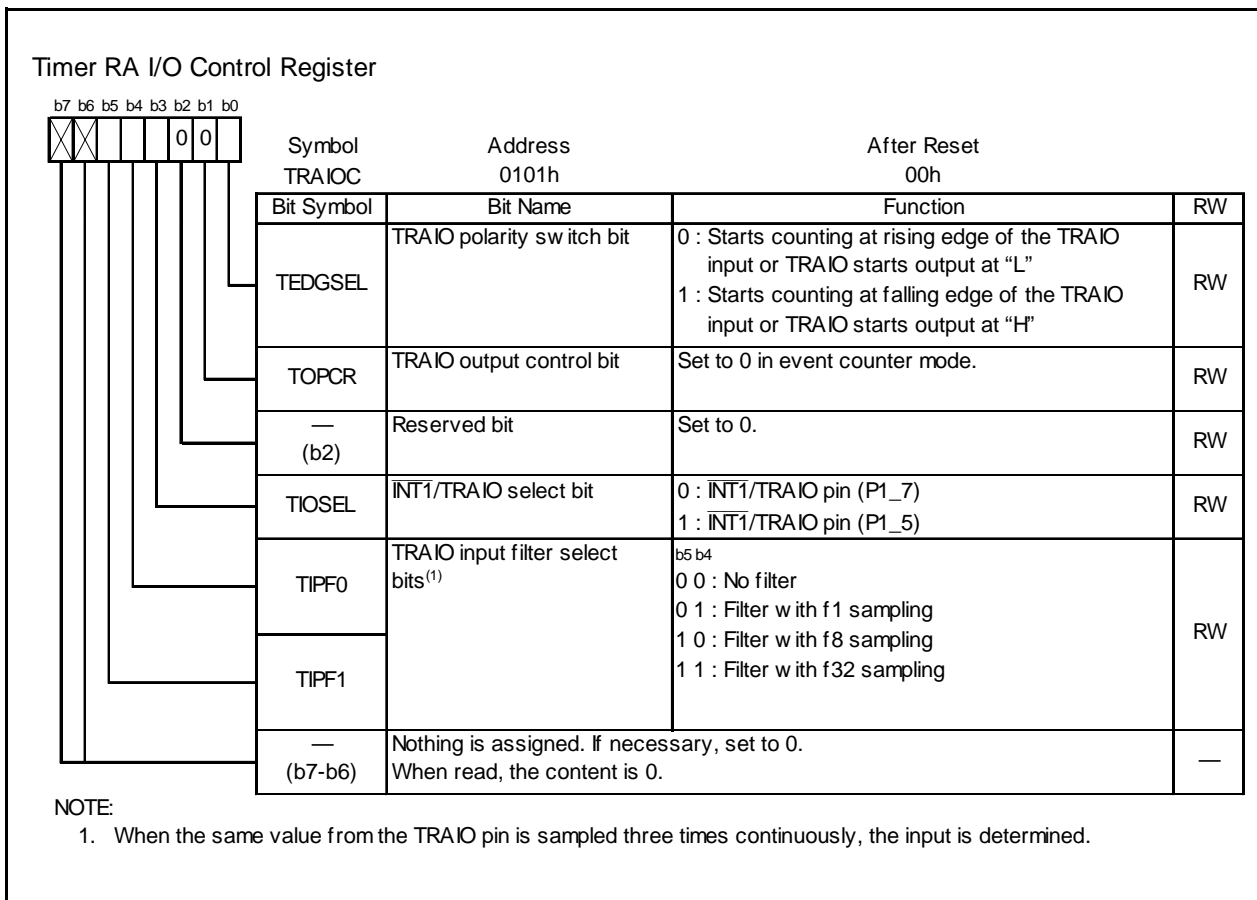


Figure 16.7 TRAI0C Register in Event Counter Mode

### 16.1.4 Pulse Width Measurement Mode

In pulse width measurement mode, the pulse width of an external signal input to the  $\overline{\text{INT1}}$ /TRAIO pin is measured (refer to **Table 16.5 Pulse Width Measurement Mode Specifications**).

Figure 16.8 shows the TRAIOC Register in Pulse Width Measurement Mode and Figure 16.9 shows an Operating Example of Pulse Width Measurement Mode.

**Table 16.5 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 of the reload register are reloaded and the count is continued.</li> </ul>
Count start condition	1 (count starts) is written to the TSTART bit in the TRACR register.
Count stop conditions	<ul style="list-style-type: none"> <li>• 0 (count stops) is written to the TSTART bit in the TRACR register.</li> <li>• 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register.</li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When timer RA underflows [timer RA interrupt].</li> <li>• Rising or falling of the TRAIO input (end of measurement period) [timer RA interrupt]</li> </ul>
$\overline{\text{INT1}}$ /TRAIO pin function	Measured pulse input ( $\overline{\text{INT1}}$ interrupt input)
Read from timer	The count value can be read by reading registers TRA and TRAPRE.
Write to timer	<ul style="list-style-type: none"> <li>• When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter.</li> <li>• When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter. (Refer to <b>16.1.1.1 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 selects the “H” or “L” level period.</li> <li>• Measured pulse input pin select function P1_7 or P1_5 is selected by the TIOSEL bit in the 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>

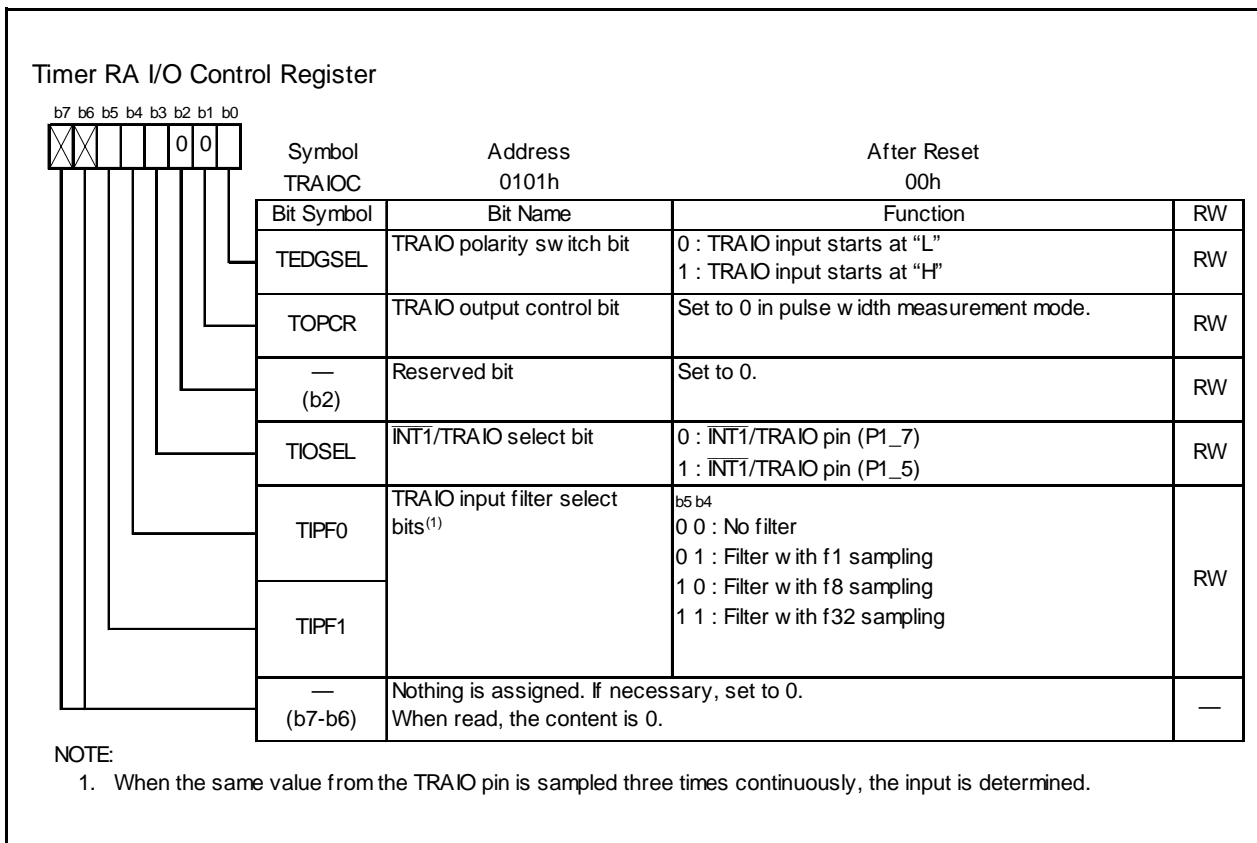


Figure 16.8 TRAI0C Register in Pulse Width Measurement Mode

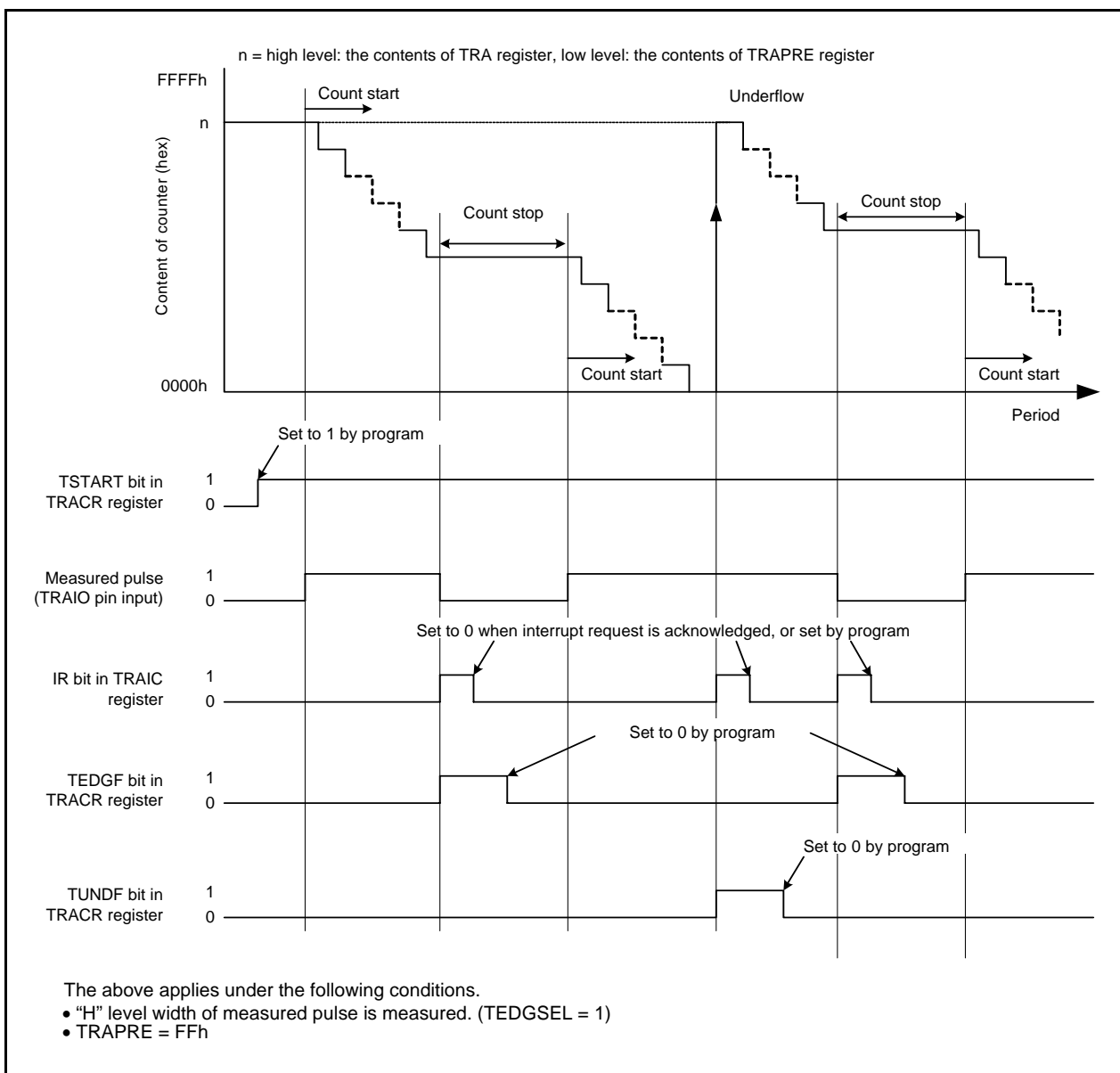


Figure 16.9 Operating Example of Pulse Width Measurement Mode

### 16.1.5 Pulse Period Measurement Mode

In pulse period measurement mode, the pulse period of an external signal input to the  $\overline{\text{INT1}}$ /TRAIO pin is measured (refer to **Table 16.6 Pulse Period Measurement Mode Specifications**).

Figure 16.10 shows the TRAIOC Register in Pulse Period Measurement Mode and Figure 16.11 shows an Operating Example of Pulse Period Measurement Mode.

**Table 16.6 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 the active edge of the measured pulse is input, the contents of the read-out buffer are retained at the first underflow of timer RA prescaler. Then timer RA reloads the contents in the reload register at the second underflow of timer RA prescaler and continues counting.</li> </ul>
Count start condition	1 (count start) is written to the TSTART bit in the TRACR register.
Count stop conditions	<ul style="list-style-type: none"> <li>• 0 (count stop) is written to TSTART bit in the TRACR register.</li> <li>• 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register.</li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When timer RA underflows or reloads [timer RA interrupt].</li> <li>• Rising or falling of the TRAIO input (end of measurement period) [timer RA interrupt]</li> </ul>
$\overline{\text{INT1}}$ /TRAIO pin function	Measured pulse input <sup>(1)</sup> ( $\overline{\text{INT1}}$ interrupt input)
Read from timer	The count value can be read by reading registers TRA and TRAPRE.
Write to timer	<ul style="list-style-type: none"> <li>• When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter.</li> <li>• When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter. (Refer to <b>16.1.1.1 Timer Write Control during Count Operation.</b>)</li> </ul>
Select functions	<ul style="list-style-type: none"> <li>• Measurement period select The TEDGSEL bit in the TRAIOC register selects the measurement period of the input pulse.</li> <li>• Measured pulse input pin select function P1_7 or P1_5 is selected by the TIOSEL bit in the TRAIOC register.</li> <li>• Digital filter function Bits TIPF0 and TIPF1 in the TRAIOC register enable or disable the digital filter and select the sampling frequency.</li> </ul>

**NOTE:**

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

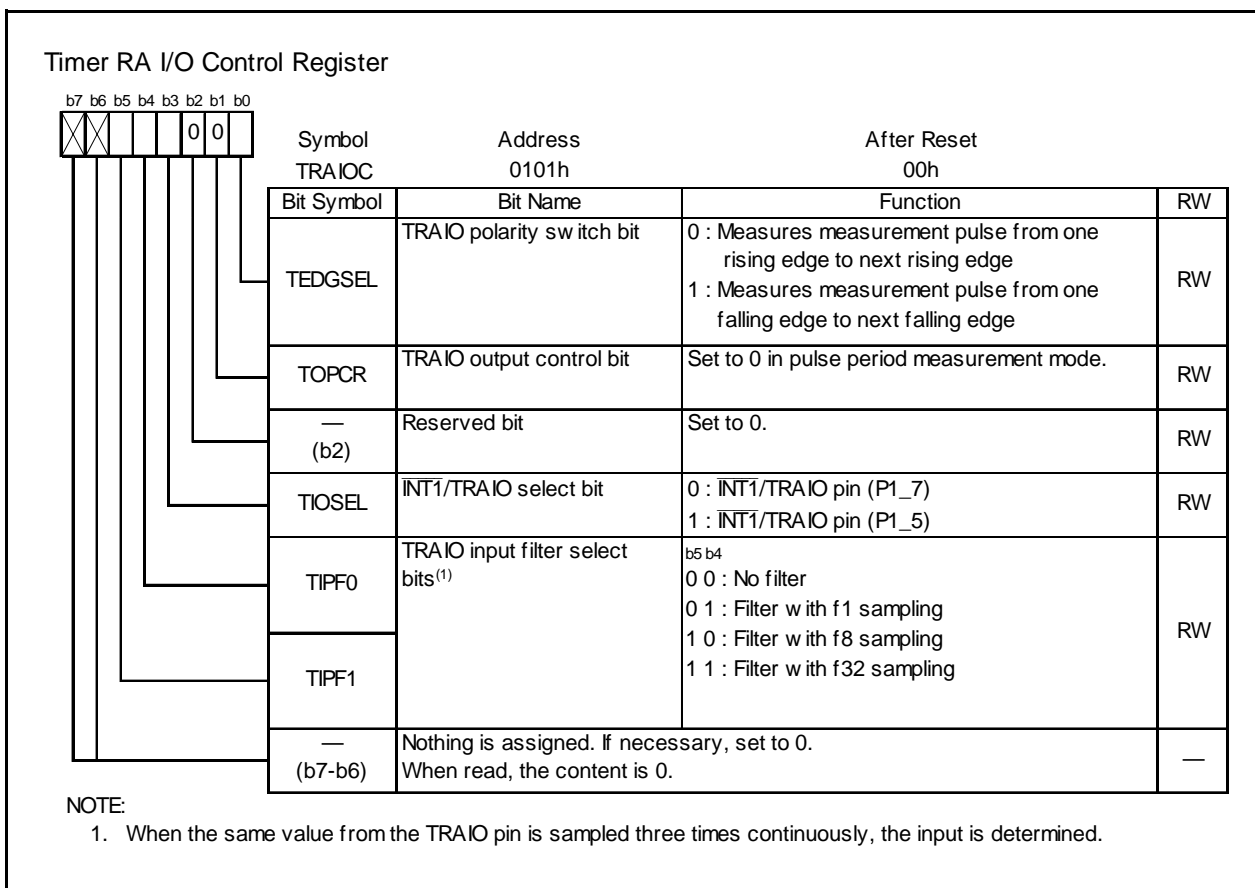
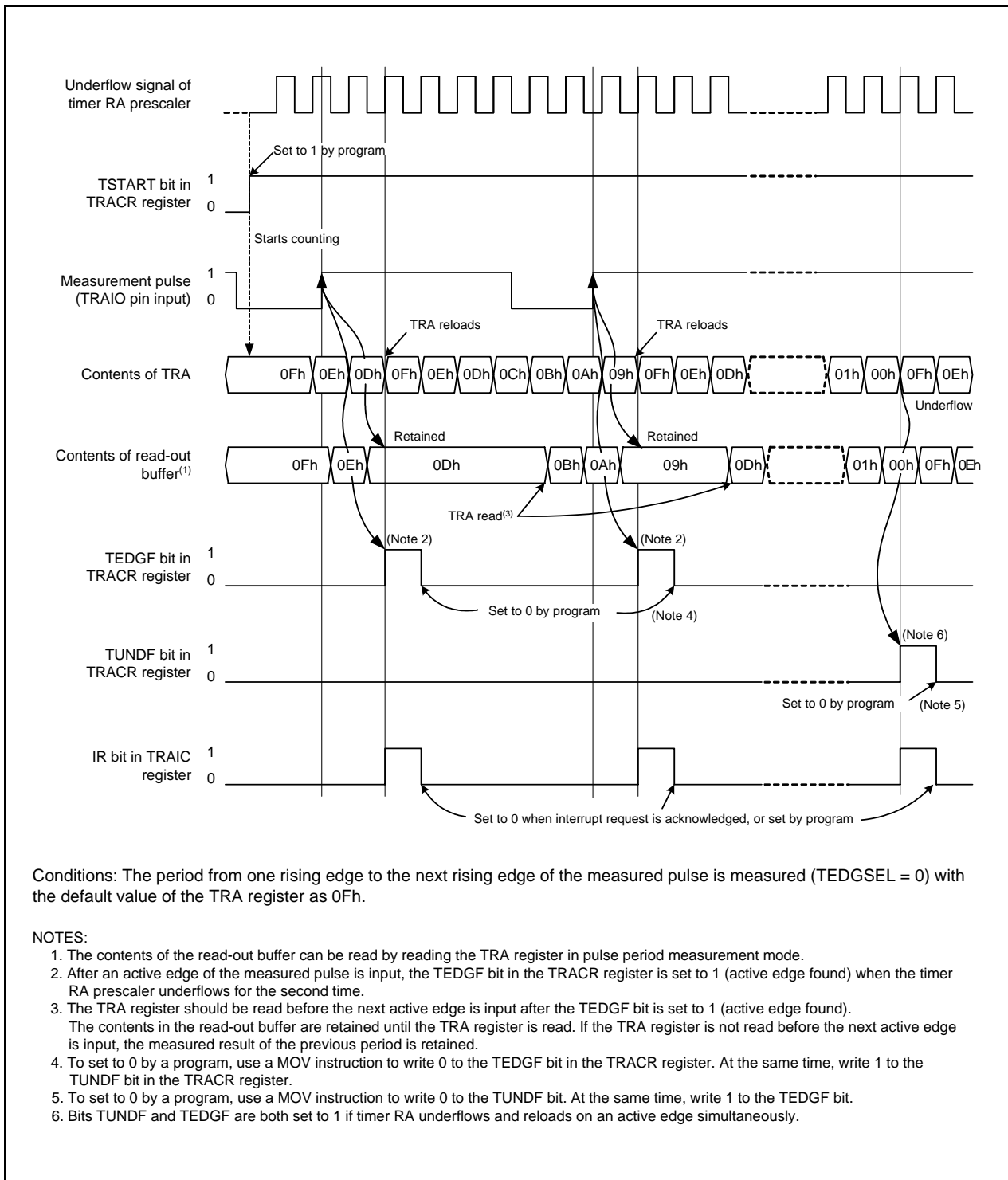


Figure 16.10 TRAIOC Register in Pulse Period Measurement Mode





**Figure 16.11 Operating Example of Pulse Period Measurement Mode**

### 16.1.6 Notes on Timer RA

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

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

NOTE:

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

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

## 16.2 Timer RB

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

The prescaler and timer each consist of a reload register and counter (refer to **Tables 16.7 to 16.10 the Specifications of Each Mode**).

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

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

Figure 16.12 shows a Block Diagram of Timer RB. Figures 16.13 to 16.16 show the registers associated with timer RB.

Timer RB has four operation modes listed as follows:

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

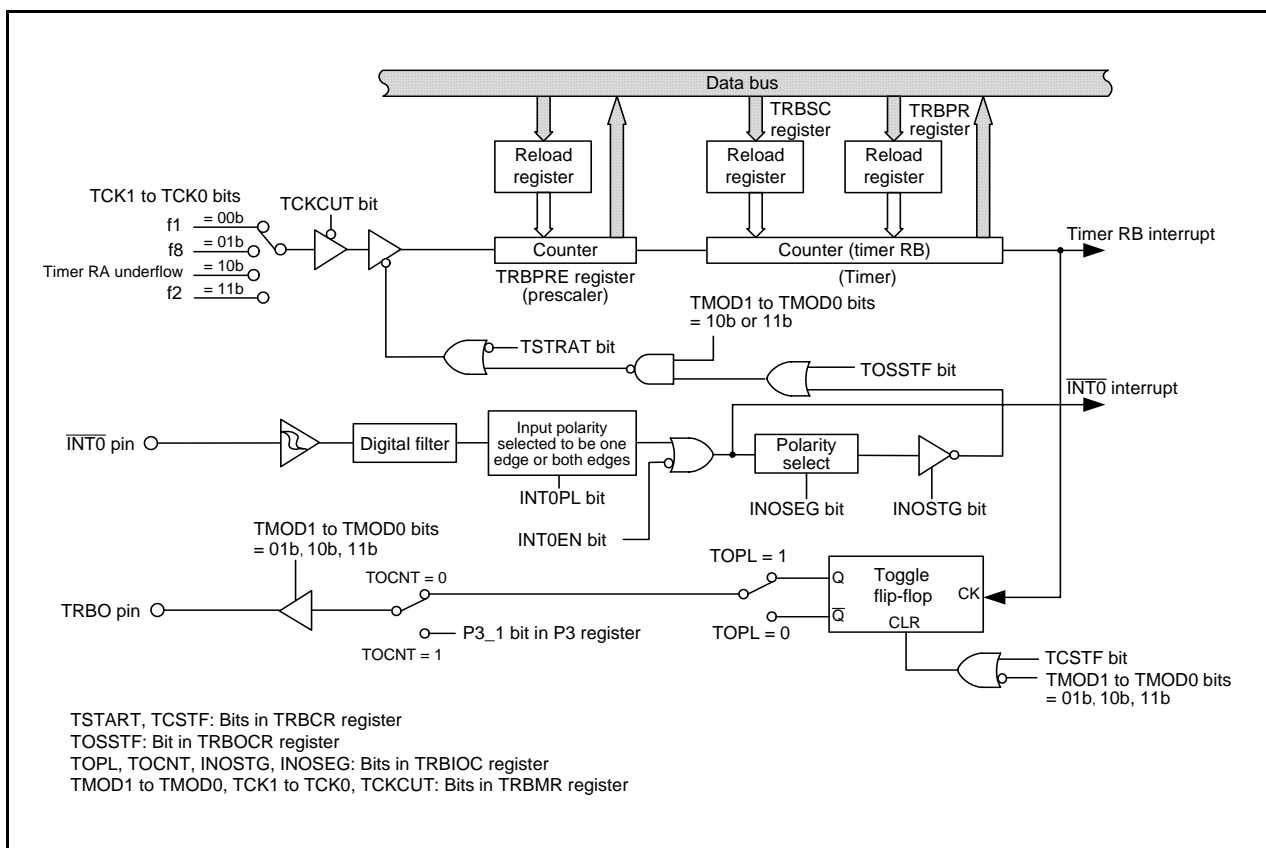


Figure 16.12 Block Diagram of Timer RB

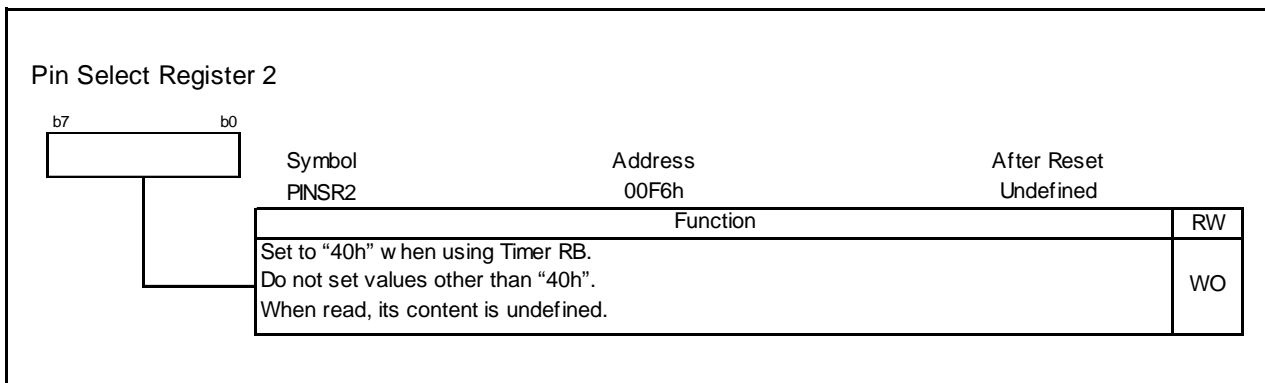


Figure 16.13 PINSR2 Register

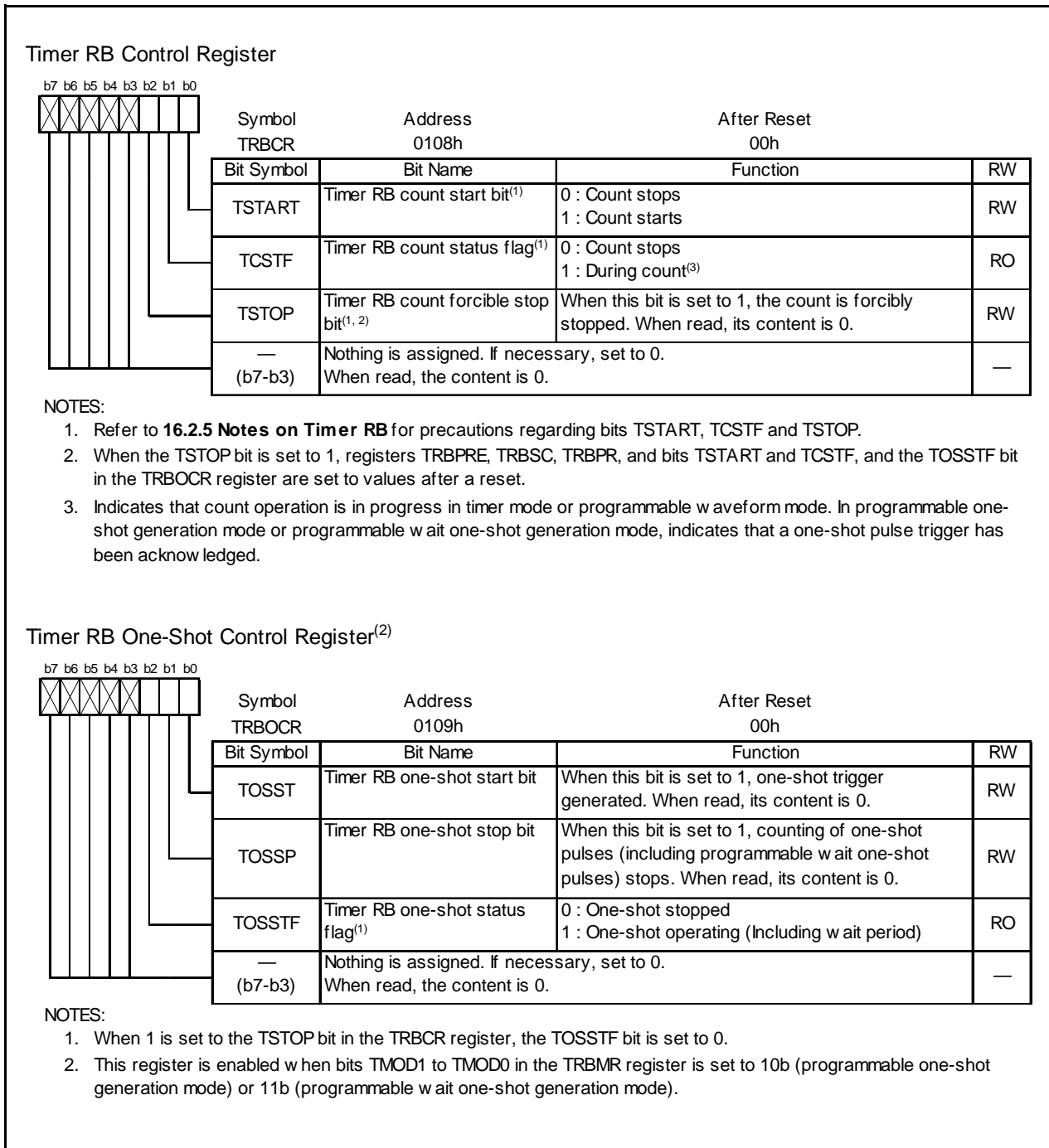


Figure 16.14 Registers TRBCR and TRBOCR

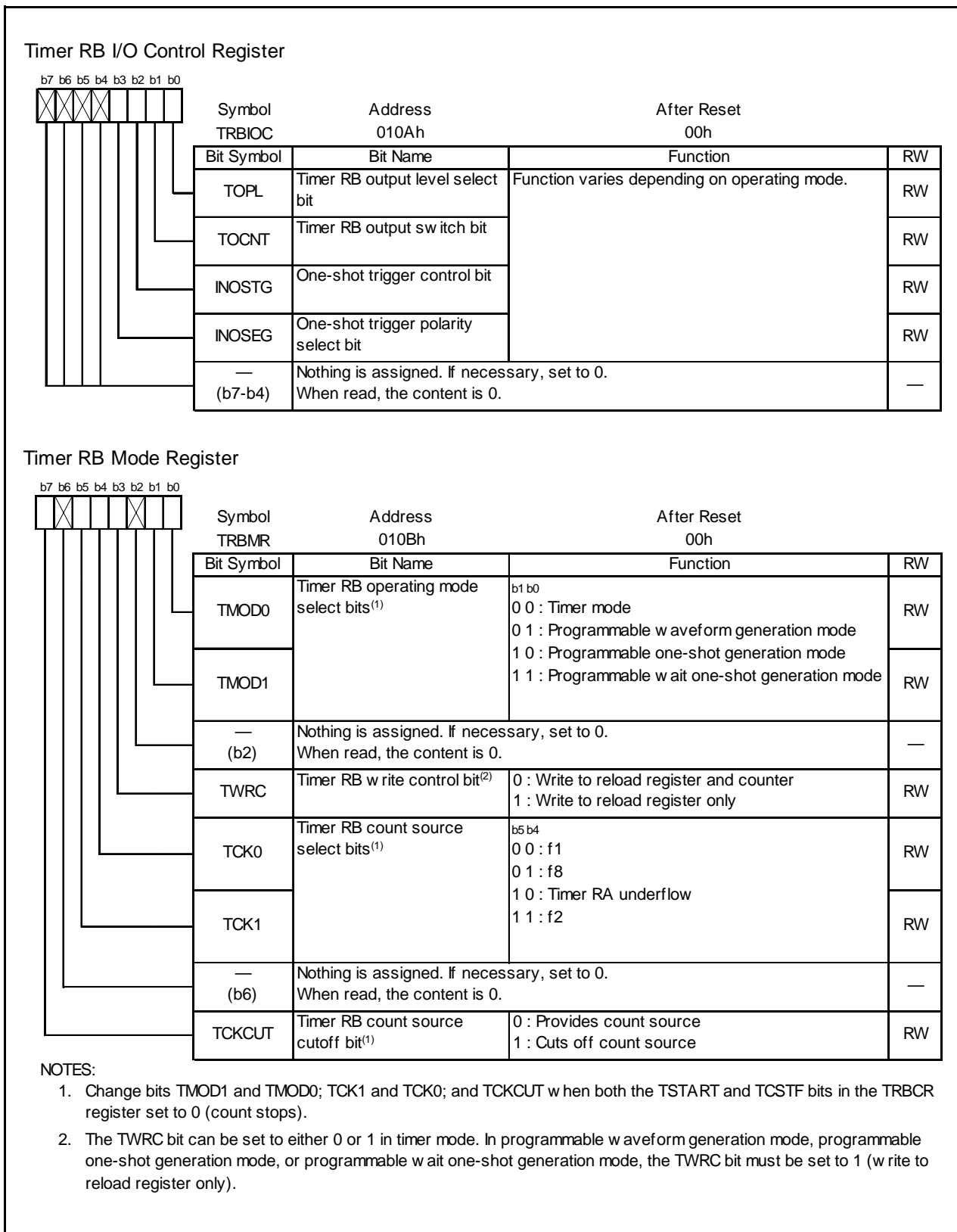


Figure 16.15 Registers TRBIOC and TRBMR

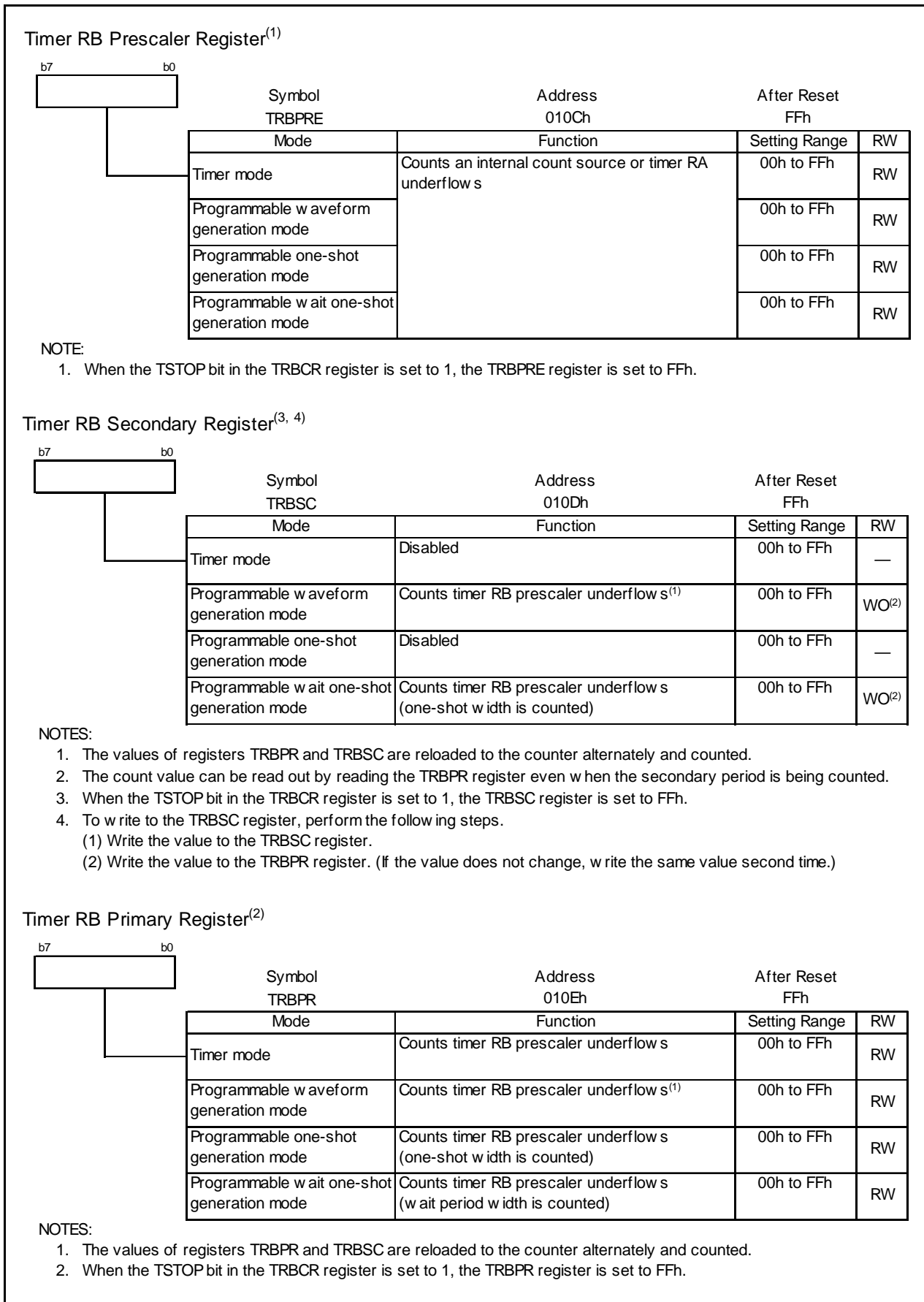


Figure 16.16 Registers TRBPRES, TRBSC, and TRBPR

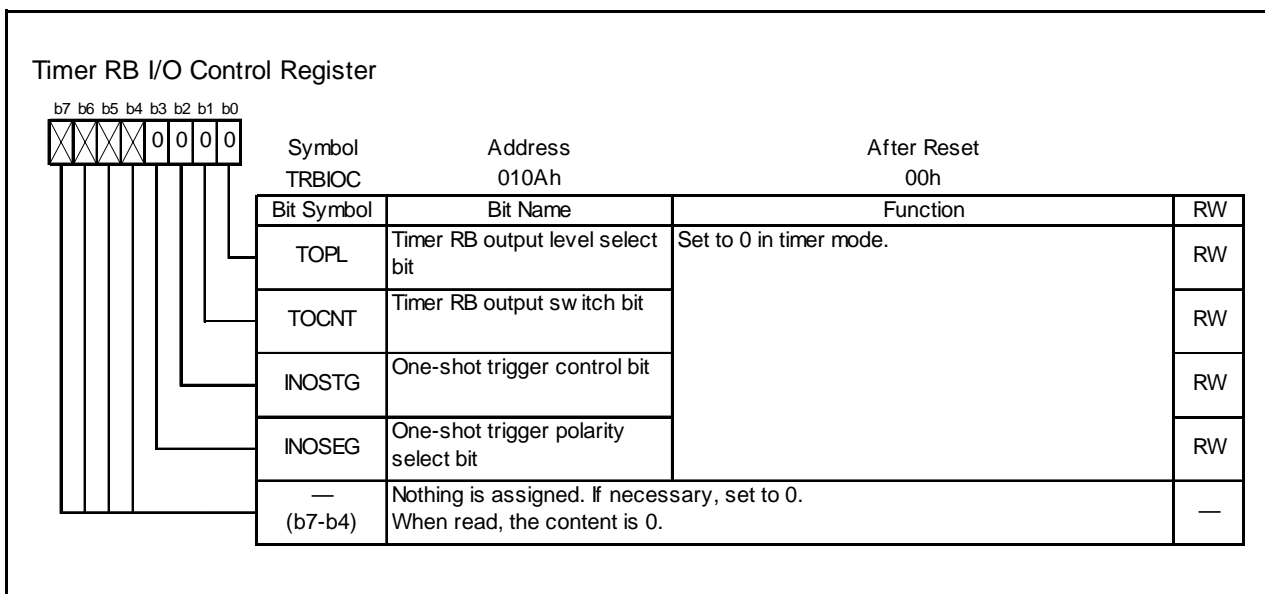
### 16.2.1 Timer Mode

In timer mode, a count source which is internally generated or timer RA underflows are counted (refer to **Table 16.7 Timer Mode Specifications**). Registers TRBOCR and TRBSC are not used in timer mode.

Figure 16.17 shows the TRBIOC Register in Timer Mode.

**Table 16.7 Timer Mode Specifications**

Item	Specification
Count sources	f1, f2, f8, timer RA underflow
Count operations	<ul style="list-style-type: none"> <li>Decrement</li> <li>When the timer underflows, it reloads the reload register contents before the count continues (when timer RB underflows, the contents of timer RB primary reload register is reloaded).</li> </ul>
Divide ratio	$1/(n+1)(m+1)$ n: setting value in TRBPRES register, m: setting value in TRBPR register
Count start condition	1 (count starts) is written to the TSTART bit in the TRBCR register.
Count stop conditions	<ul style="list-style-type: none"> <li>0 (count stops) is written to the TSTART bit in the TRBCR register.</li> <li>1 (count forcibly stop) is written to the TSTOP bit in the TRBCR register.</li> </ul>
Interrupt request generation timing	When timer RB underflows [timer RB interrupt].
TRBO pin function	Programmable I/O port
INT0 pin function	Programmable I/O port or $\overline{\text{INT0}}$ interrupt input
Read from timer	The count value can be read out by reading registers TRBPR and TRBPRES.
Write to timer	<ul style="list-style-type: none"> <li>When registers TRBPRES and TRBPR are written while the count is stopped, values are written to both the reload register and counter.</li> <li>When registers TRBPRES and TRBPR are written to while count operation is in progress:                If the TWRC bit in the TRBMR register is set to 0, the value is written to both the reload register and the counter.                If the TWRC bit is set to 1, the value is written to the reload register only.                (Refer to <b>16.2.1.1 Timer Write Control during Count Operation</b>.)</li> </ul>



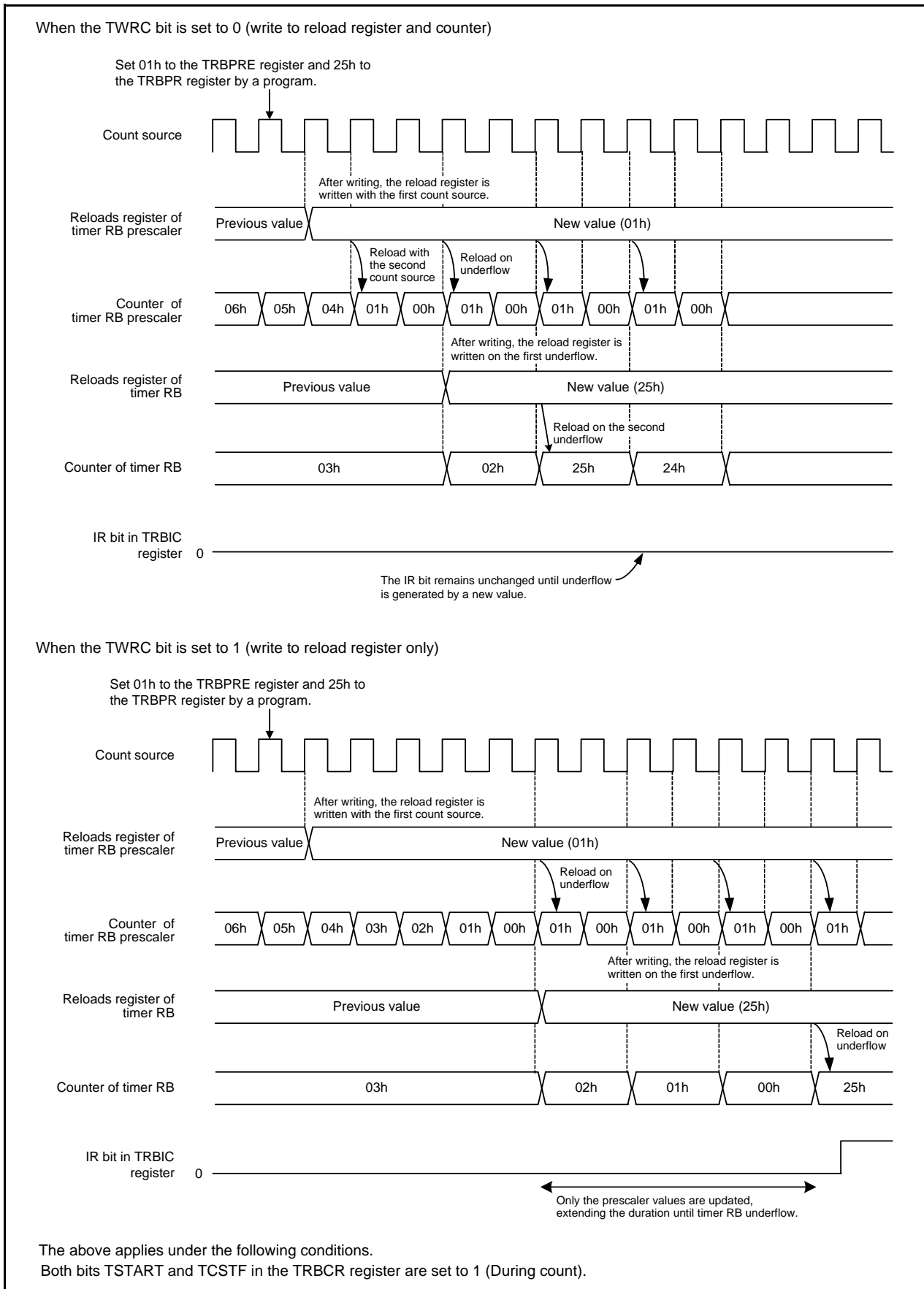
**Figure 16.17 TRBIOC Register in Timer Mode**



### 16.2.1.1 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 16.18 shows an Operating Example of Timer RB when Counter Value is Rewritten during Count Operation.



**Figure 16.18 Operating Example of Timer RB when Counter Value is Rewritten during Count Operation**

## 16.2.2 Programmable Waveform Generation Mode

In programmable waveform generation mode, the signal output from the TRBO pin is inverted each time the counter underflows, while the values in registers TRBPR and TRBSC are counted alternately (refer to **Table 16.8 Programmable Waveform Generation Mode Specifications**). Counting starts by counting the setting value in the TRBPR register. The TRBOCR register is unused in this mode.

Figure 16.19 shows the TRBIOC Register in Programmable Waveform Generation Mode. Figure 16.20 shows an Operating Example of Timer RB in Programmable Waveform Generation Mode.

**Table 16.8 Programmable Waveform Generation Mode Specifications**

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

### NOTES:

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

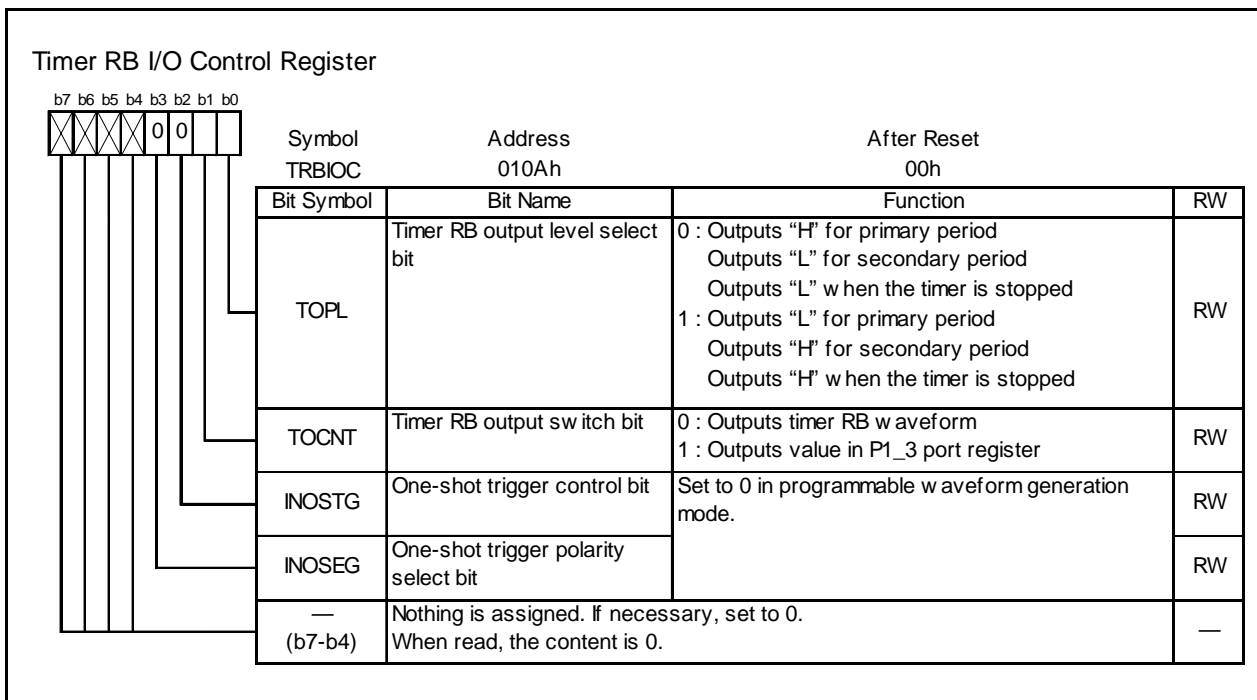


Figure 16.19 TRBIOC Register in Programmable Waveform Generation Mode

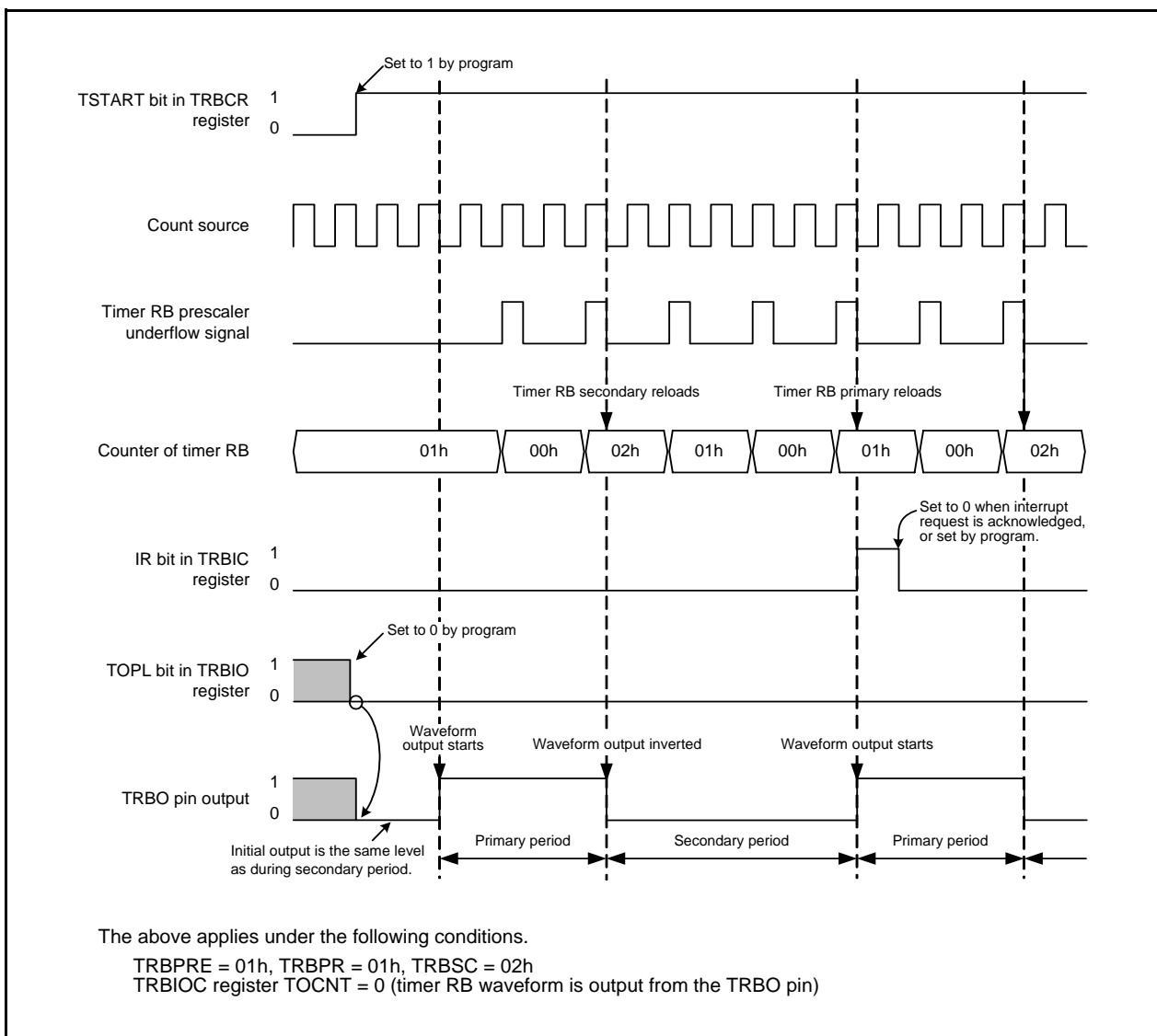


Figure 16.20 Operating Example of Timer RB in Programmable Waveform Generation Mode

### 16.2.3 Programmable One-shot Generation Mode

In programmable one-shot generation mode, a one-shot pulse is output from the TRBO pin by a program or an external trigger input (input to the INT0 pin) (refer to **Table 16.9 Programmable One-Shot Generation Mode Specifications**). When a trigger is generated, the timer starts operating from the point only once for a given period equal to the set value in the TRBPR register. The TRBSC register is not used in this mode.

Figure 16.21 shows the TRBIOC Register in Programmable One-Shot Generation Mode. Figure 16.22 shows an Operating Example of Programmable One-Shot Generation Mode.

**Table 16.9 Programmable One-Shot Generation Mode Specifications**

Item	Specification
Count sources	f1, f2, f8, timer RA underflow
Count operations	<ul style="list-style-type: none"> <li>• Decrement the setting value in the TRBPR register</li> <li>• When the timer underflows, it reloads the contents of the reload register before the count completes and the TOSSTF bit is set to 0 (one-shot stops).</li> <li>• When the count stops, the timer reloads the contents of the reload register before it stops.</li> </ul>
One-shot pulse output time	$(n+1)(m+1)/f_i$ $f_i$ : Count source frequency, $n$ : Setting value in TRBPRES register, $m$ : Setting value in TRBPR register <sup>(2)</sup>
Count start conditions	<ul style="list-style-type: none"> <li>• The TSTART bit in the TRBCR register is set to 1 (count starts) and the next trigger is generated</li> <li>• Set the TOSST bit in the TRBOCR register to 1 (one-shot starts)</li> <li>• Input trigger to the INT0 pin</li> </ul>
Count stop conditions	<ul style="list-style-type: none"> <li>• When reloading completes after timer RB underflows during primary period</li> <li>• When the TOSSP bit in the TRBOCR register is set to 1 (one-shot stops)</li> <li>• When the TSTART bit in the TRBCR register is set to 0 (stops counting)</li> <li>• When the TSTOP bit in the TRBCR register is set to 1 (forcibly stops counting)</li> </ul>
Interrupt request generation timing	In half a cycle of the count source, after the timer underflows (at the same time as the TRBO output ends) [timer RB interrupt]
TRBP 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 (INT0 one-shot trigger disabled): programmable I/O port or INT0 interrupt input</li> <li>• When the INOSTG bit in the TRBIOC register is set to 1 (INT0 one-shot trigger enabled): external trigger (INT0 interrupt input)</li> </ul>
Read from timer	The count value can be read out by reading registers TRBPR and TRBPRES.
Write to timer	<ul style="list-style-type: none"> <li>• When registers TRBPRES and TRBPR are written while the count is stopped, values are written to both the reload register and counter.</li> <li>• When registers TRBPRES and TRBPR are written during the count, values are written to the reload register only (the data is transferred to the counter at the following reload)<sup>(1)</sup>.</li> </ul>
Select functions	<ul style="list-style-type: none"> <li>• Output level select function The TOPL bit in the TRBIOC register selects the output level of the one-shot pulse waveform.</li> <li>• One-shot trigger select function Refer to <b>16.2.3.1 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.

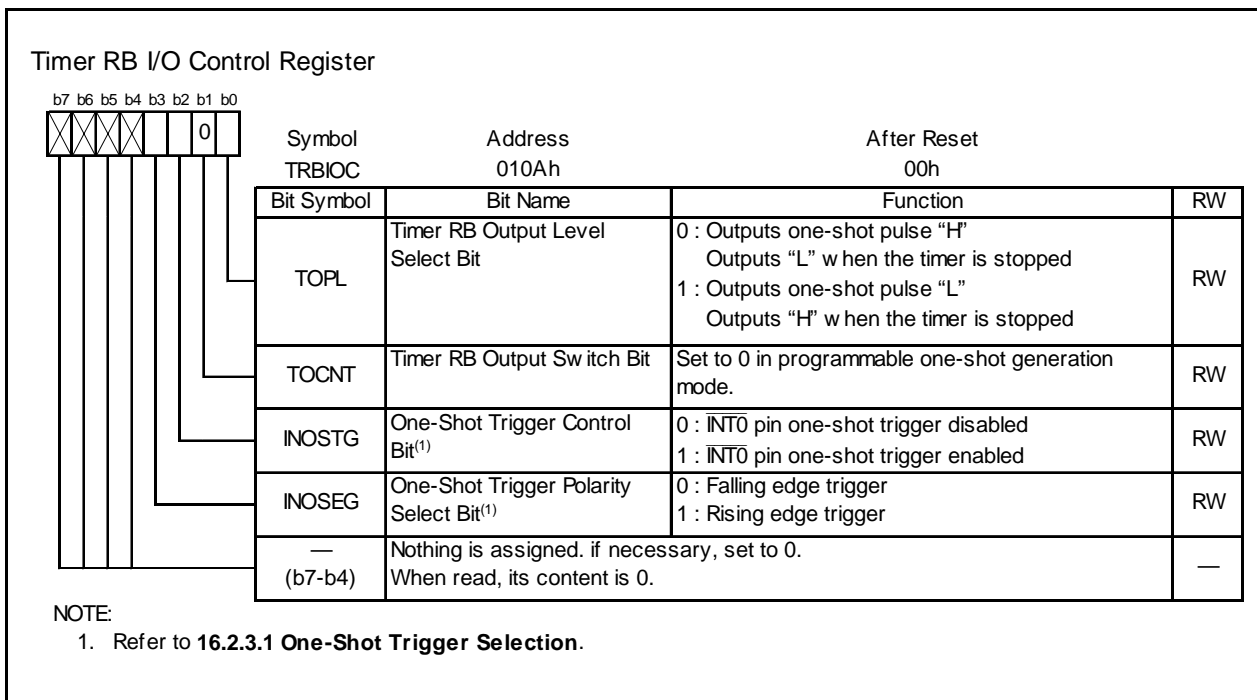


Figure 16.21 TRBIOC Register in Programmable One-Shot Generation Mode

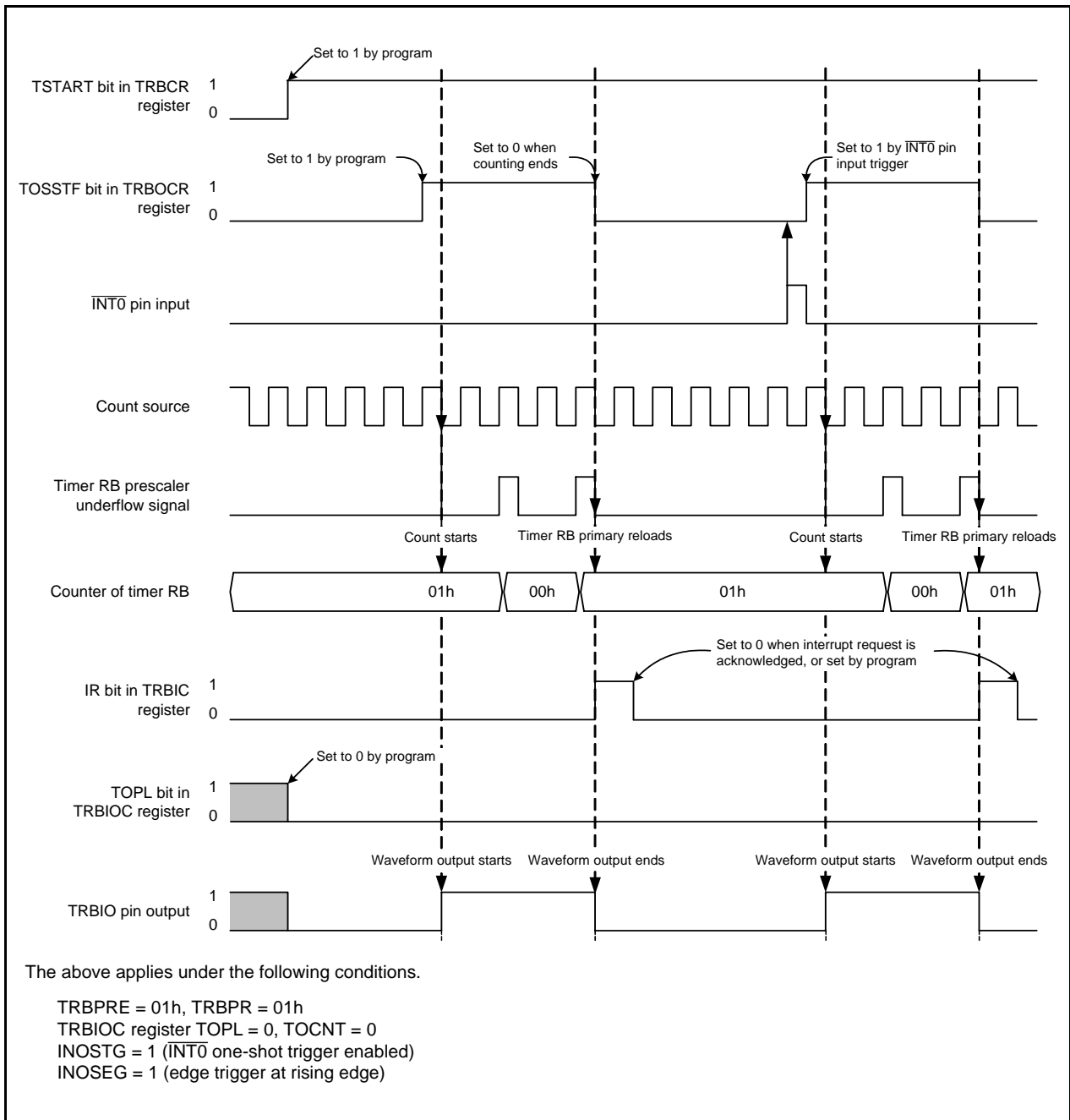


Figure 16.22 Operating Example of Programmable One-Shot Generation Mode



### 16.2.3.1 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 INTOIC 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.

### 16.2.4 Programmable Wait One-Shot Generation Mode

In programmable wait one-shot generation mode, a one-shot pulse is output from the TRBO pin by a program or an external trigger input (input to the  $\overline{\text{INT0}}$  pin) (refer to **Table 16.10 Programmable Wait One-Shot Generation Mode Specifications**). When a trigger is generated from that point, the timer outputs a pulse only once for a given length of time equal to the setting value in the TRBSC register after waiting for a given length of time equal to the setting value in the TRBPR register.

Figure 16.23 shows the TRBIOC Register in Programmable Wait One-Shot Generation Mode. Figure 16.24 shows an Operating Example of Programmable Wait One-Shot Generation Mode.

**Table 16.10 Programmable Wait One-Shot Generation Mode Specifications**

Item	Specification
Count sources	f1, f2, f8, timer RA underflow
Count operations	<ul style="list-style-type: none"> <li>Decrement the timer RB primary setting value.</li> <li>When a count of the timer RB primary underflows, the timer reloads the contents of timer RB secondary before the count continues.</li> <li>When a count of the timer RB secondary underflows, the timer reloads the contents of timer RB primary before the count completes and the TOSSTF bit is set to 0 (one-shot stops).</li> <li>When the count stops, the timer reloads the contents of the reload register before it stops.</li> </ul>
Wait time	$(n+1)(m+1)/f_i$ $f_i$ : Count source frequency $n$ : Value set in the TRBPRES register, $m$ : Value set in the TRBPR register <sup>(2)</sup>
One-shot pulse output time	$(n+1)(p+1)/f_i$ $f_i$ : Count source frequency $n$ : Value set in the TRBPRES register, $p$ : Value set in the TRBSC register
Count start conditions	<ul style="list-style-type: none"> <li>The TSTART bit in the TRBCR register is set to 1 (count starts) and the next trigger is generated.</li> <li>Set the TOSST bit in the TRBOCR register to 1 (one-shot starts).</li> <li>Input trigger to the <math>\overline{\text{INT0}}</math> pin</li> </ul>
Count stop conditions	<ul style="list-style-type: none"> <li>When reloading completes after timer RB underflows during secondary period.</li> <li>When the TOSSP bit in the TRBOCR register is set to 1 (one-shot stops).</li> <li>When the TSTART bit in the TRBCR register is set to 0 (starts counting).</li> <li>When the TSTOP bit in the TRBCR register is set to 1 (forcibly stops counting).</li> </ul>
Interrupt request generation timing	In half a cycle of the count source after timer RB underflows during secondary period (complete at the same time as waveform output from the TRBO pin) [timer RB interrupt].
TRBO pin function	Pulse output
$\overline{\text{INT0}}$ pin functions	<ul style="list-style-type: none"> <li>When the INOSTG bit in the TRBIOC register is set to 0 (<math>\overline{\text{INT0}}</math> one-shot trigger disabled): programmable I/O port or <math>\overline{\text{INT0}}</math> interrupt input</li> <li>When the INOSTG bit in the TRBIOC register is set to 1 (<math>\overline{\text{INT0}}</math> one-shot trigger enabled): external trigger (<math>\overline{\text{INT0}}</math> interrupt input)</li> </ul>
Read from timer	The count value can be read out by reading registers TRBPR and TRBPRES.
Write to timer	<ul style="list-style-type: none"> <li>When registers TRBPRES, TRBSC, and TRBPR are written while the count stops, values are written to both the reload register and counter.</li> <li>When registers TRBPRES, TRBSC, and TRBPR are written to during count operation, values are written to the reload registers only.<sup>(1)</sup></li> </ul>
Select functions	<ul style="list-style-type: none"> <li>Output level select function The TOPL bit in the TRBIOC register selects the output level of the one-shot pulse waveform.</li> <li>One-shot trigger select function Refer to <b>16.2.3.1 One-Shot Trigger Selection</b>.</li> </ul>

**NOTES:**

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

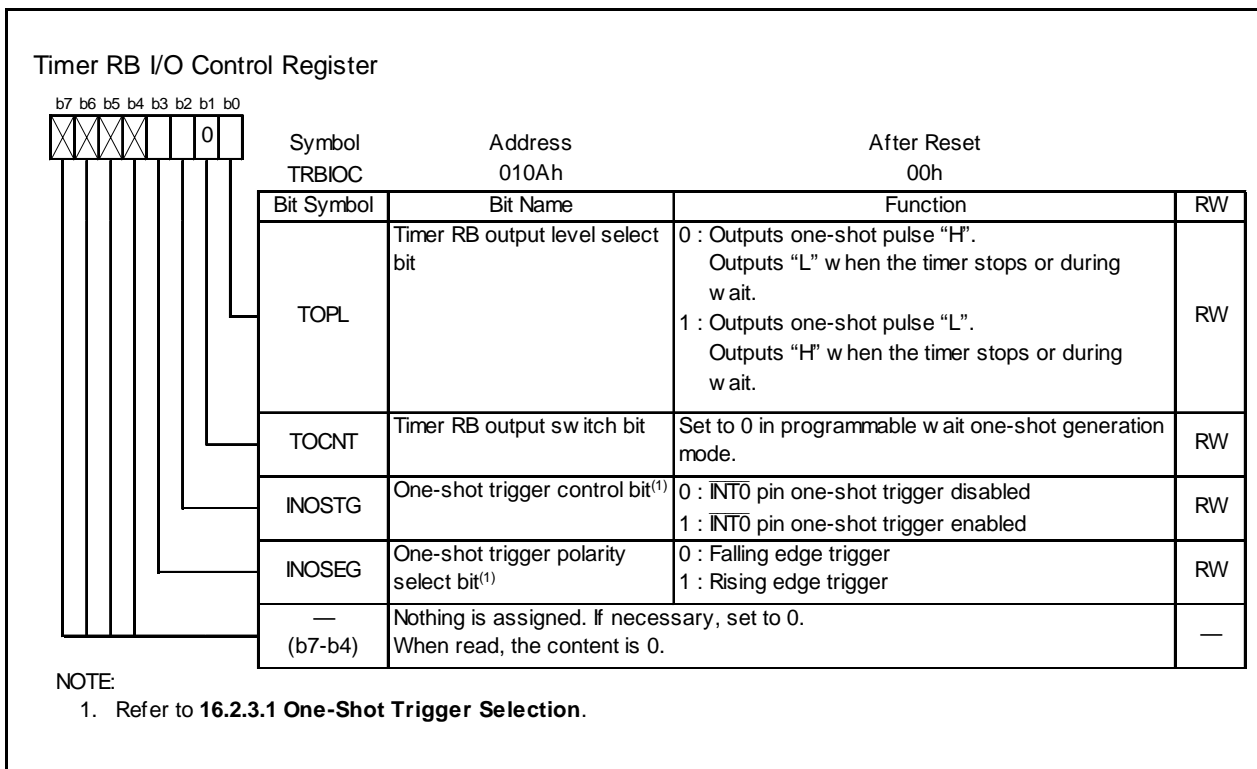


Figure 16.23 TRBIOC Register in Programmable Wait One-Shot Generation Mode

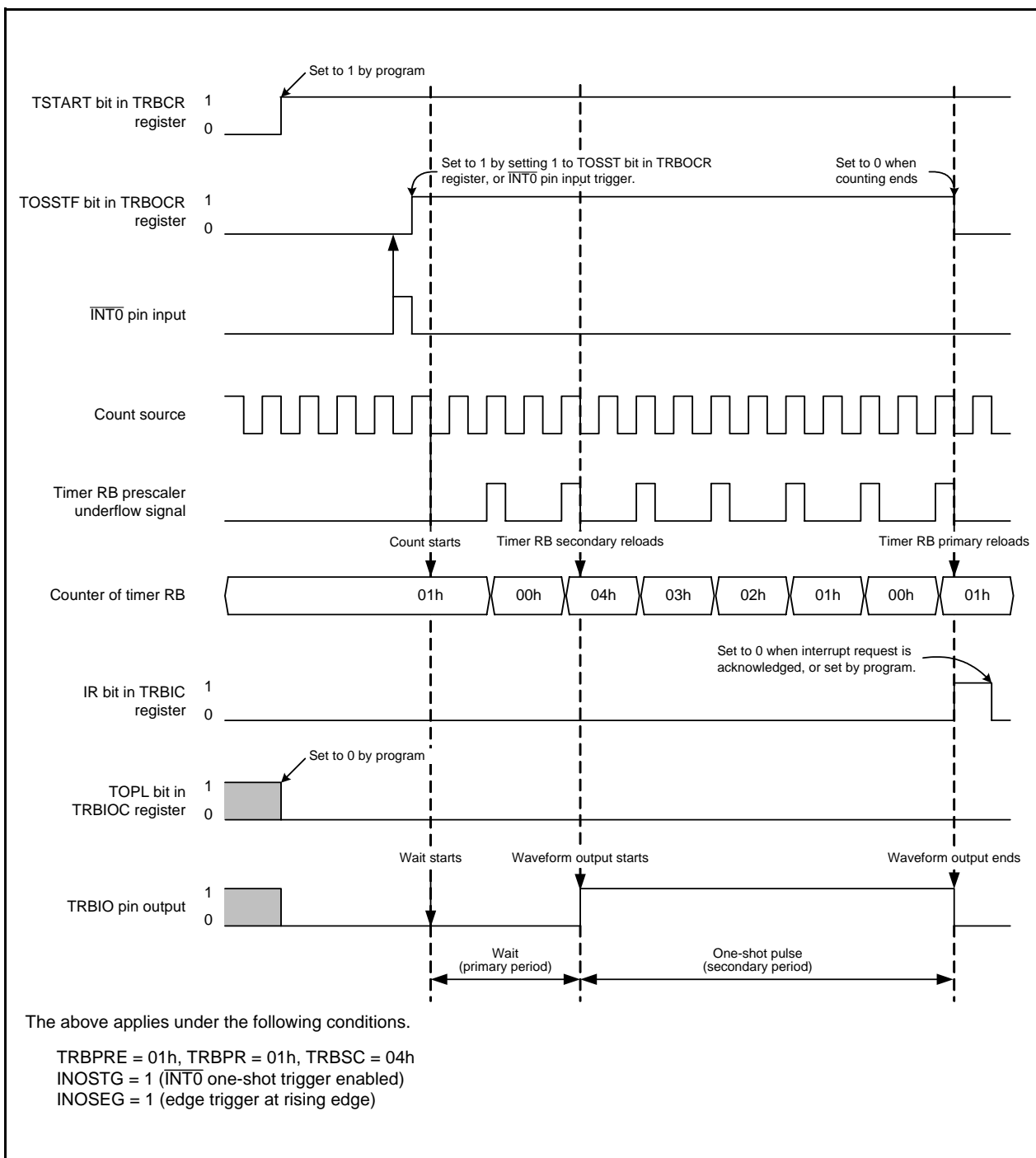


Figure 16.24 Operating Example of Programmable Wait One-Shot Generation Mode

## 16.2.5 Notes on Timer RB

- Timer RB stops counting after a reset. Set the values in the timer RB and timer RB prescalers before the count starts.
- Even if the prescaler and timer RB is read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- In programmable one-shot generation mode and programmable wait one-shot generation mode, when setting the TSTART bit in the TRBCR register to 0, 0 (stops counting) or setting the TOSSP bit in the TRBOCR register to 1 (stops one-shot), the timer reloads the value of reload register and stops. Therefore, in programmable one-shot generation mode and programmable wait one-shot generation mode, read the timer count value before the timer stops.
- The TCSTF bit remains 0 (count stops) for 1 to 2 cycles of the count source after setting the TSTART bit to 1 (count starts) while the count is stopped.  
During this time, do not access registers associated with timer RB<sup>(1)</sup> other than the TCSTF bit. Timer RB starts counting at the first valid edge of the count source after the TCSTF bit is set to 1 (during count). The TCSTF bit remains 1 for 1 to 2 cycles of the count source after setting the TSTART bit to 0 (count stops) while the count is in progress. Timer RB counting is stopped when the TCSTF bit is set to 0. During this time, do not access registers associated with timer RB<sup>(1)</sup> other than the TCSTF bit.

### NOTE:

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

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

### 16.2.5.1 Timer mode

The following workaround should be performed in timer mode.

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

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

### 16.2.5.2 Programmable waveform generation mode

The following three workarounds should be performed in programmable waveform 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) To change registers TRBPRES 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 16.25 and 16.26.

The following shows the detailed workaround examples.

- Workaround example (a):

As shown in Figure 16.25, write to registers TRBSC and TRBPR in the timer RB interrupt routine. These write operations must be completed by the beginning of period A.

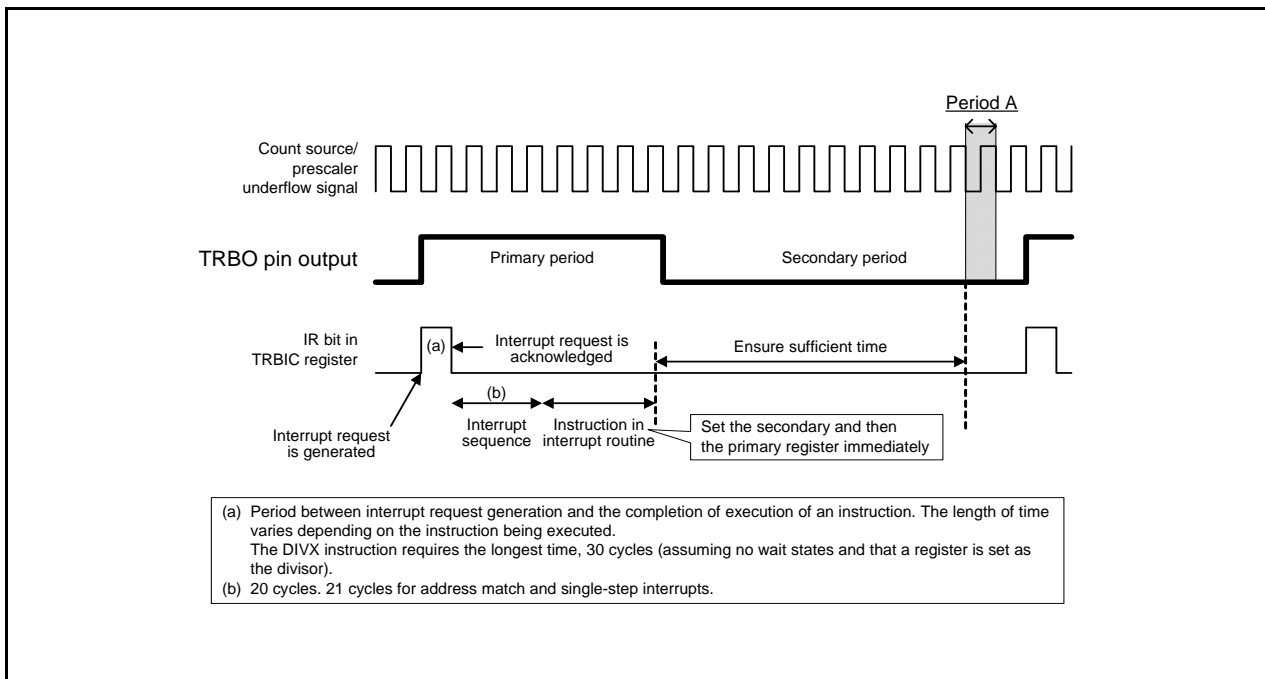
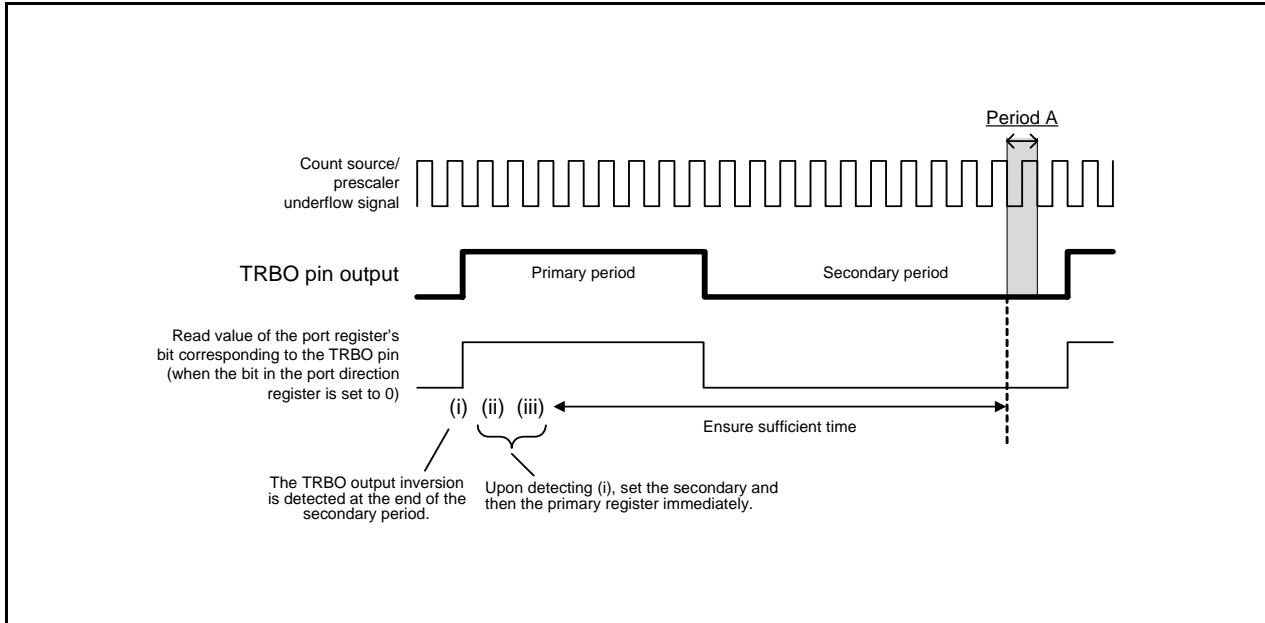


Figure 16.25 Workaround Example (a) When Timer RB interrupt is Used

- Workaround example (b):

As shown in Figure 16.26 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 16.26 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.

### 16.2.5.3 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.

#### 16.2.5.4 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.



## 16.3 Timer RC

### 16.3.1 Overview

Timer RC is a 16-bit timer with four I/O pins.

Timer RC uses either f1 or fOCO40M as its operation clock. Table 16.11 lists the Timer RC Operation Clock.

**Table 16.11 Timer RC Operation Clock**

Condition	Timer RC Operation Clock
Count source is f1, f2, f4, f8, f32, or TRCCLK input (bits TCK2 to TCK0 in TRCCR1 register are set to a value from 000b to 101b)	f1
Count source is fOCO40M (bits TCK2 to TCK0 in TRCCR1 register are set to 110b)	fOCO40M

Table 16.12 lists the Timer RC I/O Pins, and Figure 16.27 shows a Timer RC Block Diagram.

Timer RC has three modes.

- Timer mode

- Input capture function      The counter value is captured to a register, using an external signal as the trigger.
- Output compare function    Matches between the counter and register values are detected. (Pin output state changes when a match is detected.)

The following two modes use the output compare function.

- PWM mode                      Pulses of a given width are output continuously.
- PWM2 mode                    A one-shot waveform or PWM waveform is output following the trigger after the wait time has elapsed.

Input capture function, output compare function, and PWM mode settings may be specified independently for each pin.

In PWM2 mode waveforms are output based on a combination of the counter or the register.

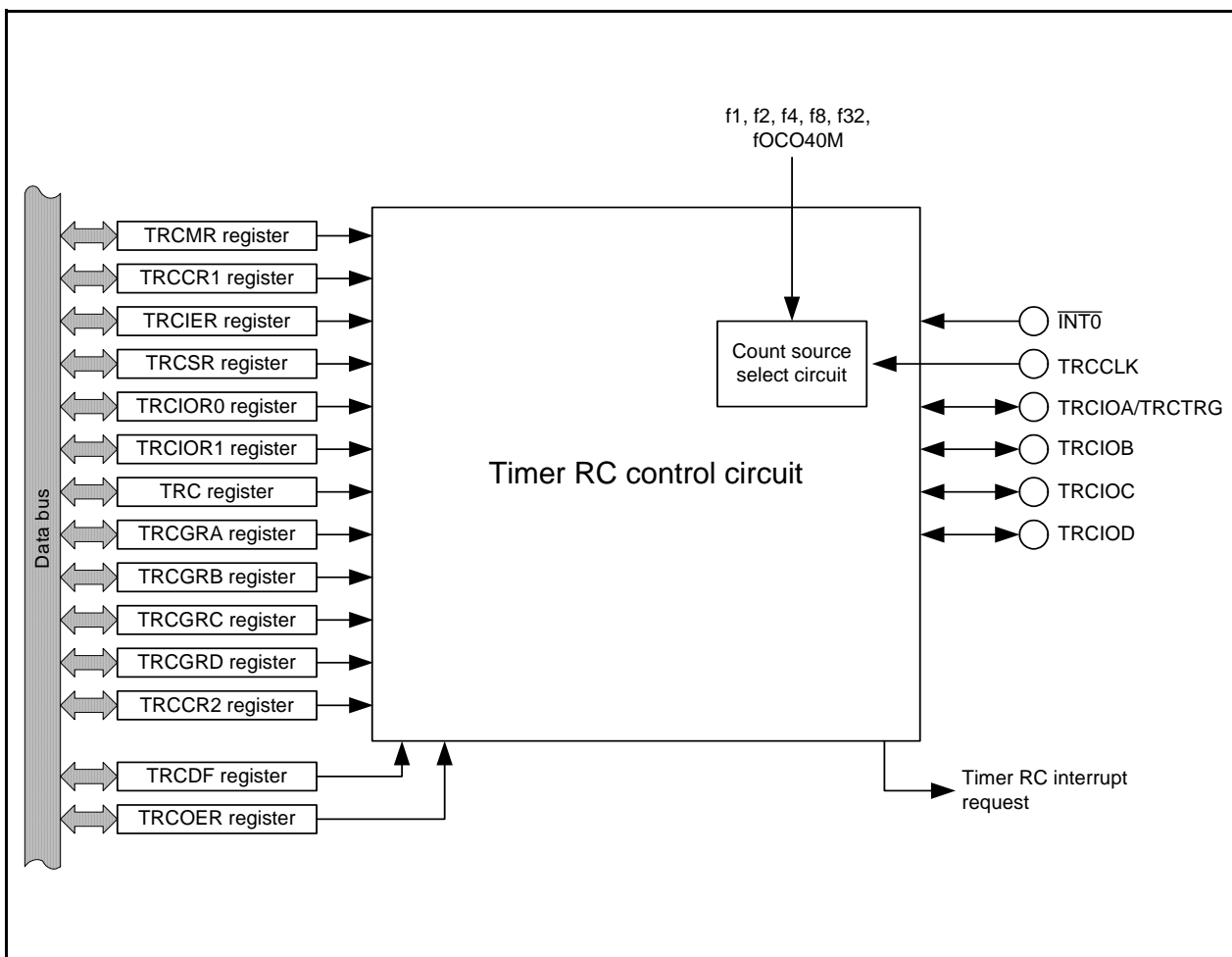


Figure 16.27 Timer RC Block Diagram

Table 16.12 Timer RC I/O Pins

Pin Name	I/O	Function
TRCIOA(P1_1)	I/O	Function differs according to the mode. Refer to descriptions of individual modes for details
TRCIOB(P1_2)		
TRCIOC(P3_4)		
TRCIOD(P3_5)		
TRCCLK(P3_3)	Input	External clock input
TRCTRG(P1_1)	Input	PWM2 mode external trigger input

### 16.3.2 Registers Associated with Timer RC

Table 16.13 lists the Registers Associated with Timer RC. Figures 16.28 to 16.38 show details of the registers associated with timer RC.

**Table 16.13 Registers Associated with Timer RC**

Address	Symbol	Mode				Related Information
		Timer		PWM	PWM2	
		Input Capture Function	Output Compare Function			
00F7h	PINSR3	Valid	Valid	Valid	Valid	Pin Select Register 3 Figure 16.28 PINSR3 Register
0120h	TRCMR	Valid	Valid	Valid	Valid	Timer RC mode register Figure 16.29 TRCMR Register
0121h	TRCCR1	Valid	Valid	Valid	Valid	Timer RC control register 1 Figure 16.30 TRCCR1 Register Figure 16.51 TRCCR1 Register for Output Compare Function Figure 16.54 TRCCR1 Register in PWM Mode Figure 16.58 TRCCR1 Register in PWM2 Mode
0122h	TRCIER	Valid	Valid	Valid	Valid	Timer RC interrupt enable register Figure 16.31 TRCIER Register
0123h	TRCSR	Valid	Valid	Valid	Valid	Timer RC status register Figure 16.32 TRCSR Register
0124h	TRCIOR0	Valid	Valid	–	–	Timer RC I/O control register 0, timer RC I/O control register 1 Figure 16.38 Registers TRCIOR0 and TRCIOR1 Figure 16.45 TRCIOR0 Register for Input Capture Function Figure 16.46 TRCIOR1 Register for Input Capture Function Figure 16.49 TRCIOR0 Register for Output Compare Function Figure 16.50 TRCIOR1 Register for Output Compare Function
0125h	TRCIOR1					
0126h 0127h	TRC	Valid	Valid	Valid	Valid	Timer RC counter Figure 16.33 TRC Register
0128h 0129h	TRCGRA	Valid	Valid	Valid	Valid	Timer RC general registers A, B, C, and D Figure 16.34 Registers TRCGRA, TRCGRB, TRCGRC, and TRCGRD
012Ah 012Bh	TRCGRB					
012Ch 012Dh	TRCGRC					
012Eh 012Fh	TRCGRD					
0130h	TRCCR2	–	–	–	Valid	Timer RC control register 2 Figure 16.35 TRCCR2 Register
0131h	TRCDF	Valid	–	–	Valid	Timer RC digital filter function select register Figure 16.36 TRCDF Register
0132h	TRCOER	–	Valid	Valid	Valid	Timer RC output mask enable register Figure 16.37 TRCOER Register

– : Invalid

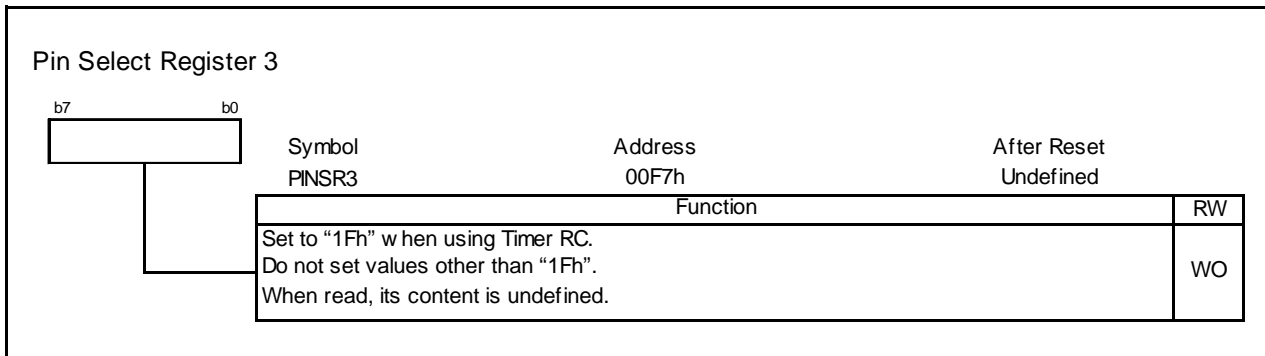


Figure 16.28 PINSR3 Register

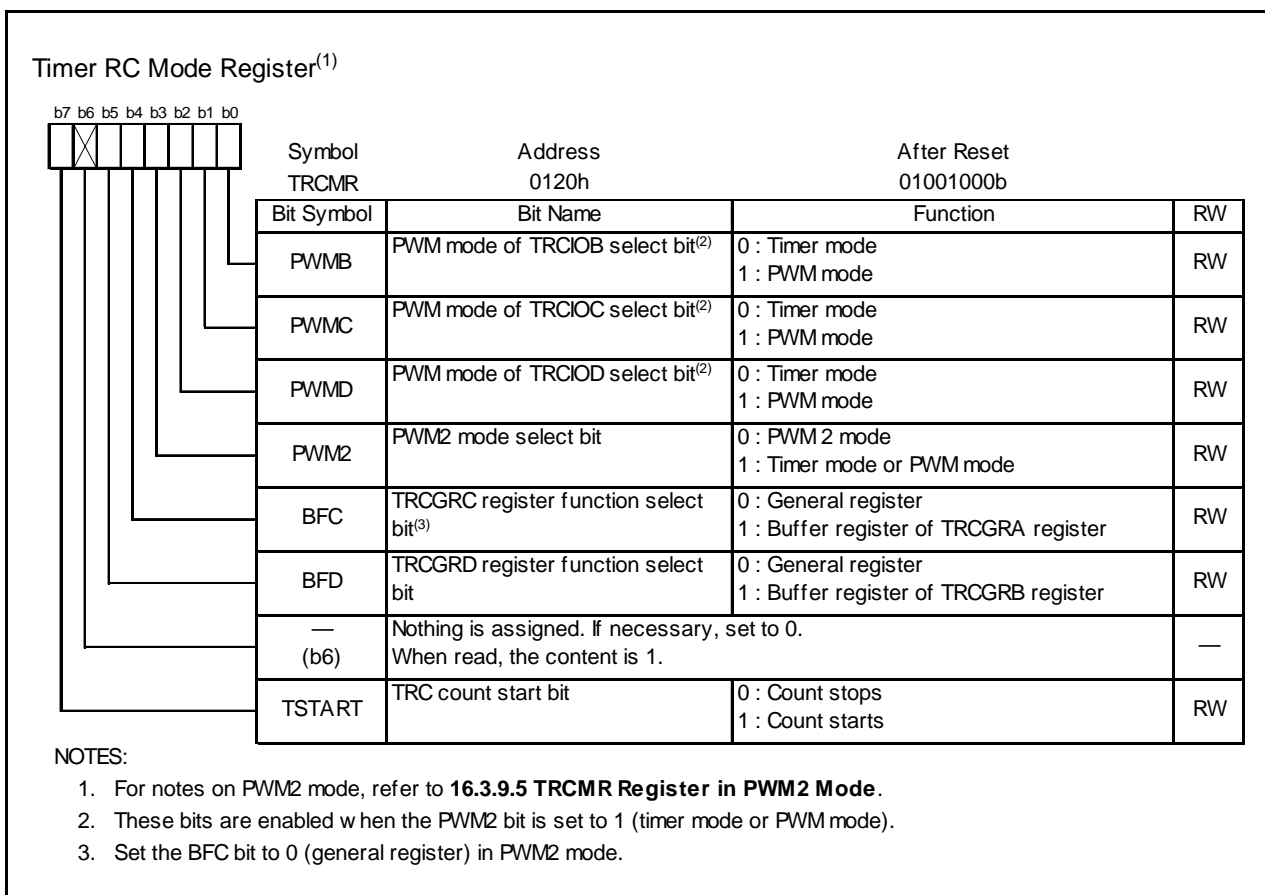


Figure 16.29 TRCMR Register

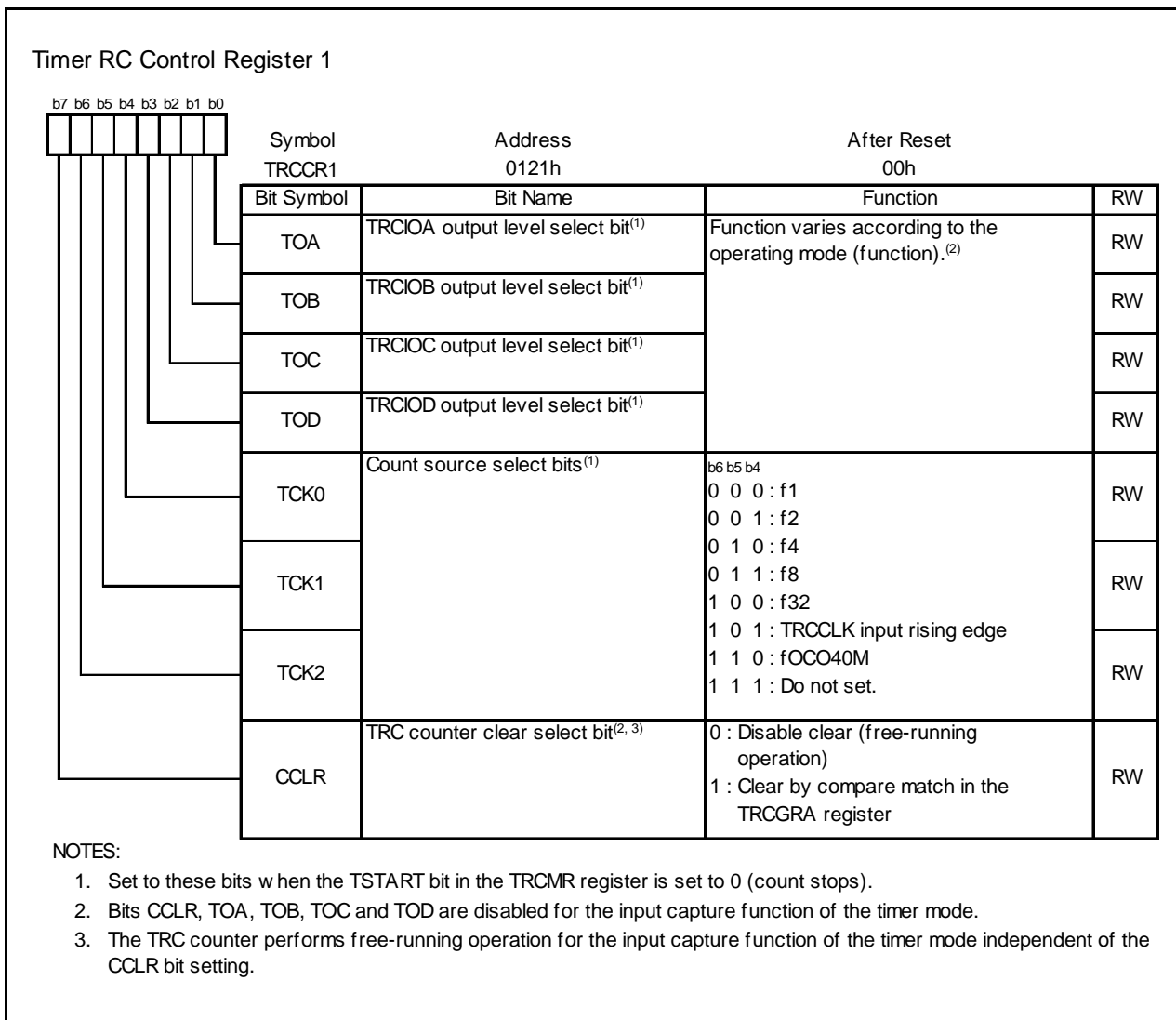


Figure 16.30 TRCCR1 Register

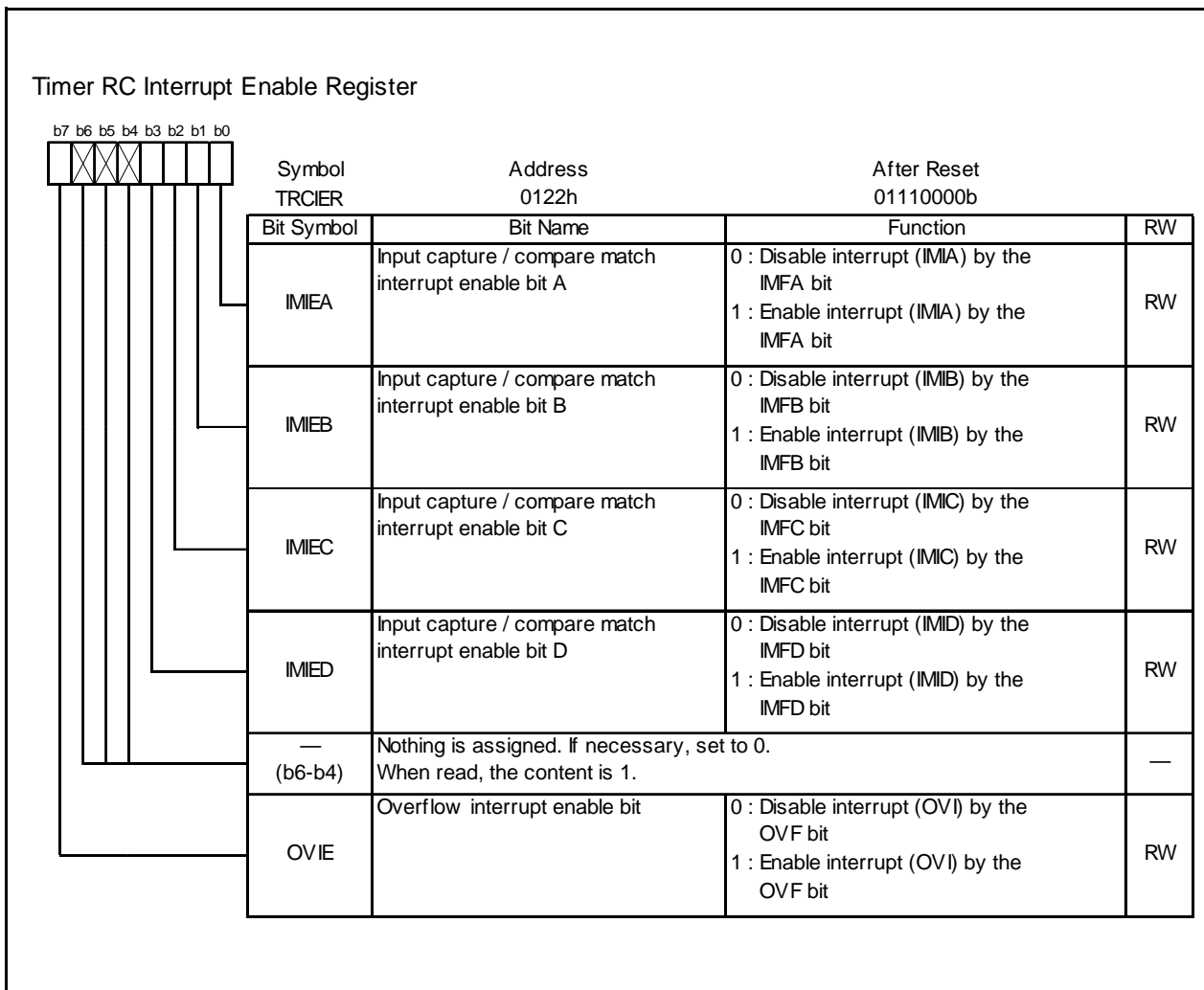


Figure 16.31 TRCIER Register

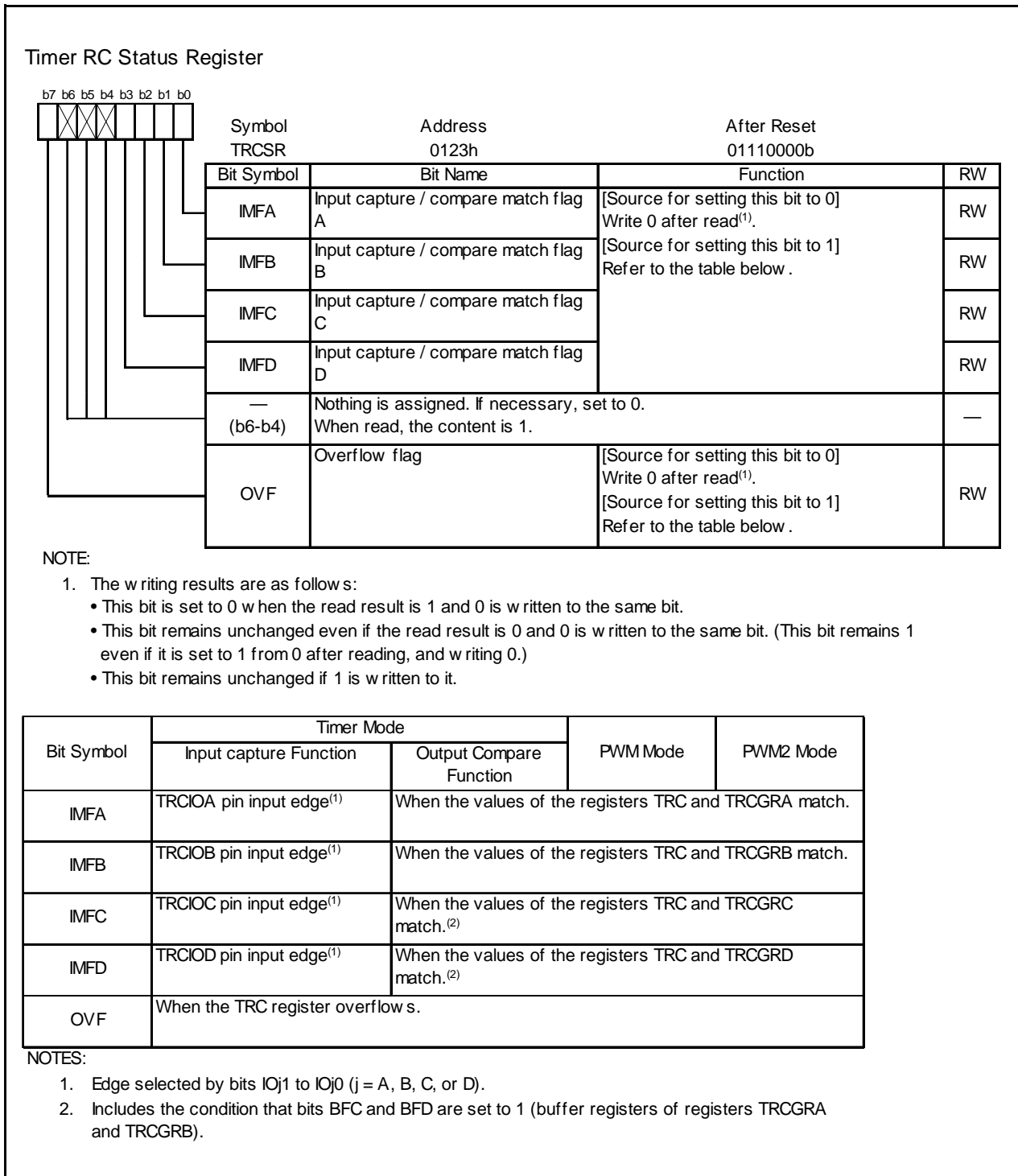
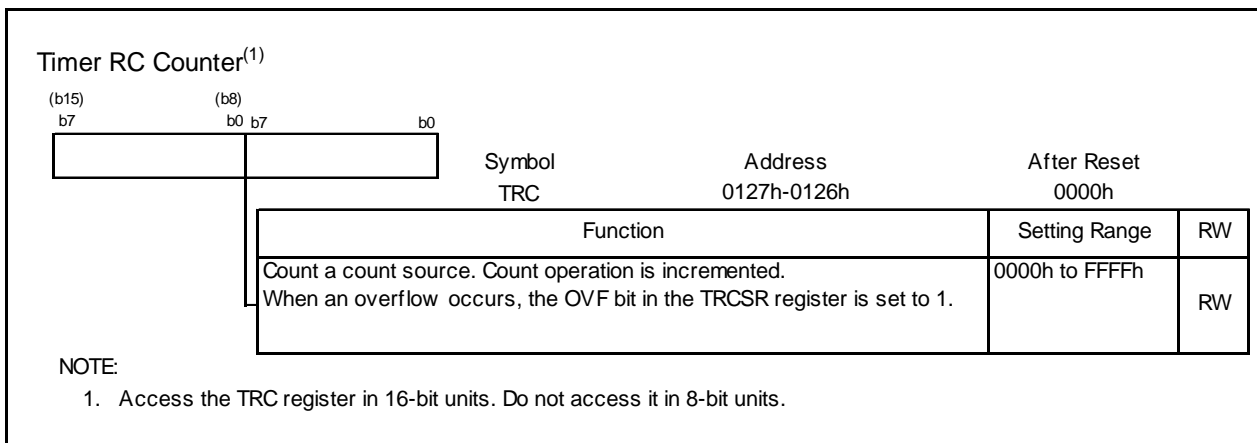
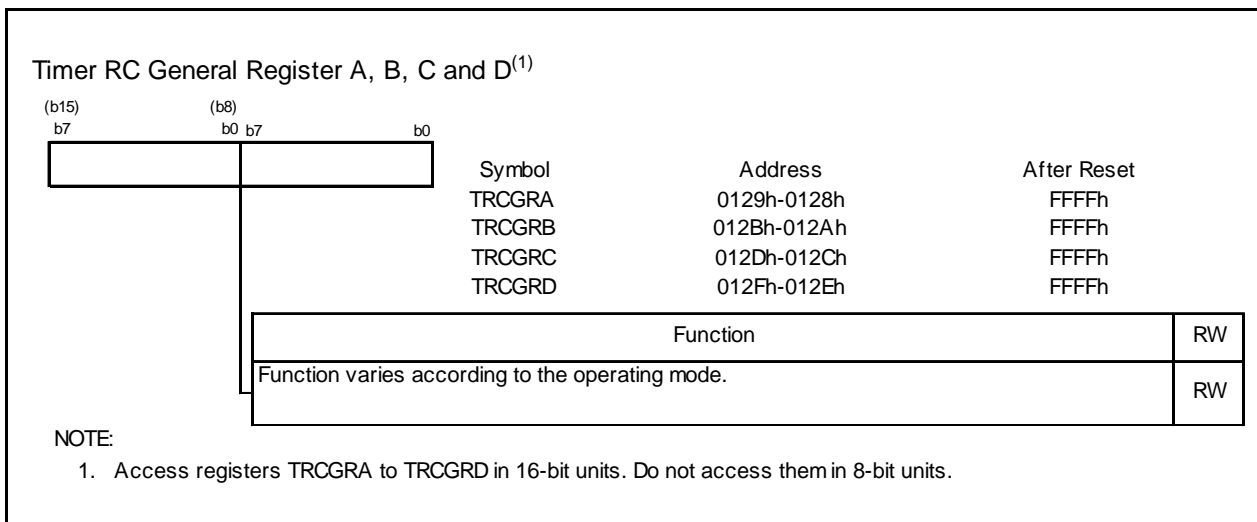


Figure 16.32 TRCSR Register



**Figure 16.33 TRC Register**



**Figure 16.34 Registers TRCGRA, TRCGRB, TRCGRC, and TRCGRD**



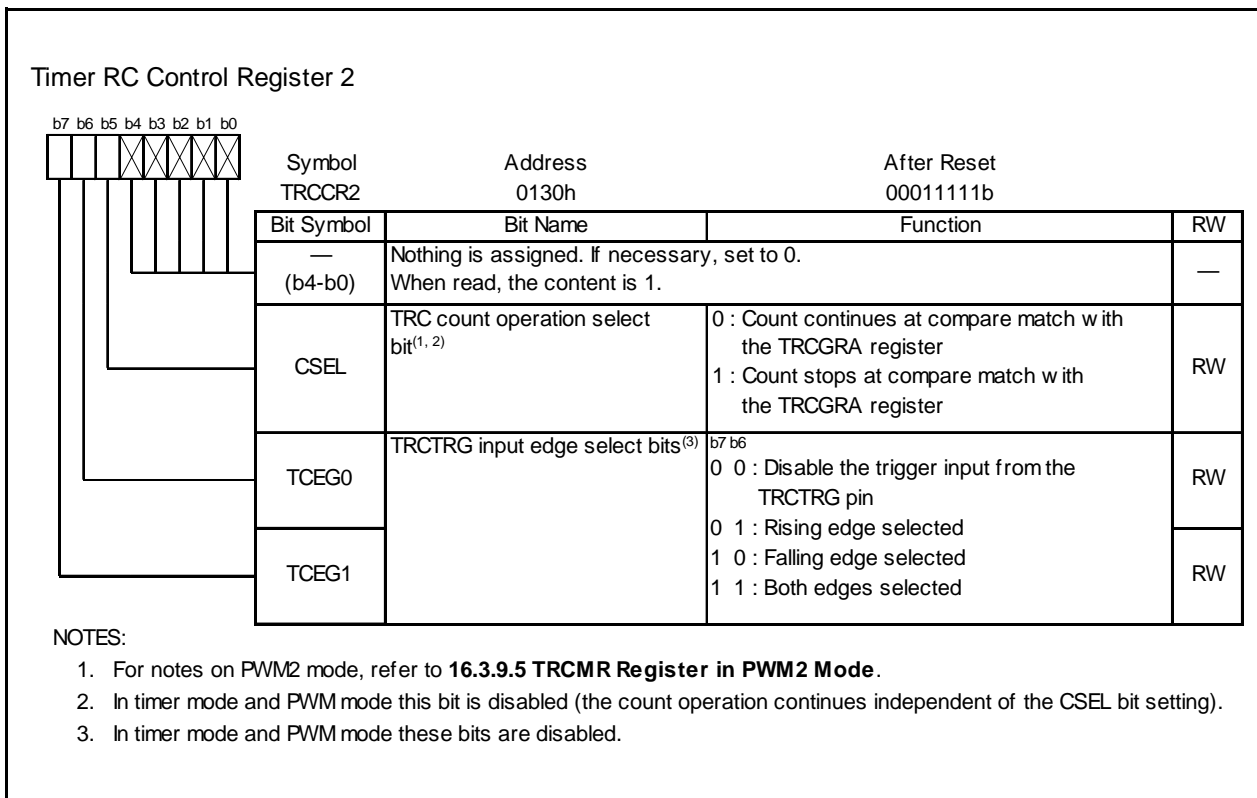


Figure 16.35 TRCCR2 Register

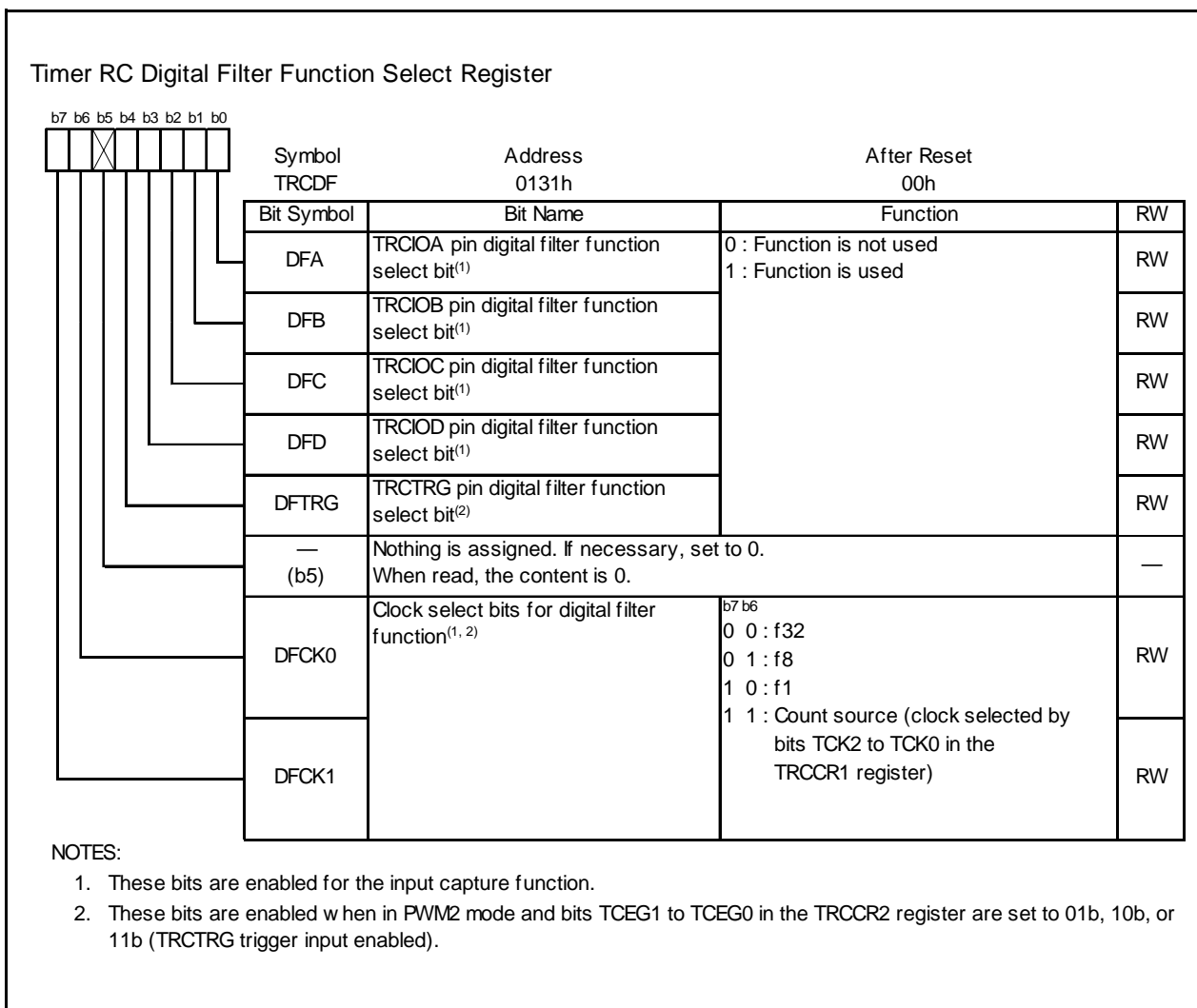


Figure 16.36 TRCDF Register

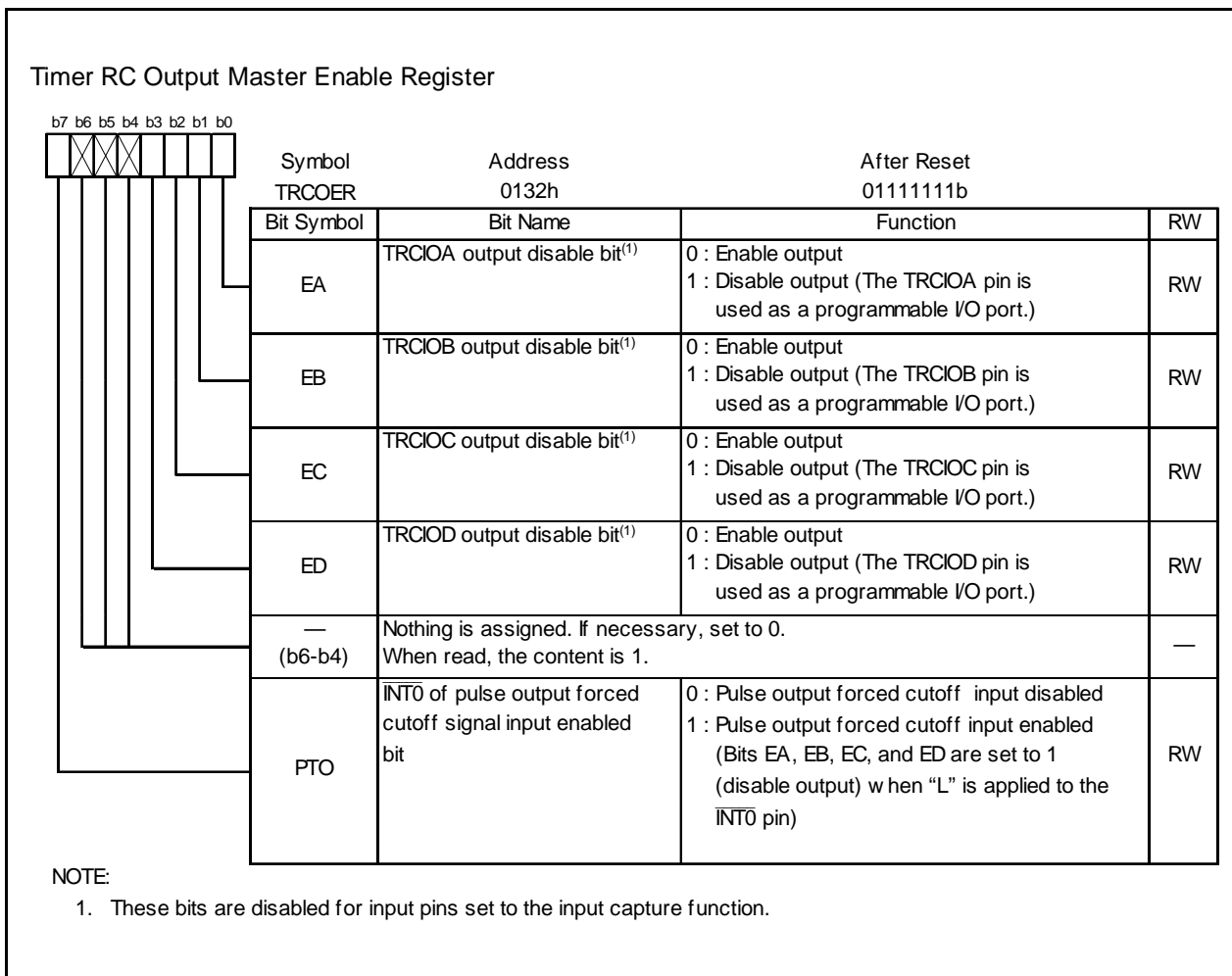


Figure 16.37 TRCOER Register

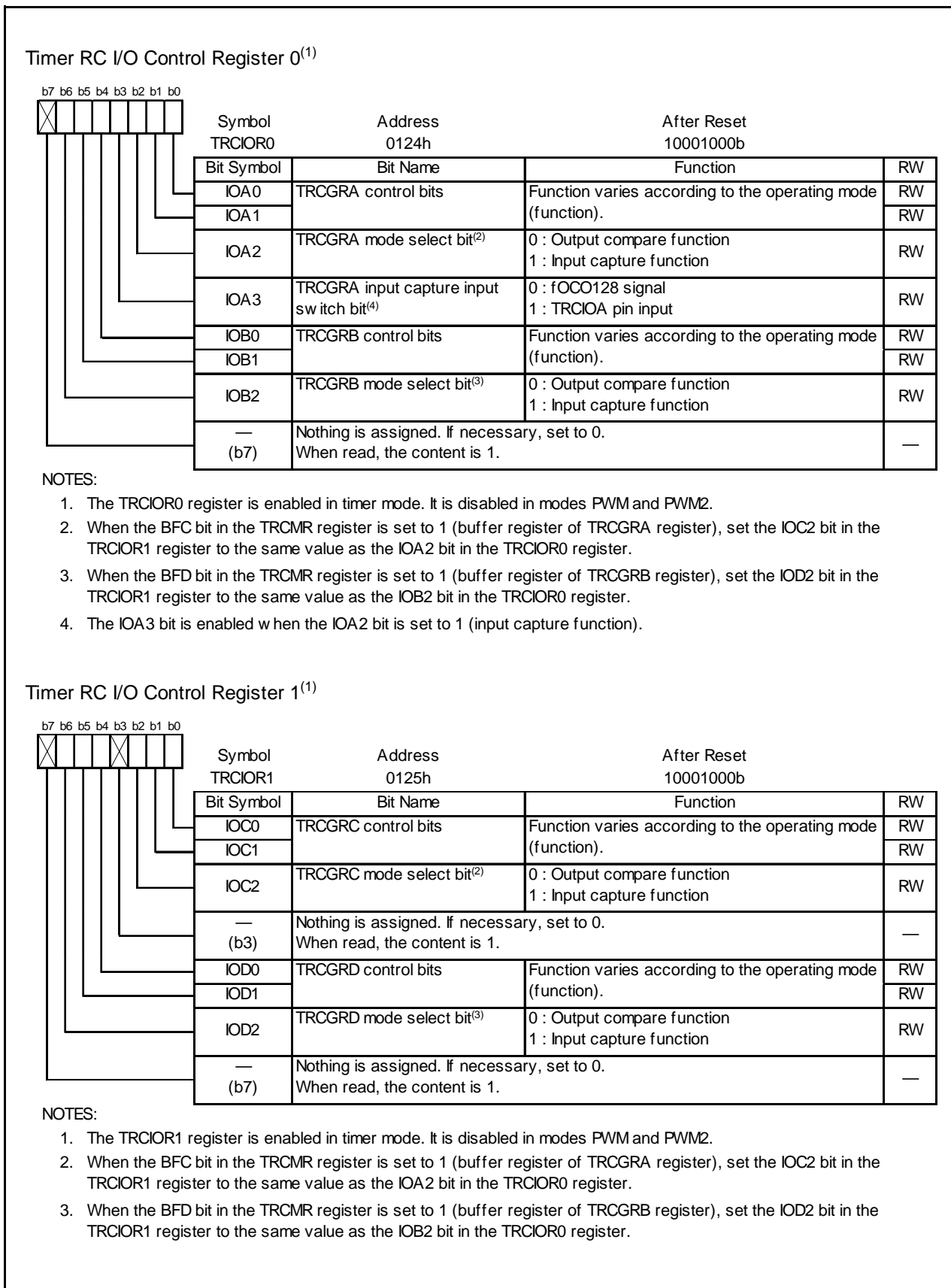


Figure 16.38 Registers TRCIOR0 and TRCIOR1

### 16.3.3 Common Items for Multiple Modes

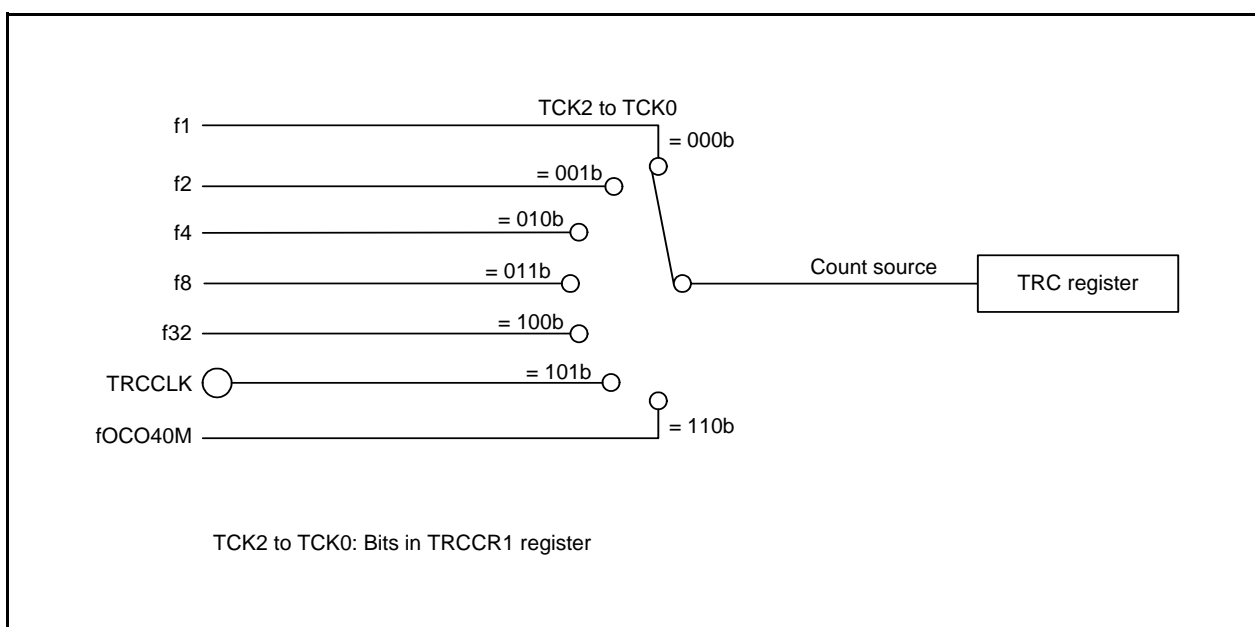
#### 16.3.3.1 Count Source

The method of selecting the count source is common to all modes.

Table 16.14 lists the Count Source Selection, and Figure 16.39 shows a Count Source Block Diagram.

**Table 16.14 Count Source Selection**

Count Source	Selection Method
f1, f2, f4, f8, f32	Count source selected using bits TCK2 to TCK0 in TRCCR1 register
fOCO40M	FRA00 bit in FRA0 register set to 1 (high-speed on-chip oscillator on) and bits TCK2 to TCK0 in TRCCR1 register are set to 110b (fOCO40M)
External signal input to TRCCLK pin	Bits TCK2 to TCK0 in TRCCR1 register are set to 101b (count source is rising edge of external clock) and PD5_0 bit in PD5 register is set to 0 (input mode)



**Figure 16.39 Count Source Block Diagram**

The pulse width of the external clock input to the TRCCLK pin should be three cycles or more of the timer RC operation clock (see **Table 16.11 Timer RC Operation Clock**).

To select fOCO40M as the count source, set the FRA00 bit in the FRA0 register set to 1 (high-speed on-chip oscillator on), and then set bits TCK2 to TCK0 in the TRCCR1 register to 110b (fOCO40M).

### 16.3.3.2 Buffer Operation

Bits BFC and BFD in the TRCMR register are used to select the TRCGRC or TRCGRD register as the buffer register for the TRCGRA or TRCGRB register.

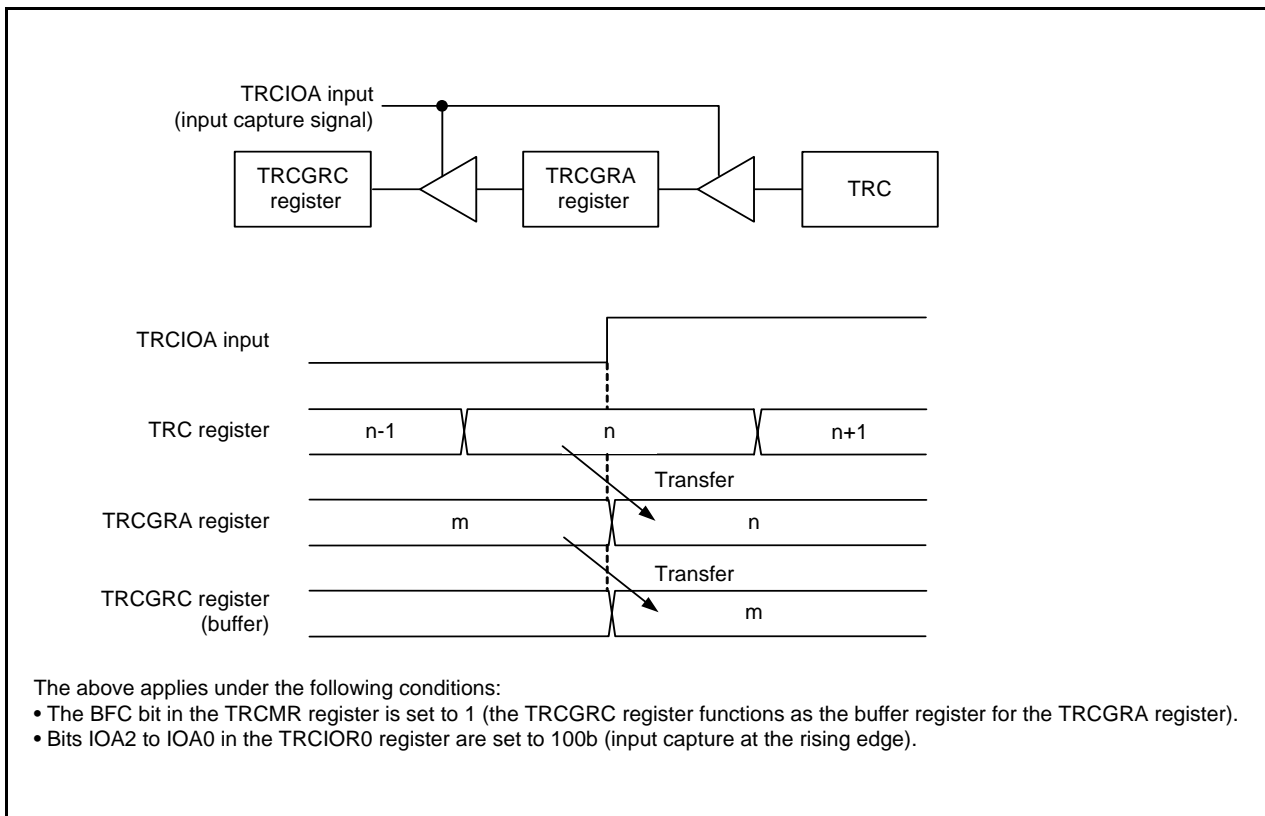
- Buffer register for TRCGRA register: TRCGRC register
- Buffer register for TRCGRB register: TRCGRD register

Buffer operation differs depending on the mode.

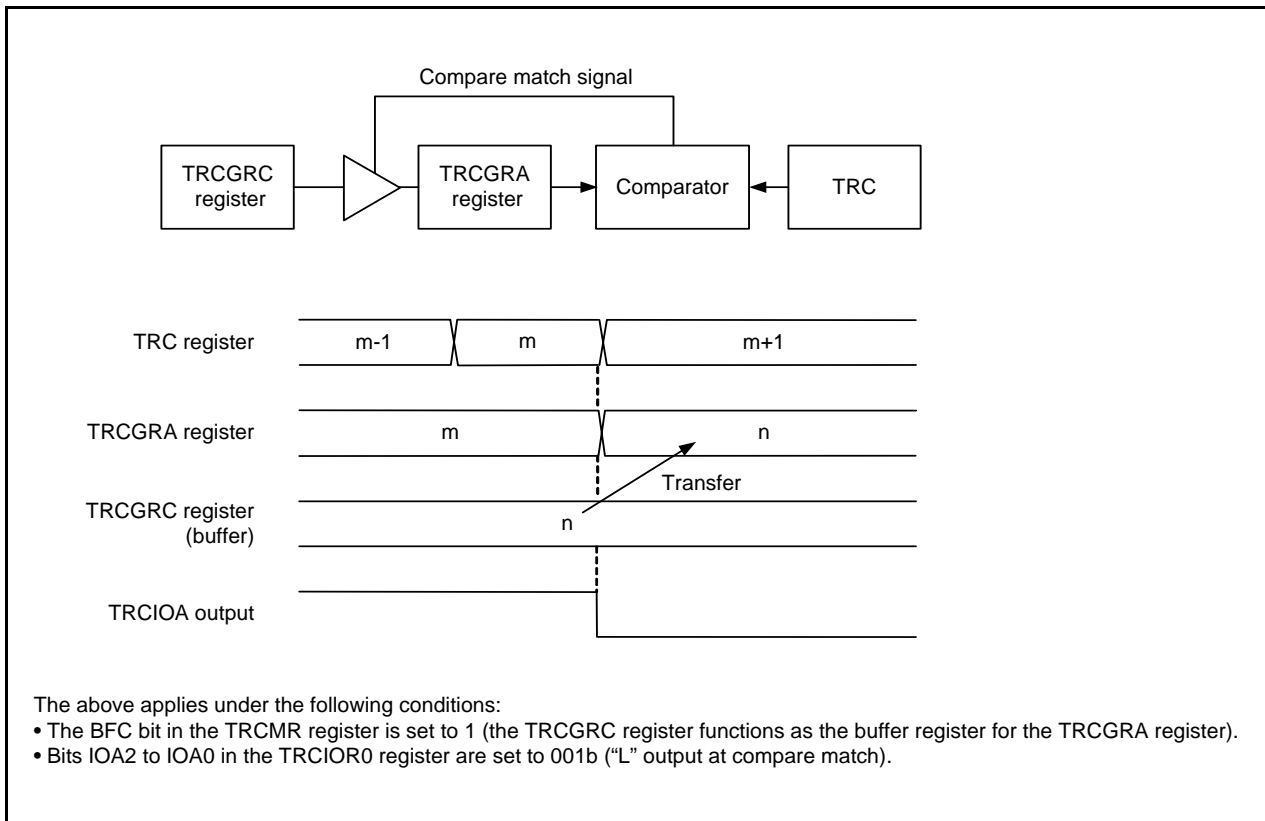
Table 16.15 lists the Buffer Operation in Each Mode, Figure 16.40 shows the Buffer Operation for Input Capture Function, and Figure 16.41 shows the Buffer Operation for Output Compare Function.

**Table 16.15 Buffer Operation in Each Mode**

Function, Mode	Transfer Timing	Transfer Destination Register
Input capture function	Input capture signal input	Contents of TRCGRA (TRCGRB) register are transferred to buffer register
Output compare function	Compare match between TRC register and TRCGRA (TRCGRB) register	Contents of buffer register are transferred to TRCGRA (TRCGRB) register
PWM mode		
PWM2 mode	<ul style="list-style-type: none"> <li>• Compare match between TRC register and TRCGRA register</li> <li>• TRCTRГ pin trigger input</li> </ul>	Contents of buffer register (TRCGRD) are transferred to TRCGRB register



**Figure 16.40 Buffer Operation for Input Capture Function**



**Figure 16.41 Buffer Operation for Output Compare Function**

Make the following settings in timer mode.

- To use the TRCGRC register as the buffer register for the TRCGRA register:  
Set the IOC2 bit in the TRCIOR1 register to the same value as the IOA2 bit in the TRCIOR0 register.
- To use the TRCGRD register as the buffer register for the TRCGRB register:  
Set the IOD2 bit in the TRCIOR1 register to the same value as the IOB2 bit in the TRCIOR0 register.

The output compare function, PWM mode, or PWM2 mode, and the TRCGRC or TRCGRD register is functioning as a buffer register, the IMFC bit or IMFD bit in the TRCSR register is set to 1 when a compare match with the TRC register occurs.

The input capture function and the TRCGRC register or TRCGRD register is functioning as a buffer register, the IMFC bit or IMFD bit in the TRCSR register is set to 1 at the input edge of a signal input to the TRCIO pin or TRCIOD pin.

### 16.3.3.3 Digital Filter

The input to TRCTR<sub>j</sub> or TRCIO<sub>j</sub> (j = A, B, C, or D) is sampled, and the level is considered to be determined when three matches occur. The digital filter function and sampling clock are selected using the TRCDF register. Figure 16.42 shows a Block Diagram of Digital Filter.

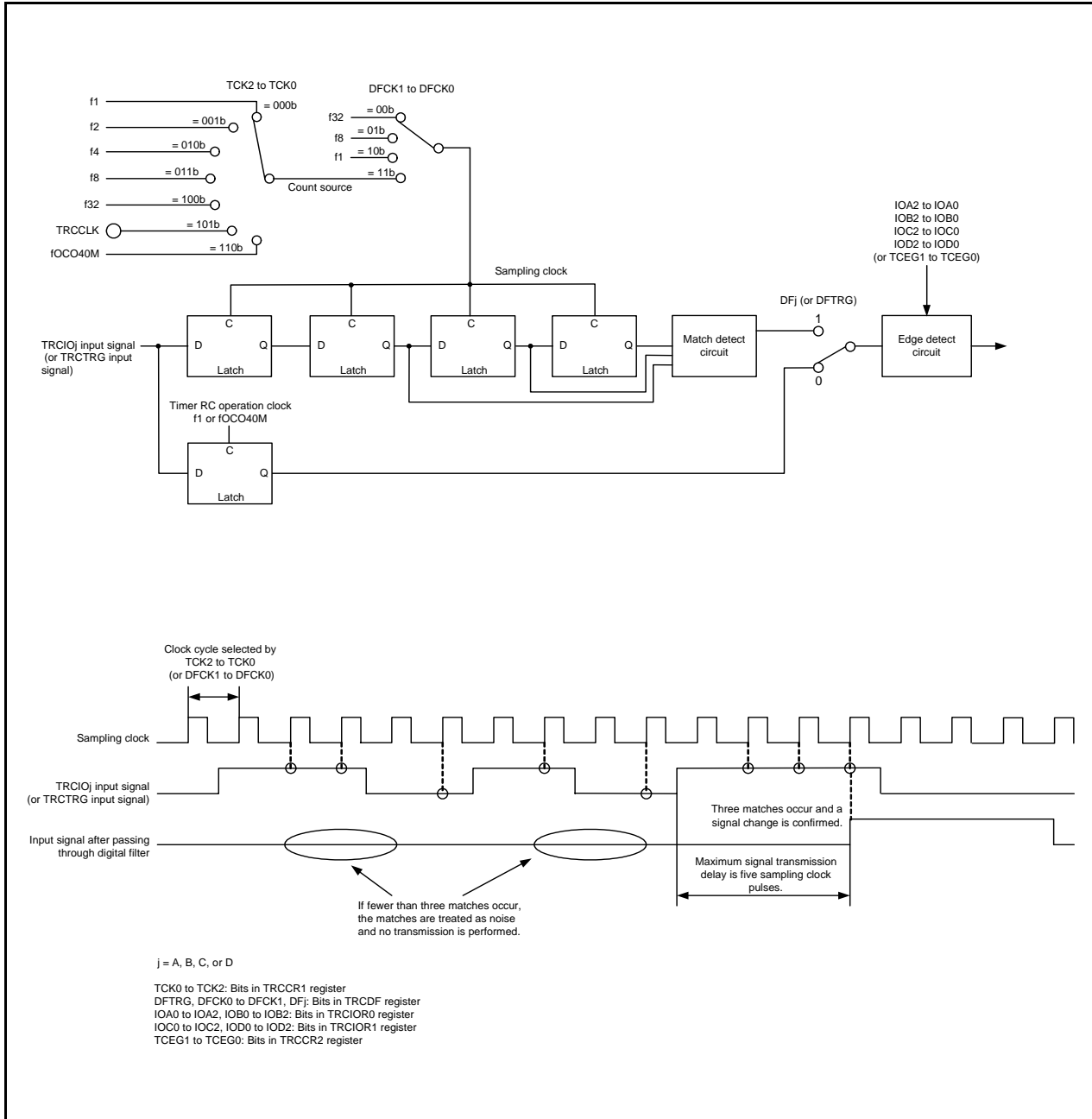


Figure 16.42 Block Diagram of Digital Filter



### 16.3.3.4 Forced Cutoff of Pulse Output

When using the timer mode's output compare function, the PWM mode, or the PWM2 mode, pulse output from the TRCIOj (j = A, B, C, or D) output pin can be forcibly cut off and the TRCIOj pin set to function as a programmable I/O port by means of input to the  $\overline{\text{INT0}}$  pin.

A pin used for output by the timer mode's output compare function, the PWM mode, or the PWM2 mode can be set to function as the timer RC output pin by setting the Ej bit in the TRCOER register to 0 (timer RC output enabled). If "L" is input to the  $\overline{\text{INT0}}$  pin while the PTO bit in the TRCOER register is set to 1 (pulse output forced cutoff signal input  $\overline{\text{INT0}}$  enabled), bits EA, EB, EC, and ED in the TRCOER register are all set to 1 (timer RC output disabled, TRCIOj output pin functions as the programmable I/O port). When one or two cycles of the timer RC operation clock after "L" input to the  $\overline{\text{INT0}}$  pin (refer to **Table 16.11 Timer RC Operation Clock**) has elapsed, the TRCIOj output pin becomes a programmable I/O port.

Make the following settings to use this function.

- Set the pin state following forced cutoff of pulse output (high impedance (input), "L" output, or "H" output). (Refer to **7. Programmable I/O Ports**.)
- Set the INT0EN bit to 1 ( $\overline{\text{INT0}}$  input enabled) and the INT0PL bit to 0 (one edge) in the INTEN register.
- Set the PD4\_5 bit in the PD4 register to 0 (input mode).
- Select the  $\overline{\text{INT0}}$  digital filter by means of bits INT0F1 to INT0F0 in the INTF register.
- Set the PTO bit in the TRCOER register to 1 (pulse output forced cutoff signal input  $\overline{\text{INT0}}$  enabled).

The IR bit in the INT0IC register is set to 1 (interrupt request) in accordance with the setting of the POL bit and a change in the  $\overline{\text{INT0}}$  pin input (refer to **12.6 Notes on Interrupts**).

For details on interrupts, refer to **12. Interrupts**.

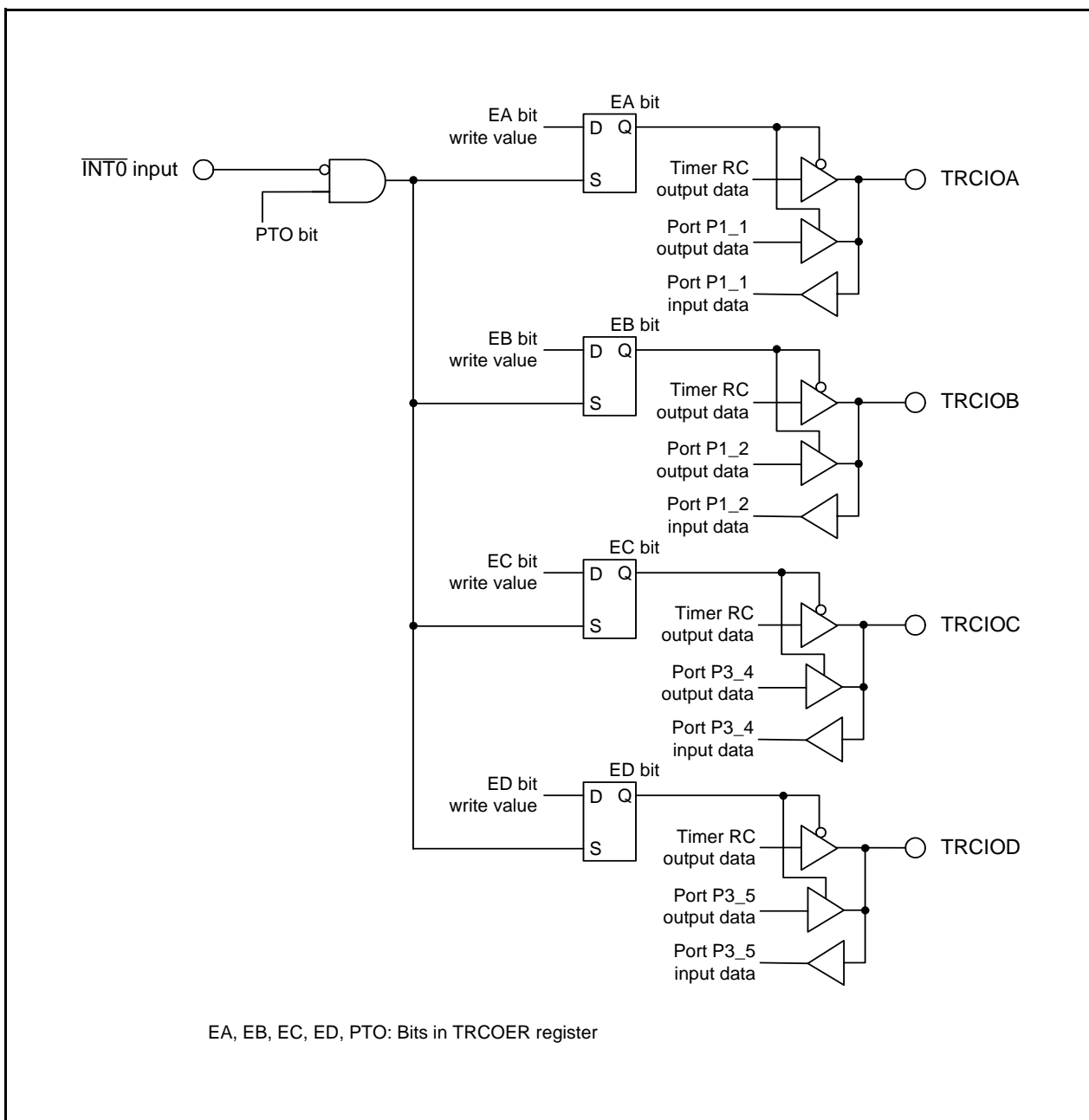


Figure 16.43 Forced Cutoff of Pulse Output

### 16.3.4 Timer Mode (Input Capture Function)

This function measures the width or period of an external signal. An external signal input to the TRCIO<sub>j</sub> (j = A, B, C, or D) pin acts as a trigger for transferring the contents of the TRC register (counter) to the TRCGR<sub>j</sub> register (input capture). The input capture function, or any other mode or function, can be selected for each individual pin.

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

Table 16.16 lists the Specifications of Input Capture Function, Figure 16.44 shows a Block Diagram of Input Capture Function, Figures 16.45 and 16.46 show the registers associated with the input capture function, Table 16.17 lists the Functions of TRCGR<sub>j</sub> Register when Using Input Capture Function, and Figure 16.47 shows an Operating Example of Input Capture Function.

**Table 16.16 Specifications of Input Capture Function**

Item	Specification
Count source	f1, f2, f4, f8, f32, fOCO40M, or external signal (rising edge) input to TRCCLK pin
Count operation	Increment
Count period	$1/f_k \times 65,536$ f <sub>k</sub> : Count source frequency
Count start condition	1 (count starts) is written to the TSTART bit in the TRCMR register.
Count stop condition	0 (count stops) is written to the TSTART bit in the TRCMR register. The TRC register retains a value before count stops.
Interrupt request generation timing	<ul style="list-style-type: none"> <li>Input capture (valid edge of TRCIO<sub>j</sub> input or fOCO128 signal edge)</li> <li>The TRC register overflows.</li> </ul>
TRCIOA, TRCIOB, TRCIOC, and TRCIOD pin functions	Programmable I/O port or input capture input (selectable individually by pin)
INT0 pin function	Programmable I/O port or INT0 interrupt input
Read from timer	The count value can be read by reading TRC register.
Write to timer	The TRC register can be written to.
Select functions	<ul style="list-style-type: none"> <li>Input capture input pin select One or more of pins TRCIOA, TRCIOB, TRCIOC, and TRCIOD</li> <li>Input capture input valid edge selected Rising edge, falling edge, or both rising and falling edges</li> <li>Buffer operation (Refer to <b>16.3.3.2 Buffer Operation</b>.)</li> <li>Digital filter (Refer to <b>16.3.3.3 Digital Filter</b>.)</li> <li>Input-capture trigger selected fOCO128 can be selected for input-capture trigger input of the TRCGRA register.</li> </ul>

j = A, B, C, or D

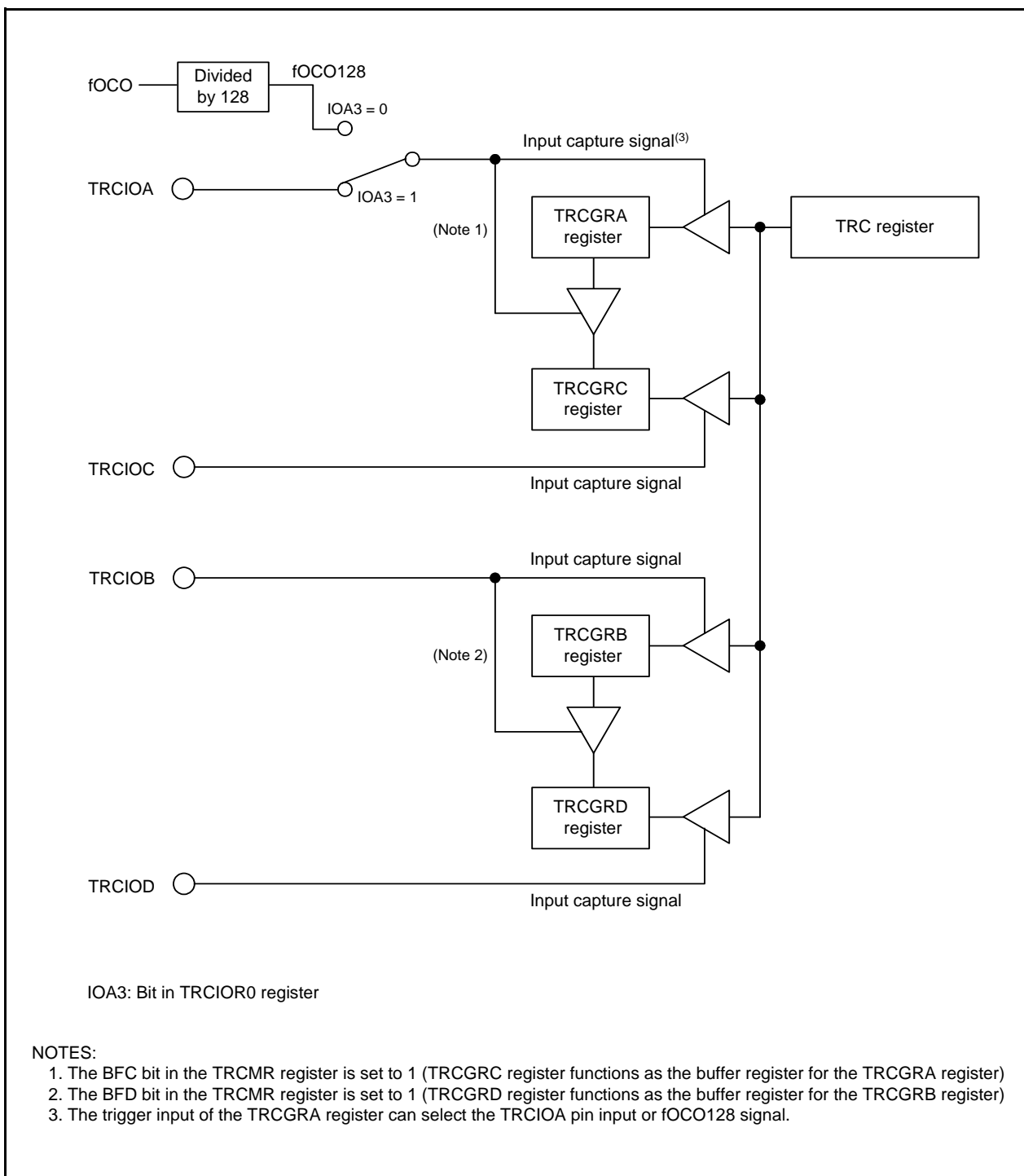


Figure 16.44 Block Diagram of Input Capture Function

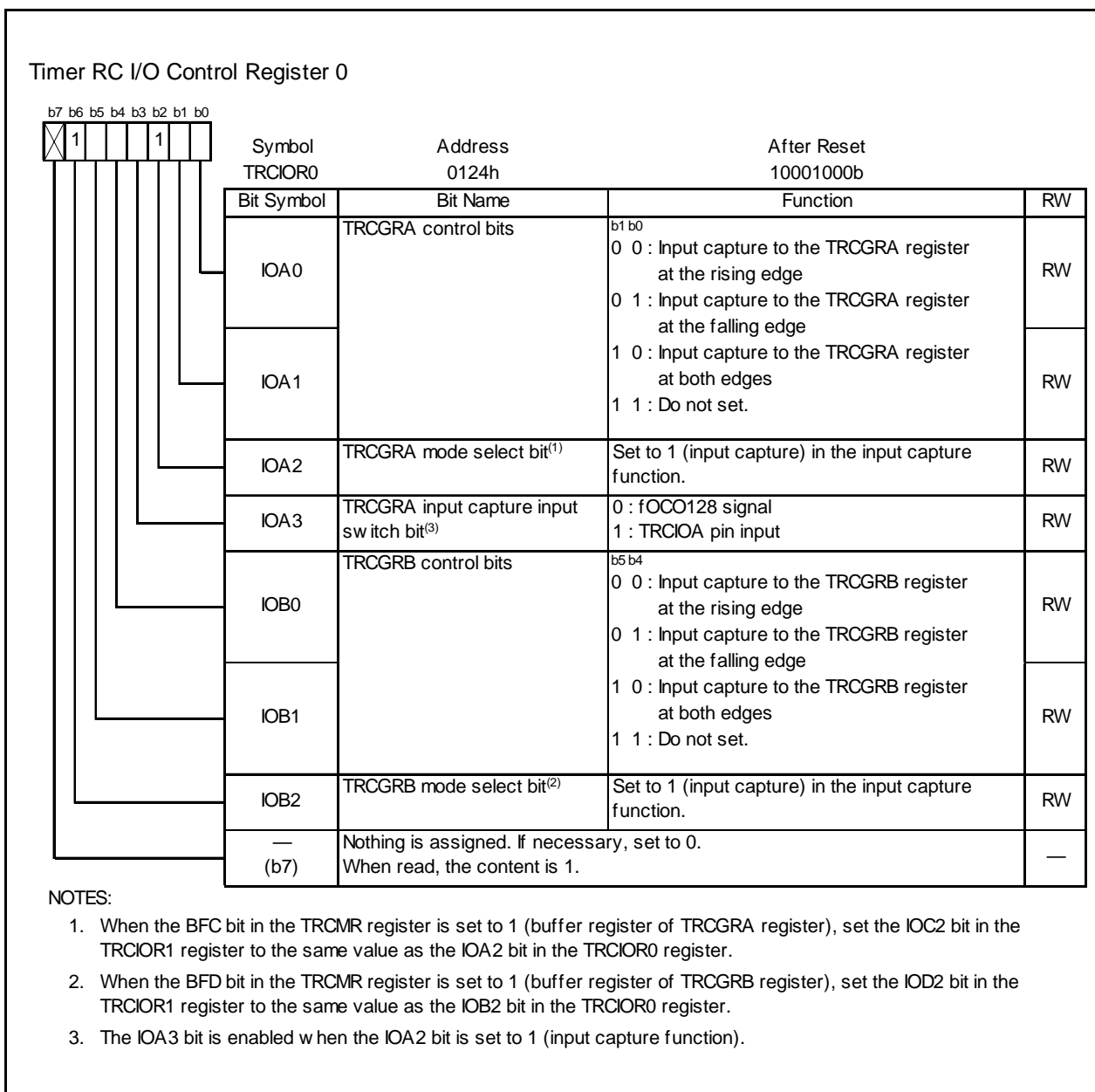


Figure 16.45 TRCIOR0 Register for Input Capture Function

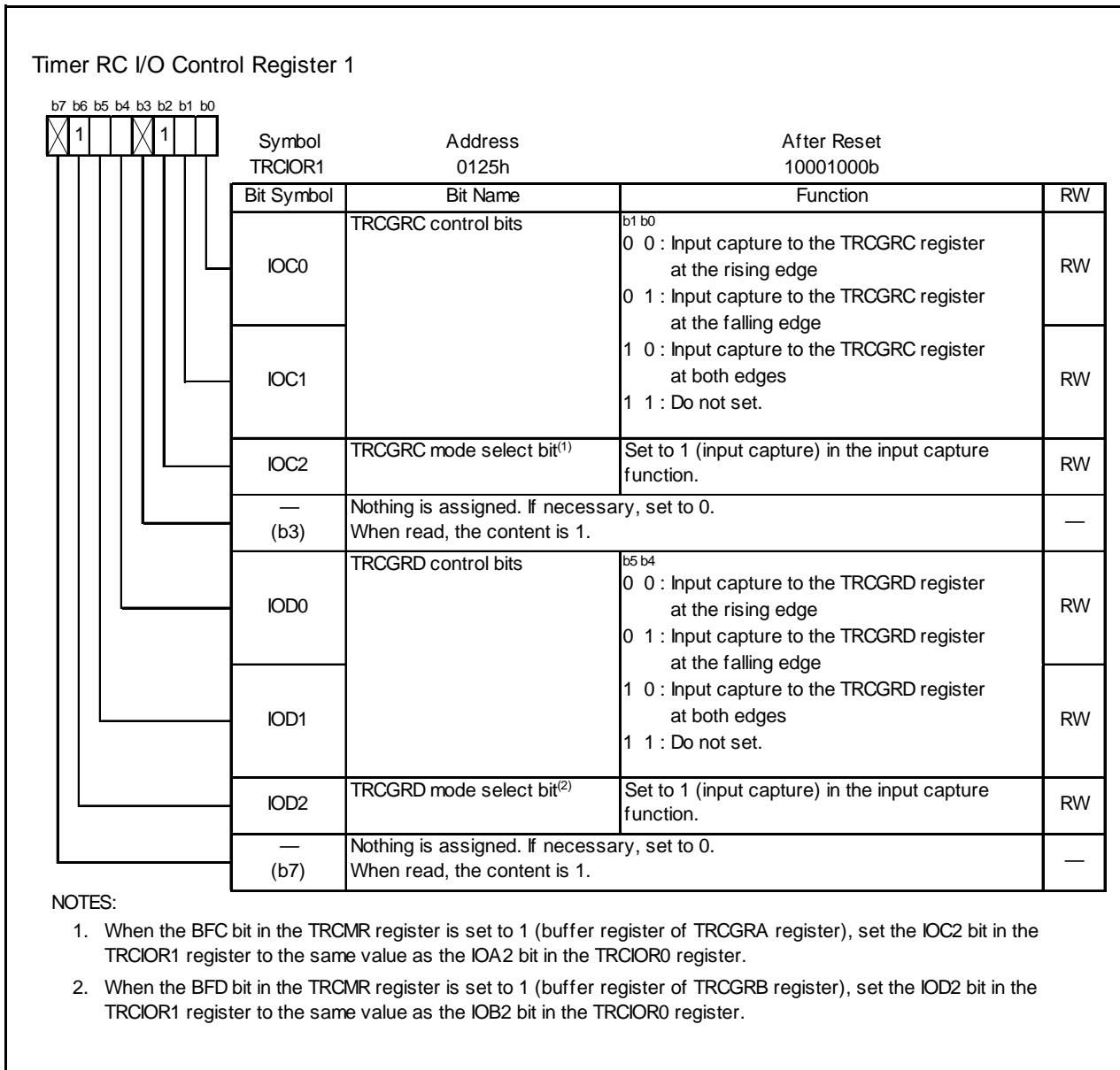


Figure 16.46 TRCIOR1 Register for Input Capture Function

Table 16.17 Functions of TRCGRj Register when Using Input Capture Function

Register	Setting	Register Function	Input Capture Input Pin
TRCGRA	—	General register. Can be used to read the TRC register value at input capture.	TRCIOA
TRCGRB			TRCIOB
TRCGRC	BFC = 0	General register. Can be used to read the TRC register value at input capture.	TRCIOC
TRCGRD	BFD = 0		TRCIOD
TRCGRC	BFC = 1	Buffer registers. Can be used to hold transferred value from the general register. (Refer to <b>16.3.3.2 Buffer Operation</b> .)	TRCIOA
TRCGRD	BFD = 1		TRCIOB

j = A, B, C, or D

BFC, BFD: Bits in TRCMR register

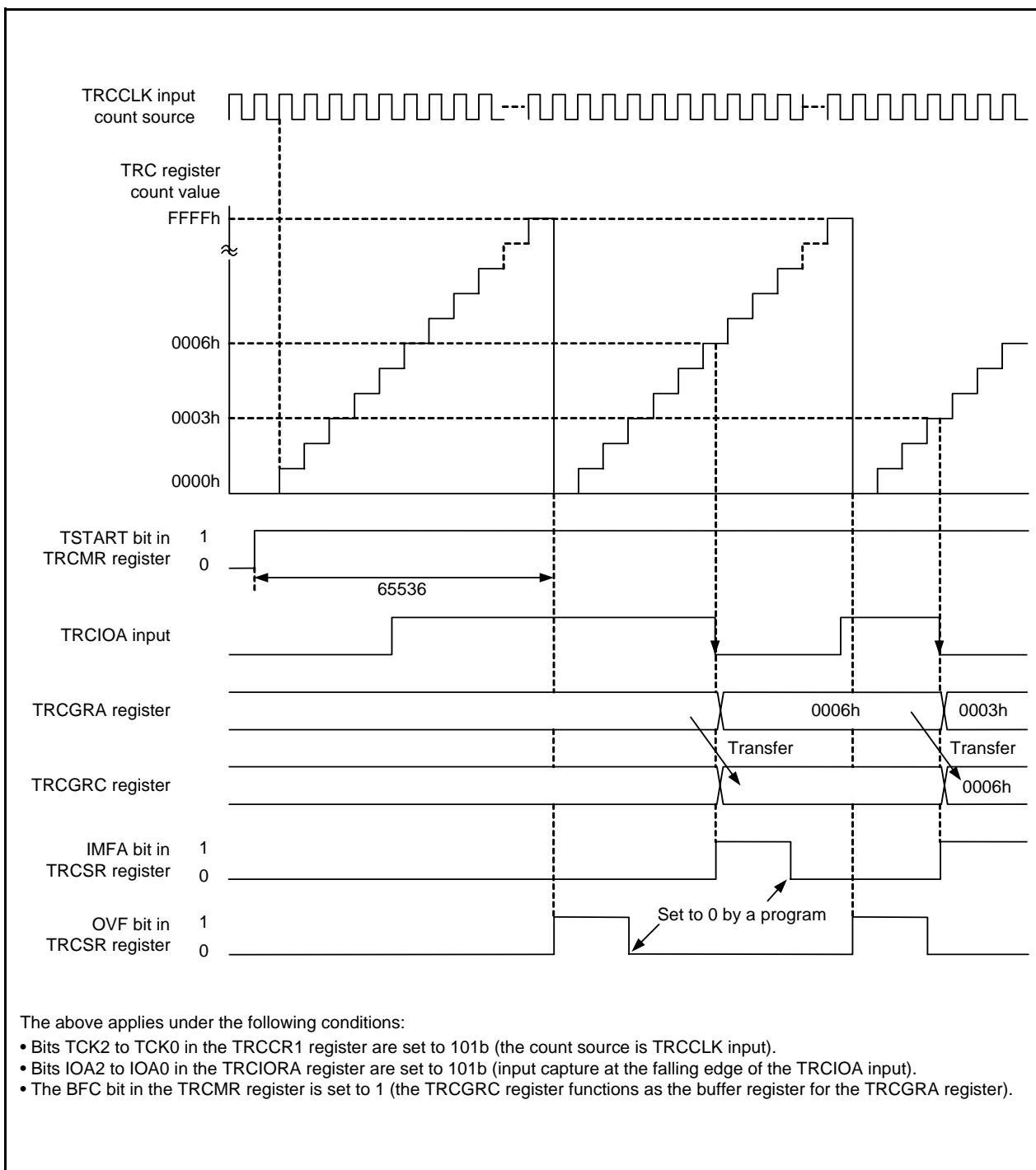


Figure 16.47 Operating Example of Input Capture Function

### 16.3.5 Timer Mode (Output Compare Function)

This function detects when the contents of the TRC register (counter) and the TRCGRj register (j = A, B, C, or D) match (compare match). When a match occurs a signal is output from the TRCIOj pin at a given level. The output compare function, or other mode or function, can be selected for each individual pin.

Table 16.18 lists the Specifications of Output Compare Function, Figure 16.48 shows a Block Diagram of Output Compare Function, Figures 16.49 to 16.51 show the registers associated with the output compare function, Table 16.19 lists the Functions of TRCGRj Register when Using Output Compare Function, and Figure 16.52 shows an Operating Example of Output Compare Function.

**Table 16.18 Specifications of Output Compare Function**

Item	Specification
Count source	f1, f2, f4, f8, f32, fOCO40M, or external signal (rising edge) input to TRCCLK pin
Count operation	Increment
Count period	<ul style="list-style-type: none"> <li>• The CCLR bit in the TRCCR1 register is set to 0 (free running operation): <math>1/fk \times 65,536</math> fk: Count source frequency</li> <li>• The CCLR bit in the TRCCR1 register is set to 1 (TRC register set to 0000h at TRCGRA compare match): <math>1/fk \times (n + 1)</math> n: TRCGRA register setting value</li> </ul>
Waveform output timing	Compare match
Count start condition	1 (count starts) is written to the TSTART bit in the TRCMR register.
Count stop condition	0 (count stops) is written to the TSTART bit in the TRCMR register. The output compare output pin retains output level before count stops, the TRC register retains a value before count stops.
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• Compare match (contents of registers TRC and TRCGRj match)</li> <li>• The TRC register overflows.</li> </ul>
TRCIOA, TRCIOB, TRCIOC, and TRCIOD pin functions	Programmable I/O port or output compare output (selectable individually by 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 TRC register.
Write to timer	The TRC register can be written to.
Select functions	<ul style="list-style-type: none"> <li>• Output compare output pin selected One or more of pins TRCIOA, TRCIOB, TRCIOC, and TRCIOD</li> <li>• Compare match output level select “L” output, “H” output, or toggle output</li> <li>• Initial output level select Sets output level for period from count start to compare match</li> <li>• Timing for clearing the TRC register to 0000h Overflow or compare match with the TRCGRA register</li> <li>• Buffer operation (Refer to <b>16.3.3.2 Buffer Operation</b>.)</li> <li>• Pulse output forced cutoff signal input (Refer to <b>16.3.3.4 Forced Cutoff of Pulse Output</b>.)</li> <li>• Can be used as an internal timer by disabling timer RC output</li> </ul>

j = A, B, C, or D



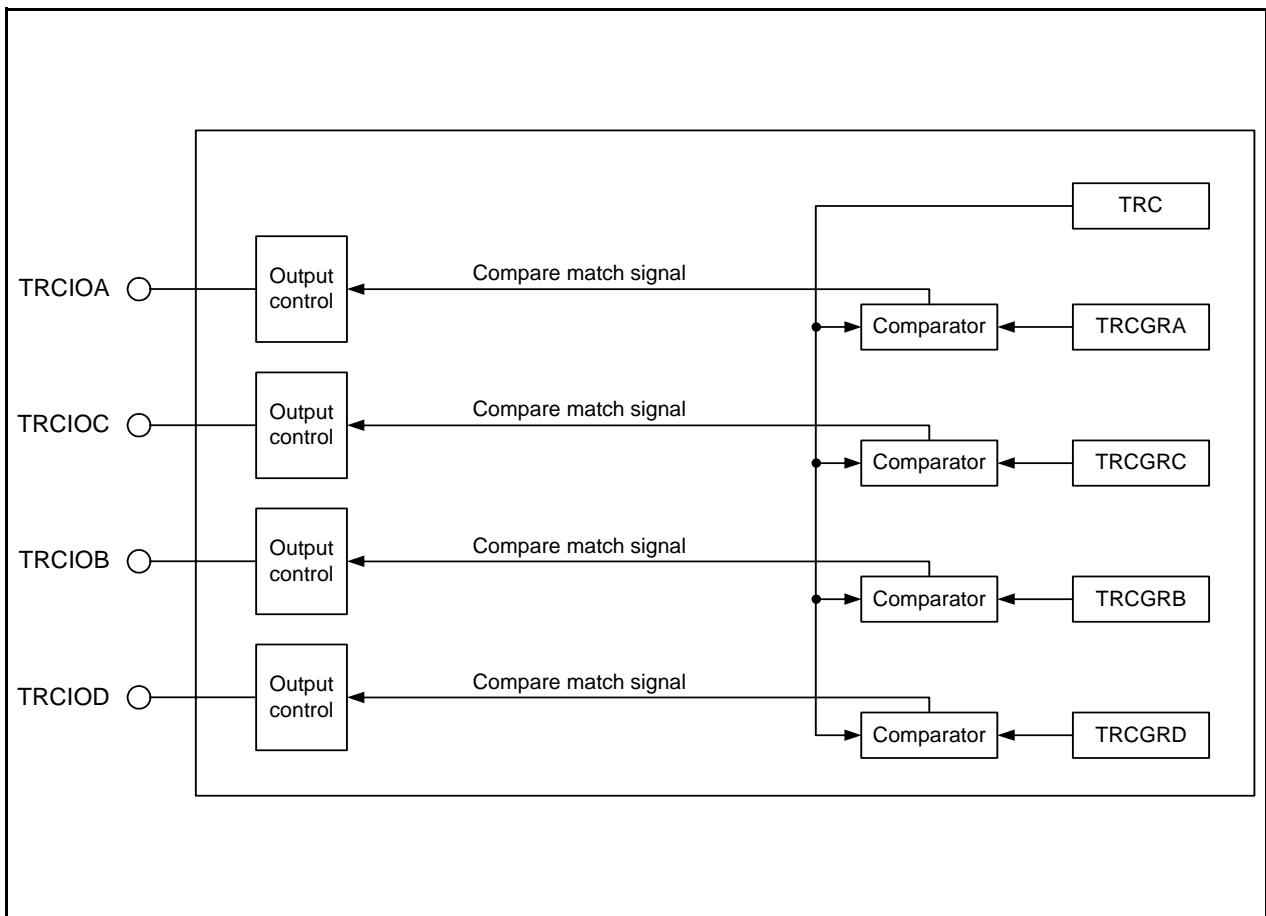


Figure 16.48 Block Diagram of Output Compare Function

Timer RC I/O Control Register 0

Bit	Symbol	Address	After Reset
b7	0	0124h	10001000b
b6	0		
b5	1		
b4	0		
b3	0		
b2	0		
b1	0		
b0	0		

Bit Symbol	Bit Name	Function	RW
IOA0	TRCGRA control bits	b1 b0 0 0 : Disable pin output by compare match (TRCIOA pin functions as the programmable I/O port)	RW
		0 1 : "L" output by compare match in the TRCGRA register	
IOA1	TRCGRA control bits	1 0 : "H" output by compare match in the TRCGRA register	RW
		1 1 : Toggle output by compare match in the TRCGRA register	
IOA2	TRCGRA mode select bit <sup>(1)</sup>	Set to 0 (output compare) in the output compare function.	RW
IOA3	TRCGRA input capture input switch bit	Set to 1.	RW
IOB0	TRCGRB control bits	b5 b4 0 0 : Disable pin output by compare match (TRCIOB pin functions as the programmable I/O port)	RW
		0 1 : "L" output by compare match in the TRCGRB register	
IOB1	TRCGRB control bits	1 0 : "H" output by compare match in the TRCGRB register	RW
		1 1 : Toggle output by compare match in the TRCGRB register	
IOB2	TRCGRB mode select bit <sup>(2)</sup>	Set to 0 (output compare) in the output compare function.	RW
— (b7)	Nothing is assigned. If necessary, set to 0. When read, the content is 1.		—

NOTES:

- When the BFC bit in the TRCMR register is set to 1 (buffer register of TRCGRA register), set the IOC2 bit in the TRCIOR1 register to the same value as the IOA2 bit in the TRCIOR0 register.
- When the BFD bit in the TRCMR register is set to 1 (buffer register of TRCGRB register), set the IOD2 bit in the TRCIOR1 register to the same value as the IOB2 bit in the TRCIOR0 register.

Figure 16.49 TRCIOR0 Register for Output Compare Function

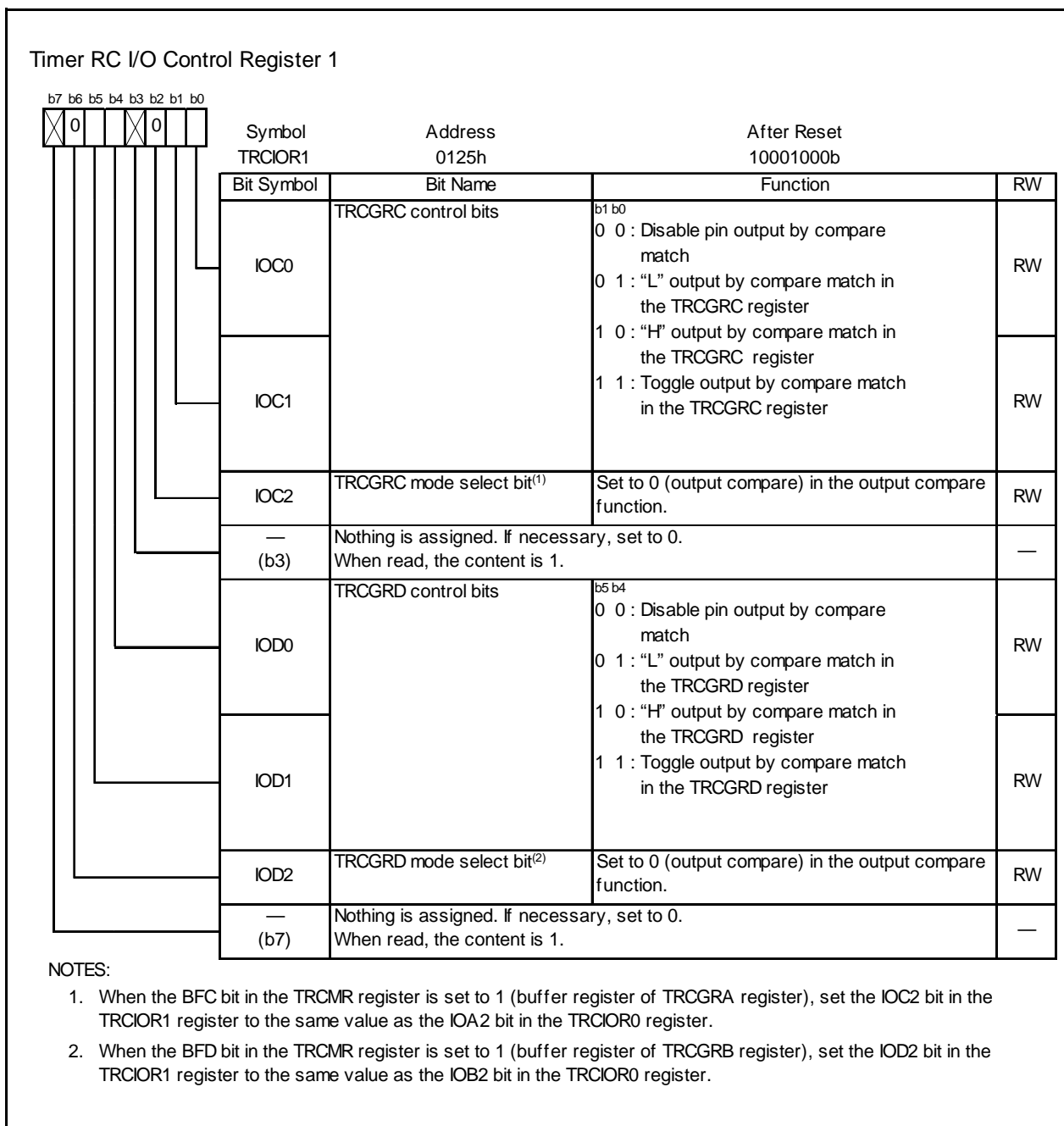


Figure 16.50 TRCIOR1 Register for Output Compare Function

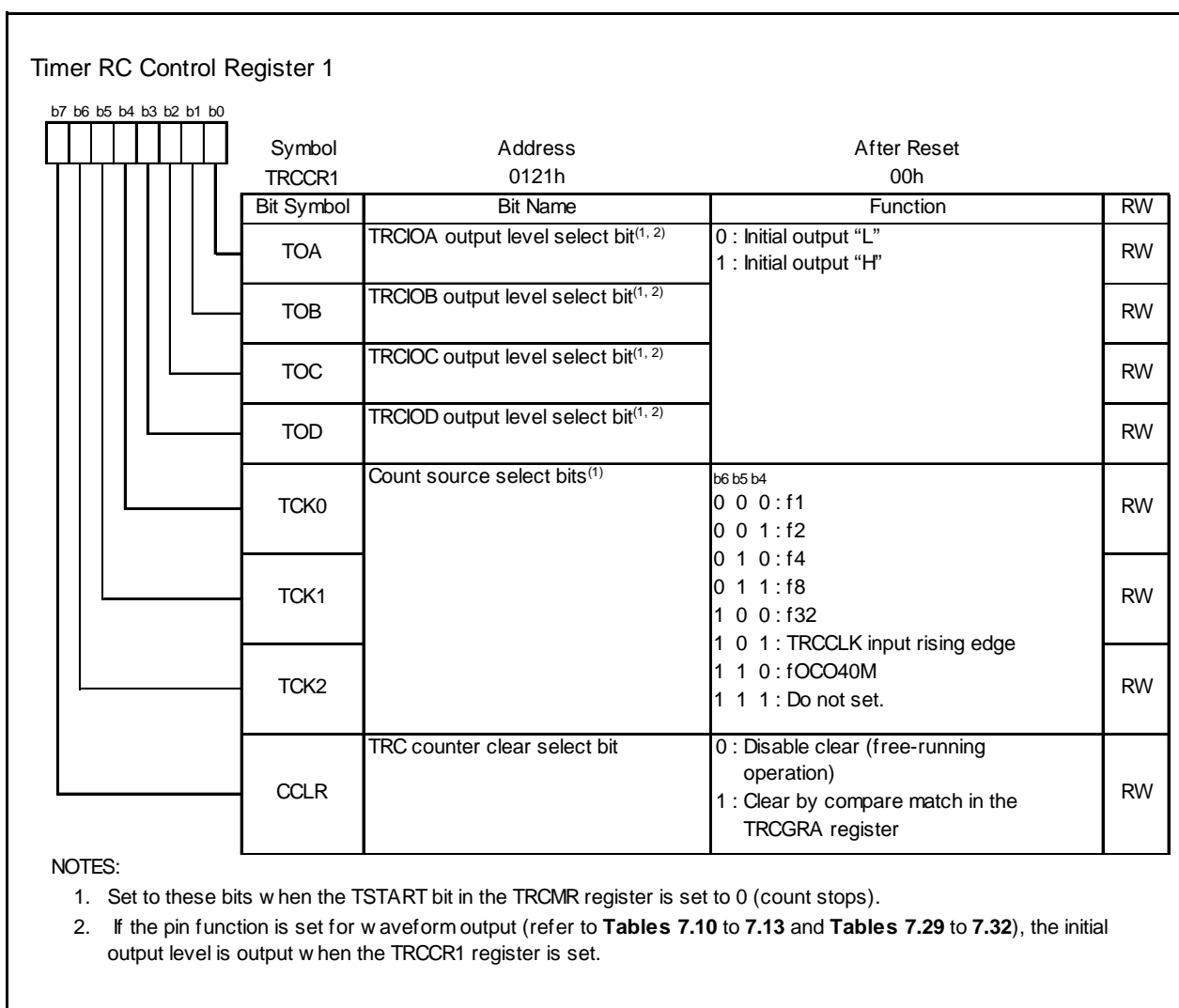


Figure 16.51 TRCCR1 Register for Output Compare Function

Table 16.19 Functions of TRCGRj Register when Using Output Compare Function

Register	Setting	Register Function	Output Compare Output Pin
TRCGRA	-	General register. Write a compare value to one of these registers.	TRCIOA
TRCGRB			TRCIOB
TRCGRC	BFC = 0	General register. Write a compare value to one of these registers.	TRCIOC
TRCGRD	BFD = 0		TRCIOD
TRCGRC	BFC = 1	Buffer register. Write the next compare value to one of these registers. (Refer to <b>16.3.3.2 Buffer Operation</b> .)	TRCIOA
TRCGRD	BFD = 1		TRCIOB

j = A, B, C, or D

BFC, BFD: Bits in TRCMR register

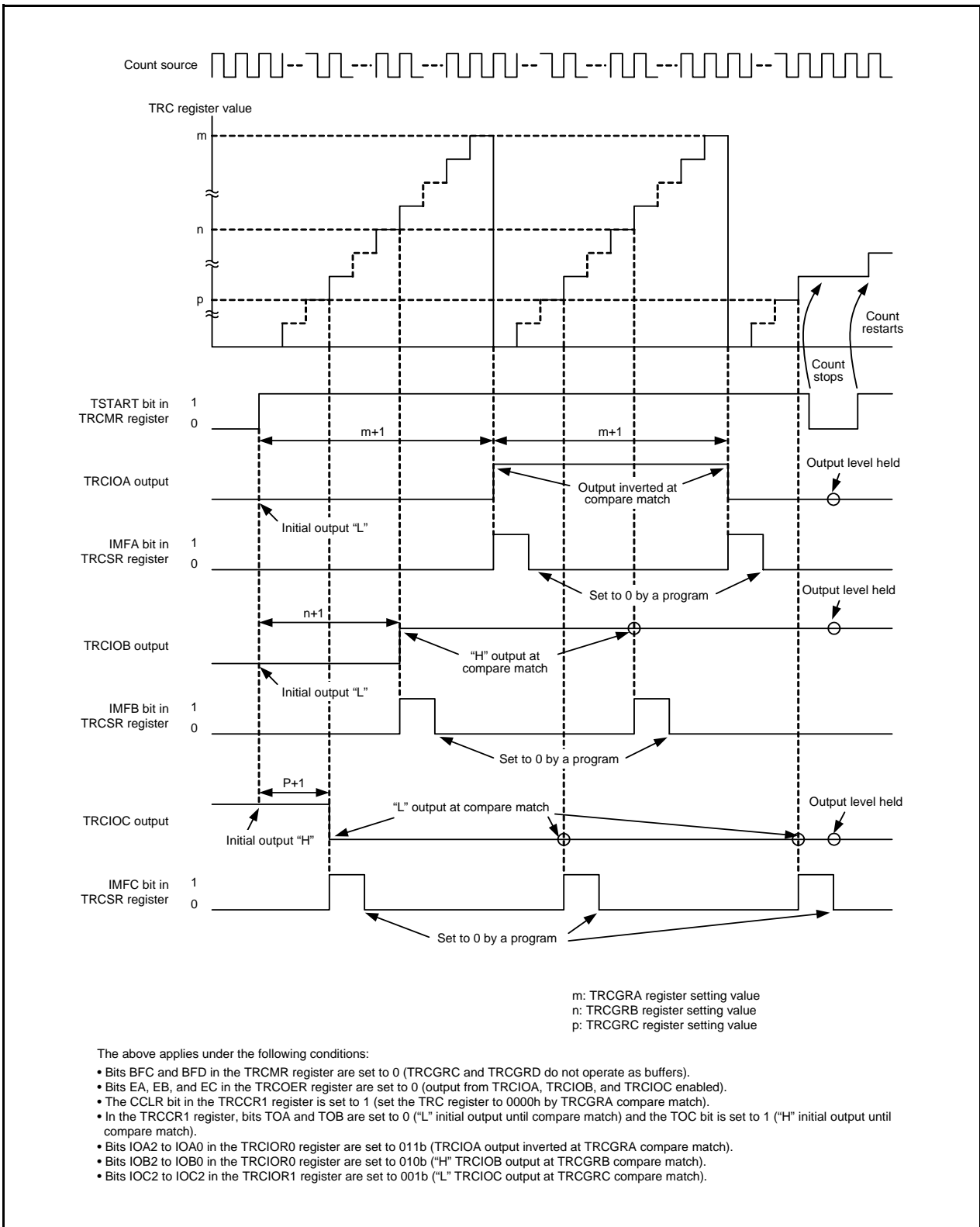


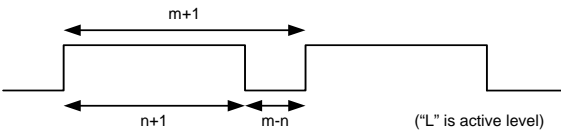
Figure 16.52 Operating Example of Output Compare Function

### 16.3.6 PWM Mode

This mode outputs PWM waveforms. A maximum of three PWM waveforms with the same period are output. The PWM mode, or the timer mode, can be selected for each individual pin. (However, since the TRCGRA register is used when using any pin for the PWM mode, the TRCGRA register cannot be used for the timer mode.)

Table 16.20 lists the Specifications of PWM Mode, Figure 16.53 shows a Block Diagram of PWM Mode, Figure 16.54 shows the registers associated with the PWM mode, Table 16.21 lists the Functions of TRCGRj Register in PWM Mode, and Figures 16.55 and 16.56 show Operating Examples of PWM Mode.

**Table 16.20 Specifications of PWM Mode**

Item	Specification
Count source	f1, f2, f4, f8, f32, fOCO40M, or external signal (rising edge) input to TRCCLK pin
Count operation	Increment
PWM waveform	PWM period: $1/f_k \times (m + 1)$ Active level width: $1/f_k \times (m - n)$ Inactive width: $1/f_k \times (n + 1)$ f <sub>k</sub> : Count source frequency m: TRCGRA register setting value n: TRCGR <sub>j</sub> register setting value 
Count start condition	1 (count starts) is written to the TSTART bit in the TRCMR register.
Count stop condition	0 (count stops) is written to the TSTART bit in the TRCMR register. PWM output pin retains output level before count stops, TRC register retains value before count stops.
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• Compare match (contents of registers TRC and TRCGR<sub>h</sub> match)</li> <li>• The TRC register overflows.</li> </ul>
TRCIOA pin function	Programmable I/O port
TRCIOB, TRCIOC, and TRCIOD pin functions	Programmable I/O port or PWM output (selectable individually by 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 TRC register.
Write to timer	The TRC register can be written to.
Select functions	<ul style="list-style-type: none"> <li>• One to three pins selectable as PWM output pins per channel One or more of pins TRCIOB, TRCIOC, and TRCIOD</li> <li>• Active level selectable by individual pin</li> <li>• Buffer operation (Refer to <b>16.3.3.2 Buffer Operation.</b>)</li> <li>• Pulse output forced cutoff signal input (Refer to <b>16.3.3.4 Forced Cutoff of Pulse Output.</b>)</li> </ul>

j = B, C, or D

h = A, B, C, or D

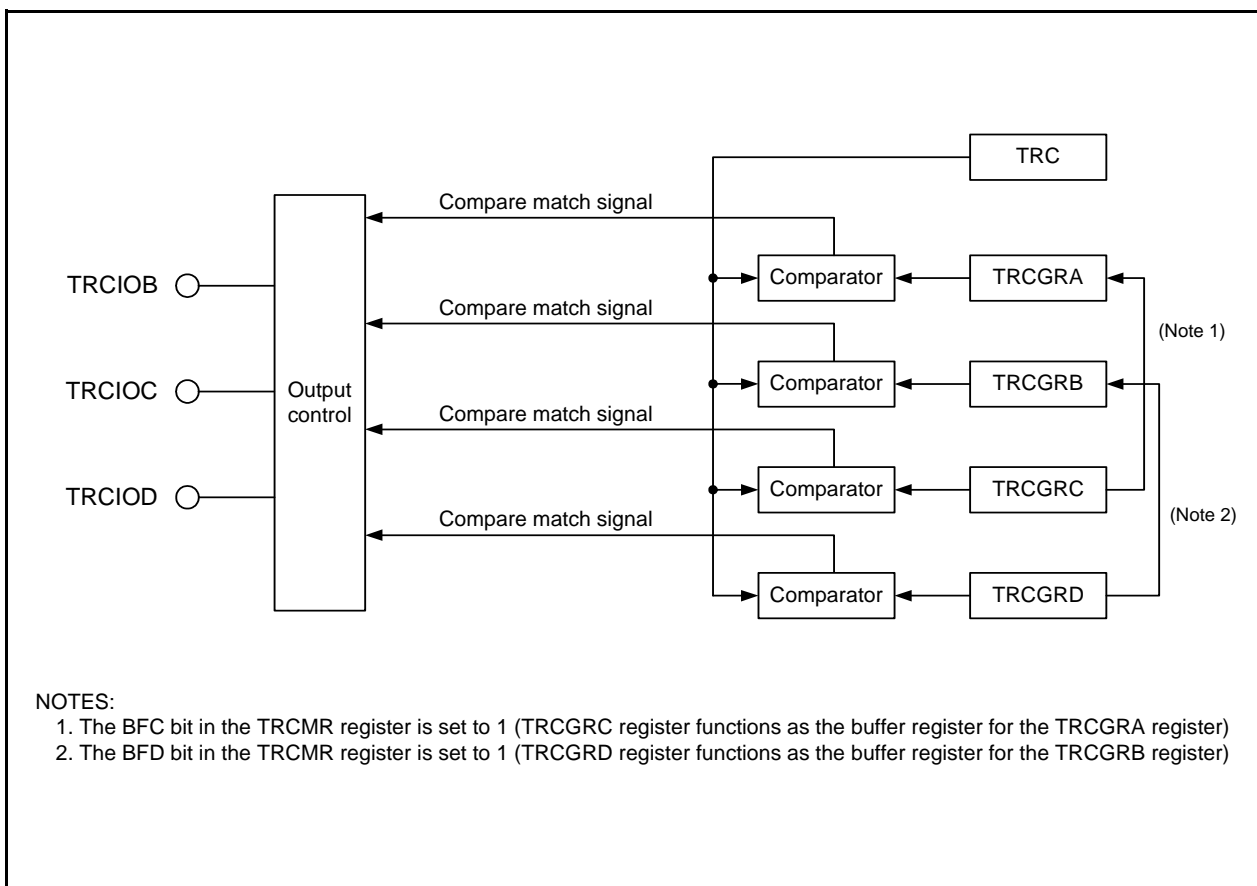


Figure 16.53 Block Diagram of PWM Mode

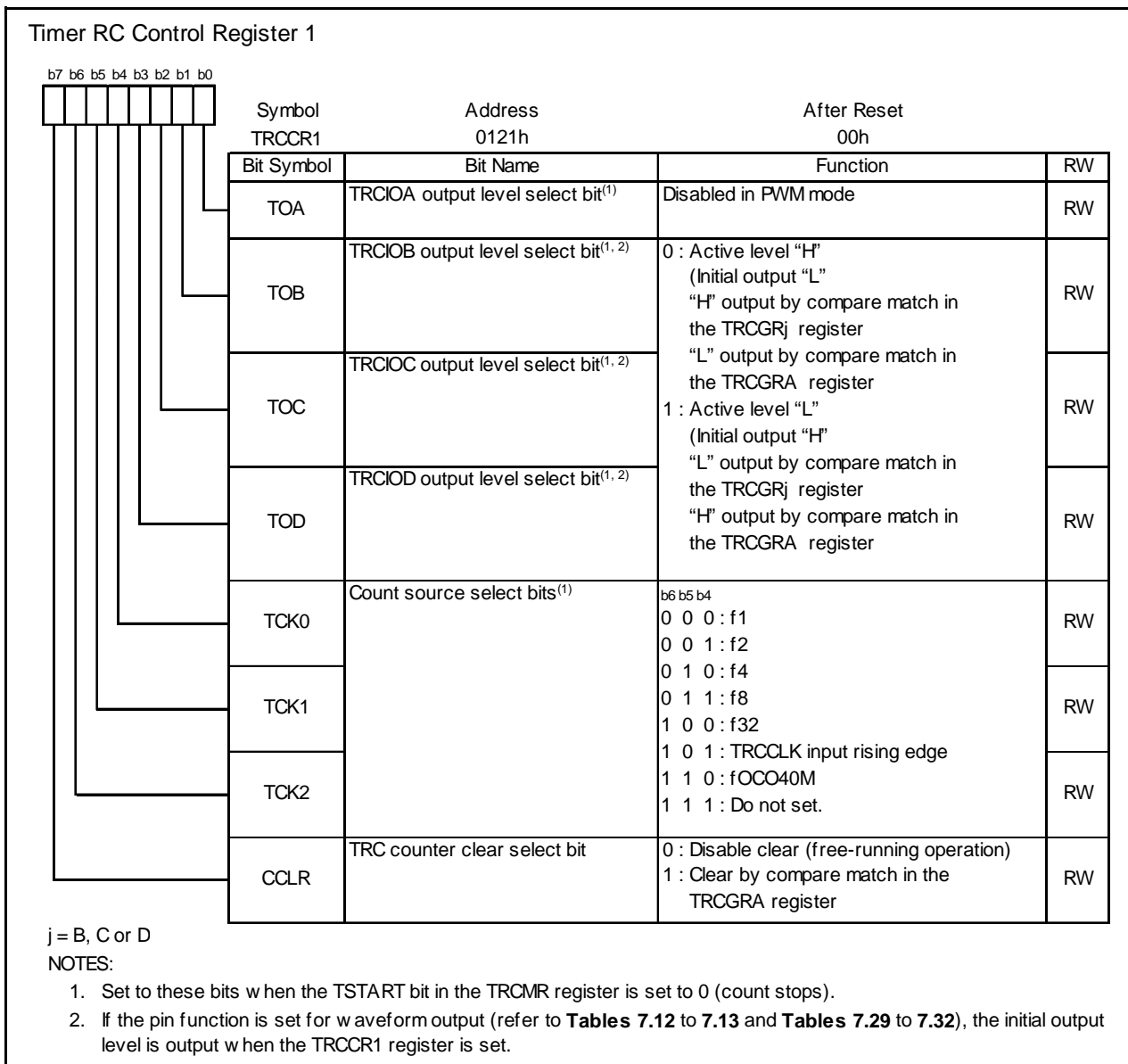


Figure 16.54 TRCCR1 Register in PWM Mode

Table 16.21 Functions of TRCGRj Register in PWM Mode

Register	Setting	Register Function	PWM Output Pin
TRCGRA	–	General register. Set the PWM period.	–
TRCGRB	–	General register. Set the PWM output change point.	TRCIOB
TRCGRC	BFC = 0	General register. Set the PWM output change point.	TRCIOC
TRCGRD	BFD = 0		TRCIOD
TRCGRC	BFC = 1	Buffer register. Set the next PWM period. (Refer to <b>16.3.3.2 Buffer Operation.</b> )	–
TRCGRD	BFD = 1	Buffer register. Set the next PWM output change point. (Refer to <b>16.3.3.2 Buffer Operation.</b> )	TRCIOB

j = A, B, C, or D

BFC, BFD: Bits in TRCMR register

NOTE:

- The output level does not change even when a compare match occurs if the TRCGRA register value (PWM period) is the same as the TRCGRB, TRCGRC, or TRCGRD register value.



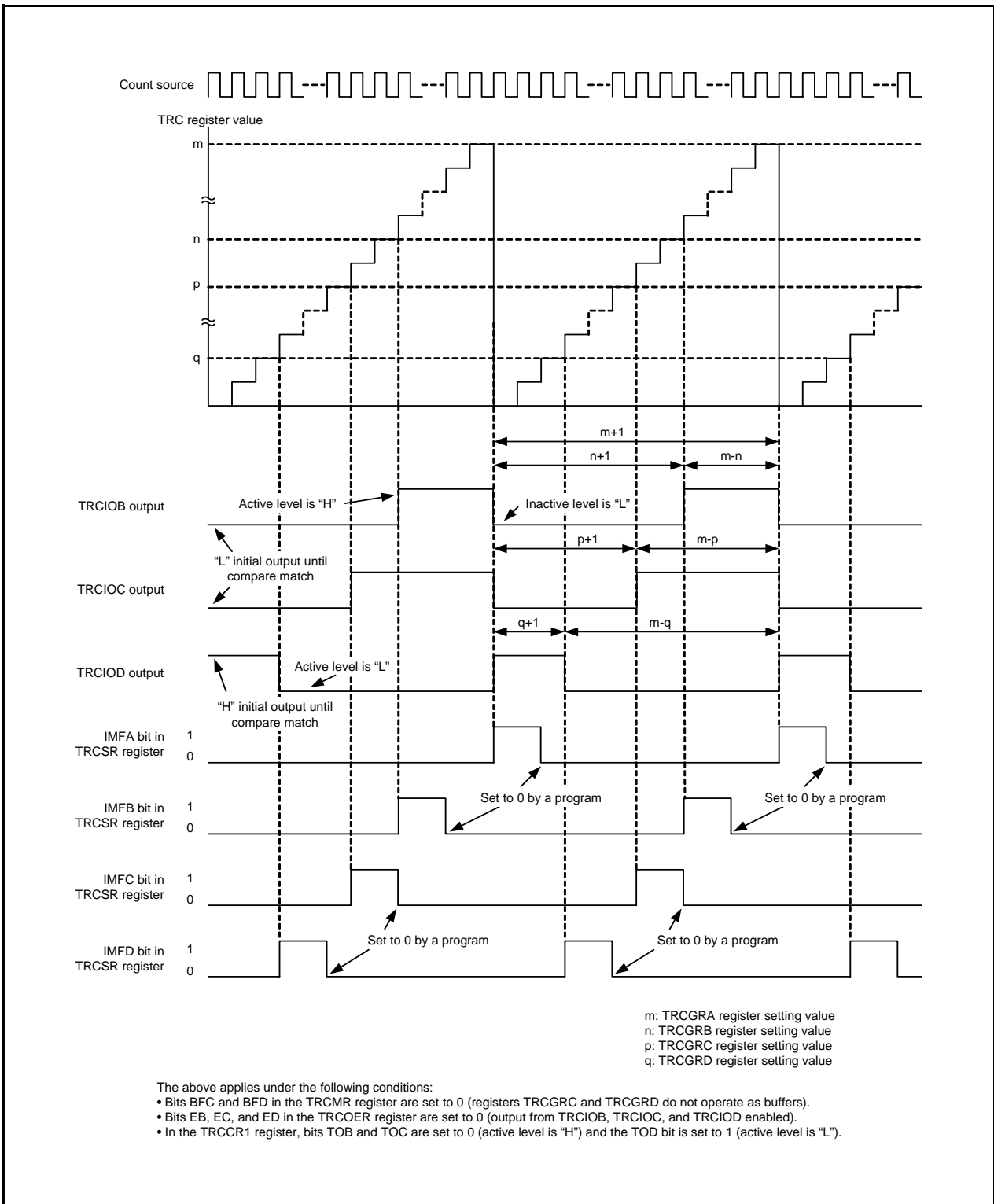


Figure 16.55 Operating Example of PWM Mode

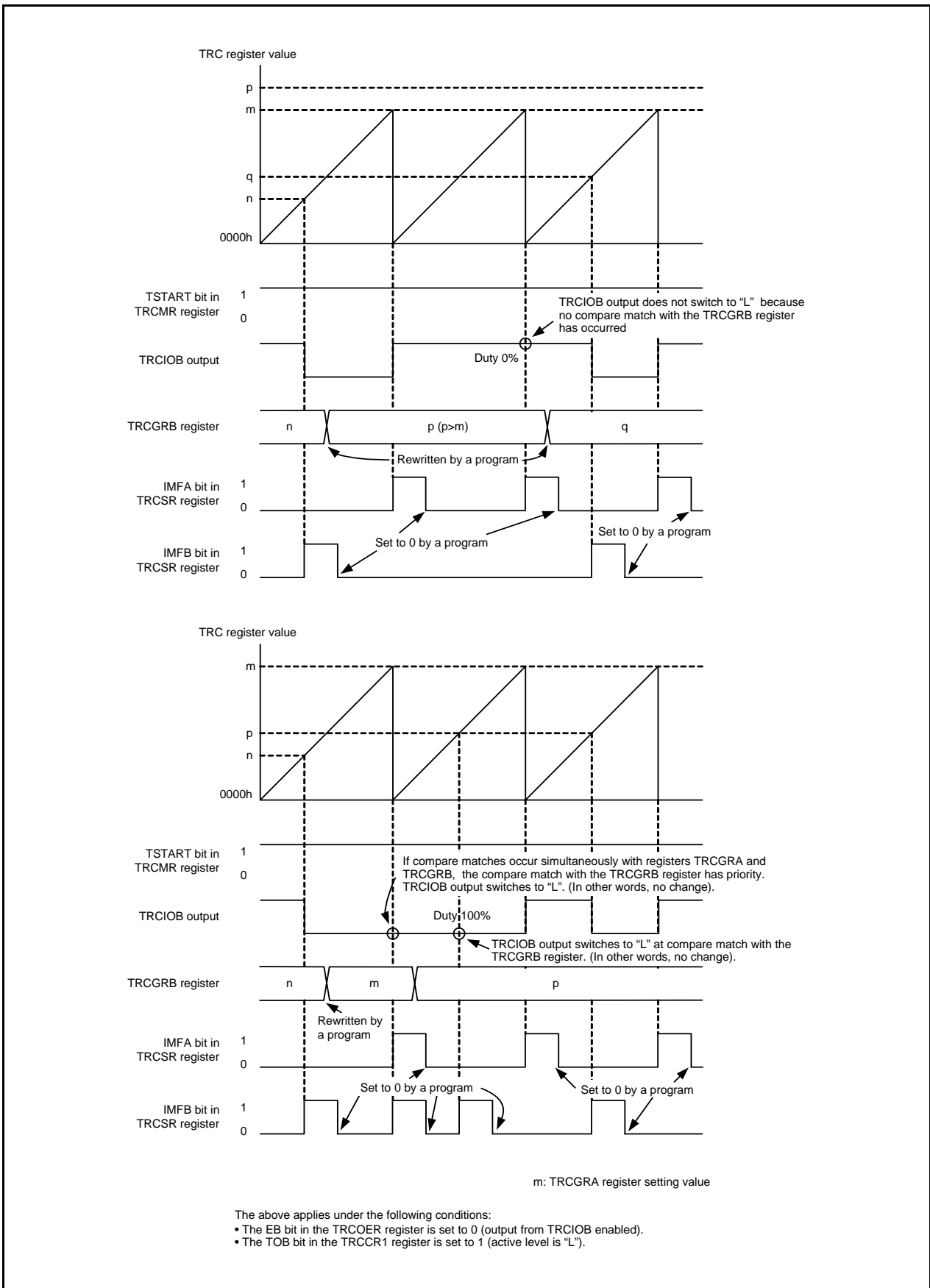


Figure 16.56 Operating Example of PWM Mode (Duty 0% and Duty 100%)

### 16.3.7 PWM2 Mode

This mode outputs a single PWM waveform. After a given wait duration has elapsed following the trigger, the pin output switches to active level. Then, after a given duration, the output switches back to inactive level. Furthermore, the counter stops at the same time the output returns to inactive level, making it possible to use PWM2 mode to output a programmable wait one-shot waveform.

Since timer RC uses multiple general registers in PWM2 mode, other modes cannot be used in conjunction with it.

Figure 16.57 shows a Block Diagram of PWM2 Mode, Table 16.22 lists the Specifications of PWM2 Mode, Figure 16.58 shows the register associated with PWM2 mode, Table 16.23 lists the Functions of TRCGRj Register in PWM2 Mode, and Figures 16.59 to 16.61 show Operating Examples of PWM2 Mode.

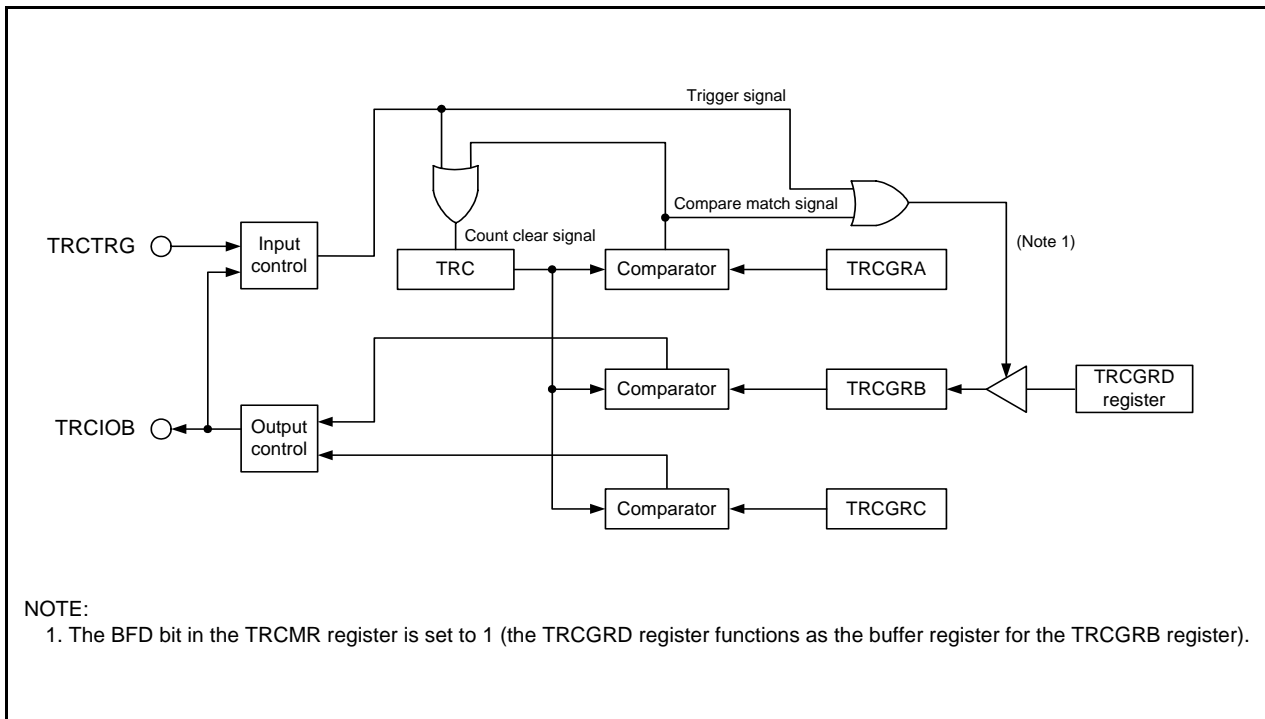


Figure 16.57 Block Diagram of PWM2 Mode

Table 16.22 Specifications of PWM2 Mode

Item	Specification
Count source	f1, f2, f4, f8, f32, fOCO40M, or external signal (rising edge) input to TRCCLK pin
Count operation	Increase TRC register
PWM waveform	<p>PWM period: <math>1/f_k \times (m + 1)</math> (no TRCTRГ input)            Active level width: <math>1/f_k \times (n - p)</math>            Wait time from count start or trigger: <math>1/f_k \times (p + 1)</math>  <math>f_k</math>: Count source frequency  <math>m</math>: TRCGRA register setting value  <math>n</math>: TRCGRB register setting value  <math>p</math>: TRCGRC register setting value</p> <p style="text-align: center;">(TRCTRГ: Rising edge, active level is "H")</p>
Count start conditions	<ul style="list-style-type: none"> <li>• Bits TCEG1 to TCEG0 in the TRCCR2 register are set to 00b (TRCTRГ trigger disabled) or the CSEL bit in the TRCCR2 register is set to 0 (count continues). 1 (count starts) is written to the TSTART bit in the TRCMR register.</li> <li>• Bits TCEG1 to TCEG0 in the TRCCR2 register are set to 01b, 10b, or 11b (TRCTRГ trigger enabled) and the TSTART bit in the TRCMR register is set to 1 (count starts). A trigger is input to the TRCTRГ pin</li> </ul>
Count stop conditions	<ul style="list-style-type: none"> <li>• 0 (count stops) is written to the TSTART bit in the TRCMR register while the CSEL bit in the TRCCR2 register is set to 0 or 1. The TRCIOB pin outputs the initial level in accordance with the value of the TOB bit in the TRCCR1 register. The TRC register retains the value before count stops.</li> <li>• The count stops due to a compare match with TRCGRA while the CSEL bit in the TRCCR2 register is set to 1. The TRCIOB pin outputs the initial level. The TRC register retains the value before count stops if the CCLR bit in the TRCCR1 register is set to 0. The TRC register is set to 0000h if the CCLR bit in the TRCCR1 register is set to 1.</li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• Compare match (contents of TRC and TRCGRj registers match)</li> <li>• The TRC register overflows</li> </ul>
TRCIOA/TRCTRГ pin function	Programmable I/O port or TRCTRГ input
TRCIOB pin function	PWM output
TRCIOC and TRCIOD 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 TRC register.
Write to timer	The TRC register can be written to.
Select functions	<ul style="list-style-type: none"> <li>• External trigger and valid edge selected The edge or edges of the signal input to the TRCTRГ pin can be used as the PWM output trigger: rising edge, falling edge, or both rising and falling edges</li> <li>• Buffer operation (Refer to <b>16.3.3.2 Buffer Operation.</b>)</li> <li>• Pulse output forced cutoff signal input (Refer to <b>16.3.3.4 Forced Cutoff of Pulse Output.</b>)</li> <li>• Digital filter (Refer to <b>16.3.3.3 Digital Filter.</b>)</li> </ul>

j = A, B, or C

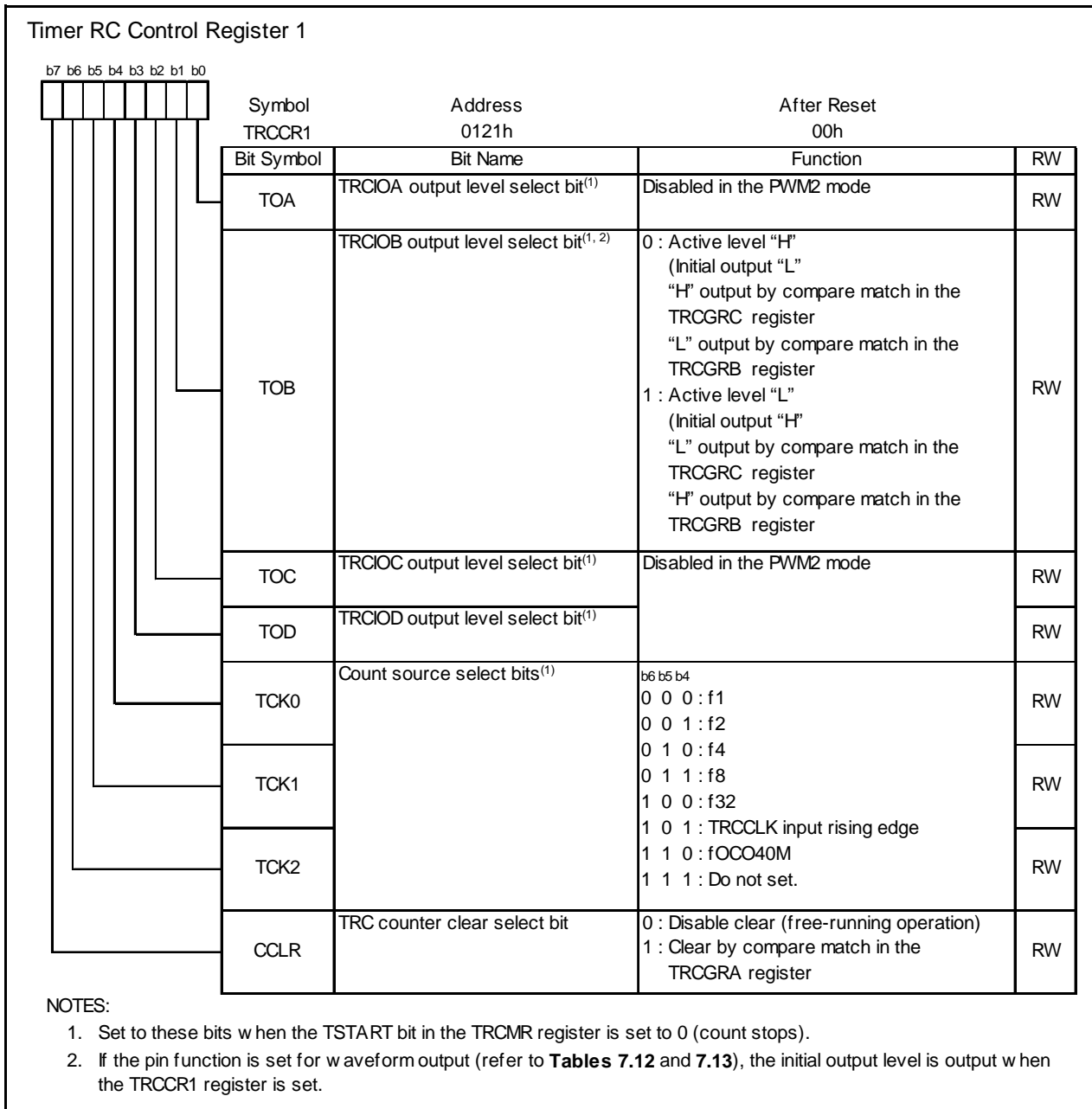


Figure 16.58 TRCCR1 Register in PWM2 Mode

Table 16.23 Functions of TRCGRj Register in PWM2 Mode

Register	Setting	Register Function	PWM2 Output Pin
TRCGRA	–	General register. Set the PWM period.	TRCIOB pin
TRCGRB	–	General register. Set the PWM output change point.	
TRCGRC	BFC = 0	General register. Set the PWM output change point (wait time after trigger).	
TRCGRD	BFD = 0	(Not used in PWM2 mode)	–
TRCGRD	BFD = 1	Buffer register. Set the next PWM output change point. (Refer to <b>16.3.3.2 Buffer Operation.</b> )	TRCIOB pin

j = A, B, C, or D

BFC, BFD: Bits in TRCMR register

NOTE:

- Do not set the TRCGRB and TRCGRC registers to the same value.

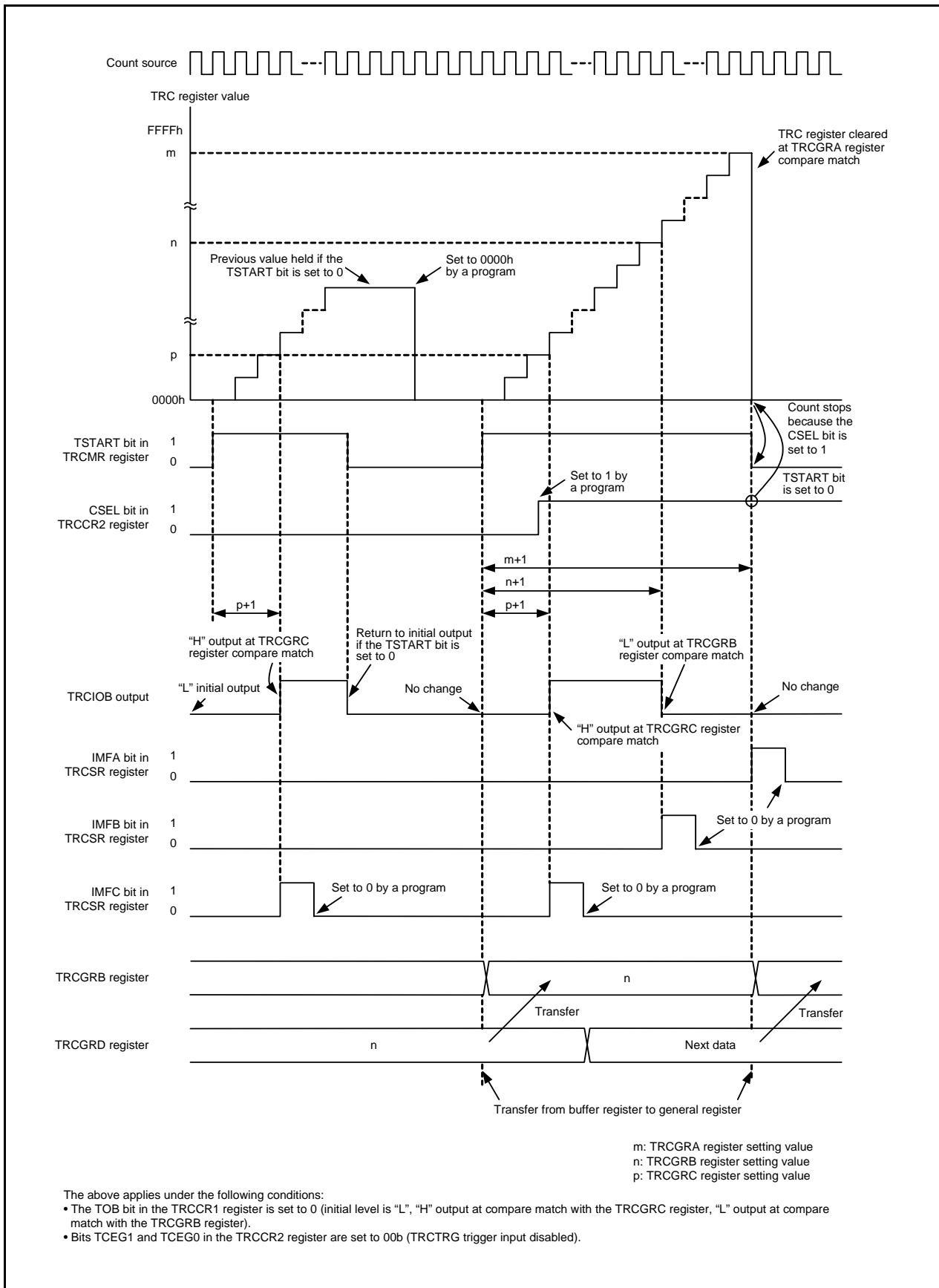


Figure 16.59 Operating Example of PWM2 Mode (TRCTRIG Trigger Input Disabled)

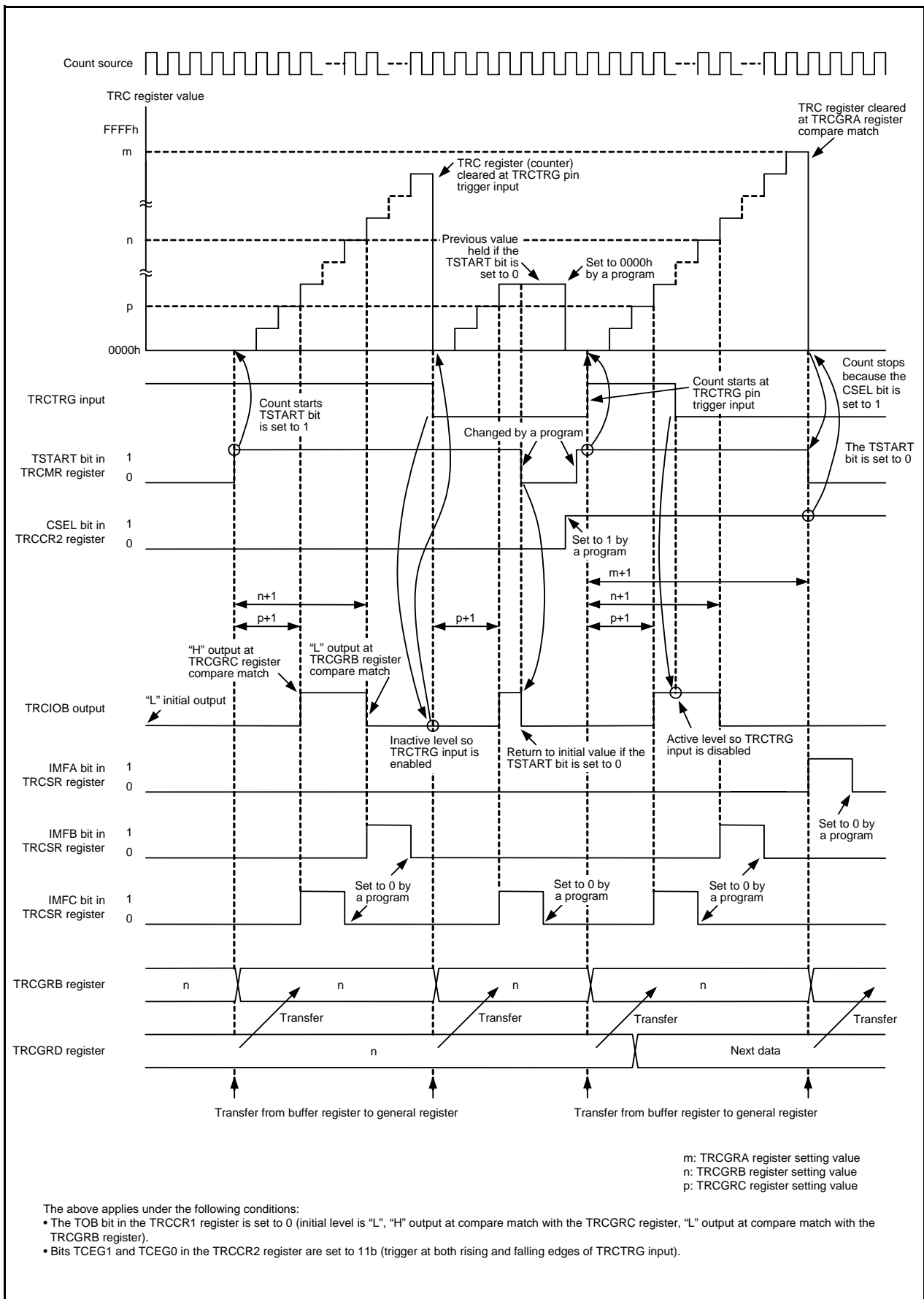


Figure 16.60 Operating Example of PWM2 Mode (TRCTRГ Trigger Input Enabled)

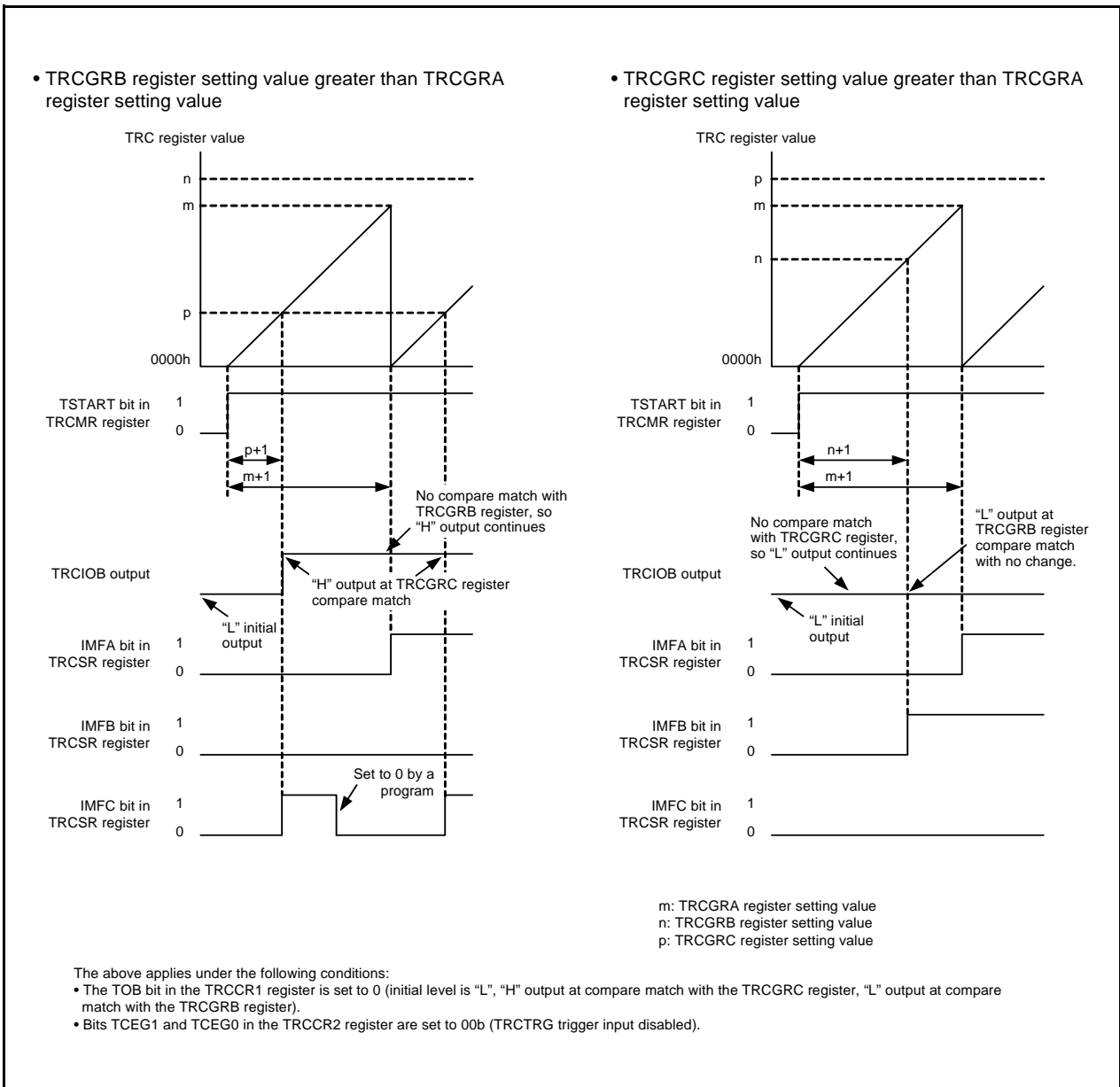


Figure 16.61 Operating Example of PWM2 Mode (Duty 0% and Duty 100%)



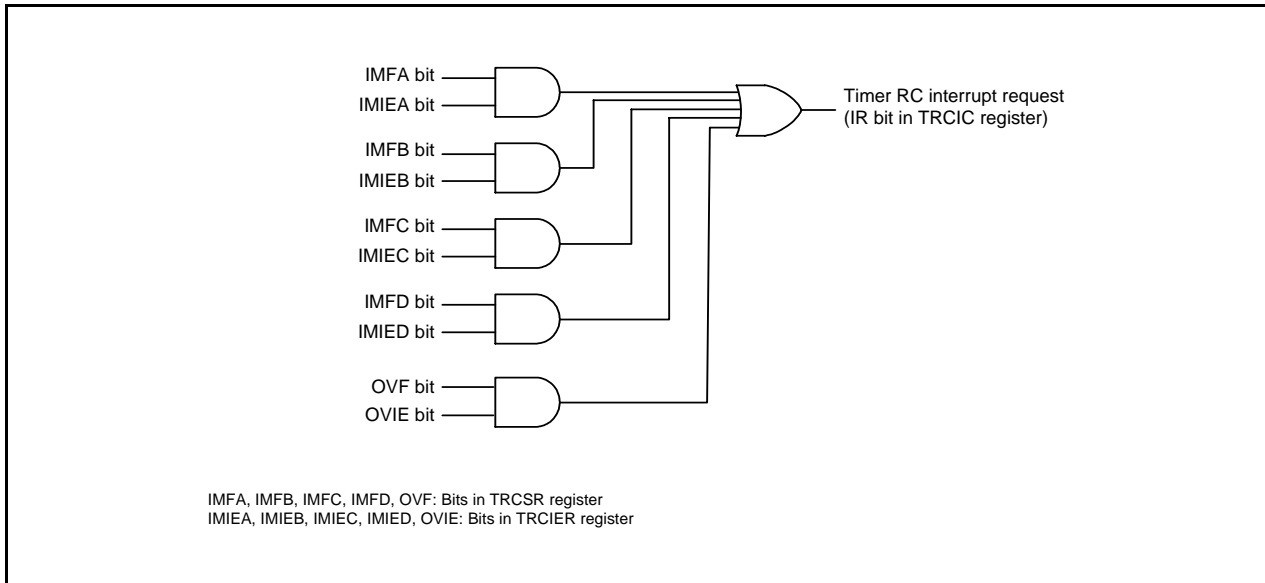
### 16.3.8 Timer RC Interrupt

Timer RC generates a timer RC interrupt request from five sources. The timer RC interrupt uses the single TRCIC register (bits IR and ILVL0 to ILVL2) and a single vector.

Table 16.24 lists the Registers Associated with Timer RC Interrupt, and Figure 16.62 is a Timer RC Interrupt Block Diagram.

**Table 16.24 Registers Associated with Timer RC Interrupt**

Timer RC Status Register	Timer RC Interrupt Enable Register	Timer RC Interrupt Control Register
TRCSR	TRCIER	TRCIC



**Figure 16.62 Timer RC Interrupt Block Diagram**

Like other maskable interrupts, the timer RC interrupt is controlled by the combination of the I flag, IR bit, bits ILVL0 to ILVL2, and IPL. However, it differs from other maskable interrupts in the following respects because a single interrupt source (timer RC interrupt) is generated from multiple interrupt request sources.

- The IR bit in the TRCIC register is set to 1 (interrupt requested) when a bit in the TRCSR register is set to 1 and the corresponding bit in the TRCIER register is also set to 1 (interrupt enabled).
- The IR bit is set to 0 (no interrupt request) when the bit in the TRCSR register or the corresponding bit in the TRCIER register is set to 0, or both are set to 0. In other words, the interrupt request is not maintained if the IR bit is once set to 1 but the interrupt is not acknowledged.
- If after the IR bit is set to 1 another interrupt source is triggered, the IR bit remains set to 1 and does not change.
- If multiple bits in the TRCIER register are set to 1, use the TRCSR register to determine the source of the interrupt request.
- The bits in the TRCSR register are not automatically set to 0 when an interrupt is acknowledged. Set them to 0 within the interrupt routine. Refer to **Figure 16.32 TRCSR Register**, for the procedure for setting these bits to 0.

Refer to **Figure 16.31 TRCIER Register**, for details of the TRCIER register.

Refer to **12.1.6 Interrupt Control**, for details of the TRCIC register and **12.1.5.2 Relocatable Vector Tables**, for information on interrupt vectors.

## 16.3.9 Notes on Timer RC

### 16.3.9.1 TRC Register

- The following note applies when the CCLR bit in the TRCCR1 register is set to 1 (clear TRC register at compare match with TRCGRA register).

When using a program to write a value to the TRC register while the TSTART bit in the TRCMR register is set to 1 (count starts), ensure that the write does not overlap with the timing with which the TRC register is set to 0000h.

If the timing of the write to the TRC register and the setting of the TRC register to 0000h coincide, the write value will not be written to the TRC register and the TRC register will be set to 0000h.

- Reading from the TRC register immediately after writing to it can result in the value previous to the write being read out. To prevent this, execute the JMP.B instruction between the read and the write instructions.

```

Program Example      MOV.W      #XXXXh, TRC          ;Write
                    JMP.B      L1              ;JMP.B instruction
                    L1:         MOV.W      TRC,DATA        ;Read
  
```

### 16.3.9.2 TRCSR Register

Reading from the TRCSR register immediately after writing to it can result in the value previous to the write being read out. To prevent this, execute the JMP.B instruction between the read and the write instructions.

```

Program Example      MOV.B      #XXh, TRCSR        ;Write
                    JMP.B      L1              ;JMP.B instruction
                    L1:         MOV.B      TRCSR,DATA    ;Read
  
```

### 16.3.9.3 Count Source Switching

- Stop the count before switching the count source.

Switching procedure

- Set the TSTART bit in the TRCMR register to 0 (count stops).
- Change the settings of bits TCK2 to TCK0 in the TRCCR1 register.

- After switching the count source from fOCO40M to another clock, allow a minimum of two cycles of f1 to elapse after changing the clock setting before stopping fOCO40M.

Switching procedure

- Set the TSTART bit in the TRCMR register to 0 (count stops).
- Change the settings of bits TCK2 to TCK0 in the TRCCR1 register.
- Wait for a minimum of two cycles of f1.
- Set the FRA00 bit in the FRA0 register to 0 (high-speed on-chip oscillator off).

### 16.3.9.4 Input Capture Function

- The pulse width of the input capture signal should be three cycles or more of the timer RC operation clock (refer to **Table 16.11 Timer RC Operation Clock**).
- The value of the TRC register is transferred to the TRCGRj register one or two cycles of the timer RC operation clock after the input capture signal is input to the TRCIOj (j = A, B, C, or D) pin (when the digital filter function is not used).

### 16.3.9.5 TRCMR Register in PWM2 Mode

When the CSEL bit in the TRCCR2 register is set to 1 (count stops at compare match with the TRCGRA register), do not set the TRCMR register at compare match timing of registers TRC and TRCGRA.

## 16.4 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 16.25 lists the Timer RD Operation Clocks.

**Table 16.25 Timer RD Operation Clocks**

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

Figure 16.63 shows a Block Diagram of Timer RD. Timer RD has 5 modes:

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

The following 4 modes use the output compare function.

- PWM mode    Output pulse of any width 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 waveforms (2) with a fixed 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 individually for each 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 16.26 to 16.34 list the Pin Functions of timer RD.

**Table 16.26 Pin Functions TRDIOA0/TRDCLK(P2\_0)**

Register	TRDOER1	TRDFCR			TRDIORA0		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						I/O port

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 16.27 Pin Functions TRDIOB0(P2\_1)**

Register	TRDOER1	TRDFCR		TRDPMR	TRDIORA0	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					I/O port

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 16.28 Pin Functions TRDIOC0(P2\_2)**

Register	TRDOER1	TRDFCR		TRDPMR	TRDIORC0	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					I/O port

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 16.29 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 16.30 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 16.31 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 16.32 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 16.33 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 16.34 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

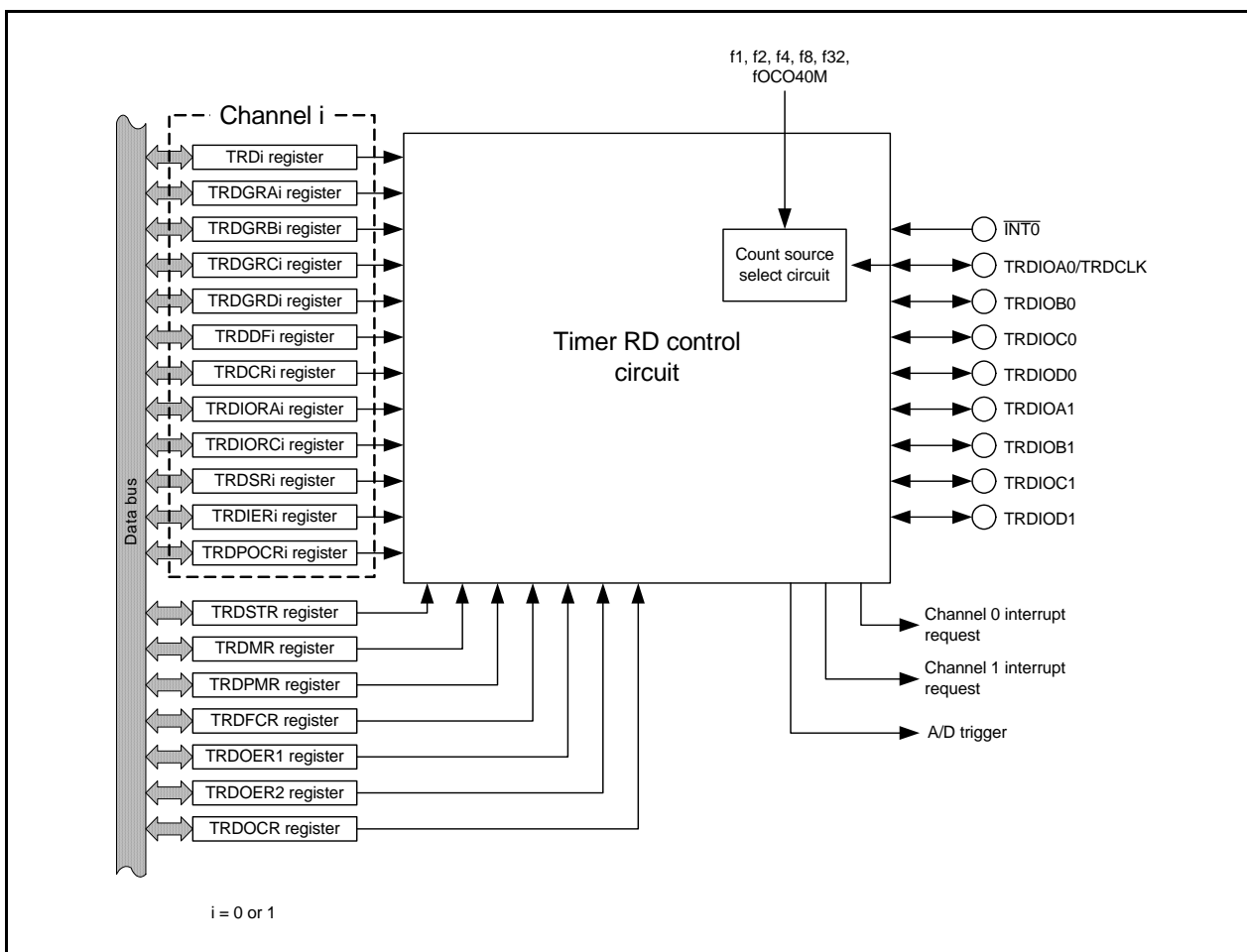


Figure 16.63 Block Diagram of Timer RD

### 16.4.1 Count Sources

The count source selection method is the same in all modes. However, in PWM3 mode, the external clock cannot be selected.

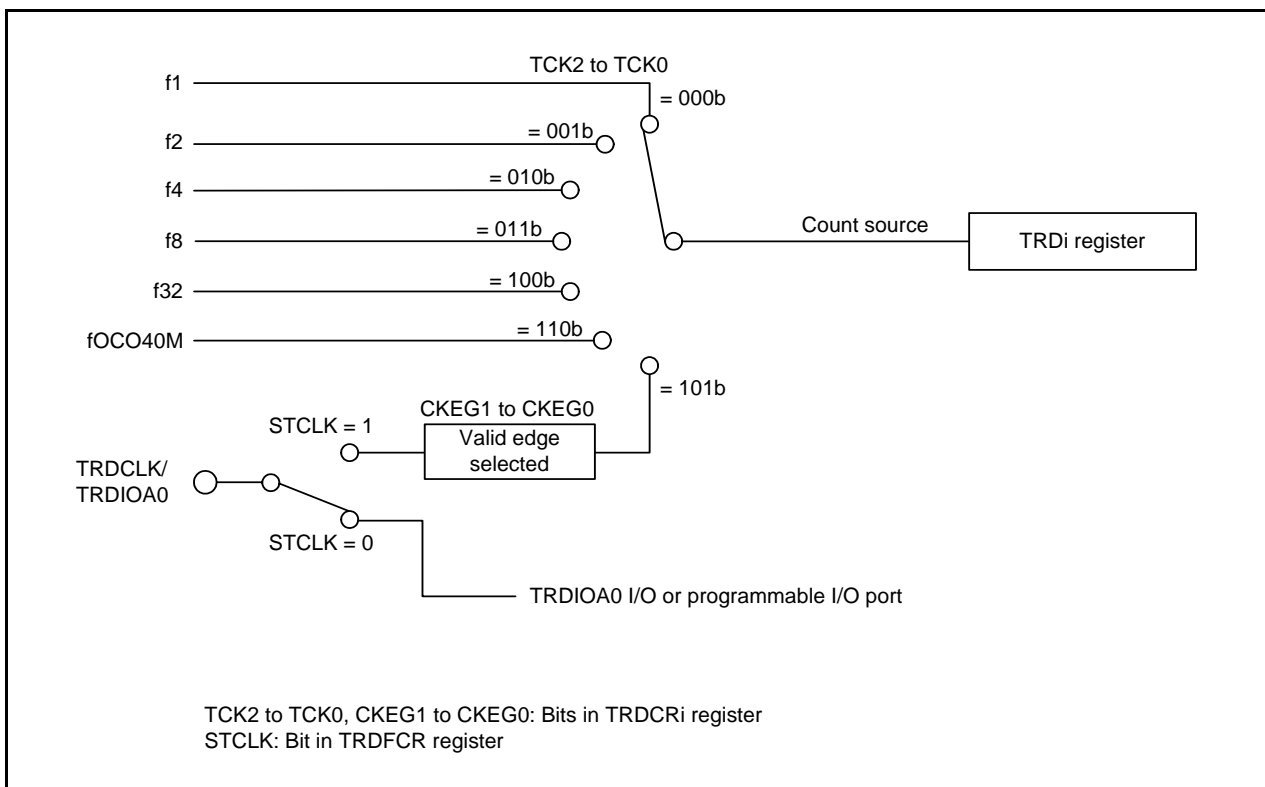
**Table 16.35 Count Source Selection**

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). Bits TCK2 to TCK0 in the TRDCR <sub>i</sub> register are set to 101b (count source: external clock). The valid edge is selected by bits CKEG1 to CKEG0 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 16.64 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 16.25 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 bits TCK2 to TCK0 in the TRDCR<sub>i</sub> register (i = 0 or 1) to 110b (fOCO40M).



### 16.4.2 Buffer Operation

The TRDGRC<sub>i</sub> ( $i = 0$  or  $1$ ) 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 means of bits BFC<sub>i</sub> and BFD<sub>i</sub> in the TRDMR register.

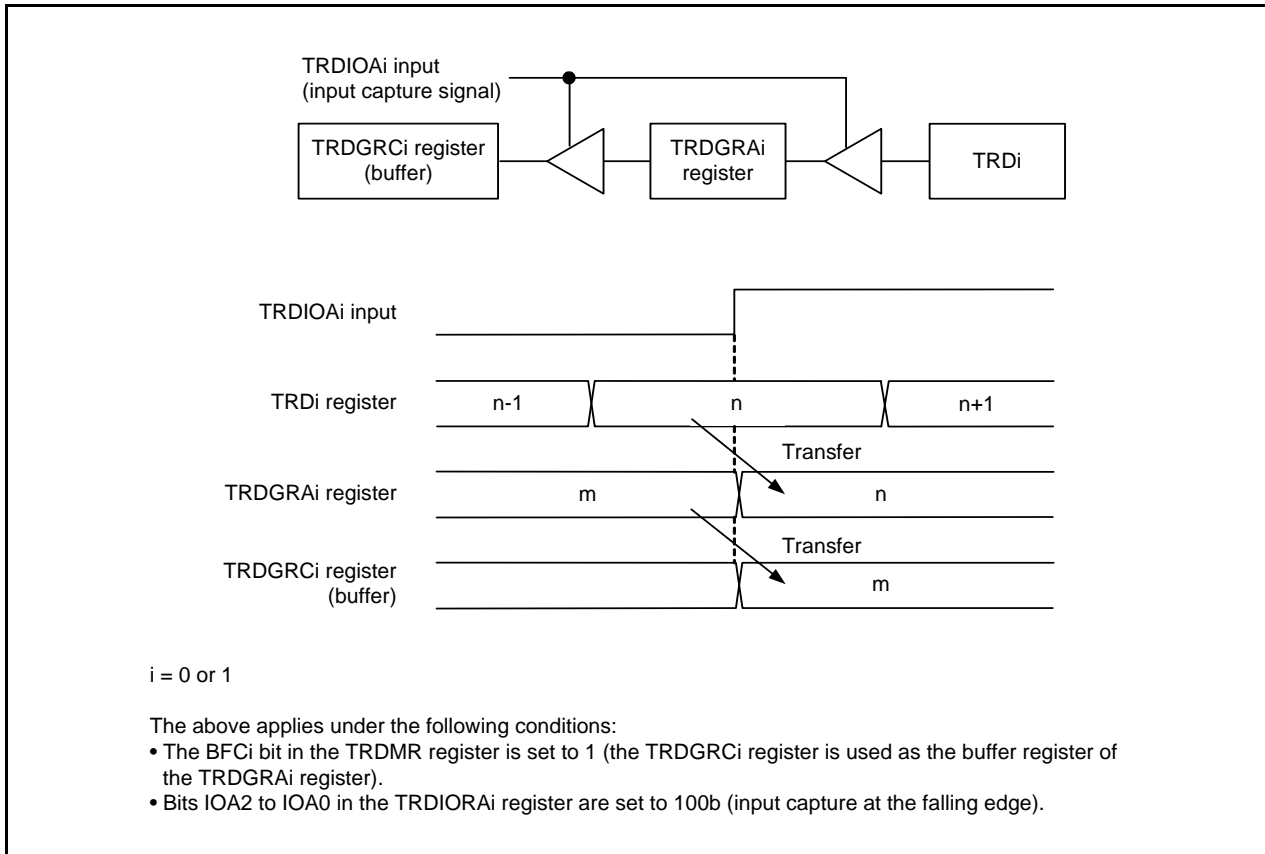
- TRDGRA<sub>i</sub> buffer register: TRDGRC<sub>i</sub> register
- TRDGRB<sub>i</sub> buffer register: TRDGRD<sub>i</sub> register

Buffer operation depends on the mode. Table 16.36 lists the Buffer Operation in Each Mode.

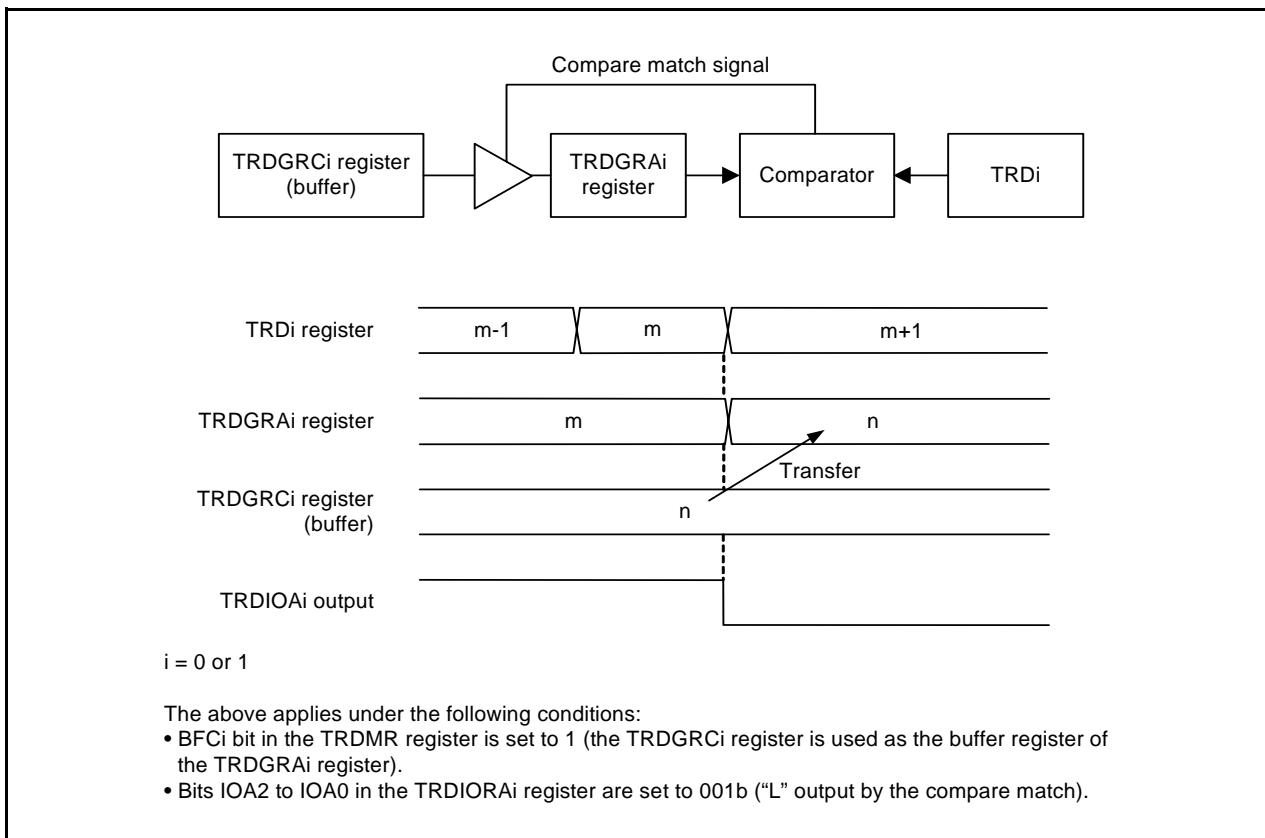
**Table 16.36 Buffer Operation in Each Mode**

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		
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 registers TRDGRB <sub>0</sub> , TRDGRA <sub>1</sub> , and TRDGRB <sub>1</sub>
PWM3 mode	Compare match with TRD <sub>0</sub> register and TRDGRA <sub>0</sub> register	Transfer content in buffer register to registers TRDGRA <sub>0</sub> , TRDGRB <sub>0</sub> , TRDGRA <sub>1</sub> , and TRDGRB <sub>1</sub>

$i = 0$  or  $1$



**Figure 16.65 Buffer Operation in Input Capture Function**



**Figure 16.66 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 value 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 TRDIORDi 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 also using registers TRDGRCi and TRDGRDi as buffer registers for the output compare function, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode, bits IMFC and IMFD in the TRDSRi register are set to 1 by a compare match with the TRDi register.

### 16.4.3 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 bits CCLR2 to CCLR0 in the TRDCRi register are set to 011b (synchronous clear), 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 bits CCLR2 to CCLR0 in the TRDCRi register are set to 011b (synchronous clear), the TRD1 register is set to 0000h at the same time as the TRD0 register is set to 0000h.

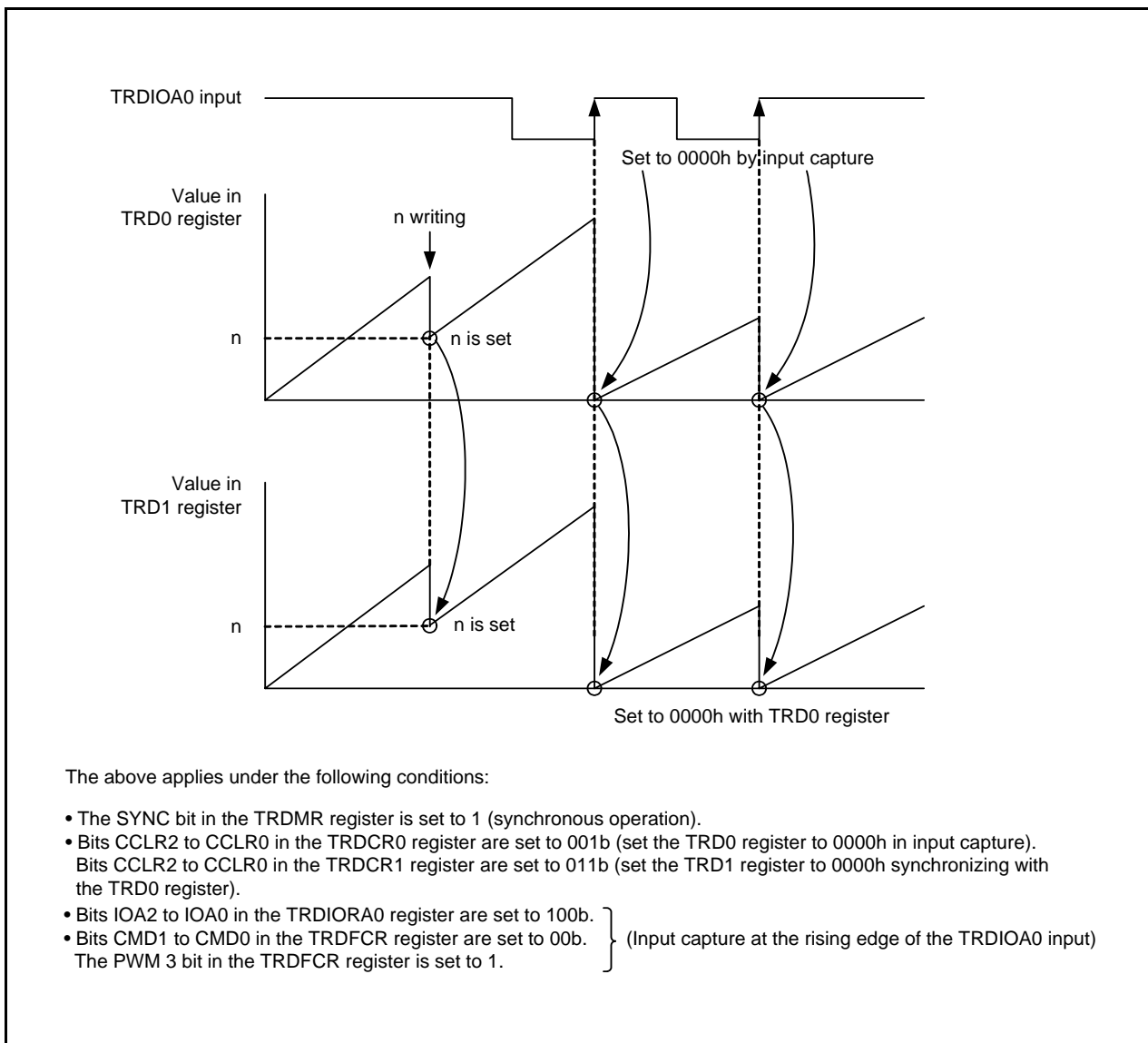


Figure 16.67 Synchronous Operation

#### 16.4.4 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> (i = 0 or 1, j = either A, B, C, or D) output pin can be forcibly set to a programmable I/O port by the  $\overline{\text{INT0}}$  pin input, and pulse output can be cut off.

The pins used for output in these functions or modes can function as the output pin of timer RD when the applicable bit in the TRDOER1 register is set to 0 (enable timer RD output). When the PTO bit in the TRDOER2 register is set 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 16.25 Timer RD Operation Clocks**).

Set as below when using this function:

- Set the pin status (high impedance, “L” or “H” output) to pulse output forced cutoff by registers P2 and PD2.
- 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 bits INT0F1 to INT0F0 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.

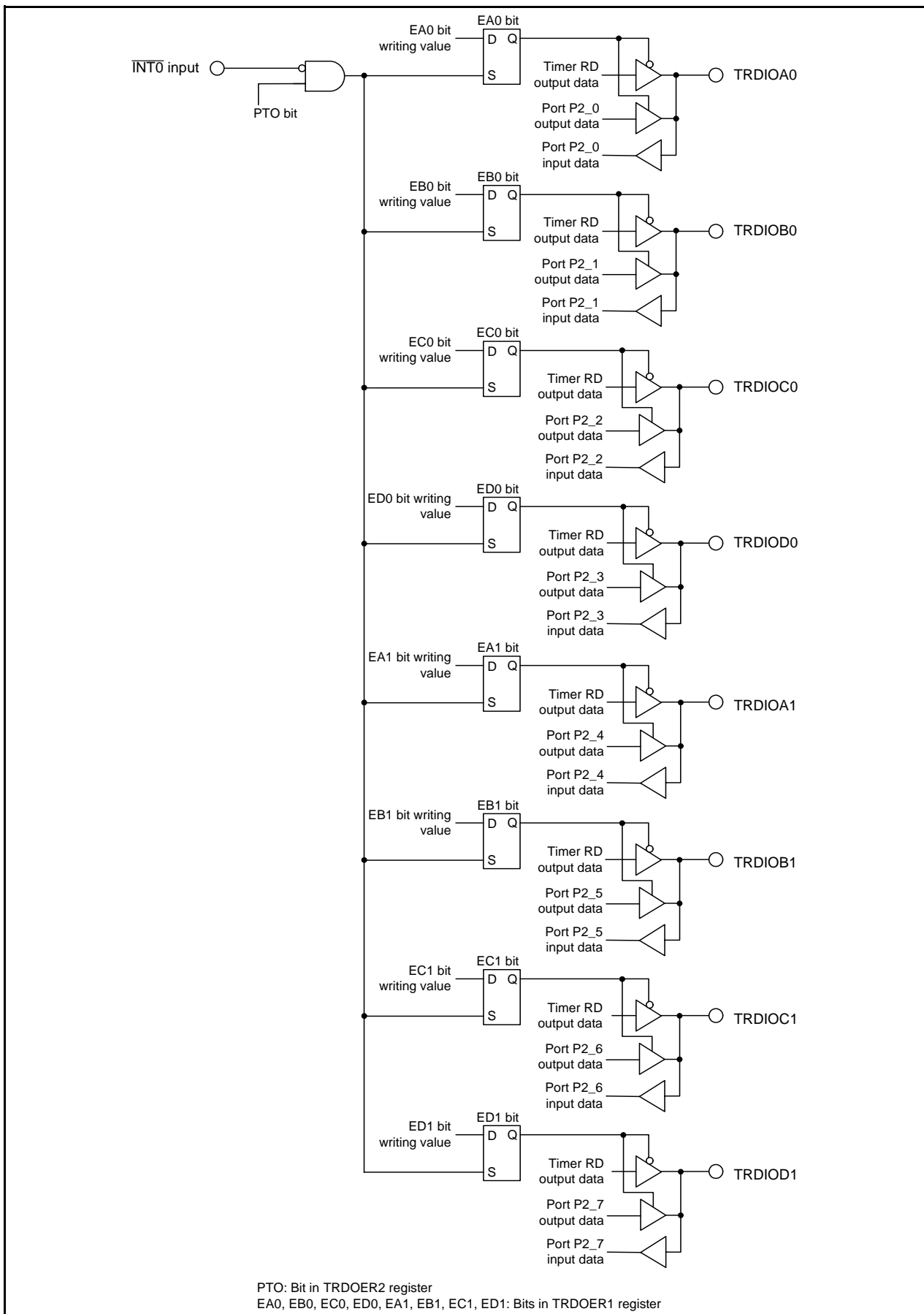


Figure 16.68 Pulse Output Forced Cutoff

### 16.4.5 Input Capture Function

The input capture function measures the external signal width and period. The content of 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, the input capture function, or any other mode or function, can be selected for each individual pin.

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

Figure 16.69 shows a Block Diagram of Input Capture Function, Table 16.37 lists the Input Capture Function Specifications. Figures 16.70 to 16.80 show the Registers Associated with Input Capture Function, and Figure 16.81 shows an Operating Example of Input Capture Function.

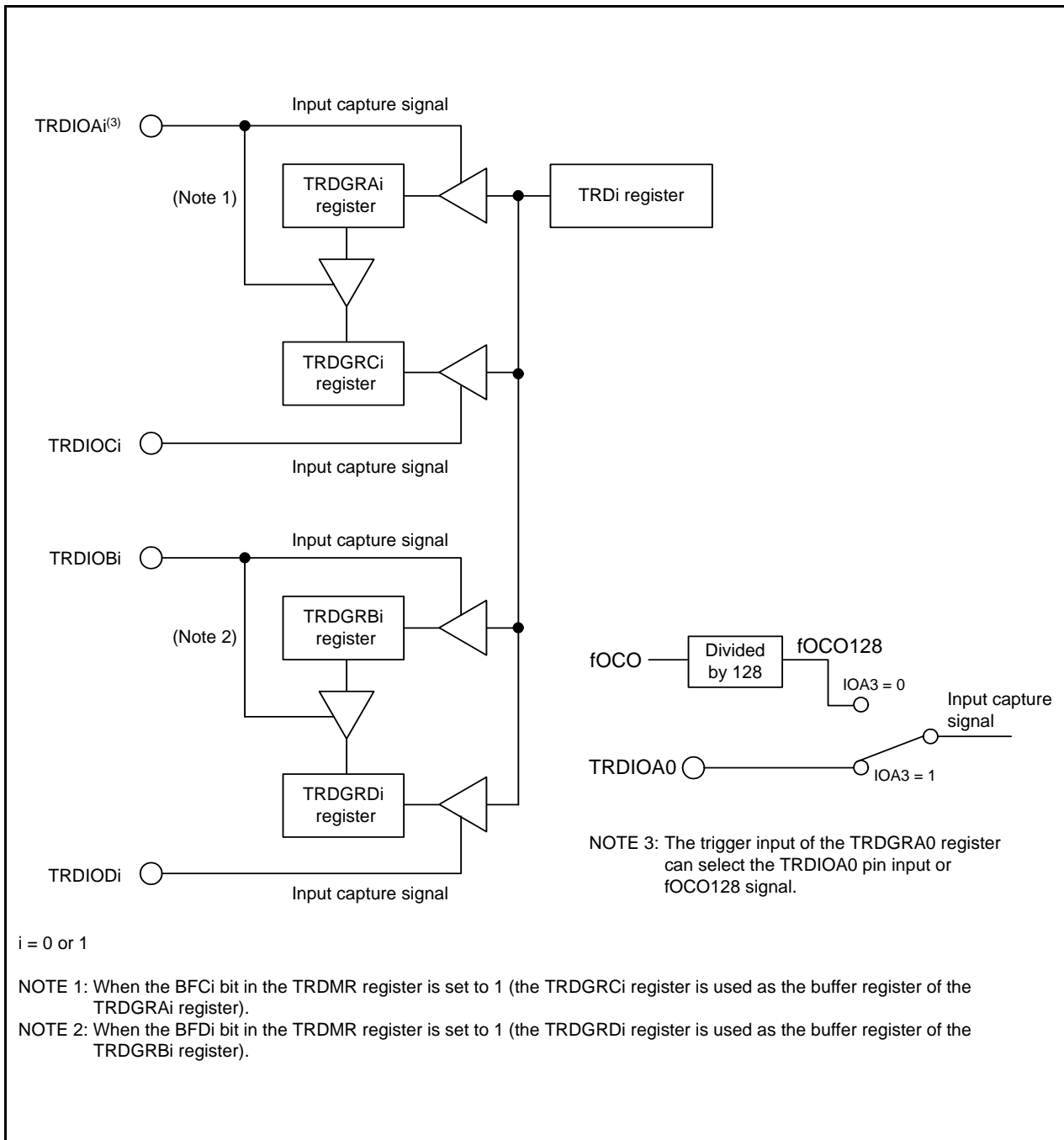


Figure 16.69 Block Diagram of Input Capture Function

**Table 16.37 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 bits CCLR2 to CCLR0 in the TRDCRi register are set to 000b (free-running operation). $1/fk \times 65536$ fk: Frequency of count source
Count start condition	1 (count starts) is written to the TSTARTi bit in the TRDSTR register.
Count stop condition	0 (count stops) is written 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 (selectable by 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). 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>
Select functions	<ul style="list-style-type: none"> <li>Input-capture input pin selected Either 1 pin or multiple pins among TRDIOAi, TRDIOBi, TRDIOCi, or TRDIODi.</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>16.4.2 Buffer Operation.</b>)</li> <li>Synchronous operation (Refer to <b>16.4.3 Synchronous Operation.</b>)</li> <li>Digital filter The TRDIOji input is sampled, and when the sampled input level match as 3 times, the level is determined.</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

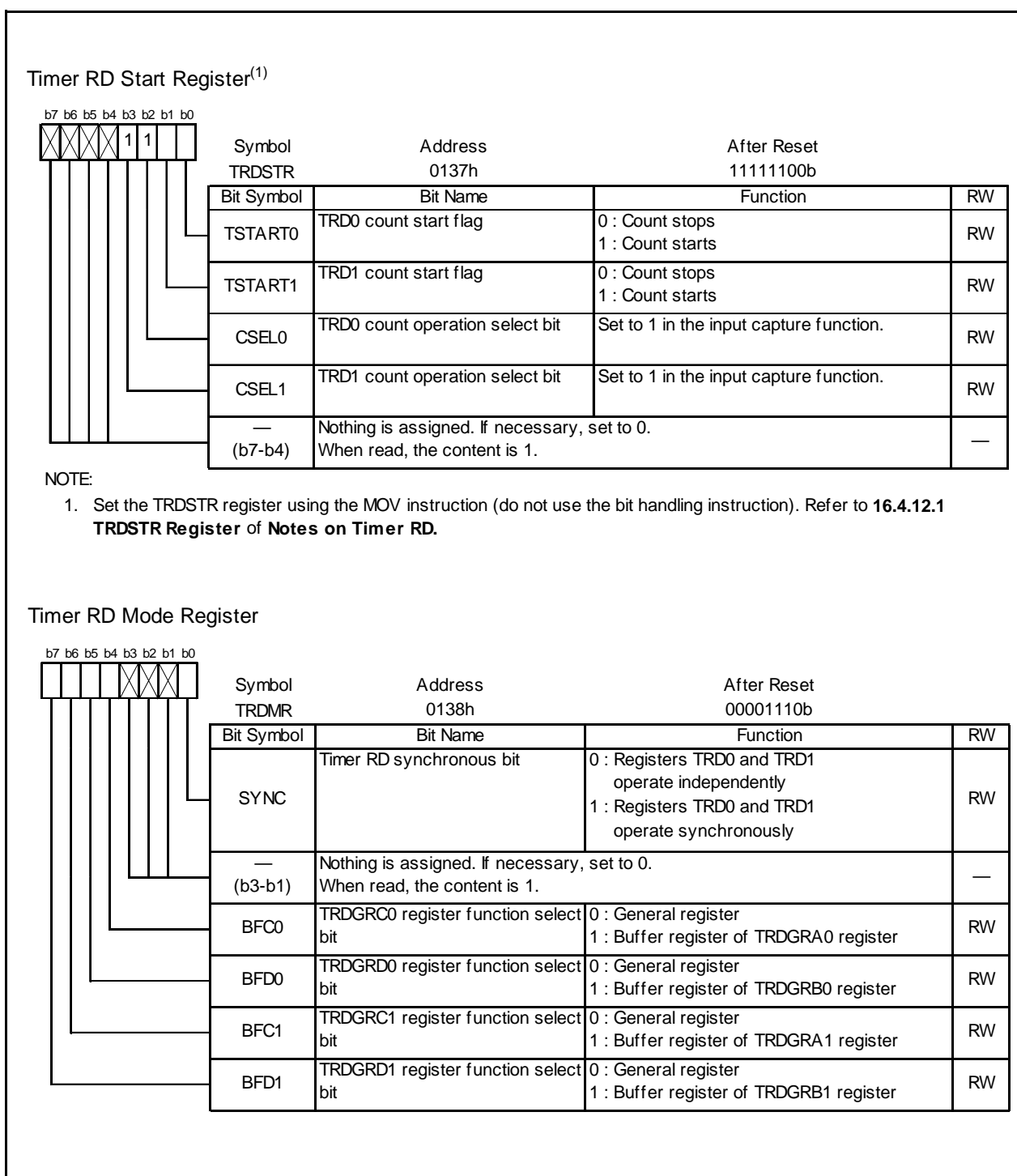


Figure 16.70 Registers TRDSTR and TRDMR in Input Capture Function



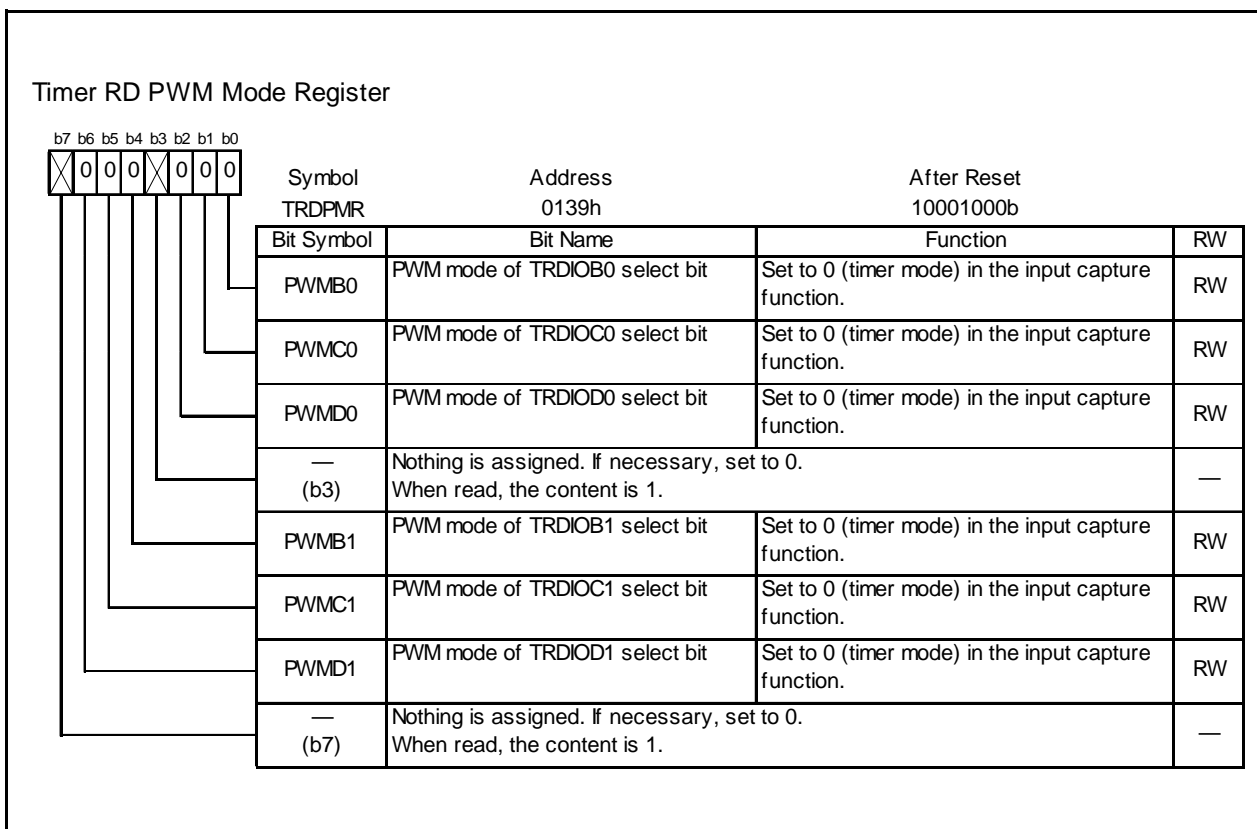


Figure 16.71 TRDPMR Register in Input Capture Function

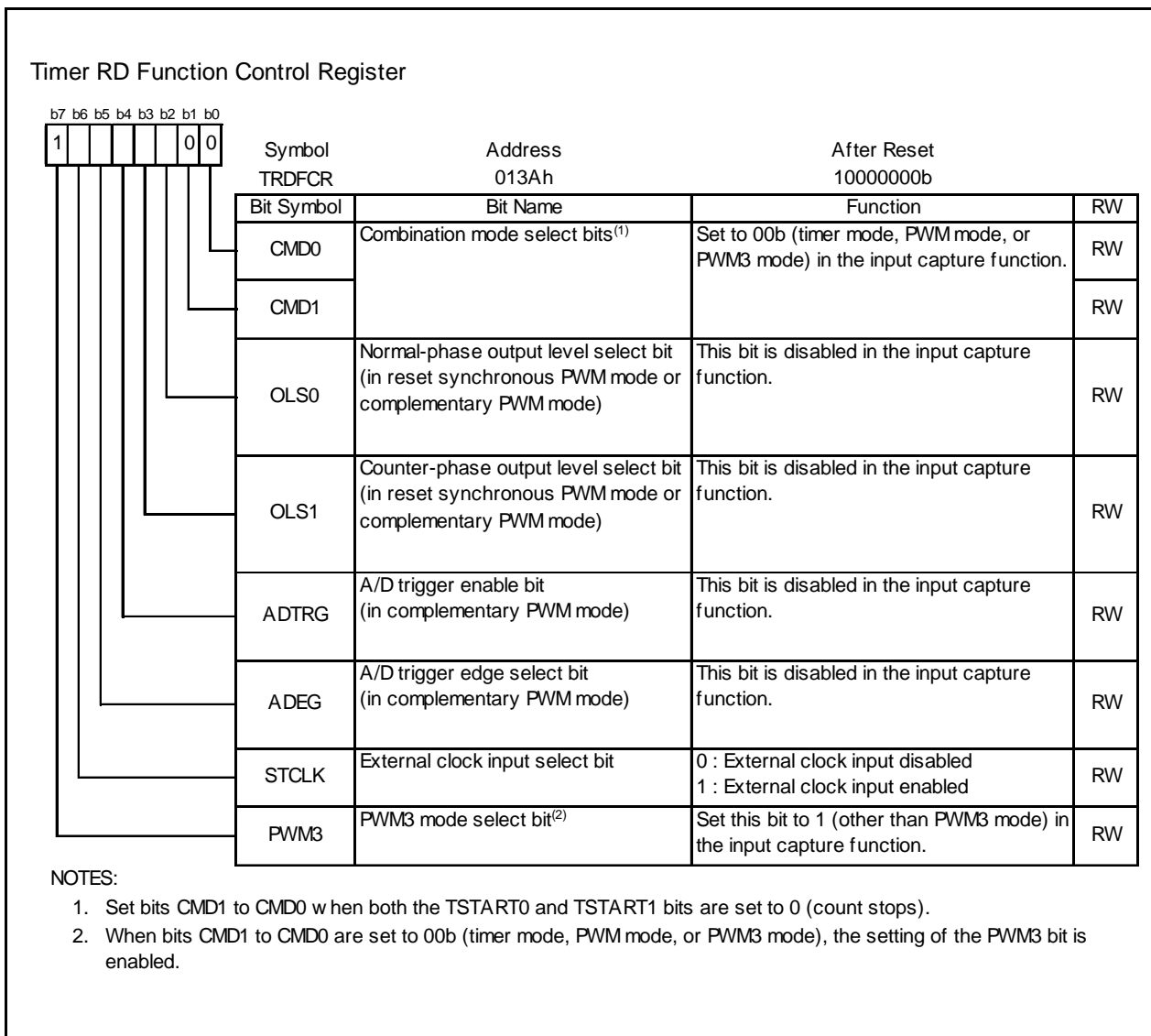


Figure 16.72 TRDFCR Register in Input Capture Function

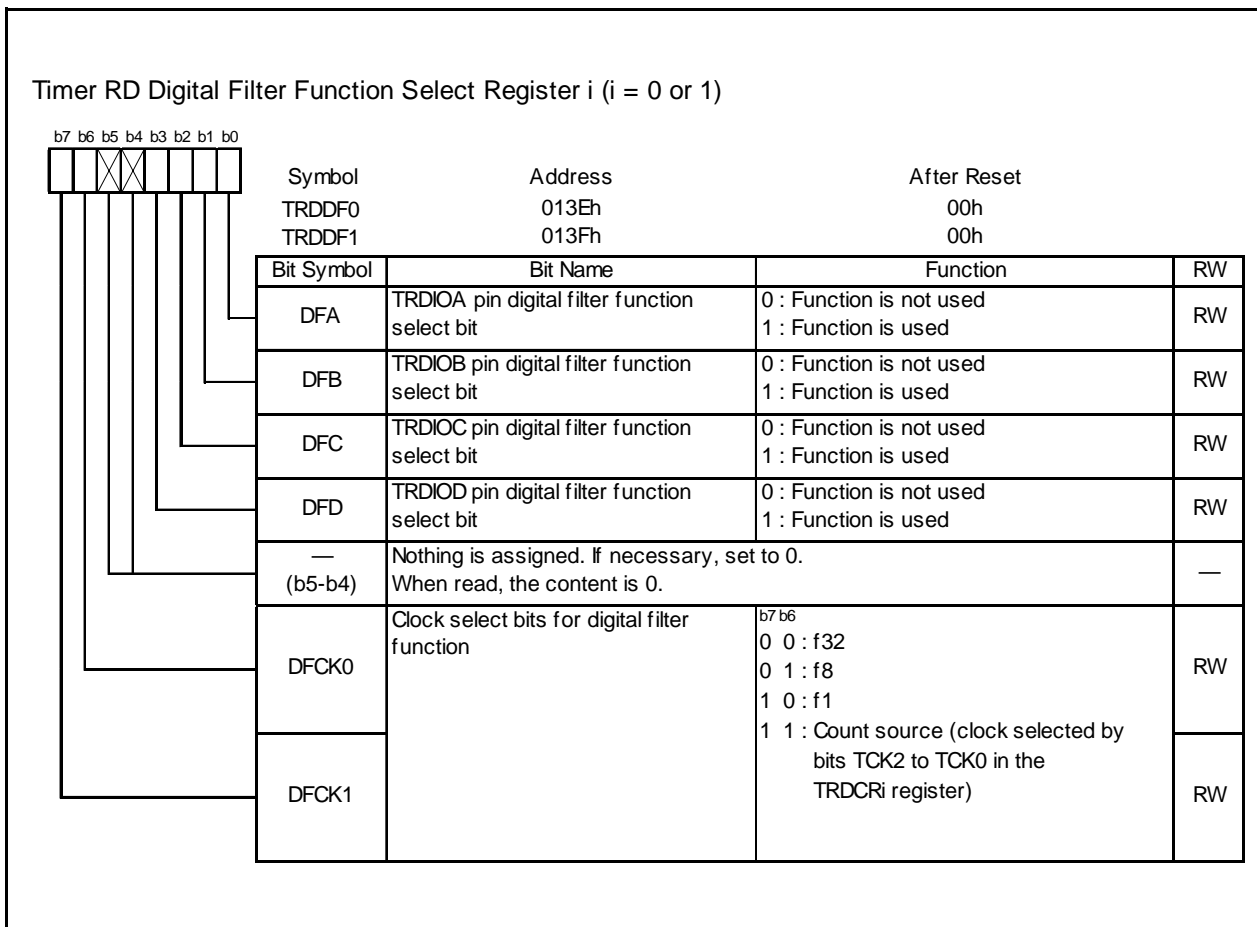


Figure 16.73 Registers TRDDF0 to TRDDF1 in Input Capture Function

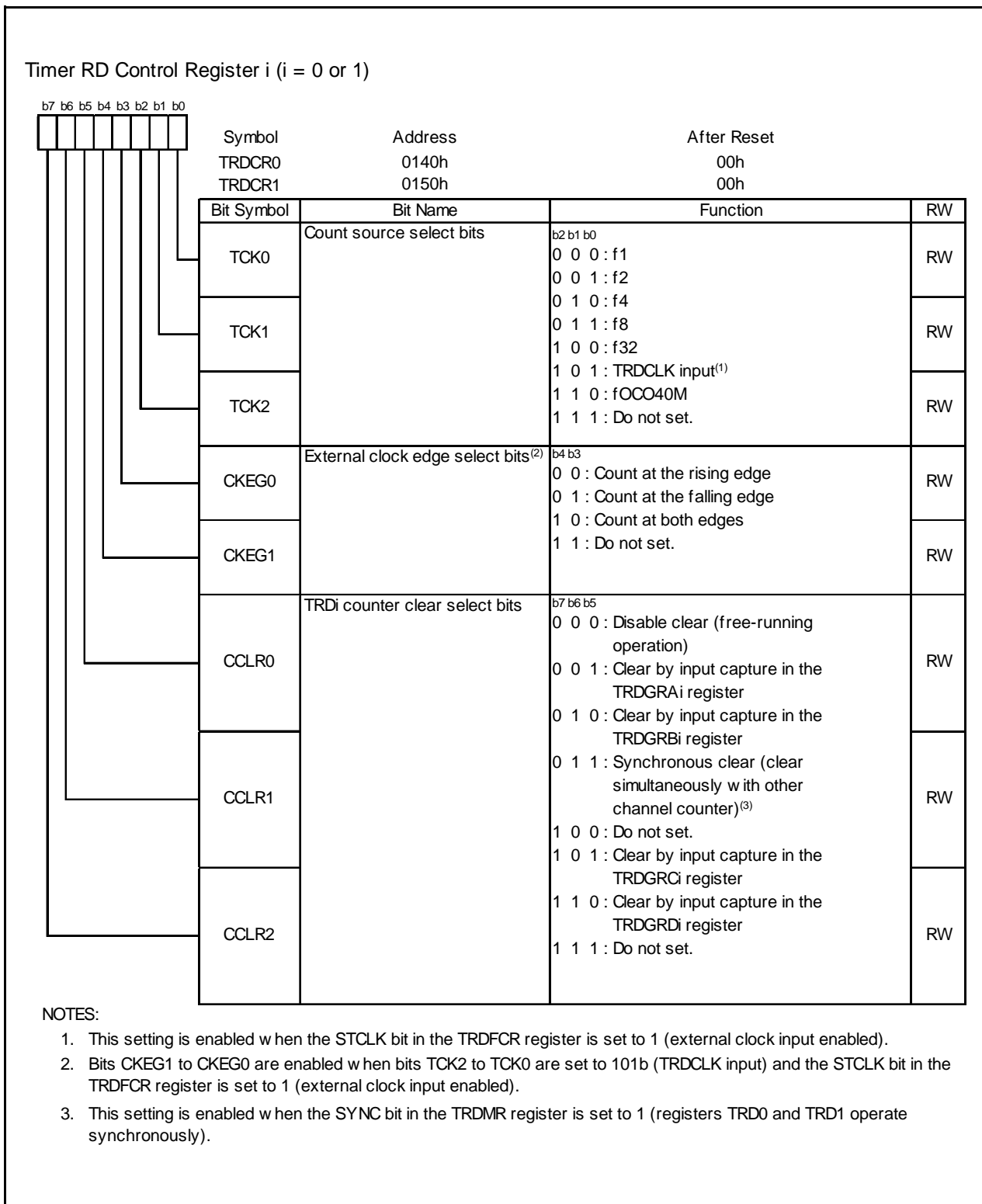


Figure 16.74 Registers TRDCR0 to TRDCR1 in Input Capture Function

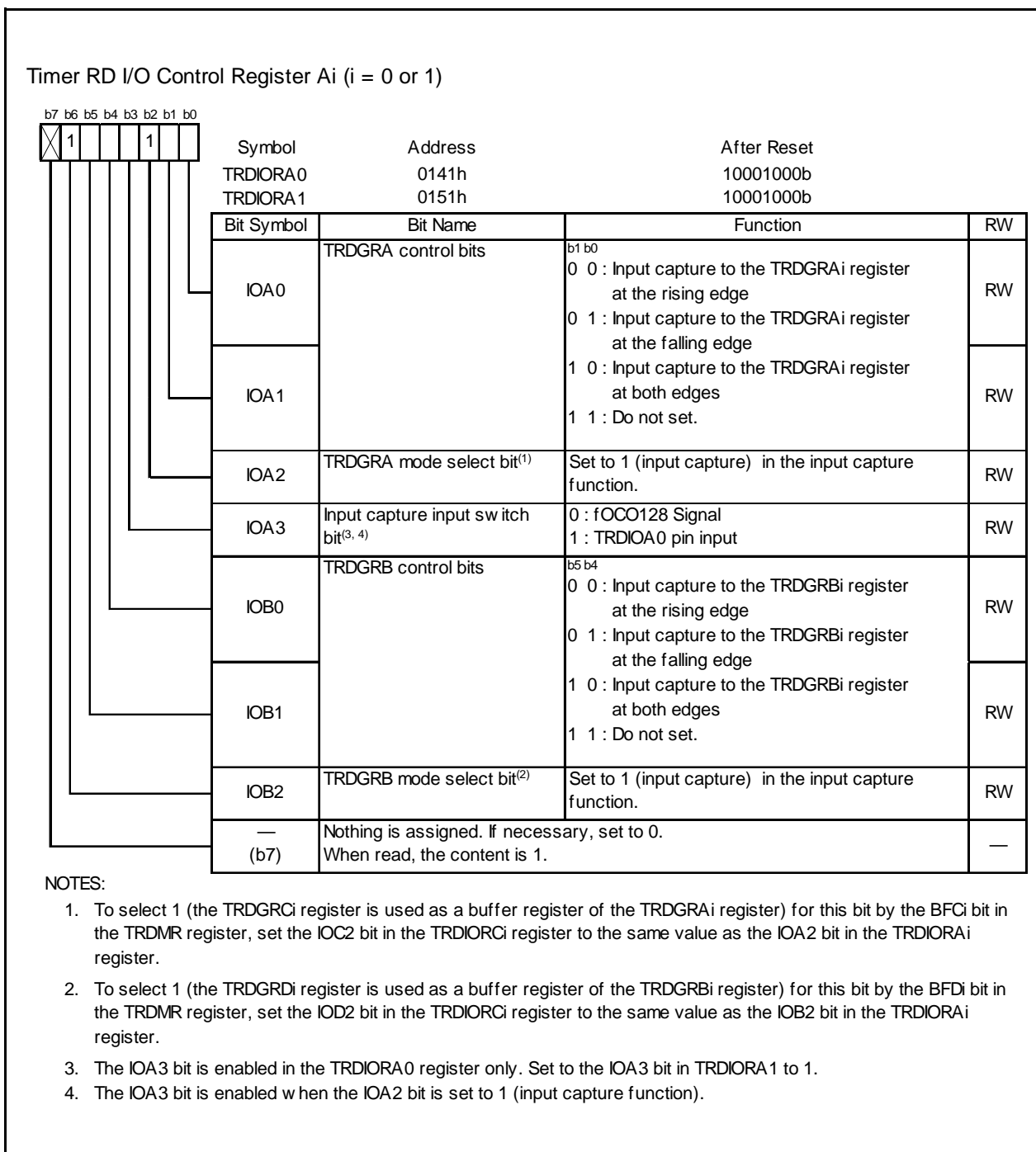


Figure 16.75 Registers TRDIORA0 to TRDIORA1 in Input Capture Function

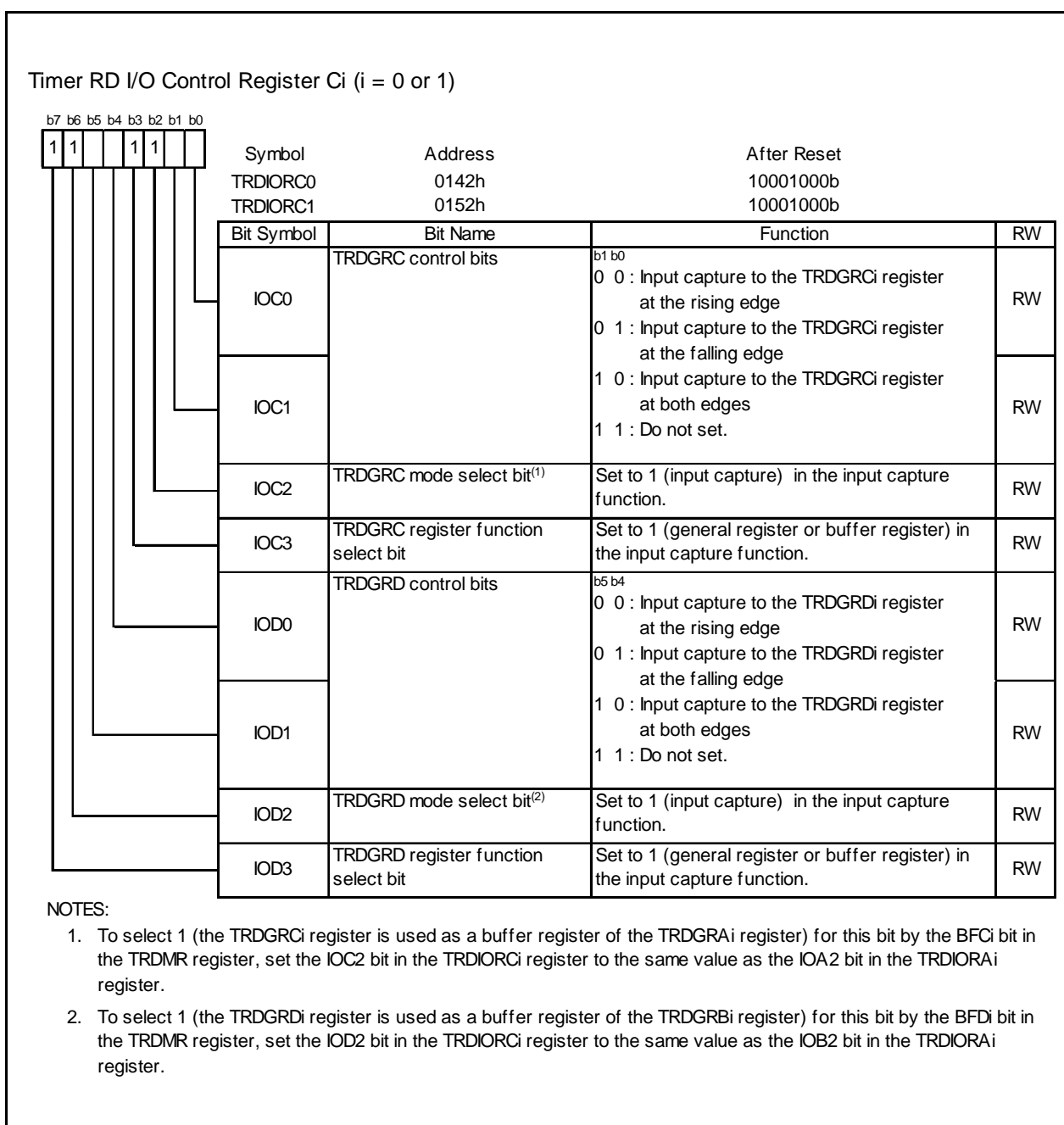


Figure 16.76 Registers TRDIORC0 to TRDIORC1 in Input Capture Function

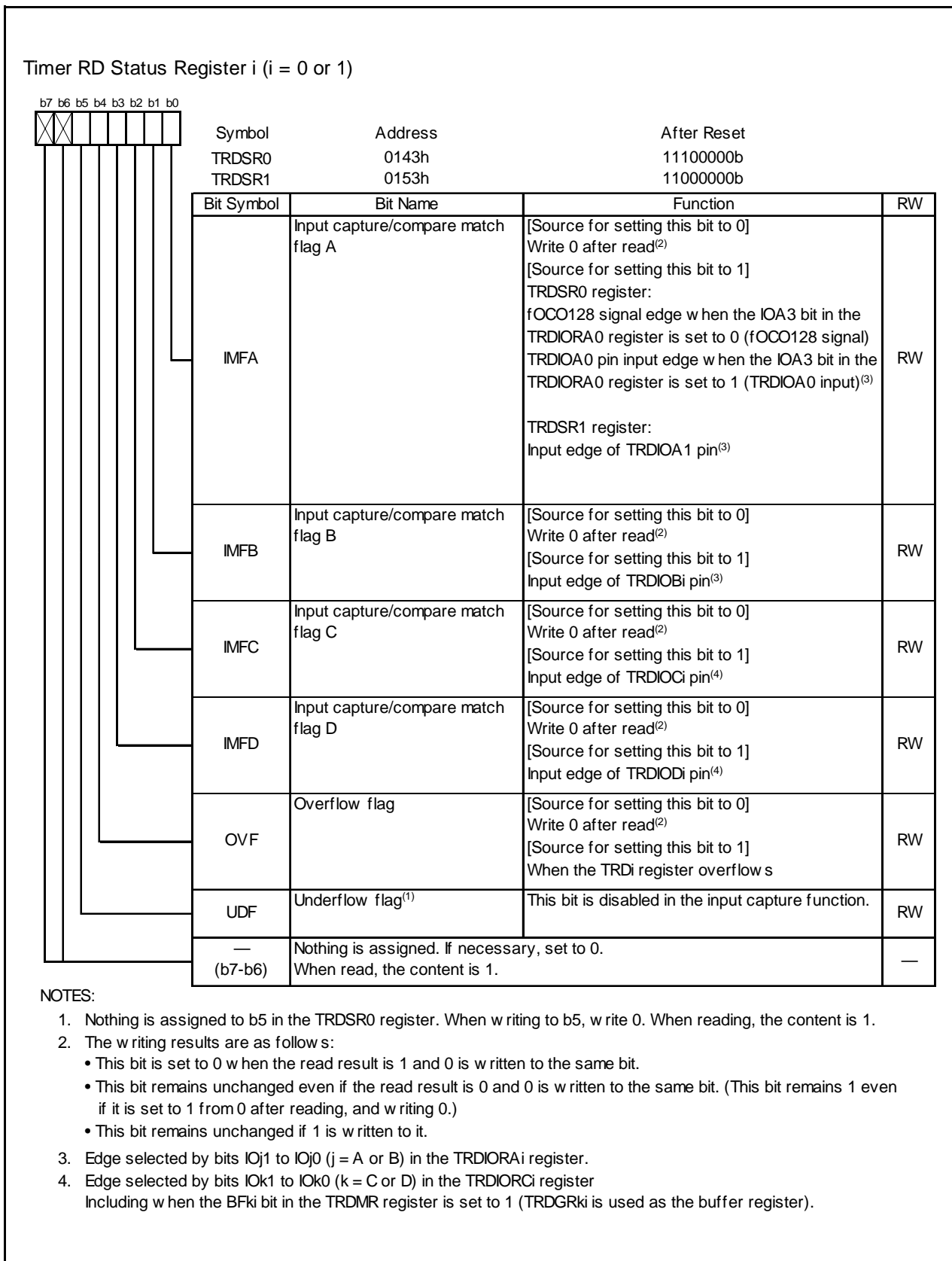


Figure 16.77 Registers TRDSR0 to TRDSR1 in Input Capture Function

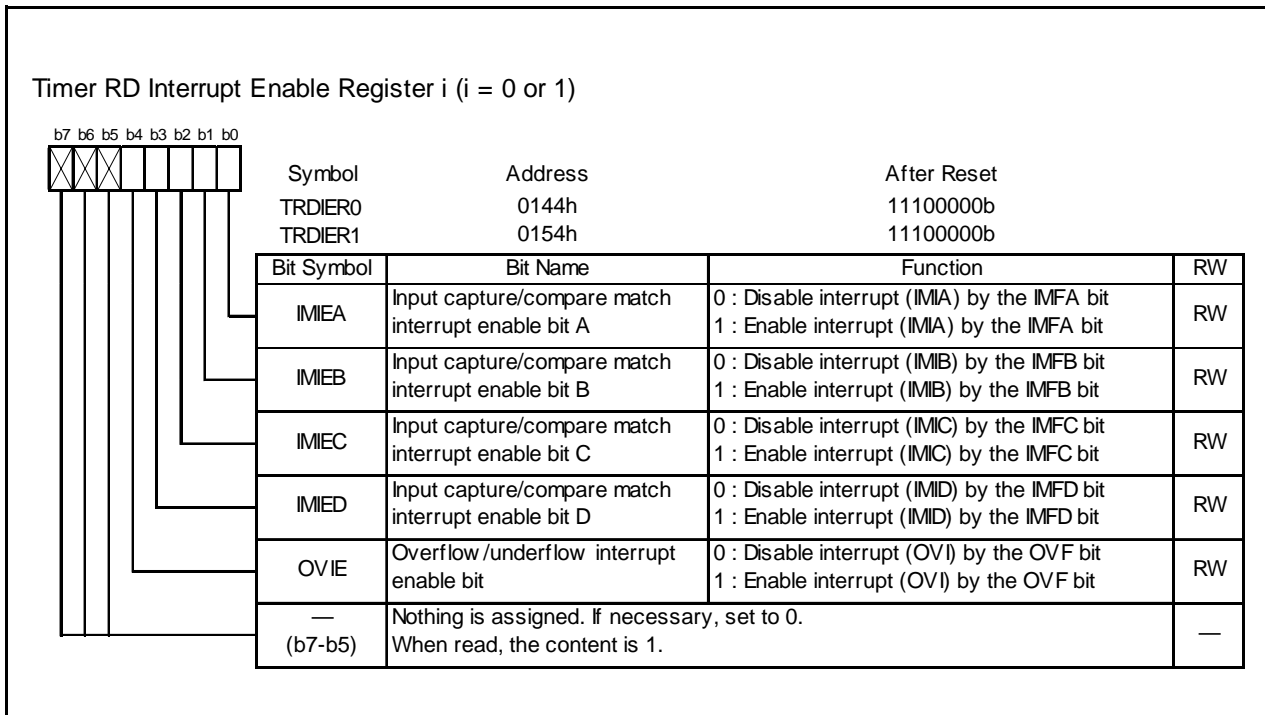


Figure 16.78 Registers TRDIER0 to TRDIER1 in Input Capture Function

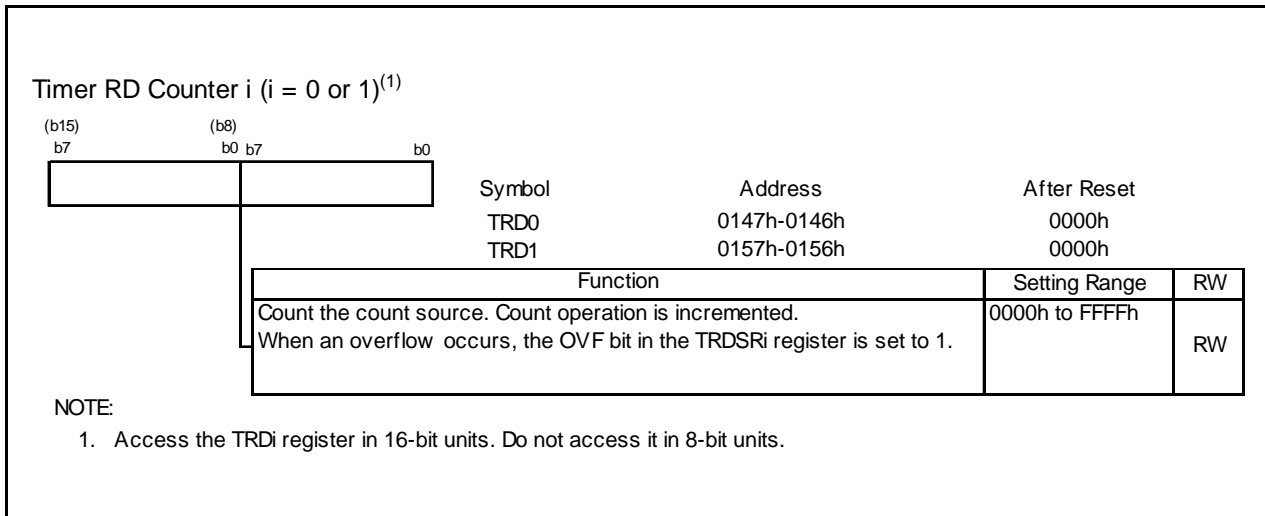


Figure 16.79 Registers TRD0 to TRD1 in Input Capture Function



Timer RD General Registers 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 16.38 TRDGRji Register Functions in Input Capture Function</b>				RW

NOTE:

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

**Figure 16.80 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.

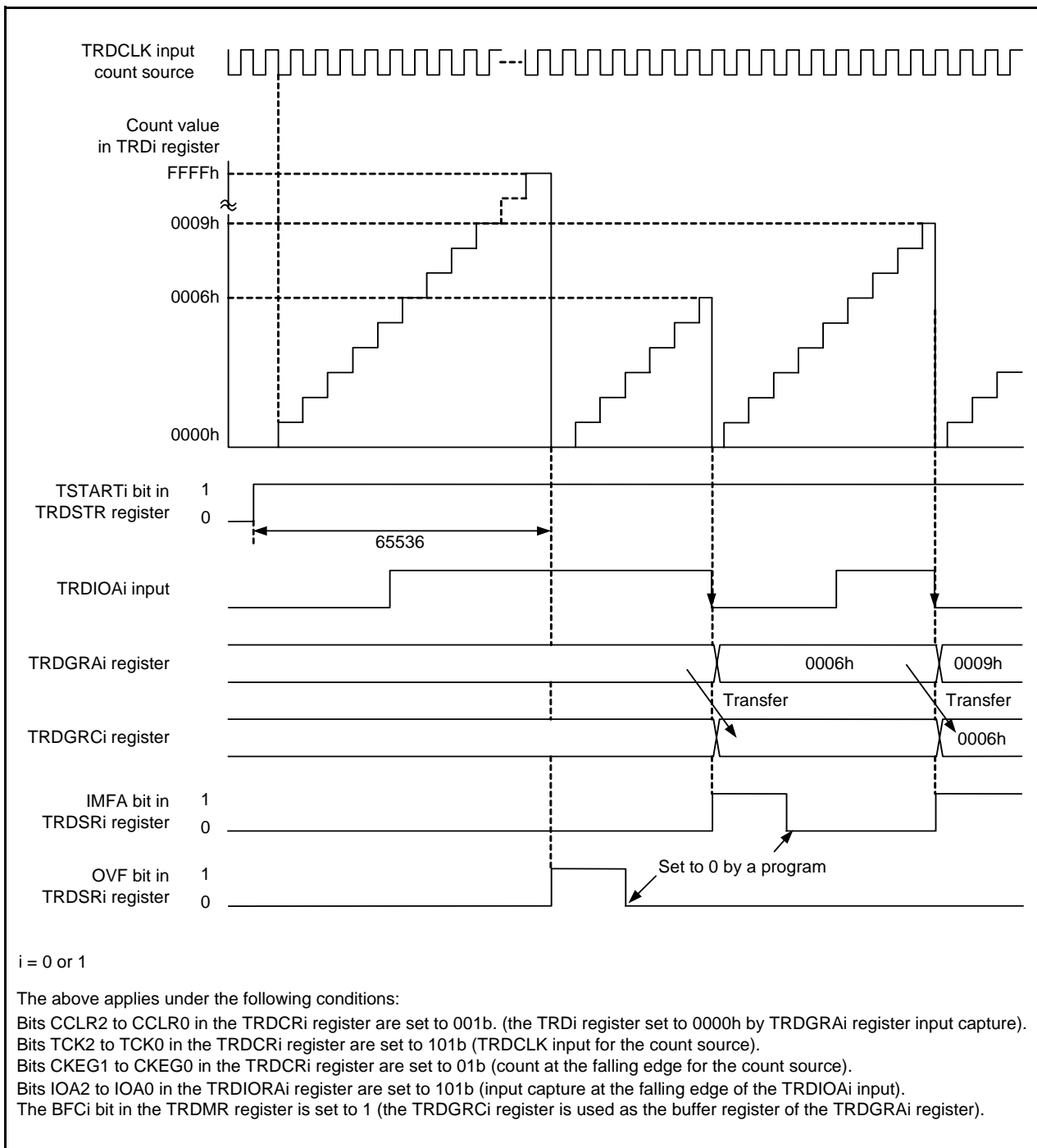
**Table 16.38 TRDGRji Register Functions in Input Capture Function**

Register	Setting	Register Function	Input-Capture Input Pin
TRDGRAi	–	General register The value in the TRDi register can be read at input capture.	TRDIOAi
TRDGRBi			TRDIOBi
TRDGRCi	BFCi = 0	General register The value in the TRDi register can be read at input capture.	TRDIOCi
TRDGRDi	BFDi = 0		TRDIODi
TRDGRCi	BFCi = 1	Buffer register The value in the TRDi register can be read at input capture. (Refer to <b>16.4.2 Buffer Operation</b> )	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 16.25 Timer RD Operation Clocks**) for no digital filter (the DFj bit in the TRDDFi register set to 0).



**Figure 16.81 Operating Example of Input Capture Function**

### 16.4.5.1 Digital Filter

The TRDIO<sub>j</sub> input is sampled, and when the sampled input level matches 3 times, its level is determined. Select the digital filter function and sampling clock by the TRDDFi register.

Figure 16.82 shows a Block Diagram of Digital Filter.

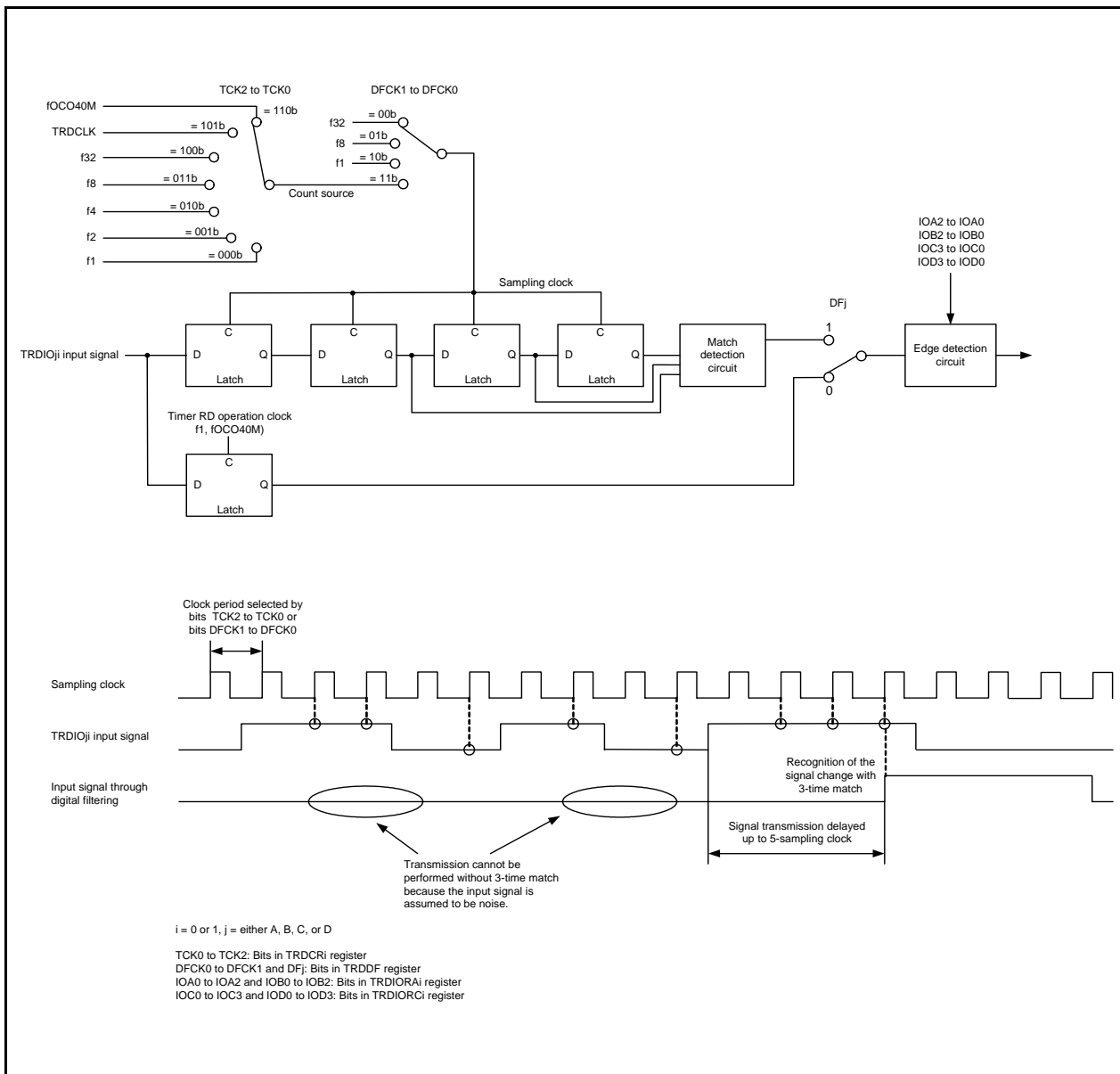


Figure 16.82 Block Diagram of Digital Filter

### 16.4.6 Output Compare Function

This function detects matches (compare match) between the content of the TRDGR<sub>ji</sub> (j = either A, B, C, or D) register and the content of the TRDi (i = 0 or 1) register. When the content matches, a user-set level is output from the TRDIO<sub>ji</sub> pin. Since this function is enabled with a combination of the TRDIO<sub>ji</sub> pin and TRDGR<sub>ji</sub> register, the output compare function, or any other mode or function, can be selected for each individual pin. Figure 16.83 shows a Block Diagram of Output Compare Function, Table 16.39 lists the Output Compare Function Specifications. Figures 16.84 to 16.95 list the Registers Associated with Output Compare Function, and Figure 16.96 shows an Operating Example of Output Compare Function.

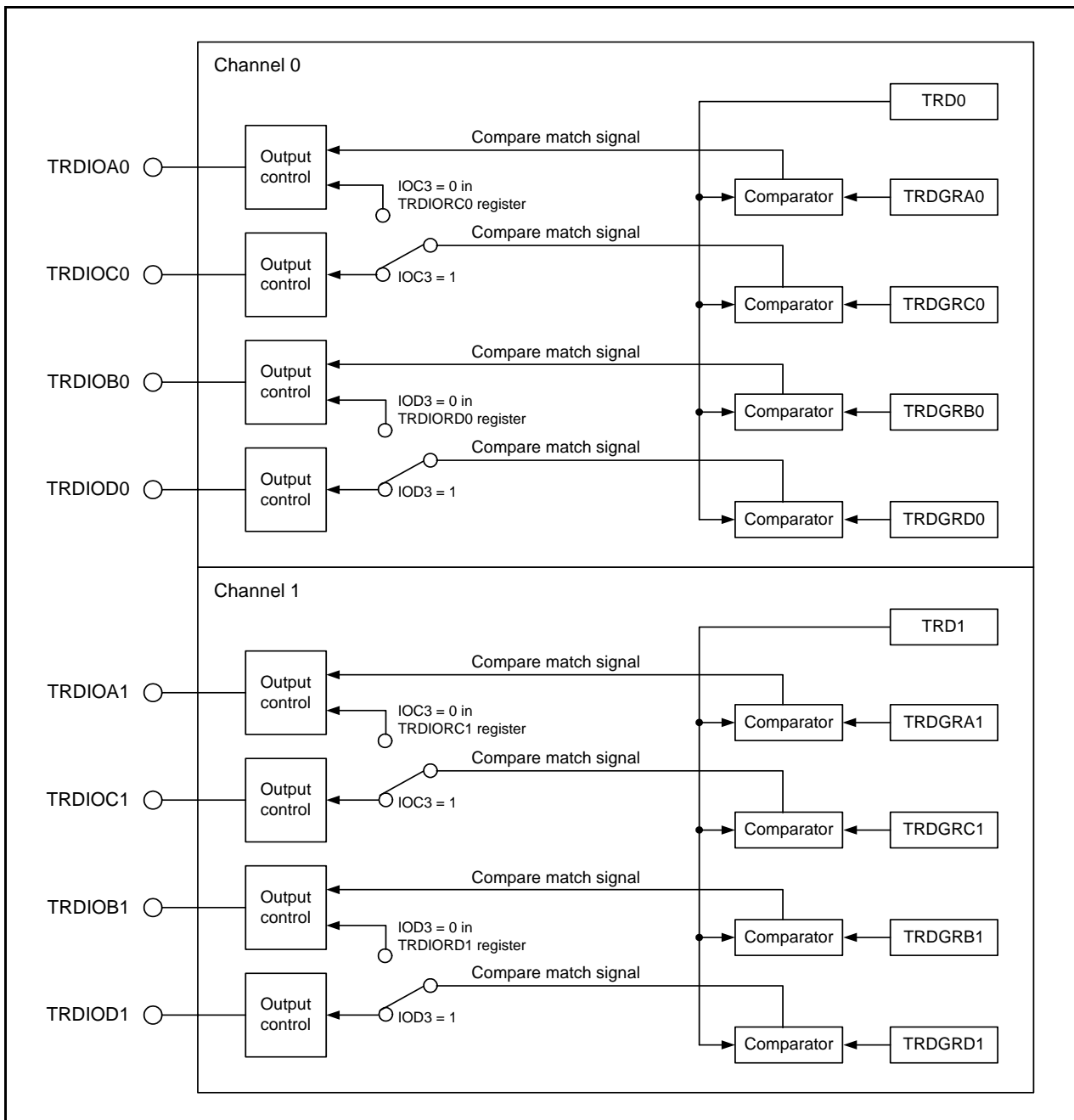


Figure 16.83 Block Diagram of Output Compare Function

**Table 16.39 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 bits CCLR2 to CCLR0 in the TRDCRi register are set to 000b (free-running operation) 1/fk × 65536 fk: Frequency of count source</li> <li>Bits CCLR1 to CCLR0 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	1 (count starts) is written to the TSTARTi bit in the TRDSTR register.
Count stop conditions	<ul style="list-style-type: none"> <li>0 (count stops) is written 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 (content of the TRDi register matches content of 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 (Selectable by 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>
Select functions	<ul style="list-style-type: none"> <li>Output-compare output pin selected Either 1 pin or multiple pins among TRDIOAi, TRDIOBi, TRDIOCi, or TRDIODi.</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>16.4.2 Buffer Operation.</b>)</li> <li>Synchronous operation (Refer to <b>16.4.3 Synchronous Operation.</b>)</li> <li>Output pin in registers TRDGRCi and TRDGRDi 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>16.4.4 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

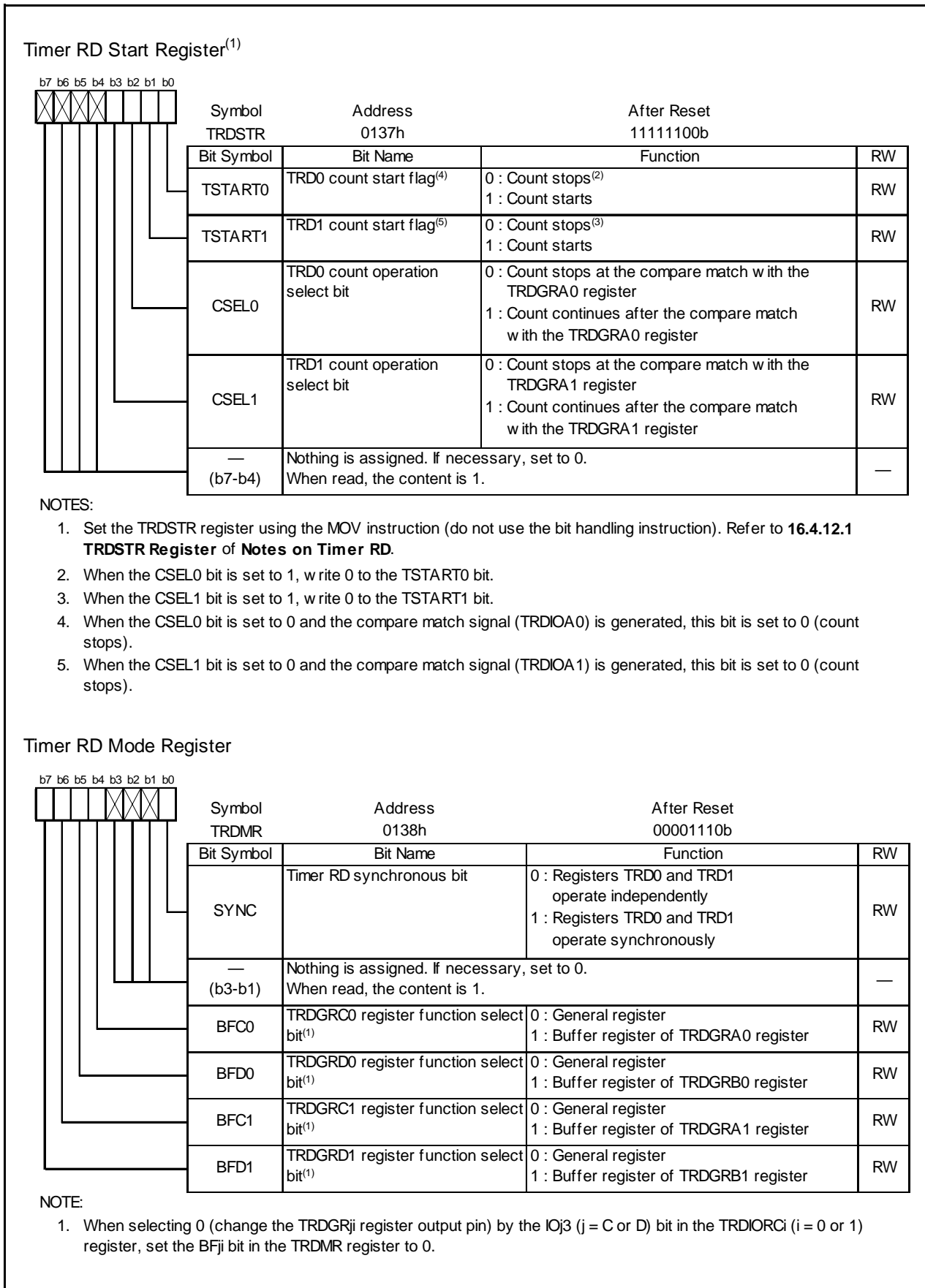


Figure 16.84 Registers TRDSTR and TRDMR in Output Compare Function

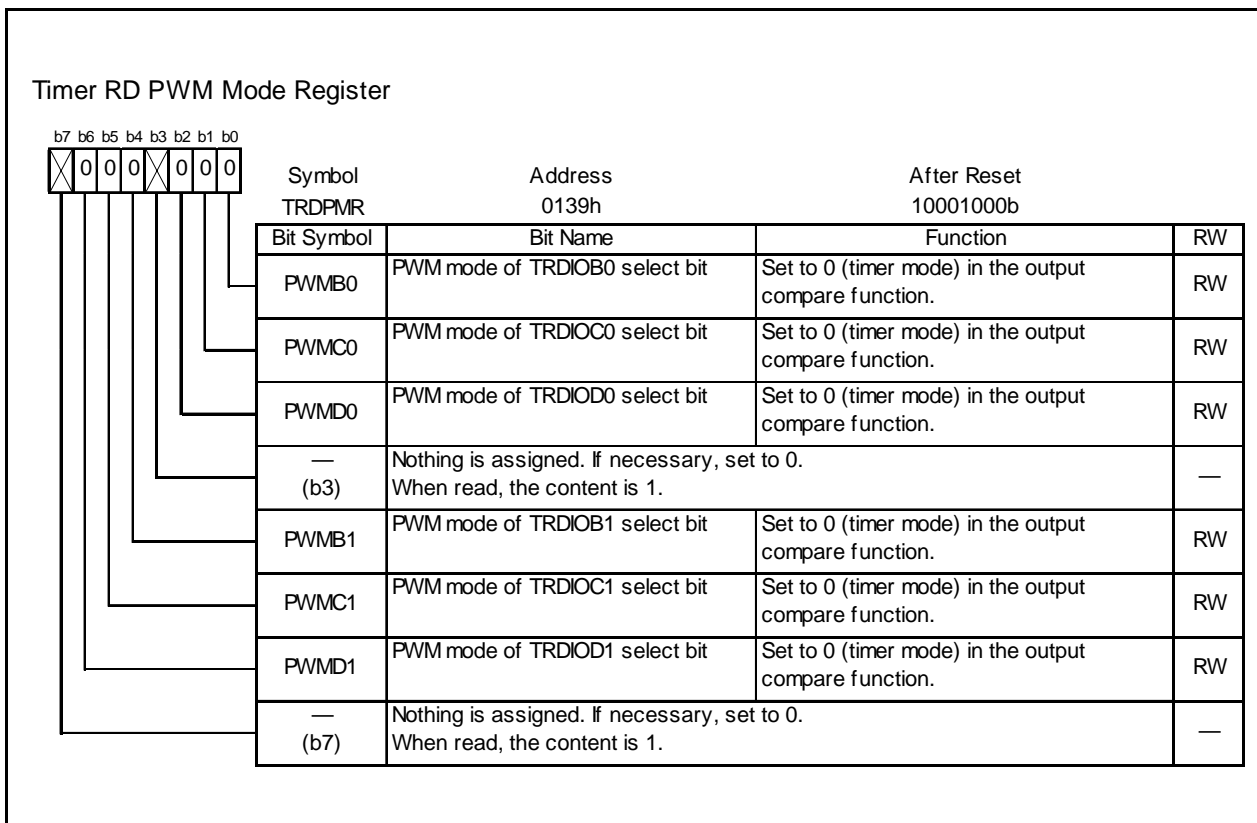


Figure 16.85 TRDPMR Register in Output Compare Function

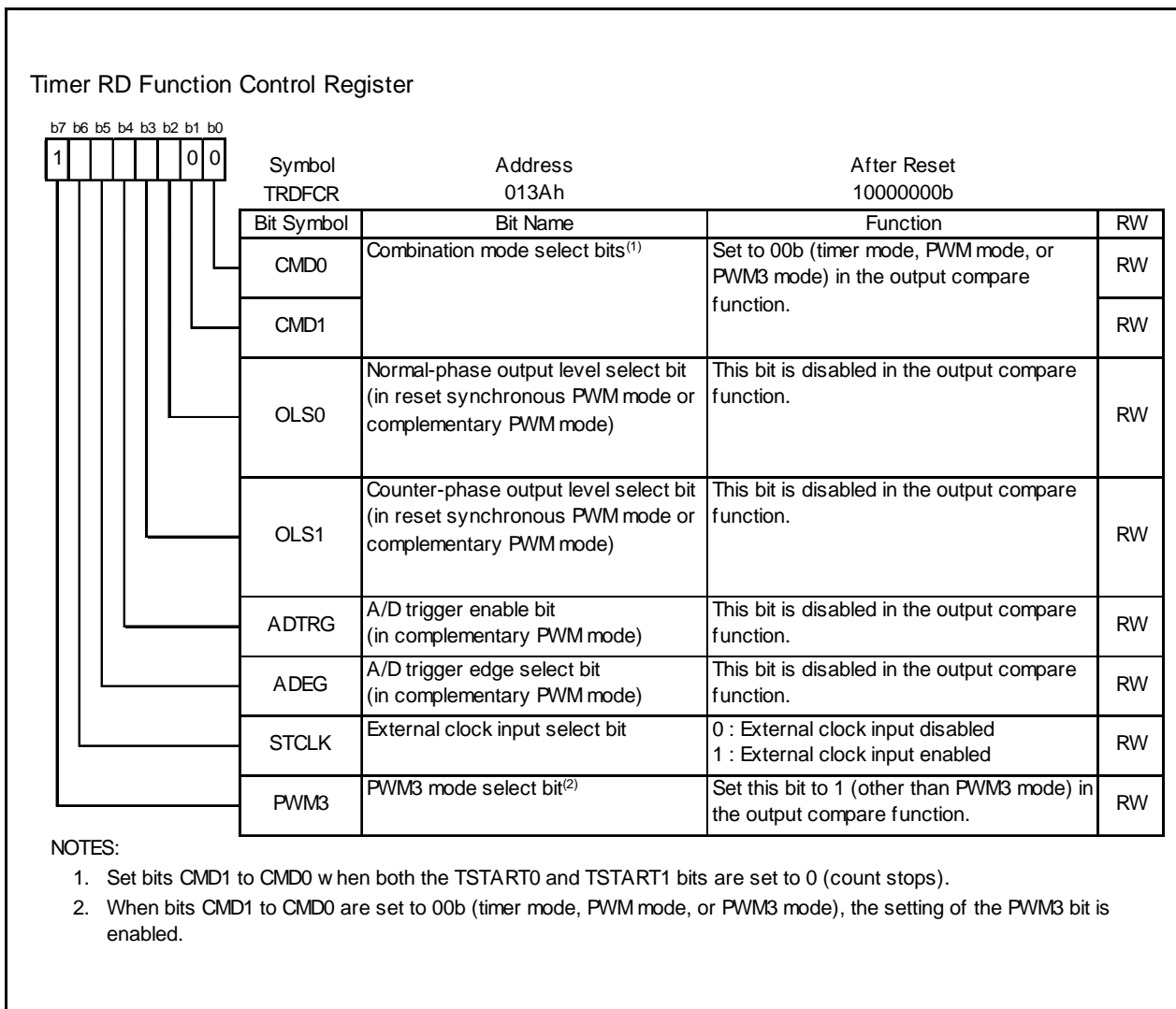


Figure 16.86 TRDFCR Register in Output Compare Function



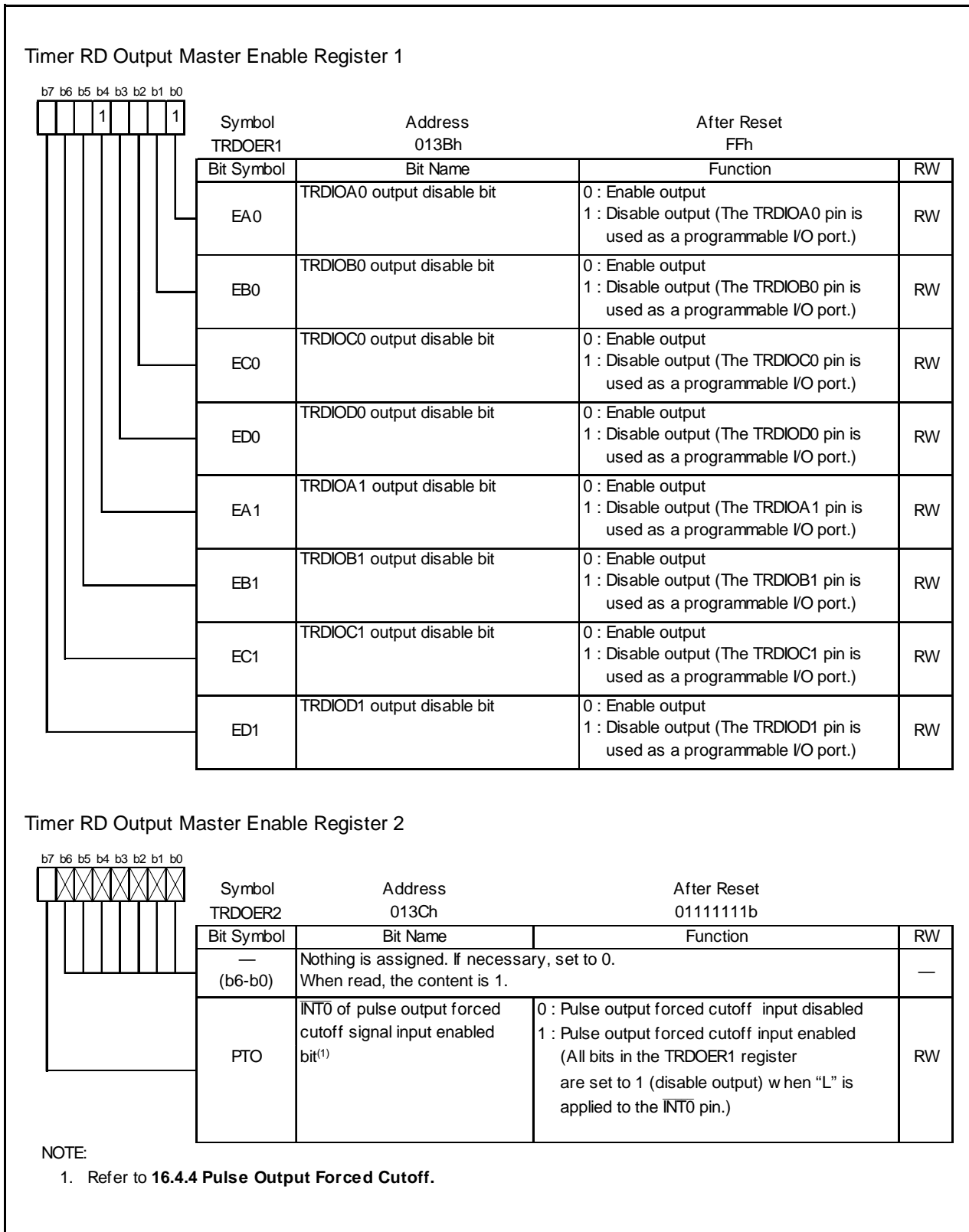


Figure 16.87 Registers TRDOER1 to TRDOER2 in Output Compare Function

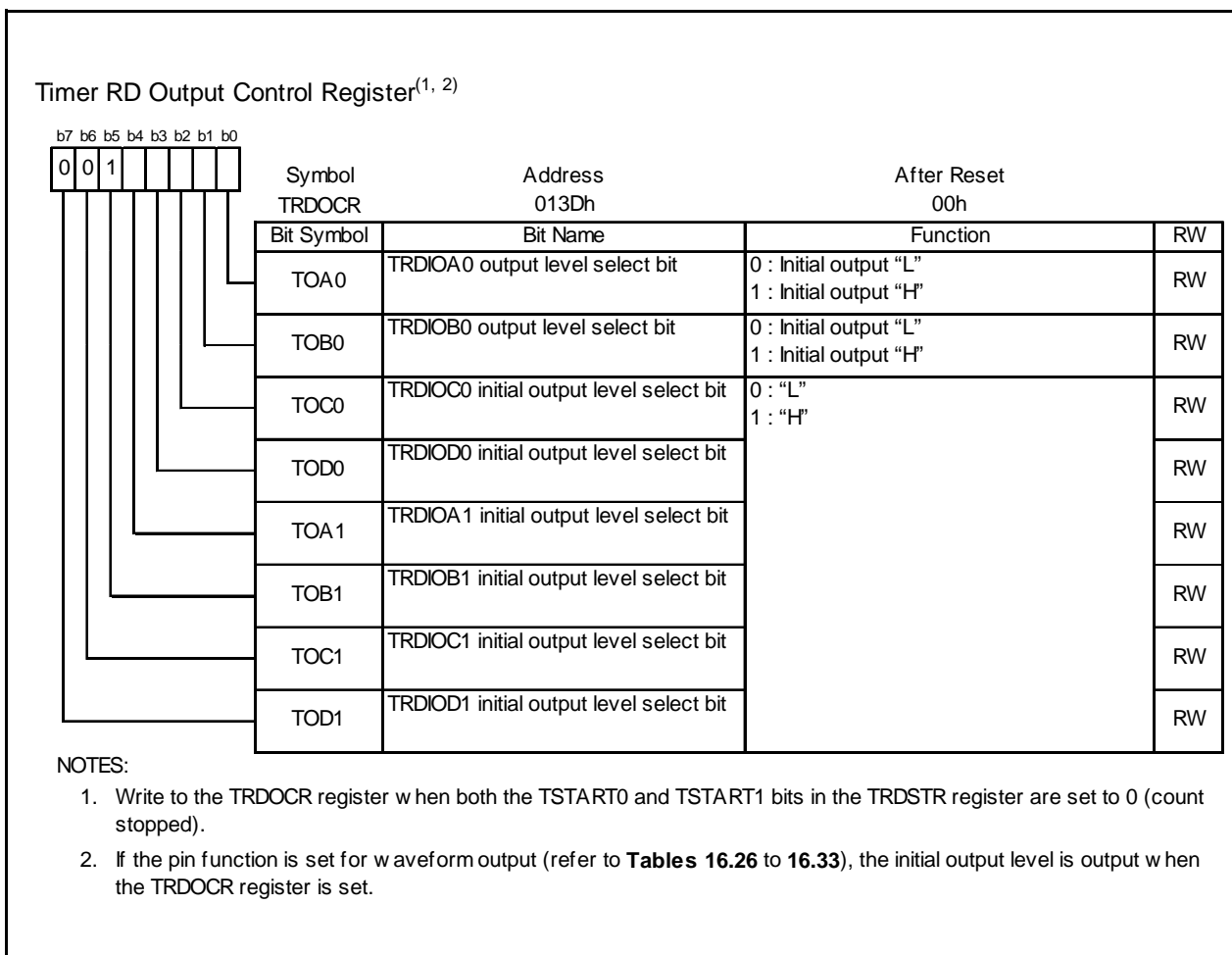


Figure 16.88 TRDOCR Register in Output Compare Function

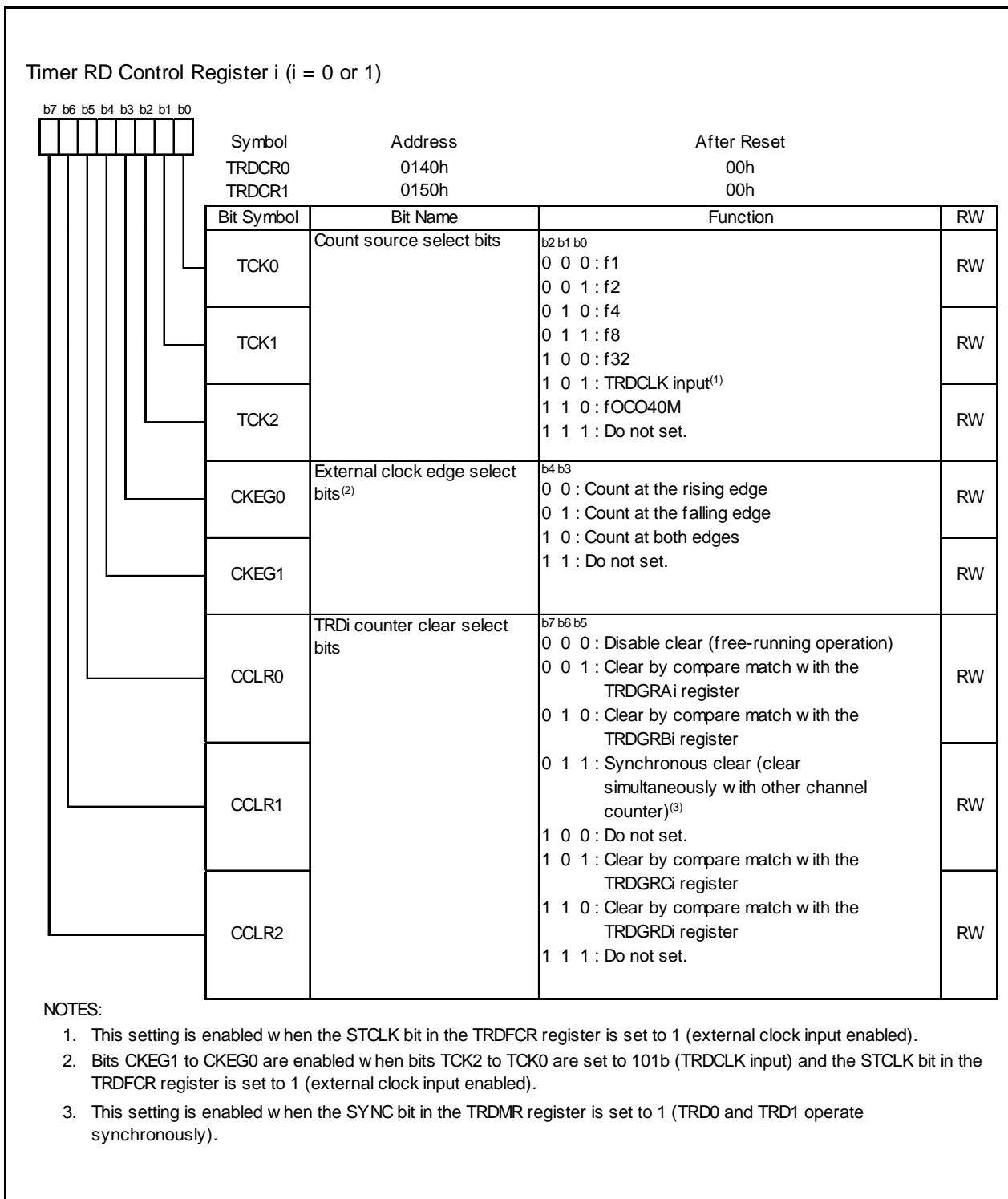


Figure 16.89 Registers TRDCR0 to TRDCR1 in Output Compare Function

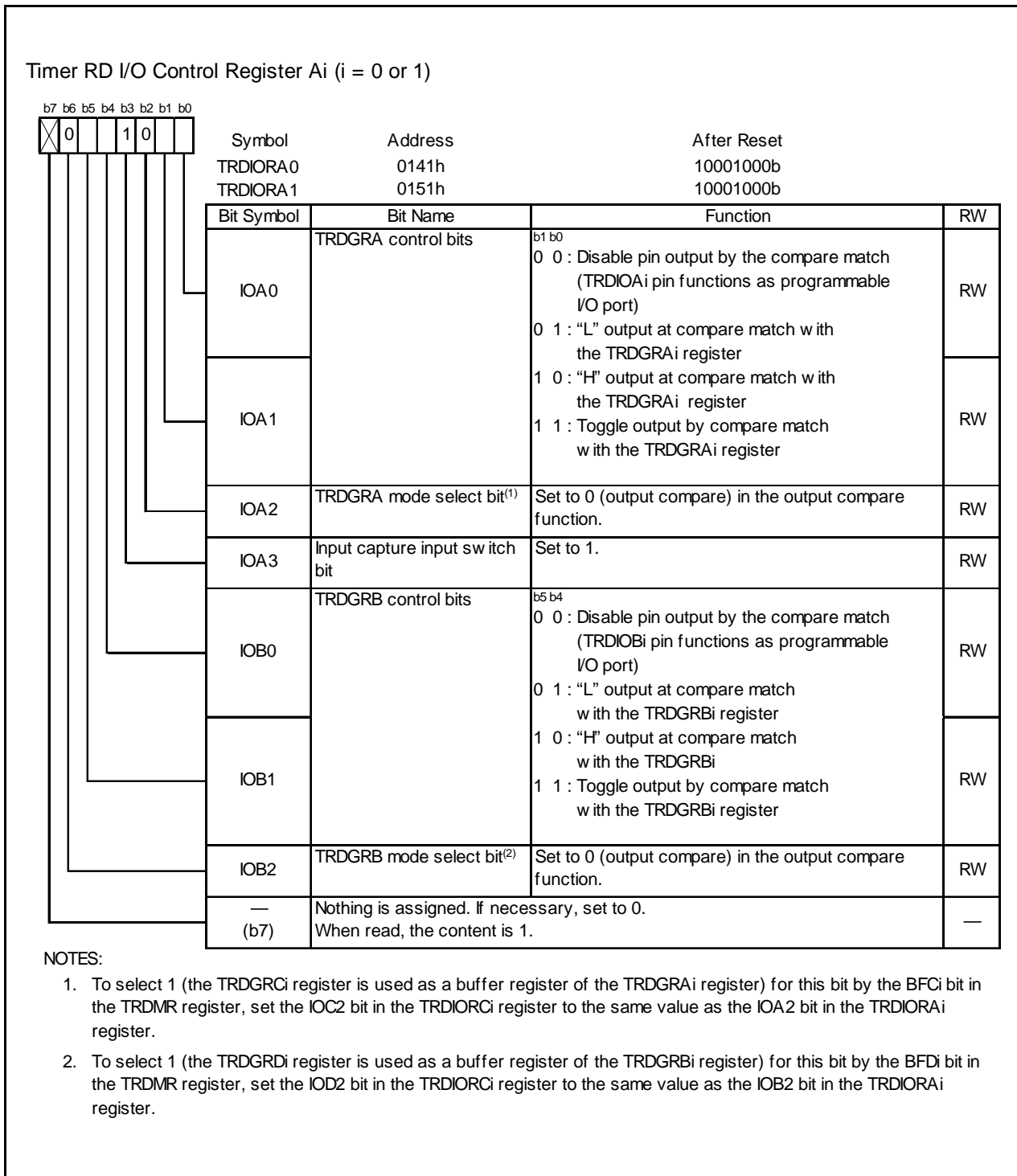


Figure 16.90 Registers TRDIORA0 to TRDIORA1 in Output Compare Function

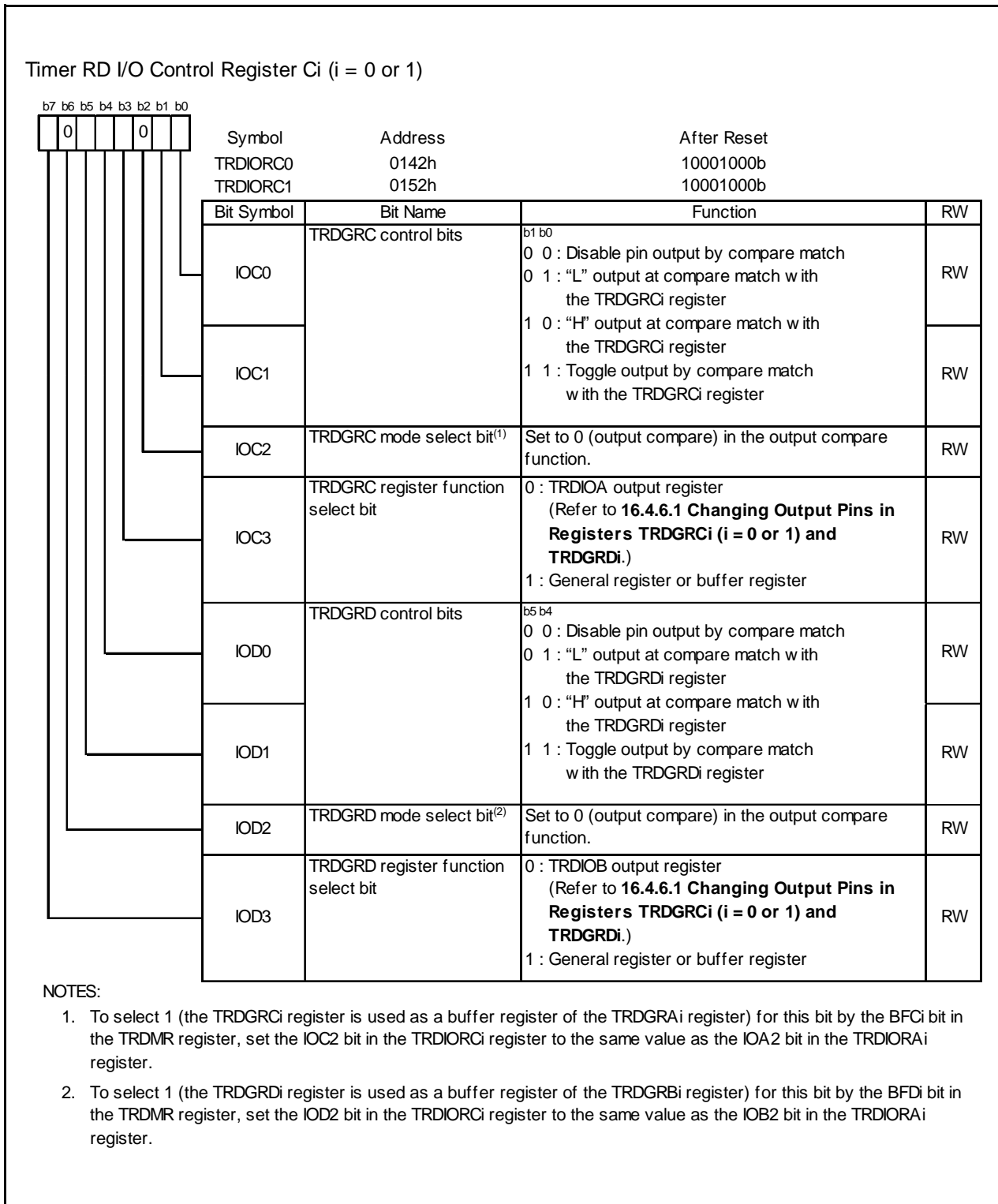


Figure 16.91 Registers TRDIORC0 to TRDIORC1 in Output Compare Function

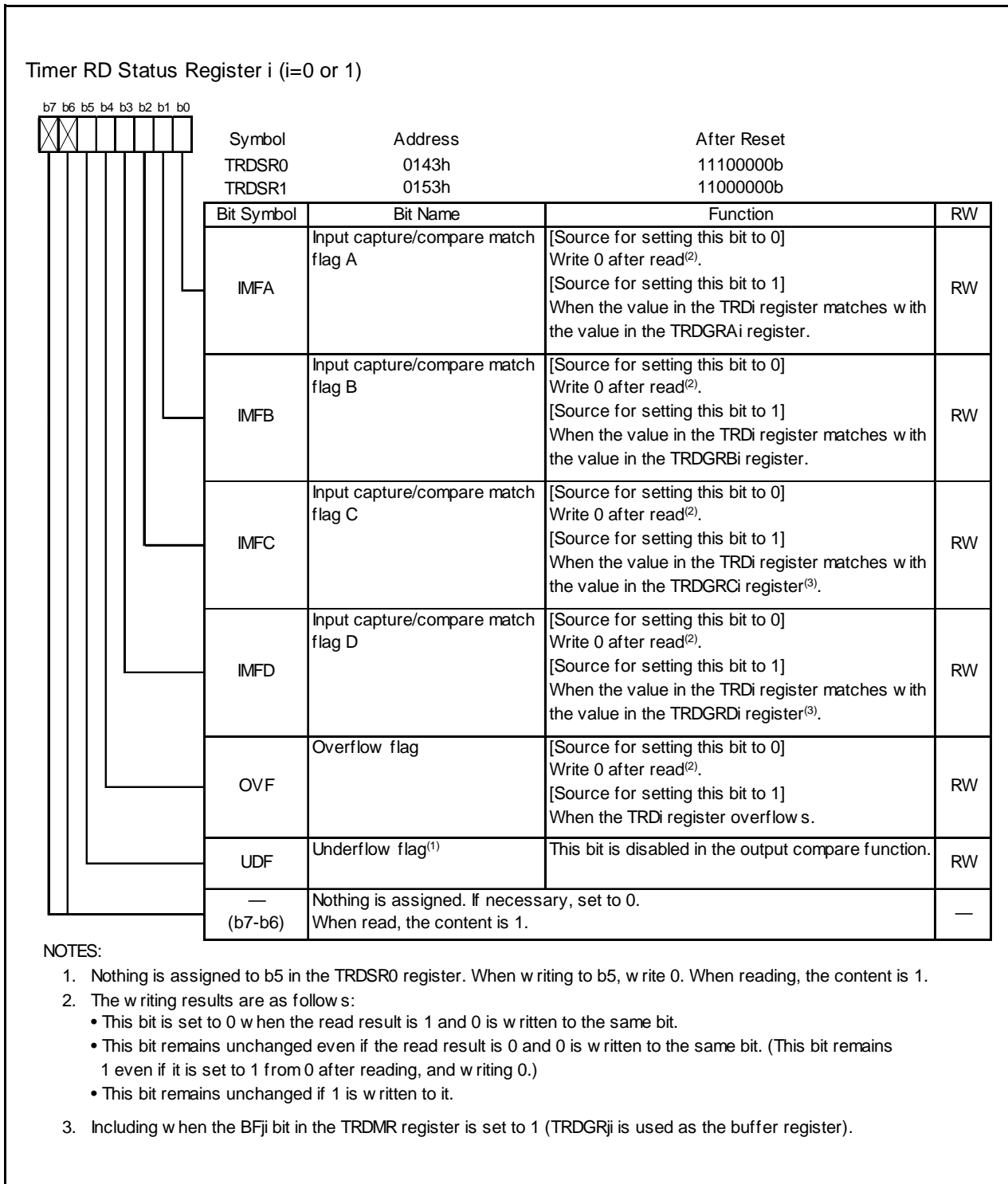


Figure 16.92 Registers TRDSR0 to TRDSR1 in Output Compare Function

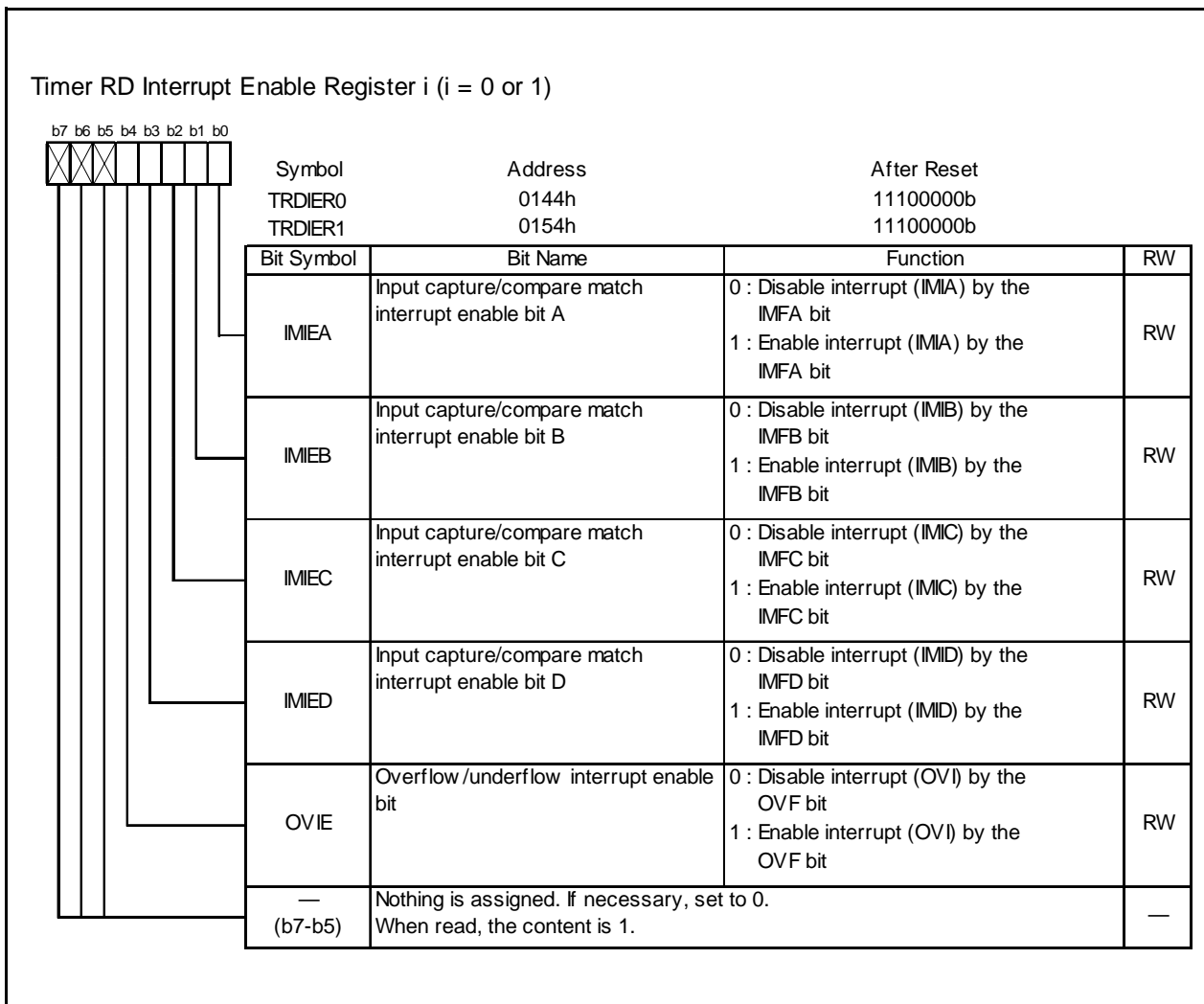


Figure 16.93 Registers TRDIER0 to TRDIER1 in Output Compare Function

Timer RD Counter $i$ ( $i = 0$ or $1$ ) <sup>(1)</sup>		Symbol	Address	After Reset	
(b15) b7	(b8) b0 b7				
		TRD0	0147h-0146h	0000h	
		TRD1	0157h-0156h	0000h	
Function				Setting Range	RW
Count a count source. Count operation is incremented.				0000h to FFFFh	RW
When an overflow occurs, the OVF bit in the TRDSR $i$ register is set to 1.					

NOTE:

- Access the TRD $i$  register in 16-bit units. Do not access it in 8-bit units.

Figure 16.94 Registers TRD0 to TRD1 in Output Compare Function

Timer RD General Register $A_i$ , $B_i$ , $C_i$ and $D_i$ ( $i = 0$ 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 Table 16.40 TRDGR $j_i$ Register Function in Output Compare Function				RW

NOTE:

- Access registers TRDGRA $i$  to TRDGRD $i$  in 16-bit units. Do not access them in 8-bit units.

Figure 16.95 Registers TRDGRA $i$ , TRDGRB $i$ , TRDGRC $i$ , and TRDGRD $i$  in Output Compare Function

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

Table 16.40 TRDGR $j_i$  Register Function in Output Compare Function

Register	Setting		Register Function	Output-Compare Output Pin
	BF $j_i$	IO $j_3$		
TRDGRA $i$	–	–	General register. Write the compare value.	TRDIOA $i$
TRDGRB $i$				TRDIOB $i$
TRDGRC $i$	0	1	General register. Write the compare value.	TRDIOC $i$
TRDGRD $i$				TRDIOD $i$
TRDGRC $i$	1	1	Buffer register. Write the next compare value (Refer to 16.4.2 Buffer Operation.)	TRDIOA $i$
TRDGRD $i$				TRDIOB $i$
TRDGRC $i$	0	0	TRDIOA $i$ output control (Refer to 16.4.6.1 Changing Output Pins in Registers TRDGRC $i$ ( $i = 0$ or $1$ ) and TRDGRD $i$ .)	TRDIOA $i$
TRDGRD $i$				TRDIOB $i$

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

BF $j_i$ : Bit in TRDMR register

IO $j_3$ : Bit in TRDIORC $i$  register



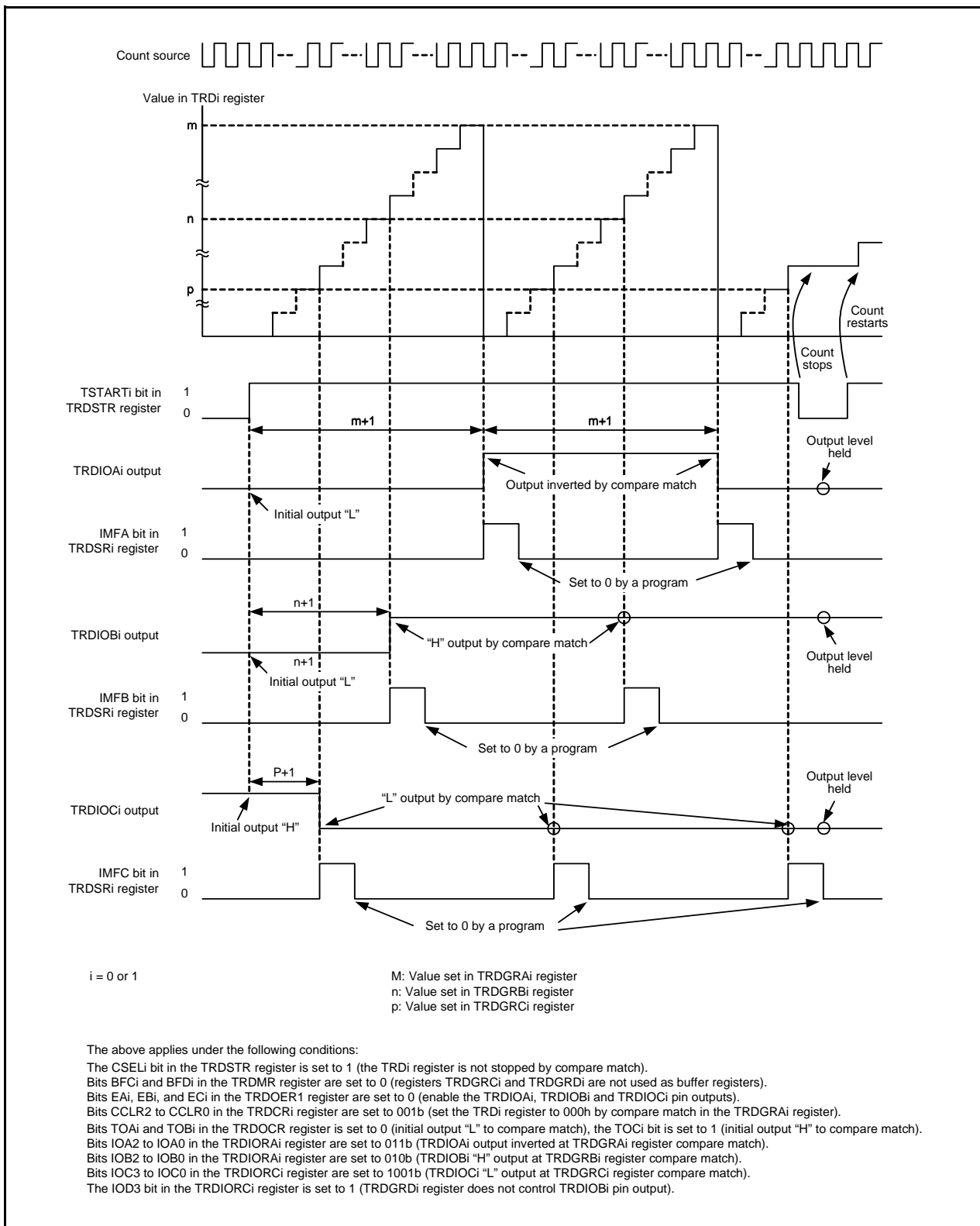


Figure 16.96 Operating Example of Output Compare Function

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

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

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

Change output pins in registers TRDGRCi and TRDGRDi as follows:

- 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 different values in registers TRDGRCi and TRDGRAi. Also, set different values in registers TRDGRDi and TRDGRBi.

Figure 16.98 shows an Operating Example When TRDGRCi Register is Used for Output Control of TRDIOAi Pin and TRDGRDi Register is Used for Output Control of TRDIOBi Pin.

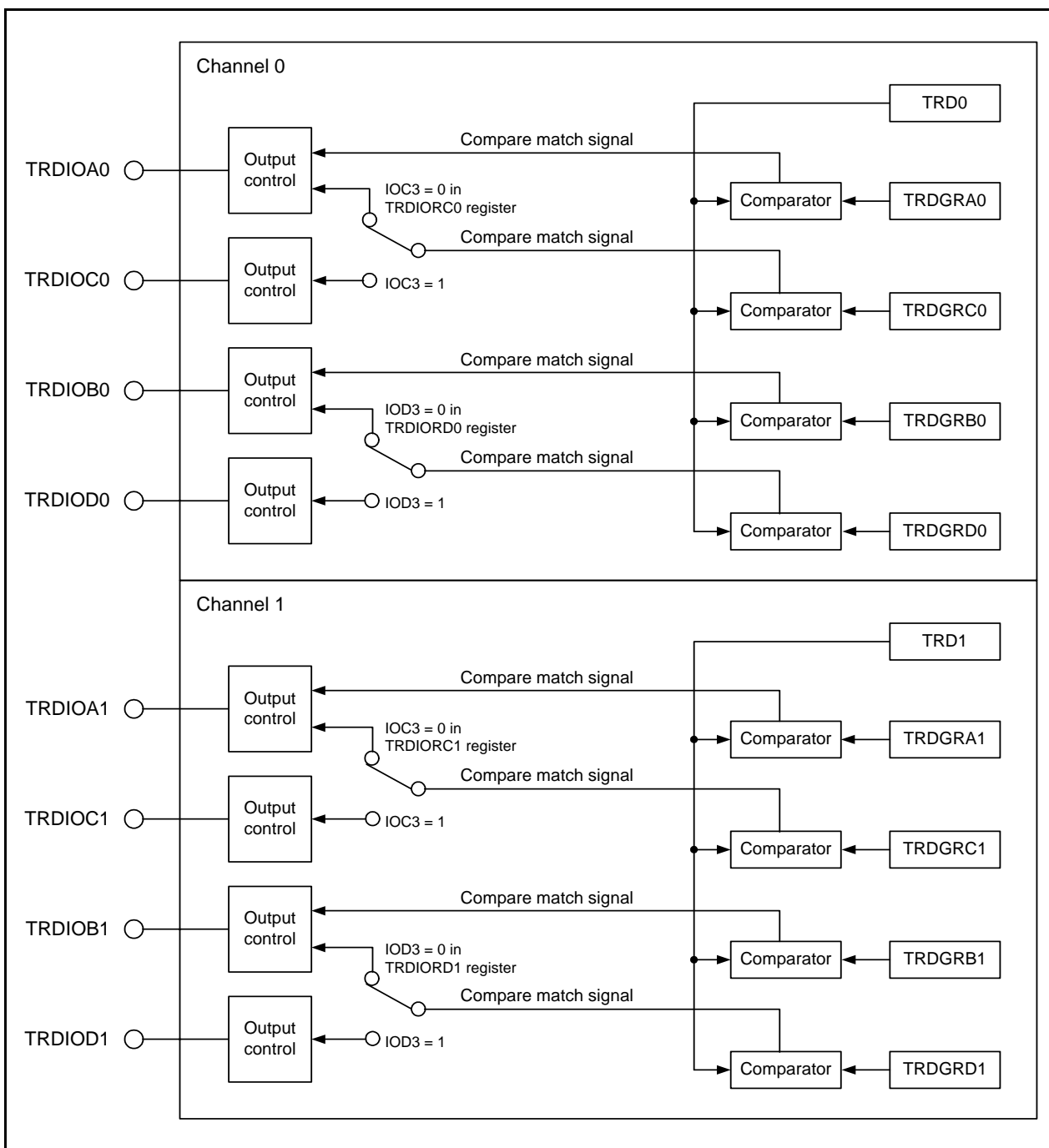
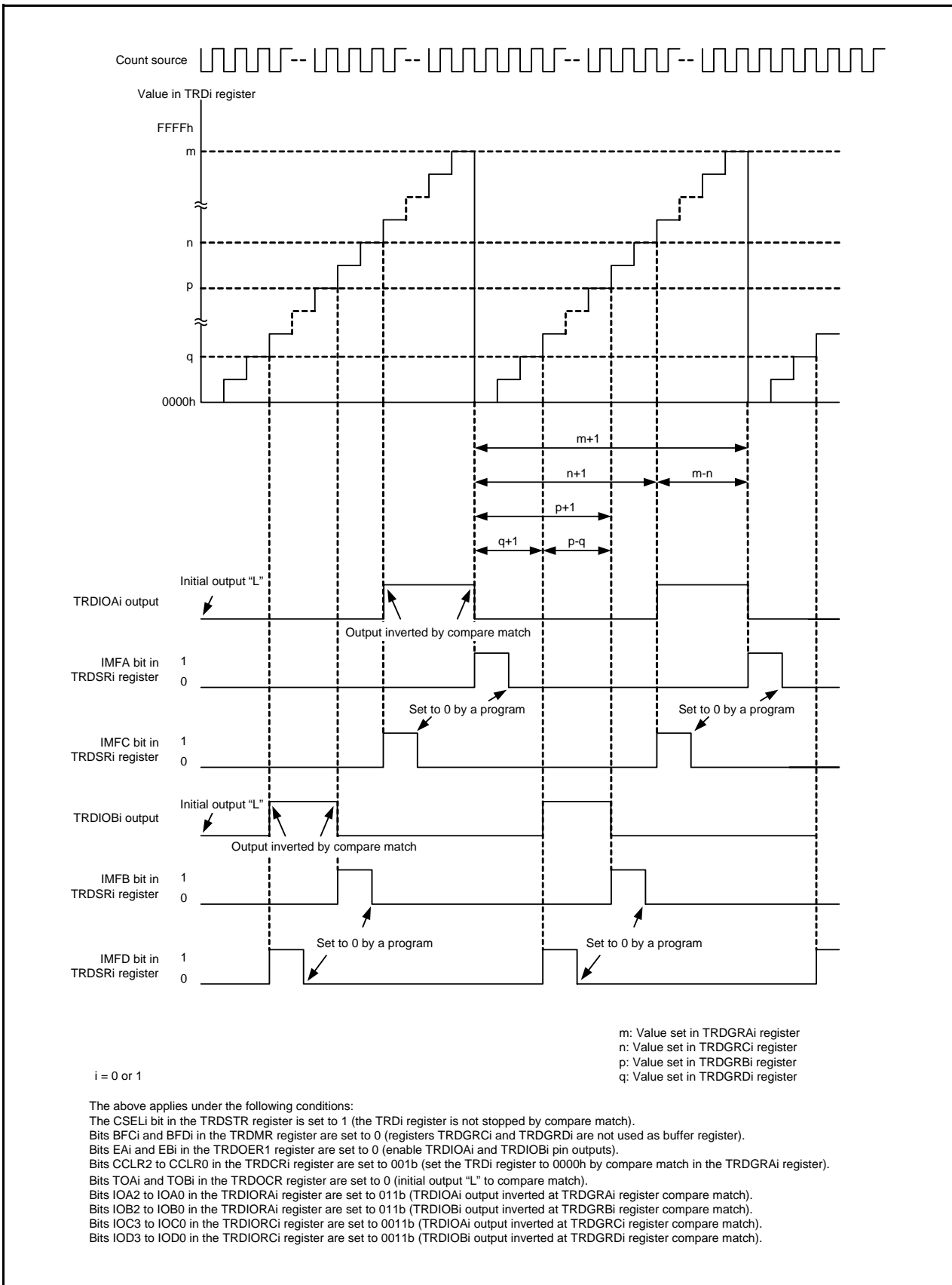


Figure 16.97 Changing Output Pins in Registers TRDGRCi and TRDGRDi

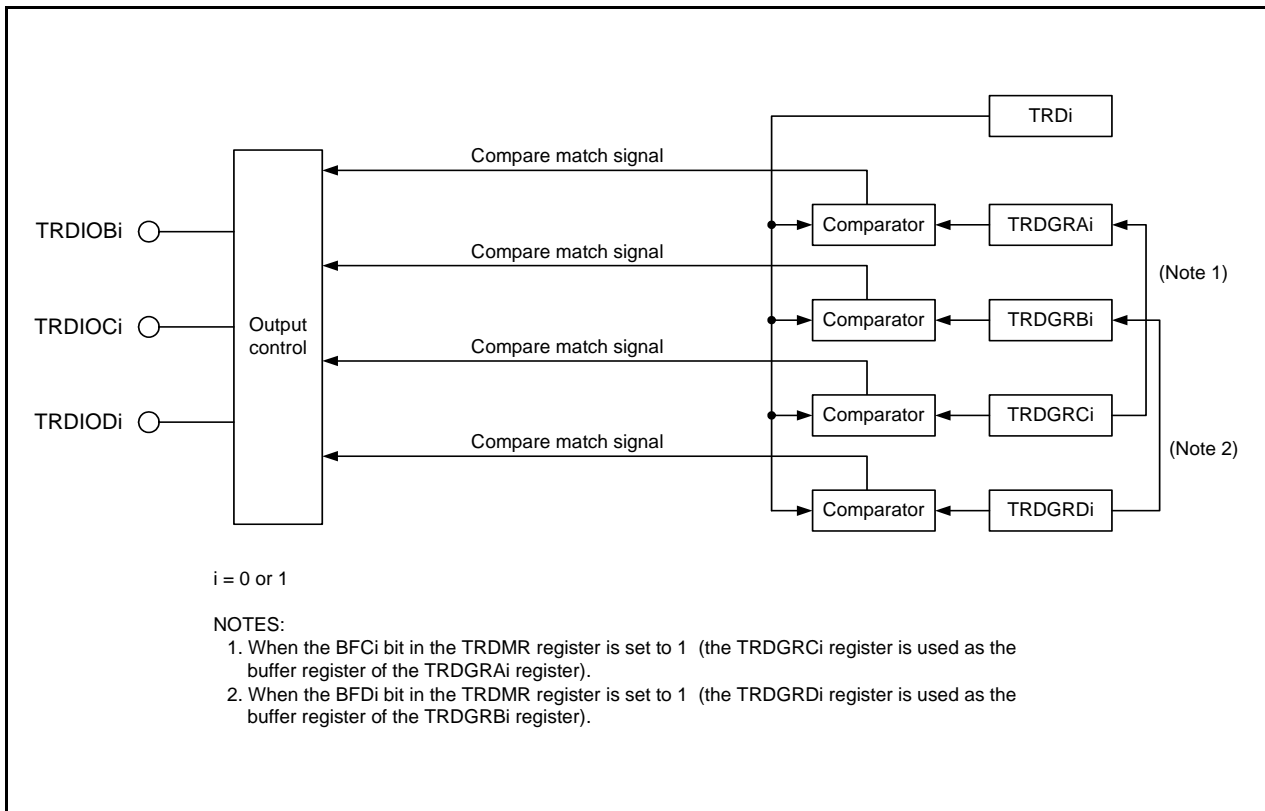


**Figure 16.98 Operating Example When TRDGRCi Register is Used for Output Control of TRDIOAi Pin and TRDGRDi Register is Used for Output Control of TRDIOBi Pin**

### 16.4.7 PWM Mode

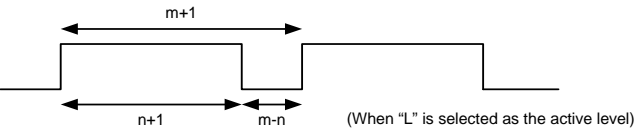
In PWM mode, a PWM waveform is output. Up to 3 PWM waveforms with the same period can be output by 1 channel. Also, up to 6 PWM waveforms with the same period can be output by synchronizing channels 0 and 1. Since this mode functions by a combination of the TRDIO<sub>ji</sub> ( $i = 0$  or  $1$ ,  $j = B, C,$  or  $D$ ) pin and TRDGR<sub>ji</sub> register, the PWM mode, or any other mode or function, can be selected for each individual 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 16.99 shows a Block Diagram of PWM Mode, and Table 16.41 lists the PWM Mode Specifications. Figures 16.100 to 16.109 show the Registers Associated with PWM Mode, and Figures 16.110 and 16.111 show Operating Examples of PWM Mode.



**Figure 16.99 Block Diagram of PWM Mode**

**Table 16.41 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/fk \times (m+1)</math>            Active level width: <math>1/fk \times (m-n)</math>            Inactive level width: <math>1/fk \times (n+1)</math>            fk: Frequency of count source            m: Value set in the TRDGRAi (i = 0 or 1) register            n: Value set in the TRDGRji (j = B, C, or D) register</p> 
Count start condition	1 (count starts) is written to the TSTARTi bit in the TRDSTR register.
Count stop conditions	<ul style="list-style-type: none"> <li>• 0 (count stops) is written to the TSTARTi bit in the TRDSTR register when the CSELi bit in the TRDSTR register is set to 1. The PWM 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 PWM output pin holds level after output change by compare match.</li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• Compare match (The content of the TRDi register matches content of the TRDGRji register.)</li> <li>• TRDi 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 (selectable by pin)
$\overline{\text{INT0}}$ 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 TRDi register.
Write to timer	The value can be written to the TRDi register.
Select functions	<ul style="list-style-type: none"> <li>• 1 to 3 PWM output pins selected per 1 channel            Either 1 pin or multiple pins of the TRDIOBi, TRDIOCi or TRDIODi pin.</li> <li>• The active level selected by pin.</li> <li>• Initial output level selected by pin.</li> <li>• Synchronous operation (Refer to <b>16.4.3 Synchronous Operation.</b>)</li> <li>• Buffer operation (Refer to <b>16.4.2 Buffer Operation.</b>)</li> <li>• Pulse output forced cutoff signal input (Refer to <b>16.4.4 Pulse Output Forced Cutoff.</b>)</li> </ul>

i = 0 or 1

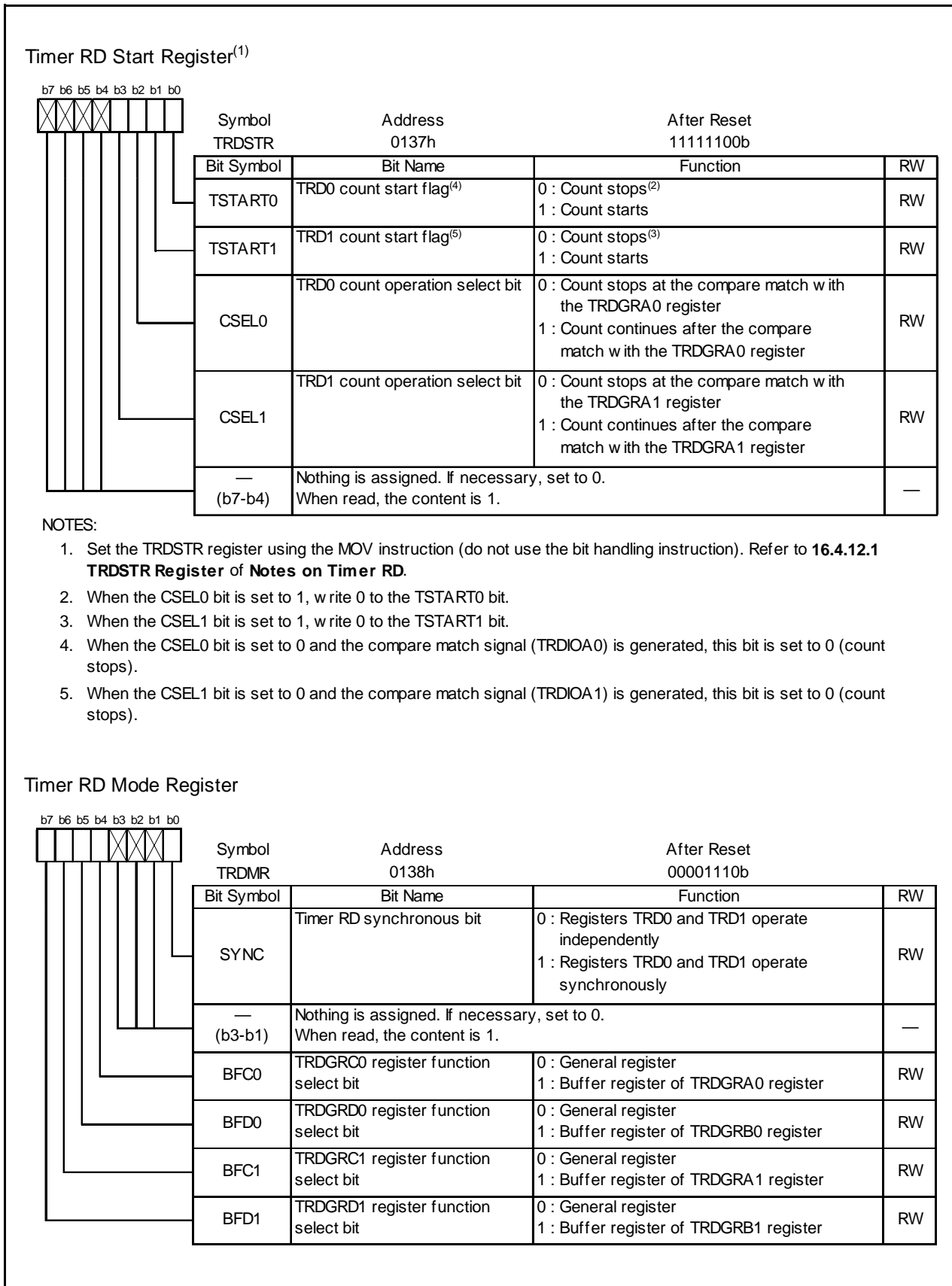


Figure 16.100 Registers TRDSTR and TRDMR in PWM Mode

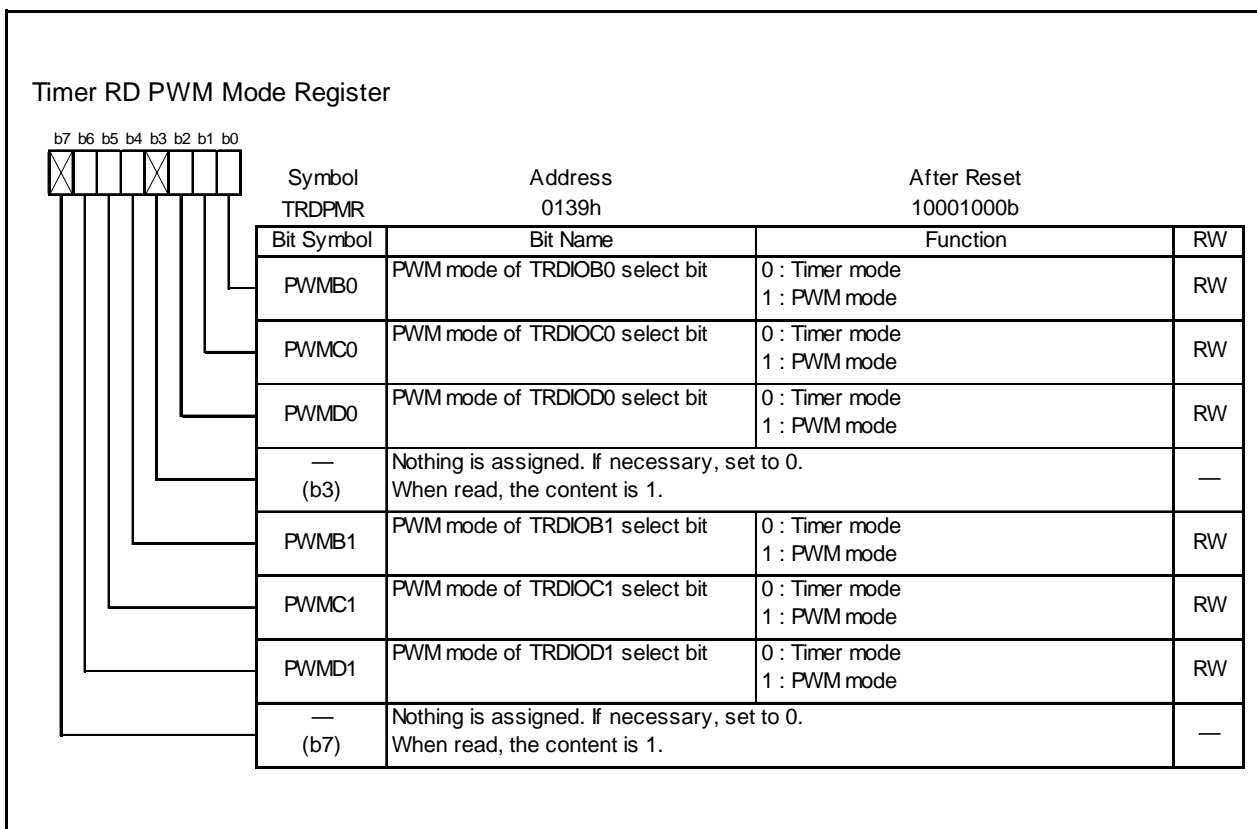


Figure 16.101 TRDPMR Register in PWM Mode



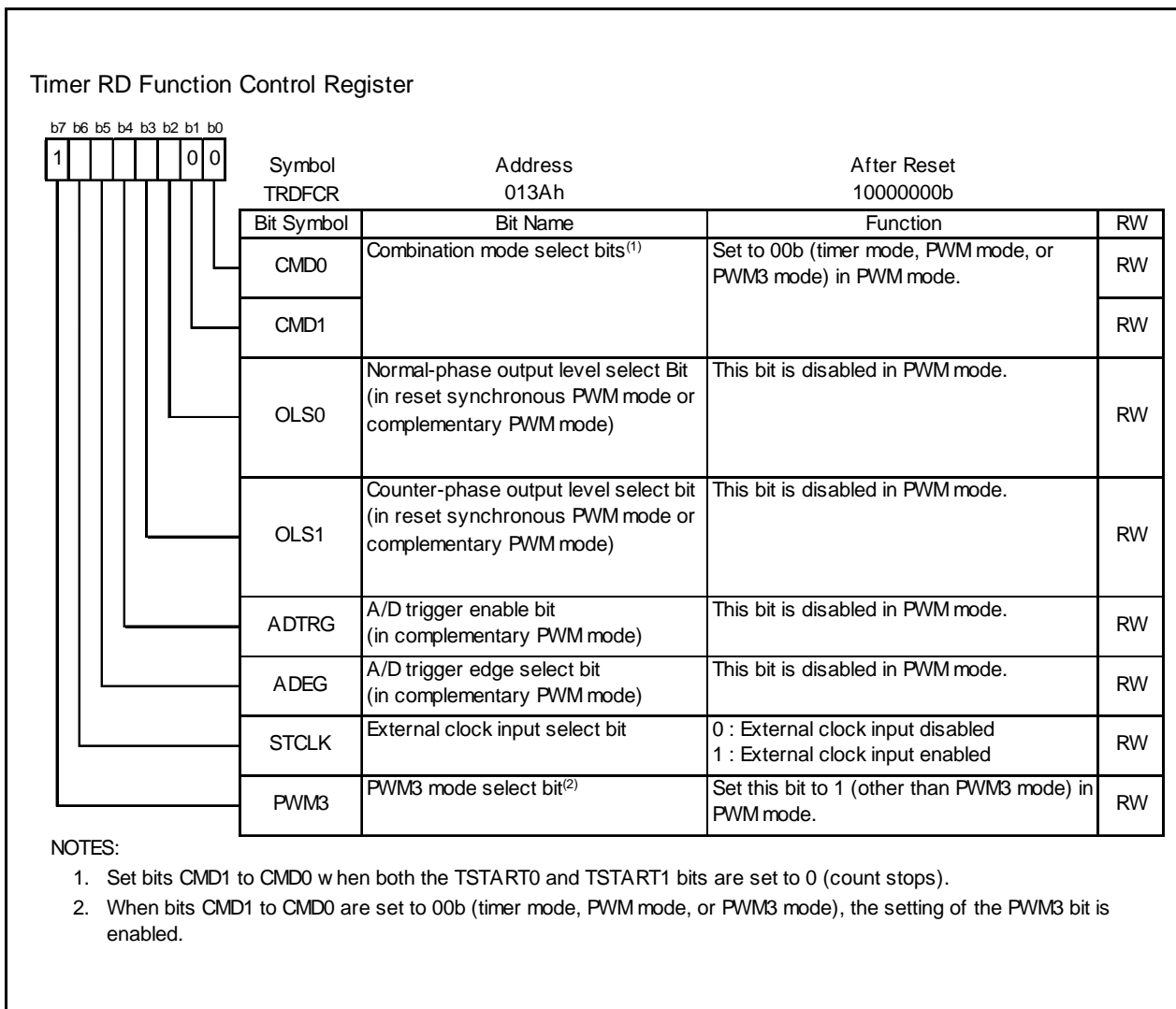


Figure 16.102 TRDFCR Register in PWM Mode

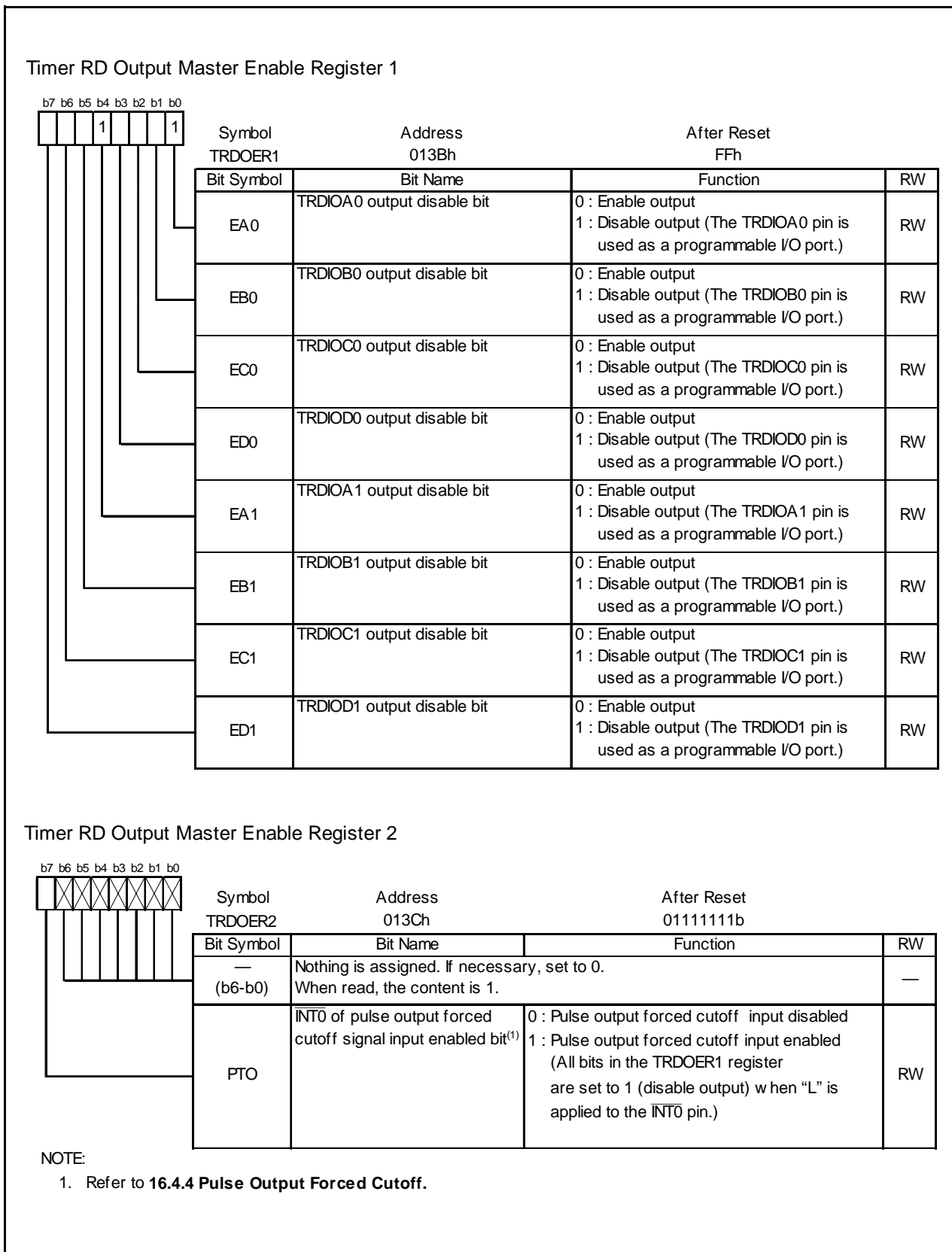


Figure 16.103 Registers TRDOER1 to TRDOER2 in PWM Mode

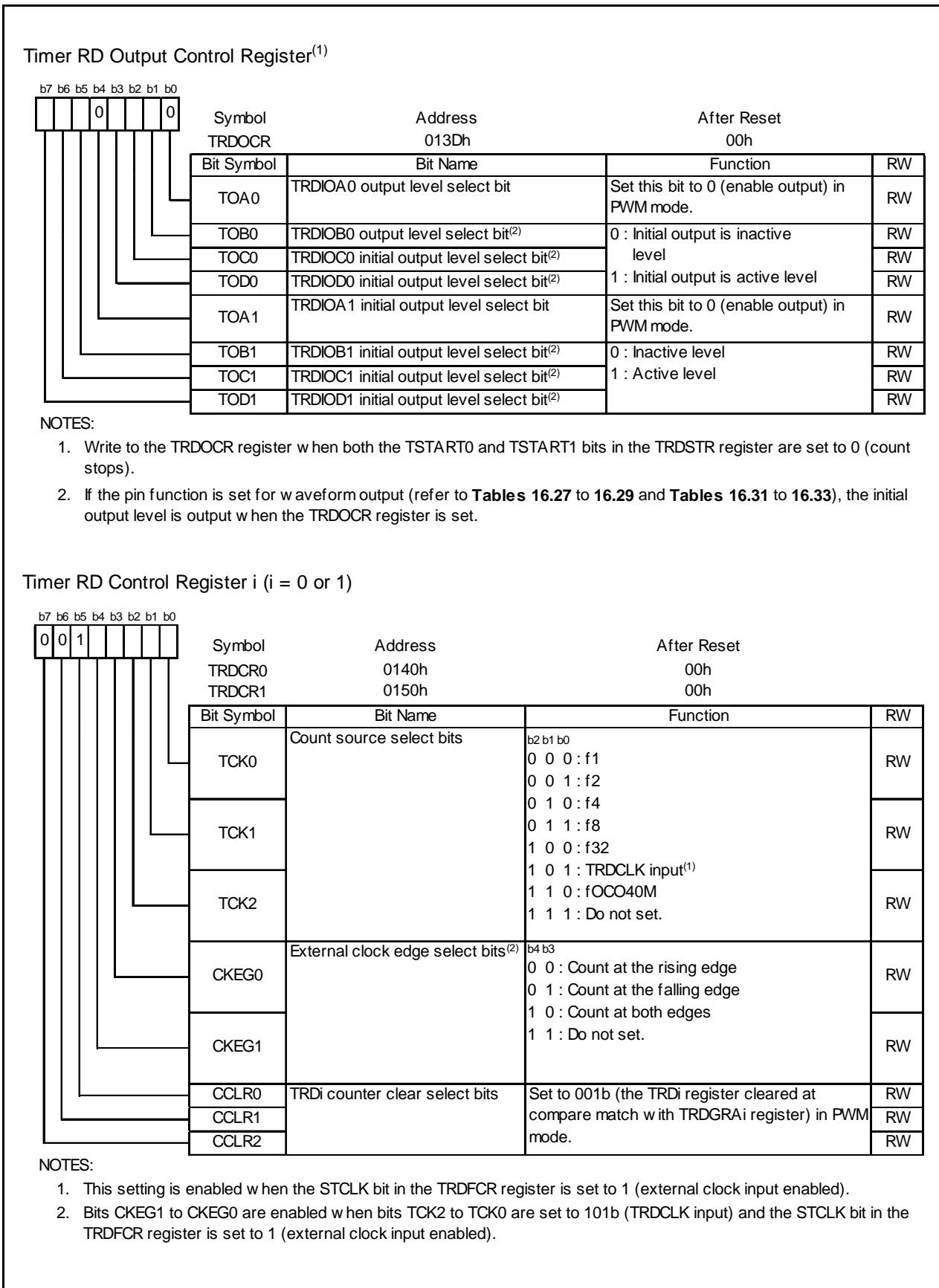


Figure 16.104 Registers TRDOCR and TRDCR0 to TRDCR1 in PWM Mode

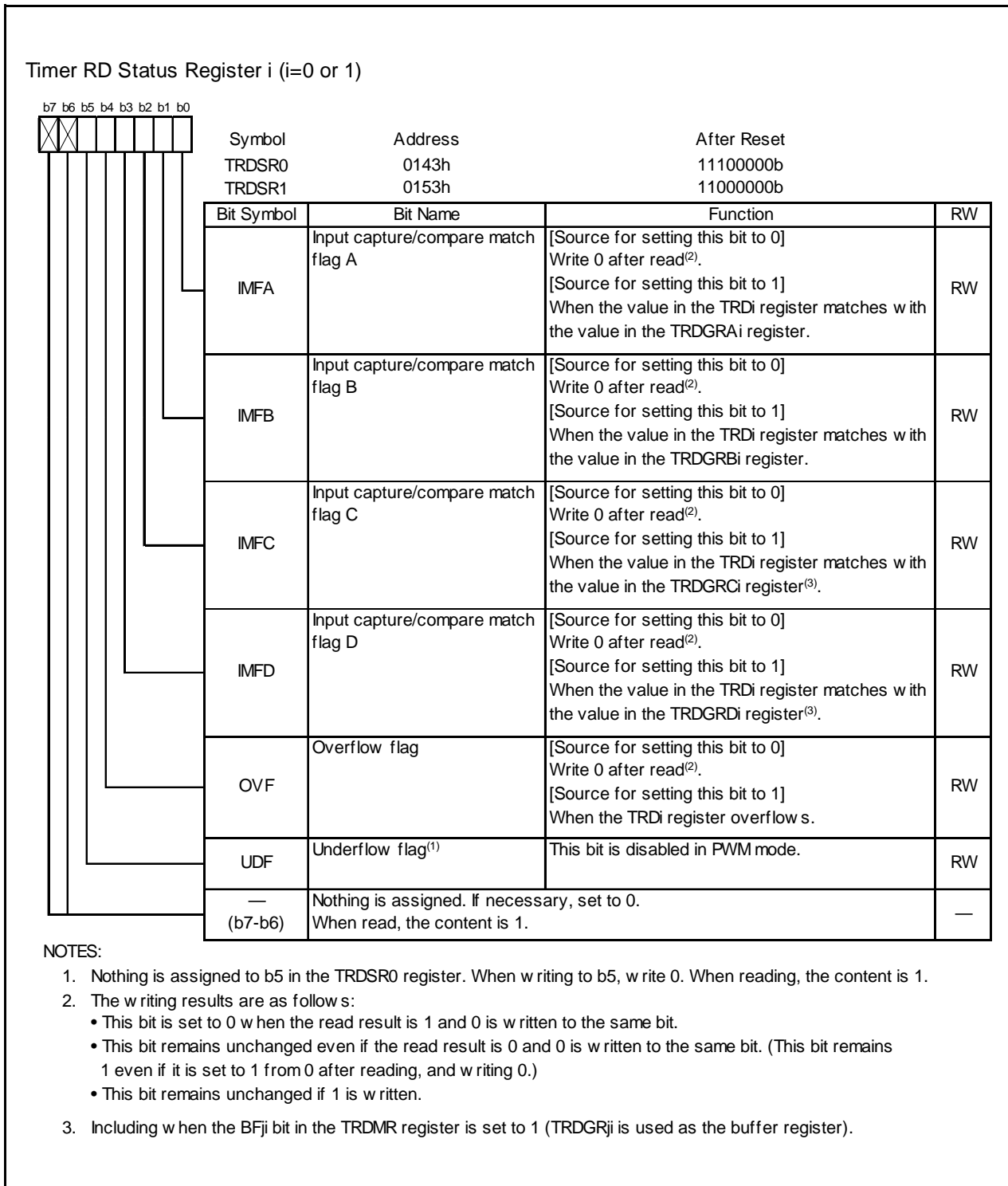


Figure 16.105 Registers TRDSR0 to TRDSR1 in PWM Mode

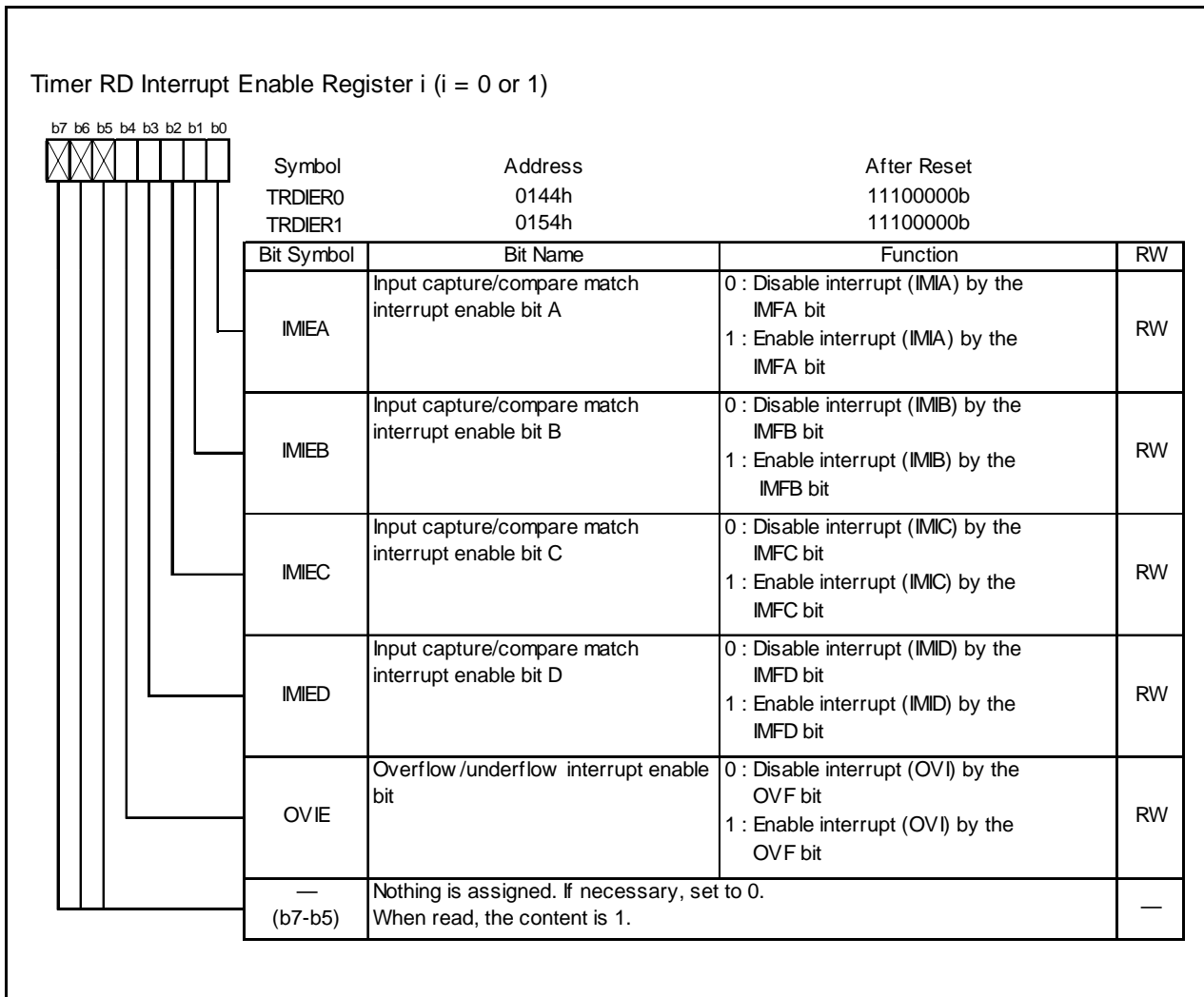


Figure 16.106 Registers TRDIER0 to TRDIER1 in PWM Mode

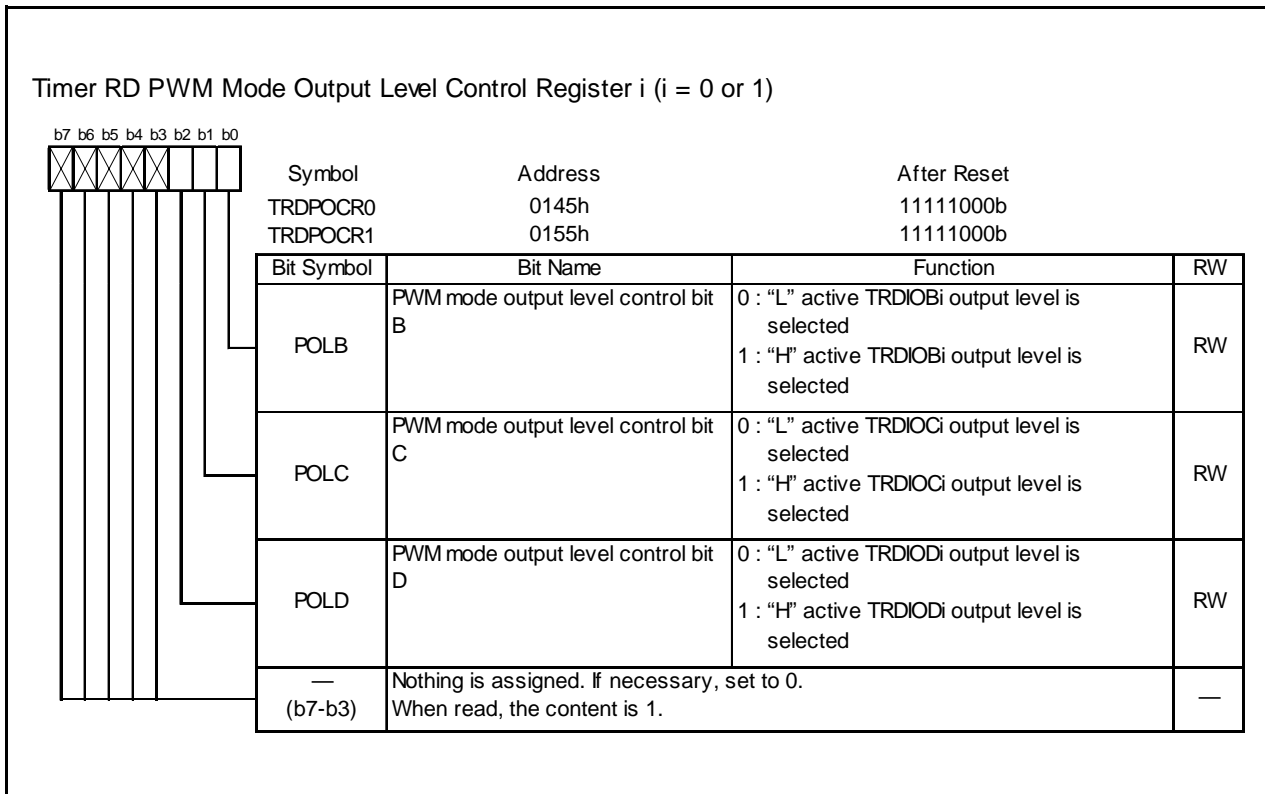


Figure 16.107 Registers TRDPOCR0 to TRDPOCR1 in PWM Mode

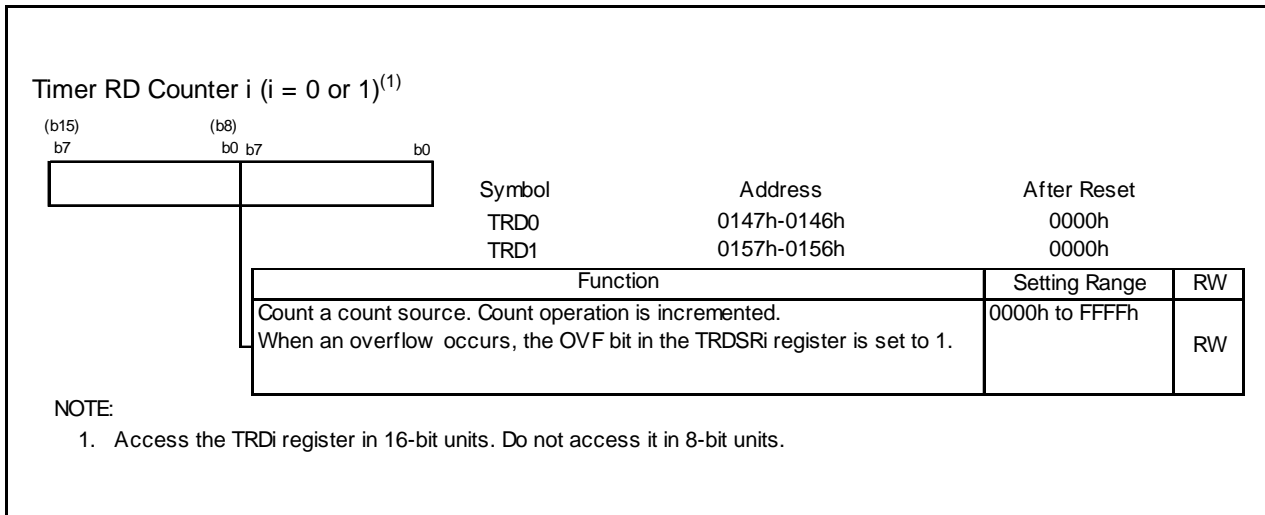


Figure 16.108 Registers TRD0 to TRD1 in PWM Mode

Timer RD General Registers 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 16.42 TRDGRji Register Functions in PWM Mode.</b>					RW

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

**Figure 16.109 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 16.42 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>16.4.2 Buffer Operation.</b> )	–
TRDGRDi	BFDi = 1	Buffer register. Set the changing point of the next PWM output (Refer to <b>16.4.2 Buffer Operation.</b> )	TRDIOBi

i = 0 or 1

BFCi, BFDi: Bits in TRDMR register

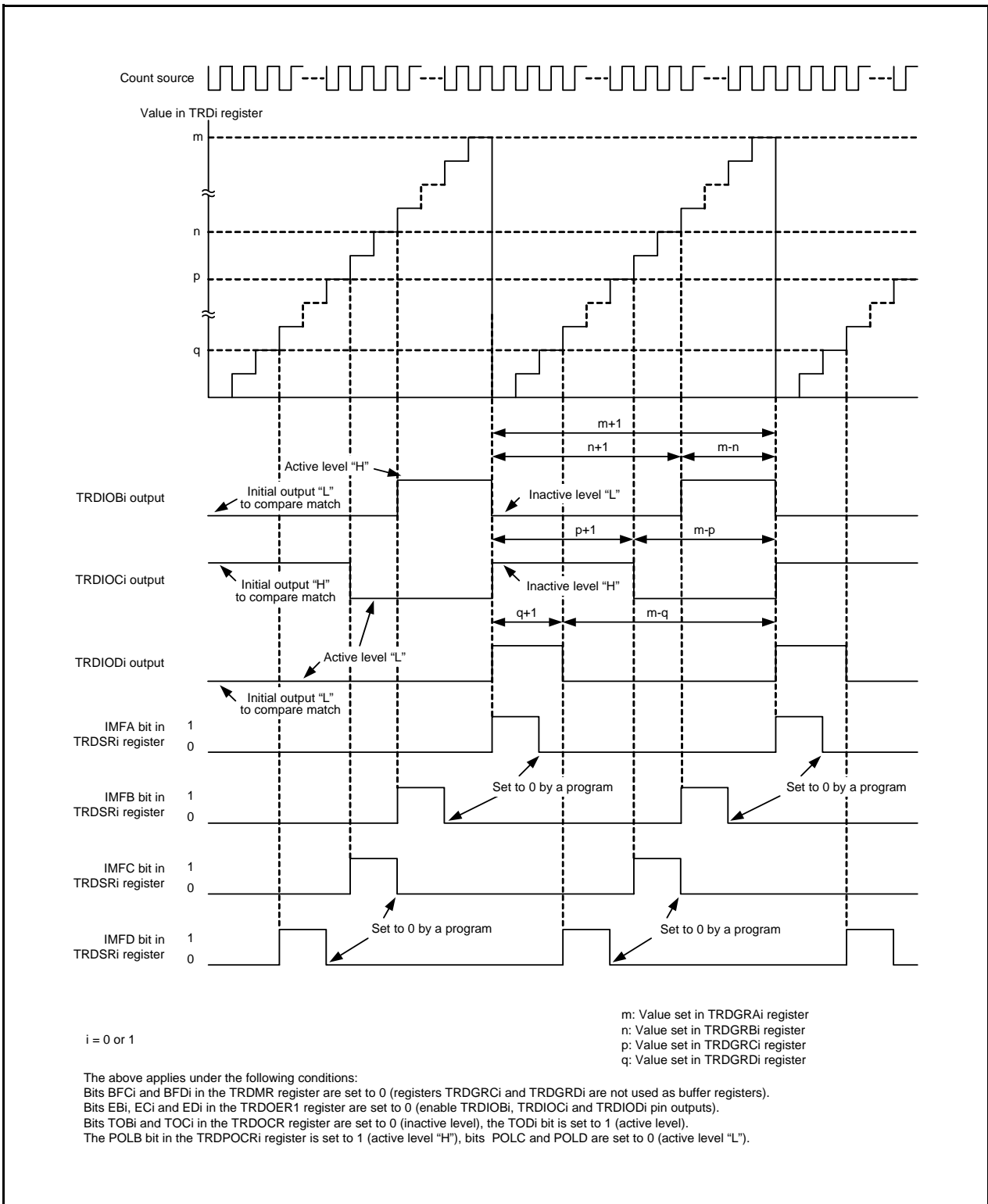


Figure 16.110 Operating Example of PWM Mode



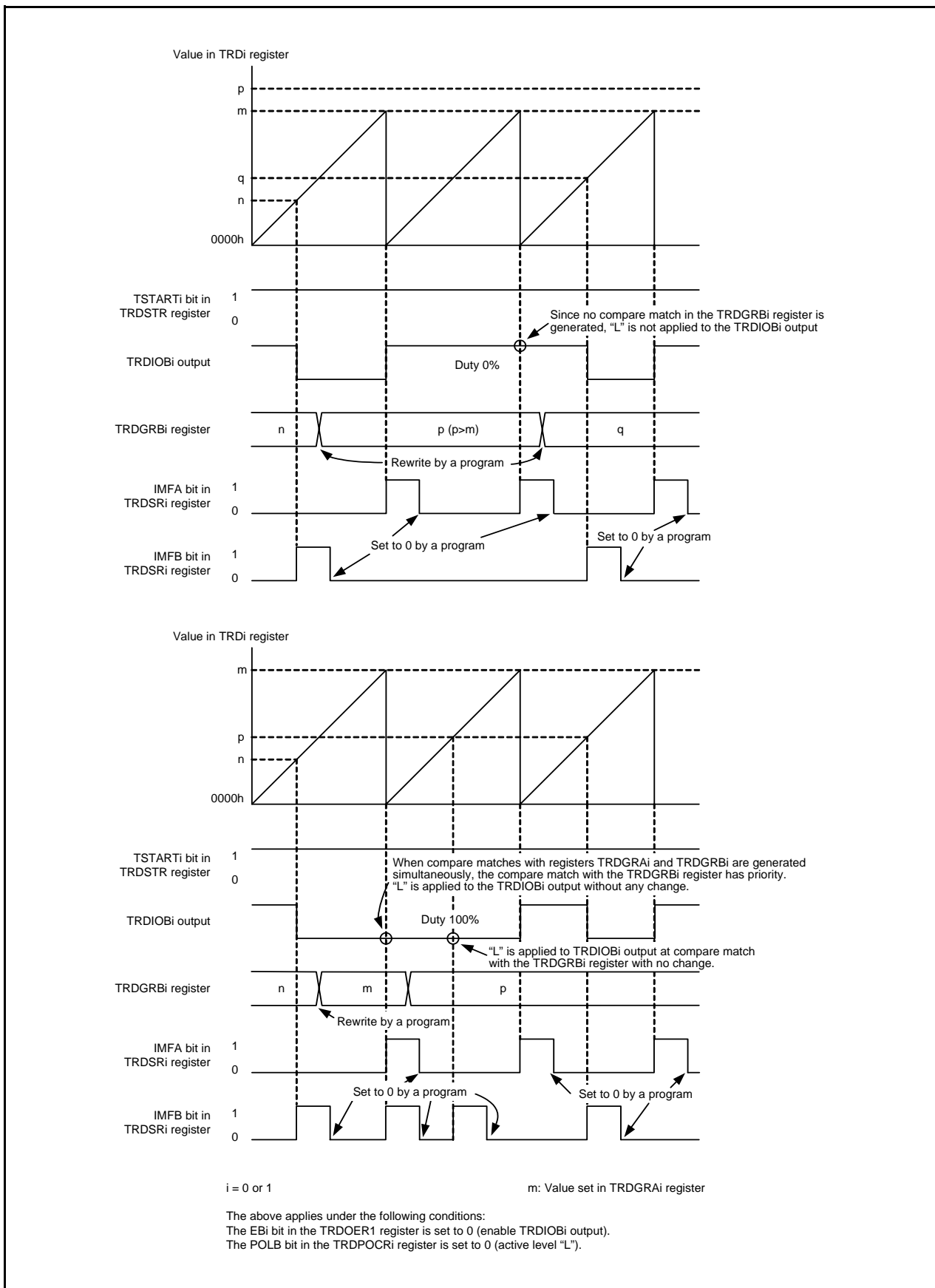
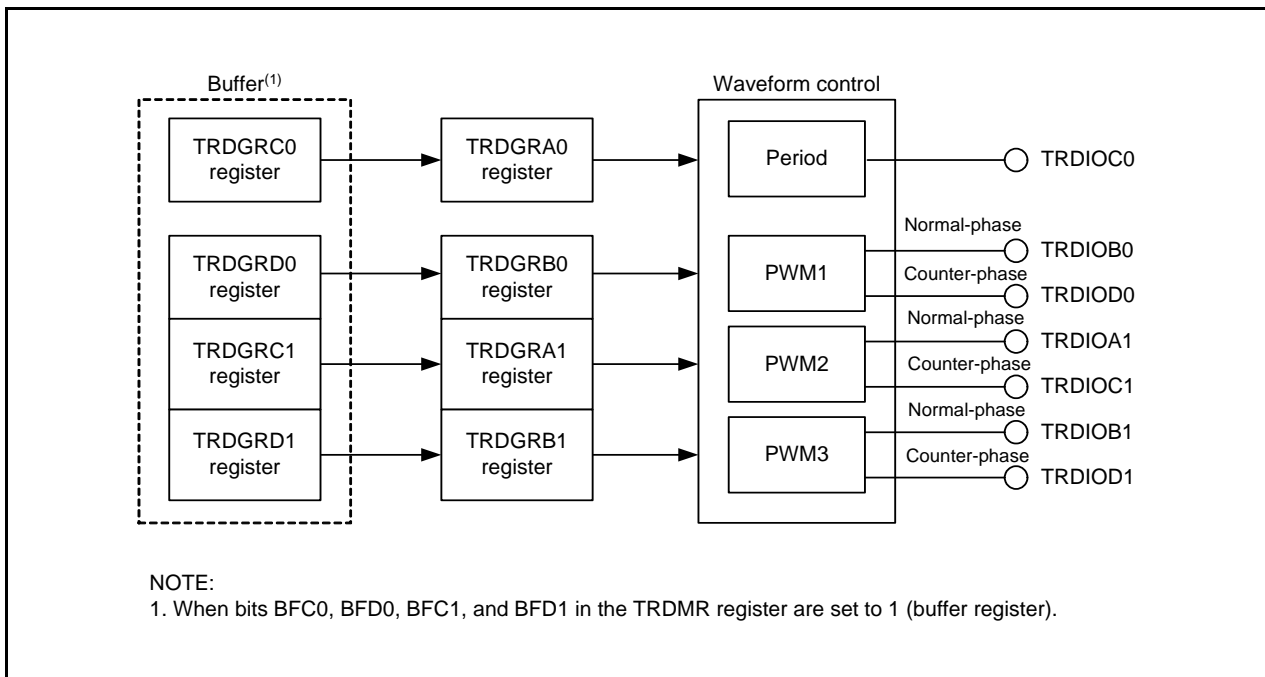


Figure 16.111 Operating Example of PWM Mode (Duty 0%, Duty 100%)

### 16.4.8 Reset Synchronous PWM Mode

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

Figure 16.112 shows a Block Diagram of Reset Synchronous PWM Mode, and Table 16.43 lists the Reset Synchronous PWM Mode Specifications. Figures 16.113 to 16.120 show the Registers Associated with Reset Synchronous PWM Mode and Figure 16.121 shows an Operating Example of Reset Synchronous PWM Mode. Refer to **Figure 16.111 Operating Example of PWM Mode (Duty 0%, Duty 100%)** for an operating example of PWM Mode with duty 0% and duty 100%.



**Figure 16.112 Block Diagram of Reset Synchronous PWM Mode**

**Table 16.43 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>: Value set in the TRDGRA0 register  <math>n</math>: Value set in the TRDGRB0 register (PWM1 output),  Value set in the TRDGRA1 register (PWM2 output),  Value set in the TRDGRB1 register (PWM3 output)</p> <p>(When "L" is selected as the active level)</p>
Count start condition	1 (count starts) is written to the TSTART0 bit in the TRDSTR register.
Count stop conditions	<ul style="list-style-type: none"> <li>• 0 (count stops) is written to the TSTART0 bit 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</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. The PWM output pin holds level after output change at compare match.</li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• Compare match (the content of the TRD0 register matches content of registers TRDGRj0, TRDGRA1, and TRDGRB1).</li> <li>• The TRD0 register overflows</li> </ul>
TRDIOA0 pin function	Programmable I/O port or TRDCLK (external clock) input
TRDIOB0 pin function	PWM1 output normal-phase output
TRDIOD0 pin function	PWM1 output counter-phase output
TRDIOA1 pin function	PWM2 output normal-phase output
TRDIOC1 pin function	PWM2 output counter-phase output
TRDIOB1 pin function	PWM3 output normal-phase output
TRDIOD1 pin function	PWM3 output counter-phase output
TRDIOC0 pin function	Output inverted every PWM period
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.
Select 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>16.4.2 Buffer Operation.</b>)</li> <li>• Pulse output forced cutoff signal input (Refer to <b>16.4.4 Pulse Output Forced Cutoff.</b>)</li> </ul>

j = either A, B, C, or D

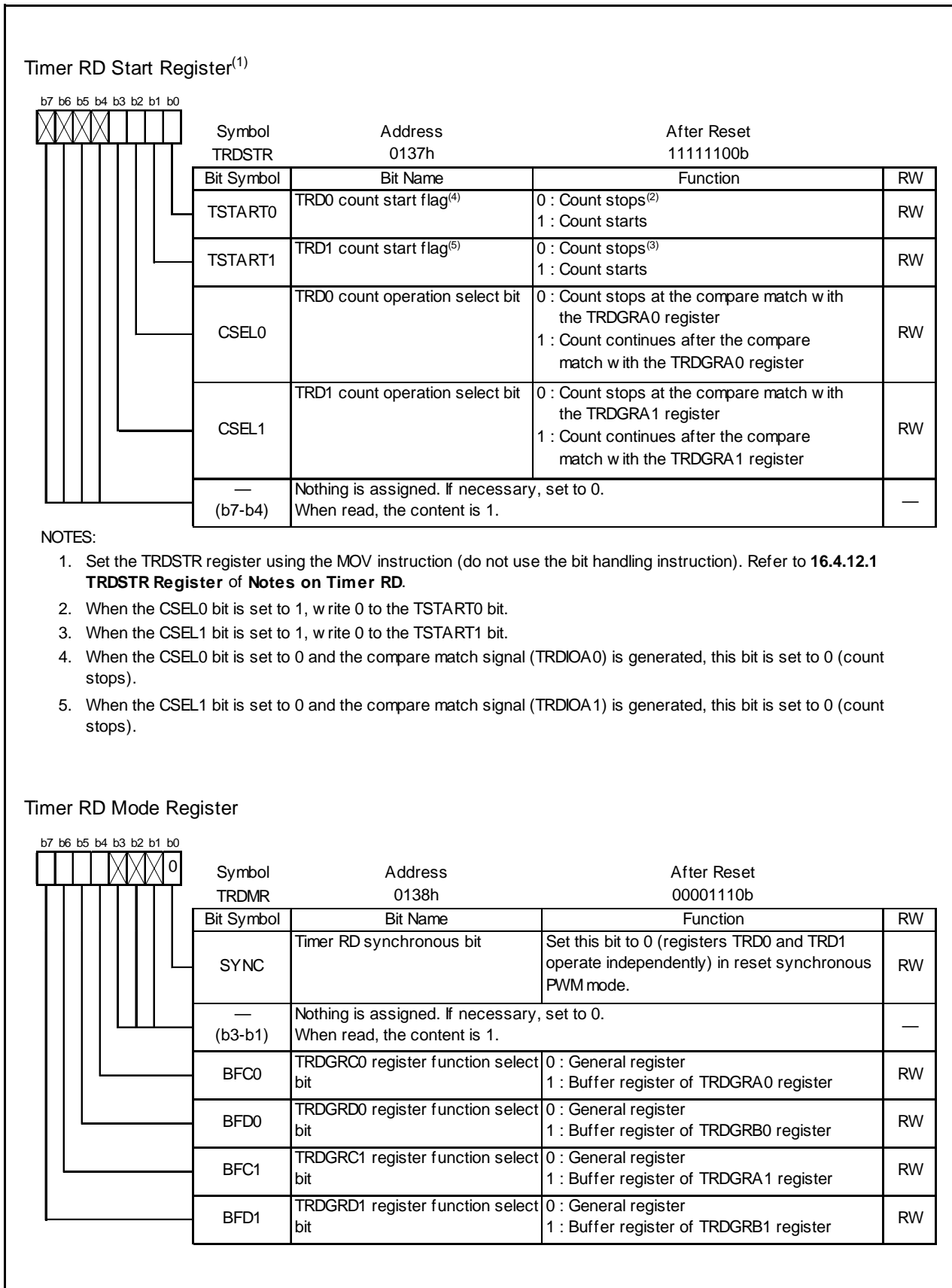


Figure 16.113 Registers TRDSTR and TRDMR in Reset Synchronous PWM Mode

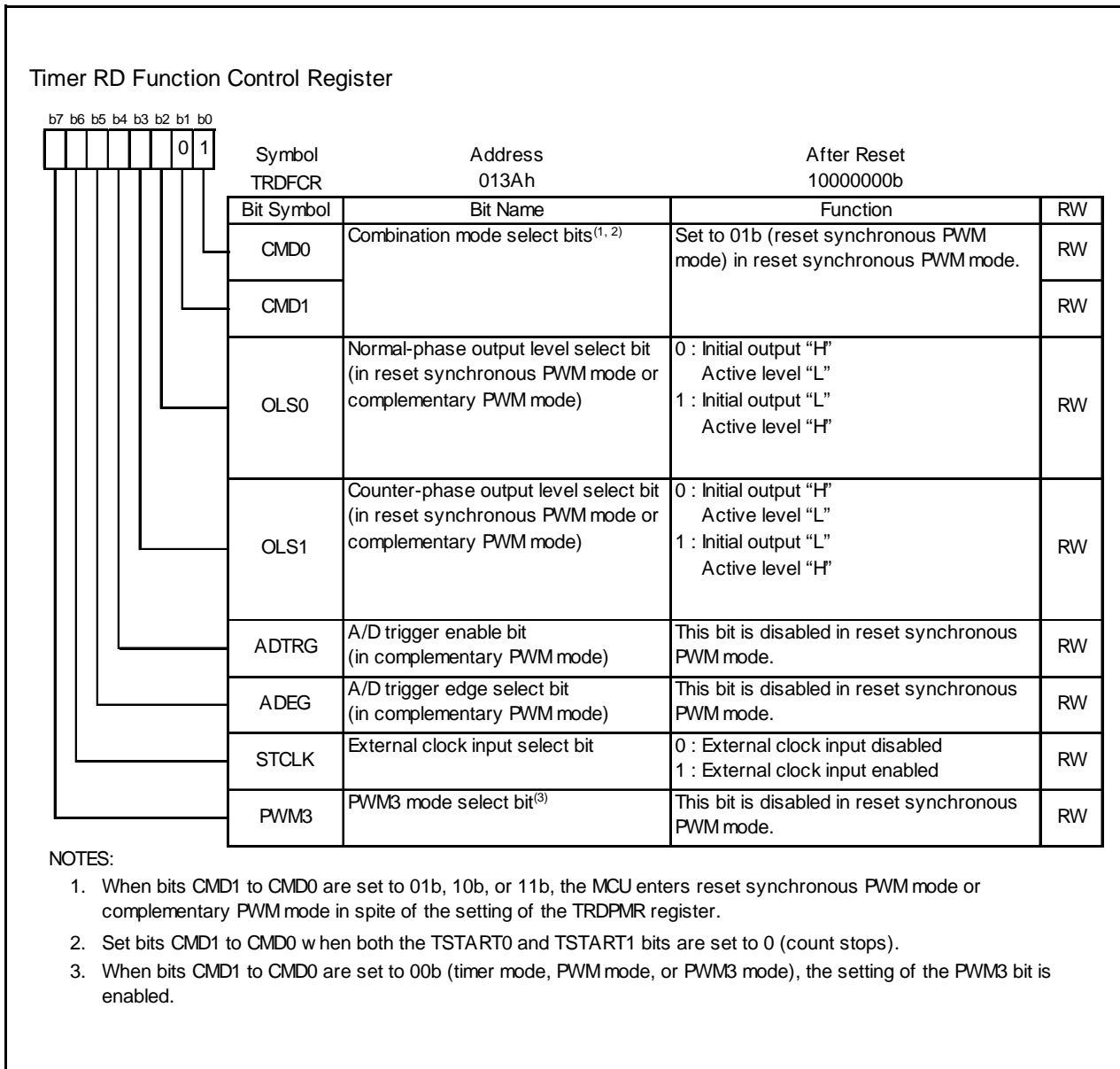


Figure 16.114 TRDFCR Register in Reset Synchronous PWM Mode

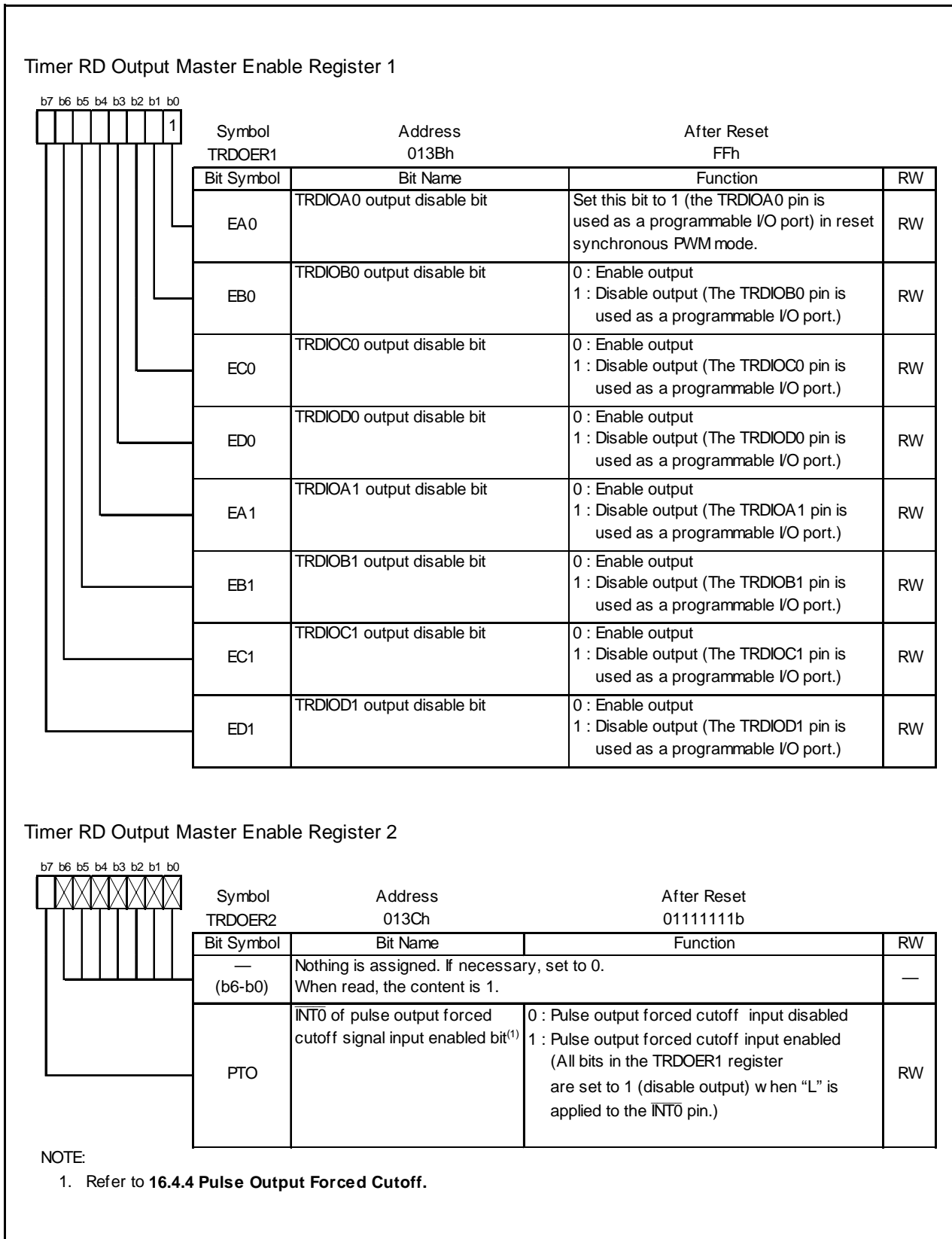


Figure 16.115 Registers TRDOER1 to TRDOER2 in Reset Synchronous PWM Mode

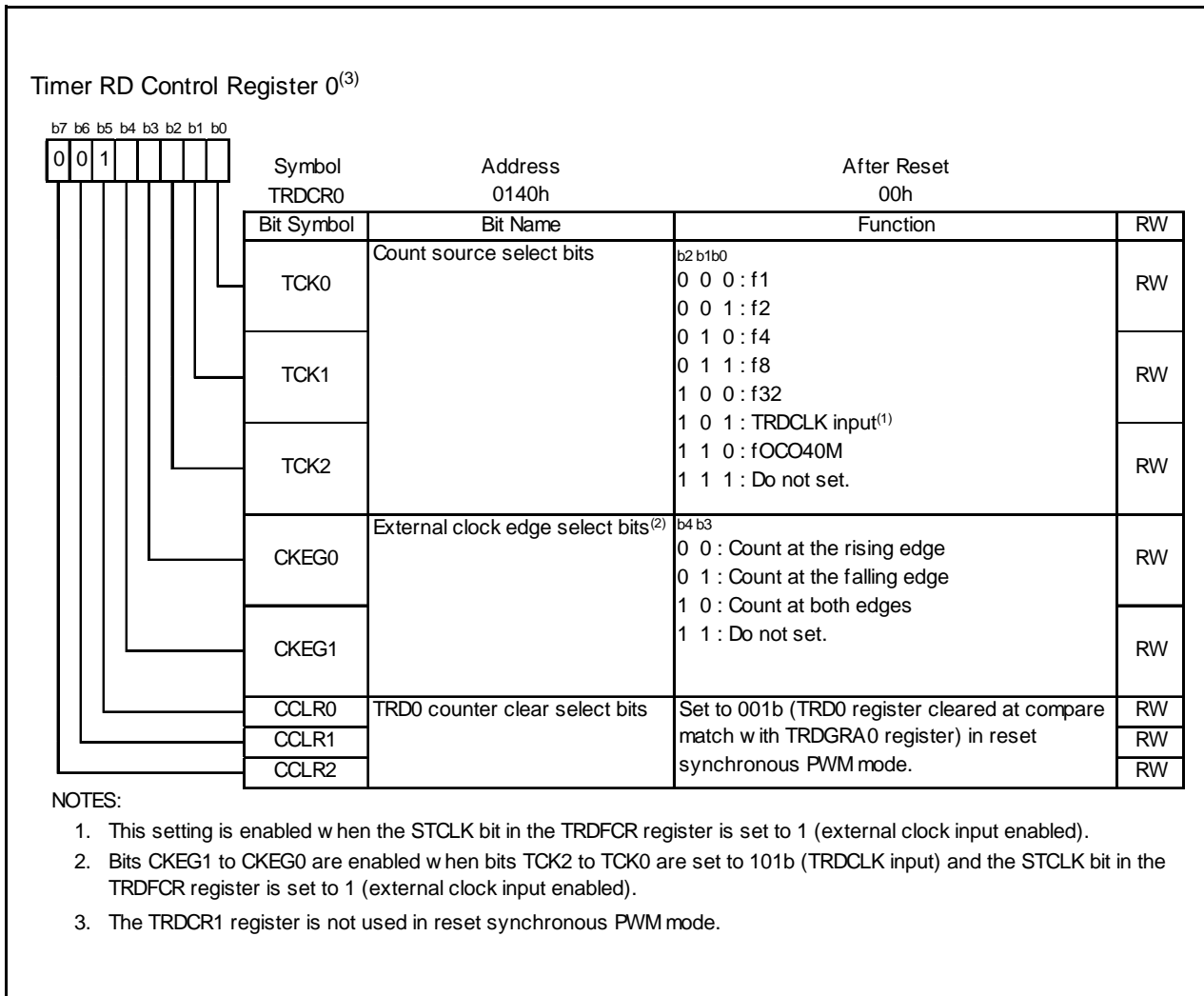


Figure 16.116 TRDCR0 Register in Reset Synchronous PWM Mode

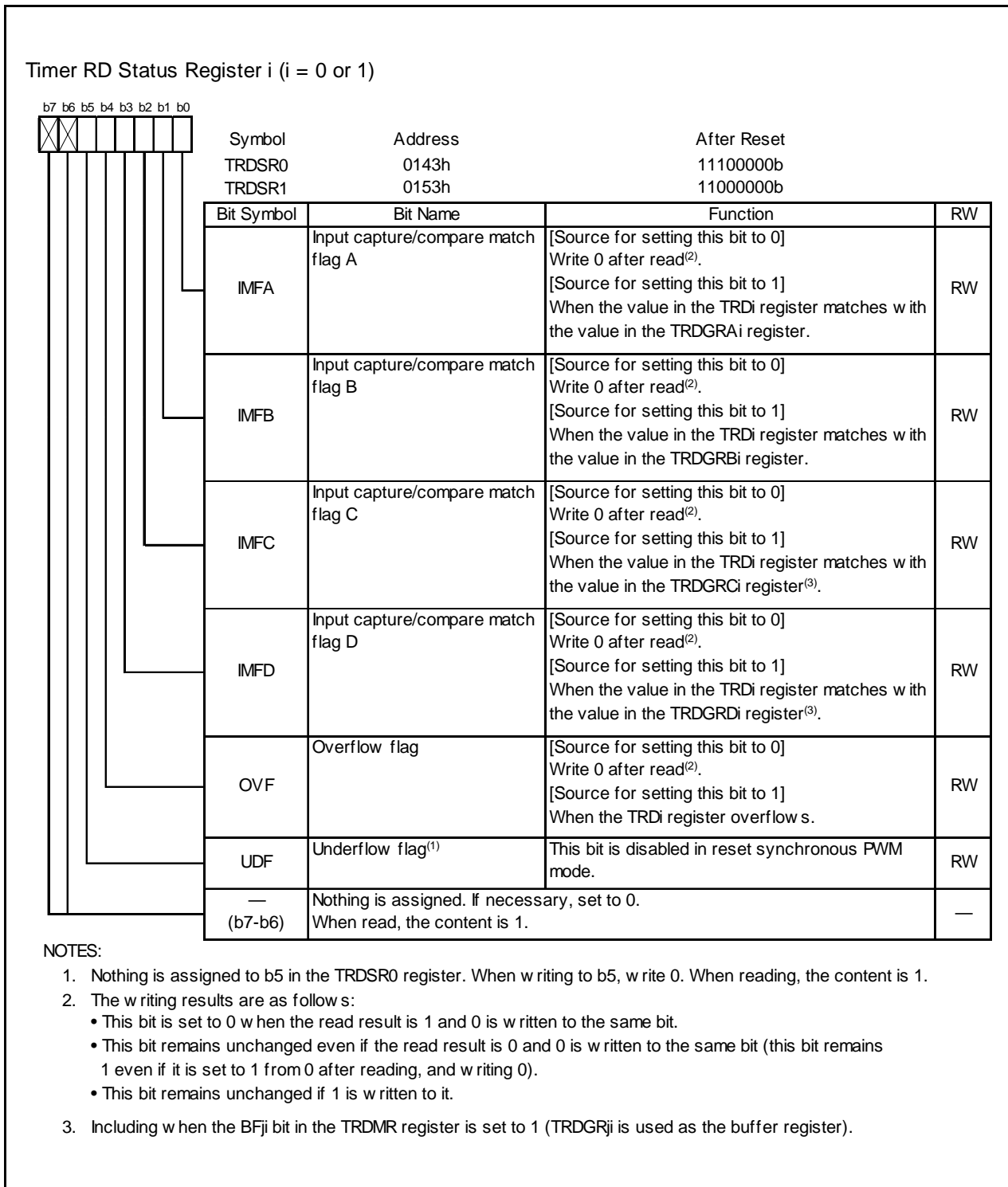


Figure 16.117 Registers TRDSR0 to TRDSR1 in Reset Synchronous PWM Mode



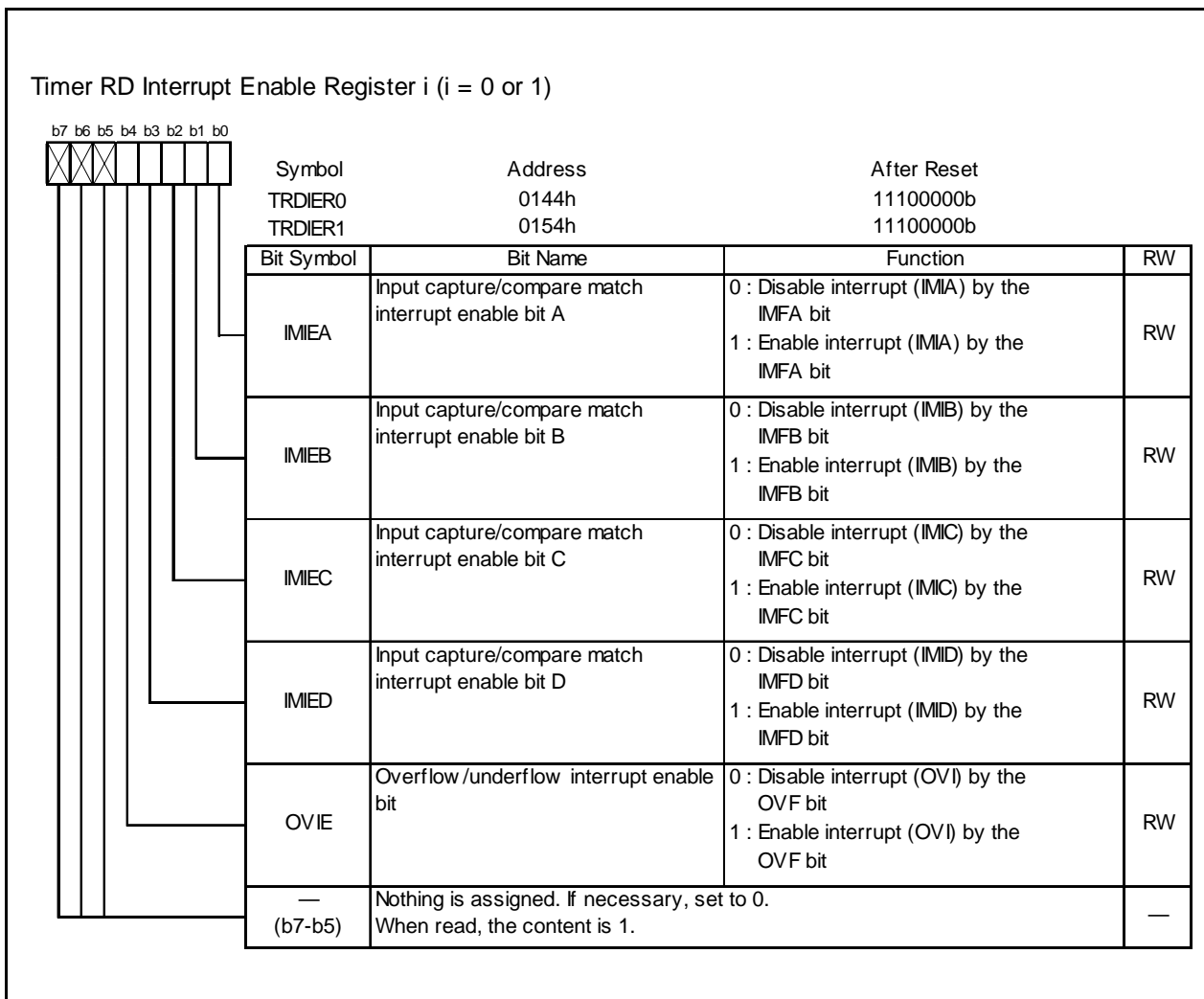


Figure 16.118 Registers TRDIER0 to TRDIER1 in Reset Synchronous PWM Mode

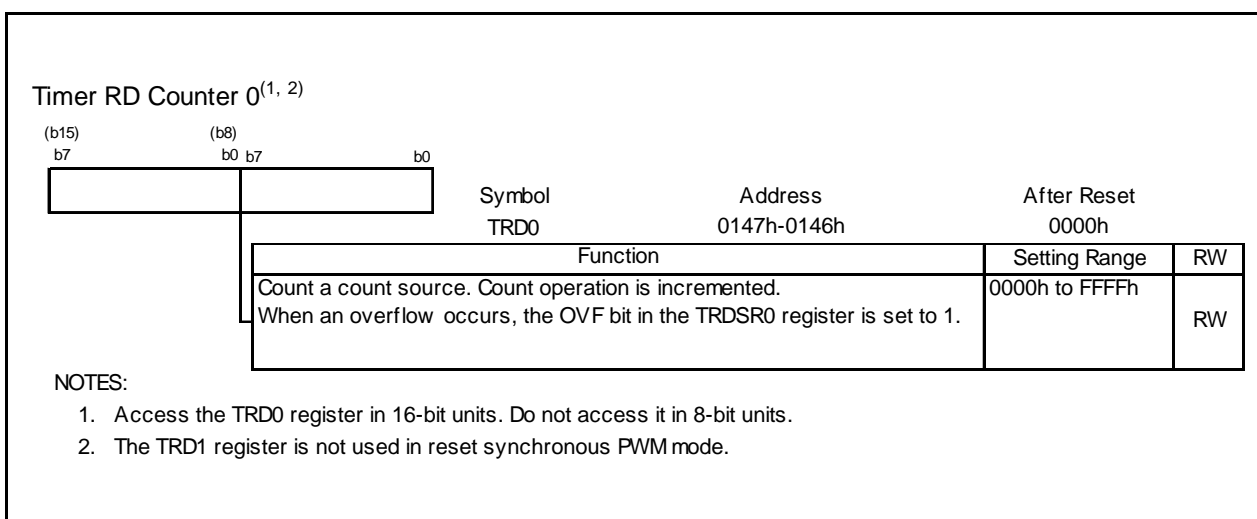


Figure 16.119 TRD0 Registrar in Reset Synchronous PWM Mode

Timer RD General Registers Ai, Bi, Ci, and Di (i = 0 or 1) <sup>(1)</sup>		Symbol	Address	After Reset
(b15) b7	(b8) b0 b7 b0			
		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 16.44 TRDGRji Register Function in Reset Synchronous PWM Mode.</b>				RW

NOTE:

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

**Figure 16.120 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, TRDIOA0, TRDIORC0, TRDPOCR0, TRDIOA1, TRDIORC1, and TRDPOCR1.

**Table 16.44 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 PWM period and TRDIOC0 pin)
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>16.4.2 Buffer Operation.</b> )	(Output inverted every PWM period and TRDIOC0 pin)
TRDGRD0	BFD0 = 1	Buffer register. Set the changing point of the next PWM1 output. (Refer to <b>16.4.2 Buffer Operation.</b> )	TRDIOB0 TRDIOD0
TRDGRC1	BFC1 = 1	Buffer register. Set the changing point of the next PWM2 output. (Refer to <b>16.4.2 Buffer Operation.</b> )	TRDIOA1 TRDIOC1
TRDGRD1	BFD1 = 1	Buffer register. Set the changing point of the next PWM3 output. (Refer to <b>16.4.2 Buffer Operation.</b> )	TRDIOB1 TRDIOD1

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

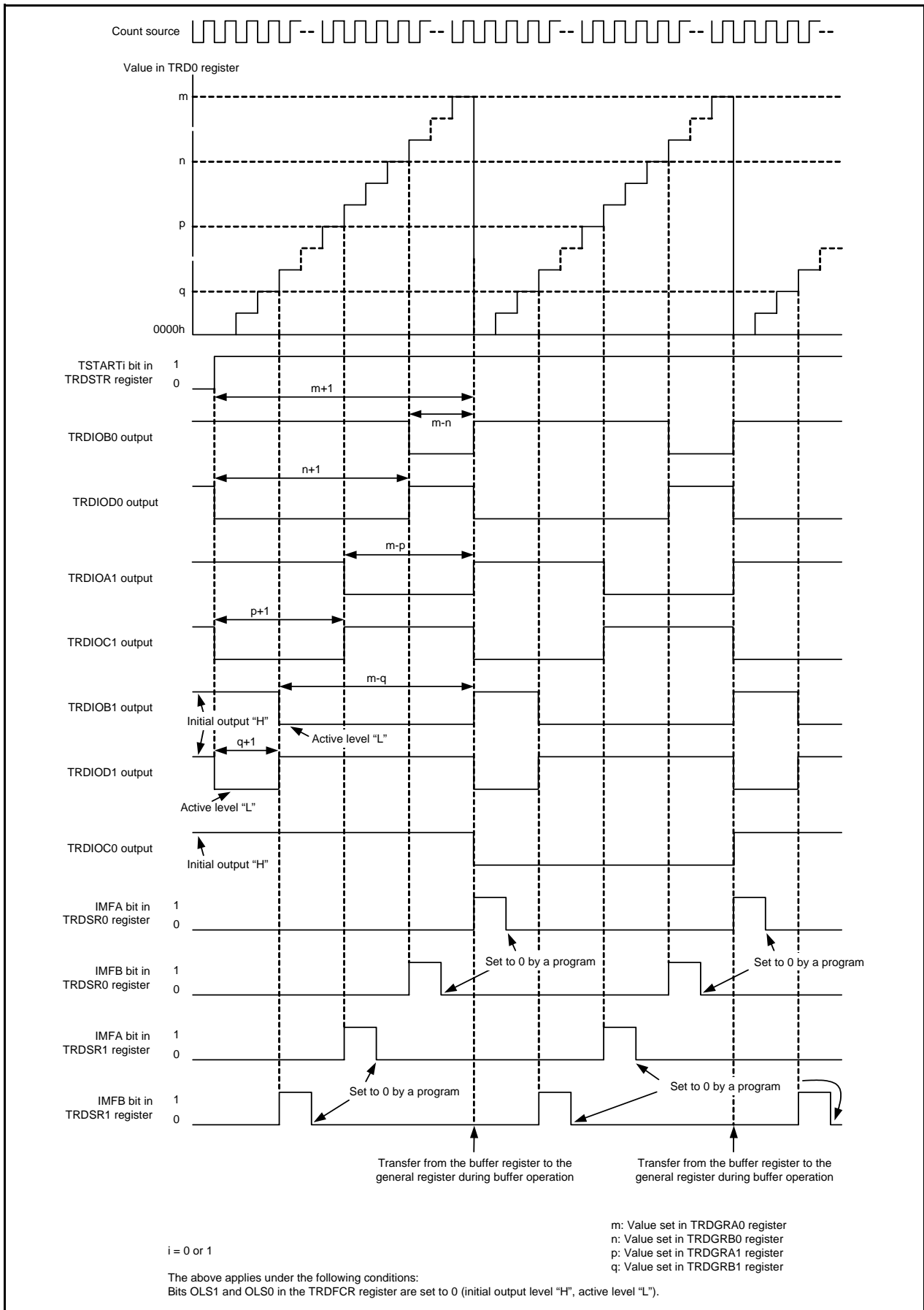


Figure 16.121 Operating Example of Reset Synchronous PWM Mode

### 16.4.9 Complementary PWM Mode

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

Figure 16.122 shows a Block Diagram of Complementary PWM Mode, and Table 16.45 lists the Complementary PWM Mode Specifications. Figures 16.123 to 16.131 show the Registers Associated with Complementary PWM Mode, Figure 16.132 shows the Output Model of Complementary PWM Mode, and Figure 16.133 shows an Operating Example of Complementary PWM Mode.

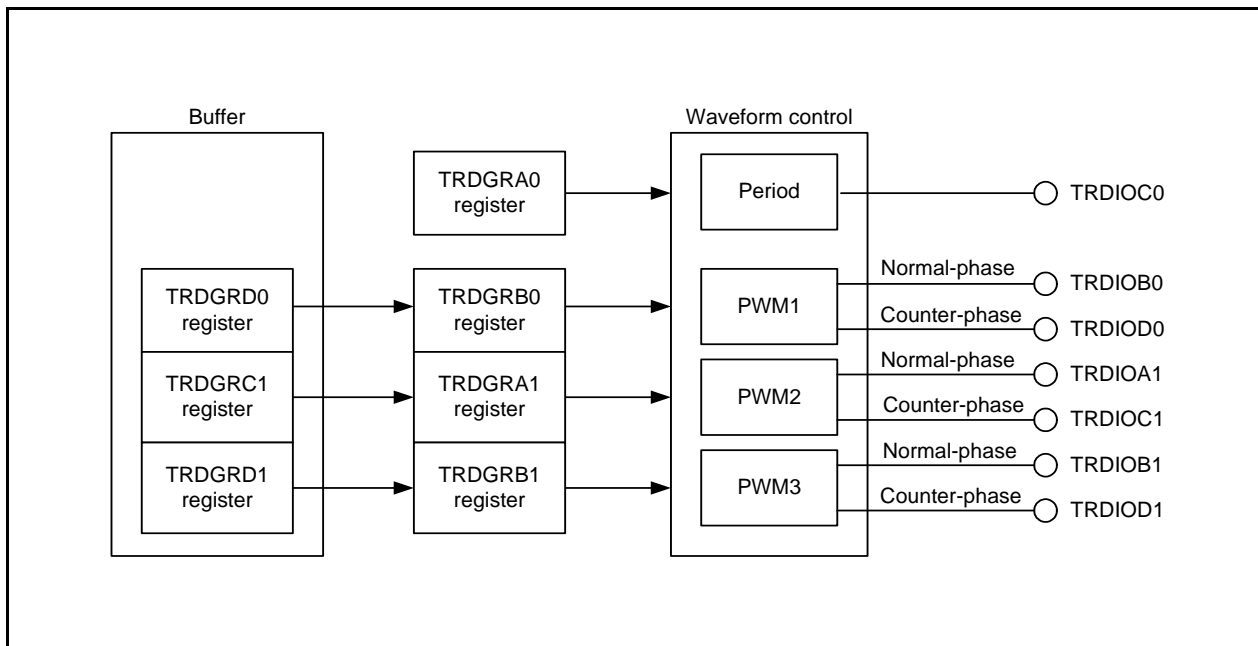
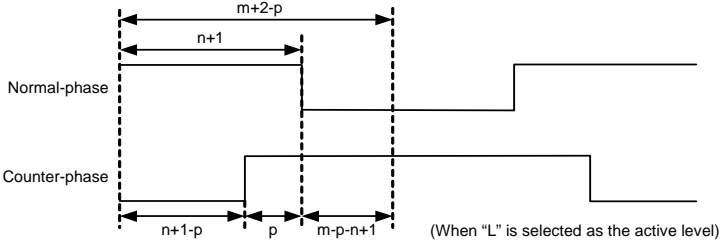


Figure 16.122 Block Diagram of Complementary PWM Mode

**Table 16.45 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 bits TCK2 to TCK0 in the TRDCR1 register to the same value (same count source) as bits TCK2 to TCK0 in the TRDCR0 register.
Count operations	Increment or decrement Registers TRD0 and TRD1 are decremented with the compare match in registers TRD0 and TRDGRA0 during increment operation. The TRD1 register value is changed from 0000h to FFFFh during decrement operation, and registers TRD0 and TRD1 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: Value set in the TRDGRA0 register n: Value set in the TRDGRB0 register (PWM1 output) Value set in the TRDGRA1 register (PWM2 output) Value set in the TRDGRB1 register (PWM3 output) p: Value set in the TRD0 register  (When "L" is selected as the active level)
Count start condition	1 (count starts) is written to bits TSTART0 and TSTART1 in the TRDSTR register.
Count stop conditions	0 (count stops) is written to bits TSTART0 and TSTART1 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 of the TRDi register matches content of the TRDGRji register.)</li> <li>The TRD1 register underflows</li> </ul>
TRDIOA0 pin function	Programmable I/O port or TRDCLK (external clock) input
TRDIOB0 pin function	PWM1 output normal-phase output
TRDIOD0 pin function	PWM1 output counter-phase output
TRDIOA1 pin function	PWM2 output normal-phase output
TRDIOC1 pin function	PWM2 output counter-phase output
TRDIOB1 pin function	PWM3 output normal-phase output
TRDIOD1 pin function	PWM3 output counter-phase output
TRDIOC0 pin function	Output inverted 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.
Select functions	<ul style="list-style-type: none"> <li>Pulse output forced cutoff signal input (Refer to <b>16.4.4 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 fixed.

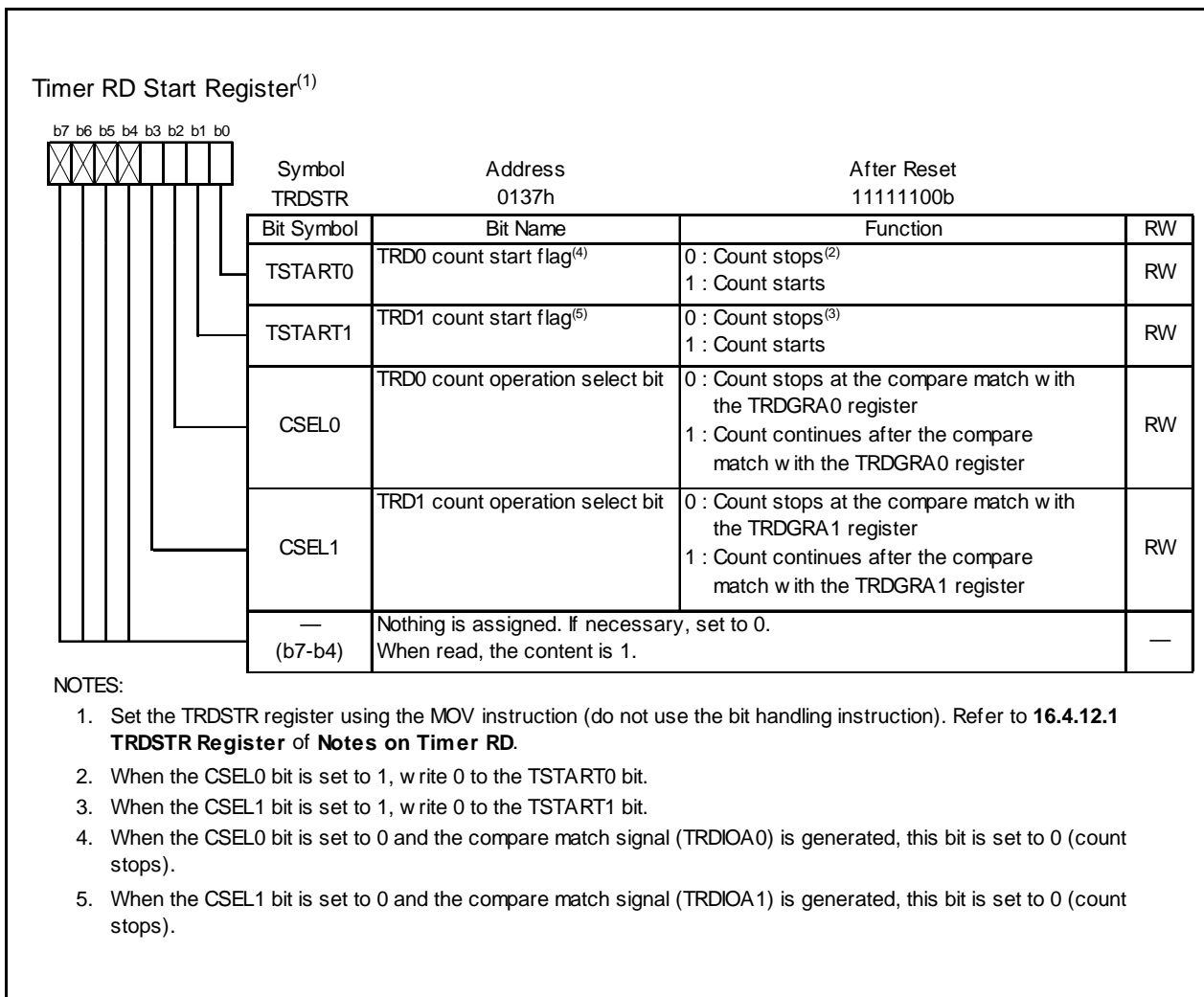


Figure 16.123 TRDSTR Register in Complementary PWM Mode

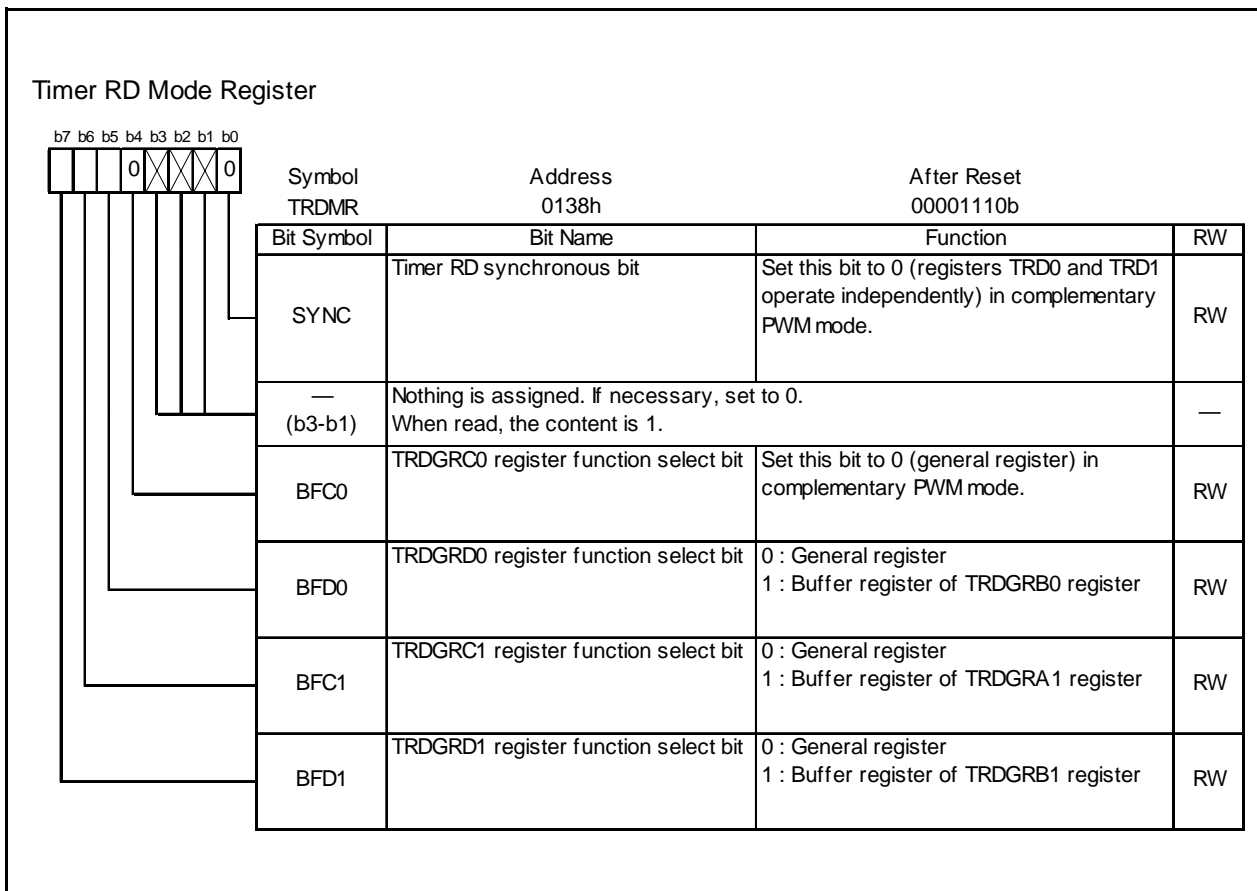


Figure 16.124 TRDMR Register in Complementary PWM Mode

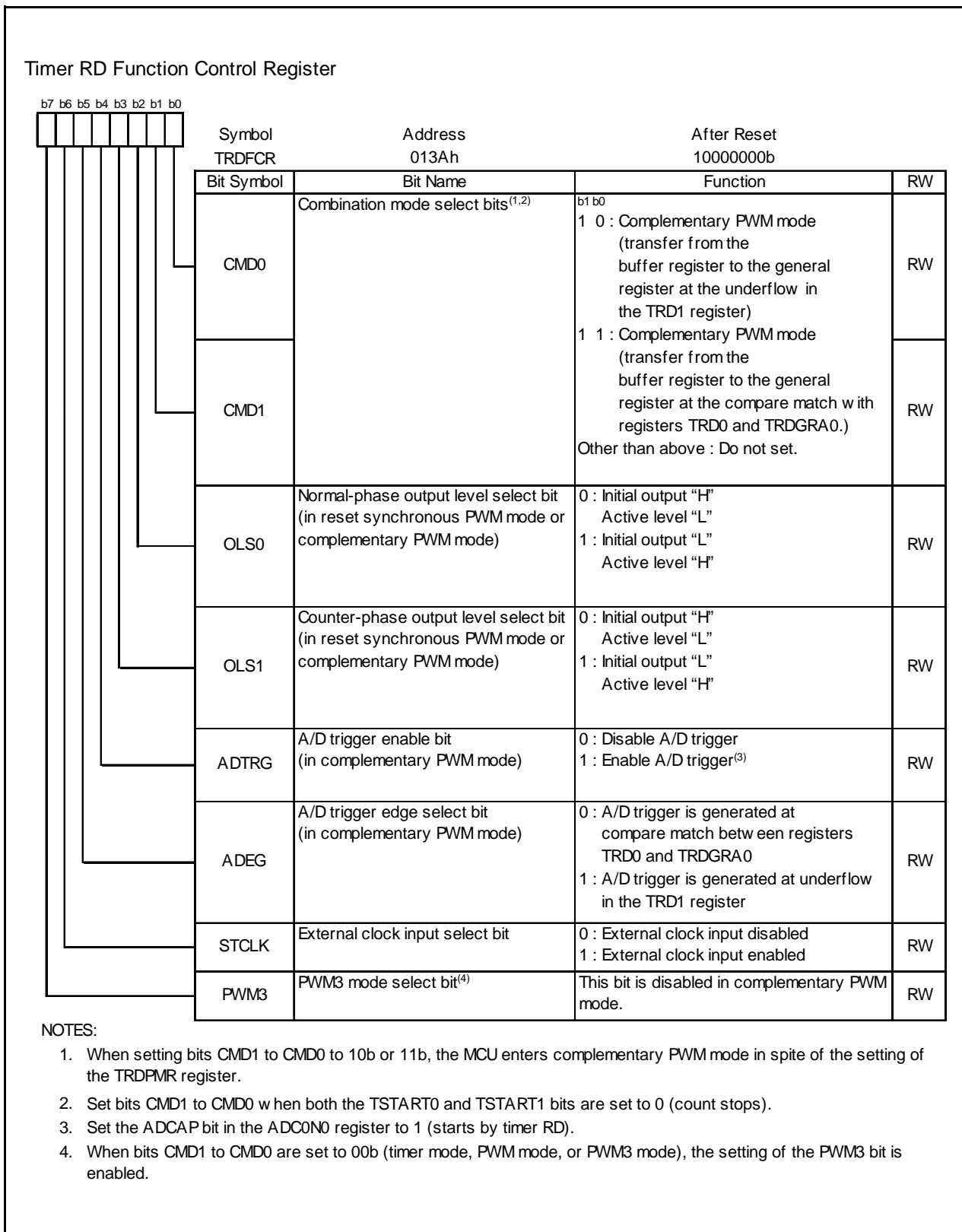


Figure 16.125 TRDFCR Register in Complementary PWM Mode



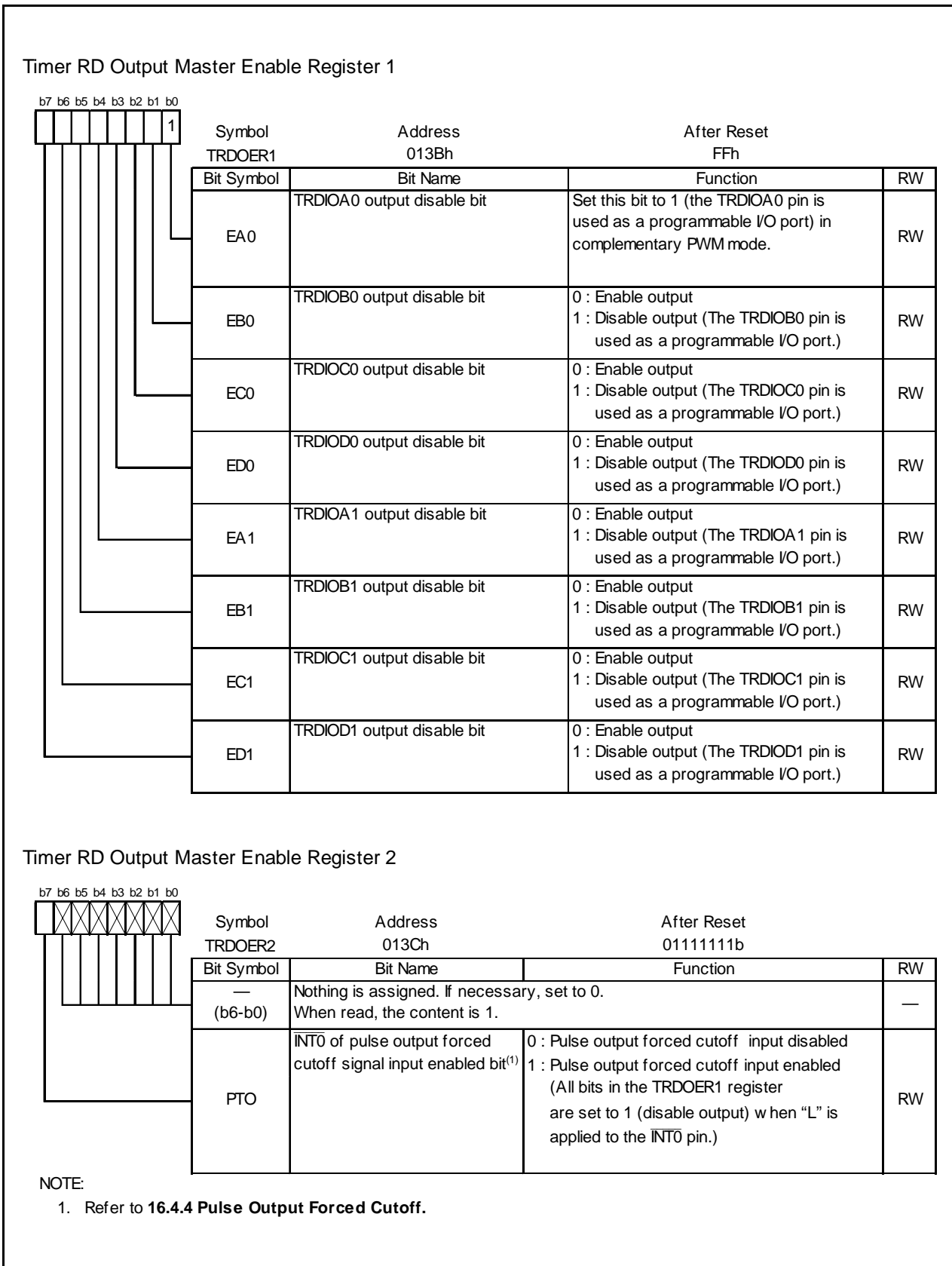


Figure 16.126 Registers TRDOER1 to TRDOER2 in Complementary PWM Mode

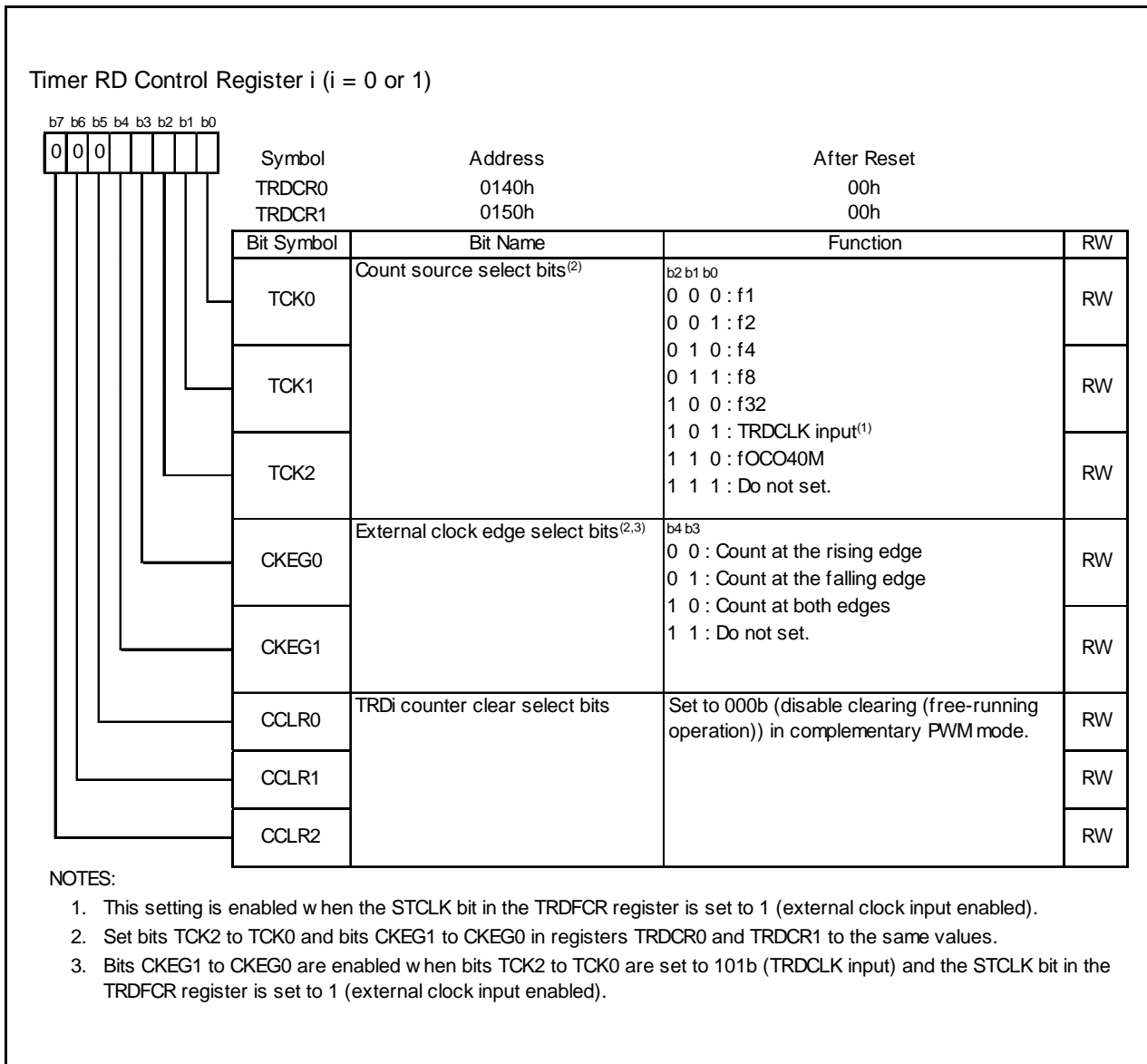


Figure 16.127 Registers TRDCR0 to TRDCR1 in Complementary PWM Mode

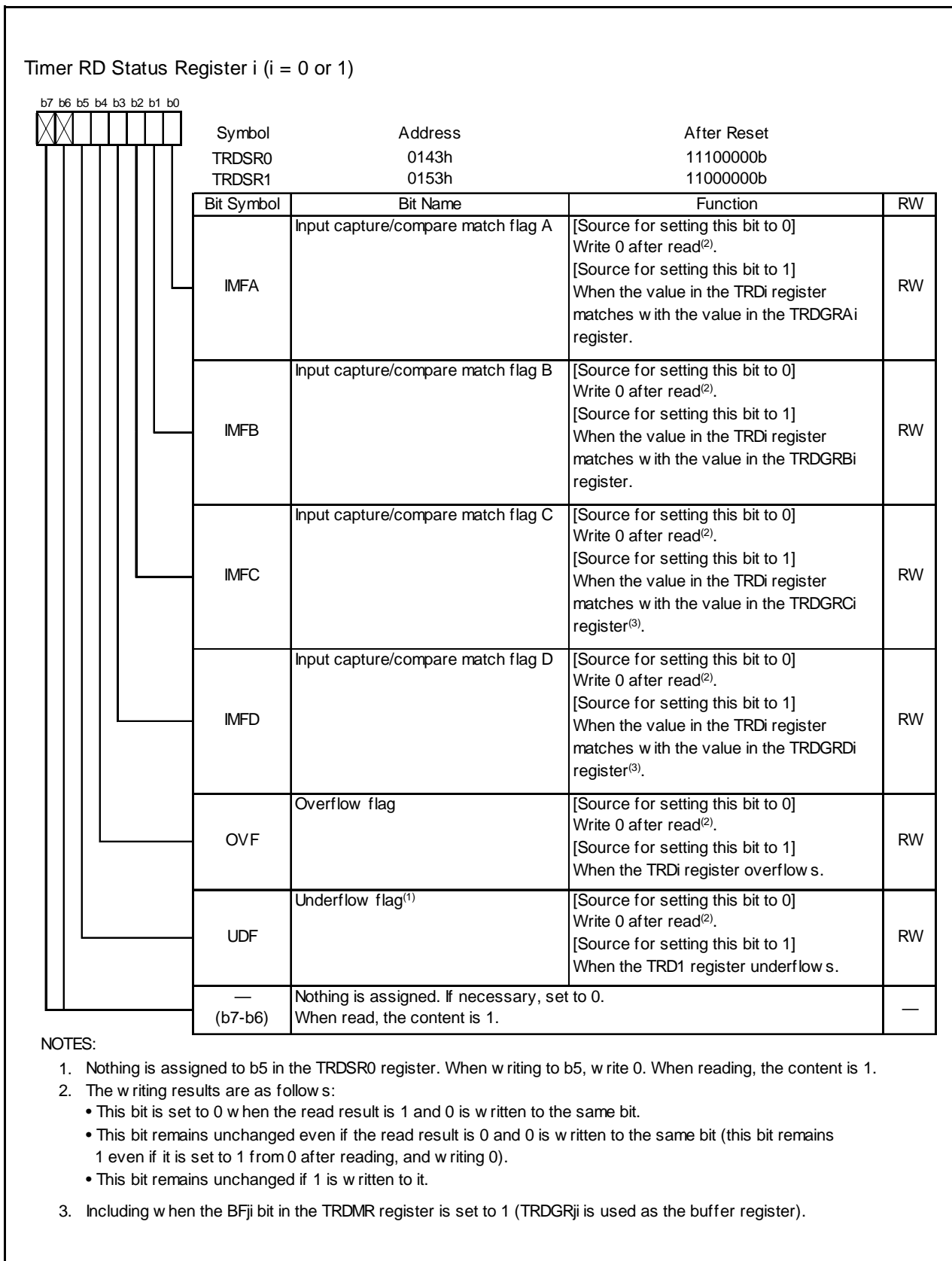


Figure 16.128 Registers TRDSR0 to TRDSR1 in Complementary PWM Mode

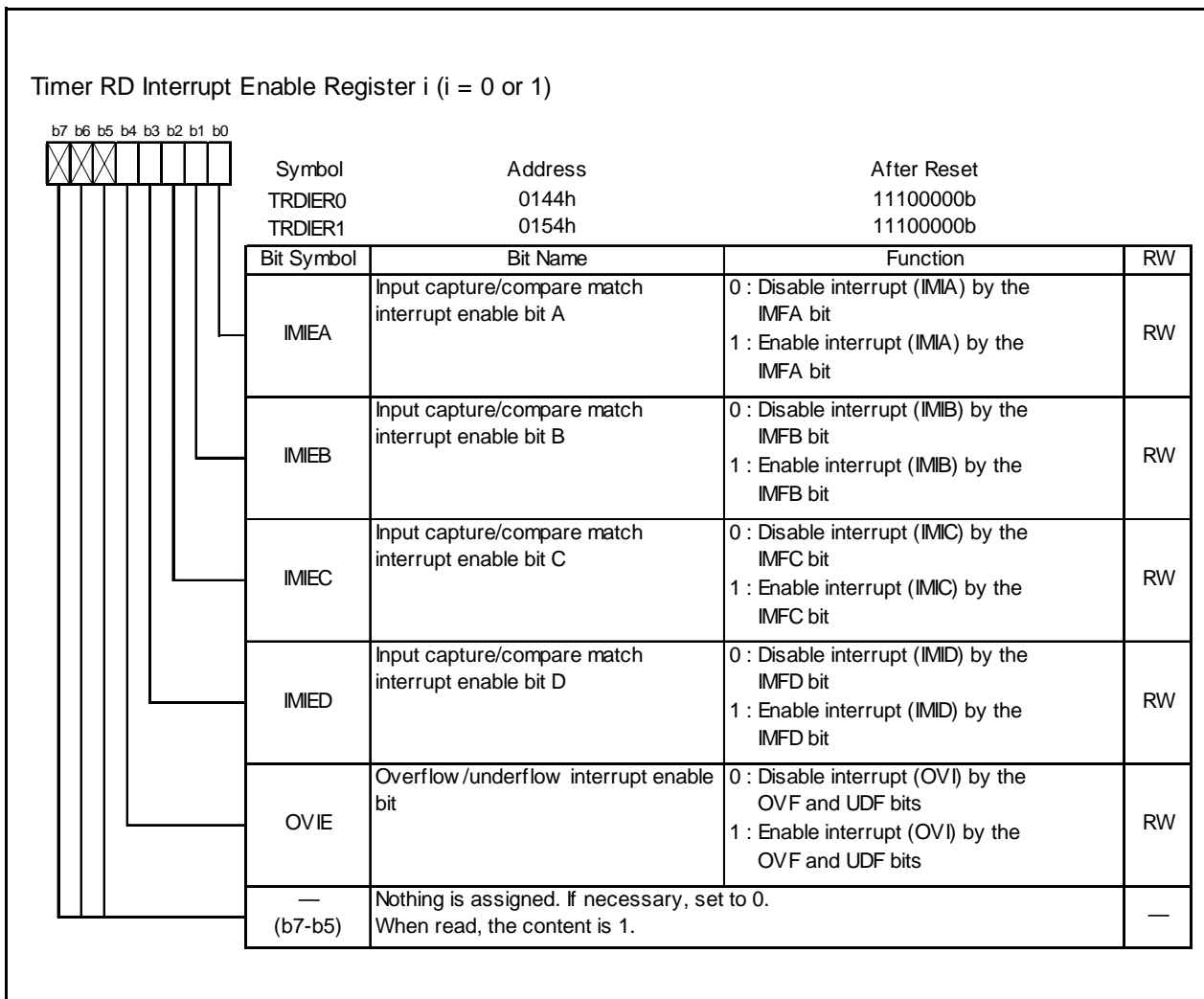


Figure 16.129 Registers TRDIER0 to TRDIER1 in Complementary PWM Mode

Timer RD Counter 0 <sup>(1)</sup>		Symbol	Address	After Reset	
(b15) b7	(b8) b0 b7	TRD0	0147h-0146h	0000h	
Function				Setting Range	RW
Set the dead time. Count a count source. Count operation is incremented or decremented. When an overflow occurs, the OVF bit in the TRDSR0 register is set to 1.				0000h to FFFFh	RW
NOTE:					
1. Access the TRD0 register in 16-bit units. Do not access it in 8-bit units.					
Timer RD Counter 1 <sup>(1)</sup>		Symbol	Address	After Reset	
(b15) b7	(b8) b0 b7	TRD1	0157h-0156h	0000h	
Function				Setting Range	RW
Select 0000h. Count a count source. Count operation is incremented or decremented. When an underflow occurs, the UDF bit in the TRDSR1 register is set to 1.				0000h to FFFFh	RW
NOTE:					
1. Access the TRD1 register in 16-bit units. Do not access it in 8-bit units.					

Figure 16.130 Registers TRD0 to TRD1 in Complementary PWM Mode

Timer RD General Registers Ai, Bi, C1, and Di (i = 0 or 1) <sup>(1, 2)</sup>		Symbol	Address	After Reset	
(b15) b7	(b8) b0 b7	TRDGRA0	0149h-0148h	FFFFh	
		TRDGRB0	014Bh-014Ah	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 16.46 TRDGRji Register Functions in Complementary PWM Mode.				RW	
NOTES:					
1. Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.					
2. The TRDGRC0 register is not used in complementary PWM mode.					

Figure 16.131 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 16.46 TRDGR<sub>j</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 to this register 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 to this register 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 to this register 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 to this register when the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts).	TRDIOB1 TRDIOD1
TRDGRC0	–	This register is not used in complementary PWM mode.	–
TRDGRD0	BFD0 = 1	Buffer register. Set the changing point of next PWM1 output. (Refer to <b>16.4.2 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 initialization.	TRDIOB0 TRDIOD0
TRDGRC1	BFC1 = 1	Buffer register. Set the changing point of next PWM2 output. (Refer to <b>16.4.2 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 initialization.	TRDIOA1 TRDIOC1
TRDGRD1	BFD1 = 1	Buffer register. Set the changing point of next PWM3 output. (Refer to <b>16.4.2 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 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).

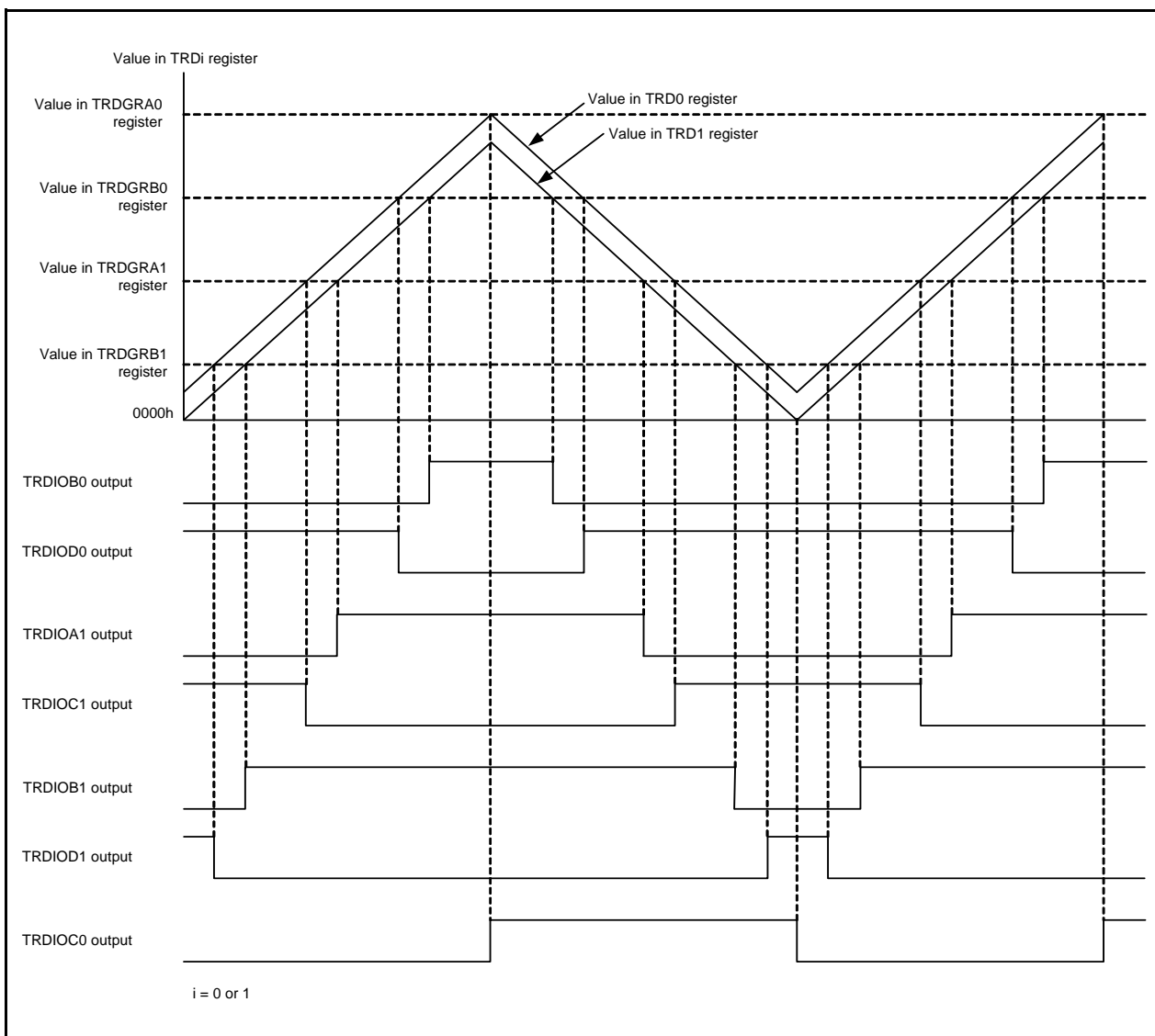


Figure 16.132 Output Model of Complementary PWM Mode

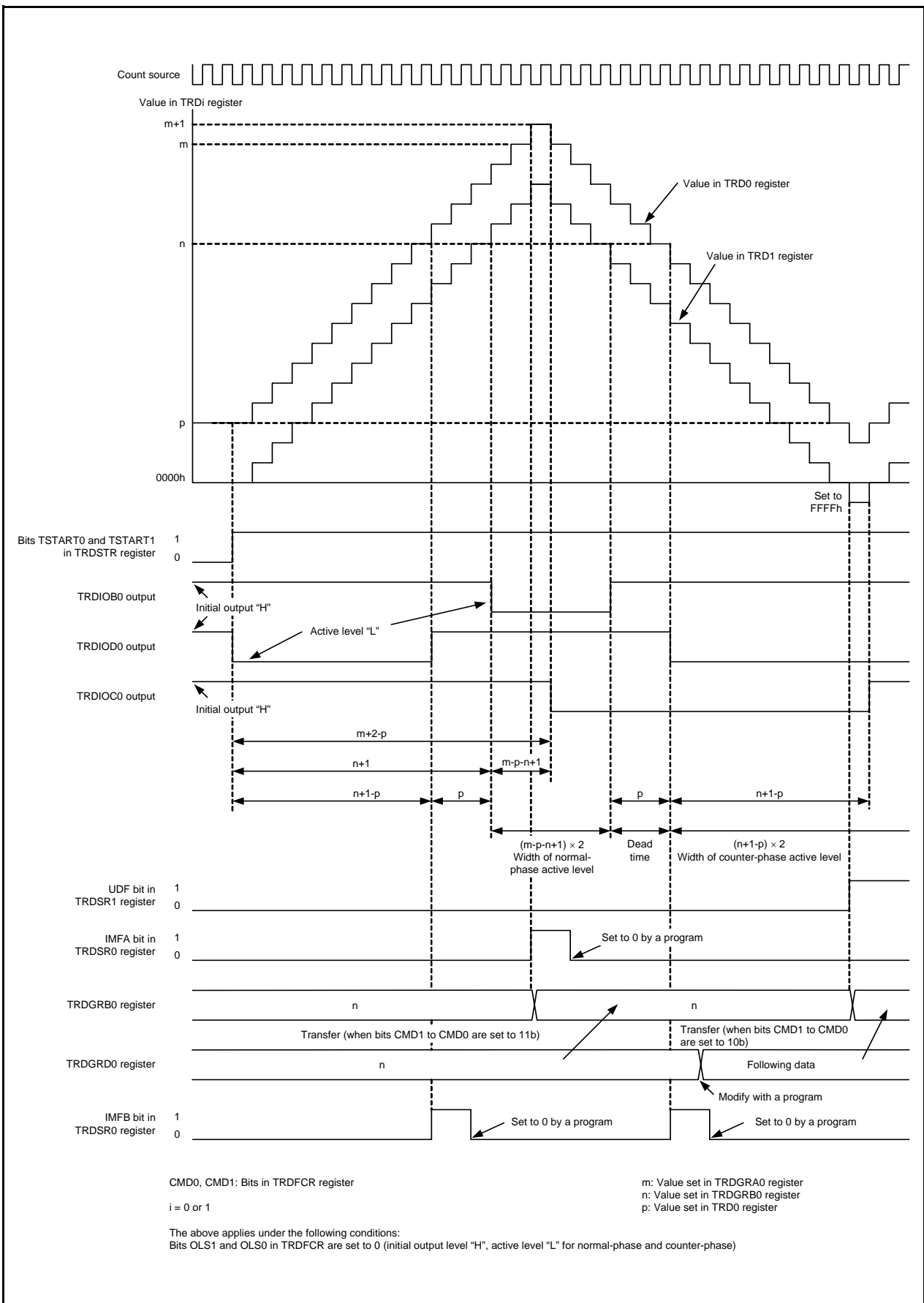


Figure 16.133 Operating Example of Complementary PWM Mode



### 16.4.9.1 Transfer Timing from Buffer Register

- Transfer from the TRDGRD0, TRDGRC1, or TRDGRD1 register to the TRDGRB0, TRDGRA1, or TRDGRB1 register.

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

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

### 16.4.9.2 A/D Trigger Generation

Compare match between registers TRD0 and TRDGRA0 and TRD1 underflow can be used as the conversion start trigger of the A/D converter. The trigger is selected by bits ADEG and ADTRG in the TRDFCR register. Also, set the ADCAP bit in the ADCON0 register to 1 (starts by timer RD).

### 16.4.10 PWM3 Mode

In this mode, 2 PWM waveforms are output with the same period.

Figure 16.134 shows a Block Diagram of PWM3 Mode, and Table 16.47 lists the PWM3 Mode Specifications. Figures 16.135 to 16.144 show the Registers Associated with PWM3 Mode, and Figure 16.145 shows an Operating Example of PWM3 Mode.

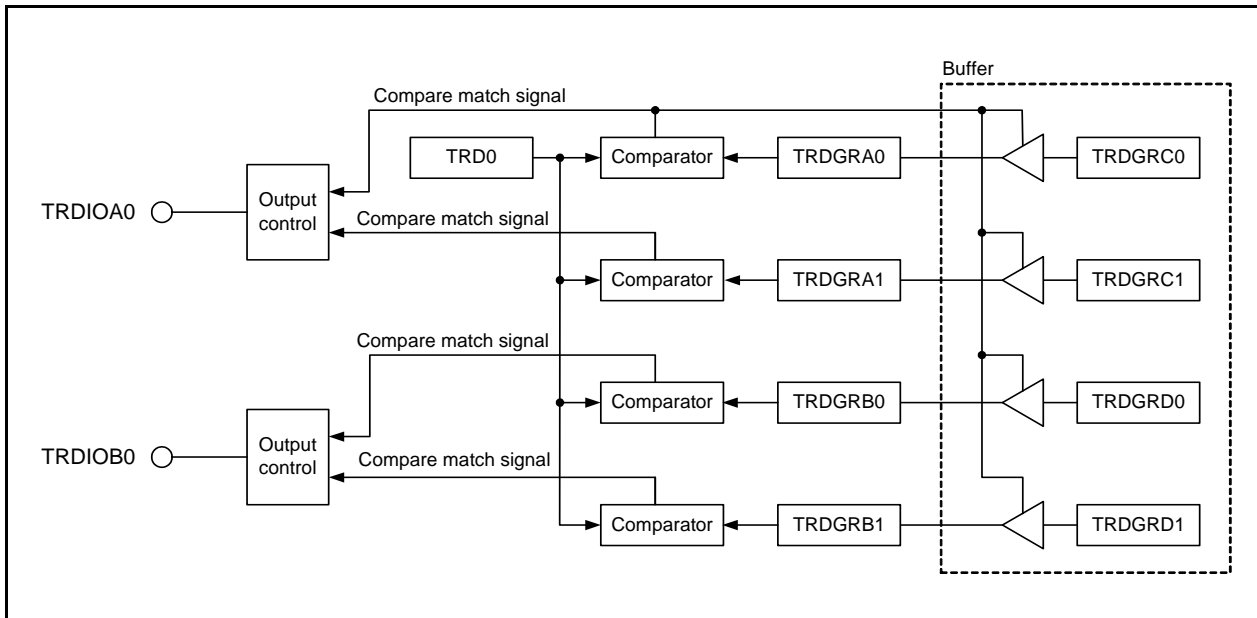
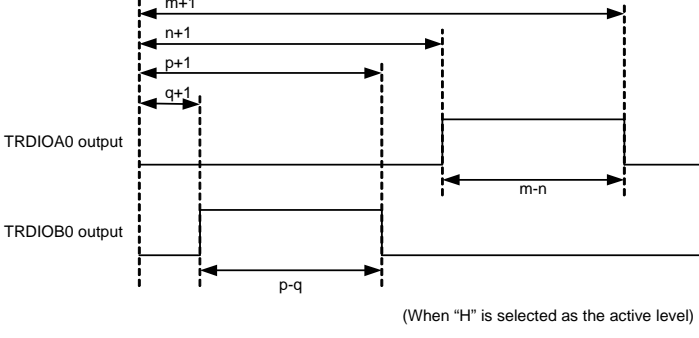


Figure 16.134 Block Diagram of PWM3 Mode

**Table 16.47 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>: Value set in the TRDGRA0 register  <math>n</math>: Value set in the TRDGRA1 register  <math>p</math>: Value set in the TRDGRB0 register  <math>q</math>: Value set in the TRDGRB1 register</p>  <p>(When "H" is selected as the active level)</p>
Count start condition	1 (count starts) is written to the TSTART0 bit in the TRDSTR register.
Count stop conditions	<ul style="list-style-type: none"> <li>• 0 (count stops) is written to the TSTART0 bit 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</li> <li>• When the CSEL0 bit in the TRDSTR register is set to 0, the count stops at compare match with the TRDGRA0 register. The PWM output pin holds level after output change by compare match.</li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• Compare match (The content of the TRDi register matches content of 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.
Select functions	<ul style="list-style-type: none"> <li>• Pulse output forced cutoff signal input (Refer to <b>16.4.4 Pulse Output Forced Cutoff.</b>)</li> <li>• Buffer Operation (Refer to <b>16.4.2 Buffer Operation.</b>)</li> <li>• Active level selectable by pin</li> </ul>

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

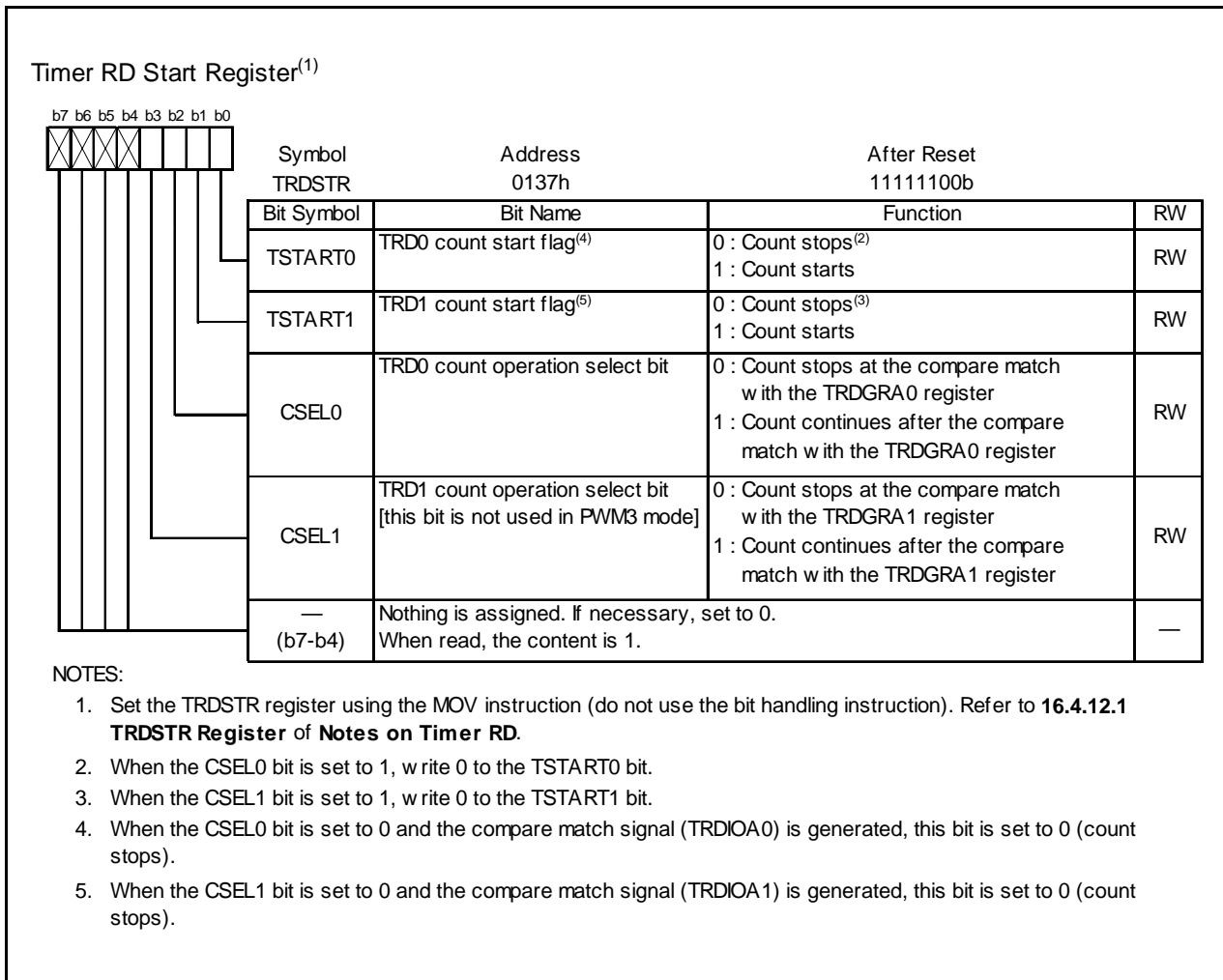


Figure 16.135 TRDSTR Register in PWM3 Mode

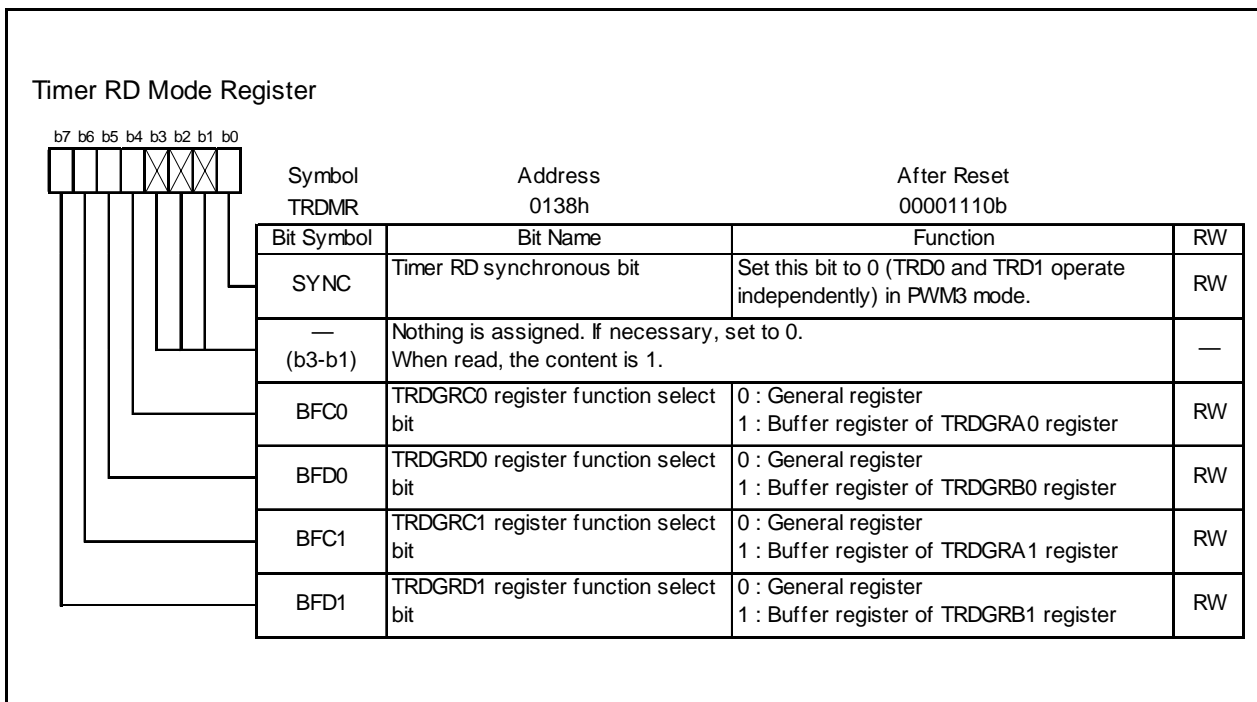


Figure 16.136 TRDMR Register in PWM3 Mode

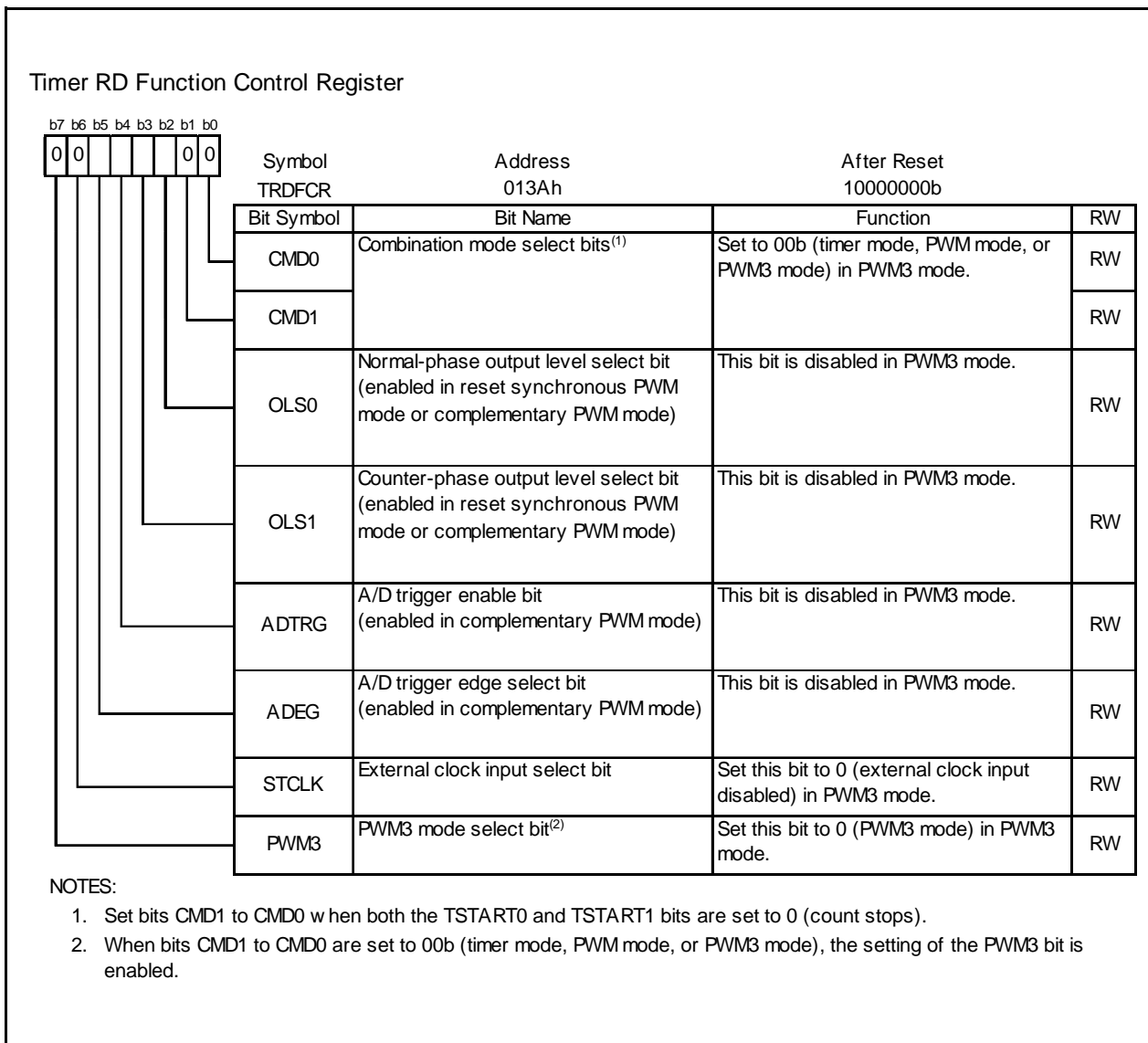


Figure 16.137 TRDFCR Register in PWM3 Mode

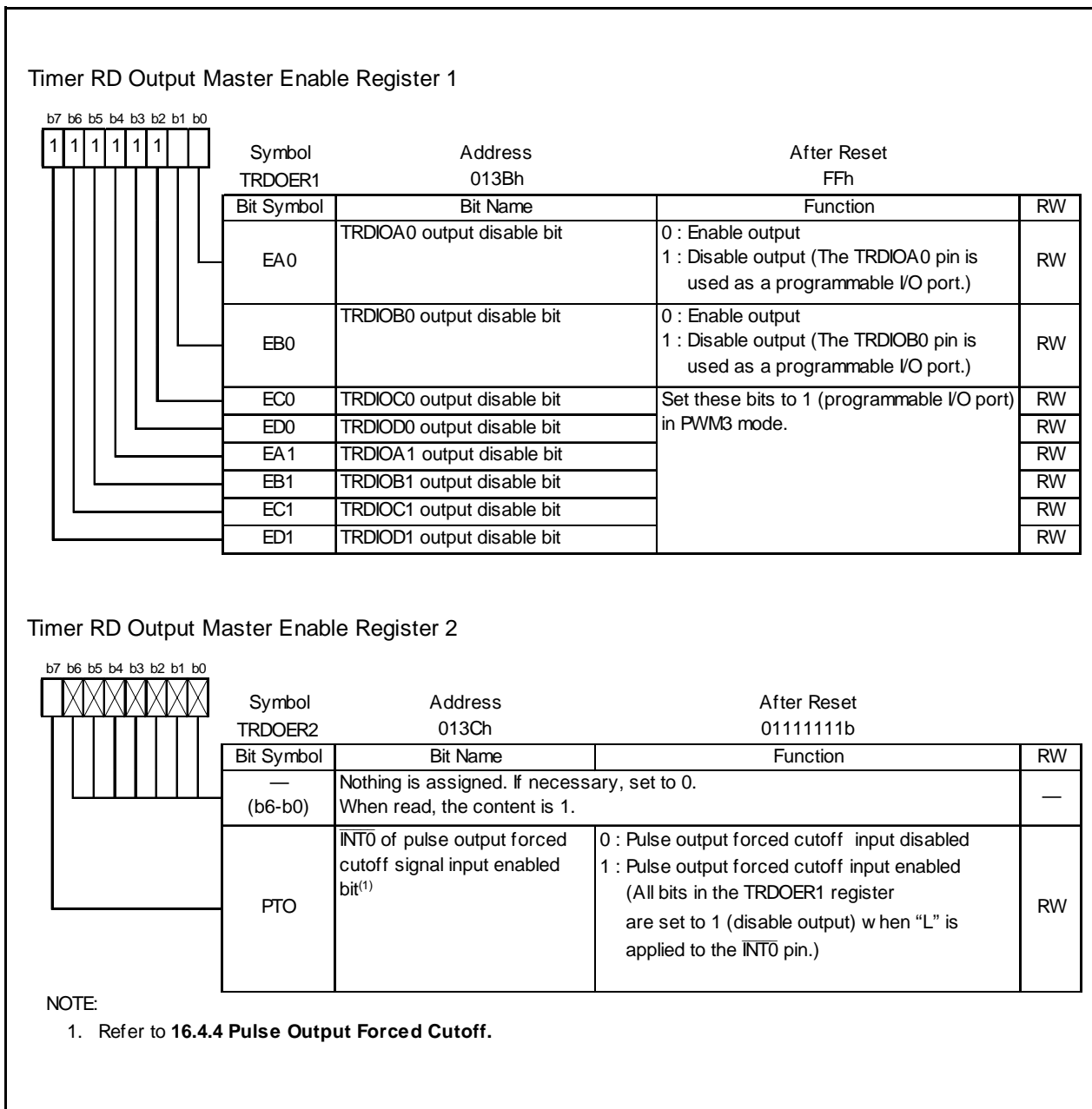


Figure 16.138 Registers TRDOER1 to TRDOER2 in PWM3 Mode

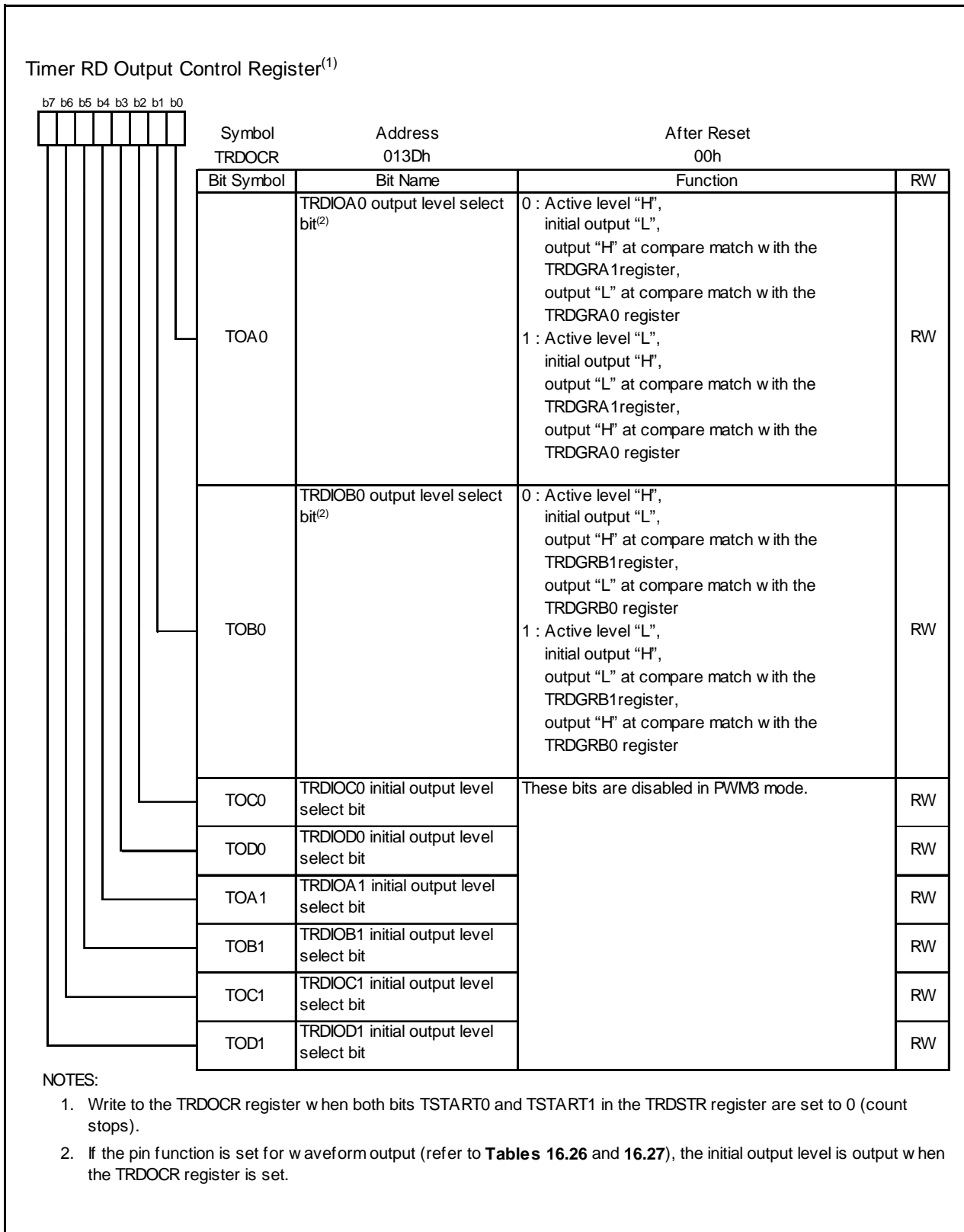


Figure 16.139 TRDOCR Register in PWM3 Mode



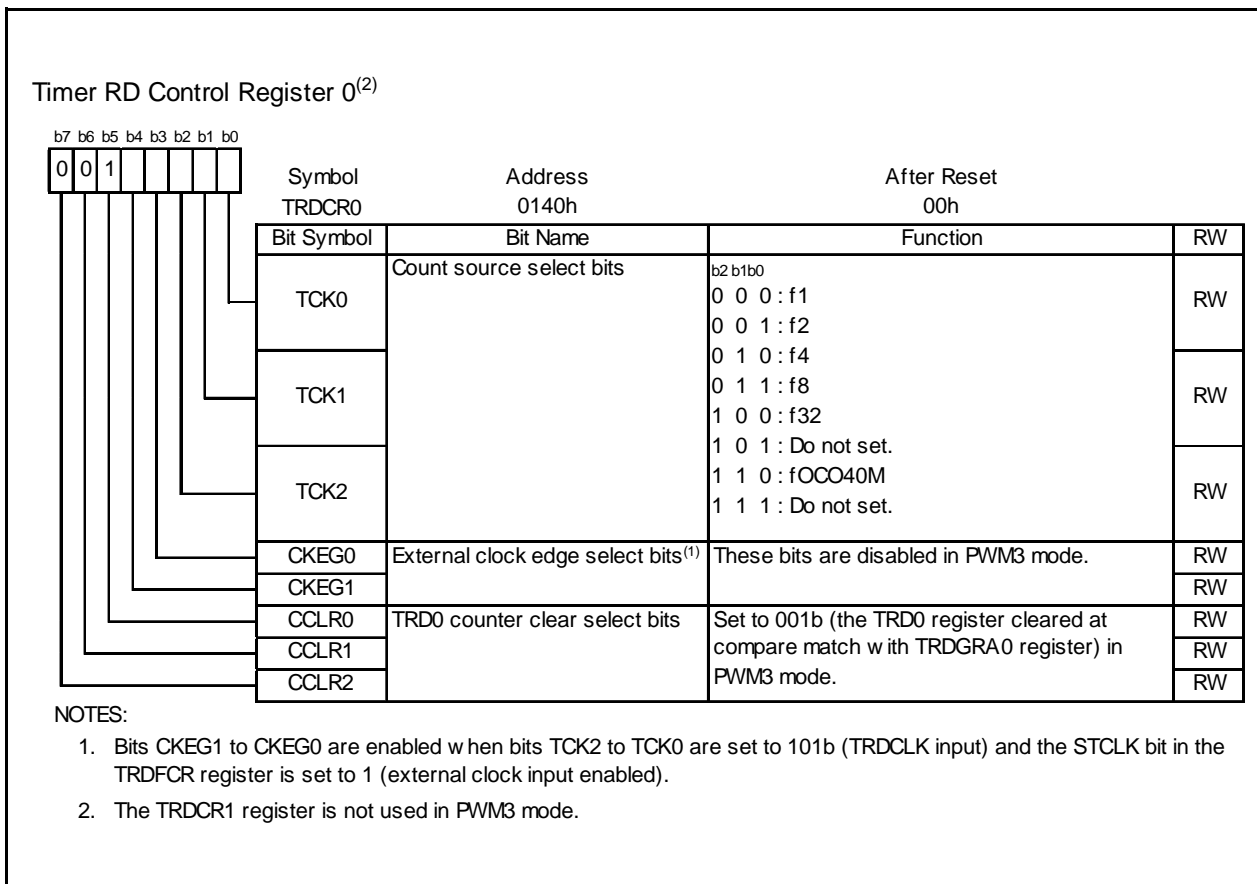


Figure 16.140 TRDCR0 Register in PWM3 Mode

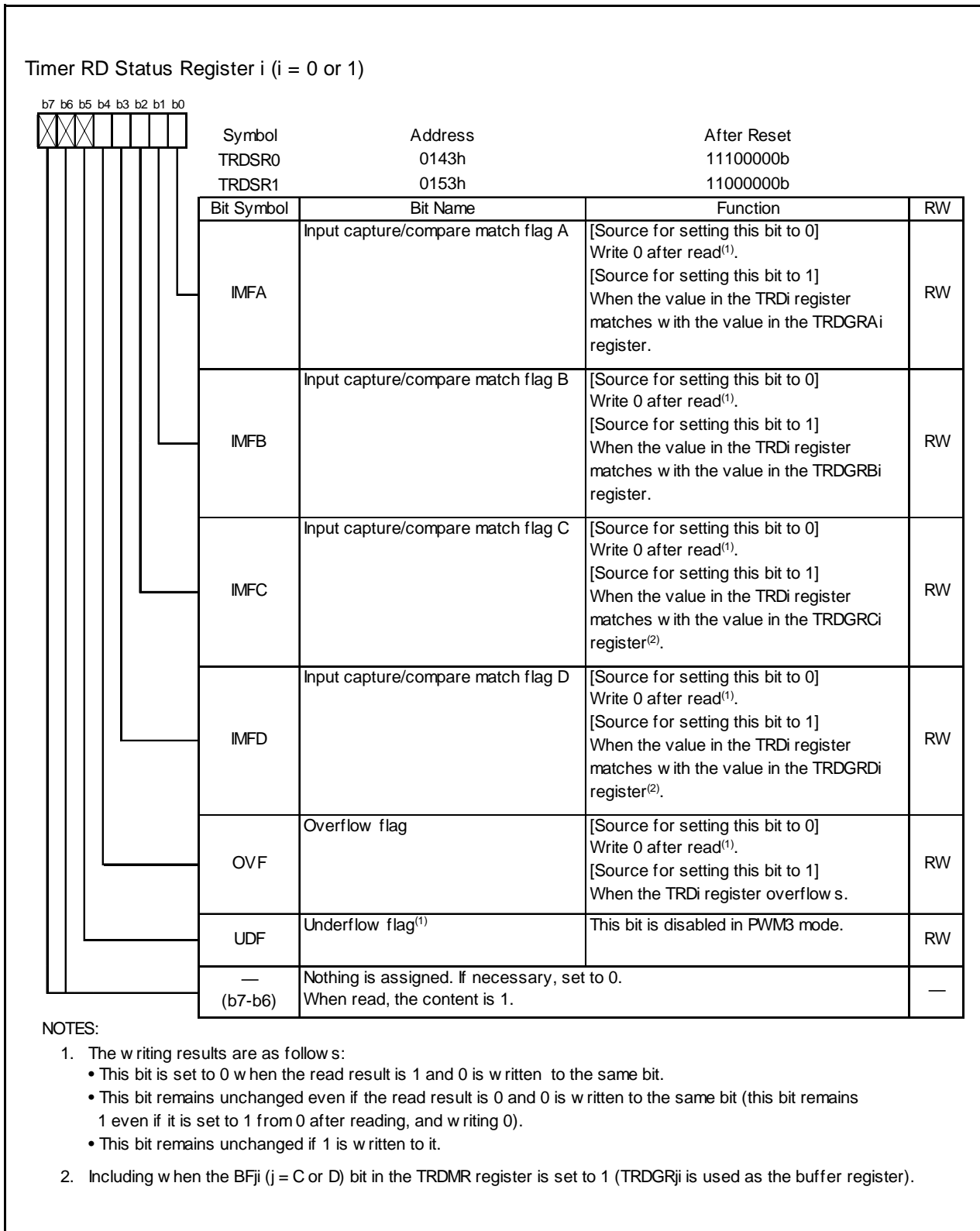


Figure 16.141 Registers TRDSR0 to TRDSR1 in PWM3 Mode

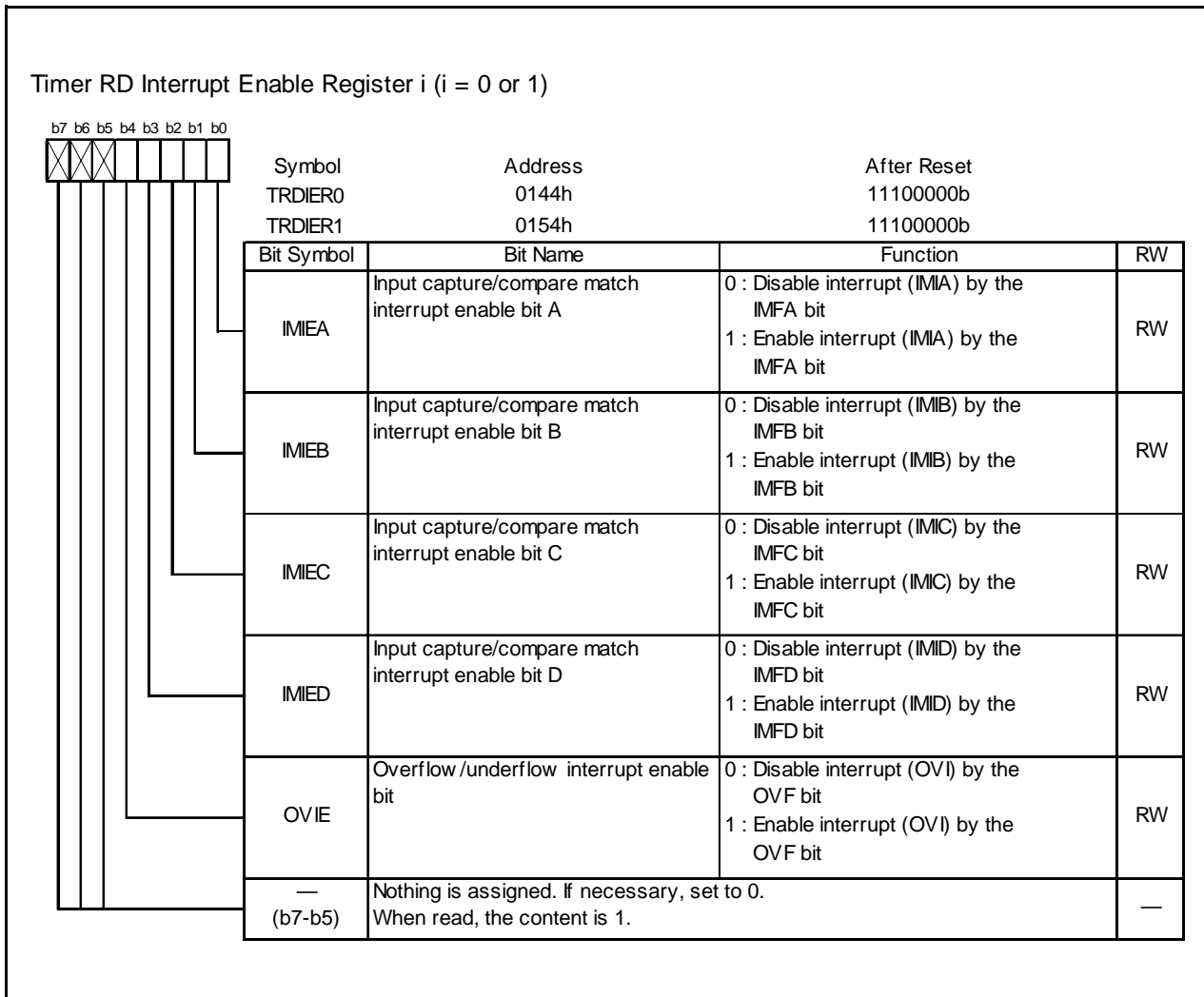


Figure 16.142 Registers TRDIER0 to TRDIER1 in PWM3 Mode

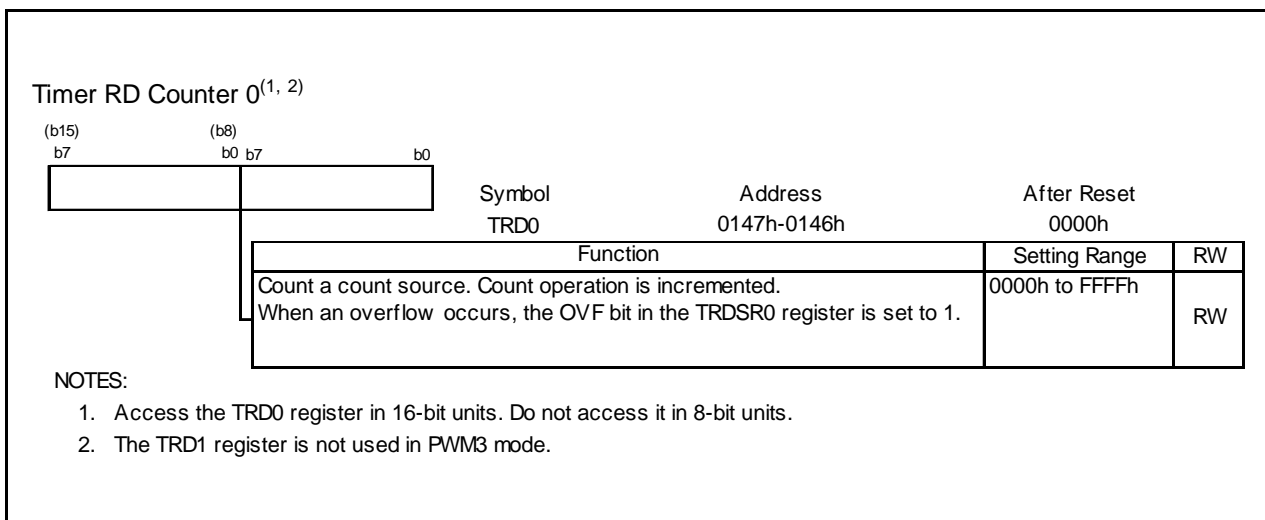


Figure 16.143 TRD0 Register in PWM3 Mode

Timer RD General Registers 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 <b>Table 16.48 TRDGRji Register Functions in PWM3 Mode.</b>						RW

NOTE:

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

**Figure 16.144 Registers TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi in PWM3 Mode**

The following registers are disabled in the PWM3 mode function: TRDPMR, TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDPOCR0, TRDIORA1, TRDIORC1, and TRDPOCR1.

**Table 16.48 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 is 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>16.4.2 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>16.4.2 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>16.4.2 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>16.4.2 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).

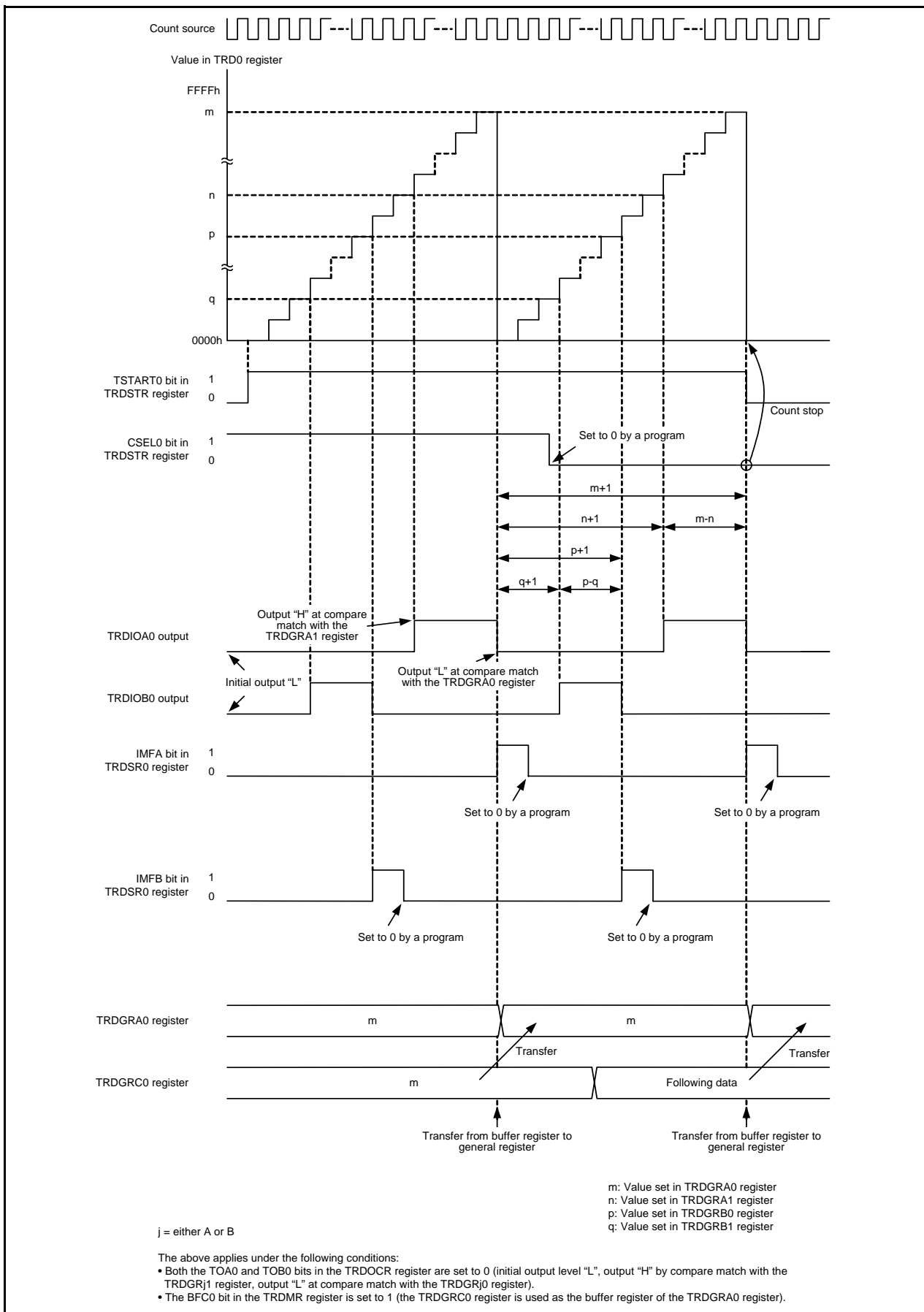


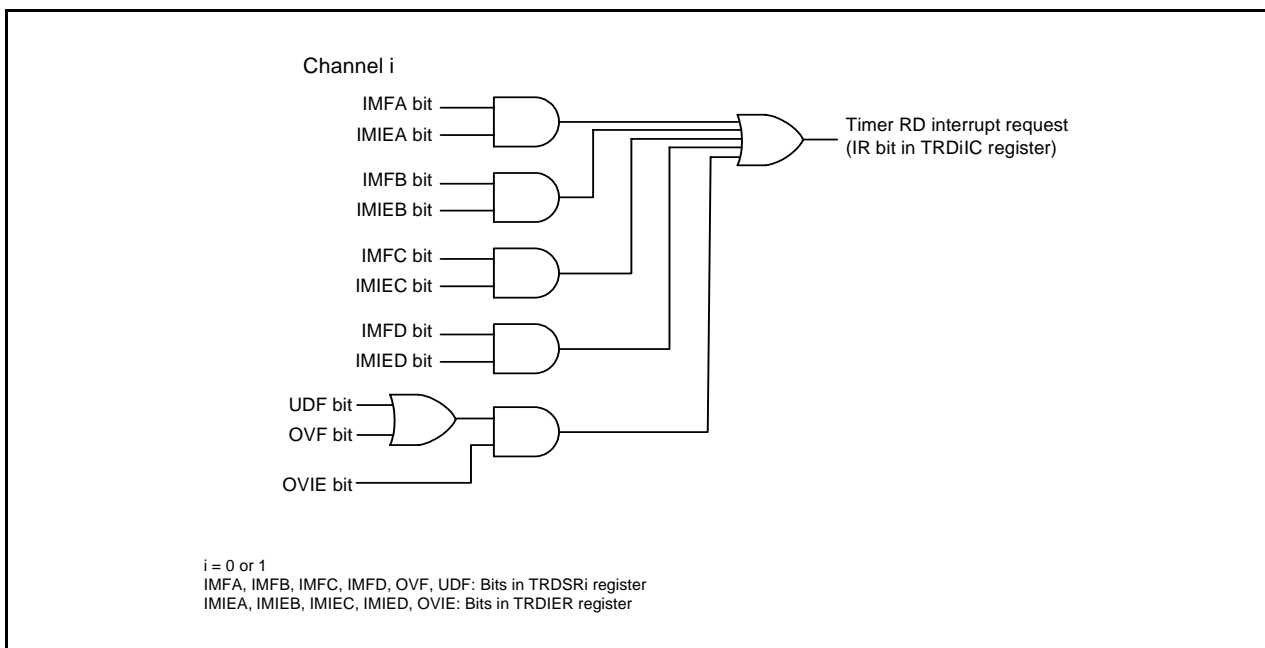
Figure 16.145 Operating Example of PWM3 Mode

### 16.4.11 Timer RD Interrupt

Timer RD generates the timer RD interrupt request based on 6 sources for each channel. The timer RD interrupt has 1 TRDiIC register (bits IR, and ILVL0 to ILVL2), and 1 vector for each channel. Table 16.49 lists the Registers Associated with Timer RD Interrupt, and Figure 16.146 shows a Block Diagram of Timer RD Interrupt.

**Table 16.49 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 16.146 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 TRDSRi register corresponding to bits set to 1 in the TRDIERi 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 TRDSRi register or bits in the TRDIERi register corresponding to bits in the TRDSRi 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 TRDIERi register are set to 1, which request source causes an interrupt is determined by the TRDSRi register.
- Since each bit in the TRDSRi 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 16.77, 16.92, 16.105, 16.117, 16.128, and 16.141).

Refer to **Registers TRDSR0 to TRDSR1 in each mode (Figures 16.77, 16.92, 16.105, 16.117, 16.128, and 16.141)** for the TRDSRi register. Refer to **Registers TRDIER0 to TRDIER1 in each mode (Figures 16.78, 16.93, 16.106, 16.118, 16.129, and 16.142)** for the TRDIERi register.

Refer to **12.1.6 Interrupt Control** for information on the TRDiIC register and **12.1.5.2 Relocatable Vector Tables** for the interrupt vectors.



## 16.4.12 Notes on Timer RD

### 16.4.12.1 TRDSTR Register

- Set the TRDSTR register using the MOV instruction.
- When the CSELi (i = 0 to 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 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 16.50 lists the TRDIOji (j = A, B, C, or D) Pin Output Level when Count Stops to use the TRDIOji (j = A, B, C, or D) pin with the timer RD output.

**Table 16.50 TRDIOji (j = A, B, C, or D) Pin Output Level when Count Stops**

Count Stop	TRDIOji 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.

### 16.4.12.2 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 overlapping with the timing for setting the TRDi register to 0000h, and then write. If the timing for setting the TRDi register to 0000h overlaps with the timing for writing 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 bits CCLR2 to CCLR0 in the TRDCRi register.
  - 001b (Clear by the TRDi register at compare match with the TRDGRAi register.)
  - 010b (Clear by the TRDi register at compare match with the TRDGRBi register.)
  - 011b (Synchronous clear)
  - 101b (Clear by the TRDi register at compare match with the TRDGRCi register.)
  - 110b (Clear by the TRDi register at 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
  
```

### 16.4.12.3 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
  
```

#### 16.4.12.4 Count Source Switch

- Switch the count source 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 bits TCK2 to TCK0 in the TRDCR<sub>i</sub> register.

- When changing the count source from fOCO40M to another source and stopping fOCO40M, wait 2 cycles of f<sub>1</sub> or more 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 bits TCK2 to TCK0 in the TRDCR<sub>i</sub> register.
- (3) Wait 2 or more cycles of f<sub>1</sub>.
- (4) Set the FRA00 bit in the FRA0 register to 0 (high-speed on-chip oscillator stops).

#### 16.4.12.5 Input Capture Function

- Set the pulse width of the input capture signal to 3 or more cycles of the timer RD operation clock (refer to **Table 16.25 Timer RD Operation Clocks**).
- The value in the TRD<sub>i</sub> register is transferred to the TRDGR<sub>ji</sub> register 2 to 3 cycles of the timer RD operation clock after 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).

#### 16.4.12.6 Reset Synchronous PWM Mode

- When reset synchronous PWM mode is used for motor control, make sure OLS0 = OLS1.
- Set to reset synchronous PWM mode by the following procedure:

Change procedure

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

#### 16.4.12.7 Complementary PWM Mode

- When complementary PWM mode is used for motor control, make sure OLS0 = OLS1.
- Change bits CMD1 to CMD0 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 bits CMD1 to CMD0 in the TRDFCR register to 00b (timer mode, PWM mode, and PWM3 mode).
- (3) Set bits CMD1 to CMD0 to 10b or 11b (complementary PWM mode).
- (4) Set the registers associated with other timer RD again.

Change procedure: When stopping complementary PWM mode

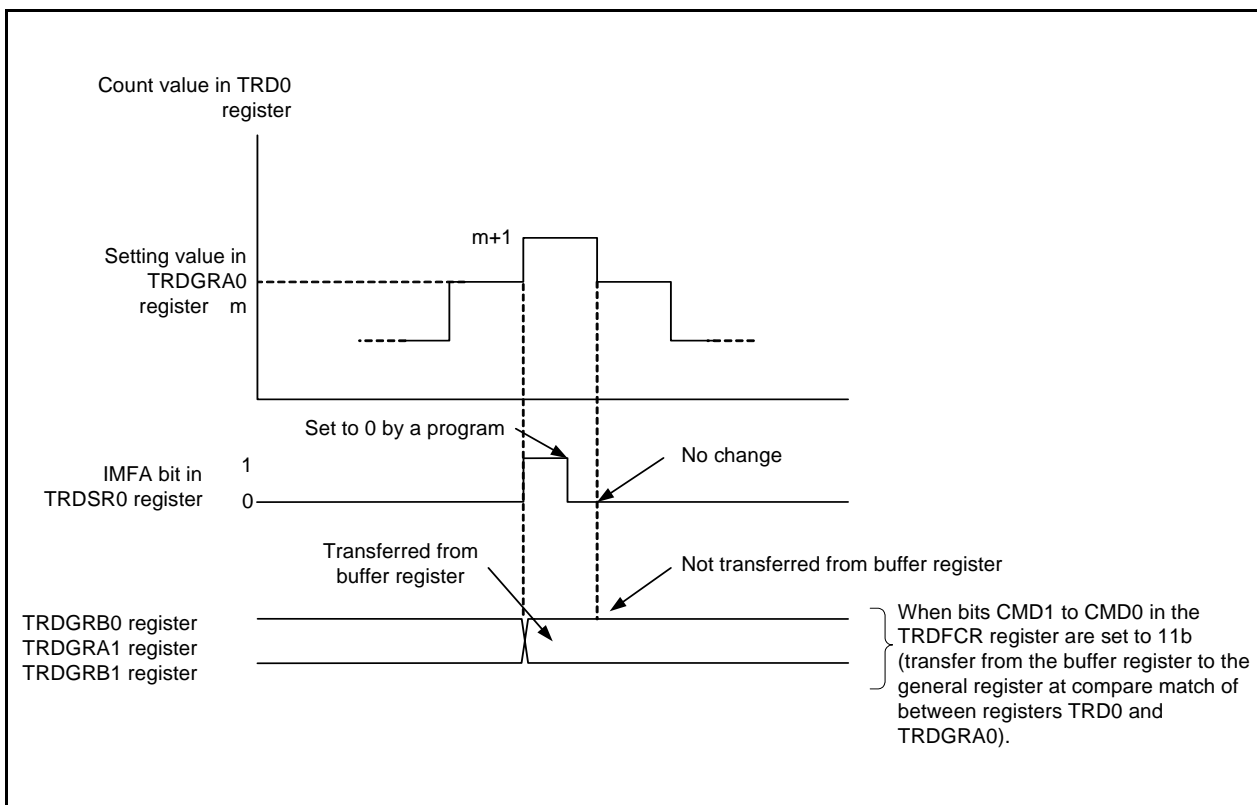
- (1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
- (2) Set bits CMD1 to CMD0 to 00b (timer mode, PWM mode, and PWM3 mode).

- Do not write to TRDGRA0, TRDGRB0, TRDGRA1, or TRDGRB1 register during operation. When changing the PWM waveform, transfer the values written to registers TRDGRD0, TRDGRC1, and TRDGRD1 to registers TRDGRB0, TRDGRA1, and TRDGRB1 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.

- If the value in the TRDGRA0 register is assumed to be  $m$ , the TRD0 register counts  $m-1$ ,  $m$ ,  $m+1$ ,  $m$ ,  $m-1$ , in that order, when changing from increment to decrement operation.

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

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



**Figure 16.147 Operation at Compare Match between Registers TRD0 and TRDGRA0 in Complementary PWM Mode**

- The TRD1 register counts 1, 0, FFFFh, 0, 1, in that order, when changing from decrement to increment operation.

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

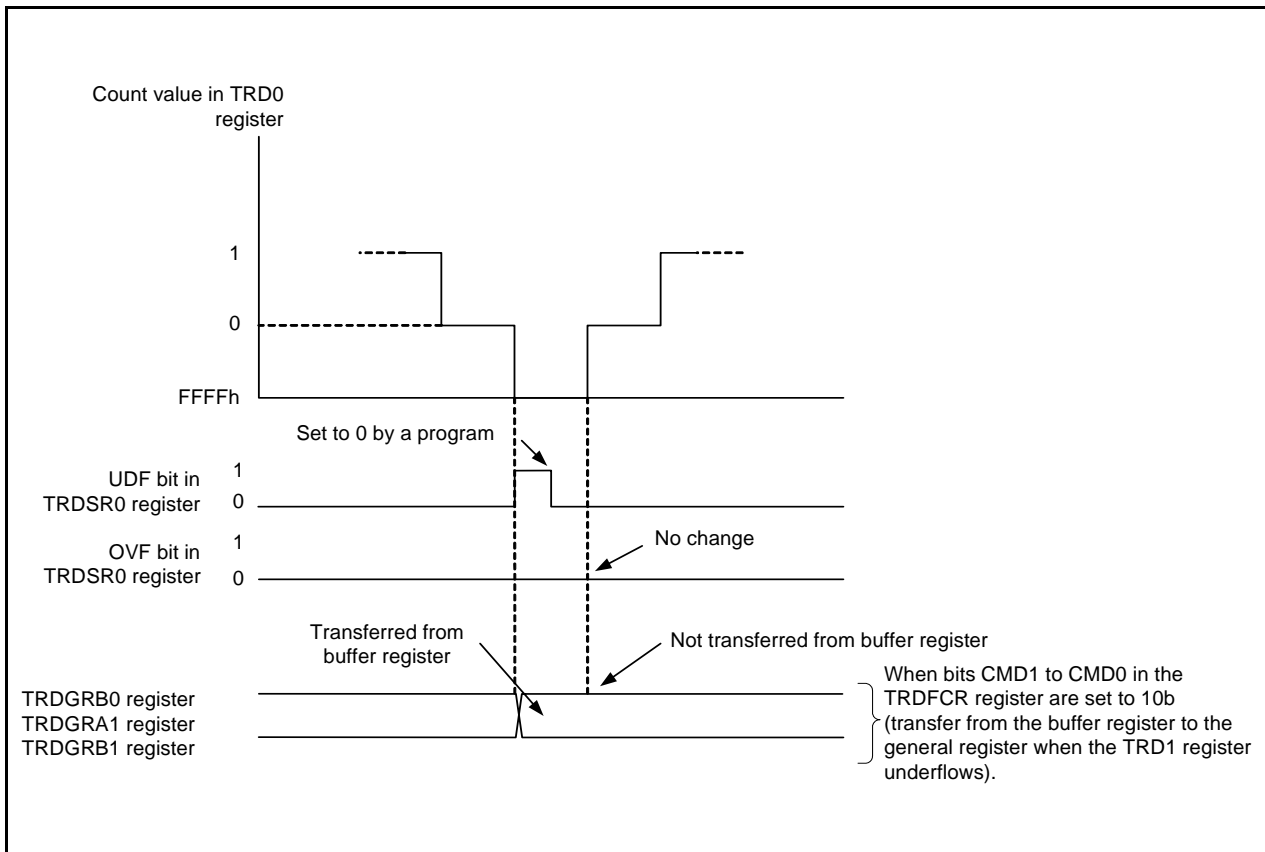


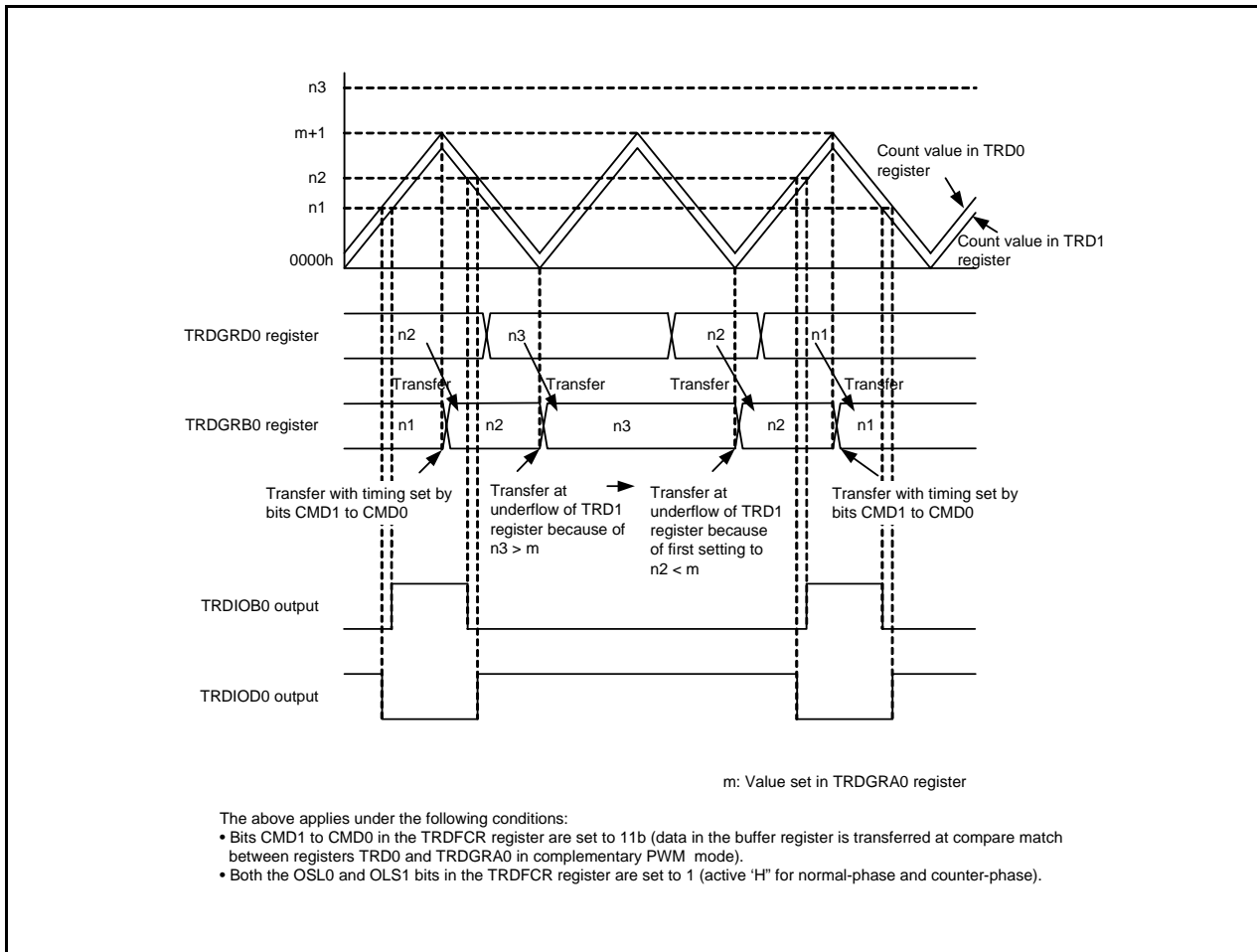
Figure 16.148 Operation when TRD1 Register Underflows in Complementary PWM Mode

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

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

Transfer take place at underflow of the TRD1 register.

After this, when the buffer register is set to 0001h or above and a smaller value than the value of the TRDGRA0 register, and the TRD1 register underflows for the first time after setting, the value is transferred to the general register. After that, the value is transferred with the timing selected by bits CMD1 to CMD0.

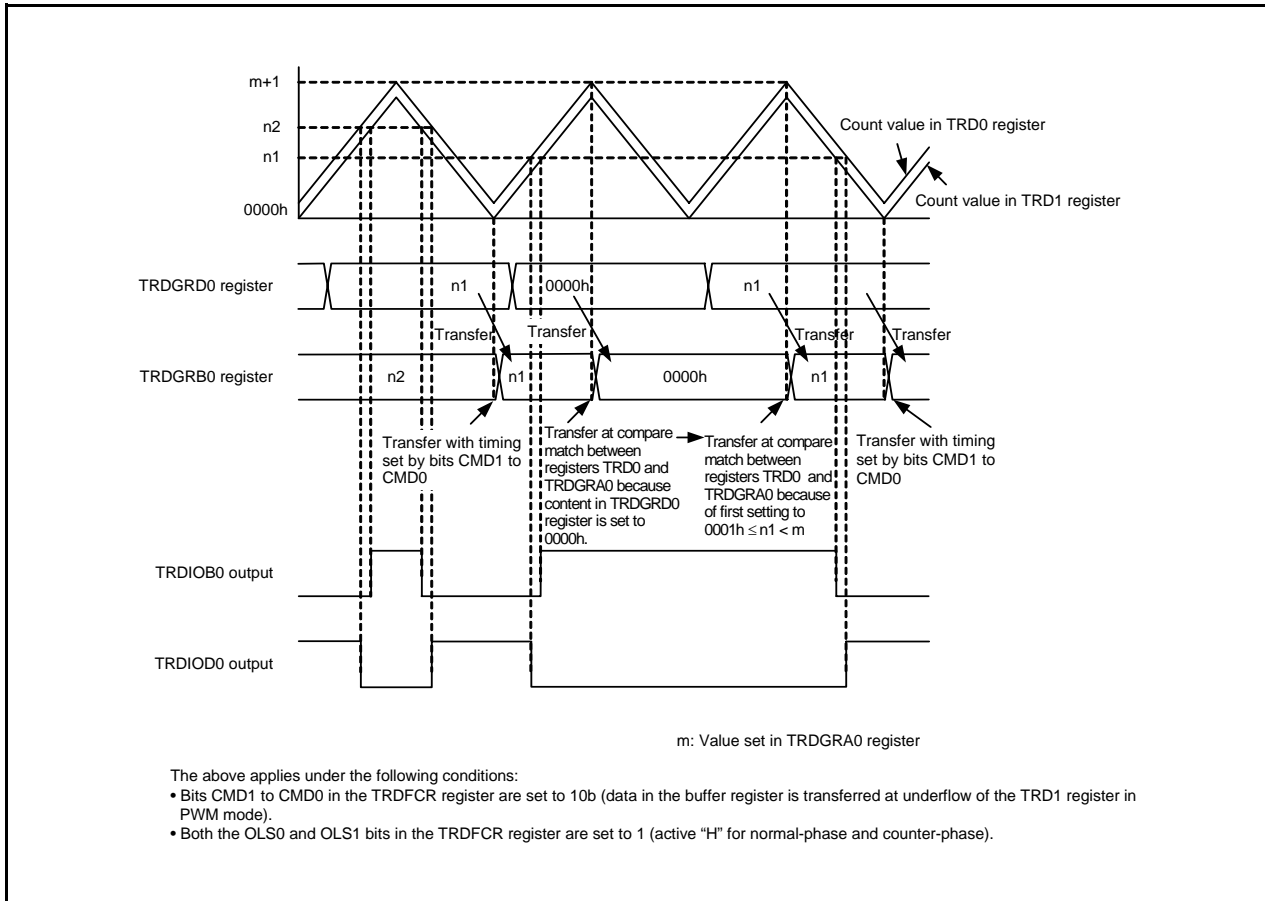


**Figure 16.149 Operation when Value in Buffer Register  $\geq$  Value in TRDGRA0 Register in Complementary PWM Mode**

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

Transfer takes place at compare match between registers TRD0 and TRDGRA0.

After this, when the buffer register is set to 0001h or above and a smaller value than the value of the TRDGRA0 register, and a compare match occurs between registers TRD0 and TRDGRA0 for the first time after setting, the value is transferred to the general register. After that, the value is transferred with the timing selected by bits CMD1 to CMD0.



**Figure 16.150 Operation when Value in Buffer Register Is Set to 0000h in Complementary PWM Mode**

#### 16.4.12.8 Count Source fOCO40M

- The count source fOCO40M can be used with supply voltage  $V_{CC} = 3.0$  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).

## 17. Serial Interface

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

Figure 17.1 shows a UART<sub>i</sub> ( $i = 0$  or  $2$ ) Block Diagram. Figure 17.2 shows a UART<sub>i</sub> Transmit/Receive Unit.

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

Figures 17.3 to 17.5 show the Registers Associated with UART<sub>i</sub>, and Figure 17.6 shows the PINSR1 Register.

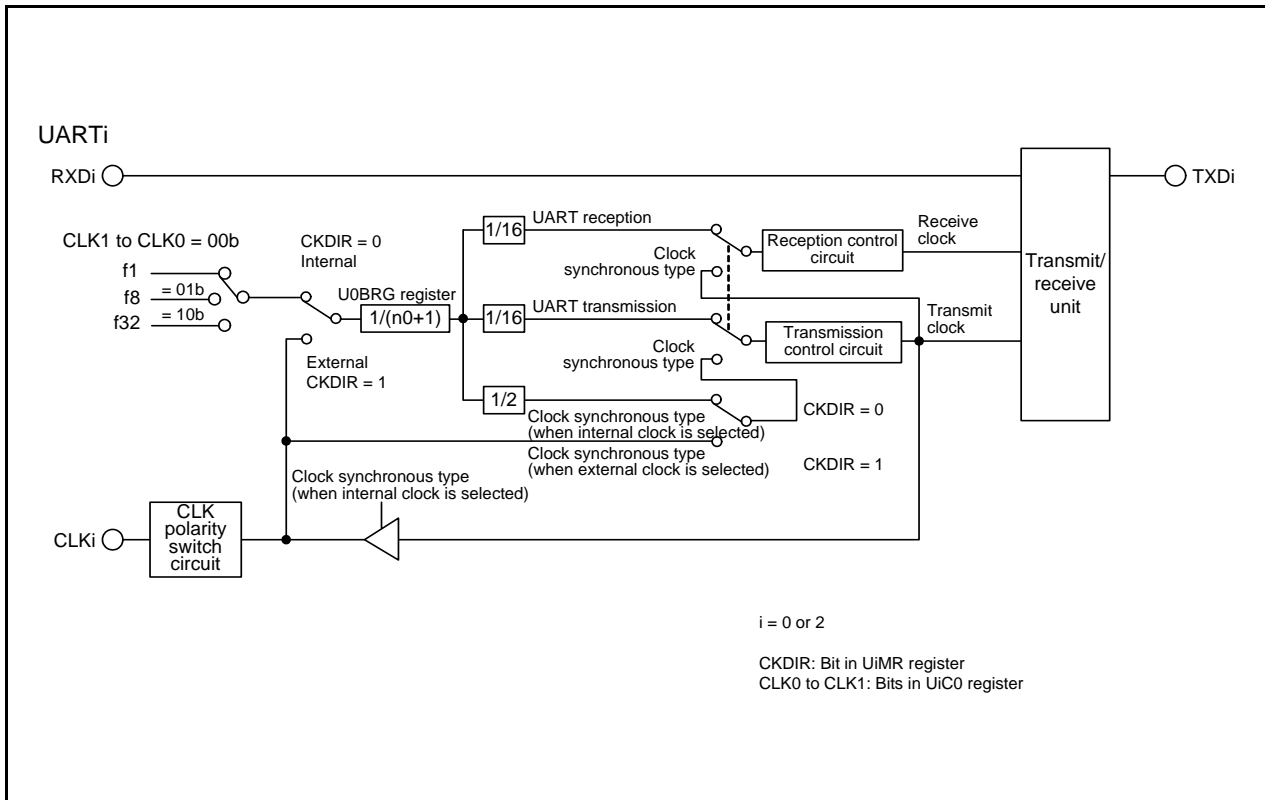


Figure 17.1 UART<sub>i</sub> ( $i = 0$  or  $2$ ) Block Diagram

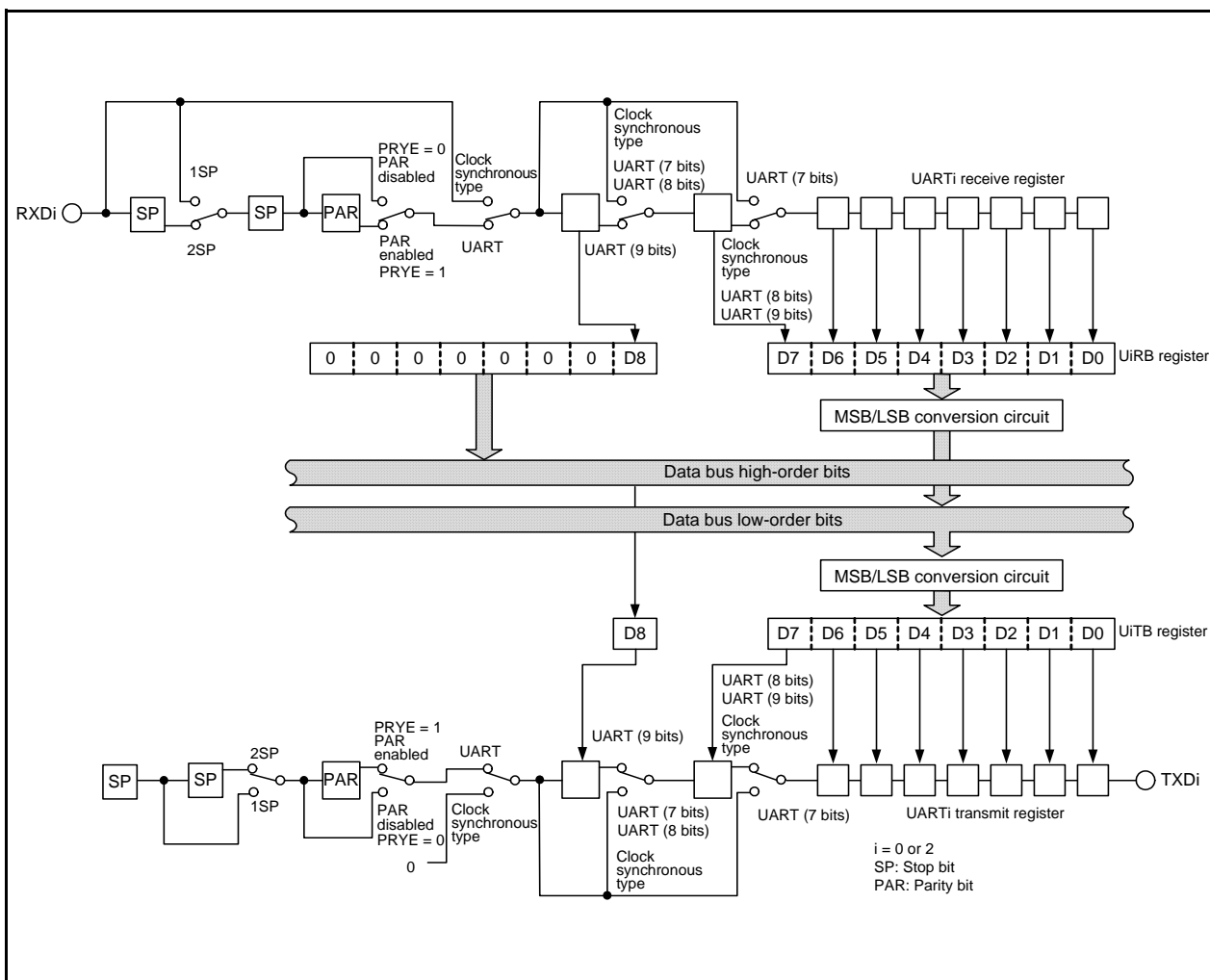


Figure 17.2 UARTi Transmit/Receive Unit



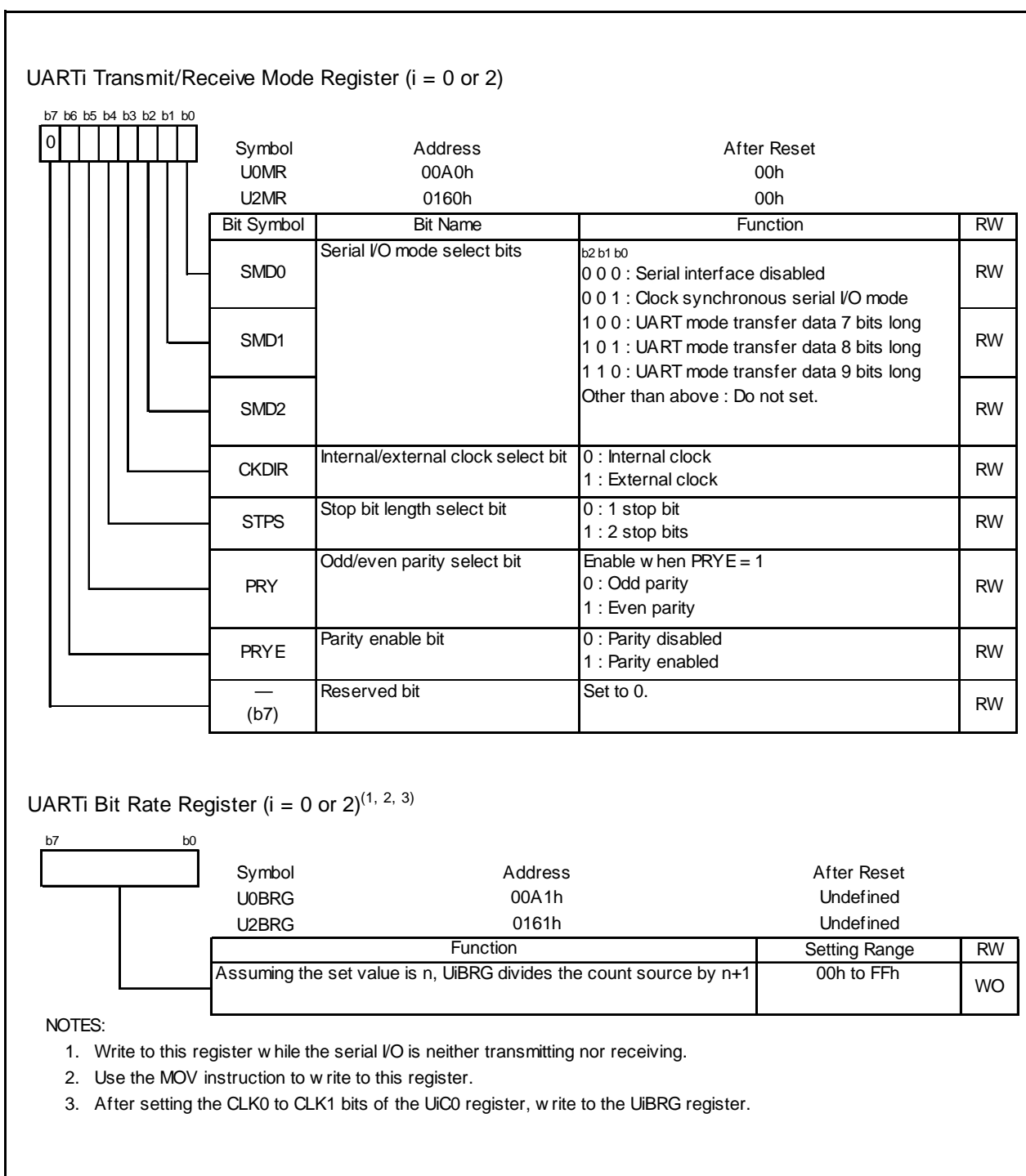


Figure 17.3 Registers U0MR, U2MR and U0BRG, U2BRG

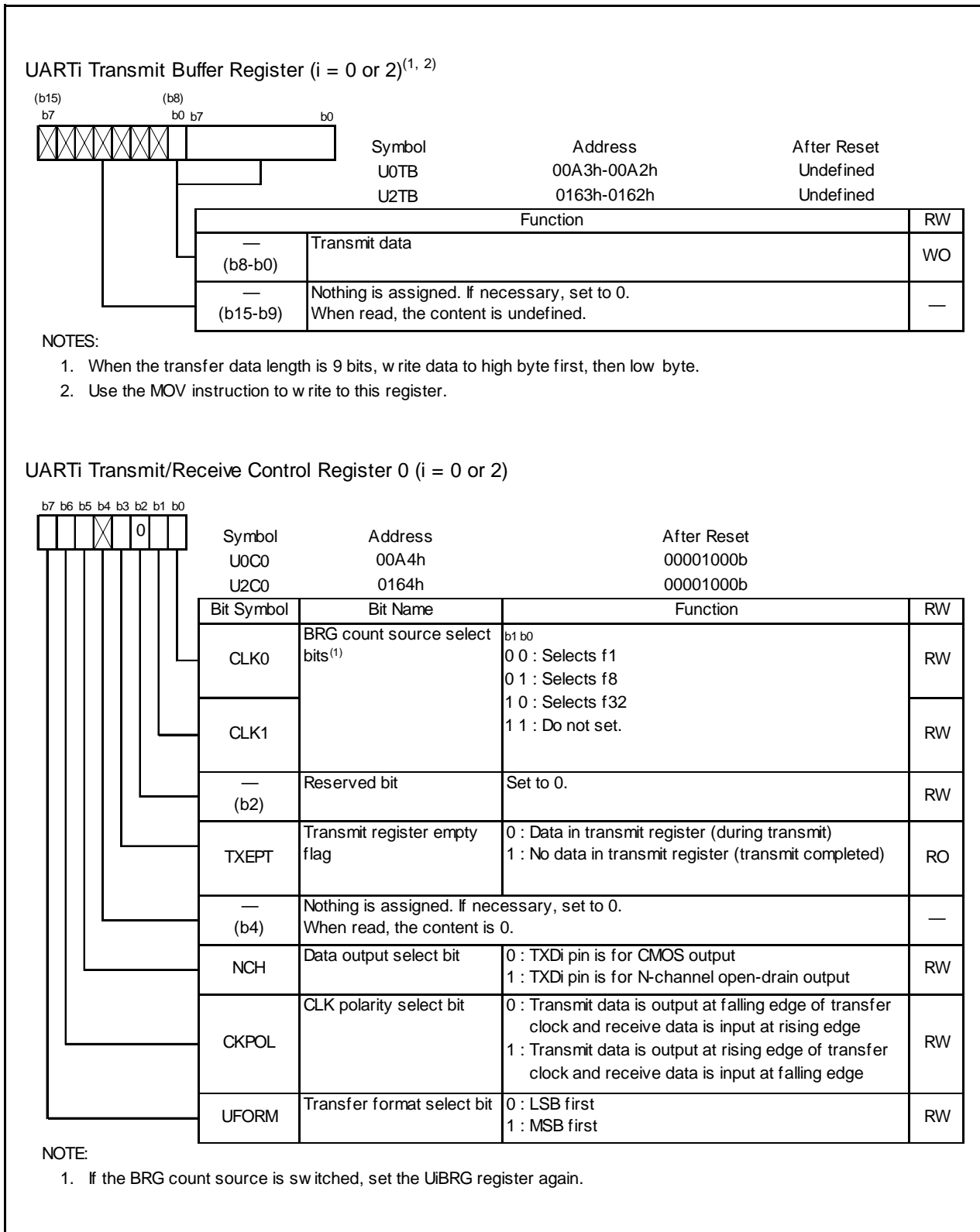


Figure 17.4 Registers U0TB, U2TB and U0C0, U2C0

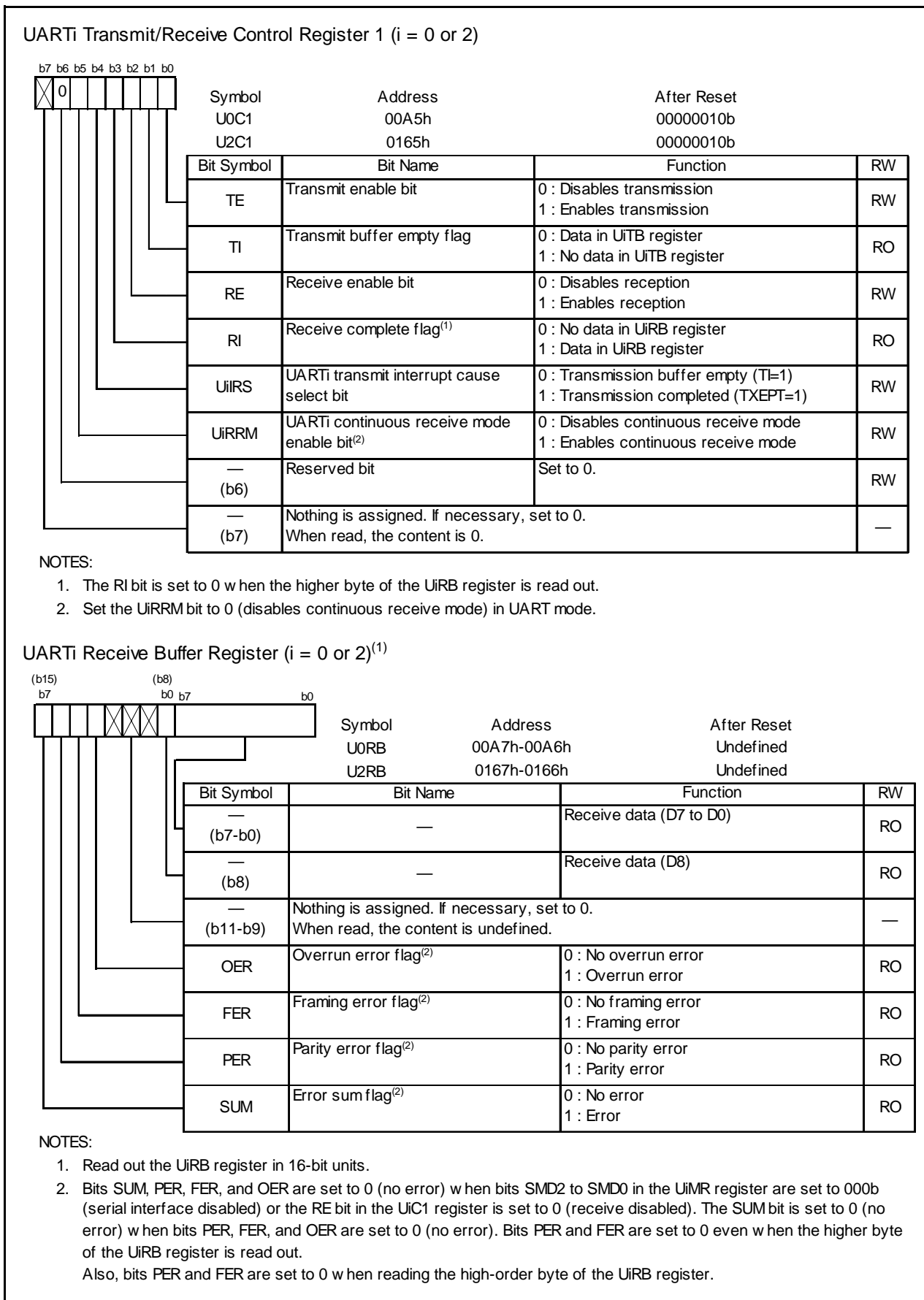


Figure 17.5 Registers U0C1, U2C1 and U0RB, U2RB

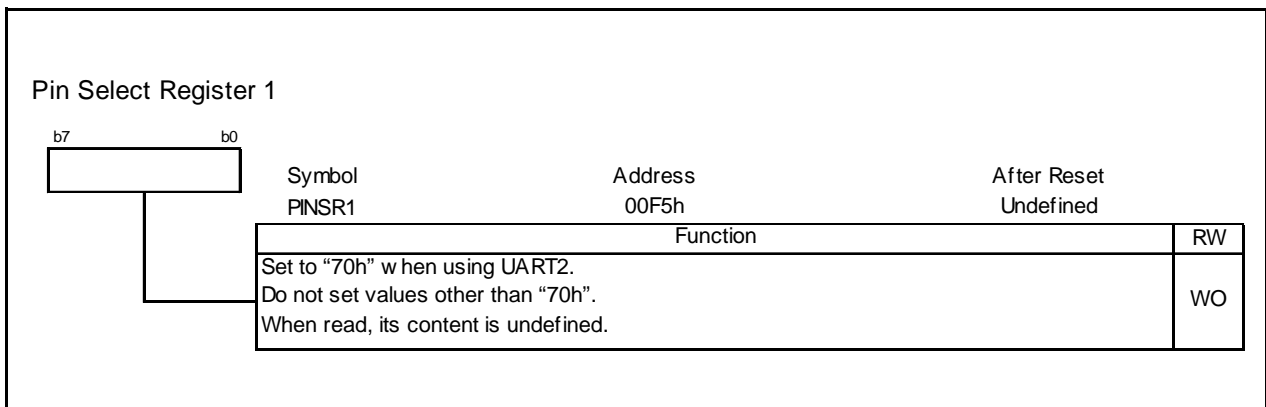


Figure 17.6 PINSR1 Register

## 17.1 Clock Synchronous Serial I/O Mode

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

Table 17.1 lists the Clock Synchronous Serial I/O Mode Specifications. Table 17.2 lists the Registers Used and Settings in Clock Synchronous Serial I/O Mode.

**Table 17.1 Clock Synchronous Serial I/O Mode Specifications**

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

i = 0 or 2

### NOTES:

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

**Table 17.2 Registers Used and Settings in Clock Synchronous Serial I/O Mode<sup>(1)</sup>**

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

i = 0 or 2

**NOTE:**

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

Table 17.3 lists the I/O Pin Functions in Clock Synchronous Serial I/O Mode. The TXDi pin outputs “H” level between the operating mode selection of UARTi (i = 0 or 2) and transfer start. (If the NCH bit is set to 1 (N-channel open-drain output), this pin is in a high-impedance state.)

**Table 17.3 I/O Pin Functions in Clock Synchronous Serial I/O Mode**

Pin Name	Function	Selection Method
TXD0 (P1_4)	Output serial data	(Outputs dummy data when performing reception only)
RXD0 (P1_5)	Input serial data	PD1_5 bit in PD1 register = 0 (P1_5 can be used as an input port when performing transmission only)
CLK0 (P1_6)	Output transfer clock	CKDIR bit in U0MR register = 0
	Input transfer clock	CKDIR bit in U0MR register = 1 PD1_6 bit in PD1 register = 0
TXD2 (P0_1)	Output serial data	(Outputs dummy data when performing reception only)
RXD2 (P0_2)	Input serial data	PD0_2 bit in PD0 register = 0 (P0_2 can be used as an input port when performing transmission only)
CLK2 (P0_3)	Output transfer clock	CKDIR bit in U2MR register = 0
	Input transfer clock	CKDIR bit in U2MR register = 1 PD0_3 bit in PD0 register = 0

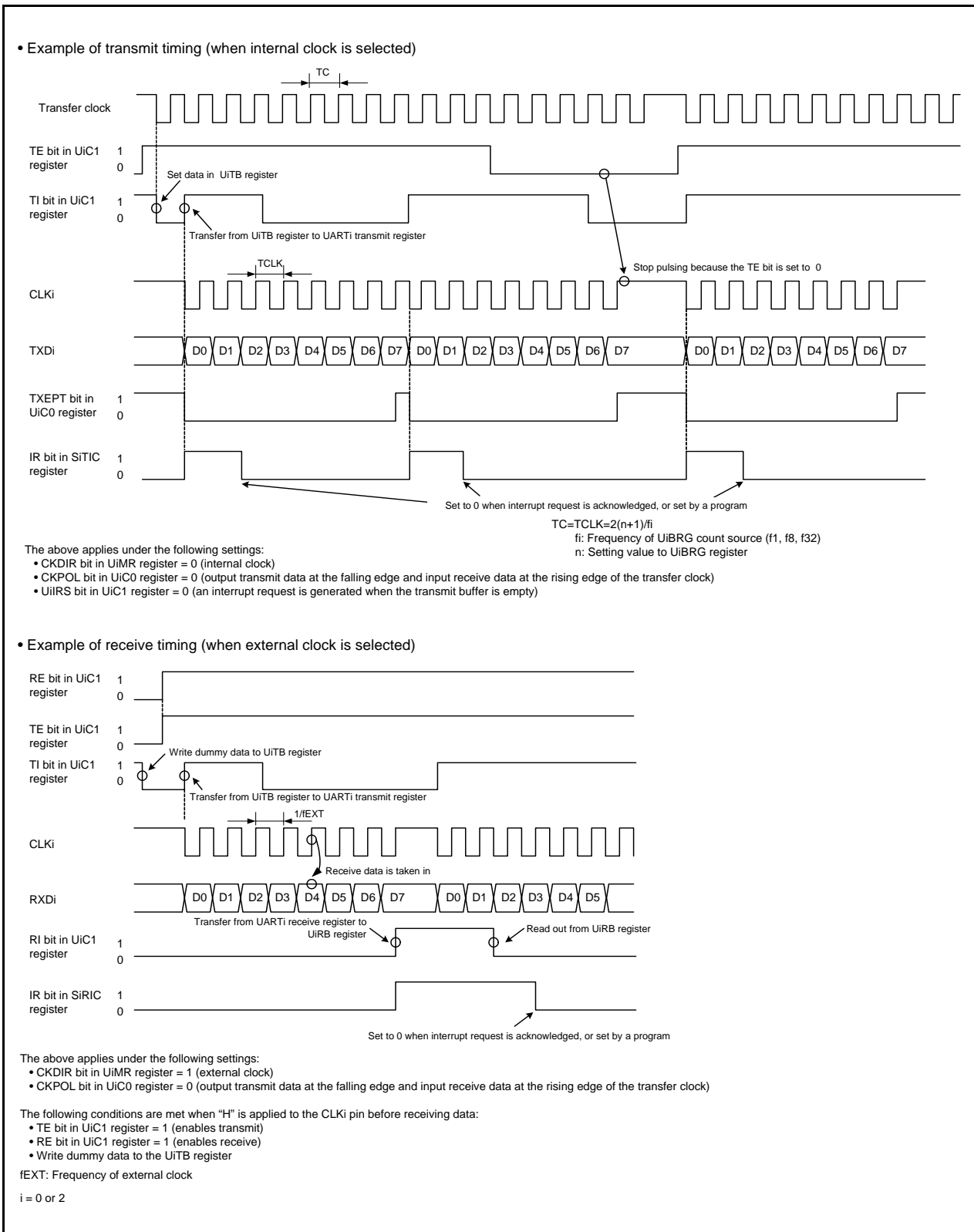


Figure 17.7 Transmit and Receive Timing Example in Clock Synchronous Serial I/O Mode

### 17.1.1 Polarity Select Function

Figure 17.8 shows the Transfer Clock Polarity. Use the CKPOL bit in the UiC0 (i = 0 or 2) register to select the transfer clock polarity.

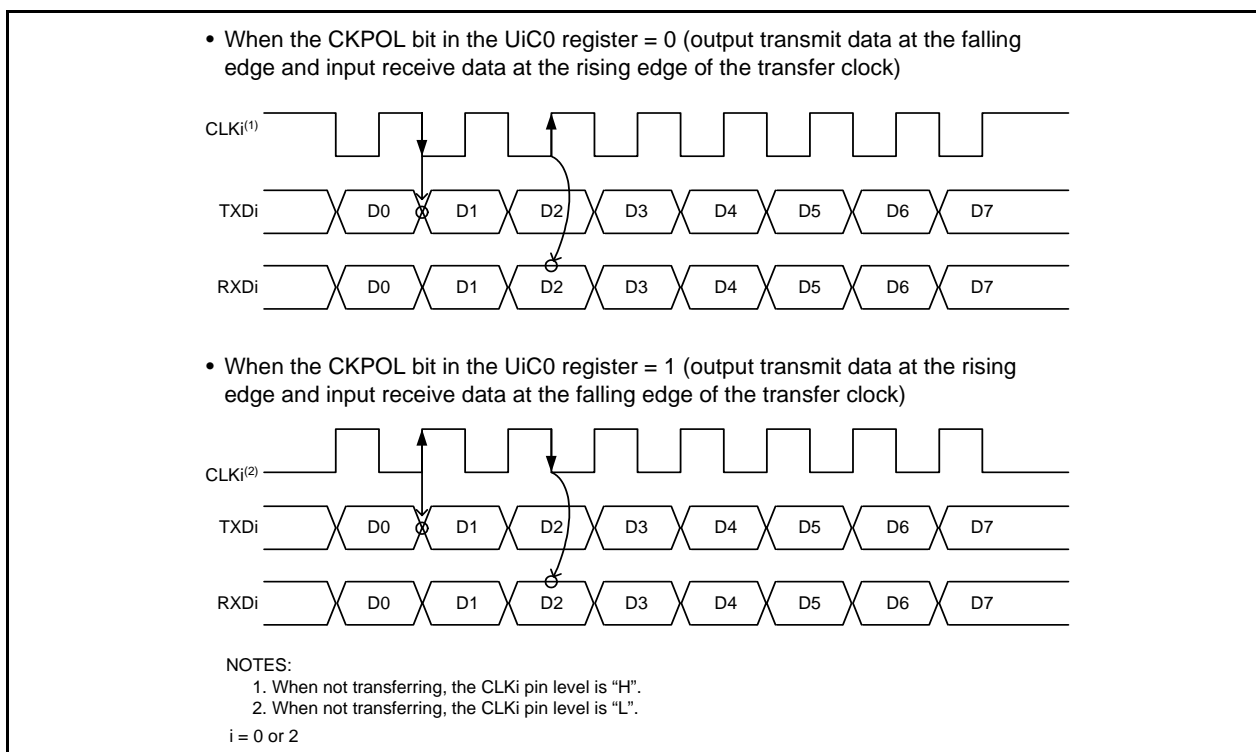


Figure 17.8 Transfer Clock Polarity

### 17.1.2 LSB First/MSB First Select Function

Figure 17.9 shows the Transfer Format. Use the UFORM bit in the UiC0 (i = 0 or 2) register to select the transfer format.

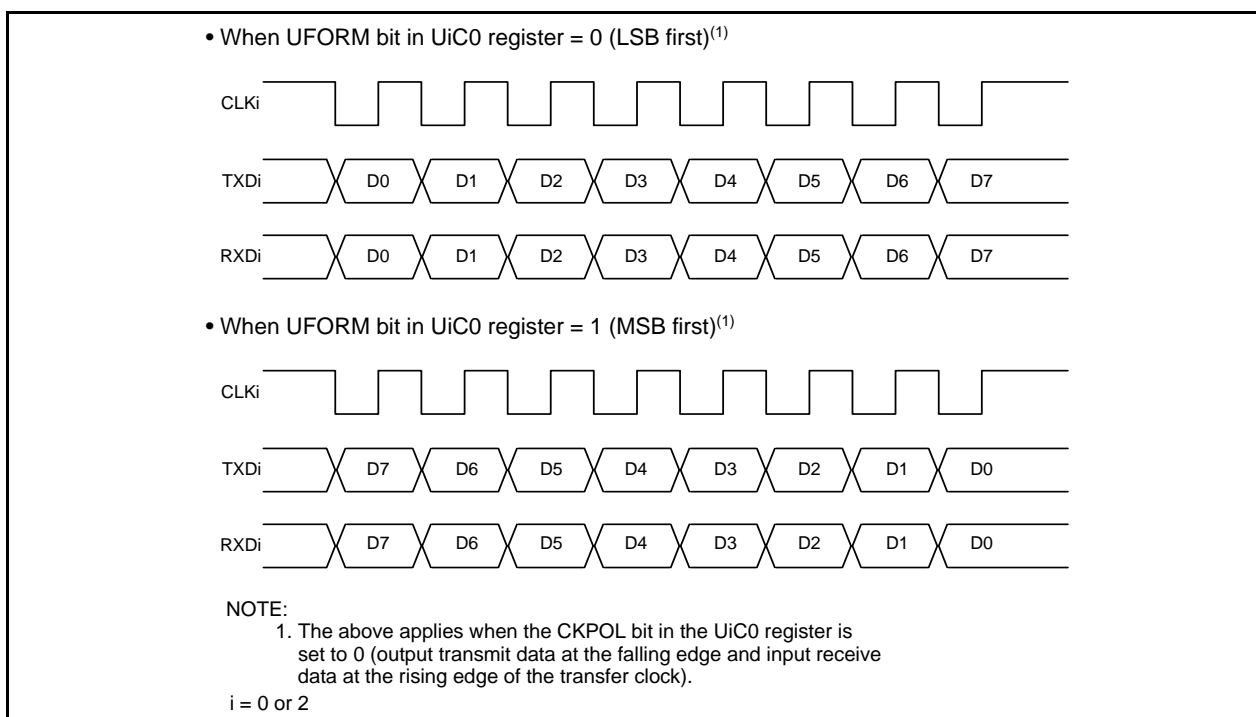


Figure 17.9 Transfer Format



### 17.1.3 Continuous Receive Mode

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

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

The UART mode allows data transmission and reception after setting the desired bit rate and transfer data format. Table 17.4 lists the UART Mode Specifications. Table 17.5 lists the Registers Used and Settings for UART Mode.

**Table 17.4 UART Mode Specifications**

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

i = 0 or 2

NOTE:

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

**Table 17.5 Registers Used and Settings for UART Mode**

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

i = 0 or 2

## NOTES:

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

Table 17.6 lists the I/O Pin Functions in UART Mode. After the UARTi (i = 0 or 2) operating mode is selected, the TXDi pin outputs “H” level. (If the NCH bit is set to 1 (N-channel open-drain output), this pin is in a high-impedance state) until transfer starts.)

**Table 17.6 I/O Pin Functions in UART Mode**

Pin name	Function	Selection Method
TXD0 (P1_4)	Output serial data	(Cannot be used as a port when performing reception only)
RXD0 (P1_5)	Input serial data	PD1_5 bit in PD1 register = 0 (P1_5 can be used as an input port when performing transmission only)
CLK0 (P1_6)	Programmable I/O Port	CKDIR bit in U0MR register = 0
	Input transfer clock	CKDIR bit in U0MR register = 1 PD1_6 bit in PD1 register = 0
TXD2 (P0_1)	Output serial data	(Cannot be used as a port when performing reception only)
RXD2 (P0_2)	Input serial data	PD0_2 bit in PD0 register = 0 (P0_2 can be used as an input port when performing transmission only)
CLK2 (P0_3)	Programmable I/O Port	CKDIR bit in U2MR register = 0
	Input transfer clock	CKDIR bit in U2MR register = 1 PD0_3 bit in PD0 register = 0

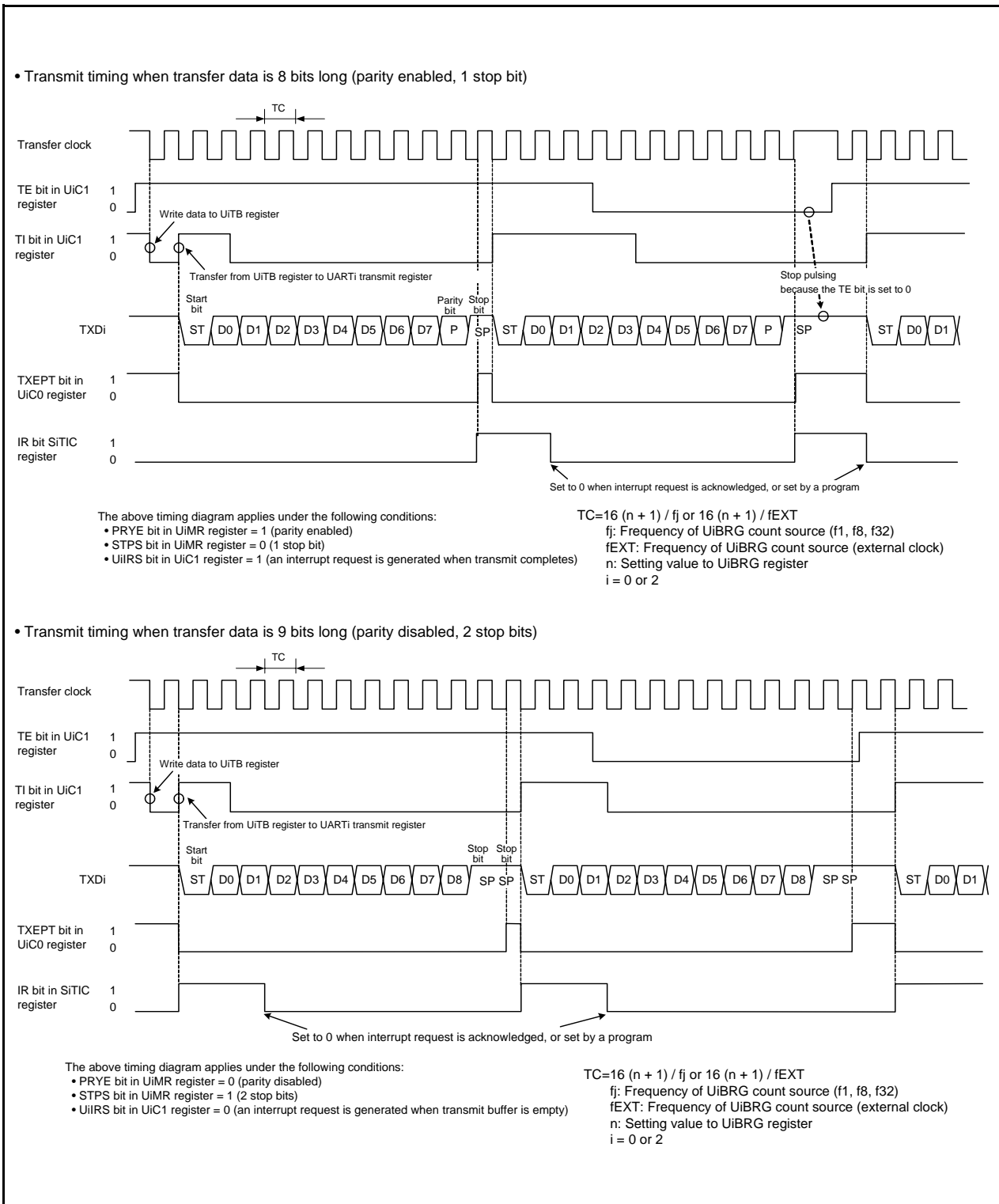


Figure 17.10 Transmit Timing in UART Mode

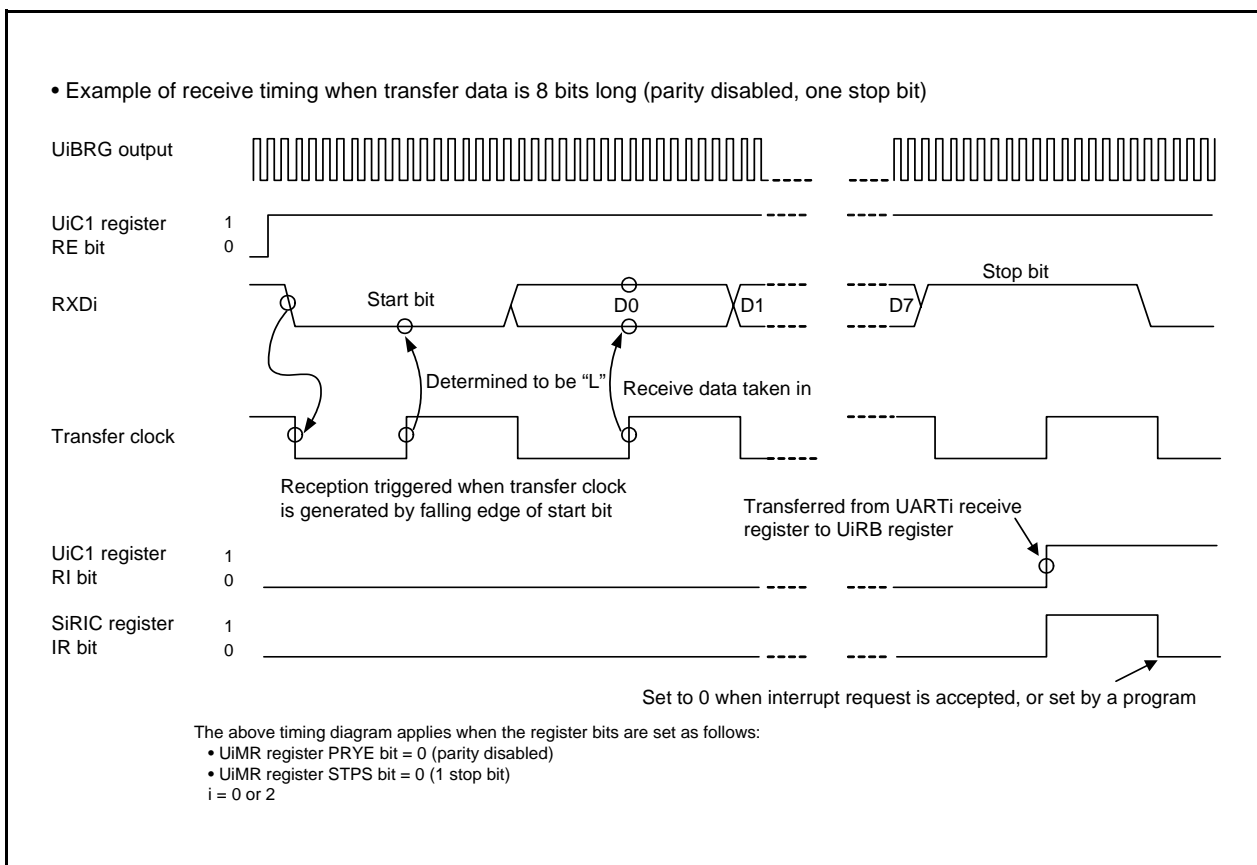
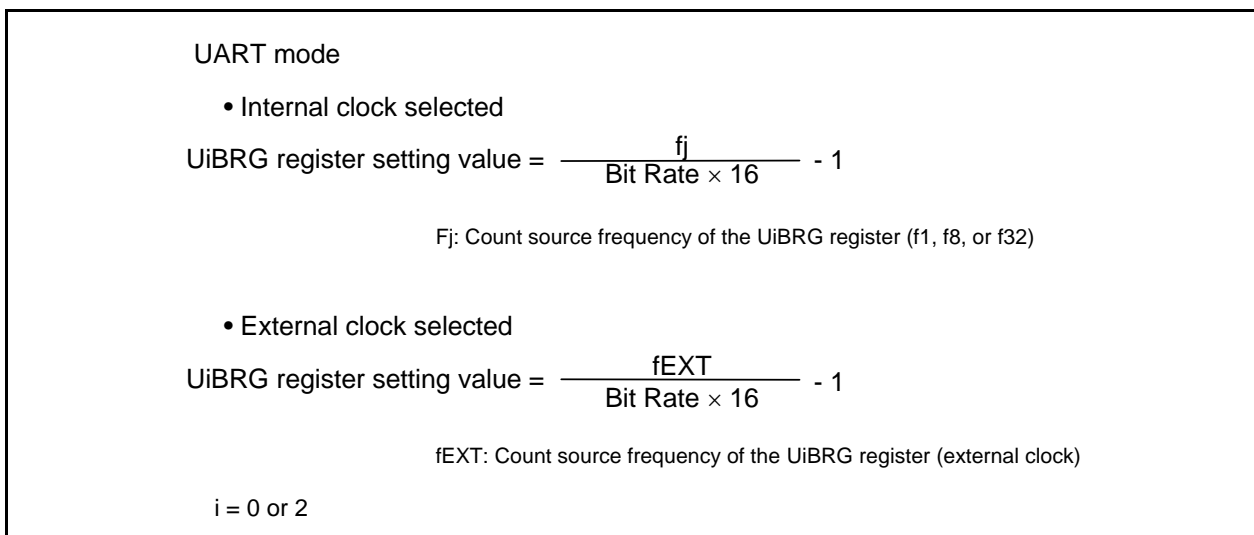


Figure 17.11 Receive Timing Example in UART Mode

### 17.2.1 Bit Rate

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

Figure 17.12 shows the Calculation Formula of UiBRG (i = 0 or 2) Register Setting Value. Table 17.7 lists the Bit Rate Setting Example in UART Mode (Internal Clock Selected).



**Figure 17.12 Calculation Formula of UiBRG (i = 0 or 2) Register Setting Value**

**Table 17.7 Bit Rate Setting Example in UART Mode (Internal Clock Selected)**

Bit Rate (bps)	UiBRG Count Source	System Clock = 20 MHz			System Clock = 18.432 MHz <sup>(1)</sup>			System Clock = 8 MHz		
		UiBRG Setting Value	Actual Time (bps)	Setting Error (%)	UiBRG Setting Value	Actual Time (bps)	Setting Error (%)	UiBRG Setting Value	Actual Time (bps)	Setting Error (%)
1200	f8	129 (81h)	1201.92	0.16	119 (77h)	1200.00	0.00	51 (33h)	1201.92	0.16
2400	f8	64 (40h)	2403.85	0.16	59 (3Bh)	2400.00	0.00	25 (19h)	2403.85	0.16
4800	f8	32 (20h)	4734.85	-1.36	29 (1Dh)	4800.00	0.00	12 (0Ch)	4807.69	0.16
9600	f1	129 (81h)	9615.38	0.16	119 (77h)	9600.00	0.00	51 (33h)	9615.38	0.16
14400	f1	86 (56h)	14367.82	-0.22	79 (4Fh)	14400.00	0.00	34 (22h)	14285.71	-0.79
19200	f1	64 (40h)	19230.77	0.16	59 (3Bh)	19200.00	0.00	25 (19h)	19230.77	0.16
28800	f1	42 (2Ah)	29069.77	0.94	39 (27h)	28800.00	0.00	16 (10h)	29411.76	2.12
38400	f1	32 (20h)	37878.79	-1.36	29 (1Dh)	38400.00	0.00	12 (0Ch)	38461.54	0.16
57600	f1	21 (15h)	56818.18	-1.36	19 (13h)	57600.00	0.00	8 (08h)	55555.56	-3.55
115200	f1	10 (0Ah)	113636.36	-1.36	9 (09h)	115200.00	0.00	–	–	–

i = 0 to 1

NOTES:

- For the high-speed on-chip oscillator, the correction value in the FRA7 register should be written into the FRA1 register.

This applies when the high-speed on-chip oscillator is selected as the system clock and bits FRA22 to FRA20 in the FRA2 register are set to 000b (divide-by-2 mode). For the precision of the high-speed on-chip oscillator, refer to **22. Electrical Characteristics**.

### 17.3 Notes on Serial Interface

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

Example (when reading receive buffer register):

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

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

Example (when reading transmit buffer register):

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

## 18. Hardware LIN

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

### 18.1 Features

The hardware LIN has the features listed below.

Figure 18.1 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.

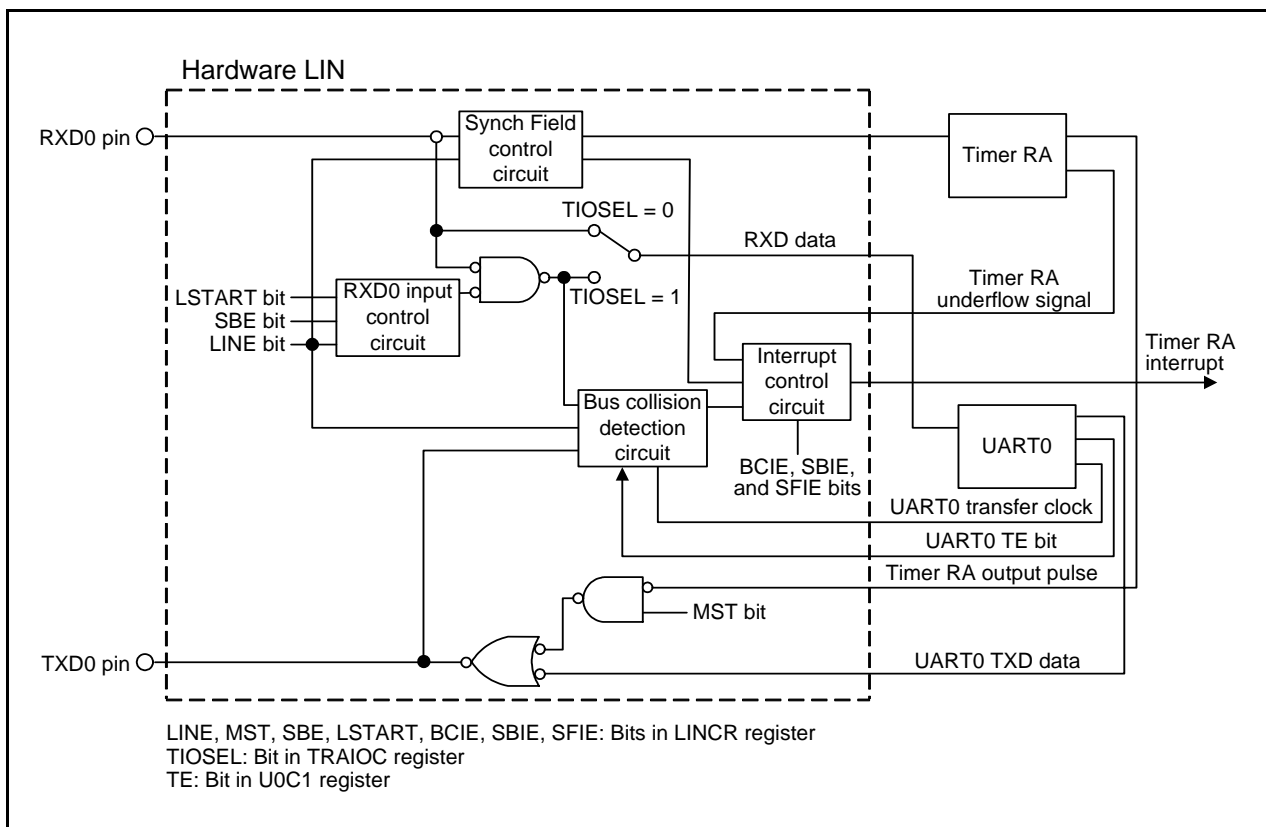


Figure 18.1 Block Diagram of Hardware LIN



## 18.2 Input/Output Pins

The pin configuration of the hardware LIN is listed in Table 18.1.

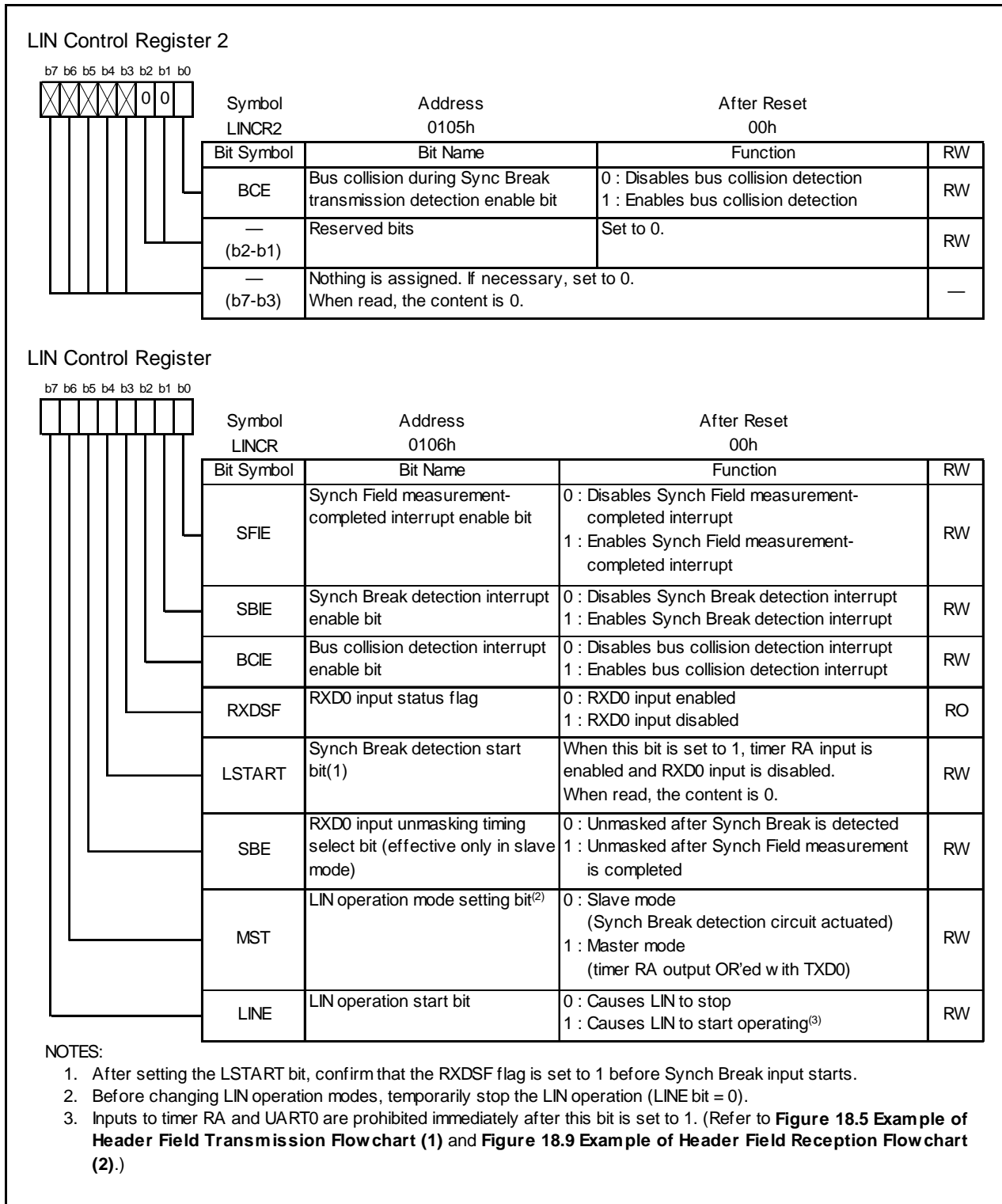
**Table 18.1 Pin Configuration**

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

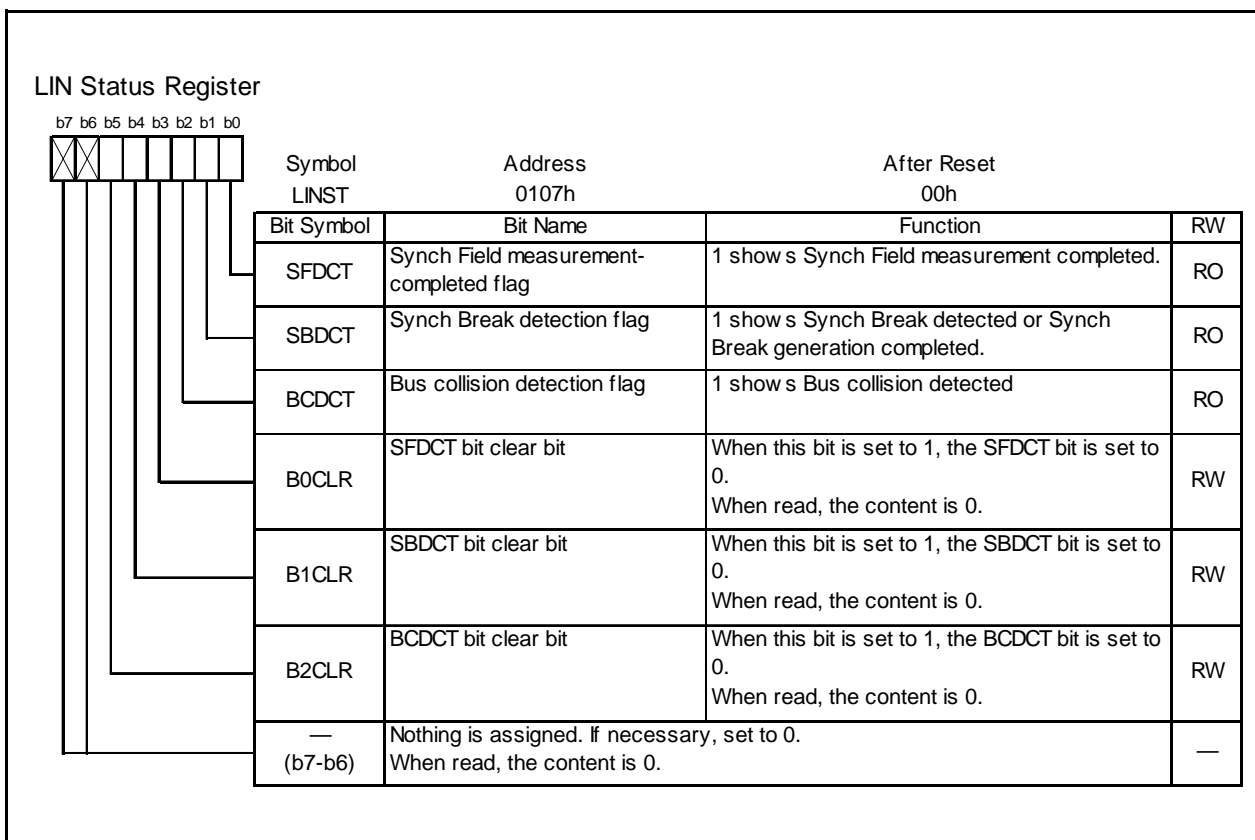
### 18.3 Register Configuration

The hardware LIN contains the registers listed below.  
These registers are detailed in Figures 18.2 and 18.3.

- LIN Control Register 2 (LINCR2)
- LIN Control Register (LINCR)
- LIN Status Register (LINST)



**Figure 18.2 Registers LINCR2 and LINCR**



**Figure 18.3 LINST Register**

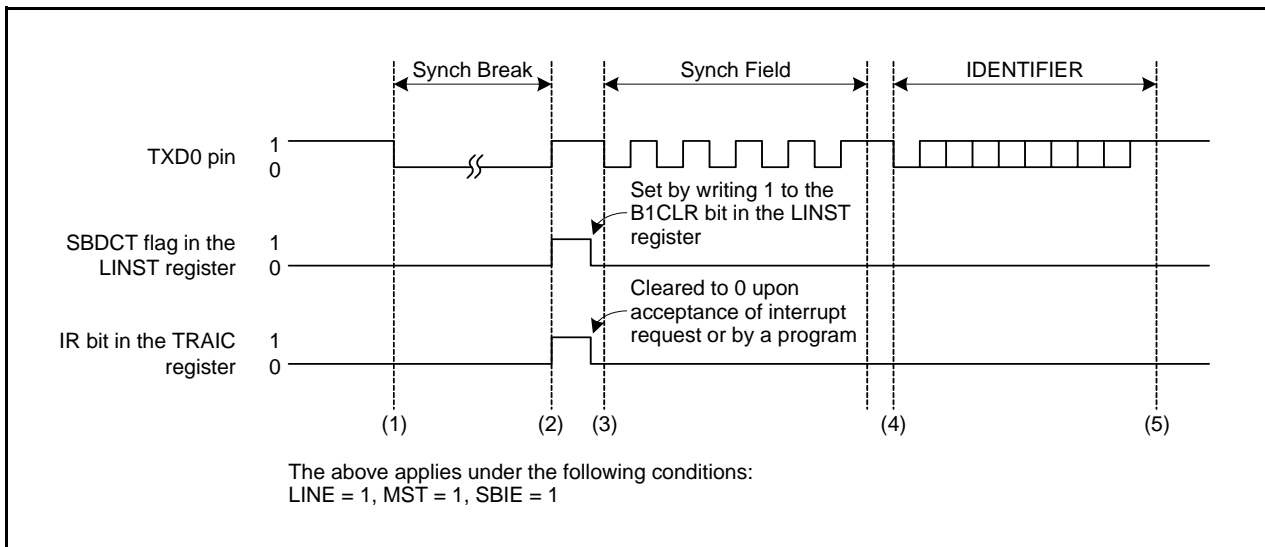
## 18.4 Functional Description

### 18.4.1 Master Mode

Figure 18.4 shows typical operation of the hardware LIN when transmitting a header field in master mode. Figures 18.5 and 18.6 show an Example of Header Field Transmission Flowchart.

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

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



**Figure 18.4** Typical Operation when Sending a Header Field

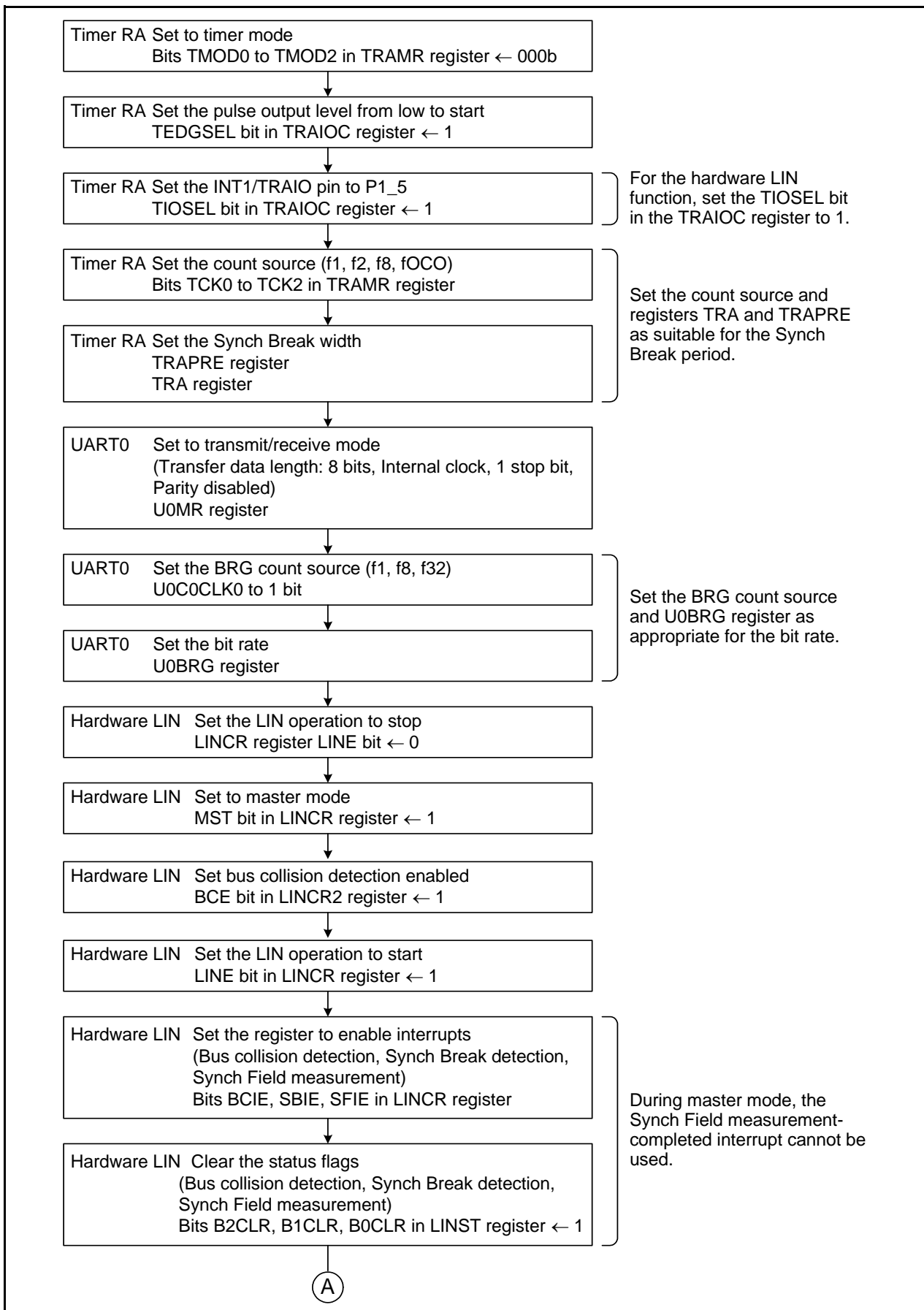


Figure 18.5 Example of Header Field Transmission Flowchart (1)

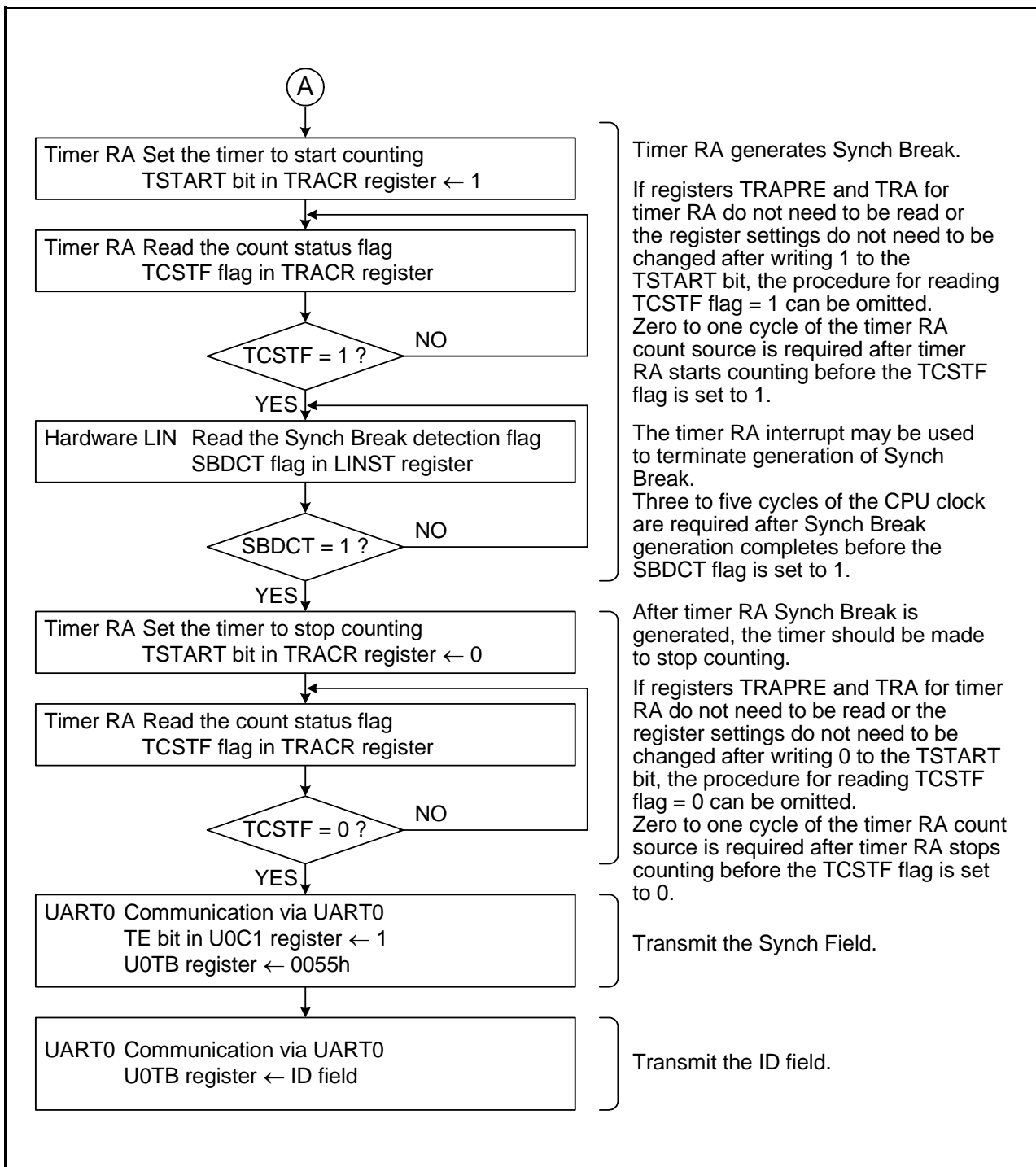


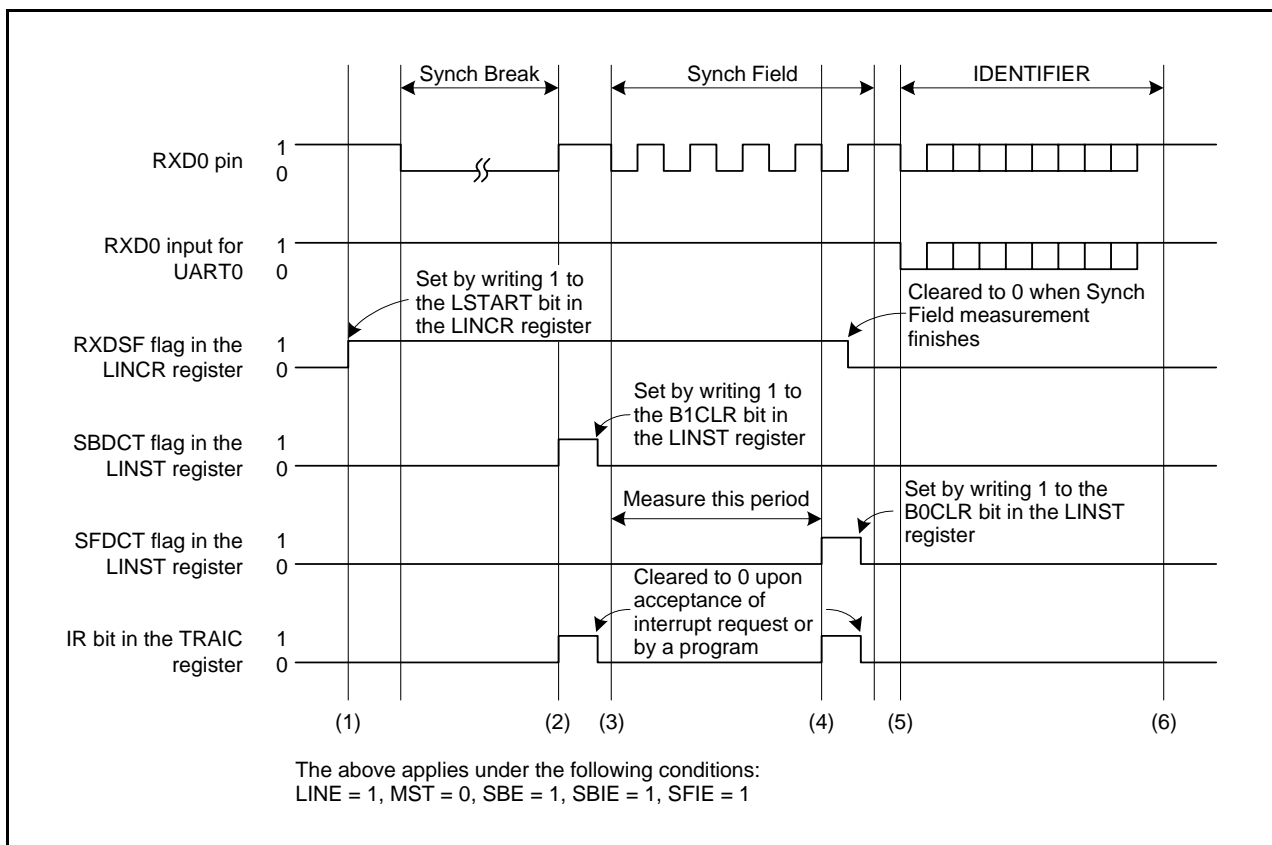
Figure 18.6 Example of Header Field Transmission Flowchart (2)

### 18.4.2 Slave Mode

Figure 18.7 shows typical operation of the hardware LIN when receiving a header field in slave mode. Figure 18.8 through Figure 18.10 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 LINCRC register of the hardware LIN.
- (2) When “L” level is input for a duration equal to or greater than the period set in timer RA, the hardware LIN detects it as Synch Break. At this time, the SBDCT flag in the LINST register is set to 1. Furthermore, if the SBIE bit in the LINCRC 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 LINCRC register accordingly.
- (4) The hardware LIN sets the SFDCT flag in the LINST register to 1 when it finishes measuring the Synch Field. Furthermore, if the SFIE bit in the LINCRC register is set to 1, it generates a timer RA interrupt.
- (5) After it finishes measuring the Synch Field, calculate a transfer rate from the count value of timer RA and set to UART0 and registers TRAPRE and TRA of timer RA again.
- (6) The hardware LIN performs communication for a response field after it finishes receiving the ID field.



**Figure 18.7 Typical Operation when Receiving a Header Field**

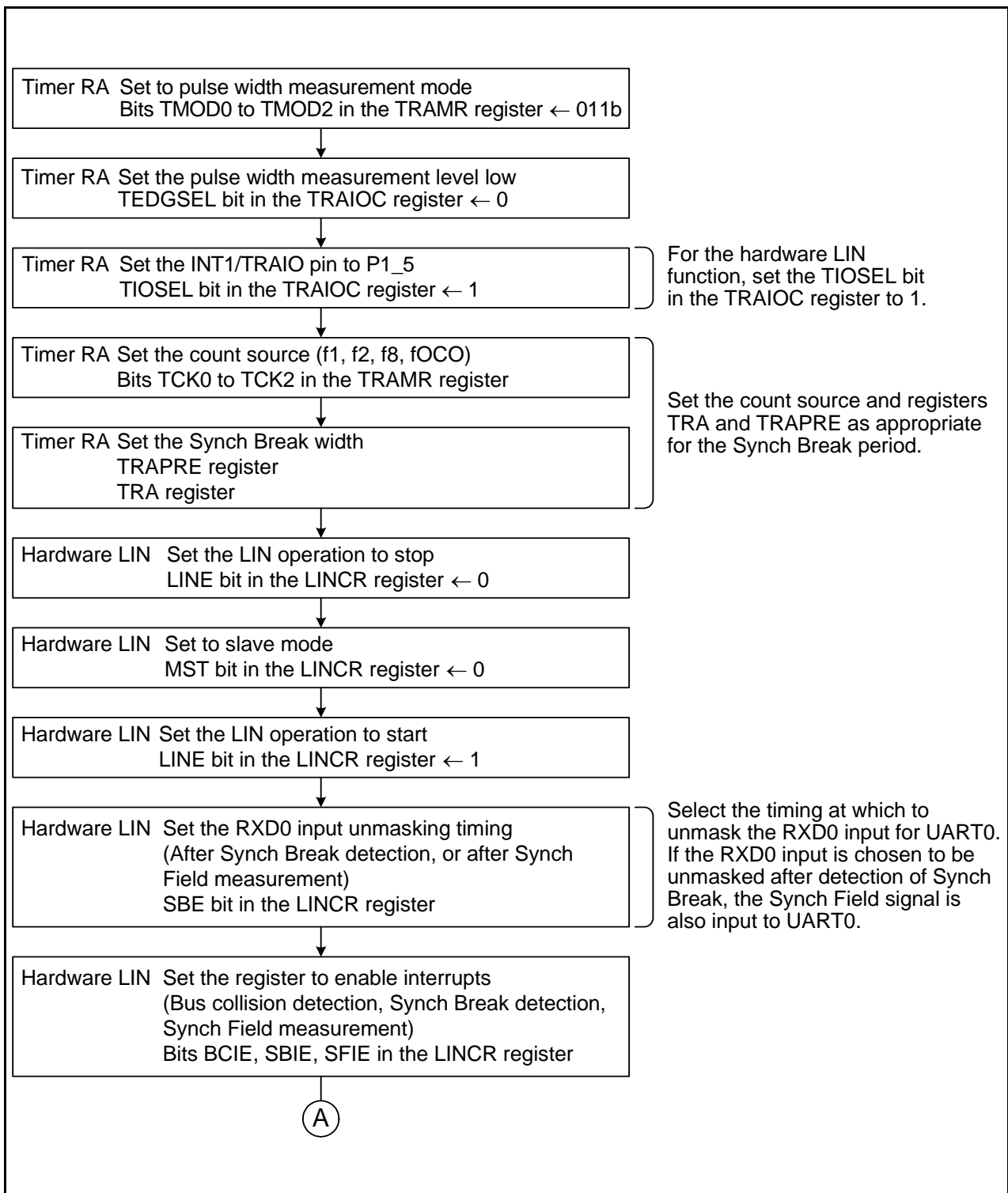


Figure 18.8 Example of Header Field Reception Flowchart (1)



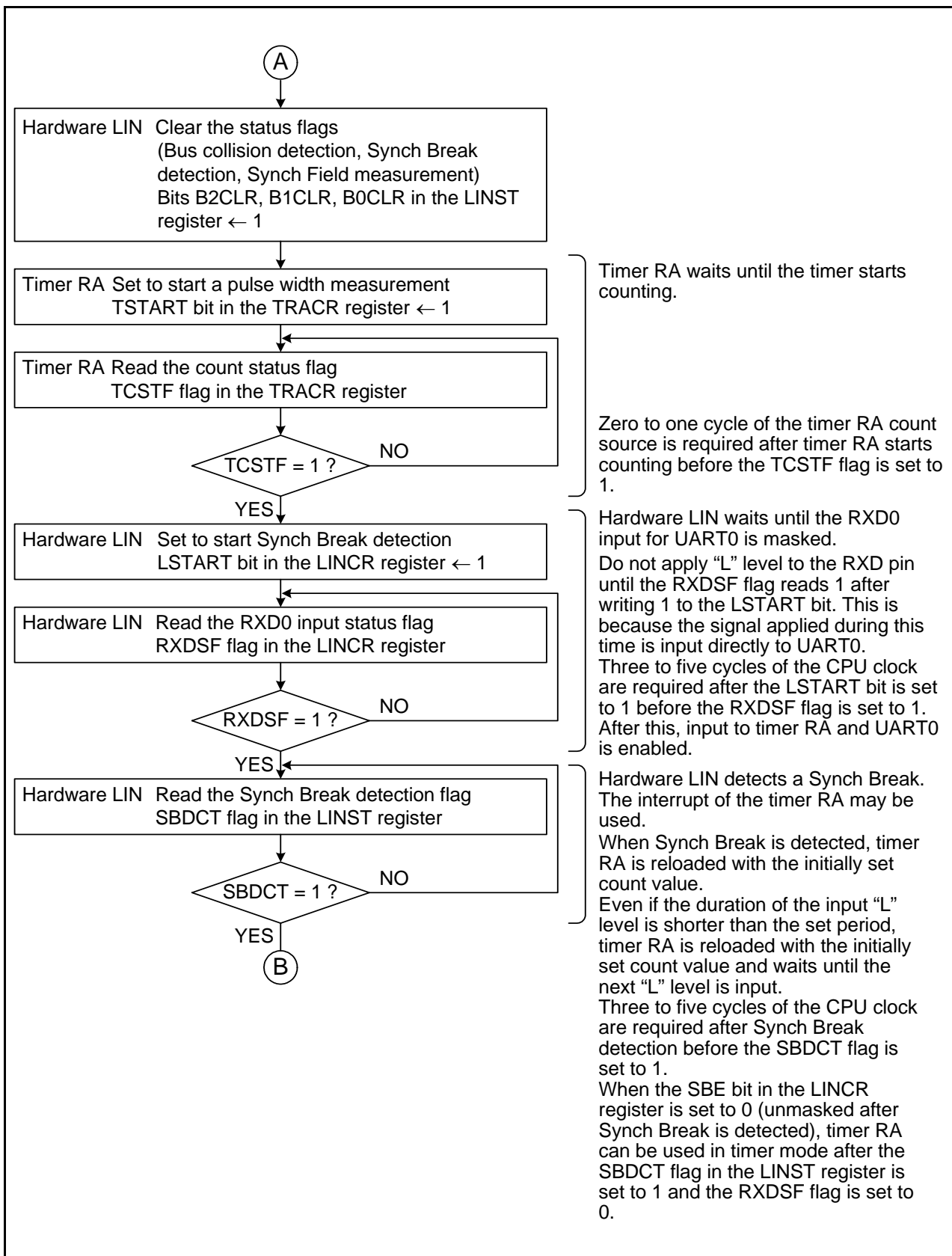


Figure 18.9 Example of Header Field Reception Flowchart (2)

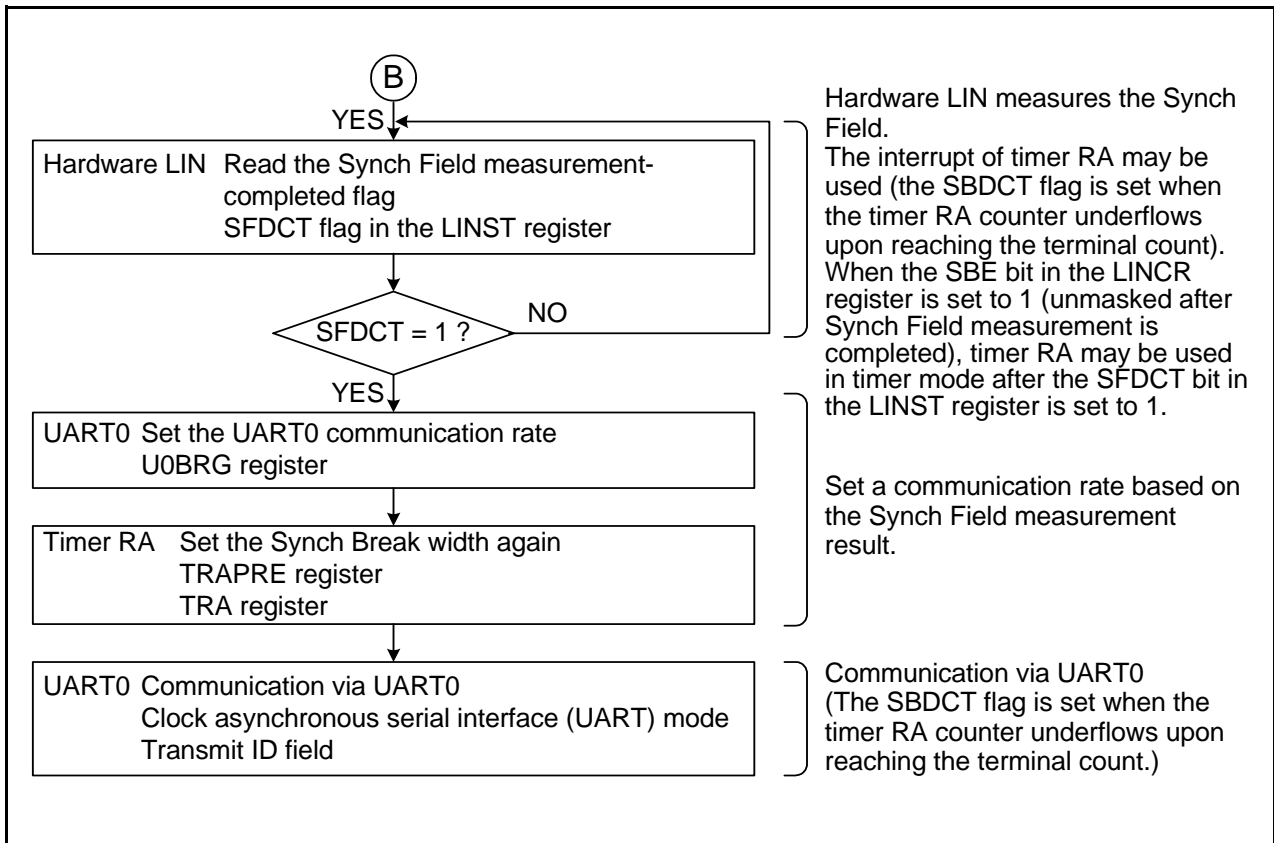


Figure 18.10 Example of Header Field Reception Flowchart (3)

### 18.4.3 Bus Collision Detection Function

The bus collision detection function can be used when UART0 is enabled for transmission (TE bit in the U0C1 register = 1). To detect a bus collision during Synch Break transmission, set the BCE bit in the LINCR2 register to 1 (bus collision detection enabled).

Figure 18.11 shows the Typical Operation when a Bus Collision is Detected.

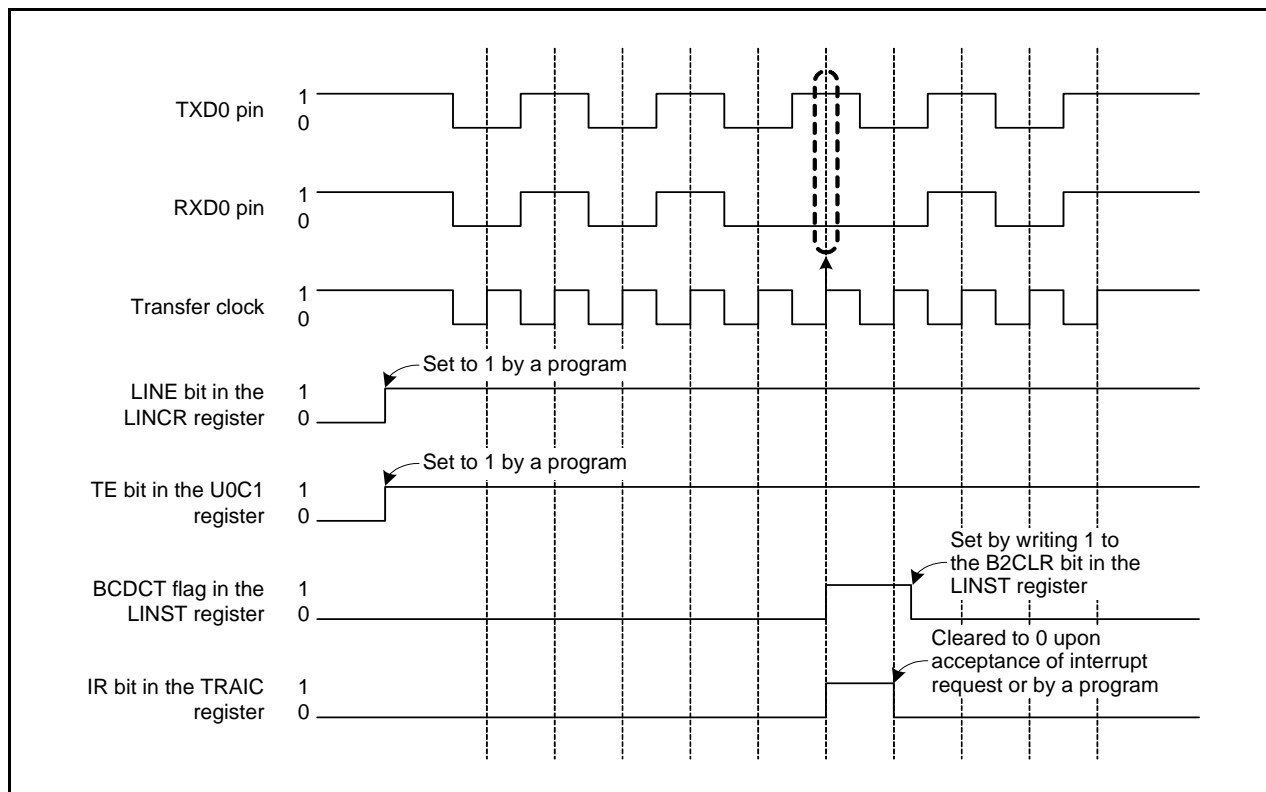


Figure 18.11 Typical Operation when a Bus Collision is Detected

### 18.4.4 Hardware LIN End Processing

Figure 18.12 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.

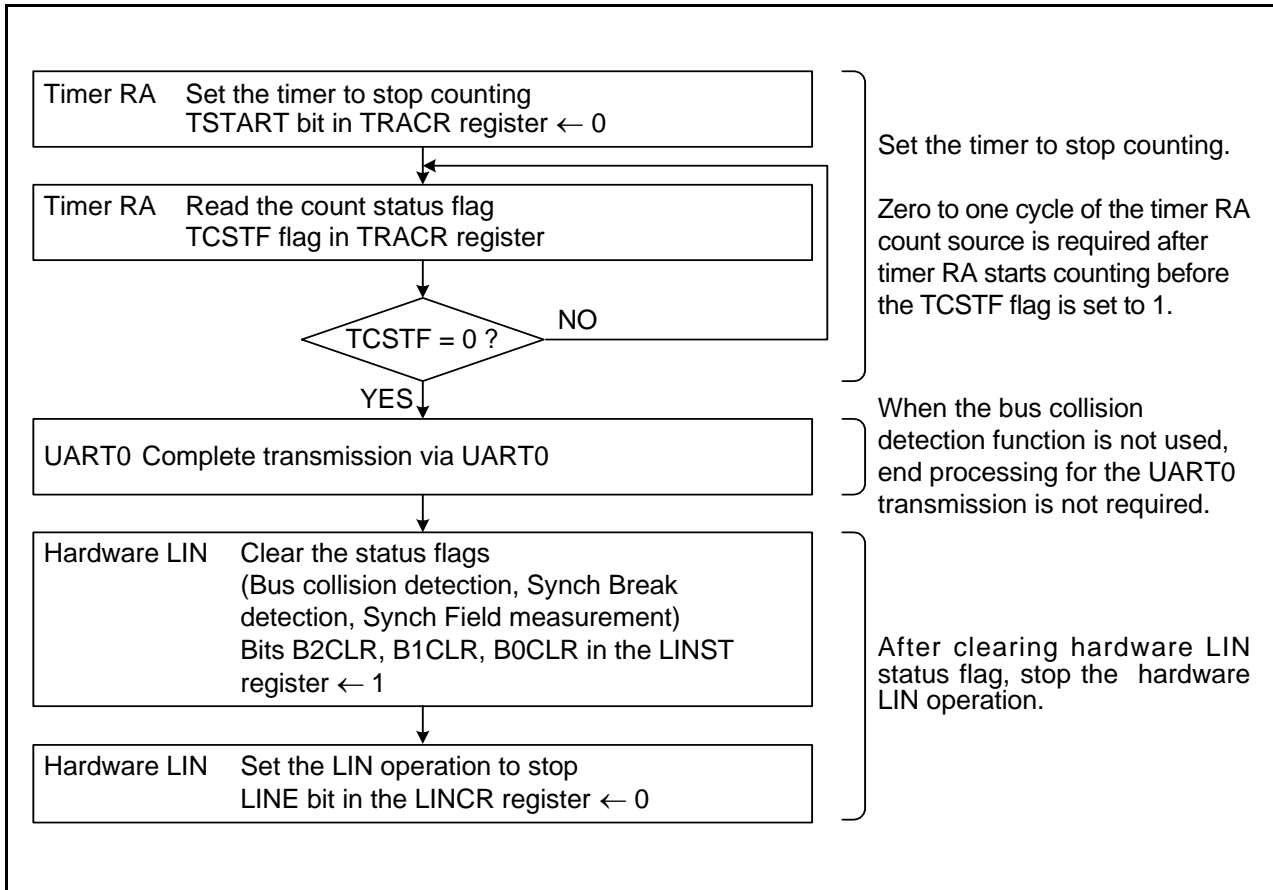


Figure 18.12 Example of Hardware LIN Communication Completion Flowchart

## 18.5 Interrupt Requests

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

Table 18.2 lists the Interrupt Requests of Hardware LIN.

**Table 18.2 Interrupt Requests of Hardware LIN**

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

## 18.6 Notes on Hardware LIN

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

## 19. 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 pins P0\_0 to P0\_3, P0\_5, and P1\_0 to P1\_3. Therefore, when using these pins, ensure that 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. This helps to reduce the power consumption of the chip.

The result of A/D conversion is stored in the AD register.

Table 19.1 lists the Performance of A/D converter. Figure 19.1 shows a Block Diagram of A/D Converter.

Figures 19.2 and 19.3 show the A/D converter-related registers.

**Table 19.1 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>	$4.2\text{ V} \leq AVCC \leq 5.5\text{ V}$ f1, f2, f4, fOCO-F $2.7\text{ V} \leq AVCC < 4.2\text{ V}$ f2, f4, fOCO-F
Resolution	8 bits or 10 bits selectable
Absolute accuracy	$AVCC = V_{ref} = 5\text{ V}$ , $\phi_{AD} = 10\text{ MHz}$ <ul style="list-style-type: none"> <li>• 8-bit resolution <math>\pm 2\text{ LSB}</math></li> <li>• 10-bit resolution <math>\pm 3\text{ LSB}</math></li> </ul> $AVCC = V_{ref} = 3.3\text{ V}$ , $\phi_{AD} = 10\text{ MHz}$ <ul style="list-style-type: none"> <li>• 8-bit resolution <math>\pm 2\text{ LSB}</math></li> <li>• 10-bit resolution <math>\pm 5\text{ LSB}</math></li> </ul>
Operating mode	One-shot and repeat <sup>(3)</sup>
Analog input pin	9 pins (AN2, AN4 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: <math>49\phi_{AD}</math> cycles, 10-bit resolution: <math>59\phi_{AD}</math> cycles</li> <li>• With sample and hold function 8-bit resolution: <math>28\phi_{AD}</math> cycles, 10-bit resolution: <math>33\phi_{AD}</math> cycles</li> </ul>

### NOTES:

1. The analog input voltage does not depend on use of a sample and hold function.  
When the analog input voltage is over the reference voltage, the A/D conversion result will be 3FFh in 10-bit mode and FFh in 8-bit mode.
2. When  $2.7\text{ V} \leq AVCC \leq 5.5\text{ V}$ , the frequency of  $\phi_{AD}$  must be 10 MHz or below.  
Without a sample and hold function, the  $\phi_{AD}$  frequency should be 250 kHz or above.  
With a 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.

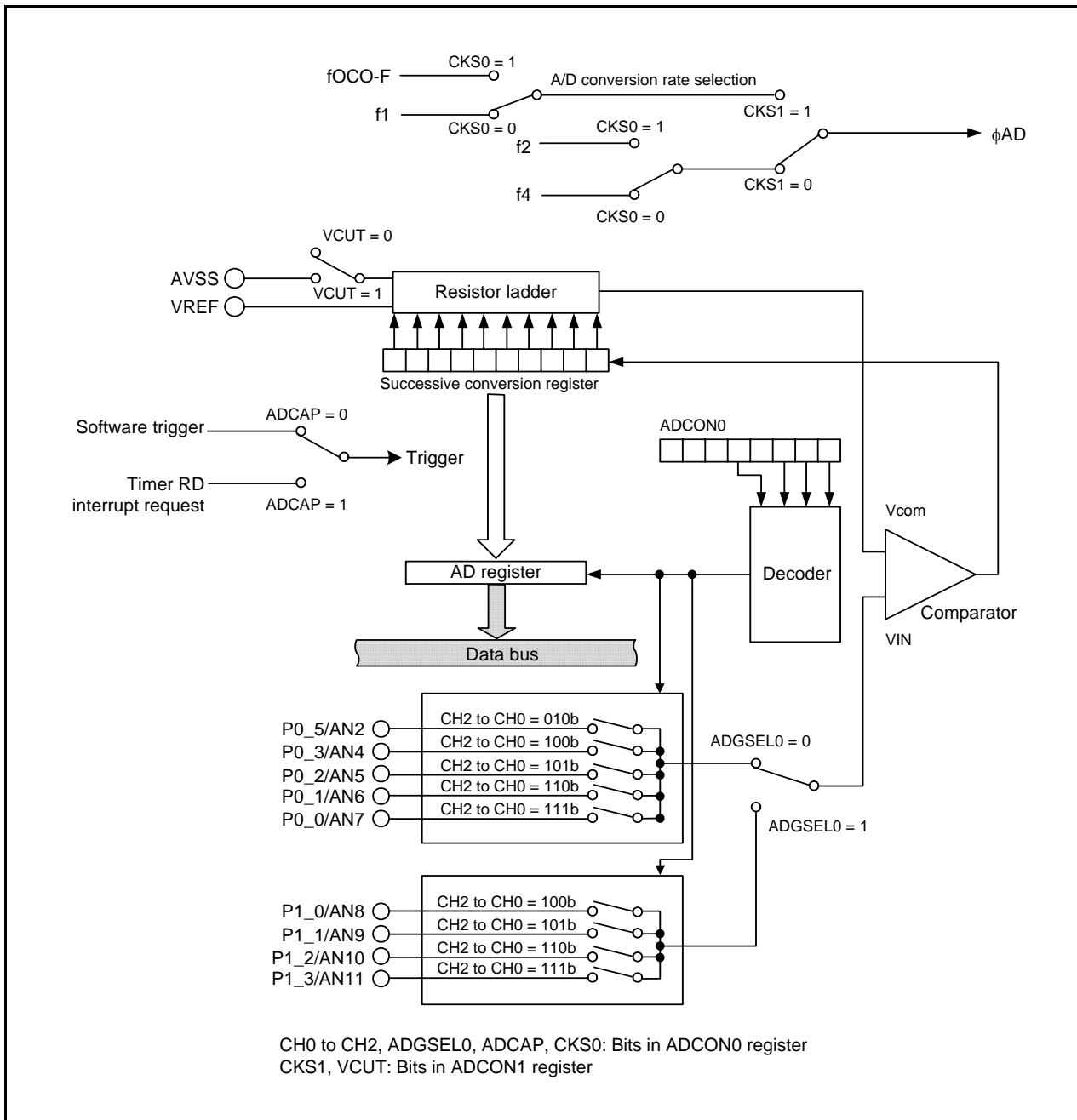


Figure 19.1 Block Diagram of A/D Converter



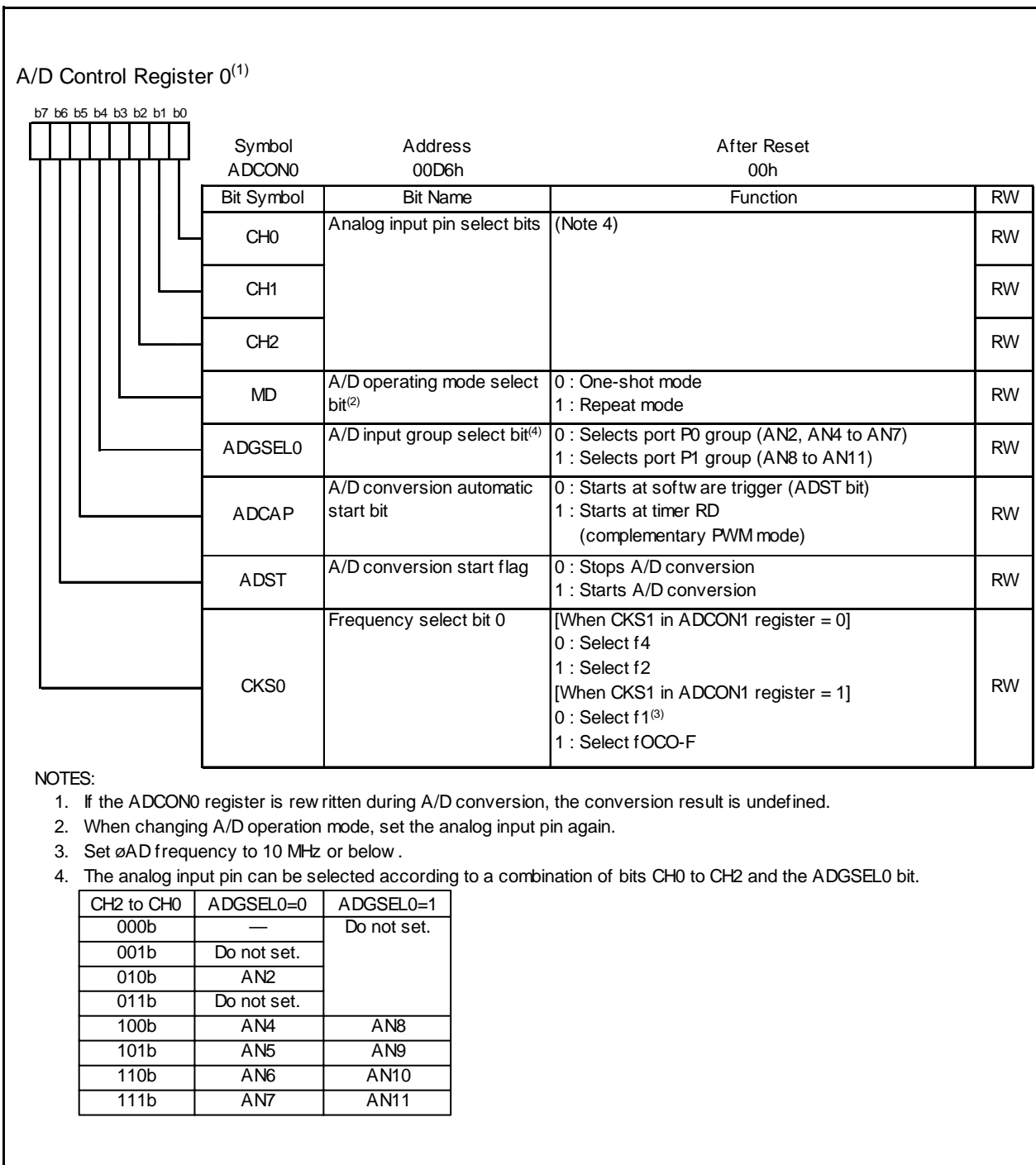


Figure 19.2 ADCON0 Register

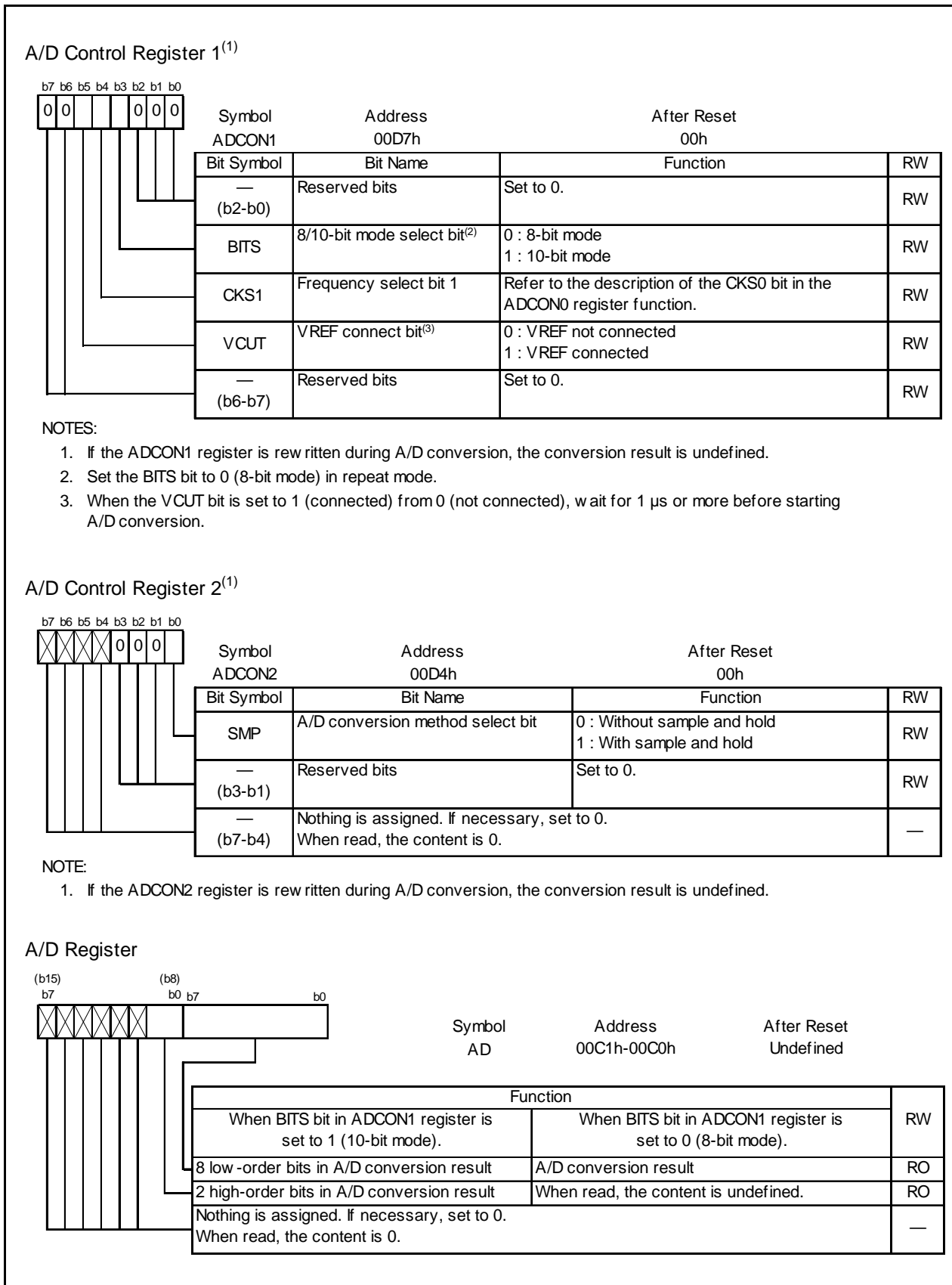


Figure 19.3 Registers ADCON1, ADCON2, and AD

## 19.1 One-Shot Mode

In one-shot mode, the input voltage of one selected pin is A/D converted once.

Table 19.2 lists the Specification of One-Shot Mode. Figures 19.4 and 19.5 show Registers ADCON0 and ADCON1 in One-Shot Mode.

**Table 19.2 Specification of One-Shot Mode**

Item	Specification
Function	The input voltage of one pin selected 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): A compare match between registers TRD0 and TRDGRA0 or a 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), 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 AN2, AN4 to AN11
Reading of A/D conversion result	Read AD register

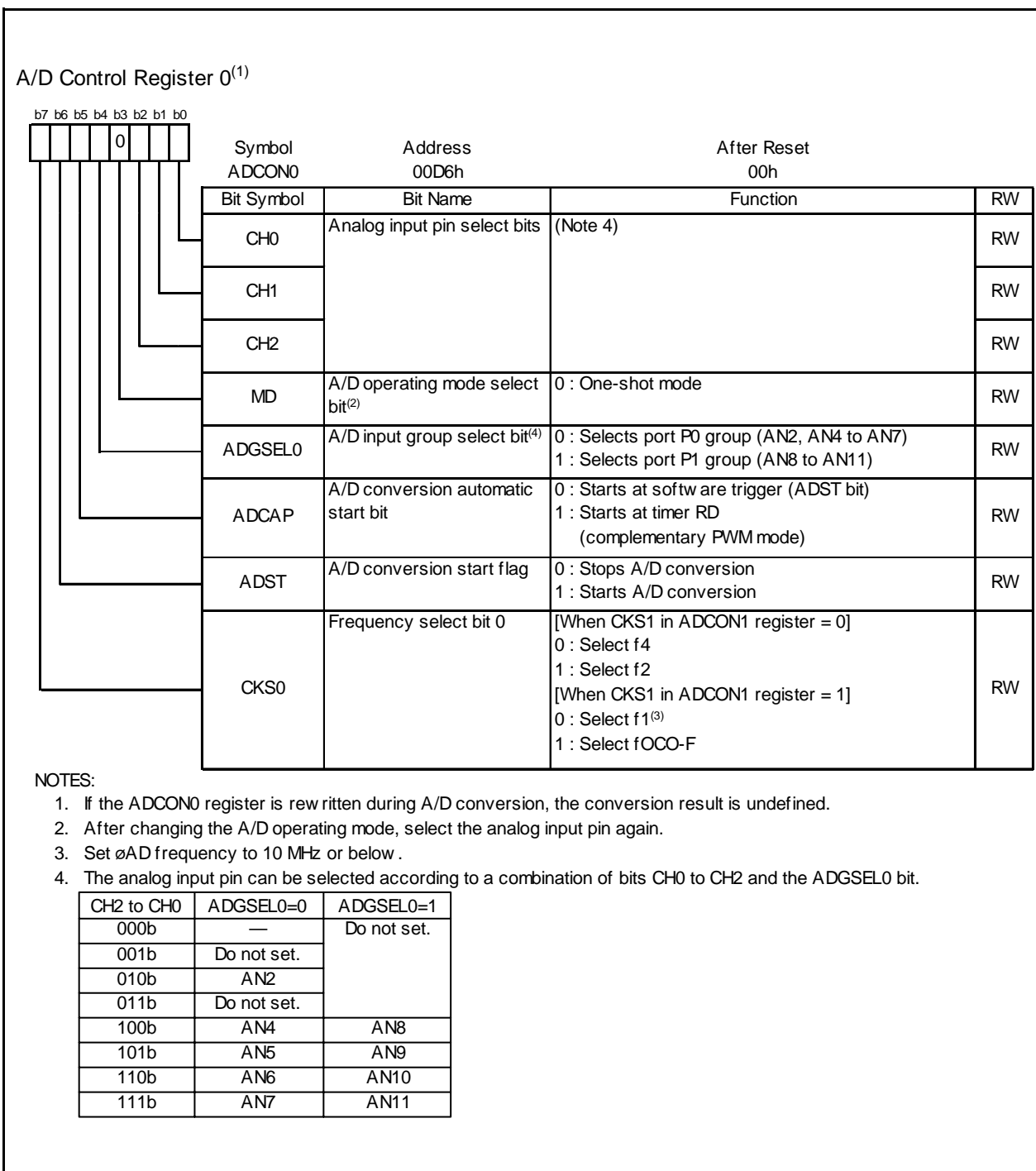


Figure 19.4 ADCON0 Register in One-Shot Mode

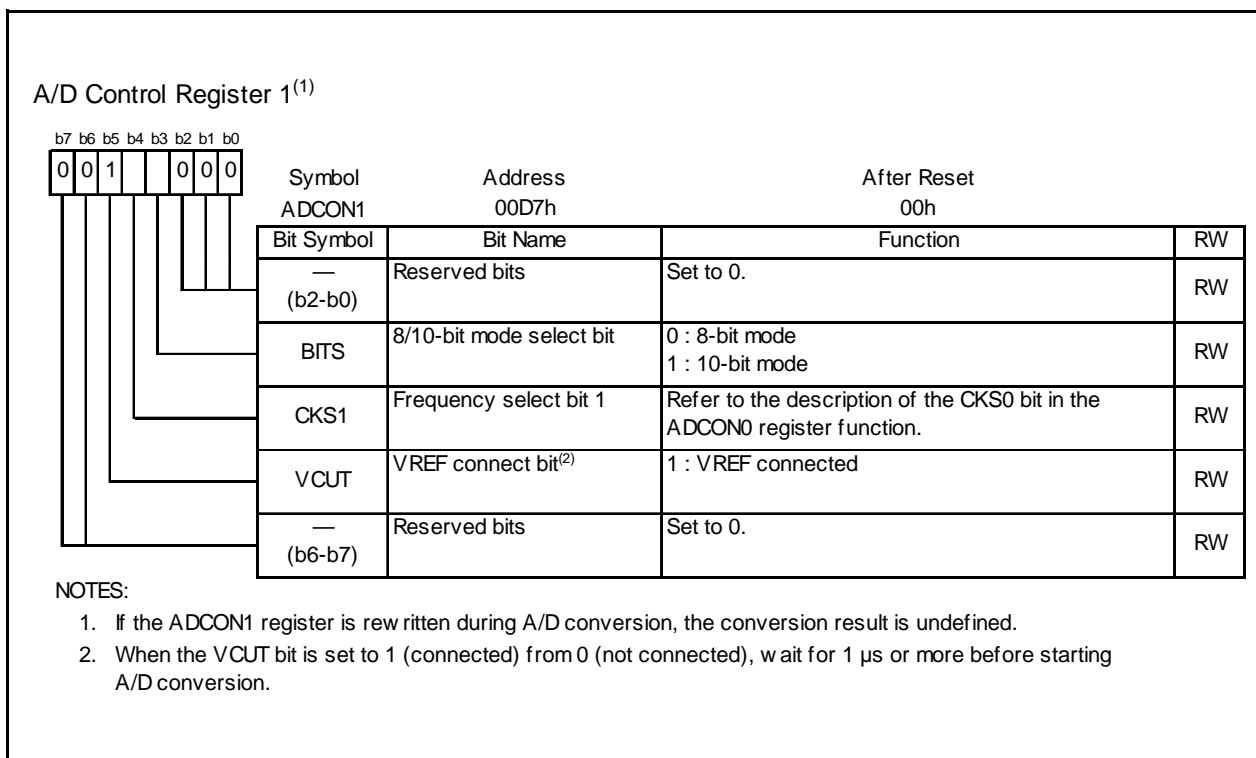


Figure 19.5 ADCON1 Register in One-Shot Mode

## 19.2 Repeat Mode

In repeat mode, the input voltage of one selected pin is A/D converted repeatedly.

Table 19.3 lists the Repeat Mode Specifications. Figures 19.6 and 19.7 show Registers ADCON0 and ADCON1 in Repeat Mode.

**Table 19.3 Repeat Mode Specifications**

Item	Specification
Function	The Input voltage of one pin selected by bits CH2 to CH0 and ADGSEL0 is A/D converted repeatedly
Start conditions	<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): A compare match between registers TRD0 and TRDGRA0 or a 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 AN2, AN4 to AN11
Reading of result of A/D converter	Read AD register

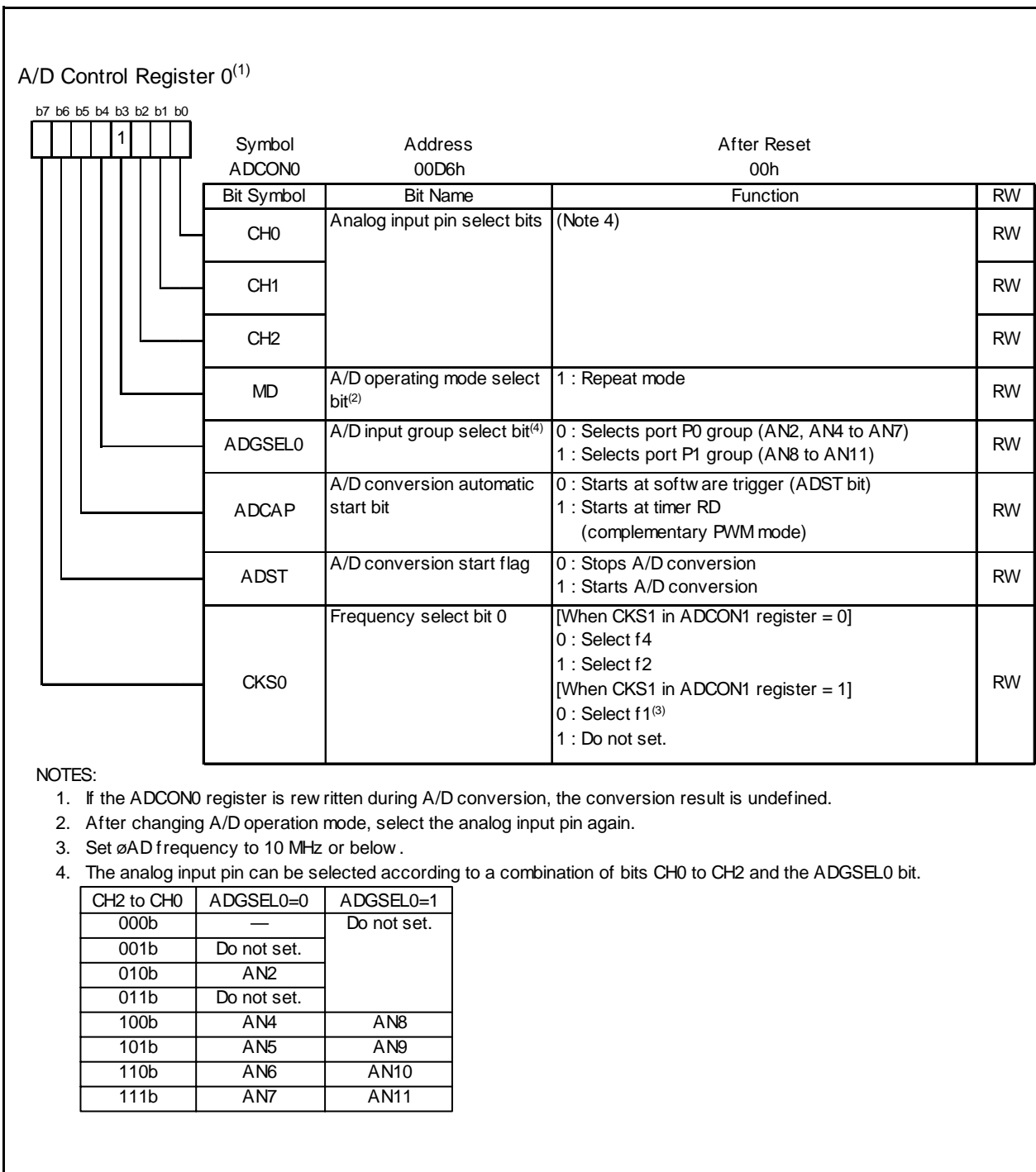


Figure 19.6 ADCON0 Register in Repeat Mode

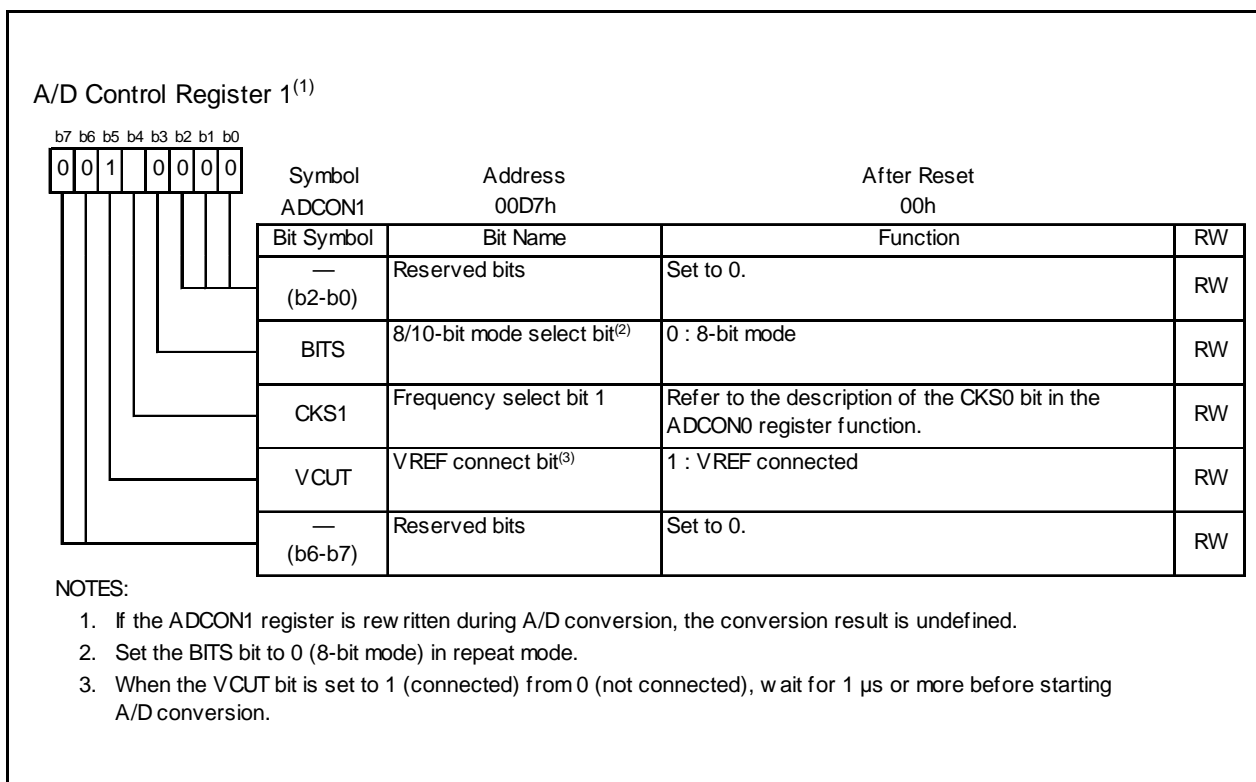


Figure 19.7 ADCON1 Register in Repeat Mode



### 19.3 Sample and Hold

When the SMP bit in the ADCON2 register is set to 1 (sample and hold function enabled), the A/D conversion rate per pin increases. The sample and hold function is available in all operating modes. Start A/D conversion after selecting whether the sample and hold circuit is to be used or not.

Figure 19.8 shows a Timing Diagram of A/D Conversion.

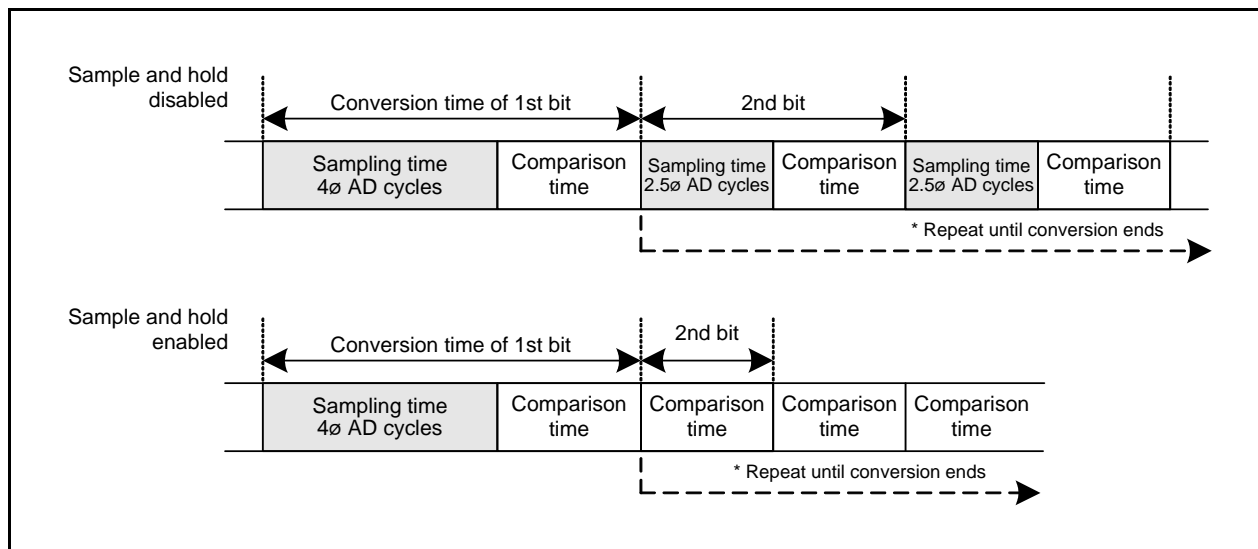


Figure 19.8 Timing Diagram of A/D Conversion

### 19.4 A/D Conversion Cycles

Figure 19.9 shows the A/D Conversion Cycles.

A/D Conversion Mode		Conversion Time	Sampling Time	Comparison Time	Sampling Time	Comparison Time	End process
Without Sample & Hold	8 bits	49φAD	4φAD	2.0φAD	2.5φAD	2.5φAD	8.0φAD
Without Sample & Hold	10 bits	59φAD	4φAD	2.0φAD	2.5φAD	2.5φAD	8.0φAD
With Sample & Hold	8 bits	28φAD	4φAD	2.5φAD	0.0φAD	2.5φAD	4.0φAD
With Sample & Hold	10 bits	33φAD	4φAD	2.5φAD	0.0φAD	2.5φAD	4.0φAD

Figure 19.9 A/D Conversion Cycles

### 19.5 Internal Equivalent Circuit of Analog Input

Figure 19.10 shows the Internal Equivalent Circuit of Analog Input.

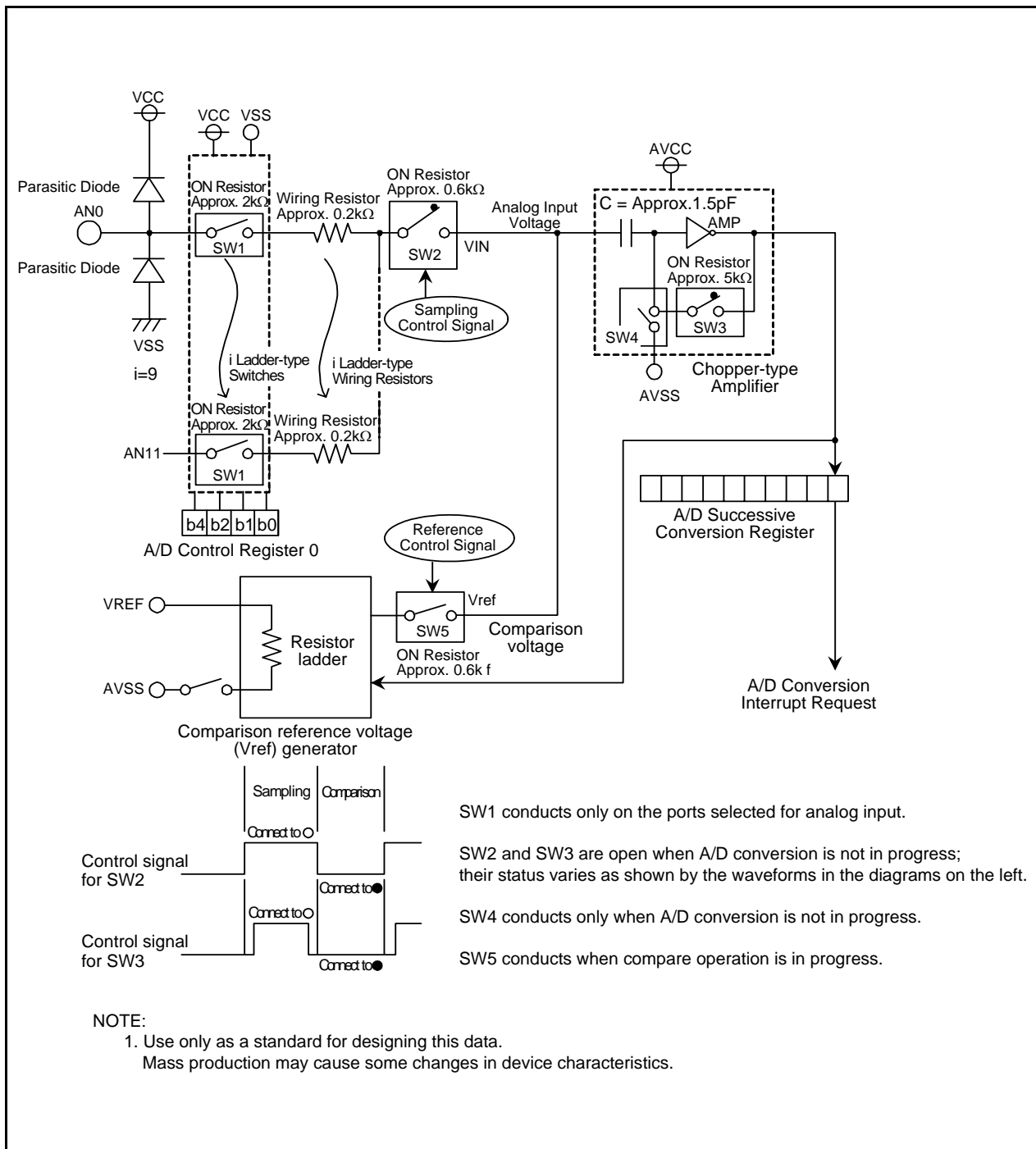


Figure 19.10 Internal Equivalent Circuit of Analog Input

## 19.6 Output Impedance of Sensor under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor  $C$  shown in Figure 19.11 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).

$$VC \text{ is generally } VC = VIN \left\{ 1 - e^{-\frac{1}{C(R_0 + R)} t} \right\}$$

$$\text{And when } t = T, \quad VC = VIN - \frac{X}{Y} VIN = VIN \left( 1 - \frac{X}{Y} \right)$$

$$e^{-\frac{1}{C(R_0 + R)} T} = \frac{X}{Y}$$

$$-\frac{1}{C(R_0 + R)} T = \ln \frac{X}{Y}$$

$$\text{Hence, } R_0 = -\frac{T}{C \cdot \ln \frac{X}{Y}} - R$$

Figure 19.11 shows the 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  $f(XIN) = 10$  MHz,  $T = 0.25 \mu\text{s}$  in the A/D conversion mode without sample and 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.

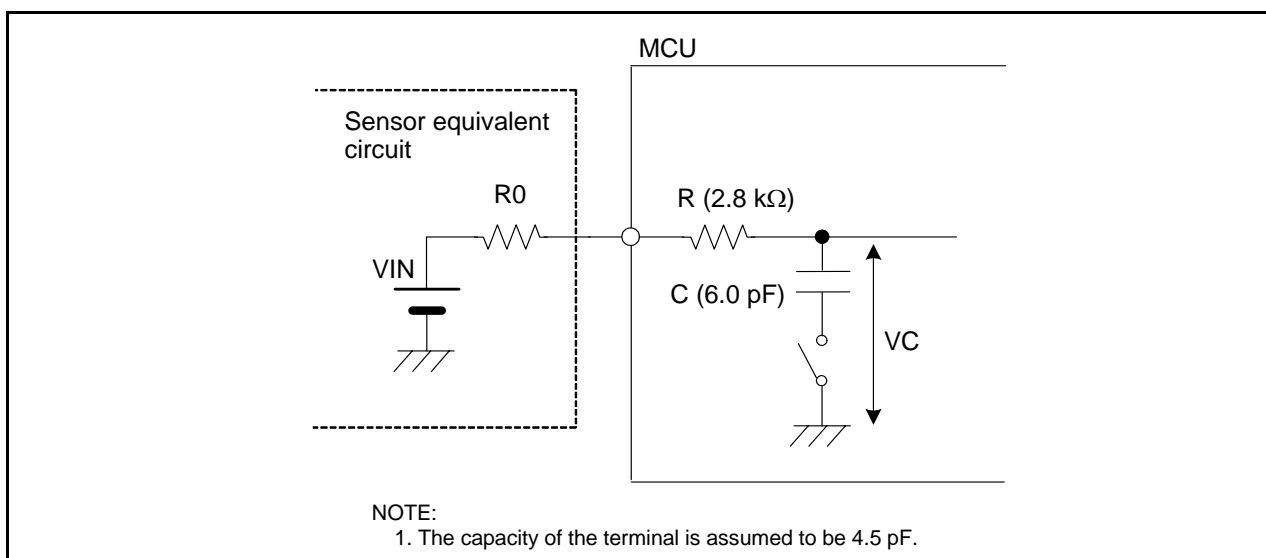


Figure 19.11 Analog Input Pin and External Sensor Equivalent Circuit

## 19.7 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 A/D conversion is stopped (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 before starting the A/D conversion.
- After changing the A/D operating mode, select an analog input pin again.
- When using the one-shot mode, ensure that A/D conversion is completed before reading the AD register. The IR bit in the ADIC register or the ADST bit in the ADCON0 register can be used to determine whether 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.
- If the ADST bit in the ADCON0 register is set to 0 (A/D conversion stops) by a program and A/D conversion is forcibly terminated during an A/D conversion operation, the conversion result of the A/D converter will be undefined. If the ADST bit is set to 0 by a program, do not use the value of the 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.

## 20. Flash Memory

### 20.1 Overview

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

Table 20.1 lists the Flash Memory Version Performance (refer to **Tables 1.1** and **1.4 Specifications** for items not listed in **Table 20.1**).

**Table 20.1 Flash Memory Version Performance**

Item		Specification
Flash memory operating modes		3 modes (CPU rewrite, standard serial I/O, and parallel I/O)
Division of erase block		Refer to <b>Figures 20.1</b> and <b>20.2</b> .
Programming method		Byte unit
Erase method		Block erase
Programming and erasure control method		Programming and erasure control by software command
Suspend functions		Program-suspend and erase-suspend
Protection method		Program ROM protection by FMR0 register
Number of commands		5 commands
Programming and erasure endurance <sup>(1)</sup>	Blocks 0 and 1 (program ROM)	R8C/2K Group: 100 times; R8C/2L Group: 1,000 times
	Blocks A and B (data flash) <sup>(2)</sup>	10,000 times
Programming and erasure voltage		VCC = 2.7 to 5.5 V
ID code check function		Standard serial I/O mode supported
ROM code protect		Parallel I/O mode supported

NOTES:

- Definition of programming and erasure endurance.  
The programming and erasure endurance is defined on a per-block basis.
- Blocks A and B (data flash) are included in the R8C/2L Group.

**Table 20.2 Flash Memory Rewrite Modes**

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

## 20.2 Memory Map

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

Figure 20.1 shows the Flash Memory Block Diagram for R8C/2K Group and Figure 20.2 shows the Flash Memory Block Diagram for R8C/2L Group.

The user ROM area contains program ROM. In addition, the R8C/2L Group has on-chip data flash.

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

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

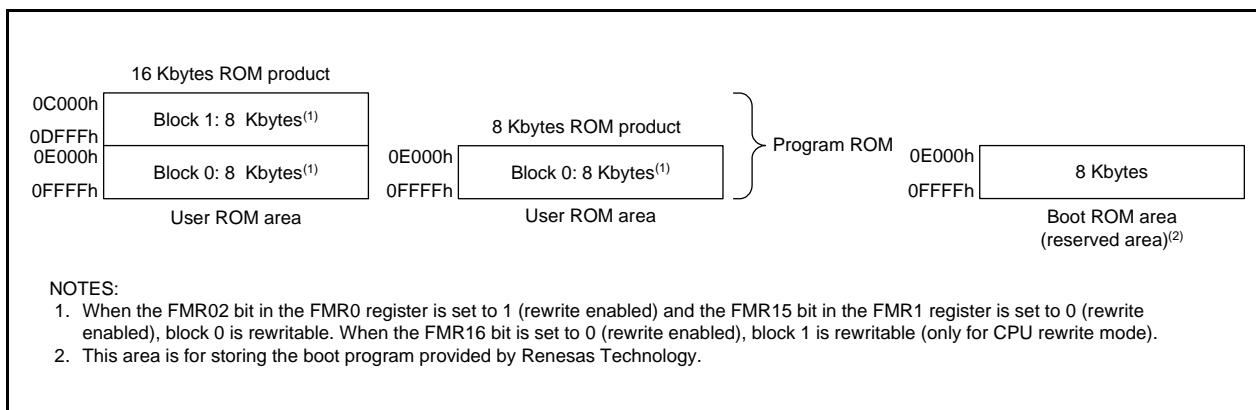


Figure 20.1 Flash Memory Block Diagram for R8C/2K Group

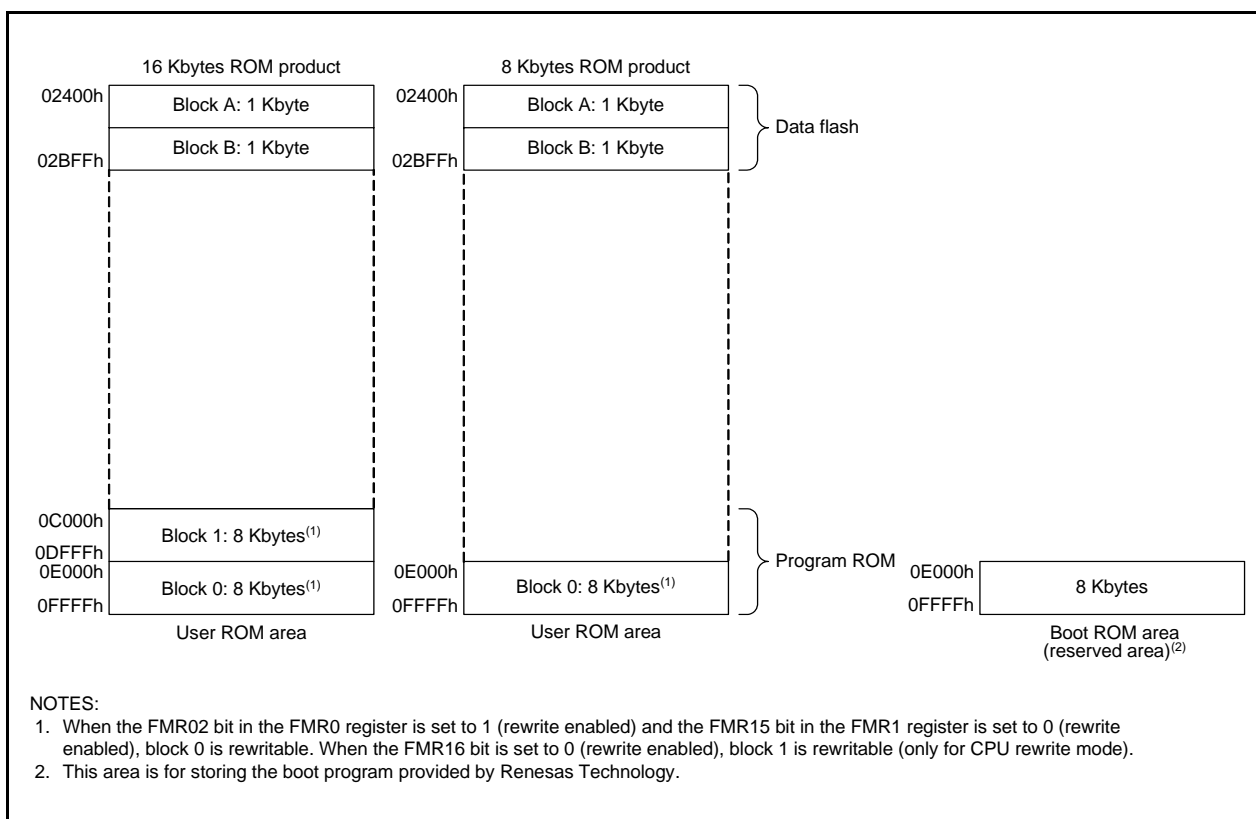


Figure 20.2 Flash Memory Block Diagram for R8C/2L Group

## 20.3 Functions to Prevent Rewriting of Flash Memory

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

### 20.3.1 ID Code Check Function

The ID code check function is used in standard serial I/O mode. Unless 3 bytes (addresses from 0FFFCh to 0FFFEh) of the reset vector are set to FFFFFFFh, the ID codes sent from the serial programmer or the on-chip debugging emulator and the 7-byte ID codes written in the flash memory are checked to see if they match. If the ID codes do not match, the commands sent from the serial programmer or the on-chip debugging emulator are not acknowledged. For details of the ID code check function, refer to **13. ID Code Areas**.

### 20.3.2 ROM Code Protect Function

The ROM protect function prevents the contents of the flash memory from being read, rewritten, or erased by means of the OFS register when parallel I/O mode is used.

Figure 20.3 shows the OFS Register. Refer to **14. Option Function Select Area** for details of the OFS register.

The ROM code protect function is enabled by writing 0 to the ROMCP1 bit and 1 to the ROMCR bit. It disables reading or changing the contents of the on-chip flash memory.

Once ROM code protect is enabled, the content in the internal flash memory cannot be rewritten in parallel I/O mode. To disable ROM code protect, erase the block including the OFS register with CPU rewrite mode or standard serial I/O mode.

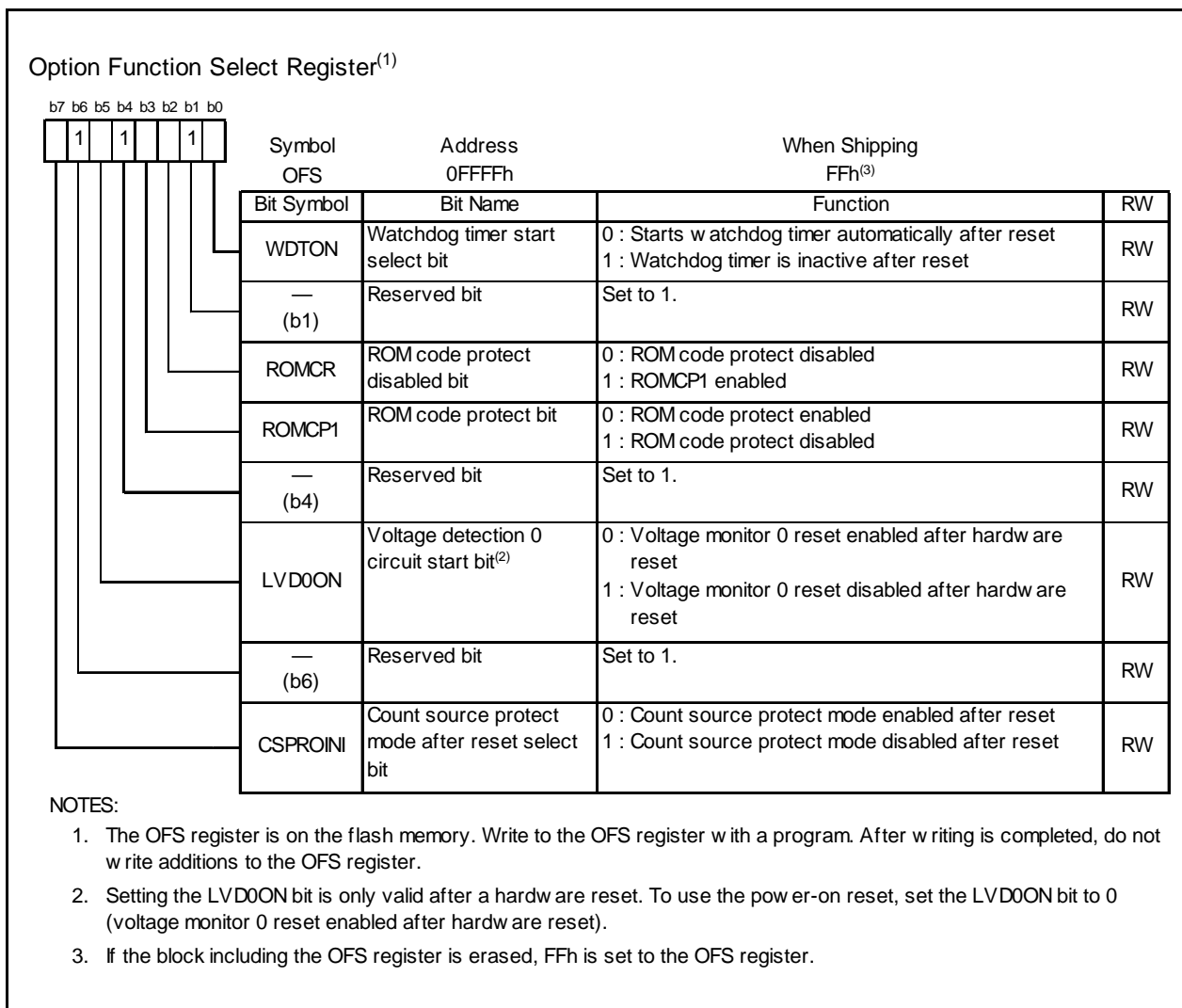


Figure 20.3 OFS Register



## 20.4 CPU Rewrite Mode

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

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

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

CPU rewrite mode has an erase write 0 mode (EW0 mode) and an erase write 1 mode (EW1 mode). Table 20.3 lists the Differences between EW0 Mode and EW1 Mode.

**Table 20.3 Differences between EW0 Mode and EW1 Mode**

Item	EW0 Mode	EW1 Mode
Operating mode	Single-chip mode	Single-chip mode
Areas in which a rewrite control program can be executed	RAM (Rewrite control program is executed after being transferred)	User ROM or RAM
Rewritable areas	User ROM	User ROM However, blocks which contain a rewrite control program are excluded
Software command restrictions	None	<ul style="list-style-type: none"> <li>• Program and block erase commands Cannot be run on any block which contains a rewrite control program</li> <li>• Read status register command Cannot be executed</li> </ul>
Modes after program or erase	Read status register mode	Read array mode
Modes after read status register	Read status register mode	Do not execute this command
CPU status during auto-write and auto-erase	Operating	Hold state (I/O ports hold state before the command is executed)
Flash memory status detection	<ul style="list-style-type: none"> <li>• Read bits FMR00, FMR06, and FMR07 in the FMR0 register by a program</li> <li>• Execute the read status register command and read bits SR7, SR5, and SR4 in the status register.</li> </ul>	Read bits FMR00, FMR06, and FMR07 in the FMR0 register by a program
Conditions for transition to erase-suspend	Set bits FMR40 and FMR41 in the FMR4 register to 1 by a program.	The FMR40 bit in the FMR4 register is set to 1 and the interrupt request of the enabled maskable interrupt is generated
Conditions for transitions to program-suspend	Set bits FMR40 and FMR42 in the FMR4 register to 1 by a program.	The FMR40 bit in the FMR4 register is set to 1 and the interrupt request of the enabled maskable interrupt is generated
CPU clock	5 MHz or below	No restriction (on clock frequency to be used)

## 20.4.1 Register Description

The registers used in CPU rewrite mode are described.

### 20.4.1.1 FMR0 Register (FMR0)

Figure 20.4 shows the FMR0 Register.

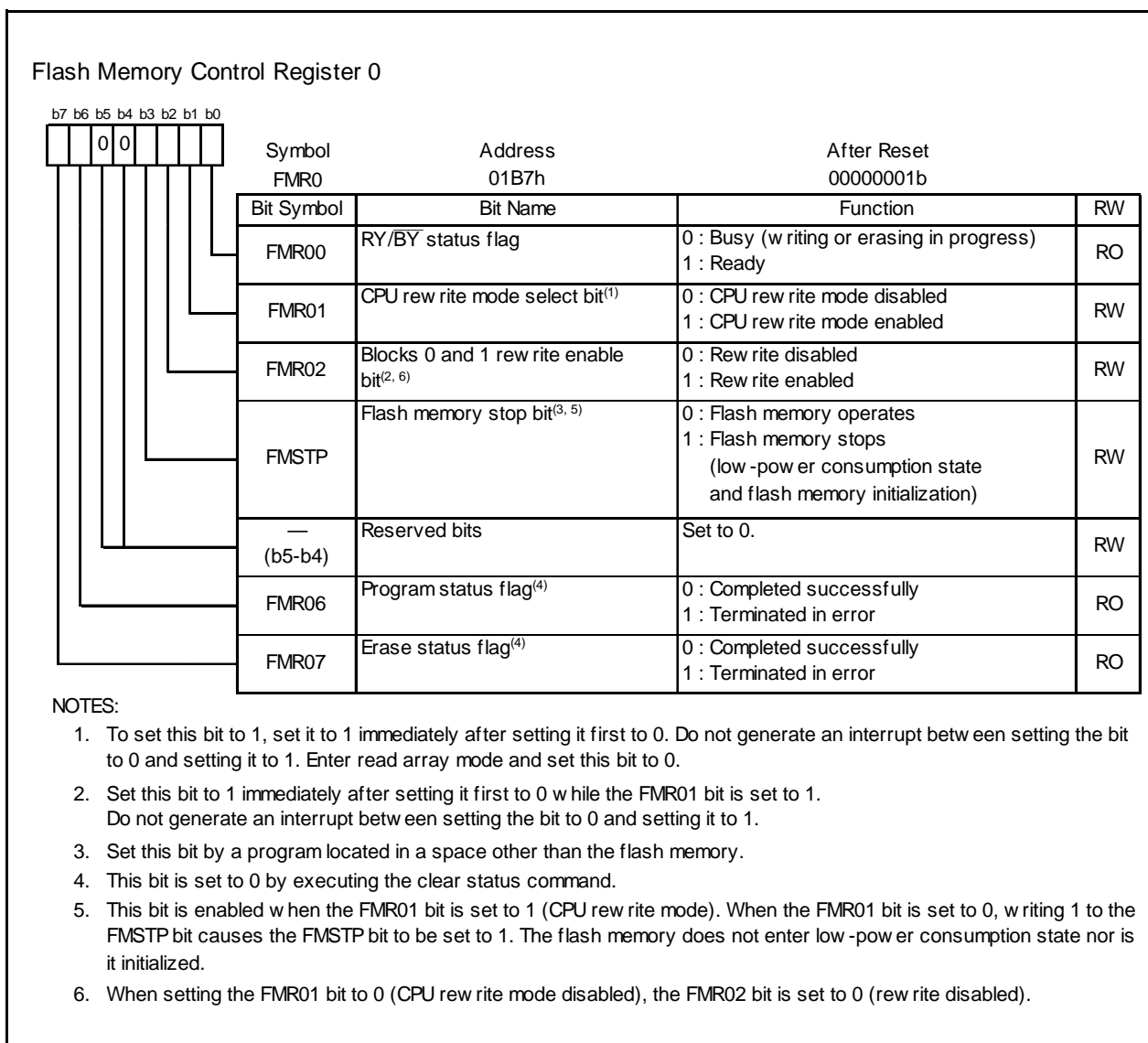


Figure 20.4 FMR0 Register

- FMR00 Bit

This bit indicates the operating status of the flash memory. The bit's value is 0 during programming, erasure (including suspend periods), or erase-suspend mode; otherwise, it is 1.

- FMR01 Bit

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

- FMR02 Bit

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

Rewriting of blocks 0 and 1 is controlled by bits FMR15 and FMR16 if the FMR02 bit is set to 1 (rewrite enabled).

- FMSTP Bit

This bit is used to initialize the flash memory control circuits, and also to reduce the amount of current consumed by the flash memory. Access to the flash memory is disabled by setting the FMSTP bit to 1. Therefore, the FMSTP bit must be written to by a program transferred to the RAM.

In the following cases, set the FMSTP bit to 1:

- When flash memory access resulted in an error while erasing or programming in EW0 mode (FMR00 bit not initialized to 1 (ready))
- To provide lower consumption in low-speed on-chip oscillator mode and low-speed clock mode.

Note that when going to stop or wait mode while the CPU rewrite mode is disabled, the FMR0 register does not need to be set because the power for the flash memory is automatically turned off and is turned back on again after returning from stop or wait mode.

- FMR06 Bit

This is a read-only bit indicating the status of an auto-program operation. The bit is set to 1 when a program error occurs; otherwise, it is set to 0. For details, refer to the description in **Table 20.4 Errors and FMR0 Register Status**.

- FMR07 Bit

This is a read-only bit indicating the status of an auto-erase operation. The bit is set to 1 when an erase error occurs; otherwise, it is set to 0. Refer to **Table 20.4 Errors and FMR0 Register Status** for details.

**Table 20.4 Errors and FMR0 Register Status**

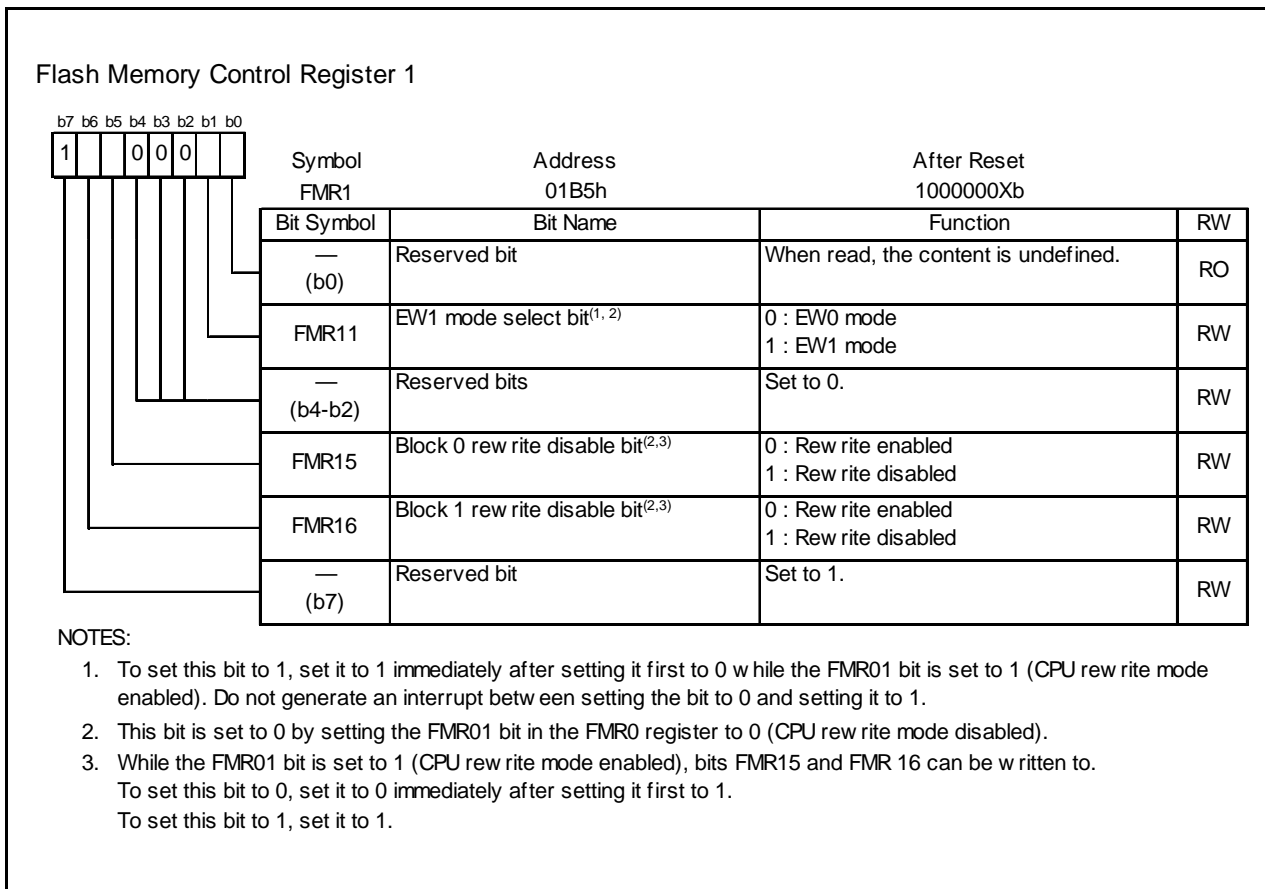
FRM0 Register (Status Register) Status		Error	Error Occurrence Condition
FMR07 (SR5)	FMR06 (SR4)		
1	1	Command sequence error	<ul style="list-style-type: none"> <li>• When a command is not written correctly.</li> <li>• When D0h or FFh is not written in the 2nd byte of the block erase command.<sup>(1)</sup></li> <li>• When the program command or block erase command is executed while rewriting is disabled by the FMR02 bit in the FMR0 register, or the FMR15 bit in the FMR1 register.</li> <li>• When an address not allocated in flash memory is input during erase command input</li> <li>• When attempting to erase the block for which rewriting is disabled during erase command input.</li> <li>• When an address not allocated in flash memory is input during write command input.</li> <li>• When attempting to write to a block for which rewriting is disabled during write command input.</li> </ul>
1	0	Erase error	<ul style="list-style-type: none"> <li>• When the block erase command is executed but auto-erasure does not complete correctly</li> </ul>
0	1	Program error	<ul style="list-style-type: none"> <li>• When the program command is executed but not auto-programming does not complete.</li> </ul>
0	0	Completed successfully	–

**NOTE:**

1. When FFh is written in the 2nd byte of the block erase command, the MCU enters read array mode, and the command code written in the 1st byte is disabled.

### 20.4.1.2 FMR1 Register (FMR1)

Figure 20.5 shows the FMR1 Register.



**Figure 20.5 FMR1 Register**

- **FMR11 Bit**  
Setting this bit to 1 (EW1 mode) places the MCU in EW1 mode.
- **FMR15 Bit**  
When the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit is set to 0 (rewrite enabled), block 0 accepts program and block erase commands.
- **FMR16 Bit**  
When the FMR02 bit is set to 1 (rewrite enabled) and the FMR16 bit is set to 0 (rewrite enabled), block 1 accepts program and block erase commands.

### 20.4.1.3 FMR4 Register (FMR4)

Figure 20.6 shows the FMR4 Register.

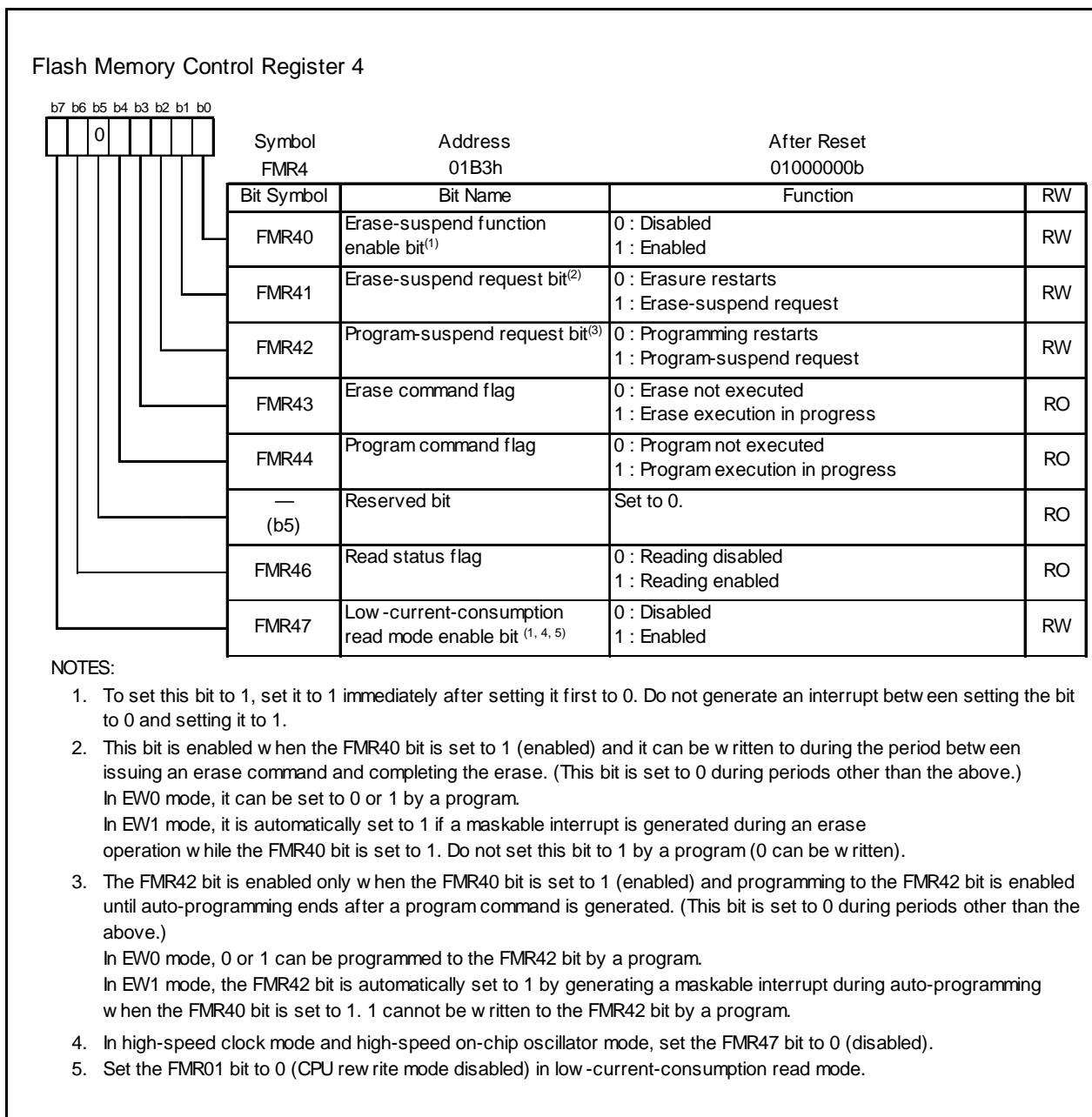


Figure 20.6 FMR4 Register

- **FMR40 Bit**  
The suspend function is enabled by setting the FMR40 bit to 1 (enabled).
- **FMR41 Bit**  
In EW0 mode, the MCU enters erase-suspend mode when the FMR41 bit is set to 1 by a program. The FMR41 bit is automatically set to 1 (erase-suspend request) 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 (erasure restarts) when the auto-erase operation restarts.
- **FMR42 Bit**  
In EW0 mode, the MCU enters program-suspend mode when the FMR42 bit is set to 1 by a program. The FMR42 bit is automatically set to 1 (program-suspend request) 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 (programming restarts) when the auto-program operation restarts.
- **FMR43 Bit**  
When the auto-erase operation starts, the FMR43 bit is set to 1 (erase execution in progress). The FMR43 bit remains set to 1 (erase execution in progress) during erase-suspend operation.  
When the auto-erase operation ends, the FMR43 bit is set to 0 (erase not executed).
- **FMR44 Bit**  
When the auto-program operation starts, the FMR44 bit is set to 1 (program execution in progress). The FMR44 bit remains set to 1 (program execution in progress) during program-suspend operation.  
When the auto-program operation ends, the FMR44 bit is set to 0 (program not executed).
- **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.
- **FMR47 Bit**  
Current consumption when reading the flash memory can be reduced by setting the FMR47 bit to 1 (enabled) in low-speed clock mode and low-speed on-chip oscillator mode.  
Refer to **21.2.10 Low-Current-Consumption Read Mode** for details of the handling procedure.

### 20.4.2 Status Check Procedure

When an error occurs, bits FMR06 to FMR07 in the FMR0 register are set to 1, indicating the occurrence of an error. Therefore, checking these status bits (full status check) can be used to determine the execution result. Figure 20.7 shows the Full Status Check and Handling Procedure for Individual Errors.

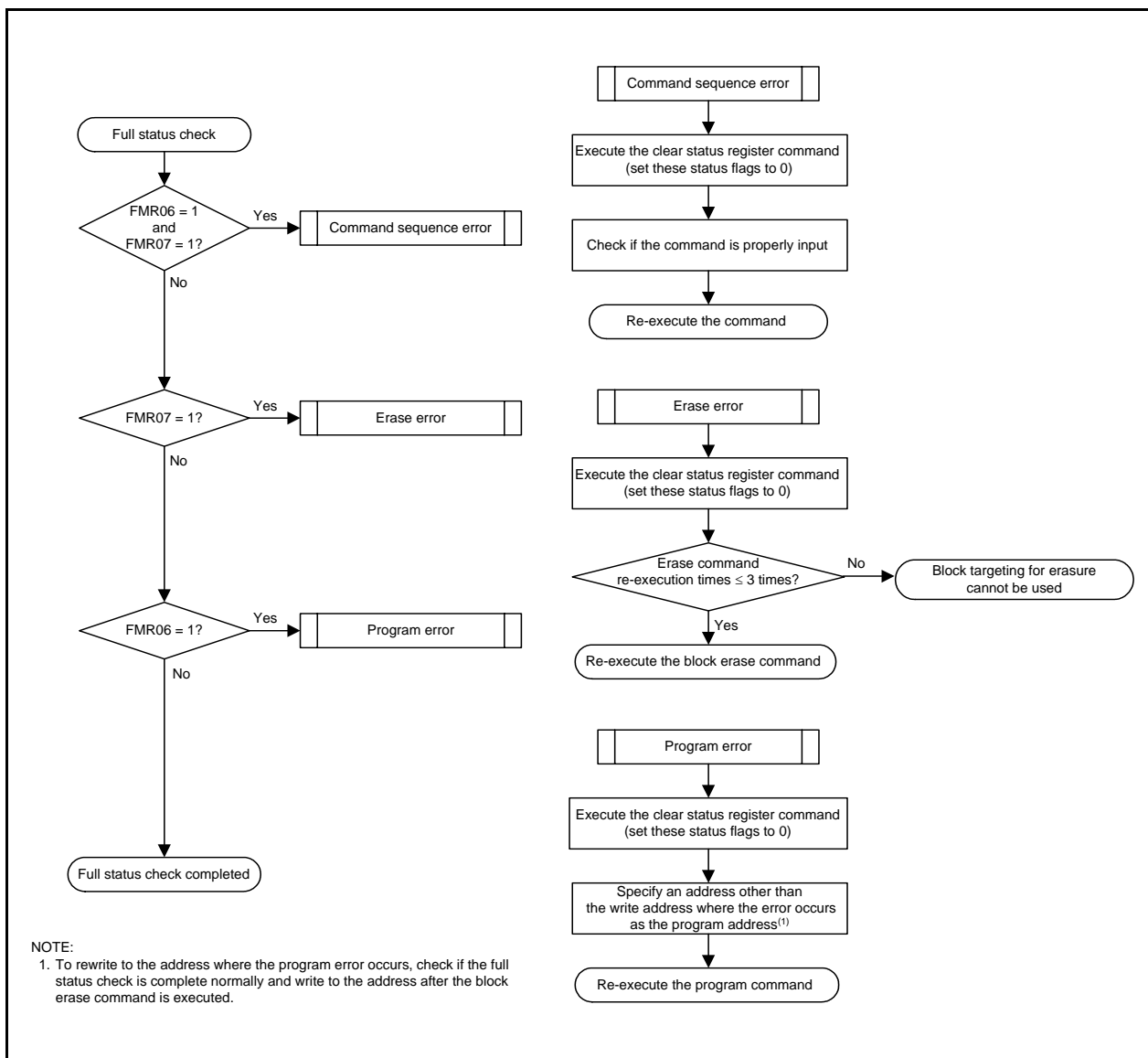


Figure 20.7 Full Status Check and Handling Procedure for Individual Errors

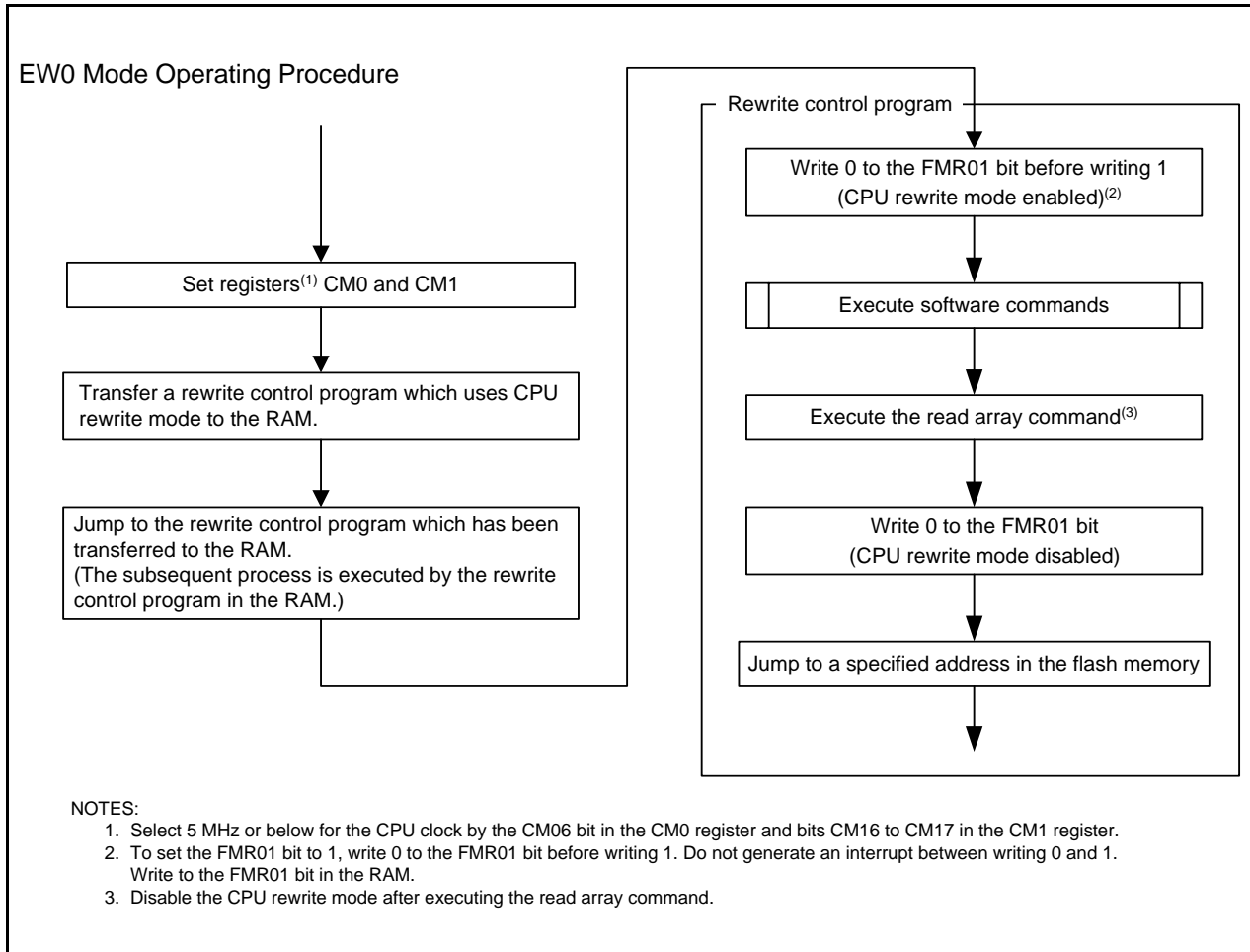


### 20.4.3 EW0 Mode

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

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

Figure 20.8 shows the How to Set and Exit EW0 Mode.



**Figure 20.8 How to Set and Exit EW0 Mode**

### 20.4.3.1 Software Commands

There are five types of software commands:

- Read array
- Read status register
- Clear status register
- Program
- Block erase

Figure 20.9 shows the Software Command Status Transition Diagram in EW0 Mode.

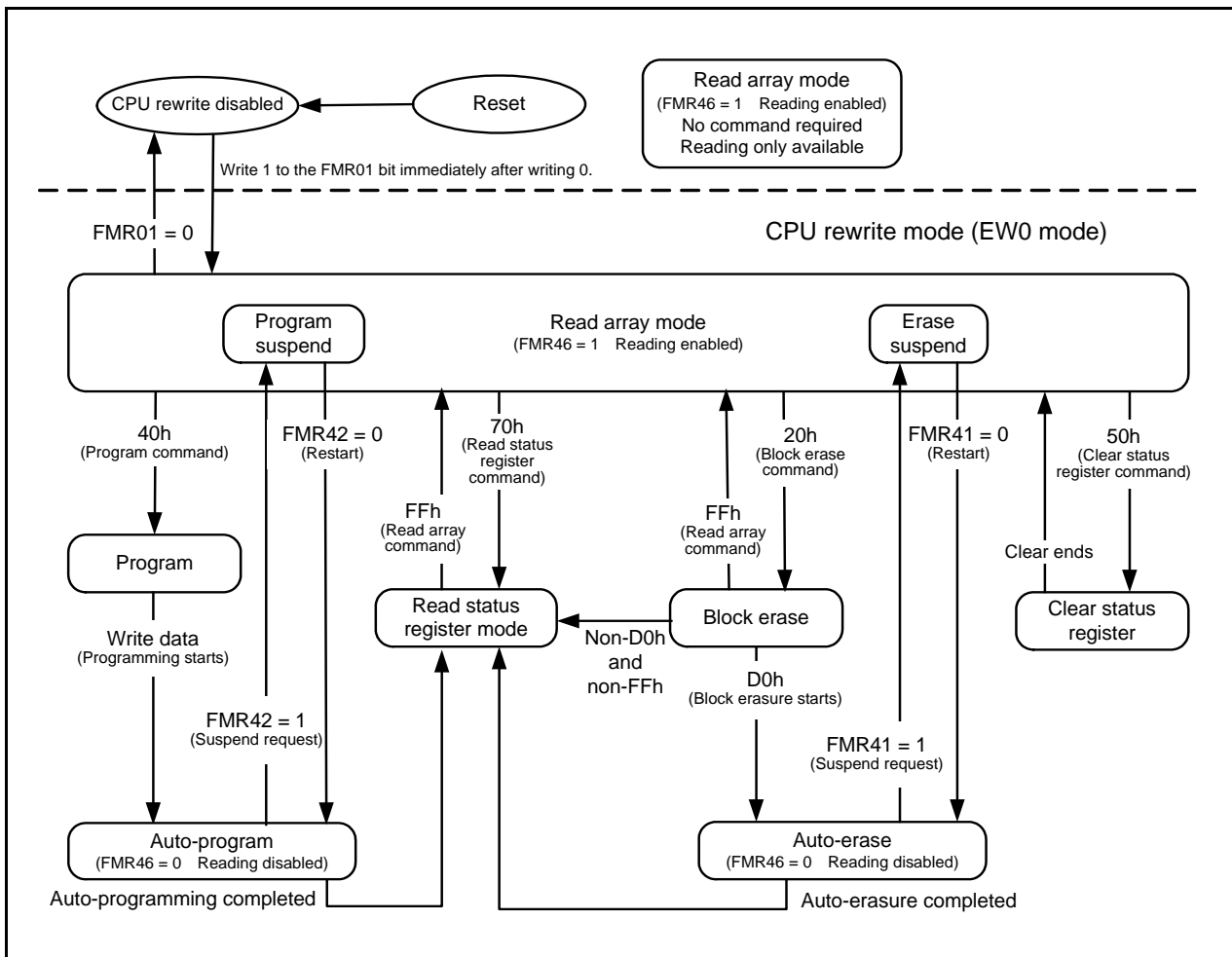


Figure 20.9 Software Command Status Transition Diagram in EW0 Mode

- Read Array Command

The read array command reads the flash memory.

When FFh is written to an address in the user ROM area, the MCU enters read array mode. In this mode, the contents of the specified address can be read.

Read array mode continues until other commands are written. The MCU enters this mode after a reset is deasserted.

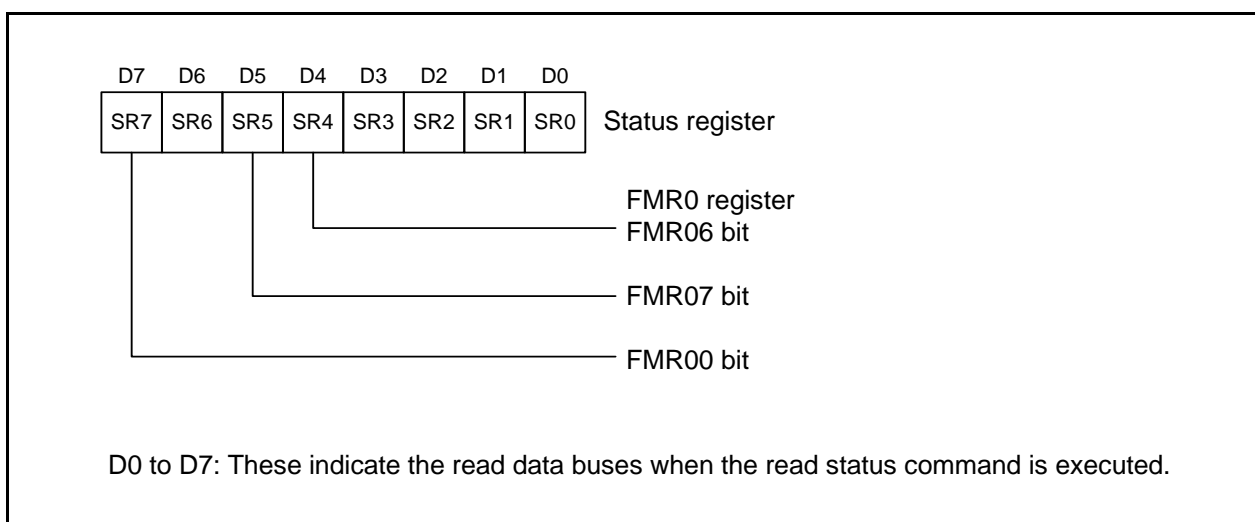
- Read Status Register Command

The read status register command is used to read the status register. Figure 20.10 shows the Status Register.

The status register indicates the operating status of the flash memory and whether an erase or program operation has completed normally or in error (refer to **Table 20.4 Errors and FMR0 Register Status**). When 70h is written to an address in the user ROM area, the MCU enters read status register mode. When the address in the user ROM area is read subsequently, the status register can be read.

The MCU remains in read status register mode until the next read array command is written.

The status of the status register can be determined by reading bits FMR00, FMR06, and FMR07 in the FMR0 register.



**Figure 20.10 Status Register**

- Clear Status Register Command

The clear status register command sets the status register to 0.

When 50h is written to an address in the user ROM area, bits FMR07 and FMR06 in the FMR0 register and bits SR5 and SR4 in the status register are set to 00b.

- Program Command

The program command writes data to the flash memory in 1-byte units.

When 40h is written and then data is written to the write address, an auto-program operation (data program and verify) starts.

The FMR00 bit in the FMR0 register can be used to determine whether auto-programming has completed. 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.

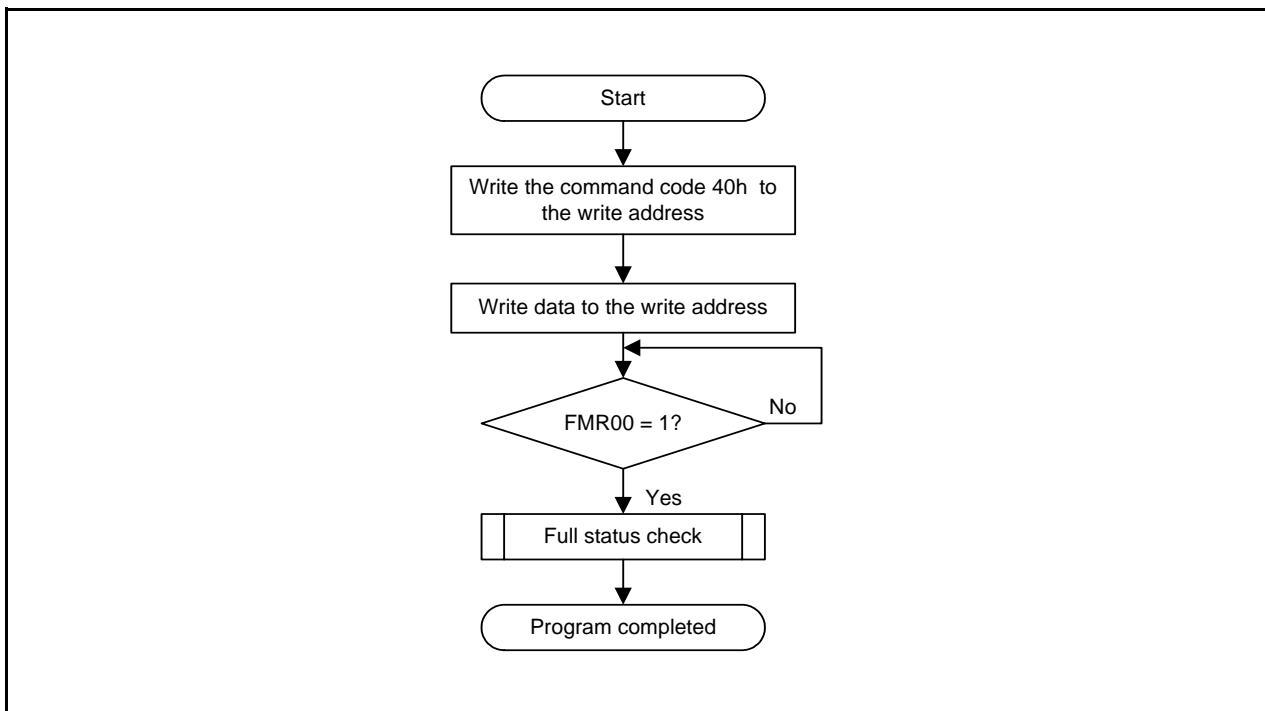
The FMR06 bit in the FMR0 register can be used to determine the result of auto-programming after it has been finished (refer to **20.4.2 Status Check Procedure**).

Do not write additions to the already programmed addresses.

Also, when the FMR02 bit in the FMR0 register is set to 0 (rewrite disabled), or the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register is set to 1 (rewrite disabled), program commands targeting block 0 are not acknowledged.

**Figure 20.11** shows the Program Command in EW0 Mode (When Suspend Function Disabled). **Figure 20.12** shows the Program Command in EW0 Mode (When Suspend Function Enabled).

In EW0 mode, the MCU enters read status register mode at the same time auto-programming starts and the status register can be read. In this case, the MCU remains in read status register mode until the next read array command is written.



**Figure 20.11 Program Command in EW0 Mode (When Suspend Function Disabled)**

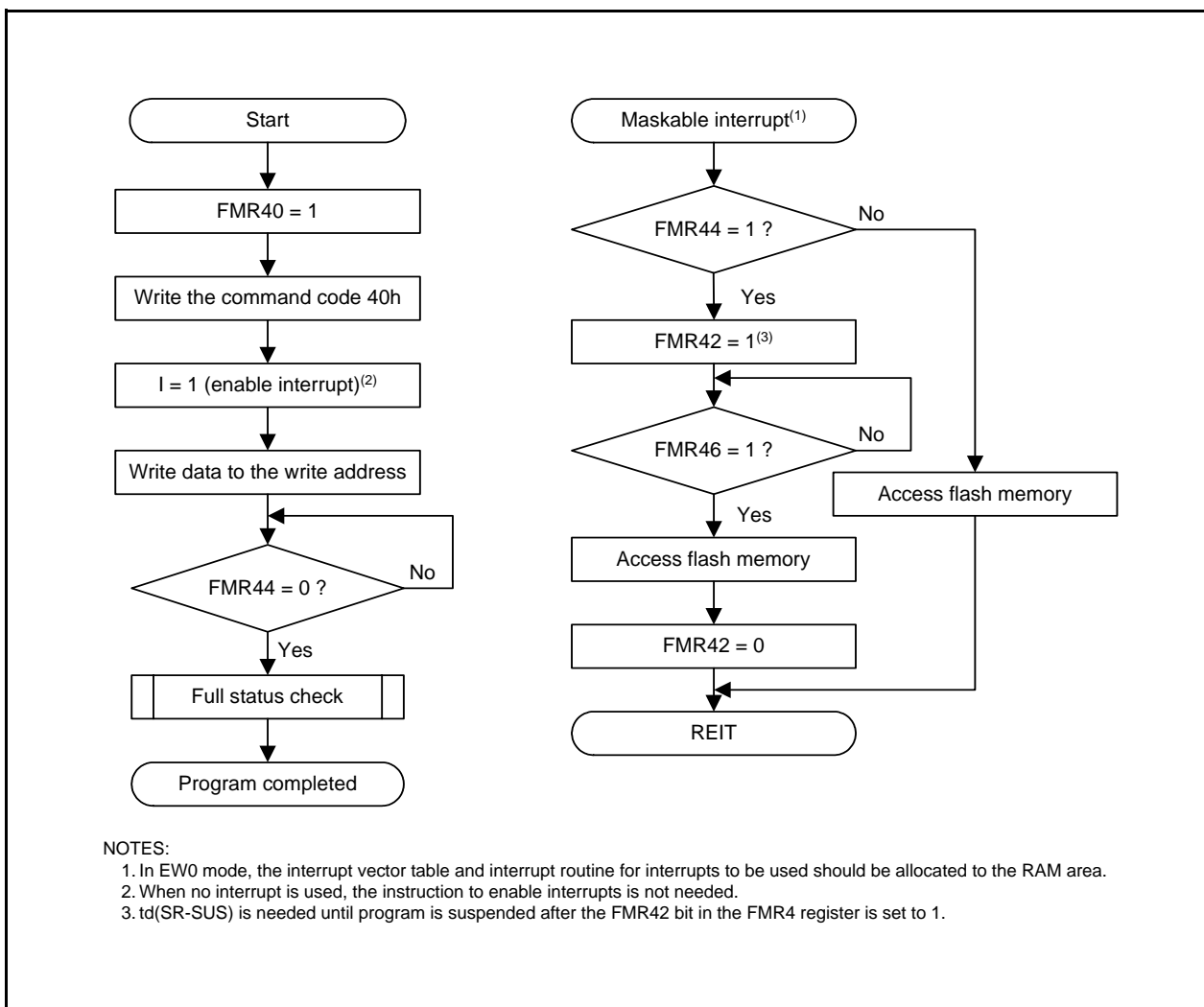


Figure 20.12 Program Command in EW0 Mode (When Suspend Function Enabled)

- Block Erase

When 20h is first written and then D0h is written to a given block address, an auto-erase operation (erase and verify) of the specified block starts.

The FMR00 bit in the FMR0 register can be used to determine whether auto-erase has completed.

The FMR00 bit is set to 0 during auto-erase and set to 1 when auto-erase completes.

The FMR07 bit in the FMR0 register can be used to determine the result of auto-erase after auto-erase has completed (refer to **20.4.2 Status Check Procedure**).

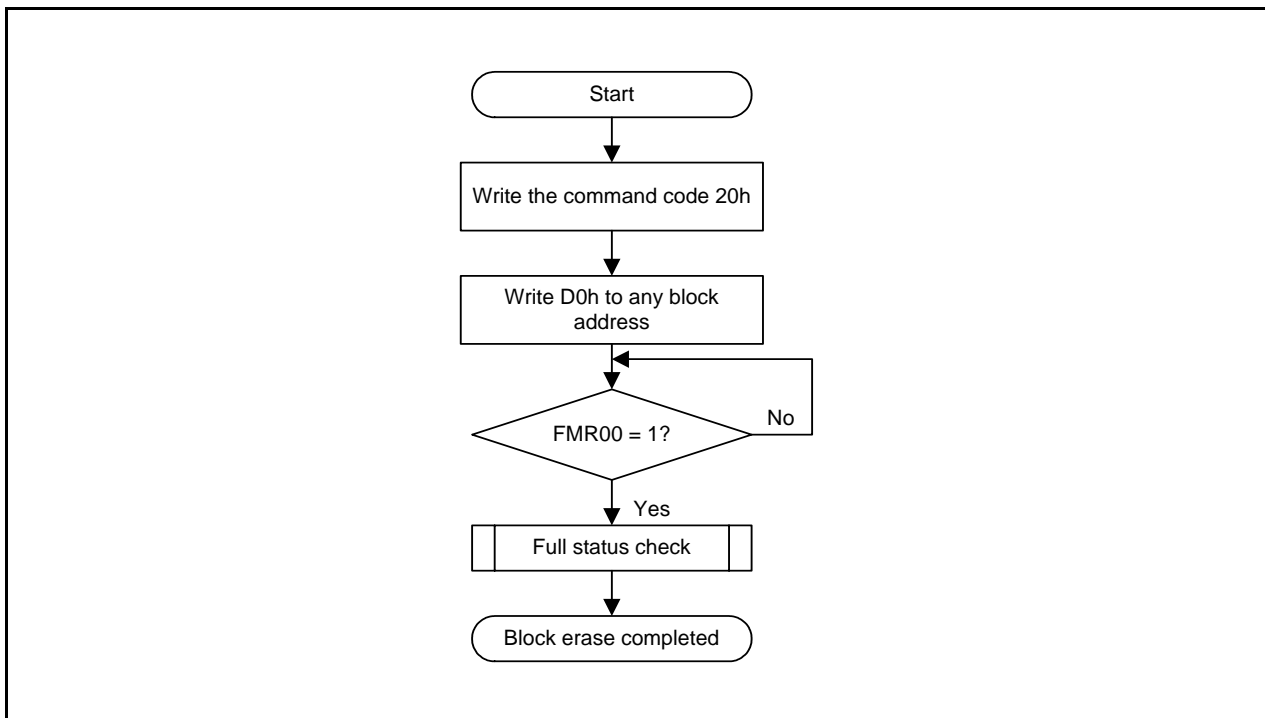
Also, when the FMR02 bit in the FMR0 register is set to 0 (rewrite disabled), or the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register is set to 1 (rewrite disabled), block erase commands targeting block 0 are not acknowledged.

Do not use the block erase command during program-suspend.

In EW0 mode, the MCU enters read status register mode at the same time auto-erase starts and the status register can be read. In this case, the MCU remains in read status register mode until the next read array command is written.

**Figure 20.13** shows the Block Erase Command in EW0 Mode (When Suspend Function Disabled). **Figure 20.14** shows the Block Erase Command in EW0 Mode (When Suspend Function Enabled).

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



**Figure 20.13** Block Erase Command in EW0 Mode (When Suspend Function Disabled)

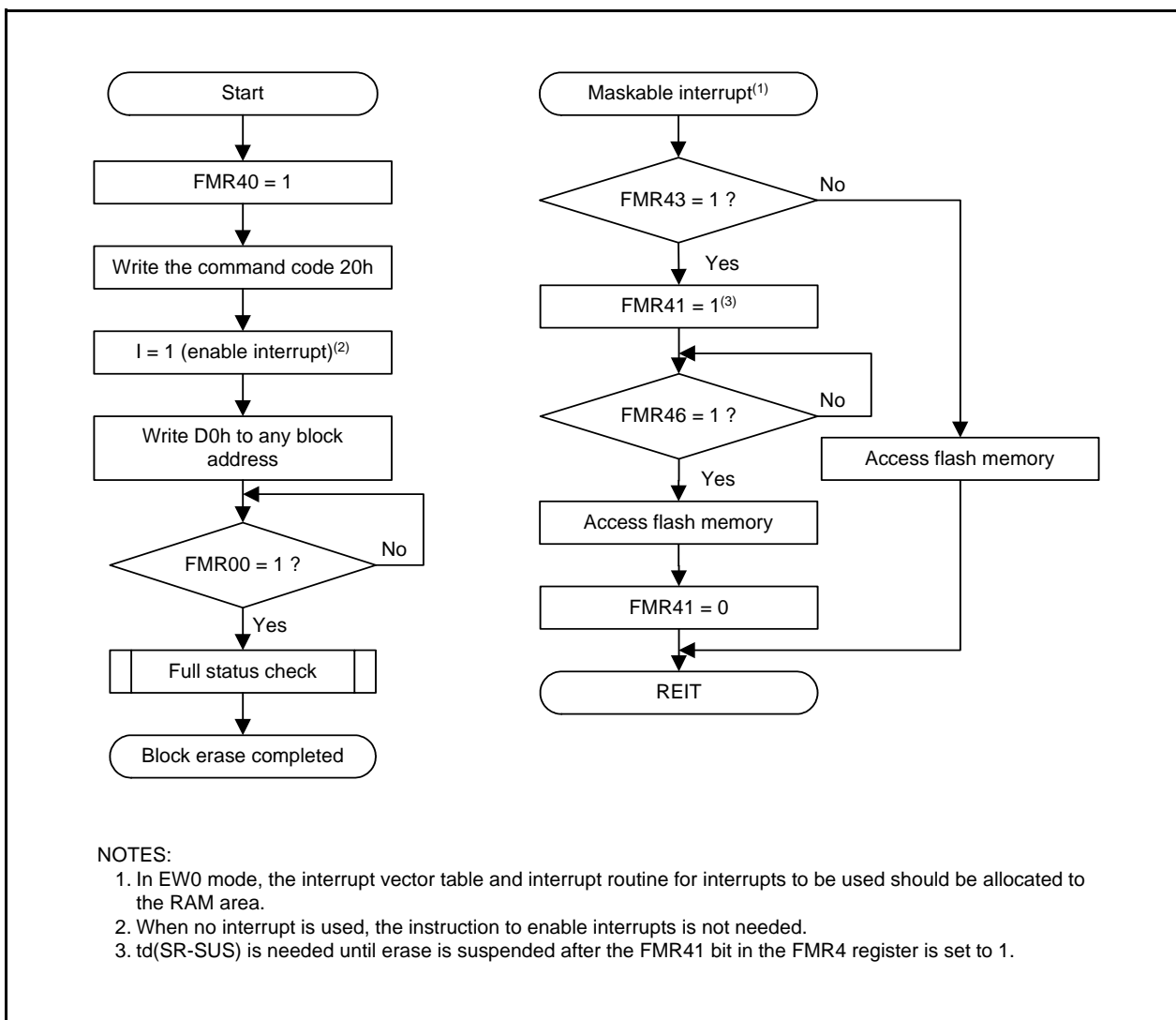


Figure 20.14 Block Erase Command in EW0 Mode (When Suspend Function Enabled)

### 20.4.3.2 Suspend Function

The suspend function halts auto-erase and auto-programming temporarily while these operations are in progress. This function is used for interrupt handling as the user ROM area can be read after the above operations have been suspended.

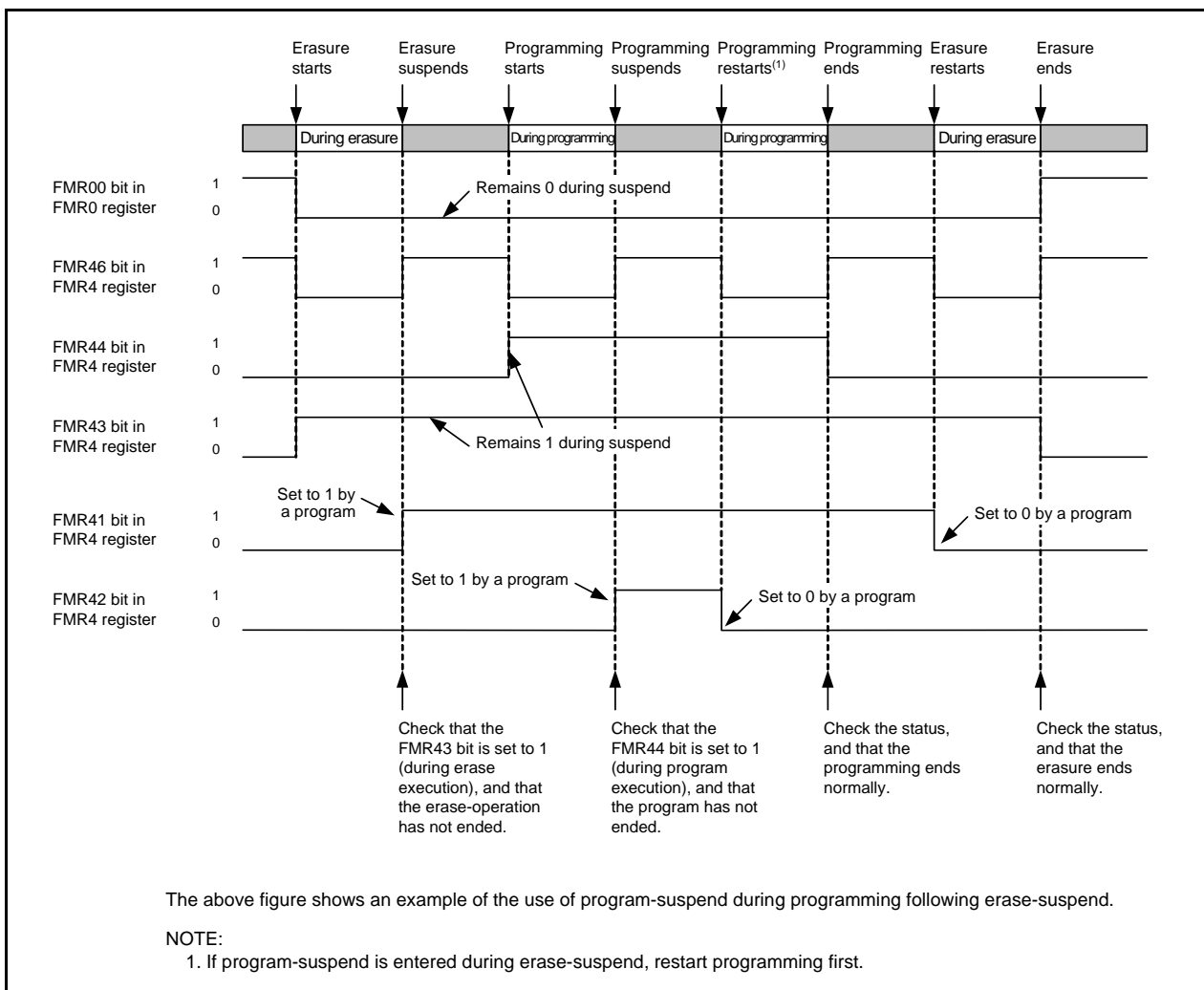
When using erase-suspend or program-suspend in EW0 mode, first check the status of the flash memory in the interrupt routine and then enter erase-suspend or program-suspend. **Figure 20.15** shows the Timing of Suspend Operation in EW0 Mode.

The procedure for entering erase-suspend during auto-erase operation is as follows:

- (1) Set the FMR40 bit to 1 (suspend enabled)
- (2) Set the FMR41 bit to 1 (erase-suspend request).
- (3) Wait for  $t_d$  (SR-SUS).
- (4) Confirm that the FMR46 bit is set to 1 (reading enabled)
- (5) Access the user ROM area.
- (6) When the FMR41 bit is set to 0 (erasure restarts), auto-erase operation restarts.

The procedure for entering program-suspend during auto-programming operation is as follows:

- (1) Set the FMR40 bit to 1 (suspend enabled)
- (2) Set the FMR42 bit to 1 (program-suspend request).
- (3) Wait for  $t_d$  (SR-SUS).
- (4) Confirm that the FMR46 bit is set to 1 (reading enabled)
- (5) Access the user ROM area.
- (6) When the FMR42 bit is set to 0 (programming restarts), auto-programming operation restarts.



**Figure 20.15** Timing of Suspend Operation in EW0 Mode



Figure 20.16 shows the Program Flowchart during Erase-Suspend in EW0 Mode.

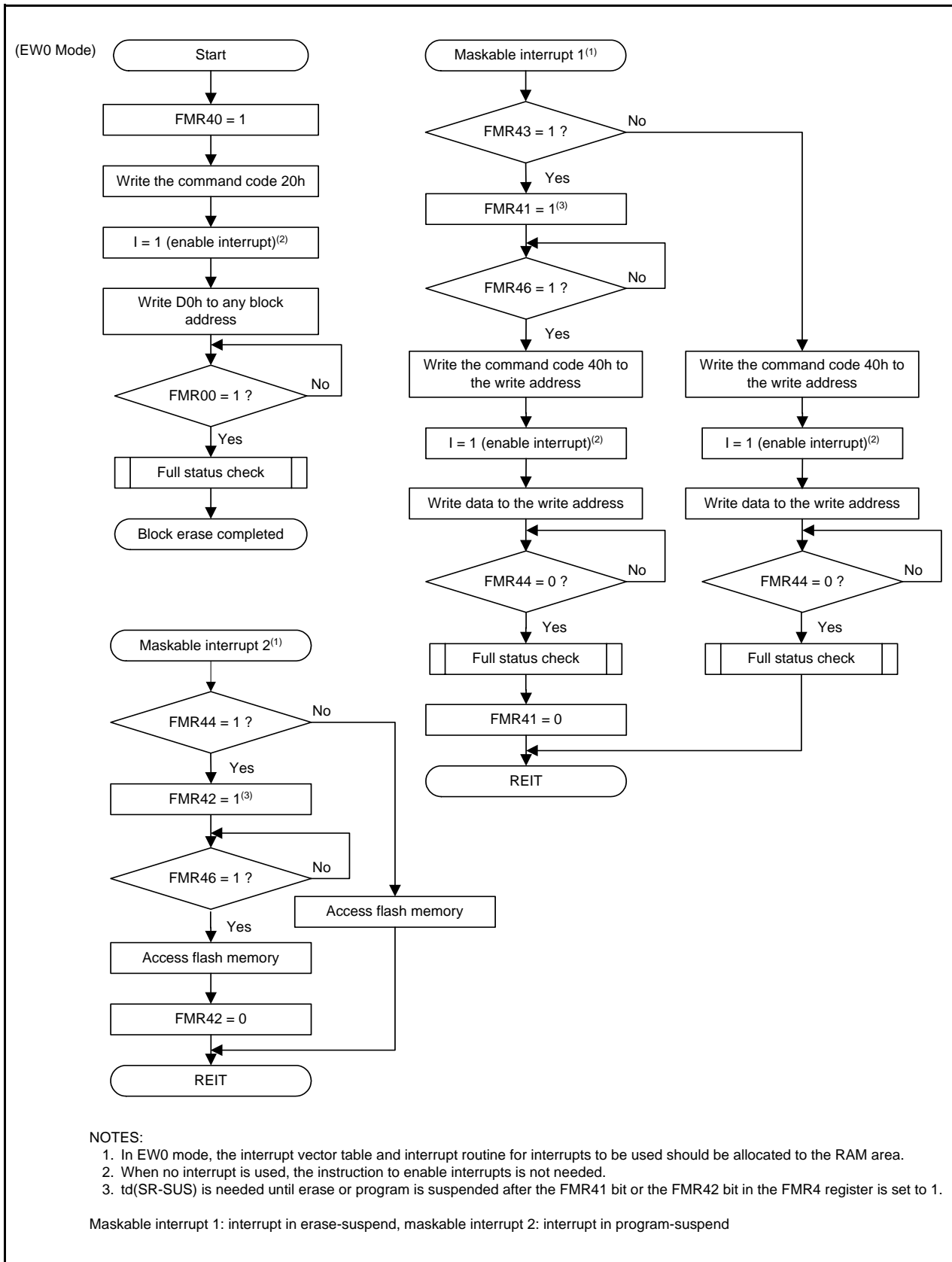


Figure 20.16 Program Flowchart during Erase-Suspend in EW0 Mode

### 20.4.3.3 EW0 Mode Interrupts

In EW0 mode, maskable interrupts can be used by allocating a vector in RAM. Table 20.5 lists the EW0 Mode Interrupts. Refer to **20.7.1.3 Non-Maskable Interrupts** for details of the non-maskable interrupt.

**Table 20.5 EW0 Mode Interrupts**

Status	When Maskable Interrupt Request is Acknowledged
During auto-erasure	Interrupt handling is executed.
Auto-programming	

#### 20.4.4 EW1 Mode

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

The FMR0 register can be used to determine when program and erase operations complete. Figure 20.17 shows the How to Set and Exit EW1 Mode.

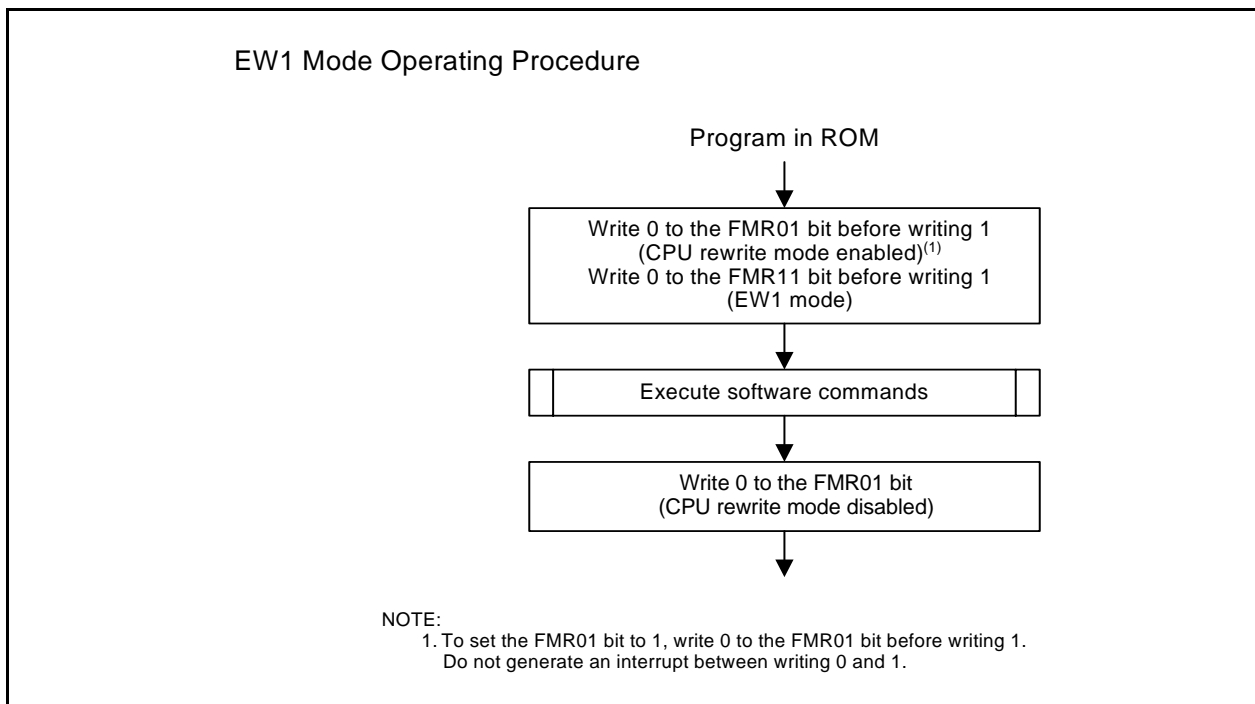


Figure 20.17 How to Set and Exit EW1 Mode

### 20.4.4.1 Software Commands

There are four types of software commands:

- Read array
- Clear status register
- Program
- Block erase

Do not execute read status register command in EW1 mode.

Figure 20.18 shows the Software Command Status Transition Diagram in EW1 Mode.

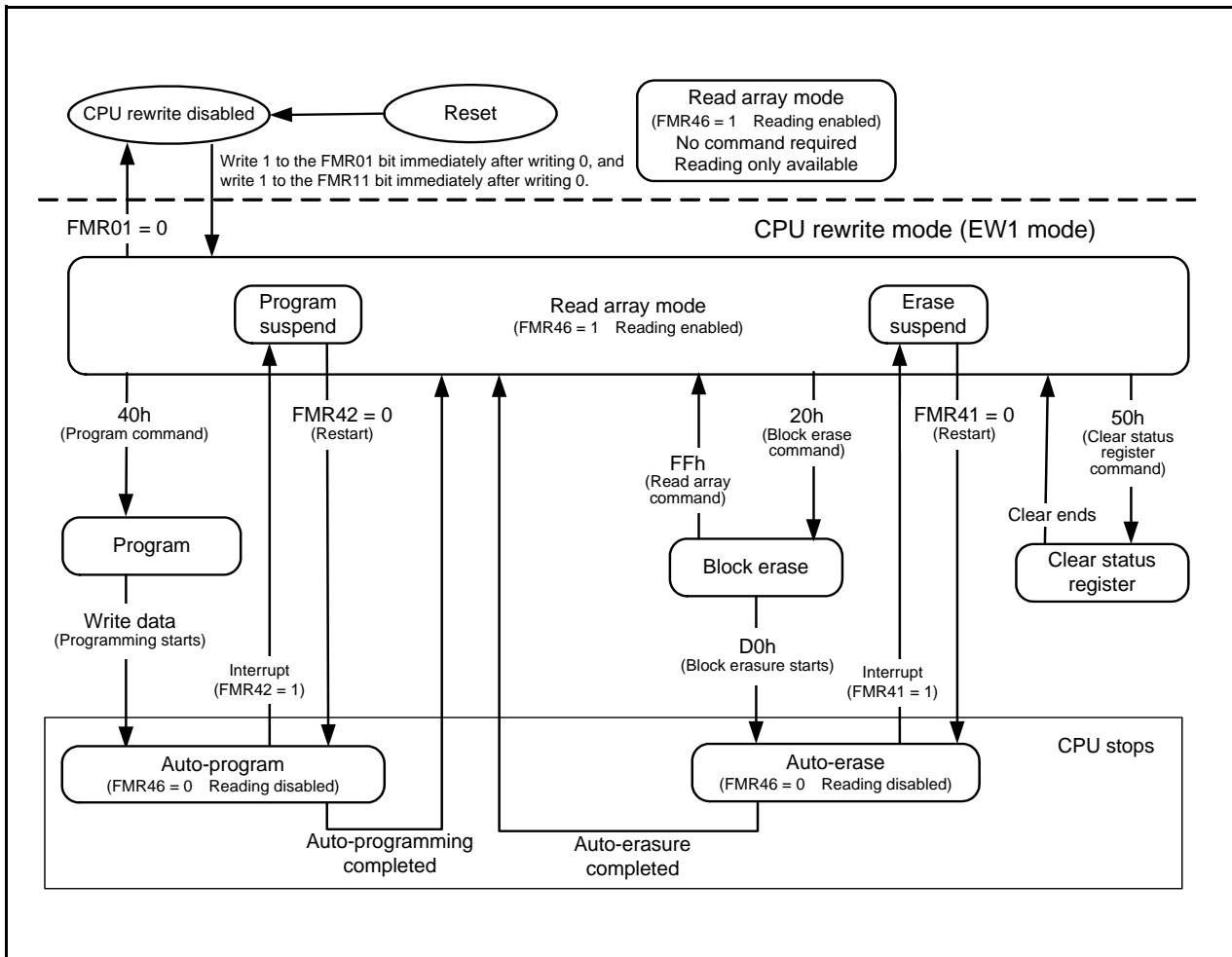


Figure 20.18 Software Command Status Transition Diagram in EW1 Mode

- Read Array Command

The read array command reads the flash memory.

When FFh is written to an address in the user ROM area, the MCU enters read array mode. In this mode, the contents of the specified address can be read.

Read array mode continues until other commands are written. The MCU enters this mode after a reset is deasserted.

- Clear Status Register Command

The clear status register command sets the status register to 0.

When 50h is written to an address in the user ROM area, bits FMR07 and FMR06 in the FMR0 register and bits SR5 and SR4 in the status register are set to 00b.

- Program Command

The program command writes data to the flash memory in 1-byte units.

When 40h is written and then data is written to the write address, an auto-program operation (data program and verify) starts.

The FMR00 bit in the FMR0 register can be used to determine whether auto-programming has completed. 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 be used to determine the result of auto-programming after it has been finished (refer to **20.4.2 Status Check Procedure**).

Do not write additions to the already programmed addresses.

Also, when the FMR02 bit in the FMR0 register is set to 0 (rewrite disabled), or the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register is set to 1 (rewrite disabled), program commands targeting block 0 are not acknowledged.

In EW1 mode, do not execute this command for any address which a rewrite control program is allocated.

Figure 20.19 shows the Program Command in EW1 Mode (When Suspend Function Disabled). Figure 20.20 shows the Program Command in EW1 Mode (When Suspend Function Enabled).

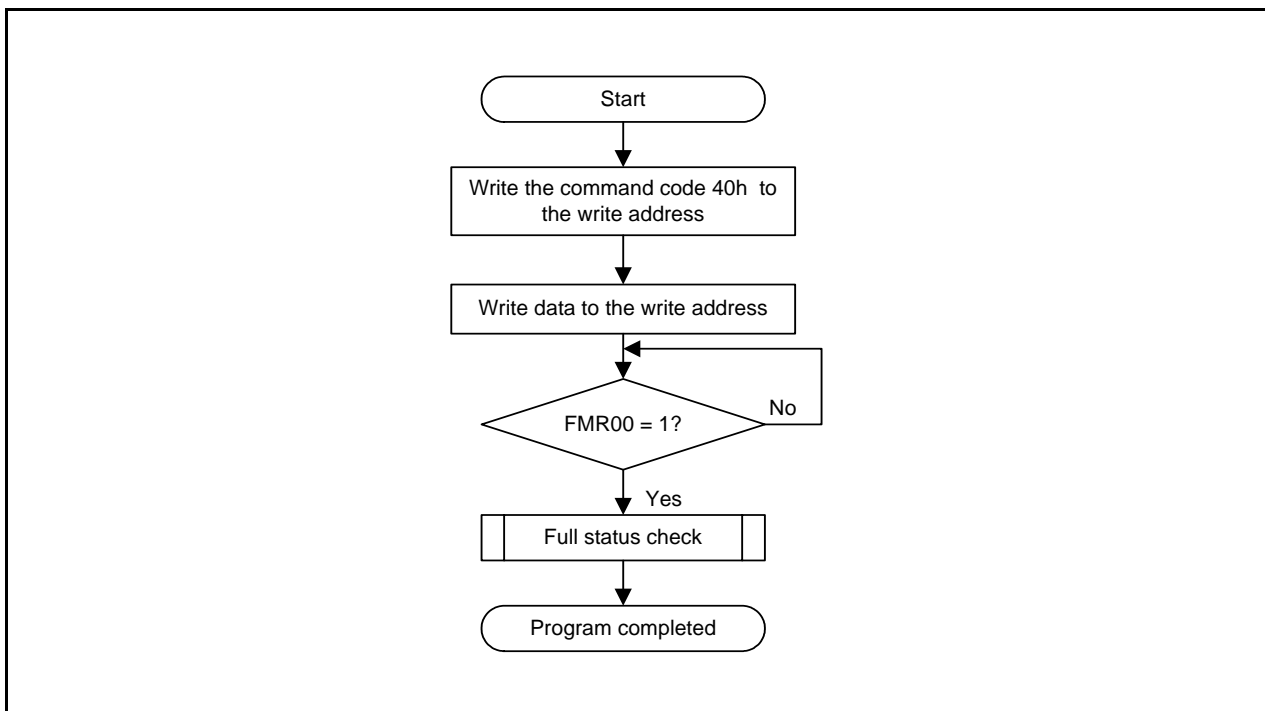


Figure 20.19 Program Command in EW1 Mode (When Suspend Function Disabled)

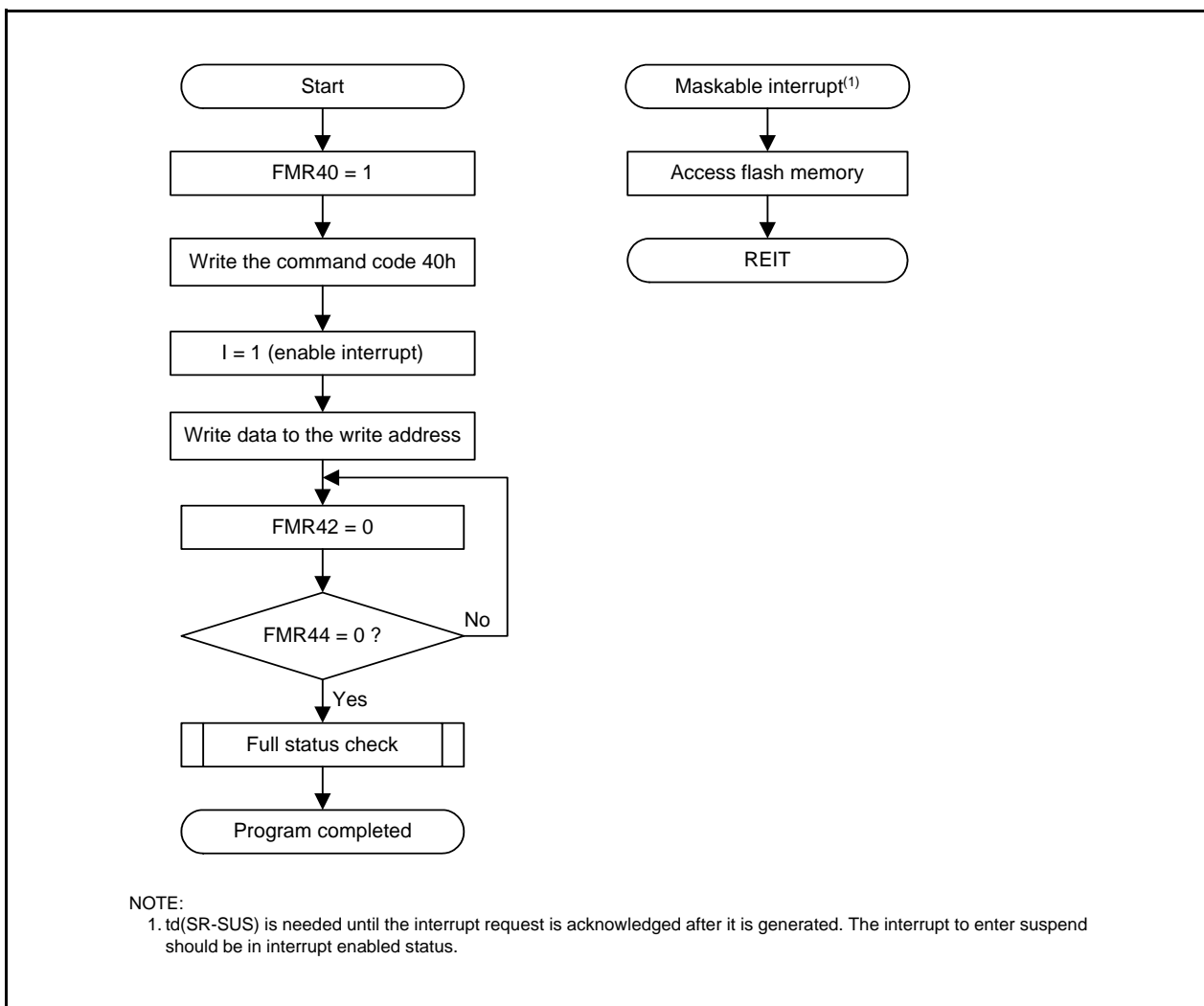


Figure 20.20 Program Command in EW1 Mode (When Suspend Function Enabled)

- Block Erase

When 20h is first written and then D0h is written to a given block address, an auto-erase operation (erase and verify) of the specified block starts.

The FMR00 bit in the FMR0 register can be used to determine whether auto-erasure has completed.

The FMR00 bit is set to 0 during auto-erasure and set to 1 when auto-erasure completes.

The FMR07 bit in the FMR0 register can be used to determine the result of auto-erasure after auto-erasure has completed (refer to **20.4.2 Status Check Procedure**).

Also, when the FMR02 bit in the FMR0 register is set to 0 (rewrite disabled), or the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register is set to 1 (rewrite disabled), block erase commands targeting block 0 are not acknowledged.

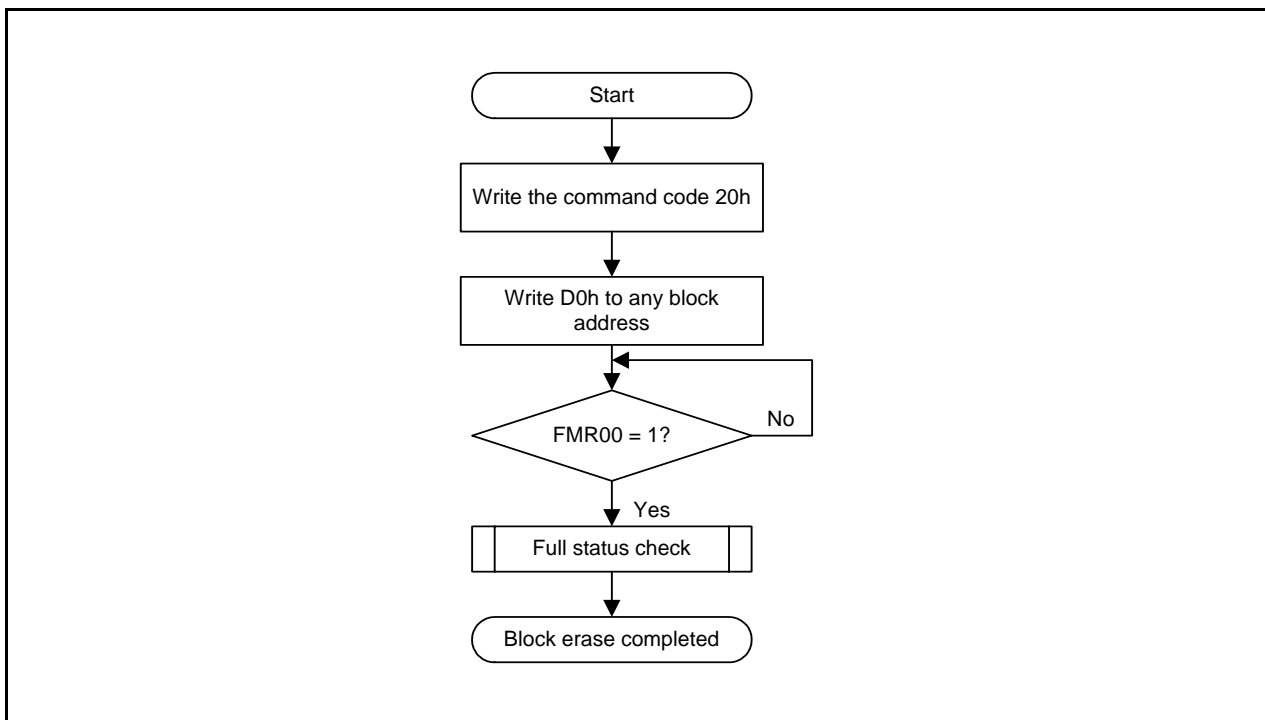
Do not use the block erase command during program-suspend.

Do not execute this command for any address to which a rewrite control program is allocated.

Figure 20.21 shows the Block Erase Command in EW1 Mode (When Suspend Function Disabled). Figure 20.22 shows the Block Erase Command in EW1 Mode (When Suspend Function Enabled).

If the programming and erasure endurance is  $n$  ( $n = 100, 1000, \text{ or } 10,000$ ), each block can be erased  $n$  times. For example, if 1,024 1-byte writes are performed to block A, a 1-Kbyte block, and then the block is erased, the erase count stands at one. When performing 100 or more rewrites, the actual erase count can be reduced by executing programming operations in such a way that all blank areas are used before performing an erase operation. Avoid rewriting only particular blocks and try to average out the programming and erasure endurance of the blocks.

It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.



**Figure 20.21 Block Erase Command in EW1 Mode (When Suspend Function Disabled)**

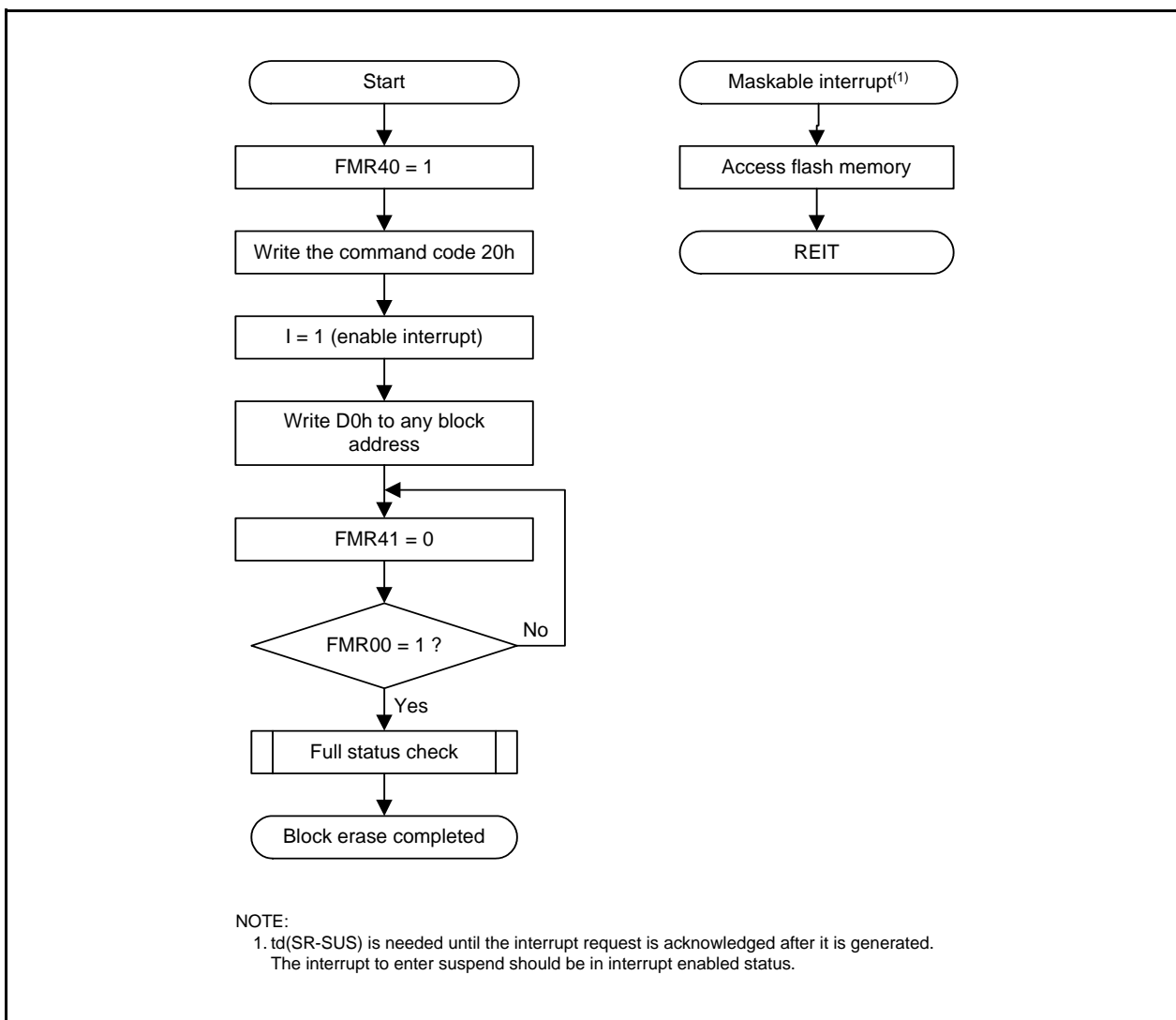


Figure 20.22 Block Erase Command in EW1 Mode (When Suspend Function Enabled)



### 20.4.4.2 Suspend Function

The suspend function halts auto-erase and auto-programming temporarily while these operations are in progress. This function is used for interrupt handling as the user ROM area can be read after the above operations have been suspended.

When the suspend function is used in EW1 mode, the MCU enters erase-suspend or program-suspend after an interrupt request is acknowledged.

To enable the suspend function, set the FMR40 bit to 1 (suspend enabled). The interrupt to enter suspend should also be set to enable beforehand. When  $t_d$  (SR-SUS) has elapsed after the interrupt request is generated, the request is acknowledged.

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

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

Figure 20.23 shows the Timing of Suspend Operation in EW1 Mode. Figure 20.24 shows the Program Flowchart during Erase-Suspend in EW1 Mode.

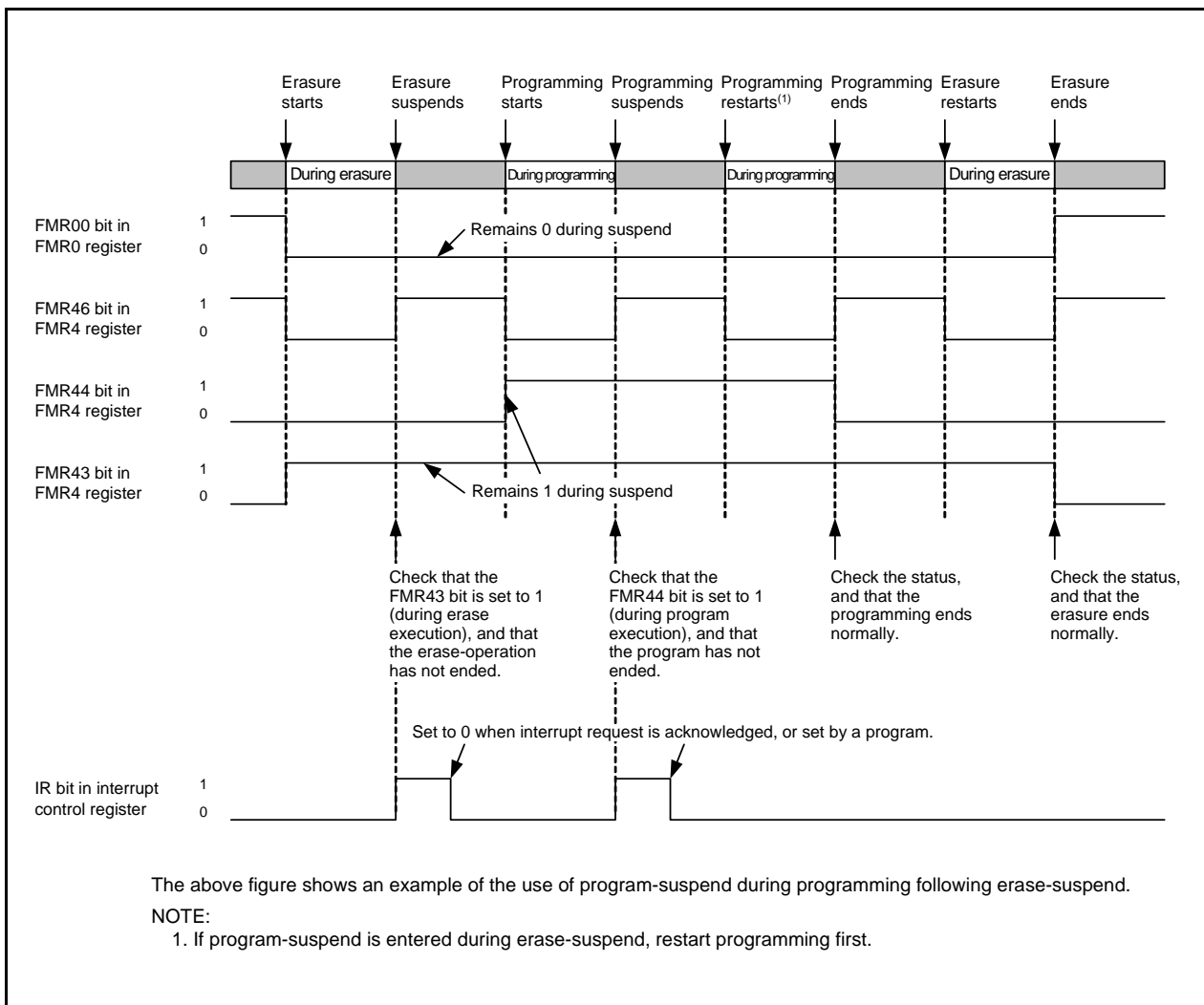


Figure 20.23 Timing of Suspend Operation in EW1 Mode

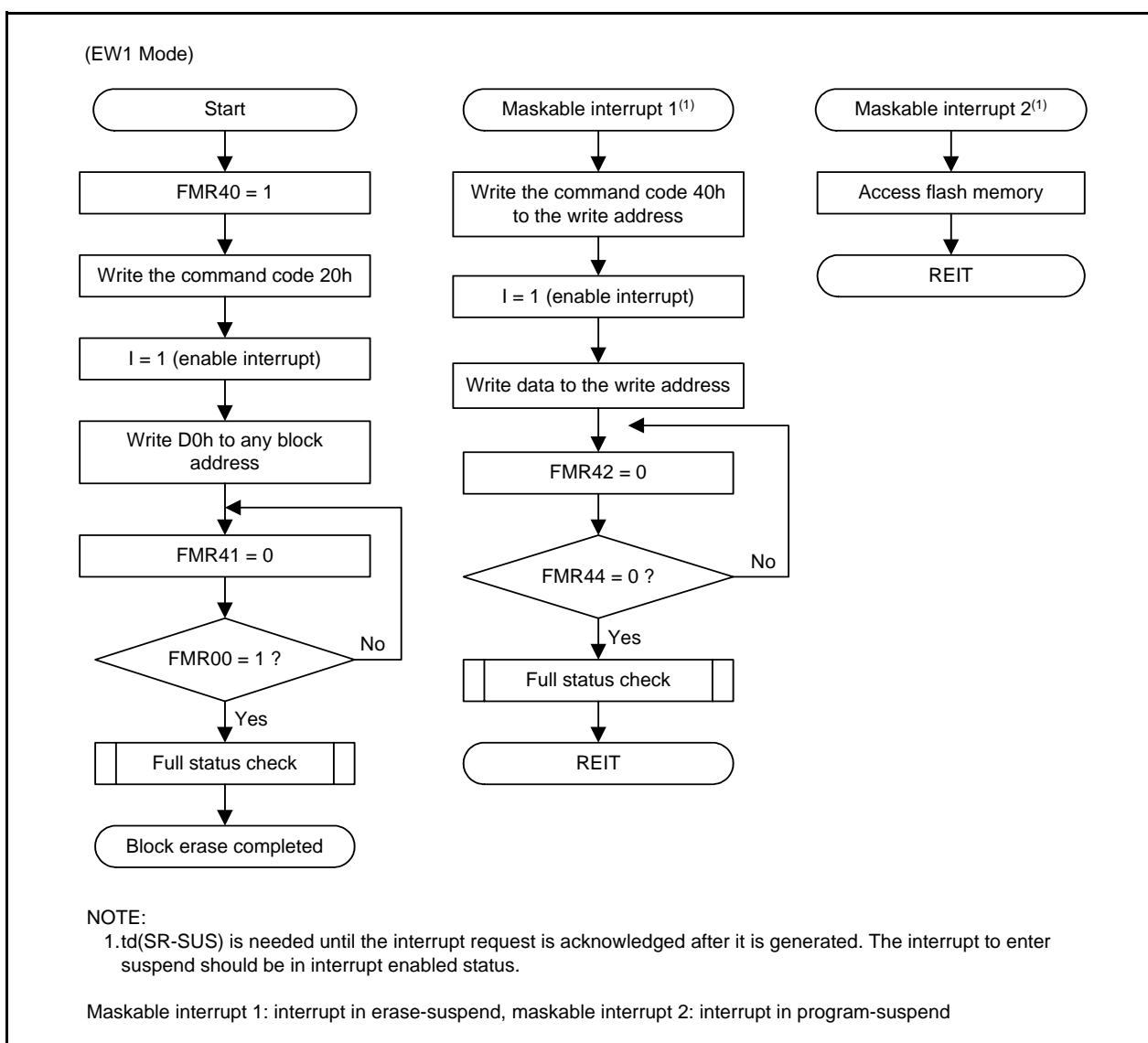


Figure 20.24 Program Flowchart during Erase-Suspend in EW1 Mode

### 20.4.4.3 EW1 Mode Interrupts

In EW1 mode, maskable interrupts can be used.

Table 20.6 lists the EW1 Mode Interrupts. Refer to **20.7.1.3 Non-Maskable Interrupts** for details of the non-maskable interrupt.

Table 20.6 EW1 Mode Interrupts

Status	When Maskable Interrupt Request is Acknowledged
During auto-erasure (erase-suspend function enabled)	Auto-erasure is suspended after td(SR-SUS) and interrupt handling is executed. Auto-erasure can be restarted by setting the FMR41 bit in the FMR4 register to 0 (erasure restarts) after interrupt handling completes.
During auto-erasure (erase-suspend function disabled)	Auto-erasure has priority and the interrupt request acknowledgement is put on standby. Interrupt handling is executed after auto-erasure completes.
During auto-programming (program suspend function enabled)	Auto-programming is suspended after td(SR-SUS) and interrupt handling is executed. Auto-programming can be restarted by setting the FMR42 bit in the FMR4 register to 0 (programming restarts) after interrupt handling completes.
During auto-programming (program suspend function disabled)	Auto-programming has priority and the interrupt request acknowledgement is put on standby. Interrupt handling is executed after auto-programming completes.

## 20.5 Standard Serial I/O Mode

In standard serial I/O mode, the user ROM area can be rewritten while the MCU is mounted on-board by using a serial programmer which is suitable for the MCU.

There are three types of standard serial I/O modes:

- Standard serial I/O mode 1 .....Clock synchronous serial I/O used to connect with a serial programmer
- Standard serial I/O mode 2 .....Clock asynchronous serial I/O used to connect with a serial programmer
- Standard serial I/O mode 3 .....Special clock asynchronous serial I/O used to connect with a serial programmer

This MCU uses standard serial I/O mode 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 details. Refer to the user's manual of your serial programmer for instructions on how to use it.

Table 20.7 lists the Pin Functions (Flash Memory Standard Serial I/O Mode 2) and Figure 20.25 shows an Example of Pin Processing in Standard Serial I/O Mode 2. Table 20.8 lists the Pin Functions (Flash Memory Standard Serial I/O Mode 3) and Figure 20.26 shows an Example of Pin Processing in Standard Serial I/O Mode 3.

After processing the pins shown in Table 20.8 and rewriting the flash memory using the programmer, apply "H" to the MODE pin and reset the hardware to run a program in the flash memory in single-chip mode.

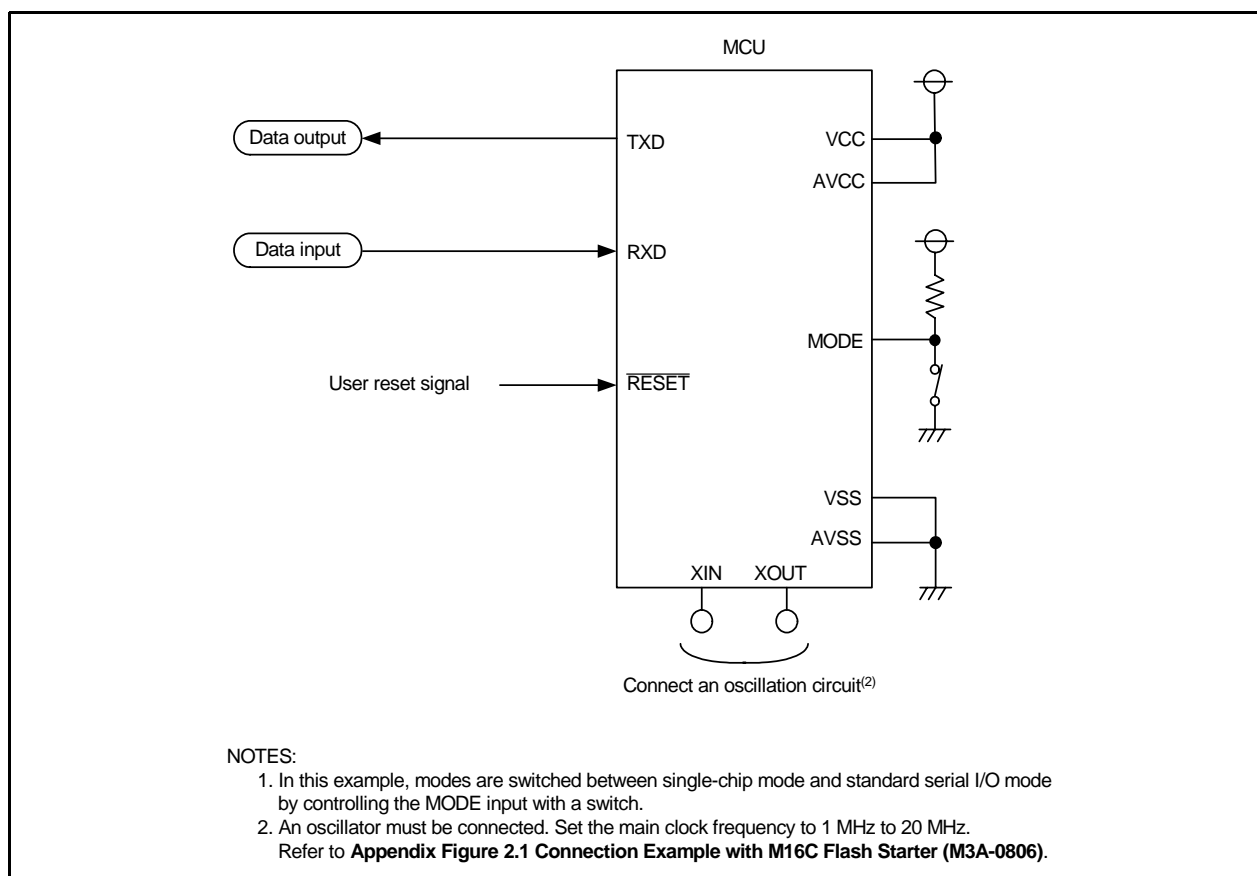
### 20.5.1 ID Code Check Function

The ID code check function determines whether the ID codes sent from the serial programmer and those written in the flash memory match.

Refer to **13. ID Code Areas** for details of the ID code check.

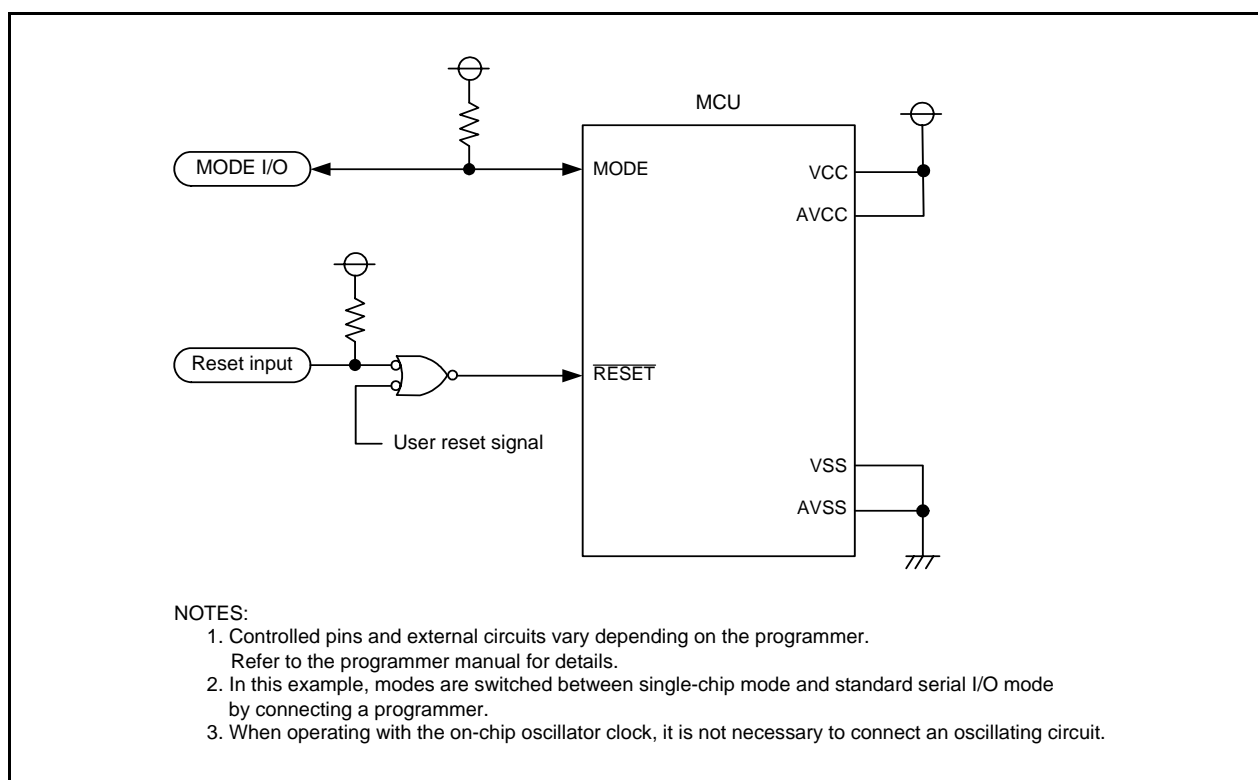
**Table 20.7 Pin Functions (Flash Memory Standard Serial I/O Mode 2)**

Pin	Name	I/O	Description
VCC, VSS	Power input		Apply the voltage guaranteed for programming and erasure to the VCC pin and 0 V to the VSS pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin.
P4_6/XIN	P4_6 input/clock input	I	Connect a ceramic resonator or crystal oscillator between the XIN and XOUT pins.
P4_7/XOUT	P4_7 input/clock output	I/O	
P0_1 to P0_3, P0_5	Input port P0	I	Input "H" or "L" level signal or leave the pin open.
P1_0 to P1_7	Input port P1	I	
P2_0 to P2_7	Input port P2	I	
P3_3 to P3_5	Input port P3	I	
P4_2/VREF	Input port P4	I	
MODE	MODE	I/O	Input "L".
P0_0	TXD output	O	Serial data output pin.
P4_5	RXD input	I	Serial data input pin.

**Figure 20.25 Example of Pin Processing in Standard Serial I/O Mode 2**

**Table 20.8 Pin Functions (Flash Memory Standard Serial I/O Mode 3)**

Pin	Name	I/O	Description
VCC, VSS	Power input		Apply the voltage guaranteed for programming and erasure to the VCC pin and 0 V to the VSS pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin.
P4_6/XIN	P4_6 input/clock input	I	Connect a ceramic resonator or crystal oscillator between the 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_3, P0_5	Input port P0	I	Input "H" or "L" level signal or leave the pin open.
P1_0 to P1_7	Input port P1	I	
P2_0 to P2_7	Input port P2	I	
P3_3 to P3_5	Input port P3	I	
P4_2/VREF, P4_5	Input port P4	I	
MODE	MODE	I/O	Serial data I/O pin. Connect to the flash programmer.

**Figure 20.26 Example of Pin Processing in Standard Serial I/O Mode 3**

## 20.6 Parallel I/O Mode

Parallel I/O mode is used to input and output software commands, addresses and data necessary to control (read, program, and erase) the on-chip flash memory. Use a parallel programmer which supports this MCU. Contact the manufacturer of the parallel programmer for more information, and refer to the user's manual of the parallel programmer for details on how to use it.

ROM areas shown in Figures 20.1 and 20.2 can be rewritten in parallel I/O mode.

### 20.6.1 ROM Code Protect Function

The ROM code protect function disables the reading and rewriting of the flash memory. (Refer to the **20.3.2 ROM Code Protect Function**.)

## 20.7 Notes on Flash Memory

### 20.7.1 CPU Rewrite Mode

#### 20.7.1.1 Operating Speed

Before entering CPU rewrite mode (EW0 mode), select 5 MHz or below for the CPU clock using the CM06 bit in the CM0 register and bits CM16 to CM17 in the CM1 register. This does not apply to EW1 mode.

#### 20.7.1.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because they reference data in the flash memory: UND, INTO, and BRK.

#### 20.7.1.3 Non-Maskable Interrupts

- EW0 Mode

Once a watchdog timer, oscillation stop detection, voltage monitor1, or voltage monitor 2 interrupt request is acknowledged, auto-erasure or auto-programming is forcibly stopped immediately and the flash memory is reset. Interrupt handling starts after a fixed period and the flash memory restarts.

As the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be readable. Execute auto-erasure again and ensure it completes normally.

The watchdog timer does not stop during command operation, so that interrupt requests may be generated. Initialize the watchdog timer regularly.

Do not use the address match interrupt while a command is being executed because the vector of the address match interrupt is allocated in ROM.

Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

- EW1 Mode

Once a watchdog timer, oscillation stop detection, voltage monitor1, or voltage monitor 2 interrupt request is acknowledged, auto-erasure or auto-programming is forcibly stopped immediately and the flash memory is reset. Interrupt handling starts after a fixed period and the flash memory restarts.

As the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be readable. Execute auto-erasure again and ensure it completes normally.

The watchdog timer does not stop even during command operation, so that interrupt requests may be generated. Initialize the watchdog timer by using the erase-suspend function.

Do not use the address match interrupt while a command is being executed because the vector of the address match interrupt is allocated in ROM.

Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

#### **20.7.1.4 How to Access**

Write 0 before writing 1 when setting Bits FMR01, FMR02 in the FMR0 register, or FMR11 bit in the FMR1 register to 1. Do not generate an interrupt between writing 0 and 1.

#### **20.7.1.5 Rewriting User ROM Area**

In EW0 Mode, if the supply voltage drops while rewriting any block in which a rewrite control program is stored, it may not be possible to rewrite the flash memory because the rewrite control program cannot be rewritten correctly. In this case, use standard serial I/O mode.

#### **20.7.1.6 Program**

Do not write additions to the already programmed address.

#### **20.7.1.7 Suspend**

Do not use the block erase command during program-suspend.

#### **20.7.1.8 Entering Stop Mode or Wait Mode**

Do not enter stop mode or wait mode during erase-suspend.

#### **20.7.1.9 Program and Erase Voltage for Flash Memory**

To perform programming and erasure, use  $V_{CC} = 2.7\text{ V}$  to  $5.5\text{ V}$  as the supply voltage. Do not perform programming and erasure at less than  $2.7\text{ V}$ .



## 21. Reducing Power Consumption

### 21.1 Overview

This chapter describes key points and processing methods for reducing power consumption.

### 21.2 Key Points and Processing Methods for Reducing Power Consumption

Key points for reducing power consumption are shown below. They should be referred to when designing a system or creating a program.

#### 21.2.1 Voltage Detection Circuit

When voltage monitor 1 is not used, set the VCA26 bit in the VCA2 register to 0 (voltage detection 1 circuit disabled). When voltage monitor 2 is not used, set the VCA27 bit in the VCA2 register to 0 (voltage detection 2 circuit disabled).

If the power-on reset and voltage monitor 0 reset are not used, set the VCA25 bit in the VCA2 register to 0 (voltage detection 0 circuit disabled).

#### 21.2.2 Ports

Even after the MCU enters wait mode or stop mode, the states of the I/O ports are retained. Current flows into the output ports in the active state, and shoot-through current flows into the input ports in the high-impedance state. Unnecessary ports should be set to input and fixed to a stable electric potential before the MCU enters wait mode or stop mode.

#### 21.2.3 Clocks

Power consumption generally depends on the number of the operating clocks and their frequencies. The fewer the number of operating clocks or the lower their frequencies, the more power consumption decreases. Unnecessary clocks should be stopped accordingly.

Stopping XIN clock: CM05 bit in CM0 register

Stopping low-speed on-chip oscillator oscillation: CM14 bit in CM1 register

Stopping high-speed on-chip oscillator oscillation: HRA00 bit in HRA0 register

#### 21.2.4 Wait Mode, Stop Mode

Power consumption can be reduced in wait mode and stop mode. Refer to **10.4 Power Control** for details.

#### 21.2.5 Stopping Peripheral Function Clocks

If the peripheral function f1, f2, f4, f8, and f32 clocks are not necessary in wait mode, set the CM02 bit in the CM0 register to 1 (peripheral function clock stops in wait mode). This will stop the f1, f2, f4, f8, and f32 clocks in wait mode.

#### 21.2.6 Timers

If timer RA is not used, set the TCKCUT bit in the TRAMR register to 1 (count source cutoff).

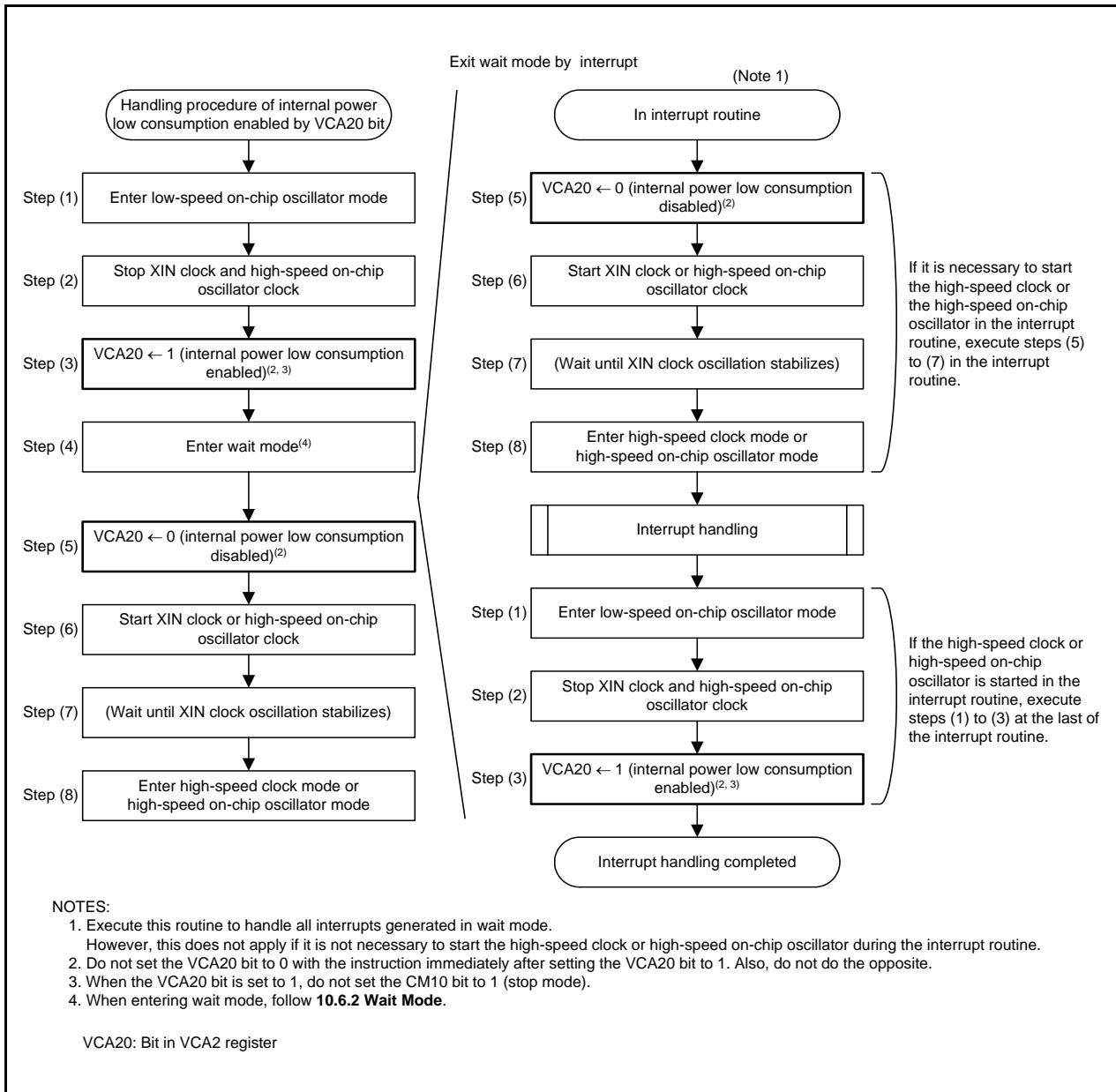
If timer RB is not used, set the TCKCUT bit in the TRBMR register to 1 (count source cutoff).

#### 21.2.7 A/D Converter

When A/D conversion is not performed, set the VCUT bit in the ADCON1 register to 0 (VREF unconnected). To perform A/D conversion, wait for at least 1  $\mu$ s after setting the VCUT bit to 1 (VREF connected) before starting the A/D conversion.

### 21.2.8 Reducing Internal Power Consumption

When the MCU enters wait mode using low-speed on-chip oscillator mode, internal power consumption can be reduced by using the VCA20 bit in the VCA2 register. Figure 21.1 shows the Handling Procedure of Internal Power Low Consumption Using VCA20 Bit. To enable internal power low consumption by the VCA20 bit, follow Figure 21.1 Handling Procedure of Internal Power Low Consumption Using VCA20 Bit.



**Figure 21.1 Handling Procedure of Internal Power Low Consumption Using VCA20 Bit**

### 21.2.9 Stopping Flash Memory

In low-speed on-chip oscillator mode, power consumption can be further reduced by stopping the flash memory using the FMSTP bit in the FMR0 register.

Access to the flash memory is disabled by setting the FMSTP bit to 1 (flash memory stops). The FMSTP bit must be written to by a program transferred to RAM.

When the MCU enters stop mode or wait mode while CPU rewrite mode is disabled, the power for the flash memory is automatically turned off. It is turned back on again after the MCU exit stop mode or wait mode. This eliminates the need to set the FMR0 register.

Figure 21.2 shows the Handling Procedure Example of Low Power Consumption Using FMSTP Bit.

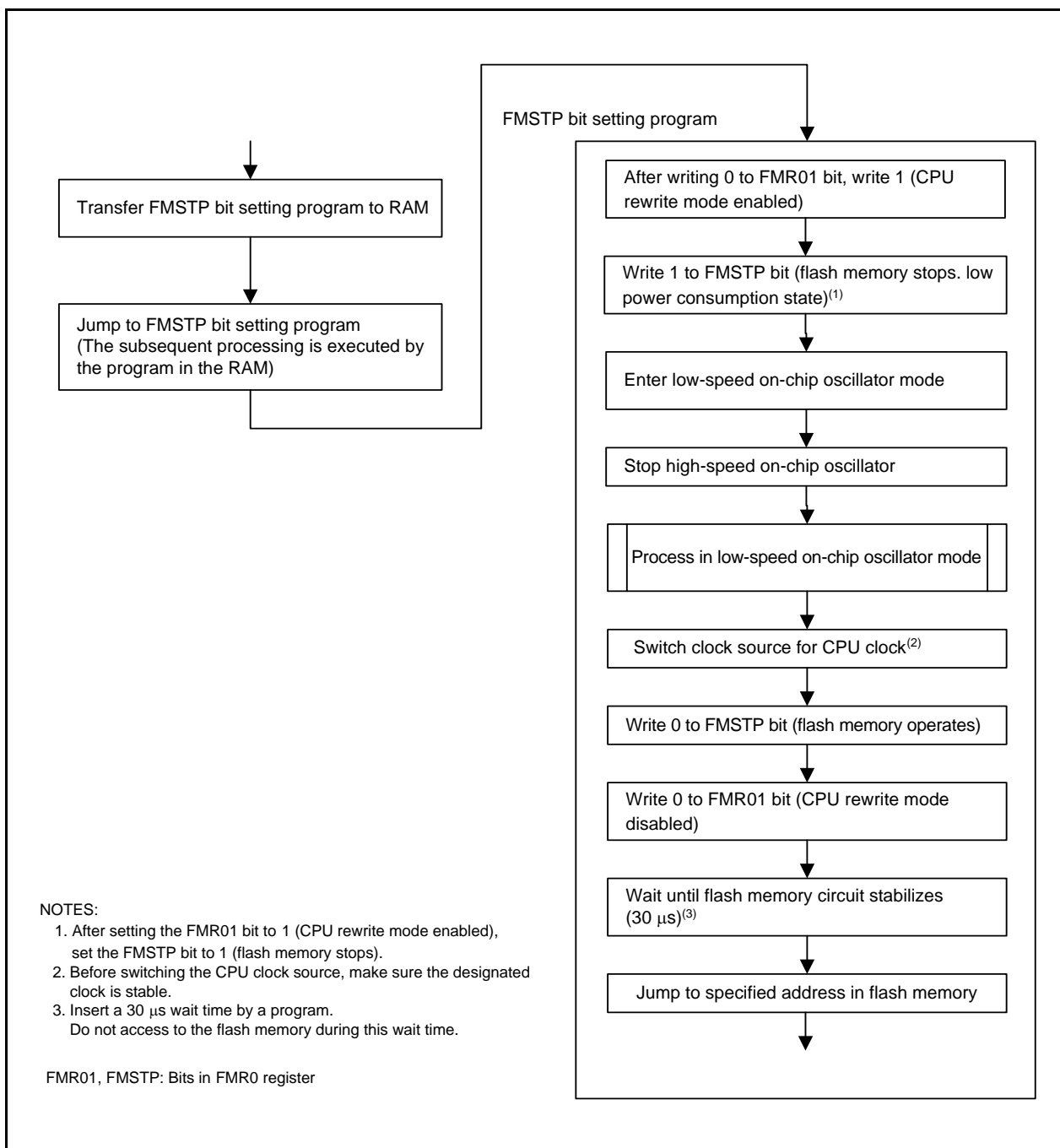


Figure 21.2 Handling Procedure Example of Low Power Consumption Using FMSTP Bit

### 21.2.10 Low-Current-Consumption Read Mode

In low-speed on-chip oscillator mode, the current consumption when reading the flash memory can be reduced by setting the FMR47 bit in the FMR4 register to 1 (enabled).

Figure 21.3 shows the Handling Procedure Example of Low-Current-Consumption Read Mode.

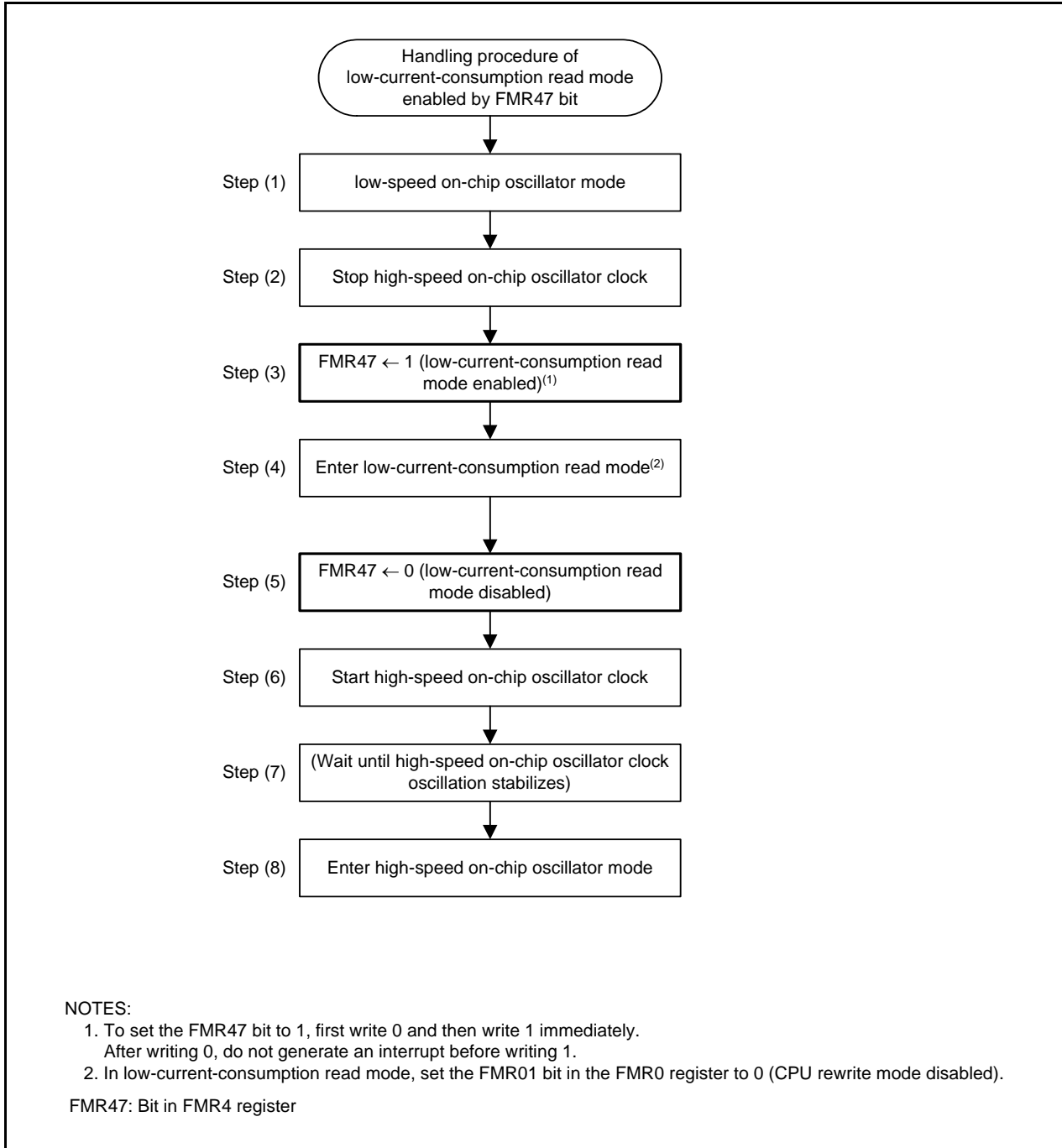


Figure 21.3 Handling Procedure Example of Low-Current-Consumption Read Mode

## 22. Electrical Characteristics

The electrical characteristics of N version ( $T_{opr} = -20^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ ) and D version ( $T_{opr} = -40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ ) are listed below.

Please contact Renesas Technology sales offices for the electrical characteristics in the Y version ( $T_{opr} = -20^{\circ}\text{C}$  to  $105^{\circ}\text{C}$ ).

**Table 22.1 Absolute Maximum Ratings**

Symbol	Parameter	Condition	Rated Value	Unit
$V_{cc}/AV_{cc}$	Supply voltage		-0.3 to 6.5	V
$V_i$	Input voltage		-0.3 to $V_{cc} + 0.3$	V
$V_o$	Output voltage		-0.3 to $V_{cc} + 0.3$	V
$P_d$	Power dissipation	$T_{opr} = 25^{\circ}\text{C}$	500	mW
$T_{opr}$	Operating ambient temperature		-20 to 85 (N version) / -40 to 85 (D version)	$^{\circ}\text{C}$
$T_{stg}$	Storage temperature		-65 to 150	$^{\circ}\text{C}$

**Table 22.2 Recommended Operating Conditions**

Symbol	Parameter		Conditions	Standard			Unit	
				Min.	Typ.	Max.		
V <sub>CC</sub>	Supply voltage			2.2	–	5.5	V	
AV <sub>CC</sub>	Supply voltage			2.7	–	5.5	V	
V <sub>SS</sub> /AV <sub>SS</sub>	Supply voltage			–	0	–	V	
V <sub>IH</sub>	Input “H” voltage			0.8 V <sub>CC</sub>	–	V <sub>CC</sub>	V	
V <sub>IL</sub>	Input “L” voltage			0	–	0.2 V <sub>CC</sub>	V	
I <sub>OH(sum)</sub>	Peak sum output “H” current	Sum of all pins I <sub>OH(peak)</sub>		–	–	–160	mA	
I <sub>OH(sum)</sub>	Average sum output “H” current	Sum of all pins I <sub>OH(avg)</sub>		–	–	–80	mA	
I <sub>OH(peak)</sub>	Peak output “H” current	Except P2_0 to P2_7		–	–	–10	mA	
		P2_0 to P2_7		–	–	–40	mA	
I <sub>OH(avg)</sub>	Average output “H” current	Except P2_0 to P2_7		–	–	–5	mA	
		P2_0 to P2_7		–	–	–20	mA	
I <sub>OL(sum)</sub>	Peak sum output “L” currents	Sum of all pins I <sub>OL(peak)</sub>		–	–	160	mA	
I <sub>OL(sum)</sub>	Average sum output “L” currents	Sum of all pins I <sub>OL(avg)</sub>		–	–	80	mA	
I <sub>OL(peak)</sub>	Peak output “L” currents	Except P2_0 to P2_7		–	–	10	mA	
		P2_0 to P2_7		–	–	40	mA	
I <sub>OL(avg)</sub>	Average output “L” current	Except P2_0 to P2_7		–	–	5	mA	
		P2_0 to P2_7		–	–	20	mA	
f <sub>(XIN)</sub>	XIN clock input oscillation frequency		3.0 V ≤ V <sub>CC</sub> ≤ 5.5 V	0	–	20	MHz	
			2.7 V ≤ V <sub>CC</sub> < 3.0 V	0	–	10	MHz	
			2.2 V ≤ V <sub>CC</sub> < 2.7 V	0	–	5	MHz	
–	System clock	OCD2 = 0 XIN clock selected	3.0 V ≤ V <sub>CC</sub> ≤ 5.5 V	0	–	20	MHz	
			2.7 V ≤ V <sub>CC</sub> < 3.0 V	0	–	10	MHz	
			2.2 V ≤ V <sub>CC</sub> < 2.7 V	0	–	5	MHz	
		OCD2 = 1 On-chip oscillator clock selected	FRA01 = 0 Low-speed on-chip oscillator clock selected	–	125	–	–	kHz
			FRA01 = 1 High-speed on-chip oscillator clock selected 3.0 V ≤ V <sub>CC</sub> ≤ 5.5 V	–	–	20	–	MHz
			FRA01 = 1 High-speed on-chip oscillator clock selected 2.7 V ≤ V <sub>CC</sub> ≤ 5.5 V	–	–	10	–	MHz
			FRA01 = 1 High-speed on-chip oscillator clock selected 2.2 V ≤ V <sub>CC</sub> ≤ 5.5 V	–	–	5	–	MHz

## NOTES:

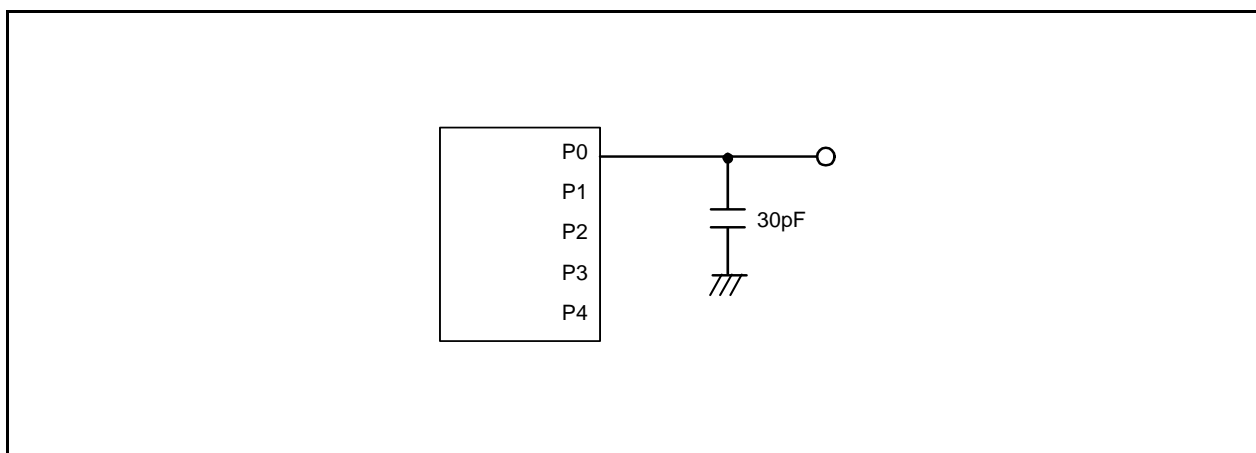
- V<sub>CC</sub> = 2.2 to 5.5 V at T<sub>opr</sub> = –20 to 85°C (N version) / –40 to 85°C (D version), unless otherwise specified.
- The average output current indicates the average value of current measured during 100 ms.

**Table 22.3 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			2.2	–	$AV_{CC}$	V
$V_{IA}$	Analog input voltage <sup>(2)</sup>			0	–	$AV_{CC}$	V
–	A/D operating clock frequency	Without sample and hold	$V_{ref} = AV_{CC} = 2.7 \text{ to } 5.5 \text{ V}$	0.25	–	10	MHz
		With sample and hold	$V_{ref} = AV_{CC} = 2.7 \text{ to } 5.5 \text{ V}$	1	–	10	MHz

## NOTES:

1.  $AV_{CC} = 2.7$  to  $5.5 \text{ V}$  at  $T_{opr} = -20$  to  $85^\circ\text{C}$  (N version) /  $-40$  to  $85^\circ\text{C}$  (D version), unless otherwise specified.
2. When the analog input voltage is over the reference voltage, the A/D conversion result will be 3FFh in 10-bit mode and FFh in 8-bit mode.

**Figure 22.1 Ports P0 to P4 Timing Measurement Circuit**

**Table 22.4 Flash Memory (Program ROM) Electrical Characteristics**

Symbol	Parameter	Conditions	Standard			Unit
			Min.	Typ.	Max.	
–	Program/erase endurance <sup>(2)</sup>	R8C/2K Group	100 <sup>(3)</sup>	–	–	times
		R8C/2L 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 suspend		–	–	97+CPU clock × 6 cycles	μs
–	Interval from erase start/restart until following suspend request		650	–	–	μs
–	Interval from program start/restart until following suspend request		0	–	–	ns
–	Time from suspend until program/erase restart		–	–	3+CPU clock × 4 cycles	μs
–	Program, erase voltage		2.7	–	5.5	V
–	Read voltage		2.2	–	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 T<sub>opr</sub> = 0 to 60°C, 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 10,000), each block can be erased n times. For example, if 1,024 1-byte writes are performed to block A, a 1 Kbyte block, and then the block is erased, the programming/erasure endurance still stands at one.  
However, the same address must not be programmed more than once per erase operation (overwriting prohibited).
- 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 count can be reduced by writing to sequential addresses in turn so that as much of the block as possible is used up before performing an erase operation. For example, when programming groups of 16 bytes, the effective number of rewrites can be minimized by programming up to 128 groups before erasing them all in one operation. It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.
- If an error occurs during block erase, attempt to execute the clear status register command, then execute the block erase command at least three times until the erase error does not occur.
- 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 22.5 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</sub> (SR-SUS)	Time delay from suspend request until suspend		–	–	97+CPU clock × 6 cycles	μs
–	Interval from erase start/restart until following suspend request		650	–	–	μs
–	Interval from program start/restart until following suspend request		0	–	–	ns
–	Time from suspend until program/erase restart		–	–	3+CPU clock × 4 cycles	μs
–	Program, erase voltage		2.7	–	5.5	V
–	Read voltage		2.2	–	5.5	V
–	Program, erase temperature		–20 <sup>(8)</sup>	–	85	°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 T<sub>opr</sub> = –20 to 85°C (N version) / –40 to 85°C (D 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 10,000), each block can be erased n times. For example, if 1,024 1-byte writes are performed to block A, a 1 Kbyte block, and then the block is erased, the programming/erasure endurance still stands at one.  
However, the same address must not be programmed more than once per erase operation (overwriting prohibited).
- 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 is the same as that in program ROM.
- In a system that executes multiple programming operations, the actual erasure count can be reduced by writing to sequential addresses in turn so that as much of the block as possible is used up before performing an erase operation. For example, when programming groups of 16 bytes, the effective number of rewrites can be minimized by programming up to 128 groups before erasing them all in one operation. It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.
- If an error occurs during block erase, attempt to execute the clear status register command, then execute the block erase command at least three times until the erase error does not occur.
- Customers desiring program/erase failure rate information should contact their Renesas technical support representative.
- 40°C for D version.
- The data hold time includes time that the power supply is off or the clock is not supplied.

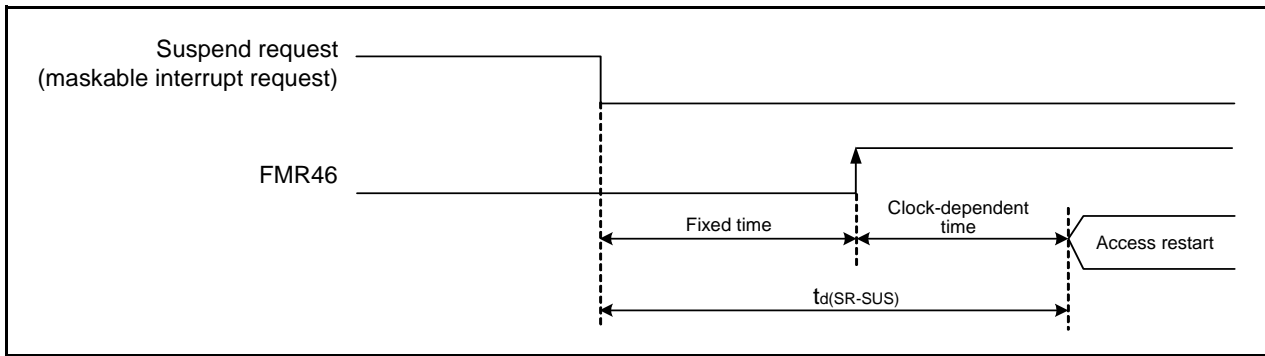


Figure 22.2 Time delay until Suspend

Table 22.6 Voltage Detection 0 Circuit Electrical Characteristics

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V <sub>det0</sub>	Voltage detection level		2.2	2.3	2.4	V
–	Voltage detection circuit self power consumption	VCA25 = 1, V <sub>CC</sub> = 5.0 V	–	0.9	–	μA
t <sub>d(E-A)</sub>	Waiting time until voltage detection circuit operation starts <sup>(2)</sup>		–	–	300	μs
V <sub>ccmin</sub>	MCU operating voltage minimum value		2.2	–	–	V

## NOTES:

- The measurement condition is V<sub>CC</sub> = 2.2 to 5.5 V and T<sub>opr</sub> = –20 to 85°C (N version) / –40 to 85°C (D version).
- Necessary time until the voltage detection circuit operates when setting to 1 again after setting the VCA25 bit in the VCA2 register to 0.

Table 22.7 Voltage Detection 1 Circuit Electrical Characteristics

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V <sub>det1</sub>	Voltage detection level <sup>(4)</sup>		2.70	2.85	3.00	V
–	Voltage monitor 1 interrupt request generation time <sup>(2)</sup>		–	40	–	μs
–	Voltage detection circuit self power consumption	VCA26 = 1, V <sub>CC</sub> = 5.0 V	–	0.6	–	μA
t <sub>d(E-A)</sub>	Waiting time until voltage detection circuit operation starts <sup>(3)</sup>		–	–	100	μs

## NOTES:

- The measurement condition is V<sub>CC</sub> = 2.2 to 5.5 V and T<sub>opr</sub> = –20 to 85°C (N version) / –40 to 85°C (D version).
- Time until the voltage monitor 1 interrupt request is generated after the voltage passes V<sub>det1</sub>.
- Necessary time until the voltage detection circuit operates when setting to 1 again after setting the VCA26 bit in the VCA2 register to 0.
- This parameter shows the voltage detection level when the power supply drops.  
The voltage detection level when the power supply rises is higher than the voltage detection level when the power supply drops by approximately 0.1 V.

Table 22.8 Voltage Detection 2 Circuit Electrical Characteristics

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
V <sub>det2</sub>	Voltage detection level		3.3	3.6	3.9	V
–	Voltage monitor 2 interrupt request generation time <sup>(2)</sup>		–	40	–	μs
–	Voltage detection circuit self power consumption	VCA27 = 1, V <sub>CC</sub> = 5.0 V	–	0.6	–	μA
t <sub>d(E-A)</sub>	Waiting time until voltage detection circuit operation starts <sup>(3)</sup>		–	–	100	μs

## NOTES:

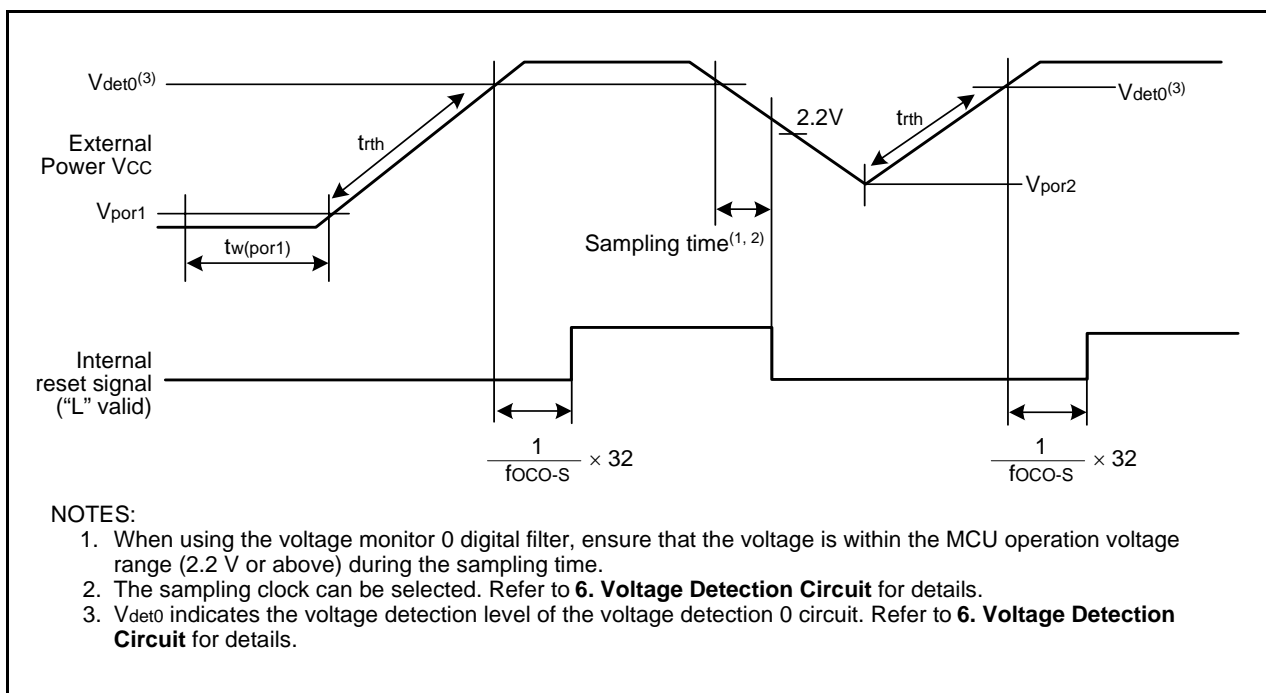
- The measurement condition is V<sub>CC</sub> = 2.2 to 5.5 V and T<sub>opr</sub> = –20 to 85°C (N version) / –40 to 85°C (D version).
- Time until the voltage monitor 2 interrupt request is generated after the voltage passes V<sub>det2</sub>.
- Necessary time until the voltage detection circuit operates after setting to 1 again after setting the VCA27 bit in the VCA2 register to 0.

**Table 22.9 Power-on Reset Circuit, Voltage Monitor 0 Reset 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 0 reset valid voltage		0	–	$V_{det0}$	V
$tr_{th}$	External power $V_{CC}$ rise gradient <sup>(2)</sup>		20	–	–	mV/msec

## NOTES:

1. The measurement condition is  $T_{opr} = -20$  to  $85^{\circ}\text{C}$  (N version) /  $-40$  to  $85^{\circ}\text{C}$  (D version), unless otherwise specified.
2. This condition (external power  $V_{CC}$  rise gradient) does not apply if  $V_{CC} \geq 1.0$  V.
3. To use the power-on reset function, enable voltage monitor 0 reset by setting the LVD0ON bit in the OFS register to 0, the VW0C0 and VW0C6 bits in the VW0C register to 1 respectively, and the VCA25 bit in the VCA2 register to 1.
4.  $t_{w(por1)}$  indicates the duration the external power  $V_{CC}$  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 30 s or more if  $-20^{\circ}\text{C} \leq T_{opr} \leq 85^{\circ}\text{C}$ , maintain  $t_{w(por1)}$  for 3,000 s or more if  $-40^{\circ}\text{C} \leq T_{opr} < -20^{\circ}\text{C}$ .

**Figure 22.3 Reset Circuit Electrical Characteristics**

**Table 22.10 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	$V_{CC} = 2.7 \text{ V to } 5.5 \text{ V}$ $-20^{\circ}\text{C} \leq T_{opr} \leq 85^{\circ}\text{C}^{(2)}$	39.2	40	40.8	MHz
		$V_{CC} = 2.7 \text{ V to } 5.5 \text{ V}$ $-40^{\circ}\text{C} \leq T_{opr} \leq 85^{\circ}\text{C}^{(2)}$	39.0	40	41.0	MHz
		$V_{CC} = 2.2 \text{ V to } 5.5 \text{ V}$ $-20^{\circ}\text{C} \leq T_{opr} \leq 85^{\circ}\text{C}^{(3)}$	35.2	40	44.8	MHz
		$V_{CC} = 2.2 \text{ V to } 5.5 \text{ V}$ $-40^{\circ}\text{C} \leq T_{opr} \leq 85^{\circ}\text{C}^{(3)}$	34.0	40	46.0	MHz
	High-speed on-chip oscillator frequency when correction value in FRA7 register is written to FRA1 register <sup>(4)</sup>	$V_{CC} = 5.0 \text{ V}, T_{opr} = 25^{\circ}\text{C}$	–	36.864	–	MHz
$V_{CC} = 2.7 \text{ V to } 5.5 \text{ V}$ $-20^{\circ}\text{C} \leq T_{opr} \leq 85^{\circ}\text{C}$		–3%	–	3%	%	
–	Value in FRA1 register after reset		08h	–	F7h	–
–	Oscillation frequency adjustment unit of high-speed on-chip oscillator	Adjust FRA1 register (value after reset) to -1	–	+0.3	–	MHz
–	Oscillation stability time	$V_{CC} = 5.0 \text{ V}, T_{opr} = 25^{\circ}\text{C}$	–	10	100	$\mu\text{s}$
–	Self power consumption at oscillation	$V_{CC} = 5.0 \text{ V}, T_{opr} = 25^{\circ}\text{C}$	–	550	–	$\mu\text{A}$

## NOTES:

- $V_{CC} = 2.2 \text{ to } 5.5 \text{ V}, T_{opr} = -20 \text{ to } 85^{\circ}\text{C}$  (N version) /  $-40 \text{ to } 85^{\circ}\text{C}$  (D version), unless otherwise specified.
- These standard values show when the FRA1 register value after reset is assumed.
- These standard values show when the corrected value of the FRA6 register is written to the FRA1 register.
- This enables the setting errors of bit rates such as 9600 bps and 38400 bps to be 0% when the serial interface is used in UART mode.

**Table 22.11 Low-speed On-Chip Oscillator Circuit Electrical Characteristics**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
fOCO-S	Low-speed on-chip oscillator frequency		30	125	250	kHz
–	Oscillation stability time		–	10	100	$\mu\text{s}$
–	Self power consumption at oscillation	$V_{CC} = 5.0 \text{ V}, T_{opr} = 25^{\circ}\text{C}$	–	15	–	$\mu\text{A}$

## NOTE:

- $V_{CC} = 2.2 \text{ to } 5.5 \text{ V}, T_{opr} = -20 \text{ to } 85^{\circ}\text{C}$  (N version) /  $-40 \text{ to } 85^{\circ}\text{C}$  (D version), unless otherwise specified.

**Table 22.12 Power Supply Circuit Timing Characteristics**

Symbol	Parameter	Condition	Standard			Unit
			Min.	Typ.	Max.	
t <sub>d</sub> (P-R)	Time for internal power supply stabilization during power-on <sup>(2)</sup>		1	–	2000	$\mu\text{s}$
t <sub>d</sub> (R-S)	STOP exit time <sup>(3)</sup>		–	–	150	$\mu\text{s}$

## NOTES:

- The measurement condition is  $V_{CC} = 2.2 \text{ to } 5.5 \text{ V}$  and  $T_{opr} = 25^{\circ}\text{C}$ .
- Waiting time until the internal power supply generation circuit stabilizes during power-on.
- Time until system clock supply starts after the interrupt is acknowledged to exit stop mode.

**Table 22.13 Electrical Characteristics (1) [V<sub>CC</sub> = 5 V]**

Symbol	Parameter		Condition	Standard			Unit	
				Min.	Typ.	Max.		
V <sub>OH</sub>	Output "H" voltage	Except P2_0 to P2_7, XOUT	I <sub>OH</sub> = -5 mA		V <sub>CC</sub> - 2.0	-	V <sub>CC</sub>	V
			I <sub>OH</sub> = -200 μA		V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
		P2_0 to P2_7	Drive capacity HIGH	I <sub>OH</sub> = -20 mA	V <sub>CC</sub> - 2.0	-	V <sub>CC</sub>	V
			Drive capacity LOW	I <sub>OH</sub> = -5 mA	V <sub>CC</sub> - 2.0	-	V <sub>CC</sub>	V
		XOUT	Drive capacity HIGH	I <sub>OH</sub> = -1 mA	V <sub>CC</sub> - 2.0	-	V <sub>CC</sub>	V
			Drive capacity LOW	I <sub>OH</sub> = -500 μA	V <sub>CC</sub> - 2.0	-	V <sub>CC</sub>	V
V <sub>OL</sub>	Output "L" voltage	Except P2_0 to P2_7, XOUT	I <sub>OL</sub> = 5 mA		-	-	2.0	V
			I <sub>OL</sub> = 200 μA		-	-	0.45	V
		P2_0 to P2_7	Drive capacity HIGH	I <sub>OL</sub> = 20 mA	-	-	2.0	V
			Drive capacity LOW	I <sub>OL</sub> = 5 mA	-	-	2.0	V
		XOUT	Drive capacity HIGH	I <sub>OL</sub> = 1 mA	-	-	2.0	V
			Drive capacity LOW	I <sub>OL</sub> = 500 μA	-	-	2.0	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	$\overline{\text{INT0}}, \overline{\text{INT1}}, \overline{\text{INT3}},$ $\overline{\text{K10}}, \overline{\text{K11}}, \overline{\text{K12}}, \overline{\text{K13}},$ $\overline{\text{TRAIO}}, \overline{\text{RXD0}}, \overline{\text{RXD2}},$ $\overline{\text{CLK0}}, \overline{\text{CLK2}}$			0.1	0.5	-	V
		$\overline{\text{RESET}}$			0.1	1.0	-	V
I <sub>IH</sub>	Input "H" current	V <sub>I</sub> = 5 V, V <sub>CC</sub> = 5 V		-	-	5.0	μA	
I <sub>IL</sub>	Input "L" current	V <sub>I</sub> = 0 V, V <sub>CC</sub> = 5 V		-	-	-5.0	μA	
R <sub>PULLUP</sub>	Pull-up resistance	V <sub>I</sub> = 0 V, V <sub>CC</sub> = 5 V		30	50	167	kΩ	
R <sub>iXIN</sub>	Feedback resistance	XIN			-	1.0	-	MΩ
V <sub>RAM</sub>	RAM hold voltage	During stop mode		1.8	-	-	V	

## NOTE:

- V<sub>CC</sub> = 4.2 to 5.5 V at T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), f(XIN) = 20 MHz, unless otherwise specified.

**Table 22.14 Electrical Characteristics (2) [V<sub>CC</sub> = 5 V]**  
**(T<sub>opr</sub> = –20 to 85°C (N version) / –40 to 85°C (D version), unless otherwise specified.)**

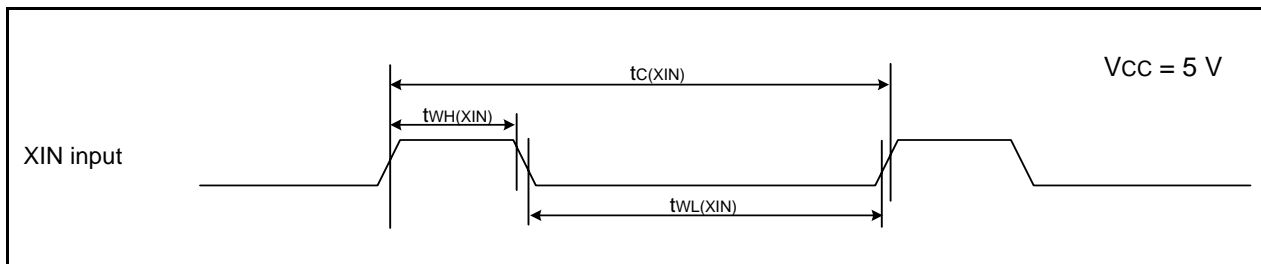
Symbol	Parameter	Condition	Standard			Unit	
			Min.	Typ.	Max.		
I <sub>CC</sub>	Power supply current (V <sub>CC</sub> = 3.3 to 5.5 V) Single-chip mode, output pins are open, other pins are V <sub>SS</sub>	High-speed clock mode	XIN = 20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	10	17	mA
			XIN = 16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	9	15	mA
			XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	6	–	mA
			XIN = 20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	5	–	mA
			XIN = 16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	4	–	mA
			XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	2.5	–	mA
		High-speed on-chip oscillator mode	XIN clock off High-speed on-chip oscillator on f <sub>OCO</sub> = 20 MHz Low-speed on-chip oscillator on = 125 kHz No division	–	10	15	mA
			XIN clock off High-speed on-chip oscillator on f <sub>OCO</sub> = 20 MHz Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	4	–	mA
			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	–	5.5	10	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	–	130	300	μA

**Table 22.15 Electrical Characteristics (3) [V<sub>CC</sub> = 5 V]**  
**(T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.)**

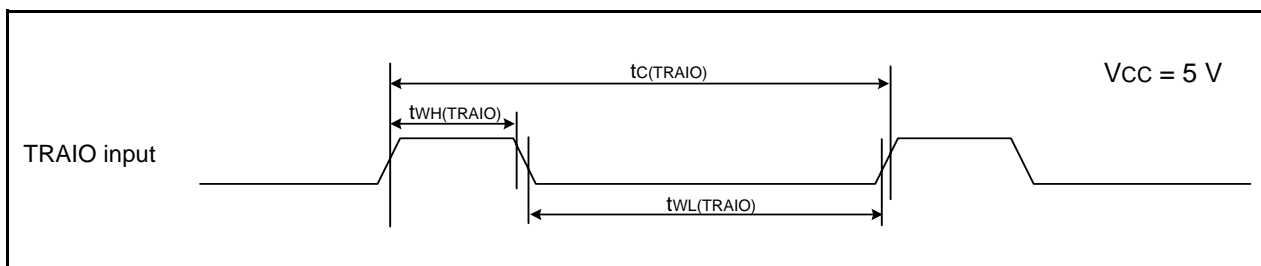
Symbol	Parameter	Condition	Standard			Unit	
			Min.	Typ.	Max.		
I <sub>CC</sub>	Power supply current (V <sub>CC</sub> = 3.3 to 5.5 V) Single-chip mode, output pins are open, other pins are V <sub>SS</sub>	Wait mode	XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock operation VCA27 = VCA26 = VCA25 = 0 VCA20 = 1	-	25	75	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 VCA27 = VCA26 = VCA25 = 0 VCA20 = 1	-	23	60	μA
		Stop mode	XIN clock off, T <sub>opr</sub> = 25°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0	-	0.8	3.0	μA
			XIN clock off, T <sub>opr</sub> = 85°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0	-	1.2	-	μA

**Timing Requirements****(Unless Otherwise Specified:  $V_{CC} = 5\text{ V}$ ,  $V_{SS} = 0\text{ V}$  at  $T_{opr} = 25^\circ\text{C}$ ) [ $V_{CC} = 5\text{ V}$ ]****Table 22.16 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 22.4 XIN Input Timing Diagram when  $V_{CC} = 5\text{ V}$** **Table 22.17 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

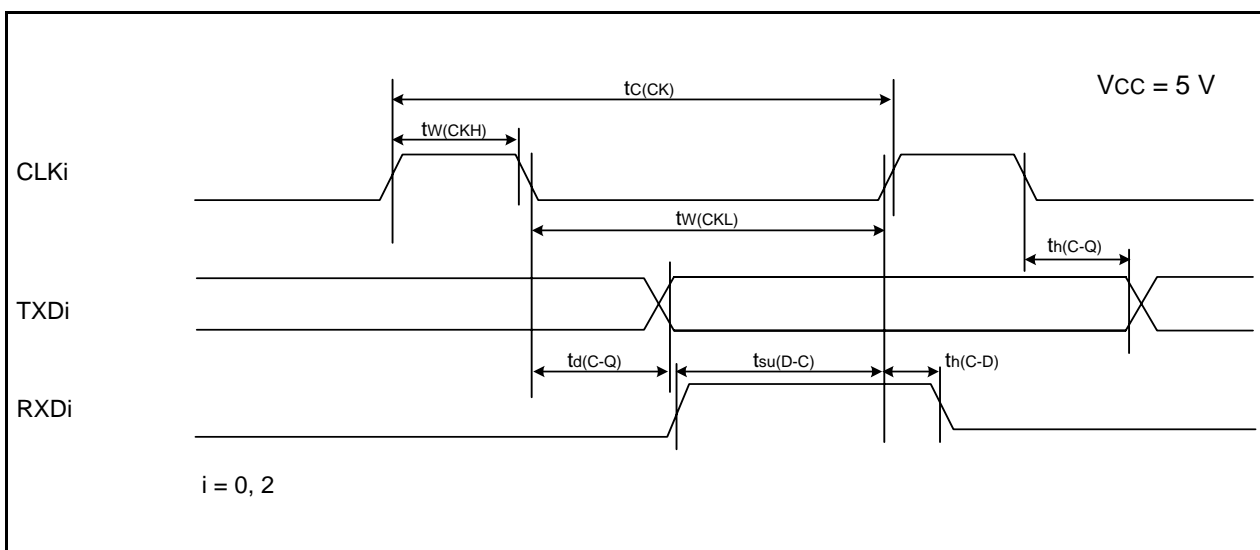
**Figure 22.5 TRAIO Input Timing Diagram when  $V_{CC} = 5\text{ V}$**



**Table 22.18 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200	–	ns
$t_{w(CKH)}$	CLKi input “H” width	100	–	ns
$t_{w(CKL)}$	CLKi input “L” width	100	–	ns
$t_{d(C-Q)}$	TXDi output delay time	–	50	ns
$t_{h(C-Q)}$	TXDi hold time	0	–	ns
$t_{su(D-C)}$	RXDi input setup time	50	–	ns
$t_{h(C-D)}$	RXDi input hold time	90	–	ns

$i = 0, 2$



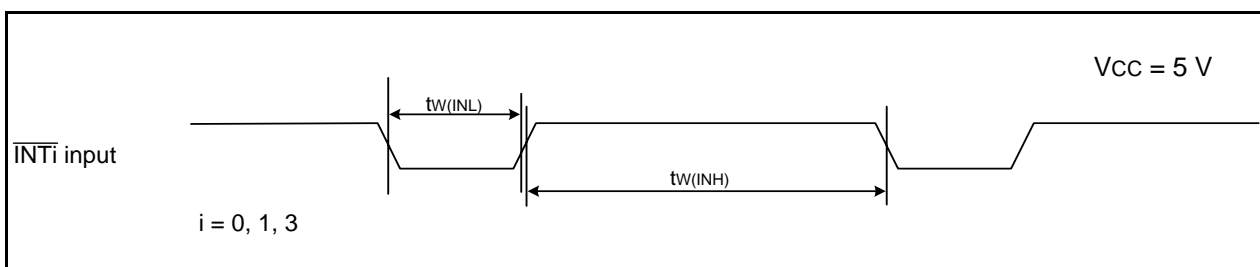
**Figure 22.6 Serial Interface Timing Diagram when Vcc = 5 V**

**Table 22.19 External Interrupt  $\overline{INTi}$  ( $i = 0, 1, 3$ ) 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:

1. When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use an  $\overline{INTi}$  input HIGH width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.
2. When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use an  $\overline{INTi}$  input LOW width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.



**Figure 22.7 External Interrupt  $\overline{INTi}$  Input Timing Diagram when Vcc = 5 V**

**Table 22.20 Electrical Characteristics (1) [V<sub>CC</sub> = 3 V]**

Symbol	Parameter		Condition		Standard			Unit
					Min.	Typ.	Max.	
V <sub>OH</sub>	Output "H" voltage	Except P2_0 to P2_7, XOUT	I <sub>OH</sub> = -1 mA		V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
		P2_0 to P2_7	Drive capacity HIGH	I <sub>OH</sub> = -5 mA	V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
			Drive capacity LOW	I <sub>OH</sub> = -1 mA	V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
		XOUT	Drive capacity HIGH	I <sub>OH</sub> = -0.1 mA	V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
			Drive capacity LOW	I <sub>OH</sub> = -50 μA	V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
V <sub>OL</sub>	Output "L" voltage	Except P2_0 to P2_7, XOUT	I <sub>OL</sub> = 1 mA		-	-	0.5	V
		P2_0 to P2_7	Drive capacity HIGH	I <sub>OL</sub> = 5 mA	-	-	0.5	V
			Drive capacity LOW	I <sub>OL</sub> = 1 mA	-	-	0.5	V
		XOUT	Drive capacity HIGH	I <sub>OL</sub> = 0.1 mA	-	-	0.5	V
			Drive capacity LOW	I <sub>OL</sub> = 50 μA	-	-	0.5	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	INT0, INT1, INT3, KI0, KI1, KI2, KI3, TRAIO, RXD0, RXD2, CLK0, CLK2			0.1	0.3	-	V
		RESET			0.1	0.4	-	V
I <sub>IH</sub>	Input "H" current	V <sub>I</sub> = 3 V, V <sub>CC</sub> = 3 V		-	-	4.0	μA	
I <sub>IL</sub>	Input "L" current	V <sub>I</sub> = 0 V, V <sub>CC</sub> = 3 V		-	-	-4.0	μA	
R <sub>PULLUP</sub>	Pull-up resistance	V <sub>I</sub> = 0 V, V <sub>CC</sub> = 3 V		66	160	500	kΩ	
R <sub>IXIN</sub>	Feedback resistance	XIN			-	3.0	-	MΩ
V <sub>RAM</sub>	RAM hold voltage	During stop mode		1.8	-	-	V	

## NOTE:

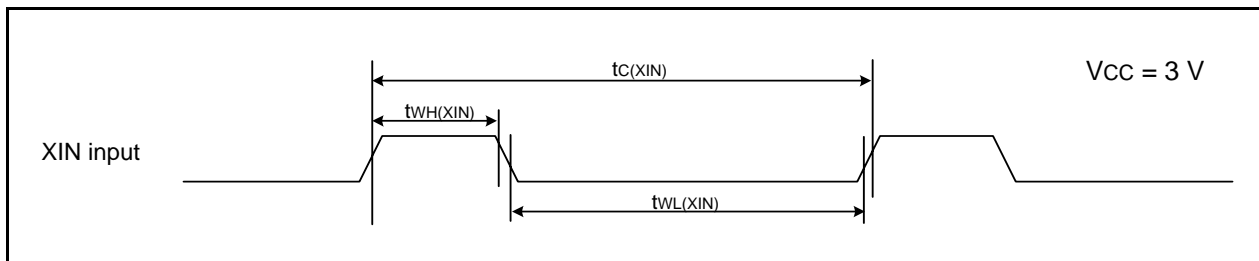
- V<sub>CC</sub> = 2.7 to 3.3 V at T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), f(XIN) = 10 MHz, unless otherwise specified.

**Table 22.21 Electrical Characteristics (2) [ $V_{CC} = 3\text{ V}$ ]  
( $T_{opr} = -20\text{ to }85^{\circ}\text{C}$  (N version) /  $-40\text{ to }85^{\circ}\text{C}$  (D version), unless otherwise specified.)**

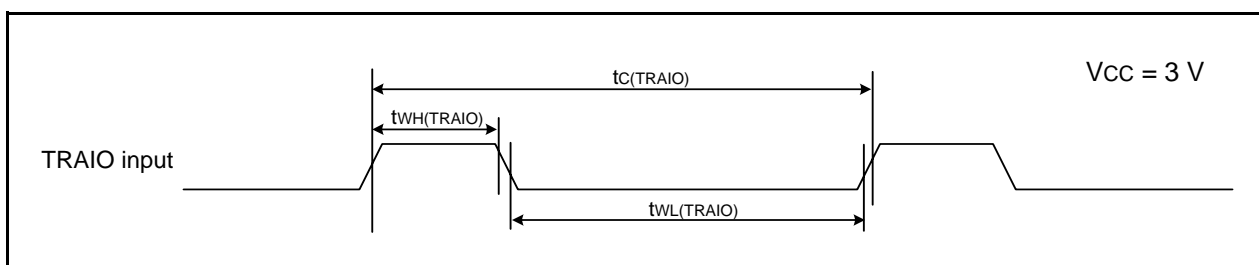
Symbol	Parameter	Condition	Standard			Unit	
			Min.	Typ.	Max.		
I <sub>CC</sub>	Power supply current ( $V_{CC} = 2.7\text{ to }3.3\text{ V}$ ) Single-chip mode, output pins are open, other pins are $V_{SS}$	High-speed clock mode	XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	6	–	mA
			XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	2	–	mA
		High-speed on-chip oscillator mode	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	9	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	–	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	–	130	300	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 VCA27 = VCA26 = VCA25 = 0 VCA20 = 1	–	25	70	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 VCA27 = VCA26 = VCA25 = 0 VCA20 = 1	–	23	55	μA
		Stop mode	XIN clock off, $T_{opr} = 25^{\circ}\text{C}$ High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0	–	0.7	3.0	μA
			XIN clock off, $T_{opr} = 85^{\circ}\text{C}$ High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0	–	1.1	–	μA

**Timing requirements****(Unless Otherwise Specified:  $V_{CC} = 3\text{ V}$ ,  $V_{SS} = 0\text{ V}$  at  $T_{opr} = 25^{\circ}\text{C}$ ) [ $V_{CC} = 3\text{ V}$ ]****Table 22.22 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 22.8 XIN Input Timing Diagram when  $V_{CC} = 3\text{ V}$** **Table 22.23 TRAIO Input**

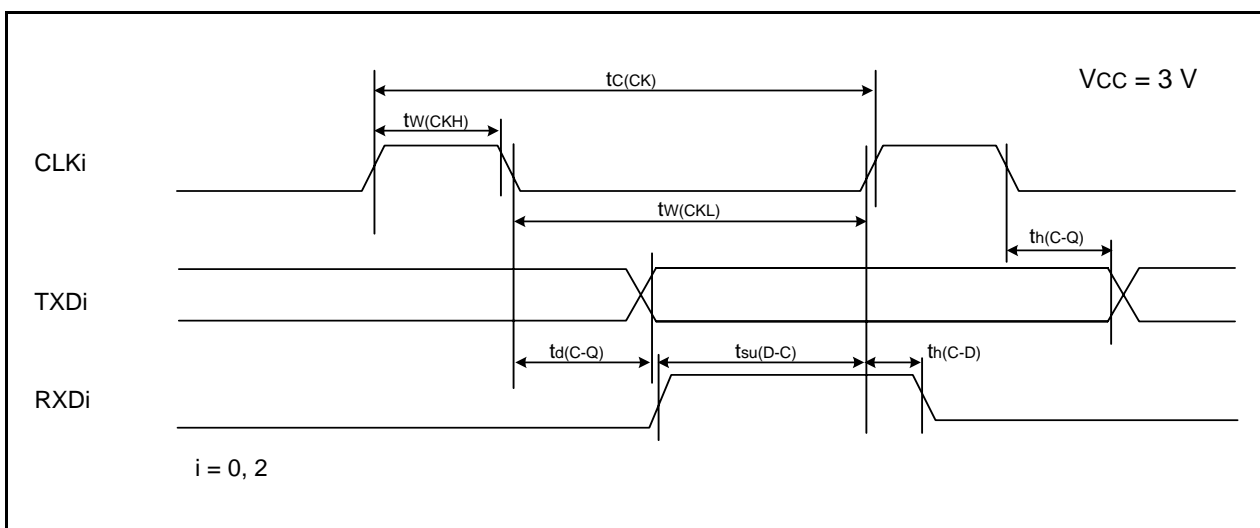
Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TRAIO)}$	TRAIO input cycle time	300	–	ns
$t_{WH(TRAIO)}$	TRAIO input "H" width	120	–	ns
$t_{WL(TRAIO)}$	TRAIO input "L" width	120	–	ns

**Figure 22.9 TRAIO Input Timing Diagram when  $V_{CC} = 3\text{ V}$**

**Table 22.24 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300	–	ns
$t_{w(CKH)}$	CLKi input “H” width	150	–	ns
$t_{w(CKL)}$	CLKi Input “L” width	150	–	ns
$t_{d(C-Q)}$	TXDi output delay time	–	80	ns
$t_{h(C-Q)}$	TXDi hold time	0	–	ns
$t_{su(D-C)}$	RXDi input setup time	70	–	ns
$t_{h(C-D)}$	RXDi input hold time	90	–	ns

$i = 0, 2$



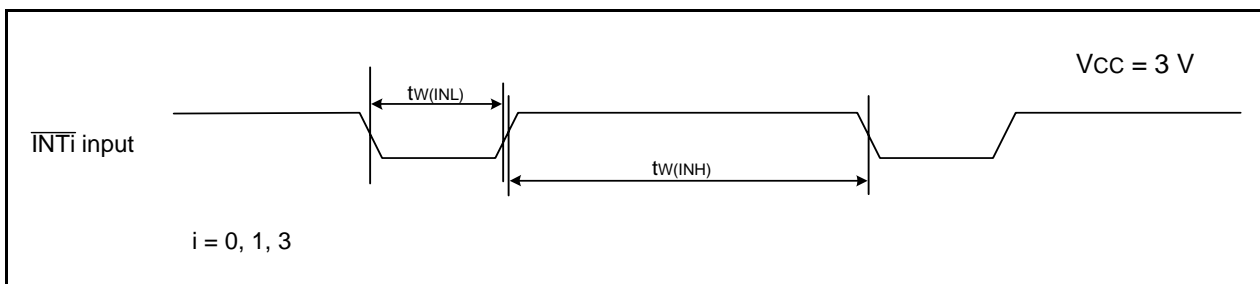
**Figure 22.10 Serial Interface Timing Diagram when Vcc = 3 V**

**Table 22.25 External Interrupt  $\overline{INTi}$  ( $i = 0, 1, 3$ ) 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:

1. When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use an  $\overline{INTi}$  input HIGH width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.
2. When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use an  $\overline{INTi}$  input LOW width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.



**Figure 22.11 External Interrupt  $\overline{INTi}$  Input Timing Diagram when Vcc = 3 V**

**Table 22.26 Electrical Characteristics (1) [V<sub>CC</sub> = 2.2 V]**

Symbol	Parameter		Condition		Standard			Unit
					Min.	Typ.	Max.	
V <sub>OH</sub>	Output "H" voltage	Except P2_0 to P2_7, XOUT	I <sub>OH</sub> = -1 mA		V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
		P2_0 to P2_7	Drive capacity HIGH	I <sub>OH</sub> = -2 mA	V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
			Drive capacity LOW	I <sub>OH</sub> = -1 mA	V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
		XOUT	Drive capacity HIGH	I <sub>OH</sub> = -0.1 mA	V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
			Drive capacity LOW	I <sub>OH</sub> = -50 μA	V <sub>CC</sub> - 0.5	-	V <sub>CC</sub>	V
V <sub>OL</sub>	Output "L" voltage	Except P2_0 to P2_7, XOUT	I <sub>OL</sub> = 1 mA		-	-	0.5	V
		P2_0 to P2_7	Drive capacity HIGH	I <sub>OL</sub> = 2 mA	-	-	0.5	V
			Drive capacity LOW	I <sub>OL</sub> = 1 mA	-	-	0.5	V
		XOUT	Drive capacity HIGH	I <sub>OL</sub> = 0.1 mA	-	-	0.5	V
			Drive capacity LOW	I <sub>OL</sub> = 50 μA	-	-	0.5	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	$\overline{\text{INT0}}, \overline{\text{INT1}}, \overline{\text{INT3}}, \overline{\text{KI0}}, \overline{\text{KI1}}, \overline{\text{KI2}}, \overline{\text{KI3}}, \overline{\text{TRAIO}}, \overline{\text{RXD0}}, \overline{\text{RXD2}}, \overline{\text{CLK0}}, \overline{\text{CLK2}}$			0.05	0.3	-	V
		$\overline{\text{RESET}}$			0.05	0.15	-	V
I <sub>IH</sub>	Input "H" current		V <sub>I</sub> = 2.2 V		-	-	4.0	μA
I <sub>IL</sub>	Input "L" current		V <sub>I</sub> = 0 V		-	-	-4.0	μA
R <sub>PULLUP</sub>	Pull-up resistance		V <sub>I</sub> = 0 V		100	200	600	kΩ
R <sub>IXIN</sub>	Feedback resistance	XIN			-	5	-	MΩ
V <sub>RAM</sub>	RAM hold voltage		During stop mode		1.8	-	-	V

## NOTE:

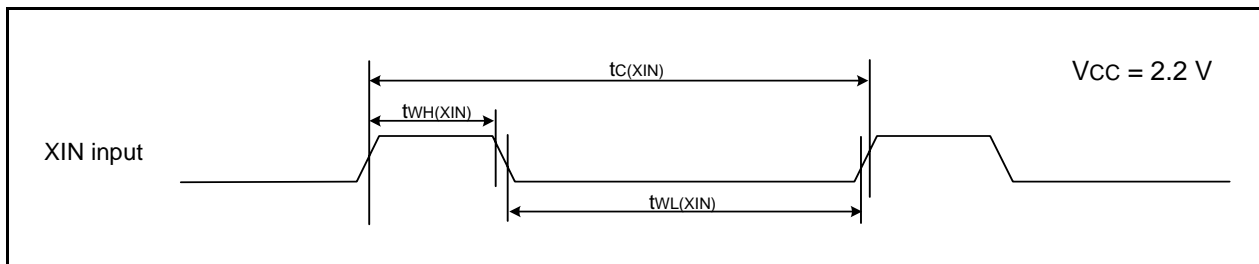
- V<sub>CC</sub> = 2.2 V at T<sub>opr</sub> = -20 to 85°C (N version) / -40 to 85°C (D version), f(XIN) = 5 MHz, unless otherwise specified.

**Table 22.27 Electrical Characteristics (2) [ $V_{CC} = 2.2\text{ V}$ ]  
( $T_{opr} = -20\text{ to }85^{\circ}\text{C}$  (N version) /  $-40\text{ to }85^{\circ}\text{C}$  (D version), unless otherwise specified.)**

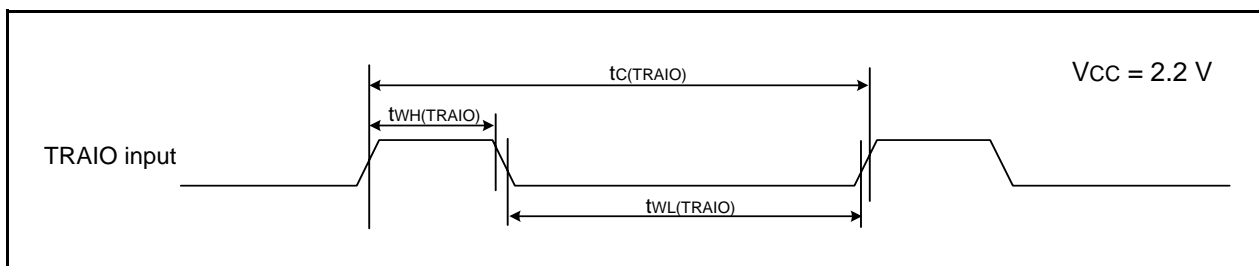
Symbol	Parameter	Condition	Standard			Unit	
			Min.	Typ.	Max.		
I <sub>CC</sub>	Power supply current ( $V_{CC} = 2.2\text{ to }2.7\text{ V}$ ) Single-chip mode, output pins are open, other pins are $V_{SS}$	High-speed clock mode	XIN = 5 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division	–	3.5	–	mA
			XIN = 5 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	1.5	–	mA
		High-speed on-chip oscillator mode	XIN clock off High-speed on-chip oscillator on f <sub>OCO</sub> = 5 MHz Low-speed on-chip oscillator on = 125 kHz No division	–	3.5	–	mA
			XIN clock off High-speed on-chip oscillator on f <sub>OCO</sub> = 5 MHz Low-speed on-chip oscillator on = 125 kHz Divide-by-8	–	1.5	–	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	–	100	230	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 VCA27 = VCA26 = VCA25 = 0 VCA20 = 1	–	22	60	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 VCA27 = VCA26 = VCA25 = 0 VCA20 = 1	–	20	55	μA
		Stop mode	XIN clock off, $T_{opr} = 25^{\circ}\text{C}$ High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0	–	0.7	3.0	μA
			XIN clock off, $T_{opr} = 85^{\circ}\text{C}$ High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0	–	1.1	–	μA

**Timing requirements****(Unless Otherwise Specified:  $V_{CC} = 2.2\text{ V}$ ,  $V_{SS} = 0\text{ V}$  at  $T_{opr} = 25^{\circ}\text{C}$ ) [ $V_{CC} = 2.2\text{ V}$ ]****Table 22.28 XIN Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(XIN)}$	XIN input cycle time	200	–	ns
$t_{WH(XIN)}$	XIN input "H" width	90	–	ns
$t_{WL(XIN)}$	XIN input "L" width	90	–	ns

**Figure 22.12 XIN Input Timing Diagram when  $V_{CC} = 2.2\text{ V}$** **Table 22.29 TRAI0 Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TRAIO)}$	TRAIO input cycle time	500	–	ns
$t_{WH(TRAIO)}$	TRAIO input "H" width	200	–	ns
$t_{WL(TRAIO)}$	TRAIO input "L" width	200	–	ns

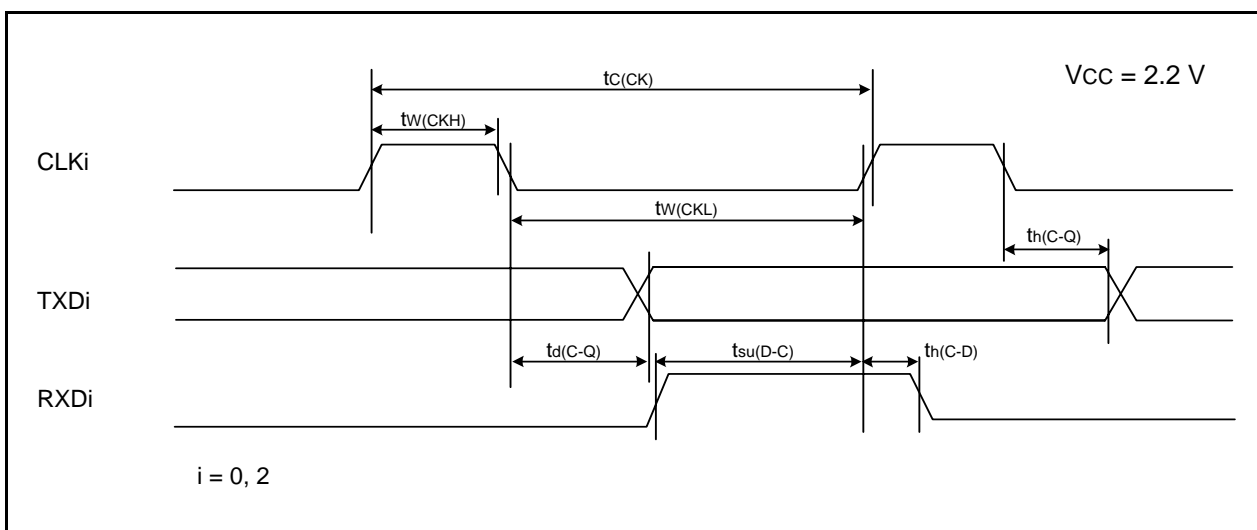
**Figure 22.13 TRAI0 Input Timing Diagram when  $V_{CC} = 2.2\text{ V}$**



**Table 22.30 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	800	–	ns
$t_{w(CKH)}$	CLKi input “H” width	400	–	ns
$t_{w(CKL)}$	CLKi input “L” width	400	–	ns
$t_{d(C-Q)}$	TXDi output delay time	–	200	ns
$t_{h(C-Q)}$	TXDi hold time	0	–	ns
$t_{su(D-C)}$	RXDi input setup time	150	–	ns
$t_{h(C-D)}$	RXDi input hold time	90	–	ns

$i = 0, 2$



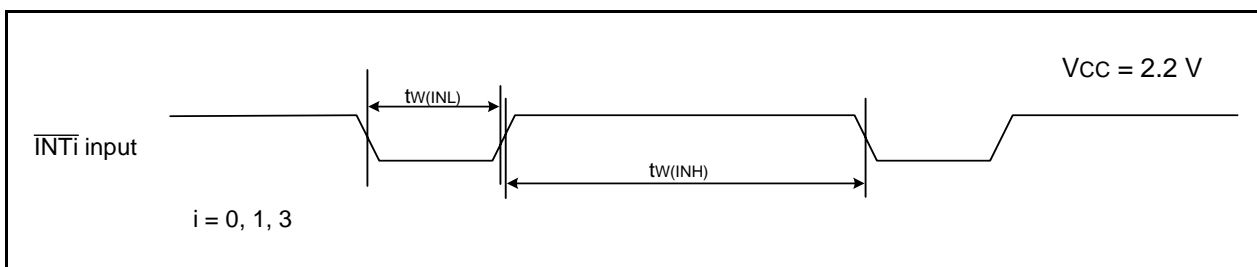
**Figure 22.14 Serial Interface Timing Diagram when Vcc = 2.2 V**

**Table 22.31 External Interrupt  $\overline{INTi}$  ( $i = 0, 1, 3$ ) Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input “H” width	1000 <sup>(1)</sup>	–	ns
$t_{w(INL)}$	$\overline{INTi}$ input “L” width	1000 <sup>(2)</sup>	–	ns

NOTES:

1. When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use an  $\overline{INTi}$  input HIGH width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.
2. When selecting the digital filter by the  $\overline{INTi}$  input filter select bit, use an  $\overline{INTi}$  input LOW width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.



**Figure 22.15 External Interrupt  $\overline{INTi}$  Input Timing Diagram when Vcc = 2.2 V**

## 23. Usage Notes

### 23.1 Notes on Clock Generation Circuit

#### 23.1.1 Stop Mode

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

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

- Program example to enter stop mode

```

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

```

#### 23.1.2 Wait Mode

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

- Program example to execute the WAIT instruction

```

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

```

#### 23.1.3 Oscillation Stop Detection Function

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

#### 23.1.4 Oscillation Circuit Constants

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

To use this MCU with supply voltage below  $VCC = 2.7$  V, it is recommended to set the CM11 bit in the CM1 register to 1 (on-chip feedback resistor disabled), the CM15 bit to 1 (high drive capacity), and connect the feedback resistor to the chip externally.

## 23.2 Notes on Interrupts

### 23.2.1 Reading Address 00000h

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

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

### 23.2.2 SP Setting

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

### 23.2.3 External Interrupt and Key Input Interrupt

Either “L” level or an “H” level of width shown in the Electrical Characteristics is necessary for the signal input to pins  $\overline{\text{INT0}}$ ,  $\overline{\text{INT1}}$ ,  $\overline{\text{INT3}}$  and pins  $\overline{\text{KI0}}$  to  $\overline{\text{KI3}}$ , regardless of the CPU clock.

For details, refer to **Table 22.19** (VCC = 5V), **Table 22.25** (VCC = 3V), **Table 22.31** (VCC = 2.2V) **External Interrupt INTi (i = 0, 1, 3) Input**.

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

Figure 23.1 shows an Example of Procedure for Changing Interrupt Sources.

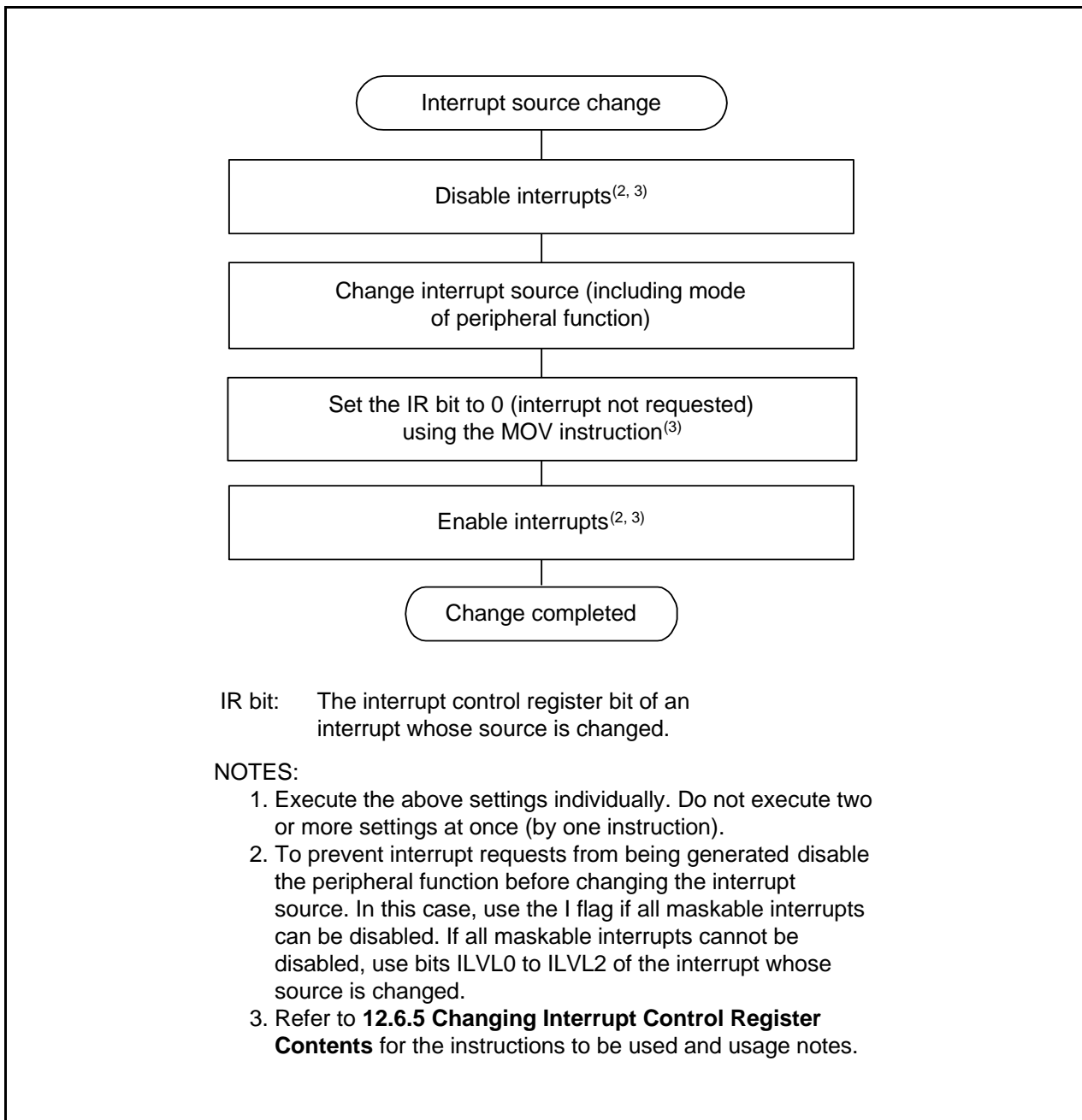


Figure 23.1 Example of Procedure for Changing Interrupt Sources

### 23.2.5 Changing Interrupt Control Register Contents

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

#### Changing any bit other than IR bit

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

#### Changing IR bit

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

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

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

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

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

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

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

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

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

## 23.3 Notes on Timers

### 23.3.1 Notes on Timer RA

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

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

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

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

NOTE:

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

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

### 23.3.2 Notes on Timer RB

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

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

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

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

#### NOTE:

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

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

#### 23.3.2.1 Timer mode

The following workaround should be performed in timer mode.

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

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

### 23.3.2.2 Programmable waveform generation mode

The following three workarounds should be performed in programmable waveform 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) To change registers TRBPRES and TRBPR during count operation (TCSTF bit is set to 1), synchronize the TRBO output cycle using a timer RB interrupt, etc. This operation should be performed only once in the same output cycle. Also, make sure that writing to the TRBPR register does not occur during period A shown in Figures 23.2 and 23.3.

The following shows the detailed workaround examples.

- Workaround example (a):

As shown in Figure 23.2, write to registers TRBSC and TRBPR in the timer RB interrupt routine. These write operations must be completed by the beginning of period A.

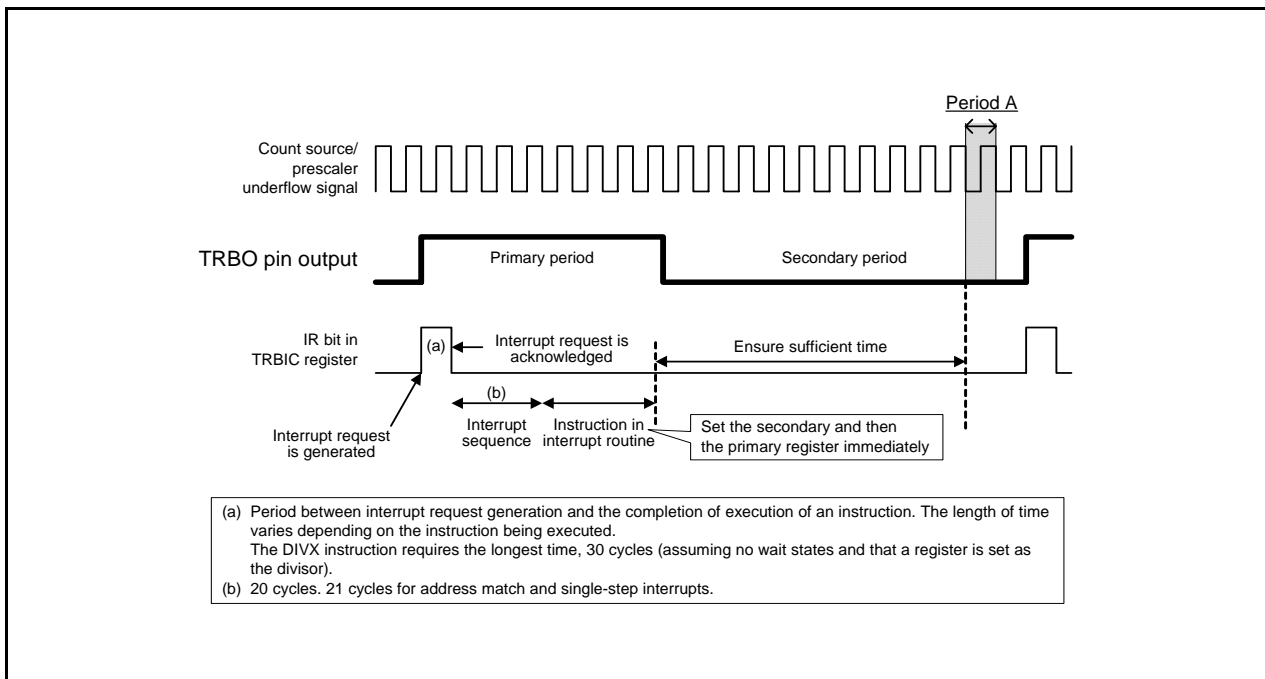


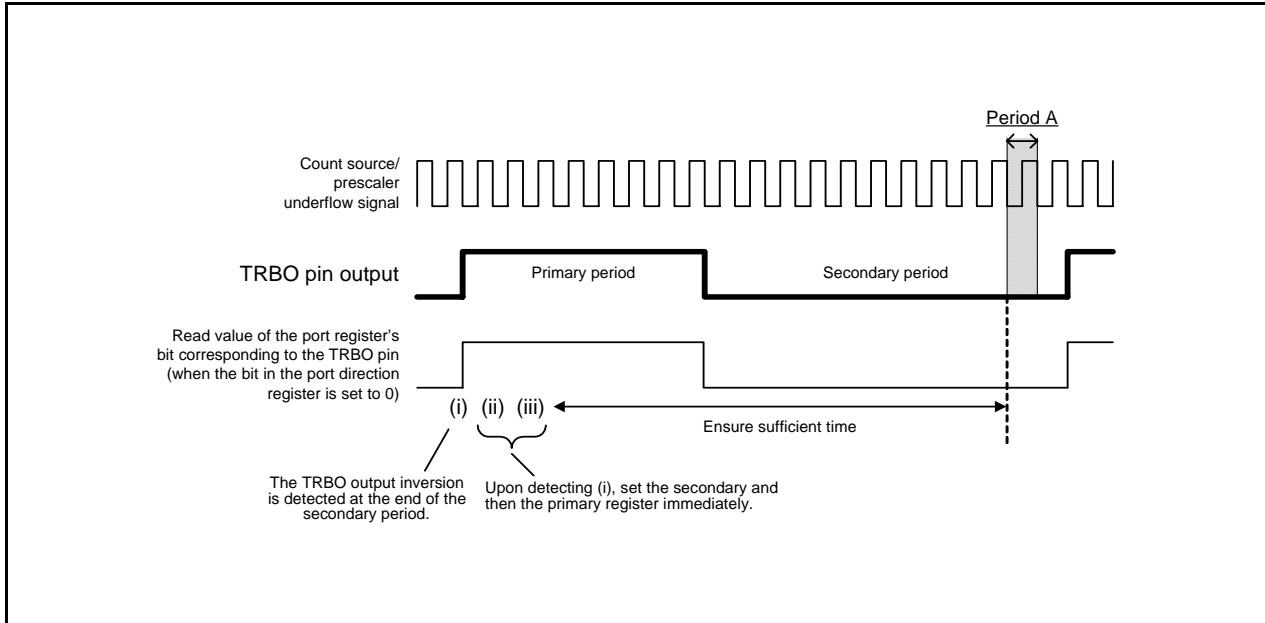
Figure 23.2 Workaround Example (a) When Timer RB interrupt is Used



- Workaround example (b):

As shown in Figure 23.3 detect the start of the primary period by the TRBO pin output level and write to registers TRBSC and TRBPR. These write operations must be completed by the beginning of period A.

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



**Figure 23.3 Workaround Example (b) When TRBO Pin Output Value is Read**

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

### 23.3.2.3 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.

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

### 23.3.3 Notes on Timer RC

#### 23.3.3.1 TRC Register

- The following note applies when the CCLR bit in the TRCCR1 register is set to 1 (clear TRC register at compare match with TRCGRA register).

When using a program to write a value to the TRC register while the TSTART bit in the TRCMR register is set to 1 (count starts), ensure that the write does not overlap with the timing with which the TRC register is set to 0000h.

If the timing of the write to the TRC register and the setting of the TRC register to 0000h coincide, the write value will not be written to the TRC register and the TRC register will be set to 0000h.

- Reading from the TRC register immediately after writing to it can result in the value previous to the write being read out. To prevent this, execute the JMP.B instruction between the read and the write instructions.

```

Program Example      MOV.W      #XXXXh, TRC          ;Write
                    JMP.B      L1              ;JMP.B instruction
                    L1:        MOV.W      TRC,DATA      ;Read
  
```

#### 23.3.3.2 TRCSR Register

Reading from the TRCSR register immediately after writing to it can result in the value previous to the write being read out. To prevent this, execute the JMP.B instruction between the read and the write instructions.

```

Program Example      MOV.B      #XXh, TRCSR        ;Write
                    JMP.B      L1              ;JMP.B instruction
                    L1:        MOV.B      TRCSR,DATA ;Read
  
```

#### 23.3.3.3 Count Source Switching

- Stop the count before switching the count source.

Switching procedure

- Set the TSTART bit in the TRCMR register to 0 (count stops).
- Change the settings of bits TCK2 to TCK0 in the TRCCR1 register.

- After switching the count source from fOCO40M to another clock, allow a minimum of two cycles of f1 to elapse after changing the clock setting before stopping fOCO40M.

Switching procedure

- Set the TSTART bit in the TRCMR register to 0 (count stops).
- Change the settings of bits TCK2 to TCK0 in the TRCCR1 register.
- Wait for a minimum of two cycles of f1.
- Set the FRA00 bit in the FRA0 register to 0 (high-speed on-chip oscillator off).

#### 23.3.3.4 Input Capture Function

- The pulse width of the input capture signal should be three cycles or more of the timer RC operation clock (refer to **Table 16.11 Timer RC Operation Clock**).
- The value of the TRC register is transferred to the TRCGRj register one or two cycles of the timer RC operation clock after the input capture signal is input to the TRCIOj (j = A, B, C, or D) pin (when the digital filter function is not used).

#### 23.3.3.5 TRCMR Register in PWM2 Mode

When the CSEL bit in the TRCCR2 register is set to 1 (count stops at compare match with the TRCGRA register), do not set the TRCMR register at compare match timing of registers TRC and TRCGRA.

## 23.3.4 Notes on Timer RD

### 23.3.4.1 TRDSTR Register

- Set the TRDSTR register using the MOV instruction.
- When the CSELi ( $i = 0$  to  $1$ ) 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 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 23.1 lists the TRDIOji ( $j = A, B, C, \text{ or } D$ ) Pin Output Level when Count Stops to use the TRDIOji ( $j = A, B, C, \text{ or } D$ ) pin with the timer RD output.

**Table 23.1 TRDIOji ( $j = A, B, C, \text{ or } D$ ) Pin Output Level when Count Stops**

Count Stop	TRDIOji 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.

### 23.3.4.2 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 overlapping with the timing for setting the TRDi register to  $0000h$ , and then write. If the timing for setting the TRDi register to  $0000h$  overlaps with the timing for writing 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 bits CCLR2 to CCLR0 in the TRDCRi register.
  - $001b$  (Clear by the TRDi register at compare match with the TRDGRAi register.)
  - $010b$  (Clear by the TRDi register at compare match with the TRDGRBi register.)
  - $011b$  (Synchronous clear)
  - $101b$  (Clear by the TRDi register at compare match with the TRDGRCi register.)
  - $110b$  (Clear by the TRDi register at 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
  
```

### 23.3.4.3 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
  
```

### 23.3.4.4 Count Source Switch

- Switch the count source 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 bits TCK2 to TCK0 in the TRDCR<sub>i</sub> register.

- When changing the count source from fOCO40M to another source and stopping fOCO40M, wait 2 cycles of f1 or more 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 bits TCK2 to TCK0 in the TRDCR<sub>i</sub> register.
- (3) Wait 2 or more cycles of f1.
- (4) Set the FRA00 bit in the FRA0 register to 0 (high-speed on-chip oscillator stops).

### 23.3.4.5 Input Capture Function

- Set the pulse width of the input capture signal to 3 or more cycles of the timer RD operation clock (refer to **Table 16.25 Timer RD Operation Clocks**).
- The value in the TRD<sub>i</sub> register is transferred to the TRDGR<sub>ji</sub> register 2 to 3 cycles of the timer RD operation clock after 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).

### 23.3.4.6 Reset Synchronous PWM Mode

- When reset synchronous PWM mode is used for motor control, make sure OLS0 = OLS1.
- Set to reset synchronous PWM mode by the following procedure:

Change procedure

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

### 23.3.4.7 Complementary PWM Mode

- When complementary PWM mode is used for motor control, make sure OLS0 = OLS1.
- Change bits CMD1 to CMD0 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 bits CMD1 to CMD0 in the TRDFCR register to 00b (timer mode, PWM mode, and PWM3 mode).
- (3) Set bits CMD1 to CMD0 to 10b or 11b (complementary PWM mode).
- (4) Set the registers associated with other timer RD again.

Change procedure: When stopping complementary PWM mode

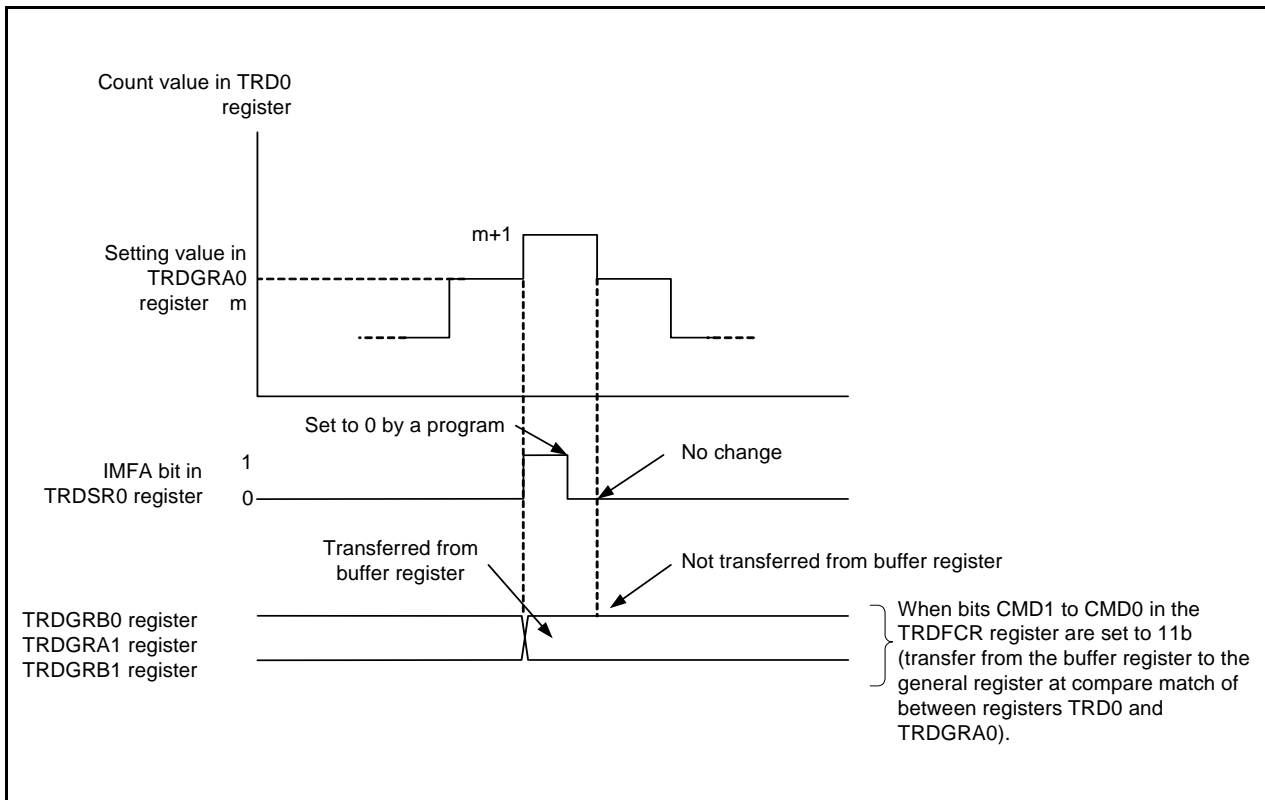
- (1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
- (2) Set bits CMD1 to CMD0 to 00b (timer mode, PWM mode, and PWM3 mode).

- Do not write to TRDGRA0, TRDGRB0, TRDGRA1, or TRDGRB1 register during operation. When changing the PWM waveform, transfer the values written to registers TRDGRD0, TRDGRC1, and TRDGRD1 to registers TRDGRB0, TRDGRA1, and TRDGRB1 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.

- If the value in the TRDGRA0 register is assumed to be  $m$ , the TRD0 register counts  $m-1$ ,  $m$ ,  $m+1$ ,  $m$ ,  $m-1$ , in that order, when changing from increment to decrement operation.

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

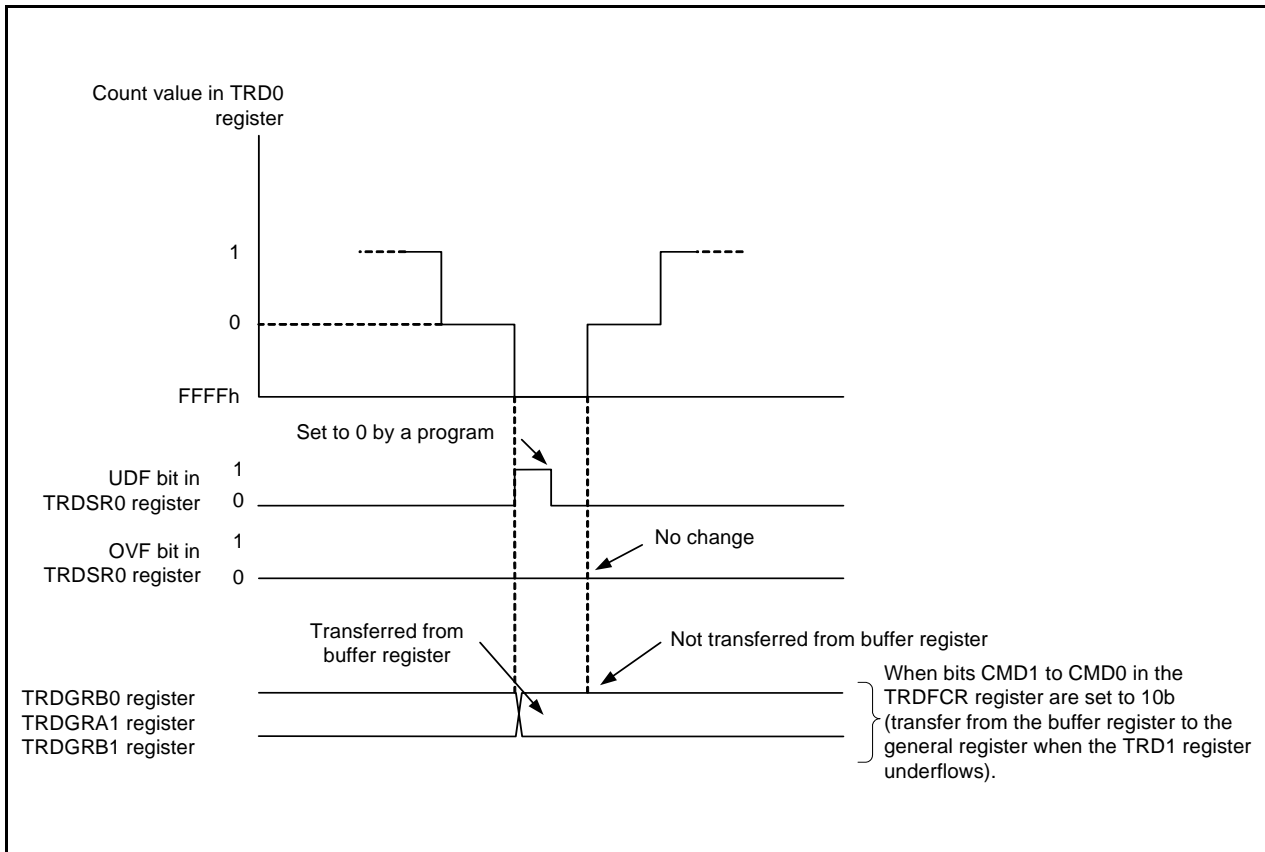
During  $m+1$ ,  $m$ , and  $m-1$  operation, the IMFA bit remains unchanged and data are not transferred to registers such as the TRDGRA0 register.



**Figure 23.4 Operation at Compare Match between Registers TRD0 and TRDGRA0 in Complementary PWM Mode**

- The TRD1 register counts 1, 0, FFFFh, 0, 1, in that order, when changing from decrement to increment operation.

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



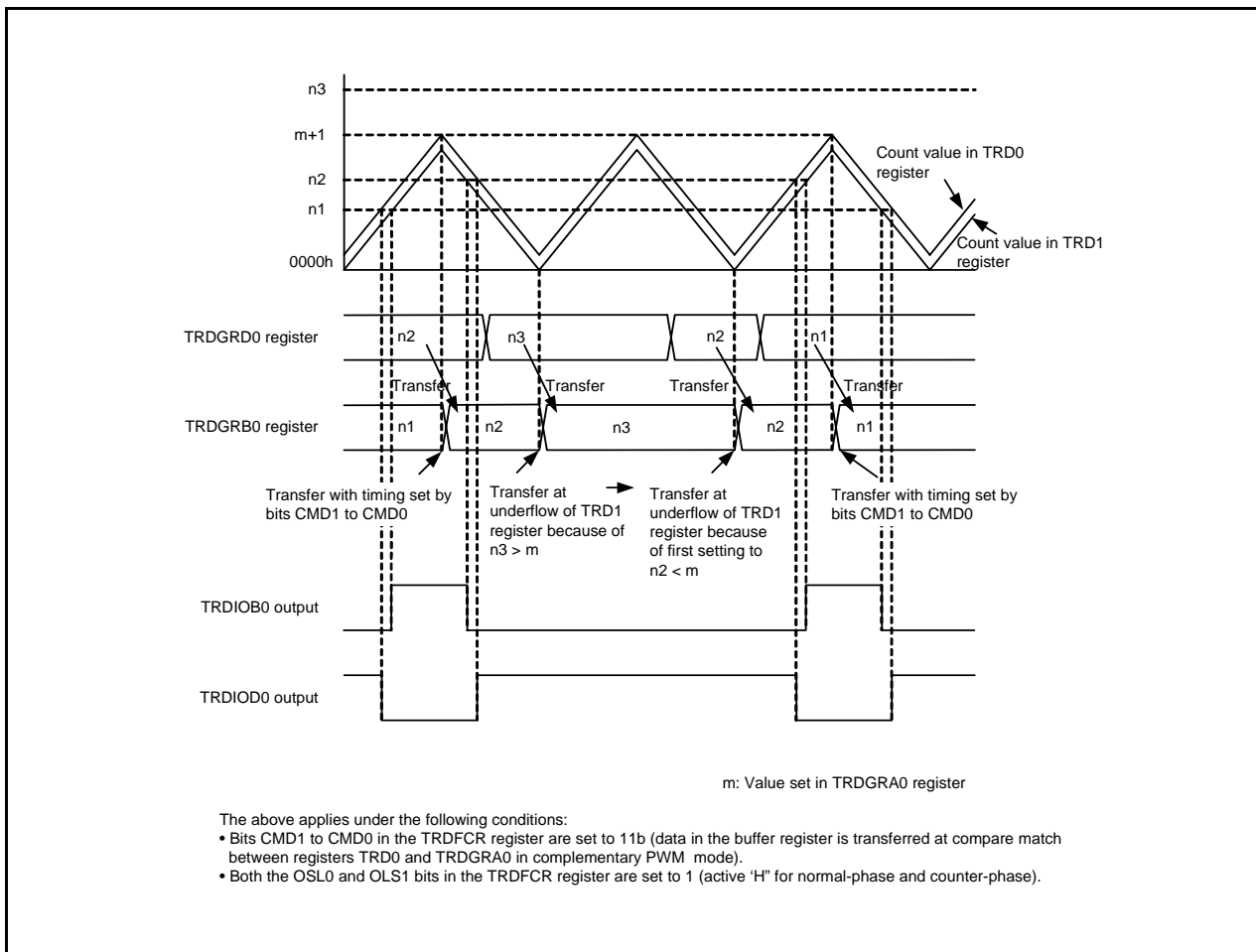
**Figure 23.5 Operation when TRD1 Register Underflows in Complementary PWM Mode**

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

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

Transfer take place at underflow of the TRD1 register.

After this, when the buffer register is set to 0001h or above and a smaller value than the value of the TRDGRA0 register, and the TRD1 register underflows for the first time after setting, the value is transferred to the general register. After that, the value is transferred with the timing selected by bits CMD1 to CMD0.



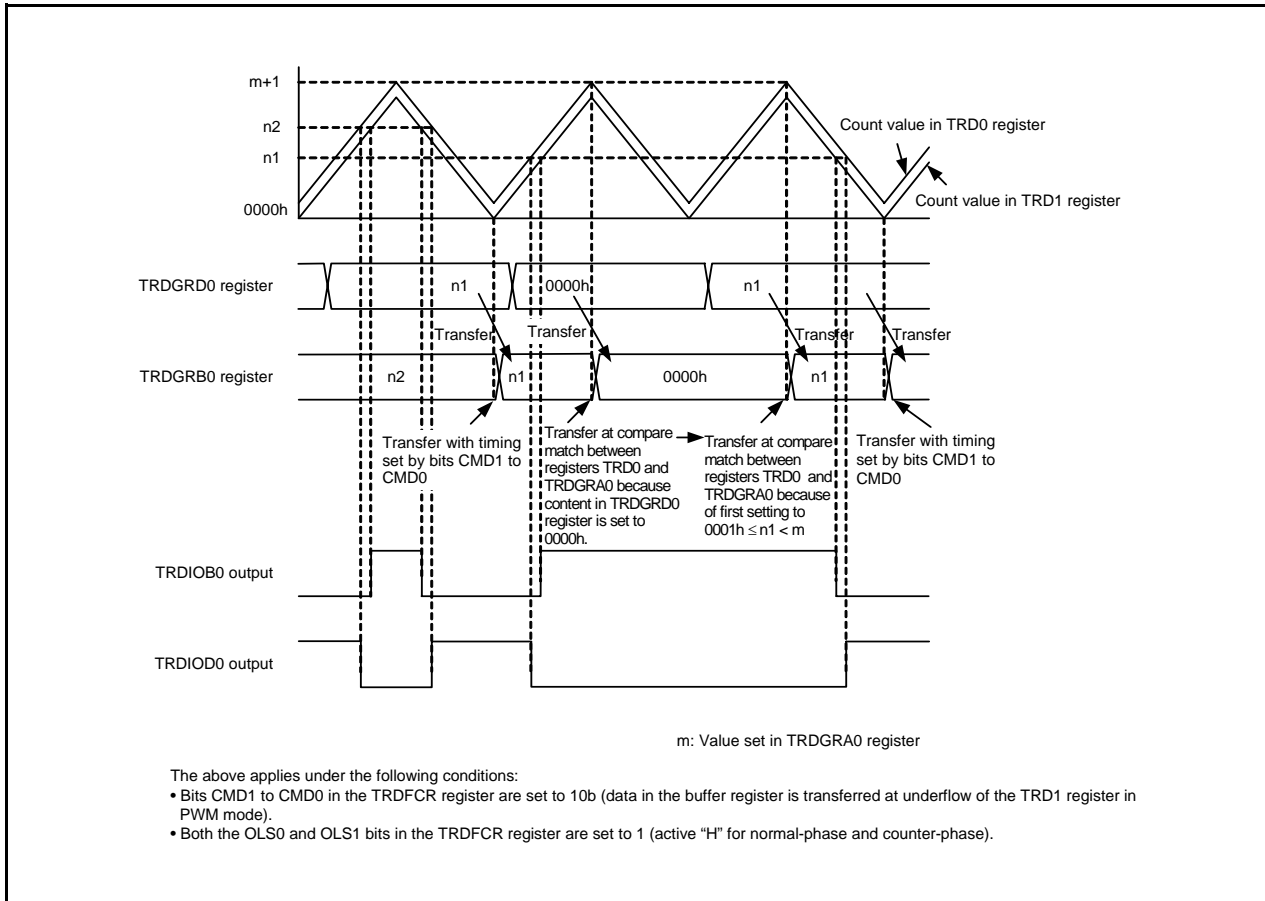
**Figure 23.6** Operation when Value in Buffer Register  $\geq$  Value in TRDGRA0 Register in Complementary PWM Mode



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

Transfer takes place at compare match between registers TRD0 and TRDGRA0.

After this, when the buffer register is set to 0001h or above and a smaller value than the value of the TRDGRA0 register, and a compare match occurs between registers TRD0 and TRDGRA0 for the first time after setting, the value is transferred to the general register. After that, the value is transferred with the timing selected by bits CMD1 to CMD0.



**Figure 23.7** Operation when Value in Buffer Register Is Set to 0000h in Complementary PWM Mode

### 23.3.4.8 Count Source fOCO40M

- The count source fOCO40M can be used with supply voltage  $V_{CC} = 3.0$  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).

## 23.4 Notes on Serial Interface

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

Example (when reading receive buffer register):

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

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

Example (when reading transmit buffer register):

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

### 23.5 Notes on Hardware LIN

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

## 23.6 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 A/D conversion is stopped (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 before starting the A/D conversion.
- After changing the A/D operating mode, select an analog input pin again.
- When using the one-shot mode, ensure that A/D conversion is completed before reading the AD register. The IR bit in the ADIC register or the ADST bit in the ADCON0 register can be used to determine whether 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.
- If the ADST bit in the ADCON0 register is set to 0 (A/D conversion stops) by a program and A/D conversion is forcibly terminated during an A/D conversion operation, the conversion result of the A/D converter will be undefined. If the ADST bit is set to 0 by a program, do not use the value of the 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.

## 23.7 Notes on Flash Memory

### 23.7.1 CPU Rewrite Mode

#### 23.7.1.1 Operating Speed

Before entering CPU rewrite mode (EW0 mode), select 5 MHz or below for the CPU clock using the CM06 bit in the CM0 register and bits CM16 to CM17 in the CM1 register. This does not apply to EW1 mode.

#### 23.7.1.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because they reference data in the flash memory: UND, INTO, and BRK.

#### 23.7.1.3 Non-Maskable Interrupts

- EW0 Mode

Once a watchdog timer, oscillation stop detection, voltage monitor1, or voltage monitor 2 interrupt request is acknowledged, auto-erasure or auto-programming is forcibly stopped immediately and the flash memory is reset. Interrupt handling starts after a fixed period and the flash memory restarts.

As the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be readable. Execute auto-erasure again and ensure it completes normally.

The watchdog timer does not stop during command operation, so that interrupt requests may be generated. Initialize the watchdog timer regularly.

Do not use the address match interrupt while a command is being executed because the vector of the address match interrupt is allocated in ROM.

Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

- EW1 Mode

Once a watchdog timer, oscillation stop detection, voltage monitor1, or voltage monitor 2 interrupt request is acknowledged, auto-erasure or auto-programming is forcibly stopped immediately and the flash memory is reset. Interrupt handling starts after a fixed period and the flash memory restarts.

As the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be readable. Execute auto-erasure again and ensure it completes normally.

The watchdog timer does not stop even during command operation, so that interrupt requests may be generated. Initialize the watchdog timer by using the erase-suspend function.

Do not use the address match interrupt while a command is being executed because the vector of the address match interrupt is allocated in ROM.

Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

#### **23.7.1.4 How to Access**

Write 0 before writing 1 when setting Bits FMR01, FMR02 in the FMR0 register, or FMR11 bit in the FMR1 register to 1. Do not generate an interrupt between writing 0 and 1.

#### **23.7.1.5 Rewriting User ROM Area**

In EW0 Mode, if the supply voltage drops while rewriting any block in which a rewrite control program is stored, it may not be possible to rewrite the flash memory because the rewrite control program cannot be rewritten correctly. In this case, use standard serial I/O mode.

#### **23.7.1.6 Program**

Do not write additions to the already programmed address.

#### **23.7.1.7 Suspend**

Do not use the block erase command during program-suspend.

#### **23.7.1.8 Entering Stop Mode or Wait Mode**

Do not enter stop mode or wait mode during erase-suspend.

#### **23.7.1.9 Program and Erase Voltage for Flash Memory**

To perform programming and erasure, use  $V_{CC} = 2.7\text{ V}$  to  $5.5\text{ V}$  as the supply voltage. Do not perform programming and erasure at less than  $2.7\text{ V}$ .

## 23.8 Notes on Noise

### 23.8.1 Inserting a Bypass Capacitor between VCC and VSS Pins as a Countermeasure against Noise and Latch-up

Connect a bypass capacitor (at least 0.1  $\mu\text{F}$ ) using the shortest and thickest wire possible.

### 23.8.2 Countermeasures against Noise Error of Port Control Registers

During rigorous noise testing or the like, external noise (mainly power supply system noise) can exceed the capacity of the MCU's internal noise control circuitry. In such cases the contents of the port related registers may be changed.

As a firmware countermeasure, it is recommended that the port registers, port direction registers, and pull-up control registers be reset periodically. However, examine the control processing fully before introducing the reset routine as conflicts may be created between the reset routine and interrupt routines.

## 24. Notes for On-Chip Debugger

When using the on-chip debugger to develop and debug programs for the R8C/2K Group and R8C/2L Group take note of the following.

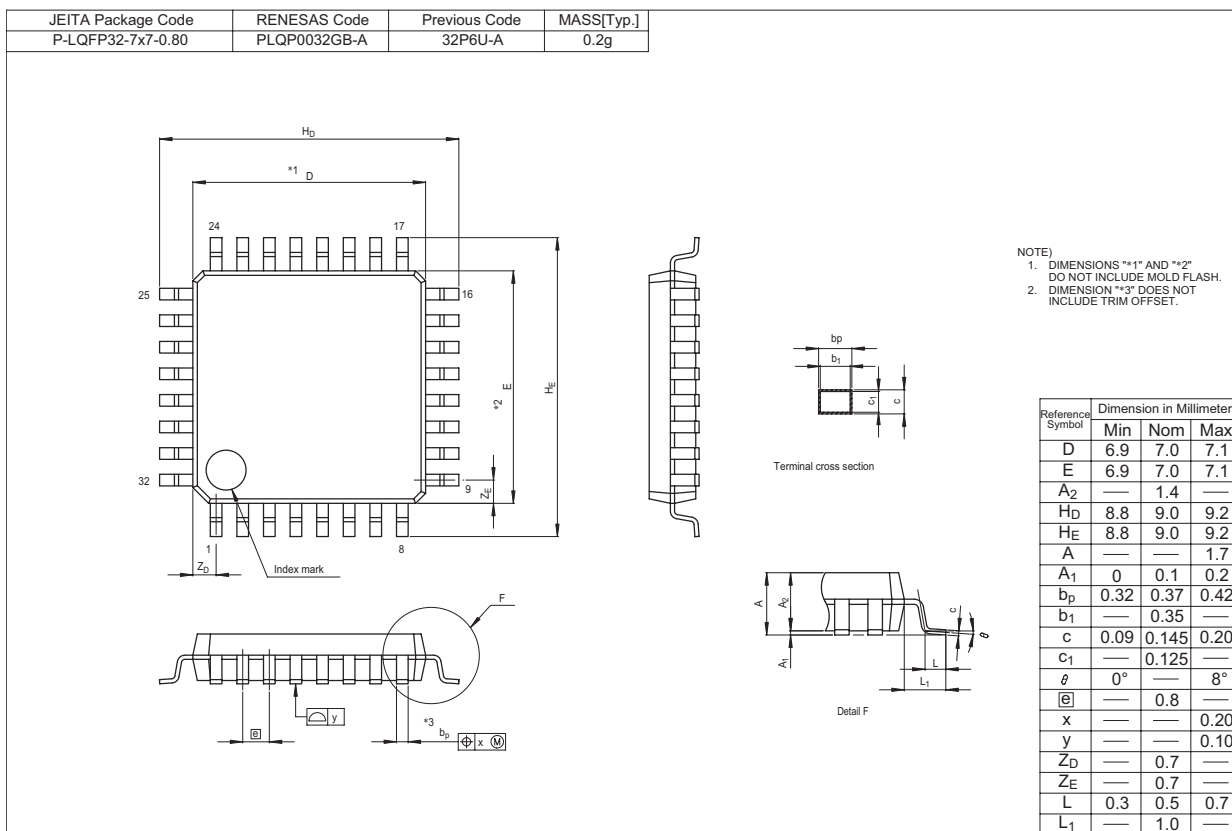
- (1) Some of the user flash memory and RAM areas are used by the on-ship debugger. These areas cannot be accessed by the user.  
Refer to the on-chip debugger manual for which areas are used.
- (2) Do not set the address match interrupt (registers AIER, RMAD0, and RMAD1 and fixed vector tables) in a user system.
- (3) Do not use the BRK instruction in a user system.
- (4) Debugging is available under the condition of supply voltage  $VCC = 2.7$  to  $5.5$  V. Debugging with the on-chip debugger under less than  $2.7$  V is not allowed.

Connecting and using the on-chip debugger has some special restrictions. Refer to the on-chip debugger manual for details.



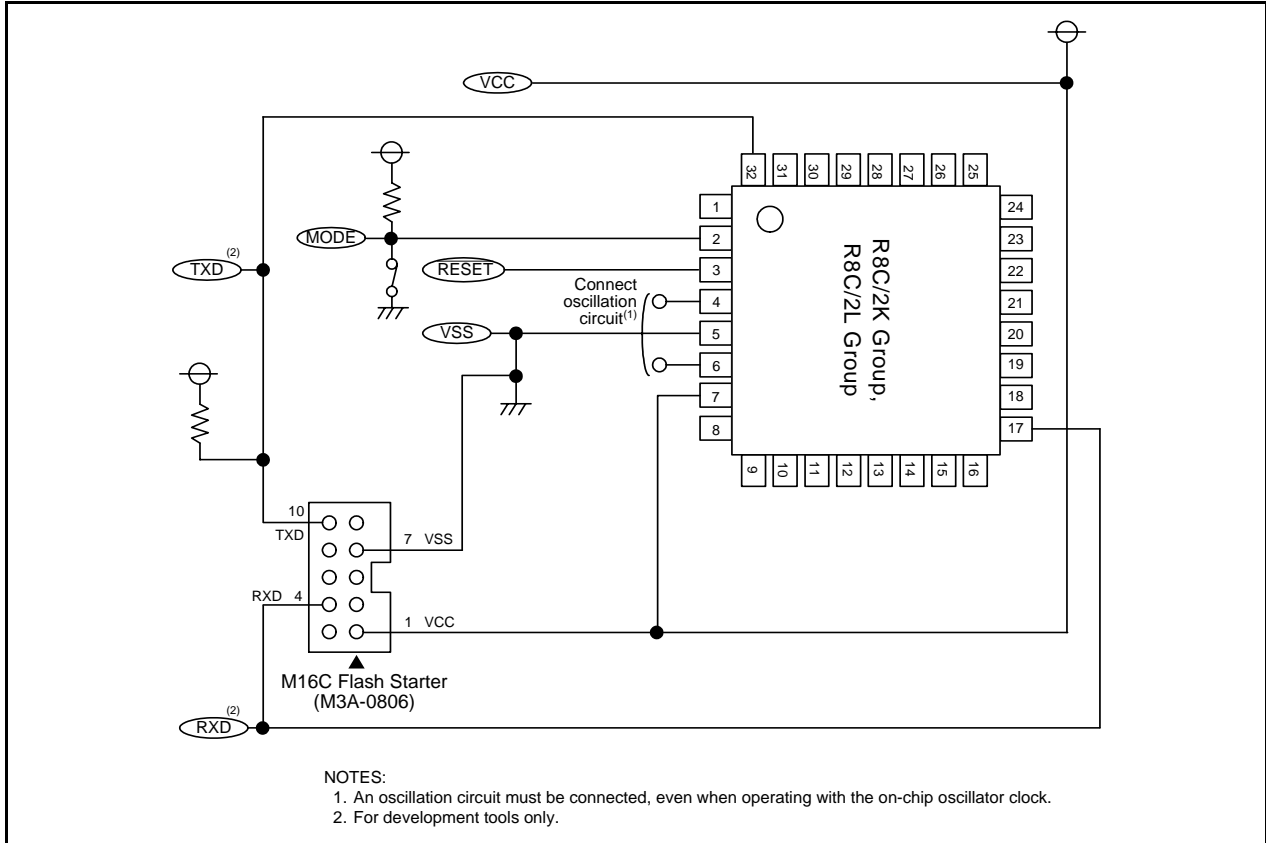
## Appendix 1. Package Dimensions

Diagrams showing the latest package dimensions and mounting information are available in the “Packages” section of the Renesas Technology website.

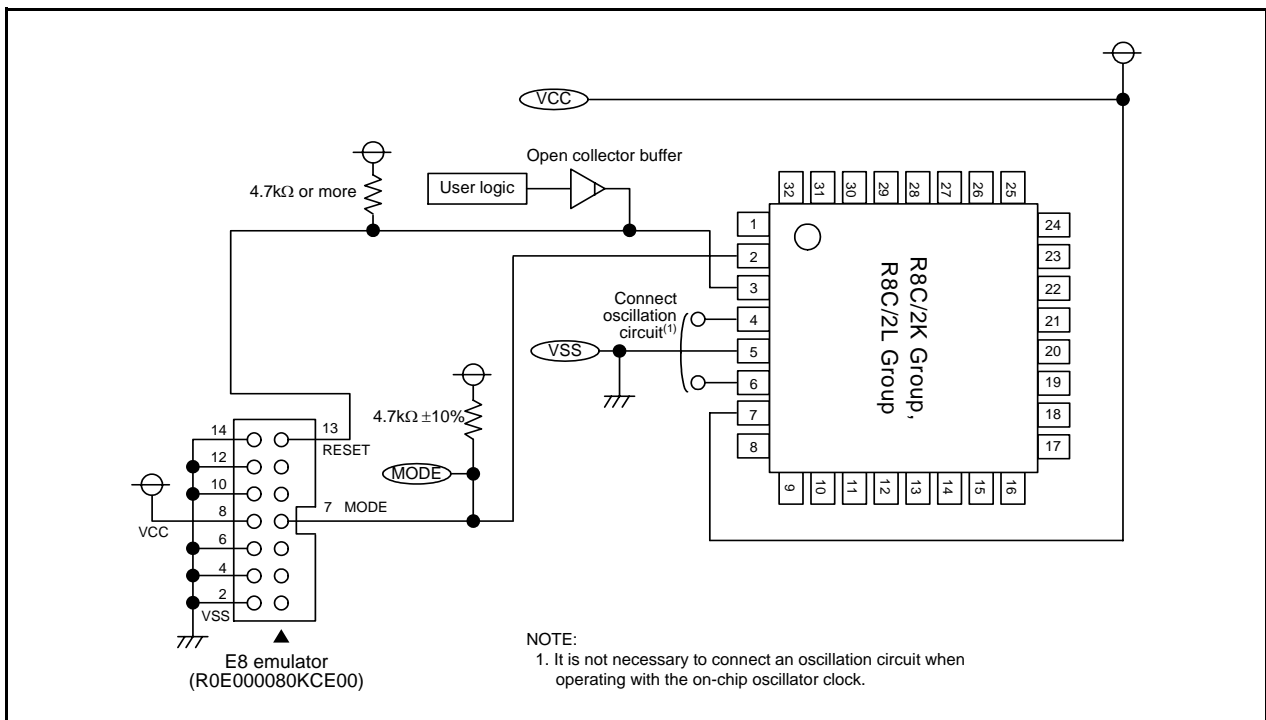


## Appendix 2. Connection Examples between Serial Writer and On-Chip Debugging Emulator

Appendix Figure 2.1 shows a Connection Example with M16C Flash Starter (M3A-0806) and Appendix Figure 2.2 shows a Connection Example with E8 Emulator (R0E000080KCE00).



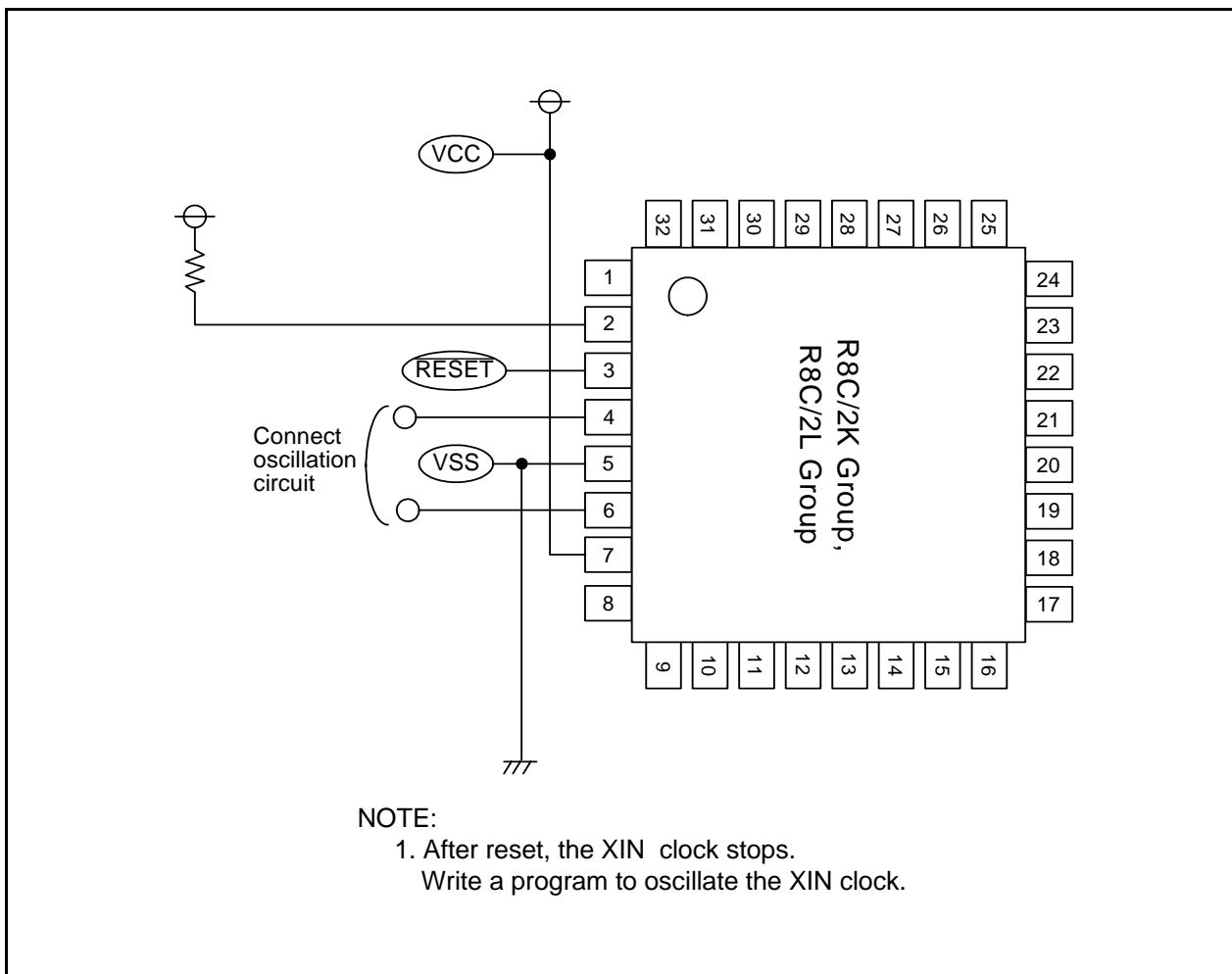
Appendix Figure 2.1 Connection Example with M16C Flash Starter (M3A-0806)



Appendix Figure 2.2 Connection Example with E8 Emulator (R0E000080KCE00)

### Appendix 3. Example of Oscillation Evaluation Circuit

Appendix Figure 3.1 shows an Example of Oscillation Evaluation Circuit.



Appendix Figure 3.1 Example of Oscillation Evaluation Circuit

## Index

[ A ]	
AD .....	351
ADCON0 .....	350
ADCON1 .....	351
ADCON2 .....	351
ADIC .....	102
AIER .....	117
[ C ]	
CM0 .....	74
CM1 .....	75
CSPR .....	131
[ F ]	
FMR0 .....	367
FMR1 .....	370
FMR4 .....	371
FRA0 .....	77
FRA1 .....	77
FRA2 .....	78
FRA6 .....	78
FRA7 .....	78
[ I ]	
INT0IC .....	104
INT1IC .....	104
INT3IC .....	104
INTEN .....	111
INTF .....	112
[ K ]	
KIEN .....	115
KUPIC .....	102
[ L ]	
LINCR .....	335
LINCR2 .....	335
LINST .....	336
[ O ]	
OCD .....	76
OFS .....	26, 126, 131, 365
[ P ]	
P2DRR .....	55
PD <sub>i</sub> (i = 0 to 4) .....	54
Pi (i = 0 to 4) .....	55
PINSR1 .....	56, 321
PINSR2 .....	56, 153
PINSR3 .....	56, 177
PM0 .....	69
PM1 .....	69
PMR .....	56, 111
PRCR .....	96
PUR0 .....	57
PUR1 .....	57
[ R ]	
RMAD0 .....	117
RMAD1 .....	117
[ S ]	
S0RIC .....	102
S0TIC .....	102
S2RIC .....	102
S2TIC .....	102
[ T ]	
TRA .....	138
TRACR .....	137
TRAIAC .....	102
TRAIOC .....	137, 139, 142, 144, 146, 149
TRAMR .....	138
TRAPRE .....	138
TRBCR .....	154
TRBIC .....	102
TRBIOC .....	155, 157, 161, 164, 168
TRBMR .....	155
TRBOCR .....	154
TRBPR .....	156
TRBPRE .....	156
TRBSC .....	156
TRC .....	181
TRCCR1 .....	178, 201, 205, 210
TRCCR2 .....	182
TRCDF .....	183
TRCGRA .....	181
TRCGRB .....	181
TRCGRC .....	181
TRCGRD .....	181
TRCIC .....	103
TRCIER .....	179
TRCIOR0 .....	185, 194, 199
TRCIOR1 .....	185, 195, 200
TRCMR .....	177
TRCOER .....	184
TRCSR .....	180
TRD0 .....	237, 253, 267, 278, 290, 304
TRDOIC .....	103
TRD1 .....	237, 253, 267, 290
TRD1IC .....	103
TRDCR0 .....	233, 248, 264, 276, 287, 302
TRDCR1 .....	233, 248, 264, 287
TRDDF0 .....	232
TRDDF1 .....	232
TRDFCR .....	231, 245, 262, 274, 285, 299
TRDGRA <sub>i</sub> (i = 0 to 1) .....	238, 253, 268, 279, 290, 305
TRDGRB <sub>i</sub> (i = 0 to 1) .....	238, 253, 268, 279, 290, 305
TRDGRC <sub>i</sub> (i = 0 to 1) .....	238, 253, 268, 279, 290, 305
TRDGRD <sub>i</sub> (i = 0 to 1) .....	238, 253, 268, 279, 290, 305
TRDIER0 .....	237, 252, 266, 278, 289, 304
TRDIER1 .....	237, 252, 266, 278, 289, 304
TRDIORA0 .....	234, 249
TRDIORA1 .....	234, 249
TRDIORC0 .....	235, 250
TRDIORC1 .....	235, 250
TRDMR .....	229, 243, 260, 273, 284, 298
TRDOCR .....	247, 264, 301
TRDOER1 .....	246, 263, 275, 286, 300
TRDOER2 .....	246, 263, 275, 286, 300
TRDPMR .....	230, 244, 261
TRDPOCR0 .....	267
TRDPOCR1 .....	267
TRDSR0 .....	236, 251, 265, 277, 288, 303
TRDSR1 .....	236, 251, 265, 277, 288, 303
TRDSTR .....	229, 243, 260, 273, 283, 297

[ U ]	
U0BRG .....	318
U0C0 .....	319
U0C1 .....	320
U0MR .....	318
U0RB .....	320
U0TB .....	319
U2BRG .....	318
U2C0 .....	319
U2C1 .....	320
U2MR .....	318
U2RB .....	320
U2TB .....	319

[ V ]	
VCA1 .....	35
VCA2 .....	35, 79
VW0C .....	36
VW1C .....	37
VW2C .....	38

[ W ]	
WDC .....	130
WDTR .....	130
WDTS .....	130

REVISION HISTORY

R8C/2K Group, R8C/2L Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.10	Jul 20, 2007	–	First Edition issued
0.20	Aug 31, 2007	108	Figure 12.11 “UART1 receive”, “UART1 transmit” deleted
		199	Figure 16.50 “• The CCLR bit in the TRCCR1 register is set to 0 .... compare match).” → “• The CCLR bit in the TRCCR1 register is set to 1 .... compare match).”
		200	Table 16.20 “j = A, B, C, or D” → “j = B, C, or D”
		254	Figure 16.96 revised
		338	Figure 18.9 “When the SBE bit .... timer RA may be used in timer mode after the SBDCT flag in the LINST register is set to 1.” → “When the SBE bit .... timer RA can be used in timer mode after the SBDCT flag in the LINST register is set to 1 and the RXDSF flag is set to 0.”
		355	Figure 19.10 “SW5 conducts when compare operation is in progress.” added
1.00	Nov 7, 2007	All pages	“Preliminary” deleted
		3, 5	Table 1.2, Table 1.4; Current consumption: “TBD” → “Typ. 10 mA” “Typ. 6 mA” “Typ. 2.0 μA” “Typ. 0.7 μA” revised
		6, 7	Table 1.5, Table 1.6 revised Figure 1.1, Figure 1.2; ROM number “XXX” added, NOTE1 added
		20	Table 4.4 “005Fh” “006Fh” “007Fh” “008Fh” added
		45, 56	Figure 7.11 added
		58 to 60	Table 7.5 NOTE3 added Table 7.6, Table 7.7, Table 7.10, Table 7.12, Table 7.14 NOTE2 added
		65	Table 7.28 NOTE2 and NOTE3 added Table 7.29, Table 7.31 NOTE2 added
		111	Figure 12.12 added
		136	Figure 16.1 “TSTART” → “TCSTF” revised
		153	Figure 16.13 added
		176	Table 16.13 “00F7h” added
		177	Figure 16.28 added
		307	Figure 16.145 “TSTP0” → “CSEL0” revised
		321	Figure 17.6 added
402	Table 22.2 NOTE2 revised		
410, 411	Table 22.14, Table 22.15 revised		
415, 419	Table 22.21, Table 22.27 revised		
447	Appendix Figure 3.1 revised		

REVISION HISTORY

R8C/2K Group, R8C/2L Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.10	Dec 21, 2007	3, 5	Table 1.2, Table 1.4: revised, NOTE2 added
		6, 7	Figure 1.1, Figure 1.2: "Y: Operating ambient ....", NOTE1 added
		15, 16	Figure 3.1, Figure 3.2: "Expanded area" deleted
		17	Table 4.1 "002Ch" added, "003Bh" "003Ch" "003Dh" deleted
		20	Table 4.4 "00D4h" "00D6h" revised
		22	Table 4.6 "0143h" revised
		35	Figure 6.5 "VCA Register" NOTE7 deleted
		59	Table 7.11 revised
		73	Figure 10.2 revised
		78	Figure 10.7 "FRA7 Register" added
		79	Figure 10.8 NOTE7 deleted
		82	10.2.2 revised
		83	10.3.8 revised
		185	Figure 16.38 TRCIOR0: b3 revised, NOTE4 added
		192	16.3.4, Table 16.16: revised
		193	Figure 16.44 revised
		194	Figure 14.45 b3 revised, NOTE3 added
		199	Figure 14.49 b3 revised
		203	Table 16.20 Interrupt request generation timing: Specification "... and TRCGRj match)" → "... and TRCGRh match)" "h = A, B, C, or D" added
		209	Table 16.22 " j = A, B, C, or D" → " j = A, B, or C"
297	Figure 16.135 b1 revised		
331	Table 17.7 revised		
402	22. "The electrical characteristics ...." added		
408	Table 21.10 Symbol "fOCO40M": Parameter added, NOTE4 added		

---

R8C/2K Group, R8C/2L Group Hardware Manual

Publication Date: Rev.0.10 Jul 20, 2007  
Rev.1.10 Dec 21, 2007

Published by: Sales Strategic Planning Div.  
Renesas Technology Corp.

---

© 2007. Renesas Technology Corp., All rights reserved. Printed in Japan



# R8C/2K Group, R8C/2L Group Hardware Manual



Renesas Electronics Corporation

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0406-0110

## X-ON Electronics

Largest Supplier of Electrical and Electronic Components

*Click to view similar products for [16-bit Microcontrollers - MCU category](#):*

*Click to view products by [Renesas manufacturer](#):*

Other Similar products are found below :

[MB90F036APMC-GSE1](#) [MB90F342CASPMC-GSE1](#) [MB90F345CESPMC-GE1](#) [MB90F349CAPFR-GSE1](#) [MB90F428GCPFR-GSE1](#)  
[MB90F462APFM-GE1](#) [MB90F462APMC-G-SNE1](#) [MB90F497GPF-GE1](#) [MB90F546GSPFR-GE1](#) [MB90F947APFR-GS-SPE1](#)  
[MB96F346RSBPMC-GS-N2E2](#) [MB96F683RBPMC-GSAE1](#) [R5F11BGEAFB#30](#) [DF3026XBL25V](#) [S912ZVFP64F1VLL](#)  
[R4F24268NVRFQV](#) [R5F107DEGSP#X0](#) [R5F11B7EANA#U0](#) [R5F21172DSP#U0](#) [M30622F8PGP#U3C](#) [MB90092PF-G-BNDE1](#)  
[MB90F335APMC1-G-SPE1](#) [MB90F342CASPF-R-GS-N2E1](#) [MB90F345CAPFR-GSE1](#) [MB90F543GPF-GE1](#) [MB90F546GSPF-GE1](#)  
[MB90F568PMCR-GE1](#) [MB90F594APFR-GE1](#) [MB90F882ASPMC-GE1](#) [MB96F346RSAPQCR-GS-N2E2](#) [MB96F387RSBPMC-GSE2](#)  
[MB96F387RSBPMC-GS-N2E2](#) [MB96F395RSAPMC-GSE2](#) [MB96F623RBPMC1-GSE1](#) [MB96F646RBPMC-GSE1](#)  
[XE167F96F66LACFXUMA1](#) [MB96F696RBPMC-GSAE1](#) [MB96F018RBPMC-GSE1](#) [MB90F962SPMCR-GE1](#) [MB90F867ASPFR-GE1](#)  
[MB90F543GPF-G-FLE1](#) [MB90F345CESPF-GE1](#) [M30290FCHP#U3A](#) [DF2239FA20IV](#) [HD64F3672FPV](#) [R5F104AEASP#V0](#)  
[R5F100BCANA#U0](#) [R5F100BFANA#U0](#) [S9S12H256J2VFVER](#) [R5F100ACASP#V0](#)