

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.

# M32C/88 Group (M32C/88T)

Hardware Manual

RENESAS 16/32-BIT SINGLE-CHIP  
MICROCOMPUTER  
M16C FAMILY / M32C/80 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>).

### Keep safety first in your circuit designs!

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

### Notes regarding these materials

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

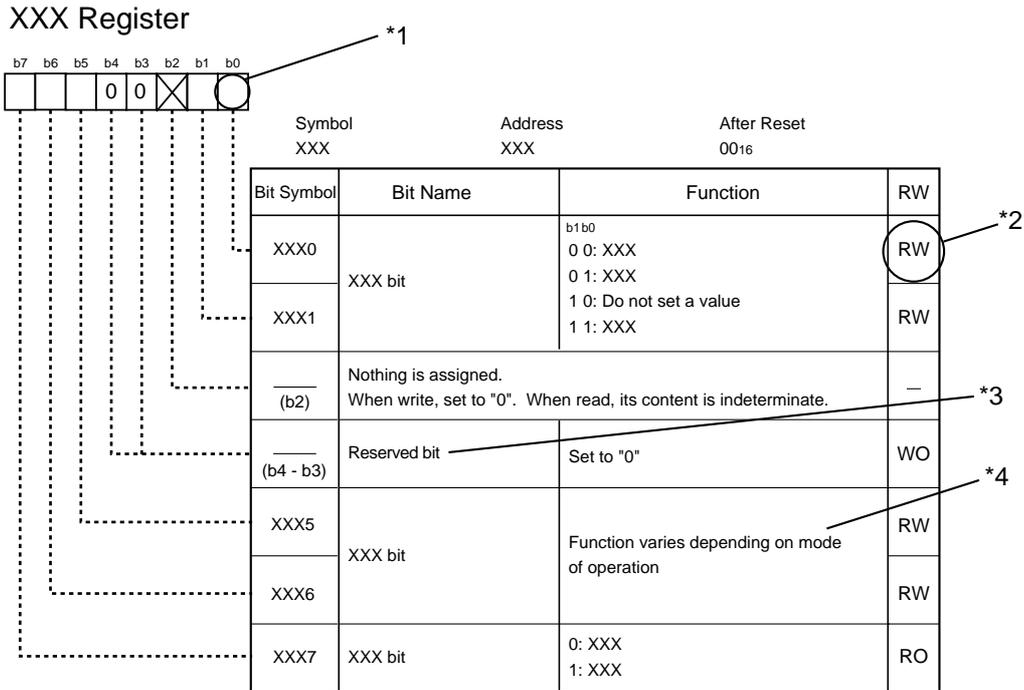
# How to Use This Manual

## 1. Introduction

This hardware manual provides detailed information on the M32C/88 Group (M32C/88T) microcomputer. Users are expected to have basic knowledge of electric circuits, logical circuits and microcomputers.

## 2. Register Diagram

The symbols, and descriptions, used for bit function in each register are shown 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 concerned. As the bit may be use for future functions, set to "0" when writing to this bit.  
• Do not set a value  
The operation is not guaranteed when a value is set.  
• Function varies depending on mode of operation  
Bit function varies depending on peripheral function mode.  
Refer to respective register for each mode.

### 3. M16C Family Documents

The following documents were prepared for the M16C family. <sup>(1)</sup>

Document	Contents
Short Sheet	Hardware overview
Data Sheet	Hardware overview and electrical characteristics
Hardware Manual	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, timing charts)
Software Manual	Detailed description of assembly instructions and microcomputer performance of each instruction
Application Note	<ul style="list-style-type: none"><li>• Application examples of peripheral functions</li><li>• Sample programs</li><li>• Introduction to the basic functions in the M16C family</li><li>• Programming method with Assembly and C languages</li></ul>
RENESAS TECHNICAL UPDATE	Preliminary report about the specification of a product, a document, etc.

#### NOTES :

1. Before using this material, please visit the our website to verify that this is the most current document available.

# Table of Contents

<b>Quick Reference by Address</b> .....	<b>1</b>
<b>1. Overview</b> .....	<b>1</b>
1.1 Applications .....	1
1.2 Performance Overview .....	2
1.3 Block Diagram .....	4
1.4 Product Information .....	5
1.5 Pin Assignment .....	7
1.6 Pin Description .....	14
<b>2. Central Processing Unit (CPU)</b> .....	<b>17</b>
2.1 General Registers .....	18
2.1.1 Data Registers (R0, R1, R2 and R3).....	18
2.1.2 Address Registers (A0 and A1) .....	18
2.1.3 Static Base Register (SB).....	18
2.1.4 Frame Base Register (FB) .....	18
2.1.5 Program Counter (PC) .....	18
2.1.6 Interrupt Table Register (INTB) .....	18
2.1.7 User Stack Pointer (USP), Interrupt Stack Pointer (ISP) .....	18
2.1.8 Flag Register (FLG).....	18
2.2 High-Speed Interrupt Registers .....	19
2.3 DMAC-Associated Registers .....	19
<b>3. Memory</b> .....	<b>20</b>
<b>4. Special Function Registers (SFRs)</b> .....	<b>21</b>
<b>5. Reset</b> .....	<b>45</b>
5.1 Hardware Reset 1 .....	45
5.1.1 Reset on a Stable Supply Voltage .....	45
5.1.2 Power-on Reset .....	45
5.2 Software Reset .....	47
5.3 Watchdog Timer Reset .....	47
5.4 Internal Space .....	47
<b>6. Cold Start-up/Warm Start-up Determine Function</b> ____	<b>48</b>
<b>7. Processor Mode</b> .....	<b>50</b>
7.1 Types of Processor Mode .....	50
7.2 Setting of Processor Mode .....	50

<b>8. Clock Generation Circuit</b>	<b>54</b>
8.1 Types of the Clock Generation Circuit	54
8.1.1 Main Clock	63
8.1.2 Sub Clock	64
8.1.3 On-Chip Oscillator Clock	65
8.1.4 PLL Clock	67
8.2 CPU Clock and BCLK	68
8.3 Peripheral Function Clock	68
8.3.1 f1, f8, f32 and f2n	68
8.3.2 fAD	68
8.3.3 fc32	69
8.3.4 fCAN	69
8.4 Clock Output Function	69
8.5 Power Consumption Control	70
8.5.1 Normal Operating Mode	70
8.5.2 Wait Mode	71
8.5.3 Stop Mode	73
8.6 System Clock Protect Function	77
<b>9. Protection</b>	<b>78</b>
<b>10. Interrupts</b>	<b>79</b>
10.1 Types of Interrupts	79
10.2 Software Interrupts	80
10.2.1 Undefined Instruction Interrupt	80
10.2.2 Overflow Interrupt	80
10.2.3 BRK Interrupt	80
10.2.4 BRK2 Interrupt	80
10.2.5 INT Instruction Interrupt	80
10.3 Hardware Interrupts	81
10.3.1 Special Interrupts	81
10.3.2 Peripheral Function Interrupt	81
10.4 High-Speed Interrupt	82
10.5 Interrupts and Interrupt Vectors	82
10.5.1 Fixed Vector Tables	83
10.5.2 Relocatable Vector Tables	83
10.6 Interrupt Request Acknowledgement	86
10.6.1 I Flag and IPL	86
10.6.2 Interrupt Control Register and RLVL Register	86
10.6.3 Interrupt Sequence	90

10.6.4	Interrupt Response Time .....	91
10.6.5	IPL Change when Interrupt Request is Acknowledged .....	92
10.6.6	Saving a Register .....	93
10.6.7	Restoration from Interrupt Routine .....	93
10.6.8	Interrupt Priority .....	94
10.6.9	Interrupt Priority Level Select Circuit .....	94
10.7	$\overline{\text{INT}}$ Interrupt .....	96
10.8	$\overline{\text{NMI}}$ Interrupt .....	97
10.9	Key Input Interrupt .....	97
10.10	Address Match Interrupt .....	98
10.11	Intelligent I/O Interrupt and CAN Interrupt .....	99
<b>11.</b>	<b>Watchdog Timer</b> .....	<b>103</b>
11.1	Count Source Protection Mode .....	106
<b>12.</b>	<b>DMAC</b> .....	<b>107</b>
12.1	Transfer Cycle .....	114
12.1.1	Effect of Source and Destination Addresses .....	114
12.1.2	Effect of Software Wait State .....	114
12.2	DMAC Transfer Cycle .....	116
12.3	Channel Priority and DMA Transfer Timing .....	116
<b>13.</b>	<b>DMAC II</b> .....	<b>118</b>
13.1	DMAC II Settings .....	118
13.1.1	RLVL Register .....	118
13.1.2	DMAC II Index .....	120
13.1.3	Interrupt Control Register for the Peripheral Function .....	122
13.1.4	Relocatable Vector Table for the Peripheral Function .....	122
13.1.5	IRLT Bit in the IIOiE Register (i=0 to 6, 8 to 11) .....	122
13.2	DMAC II Performance .....	122
13.3	Transfer Data .....	122
13.3.1	Memory-to-memory Transfer .....	122
13.3.2	Immediate Data Transfer .....	123
13.3.3	Calculation Transfer .....	123
13.4	Transfer Modes .....	123
13.4.1	Single Transfer .....	123
13.4.2	Burst Transfer .....	123
13.5	Multiple Transfer .....	123
13.6	Chained Transfer .....	124
13.7	End-of-Transfer Interrupt .....	124
13.8	Execution Time .....	125

<b>14. Timer</b>	<b>126</b>
14.1 Timer A	128
14.1.1 Timer Mode	134
14.1.2 Event Counter Mode	136
14.1.3 One-Shot Timer Mode	140
14.1.4 Pulse Width Modulation Mode	142
14.2 Timer B	145
14.2.1 Timer Mode	148
14.2.2 Event Counter Mode	149
14.2.3 Pulse Period/Pulse Width Measurement Mode	151
<b>15. Three-Phase Motor Control Timer Functions</b>	<b>154</b>
<b>16. Serial I/O</b>	<b>165</b>
16.1 Clock Synchronous Serial I/O Mode	175
16.1.1 Selecting CLK Polarity Selecting	179
16.1.2 Selecting LSB First or MSB First	179
16.1.3 Continuous Receive Mode	180
16.1.4 Serial Data Logic Inverse	180
16.2 Clock Asynchronous Serial I/O (UART) Mode	181
16.2.1 Bit Rate	185
16.2.2 Selecting LSB First or MSB First	186
16.2.3 Serial Data Logic Inverse	186
16.2.4 TxD and RxD I/O Polarity Inverse	187
16.3 Special Mode 1 (I <sup>2</sup> C Mode)	188
16.3.1 Detecting Start Condition and Stop Condition	194
16.3.2 Start Condition or Stop Condition Output	194
16.3.3 Arbitration	196
16.3.4 Transfer Clock	196
16.3.5 SDA Output	196
16.3.6 SDA Input	197
16.3.7 ACK, NACK	197
16.3.8 Transmit and Receive Reset	197
16.4 Special Mode 2	198
16.4.1 $\overline{SS}_i$ Input Pin Function (i=0 to 4)	201
16.4.2 Clock Phase Setting Function	202
16.5 Special Mode 3 (GCI Mode)	204
16.6 Special Mode 4 (IE Mode)	208

16.7 Special Mode 5 (SIM Mode) .....	212
16.7.1 Parity Error Signal .....	216
16.7.2 Format .....	217
<b>17. A/D Converter _____</b>	<b>218</b>
17.1 Mode Description .....	226
17.1.1 One-shot Mode .....	226
17.1.2 Repeat Mode .....	227
17.1.3 Single Sweep Mode .....	228
17.1.4 Repeat Sweep Mode 0 .....	229
17.1.5 Repeat Sweep Mode 1 .....	230
17.1.6 Multi-Port Single Sweep Mode .....	231
17.1.7 Multi-Port Repeat Sweep Mode 0 .....	232
17.2 Functions .....	233
17.2.1 Resolution Select Function .....	233
17.2.2 Sample and Hold Function .....	233
17.2.3 Trigger Select Function .....	233
17.2.4 DMAC Operating Mode .....	233
17.2.5 Extended Analog Input Pins .....	234
17.2.6 External Operating Amplifier (Op-Amp) Connection Mode .....	234
17.2.7 Power Consumption Reducing Function .....	235
17.2.8 Output Impedance of Sensor Equivalent Circuit under A/D Conversion ...	235
<b>18. D/A Converter _____</b>	<b>237</b>
<b>19. CRC Calculation _____</b>	<b>240</b>
<b>20. X/Y Conversion _____</b>	<b>242</b>
<b>21. Intelligent I/O _____</b>	<b>245</b>
21.1 Base Timer .....	254
21.2 Time Measurement Function .....	259
21.3 Waveform Generating Function .....	264
21.3.1 Single-Phase Waveform Output Mode .....	265
21.3.2 Phase-Delayed Waveform Output Mode .....	267
21.3.3 Set/Reset Waveform Output (SR Waveform Output) Mode .....	269
21.4 Communication Unit 0 and 1 Communication Function .....	272
21.4.1 Clock Synchronous Serial I/O Mode (Communication Units 0 and 1) .....	282
21.4.2 Clock Asynchronous Serial I/O (UART) Mode (Communication Unit 1) ..	286
21.4.3 HDLC Data Processing Mode (Communication Units 0 and 1) .....	289

## 22. CAN Module 292

22.1 CAN-Associated Registers .....	296
22.1.1 CANi Control Register 0 (CiCTRL0 Register) (i=0 to 2) .....	296
22.1.2 CANi Control Register 1 (CiCTRL1 Register) (i=0 to 2) .....	299
22.1.3 CANi Sleep Control Register (CiSLPR Register) (i=0 to 2) .....	300
22.1.4 CANi Status Register (CiSTR Register) (i=0 to 2) .....	301
22.1.5 CANi Extended ID Register (CiIDR Register) (i=0 to 2) .....	304
22.1.6 CANi Configuration Register (CiCONR Register) (i=0 to 2) .....	305
22.1.7 CANi Baud Rate Prescaler (CiBRP Register) (i=0 to 2) .....	307
22.1.8 CANi Time Stamp Register (CiTSR Register) (i=0 to 2) .....	308
22.1.10 CANi Receive Error Count Register (CiREC Register) (i=0 to 2) .....	309
22.1.9 CANi Transmit Error Count Register (CiTEC Register) (i=0 to 2) .....	309
22.1.11 CANi Slot Interrupt Status Register (CiSISTR Register) (i=0 to 2).....	310
22.1.12 CANi Slot Interrupt Mask Register (CiSIMKR Register) (i=0 to 2) .....	312
22.1.13 CANi Error Interrupt Mask Register (CiEIMKR Register) (i=0 to 2) .....	313
22.1.14 CANi Error Interrupt Status Register (CiEISTR Register) (i=0 to 2) .....	314
22.1.15 CANi Error Factor Register (CiEFR Register) (i=0 to 2) .....	315
22.1.16 CANi Mode Register (CiMDR Register) (i=0 to 2) .....	316
22.1.17 CANi Single-Shot Control Register (CiSSCTRL Register) (i=0 to 2) .....	318
22.1.18 CANi Single-Shot Status Register (CiSSSTR Register) (i=0 to 2) .....	319
22.1.19 CANi Global Mask Register, CANi Local Mask Register A and CANi Local Mask Register B (CiGMRk, CiLMARk and CiLMBRk Registers) (i=0 to 2, k=0 to 4) ...	320
22.1.20 CANi Message Slot j Control Register (CiMCTLj Register) (i=0 to 2, j=0 to 15).	327
22.1.21 CANi Slot Buffer Select Register (CiSBS Register) (i=0 to 2) .....	331
22.1.22 CANi Message Slot Buffer j (i=0 to 2, j=0,1).....	332
22.1.23 CANi Acceptance Filter Support Register (CiAFS Register) (i=0 to 2) ...	336
22.2 CAN Clock.....	337
22.2.1 Main Clock Direct Mode .....	337
22.3 Timing with CAN-Associated Registers .....	338
22.3.1 CAN Module Reset Timing .....	338
22.3.2 CAN Transmit Timing .....	338
22.3.3 CAN Receive Timing .....	339
22.3.4 CAN Bus Error Timing .....	340
22.4 CAN Interrupts .....	340
22.4.1 CANi Wake-Up Interrupt .....	340
22.4.2 CANij Interrupts .....	341
22.5 CAN0/CAN2 Combination Mode .....	344
22.5.1 Notes for CAN0/CAN2 Combination Mode .....	345

<b>23. Programmable I/O Ports</b>	<b>346</b>
23.1 Port Pi Direction Register (PDi Register, i=0 to 15)	346
23.2 Port Pi Register (Pi Register, i=0 to 15)	346
23.3 Function Select Register Aj (PSj Register) (j=0 to 3, 5, 8, 9)	346
23.4 Function Select Register B0 to B3 (PSL0 to PSL3 Registers)	346
23.5 Function Select Register C, C2, C3 (PSC, PSC2, PSC3 Registers)	346
23.6 Function Select Register D (PSD1 Register)	347
23.7 Pull-up Control Register 0 to 4 (PUR0 to PUR4 Registers)	347
23.8 Port Control Register (PCR Register)	347
23.9 Input Function Select Register (IPS and IPSA Registers)	347
23.10 Analog Input and Other Peripheral Function Input	347
<b>24. Flash Memory Version</b>	<b>369</b>
24.1 Memory Map	370
24.1.1 Boot Mode	371
24.2 Functions to Prevent Rewriting of Flash Memory	371
24.2.1 ROM Code Protect Function	371
24.2.2 ID Code Verify Function	371
24.3 CPU Rewrite Mode	373
24.3.1 EW Mode 0	373
24.3.2 EW Mode 1	373
24.3.3 Flash Memory Control Register (FMR0 Register and FMR1 Register)	374
24.3.4 Precautions in CPU Rewrite Mode	380
24.3.5 Software Commands	382
24.3.6 Data Protect Function	388
24.3.7 Status Register (SRD Register)	388
24.3.8 Full Status Check	390
24.4 Standard Serial I/O Mode	392
24.4.1 ID Code Verify Function	392
24.4.2 Circuit Application in Standard Serial I/O Mode	396
24.5 Parallel I/O Mode	398
24.5.1 Boot ROM Area	398
24.5.2 ROM Code Protect Function	398
<b>25. Electrical Characteristics</b>	<b>399</b>
<b>26. Precautions</b>	<b>411</b>
26.1 Special Function Registers (SFRs)	411
26.1.1 100-Pin Package	411
26.1.2 Register Settings	411

26.2	Clock Generation Circuit .....	412
26.2.1	CPU Clock.....	412
26.2.2	Sub Clock .....	412
26.2.3	PLL Frequency Synthesizer .....	413
26.2.4	External Clock .....	413
26.2.5	Clock Divide Ratio .....	413
26.2.6	Power Consumption Control .....	413
26.3	Protection .....	416
26.4	Interrupts .....	417
26.4.1	ISP Setting .....	417
26.4.2	$\overline{\text{NMI}}$ Interrupt.....	417
26.4.3	$\overline{\text{INT}}$ Interrupt .....	417
26.4.4	Watchdog Timer Interrupt .....	418
26.4.5	Changing Interrupt Control Register .....	418
26.4.6	Changing IIOiLR Register (i = 0 to 6, 8 to 11) .....	418
26.4.7	Changing RLVL Register .....	418
26.5	DMAC .....	419
26.6	Timer.....	420
26.6.1	Timers A and B .....	420
26.6.2	Timer A.....	420
26.6.3	Timer B.....	422
26.7	Serial I/O.....	423
26.7.1	Clock Synchronous Serial I/O Mode .....	423
26.7.2	UART Mode.....	424
26.7.3	Special Mode 1 (I <sup>2</sup> C Mode) .....	424
26.8	A/D Converter .....	425
26.9	Intelligent I/O .....	427
26.9.1	Register Setting .....	427
26.10	Programmable I/O Ports .....	428
26.11	Flash Memory Version .....	429
26.11.1	Boot Mode.....	429
26.12	Noise .....	430
<b>Package Dimensions</b> .....		<b>431</b>
<b>Register Index</b> .....		<b>432</b>

## Quick Reference by Address

Address	Register	Page	Address	Register	Page
0000 <sub>16</sub>			0030 <sub>16</sub>		
0001 <sub>16</sub>			0031 <sub>16</sub>		
0002 <sub>16</sub>			0032 <sub>16</sub>		
0003 <sub>16</sub>			0033 <sub>16</sub>		
0004 <sub>16</sub>	Processor Mode Register 0 (PM0)	51	0034 <sub>16</sub>		
0005 <sub>16</sub>	Processor Mode Register 1 (PM1)	52	0035 <sub>16</sub>		
0006 <sub>16</sub>	System Clock Control Register 0 (CM0)	56	0036 <sub>16</sub>		
0007 <sub>16</sub>	System Clock Control Register 1 (CM1)	57	0037 <sub>16</sub>		
0008 <sub>16</sub>			0038 <sub>16</sub>		
0009 <sub>16</sub>	Address Match Interrupt Enable Register (AIER)	98	0039 <sub>16</sub>	Address Match Interrupt Register 6 (RMAD6)	98
000A <sub>16</sub>	Protect Register (PRCR)	78	003A <sub>16</sub>		
000B <sub>16</sub>			003B <sub>16</sub>		
000C <sub>16</sub>	Main Clock Division Register (MCD)	58	003C <sub>16</sub>		
000D <sub>16</sub>	Oscillation Stop Detection Register (CM2)	59	003D <sub>16</sub>	Address Match Interrupt Register 7 (RMAD7)	98
000E <sub>16</sub>	Watchdog Timer Start Register (WDTS)		003E <sub>16</sub>		
000F <sub>16</sub>	Watchdog Timer Control Register (WDC)	104	003F <sub>16</sub>		
0010 <sub>16</sub>			0040 <sub>16</sub>		
0011 <sub>16</sub>	Address Match Interrupt Register 0 (RMAD0)	98	0041 <sub>16</sub>		
0012 <sub>16</sub>			0042 <sub>16</sub>		
0013 <sub>16</sub>	Processor Mode Register 2 (PM2)	62	0043 <sub>16</sub>		
0014 <sub>16</sub>			0044 <sub>16</sub>		
0015 <sub>16</sub>	Address Match Interrupt Register 1 (RMAD1)	98	0045 <sub>16</sub>		
0016 <sub>16</sub>			0046 <sub>16</sub>		
0017 <sub>16</sub>			0047 <sub>16</sub>		
0018 <sub>16</sub>			0048 <sub>16</sub>		
0019 <sub>16</sub>	Address Match Interrupt Register 2 (RMAD2)	98	0049 <sub>16</sub>		
001A <sub>16</sub>			004A <sub>16</sub>		
001B <sub>16</sub>			004B <sub>16</sub>		
001C <sub>16</sub>			004C <sub>16</sub>		
001D <sub>16</sub>	Address Match Interrupt Register 3 (RMAD3)	98	004D <sub>16</sub>		
001E <sub>16</sub>			004E <sub>16</sub>		
001F <sub>16</sub>			004F <sub>16</sub>		
0020 <sub>16</sub>			0050 <sub>16</sub>		
0021 <sub>16</sub>			0051 <sub>16</sub>		
0022 <sub>16</sub>			0052 <sub>16</sub>		
0023 <sub>16</sub>			0053 <sub>16</sub>		
0024 <sub>16</sub>			0054 <sub>16</sub>		
0025 <sub>16</sub>			0055 <sub>16</sub>	Flash Memory Control Register 1 (FMR1)	375
0026 <sub>16</sub>	PLL Control Register 0 (PLC0)	61	0056 <sub>16</sub>		
0027 <sub>16</sub>	PLL Control Register 1 (PLC1)		0057 <sub>16</sub>	Flash Memory Control Register 0 (FMR0)	374
0028 <sub>16</sub>			0058 <sub>16</sub>		
0029 <sub>16</sub>	Address Match Interrupt Register 4 (RMAD4)	98	0059 <sub>16</sub>		
002A <sub>16</sub>			005A <sub>16</sub>		
002B <sub>16</sub>			005B <sub>16</sub>		
002C <sub>16</sub>			005C <sub>16</sub>		
002D <sub>16</sub>	Address Match Interrupt Register 5 (RMAD5)	98	005D <sub>16</sub>		
002E <sub>16</sub>			005E <sub>16</sub>		
002F <sub>16</sub>			005F <sub>16</sub>		

Blank spaces are reserved. No access is allowed.

## Quick Reference by Address

Address	Register	Page
0060 <sub>16</sub>		
0061 <sub>16</sub>		
0062 <sub>16</sub>		
0063 <sub>16</sub>		
0064 <sub>16</sub>		
0065 <sub>16</sub>		
0066 <sub>16</sub>		
0067 <sub>16</sub>		
0068 <sub>16</sub>	DMA0 Interrupt Control Register (DM0IC)	
0069 <sub>16</sub>	Timer B5 Interrupt Control Register (TB5IC)	
006A <sub>16</sub>	DMA2 Interrupt Control Register (DM2IC)	
006B <sub>16</sub>	UART2 Receive /ACK Interrupt Control Register (S2RIC)	
006C <sub>16</sub>	Timer A0 Interrupt Control Register (TA0IC)	
006D <sub>16</sub>	UART3 Receive /ACK Interrupt Control Register (S3RIC)	
006E <sub>16</sub>	Timer A2 Interrupt Control Register (TA2IC)	
006F <sub>16</sub>	UART4 Receive /ACK Interrupt Control Register (S4RIC)	
0070 <sub>16</sub>	Timer A4 Interrupt Control Register (TA4IC)	
0071 <sub>16</sub>	UART0 Bus Conflict Detect Interrupt Control Register (BCN0IC)/ UART3 Bus Conflict Detect Interrupt Control Register (BCN3IC)	87
0072 <sub>16</sub>	UART0 Receive/ACK Interrupt Control Register (S0RIC)	
0073 <sub>16</sub>	A/D0 Conversion Interrupt Control Register (AD0IC)	
0074 <sub>16</sub>	UART1 Receive/ACK Interrupt Control Register (S1RIC)	
0075 <sub>16</sub>	Intelligent I/O Interrupt Control Register 0 (IIO0IC)/ CAN Interrupt 3 Control Register (CAN3IC)	
0076 <sub>16</sub>	Timer B1 Interrupt Control Register (TB1IC)	
0077 <sub>16</sub>	Intelligent I/O Interrupt Control Register 2 (IIO2IC)	
0078 <sub>16</sub>	Timer B3 Interrupt Control Register (TB3IC)	
0079 <sub>16</sub>	Intelligent I/O Interrupt Control Register 4 (IIO4IC)	
007A <sub>16</sub>	INT5 Interrupt Control Register (INT5IC)	88
007B <sub>16</sub>	CAN Interrupt 8 Control Register (CAN8IC)	87
007C <sub>16</sub>	INT3 Interrupt Control Register (INT3IC)	88
007D <sub>16</sub>	Intelligent I/O Interrupt Control Register 8 (IIO8IC)	87
007E <sub>16</sub>	INT1 Interrupt Control Register (INT1IC)	88
007F <sub>16</sub>	Intelligent I/O Interrupt Control Register 10 (IIO10IC)/ CAN Interrupt 1 Control Register (CAN1IC)	87
0080 <sub>16</sub>		
0081 <sub>16</sub>	CAN Interrupt 2 Control Register (CAN2IC)	87
0082 <sub>16</sub>		
0083 <sub>16</sub>		
0084 <sub>16</sub>		
0085 <sub>16</sub>		
0086 <sub>16</sub>		
0087 <sub>16</sub>		
0088 <sub>16</sub>	DMA1 Interrupt Control Register (DM1IC)	
0089 <sub>16</sub>	UART2 Transmit /NACK Interrupt Control Register (S2TIC)	
008A <sub>16</sub>	DMA3 Interrupt Control Register (DM3IC)	
008B <sub>16</sub>	UART3 Transmit /NACK Interrupt Control Register (S3TIC)	
008C <sub>16</sub>	Timer A1 Interrupt Control Register (TA1IC)	87
008D <sub>16</sub>	UART4 Transmit /NACK Interrupt Control Register (S4TIC)	
008E <sub>16</sub>	Timer A3 Interrupt Control Register (TA3IC)	
008F <sub>16</sub>	UART2 Bus Conflict Detect Interrupt Control Register (BCN2IC)	

Address	Register	Page
0090 <sub>16</sub>	UART0 Transmit /NACK Interrupt Control Register (S0TIC)	
0091 <sub>16</sub>	UART1 Bus Conflict Detect Interrupt Control Register (BCN1IC)/ UART4 Bus Conflict Detect Interrupt Control Register (BCN4IC)	
0092 <sub>16</sub>	UART1 Transmit/NACK Interrupt Control Register (S1TIC)	
0093 <sub>16</sub>	Key Input Interrupt Control Register (KUPIC)	
0094 <sub>16</sub>	Timer B0 Interrupt Control Register (TB0IC)	
0095 <sub>16</sub>	Intelligent I/O Interrupt Control Register 1 (IIO1IC)/ CAN Interrupt 4 Control Register (CAN4IC)	87
0096 <sub>16</sub>	Timer B2 Interrupt Control Register (TB2IC)	
0097 <sub>16</sub>	Intelligent I/O Interrupt Control Register 3 (IIO3IC)/ CAN Interrupt 7 Control Register (CAN7IC)	
0098 <sub>16</sub>	Timer B4 Interrupt Control Register (TB4IC)	
0099 <sub>16</sub>	CAN Interrupt 5 Control Register (CAN5IC)	
009A <sub>16</sub>	INT4 Interrupt Control Register (INT4IC)	88
009B <sub>16</sub>		
009C <sub>16</sub>	INT2 Interrupt Control Register (INT2IC)	88
009D <sub>16</sub>	Intelligent I/O Interrupt Control Register 9 (IIO9IC)/ CAN Interrupt 0 Control Register (CAN0IC)	87
009E <sub>16</sub>	INT0 Interrupt Control Register (INT0IC)	88
009F <sub>16</sub>	Exit Priority Control Register (RLVL)	89
00A0 <sub>16</sub>	Interrupt Request Register 0 (IIO0IR)	
00A1 <sub>16</sub>	Interrupt Request Register 1 (IIO1IR)	
00A2 <sub>16</sub>	Interrupt Request Register 2 (IIO2IR)	101
00A3 <sub>16</sub>	Interrupt Request Register 3 (IIO3IR)	
00A4 <sub>16</sub>	Interrupt Request Register 4 (IIO4IR)	
00A5 <sub>16</sub>	Interrupt Request Register 5 (IIO5IR)	
00A6 <sub>16</sub>	Interrupt Request Register 6 (IIO6IR)	
00A7 <sub>16</sub>		
00A8 <sub>16</sub>	Interrupt Request Register 8 (IIO8IR)	
00A9 <sub>16</sub>	Interrupt Request Register 9 (IIO9IR)	101
00AA <sub>16</sub>	Interrupt Request Register 10 (IIO10IR)	
00AB <sub>16</sub>	Interrupt Request Register 11 (IIO11IR)	
00AC <sub>16</sub>		
00AD <sub>16</sub>		
00AE <sub>16</sub>		
00AF <sub>16</sub>		
00B0 <sub>16</sub>	Interrupt Enable Register 0 (IIO0IE)	
00B1 <sub>16</sub>	Interrupt Enable Register 1 (IIO1IE)	
00B2 <sub>16</sub>	Interrupt Enable Register 2 (IIO2IE)	102
00B3 <sub>16</sub>	Interrupt Enable Register 3 (IIO3IE)	
00B4 <sub>16</sub>	Interrupt Enable Register 4 (IIO4IE)	
00B5 <sub>16</sub>	Interrupt Enable Register 5 (IIO5IE)	
00B6 <sub>16</sub>	Interrupt Enable Register 6 (IIO6IE)	
00B7 <sub>16</sub>		
00B8 <sub>16</sub>	Interrupt Enable Register 8 (IIO8IE)	
00B9 <sub>16</sub>	Interrupt Enable Register 9 (IIO9IE)	102
00BA <sub>16</sub>	Interrupt Enable Register 10 (IIO10IE)	
00BB <sub>16</sub>	Interrupt Enable Register 11 (IIO11IE)	
00BC <sub>16</sub>		
00BD <sub>16</sub>		
00BE <sub>16</sub>		
00BF <sub>16</sub>		

Blank spaces are reserved. No access is allowed.

## Quick Reference by Address

Address	Register	Page
00C0 <sub>16</sub>		
00C1 <sub>16</sub>		
00C2 <sub>16</sub>		
00C3 <sub>16</sub>		
00C4 <sub>16</sub>		
00C5 <sub>16</sub>		
00C6 <sub>16</sub>		
00C7 <sub>16</sub>		
00C8 <sub>16</sub>		
00C9 <sub>16</sub>		
00CA <sub>16</sub>		
00CB <sub>16</sub>		
00CC <sub>16</sub>		
00CD <sub>16</sub>		
00CE <sub>16</sub>		
00CF <sub>16</sub>		
00D0 <sub>16</sub>		
00D1 <sub>16</sub>		
00D2 <sub>16</sub>		
00D3 <sub>16</sub>		
00D4 <sub>16</sub>		
00D5 <sub>16</sub>		
00D6 <sub>16</sub>		
00D7 <sub>16</sub>		
00D8 <sub>16</sub>		
00D9 <sub>16</sub>		
00DA <sub>16</sub>		
00DB <sub>16</sub>		
00DC <sub>16</sub>		
00DD <sub>16</sub>		
00DE <sub>16</sub>		
00DF <sub>16</sub>		
00E0 <sub>16</sub>		
00E1 <sub>16</sub>		
00E2 <sub>16</sub>		
00E3 <sub>16</sub>		
00E4 <sub>16</sub>		
00E5 <sub>16</sub>		
00E6 <sub>16</sub>		
00E7 <sub>16</sub>		
00E8 <sub>16</sub>		
00E9 <sub>16</sub>	SI/O Receive Buffer Register0 (G0RB)	273
00EA <sub>16</sub>	Transmit Buffer/Receive Data Register 0 (G0TB/G0DR)	279
00EB <sub>16</sub>		
00EC <sub>16</sub>	Receive Input Register 0 (G0RI)	272
00ED <sub>16</sub>	SI/O Communication Mode Register 0 (G0MR)	274
00EE <sub>16</sub>	Transmit Output Register 0 (G0TO)	272
00EF <sub>16</sub>	SI/O Communication Control Register 0 (G0CR)	273

Address	Register	Page
00F0 <sub>16</sub>	Data Compare Register 00 (G0CMP0)	
00F1 <sub>16</sub>	Data Compare Register 01 (G0CMP1)	
00F2 <sub>16</sub>	Data Compare Register 02 (G0CMP2)	
00F3 <sub>16</sub>	Data Compare Register 03 (G0CMP3)	
00F4 <sub>16</sub>	Data Mask Register 00 (G0MSK0)	
00F5 <sub>16</sub>	Data Mask Register 01 (G0MSK1)	
00F6 <sub>16</sub>	Communication Clock Select Register (CCS)	281
00F7 <sub>16</sub>		
00F8 <sub>16</sub>		
00F9 <sub>16</sub>	Receive CRC Code Register 0 (G0RCRC)	
00FA <sub>16</sub>		
00FB <sub>16</sub>	Transmit CRC Code Register 0 (G0TCRC)	
00FC <sub>16</sub>	SI/O Extended Mode Register 0 (G0EMR)	275
00FD <sub>16</sub>	SI/O Extended Receive Control Register 0 (G0ERC)	272
00FE <sub>16</sub>	SI/O Special Communication Interrupt Detect Register 0 (G0IRF)	278
00FF <sub>16</sub>	SI/O Extended Transmit Control Register 0 (G0ETC)	276
0100 <sub>16</sub>	Time Measurement Register 10 (G1TM0)/	
0101 <sub>16</sub>	Waveform Generating Register 10 (G1PO0)	
0102 <sub>16</sub>	Time Measurement Register 11 (G1TM1)/	
0103 <sub>16</sub>	Waveform Generating Register 11 (G1PO1)	
0104 <sub>16</sub>	Time Measurement Register 12 (G1TM2)/	
0105 <sub>16</sub>	Waveform Generating Register 12 (G1PO2)	
0106 <sub>16</sub>	Time Measurement Register 13 (G1TM3)/	
0107 <sub>16</sub>	Waveform Generating Register 13 (G1PO3)	
0108 <sub>16</sub>	Time Measurement Register 14 (G1TM4)/	
0109 <sub>16</sub>	Waveform Generating Register 14 (G1PO4)	
010A <sub>16</sub>	Time Measurement Register 15 (G1TM5)/	
010B <sub>16</sub>	Waveform Generating Register 16 (G1PO5)	
010C <sub>16</sub>	Time Measurement Register 16 (G1TM6)/	
010D <sub>16</sub>	Waveform Generating Register 16 (G1PO6)	
010E <sub>16</sub>	Time Measurement Register 17 (G1TM7)/	
010F <sub>16</sub>	Waveform Generating Register 17 (G1PO7)	
0110 <sub>16</sub>	Waveform Generating Control Register 10 (G1POCR0)	
0111 <sub>16</sub>	Waveform Generating Control Register 11 (G1POCR1)	
0112 <sub>16</sub>	Waveform Generating Control Register 12 (G1POCR2)	
0113 <sub>16</sub>	Waveform Generating Control Register 13 (G1POCR3)	
0114 <sub>16</sub>	Waveform Generating Control Register 14 (G1POCR4)	
0115 <sub>16</sub>	Waveform Generating Control Register 15 (G1POCR5)	
0116 <sub>16</sub>	Waveform Generating Control Register 16 (G1POCR6)	
0117 <sub>16</sub>	Waveform Generating Control Register 17 (G1POCR7)	
0118 <sub>16</sub>	Time Measurement Control Register 10 (G1TMCR0)	
0119 <sub>16</sub>	Time Measurement Control Register 11 (G1TMCR1)	
011A <sub>16</sub>	Time Measurement Control Register 12 (G1TMCR2)	
011B <sub>16</sub>	Time Measurement Control Register 13 (G1TMCR3)	
011C <sub>16</sub>	Time Measurement Control Register 14 (G1TMCR4)	
011D <sub>16</sub>	Time Measurement Control Register 15 (G1TMCR5)	
011E <sub>16</sub>	Time Measurement Control Register 16 (G1TMCR6)	
011F <sub>16</sub>	Time Measurement Control Register 17 (G1TMCR7)	

Blank spaces are reserved. No access is allowed.

## Quick Reference by Address

Address	Register	Page
0120 <sub>16</sub>	Base Timer Register1 (G1BT)	248
0121 <sub>16</sub>		
0122 <sub>16</sub>	Base Timer Control Register 10 (G1BCR0)	249
0123 <sub>16</sub>	Base Timer Control Register 11 (G1BCR1)	
0124 <sub>16</sub>	Time Measurement Prescaler Register 16 (G1TPR6)	250
0125 <sub>16</sub>	Time Measurement Prescaler Register 17 (G1TPR7)	
0126 <sub>16</sub>	Function Enable Register 1 (G1FE)	253
0127 <sub>16</sub>	Function Select Register 1 (G1FS)	252
0128 <sub>16</sub>	SI/O Receive Buffer Register 1 (G1RB)	273
0129 <sub>16</sub>		
012A <sub>16</sub>	Transmit Buffer/Receive Data Register 1 (G1TB/G1DR)	279
012B <sub>16</sub>		
012C <sub>16</sub>	Receive Input Register 1 (G1RI)	272
012D <sub>16</sub>	SI/O Communication Mode Register 1 (G1MR)	274
012E <sub>16</sub>	Transmit Output Register 1 (G1TO)	272
012F <sub>16</sub>	SI/O Communication Control Register 1 (G1CR)	273
0130 <sub>16</sub>	Data Compare Register 10 (G1CMP0)	280
0131 <sub>16</sub>	Data Compare Register 11 (G1CMP1)	
0132 <sub>16</sub>	Data Compare Register 12 (G1CMP2)	
0133 <sub>16</sub>	Data Compare Register 13 (G1CMP3)	
0134 <sub>16</sub>	Data Mask Register 10 (G1MSK0)	
0135 <sub>16</sub>	Data Mask Register 11 (G1MSK1)	
0136 <sub>16</sub>		
0137 <sub>16</sub>		
0138 <sub>16</sub>	Receive CRC Code Register1 (G1RCRC)	280
0139 <sub>16</sub>		
013A <sub>16</sub>	Transmit CRC Code Register1 (G1TCRC)	
013B <sub>16</sub>		
013C <sub>16</sub>	SI/O Extended Mode Register 1 (G1EMR)	275
013D <sub>16</sub>	SI/O Extended Receive Control Register 1 (G1ERC)	277
013E <sub>16</sub>	SI/O Special Communication Interrupt Detect Register 1 (G1IRF)	279
013F <sub>16</sub>	SI/O Extended Transmit Control Register 1 (G1ETC)	276
0140 <sub>16</sub>		
0141 <sub>16</sub>		
0142 <sub>16</sub>		
0143 <sub>16</sub>		
0144 <sub>16</sub>		
0145 <sub>16</sub>		
0146 <sub>16</sub>		
0147 <sub>16</sub>		
0148 <sub>16</sub>		
0149 <sub>16</sub>		
014A <sub>16</sub>		
014B <sub>16</sub>		
014C <sub>16</sub>		
014D <sub>16</sub>		
to		
016F <sub>16</sub>		

Address	Register	Page
0170 <sub>16</sub>	CAN2 Slot Buffer Select Register (C2SBS)	331
0171 <sub>16</sub>	CAN2 Control Register 1 (C2CTLR1)	299
0172 <sub>16</sub>	CAN2 Sleep Control Register (C2SLPR)	300
0173 <sub>16</sub>		
0174 <sub>16</sub>	CAN2 Acceptance Filter Support Register (C2AFS)	336
0175 <sub>16</sub>		
0176 <sub>16</sub>		
0177 <sub>16</sub>		
0178 <sub>16</sub>	Input Function Select Register (IPS)	363
0179 <sub>16</sub>	Input Function Select Register A (IPSA)	364
017A <sub>16</sub>		
017B <sub>16</sub>		
017C <sub>16</sub>		
017D <sub>16</sub>		
017F <sub>16</sub>		
0180 <sub>16</sub>	CAN2 Message Slot Buffer 0 Standard ID0 (C2SLOT0_0)	332
0181 <sub>16</sub>	CAN2 Message Slot Buffer 0 Standard ID1 (C2SLOT0_1)	
0182 <sub>16</sub>	CAN2 Message Slot Buffer 0 Extended ID0 (C2SLOT0_2)	333
0183 <sub>16</sub>	CAN2 Message Slot Buffer 0 Extended ID1 (C2SLOT0_3)	
0184 <sub>16</sub>	CAN2 Message Slot Buffer 0 Extended ID2 (C2SLOT0_4)	334
0185 <sub>16</sub>	CAN2 Message Slot Buffer 0 Data Length Code (C2SLOT0_5)	
0186 <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 0 (C2SLOT0_6)	335
0187 <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 1 (C2SLOT0_7)	
0188 <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 2 (C2SLOT0_8)	
0189 <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 3 (C2SLOT0_9)	
018A <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 4 (C2SLOT0_10)	
018B <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 5 (C2SLOT0_11)	
018C <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 6 (C2SLOT0_12)	
018D <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 7 (C2SLOT0_13)	
018E <sub>16</sub>	CAN2 Message Slot Buffer 0 Time Stamp High-Order (C2SLOT0_14)	
018F <sub>16</sub>	CAN2 Message Slot Buffer 0 Time Stamp Low-Order (C2SLOT0_15)	
0190 <sub>16</sub>	CAN2 Message Slot Buffer 1 Standard ID0 (C2SLOT1_0)	332
0191 <sub>16</sub>	CAN2 Message Slot Buffer 1 Standard ID1 (C2SLOT1_1)	
0192 <sub>16</sub>	CAN2 Message Slot Buffer 1 Extended ID0 (C2SLOT1_2)	333
0193 <sub>16</sub>	CAN2 Message Slot Buffer 1 Extended ID1 (C2SLOT1_3)	
0194 <sub>16</sub>	CAN2 Message Slot Buffer 1 Extended ID2 (C2SLOT1_4)	334
0195 <sub>16</sub>	CAN2 Message Slot Buffer 1 Data Length Code (C2SLOT1_5)	
0196 <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 0 (C2SLOT1_6)	335
0197 <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 1 (C2SLOT1_7)	
0198 <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 2 (C2SLOT1_8)	
0199 <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 3 (C2SLOT1_9)	
019A <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 4 (C2SLOT1_10)	
019B <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 5 (C2SLOT1_11)	
019C <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 6 (C2SLOT1_12)	
019D <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 7 (C2SLOT1_13)	
019E <sub>16</sub>	CAN2 Message Slot Buffer 1 Time Stamp High-Order (C2SLOT1_14)	
019F <sub>16</sub>	CAN2 Message Slot Buffer 1 Time Stamp Low-Order (C2SLOT1_15)	

Blank spaces are reserved. No access is allowed.

## Quick Reference by Address

Address	Register	Page
01A0 <sub>16</sub> 01A1 <sub>16</sub>	CAN2 Control Register0 (C2CTRL0)	296
01A2 <sub>16</sub> 01A3 <sub>16</sub>	CAN2 Status Register (C2STR)	301
01A4 <sub>16</sub> 01A5 <sub>16</sub>	CAN2 Extended ID Register (C2IDR)	304
01A6 <sub>16</sub> 01A7 <sub>16</sub>	CAN2 Configuration Register (C2CONR)	305
01A8 <sub>16</sub> 01A9 <sub>16</sub>	CAN2 Time Stamp Register (C2TSR)	308
01AA <sub>16</sub>	CAN2 Transmit Error Count Register (C2TEC)	309
01AB <sub>16</sub>	CAN2 Receive Error Count Register (C2REC)	333
01AC <sub>16</sub> 01AD <sub>16</sub>	CAN2 Slot Interrupt Status Register (C2SISTR)	310
01AE <sub>16</sub>		
01AF <sub>16</sub>		
01B0 <sub>16</sub> 01B1 <sub>16</sub>	CAN2 Slot Interrupt Mask Register (C2SIMKR)	312
01B2 <sub>16</sub>		
01B3 <sub>16</sub>		
01B4 <sub>16</sub>	CAN2 Error Interrupt Mask Register (C2EIMKR)	313
01B5 <sub>16</sub>	CAN2 Error Interrupt Status Register (C2EISTR)	314
01B6 <sub>16</sub>	CAN2 Error Cause Register (C2EFR)	315
01B7 <sub>16</sub>	CAN2 Baud Rate Prescaler (C2BRP)	307
01B8 <sub>16</sub>		
01B9 <sub>16</sub>	CAN2 Mode Register (C2MDR)	316
01BA <sub>16</sub>		
01BB <sub>16</sub>		
01BC <sub>16</sub>		
01BD <sub>16</sub>		
01BE <sub>16</sub>		
01BF <sub>16</sub>		
01C0 <sub>16</sub> 01C1 <sub>16</sub>	CAN2 Single Shot Control Register (C2SSCTLR)	318
01C2 <sub>16</sub>		
01C3 <sub>16</sub>		
01C4 <sub>16</sub> 01C5 <sub>16</sub>	CAN2 Single Shot Status Register (C2SSSTR)	319
01C6 <sub>16</sub>		
01C7 <sub>16</sub>		
01C8 <sub>16</sub>	CAN2 Global Mask Register Standard ID0 (C2GMR0)	320
01C9 <sub>16</sub>	CAN2 Global Mask Register Standard ID1 (C2GMR1)	321
01CA <sub>16</sub>	CAN2 Global Mask Register Extended ID0 (C2GMR2)	322
01CB <sub>16</sub>	CAN2 Global Mask Register Extended ID1 (C2GMR3)	323
01CC <sub>16</sub>	CAN2 Global Mask Register Extended ID2 (C2GMR4)	324
01CD <sub>16</sub>		
01CE <sub>16</sub>		
01CF <sub>16</sub>		

Address	Register	Page
01D0 <sub>16</sub>	CAN2 Message Slot 0 Control Register (C2MCTL0)/ CAN2 Local Mask Register A Standard ID0 (C2LMAR0)	327/ 320
01D1 <sub>16</sub>	CAN2 Message Slot 1 Control Register (C2MCTL1)/ CAN2Local Mask Register A Standard ID1 (C2LMAR1)	327/ 321
01D2 <sub>16</sub>	CAN2 Message Slot 2 Control Register (C2MCTL2)/ CAN2 Local Mask Register A Extended ID0 (C2LMAR2)	327/ 322
01D3 <sub>16</sub>	CAN2 Message Slot 3 Control Register (C2MCTL3)/ CAN2 Local Mask Register A Extended ID1 (C2LMAR3)	327/ 323
01D4 <sub>16</sub>	CAN2 Message Slot 4 Control Register (C2MCTL4)/ CAN2 Local Mask Register A Extended ID2 (C2LMAR4)	327/ 324
01D5 <sub>16</sub>	CAN2 Message Slot 5 Control Register (C2MCTL5)	
01D6 <sub>16</sub>	CAN2 Message Sot 6 Control Register (C2MCTL6)	327
01D7 <sub>16</sub>	CAN2 Message Slot 7 Control Register (C2MCTL7)	
01D8 <sub>16</sub>	CAN2 Message Slot 8 Control register (C2MCTL8)/ CAN2 Local Mask Register B Standard ID0 (C2LMBR0)	327/ 320
01D9 <sub>16</sub>	CAN2 Message Slot 9 Control register (C2MCTL9)/ CAN2 Local Mask Register B Standard ID1 (C2LMBR1)	327/ 321
01DA <sub>16</sub>	CAN2 Message Slot 10 Control register (C2MCTL10)/ CAN2 Local Mask Register B Standard ID2 (C2LMBR2)	327/ 322
01DB <sub>16</sub>	CAN2 Message Slot 11 Control register (C2MCTL11)/ CAN2 Local Mask Register B Standard ID3 (C2LMBR3)	327/ 323
01DC <sub>16</sub>	CAN2 Message Slot 12 Control register (C2MCTL12)/ CAN2 Local Mask Register B Standard ID4 (C2LMBR4)	327/ 324
01DD <sub>16</sub>	CAN2 Message Slot 13 Control Register (C2MCTL13)	
01DE <sub>16</sub>	CAN2 Message Sot 14 Control Register (C2MCTL14)	327
01DF <sub>16</sub>	CAN2 Message Slot 15 Control Register (C2MCTL15)	
01E0 <sub>16</sub>	CAN0 Message Slot Buffer 0 Standard ID0 (C0SLOT0_0)	332
01E1 <sub>16</sub>	CAN0 Message Slot Buffer 0 Standard ID1 (C0SLOT0_1)	
01E2 <sub>16</sub>	CAN0 Message Slot Buffer 0 Extended ID0 (C0SLOT0_2)	333
01E3 <sub>16</sub>	CAN0 Message Slot Buffer 0 Extended ID1 (C0SLOT0_3)	
01E4 <sub>16</sub>	CAN0 Message Slot Buffer 0 Extended ID2 (C0SLOT0_4)	334
01E5 <sub>16</sub>	CAN0 Message Slot Buffer 0 Data Length Code (C0SLOT0_5)	
01E6 <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 0 (C0SLOT0_6)	
01E7 <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 1 (C0SLOT0_7)	
01E8 <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 2 (C0SLOT0_8)	
01E9 <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 3 (C0SLOT0_9)	
01EA <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 4 (C0SLOT0_10)	335
01EB <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 5 (C0SLOT0_11)	
01EC <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 6 (C0SLOT0_12)	
01ED <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 7 (C0SLOT0_13)	
01EE <sub>16</sub>	CAN0 Message Slot Buffer 0 Time Stamp High-Order (C0SLOT0_14)	
01EF <sub>16</sub>	CAN0 Message Slot Buffer 0 Time Stamp Low-Order (C0SLOT0_15)	

Blank spaces are reserved. No access is allowed.

## Quick Reference by Address

Address	Register	Page	Address	Register	Page
01F0 <sub>16</sub>	CAN0 Message Slot Buffer 1 Standard ID0 (C0SLOT1_0)	332	0220 <sub>16</sub>	CAN0 Single Shot Control Register (C0SSCTLR)	318
01F1 <sub>16</sub>	CAN0 Message Slot Buffer 1 Standard ID1 (C0SLOT1_1)				
01F2 <sub>16</sub>	CAN0 Message Slot Buffer 1 Extended ID0 (C0SLOT1_2)	333	0222 <sub>16</sub>		
01F3 <sub>16</sub>	CAN0 Message Slot Buffer 1 Extended ID1 (C0SLOT1_3)				
01F4 <sub>16</sub>	CAN0 Message Slot Buffer 1 Extended ID2 (C0SLOT1_4)	334	0224 <sub>16</sub>	CAN0 Single Shot Status Register (C0SSSTR)	319
01F5 <sub>16</sub>	CAN0 Message Slot Buffer 1 Data Length Code (C0SLOT1_5)				
01F6 <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 0 (C0SLOT1_6)	335	0226 <sub>16</sub>		
01F7 <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 1 (C0SLOT1_7)				
01F8 <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 2 (C0SLOT1_8)				
01F9 <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 3 (C0SLOT1_9)				
01FA <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 4 (C0SLOT1_10)				
01FB <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 5 (C0SLOT1_11)				
01FC <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 6 (C0SLOT1_12)				
01FD <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 7 (C0SLOT1_13)				
01FE <sub>16</sub>	CAN0 Message Slot Buffer 1 Time Stamp High-Order (C0SLOT1_14)				
01FF <sub>16</sub>	CAN0 Message Slot Buffer 1 Time Stamp Low-Order (C0SLOT1_15)				
0200 <sub>16</sub>	CAN0 Control Register0 (C0CTRL0)		296	0228 <sub>16</sub>	CAN0 Global Mask Register Standard ID0 (C0GMR0)
0201 <sub>16</sub>					
0202 <sub>16</sub>	CAN0 Status Register (C0STR)	301	0229 <sub>16</sub>	CAN0 Global Mask Register Standard ID1 (C0GMR1)	321
0203 <sub>16</sub>					
0204 <sub>16</sub>	CAN0 Extended ID Register (C0IDR)	304	022A <sub>16</sub>	CAN0 Global Mask Register Extended ID0 (C0GMR2)	322
0205 <sub>16</sub>					
0206 <sub>16</sub>	CAN0 Configuration Register (C0CONR)	305	022B <sub>16</sub>	CAN0 Global Mask Register Extended ID1 (C0GMR3)	323
0207 <sub>16</sub>					
0208 <sub>16</sub>	CAN0 Time Stamp Register (C0TSR)	308	022C <sub>16</sub>	CAN0 Global Mask Register Extended ID2 (C0GMR4)	324
0209 <sub>16</sub>					
020A <sub>16</sub>	CAN0 Transmit Error Count Register (C0TEC)	309	022D <sub>16</sub>		
020B <sub>16</sub>	CAN0 Receive Error Count Register (C0REC)		022E <sub>16</sub>		
020C <sub>16</sub>	CAN0 Slot Interrupt Status Register (C0SISTR)	310	022F <sub>16</sub>		
020D <sub>16</sub>					
020E <sub>16</sub>			0230 <sub>16</sub>	CAN0 Message Slot 0 Control Register (C0MCTL0)/ CAN0 Local Mask Register A Standard ID0 (C0LMAR0)	327/ 320
020F <sub>16</sub>	CAN0 Slot Interrupt Mask Register (C0SIMKR)	312	0231 <sub>16</sub>	CAN0 Message Slot 1 Control Register (C0MCTL1)/ CAN0 Local Mask Register A Standard ID1 (C0LMAR1)	327/ 321
0210 <sub>16</sub>					
0211 <sub>16</sub>					
0212 <sub>16</sub>					
0213 <sub>16</sub>			0232 <sub>16</sub>	CAN0 Message Slot 2 Control Register (C0MCTL2)/ CAN0 Local Mask Register A Extended ID0 (C0LMAR2)	327/ 322
0214 <sub>16</sub>	CAN0 Error Interrupt Mask Register (C0EIMKR)	313	0233 <sub>16</sub>	CAN0 Message Slot 3 Control Register (C0MCTL3)/ CAN0 Local Mask Register A Extended ID1 (C0LMAR3)	327/ 323
0215 <sub>16</sub>	CAN0 Error Interrupt Status Register (C0EISTR)	314	0234 <sub>16</sub>	CAN0 Message Slot 4 Control Register (C0MCTL4)/ CAN0 Local Mask Register A Extended ID2 (C0LMAR4)	327/ 324
0216 <sub>16</sub>	CAN0 Error Cause Register (C0EFR)	315	0235 <sub>16</sub>	CAN0 Message Slot 5 Control Register (C0MCTL5)	
0217 <sub>16</sub>	CAN0 Baud Rate Prescaler (C0BRP)	307	0236 <sub>16</sub>	CAN0 Message Slot 6 Control Register (C0MCTL6)	327
0218 <sub>16</sub>			0237 <sub>16</sub>	CAN0 Message Slot 7 Control Register (C0MCTL7)	
0219 <sub>16</sub>	CAN0 Mode Register (C0MDR)	316	0238 <sub>16</sub>	CAN0 Message Slot 8 Control Register (C0MCTL8)/ CAN0 Local Mask Register B Standard ID0 (C0LMBR0)	327/ 320
021A <sub>16</sub>			0239 <sub>16</sub>	CAN0 Message Slot 9 Control Register (C0MCTL9)/ CAN0 Local Mask Register B Standard ID1 (C0LMBR1)	327/ 321
021B <sub>16</sub>			023A <sub>16</sub>	CAN0 Message Slot 10 Control Register (C0MCTL10)/ CAN0 Local Mask Register B Extended ID0 (C0LMBR2)	327/ 322
021C <sub>16</sub>			023B <sub>16</sub>	CAN0 Message Slot 11 Control Register (C0MCTL11)/ CAN0 Local Mask Register B Extended ID1 (C0LMBR3)	327/ 323
021D <sub>16</sub>			023C <sub>16</sub>	CAN0 Message Slot 12 Control Register (C0MCTL12)/ CAN0 Local Mask Register B Extended ID2 (C0LMBR4)	327/ 324
021E <sub>16</sub>			023D <sub>16</sub>	CAN0 Message Slot 13 Control Register (C0MCTL13)	
021F <sub>16</sub>			023E <sub>16</sub>	CAN0 Message Slot 14 Control Register (C0MCTL14)	327
			023F <sub>16</sub>	CAN0 Message Slot 15 Control Register (C0MCTL15)	

Blank spaces are reserved. No access is allowed.

## Quick Reference by Address

Address	Register	Page	Address	Register	Page
0240 <sub>16</sub>	CAN0 Slot Buffer Select Register (C0SBS)	331	0270 <sub>16</sub>	CAN1 Message Slot Buffer 1 Standard ID0 (C1SLOT1_0)	332
0241 <sub>16</sub>	CAN0 Control Register 1 (C0CTLR1)	299	0271 <sub>16</sub>	CAN1 Message Slot Buffer 1 Standard ID1 (C1SLOT1_1)	
0242 <sub>16</sub>	CAN0 Sleep Control Register (C0SLPR)	300	0272 <sub>16</sub>	CAN1 Message Slot Buffer 1 Extended ID0 (C1SLOT1_2)	333
0243 <sub>16</sub>			0273 <sub>16</sub>	CAN1 Message Slot Buffer 1 Extended ID1 (C1SLOT1_3)	
0244 <sub>16</sub>	CAN0 Acceptance Filter Support Register (C0AFS)	336	0274 <sub>16</sub>	CAN1 Message Slot Buffer 1 Extended ID2 (C1SLOT1_4)	334
0245 <sub>16</sub>				0275 <sub>16</sub>	
0246 <sub>16</sub>			0276 <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 0 (C1SLOT1_6)	335
0247 <sub>16</sub>			0277 <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 1 (C1SLOT1_7)	
0248 <sub>16</sub>			0278 <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 2 (C1SLOT1_8)	
0249 <sub>16</sub>			0279 <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 3 (C1SLOT1_9)	
024A <sub>16</sub>			027A <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 4 (C1SLOT1_10)	
024B <sub>16</sub>			027B <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 5 (C1SLOT1_11)	
024C <sub>16</sub>			027C <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 6 (C1SLOT1_12)	
024D <sub>16</sub>			027D <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 7 (C1SLOT1_13)	
024E <sub>16</sub>			027E <sub>16</sub>	CAN1 Message Slot Buffer 1 Time Stamp High-Order (C1SLOT1_14)	
024F <sub>16</sub>			027F <sub>16</sub>	CAN1 Message Slot Buffer 1 Time Stamp Low-Order (C1SLOT1_15)	
0250 <sub>16</sub>	CAN1 Slot Buffer Select Register (C1SBS)	331	0280 <sub>16</sub>	CAN1 Control Register0 (C1CTLR0)	296
0251 <sub>16</sub>	CAN1 Control Register 1 (C1CTLR1)	299	0281 <sub>16</sub>		
0252 <sub>16</sub>	CAN1 Sleep Control Register (C1SLPR)	300	0282 <sub>16</sub>	CAN1 Status Register (C1STR)	301
0253 <sub>16</sub>			0283 <sub>16</sub>		
0254 <sub>16</sub>	CAN1 Acceptance Filter Support Register (C1AFS)	336	0284 <sub>16</sub>	CAN1 Extended ID Register (C1IDR)	304
0255 <sub>16</sub>				0285 <sub>16</sub>	
0256 <sub>16</sub>			0286 <sub>16</sub>	CAN1 Configuration Register (C1CONR)	305
0257 <sub>16</sub>			0287 <sub>16</sub>		
0258 <sub>16</sub>			0288 <sub>16</sub>	CAN1 Time Stamp Register (C1TSR)	308
0259 <sub>16</sub>			0289 <sub>16</sub>		
025A <sub>16</sub>			028A <sub>16</sub>	CAN1 Transmit Error Count Register (C1TEC)	309
025B <sub>16</sub>			028B <sub>16</sub>	CAN1 Receive Error Count Register (C1REC)	
025C <sub>16</sub>			028C <sub>16</sub>	CAN1 Slot Interrupt Control Register (C1SISTR)	310
025D <sub>16</sub>			028D <sub>16</sub>		
025E <sub>16</sub>			028E <sub>16</sub>		
025F <sub>16</sub>			028F <sub>16</sub>		
0260 <sub>16</sub>	CAN1 Message Slot Buffer 0 Standard ID0 (C1SLOT0_0)	332	0290 <sub>16</sub>	CAN1 Slot Interrupt Mask Register (C1SIMKR)	312
0261 <sub>16</sub>	CAN1 Message Slot Buffer 0 Standard ID1 (C1SLOT0_1)		0291 <sub>16</sub>		
0262 <sub>16</sub>	CAN1 Message Slot Buffer 0 Extended ID0 (C1SLOT0_2)	333	0292 <sub>16</sub>		
0263 <sub>16</sub>	CAN1 Message Slot Buffer 0 Extended ID1 (C1SLOT0_3)		0293 <sub>16</sub>		
0264 <sub>16</sub>	CAN1 Message Slot Buffer 0 Extended ID2 (C1SLOT0_4)	334	0294 <sub>16</sub>	CAN1 Error Interrupt Mask Register (C1EIMKR)	313
0265 <sub>16</sub>	CAN1 Message Slot Buffer 0 Data Length Code (C1SLOT0_5)		0295 <sub>16</sub>	CAN1 Error Interrupt Status Register (C1EISTR)	314
0266 <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 0 (C1SLOT0_6)	335	0296 <sub>16</sub>	CAN1 Error Factor Register (C1EFR)	315
0267 <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 1 (C1SLOT0_7)		0297 <sub>16</sub>	CAN1 Baud Rate Prescaler (C1BRP)	307
0268 <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 2 (C1SLOT0_8)		0298 <sub>16</sub>		
0269 <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 3 (C1SLOT0_9)		0299 <sub>16</sub>	CAN1 Mode Register (C1MDR)	316
026A <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 4 (C1SLOT0_10)		029A <sub>16</sub>		
026B <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 5 (C1SLOT0_11)		029B <sub>16</sub>		
026C <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 6 (C1SLOT0_12)		029C <sub>16</sub>		
026D <sub>16</sub>	CAN1 message Slot Buffer 0 Data 7 (C1SLOT0_13)		029D <sub>16</sub>		
026E <sub>16</sub>	CAN1 Message Slot Buffer 0 Time Stamp High-Order (C1SLOT0_14)		029E <sub>16</sub>		
026F <sub>16</sub>	CAN1 Message Slot Buffer 0 Time Stamp Low-Order (C1SLOT0_15)		029F <sub>16</sub>		

Blank spaces are reserved. No access is allowed.

## Quick Reference by Address

Address	Register	Page	Address	Register	Page
02A0 <sub>16</sub>	CAN1 Single Shot Control Register (C1SSCTLR)	318	02C0 <sub>16</sub>	X0 Register Y0 Register (X0R,Y0R)	242
02A1 <sub>16</sub>			02C1 <sub>16</sub>		
02A2 <sub>16</sub>			02C2 <sub>16</sub>	X1 Register Y1 Register (X1R,Y1R)	
02A3 <sub>16</sub>	02C3 <sub>16</sub>				
02A4 <sub>16</sub>	CAN1 Single Shot Status Register (C1SSSTR)	319	02C4 <sub>16</sub>	X2 Register Y2 Register (X2R,Y2R)	
02A5 <sub>16</sub>			02C5 <sub>16</sub>		
02A6 <sub>16</sub>			02C6 <sub>16</sub>	X3 Register Y3 Register (X3R,Y3R)	
02A7 <sub>16</sub>	02C7 <sub>16</sub>				
02A8 <sub>16</sub>	CAN1 Global Mask Register Standard ID0 (C1GMR0)	320	02C8 <sub>16</sub>	X4 Register Y4 Register (X4R,Y4R)	
02A9 <sub>16</sub>	CAN1 Global Mask Register Standard ID1 (C1GMR1)	321	02C9 <sub>16</sub>		
02AA <sub>16</sub>	CAN1 Global Mask Register Extended ID0 (C1GMR2)	322	02CA <sub>16</sub>	X5 Register Y5 Register (X5R,Y5R)	
02AB <sub>16</sub>	CAN1 Global Mask Register Extended ID1 (C1GMR3)	323	02CB <sub>16</sub>		
02AC <sub>16</sub>	CAN1 Global Mask Register Extended ID2 (C1GMR4)	324	02CC <sub>16</sub>	X6 Register Y6 Register (X6R,Y6R)	
02AD <sub>16</sub>			02CD <sub>16</sub>		
02AE <sub>16</sub>			02CE <sub>16</sub>	X7 Register Y7 Register (X7R,Y7R)	
02AF <sub>16</sub>			02CF <sub>16</sub>		
02B0 <sub>16</sub>	CAN1 Message Slot 0 Control Register (C1MCTL0)/ CAN1 Local Mask Register A Standard ID0 (C1LMAR0)	327/ 320	02D0 <sub>16</sub>	X8 Register Y8 Register (X8R,Y8R)	
02B1 <sub>16</sub>	CAN1 Message Slot 1 Control Register (C1MCTL1)/ CAN1 Local Mask Register A Standard ID1 (C1LMAR1)	327/ 321	02D1 <sub>16</sub>		
02B2 <sub>16</sub>	CAN1 Message Slot 2 Control Register (C1MCTL2)/ CAN1 Local Mask Register A Extended ID0 (C1LMAR2)	327/ 322	02D2 <sub>16</sub>	X9 Register Y9 Register (X9R,Y9R)	
02B3 <sub>16</sub>	CAN1 Message Slot 3 Control Register (C1MCTL3)/ CAN1 Local Mask Register A Extended ID1 (C1LMAR3)	327/ 323	02D3 <sub>16</sub>		
02B4 <sub>16</sub>	CAN1 Message Slot 4 Control Register (C1MCTL4)/ CAN1 Local Mask Register A Extended ID2 (C1LMAR4)	327/ 324	02D4 <sub>16</sub>	X10 Register Y10 Register (X10R,Y10R)	
02B5 <sub>16</sub>	CAN1 Message Slot 5 Control Register (C1MCTL5)	327	02D5 <sub>16</sub>		
02B6 <sub>16</sub>	CAN1 Message Slot 6 Control Register (C1MCTL6)				
02B7 <sub>16</sub>	CAN1 Message Slot 7 Control Register (C1MCTL7)				
02B8 <sub>16</sub>	CAN1 Message Slot 8 Control Register (C1MCTL8)/ CAN1 Local Mask Register B Standard ID0 (C1LMBR0)	327/ 320	02D6 <sub>16</sub>	X11 Register Y11 Register (X11R,Y11R)	
02B9 <sub>16</sub>	CAN1 Message Slot 9 Control Register (C1MCTL9)/ CAN1 Local Mask Register B Standard ID1 (C1LMBR1)	353/ 321	02D7 <sub>16</sub>		
02BA <sub>16</sub>	CAN1 Message Slot 10 Control Register (C1MCTL10)/ CAN1 Local Mask Register B Extended ID0 (C1LMBR2)	327/ 322	02D8 <sub>16</sub>	X12 Register Y12 Register (X12R,Y12R)	
02BB <sub>16</sub>	CAN1 Message Slot 11 Control Register (C1MCTL11)/ CAN1 Local Mask Register B Extended ID1 (C1LMBR3)	327/ 323	02D9 <sub>16</sub>		
02BC <sub>16</sub>	CAN1 Message Slot 12 Control Register (C1MCTL12)/ CAN1 Local Mask Register B Extended ID2 (C1LMBR4)	327/ 324	02DA <sub>16</sub>	X13 Register Y13 Register (X13R,Y13R)	
02BD <sub>16</sub>	CAN1 Message Slot 13 Control Register (C1MCTL13)	327	02DB <sub>16</sub>		
02BE <sub>16</sub>	CAN1 Message Slot 14 Control Register (C1MCTL14)				
02BF <sub>16</sub>	CAN1 Message Slot 15 Control Register (C1MCTL15)				
			02DC <sub>16</sub>	X14 Register Y14 Register (X14R,Y14R)	
			02DD <sub>16</sub>		
			02DE <sub>16</sub>	X15 Register Y15 Register (X15R,Y15R)	
			02DF <sub>16</sub>		

Blank spaces are reserved. No access is allowed.

## Quick Reference by Address

Address	Register	Page	
02E0 <sub>16</sub>	X/Y Control Register (XYC)	242	
02E1 <sub>16</sub>			
02E2 <sub>16</sub>			
02E3 <sub>16</sub>			
02E4 <sub>16</sub>	UART1 Special Mode Register 4 (U1SMR4)	173	
02E5 <sub>16</sub>	UART1 Special Mode Register 3 (U1SMR3)	172	
02E6 <sub>16</sub>	UART1 Special Mode Register 2 (U1SMR2)	171	
02E7 <sub>16</sub>	UART1 Special Mode Register (U1SMR)	170	
02E8 <sub>16</sub>	UART1 Transmit/Receive Mode Register (U1MR)	168	
02E9 <sub>16</sub>	UART1 Bit Rate Register (U1BRG)		
02EA <sub>16</sub>	UART1 Transmit Buffer Register (U1TB)	167	
02EB <sub>16</sub>			
02EC <sub>16</sub>	UART1 Transmit/Receive Control Register 0 (U1C0)	169	
02ED <sub>16</sub>	UART1 Transmit/Receive Control Register 1 (U1C1)	170	
02EE <sub>16</sub>	UART1 Receive Buffer Register (U1RB)	167	
02EF <sub>16</sub>			
02F0 <sub>16</sub>			
02F1 <sub>16</sub>			
02F2 <sub>16</sub>			
02F3 <sub>16</sub>			
02F4 <sub>16</sub>	UART4 Special Mode Register 4 (U4SMR4)	173	
02F5 <sub>16</sub>	UART4 Special Mode Register 3 (U4SMR3)	172	
02F6 <sub>16</sub>	UART4 Special Mode Register 2 (U4SMR2)	171	
02F7 <sub>16</sub>	UART4 Special Mode Register (U4SMR)	170	
02F8 <sub>16</sub>	UART4 Transmit/Receive Mode Register (U4MR)	168	
02F9 <sub>16</sub>	UART4 Bit Rate Register (U4BRG)		
02FA <sub>16</sub>	UART4 Transmit Buffer Register (U4TB)	167	
02FB <sub>16</sub>			
02FC <sub>16</sub>	UART4 Transmit/Receive Control Register 0 (U4C0)	169	
02FD <sub>16</sub>	UART4 Transmit/Receive Control Register 1 (U4C1)	170	
02FE <sub>16</sub>	UART4 Receive Buffer Register (U4RB)	167	
02FF <sub>16</sub>			
0300 <sub>16</sub>	Timer B3,B4,B5 Count Start Flag (TBSR)	147	
0301 <sub>16</sub>			
0302 <sub>16</sub>	Timer A1-1 Register (TA11)	160	
0303 <sub>16</sub>			
0304 <sub>16</sub>	Timer A2-1 Register (TA21)		
0305 <sub>16</sub>			
0306 <sub>16</sub>	Timer A4-1 Register (TA41)		
0307 <sub>16</sub>			
0308 <sub>16</sub>	Three-Phase PWM Control Register 0 (INVC0)	157	
0309 <sub>16</sub>	Three-Phase PWM Control Register 1 (INVC1)	158	
030A <sub>16</sub>	Three-Phase Output Buffer Register 0 (IDB0)	159	
030B <sub>16</sub>	Three-Phase Output Buffer Register 1 (IDB1)		
030C <sub>16</sub>	Dead Time Timer (DTT)		
030D <sub>16</sub>	Timer B2 Interrupt Generating Frequency Set Counter (ICTB2)	160	
030E <sub>16</sub>			
030F <sub>16</sub>			

Address	Register	Page	
0310 <sub>16</sub>	Timer B3 Register (TB3)	145	
0311 <sub>16</sub>			
0312 <sub>16</sub>	Timer B4 Register (TB4)		
0313 <sub>16</sub>			
0314 <sub>16</sub>	Timer B5 Register (TB5)		
0315 <sub>16</sub>			
0316 <sub>16</sub>			
0317 <sub>16</sub>			
0318 <sub>16</sub>			
0319 <sub>16</sub>			
031A <sub>16</sub>		146	
031B <sub>16</sub>	Timer B3 Mode Register (TB3MR)		
031C <sub>16</sub>	Timer B4 Mode Register (TB4MR)		
031D <sub>16</sub>	Timer B5 Mode Register (TB5MR)	96	
031E <sub>16</sub>			
031F <sub>16</sub>	External Interrupt Request Source Select Register (IFSR)		
0320 <sub>16</sub>			
0321 <sub>16</sub>			
0322 <sub>16</sub>			
0323 <sub>16</sub>			
0324 <sub>16</sub>	UART3 Special Mode Register 4 (U3SMR4)	173	
0325 <sub>16</sub>	UART3 Special Mode Register 3 (U3SMR3)	172	
0326 <sub>16</sub>	UART3 Special Mode Register 2 (U3SMR2)	171	
0327 <sub>16</sub>	UART3 Special Mode Register (U3SMR)	170	
0328 <sub>16</sub>	UART3 Transmit/Receive Mode Register (U3MR)	168	
0329 <sub>16</sub>	UART3 Bit Rate Register (U3BRG)		
032A <sub>16</sub>	UART3 Transmit Buffer Register (U3TB)	167	
032B <sub>16</sub>			
032C <sub>16</sub>	UART3 Transmit/Receive Control Register 0 (U3C0)	169	
032D <sub>16</sub>	UART3 Transmit/Receive Control Register 1 (U3C1)	170	
032E <sub>16</sub>	UART3 Receive Buffer Register (U3RB)	167	
032F <sub>16</sub>			
0330 <sub>16</sub>			
0331 <sub>16</sub>			
0332 <sub>16</sub>			
0333 <sub>16</sub>			
0334 <sub>16</sub>	UART2 Special Mode Register 4 (U2SMR4)	173	
0335 <sub>16</sub>	UART2 Special Mode Register 3 (U2SMR3)	172	
0336 <sub>16</sub>	UART2 Special Mode Register 2 (U2SMR2)	171	
0337 <sub>16</sub>	UART2 Special Mode Register (U2SMR)	170	
0338 <sub>16</sub>	UART2 Transmit/Receive Mode Register (U2MR)	168	
0339 <sub>16</sub>	UART2 Bit Rate Register (U2BRG)		
033A <sub>16</sub>	UART2 Transmit Buffer Register (U2TB)	167	
033B <sub>16</sub>			
033C <sub>16</sub>	UART2 Transmit/Receive Control Register 0 (U2C0)	169	
033D <sub>16</sub>	UART2 Transmit/Receive Control Register 1 (U2C1)	170	
033E <sub>16</sub>	UART2 Receive Buffer Register (U2RB)	167	
033F <sub>16</sub>			

Blank spaces are reserved. No access is allowed.

## Quick Reference by Address

Address	Register	Page	Address	Register	Page
0340 <sub>16</sub>	Count Start Flag (TABSR)	130	0370 <sub>16</sub>		
0341 <sub>16</sub>	Clock Prescaler Reset Flag (CPSRF)	60	0371 <sub>16</sub>		
0342 <sub>16</sub>	One-Shot Start Flag (ONSF)	131	0372 <sub>16</sub>		
0343 <sub>16</sub>	Trigger Select Register (TRGSR)	132	0373 <sub>16</sub>		
0344 <sub>16</sub>	Up/Down Flag (UDF)	131	0374 <sub>16</sub>		
0345 <sub>16</sub>			0375 <sub>16</sub>		
0346 <sub>16</sub>	Timer A0 Register (TA0)		0376 <sub>16</sub>		
0347 <sub>16</sub>				0377 <sub>16</sub>	
0348 <sub>16</sub>	Timer A1 Register (TA1)	129	0378 <sub>16</sub>	DMA0 Request Source Select Register (DM0SL)	109
0349 <sub>16</sub>				0379 <sub>16</sub>	
034A <sub>16</sub>	Timer A2 Register (TA2)		037A <sub>16</sub>	DMA2 Request Source Select Register (DM2SL)	
034B <sub>16</sub>			037B <sub>16</sub>	DMA3 Request Source Select Register (DM3SL)	
034C <sub>16</sub>	Timer A3 Register (TA3)	129	037C <sub>16</sub>	CRC Data Register (CRCD)	240
034D <sub>16</sub>					
034E <sub>16</sub>	Timer A4 Register (TA4)	129	037E <sub>16</sub>	CRC Input Register (CRCIN)	
034F <sub>16</sub>				037F <sub>16</sub>	
0350 <sub>16</sub>	Timer B0 Register (TB0)	145	0380 <sub>16</sub>	A/D0 Register0 (AD00)	225
0351 <sub>16</sub>					
0352 <sub>16</sub>	Timer B1 Register (TB1)	145	0382 <sub>16</sub>	A/D0 Register1 (AD01)	
0353 <sub>16</sub>					
0354 <sub>16</sub>	Timer B2 Register (TB2)	145	0384 <sub>16</sub>	A/D0 Register2 (AD02)	
0355 <sub>16</sub>					
0356 <sub>16</sub>	Timer A0 Mode Register (TA0MR)	130	0386 <sub>16</sub>	A/D0 Register3 (AD03)	
0357 <sub>16</sub>	Timer A1 Mode Register (TA1MR)		0387 <sub>16</sub>		
0358 <sub>16</sub>	Timer A2 Mode Register (TA2MR)		0388 <sub>16</sub>	A/D0 Register4 (AD04)	
0359 <sub>16</sub>	Timer A3 Mode Register (TA3MR)		0389 <sub>16</sub>		
035A <sub>16</sub>	Timer A4 Mode Register (TA4MR)		038A <sub>16</sub>	A/D0 Register5 (AD05)	
035B <sub>16</sub>	Timer B0 Mode Register (TB0MR)	038B <sub>16</sub>			
035C <sub>16</sub>	Timer B1 Mode Register (TB1MR)	146	038C <sub>16</sub>	A/D0 Register6 (AD06)	
035D <sub>16</sub>			Timer B2 Mode Register (TB2MR)		038D <sub>16</sub>
035E <sub>16</sub>	Timer B2 Special Mode Register (TB2SC)	160	038E <sub>16</sub>	A/D0 Register7 (AD07)	
035F <sub>16</sub>	Count Source Prescaler Register (TCSPR)	60	038F <sub>16</sub>		
0360 <sub>16</sub>			0390 <sub>16</sub>		
0361 <sub>16</sub>			0391 <sub>16</sub>		
0362 <sub>16</sub>			0392 <sub>16</sub>	A/D0 Control Register 4 (AD0CON4)	225
0363 <sub>16</sub>			0393 <sub>16</sub>		
0364 <sub>16</sub>	UART0 Special Mode Register 4 (U0SMR4)	173	0394 <sub>16</sub>	A/D0 Control Register 2 (AD0CON2)	224
0365 <sub>16</sub>	UART0 Special Mode Register 3 (U0SMR3)	172	0395 <sub>16</sub>	A/D0 Control Register 3 (AD0CON3)	223
0366 <sub>16</sub>	UART0 Special Mode Register 2 (U0SMR2)	171	0396 <sub>16</sub>	A/D0 Control Register 0 (AD0CON0)	222
0367 <sub>16</sub>	UART0 Special Mode Register (U0SMR)	170	0397 <sub>16</sub>	A/D0 Control Register 1 (AD0CON1)	221
0368 <sub>16</sub>	UART0 Transmit/Receive Mode Register (U0MR)	168	0398 <sub>16</sub>	D/A Register 0 (DA0)	239
0369 <sub>16</sub>	UART0 Bit Rate Register (U0BRG)		0399 <sub>16</sub>		
036A <sub>16</sub>	UART0 Transmit Buffer Register (U0TB)	167	039A <sub>16</sub>	D/A Register 1 (DA1)	239
036B <sub>16</sub>				039B <sub>16</sub>	
036C <sub>16</sub>	UART0 Transmit/Receive Control Register 0 (U0C0)	169	039C <sub>16</sub>	D/A Control Register (DACON)	239
036D <sub>16</sub>	UART0 Transmit/Receive Control Register 1 (U0C1)	170	039D <sub>16</sub>		
036E <sub>16</sub>	UART0 Receive Buffer Register (U0RB)	167	039E <sub>16</sub>		
036F <sub>16</sub>				039F <sub>16</sub>	

Blank spaces are reserved. No access is allowed.

## Quick Reference by Address

Address	Register	Page	Address	Register	Page	
03A0 <sub>16</sub>	Function Select Register A8 (PS8)	355	03D0 <sub>16</sub>	Port P14 Register (P14)	352	
03A1 <sub>16</sub>	Function Select Register A9 (PS9)	356	03D1 <sub>16</sub>	Port P15 Register (P15)		
03A2 <sub>16</sub>			03D2 <sub>16</sub>	Port P14 Direction Register (PD14)	351	
03A3 <sub>16</sub>						
03A4 <sub>16</sub>						
03A5 <sub>16</sub>						
03A6 <sub>16</sub>						
03A7 <sub>16</sub>	Function Select Register D1 (PSD1)	360	03D3 <sub>16</sub>	Port P15 Direction Register (PD15)	361	
03A8 <sub>16</sub>			03D4 <sub>16</sub>			
03A9 <sub>16</sub>						
03AA <sub>16</sub>						
03AB <sub>16</sub>						
03AC <sub>16</sub>	Function Select Register C2 (PSC2)	359	03DA <sub>16</sub>	Pull-Up Control Register 2 (PUR2)		361
03AD <sub>16</sub>	Function Select Register C3 (PSC3)	360	03DB <sub>16</sub>	Pull-Up Control Register 3 (PUR3)		362
03AE <sub>16</sub>			03DC <sub>16</sub>	Pull-Up Control Register 4 (PUR4)		
03AF <sub>16</sub>	Function Select Register C (PSC)		359	03DD <sub>16</sub>		
03B0 <sub>16</sub>	Function Select Register A0 (PS0)		353	03DE <sub>16</sub>		
03B1 <sub>16</sub>	Function Select Register A1 (PS1)	357		03DF <sub>16</sub>		
03B2 <sub>16</sub>	Function Select Register B0 (PSL0)		354	03E0 <sub>16</sub>	Port P0 Register (P0)	352
03B3 <sub>16</sub>	Function Select Register B1 (PSL1)	358		03E1 <sub>16</sub>	Port P1 Register (P1)	
03B4 <sub>16</sub>	Function Select Register A2 (PS2)		355	03E2 <sub>16</sub>	Port P0 Direction Register (PD0)	351
03B5 <sub>16</sub>	Function Select Register A3 (PS3)	351		03E3 <sub>16</sub>	Port P1 Direction Register (PD1)	
03B6 <sub>16</sub>	Function Select Register B2 (PSL2)		351	03E4 <sub>16</sub>	Port P2 Register (P2)	352
03B7 <sub>16</sub>	Function Select Register B3 (PSL3)	351		03E5 <sub>16</sub>	Port P3 Register (P3)	
03B8 <sub>16</sub>			352	03E6 <sub>16</sub>	Port P2 Direction Register (PD2)	351
03B9 <sub>16</sub>	Function Select Register A5 (PS5)	351		03E7 <sub>16</sub>	Port P3 Direction Register (PD3)	
03BA <sub>16</sub>			352	03E8 <sub>16</sub>	Port P4 Register (P4)	351
03BB <sub>16</sub>		351		03E9 <sub>16</sub>	Port P5 Register (P5)	
03BC <sub>16</sub>			351	03EA <sub>16</sub>	Port P4 Direction Register (PD4)	361
03BD <sub>16</sub>		351		03EB <sub>16</sub>	Port P5 Direction Register (PD5)	
03BE <sub>16</sub>			351	03EC <sub>16</sub>		
03BF <sub>16</sub>		351		03ED <sub>16</sub>		
03C0 <sub>16</sub>	Port P6 Register (P6)		352	03EE <sub>16</sub>		
03C1 <sub>16</sub>	Port P7 Register (P7)	351		03EF <sub>16</sub>		
03C2 <sub>16</sub>	Port P6 Direction Register (PD6)		352	03F0 <sub>16</sub>	Pull-up Control Register 0 (PUR0)	361
03C3 <sub>16</sub>	Port P7 Direction Register (PD7)	351		03F1 <sub>16</sub>	Pull-up Control Register 1 (PUR1)	
03C4 <sub>16</sub>	Port P8 Register (P8)		351	03F2 <sub>16</sub>		
03C5 <sub>16</sub>	Port P9 Register (P9)	351		03F3 <sub>16</sub>		
03C6 <sub>16</sub>	Port P8 Direction Register (PD8)		352	03F4 <sub>16</sub>		
03C7 <sub>16</sub>	Port P9 Direction Register (PD9)	351		03F5 <sub>16</sub>		
03C8 <sub>16</sub>	Port P10 Register (P10)		351	03F6 <sub>16</sub>		
03C9 <sub>16</sub>	Port P11 Register (P11)	351		03F7 <sub>16</sub>		
03CA <sub>16</sub>	Port P10 Direction Register (PD10)		351	03F8 <sub>16</sub>		
03CB <sub>16</sub>	Port P11 Direction Register (PD11)	352		03F9 <sub>16</sub>		
03CC <sub>16</sub>	Port P12 Register (P12)		351	03FA <sub>16</sub>		
03CD <sub>16</sub>	Port P13 Register (P13)	351		03FB <sub>16</sub>		
03CE <sub>16</sub>	Port P12 Direction Register (PD12)		363	03FC <sub>16</sub>		
03CF <sub>16</sub>	Port P13 Direction Register (PD13)			03FD <sub>16</sub>		
			03FE <sub>16</sub>			
			03FF <sub>16</sub>	Port Control Register (PCR)		

Blank spaces are reserved. No access is allowed.

# M32C/88 Group (M32C/88T)

## SINGLE-CHIP 16/32-BIT CMOS MICROCOMPUTER

---

## 1. Overview

The M32C/88 Group (M32C/88T) microcomputer is a single-chip control unit that utilizes high-performance silicon gate CMOS technology with the M32C/80 Series CPU core. The M32C/88 Group (M32C/88T) is available in 144-pin and 100-pin plastic molded LQFP packages.

With a 16-Mbyte address space, this microcomputer combines advanced instruction manipulation capabilities to process complex instructions by less bytes and execute instructions at higher speed.

It includes a multiplier and DMAC adequate for office automation, communication devices and industrial equipments, and other high-speed processing applications.

### 1.1 Applications

Automobiles, audio, cameras, office equipment, communications equipment, portable equipment, etc.

## 1.2 Performance Overview

Tables 1.1 and 1.2 list performance overview of the M32C/88 Group (M32C/88T).

**Table 1.1 M32C/88 Group (M32C/88T) Performance (144-Pin Package)**

Characteristic		Performance
CPU	Basic Instructions	108 instructions
	Minimum Instruction Execution Time	31.3 ns (f(BCLK)=32 MHz, V <sub>CC</sub> =4.2 V to 5.5 V)
	Operating Mode	Single-chip mode
	Address Space	16 Mbytes
	Memory Capacity	See Table 1.3
Peripheral Function	I/O Port	123 I/O pins and 1 input pin
	Multifunction Timer	Timer A: 16 bits x 5 channels, Timer B: 16 bits x 6 channels Three-phase motor control circuit
	Intelligent I/O	Time measurement function or Waveform generating function: 16 bits x 8 channels Communication function (Clock synchronous serial I/O, Clock asynchronous serial I/O, HDLC data processing)
	Serial I/O	5 Channels Clock synchronous serial I/O, Clock asynchronous serial I/O, IEBus <sup>(1)</sup> , I <sup>2</sup> C bus <sup>(2)</sup>
	CAN Module	3 channels Supporting CAN 2.0B specification
	A/D Converter	10-bit A/D converter: 1 circuit, 34 channels
	D/A Converter	8 bits x 2 channels
	DMAC	4 channels
	DMAC II	Can be activated by all peripheral function interrupt sources Immediate transfer, Calculation transfer and Chain transfer functions
	CRC Calculation Circuit	CRC-CCITT
	X/Y Converter	16 bits x 16 bits
	Watchdog Timer	15 bits x 1 channel (with prescaler)
	Interrupt	40 internal and 8 external sources, 5 software sources Interrupt priority level: 7
	Clock Generation Circuit	4 circuits Main clock oscillation circuit(*), Sub clock oscillation circuit(*), On-chip oscillator, PLL frequency synthesizer (* )Equipped with a built-in feedback resistor. Ceramic resonator or crystal oscillator must be connected externally
	Oscillation Stop Detect Function	Main clock oscillation stop detect function
	Cold Start-up/Warm Start-up Determine Function	On-chip (option)
	Electrical Characteristics	Supply Voltage
Power Consumption		28 mA (V <sub>CC</sub> =5 V, f(BCLK)=32 MHz) 10μA (V <sub>CC</sub> =5 V, f(BCLK)=32 kHz, in wait mode)
Flash Memory	Program/Erase Supply Voltage	5.0 V ± 0.5 V
	Program and Erase Endurance	100 times (all space)
Operating Ambient Temperature		-40 to 85°C (T version) -40 to 105°C (U version)
Package		144-pin plastic molded LQFP

NOTES:

1. IEBus is a trademark of NEC Electronics Corporation.
2. I<sup>2</sup>C bus is a trademark of Koninklijke Philips Electronics N. V.

All options are on a request basis.

**Table 1.2 M32C/88 Group (M32C/88T) Performance (100-Pin Package)**

Characteristic		Performance
CPU	Basic Instructions	108 instructions
	Minimum Instruction Execution Time	31.3 ns (f(BCLK)=32 MHz, V <sub>CC</sub> =4.2 V to 5.5 V)
	Operating Mode	Single-chip mode
	Address Space	16 Mbytes
	Memory Capacity	See Table 1.3
	I/O Port	87 I/O pins and 1 input pin
Peripheral Function	Multifunction Timer	Timer A: 16 bits x 5 channels, Timer B: 16 bits x 6 channels Three-phase motor control circuit
	Intelligent I/O	Time measurement function or Waveform generating function: 16 bits x 8 channels Communication function (Clock synchronous serial I/O, Clock asynchronous serial I/O, HDLC data processing)
	Serial I/O	5 Channels Clock synchronous serial I/O, Clock asynchronous serial I/O, IEBus <sup>(1)</sup> , I <sup>2</sup> C bus <sup>(2)</sup>
	CAN Module	3 channels Supporting CAN 2.0B specification
	A/D Converter	10-bit A/D converter: 1 circuit, 34 channels
	D/A Converter	8 bits x 2 channels
	DMAC	4 channels
	DMAC II	Can be activated by all peripheral function interrupt sources Immediate transfer, Calculation transfer and Chain transfer functions
	CRC Calculation Circuit	CRC-CCITT
	X/Y Converter	16 bits x 16 bits
	Watchdog Timer	15 bits x 1 channel (with prescaler)
	Interrupt	40 internal and 8 external sources, 5 software sources Interrupt priority level: 7
	Clock Generation Circuit	4 circuits Main clock oscillation circuit(*), Sub clock oscillation circuit(*), On-chip oscillator, PLL frequency synthesizer (* )Equipped with a built-in feedback resistor. Ceramic resonator or crystal oscillator must be connected externally
	Oscillation Stop Detect Function	Main clock oscillation stop detect function
	Cold Start-up/Warm Start-up Determine Function	On-chip (option)
	Electrical Characteristics	Supply Voltage
Power Consumption		28 mA (V <sub>CC</sub> =5 V, f(BCLK)=32 MHz) 10μA (V <sub>CC</sub> =5 V, f(BCLK)=32 kHz, in wait mode)
Flash Memory	Program/Erase Supply Voltage	5.0 V ± 0.5 V
	Program and Erase Endurance	100 times (all space)
Operating Ambient Temperature		-40 to 85°C (T version) -40 to 105°C (U version)
Package		100-pin plastic molded LQFP

## NOTES:

1. IEBus is a trademark of NEC Electronics Corporation.
2. I<sup>2</sup>C bus is a trademark of Koninklijke Philips Electronics N. V.

All options are on a request basis.

### 1.3 Block Diagram

Figure 1.1 shows a block diagram of the M32C/88 Group (M32C/88T) microcomputer.

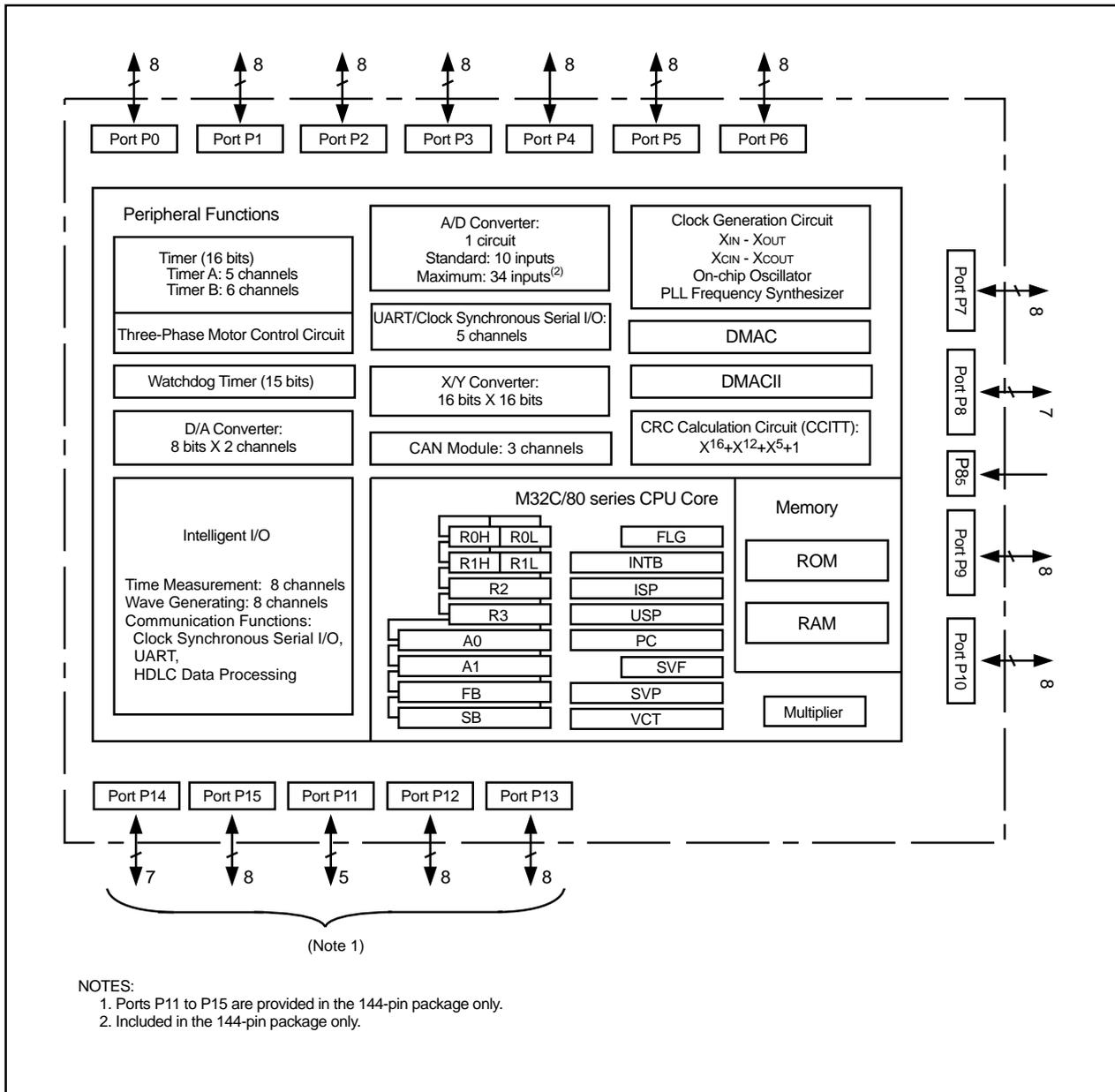


Figure 1.1 M32C/88 Group (M32C/88T) Block Diagram

## 1.4 Product Information

Table 1.3 lists the product information. Figure 1.2 shows the product numbering system.

**Table 1.3 M32C/88 Group (1) (T version, M32C/88T) As of October, 2005**

Type Number	Package Type	ROM Capacity	RAM Capacity	Remarks
M30882FJTGP (D)	PLQP0144KA-A (144P6Q-A)	512K+4K	18K	Flash Memory T version (High-reliability 85° C)
M30880FJTGP (D)	PLQP0100KB-A (100P6Q-A)			
M30882FHTGP (D)	PLQP0144KA-A (144P6Q-A)	384K+4K		
M30880FHTGP (D)	PLQP0100KB-A (100P6Q-A)			
M30882FWTGP (D)	PLQP0144KA-A (144P6Q-A)	320K+4K		
M30880FWTGP (D)	PLQP0100KB-A (100P6Q-A)			

(D): Under development

**Table 1.3 M32C/88 Group (2) (U version, M32C/88T) As of October, 2005**

Type Number	Package Type	ROM Capacity	RAM Capacity	Remarks
M30882FJUGP (D)	PLQP0144KA-A (144P6Q-A)	512K+4K	18K	Flash Memory U version (High-reliability 105° C)
M30880FJUGP (D)	PLQP0100KB-A (100P6Q-A)			
M30882FHUGP (D)	PLQP0144KA-A (144P6Q-A)	384K+4K		
M30880FHUGP (D)	PLQP0100KB-A (100P6Q-A)			
M30882FWUGP (D)	PLQP0144KA-A (144P6Q-A)	320K+4K		
M30880FWUGP (D)	PLQP0100KB-A (100P6Q-A)			

(D): Under development

### NOTE:

Contact our sales office if you are interested in the V version.

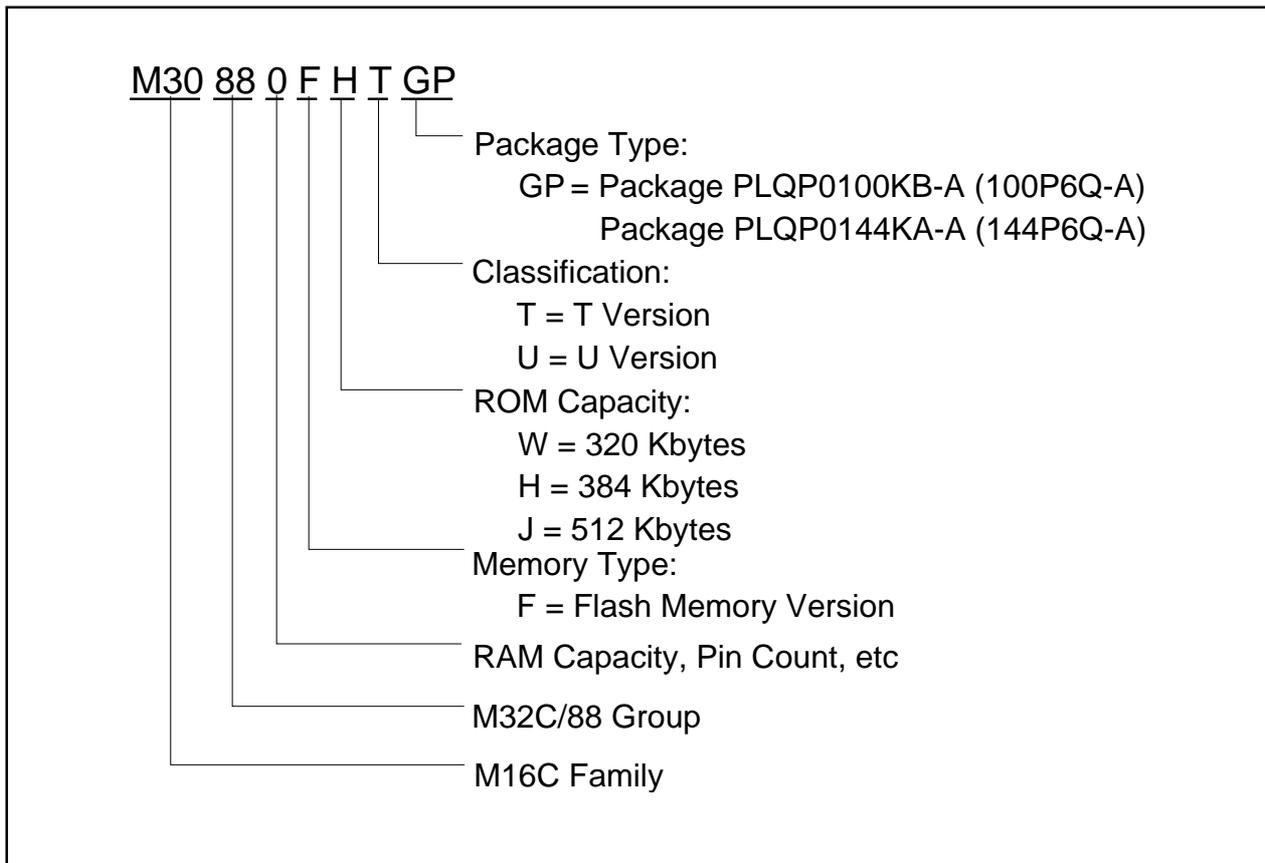
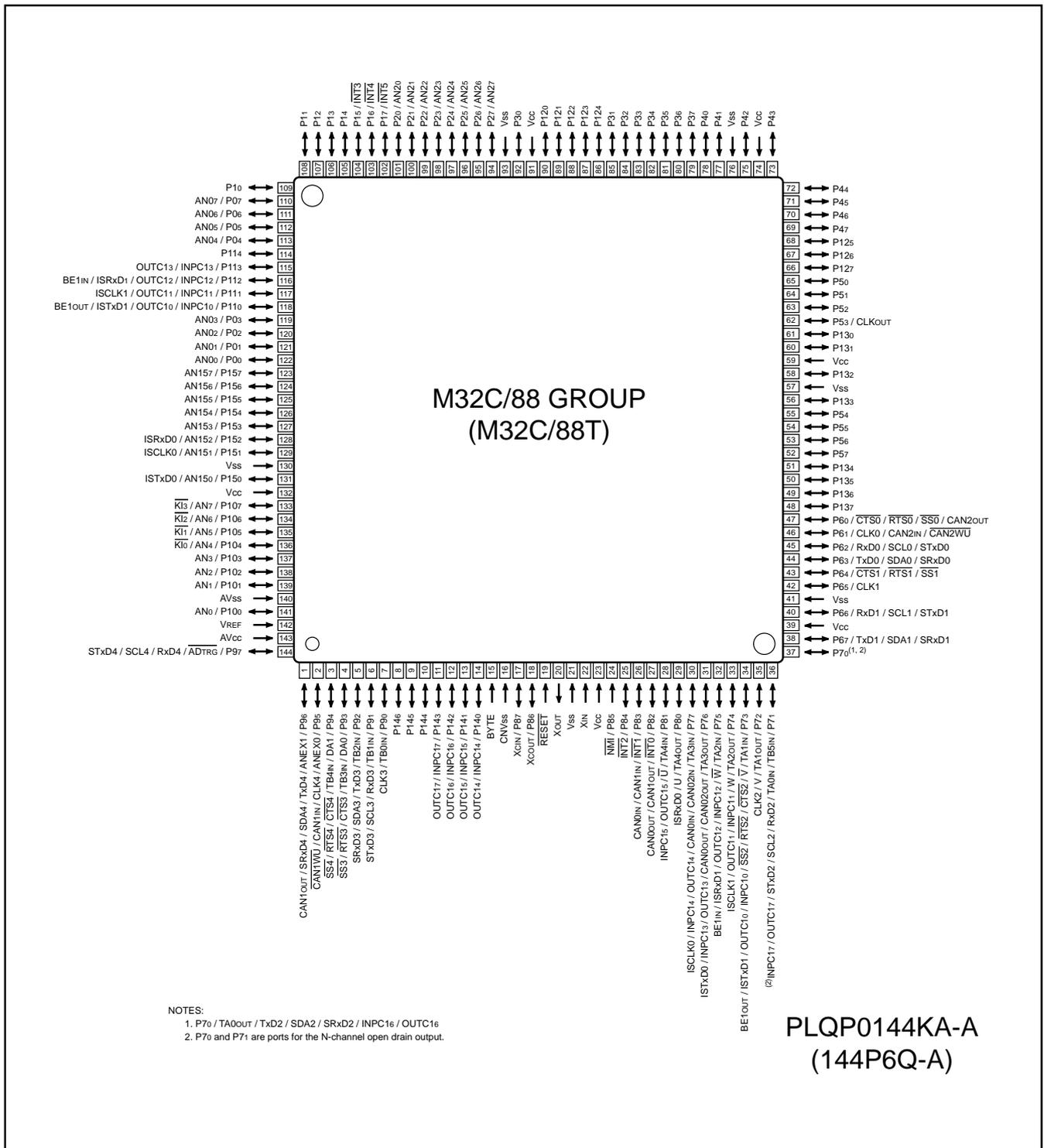


Figure 1.2 Product Numbering System

### 1.5 Pin Assignment

Figures 1.3 and 1.4 show pin assignments (top view).



NOTES:  
 1. P70 / TA0out / TxD2 / SDA2 / SRxD2 / INPC16 / OUTC16  
 2. P70 and P71 are ports for the N-channel open drain output.

Figure 1.3 Pin Assignment for 144-Pin Package

Table 1.4 Pin Characteristics for 144-Pin Package

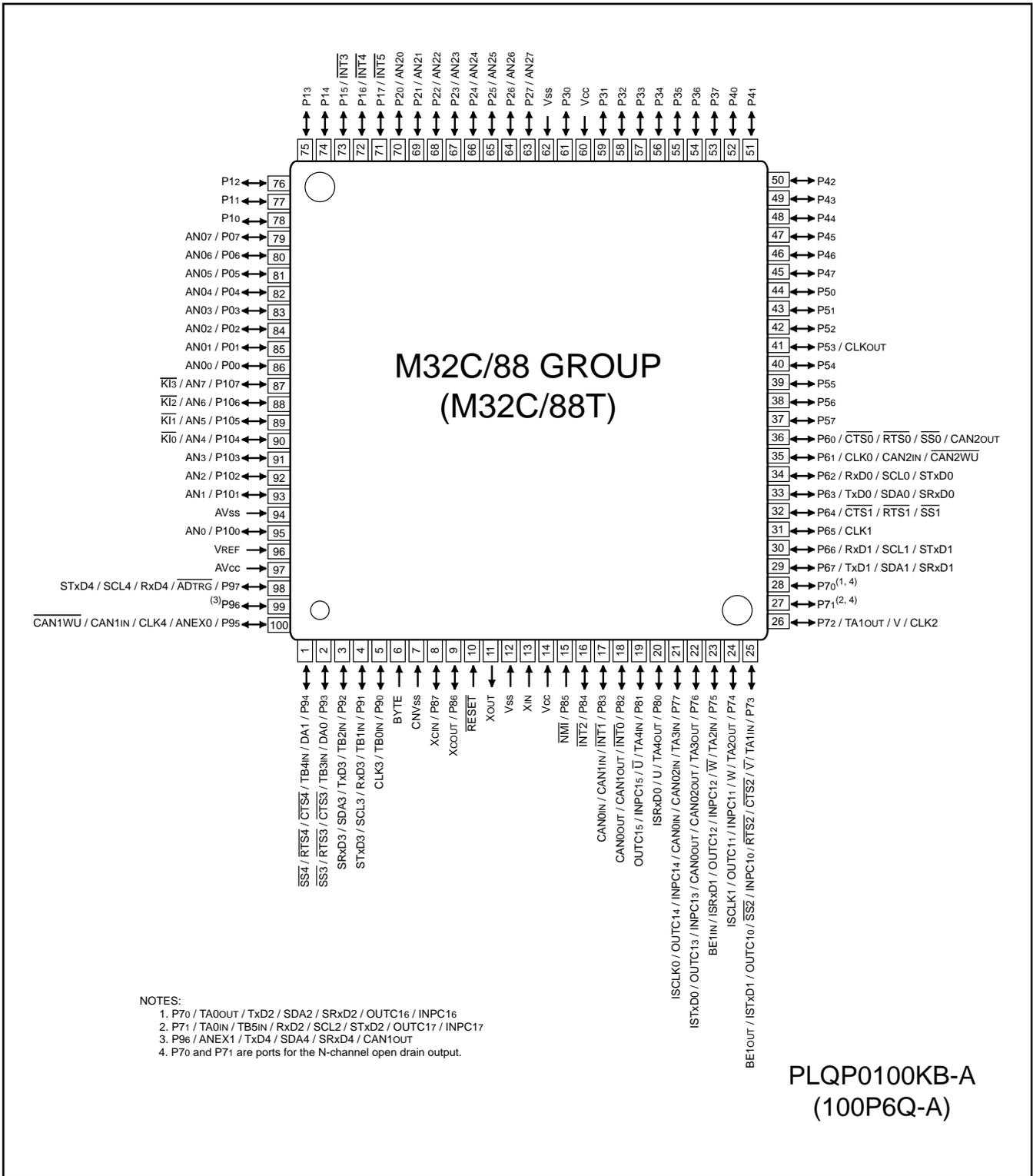
Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN Pin	Intelligent I/O Pin	Analog Pin
1		P96			TxD4/SDA4/SRxD4/CAN1out		ANEX1
2		P95			CLK4/CAN1in/CAN1WU		ANEX0
3		P94		TB4IN	CTS4/RTS4/SS4		DA1
4		P93		TB3IN	CTS3/RTS3/SS3		DA0
5		P92		TB2IN	TxD3/SDA3/SRxD3		
6		P91		TB1IN	RxD3/SCL3/STxD3		
7		P90		TB0IN	CLK3		
8		P146					
9		P145					
10		P144					
11		P143				INPC17/OUTC17	
12		P142				INPC16/OUTC16	
13		P141				INPC15/OUTC15	
14		P140				INPC14/OUTC14	
15	BYTE						
16	CNVss						
17	XcIN	P87					
18	XcOUT	P86					
19	RESET						
20	XOUT						
21	Vss						
22	Xin						
23	Vcc						
24		P85	NMI				
25		P84	INT2				
26		P83	INT1		CAN0in/CAN1in		
27		P82	INT0		CAN0out/CAN1out		
28		P81		TA4in/U		INPC15/OUTC15	
29		P80		TA4out/U		ISRxD0	
30		P77		TA3in	CAN0in/CAN02in	INPC14/OUTC14/ISCLK0	
31		P76		TA3out	CAN0out/CAN02out	INPC13/OUTC13/ISTxD0	
32		P75		TA2in/W		INPC12/OUTC12/ISRxD1/BE1in	
33		P74		TA2out/W		INPC11/OUTC11/ISCLK1	
34		P73		TA1in/V	CTS2/RTS2/SS2	INPC10/OUTC10/ISTxD1/BE1out	
35		P72		TA1out/V	CLK2		
36		P71		TB5in/TA0in	RxD2/SCL2/STxD2	INPC17/OUTC17	
37		P70		TA0out	TxD2/SDA2/SRxD2	INPC16/OUTC16	
38	Vcc	P67			TxD1/SDA1/SRxD1		
39	Vss						
40		P66			RxD1/SCL1/STxD1		
41							
42		P65			CLK1		
43		P64			CTS1/RTS1/SS1		
44		P63			TxD0/SDA0/SRxD0		
45		P62			RxD0/SCL0/STxD0		
46		P61			CLK0/CAN2in/CAN2WU		
47		P60			CTS0/RTS0/SS0/CAN2out		
48		P137					

**Table 1.4 Pin Characteristics for 144-Pin Package (Continued)**

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN Pin	Intelligent I/O Pin	Analog Pin
49		P136					
50		P135					
51		P134					
52		P57					
53		P56					
54		P55					
55		P54					
56		P133					
57	Vss						
58		P132					
59	Vcc						
60		P131					
61		P130					
62		P53					
63		P52					
64		P51					
65		P50					
66		P127					
67		P126					
68		P125					
69		P47					
70		P46					
71		P45					
72		P44					
73	Vcc	P43					
74							
75	Vss	P42					
76							
77		P41					
78		P40					
79		P37					
80		P36					
81		P35					
82		P34					
83		P33					
84		P32					
85		P31					
86		P124					
87		P123					
88		P122					
89		P121					
90	Vcc	P120					
91	Vss						
92		P30					
93							
94		P27					AN27
95		P26					AN26
96		P25					AN25

Table 1.4 Pin Characteristics for 144-Pin Package (Continued)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN Pin	Intelligent I/O Pin	Analog Pin
97		P24					AN24
98		P23					AN23
99		P22					AN22
100		P21					AN21
101		P20					AN20
102		P17	$\overline{\text{INT5}}$				
103		P16	$\overline{\text{INT4}}$				
104		P15	$\overline{\text{INT3}}$				
105		P14					
106		P13					
107		P12					
108		P11					
109		P10					
110		P07					AN07
111		P06					AN06
112		P05					AN05
113		P04					AN04
114		P114					
115		P113				INPC13/OUTC13	
116		P112				INPC12/OUTC12/ISRxD1/BE1IN	
117		P111				INPC11/OUTC11/ISCLK1	
118		P110				INPC10/OUTC10/ISTxD1/BE1OUT	
119		P03					AN03
120		P02					AN02
121		P01					AN01
122		P00					AN00
123		P157					AN157
124		P156					AN156
125		P155					AN155
126		P154					AN154
127		P153					AN153
128		P152				ISRxD0	AN152
129		P151				ISCLK0	AN151
130	Vss						
131		P150				ISTxD0	AN150
132	Vcc						
133		P107	$\overline{\text{KI3}}$				AN7
134		P106	$\overline{\text{KI2}}$				AN6
135		P105	$\overline{\text{KI1}}$				AN5
136		P104	$\overline{\text{KI0}}$				AN4
137		P103					AN3
138		P102					AN2
139		P101					AN1
140	AVss						
141		P100					AN0
142	VREF						
143	AVcc						
144		P97			RxD4/SCL4/STxD4		$\overline{\text{ADTRG}}$



- NOTES:
1. P70 / TA0OUT / TxD2 / SDA2 / SRxD2 / OUTC16 / INPC16
  2. P71 / TA0IN / TB5IN / RxD2 / SCL2 / STxD2 / OUTC17 / INPC17
  3. P96 / ANEX1 / TxD4 / SDA4 / SRxD4 / CAN1OUT
  4. P70 and P71 are ports for the N-channel open drain output.

Figure 1.4 Pin Assignment for 100-Pin Package

Table 1.5 Pin Characteristics for 100-Pin Package

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN Pin	Intelligent I/O Pin	Analog Pin
1		P94		TB4IN	$\overline{\text{CTS4}}/\overline{\text{RTS4}}/\overline{\text{SS4}}$		DA1
2		P93		TB3IN	$\overline{\text{CTS3}}/\overline{\text{RTS3}}/\overline{\text{SS3}}$		DA0
3		P92		TB2IN	TxD3/SDA3/SRx3D3		
4		P91		TB1IN	RxD3/SCL3/STxD3		
5		P90		TB0IN	CLK3		
6	BYTE						
7	CNVss						
8	XCIN	P87					
9	XCOUT	P86					
10	$\overline{\text{RESET}}$						
11	XOUT						
12	Vss						
13	XIN						
14	Vcc						
15		P85	$\overline{\text{NMI}}$				
16		P84	$\overline{\text{INT2}}$				
17		P83	$\overline{\text{INT1}}$		CAN0IN/CAN1IN		
18		P82	$\overline{\text{INT0}}$		CAN0OUT/CAN1OUT		
19		P81		TA4IN $\overline{\text{U}}$		INPC15/OUTC15	
20		P80		TA4OUT/U		ISRxD0	
21		P77		TA3IN	CAN0IN/CAN02IN	INPC14/OUTC14/ISCLK0	
22		P76		TA3OUT	CAN0OUT/CAN02OUT	INPC13/OUTC13/ISTxD0	
23		P75		TA2IN $\overline{\text{W}}$		INPC12/OUTC12/ISRxD1/BE1IN	
24		P74		TA2OUT/W		INPC11/OUTC11/ISCLK1	
25		P73		TA1IN $\overline{\text{V}}$	$\overline{\text{CTS2}}/\overline{\text{RTS2}}/\overline{\text{SS2}}$	INPC10/OUTC10/ISTxD1/BE1OUT	
26		P72		TA1OUT/V	CLK2		
27		P71		TB5IN/TA0IN	RxD2/SCL2/STxD2	INPC17/OUTC17	
28		P70		TA0OUT	TxD2/SDA2/SRx2D2	INPC16/OUTC16	
29		P67			TxD1/SDA1/SRx1D1		
30		P66			RxD1/SCL1/STxD1		
31		P65			CLK1		
32		P64			$\overline{\text{CTS1}}/\overline{\text{RTS1}}/\overline{\text{SS1}}$		
33		P63			TxD0/SDA0/SRx0D0		
34		P62			RxD0/SCL0/STxD0		
35		P61			CLK0/CAN2IN/CAN2WU		
36		P60			$\overline{\text{CTS0}}/\overline{\text{RTS0}}/\overline{\text{SS0}}/\text{CAN2OUT}$		
37		P57					
38		P56					
39		P55					
40		P54					
41		P53					
42		P52					
43		P51					
44		P50					
45		P47					
46		P46					
47		P45					
48		P44					
49		P43					
50		P42					

Table 1.5 Pin Characteristics for 100-Pin Package (Continued)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN Pin	Intelligent I/O Pin	Analog Pin
51		P41					
52		P40					
53		P37					
54		P36					
55		P35					
56		P34					
57		P33					
58		P32					
59		P31					
60	Vcc						
61		P30					
62	Vss						
63		P27					AN27
64		P26					AN26
65		P25					AN25
66		P24					AN24
67		P23					AN23
68		P22					AN22
69		P21					AN21
70		P20					AN20
71		P17	$\overline{\text{INT5}}$				
72		P16	$\overline{\text{INT4}}$				
73		P15	$\overline{\text{INT3}}$				
74		P14					
75		P13					
76		P12					
77		P11					
78		P10					
79		P07					AN07
80		P06					AN06
81		P05					AN05
82		P04					AN04
83		P03					AN03
84		P02					AN02
85		P01					AN01
86		P00					AN00
87		P107	$\overline{\text{K13}}$				AN7
88		P106	$\overline{\text{K12}}$				AN6
89		P105	$\overline{\text{K11}}$				AN5
90		P104	$\overline{\text{K10}}$				AN4
91		P103					AN3
92		P102					AN2
93		P101					AN1
94	AVss						
95		P100					AN0
96	VREF						
97	AVcc						
98		P97			RxD4/SCL4/STxD4		$\overline{\text{ADTRG}}$
99		P96			TxD4/SDA4/SRxD4/CAN1out		ANEX1
100		P95			CLK4/CAN1IN/CAN1WU		ANEX0

## 1.6 Pin Description

**Table 1.6 Pin Description (100-Pin and 144-Pin Packages)**

Classification	Symbol	I/O Type	Function
Power Supply	Vcc Vss	I	Apply 4.2 to 5.5 V to both Vcc pins. Apply 0 V to the Vss pin
Analog Power Supply	AVCC AVSS	I	Supplies power to the A/D converter. Connect the AVCC pin to Vcc and the AVSS pin to Vss
Reset Input	RESET	I	The microcomputer is in a reset state when "L" is applied to the RESET pin
CNVss	CNVss	I	Switches processor mode. Connect the CNVss pin to Vss
Input to Switch External Data Bus Width	BYTE	I	Connect the BYTE pin to Vss
Main Clock Input	XIN	I	I/O pins for the main clock oscillation circuit. Connect a ceramic resonator or crystal oscillator between XIN and XOUT. To apply external clock, apply it to XIN and leave XOUT open.
Main Clock Output	XOUT	O	
Sub Clock Input	XCIN	I	I/O pins for the sub clock oscillation circuit. Connect a crystal oscillator between XCIN and Xcout. To apply external clock, apply it to XCIN and leave Xcout open
Sub Clock output	Xcout	O	
Clock Output	CLKOUT	O	Outputs the clock having the same frequency as fc, f8 or f32
INT Interrupt Input	INT0 to INT5	I	Input pins for the INT interrupt
NMI Interrupt Input	NMI	I	Input pin for the NMI interrupt
Key Input Interrupt	KI0 to KI3	I	Input pins for the key input interrupt
Timer A	TA0OUT to TA4OUT	I/O	I/O pins for the timer A0 to A4 (TA0OUT is a pin for the N-channel open drain output.)
	TA0IN to TA4IN	I	Input pins for Timer A0 to A4
Timer B	TB0IN to TB5IN	I	Input pins for Timer B0 to B5
Three-phase Motor Control Timer Output	U, $\bar{U}$ , V, $\bar{V}$ , W, $\bar{W}$	O	Output pins for the three-phase motor control timer
Serial I/O	CTS0 to CTS4	I	Input pins for data transmission control
	RTS0 to RTS4	O	Output pins for data reception control
	CLK0 to CLK4	I/O	Inputs and outputs the transfer clock
	RxD0 to RxD4	I	Inputs serial data
	TxD0 to TxD4	O	Outputs serial data (TxD2 is a pin for the N-channel open drain output.)
I <sup>2</sup> C Mode	SDA0 to SDA4	I/O	Inputs and outputs serial data (SDA2 is a pin for the N-channel open drain output.)
	SCL0 to SCL4		Inputs and outputs the transfer clock (SCL2 is a pin for the N-channel open drain output.)
Serial I/O Special Function	STxD0 to STxD4	O	Outputs serial data when slave mode is selected (STxD2 is a pin for the N-channel open drain output.)
	SRxD0 to SRxD4	I	Inputs serial data when slave mode is selected
	SS0 to SS4	I	Input pins to control serial I/O special function

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

**Table 1.6 Pin Description (100-Pin and 144-Pin Packages) (Continued)**

Classification	Symbol	I/O Type	Function	
Reference Voltage Input	VREF	I	Applies reference voltage to the A/D converter and D/A converter	
A/D Converter	AN0 to AN7 AN00 to AN07 AN20 to AN27	I	Analog input pins for the A/D converter	
	ADTRG	I	Input pin for an external A/D trigger	
	ANEX0	I/O	Extended analog input pin for the A/D converter and output pin in external op-amp connection mode	
	ANEX1	I	Extended analog input pin for the A/D converter	
D/A Converter	DA0, DA1	O	Output pin for the D/A converter	
Intelligent I/O	INPC10 to INPC17	I	Input pins for the time measurement function	
	OUTC10 to OUTC17	O	Output pins for the waveform generating function (OUTC16 and OUTC17 assigned to P70 and P71 are pins for the N-channel open drain output.)	
	ISCLK0 ISCLK1	I/O	Inputs and outputs the clock for the intelligent I/O communication function	
	ISRXD0 ISRXD1	I	Inputs data for the intelligent I/O communication function	
	ISTXD0 ISTXD1	O	Outputs data for the intelligent I/O communication function	
	BE1IN	I	Inputs data for the intelligent I/O communication function	
	BE1OUT	O	Outputs data for the intelligent I/O communication function	
	CAN	CAN0IN CAN02IN CAN1IN CAN2IN	I	Input pin for the CAN communication function
		CAN0OUT CAN02OUT CAN1OUT CAN2OUT	O	Output pin for the CAN communication function
CAN1WU CAN2WU		I	Input pin for the CANi wake-up interrupt (i=1, 2)	
I/O Ports		P00 to P07 P10 to P17 P20 to P27 P30 to P37 P40 to P47 P50 to P57	I/O	8-bit I/O ports for CMOS. Each port can be programmed for input or output under the control of the direction register. An input port can be set, by program, for a pull-up resistor available or for no pull-up resistor available in 4-bit units
		P60 to P67 P70 to P77 P90 to P97 P100 to P107	I/O	I/O ports having equivalent functions to P0 (P70 and P71 are ports for the N-channel open drain output.)
		P80 to P84 P86, P87	I/O	I/O ports having equivalent functions to P0
Input Port		P85	I	Shares a pin with NMI. NMI input state can be got by reading P85

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

**Table 1.6 Pin Description (144-Pin Package Only) (Continued)**

Classification	Symbol	I/O Type	Function
A/D Converter	AN150 to AN157	I	Analog input pins for the A/D converter
I/O Ports	P110 to P114 P120 to P127 P130 to P137 P140 to P146 P150 to P157	I/O	I/O ports having equivalent functions to P0

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

## 2. Central Processing Unit (CPU)

Figure 2.1 shows the CPU registers.

The register bank is comprised of 8 registers (R0, R1, R2, R3, A0, A1, SB and FB) out of 28 CPU registers. Two sets of register banks are provided.

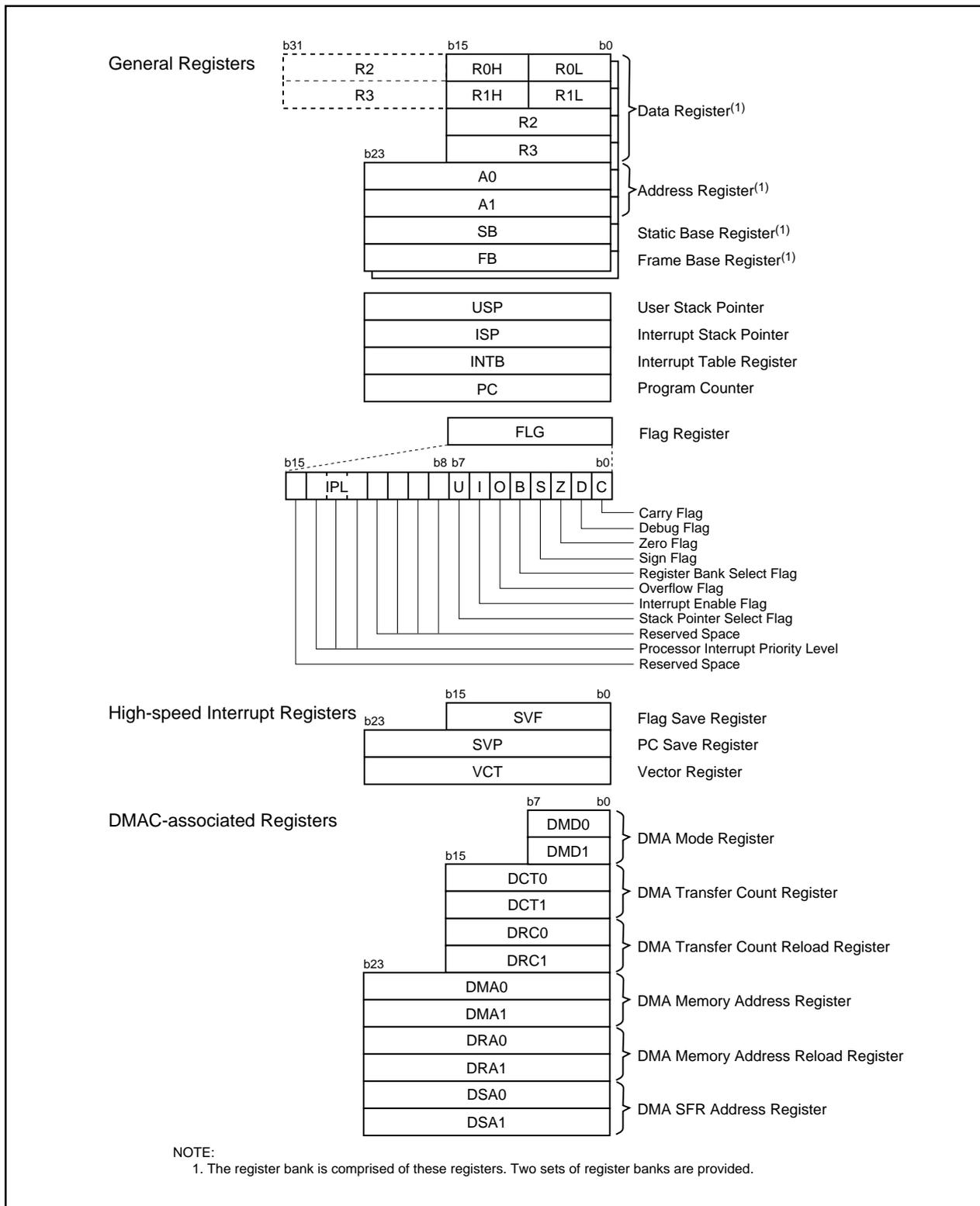


Figure 2.1 CPU Register

## 2.1 General Registers

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

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

### 2.1.2 Address Registers (A0 and A1)

A0 and A1 are 24-bit registers for A0-/A1-indirect addressing, A0-/A1-relative addressing, transfer, arithmetic and logic operations.

### 2.1.3 Static Base Register (SB)

SB is a 24-bit register for SB-relative addressing.

### 2.1.4 Frame Base Register (FB)

FB is a 24-bit register for FB-relative addressing.

### 2.1.5 Program Counter (PC)

PC, 24 bits wide, indicates the address of an instruction to be executed.

### 2.1.6 Interrupt Table Register (INTB)

INTB is a 24-bit register indicating the starting address of an relocatable interrupt vector table.

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

The stack pointers (SP), USP and ISP, are 24 bits wide each. The U flag is used to switch between USP and ISP. Refer to **2.1.8 Flag Register (FLG)** for details on the U flag. Set USP and ISP to even addresses to execute an interrupt sequence efficiently.

### 2.1.8 Flag Register (FLG)

FLG is a 16-bit register indicating a CPU state.

#### 2.1.8.1 Carry Flag (C)

The C flag indicates whether carry or borrow has occurred after executing an instruction.

#### 2.1.8.2 Debug Flag (D)

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

#### 2.1.8.3 Zero Flag (Z)

The Z flag is set to "1" when the value of zero is obtained from an arithmetic operation; otherwise "0".

#### 2.1.8.4 Sign Flag (S)

The S flag is set to "1" when a negative value is obtained from an arithmetic operation; otherwise "0".

### 2.1.8.5 Register Bank Select Flag (B)

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

### 2.1.8.6 Overflow Flag (O)

The O flag is set to "1" when the result of an arithmetic operation overflows; otherwise "0".

### 2.1.8.7 Interrupt Enable Flag (I)

The I flag enables a maskable interrupt.

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

### 2.1.8.8 Stack Pointer Select Flag (U)

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

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

### 2.1.8.9 Processor Interrupt Priority Level (IPL)

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

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

### 2.1.8.10 Reserved Space

When writing to a reserved space, set to "0". When reading, its content is indeterminate.

## 2.2 High-Speed Interrupt Registers

Registers associated with the high-speed interrupt are as follows:

- Flag save register (SVF)
- PC save register (SVP)
- Vector register (VCT)

Refer to **10.4 High-Speed Interrupt** for details.

## 2.3 DMAC-Associated Registers

Registers associated with DMAC are as follows:

- DMA mode register (DMD0, DMD1)
- DMA transfer count register (DCT0, DCT1)
- DMA transfer count reload register (DRC0, DRC1)
- DMA memory address register (DMA0, DMA1)
- DMA SFR address register (DSA0, DSA1)
- DMA memory address reload register (DRA0, DRA1)

Refer to **12. DMAC** for details.

### 3. Memory

Figure 3.1 shows a memory map of the M32C/88 Group (M32C/88T).

The M32C/88 Group (M32C/88T) provides 16-Mbyte address space addressed from 000000<sub>16</sub> to FFFFFFF<sub>16</sub>.

The internal ROM is allocated from address FFFFFFF<sub>16</sub> to lower. For example, a 64-Kbyte internal ROM is addressed from FF0000<sub>16</sub> to FFFFFFF<sub>16</sub>.

The fixed interrupt vectors are allocated from address FFFFDC<sub>16</sub> to FFFFFFF<sub>16</sub>. It stores the starting address of each interrupt routine.

The internal RAM is allocated from address 000400<sub>16</sub> to higher. For example, a 10-Kbyte internal RAM is allocated from address 000400<sub>16</sub> to 002BFF<sub>16</sub>. Besides storing data, it becomes stacks when the subroutine is called or an interrupt is acknowledged.

SFRs, consisting of control registers for peripheral functions such as I/O port, A/D converter, serial I/O, timers, is allocated from address 000000<sub>16</sub> to 0003FF<sub>16</sub>. All blank spaces within SFRs are reserved and cannot be accessed by users.

The special page vectors are addressed from FFFE00<sub>16</sub> to FFFFDB<sub>16</sub>. It is used for the JMPS instruction and JSRS instruction. Refer to the Renesas publication **M32C/80 Series Software Manual** for details.

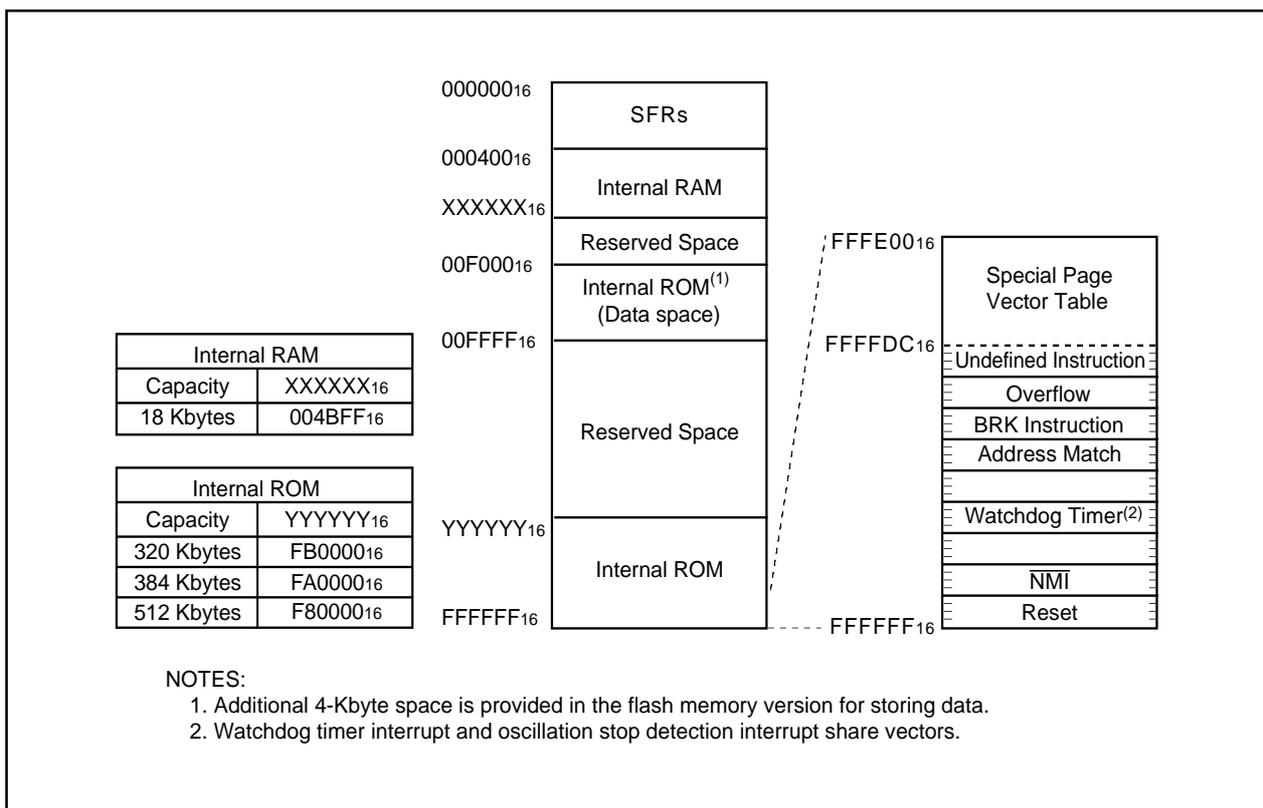


Figure 3.1 Memory Map

## 4. Special Function Registers (SFRs)

Address	Register	Symbol	Value after RESET
0000 <sub>16</sub>			
0001 <sub>16</sub>			
0002 <sub>16</sub>			
0003 <sub>16</sub>			
0004 <sub>16</sub>	Processor Mode Register <sup>(1)</sup>	PM0	1000 0000 <sub>2</sub> (CNVss pin ="L")
0005 <sub>16</sub>	Processor Mode Register 1	PM1	00 <sub>16</sub>
0006 <sub>16</sub>	System Clock Control Register 0	CM0	0000 1000 <sub>2</sub>
0007 <sub>16</sub>	System Clock Control Register 1	CM1	0010 0000 <sub>2</sub>
0008 <sub>16</sub>			
0009 <sub>16</sub>	Address Match Interrupt Enable Register	AIER	00 <sub>16</sub>
000A <sub>16</sub>	Protect Register	PRCR	XXXX 0000 <sub>2</sub>
000B <sub>16</sub>			
000C <sub>16</sub>	Main Clock Division Register	MCD	XXX0 1000 <sub>2</sub>
000D <sub>16</sub>	Oscillation Stop Detection Register	CM2	00 <sub>16</sub>
000E <sub>16</sub>	Watchdog Timer Start Register	WDTS	XX <sub>16</sub>
000F <sub>16</sub>	Watchdog Timer Control Register	WDC	000X XXXX <sub>2</sub>
0010 <sub>16</sub>	Address Match Interrupt Register 0	RMAD0	000000 <sub>16</sub>
0011 <sub>16</sub>			
0012 <sub>16</sub>			
0013 <sub>16</sub>	Processor Mode Register 2	PM2	00 <sub>16</sub>
0014 <sub>16</sub>	Address Match Interrupt Register 1	RMAD1	000000 <sub>16</sub>
0015 <sub>16</sub>			
0016 <sub>16</sub>			
0017 <sub>16</sub>			
0018 <sub>16</sub>	Address Match Interrupt Register 2	RMAD2	000000 <sub>16</sub>
0019 <sub>16</sub>			
001A <sub>16</sub>			
001B <sub>16</sub>			
001C <sub>16</sub>	Address Match Interrupt Register 3	RMAD3	000000 <sub>16</sub>
001D <sub>16</sub>			
001E <sub>16</sub>			
001F <sub>16</sub>			
0020 <sub>16</sub>			
0021 <sub>16</sub>			
0022 <sub>16</sub>			
0023 <sub>16</sub>			
0024 <sub>16</sub>			
0025 <sub>16</sub>			
0026 <sub>16</sub>	PLL Control Register 0	PLC0	0001 X010 <sub>2</sub>
0027 <sub>16</sub>	PLL Control Register 1	PLC1	000X 0000 <sub>2</sub>
0028 <sub>16</sub>	Address Match Interrupt Register 4	RMAD4	000000 <sub>16</sub>
0029 <sub>16</sub>			
002A <sub>16</sub>			
002B <sub>16</sub>			
002C <sub>16</sub>	Address Match Interrupt Register 5	RMAD5	000000 <sub>16</sub>
002D <sub>16</sub>			
002E <sub>16</sub>			
002F <sub>16</sub>			

X: Indeterminate

Blank spaces are reserved. No access is allowed.

NOTE:

- The PM01 and PM00 bits in the PM0 register maintain values set before reset, even after software reset or watch-dog timer reset has been performed.

Address	Register	Symbol	Value after RESET
0060 <sub>16</sub>			
0030 <sub>16</sub>			
0031 <sub>16</sub>			
0032 <sub>16</sub>			
0033 <sub>16</sub>			
0034 <sub>16</sub>			
0035 <sub>16</sub>			
0036 <sub>16</sub>			
0037 <sub>16</sub>			
0038 <sub>16</sub> 0039 <sub>16</sub> 003A <sub>16</sub>	Address Match Interrupt Register 6	RMAD6	000000 <sub>16</sub>
003B <sub>16</sub>			
003C <sub>16</sub> 003D <sub>16</sub> 003E <sub>16</sub>	Address Match Interrupt Register 7	RMAD7	000000 <sub>16</sub>
003F <sub>16</sub>			
0040 <sub>16</sub>			
0041 <sub>16</sub>			
0042 <sub>16</sub>			
0043 <sub>16</sub>			
0044 <sub>16</sub>			
0045 <sub>16</sub>			
0046 <sub>16</sub>			
0047 <sub>16</sub>			
0048 <sub>16</sub>			
0049 <sub>16</sub>			
004A <sub>16</sub>			
004B <sub>16</sub>			
004C <sub>16</sub>			
004D <sub>16</sub>			
004E <sub>16</sub>			
004F <sub>16</sub>			
0050 <sub>16</sub>			
0051 <sub>16</sub>			
0052 <sub>16</sub>			
0053 <sub>16</sub>			
0054 <sub>16</sub>			
0055 <sub>16</sub> 0056 <sub>16</sub>	Flash Memory Control Register 1	FMR1	0000 0101 <sub>2</sub>
0057 <sub>16</sub> 0058 <sub>16</sub>	Flash Memory Control Register 0	FMR0	0000 0001 <sub>2</sub>
0059 <sub>16</sub>			
005A <sub>16</sub>			
005B <sub>16</sub>			
005C <sub>16</sub>			
005D <sub>16</sub>			
005E <sub>16</sub>			
005F <sub>16</sub>			

X: Indeterminate

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Value after RESET
0060 <sub>16</sub>			
0061 <sub>16</sub>			
0062 <sub>16</sub>			
0063 <sub>16</sub>			
0064 <sub>16</sub>			
0065 <sub>16</sub>			
0066 <sub>16</sub>			
0067 <sub>16</sub>			
0068 <sub>16</sub>	DMA0 Interrupt Control Register	DM0IC	XXXX X000 <sub>2</sub>
0069 <sub>16</sub>	Timer B5 Interrupt Control Register	TB5IC	XXXX X000 <sub>2</sub>
006A <sub>16</sub>	DMA2 Interrupt Control Register	DM2IC	XXXX X000 <sub>2</sub>
006B <sub>16</sub>	UART2 Receive /ACK Interrupt Control Register	S2RIC	XXXX X000 <sub>2</sub>
006C <sub>16</sub>	Timer A0 Interrupt Control Register	TA0IC	XXXX X000 <sub>2</sub>
006D <sub>16</sub>	UART3 Receive /ACK Interrupt Control Register	S3RIC	XXXX X000 <sub>2</sub>
006E <sub>16</sub>	Timer A2 Interrupt Control Register	TA2IC	XXXX X000 <sub>2</sub>
006F <sub>16</sub>	UART4 Receive /ACK Interrupt Control Register	S4RIC	XXXX X000 <sub>2</sub>
0070 <sub>16</sub>	Timer A4 Interrupt Control Register	TA4IC	XXXX X000 <sub>2</sub>
0071 <sub>16</sub>	UART0/UART3 Bus Conflict Detect Interrupt Control Register	BCN0IC/BCN3IC	XXXX X000 <sub>2</sub>
0072 <sub>16</sub>	UART0 Receive/ACK Interrupt Control Register	S0RIC	XXXX X000 <sub>2</sub>
0073 <sub>16</sub>	A/D0 Conversion Interrupt Control Register	AD0IC	XXXX X000 <sub>2</sub>
0074 <sub>16</sub>	UART1 Receive/ACK Interrupt Control Register	S1RIC	XXXX X000 <sub>2</sub>
0075 <sub>16</sub>	Intelligent I/O Interrupt Control Register 0/ CAN Interrupt 3 Control Register	IIO0IC/ CAN3IC	XXXX X000 <sub>2</sub>
0076 <sub>16</sub>	Timer B1 Interrupt Control Register	TB1IC	XXXX X000 <sub>2</sub>
0077 <sub>16</sub>	Intelligent I/O Interrupt Control Register 2/ CAN Interrupt 6 Control Register	IIO2IC/ CAN6IC	XXXX X000 <sub>2</sub>
0078 <sub>16</sub>	Timer B3 Interrupt Control Register	TB3IC	XXXX X000 <sub>2</sub>
0079 <sub>16</sub>	Intelligent I/O Interrupt Control Register 4	IIO4IC	XXXX X000 <sub>2</sub>
007A <sub>16</sub>	INT5 Interrupt Control Register	INT5IC	XX00 X000 <sub>2</sub>
007B <sub>16</sub>	CAN Interrupt 8 Control Register	CAN8IC	XXXX X000 <sub>2</sub>
007C <sub>16</sub>	INT3 Interrupt Control Register	INT3IC	XX00 X000 <sub>2</sub>
007D <sub>16</sub>	Intelligent I/O Interrupt Control Register 8	IIO8IC	XXXX X000 <sub>2</sub>
007E <sub>16</sub>	INT1 Interrupt Control Register	INT1IC	XX00 X000 <sub>2</sub>
007F <sub>16</sub>	Intelligent I/O Interrupt Control Register 10/ CAN Interrupt 1 Control Register	IIO10IC/ CAN1IC	XXXX X000 <sub>2</sub>
0080 <sub>16</sub>			
0081 <sub>16</sub>	CAN Interrupt 2 Control Register	CAN2IC	XXXX X000 <sub>2</sub>
0082 <sub>16</sub>			
0083 <sub>16</sub>			
0084 <sub>16</sub>			
0085 <sub>16</sub>			
0086 <sub>16</sub>			
0087 <sub>16</sub>			
0088 <sub>16</sub>	DMA1 Interrupt Control Register	DM1IC	XXXX X000 <sub>2</sub>
0089 <sub>16</sub>	UART2 Transmit /NACK Interrupt Control Register	S2TIC	XXXX X000 <sub>2</sub>
008A <sub>16</sub>	DMA3 Interrupt Control Register	DM3IC	XXXX X000 <sub>2</sub>
008B <sub>16</sub>	UART3 Transmit /NACK Interrupt Control Register	S3TIC	XXXX X000 <sub>2</sub>
008C <sub>16</sub>	Timer A1 Interrupt Control Register	TA1IC	XXXX X000 <sub>2</sub>
008D <sub>16</sub>	UART4 Transmit /NACK Interrupt Control Register	S4TIC	XXXX X000 <sub>2</sub>
008E <sub>16</sub>	Timer A3 Interrupt Control Register	TA3IC	XXXX X000 <sub>2</sub>
008F <sub>16</sub>	UART2 Bus Conflict Detect Interrupt Control Register	BCN2IC	XXXX X000 <sub>2</sub>

X: Indeterminate

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Value after RESET
0090 <sub>16</sub>	UART0 Transmit /NACK Interrupt Control Register	S0TIC	XXXX X0002
0091 <sub>16</sub>	UART1/UART4 Bus Conflict Detect Interrupt Control Register	BCN1IC/BCN4IC	XXXX X0002
0092 <sub>16</sub>	UART1 Transmit/NACK Interrupt Control Register	S1TIC	XXXX X0002
0093 <sub>16</sub>	Key Input Interrupt Control Register	KUPIC	XXXX X0002
0094 <sub>16</sub>	Timer B0 Interrupt Control Register	TB0IC	XXXX X0002
0095 <sub>16</sub>	Intelligent I/O Interrupt Control Register 1/ CAN Interrupt 4 Control Register	IIO1IC/ CAN4IC	XXXX X0002
0096 <sub>16</sub>	Timer B2 Interrupt Control Register	TB2IC	XXXX X0002
0097 <sub>16</sub>	Intelligent I/O Interrupt Control Register 3/ CAN Interrupt 7 Control Register	IIO3IC/ CAN7IC	XXXX X0002
0098 <sub>16</sub>	Timer B4 Interrupt Control Register	TB4IC	XXXX X0002
0099 <sub>16</sub>	CAN Interrupt 5 Control Register	CAN5IC	XXXX X0002
009A <sub>16</sub>	INT4 Interrupt Control Register	INT4IC	XX00 X0002
009B <sub>16</sub>			
009C <sub>16</sub>	INT2 Interrupt Control Register	INT2IC	XX00 X0002
009D <sub>16</sub>	Intelligent I/O Interrupt Control Register 9/ CAN Interrupt 0 Control Register	IIO9IC/ CAN0IC	XXXX X0002
009E <sub>16</sub>	INT0 Interrupt Control Register	INT0IC	XX00 X0002
009F <sub>16</sub>	Exit Priority Control Register	RLVL	XXXX 00002
00A0 <sub>16</sub>	Interrupt Request Register 0	IIO0IR	0000 000X2
00A1 <sub>16</sub>	Interrupt Request Register 1	IIO1IR	0000 000X2
00A2 <sub>16</sub>	Interrupt Request Register 2	IIO2IR	0000 000X2
00A3 <sub>16</sub>	Interrupt Request Register 3	IIO3IR	0000 000X2
00A4 <sub>16</sub>	Interrupt Request Register 4	IIO4IR	0000 000X2
00A5 <sub>16</sub>	Interrupt Request Register 5	IIO5IR	0000 000X2
00A6 <sub>16</sub>	Interrupt Request Register 6	IIO6IR	0000 000X2
00A7 <sub>16</sub>			
00A8 <sub>16</sub>	Interrupt Request Register 8	IIO8IR	0000 000X2
00A9 <sub>16</sub>	Interrupt Request Register 9	IIO9IR	0000 000X2
00AA <sub>16</sub>	Interrupt Request Register 10	IIO10IR	0000 000X2
00AB <sub>16</sub>	Interrupt Request Register 11	IIO11IR	0000 000X2
00AC <sub>16</sub>			
00AD <sub>16</sub>			
00AE <sub>16</sub>			
00AF <sub>16</sub>			
00B0 <sub>16</sub>	Interrupt Enable Register 0	IIO0IE	00 <sub>16</sub>
00B1 <sub>16</sub>	Interrupt Enable Register 1	IIO1IE	00 <sub>16</sub>
00B2 <sub>16</sub>	Interrupt Enable Register 2	IIO2IE	00 <sub>16</sub>
00B3 <sub>16</sub>	Interrupt Enable Register 3	IIO3IE	00 <sub>16</sub>
00B4 <sub>16</sub>	Interrupt Enable Register 4	IIO4IE	00 <sub>16</sub>
00B5 <sub>16</sub>	Interrupt Enable Register 5	IIO5IE	00 <sub>16</sub>
00B6 <sub>16</sub>	Interrupt Enable Register 6	IIO6IE	00 <sub>16</sub>
00B7 <sub>16</sub>			
00B8 <sub>16</sub>	Interrupt Enable Register 8	IIO8IE	00 <sub>16</sub>
00B9 <sub>16</sub>	Interrupt Enable Register 9	IIO9IE	00 <sub>16</sub>
00BA <sub>16</sub>	Interrupt Enable Register 10	IIO10IE	00 <sub>16</sub>
00BB <sub>16</sub>	Interrupt Enable Register 11	IIO11IE	00 <sub>16</sub>
00BC <sub>16</sub>			
00BD <sub>16</sub>			
00BE <sub>16</sub>			
00BF <sub>16</sub>			

X: Indeterminate

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Value after RESET
00C0 <sup>16</sup>			
00C1 <sup>16</sup>			
00C2 <sup>16</sup>			
00C3 <sup>16</sup>			
00C4 <sup>16</sup>			
00C5 <sup>16</sup>			
00C6 <sup>16</sup>			
00C7 <sup>16</sup>			
00C8 <sup>16</sup>			
00C9 <sup>16</sup>			
00CA <sup>16</sup>			
00CB <sup>16</sup>			
00CC <sup>16</sup>			
00CD <sup>16</sup>			
00CE <sup>16</sup>			
00CF <sup>16</sup>			
00D0 <sup>16</sup>			
00D1 <sup>16</sup>			
00D2 <sup>16</sup>			
00D3 <sup>16</sup>			
00D4 <sup>16</sup>			
00D5 <sup>16</sup>			
00D6 <sup>16</sup>			
00D7 <sup>16</sup>			
00D8 <sup>16</sup>			
00D9 <sup>16</sup>			
00DA <sup>16</sup>			
00DB <sup>16</sup>			
00DC <sup>16</sup>			
00DD <sup>16</sup>			
00DE <sup>16</sup>			
00DF <sup>16</sup>			
00E0 <sup>16</sup>			
00E1 <sup>16</sup>			
00E2 <sup>16</sup>			
00E3 <sup>16</sup>			
00E4 <sup>16</sup>			
00E5 <sup>16</sup>			
00E6 <sup>16</sup>			
00E7 <sup>16</sup>			
00E8 <sup>16</sup> 00E9 <sup>16</sup>	SI/O Receive Buffer Register 0	G0RB	XXXX XXXX <sub>2</sub> XXX0 XXXX <sub>2</sub>
00EA <sup>16</sup> 00EB <sup>16</sup>	Transmit Buffer/Receive Data Register 0	G0TB/G0DR	XX <sub>16</sub>
00EC <sup>16</sup>	Receive Input Register 0	G0RI	XX <sub>16</sub>
00ED <sup>16</sup>	SI/O Communication Mode Register 0	G0MR	00 <sub>16</sub>
00EE <sup>16</sup>	Transmit Output Register 0	G0TO	XX <sub>16</sub>
00EF <sup>16</sup>	SI/O Communication Control Register 0	G0CR	0000 X011 <sub>2</sub>

X: Indeterminate

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Value after RESET
00F0 <sub>16</sub>	Data Compare Register 00	G0CMP0	XX <sub>16</sub>
00F1 <sub>16</sub>	Data Compare Register 01	G0CMP1	XX <sub>16</sub>
00F2 <sub>16</sub>	Data Compare Register 02	G0CMP2	XX <sub>16</sub>
00F3 <sub>16</sub>	Data Compare Register 03	G0CMP3	XX <sub>16</sub>
00F4 <sub>16</sub>	Data Mask Register 00	G0MSK0	XX <sub>16</sub>
00F5 <sub>16</sub>	Data Mask Register 01	G0MSK1	XX <sub>16</sub>
00F6 <sub>16</sub>	Communication Clock Select Register	CCS	XXXX 0000 <sub>2</sub>
00F7 <sub>16</sub>			
00F8 <sub>16</sub> 00F9 <sub>16</sub>	Receive CRC Code Register 0	G0RCRC	XX <sub>16</sub> XX <sub>16</sub>
00FA <sub>16</sub> 00FB <sub>16</sub>	Transmit CRC Code Register 0	G0TCRC	00 <sub>16</sub> 00 <sub>16</sub>
00FC <sub>16</sub>	SI/O Extended Mode Register 0	G0EMR	00 <sub>16</sub>
00FD <sub>16</sub>	SI/O Extended Receive Control Register 0	G0ERC	00 <sub>16</sub>
00FE <sub>16</sub>	SI/O Special Communication Interrupt Detect Register 0	G0IRF	00 <sub>16</sub>
00FF <sub>16</sub>	SI/O Extended Transmit Control Register 0	G0ETC	0000 0XXX <sub>2</sub>
0100 <sub>16</sub> 0101 <sub>16</sub>	Time Measurement/Waveform Generating Register 10	G1TM0/G1PO0	XX <sub>16</sub> XX <sub>16</sub>
0102 <sub>16</sub> 0103 <sub>16</sub>	Time Measurement/Waveform Generating Register 11	G1TM1/G1PO1	XX <sub>16</sub> XX <sub>16</sub>
0104 <sub>16</sub> 0105 <sub>16</sub>	Time Measurement/Waveform Generating Register 12	G1TM2/G1PO2	XX <sub>16</sub> XX <sub>16</sub>
0106 <sub>16</sub> 0107 <sub>16</sub>	Time Measurement/Waveform Generating Register 13	G1TM3/G1PO3	XX <sub>16</sub> XX <sub>16</sub>
0108 <sub>16</sub> 0109 <sub>16</sub>	Time Measurement/Waveform Generating Register 14	G1TM4/G1PO4	XX <sub>16</sub> XX <sub>16</sub>
010A <sub>16</sub> 010B <sub>16</sub>	Time Measurement/Waveform Generating Register 15	G1TM5/G1PO5	XX <sub>16</sub> XX <sub>16</sub>
010C <sub>16</sub> 010D <sub>16</sub>	Time Measurement/Waveform Generating Register 16	G1TM6/G1PO6	XX <sub>16</sub> XX <sub>16</sub>
010E <sub>16</sub> 010F <sub>16</sub>	Time Measurement/Waveform Generating Register 17	G1TM7/G1PO7	XX <sub>16</sub> XX <sub>16</sub>
0110 <sub>16</sub>	Waveform Generating Control Register 10	G1POCR0	0000 X000 <sub>2</sub>
0111 <sub>16</sub>	Waveform Generating Control Register 11	G1POCR1	0X00 X000 <sub>2</sub>
0112 <sub>16</sub>	Waveform Generating Control Register 12	G1POCR2	0X00 X000 <sub>2</sub>
0113 <sub>16</sub>	Waveform Generating Control Register 13	G1POCR3	0X00 X000 <sub>2</sub>
0114 <sub>16</sub>	Waveform Generating Control Register 14	G1POCR4	0X00 X000 <sub>2</sub>
0115 <sub>16</sub>	Waveform Generating Control Register 15	G1POCR5	0X00 X000 <sub>2</sub>
0116 <sub>16</sub>	Waveform Generating Control Register 16	G1POCR6	0X00 X000 <sub>2</sub>
0117 <sub>16</sub>	Waveform Generating Control Register 17	G1POCR7	0X00 X000 <sub>2</sub>
0118 <sub>16</sub>	Time Measurement Control Register 10	G1TMCR0	00 <sub>16</sub>
0119 <sub>16</sub>	Time Measurement Control Register 11	G1TMCR1	00 <sub>16</sub>
011A <sub>16</sub>	Time Measurement Control Register 12	G1TMCR2	00 <sub>16</sub>
011B <sub>16</sub>	Time Measurement Control Register 13	G1TMCR3	00 <sub>16</sub>
011C <sub>16</sub>	Time Measurement Control Register 14	G1TMCR4	00 <sub>16</sub>
011D <sub>16</sub>	Time Measurement Control Register 15	G1TMCR5	00 <sub>16</sub>
011E <sub>16</sub>	Time Measurement Control Register 16	G1TMCR6	00 <sub>16</sub>
011F <sub>16</sub>	Time Measurement Control Register 17	G1TMCR7	00 <sub>16</sub>

X: Indeterminate

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Value after RESET
0120 <sub>16</sub> 0121 <sub>16</sub>	Base Timer Register 1	G1BT	XX <sub>16</sub> XX <sub>16</sub>
0122 <sub>16</sub>	Base Timer Control Register 10	G1BCR0	00 <sub>16</sub>
0123 <sub>16</sub>	Base Timer Control Register 11	G1BCR1	X000 000X <sub>2</sub>
0124 <sub>16</sub>	Time Measurement Prescaler Register 16	G1TPR6	00 <sub>16</sub>
0125 <sub>16</sub>	Time Measurement Prescaler Register 17	G1TPR7	00 <sub>16</sub>
0126 <sub>16</sub>	Function Enable Register 1	G1FE	00 <sub>16</sub>
0127 <sub>16</sub>	Function Select Register 1	G1FS	00 <sub>16</sub>
0128 <sub>16</sub> 0129 <sub>16</sub>	SI/O Receive Buffer Register 1	G1RB	XXXX XXXX <sub>2</sub> X000 XXXX <sub>2</sub>
012A <sub>16</sub> 012B <sub>16</sub>	Transmit Buffer/Receive Data Register 1	G1TB/G1DR	XX <sub>16</sub>
012C <sub>16</sub>	Receive Input Register 1	G1RI	XX <sub>16</sub>
012D <sub>16</sub>	SI/O Communication Mode Register 1	G1MR	00 <sub>16</sub>
012E <sub>16</sub>	Transmit Output Register 1	G1TO	XX <sub>16</sub>
012F <sub>16</sub>	SI/O Communication Control Register 1	G1CR	0000 X011 <sub>2</sub>
0130 <sub>16</sub>	Data Compare Register 10	G1CMP0	XX <sub>16</sub>
0131 <sub>16</sub>	Data Compare Register 11	G1CMP1	XX <sub>16</sub>
0132 <sub>16</sub>	Data Compare Register 12	G1CMP2	XX <sub>16</sub>
0133 <sub>16</sub>	Data Compare Register 13	G1CMP3	XX <sub>16</sub>
0134 <sub>16</sub>	Data Mask Register 10	G1MSK0	XX <sub>16</sub>
0135 <sub>16</sub>	Data Mask Register 11	G1MSK1	XX <sub>16</sub>
0136 <sub>16</sub>			
0137 <sub>16</sub>			
0138 <sub>16</sub> 0139 <sub>16</sub>	Receive CRC Code Register 1	G1RCRC	XX <sub>16</sub> XX <sub>16</sub>
013A <sub>16</sub> 013B <sub>16</sub>	Transmit CRC Code Register 1	G1TCRC	00 <sub>16</sub> 00 <sub>16</sub>
013C <sub>16</sub>	SI/O Extended Mode Register 1	G1EMR	00 <sub>16</sub>
013D <sub>16</sub>	SI/O Extended Receive Control Register 1	G1ERC	00 <sub>16</sub>
013E <sub>16</sub>	SI/O Special Communication Interrupt Detection Register 1	G1IRF	00 <sub>16</sub>
013F <sub>16</sub>	SI/O Extended Transmit Control Register 1	G1ETC	0000 0XXX <sub>2</sub>
0140 <sub>16</sub>			
0141 <sub>16</sub>			
0142 <sub>16</sub>			
0143 <sub>16</sub>			
0144 <sub>16</sub>			
0145 <sub>16</sub>			
0146 <sub>16</sub>			
0147 <sub>16</sub>			
0148 <sub>16</sub>			
0149 <sub>16</sub>			
014A <sub>16</sub>			
014B <sub>16</sub>			
014C <sub>16</sub>			
014D <sub>16</sub> to 016F <sub>16</sub>			

X: Indeterminate

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Value after RESET
0170 <sub>16</sub>	CAN2 Slot Buffer Select Register	C2SBS	00 <sub>16</sub> <sup>(1)</sup>
0171 <sub>16</sub>	CAN2 Control Register 1	C2CTLR1	X000 00XX <sub>2</sub> <sup>(1)</sup>
0172 <sub>16</sub>	CAN2 Sleep Control Register	C2SLPR	XXXX XXX0 <sub>2</sub>
0173 <sub>16</sub>			
0174 <sub>16</sub>	CAN2 Acceptance Filter Support Register	C2AFS	00 <sub>16</sub> <sup>(1)</sup>
0175 <sub>16</sub>			01 <sub>16</sub> <sup>(1)</sup>
0176 <sub>16</sub>			
0177 <sub>16</sub>			
0178 <sub>16</sub>	Input Function Select Register	IPS	00 <sub>16</sub>
0179 <sub>16</sub>	Input Function Select Register A	IPSA	00 <sub>16</sub>
017A <sub>16</sub>			
017B <sub>16</sub>			
017C <sub>16</sub>			
017D <sub>16</sub>			
017E <sub>16</sub>			
017F <sub>16</sub>			
0180 <sub>16</sub>	CAN2 Message Slot Buffer 0 Standard ID0	C2SLOT0_0	XX <sub>16</sub>
0181 <sub>16</sub>	CAN2 Message Slot Buffer 0 Standard ID1	C2SLOT0_1	XX <sub>16</sub>
0182 <sub>16</sub>	CAN2 Message Slot Buffer 0 Extended ID0	C2SLOT0_2	XX <sub>16</sub>
0183 <sub>16</sub>	CAN2 Message Slot Buffer 0 Extended ID1	C2SLOT0_3	XX <sub>16</sub>
0184 <sub>16</sub>	CAN2 Message Slot Buffer 0 Extended ID2	C2SLOT0_4	XX <sub>16</sub>
0185 <sub>16</sub>	CAN2 Message Slot Buffer 0 Data Length Code	C2SLOT0_5	XX <sub>16</sub>
0186 <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 0	C2SLOT0_6	XX <sub>16</sub>
0187 <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 1	C2SLOT0_7	XX <sub>16</sub>
0188 <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 2	C2SLOT0_8	XX <sub>16</sub>
0189 <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 3	C2SLOT0_9	XX <sub>16</sub>
018A <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 4	C2SLOT0_10	XX <sub>16</sub>
018B <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 5	C2SLOT0_11	XX <sub>16</sub>
018C <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 6	C2SLOT0_12	XX <sub>16</sub>
018D <sub>16</sub>	CAN2 Message Slot Buffer 0 Data 7	C2SLOT0_13	XX <sub>16</sub>
018E <sub>16</sub>	CAN2 Message Slot Buffer 0 Time Stamp High-Order	C2SLOT0_14	XX <sub>16</sub>
018F <sub>16</sub>	CAN2 Message Slot Buffer 0 Time Stamp Low-Order	C2SLOT0_15	XX <sub>16</sub>
0190 <sub>16</sub>	CAN2 Message Slot Buffer 1 Standard ID0	C2SLOT1_0	XX <sub>16</sub>
0191 <sub>16</sub>	CAN2 Message Slot Buffer 1 Standard ID1	C2SLOT1_1	XX <sub>16</sub>
0192 <sub>16</sub>	CAN2 Message Slot Buffer 1 Extended ID0	C2SLOT1_2	XX <sub>16</sub>
0193 <sub>16</sub>	CAN2 Message Slot Buffer 1 Extended ID1	C2SLOT1_3	XX <sub>16</sub>
0194 <sub>16</sub>	CAN2 Message Slot Buffer 1 Extended ID2	C2SLOT1_4	XX <sub>16</sub>
0195 <sub>16</sub>	CAN2 Message Slot Buffer 1 Data Length Code	C2SLOT1_5	XX <sub>16</sub>
0196 <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 0	C2SLOT1_6	XX <sub>16</sub>
0197 <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 1	C2SLOT1_7	XX <sub>16</sub>
0198 <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 2	C2SLOT1_8	XX <sub>16</sub>
0199 <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 3	C2SLOT1_9	XX <sub>16</sub>
019A <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 4	C2SLOT1_10	XX <sub>16</sub>
019B <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 5	C2SLOT1_11	XX <sub>16</sub>
019C <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 6	C2SLOT1_12	XX <sub>16</sub>
019D <sub>16</sub>	CAN2 Message Slot Buffer 1 Data 7	C2SLOT1_13	XX <sub>16</sub>
019E <sub>16</sub>	CAN2 Message Slot Buffer 1 Time Stamp High-Order	C2SLOT1_14	XX <sub>16</sub>
019F <sub>16</sub>	CAN2 Message Slot Buffer 1 Time Stamp Low-Order	C2SLOT1_15	XX <sub>16</sub>

X: Indeterminate

Blank spaces are reserved. No access is allowed.

NOTE:

1. Values are obtained by setting the SLEEP bit in the C2SLPR register to "1" (sleep mode exited) after reset and supplying a clock to the CAN module.

Address	Register	Symbol	Value after RESET
01A0 <sub>16</sub> 01A1 <sub>16</sub>	CAN2 Control Register 0	C2CTLR0	XX01 0X01 <sub>2</sub> <sup>(2)</sup> XXXX 0000 <sub>2</sub> <sup>(2)</sup>
01A2 <sub>16</sub> 01A3 <sub>16</sub>	CAN2 Status Register	C2STR	0000 0000 <sub>2</sub> <sup>(2)</sup> X000 0X01 <sub>2</sub> <sup>(2)</sup>
01A4 <sub>16</sub> 01A5 <sub>16</sub>	CAN2 Extended ID Register	C2IDR	00 <sub>16</sub> <sup>(2)</sup> 00 <sub>16</sub> <sup>(2)</sup>
01A6 <sub>16</sub> 01A7 <sub>16</sub>	CAN2 Configuration Register	C2CONR	0000 XXXX <sub>2</sub> <sup>(2)</sup> 0000 0000 <sub>2</sub> <sup>(2)</sup>
01A8 <sub>16</sub> 01A9 <sub>16</sub>	CAN2 Time Stamp Register	C2TSR	00 <sub>16</sub> <sup>(2)</sup> 00 <sub>16</sub> <sup>(2)</sup>
01AA <sub>16</sub>	CAN2 Transmit Error Count Register	C2TEC	00 <sub>16</sub> <sup>(2)</sup>
01AB <sub>16</sub>	CAN2 Receive Error Count Register	C2REC	00 <sub>16</sub> <sup>(2)</sup>
01AC <sub>16</sub> 01AD <sub>16</sub>	CAN2 Slot Interrupt Status Register	C2SISTR	00 <sub>16</sub> <sup>(2)</sup> 00 <sub>16</sub> <sup>(2)</sup>
01AE <sub>16</sub>			
01AF <sub>16</sub>			
01B0 <sub>16</sub> 01B1 <sub>16</sub>	CAN2 Slot Interrupt Mask Register	C2SIMKR	00 <sub>16</sub> <sup>(2)</sup> 00 <sub>16</sub> <sup>(2)</sup>
01B2 <sub>16</sub>			
01B3 <sub>16</sub>			
01B4 <sub>16</sub>	CAN2 Error Interrupt Mask Register	C2EIMKR	XXXX X000 <sub>2</sub> <sup>(2)</sup>
01B5 <sub>16</sub>	CAN2 Error Interrupt Status Register	C2EISTR	XXXX X000 <sub>2</sub> <sup>(2)</sup>
01B6 <sub>16</sub>	CAN2 Error Cause Register	C2EFR	00 <sub>16</sub> <sup>(2)</sup>
01B7 <sub>16</sub>	CAN2 Baud Rate Prescaler	C2BRP	0000 0001 <sub>2</sub> <sup>(2)</sup>
01B8 <sub>16</sub>			
01B9 <sub>16</sub>	CAN2 Mode Register	C2MDR	XXXX XX00 <sub>2</sub> <sup>(2)</sup>
01BA <sub>16</sub>			
01BB <sub>16</sub>			
01BC <sub>16</sub>			
01BD <sub>16</sub>			
01BE <sub>16</sub>			
01BF <sub>16</sub>			
01C0 <sub>16</sub> 01C1 <sub>16</sub>	CAN2 Single Shot Control Register	C2SSCTLR	00 <sub>16</sub> <sup>(2)</sup> 00 <sub>16</sub> <sup>(2)</sup>
01C2 <sub>16</sub>			
01C3 <sub>16</sub>			
01C4 <sub>16</sub> 01C5 <sub>16</sub>	CAN2 Single Shot Status Register	C2SSSTR	00 <sub>16</sub> <sup>(2)</sup> 00 <sub>16</sub> <sup>(2)</sup>
01C6 <sub>16</sub>			
01C7 <sub>16</sub>			
01C8 <sub>16</sub>	CAN2 Global Mask Register Standard ID0	C2GMR0	XXX0 0000 <sub>2</sub> <sup>(2)</sup>
01C9 <sub>16</sub>	CAN2 Global Mask Register Standard ID1	C2GMR1	XX00 0000 <sub>2</sub> <sup>(2)</sup>
01CA <sub>16</sub>	CAN2 Global Mask Register Extended ID0	C2GMR2	XXXX 0000 <sub>2</sub> <sup>(2)</sup>
01CB <sub>16</sub>	CAN2 Global Mask Register Extended ID1	C2GMR3	00 <sub>16</sub> <sup>(2)</sup>
01CC <sub>16</sub>	CAN2 Global Mask Register Extended ID2	C2GMR4	XX00 0000 <sub>2</sub> <sup>(2)</sup>
01CD <sub>16</sub>			
01CE <sub>16</sub>			
01CF <sub>16</sub>			

(Note 1)

X: Indeterminate

Blank spaces are reserved. No access is allowed.

## NOTES:

1. The BANKSEL bit in the C2CTLR1 register switches functions for addresses 01C0<sub>16</sub> to 01DF<sub>16</sub>.
2. Values are obtained by setting the SLEEP bit in the C2SLPR register to "1" (sleep mode exited) after reset and supplying a clock to the CAN module.

Address	Register	Symbol	Value after RESET
01D0 <sub>16</sub>	CAN2 Message Slot 0 Control Register /	C2MCTL0/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN2 Local Mask Register A Standard ID0	C2LMAR0	XXX0 0000 <sub>2</sub> <sup>(2)</sup>
01D1 <sub>16</sub>	CAN2 Message Slot 1 Control Register /	C2MCTL1/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN2 Local Mask Register A Standard ID1	C2LMAR1	XX00 0000 <sub>2</sub> <sup>(2)</sup>
01D2 <sub>16</sub>	CAN2 Message Slot 2 Control Register /	C2MCTL2/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN2 Local Mask Register A Extended ID0	C2LMAR2	XXXX 0000 <sub>2</sub> <sup>(2)</sup>
01D3 <sub>16</sub>	CAN2 Message Slot 3 Control Register /	C2MCTL3/	00 <sub>16</sub> <sup>(2)</sup>
	CAN2 local Mask Register A Extended ID1	C2LMAR3	00 <sub>16</sub> <sup>(2)</sup>
01D4 <sub>16</sub>	CAN2 Message Slot 4 Control Register /	C2MCTL4/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN2 Local Mask Register A Extended ID2	C2LMAR4	XX00 0000 <sub>2</sub> <sup>(2)</sup>
01D5 <sub>16</sub>	CAN2 Message Slot 5 Control Register	C2MCTL5	00 <sub>16</sub> <sup>(2)</sup>
01D6 <sub>16</sub>	CAN2 Message Slot 6 Control Register	C2MCTL6	00 <sub>16</sub> <sup>(2)</sup>
01D7 <sub>16</sub>	CAN2 Message Slot 7 Control Register	C2MCTL7	00 <sub>16</sub> <sup>(2)</sup>
01D8 <sub>16</sub>	CAN2 Message Slot 8 Control Register /	C2MCTL8/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN2 Local Mask Register B Standard ID0	C2LMBR0	XXX0 0000 <sub>2</sub> <sup>(2)</sup>
01D9 <sub>16</sub>	CAN2 Message Slot 9 Control Register /	C2MCTL9/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN2 Local Mask Register B Standard ID1	C2LMBR1	XX00 0000 <sub>2</sub> <sup>(2)</sup>
01DA <sub>16</sub>	CAN2 Message Slot 10 Control Register /	C2MCTL10/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN2 Local Mask Register B Extended ID2	C2LMBR2	XXXX 0000 <sub>2</sub> <sup>(2)</sup>
01DB <sub>16</sub>	CAN2 Message Slot 11 Control Register /	C2MCTL11/	00 <sub>16</sub> <sup>(2)</sup>
	CAN2 Local Mask Register B Extended ID3	C2LMBR3	00 <sub>16</sub> <sup>(2)</sup>
01DC <sub>16</sub>	CAN2 Message Slot 12 Control Register /	C2MCTL12/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN2 Local Mask Register B Extended ID4	C2LMBR4	XX00 0000 <sub>2</sub> <sup>(2)</sup>
01DD <sub>16</sub>	CAN2 Message Slot 13 Control Register	C2MCTL13	00 <sub>16</sub> <sup>(2)</sup>
01DE <sub>16</sub>	CAN2 Message Slot 14 Control Register	C2MCTL14	00 <sub>16</sub> <sup>(2)</sup>
01DF <sub>16</sub>	CAN2 Message Slot 15 Control Register	C2MCTL15	00 <sub>16</sub> <sup>(2)</sup>

(Note 1)

X: Indeterminate

Blank spaces are reserved. No access is allowed.

## NOTES:

1. The BANKSEL bit in the C2CTLR1 register switches functions for addresses 01C0<sub>16</sub> to 01DF<sub>16</sub>.
2. Values are obtained by setting the SLEEP bit in the C2SLPR register to "1" (sleep mode exited) after reset and supplying a clock to the CAN module.

Address	Register	Symbol	Value after RESET
01E0 <sub>16</sub>	CAN0 Message Slot Buffer 0 Standard ID0	C0SLOT0_0	XX <sub>16</sub>
01E1 <sub>16</sub>	CAN0 Message Slot Buffer 0 Standard ID1	C0SLOT0_1	XX <sub>16</sub>
01E2 <sub>16</sub>	CAN0 Message Slot Buffer 0 Extended ID0	C0SLOT0_2	XX <sub>16</sub>
01E3 <sub>16</sub>	CAN0 Message Slot Buffer 0 Extended ID1	C0SLOT0_3	XX <sub>16</sub>
01E4 <sub>16</sub>	CAN0 Message Slot Buffer 0 Extended ID2	C0SLOT0_4	XX <sub>16</sub>
01E5 <sub>16</sub>	CAN0 Message Slot Buffer 0 Data Length Code	C0SLOT0_5	XX <sub>16</sub>
01E6 <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 0	C0SLOT0_6	XX <sub>16</sub>
01E7 <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 1	C0SLOT0_7	XX <sub>16</sub>
01E8 <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 2	C0SLOT0_8	XX <sub>16</sub>
01E9 <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 3	C0SLOT0_9	XX <sub>16</sub>
01EA <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 4	C0SLOT0_10	XX <sub>16</sub>
01EB <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 5	C0SLOT0_11	XX <sub>16</sub>
01EC <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 6	C0SLOT0_12	XX <sub>16</sub>
01ED <sub>16</sub>	CAN0 Message Slot Buffer 0 Data 7	C0SLOT0_13	XX <sub>16</sub>
01EE <sub>16</sub>	CAN0 Message Slot Buffer 0 Time Stamp High-Order	C0SLOT0_14	XX <sub>16</sub>
01EF <sub>16</sub>	CAN0 Message Slot Buffer 0 Time Stamp Low-Order	C0SLOT0_15	XX <sub>16</sub>
01F0 <sub>16</sub>	CAN0 Message Slot Buffer 1 Standard ID0	C0SLOT1_0	XX <sub>16</sub>
01F1 <sub>16</sub>	CAN0 Message Slot Buffer 1 Standard ID1	C0SLOT1_1	XX <sub>16</sub>
01F2 <sub>16</sub>	CAN0 Message Slot Buffer 1 Extended ID0	C0SLOT1_2	XX <sub>16</sub>
01F3 <sub>16</sub>	CAN0 Message Slot Buffer 1 Extended ID1	C0SLOT1_3	XX <sub>16</sub>
01F4 <sub>16</sub>	CAN0 Message Slot Buffer 1 Extended ID2	C0SLOT1_4	XX <sub>16</sub>
01F5 <sub>16</sub>	CAN0 Message Slot Buffer 1 Data Length Code	C0SLOT1_5	XX <sub>16</sub>
01F6 <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 0	C0SLOT1_6	XX <sub>16</sub>
01F7 <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 1	C0SLOT1_7	XX <sub>16</sub>
01F8 <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 2	C0SLOT1_8	XX <sub>16</sub>
01F9 <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 3	C0SLOT1_9	XX <sub>16</sub>
01FA <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 4	C0SLOT1_10	XX <sub>16</sub>
01FB <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 5	C0SLOT1_11	XX <sub>16</sub>
01FC <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 6	C0SLOT1_12	XX <sub>16</sub>
01FD <sub>16</sub>	CAN0 Message Slot Buffer 1 Data 7	C0SLOT1_13	XX <sub>16</sub>
01FE <sub>16</sub>	CAN0 Message Slot Buffer 1 Time Stamp High-Order	C0SLOT1_14	XX <sub>16</sub>
01FF <sub>16</sub>	CAN0 Message Slot Buffer 1 Time Stamp Low-Order	C0SLOT1_15	XX <sub>16</sub>
0200 <sub>16</sub> 0201 <sub>16</sub>	CAN0 Control Register 0	C0CTRL0	XX01 0X01 <sub>2</sub> <sup>(1)</sup> XXXX 0000 <sub>2</sub> <sup>(1)</sup>
0202 <sub>16</sub> 0203 <sub>16</sub>	CAN0 Status Register	C0STR	0000 0000 <sub>2</sub> <sup>(1)</sup> X000 0X01 <sub>2</sub> <sup>(1)</sup>
0204 <sub>16</sub> 0205 <sub>16</sub>	CAN0 Extended ID Register	C0IDR	00 <sub>16</sub> <sup>(1)</sup> 00 <sub>16</sub> <sup>(1)</sup>
0206 <sub>16</sub> 0207 <sub>16</sub>	CAN0 Configuration Register	C0CONR	0000 XXXX <sub>2</sub> <sup>(1)</sup> 0000 0000 <sub>2</sub> <sup>(1)</sup>
0208 <sub>16</sub> 0209 <sub>16</sub>	CAN0 Time Stamp Register	C0TSR	00 <sub>16</sub> <sup>(1)</sup> 00 <sub>16</sub> <sup>(1)</sup>
020A <sub>16</sub>	CAN0 Transmit Error Count Register	C0TEC	00 <sub>16</sub> <sup>(1)</sup>
020B <sub>16</sub>	CAN0 Receive Error Count Register	C0REC	00 <sub>16</sub> <sup>(1)</sup>
020C <sub>16</sub> 020D <sub>16</sub>	CAN0 Slot Interrupt Status Register	C0SISTR	00 <sub>16</sub> <sup>(1)</sup> 00 <sub>16</sub> <sup>(1)</sup>
020E <sub>16</sub>			
020F <sub>16</sub>			

X: Indeterminate

Blank spaces are reserved. No access is allowed.

NOTE:

1. Values are obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) after reset and supplying a clock to the CAN module.

Address	Register	Symbol	Value after RESET
0210 <sub>16</sub>	CAN0 Slot Interrupt Mask Register	C0SIMKR	00 <sub>16</sub> <sup>(2)</sup>
0211 <sub>16</sub>			00 <sub>16</sub> <sup>(2)</sup>
0212 <sub>16</sub>			
0213 <sub>16</sub>			
0214 <sub>16</sub>	CAN0 Error Interrupt Mask Register	C0EIMKR	XXXX X000 <sub>2</sub> <sup>(2)</sup>
0215 <sub>16</sub>	CAN0 Error Interrupt Status Register	C0EISTR	XXXX X000 <sub>2</sub> <sup>(2)</sup>
0216 <sub>16</sub>	CAN0 Error Cause Register	C0EFR	00 <sub>16</sub> <sup>(2)</sup>
0217 <sub>16</sub>	CAN0 Baud Rate Prescaler	C0BRP	0000 0001 <sub>2</sub> <sup>(2)</sup>
0218 <sub>16</sub>			
0219 <sub>16</sub>	CAN0 Mode Register	C0MDR	XXXX XX00 <sub>2</sub> <sup>(2)</sup>
021A <sub>16</sub>			
021B <sub>16</sub>			
021C <sub>16</sub>			
021D <sub>16</sub>			
021E <sub>16</sub>			
021F <sub>16</sub>			
0220 <sub>16</sub>	CAN0 Single Shot Control Register	C0SSCTLR	00 <sub>16</sub> <sup>(2)</sup>
0221 <sub>16</sub>			00 <sub>16</sub> <sup>(2)</sup>
0222 <sub>16</sub>			
0223 <sub>16</sub>			
0224 <sub>16</sub>	CAN0 Single Shot Status Register	C0SSSTR	00 <sub>16</sub> <sup>(2)</sup>
0225 <sub>16</sub>			00 <sub>16</sub> <sup>(2)</sup>
0226 <sub>16</sub>			
0227 <sub>16</sub>			
0228 <sub>16</sub>	CAN0 Global Mask Register Standard ID0	C0GMR0	XXX0 0000 <sub>2</sub> <sup>(2)</sup>
0229 <sub>16</sub>	CAN0 Global Mask Register Standard ID1	C0GMR1	XX00 0000 <sub>2</sub> <sup>(2)</sup>
022A <sub>16</sub>	CAN0 Global Mask Register Extended ID0	C0GMR2	XXXX 0000 <sub>2</sub> <sup>(2)</sup>
022B <sub>16</sub>	CAN0 Global Mask Register Extended ID1	C0GMR3	00 <sub>16</sub> <sup>(2)</sup>
022C <sub>16</sub>	CAN0 Global Mask Register Extended ID2	C0GMR4	XX00 0000 <sub>2</sub> <sup>(2)</sup>
022D <sub>16</sub>			
022E <sub>16</sub>			
022F <sub>16</sub>			
0230 <sub>16</sub>	CAN0 Message Slot 0 Control Register / CAN0 Local Mask Register A Standard ID0	C0MCTL0/ C0LMAR0	0000 0000 <sub>2</sub> <sup>(2)</sup> XXX0 0000 <sub>2</sub> <sup>(2)</sup>
0231 <sub>16</sub>	CAN0 Message Slot 1 Control Register / CAN0 Local Mask Register A Standard ID1	C0MCTL1/ C0LMAR1	0000 0000 <sub>2</sub> <sup>(2)</sup> XX00 0000 <sub>2</sub> <sup>(2)</sup>
0232 <sub>16</sub>	CAN0 Message Slot 2 Control Register / CAN0 Local Mask Register A Extended ID0	C0MCTL2/ C0LMAR2	0000 0000 <sub>2</sub> <sup>(2)</sup> XXXX 0000 <sub>2</sub> <sup>(2)</sup>
0233 <sub>16</sub>	CAN0 Message Slot 3 Control Register / CAN0 local Mask Register A Extended ID1	C0MCTL3/ C0LMAR3	00 <sub>16</sub> <sup>(2)</sup> 00 <sub>16</sub> <sup>(2)</sup>
0234 <sub>16</sub>	CAN0 Message Slot 4 Control Register / CAN0 Local Mask Register A Extended ID2	C0MCTL4/ C0LMAR4	0000 0000 <sub>2</sub> <sup>(2)</sup> XX00 0000 <sub>2</sub> <sup>(2)</sup>
0235 <sub>16</sub>	CAN0 Message Slot 5 Control Register	C0MCTL5	00 <sub>16</sub> <sup>(2)</sup>
0236 <sub>16</sub>	CAN0 Message Slot 6 Control Register	C0MCTL6	00 <sub>16</sub> <sup>(2)</sup>
0237 <sub>16</sub>	CAN0 Message Slot 7 Control Register	C0MCTL7	00 <sub>16</sub> <sup>(2)</sup>
0238 <sub>16</sub>	CAN0 Message Slot 8 Control Register / CAN0 Local Mask Register B Standard ID0	C0MCTL8/ C0LMBR0	0000 0000 <sub>2</sub> <sup>(2)</sup> XXX0 0000 <sub>2</sub> <sup>(2)</sup>
0239 <sub>16</sub>	CAN0 Message Slot 9 Control Register / CAN0 Local Mask Register B Standard ID1	C0MCTL9/ C0LMBR1	0000 0000 <sub>2</sub> <sup>(2)</sup> XX00 0000 <sub>2</sub> <sup>(2)</sup>

(Note 1)

X: Indeterminate

Blank spaces are reserved. No access is allowed.

NOTES:

1. The BANKSEL bit in the C0CTLR1 register switches functions for addresses 0220<sub>16</sub> to 023F<sub>16</sub>.
2. Values are obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) after reset and supplying a clock to the CAN module.

Address	Register	Symbol	Value after RESET
023A <sub>16</sub>	CAN0 Message Slot 10 Control Register / CAN0 Local Mask Register B Extended ID0	C0MCTL10/ C0LMBR2	0000 0000 <sub>2</sub> <sup>(2)</sup> XXXX 0000 <sub>2</sub> <sup>(2)</sup>
023B <sub>16</sub>	CAN0 Message Slot 11 Control Register / CAN0 Local Mask Register B Extended ID1	C0MCTL11/ C0LMBR3	00 <sub>16</sub> <sup>(2)</sup> 00 <sub>16</sub> <sup>(2)</sup>
023C <sub>16</sub>	CAN0 Message Slot 12 Control Register / CAN0 Local Mask Register B Extended ID2	C0MCTL12/ C0LMBR4	0000 0000 <sub>2</sub> <sup>(2)</sup> XX00 0000 <sub>2</sub> <sup>(2)</sup>
023D <sub>16</sub>	CAN0 Message Slot 13 Control Register	C0MCTL13	00 <sub>16</sub> <sup>(2)</sup>
023E <sub>16</sub>	CAN0 Message Slot 14 Control Register	C0MCTL14	00 <sub>16</sub> <sup>(2)</sup>
023F <sub>16</sub>	CAN0 Message Slot 15 Control Register	C0MCTL15	00 <sub>16</sub> <sup>(2)</sup>
0240 <sub>16</sub>	CAN0 Slot Buffer Select Register	C0SBS	00 <sub>16</sub> <sup>(2)</sup>
0241 <sub>16</sub>	CAN0 Control Register 1	C0CTLR1	X000 00XX <sub>2</sub> <sup>(2)</sup>
0242 <sub>16</sub>	CAN0 Sleep Control Register	C0SLPR	XXXX XXX0 <sub>2</sub>
0243 <sub>16</sub>			
0244 <sub>16</sub> 0245 <sub>16</sub>	CAN0 Acceptance Filter Support Register	C0AFS	00 <sub>16</sub> <sup>(2)</sup> 01 <sub>16</sub> <sup>(2)</sup>
0246 <sub>16</sub>			
0247 <sub>16</sub>			
0248 <sub>16</sub>			
0249 <sub>16</sub>			
024A <sub>16</sub>			
024B <sub>16</sub>			
024C <sub>16</sub>			
024D <sub>16</sub>			
024E <sub>16</sub>			
024F <sub>16</sub>			
0250 <sub>16</sub>	CAN1 Slot Buffer Select Register	C1SBS	00 <sub>16</sub> <sup>(3)</sup>
0251 <sub>16</sub>	CAN1 Control Register 1	C1CTLR1	X000 00XX <sub>2</sub> <sup>(3)</sup>
0252 <sub>16</sub>	CAN1 Sleep Control Register	C1SLPR	XXXX XXX0 <sub>2</sub> <sup>(3)</sup>
0253 <sub>16</sub>			
0254 <sub>16</sub> 0255 <sub>16</sub>	CAN1 Acceptance Filter Support Register	C1AFS	00 <sub>16</sub> <sup>(3)</sup> 01 <sub>16</sub> <sup>(3)</sup>
0256 <sub>16</sub>			
0257 <sub>16</sub>			
0258 <sub>16</sub>			
0259 <sub>16</sub>			
025A <sub>16</sub>			
025B <sub>16</sub>			
025C <sub>16</sub>			
025D <sub>16</sub>			
025E <sub>16</sub>			
025F <sub>16</sub>			

(Note 1)

X: Indeterminate

Blank spaces are reserved. No access is allowed.

## NOTES:

1. The BANKSEL bit in the C0CTLR1 register switches functions for addresses 0220<sub>16</sub> to 023F<sub>16</sub>.
2. Values are obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) after reset and supplying a clock to the CAN module.
3. Values are obtained by setting the SLEEP bit in the C1SLPR register to "1" (sleep mode exited) after reset and supplying a clock to the CAN module.

Address	Register	Symbol	Value after RESET
0260 <sub>16</sub>	CAN1 Message Slot Buffer 0 Standard ID0	C1SLOT0_0	XX <sub>16</sub>
0261 <sub>16</sub>	CAN1 Message Slot Buffer 0 Standard ID1	C1SLOT0_1	XX <sub>16</sub>
0262 <sub>16</sub>	CAN1 Message Slot Buffer 0 Extended ID0	C1SLOT0_2	XX <sub>16</sub>
0263 <sub>16</sub>	CAN1 Message Slot Buffer 0 Extended ID1	C1SLOT0_3	XX <sub>16</sub>
0264 <sub>16</sub>	CAN1 Message Slot Buffer 0 Extended ID2	C1SLOT0_4	XX <sub>16</sub>
0265 <sub>16</sub>	CAN1 Message Slot Buffer 0 Data Length Code	C1SLOT0_5	XX <sub>16</sub>
0266 <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 0	C1SLOT0_6	XX <sub>16</sub>
0267 <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 1	C1SLOT0_7	XX <sub>16</sub>
0268 <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 2	C1SLOT0_8	XX <sub>16</sub>
0269 <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 3	C1SLOT0_9	XX <sub>16</sub>
026A <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 4	C1SLOT0_10	XX <sub>16</sub>
026B <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 5	C1SLOT0_11	XX <sub>16</sub>
026C <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 6	C1SLOT0_12	XX <sub>16</sub>
026D <sub>16</sub>	CAN1 Message Slot Buffer 0 Data 7	C1SLOT0_13	XX <sub>16</sub>
026E <sub>16</sub>	CAN1 Message Slot Buffer 0 Time Stamp High-Order	C1SLOT0_14	XX <sub>16</sub>
026F <sub>16</sub>	CAN1 Message Slot Buffer 0 Time Stamp Low-Order	C1SLOT0_15	XX <sub>16</sub>
0270 <sub>16</sub>	CAN1 Message Slot Buffer 1 Standard ID0	C1SLOT1_0	XX <sub>16</sub>
0271 <sub>16</sub>	CAN1 Message Slot Buffer 1 Standard ID1	C1SLOT1_1	XX <sub>16</sub>
0272 <sub>16</sub>	CAN1 Message Slot Buffer 1 Extended ID0	C1SLOT1_2	XX <sub>16</sub>
0273 <sub>16</sub>	CAN1 Message Slot Buffer 1 Extended ID1	C1SLOT1_3	XX <sub>16</sub>
0274 <sub>16</sub>	CAN1 Message Slot Buffer 1 Extended ID2	C1SLOT1_4	XX <sub>16</sub>
0275 <sub>16</sub>	CAN1 Message Slot Buffer 1 Data Length Code	C1SLOT1_5	XX <sub>16</sub>
0276 <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 0	C1SLOT1_6	XX <sub>16</sub>
0277 <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 1	C1SLOT1_7	XX <sub>16</sub>
0278 <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 2	C1SLOT1_8	XX <sub>16</sub>
0279 <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 3	C1SLOT1_9	XX <sub>16</sub>
027A <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 4	C1SLOT1_10	XX <sub>16</sub>
027B <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 5	C1SLOT1_11	XX <sub>16</sub>
027C <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 6	C1SLOT1_12	XX <sub>16</sub>
027D <sub>16</sub>	CAN1 Message Slot Buffer 1 Data 7	C1SLOT1_13	XX <sub>16</sub>
027E <sub>16</sub>	CAN1 Message Slot Buffer 1 Time Stamp High-Order	C1SLOT1_14	XX <sub>16</sub>
027F <sub>16</sub>	CAN1 Message Slot Buffer 1 Time Stamp Low-Order	C1SLOT1_15	XX <sub>16</sub>
0280 <sub>16</sub> 0281 <sub>16</sub>	CAN1 Control Register 0	C1CTRL0	XX01 0X01 <sub>2</sub> <sup>(1)</sup> XXXX 0000 <sub>2</sub> <sup>(1)</sup>
0282 <sub>16</sub> 0283 <sub>16</sub>	CAN1 Status Register	C1STR	0000 0000 <sub>2</sub> <sup>(1)</sup> X000 0X01 <sub>2</sub> <sup>(1)</sup>
0284 <sub>16</sub> 0285 <sub>16</sub>	CAN1 Extended ID Register	C1IDR	00 <sub>16</sub> <sup>(1)</sup> 00 <sub>16</sub> <sup>(1)</sup>
0286 <sub>16</sub> 0287 <sub>16</sub>	CAN1 Configuration Register	C1CONR	0000 XXXX <sub>2</sub> <sup>(1)</sup> 0000 0000 <sub>2</sub> <sup>(1)</sup>
0288 <sub>16</sub> 0289 <sub>16</sub>	CAN1 Time Stamp Register	C1TSR	00 <sub>16</sub> <sup>(1)</sup> 00 <sub>16</sub> <sup>(1)</sup>
028A <sub>16</sub>	CAN1 Transmit Error Count Register	C1TEC	00 <sub>16</sub> <sup>(1)</sup>
028B <sub>16</sub>	CAN1 Receive Error Count Register	C1REC	00 <sub>16</sub> <sup>(1)</sup>
028C <sub>16</sub> 028D <sub>16</sub>	CAN1 Slot Interrupt Status Register	C1SISTR	00 <sub>16</sub> <sup>(1)</sup> 00 <sub>16</sub> <sup>(1)</sup>
028E <sub>16</sub>			
028F <sub>16</sub>			

X: Indeterminate

Blank spaces are reserved. No access is allowed.

NOTE:

1. Values are obtained by setting the SLEEP bit in the C1SLPR register to "1" (sleep mode exited) after reset and supplying the clock to the CAN module.

Address	Register	Symbol	Value after RESET
0290 <sub>16</sub>	CAN1 Slot Interrupt Mask Register	C1SIMKR	00 <sub>16</sub> <sup>(2)</sup>
0291 <sub>16</sub>			00 <sub>16</sub> <sup>(2)</sup>
0292 <sub>16</sub>			
0293 <sub>16</sub>			
0294 <sub>16</sub>	CAN1 Error Interrupt Mask Register	C1EIMKR	XXXX X000 <sub>2</sub> <sup>(2)</sup>
0295 <sub>16</sub>	CAN1 Error Interrupt Status Register	C1EISTR	XXXX X000 <sub>2</sub> <sup>(2)</sup>
0296 <sub>16</sub>	CAN1 Error Factor Register	C1EFR	00 <sub>16</sub> <sup>(2)</sup>
0297 <sub>16</sub>	CAN1 Baud Rate Prescaler	C1BRP	0000 0001 <sub>2</sub> <sup>(2)</sup>
0298 <sub>16</sub>			
0299 <sub>16</sub>	CAN1 Mode Register	C1MDR	XXXX XX00 <sub>2</sub> <sup>(2)</sup>
029A <sub>16</sub>			
029B <sub>16</sub>			
029C <sub>16</sub>			
029D <sub>16</sub>			
029E <sub>16</sub>			
029F <sub>16</sub>			
02A0 <sub>16</sub>	CAN1 Single Shot Control Register	C1SSCTLR	00 <sub>16</sub> <sup>(2)</sup>
02A1 <sub>16</sub>			00 <sub>16</sub> <sup>(2)</sup>
02A2 <sub>16</sub>			
02A3 <sub>16</sub>			
02A4 <sub>16</sub>	CAN1 Single Shot Status Register	C1SSSTR	00 <sub>16</sub> <sup>(2)</sup>
02A5 <sub>16</sub>			00 <sub>16</sub> <sup>(2)</sup>
02A6 <sub>16</sub>			
02A7 <sub>16</sub>			
02A8 <sub>16</sub>	CAN1 Global Mask Register Standard ID0	C1GMR0	XXX0 0000 <sub>2</sub> <sup>(2)</sup>
02A9 <sub>16</sub>	CAN1 Global Mask Register Standard ID1	C1GMR1	XX00 0000 <sub>2</sub> <sup>(2)</sup>
02AA <sub>16</sub>	CAN1 Global Mask Register Extended ID0	C1GMR2	XXXX 0000 <sub>2</sub> <sup>(2)</sup>
02AB <sub>16</sub>	CAN1 Global Mask Register Extended ID1	C1GMR3	00 <sub>16</sub> <sup>(2)</sup>
02AC <sub>16</sub>	CAN1 Global Mask Register Extended ID2	C1GMR4	XX00 0000 <sub>2</sub> <sup>(2)</sup>
02AD <sub>16</sub>			
02AE <sub>16</sub>			
02AF <sub>16</sub>			
02B0 <sub>16</sub>	CAN1 Message Slot 0 Control Register /	C1MCTL0/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN1 Local Mask Register A Standard ID0	C1LMAR0	XXX0 0000 <sub>2</sub> <sup>(2)</sup>
02B1 <sub>16</sub>	CAN1 Message Slot 1 Control Register /	C1MCTL1/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN1 Local Mask Register A Standard ID1	C1LMAR1	XX00 0000 <sub>2</sub> <sup>(2)</sup>
02B2 <sub>16</sub>	CAN1 Message Slot 2 Control Register /	C1MCTL2/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN1 Local Mask Register A Extended ID0	C1LMAR2	XXXX 0000 <sub>2</sub> <sup>(2)</sup>
02B3 <sub>16</sub>	CAN1 Message Slot 3 Control Register /	C1MCTL3/	00 <sub>16</sub> <sup>(2)</sup>
	CAN1 Local Mask Register A Extended ID1	C1LMAR3	00 <sub>16</sub> <sup>(2)</sup>
02B4 <sub>16</sub>	CAN1 Message Slot 4 Control Register /	C1MCTL4/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN1 Local Mask Register A Extended ID2	C1LMAR4	XX00 0000 <sub>2</sub> <sup>(2)</sup>
02B5 <sub>16</sub>	CAN1 Message Slot 5 Control Register	C1MCTL5	00 <sub>16</sub> <sup>(2)</sup>
02B6 <sub>16</sub>	CAN1 Message Slot 6 Control Register	C1MCTL6	00 <sub>16</sub> <sup>(2)</sup>
02B7 <sub>16</sub>	CAN1 Message Slot 7 Control Register	C1MCTL7	00 <sub>16</sub> <sup>(2)</sup>
02B8 <sub>16</sub>	CAN1 Message Slot 8 Control Register /	C1MCTL8/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN1 Local Mask Register B Standard ID0	C1LMBR0	XXX0 0000 <sub>2</sub> <sup>(2)</sup>
02B9 <sub>16</sub>	CAN1 Message Slot 9 Control Register /	C1MCTL9/	0000 0000 <sub>2</sub> <sup>(2)</sup>
	CAN1 Local Mask Register B Standard ID1	C1LMBR1	XX00 0000 <sub>2</sub> <sup>(2)</sup>

(Note 1)

X: Indeterminate

Blank spaces are reserved. No access is allowed.

## NOTES:

1. The BANKSEL bit in the C1CTLR1 register switches functions for addresses 02A0<sub>16</sub> to 02BF<sub>16</sub>.
2. Values are obtained by setting the SLEEP bit in the C1SLPR register to "1" (sleep mode exited) after reset and supplying a clock to the CAN module.

Address	Register	Symbol	Value after RESET
02BA <sub>16</sub>	CAN1 Message Slot 10 Control Register / CAN1 Local Mask Register B Extended ID0	C1MCTL10/ C1LMBR2	0000 0000 <sub>2</sub> <sup>(2)</sup> XXXX 0000 <sub>2</sub> <sup>(2)</sup>
02BB <sub>16</sub>	CAN1 Message Slot 11 Control Register / CAN1 Local Mask Register B Extended ID1	C1MCTL11/ C1LMBR3	00 <sub>16</sub> <sup>(2)</sup> 00 <sub>16</sub> <sup>(2)</sup>
02BC <sub>16</sub>	CAN1 Message Slot 12 Control Register / CAN1 Local Mask Register B Extended ID2	C1MCTL12/ C1LMBR4	0000 0000 <sub>2</sub> <sup>(2)</sup> XX00 0000 <sub>2</sub> <sup>(2)</sup>
02BD <sub>16</sub>	CAN1 Message Slot 13 Control Register	C1MCTL13	00 <sub>16</sub> <sup>(2)</sup>
02BE <sub>16</sub>	CAN1 Message Slot 14 Control Register	C1MCTL14	00 <sub>16</sub> <sup>(2)</sup>
02BF <sub>16</sub>	CAN1 Message Slot 15 Control Register	C1MCTL15	00 <sub>16</sub> <sup>(2)</sup>
02C0 <sub>16</sub> 02C1 <sub>16</sub>	X0 Register Y0 Register	X0R,Y0R	XX <sub>16</sub> XX <sub>16</sub>
02C2 <sub>16</sub> 02C3 <sub>16</sub>	X1 Register Y1 Register	X1R,Y1R	XX <sub>16</sub> XX <sub>16</sub>
02C4 <sub>16</sub> 02C5 <sub>16</sub>	X2 Register Y2 Register	X2R,Y2R	XX <sub>16</sub> XX <sub>16</sub>
02C6 <sub>16</sub> 02C7 <sub>16</sub>	X3 Register Y3 Register	X3R,Y3R	XX <sub>16</sub> XX <sub>16</sub>
02C8 <sub>16</sub> 02C9 <sub>16</sub>	X4 Register Y4 Register	X4R,Y4R	XX <sub>16</sub> XX <sub>16</sub>
02CA <sub>16</sub> 02CB <sub>16</sub>	X5 Register Y5 Register	X5R,Y5R	XX <sub>16</sub> XX <sub>16</sub>
02CC <sub>16</sub> 02CD <sub>16</sub>	X6 Register Y6 Register	X6R,Y6R	XX <sub>16</sub> XX <sub>16</sub>
02CE <sub>16</sub> 02CF <sub>16</sub>	X7 Register Y7 Register	X7R,Y7R	XX <sub>16</sub> XX <sub>16</sub>
02D0 <sub>16</sub> 02D1 <sub>16</sub>	X8 Register Y8 Register	X8R,Y8R	XX <sub>16</sub> XX <sub>16</sub>
02D2 <sub>16</sub> 02D3 <sub>16</sub>	X9 Register Y9 Register	X9R,Y9R	XX <sub>16</sub> XX <sub>16</sub>
02D4 <sub>16</sub> 02D5 <sub>16</sub>	X10 Register Y10 Register	X10R,Y10R	XX <sub>16</sub> XX <sub>16</sub>
02D6 <sub>16</sub> 02D7 <sub>16</sub>	X11 Register Y11 Register	X11R,Y11R	XX <sub>16</sub> XX <sub>16</sub>
02D8 <sub>16</sub> 02D9 <sub>16</sub>	X12 Register Y12 Register	X12R,Y12R	XX <sub>16</sub> XX <sub>16</sub>
02DA <sub>16</sub> 02DB <sub>16</sub>	X13 Register Y13 Register	X13R,Y13R	XX <sub>16</sub> XX <sub>16</sub>
02DC <sub>16</sub> 02DD <sub>16</sub>	X14 Register Y14 Register	X14R,Y14R	XX <sub>16</sub> XX <sub>16</sub>
02DE <sub>16</sub> 02DF <sub>16</sub>	X15 Register Y15 Register	X15R,Y15R	XX <sub>16</sub> XX <sub>16</sub>

(Note 1)

X: Indeterminate

Blank spaces are reserved. No access is allowed.

## NOTES:

1. The BANKSEL bit in the C1CTRL1 register switches functions for addresses 02A0<sub>16</sub> to 02BF<sub>16</sub>.
2. Values are obtained by setting the SLEEP bit in the C1SLPR register to "1" (sleep mode exited) after reset and supplying a clock to the CAN module.

Address	Register	Symbol	Value after RESET
02E0 <sub>16</sub>	X/Y Control Register	XYC	XXXX XX00 <sub>2</sub>
02E1 <sub>16</sub>			
02E2 <sub>16</sub>			
02E3 <sub>16</sub>			
02E4 <sub>16</sub>	UART1 Special Mode Register 4	U1SMR4	00 <sub>16</sub>
02E5 <sub>16</sub>	UART1 Special Mode Register 3	U1SMR3	00 <sub>16</sub>
02E6 <sub>16</sub>	UART1 Special Mode Register 2	U1SMR2	00 <sub>16</sub>
02E7 <sub>16</sub>	UART1 Special Mode Register	U1SMR	00 <sub>16</sub>
02E8 <sub>16</sub>	UART1 Transmit/Receive Mode Register	U1MR	00 <sub>16</sub>
02E9 <sub>16</sub>	UART1 Bit Rate Register	U1BRG	XX <sub>16</sub>
02EA <sub>16</sub>	UART1 Transmit Buffer Register	U1TB	XX <sub>16</sub>
02EB <sub>16</sub>			XX <sub>16</sub>
02EC <sub>16</sub>	UART1 Transmit/Receive Control Register 0	U1C0	0000 1000 <sub>2</sub>
02ED <sub>16</sub>	UART1 Transmit/Receive Control Register 1	U1C1	0000 0010 <sub>2</sub>
02EE <sub>16</sub>	UART1 Receive Buffer Register	U1RB	XX <sub>16</sub>
02EF <sub>16</sub>			XX <sub>16</sub>
02F0 <sub>16</sub>			
02F1 <sub>16</sub>			
02F2 <sub>16</sub>			
02F3 <sub>16</sub>			
02F4 <sub>16</sub>	UART4 Special Mode Register 4	U4SMR4	00 <sub>16</sub>
02F5 <sub>16</sub>	UART4 Special Mode Register 3	U4SMR3	00 <sub>16</sub>
02F6 <sub>16</sub>	UART4 Special Mode Register 2	U4SMR2	00 <sub>16</sub>
02F7 <sub>16</sub>	UART4 Special Mode Register	U4SMR	00 <sub>16</sub>
02F8 <sub>16</sub>	UART4 Transmit/Receive Mode Register	U4MR	00 <sub>16</sub>
02F9 <sub>16</sub>	UART4 Bit Rate Register	U4BRG	XX <sub>16</sub>
02FA <sub>16</sub>	UART4 Transmit Buffer Register	U4TB	XX <sub>16</sub>
02FB <sub>16</sub>			XX <sub>16</sub>
02FC <sub>16</sub>	UART4 Transmit/Receive Control Register 0	U4C0	0000 1000 <sub>2</sub>
02FD <sub>16</sub>	UART4 Transmit/Receive Control Register 1	U4C1	0000 0010 <sub>2</sub>
02FE <sub>16</sub>	UART4 Receive Buffer Register	U4RB	XX <sub>16</sub>
02FF <sub>16</sub>			XX <sub>16</sub>
0300 <sub>16</sub>	Timer B3, B4, B5 Count Start Flag	TBSR	000X XXXX <sub>2</sub>
0301 <sub>16</sub>			
0302 <sub>16</sub>	Timer A1-1 Register	TA11	XX <sub>16</sub>
0303 <sub>16</sub>			XX <sub>16</sub>
0304 <sub>16</sub>	Timer A2-1 Register	TA21	XX <sub>16</sub>
0305 <sub>16</sub>			XX <sub>16</sub>
0306 <sub>16</sub>	Timer A4-1 Register	TA41	XX <sub>16</sub>
0307 <sub>16</sub>			XX <sub>16</sub>
0308 <sub>16</sub>	Three-Phase PWM Control Register 0	INVC0	00 <sub>16</sub>
0309 <sub>16</sub>	Three-Phase PWM Control Register 1	INVC1	00 <sub>16</sub>
030A <sub>16</sub>	Three-Phase Output Buffer Register 0	IDB0	XX11 1111 <sub>2</sub>
030B <sub>16</sub>	Three-Phase Output Buffer Register 1	IDB1	XX11 1111 <sub>2</sub>
030C <sub>16</sub>	Dead Time Timer	DTT	XX <sub>16</sub>
030D <sub>16</sub>	Timer B2 Interrupt Generation Frequency Set Counter	ICTB2	XX <sub>16</sub>
030E <sub>16</sub>			
030F <sub>16</sub>			

X: Indeterminate

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Value after RESET
0310 <sub>16</sub>	Timer B3 Register	TB3	XX <sub>16</sub>
0311 <sub>16</sub>			XX <sub>16</sub>
0312 <sub>16</sub>	Timer B4 Register	TB4	XX <sub>16</sub>
0313 <sub>16</sub>			XX <sub>16</sub>
0314 <sub>16</sub>	Timer B5 Register	TB5	XX <sub>16</sub>
0315 <sub>16</sub>			XX <sub>16</sub>
0316 <sub>16</sub>			
0317 <sub>16</sub>			
0318 <sub>16</sub>			
0319 <sub>16</sub>			
031A <sub>16</sub>			
031B <sub>16</sub>	Timer B3 Mode Register	TB3MR	00XX 0000 <sub>2</sub>
031C <sub>16</sub>	Timer B4 Mode Register	TB4MR	00XX 0000 <sub>2</sub>
031D <sub>16</sub>	Timer B5 Mode Register	TB5MR	00XX 0000 <sub>2</sub>
031E <sub>16</sub>			
031F <sub>16</sub>	External Interrupt Request Source Select Register	IFSR	00 <sub>16</sub>
0320 <sub>16</sub>			
0321 <sub>16</sub>			
0322 <sub>16</sub>			
0323 <sub>16</sub>			
0324 <sub>16</sub>	UART3 Special Mode Register 4	U3SMR4	00 <sub>16</sub>
0325 <sub>16</sub>	UART3 Special Mode Register 3	U3SMR3	00 <sub>16</sub>
0326 <sub>16</sub>	UART3 Special Mode Register 2	U3SMR2	00 <sub>16</sub>
0327 <sub>16</sub>	UART3 Special Mode Register	U3SMR	00 <sub>16</sub>
0328 <sub>16</sub>	UART3 Transmit/Receive Mode Register	U3MR	00 <sub>16</sub>
0329 <sub>16</sub>	UART3 Bit Rate Register	U3BRG	XX <sub>16</sub>
032A <sub>16</sub>	UART3 Transmit Buffer Register	U3TB	XX <sub>16</sub>
032B <sub>16</sub>			XX <sub>16</sub>
032C <sub>16</sub>	UART3 Transmit/Receive Control Register 0	U3C0	0000 1000 <sub>2</sub>
032D <sub>16</sub>	UART3 Transmit/Receive Control Register 1	U3C1	0000 0010 <sub>2</sub>
032E <sub>16</sub>	UART3 Receive Buffer Register	U3RB	XX <sub>16</sub>
032F <sub>16</sub>			XX <sub>16</sub>
0330 <sub>16</sub>			
0331 <sub>16</sub>			
0332 <sub>16</sub>			
0333 <sub>16</sub>			
0334 <sub>16</sub>	UART2 Special Mode Register 4	U2SMR4	00 <sub>16</sub>
0335 <sub>16</sub>	UART2 Special Mode Register 3	U2SMR3	00 <sub>16</sub>
0336 <sub>16</sub>	UART2 Special Mode Register 2	U2SMR2	00 <sub>16</sub>
0337 <sub>16</sub>	UART2 Special Mode Register	U2SMR	00 <sub>16</sub>
0338 <sub>16</sub>	UART2 Transmit/Receive Mode Register	U2MR	00 <sub>16</sub>
0339 <sub>16</sub>	UART2 Bit Rate Register	U2BRG	XX <sub>16</sub>
033A <sub>16</sub>	UART2 Transmit Buffer Register	U2TB	XX <sub>16</sub>
033B <sub>16</sub>			XX <sub>16</sub>
033C <sub>16</sub>	UART2 Transmit/Receive Control Register 0	U2C0	0000 1000 <sub>2</sub>
033D <sub>16</sub>	UART2 Transmit/Receive Control Register 1	U2C1	0000 0010 <sub>2</sub>
033E <sub>16</sub>	UART2 Receive Buffer Register	U2RB	XX <sub>16</sub>
033F <sub>16</sub>			XX <sub>16</sub>

X: Indeterminate

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Value after RESET
0340 <sub>16</sub>	Count Start Flag	TABSR	00 <sub>16</sub>
0341 <sub>16</sub>	Clock Prescaler Reset Flag	CPSRF	0XXX XXXX <sub>2</sub>
0342 <sub>16</sub>	One-Shot Start Flag	ONSF	00 <sub>16</sub>
0343 <sub>16</sub>	Trigger Select Register	TRGSR	00 <sub>16</sub>
0344 <sub>16</sub>	Up/Down Flag	UDF	00 <sub>16</sub>
0345 <sub>16</sub>			
0346 <sub>16</sub> 0347 <sub>16</sub>	Timer A0 Register	TA0	XX <sub>16</sub> XX <sub>16</sub>
0348 <sub>16</sub> 0349 <sub>16</sub>	Timer A1 Register	TA1	XX <sub>16</sub> XX <sub>16</sub>
034A <sub>16</sub> 034B <sub>16</sub>	Timer A2 Register	TA2	XX <sub>16</sub> XX <sub>16</sub>
034C <sub>16</sub> 034D <sub>16</sub>	Timer A3 Register	TA3	XX <sub>16</sub> XX <sub>16</sub>
034E <sub>16</sub> 034F <sub>16</sub>	Timer A4 Register	TA4	XX <sub>16</sub> XX <sub>16</sub>
0350 <sub>16</sub> 0351 <sub>16</sub>	Timer B0 Register	TB0	XX <sub>16</sub> XX <sub>16</sub>
0352 <sub>16</sub> 0353 <sub>16</sub>	Timer B1 Register	TB1	XX <sub>16</sub> XX <sub>16</sub>
0354 <sub>16</sub> 0355 <sub>16</sub>	Timer B2 Register	TB2	XX <sub>16</sub> XX <sub>16</sub>
0356 <sub>16</sub>	Timer A0 Mode Register	TA0MR	00 <sub>16</sub>
0357 <sub>16</sub>	Timer A1 Mode Register	TA1MR	00 <sub>16</sub>
0358 <sub>16</sub>	Timer A2 Mode Register	TA2MR	00 <sub>16</sub>
0359 <sub>16</sub>	Timer A3 Mode Register	TA3MR	00 <sub>16</sub>
035A <sub>16</sub>	Timer A4 Mode Register	TA4MR	00 <sub>16</sub>
035B <sub>16</sub>	Timer B0 Mode Register	TB0MR	00XX 0000 <sub>2</sub>
035C <sub>16</sub>	Timer B1 Mode Register	TB1MR	00XX 0000 <sub>2</sub>
035D <sub>16</sub>	Timer B2 Mode Register	TB2MR	00XX 0000 <sub>2</sub>
035E <sub>16</sub>	Timer B2 Special Mode Register	TB2SC	XXXX XXX0 <sub>2</sub>
035F <sub>16</sub>	Count Source Prescaler Register <sup>(1)</sup>	TCSPR	0XXX 0000 <sub>2</sub>
0360 <sub>16</sub>			
0361 <sub>16</sub>			
0362 <sub>16</sub>			
0363 <sub>16</sub>			
0364 <sub>16</sub>	UART0 Special Mode Register 4	U0SMR4	00 <sub>16</sub>
0365 <sub>16</sub>	UART0 Special Mode Register 3	U0SMR3	00 <sub>16</sub>
0366 <sub>16</sub>	UART0 Special Mode Register 2	U0SMR2	00 <sub>16</sub>
0367 <sub>16</sub>	UART0 Special Mode Register	U0SMR	00 <sub>16</sub>
0368 <sub>16</sub>	UART0 Transmit/Receive Mode Register	U0MR	00 <sub>16</sub>
0369 <sub>16</sub>	UART0 Bit Rate Register	U0BRG	XX <sub>16</sub>
036A <sub>16</sub> 036B <sub>16</sub>	UART0 Transmit Buffer Register	U0TB	XX <sub>16</sub> XX <sub>16</sub>
036C <sub>16</sub>	UART0 Transmit/Receive Control Register 0	U0C0	0000 1000 <sub>2</sub>
036D <sub>16</sub>	UART0 Transmit/Receive Control Register 1	U0C1	0000 0010 <sub>2</sub>
036E <sub>16</sub> 036F <sub>16</sub>	UART0 Receive Buffer Register	U0RB	XX <sub>16</sub> XX <sub>16</sub>

X: Indeterminate

Blank spaces are reserved. No access is allowed.

NOTE:

1. The TCSPR register maintains values set before reset, even after software reset or watchdog timer reset has been performed.

Address	Register	Symbol	Value after RESET
0370 <sub>16</sub>			
0371 <sub>16</sub>			
0372 <sub>16</sub>			
0373 <sub>16</sub>			
0374 <sub>16</sub>			
0375 <sub>16</sub>			
0376 <sub>16</sub>			
0377 <sub>16</sub>			
0378 <sub>16</sub>	DMA0 Request Source Select Register	DM0SL	0X00 0000 <sub>2</sub>
0379 <sub>16</sub>	DMA1 Request Source Select Register	DM1SL	0X00 0000 <sub>2</sub>
037A <sub>16</sub>	DMA2 Request Source Select Register	DM2SL	0X00 0000 <sub>2</sub>
037B <sub>16</sub>	DMA3 Request Source Select Register	DM3SL	0X00 0000 <sub>2</sub>
037C <sub>16</sub>	CRC Data Register	CRCD	XX <sub>16</sub>
037D <sub>16</sub>			XX <sub>16</sub>
037E <sub>16</sub>	CRC Input Register	CRCIN	XX <sub>16</sub>
037F <sub>16</sub>			
0380 <sub>16</sub>	A/D0 Register 0	AD00	XXXX XXXX <sub>2</sub>
0381 <sub>16</sub>			0000 0000 <sub>2</sub>
0382 <sub>16</sub>	A/D0 Register 1	AD01	XX <sub>16</sub>
0383 <sub>16</sub>			XX <sub>16</sub>
0384 <sub>16</sub>	A/D0 Register 2	AD02	XX <sub>16</sub>
0385 <sub>16</sub>			XX <sub>16</sub>
0386 <sub>16</sub>	A/D0 Register 3	AD03	XX <sub>16</sub>
0387 <sub>16</sub>			XX <sub>16</sub>
0388 <sub>16</sub>	A/D0 Register 4	AD04	XX <sub>16</sub>
0389 <sub>16</sub>			XX <sub>16</sub>
038A <sub>16</sub>	A/D0 Register 5	AD05	XX <sub>16</sub>
038B <sub>16</sub>			XX <sub>16</sub>
038C <sub>16</sub>	A/D0 Register 6	AD06	XX <sub>16</sub>
038D <sub>16</sub>			XX <sub>16</sub>
038E <sub>16</sub>	A/D0 Register 7	AD07	XX <sub>16</sub>
038F <sub>16</sub>			XX <sub>16</sub>
0390 <sub>16</sub>			
0391 <sub>16</sub>			
0392 <sub>16</sub>	A/D0 Control Register 4	AD0CON4	XXXX 00XX <sub>2</sub>
0393 <sub>16</sub>			
0394 <sub>16</sub>	A/D0 Control Register 2	AD0CON2	XX0X X000 <sub>2</sub>
0395 <sub>16</sub>	A/D0 Control Register 3	AD0CON3	XXXX X000 <sub>2</sub>
0396 <sub>16</sub>	A/D0 Control Register 0	AD0CON0	00 <sub>16</sub>
0397 <sub>16</sub>	A/D0 Control Register 1	AD0CON1	00 <sub>16</sub>
0398 <sub>16</sub>	D/A Register 0	DA0	XX <sub>16</sub>
0399 <sub>16</sub>			
039A <sub>16</sub>	D/A Register 1	DA1	XX <sub>16</sub>
039B <sub>16</sub>			
039C <sub>16</sub>	D/A Control Register	DACON	XXXX XX00 <sub>2</sub>
039D <sub>16</sub>			
039E <sub>16</sub>			
039F <sub>16</sub>			

X: Indeterminate

Blank spaces are reserved. No access is allowed.

&lt;144-pin package&gt;

Address	Register	Symbol	Value after RESET
03A0 <sub>16</sub>	Function Select Register A8	PS8	X000 0000 <sub>2</sub>
03A1 <sub>16</sub>	Function Select Register A9	PS9	00 <sub>16</sub>
03A2 <sub>16</sub>			
03A3 <sub>16</sub>			
03A4 <sub>16</sub>			
03A5 <sub>16</sub>			
03A6 <sub>16</sub>			
03A7 <sub>16</sub>	Function Select Register D1	PSD1	X0XX XX00 <sub>2</sub>
03A8 <sub>16</sub>			
03A9 <sub>16</sub>			
03AA <sub>16</sub>			
03AB <sub>16</sub>			
03AC <sub>16</sub>	Function Select Register C2	PSC2	XXXX X00X <sub>2</sub>
03AD <sub>16</sub>	Function Select Register C3	PSC3	X0XX XXXX <sub>2</sub>
03AE <sub>16</sub>			
03AF <sub>16</sub>	Function Select Register C	PSC	00X0 0000 <sub>2</sub>
03B0 <sub>16</sub>	Function Select Register A0	PS0	00 <sub>16</sub>
03B1 <sub>16</sub>	Function Select Register A1	PS1	00 <sub>16</sub>
03B2 <sub>16</sub>	Function Select Register B0	PSL0	00 <sub>16</sub>
03B3 <sub>16</sub>	Function Select Register B1	PSL1	00 <sub>16</sub>
03B4 <sub>16</sub>	Function Select Register A2	PS2	00X0 0000 <sub>2</sub>
03B5 <sub>16</sub>	Function Select Register A3	PS3	00 <sub>16</sub>
03B6 <sub>16</sub>	Function Select Register B2	PSL2	00X0 0000 <sub>2</sub>
03B7 <sub>16</sub>	Function Select Register B3	PSL3	00 <sub>16</sub>
03B8 <sub>16</sub>			
03B9 <sub>16</sub>	Function Select Register A5	PS5	XXX0 0000 <sub>2</sub>
03BA <sub>16</sub>			
03BB <sub>16</sub>			
03BC <sub>16</sub>			
03BD <sub>16</sub>			
03BE <sub>16</sub>			
03BF <sub>16</sub>			
03C0 <sub>16</sub>	Port P6 Register	P6	XX <sub>16</sub>
03C1 <sub>16</sub>	Port P7 Register	P7	XX <sub>16</sub>
03C2 <sub>16</sub>	Port P6 Direction Register	PD6	00 <sub>16</sub>
03C3 <sub>16</sub>	Port P7 Direction Register	PD7	00 <sub>16</sub>
03C4 <sub>16</sub>	Port P8 Register	P8	XX <sub>16</sub>
03C5 <sub>16</sub>	Port P9 Register	P9	XX <sub>16</sub>
03C6 <sub>16</sub>	Port P8 Direction Register	PD8	00X0 0000 <sub>2</sub>
03C7 <sub>16</sub>	Port P9 Direction Register	PD9	00 <sub>16</sub>
03C8 <sub>16</sub>	Port P10 Register	P10	XX <sub>16</sub>
03C9 <sub>16</sub>	Port P11 Register	P11	XX <sub>16</sub>
03CA <sub>16</sub>	Port P10 Direction Register	PD10	00 <sub>16</sub>
03CB <sub>16</sub>	Port P11 Direction Register	PD11	XXX0 0000 <sub>2</sub>
03CC <sub>16</sub>	Port P12 Register	P12	XX <sub>16</sub>
03CD <sub>16</sub>	Port P13 Register	P13	XX <sub>16</sub>
03CE <sub>16</sub>	Port P12 Direction Register	PD12	00 <sub>16</sub>
03CF <sub>16</sub>	Port P13 Direction Register	PD13	00 <sub>16</sub>

X: Indeterminate

Blank spaces are reserved. No access is allowed.

&lt;144-pin package&gt;

Address	Register	Symbol	Value after RESET
03D0 <sub>16</sub>	Port P14 Register	P14	XX <sub>16</sub>
03D1 <sub>16</sub>	Port P15 Register	P15	XX <sub>16</sub>
03D2 <sub>16</sub>	Port P14 Direction Register	PD14	X000 0000 <sub>2</sub>
03D3 <sub>16</sub>	Port P15 Direction Register	PD15	00 <sub>16</sub>
03D4 <sub>16</sub>			
03D5 <sub>16</sub>			
03D6 <sub>16</sub>			
03D7 <sub>16</sub>			
03D8 <sub>16</sub>			
03D9 <sub>16</sub>			
03DA <sub>16</sub>	Pull-Up Control Register 2	PUR2	00 <sub>16</sub>
03DB <sub>16</sub>	Pull-Up Control Register 3	PUR3	00 <sub>16</sub>
03DC <sub>16</sub>	Pull-Up Control Register 4	PUR4	XXXX 0000 <sub>2</sub>
03DD <sub>16</sub>			
03DE <sub>16</sub>			
03DF <sub>16</sub>			
03E0 <sub>16</sub>	Port P0 Register	P0	XX <sub>16</sub>
03E1 <sub>16</sub>	Port P1 Register	P1	XX <sub>16</sub>
03E2 <sub>16</sub>	Port P0 Direction Register	PD0	00 <sub>16</sub>
03E3 <sub>16</sub>	Port P1 Direction Register	PD1	00 <sub>16</sub>
03E4 <sub>16</sub>	Port P2 Register	P2	XX <sub>16</sub>
03E5 <sub>16</sub>	Port P3 Register	P3	XX <sub>16</sub>
03E6 <sub>16</sub>	Port P2 Direction Register	PD2	00 <sub>16</sub>
03E7 <sub>16</sub>	Port P3 Direction Register	PD3	00 <sub>16</sub>
03E8 <sub>16</sub>	Port P4 Register	P4	XX <sub>16</sub>
03E9 <sub>16</sub>	Port P5 Register	P5	XX <sub>16</sub>
03EA <sub>16</sub>	Port P4 Direction Register	PD4	00 <sub>16</sub>
03EB <sub>16</sub>	Port P5 Direction Register	PD5	00 <sub>16</sub>
03EC <sub>16</sub>			
03ED <sub>16</sub>			
03EE <sub>16</sub>			
03EF <sub>16</sub>			
03F0 <sub>16</sub>	Pull-Up Control Register 0	PUR0	00 <sub>16</sub>
03F1 <sub>16</sub>	Pull-Up Control Register 1	PUR1	XXXX 0000 <sub>2</sub>
03F2 <sub>16</sub>			
03F3 <sub>16</sub>			
03F4 <sub>16</sub>			
03F5 <sub>16</sub>			
03F6 <sub>16</sub>			
03F7 <sub>16</sub>			
03F8 <sub>16</sub>			
03F9 <sub>16</sub>			
03FA <sub>16</sub>			
03FB <sub>16</sub>			
03FC <sub>16</sub>			
03FD <sub>16</sub>			
03FE <sub>16</sub>			
03FF <sub>16</sub>	Port Control Register	PCR	XXXX XXX0 <sub>2</sub>

X: Indeterminate

Blank spaces are reserved. No access is allowed.

&lt;100-pin package&gt;

Address	Register	Symbol	Value after RESET
03A0 <sub>16</sub>			
03A1 <sub>16</sub>			
03A2 <sub>16</sub>			
03A3 <sub>16</sub>			
03A4 <sub>16</sub>			
03A5 <sub>16</sub>			
03A6 <sub>16</sub>			
03A7 <sub>16</sub>	Function Select Register D1	PSD1	X0XX XX00 <sub>2</sub>
03A8 <sub>16</sub>			
03A9 <sub>16</sub>			
03AA <sub>16</sub>			
03AB <sub>16</sub>			
03AC <sub>16</sub>	Function Select Register C2	PSC2	XXXX X00X <sub>2</sub>
03AD <sub>16</sub>	Function Select Register C3	PSC3	X0XX XXXX <sub>2</sub>
03AE <sub>16</sub>			
03AF <sub>16</sub>	Function Select Register C	PSC	00X0 0000 <sub>2</sub>
03B0 <sub>16</sub>	Function Select Register A0	PS0	00 <sub>16</sub>
03B1 <sub>16</sub>	Function Select Register A1	PS1	00 <sub>16</sub>
03B2 <sub>16</sub>	Function Select Register B0	PSL0	00 <sub>16</sub>
03B3 <sub>16</sub>	Function Select Register B1	PSL1	00 <sub>16</sub>
03B4 <sub>16</sub>	Function Select Register A2	PS2	00X0 0000 <sub>2</sub>
03B5 <sub>16</sub>	Function Select Register A3	PS3	00 <sub>16</sub>
03B6 <sub>16</sub>	Function Select Register B2	PSL2	00X0 0000 <sub>2</sub>
03B7 <sub>16</sub>	Function Select Register B3	PSL3	00 <sub>16</sub>
03B8 <sub>16</sub>			
03B9 <sub>16</sub>			
03BA <sub>16</sub>			
03BB <sub>16</sub>			
03BC <sub>16</sub>			
03BD <sub>16</sub>			
03BE <sub>16</sub>			
03BF <sub>16</sub>			
03C0 <sub>16</sub>	Port P6 Register	P6	XX <sub>16</sub>
03C1 <sub>16</sub>	Port P7 Register	P7	XX <sub>16</sub>
03C2 <sub>16</sub>	Port P6 Direction Register	PD6	00 <sub>16</sub>
03C3 <sub>16</sub>	Port P7 Direction Register	PD7	00 <sub>16</sub>
03C4 <sub>16</sub>	Port P8 Register	P8	XX <sub>16</sub>
03C5 <sub>16</sub>	Port P9 Register	P9	XX <sub>16</sub>
03C6 <sub>16</sub>	Port P8 Direction Register	PD8	00X0 0000 <sub>2</sub>
03C7 <sub>16</sub>	Port P9 Direction Register	PD9	00 <sub>16</sub>
03C8 <sub>16</sub>	Port P10 Register	P10	XX <sub>16</sub>
03C9 <sub>16</sub>			
03CA <sub>16</sub>	Port P10 Direction Register	PD10	00 <sub>16</sub>
03CB <sub>16</sub>	Set default value to "FF <sub>16</sub> "		
03CC <sub>16</sub>			
03CD <sub>16</sub>			
03CE <sub>16</sub>	Set default value to "FF <sub>16</sub> "		
03CF <sub>16</sub>	Set default value to "FF <sub>16</sub> "		

X: Indeterminate

Blank spaces are reserved. No access is allowed.

&lt;100-pin package&gt;

Address	Register	Symbol	Value after RESET
03D0 <sub>16</sub>			
03D1 <sub>16</sub>			
03D2 <sub>16</sub>	Set default value to "FF <sub>16</sub> "		
03D3 <sub>16</sub>	Set default value to "FF <sub>16</sub> "		
03D4 <sub>16</sub>			
03D5 <sub>16</sub>			
03D6 <sub>16</sub>			
03D7 <sub>16</sub>			
03D8 <sub>16</sub>			
03D9 <sub>16</sub>			
03DA <sub>16</sub>	Pull-Up Control Register 2	PUR2	00 <sub>16</sub>
03DB <sub>16</sub>	Pull-Up Control Register 3	PUR3	00 <sub>16</sub>
03DC <sub>16</sub>	Set default value to "00 <sub>16</sub> "		
03DD <sub>16</sub>			
03DE <sub>16</sub>			
03DF <sub>16</sub>			
03E0 <sub>16</sub>	Port P0 Register	P0	XX <sub>16</sub>
03E1 <sub>16</sub>	Port P1 Register	P1	XX <sub>16</sub>
03E2 <sub>16</sub>	Port P0 Direction Register	PD0	00 <sub>16</sub>
03E3 <sub>16</sub>	Port P1 Direction Register	PD1	00 <sub>16</sub>
03E4 <sub>16</sub>	Port P2 Register	P2	XX <sub>16</sub>
03E5 <sub>16</sub>	Port P3 Register	P3	XX <sub>16</sub>
03E6 <sub>16</sub>	Port P2 Direction Register	PD2	00 <sub>16</sub>
03E7 <sub>16</sub>	Port P3 Direction Register	PD3	00 <sub>16</sub>
03E8 <sub>16</sub>	Port P4 Register	P4	XX <sub>16</sub>
03E9 <sub>16</sub>	Port P5 Register	P5	XX <sub>16</sub>
03EA <sub>16</sub>	Port P4 Direction Register	PD4	00 <sub>16</sub>
03EB <sub>16</sub>	Port P5 Direction Register	PD5	00 <sub>16</sub>
03EC <sub>16</sub>			
03ED <sub>16</sub>			
03EE <sub>16</sub>			
03EF <sub>16</sub>			
03F0 <sub>16</sub>	Pull-up Control Register 0	PUR0	00 <sub>16</sub>
03F1 <sub>16</sub>	Pull-up Control Register 1	PUR1	XXXX 0000 <sub>2</sub>
03F2 <sub>16</sub>			
03F3 <sub>16</sub>			
03F4 <sub>16</sub>			
03F5 <sub>16</sub>			
03F6 <sub>16</sub>			
03F7 <sub>16</sub>			
03F8 <sub>16</sub>			
03F9 <sub>16</sub>			
03FA <sub>16</sub>			
03FB <sub>16</sub>			
03FC <sub>16</sub>			
03FD <sub>16</sub>			
03FE <sub>16</sub>			
03FF <sub>16</sub>	Port Control Register	PCR	XXXX XXX0 <sub>2</sub>

X: Indeterminate

Blank spaces are reserved. No access is allowed.

## 5. Reset

Hardware reset 1, software reset, and watchdog timer reset are available to reset the microcomputer.

### 5.1 Hardware Reset 1

Pins, the CPU and SFRs are reset by setting the  $\overline{\text{RESET}}$  pin. If the supply voltage meets the recommended operating conditions, all pins are reset and become input ports<sup>(1)</sup> when a low-level ("L") signal is applied to the  $\overline{\text{RESET}}$  pin. The oscillation circuit is also reset and the main clock starts oscillating. The CPU and SFRs are reset when a signal applied to the  $\overline{\text{RESET}}$  pin changes "L" to high ("H"). The microcomputer executes the program in an address indicated by the reset vector. The internal RAM is not reset. When an "L" signal is applied to the  $\overline{\text{RESET}}$  pin while writing data to the internal RAM, the internal RAM is in an indeterminate state.

Figure 5.1 shows an example of the reset circuit. Figure 5.2 shows a reset sequence.

#### NOTE:

- Whether ports are pulled up or not is indeterminate until internal supply voltage stabilizes.

#### 5.1.1 Reset on a Stable Supply Voltage

- Apply an "L" signal to the  $\overline{\text{RESET}}$  pin
- Provide 20 or more clock cycle inputs into the XIN pin
- Apply an "H" signal to the  $\overline{\text{RESET}}$  pin

#### 5.1.2 Power-on Reset

- Apply an "L" signal to the  $\overline{\text{RESET}}$  pin
- Raise the supply voltage to the recommended operating level
- Wait for  $t_d(P-R)$  ms to allow the internal voltage to stabilize
- Provide 20 or more clock cycle inputs into the XIN pin
- Apply an "H" signal to the  $\overline{\text{RESET}}$  pin

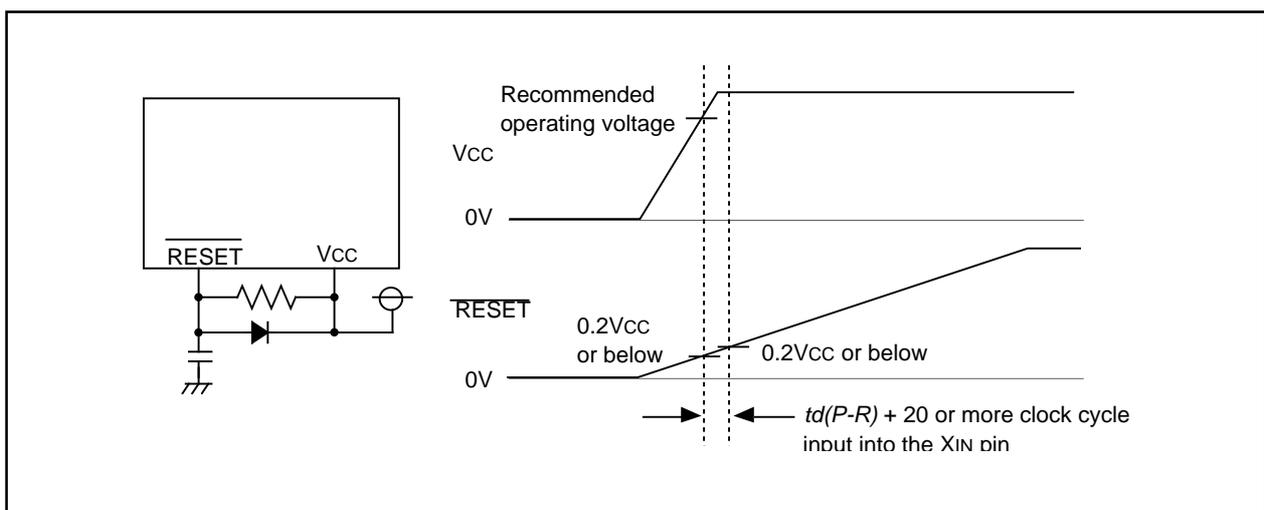


Figure 5.1 Reset Circuit

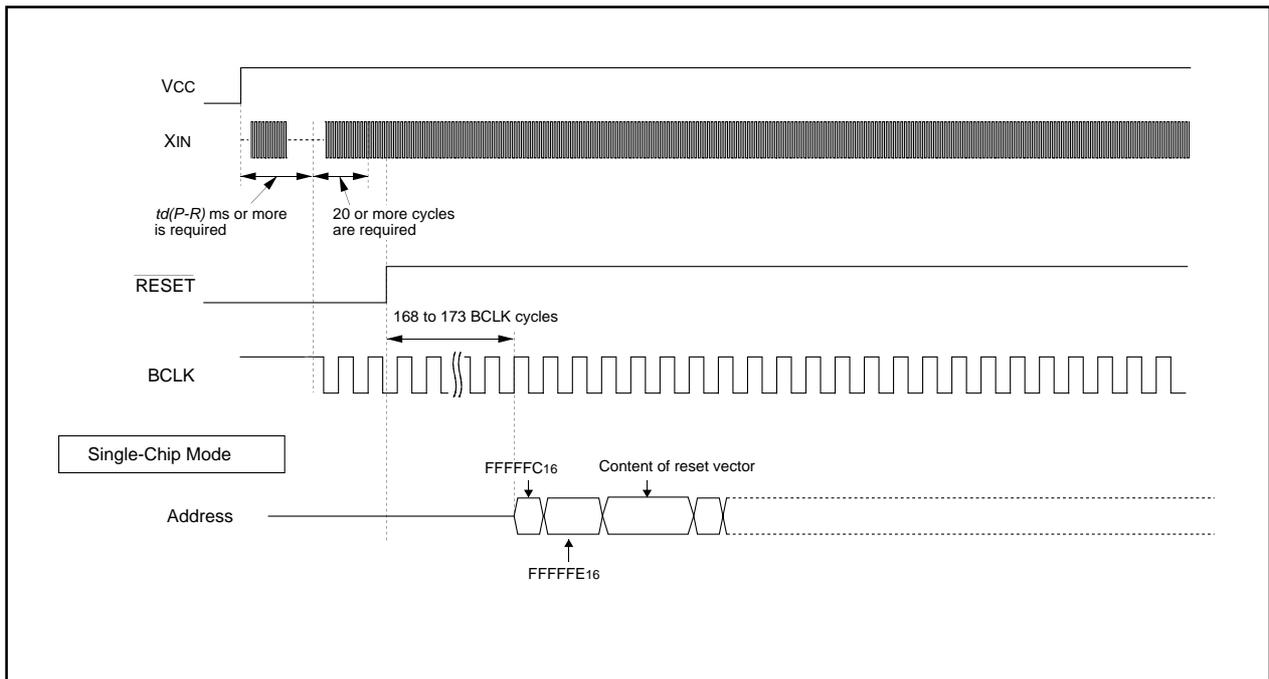


Figure 5.2 Reset Sequence

### 5.2 Software Reset

Pins, the CPU and SFRs are reset when the PM03 bit in the PM0 register is set to "1" (microcomputer reset). Then the microcomputer executes the program in an address determined by the reset vector. Set the PM03 bit to "1" while the main clock is selected as the CPU clock and the main clock oscillation is stable.

In the software reset, the microcomputer does not reset a part of SFRs. Refer to **4. Special Function Registers (SFRs)** for details. Processor mode remains unchanged since the PM01 and PM00 bits in the PM0 register are not reset.

### 5.3 Watchdog Timer Reset

Pins, the CPU and SFRs are reset when the CM06 bit in the CM0 register is set to "1" (reset) and the watchdog timer underflows. Then the microcomputer executes the program in an address determined by the reset vector.

In the watchdog timer reset, the microcomputer does not reset a part of SFRs. Refer to **4. Special Function Registers (SFRs)** for details. Processor mode remains unchanged since the PM01 and PM00 bits in the PM0 register are not reset.

### 5.4 Internal Space

Figure 5.3 shows CPU register states after reset. Refer to **4. Special Function Registers (SFRs)** for SFR states after reset.

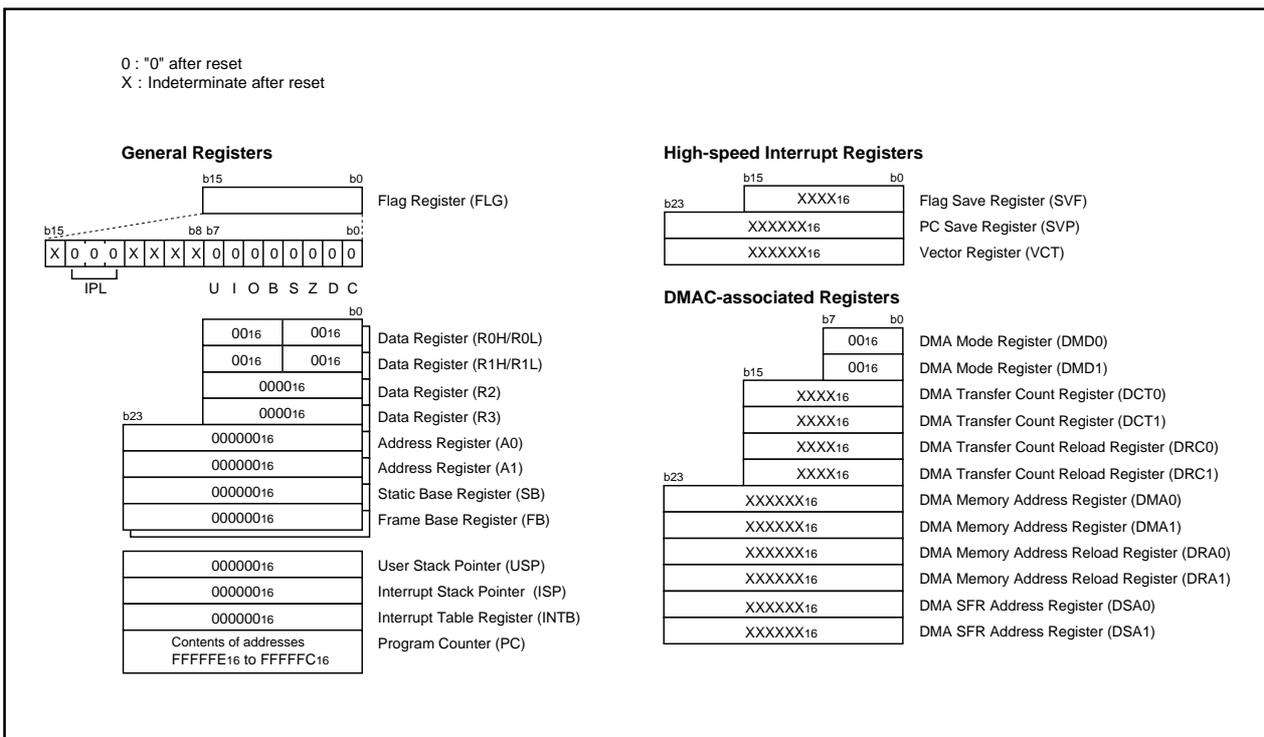
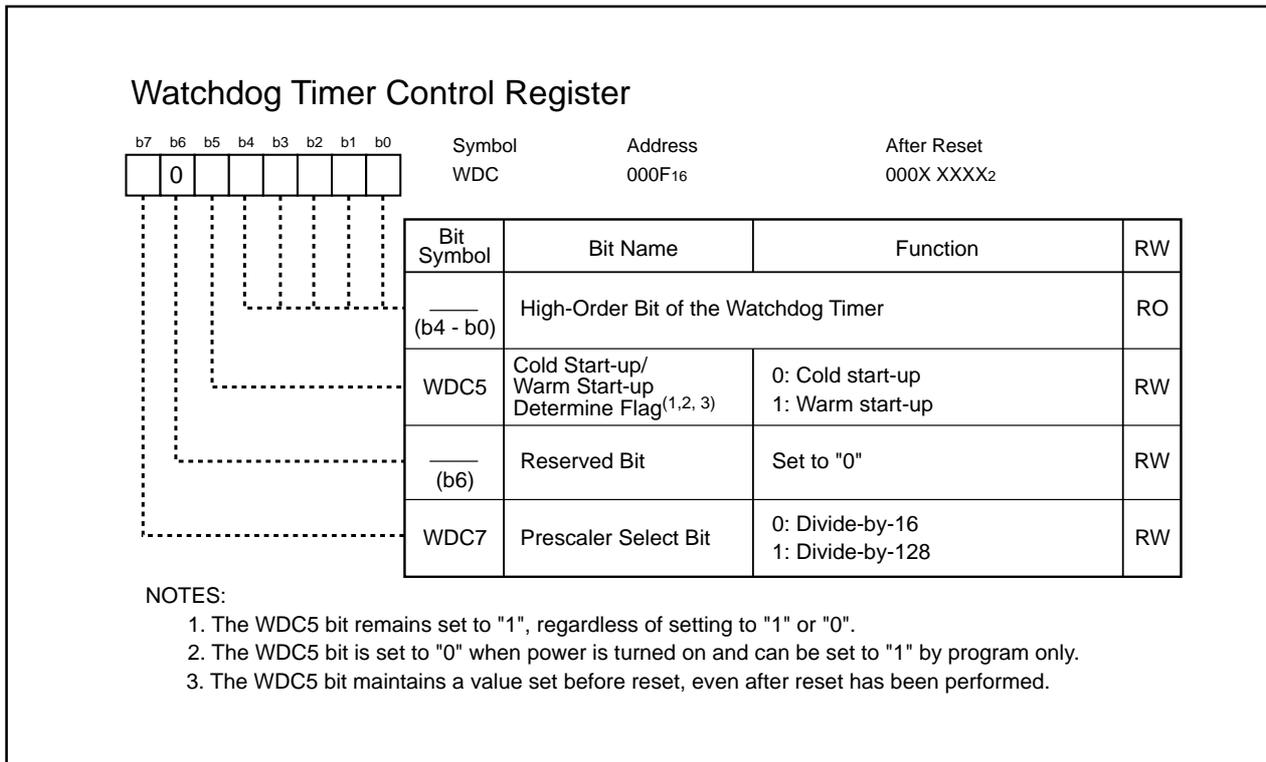


Figure 5.3 CPU Register States after Reset

## 6. Cold Start-up/Warm Start-up Determine Function

The WDC5 bit in the WDC register determines either cold start-up, power-on reset, or warm start-up, reset during the microcomputer running. Default value of the WDC5 bit is "0" (cold start-up) when power-on. It is set to "1" (warm start-up) by writing desired values to the WDC register. The WDC5 bit is not reset, regardless of a software reset or reset signal input.

Figure 6.1 shows the WDC register. Figure 6.2 shows a block diagram of the cold start-up/warm start-up determine function. Figure 6.3 shows its operation example.



**Figure 6.1 WDC Register**

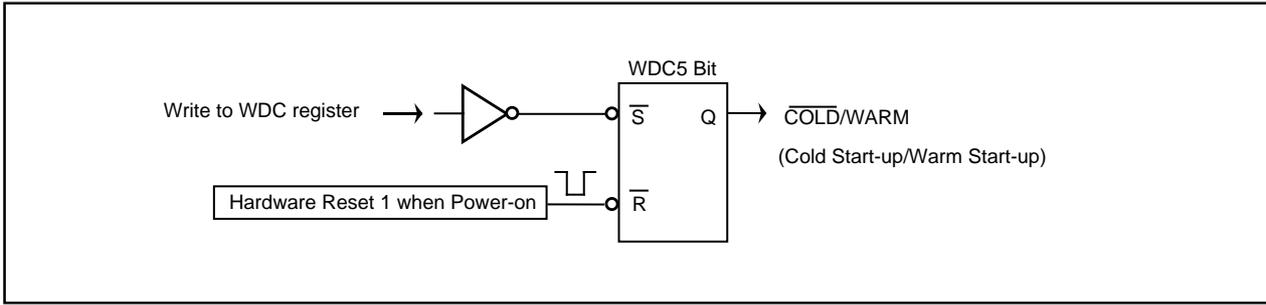


Figure 6.2 Cold Start-up/Warm Start-up Determine Function Block Diagram

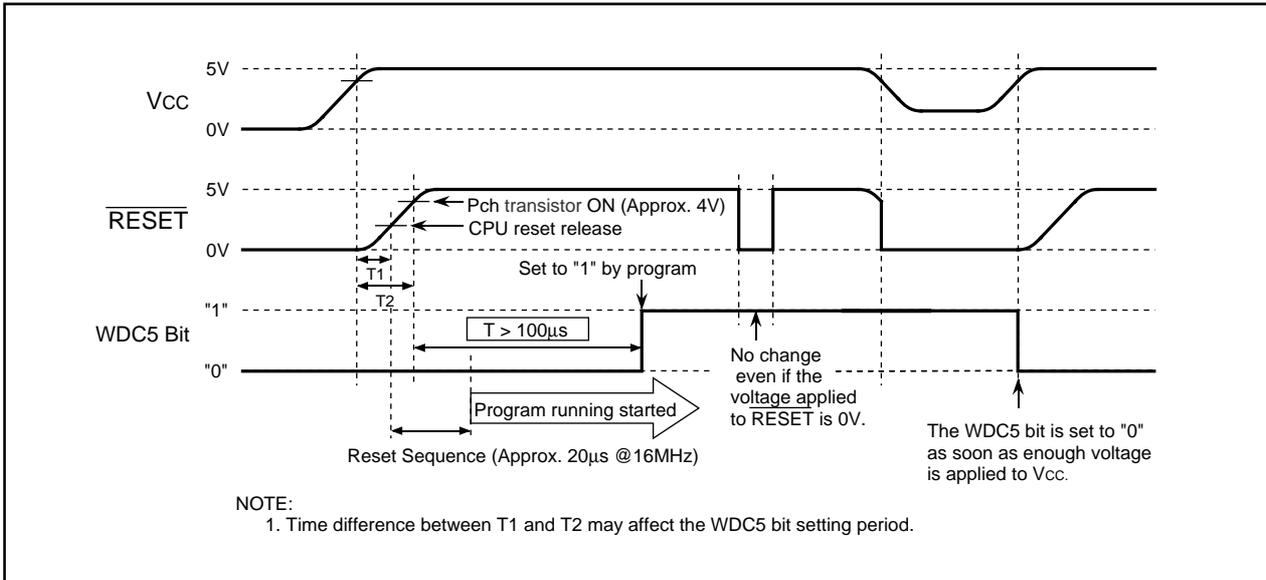


Figure 6.3 Cold Start-up/Warm Start-up Determine Function Operation

## 7. Processor Mode

### NOTE

Use M32C/88T in single-chip mode only.  
M32C/88T cannot be used in memory expansion mode and microprocessor mode.

### 7.1 Types of Processor Mode

Only single-chip mode can be selected as a processor mode. SFRs, internal RAM, and internal ROM can be accessed. All pins are assigned to input/output ports or peripheral function input/output ports.

### 7.2 Setting of Processor Mode

The CNVss pin and the PM01 and PM00 bits in the PM0 register determine which processor mode is selected. Apply an low-level ("L") signal to the CNVss pin. Set the PM01 and PM00 bits to "002" (single-chip mode).

Figures 7.1 and 7.2 show the PM0 register and PM1 register. Figure 7.3 shows a memory map in single-chip mode.

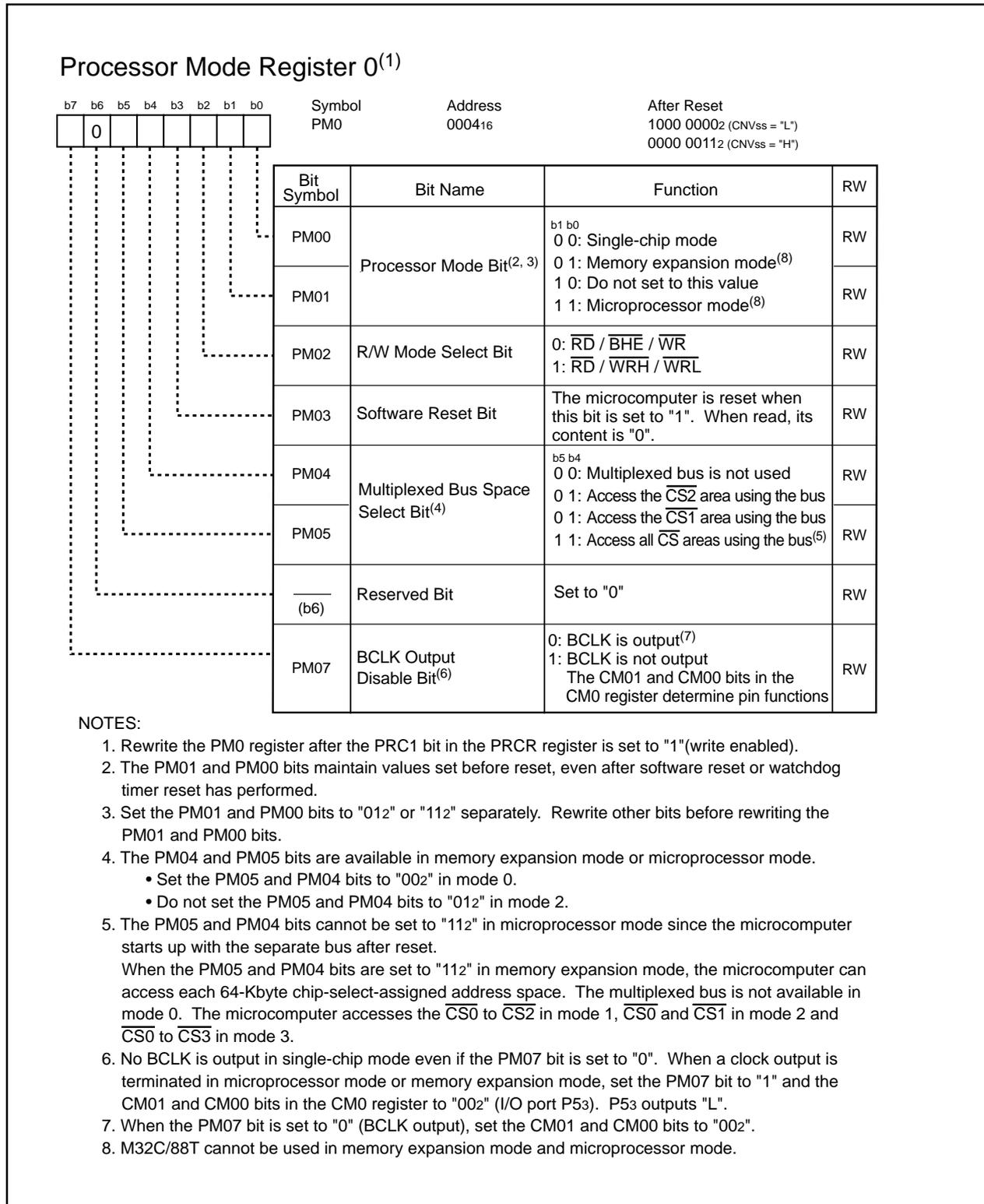


Figure 7.1 PM0 Register

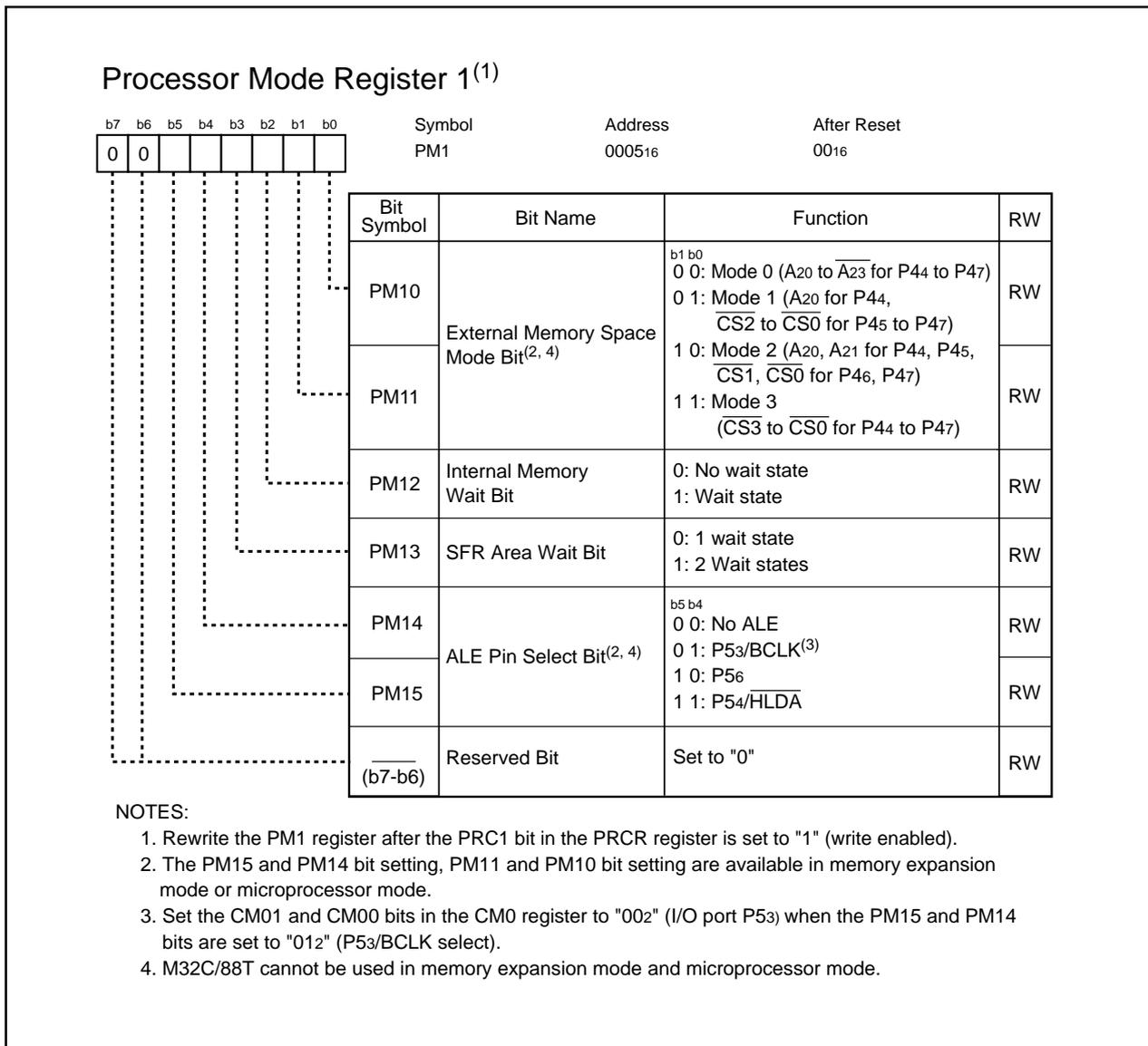


Figure 7.2 PM1 Register



## 8. Clock Generation Circuit

### 8.1 Types of the Clock Generation Circuit

Four circuits are included to generate the system clock signal:

- Main clock oscillation circuit
- Sub clock oscillation circuit
- On-chip oscillator
- PLL frequency synthesizer

Table 8.1 lists specifications of the clock generation circuit. Figure 8.1 shows a block diagram of the clock generation circuit. Figures 8.2 to 8.8 show registers controlling the clock.

**Table 8.1 Clock Generation Circuit Specification**

Item	Main Clock Oscillation Circuit	Sub Clock Oscillation Circuit	On-chip Oscillator	PLL Frequency Synthesizer
Use	CPU clock source, Peripheral function clock source	CPU clock source, Timer A and B clock source	CPU clock source, Peripheral function clock source	CPU clock source, Peripheral function clock source
Clock Frequency	Up to 32 MHz	32.768 kHz	Approx. 1 MHz	Up to 32 MHz (See Table 8.3)
Connectable Oscillator or Additional Circuit	Ceramic resonator Crystal oscillator	Crystal oscillator	---	---
Pins for Oscillator or for Additional Circuit	XIN, XOUT	XCIN, XCOUT	---	---
Oscillation Stop/Restart Function	Available	Available	Available	Available
Oscillator State after Reset	Oscillating	Stopped	Stopped	Stopped
Other	Externally generated clock can be applied.	Externally generated clock can be applied.	When the main clock stops oscillating, the on-chip oscillator starts oscillating automatically and becomes clock source for the CPU and peripheral function.	---

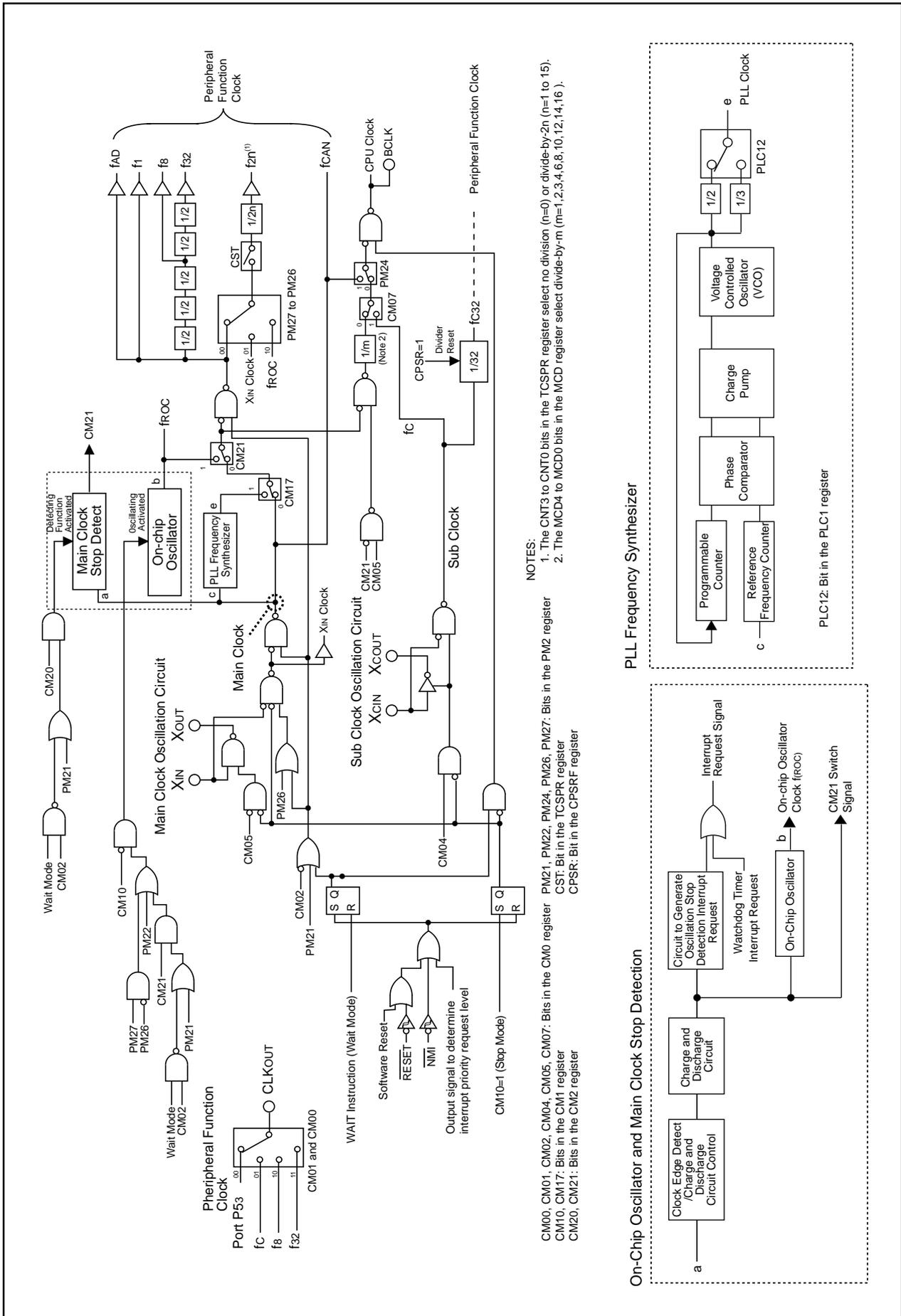


Figure 8.1 Clock Generation Circuit

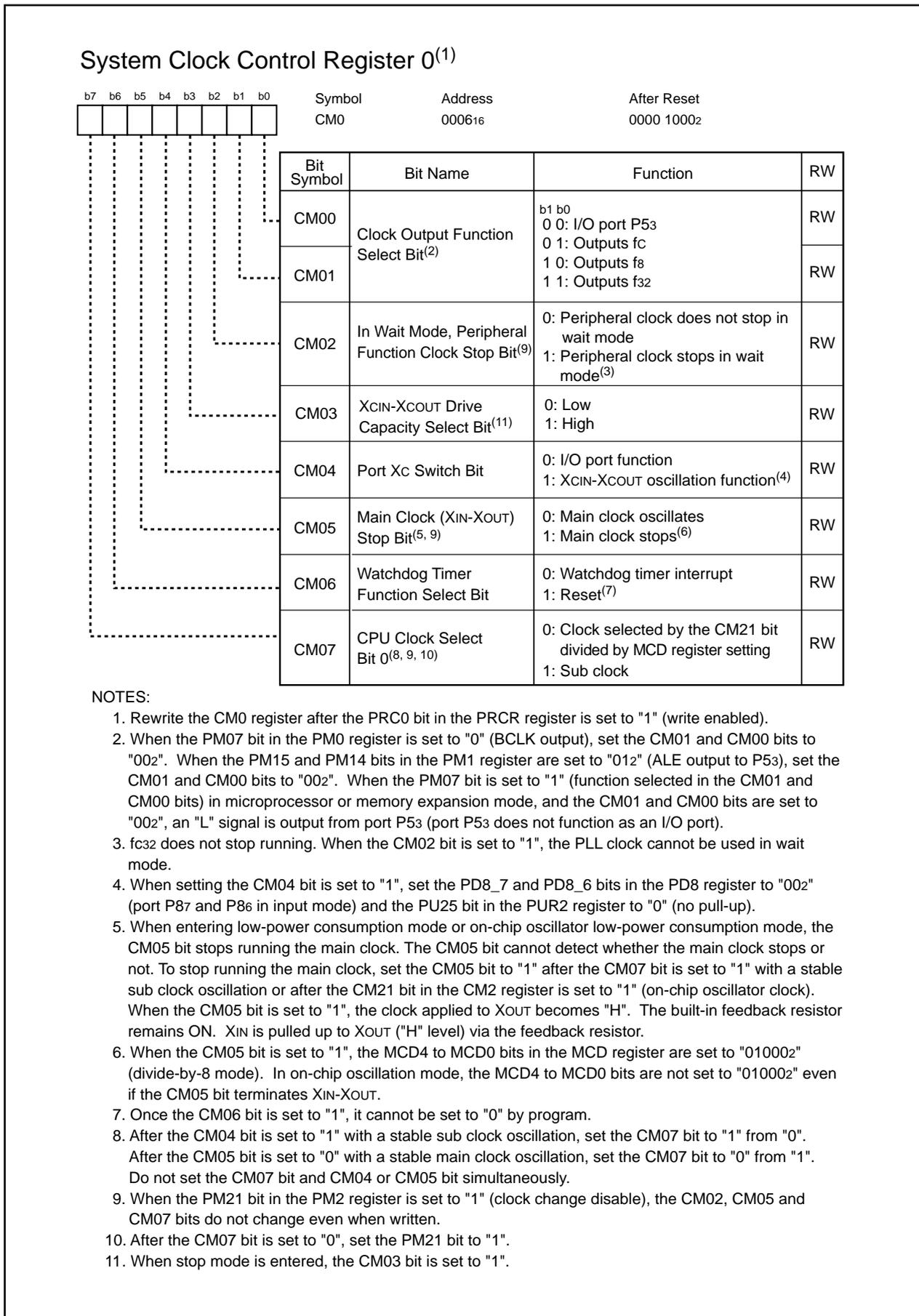
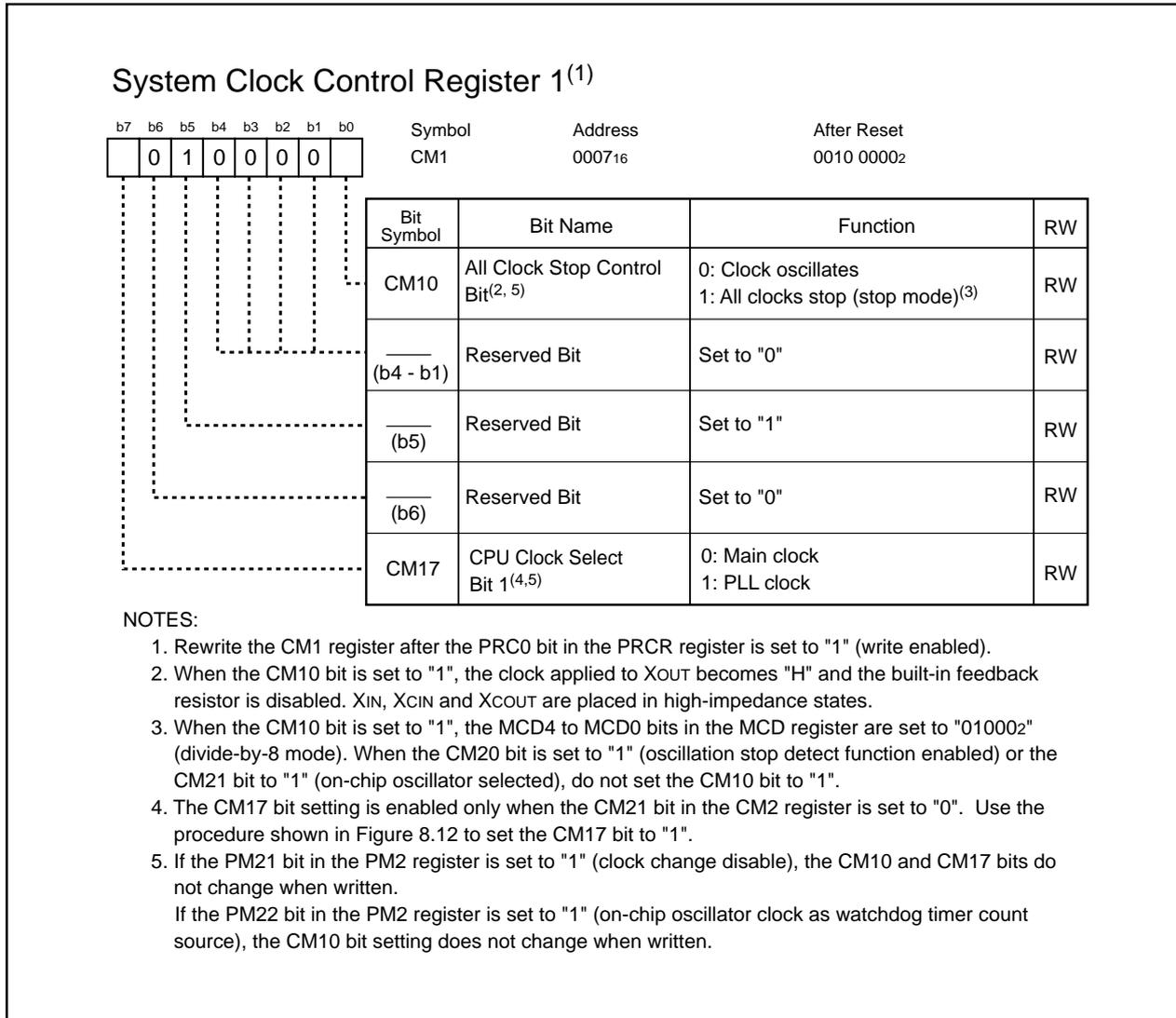
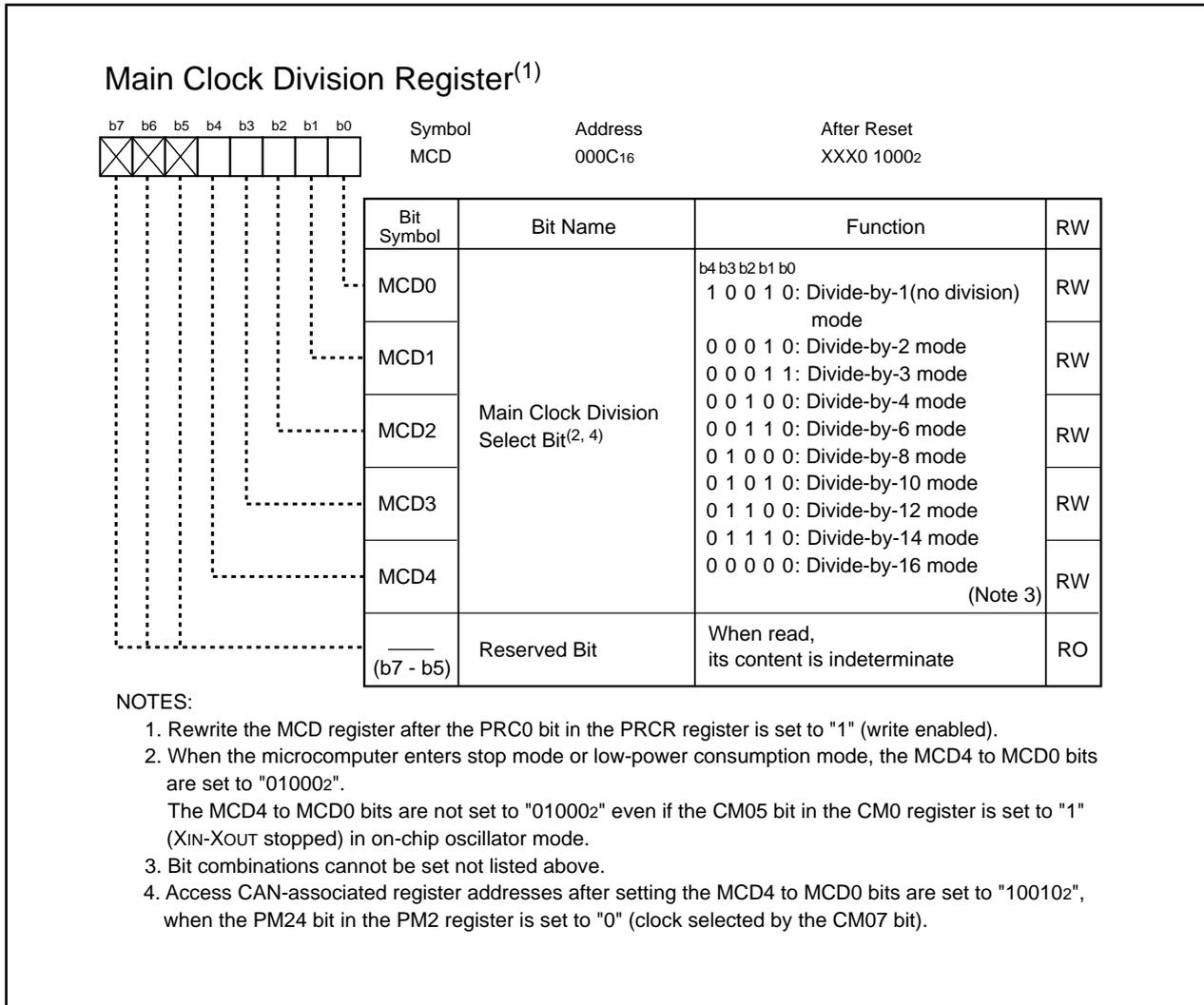


Figure 8.2 CM0 Register



**Figure 8.3 CM1 Register**



**Figure 8.4 MCD Register**

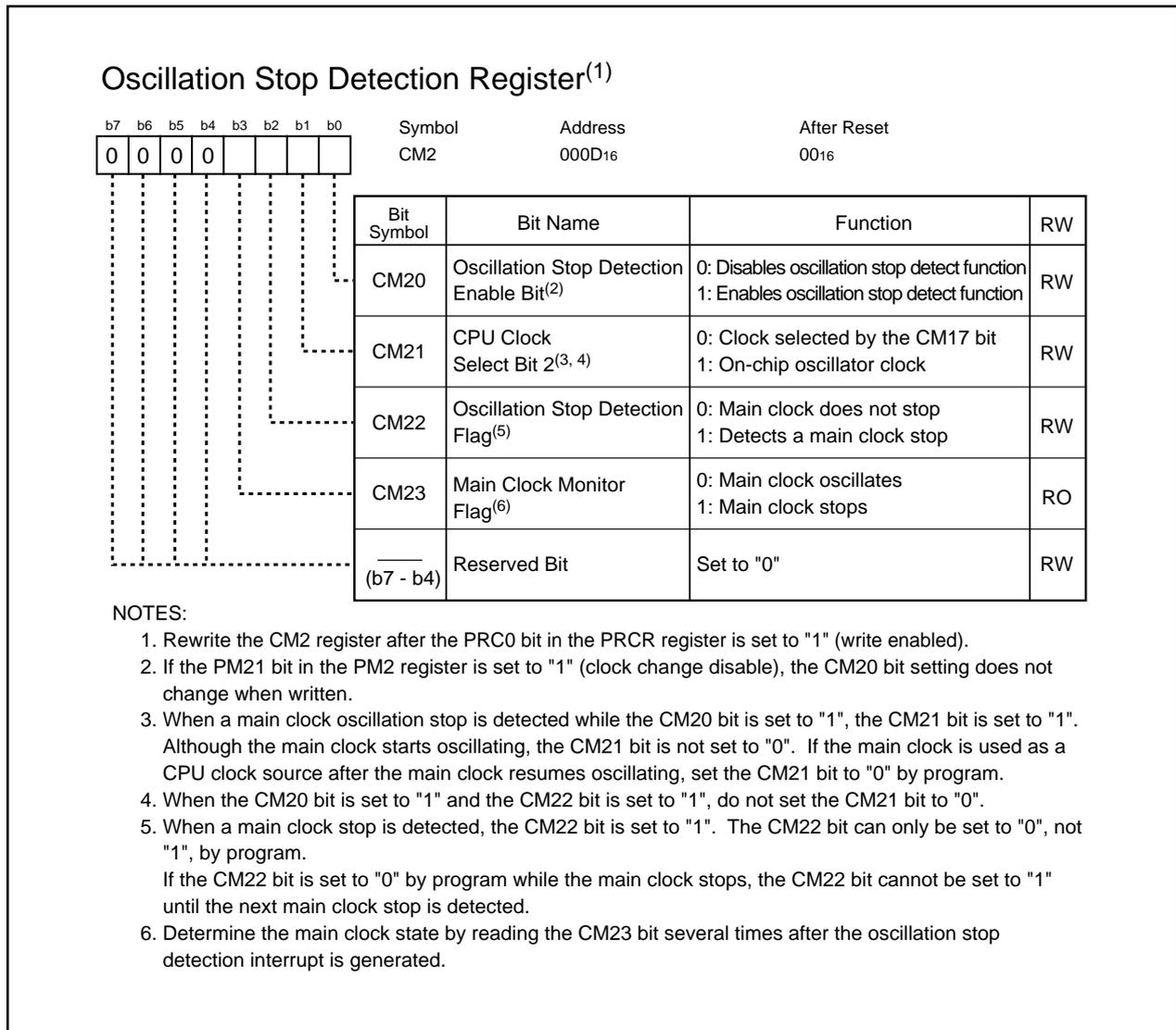


Figure 8.5 CM2 Register

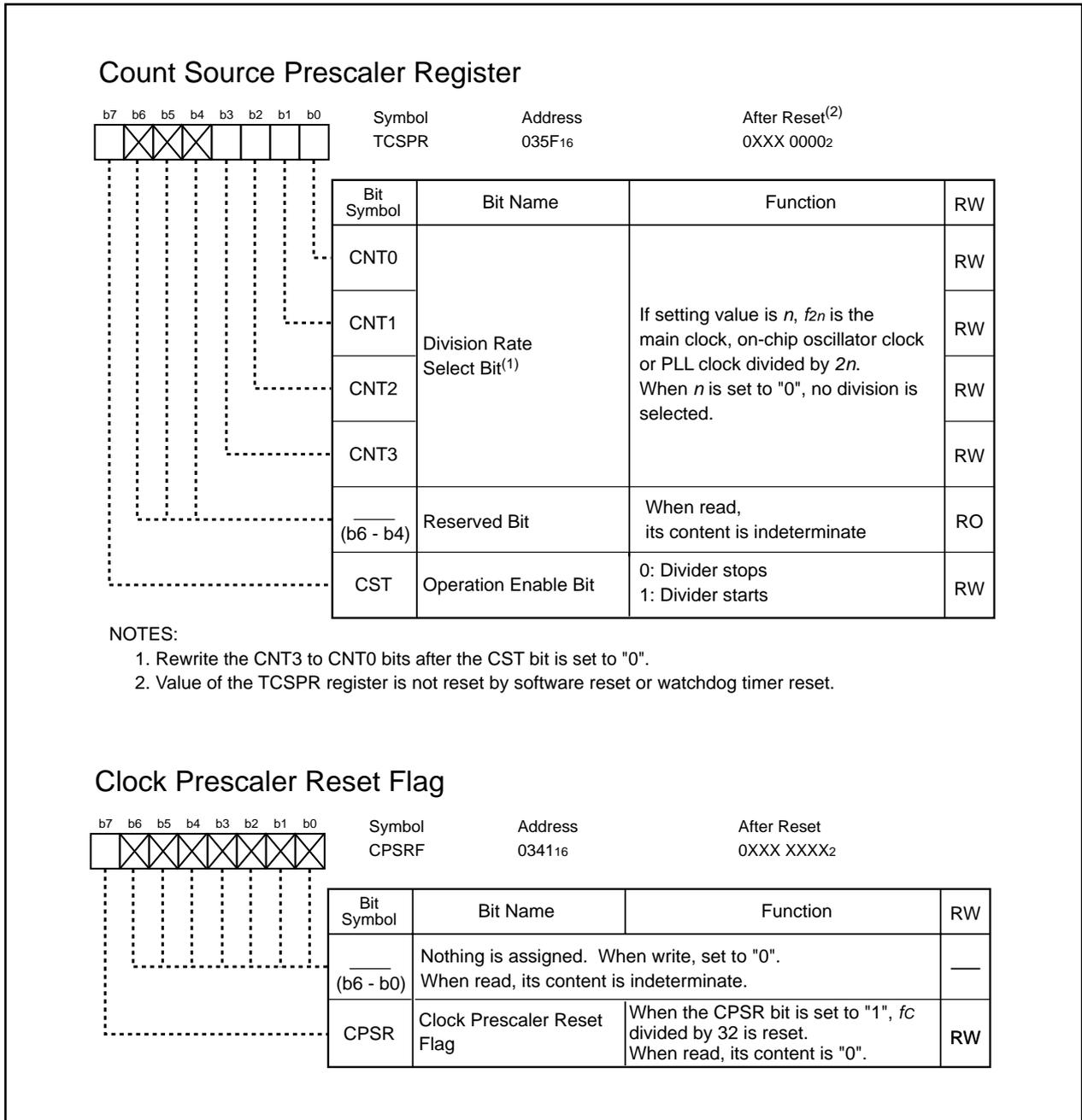


Figure 8.6 TCSR and CPSRF Registers

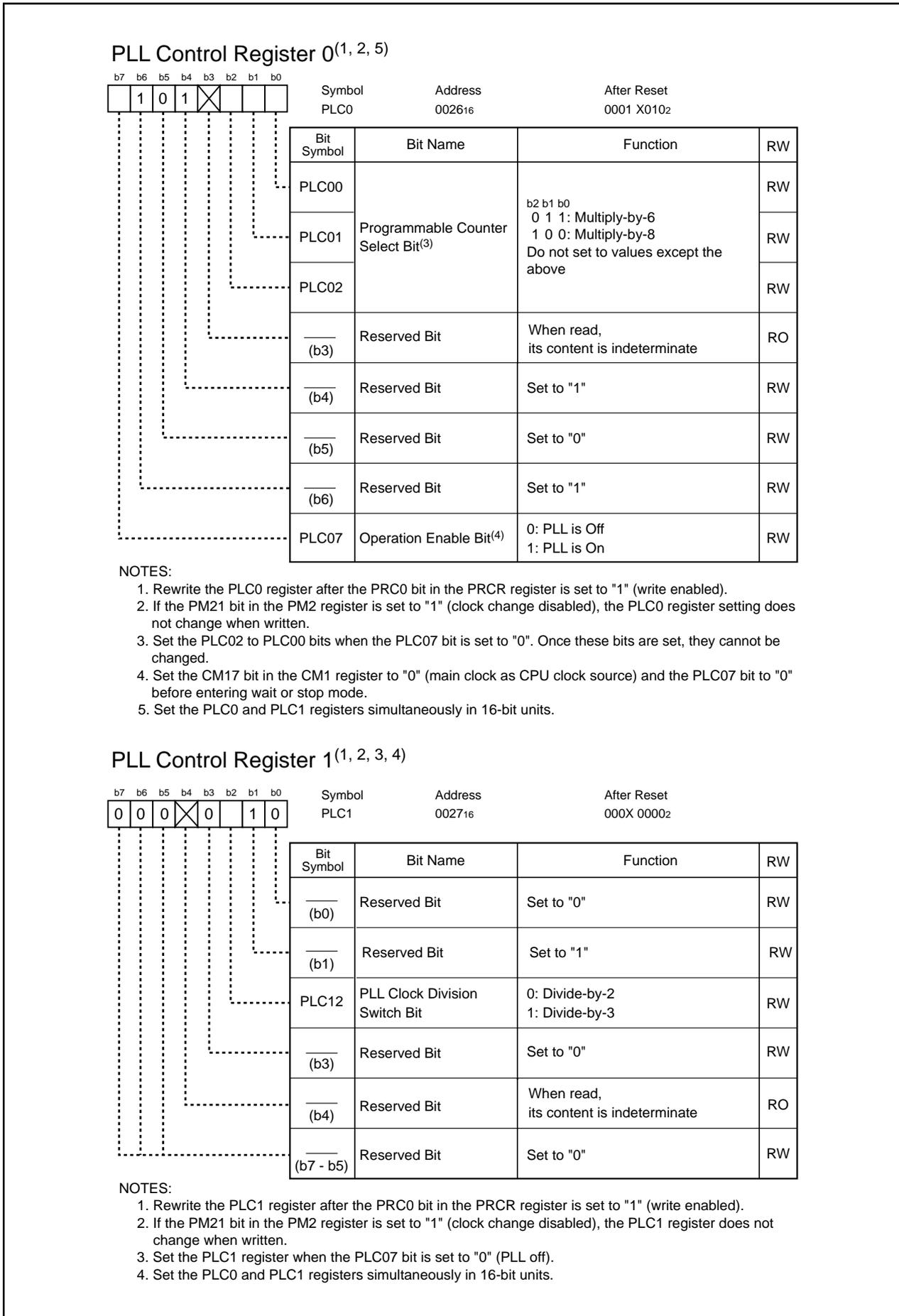
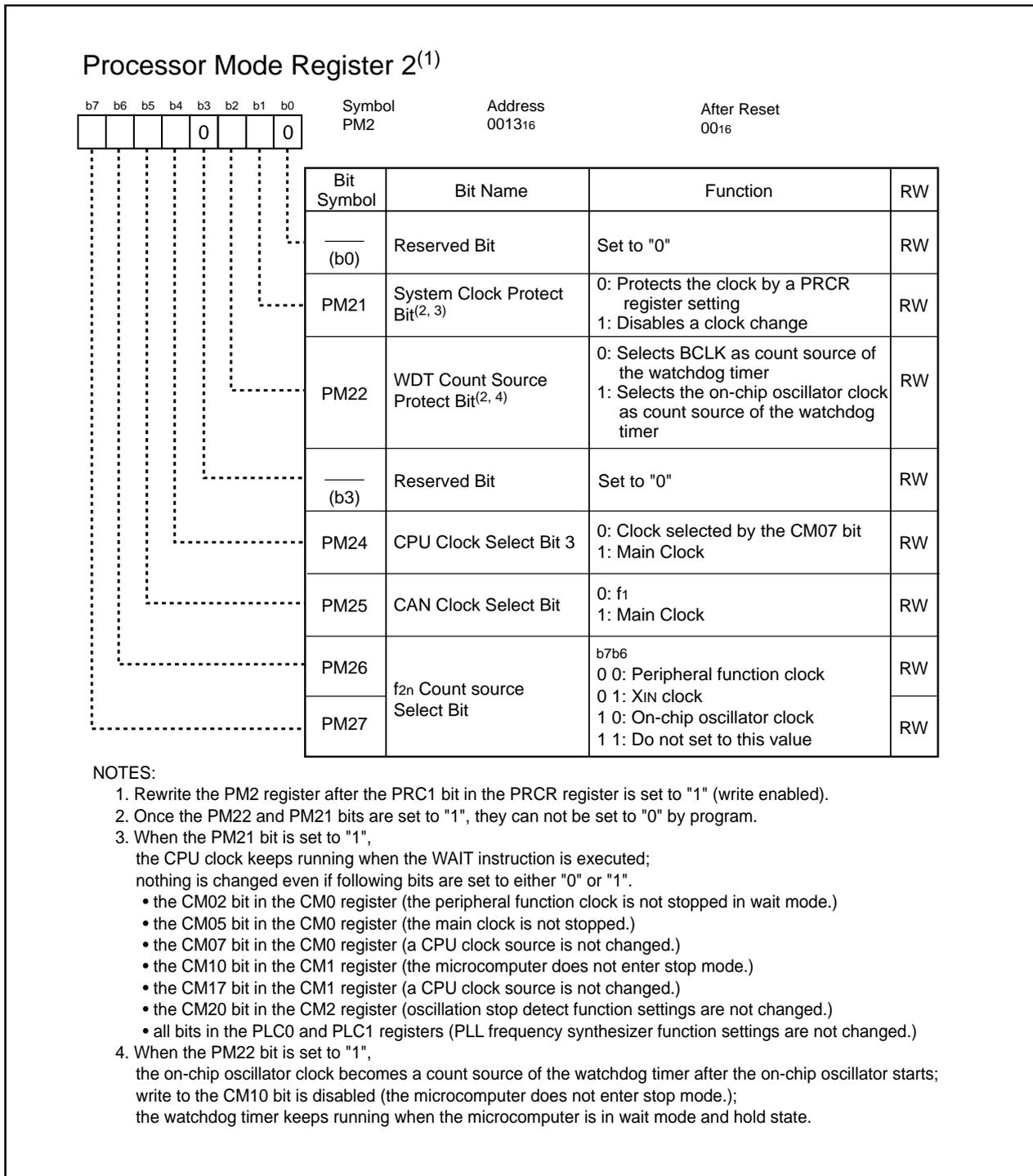


Figure 8.7 PLC0 and PLC1 Registers

**Figure 8.8 PM2 Register**

### 8.1.1 Main Clock

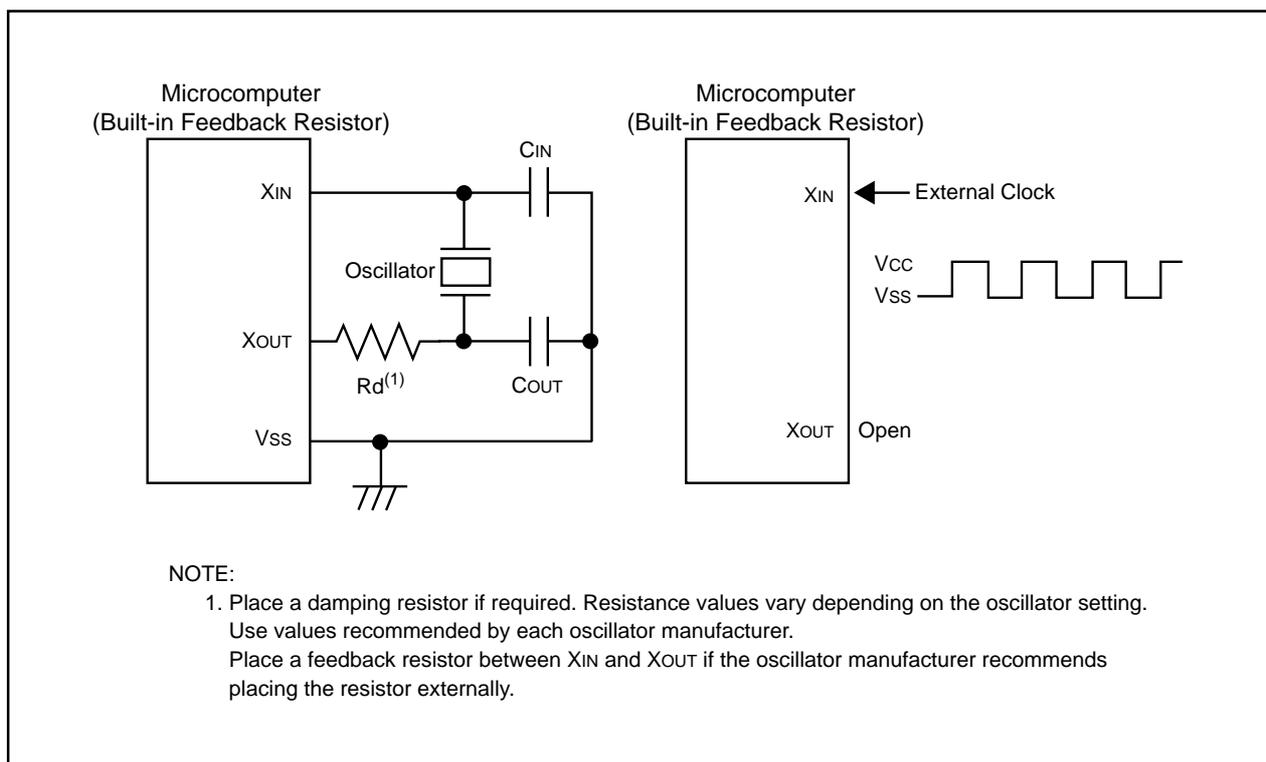
Main clock oscillation circuit generates the main clock. The main clock becomes clock source of the CPU clock and peripheral function clock.

The main clock oscillation circuit is configured by connecting an oscillator or resonator between the XIN and XOUT pins. The circuit has a built-in feedback resistor. The feedback resistor is separated from the oscillation circuit in stop mode to reduce power consumption. An external clock can be applied to the XIN pin in the main clock oscillation circuit. Figure 8.9 shows an example of a main clock circuit connection. Circuit constants vary depending on each oscillator. Use the circuit constant recommended by each oscillator manufacturer.

The main clock divided-by-eight becomes a CPU clock source after reset.

To reduce power consumption, set the CM05 bit in the CM0 register to "1" (main clock stopped) after switching the CPU clock source to the sub clock or on-chip oscillator clock. In this case, the clock applied to XOUT becomes high ("H"). XIN is pulled up by XOUT via the feedback resistor which remains on. When an external clock is applied to the XIN pin, do not set the CM05 bit to "1".

All clocks, including the main clock, stop in stop mode. Refer to **8.5 Power Consumption Control** for details.



**Figure 8.9 Main Clock Circuit Connection**

### 8.1.2 Sub Clock

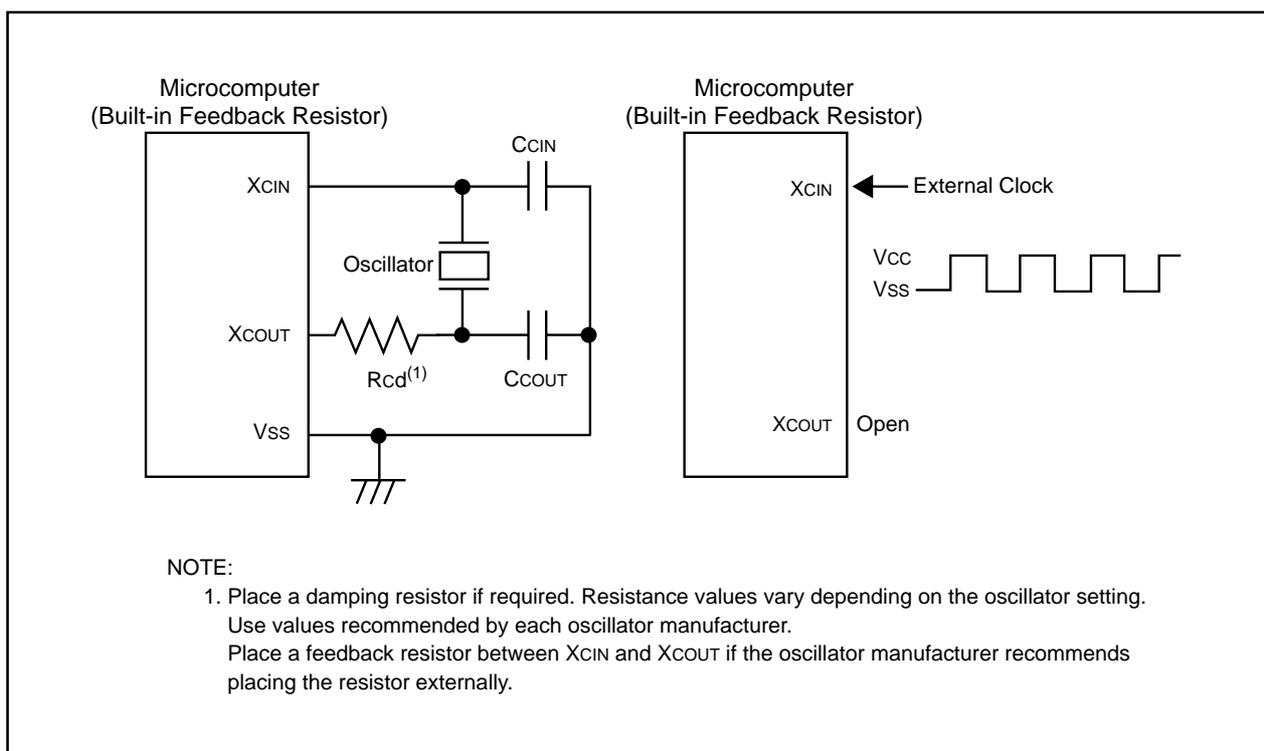
Sub clock oscillation circuit generates the sub clock. The sub clock becomes clock source of the CPU clock and for the timers A and B. The same frequency,  $f_c$ , as the sub clock can be output from the CLKOUT pin.

The sub clock oscillation circuit is configured by connecting a crystal oscillator between the XCIN and XCOUT pins. The circuit has a built-in feedback resistor. The feedback resistor is separated from the oscillation circuit in stop mode to reduce power consumption. An external clock can be applied to the XCIN pin. Figure 8.10 shows an example of a sub clock circuit connection. Circuit constants vary depending on each oscillator. Use the circuit constant recommended by each oscillator manufacturer.

The sub clock stops after reset. The feedback resistor is separated from the oscillation circuit. When the PD8\_6 and PD8\_7 bits in the PD8 register are set to "0" (input mode) and the PU25 bit in the PUR2 register is set to "0" (no pull-up), set the CM04 bit in the CM0 register to "1" (XCIN-XCOUT oscillation function). The sub clock oscillation circuit starts oscillating. To apply an external clock to the XCIN pin, set the CM04 bit to "1" when the PD8\_7 bit is set to "0" and the PU25 bit to "0". The clock applied to the XCIN pin becomes a clock source of the sub clock.

When the CM07 bit in the CM0 register is set to "1" (sub clock) after the sub clock oscillation has stabilized, the sub clock becomes a CPU clock source.

All clocks, including the sub clock, stop in stop mode. Refer to **8.5 Power Consumption Control** for details.



**Figure 8.10 Sub Clock Circuit Connection**

### 8.1.3 On-Chip Oscillator Clock

On-chip oscillator generates the on-chip oscillator clock. The 1-MHz on-chip oscillator clock becomes a clock source of the CPU clock and peripheral function clock.

The on-chip oscillator clock stops after reset. When the CM21 bit in the CM2 register is set to "1" (on-chip oscillator clock), the on-chip oscillator starts oscillating. Instead of the main clock, the on-chip oscillator clock becomes clock source of the CPU clock and peripheral function clock.

Table 8.2 shows bit settings for on-chip oscillator start condition.

**Table 8.2 Bit Settings for On-Chip Oscillator Start Condition**

CM2 Register	PM2 Register		Used as
CM21 Bit	PM22 Bit	PM27 and PM26 Bits	
1	0	0 0	CPU clock source or peripheral function clock source
0	1	0 0	Watchdog timer operating clock source (The clock keeps running when entering stop mode.)
0	0	0 1	f <sub>2n</sub> count source

#### 8.1.3.1 Oscillation Stop Detect Function

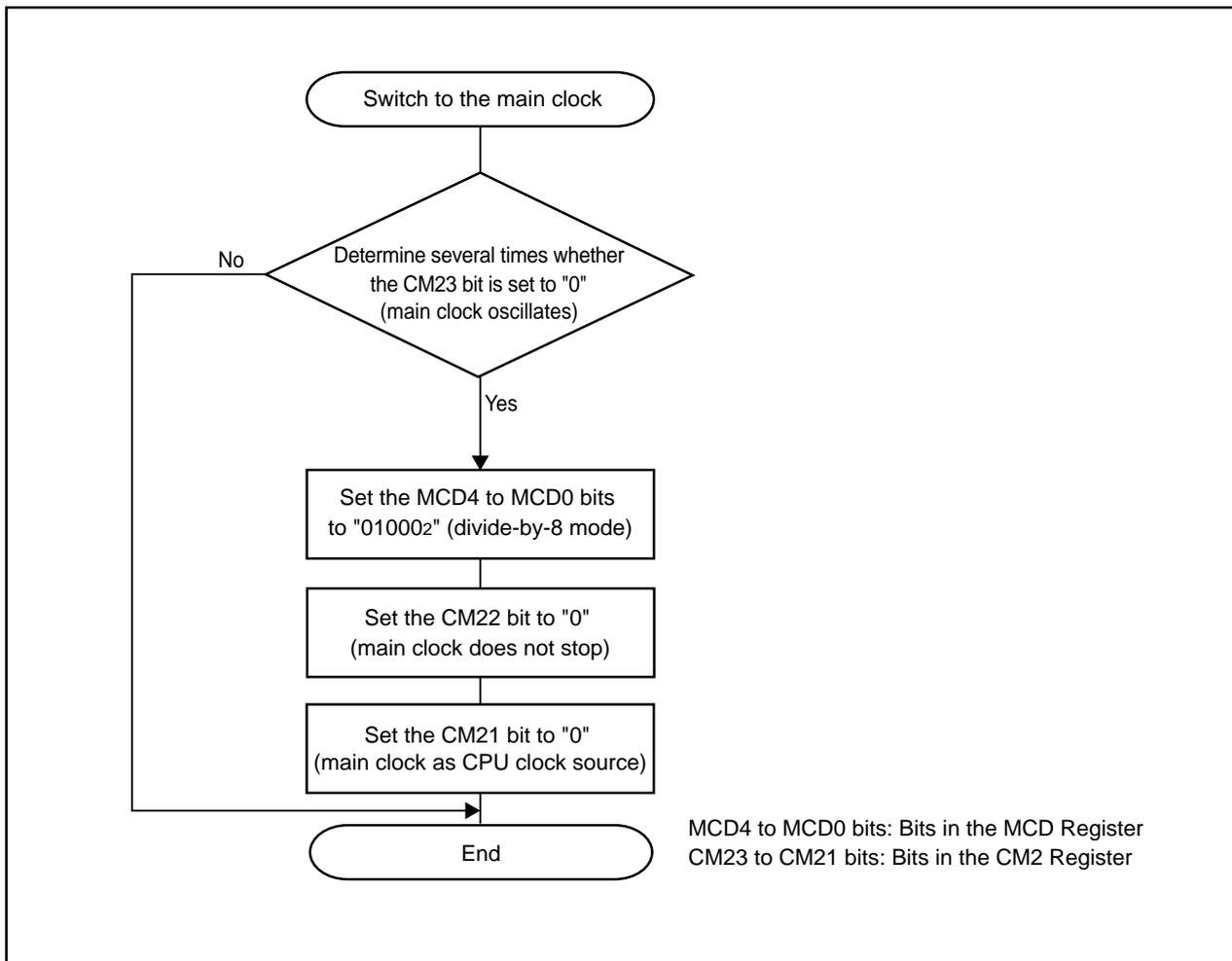
When the main clock is terminated by external source, the on-chip oscillator automatically starts oscillating to generate another clock.

When the CM 20 bit in the CM2 register is set to "1" (oscillation stop detect function enabled), an oscillation stop detection interrupt request is generated as soon as the main clock stops. Simultaneously, the on-chip oscillator starts oscillating. Instead of the main clock, the on-chip oscillator clock becomes clock source for the CPU clock and peripheral function clock. Associated bits are set as follows:

- The CM21 bit is set to "1" (on-chip oscillator clock becomes a clock source of the CPU clock.)
- The CM22 bit is set to "1" (main clock stop is detected.)
- The CM23 bit is set to "1" (main clock stops.) (See **Figure 8.14**)

#### 8.1.3.2 How to Use Oscillation Stop Detect Function

- The oscillation stop detection interrupt shares vectors with the watchdog timer interrupt. When these interrupts are used simultaneously, read the CM22 bit with an interrupt routine to determine if an oscillation stop detection interrupt request has been generated.
- When the main clock resumes running after an oscillation stop is detected, set the main clock as clock source of the CPU clock and peripheral function clock. Figure 8.11 shows the procedure to switch the on-chip oscillator clock to the main clock.
- In low-speed mode, when the main clock is stopped by setting the CM20 bit to "1", the oscillation stop detection interrupt request is generated. Simultaneously, the on-chip oscillator starts oscillating. The sub clock remains the CPU clock source. The on-chip oscillator clock becomes a clock source for the peripheral function clock.
- When the peripheral function clock stops running, the oscillation stop detect function is also disabled. To enter wait mode while the oscillation stop detect function is in use, set the CM02 bit in the CM0 register to "0" (peripheral clock does not stop in wait mode).
- The oscillation stop detect function is provided to handle main clock stop caused by external source. Set the CM20 bit to "0" (oscillation stop detect function disabled) when the main clock is terminated by program, i.e., entering stop mode or setting the CM05 bit to "1" (main clock oscillation stop).
- When the main clock frequency is 2MHz or less, the oscillation stop detect function is not available. Set the CM20 bit to "0".



**Figure 8.11 Switching Procedure from On-chip Oscillator Clock to Main Clock**

### 8.1.4 PLL Clock

The PLL frequency synthesizer generates the PLL clock based on the main clock. The PLL clock can be used as clock source for the CPU clock and peripheral function clock.

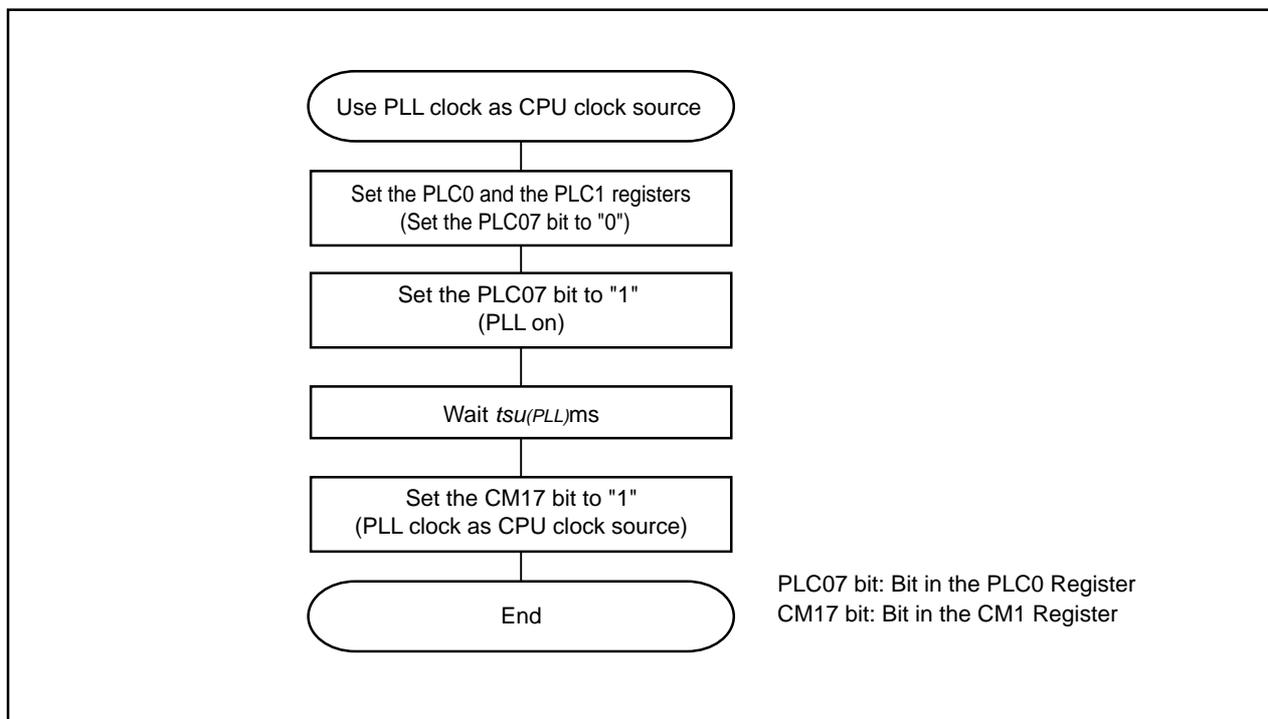
The PLL frequency synthesizer stops after reset. When the PLC07 bit is set to "1" (PLL on), the PLL frequency synthesizer starts operating. Wait  $t_{su}(PLL)$  ms for the PLL clock to stabilize.

The PLL clock can either be the clock output from the voltage controlled oscillator (VCO) divided-by-2 or divided-by-3. When the PLL clock is used as a clock source for the CPU clock or peripheral function clock, set each bit as is shown in Table 8.3. Figure 8.12 shows the procedure to use the PLL clock as the CPU clock source.

To enter wait or stop mode, set the CM17 bit to "0" (main clock as CPU clock source), set the PLC07 bit in the PLC0 register to "0" (PLL off) and then enter wait or stop mode.

**Table 8.3 Bit Settings to Use PLL Clock as CPU Clock Source**

f(X <sub>IN</sub> )	PLC0 Register			PLC1 Register	PLL Clock
	PLC02 Bit	PLC01 Bit	PLC00 Bit	PLC12 Bit	
10 MHz	0	1	1	0	30 MHz
				1	20 MHz
8 MHz	1	0	0	0	32 MHz
				1	21.3 MHz



**Figure 8.12 Procedure to Use PLL Clock as CPU Clock Source**

## 8.2 CPU Clock and BCLK

The CPU operating clock is referred to as the CPU clock. The CPU clock is also a count source for the watchdog timer. After reset, the CPU clock is the main clock divided-by-8. In memory expansion or micro-processor mode, the clock having the same frequency as the CPU clock can be output from the BCLK pin as BCLK. Refer to **8.4 Clock Output Function** for details.

The main clock, sub clock, on-chip oscillator clock or PLL clock can be selected as a clock source for the CPU clock. Table 8.4 shows CPU clock source and bit settings.

When the main clock, on-chip oscillator clock or PLL clock is selected as a clock source of the CPU clock, the selected clock divided-by-1 (no division), -2, -3, -4, -6, -8, -10, -12, -14 or -16 becomes the CPU clock. The MCD4 to MCD0 bits in the MCD register select the clock division.

When the microcomputer enters stop mode or low-power consumption mode (except when the on-chip oscillator clock is the CPU clock), the MCD4 to MCD0 bits are set to "010002" (divide-by-8 mode). Therefore, when the main clock starts running, the CPU clock enters medium-speed mode (divide-by-8).

**Table 8.4 CPU Clock Source and Bit Settings**

CPU Clock Source	CM0 Register	CM1 Register	CM2 Register	PM2 Register
	CM07 Bit	CM17 Bit	CM21 Bit	PM24 Bit
Main Clock	0	0	0	0
Main Clock (Main Clock Direct Mode) <sup>(1)</sup>	0	0	0	1
Sub Clock	1	0	0	0
On-Chip Oscillator Clock	0	0	1	0
PLL Clock	0	1	0	0

NOTE:

1. Refer to **22.2 CAN Clock** for details.

## 8.3 Peripheral Function Clock

The peripheral function clock becomes an operating clock or count source for peripheral functions excluding the watchdog timer.

### 8.3.1 f<sub>1</sub>, f<sub>8</sub>, f<sub>32</sub> and f<sub>2n</sub>

f<sub>1</sub>, f<sub>8</sub> and f<sub>32</sub> are the peripheral function clock, selected by the CM21 bit, divided-by-1, -8, or -32. The PM27 and PM26 bits in the PM2 register selects a f<sub>2n</sub> count source from the peripheral clock, X<sub>IN</sub> clock, and the on-chip oscillator clock. The CNT3 to CNT0 bits in the TCSPR register selects a f<sub>2n</sub> division. (n=0 to 15. No division when n=0.)

f<sub>1</sub>, f<sub>8</sub>, f<sub>32</sub> and f<sub>2n</sub> stop when the CM02 bit in the CM0 register to "1" (peripheral function stops in wait mode) to enter wait mode or when in low-power consumption mode.

f<sub>1</sub>, f<sub>8</sub> and f<sub>2n</sub> are used as an operating clock of the serial I/O and count source of the timers A and B. f<sub>1</sub> is also used as an operating clock for the intelligent I/O.

The CLKOUT pin outputs f<sub>8</sub> and f<sub>32</sub>. Refer to **8.4 Clock Output Function** for details.

### 8.3.2 f<sub>AD</sub>

f<sub>AD</sub> is an operating clock for the A/D converter and has the same frequency as either the main clock<sup>(1)</sup> or the on-chip oscillator clock. The CM21 bit determines which clock is selected.

If the CM02 bit is set to "1" (peripheral function stop in wait mode) to enter wait mode, f<sub>AD</sub> stops. f<sub>AD</sub> also stops in low-power consumption mode.

NOTE:

1. The PLL clock, instead of the main clock, when the CM17 bit is set to "1" (PLL clock).

### 8.3.3 fc32

fc32 is the sub clock divided by 32. fc32 is used as a count source for the timers A and B. fc32 is available when the sub clock is running.

### 8.3.4 fCAN

fCAN has the same frequency as the main clock. It is a clock for the CAN module only.

## 8.4 Clock Output Function

The CLKOUT pin outputs fc, f8 or f32.

In memory expansion mode or microprocessor mode, a clock having the same frequency as the CPU clock can be output from the BCLK pin as BCLK.

Table 8.5 lists CLKOUT pin function in single-chip mode.

**Table 8.5 CLKOUT Pin in Single-Chip Mode**

PM0 Register <sup>(1)</sup>	CM0 Register <sup>(2)</sup>		CLKOUT Pin Function
PM07 Bit	CM01 Bit	CM00 Bit	
—	0	0	P53 I/O port
1	0	1	Outputs fc
1	1	0	Outputs f8
1	1	1	Outputs f32

- : Can be set to either "0" or "1"

**NOTES:**

1. Rewrite the PM0 register after the PRC1 bit in the PRCR register is set to "1" (write enabled).
2. Rewrite the CM0 register after the PRC0 bit in the PRCR register is set to "1" (write enabled).

## 8.5 Power Consumption Control

Normal operating mode, wait mode and stop mode are provided as the power consumption control.

All mode states, except wait mode and stop mode, are called normal operating mode in this section. Figure 8.13 shows a block diagram of status transition in wait mode and stop mode. Figure 8.14 shows a block diagram of status transition in all modes.

### 8.5.1 Normal Operating Mode

The normal operating mode is further separated into six modes.

In normal operating mode, the CPU clock and peripheral function clock are supplied to operate the CPU and peripheral function. The power consumption control is enabled by controlling a CPU clock frequency. The higher the CPU clock frequency is, the more processing power increases. The lower the CPU clock frequency is, the more power consumption decreases. When unnecessary oscillation circuit stops, power consumption is further reduced.

#### 8.5.1.1 High-Speed Mode

The main clock<sup>(1)</sup> becomes the CPU clock and a clock source of the peripheral function clock. When the sub clock runs, fc32 can be used as a count source for the timers A and B.

#### 8.5.1.2 Medium-Speed Mode

The main clock<sup>(1)</sup> divided-by-2, -3, -4, -6, -8, -10, -12, -14, or -16 becomes the CPU clock. The main clock<sup>(1)</sup> is a clock source for the peripheral function clock. When the sub clock runs, fc32 can be used as a count source for the timers A and B.

#### 8.5.1.3 Low-Speed Mode

The sub clock becomes the CPU clock. The main clock<sup>(1)</sup> is a clock source for the peripheral function clock. fc32 can be used as a count source for the timers A and B.

#### 8.5.1.4 Low-Power Consumption Mode

The microcomputer enters low-power consumption mode when the main clock stops in low-speed mode. The sub clock becomes the CPU clock. Only fc32 can be used as a count source for the timers A and B and the peripheral function clock. In low-power consumption mode, the MCD4 to MCD0 bits in the MCD register are set to "010002" (divide-by-8 mode). Therefore, when the main clock resumes running, the microcomputer is in medium-speed mode (divide-by-8 mode).

#### 8.5.1.5 On-Chip Oscillator Mode

The on-chip oscillator clock divided-by-1 (no division), -2, -3, 4-, -6, -8, -10, -12, -14, or -16 becomes the CPU clock. The on-chip oscillator clock is a clock source for the peripheral function clock. When the sub clock runs, fc32 can be used as a count source for the timers A and B.

### 8.5.1.6 On-Chip Oscillator Low-Power Consumption Mode

The microcomputer enters on-chip oscillator low-power consumption mode when the main clock stops in on-chip oscillator mode. The on-chip oscillator clock divided-by-1 (no division), -2, -3, -4, -6, -8, -10, -12, -14, or -16 becomes the CPU clock. The on-chip oscillator clock is a clock source for the peripheral function clock. When the sub clock runs, fc32 can be used as a count source for the timers A and B. Switch the CPU clock after the clock to be switched to stabilize. Sub clock oscillation will take longer<sup>(2)</sup> to stabilize. Wait, by program, until the clock stabilizes directly after turning the microcomputer on or exiting stop mode.

To switch the on-chip oscillator clock to the main clock, enter medium-speed mode (divide-by-8) after the main clock is divided by eight in on-chip oscillator mode (the MCD4 to MCD0 bits in the MCD register are set to "010002").

Do not enter on-chip oscillator mode or on-chip oscillator low-power consumption mode from low-speed mode or low-power consumption mode and vice versa.

#### NOTES:

1. The PLL clock, instead of the main clock, when the CM17 bit is set to "1" (PLL clock).
2. Contact your oscillator manufacturer for oscillation stabilization time.

## 8.5.2 Wait Mode

In wait mode, the CPU clock stops running. The CPU and watchdog timer, operated by the CPU clock, also stop. When the PM22 bit in the PM2 register is set to "1" (on-chip oscillator clock as watchdog timer count source), the watchdog timer continues operating. Because the main clock, sub clock and on-chip oscillator clock continue running, peripheral functions using these clocks also continue operating.

### 8.5.2.1 Peripheral Function Clock Stop Function

If the CM02 bit in the CM0 register is set to "1" (peripheral function clock stops in wait mode), f1, f8, f32, f2n (when peripheral clock is selected as a count source), and fAD stop in wait mode. Power consumption can be reduced. f2n, when XIN clock or on-chip oscillator clock is selected as a count source, and fc32 do not stop running.

### 8.5.2.2 Entering Wait Mode

If wait mode is entered after setting the CM02 bit to "1", set the MCD4 to MCD0 bits in the MCD register to be the 10-MHz or less CPU clock frequency after dividing the main clock.

Enter wait mode after setting the followings.

- Initial Setting

Set each interrupt priority level after setting the exit priority level required to exit wait mode, controlled by the RLVL2 to RLVL0 bits in the RLVL register, to "7".

- Before Entering Wait Mode
  - (1) Set the I flag to "0"
  - (2) Set the interrupt priority level of the interrupt being used to exit wait mode
  - (3) Set the interrupt priority levels of the interrupts, not being used to exit wait mode, to "0"
  - (4) Set IPL in the FLG register. Then set the exit priority level to the same level as IPL  
 Interrupt priority level of the interrupt used to exit wait mode > IPL = the exit priority level
  - (5) Set the PRC0 bit in the PRCR register to "1"
  - (6) If the CPU clock source is the PLL clock, set the CM17 bit in the CM1 register to "0" (main clock) and PLC07 bit in the PLC0 register to "0" (PLL off)
  - (7) Set the I flag to "1"
  - (8) Execute the WAIT instruction
  
- After Exiting Wait Mode
 

Set the exit priority level to "7" as soon as exiting wait mode.

### 8.5.2.3 Pin Status in Wait Mode

Table 8.6 lists pin states in wait mode.

**Table 8.6 Pin States in Wait Mode**

Pin		Single-Chip Mode
Ports		Maintains state immediately before entering wait mode
CLKOUT	When fc is selected	Outputs clock
	When f8, f32 are selected	Outputs the clock when the CM02 bit in the CM0 register is set to "0" (peripheral function clock does not stop in wait mode). Maintains state immediately before entering wait mode when the CM02 bit is set to "1" (peripheral function clock stops in wait mode).

### 8.5.2.4 Exiting Wait Mode

Wait mode is exited by the hardware reset,  $\overline{\text{NMI}}$  interrupt or peripheral function interrupts.

When the hardware reset or  $\overline{\text{NMI}}$  interrupt, but not the peripheral function interrupts, is used to exit wait mode, set the ILVL2 to ILVL0 bits for the peripheral function interrupts to "0002" (interrupt disabled) before executing the WAIT instruction.

CM02 bit setting affects the peripheral function interrupts. When the CM02 bit in the CM0 register 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), peripheral functions using the peripheral function clock stop. Therefore, the peripheral function interrupts cannot be used to exit wait mode. However, the peripheral function interrupts caused by an external clock, fc32, or f2n whose count source is the XIN clock or on-chip oscillator clock, can be used to exit wait mode.

The CPU clock used when exiting wait mode by the peripheral function interrupts or  $\overline{\text{NMI}}$  interrupt is the same CPU clock used when the WAIT instruction is executed.

Table 8.7 shows interrupts to be used to exit wait mode and usage conditions.

**Table 8.7 Interrupts to Exit Wait Mode**

Interrupt	When CM02=0	When CM02=1
NMI Interrupt	Can be used	Can be used
Serial I/O Interrupt	Can be used when either internal or external clock is selected	Can be used when external clock or f <sub>2n</sub> (when X <sub>IN</sub> clock or on-chip oscillator is selected) is selected
Key Input Interrupt	Can be used	Can be used
A/D Conversion Interrupt	Can be used in single or single-sweep mode	Do not use
Timer A Interrupt Timer B Interrupt	Can be used in all modes	Can be used in event counter mode or when count source is f <sub>C32</sub> or f <sub>2n</sub> (when X <sub>IN</sub> clock or on-chip oscillator is selected)
$\overline{\text{INT}}$ Interrupt	Can be used	Can be used
CAN Interrupt	Can be used	Do not use
Intelligent I/O Interrupt	Can be used	Do not use

### 8.5.3 Stop Mode

In stop mode, all oscillators and resonators stop. The CPU clock and peripheral function clock, as well as the CPU and peripheral functions operated by these clocks, also stop. The least power required to operate the microcomputer is in stop mode. The internal RAM holds its data when the voltage applied to the VCC pin is V<sub>RAM</sub> or more. If the voltage applied to the VCC pin is 2.7V or less, the voltage must be V<sub>CC</sub> ≥ V<sub>RAM</sub>.

The following interrupts can be used to exit stop mode:

- $\overline{\text{NMI}}$  interrupt
- Key Input Interrupt
- $\overline{\text{INT}}$  interrupt
- Timer A and B interrupt (Available when the timer counts external pulse, having its 100Hz or less frequency, in event counter mode)

### 8.5.3.1 Entering Stop Mode

Stop mode is entered when setting the CM10 bit in the CM10 register to "1" (all clocks stops). The MCD4 to MCD0 bits in the MCD register become set to "010002" (divide-by-8 mode).

Enter stop mode after setting the followings.

- Initial Setting

Set each interrupt priority level after setting the exit priority level required to exit stop mode, controlled by the RLVL2 to RLVL0 bits in the RLVL register, to "7".

- Before Entering stop mode

- (1) Set the I flag to "0"

- (2) Set the interrupt priority level of the interrupt being used to exit stop mode

- (3) Set the interrupt priority levels of the interrupts, not being used to exit stop mode, to "0"

- (4) Set IPL in the FLG register. Then set the exit priority level to the same level as IPL

Interrupt priority level of the interrupt used to exit stop mode > IPL = the exit priority level

- (5) Set the PRC0 bit in the PRCR register to "1" (write enabled)

- (6) Select the main clock as the CPU clock

- When the CPU clock source is the sub clock,

- (a) set the CM05 bit in the CM0 register to "0" (main clock oscillates)

- (b) set the CM07 bit in the CM0 register to "0" (clock selected by the CM21 bit divided by MCD register setting)

- When the CPU clock source is the PLL clock,

- (a) set the CM17 bit in the CM1 register to "0" (main clock)

- (b) set the PLC07 bit in the PLC0 register to "0" (PLL off)

- When main clock direct mode is used,

- (a) set the PRC1 bit in the PRCR register to "1" (write enabled)

- (b) set the PM24 bit in the PM2 register to "0" (clock selected by the CM07 bit)

- When the CPU clock source is the on-chip oscillator clock,

- (a) set MCD4 to MCD0 bits to "010002" (divide-by-8 mode)

- (b) set the CM05 bit to "0" (main clock oscillates)

- (c) set the CM21 bit in the CM2 register to "0" (clock selected by the CM17 bit)

- (7) The oscillation stop detect function is used, set the CM20 bit in the CM2 register to "0" (oscillation stop detect function disabled)

- (8) Set the I flag to "1"

- (9) Set the CM10 bit to "1" (all clocks stops)

- After Exiting Stop Mode

Set the exit priority level to "7" as soon as exiting stop mode.

### 8.5.3.2 Exiting Stop Mode

Stop mode is exited by the hardware reset,  $\overline{\text{NMI}}$  interrupt or peripheral function interrupts (key input interrupt and  $\overline{\text{INT}}$  interrupt).

When the hardware reset or  $\overline{\text{NMI}}$  interrupt, but not the peripheral function interrupts, is used to exit wait mode, set all ILVL2 to ILVL0 bits in the interrupt control registers for the peripheral function interrupt to "0002" (interrupt disabled) before setting the CM10 bit to "1" (all clocks stops).

8.5.3.3 Pin Status in Stop Mode

Table 8.8 lists pin status in stop mode.

Table 8.8 Pin Status in Stop Mode

Pin		Single-Chip Mode
Ports		Maintains state immediately before entering stop mode
CLKOUT	When fc selected	"H"
	When f8, f32 selected	Maintains state immediately before entering stop mode
XIN		Placed in a high-impedance state
XOUT		"H"
XCIN, XCOUT		Placed in a high-impedance state

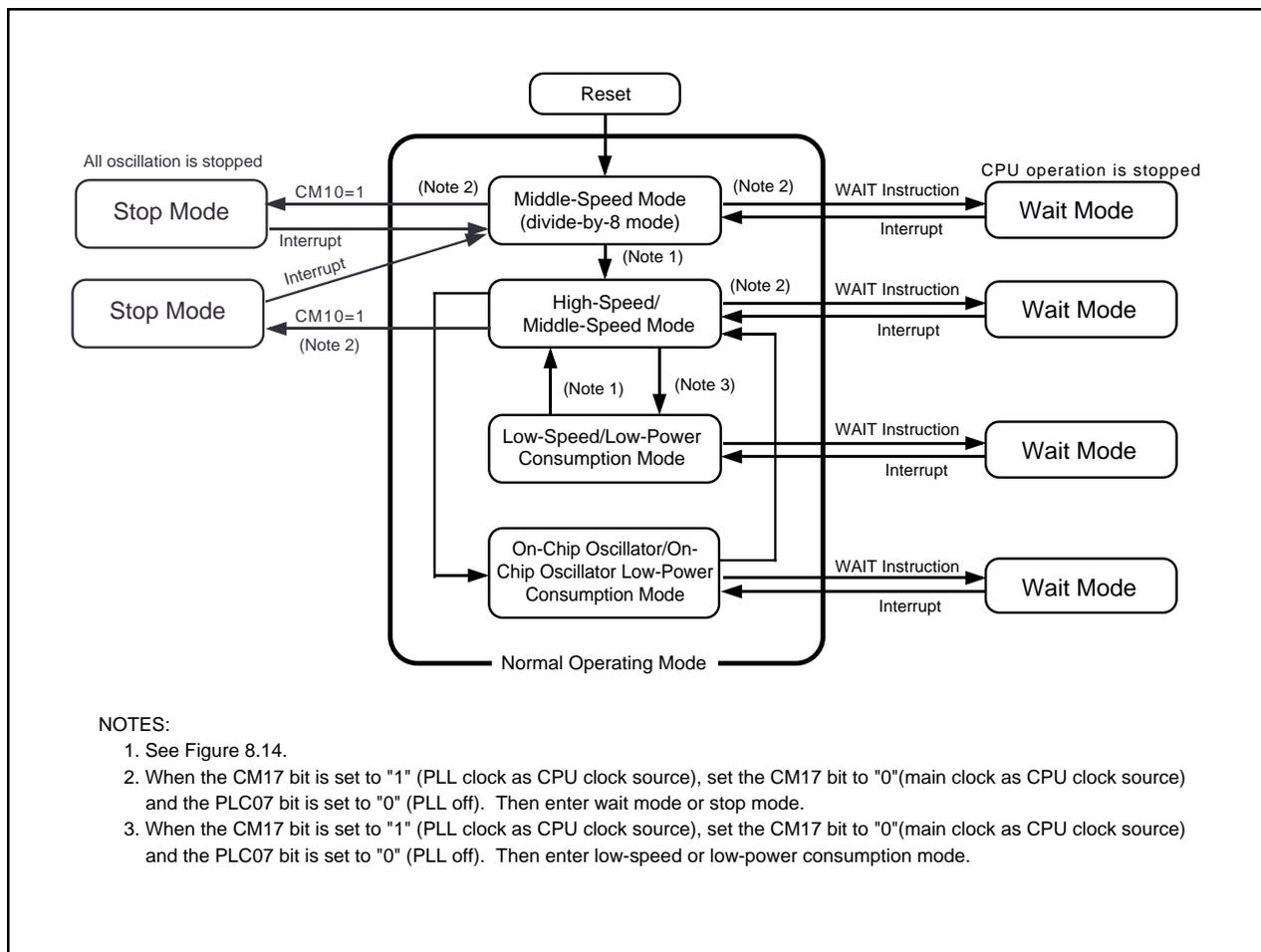


Figure 8.13 Status Transition in Wait Mode and Stop Mode

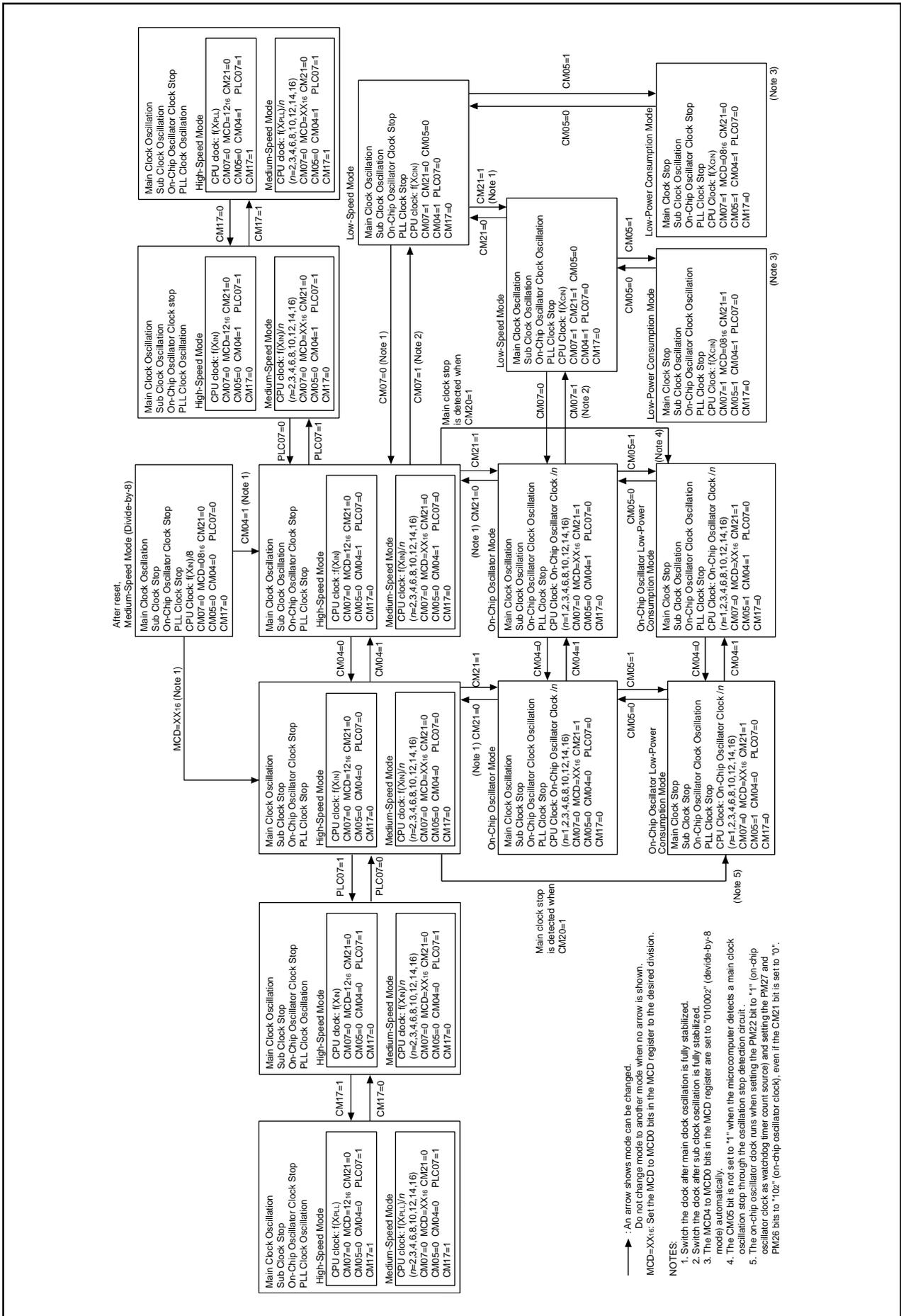


Figure 8.14 Status Transition

## 8.6 System Clock Protect Function

The system clock protect function prohibits the CPU clock from changing clock sources when the main clock is selected as the CPU clock source. This prevents the CPU clock from stopping the program crash. When the PM21 bit in the PM2 register is set to "1" (clock change disabled), the following bits cannot be written to:

- The CM02 bit, CM05 bit and CM07 bit in the CM0 register
- The CM10 bit and CM17 bit in the CM1 register
- The CM20 bit in the CM2 register
- All bits in the PLC0 and PLC1 registers

The CPU clock continues running when the WAIT instruction is executed.

To use the system clock protect function, set the CM05 bit in the CM0 register to "0" (main clock oscillation) and CM07 bit to "0" (main clock as BCLK clock source) and follow the procedure below.

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

When the PM21 bit is set to "1", do not execute the WAIT instruction.

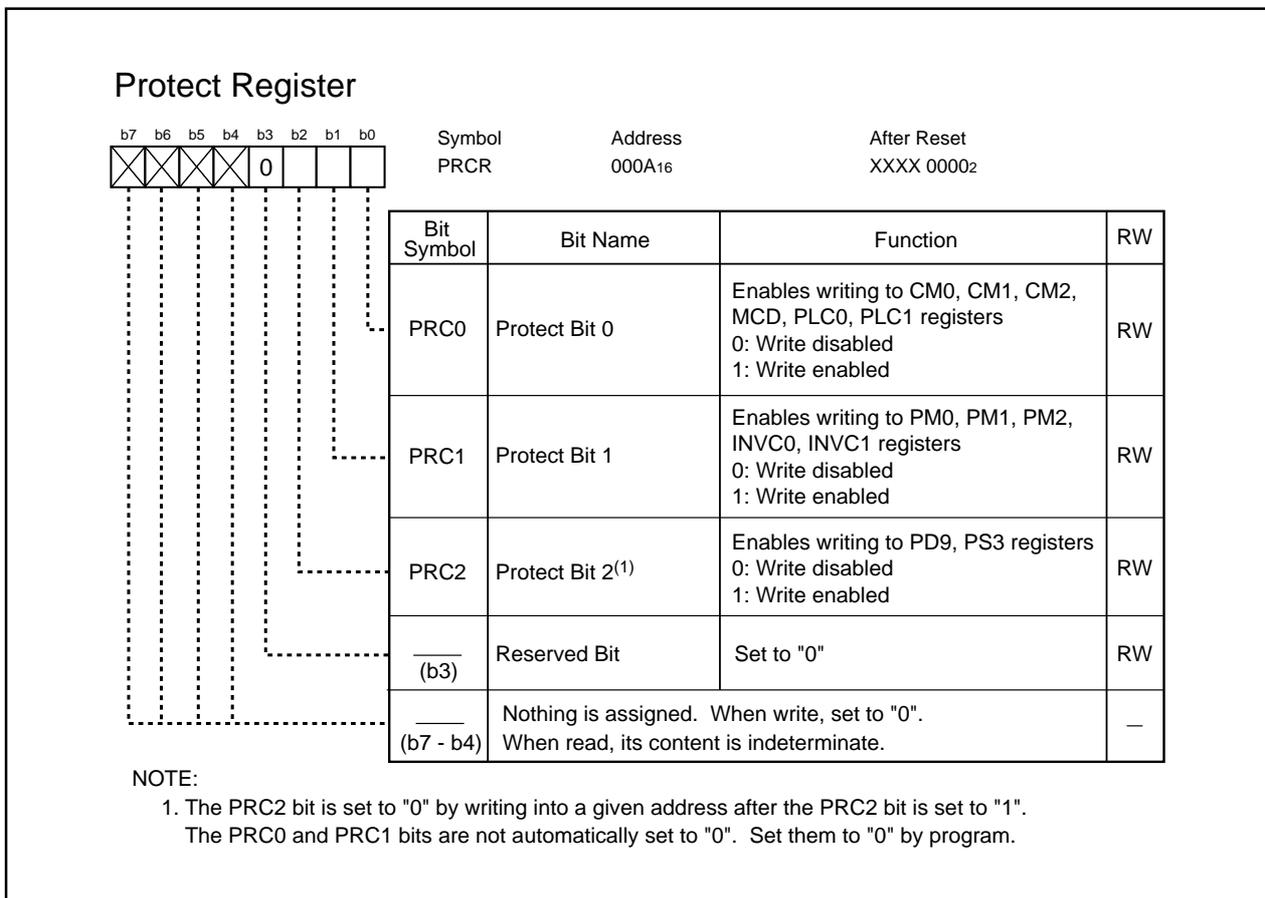
## 9. Protection

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

Figure 9.1 shows the PRCR register. Individual bit in the PRCR register protects the following registers:

- The PRC0 bit protects the CM0, CM1, CM2, MCD, PLC0, and PLC1 registers;
- The PRC1 bit protects the PM0, PM1, PM2, INVC0, and INVC1 registers;
- The PRC2 bit protects the PD9 and PS3 registers.

The PRC2 bit is set to "0" (write disabled) when data is written to a given address after setting the PRC2 bit to "1" (write enabled). Set the PD9 and PS3 registers immediately after setting the PRC2 bit in the PRCR register to "1" (write enabled). Do not generate an interrupt or a DMA transfer between the instruction to set to the PRC2 bit to "1" and the following instruction. The PRC0 and PRC1 bits are not set to "0" even if data is written to a given address. Set the PRC0 and PRC1 bits to "0" by program.

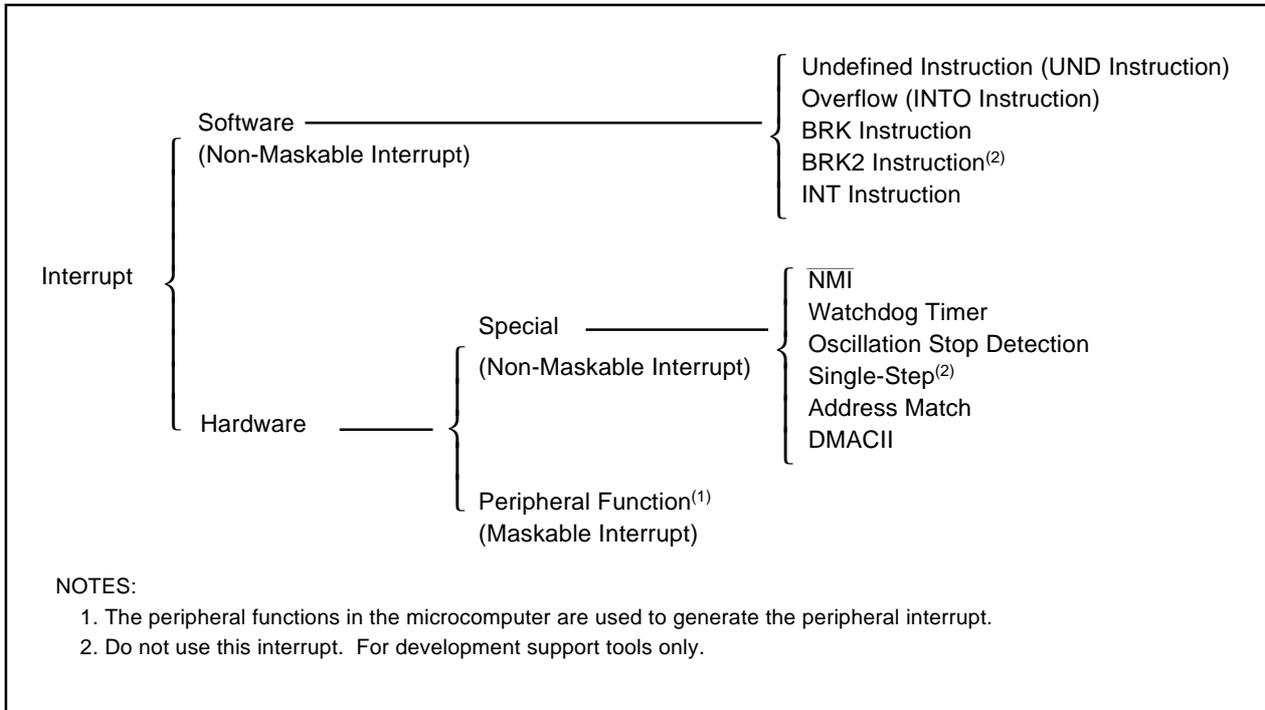


**Figure 9.1 PRCR Register**

# 10. Interrupts

## 10.1 Types of Interrupts

Figure 10.1 shows types of interrupts.



**Figure 10.1 Interrupts**

- Maskable Interrupt

The I flag enables or disables an interrupt.

The interrupt priority order based on interrupt priority level **can be changed**.

- Non-Maskable Interrupt

The I flag does not enable nor disable an interrupt .

The interrupt priority order based on interrupt priority level **cannot be changed**.

## 10.2 Software Interrupts

Software interrupt occurs when an instruction is executed. The software interrupts are non-maskable interrupts.

### 10.2.1 Undefined Instruction Interrupt

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

### 10.2.2 Overflow Interrupt

The overflow interrupt occurs when the O flag in the FLG register is set to "1" (overflow of arithmetic operation) and the INTO instruction is executed.

Instructions to set the O flag are :

ABS, ADC, ADCF, ADD, ADDX, CMP, CMPX, DIV, DIVU, DIVX, NEG, RMPA, SBB, SCMPU, SHA, SUB, SUBX

### 10.2.3 BRK Interrupt

The BRK interrupt occurs when the BRK instruction is executed.

### 10.2.4 BRK2 Interrupt

The BRK2 interrupt occurs when the BRK2 instruction is executed.

Do not use this interrupt. For development support tools only.

### 10.2.5 INT Instruction Interrupt

The INT instruction interrupt occurs when the INT instruction is executed. The INT instruction can select software interrupt numbers 0 to 63. Software interrupt numbers 8 to 50, 52 to 54 and 57 are assigned to the vector table used for the peripheral function interrupt. Therefore, the microcomputer executes the same interrupt routine when the INT instruction is executed as when a peripheral function interrupt occurs.

When the INT instruction is executed, the FLG register and PC are saved to the stack. PC also stores the relocatable vector of specified software interrupt numbers. Where the stack is saved varies depending on a software interrupt number. ISP is selected as the stack for software interrupt numbers 0 to 31 (setting the U flag to "0"). SP, which is set before the INT instruction is executed, is selected as the stack for software interrupt numbers 32 to 63 (the U flag is not changed).

With the peripheral function interrupt, the FLG register is saved and the U flag is set to "0" (ISP select) when an interrupt request is acknowledged. With software interrupt numbers 32 to 50, 52 to 54 and 57, SP to be used varies depending on whether the interrupt is generated by the peripheral function interrupt request or by the INT instruction.

## 10.3 Hardware Interrupts

Special interrupts and peripheral function interrupts are available as hardware interrupts.

### 10.3.1 Special Interrupts

Special interrupts are non-maskable interrupts.

#### 10.3.1.1 $\overline{\text{NMI}}$ Interrupt

The  $\overline{\text{NMI}}$  interrupt occurs when a signal applied to the  $\overline{\text{NMI}}$  pin changes from a high-level ("H") signal to a low-level ("L") signal. Refer to **10.8 NMI Interrupt** for details.

#### 10.3.1.2 Watchdog Timer Interrupt

The watchdog timer interrupt occurs when a count source of the watchdog timer underflows. Refer to **11. Watchdog Timer** for details.

#### 10.3.1.3 Oscillation Stop Detection Interrupt

The oscillation stop detection interrupt occurs when the microcomputer detects a main clock oscillation stop. Refer to **8. Clock Generation Circuit** for details.

#### 10.3.1.4 Single-Step Interrupt

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

#### 10.3.1.5 Address Match Interrupt

The address match interrupt occurs immediately before executing an instruction that is stored into an address indicated by the RMAD<sub>i</sub> register (i=0 to 7) when the AIER<sub>i</sub> bit in the AIER register is set to "1" (address match interrupt enabled). Set the starting address of the instruction in the RMAD<sub>i</sub> register. The address match interrupt does not occur when a table data or addresses of the instruction other than the starting address, if the instruction has multiple addresses, is set. Refer to **10.10 Address Match Interrupt** for details.

### 10.3.2 Peripheral Function Interrupt

The peripheral function interrupt occurs when a request from the peripheral functions in the microcomputer is acknowledged. The peripheral function interrupts and software interrupt numbers 8 to 50, 52 to 54 and 57 for the INT instruction use the same interrupt vector table. The peripheral function interrupt is a maskable interrupt.

See **Table 10.2** about how the peripheral function interrupt occurs. Refer to the descriptions of each function for details.

## 10.4 High-Speed Interrupt

The high-speed interrupt executes an interrupt sequence in five cycles and returns from the interrupt in three cycles.

When the FSIT bit in the RLVL register is set to "1" (interrupt priority level 7 available for the high-speed interrupt), the ILVL2 to ILVL0 bits in the interrupt control registers can be set to "1112" (level 7) to use the high-speed interrupt.

Only one interrupt can be set as the high-speed interrupt. When using the high-speed interrupt, do not set multiple interrupts to interrupt priority level 7. Set the DMAII bit in the RLVL register to "0" (interrupt priority level 7 available for interrupts).

Set the starting address of the high-speed interrupt routine in the VCT register.

When the high-speed interrupt is acknowledged, the FLG register is saved into the SVF register and PC is saved into the SVP register. The program is executed from an address indicated by the VCT register.

Execute the FREIT instruction to return from the high-speed interrupt routine.

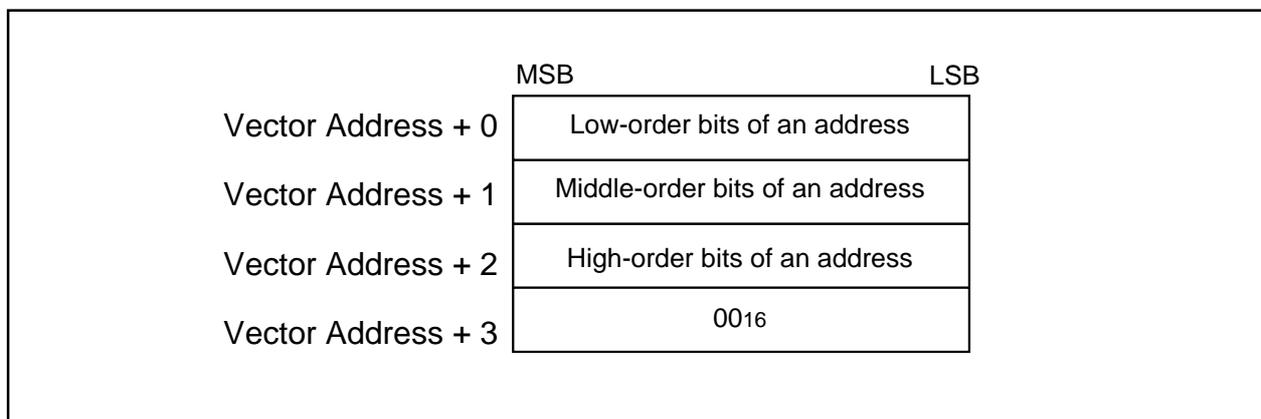
The values saved into the SVF and SVP registers are restored to the FLG register and PC by executing the FREIT instruction.

The high-speed interrupt and the DMA2 and DMA3 use the same register. When using the high-speed interrupt, neither DMA2 nor DMA3 is available. DMA0 and DMA1 can be used.

## 10.5 Interrupts and Interrupt Vectors

There are four bytes in one vector. Set the starting address of interrupt routine in each vector table. When an interrupt request is acknowledged, the interrupt routine is executed from the address set in the interrupt vectors.

Figure 10.2 shows the interrupt vector.



**Figure 10.2** Interrupt Vector

### 10.5.1 Fixed Vector Tables

The fixed vector tables are allocated addresses FFFFDC<sub>16</sub> to FFFFFFF<sub>16</sub>. Table 10.1 lists the fixed vector tables. Refer to **24.2 Functions to Prevent Rewriting of Flash Memory** for fixed vectors of flash memory.

**Table 10.1 Fixed Vector Table**

Interrupt Generated by	Vector Addresses Address (L) to Address (H)	Remarks	Reference
Undefined Instruction	FFFFDC <sub>16</sub> to FFFFDF <sub>16</sub>		M32C/80 Series Software Manual
Overflow	FFFFE0 <sub>16</sub> to FFFF E3 <sub>16</sub>		
BRK Instruction	FFFFE4 <sub>16</sub> to FFFF E7 <sub>16</sub>	If the content of address FFFF E7 <sub>16</sub> is FF <sub>16</sub> , a program is executed from the address stored into software interrupt number 0 in the relocatable vector table	
Address Match	FFFFE8 <sub>16</sub> to FFFF EB <sub>16</sub>		
-	FFFFEC <sub>16</sub> to FFFF EF <sub>16</sub>	Reserved space	
Watchdog Timer	FFFFF0 <sub>16</sub> to FFFF F3 <sub>16</sub>	These addresses are used for the watchdog timer interrupt and oscillation stop detection interrupt	Reset, Clock Generation Circuit, Watchdog Timer
-	FFFFF4 <sub>16</sub> to FFFF F7 <sub>16</sub>	Reserved space	
NMI	FFFFF8 <sub>16</sub> to FFFF FB <sub>16</sub>		
Reset	FFFFFC <sub>16</sub> to FFFF FF <sub>16</sub>		Reset

### 10.5.2 Relocatable Vector Tables

The relocatable vector tables occupy 256 bytes from the starting address set in the INTB register. Table 10.2 lists the relocatable vector tables.

Set an even address as the starting address of the vector table set in the INTB register to increase interrupt sequence execution rate.

**Table 10.2 Relocatable Vector Tables**

Interrupt Generated by	Vector Table Address Address(L) to Address(H) <sup>(1)</sup>	Software Interrupt Number	Reference
BRK Instruction <sup>(2)</sup>	+0 to +3 (0000 <sub>16</sub> to 0003 <sub>16</sub> )	0	M32C/80 Series
Reserved Space	+4 to +31 (0004 <sub>16</sub> to 001F <sub>16</sub> )	1 to 7	Software Manual
DMA0	+32 to +35 (0020 <sub>16</sub> to 0023 <sub>16</sub> )	8	DMAC
DMA1	+36 to +39 (0024 <sub>16</sub> to 0027 <sub>16</sub> )	9	
DMA2	+40 to +43 (0028 <sub>16</sub> to 002B <sub>16</sub> )	10	
DMA3	+44 to +47 (002C <sub>16</sub> to 002F <sub>16</sub> )	11	
Timer A0	+48 to +51 (0030 <sub>16</sub> to 0033 <sub>16</sub> )	12	Timer A
Timer A1	+52 to +55 (0034 <sub>16</sub> to 0037 <sub>16</sub> )	13	
Timer A2	+56 to +59 (0038 <sub>16</sub> to 003B <sub>16</sub> )	14	
Timer A3	+60 to +63 (003C <sub>16</sub> to 003F <sub>16</sub> )	15	
Timer A4	+64 to +67 (0040 <sub>16</sub> to 0043 <sub>16</sub> )	16	
UART0 Transmission, NACK <sup>(3)</sup>	+68 to +71 (0044 <sub>16</sub> to 0047 <sub>16</sub> )	17	Serial I/O
UART0 Reception, ACK <sup>(3)</sup>	+72 to +75 (0048 <sub>16</sub> to 004B <sub>16</sub> )	18	
UART1 Transmission, NACK <sup>(3)</sup>	+76 to +79 (004C <sub>16</sub> to 004F <sub>16</sub> )	19	
UART1 Reception, ACK <sup>(3)</sup>	+80 to +83 (0050 <sub>16</sub> to 0053 <sub>16</sub> )	20	
Timer B0	+84 to +87 (0054 <sub>16</sub> to 0057 <sub>16</sub> )	21	Timer B
Timer B1	+88 to +91 (0058 <sub>16</sub> to 005B <sub>16</sub> )	22	
Timer B2	+92 to +95 (005C <sub>16</sub> to 005F <sub>16</sub> )	23	
Timer B3	+96 to +99 (0060 <sub>16</sub> to 0063 <sub>16</sub> )	24	
Timer B4	+100 to +103 (0064 <sub>16</sub> to 0067 <sub>16</sub> )	25	
INT5	+104 to +107 (0068 <sub>16</sub> to 006B <sub>16</sub> )	26	Interrupt
INT4	+108 to +111 (006C <sub>16</sub> to 006F <sub>16</sub> )	27	
INT3	+112 to +115 (0070 <sub>16</sub> to 0073 <sub>16</sub> )	28	
INT2	+116 to +119 (0074 <sub>16</sub> to 0077 <sub>16</sub> )	29	
INT1	+120 to +123 (0078 <sub>16</sub> to 007B <sub>16</sub> )	30	
INT0	+124 to +127 (007C <sub>16</sub> to 007F <sub>16</sub> )	31	
Timer B5	+128 to +131 (0080 <sub>16</sub> to 0083 <sub>16</sub> )	32	Timer B
UART2 Transmission, NACK <sup>(3)</sup>	+132 to +135 (0084 <sub>16</sub> to 0087 <sub>16</sub> )	33	Serial I/O
UART2 Reception, ACK <sup>(3)</sup>	+136 to +139 (0088 <sub>16</sub> to 008B <sub>16</sub> )	34	
UART3 Transmission, NACK <sup>(3)</sup>	+140 to +143 (008C <sub>16</sub> to 008F <sub>16</sub> )	35	
UART3 Reception, ACK <sup>(3)</sup>	+144 to +147 (0090 <sub>16</sub> to 0093 <sub>16</sub> )	36	
UART4 Transmission, NACK <sup>(3)</sup>	+148 to +151 (0094 <sub>16</sub> to 0097 <sub>16</sub> )	37	
UART4 Reception, ACK <sup>(3)</sup>	+152 to +155 (0098 <sub>16</sub> to 009B <sub>16</sub> )	38	

**Table 10.2 Relocatable Vector Tables (Continued)**

Interrupt Generated by	Vector Table Address Address(L) to Address(H) <sup>(1)</sup>	Software Interrupt Number	Reference
Bus Conflict Detect, Start Condition Detect, Stop Condition Detect (UART2) <sup>(3)</sup> ,	+156 to +159 (009C <sub>16</sub> to 009F <sub>16</sub> )	39	Serial I/O
Bus Conflict Detect, Start Condition Detect, Stop Condition Detect (UART3/UART0) <sup>(4)</sup>	+160 to +163 (00A0 <sub>16</sub> to 00A3 <sub>16</sub> )	40	
Bus Conflict Detect, Start Condition Detect, Stop Condition Detect (UART4/UART1) <sup>(4)</sup>	+164 to +167 (00A4 <sub>16</sub> to 00A7 <sub>16</sub> )	41	
A/D0	+168 to +171 (00A8 <sub>16</sub> to 00AB <sub>16</sub> )	42	A/D Converter
Key Input	+172 to +175 (00AC <sub>16</sub> to 00AF <sub>16</sub> )	43	Interrupts
Intelligent I/O Interrupt 0, CAN 3	+176 to +179 (00B0 <sub>16</sub> to 00B3 <sub>16</sub> )	44	Intelligent I/O CAN
Intelligent I/O Interrupt 1, CAN 4	+180 to +183 (00B4 <sub>16</sub> to 00B7 <sub>16</sub> )	45	
Intelligent I/O Interrupt 2, CAN 6	+184 to +187 (00B8 <sub>16</sub> to 00BB <sub>16</sub> )	46	
Intelligent I/O Interrupt 3, CAN 7	+188 to +191 (00BC <sub>16</sub> to 00BF <sub>16</sub> )	47	
Intelligent I/O Interrupt 4	+192 to +195 (00C0 <sub>16</sub> to 00C3 <sub>16</sub> )	48	
CAN 5	+196 to +199 (00C4 <sub>16</sub> to 00C7 <sub>16</sub> )	49	CAN
CAN 8	+200 to +203 (00C4 <sub>16</sub> to 00C7 <sub>16</sub> )	50	
Reserved Space	+204 to +207 (00C8 <sub>16</sub> to 00CF <sub>16</sub> )	51	—
Intelligent I/O Interrupt 8	+208 to +211 (00D0 <sub>16</sub> to 00D3 <sub>16</sub> )	52	Intelligent I/O CAN
Intelligent I/O Interrupt 9, CAN 0	+212 to +215 (00D4 <sub>16</sub> to 00D7 <sub>16</sub> )	53	
Intelligent I/O Interrupt 10, CAN 1	+216 to +219 (00D8 <sub>16</sub> to 00DB <sub>16</sub> )	54	
Reserved Space	+220 to +227 (00DC <sub>16</sub> to 00E3 <sub>16</sub> )	55, 56	—
CAN 2	+228 to +231 (00E4 <sub>16</sub> to 00E7 <sub>16</sub> )	57	CAN
Reserved Space	+232 to +255 (00E8 <sub>16</sub> to 00FF <sub>16</sub> )	58 to 63	—
INT Instruction <sup>(2)</sup>	+0 to +3 (0000 <sub>16</sub> to 0003 <sub>16</sub> ) to +252 to +255 (00FC <sub>16</sub> to 00FF <sub>16</sub> )	0 to 63	Interrupts

## NOTES:

1. These addresses are relative to those in the INTB register.
2. The I flag does not disable interrupts.
3. In I<sup>2</sup>C mode, NACK, ACK or start/stop condition detection causes interrupts to be generated.
4. The IFSR6 bit in the IFSR register determines whether these addresses are used for an interrupt in UART0 or in UART3.  
The IFSR7 bit in the IFSR register determines whether these addresses are used for an interrupt in UART1 or in UART4.

## 10.6 Interrupt Request Acknowledgement

Software interrupts and special interrupts occur when conditions to generate an interrupt are met. The peripheral function interrupts are acknowledged when all conditions below are met.

- I flag = "1"
- IR bit = "1"
- ILVL2 to ILVL0 bits > IPL

The I flag, IPL, IR bit and ILVL2 to ILVL0 bits are independent of each other. The I flag and IPL are in the FLG register. The IR bit and ILVL2 to ILVL0 bits are in the interrupt control register.

### 10.6.1 I Flag and IPL

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

IPL, consisting of three bits, indicates the interrupt priority level from level 0 to level 7.

If a requested interrupt has higher priority level than indicated by IPL, the interrupt is acknowledged.

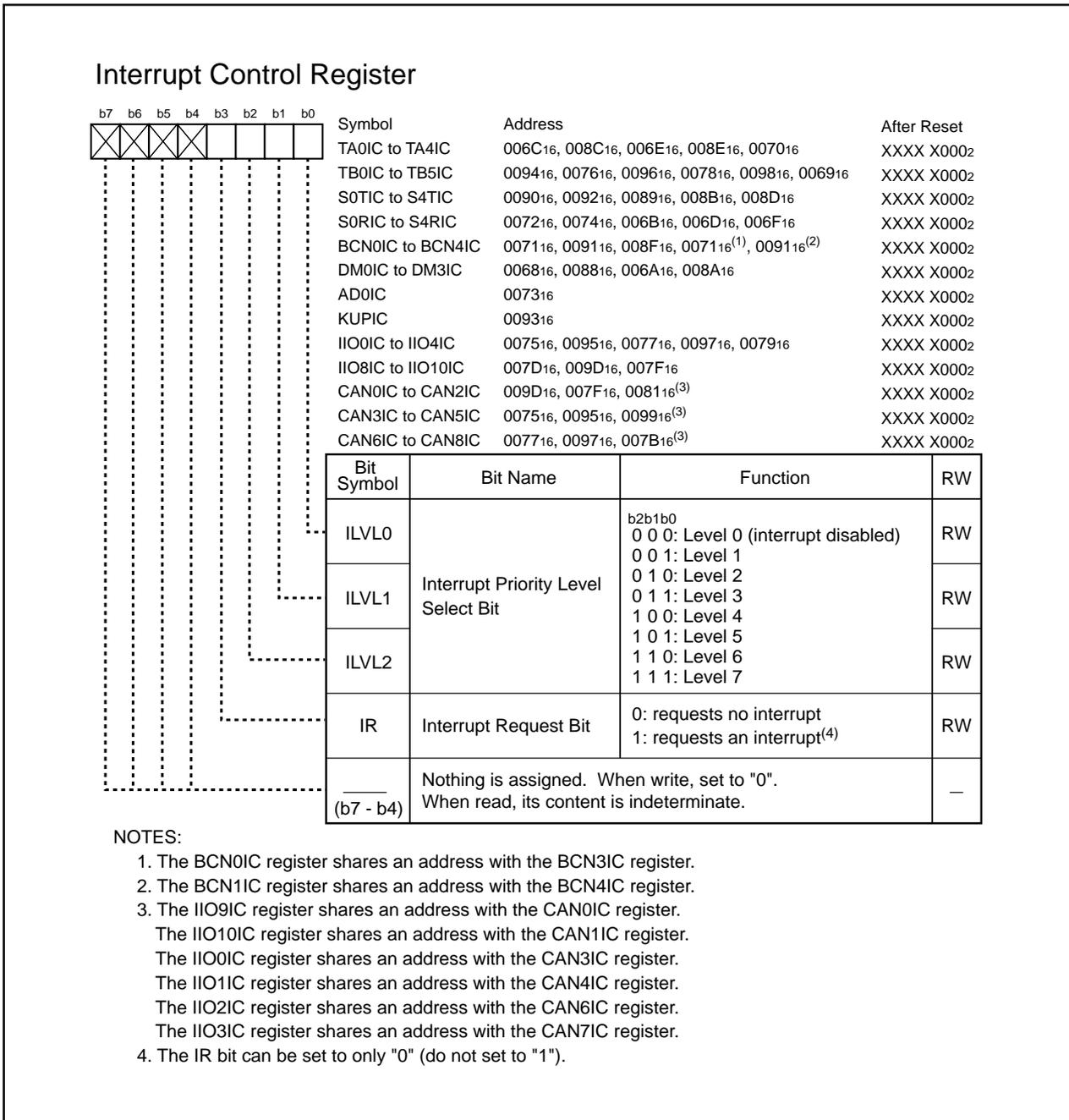
Table 10.3 lists interrupt priority levels associated with IPL.

**Table 10.3 Interrupt Priority Levels**

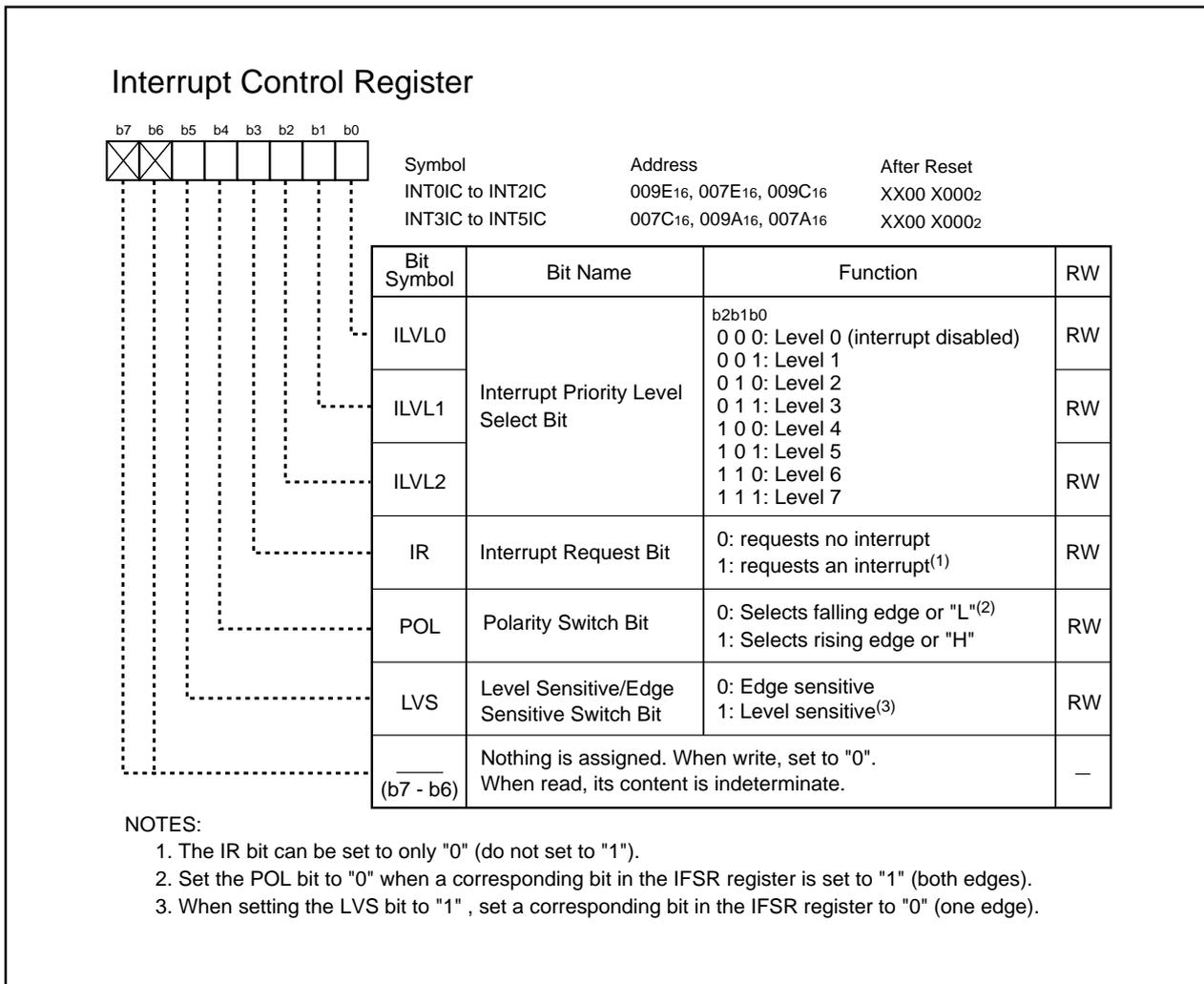
IPL2	IPL1	IPL0	Interrupt Priority Levels
0	0	0	Level 1 and above
0	0	1	Level 2 and above
0	1	0	Level 3 and above
0	1	1	Level 4 and above
1	0	0	Level 5 and above
1	0	1	Level 6 and above
1	1	0	Level 7 and above
1	1	1	All maskable interrupts are disabled

### 10.6.2 Interrupt Control Register and RLVL Register

The peripheral function interrupts use interrupt control registers to control each interrupt. Figures 10.3 and 10.4 show the interrupt control register. Figure 10.5 shows the RLVL register.



**Figure 10.3 Interrupt Control Register (1)**

**Figure 10.4 Interrupt Control Register (2)****10.6.2.1 ILVL2 to ILVL0 Bits**

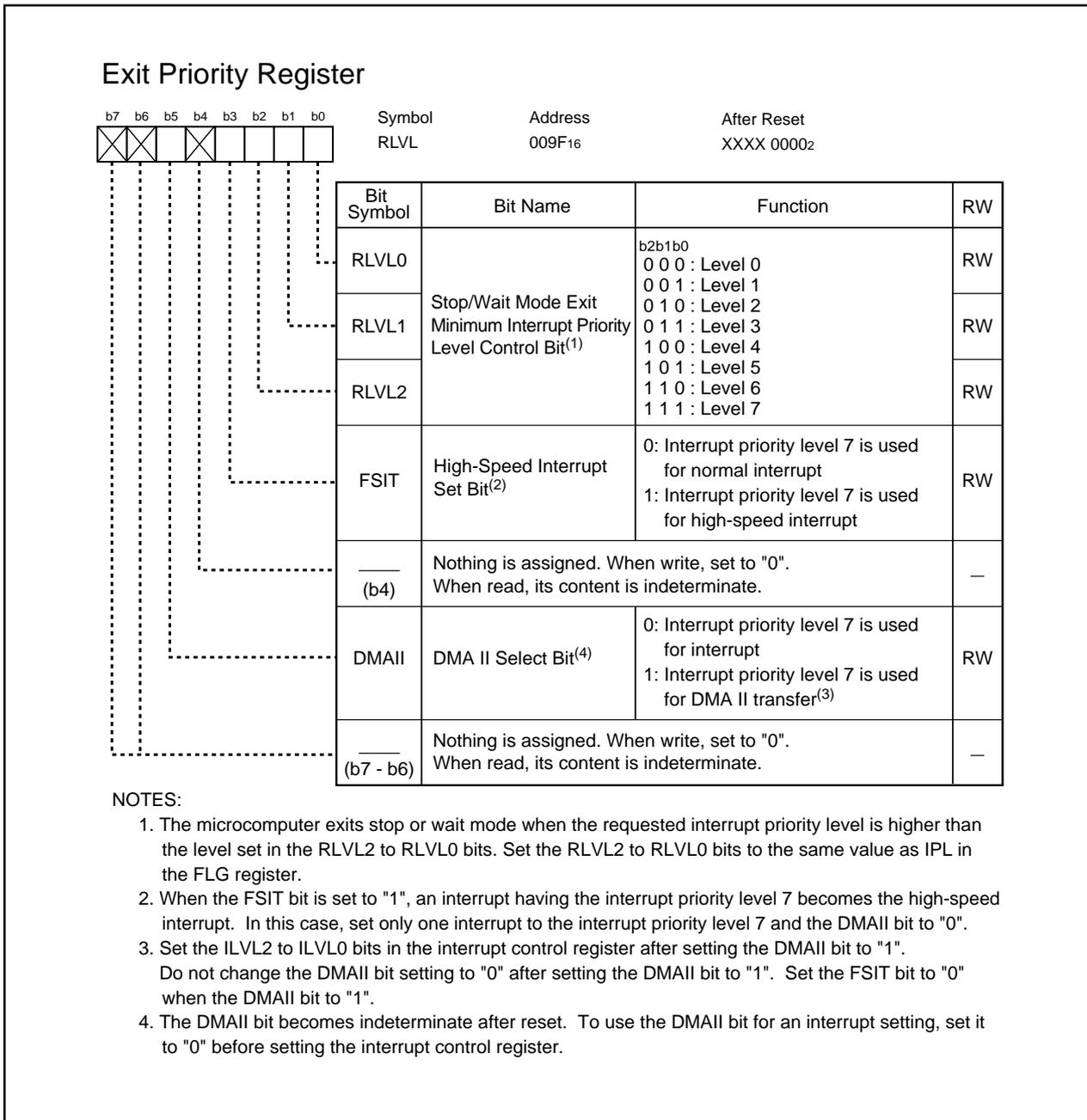
The ILVL2 to ILVL0 bits determines an interrupt priority level. The higher the interrupt priority level is, the higher interrupt priority is.

When an interrupt request is generated, its interrupt priority level is compared to IPL. This interrupt is acknowledged only when its interrupt priority level is higher than IPL. When the ILVL2 to ILVL0 bits are set to "0002" (level 0), its interrupt is ignored.

**10.6.2.2 IR Bit**

The IR bit is automatically set to "1" (interrupt requested) when an interrupt request is generated. The IR bit is automatically set to "0" (no interrupt requested) after an interrupt request is acknowledged and an interrupt routine in the corresponding interrupt vector is executed.

The IR bit can be set to "0" by program. Do not set to "1".



**Figure 10.5 RLVL Register**

**10.6.2.3 RLVL2 to RLVL0 Bits**

When using an interrupt to exit stop or wait mode, refer to **8.5.2 Wait Mode** and **8.5.3 Stop Mode** for details.

### 10.6.3 Interrupt Sequence

The interrupt sequence is performed between an interrupt request acknowledgment and interrupt routine execution.

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

The interrupt sequence is performed as follows:

- (1) The CPU obtains interrupt information (interrupt number and interrupt request level) by reading address 000000<sub>16</sub> (address 000002<sub>16</sub> for the high-speed interrupt). Then, the IR bit applicable to the interrupt information is set to "0" (interrupt requested).
- (2) The FLG register, prior to an interrupt sequence, is saved to a temporary register<sup>(1)</sup> within the CPU.
- (3) Each bit in the FLG register is set as follows:
  - The I flag is set to "0" (interrupt disabled)
  - The D flag is set to "0" (single-step disabled)
  - The U flag is set to "0" (ISP selected)
- (4) A temporary register within the CPU is saved to the stack; or to the SVF register for the high-speed interrupt.
- (5) PC is saved to the stack; or to the SVP register for the high-speed interrupt.
- (6) The interrupt priority level of the acknowledged interrupt is set in IPL .
- (7) A relocatable vector corresponding to the acknowledged interrupt is stored into PC.

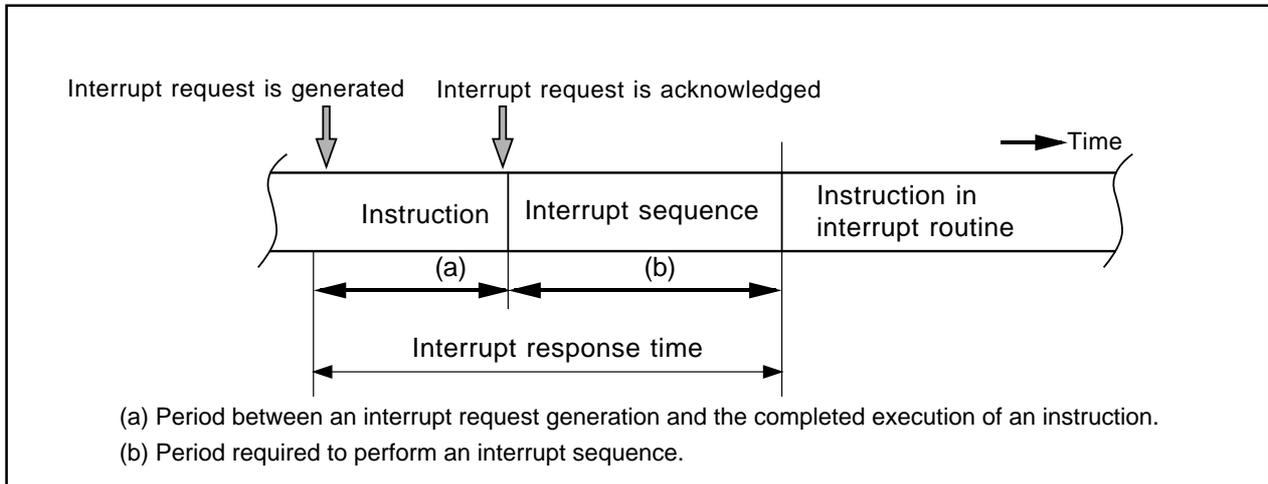
After the interrupt sequence is completed, an instruction is executed from the starting address of the interrupt routine.

**NOTE:**

1. Temporary register cannot be modified by users.

### 10.6.4 Interrupt Response Time

Figure 10.6 shows an interrupt response time. Interrupt response time is the period between an interrupt generation and the execution of the first instruction in an interrupt routine. Interrupt response time includes the period between an interrupt request generation and the completed execution of an instruction ((a) on Figure 10.6) and the period required to perform an interrupt sequence ((b) on Figure 10.6).



**Figure 10.6 Interrupt Response Time**

Time (a) varies depending on an instruction being executed. The DIV, DIVX and DIVU instructions require the longest time (a); 42 cycles when an immediate value or register is set as the divisor. When the divisor is a value in the memory, the following value is added.

- Normal addressing :  $2 + X$
- Index addressing :  $3 + X$
- Indirect addressing :  $5 + X + 2Y$
- Indirect index addressing :  $6 + X + 2Y$

X is the number of wait states for a divisor space. Y is the number of wait states for the space that stores indirect addresses. If X and Y are in an odd address or in 8-bit bus space, the X and Y value must be doubled.

Table 10.4 lists time (b), shown Figure 10.6.

**Table 10.4 Interrupt Sequence Execution Time**

Interrupt	Interrupt Vector Address	16-Bit Bus	8-Bit Bus
Peripheral Function	Even address	14 cycles	16 cycles
	Odd address <sup>(1)</sup>	16 cycles	16 cycles
INT Instruction	Even address	12 cycles	14 cycles
	Odd address <sup>(1)</sup>	14 cycles	14 cycles
NMI Watchdog Timer Undefined Instruction Address Match	Even address <sup>(2)</sup>	13 cycles	15 cycles
Overflow	Even address <sup>(2)</sup>	14 cycles	16 cycles
BRK Instruction (relocatable vector table)	Even address	17 cycles	19 cycles
	Odd address <sup>(1)</sup>	19 cycles	19 cycles
BRK Instruction (fixed vector table)	Even address <sup>(2)</sup>	19 cycles	21 cycles
High-Speed Interrupt	Vector table is internal register	5 cycles	

## NOTES:

1. Allocate interrupt vectors in even addresses.
2. Vectors are fixed to even addresses.

**10.6.5 IPL Change when Interrupt Request is Acknowledged**

When a peripheral function interrupt request is acknowledged, IPL sets the priority level for the acknowledged interrupt.

Software interrupts and special interrupts have no interrupt priority level. If an interrupt request that has no interrupt priority level is acknowledged, the value shown in Table 10.5 is set in IPL as the interrupt priority level.

**Table 10.5 Interrupts without Interrupt Priority Levels and IPL**

Interrupt Source	Level Set to IPL
Watchdog Timer, NMI, Oscillation Stop Detection	7
Reset	0
Software, Address Match	Not changed

### 10.6.6 Saving a Register

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

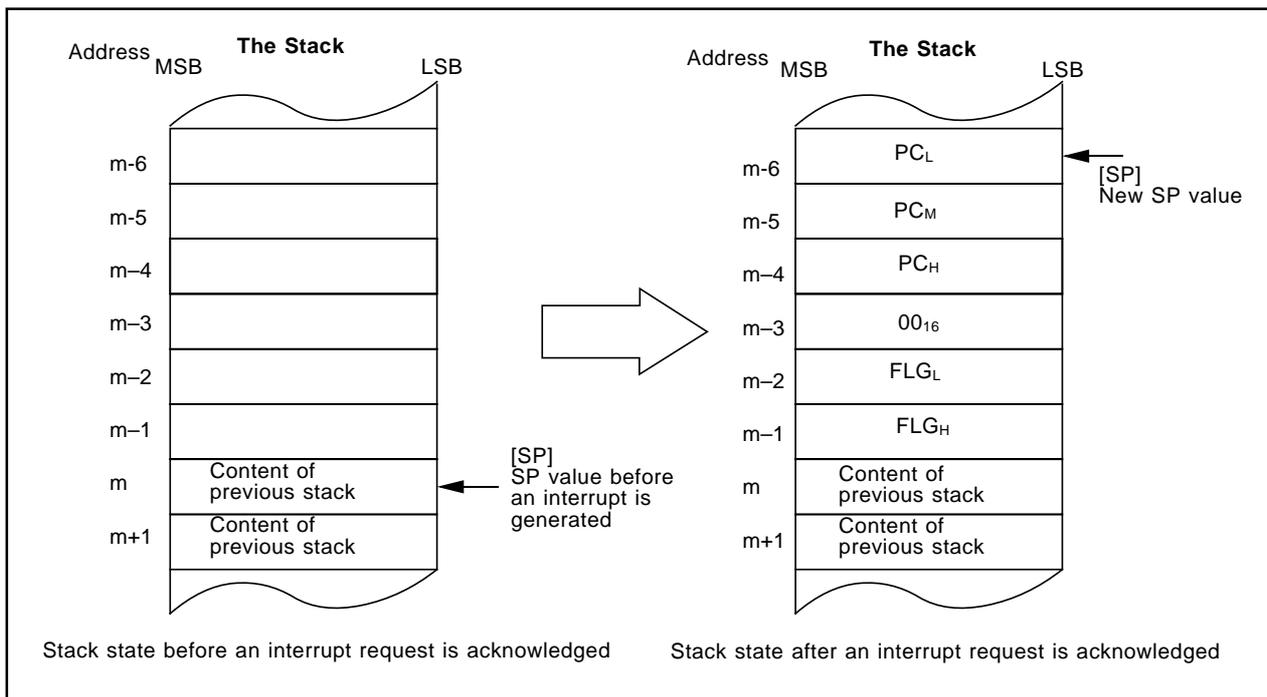
After the FLG register is saved to the stack, 16 high-order bits and 16 low-order bits of PC, extended to 32 bits, are saved to the stack. Figure 10.7 shows stack states before and after an interrupt request is acknowledged.

Other important registers are saved by program at the beginning of an interrupt routine. The PUSHM instruction can save several registers<sup>(1)</sup> in the register bank used.

Refer to **10.4 High-Speed Interrupt** for the high-speed interrupt.

#### NOTE:

1. Can be selected from the R0, R1, R2, R3, A0, A1, SB and FB registers.



**Figure 10.7 Stack States**

### 10.6.7 Restoration from Interrupt Routine

When the REIT instruction is executed at the end of an interrupt routine, the FLG register and PC before the interrupt sequence is performed, which have been saved to the stack, are automatically restored. The program, executed before an interrupt request was acknowledged, starts running again. Refer to **10.4 High-Speed Interrupt** for the high-speed interrupt.

Restore registers saved by program in an interrupt routine by the POPM instruction or others before the REIT and FREIT instructions. Register bank is switched back to the bank used prior to the interrupt sequence by the REIT or FREIT instruction.

### 10.6.8 Interrupt Priority

If two or more interrupt requests are sampled at the same sampling points (a timing to detect whether an interrupt request is generated or not), the interrupt with the highest priority is acknowledged.

Set the ILVL2 to ILVL0 bits to select the desired priority level for maskable interrupts (peripheral function interrupt).

Priority levels of special interrupts such as reset (reset has the highest priority) and watchdog timer are set by hardware. Figure 10.8 shows priority levels of hardware interrupts.

The interrupt priority does not affect software interrupts. Executing instruction causes the microcomputer to execute an interrupt routine.

<b>Reset</b> > <b><math>\overline{\text{NMI}}</math></b> > <b>Oscillation Stop Detection</b> <b>Watchdog Timer</b> > <b>Peripheral Function</b> > <b>Address Match</b>
---

Figure 10.8 Interrupt Priority

### 10.6.9 Interrupt Priority Level Select Circuit

The interrupt priority level select circuit selects the highest priority interrupt when two or more interrupt requests are sampled at the same sampling point.

Figure 10.9 shows the interrupt priority level select circuit.

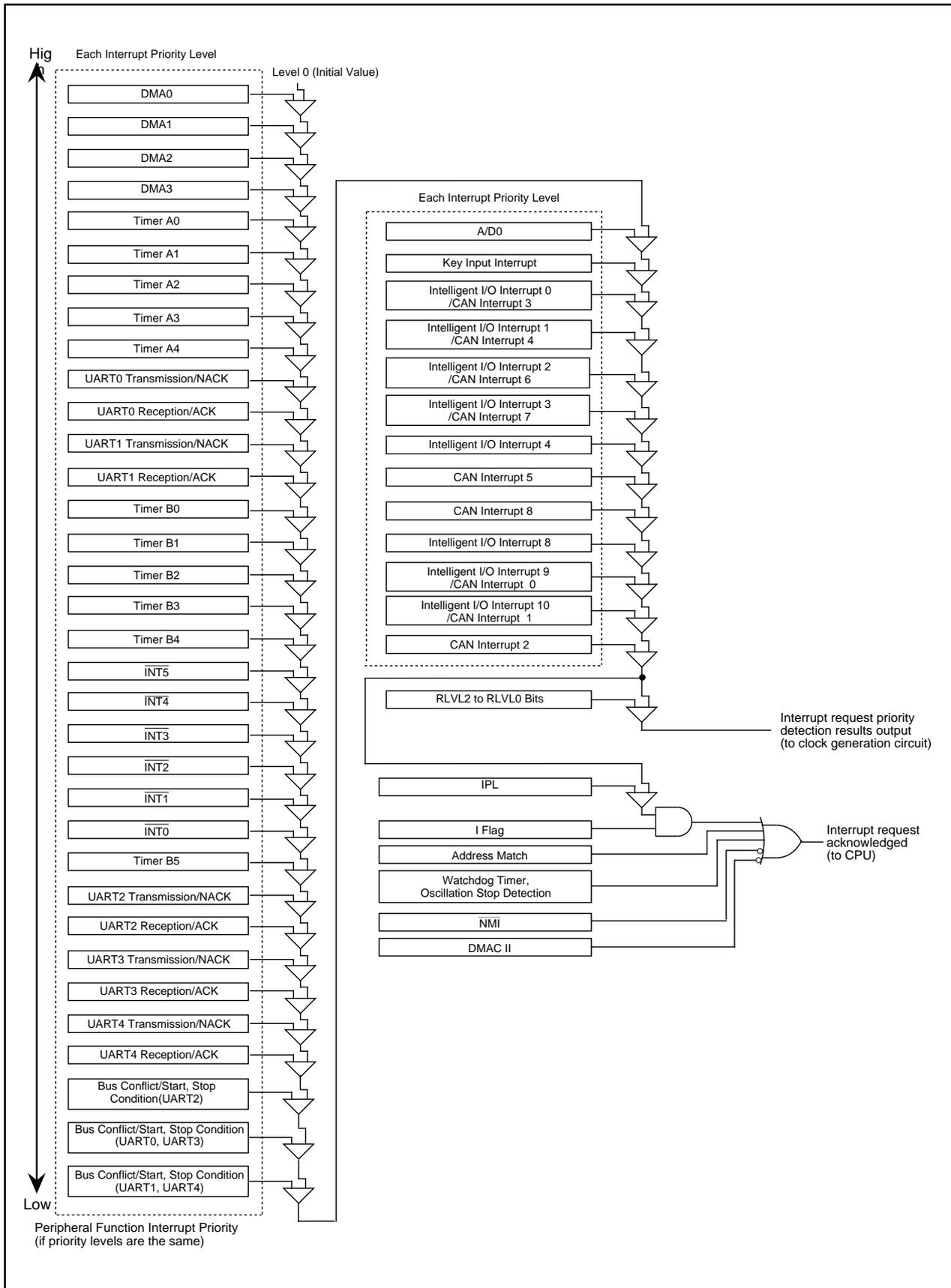


Figure 10.9 Interrupt Priority Level Select Circuit

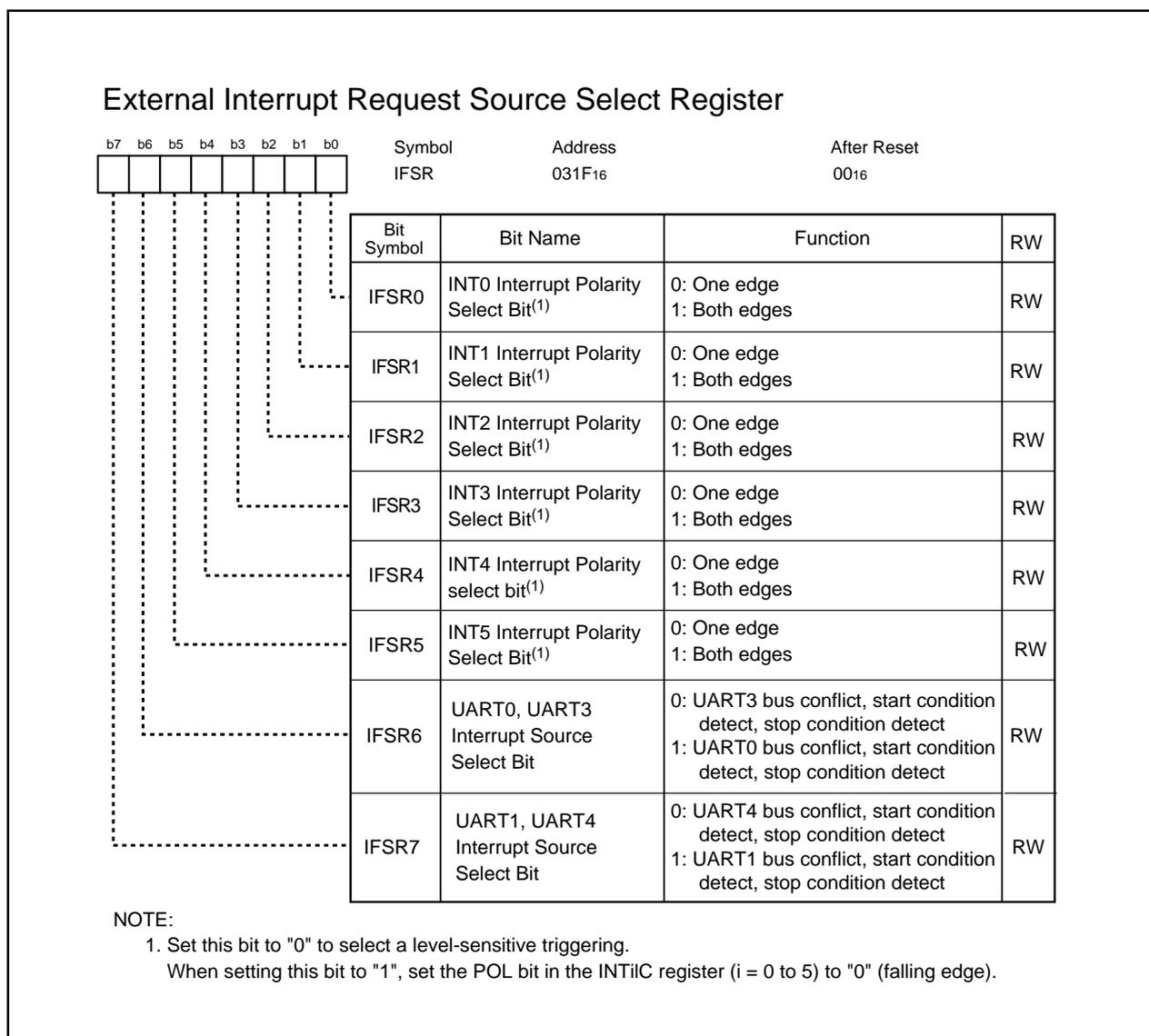
## 10.7 $\overline{\text{INT}}$ Interrupt

External input generates the  $\overline{\text{INT}}_i$  interrupt ( $i = 0$  to  $5$ ). The LVS bit in the  $\text{INTiIC}$  register selects either edge sensitive triggering to generate an interrupt on any edge or level sensitive triggering to generate an interrupt at an applied signal level. The POL bit in the  $\text{INTiIC}$  register determines the polarity.

For edge sensitive, when the  $\text{IFSR}_i$  bit in the  $\text{IFSR}$  register is set to "1", an interrupt occurs on both rising and falling edges of the external input. If the  $\text{IFSR}_i$  bit is set to "0", set the POL bit in the corresponding register to "0" (falling edge).

For level sensitive, set the  $\text{IFSR}_i$  bit to "0" (single edge). When the  $\overline{\text{INT}}_i$  pin input level reaches the level set in the POL bit, the IR bit in the  $\text{INTiIC}$  register is set to "1". The IR bit remains unchanged even if the  $\overline{\text{INT}}_i$  pin level is changed. The IR bit is set to "0" when the  $\overline{\text{INT}}_i$  interrupt is acknowledged or when the IR bit is written to "0" by program.

Figure 10.10 shows the  $\text{IFSR}$  register.



**Figure 10.10** IFSR Register

## 10.8 $\overline{\text{NMI}}$ Interrupt

The  $\overline{\text{NMI}}$  interrupt<sup>(1)</sup> occurs when a signal applied to the  $\overline{\text{NMI}}$  pin changes from a high-level ("H") signal to a low-level ("L") signal. The  $\overline{\text{NMI}}$  interrupt is a non-maskable interrupt. Although the P85/ $\overline{\text{NMI}}$  pin is used as the  $\overline{\text{NMI}}$  interrupt input pin, the P8\_5 bit in the P8 register indicates the input level for this pin.

### NOTE:

1. When the  $\overline{\text{NMI}}$  interrupt is not used, connect the  $\overline{\text{NMI}}$  pin to VCC via a resistor. Because the  $\overline{\text{NMI}}$  interrupt cannot be ignored, the pin must be connected.

## 10.9 Key Input Interrupt

Key input interrupt request is generated when one of the signals applied to the P104 to P107 pins in input mode is on the falling edge. The key input interrupt can be also used as key-on wake-up function to exit wait or stop mode. To use the key input interrupt, do not use P104 to P107 as A/D input ports. Figure 10.11 shows a block diagram of the key input interrupt. When an "L" signal is applied to any pins in input mode, signals applied to other pins are not detected as an interrupt request signal.

When the PSC\_7 bit in the PSC register<sup>(2)</sup> is set to "1" (key input interrupt disabled), no key input interrupt occurs regardless of interrupt control register settings. When the PSC\_7 bit is set to "0", no input from a port pin is available even when in input mode.

### NOTE:

2. Refer to **23. Programmable I/O Ports** about the PSC register.

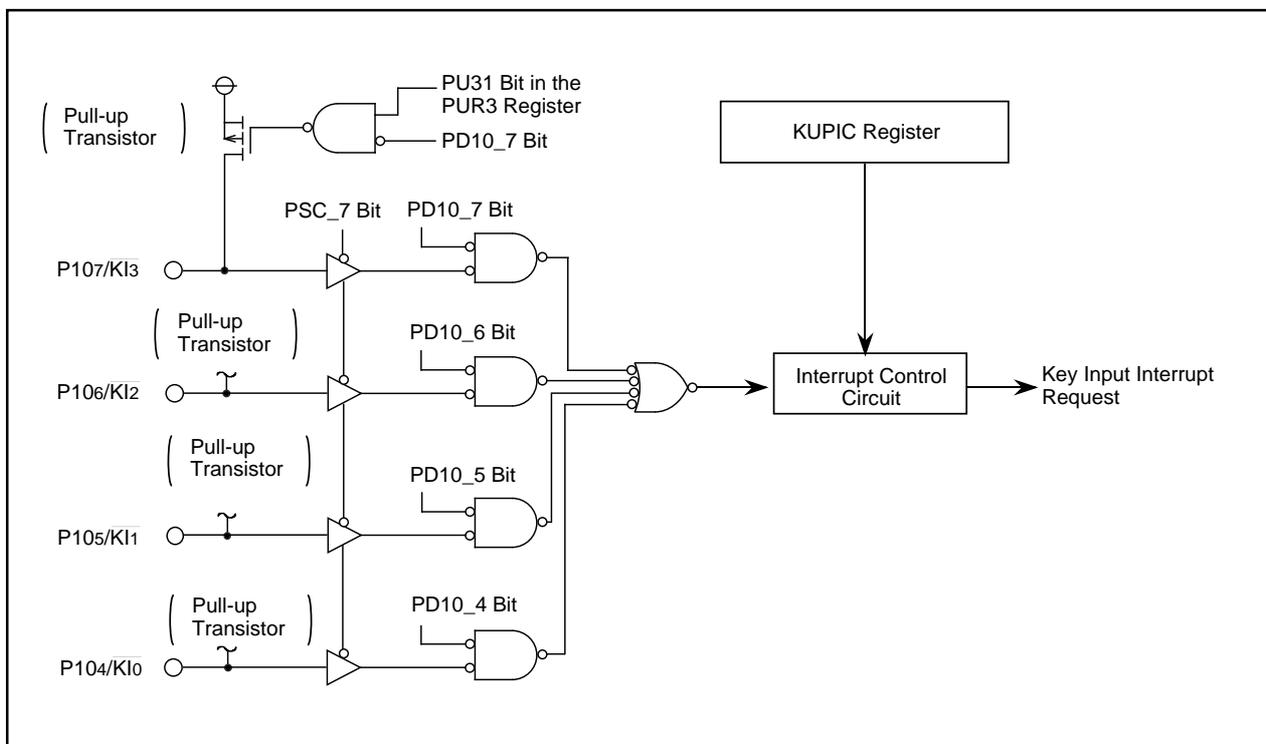


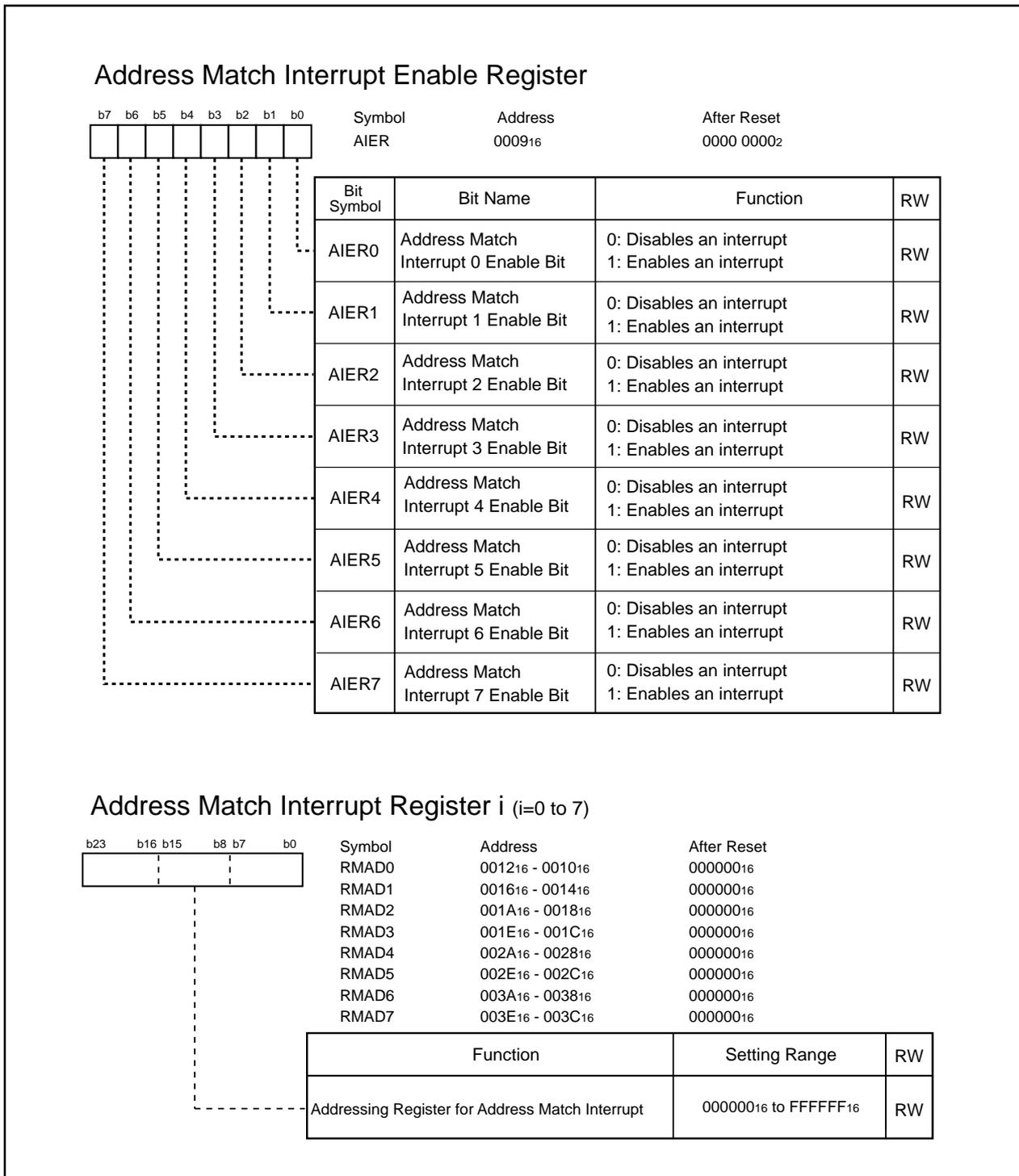
Figure 10.11 Key Input Interrupt

## 10.10 Address Match Interrupt

The address match interrupt occurs immediately before executing an instruction that is stored into an address indicated by the RMAD<sub>i</sub> register (i=0 to 7). The address match interrupt can be set in eight addresses. The AIER<sub>i</sub> bit in the AIER register determines whether the interrupt is enabled or disabled. The I flag and IPL do not affect the address match interrupt.

Figure 10.12 shows registers associated with the address match interrupt.

The starting address of an instruction must be set in the RMAD<sub>i</sub> register. The address match interrupt does not occur when a table data or addresses other than the starting address of the instruction is set.



**Figure 10.12 AIER Register and RMAD0 to RMAD7 Registers**

### 10.11 Intelligent I/O Interrupt and CAN Interrupt

The intelligent I/O interrupt and CAN interrupt are assigned to software interrupt numbers 44 to 50, 52 to 54, and 57.

When using the intelligent I/O interrupt or CAN interrupt, set the IRLT bit in the IIOiE register ( $i = 0$  to 6, 8 to 11) to "1" (interrupt request for interrupt used).

Various interrupt requests cause the intelligent I/O interrupt to occur. When an interrupt request is generated with each intelligent I/O or CAN functions, the corresponding bit in the IIOiR register is set to "1" (interrupt requested). When the corresponding bit in the IIOiE register is set to "1" (interrupt enabled), the IR bit in the corresponding IIOiC register is set to "1" (interrupt requested).

After the IR bit setting changes "0" to "1", the IR bit remains set to "1" when a bit in the IIOiR register is set to "1" by another interrupt request and the corresponding bit in the IIOiE register is set to "1".

Bits in the IIOiR register are not set to "0" automatically, even if an interrupt is acknowledged. Set each bit to "0" by program. If these bit settings are left "1", all generated interrupt requests are ignored.

Figure 10.13 shows a block diagram of the intelligent I/O interrupt and CAN interrupt. Figure 10.14 shows the IIOiR register. Figure 10.15 shows the IIOiE register.

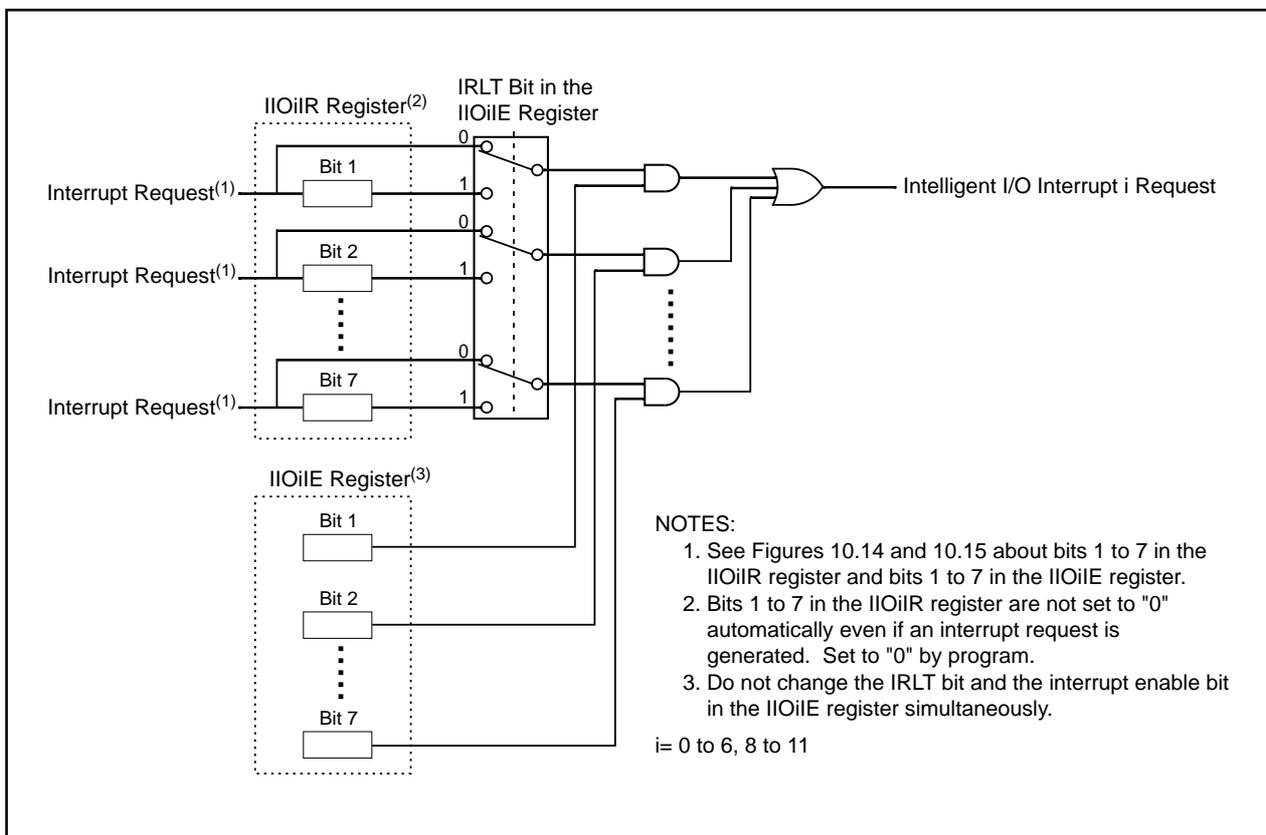


Figure 10.13 Intelligent I/O Interrupt and CAN Interrupt

The CAN<sub>jk</sub> (j=0 to 2, k=0 to 2) interrupt and CAN<sub>n</sub> (n=1, 2) wake-up interrupt are provided as the CAN interrupt. The following registers are required for the CAN interrupts:

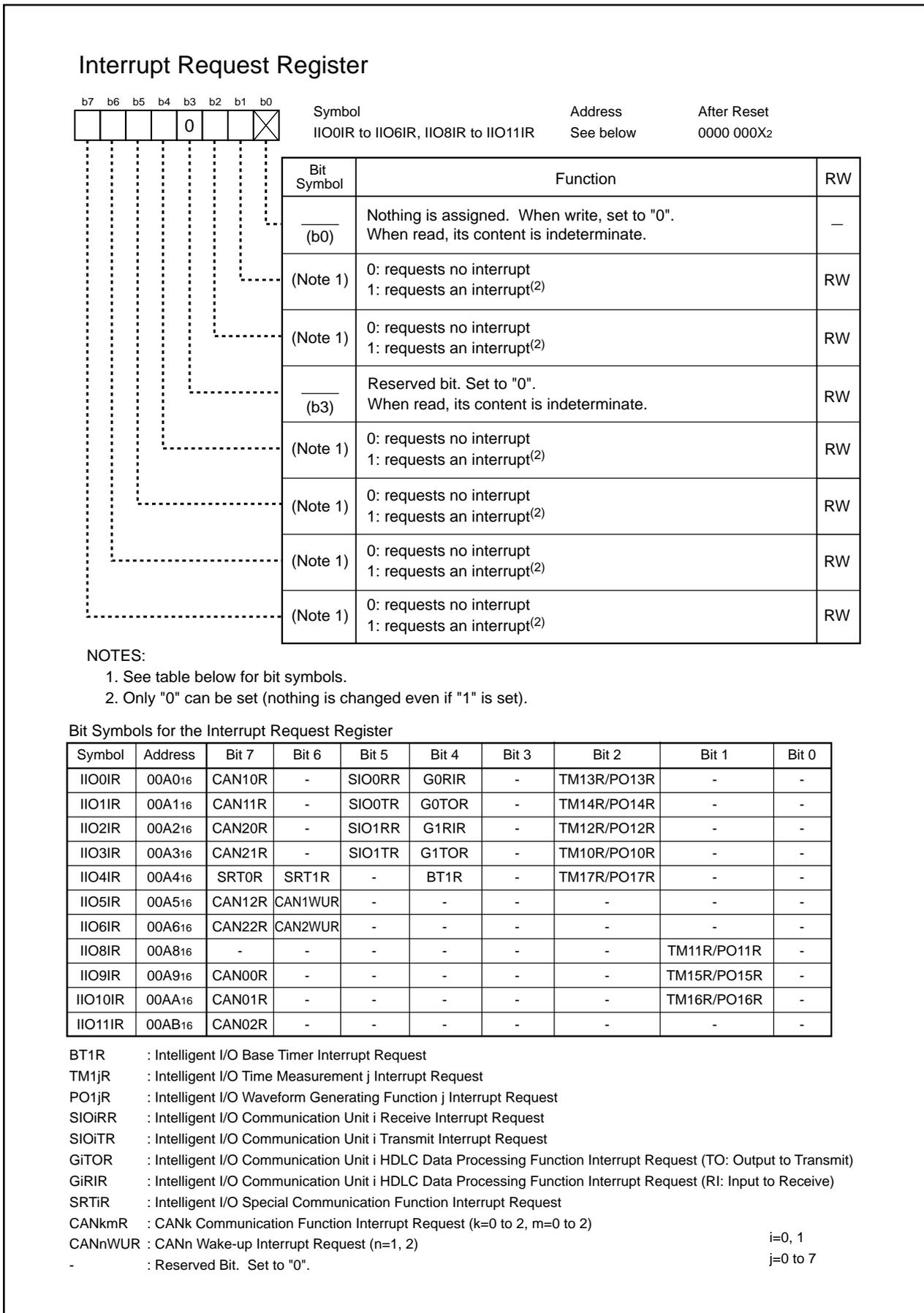
- Bits 7 in the IIO9IR to IIO11IR registers and bits 7 in the IIO9IE to IIO11IE registers for the CAN00 to CAN02 interrupts.
- Bits 7 in the IIO0IR, IIO1IR and IIO5IR registers and bits 7 in the IIO0IE, IIO1IE and IIO5IE registers for the CAN10 to CAN12 interrupts.
- Bits 7 in the IIO2IR, IIO3IR and IIO6IR registers and bits 7 in the IIO2IE, IIO3IE and IIO6IE registers for the CAN20 to CAN22 interrupts.
- Bit 6 in the IIO5IR register and bit 6 in the IIO5IE register for the CAN1 wake-up interrupt.
- Bit 6 in the IIO6IR register and bit 6 in the IIO6IE register for the CAN2 wake-up interrupt.

The CAN0IC, CAN1IC, CAN3IC, CAN4IC, CAN6IC, and CAN7IC registers share addresses with the following registers:

- The CAN0IC register shares an address with the IIO9IC register.
- The CAN1IC register shares an address with the IIO10IC register.
- The CAN3IC register shares an address with the IIO0IC register.
- The CAN4IC register shares an address with the IIO1IC register.
- The CAN6IC register shares an address with the IIO2IC register.
- The CAN7IC register shares an address with the IIO3IC register.

Refer to **22.4 CAN Interrupt** for details.

When using the intelligent I/O interrupt or CAN interrupt to activate DMAC II, set the IRLT bit in the IIOiIE register to "0" (interrupt used for DMAC, DMAC II) to enable the interrupt request that the IIOiIE (i=0 to 6, 8 to 11) register requires.



**Figure 10.14 IIO0IR to IIO6IR, IIO8IR to IIO11IR Registers**

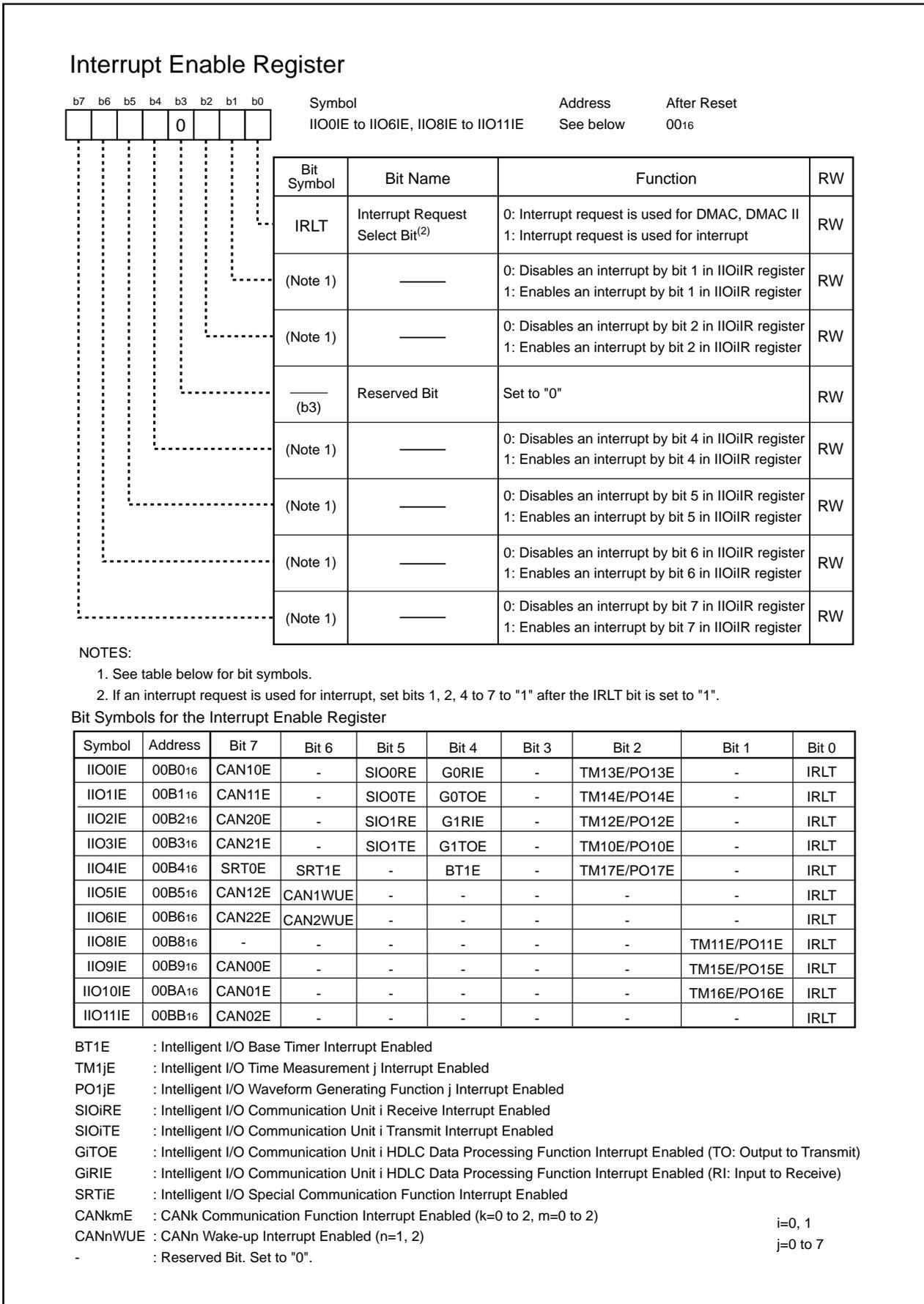


Figure 10.15 IIO01E to IIO61E, IIO81E to IIO111E Registers

# 11. Watchdog Timer

The watchdog timer monitors the program executions and detects defective program. It allows the microcomputer to trigger a reset or to generate an interrupt if the program error occurs. The watchdog timer contains a 15-bit counter, which is decremented by the CPU clock that the prescaler divides. The CM06 bit in the CM0 register determines whether a watchdog timer interrupt request or reset is generated if the watchdog timer underflows. The CM06 bit can only be set to "1" (reset). Once the CM06 bit is set to "1", it cannot be changed to "0" ( watchdog timer interrupt) by program. The CM06 bit is set to "0" only after reset. When the main clock, on-chip oscillator clock, or PLL clock runs as the CPU clock, the WDC7 bit in the WDC register determine whether the prescaler divides the clock by 16 or by 128. When the sub clock runs as the CPU clock, the prescaler divides the clock by 2 regardless of the WDC7 bit setting. Watchdog timer cycle is calculated as follows. Marginal errors, due to the prescaler, may occur in watchdog timer cycle.

When the main clock, on-chip oscillator clock, or PLL clock is selected as the CPU clock,

$$\text{Watchdog timer cycle} = \frac{\text{Divide-by-16 or -128 prescaler} \times \text{counter value of watchdog timer (32768)}}{\text{CPU clock}}$$

When the sub clock is selected as the CPU clock,

$$\text{Watchdog timer cycle} = \frac{\text{Divide-by-2 prescaler} \times \text{counter value of watchdog timer (32768)}}{\text{CPU clock}}$$

For example, if the CPU clock frequency is 30MHz and the prescaler divides it by 16, the watchdog timer cycle is approximately 17.5 ms.

The watchdog timer is reset when the WDTS register is set and when a watchdog timer interrupt request is generated. The prescaler is reset only when the microcomputer is reset. Both watchdog timer and prescaler stop after reset. They begin counting when the WDTS register is set.

The watchdog timer and prescaler stop in stop mode, wait mode and hold state. They resume counting from the value held when the mode or state is exited.

Figure 11.1 shows a block diagram of the watchdog timer. Figure 11.2 shows registers associated with the watchdog timer.

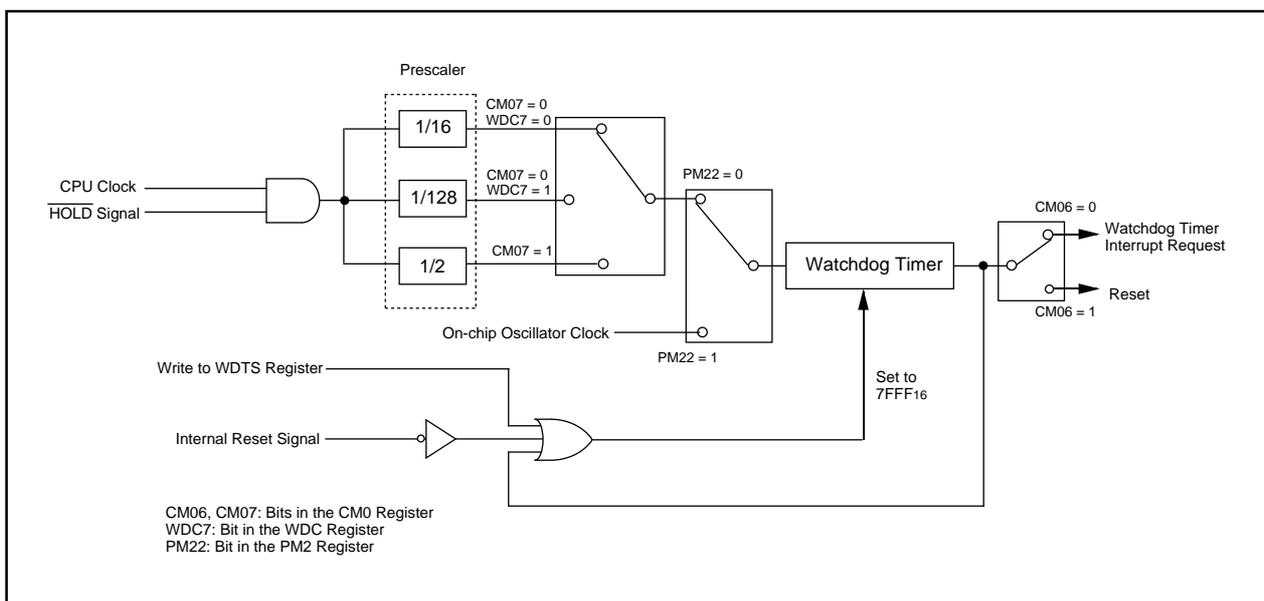


Figure 11.1 Watchdog Timer Block Diagram

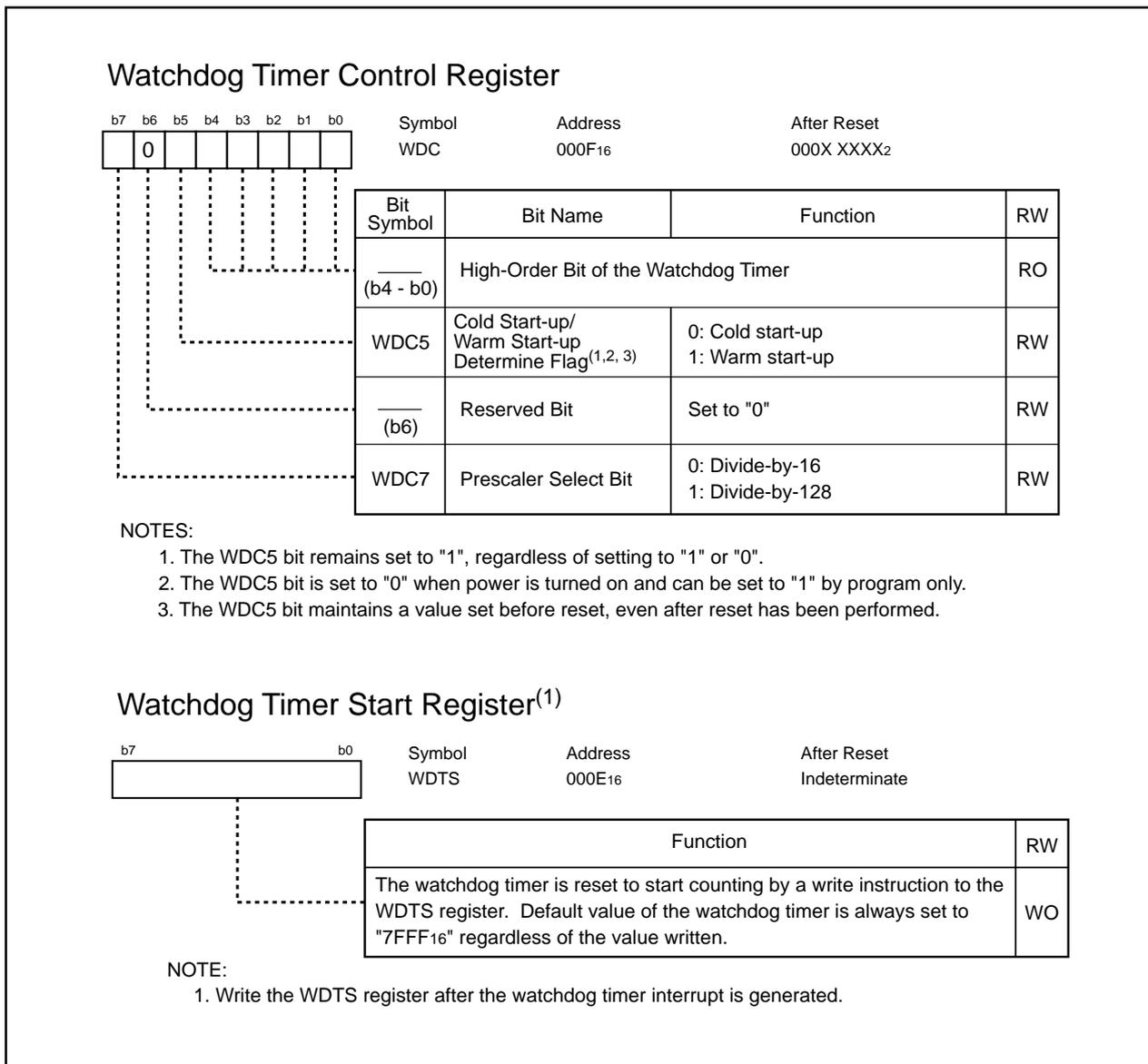


Figure 11.2 WDC Register and WDTS Register

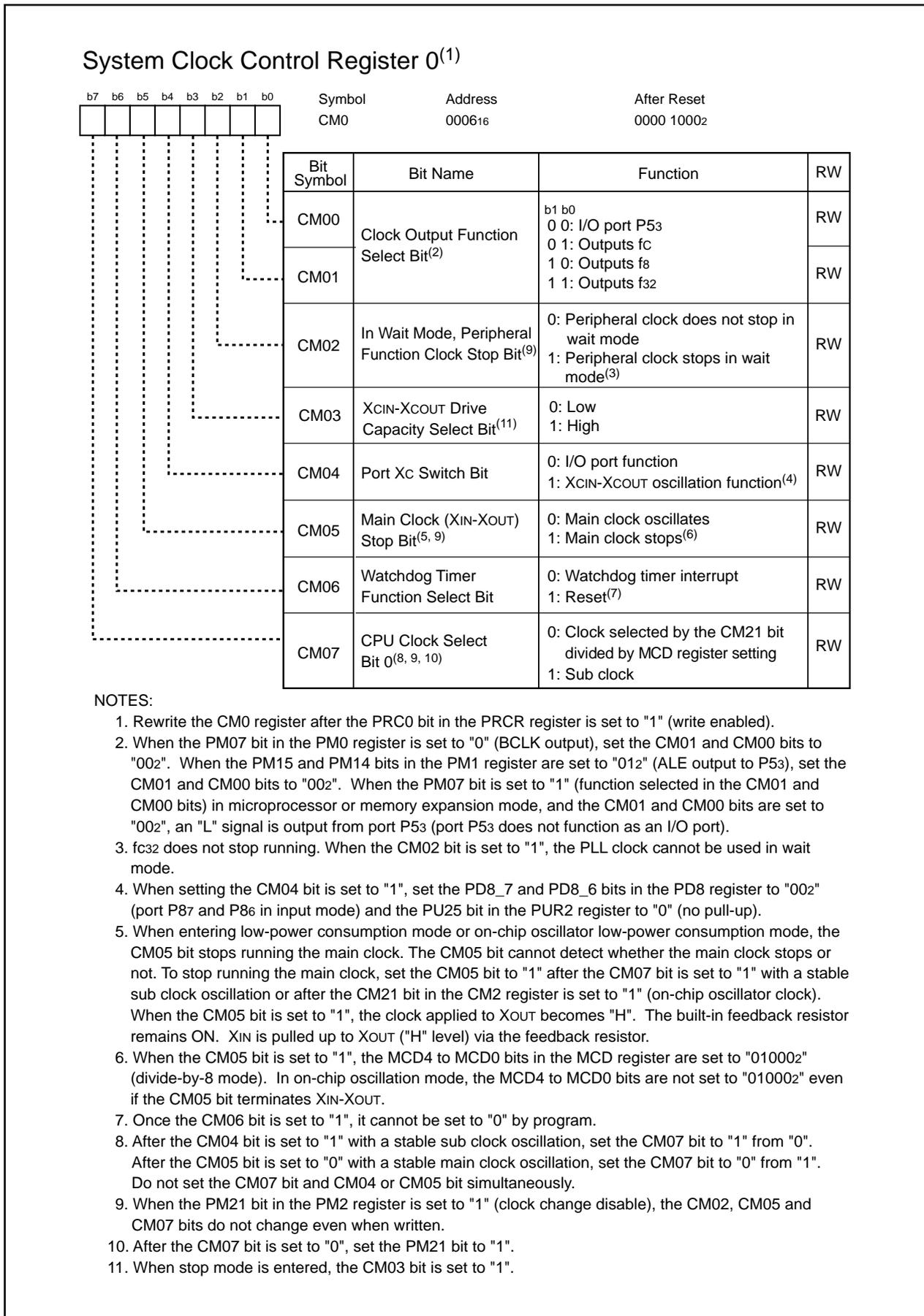


Figure 11.3 CM0 Register

## 11.1 Count Source Protection Mode

In count source protection mode, the on-chip oscillator clock is used as a count source for the watchdog timer. The count source protection mode allows the on-chip oscillator clock to run continuously, maintaining watchdog timer operation even if the program error occurs and the CPU clock stops running.

Follow the procedures below when using this mode.

- (1) Set the PRC0 bit in the PRCR register to "1" (write to CM0 register enabled)
- (2) Set the PRC1 bit in the PRCR register to "1" (write to PM2 register enabled)
- (3) Set the CM06 bit in the CM0 register to "1" (reset when the watchdog timer overflows)
- (4) Set the PM22 bit in the PM2 register to "1" (the on-chip oscillator clock as a count source of the watchdog timer)
- (5) Set the PRC0 bit to "0" (write to CM0 register disabled)
- (6) Set the PRC1 bit to "0" (write to PM2 register disabled)
- (7) Write to the WDTS register (the watchdog timer starts counting)

The followings will occur when the PM22 bit is set to "1".

- The on-chip oscillator starts oscillating and the on-chip oscillator clock becomes a count source for the watchdog timer.

$$\text{Watchdog timer cycle} = \frac{\text{Counter value of watchdog timer (32768)}}{\text{On-chip oscillator clock}}$$

- Write to the CM10 bit in the CM1 register is disabled. (The bit setting remains unchanged even if set it to "1". The microcomputer does not enter stop mode.)
- In wait mode or hold state, the watchdog timer continues running. However, the watchdog timer interrupt cannot be used to exit wait mode.

## 12. DMAC

This microcomputer contains four DMAC (direct memory access controller) channels that allow data to be sent to memory without using the CPU. DMAC transmits a 8- or 16-bit data from a source address to a destination address whenever a transmit request occurs. DMA0 and DMA1 must be prioritized if using DMAC. DMA2 and DMA3 share registers required for high-speed interrupts. High-speed interrupts cannot be used when using three or more DMAC channels.

The CPU and DMAC use the same data bus, but DMAC has a higher bus access privilege than the CPU. The cycle-steal method employed on DMAC enables high-speed operation between a transfer request and the complete transmission of 16-bit (word) or 8-bit (byte) data. Figure 12.1 shows a mapping of registers to be used for DMAC. Table 12.1 lists specifications of DMAC. Figures 12.2 to 12.5 show registers associated with DMAC.

Because the registers shown in Figure 12.1 are allocated in the CPU, use the LDC instruction to write to the registers. To set the DCT2, DCT3, DRC2, DRC3, DMA2 and DMA3 registers, set the B flag to "1" (register bank 1) and set the R0 to R3, A0, A1 registers with the MOV instruction.

To set the DSA2 and DSA3 registers, set the B flag to "1" and set the SB and FB registers with the LDC instruction. To set the DRA2 and DRA3 registers, set the SVP and VCT registers with the LDC instruction.

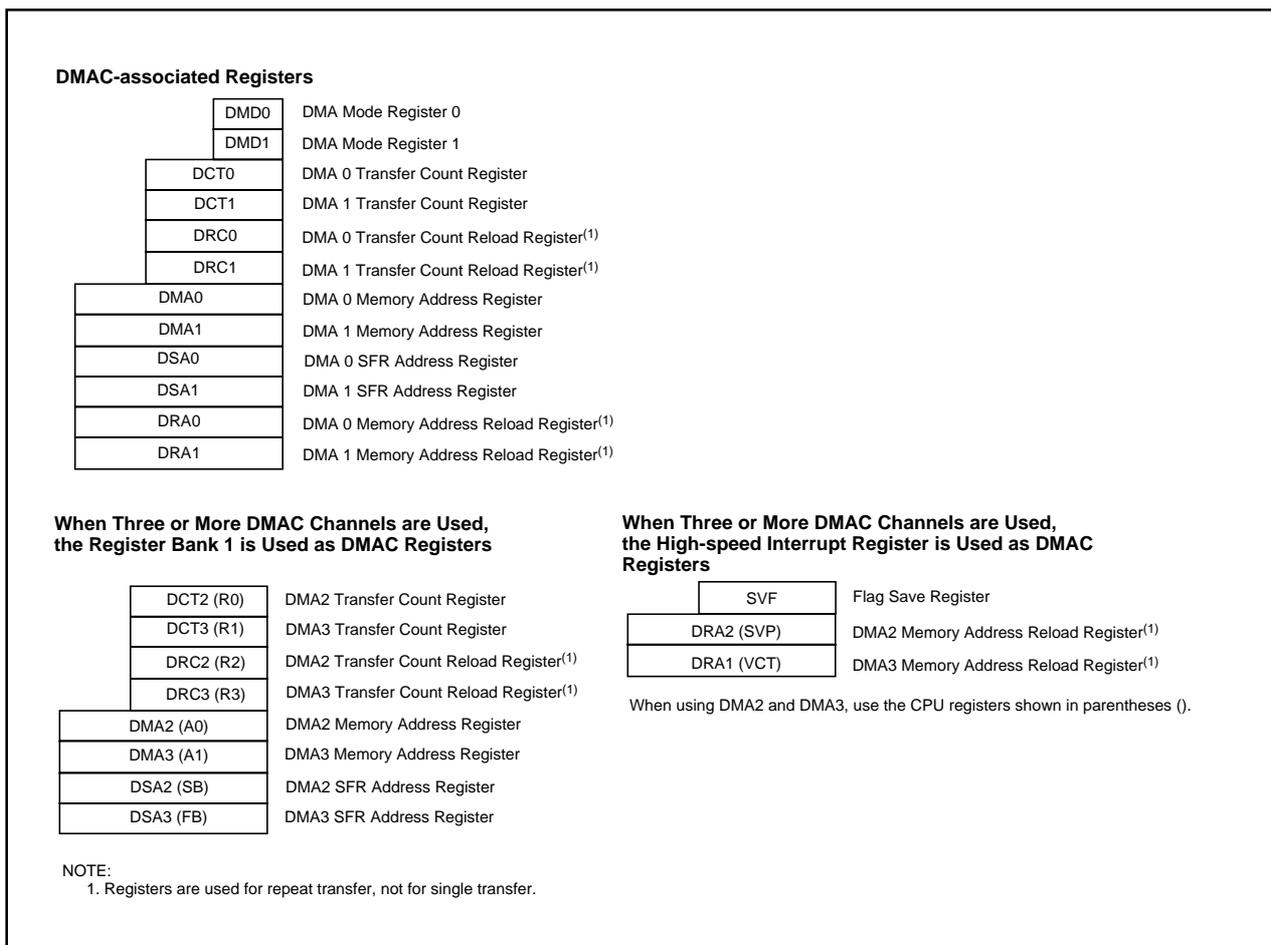


Figure 12.1 Register Mapping for DMAC

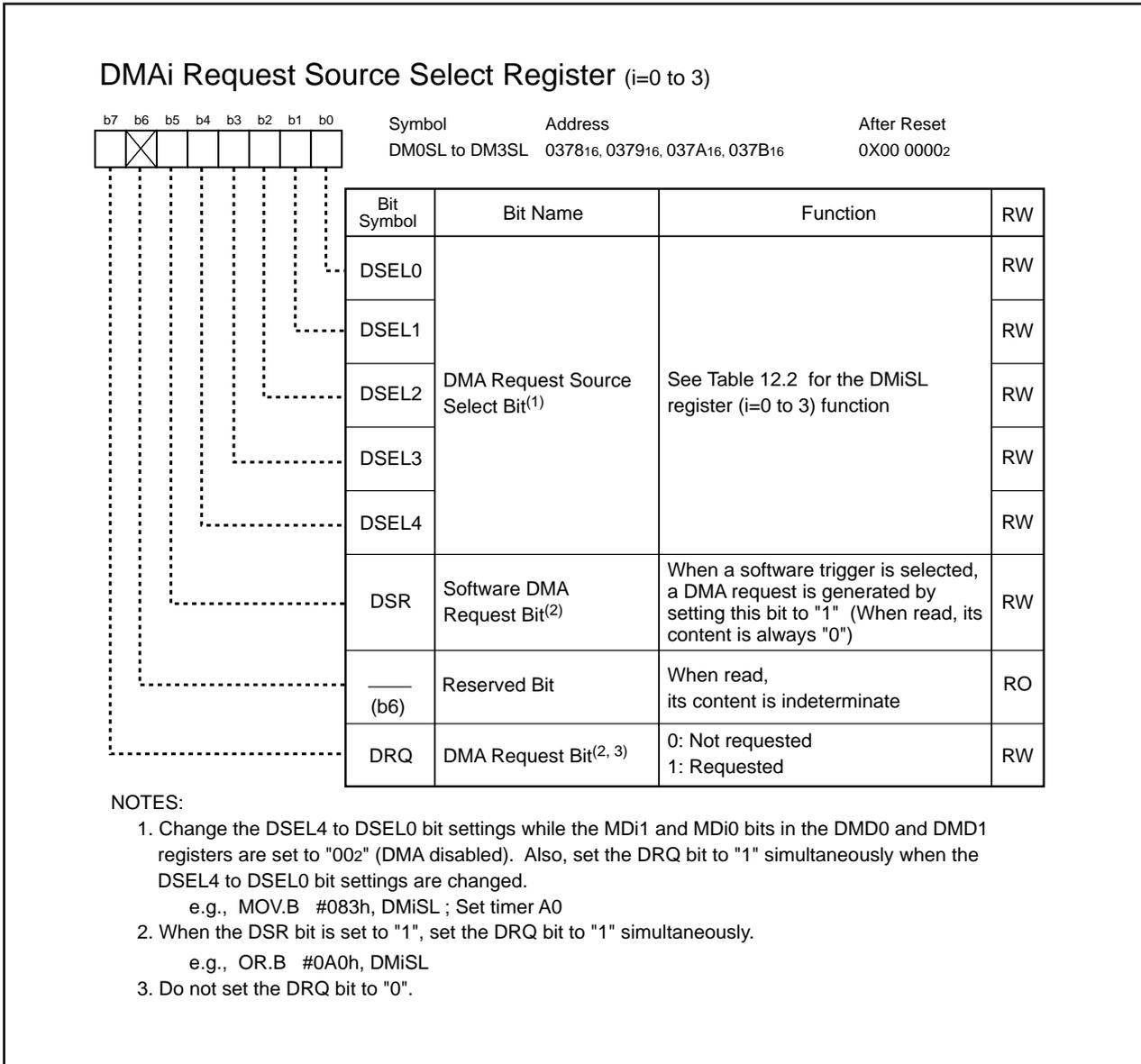
DMAC starts a data transfer by setting the DSR bit in the DMiSL register (i=0 to 3) or by using an interrupt request, generated by the functions determined by the DSEL 4 to DSEL0 bits in the DMiSL register, as a DMA request. Unlike interrupt requests, the I flag and interrupt control register do not affect DMA. Therefore, a DMA request can be acknowledged even if an interrupt is disabled and cannot be acknowledged. In addition, the IR bit in the interrupt control register does not change when a DMA request is acknowledged.

**Table 12.1 DMAC Specifications**

Item		Specification
Channels		4 channels (cycle-steal method)
Transfer Memory Space		<ul style="list-style-type: none"> <li>From a desired address in a 16-Mbyte space to a fixed address in a 16-Mbyte space</li> <li>From a fixed address in a 16-Mbyte space to a desired address in a 16-Mbyte space</li> </ul>
Maximum Bytes Transferred		128 Kbytes (when a 16-bit data is transferred) or 64 Kbytes (with an 8-bit data is transferred)
DMA Request Source <sup>(1)</sup>		Falling edge or both edges of signals applied to the INT0 to INT3 pins Timers A0 to A4 interrupt requests Timers B0 to B5 interrupt requests UART0 to UART4 transmit and receive interrupt requests A/D0 conversion interrupt request Intelligent I/O interrupt request CAN interrupt request Software trigger
Channel Priority		DMA0 > DMA1 > DMA2 > DMA3 (DMA0 has highest priority)
Transfer Unit		8 bits, 16 bits
Destination Address		Forward/fixed (forward and fixed directions cannot be specified when specifying source and destination addresses simultaneously)
Transfer Mode	Single Transfer	Transfer is completed when the DCTi register (i = 0 to 3) is set to "0000 <sub>16</sub> "
	Repeat Transfer	When the DCTi register is set to "0000 <sub>16</sub> ", the value of the DRCi register is reloaded into the DCTi register and the DMA transfer is continued
DMA Interrupt Request Generation Timing		When the DCTi register changes "0001 <sub>16</sub> " to "0000 <sub>16</sub> "
DMA Startup	Single Transfer	DMA starts when a DMA request is generated after the DCTi register is set to "0001 <sub>16</sub> " or more and the MDi1 and MDi0 bits in the DMDj register (j = 0,1) are set to "012" (single transfer)
	Repeat Transfer	DMA starts when a DMA request is generated after the DCTi register is set to "0001 <sub>16</sub> " or more and the MDi1 and MDi0 bits are set to "112" (repeat transfer)
DMA Stop	Single Transfer	DMA stops when the MDi1 and MDi0 bits are set to "002" (DMA disabled) and the DCTi register is set to "0000 <sub>16</sub> " (0 DMA transfer) by DMA transfer or write
	Repeat Transfer	DMA stops when the MDi1 and MDi0 bits are set to "002" and the DCTi register is set to "0000 <sub>16</sub> " and the DRCi register set to "0000 <sub>16</sub> "
Reload Timing to the DCTi or DMAi Register		When the DCTi register is set to "0000 <sub>16</sub> " from "0001 <sub>16</sub> " in repeat transfer mode
DMA Transfer Cycles		Minimum 3 cycles between SFRs and internal RAM

NOTE:

1. The IR bit in the interrupt control register does not change when a DMA request is acknowledged.



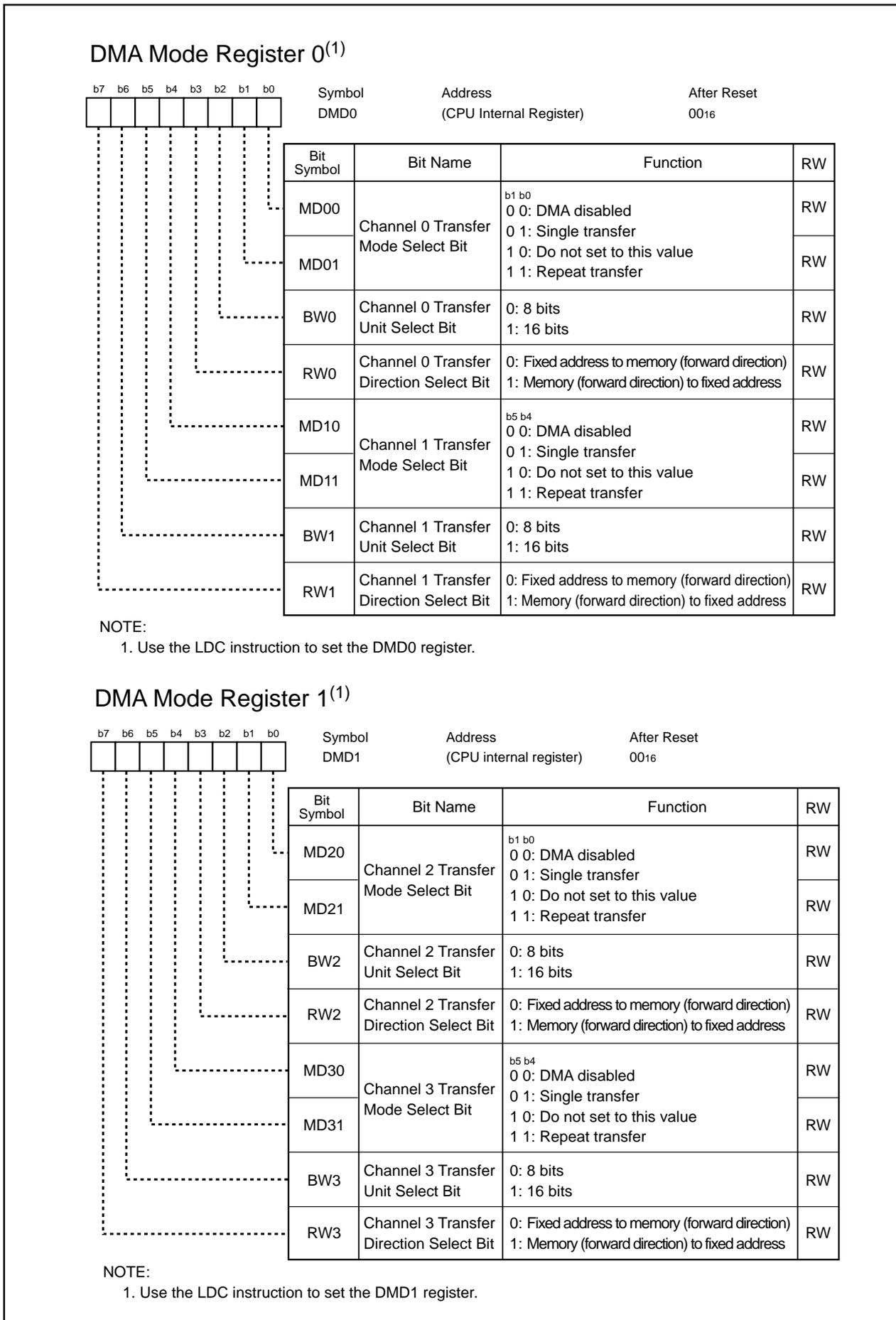
**Figure 12.2 DM0SL to DM3SL Registers**

**Table 12.2 DMiSL Register (i = 0 to 3) Function**

Setting Value	DMA Request Source			
b4 b3 b2 b1 b0	DMA0	DMA1	DMA2	DMA3
0 0 0 0 0	Software trigger			
0 0 0 0 1	Falling Edge of $\overline{INT0}$	Falling Edge of $\overline{INT1}$	Falling Edge of $\overline{INT2}$	Falling Edge of $\overline{INT3}^{(1)}$ (Note 2)
0 0 0 1 0	Both Edges of $\overline{INT0}$	Both Edges of $\overline{INT1}$	Both Edges of $\overline{INT2}$	Both Edges of $\overline{INT3}^{(1)}$ (Note 2)
0 0 0 1 1	Timer A0 Interrupt Request			
0 0 1 0 0	Timer A1 Interrupt Request			
0 0 1 0 1	Timer A2 Interrupt Request			
0 0 1 1 0	Timer A3 Interrupt Request			
0 0 1 1 1	Timer A4 Interrupt Request			
0 1 0 0 0	Timer B0 Interrupt Request			
0 1 0 0 1	Timer B1 Interrupt Request			
0 1 0 1 0	Timer B2 Interrupt Request			
0 1 0 1 1	Timer B3 Interrupt Request			
0 1 1 0 0	Timer B4 Interrupt Request			
0 1 1 0 1	Timer B5 Interrupt Request			
0 1 1 1 0	UART0 Transmit Interrupt Request			
0 1 1 1 1	UART0 Receive or ACK Interrupt Request <sup>(3)</sup>			
1 0 0 0 0	UART1 Transmit Interrupt Request			
1 0 0 0 1	UART1 Receive or ACK Interrupt Request <sup>(3)</sup>			
1 0 0 1 0	UART2 Transmit Interrupt Request			
1 0 0 1 1	UART2 Receive or ACK Interrupt Request <sup>(3)</sup>			
1 0 1 0 0	UART3 Transmit Interrupt Request			
1 0 1 0 1	UART3 Receive or ACK Interrupt Request <sup>(3)</sup>			
1 0 1 1 0	UART4 Transmit Interrupt Request			
1 0 1 1 1	UART4 Receive or ACK Interrupt Request <sup>(3)</sup>			
1 1 0 0 0	A/D0 Interrupt Request			
1 1 0 0 1	Intelligent I/O Interrupt 0 Request <sup>(6)</sup>	—	Intelligent I/O Interrupt 2 Request	Intelligent I/O Interrupt 9 Request <sup>(4)</sup>
1 1 0 1 0	Intelligent I/O Interrupt 1 Request <sup>(7)</sup>	Intelligent I/O Interrupt 8 Request	Intelligent I/O Interrupt 3 Request	Intelligent I/O Interrupt 10 Request <sup>(5)</sup>
1 1 0 1 1	Intelligent I/O Interrupt 2 Request <sup>(8)</sup>	Intelligent I/O Interrupt 9 Request <sup>(4)</sup>	Intelligent I/O Interrupt 4 Request	CAN Interrupt 2 Request
1 1 1 0 0	Intelligent I/O Interrupt 3 Request <sup>(9)</sup>	Intelligent I/O Interrupt 10 Request <sup>(5)</sup>	CAN Interrupt 5 Request	Intelligent I/O Interrupt 0 Request <sup>(6)</sup>
1 1 1 0 1	Intelligent I/O Interrupt 4 Request	CAN Interrupt 2 Request	CAN Interrupt 8 Request	Intelligent I/O Interrupt 1 Request <sup>(7)</sup>
1 1 1 1 0	CAN Interrupt 5 Request	Intelligent I/O Interrupt 0 Request <sup>(6)</sup>	—	Intelligent I/O Interrupt 2 Request <sup>(6)</sup>
1 1 1 1 1	CAN Interrupt 8 Request	Intelligent I/O Interrupt 1 Request <sup>(7)</sup>	Intelligent I/O Interrupt 8 Request	Intelligent I/O Interrupt 3 Request <sup>(9)</sup>

## NOTES:

1. If the  $\overline{INT3}$  pin is used for data bus in memory expansion mode or microprocessor mode, a DMA3 interrupt request cannot be generated by a signal applied to the  $\overline{INT3}$  pin.
2. The falling edge and both edges of signals applied to the  $\overline{INTj}$  pin (j=0 to 3) cause a DMA request generation. The  $\overline{INT}$  interrupt (the POL bit in the INTjIC register, the LVS bit, the IFSR register) is not affected and vice versa.
3. Use the UkSMR register and UkSMR2 register (k=0 to 4) to switch between the UARTk receive and ACK interrupt as a DMA request source.  
To use the ACK interrupt for a DMA request, set the IICM bit in the UkSMR register to "1" and the IICM2 bit in the UkSMR2 register to "0".
4. The same setting is used to generate an intelligent I/O interrupt 9 request and a CAN interrupt 0 request.
5. The same setting is used to generate an intelligent I/O interrupt 10 request and a CAN interrupt 1 request.
6. The same setting is used to generate an intelligent I/O interrupt 0 request and a CAN interrupt 3 request.
7. The same setting is used to generate an intelligent I/O interrupt 1 request and a CAN interrupt 4 request.
8. The same setting is used to generate an intelligent I/O interrupt 2 request and a CAN interrupt 6 request.
9. The same setting is used to generate an intelligent I/O interrupt 3 request and a CAN interrupt 7 request.



**Figure 12.3 DMD0 and DMD1 Registers**

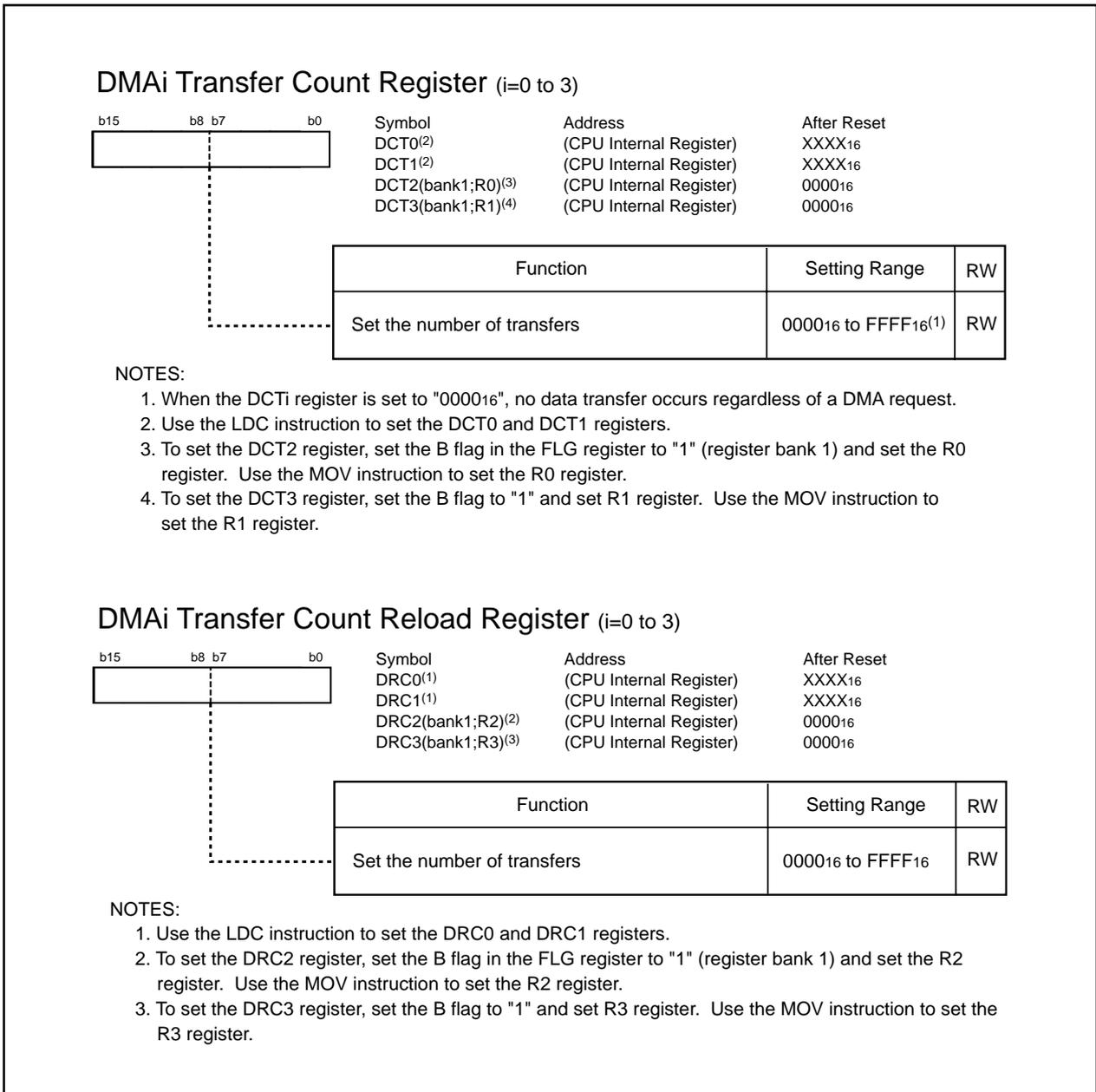


Figure 12.4 DCT0 to DCT3 Registers and DRC0 to DRC3 Registers

## DMAi Memory Address Register (i=0 to 3)

b23	b16 b15	b8 b7	b0	Symbol	Address	After Reset
[Register Bit Fields]				DMA0 <sup>(2)</sup>	(CPU Internal Register)	XXXXXX <sub>16</sub>
[Register Bit Fields]				DMA1 <sup>(2)</sup>	(CPU Internal Register)	XXXXXX <sub>16</sub>
[Register Bit Fields]				DMA2(bank1;A0) <sup>(3)</sup>	(CPU Internal Register)	000000 <sub>16</sub>
[Register Bit Fields]				DMA3(bank1;A1) <sup>(4)</sup>	(CPU Internal Register)	000000 <sub>16</sub>

Function	Setting Range	RW
Set source memory address or destination memory address <sup>(1)</sup>	000000 <sub>16</sub> to FFFFFFF <sub>16</sub> (16-Mbyte space)	RW

## NOTES:

- When the RWk bit (k=0 to 3) in the DMDj register (j=0, 1) is set to "0" (fixed address to memory), a destination address is selected. When the RWk bit is set to "1" (memory to fixed address), a source address is selected.
- Use the LDC instruction to set the DMA0 and DMA1 registers.
- To set the DMA2 register, set the B flag in the FLG register to "1" (register bank 1) and set the A0 register. Use the MOV instruction to set the A0 register.
- To set the DMA3 register, set the B flag to "1" and set the A1 register. Use the MOV instruction to set the A1 register.

## DMAi SFR Address Register (i=0 to 3)

b23	b16 b15	b8 b7	b0	Symbol	Address	After Reset
[Register Bit Fields]				DSA0 <sup>(2)</sup>	(CPU Internal Register)	XXXXXX <sub>16</sub>
[Register Bit Fields]				DSA1 <sup>(2)</sup>	(CPU Internal Register)	XXXXXX <sub>16</sub>
[Register Bit Fields]				DSA2(bank1;SB) <sup>(3)</sup>	(CPU Internal Register)	000000 <sub>16</sub>
[Register Bit Fields]				DSA3(bank1;FB) <sup>(4)</sup>	(CPU Internal Register)	000000 <sub>16</sub>

Function	Setting Range	RW
Set source fixed address or destination fixed address <sup>(1)</sup>	000000 <sub>16</sub> to FFFFFFF <sub>16</sub> (16-Mbyte space)	RW

## NOTES:

- When the RWk bit (k=0 to 3) in the DMDj register (j=0, 1) is set to "0" (fixed address to memory), a source address is selected. When the RWk bit is set to "1" (memory to fixed address), a destination address is selected.
- Use the LDC instruction to set the DSA0 and DSA1 registers.
- To set the DSA2 register, set the B flag in the FLG register to "1" (register bank 1) and the set the SB register. Use the LDC instruction to set the SB register.
- To set the DSA3 register, set the B flag to "1" and set the FB register. Use the LDC instruction to set the PB register.

DMAi Memory Address Reload Register<sup>(1)</sup> (i=0 to 3)

b23	b16 b15	b8 b7	b0	Symbol	Address	After Reset
[Register Bit Fields]				DRA0	(CPU Internal Register)	XXXXXX <sub>16</sub>
[Register Bit Fields]				DRA1	(CPU Internal Register)	XXXXXX <sub>16</sub>
[Register Bit Fields]				DRA2(SVP) <sup>(2)</sup>	(CPU Internal Register)	XXXXXX <sub>16</sub>
[Register Bit Fields]				DRA3(VCT) <sup>(3)</sup>	(CPU Internal Register)	XXXXXX <sub>16</sub>

Function	Setting Range	RW
Set source memory address or destination memory address <sup>(1)</sup>	000000 <sub>16</sub> to FFFFFFF <sub>16</sub> (16-Mbyte space)	RW

## NOTES:

- Use the LDC instruction to set the DRA0 and DRA1 registers.
- To set the DRA2 register, set the SVP register.
- To set the DRA3 register, set the VCT register.

Figure 12.5 DMA0 to DMA3 Registers, DSA0 to DSA3 Registers and DRA0 to DRA3 Registers

## 12.1 Transfer Cycle

Transfer cycle contains a bus cycle to read data from a memory or the SFR area (source read) and a bus cycle to write data to a memory space or the SFR area (destination write). The number of read and write bus cycles depends on source and destination addresses.

### 12.1.1 Effect of Source and Destination Addresses

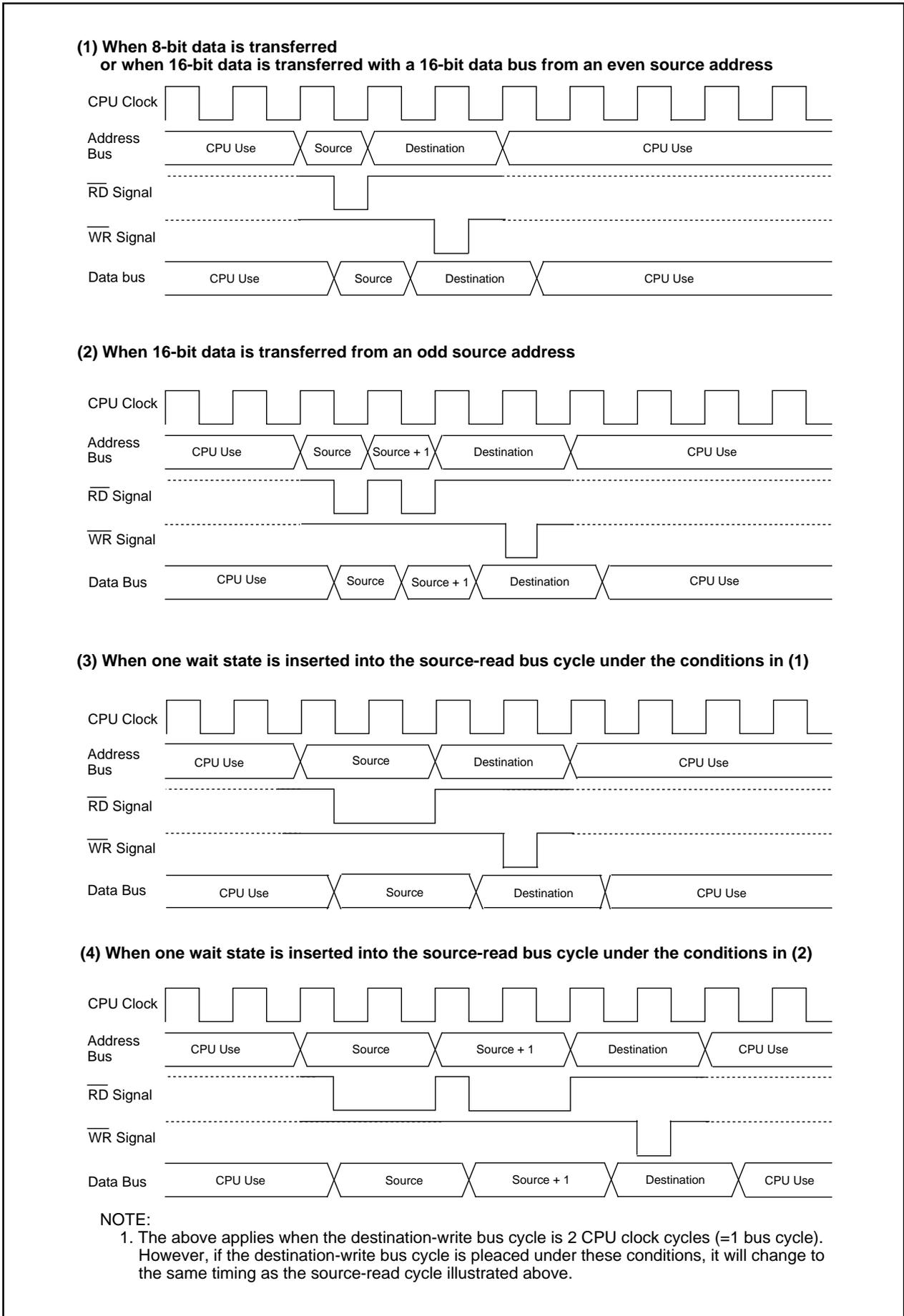
When a 16-bit data is transferred with a 16-bit data bus and a source address starting with an odd address, source read cycle is incremented by one bus cycle, compared to a source address starting with an even address.

When a 16-bit data is transferred with a 16-bit data bus and a destination address starting with an odd address, a destination write cycle is incremented by one bus cycle, compared to a destination address starting with an even address.

### 12.1.2 Effect of Software Wait State

When the SFR area or memory space with software wait states is accessed, the number of CPU clock cycles is incremented by software wait states.

Figure 12.6 shows an example of a transfer cycle for the source-read bus cycle. In Figure 12.6, the number of source-read bus cycles is illustrated under different conditions, provided that the destination address is an address of an external space with the destination-write cycle as two CPU clock cycles (=one bus cycle). In effect, the destination-write bus cycle is also affected by each condition and the transfer cycles change accordingly. To calculate a transfer cycle, apply respective conditions to both destination-write bus cycle and source-read bus cycle.



**Figure 12.6 Transfer Cycle Examples with the Source-Read Bus Cycle**

## 12.2 DMAC Transfer Cycle

The number of DMAC transfer cycle can be calculated as follows.

Any combination of even or odd transfer read and write addresses are possible. Table 12.3 lists the number of DMAC transfer cycles. Table 12.4 lists coefficient j, k.

$$\text{Transfer cycles per transfer} = \text{Number of read cycle} \times j + \text{Number of write cycle} \times k$$

**Table 12.3 DMAC Transfer Cycles**

Transfer Unit	Bus Width	Access Address	Single-chip Mode	
			Read Cycle	Write Cycle
8-bit Transfer (BWi=0)	16 Bits	Even	1	1
		Odd	1	1
	8 Bits	Even	-	-
		Odd	-	-
16-bit Transfer (BWi=1)	16 Bits	Even	1	1
		Odd	2	2
	8 Bits	Even	-	-
		Odd	-	-

BWi: Bit in the DMDp register (i=0 to 3, p=0, 1)

**Table 12.4 Coefficient j, k**

Internal space		
Internal ROM or internal RAM with no wait state	Internal ROM or internal RAM with a wait state	SFR area
j = 1 k = 1	j = 2 k = 2	j = 2 k = 2

## 12.3 Channel Priority and DMA Transfer Timing

When multiple DMA requests are generated in the same sampling period, between the falling edge of the CPU clock and the next falling edge, the DRQ bit in the DMiSL register (i = 0 to 3) is set to "1" (requested) simultaneously. Channel priority in this case is : DMA0 > DMA1 > DMA2 > DMA3.

Figure 12.7 shows an example of the DMA transfer by external source.

In Figure 12.7, the DMA0 request having highest priority is received first to start a transfer when a DMA0 request and DMA1 request are generated simultaneously. After one DMA0 transfer is completed, the bus privilege is returned to the CPU. When the CPU has completed one bus access, the DMA1 transfer starts. After one DMA1 transfer is completed, the privilege is again returned to the CPU.

In addition, DMA requests cannot be counted up since each channel has one DRQ bit. Therefore, when DMA requests, as DMA1 in Figure 12.7, occur more than once before receiving bus privilege, the DRQ bit is set to "0" as soon as privilege is acquired. The bus privilege is returned to the CPU when one transfer is completed.

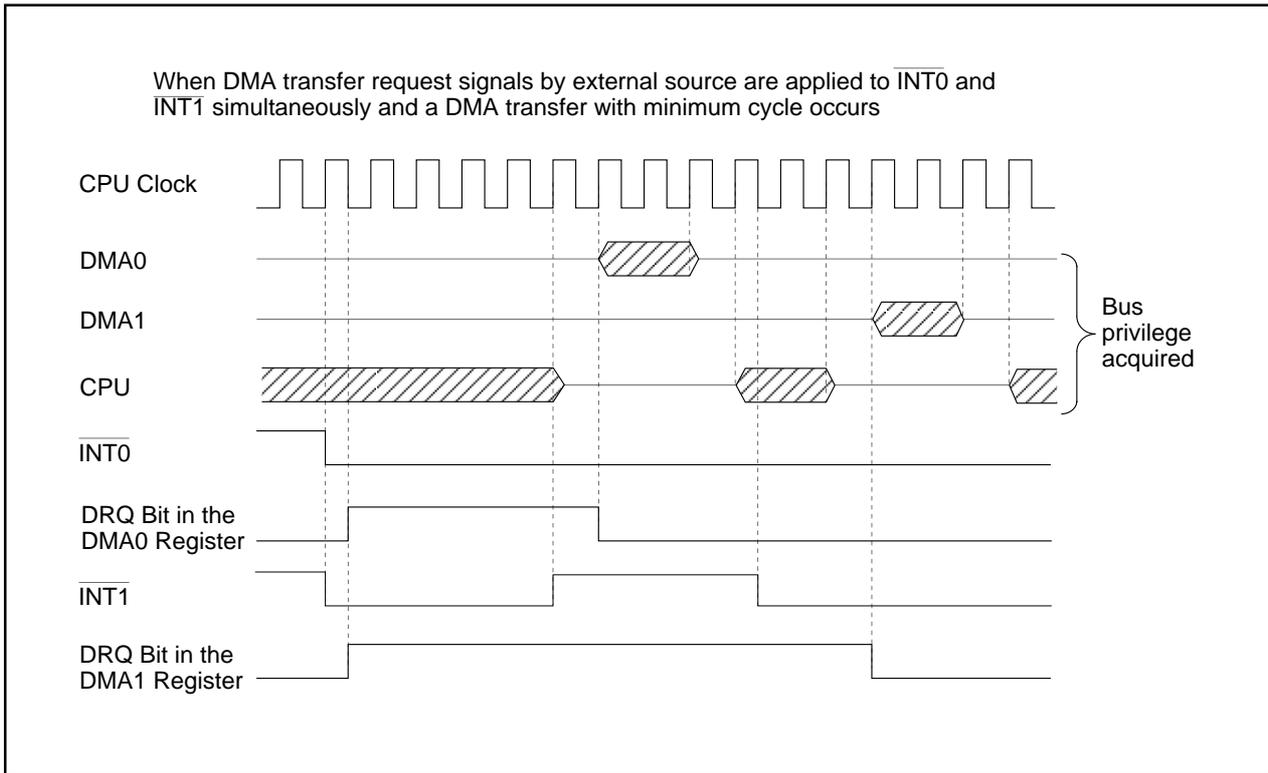


Figure 12.7 DMA Transfer by External Source

## 13. DMAC II

DMAC II performs memory-to-memory transfer, immediate data transfer and calculation transfer, which transfers the sum of two data added by an interrupt request from any peripheral functions.

Table 13.1 lists specifications of DMAC II.

**Table 13.1 DMAC II Specifications**

Item	Specification
DMAC II Request Source	Interrupt requests generated by all peripheral functions when the ILVL2 to ILVL0 bits are set to "1112"
Transfer Data	<ul style="list-style-type: none"> <li>• Data in memory is transferred to memory (memory-to-memory transfer)</li> <li>• Immediate data is transferred to memory (immediate data transfer)</li> <li>• Data in memory (or immediate data) + data in memory are transferred to memory (calculation transfer)</li> </ul>
Transfer Block	8 bits or 16 bits
Transfer Space	64-Kbyte space in addresses 00000 <sub>16</sub> to 0FFFF <sub>16</sub> <sup>(1, 2)</sup>
Transfer Direction	Fixed or forward address Selected separately for each source address and destination address
Transfer Mode	Single transfer, burst transfer
Chained Transfer Function	Parameters (transfer count, transfer address and other information) are switched when transfer counter reaches zero
End-of-Transfer Interrupt	Interrupt occurs when a transfer counter reaches zero
Multiple Transfer Function	Multiple data can be transferred by a generated request for one DMAC II transfer

**NOTES:**

1. When transferring a 16-bit data to destination address 0FFFF<sub>16</sub>, it is transferred to 0FFFF<sub>16</sub> and 10000<sub>16</sub>. The same transfer occurs when the source address is 0FFFF<sub>16</sub>.
2. The actual space where transfer can occur is limited due to internal RAM capacity.

### 13.1 DMAC II Settings

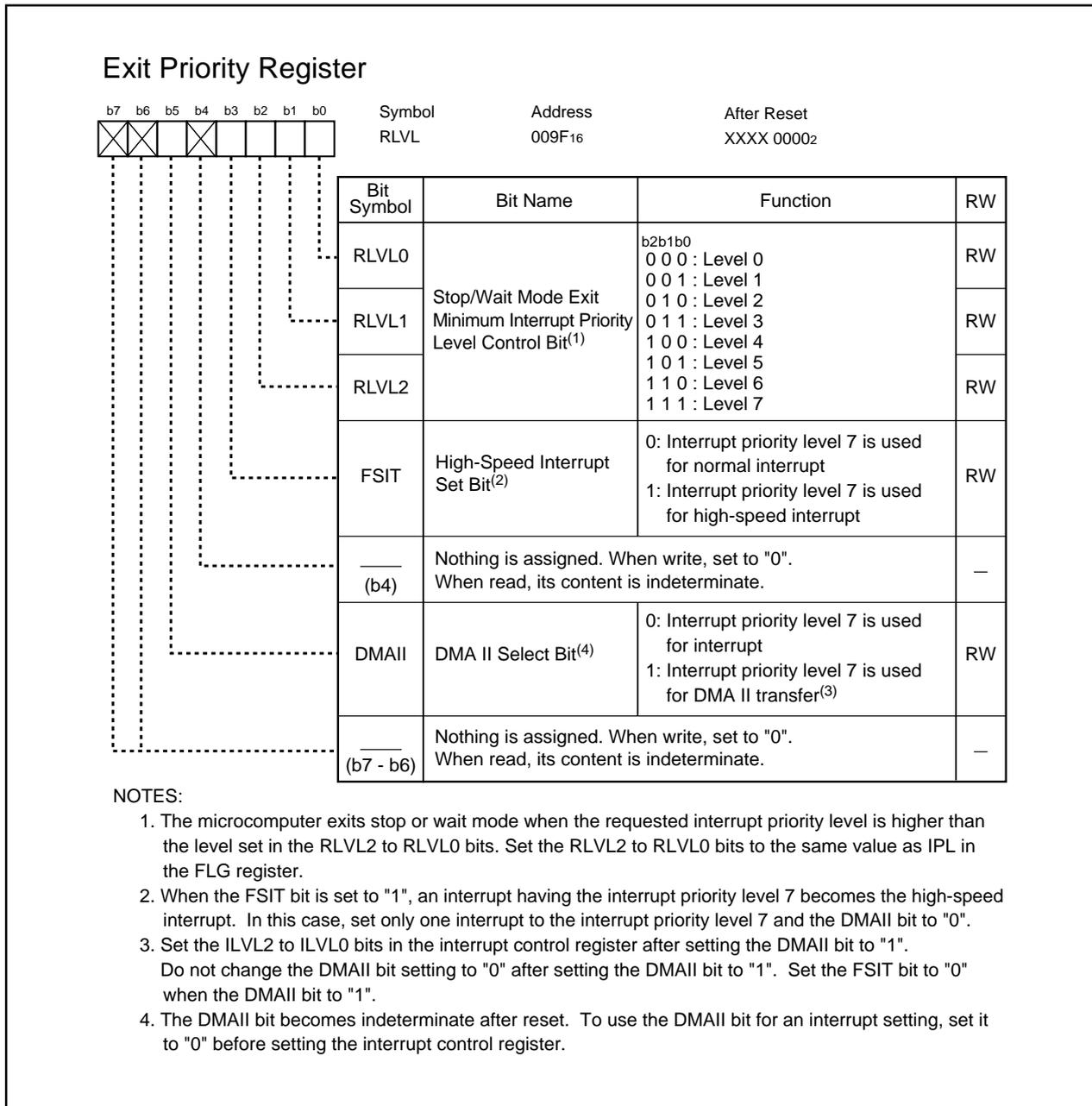
DMAC II can be made available by setting up the following registers and tables.

- RLVL register
  - DMAC II Index
  - Interrupt control register of the peripheral function causing a DMAC II request
  - The relocatable vector table of the peripheral function causing a DMAC II request
  - IRLT bit in the IIOiE register (i = 0 to 5, 8 to 11) if using the intelligent I/O or CAN interrupt
- Refer to **10. Interrupts** for details on the IIOiE register.

#### 13.1.1 RLVL Register

When the DMAII bit is set to "1" (DMAC II transfer) and the FSIT bit to "0" (normal interrupt), DMAC II is activated by an interrupt request from any peripheral function with the ILVL2 to ILVL0 bits in the interrupt control register set to "1112" (level 7).

Figure 13.1 shows the RLVL register.



**Figure 13.1 RLVL Register**

### 13.1.2 DMAC II Index

The DMAC II index is a data table which comprises 8 to 18 bytes (maximum 32 bytes when the multiple transfer function is selected). The DMAC II index stores parameters for transfer mode, transfer counter, source address (or immediate data), operation address as an address to be calculated, destination address, chained transfer address, and end-of-transfer interrupt address.

This DMAC II index must be located on the RAM area.

Figure 13.2 shows a configuration of the DMAC II index. Table 13.2 lists a configuration of the DMAC II index in transfer mode.

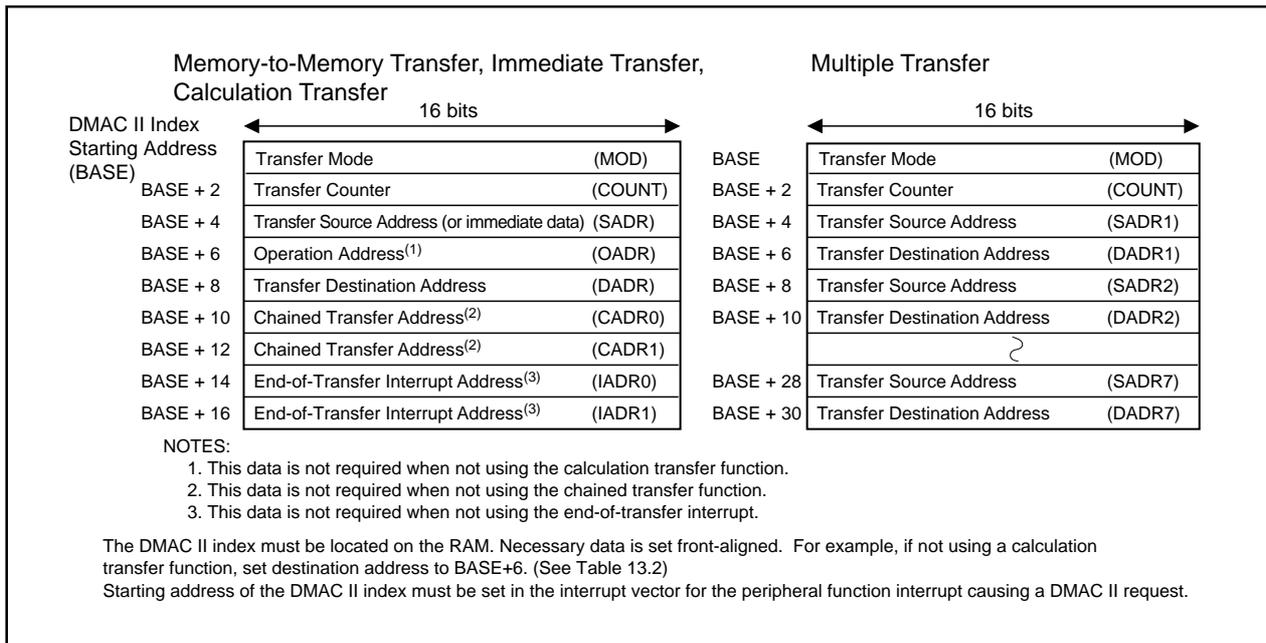


Figure 13.2 DMAC II Index

The followings are details of the DMAC II index. Set these parameters in the specified order listed in Table 13.2, according to DMAC II transfer mode.

- **Transfer mode (MOD)**

Two-byte data is required to set transfer mode. Figure 13.3 shows a configuration for transfer mode.

- **Transfer counter (COUNT)**

Two-byte data is required to set the number of transfer.

- **Transfer source address (SADR)**

Two-byte data is required to set the source memory address or immediate data.

- **Operation address (OADR)**

Two-byte data is required to set a memory address to be calculated. Set this data only when using the calculation transfer function.

- **Transfer destination address (DADR)**

Two-byte data is required to set the destination memory address.

- **Chained transfer address (CADR)**

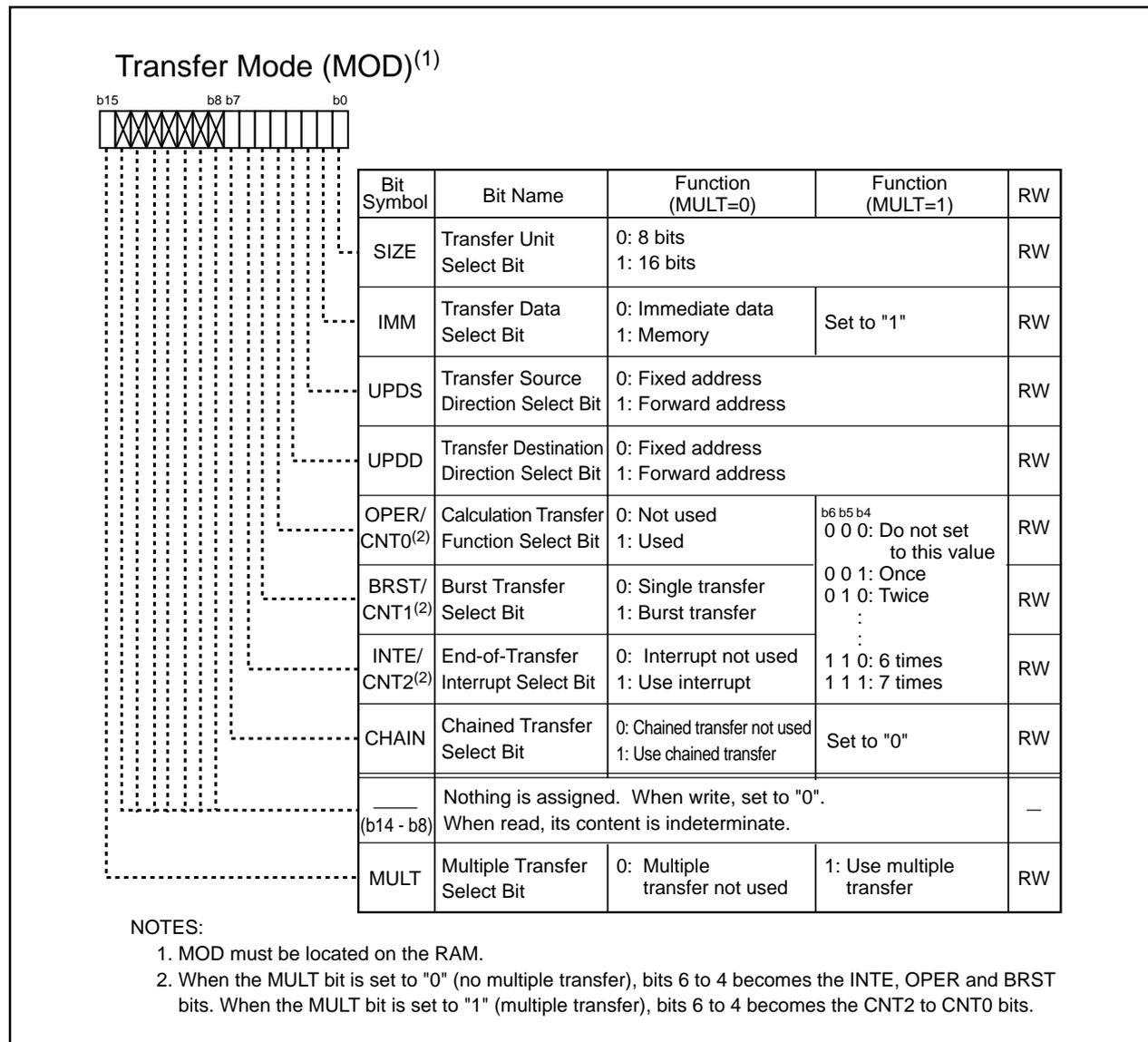
Four-byte data is required to set the starting address of the DMAC II index for the next transfer. Set this data only when using the chained transfer function.

- **End-of-transfer interrupt address (IADR)**

Four-byte data is required to set a jump address for end-of-transfer interrupt processing. Set this data only when using the end-of-transfer interrupt.

**Table 13.2 DMAC II Index Configuration in Transfer Mode**

Transfer Data	Memory-to-Memory Transfer /Immediate Data Transfer				Calculation Transfer				Multiple Transfer																																																														
	Not Used	Used	Not Used	Used	Not Used	Used	Not Used	Used																																																															
Chained Transfer	Not Used	Used	Not Used	Used	Not Used	Used	Not Used	Used	Not Available																																																														
End-of-Transfer Interrupt	Not Used	Not Used	Used	Used	Not Used	Not Used	Used	Used	Not Available																																																														
DMAC II Index	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>DADR</td></tr> </table> <p>8 bytes</p>	MOD	COUNT	SADR	DADR	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>DADR</td></tr> <tr><td>CADR0</td></tr> <tr><td>CADR1</td></tr> </table> <p>12 bytes</p>	MOD	COUNT	SADR	DADR	CADR0	CADR1	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>DADR</td></tr> <tr><td>IADR0</td></tr> <tr><td>IADR1</td></tr> </table> <p>12 bytes</p>	MOD	COUNT	SADR	DADR	IADR0	IADR1	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>DADR</td></tr> <tr><td>CADR0</td></tr> <tr><td>CADR1</td></tr> <tr><td>IADR0</td></tr> <tr><td>IADR1</td></tr> </table> <p>16 bytes</p>	MOD	COUNT	SADR	DADR	CADR0	CADR1	IADR0	IADR1	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>OADR</td></tr> <tr><td>DADR</td></tr> <tr><td>CADR0</td></tr> <tr><td>CADR1</td></tr> </table> <p>10 bytes</p>	MOD	COUNT	SADR	OADR	DADR	CADR0	CADR1	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>OADR</td></tr> <tr><td>DADR</td></tr> <tr><td>CADR0</td></tr> <tr><td>CADR1</td></tr> </table> <p>14 bytes</p>	MOD	COUNT	SADR	OADR	DADR	CADR0	CADR1	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>OADR</td></tr> <tr><td>DADR</td></tr> <tr><td>IADR0</td></tr> <tr><td>IADR1</td></tr> </table> <p>14 bytes</p>	MOD	COUNT	SADR	OADR	DADR	IADR0	IADR1	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>OADR</td></tr> <tr><td>DADR</td></tr> <tr><td>CADR0</td></tr> <tr><td>CADR1</td></tr> <tr><td>IADR0</td></tr> <tr><td>IADR1</td></tr> </table> <p>18 bytes</p>	MOD	COUNT	SADR	OADR	DADR	CADR0	CADR1	IADR0	IADR1	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR1</td></tr> <tr><td>DADR1</td></tr> <tr><td colspan="2">...</td></tr> <tr><td>SADRi</td></tr> <tr><td>DADRi</td></tr> </table> <p>i=1 to 7 max. 32 bytes (when i=7)</p>	MOD	COUNT	SADR1	DADR1	...		SADRi	DADRi
	MOD																																																																						
COUNT																																																																							
SADR																																																																							
DADR																																																																							
MOD																																																																							
COUNT																																																																							
SADR																																																																							
DADR																																																																							
CADR0																																																																							
CADR1																																																																							
MOD																																																																							
COUNT																																																																							
SADR																																																																							
DADR																																																																							
IADR0																																																																							
IADR1																																																																							
MOD																																																																							
COUNT																																																																							
SADR																																																																							
DADR																																																																							
CADR0																																																																							
CADR1																																																																							
IADR0																																																																							
IADR1																																																																							
MOD																																																																							
COUNT																																																																							
SADR																																																																							
OADR																																																																							
DADR																																																																							
CADR0																																																																							
CADR1																																																																							
MOD																																																																							
COUNT																																																																							
SADR																																																																							
OADR																																																																							
DADR																																																																							
CADR0																																																																							
CADR1																																																																							
MOD																																																																							
COUNT																																																																							
SADR																																																																							
OADR																																																																							
DADR																																																																							
IADR0																																																																							
IADR1																																																																							
MOD																																																																							
COUNT																																																																							
SADR																																																																							
OADR																																																																							
DADR																																																																							
CADR0																																																																							
CADR1																																																																							
IADR0																																																																							
IADR1																																																																							
MOD																																																																							
COUNT																																																																							
SADR1																																																																							
DADR1																																																																							
...																																																																							
SADRi																																																																							
DADRi																																																																							



**Figure 13.3 MOD**

### 13.1.3 Interrupt Control Register for the Peripheral Function

For the peripheral function interrupt activating DMAC II, set the ILVL2 to ILVL0 bits to "1112" (level 7).

### 13.1.4 Relocatable Vector Table for the Peripheral Function

Set the starting address of the DMAC II index in the interrupt vector for the peripheral function interrupt activating DMAC II.

When using the chained transfer, the relocatable vector table must be located in the RAM.

### 13.1.5 IRLT Bit in the IIOiE Register (i=0 to 6, 8 to 11)

When the intelligent I/O interrupt or CAN interrupt is used to activate DMAC II, set the IRLT bit in the IIOiE register of the interrupt to "0".

## 13.2 DMAC II Performance

Function to activate DMAC II is selected by setting the DMA II bit to "1" (DMAC II transfer). DMAC II is activated by all peripheral function interrupts with the ILVL2 to ILVL0 bits set to "1112" (level 7). These peripheral function interrupt request signals become DMAC II transfer request signals and the peripheral function interrupt cannot be used.

When an interrupt request is generated by setting the ILVL2 to ILVL0 bits to "1112" (level 7), DMAC II is activated regardless of what state the I flag and IPL are in.

## 13.3 Transfer Data

DMAC II transfers 8-bit or 16-bit data.

- Memory-to-memory transfer : Data is transferred from a desired memory location in a 64-Kbyte space (Addresses 00000<sub>16</sub> to 0FFFF<sub>16</sub>) to another desired memory location in the same space.
- Immediate data transfer : Immediate data is transferred to a desired memory location in a 64-Kbyte space.
- Calculation transfer : Two 8-bit or 16-bit data are added together and the result is transferred to a desired memory location in a 64-Kbyte space.

When a 16-bit data is transferred to the destination address 0FFFF<sub>16</sub>, it is transferred to 0FFFF<sub>16</sub> and 10000<sub>16</sub>. The same transfer occurs when the source address is 0FFFF<sub>16</sub>. Actual transferable space varies depending on the internal RAM capacity.

### 13.3.1 Memory-to-memory Transfer

Data transfer between any two memory locations can be:

- a transfer from a fixed address to another fixed address
- a transfer from a fixed address to a relocatable address
- a transfer from a relocatable address to a fixed address
- a transfer from a relocatable address to another relocatable address

When a relocatable address is selected, the address is incremented, after a transfer, for the next transfer. In a 8-bit transfer, the transfer address is incremented by one. In a 16-bit transfer, the transfer address is incremented by two.

When a source or destination address exceeds address 0FFFF<sub>16</sub> as a result of address incrementation, the source or destination address returns to address 00000<sub>16</sub> and continues incrementation. Maintain source and destination address at address 0FFFF<sub>16</sub> or below.

### 13.3.2 Immediate Data Transfer

DMAC II transfers immediate data to any memory location. A fixed or relocatable address can be selected as the destination address. Store the immediate data into SADR. To transfer an 8-bit immediate data, write the data in the low-order byte of SADR (high-order byte is ignored).

### 13.3.3 Calculation Transfer

After two memory data or an immediate data and memory data are added together, DMAC II transfers calculated result to any memory location. SADR must have one memory location address to be calculated or immediate data and OADR must have the other memory location address to be calculated. Fixed or relocatable address can be selected as source and destination addresses when using a memory + memory calculation transfer. If the transfer source address is relocatable, the operation address also becomes relocatable. Fixed or relocatable address can be selected as the transfer destination address when using an immediate data + memory calculation transfer.

## 13.4 Transfer Modes

Single and burst transfers are available. The BRST bit in MOD selects transfer method, either single transfer or burst transfer. COUNT determines how many transfers occur. No transfer occurs when COUNT is set to "0000<sub>16</sub>".

### 13.4.1 Single Transfer

For every transfer request source, DMAC II transfers one transfer unit of 8-bit or 16-bit data once. When the source or destination address is relocatable, the address is incremented, after a transfer, for the next transfer.

COUNT is decremented every time a transfer occurs. When using the end-of-transfer interrupt, the interrupt is acknowledged when COUNT reaches "0".

### 13.4.2 Burst Transfer

For every transfer request source, DMAC II continuously transfers data the number of times determined by COUNT. COUNT is decremented every time a transfer occurs. The burst transfer ends when COUNT reaches "0". The end-of-transfer interrupt is acknowledged when the burst transfer ends if using the end-of-transfer interrupt. All interrupts are ignored while the burst transfer is in progress.

## 13.5 Multiple Transfer

The MULT bit in MOD selects the multiple transfer. When using the multiple transfer, select the memory-to-memory transfer. One transfer request source initiates multiple transfers. The CNT2 to CNT0 bits in MOD selects the number of transfers from "001<sub>2</sub>" (once) to "111<sub>2</sub>" (7 times). Do not set the CNT2 to CNT0 bits to "000<sub>2</sub>".

The transfer source and destination addresses for each transfer must be allocated alternately in addresses following MOD and COUNT. When the multiple transfer is selected, the calculation transfer, burst transfer, end-of-transfer interrupt and chained transfer cannot be used.

### 13.6 Chained Transfer

The CHAIN bit in MOD selects the chained transfer.

The following process initiates the chained transfer.

- (1) Transfer, caused by a transfer request source, occurs according to the content of the DMAC II index. The vectors of the request source indicates where the DMAC II index is allocated. For each request, the BRST bit selects either single or burst transfer.
- (2) When COUNT reaches "0", the contents of CADR1 and CADR0 are written to the vector of the request source. When the INTE bit in MOD is set to "1", the end-of-transfer interrupt is generated simultaneously.
- (3) When the next DMAC II transfer request is generated, transfer occurs according to the contents of the DMAC II index indicated by the peripheral function interrupt vector rewritten in (2).

Figure 13.4 shows the relocatable vector and DMACII index when the chained transfer is in progress. For the chained transfer, the relocatable vector table must be located in the RAM.

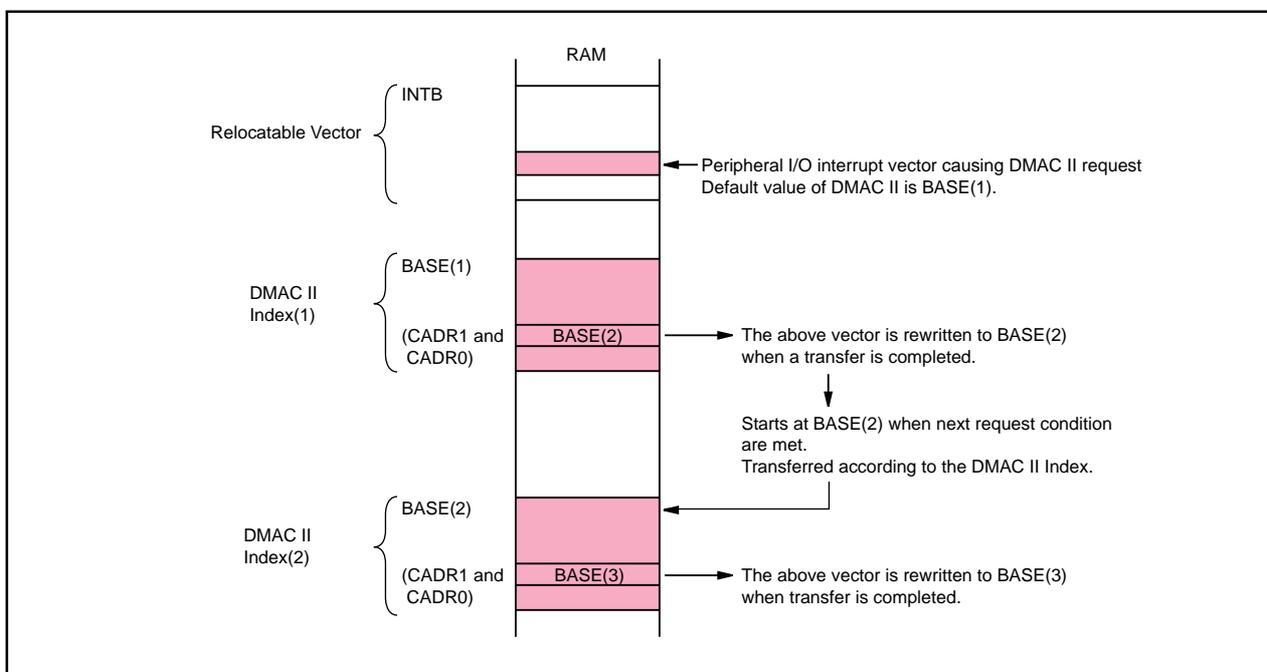


Figure 13.4 Relocatable Vector and DMAC II Index

### 13.7 End-of-Transfer Interrupt

The INTE bit in MOD selects the end-of-transfer interrupt. Set the starting address of the end-of-transfer interrupt routine in IADR1 and IADR0. The end-of-transfer interrupt is generated when COUNT reaches "0."

### 13.8 Execution Time

DMAC II execution cycle is calculated by the following equations:

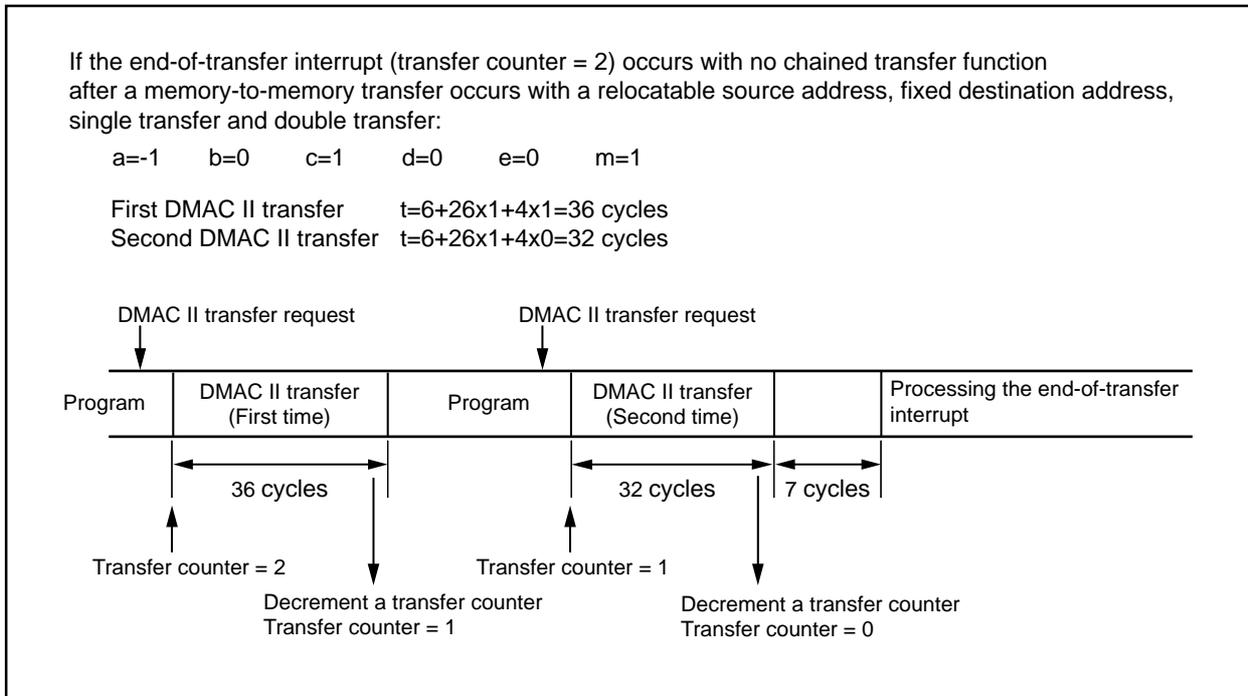
Multiple transfers:  $t = 21 + (11 + b + c) \times k$  cycles

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

- a: If IMM = 0 (source of transfer is immediate data), a = 0;  
if IMM = 1 (source of transfer is memory), a = -1
- b: If UPDS = 1 (source transfer address is a relocatable address), b = 0;  
if UPDS = 0 (source transfer address is a fixed address), b = 1
- c: If UPDD = 1 (destination transfer address is a relocatable address), c = 0;  
if UPDD = 0 (destination transfer address is a fixed address), c = 1
- d: If OPER = 0 (calculation function is not selected), d = 0;  
if OPER = 1 (calculation function is selected) and UPDS = 0 (source of transfer is immediate data or fixed address memory), d = 7;  
if OPER = 1 (calculation function is selected) and UPDS = 1 (source of transfer is relocatable address memory), d = 8
- e: If CHAIN = 0 (chained transfer is not selected), e = 0; if CHAIN = 1 (chained transfer is selected), e = 4
- m: BRST = 0 (single transfer), m = 1; BRST = 1 (burst transfer), m = the value set in transfer counter
- n: If COUNT = 1, n = 0; if COUNT = 2 or more, n = 1
- k: Number of transfers set in the CNT2 to CNT0 bits

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

The first instruction from the end-of-transfer interrupt routine is executed in the eighth cycle after the DMAC II transfer is completed.



**Figure 13.5 Transfer Cycle**

When an interrupt request as a DMAC II transfer request source and another interrupt request with higher priority (e.g.,  $\overline{\text{NMI}}$  or watchdog timer) are generated simultaneously, the interrupt with higher priority takes precedence over the DMAC II transfer. The pending DMAC II transfer starts after the interrupt sequence has been completed.

# 14. Timer

The microcomputer has eleven 16-bit timers. Five timers A and six timers B have different functions. Each timer functions independently. The count source for each timer becomes the clock for timer operations including counting and reloading, etc. Figures 14.1 and 14.2 show block diagrams of timer A and timer B configuration.

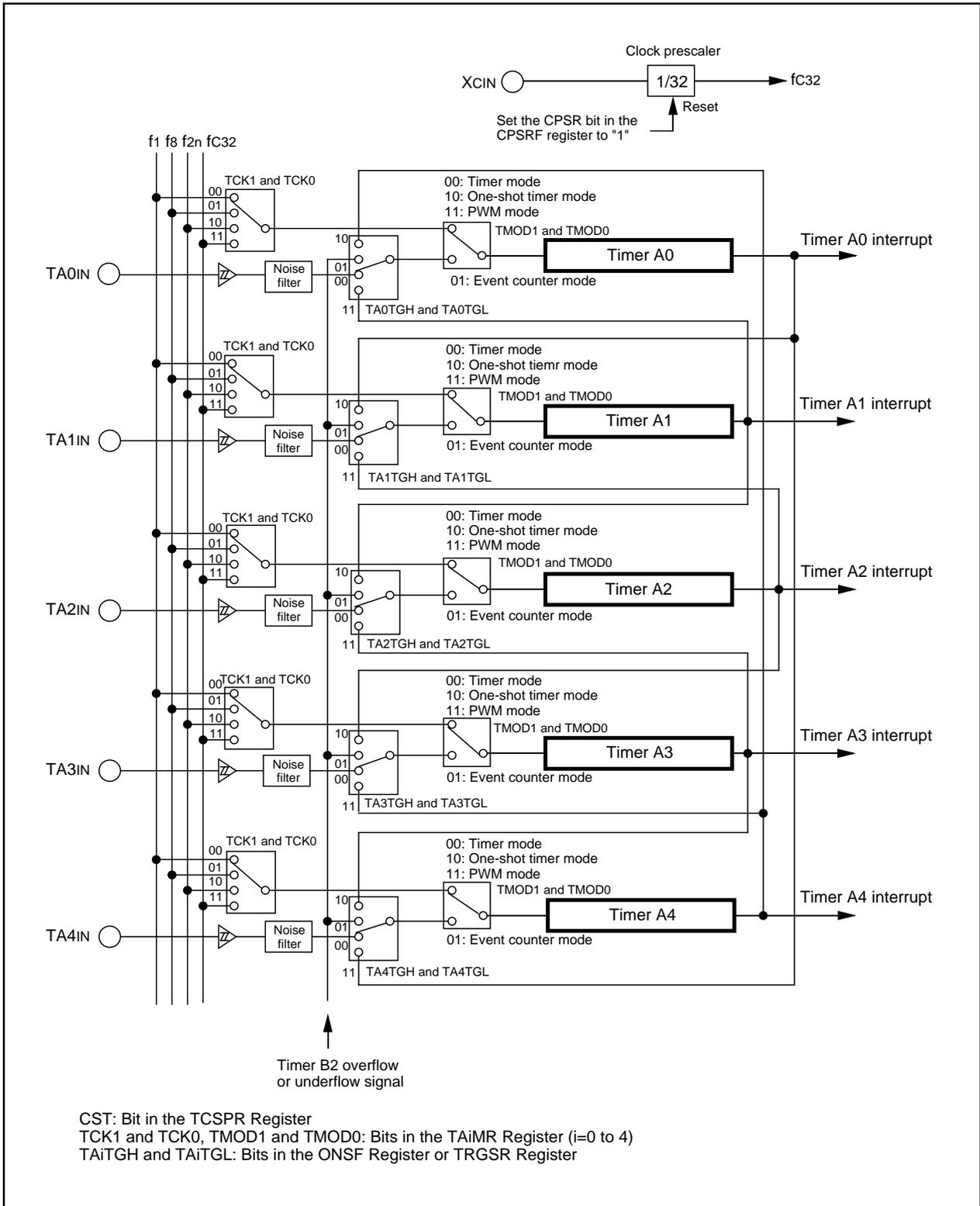


Figure 14.1 Timer A Configuration

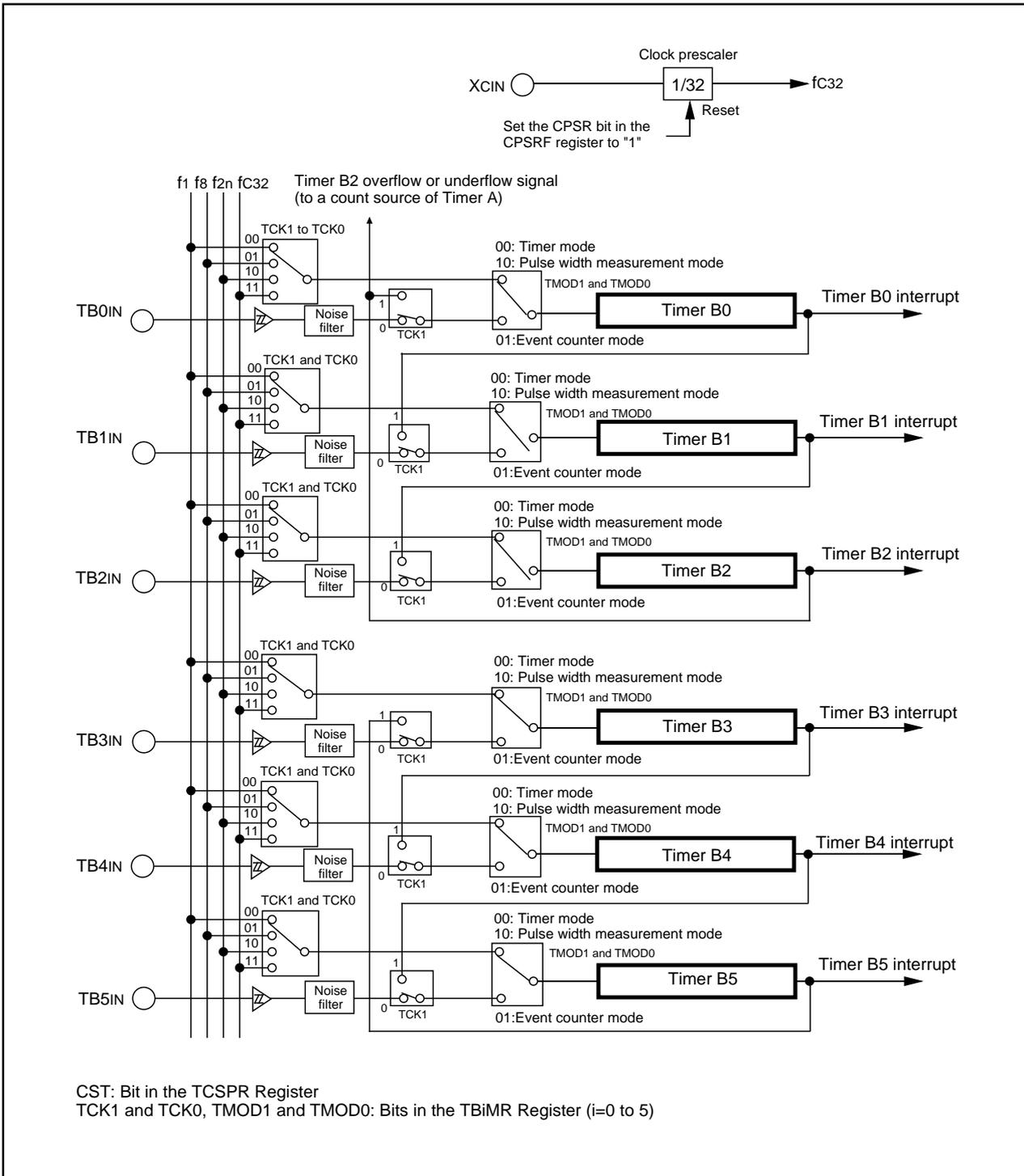


Figure 14.2 Timer B Configuration

### 14.1 Timer A

Figure 14.3 shows a block diagram of the timer A. Figures 14.4 to 14.7 show registers associated with the timer A.

The timer A supports the following four modes. Except in event counter mode, all timers A0 to A4 have the same function. The TMOD1 and TMOD0 bits in the TAI<sub>MR</sub> register (i=0 to 4) determine which mode is used.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts an external pulse or an overflow and underflow of other timers.
- One-shot timer mode: The timer outputs one valid pulse until a counter value reaches "000016".
- Pulse width modulation mode: The timer continuously outputs desired pulse widths.

Table 14.1 lists TAI<sub>OUT</sub> pin settings when used as an output. Table 14.2 lists TAI<sub>IN</sub> and TAI<sub>OUT</sub> pin settings when used as an input.

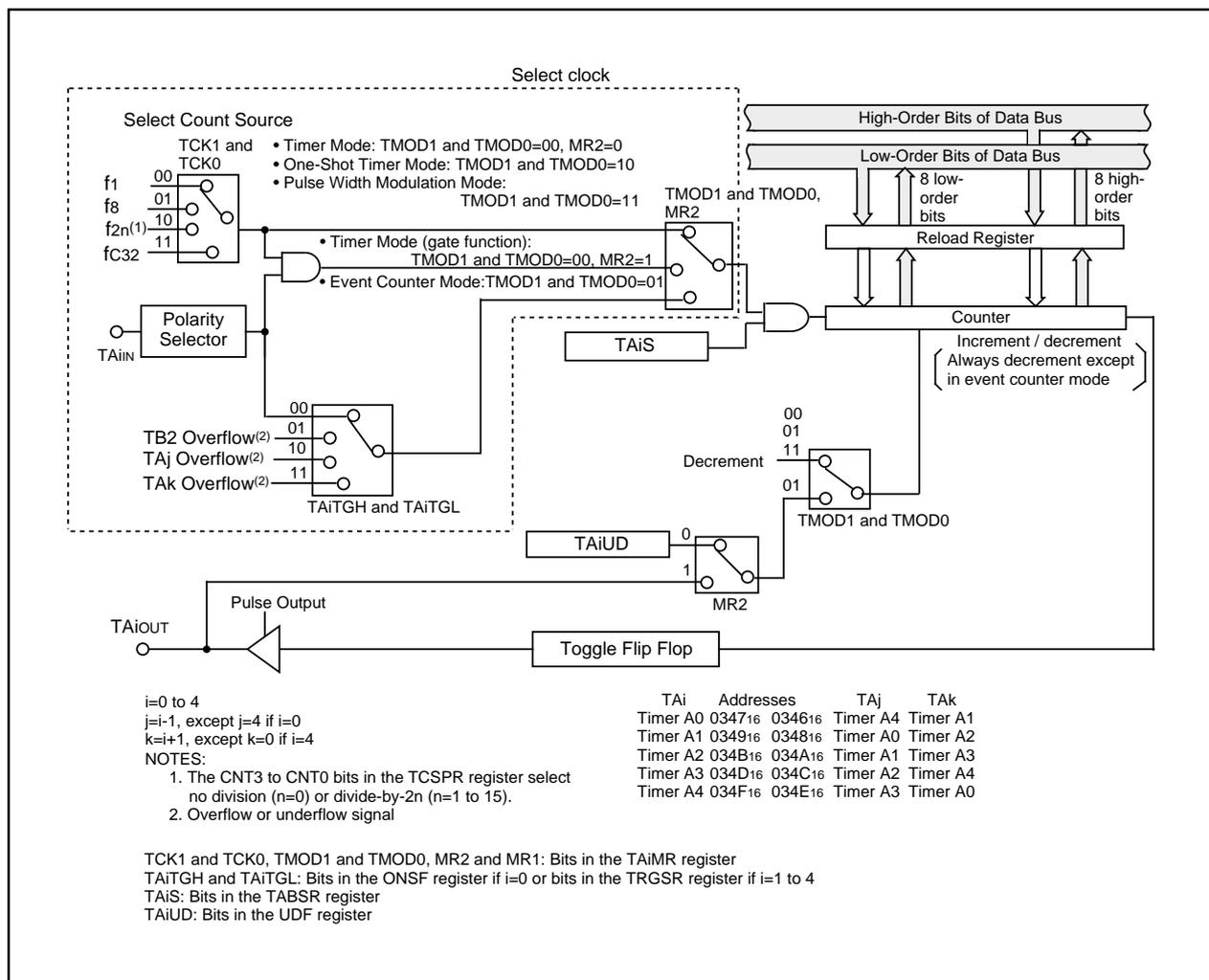


Figure 14.3 Timer A Block Diagram

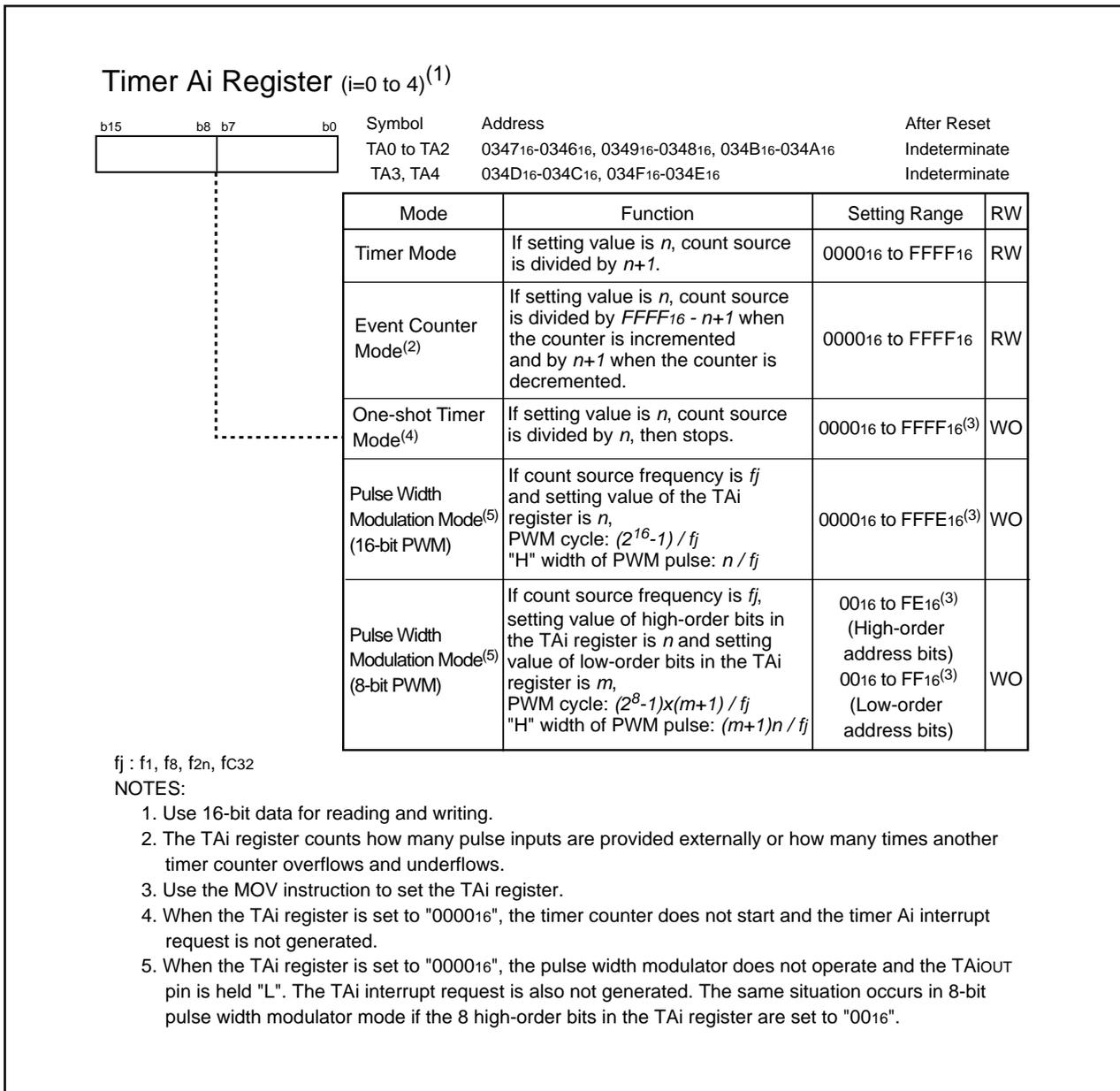


Figure 14.4 TA0 to TA4 Registers

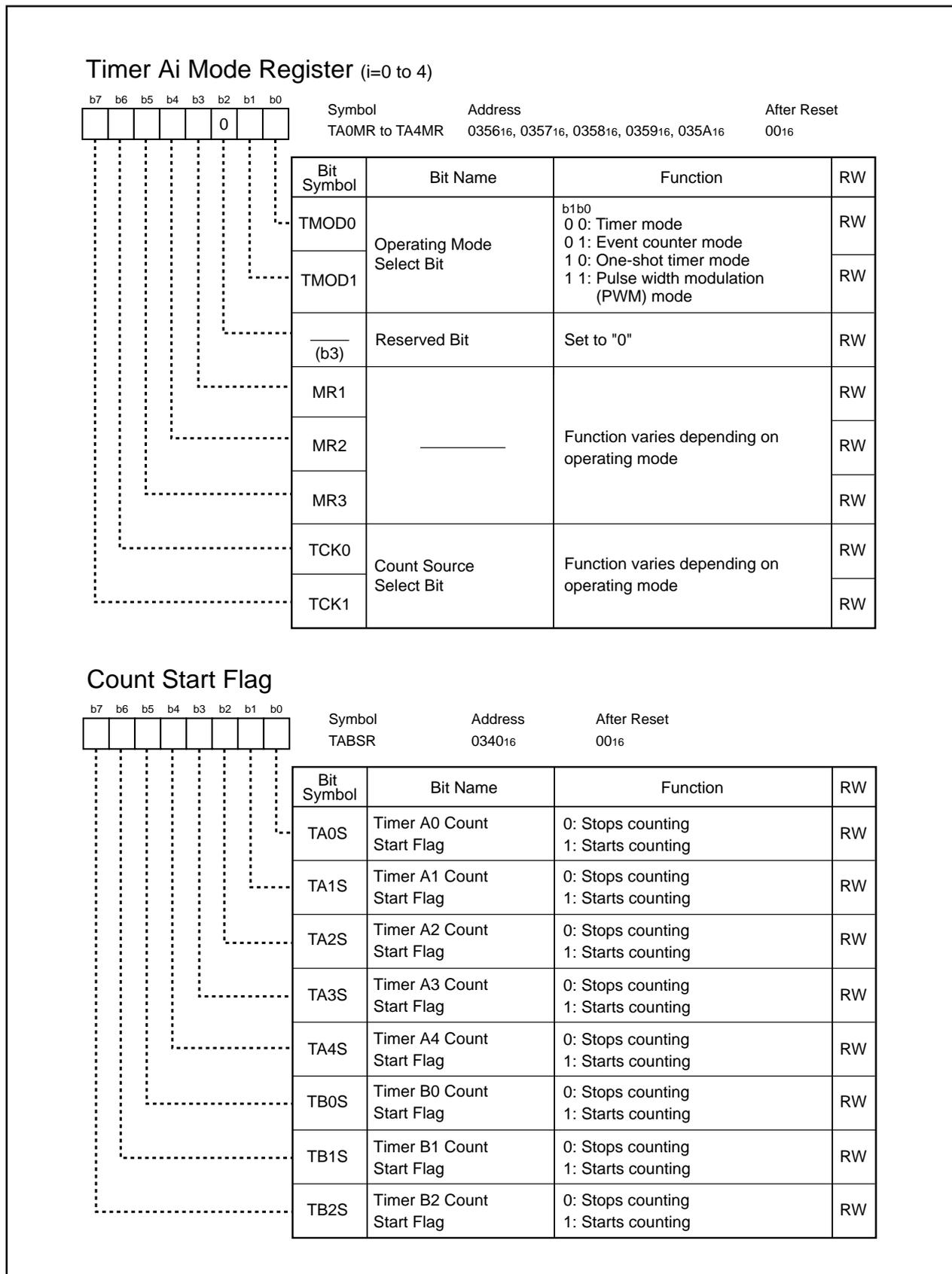


Figure 14.5 TA0MR to TA4MR Registers and TABSR Register

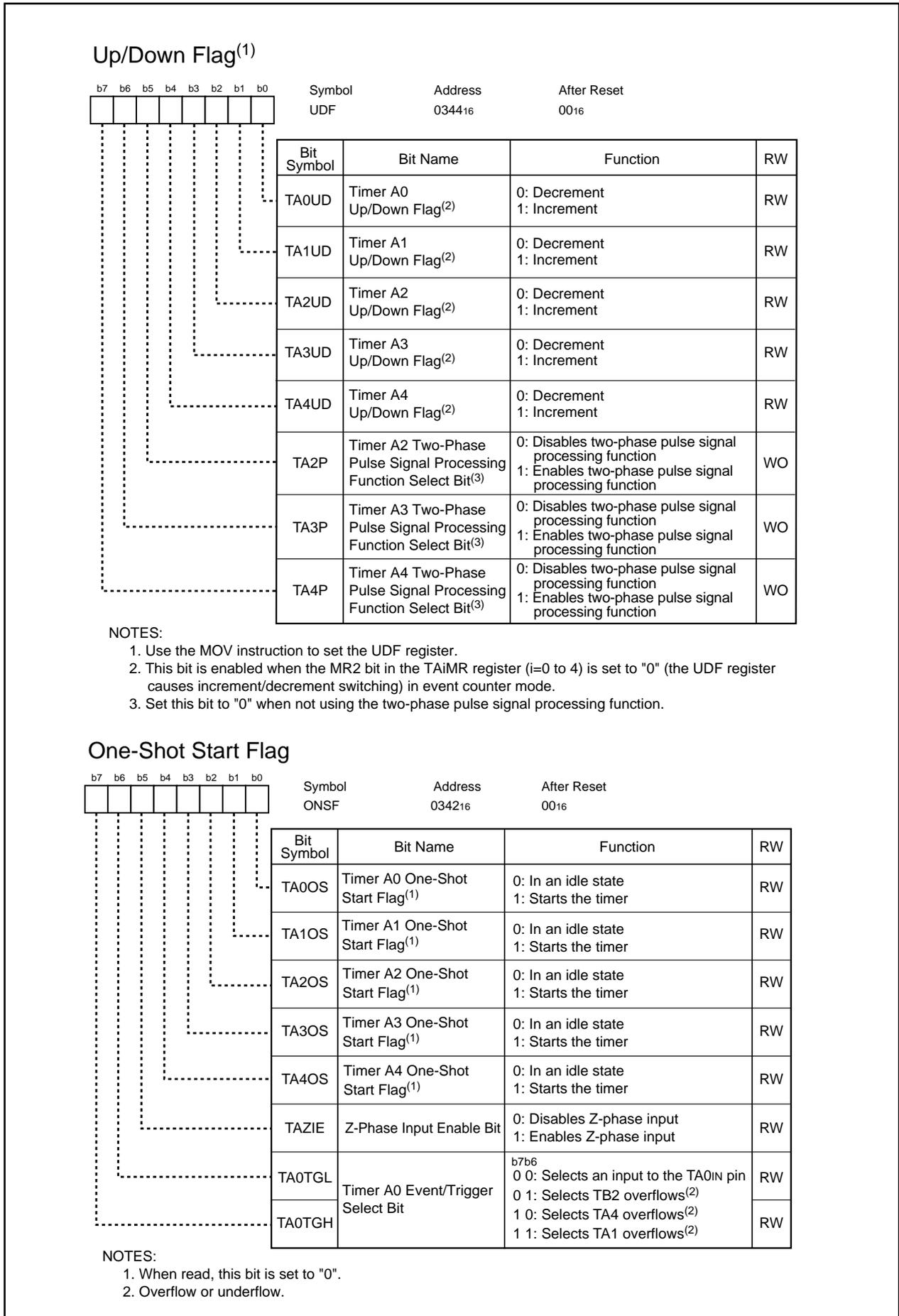
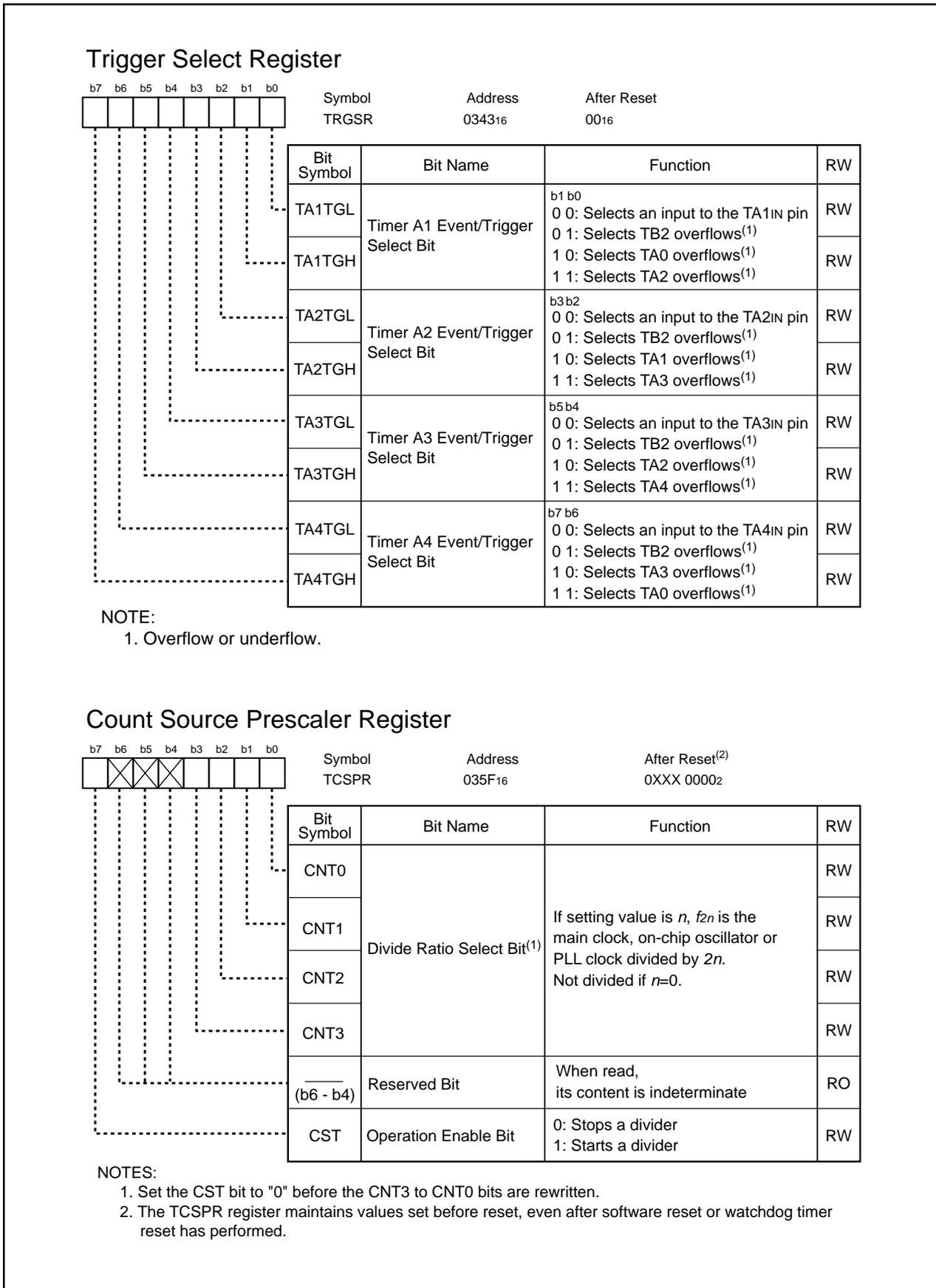


Figure 14.6 UDF Register and ONSF Register



**Figure 14.7 TRGSR Register and TCSPR Register**

**Table 14.1 Pin Settings for Output from TAIOUT Pin (i=0 to 4)**

Pin	Setting		
	PS1, PS2 Registers	PSL1, PSL2 Registers	PSC Register
P70/TA0OUT <sup>(1)</sup>	PS1_0= 1	PSL1_0=1	PSC_0= 0
P72/TA1OUT	PS1_2= 1	PSL1_2=1	PSC_2= 0
P74/TA2OUT	PS1_4= 1	PSL1_4=0	PSC_4= 0
P76/TA3OUT	PS1_6= 1	PSL1_6=1	PSC_6= 0
P80/TA4OUT	PS2_0= 1	PSL2_0=0	–

**NOTE:**

1. P70/TA0OUT is a port for the N-channel open drain output.

**Table 14.2 Pin Settings for Input to TAIIN and TAIOUT Pins (i=0 to 4)**

Pin	Setting	
	PS1, PS2 Registers	PD7, PD8 Registers
P70/TA0OUT	PS1_0=0	PD7_0=0
P71/TA0IN	PS1_1=0	PD7_1=0
P72/TA1OUT	PS1_2=0	PD7_2=0
P73/TA1IN	PS1_3=0	PD7_3=0
P74/TA2OUT	PS1_4=0	PD7_4=0
P75/TA2IN	PS1_5=0	PD7_5=0
P76/TA3OUT	PS1_6=0	PD7_6=0
P77/TA3IN	PS1_7=0	PD7_7=0
P80/TA4OUT	PS2_0=0	PD8_0=0
P81/TA4IN	PS2_1=0	PD8_1=0

### 14.1.1 Timer Mode

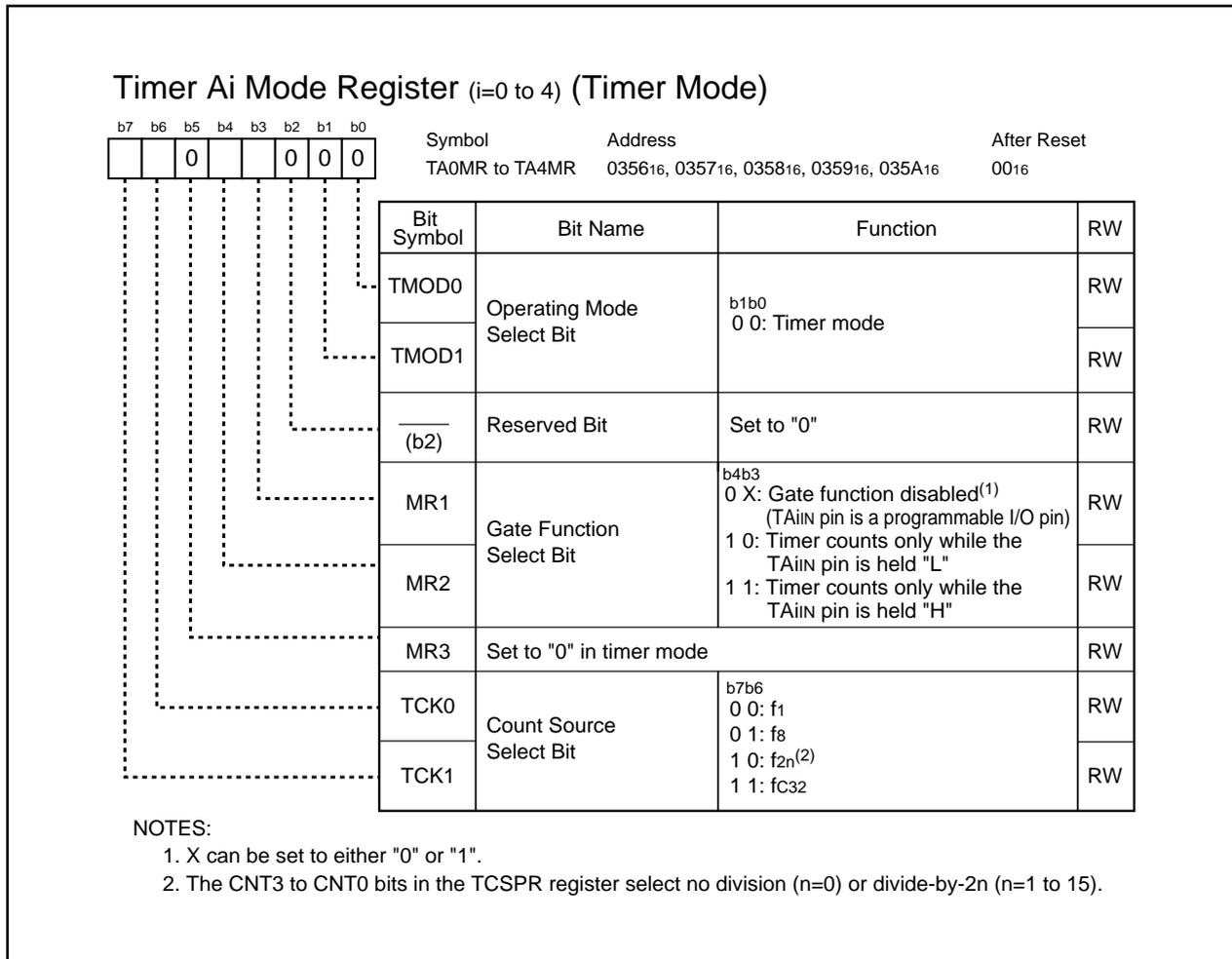
In timer mode, the timer counts an internally generated count source (see **Table 14.3**). Figure 14.8 shows the TAI<sub>MR</sub> register (i=0 to 4) in timer mode.

**Table 14.3 Timer Mode Specifications**

Item	Specification
Count Source	f1, f8, f2 <sub>n</sub> <sup>(1)</sup> , fc32
Counting Operation	<ul style="list-style-type: none"> <li>The timer decrements a counter value</li> </ul> When the timer counter underflows, content of the reload register is reloaded into the count register and counting resumes.
Divide Ratio	1/(n+1)    n: setting value of the TAI register (i=0 to 4)    0000 <sub>16</sub> to FFFF <sub>16</sub>
Counter Start Condition	The TAI <sub>S</sub> bit in the TABSR register is set to "1" (starts counting)
Counter Stop Condition	The TAI <sub>S</sub> bit is set to "0" (stops counting)
Interrupt Request Generation Timing	The timer counter underflows
TAI <sub>IN</sub> Pin Function	Programmable I/O port or gate input
TAI <sub>OUT</sub> Pin Function	Programmable I/O port or pulse output
Read from Timer	The TAI register indicates counter value
Write to Timer	<ul style="list-style-type: none"> <li>While the timer counter stops, the value written to the TAI register is also written to both reload register and counter</li> <li>While counting, the value written to the TAI register is written to the reload register (It is transferred to the counter at the next reload timing)</li> </ul>
Selectable Function	<ul style="list-style-type: none"> <li>Gate function Input signal to the TAI<sub>IN</sub> pin determines whether the timer counter starts or stops counting</li> <li>Pulse output function The polarity of the TAI<sub>OUT</sub> pin is inverted whenever the timer counter underflows</li> </ul>

**NOTE:**

- The CNT3 to CNT0 bits in the TCSPR register select no division (n=0) or divide-by-2<sup>n</sup> (n=1 to 15).



**Figure 14.8 TA0MR to TA4MR Registers**

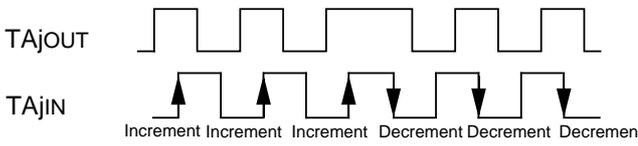
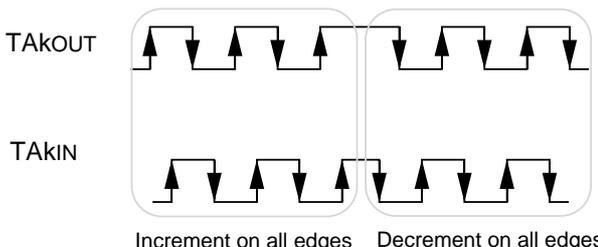
### 14.1.2 Event Counter Mode

In event counter mode, the timer counts how many external signals are applied or how many times another timer counter overflows and underflows. The timers A2, A3 and A4 can count externally generated two-phase signals. Table 14.4 lists specifications in event counter mode (when not handling a two-phase pulse signal). Table 14.5 lists specifications in event counter mode (when handling a two-phase pulse signal with the timers A2, A3 and A4). Figure 14.9 shows the TAI<sub>MR</sub> register (i=0 to 4) in event counter mode.

**Table 14.4 Event Counter Mode Specifications (When Not Processing Two-phase Pulse Signal)**

Item	Specification
Count Source	<ul style="list-style-type: none"> <li>External signal applied to the TAI<sub>IN</sub> pin (i = 0 to 4) (valid edge can be selected by program)</li> <li>Timer B2 overflow or underflow signal, timer A<sub>j</sub> overflow or underflow signal (j=i-1, except j=4 if i=0) and timer A<sub>k</sub> overflow or underflow signal (k=i+1, except k=0 if i=4)</li> </ul>
Counting Operation	<ul style="list-style-type: none"> <li>External signal and program can determine whether the timer increments or decrements a counter value</li> <li>When the timer counter underflows or overflows, content of the reload register is reloaded into the count register and counting resumes. When the free-running count function is selected, the timer counter continues running without reloading.</li> </ul>
Divide Ratio	<ul style="list-style-type: none"> <li><math>1/(FFFF_{16} - n + 1)</math> for counter increment</li> <li><math>1/(n + 1)</math> for counter decrement    <i>n</i>: setting value of the TAI register 0000<sub>16</sub> to FFFF<sub>16</sub></li> </ul>
Counter Start Condition	The TAI <sub>S</sub> bit in the TABSR register is set to "1" (starts counting)
Counter Stop Condition	The TAI <sub>S</sub> bit is set to "0" (stops counting)
Interrupt Request Generation Timing	The timer counter overflows or underflows
TAI <sub>IN</sub> Pin Function	Programmable I/O port or count source input
TAI <sub>OUT</sub> Pin Function	Programmable I/O port, pulse output or input selecting a counter increment or decrement
Read from Timer	The TAI register indicates counter value
Write to Timer	<ul style="list-style-type: none"> <li>When the timer counter stops, the value written to the TAI register is also written to both reload register and counter</li> <li>While counting, the value written to the TAI register is written to the reload register (It is transferred to the counter at the next reload timing)</li> </ul>
Selectable Function	<ul style="list-style-type: none"> <li>Free-running count function Content of the reload register is not reloaded even if the timer counter overflows or underflows</li> <li>Pulse output function The polarity of the TAI<sub>OUT</sub> pin is inversed whenever the timer counter overflows or underflows</li> </ul>

**Table 14.5 Event Counter Mode Specifications (When Processing Two-phase Pulse Signal on Timer A2, A3 and A4)**

Item	Specification
Count Source	Two-phase pulse signal applied to the TAIIN and TAIOUT pins (i = 2 to 4)
Counting Operation	<ul style="list-style-type: none"> <li>Two-phase pulse signal determines whether the timer increments or decrements a counter value</li> <li>When the timer counter overflows or underflows, content of the reload register is reloaded into the count register and counting resumes. With the free-running count function, the timer counter continues running without reloading.</li> </ul>
Divide Ratio	<ul style="list-style-type: none"> <li><math>1/(FFFF_{16} - n + 1)</math> for counter increment</li> <li><math>1/(n + 1)</math> for counter decrement <math>n</math>: setting value of the TAI register <math>0000_{16}</math> to <math>FFFF_{16}</math></li> </ul>
Counter Start Condition	The TAI <sub>S</sub> bit in the TABSR register is set to "1" (starts counting)
Counter Stop Condition	The TAI <sub>S</sub> bit is set to "0" (stops counting)
Interrupt Request Generation Timing	The timer counter overflows or underflows
TAiIN Pin Function	Two-phase pulse signal is applied
TAiOUT Pin Function	Two-phase pulse signal is applied
Read from Timer	The TAI register indicates the counter value
Write to Timer	<ul style="list-style-type: none"> <li>When the timer counter stops, the value written to the TAI register is also written to both reload register and counter</li> <li>While counting, the value written to the TAI register is written to the reload register (It is transferred to the counter at the next reload timing)</li> </ul>
Selectable Function <sup>(1)</sup>	<ul style="list-style-type: none"> <li>Normal processing operation (the timer A2 and timer A3) While a high-level ("H") signal is applied to the TAJOUT pin (j = 2 or 3), the timer increments a counter value on the rising edge of the TAJIN pin or decrements a counter on the falling edge.</li> </ul>  <ul style="list-style-type: none"> <li>Multiply-by-4 processing operation (the timer A3 and timer A4) While an "H" signal is applied to the TAKOUT pin (k = 3 or 4) on the rising edge of the TAKIN pin, the timer increments a counter value on the rising and falling edges of the TAKOUT and TAKIN pins. While an "H" signal is applied to the TAKOUT pin on the falling edge of the TAKIN pin, the timer decrements a counter value on the rising and falling edges of the TAKOUT and TAKIN pins.</li> </ul> 

**NOTE:**

- Only timer A3 operation can be selected. The timer A2 is for the normal processing operation. The timer A4 is for the multiply-by-4 operation.

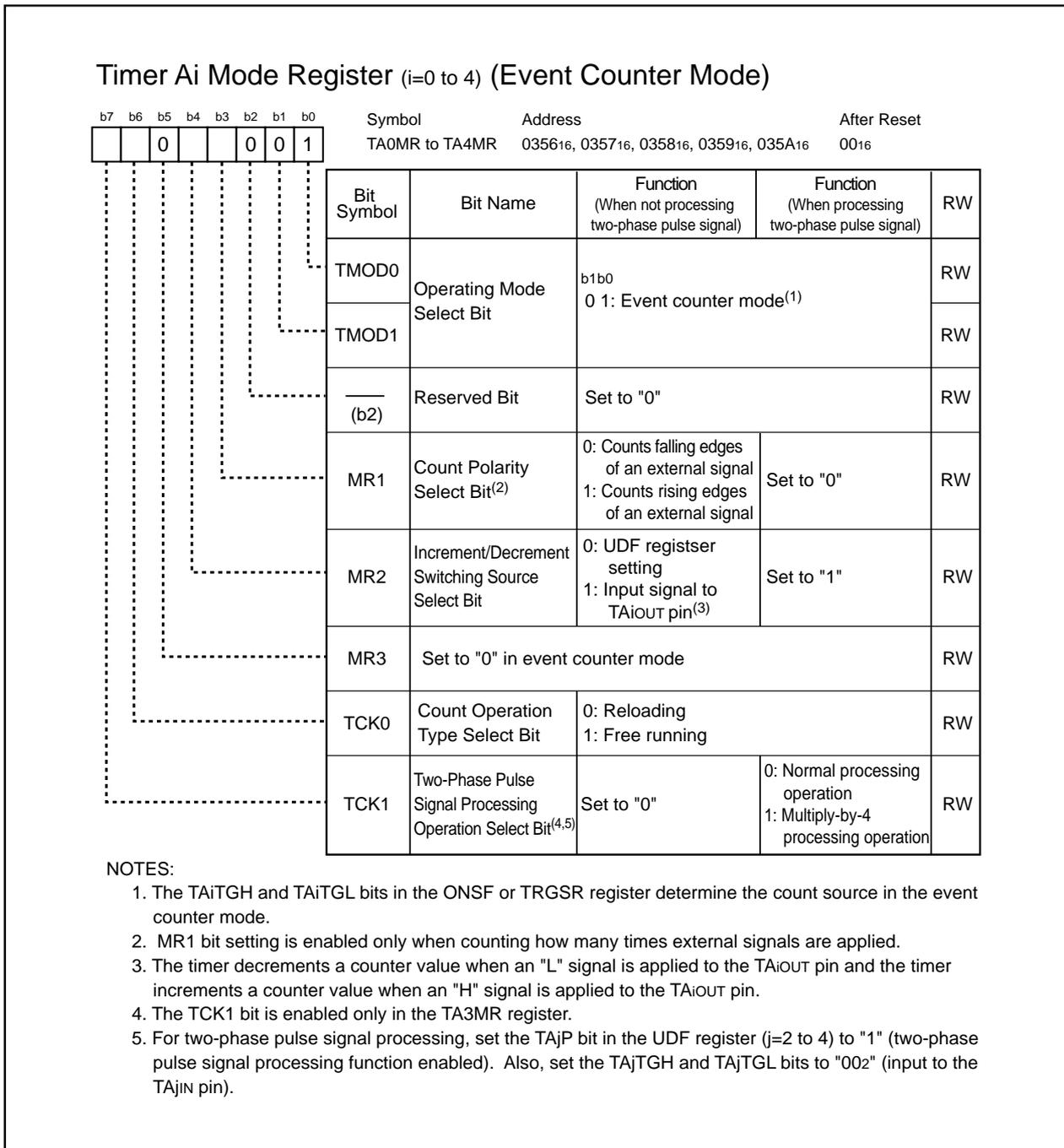


Figure 14.9 TA0MR to TA4MR Registers

### 14.1.2.1 Counter Reset by Two-Phase Pulse Signal Processing

Z-phase input resets the timer counter when processing a two-phase pulse signal.

This function can be used in timer A3 event counter mode, two-phase pulse signal processing, free-running count operation type or multiply-by-4 processing. The Z-phase signal is applied to the  $\overline{\text{INT2}}$  pin. When the TAZIE bit in the ONSF register is set to "1" (Z-phase input enabled), Z-phase input can reset the timer counter. To reset the counter by a Z-phase input, set the TA3 register to "000016" beforehand.

Z-phase input is enabled when the edge of the signal applied to the  $\overline{\text{INT2}}$  pin is detected. The POL bit in the INT2IC register can determine edge polarity. The Z-phase must have a pulse width of one timer A3 count source cycle or more. Figure 14.10 shows two-phase pulses (A-phase and B-phase) and the Z-phase.

Z-phase input resets the timer counter in the next count source following Z-phase input. Figure 14.11 shows the counter reset timing.

Timer A3 interrupt request is generated twice continuously when a timer A3 overflow or underflow, and a counter reset by  $\overline{\text{INT2}}$  input occur at the same time. Do not use the timer A3 interrupt request when this function is used.

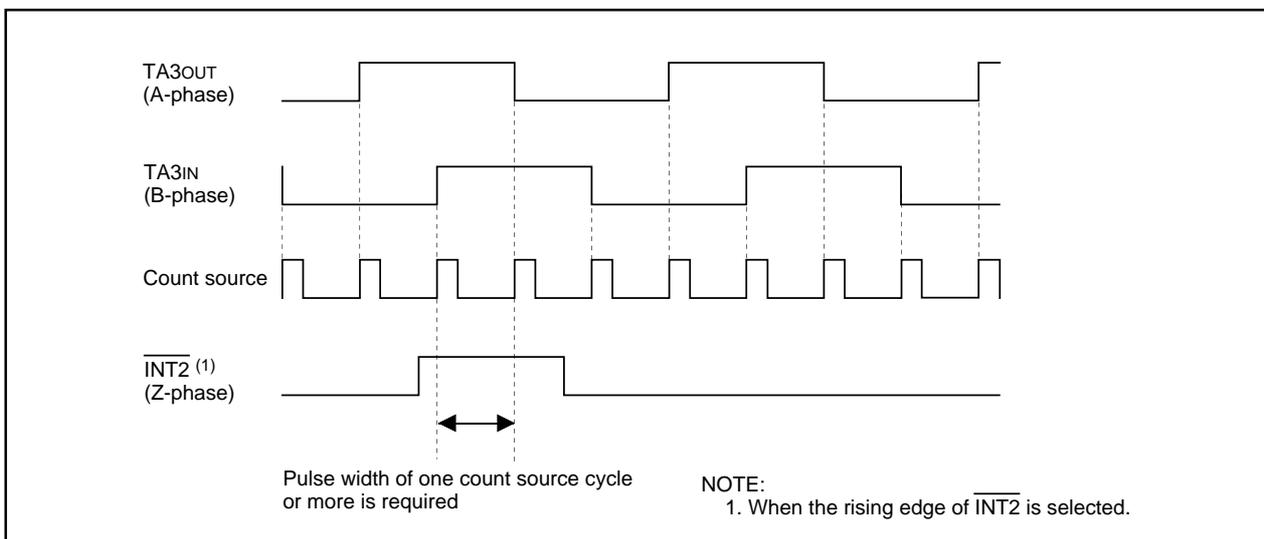


Figure 14.10 Two-Phase Pulse (A-phase and B-phase) and Z-phase

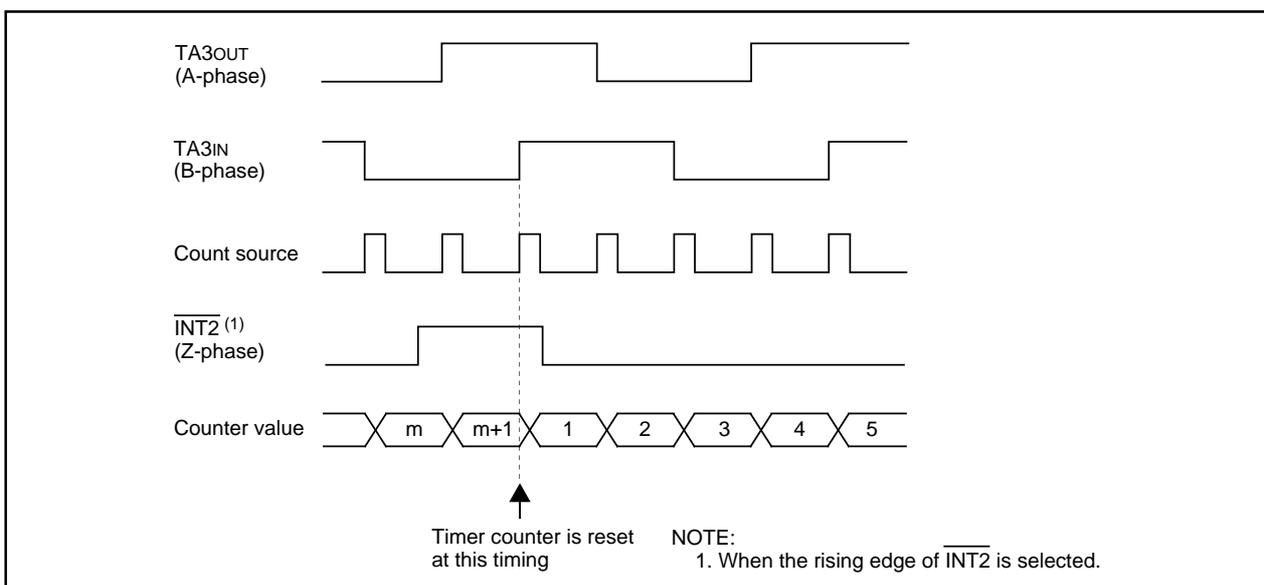


Figure 14.11 Counter Reset Timing

### 14.1.3 One-Shot Timer Mode

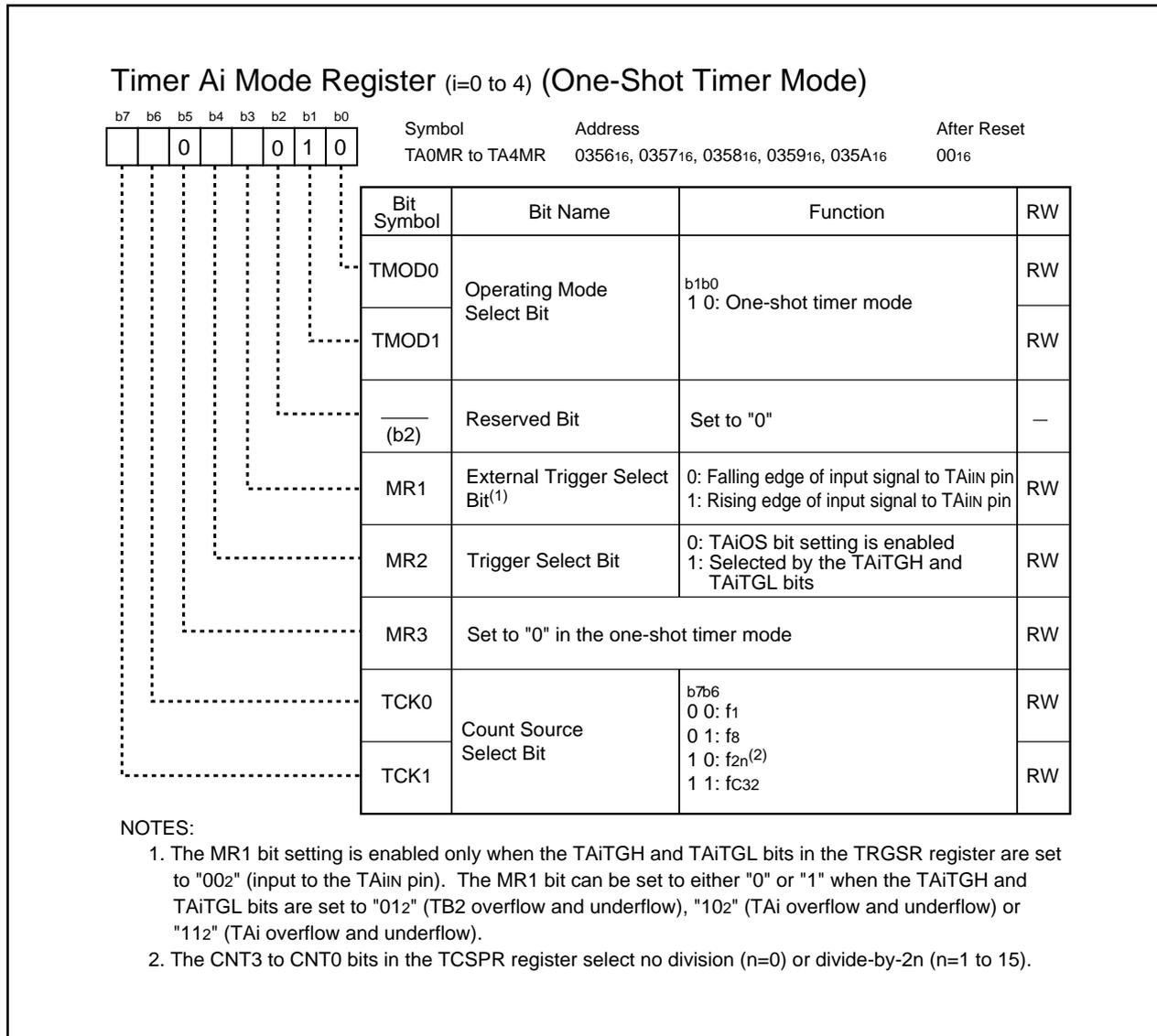
In one-shot timer mode, the timer operates only once for each trigger (see **Table 14.6**). Once a trigger occurs, the timer starts and continues operating for a desired period. Figure 14.12 shows the TAI<sub>M</sub>R register (i=0 to 4) in one-shot timer mode.

**Table 14.6 One-Shot Timer Mode Specifications**

Item	Specification
Count Source	f <sub>1</sub> , f <sub>8</sub> , f <sub>2<sup>n</sup></sub> <sup>(1)</sup> , f <sub>C32</sub>
Counting Operation	<ul style="list-style-type: none"> <li>The timer decrements a counter value</li> </ul> When the timer counter reaches "0000 <sub>16</sub> ", it stops counting after reloading. If a trigger occurs while counting, content of the reload register is reloaded into the count register and counting resumes.
Divide Ratio	1/n    n: setting value of the TAI register (i=0 to 4) 0000 <sub>16</sub> to FFFF <sub>16</sub> , but the timer counter does not run if n=0000 <sub>16</sub>
Counter Start Condition	The TAI <sub>S</sub> bit in the TABSR register is set to "1" (starts counting) and following triggers occur: <ul style="list-style-type: none"> <li>External trigger input is provided</li> <li>Timer counter overflows or underflows</li> <li>The TAI<sub>OS</sub> bit in the ONSF register is set to "1" (timer started)</li> </ul>
Counter Stop Condition	<ul style="list-style-type: none"> <li>After the timer counter has reached "0000<sub>16</sub>" and is reloaded</li> <li>When the TAI<sub>S</sub> bit is set to "0" (stops counting)</li> </ul>
Interrupt Request Generation Timing	The timer counter reaches "0000 <sub>16</sub> "
TAI <sub>IN</sub> Pin Function	Programmable I/O port or trigger input
TAI <sub>OUT</sub> Pin Function	Programmable I/O port or pulse output
Read from Timer	The value in the TAI register is indeterminate when read
Write to Timer	<ul style="list-style-type: none"> <li>When the timer counter stops, the value written to the TAI register is also written to both reload register and counter</li> <li>While counting, the value written to the TAI register is written to the reload register (It is transferred to the counter at the next reload timing)</li> </ul>

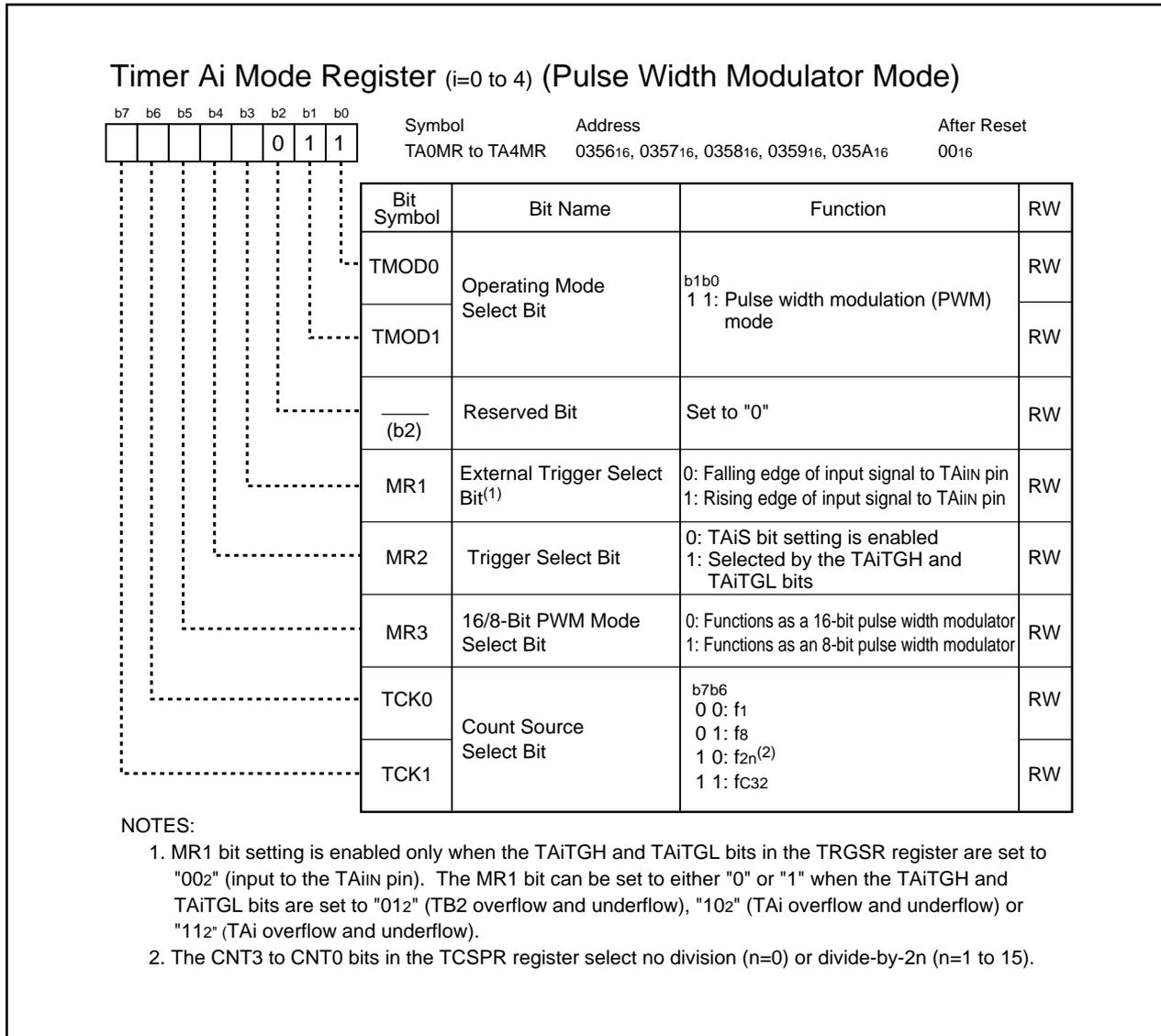
**NOTE:**

- The CNT3 to CNT0 bits in the TCSPR register select no division (n=0) or divide-by-2<sup>n</sup> (n=1 to 15).



**Figure 14.12 TA0MR to TA4MR Registers**





**Figure 14.13 TA0MR to TA4MR Registers**

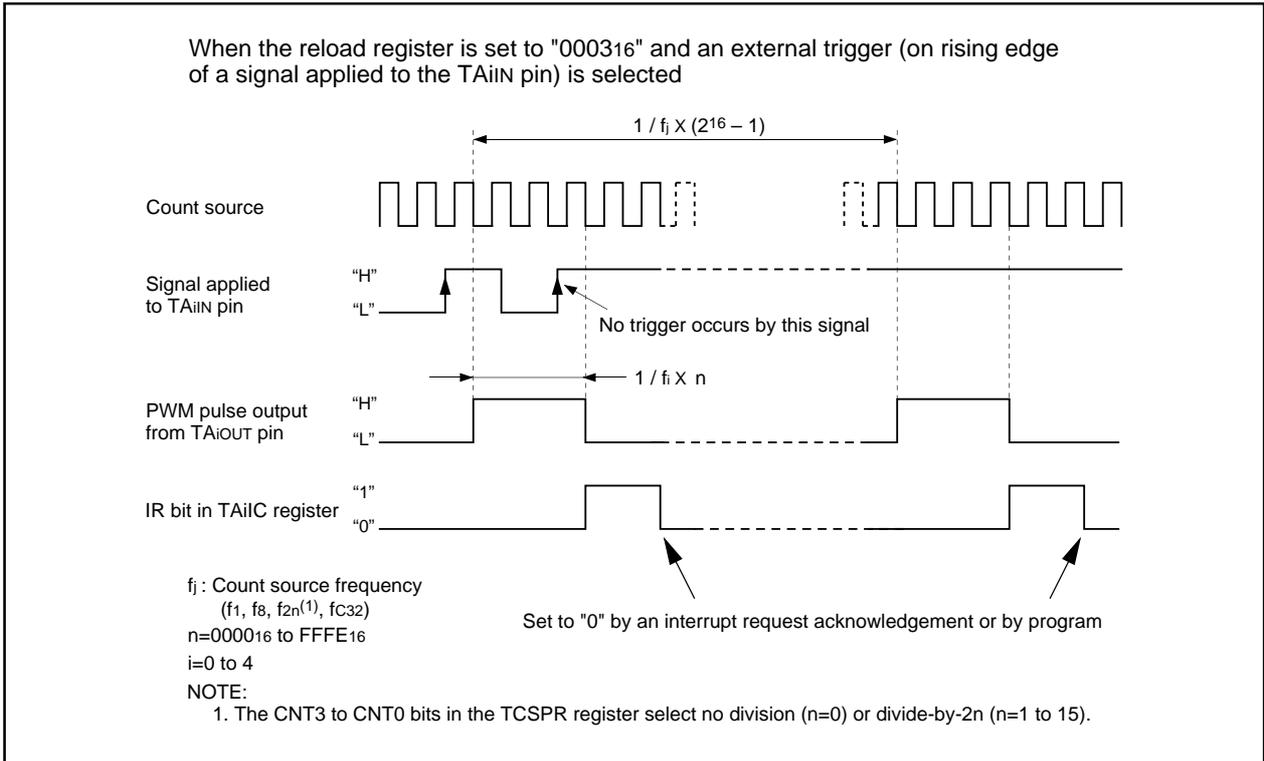


Figure 14.14 16-bit Pulse Width Modulator Operation

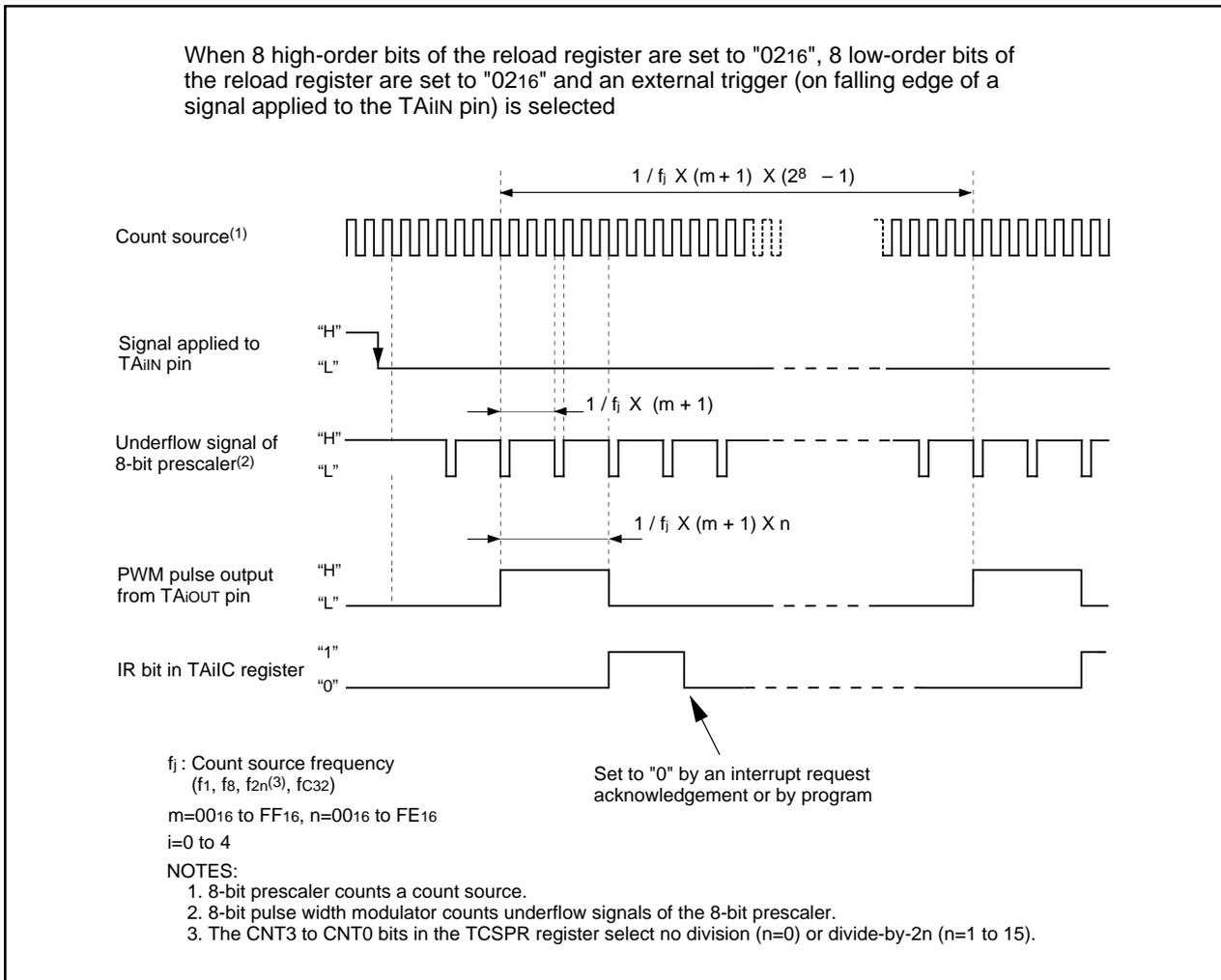


Figure 14.15 8-bit Pulse Width Modulator Operation

### 14.2 Timer B

Figure 14.16 shows a block diagram of the timer B. Figures 14.17 to 14.19 show registers associated with the timer B. The timer B supports the following three modes. The TMOD1 and TMOD0 bits in the TBiMR register ( $i=0$  to 5) determine which mode is used.

- Timer mode : The timer counts an internal count source.
- Event counter mode : The timer counts pulses from an external source or overflow and underflow of another timer.
- Pulse period/pulse width measurement mode : The timer measures pulse period or pulse width of an external signal.

Table 14.8 lists TBiIN pin settings.

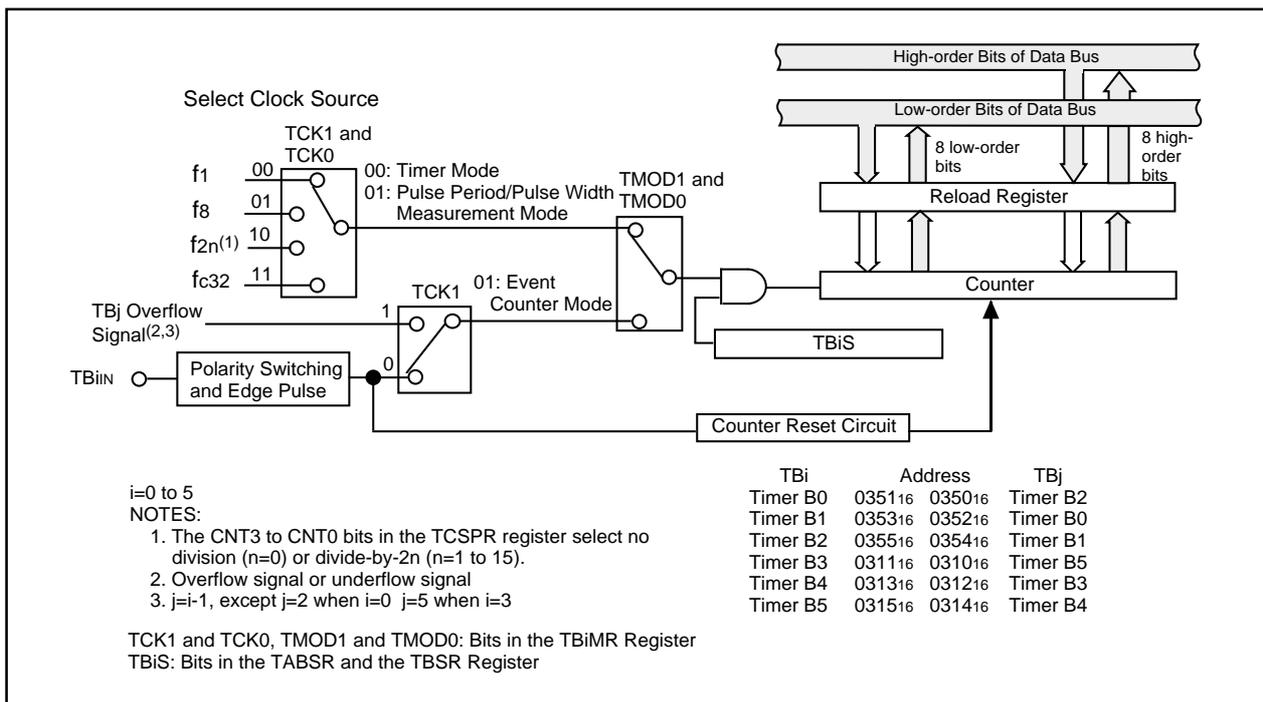


Figure 14.16 Timer B Block Diagram

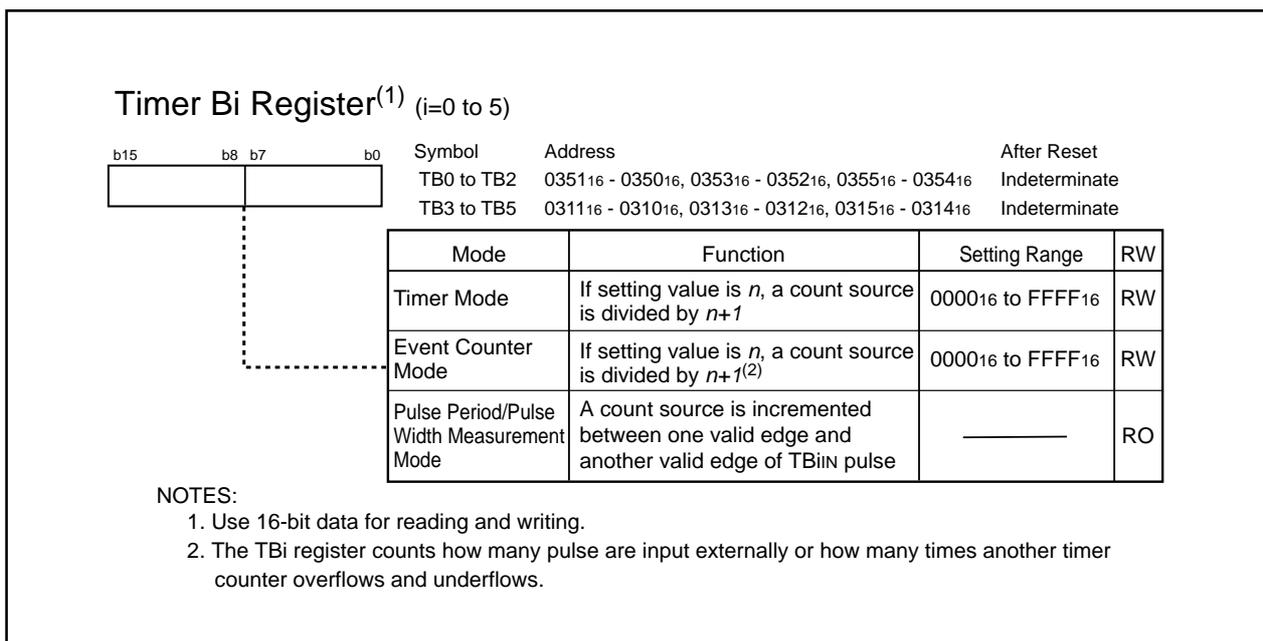


Figure 14.17 TB0 to TB5 Registers

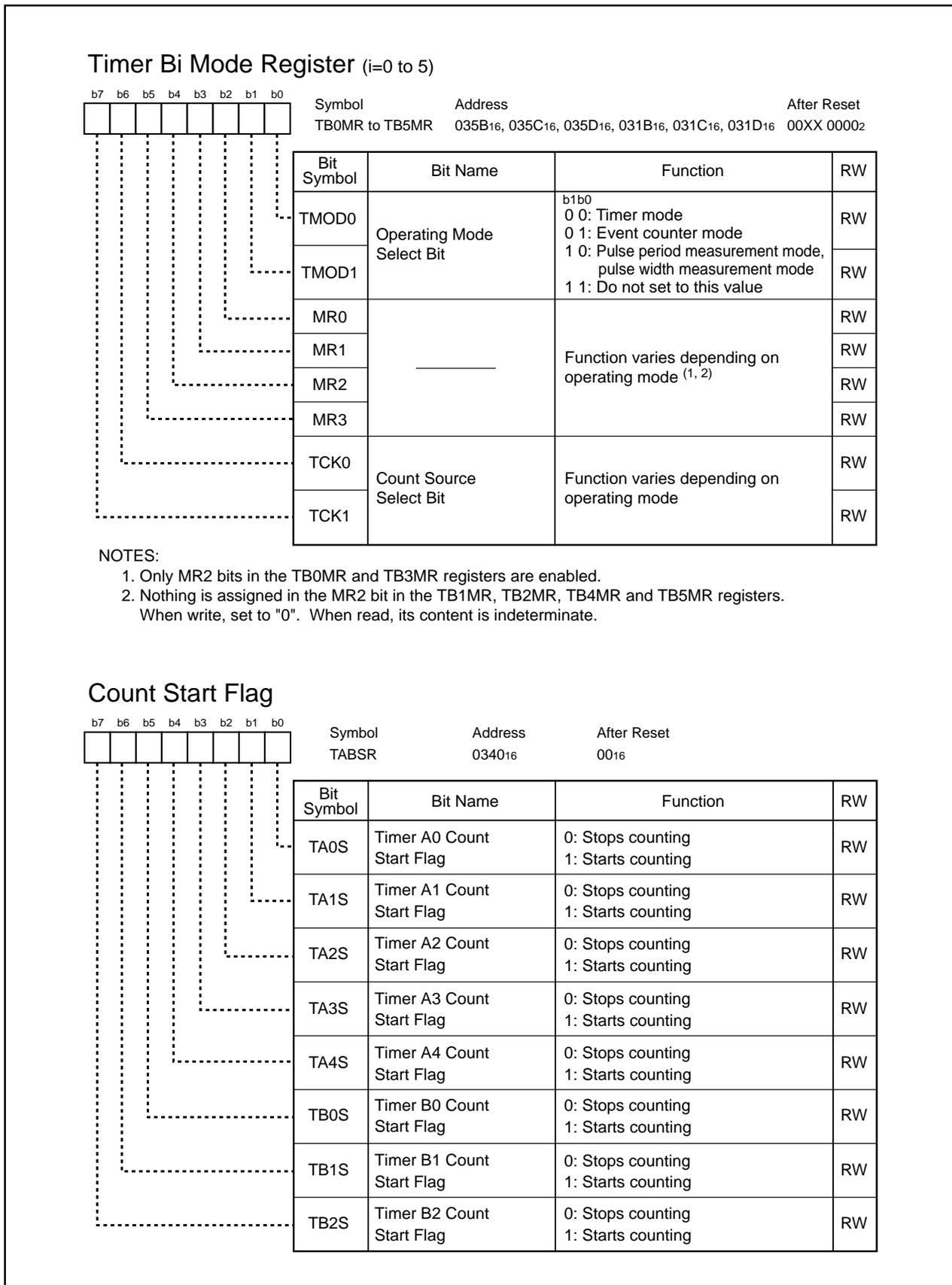
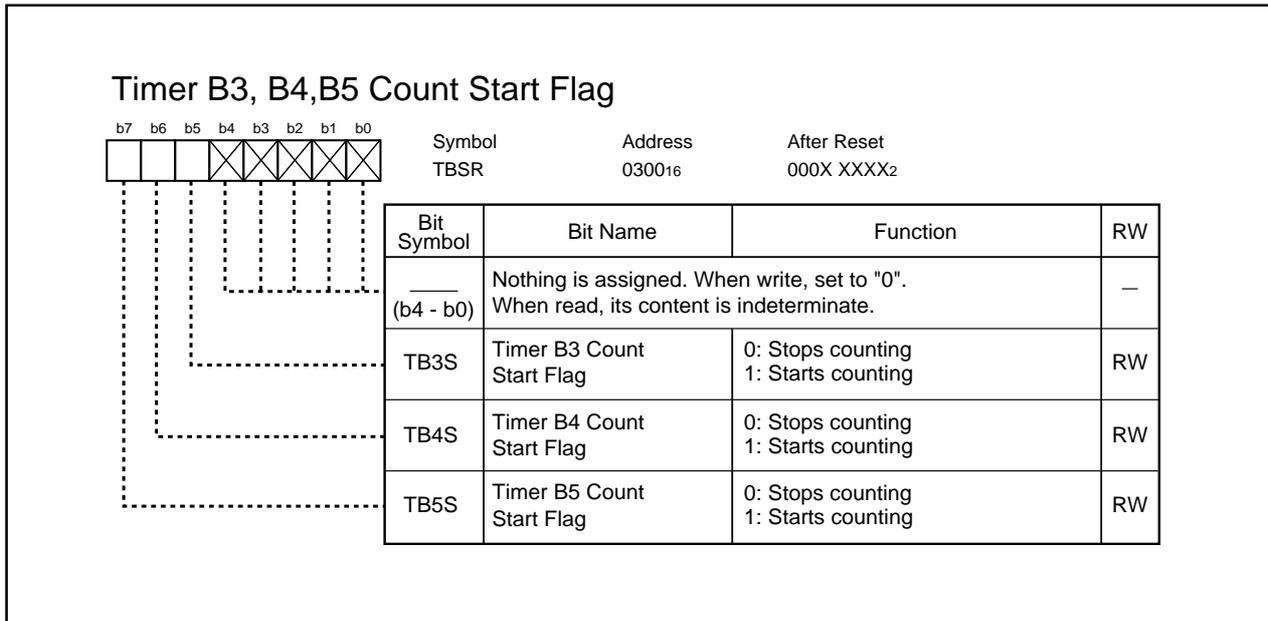


Figure 14.18 TB0MR to TB5MR Registers, TABSR Register



**Figure 14.19 TBSR Register**

**Table 14.8 Settings for the TBiIN Pins (i=0 to 5)**

Port Name	Function	Setting	
		PS1, PS3 <sup>(1)</sup> Registers	PD7, PD9 <sup>(1)</sup> Registers
P90	TB0IN	PS3_0=0	PD9_0=0
P91	TB1IN	PS3_1=0	PD9_1=0
P92	TB2IN	PS3_2=0	PD9_2=0
P93	TB3IN	PS3_3=0	PD9_3=0
P94	TB4IN	PS3_4=0	PD9_4=0
P71	TB5IN	PS1_1=0	PD7_1=0

**NOTE:**

1. Set the PD9 and PS3 registers immediately after the PRC2 bit in the PRCR register is set to "1" (write enabled). Do not generate an interrupt or a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to set the PD9 and PS3 registers.

### 14.2.1 Timer Mode

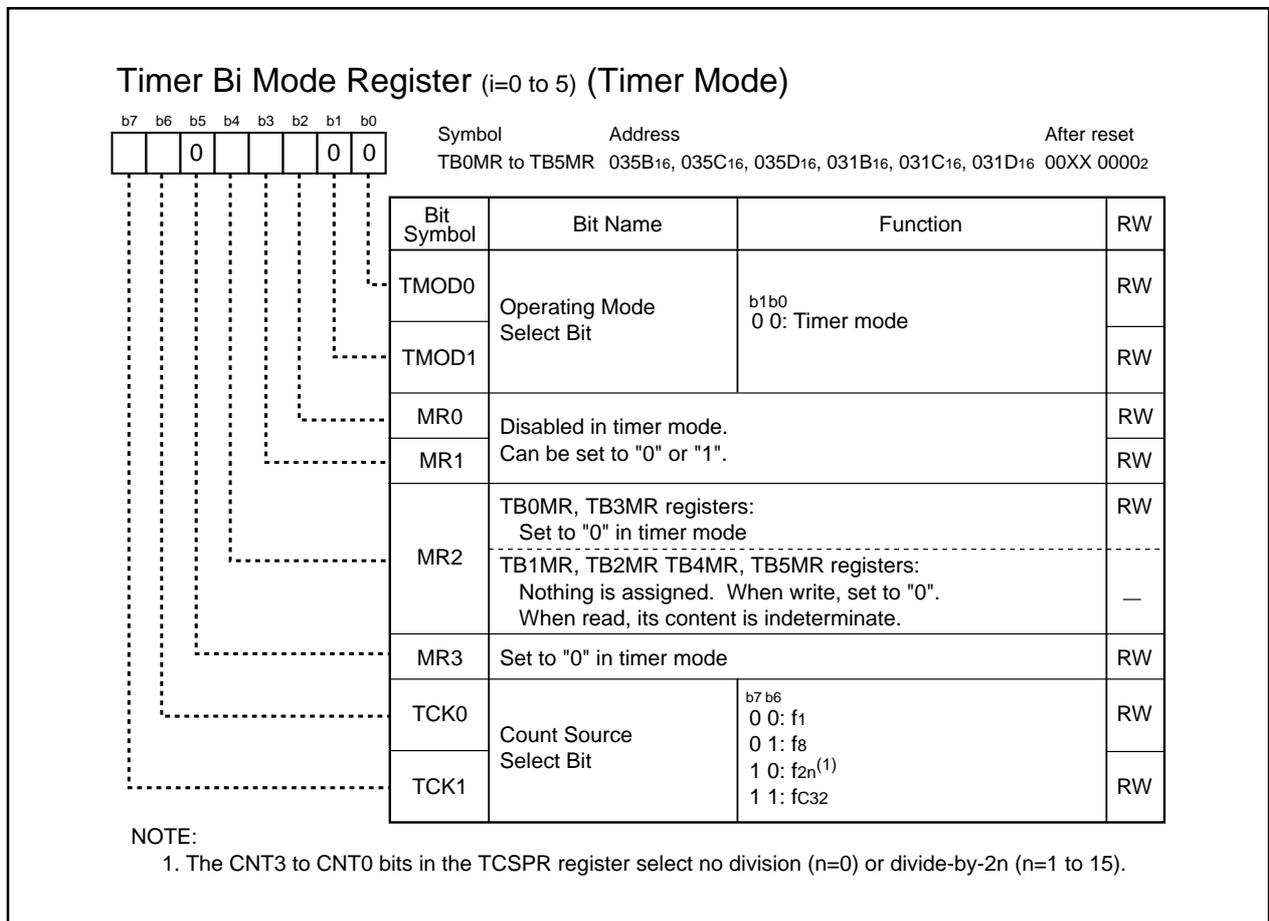
In timer mode, the timer counts an internally generated count source (see **Table 14.9**). Figure 14.20 shows the TBiMR register (i=0 to 5) in timer mode.

**Table 14.9 Timer Mode Specifications**

Item	Specification
Count Source	f1, f8, f2n <sup>(1)</sup> , fc32
Counting Operation	<ul style="list-style-type: none"> <li>The timer decrements a counter value</li> </ul> When the timer counter underflows, content of the reload register is reloaded into the count register and counting resumes
Divide Ratio	$1/(n+1)$ <i>n</i> : setting value of the TBi register (i=0 to 5) 0000 <sub>16</sub> to FFFF <sub>16</sub>
Counter Start Condition	The TBiS bits in the TABSR and TBSR registers are set to "1" (starts counting)
Counter Stop Condition	The TBiS bit is set to "0" (stops counting)
Interrupt Request Generation Timing	Timer counter underflows
TBiIn Pin Function	Programmable I/O port
Read from Timer	The TBi register indicates counter value
Write to Timer	<ul style="list-style-type: none"> <li>When the timer counter stops, the value written to the TBi register is also written to both reload register and counter</li> <li>While counting, the value written to the TBi register is written to the reload register (It is transferred to the counter at the next reload timing)</li> </ul>

NOTE:

- The CNT3 to CNT0 bits in the TCSPR register select no division (n=0) or divide-by-2n (n=1 to 15).



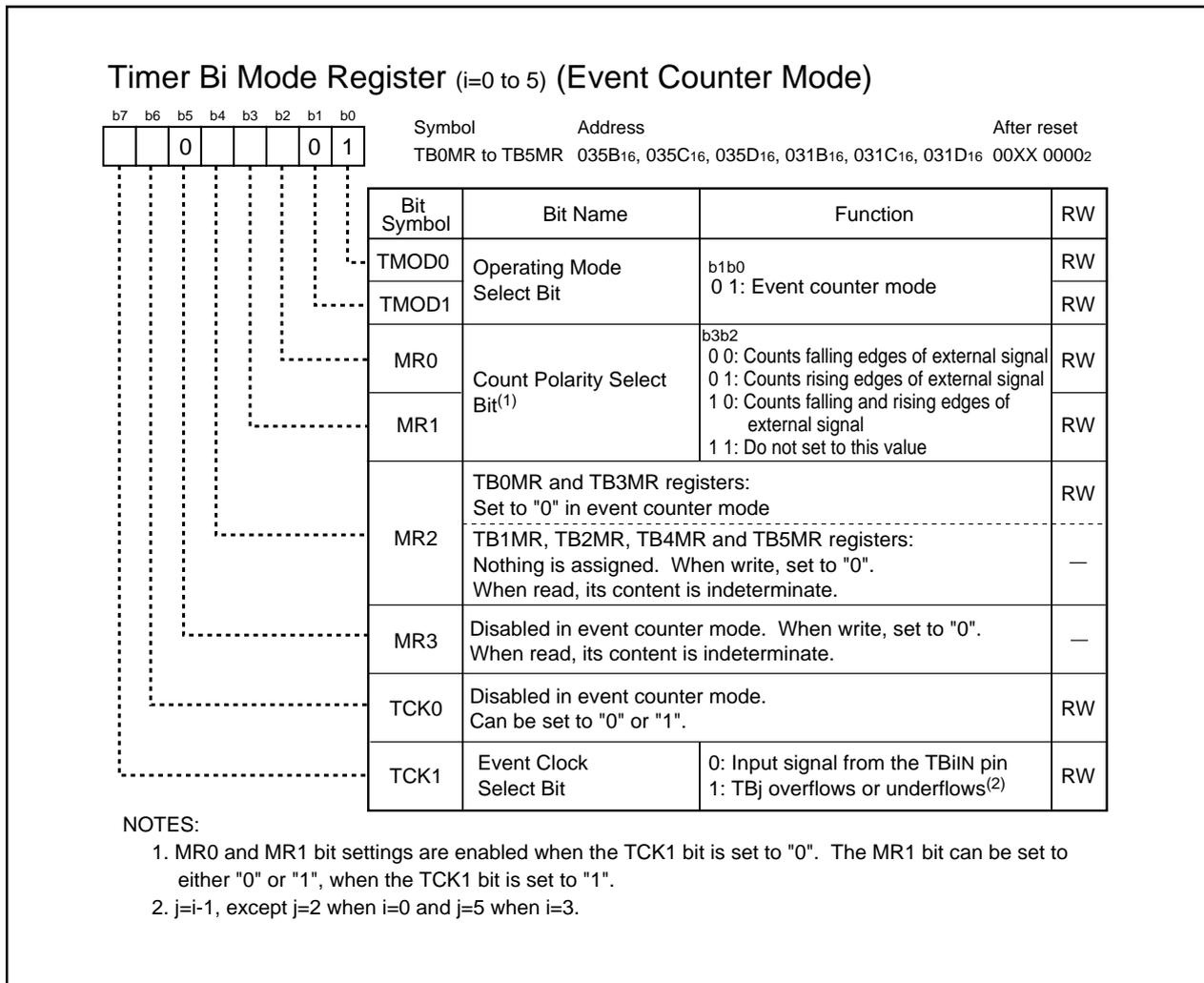
**Figure 14.20 TB0MR to TB5MR Registers**

### 14.2.2 Event Counter Mode

In event counter mode, the timer counts how many external signals are applied or how many times another timer overflows and underflows. (See **Table 14.10**) Figure 14.21 shows the TBiMR register (i=0 to 5) in event counter mode.

**Table 14.10 Event Counter Mode Specifications**

Item	Specification
Count Source	<ul style="list-style-type: none"> <li>External signal applied to the TBiIN pin (i = 0 to 5) (valid edge can be selected by program)</li> <li>TBj overflow or underflow signal (j=i-1, except j=2 when i=0, j=5 when i=3)</li> </ul>
Counting Operation	<ul style="list-style-type: none"> <li>The timer decrements a counter value</li> </ul> <p>When the timer counter underflows, content of the reload register is reloaded into the count register to continue counting</p>
Divide Ratio	$1/(n+1)$ $n$ : setting value of the TBi register      0000 <sub>16</sub> to FFFF <sub>16</sub>
Counter Start Condition	The TBiS bits in the TABSR and TBSR register are set to "1" (starts counting)
Counter Stop Condition	The TBiS bit is set to "0" (stops counting)
Interrupt Request Generation Timing	The timer counter underflows
TBiIN Pin Function	Programmable I/O port or count source input
Read from Timer	The TBi register indicates counter value
Write to Timer	<ul style="list-style-type: none"> <li>When the timer counter stops, the value written to the TBi register is also written to both reload register and counter</li> <li>While counting, the value written to the TBi register is written to the reload register (It is transferred to the counter at the next reload timing)</li> </ul>



**Figure 14.21 TB0MR to TB5MR Registers**

### 14.2.3 Pulse Period/Pulse Width Measurement Mode

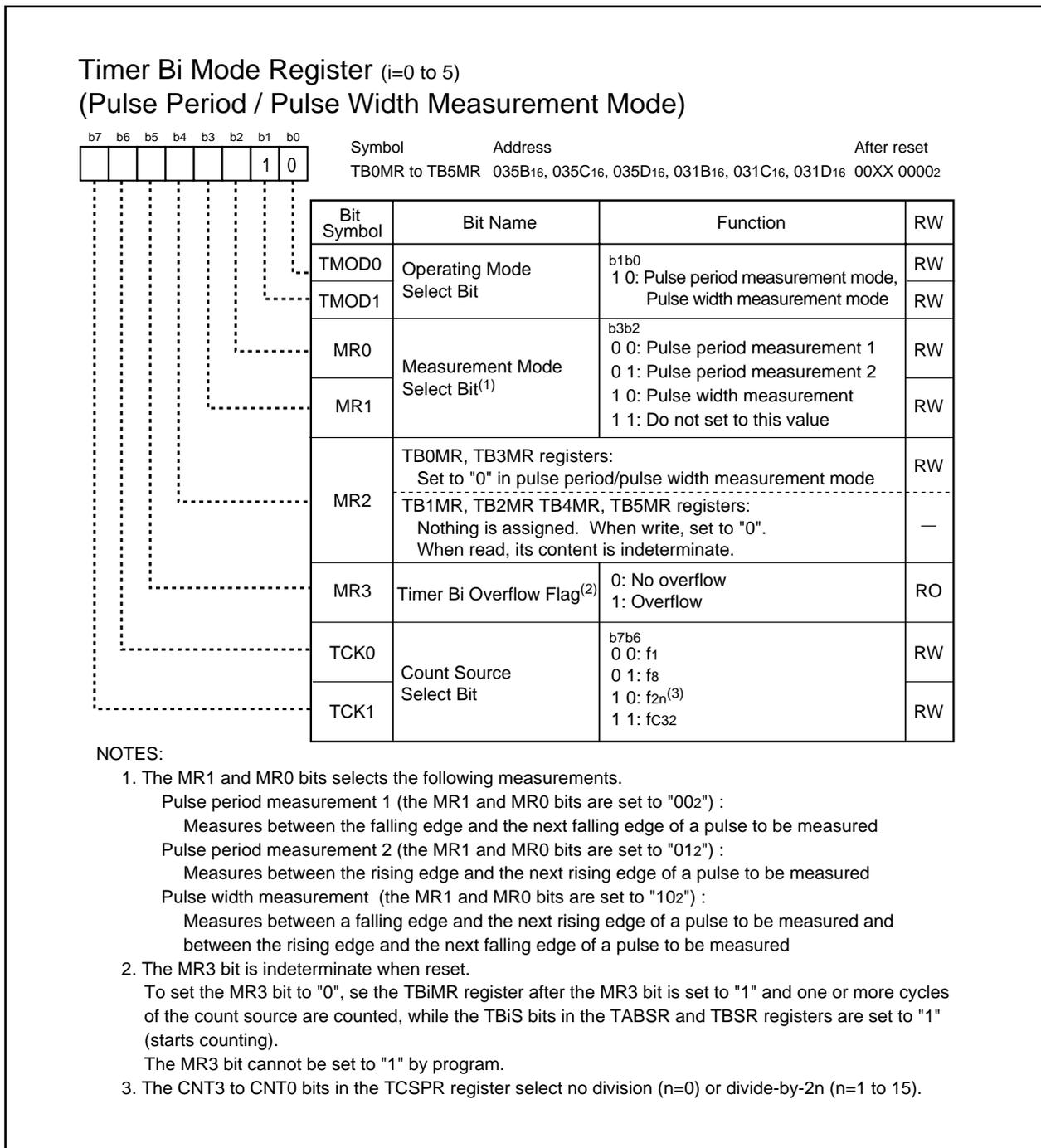
In pulse period/pulse width measurement mode, the timer measures pulse period or pulse width of an external signal. (See **Table 14.11**) Figure 14.22 shows the TBiMR register (i=0 to 5) in pulse period/pulse width measurement mode. Figure 14.23 shows an operation example in pulse period measurement mode. Figure 14.24 shows an operation example in the pulse width measurement mode.

**Table 14.11 Pulse Period/Pulse Width Measurement Mode Specifications**

Item	Specification
Count Source	f1, f8, f2n <sup>(3)</sup> , fc32
Counting Operation	<ul style="list-style-type: none"> <li>The timer increments a counter value</li> </ul> Counter value is transferred to the reload register on the valid edge of a pulse to be measured. It is set to "000016" and the timer continues counting
Counter Start Condition	The TBiS bits (i=0 to 5) in the TABSR and TBSR register are set to "1" (starts counting)
Counter Stop Condition	The TBiS bit is set to "0" (stops counting)
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>On the valid edge of a pulse to be measured<sup>(1)</sup></li> <li>The timer counter overflows</li> </ul> The MR3 bit in the TBiMR register is set to "1" (overflow) simultaneously. When the TBiS bit is set to "1" (start counting) and the next count source is counted after setting the MR3 bit to "1" (overflow), the MR3 bit can be set to "0" (no overflow) by writing to the TBiMR register.
TBiIN Pin Function	Input for a pulse to be measured
Read from Timer	The TBi register indicates reload register values (measurement results) <sup>(2)</sup>
Write to Timer	Value written to the TBi register can be written to neither reload register nor counter

**NOTES:**

1. No interrupt request is generated when the pulse to be measured is on the first valid edge after the timer has started counting.
2. The TBi register is in an indeterminate state until the pulse to be measured is on the second valid edge after the timer has started counting.
3. The CNT3 to CNT0 bits in the TCSPR register select no division (n=0) or divide-by-2n (n=1 to 15).

**Figure 14.22 TB0MR to TB5MR Registers**

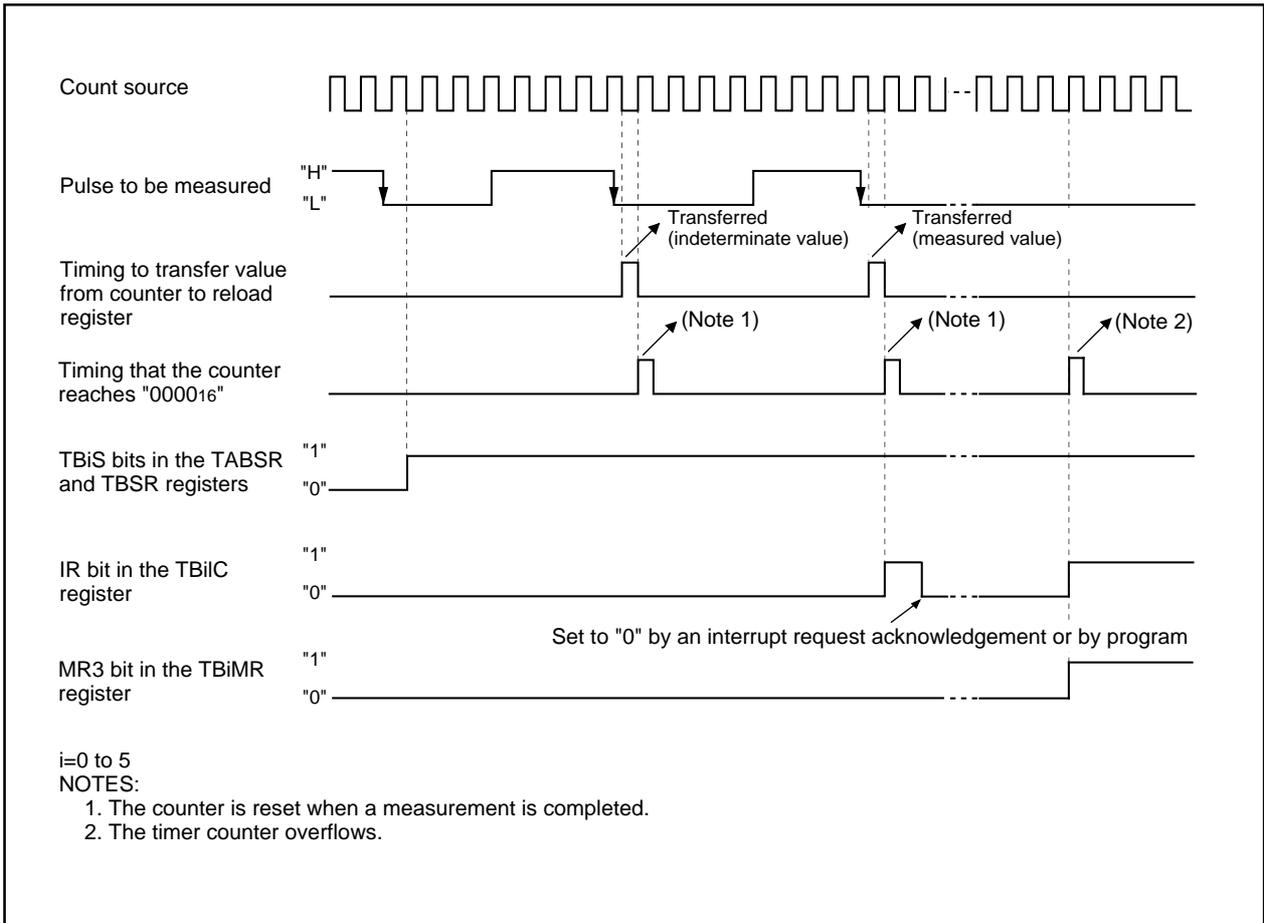


Figure 14.23 Operation Example in Pulse Period Measurement Mode

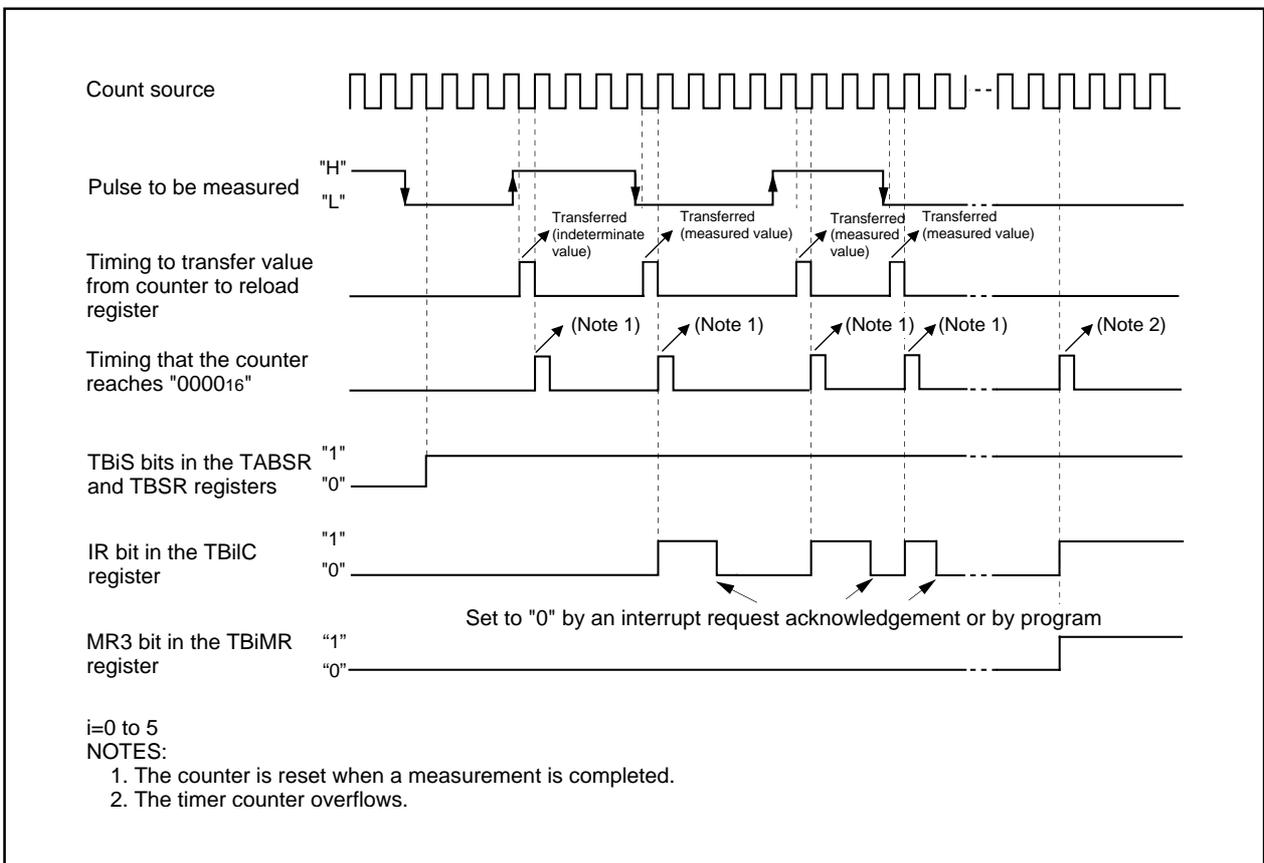


Figure 14.24 Operation Example in Pulse Width Measurement Mode

## 15. Three-Phase Motor Control Timer Functions

Three-phase motor driving waveform can be output by using the timers A1, A2, A4 and B2. Table 15.1 lists specifications of the three-phase motor control timer functions. Table 15.2 lists pin settings. Figure 15.1 shows a block diagram. Figures 15.2 to 15.7 show registers associated with the three-phase motor control timer functions.

**Table 15.1 Three-Phase Motor Control Timer Functions Specification**

Item	Specification
Three-Phase Waveform Output Pin	Six pins (U, $\bar{U}$ , V, $\bar{V}$ , W, $\bar{W}$ )
Forced Cutoff <sup>(1)</sup>	Apply a low-level ("L") signal to the NMI pin
Timers to be Used	Timer A4, A1, A2 (used in one-shot timer mode): Timer A4: U- and $\bar{U}$ -phase waveform control Timer A1: V- and $\bar{V}$ -phase waveform control Timer A2: W- and $\bar{W}$ -phase waveform control Timer B2 (used in timer mode): Carrier wave cycle control Dead time timer (three 8-bit timers share reload register): Dead time control
Output Waveform	Triangular wave modulation, Sawtooth wave modulation Can output a high-level waveform or a low-level waveform for one cycle; Can set positive-phase level and negative-phase level separately
Carrier Wave Cycle	Triangular wave modulation: $count\ source \times (m+1) \times 2$ Sawtooth wave modulation: $count\ source \times (m+1)$ $m$ : setting value of the TB2 register, 0000 <sub>16</sub> to FFFF <sub>16</sub> Count source: f <sub>1</sub> , f <sub>8</sub> , f <sub>2n</sub> <sup>(2)</sup> , f <sub>c32</sub>
Three-Phase PWM Output Width	Triangular wave modulation: $count\ source \times n \times 2$ Sawtooth wave modulation: $count\ source \times n$ $n$ : setting value of the TA4, TA1 and TA2 register (of the TA4, TA41, TA1, TA11, TA2 and TA21 registers when setting the INV11 bit to "1"), 0001 <sub>16</sub> to FFFF <sub>16</sub> Count source: f <sub>1</sub> , f <sub>8</sub> , f <sub>2n</sub> <sup>(2)</sup> , f <sub>c32</sub>
Dead Time	$Count\ source \times p$ , or no dead time $p$ : setting value of the DTT register, 01 <sub>16</sub> to FF <sub>16</sub> Count source: f <sub>1</sub> , or f <sub>1</sub> divided by 2
Active Level	Selected from a high level ("H") or low level ("L")
Positive- and Negative-Phase Concurrent Active Disable Function	Positive and negative-phases concurrent active disable function Positive and negative-phases concurrent active detect function
Interrupt Frequency	For the timer B2 interrupt, one carrier wave cycle-to-cycle basis through 15 time- carrier wave cycle-to-cycle basis can be selected

### NOTES:

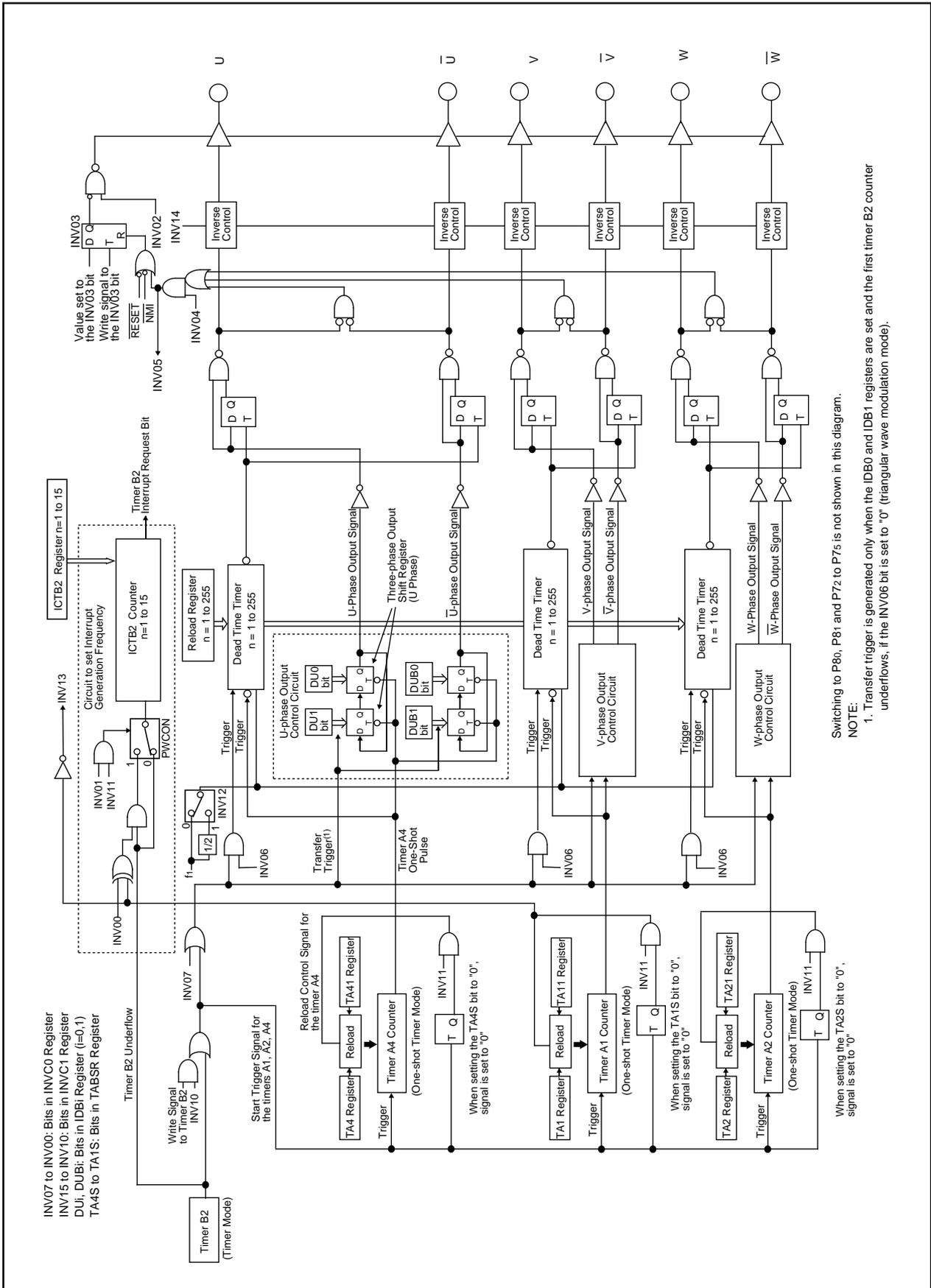
1. Forced cutoff by the signal applied to the  $\overline{NMI}$  pin is available when the INV02 bit is set to "1" (three-phase motor control timer functions) and the INV03 bit is set to "1" (three-phase motor control timer output enabled).
2. The CNT3 to CNT0 bits in the TCSPPR register select no division (n=0) or divide-by-2n (n=1 to 15).

**Table 15.2 Pin Settings**

Pin	Setting		
	PS1, PS2 Registers <sup>(1)</sup>	PSL1, PSL2 Registers	PSC Register
P72/V	PS1_2 =1	PSL1_2 =0	PSC_2 =1
P73/ $\bar{V}$	PS1_3 =1	PSL1_3 =1	PSC_3 =0
P74/W	PS1_4 =1	PSL1_4 =1	PSC_4 =0
P75/ $\bar{W}$	PS1_5 =1	PSL1_5 =0	—
P80/U	PS2_0 =1	PSL2_0 =1	—
P81/ $\bar{U}$	PS2_1 =1	PSL2_1 =0	—

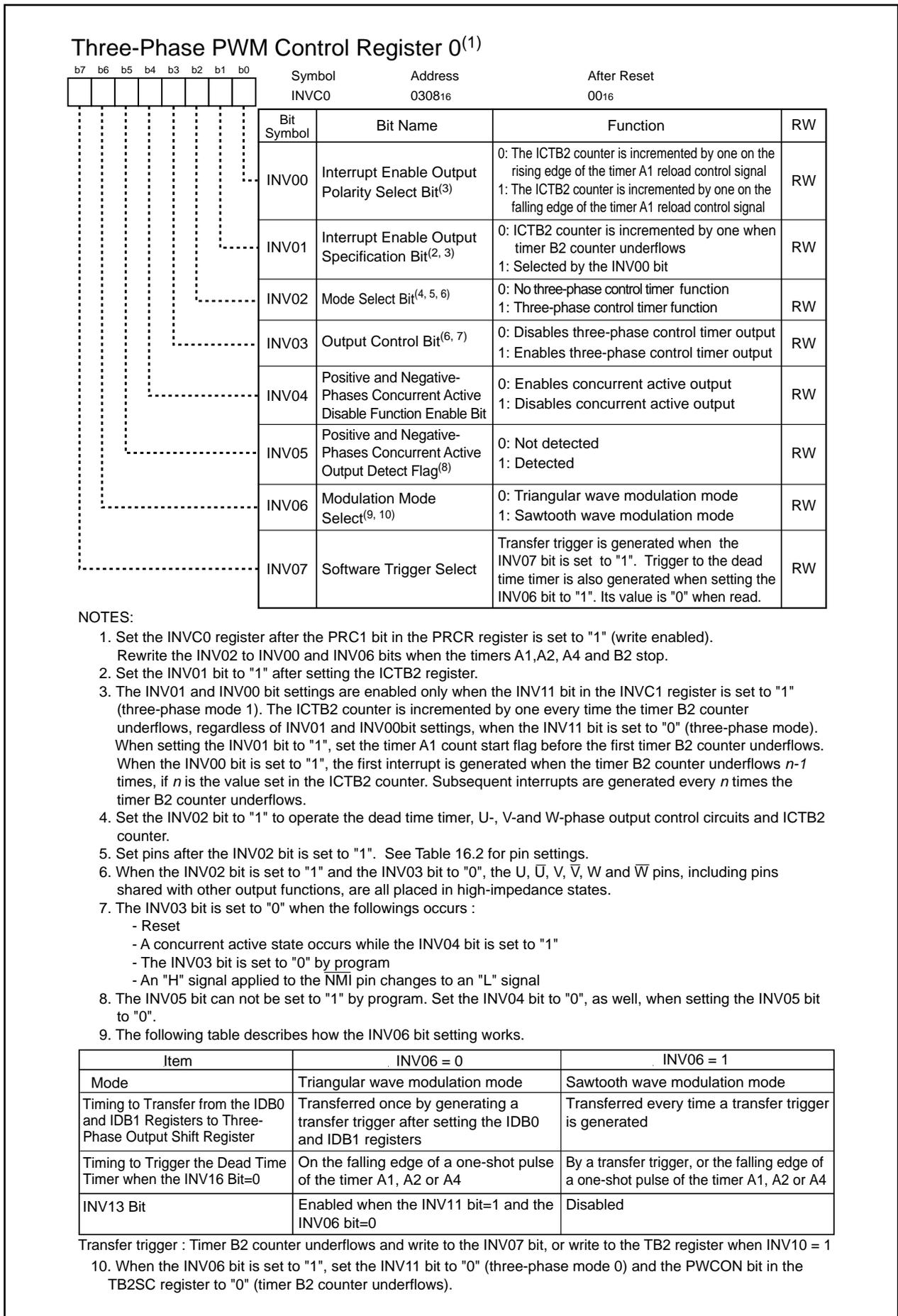
**NOTE:**

1. Set the PS1\_5 to PS1\_2 bits and PS2\_1 and PS2\_0 bits in the PS1 and PS2 registers to "1" after the INV02 bit is set to "1".

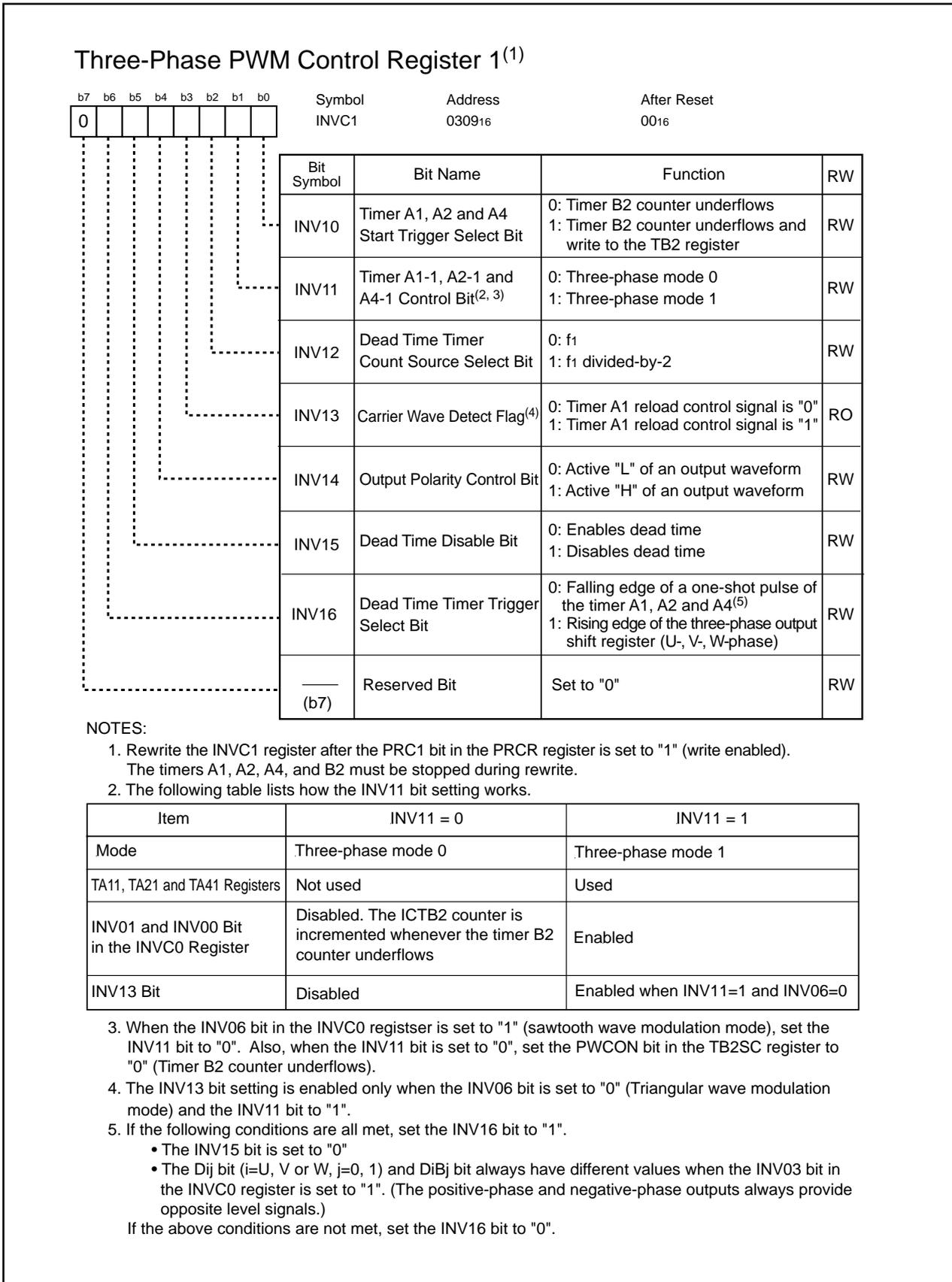


Switching to P80, P81 and P72 to P75 is not shown in this diagram.  
 NOTE:  
 1. Transfer trigger is generated only when the IDB0 and IDB1 registers are set and the first timer B2 counter underflows, if the INV06 bit is set to "0" (triangular wave modulation mode).

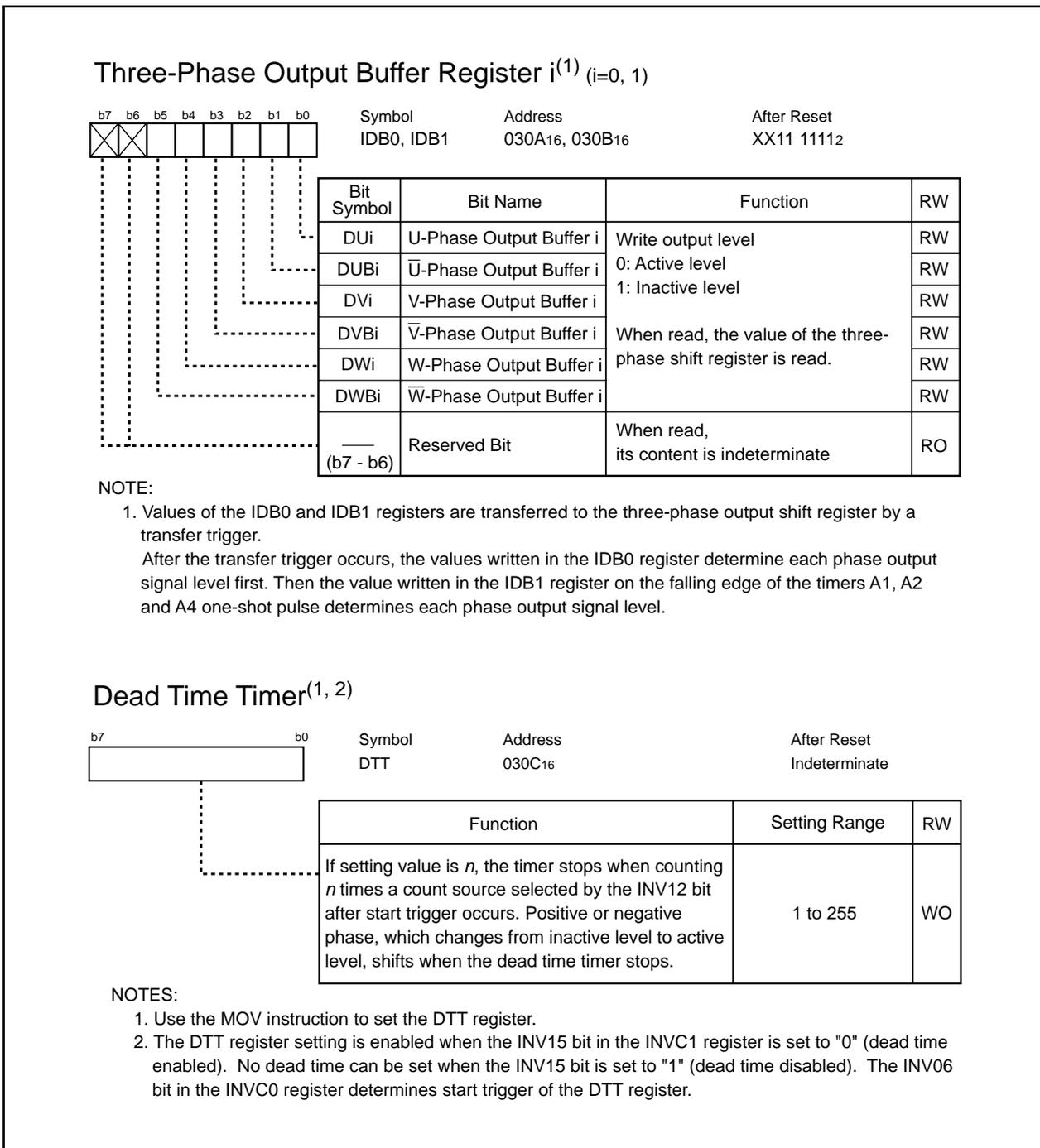
Figure 15.1 Three-Phase Motor Control Timer Functions Block Diagram



**Figure 15.2 INVC0 Register**



**Figure 15.3 INVC1 Register**



**Figure 15.4 IDB0 and IDB1 registers, DTT Register**



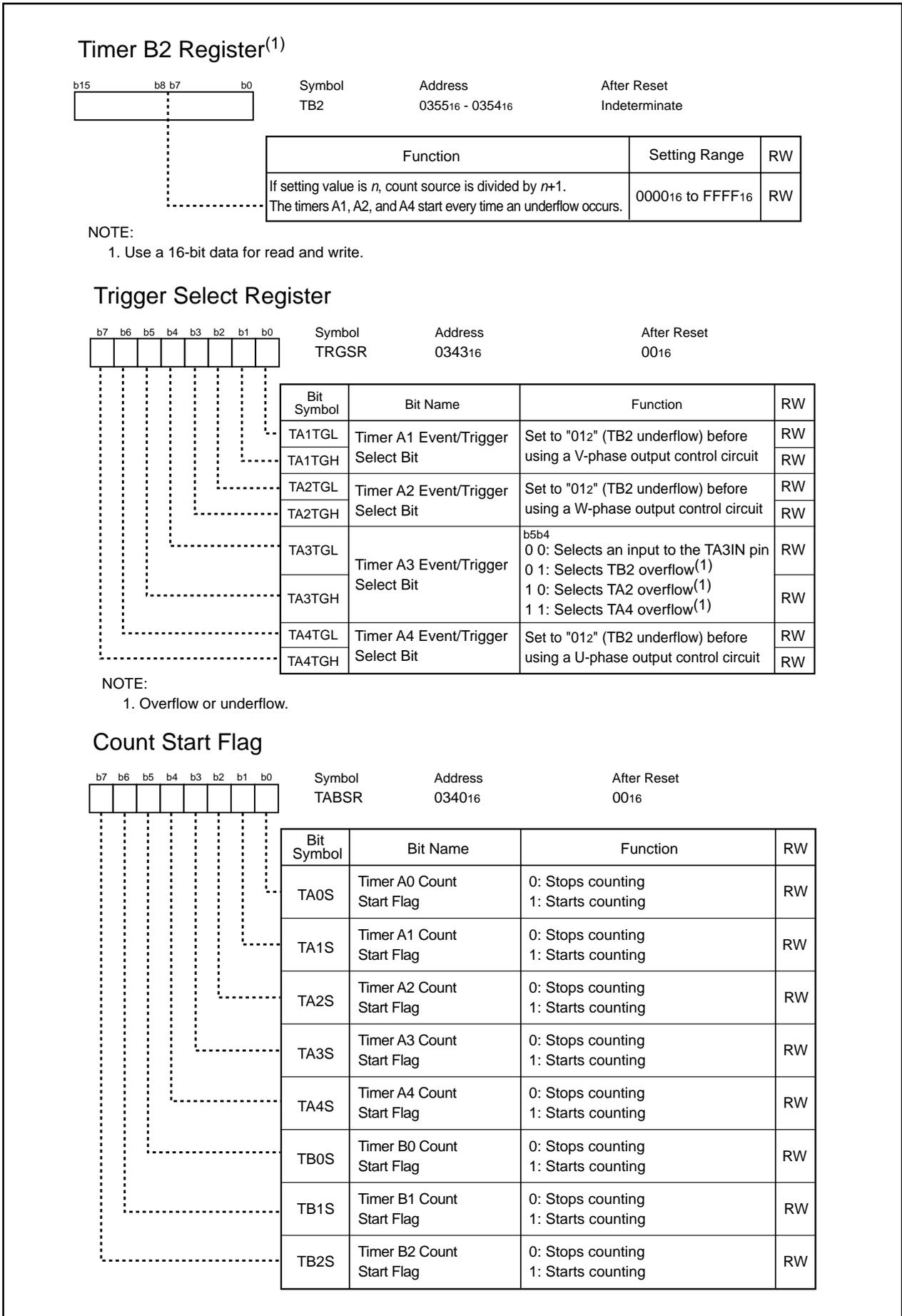


Figure 15.6 TB2, TRGSR, and TABSR Registers

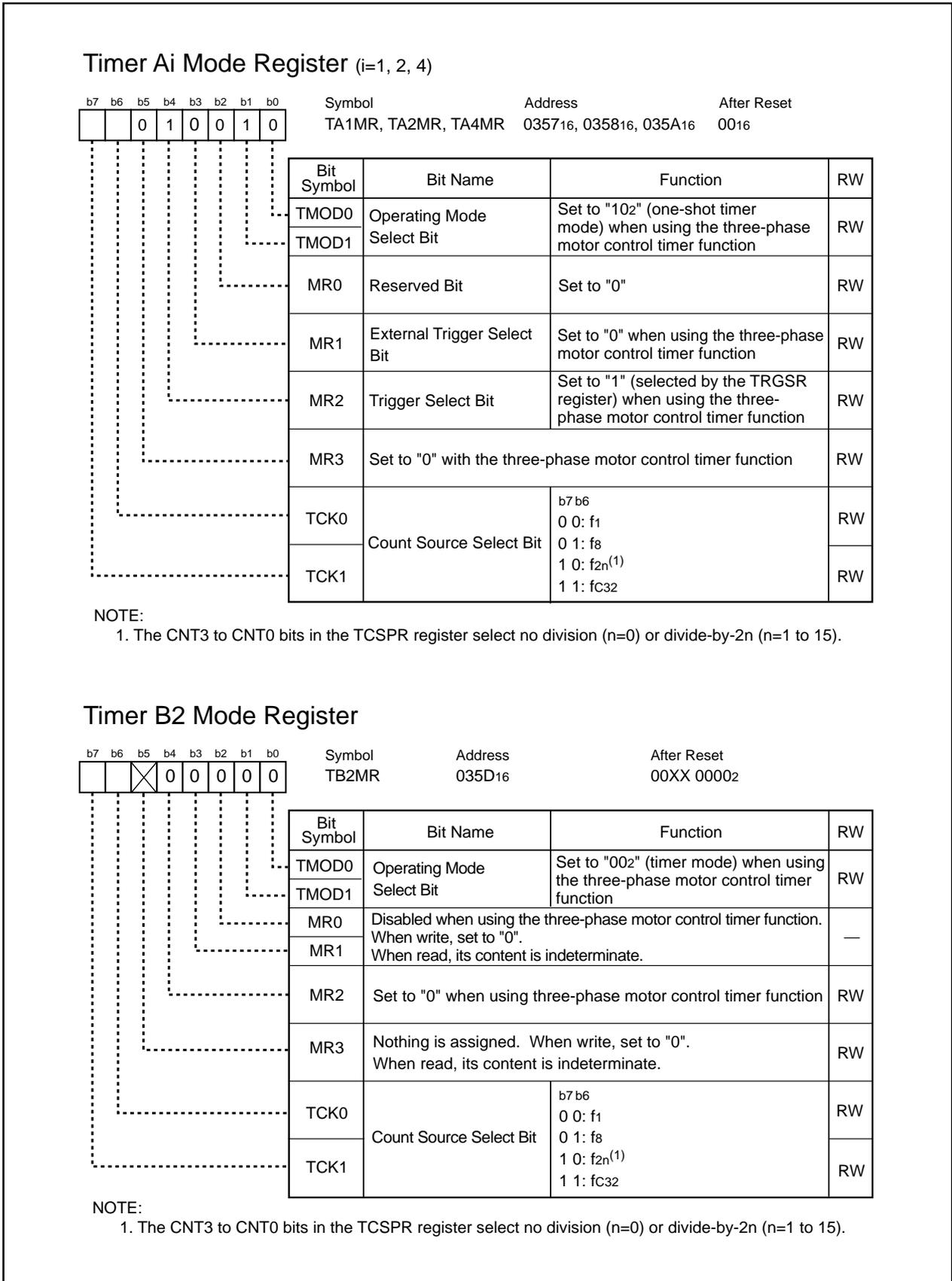


Figure 15.7 TA1MR, TA2MR, and TA4MR Registers, TB2MR Register

The three-phase motor control timer function is available by setting the INV02 bit in the INVC0 register to "1". The timer B2 is used for carrier wave control and the timers A1, A2, A4 for three-phase PWM output (U,  $\bar{U}$ , V,  $\bar{V}$ , W,  $\bar{W}$ ) control. An exclusive dead time timer controls dead time. Figure 15.8 shows an example of the triangular modulation waveform. Figure 15.9 shows an example of the sawtooth modulation waveform.

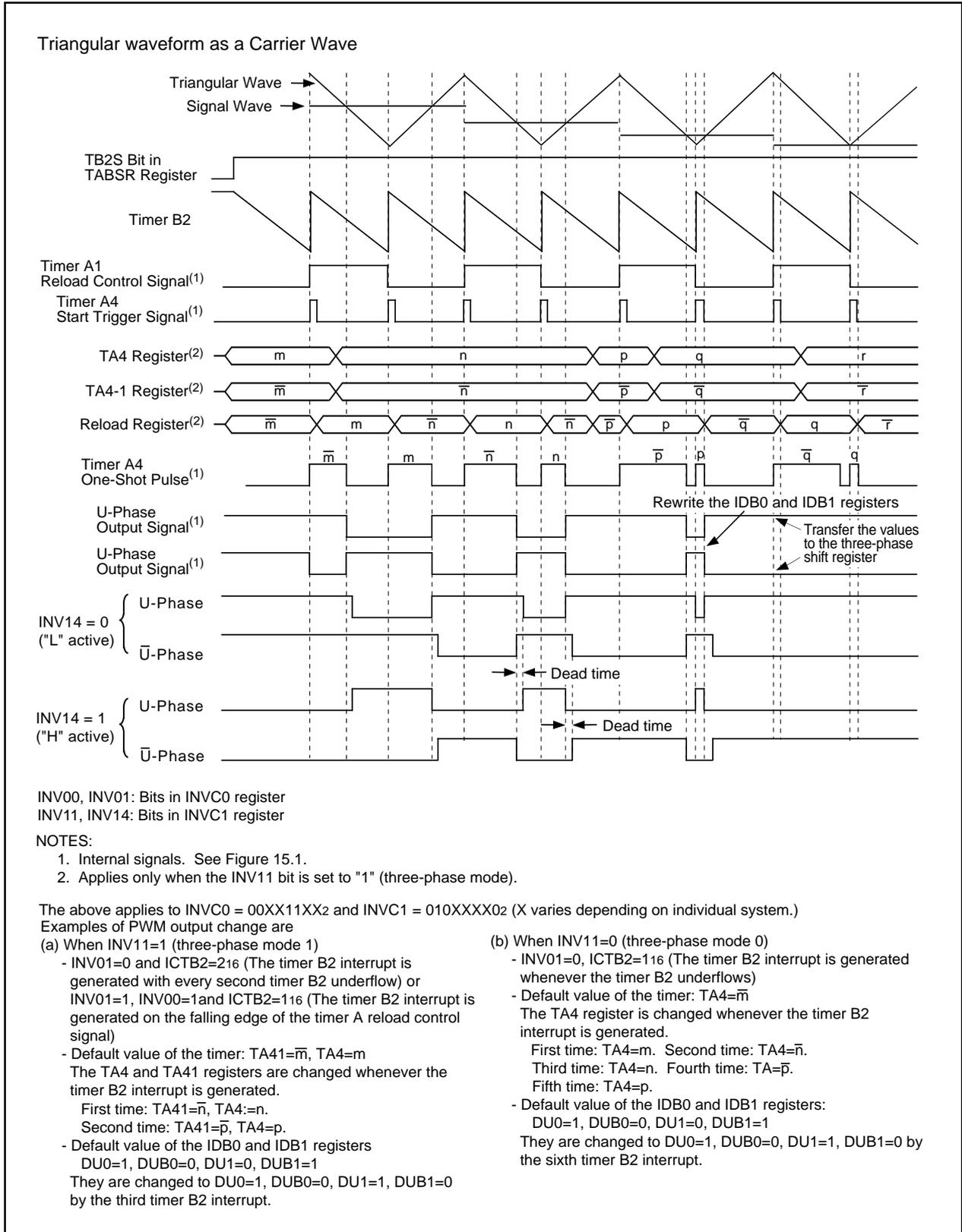


Figure 15.8 Triangular Wave Modulation Operation

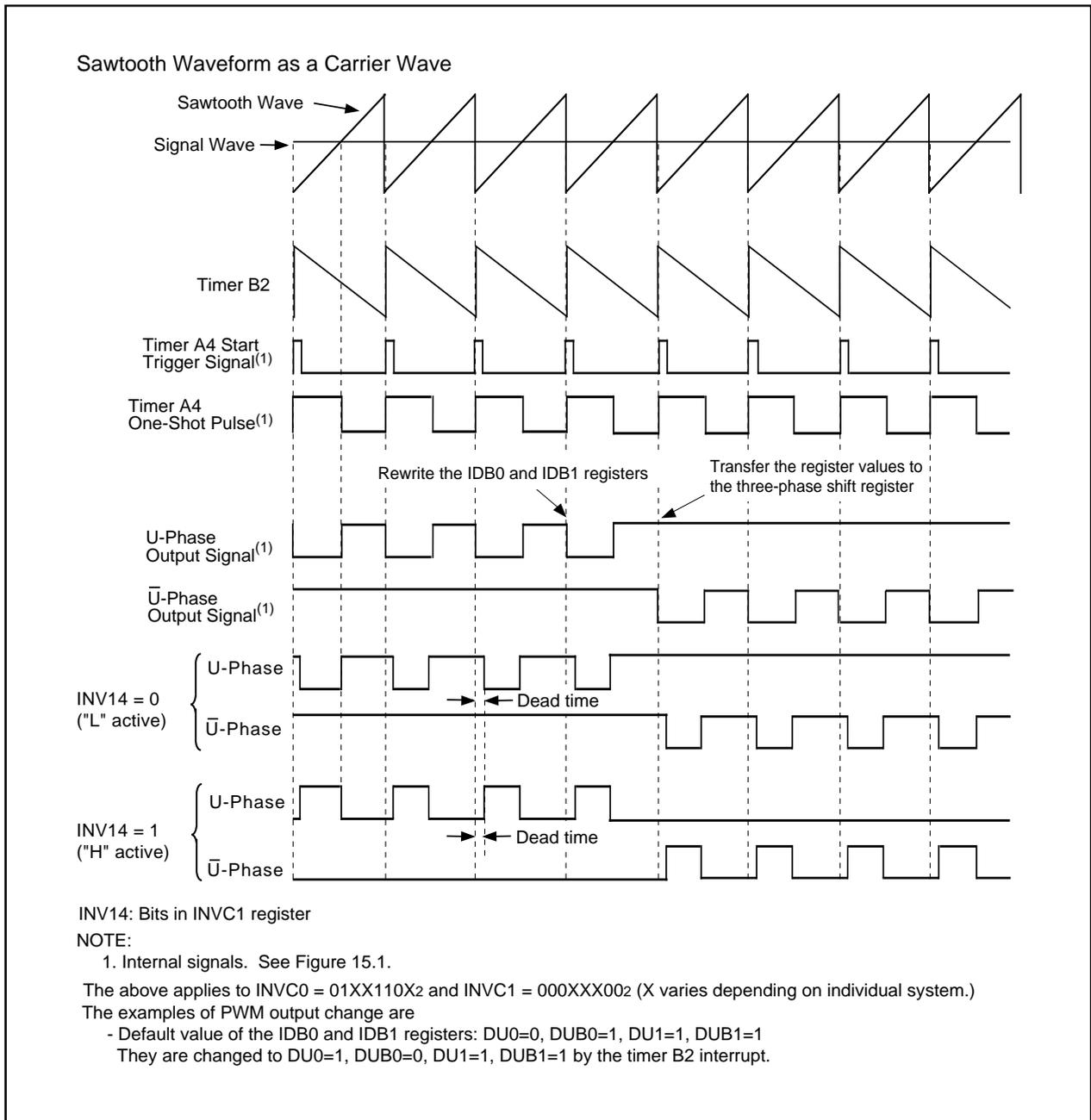


Figure 15.9 Sawtooth Wave Modulation Operation

## 16. Serial I/O

Serial I/O consists of five channels (UART0 to UART4).

Each UART<sub>i</sub> (i=0 to 4) has an exclusive timer to generate the transfer clock and operates independently.

Figure 16.1 shows a UART<sub>i</sub> block diagram.

UART<sub>i</sub> supports the following modes :

- Clock synchronous serial I/O mode
- Clock asynchronous serial I/O mode (UART mode)
- Special mode 1 (I<sup>2</sup>C mode)
- Special mode 2
- Special mode 3 (Clock-divided synchronous function, GCI mode)
- Special mode 4 (Bus conflict detect function, IE mode)
- Special mode 5 (SIM mode)

Figures 16.2 to 16.9 show registers associated with UART<sub>i</sub>.

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

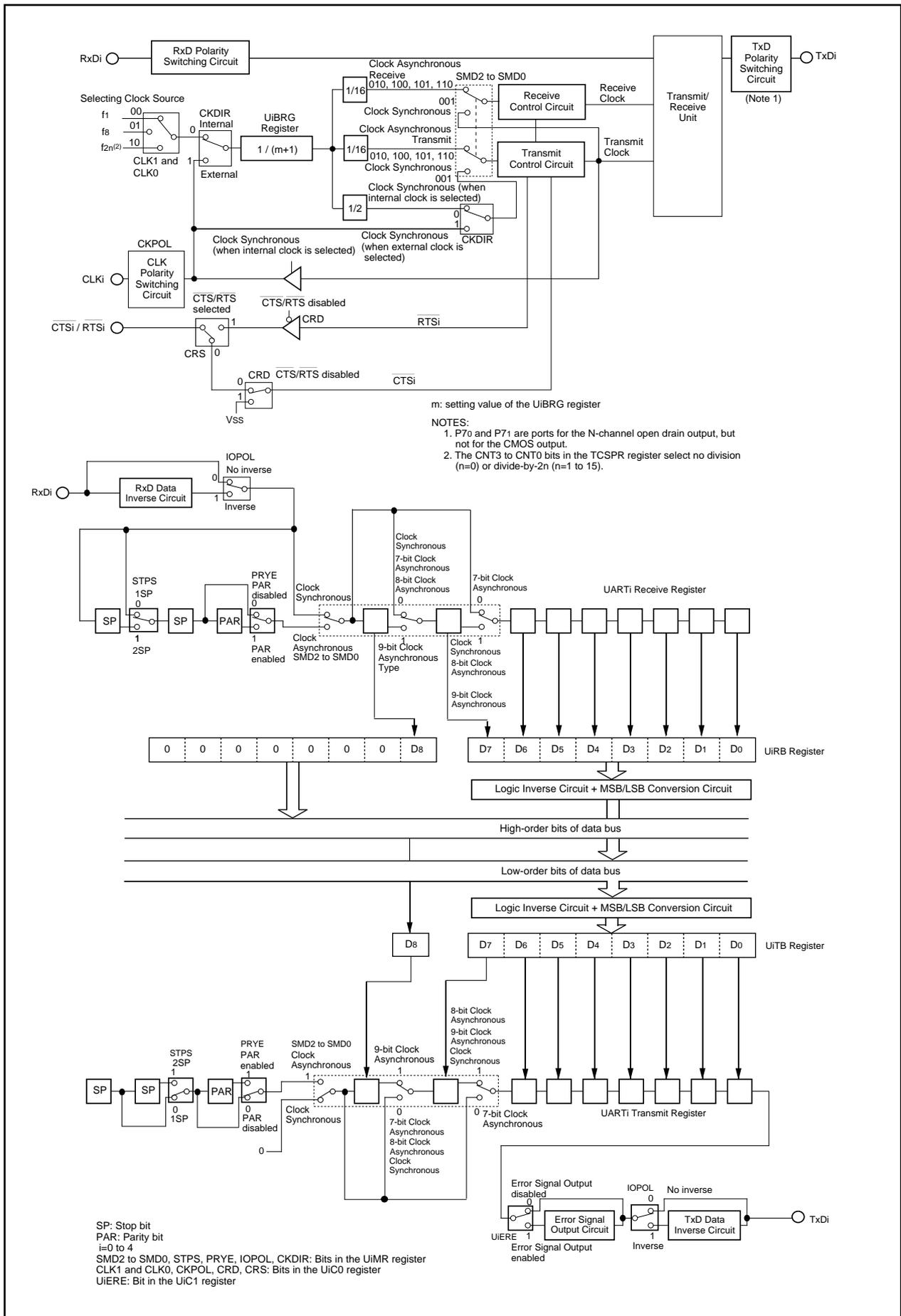
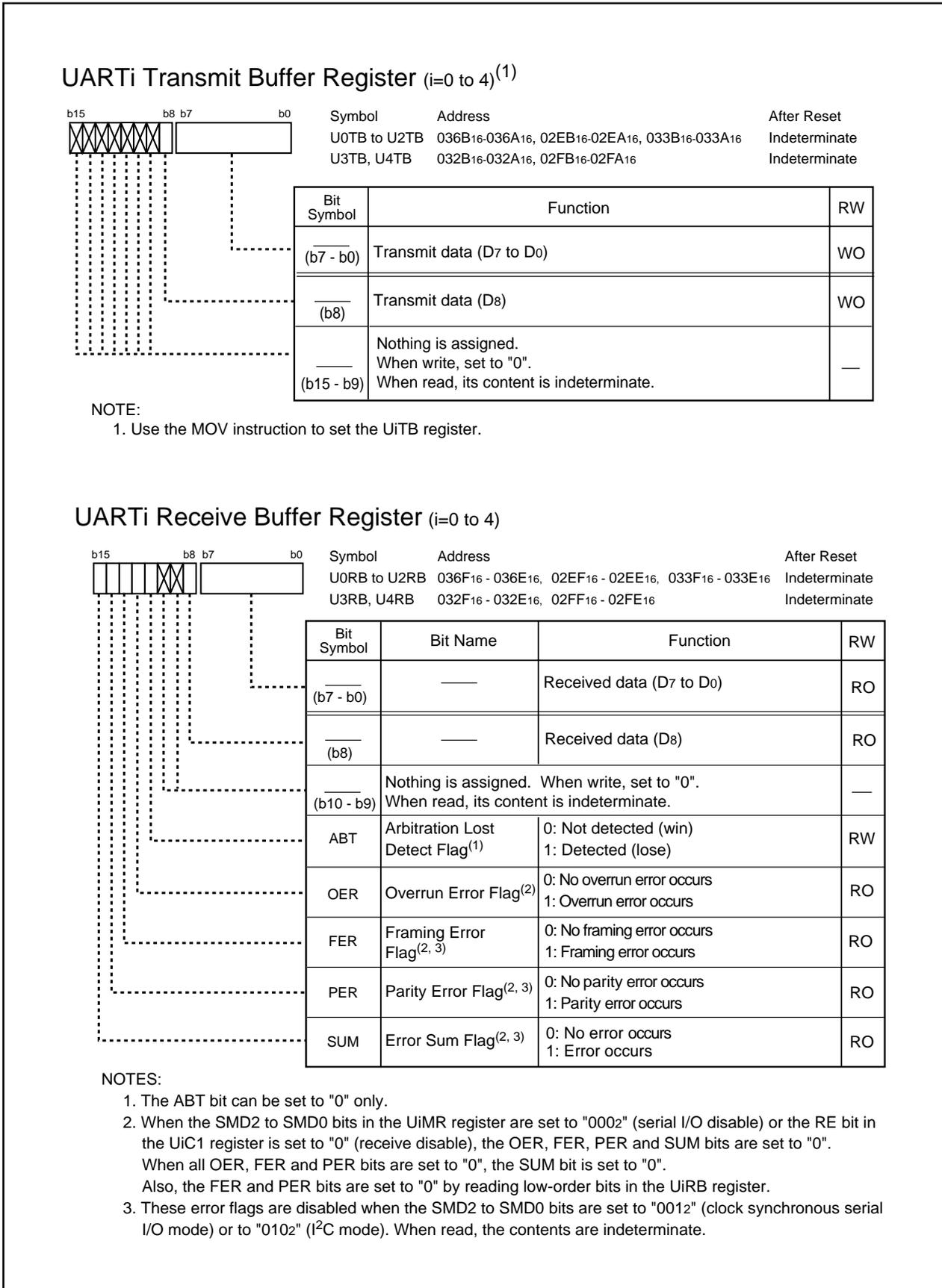
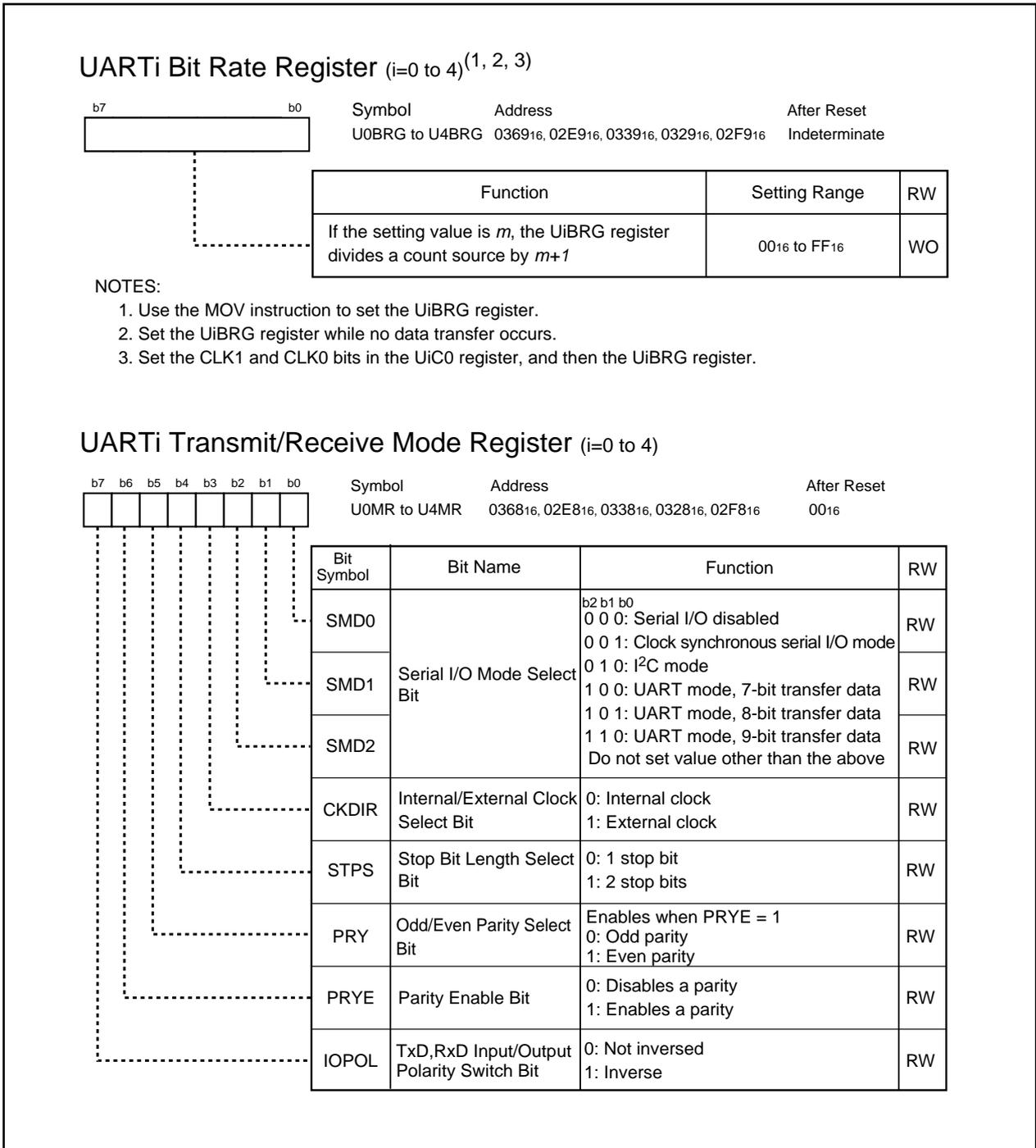


Figure 16.1 UARTi Block Diagram



**Figure 16.2 U0TB to U4TB Registers and U0RB to U4RB Registers**



**Figure 16.3 U0BRG to U4BRG Registers and U0MR to U4MR Registers**

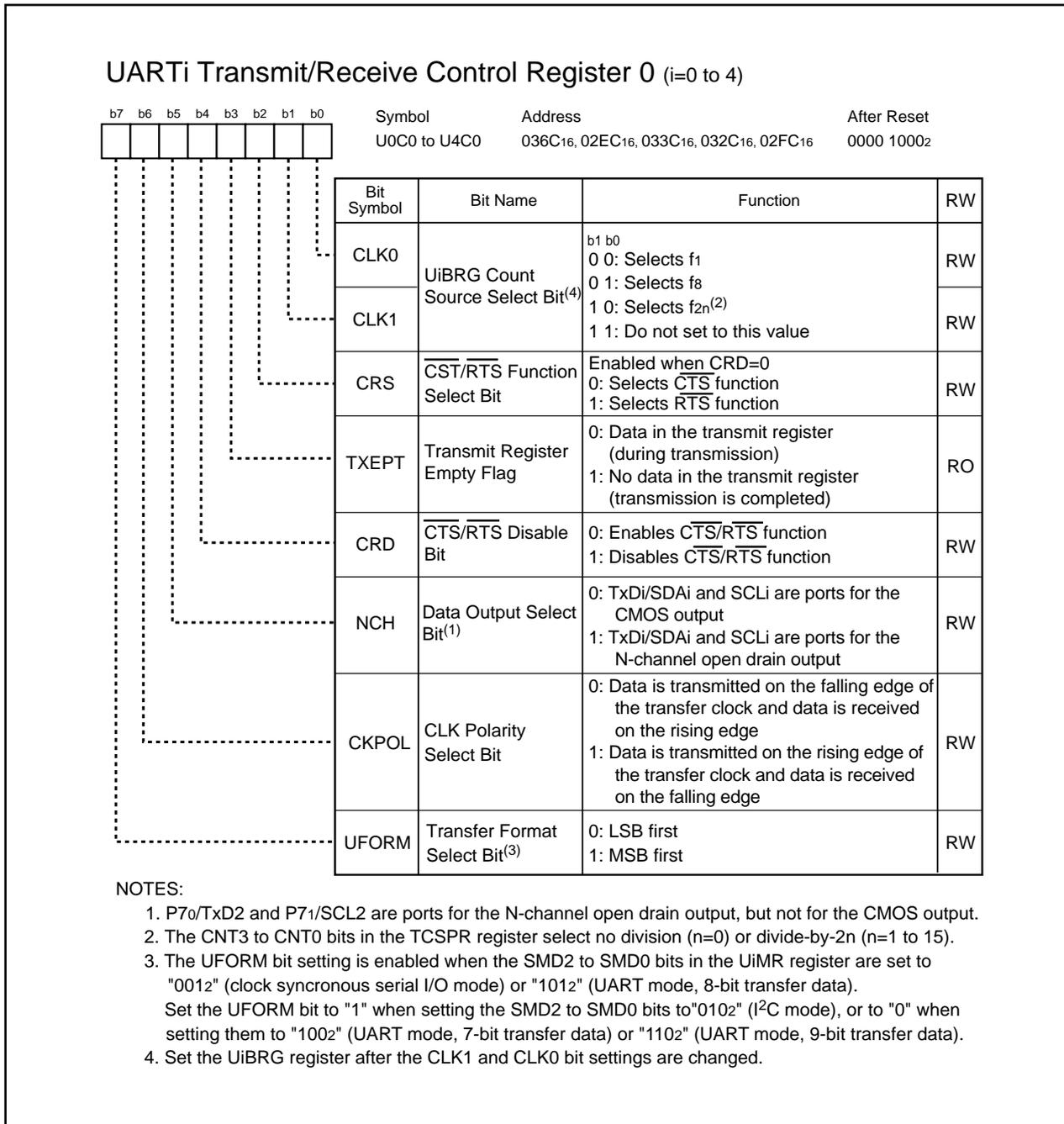
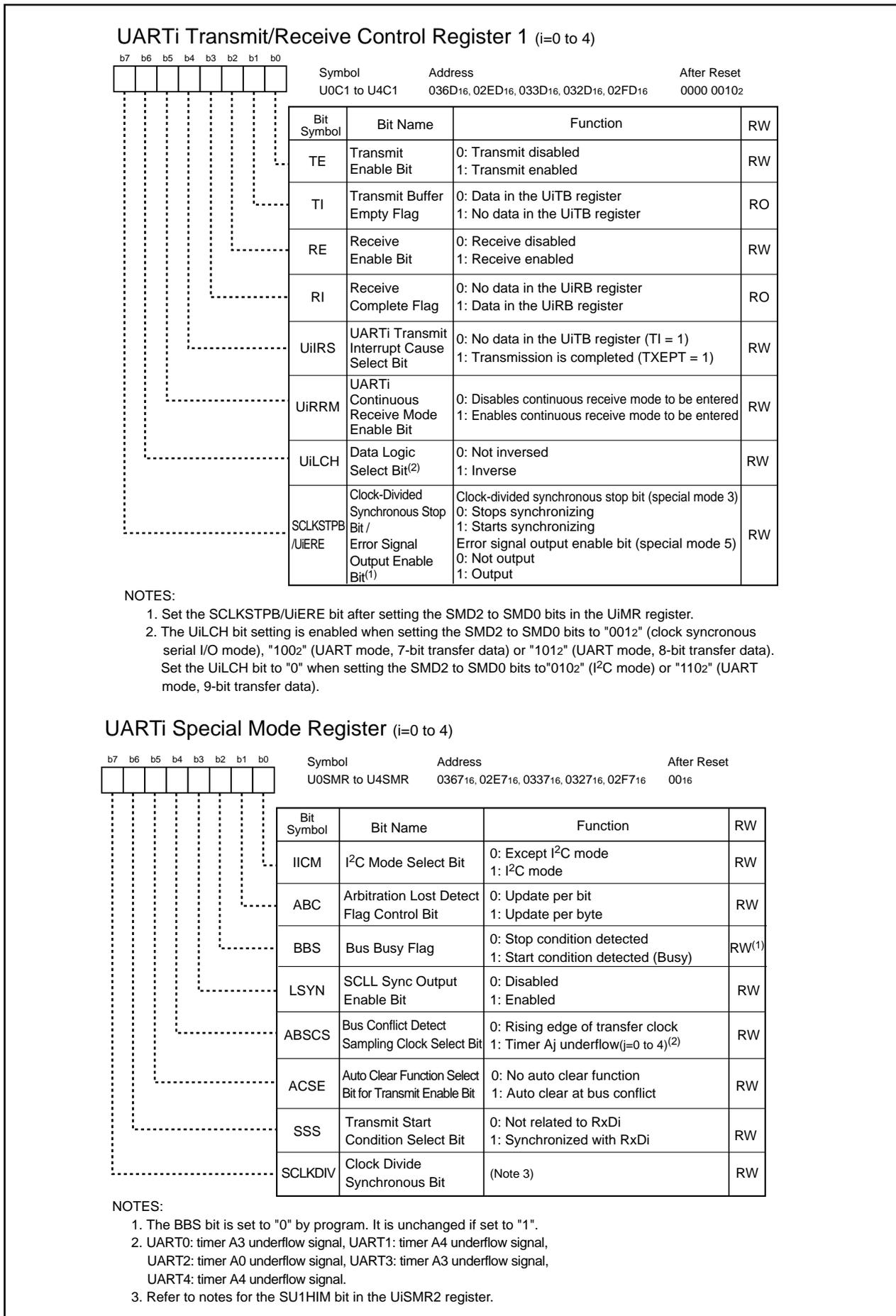
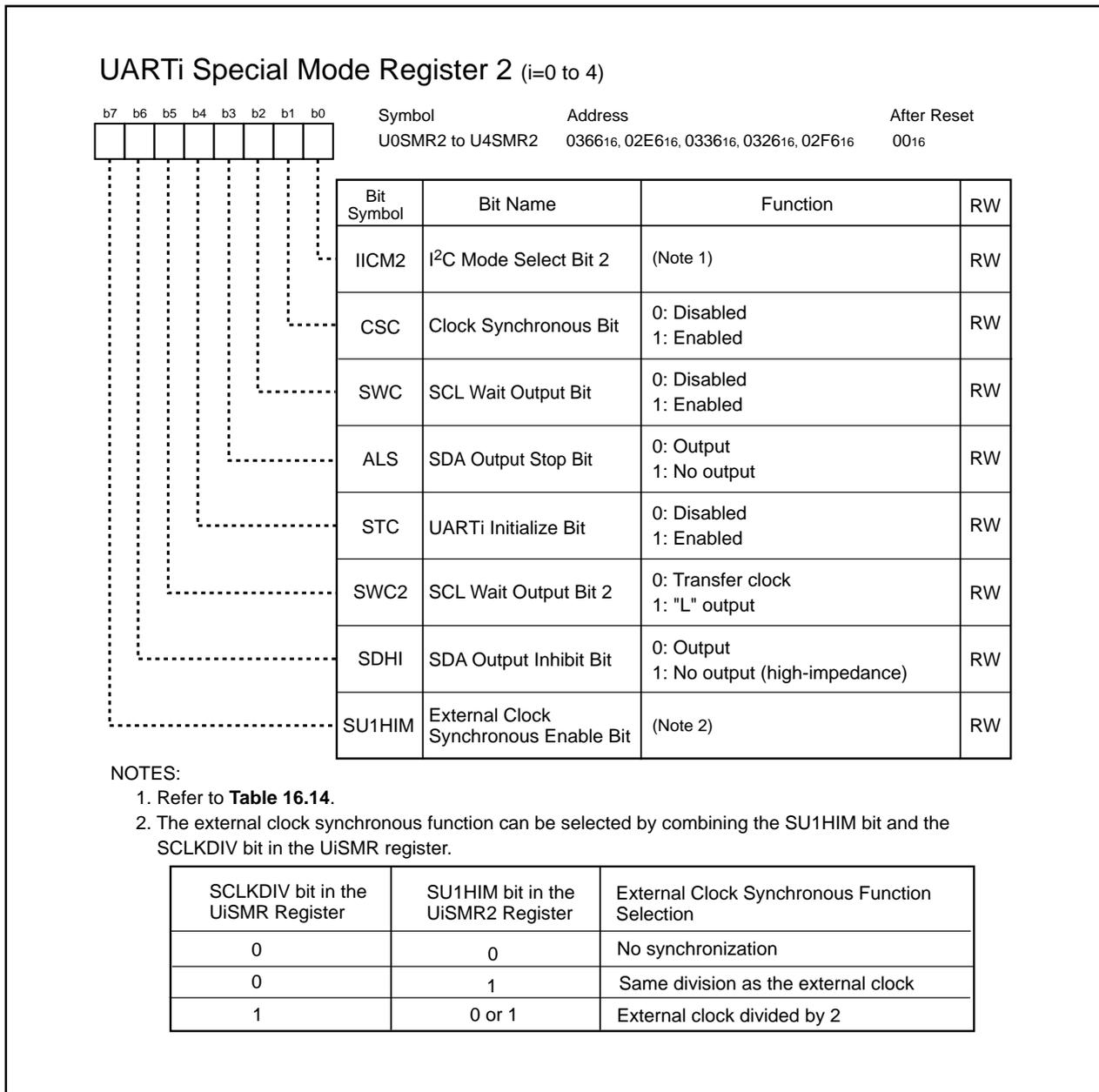


Figure 16.4 U0C0 to U4C0 Registers

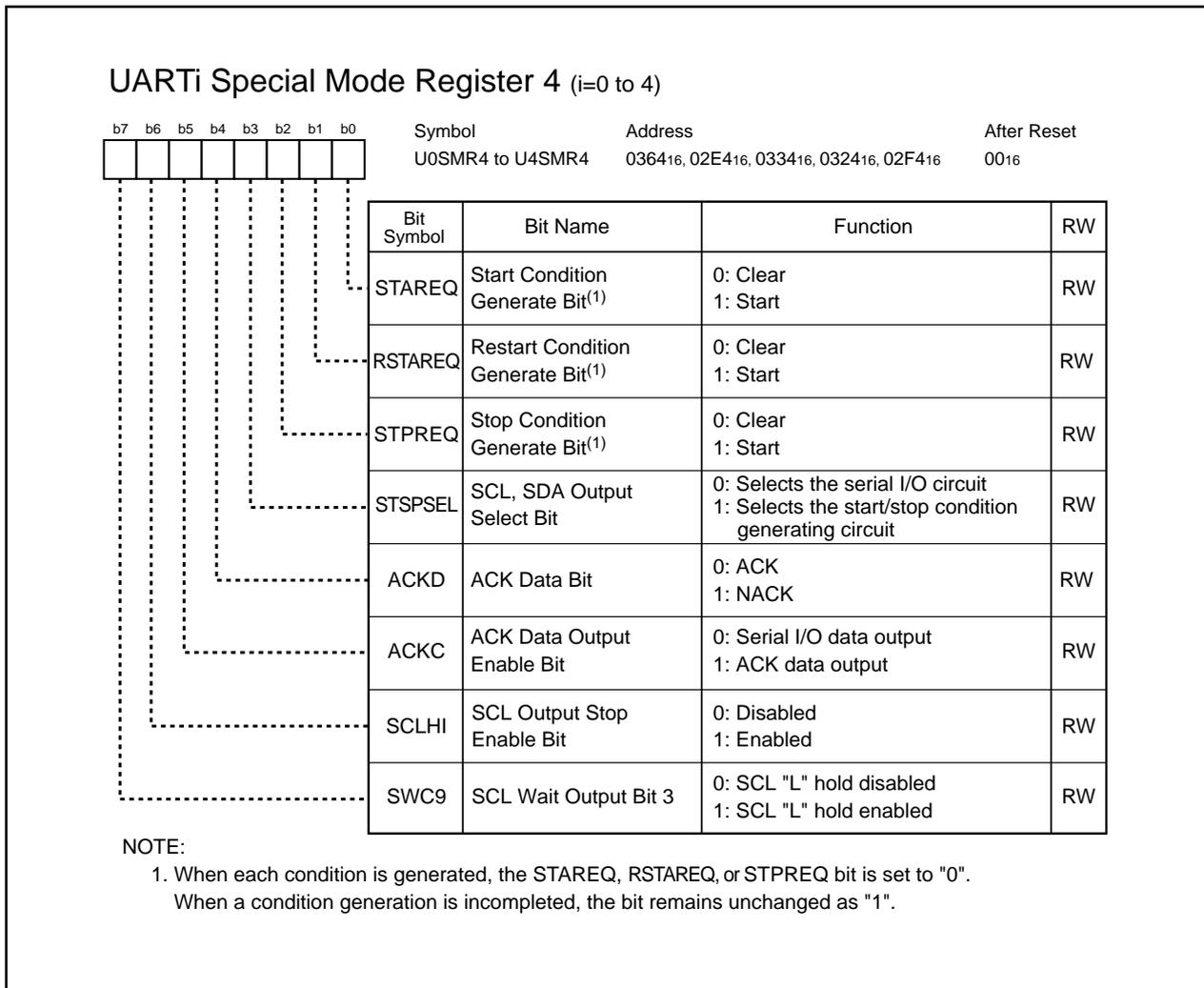


**Figure 16.5 U0C1 to U4C1 Registers and U0SMR to U4SMR Registers**

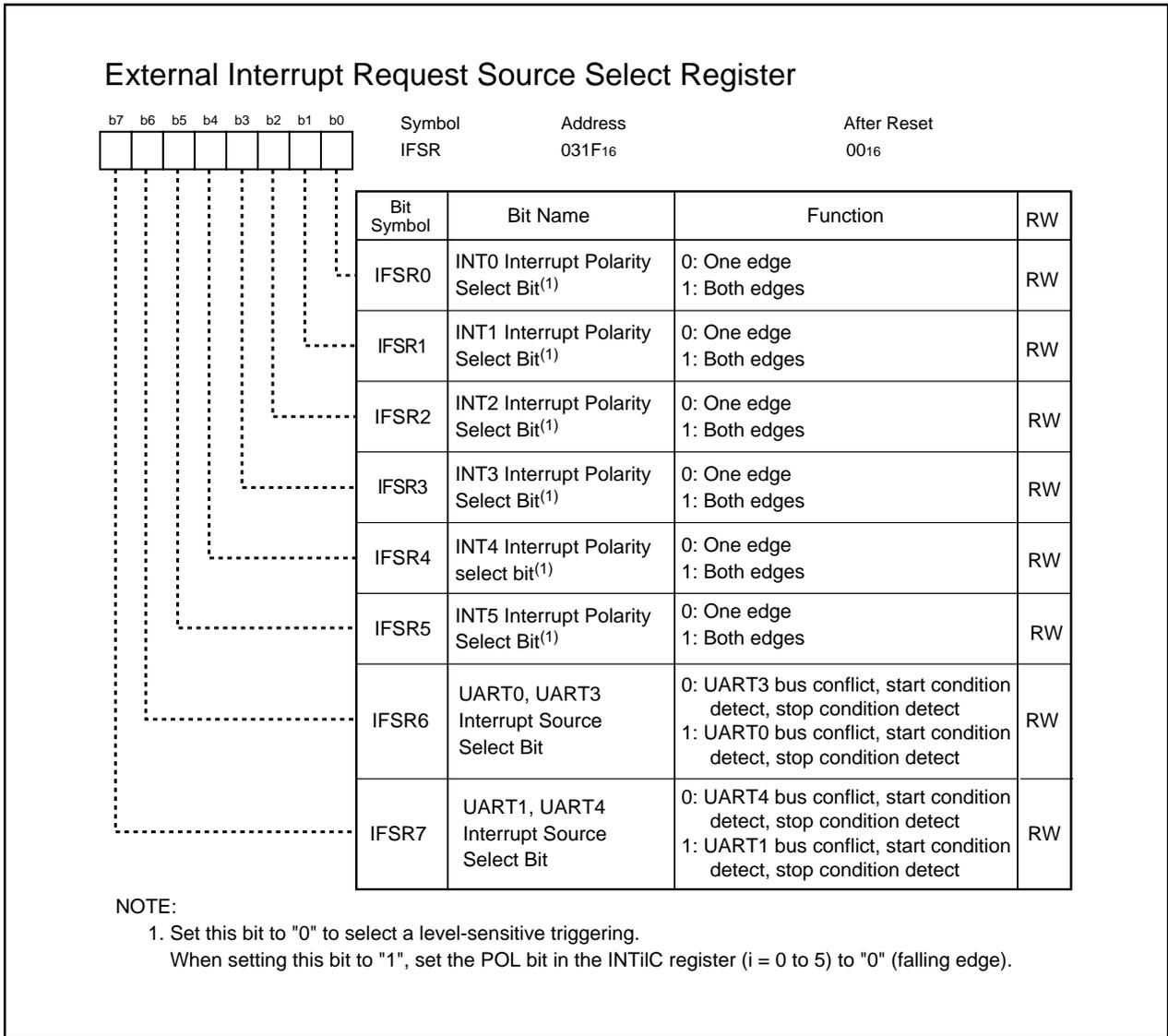


**Figure 16.6 U0SMR2 to U4SMR2 Registers**





**Figure 16.8 U0SMR4 to U4SMR4 Registers**



**Figure 16.9 IFSR Register**

## 16.1 Clock Synchronous Serial I/O Mode

In clock synchronous serial I/O mode, data is transmitted and received with the transfer clock. Table 16.1 lists specifications of clock synchronous serial I/O mode. Table 16.2 lists register settings. Tables 16.3 to 16.5 list pin settings. When UART<sub>i</sub> (i=0 to 4) operating mode is selected, the TxDi pin outputs a high-level ("H") signal before transfer starts (the TxDi pin is in a high-impedance state when the N-channel open drain output is selected). Figure 16.10 shows transmit and receive timings in clock synchronous serial I/O mode.

**Table 16.1 Clock Synchronous Serial I/O Mode Specifications**

Item	Specification
Transfer Data Format	Transfer data : 8 bits long
Transfer Clock	<ul style="list-style-type: none"> <li>The CKDIR bit in the UiMR register (i=0 to 4) is set to "0" (internal clock selected):  <math display="block">\frac{f_j}{2^{(m+1)}} \quad f_j=f_1, f_8, f_{2n^{(1)}} \quad m:\text{setting value of the UiBRG register, } 00_{16} \text{ to } FF_{16}</math> </li> <li>The CKDIR bit is set to "1" (external clock selected) : an input from the CLK<sub>i</sub> pin</li> </ul>
Transmit/Receive Control	Selected from the CTS function, RTS function or CTS/RTS function disabled
Transmit Start Condition	To start transmitting, the following requirements must be met <sup>(2)</sup> : <ul style="list-style-type: none"> <li>- Set the TE bit in the UiC1 register to "1" (transmit enabled)</li> <li>- Set the TI bit in the UiC1 register to "0" (data in the UiTB register)</li> <li>- Apply a low-level ("L") signal to the CTS<sub>i</sub> pin when the CTS function is selected</li> </ul>
Receive Start Condition	To start receiving, the following requirements must be met <sup>(2)</sup> : <ul style="list-style-type: none"> <li>- Set the RE bit in the UiC1 register to "1" (receive enabled)</li> <li>- Set the TE bit to "1" (transmit enabled)</li> <li>- Set the TI bit to "0" (data in the UiTB register)</li> </ul>
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>• While transmitting, the following conditions can be selected: <ul style="list-style-type: none"> <li>- The UiIRS bit in the UiC1 register is set to "0" (no data in the transmit buffer): when data is transferred from the UiTB register to the UART<sub>i</sub> transmit register (transfer started)</li> <li>- The UiIRS bit is set to "1" (transmission completed): when a data transfer from the UART<sub>i</sub> transmit register is completed</li> </ul> </li> <li>• While receiving When data is transferred from the UART<sub>i</sub> receive register to the UiRB register (reception completed)</li> </ul>
Error Detection	Overrun error <sup>(3)</sup> This error occurs when the seventh bit of the next received data is read before reading the UiRB register
Selectable Function	<ul style="list-style-type: none"> <li>• CLK polarity Selectable from the rising edge or falling edge of the transfer clock at transferred data output or input timing</li> <li>• LSB first or MSB first Selectable from data transmission or reception in either bit 0 or in bit 7</li> <li>• Continuous receive mode Data can be received simultaneously by reading the UiRB register</li> <li>• Serial data logic inverse This function inverses transmitted/received data logically</li> </ul>

**NOTES:**

1. The CNT3 to CNT0 bits in the TCSPR register select no division (n=0) or divide-by-2<sup>n</sup> (n=1 to 15).
2. To start transmission/reception when selecting the external clock, these conditions must be met after the CKPOL bit in the UiC0 register is set to "0" (data is transmitted on the falling edge of the transfer clock and data is received on the rising edge) and the CLK<sub>i</sub> pin is held "H", or when the CKPOL bit is set to "1" (data is transmitted on the rising edge of the transfer clock and data is received on the falling edge) and the CLK<sub>i</sub> pin is held "L".
3. If an overrun error occurs, the UiRB register is indeterminate. The IR bit setting in the SiRIC register does not change to "1" (interrupt requested).

**Table 16.2 Register Settings in Clock Synchronous Serial I/O Mode**

Register	Bit	Function
UiTB	7 to 0	Set transmit data
UiRB	7 to 0	Received data can be read
	OER	Overrun error flag
UiBRG	7 to 0	Set bit rate
UiMR	SMD2 to SMD0	Set to "0012"
	CKDIR	Select the internal clock or external clock
	IOPOL	Set to "0"
UiC0	CLK1, CLK0	Select count source for the UiBRG register
	CRS	Select $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ when using either
	TXEPT	Transmit register empty flag
	CRD	Enables or disables the CTS or RTS function
	NCH	Select output format of the TxDi pin
	CKPOL	Select transmit clock polarity
	UFORM	Select either LSB first or MSB first
UiC1	TE	Set to "1" to enable data transmission and reception
	TI	Transmit buffer empty flag
	RE	Set to "1" to enable data reception
	RI	Reception complete flag
	UiIRS	Select what causes the UARTi transmit interrupt to be generated
	UiRRM	Set to "1" when using continuous receive mode
	UiLCH	Set to "1" when using data logic inverse
	SCLKSTPB	Set to "0"
UiSMR	7 to 0	Set to "0016"
UiSMR2	7 to 0	Set to "0016"
UiSMR3	2 to 0	Set to "0002"
	NODC	Select clock output format
	7 to 4	Set to "00002"
UiSMR4	7 to 0	Set to "0016"

i=0 to 4

**Table 16.3 Pin Settings in Clock Synchronous Serial I/O Mode (1)**

Port	Function	Setting		
		PS0 Register	PSL0 Register	PD6 Register
P60	CTS0 input	PS0_0=0	-	PD6_0=0
	RTS0 output	PS0_0=1	PSL0_0=0	-
P61	CLK0 input	PS0_1=0	-	PD6_1=0
	CLK0 output	PS0_1=1	-	-
P62	RxD0 input	PS0_2=0	-	PD6_2=0
P63	TxD0 output	PS0_3=1	-	-
P64	CTS1 input	PS0_4=0	-	PD6_4=0
	RTS1 output	PS0_4=1	PSL0_4=0	-
P65	CLK1 input	PS0_5=0	-	PD6_5=0
	CLK1 output	PS0_5=1	-	-
P66	RxD1 input	PS0_6=0	-	PD6_6=0
P67	TxD1 output	PS0_7=1	-	-

**Table 16.4 Pin Settings (2)**

Port	Function	Setting			
		PS1 Register	PSL1 Register	PSC Register	PD7 Register
P70 <sup>(1)</sup>	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	-
P71 <sup>(1)</sup>	RxD2 input	PS1_1=0	-	-	PD7_1=0
P72	CLK2 input	PS1_2=0	-	-	PD7_2=0
	CLK2 output	PS1_2=1	PSL1_2=0	PSC_2=0	-
P73	CTS2 input	PS1_3=0	-	-	PD7_3=0
	RTS2 output	PS1_3=1	PSL1_3=0	PSC_3=0	-

NOTE:

1. P70 and P71 are ports for the N-channel open drain output.

**Table 16.5 Pin Settings (3)**

Port	Function	Setting			
		PS3 Register <sup>(1)</sup>	PSL3 Register	PSC3 Register	PD9 Register <sup>(1)</sup>
P90	CLK3 input	PS3_0=0	-	-	PD9_0=0
	CLK3 output	PS3_0=1	-	-	-
P91	RxD3 input	PS3_1=0	-	-	PD9_1=0
P92	TxD3 output	PS3_2=1	PSL3_2=0	-	-
P93	CTS3 input	PS3_3=0	PSL3_3=0	-	PD9_3=0
	RTS3 output	PS3_3=1	-	-	-
P94	CTS4 input	PS3_4=0	PSL3_4=0	-	PD9_4=0
	RTS4 output	PS3_4=1	-	-	-
P95	CLK4 input	PS3_5=0	PSL3_5=0	-	PD9_5=0
	CLK4 output	PS3_5=1	-	-	-
P96	TxD4 output	PS3_6=1	-	PSC3_6=0	-
P97	RxD4 input	PS3_7=0	-	-	PD9_7=0

NOTE:

1. Set the PD9 and PS3 registers immediately after the PRC2 bit in the PRCR register is set to "1" (write enabled). Do not generate an interrupt or a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to set the PD9 and PS3 registers.

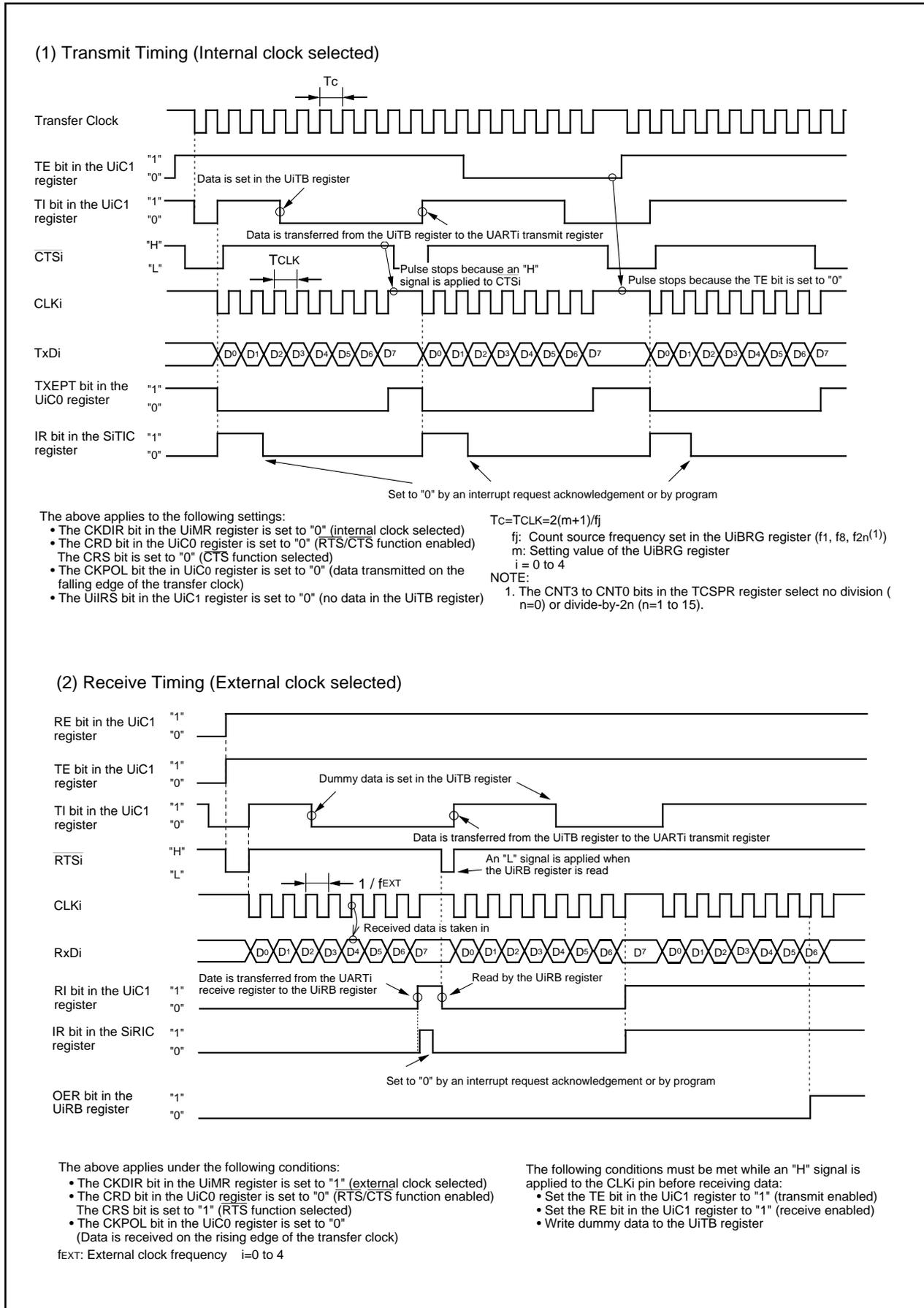


Figure 16.10 Transmit and Receive Operation

### 16.1.1 Selecting CLK Polarity Selecting

As shown in Figure 16.11, the CKPOL bit in the UiC0 register (i=0 to 4) determines the polarity of the transfer clock.

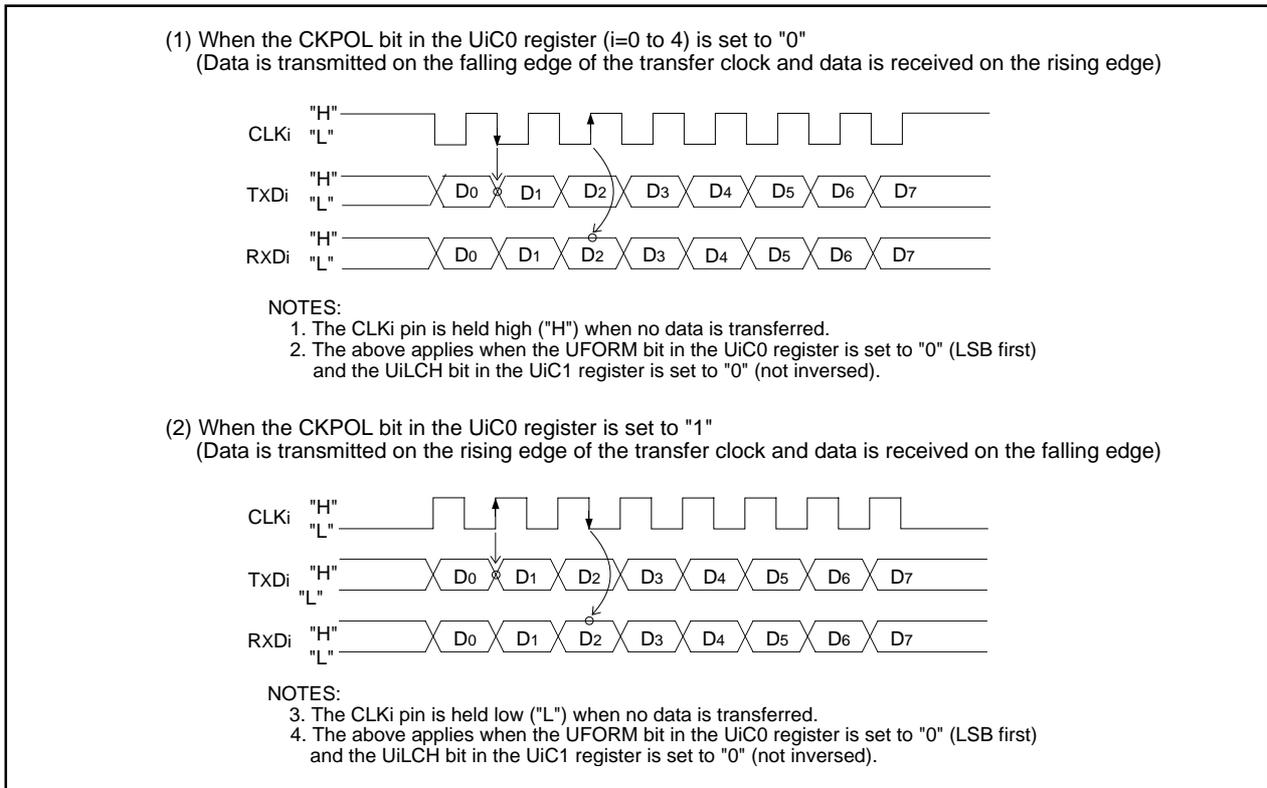


Figure 16.11 Transfer Clock Polarity

### 16.1.2 Selecting LSB First or MSB First

As shown in Figure 16.12, the UFORM bit in the UiC0 register (i=0 to 4) determines a data transfer format.

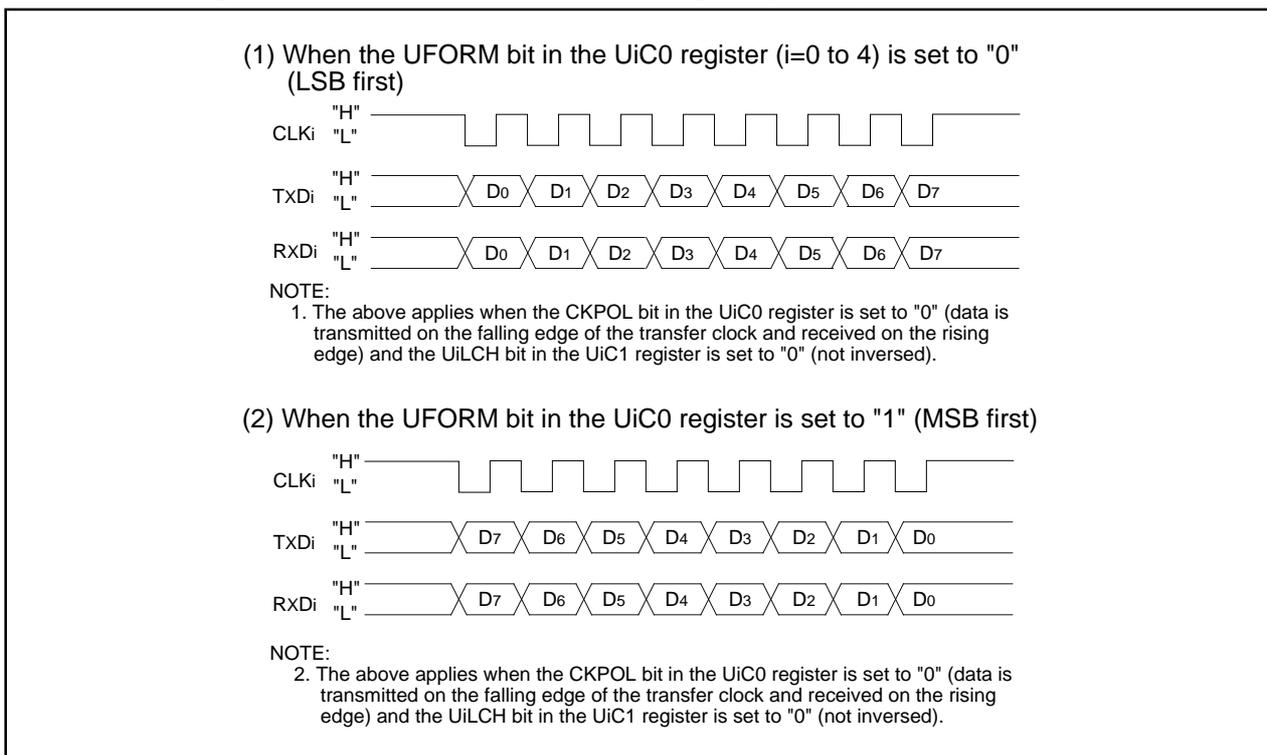


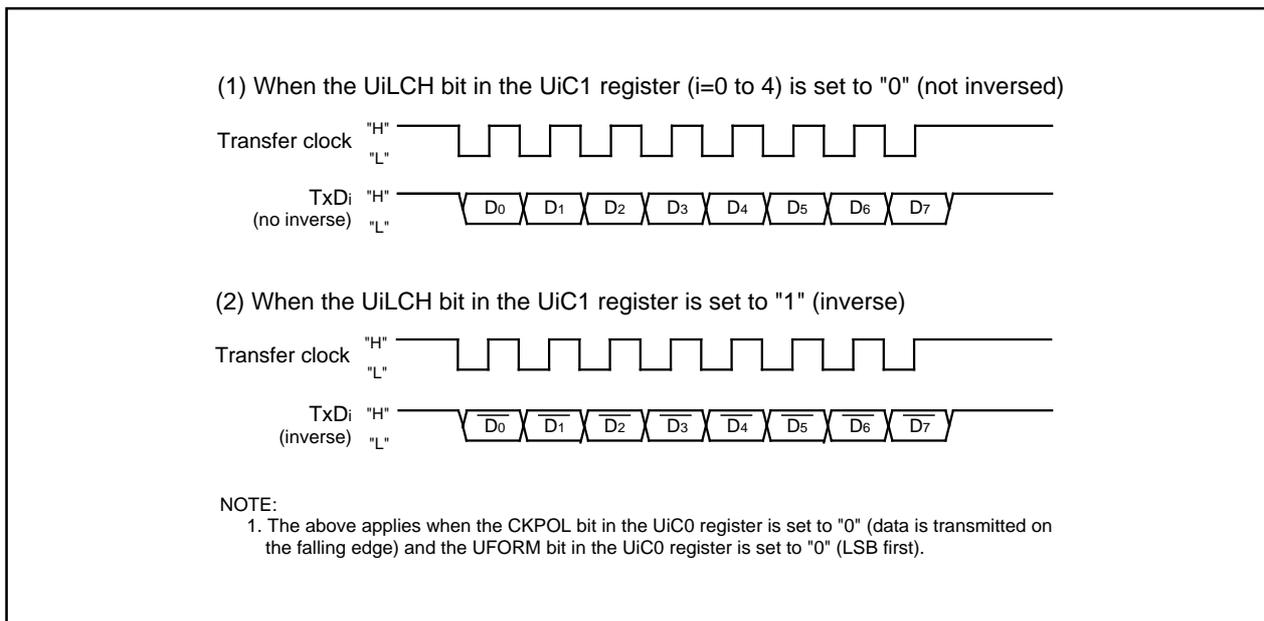
Figure 16.12 Transfer Format

### 16.1.3 Continuous Receive Mode

When the UiRRM bit in the UiC1 register (i=0 to 4) is set to "1" (continuous receive mode), the TI bit is set to "0" (data in the UiTB register) by reading the UiRB register. When the UiRRM bit is set to "1", do not set dummy data in the UiTB register by program.

### 16.1.4 Serial Data Logic Inverse

When the UiLCH bit (i=0 to 4) in the UiC1 register is set to "1" (inverse), data logic written in the UiTB register is inverted when transmitted. The inverted receive data logic can be read by reading the UiRB register. Figure 16.13 shows a switching example of the serial data logic.



**Figure 16.13 Serial Data Logic Inverse**

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

In UART mode, data is transmitted and received after setting a desired bit rate and data transfer format. Table 16.6 lists specifications of UART mode.

**Table 16.6 UART Mode Specifications**

Item	Specification
Transfer Data Format	<ul style="list-style-type: none"> <li>• Character bit (transfer data) : selected from 7 bits, 8 bits, or 9 bits long</li> <li>• Start bit: 1 bit long</li> <li>• Parity bit: selected from odd, even, or none</li> <li>• Stop bit: selected from 1 bit or 2 bits long</li> </ul>
Transfer Clock	<ul style="list-style-type: none"> <li>• The CKDIR bit in the UiMR register is set to "0" (internal clock selected):  <math>f_j/16(m+1)</math> <math>f_j = f_1, f_8, f_{2n}^{(1)}</math> <math>m</math>: setting value of the UiBRG register, 0016 to FF16</li> <li>• The CKDIR bit is set to "1" (external clock selected):  <math>f_{EXT}/16(m+1)</math> <math>f_{EXT}</math>: clock applied to the CLKi pin</li> </ul>
Transmit/Receive Control	Select from CTS function, RTS function or CTS/RTS function disabled
Transmit Start Condition	To start transmitting, the following requirements must be met: <ul style="list-style-type: none"> <li>- Set the TE bit in the UiC1 register to "1" (transmit enabled)</li> <li>- Set the TI bit in the UiC1 register to "0" (data in the UiTB register)</li> <li>- Apply a low-level ("L") signal to the CTSi pin when the CTS function is selected</li> </ul>
Receive Start Condition	To start receiving, the following requirements must be met: <ul style="list-style-type: none"> <li>- Set the RE bit in the UiC1 register to "1" (receive enabled)</li> <li>- The start bit is detected</li> </ul>
Interrupt Request Generation Timing	<p>While transmitting, the following condition can be selected:</p> <ul style="list-style-type: none"> <li>- The UiIRS bit in the UiC1 register is set to "0" (no data in the UiTB register): when data is transferred from the UiTB register to the UARTi transmit register (transfer started)</li> <li>- The UiIRS bit is set to "1" (transmission completed): when data transmission from the UARTi transfer register is completed</li> </ul> <p>While receiving</p> <ul style="list-style-type: none"> <li>- when data is transferred from the UARTi receive register to the UiRB register (reception completed)</li> </ul>
Error Detect	<ul style="list-style-type: none"> <li>• Overrun error<sup>(2)</sup> This error occurs when the bit before the last stop bit of the next received data is read prior to reading the UiRB register (the first stop bit when selecting 2 stop bits)</li> <li>• Framing error This error occurs when the number of stop bits set is not detected</li> <li>• Parity error When parity is enabled, this error occurs when the number of "1" in parity and character bits does not match the number of "1" set</li> <li>• Error sum flag This flag is set to "1" when any of an overrun, framing or parity errors occur</li> </ul>
Selectable Function	<ul style="list-style-type: none"> <li>• LSB first or MSB first Selectable from data transmission or reception in either bit 0 or in bit 7</li> <li>• Serial data logic inverse Logic values of data to be transmitted and received data are inversed. The start bit and stop bit are not inversed</li> <li>• TxD and RxD I/O polarity Inverse TxD pin output and RxD pin input are inversed. All I/O data levels are also inversed</li> </ul>

**NOTES:**

1. The CNT3 to CNT0 bits in the TCSPR register select no division (n=0) or divide-by-2n (n=1 to 15).
2. If an overrun error occurs, the UiRB register is indeterminate. The IR bit setting in the SiRIC register does not change to "1" (interrupt requested).

Table 16.7 lists register settings. Tables 16.8 to 16.10 list pin settings. When UART<sub>i</sub> (i=0 to 4) operating mode is selected, the TxDi pin outputs a high-level ("H") signal before transfer is started (the TxDi pin is in a high-impedance state when the N-channel open drain output is selected). Figure 16.14 shows an example of a transmit operation in UART mode. Figure 16.15 shows an example of a receive operation in UART mode.

**Table 16.7 Register Settings in UART Mode**

Register	Bit	Function
UiTB	8 to 0	Set transmit data <sup>(1)</sup>
UiRB	8 to 0	Received data can be read <sup>(1)</sup>
	OER, FER, PER, SUM	Error flags
UiBRG	7 to 0	Set bit rate
UiMR	SMD2 to SMD0	Set to "1002" when transfer data is 7 bits long Set to "1012" when transfer data is 8 bits long Set to "1102" when transfer data is 9 bits long
	CKDIR	Select the internal clock or external clock
	STPS	Select stop bit length
	PRY, PRYE	Select parity enable or disable, odd or even
	IOPOL	Select TxD and RxD I/O polarity
UiC0	CLK1, CLK0	Select count source for the UiBRG register
	CRS	Select either $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ when using either
	TXEPT	Transfer register empty flag
	CRD	Select the $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ function enabled or disabled
	NCH	Select output format of the TxDi pin
	CKPOL	Set to "0"
	UFORM	Select the LSB first or MSB first when a transfer data is 8 bits long Set to "0" when transfer data is 7 bits or 9 bits long
UiC1	TE	Set to "1" to enable data transmission
	TI	Transfer buffer empty flag
	RE	Set to "1" to enable data reception
	RI	Reception complete flag
	UiIRS	Select what causes the UART <sub>i</sub> transmit interrupt to be generated
	UiRRM	Set to "0"
	UiLCH	Select whether data logic is inversed or not inversed when a transfer data is 7 bits or 8 bits long. Set to "0" when transfer data is 9 bits long
	UiERE	Set to either "0" or "1"
UiSMR	7 to 0	Set to "0016"
UiSMR2	7 to 0	Set to "0016"
UiSMR3	7 to 0	Set to "0016"
UiSMR4	7 to 0	Set to "0016"

**NOTE:**

1. Use bits 0 to 6 when transfer data is 7 bits long, bits 0 to 7 when 8 bits long, bits 0 to 8 when 9 bits long.

**Table 16.8 Pin Settings in UART Mode (1)**

Port	Function	Setting		
		PS0 Register	PSL0 Register	PD6 Register
P60	CTS0 input	PS0_0=0	–	PD6_0=0
	RTS0 output	PS0_0=1	PSL0_0=0	–
P61	CLK0 input	PS0_1=0	–	PD6_1=0
P62	RxD0 input	PS0_2=0	–	PD6_2=0
P63	TxD0 output	PS0_3=1	–	–
P64	CTS1 input	PS0_4=0	–	PD6_4=0
	RTS1 output	PS0_4=1	PSL0_4=0	–
P65	CLK1 input	PS0_5=0	–	PD6_5=0
P66	RxD1 input	PS0_6=0	–	PD6_6=0
P67	TxD1 output	PS0_7=1	–	–

**Table 16.9 Pin Settings (2)**

Port	Function	Setting			
		PS1 Register	PSL1 Register	PSC Register	PD7 Register
P70 <sup>(1)</sup>	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	–
P71 <sup>(1)</sup>	RxD2 input	PS1_1=0	–	–	PD7_1=0
P72	CLK2 input	PS1_2=0	–	–	PD7_2=0
P73	CTS2 input	PS1_3=0	–	–	PD7_3=0
	RTS2 output	PS1_3=1	PSL1_3=0	PSC_3=0	–

NOTE:

1. P70 and P71 are ports for the N-channel open drain output.

**Table 16.10 Pin Settings (3)**

Port	Function	Setting			
		PS3 Register <sup>(1)</sup>	PSL3 Register	PSC3 Register	PD9 Register <sup>(1)</sup>
P90	CLK3 input	PS3_0=0	–	–	PD9_0=0
P91	RxD3 input	PS3_1=0	–	–	PD9_1=0
P92	TxD3 output	PS3_2=1	PSL3_2=0	–	–
P93	CTS3 input	PS3_3=0	PSL3_3=0	–	PD9_3=0
	RTS3 output	PS3_3=1	–	–	–
P94	CTS4 input	PS3_4=0	PSL3_4=0	–	PD9_4=0
	RTS4 output	PS3_4=1	–	–	–
P95	CLK4 input	PS3_5=0	PSL3_5=0	–	PD9_5=0
P96	TxD4 output	PS3_6=1	–	PSC3_6=0	–
P97	RxD4 input	PS3_7=0	–	–	PD9_7=0

NOTE:

1. Set the PD9 and PS3 registers set immediately after the PRC2 bit in the PRCR register is set to "1" (write enabled). Do not generate an interrupt or a DMA transfer between the instruction to set to the PRC2 bit to "1" and the instruction to set the PD9 and PS3 registers.

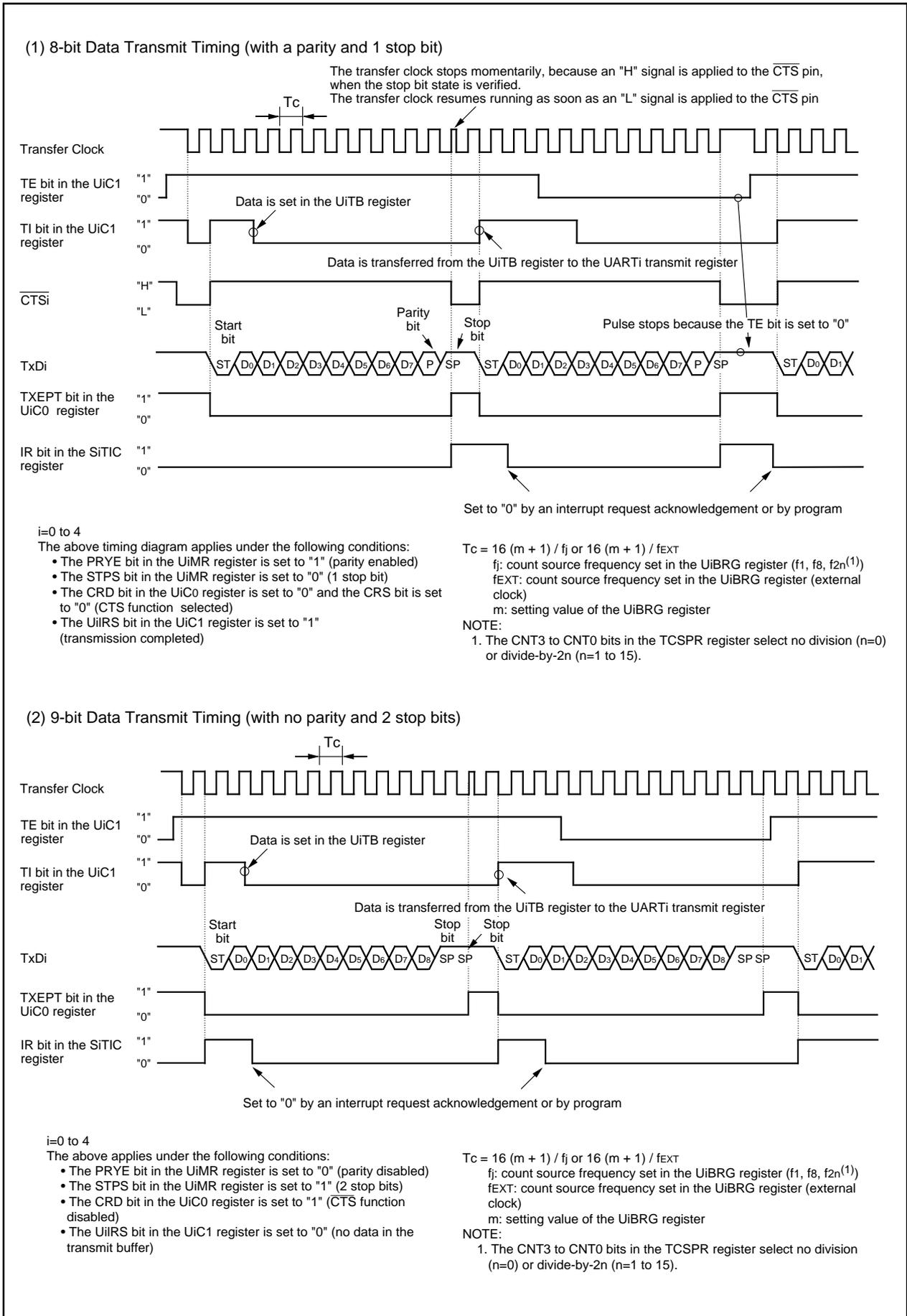


Figure 16.14 Transmit Operation

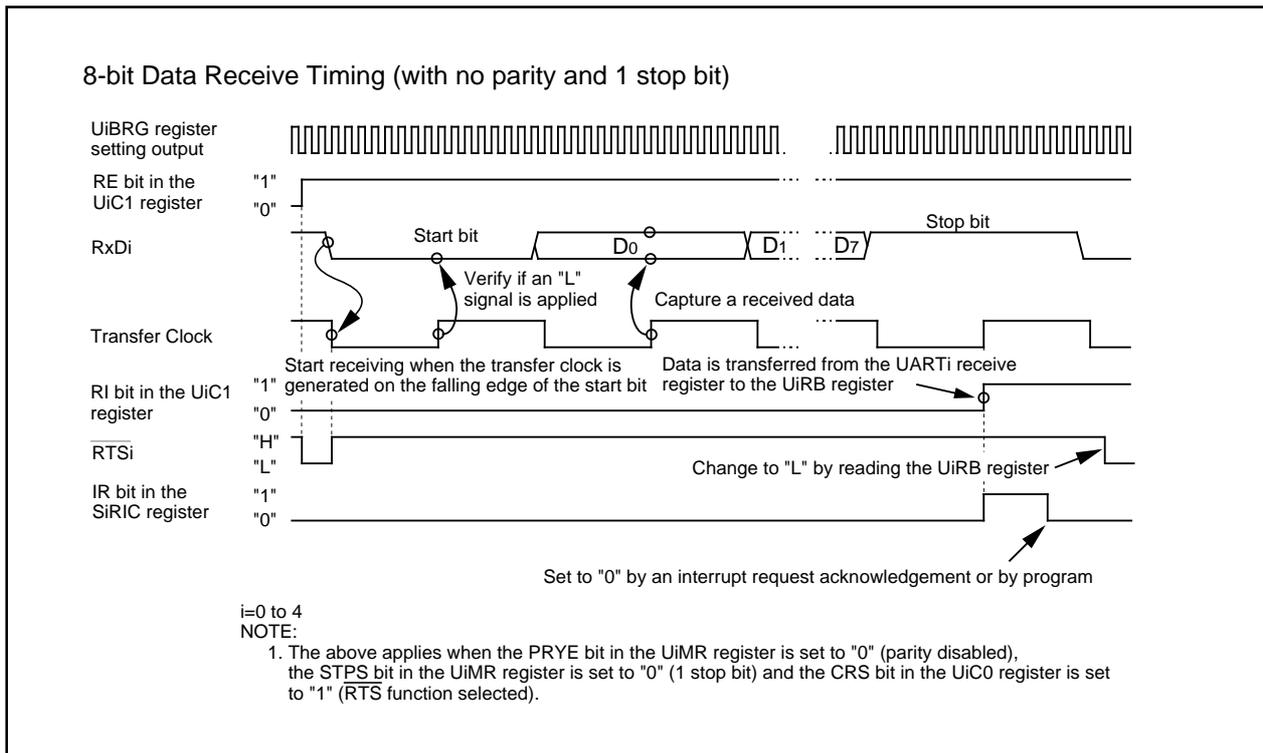


Figure 16.15 Receive Operation

### 16.2.1 Bit Rate

In UART mode, bit rate is clock frequency which is divided by a setting value of the UIBRG ( $i=0$  to 4) register and again divided by 16. Table 16.11 lists an example of bit rate setting.

Table 16.11 Bit Rate

Bit Rate (bps)	Count Source of UIBRG	Peripheral Function Clock: 16MHz		Peripheral Function Clock: 24MHz		Peripheral Function Clock: 32MHz	
		Setting Value of UIBRG: $n$	Actual Bit Rate (bps)	Setting Value of UIBRG: $n$	Actual Bit Rate (bps)	Setting Value of UIBRG: $n$	Actual Bit Rate (bps)
1200	f8	103 (67h)	1202	155 (96h)	1202	207 (CFh)	1202
2400	f8	51 (33h)	2404	77 (46h)	2404	103 (67h)	2404
4800	f8	25 (19h)	4808	38 (26h)	4808	51 (33h)	4808
9600	f1	103 (67h)	9615	155 (96h)	9615	207 (CFh)	9615
14400	f1	68 (44h)	14493	103 (67h)	14423	138 (8Ah)	14388
19200	f1	51 (33h)	19231	77 (46h)	19231	103 (67h)	19231
28800	f1	34 (22h)	28571	51 (33h)	28846	68 (44h)	28986
31250	f1	31 (1Fh)	31250	47 (2Fh)	31250	63 (3Fh)	31250
38400	f1	25 (19h)	38462	38 (26h)	38462	51 (33h)	38462
51200	f1	19 (13h)	50000	28 (1Ch)	51724	38 (26h)	51282

### 16.2.2 Selecting LSB First or MSB First

As shown in Figure 16.16, the UFORM bit in the UiC0 register (i=0 to 4) determines data transfer format. This function is available for 8-bit transfer data.

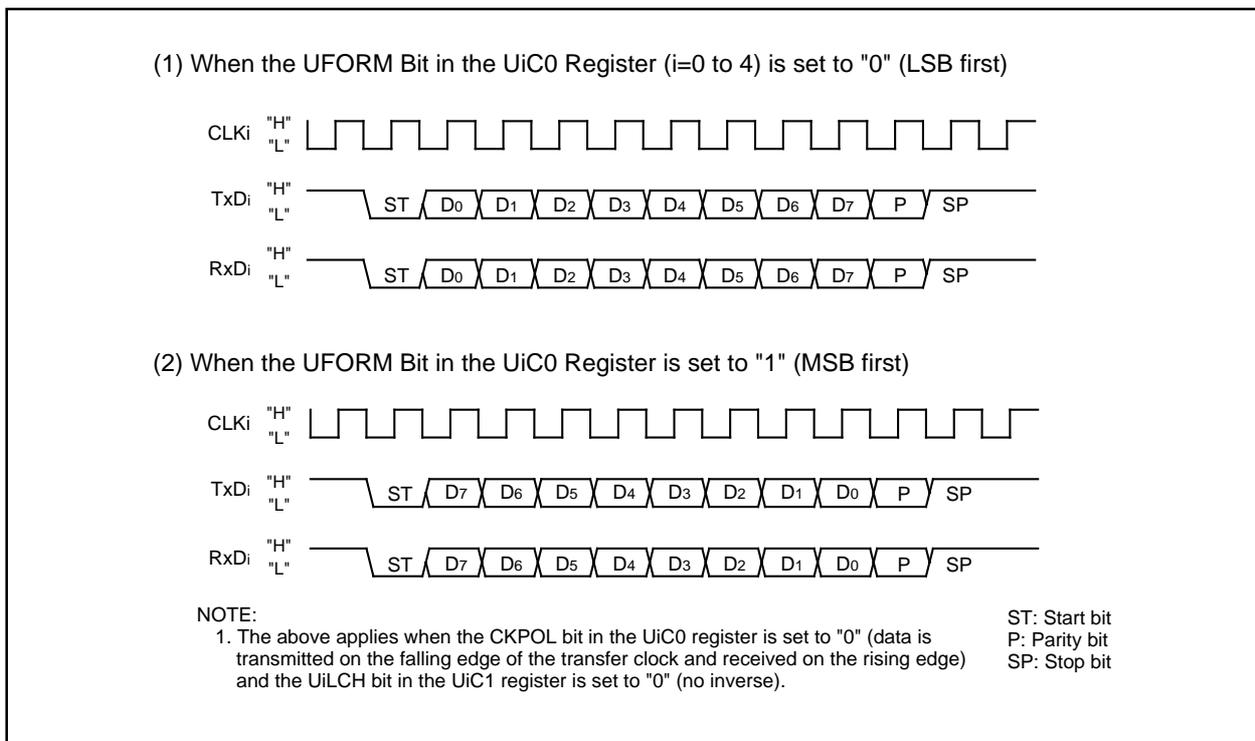


Figure 16.16 Transfer Format

### 16.2.3 Serial Data Logic Inverse

When the UiLCH bit (i=0 to 4) in the UiC1 register is set to "1" (inverse), data logic written in the UiTB register is inverted when transmitted. The inverted receive data logic can be read by reading the UiRB register. Figure 16.17 shows a switching example of the serial data logic.

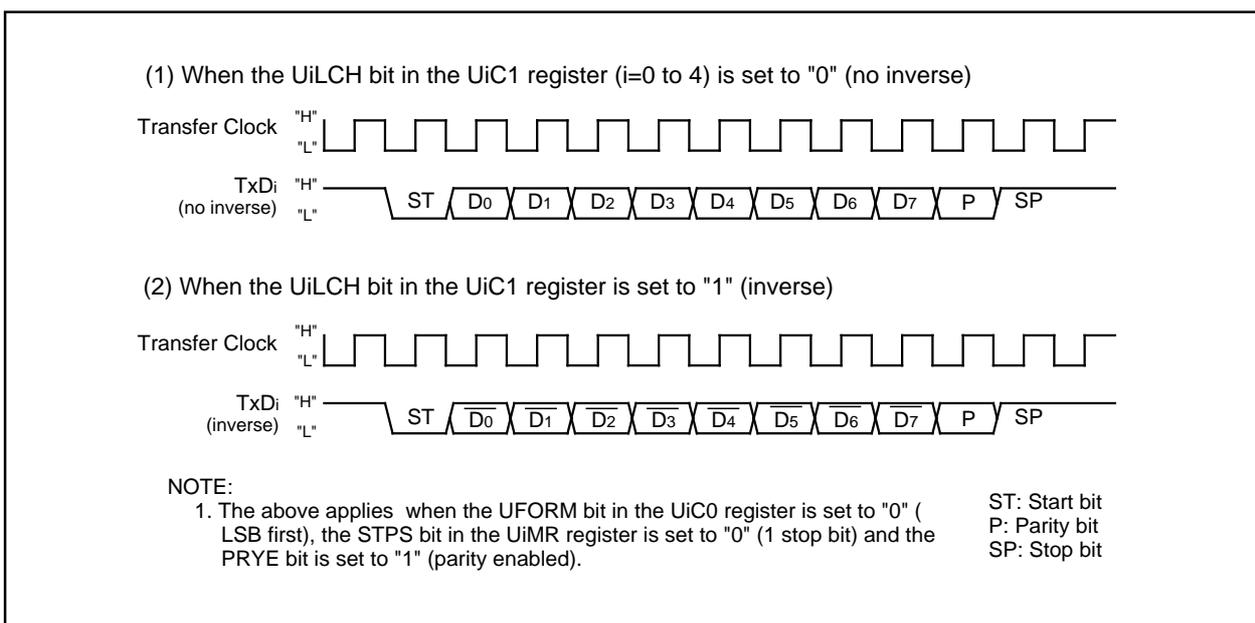


Figure 16.17 Serial Data Logic Inverse

### 16.2.4 TxD and RxD I/O Polarity Inverse

TxD pin output and RxD pin input are inverted. All I/O data level, including the start bit, stop bit and parity bit, are inverted. Figure 16.18 shows TxD and RxD I/O polarity inverse.

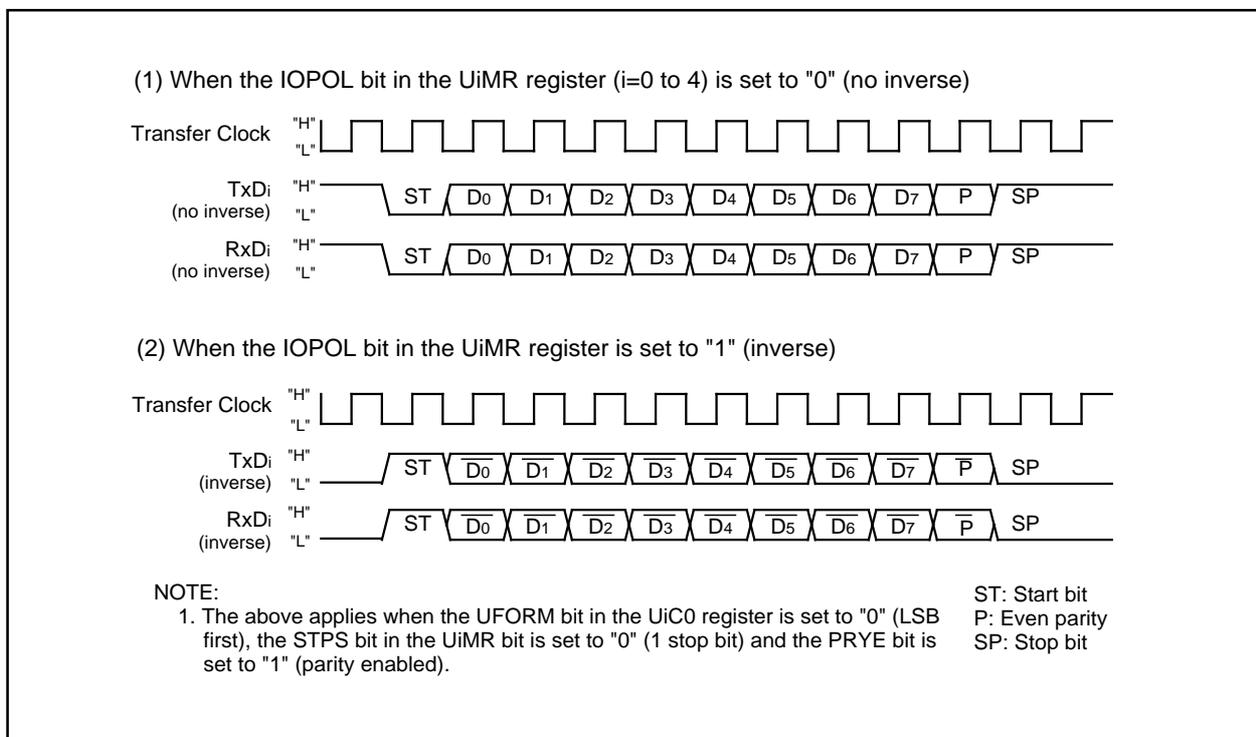


Figure 16.18 TxD and RxD I/O Polarity Inverse

### 16.3 Special Mode 1 (I<sup>2</sup>C Mode)

I<sup>2</sup>C mode is a mode to communicate with external devices with a simplified I<sup>2</sup>C. Table 16.12 lists specifications of I<sup>2</sup>C mode. Table 16.13 lists register settings, Table 16.14 lists each function. Figure 16.19 shows a block diagram of I<sup>2</sup>C mode. Figure 16.20 shows timings for transfer to the UiRB register (i=0 to 4) and interrupts. Tables 16.15 to 16.17 list pin settings.

As shown in Table 16.12, I<sup>2</sup>C mode is entered when the SMD2 to SMD0 bits in the UiMR register is set to "0102" and the IICM bit in the UiSMR register is set to "1". Output signal from the SDAi pin changes after the SCLi pin level becomes low ("L") and stabilizes due to a SDAi transmit output via the delay circuit.

**Table 16.12 I<sup>2</sup>C Mode Specifications**

Item	Specifications
Interrupt	Start condition detect, stop condition detect, no acknowledgment detect, acknowledgment detect
Selectable Function	<ul style="list-style-type: none"> <li>• Arbitration lost            Selectable from update timing of the ABT bit in the UiRB register.            Refer to <b>16.3.3 Arbitration</b></li> <li>• SDAi digital delay            Selectable from no digital delay or 2 to 8 cycle delay of the count source of the UiBRG register. Refer to <b>16.3.5 SDA Output</b></li> <li>• Clock phase setting            Selectable from clock delay or no clock delay. Refer to <b>16.3.4 Transfer Clock</b></li> </ul>

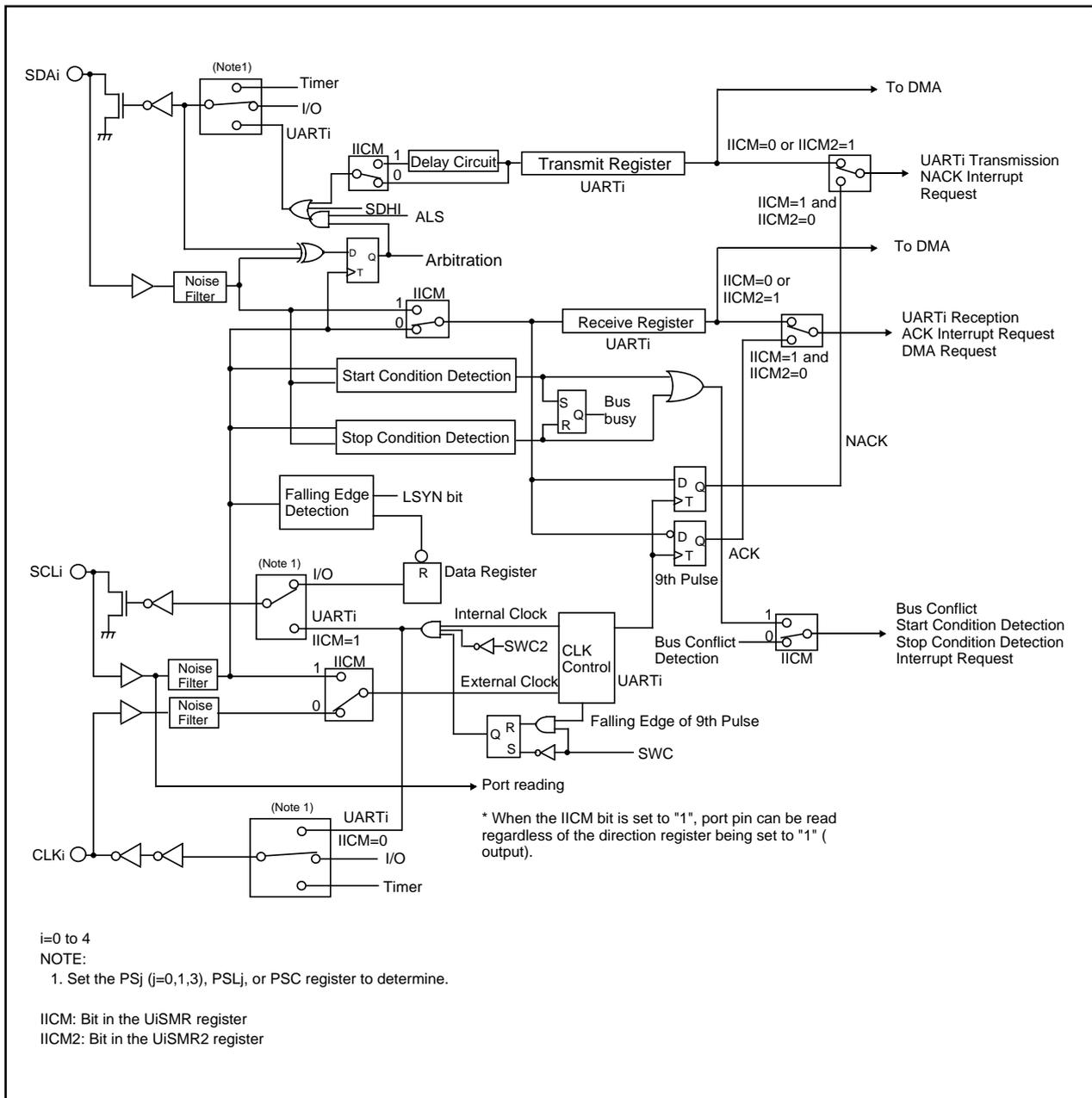


Figure 16.19 I²C Mode Block Diagram

**Table 16.13 Register Settings in I<sup>2</sup>C Mode**

Register	Bit	Function	
		Master	Slave
UiTB	7 to 0	Set transmit data	
UiRB	7 to 0	Received data can be read	
	8	ACK or NACK bit can be read	
	ABT	Arbitration lost detect flag	Disabled
	OER	Overrun error flag	
UiBRG	7 to 0	Set bit rate	Disabled
UiMR	SMD2 to SMD0	Set to "0102"	
	CKDIR	Set to "0"	Set to "1"
	IOPOL	Set to "0"	
UiC0	CLK1, CLK0	Select count source of the UiBRG register	Disabled
	CRS	Disabled because the CRD bit is set to "1"	
	TXEPT	Transfer register empty flag	
	CRD, NCH	Set to "1"	
	CKPOL	Set to "0"	
	UFORM	Set to "1"	
UiC1	TE	Set to "1" to enable data transmission	
	TI	Transfer buffer empty flag	
	RE	Set to "1" to enable data reception	
	RI	Reception complete flag	
	UiRRM, UiLCH, UIERE	Set to "0"	
UiSMR	IICM	Set to "1"	
	ABC	Select an arbitration lost detect timing	Disabled
	BBS	Bus busy flag	
	7 to 3	Set to "000002"	
UiSMR2	IICM2	See Table 16.14	
	CSC	Set to "1" to enable clock synchronization	Set to "0"
	SWC	Set to "1" to fix an "L" signal output from SCLi on the falling edge of the ninth bit of the transfer clock	
	ALS	Set to "1" to terminate SDAi output when detecting the arbitration lost	Not used. Set to "0"
	STC	Not used. Set to "0"	Set to "1" to reset UARTi by detecting the start condition
	SWC2	Set to "1" for an "L" signal output from SCL forcibly	
	SDHI	Set to "1" to disable SDA output	
	SU1HIM	Set to "0"	
UiSMR3	SSE	Set to "0"	
	CKPH	See Table 16.14	
	DINC, NODC, ERR	Set to "0"	
	DL2 to DL0	Set digital delay value	
UiSMR4	STAREQ	Set to "1" when generating a start condition	Not used. Set to "0"
	RSTAREQ	Set to "1" when generating a restart condition	
	STPREQ	Set to "1" when generating a stop condition	
	STSPSEL	Set to "1" when using a condition generating function	
	ACKD	Select ACK or NACK	
	ACKC	Set to "1" for ACK data output	
	SCLHI	Set to "1" to enable SCL output stop when detecting stop condition	Not used. Set to "0"
	SWC9	Not used. Set to "0"	Set to "1" to fix an "L" signal output from SCLi on the falling edge of the ninth bit of the transfer clock
IFSR	IFSR6, IFSR7	Set to "1"	

i=0 to 4

**Table 16.14 I<sup>2</sup>C Mode Functions**

Function	Clock Synchronous Serial I/O Mode (SMD2 to SMD0=0012, IICM=0)	I <sup>2</sup> C Mode (SMD2 to SMD0=0102, IICM=1)			
		IICM2=0 (NACK/ACK interrupt)		IICM2=1 (UART transmit/UART receive interrupt)	
		CKPH=0 (No clock delay)	CKPH=1 (Clock delay)	CKPH=0 (No clock delay)	CKPH=1 (Clock delay)
Source for Interrupt Numbers 39 to 41 <sup>(1)</sup> (See Figure 16.20)	-	Start condition or stop condition detection (See Table 16.18)			
Source for Interrupt Numbers 17, 19, 33, 35, and 37 <sup>(1)</sup> (See Figure 16.20)	UARTi transmission - Transmission started or completed (selected by the UiIRS register)	No acknowledgement detection (NACK) - Rising edge of 9th bit of SCLi	UARTi transmission - Rising edge of 9th bit of SCLi	UARTi transmission - Next falling edge after the 9th bit of SCLi	
Source for Interrupt Numbers 18, 20, 34, 36, and 38 <sup>(1)</sup> (See Figure 16.20)	UARTi reception - Receiving at 8th bit CKPOL=0(rising edge) CKPOL=1(falling edge)	Acknowledgement detection (ACK) - Rising edge of 9th bit of SCLi	UARTi Reception - Falling edge of 9th bit of SCLi		
Data Transfer Timing from the UART Receive Shift Register to the UiRB Register	CKPOL=0(rising edge) CKPOL=1(falling edge)	Rising edge of 9th bit of SCLi	Falling edge of 9th bit of SCLi	Falling edge and rising edge of 9th bit of SCLi	
UARTi Transmit Output Delay	No delay	Delay			
P63, P67, P70, P92, P96 Pin Functions	TxDi output	SDAi input and output			
P62, P66, P71, P91, P97 Pin Functions	RxDi input	SCLi input and output			
P61, P65, P72, P90, P95 Pin Functions	Select CLKi input or output	- (Not used in I <sup>2</sup> C mode)			
Noise Filter Width	15 ns	200 ns			
Reading RxDi and SCLi Pin Levels	Can be read if port direction bit is set to "0"	Can be read regardless of the port direction bit			
Default Value of TxDi and SDAi Output	CKPOL=0 (H) CKPOL=1 (L)	Values set in the port register before entering I <sup>2</sup> C mode <sup>(2)</sup>			
SCLi Default and End Values	-	H	L	H	L
Source for DMA (See Figure 16.20)	UARTi reception	Acknowledgement detection (ACK)	UARTi reception - Falling edge of 9th bit of SCLi		
Store Received Data	1st to 8th bits of the received data are stored into bits 7 to 0 in the UiRB register	1st to 8th bits of the received data are stored into bits 7 to 0 in the UiRB register	1st to 7th bits of the received data are stored into bits 6 to 0 in the UiRB register. 8th bit is stored into bit 8 in the UiRB register.		
Reading Received Data	The UiRB register status is read				1st to 8th bits are stored into bits 7 to 0 in the UiRB register <sup>(3)</sup>
					Bits 6 to 0 in the UiRB registers <sup>(4)</sup> are read as bit 7 to 1. Bit 8 in the UiRB register is read as bit 0

i=0 to 4

## NOTES:

- Use the following procedure to change what causes an interrupt to be generated.
  - Disable interrupt of corresponding interrupt number.
  - Change what causes an interrupt to be generated.
  - Set the IR bit of a corresponding interrupt number to "0" (no interrupt requested).
  - Set the ILVL2 to ILVL0 bits of a corresponding interrupt number.
- Set default value of the SDAi output when the SMD2 to SMD0 bits in the UiMR register are set to "0002" (serial I/O disabled).
- Second data transfer to the UiRB register (on the rising edge of the ninth bit of SCLi).
- First data transfer to the UiRB register (on the falling edge of the ninth bit of SCLi).

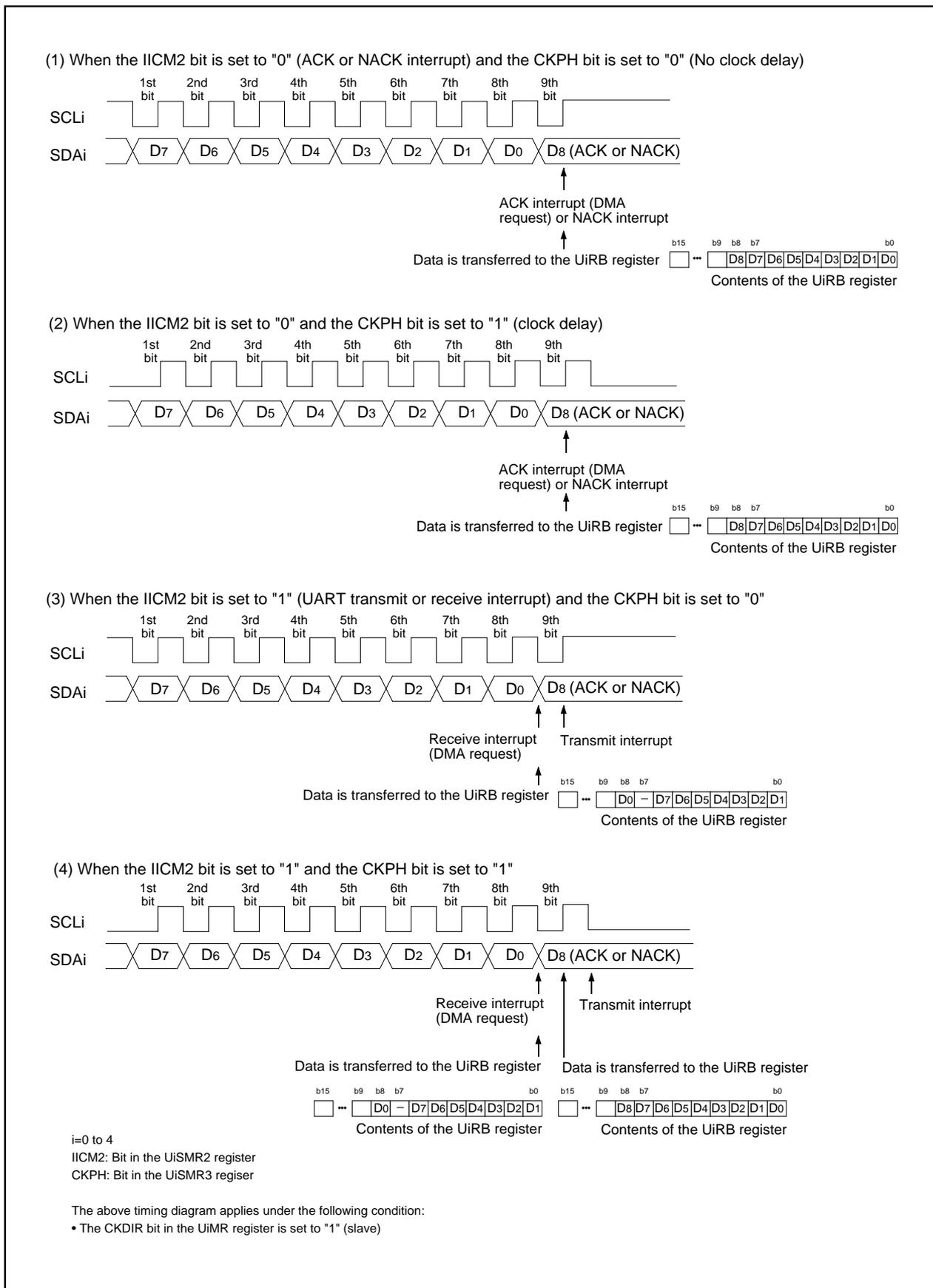


Figure 16.20 SCLi Timing

**Table 16.15 Pin Settings in I<sup>2</sup>C Mode (1)**

Port	Function	Setting		
		PS0 Register	PSL0 Register	PD6 Register
P62	SCL0 output	PS0_2=1	PSL0_2=0	-
	SCL0 input	PS0_2=0	-	PD6_2=0
P63	SDA0 output	PS0_3=1	-	-
	SDA0 input	PS0_3=0	-	PD6_3=0
P66	SCL1 output	PS0_6=1	PSL0_6=0	-
	SCL1 input	PS0_6=0	-	PD6_6=0
P67	SDA1 output	PS0_7=1	-	-
	SDA1 input	PS0_7=0	-	PD6_7=0

**Table 16.16 Pin Settings (2)**

Port	Function	Setting			
		PS1 Register	PSL1 Register	PSC Register	PD7 Register
P70 <sup>(1)</sup>	SDA2 output	PS1_0=1	PSL1_0=0	PSC_0=0	-
	SDA2 input	PS1_0=0	-	-	PD7_0=0
P71 <sup>(1)</sup>	SCL2 output	PS1_1=1	PSL1_1=1	PSC_1=0	-
	SCL2 input	PS1_1=0	-	-	PD7_1=0

NOTE:

1. P70 and P71 are ports for the N-channel open drain output.

**Table 16.17 Pin Settings (3)**

Port	Function	Setting			
		PS3 Register <sup>(1)</sup>	PSL3 Register	PSC3 Register	PD9 Register <sup>(1)</sup>
P91	SCL3 output	PS3_1=1	PSL3_1=0	-	-
	SCL3 input	PS3_1=0	-	-	PD9_1=0
P92	SDA3 output	PS3_2=1	PSL3_2=0	-	-
	SDA3 input	PS3_2=0	-	-	PD9_2=0
P96	SDA4 output	PS3_6=1	-	PSC3_6=0	-
	SDA4 input	PS3_6=0	-	-	PD9_6=0
P97	SCL4 output	PS3_7=1	PSL3_7=0	-	-
	SCL4 input	PS3_7=0	-	-	PD9_7=0

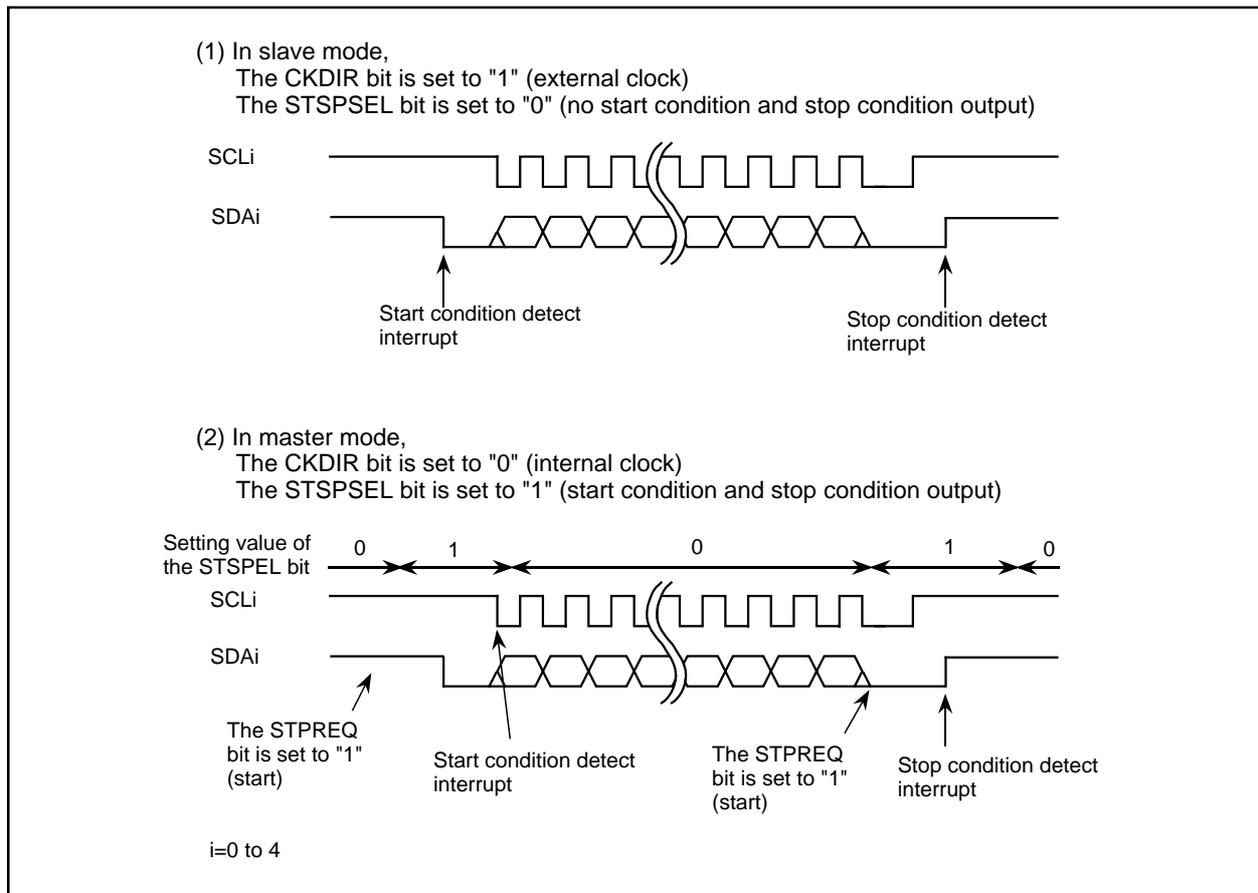
NOTE:

1. Set the PD9 and PS3 registers immediately after the PRC2 bit in the PRCR register is set to "1" (write enabled). Do not generate an interrupt or a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to set the PD9 and PS3 registers.



**Table 16.18 STSPSEL Bit Function**

Function	STSPSEL = 0	STSPSEL = 1
Start condition and stop condition output	Program with ports determines how the start condition or stop condition is output	The STAREQ bit, RSTAREQ bit and STPREQ bit determine how the start condition or stop condition is output
Timing to generate start condition and stop condition interrupt requests	Start condition and stop condition are detected	Start condition and stop condition generation are completed



**Figure 16.22 STSPSEL Bit Function**

### 16.3.3 Arbitration

The ABC bit in the UiSMR register (i=0 to 4) determines an update timing for the ABT bit in the UiRB register. On the rising edge of the SCLi pin, the microcomputer determines whether a transmit data matches data input to the SDAi pin.

When the ABC bit is set to "0" (update per bit), the ABT bit is set to "1" (detected-arbitration is lost) as soon as a data discrepancy is detected. The ABT bit is set to "0" (not detected-arbitration is won) if not detected. When the ABC bit is set to "1" (update per byte), the ABT bit is set to "1" on the falling edge of the ninth bit of the transfer clock if any discrepancy is detected. When the ABT bit is updated per byte, set the ABT bit to "0" between an ACK detection in the first byte data and the next byte data to be transferred. When the ALS bit in the UiSMR2 register is set to "1" (SDA output stop enabled), the arbitration lost occurs. As soon as the ABT bit is set to "1", the SDAi pin is placed in a high-impedance state.

### 16.3.4 Transfer Clock

The transfer clock transmits and receives data as is shown in Figure 16.20.

The CSC bit in the UiSMR2 register (i=0 to 4) synchronizes an internally generated clock (internal SCLi) with the external clock applied to the SCLi pin. When the CSC bit is set to "1" (clock synchronous enabled) and the internal SCLi is held high ("H"), the internal SCLi become low ("L") if signal applied to the SCLi pin is on the falling edge. Value of the UiBRG register is reloaded to start counting for low level. A counter stops when the SCLi pin is held "L" and then the internal SCLi changes "L" to "H". Counting is resumed when the SCLi pin become "H". The transfer clock of UARTi is equivalent to the AND for signals from the internal SCLi and the SCLi pin.

The transfer clock is synchronized between a half cycle before the falling edge of first bit of the internal SCLi and the rising edge of the ninth bit. Select the internal clock as the transfer clock while the CSC bit is set to "1".

The SWC bit in the UiSMR2 register determines whether the SCLi pin is fixed to be an "L" signal output on the falling edge of the ninth cycle of the transfer clock or not.

When the SCLHI bit in the UiSMR4 register is set to "1" (enabled), a SCLi output stops when a stop condition is detected (high-impedance).

When the SWC2 bit in the UiSMR2 register is set to "1" (0 output), the SCLi pin forcibly outputs an "L" signal while transmitting and receiving. The fixed "L" signal applied to the SCLi pin is cancelled by setting the SWC2 bit to "0" (transfer clock) and the transfer clock input to and output from the SCLi pin are provided.

When the CKPH bit in the UiSMR3 register is set to "1" and the SWC9 bit in the UiSMR4 register is set to "1" (SCL "L" hold enabled), the SCLi pin is fixed to be an "L" signal output on the next falling edge after the ninth bit of the clock. The fixed "L" signal applied to the SCLi pin is cancelled by setting the SWC9 bit to "0" (SCL "L" hold disabled).

### 16.3.5 SDA Output

Values output set in bits 7 to 0 (D7 to D0) in the UiTB register (i=0 to 4) are provided in descending order from D7. The ninth bit (D8) is ACK or NACK.

Set the default value of SDAi transmit output when the IICM bit is set to "1" (I<sup>2</sup>C mode) and the SMD2 to SMD0 bits in the UiMR register are set to "0002" (serial I/O disabled).

The DL2 to DL0 bits in the UiSMR3 register determine no delay in the SDAi output or a delay of 2 to 8 UiBRG register count source cycles.

When the SDHI bit in the UiSMR2 register is set to "1" (SDA output disabled), the SDAi pin is forcibly placed in a high-impedance state. Do not set the SDHI bit on the rising edge of the UARTi transfer clock. The ABT bit in the UiRB register may be set to "1" (detected).

### 16.3.6 SDA Input

When the IICM2 bit in the UiSMR2 register (i=0 to 4) is set to "0", the first eight bits of received data are stored into bits 7 to 0 (D7 to D0) in the UiRB register. The ninth bit (D8) is ACK or NACK.

When the IICM2 bit is set to "1", the first seven bits (D7 to D1) of received data are stored into bits 6 to 0 in the UiRB register. Store the eighth bit (D0) into bit 8 in the UiRB register.

If the IICM2 bit is set to "1" and the CKPH bit in the UiSMR3 register is set to "1", the same data as that of when setting the IICM2 bit to "0" can be read. To read the data, read the UiRB register after the rising edge of the ninth bit of the transfer clock.

### 16.3.7 ACK, NACK

When the STPSEL bit in the UiSMR4 register (i=0 to 4) is set to "0" (serial I/O circuit selected) and the ACKC bit in the UiSMR4 register is set to "1" (ACK data output), the SDAi pin provides the value output set in the ACKD bit in the UiSMR4 register.

If the IICM2 bit is set to "0", the NACK interrupt request is generated when the SDAi pin is held high ("H") on the rising edge of the ninth bit of the transfer clock. The ACK interrupt request is generated when the SDAi pin is held low ("L") on the rising edge of the ninth bit of the transfer clock.

When ACK is selected to generate a DMA request, the DMA transfer is activated by an ACK detection.

### 16.3.8 Transmit and Receive Reset

When the STC bit in the UiSMR2 register (i=0 to 4) is set to "1" (UARTi initialization enabled) and a start condition is detected,

- the transmit shift register is reset and the content of the UiTB register is transferred to the transmit shift register. The first bit starts transmitting when the next clock is input. UARTi output value remains unchanged between when the clock is applied and when the first bit data output is provided. The value remains the same as when start condition was detected.
- the receive shift register is reset and the first bit start receiving when the next clock is applied.
- the SWC bit is set to "1" (SCL wait output enabled). The SCLi pin becomes "L" on the falling edge of the ninth bit of the transfer clock.

If UARTi transmission and reception are started with this function, the TI bit in the UiC1 register remains unchanged. Select the external clock as the transfer clock when using this function.

## 16.4 Special Mode 2

In special mode 2, serial communication between one or multiple masters and multiple slaves is available. The  $\overline{SSi}$  input pin ( $i=0$  to 4) controls the serial bus communication. Table 16.19 lists specifications of special mode 2. Table 16.20 lists register settings. Tables 16.21 to 16.23 list pin settings.

**Table 16.19 Special Mode 2 Specifications**

Item	Specification
Transfer Data Format	Transfer data : 8 bits long
Transfer Clock	<ul style="list-style-type: none"> <li>The CKDIR bit in the UiMR register (<math>i=0</math> to 4) is set to "0" (internal clock selected):  <math>f_j/2(m+1)</math> <math>f_j = f_1, f_8, f_{2n}^{(1)}</math> <math>m</math> : setting value of the UiBRG register, 0016 to FF16</li> <li>The CKDIR bit to "1" (external clock selected) : input from the CLKi pin</li> </ul>
Transmit/Receive Control	$\overline{SSi}$ input pin function
Transmit Start Condition	To start transmitting, the following requirements must be met <sup>(2)</sup> : <ul style="list-style-type: none"> <li>- Set the TE bit in the UiC1 register to "1" (transmit enabled)</li> <li>- Set the TI bit in the UiC1 register to "0" (data in the UiTB register)</li> </ul>
Receive Start Condition	To start receiving, the following requirement must be met <sup>(2)</sup> : <ul style="list-style-type: none"> <li>- Set the RE bit in the UiC1 register to "1" (receive enabled)</li> <li>- Set the TE bit in the UiC1 register to "1" (transmit enabled)</li> <li>- Set the TI bit in the UiC1 register to "0" (data in the UiTB register)</li> </ul>
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>While transmitting, the following conditions can be selected: <ul style="list-style-type: none"> <li>- The UiIRS bit in the UiC1 register is set to "0" (no data in a transmit buffer) : when data is transferred from the UiTB register to the UARTi transmit register (transmission started)</li> <li>- The UiIRS register is set to "1" (transmission completed): when data transmission from UARTi transfer register is completed</li> </ul> </li> <li>While receiving When data is transferred from the UARTi receive register to the UiRB register (reception completed)</li> </ul>
Error Detection	<ul style="list-style-type: none"> <li>• Overrun error<sup>(3)</sup> This error occurs when the seventh bit of the next received data is read before reading the UiRB register</li> <li>• Fault error In master mode, the fault error occurs an "L" signal is applied to the <math>\overline{SSi}</math> pin</li> </ul>
Selectable Function	<ul style="list-style-type: none"> <li>• CLK polarity Selectable from the rising edge or falling edge of the transfer clock at transferred data output or input timing</li> <li>• LSB first or MSB first Selectable from data transmission or reception in either bit 0 or in bit 7</li> <li>• Continuous receive mode Reception is enabled simultaneously by reading the UiRB register</li> <li>• Serial data logic inverse This function inverses transmitted or received data logically</li> <li>• TxD and RxD I/O polarity inverse TxD pin output and RxD pin input are inversed. All I/O data levels are also inversed</li> <li>• Clock phase Selectable from one of 4 combinations of transfer data polarity and phases</li> <li>• <math>\overline{SSi}</math> input pin function Output pin is placed in a high-impedance state to avoid data conflict between master and other masters or slaves</li> </ul>

NOTES:

- The CNT3 to CNT0 bits in the TCSPPR register select no division ( $n=0$ ) or divide-by-2n ( $n=1$  to 15).
- To start transmission/reception when selecting the external clock, these conditions must be met after the CKPOL bit in the UiC0 register is set to "0" (data is transmitted on the falling edge of the transfer clock and data is received on the rising edge) and the CLKi pin is held high ("H"), or when the CKPOL bit is set to "1" (Data is transmitted on the rising edge of the transfer clock and data is received on the falling edge) and the CLKi pin is held low ("L").
- If an overrun error occurs, the UiRB register is in an indeterminate state. The IR bit setting in the SiRIC register does not change to "1" (interrupt requested).

**Table 16.20 Register Settings in Special Mode 2**

Register	Bit	Function
UiTB	7 to 0	Set transmit data
UiRB	7 to 0	Received data can be read
	OER	Overflow error flag
UiBRG	7 to 0	Set bit rate
UiMR	SMD2 to SMD0	Set to "0012"
	CKDIR	Set to "0" in master mode or "1" in slave mode
	IOPOL	Set to "0"
UiC0	CLK1, CLK0	Select count source for the UiBRG register
	CRS	Disabled because the CRD bit is set to "1"
	TXEPT	Transfer register empty flag
	CRD	Set to "1"
	NCH	Select the output format of the TxDi pin
	CKPOL	Clock phase can be set by the combination of the CKPOL bit and the CKPH bit in the UiSMR3 register
	UFORM	Select either LSB first or MSB first
UiC1	TE	Set to "1" to enable data transmission and reception
	TI	Transfer buffer empty flag
	RE	Set to "1" to enable data reception
	RI	Reception complete flag
	UIRS	Select what causes the UARTi transmit interrupt to be generated
	UiRRM	Set to "1" to enable continuous receive mode
	UiLCH, SCLKSTPB	Set to "0"
UiSMR	7 to 0	Set to "0016"
UiSMR2	7 to 0	Set to "0016"
UiSMR3	SSE	Set to "1"
	CKPH	Clock phase can be set by the combination of the CKPH bit and the CKPOL bit in the UiC0 register
	DINC	Set to "0" in master mode or "1" in slave mode
	NODC	Set to "0"
	ERR	Fault error flag
	7 to 5	Set to "0002"
UiSMR4	7 to 0	Set to "0016"

i=0 to 4

**Table 16.21 Pin Settings in Special Mode 2 (1)**

Port	Function	Setting		
		PS0 Register	PSL0 Register	PD6 Register
P60	SS0 input	PS0_0=0	–	PD6_0=0
P61	CLK0 input (slave)	PS0_1=0	–	PD6_1=0
	CLK0 output (master)	PS0_1=1	–	–
P62	RxD0 input (master)	PS0_2=0	–	PD6_2=0
	STxD0 output (slave)	PS0_2=1	PSL0_2=1	–
P63	TxD0 output (master)	PS0_3=1	–	–
	SRxD0 input (slave)	PS0_3=0	–	PD6_3=0
P64	SS1 input	PS0_4=0	–	PD6_4=0
P65	CLK1 input (slave)	PS0_5=0	–	PD6_5=0
	CLK1 output (master)	PS0_5=1	–	–
P66	RxD1 input (master)	PS0_6=0	–	PD6_6=0
	STxD1 output (slave)	PS0_6=1	PSL0_6=1	–
P67	TxD1 output (master)	PS0_7=1	–	–
	SRxD1 input (slave)	PS0_7=0	–	PD6_7=0

**Table 16.22 Pin Settings (2)**

Port	Function	Setting			
		PS1 Register	PSL1 Register	PSC Register	PD7 Register
P70 <sup>(1)</sup>	TxD2 output (master)	PS1_0=1	PSL1_0=0	PSC_0=0	–
	SRxD2 input (slave)	PS1_0=0	–	–	PD7_0=0
P71 <sup>(1)</sup>	RxD2 input (master)	PS1_1=0	–	–	PD7_1=0
	STxD2 output (slave)	PS1_1=1	PSL1_1=1	PSC_1=0	–
P72	CLK2 input (slave)	PS1_2=0	–	–	PD7_2=0
	CLK2 output (master)	PS1_2=1	PSL1_2=0	PSC_2=0	–
P73	SS2 input	PS1_3=0	–	–	PD7_3=0

NOTE:

1. P70 and P71 are ports for the N-channel open drain output.

**Table 16.23 Pin Settings (3)**

Port	Function	Setting			
		PS3 Register <sup>(1)</sup>	PSL3 Register	PSC3 Register	PD9 Register <sup>(1)</sup>
P90	CLK3 input (slave)	PS3_0=0	–	–	PD9_0=0
	CLK3 output (master)	PS3_0=1	–	–	–
P91	RxD3 input (master)	PS3_1=0	–	–	PD9_1=0
	STxD3 output (slave)	PS3_1=1	PSL3_1=1	–	–
P92	TxD3 output (master)	PS3_2=1	PSL3_2=0	–	–
	SRxD3 input (slave)	PS3_2=0	–	–	PD9_2=0
P93	SS3 input	PS3_3=0	PSL3_3=0	–	PD9_3=0
P94	SS4 input	PS3_4=0	PSL3_4=0	–	PD9_4=0
P95	CLK4 input (slave)	PS3_5=0	PSL3_5=0	–	PD9_5=0
	CLK4 output (master)	PS3_5=1	–	–	–
P96	TxD4 output (master)	PS3_6=1	–	PSC3_6=0	–
	SRxD4 input (slave)	PS3_6=0	PSL3_6=0	–	PD9_6=0
P97	RxD4 input (master)	PS3_7=0	–	–	PD9_7=0
	STxD4 output (slave)	PS3_7=1	PSL3_7=1	–	–

NOTE:

1. Set the PD9 and PS3 registers immediately after the PRC2 bit in the PRCR register is set to "1" (write enabled). Do not generate an interrupt or a DMA transfer between the instruction to set to the PRC2 bit to "1" and the instruction to set the PD9 and PS3 registers.

### 16.4.1 $\overline{SS}_i$ Input Pin Function (i=0 to 4)

When the SSE bit in the UiSMR3 register is set to "1" ( $\overline{SS}$  function enabled), the special mode 2 is selected, activating the pin function.

The DINC bit in the UiSMR3 register determines which microcomputer performs as master or slave. When multiple microcomputers perform as the masters (multi-master system), the  $\overline{SS}_i$  pin setting determines which master microcomputer is active and when.

#### 16.4.1.1 When Setting the DINC Bit to "1" (Slave Mode)

When a high-level ("H") signal is applied to the  $\overline{SS}_i$  pin, the STxDi and SRxDi pins are placed in a high-impedance state and the transfer clock applied to the CLKi pin is ignored. When a low-level ("L") signal is applied to the  $\overline{SS}_i$  input pin, the transfer clock input is valid and serial communication is enabled.

#### 16.4.1.2 When Setting the DINC Bit to "0" (Master Mode)

When using the  $\overline{SS}_i$  pin function in master mode, set the UilRS bit in the UiC1 register to "1" (transmission completed).

When an "H" signal is applied to the  $\overline{SS}_i$  pin, serial communication is available due to transmission privilege. The master provides the transfer clock output. When an "L" signal is applied to the  $\overline{SS}_i$  pin, it indicates that another master is active. The TxDi and CLKi pins are placed in high-impedance states and the ERR bit in the UiSMR3 register is set to "1" (fault error). Use the transmit complete interrupt routine to verify the ERR bit state.

To resume the serial communication after the fault error occurs, set the ERR bit to "0" while applying the "H" signal to the  $\overline{SS}_i$  pin. The TxDi and CLKi pins become ready for signal outputs.

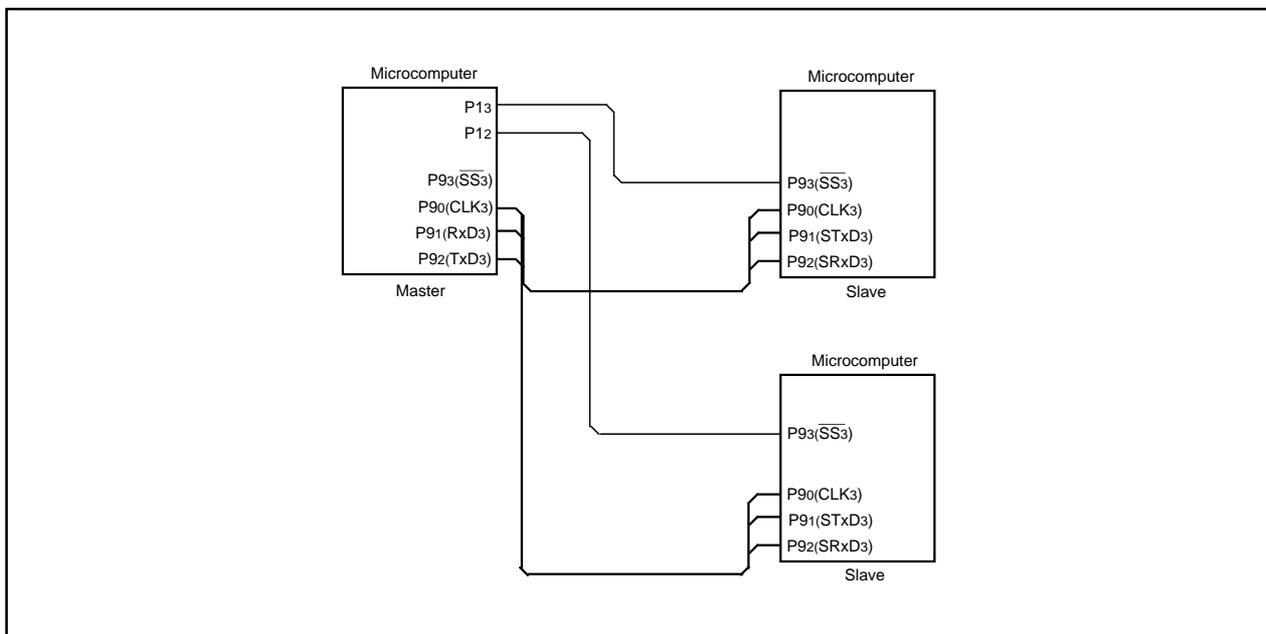


Figure 16.23 Serial Bus Communication Control with SS Pin

### 16.4.2 Clock Phase Setting Function

The CKPH bit in the UiSMR3 register (i=0 to 4) and the CKPOL bit in the UiC0 register select one of four combinations of transfer clock polarity and phases.

The transfer clock phase and polarity must be the same between the master and the slave involved in the transfer.

#### 16.4.2.1 When setting the DINC Bit to "0" (Master (Internal Clock))

Figure 16.24 shows transmit and receive timing.

#### 16.4.2.2 When Setting the DINC Bit to "1" (Slave (External Clock))

When the CKPH bit is set to "0" (no clock delay) and the  $\overline{SSi}$  input pin is held high ("H"), the STxDi pin is placed in a high-impedance state. When the  $\overline{SSi}$  input pin becomes low ("L"), conditions to start a serial transfer are met, but output is indeterminate. The serial transmission is synchronized with the transfer clock. Figure 16.25 shows the transmit and receive timing.

When the CKPH bit is set to "1" (clock delay) and the  $\overline{SSi}$  input pin is held high, the STxDi pin is placed in a high-impedance state. When the  $\overline{SSi}$  pin becomes low, the first data is output. The serial transmission is synchronized with the transfer clock. Figure 16.26 shows the transmit and receive timing.

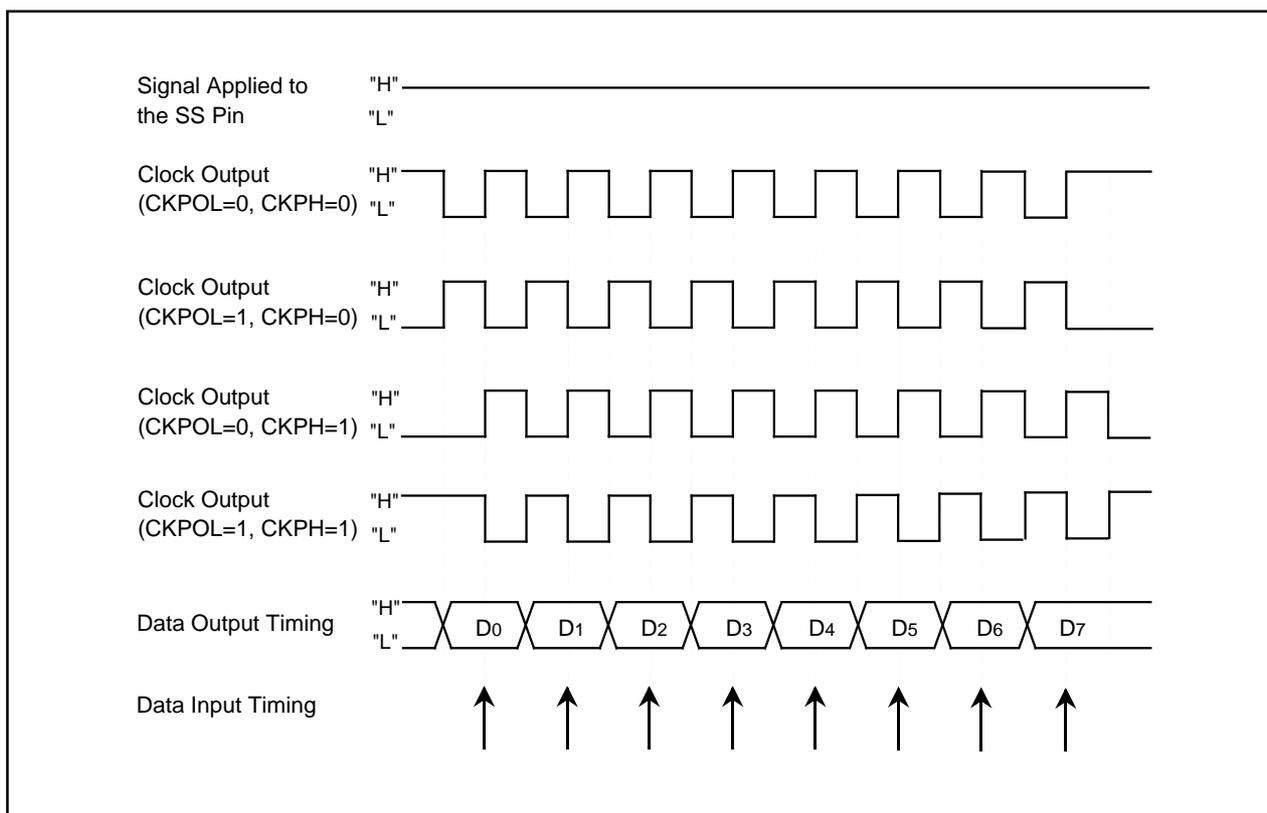


Figure 16.24 Transmit and Receive Timing in Master Mode (Internal Clock)

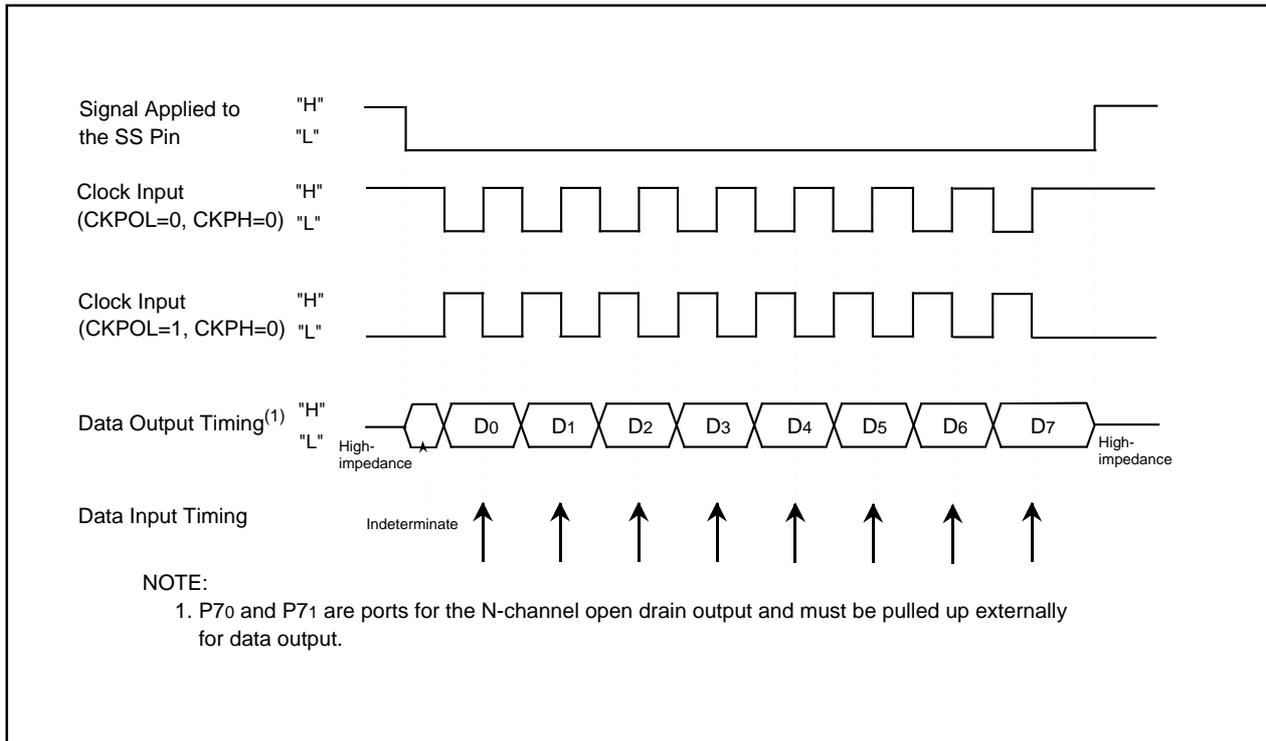


Figure 16.25 Transmit and Receive Timing in Slave Mode (External Clock) (CKPH=0)

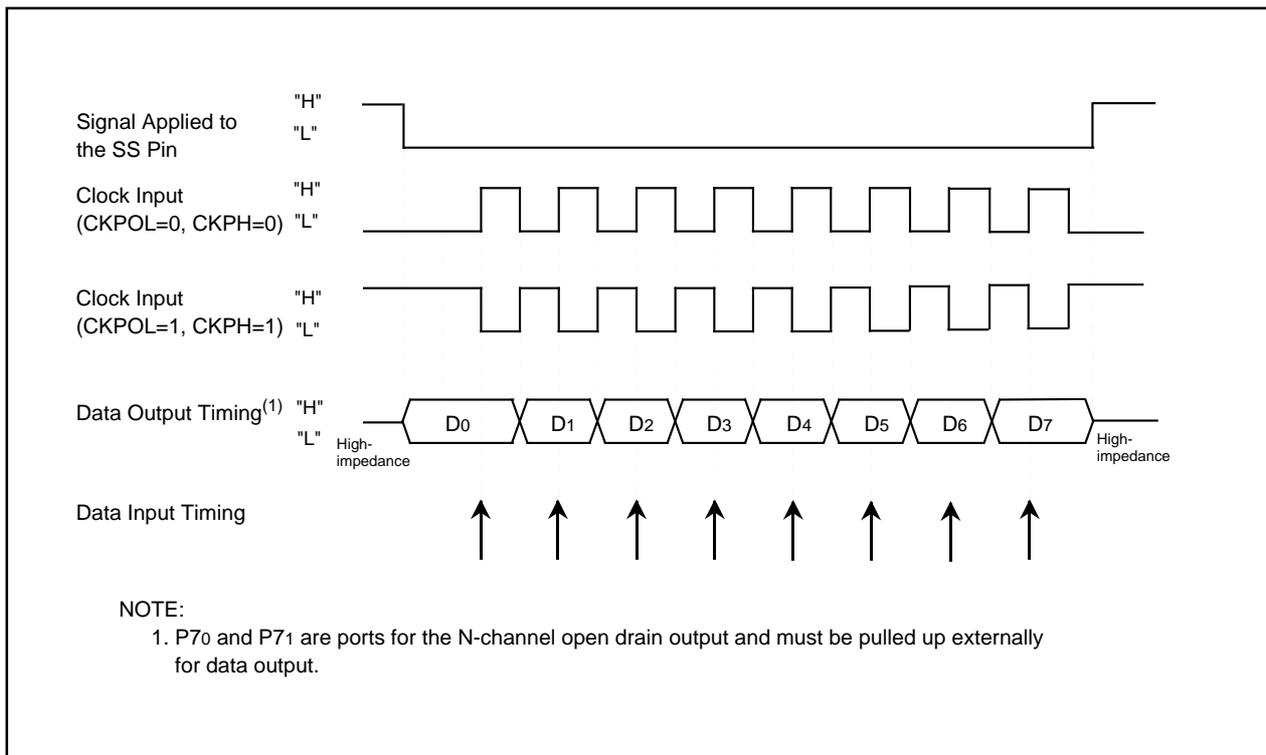


Figure 16.26 Transmit and Receive Timing in Slave Mode (External Clock) (CKPH=1)

### 16.5 Special Mode 3 (GCI Mode)

In GCI mode, the external clock is synchronized with the transfer clock used in the clock synchronous serial I/O mode.

Table 16.24 lists specifications of GCI mode. Table 16.25 lists registers settings. Tables 16.26 to 16.28 list pin settings.

**Table 16.24 GCI Mode Specifications**

Item	Specification
Transfer Data Format	Transfer data : 8 bits long
Transfer Clock	The CKDIR bit in the UiMR register (i=0 to 4) is set to "1" (external clock selected): input from the CLKi pin
Clock Synchronization Function	Trigger signal input from the $\overline{CTS}_i$ pin
Transmit/Receive Start Condition	To start data transmission and reception, meet the following conditions and then apply a trigger signal to the $\overline{CTS}_i$ pin: <ul style="list-style-type: none"> <li>- Set the TE bit in the UiC1 register to "1" (transmit enabled)</li> <li>- Set the RE bit in the UiC1 register to "1" (receive enabled)</li> <li>- Set the TI bit in the UiC1 register to "0" (Data in the UiTB register)</li> </ul>
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>• While transmitting, the following condition can be selected: <ul style="list-style-type: none"> <li>- The UiIRS bit in the UiC1 register is set to "0" (UiTB register empty): when data is transferred from the UiTB register to the UARTi transmit register (transmission started)</li> <li>- The UiIRS bit is set to "1" (Transmit completed): when a data transmission from the UARTi transfer register is completed</li> </ul> </li> <li>• While receiving, when data is transferred from the UARTi receive register to the UiRB register (reception completed)</li> </ul>
Error Detection	<p>Overflow error<sup>(1)</sup></p> <p>This error occurs when the seventh bit of the next received data is read before reading the UiRB register.</p>

**NOTE:**

1. If an overflow error occurs, the UiRB register is indeterminate. The IR bit setting in the SiRIC register does not change to "1" (interrupt requested).

**Table 16.25 Register Settings in GCI Mode**

Register	Bit	Function
UiTB	7 to 0	Set transmit data
UiRB	7 to 0	Received data
	OER	Overflow error flag
UiBRG	7 to 0	Set to "0016"
UiMR	SMD2 to SMD0	Set to "0012"
	CKDIR	Set to "1"
	IOPOL	Set to "0"
UiC0	CLK1, CLK0	Set to "002"
	CRS	Disabled because the CRD bit is set to "1"
	TXEPT	Transfer register empty flag
	CRD	Set to "1"
	NCH	Select the output format of the TxDi pin
	CKPOL	Set to "0"
	UFORM	Set to "0"
UiC1	TE	Set to "1" to enable data transmission and reception
	TI	Transfer buffer empty flag
	RE	Set to "1" to enable data reception
	RI	Reception complete flag
	UiIRS	Select what causes the UARTi transmit interrupt to be generated
	UiRRM, UiLCH	Set to "0"
	SCLKSTPB	Set to "0"
UiSMR	6 to 0	Set to "00000002"
	SCLKDIV	See Table 16.29
UiSMR2	6 to 0	Set to "00000002"
	SU1HIM	See Table 16.29
UiSMR3	2 to 0	Set to "0002"
	NODC	Set to "0"
	7 to 4	Set to "00002"
UiSMR4	7 to 0	Set to "0016"

i=0 to 4

**Table 16.26 Pin Settings in GCI Mode (1)**

Port	Function	Setting	
		PS0 Register	PD6 Register
P60	$\overline{\text{CTS0}}$ input <sup>(1)</sup>	PS0_0=0	PD6_0=0
P61	CLK0 input	PS0_1=0	PD6_1=0
P62	RxD0 input	PS0_2=0	PD6_2=0
P63	TxD0 output	PS0_3=1	–
P64	$\overline{\text{CTS1}}$ input <sup>(1)</sup>	PS0_4=0	PD6_4=0
P65	CLK1 input	PS0_5=0	PD6_5=0
P66	RxD1 input	PS0_6=0	PD6_6=0
P67	TxD1 output	PS0_7=1	–

NOTE:

1. CTS input is used as a trigger signal input.

**Table 16.27 Pin Settings (2)**

Port	Function	Setting			
		PS1 Register	PSL1 Register	PSC Register	PD7 Register
P70 <sup>(1)</sup>	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	–
P71 <sup>(1)</sup>	RxD2 input	PS1_1=0	–	–	PD7_1=0
P72	CLK2 input	PS1_2=0	–	–	PD7_2=0
P73	$\overline{\text{CTS2}}$ input <sup>(2)</sup>	PS1_3=0	–	–	PD7_3=0

NOTES:

1. P70 and P71 are ports for the N-channel open drain output.
2. CTS input is used as a trigger signal input.

**Table 16.28 Pin Settings (3)**

Port	Function	Setting			
		PS3 Register <sup>(1)</sup>	PSL3 Register	PSL3 Register	PD9 Register <sup>(1)</sup>
P90	CLK3 input	PS3_0=0	–	–	PD9_0=0
P91	RxD3 input	PS3_1=0	–	–	PD9_1=0
P92	TxD3 output	PS3_2=1	PSL3_2=0	–	–
P93	$\overline{\text{CTS3}}$ input <sup>(2)</sup>	PS3_3=0	PSL3_3=0	–	PD9_3=0
P94	$\overline{\text{CTS4}}$ input <sup>(2)</sup>	PS3_4=0	PSL3_4=0	–	PD9_4=0
P95	CLK4 input	PS3_5=0	PSL3_5=0	–	PD9_5=0
P96	TxD4 output	PS3_6=1	PSL3_6=0	PSL3_6=0	–
P97	RxD4 input	PS3_7=0	–	–	PD9_7=0

NOTES:

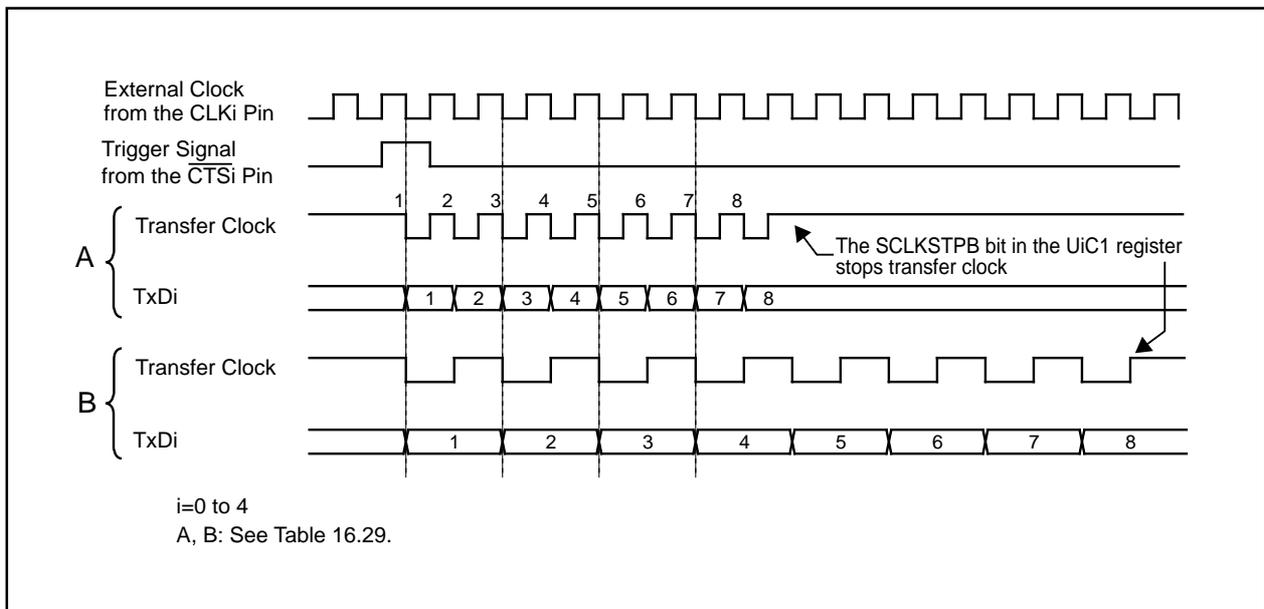
1. Set the PD9 and PS3 registers immediately after the PRC2 bit in the PRCR register is set to "1" (write enabled). Do not generate an interrupt or a DMA transfer between the instruction to set to the PRC2 bit to "1" and the instruction to set the PD9 and PS3 registers.
2. CTS input is used for a trigger signal input.

To generate the internal clock synchronized with the external clock, set the SU1HIM bit in the UiSMR2 register (i=0 to 4) and the SCLKDIV bit in the UiSMR register to values shown in Table 16.29. Then apply a trigger signal to the  $\overline{CTS_i}$  pin. Either the same clock cycle as the external clock or external clock divided by two can be selected as the transfer clock. The SCLKSTPB bit in the UiC1 register controls the transfer clock. Set the SCLKSTPB bit accordingly, to start or stop the transfer clock during an external clock operation. Figure 16.27 shows an example of the clock-divided synchronous function.

**Table 16.29 Clock-Divided Synchronous Function Select**

SCLKDIV Bit in UiSMR Register	SU1HIM Bit in UiSMR2 Register	Clock-Divided Synchronous Function	Example of Waveform
0	0	Not synchronized	-
0	1	Same division as the external clock	A in Figure 16.27
1	0 or 1	Same division as the external clock divided by 2	B in Figure 16.27

i=0 to 4



**Figure 16.27 Clock-Divided Synchronous Function**

## 16.6 Special Mode 4 (IE Mode)

In IE mode, devices connected with the IEBus can communicate in UART mode.

Table 16.30 lists register settings. Tables 16.31 to 16.33 list pin settings.

**Table 16.30 Register Settings in IE Mode**

Register	Bit	Function
UiTB	8 to 0	Set transmit data
UiRB	8 to 0	Received data can be read
	OER, FER, PER, SUM	Error flags
UiBRG	7 to 0	Set bit rate
UiMR	SMD2 to SMD0	Set to "1102"
	CKDIR	Select the internal clock or external clock
	STPS	Set to "0"
	PRY	Disabled because the PRYE bit is set to "0"
	PRYE	Set to "0"
	IOPOL	Select TxD and RxD I/O polarity
UiC0	CLK1, CLK0	Select count source for the UiBRG register
	CRS	Disabled because the CRD bit is set to "1"
	TXEPT	Transfer register empty flag
	CRD	Set to "1"
	NCH	Select output format of the TxDi pin
	CKPOL	Set to "0"
	UFORM	Set to "0"
UiC1	TE	Set to "1" to enable data transmission
	TI	Transfer buffer empty flag
	RE	Set to "1" to enable data reception
	RI	Reception complete flag
	UiIRS	Select what causes the UARTi transmit interrupt to be generated
	UiRRM, UiLCH, SCLKSTPB	Set to "0"
UiSMR	3 to 0	Set to "00002"
	ABSCS	Select bus conflict detect sampling timing
	ACSE	Set to "1" to automatically clear the transmit enable bit
	SSS	Select transmit start condition
	SCLKDIV	Set to "0"
UiSMR2	7 to 0	Set to "0016"
UiSMR3	7 to 0	Set to "0016"
UiSMR4	7 to 0	Set to "0016"
IFSR	IFSR6, IFSR7	Select how the bus conflict interrupt occurs

i=0 to 4

**Table 16.31 Pin Settings in IE Mode (1)**

Port	Function	Setting	
		PS0 Register	PD6 Register
P61	CLK0 input	PS0_1=0	PD6_1=0
	CLK0 output	PS0_1=1	–
P62	RxD0 input	PS0_2=0	PD6_2=0
P63	TxD0 output	PS0_3=1	–
P65	CLK1 input	PS0_5=0	PD6_5=0
	CLK1 output	PS0_5=1	–
P66	RxD1 input	PS0_6=0	PD6_6=0
P67	TxD1 output	PS0_7=1	–

**Table 16.32 Pin Settings (2)**

Port	Function	Setting			
		PS1 Register	PSL1 Register	PSC Register	PD7 Register
P70 <sup>(1)</sup>	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	–
P71 <sup>(1)</sup>	RxD2 input	PS1_1=0	–	–	PD7_1=0
P72	CLK2 input	PS1_2=0	–	–	PD7_2=0
	CLK2 output	PS1_2=1	PSL1_2=0	PSC_2=0	–

NOTE:

1. P70 and P71 are ports for the N-channel open drain output.

**Table 16.33 Pin Settings (3)**

Port	Function	Setting			
		PS3 Register <sup>(1)</sup>	PSL3 Register	PSC3 Register	PD9 Register <sup>(1)</sup>
P90	CLK3 input	PS3_0=0	–	–	PD9_0=0
	CLK3 output	PS3_0=1	–	–	–
P91	RxD3 input	PS3_1=0	–	–	PD9_1=0
P92	TxD3 output	PS3_2=1	PSL3_2=0	–	–
P95	CLK4 input	PS3_5=0	PSL3_5=0	–	PD9_5=0
	CLK4 output	PS3_5=1	–	–	–
P96	TxD4 output	PS3_6=1	–	PSC3_6=0	–
P97	RxD4 input	PS3_7=0	–	–	PD9_7=0

NOTE:

1. Set the PD9 and PS3 registers immediately after the PRC2 bit in the PRCR register is set to "1" (write enabled). Do not generate an interrupt or a DMA transfer between the instruction to set to the PRC2 bit to "1" and the instruction to set the PD9 and PS3 registers.

If the output signal level of the TxDi pin (i=0 to 4) differs from the input signal level of the RxDi pin, an interrupt request is generated.

UART0 and UART3 are assigned software interrupt number 40. UART1 and UART4 are assigned number 41. When using the bus conflict detect function of UART0 or UART3, of UART1 or UART4, set the IFSR6 bit and the IFSR7 bit in the IFSR register accordingly.

When the ABSCS bit in the UiSMR register is set to "0" (rising edge of the transfer clock), it is determined, on the rising edge of the transfer clock, if the output level of the TxD pin and the input level of the RxD pin match. When the ABSCS bit is set to "1" (timer Aj underflow), it is determined when the timer Aj (timer A3 in UART0, timer A4 in UART1, timer A0 in UART2, timer A3 in UART3, the timer A4 in UART4) counter overflows. Use the timer Aj in one-shot timer mode.

When the ACSE bit in the UiSMR register is set to "1" (automatic clear at bus conflict) and the IR bit in the BCNiIC register to "1" (discrepancy detected), the TE bit in the UiC1 register is set to "0" (transmit disable).

When the SSS bit in the UiSMR register is set to "1" (synchronized with RxDi), data is transmitted from the TxDi pin on the falling edge of the RxDi pin. Figure 16.28 shows bits associated with the bus conflict detect function.

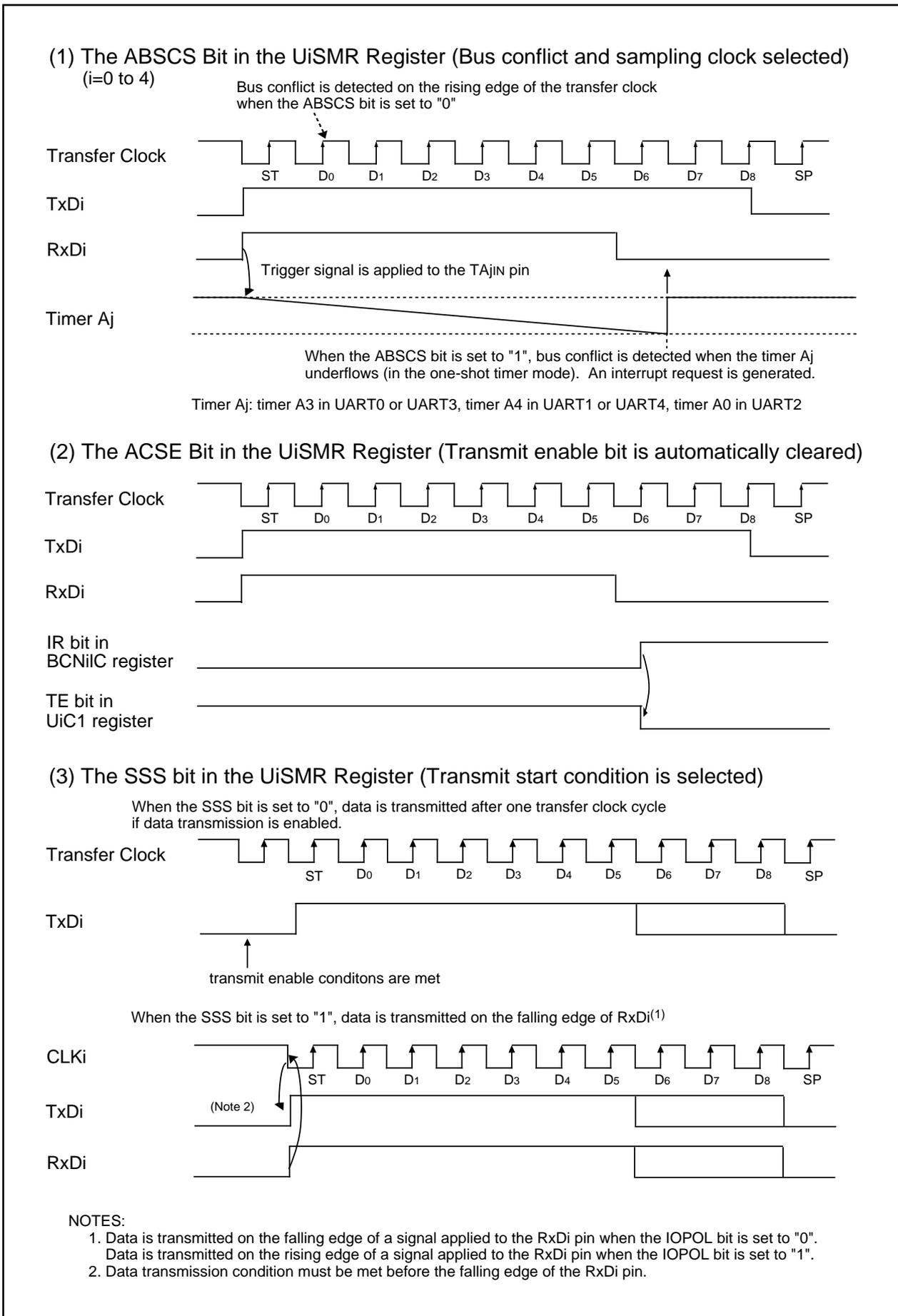


Figure 16.28 Bit Function Related Bus Conflict Detection

## 16.7 Special Mode 5 (SIM Mode)

In SIM mode, SIM interface devices can communicate in UART mode. Both direct and inverse formats are available and a low-level ("L") signal output can be provided from the TxDi pin (i=0 to 4) when a parity error is detected.

Table 16.34 lists specifications of SIM mode. Table 16.35 lists register settings. Tables 16.36 to 16.38 list pin settings.

**Table 16.34 SIM Mode Specifications**

Item	Specification
Transfer Data Format	<ul style="list-style-type: none"> <li>• Transfer data: 8-bit UART mode</li> <li>• In direct format</li> <li>Parity: Even</li> <li>Data logic: Direct</li> <li>Transfer format: LSB first</li> </ul> <ul style="list-style-type: none"> <li>• One stop bit</li> <li>• In inverse format</li> <li>Parity: Odd</li> <li>Data logic: Inverse</li> <li>Transfer format: MSB first</li> </ul>
Transfer Clock	<ul style="list-style-type: none"> <li>• The CKDIR bit in the UiMR register (i=0 to 4) is "0" (internal clock selected):  <math>f_i/16(m+1)^{f_1}</math> <math>f_j = f_1, f_8, f_{2n}^{(2)}</math> <math>m</math>: setting value of the UiBRG register, 00<sub>16</sub> to FF<sub>16</sub></li> <li>Do not set the CKDIR bit to "1" (external clock selected)</li> </ul>
Transmit/Receive Control	The CRD bit in the UiC0 register is set to "1" ( $\overline{\text{CTS}}$ , $\overline{\text{RTS}}$ function disabled)
Other Setting Items	The UiIRS bit in the UiC1 register is set to "1" (transmission completed)
Transmit Start Condition	To start transmitting, the following requirements must be met: <ul style="list-style-type: none"> <li>- Set the TE bit in the UiC1 register to "1" (transmit enabled)</li> <li>- Set the TI bit in the UiC1 register to "0" (data in the UiTB register)</li> </ul>
Receive Start Condition	To start receiving, the following requirements must be met: <ul style="list-style-type: none"> <li>- Set the RE bit in the UiC1 register to "1" (receive enabled)</li> <li>- Detect the start bit</li> </ul>
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>• While transmitting, <ul style="list-style-type: none"> <li>-The UiIRS bit is set to "1" (transmission completed): when data transmission from the UARTi transfer register is completed</li> </ul> </li> <li>• While receiving, <ul style="list-style-type: none"> <li>when data is transferred from the UARTi receive register to the UiRB register (reception completed)</li> </ul> </li> </ul>
Error Detection	<ul style="list-style-type: none"> <li>• Overrun error<sup>(1)</sup> <ul style="list-style-type: none"> <li>This error occurs when the eighth bit of the next data is received before reading the UiRB register</li> </ul> </li> <li>• Framing error <ul style="list-style-type: none"> <li>This error occurs when the number of the stop bit set is not detected</li> </ul> </li> <li>• Parity error <ul style="list-style-type: none"> <li>This error occurs when the number of "1" in parity bit and character bits differs from the number set</li> </ul> </li> <li>• Error sum flag <ul style="list-style-type: none"> <li>The SUM bit is set to "1" when an overrun error, framing error or parity error occurs</li> </ul> </li> </ul>

**NOTES:**

1. If an overrun error occurs, the UiRB register is indeterminate. The IR bit setting in the SiRIC register does not change to "1" (interrupt requested).
2. The CNT3 to CNT0 bits in the TCSPR register select no division (n=0) or divide-by-2<sup>n</sup> (n=1 to 15).

**Table 16.35 Register Settings in SIM Mode**

Register	Bit	Function
UiTB	7 to 0	Set transmit data
UiRB	7 to 0	Received data can be read
	OER, FER, PER, SUM	Error flags
UiBRG	7 to 0	Set bit rate
UiMR	SMD2 to SMD0	Set to "1012"
	CKDIR	Set to "0"
	STPS	Set to "0"
	PRY	Set to "1" for direct format or "0" for inverse format
	PRYE	Set to "1"
	IOPOL	Set to "0"
UiC0	CLK1, CLK0	Select count source for the UiBRG register
	CRS	Disabled because the CRD bit is set to "1"
	TXEPT	Transfer register empty flag
	CRD	Set to "1"
	NCH	Set to "1"
	CKPOL	Set to "0"
	UFORM	Set to "0" for direct format or "1" for inverse format
UiC1	TE	Set to "1" to enable data transmission
	TI	Transfer buffer empty flag
	RE	Set to "1" to enable data reception
	RI	Reception complete flag
	UiIRS	Set to "1"
	UiRRM	Set to "0"
	UiLCH	Set to "0" for direct format or "1" for inverse format
	UiERE	Set to "1"
UiSMR	7 to 0	Set to "0016"
UiSMR2	7 to 0	Set to "0016"
UiSMR3	7 to 0	Set to "0016"
UiSMR4	7 to 0	Set to "0016"

i=0 to 4

**Table 16.36 Pin Settings in SIM Mode (1)**

Port	Function	Setting	
		PS0 Register	PD6 Register
P62	RxD0 input	PS0_2=0	PD6_2=0
P63	TxD0 output	PS0_3=1	–
P66	RxD1 input	PS0_6=0	PD6_6=0
P67	TxD1 output	PS0_7=1	–

**Table 16.37 Pin Settings (2)**

Port	Function	Setting			
		PS1 Register	PSL1 Register	PSC Register	PD7 Register
P70 <sup>(1)</sup>	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	–
P71 <sup>(1)</sup>	RxD2 input	PS1_1=0	–	–	PD7_1=0

NOTE:

1. P70 and P71 are ports for the N-channel open drain output.

**Table 16.38 Pin Settings (3)**

Port	Function	Setting			
		PS3 Register <sup>(1)</sup>	PSL3 Register	PSC3 Register	PD9 Register <sup>(1)</sup>
P91	RxD3 input	PS3_1=0	–		PD9_1=0
P92	TxD3 output	PS3_2=1	PSL3_2=0		–
P96	TxD4 output	PS3_6=1	–	PSC3_6=0	–
P97	RxD4 input	PS3_7=0	–		PD9_7=0

NOTE:

1. Set the PD9 and PS3 registers immediately after the PRC2 bit in the PRCR register is set to "1" (write enabled). Do not generate an interrupt or a DMA transfer between the instruction to set to the PRC2 bit to "1" and the instruction to set the PD9 and PS3 registers.

Figure 16.29 shows an example of a SIM interface operation. Figure 16.30 shows an example of a SIM interface connection. Connect the TxDi pin to the RxDi pin for a pull-up.

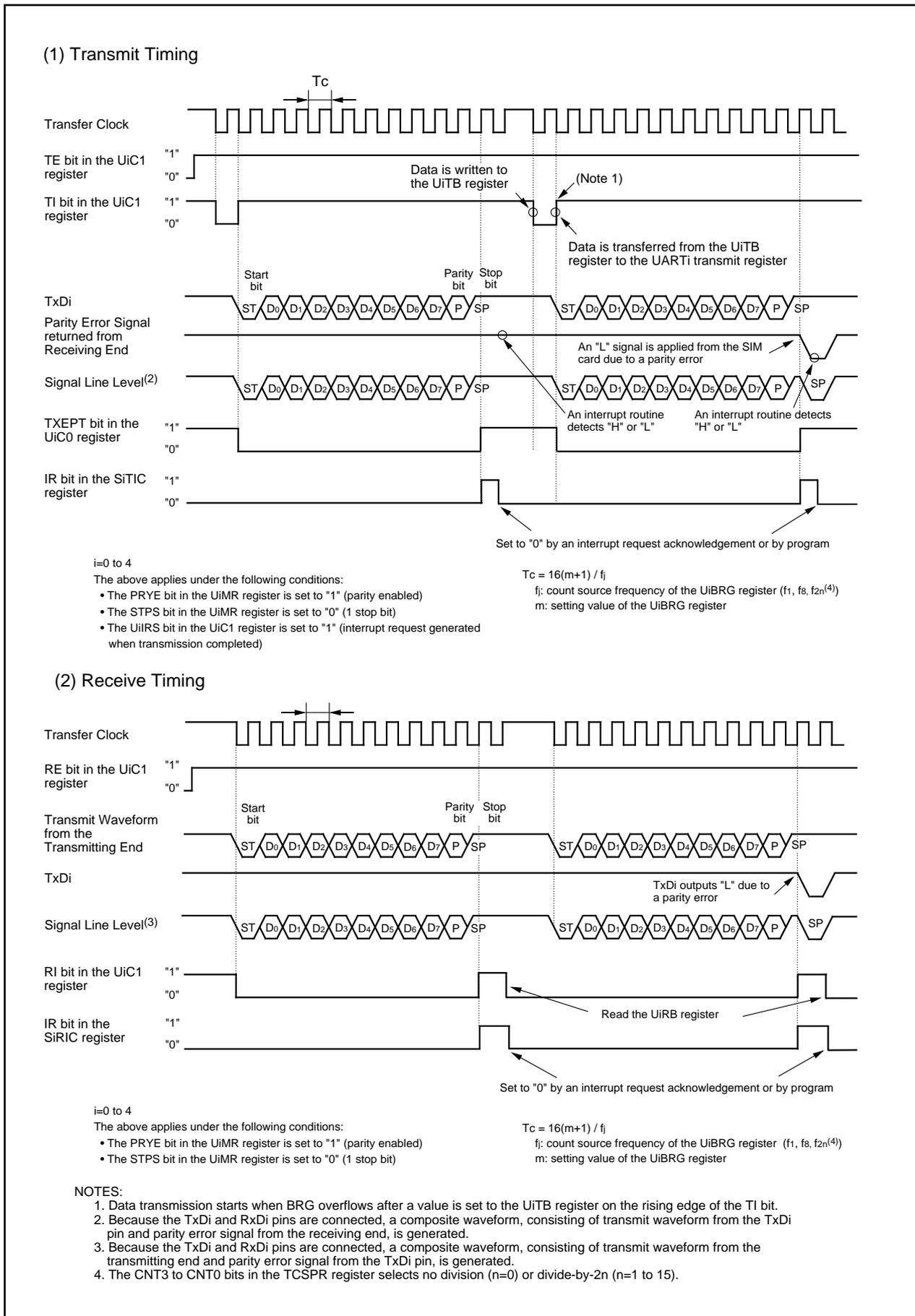


Figure 16.29 SIM Interface Operation

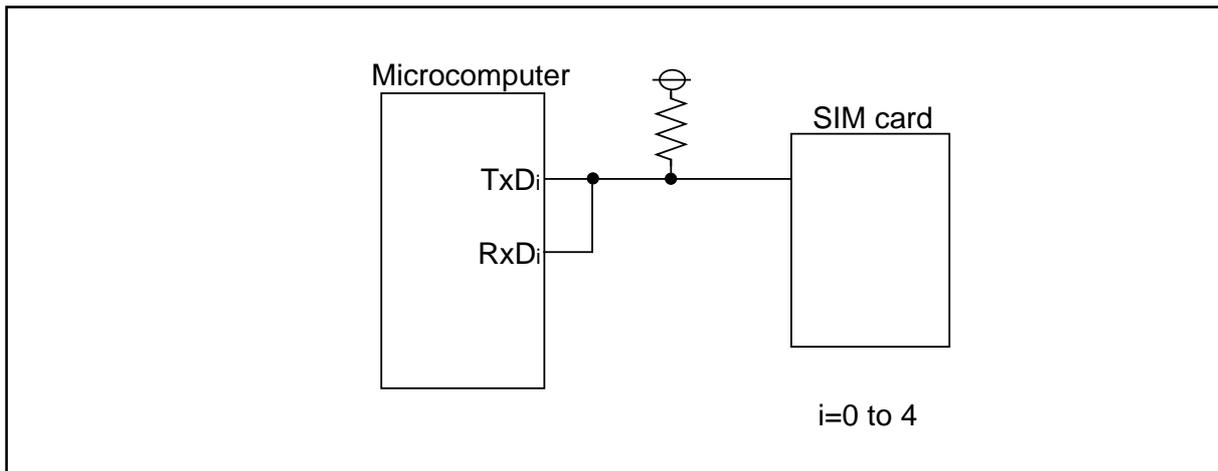


Figure 16.30 SIM Interface Connection

### 16.7.1 Parity Error Signal

#### 16.7.1.1 Parity Error Signal Output Function

When the UiERE bit in the UiC1 register (i=0 to 4) is set to "1" (output), the parity error signal output can be provided. The parity error signal output is provided when a parity error is detected upon receiving data. A low-level ("L") signal output is provided from the TxDi pin in the timing shown in Figure 16.31. When reading the UiRB register during a parity error output, the PER bit in the UiRB register is set to "0" (no error occurs) and a high-level ("H") signal output is again provided simultaneously.

#### 16.7.1.2 Parity Error Signal

To determine whether the parity error signal is output, the port that shares a pin with the RxDi pin is read by using an end-of-transmit interrupt routine.

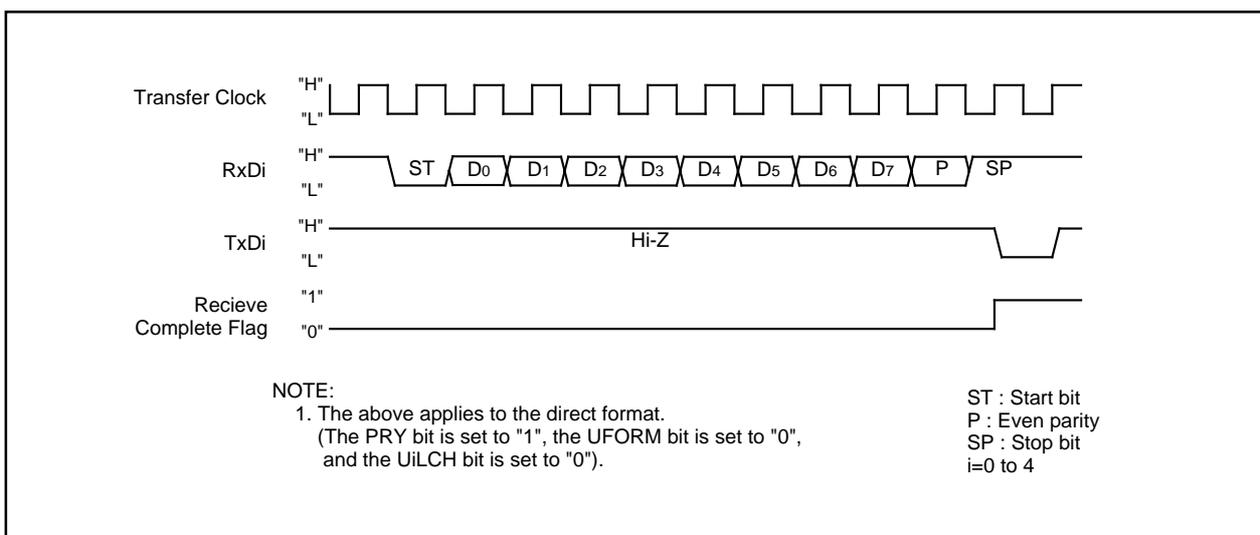


Figure 16.31 Parity Error Signal Output Timing (LSB First)

## 16.7.2 Format

### 16.7.2.1 Direct Format

Set the PRYE bit in the UiMR register (i=0 to 4) to "1" (parity enabled), the PRY bit to "1" (even parity), the UFORM bit in the UiC0 register to "0" (LSB first) and the UiLCH bit in the UiC1 register to "0" (not inverted). When data are transmitted, data set in the UiTB register are transmitted with the even-numbered parity, starting from D<sub>0</sub>. When data are received, received data are stored in the UiRB register, starting from D<sub>0</sub>. The even-numbered parity determines whether a parity error occurs.

### 16.7.2.2 Inverse Format

Set the PRYE bit to "1", the PRY bit to "0" (odd parity), the UFORM bit to "1" (MSB first) and the UiLCH bit to "1" (inverted). When data are transmitted, values set in the UiTB register are logically inverted and are transmitted with the odd-numbered parity, starting from D<sub>7</sub>. When data are received, received data are logically inverted to be stored in the UiRB register, starting from D<sub>7</sub>. The odd-numbered parity determines whether a parity error occurs.

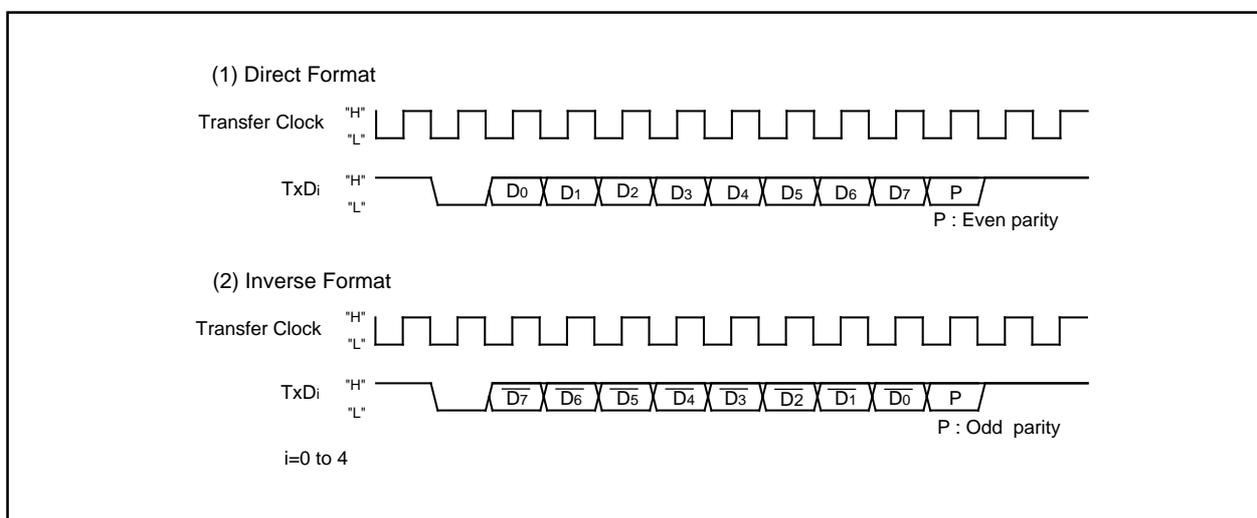


Figure 16.32 SIM Interface Format

## 17. A/D Converter

The A/D converter consists of one 10-bit successive approximation A/D converter with a capacitive coupling amplifier.

The result of an A/D conversion is stored into the A/D registers corresponding to selected pins. It is stored into the AD00 register only when DMAC operating mode is entered.

Table 17.1 lists specifications of the A/D converter. Figure 17.1 shows a block diagram of the A/D converter. Figures 17.2 to 17.6 show registers associated with the A/D converter.

### NOTE

This section is described in the 144-pin package only as an example.  
The AN150 to AN157 pins are not included in the 100-pin package.

**Table 17.1 A/D Converter Specifications**

Item	Specification
A/D Conversion Method	Successive approximation (with a capacitive coupling amplifier)
Analog Input Voltage <sup>(1)</sup>	0V to AVCC (VCC)
Operating Clock, $\phi_{AD}$ <sup>(2)</sup>	fAD, fAD/2, fAD/3, fAD/4, fAD/6, fAD/8
Resolution	8 bits or 10 bits
Operating Mode	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, repeat sweep mode 1, multi-port single sweep mode, multi-port repeat sweep mode 0
Analog Input Pins <sup>(3)</sup>	34 pins 8 pins each for AN (AN0 to AN7), AN0 (AN00 to AN07), AN2 (AN20 to AN27), AN15 (AN150 to AN157) 2 extended input pins (ANEX0 and ANEX1)
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>• Software trigger The ADST bit in the AD0CON0 register is set to "1" (A/D conversion started) by program</li> <li>• External trigger (re-trigger is enabled) When a falling edge is applied to the <math>\overline{ADTRG}</math> pin after the ADST bit is set to "1" by program</li> <li>• Hardware trigger (re-trigger is enabled) The timer B2 interrupt request of the three-phase motor control timer functions (after the ICTB2 counter completes counting) is generated after the ADST bit is set to "1" by program</li> </ul>
Conversion Rate Per Pin	<ul style="list-style-type: none"> <li>• Without the sample and hold function 8-bit resolution : 49 <math>\phi_{AD}</math> cycles 10-bit resolution : 59 <math>\phi_{AD}</math> cycles</li> <li>• With the sample and hold function 8-bit resolution : 28 <math>\phi_{AD}</math> cycles 10-bit resolution : 33 <math>\phi_{AD}</math> cycles</li> </ul>

## NOTES:

1. Analog input voltage is not affected by the sample and hold function status.
2.  $\phi_{AD}$  frequency must be 16 MHz or below when VCC=5V.  
Without the sample and hold function, the  $\phi_{AD}$  frequency is 250 kHz or above.  
With the sample and hold function, the  $\phi_{AD}$  frequency is 1 MHz or above.
3. AVCC = VREF = VCC, A/D input voltage (for AN0 to AN7, AN00 to AN07 and AN20 to AN27, AN150 to AN157, ANEX0 and ANEX1)  $\leq$  VCC.

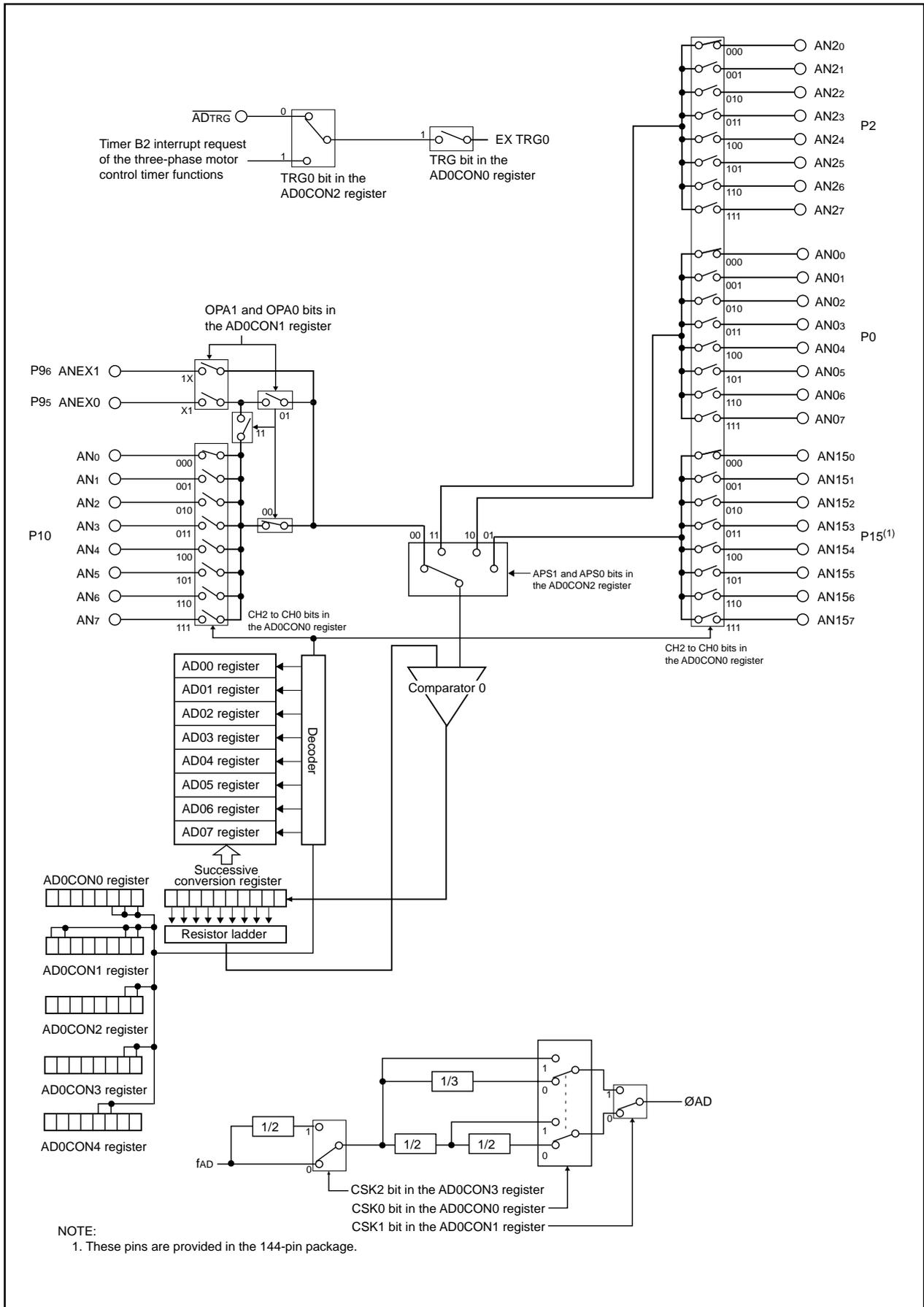


Figure 17.1 A/D Converter Block Diagram

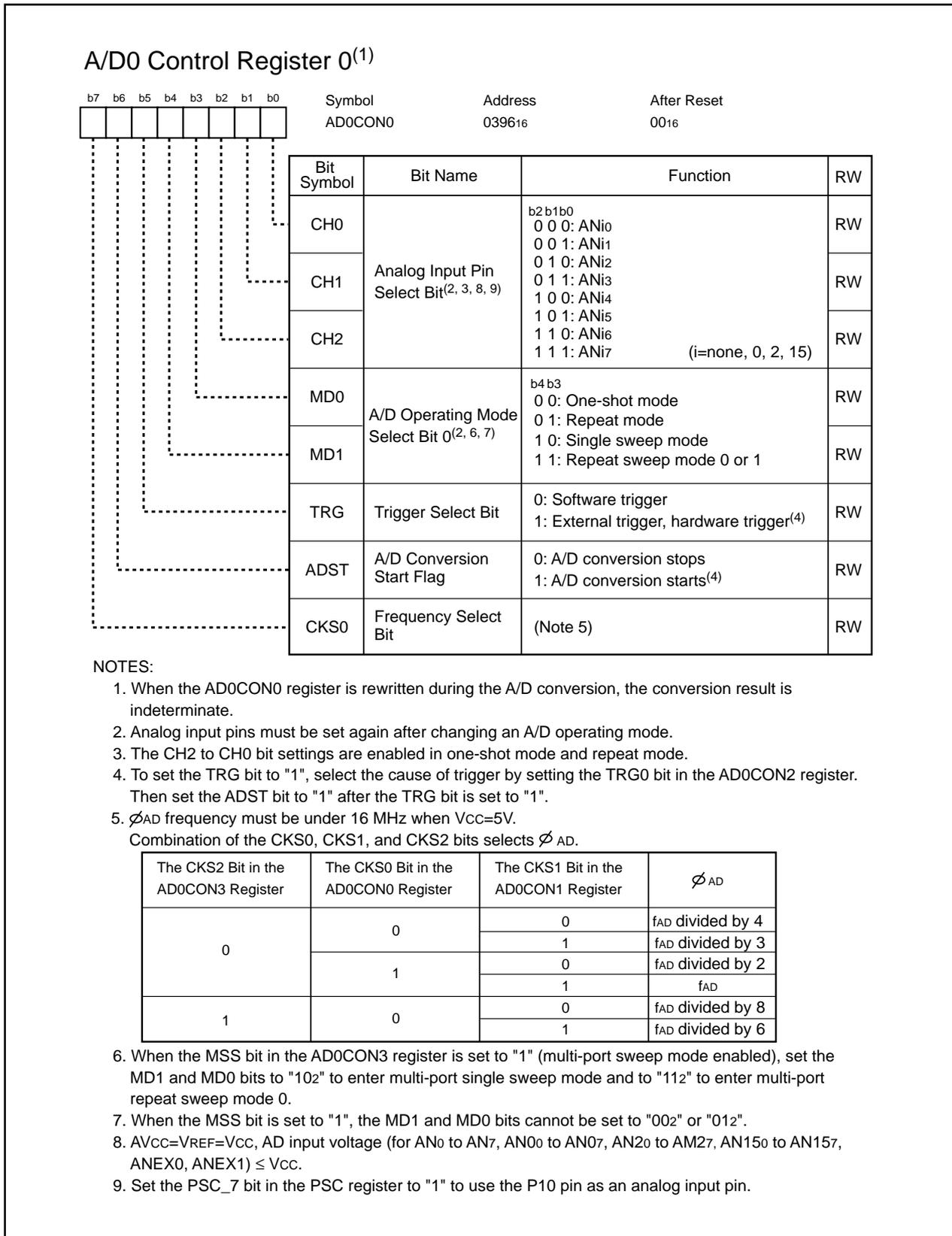


Figure 17.2 AD0CON0 Register

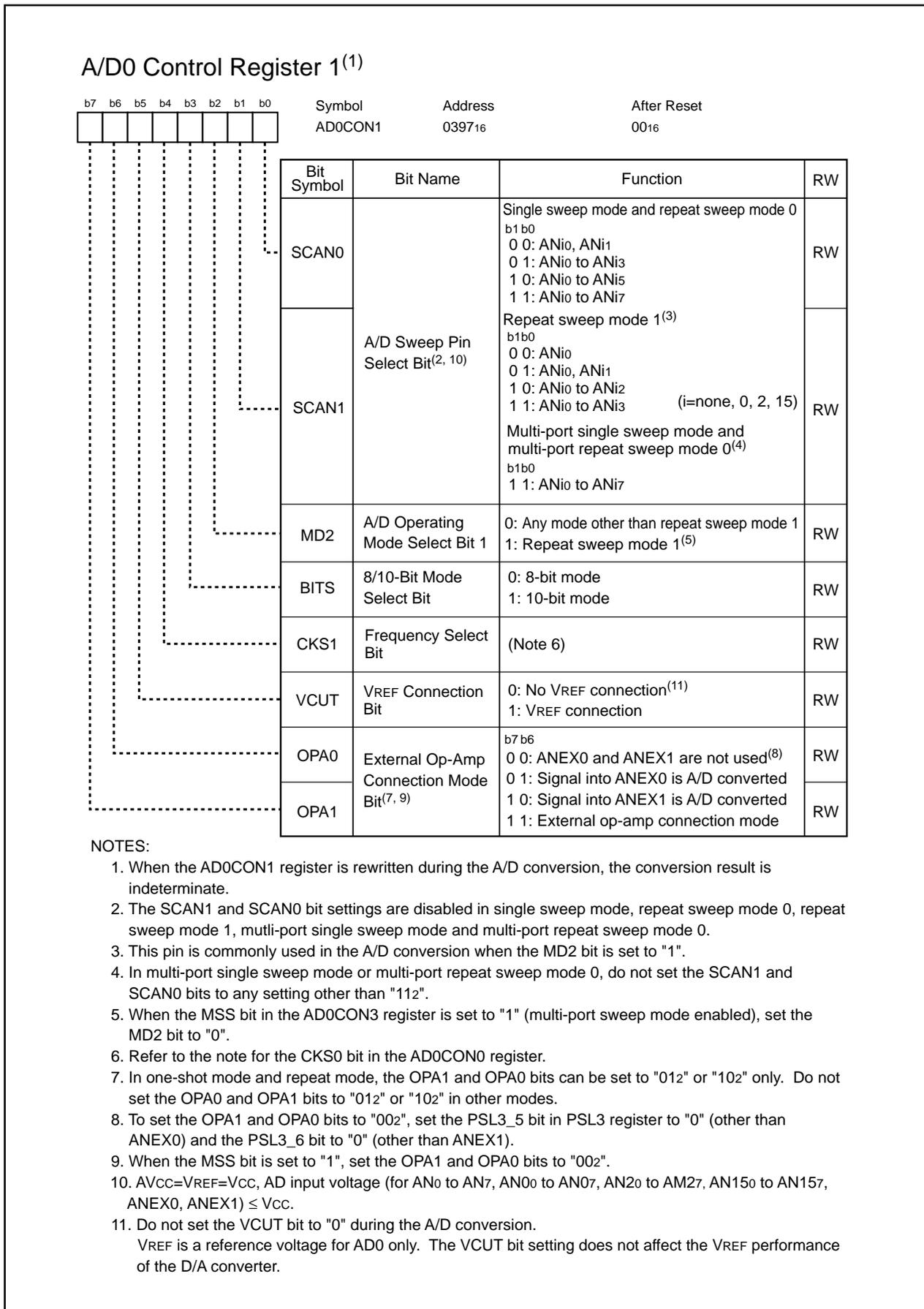


Figure 17.3 AD0CON1 Register

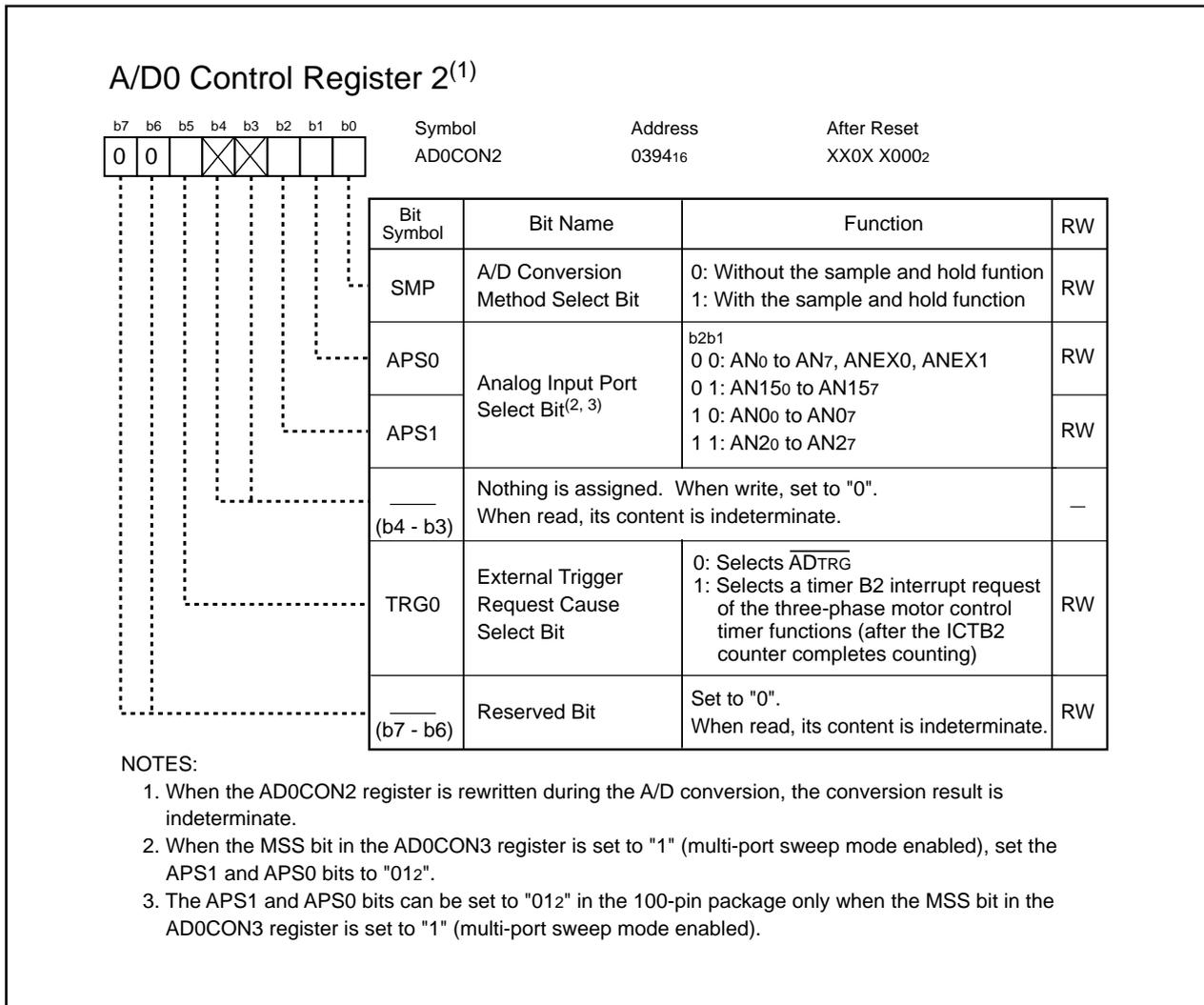


Figure 17.4 AD0CON2 Register

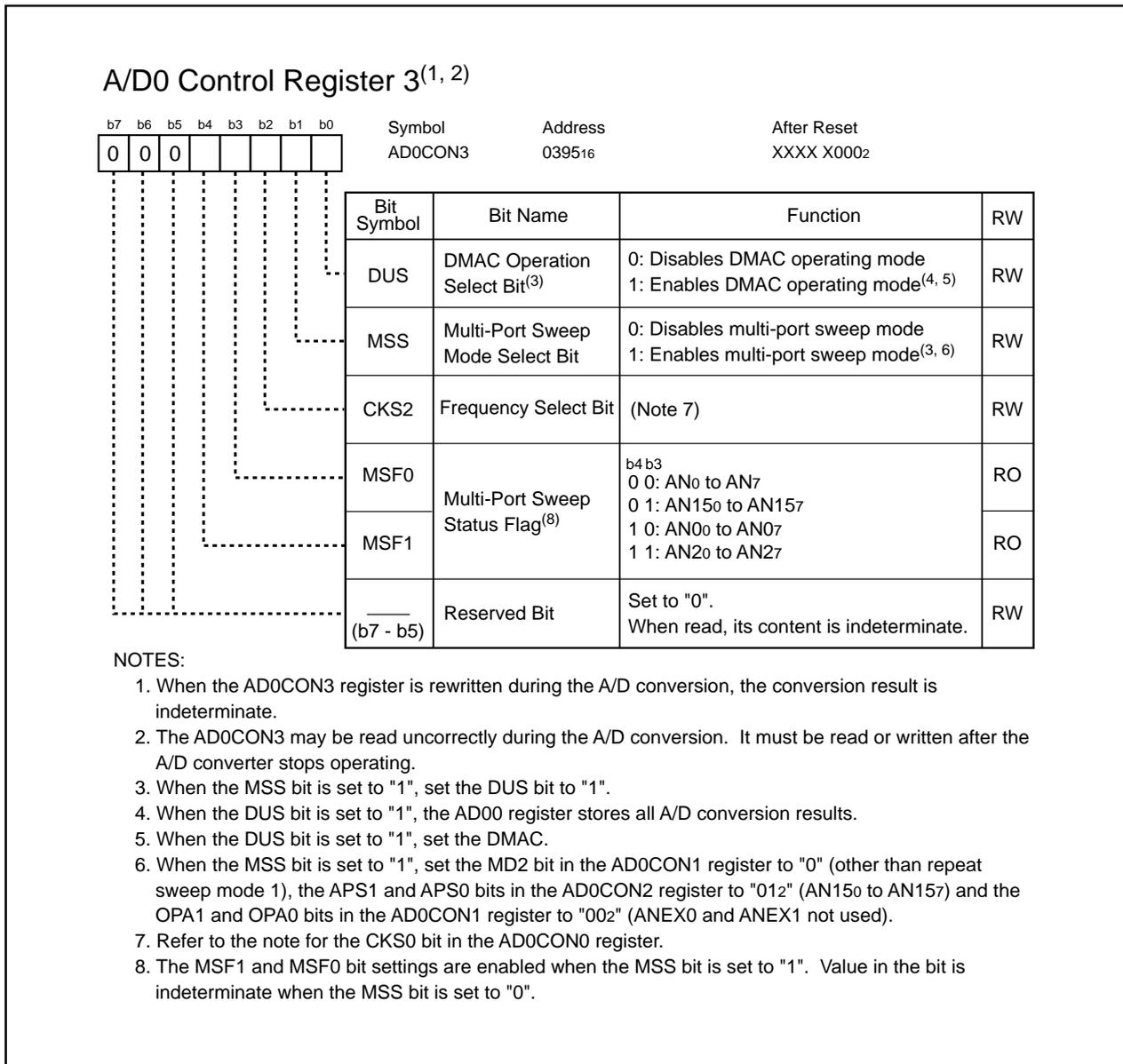
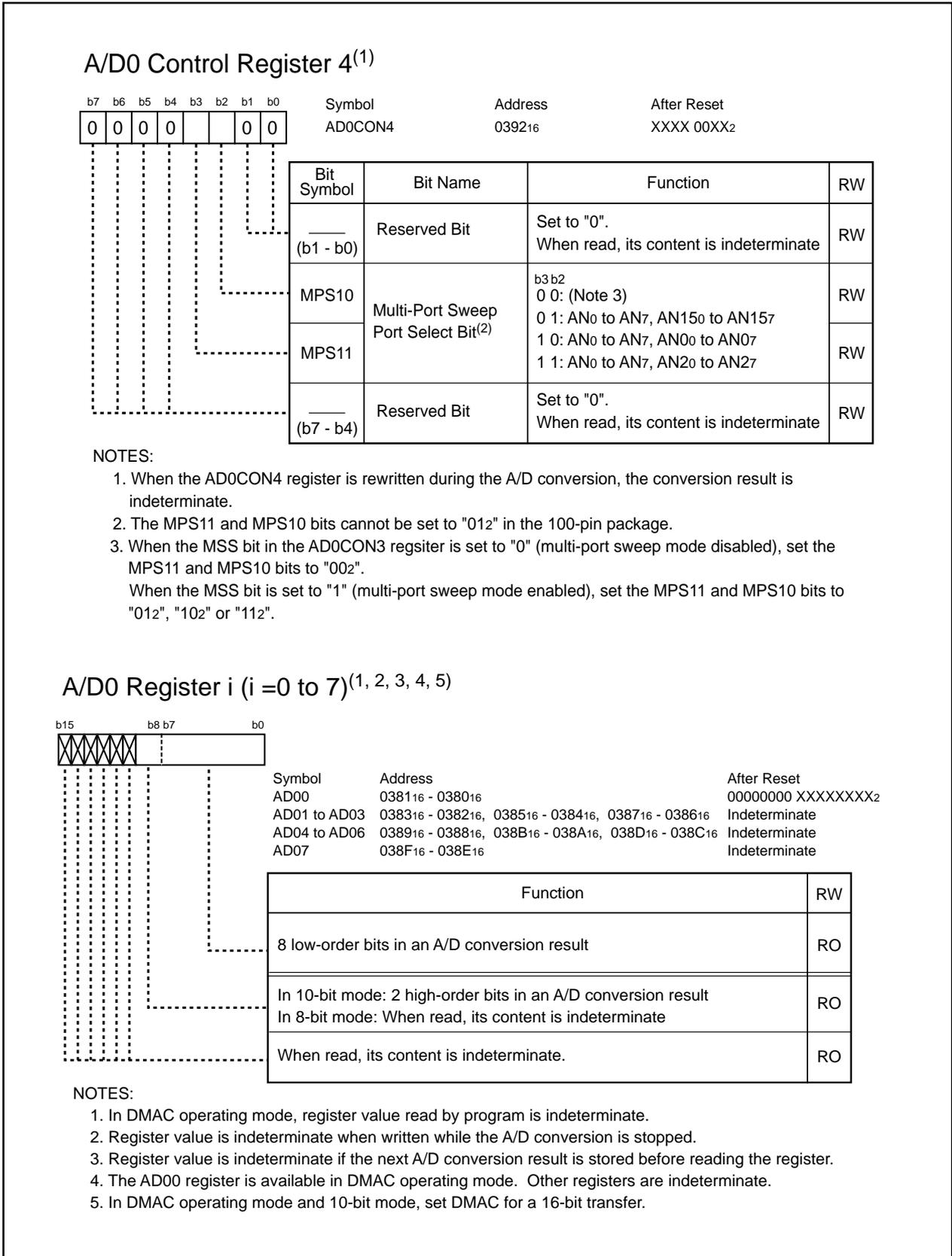


Figure 17.5 AD0CON3 Register



**Figure 17.6 AD0CON4 Register and AD00 to AD07 Registers**

## 17.1 Mode Description

### 17.1.1 One-shot Mode

In one-shot mode, analog voltage applied to a selected pin is converted to a digital code once. Table 17.2 lists specifications of one-shot mode.

**Table 17.2 One-shot Mode Specifications**

Item	Specification
Function	The CH2 to CH0 bits in the AD0CON0 register, the OPA1 and OPA0 bits in the AD0CON1 register and the APS1 and APS0 bits in the AD0CON2 register select a pin. Analog voltage applied to the pin is converted to a digital code once
Start Condition	<ul style="list-style-type: none"> <li>• When the TRG bit in the AD0CON0 register is set to "0" (software trigger), the ADST bit in the AD0CON0 register is set to "1" (A/D conversion starts) by program</li> <li>• When the TRG bit is set to "1" (external trigger, hardware trigger): <ul style="list-style-type: none"> <li>- a falling edge is applied to the <math>\overline{ADTRG}</math> pin after the ADST bit is set to "1" by program</li> <li>- The timer B2 interrupt request of three-phase motor control timer functions (after the ICTB2 register counter completes counting) is generated after the ADST bit is set to "1" by program</li> </ul> </li> </ul>
Stop Condition	<ul style="list-style-type: none"> <li>• A/D conversion is completed (the ADST bit is set to "0" when the software trigger is selected)</li> <li>• The ADST bit is set to "0" (A/D conversion stopped) by program</li> </ul>
Interrupt Request Generation Timing	A/D conversion is completed
Analog Voltage Input Pins	Select one pin from ANi0 to ANi7 (i=none, 0, 2, 15), ANEX0 or ANEX1
Reading of A/D Conversion Result	<ul style="list-style-type: none"> <li>• When the DUS bit in the AD0CON3 register is set to "0" (DMAC operating mode disabled), the microcomputer reads the AD0j register (j=0 to 7) corresponding to selected pin</li> <li>• When the DUS bit is set to "1" (DMAC operating mode enabled), do not read the AD00 register. A/D conversion result is stored in the AD00 register after the A/D conversion is completed. DMAC transfers the conversion result to any memory space. Refer to <b>12. DMAC</b> for DMAC settings</li> </ul>

### 17.1.2 Repeat Mode

In repeat mode, analog voltage applied to a selected pin is repeatedly converted to a digital code. Table 17.3 lists specifications of repeat mode.

**Table 17.3 Repeat Mode Specifications**

Item	Specification
Function	The CH2 to CH0 bits in the AD0CON0 register, the OPA1 and OPA0 bits in the AD0CON1 register and the APS1 and APS0 bits in the AD0CON2 register select a pin. Analog voltage applied to the pin is repeatedly converted to a digital code
Start Condition	Same as one-shot mode
Stop Condition	The ADST bit in the AD0CON0 register is set to "0" (A/D conversion stopped) by program
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>• When the DUS bit in the AD0CON3 register is set to "0" (DMAC operating mode disabled), no interrupt request is generated.</li> <li>• When DUS bit is set to "1" (DMAC operating mode enabled), an interrupt request is generated every time an A/D conversion is completed.</li> </ul>
Analog Voltage Input Pins	Select one pin from ANi0 to ANi7 (i=none, 0, 2, 15), ANEX0 or ANEX1
Reading of A/D Conversion Result	<ul style="list-style-type: none"> <li>• When the DUS bit is set to "0", the microcomputer reads the AD0j register (j=0 to 7) corresponding to the selected pin.</li> <li>• When DUS bit is set to "1", do not read the AD00 register. A/D conversion result is stored in the AD00 register after the A/D conversion is completed. DMAC transfers the conversion result to any memory space.</li> </ul> <p>Refer to <b>12. DMAC</b> for DMAC settings</p>

### 17.1.3 Single Sweep Mode

In single sweep mode, analog voltage that is applied to selected pins is converted one-by-one to a digital code. Table 17.4 lists specifications of single sweep mode.

**Table 17.4 Single Sweep Mode Specifications**

Item	Specification
Function	The SCAN1 and SCAN0 bits in the AD0CON1 register and the APS1 and APS0 bits in the AD0CON2 register select pins. Analog voltage applied to the pin is converted one-by-one to a digital code
Start Condition	Same as one-shot mode
Stop Condition	Same as one-shot mode
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>• When the DUS bit in the AD0CON3 register is set to "0" (DMAC operating mode disabled), an interrupt request is generated after a sweep is completed.</li> <li>• When DUS bit is set to "1" (DMAC operating mode enabled), an interrupt request is generated every time an A/D conversion is completed</li> </ul>
Analog Voltage Input Pins	Select from ANi0 and ANi1 (2 pins) (i=none, 0, 2, 15), ANi0 to ANi3 (4 pins), ANi0 to ANi5 (6 pins) or ANi0 to ANi7 (8 pins)
Reading of A/D Conversion Result	<ul style="list-style-type: none"> <li>• When the DUS bit is set to "0", the microcomputer reads the AD0j register corresponding to selected pins</li> <li>• When DUS bit is set to "1", do not read the AD00 register. A/D conversion result is stored in the AD00 register after the A/D conversion is completed. DMAC transfers the conversion result to any memory space. Refer to <b>12. DMAC</b> for DMAC settings</li> </ul>

### 17.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, analog voltage applied to selected pins is repeatedly converted to a digital code. Table 17.5 lists specifications of repeat sweep mode 0.

**Table 17.5 Repeat Sweep Mode 0 Specifications**

Item	Specification
Function	The SCAN1 and SCAN0 bits in the AD0CON1 register and the APS1 and APS0 bits in the AD0CON2 register select pins. Analog voltage applied to the pins is repeatedly converted to a digital code
Start Condition	Same as one-shot mode
Stop Condition	The ADST bit in the AD0CON0 register is set to "0" (A/D conversion stopped) by program
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>• When the DUS bit in the AD0CON3 register is set to "0" (DMAC operating mode disabled), no interrupt request is generated</li> <li>• When DUS bit is set to "1" (DMAC operating mode enabled), an interrupt request is generated every time an A/D conversion is completed</li> </ul>
Analog Voltage Input Pins	Select from ANi0 and ANi1 (2 pins) (i=none, 0, 2, 15), ANi0 to ANi3 (4 pins), ANi0 to ANi5 (6 pins) or ANi0 to ANi7 (8 pins)
Reading of A/D Conversion Result	<ul style="list-style-type: none"> <li>• When the DUS bit is set to "0", the microcomputer reads the AD0j register (j=0 to 7) corresponding to selected pins</li> <li>• When the DUS bit is set to "1", do not read the AD00 register. A/D conversion result is stored in the AD00 register after the A/D conversion is completed. DMAC transfers the conversion result to any memory space. Refer to <b>12. DMAC</b> for DMAC settings</li> </ul>

### 17.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltage selectively applied to eight pins is repeatedly converted to a digital code. Table 17.6 lists specifications of repeat sweep mode 1.

**Table 17.6 Repeat Sweep Mode 1 Specifications**

Item	Specification
Function	The SCAN1 and SCAN0 bits in the AD0CON1 register and the APS1 and APS0 bits in the AD0CON2 register select 8 pins. Analog voltage selectively applied to 8 pins is repeatedly converted to a digital code e.g., When ANi0 is selected (i =none, 0, 2, 15), analog voltage is converted to a digital code in the following order: ANi0 → ANi1 → ANi0 → ANi2 → ANi0 → ANi3 ..... etc.
Start Condition	Same as one-shot mode (Any trigger generated during an A/D conversion is invalid)
Stop Condition	The ADST bit is set to "0" (A/D conversion stopped) by program
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>• When the DUS bit in the AD0CON3 register is set to "0" (DMAC operating mode disabled), no interrupt request is generated</li> <li>• When DUS bit is set to "1" (DMAC operating mode enabled), an interrupt request is generated every time an A/D conversion is completed</li> </ul>
Analog Voltage Input Pins	ANi0 to ANi7 (8 pins)
Prioritized Pins	ANi0 (1 pin), ANi0 and ANi1 (2 pins), ANi0 to ANi2 (3 pins) or ANi0 to ANi3 (4 pins)
Reading of A/D Conversion Result	<ul style="list-style-type: none"> <li>• When the DUS bit is set to "0", the microcomputer reads the AD0j register (j=0 to 7) corresponding to selected pins</li> <li>• When the DUS bit is set to "1", do not read the AD00 register. A/D conversion result is stored in the AD00 register after the A/D conversion is completed. DMAC transfers the conversion result to any memory space. Refer to <b>12. DMAC</b> for DMAC settings</li> </ul>

### 17.1.6 Multi-Port Single Sweep Mode

In multi-port single sweep mode, analog voltage applied to 16 selected pins is converted one-by-one to a digital code. Set the DUS bit in the AD0CON3 register to "1" (DMAC operating mode enabled). Table 17.7 lists specifications of multi-port single sweep mode.

**Table 17.7 Multi-Port Single Sweep Mode Specifications**

Item	Specification
Function	The MPS11 and MPS10 bits in the AD0CON4 register select 16 pins. Analog voltage applied to 16 pins is converted one-by-one to a digital code in the following order: AN <sub>0</sub> to AN <sub>7</sub> → AN <sub>i0</sub> to AN <sub>i7</sub> (i=0, 2, 15) e.g., When the MPS11 and MPS10 bits are set to "102" (AN <sub>0</sub> to AN <sub>7</sub> , AN <sub>00</sub> to AN <sub>07</sub> ), analog voltage is converted to a digital code in the following order: AN <sub>0</sub> → AN <sub>1</sub> → AN <sub>2</sub> → AN <sub>3</sub> → AN <sub>4</sub> → AN <sub>5</sub> → AN <sub>6</sub> → AN <sub>7</sub> → AN <sub>00</sub> → AN <sub>01</sub> → ..... → AN <sub>06</sub> → AN <sub>07</sub>
Start Condition	Same as one-shot mode
Stop Condition	The ADST bit in the AD0CON0 register is set to "0" (A/D conversion stopped) by program
Interrupt Request Generation Timing	An interrupt request is generated every time A/D conversion is completed (Set the DUS bit to "1")
Analog Voltage Input Pins	Select from AN <sub>0</sub> to AN <sub>7</sub> → AN <sub>150</sub> to AN <sub>157</sub> , AN <sub>0</sub> to AN <sub>7</sub> → AN <sub>00</sub> to AN <sub>07</sub> or AN <sub>0</sub> to AN <sub>7</sub> → AN <sub>20</sub> to AN <sub>27</sub>
Reading of A/D Conversion Result	Do not read the AD00 register. A/D conversion result is stored in the AD00 register after the A/D conversion is completed. DMAC transfers the conversion result to any memory space. Refer to <b>12. DMAC</b> for DMAC settings (Set the DUS bit to "1")

### 17.1.7 Multi-Port Repeat Sweep Mode 0

In multi-port repeat sweep mode 0, analog voltage that is applied to 16 selected pins is repeatedly converted to a digital code. Set the DUS bit in the AD0CON3 register to "1" (DMAC operating mode enabled). Table 17.8 lists specifications of multi-port repeat sweep mode 0.

**Table 17.8 Multi-Port Repeat Sweep Mode 0 Specifications**

Item	Specification
Function	The MPS11 and MPS10 bits in the AD0CON4 register select 16 pins. Analog voltage applied to the 16 pins is repeatedly converted to a digital code in the following order: AN <sub>0</sub> to AN <sub>7</sub> → AN <sub>i0</sub> to AN <sub>i7</sub> (i=0, 2, 15) e.g., When the MPS11 and MPS10 bits are set to "102" (AN <sub>0</sub> to AN <sub>7</sub> , AN <sub>00</sub> to AN <sub>07</sub> ), analog voltage is repeatedly converted to a digital code in the following order: AN <sub>0</sub> → AN <sub>1</sub> → AN <sub>2</sub> → AN <sub>3</sub> → AN <sub>4</sub> → AN <sub>5</sub> → AN <sub>6</sub> → AN <sub>7</sub> → AN <sub>00</sub> → AN <sub>01</sub> → ..... → AN <sub>06</sub> → AN <sub>07</sub>
Start Condition	Same as one-shot mode
Stop Condition	The ADST bit is set to "0" (A/D conversion stopped) by program
Interrupt Request Generation Timing	An interrupt request is generated after each A/D conversion is completed (Set the DUS bit to "1")
Analog Voltage Input Pins	Selectable from AN <sub>0</sub> to AN <sub>7</sub> → AN <sub>150</sub> to AN <sub>157</sub> , AN <sub>0</sub> to AN <sub>7</sub> → AN <sub>00</sub> to AN <sub>07</sub> or AN <sub>0</sub> to AN <sub>7</sub> → AN <sub>20</sub> to AN <sub>27</sub>
Reading of A/D Conversion Result	Do not read the AD00 register. A/D conversion result is stored in the AD00 register after the A/D conversion is completed. DMAC transfers the conversion result to any memory space. Refer to <b>12. DMAC</b> for DMAC settings (Set the DUS bit to "1")

## 17.2 Functions

### 17.2.1 Resolution Select Function

The BITS bit in the AD0CON1 register determines the resolution. When the BITS bit is set to "1" (10-bit precision), the A/D conversion result is stored into bits 9 to 0 in the AD0j register (j = 0 to 7). When the BITS bit is set to "0" (8-bit precision), the A/D conversion result is stored into bits 7 to 0 in the AD0j register.

### 17.2.2 Sample and Hold Function

When the SMP bit in the AD0CON2 register is set to "1" (with the sample and hold function), A/D conversion rate per pin increases to  $28 \varnothing_{AD}$  cycles for 8-bit resolution and  $33 \varnothing_{AD}$  cycles for 10-bit resolution. The sample and hold function is available in all operating modes. Start the A/D conversion after selecting whether the sample and hold function is to be used or not.

### 17.2.3 Trigger Select Function

The TRG bit in the AD0CON0 register and the TRG0 bit in the AD0CON2 register select the trigger to start the A/D conversion. Table 17.9 lists settings of the trigger select function.

**Table 17.9 Trigger Select Function Settings**

Bit and Setting		Trigger
AD0CON0 Register	AD0CON2 Register	
TRG = 0	-	Software trigger The A/D0 starts the A/D conversion when the ADST bit in the AD0CON0 register is set to "1"
TRG = 1 <sup>(1)</sup>	TRG0 = 0	External trigger <sup>(2)</sup> Falling edge of a signal applied to $\overline{ADTRG}$
	TRG0 = 1	Hardware trigger <sup>(2)</sup> The timer B2 interrupt request of three-phase motor control timer functions (after the ICTB2 counter completes counting)

NOTES:

1. A/D0 starts the A/D conversion when the ADST bit is set to "1" (A/D conversion started) and a trigger is generated.
2. The A/D conversion is restarted if an external trigger or a hardware trigger is inserted during the A/D conversion. (The A/D conversion in process is aborted.)

### 17.2.4 DMAC Operating Mode

DMAC operating mode is available with all operating modes. When the A/D converter is in multi-port single sweep mode or multi-port repeat sweep mode 0, the DMAC operating mode must be used. When the DUS bit in the AD0CON3 register is set to "1" (DMAC operating mode enabled), all A/D conversion results are stored into the AD00 register. DMAC transfers data from the AD00 register to any memory space every time an A/D conversion is completed in each pin. 8-bit DMA transfer must be selected for 8-bit resolution and 16-bit DMA transfer for 10-bit resolution. Refer to **12. DMAC** for instructions.

### 17.2.5 Extended Analog Input Pins

In one-shot mode and repeat mode, the ANEX0 and ANEX1 pins can be used as analog input pins. The OPA1 and OPA0 bits in the AD0CON1 register select which pins to use as analog input pins. An A/D conversion result for the ANEX0 pin is stored into the AD00 register. The result for the ANEX1 pin is stored into the AD01 register, but is stored into the AD00 register when the DUS bit in the AD0CON3 register is set to "1" (DMAC operating mode enabled).

Set the APS1 and APS0 bits in the AD0CON2 register to "002" (AN<sub>0</sub> to AN<sub>7</sub>, ANEX0, ANEX1) and the MSS bit in the AD0CON3 register to "0" (multi-port sweep mode disabled).

### 17.2.6 External Operating Amplifier (Op-Amp) Connection Mode

In external op-amp connection mode, multiple analog voltage can be amplified by one external op-amp using extended analog input pins ANEX0 and ANEX1.

When the OPA1 and OPA0 bits in the AD0CON1 register are set to "112" (external op-amp connection), voltage applied to the AN<sub>0</sub> to AN<sub>7</sub> pins are output from ANEX0. Amplify this output signal by an external op-amp and apply it to ANEX1.

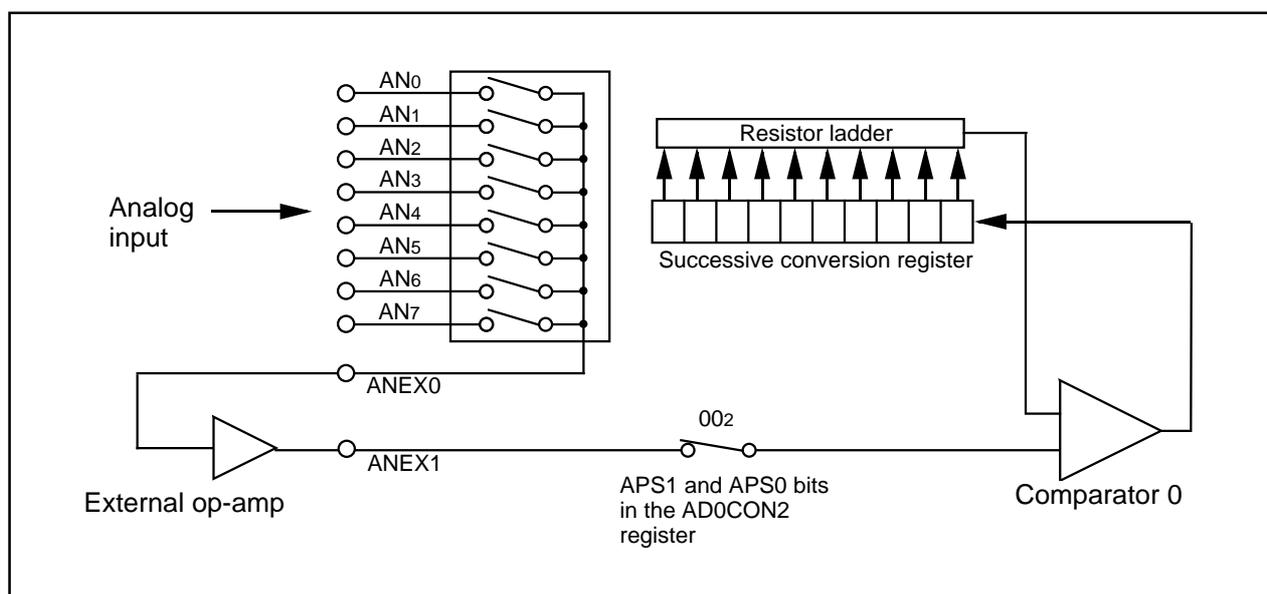
Analog voltage applied to ANEX1 is converted to a digital code and the A/D conversion result is stored into the corresponding AD0j register (j=0 to 7). A/D conversion rate varies depending on the response of the external op-amp. The ANEX0 pin cannot be connected to the ANEX1 pin directly.

Set the APS1 and APS0 bits in the AD0CON2 register to "002" (AN<sub>0</sub> to AN<sub>7</sub>, ANEX0, ANEX1).

Figure 17.7 shows an example of an external op-amp connection.

**Table 17.10 Extended Analog Input Pin Settings**

AD0CON1 Register		ANEX0 Function	ANEX1 Function
OPA1 Bit	OPA0 Bit		
0	0	Not used	Not used
0	1	P95 as an analog input	Not used
1	0	Not used	P96 as an analog input
1	1	Output to an external op-amp	Input from an external op-amp



**Figure 17.7 External Op-Amp Connection**

### 17.2.7 Power Consumption Reducing Function

When the A/D converter is not used, the VCUT bit in the AD0CON1 register isolates the resistor ladder of the A/D converter from the reference voltage input pin (VREF). Power consumption is reduced by shutting off any current flow into the resistor ladder from the VREF pin.

When using the A/D converter, set the VCUT bit to "1" (VREF connection) before setting the ADST bit in the AD0CON0 register to "1" (A/D conversion started). Do not set the ADST bit and VCUT bit to "1" simultaneously, nor set the VCUT bit to "0" (no VREF connection) during the A/D conversion. The VCUT bit does not affect the VREF performance of the D/A converter.

### 17.2.8 Output Impedance of Sensor Equivalent Circuit under A/D Conversion

For perfect A/D converter performance, complete internal capacitor (C) charging, shown in Figure 17.8, for the specified period (T) as sampling time. Output Impedance of the sensor equivalent circuit (R<sub>0</sub>) is determined by the following equations:

$$V_C = V_{IN} \left\{ 1 - e^{-\frac{1}{C(R_0 + R)} t} \right\}$$

$$\text{When } t = T, \quad V_C = V_{IN} - \frac{X}{Y} V_{IN} = V_{IN} \left( 1 - \frac{X}{Y} \right)$$

$$e^{-\frac{1}{C(R_0 + R)} T} = \frac{X}{Y}$$

$$-\frac{1}{C(R_0 + R)} T = \ln \frac{X}{Y}$$

$$R_0 = -\frac{T}{C \cdot \ln \frac{X}{Y}} - R$$

where:

V<sub>C</sub> = Voltage between pins

R = Internal resistance of the microcomputer

X = Precision (error) of the A/D converter

Y = Resolution of the A/D converter (1024 in 10-bit mode, and 256 in 8-bit mode)

Figure 17.8 shows analog input pin and external sensor equivalent circuit. The impedance (R<sub>0</sub>) can be obtained if the voltage between pins (V<sub>C</sub>) changes from 0 to V<sub>IN</sub> - (0.1/1024) V<sub>IN</sub> in the time (T), when the difference between V<sub>IN</sub> and V<sub>C</sub> becomes 0.1LSB.

(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 ØAD = 10 MHz, T = 0.3 µs in the A/D conversion mode with the sample and hold function. Output impedance (R<sub>0</sub>) for sufficiently charging capacitor (C) in the time (T) is determined by the following equation:

Using T = 0.3 µs, R = 7.8 kΩ, C = 1.5 pF, X = 0.1, Y = 1024,

$$R_0 = -\frac{0.3 \times 10^{-6}}{1.5 \times 10^{-12} \cdot \ln \frac{0.1}{1024}} - 7.8 \times 10^3 = 13.9 \times 10^3$$

Thus, the allowable output impedance of the sensor equivalent circuit, making the precision (error) 0.1LSB or less, is approximately 13.9 kΩ maximum.

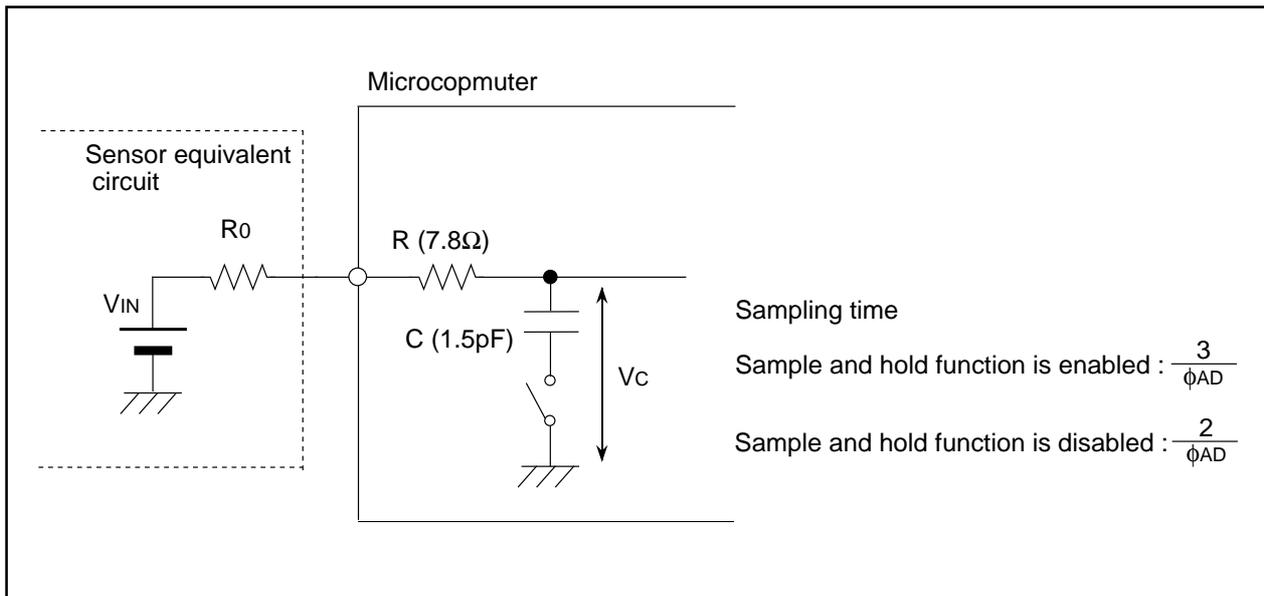


Figure 17.8 Analog Input Pin and External Sensor Equivalent Circuit

## 18. D/A Converter

The D/A converter consists of two separate 8-bit R-2R ladder D/A converters.

Digital code is converted to an analog voltage when a value is written to the corresponding DAi registers (i=0,1). The DAiE bit in the DACON register determines whether the D/A conversion result output is provided or not. Set the DAiE bit to "1" (output enabled) to disable a pull-up of a corresponding port.

Output analog voltage ( $V$ ) is calculated from value  $n$  ( $n$ =decimal) set in the DAi register.

$$V = \frac{V_{REF} \times n}{256} \quad (n = 0 \text{ to } 255)$$

$V_{REF}$  : reference voltage (not related to VCUT bit setting in the AD0CON1 register)

Table 18.1 lists specifications of the D/A converter. Table 18.2 lists the DA0 and DA1 pin settings. Figure 18.1 shows a block diagram of the D/A converter. Figure 18.2 shows the D/A control register. Figure 18.3 shows a D/A converter equivalent circuit.

When the D/A converter is not used, set the DAi register to "0016" and the DAiE bit to "0" (output disabled).

**Table 18.1 D/A Converter Specifications**

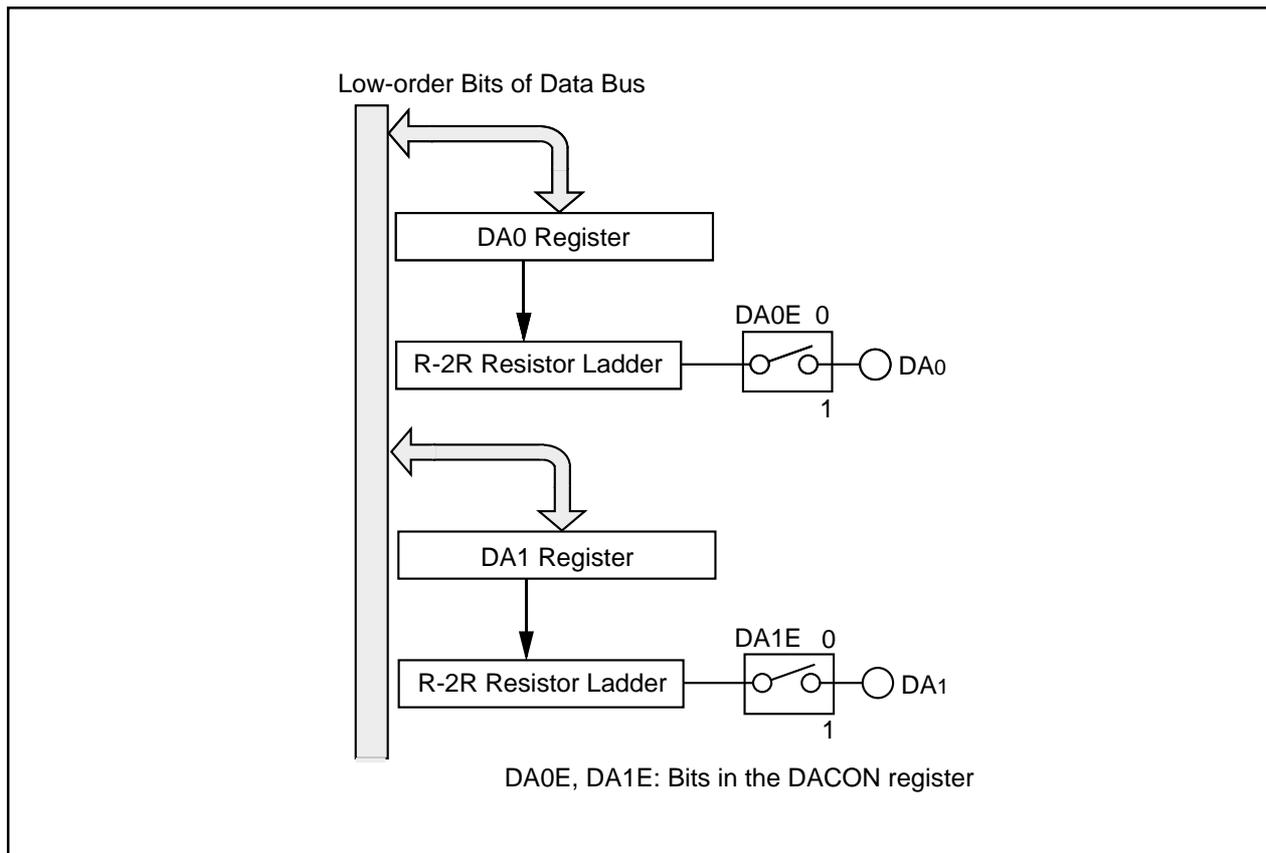
Item	Specification
D/A Conversion Method	R-2R
Resolution	8 bits
Analog Output Pin	2 channels

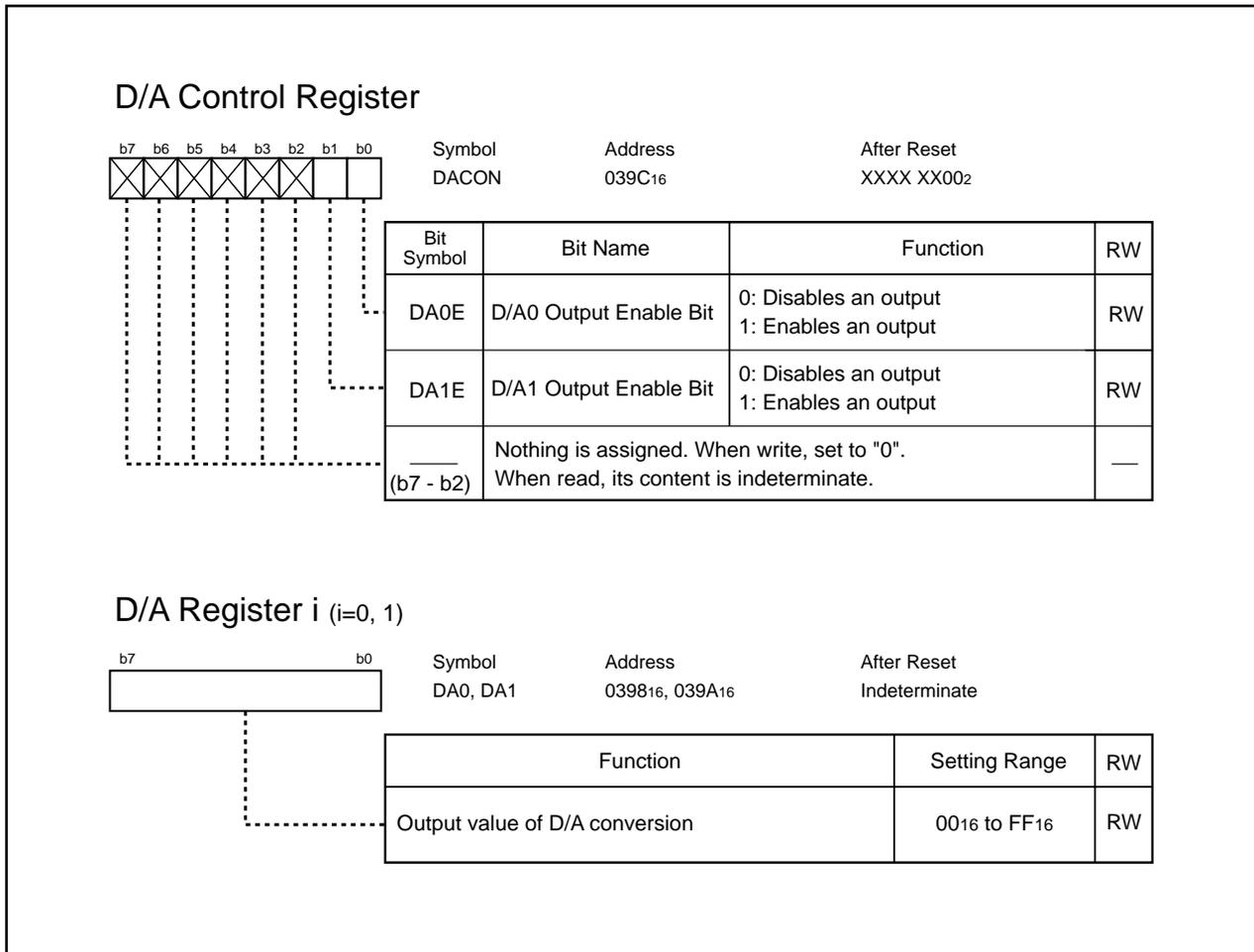
**Table 18.2 Pin Settings**

Port	Function	Bit and Setting		
		PD9 Register <sup>(1)</sup>	PS3 Register <sup>(1)</sup>	PSL3 Register
P93	DA0 output	PD9_3=0	PS3_3=0	PSL3_3=1
P94	DA1 output	PD9_4=0	PS3_4=0	PSL3_4=1

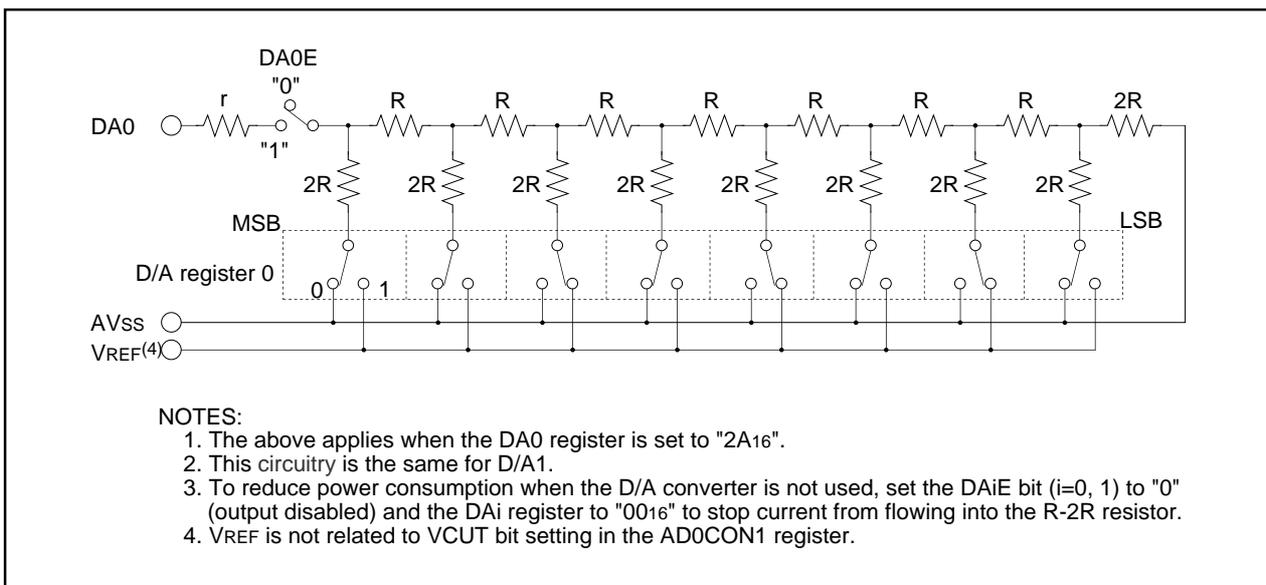
NOTE:

1. Set the PD9 and PS3 registers immediately after the PRC2 bit in the PRCR register is set to "1" (write enabled). Do not generate an interrupt or a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to set the PD9 and PS3 registers.

**Figure 18.1 D/A Converter**



**Figure 18.2 DACON Register, DA0 and DA1 Registers**



**Figure 18.3 D/A Converter Equivalent Circuit**

# 19. CRC Calculation

The CRC (Cyclic Redundancy Check) calculation detects an error in data blocks. A generator polynomial of CRC\_CCITT ( $X^{16} + X^{12} + X^5 + 1$ ) generates CRC code.

The CRC code is a 16-bit code generated for a block of data of desired length. This block of data is in 8-bit units. The CRC code is set in the CRCD register every time one-byte data is transferred to the CRCIN register after a default value is written to the CRCD register. CRC code generation for one-byte data is completed in two cycles.

Figure 19.1 shows a block diagram of a CRC circuit. Figure 19.2 shows CRC-associated registers. Figure 19.3 shows an example of the CRC calculation.

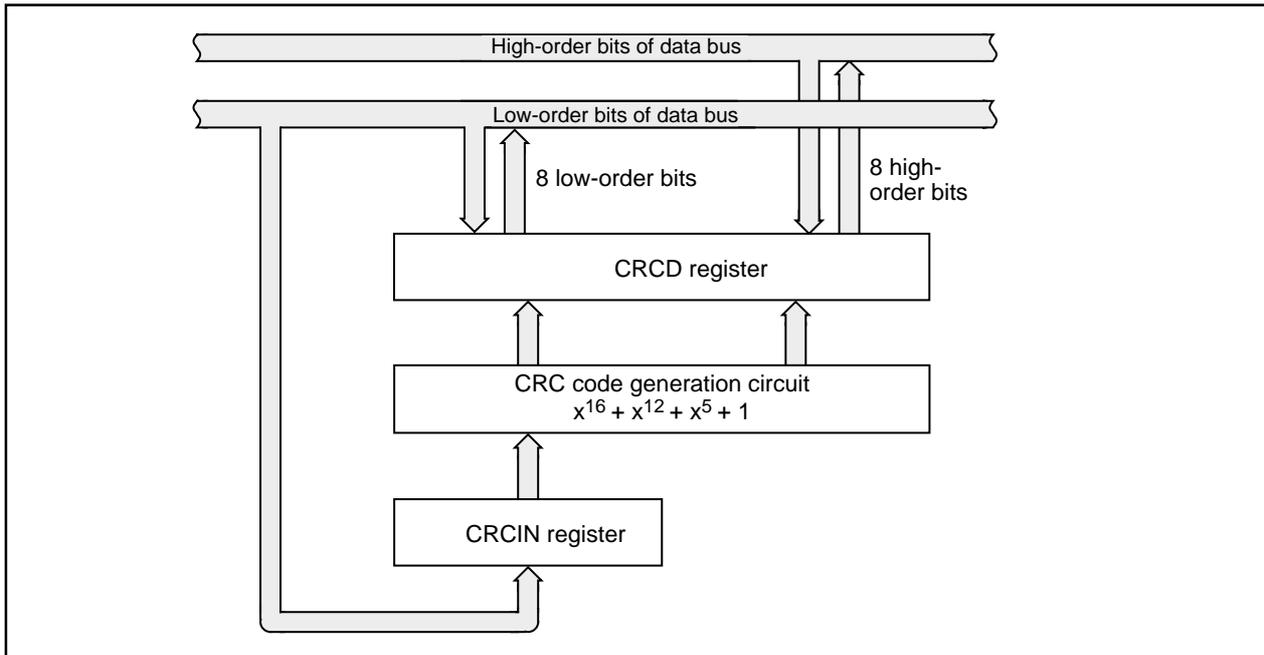


Figure 19.1 CRC Calculation Block Diagram

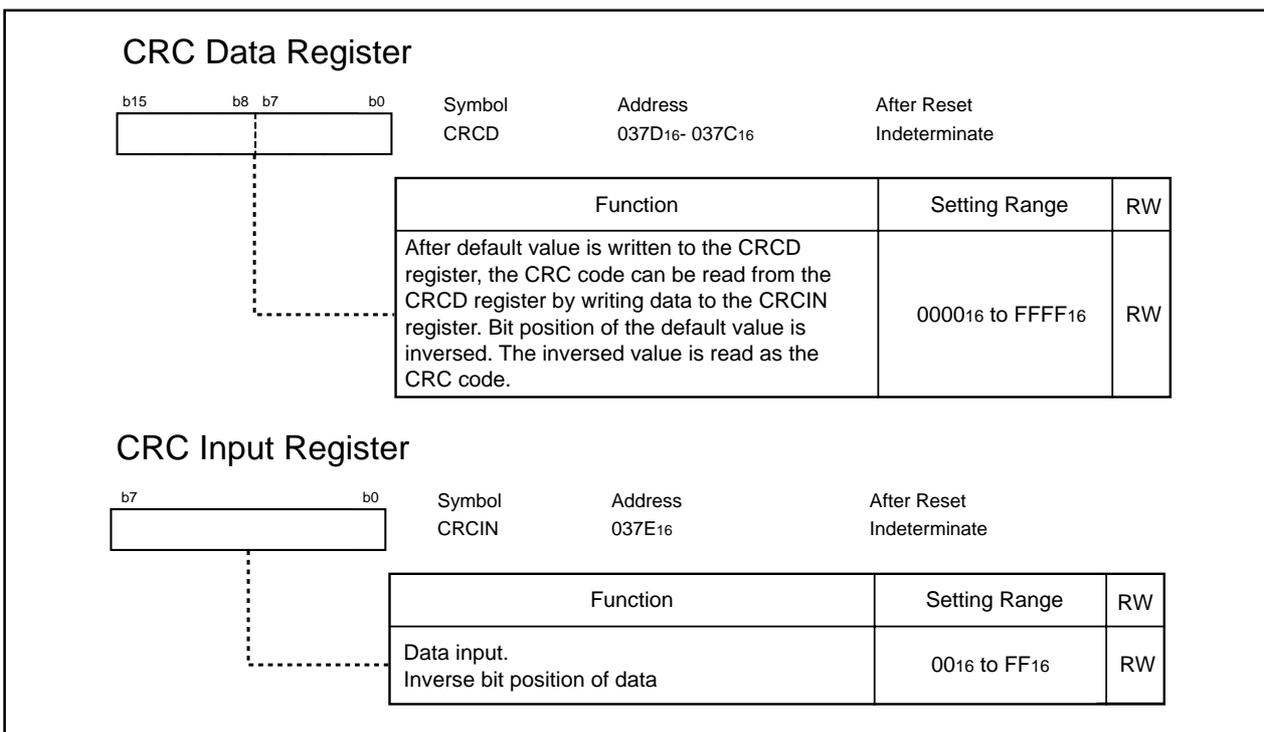


Figure 19.2 CRCD Register and CRCIN Register

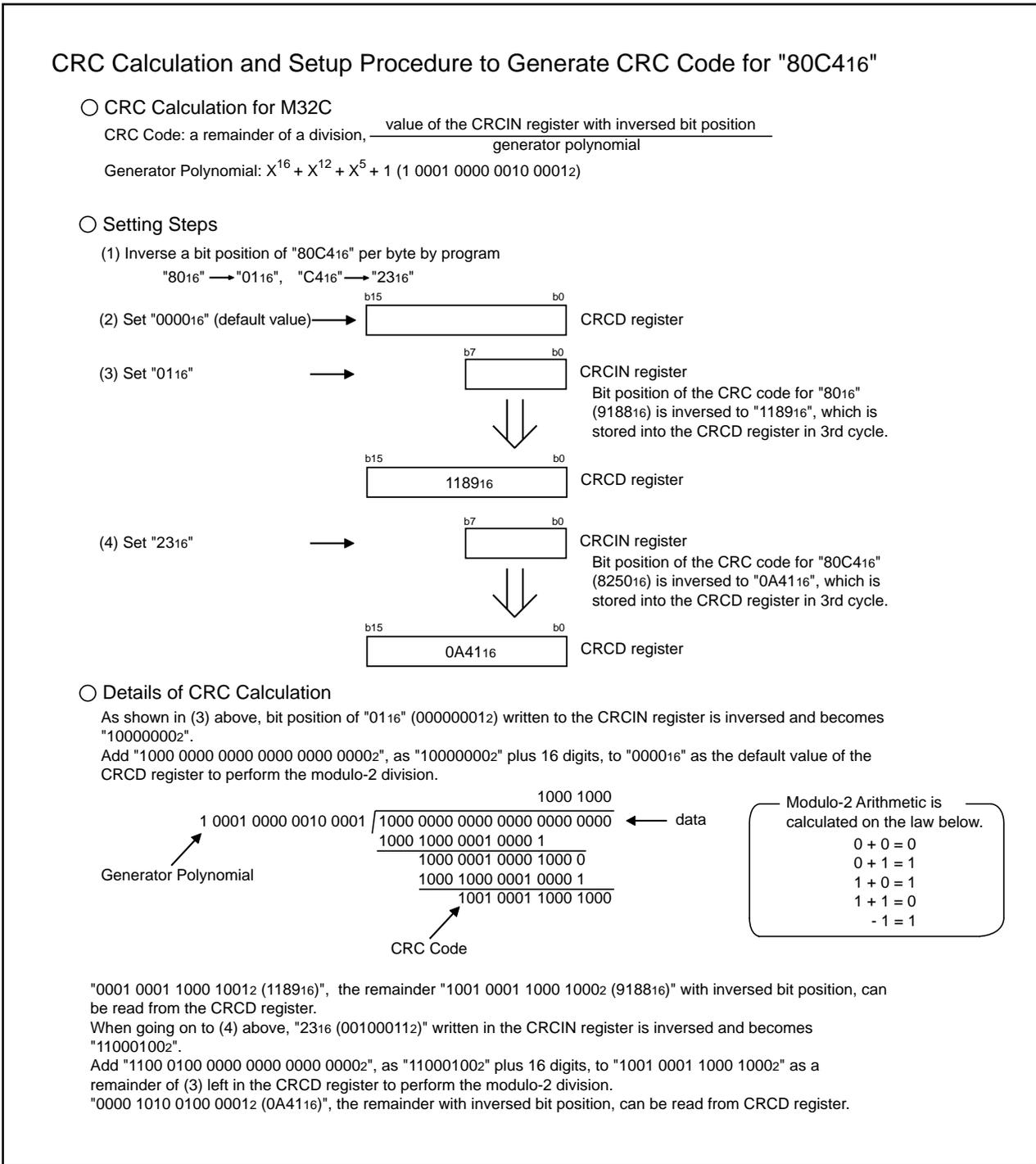
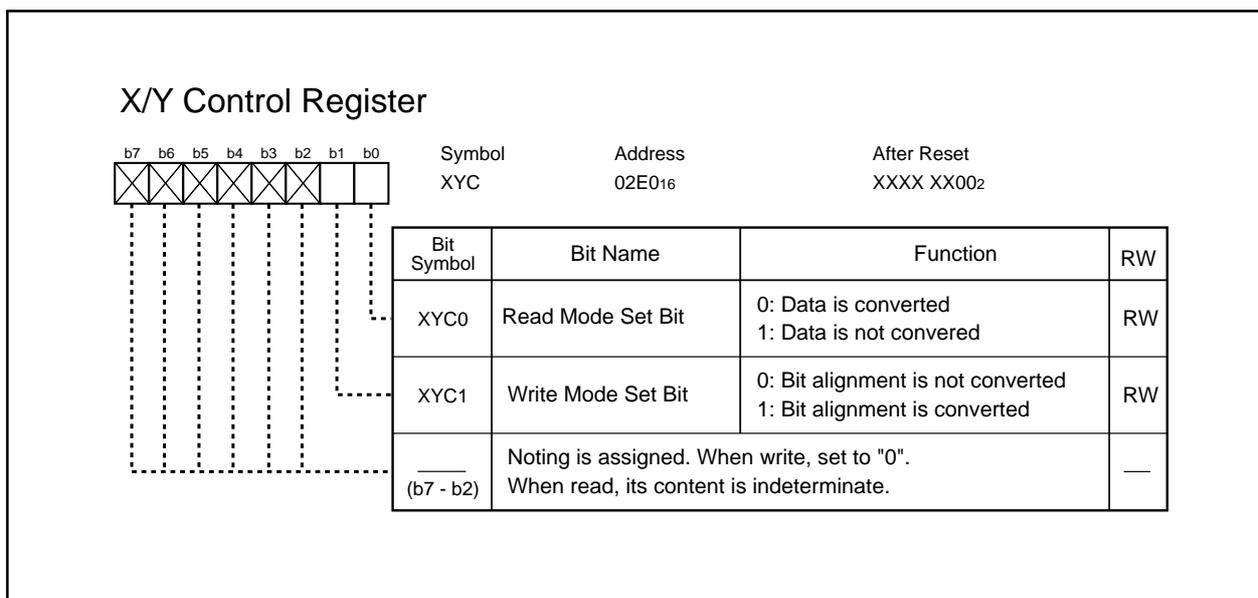


Figure 19.3 CRC Calculation

## 20. X/Y Conversion

The X/Y conversion rotates a 16 x 16 matrix data by 90 degrees and inverses high-order bits and low-order bits of a 16-bit data. Figure 20.1 shows the XYC register.

The 16-bit XiR register (i=0 to 15) and 16-bit YjR register (j=0 to 15) are allocated to the same address. The XiR register is a write-only register, while the YjR register is a read-only register. Access the XiR and YjR registers from an even address in 16-bit units. Performance cannot be guaranteed if the XiR and YiR registers are accessed in 8-bit units.



**Figure 20.1** XYC Register

The XYC0 bit in the XYC register determines how to read the YjR register.

By reading the YjR register when the XYC0 bit is set to "0" (data conversion), bit j in the X0R to X15R registers can be read simultaneously.

For example, bit 0 in the X0R register can be read if reading bit 0 in the Y0R register, bit 0 in the X1R register if reading bit 1 in the Y0R register..., bit 0 in the X14R register if reading bit 14 in the Y0R register and bit 0 in the X15R register if reading bit 15 in the Y0R register.

Figure 20.2 shows the conversion table when the XYC0 bit is set to "0". Figure 20.3 shows an example of the X/Y conversion.

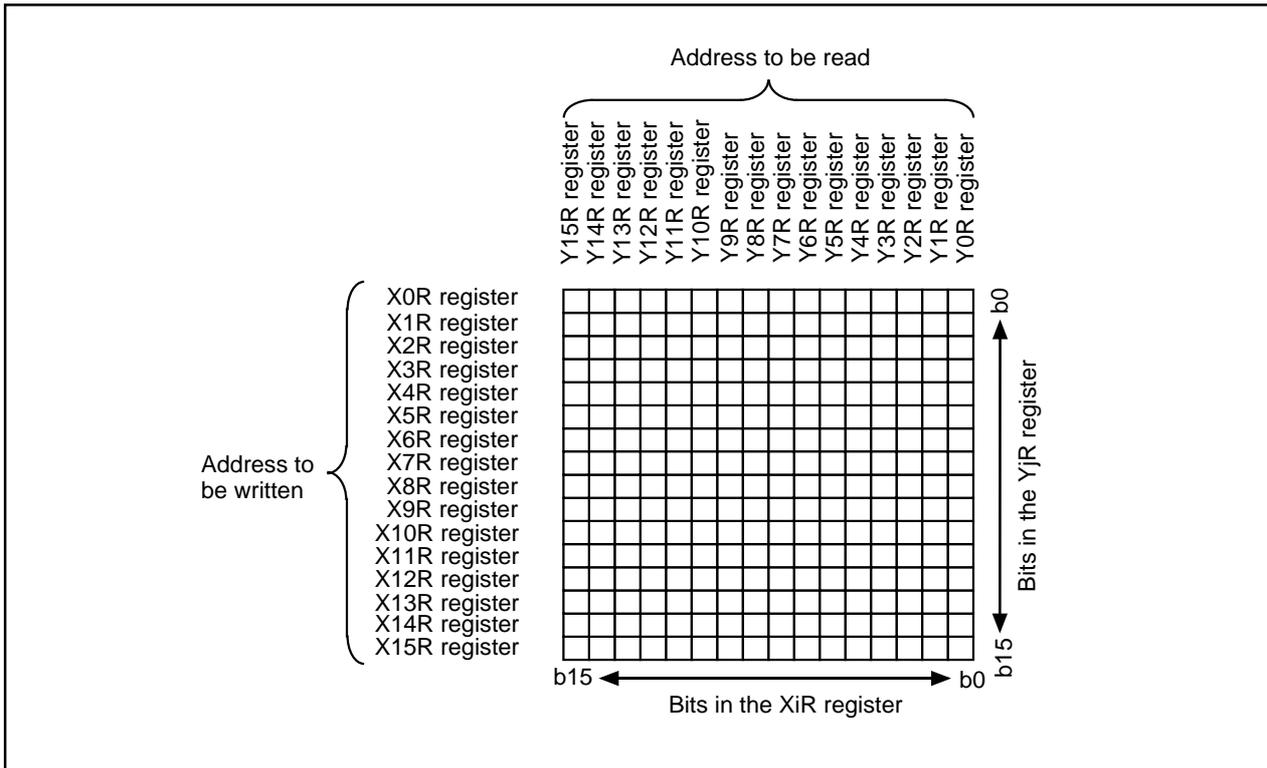


Figure 20.2 Conversion Table when Setting the XYC0 Bit to "0"

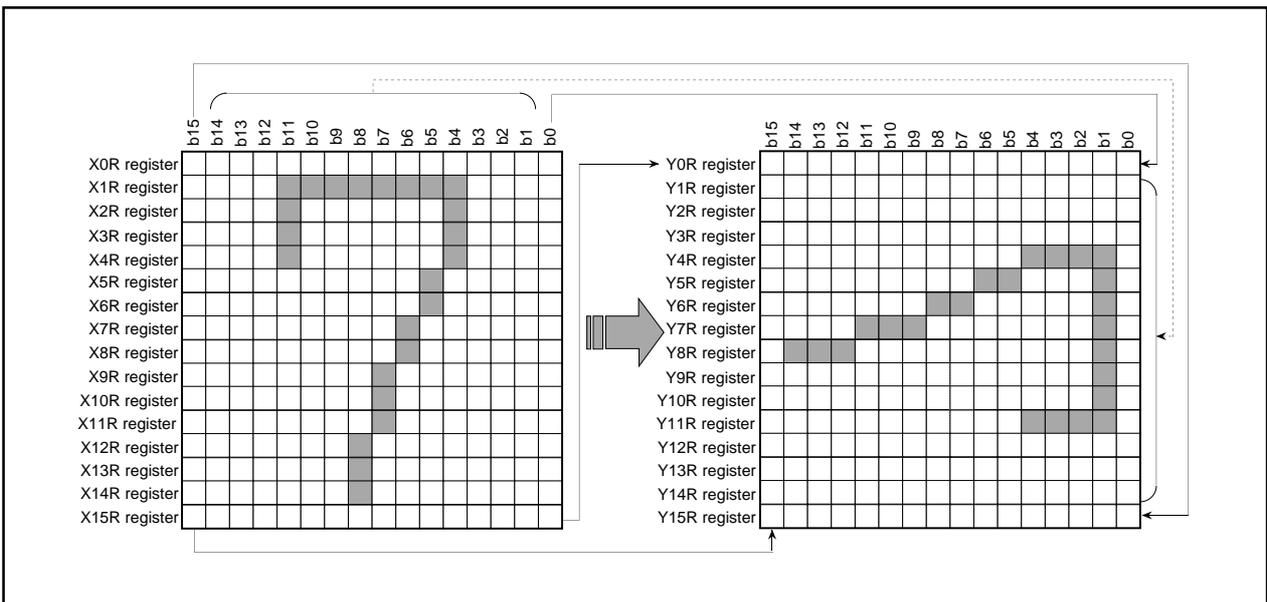


Figure 20.3 X/Y Conversion

By reading the YjR register when the XYC0 bit in the XYC register is set to "1" (no data conversion), the value written to the XiR register can be read directly. Figure 20.4 shows the conversion table when the XYC0 bit is set to "1."

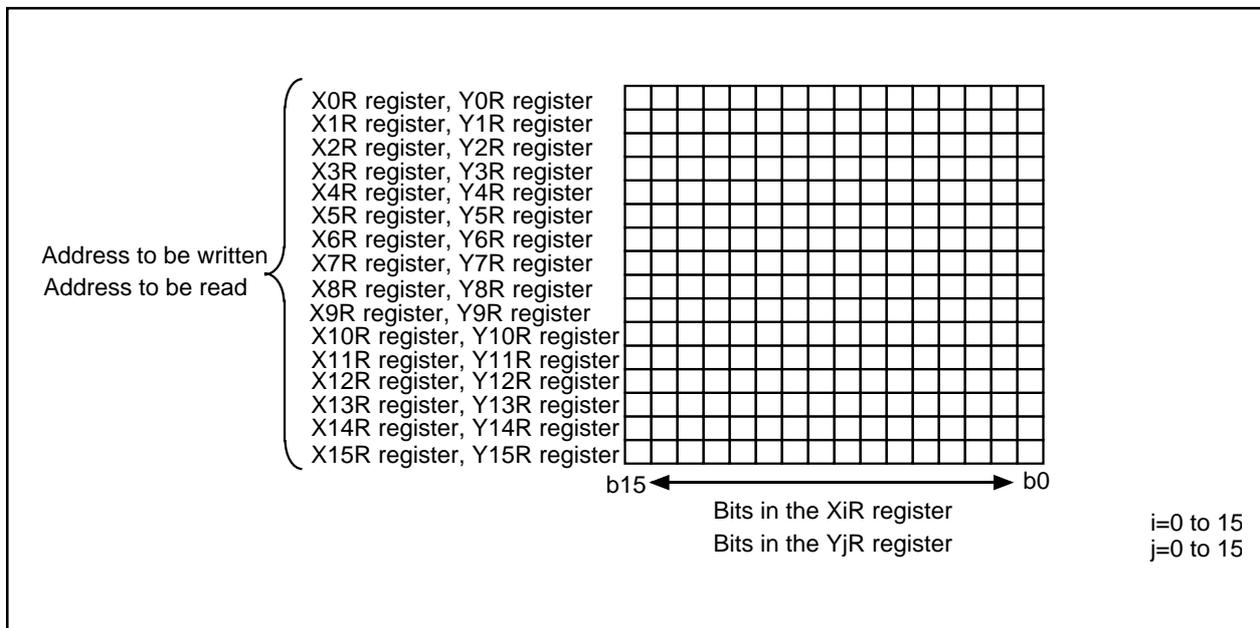


Figure 20.4 Conversion Table when Setting the XYC0 Bit to "1"

The XYC1 bit in the XYC register selects bit alignment of the value in the XiR register.

By writing to the XiR register while the XYC1 bit is set to "0" (no bit alignment conversion), bit alignment is written as is. By writing to the XiR register while the XYC1 bit is set to "1" (bit sequence replaced), bit alignment is written inversed.

Figure 20.5 shows the conversion table when the XYC1 bit is set to "1".

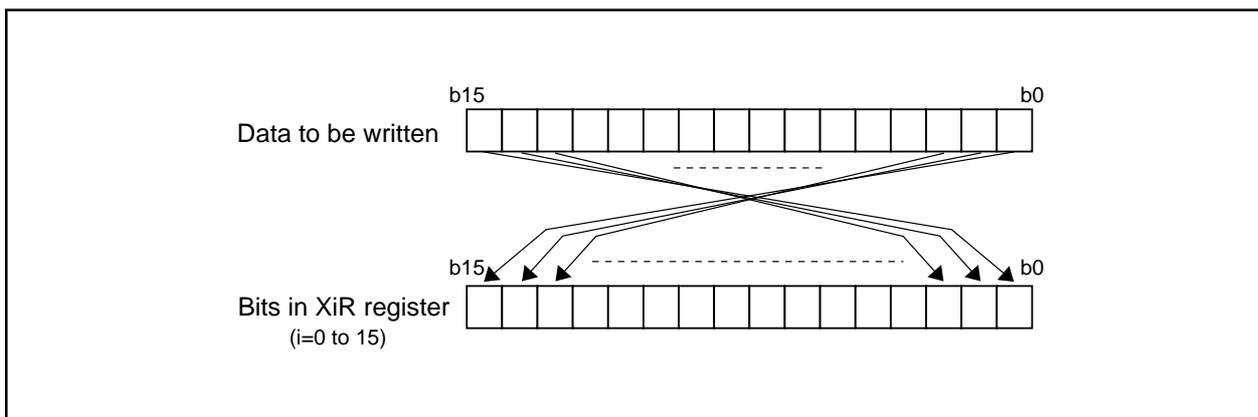


Figure 20.5 Conversion Table when Setting the XYC1 Bit to "1"

## 21. Intelligent I/O

The intelligent I/O is a multifunctional I/O port for time measurement, waveform generating, clock synchronous serial I/O, clock asynchronous serial I/O (UART), HDLC data processing and more.

The intelligent I/O has one 16-bit base timer for free-running operation, eight 16-bit registers for time measurement and waveform generating and two sets of two 8-bit shift registers for communications.

Table 21.1 lists functions and channels of the intelligent I/O.

**Table 21.1 Intelligent I/O Functions and Channels**

Function	Description	
Time Measurement <sup>(1)</sup>	8 channels	
Digital Filter	8 channels	
Trigger Input Prescaler	2 channels (channel 6 and channel 7)	
Trigger Input Gate	2 channels (channel 6 and channel 7)	
Waveform Generating <sup>(1)</sup>	8 channels	
Single-Phase Waveform Output Mode	8 channels	
Phase-Delayed Waveform Output Mode	8 channels	
SR Waveform Output Mode	8 channels	
Communication	Communication unit 0	Communication unit 1
Clock Synchronous Serial I/O Mode	Available	Available
UART Mode	Not Available	
HDLC Data Processing Mode	Available	

**NOTE:**

1. The time measurement function and the waveform generating function share a pin.

The time measurement function and waveform generating function can be selected for each channel.

The communication function is available by a combining multiple channels.



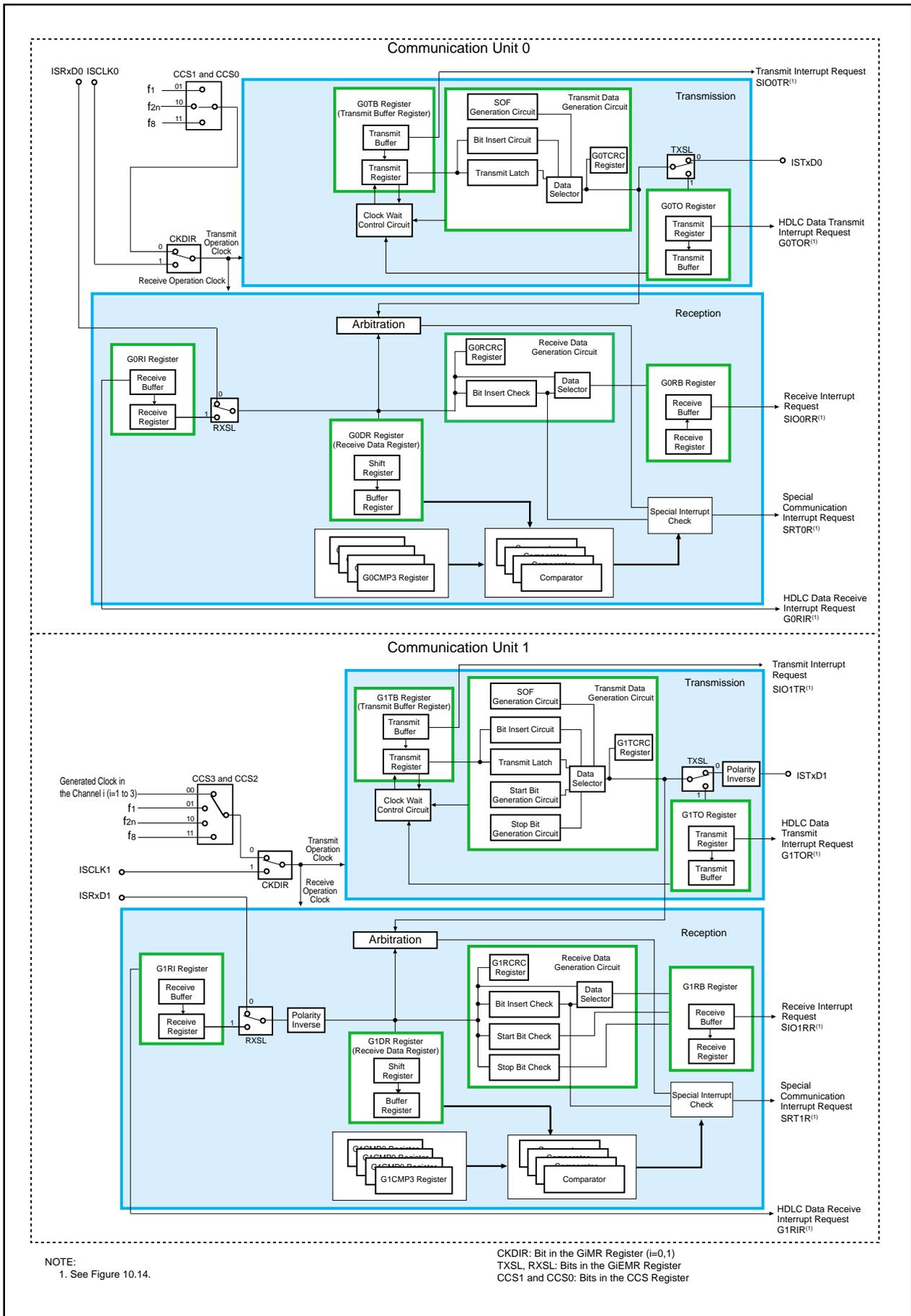


Figure 21.2 Intelligent I/O Communication Block Diagram

Figures 21.3 to 21.8 show registers associated with the intelligent I/O base timer, the time measurement function and waveform generating function. (For registers associated with the communication function, see Figures 21.19 to 21.28.)

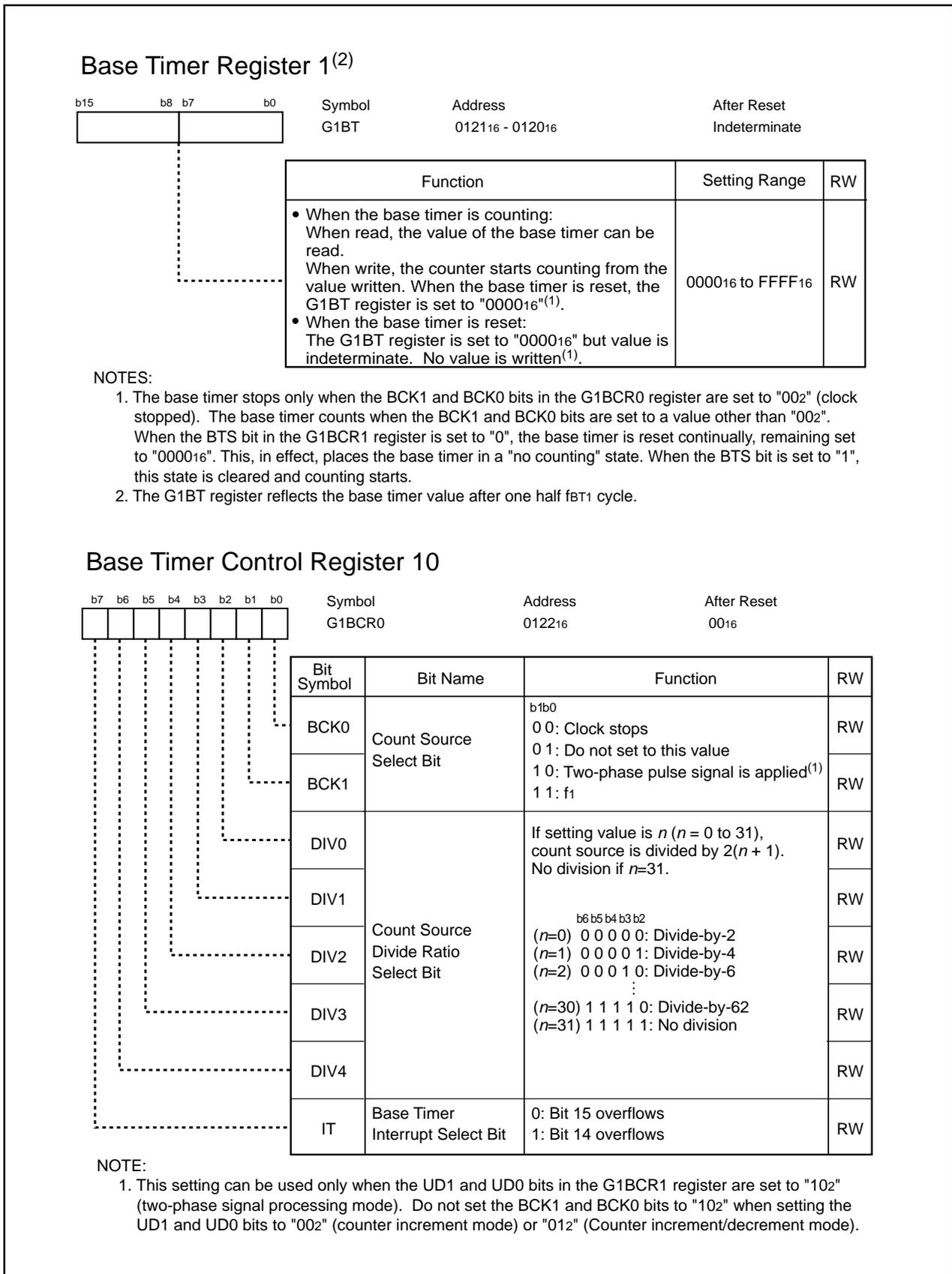


Figure 21.3 G1BT Register and G1BCR0 Register

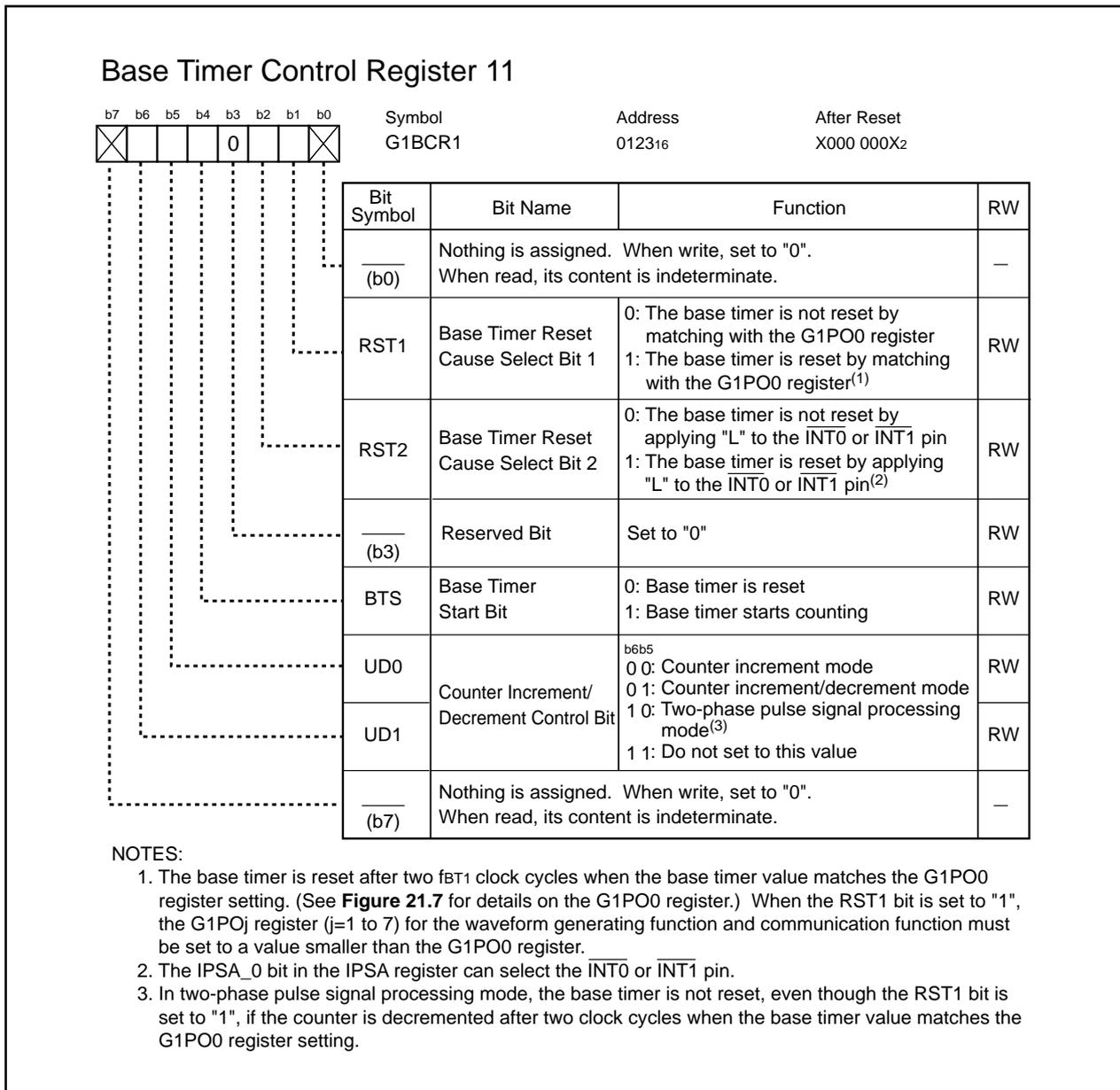
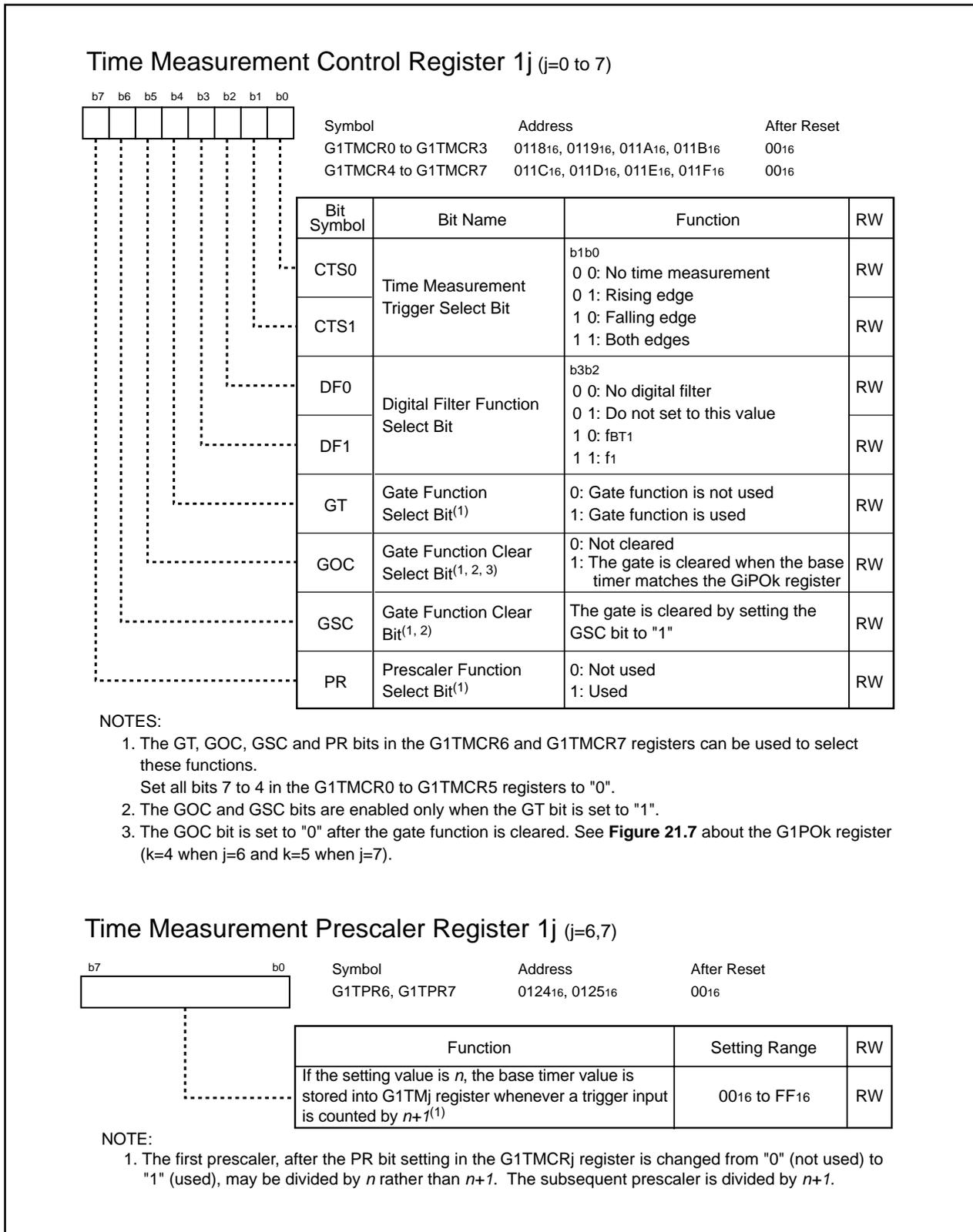
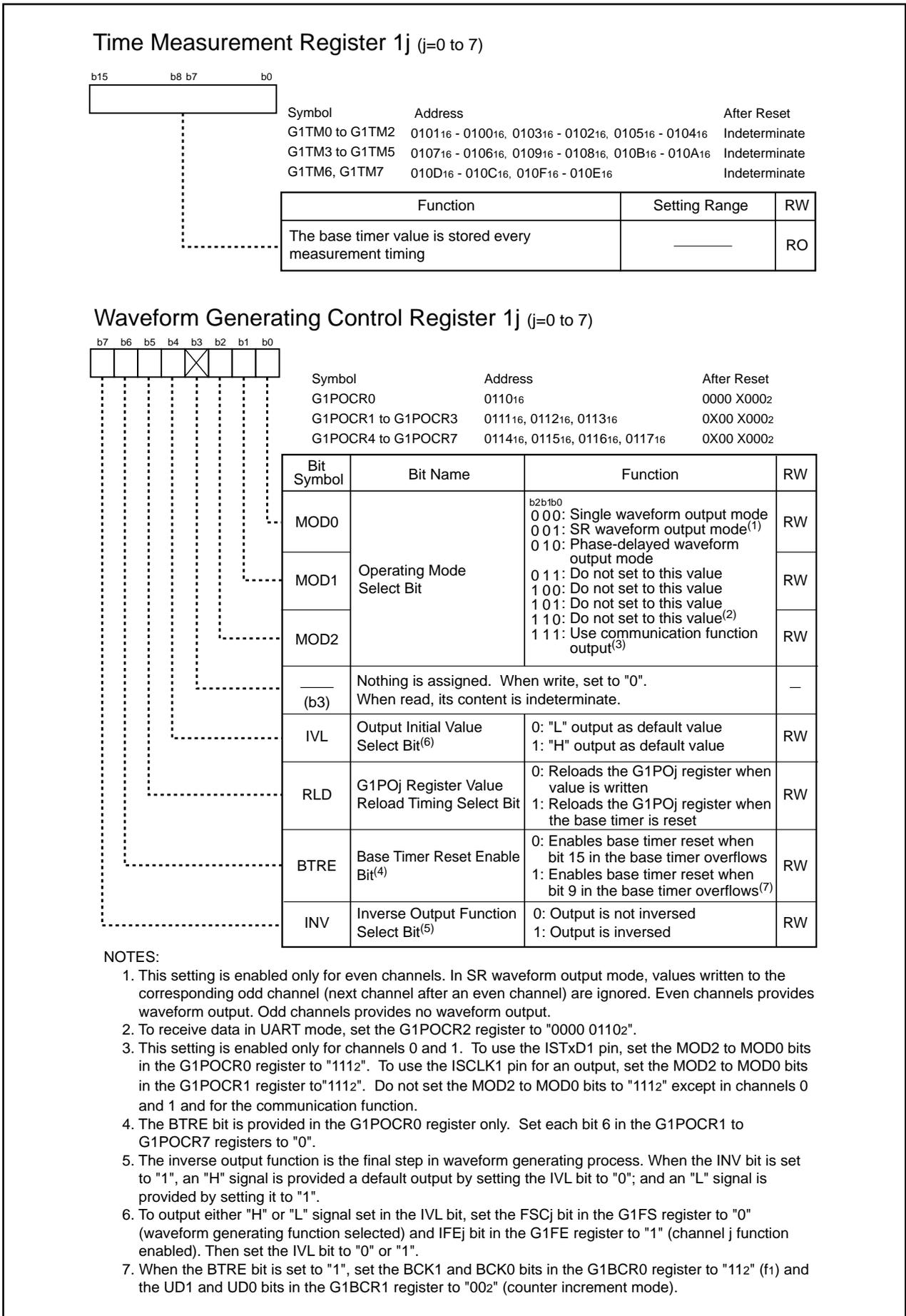


Figure 21.4 G1BCR1 Register



**Figure 21.5 G1TMCR0 to G1TMCR7 Registers, G1TPR6 and G1TPR7 Registers**



**Figure 21.6 G1TM0 to G1TM7 Registers and G1POCR0 to G1POCR7 Registers**

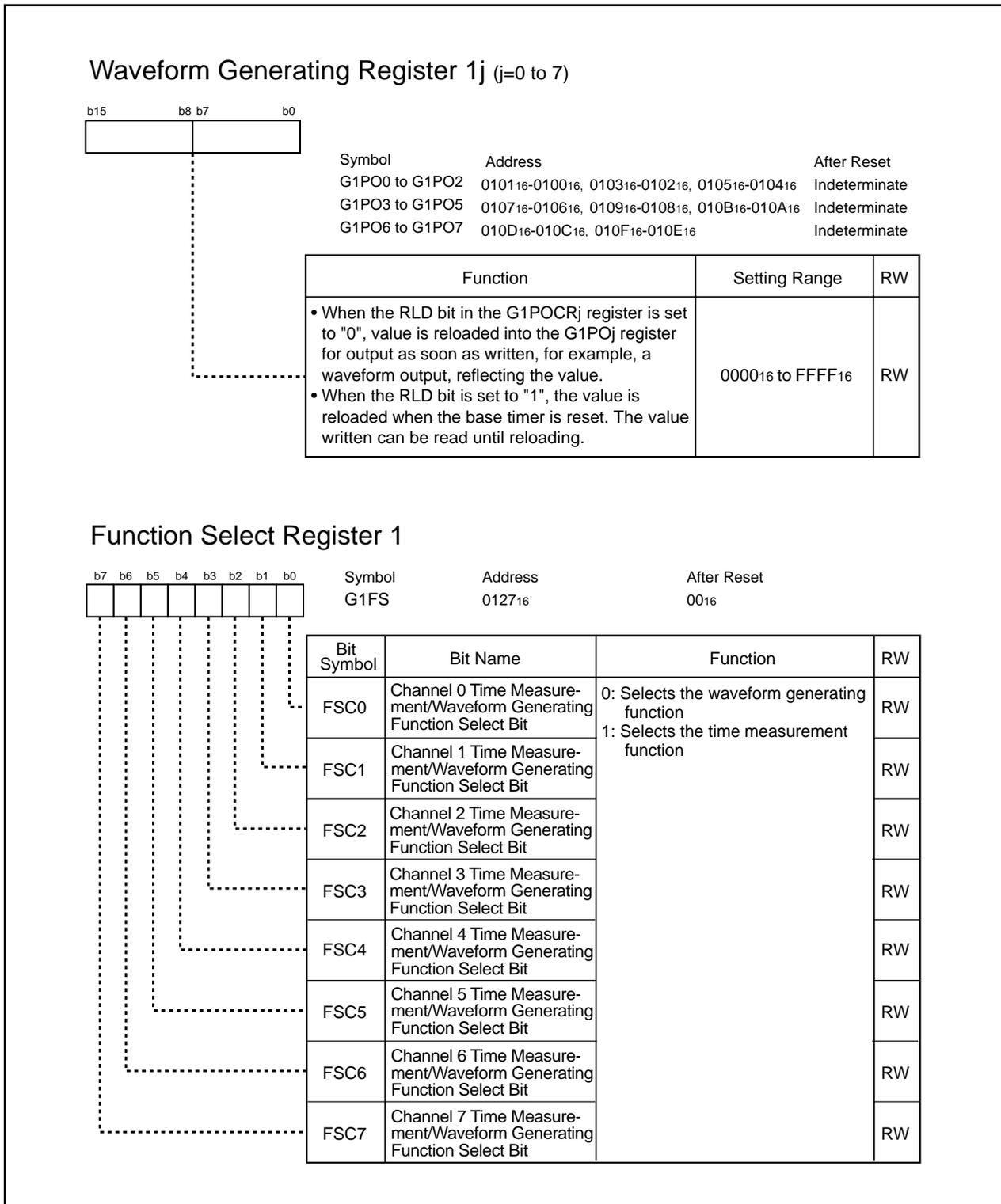
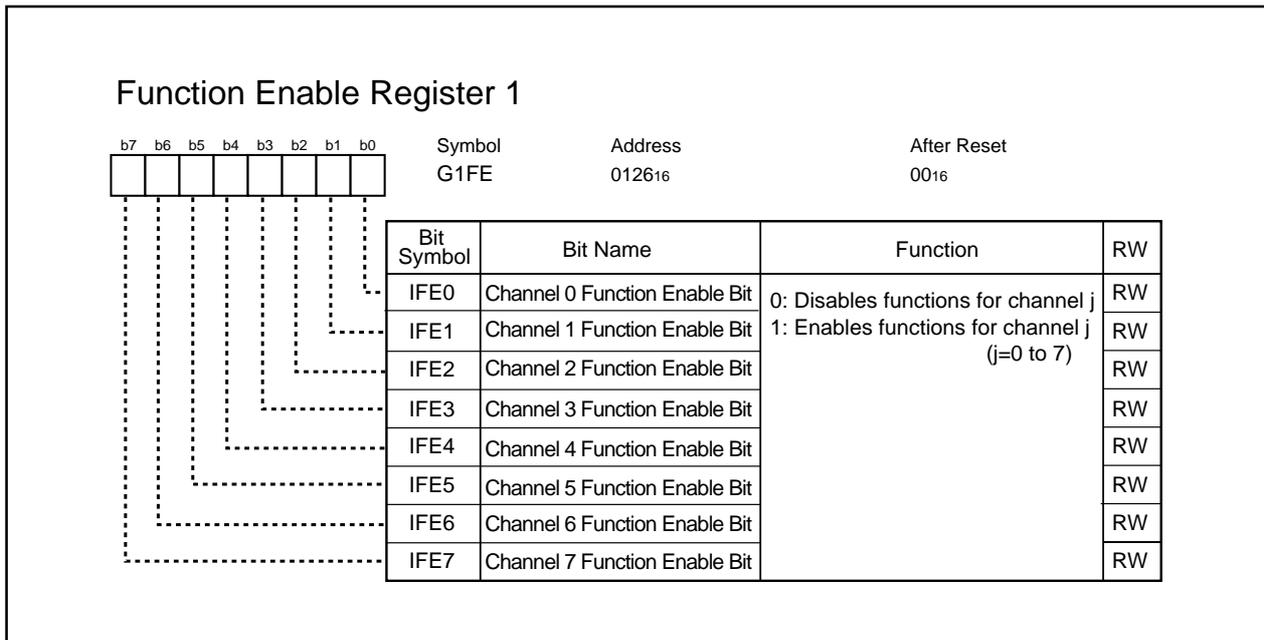


Figure 21.7 G1PO0 to G1PO7 Registers and G1FS Register



**Figure 21.8 G1FE Register**

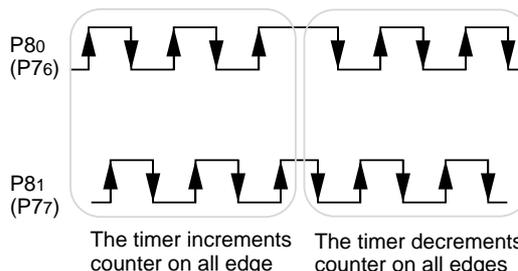
## 21.1 Base Timer

The base timer is a free-running counter that counts an internally generated count source.

Table 21.2 lists specifications of the base timer. Figures 21.3 and 21.4 show registers associated with the base timer. Figure 21.9 shows a block diagram of the base timer. Figure 21.10 shows an example of the base timer in counter increment mode. Figure 21.11 shows an example of the base timer in counter increment/decrement mode. Figure 21.12 shows an example of two-phase pulse signal processing mode.

**Table 21.2 Base Timer Specifications**

Item	Specification
Count Source (fBT1)	f1 divided by $2^{(n+1)}$ , two-phase pulse input divided by $2^{(n+1)}$ $n$ : determined by the DIV4 to DIV0 bits in the G1BCR0 register $n=0$ to 31; however no division when $n=31$
Counting Operation	The base timer increments the counter value The base timer increments and decrements the counter value Two-phase pulse signal processing
Counter Start Condition	The BTS bit in the G1BCR1 register is set to "1" (base timer starts counting)
Counter Stop Condition	The BTS bit in the G1BCR1 register is set to "0" (base timer reset)
Base Timer Reset Condition	<ul style="list-style-type: none"> <li>The value of the base timer matches the value of the G1PO0 register</li> <li>An low-level ("L") signal is applied to the <math>\overline{\text{INT0}}</math> or <math>\overline{\text{INT1}}</math> pin</li> <li>Bit 15 or bit 9 in the base timer overflows</li> </ul>
Value when the Base Timer is Reset	"0000 <sub>16</sub> "
Interrupt Request	The BT1R bit in the IIO4IR register is set to "1" (interrupt requested) when bit 9, bit 14 or bit 15 in the base timer overflows (See Figure 10.14.)
Read from Base Timer	<ul style="list-style-type: none"> <li>The G1BT register indicates the counter value while the base timer is running</li> <li>The G1BT register is indeterminate when the base timer is reset</li> </ul>
Write to Base Timer	When a value is written while the base timer is running, the timer counter immediately starts counting from this value. No value can be written while the base timer is reset
Selectable Function	<ul style="list-style-type: none"> <li>Counter increment/decrement mode  The base timer starts counting when the BTS bit is set to "1". After reaching to "FFFF<sub>16</sub>", the timer counter is then decremented back to "0000<sub>16</sub>". If the RST1 bit in the G1BCR1 register is set to "1" (the base timer is reset by matching with the G1PO0 register), the timer counter starts decrementing in two counts after the base timer matches the G1PO0 register. The base timer increments the counter value again when the timer counter reaches "0000<sub>16</sub>". (See <b>Figure 21.11</b>.)</li> <li>Two-phase pulse processing mode  Two-phase pulse signals from P76 and P77 pins or P80 and P81 pins are counted as well. (See Figure 21.12.)  The IPSA_0 bit in the IPSA register controls input pin selection. (Refer to <b>23. Programmable I/O Ports</b>)</li> </ul>



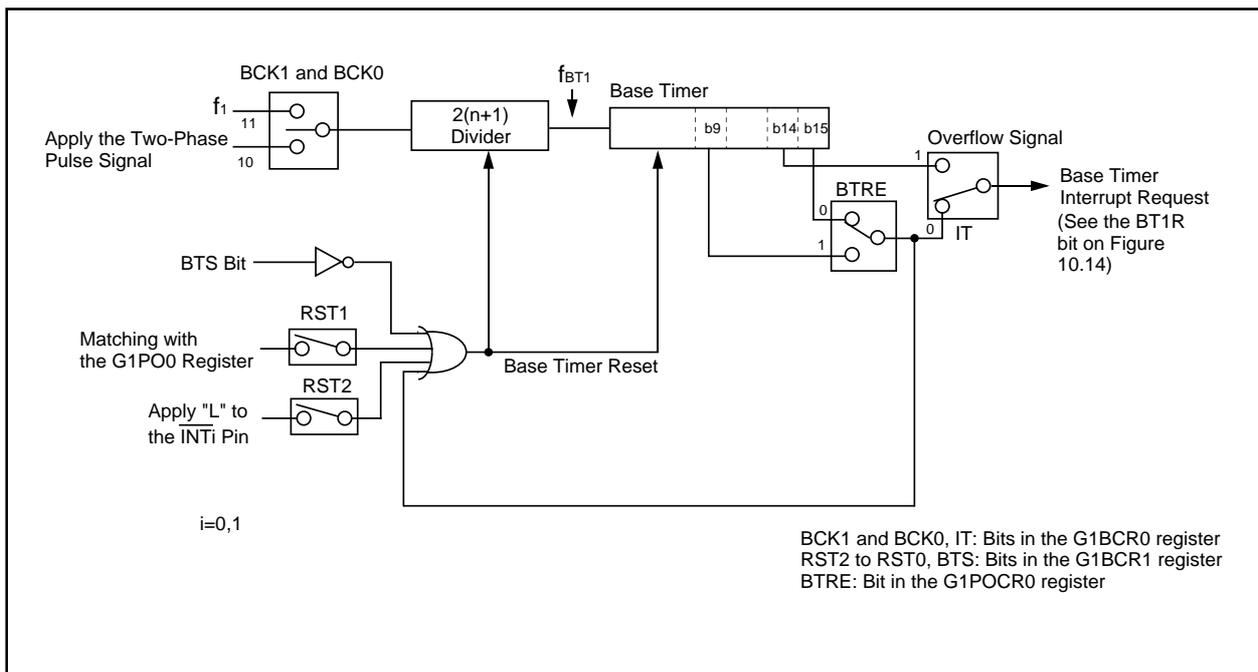


Figure 21.9 Base Timer Block Diagram

Table 21.3 Base Timer Associated Register Settings

(Also applies when using time measurement function, waveform generating function and communication function)

Register	Bit	Function
G1BCR0	BCK1, BCK0	Select count source
	DIV4 to DIV0	Select divide ratio of count source
	IT	Select the base timer interrupt
G1BCR1	RST2, RST1	Select source for a base timer reset
	BTS	Used to start the base timer independently
	UD1, UD0	Select how to count
G1POCR0	BTRE	Select source for a base timer reset
G1BT	-	Read or write base timer value

Set the following registers to set the RST1 bit to "1" (base timer reset by matching the base timer with the G1PO0 register).

G1POCR0	MOD2 to MOD0	Set to "0002" (single-phase waveform output mode)
G1PO0	-	Set reset cycle
G1FS	FSC0	Set to "0" (waveform generating function)
G1FE	IFE0	Set to "1" (channel operation start)

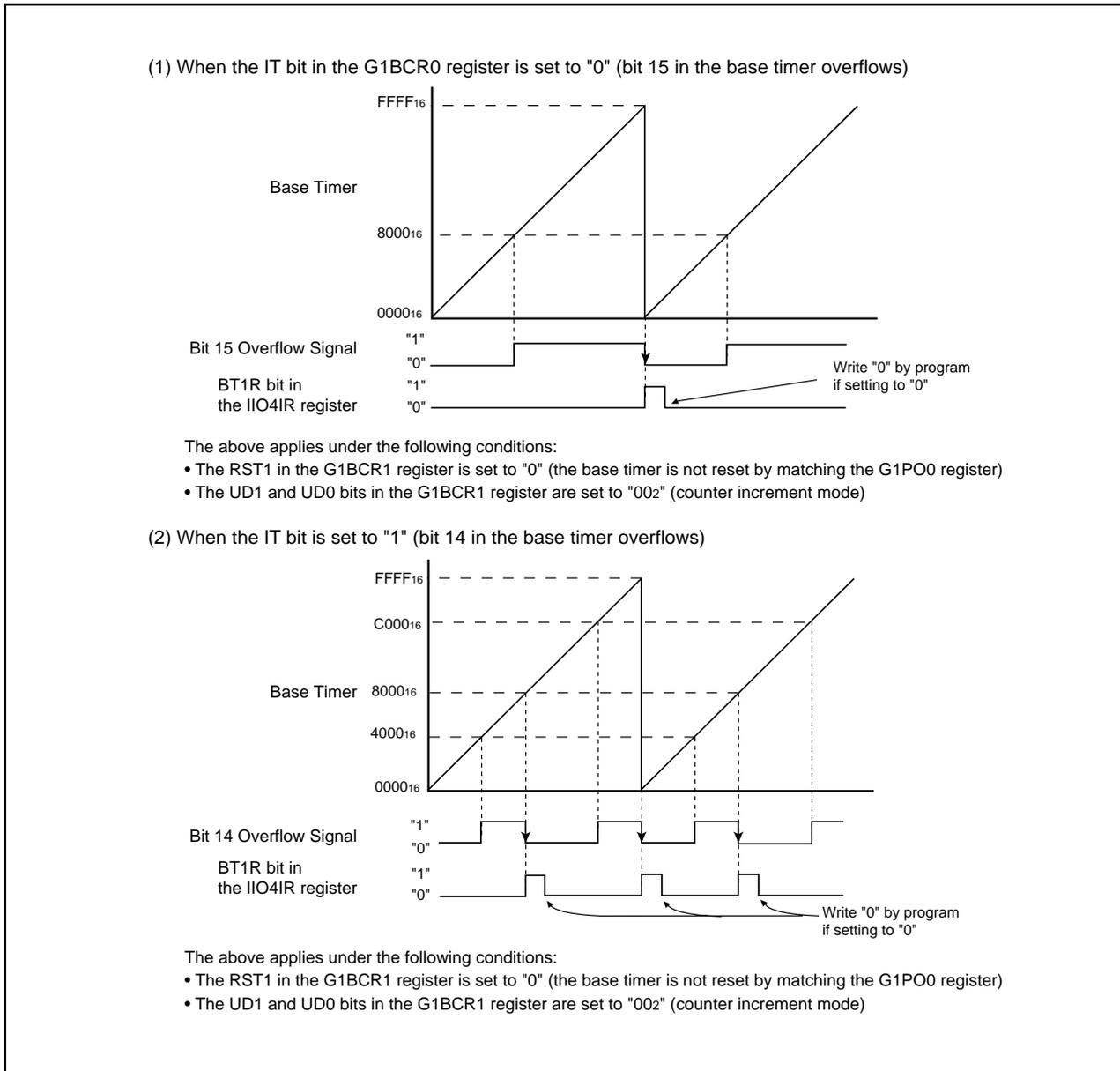
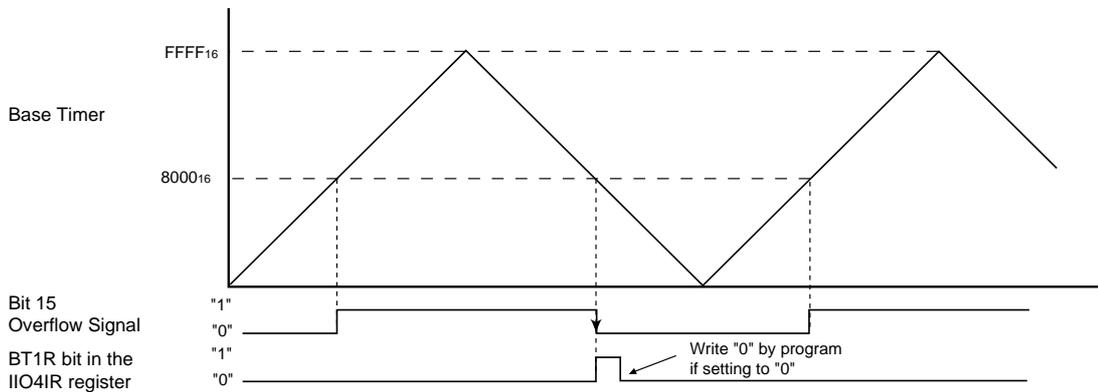


Figure 21.10 Counter Increment Mode

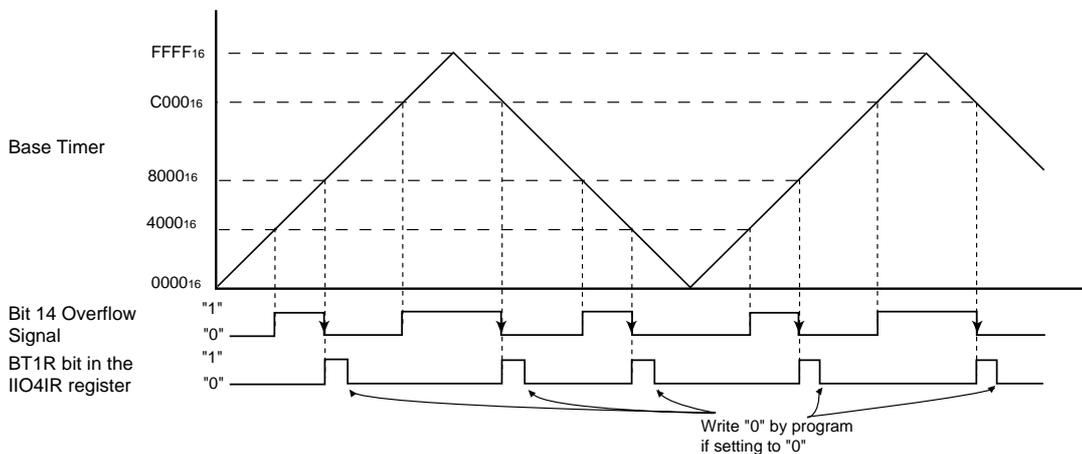
(1) When the IT bit in the G1BCR0 register is set to "0" (bit 15 in the base timer overflows)



The above applies under the following conditions:

- The RST1 in the G1BCR1 register is set to "0" (the base timer is not reset by matching the G1PO0 register)
- The UD1 and UD0 bits in the G1BCR1 register are set to "012" (counter increment/decrement mode)

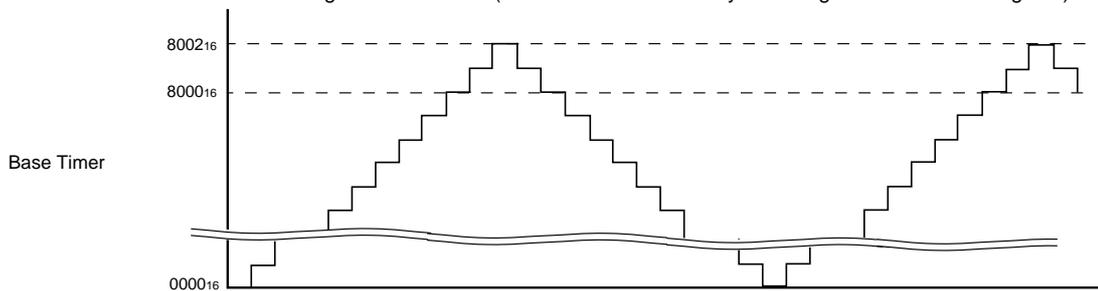
(2) When the IT bit is set to "1" (bit 14 in the base timer overflows)



The above applies under the following conditions:

- The RST1 in the G1BCR1 register is set to "0" (the base timer is not reset by matching the G1PO0 register)
- The UD1 and UD0 bits in the G1BCR1 register are set to "012" (counter increment/decrement mode)

(3) When the RST1 bit in the G1BCR1 register is set to "1" (the base timer is reset by matching with the G1PO0 register)



The above applies under the following conditions:

- Value of G1PO0 register: "8000h"
- The UD1 and UD0 bits in the G1BCR1 register are set to "012" (counter increment/decrement mode)

Figure 21.11 Counter Increment/Decrement Mode

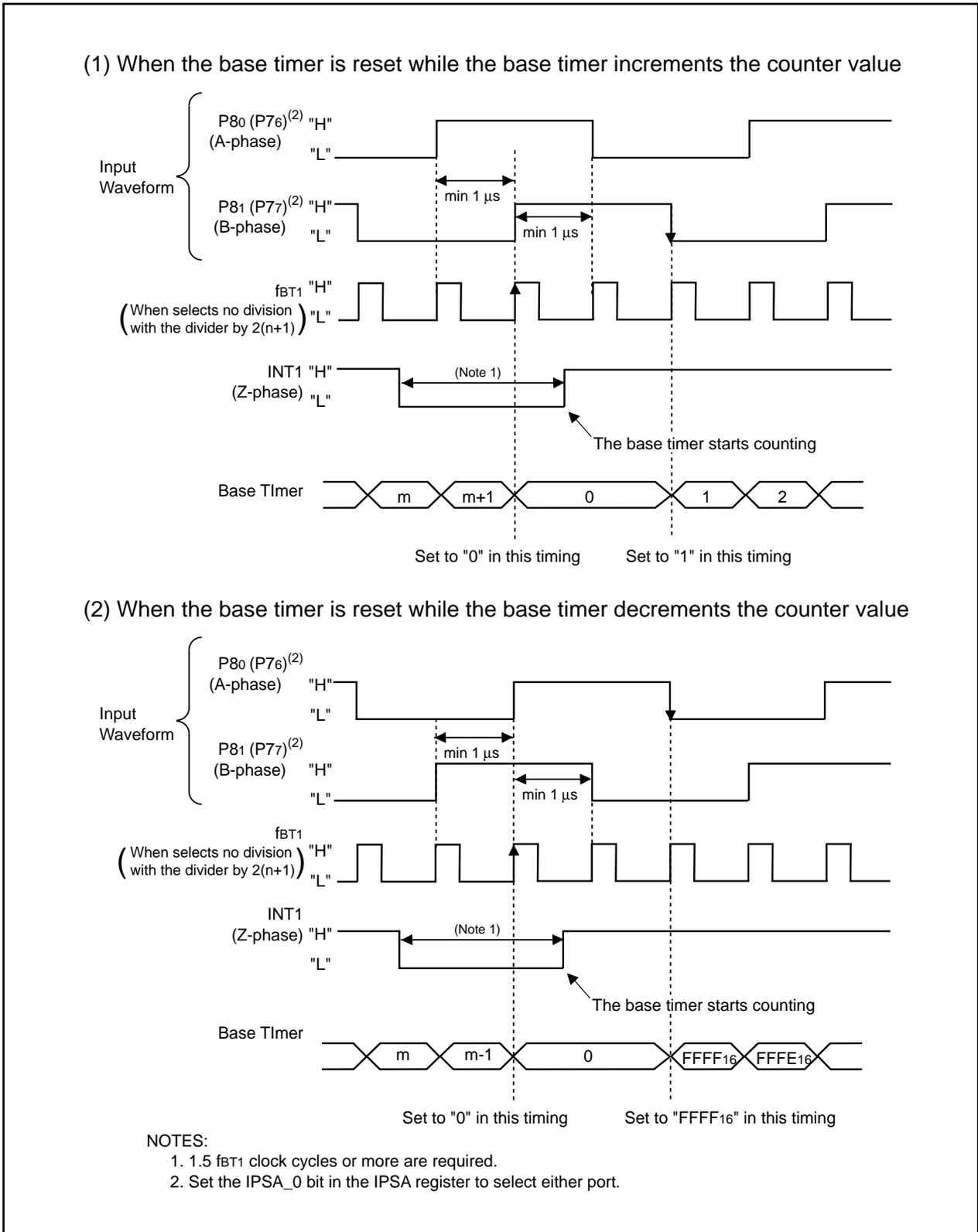


Figure 21.12 Base Timer Operation in Two-phase Pulse Signal Processing Mode

## 21.2 Time Measurement Function

When external trigger is applied, the base timer value is stored into the G1TMj register (j=0 to 7). Table 21.4 shows specifications of the time measurement function. Tables 21.5 and 21.6 list pin settings of the time measurement function. Figures 21.13 and 21.14 show operation examples of the time measurement function. Figure 21.15 shows an operation example of the prescaler function and gate function.

**Table 21.4 Time Measurement Function Specifications**

Item	Specification
Measurement Channel	Channels 0 to 7
Trigger Input Polarity	Rising edge, falling edge and both edges of the INPC1j pin
Measurement Start Condition	The IFEj bit in the G1FE register is set to "1" (channel j function enabled) while the FSCj bit (j=0 to 7) in the G1FS register is set to "1" (time measurement function selected)
Measurement Stop Condition	The IFEj bit is set to "0" (channel j function disabled)
Time Measurement Timing	<ul style="list-style-type: none"> <li>• No prescaler: every time a trigger signal is applied</li> <li>• Prescaler (for channel 6 and channel 7): every <i>G1TPRk register (k=6,7) value +1</i> times a trigger signal is applied</li> </ul>
Interrupt Request Generating Timing	The TM1jR bit in the interrupt request register (See Figure 10.14) is set to "1" (interrupt requested) at time measurement timing
INPC1j Pin Function	Trigger input pin
Selectable Function	<ul style="list-style-type: none"> <li>• Digital filter function The digital filter samples a trigger input signal level every f1 or fBT1 cycles and passes pulse signals, matching trigger input signal level three times</li> <li>• Prescaler function (for channel 6 and channel 7) Time measurement is executed every <i>G1TPRk register value +1</i> times a trigger signal is applied</li> <li>• Gate function (for channel 6 and channel 7) After time measurement by the first trigger input, trigger input cannot be accepted. However, while the GOC bit in the G1TMCRk register is set to "1" (gate cleared by matching the base timer with the G1POp register (p=4 when k=6, p=5 when k=7), trigger input can be accepted again by matching the base timer value with the G1POp register setting or by setting the GSC bit in the G1TMCRk register is set to "1"</li> </ul>

**Table 21.5 Pin Settings for Time Measurement Function**

Pin	Bit and Setting		
	PS1, PS2, PS5, PS8 Registers	PD7, PD8, PD11, PD14 Registers	IPS Register
P70/INPC16	PS1_0 = 0	PD7_0 = 0	IPS1 = 0
P71/INPC17	PS1_1 = 0	PD7_1 = 0	
P73/INPC10	PS1_3 = 0	PD7_3 = 0	
P74/INPC11	PS1_4 = 0	PD7_4 = 0	
P75/INPC12	PS1_5 = 0	PD7_5 = 0	
P76/INPC13	PS1_6 = 0	PD7_6 = 0	
P77/INPC14	PS1_7 = 0	PD7_7 = 0	
P81/INPC15	PS2_1 = 0	PD8_1 = 0	
P110/INPC10 <sup>(1)</sup>	PS5_0 = 0	PD11_0 = 0	IPS1 = 1
P111/INPC11 <sup>(1)</sup>	PS5_1 = 0	PD11_1 = 0	
P112/INPC12 <sup>(1)</sup>	PS5_2 = 0	PD11_2 = 0	
P113/INPC13 <sup>(1)</sup>	PS5_3 = 0	PD11_3 = 0	
P140/INPC14 <sup>(1)</sup>	PS8_0 = 0	PD14_0 = 0	
P141/INPC15 <sup>(1)</sup>	PS8_1 = 0	PD14_1 = 0	
P142/INPC16 <sup>(1)</sup>	PS8_2 = 0	PD14_2 = 0	
P143/INPC17 <sup>(1)</sup>	PS8_3 = 0	PD14_3 = 0	

## NOTE:

1. This port is provided in the 144-pin package only.

**Table 21.6 Time Measurement Function Associated Register Settings**

Register	Bit	Function
G1TMCRj	CTS1, CTS0	Select a time measurement trigger
	DF1, DF0	Select the digital filter function
	GT, GOC, GSC	Select the gate function
	PR	Select the prescaler function
G1TPRk	-	Setting value of the prescaler
G1FS	FSCj	Set to "1" (time measurement function)
G1FE	IFEj	Set to "1" (channel j function enabled)

j = 0 to 7 k = 6, 7

Bit configurations and functions vary with channels used.

Registers associated with the time measurement function must be set after setting registers associated with the base timer.

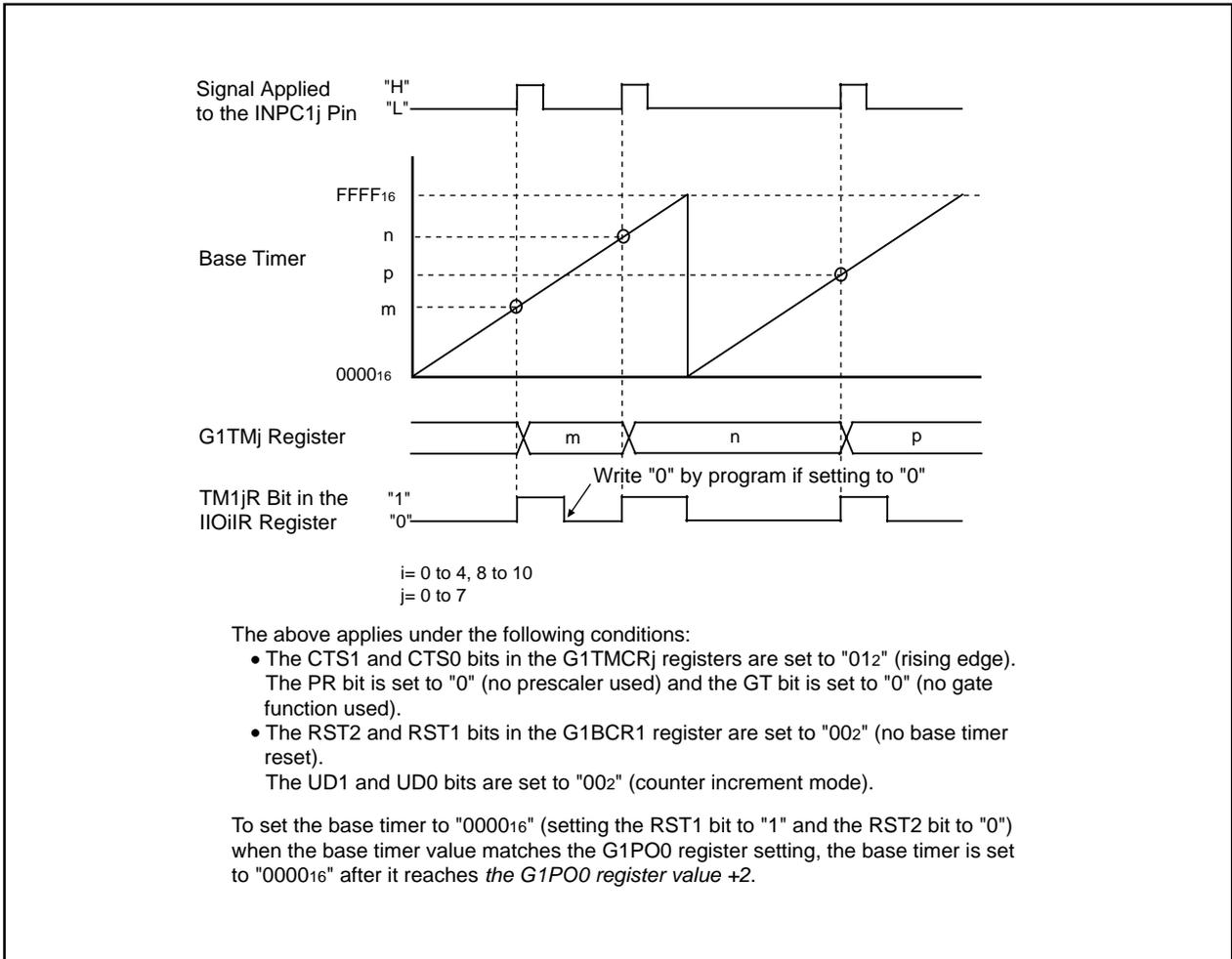


Figure 21.13 Time Measurement Function (1)

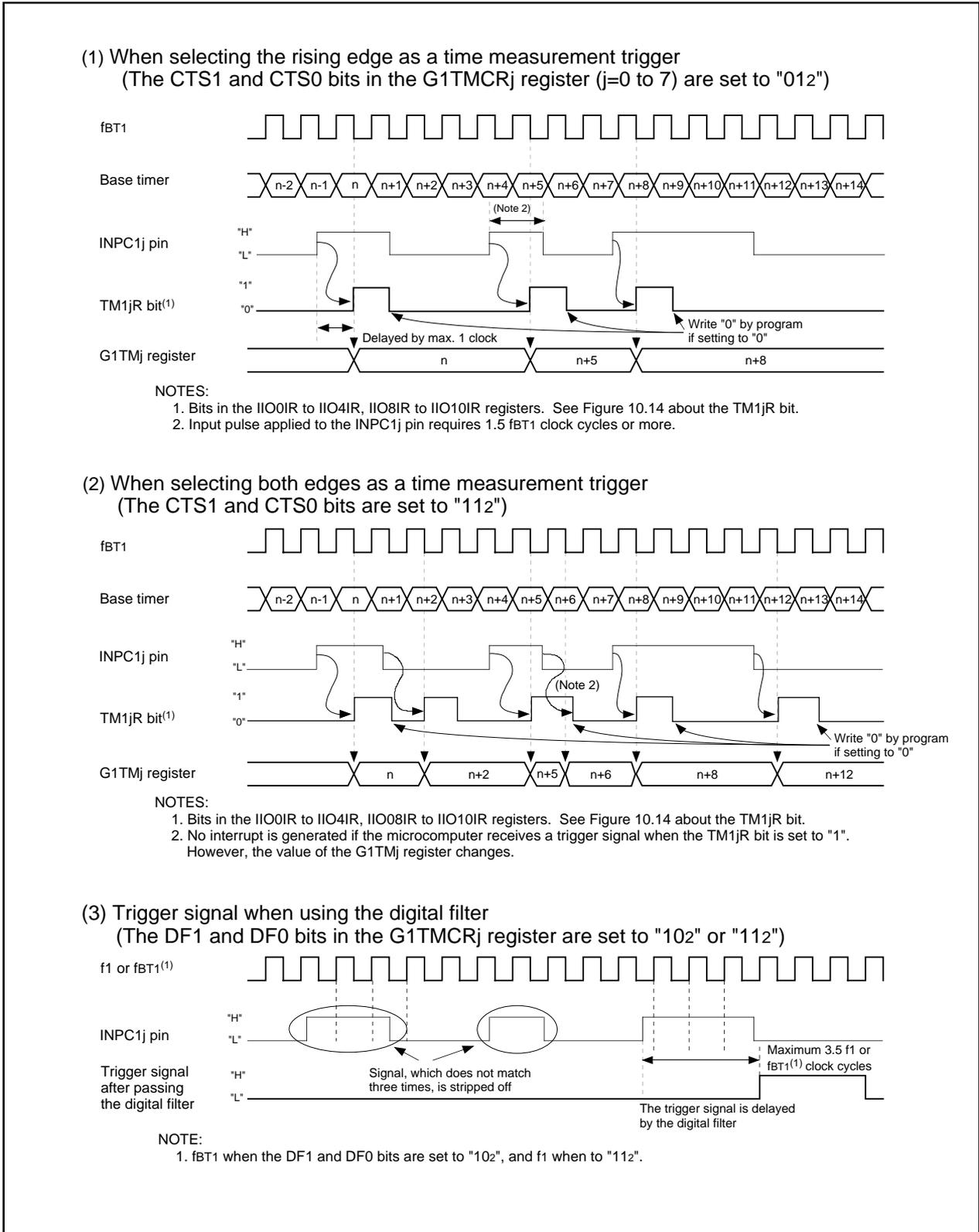


Figure 21.14 Time Measurement Function (2)

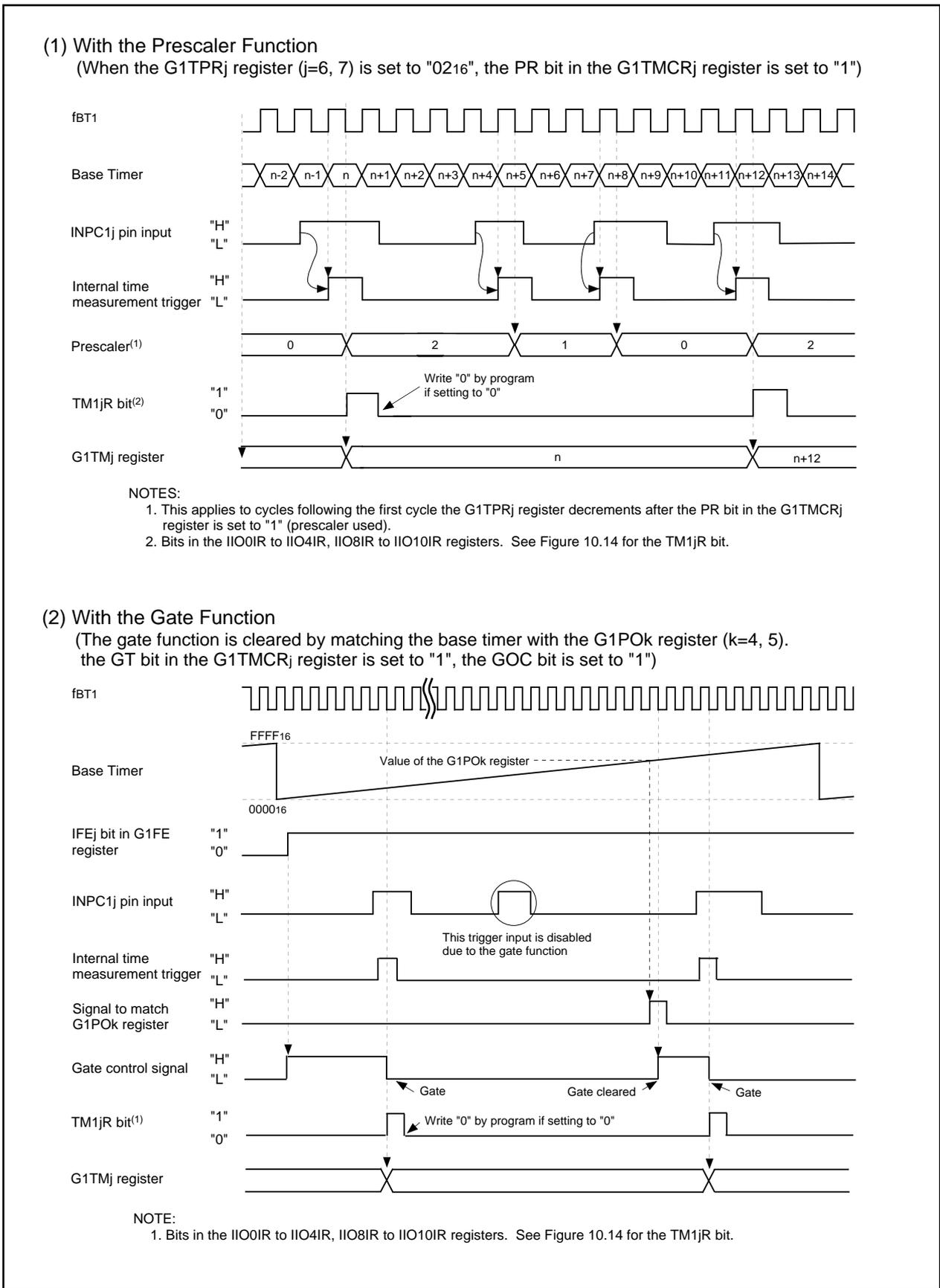


Figure 21.15 Prescaler Function and Gate Function

### 21.3 Waveform Generating Function

Waveforms are generated when the value of the base timer matches that of the G1POj register (j=0 to 7). The waveform generating function has the following three modes :

- Single-phase waveform output mode
- Phase-delayed waveform output mode
- Set/Reset waveform output (SR waveform output) mode

Table 21.7 lists pin settings of the waveform generating function. Table 21.8 lists registers associated with the waveform generating function.

**Table 21.7 Pin Settings for Waveform Generating Function**

Pin	Bit and Setting			
	PS1, PS2, PS5 to PS8 Registers	PSL1, PSL2 Registers	PSC, PSC2 Registers	PSD1 Register
P70/OUTC16	PS1_0 = 1	PSL1_0 = 0	PSC_0 = 1	PSD1_0=1
P71/OUTC17	PS1_1 = 1	PSL1_1 = 0	PSC_1 = 1	PSD1_1=1
P73/OUTC10	PS1_3 = 1	PSL1_3 = 0	PSC_3 = 1	-
P74/OUTC11	PS1_4 = 1	PSL1_4 = 0	PSC_4 = 1	-
P75/OUTC12	PS1_5 = 1	PSL1_5 = 1	-	-
P76/OUTC13	PS1_6 = 1	PSL1_6 = 0	PSC_6 = 0	PSD1_6=1
P77/OUTC14	PS1_7 = 1	PSL1_7 = 1	-	-
P81/OUTC15	PS2_1 = 1	PSL2_1 = 1	PSC2_1=1	-
P110/OUTC10 <sup>(1)</sup>	PS5_0 = 1	-	-	-
P111/OUTC11 <sup>(1)</sup>	PS5_1 = 1			
P112/OUTC12 <sup>(1)</sup>	PS5_2 = 1			
P113/OUTC13 <sup>(1)</sup>	PS5_3 = 1			
P140/OUTC14 <sup>(1)</sup>	PS8_0 = 1			
P141/OUTC15 <sup>(1)</sup>	PS8_1 = 1			
P142/OUTC16 <sup>(1)</sup>	PS8_2 = 1			
P143/OUTC17 <sup>(1)</sup>	PS8_3 = 1			

NOTE:

1. This port is provided in the 144-pin package only.

**Table 21.8 Waveform Generating Function Associated Register Settings**

Register	Bit	Function
G1POCRj	MOD2 to MOD0	Select waveform output mode
	IVL	Select default output value
	RLD	Select a timing to reload the value of the G1POj register
	INV	Select if output level is inversed
G1POj	-	Select when output waveform is inversed
G1FS	FSCj	Set to "0" (waveform generating function)
G1FE	IFEj	Set to "1" (channel j function enabled)

j = 0 to 7

Bit configurations and functions vary with channels used.

Registers associated with the waveform generating measurement function must be set after setting registers associated with the base timer.

### 21.3.1 Single-Phase Waveform Output Mode

Output signal level of the OUTC1j pin becomes high ("H") when the base timer value matches the G1POj register (j=0 to 7) setting. The "H" signal switches to a low-level ("L") signal when the base timer reaches "0000<sub>16</sub>". If the IVL bit in the G1POCRj register is set to "1" ("H" output as default value), an "H" signal output is provided when waveform output starts. If the INV bit is set to "1" (output inversed), the level of the waveform output is inversed. See Figure 21.16 for details on single-phase waveform output mode operation. Table 21.9 lists specifications of single-phase waveform output mode.

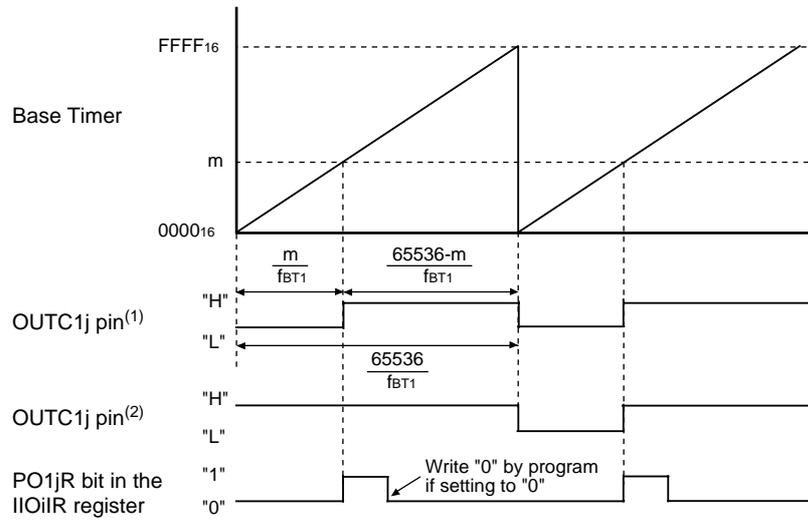
**Table 21.9 Single-Phase Waveform Output Mode Specifications**

Item	Specification
Output Waveform <sup>(2)</sup>	<ul style="list-style-type: none"> <li>Free-running operation (the RST2 and RST1 bits in the G1BCR1 register are set to "002")</li> </ul> <p>Cycle : <math>\frac{65536}{f_{BT1}}</math></p> <p>"L" width : <math>\frac{m}{f_{BT1}}</math></p> <p>"H" width : <math>\frac{65536-m}{f_{BT1}}</math></p> <p>m : setting value of the G1POj register (j=0 to 7), 0000<sub>16</sub> to FFFF<sub>16</sub></p> <ul style="list-style-type: none"> <li>The base timer is cleared to "0000<sub>16</sub>" by matching the base timer with the G1PO0 register (the RST1 bit is set to "1" and the RST2 bit is set to "0")</li> </ul> <p>Cycle : <math>\frac{n+2}{f_{BT1}}</math></p> <p>"L" width : <math>\frac{m}{f_{BT1}}</math></p> <p>"H" width : <math>\frac{n+2-m}{f_{BT1}}</math></p> <p>m : setting value of the G1POj register (j=1 to 7), 0000<sub>16</sub> to FFFF<sub>16</sub></p> <p>n : setting value of the G1PO0 register, 0001<sub>16</sub> to FFFD<sub>16</sub></p> <p>If <math>m \geq n+2</math>, the output level is fixed to "L"</p>
Waveform Output Start Condition <sup>(1)</sup>	The IFEj bit in the G1FE register is set to "1" (channel j function enabled)
Waveform Output Stop Condition	The IFEj bit is set to "0" (channel j function disabled)
Interrupt Request	The PO1jR bit in the interrupt request register is set to "1" (interrupt requested) when the base timer value matches the G1POj register setting. (See Figure 10.14)
OUTC1j Pin	Pulse signal output pin
Selectable Function	<ul style="list-style-type: none"> <li>Default value set function: Set starting waveform output level</li> <li>Inversed output function: Waveform output signal is inversed and provided from the OUTC1j pin</li> </ul>

**NOTES:**

- Set the FSCj bit in the G1FS register to "0" (waveform generating function selected).
- When the INV bit in the G1POCRj register is set to "1" (output inversed), the "L" width and "H" width are inversed.

(1) Free-Running Operation  
 (The RST2 and RST1 bits in the G1BCR1 register are set to "002")



$i=0$  to 4, 8 to 10;  $j=0$  to 7  
 m: Setting value of the G1POj register, 0000<sub>16</sub> to FFFF<sub>16</sub>

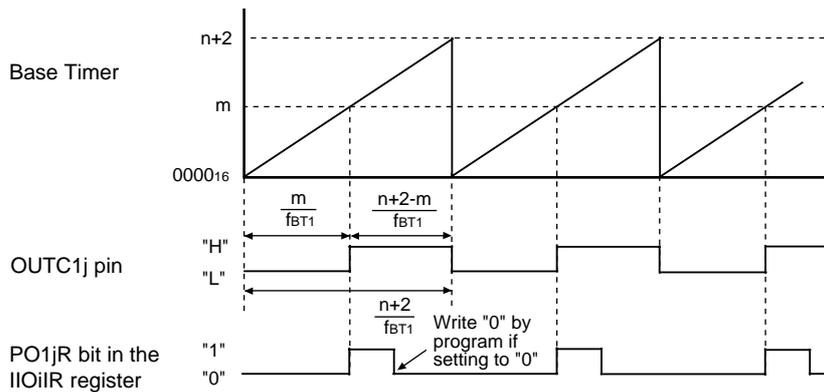
NOTES:

1. Waveform output when the INV bit in the G1POCRj register is set to "0" (not inverted) and the IVL bit in the G1POCRi register is set to "0" (output "L" as default value).
2. Waveform output when the INV bit is set to "0" (not inverted) and the IVL bit is set to "1" ("H" output as default value).

The above applies under the following condition:

- The RST2 and RST1 bits in the G1BCR1 register are set to "002" (no base timer reset and the UD1 and UD0 bits in the G1BCR1 register to "002" (counter increment mode).

(2) The Base Timer is Reset by Matching the Base Timer with the G1PO0 Register  
 (The RST1 bit is set to "1" and the RST2 bit is set to "0")



$i=0$  to 4, 8 to 10;  $j=1$  to 7  
 m: Setting value of the G1POj register, 0000<sub>16</sub> to FFFF<sub>16</sub>  
 n: Setting value of the G1PO0 register, 0001<sub>16</sub> to FFFD<sub>16</sub>

The above applies under the following conditions:

- The IVL bit in the G1POCRj register is set to "0" ("L" output as default value) and the INV bit is set to "0" (not inverted).
- The UD1 and UD0 bits are set to "002" (counter increment mode).
- $m < n+2$

Figure 21.16 Single-Phase Waveform Output Mode

### 21.3.2 Phase-Delayed Waveform Output Mode

Output signal level of the OUTC1j pin is inverted every time the base timer value matches the G1POj register (j=0 to 7) setting. Table 21.10 lists specifications of phase-delayed waveform output mode. Figure 21.17 lists an example of phase-delayed waveform output mode operation.

**Table 21.10 Phase-Delayed Waveform Output Mode Specifications**

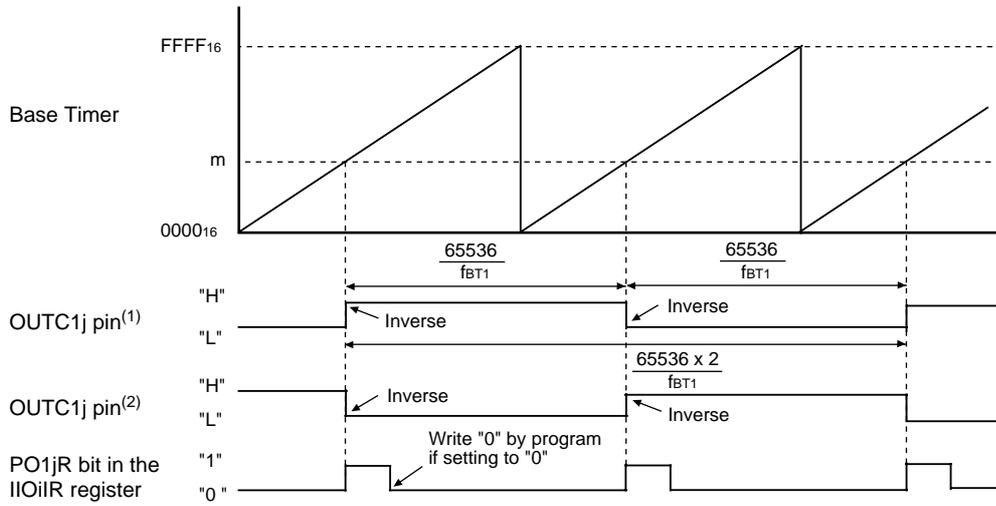
Item	Specification
Output Waveform	<ul style="list-style-type: none"> <li>Free-running operation (the RST2 and RST1 bits in the G1BCR1 register are set to "002")</li> </ul> <p>Cycle : <math>\frac{65536 \times 2}{f_{BT1}}</math></p> <p>"H" and "L" widths : <math>\frac{65536}{f_{BT1}}</math></p> <p>Setting value of the G1POj (j=0 to 7) register is 0000<sub>16</sub> to FFFF<sub>16</sub></p> <ul style="list-style-type: none"> <li>The base timer is cleared to "0000<sub>16</sub>" by matching the base timer with the G1PO0 register (the RST1 bit is set to "1" and the RST2 bit is set to "0")</li> </ul> <p>Cycle : <math>\frac{2(n+2)}{f_{BT1}}</math></p> <p>"H" and "L" widths : <math>\frac{n+2}{f_{BT1}}</math></p> <p>n : setting value of the G1PO0 register, 0001<sub>16</sub> to FFFD<sub>16</sub></p> <p>Setting value of the G1POj (j=1 to 7) register is 0000<sub>16</sub> to FFFF<sub>16</sub></p> <p>If G1POj register <math>\geq n+2</math>, the output level is not inverted</p>
Waveform Output Start Condition <sup>(1)</sup>	The IFEj bit (j=0 to 7) in the G1FE register is set to "1" (channel j function enabled)
Waveform Output Stop Condition	The IFEj bit is set to "0" (channel j function disabled)
Interrupt Request	The PO1jR bit in the interrupt request register is set to "1" (interrupt requested) when the base timer vslur matches the G1POj register setting. (See Figure 10.14)
OUTC1j Pin	Pulse signal output pin
Selectable Function	<ul style="list-style-type: none"> <li>Default value set function: Set starting waveform output level</li> <li>Inversed output function</li> </ul> <p>Waveform output level is inverted to output a waveform from the OUTC1j pin</p>

**NOTE:**

- Set the FSCj bit in the G1FS register to "0" (waveform generating function selected).

(1) Free-Running Operation

(The RST2 and RST1 bits in the G1BCR1 register are set to "002")



$i=0$  to 4, 8 to 10;  $j=0$  to 7

m: Setting value of the G1POj register, 0000<sub>16</sub> to FFFF<sub>16</sub>

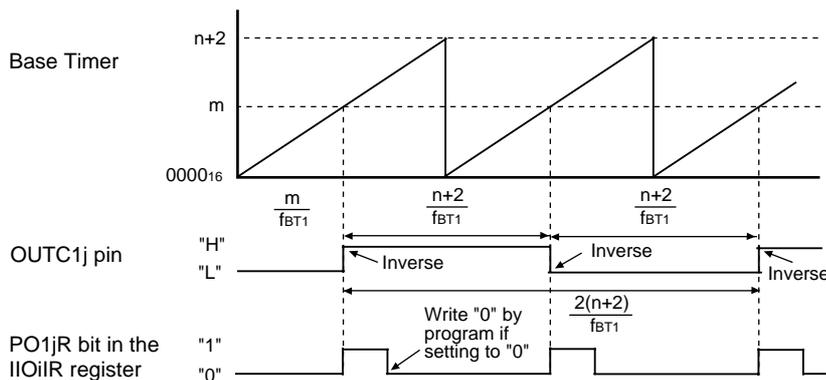
NOTES:

1. Waveform output when the INV bit in the G1POCRj register is set to "0" (not inverted) and the IVL bit in the G1POCRj register is set to "0" ("L" output as default value).
2. Waveform output when the INV bit is set to "0" (not inverted) and the IVL bit is set to "1" ("H" output as default value).

The above applies under the following condition:

- The RST2 and RST1 bits in the G1BCR1 register are set to "002" (no base timer reset) and the UD1 and UD0 bits in the G1BCR1 register to "002" (counter increment mode).

(2) The Base Timer is Reset when the Base Timer Matches the G1PO0 Register  
(The RST1 bit is set to "1" and the RST2 bit is set to "0")



$i=0$  to 4, 8 to 10;  $j=1$  to 7

m: Setting value of the G1POj register, 0000<sub>16</sub> to FFFF<sub>16</sub>

n: Setting value of the G1PO0 register, 0001<sub>16</sub> to FFFD<sub>16</sub>

The above applies under the following conditions:

- The IVL bit in the G1POCRj register is set to "0" ("L" output as default value) and the INV bit is set to "0" (not inverted).
- The UD1 and UD0 bits are set to "002" (counter increment mode).
- $m < n+2$

Figure 21.17 Phase-delayed Waveform Output Mode

### 21.3.3 Set/Reset Waveform Output (SR Waveform Output) Mode

Output signal level of the OUTC1j pin becomes high ("H") when the base timer value matches the G1POj register ( $j=0, 2, 4, 6$ ) setting. The "H" signal switches to a low-level ("L") signal when the base timer value matches the G1POk register ( $k=j+1$ ) setting or when the base timer is set to "0000<sub>16</sub>". If the IVL bit in the G1POCRj register is set to "1" ("H" output as default value), an "H" signal output is provided when waveform output starts. If the INV bit is set to "1" (output inversed), the level of the output waveform is inversed. Table 21.11 lists specifications of SR waveform output mode. Figure 21.18 shows an example of a SR waveform output mode operation.

**Table 21.11 SR Waveform Output Mode Specifications**

Item	Specification
Output Waveform <sup>(2)</sup>	<ul style="list-style-type: none"> <li>• Free-running operation (the RST2 and RST1 bits in the G1BCR1 register are set to "002")</li> <li>(1) <math>m &lt; n</math> <ul style="list-style-type: none"> <li>"H" width : <math>\frac{n-m}{f_{BT1}}</math></li> <li>"L" width : <math>\frac{m^{(3)}}{f_{BT1}} + \frac{65536 - n^{(4)}}{f_{BT1}}</math></li> </ul> </li> <li>(2) <math>m \geq n</math> <ul style="list-style-type: none"> <li>"H" width : <math>\frac{65536 - m}{f_{BT1}}</math></li> <li>"L" width : <math>\frac{m}{f_{BT1}}</math></li> </ul> </li> <li style="padding-left: 40px;"><math>m</math> : setting value of the G1POj register (<math>j=0, 2, 4, 6</math>)</li> <li style="padding-left: 40px;"><math>n</math> : setting value of the G1POk register (<math>k=j+1</math>)</li> <li>• The base timer is cleared to "0000<sub>16</sub>" by matching the base timer with the G1PO0 register<sup>(1)</sup> (the RST1 bit is set to "1" and the RST2 bit is set to "0")</li> <li>(1) <math>m &lt; n &lt; p+2</math> <ul style="list-style-type: none"> <li>"H" width : <math>\frac{n-m}{f_{BT1}}</math></li> <li>"L" width : <math>\frac{m^{(3)}}{f_{BT1}} + \frac{p+2 - n^{(4)}}{f_{BT1}}</math></li> </ul> </li> <li>(2) <math>m &lt; p+2 \leq n</math> <ul style="list-style-type: none"> <li>"H" width : <math>\frac{p+2 - m}{f_{BT1}}</math></li> <li>"L" width : <math>\frac{m}{f_{BT1}}</math></li> </ul> </li> <li>(3) If <math>m \geq p+2</math>, the output level is fixed to "L"</li> <li style="padding-left: 40px;"><math>m</math> : setting value of the G1POj register (<math>j=2, 4, 6</math>), 0000<sub>16</sub> to FFFF<sub>16</sub></li> <li style="padding-left: 40px;"><math>n</math> : setting value of the G1POk register (<math>k=j+1</math>), 0000<sub>16</sub> to FFFF<sub>16</sub></li> <li style="padding-left: 40px;"><math>p</math> : setting value of the G1PO0 register, 0001<sub>16</sub> to FFFD<sub>16</sub></li> </ul>

**NOTES:**

1. When the G1PO0 register resets the base timer, the channel 0 and 1 SR waveform generating functions are not available.
2. When the INV bit in the G1POCRj register is set to "1" (output inversed), the "L" width and "H" width are inversed.
3. Waveform from base timer reset until when output level becomes "H".
4. Waveform from when output level becomes "L" until base timer reset.

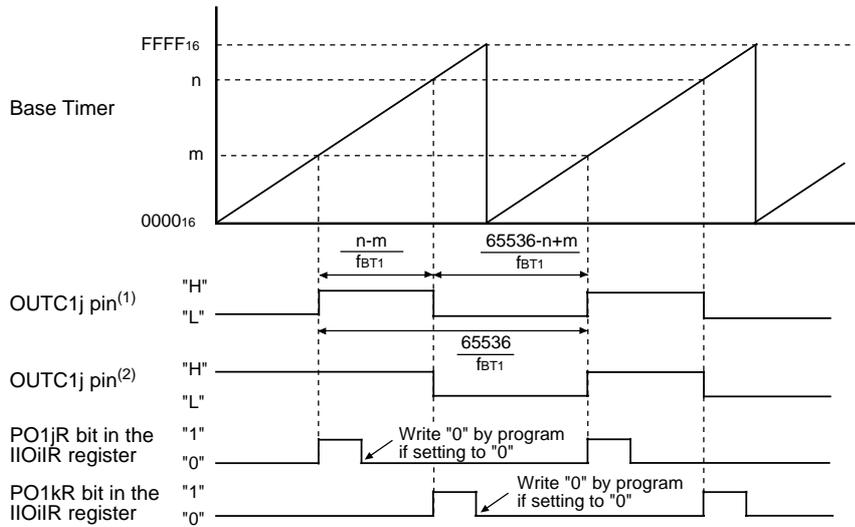
**Table 21.11 SR Waveform Output Mode Specifications (Continued)**

Item	Specification
Waveform Output Start Condition <sup>(5)</sup>	The IFEq bit (q=0 to 7) in the G1FE register is set to "1" (channel q function enabled)
Waveform Output Stop Condition	The IFEq bit is set to "0" (channel q function disabled)
Interrupt Request	The PO1jR bit in the interrupt request register is set to "1" (interrupt requested) when the value of the base timer matches that of the G1POj register. The PO1kR bit in the interrupt request register is set to "1" (interrupt requested) when the value of the base timer matches that of the G1POk register. (See Figure 10.14)
OUTC1j Pin	Pulse signal output pin
Selectable Function	<ul style="list-style-type: none"> <li>• Default value set function: Set starting waveform output level</li> <li>• Inversed output function</li> </ul> <p>Waveform output level is inversed to provide a waveform from the OUTC1j pin</p>

**NOTE:**

5. Set the FSCj bit in the G1FS register to "0" (waveform generating function selected).

(1) Free-Running Operation  
 (The RST2 to RST0 bits in the G1BCR1 register are set to "002")



$i=0$  to 4, 8 to 10;  $j=0, 2, 4, 6$ ;  $k=j+1$   
 $m$ : Setting value of the G1POj register,  $0000_{16}$  to  $FFFF_{16}$   
 $n$ : Setting value of the G1POk register,  $0000_{16}$  to  $FFFF_{16}$

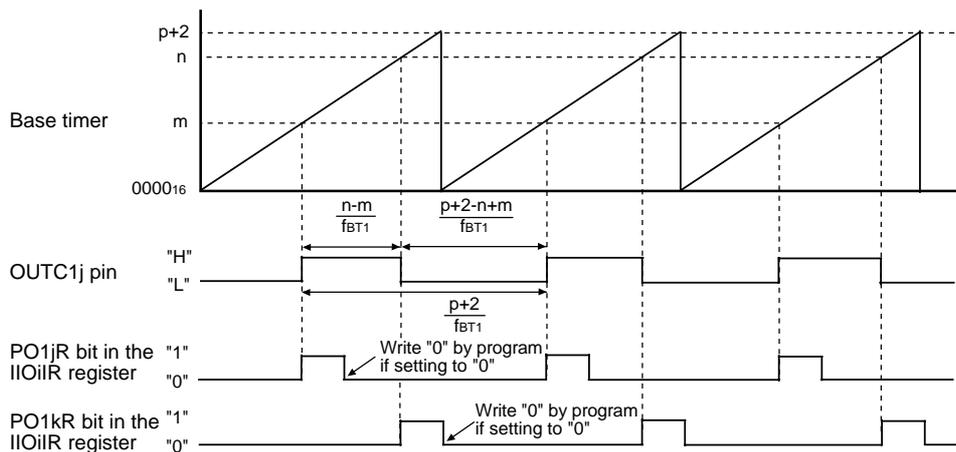
NOTES:

- Waveform output when the INV bit in the G1POCRj register is set to "0" (not inverted) and the IVL bit in the G1POCRj register is set to "0" (output "L" as default value).
- Waveform output when the INV bit is set to "0" (not inverted) and the IVL bit is set to "1" ("H" output as default value).

The above applies under the following conditions:

- The RST2 and RST1 bits in the G1BCR1 register are set to "002" (no base timer reset) and the UD1 and UD0 bits in the G1BCR1 register to "002" (counter increment mode).
- $m < n$

(2) The Base Timer is Reset when the Base Timer Matches the G1PO0 Register  
 (The RST1 bit is set to "1" and the RST2 bit is set to "0")



$i=0$  to 4, 8 to 10;  $j=2, 4, 6$ ;  $k=j+1$   
 $m$ : Setting value of the G1POj register,  $0000_{16}$  to  $FFFF_{16}$   
 $n$ : Setting value of the G1POk register,  $0000_{16}$  to  $FFFF_{16}$   
 $p$ : Setting value of the G1PO0 register,  $0001_{16}$  to  $FFFD_{16}$

The above applies under the following conditions:

- The IVL bit in the G1POCRj register is set to "0" ("L" output as default value) and the INV bit is set to "0" (not inverted).
- The UD1 and UD0 bits are set to "002" (counter increment mode).
- $m < n < p+2$

Figure 21.18 SR Waveform Output Mode

### 21.4 Communication Unit 0 and 1 Communication Function

In the intelligent I/O communication unit 1, 8-bit clock synchronous serial I/O, 8-bit clock asynchronous serial I/O (UART) or HDLC data processing is available. In the communication unit 0, 8-bit clock synchronous serial I/O or HDLC data processing is available.

Figures 21.19 to 21.28 show registers associated with the communication function.

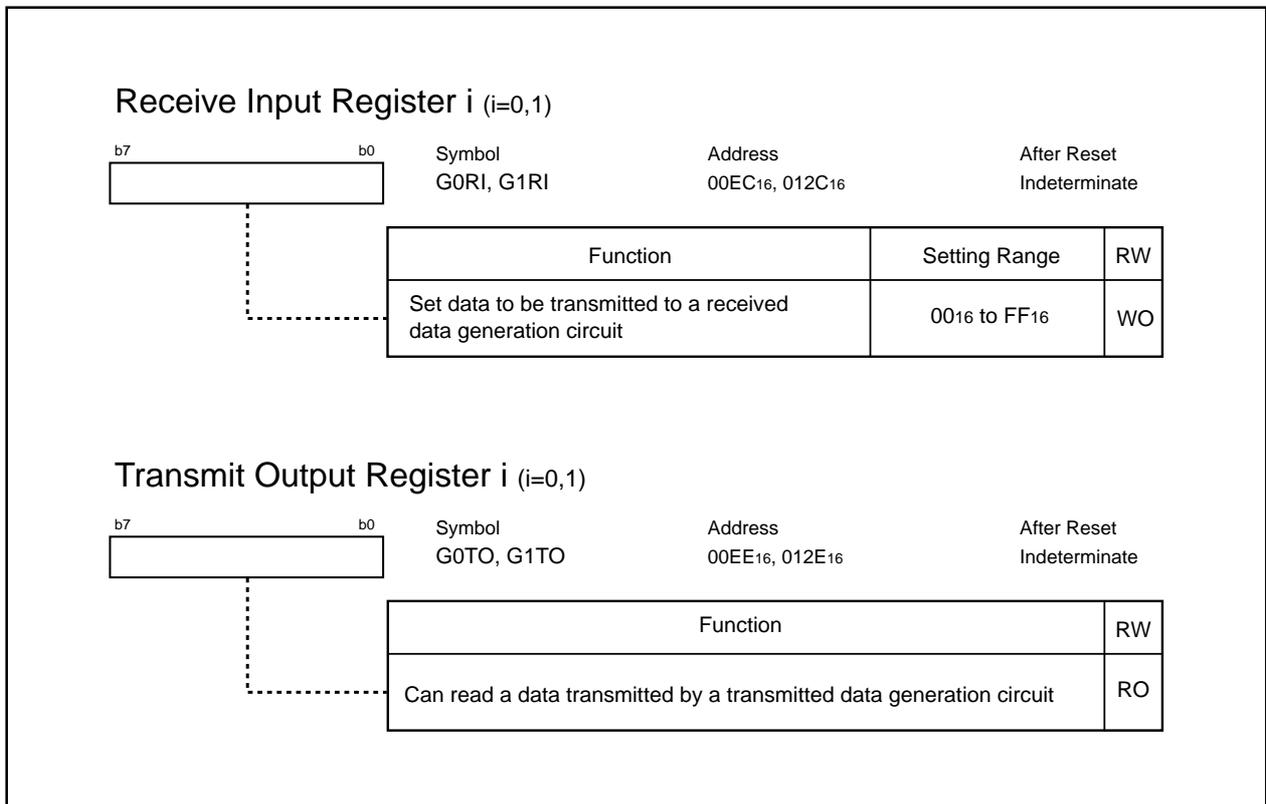
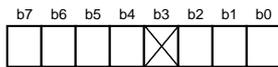


Figure 21.19 G0RI and G1RI Registers, G0TO and G1TO Registers

### SI/O Communication Control Register i (i=0, 1)



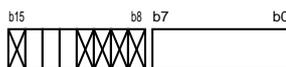
Symbol: G0CR, G1CR      Address: 00EF<sub>16</sub>, 012F<sub>16</sub>      After Reset: 0000 X011<sub>2</sub>

Bit Symbol	Bit Name	Function	RW
TI	Transmit Buffer Empty Flag	0: Data in the GiTB register 1: No data in the GiTB register	RO
TXEPT	Transmit Register Empty Flag	0: Data in the transmit register (during transmission) 1: No data in the transmit register (transmit completed)	RO
RI	Receive Complete Flag	0: No data in the GiRB register 1: Data in the GiRB register	RO
— (b3)	Nothing is assigned. When write, set to "0". When read, its contents is indeterminate.		—
TE	Transmit Enable Bit	0: Transmit disabled 1: Transmit enabled	RW
RE	Receive Enable Bit	0: Receive disabled 1: Receive enabled	RW
IPOL	ISRxD Input Polarity Switch Bit	0: No inverse 1: Inverse <sup>(1)</sup>	RW
OPOL	ISTxD Output Polarity Switch Bit	0: No inverse 1: Inverse <sup>(1)</sup>	RW

NOTE:

- Set this bit to "1" when using UART mode.

### SI/O Receive Buffer Register i (i=0, 1)



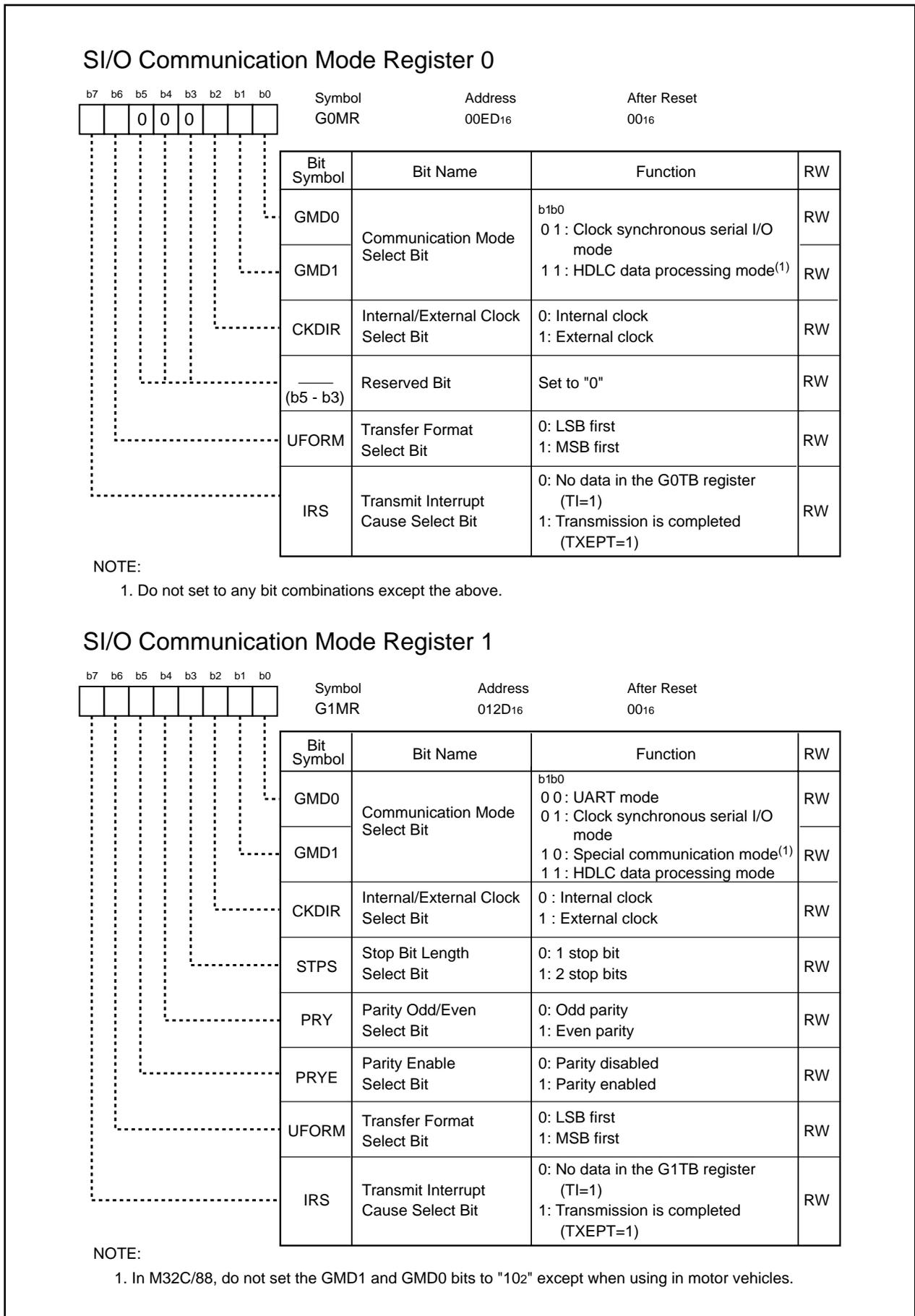
Symbol: G0RB, G1RB      Address: 00E9<sub>16</sub>-00E8<sub>16</sub>, 0129<sub>16</sub>-0128<sub>16</sub>      After Reset: X000 XXXX XXXX XXXX<sub>2</sub>

Bit Symbol	Bit Name	Function	RW
— (b7 - b0)	—	Received data	RW
— (b11 - b8)	Nothing is assigned. When read, its content is indeterminate.		—
OER	Overrun Error Flag	0: No overrun error 1: Overrun error found	RO
FER	Framing Error Flag <sup>(1)</sup>	0: No framing error 1: Framing error found	RO
PER	Parity Error Flag <sup>(1)</sup>	0: No parity error 1: Parity error found	RO
— (b15)	Nothing is assigned. When read, its content is indeterminate.		—

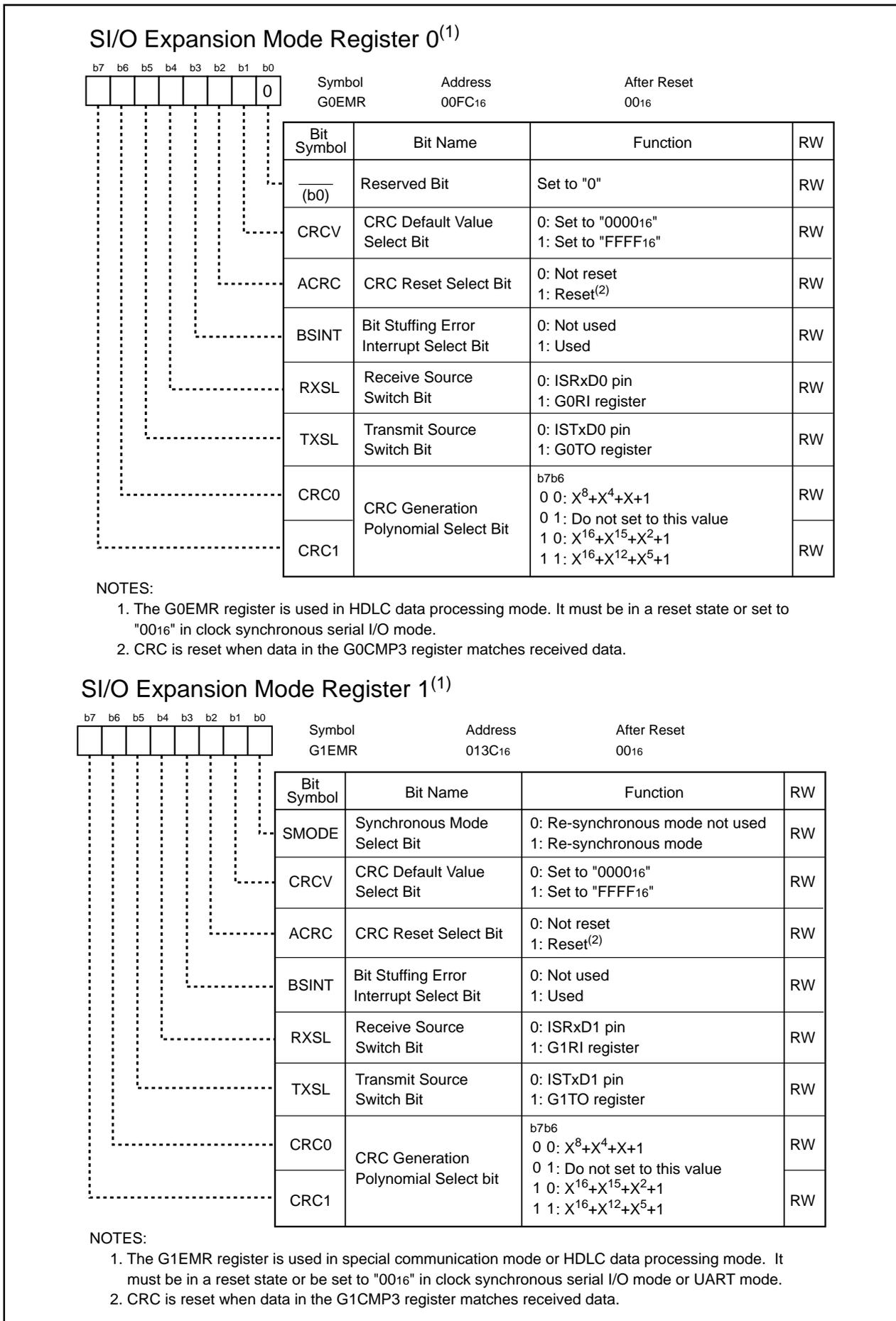
NOTE:

- Nothing is assigned in the FER and PER bits in the G0RB register. When read, its content is indeterminate.

Figure 21.20 G0CR and G1CR Registers, G0RB and G1RB Registers



**Figure 21.21 G0MR and G1MR Registers**



**Figure 21.22 G0EMR and G1EMR Registers**

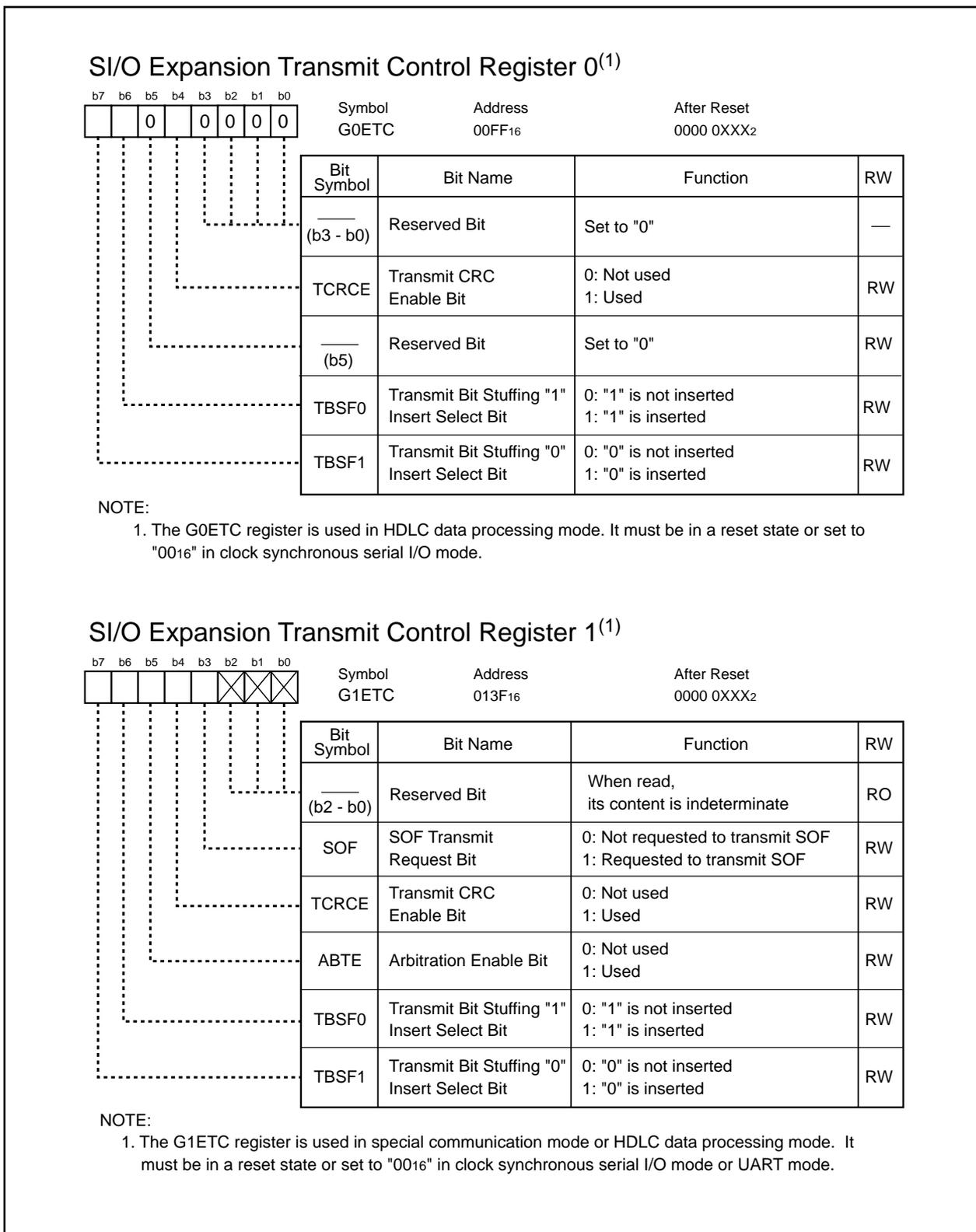
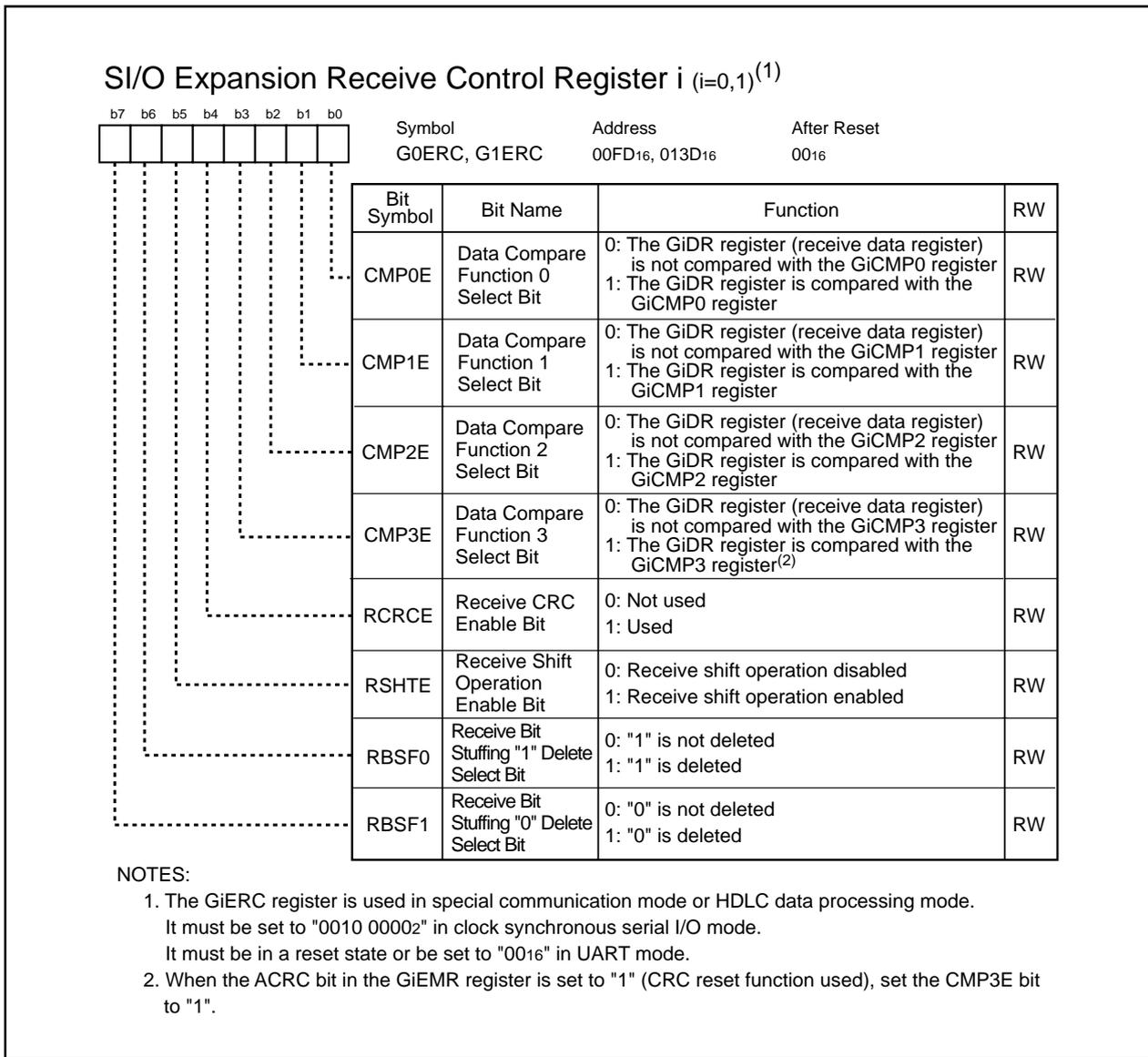
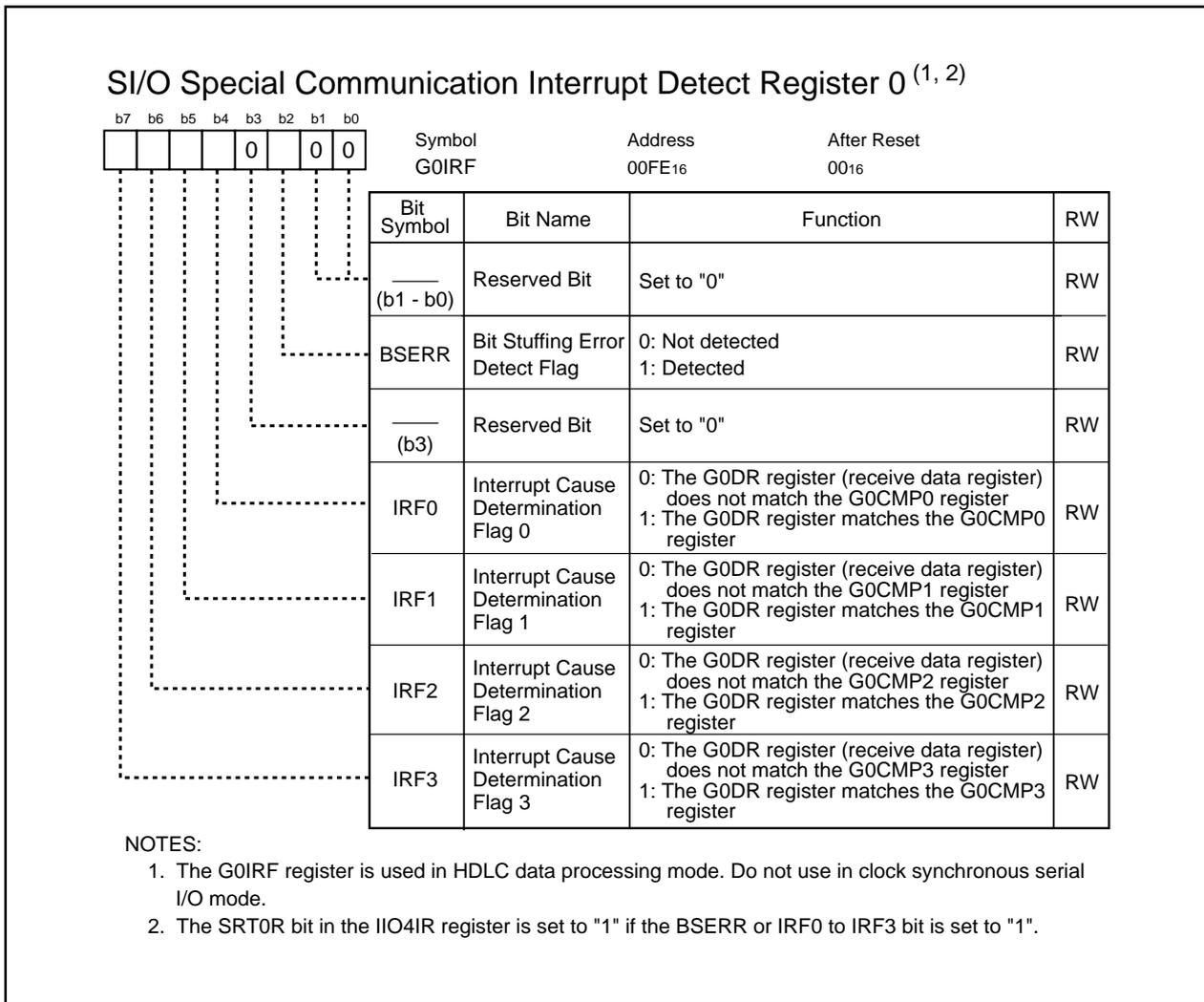


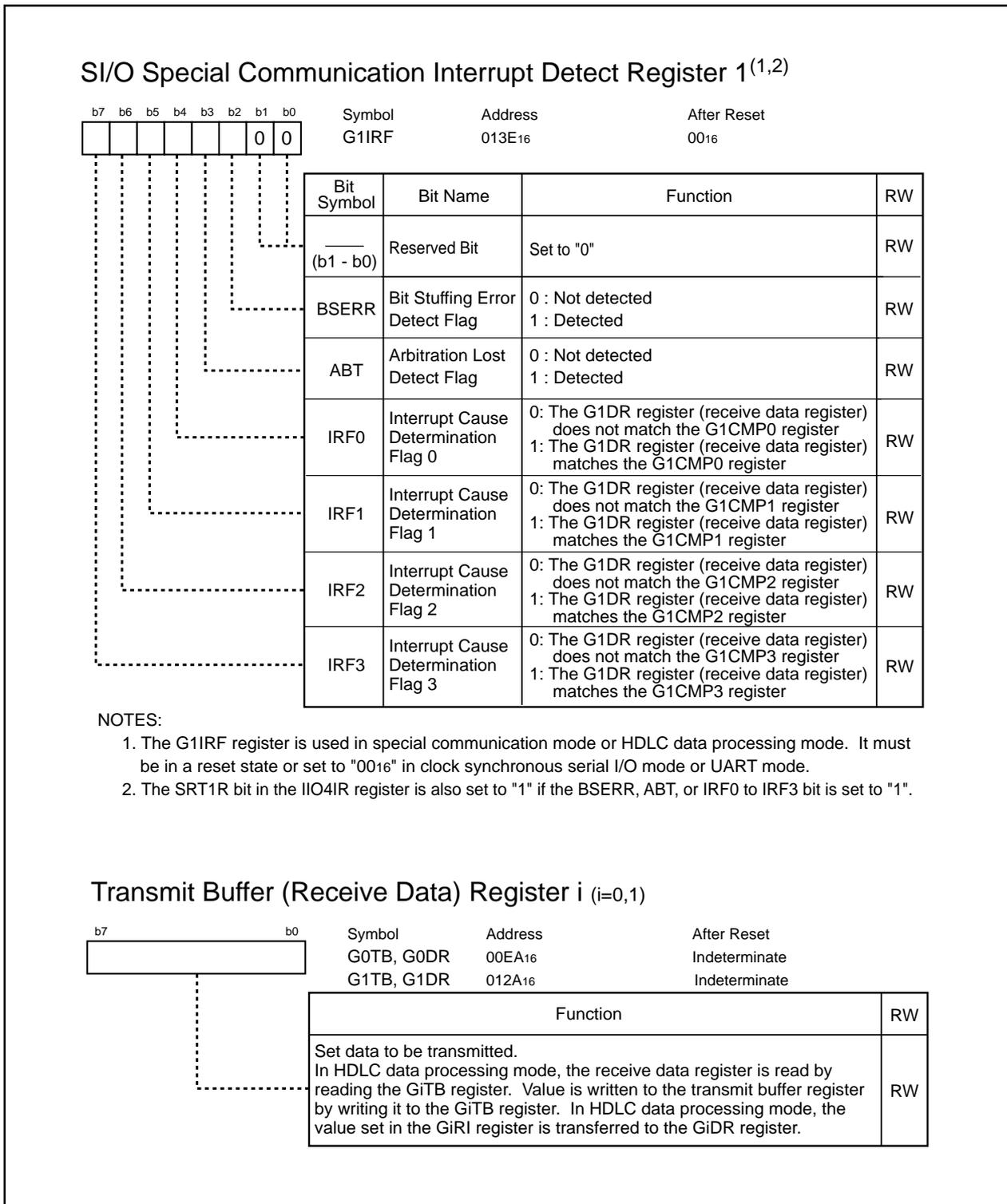
Figure 21.23 G0ETC and G1ETC Registers



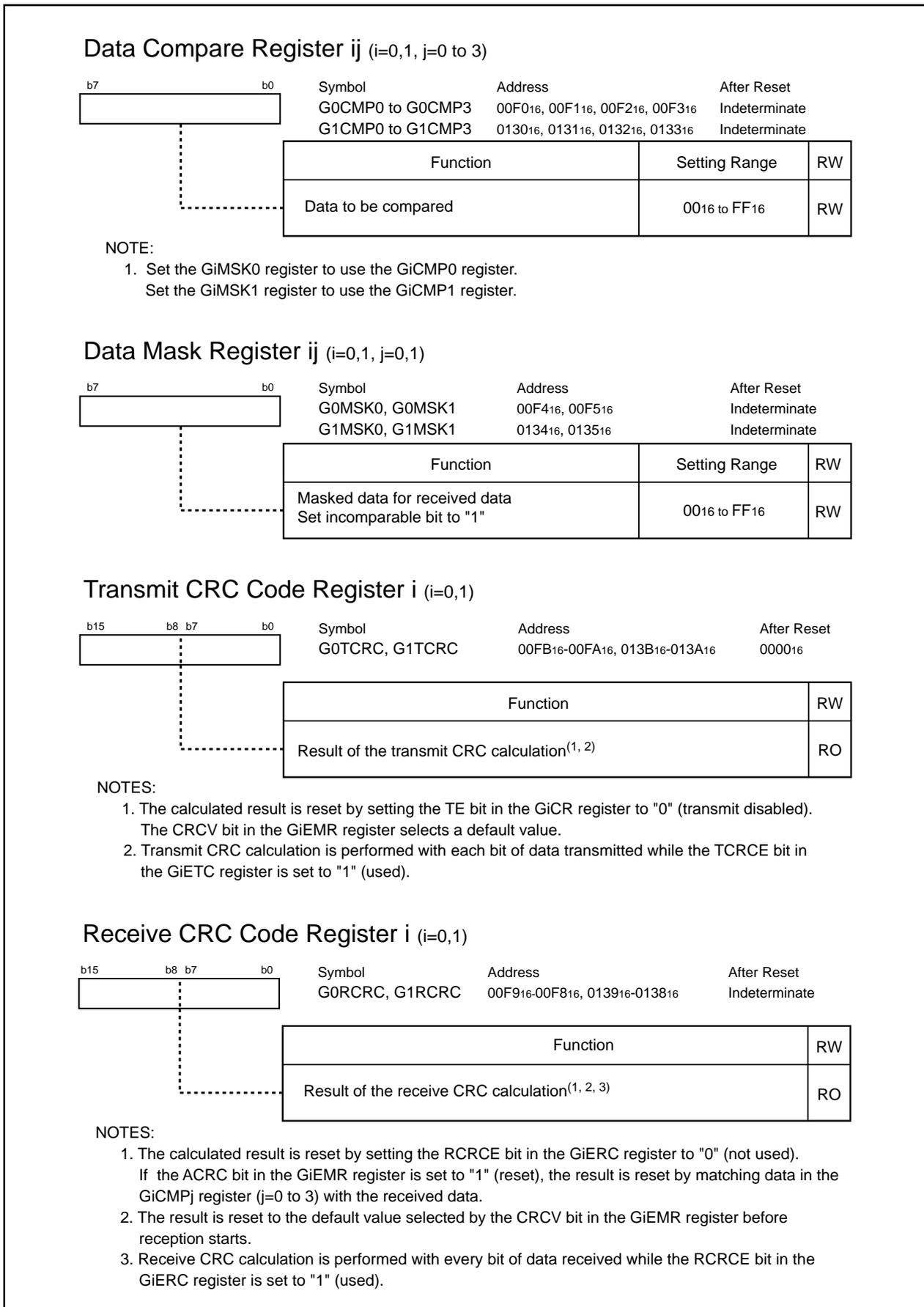
**Figure 21.24 G0ERC and G1ERC Registers**



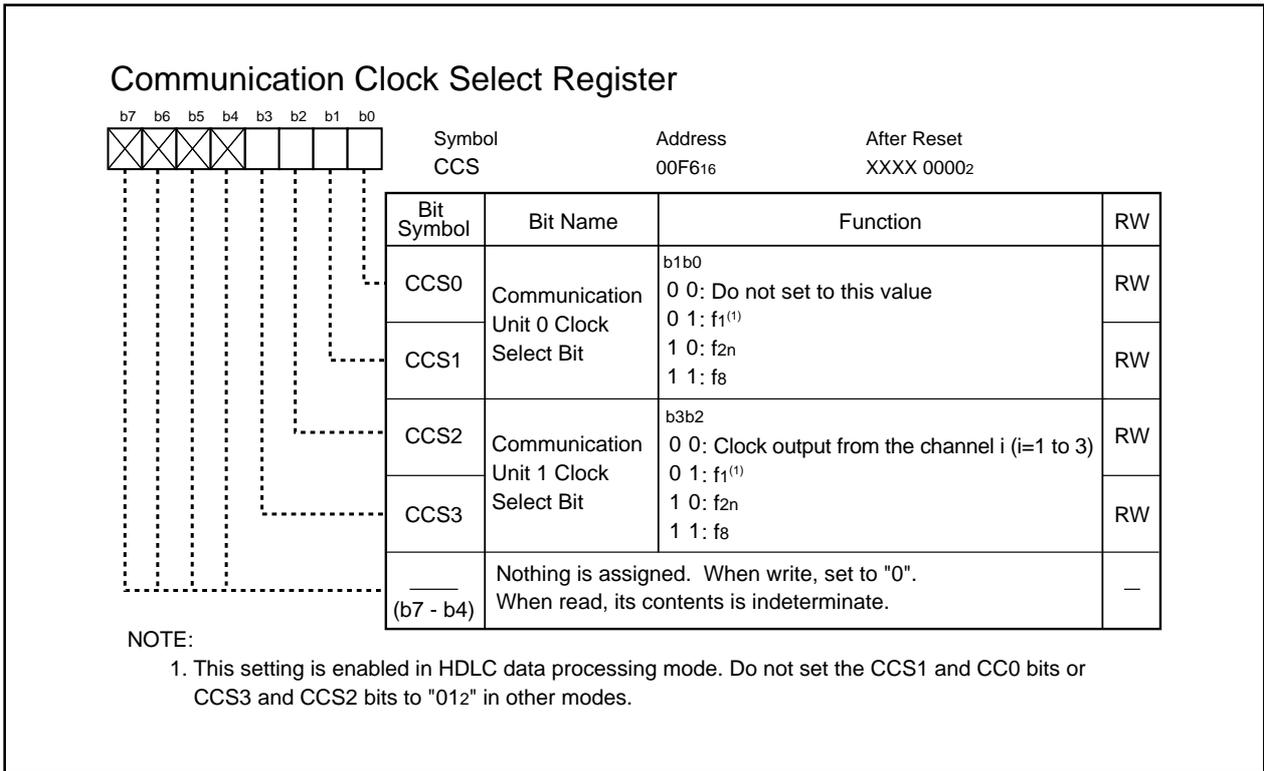
**Figure 21.25 G0IRF Register**



**Figure 21.26 G1IRF Register, G0TB and G1TB / G0DR and G1DR Registers**



**Figure 21.27 G0CMP0 to G0CMP3 Registers and G1CMP0 to G1CMP3 Registers  
G0MSK0 and G0MSK1 Registers, G1MSK0 and G1MSK1 Registers  
G0TCRC and G1TCRC Registers, G0RCRC and G1RCRC Registers**



**Figure 21.28 CCS Register**

### 21.4.1 Clock Synchronous Serial I/O Mode (Communication Units 0 and 1)

In clock synchronous serial I/O mode, data is transmitted and received with the transfer clock.  $f_8$  or  $f_{2n}$  can be selected as the communication unit 0 transfer clock.  $f_8$ ,  $f_{2n}$  or the clock generated in channels 0 and 3 can be selected as the communication unit 1 transfer clock.

Table 21.12 lists specifications of clock synchronous serial I/O mode for the communication units 0 and 1. Tables 21.13 and 21.14 list clock settings. Table 21.15 lists register settings. Tables 21.16 to 21.19 list pin settings. Figure 21.29 shows an example of transmit and receive operation.

**Table 21.12 Clock Synchronous Serial I/O Mode Specifications (Communication Units 0 and 1)**

Item	Specification
Transfer Data Format	Transfer data : 8 bits long
Transfer Clock <sup>(1)</sup>	See Tables 21.13 and 21.14
Transmit Start Condition	Set registers associated with the waveform generating function, the GiMR and GiERC registers ( $i=0,1$ ). Then, set as is written below after at least one transfer clock cycle. <ul style="list-style-type: none"> <li>• Set the TE bit in the GiCR register to "1" (transmit enabled)</li> <li>• Set the TI bit in the GiCR register to "0" (data in the GiTB register)</li> </ul>
Receive Start Condition	Set registers associated with the waveform generating function, the GiMR and GiERC registers. Then, set as is written below after at least one transfer clock cycle. <ul style="list-style-type: none"> <li>• Set the RE bit in the GiCR register to "1" (receive enabled)</li> <li>• Set the TE bit to "1" (transmit enabled)</li> <li>• Set the TI bit to "0" (data in the GiTB register)</li> </ul>
Interrupt Request	<ul style="list-style-type: none"> <li>• While transmitting, one of the following conditions can be selected to set the SIOiTR bit to "1" (interrupt requested) (see Figure 11.14) : <ul style="list-style-type: none"> <li>– The IRS bit in the GiMR register is set to "0" (no data in the GiTB register) and data is transferred to the transmit register from the GiTB register</li> <li>– The IRS bit is set to "1" (transmission completed) and data transfer from the transmit register is completed</li> </ul> </li> <li>• While receiving, the following condition can be selected to set SIOiRR bit is set to "1" (data reception is completed): <ul style="list-style-type: none"> <li>– Data is transferred from the receive register to the GiRB register</li> </ul> </li> </ul>
Error Detection	<p>Overrun error<sup>(2)</sup></p> <p>This error occurs, when the next data reception is started and the 8th bit of the next data is received before reading the GiRB register</p>
Selectable Function	<ul style="list-style-type: none"> <li>• LSB first or MSB first <ul style="list-style-type: none"> <li>– Select either bit 0 or bit 7 to transmit or receive data</li> </ul> </li> <li>• ISTxDi and ISRxDi I/O polarity inverse <ul style="list-style-type: none"> <li>– ISTxDi pin output level and ISRxDi pin input level are inverted</li> </ul> </li> </ul>

**NOTES:**

1. In clock synchronous serial I/O mode, set the RSHTTE bit in the GiERC register ( $i=0, 1$ ) to "1" (receive shift operation enabled).
2. When an overrun error occurs, the GiRB register is indeterminate.

When the OPOL bit in the GiCR register is set to "0" (ISTxD output polarity not inverted), the ISTxDi pin puts in a high-level ("H") signal output after selecting operating mode until transfer starts. When the OPOL bit is set to "1" (ISTxD output polarity inverted), the ISTxDi pin puts in a low-level ("L") signal output.

**Table 21.13 Clock Settings (Communication Unit 0)**

Transfer Clock	GOMR Register		CCS Register	
	CKDIR Bit		CCS0 Bit	CCS1 Bit
$f_8$	0		1	1
$f_{2n}^{(1)}$	0		0	1
Input from ISCLK0	1		-	-

**NOTE:**

1. The CNT3 to CNT0 bits in the TCSPR register select no division ( $n=0$ ) or divide-by- $2n$  ( $n=1$  to 15).

**Table 21.14 Clock Settings (Communication Unit 1)**

Transfer Clock <sup>(3)</sup>	G1MR Register	CCS Register	
	CKDIR Bit	CCS2 Bit	CCS3 Bit
$\frac{f_{BT1}}{2(n+2)}$ <sup>(1)</sup>	0	0	0
$f_8$	0	1	1
$f_{2n}$ <sup>(2)</sup>	0	0	1
Input from ISCLK1	1	-	-

$n$ : Setting value of the G1PO0 register, 000116 to FFFD16

**NOTES:**

1. The transfer clock is generated in phase-delayed waveform output mode of the channel 3 waveform generating function.
2. The CNT3 to CNT0 bits in the TCSPR register select no division ( $n=0$ ) or divide-by- $2n$  ( $n=1$  to 15).
3. The transfer clock must be  $f_{BT1}$  divided by six or more.

**Table 21.15 Register Settings in Clock Synchronous Serial I/O Mode (Communication Units 0 and 1)**

Register	Bit	Function	
		Communication Unit 1	Communication Unit 0
CCS	CCS1, CCS0	Setting not required when using the communication unit 1 only	Select transfer clock
	CCS3, CSS2	Select transfer clock	Setting not required when using the communication unit 0 only
G1BCR0 <sup>(2)</sup>	BCK1, BCK0	Set to "112" ( $f_1$ )	
	DIV4 to DIV0	Select divide ratio of count source	
	IT	Set to "0"	
G1BCR1 <sup>(2)</sup>	7 to 0	Set to "0001 00102"	
G1POCR0 <sup>(2)</sup>	7 to 0	Set to "0000 01112"	
G1POCR1 <sup>(2)</sup>	7 to 0	Set to "0000 01112"	
G1POCR3 <sup>(2)</sup>	MOD2 to MOD0	Set to "0102" <sup>(1)</sup>	
	IVL	Select default ISCLKi output value <sup>(1)</sup>	
	RLD	Set to "0"	
	INV	Select whether ISCLKi puts in an inversed signal or not <sup>(1)</sup>	
G1PO0 <sup>(2)</sup>	15 to 0	Set bit rate $\frac{f_{BT1}}{2 \times (\text{setting value} + 2)} = \text{transfer clock frequency}$	
G1PO3 <sup>(2)</sup>	15 to 0	Set to a value smaller than the G1PO0 register <sup>(1)</sup>	
G1FS <sup>(2)</sup>	FSC3,FSC1,FSC0	Set to "0" <sup>(1)</sup>	
G1FE <sup>(2)</sup>	IFE3,IFE1,IFE0	Set to "1" <sup>(1)</sup>	
GiERC	7 to 0	Set to "0010 00002"	
GiMR	GMD1, GMD0	Set to "012"	
	CKDIR	Select the internal clock or external clock	
	STPS	Set to "0"	
	UFORM	Select either LSB first or MSB first	
	IRS	Select what cause the transmit interrupt to be generated	
GiCR	TI	Transmit buffer empty flag	
	TXEPT	Transmit register empty flag	
	RI	Receive complete flag	
	TE	Set to "1" to enable transmission and reception	
	RE	Set to "1" to enable reception	
	IPOL	Select ISRxDi input polarity (usually set to "0")	
	OPOL	Select ISTxDi output polarity (usually set to "0")	
GiTB	–	Write data to be transmitted	
GiRB	–	Received data and error flag are stored	

$i = 0$  to 1

**NOTES:**

1. The CKDIR bit in the GiMR register is set to "0" (internal clock).
2. These registers must be set, when  $f_8$  or  $f_{2n}$  is selected as transfer clock source notwithstanding.

**Table 21.16 Pin Settings in Clock Synchronous Serial I/O Mode (Communication Units 0 and 1)(1)**

Port Name	Function	Setting						Register <sup>(1)</sup>
		PS1 Register	PSL1 Register	PSC Register	PSD1 Register	PD7 Register	IPS Register	
P73	ISTxD1 Output	PS1_3=1	PSL1_3=0	PSC_3=1	-	-	-	G1POCR0
P74	ISCLK1 Input	PS1_4=0	-	-	-	PD7_4=0	IPS1=0	-
	ISCLK1 Output	PS1_4=1	PSL1_4=0	PSC_4=1	-	-	-	G1POCR1
P75	ISRxD1 Input	PS1_5=0	-	-	-	PD7_5=0	IPS1=0	-
p76	ISTxD0 Output	PS1_6=1	PSL1_6=0	PSC_6=0	PSD1_6=0	-	-	-
p77	ISCLK0 Input	PS1_7=0	-	-	-	PD7_7=0	IPS0=0	-
	ISCLK0 Output	PS1_7=1	PSL1_7=0	-	-	-	-	-

**NOTE:**

1. Set the MOD2 to MOD0 bits in the corresponding register to "1112" (output from the communication function used).

**Table 21.17 Pin Settings (2)**

Port Name	Function	Setting		
		PS2 Register	PD8 Register	IPS Register
P80	ISRxD0 input	PS2_0 = 0	PD8_0 = 0	IPS0 = 0

**Table 21.18 Pin Settings (3)**

Port Name	Function	Setting			Register <sup>(1)</sup>
		PS5 Register	PD11 Register	IPS Register	
P110	ISTxD1 output	PS5_0 = 1	-	-	G1POCR0
P111	ISCLK1 input	PS5_1 = 0	PD11_1 = 0	IPS1 = 1	-
	ISCLK1 output	PS5_1 = 1	-	-	G1POCR1
P112	ISRxD1 input	PS5_2 = 0	PD11_2 = 0	IPS1 = 1	-

**NOTE:**

1. Set the MOD2 to MOD0 bits in the corresponding register to "1112" (output from communication function used).

**Table 21.19 Pin Settings (4)**

Port Name	Function	Setting		
		PS9 Register	PD15 Register	IPS Register
P150	ISTxD0 output	PS9_0 = 1	-	-
P151	ISCLK0 input	PS9_1 = 0	PD15_2 = 0	IPS0 = 1
	ISCLK0 output	PS9_1 = 1	-	-
P152	ISRxD0 input	-	PD15_2 = 0	IPS0 = 1

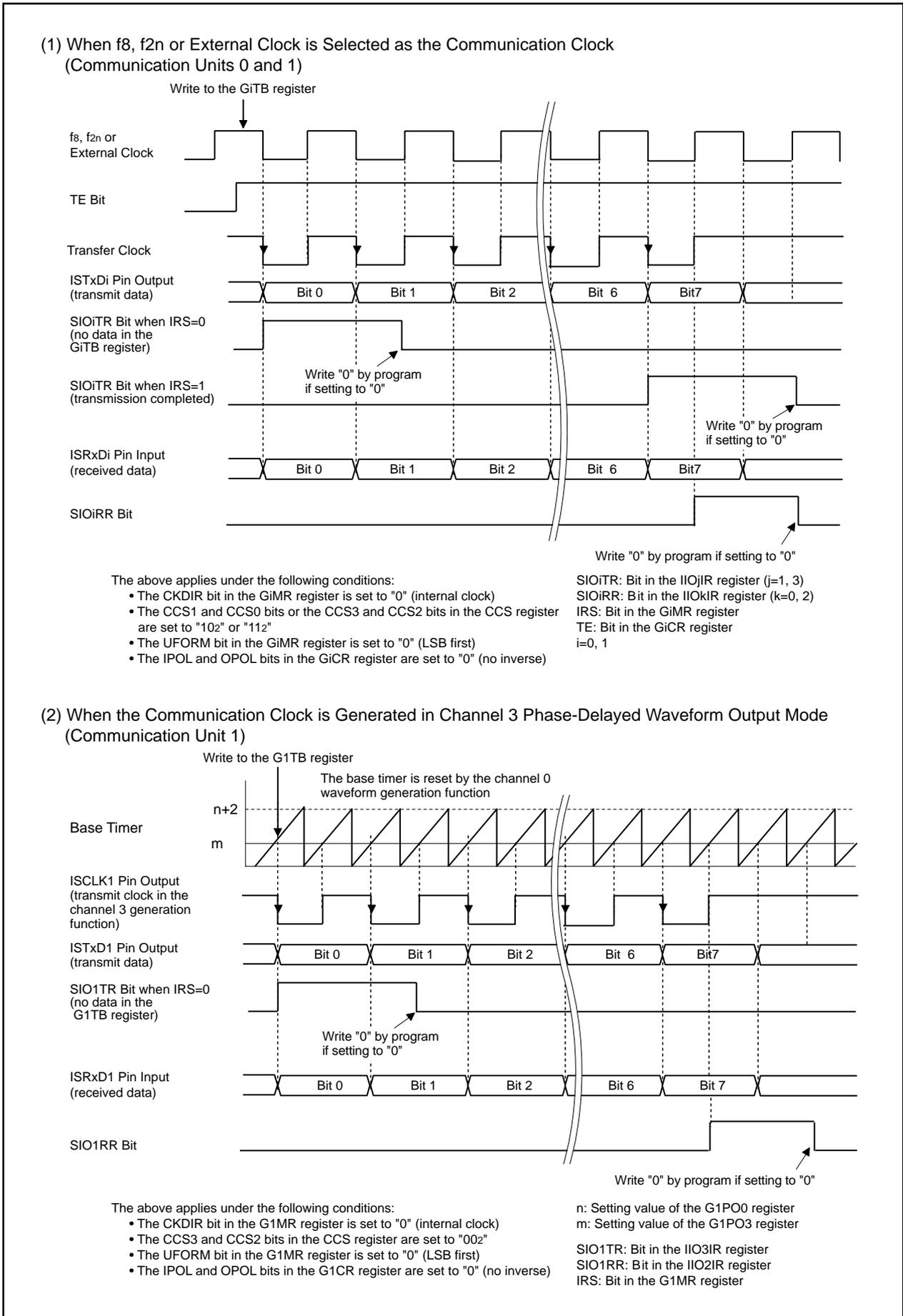


Figure 21.29 Transmit and Receive Operation

### 21.4.2 Clock Asynchronous Serial I/O (UART) Mode (Communication Unit 1)

In clock asynchronous serial I/O (UART) mode, data is transmitted at a desired bit rate and in a desired transfer data format. Table 21.20 lists specifications of UART mode in the communication unit 1. Table 21.21 lists clock settings. Table 21.22 lists register settings. Tables 21.23 and 21.24 list pin settings. Figure 21.30 shows an example of transmit operation. Figure 21.31 shows an example of receive operation.

**Table 21.20 UART Mode Specifications (Communication Unit 1)**

Item	Specification
Transfer Data Format	<ul style="list-style-type: none"> <li>• Character bit (transfer data) : 8 bits long</li> <li>• Start bit : 1 bit long</li> <li>• Parity bit: selected from odd, even, or none</li> <li>• Stop bit : selected length from 1 bit or 2 bits</li> </ul>
Transfer Clock <sup>(1)</sup>	See Table 21.21
Transmit Start Condition	Set registers associated with the waveform generating function, the G1MR and G1ERC registers. Then, set as is written below after at least one transfer clock cycle: <ul style="list-style-type: none"> <li>• Set the TE bit in the G1CR register to "1" (transmit enabled)</li> <li>• Set the TI bit in the G1CR register to "0" (data written to the G1TB register)</li> </ul>
Receive Start Condition	Set registers associated with the waveform generating function, the G1MR and G1ERC registers. Then, set as is written below after at least one transfer clock cycle: <ul style="list-style-type: none"> <li>• Set the RE bit in the G1CR register to "1" (receive enabled)</li> <li>• Detect the start bit</li> </ul>
Interrupt Request	<ul style="list-style-type: none"> <li>• While transmitting, one of the following conditions can be selected to set the SIO1TR bit to "1" (interrupt requested) (See Figure 10.14.) :               <ul style="list-style-type: none"> <li>– The IRS bit in the G1MR register is set to "0" (no data in the G1TB register) and data is transferred to the transmit register from the G1TB register.</li> <li>– The IRS bit is set to "1" (transmission completed) and data transfer from the transmit register is completed</li> </ul> </li> <li>• While receiving, the following condition can be selected to set the SIO1RR bit is set to "1": Data is transferred from the receive register to the G1RB register (data reception is completed)</li> </ul>
Error Detection	<ul style="list-style-type: none"> <li>• Overrun error<sup>(2)</sup> This error occurs, when the next data reception is started and the final stop bit of the next data is received before reading the G1RB register</li> <li>• Parity error While parity is enabled, this error occurs when the number of "1" in parity and character bits does not match the number of "1" set</li> <li>• Framing error This error occurs when the number of the stop bits set is not detected</li> </ul>
Selectable Function	<ul style="list-style-type: none"> <li>• Stop bit length The length of the stop bit is selected from 1 bit or 2 bits</li> <li>• LSB first or MSB first Select either bit 0 or bit 7 to transmit or receive data</li> </ul>

**NOTES:**

1. The transfer clock must be fBT1 divided by six or more.
2. When an overrun error occurs, the G1RB register is indeterminate.

**Table 21.21 Clock Settings (Communication Unit 1)**

Transfer Clock <sup>(3)</sup>	G1MR Register			CCS Register		
	CKDIR Bit			CCS2 Bit		CCS3 Bit
$\frac{f_{BT1}}{2(n+2)}$ <sup>(1, 2)</sup>	0			0		0

*n*: Setting value of the G1PO0 register 0001<sub>16</sub> to FFFD<sub>16</sub>

**NOTES:**

1. Transmit clock is generated in phase-delayed waveform output mode of the channel 3 waveform generating function.
2. Received clock is generated when phase-delayed waveform mode of the channel 2 waveform generating function and the channel 2 time measurement function is simultaneously performed.
3. The transfer clock must be *f*<sub>BT1</sub> divided by six or more.

**Table 21.22 Register Settings in UART Mode (Communication Unit 1)**

Register	Bit	Function
G1BCR0	BCK1, BCK0	Set to "112" (f <sub>1</sub> )
	DIV4 to DIV0	Select divide ratio of count source
	IT	Set to "0"
G1BCR1	7 to 0	Set to "0001 0010 <sub>2</sub> "
G1POCR0	7 to 0	Set to "0000 0111 <sub>2</sub> "
G1POCR2	7 to 0	Set to "0000 0110 <sub>2</sub> "
G1POCR3	7 to 0	Set to "0000 0010 <sub>2</sub> "
G1TMCR2	7 to 0	Set to "0000 0010 <sub>2</sub> "
G1PO0	15 to 0	Set bit rate $\frac{f_{BT1}}{2 \times (\text{setting value} + 2)} = \text{transfer clock frequency}$
G1PO3	15 to 0	Set to a value smaller than the G1PO0 register
G1FS	FSC3 to FSC0	Set to "0100 <sub>2</sub> "
G1FE	IFE3 to IFE0	Set to "1101 <sub>2</sub> "
G1MR	GMD1, GMD0	Set to "00 <sub>2</sub> "
	CKDIR	Set to "0"
	STPS	Select stop bit length
	PRY, PRYE	Select either parity enabled or disabled and either odd parity or even parity
	UFORM	Select either the LSB first or MSB first
	IRS	Select what causes the receive interrupt to be generated
G1CR	TI	Transmit buffer empty flag
	TXEPT	Transmit register empty flag
	RI	Receive complete flag
	TE	Set to "1" to enable transmission and reception
	RE	Set to "1" to enable reception
	IPOL	Set to "1"
OPOL	Set to "1"	
G1TB	7 to 0	Write data to be transmitted
G1RB	15 to 0	Received data and error flag are stored
CCS	CCS3, CCS2	Set to "00 <sub>2</sub> "

**Table 21.23 Pin Settings in UART Mode**

Port Name	Function	Setting					Register <sup>(1)</sup>
		PS1 Register	PSL1 Register	PSC Register	PD7 Register	IPS Register	
P7 <sub>3</sub>	ISTxD1 output	PS1_3 = 1	PSL1_3 = 0	PSC_3 = 1	-	-	G1POCR0
P7 <sub>5</sub>	ISRxD1 input	PS1_5 = 0	-	-	PD7_5 = 0	IPS1 = 0	-

**NOTE:**

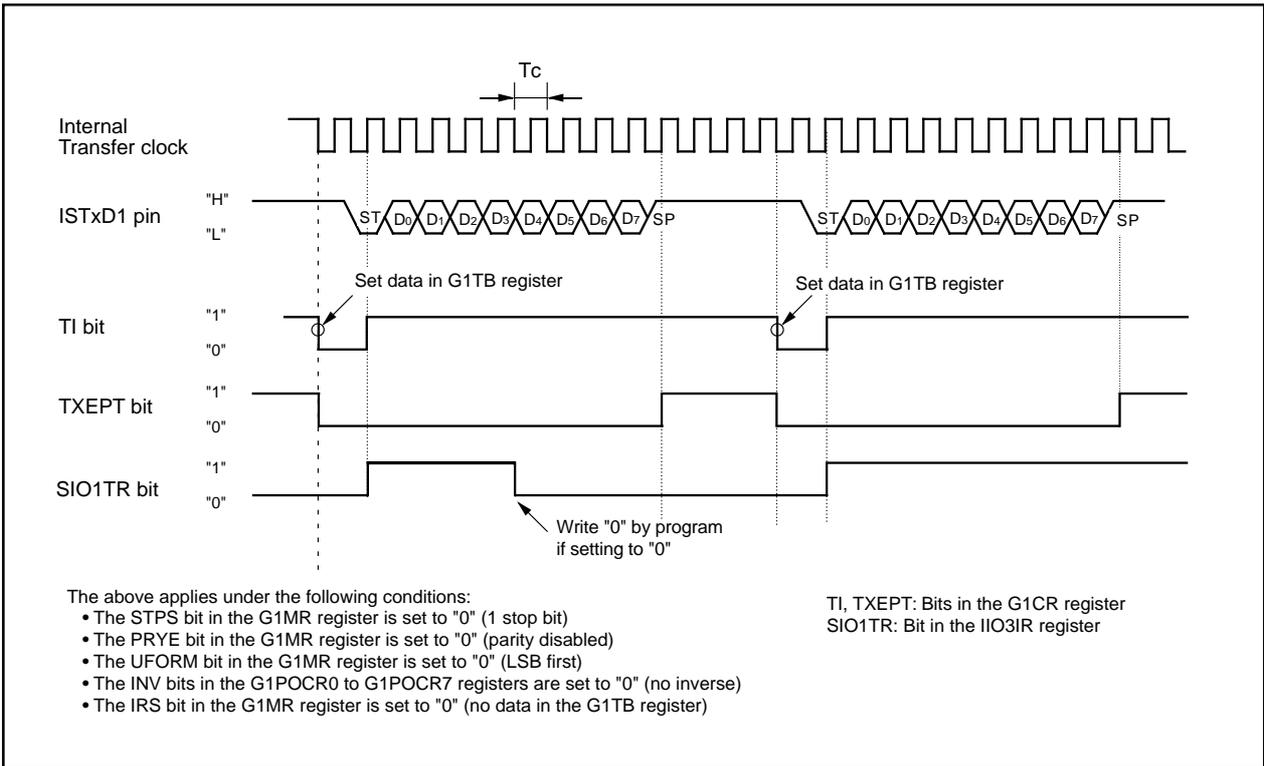
1. Set the MOD2 to MOD0 bits in the corresponding register to "111<sub>2</sub>" (output from communication function used).

**Table 21.24 Pin Settings (Continued)**

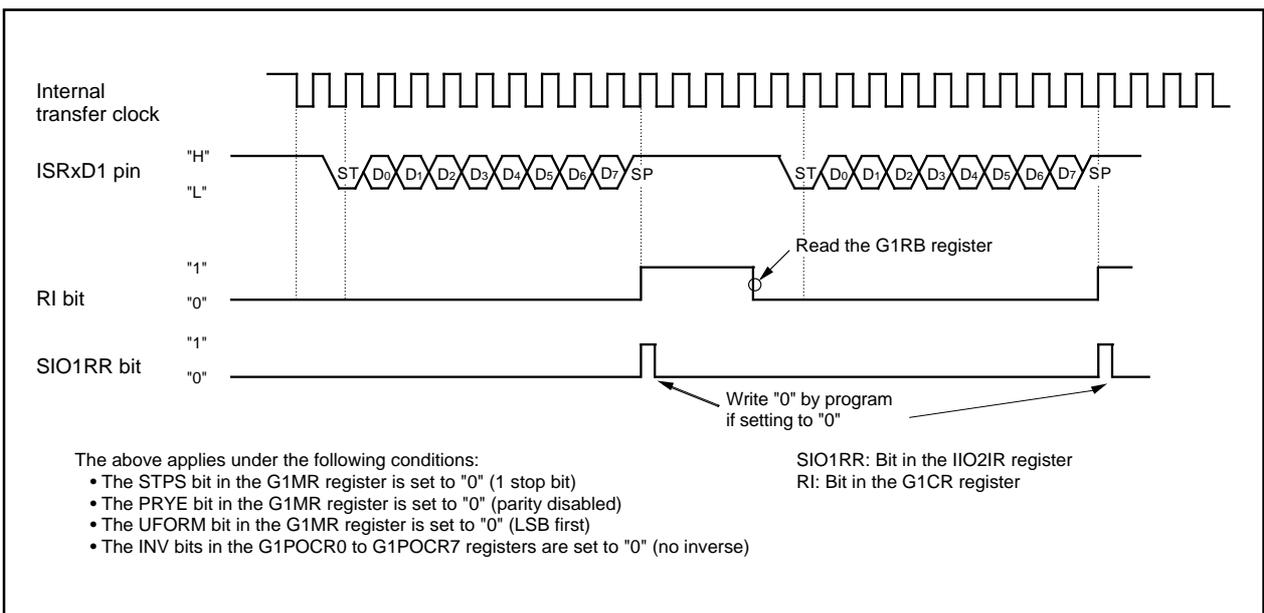
Port Name	Function	Setting			Register <sup>(1)</sup>
		PS5 Register	PD11 Register	IPS Register	
P110	ISTxD1 output	PS5_0 = 1	-	-	G1POCR0
P112	ISRxD1 input	PS5_2 = 0	PD11_2 = 0	IPS1 = 1	-

**NOTE:**

1. Set the MOD2 to MOD0 bits in the corresponding register to "1112" (output from the communication function used).



**Figure 21.30 Transmit Operation**



**Figure 21.31 Receive Operation**

### 21.4.3 HDLC Data Processing Mode (Communication Units 0 and 1)

In HDLC data processing mode, bit stuffing, flag detection, abort detection and CRC calculation are available for HDLC control. f<sub>1</sub>, f<sub>8</sub> or f<sub>2n</sub> can be selected as the communication unit 0 transfer clock. f<sub>1</sub>, f<sub>8</sub>, f<sub>2n</sub> or clock, generated in the channel 0 or 1, can be selected as the communication unit 1 transfer clock. No pin is used. To convert data, data to be transmitted is written to the GiTB register (i=0,1) and the data conversion result is restored after data conversion. If any data are in the GiTO register after data conversion, the conversion is terminated. If no data is in the GiTO register, bit stuffing processing is executed regardless of no data available in the transmit output buffer. A CRC value is calculated every time one bit is converted. If no data is in the GiRI register, received data conversion is terminated.

Table 21.25 list specifications of the HDLC data processing mode. Tables 21.26 and 21.27 list clock settings. Table 21.28 lists register settings.

**Table 21.25 HDLC Processing Mode Specifications (Communication Units 0 and 1)**

Item	Specification
Input Data Format	8-bit data fixed, bit alignment is optional
Output Data Format	8-bit data fixed
Transfer Clock	See Tables 21.26 and 21.27
I/O Method	<ul style="list-style-type: none"> <li>• During transmit data processing, value set in the GiTB register is converted in HDLC data processing mode and transferred to the GiTO register.</li> <li>• During received data processing, value set in the GiRI register is converted in HDLC data processing mode and transferred to the GiRB register. The value in the GiRI register is also transferred to the GiTB register (received data register).</li> </ul>
Bit Stuffing	During transmit data processing, "0" following five continuous "1" is inserted. During received data processing, "0" following five continuous "1" is deleted.
Flag Detection	Write the flag data "7E16" to the GiCMP <sub>j</sub> register (j=0 to 3) to use the special communication interrupt (the SRTiR bit in the IIO4IR register)
Abort Detection	Write the masked data "0116" to the GiMSK <sub>j</sub> register
CRC	<p>The CRC1 and CRC0 bits are set to "112" (<math>X^{16}+X^{12}+X^5+1</math>). The CRCV bit is set to "1" (set to "FFFF16").</p> <ul style="list-style-type: none"> <li>• During transmit data processing, CRC calculation result is stored into the GiTCRC register. The TCRCE bit in the GiETC register is set to "1" (transmit CRC used). The CRC calculation result is reset when the TE bit in the GiCR register is set to "0" (transmit disabled).</li> <li>• During received data processing, CRC calculation result is stored into the GiRCRC register. The RCRCE bit in the GiERC register is set to "1" (receive CRC used). The CRC calculation result is reset by comparing the flag data "7E16" and matching the result with the value in the GiCMP<sub>3</sub> register. The ACRC bit in the GiEMR register is set to "1" (CRC reset).</li> </ul>
Data Processing Start Condition	<p>The following conditions are required to start transmit data processing:</p> <ul style="list-style-type: none"> <li>• The TE bit in the GiCR register is set to "1" (transmit enabled)</li> <li>• Data is written to the GiTB register</li> </ul> <p>The following conditions are required to start receive data processing:</p> <ul style="list-style-type: none"> <li>• The RE bit in the GiCR register is set to "1" (receive enabled)</li> <li>• Data is written to the GiRI register</li> </ul>

**Table 21.25 HDLC Processing Mode Specifications (Continued)**

Item	Specification
Interrupt Request	<p>During transmit data processing,</p> <ul style="list-style-type: none"> <li>• One of the following conditions can be selected to set the GiTOR bit<sup>(1)</sup> in the interrupt request register to "1" (interrupt requested). <ul style="list-style-type: none"> <li>– When the IRS bit in the GiMR register is set to "0" (no data in the GiTB register) and data is transferred from the GiTB register to the transmit register (transmit start).</li> <li>– When the IRS bit is set to "1" (transmission completed) and data transfer from the transmit register to the GiTO register is completed.</li> </ul> </li> <li>• When data, which is already converted to HDLC data, is transferred from the receive register of the GiTO register to the transmit buffer, the GiTOR bit is set to "1"</li> </ul> <p>During received data processing,</p> <ul style="list-style-type: none"> <li>• When data is transferred from the GiRI register to the GiRB register (reception completed), the GiRIR bit<sup>(1)</sup> is set to "1".</li> <li>• When received data is transferred from the receive buffer of the GiRI register to the receive register, the GiRIR bit is set to "1".</li> <li>• When the GiTB register is compared to the GiCMPj register (j=0 to 3), the SRTiR bit<sup>(1)</sup> is set to "1".</li> </ul>

**NOTE:**

1. See **Figure 10.14** for details on the GiTOR bit, GiRIR bit and SRTiR bit.

**Table 21.26 Clock Settings (Communication Unit 0)**

Transfer Clock <sup>(1)</sup>	CCS Register	
	CCS0 Bit	CCS1 Bit
f1	1	0
f8	1	1
f <sub>2n</sub> <sup>(2)</sup>	0	1

**NOTES:**

1. The transfer clock for reception is generated when the RSHTTE bit in the G0ERC register is set to "1" (receive shift operation enabled).
2. The CNT3 to CNT0 bits in the TCSPR register select no division ( $n=0$ ) or divide-by- $2n$  ( $n=1$  to 15).

**Table 21.27 Clock Settings (Communication Unit 1)**

Transfer Clock <sup>(1)</sup>	CCS Register	
	CCS2 Bit	CCS3 Bit
$\frac{f_{BT1}}{2 \times (n+2)}$ <sup>(2)</sup>	0	0
f1	1	0
f8	1	1
f <sub>2n</sub> <sup>(3)</sup>	0	1

$n$ : Setting value of the G1PO0 register, 0001<sub>16</sub> to FFFD<sub>16</sub>

**NOTES:**

1. The transfer clock for reception is generated when the RSHTTE bit in the G1ERC register is set to "1" (receive shift operation enabled).
2. The transfer clock is generated in single-phase waveform output mode of the channel 1.
3. The CNT3 to CNT0 bits in the TCSPR register select no division ( $n=0$ ) or divide-by- $2n$  ( $n=1$  to 15).

**Table 21.28 Register Settings in HDLC Processing Mode (Communication Units 0 and 1)**

Register	Bit	Function
G1BCR0	BCK1, BCK0	Select count source
	DIV4 to DIV0	Select divide ratio of count source
	IT	Select the base timer interrupt
G1BCR1 <sup>(1)</sup>	7 to 0	Set to "0001 0010 <sub>2</sub> "
G1POCR0 <sup>(1)</sup>	7 to 0	Set to "0000 0000 <sub>2</sub> "
G1POCR1 <sup>(1)</sup>	7 to 0	Set to "0000 0000 <sub>2</sub> "
G1PO0 <sup>(1)</sup>	15 to 0	Set bit rate
G1PO1 <sup>(1)</sup>	15 to 0	Set the timing of the rising edge of the transfer clock. Timing of the falling edge ("H" width of the transfer clock) is fixed. Setting value of the G1PO1 register ≤ Setting value of the G1PO0 register
G1FS <sup>(1)</sup>	FSC1, FSC0	Set to "00 <sub>2</sub> "
G1FE <sup>(1)</sup>	IFE1, IFE0	Set to "11 <sub>2</sub> "
GiMR	GMD1, GMD0	Set to "11 <sub>2</sub> "
	CKDIR	Set to "0"
	UFORM	Set to "0"
	IRS	Select what causes the transmit interrupt to be generated
GiEMR	7 to 0	Set to "1111 0110 <sub>2</sub> "
GiCR	TI	Transmit buffer empty flag
	TXEPT	Transmit register empty flag
	RI	Receive complete flag
	TE	Transmit enable bit
	RE	Receive enable bit
GiETC	SOF	Set to "0"
	TCRCE	Select whether transmit CRC is used or not
	ABTE	Set to "0"
	TBSF1, TBSF0	Transmit bit stuffing
GiERC	CMP2E to CMP0E	Select whether received data is compared or not
	CMP3E	Set to "1"
	RCRCE	Select whether receive CRC is used or not
	RSHTE	Set to "1" to use it in the receiver
	RBSF1, RBSF0	Receive bit stuffing
GiIRF	BSERR, ABT	Set to "0"
	IRF3 to IRF0	Select what causes an interrupt to be generated
GiCMP0, GiCMP1	7 to 0	Write "FE <sub>16</sub> " to abort processing
GiCMP2	7 to 0	Data to be compared
GiCMP3	7 to 0	Write "7E <sub>16</sub> "
GiMSK0, GiMSK1	7 to 0	Write "01 <sub>16</sub> " to abort processing
GiTCRC	15 to 0	Transmit CRC calculation result can be read
GiRCRC	15 to 0	Receive CRC calculation result can be read
GiTO	7 to 0	Data, which is output from a transmit data generation circuit, can be read
GiRI	7 to 0	Set data input to a receive data generation circuit
GiRB	7 to 0	Received data is stored
GiTB	7 to 0	For transmission: write data to be transmitted For reception : received data for comparison is stored
CCS	CCS1, CCS0	Select the HDLC processing clock
	CCS3, CCS2	Select the HDLC processing clock

i=0, 1

**NOTE:**

1. These register settings are required when the CCS3 and CCS2 bit in the CCS register are set to "00<sub>2</sub>" (clock output from channel j (j=1 to 3)).

## 22. CAN Module

The CAN (Controller Area Network) module included in the M32C/88 Group (M32C/88T) is a Full CAN module, compatible with CAN Specification 2.0 Part B. Three channels, CAN0, CAN1, and CAN2, can be used. Table 22.1 lists specifications of the CAN module.

**Table 22.1 CAN Module Specifications**

Item	Specification
Protocol	CAN Specification 2.0 Part B
Message Slots	16 slots
Polarity	Dominant: "L" Recessive: "H"
Acceptance Filter	Global mask: 1 (for message slots 0 to 13) Local mask: 2 (for message slots 14 and 15 respectively)
Baud Rate	$\text{Baud rate} = \frac{1}{Tq \text{ clock cycle} \times Tq \text{ per bit}} \quad \text{--- Max. 1 Mbps}$ $Tq \text{ clock cycle} = \frac{BRP + 1}{\text{CAN clock}}$ $Tq \text{ per bit} = SS + PTS + PBS1 + PBS2$ <p>Tq: Time quantum BRP: Setting value of the C0BRP and C1BRP registers, 1-255 SS: Synchronization Segment; 1 Tq PTS: Propagation Time Segment; 1 to 8 Tq PBS1: Phase Buffer Segment 1; 2 to 8 Tq PBS2: Phase Buffer Segment 2; 2 to 8 Tq</p>
Remote Frame Automatic Answering Function	Message slot that receives the remote frame transmits the data frame automatically
Time Stamp Function	Time stamp function with a 16-bit counter. Count source can be selected from the CAN bus bit clock divided by 1, 2, 3 or 4 $\text{CAN bus bit clock} = \frac{1}{\text{CAN bit time}}$
BasicCAN Mode	BasicCAN function can be used with the CANi message slots 14 and 15
Transmit Abort Function	Transmit request is aborted
Loopback Function	Frame transmitted by the CAN module is received by the same CAN module
Forcible Error Active Transition Function	The CAN module is forced into an error active state by resetting an error counter.
Single-Shot Transmit Function	The CAN module does not transmit data again even if arbitration lost or transmission error causes a transmission failure
Self-Test Function	The CAN module communicates internally and diagnoses its CAN module state

**NOTE:**

1. Use an oscillator with maximum 1.58% oscillator tolerance.

Figure 22.1 shows a block diagram of the CAN module. Figure 22.2 shows CAN<sub>i</sub> message slot (the message slot) *j* (*j* = 0 to 15) and CAN<sub>i</sub> message slot buffer (*i*=0 to 2). Table 22.2 lists pin settings of the CAN module.

The message slot cannot be accessed directly from the CPU. Allocate the message slot *j* to be used to the message slot buffer 0 or 1. The message slot *j* is accessed via the message slot buffer address. The CiSBS register selects the message slot *j* to be allocated. Figure 22.2 shows the 16-byte message slot buffer and message slot.

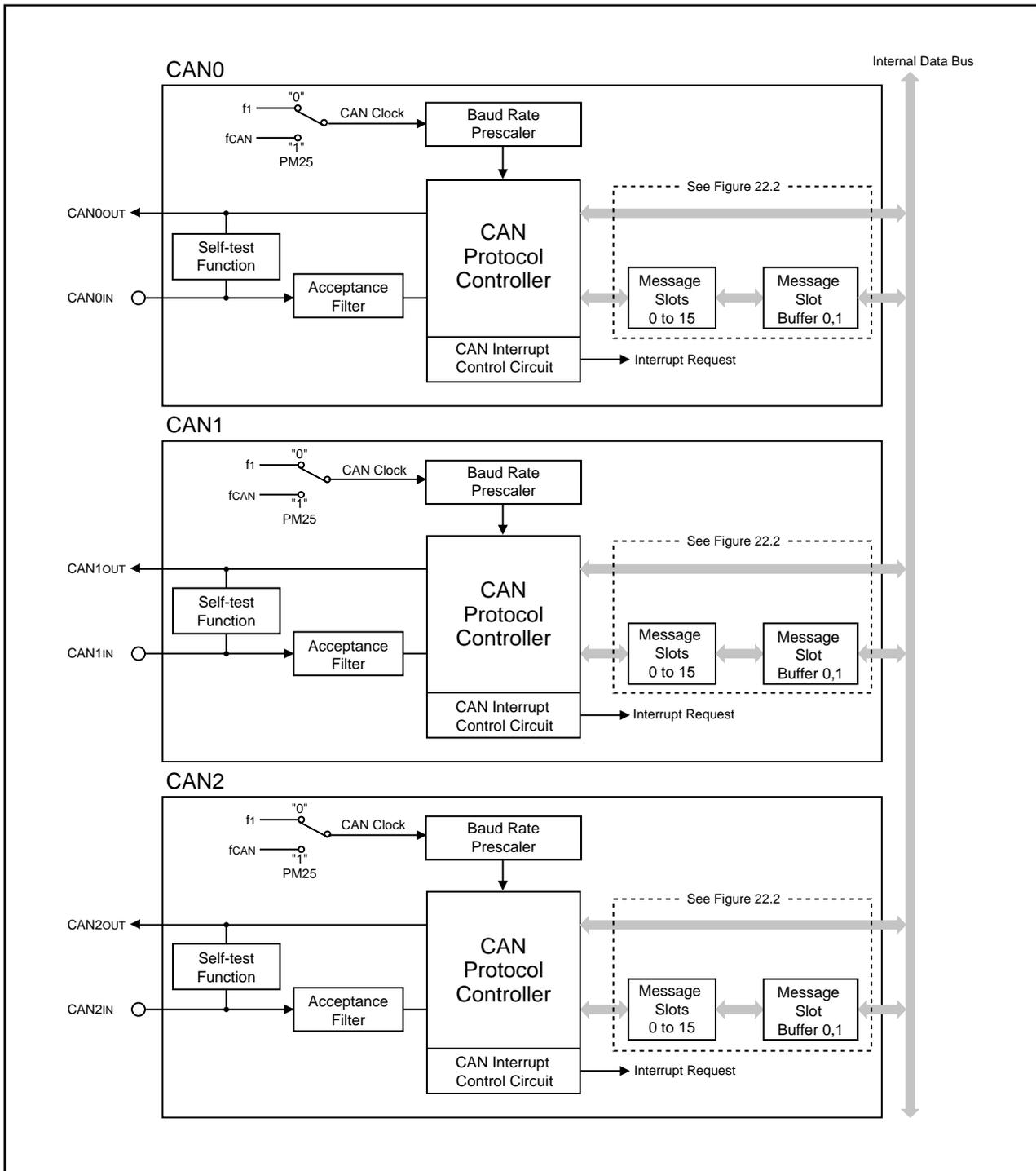


Figure 22.1 CAN Module Block Diagram

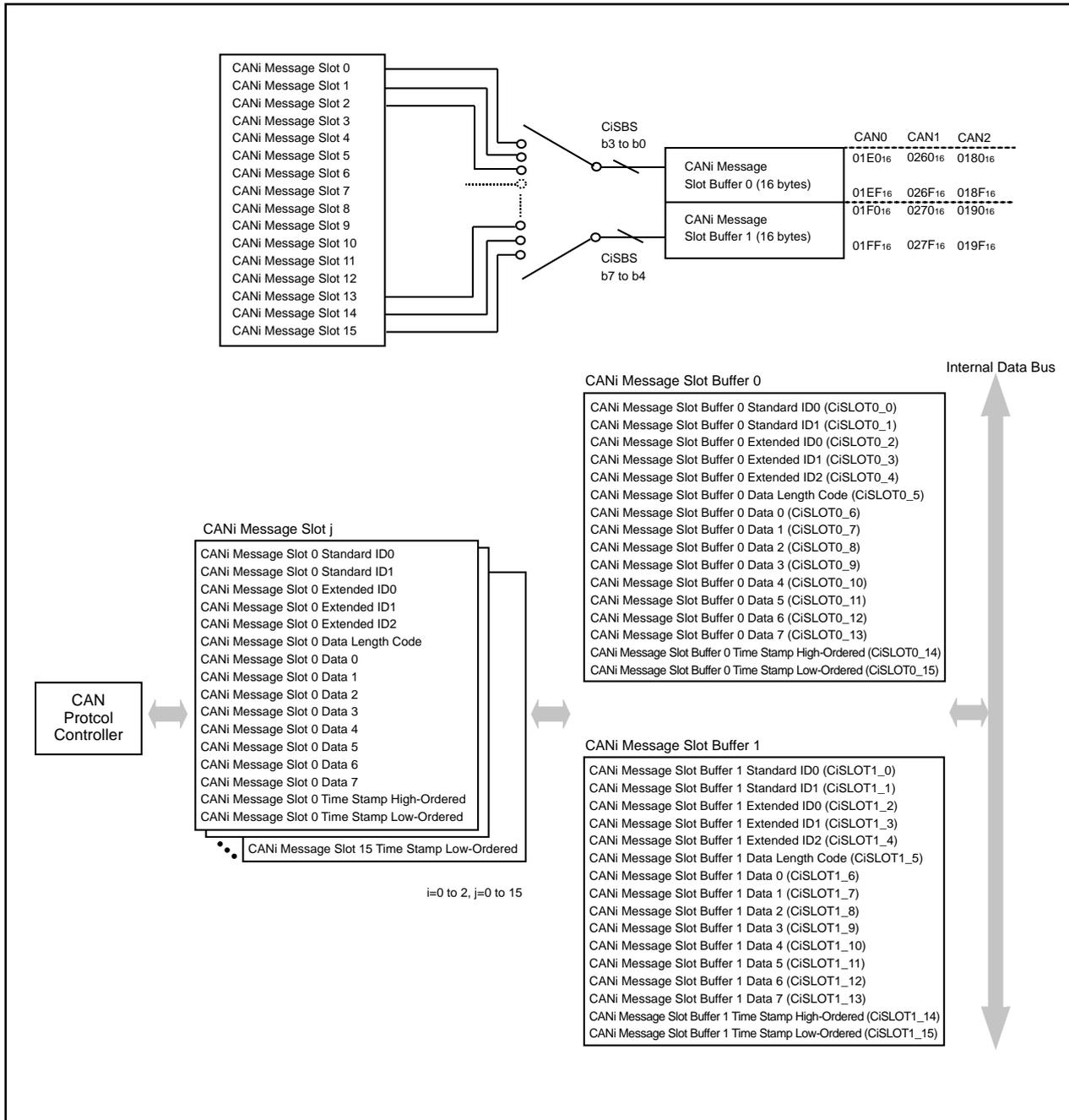


Figure 22.2 CANi Message Slot and CANi Message Slot Buffer

**Table 22.2 Pin Settings**

Port	Function	Bit and Setting <sup>(2)</sup>				
		IPS, IPSA Registers	PS0, PS1, PS2, PS3 Registers	PSL0, PSL1, PSL2, PSL3 Registers	PSC, PSC2, PSC3 Registers	PD6, PD7, PD8, PD9 <sup>(1)</sup> Registers
P6 <sub>0</sub>	CAN2 <sub>OUT</sub>	–	PS0_0=1	PSL0_0=1	–	–
P6 <sub>1</sub>	CAN2 <sub>IN</sub>	IPSA_7=0	PS0_1=0		–	PD6_1=0
P7 <sub>6</sub>	CAN0 <sub>OUT</sub>	IPSA_7=0	PS1_6=1	PSL1_6=0	PSC_6=1	–
	CAN02 <sub>OUT</sub>	IPSA_7=1	PS1_6=1	PSL1_6=0	PSC_6=1	–
P7 <sub>7</sub>	CAN0 <sub>IN</sub>	IPS_3=0	PS1_7=0	–	–	PD7_7=0
	CAN02 <sub>IN</sub>	IPS_3=0, IPSA_7=1	PS1_7=0	–	–	PD7_7=0
P8 <sub>2</sub>	CAN0 <sub>OUT</sub>	–	PS2_2=1	PSL2_2=1	PSC2_2=0	–
	CAN1 <sub>OUT</sub>	–	PS2_2=1	PSL2_2=1	PSC2_2=1	–
P8 <sub>3</sub>	CAN0 <sub>IN</sub>	IPS_3=1	–	–	–	PD8_3=0
	CAN1 <sub>IN</sub>	IPSA_3=1	–	–	–	PD8_3=0
P9 <sub>5</sub>	CAN1 <sub>IN</sub>	IPSA_3=0	PS3_5=0	PSL3_5=0	–	PD9_5=0
P9 <sub>6</sub>	CAN1 <sub>OUT</sub>	–	PS3_6=1	–	PSC3_6=1	–

NOTE:

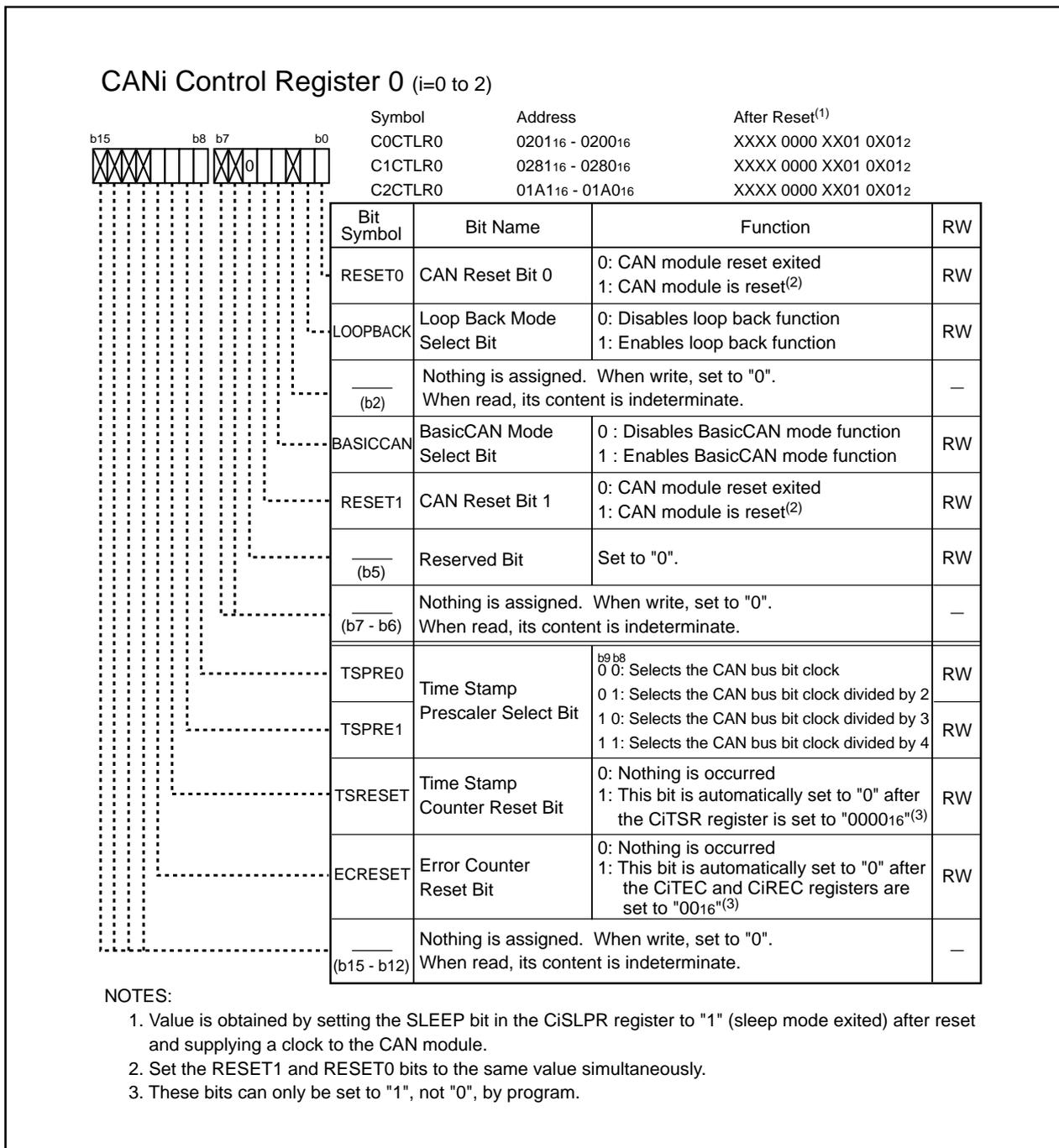
1. Set the PD9 and PS3 registers immediately after the PRC2 bit in the PRCR register is set to "1" (write enabled). Do not generate an interrupt or a DMA transfer between the instruction to set to the PRC2 bit to "1" and the instruction to set the PD9 and PS3 registers.

## 22.1 CAN-Associated Registers

Figures 22.3 to 22.18, and Figures 22.20 to 22.33 show registers associated with CAN. To access the CAN-associated registers, set the CM21 bit in the CM2 register to "0" (main clock or PLL clock as CPU clock) and the MCD4 to MCD0 bits in the MCD register to "100102" (no division mode). Or, set the PM24 bit in the PM2 register to "1" (main clock direct mode) and the PM25 bit in the PM2 register to "1" (CAN clock). Two wait states are added into the bus cycle.

Refer to **7. Processor Mode** and **8. Clock Generation Circuit**.

### 22.1.1 CAN<sub>i</sub> Control Register 0 (CiCTRL0 Register) (i=0 to 2)



**Figure 22.3 COCTRL0, C1CTRL0 and C2CTRL0 Registers**

### 22.1.1.1 RESET1 and RESET0 Bits

When both RESET1 and RESET0 bits are set to "1" (CAN module reset), the CAN module is immediately initialized regardless of ongoing CAN communication.

After the RESET1 and RESET0 bits are set to "1" and the CAN module reset is completed, the CiTSR register (i=0 to 2) is set to "0000<sub>16</sub>". The CiTEC and CiREC registers are set to "00<sub>16</sub>" and the STATE\_ERRPAS and STATE\_BUSOFF bits in the CiSTR register are set to "0" as well.

When both RESET1 and RESET0 bit settings are changed "1" to "0", the CiTSR register starts counting. CAN communication is available after 11 continuous recessive bits are detected.

NOTES:

1. Set the same value in both RESET1 and RESET0 bits simultaneously.
2. Confirm that the STATE\_RESET bit in the CiSTR register is set to "1" (CAN module reset completed) after setting the RESET1 and RESET0 bits to "1".
3. The CANOUT pin puts in a high-level ("H") signal as soon as the RESET1 and RESET0 bits are set to "1". CAN bus error may occur when the RESET1 and RESET0 bits are set to "1" while the CAN frame is transmitting.
4. For CAN communication, set the PS0, PS1, PS2, PS3, PSL0, PSL1, PSL2, PSL3, PSC, PSC2, PSC3, IPS, IPSA, PD6, PD7, PD8, and PD9 registers when the STATE\_RESET bit is set to "1" (CAN module reset completed).

### 22.1.1.2 LOOPBACK Bit

When the LOOPBACK bit is set to "1" (loopback function enabled) and the receive message slot has a matched ID and frame format with a transmitted frame, the transmitted frame is stored to the receive message slot.

NOTES:

1. No ACK for the transmitted frame is returned.
2. Change the LOOPBACK bit setting only when the STATE\_RESET bit is set to "1" (CAN module reset completed).

### 22.1.1.3 BASICCAN Bit

When the BASICCAN bit is set to "1", the message slots 14 and 15 enter BasicCAN mode.

In BasicCAN mode, the message slots 14 and 15 are used as dual-structured buffers. The message slots 14 and 15 alternately store a received frame having matched ID detected by acceptance filtering. ID in the message slot 14 and the CiLMAR0 to CiLMAR4 registers are used for acceptance filtering when the message slot 14 is active (the next received frame is to be stored in the message slot 14). ID in the message slot 15 and the CiLMBR0 to CiLMBR4 registers are used when the message slot 15 is active. Both data frame and remote frame can be received.

Use the following procedure to enter BasicCAN mode.

- (1) Set the BASICCAN bit to "1".
- (2) Set the same value into IDs in the message slots 14 and 15.
- (3) Set the same value in the CiLMAR0 to CiLMAR4 registers and CiLMBR0 to CiLMBR4 registers.
- (4) Set the IDE14 and IDE15 bits in the CiIDR register to select a frame format (standard or extended) for the message slots 14 and 15. (Set to the same format.)
- (5) Set the CiMCTL14 and CiMCTL15 registers in the message slots 14 and 15 to receive the data frame.

**NOTES:**

1. Change the BASICCAN bit setting only when the STATE\_RESET bit is set to "1" (CAN module reset completed).
2. The message slot 14 is the first slot to become active after the RESET1 and RESET0 bits are set to "0".
3. The message slots 0 to 13 are not affected by entering BasicCAN mode.

**22.1.1.4 TSPRE1, TSPRE0 Bits**

The TSPRE1 and TSPRE0 bits determine which count source is used for the time stamp counter.

**NOTE:**

1. Change the TSPRE1 and TSPRE0 bit settings only when the STATE\_RESET bit is set to "1" (CAN module reset completed).

**22.1.1.5 TSRESET Bit**

When the TSRESET bit is set to "1", the CiTSR register is set to "0000<sub>16</sub>". The TSRESET bit is automatically set to "0" after the CiTSR register is set to "0000<sub>16</sub>".

**22.1.1.6 ECRESET Bit**

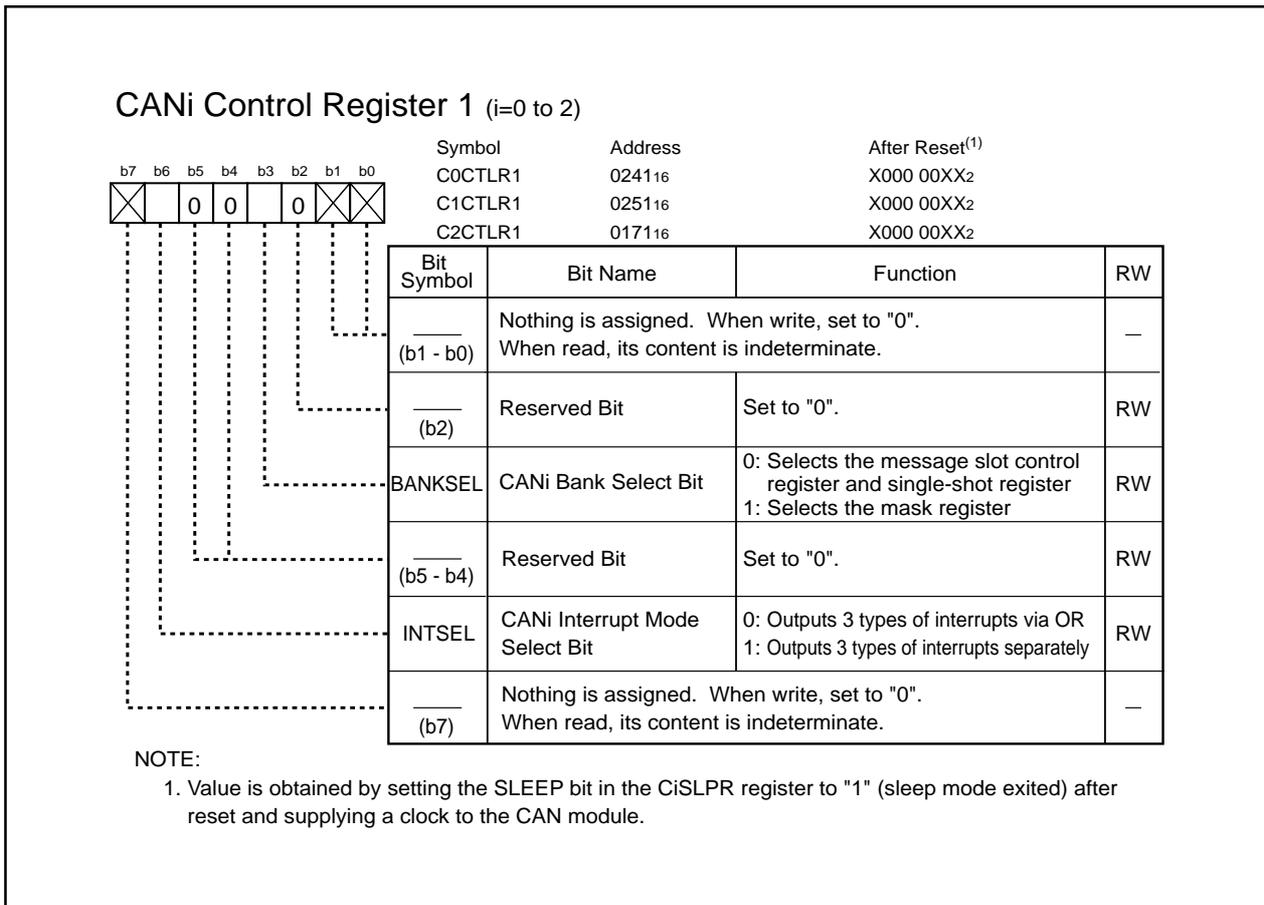
When the ECRESET bit is set to "1", the CiTEC and CiREC registers are set to "00<sub>16</sub>". The CAN module forcibly goes into an error active state.

The ECRESET bit is automatically set to "0" after the CAN module enters an error active state.

**NOTES:**

1. In an error active state, the CAN module is ready to communicate when 11 continuous recessive bits are detected on the CAN bus.
2. The CANiOUT pin provides an "H" signal output as soon as the ECRESET bit is set to "1". The CAN bus error may occur when setting the ECRESET bit to "1" during CAN frame transmission.

### 22.1.2 CANi Control Register 1 (CiCTLR1 Register) (i=0 to 2)



**Figure 22.4 C0CTLR1, C1CTLR1 and C2CTLR1 Registers**

#### 22.1.2.1 BANKSEL Bit

The BANKSEL bit in the C0CTLR1 register selects the registers allocated to addresses 0220<sub>16</sub> to 023F<sub>16</sub>. The BANKSEL bit in the C1CTLR1 register selects registers allocated to addresses 02A0<sub>16</sub> to 02BF<sub>16</sub>. The BANKSEL bit in the C2CTLR1 register selects registers allocated to addresses 01C0<sub>16</sub> to 01DF<sub>16</sub>.

The CiSSCTLR register, CiSSSTR register, and the CiMCTL0 to CiMCTL15 registers can be accessed by setting the BANKSEL bit to "0". The CiGMR0 to CiGMR4 registers, CiLMAR0 to CiLMAR4 registers and CiLMBR0 to CiLMBR4 registers can be accessed by setting the BANKSEL bit to "1".

#### 22.1.2.2 INTSEL Bit

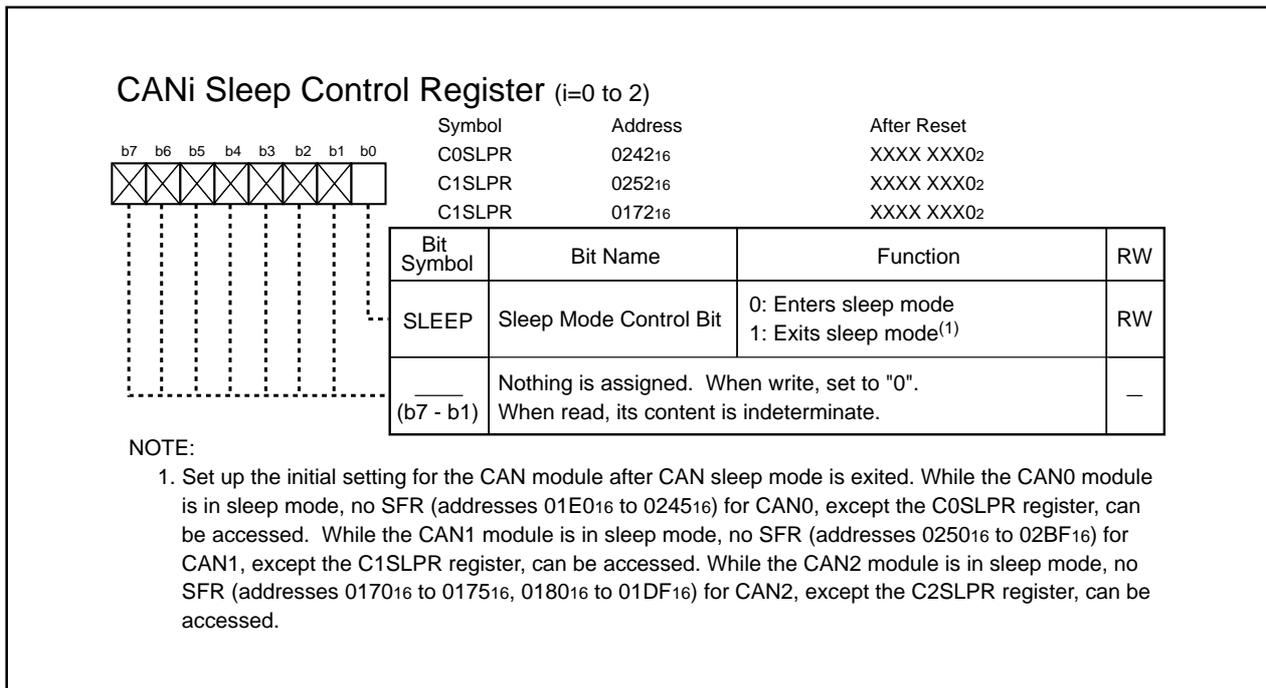
The INTSEL bit determines whether the three types of interrupt outputs (CANi transmit interrupt, CANi receive interrupt and CANi error interrupt) are provided via OR or is separately.

Refer to **22.4 CAN Interrupts** for details.

NOTE:

- Change the INTSEL bit setting when the STATE\_RESET bit is set to "1" (CAN module reset completed).

### 22.1.3 CAN<sub>i</sub> Sleep Control Register (CiSLPR Register) (i=0 to 2)



**Figure 22.5 C0SLPR, C1SLPR and C2SLPR Registers**

#### 22.1.3.1 SLEEP Bit

When the SLEEP bit is set to "0", the clock supplied to the CAN module stops running and the CAN module enters sleep mode.

When the SLEEP bit is set to "1", the clock supplied to the CAN module starts running and the CAN module exits sleep mode.

**NOTE:**

- Enter sleep mode after the STATE\_RESET bit in the CiSTR register is set to "1" (CAN module reset completed).

### 22.1.4 CANi Status Register (CiSTR Register) (i=0 to 2)

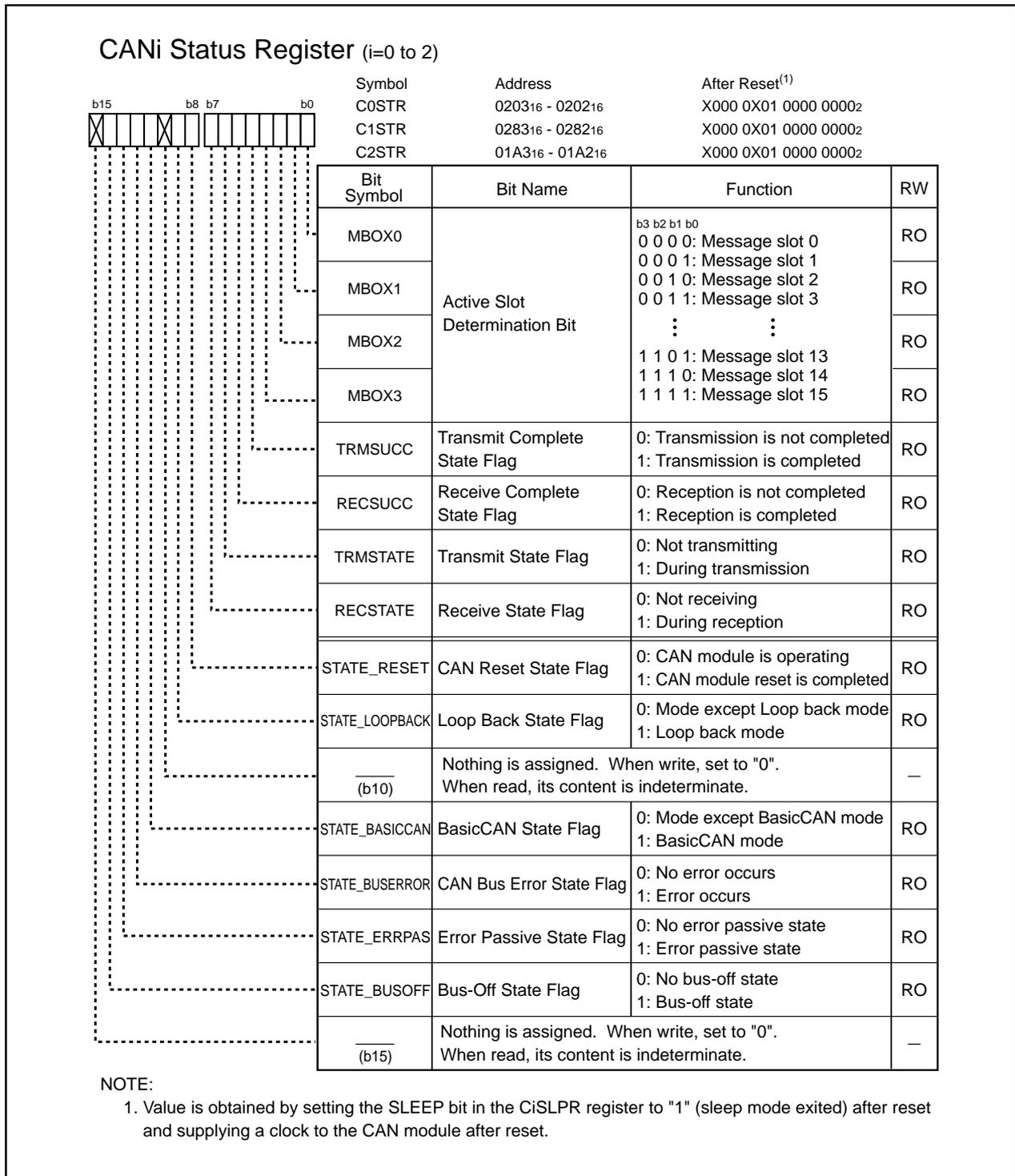


Figure 22.6 C0STR, C1STR and C2STR Registers

#### 22.1.4.1 MBOX3 to MBOX0 Bits

The MBOX3 to MBOX0 bits store relevant slot numbers when the CAN module has completed transmitting data or storing received data.

#### 22.1.4.2 TRMSUCC Bit

The TRMSUCC bit is set to "1" when the CAN module has transmitted data as expected.  
The TRMSUCC bit is set to "0" when the CAN module has received data as expected.

#### 22.1.4.3 RECSUCC Bit

The RECSUCC bit is set to "1" when the CAN module has received data as expected. (Whether received message has been stored in the message slot or not is irrelevant.) If the received message is transmitted in loopback mode, the TRMSUCC bit is set to "1" and the RECSUCC bit is set to "0". The RECSUCC bit is set to "0" when the CAN module has transmitted data as expected.

#### 22.1.4.4 TRMSTATE Bit

The TRMSTATE bit is set to "1" when the CAN module is performing as a transmit node. The TRMSTATE bit is set to "0" when the CAN module is in a bus-idle state or starts performing as a receive node.

#### 22.1.4.5 RECSTATE Bit

The RECSTATE bit is set to "1" when the CAN module is performing as a receive node. The RECSTATE bit is set to "0" when the CAN module is in a bus-idle state or starts performing as a transmit node.

#### 22.1.4.6 STATE\_RESET Bit

After both RESET1 and RESET0 bits are set to "1" (CAN module reset), the STATE\_RESET bit is set to "1" as soon as the CAN module is initialized. The STATE\_RESET bit is set to "0" when the RESET1 and RESET0 bits are set to "0".

#### 22.1.4.7 STATE\_LOOPBACK Bit

The STATE\_LOOPBACK bit is set to "1" when the CAN module is in loopback mode. The STATE\_LOOPBACK bit is set to "1" when the LOOPBACK bit in the CiCTRL0 register is set to "1" (loop back function enabled). The STATE\_LOOPBACK bit is set to "0" when the LOOPBACK bit is set to "0" (loop back function disabled).

#### 22.1.4.8 STATE\_BASICCAN Bit

The STATE\_BASICCAN bit is set to "1" when the CAN module is in BasicCAN mode. Refer to **22.1.1.3 BASICCAN bit** for BasicCAN mode. The STATE\_BASICCAN bit is set to "0" when the BASICCAN bit is set to "0" (BasicCAN mode function disabled). The STATE\_BASICCAN bit is set to "1" when the BASICCAN bit is set to "1" (BasicCAN mode function enabled), the REMACTIVE bits in the CiMCTL14 and CiMCTL15 registers in the message slots 14 and 15 are set to "0" (data frame received).

#### 22.1.4.9 STATE\_BUSERROR Bit

The STATE\_BUSERROR bit is set to "1" when a CAN communication error is detected. The STATE\_BUSERROR bit is set to "0" when the CAN module has transmitted or received data as expected. Whether a received message has been stored into the message slot or not is irrelevant. NOTE:

1. When the STATE\_BUSERROR bit is set to "1", the STATE\_BUSERROR bit remains unchanged even if both RESET1 and RESET0 bits are set to "1" (CAN module reset).

**22.1.4.10 STATE\_ERRPAS Bit**

The STATE\_ERRPAS bit is set to "1" when the value of the CiTEC or CiREC register (i=0, 1) exceeds 127 and the CAN module is placed in an error-passive state.

The STATE\_ERRPAS bit is set to "0" when the CAN module in an error-passive state is placed in another error state.

The STATE\_ERRPAS bit is set to "0" when both RESET1 and RESET0 bits are set to "1" (CAN module is reset).

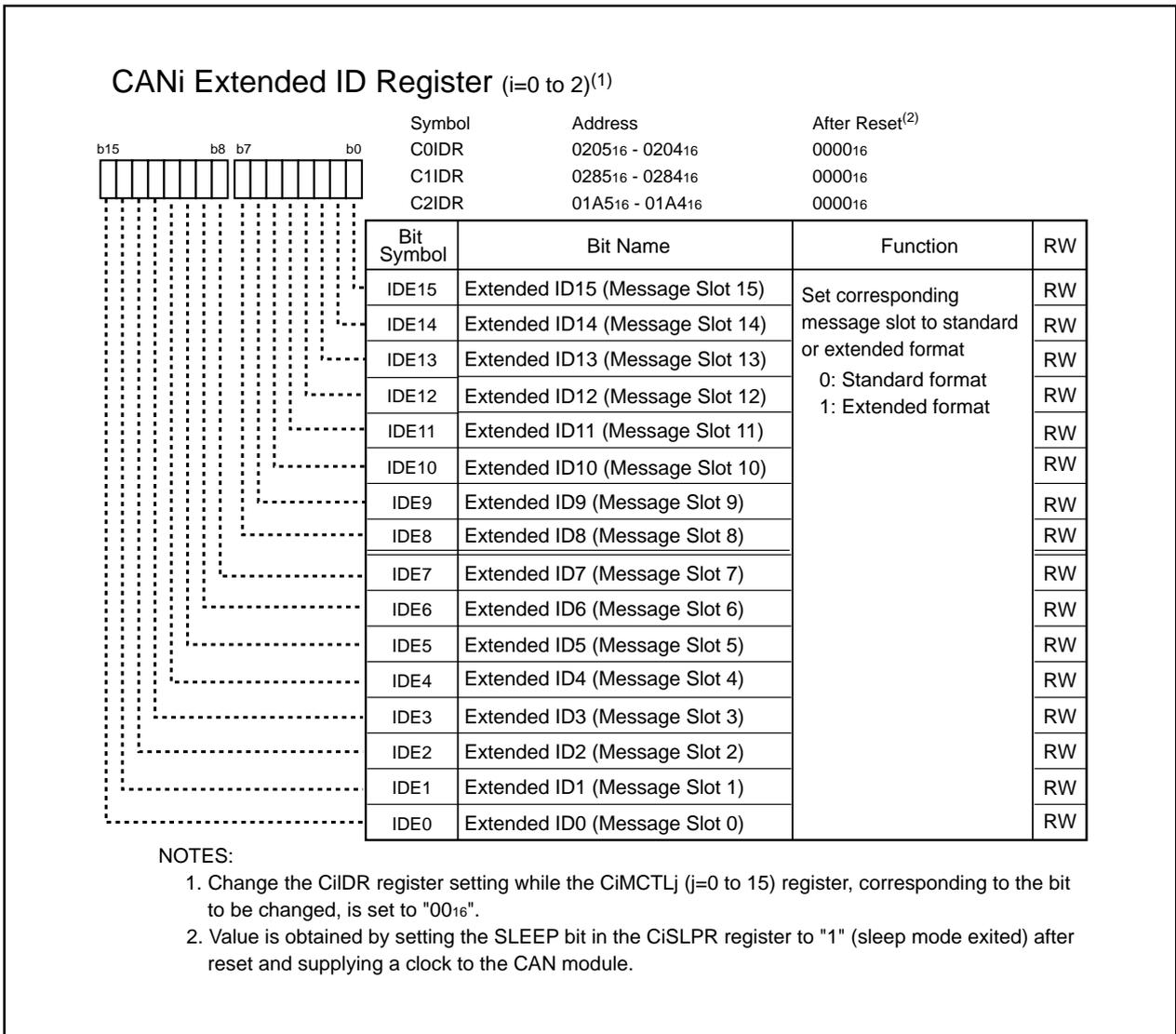
**22.1.4.11 STATE\_BUSOFF Bit**

The STATE\_BUSOFF bit is set to "1" when the value of the CiTEC register exceeds 255 and the CAN module is placed in a bus-off state.

The STATE\_BUSOFF bit is set to "0" when the CAN module in a bus-off state is placed in an error-active state.

The STATE\_BUSOFF bit is set to "0" when both RESET1 and RESET0 bits are set to "1" (CAN module reset).

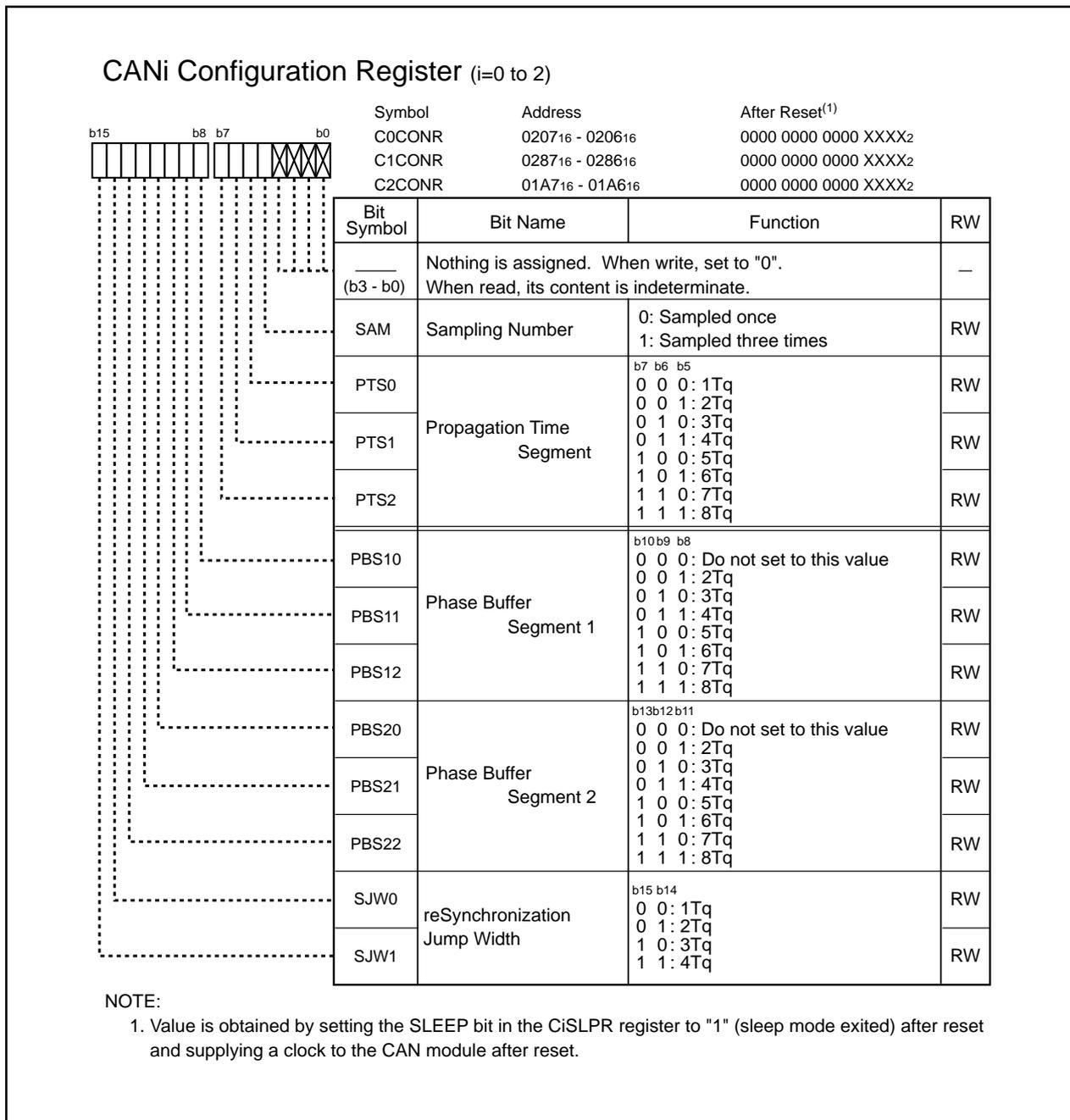
### 22.1.5 CANi Extended ID Register (CiDR Register) (i=0 to 2)



**Figure 22.7 C0IDR, C1IDR and C2IDR Registers**

Bits in the CiDR register determine the frame format in the message slot corresponding to each bit. The standard format is selected when the bit is set to "0". The extended format is selected when the bit is set to "1".

### 22.1.6 CANi Configuration Register (CiCONR Register) (i=0 to 2)



**Figure 22.8 C0CONR, C1CONR and C2CONR Registers**

**22.1.6.1 SAM Bit**

The SAM bit determines the number of sample points to be taken per bit.

When the SAM bit is set to "0", only one sample is taken per bit at the end of the Phase Buffer Segment 1 (PBS1) to determine the value of the bit.

When the SAM bit is set to "1", three samples per bit are taken; one time quantum and two time quanta before the end of PBS1, and at the end of PBS1. The sample result value which is detected more than twice becomes the value of the bit sampled.

**22.1.6.2 PTS2 to PTS0 Bits**

The PTS2 to PTS0 bits determine PTS width.

**22.1.6.3 PBS12 to PBS10 Bits**

The PBS12 to PBS10 bits determine PBS1 width. Set the PBS12 to 10 bits to "0012" or more.

**22.1.6.4 PBS22 to PBS20 Bits**

The PBS22 to PBS20 bits determine PBS2 width. Set the PBS22 to PBS20 bits to "0012" or more.

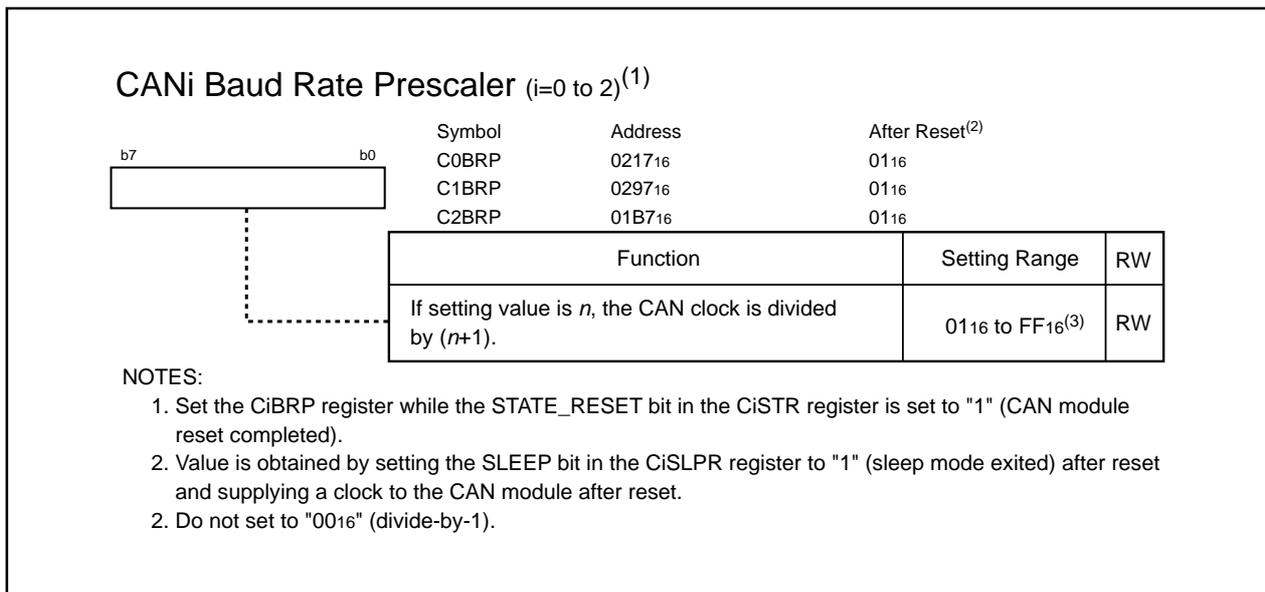
**22.1.6.5 SJW1 and SJW0 Bits**

The SJW1 and SJW0 bits determine SJW width. Set the SJW1 and SJW0 bits to values less than or equal to the PBS12 to PBS10 bit settings and the PBS22 to PBS20 bit settings.

**Table 22.3 Bit Timing when CAN Clock = 30 MHz**

Baud Rate	BRP	Tq Clock Cycles (ns)	Tq Per Bit	PTS+PBS1	PBS2	Sample Point
1Mbps	1	66.7	15	12	2	87%
	1	66.7	15	11	3	80%
	1	66.7	15	10	4	73%
	2	100	10	7	2	80%
	2	100	10	6	3	70%
	2	100	10	5	4	60%
500Kbps	2	100	20	16	3	85%
	2	100	20	15	4	80%
	2	100	20	14	5	75%
	3	133.3	15	12	2	87%
	3	133.3	15	11	3	80%
	3	133.3	15	10	4	73%
	4	166.7	12	9	2	83%
	4	166.7	12	8	3	75%
	4	166.7	12	7	4	67%
	5	200	10	7	2	80%
	5	200	10	6	3	70%
	5	200	10	5	4	60%

### 22.1.7 CANi Baud Rate Prescaler (CiBRP Register) (i=0 to 2)



**Figure 22.9 C0BRP, C1BRP and C2BRP Registers**

The CiBRP register determines the Tq clock cycle of the CAN bit time. The baud rate is obtained from Tq clock cycle x Tq per bit.

$$\text{Tq clock cycle} = (\text{BRP} + 1) / \text{CAN clock}$$

$$\text{Baud rate} = \frac{1}{\text{Tq clock cycle} \times \text{Tq per bit}}$$

$$\text{Tq per bit} = \text{SS} + \text{PTS} + \text{PBS1} + \text{PBS2}$$

Tq: Time quantum

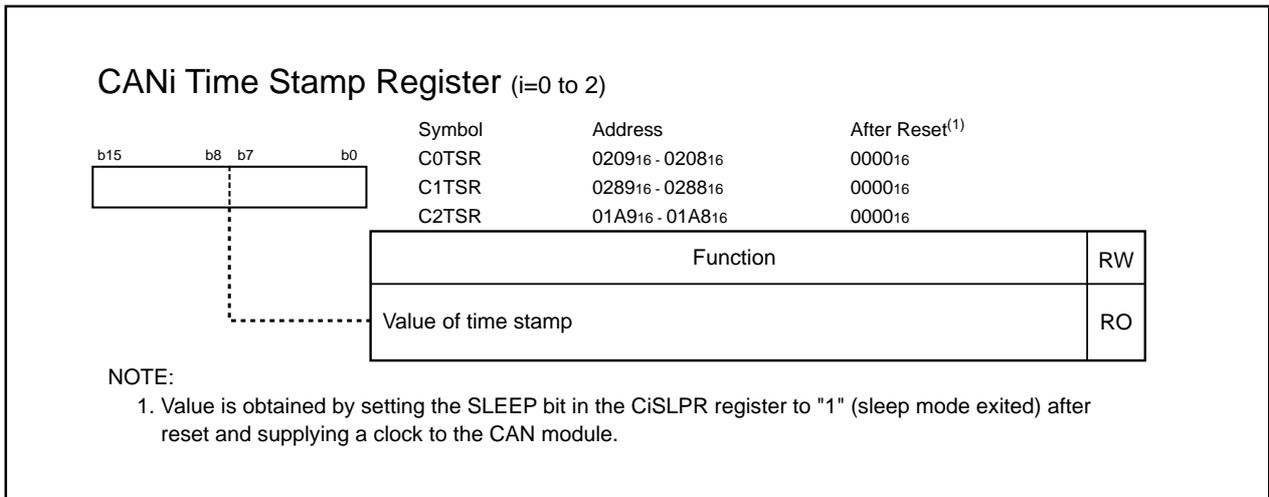
SS: Synchronization Segment; 1 Tq

PBS1: Phase Buffer Segment 1; 2 to 8 Tq

BRP: Setting value of the CiBPR register; 1-255

PTS: Propagation Time Segment; 1 to 8 Tq

PBS2: Phase Buffer Segment 2; 2 to 8 Tq



**22.1.8 CAN<sub>i</sub> Time Stamp Register (CiTSR Register) (i=0 to 2)**

**Figure 22.10 C0TSR, C1TSR and C2TSR Registers**

The CiTSR register is a 16-bit counter. The TSPRE1 and TSPRE0 bits in the CiCTRL0 register select the CAN bus bit clock divided by 1, 2, 3 or 4 as the count source for the CiTSR register. When data transmission or reception is completed, the value of the CiTSR register is automatically stored into the message slot.

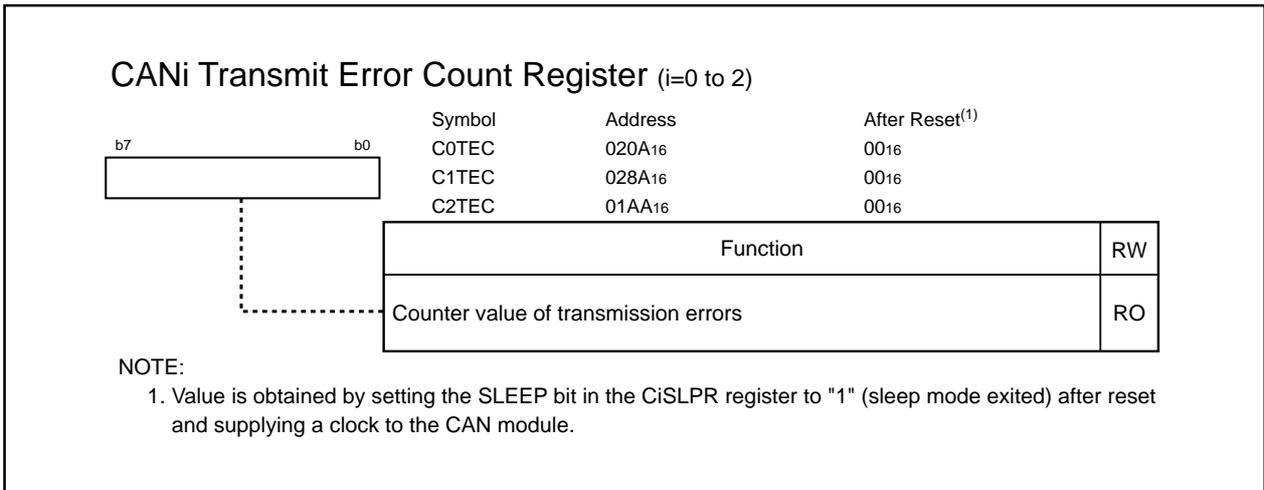
In loopback mode, when either data frame receive message slot or remote frame receive message slot is available to store the message, the value of the CiTSR register is also stored into the message slot when data reception is completed. The value of the CiTSR register is not stored when data transmission is completed.

The CiTSR register starts a counter increment when the RESET1 and RESET0 bits in the CiCTRL0 register are set to "0".

The CiTSR register is set to "0000<sub>16</sub>":

- at the next count timing after the CiTSR register is set to "FFFF<sub>16</sub>";
- when the RESET1 and RESET0 bits are set to "1" (CAN module reset) by program; or
- when the TSRESET bit is set to "1" (CiTSR register reset) by program.

CAN bus bit clock =

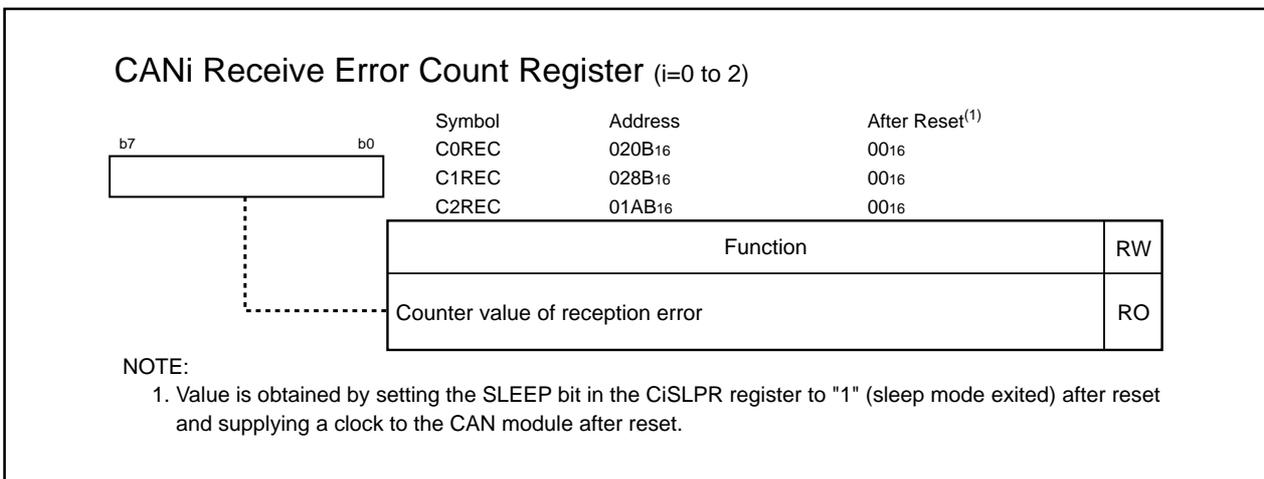


**22.1.9 CANi Transmit Error Count Register (CiTEC Register) (i=0 to 2)**

**Figure 22.11 C0TEC, C1TEC and C2TEC Registers**

In an error active or an error passive state, the counting value of a transmission error is stored into the CiTEC register. The counter is decremented when the CAN module has transmitted data as expected or is incremented when an transmit error occurs.

In a bus-off state, an indeterminate value is stored into the CiTEC register. The CiTEC register is set to "00<sub>16</sub>" when the CAN module is placed in an error active state again.



**22.1.10 CANi Receive Error Count Register (CiREC Register) (i=0 to 2)**

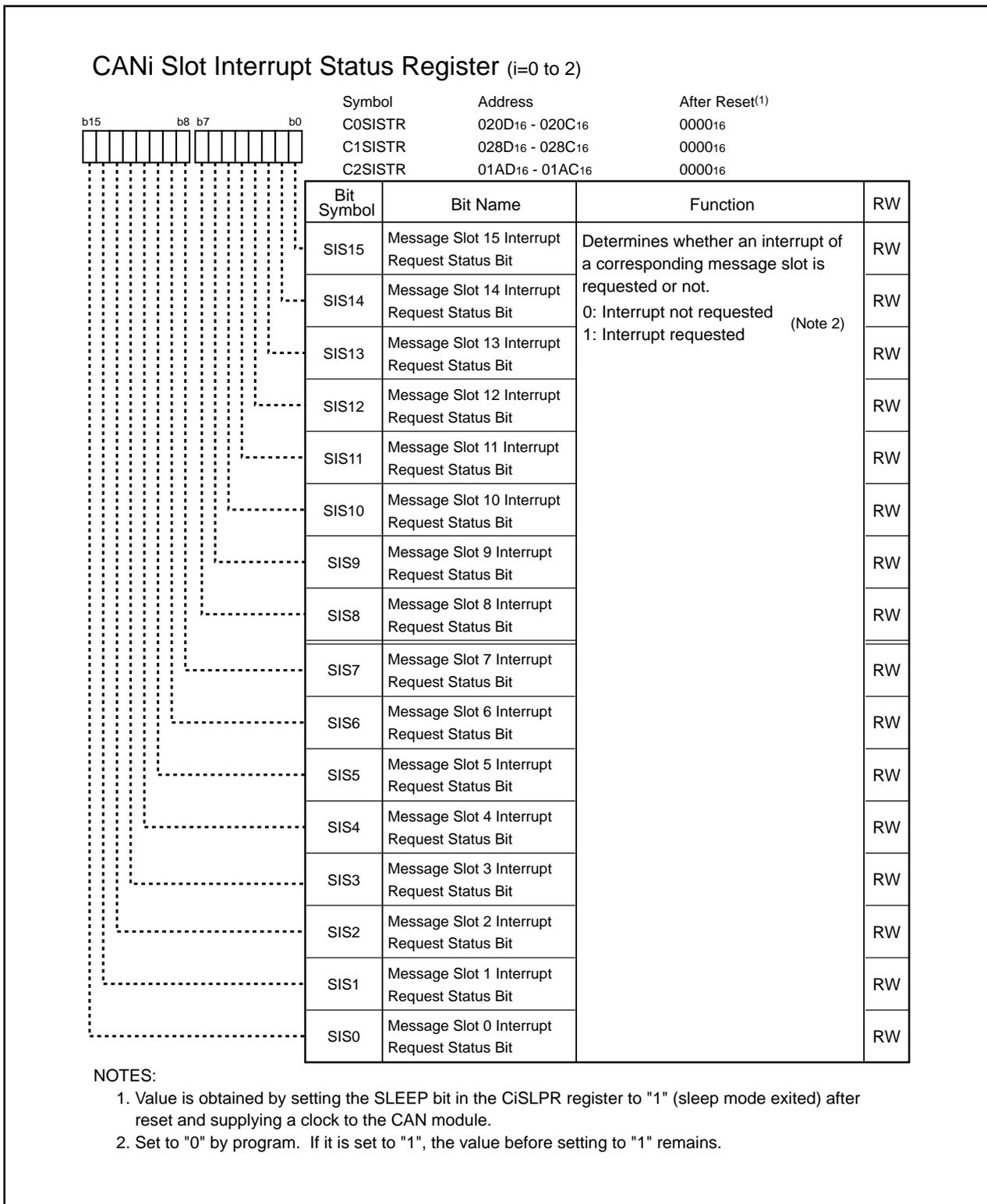
**Figure 22.12 C0REC, C1REC and C2REC Registers**

In an error active or an error passive state, a counting value of the reception error is stored into the CiREC register. The counter is decremented when the CAN module has received data as expected or it is incremented when a receive error occurs.

The CiREC register is set to 127 when the CiREC register is 128 (error passive state) or more and the CAN module has received as expected.

In a bus-off state, an indeterminate value is stored into the CiREC register. The CiREC register is set to "00<sub>16</sub>" when the CAN module is placed in an error active state again.

### 22.1.11 CANi Slot Interrupt Status Register (CiSISTR Register) (i=0 to 2)



**Figure 22.13 C0SISTR, C1SISTR and C2SISTR Registers**

When using the CAN interrupt, the CiSISTR register (i=0 to 2) indicates which message slot is requesting an interrupt. The SISj bits (j=0 to 15) are not automatically set to "0" (no interrupt requested) when an interrupt is acknowledged. Set the SISj bits to "0" by program.

Use the MOV instruction, instead of the bit clear instruction, to set the SISj bits to "0". The SISj bits, which are not being changed to "0", must be set to "1".

For example: To set the SIS0 bit to "0"

Assembly language: `mov.w #07FFFh, C0SISTR`

C language: `c0sistr = 0x7FFF;`

Refer to **22.4 CAN Interrupt** for details.

#### 22.1.11.1 Message Slot for Transmission

The SISj bit is set to "1" (interrupt requested) when the CiTSR register is stored into the message slot j after data transmission is completed.

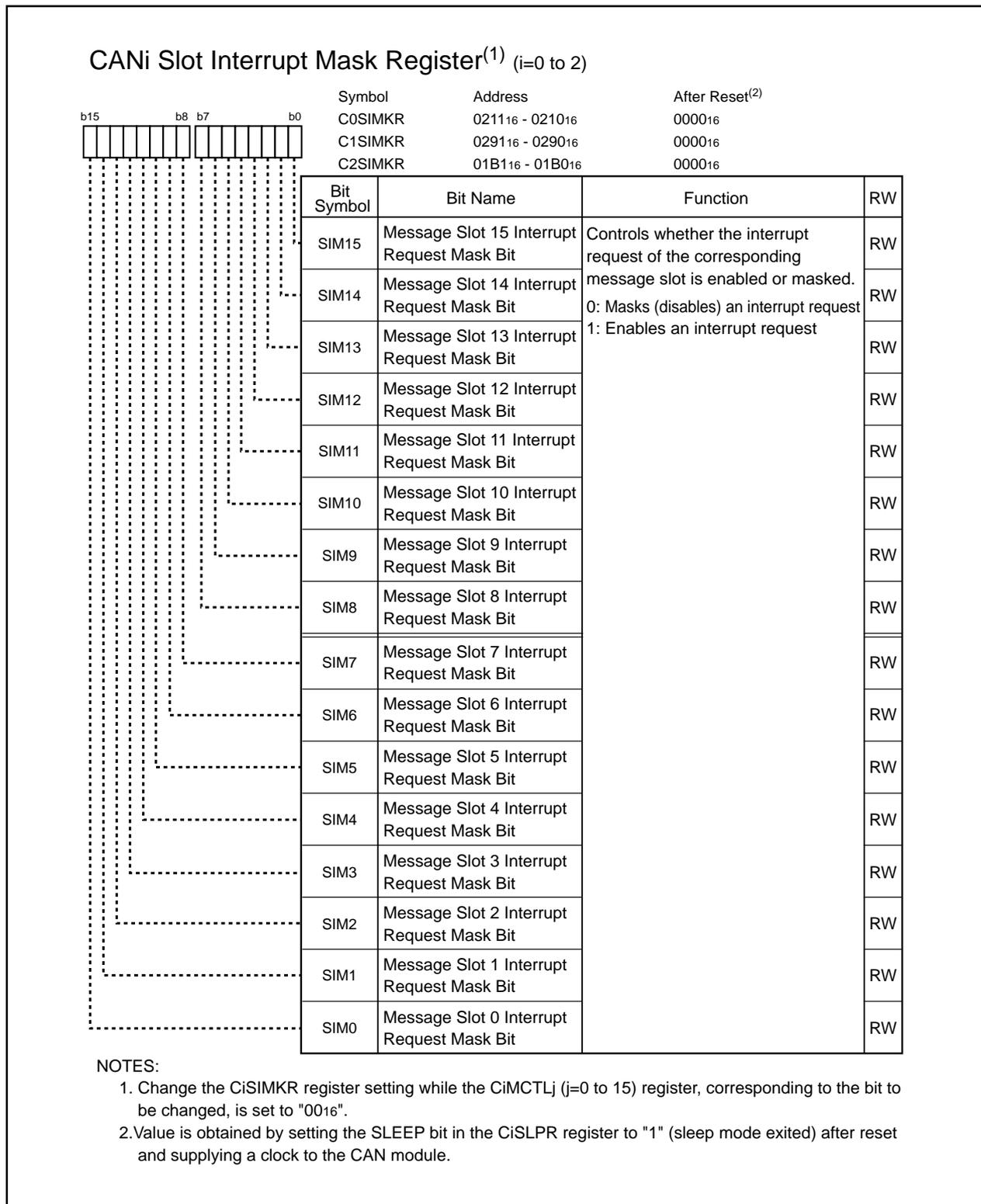
#### 22.1.11.2 Message Slot for Reception

The SISj bit is set to "1" (interrupt requested) when the received message is stored in the message slot j after data reception is completed.

#### NOTES:

- 1.If the automatic answering function is enabled in the remote frame receive message slot, the SISj bit is set to "1" after the remote frame is received and the data frame is transmitted.
- 2.In the remote frame transmit message slot, the SISj bit is set to "1" after the remote frame is transmitted and the data frame is received.
- 3.The SISj bit is set to "1" if the SISj bit is set to "1" by an interrupt request and "0" by program simultaneously.

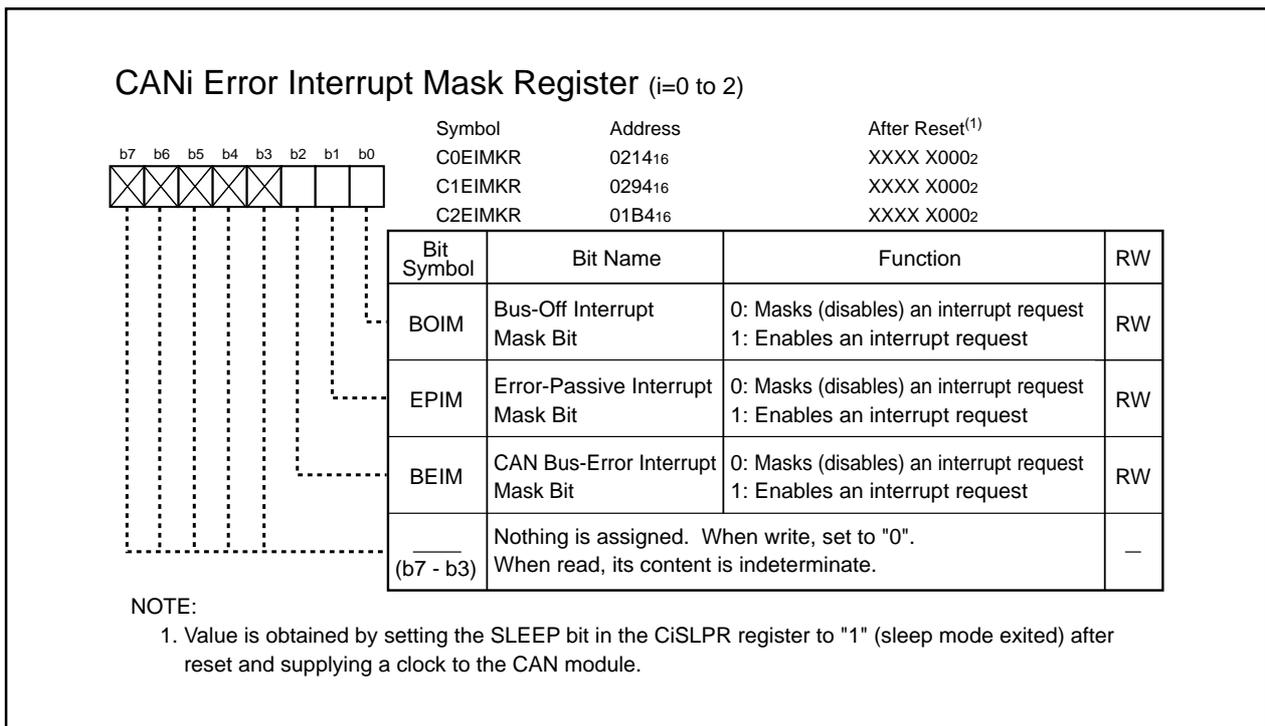
### 22.1.12 CANi Slot Interrupt Mask Register (CiSIMKR Register) (i=0 to 2)



**Figure 22.14 C0SIMKR, C1SIMKR and C2SIMKR Registers**

The CiSIMKR register determines whether an interrupt request, generated by a data transmission or reception in the corresponding message slot is enabled or disabled. When the SIM<sub>j</sub> bit (j=0 to 15) is set to "1" (no interrupt requested), an interrupt request generated by a data transmission or reception in the corresponding message slot is enabled. Refer to **22.4 CAN Interrupt** for details.

### 22.1.13 CAN<sub>i</sub> Error Interrupt Mask Register (CiEIMKR Register) (i=0 to 2)



**Figure 22.15 C0EIMKR, C1EIMKR and C2EIMKR Registers**

Refer to **22.4 CAN Interrupt** for details.

#### 22.1.13.1 BOIM Bit

The BOIM bit determines whether an interrupt request is enabled or disabled when the CAN module is placed in a bus-off state. When the BOIM bit is set to "1", the bus-off interrupt request is enabled.

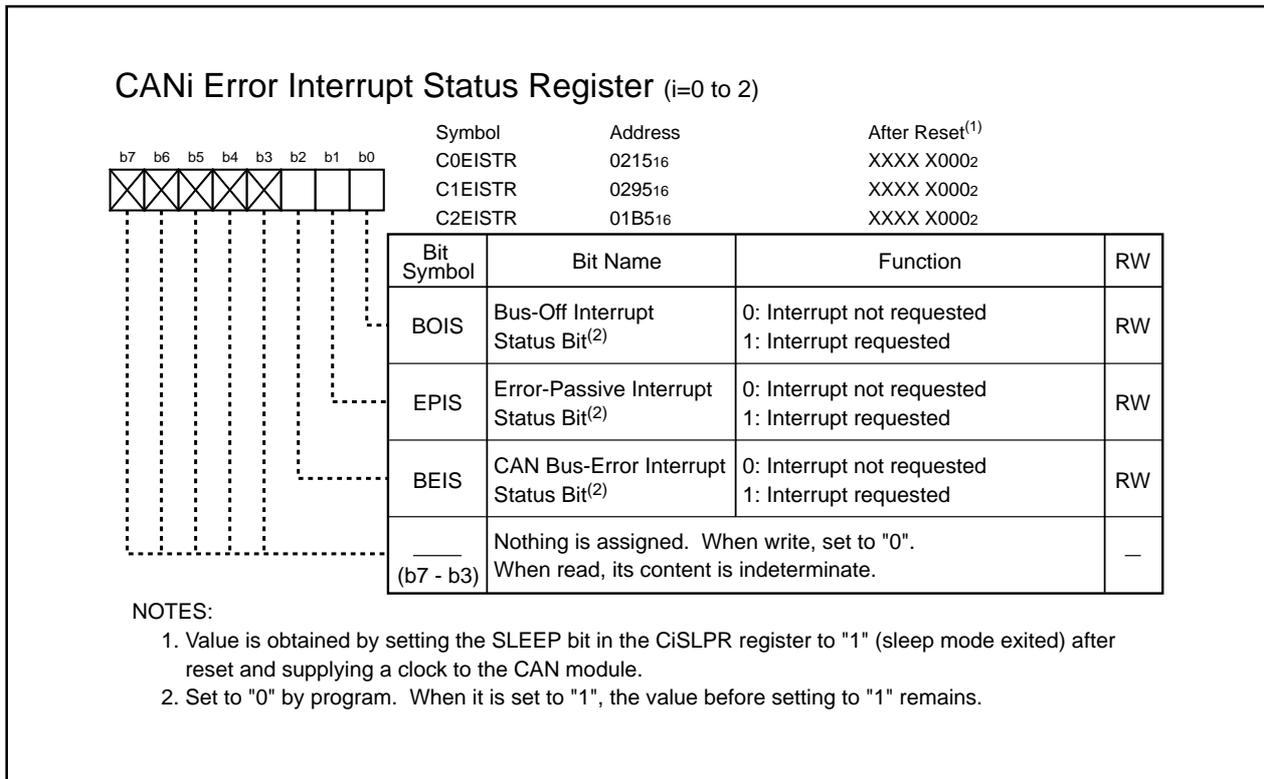
#### 22.1.13.2 EPIM Bit

The EPIM bit determines whether an interrupt request is enabled or disabled when the CAN module is placed in an error passive state. When the EPIM bit is set to "1", the error passive interrupt request is enabled.

#### 22.1.13.3 BEIM Bit

The BEIM bit determines whether an interrupt request is enabled or disabled when a CAN bus error occurs. When the BEIM bit is set to "1", the CAN bus error interrupt request is enabled.

### 22.1.14 CANi Error Interrupt Status Register (CiEISTR Register) (i=0 to 2)



**Figure 22.16 C0EISTR, C1EISTR and C2EISTR Registers**

When using the CAN interrupt, the CiEISTR register indicates the source of the generated error interrupt. The BOIS, EPIS and BEIS bits are not automatically set to "0" (no interrupt requested) even if an interrupt is acknowledged. Set these bits to "0" by program.

Use the MOV instruction, instead of the bit clear instruction, to set each bit in the CiEISTR register to "0".

Bits not being changed to "0" must be set to "1".

For example: To set the BOIS bit for CAN0 to "0"

Assembly language: `mov.b#006h, C0EISTR`

C language: `c0eistr = 0x06;`

Refer to **22.4 CAN Interrupt** for details.

#### 22.1.14.1 BOIS Bit

The BOIS bit is set to "1" when the CAN module is placed in a bus-off state.

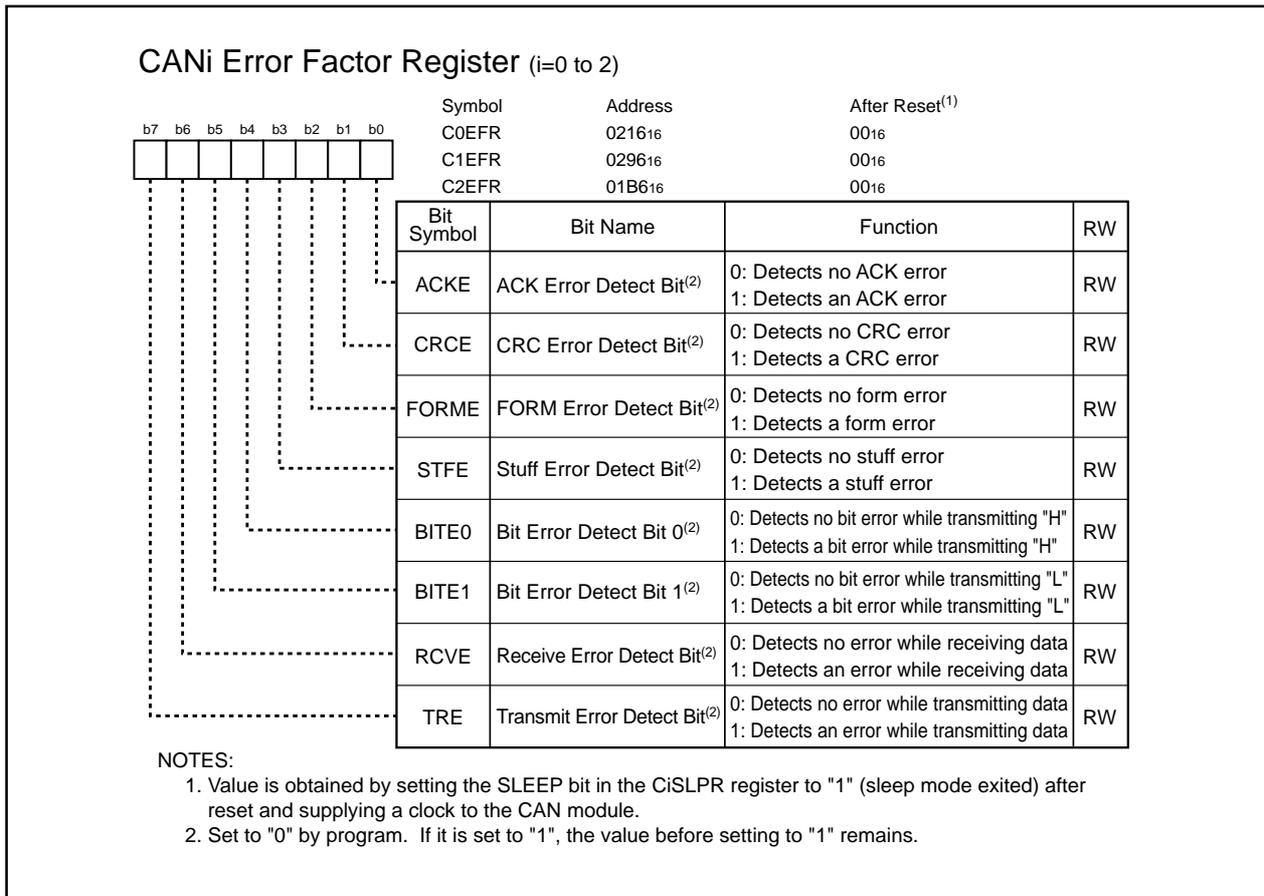
#### 22.1.14.2 EPIS Bit

The EPIS bit is set to "1" when the CAN module is placed in an error passive state.

#### 22.1.14.3 BEIS Bit

The BEIS bit is set to "1" when a CAN bus error is detected.

### 22.1.15 CAN<sub>i</sub> Error Factor Register (CiEFR Register) (i=0 to 2)



**Figure 22.17 C0EFR, C1EFR and C2EFR Registers**

The CiEFR register indicates the cause of error when a communication error is detected. Set the following bits to "0" by program because they are not changed "1" to "0" automatically.

Use the MOV instruction, instead of the bit clear instruction, to set each bit in the CiEFR register to "0".

Bits not being changed to "0" must be set to "1".

For example: To set the ACKE bit for CAN0 to "0"

Assembly language:    mov.b#0FEh, C0EFR

C language:            c0efr = 0xFE;

#### 22.1.15.1 ACKE Bit

The ACKE bit is set to "1" when an ACK error is detected.

#### 22.1.15.2 CRCE Bit

The CRCE bit is set to "1" when a CRC error is detected.

#### 22.1.15.3 FORME Bit

The FORME bit is set to "1" when a form error is detected.

#### 22.1.15.4 STFE Bit

The STFE bit is set to "1" when a stuff error is detected.

#### 22.1.15.5 BITE0 Bit

The BITE0 bit is set to "1" when a bit error is detected while transmitting recessive "H".

#### 22.1.15.6 BITE1 Bit

The BITE1 bit is set to "1" when a bit error is detected while transmitting dominant "L".

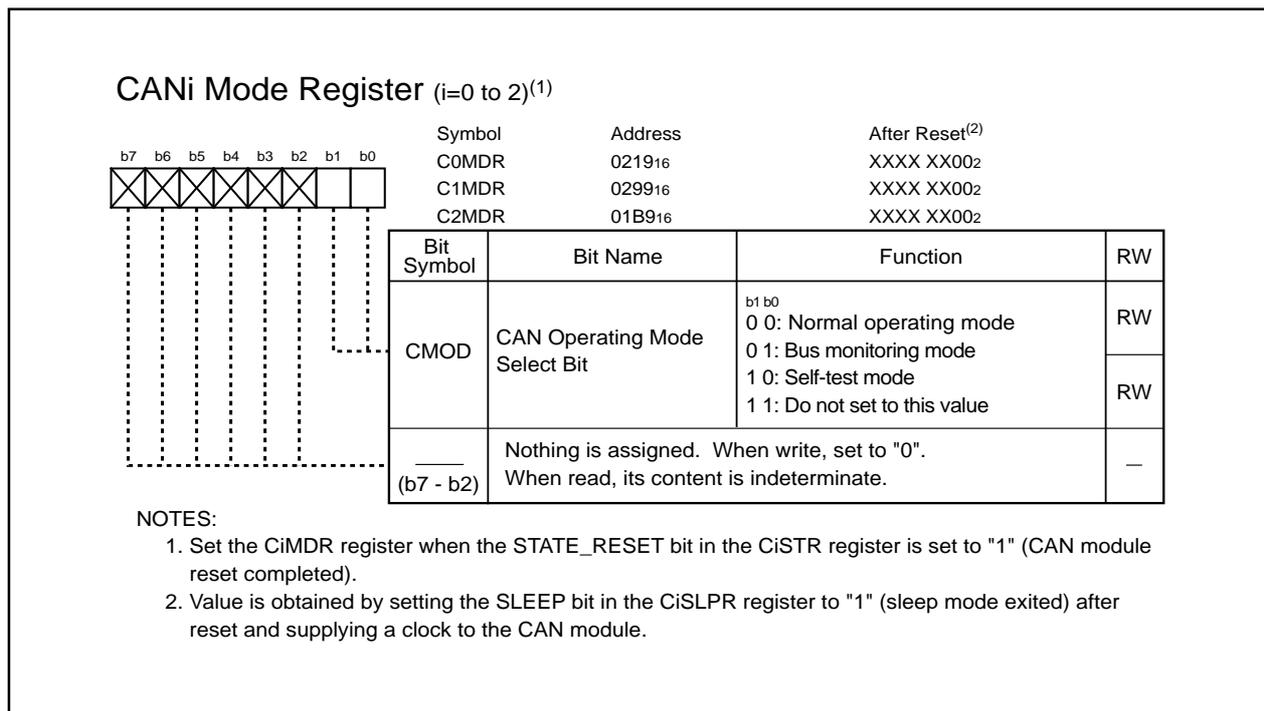
#### 22.1.15.7 RCVE Bit

The RCVE bit is set to "1" when an error is detected while receiving data.

#### 22.1.15.8 TRE Bit

The TRE bit is set to "1" when an error is detected while transmitting data.

### 22.1.16 CANi Mode Register (CiMDR Register) (i=0 to 2)



**Figure 22.18 C0MDR, C1MDR and C2MDR Registers**

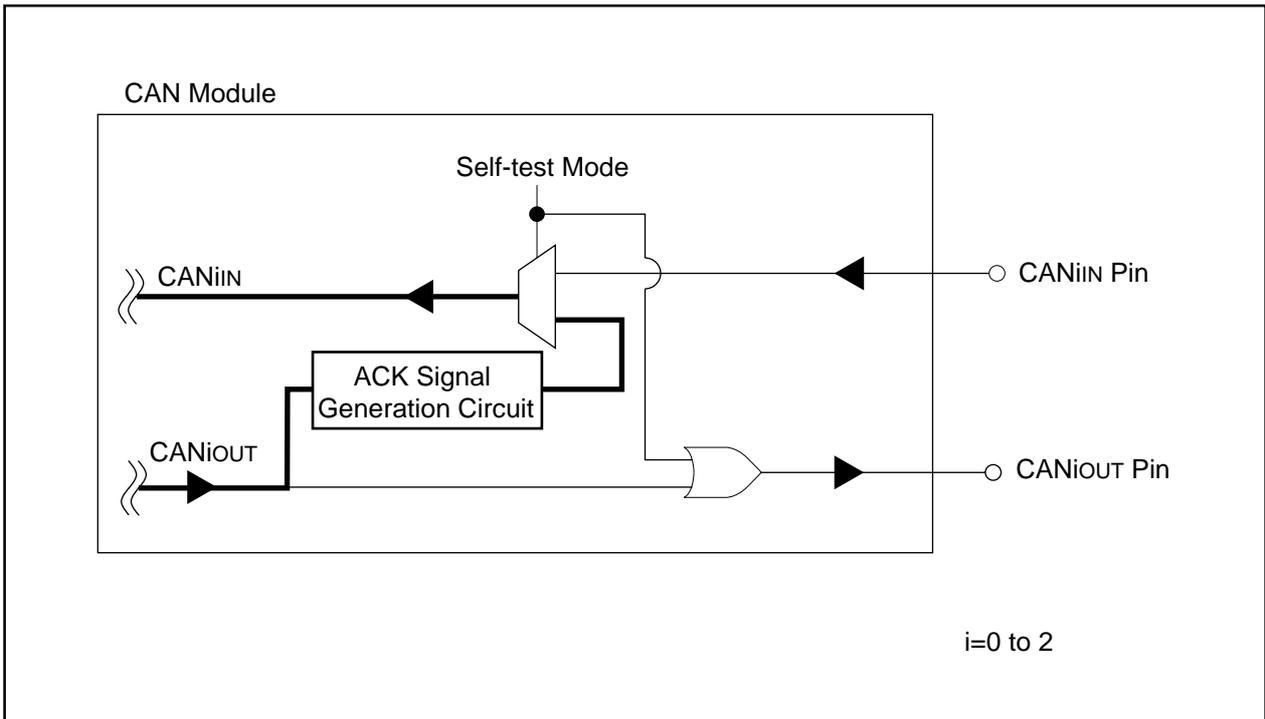
#### 22.1.16.1 CMOD Bit

The CMOD bit selects a CAN operating mode.

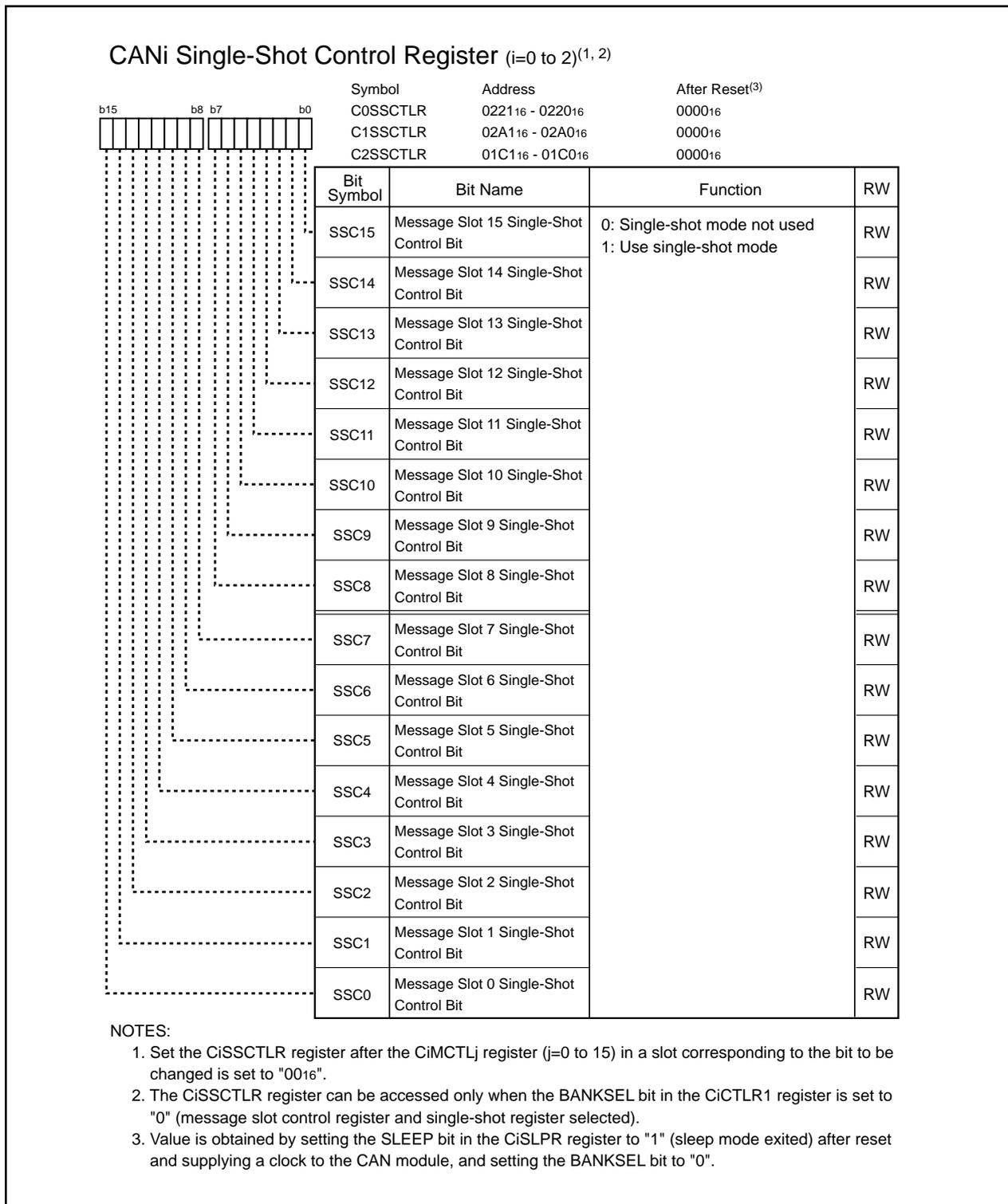
- Normal operating mode: The CAN module transmits and receives data as expected.
- Bus monitoring mode<sup>(1)</sup>: The CAN module receives data. Output signal from the CANiOUT pin is fixed as a high-level ("H") signal in bus monitoring mode. The CAN module transmits neither ACK nor error frame.
- Self-test mode: The CAN module connects the CANiOUT pin to the CANiIN pin internally. The CAN module can communicate without additional device in loop back mode. Output signal from the CANiOUT pin is fixed as an "H" signal in self-test mode while transmitting data. Figure 22.19 shows an image diagram in self-test mode.

**NOTE:**

- Do not generate a transmit request in bus monitoring mode.  
The CAN module assumes the ACK bit is set to dominant "L" regardless of the ACK bit setting. Therefore, when the CRC delimiter is received as expected, the CAN module determines the data is received with no error regardless of the ACK bit setting.

**Figure 22.19 Self-Test Mode**

### 22.1.17 CANi Single-Shot Control Register (CiSSCTLR Register) (i=0 to 2)

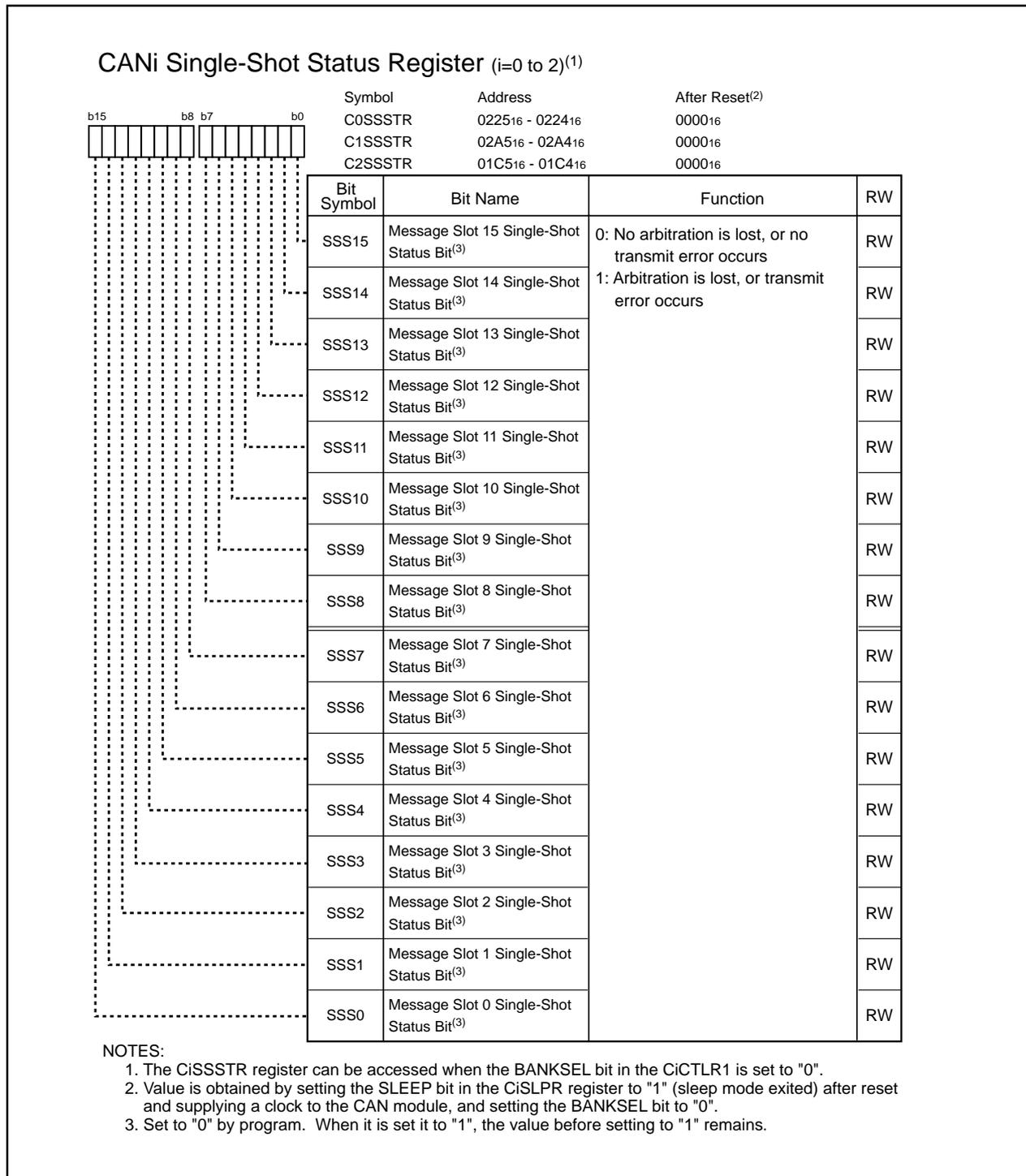


**Figure 22.20 C0SSCTLR, C1SSCTLR and C2SSCTLR Registers**

According to the CAN Specification 2.0 Part B, if the arbitration lost or transmission error causes a transmit failure, the microcomputer continues transmitting data until the transmission is completed. The CiSSCTLR register determines whether or not, and from which slot, data is re-transmitted.

In single-shot mode, if the arbitration lost or transmission error causes a transmission failure, data is not transmitted again. When the SSCj bit (j=0 to 15) is set to "1", the corresponding message slot j is in single-shot mode.

### 22.1.18 CAN<sub>i</sub> Single-Shot Status Register (CiSSSTR Register) (i=0 to 2)



**Figure 22.21 C0SSSTR, C1SSSTR and C2SSSTR Registers**

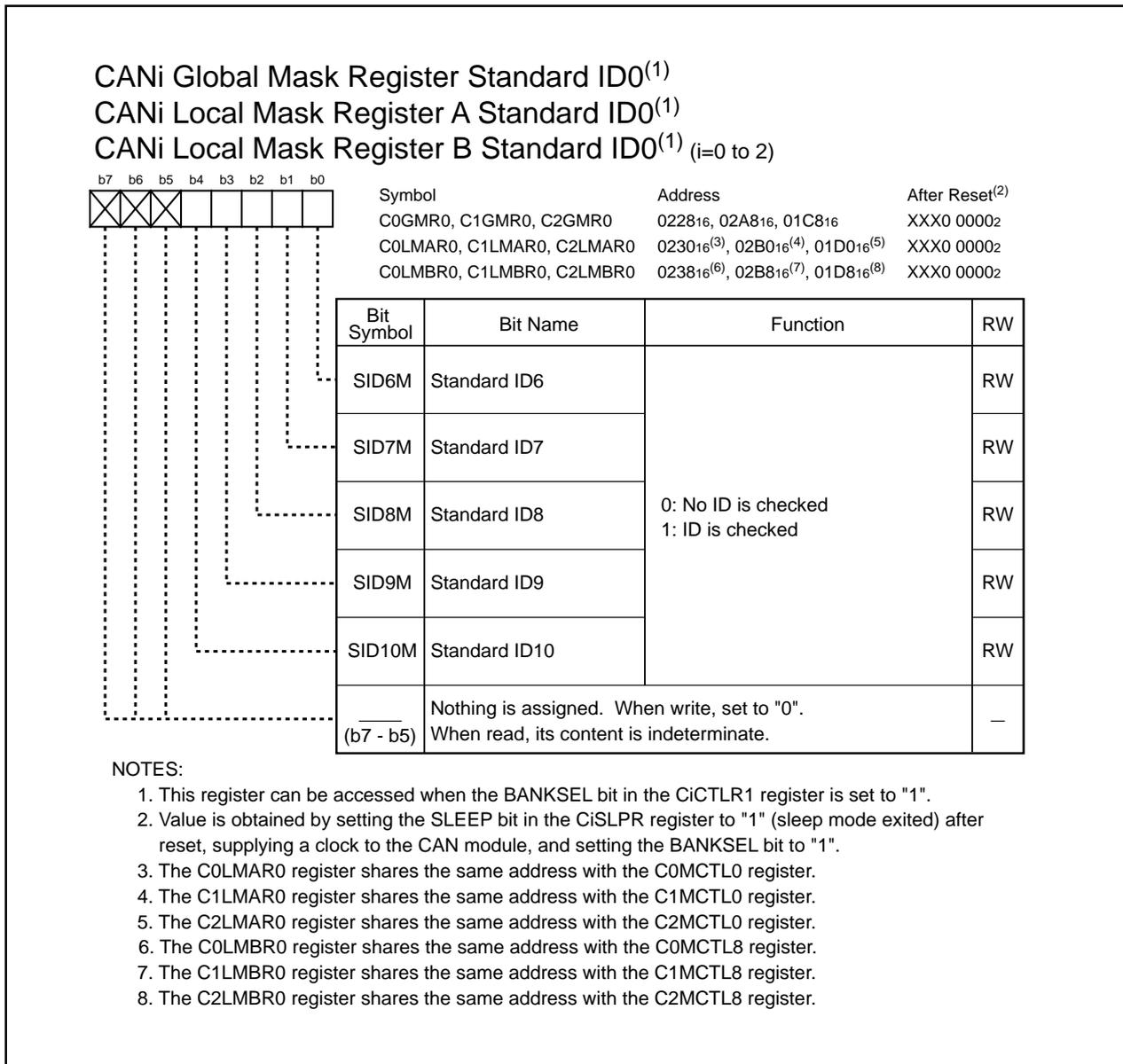
If the arbitration lost or transmission error causes a transmission failure, the bit corresponding to message slot  $j$  ( $j=0$  to 15) is set to "1". The SSS <sub>$j$</sub>  bit is set to "0" by program because it is not set to "0" automatically.

Use the MOV instruction, instead of the bit clear instruction, to set the SSS <sub>$j$</sub>  bit to "0". Bits not being changed to "0" must be set to "1".

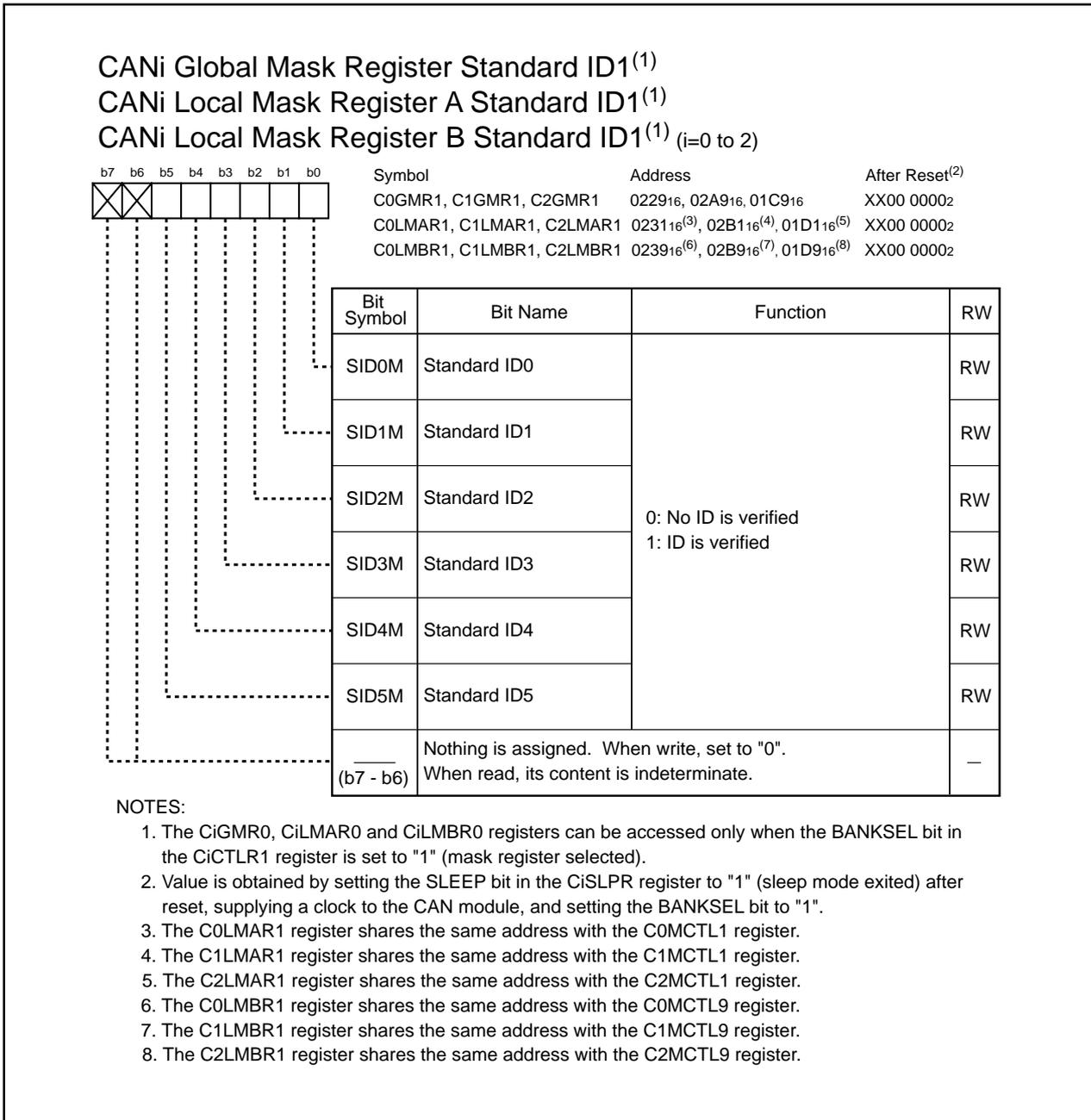
For example: To set the SSS<sub>0</sub> bit for CAN<sub>0</sub> to "0"

Assembly language: `mov.w #07FFFh, C0SSSTR`  
 C language: `c0sstr = 0x7FFF;`

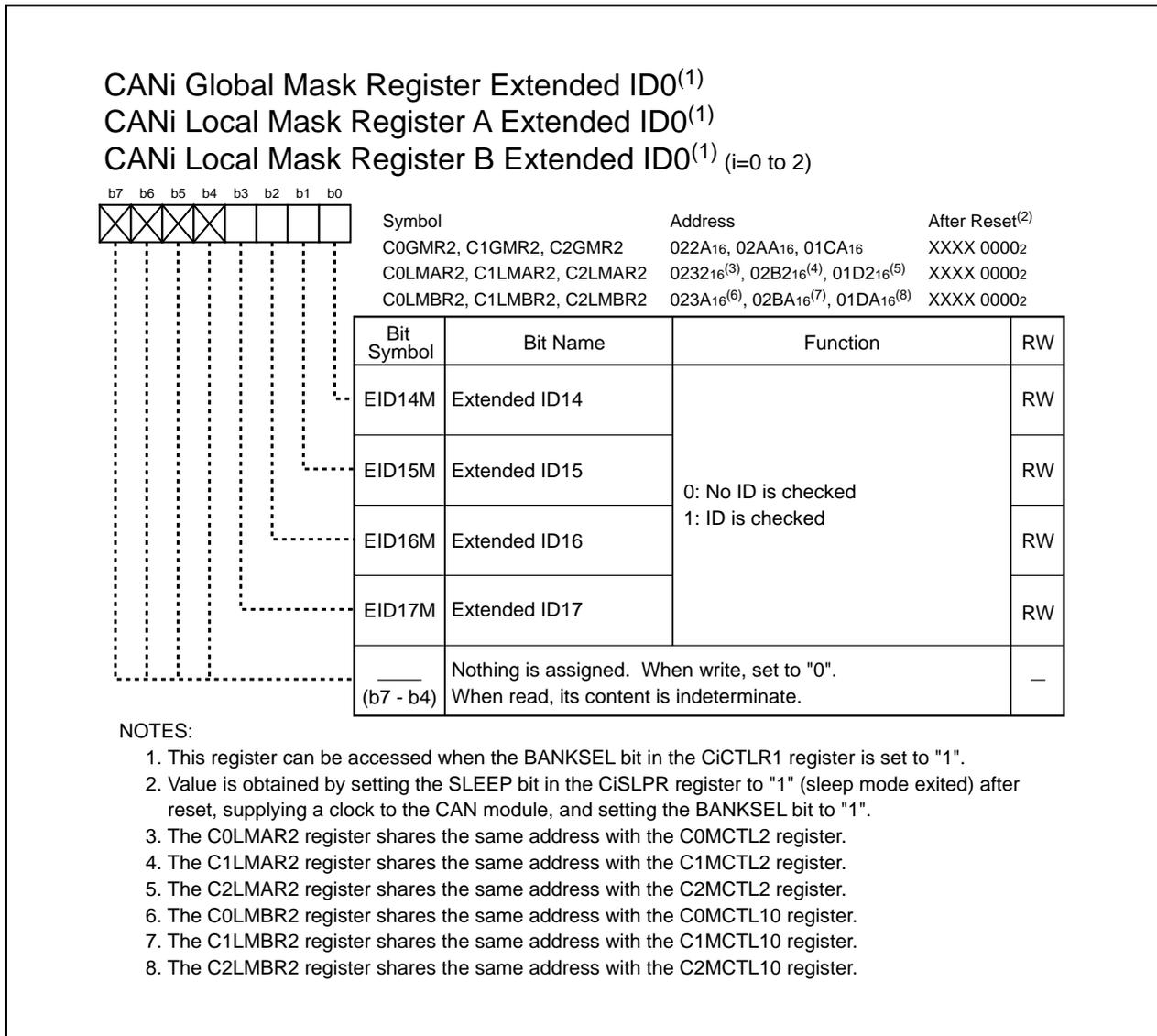
### 22.1.19 CANi Global Mask Register, CANi Local Mask Register A and CANi Local Mask Register B (CiGMRk, CiLMARk and CiLMBRk Registers) (i=0 to 2, k=0 to 4)



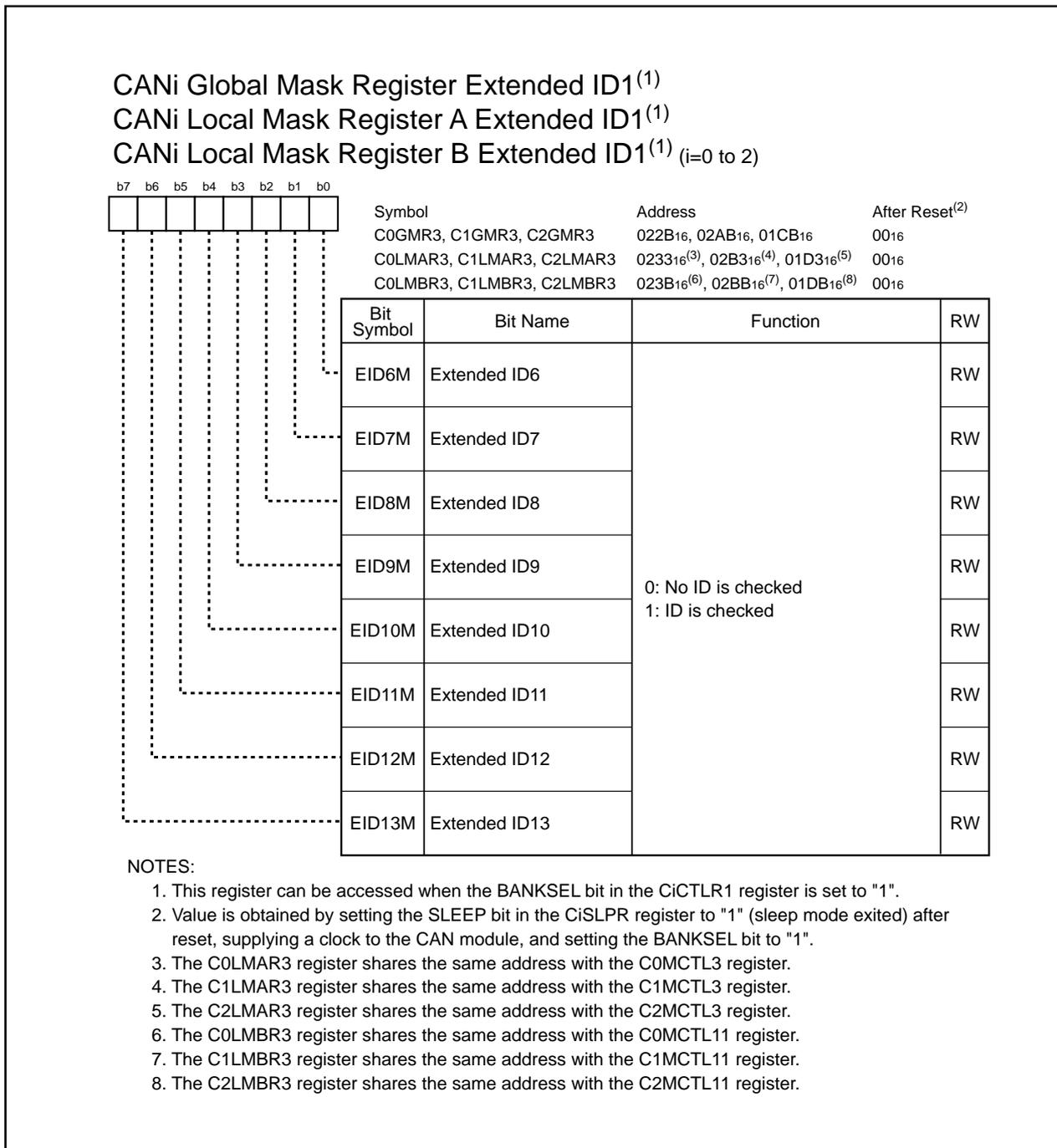
**Figure 22.22 C0GMR0, C0LMAR0 and C0LMBR0 Registers  
 C1GMR0, C1LMAR0 and C1LMBR0 Registers  
 C2GMR0, C2LMAR0 and C2LMBR0 Registers**



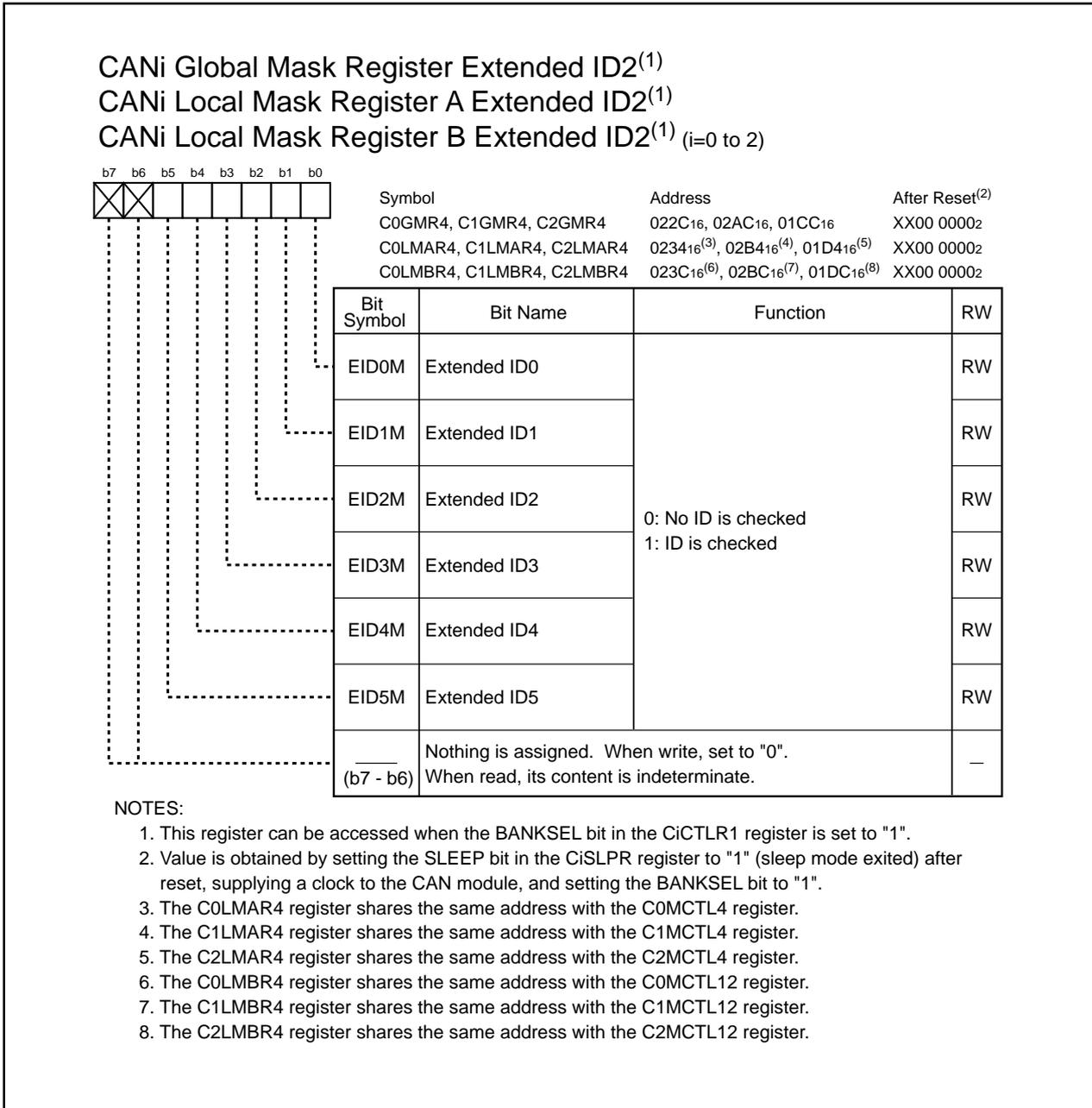
**Figure 22.23 C0GMR1, C0LMAR1 and C0LMBR1 Registers  
C1GMR1, C1LMAR1 and C1LMBR1 Registers  
C2GMR1, C2LMAR1 and C2LMBR1 Registers**



**Figure 22.24 C0GMR2, C0LMAR2 and C0LMBR2 Registers  
C1GMR2, C1LMAR2 and C1LMBR2 Registers  
C2GMR2, C2LMAR2 and C2LMBR2 Registers**



**Figure 22.25 C0GMR3, C0LMAR3 and C0LMBR3 Registers  
 C1GMR3, C1LMAR3 and C1LMBR3 Registers  
 C2GMR3, C2LMAR3 and C2LMBR3 Registers**



**Figure 22.26 C0GMR4, C0LMAR4 and C0LMBR4 Registers  
C1GMR4, C1LMAR4 and C1LMBR4 Registers  
C2GMR4, C2LMAR4 and C2LMBR4 Registers**

The CiGMRk, CiLMARk and CiLMBRk registers are used for acceptance filtering. The users can select and receive user-desired messages.

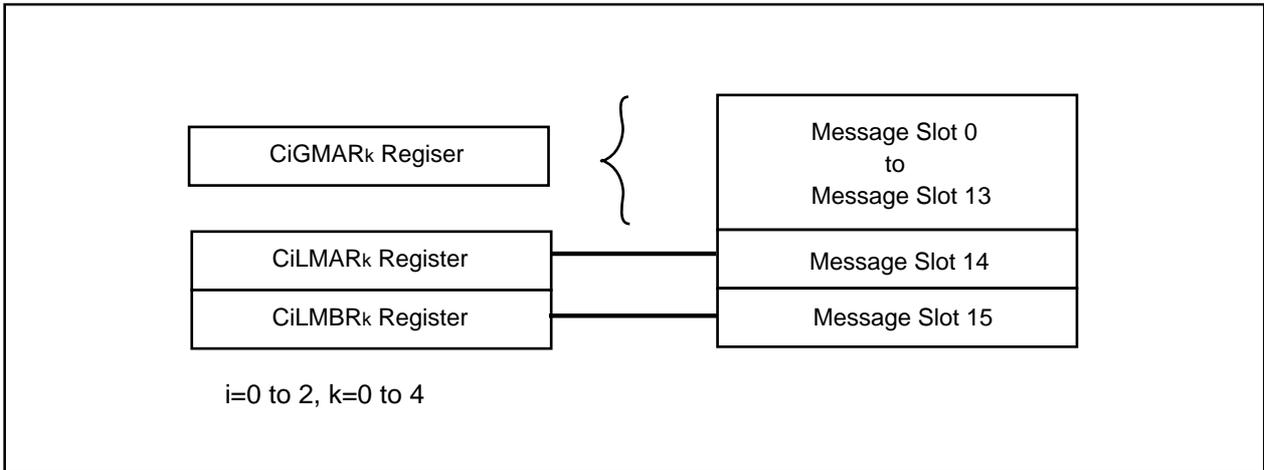
The CiGMRk register determines whether IDs in the message slots 0 to 13 are verified. The CiLMARk register determines whether ID in the message slot 14 is verified. The CiLMBRk register determines whether ID in the message slot 15 is verified.

- When bits in these registers are set to "0", each standard ID0 and standard ID1 bits (ID bit) and extended ID0 to extended ID2 bits in the CANi message slots j (j=0 to 15) corresponding to the bits in the above registers, is masked while acceptance filtering. (The corresponding bits are assumed to have matching IDs.)
- When bits in these registers are set to "1", corresponding ID bits are compared with received IDs while acceptance filtering. If the received ID matches the ID in the message slot j, the received data having the matched ID is stored into that message slot.

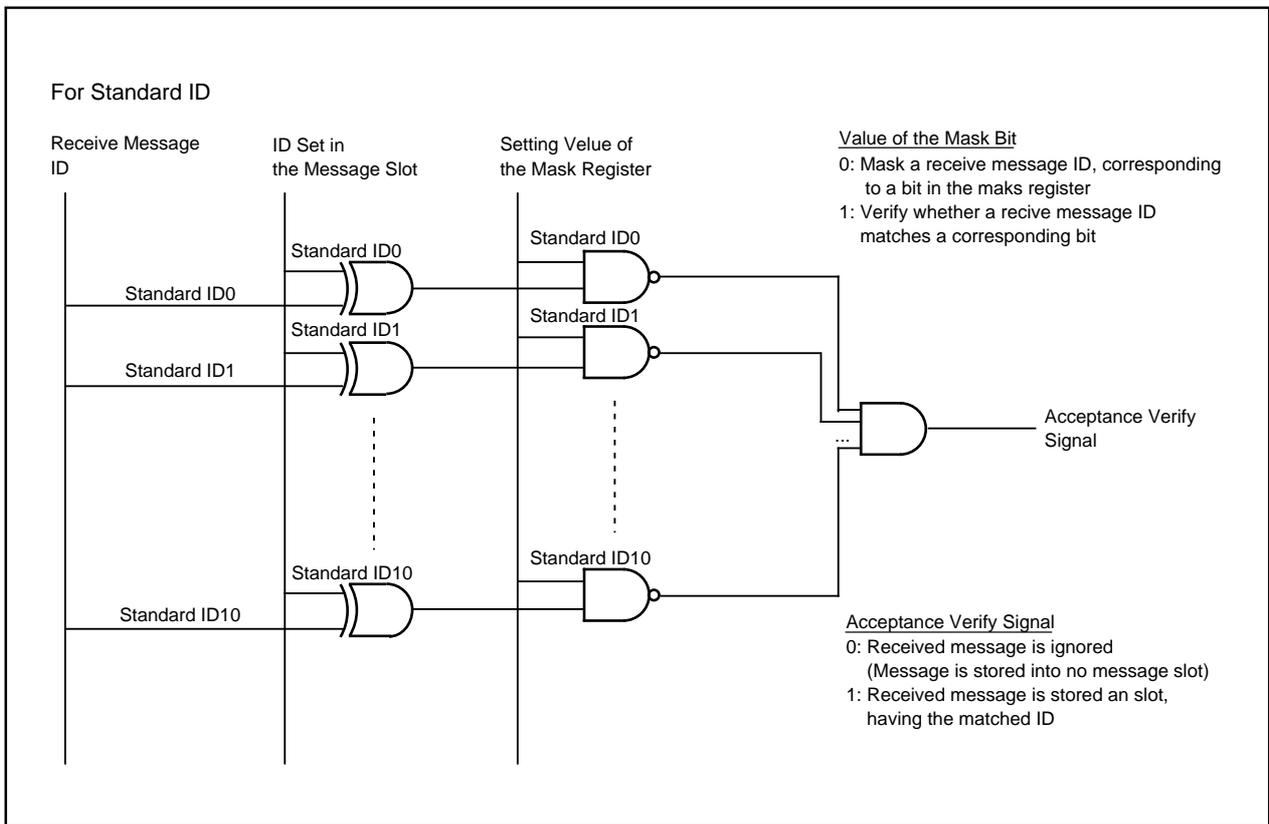
NOTES:

1. Change the CiGMRk register setting only when the message slots 0 to 13 have no receive request.
2. Change the CiLMARk register setting only when the message slot 14 has no receive request.
3. Change the CiLMBRk register setting only when the message slot 15 has no receive request.
4. More than two message slots are able to store a receive message ID, the ID is stored into the message slot, having the smallest slot number.

Figure 22.27 shows each mask register and corresponding message slot. Figure 22.28 shows the acceptance filtering.

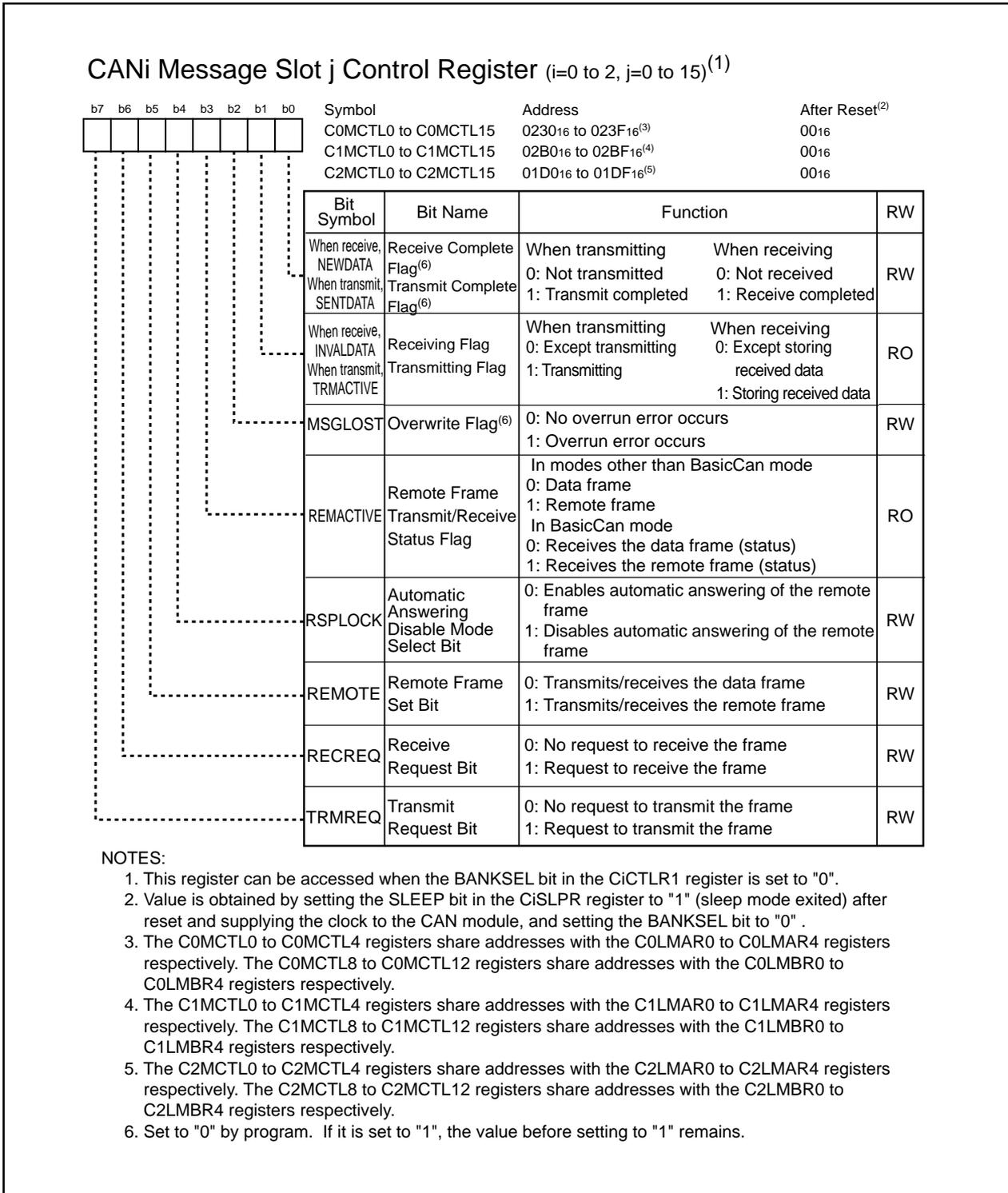


**Figure 22.27 Mask Registers and Message Slots**



**Figure 22.28 Acceptance Filtering**

**22.1.20 CANi Message Slot j Control Register (CiMCTLj Register) (i=0 to 2, j=0 to 15)**



**Figure 22.29 C0MCTL0 to C0MCTL15 Registers, C1MCTL0 to C1MCTL15 Registers and C2MCTL0 to C2MCTL15 Registers**

**Table 22.4 CiMCTLj register(i=0 to 2, j= 0 to 15) Settings and Transmit/Receive Mode**

Settings for the CiMCTLj Register								Transmit/Receive Mode
TRMREQ	RECREQ	REMOTE	RSPLOCK	REMACTIVE	MSGLOST	TRMACTIVE INVALDATA	SENTDATA NEWDATA	
0	0	0	0	0	0	0	0	No frame is transmitted or received
0	1	0	0	0	0	0	0	Data frame is received
0	1	1	1 or 0	0	0	0	0	Remote frame is received (The data frame is transmitted after receiving the remote frame.)
1	0	0	0	0	0	0	0	Data frame is transmitted
1	0	1	0	0	0	0	0	Remote frame is transmitted (The data frame is received after transmitting the remote frame)

**22.1.20.1 SENTDATA/NEWDATA Bit**

The SENTDATA/NEWDATA bit indicates that the CAN module has transmitted or received the CAN message. Set the SENTDATA/NEWDATA bit to "0" (not transmitted or not received) by program before data transmission and reception is started. The SENTDATA/NEWDATA bit is not set to "0" automatically. When the TRMACTIVE/INVALDATA bit is set to "1" (during transmission or storing received data), the SENTDATA/NEWDATA bit cannot be set to "0".

**SENTDATA :** The SENTDATA bit is set to "1" (transmit complete) when data transmission is completed in the transmit message slot.

**NEWDATA :** The NEWDATA bit is set to "1" (receive complete) when the message to be stored into the message slot j (j=0 to 15) is received in the receive message slot as expected.

**NOTES:**

1. To read a received data from the message slot j, set the NEWDATA bit to "0" before reading. If the NEWDATA bit is set to "1" immediately after reading, this indicates that new received data has been stored into the message slot while reading and the read data contains an indeterminate value. In this case, discard the data with indeterminate value and then read the message slot again after the NEWDATA bit is set to "0".
2. When the remote frame is transmitted or received, the SENTDATA/NEWDATA bit remains unchanged after the remote frame transmission or reception is completed. The SENTDATA/NEWDATA bit is set to "1" when a subsequent data frame transmission or reception is completed.

**22.1.20.2 TRMACTIVE/INVALDATA Bit**

The TRMACTIVE/INVALDATA bit indicates that the CAN protocol controller is transmitting or receiving a message and accessing the message slot j. The TRMACTIVE/INVALDATA bit is set to "1" when the CAN module is accessing the message slot and to "0" when not accessing the message slot.

**TRMACTIVE :** The TRMACTIVE bit is set to "1" (except transmitting) when a data transmission is started in the message slot. If the CAN module loses in bus arbitration, the TRMACTIVE bit is set to "0" (stops transmitting) when a CAN bus error occurs or when a data transmission is completed.

**INVALDATA :** The INVALDATA bit is set to "1" (storing received data) when receiving a received message into the message slot j, after a message reception is completed. Then the INVALDATA bit is set to "0" after a message storage is completed. Data, if read from the message slot j while this bit is set to "1", is indeterminate.

### 22.1.20.3 MSGLOST Bit

The MSGLOST bit is enabled only when the message slot is set for reception. The MSGLOST bit is set to "1" (overrun error occurred) when the message slot *j* is overwritten by a new received message while the NEWDATA bit set to "1" (already received).

The MSGLOST bit is not automatically set to "0". Set to "0" (no overrun error occurred) by program.

### 22.1.20.4 REMACTIVE Bit

The CiMCTL0 to CiMCTL15 registers all have the same function when the STATE\_BASICCAN bit is set to "0" (other than BasicCAN mode).

The REMACTIVE bit is set to "1" (remote frame) when the message slot *j* is set to transmit or receive the remote frame. The REMACTIVE bit is set to "0" (data frame) after the remote frame has been transmitted or received.

The functions of the CiMCTL14 and CiMCTL15 registers change when the STATE\_BASICCAN bit is set to "1" (BasicCAN mode). When the REMACTIVE bit is set to "0", this indicates that a message stored into the message slot is the data frame. When the REMACTIVE bit is set to "1", this indicates a message stored into the message slot is the remote frame.

### 22.1.20.5 RSPLOCK Bit

The RSPLOCK bit is enabled only when remote frame reception shown in Table 22.4 is selected. The RSPLOCK bit determines whether the received remote frame is processed or not.

When the RSPLOCK bit is set to "0" (automatic answering of the remote frame enabled), the slot automatically changes to a transmit slot after the remote frame is received and the message stored into the message slot is automatically transmitted as the data frame.

When the RSPLOCK bit is set to "1" (automatic answering of the remote frame disabled), message is not automatically transmitted upon receiving the remote frame.

Set the RSPLOCK bit to "0" to select any transmit/receive mode other than the remote frame reception.

### 22.1.20.6 REMOTE Bit

The REMOTE bit selects transmit/receive mode shown in Table 22.4. Set the REMOTE bit to "0" to transmit or receive data frame. Set to "1" to transmit or receive remote frame.

The followings occur during remote frame transmission or reception.

- Transmitting the remote frame

A message stored into the message slot *j* (*j*=0 to 15) is transmitted as the remote frame. After transmission, the slot automatically becomes ready to receive data frame.

If the data frame is received before the remote frame is transmitted, the data frame is stored into the message slot *j*. The remote frame is not transmitted.

- Receiving the remote frame

The message slot receives the remote frame. The RSPLOCK bit determines whether or not to process the received remote frame.

**22.1.20.7 RECREQ Bit**

The RECREQ bit selects transmit/receive mode shown in Table 22.4. Set the RECREQ bit to "1" (receive requested) when data frame or remote frame is received. Set the RECREQ bit to "0" (no receive requested) when data frame or remote frame is transmitted.

When a data frame is automatically transmitted after a remote frame is received, the RECREQ bit remains set to "1". Set the RECREQ bit to "0" to transmit a remote frame. After a remote frame is transmitted, a data frame is automatically received while the RECREQ bit remains set to "0".

When setting the TRMREQ bit to "1" (transmit requested), do not set the RECREQ bit to "1" (receive requested).

**22.1.20.8 TRMREQ Bit**

The TRMREQ bit selects transmit/receive mode shown in Table 22.4. Set the TRMREQ bit to "1" (transmit requested) when data frame or remote frame is transmitted.

Set the TRMREQ bit to "0" (no request to transmit the frame) when data frame or remote frame is received.

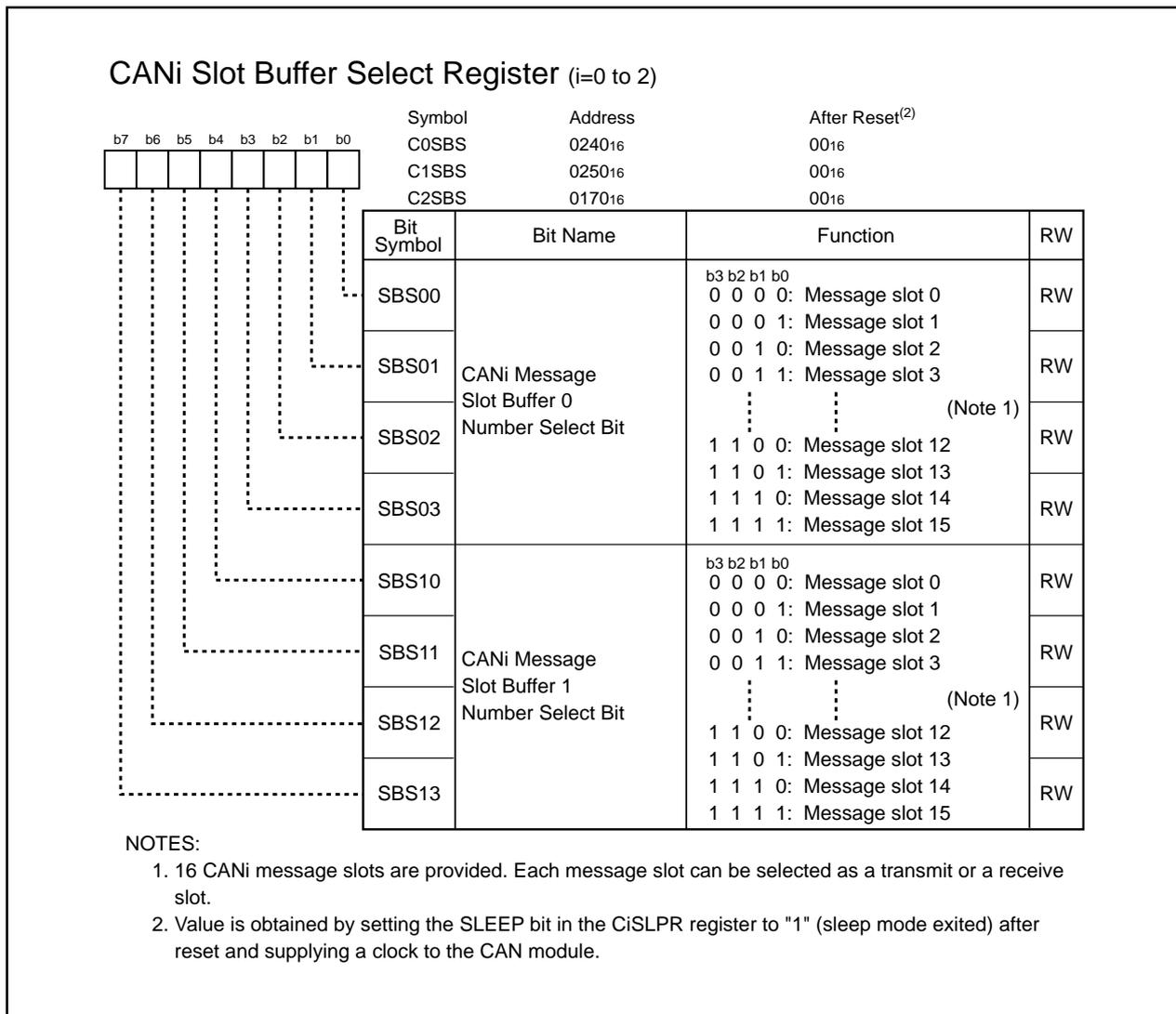
When the data frame is automatically received after the remote frame is transmitted, the TRMREQ bit remains set to "1". Set the TRMREQ bit to "0" to receive the remote frame. After the remote frame is received, data frame is automatically transmitted while the TRMREQ bit remains set to "0".

If the RECREQ bit is set to "1" (request to receive the frame), do not set the TRMREQ bit to "1" (request to transmit the frame).

**NOTES:**

1. If some message slots are requested to transmit the data frame or remote frame, the message slot, having the smallest slot number starts transmitting.
2. In single-shot mode, the CiMCTLj register is set to "00<sub>16</sub>" when data transmission is failed, due to the arbitration lost or transmission error.

### 22.1.21 CANi Slot Buffer Select Register (CiSBS Register) (i=0 to 2)



**Figure 22.30 C0SBS, C1SBS and C2SBS Registers**

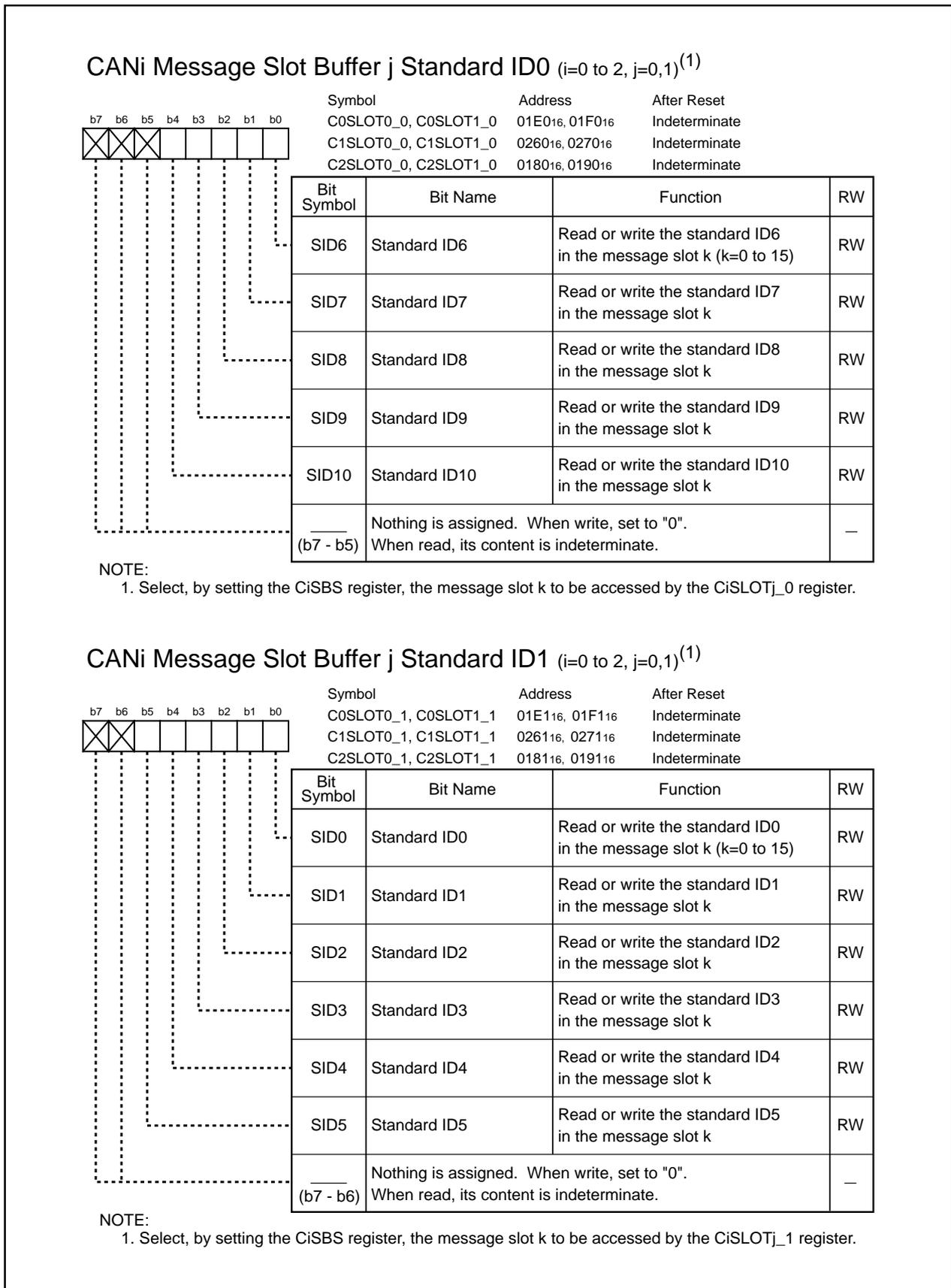
#### 22.1.21.1 SBS03 to SBS00 Bits

If the SBS03 to SBS00 bits select a number  $j$  ( $j=0$  to  $15$ ), the message slot  $j$  is allocated to the CANi message slot buffer 0. The message slot  $j$  can be accessed via addresses 01E0<sub>16</sub> to 01EF<sub>16</sub>, and 0260<sub>16</sub> to 026F<sub>16</sub>.

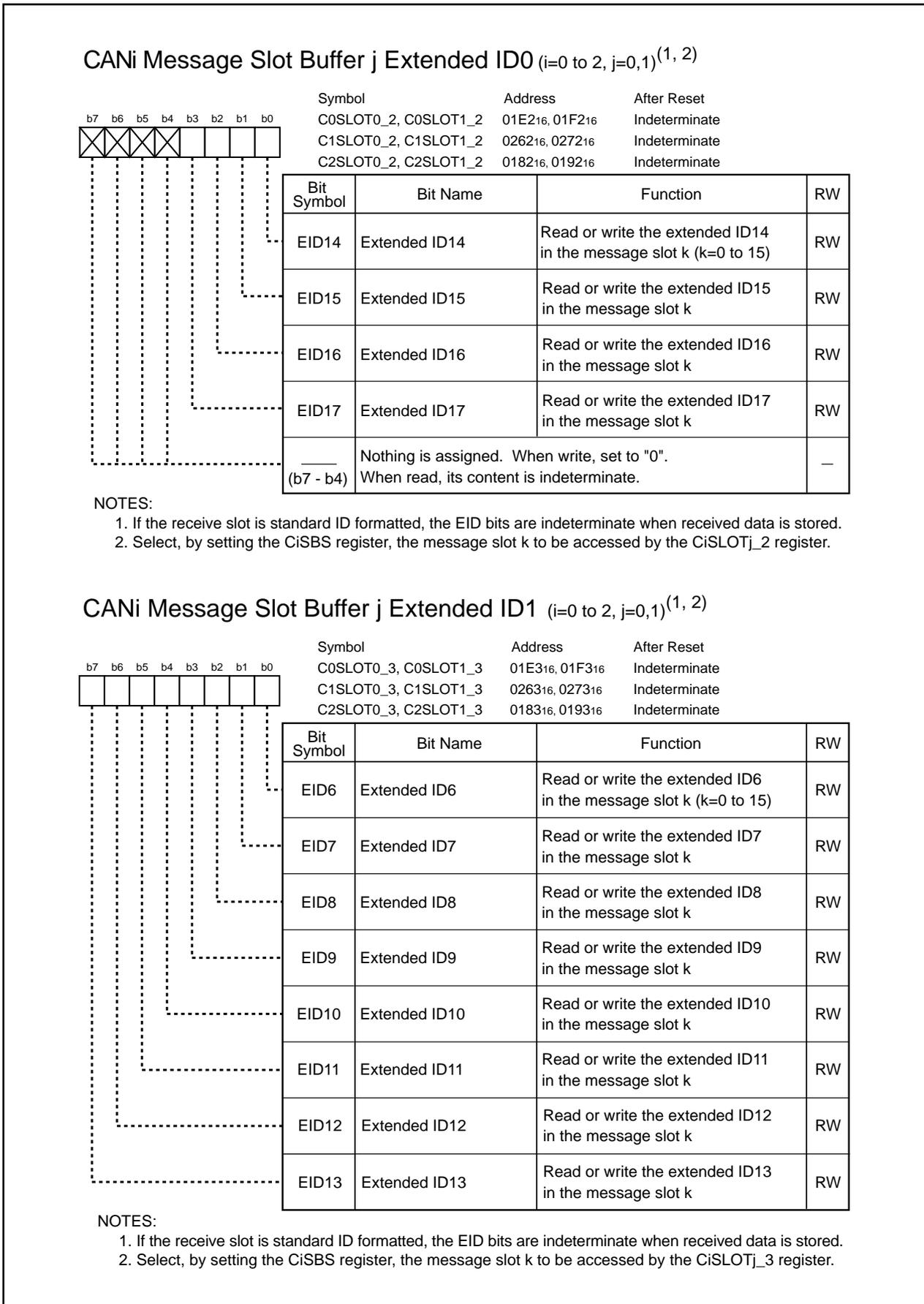
#### 22.1.21.2 SBS13 to SBS10 Bits

If the SBS13 to SBS10 bits select a number  $j$ , the message slot  $j$  is allocated to the CANi message slot buffer 1. The message slot  $j$  can be accessed via addresses 01F0<sub>16</sub> to 01FF<sub>16</sub>, and 0270<sub>16</sub> to 027F<sub>16</sub>.

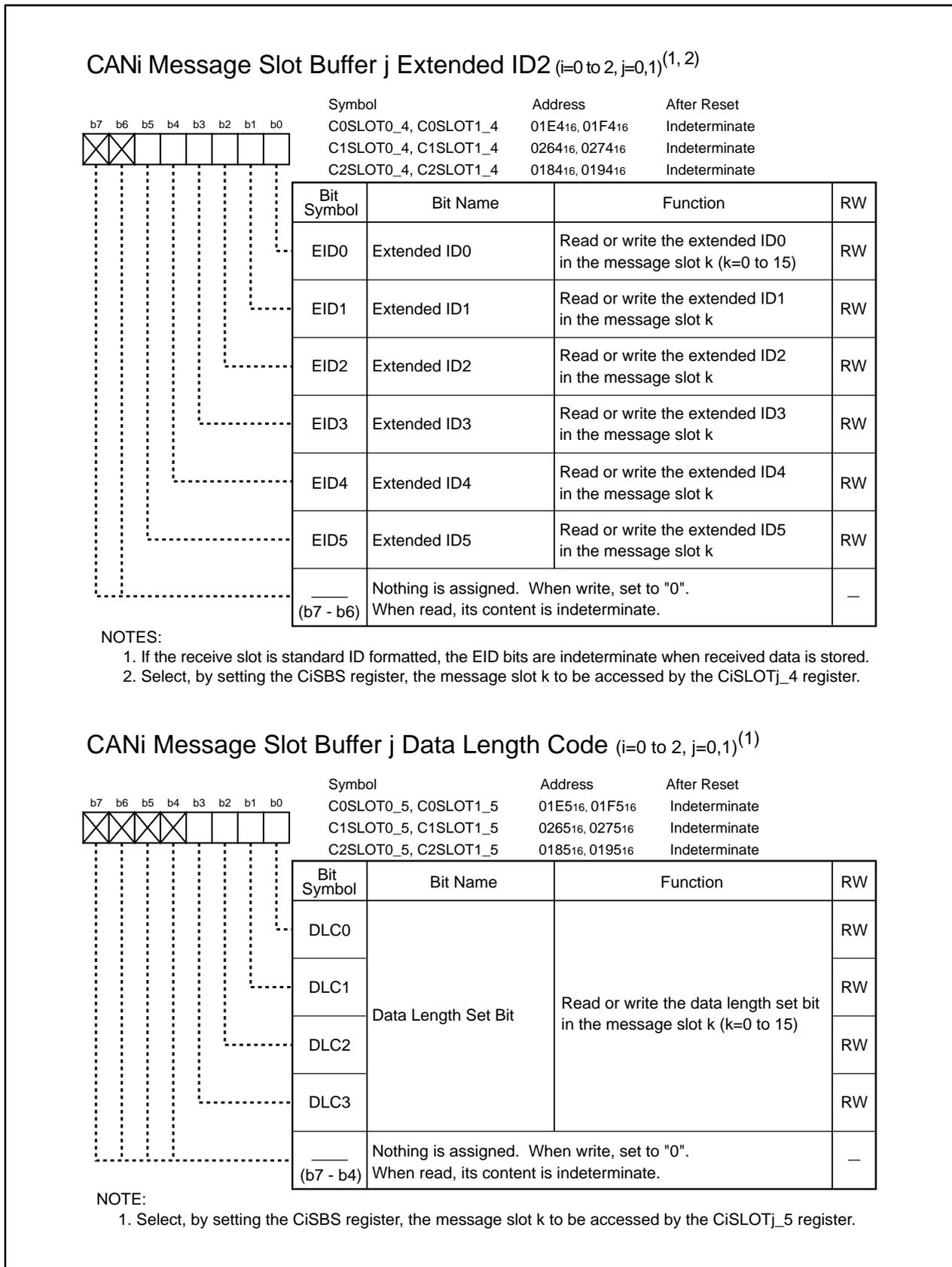
### 22.1.22 CANi Message Slot Buffer j (i=0 to 2, j=0,1)



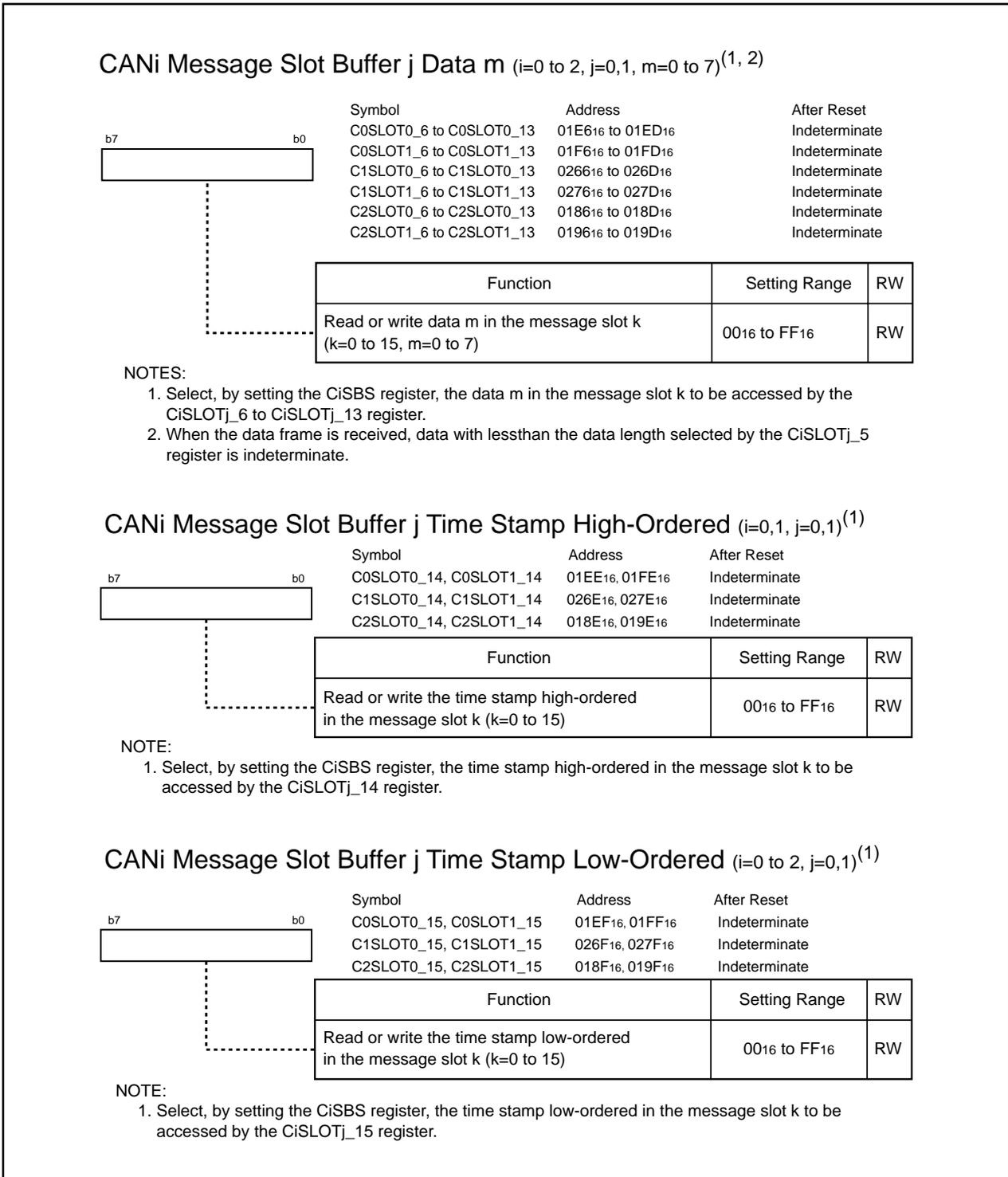
**Figure 22.31 C0SLOT0\_0, C0SLOT1\_0, C0SLOT0\_1 and C0SLOT1\_1 Registers  
C1SLOT0\_0, C1SLOT1\_0, C1SLOT0\_1 and C1SLOT1\_1 Registers  
C2SLOT0\_0, C2SLOT1\_0, C2SLOT0\_1 and C2SLOT1\_1 Registers**



**Figure 22.32 C0SLOT0\_2, C0SLOT1\_2, C0SLOT0\_3 and C0SLOT1\_3 Registers  
C1SLOT0\_2, C1SLOT1\_2, C1SLOT0\_3 and C1SLOT1\_3 Registers  
C2SLOT0\_2, C2SLOT1\_2, C2SLOT0\_3 and C2SLOT1\_3 Registers**



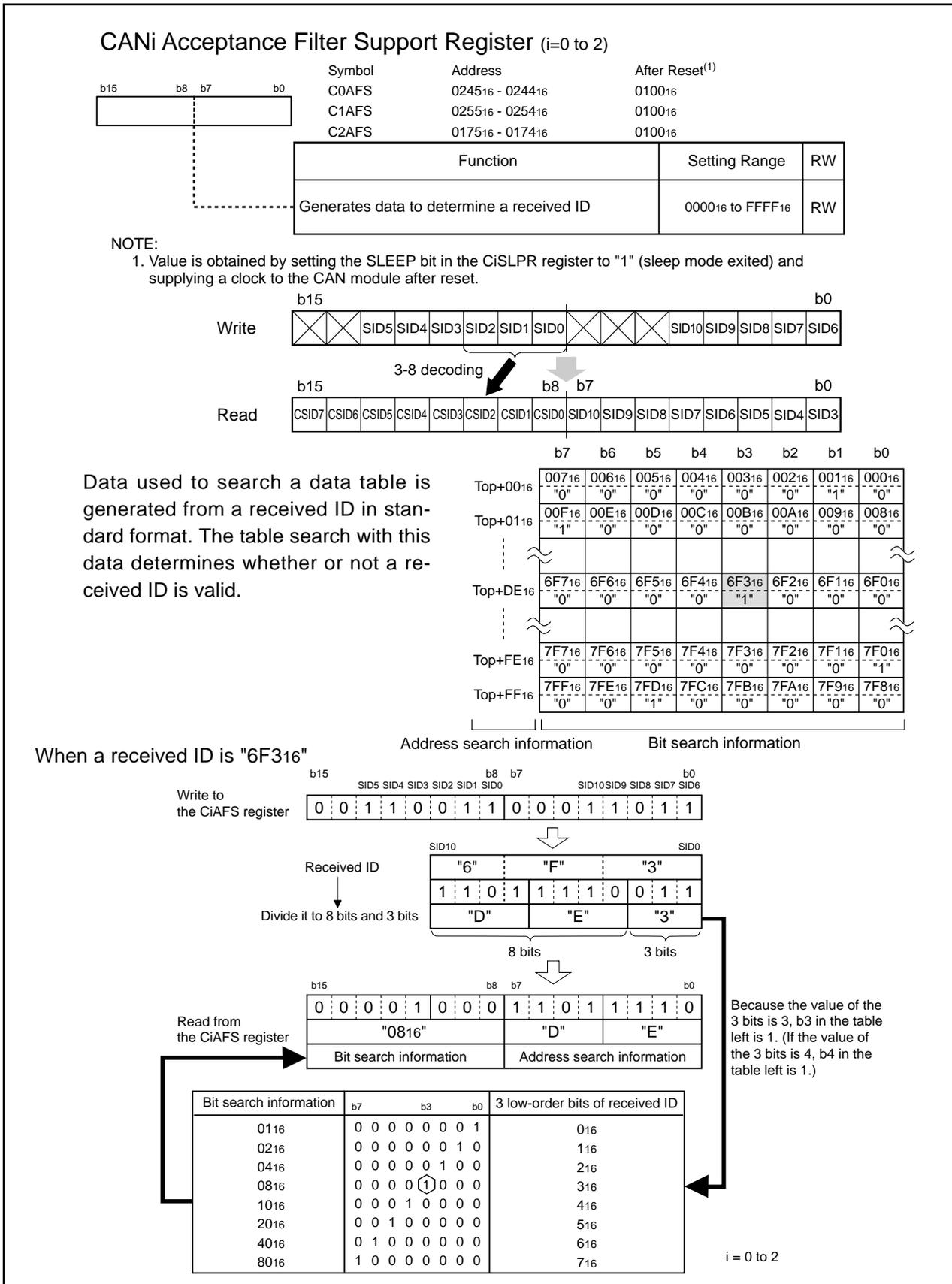
**Figure 22.33 C0SLOT0\_4, C0SLOT1\_4, C0SLOT0\_5 and C0SLOT1\_5 Registers  
C1SLOT0\_4, C1SLOT1\_4, C1SLOT0\_5 and C1SLOT1\_5 Registers  
C2SLOT0\_4, C2SLOT1\_4, C2SLOT0\_5 and C2SLOT1\_5 Registers**



**Figure 22.34 C0SLOT0\_6 to C0SLOT0\_13, C0SLOT1\_6 to C0SLOT1\_13, C0SLOT0\_14, C0SLOT1\_14, C0SLOT0\_15 and C0SLOT1\_15 Registers  
C1SLOT0\_6 to C1SLOT0\_13, C1SLOT1\_6 to C1SLOT1\_13, C1SLOT0\_14, C1SLOT1\_14, C1SLOT0\_15 and C1SLOT1\_15 Registers  
C2SLOT0\_6 to C2SLOT0\_13, C2SLOT1\_6 to C2SLOT1\_13, C2SLOT0\_14, C2SLOT1\_14, C2SLOT0\_15 and C2SLOT1\_15 Registers**

The message slot, selected by setting the CiSBS register, is read by reading the message slot buffer. A message can be written in the message slot selected by the CiSBS register if the message is written to the message slot buffer.  
Write to the message slot k (k=0 to 15) while the corresponding CiMCTLk register is set to "00<sub>16</sub>".

### 22.1.23 CANi Acceptance Filter Support Register (CiAFS Register) (i=0 to 2)



**Figure 22.35 C0AFS, C1AFS and C2AFS Registers**

The CiAFS register enables prompt performance of the table search to determine the validity of a received ID. This function is for standard-formatted ID only.

## 22.2 CAN Clock

The CAN clock is the operating clock for the CAN module.  $f_1$  or  $f_{CAN}$  can be selected as the CAN clock.  $f_{CAN}$  has the same frequency as the main clock. The PM25 bit in the PM2 register determines the CAN clock. Refer to **8. Clock Generation Circuit** for details.

### 22.2.1 Main Clock Direct Mode

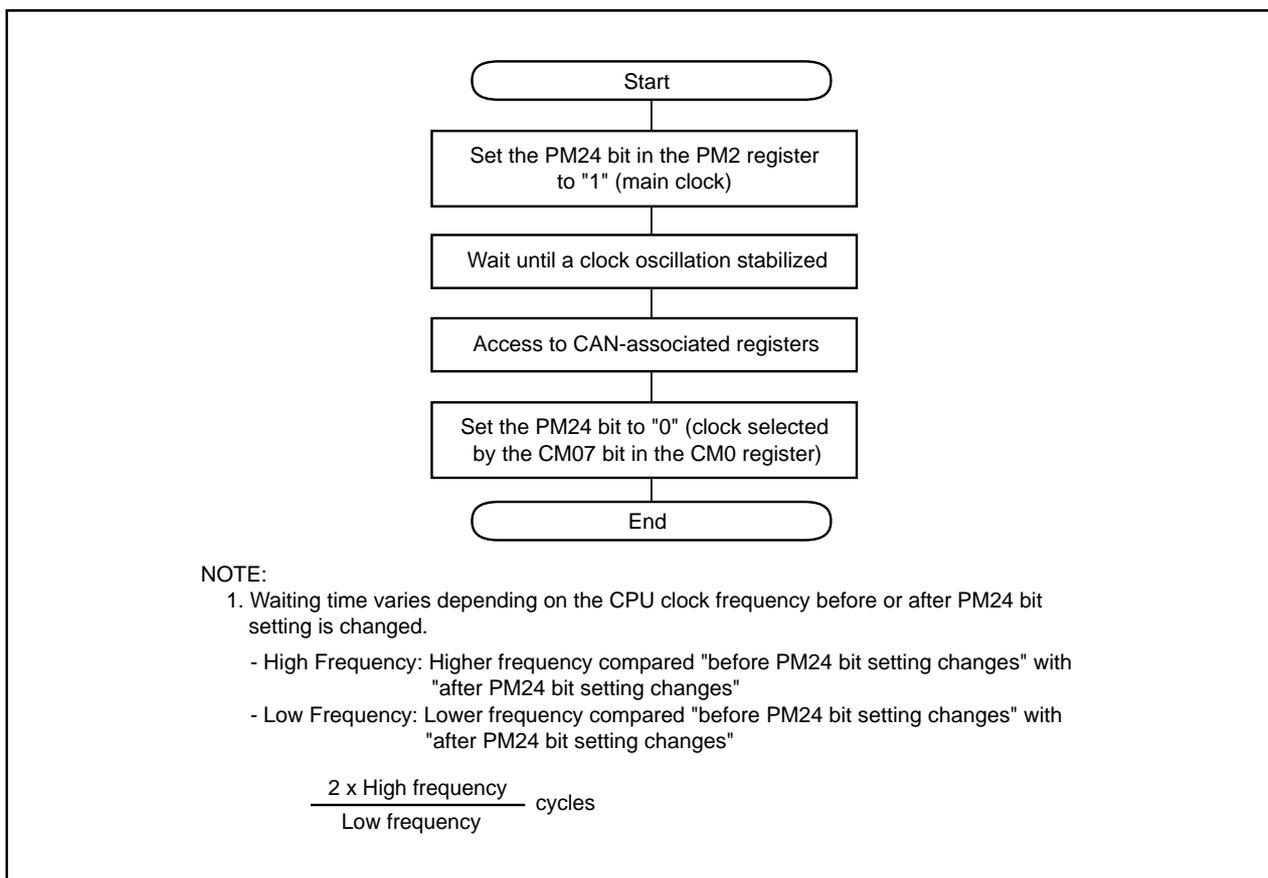
$f_{CAN}$  becomes the CAN clock in main clock direct mode. The CAN module must enter main clock direct mode while the PM25 bit is set to "1" (main clock). Set the PM25 bit in CAN sleep mode.

Set the PM24 bit in the PM2 register to "1" (main clock) before accessing CAN-associated registers in main clock direct mode. Do not enter wait mode or stop mode when the PM24 bit is set to "1".

Table 22.5 lists CAN clock settings. Figure 22.36 shows a flow chart of accessing procedure for CAN-associated registers.

**Table 22.5 CAN Clock Settings**

CAN Clock	Clock Source	CM0 Register	CM1 Register	CM2 Register	PM2 Register		MCD Register
		CM07 Bit	CM17 Bit	CM21 Bit	PM24 Bit	PM25 Bit	MCD4 to MCD0 bits
$f_{CAN}$	Main Clock (Main Clock Direct Mode)	0	1	0	1	1	---
$f_1$	Main Clock	0	0	0	0	0	10010 <sub>2</sub>
	PLL Clock	0	1	0	0	0	10010 <sub>2</sub>



**Figure 22.36 Accessing Procedure for CAN-Associated Registers**

## 22.3 Timing with CAN-Associated Registers

### 22.3.1 CAN Module Reset Timing

Figure 22.37 shows an operation example of when the CAN module is reset.

- (1) The CAN module can be reset when the STATE\_RESET bit in the CiSTR register (i=0 to 2) is set to "1" (CAN module reset completed) after the RESET1 and RESET0 bits in the CiCTRL0 register are set to "1" (CAN module reset).
- (2) Set necessary CAN-associated registers.
- (3) CAN communication can be established after the STATE\_RESET bit is set to "0" (resetting) after the RESET1 and RESET0 bits are set to "0" (CAN module reset exited) .

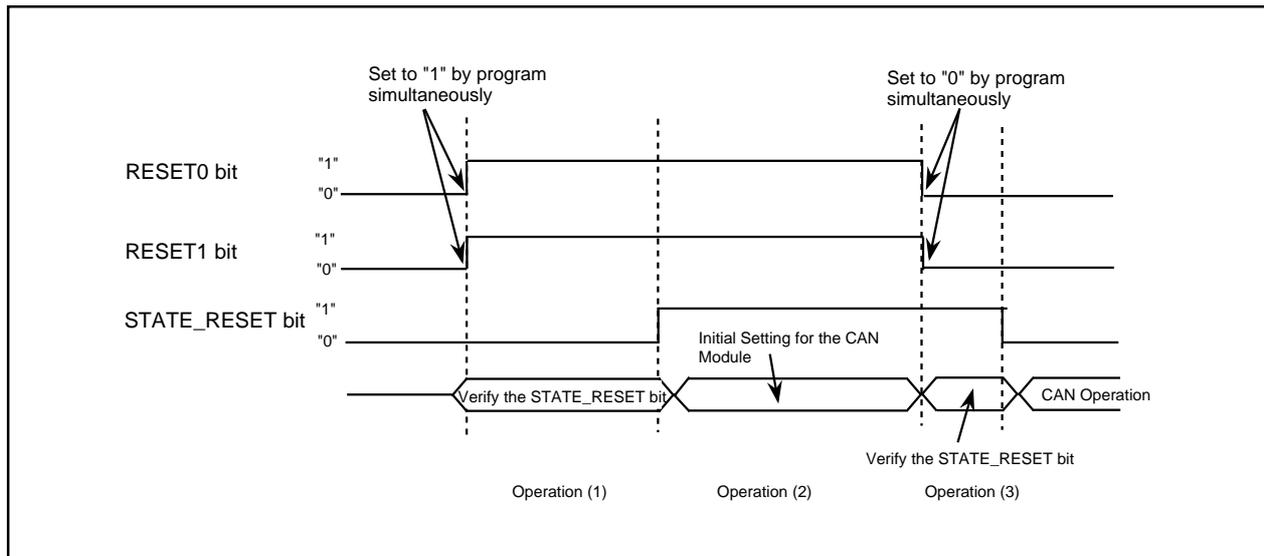


Figure 22.37 Example of CAN Module Reset Operation

### 22.3.2 CAN Transmit Timing

Figure 22.38 shows an operation example of when the CAN transmits a frame.

- (1) When the TRMREQ bit in the CiMCTLj register (j=0 to 15) is set to "1" (request to transmit the data frame) while the CAN bus is in an idle state, the TRMACTIVE bit in the CiMCTLj register is set to "1" (during transmission) and the TRMSTATE bit in the CiSTR register is set to "1" (during transmission). The CAN starts transmitting the frame.
- (2) After a CAN frame transmission is completed, the SENTDATA bit in the CiMCTLj register is set to "1" (already transmitted), the TRMSUCC bit in the CiSTR register to "1" (transmission completed) and the SISj bit in the CiSISTR register to "1" (interrupt requested). The MBOX3 to MBOX0 bits in the CiSTR register store transmitted message slot numbers.

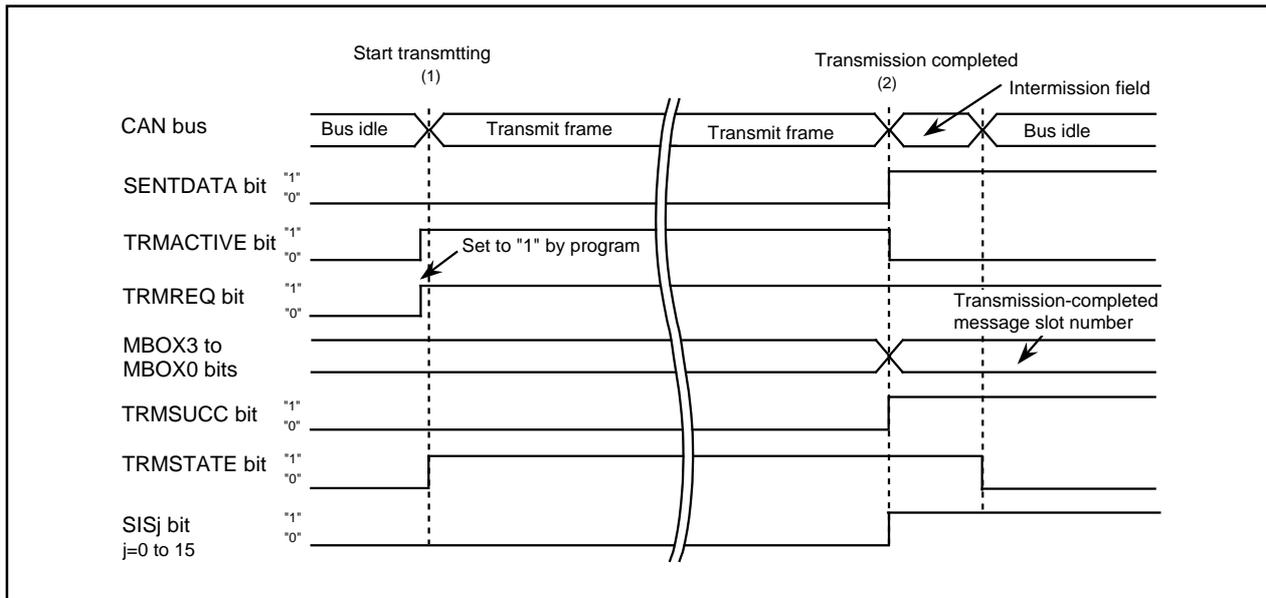


Figure 22.38 Example of CAN Data Frame Transmit Operation

### 22.3.3 CAN Receive Timing

Figure 22.39 shows an operation example of when the CAN receives a frame.

- (1) When the RECREQ bit in the CiMCTLj register (i=0 to 2, j= 0 to 15) is set to "1" (receive requested), the CAN is ready to receive the frame at anytime.
- (2) When the CAN starts receiving the frame, the RECSTATE bit in the CiSTR register is set to "1" (during reception).
- (3) After the CAN frame reception is completed, the INVALIDDATA bit in the CiMCTLj register is set to "1" (storing received data), the NEWDATA bit in the CiMCTLj register is set to "1" (receive complete) and the RECSUCC bit in the CiSTR register is set to "1" (reception completed).
- (4) After data is written to the message slot, the INVALIDDATA bit is set to "0" (storing receiving data) and the SISj bit in the CiSISTR register is set to "1" (interrupt requested). The MBOX3 to MBOX0 bits in the CiSTR register store received message slot numbers.

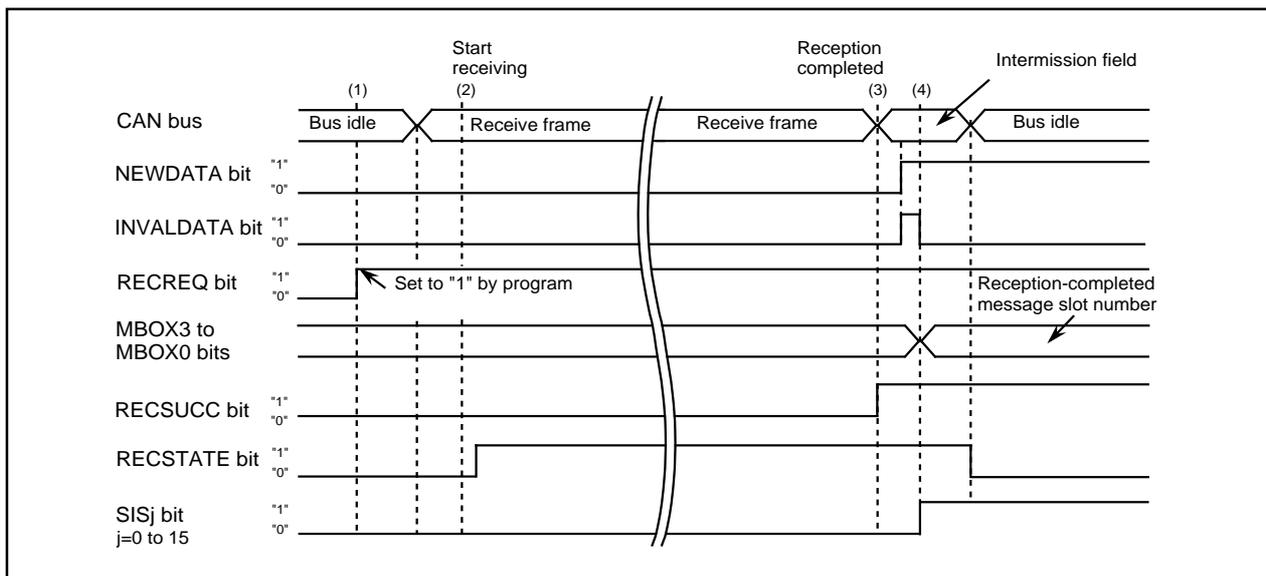


Figure 22.39 Example of CAN Data Frame Receive Operation

### 22.3.4 CAN Bus Error Timing

Figure 22.40 shows an operation example of when a CAN bus error occurs.

- (1) When a CAN bus error is detected, the STATE\_BUSERROR bit in the CiSTR register is set to "1", (error occurred) and the BEIS bit in the CiEISTR register is set to "1" (interrupt requested). The CAN starts transmitting the error frame.

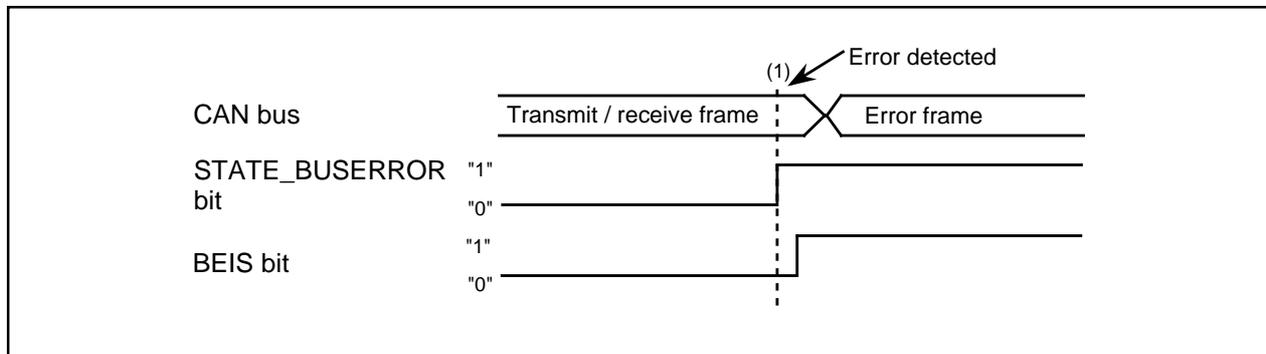


Figure 22.40 Operation Timing when CAN Bus Error Occurs

## 22.4 CAN Interrupts

The CAN<sub>i</sub> wake-up interrupt and CAN<sub>ij</sub> interrupts (i=0 to 2, j=0 to 2) are provided as the CAN interrupt.

### 22.4.1 CAN<sub>i</sub> Wake-Up Interrupt

#### 22.4.1.1 CAN<sub>0</sub> Wake-Up Interrupt

If P7<sub>7</sub> (CAN<sub>0IN</sub>/CAN<sub>02IN</sub>) is used as a CAN input port, the CAN<sub>0</sub> wake-up interrupt is available by using event counter mode of the timer A3 (TA<sub>3IN</sub>) that shares a pin with CAN<sub>0</sub>.

If P8<sub>3</sub> (CAN<sub>0IN</sub>/CAN<sub>1IN</sub>) is used as a CAN input port, the CAN<sub>0</sub> and CAN<sub>1</sub> wake-up interrupts are available by using  $\overline{\text{INT1}}$  that shares a pin with CAN<sub>0IN</sub>/CAN<sub>1IN</sub>.

#### 22.4.1.2 CAN<sub>1</sub> Wake-Up Interrupt

When a signal applied to the CAN<sub>1WU</sub> pin is on the falling edge, the CAN<sub>1WUR</sub> bit in the IIO5IR register is set to "1" (interrupt requested). At this time, the IR bit in the CAN<sub>5IC</sub> register is set to "1" (interrupt requested) if the CAN<sub>1WUE</sub> bit in the IIO5IE register is set to "1" (interrupt enabled).

If P8<sub>3</sub> (CAN<sub>0IN</sub>/CAN<sub>1IN</sub>) is used as a CAN input port, the CAN<sub>0</sub> and CAN<sub>1</sub> wake-up interrupts are available by using  $\overline{\text{INT1}}$  that shares a pin with CAN<sub>0IN</sub>/CAN<sub>1IN</sub>.

#### 22.4.1.3 CAN<sub>2</sub> Wake-Up Interrupt

When a signal applied to the CAN<sub>2WU</sub> pin is on the falling edge, the CAN<sub>2WUR</sub> bit in the IIO6IR register is set to "1" (interrupt requested). At this time, the IR bit in the CAN<sub>8IC</sub> register is set to "1" (interrupt requested) if the CAN<sub>2WUE</sub> bit in the IIO6IE register is set to "1" (interrupt enabled).

## 22.4.2 CANij Interrupts

Figure 22.41 shows a block diagram of the CANij interrupts. The followings cause the CAN-associated interrupt request to be generated.

- The CANi slot k (k=0 to 15) completes a transmission
- The CANi slot k completes a reception
- The CANi module detects a bus error
- The CANi module moves into an error-passive state
- The CANi module moves into a bus-off state

The INTSEL bit in the CiCTRL1 register determines how an interrupt request is generated. When the INTSEL bit is set to "0", one of the above CANi interrupt request sources cause the CANij interrupts to be generated by the OR circuit. When the INTSEL bit is set to "1", CANi transmission completed, CANi reception completed and CANi errors (CANi bus error detection, CANi module into error-passive state and CANi module into bus-off state) cause the CANij interrupt corresponding to each source to be generated.

### 22.4.2.1 When the INTSEL Bit is Set to "0"

If the CAN-associated interrupt is generated by one of the interrupt request sources listed in **22.4.2 CANij Interrupts**, the corresponding bit in the CiSISTR register (i=0 to 2) is set to "1" (interrupt requested) when the CANi slot k completes a transmission or a reception. The corresponding bit in the CiEISTR register (i=0 to 2) is set to "1" (interrupt requested) when the CANi module detects a bus error, moves into an error-passive state, or moves into a bus-off state.

The CANi interrupt request signal is set to "1" when the corresponding bit in the CiSISTR or CiEISTR is set to "1" and the corresponding bit in the CiSIMKR or CiEIMKR is set to "1"

When the CAN0 interrupt request signal changes "0" to "1", all CAN0jR bits (j=0 to 2) in the IIO9IR to IIO11IR registers are set to "1" (interrupt requested).

If at least one of the CAN0jE bits in the IIO9IE to IIO11IE registers is set to "1" (interrupt enabled), the IR bits in the corresponding CAN0IC to CAN2IC registers are set to "1" (interrupt requested). The CAN0 interrupt request signal remains set to "1" if another interrupt request causes a corresponding bit in the C0SISTR or C0EISTR to be set to "1" and the corresponding bit in the C0SIMKR or C0EIMKR to be set to "1" after the CAN0 interrupt request signal changes "0" to "1". The CAN0jR and IR bits also remain unchanged.

When the CAN1 interrupt request signal changes "0" to "1", all three CAN1jR bits in the IIO0IR to IIO5IR registers are set to "1" (interrupt requested).

If at least one of the CAN1jE bits in the IIO0IE to IIO5IE and IIO5IE registers is set to "1", the IR bits in the corresponding CAN3IC to CAN5IC registers are set to "1". The CAN1 interrupt request signal remains set to "1" if another interrupt request causes the corresponding bit in the C1SISTR or C1EISTR to be set to "1" and the corresponding bit in the C1SIMKR or C1EIMKR to be set to "1" after the CAN1 interrupt request signal changes "0" to "1". The CAN1jR and IR bits also remain unchanged.

When the CAN2 interrupt request signal changes "0" to "1", all three CAN2jR bits in the IIO2IR to IIO3IR and IIO6IR registers are set to "1" (interrupt requested).

If at least one of the CAN2jE bits in the IIO2IE to IIO3IE and IIO6IE registers is set to "1", the IR bits in the corresponding CAN6IC to CAN8IC registers are set to "1". The CAN2 interrupt request signal remains set to "1" if another interrupt request causes the corresponding bit in the C2SISTR or C2EISTR to be set to "1" and the corresponding bit in the C2SIMKR or C2EIMKR to be set to "1" after the CAN2 interrupt request signal changes "0" to "1". The CAN2jR and IR bits also remain unchanged.

Bits in the CiSISTR or CiEISTR register and CANijR bits (i=0 to 2, j=0 to 2) in the IIO0IR to IIO11R, IIO5IR, IIO9IR to IIO11IR, IIO2IR to IIO3IR and IIO6IR registers are not set to "0" automatically, interrupt acknowledgment notwithstanding. Set these bits to "0" by program.

The CANi interrupts are acknowledged when the CANijR bit in the IIO0IR to IIO11R, IIO5IR, IIO9IR to IIO11IR, IIO2IR to IIO3IR or IIO6IR register and the corresponding bit in the CiSISTR or CiEISTR register are set to "0". If these bits remain set to "1", all CAN-associated interrupt request sources become invalid.

#### 22.4.2.2 When the INTSEL Bit is Set to "1"

If the CAN-associated interrupt is generated by one of the interrupt request sources listed in **22.4.2 CANij Interrupts**, the corresponding bit in the CiSISTR register (i=0 to 2) is set to "1" (interrupt requested) when the CANi slot k(k=0 to 15) completes a transmission or a reception. The corresponding bit in the CiEISTR register is set to "1" (interrupt requested) when the CANi module detects a bus error, moves into an error-passive state, or moves into a bus-off state.

The CANi receive interrupt request signal is set to "1" if the corresponding bit in the CiSIMKR is set to "1" and the corresponding bit in the CiSISTR register is set to "1" when the CANi module completes a reception.

The CANi transmit interrupt request signal is set to "1" if the corresponding bit in the CiSIMKR is set to "1" and the corresponding bit in the CiSISTR register is set to "1" when the CANi module completes a transmission.

The CANi error interrupt request signal is set to "1" if corresponding bits in the CiEIMKR are set to "1" and the corresponding bit in the CiEISTR register is set to "1" when the CANi module detects a bus error, moves into an error-passive state, or moves into a bus-off state.

When the CANi receive interrupt request signal changes "0" to "1", the CAN00R bit in the IIO9IR register, the CAN10R bit in the IIO0IR register and the CAN20R bit in the IIO2IR register are set to "1" (interrupt requested). If the CAN00E in the IIO9IE register is set to "1" (interrupt enabled), the IR bit in the CAN0IC register is set to "1" (interrupt requested). If the CAN10E bit in the IIO0IE register is set to "1" (interrupt enabled), the IR bit in the CAN3IC register is set to "1" (interrupt requested). If the CAN20E bit in the IIO2IE register is set to "1" (interrupt enabled), the IR bit in the CAN6IC register is set to "1" (interrupt requested).

When the CANi transmit interrupt request signal changes "0" to "1", the CAN01R bit in the IIO10IR register, the CAN11R bit in the IIO1IR register and the CAN21R bit in the IIO3IR register are set to "1" (interrupt requested). If the CAN01E in the IIO10IE register is set to "1" (interrupt enabled), the IR bit in the CAN1IC register is set to "1" (interrupt requested). If the CAN11E bit in the IIO1IE register is set to "1" (interrupt enabled), the IR bit in the CAN4IC register is set to "1" (interrupt requested). If the CAN21E bit in the IIO3IE register is set to "1" (interrupt enabled), the IR bit in the CAN7IC register is set to "1" (interrupt requested).

When the CANi error interrupt request signal changes "0" to "1", the CAN02R bit in the IIO11IR register, the CAN12R bit in the IIO5IR register and the CAN22R bit in the IIO6IR register are set to "1" (interrupt requested). If the CAN02E in the IIO11IE register is set to "1" (interrupt enabled), the IR bit in the CAN2IC register is set to "1" (interrupt requested). If the CAN12E bit in the IIO5IE register is set to "1" (interrupt enabled), the IR bit in the CAN5IC register is set to "1" (interrupt requested). If the CAN22E bit in the IIO6IE register is set to "1" (interrupt enabled), the IR bit in the CAN8IC register is set to "1" (interrupt requested).

The CANi error interrupt request signal remains set to "1" if another interrupt request causes the corresponding bit in the CiEIMKR register is set to "1" and the corresponding bit in the CiEISTR to be set to "1" after the CANi interrupt request signal changes "0" to "1". The CAN02R, CAN12R, CAN22R and IR bits also remain unchanged.

Bits in the CiSISTR or CiEISTR register and CANijR bits (i=0 to 2, j=0 to 2) in the IIO0IR to IIO1IR, IIO5IR, IIO9IR to IIO11IR, IIO2IR to IIO3IR or IIO6IR registers are not set to "0" automatically, interrupt acknowledgment notwithstanding. Set these bits to "0" by program.

The CANi receive interrupt and CANi transmit interrupt are acknowledged when the CAN00R bit in the IIO9IR register, the CAN01R bit in the IIO10IR register, the CAN10R bit in the IIO0IR register, the CAN11R bit in the IIO1IR register, the CAN20R bit in the IIO2IR register and the CAN21R bit in the IIO3IR register are set to "0". Corresponding bits in the CiSISTR register can be set to either "0" or "1". The CANi error interrupt is acknowledged when the CAN02R bit in the IIO11IR register, the CAN12R bit in the IIO5IR register, the CAN22R bit in the IIO6IR register and corresponding bits in the CiEISTR register are set to "0".

If these bits remain set to "1", all CAN- associated interrupt request sources become invalid.

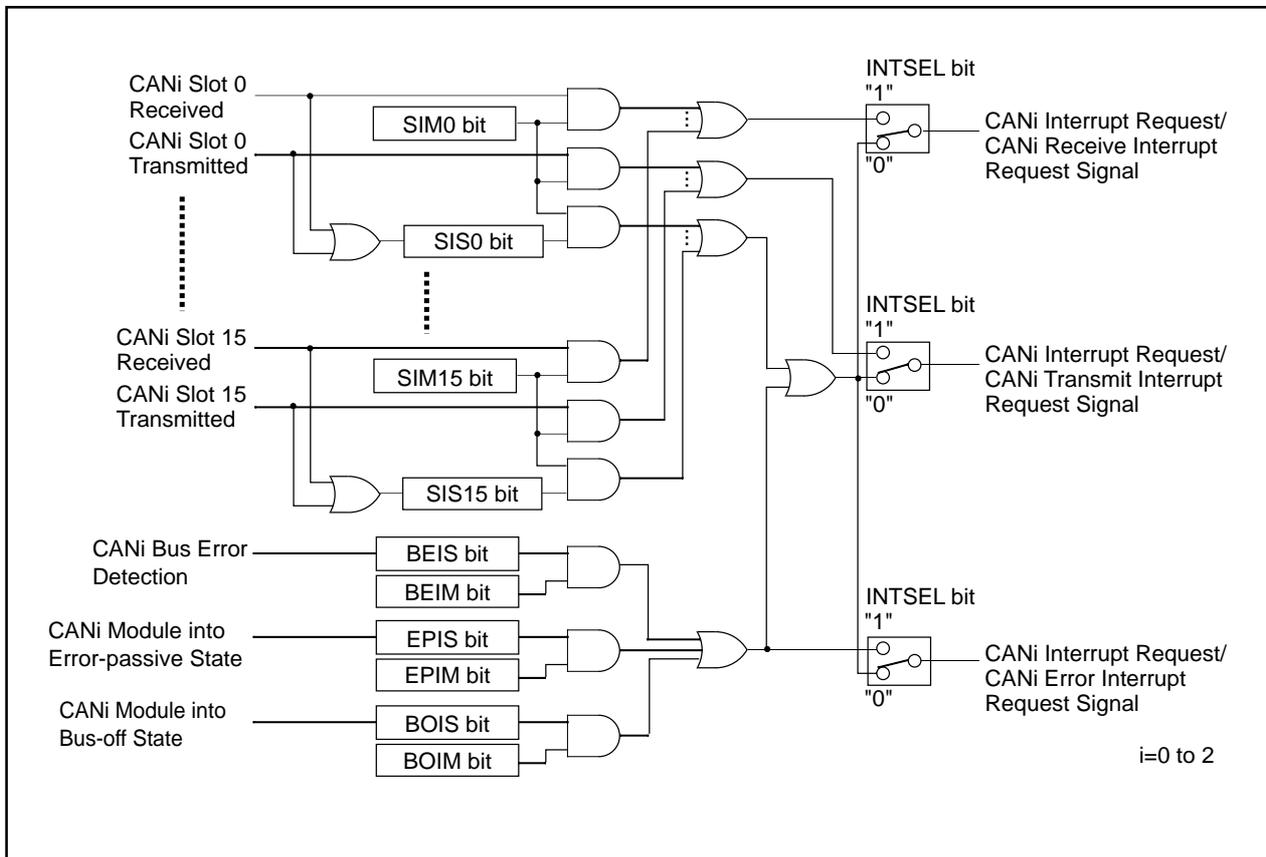


Figure 22.41 CAN Interrupts

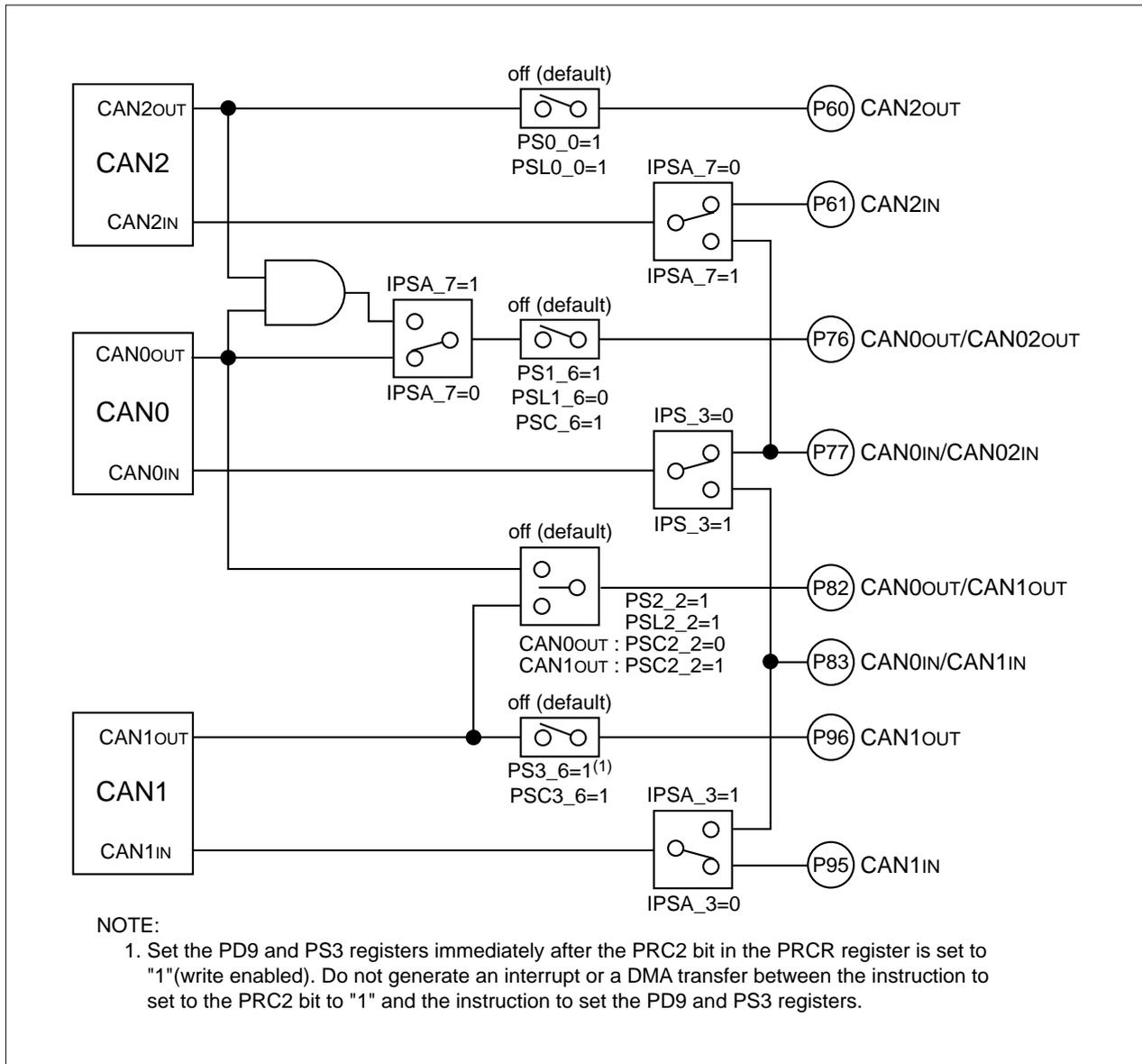
## 22.5 CAN0/CAN2 Combination Mode

In CAN0/CAN2 combination mode, CAN0 and CAN2 are combined for input/ output.

Signals output from CAN0 and CAN2 are combined and provided from the CAN02OUT pin. Signal applied to the CAN02IN pin is applied to CAN0 and CAN2.

When using CAN0/CAN2 combination mode, refer to **Table 22.2 CAN Pin Settings** for pin settings.

Figure 22.42 shows a block diagram of CAN0/CAN2 combination mode.



**Figure 22.42 CAN0/CAN2 Combination Mode Block Diagram**

## 22.5.1 Notes for CAN0/CAN2 Combination Mode

It is important to note the following information to use CAN0/CAN2 combination mode:

### 22.5.1.1 Transmission

Do not transmit frames with the same ID from CAN0 and CAN2.

When transmitting frames with different IDs simultaneously from CAN0 and CAN2, they are arbitrated.

As a result, the frame with higher priority gains the right to transmit.

### 22.5.1.2 Reception

When CAN0 transmits, ACK for the transmitted frame is transmitted from CAN2 even if other nodes are not connected to the CAN bus. The same applies to CAN2 transmission.

### 22.5.1.3 Interrupts

CAN0 and CAN2 have different completed transmission or reception interrupts.

When CAN0 completes a transmission or reception, the IR bits in the corresponding CAN0IC to CAN2IC registers are set to "1" (interrupt requested). When CAN2 completes a transmission or reception, the IR bits in the corresponding CAN6IC to CAN8IC registers are set to "1" (interrupt requested).

Use the CAN0 wake-up interrupt when operating in CAN0/CAN2 combination mode.

### 22.5.1.4 Errors

The count value of the error counter and the error status may differ between CAN0 and CAN2 depending on where the error has been generated.

When using CAN0/CAN2 combination mode, an error frame longer than the length noted in the specification may be transmitted.

### 22.4.1.5 Configuration

Set CAN configuration to both CAN0 and CAN2. Set CAN baud rates and bit timings of both CAN 0 and CAN2 to the same values.

## 23. Programmable I/O Ports

87 programmable I/O ports from P0 to P10 (excluding P85) are available in the 100-pin package and 123 programmable I/O ports from P0 to P15 (excluding P85) are in the 144-pin package. The direction registers determine each port status, input or output. The pull-up control registers determine whether the ports, divided into groups of four ports, are pulled up or not. P85 is an input port and no pull-up for this port is allowed. The P8\_5 bit in the P8 register indicates an  $\overline{\text{NMI}}$  input level since P85 shares pins with  $\overline{\text{NMI}}$ .

Figures 23.1 to 23.4 show programmable I/O port configurations.

Each pin functions as the programmable I/O port or an I/O pin for internal peripheral functions.

To use pins as input or output pins for internal peripheral functions, refer to the explanations for each function.

The registers associated with the programmable I/O ports are as follows.

### 23.1 Port Pi Direction Register (PDi Register, i=0 to 15)

Figure 23.5 shows the PDi register.

The PDi register selects input or output status of a programmable I/O port. Each bit in the PDi register corresponds to a port.

No bit controlling P85 is provided in the direction registers.

### 23.2 Port Pi Register (Pi Register, i=0 to 15)

Figure 23.6 shows the Pi register.

The Pi register writes and reads data to communicate with external devices. The Pi register consists of a port latch to hold output data and a circuit to read pin states. Each bit in the Pi register corresponds to a port.

### 23.3 Function Select Register Aj (PSj Register) (j=0 to 3, 5, 8, 9)

Figures 23.7 to 23.10 show the PSj registers.

The PSj register selects either I/O port or peripheral function output if an I/O port shares pins with a peripheral function output (excluding DA0 and DA1.)

When multiple peripheral function outputs are assigned to a pin, set the PSL0 to PSL3, PSC, PSC2, PSC3 and PSD1 registers to select which function is used.

Tables 23.2 to 23.9 list peripheral function output control settings for each pin.

### 23.4 Function Select Register B0 to B3 (PSL0 to PSL3 Registers)

Figures 23.11 and 23.12 show the PSL0 to PSL3 registers.

When multiple peripheral function outputs are assigned to a pin, the PSL0 to PSL3 registers select which peripheral function output is used.

Refer to **23.10 Analog Input and Other Peripheral Function Input** for the PSL3\_6 to PSL3\_3 bits in the PSL3 register.

### 23.5 Function Select Register C, C2, C3 (PSC, PSC2, PSC3 Registers)

Figures 23.13 and 23.14 show the PSC, PSC2 and PSC3 registers.

When multiple peripheral function outputs are assigned to a pin, the PSC, PSC2 and PSC3 registers select which peripheral function output is used.

Refer to **23.10 Analog Input and Other Peripheral Function Input** for the PSC\_7 bit in the PSC register.

### 23.6 Function Select Register D (PSD1 Register)

Figure 23.14 shows the PSD1 register.

When multiple peripheral function outputs are assigned to a pin, the PSD1 register selects which peripheral function output is used.

### 23.7 Pull-up Control Register 0 to 4 (PUR0 to PUR4 Registers)

Figures 23.15 and 23.16 show the PUR0 to PUR4 registers.

The PUR0 to PUR4 registers select whether the ports, divided into groups of four ports, are pulled up or not. Ports with bits in the PUR0 to PUR4 registers set to "1" (pull-up) and the direction registers set to "0" (input mode) are pulled up.

### 23.8 Port Control Register (PCR Register)

Figure 23.17 shows the PCR register.

The PCR register selects either CMOS output or N-channel open drain output as the P1 output format. If the PCR0 bit is set to "1", N-channel open drain output is selected because the P-channel in the CMOS port is turned off. This is, however, not a perfect open drain. Therefore, the absolute maximum rating of the input voltage is between  $-0.3V$  and  $V_{CC} + 0.3V$ .

### 23.9 Input Function Select Register (IPS and IPSA Registers)

Figures 23.17 and 23.18 show the IPS and IPSA registers.

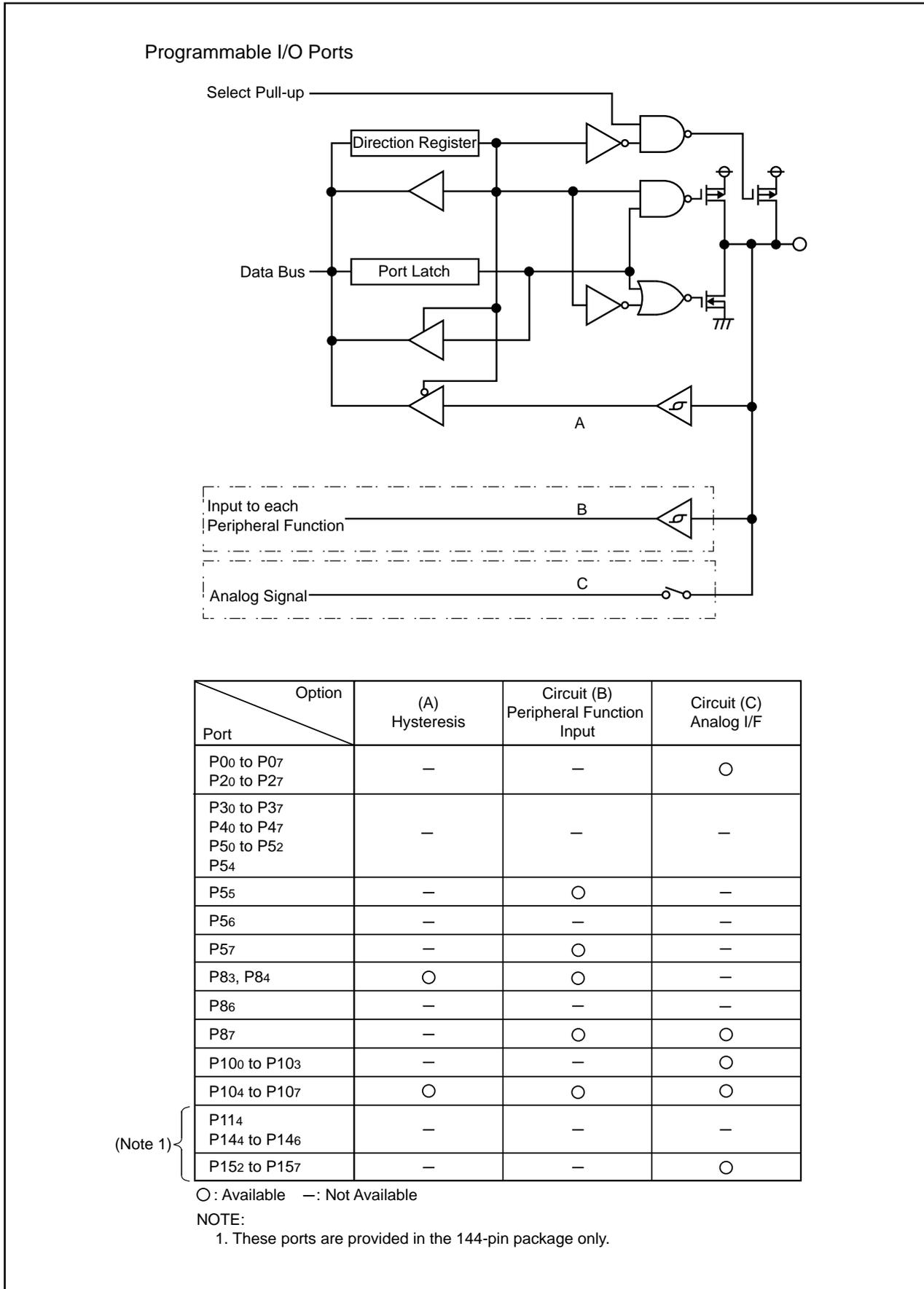
The IPS3, IPS1 and IPS0 bits in the IPS register and the IPSA\_3 and IPSA\_0 bits in the IPSA register select which pin is assigned for the intelligent I/O or CAN input functions.

Refer to **23.10 Analog Input and Other Peripheral Function Input** for the IPS2 bit.

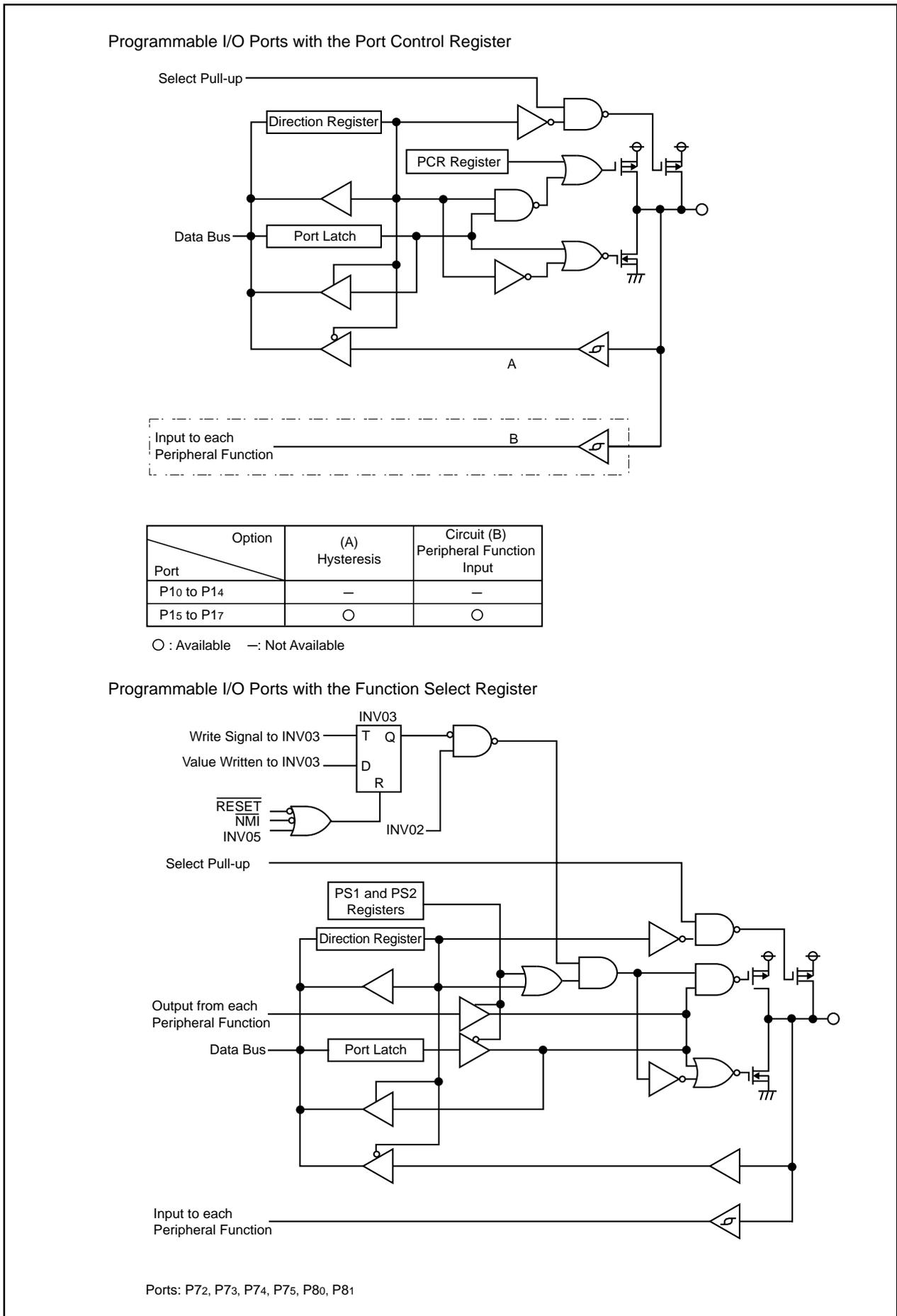
### 23.10 Analog Input and Other Peripheral Function Input

The PSL3\_6 to PSL3\_3 bits in the PSL3 register, the PSC\_7 bit in the PSC register and the IPS2 bit in the IPS register each separate analog I/O ports from other peripheral functions. Setting the corresponding bit to "1" (analog I/O) to use the analog I/O port (DA0, DA1, ANEX0, ANEX1, AN4 to AN7 or AN150 to AN157) prevents an intermediate potential from being impressed to other peripheral functions. The impressed intermediate potential may cause increase in power consumption.

Set the corresponding bit to "0" (except analog I/O) when analog I/O is not used. All peripheral function inputs except the analog I/O port are available when the corresponding bit is set to "0". These inputs are indeterminate when the bit is set to "1". When the PSC\_7 bit is set to "1", key input interrupt request remains unchanged regardless of  $\overline{KI0}$  to  $\overline{KI3}$  pin input level change.



**Figure 23.1 Programmable I/O Ports (1)**



**Figure 23.2 Programmable I/O Ports (2)**

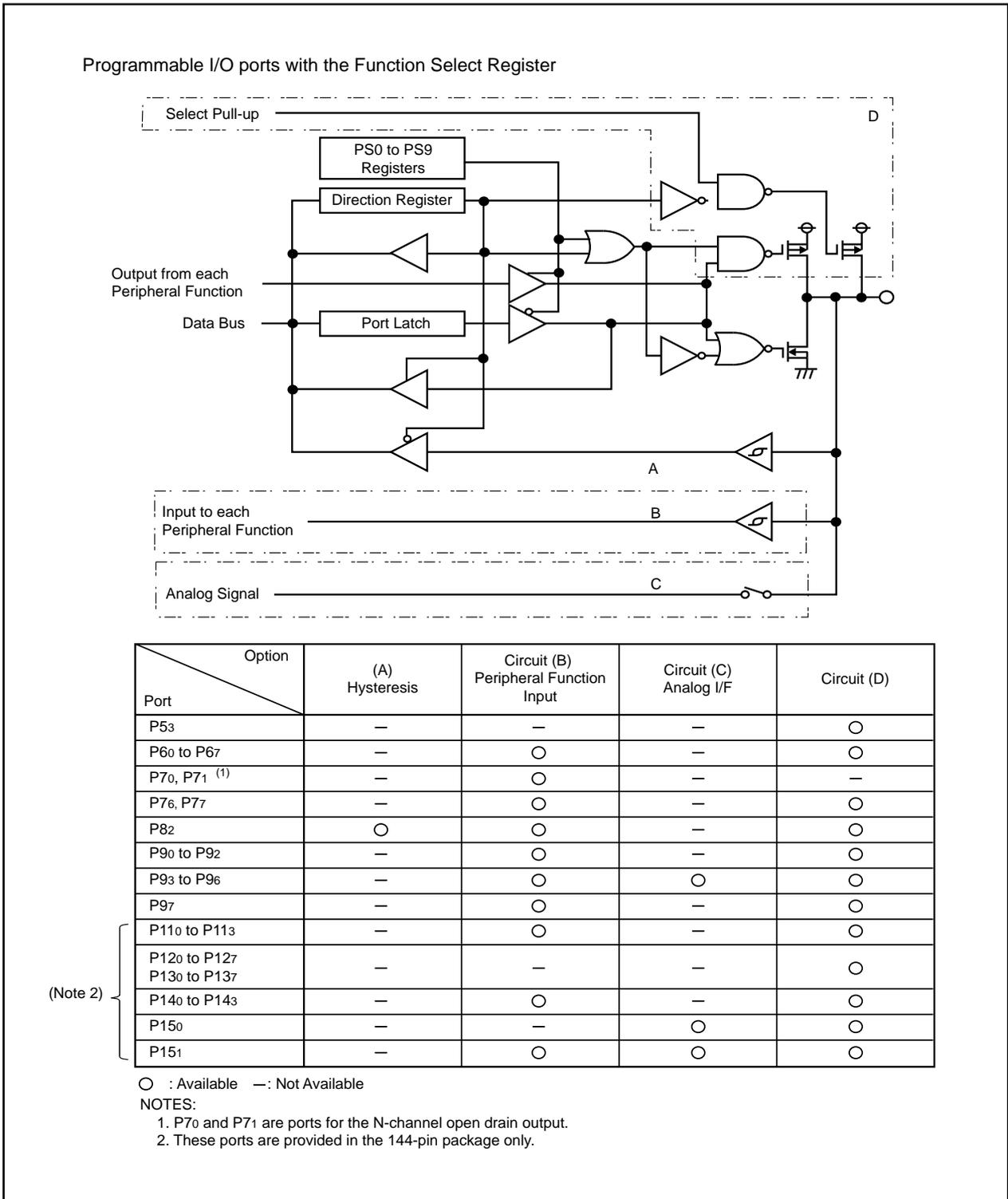


Figure 23.3 Programmable I/O Ports (3)

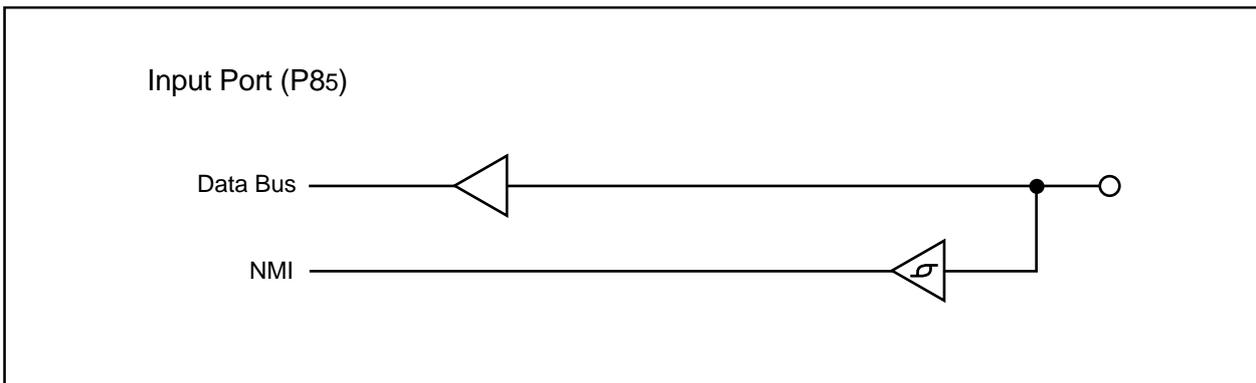


Figure 23.4 Programmable I/O Ports (4)

### Port Pi Direction Register (i=0 to 15) <sup>(2)</sup>

Symbol	Address	After Reset
PD0 to PD3	03E2 <sub>16</sub> , 03E3 <sub>16</sub> , 03E6 <sub>16</sub> , 03E7 <sub>16</sub>	00 <sub>16</sub>
PD4 to PD7	03EA <sub>16</sub> , 03EB <sub>16</sub> , 03C2 <sub>16</sub> , 03C3 <sub>16</sub>	00 <sub>16</sub>
PD8	03C6 <sub>16</sub> <sup>(3)</sup>	00X0 0000 <sub>2</sub>
PD9 to PD10	03C7 <sub>16</sub> <sup>(1)</sup> , 03CA <sub>16</sub>	00 <sub>16</sub>
PD11	03CB <sub>16</sub> <sup>(2, 3)</sup>	XXX0 0000 <sub>2</sub>
PD12 to PD13	03CE, 03CF <sub>16</sub> <sup>(2)</sup>	00 <sub>16</sub>
PD14	03D2 <sub>16</sub> <sup>(2, 3)</sup>	X000 0000 <sub>2</sub>
PD15	03D3 <sub>16</sub> <sup>(2)</sup>	00 <sub>16</sub>

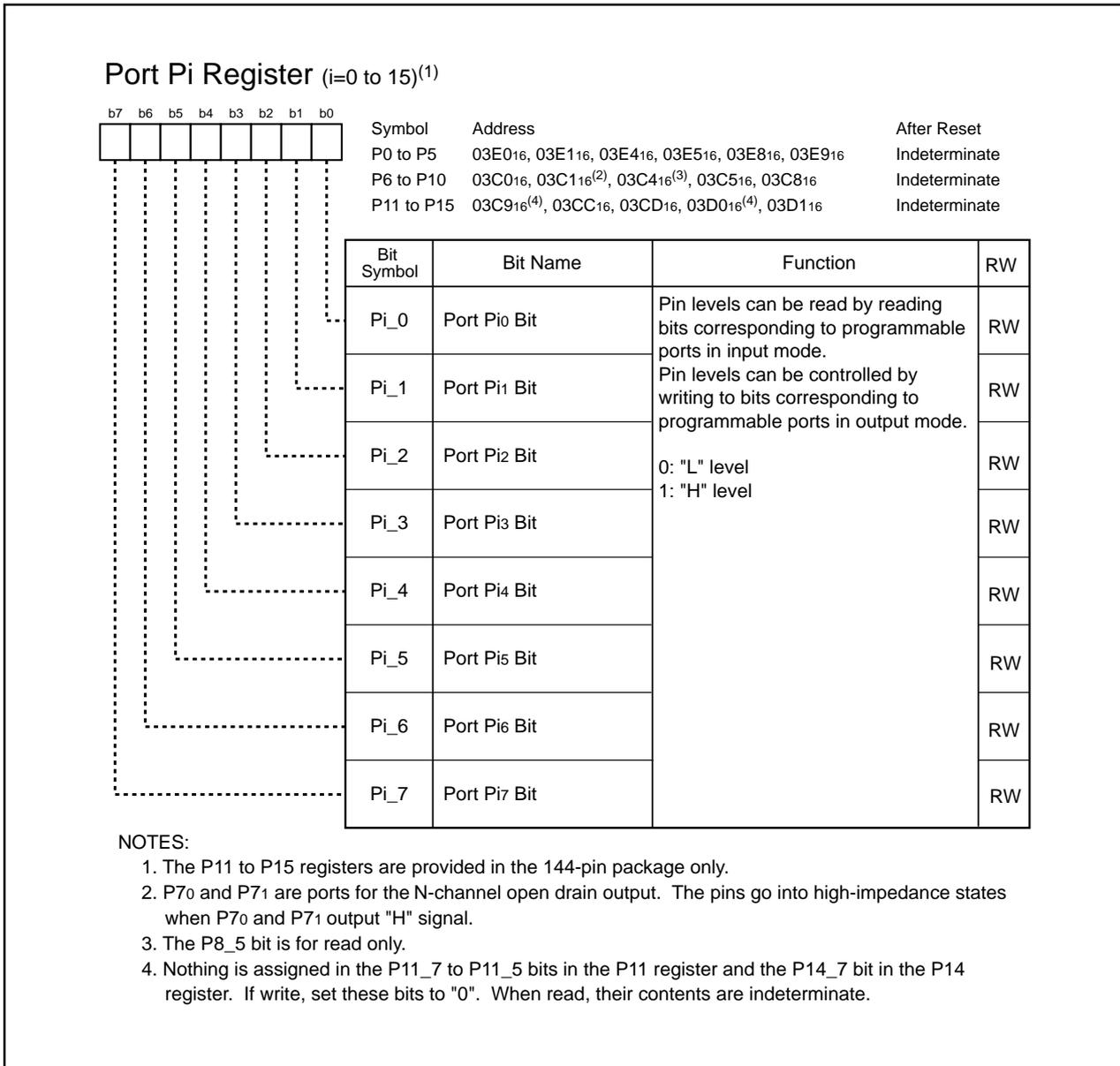
Bit Symbol	Bit Name	Function	RW
PDi_0	Port Pi0 Direction Bit	0: Input mode (Functions as input port) 1: Output mode (Functions as output port)	RW
PDi_1	Port Pi1 Direction Bit	0: Input mode (Functions as input port) 1: Output mode (Functions as output port)	RW
PDi_2	Port Pi2 Direction Bit	0: Input mode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

**NOTES:**

- Set the PD9 register immediately after the PRC2 bit in the PRCR register is set to "1" (write enabled). Do not generate an interrupt or a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to set the PD9 register.
- Set the PD11 to PD15 registers to "FF<sub>16</sub>" in the 100-pin package.
- Nothing is assigned in the PD8\_5 bit in the PD8 register, the PD11\_7 to PD11\_5 bits in the PD11 register (144-pin package only) and the PD14\_7 bit in the PD14 register (144-pin package only). If write, set these bits to "0". When read, their contents are indeterminate.

Figure 23.5 PD0 to PD15 Registers



**Figure 23.6 P0 to P15 Registers**

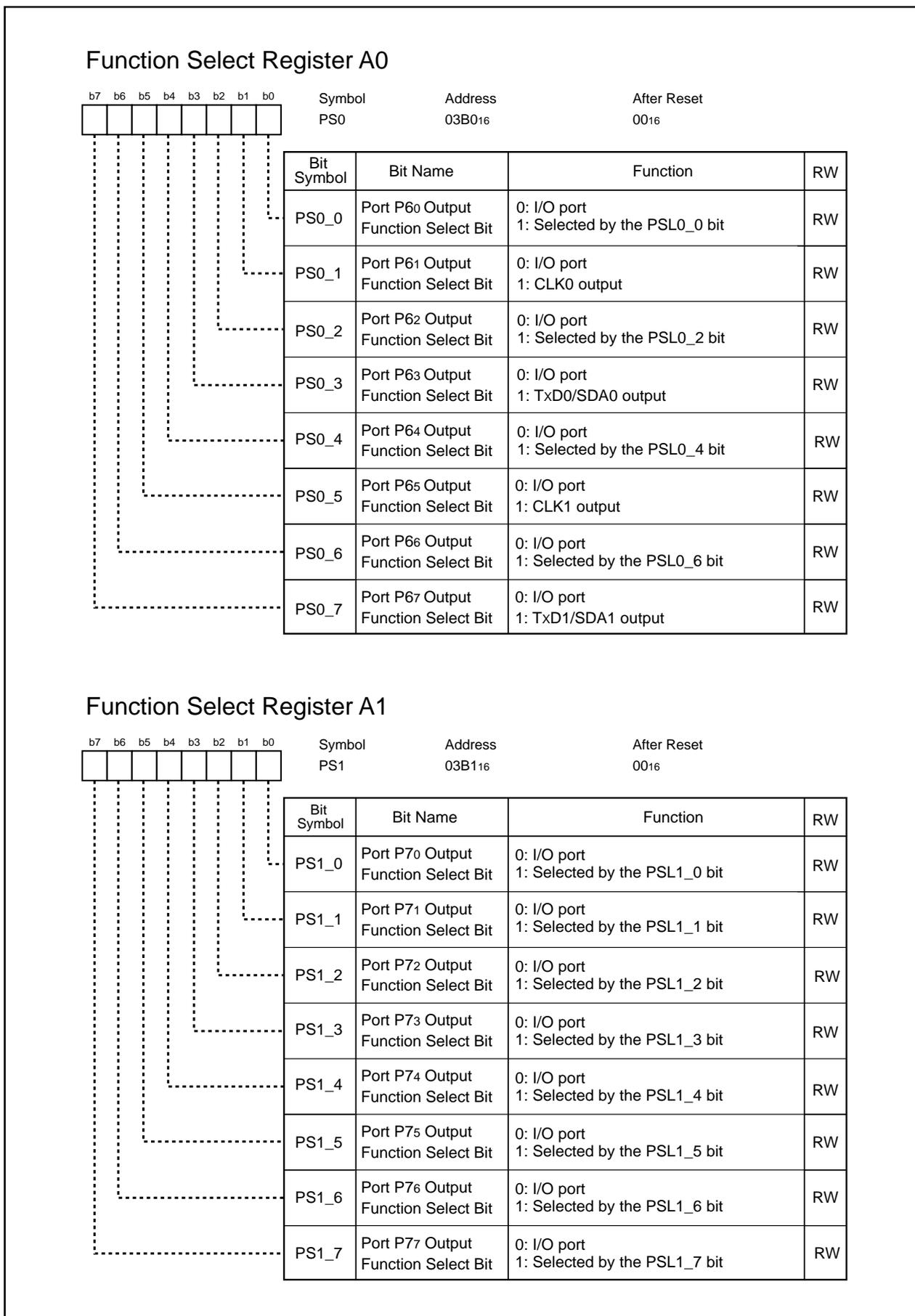
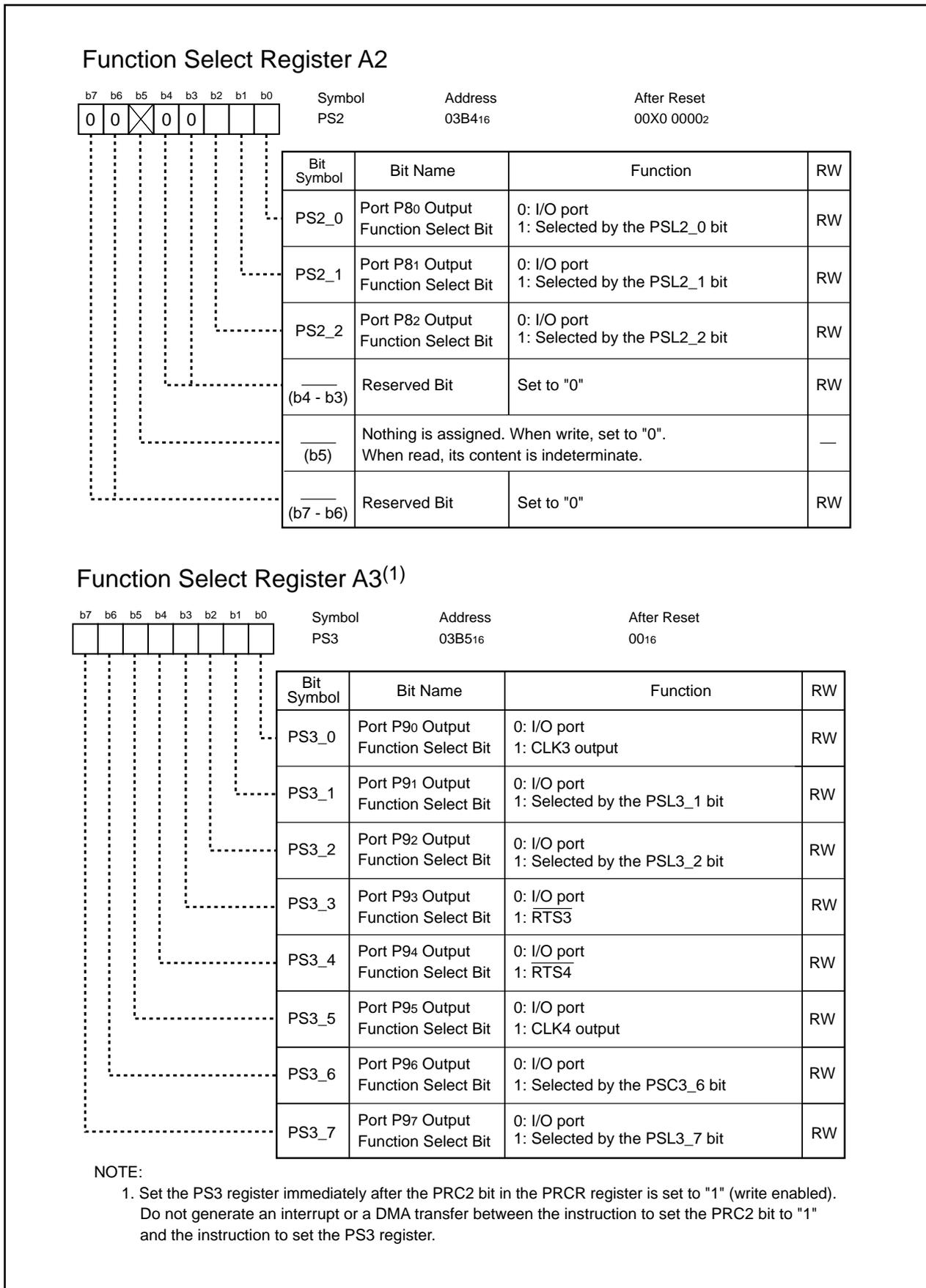


Figure 23.7 PS0 Register and PS1 Register



**Figure 23.8 PS2 Register and PS3 Register**

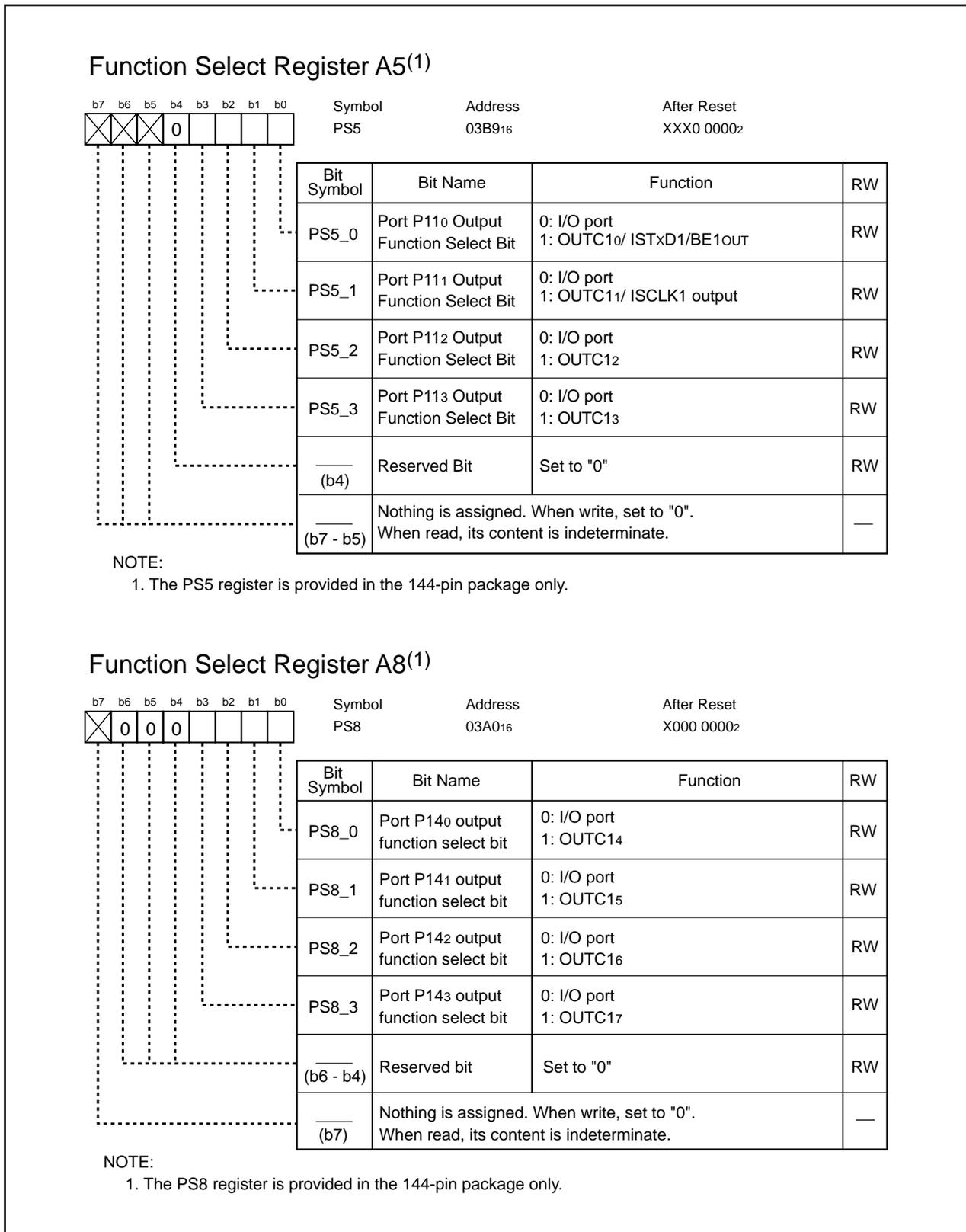
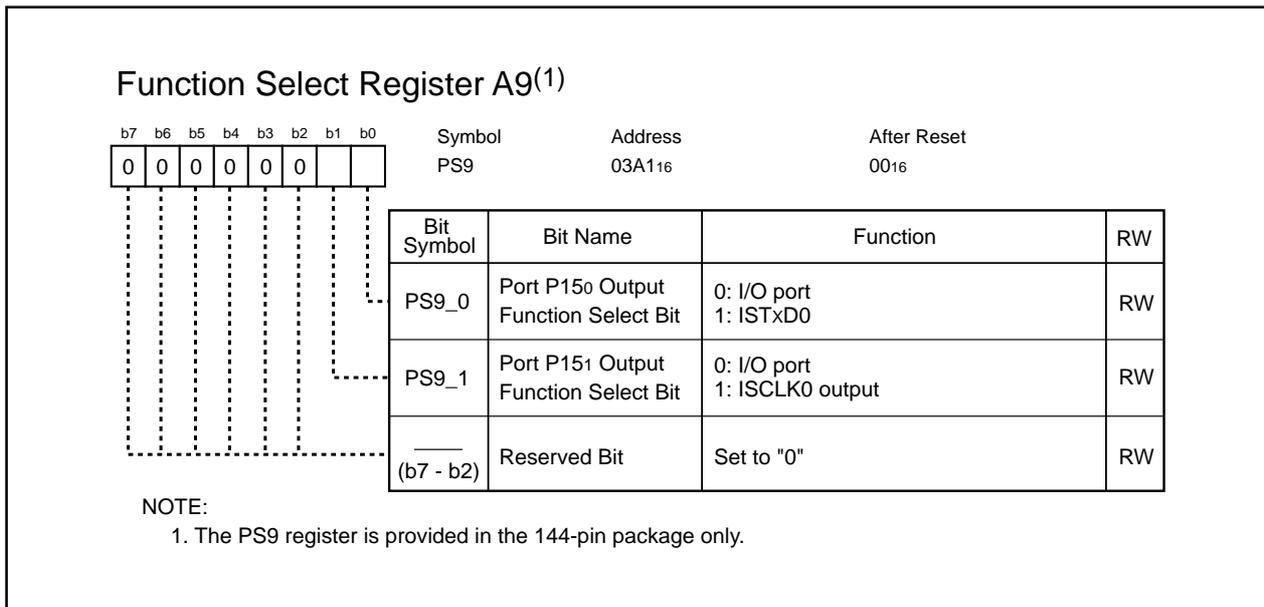
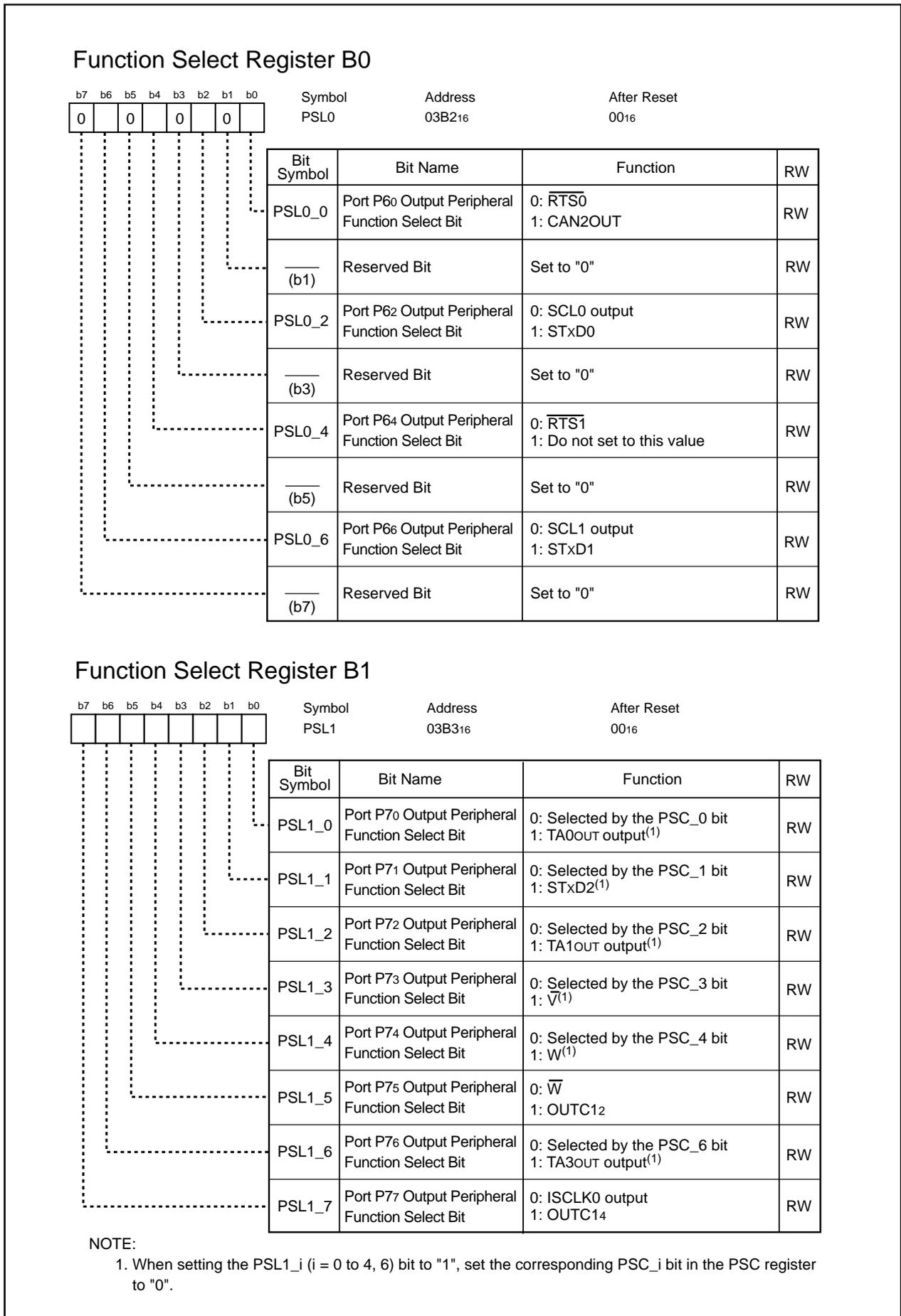
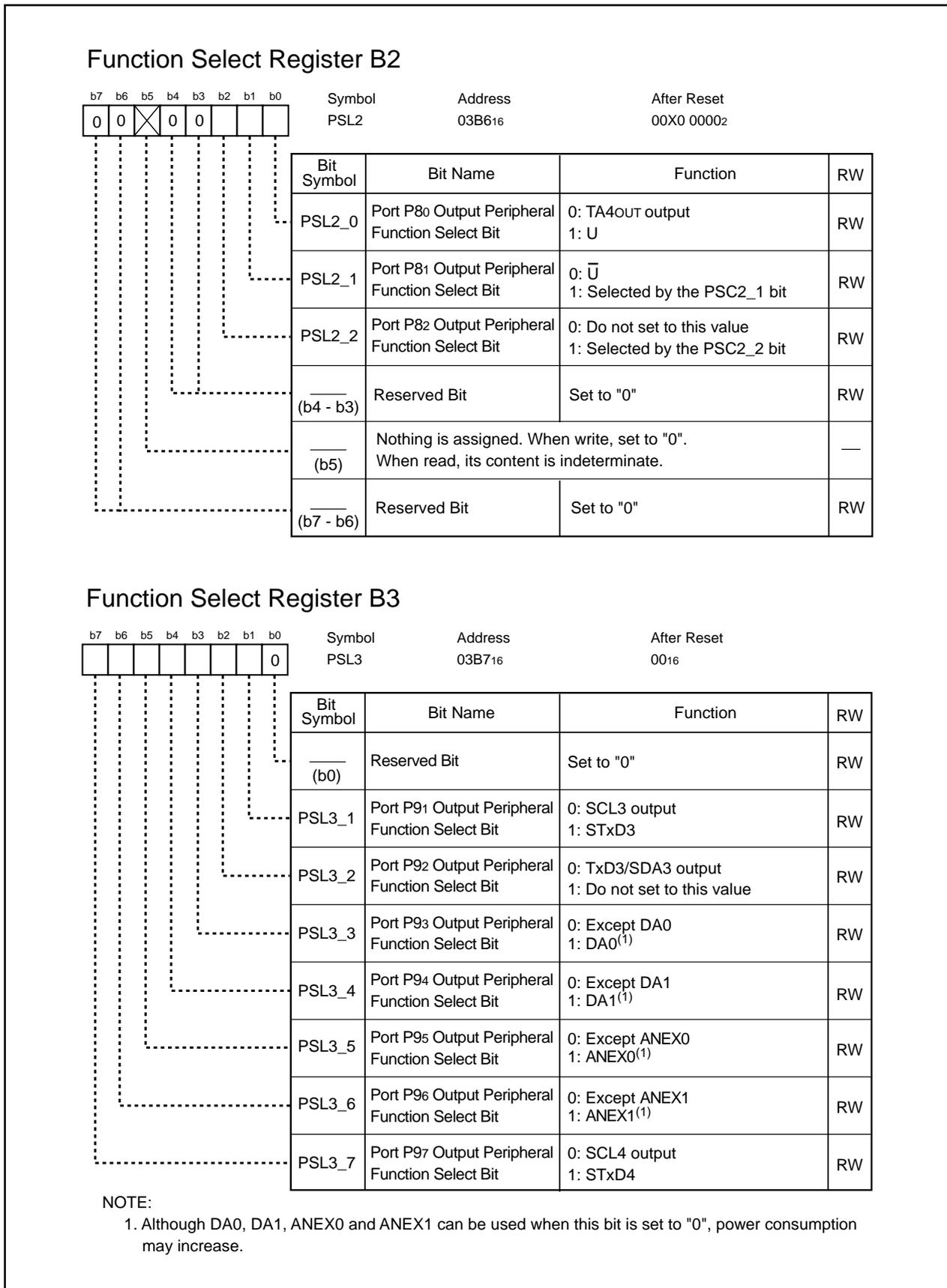


Figure 23.9 PS5 Register and PS8 Register

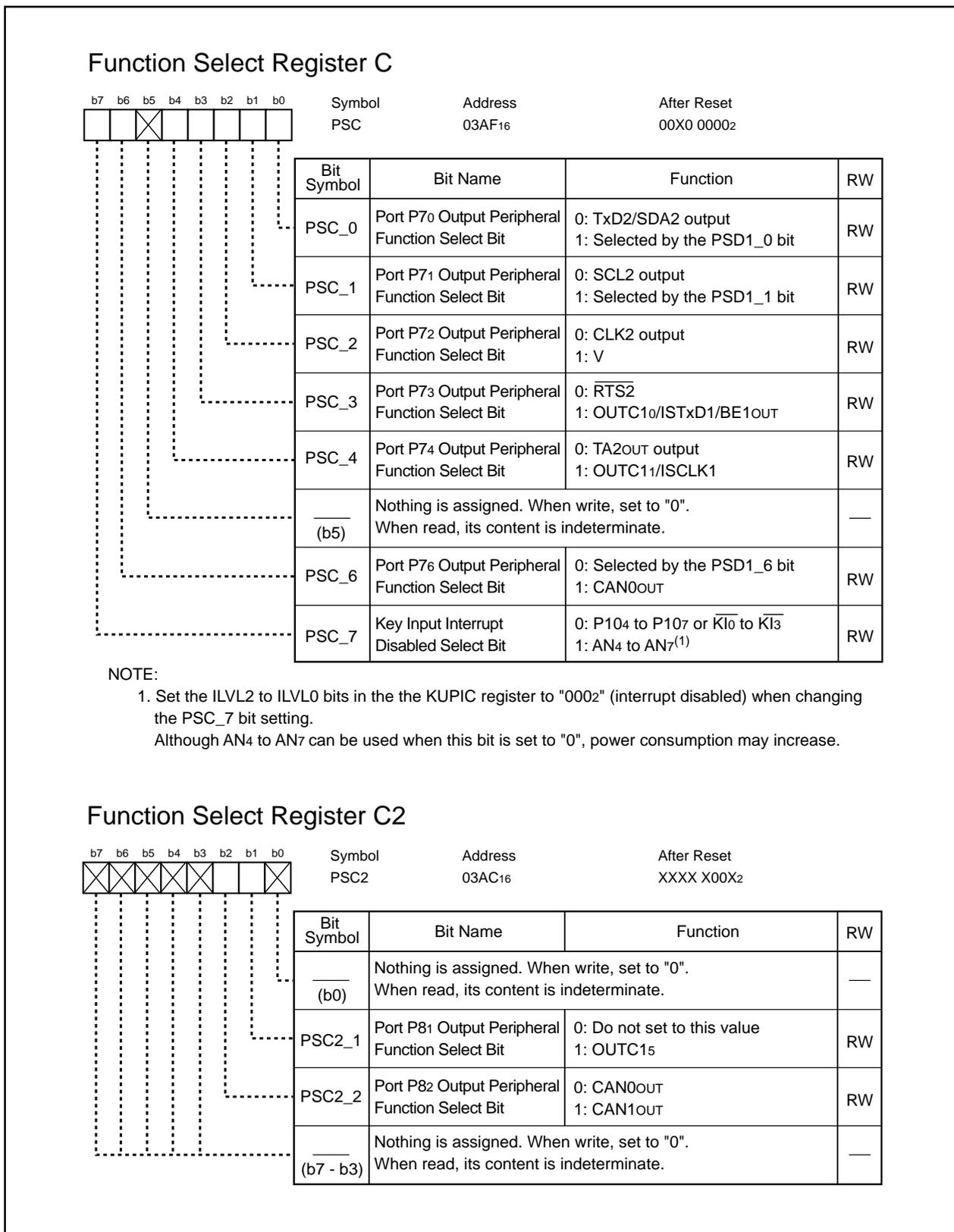
**Figure 23.10 PS9 Register**



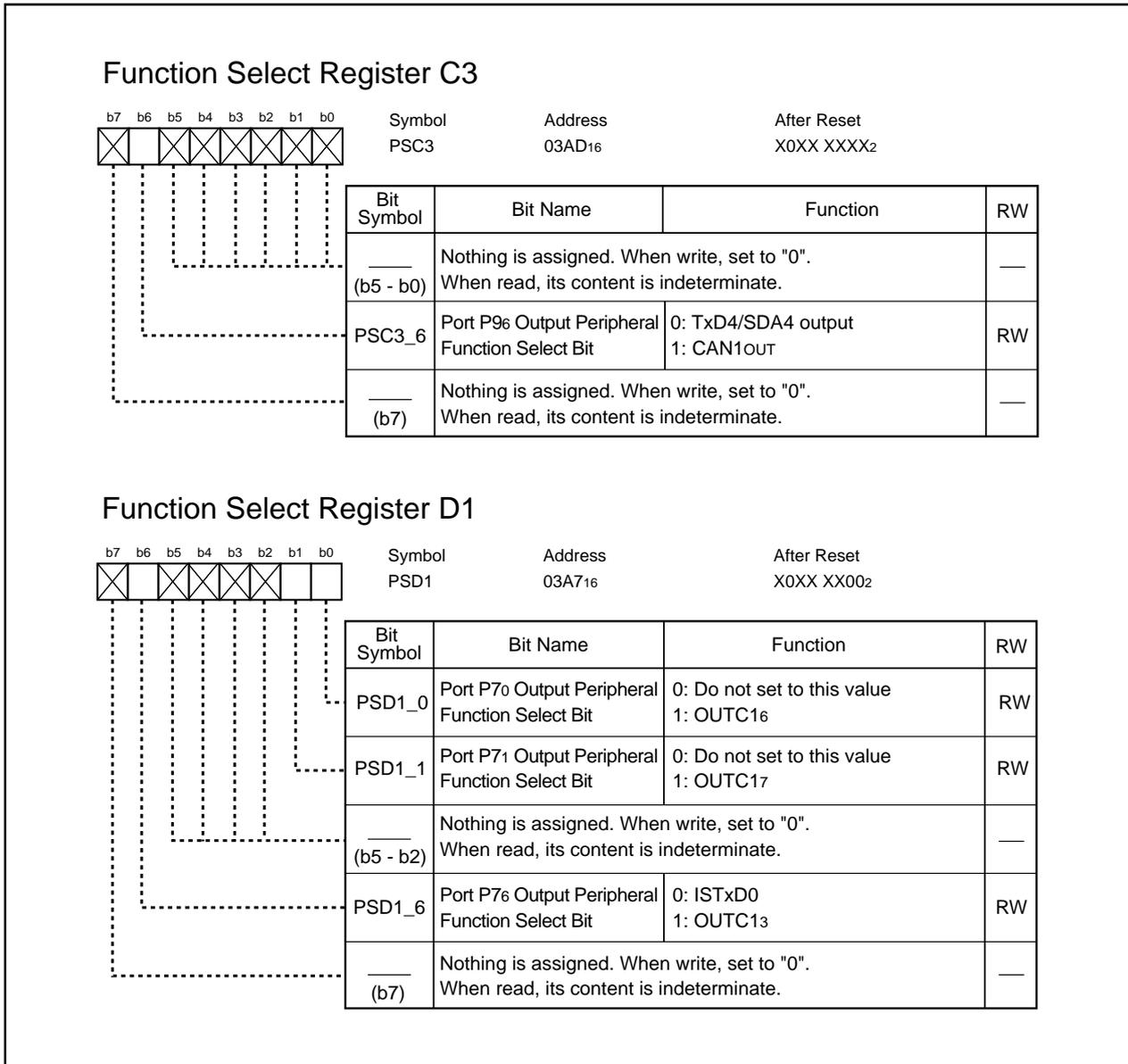
**Figure 23.11 PSL0 Register and PSL1 Register**



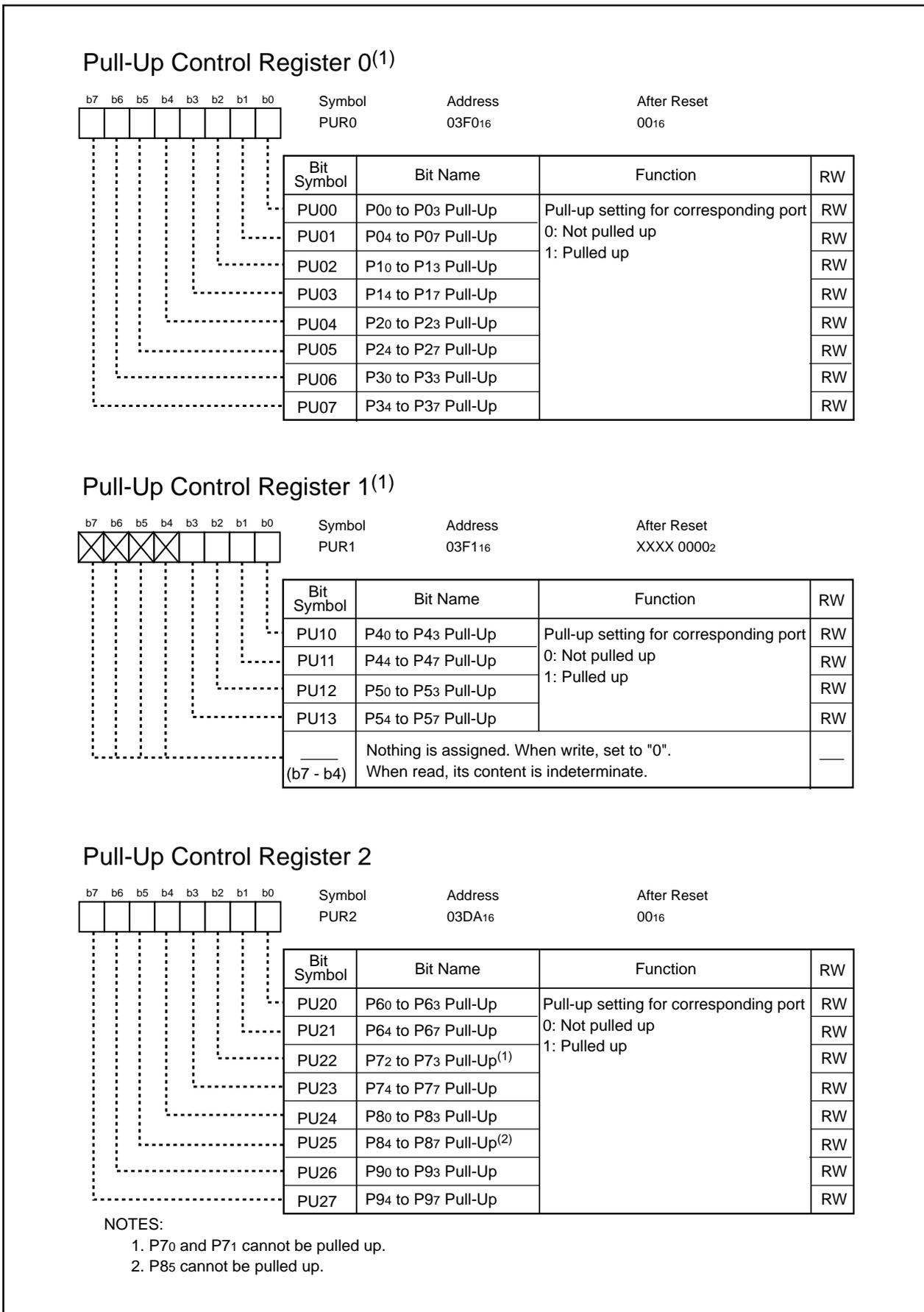
**Figure 23.12 PSL2 Register and PSL3 Register**



**Figure 23.13 PSC Register and PSC2 Register**



**Figure 23.14 PSC3 Register and PSD1 Register**



**Figure 23.15 PUR0 Register, PUR1 Register and PUR2 Register**

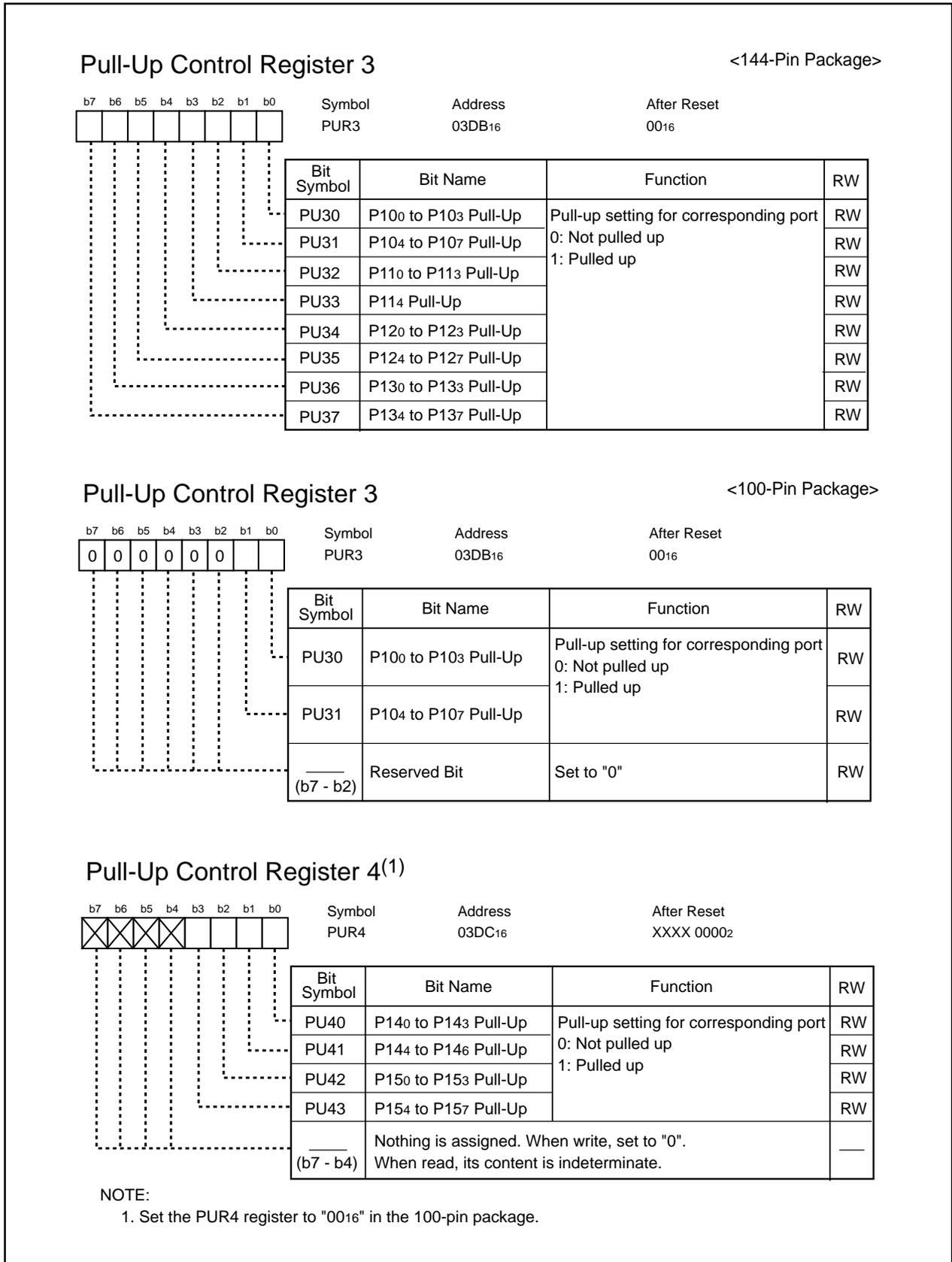


Figure 23.16 PUR3 Register and PUR4 Register

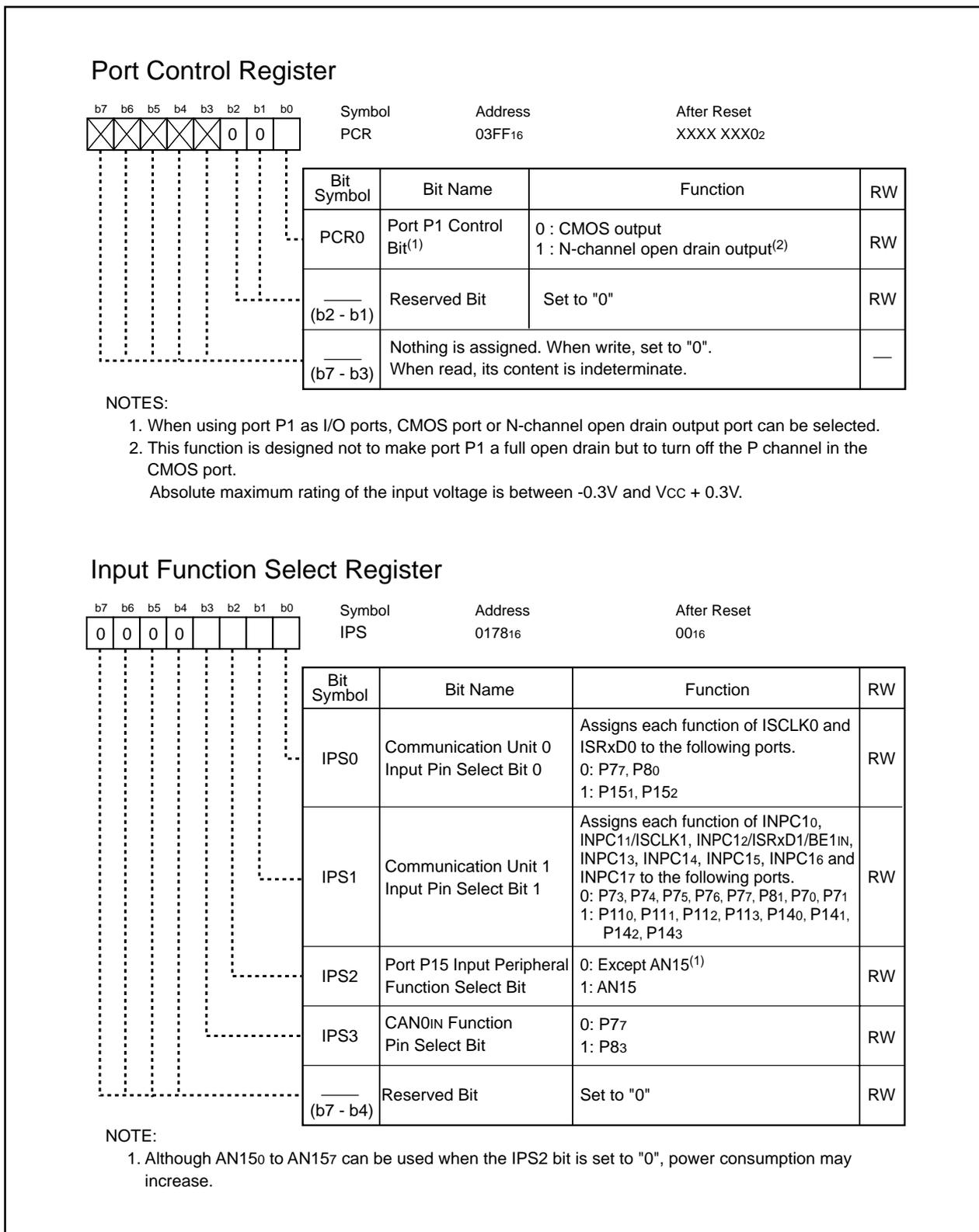
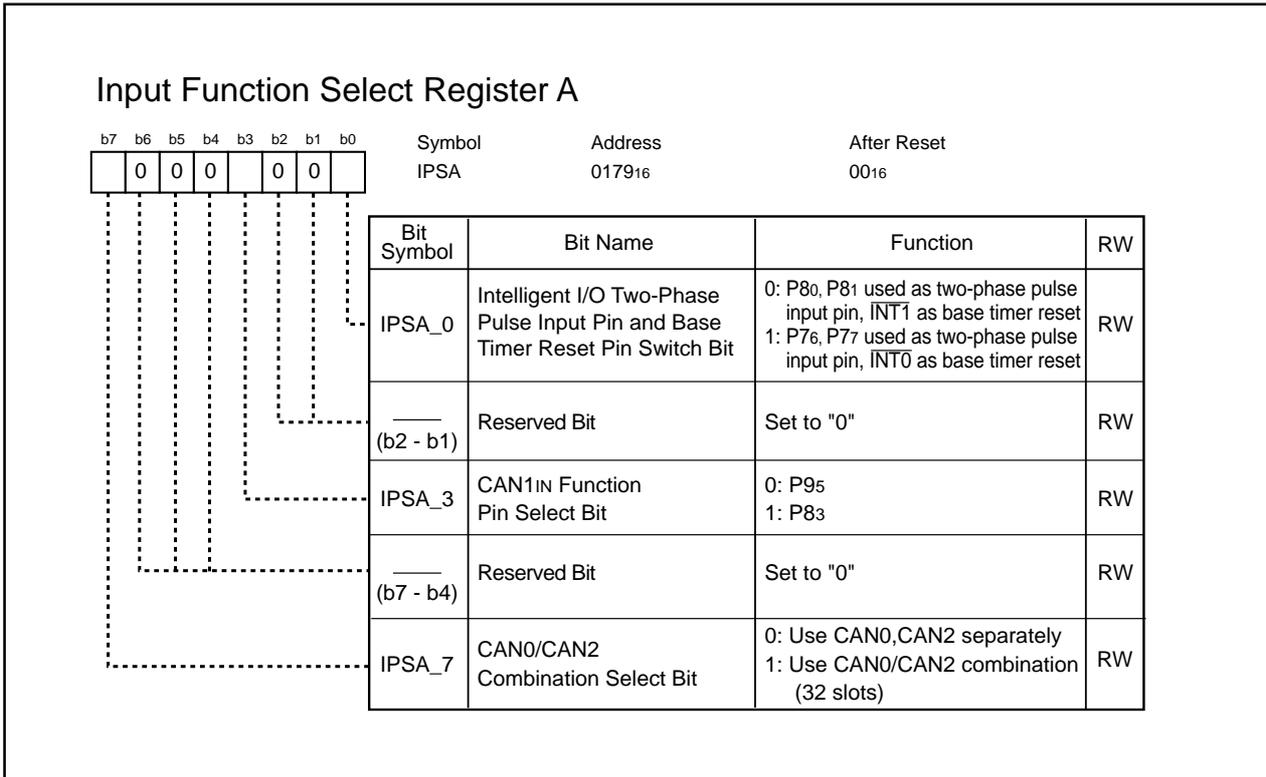


Figure 23.17 PCR Register and IPS Register



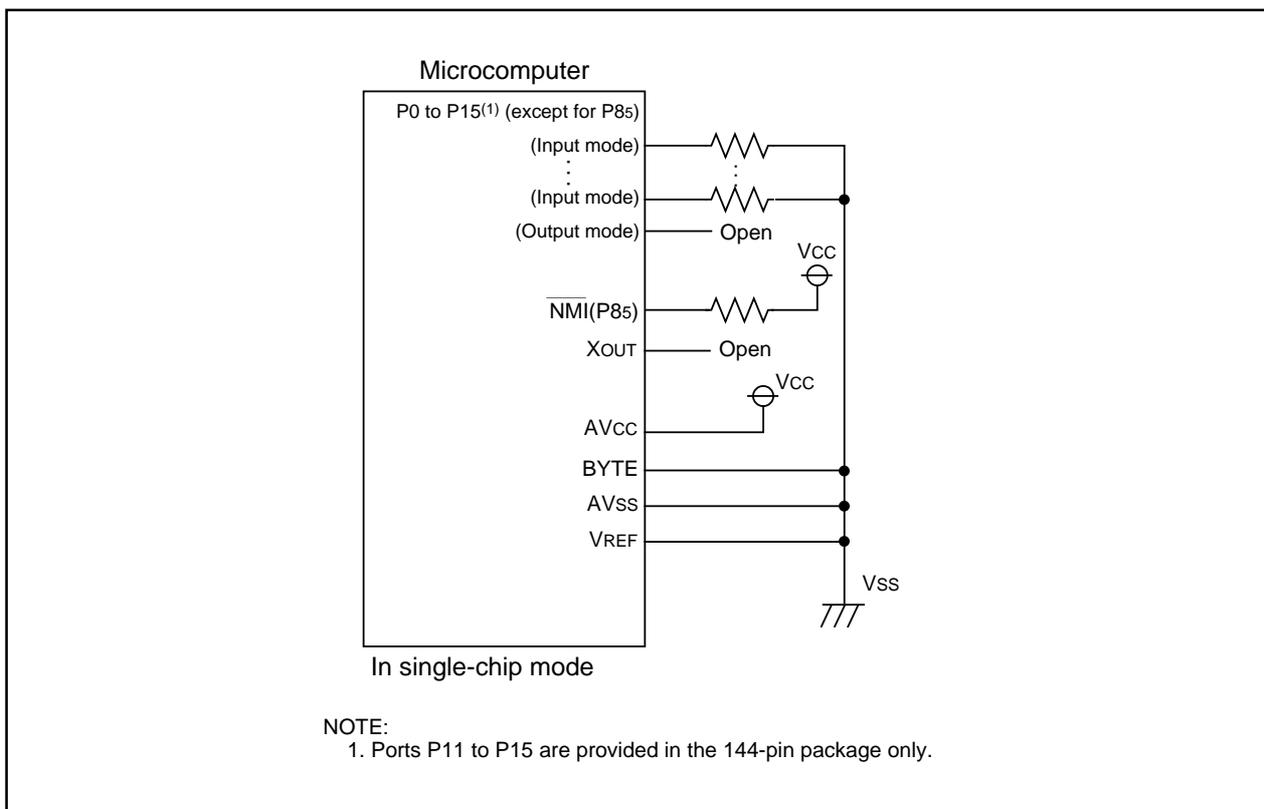
**Figure 23.18** IPSA Register

**Table 23.1 Unassigned Pin Settings in Single-Chip Mode**

Pin Name	Setting
P0 to P15 (excluding P85) <sup>(1,2,3,4,6)</sup>	Enter input mode and connect each pin to V <sub>SS</sub> via a resistor (pull-down); or enter output mode and leave the pins open
XOUT <sup>(5)</sup>	Leave pin open
NMI(P85)	Connect pin to V <sub>CC</sub> via a resistor (pull-up)
AVCC	Connect pin to V <sub>CC</sub>
AVSS, VREF, BYTE	Connect pins to V <sub>SS</sub>

**NOTES:**

- Ports P11 to P15 are provided in the 144-pin package only.
- If the port enters output mode and is left open, it is in input mode before output mode is entered by program after reset. While the port is in input mode, voltage level on the pins is indeterminate and power consumption may increase.  
Direction register settings may be changed by noise or failure caused by noise. Configure direction register settings regularly to increase the reliability of the program.
- Use the shortest possible wiring to connect the microcomputer pins to unassigned pins (within 2 cm).
- Ports P70 and P71 must output low-level ("L") signals if they are in output mode. They are ports for the N-channel open drain output.
- When the external clock is applied to the XIN pin, set the pin as written above.
- In the 100-pin package, set "FF16" in the following addresses, in addition to the above settings:  
Addresses 0003CB<sub>16</sub>, 0003CE<sub>16</sub>, 0003CF<sub>16</sub>, 0003D2<sub>16</sub>, 0003D3<sub>16</sub>

**Figure 23.19 Unassigned Pin Handling**

**Table 23.2 Port P6 Peripheral Function Output Control**

	PS0 Register	PSL0 Register
Bit 0	0: P60/CTS0/SS0 1: Selected by the PSL0 register	0: RTS0 1: CAN2OUT
Bit 1	0: P61/CLK0(input)/CAN2IN/CAN2WU 1: CLK0(output)	Set to "0"
Bit 2	0: P62/RxD0/SCL0(input) 1: Selected by the PSL0 register	0: SCL0(output) 1: STxD0
Bit 3	0: P63/SRxD0/SDA0(input) 1: TxD0/SDA0 (output)	Set to "0"
Bit 4	0: P64/CTS1/SS1 1: Selected by the PSL0 register	0: RTS1 1: Do not set this value
Bit 5	0: P65/CLK1(input) 1: CLK1(output)	Set to "0"
Bit 6	0: P66/RxD1/SCL1(input) 1: Selected by the PSL0 register	0: SCL1(output) 1: STxD1
Bit 7	0: P67/SRxD1/SDA1(input) 1: TxD1/SDA1(output)	Set to "0"

**Table 23.3 Port P7 Peripheral Function Output Control**

	PS1 Register	PSL1 Register	PSC Register <sup>(1)</sup>	PSD1 Register
Bit 0	0: P70/TA0OUT(input)/SRxD2 INPC16/SDA2 (input) 1: Selected by the PSL1 register	0: Selected by the PSC register 1: TA0OUT(output)	0: TxD2/SDA2(output) 1: Selected by the PSD1 register	0: Do not set to this value 1: OUTC16
Bit 1	0: P71/TB5IN/TA0IN/RxD2/ INPC17/SCL2 (input) 1: Selected by the PSL1 register	0: Selected by the PSC register 1: STxD2	0: SCL2(output) 1: Selected by the PSD1 register	0: Do not set to this value 1: OUTC17
Bit 2	0: P72/TA1OUT(input)/ CLK2(input) 1: Selected by the PSL1 register	0: Selected by the PSC register 1: TA1OUT(output)	0: CLK2(output) 1: V	Set to "0"
Bit 3	0: P73/TA1IN/CTS2/SS2/ INPC10 1: Selected by the PSL1 register	0: Selected by the PSC register 1: $\bar{V}$	0: RTS2 1: OUTC10/ISTxD1/BE1OUT	Set to "0"
Bit 4	0: P74/INPC11/ISCLK1(input)/ TA2OUT(input) 1: Selected by the PSL1 register	0: Selected by the PSC register 1: W	0: TA2OUT(output) 1: OUTC11/ISCLK1(output)	Set to "0"
Bit 5	0: P75/TA2IN/INPC12/ ISRxD1/BE1IN 1: Selected by the PSL1 register	0: $\bar{W}$ 1: OUTC12	Set to "0"	Set to "0"
Bit 6	0: P76/INPC13/TA3OUT(input) 1: Selected by the PSL1 register	0: Selected by the PSC register 1: TA3OUT(output)	0: Selected by the PSD1 register 1: CAN0OUT	0: ISTxD0 1: OUTC13
Bit 7	0: P77/TA3IN/CAN0IN/ ISCLK0(input)/INPC14 1: Selected by the PSL1 register	0: ISCLK0(output) 1: OUTC14	0: P104 to P107 or $\bar{K}I0$ to $\bar{K}I3$ 1: AN4 to AN7 (No relation to P77)	Set to "0"

**NOTE:**

1. When setting the PSL1\_i bit (i=0 to 4, 6) to "1", set the corresponding PSC\_i bit to "0".

**Table 23.4 Port P8 Peripheral Function Output Control**

	PS2 Register	PSL2 Register	PSC2 Register
Bit 0	0: P80/ISRxD0/TA4OUT(input) 1: Selected by the PSL2 register	0: TA4OUT(output) 1: U	Set to "0"
Bit 1	0: P81/TA4IN/INPC15 1: Selected by the PSL2 register	0: $\bar{U}$ 1: Selected by the PSC2 register	0: Do not set to this value 1: OUTC15
Bit 2	0: P82/INT0 1: Selected by the PSL2 register	0: Do not set to this value 1: Selected by the PSC2 register	0: CAN0OUT 1: CAN1OUT
Bit 3 to 7	Set to "000002"		

**Table 23.5 Port P9 Peripheral Function Output Control**

	PS3 Register	PSL3 Register	PSC3 Register
Bit 0	0: P90/TB0IN/CLK3(input) 1: CLK3(output)	Set to "0"	Set to "0"
Bit 1	0: P91/TB1IN/RxD3/SCL3(input) 1: Selected by the PSL3 register	0: SCL3(output) 1: STxD3	Set to "0"
Bit 2	0: P92/TB2IN/SRxD3/SDA3(input) 1: Selected by the PSL3 register	0: TxD3/SDA3(output) 1: Do not set to this value	Set to "0"
Bit 3	0: P93/TB3IN/CTS3/SS3/DA0(output) 1: RTS3	0: Except DA0 1: DA0	Set to "0"
Bit 4	0: P94/TB4IN/CTS4/SS4/DA1(output) 1: RTS4	0: Except DA1 1: DA1	Set to "0"
Bit 5	0: P95/ANEX0/CLK4(input)/CAN1IN/ CAN1WU 1: CLK4(output)	0: Except ANEX0 1: ANEX0	Set to "0"
Bit 6	0: P96/SRxD4/ANEX1/SDA4(input) 1: Selected by the PSC3 register	0: Except ANEX1 1: ANEX1	0: TxD4/SDA4 1: CAN1OUT
Bit 7	0: P97/RxD4/ADTRG/SCL4(input) 1: Selected by the PSL3 register	0: SCL4(output) 1: STxD4	Set to "0"

**Table 23.6 Port P10 Peripheral Function Output Control**

	PSC Register
Bit 7	0: P104 to P107 or $\bar{K}I0$ to $\bar{K}I3$ 1: AN4 to AN7

**Table 23.7 Port P11 Peripheral Function Output Control**

	PS5 Register
Bit 0	0: P110/INPC10 1: OUTC10/ISTxD1/BE1OUT
Bit 1	0: P111/INPC11/ISCLK1(input) 1: OUTC11/ISCLK1(output)
Bit 2	0: P112/INPC12/ISRxD1/BE1IN 1: OUTC12
Bit 3	0: P113/INPC13 1: OUTC13
Bit 4 to 7	Set to "00002"

**Table 23.8 Port P14 Peripheral Function Output Control**

	PS8 Register
Bit 0	0: P140/INPC14 1: OUTC14
Bit 1	0: P141/INPC15 1: OUTC15
Bit 2	0: P142/INPC16 1: OUTC16
Bit 3	0: P143/INPC17 1: OUTC17
Bit 4 to 7	Set to "00002"

**Table 23.9 Port P15 Peripheral Function Output Control**

	PS9 Register
Bit 0	0: P150/AN150 1: ISTxD0
Bit 1	0: P151/AN151/ISCLK0(input) 1: ISCLK0(output)
Bit 2 to 7	Set to "0000002"

## 24. Flash Memory Version

Aside from the built-in flash memory, the flash memory version microcomputer has the same functions as the masked ROM version.

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

Table 24.1 lists specifications of the flash memory version. See **Tables 1.1 and 1.2** for the items not listed in Table 24.1.

**Table 24.1 Flash Memory Version Specifications**

Item		Specification
Flash Memory Operating Mode		3 modes (CPU rewrite, standard serial I/O, parallel I/O)
Erase Block	User ROM Area	See Figure 24.1
	Boot ROM Area	1 block (4 Kbytes) <sup>(1)</sup>
Program Method		Per word (16 bytes), per byte (8 bits) <sup>(2)</sup>
Erase Method		All block erase, erase per block
Program and Erase Control Method		Software commands control programming and erasing on the flash memory
Protect Method		The lock bit protects each block in the flash memory
Number of Commands		8 commands
Program and Erase Endurance		100 times <sup>(3)</sup>
Data Retention		10 years
ROM Code Protection		Standard serial I/O mode and parallel I/O mode supported

**NOTES:**

1. The rewrite control program for standard serial I/O mode is stored in the boot ROM area before shipment. This space can be rewritten in parallel I/O mode only.
2. Programming per byte is available in parallel I/O mode only.
3. Program and erase endurance refers to the number of times a block erase can be performed. Every block erase performed after writing data of one word or more counts as one program and erase operation.

**Table 24.2 Flash Memory Rewrite Mode Overview**

Flash Memory Rewrite Mode	CPU Rewrite Mode	Standard Serial I/O Mode	Parallel I/O Mode
Function	Software command execution by CPU rewrites the user ROM area. EW mode 0: Rewritable in areas other than flash memory EW mode 1: Rewritable in flash memory	A dedicated serial programmer rewrites the user ROM area. Standard serial I/O mode 1: Clock synchronous serial I/O Standard serial I/O mode 2: UART Standard serial I/O mode 3: CAN	A dedicated parallel programmer rewrites the boot ROM area and user ROM area.
Rewritable Space	User ROM area	User ROM area	User ROM area Boot ROM area
Operating Mode	Single-chip mode Memory expansion mode (EW mode 0) Boot mode (EW mode 0)	Boot mode	Parallel I/O mode
Programmer	None	Serial programmer	Parallel programmer

### 24.1 Memory Map

The flash memory includes the user ROM area and the boot ROM area. The user ROM area has space to store the microcomputer operating programs in single-chip mode and a separate 4-kbyte space as the block A. Figure 24.1 shows a block diagram of the flash memory.

The user ROM area is divided into several blocks, each of which can be protected (locked) from program or erase. The user ROM area can be rewritten in CPU rewrite mode, standard serial I/O mode and parallel I/O mode.

The boot ROM area is located at the same addresses as the user ROM area. It can only be rewritten in parallel I/O mode. A program in the boot ROM area is executed after a hardware reset occurs while a high-level ("H") signal is applied to the CNVss and P50 pins and a low-level ("L") signal is applied to the P5s pin. A program in the user ROM area is executed after a hardware reset occurs while an "L" signal is applied to the CNVss pin. Consequently, the boot ROM area cannot be read.

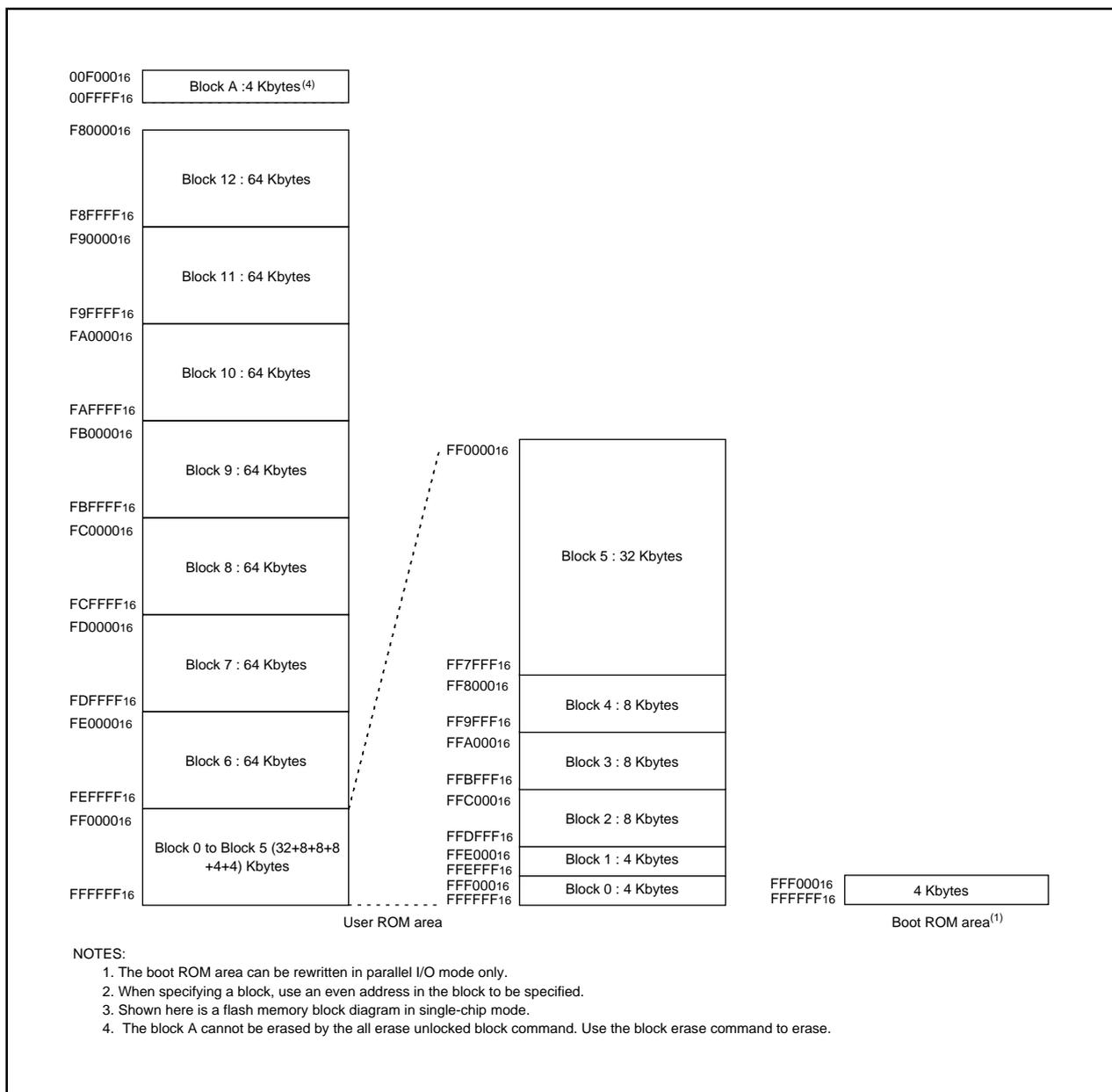


Figure 24.1 Flash Memory Block Diagram

### 24.1.1 Boot Mode

The microcomputer enters boot mode when a hardware reset is performed while a high-level ("H") signal is applied to the CNVss and P50 pins and a low-level ("L") signal is applied to the P55 pin. A program in the boot ROM area is executed.

In boot mode, the FMR05 bit in the FMR0 register selects access to either the boot ROM area or the user ROM area.

In the factory setting, the rewrite control program for standard serial I/O mode is stored into the boot ROM area.

The boot ROM area can be rewritten in parallel I/O mode only. If any rewrite control program using erase-write mode 0 (EW mode 0) is written in the boot ROM area, the flash memory can be rewritten according to the system implemented.

## 24.2 Functions to Prevent Rewriting of Flash Memory

The flash memory has the ROM code protect function for parallel I/O mode and the ID code verify function for standard I/O mode to prevent the flash memory from reading or rewriting.

### 24.2.1 ROM Code Protect Function

The ROM code protect function prevents the flash memory from reading and rewriting in parallel I/O mode.

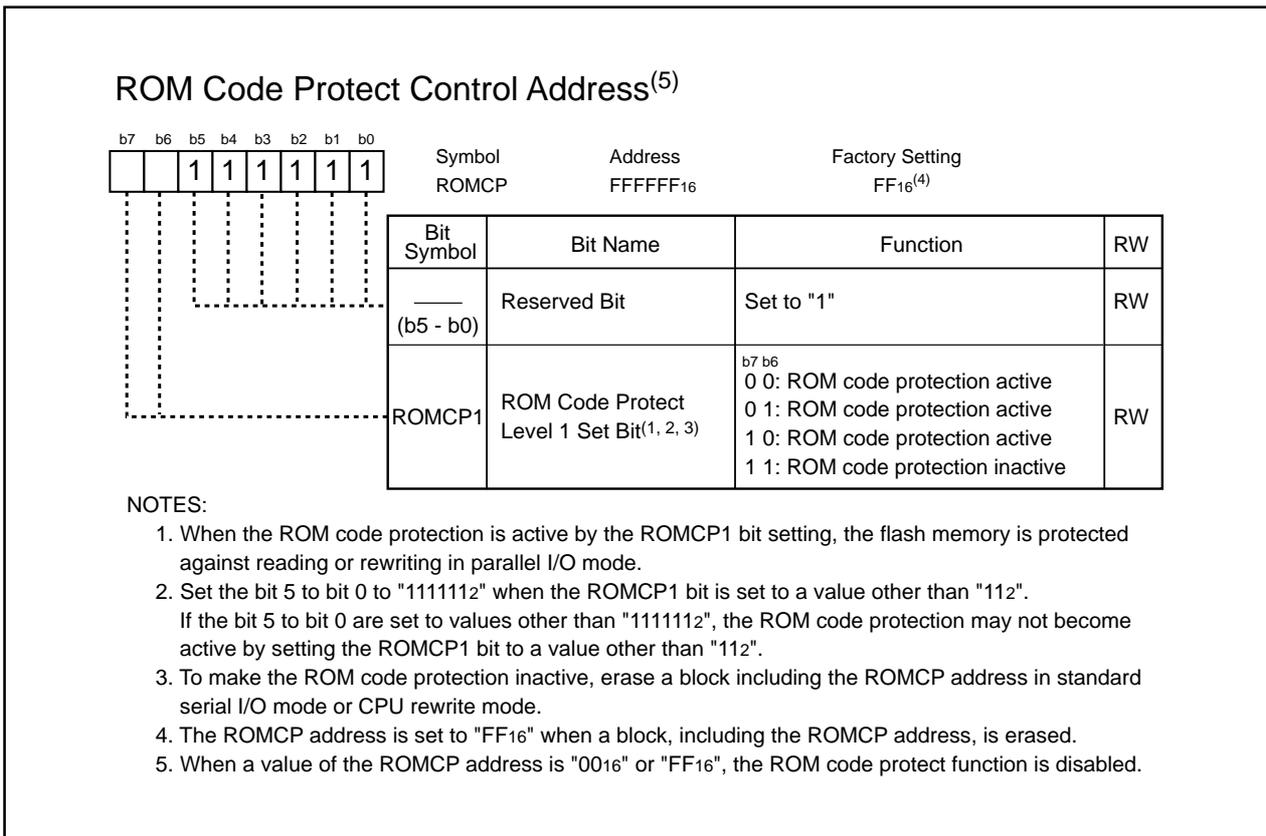
Figure 24.2 shows the ROMCP register. The ROMCP register is located in the user ROM area.

The ROM code protect function is enabled when the ROMCP1 bit is set to "002", "012" or "102".

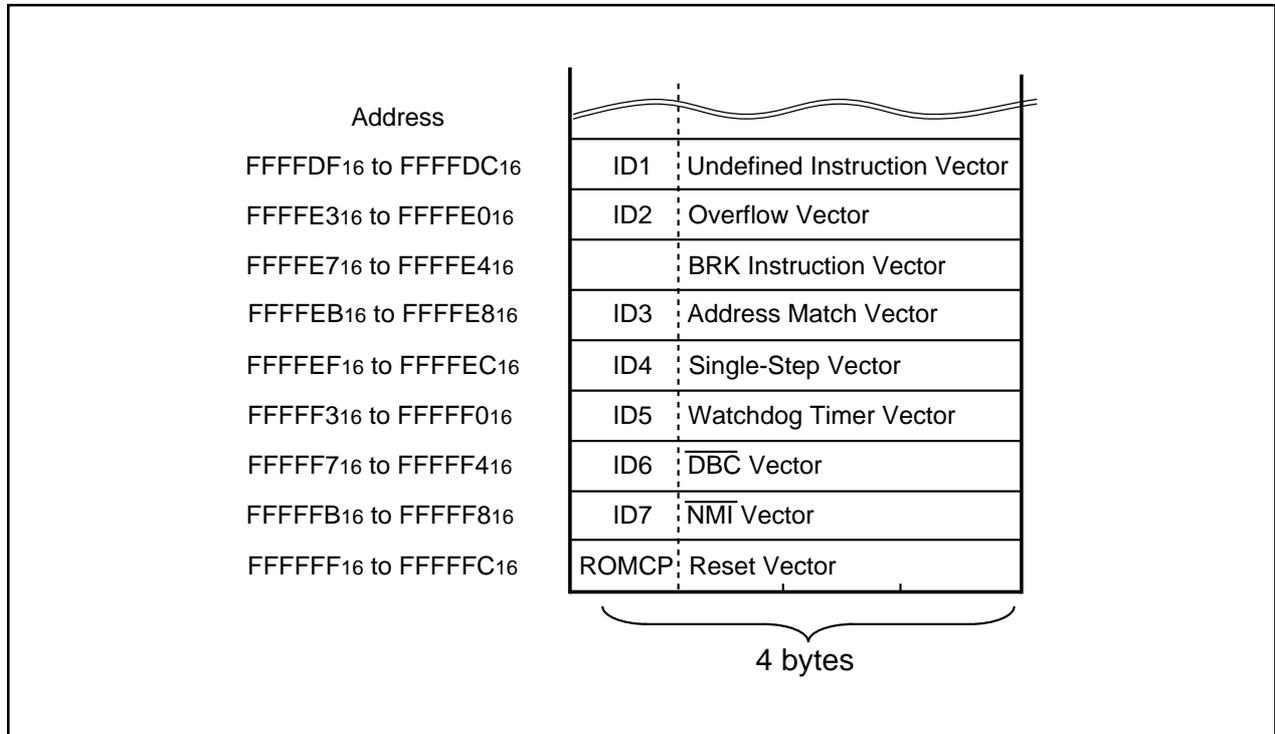
### 24.2.2 ID Code Verify Function

Use the ID code verify function in standard serial I/O mode. The ID code sent from the serial programmer is compared with the ID code written in the flash memory for a match. If the ID codes do not match, commands sent from the serial programmer are not accepted. However, if the four bytes of the reset vector are "FFFFFFFF<sub>16</sub>", ID codes are not compared, allowing all commands to be accepted.

The ID codes are 7-byte data stored consecutively, starting with the first byte, into addresses 0FFFFFFDF<sub>16</sub>, 0FFFFFFE3<sub>16</sub>, 0FFFFFFEB<sub>16</sub>, 0FFFFFFEF<sub>16</sub>, 0FFFFFFF3<sub>16</sub>, 0FFFFFFF7<sub>16</sub> and 0FFFFFFFB<sub>16</sub>. The flash memory must have a program with the ID codes set in these addresses.



**Figure 24.2 ROMCP Address**



**Figure 24.3 Address for ID Code Stored**

### 24.3 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten when the CPU executes software commands. The user ROM area can be rewritten with the microcomputer mounted on a board without using a parallel or serial programmer.

In CPU rewrite mode, only the user ROM area shown in Figure 24.1 can be rewritten. The boot ROM area cannot be rewritten. The program and block erase commands are executed only for each block in the user ROM area.

Erase-write (EW) mode 0 and erase-write mode 1 are provided as CPU rewrite mode. Table 24.3 lists differences between EW mode 0 and EW mode 1.

**Table 24.3 EW Mode 0 and EW Mode 1**

Item	EW mode 0	EW mode 1
Operating Mode	<ul style="list-style-type: none"> <li>Single-chip mode</li> <li>Boot mode</li> </ul>	Single-chip mode
Space where the rewrite control program can be placed	<ul style="list-style-type: none"> <li>User ROM area</li> <li>Boot ROM area</li> </ul>	User ROM area
Space where the rewrite control program can be executed	The rewrite control program must be transferred to any space other than the flash memory (e.g.,RAM) before being executed	The rewrite control program can be executed in the user ROM area
Space which can be rewritten	User ROM area	User ROM area However, this excludes blocks with the rewrite control program
Software Command Restriction	None	<ul style="list-style-type: none"> <li>Program and block erase commands cannot be executed in a block having the rewrite control program.</li> <li>Erase all unlocked block command cannot be executed when the lock bit in a block having the rewrite control program is set to "1"(unlocked) or when the FMR02 bit in the FMR0 register is set to "1"(lock bit disabled).</li> <li>Read status register command cannot be used.</li> </ul>
Mode after Programming or Erasing	Read status register mode	Read array mode
CPU State during Auto Program and Erase Operation	Operating	In a hold state (I/O ports maintains the state before the command was executed) <sup>(1)</sup>
Flash Memory State Detection	<ul style="list-style-type: none"> <li>Read the FMR00, FMR06 and FMR07 bits in the FMR0 register by program</li> <li>Execute the read status register command to read the SR7, SR5 and SR4 bits in the SRD register</li> </ul>	Read the FMR00, FMR06 and FMR07 bits in the FMR0 register by program

**NOTE:**

- Do not generate an interrupt (except  $\overline{\text{NMI}}$  interrupt) or a DMA transfer.

#### 24.3.1 EW Mode 0

The microcomputer enters CPU rewrite mode by setting the FMR01 bit in the FMR0 register to "1" (CPU rewrite mode enabled) and is ready to accept commands. EW mode 0 is selected by setting the FMR11 bit in the FMR1 register to "0". To set the FMR01 bit to "1", set to "1" after first writing "0".

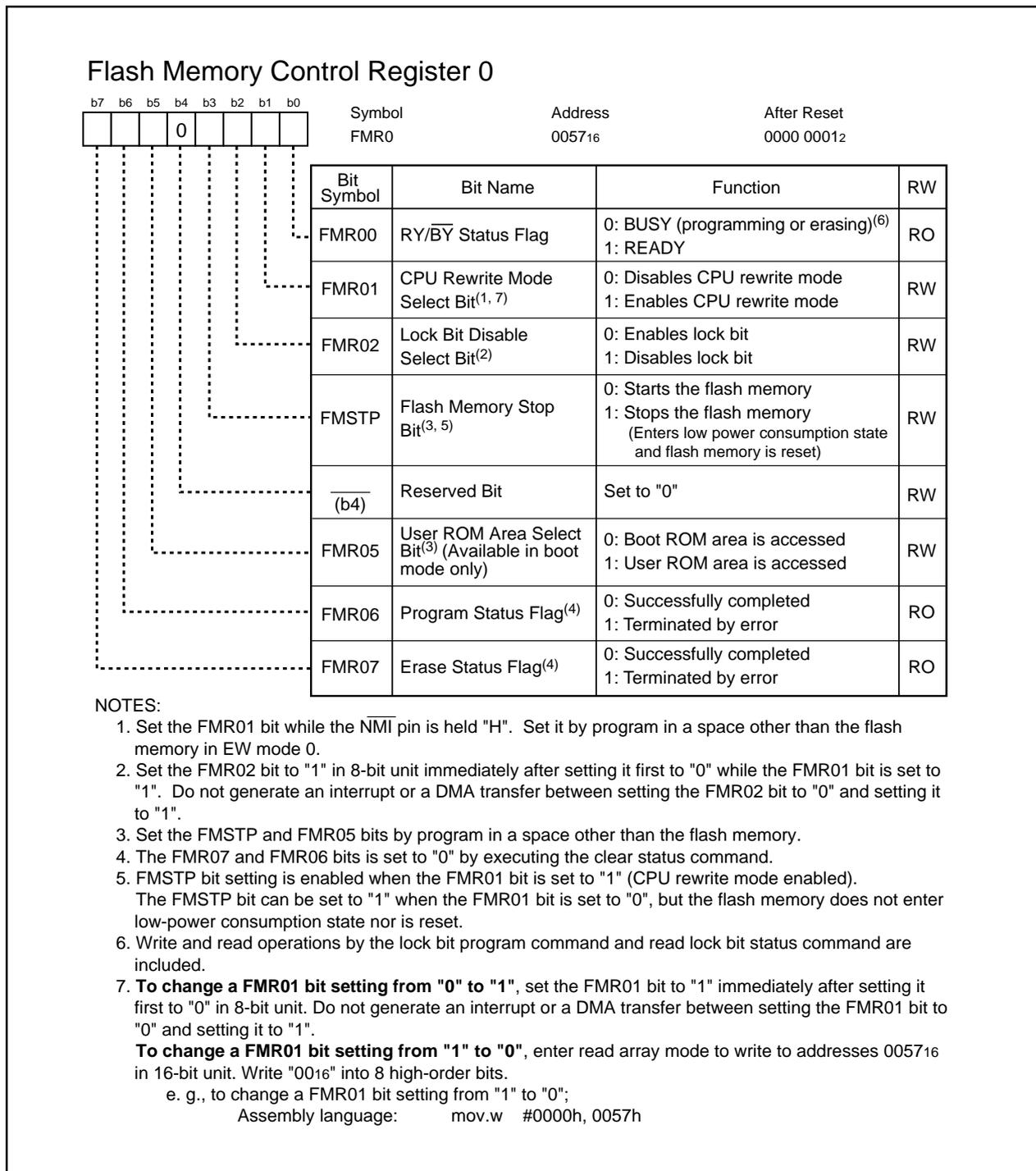
The software commands control programming and erasing. The FMR0 register or the SRD register indicates whether a program or erase operation is completed as expected or not.

#### 24.3.2 EW Mode 1

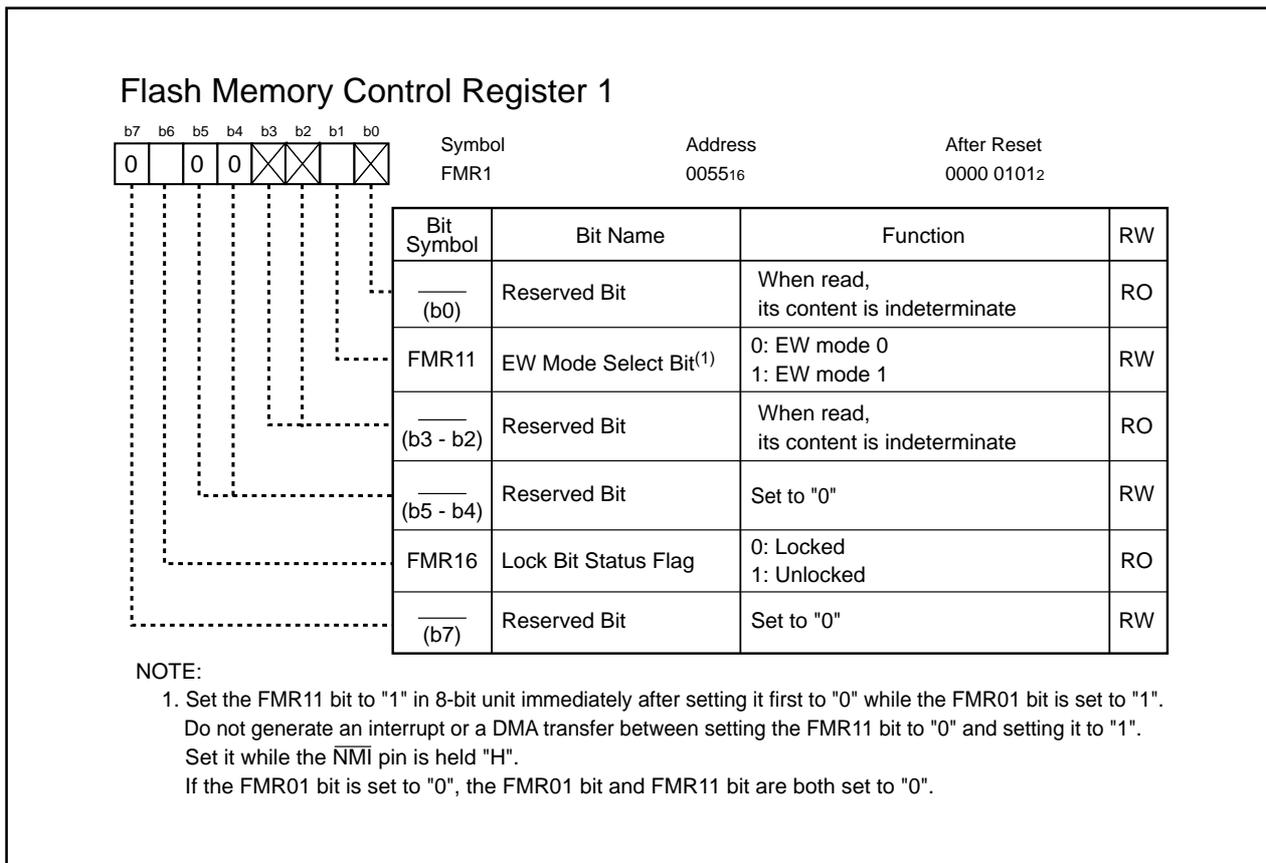
EW mode 1 is selected by setting the FMR11 bit to "1" after the FMR01 bit is set to "1". (Both bits must be set to "0" first before setting to "1".)

The FMR0 register indicates whether or not a program or erase operation has been completed as expected. The SRD register cannot be read in EW mode 1.

### 24.3.3 Flash Memory Control Register (FMR0 Register and FMR1 Register)



**Figure 24.4 FMR0 Register**



**Figure 24.5 FMR1 Register**

#### 24.3.3.1 FMR00 Bit

The FMR00 bit indicates the flash memory operating state. It is set to "0" while the program, block erase, erase all unlocked block, lock bit program, or read lock bit status command is being executed; otherwise, it is set to "1".

#### 24.3.3.2 FMR01 Bit

The microcomputer can accept commands when the FMR01 bit is set to "1" (CPU rewrite mode). Set the FMR05 bit to "1" (user ROM area access) as well if in boot mode.

#### 24.3.3.3 FMR02 Bit

The lock bit is invalid by setting the FMR02 bit to "1" (lock bit disabled). (Refer to **24.3.6 Data Protect Function**.) The lock bit is valid by setting the FMR02 bit to "0" (lock bit enabled).

The FMR02 bit does not change the lock bit status but disables the lock bit function. If the block erase or erase all unlocked block command is executed when the FMR02 bit is set to "1", the lock bit status changes "0" (locked) to "1" (unlocked) after command execution is completed.

#### 24.3.3.4 FMSTP Bit

The FMSTP bit initializes the flash memory control circuits and minimizes power consumption in the flash memory. Access to the flash memory is disabled when the FMSTP bit is set to "1". Set the FMSTP bit by program in a space other than the flash memory.

Set the FMSTP bit to "1" if one of the followings occurs:

- A flash memory access error occurs while erasing or programming in EW mode 0 (FMR00 bit does not switch back to "1" (ready)).
- Low-power consumption mode or on-chip low-power consumption mode is entered.

Use the following the procedure to change the FMSTP bit setting.

- (1) Set the FMSTP bit to "1"
- (2) Set tps (the wait time to stabilize flash memory circuit)
- (3) Set the FMSTP bit to "0"
- (4) Set tps (the wait time to stabilize flash memory circuit)

Figure 24.8 shows a flow chart illustrating how to start and stop the flash memory before and after entering low power mode. Follow the procedure on this flow chart.

When entering stop or wait mode, the flash memory is automatically turned off. When exiting stop or wait mode, the flash memory is turned back on. The FMR0 register does not need to be set.

#### 24.3.3.5 FMR05 Bit

The FMR05 bit selects the boot ROM or user ROM area in boot mode. Set to "0" to access (read) the boot ROM area or to "1" (user ROM access) to access (read, write or erase) the user ROM area.

#### 24.3.3.6 FMR06 Bit

The FMR06 bit is a read-only bit indicating an auto program operation state. The FMR06 bit is set to "1" when a program error occurs; otherwise, it is set to "0". Refer to **24.3.8 Full Status Check**.

#### 24.3.3.7 FMR07 Bit

The FMR07 bit is a read-only bit indicating the auto erase operation state. The FMR07 bit is set to "1" when an erase error occurs; otherwise, it is set to "0". For details, refer to **24.3.8 Full Status Check**.

Figure 24.6 shows how to enter and exit EW mode 0. Figure 24.7 shows how to enter and exit EW mode 1.

#### 24.3.3.8 FMR11 Bit

EW mode 0 is entered by setting the FMR11 bit to "0" (EW mode 0).

EW mode 1 is entered by setting the FMR11 bit to "1" (EW mode 1).

#### 24.3.3.9 FMR16 Bit

The FMR16 bit is a read-only bit indicating the execution result of the read lock bit status command. When the block, where the read lock bit status command is executed, is locked, the FMR16 bit is set to "0". When the block, where the read lock bit status command is executed, is unlocked, the FMR16 bit is set to "1".

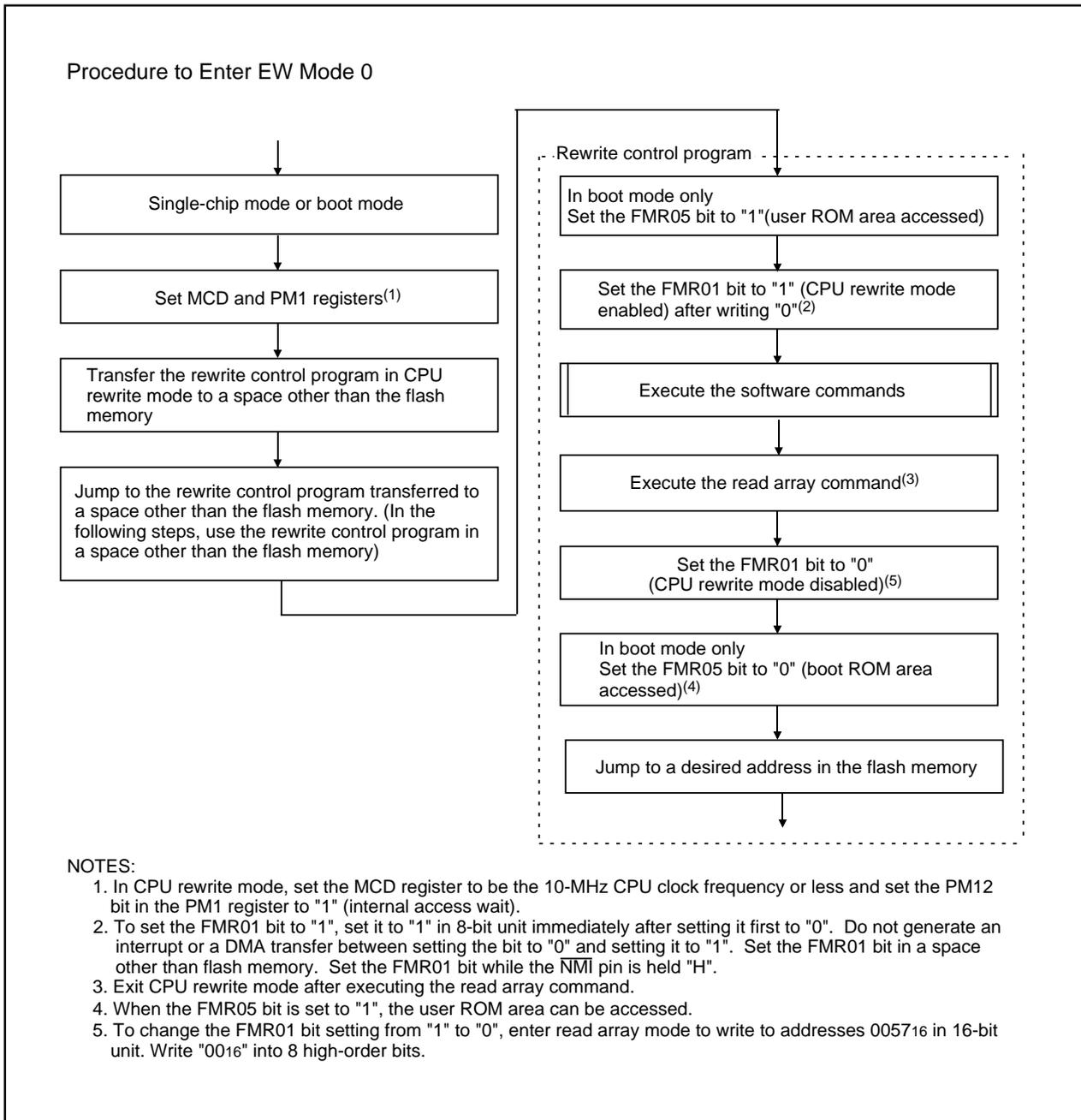
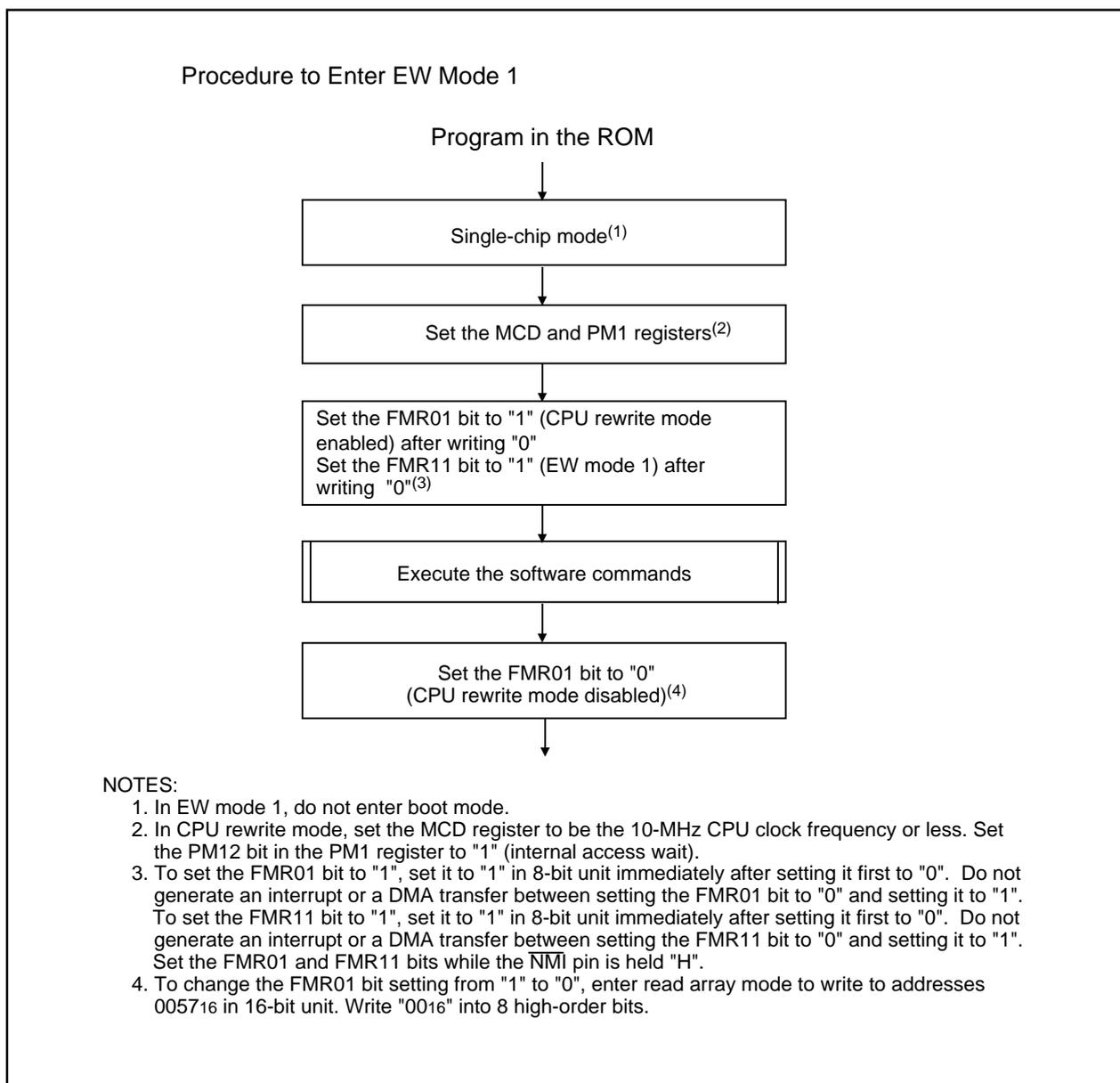
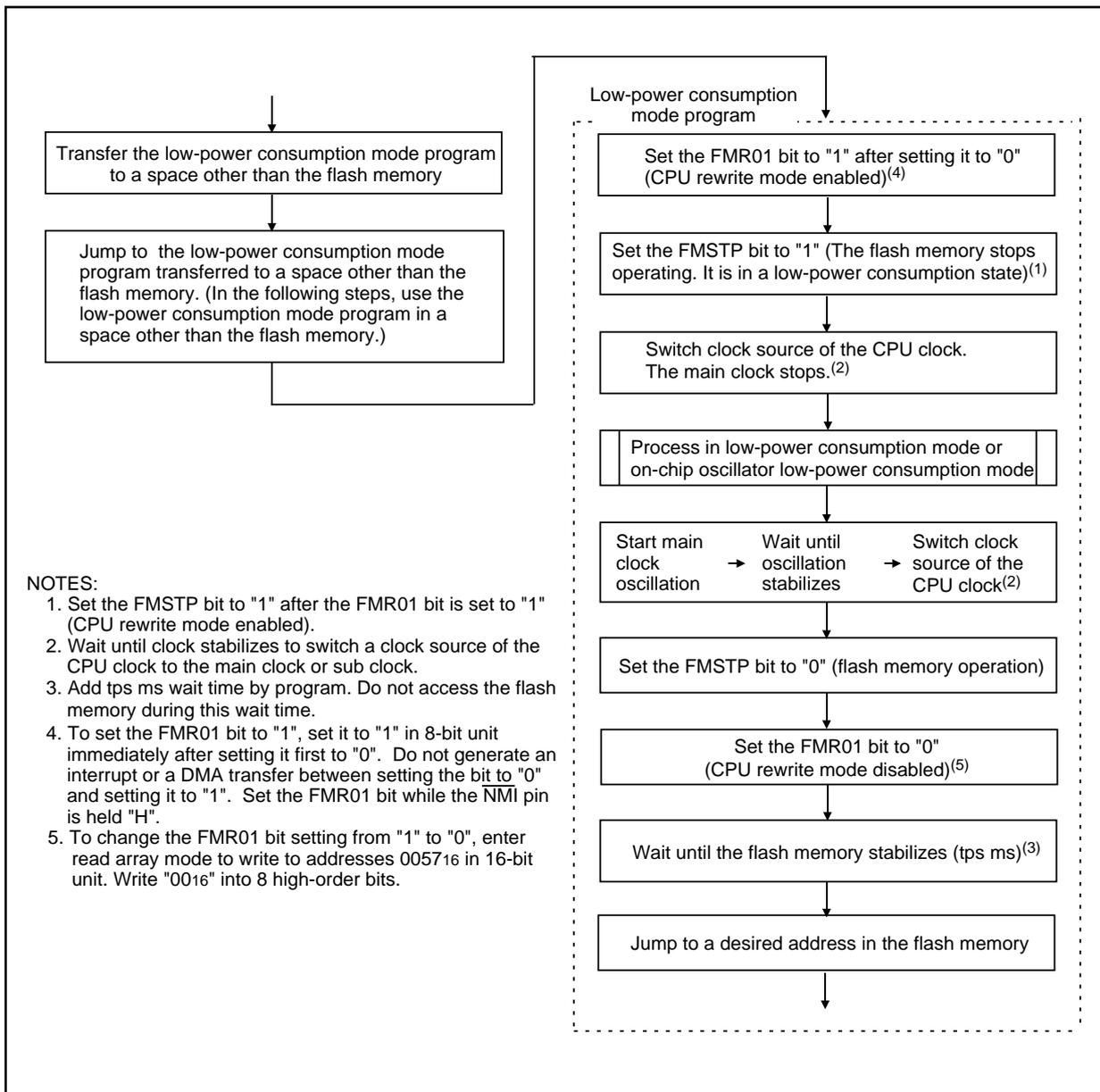


Figure 24.6 How to Enter and Exit EW Mode 0



**Figure 24.7 How to Enter and Exit EW Mode 1**



**Figure 24.8 Handling Before and After Low Power Consumption Mode**

## 24.3.4 Precautions in CPU Rewrite Mode

### 24.3.4.1 Operating Speed

Set the MCD4 to MCD0 bits in the MCD register to CPU clock frequency of 10 MHz or less before entering CPU rewrite mode (EW mode 0 or EW mode 1). Also, set the PM12 bit in the PM1 register to "1" (wait state).

### 24.3.4.2 Prohibited Instructions

The following instructions cannot be used in EW mode 0 because the CPU tries to read data in the flash memory: the UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction.

### 24.3.4.3 Interrupts (EW Mode 0)

- To use interrupts having vectors in a relocatable vector table, the vectors must be relocated to the RAM area.
- The  $\overline{\text{NMI}}$  and watchdog timer interrupts are available since the FMR0 and FMR1 registers are forcibly reset when either interrupt occurs. Allocate the forward addresses for each interrupt routine to the fixed vector table. Flash memory rewrite operation is aborted when the  $\overline{\text{NMI}}$  or watchdog timer interrupt occurs. Execute the rewrite program again after exiting the interrupt routine.
- The address match interrupt is not available since the CPU tries to read data in the flash memory.

### 24.3.4.4 Interrupts (EW Mode 1)

- Do not acknowledge any interrupts with vectors in the relocatable vector table or address match interrupt during the auto program or auto erase period.
- Do not use the watchdog timer interrupt.
- The  $\overline{\text{NMI}}$  interrupt is available since the FMR0 and FMR1 registers are forcibly reset when either interrupt occurs. Allocate the forward address for the interrupt routine to the fixed vector table. Flash memory rewrite operation is aborted when the  $\overline{\text{NMI}}$  interrupt occurs. Execute the rewrite program again after exiting the interrupt routine.

### 24.3.4.5 How to Access

To set the FMR01, FMR02 in the FMR0 register or FMR11 bit in the FMR1 register to "1", set to "1" in 8-bit units immediately after setting to "0". Do not generate an interrupt or a DMA transfer between the instruction to set the bit to "0" and the instruction to set the bit to "1". Set the bit while a high-level ("H") signal is applied to the  $\overline{\text{NMI}}$  pin.

To change the FMR01 bit from "1" to "0", enter read array mode first, and write into address 0057<sub>16</sub> in 16-bit units. Eight high-order bits must be set to "00<sub>16</sub>".

### 24.3.4.6 Rewriting in the User ROM Area (EW Mode 0)

If the supply voltage drops while rewriting the block where the rewrite control program is stored, the flash memory cannot be rewritten because the rewrite control program is not rewritten as expected. If this error occurs, rewrite the user ROM area while in standard serial I/O mode or parallel I/O mode.

### 24.3.4.7 Rewriting in the User ROM Area (EW Mode 1)

Do not rewrite the block where the rewrite control program is stored.

### 24.3.4.8 DMA Transfer

In EW mode 1, do not generate a DMA transfer while the FMR00 bit in the FMR0 register is set to "0" (busy-programming or erasing).

#### 24.2.4.9 Writing Command and Data

Write commands and data to even addresses in the user ROM area.

#### 24.3.4.10 Wait Mode

When entering wait mode, set the FMR01 bit in the FMR0 register to "0" (CPU rewrite mode disabled) before executing the WAIT instruction.

#### 24.3.4.11 Stop Mode

When entering stop mode, the following settings are required:

- Set the FMR01 bit to "0" (CPU rewrite mode disabled). Disable a DMA transfer before setting the CM10 bit to "1" (stop mode).
- Execute the instruction to set the CM10 bit to "1" (stop mode) and then the JMP.B instruction.

e.g.,                    BSET        0, CM1        ; Stop mode

                          JMP.B        L1

                          L1:

                          Program after exiting stop mode

#### 24.3.4.12 Low-Power Consumption Mode and On-Chip Oscillator Low-Power Consumption Mode

If the CM05 bit is set to "1" (main clock stopped), do not execute the following commands:

- Program
- Block erase
- Erase all unlocked blocks
- Lock bit program
- Read lock bit status

### 24.3.5 Software Commands

Read or write 16-bit commands and data from or to even addresses in the user ROM area, in 16-bit units. When writing a command code, 8 high-order bits (D15 to D8) are ignored.

**Table 24.4 Software Commands**

Command	First Bus Cycle			Second Bus Cycle		
	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)
Read Array	Write	X	xxFF <sub>16</sub>			
Read Status Register	Write	X	xx70 <sub>16</sub>	Read	X	SRD
Clear Status Register	Write	X	xx50 <sub>16</sub>			
Program	Write	WA	xx40 <sub>16</sub>	Write	WA	WD
Block Erase	Write	X	xx20 <sub>16</sub>	Write	BA	xxD0 <sub>16</sub>
Erase All Unlocked Block <sup>(1)</sup>	Write	X	xxA7 <sub>16</sub>	Write	X	xxD0 <sub>16</sub>
Lock Bit Program	Write	BA	xx77 <sub>16</sub>	Write	BA	xxD0 <sub>16</sub>
Read Lock Bit Status	Write	X	xx71 <sub>16</sub>	Write	BA	xxD0 <sub>16</sub>

**NOTE:**

1. Blocks 0 to 12 can be erased by the erase all unlocked block command.

Block A cannot be erased. The block erase command must be used to erase the block A.

SRD: Data in the SRD register (D7 to D0)

WA: Address to be written (The address specified in the the first bus cycle is the same even address as the address specified in the second bus cycle.)

WD: 16-bit write data

BA: Highest-order block address (must be an even address)

X: Any even address in the user ROM space

xx: 8 high-order bits of command code (ignored)

#### 24.3.5.1 Read Array Command

The read array command reads the flash memory.

Read array mode is entered by writing command code "xxFF<sub>16</sub>" in the first bus cycle. Content of a specified address can be read in 16-bit units after the next bus cycle.

The microcomputer remains in read array mode until another command is written. Therefore, contents from multiple addresses can be read consecutively.

#### 24.3.5.2 Read Status Register Command

The read status register command reads the SRD register (refer to **24.3.7 Status Register** for detail).

By writing command code "xx70<sub>16</sub>" in the first bus cycle, the SRD register can be read in the second bus cycle. Read an even address in the user ROM area.

Do not execute this command in EW mode 1.

#### 24.3.5.3 Clear Status Register Command

The clear status register command clears the SRD register. By writing "xx50<sub>16</sub>" in the first bus cycle, the FMR07 and FMR06 bits in the FMR0 register are set to "002" and the SR5 and SR4 bits in the SRD register are set to "002".

#### 24.3.5.4 Program Command

The program command writes 1-word, or 2-byte, data to the flash memory.

Auto program operation (data program and verify) will start by writing command code "xx4016" in the first bus cycle and data to the write address in the second bus cycle. The address value specified in the first bus cycle must be the same even address as the write address specified in the second bus cycle.

The FMR00 bit in the FMR0 register indicates whether or not an auto program operation has been completed. The FMR00 bit is set to "0" (busy) during auto program and to "1" (ready) when the auto program operation is completed.

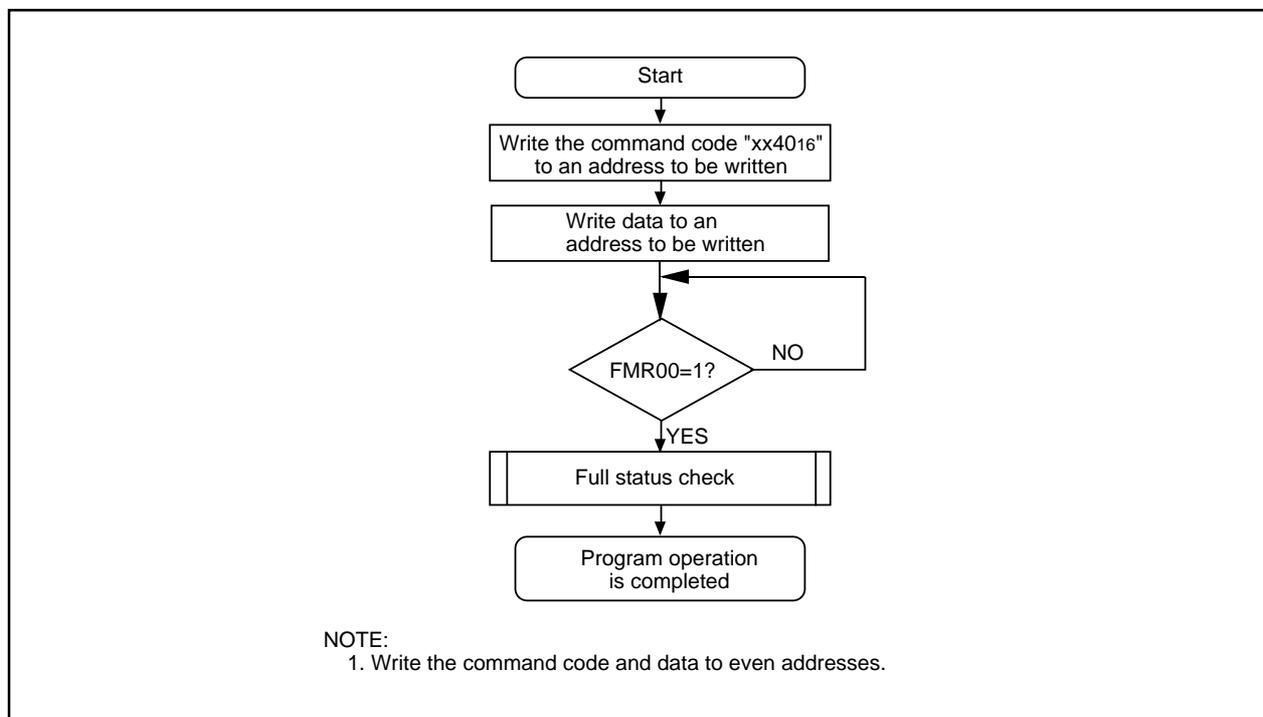
After the completion of auto program operation, the FMR06 bit in the FMR0 register indicates whether or not the auto program operation has been completed as expected. (Refer to **24.3.8 Full Status Check.**)

An address that is already written cannot be altered or rewritten.

Figure 24.9 shows a flow chart of the program command programming.

The lock bit can protect each block from being programmed inadvertently. (Refer to **24.3.6 Data Protect Function.**)

In EW mode 1, do not execute this command on the block where the rewrite control program is allocated. In EW mode 0, the microcomputer enters read status register mode as soon as an auto program operation starts. The SRD register can be read. The SR7 bit in the SRD register is set to "0" at the same time an auto program operation starts. It is set to "1" when an auto program operation is completed. The microcomputer remains in read status register mode until the read array command is written. After completion of an auto program operation, the SRD register indicates whether or not the auto program operation has been completed as expected.



**Figure 24.9 Program Command**

### 24.3.5.5 Block Erase Command

The block erase command erases each block.

Auto erase operation (erase and verify) will start in the specified block by writing command code "xx20<sub>16</sub>" in the first bus cycle and "xxD0<sub>16</sub>" to the highest-order even address of a block in the second bus cycle.

The FMR00 bit in the FMR0 register indicates whether or not an auto erase operation has been completed. The FMR00 bit is set to "0" (busy) during auto erase and to "1" (ready) when the auto erase operation is completed.

After the completion of an auto erase operation, the FMR07 bit in the FMR0 register indicates whether or not the auto erase operation has been completed as expected. (Refer to **24.3.8 Full Status Check**.)

Figure 24.10 shows a flow chart of the block erase command programming.

The lock bit can protect each block from being programmed inadvertently. (Refer to **24.3.6 Data Protect Function**.)

In EW mode 1, do not execute this command on the block where the rewrite control program is allocated. In EW mode 0, the microcomputer enters read status register mode as soon as an auto erase operation starts. The SRD register can be read. The SR7 bit in the SRD register is set to "0" at the same time an auto erase operation starts. It is set to "1" when an auto erase operation is completed. The microcomputer remains in read status register mode until the read array command or read lock bit status command is written.

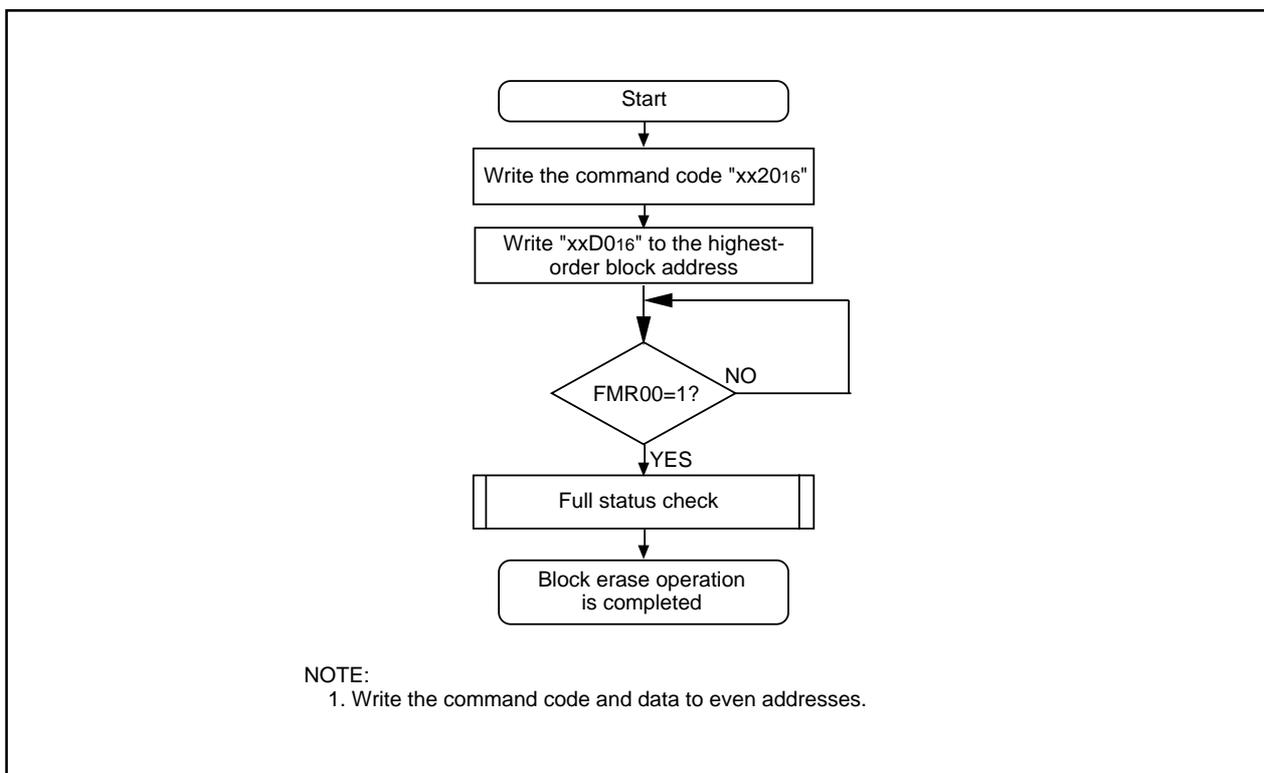


Figure 24.10 Block Erase Command

#### 24.3.5.6 Erase All Unlocked Block Command

The erase all unlocked block command erases all blocks except the block A.

By writing command code "xA716" in the first bus cycle and "xD016" in the second bus cycle, auto erase (erase and verify) operation will run continuously in all blocks except the block A.

The FMR00 bit in the FMR0 register indicates whether or not an auto erase operation has been completed.

After the completion of an auto erase operation, the FMR07 bit in the FMR0 register indicates whether or not the auto erase operation has been completed as expected.

The lock bit can protect each block from being programmed inadvertently. (Refer to **24.3.6 Data Protect Function**.)

In EW mode 1, do not execute this command when the lock bit for any block storing the rewrite control program is set to "1" (unlocked) or when the FMR02 bit in the FMR0 register is set to "1" (lock bit disabled).

In EW mode 0, the microcomputer enters read status register mode as soon as an auto erase operation starts. The SRD register can be read. The SR7 bit in the SRD register is set to "0" (busy) at the same time an auto erase operation starts. It is set to "1" (ready) when an auto erase operation is completed. The microcomputer remains in read status register mode until the read array command or read lock bit status command is written.

Only blocks 0 to 12 can be erased by the erase all unlocked block command. The block A cannot be erased. Use the block erase command to erase the block A.

### 24.3.5.7 Lock Bit Program Command

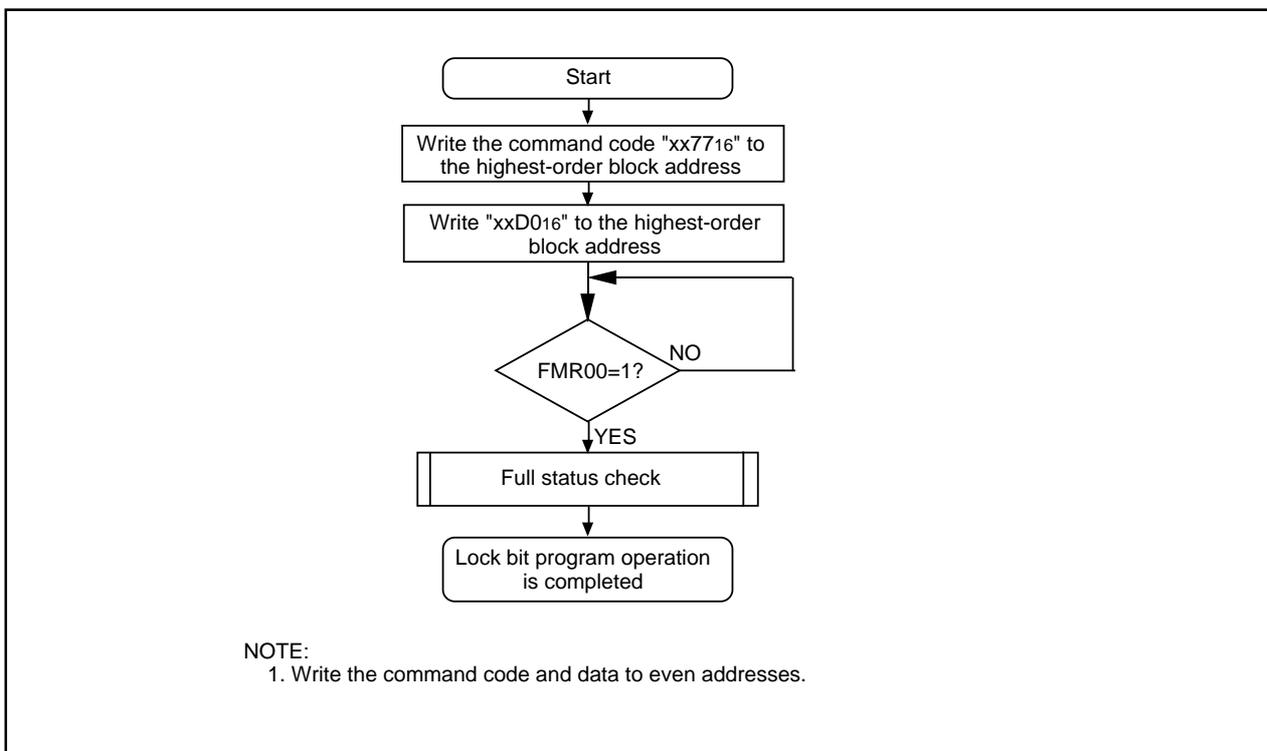
The lock bit program command sets the lock bit for a specified block to "0" (locked).

By writing command code "xx7716" in the first bus cycle and "xxD016" to the highest-order even address of a block in the second bus cycle, the lock bit for the specified block is set to "0". The address value specified in the first bus cycle must be the same highest-order even address of a block specified in the second bus cycle.

Figure 24.11 shows a flow chart of the lock bit program command programming. Execute read lock bit status command to read lock bit state (lock bit data).

The FMR00 bit in the FMR0 register indicates whether a lock bit program operation is completed.

Refer to **24.3.6 Data Protect Function** for details on lock bit functions and how to set it to "1" (unlocked).



**Figure 24.11 Lock Bit Program Command**

### 24.3.5.8 Read Lock Bit Status Command

The read lock bit status command reads the lock bit state (the lock bit data) of a specified block. By writing command code "xx7116" in the first bus cycle and "xxD016" to the highest-order even address of a block in the second bus cycle, the FMR16 bit in the FMR1 register stores information on whether or not the lock bit of a specified block is locked. Read the FMR16 bit after the FMR00 bit in the FMR0 register is set to "1" (ready).

Figure 24.12 shows a flow chart of the read lock bit status command programming.

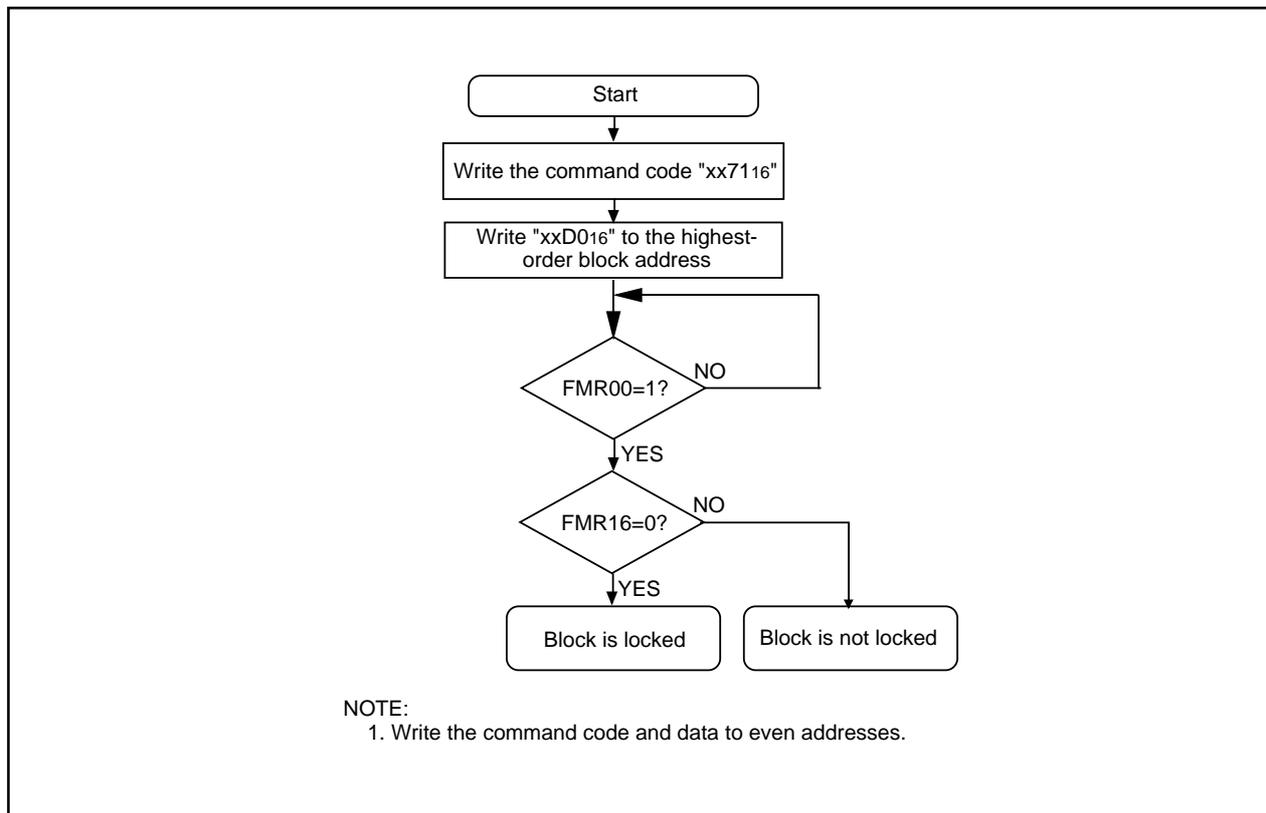


Figure 24.12 Read Lock Bit Status Command

### 24.3.6 Data Protect Function

Each block in the flash memory has a nonvolatile lock bit. The lock bit is enabled by setting the FMR02 bit to "0" (lock bit enabled). The lock bit individually protects (locks) each block against program and erase. This prevents data from being inadvertently written to or erased from the flash memory.

- When the lock bit status is set to "0", the block is locked (block is protected against program and erase).
- When the lock bit status is set to "1", the block is not locked (block can be programmed or erased).

The lock bit status is set to "0" (locked) by executing the lock bit program command and to "1" (unlocked) by erasing the block. The lock bit status cannot be set to "1" by any commands.

The lock bit status can be read by the read lock bit status command.

The lock bit function is disabled by setting the FMR02 bit to "1". All blocks are unlocked. However, individual lock bit status remains unchanged. The lock bit function is enabled by setting the FMR02 bit to "0". Lock bit status is retained.

If the block erase or erase all unlocked block command is executed while the FMR02 bit is set to "1", the target block or all blocks are erased regardless of lock bit status. The lock bit status of each block are set to "1" after an erase operation is completed.

Refer to **24.3.5 Software Commands** for details on each command.

### 24.3.7 Status Register (SRD Register)

The SRD register indicates the flash memory operating state and whether or not an erase or program operation is completed as expected. The FMR00, FMR06 and FMR07 bits in the FMR0 register indicate SRD register states.

Table 24.5 shows the SRD register.

In EW mode 0, the SRD register can be read when the followings occur.

- Any even address in the user ROM area is read after writing the read status register command
- Any even address in the user ROM area is read from when the program, block erase, erase all unlocked block, or lock bit program command is executed until when the read array command is executed.

#### 24.3.7.1 Sequencer Status (SR7 and FMR00 Bits )

The sequencer status indicates the flash memory operating state. It is set to "0" while the program, block erase, erase all unlocked block, lock bit program, or read lock bit status command is being executed; otherwise, it is set to "1".

#### 24.3.7.2 Erase Status (SR5 and FMR07 Bits)

Refer to **24.3.8 Full Status Check**.

#### 24.3.7.3 Program Status (SR4 and FMR06 Bits)

Refer to **24.3.8 Full Status Check**.

**Table 24.5 Status Register**

Bits in SRD register	Bits in FMR0 Register	Status Name	Definition		Value after Reset
			"0"	"1"	
SR7 (D7)	FMR00	Sequencer status	BUSY	READY	1
SR6 (D6)	—	Reserved bit	-	-	-
SR5 (D5)	FMR07 <sup>(1)</sup>	Erase status	Successfully completed	Error	0
SR4 (D4)	FMR06 <sup>(1)</sup>	Program status	Successfully completed	Error	0
SR3 (D3)	—	Reserved bit	-	-	-
SR2 (D2)	—	Reserved bit	-	-	-
SR1 (D1)	—	Reserved bit	-	-	-
SR0 (D0)	—	Reserved bit	-	-	-

D0 to D7: These data buses are read when the read status register command is executed.

**NOTE:**

1. The FMR07 (SR5) and FMR06 (SR4) bits are set to "0" by executing the clear status register command. When the FMR07 (SR5) or FMR06 (SR4) bit is set to "1", the program, block erase, erase all unlocked block and lock bit program commands are not accepted.

### 24.3.8 Full Status Check

If an error occurs when a program or erase operation is completed, the FMR07 and FMR06 bits in the FMR0 register are set to "1", indicating a specific error. Therefore, execution results can be confirmed by verifying these bits (full status check).

Table 24.6 lists errors and FMR0 register state. Figure 24.13 shows a flow chart of the full status check and handling procedure for each error.

**Table 24.6 Errors and FMR0 Register State**

FMR0 Register (SRD Register) State		Error	Error Occurrence Conditions
FMR07 (SR5)	FMR06 (SR4)		
1	1	Command sequence error	<ul style="list-style-type: none"> <li>• An incorrect command is written</li> <li>• A value other than "xxD016" or "xxFF16" is written in the second bus cycle of the lock bit program, block erase or erase all unlocked block command<sup>(1)</sup></li> </ul>
1	0	Erase error	<ul style="list-style-type: none"> <li>• The block erase command is executed on a locked block<sup>(2)</sup></li> <li>• The block erase or erase all unlocked block command is executed on an unlock block, but the erase operation is not successfully completed</li> </ul>
0	1	Program error	<ul style="list-style-type: none"> <li>• The program command is executed on locked blocks<sup>(2)</sup></li> <li>• The program command is executed on an unlocked block, but the program operation is not completed as expected</li> <li>• The lock bit program command is executed but the program operation is not successfully completed</li> </ul>

**NOTES:**

1. The flash memory enters read array mode when command code "xxFF16" is written in the second bus cycle of these commands. The command code written in the first bus cycle is ignored.
2. When the FMR02 bit is set to "1" (lock bit disabled), no error occurs even under the conditions above.

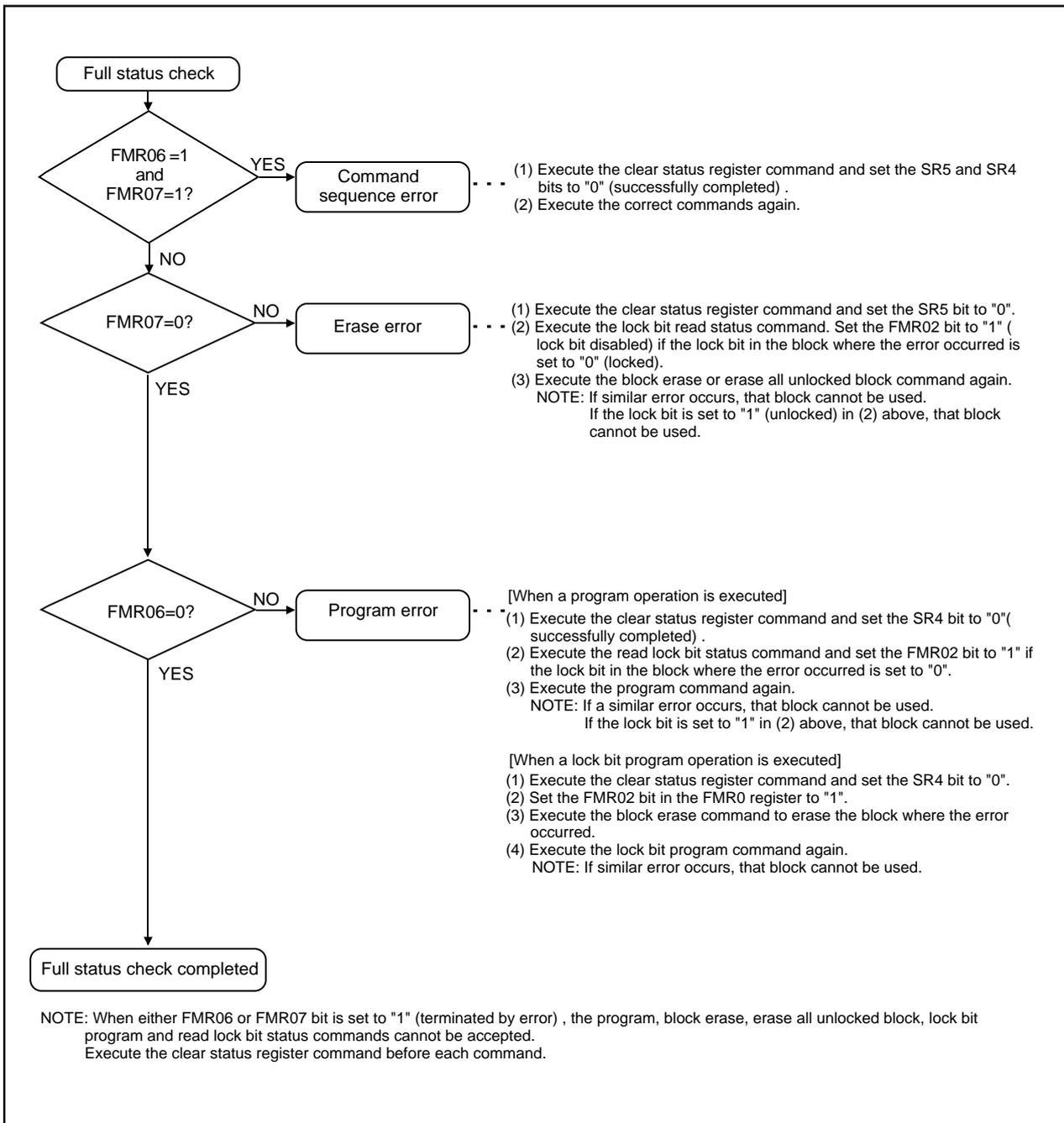


Figure 24.13 Full Status Check and Handling Procedure for Each Error

## 24.4 Standard Serial I/O Mode

In standard serial I/O mode, the serial programmer supporting the M32C/88 Group (M32C/88T) can be used to rewrite the flash memory user ROM area, while the microcomputer is mounted on a board. For more information about the serial programmer, contact your serial programmer manufacturer. Refer to the user's manual included with your serial programmer for instructions.

Table 24.7 lists pin descriptions (flash memory standard serial I/O mode). Figures 24.14 to 24.16 show pin connections in serial I/O mode.

### 24.4.1 ID Code Verify Function

The ID code verify function determines whether or not the ID codes sent from the serial programmer matches those written in the flash memory. (Refer to **24.2 Functions to Prevent Rewriting of Flash Memory.**)

**Table 24.7 Pin Description (Flash Memory Standard Serial I/O Mode)**

Symbol	Function	I/O Type	Description
VCC	Power supply	I	Apply the guaranteed program/erase supply voltage to the VCC pin.
VSS	input		Apply 0 V to the VSS pin
CNVSS	CNVSS	I	Connect this pin to VCC
RESET	Reset input	I	Reset input pin. Apply 20 or more clock cycles to the XIN pin while "L" is applied to the RESET pin
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT
XOUT	Clock output	O	To use the external clock, input the clock from XIN and leave XOUT open
BYTE	BYTE input	I	Connect this pin to VSS or VCC
AVCC	Analog power	I	Connect AVCC to VCC
AVSS	supply input		Connect AVSS to VSS
VREF	Reference voltage input	I	Reference voltage input pin for the A/D converter
P00 to P07	Input port P0	I	Apply "H" or "L" to this pin, or leave open
P10 to P17	Input port P1	I	Apply "H" or "L" to this pin, or leave open
P20 to P27	Input port P2	I	Apply "H" or "L" to this pin, or leave open
P30 to P37	Input port P3	I	Apply "H" or "L" to this pin, or leave open
P40 to P47	Input port P4	I	Apply "H" or "L" to this pin, or leave open
P50	$\overline{CE}$ input	I	Apply "H" to this pin
P55	EPM input	I	Apply "L" to this pin
P51 to P54 P56, P57	Input port P5	I	Apply "H" or "L" to this pin, or leave open
P60 to P63	Input port P6	I	Apply "H" or "L" to this pin, or leave open
P64	BUSY output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Program running verify monitor Standard serial I/O mode 3: Leave open
P65	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2, 3: Apply "L" to this pin
P66	RxD Data input	I	Standard serial I/O mode 1, 2: Serial data input pin Standard serial I/O mode 3: Apply "H" to this pin
P67	TxD Data output	O	Standard serial I/O mode 1, 2: Serial data output pin Standard serial I/O mode 3: Leave open
P70 to P75	Input port P7	I	Apply "H" or "L" to this pin, or leave open
P76	CAN output	O	Standard serial I/O mode 1, 2: Apply "H" or "L" to this pin, or leave open Standard serial I/O mode 3: CAN output pin
P77	CAN input	I	Standard serial I/O mode 1, 2: Apply "H" or "L" to this pin, or leave open Standard serial I/O mode 3: CAN input pin
P80 to P84 P86, P87	Input port P8	I	Apply "H" or "L" to this pin, or leave open
P85	$\overline{NMI}$ input	I	Connect this pin to VCC
P90 to P97	Input port P9	I	Apply "H" or "L" to this pin, or leave open
P100 to P107	Input port P10	I	Apply "H" or "L" to this pin, or leave open
P110 to P114	Input port P11	I	Apply "H" or "L" to this pin, or leave open <sup>(1)</sup>
P120 to P127	Input port P12	I	Apply "H" or "L" to this pin, or leave open <sup>(1)</sup>
P130 to P137	Input port P13	I	Apply "H" or "L" to this pin, or leave open <sup>(1)</sup>
P140 to P146	Input port P14	I	Apply "H" or "L" to this pin, or leave open <sup>(1)</sup>
P150 to P157	Input port P15	I	Apply "H" or "L" to this pin, or leave open <sup>(1)</sup>

## NOTE:

1. These pins are provided in the 144-pin package only.

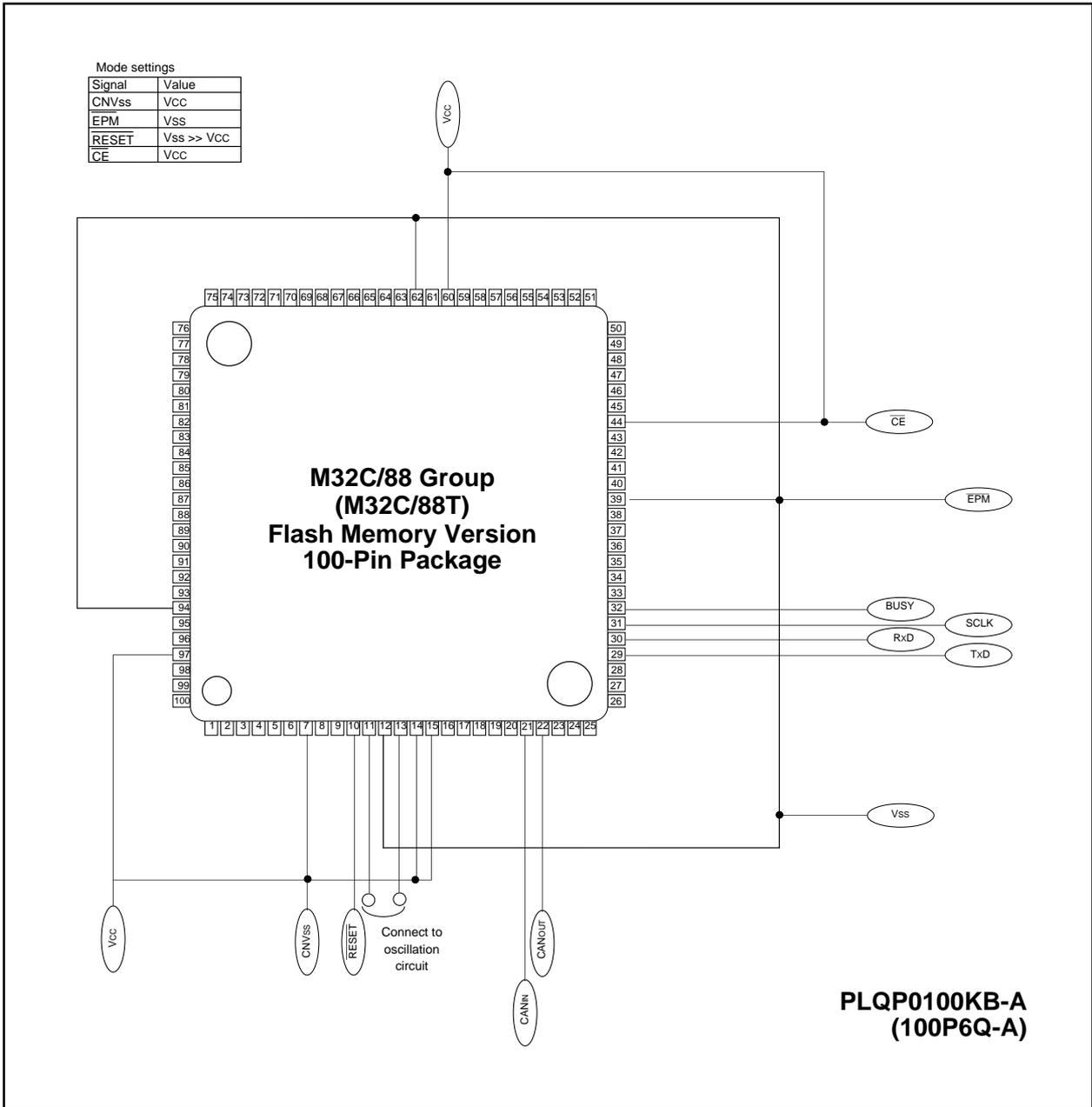


Figure 24.14 Pin Connections in Standard Serial I/O Mode (1)

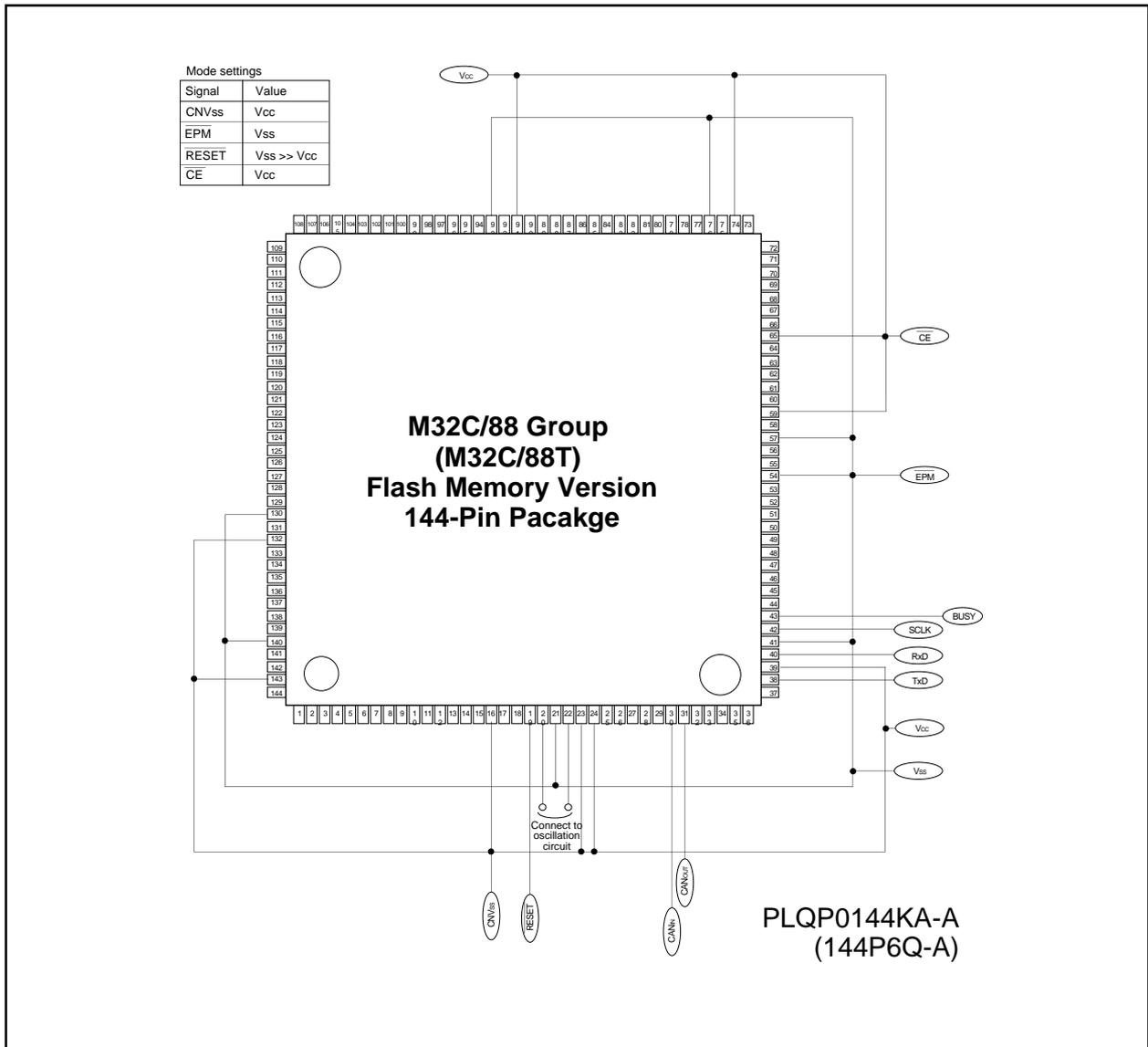
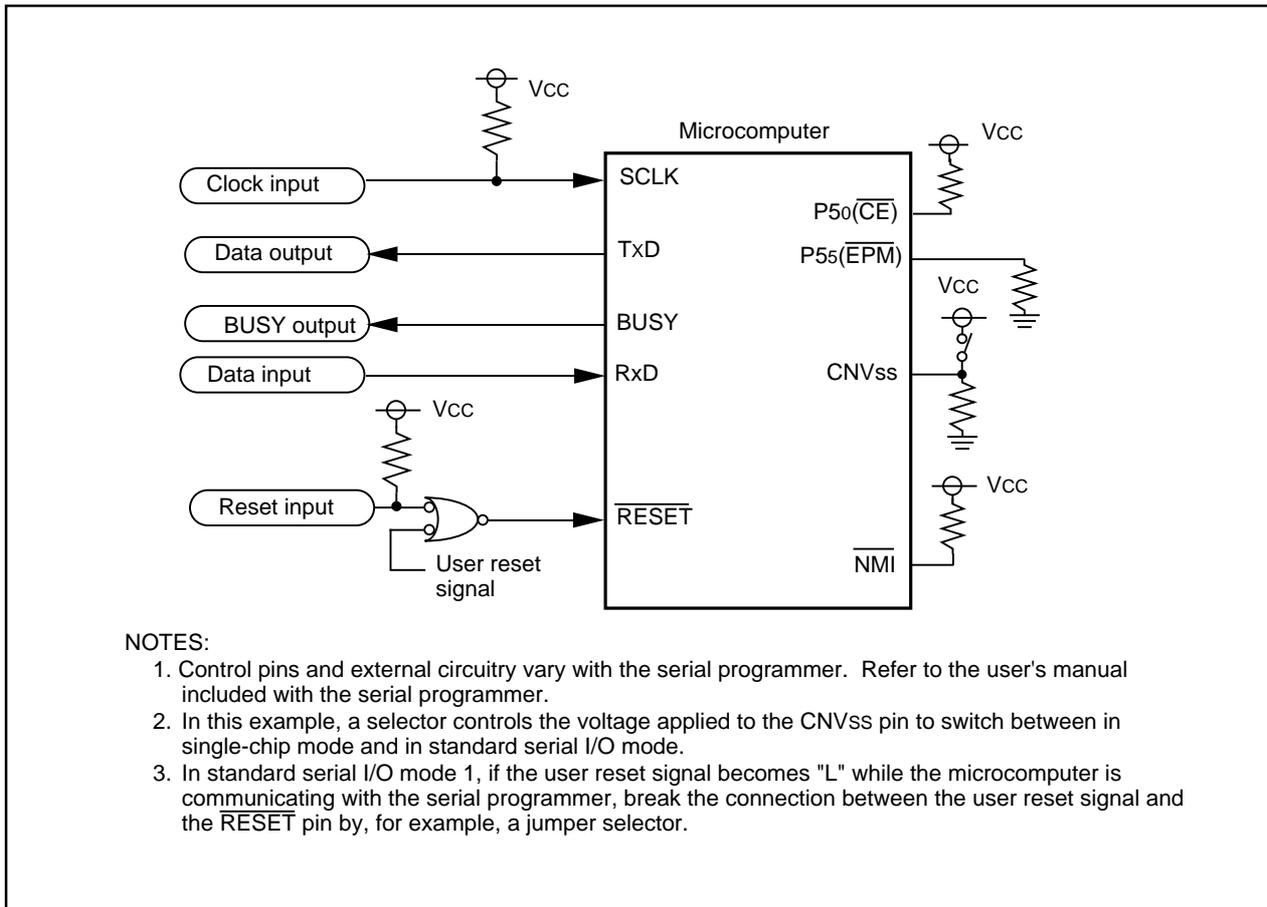


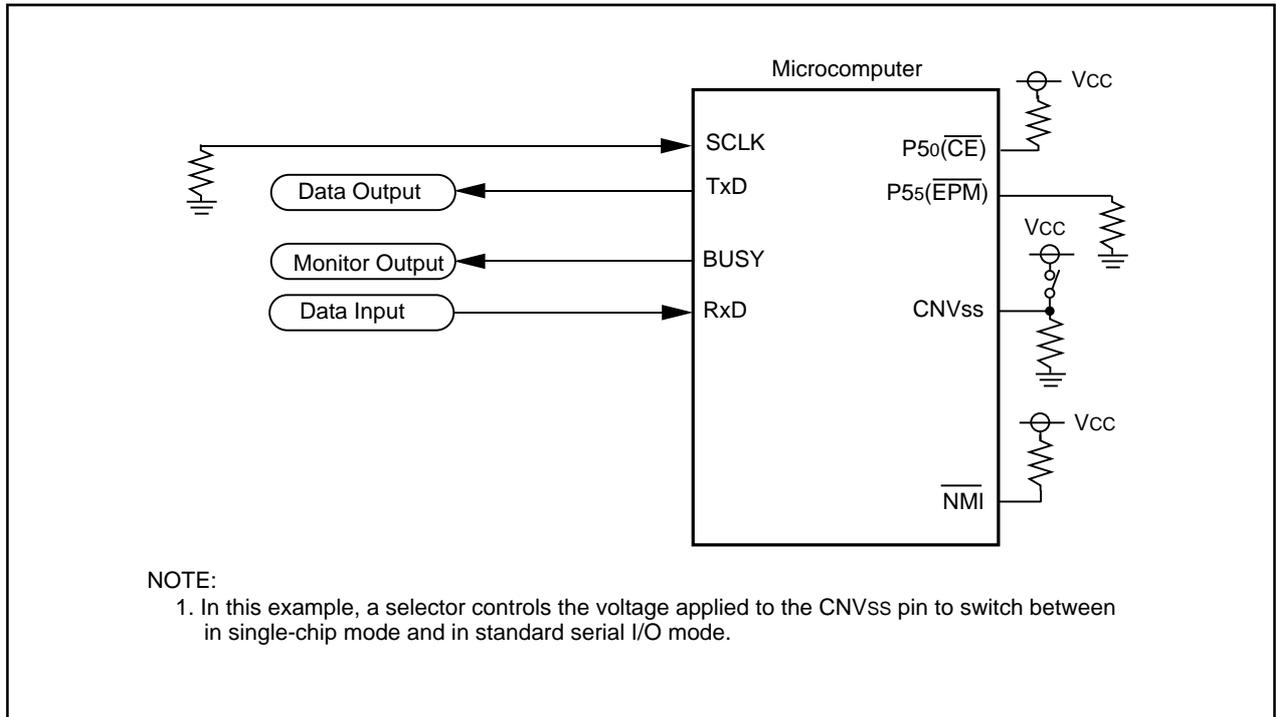
Figure 24.15 Pin Connections in Standard Serial I/O Mode (2)

### 24.4.2 Circuit Application in Standard Serial I/O Mode

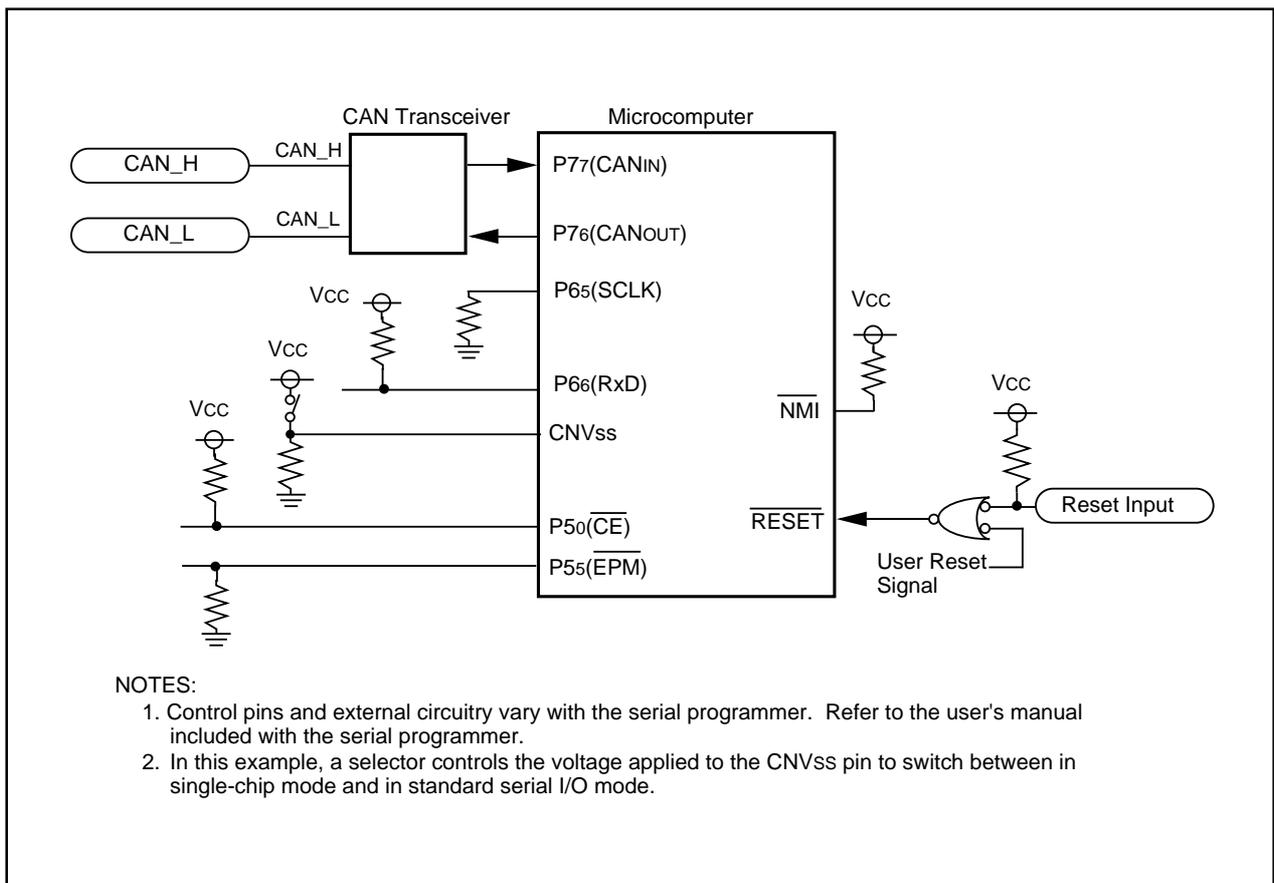
Figure 24.16 shows an example of a circuit application in standard serial I/O mode 1. Figure 24.17 shows an example of a circuit application serial I/O mode 2. Figure 24.18 shows an example of a circuit application serial I/O mode 3. Refer to the user's manual of your serial programmer to handle pins controlled by the serial programmer.



**Figure 24.16 Circuit Application in Standard Serial I/O Mode 1**



**Figure 24.17 Circuit Application in Standard Serial I/O Mode 2**



**Figure 24.18 Circuit Application in Standard Serial I/O Mode 3**

## 24.5 Parallel I/O Mode

In parallel I/O mode, the user ROM area and the boot ROM area can be rewritten by a parallel programmer supporting the M32C/88 Group (M32C/88T). Contact your parallel programmer manufacturer for more information on the parallel programmer. Refer to the user's manual included with your parallel programmer for instructions.

### 24.5.1 Boot ROM Area

An erase block operation in the boot ROM area is applied to only one 4-Kbyte block. The rewrite control program in standard serial I/O mode is written in the boot ROM area before shipment. Do not rewrite the boot ROM area if using the serial programmer.

In parallel I/O mode, the boot ROM area is located in addresses FFF000<sub>16</sub> to FFFFFFF<sub>16</sub>. Rewrite this address range only if rewriting the boot ROM area. (Do not access addresses other than addresses FFF000<sub>16</sub> to FFFFFFF<sub>16</sub>.)

### 24.5.2 ROM Code Protect Function

The ROM code protect function prevents the flash memory from being read and rewritten in parallel I/O mode. (Refer to **24.2 Functions to Prevent Rewriting of Flash Memory.**)

## 25. Electrical Characteristics

**Table 25.1 Absolute Maximum Ratings**

Symbol	Parameter		Condition	Value	Unit	
V <sub>cc</sub>	Supply Voltage		V <sub>cc</sub> =AV <sub>cc</sub>	-0.3 to 6.0	V	
AV <sub>cc</sub>	Analog Supply Voltage		V <sub>cc</sub> =AV <sub>cc</sub>	-0.3 to 6.0	V	
V <sub>i</sub>	Input Voltage	RESET, CNV <sub>ss</sub> , BYTE, P0 <sub>0</sub> -P0 <sub>7</sub> , P1 <sub>0</sub> -P1 <sub>7</sub> , P2 <sub>0</sub> -P2 <sub>7</sub> , P3 <sub>0</sub> -P3 <sub>7</sub> , P4 <sub>0</sub> -P4 <sub>7</sub> , P5 <sub>0</sub> -P5 <sub>7</sub> , P6 <sub>0</sub> -P6 <sub>7</sub> , P7 <sub>2</sub> -P7 <sub>7</sub> , P8 <sub>0</sub> -P8 <sub>7</sub> , P9 <sub>0</sub> -P9 <sub>7</sub> , P10 <sub>0</sub> -P10 <sub>7</sub> , P11 <sub>0</sub> -P11 <sub>4</sub> , P12 <sub>0</sub> -P12 <sub>7</sub> , P13 <sub>0</sub> -P13 <sub>7</sub> , P14 <sub>0</sub> -P14 <sub>6</sub> , P15 <sub>0</sub> -P15 <sub>7</sub> <sup>(1)</sup> , V <sub>REF</sub> , X <sub>IN</sub>		-0.3 to V <sub>cc</sub> +0.3	V	
		P7 <sub>0</sub> , P7 <sub>1</sub>		-0.3 to 6.0		
V <sub>o</sub>	Output Voltage	P0 <sub>0</sub> -P0 <sub>7</sub> , P1 <sub>0</sub> -P1 <sub>7</sub> , P2 <sub>0</sub> -P2 <sub>7</sub> , P3 <sub>0</sub> -P3 <sub>7</sub> , P4 <sub>0</sub> -P4 <sub>7</sub> , P5 <sub>0</sub> -P5 <sub>7</sub> , P6 <sub>0</sub> -P6 <sub>7</sub> , P7 <sub>2</sub> -P7 <sub>7</sub> , P8 <sub>0</sub> -P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> -P9 <sub>7</sub> , P10 <sub>0</sub> -P10 <sub>7</sub> , P11 <sub>0</sub> -P11 <sub>4</sub> , P12 <sub>0</sub> -P12 <sub>7</sub> , P13 <sub>0</sub> -P13 <sub>7</sub> , P14 <sub>0</sub> -P14 <sub>6</sub> , P15 <sub>0</sub> -P15 <sub>7</sub> <sup>(1)</sup> , X <sub>OUT</sub>		-0.3 to V <sub>cc</sub> +0.3	V	
		P7 <sub>0</sub> , P7 <sub>1</sub>		-0.3 to 6.0		
P <sub>d</sub>	Power Dissipation		T version	T <sub>opr</sub> =25° C	500	mW
			U version		400	
T <sub>opr</sub>	Operating Ambient Temperature	during CPU operation	T version	-40 to 85	° C	
			U version	-40 to 105		
		during flash memory program and erase operation		0 to 60		
T <sub>stg</sub>	Storage Temperature			-65 to 150	° C	

NOTE:

1. P11 to P15 are provided in the 144-pin package only.

**Table 25.2 Recommended Operating Conditions**  
**(V<sub>CC</sub>=4.2 to 5.5V, V<sub>SS</sub>=0V at T<sub>opr</sub> = -40 to 85°C (T version)/-40 to 105°C (U version)**  
**unless otherwise specified)**

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V <sub>CC</sub>	Supply Voltage		4.2	5.0	5.5	V
AV <sub>CC</sub>	Analog Supply Voltage			V <sub>CC</sub>		V
V <sub>SS</sub>	Supply Voltage			0		V
AV <sub>SS</sub>	Analog Supply Voltage			0		V
V <sub>IH</sub>	Input High ("H") Voltage	P0 <sub>0</sub> -P0 <sub>7</sub> , P1 <sub>0</sub> -P1 <sub>7</sub> , P2 <sub>0</sub> -P2 <sub>7</sub> , P3 <sub>0</sub> -P3 <sub>7</sub> , P4 <sub>0</sub> -P4 <sub>7</sub> , P5 <sub>0</sub> -P5 <sub>7</sub> , P6 <sub>0</sub> -P6 <sub>7</sub> , P7 <sub>2</sub> -P7 <sub>7</sub> , P8 <sub>0</sub> -P8 <sub>7</sub> <sup>(3)</sup> , P9 <sub>0</sub> -P9 <sub>7</sub> , P10 <sub>0</sub> -P10 <sub>7</sub> , P11 <sub>0</sub> -P11 <sub>4</sub> , P12 <sub>0</sub> -P12 <sub>7</sub> , P13 <sub>0</sub> -P13 <sub>7</sub> <sup>(4)</sup> , P14 <sub>0</sub> -P14 <sub>6</sub> , P15 <sub>0</sub> -P15 <sub>7</sub> <sup>(4)</sup> , X <sub>IN</sub> , $\overline{\text{RESET}}$ , CNV <sub>SS</sub> , BYTE P7 <sub>0</sub> , P7 <sub>1</sub>	0.8V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IL</sub>	Input Low ("L") Voltage	P0 <sub>0</sub> -P0 <sub>7</sub> , P1 <sub>0</sub> -P1 <sub>7</sub> , P2 <sub>0</sub> -P2 <sub>7</sub> , P3 <sub>0</sub> -P3 <sub>7</sub> , P4 <sub>0</sub> -P4 <sub>7</sub> , P5 <sub>0</sub> -P5 <sub>7</sub> , P6 <sub>0</sub> -P6 <sub>7</sub> , P7 <sub>2</sub> -P7 <sub>7</sub> , P8 <sub>0</sub> -P8 <sub>7</sub> <sup>(3)</sup> , P9 <sub>0</sub> -P9 <sub>7</sub> , P10 <sub>0</sub> -P10 <sub>7</sub> , P11 <sub>0</sub> -P11 <sub>4</sub> , P12 <sub>0</sub> -P12 <sub>7</sub> , P13 <sub>0</sub> -P13 <sub>7</sub> <sup>(4)</sup> , P14 <sub>0</sub> -P14 <sub>6</sub> , P15 <sub>0</sub> -P15 <sub>7</sub> <sup>(4)</sup> , X <sub>IN</sub> , $\overline{\text{RESET}}$ , CNV <sub>SS</sub> , BYTE	0		0.2V <sub>CC</sub>	V
I <sub>OH(peak)</sub>	Peak Output High ("H") Current <sup>(2)</sup>	P0 <sub>0</sub> -P0 <sub>7</sub> , P1 <sub>0</sub> -P1 <sub>7</sub> , P2 <sub>0</sub> -P2 <sub>7</sub> , P3 <sub>0</sub> -P3 <sub>7</sub> , P4 <sub>0</sub> -P4 <sub>7</sub> , P5 <sub>0</sub> -P5 <sub>7</sub> , P6 <sub>0</sub> -P6 <sub>7</sub> , P7 <sub>2</sub> -P7 <sub>7</sub> , P8 <sub>0</sub> -P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> -P9 <sub>7</sub> , P10 <sub>0</sub> -P10 <sub>7</sub> , P11 <sub>0</sub> -P11 <sub>4</sub> , P12 <sub>0</sub> -P12 <sub>7</sub> , P13 <sub>0</sub> -P13 <sub>7</sub> , P14 <sub>0</sub> -P14 <sub>6</sub> , P15 <sub>0</sub> -P15 <sub>7</sub> <sup>(4)</sup>			-10.0	mA
I <sub>OH(avg)</sub>	Average Output High ("H") Current <sup>(1)</sup>	P0 <sub>0</sub> -P0 <sub>7</sub> , P1 <sub>0</sub> -P1 <sub>7</sub> , P2 <sub>0</sub> -P2 <sub>7</sub> , P3 <sub>0</sub> -P3 <sub>7</sub> , P4 <sub>0</sub> -P4 <sub>7</sub> , P5 <sub>0</sub> -P5 <sub>7</sub> , P6 <sub>0</sub> -P6 <sub>7</sub> , P7 <sub>2</sub> -P7 <sub>7</sub> , P8 <sub>0</sub> -P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> -P9 <sub>7</sub> , P10 <sub>0</sub> -P10 <sub>7</sub> , P11 <sub>0</sub> -P11 <sub>4</sub> , P12 <sub>0</sub> -P12 <sub>7</sub> , P13 <sub>0</sub> -P13 <sub>7</sub> , P14 <sub>0</sub> -P14 <sub>6</sub> , P15 <sub>0</sub> -P15 <sub>7</sub> <sup>(4)</sup>			-5.0	mA
I <sub>OL(peak)</sub>	Peak Output Low ("L") Current <sup>(2)</sup>	P0 <sub>0</sub> -P0 <sub>7</sub> , P1 <sub>0</sub> -P1 <sub>7</sub> , P2 <sub>0</sub> -P2 <sub>7</sub> , P3 <sub>0</sub> -P3 <sub>7</sub> , P4 <sub>0</sub> -P4 <sub>7</sub> , P5 <sub>0</sub> -P5 <sub>7</sub> , P6 <sub>0</sub> -P6 <sub>7</sub> , P7 <sub>0</sub> -P7 <sub>7</sub> , P8 <sub>0</sub> -P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> -P9 <sub>7</sub> , P10 <sub>0</sub> -P10 <sub>7</sub> , P11 <sub>0</sub> -P11 <sub>4</sub> , P12 <sub>0</sub> -P12 <sub>7</sub> , P13 <sub>0</sub> -P13 <sub>7</sub> , P14 <sub>0</sub> -P14 <sub>6</sub> , P15 <sub>0</sub> -P15 <sub>7</sub> <sup>(4)</sup>			10.0	mA
I <sub>OL(avg)</sub>	Average Output Low ("L") Current <sup>(1)</sup>	P0 <sub>0</sub> -P0 <sub>7</sub> , P1 <sub>0</sub> -P1 <sub>7</sub> , P2 <sub>0</sub> -P2 <sub>7</sub> , P3 <sub>0</sub> -P3 <sub>7</sub> , P4 <sub>0</sub> -P4 <sub>7</sub> , P5 <sub>0</sub> -P5 <sub>7</sub> , P6 <sub>0</sub> -P6 <sub>7</sub> , P7 <sub>0</sub> -P7 <sub>7</sub> , P8 <sub>0</sub> -P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> -P9 <sub>7</sub> , P10 <sub>0</sub> -P10 <sub>7</sub> , P11 <sub>0</sub> -P11 <sub>4</sub> , P12 <sub>0</sub> -P12 <sub>7</sub> , P13 <sub>0</sub> -P13 <sub>7</sub> , P14 <sub>0</sub> -P14 <sub>6</sub> , P15 <sub>0</sub> -P15 <sub>7</sub> <sup>(4)</sup>			5.0	mA

## NOTES:

- Typical values when average output current is 100 ms.
- Total I<sub>OL(peak)</sub> for P0, P1, P2, P8<sub>6</sub>, P8<sub>7</sub>, P9, P10, P11, P14 and P15 must be 80 mA or less.  
 Total I<sub>OL(peak)</sub> for P3, P4, P5, P6, P7, P8<sub>0</sub> to P8<sub>4</sub>, P12 and P13 must be 80 mA or less.  
 Total I<sub>OH(peak)</sub> for P0, P1, P2, and P11 must be -40mA or less.  
 Total I<sub>OH(peak)</sub> for P8<sub>6</sub>, P8<sub>7</sub>, P9, P10, P14 and P15 must be -40 mA or less.  
 Total I<sub>OH(peak)</sub> for P3, P4, P5, P12 and P13 must be -40 mA or less.  
 Total I<sub>OH(peak)</sub> for P6, P7, and P8<sub>0</sub> to P8<sub>4</sub> must be -40 mA or less.
- V<sub>IH</sub> and V<sub>IL</sub> reference for P8<sub>7</sub> applies when P8<sub>7</sub> is used as a programmable input port.  
 It does not apply when P8<sub>7</sub> is used as X<sub>CIN</sub>.
- Ports P11 to P15 are provided in the 144-pin package only.

**Table 25.3 Recommended Operating Conditions (Continued)**  
**(V<sub>CC</sub>=4.2 to 5.5V, V<sub>SS</sub>=0V at T<sub>opr</sub> = -40 to 85°C (T version)/-40 to 105°C (U version)**  
**unless otherwise specified)**

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
f(BCLK)	CPU Operation Frequency	V <sub>CC</sub> =4.2 to 5.5 V	0		32	MHz
f(XIN)	Main Clock Input Frequency	V <sub>CC</sub> =4.2 to 5.5 V	0		24	MHz
f(XCIN)	Sub Clock Frequency			32.768	50	kHz
f(Ring)	On-chip Oscillator Frequency (V <sub>CC</sub> =5.0V, T <sub>opr</sub> =25° C)		0.5	1	2	MHz
f(PLL)	PLL Clock Frequency	V <sub>CC</sub> =4.2 to 5.5 V	10		32	MHz
tsu(PLL)	Wait Time to Stabilize PLL Frequency Synthesizer	V <sub>CC</sub> =5.0 V			5	ms

V<sub>CC</sub>=5V

**Table 25.4 Electrical Characteristics**  
**(V<sub>CC</sub>=4.2 to 5.5V, V<sub>SS</sub>=0V at T<sub>opr</sub> = -40 to 85°C (T version)/-40 to 105°C (U version),**  
**f(BCLK)=32MHz unless otherwise specified)**

Symbol	Parameter		Condition	Standard			Unit
				Min.	Typ.	Max.	
V <sub>OH</sub>	Output High ("H") Voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 <sup>(1)</sup>	I <sub>OH</sub> =-5 mA	V <sub>CC</sub> -2.0		V <sub>CC</sub>	V
		P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 <sup>(1)</sup>	I <sub>OH</sub> =-200 μA	V <sub>CC</sub> -0.3		V <sub>CC</sub>	V
		X <sub>OUT</sub>	I <sub>OH</sub> =-1 mA	3.0			V
		X <sub>COUT</sub>	High Power	No load applied		2.5	
		Low Power	No load applied		1.6		V
V <sub>OL</sub>	Output Low ("L") Voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 <sup>(1)</sup>	I <sub>OL</sub> =5mA			2.0	V
		P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 <sup>(1)</sup>	I <sub>OL</sub> =200 μA			0.45	V
		X <sub>OUT</sub>	I <sub>OL</sub> =1 mA			2.0	V
		X <sub>COUT</sub>	High Power	No load applied		0	
		Low Power	No load applied		0		V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	HOLD, RDY, TA0IN-TA4IN, TB0IN-TB5IN, INT0-INT5, ADTRG, CTS0-CTS4, CLK0-CLK4, TA0OUT-TA4OUT, NMI, KI0-KI3, RxD0-RxD4, SCL0-SCL4, SDA0-SDA4		0.2		1.0	V
		RESET		0.2		1.8	V
I <sub>IH</sub>	Input High ("H") Current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 <sup>(1)</sup> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	V <sub>I</sub> =5 V			5.0	μA
I <sub>IL</sub>	Input Low ("L") Current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 <sup>(1)</sup> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	V <sub>I</sub> =0 V			-5.0	μA
R <sub>PULLUP</sub>	Pull-up Resistance	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 <sup>(1)</sup>	V <sub>I</sub> =0 V	30	50	167	kΩ
R <sub>fXIN</sub>	Feedback Resistance	X <sub>IN</sub>			1.5		MΩ
R <sub>fXCIN</sub>	Feedback Resistance	X <sub>CIN</sub>			10		MΩ
V <sub>RAM</sub>	RAM Standby Voltage	In stop mode		2.0			V

NOTE:

1. Ports P11 to P15 are provided in the 144-pin package only.

VCC=5V

**Table 25.4 Electrical Characteristics (Continued)**  
**(VCC=4.2 to 5.5V, VSS=0V at Topr = -40 to 85°C (T version)/-40 to 105°C (U version),**  
**f(BCLK)=32MHz unless otherwise specified)**

Symbol	Parameter	Measurement Condition	Standard			Unit	
			Min.	Typ.	Max.		
Icc	Power Supply Current	In single-chip mode, output pins are left open and other pins are connected to Vss.	f(BCLK)=32 MHz, Square wave, No division		28	50	mA
			f(BCLK)=32 kHz, In low-power consumption mode, Program running on ROM		430		μA
			f(BCLK)=32 kHz, In low-power consumption mode, Program running on RAM <sup>(1)</sup>		25		μA
			f(BCLK)=32 kHz, In wait mode, Topr=25° C		10		μA
			While clock stops, Topr=25° C		0.8	5	μA
			While clock stops, Topr=85° C			50	μA
			While clock stops, Topr=105° C			100	μA
While clock stops, Topr=125° C			200	μA			

NOTE:

1. Value is obtained when setting the FMSTP bit in the FMR0 register to "1" (flash memory stopped).

VCC=5V

**Table 25.5 A/D Conversion Characteristics**

(VCC=4.2 to 5.5V, VSS=0V at Topr = -40 to 85°C (T version)/-40 to 105°C (U version),  
f(BCLK)=32MHz unless otherwise specified)

Symbol	Parameter	Measurement Condition		Standard			Unit
				Min.	Typ.	Max.	
-	Resolution	VREF=VCC				10	Bits
INL	Integral Nonlinearity Error	VREF=VCC=5V	AN0 to AN7, AN00 to AN07, AN20 to AN27, AN150 to AN157, ANEX0, ANEX1			±3	LSB
			External op-amp connection mode			±7	LSB
DNL	Differential Nonlinearity Error					±1	LSB
-	Offset Error					±3	LSB
-	Gain Error					±3	LSB
RLADDER	Resistor Ladder	VREF=VCC		8		40	kΩ
tCONV	10-bit Conversion Time <sup>(1, 2)</sup>			2.06			μs
tCONV	8-bit Conversion Time <sup>(1, 2)</sup>			1.75			μs
tsAMP	Sampling Time <sup>(1)</sup>			0.188			μs
VREF	Reference Voltage			2		VCC	V
VIA	Analog Input Voltage			0		VREF	V

## NOTES:

1. Divide f(XIN), if exceeding 16 MHz, to keep φAD frequency at 16 MHz or less.
2. With using the sample and hold function.

**Table 25.6 D/A Conversion Characteristics**

(VCC=4.2 to 5.5V, VSS=0V at Topr = -40 to 85°C (T version)/-40 to 105°C (U version),  
f(BCLK)=32MHz unless otherwise specified)

Symbol	Parameter	Measurement Condition		Standard			Unit
				Min.	Typ.	Max.	
-	Resolution					8	Bits
-	Absolute Accuracy					1.0	%
tsU	Setup Time					3	μs
Ro	Output Resistance			4	10	20	kΩ
IvREF	Reference Power Supply Input Current	(Note 1)				1.5	mA

## NOTE:

1. Measurement when using one D/A converter. The DAi register (i=0, 1) of the D/A converter, not being used, is set to "0016". The resistor ladder in the A/D converter is excluded.  
IvREF flows even if the VCUT bit in the ADOCON1 register is set to "0" (no VREF connection).

**Table 25.7 Flash Memory Version Electrical Characteristics**  
**(VCC=4.5 to 5.5V at Topr= 0 to 60°C unless otherwise specified)**

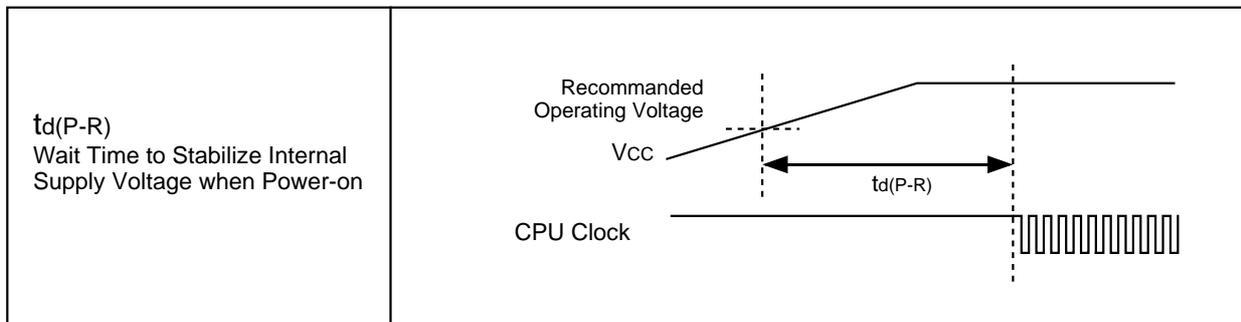
Symbol	Parameter	Standard			Unit	
		Min.	Typ.	Max.		
-	Program and Erase Endurance <sup>(2)</sup>	100			cycles	
-	Word Program Time (Vcc=5.0V, Topr=25° C)		25	200	µs	
-	Lock Bit Program Time		25	200	µs	
-	Block Erase Time (Vcc=5.0V, Topr=25° C)	4-Kbyte Block		0.3	4	s
		8-Kbyte Block		0.3	4	s
		32-Kbyte Block		0.5	4	s
		64-Kbyte Block		0.8	4	s
-	All-Unlocked-Block Erase Time <sup>(1)</sup>			4 x n	s	
tps	Wait Time to Stabilize Flash Memory Circuit			15	µs	
-	Data Hold Time (Topr=-40 to 85 ° C)	10			years	

NOTES:

1. *n* denotes the number of block to be erased.
2. Number of program-erase cycles per block.  
 If Program and Erase Endurance is *n* cycle (*n*=100), each block can be erased and programmed *n* cycles.  
 For example, if a 4-Kbyte block A is erased after programming a word data 2,048 times, each to a different address, this counts as one program and erase endurance. Data can not be programmed to the same address more than once without erasing the block. (rewrite prohibited).

**Table 25.8 Power Supply Timing**

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Wait Time to Stabilize Internal Supply Voltage when Power-on	Vcc=4.2 to 5.5V			2	ms



**Figure 25.1 Power Supply Timing Diagram**

VCC=5V

**Timing Requirements**

(VCC=4.2 to 5.5V, VSS=0V at Topr = -40 to 85°C (T version)/-40 to 105°C (U version) unless otherwise specified)

**Table 25.9 External Clock Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc	External Clock Input Cycle Time	31.25		ns
tw(H)	External Clock Input High ("H") Width	13.75		ns
tw(L)	External Clock Input Low ("L") Width	13.75		ns
tr	External Clock Rise Time		5	ns
tf	External Clock Fall Time		5	ns

**Timing Requirements**

(VCC=4.2 to 5.5V, VSS=0V at Topr = -40 to 85°C (T version)/-40 to 105°C (U version) unless otherwise specified)

**Table 25.10 Timer A Input (Count Source Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN Input Cycle Time	100		ns
tw(TAH)	TAiIN Input High ("H") Width	40		ns
tw(TAL)	TAiIN Input Low ("L") Width	40		ns

**Table 25.11 Timer A Input (Gate Input in Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN Input Cycle Time	400		ns
tw(TAH)	TAiIN Input High ("H") Width	200		ns
tw(TAL)	TAiIN Input Low ("L") Width	200		ns

**Table 25.12 Timer A Input (External Trigger Input in One-Shot Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN Input Cycle Time	200		ns
tw(TAH)	TAiIN Input High ("H") Width	100		ns
tw(TAL)	TAiIN Input Low ("L") Width	100		ns

**Table 25.13 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(TAH)	TAiIN Input High ("H") Width	100		ns
tw(TAL)	TAiIN Input Low ("L") Width	100		ns

**Table 25.14 Timer A Input (Counter Increment/Decrement Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(UP)	TAiOUT Input Cycle Time	2000		ns
tw(UPH)	TAiOUT Input High ("H") Width	1000		ns
tw(UPL)	TAiOUT Input Low ("L") Width	1000		ns
tsu(UP-TIN)	TAiOUT Input Setup Time	400		ns
th(TIN-UP)	TAiOUT Input Hold Time	400		ns

**Timing Requirements**

(VCC=4.2 to 5.5V, VSS=0V at Topr = -40 to 85°C (T version)/-40 to 105°C (U version) unless otherwise specified)

**Table 25.15 Timer B Input (Count Source Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TB)</sub>	TB <sub>iIN</sub> Input Cycle Time (counted on one edge)	100		ns
t <sub>w(TBH)</sub>	TB <sub>iIN</sub> Input High ("H") Width (counted on one edge)	40		ns
t <sub>w(TBL)</sub>	TB <sub>iIN</sub> Input Low ("L") Width (counted on one edge)	40		ns
t <sub>c(TB)</sub>	TB <sub>iIN</sub> Input Cycle Time (counted on both edges)	200		ns
t <sub>w(TBH)</sub>	TB <sub>iIN</sub> Input High ("H") Width (counted on both edges)	80		ns
t <sub>w(TBL)</sub>	TB <sub>iIN</sub> Input Low ("L") Width (counted on both edges)	80		ns

**Table 25.16 Timer B Input (Pulse Period Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TB)</sub>	TB <sub>iIN</sub> Input Cycle Time	400		ns
t <sub>w(TBH)</sub>	TB <sub>iIN</sub> Input High ("H") Width	200		ns
t <sub>w(TBL)</sub>	TB <sub>iIN</sub> Input Low ("L") Width	200		ns

**Table 25.17 Timer B Input (Pulse Width Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TB)</sub>	TB <sub>iIN</sub> Input Cycle Time	400		ns
t <sub>w(TBH)</sub>	TB <sub>iIN</sub> Input High ("H") Width	200		ns
t <sub>w(TBL)</sub>	TB <sub>iIN</sub> Input Low ("L") Width	200		ns

**Table 25.18 A/D Trigger Input**

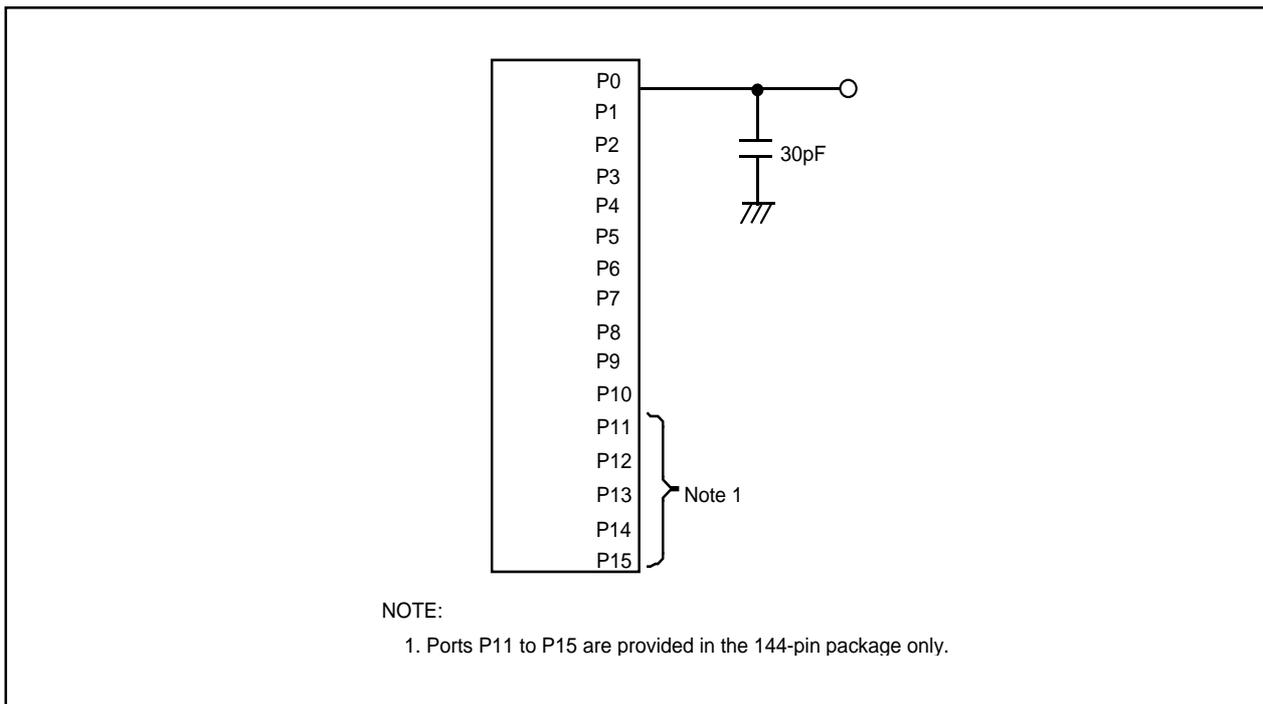
Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(AD)</sub>	AD <sub>TRG</sub> Input Cycle Time (required for trigger)	1000		ns
t <sub>w(ADL)</sub>	AD <sub>TRG</sub> Input Low ("L") Pulse Width	125		ns

**Table 25.19 Serial I/O**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(CK)</sub>	CLK <sub>i</sub> Input Cycle Time	200		ns
t <sub>w(CKH)</sub>	CLK <sub>i</sub> Input High ("H") Width	100		ns
t <sub>w(CKL)</sub>	CLK <sub>i</sub> Input Low ("L") Width	100		ns
t <sub>d(C-Q)</sub>	TxD <sub>i</sub> Output Delay Time		80	ns
t <sub>h(C-Q)</sub>	TxD <sub>i</sub> Hold Time	0		ns
t <sub>su(D-C)</sub>	RxD <sub>i</sub> Input Setup Time	30		ns
t <sub>h(C-Q)</sub>	RxD <sub>i</sub> Input Hold Time	90		ns

**Table 25.20 External Interrupt INT<sub>i</sub> Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>w(INH)</sub>	INT <sub>i</sub> Input High ("H") Width	250		ns
t <sub>w(INL)</sub>	INT <sub>i</sub> Input Low ("L") Width	250		ns

**Figure 25.2 P0 to P15 Measurement Circuit**

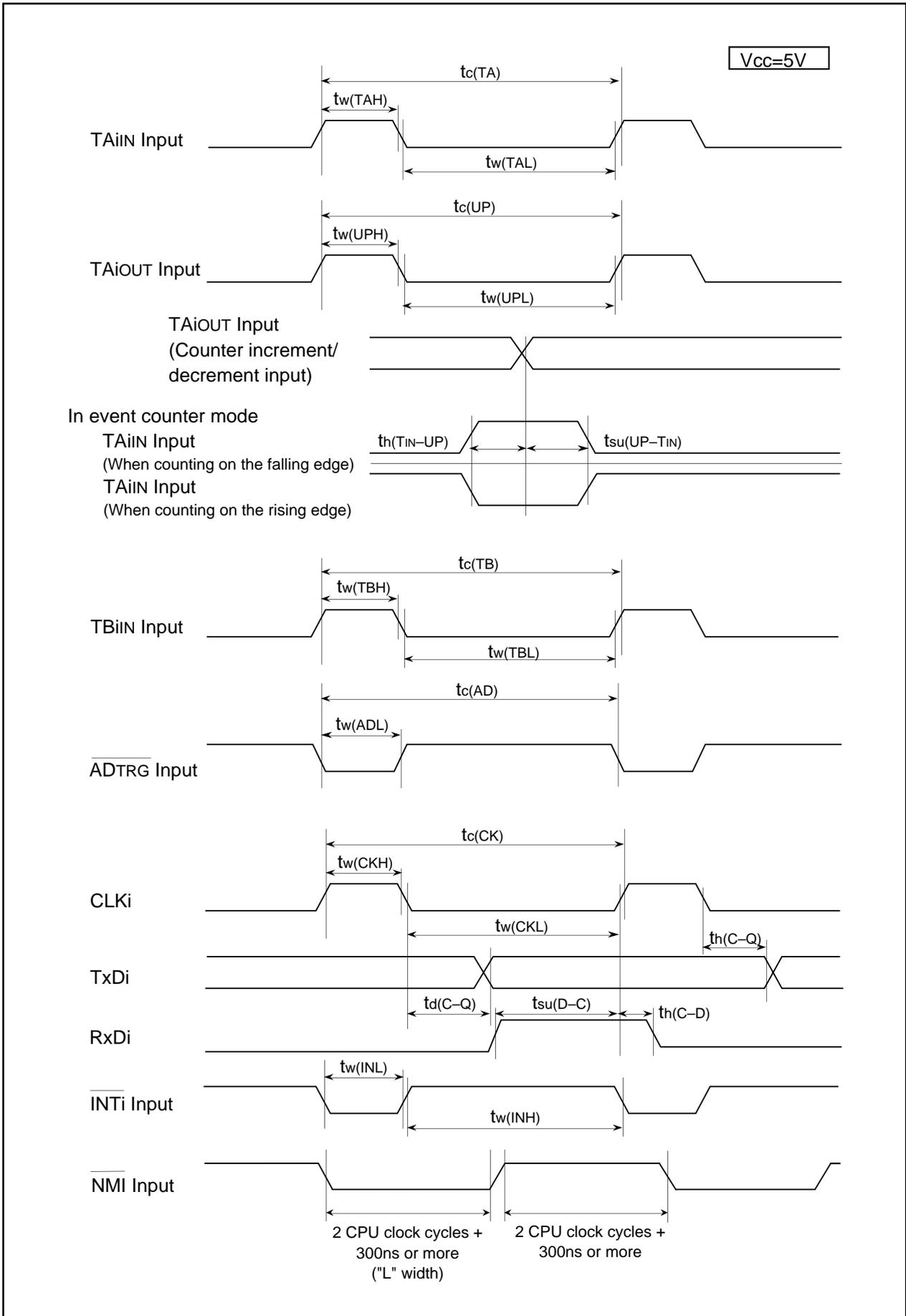


Figure 25.3 Vcc=5V Timing Diagram

## 26. Precautions

### 26.1 Special Function Registers (SFRs)

#### 26.1.1 100-Pin Package

Set address spaces 03CB<sub>16</sub>, 03CE<sub>16</sub>, 03CF<sub>16</sub>, 03D2<sub>16</sub>, 03D3<sub>16</sub> to "FF<sub>16</sub>" after reset when using the 100-pin package. Address space 03DC<sub>16</sub> must be set to "00<sub>16</sub>" after reset.

#### 26.1.2 Register Settings

Table 26.2 lists registers containing bits which can only be written to. Set these registers with immediate values. When establishing the next value by altering the present value, write the present value to the RAM as well as to the register. Transfer the next value to the register after making changes in the RAM.

**Table 26.1 Registers with Write-only Bits**

Register	Address	Register	Address
WDTS Register	000E <sub>16</sub>	U3BRG Register	0329 <sub>16</sub>
G0RI Register	00EC <sub>16</sub>	U3TB Register	032B <sub>16</sub> , 032A <sub>16</sub>
G1RI Register	012C <sub>16</sub>	U2BRG Register	0339 <sub>16</sub>
U1BRG Register	02E9 <sub>16</sub>	U2TB Register	033B <sub>16</sub> , 033A <sub>16</sub>
U1TB Register	02EB <sub>16</sub> , 02EA <sub>16</sub>	UDF Register	0344 <sub>16</sub>
U4BRG Register	02F9 <sub>16</sub>	TA0 Register <sup>(1)</sup>	0347 <sub>16</sub> , 0346 <sub>16</sub>
U4TB Register	02FB <sub>16</sub> , 02FA <sub>16</sub>	TA1 Register <sup>(1)</sup>	0349 <sub>16</sub> , 0348 <sub>16</sub>
TA11 Register	0303 <sub>16</sub> , 0302 <sub>16</sub>	TA2 Register <sup>(1)</sup>	034B <sub>16</sub> , 034A <sub>16</sub>
TA21 Register	0305 <sub>16</sub> , 0304 <sub>16</sub>	TA3 Register <sup>(1)</sup>	034D <sub>16</sub> , 034C <sub>16</sub>
TA41 Register	0307 <sub>16</sub> , 0306 <sub>16</sub>	TA4 Register <sup>(1)</sup>	034F <sub>16</sub> , 034E <sub>16</sub>
DTT Register	030C <sub>16</sub>	U0BRG Register	0369 <sub>16</sub>
ICTB2 Register	030D <sub>16</sub>	U0TB Register	036B <sub>16</sub> , 36A <sub>16</sub>

**NOTE:**

1. In one-shot timer mode and pulse width modulation mode only.

## 26.2 Clock Generation Circuit

### 26.2.1 CPU Clock

- When the CPU operating frequency is 24 MHz or more, use the following procedure for better EMC (Electromagnetic Compatibility) performance.
  - 1) Oscillator connected between the XIN and XOUT pins, or external clock applied to the XIN pin, has less than 24 MHz frequency.
  - 2) Use the PLL frequency synthesizer to multiply the main clock.
- The main clock frequency must be 24 MHz or less.

### 26.2.2 Sub Clock

Set the CM03 bit to "0" (XCIN-XCOUT drive capacity "LOW") when selecting the sub clock (XCIN-XCOUT) as the CPU clock, or Timer A or Timer B count source (fc32).

#### 26.2.2.1 Sub Clock Oscillation

When oscillating the sub clock, set the CM04 bit in the CM0 register to "1" (XCIN-XCOUT oscillation function) after setting the CM07 bit in the CM0 register to "0" (clock other than sub clock) and the CM03 bit to "1" (XCIN-XCOUT drive capacity "HIGH"). Set the CM03 bit to "0" after sub clock oscillation stabilizes.

Set the sub clock as the CPU clock, or Timer A or Timer B count source (fc32) after the above settings are completed.

#### 26.2.2.2 Using Stop Mode

When the microcomputer enters stop mode, the CM03 bit is automatically set to "1" (XCIN-XCOUT drive capacity "HIGH"). Use the following procedure to select the main clock as the CPU clock when entering stop mode.

- 1) Set the CM17 bit in the CM1 register to "0" (main clock).
- 2) Set the CM21 bit in the CM2 register to "0" (clock selected by the CM17 bit).
- 3) Set the CM07 bit in the CM0 register to "0" (clock selected by the CM21 bit divided by the MCD register setting).

After exiting stop mode, wait for the sub clock oscillation to stabilize. Then set the CM03 bit to "0" and the CM07 bit to "1" (sub clock).

#### 26.2.2.3 Oscillation Parameter Matching

If the sub clock oscillation parameters have only been evaluated with the drive capacity "HIGH", the parameters should be reevaluated for drive capacity "LOW".

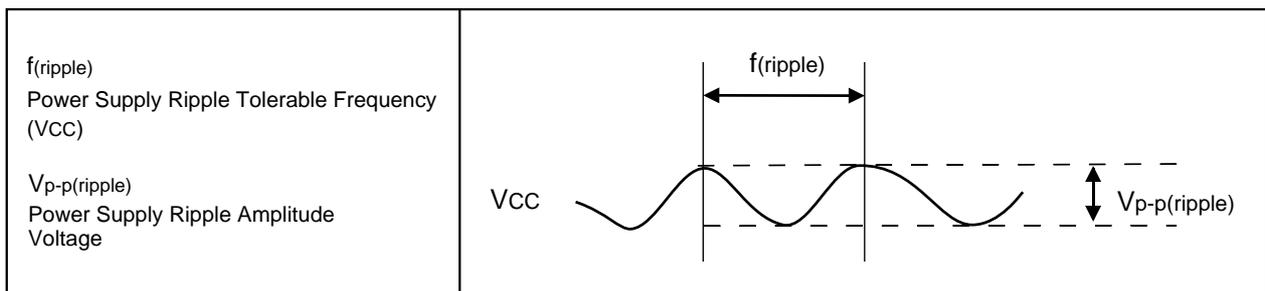
Contact your oscillator manufacturer for details on matching parameters.

### 26.2.3 PLL Frequency Synthesizer

Stabilize supply voltage to meet the power supply standard when using the PLL frequency synthesizer.

**Table 26.2 Power Supply Ripple**

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
$f_{(ripple)}$	Power Supply Ripple Tolerable Frequency (V <sub>CC</sub> )	V <sub>CC</sub> =5V			10	kHz
$V_{P-P(ripple)}$	Power Supply Ripple Voltage Fluctuation Range	V <sub>CC</sub> =5V			0.5	V
$V_{CC}(\Delta V/\Delta T)$	Power Supply Ripple Voltage Fluctuation Rate	V <sub>CC</sub> =5V			1	V/ms



**Figure 26.1 Power Supply Fluctuation Timing**

### 26.2.4 External Clock

Do not stop an external clock running if the main clock is selected as the CPU clock while the external clock is applied to the XIN pin.

Do not set the CM05 bit in the CM0 register to "1" (main clock stopped) while the external clock input is used for the CPU clock.

### 26.2.5 Clock Divide Ratio

Set the PM12 bit in the PM1 register to "0" (no wait state) when changing the MCD4 to MCD0 bit settings in the MCD register.

### 26.2.6 Power Consumption Control

Stabilize the main clock, sub clock or PLL clock to switch the CPU clock source to each clock.

#### 26.2.6.1 Wait Mode

When entering wait mode while the CM02 bit in the CM0 register is set to "1" (peripheral function stop in wait mode), set the MCD4 to MCD0 bits in the MCD register to maintain the 10-MHz CPU clock frequency or less.

When entering wait mode, the instruction queue reads ahead to instructions following the WAIT instruction, and the program stops. Write at least 4 NOP instructions after the WAIT instruction.

### 26.2.6.2 Stop Mode

- Use the following procedure to select the main clock as the CPU clock when entering stop mode.
  - 1) Set the CM17 bit in the CM1 register to "0" (main clock).
  - 2) Set the CM21 bit in the CM2 register to "0" (clock selected by the CM17 bit).
  - 3) Set the CM07 bit in the CM0 register to "0" (clock selected by the CM21 bit divided by the MCD register setting).

If the PLL clock is selected as the CPU clock source, set the CM17 bit to "0" (main clock) and the PLC07 bit in the PLC0 register to "0" (PLL off) before entering stop mode.

- The microcomputer cannot enter stop mode if a low-level signal ("L") is applied to the  $\overline{\text{NMI}}$  pin. Apply a high-level ("H") signal instead.
- If stop mode is exited by any reset, apply an "L" signal to the  $\overline{\text{RESET}}$  pin until a main clock oscillation is stabilized enough.
- If using the  $\overline{\text{NMI}}$  interrupt to exit stop mode, use the following procedure to set the CM10 bit in the CM1 register (all clocks stopped).
  - 1) Exit stop mode with using the  $\overline{\text{NMI}}$  interrupt.
  - 2) Generate a dummy interrupt.
  - 3) Set the CM10 bit to "1".

```
e.g.,      int    #63                ; dummy interrupt
           bset   cm1                ; all clocks stopped
```

```
/* dummy interrupt handling */
```

```
dummy
  reit
```

- When entering stop mode, the instruction queue reads ahead to instructions following the instruction setting the CM10 bit in the CM1 register to "1" (all clocks stopped), and the program stops. When the microcomputer exits stop mode, the instruction lined in the instruction queue is executed before the interrupt routine for recovery is done.

Write the JMP.B instruction, as follows, after the instruction setting the CM10 bit in the CM1 register to "1" (all clocks stopped).

```
e.g.,      bset 0, prcr                ; protection removed
           bset 0, cm1                ; all clocks stopped
           jmp.b LABEL_001            ; JMP.B instruction executed (no instuction between JMP.B
                                       ; and LABEL.)
```

```
LABEL_001:
  nop                    ; NOP (1)
  nop                    ; NOP (2)
  nop                    ; NOP (3)
  nop                    ; NOP (4)
  mov.b #0, prcr         ; Protection set
  .
  .
  .
```

### 26.2.6.3 Suggestions for Reducing Power Consumption

The followings are suggestions for reducing power consumption when programming or designing systems.

**Ports:** I/O ports maintains the same state despite the microcomputer entering wait mode or stop mode. Current flows through active output ports. Feedthrough current flows through input ports in a high-impedance state. Set unassigned ports as input ports and stabilize electrical potential before entering wait mode or stop mode.

**A/D Converter:** If the A/D conversion is not performed, set the VCUT bit in the AD0CON1 register to "0" (no VREF connection). Set the VCUT bit to "1" (VREF connection) and wait at least 1 $\mu$ s before starting the A/D conversion.

**D/A Converter:** Set the DAi bit (i=0, 1) in the DACON register to "0" (output disabled) and set the DAi register to "0016" when the D/A conversion is not performed.

**Peripheral Function Stop:** Set the CM02 bit in the CM0 register while in wait mode to stop unnecessary peripheral functions. However, this does not reduce power consumption because the peripheral function clock (fc32) generating from the sub clock does not stop. When in low-speed mode and low-power consumption mode, do not enter wait mode when the CM02 bit is set to "1" (peripheral clock stops in wait mode).

### 26.3 Protection

The PRC2 bit setting in the PRCR register is changed to "0" (write disabled) when an instruction is written to any address after the PRC2 bit is set to "1" (write enabled). Write instruction immediately after setting the PRC2 bit to "1" to change registers protected by the PRC2 bit. Do not generate an interrupt or a DMA transfer between the instruction to set the PRC2 bit to "1" and the next instruction.

## 26.4 Interrupts

### 26.4.1 ISP Setting

After reset, the ISP is set to "00000016". The program runs out of control if an interrupt is acknowledged before the ISP is set. Therefore, the ISP must be set before an interrupt request is generated. Set the ISP to an even address, which allows interrupt sequences to be executed at a higher speed.

To use  $\overline{\text{NMI}}$  interrupt, set the ISP at the beginning of the program. The  $\overline{\text{NMI}}$  interrupt can be acknowledged after the first instruction has been executed after reset.

### 26.4.2 $\overline{\text{NMI}}$ Interrupt

- $\overline{\text{NMI}}$  interrupt cannot be denied. Connect the  $\overline{\text{NMI}}$  pin to Vcc via a resistor (pull-up) when not in use.
- The P8\_5 bit in the P8 register indicates the  $\overline{\text{NMI}}$  pin value. Read the P8\_5 bit only to determine the pin level after a  $\overline{\text{NMI}}$  interrupt occurs.
- "H" and "L" signals applied to the  $\overline{\text{NMI}}$  pin must be over 2 CPU clock cycles + 300 ns wide.
- $\overline{\text{NMI}}$  interrupt request may not be acknowledged if this and other interrupt requests are generated simultaneously.

### 26.4.3 $\overline{\text{INT}}$ Interrupt

- Edge Sensitive  
"H" and "L" signals applied to the  $\overline{\text{INT}}0$  to  $\overline{\text{INT}}5$  pins must be at least 250 ns wide, regardless of the CPU clock.
- Level Sensitive  
"H" and "L" signals applied to the  $\overline{\text{INT}}0$  to  $\overline{\text{INT}}5$  pins must be at least 1 CPU clock cycle + 200 ns wide. For example, "H" and "L" must be at least 234ns wide if  $X_{IN}=30\text{MHz}$  with no division.
- The IR bit setting may change to "1" (interrupt requested) when switching the polarity of the  $\overline{\text{INT}}0$  to  $\overline{\text{INT}}5$  pins. Set the IR bit to "0" (no interrupt requested) after selecting the polarity. Figure 26.3 shows an example of the switching procedure for the  $\overline{\text{INT}}$  interrupt.

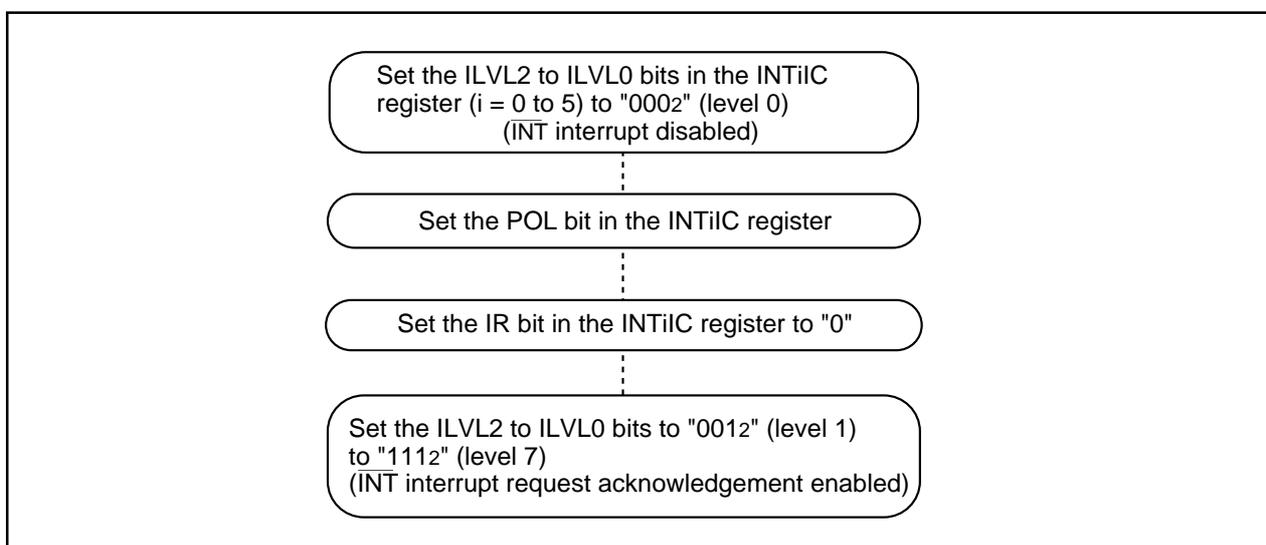


Figure 26.2 Switching Procedure for  $\overline{\text{INT}}$  Interrupt

#### 26.4.4 Watchdog Timer Interrupt

Reset the watchdog timer after a watchdog timer interrupt occurs.

#### 26.4.5 Changing Interrupt Control Register

To change the interrupt control register while the interrupt request is denied, follow the instructions below.

##### Changing IR bit

The IR bit setting may not change to "0" (no interrupt requested) depending on the instructions written. If this is a problem, use the following instruction to change the register: MOV

##### Changing Bits Except IR Bit

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

#### 26.4.6 Changing IIOiR Register (i = 0 to 6, 8 to 11)

Use the following instructions to set bits 1 to 7 in the IIOiR register to "0" (no interrupt requested): AND, BCLR

#### 26.4.7 Changing RLVL Register

The DMAII bit is indeterminate after reset. When using the DMAII bit to generate an interrupt, set the interrupt control register after setting the DMAII bit to "0" (interrupt priority level 7 available for interrupts).

## 26.5 DMAC

- Set DMAC-associated registers while the MDi1 and MDi0 bits (i=0 to 3) in the channel to be used are set to "002" (DMA disabled). Set the MDi1 and MDi0 bits to "012" (single transfer) or "112" (repeat transfer) at the end of setup procedure to start DMA requests.
- Do not set the DRQ bit in the DMiSL register to "0" (no request).  
If a DMA request is generated but the receiving channel is not ready to receive<sup>(1)</sup>, the DMA transfer does not occur and the DRQ bit is set to "0".

### NOTE:

1. The MDi1 and MDi0 bits are set to "002" or the DCTi register is set to "0000<sub>16</sub>" (transferred 0 times).
- To start a DMA transfer by a software trigger, set the DSR bit and DRQ bit in the DMiSL register to "1" simultaneously.  
e.g.,  

```
OR.B #0A0h,DMiSL ; Set the DSR and DRQ bits to "1" simultaneously
```
  - Do not generate a channel i DMA request when setting the MDi1 and MDi0 bits in the DMDj register (j=0,1) corresponding to channel i to "012" (single transfer) or "112" (repeat transfer), if the DCTi register of channel i is set to "1".
  - Select the peripheral function which causes the DMA request after setting the DMA-associated registers. If none of the conditions above (setting  $\overline{\text{INT}}$  interrupt as DMA request source) apply, do not write "1" to the DCTi register.
  - Enable DMA<sup>(2)</sup> after setting the DMiSL register (i=0 to 3) and waiting six BCLK cycles or more by program.

### NOTE:

2. DMA is enabled when the values set in the MDi1 and MDi0 bits in the DMDj register are changed from "002" (DMA disabled) to "012" (single transfer) or "112" (repeat transfer).

## 26.6 Timer

### 26.6.1 Timers A and B

Timers stop after reset. Set the TAI<sub>S</sub>(i=0 to 4) bit or TB<sub>j</sub>S(j=0 to 5) bit in the TABSR register or TBSR register to "1" (starts counting) after setting operating mode, count source and counter.

The following registers and bits must be set while the TAI<sub>S</sub> bit or TB<sub>j</sub>S bit is set to "0" (stops counting).

- TAI<sub>MR</sub>, TB<sub>j</sub>MR register
- TAI, TB<sub>j</sub> register
- UDF register
- TAZIE, TA0TGL, TA0TGH bits in the ONSF register
- TRGSR register

### 26.6.2 Timer A

The TA1<sub>OUT</sub>, TA2<sub>OUT</sub> and TA4<sub>OUT</sub> pins are placed in high-impedance states when a low-level ("L") signal is applied to the  $\overline{\text{NMI}}$  pin while the INV03 and INV02 bits in the INVC0 register are set to "112" (forced cutoff of the three-phase output by an "L" signal applied to the  $\overline{\text{NMI}}$  pin).

#### 26.6.2.1 Timer A (Timer Mode)

- The TAI<sub>S</sub> bit (i=0 to 4) in the TABSR register is set to "0" (stops counting) after reset. Set the TAI<sub>S</sub> bit to "1" (starts counting) after selecting an operating mode and setting the TAI register.
- The TAI register indicates the counter value during counting at any given time. However, the counter is "FFFF<sub>16</sub>" when reloading. The setting value can be read after setting the TAI register while the counter stops and before the counter starts counting.

#### 26.6.2.2 Timer A (Event Counter Mode)

- The TAI<sub>S</sub> (i=0 to 4) bit in the TABSR register is set to "0" (stops counting) after reset. Set the TAI<sub>S</sub> bit to "1" (starts counting) after selecting an operating mode and setting the TAI register.
- The TAI register indicates the counter values during counting at any given time. However, the counter will be "FFFF<sub>16</sub>" during underflow and "0000<sub>16</sub>" during overflow, when reloading. The setting value can be read after setting the TAI register while the counter stops and before the counter starts counting.

### 26.6.2.3 Timer A (One-shot Timer Mode)

- The TAI<sub>S</sub> (i=0 to 4) bit in the TABSR register is set to "0" (stops counting) after reset. Set the TAI<sub>S</sub> bit to "1" (starts counting) after selecting an operating mode and setting the TAI register.
- The followings occur when the TABSR register is set to "0" (stops counting) while counting:
  - The counter stops counting and the microcomputer reloads contents of the reload register.
  - The TAI<sub>OUT</sub> pin becomes low ("L").
  - The IR bit in the TAI<sub>IC</sub> register is set to "1" (interrupt requested) after one CPU clock cycle.
- The output of the one-shot timer is synchronized with an internal count source. When set to an external trigger, there is a delay of one count source cycle maximum, from trigger input to the TAI<sub>IN</sub> pin to the one-shot timer output.
- The IR bit is set to "1" when the following procedures are performed to set timer mode:
  - selecting one-shot timer mode after reset.
  - switching from timer mode to one-shot timer mode.
  - switching from event counter mode to one-shot timer mode.
 Therefore, set the IR bit to "0" to generate a timer A<sub>i</sub> interrupt (IR bit) after performing these procedures.
- When a trigger is generated while counting, the reload register reloads and continues counting after the counter has decremented once following a re-trigger. To generate a trigger while counting, wait at least 1 count source cycle after the previous trigger has been generated and generate a re-trigger.
- If an external trigger input is selected to start counting in timer A one-shot timer mode, do not provide another external trigger input again for 300 ns before the timer A counter value reaches "0000<sub>16</sub>". One-shot timer may stop counting.

### 26.6.2.4 Timer A (Pulse Width Modulation Mode)

- The TAI<sub>S</sub>(i=0 to 4) bit in the TABSR register is set to "0" (stops counting) after reset. Set the TAI<sub>S</sub> bit to "1" (starts counting) after selecting an operating mode and setting the TAI register.
- The IR bit is set to "1" when the following procedures are performed to set timer mode:
  - Selecting PWM mode after reset
  - Switching from timer mode to PWM mode
  - Switching from event counter mode to PWM mode
 Therefore, set the IR bit to "0" by program to generate a timer A<sub>i</sub> interrupt (IR bit) after performing these procedures.
- The followings occur when the TAI<sub>S</sub> bit is set to "0" (stops counting) while PWM pulse is output:
  - The counter stops counting
  - Output level changes to low ("L") and the IR bit changes to "1" when the TAI<sub>OUT</sub> pin is held high ("H")
  - The IR bit and the output level remain unchanged when TAI<sub>OUT</sub> pin is held "L"

### 26.6.3 Timer B

#### 26.6.3.1 Timer B (Timer Mode, Event Counter Mode)

- The TBiS (i=0 to 5) bit is set to "0" (stops counting) after reset. Set the TBiS bit to "1" (starts counting) after selecting an operating mode and setting TBi register.  
The TB2S to TB0S bits are bits 7 to 5 in the TABSR register. The TB5S to TB3S bits are bits 7 to 5 in the TBSR register.
- The TBi register indicates the counter value during counting at any given time. However, the counter is "FFFF16" when reloading. The setting value can be read after setting the TBi register while the counter stops and before the counter starts counting.

#### 26.6.3.2 Timer B (Pulse Period/Pulse Width Measurement Mode)

- The IR bit in the TBiIC (i=0 to 5) register is set to "1" (interrupt requested) when the valid edge of a pulse to be measured is input and when the timer Bi counter overflows. The MR3 bit in the TBiMR register determines the interrupt source within an interrupt routine.
- Use another timer to count how often the timer counter overflows when an interrupt source cannot be determined by the MR3 bit, such as when a pulse to be measured is input at the same time the timer counter overflows.
- To set the MR3 bit in the TBiMR register to "0" (no overflow), set the TBiMR register after the MR3 bit is set to "1" (overflow) and one or more cycles of the count source are counted, while the TBiS bits in the TABSR and TBSR registers are set to "1" (starts counting).
- The IR bit in the TBiIC register is used to detect overflow only. Use the MR3 bit only to determine interrupt source within an interrupt routine.
- Indeterminate values are transferred to the reload register during the first valid edge input after counting is started. Timer Bi interrupt request is not generated at this time.
- The counter value is indeterminate when counting is started. Therefore, the MR3 bit setting may change to "1" (overflow) and causes timer Bi interrupt requests to be generated until a valid edge is input after counting is started.
- The IR bit may be set to "1" (interrupt requested) if the MR1 and MR0 bits in the TBiMR register are set to a different value after a count begins. If the MR1 and MR0 bits are rewritten, but to the same value as before, the IR bit remains unchanged.
- Pulse width measurement measures pulse width continuously. Use program to determine whether measurement results are high ("H") or low ("L").

## 26.7 Serial I/O

### 26.7.1 Clock Synchronous Serial I/O Mode

The  $\overline{\text{RTS}}_2$  and  $\text{CLK}_2$  pins are placed in high-impedance states when a low-level ("L") signal is applied to the  $\overline{\text{NMI}}$  pin while the  $\text{INV03}$  to  $\text{INV02}$  bits in the  $\text{INVC0}$  register are set to "112" (forced cutoff of the three-phase output by an "L" signal applied to the  $\overline{\text{NMI}}$  pin).

#### 26.7.1.1 Transmission /Reception

When the  $\overline{\text{RTS}}$  function is used while an external clock is selected, the output level of the  $\overline{\text{RTS}}_i$  pin is held "L" indicating that the microcomputer is ready for reception. The transmitting microcomputer is notified that reception is possible. The output level of the  $\overline{\text{RTS}}_i$  pin becomes high ("H") when reception begins. Therefore, connecting the  $\overline{\text{RTS}}_i$  pin to the  $\overline{\text{CTS}}_i$  pin of the transmitting microcomputer synchronizes transmission and reception. The  $\overline{\text{RTS}}$  function is disabled if an internal clock is selected.

#### 26.7.1.2 Transmission

When an external clock is selected while the  $\text{CKPOL}$  bit in the  $\text{UiC0}$  ( $i=0$  to  $4$ ) register is set to "0" (data is transmitted on the falling edge of the transfer clock and received on the rising edge) and the external clock is held "H", or when the  $\text{CKPOL}$  bit is set to "1" (data is transmitted on the rising edge of the transfer clock and received on the falling edge) and the external clock is held "L", meet the following conditions:

- Set the  $\text{TE}$  bit in the  $\text{UiC1}$  register to "1" (receive enabled)
- Set the  $\text{TI}$  bit in the  $\text{UiC1}$  register to "0" (data in the  $\text{UiTB}$  register)
- Apply "L" signal to the  $\overline{\text{CTS}}_i$  pin if the  $\overline{\text{CTS}}$  function is selected

#### 26.7.1.3 Reception

Activating the transmitter in clock synchronous serial I/O mode generates the shift clock. Therefore, set for transmission even if the microcomputer is used for reception only. Dummy data is output from the  $\text{TxDi}$  pin while receiving.

If an internal clock is selected, the shift clock is generated when the  $\text{TE}$  bit in the  $\text{UiC1}$  registers is set to "1" (receive enabled) and dummy data is set in the  $\text{UiTB}$  register. If an external clock is selected, the shift clock is generated when the external clock is input into  $\text{CLK}_i$  pin while the  $\text{TE}$  bit is set to "1" (receive enabled) and dummy data is set in the  $\text{UiTB}$  register.

When receiving data consecutively while the  $\text{RE}$  bit in the  $\text{UiC1}$  register is set to "1" (data in the  $\text{UiRB}$  register) and the next data is received by the  $\text{UART}_i$  reception register, an overrun error occurs and the  $\text{OER}$  bit in the  $\text{UiRB}$  register is set to "1" (overrun error). In this case, the  $\text{UiRB}$  register is indeterminate. When overrun error occurs, program both reception and transmission registers to retransmit earlier data. The  $\text{IR}$  bit in the  $\text{SiRIC}$  does not change when an overrun error occurs.

When receiving data consecutively, feed dummy data to the low-order byte in the  $\text{UiTB}$  register every time a reception is made.

When an external clock is selected while the  $\text{CKPOL}$  bit in the  $\text{UiC0}$  register is set to "0" (data is transmitted on the falling edge of the transfer clock and received on the rising edge) and the external clock is held "H" or when the  $\text{CKPOL}$  bit is set to "1" (data is transmitted on the rising edge of the transfer clock and received on the falling edge) and the external clock is held "L", meet the following conditions:

- Set the  $\text{RE}$  bit in the  $\text{UiC1}$  register to "1" (receive enabled)
- Set the  $\text{TE}$  bit in the  $\text{UiC1}$  register to "1" (transmit enabled)
- Set the  $\text{TI}$  bit in the  $\text{UiC1}$  register to "0" (data in the  $\text{UiTB}$  register)

### 26.7.2 UART Mode

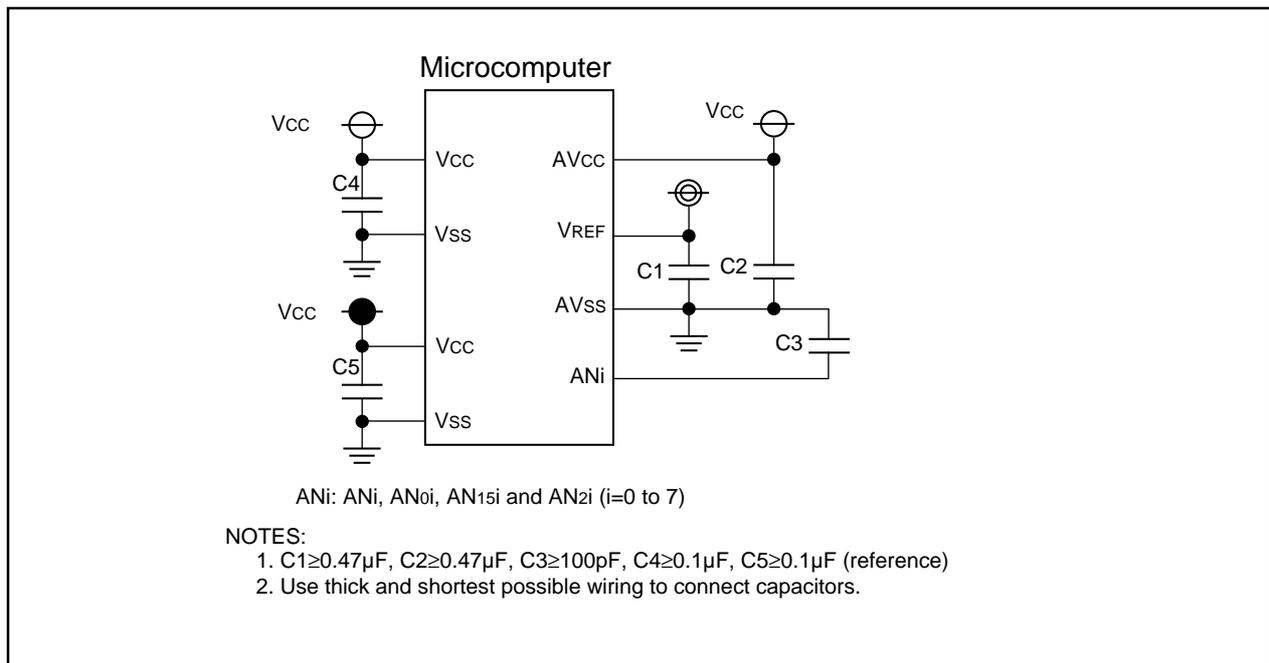
Set the UiERE bit (i=0 to 4) in the UiC1 register after setting the UiMR register.

### 26.7.3 Special Mode 1 (I<sup>2</sup>C Mode)

To generate the start condition, stop condition or restart condition, set the STSPSEL bit in the UiSMR4 register to "0" first. Then, change each condition generating bit (the STAREQ bit, STPREQ bit or RSTAREQ bit) setting from "0" to "1" after going through a half cycle of the transfer clock.

## 26.8 A/D Converter

- Set the AD0CON0 (bit 6 excluded), AD0CON1, AD0CON2, AD0CON3, and AD0CON4 registers while the A/D conversion is stopped (before a trigger is generated).
- Wait a minimum of 1 $\mu$ s before starting the A/D conversion when changing the VCUT bit setting in the AD0CON1 register from "0" (VREF no connection) to "1" (VREF connection).  
Change the VCUT bit setting from "1" to "0" after the A/D conversion is completed.
- Insert capacitors between the AVCC pin, VREF pin, analog input pin AN<sub>ij</sub> (i=none, 0, 2, 15; j=0 to 7) and AVSS pin to prevent latch-ups and malfunctions due to noise, and to minimize conversion errors. The same applies to the VCC and VSS pins. Figure 26.4 shows the use of capacitors to reduce noise.



**Figure 26.3 Use of Capacitors to Reduce Noise**

- Set the bit in the port direction register, which corresponds to the pin being used as the analog input, to "0" (input mode). Set the bit in the port direction register, which corresponds to the  $\overline{\text{ADTRG}}$  pin, to "0" (input mode) if the TRG bit in the AD0CON0 register is set to "1" (external trigger).
- When generating a key input interrupt, do not use the AN<sub>4</sub> to AN<sub>7</sub> pins as analog input pins (key input interrupt request is generated when the A/D input voltage becomes "L").
- The  $\phi_{\text{AD}}$  frequency must be 16MHz or less. When the sample and hold function is not activated, the  $\phi_{\text{AD}}$  frequency must be 250 kHz or more. If the sample and hold function is activated, the  $\phi_{\text{AD}}$  frequency must be 1MHz or more.
- Set the CH<sub>2</sub> to CH<sub>0</sub> bits in the AD0CON0 register or the SCAN<sub>1</sub> and SCAN<sub>0</sub> bits in the AD0CON1 register to re-select analog input pins when changing A/D conversion mode.

- $AVCC = VREF = VCC$ ,  
A/D input voltage (for AN0 to AN7, AN00 to AN07, and AN20 to AN27, AN150 to AN157, ANEX0, and ANEX1)  $\leq VCC$ .
- Wrong values are stored in the AD0i register (i=0 to 7) if the CPU reads the AD0i register while the AD0i register stores results from a completed A/D conversion. This occurs when the CPU clock is set to a divided main clock or a sub clock.  
In one-shot mode or single sweep mode, read the corresponding AD0i register after verifying that the A/D conversion has been completed. The IR bit in the AD0IC register determines the completion of the A/D conversion.  
In repeat mode, repeat sweep mode 0, repeat sweep mode 1, multi-port single sweep mode, and multi-port repeat sweep mode 0, use an undivided main clock as the CPU clock.
- Conversion results of the A/D converter are indeterminate if the ADST bit in the AD0CON0 register is set to "0" (A/D conversion stopped) and the conversion is forcibly terminated by program during the A/D conversion. The AD0i register not performing the A/D conversion may also be indeterminate.  
If the ADST bit is changed to "0" by program, during the A/D conversion, do not use any values obtained from the AD0i registers.
- External triggers cannot be used in DMAC operating mode. Do not read the AD00 register by program.
- Do not perform the A/D conversion in wait mode.
- Set the MCD4 to MCD0 bits in the MCD register to "100102" (no division) if using the sample and hold function.
- Do not acknowledge any interrupt requests, even if generated, before setting the ADST bit, if the A/D conversion is terminated by setting the ADST bit in the AD0CON0 register to "0" (A/D conversion stopped) while the microcomputer is A/D converting in single sweep mode.

## 26.9 Intelligent I/O

### 26.9.1 Register Setting

Operations, controlled by the values written to the G1BT, G1BCR1, G1TMCR0 to G1TMCR7, G1TPR6, G1TPR7, G1TM0 to G1TM7, G1POCR0 to G1POCR7, G1PO0 to G1PO7, G1FS and G1FE registers, are affected by the count source ( $f_{BT1}$ ) set in the BCK1 and BCK0 bits in the G1BCR0 register.

Set the BCK1 and BCK0 bits before setting the G1BT, G1BCR1, G1TMCR0 to G1TMCR7, G1TPR6, G1TPR7, G1TM0 to G1TM7, G1POCR0 to G1POCR7, G1PO0 to G1PO7, G1FS and G1FE registers.

Operations, controlled by the values written to the G0RI and G1RI, G0TO and G1TO, G0CR and G1CR, G0RB and G1RB, G0MR and G1MR, G0EMR and G1EMR, G0ETC and G1ETC, G0ERC and G1ERC, G0IRF, G1IRF, G0TB and G1TB, G0CMP0 to G0CMP3, G1CMP0 to G1CMP3, G0MSK0 and G0MSK1, G1MSK0 and G1MSK1, G0TCRC and G1TCRC, G0RCRC and G1RCRC registers are affected by the transfer clock.

Set transfer clock before setting the G0RI and G1RI, G0TO and G1TO, G0CR and G1CR, G0RB and G1RB, G0MR and G1MR, G0EMR and G1EMR, G0ETC and G1ETC, G0ERC and G1ERC, G0IRF and G1IRF, G0TB and G1TB, G0CMP0 to G0CMP3, G1CMP0 to G1CMP3, G0MSK0 and G0MSK1, G1MSK0 and G1MSK1, G0TCRC and G1TCRC, G0RCRC and G1RCRC registers.

## 26.10 Programmable I/O Ports

- Because ports P72 to P75, P80, and P81 have three-phase PWM output forced cutoff function, they are affected by the three-phase motor control timer function and the  $\overline{\text{NMI}}$  pin when these ports are set for output functions (port output, timer output, three-phase PWM output, serial I/O output, intelligent I/O output).

Table 26.4 shows the INVC0 register setting, the  $\overline{\text{NMI}}$  pin input level and the state of output ports.

**Table 26.3 INVC0 Register and the  $\overline{\text{NMI}}$  Pin**

Setting Value of the INVC0 Register		Signal level Applied to the NMI Pin	P72 to P75, P80, P81 Pin States (When Setting Them as Output Pins)
INV02 Bit	INV03 Bit		
0 (Not Using the Three-Phase Motor Control Timer Functions)	-	-	Provides functions selected by the PS1, PSL1, PSC, PS2, PSL2 registers
1 (Using the Three-Phase Motor Control Timer Functions)	0 (Three-Phase Motor Control Timer Output Disabled)	-	High-impedance state
	1 (Three-Phase Motor Control Timer Output Enabled) <sup>(1)</sup>	H	Provides functions selected by the PS1, PSL1, PSC, PS2, PSL2 registers
		L (Forcibly Terminated)	High-impedance state

**NOTE:**

1. The INV03 bit is set to "0" after a low-level ("L") signal is applied to the  $\overline{\text{NMI}}$  pin.

- The availability of pull-up resistors is indeterminate until internal power voltage stabilizes, if the RESET pin is held "L".
- The input threshold voltage varies between programmable I/O ports and peripheral functions. Therefore, if the level of the voltage applied to a pin shared by both programmable I/O ports and peripheral functions is not within the recommended operating condition,  $V_{IH}$  and  $V_{IL}$  (neither "H" nor "L"), the level may vary depending on the programmable ports and peripheral functions.

## 26.11 Flash Memory Version

### 26.11.1 Boot Mode

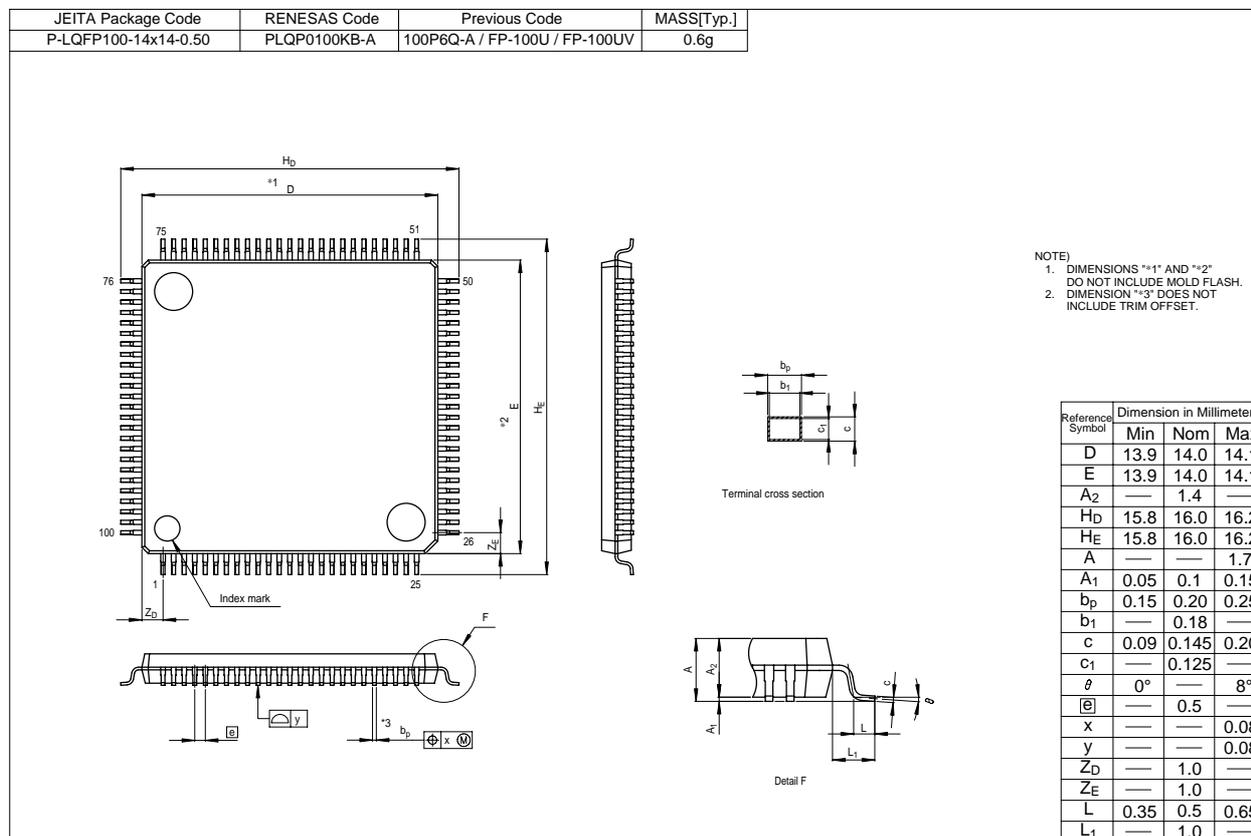
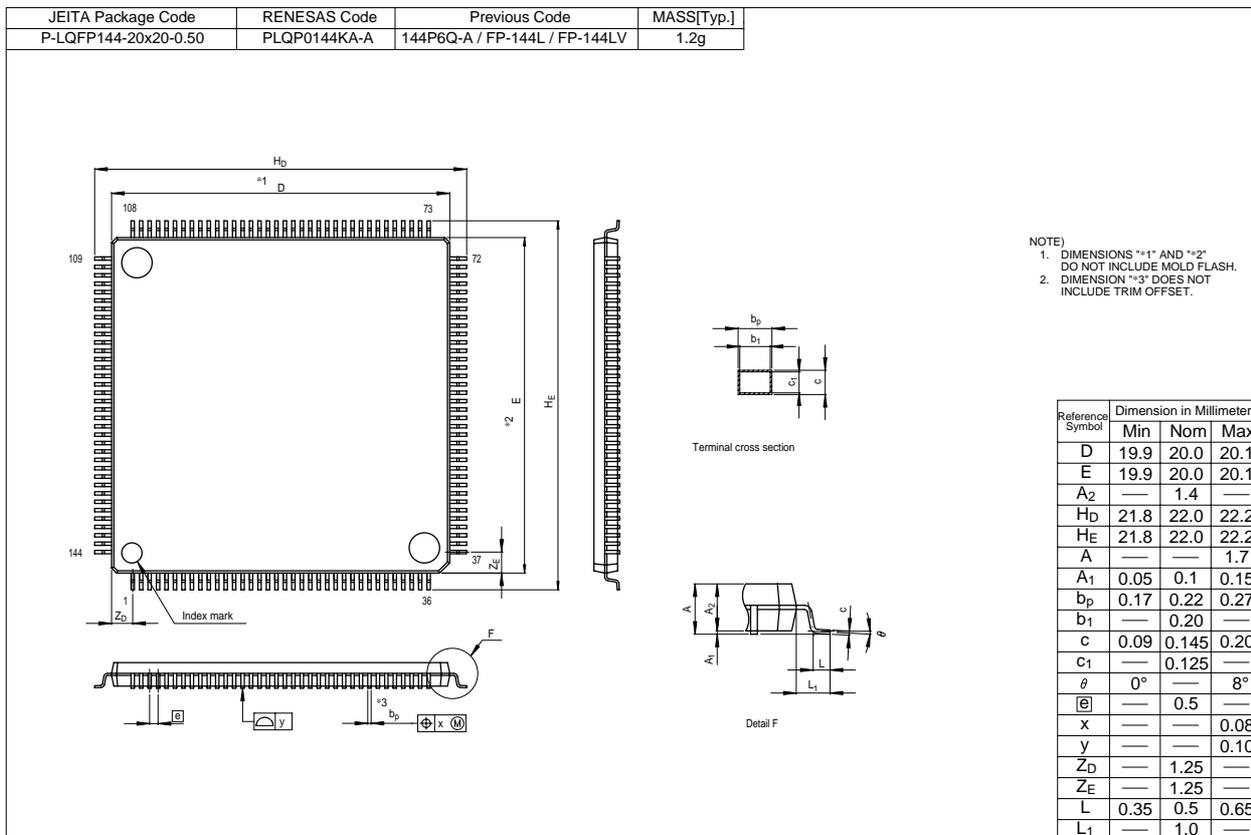
I/O pins may not be placed in high-impedance states until internal voltage stabilizes, when power is turned on in boot mode. Use the following procedure to turn on power in boot mode.

- 1) Apply a low-level ("L") signal to the  $\overline{\text{RESET}}$  and the CNVss pin
- 2) Wait a minimum of 2 ms after VCC reaches 2.7V or above (until internal voltage stabilizes)
- 3) Apply a high-level ("H") signal to the CNVss pin
- 4) Apply an "H" signal to the  $\overline{\text{RESET}}$  pin (reset exited)

**26.12 Noise**

Connect a bypass capacitor (0.1 $\mu$ F or more) between VCC and VSS by shortest path, using thick wires.

# Package Dimensions



# Register Index

## A

AD00 to AD07 **225**

AD0CON0 **221**

AD0CON1 **222**

AD0CON2 **223**

AD0CON3 **224**

AD0CON4 **225**

AIER **98**

## C

C0AFS **336**

C0BPR **307**

C0CONR **305**

C0CTLR0 **296**

C0CTLR1 **299**

C0EFR **315**

C0EIMKR **313**

C0EISTR **314**

C0GMR0 **320**

C0GMR1 **321**

C0GMR2 **322**

C0GMR3 **323**

C0GMR4 **324**

C0IDR **304**

C0LMAR0 **320**

C0LMAR1 **321**

C0LMAR2 **322**

C0LMAR3 **323**

C0LMAR4 **324**

C0LMBR0 **320**

C0LMBR1 **321**

C0LMBR2 **322**

C0LMBR3 **323**

C0LMBR4 **324**

C0MCTL0 to C0MCTL15 **327**

C0MDR **316**

C0REC **309**

C0SBS **331**

C0SIMKR **312**

C0SISTR **310**

C0SLOT0\_0 **332**

C0SLOT0\_1 **332**

C0SLOT0\_2 **333**

C0SLOT0\_3 **333**

C0SLOT0\_4 **334**

C0SLOT0\_5 **334**

C0SLOT0\_6 to C0SLOT0\_13 **335**

C0SLOT0\_14 **335**

C0SLOT0\_15 **335**

C0SLOT1\_0 **332**

C0SLOT1\_1 **332**

C0SLOT1\_2 **333**

C0SLOT1\_3 **333**

C0SLOT1\_4 **334**

C0SLOT1\_5 **334**

C0SLOT1\_6 to C0SLOT1\_13 **335**

C0SLOT1\_14 **335**

C0SLOT1\_15 **335**

C0SLPR **300**

C0SSCTLR **318**

C0SSSTR **319**

C0STR **301**

C0TEC **309**

C0TSR **308**

C1AFS **336**

C1BRP **307**

C1CONR **305**

C1CTLR0 **296**

C1CTLR1 **299**

C1EFR **315**

C1EIMKR **313**

C1EISTR **314**

C1GMR0 **320**

C1GMR1 **321**

C1GMR2 **322**

C1GMR3 **323**

C1GMR4 **324**

C1IDR **304**

C1LMAR0 **320**

C1LMAR1 **321**

C1LMAR2 **322**

C1LMAR3 **323**

C1LMAR4 **324**

C1LMBR0 **320**

C1LMBR1 **321**

C1LMBR2 **322**

C1LMBR3 **323**

C1LMBR4	324	C2LMAR1	321
C1MCTL0 to C1MCTL15	327	C2LMAR2	322
C1MDR	316	C2LMAR3	323
C1REC	309	C2LMAR4	324
C1SBS	331	C2LMBR0	320
C1SIMKR	312	C2LMBR1	321
C1SISTR	310	C2LMBR2	322
C1SLOT0_0	332	C2LMBR3	323
C1SLOT0_1	332	C2LMBR4	324
C1SLOT0_2	333	C2MCTL0 to C2MCTL15	327
C1SLOT0_3	333	C2MDR	316
C1SLOT0_4	334	C2REC	309
C1SLOT0_5	334	C2SBS	331
C1SLOT0_6 to C1SLOT0_13	335	C2SIMKR	312
C1SLOT0_14	335	C2SISTR	310
C1SLOT0_15	335	C2SLOT0_0	332
C1SLOT1_0	332	C2SLOT0_1	332
C1SLOT1_1	332	C2SLOT0_2	333
C1SLOT1_2	333	C2SLOT0_3	333
C1SLOT1_3	333	C2SLOT0_4,	334
C1SLOT1_4	334	C2SLOT0_5	334
C1SLOT1_5	334	C2SLOT0_6 to C2SLOT0_13	335
C1SLOT1_6 to C1SLOT1_13	335	C2SLOT0_14	335
C1SLOT1_14	335	C2SLOT0_15	335
C1SLOT1_15	335	C2SLOT1_0	332
C1SLPR	300	C2SLOT1_1	332
C1SSCTLR	318	C2SLOT1_2	333
C1SSSTR	319	C2SLOT1_3	333
C1STR	301	C2SLOT1_4	334
C1TEC	309	C2SLOT1_5	334
C1TSR	308	C2SLOT1_6 to C2SLOT1_13	335
C2AFS	336	C2SLOT1_14	335
C2BRP	307	C2SLOT1_15	335
C2CONR	305	C2SLPR	300
C2CTLR0	296	C2SSCTLR	318
C2CTLR1	299	C2SSSTR	319
C2EFR	315	C2STR	301
C2EIMKR	313	C2TEC	309
C2EISTR	314	C2TSR	308
C2GMR0	320	CCS	281
C2GMR1	321	CM0	56, 105
C2GMR2	322	CM1	57
C2GMR3	323	CM2	59
C2GMR4	324	CPSRF	60
C2IDR	304	CRCD	240
C2LMAR0	320	CRCIN	240

**D**

DA0, DA1 **239**  
 DACON **239**  
 DCT0 to DCT3 **112**  
 DM0SL to DM3SL **109**  
 DMA0 to DMA3 **113**  
 DMD0, DMD1 **111**  
 DRA0 to DRA3 **113**  
 DRC0 to DRC3 **112**  
 DSA0 to DSA3 **113**  
 DTT **159**

**F**

FMR0 **374**  
 FMR1 **375**

**G**

G0CMP0 to G0CMP3 **280**  
 G0CR, G1CR **273**  
 G0DR, G1DR **279**  
 G0EMR **275**  
 G0ERC, G1ERC **277**  
 G0ETC **276**  
 G0IRF **278**  
 G0MR **274**  
 G0MSK0, G0MSK1 **280**  
 G0RB, G1RB **273**  
 G0RCRC, G1RCRC **280**  
 G0RI, G1RI **272**  
 G0TB, G1TB **279**  
 G0TCRC, G1TCRC **280**  
 G0TO, G1TO **272**  
 G1BCR0 **248**  
 G1BCR1 **249**  
 G1BT **248**  
 G1CMP0 to G1CMP3 **280**  
 G1EMR **275**  
 G1ETC **276**  
 G1FE **253**  
 G1FS **252**  
 G1IRF **279**  
 G1MR **274**  
 G1MSK0, G1MSK1 **280**  
 G1PO0 to G1PO7 **252**

G1POCR0 to G1POCR7 **251**  
 G1TM0 to G1TM7 **251**  
 G1TMCR0 to G1TMCR7 **250**  
 G1TPR6, G1TPR7 **250**

**I**

ICTB2 **160**  
 IDB0, IDB1 **159**  
 IFSR **96, 174**  
 IIO0IE to IIO6IE, IIO8IE to IIO11IE **102**  
 IIO0IR to IIO6IR, IIO8IR to IIO11IR **101**  
 Interrupt Control **87, 88**  
 INVC0 **157**  
 INVC1 **158**  
 IPS **363**  
 IPSA **364**

**M**

MCD **58**

**O**

ONSF **131**

**P**

P0 to P15 **352**  
 PCR **363**  
 PD0 to PD15 **351**  
 PLC0 **61**  
 PLC1 **61**  
 PM0 **51**  
 PM1 **52**  
 PM2 **62**  
 PRCR **78**  
 PS0 **353**  
 PS1 **353**  
 PS2 **354**  
 PS3 **354**  
 PS5 **355**  
 PS8 **355**  
 PS9 **356**  
 PSC **359**  
 PSC2 **359**  
 PSC3 **360**  
 PSD1 **360**  
 PSL0 **357**

PSL1 **357**  
PSL2 **358**  
PSL3 **358**  
PUR0 **361**  
PUR1 **361**  
PUR2 **361**  
PUR3 **362**  
PUR4 **362**

## **R**

RLVL **89, 119**  
RMAD0 to RMAD7 **98**  
ROMCP **372**

## **T**

TA0 to TA4 **129**  
TA0MR to TA4MR **130, 135, 138, 141, 143**  
TA1, TA2, TA4, TA11, TA21, TA41 **160**  
TA1MR, TA2MR, TA4MR **162**  
TABSR **130, 146, 161**  
TB0 to TB5 **145**  
TB0MR to TB5MR **146, 148, 150, 152**  
TB2 **161**  
TB2MR **162**  
TB2SC **160**  
TBSR **147**  
TCSPR **60, 132**  
TRGSR **132, 161**

## **U**

U0BRG to U4BRG **168**  
U0C0 to U4C0 **169**  
U0C1 to U4C1 **170**  
U0MR to U4MR **168**  
U0RB to U4RB **167**  
U0SMR to U4SMR **170**  
U0SMR2 to U4SMR2 **171**  
U0SMR3 to U4SMR3 **172**  
U0SMR4 to U4SMR4 **173**  
U0TB to U4TB **167**  
UDF **131**

## **W**

WDC **48, 104**  
WDTS **104**

## **X**

X0R to X15R **242**  
XYC **242**

## **Y**

Y0R to Y15R **242**



---

**RENESAS 16/32-BIT SINGLE-CHIP MICROCOMPUTER  
HARDWARE MANUAL  
M32C/88 Group (M32C/88T)**

**Publication Data : Rev.1.10 Oct. 18, 2005**

**Published by : Sales Strategic Planning Div.  
Renesas Technology Corp.**

---

# M32C/88 Group (M32C/88T) Hardware Manual



**Renesas Electronics Corporation**

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0162-0110