REJ09B0534-0100

Everywhere you imagine. RENESAS

32

# R32C/118 Group
## Hardware Manual

RENESAS MCU
M16C FAMILY / R32C/100 SERIES

Hardware Manual

**Note:** All information contained in this document is provided on an “as is” basis. The information on products or specifications is subject to change without notice. Please verify the latest information available on our website prior to use.

Rev.1.00
Revision Date: Nov 24, 2009

RenesasTechnology
www.renesas.com

## Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com )
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

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

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

1. Handling of Unused Pins

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

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

2. Processing at Power-on

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

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

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

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

# About This Manual

## 1. Purpose and Target User

This manual is designed to be read primarily by application developers who have an understanding of this microcomputer (MCU) including its hardware functions and electrical characteristics. The user should have a basic understanding of electric circuits, logic circuits and, MCUs.
This manual consists of 29 chapters covering six main categories: Overview, CPU, System Control, Peripherals, Electrical Characteristics, and Usage Notes.

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

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

The R32C/118 Group includes the documents listed below. Verify this manual is the latest version by visiting the Renesas Technology website.

| Type of Document | Contents | Document Name | Document Number |
|---|---|---|---|
| Datasheet | Overview of Hardware and Electrical Characteristics | R32C/118 Group Datasheet | REJ03B0255 |
| Hardware Manual | Specifications and detailed descriptions of:<br>-pin layout<br>-memory map<br>-peripherals<br>-electrical characteristics<br>-timing characteristics<br>Refer to the Application Manual for peripheral usage. | R32C/118 Group Hardware Manual | This publication |
| Software Manual | Descriptions of instruction set | R32C/100 Series Software Manual | REJ09B0267 |
| Application Manual | -Usages<br>-Applications<br>-Sample programs<br>-Programing technics using Assembly language or C programming language | Available on the Renesas Technology website. | |
| RENESAS TECHNICAL UPDATE | Bulletins on product specifications, documents, etc. | | |

## 2. Numbers and Symbols

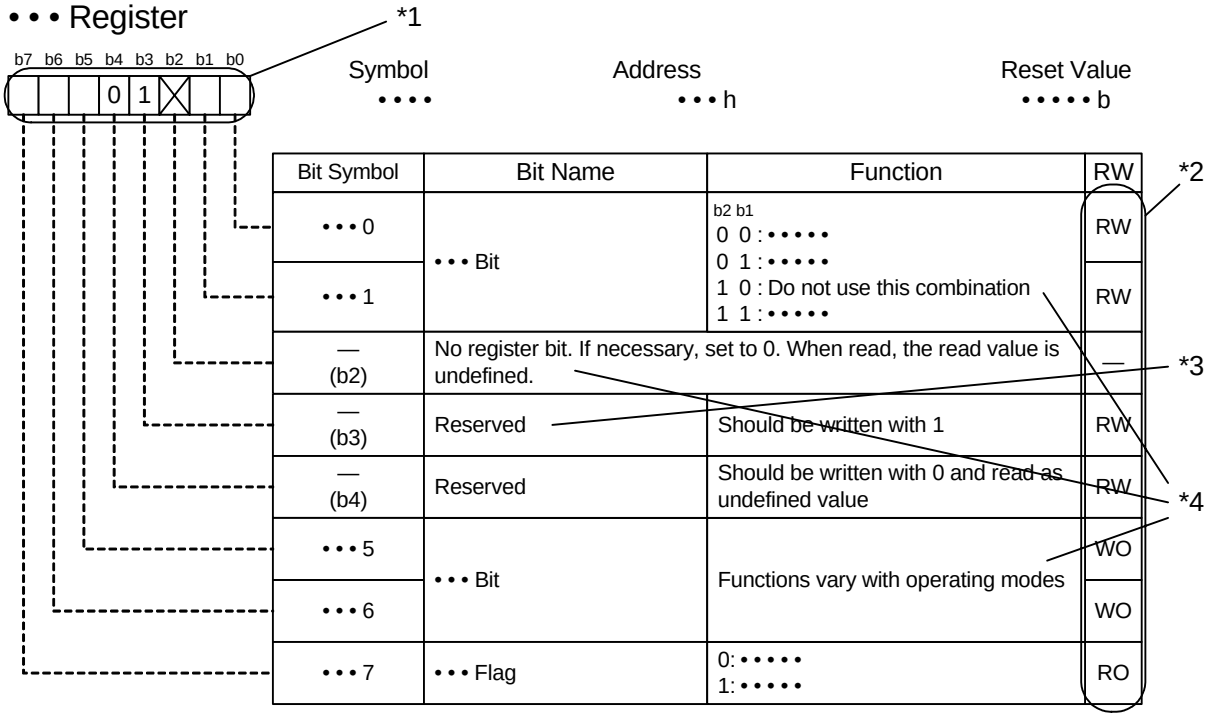
The following explains the denotations used in this manual for registers, bits, pins and various numbers.

(1) Registers, bits, and pins
Registers, bits, and pins are indicated by symbols. Each symbol has a register/bit/pin identifier after the symbol.
Example: PM03 bit in the PM0 register
P3_5 pin, VCC pin

(2) Numbers
A binary number has the suffix “b” except for a 1-bit value.
A hexadecimal number has the suffix “h”.
A decimal number has no suffix.
Example: Binary notation: 11b
Hexadecimal notation: EFA0h
Decimal notation: 1234

# 3. Registers

The following illustration describes registers used throughout this manual.




*1
Blank box: Set this bit to 0 or 1 according to the function.
0: Set this bit to 0.
1: Set this bit to 1.
X: Nothing is assigned to this bit.

*2
RW: Read and write
RO: Read only
WO: Write only (the read value is undefined)
—: Not applicable

*3
- Reserved bit: This bit field is reserved. Set this bit to a specified value. For RW bits, the written value is read unless otherwise noted.

*4
- No register bit(s): No register bit(s) is/are assigned to this field. If necessary, set to 0 for possible future implementation.
- Do not use this combination: Proper operation is not guaranteed when this value is set.
- Functions vary with operating modes: Functions vary with peripheral operating modes. Refer to register illustrations of the respective mode.

# 4. Abbreviations and Acronyms

The following acronyms and terms are used throughout this manual.

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

# TABLE OF CONTENTS


1. Overview 1

1.1 Features ... 1

1.1.1 Applications ... 1

1.1.2 Performance Overview ... 2

1.2 Product Information ... 6

1.3 Block Diagram ... 8

1.4 Pin Assignments ... 9

1.5 Pin Definitions and Functions ... 18

2. Central Processing Unit (CPU) 23

2.1 General Purpose Registers ... 24

2.1.1 Data Registers (R2R0, R3R1, R6R4, and R7R5) ... 24

2.1.2 Address Registers (A0, A1, A2, and A3) ... 24

2.1.3 Static Base Register (SB) ... 24

2.1.4 Frame Base Register (FB) ... 24

2.1.5 Program Counter (PC) ... 24

2.1.6 Interrupt Vector Table Base Register (INTB) ... 24

2.1.7 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP) ... 24

2.1.8 Flag Register (FLG) ... 24

2.2 Fast Interrupt Registers ... 26

2.2.1 Save Flag Register (SVF) ... 26

2.2.2 Save PC Register (SVP) ... 26

2.2.3 Vector Register (VCT) ... 26

2.3 DMAC-associated Registers ... 26

2.3.1 DMA Mode Registers (DMD0, DMD1, DMD2, and DMD3) ... 26

2.3.2 DMA Terminal Count Registers (DCT0, DCT1, DCT2, and DCT3) ... 26

2.3.3 DMA Terminal Count Reload Registers (DCR0, DCR1, DCR2, and DCR3) ... 26

2.3.4 DMA Source Address Registers (DSA0, DSA1, DSA2, and DSA3) ... 26

2.3.5 DMA Source Address Reload Registers (DSR0, DSR1, DSR2, and DSR3) ... 26

2.3.6 DMA Destination Address Registers (DDA0, DDA1, DDA2, and DDA3) ... 26

2.3.7 DMA Destination Address Reload Registers (DDR0, DDR1, DDR2, and DDR3) ... 26

3. Memory 27

4. Special Function Registers (SFRs) 28

5. Resets 81

5.1 Hardware Reset ... 81

5.2 Software Reset ... 84

5.3 Watchdog Timer Reset ... 84

5.4 Reset Vector ... 85

6. Power Management 86
6.1 Voltage Regulators for Internal Logic ... 86
6.1.1 Decoupling Capacitor ... 87
6.2 Low Voltage Detector ... 88
6.2.1 Operational State of Low Voltage Detector ... 91
6.2.2 Low Voltage Detection Interrupt ... 91
6.2.3 An Application of Low Voltage Detector ... 92
7. Processor Mode 93
7.1 Types of Processor Mode ... 93
7.2 Processor Mode Setting ... 93
8. Clock Generator 96
8.1 Clock Generator Types ... 96
8.1.1 Main Clock ... 105
8.1.2 Sub Clock (fC) ... 106
8.1.3 PLL Clock ... 107
8.1.4 On-chip Oscillator Clock ... 110
8.2 Oscillator Stop Detection ... 111
8.2.1 How to Use Oscillator Stop Detection ... 111
8.3 Base Clock ... 111
8.4 CPU Clock and Peripheral Bus Clock ... 112
8.5 Peripheral Clock ... 112
8.6 Clock Output Function ... 113
8.7 Power Control ... 114
8.7.1 Normal Operating Mode ... 118
8.7.2 Wait Mode ... 119
8.7.3 Stop Mode ... 122
8.8 System Clock Protection ... 124
8.9 Notes on Clock Generator ... 125
8.9.1 Sub Clock ... 125
8.9.2 Power Control ... 125
9. Bus 126
9.1 Bus Setting ... 126
9.2 Peripheral Bus Timing Setting ... 127
9.3 External Bus Setting ... 128
9.3.1 External Address Space Setting ... 128
9.3.2 External Data Bus Width Setting ... 134
9.3.3 Separate Bus/Multiplexed Bus Selection ... 136
9.3.4 Read and Write Signals ... 139
9.3.5 External Bus Timing ... 141

9.3.6 ALE Signal ... 145
9.3.7 $\overline{\text{RDY}}$ Signal ... 146
9.3.8 $\overline{\text{HOLD}}$ Signal ... 149
9.3.9 BCLK Output ... 149
9.4 External Bus Status when Accessing Internal Space ... 149
9.5 Notes on Bus ... 150
9.5.1 Notes on System Designing ... 150
9.5.2 Notes on Register Settings ... 150

10. Protection 151
10.1 PRCR Register ... 151
10.2 PRCR2 Register ... 152
10.3 PRCR3 Register ... 152
10.4 PRR Register ... 153

11. Interrupts 154
11.1 Interrupt Types ... 154
11.2 Software Interrupt ... 155
11.3 Hardware Interrupt ... 156
11.3.1 Special Interrupt ... 156
11.3.2 Peripheral Interrupt ... 156
11.4 Fast Interrupt ... 157
11.5 Interrupt Vectors ... 157
11.5.1 Fixed Vector Table ... 158
11.5.2 Relocatable Vector Table ... 158
11.6 Interrupt Request Acceptance ... 163
11.6.1 I Flag and IPL ... 163
11.6.2 Interrupt Control Register ... 164
11.6.3 Wake-up IPL Setting Register ... 167
11.6.4 Interrupt Sequence ... 168
11.6.5 Interrupt Response Time ... 169
11.6.6 IPL After Interrupt Request Acceptance ... 170
11.6.7 Register Saving ... 170
11.7 Register Restoring from Interrupt Handler ... 171
11.8 Interrupt Priority ... 171
11.9 Priority Resolver ... 171
11.10 External Interrupt ... 173
11.11 NMI ... 174
11.12 Key Input Interrupt ... 175
11.13 Intelligent I/O Interrupt ... 176
11.14 Notes on Interrupts ... 179
11.14.1 ISP Setting ... 179

11.14.2 NMI ..... 179
11.14.3 External Interrupt ..... 179

12. Watchdog Timer 180

13. DMAC 182
13.1 Transfer Cycle ..... 191
13.1.1 Effect of Transfer Address and Data Bus Width ..... 191
13.1.2 Effect of Bus Timing ..... 192
13.1.3 Effect of $\overline{\text{RDY}}$ Signal ..... 192
13.2 DMA Transfer Cycle ..... 194
13.3 Channel Priority and DMA Transfer Timing ..... 195
13.4 Notes on DMAC ..... 196
13.4.1 DMAC-associated Register Settings ..... 196
13.4.2 Read from DMAC-associated Registers ..... 196

14. DMAC II 197
14.1 DMAC II Settings ..... 197
14.1.1 Registers RIPL1 and RIPL2 ..... 198
14.1.2 DMAC II Index ..... 199
14.1.3 Interrupt Control Register of the Peripheral Function ..... 202
14.1.4 Relocatable Vector Table of the Peripheral Function ..... 202
14.1.5 IRLT Bit in the IIOiIE Register (i = 0 to 11) ..... 202
14.2 DMAC II Performance ..... 202
14.3 Transfer Types ..... 202
14.3.1 Memory-to-memory Transfer ..... 202
14.3.2 Immediate Data Transfer ..... 203
14.3.3 Calculation Transfer ..... 203
14.4 Transfer Modes ..... 203
14.4.1 Single Transfer ..... 203
14.4.2 Burst Transfer ..... 203
14.4.3 Multiple Transfer ..... 203
14.5 Chained Transfer ..... 204
14.6 DMA II Transfer Complete Interrupt ..... 204
14.7 Execution Time ..... 205

15. Programmable I/O Ports 206
15.1 Port Pi Register (Pi register, i = 0 to 15) ..... 208

16. Timers 209
16.1 Timer A ..... 211
16.1.1 Timer Mode ..... 218
16.1.2 Event Counter Mode ..... 220

16.1.3 One-shot Timer Mode ... 224
16.1.4 Pulse-width Modulation Mode ... 226
16.2 Timer B ... 229
16.2.1 Timer Mode ... 232
16.2.2 Event Counter Mode ... 234
16.2.3 Pulse Period/Pulse-width Measure Mode ... 236
16.3 Notes on Timers ... 239
16.3.1 Timer A and Timer B ... 239
16.3.2 Timer A ... 239
16.3.3 Timer B ... 241

17. Three-phase Motor Control Timers 242
17.1 Modulation Modes of Three-phase Motor Control Timers ... 249
17.2 Timer B2 ... 250
17.3 Timers A4, A1, and A2 ... 252
17.4 Simultaneous Conduction Prevention and Dead Time Timer ... 255
17.5 Three-phase Motor Control Timer Operation ... 256
17.6 Notes on Three-phase Motor Control Timers ... 259
17.6.1 Shutdown ... 259
17.6.2 Register setting ... 259

18. Serial Interface 260
18.1 Synchronous Serial Interface Mode ... 277
18.1.1 Reset Procedure on Transmit/Receive Error ... 282
18.1.2 CLK Polarity ... 282
18.1.3 LSB First and MSB First Selection ... 283
18.1.4 Continuous Receive Mode ... 283
18.1.5 Serial Data Logical Inversion ... 284
18.1.6 CTS/RTS Function ... 284
18.2 Asynchronous Serial Interface Mode (UART Mode) ... 285
18.2.1 Bit Rate ... 290
18.2.2 Reset Procedure on Transmit/Receive Error ... 291
18.2.3 LSB First and MSB First Selection ... 291
18.2.4 Serial Data Logical Inversion ... 292
18.2.5 TXD and RXD I/O Polarity Inversion ... 293
18.2.6 CTS/RTS Function ... 293
18.3 Special Mode 1 ($I^2C$ Mode) ... 294
18.3.1 Start Condition and Stop Condition Detection ... 300
18.3.2 Start Condition and Stop Condition Generation ... 300
18.3.3 Arbitration ... 301
18.3.4 SCL Control and Clock Synchronization ... 302
18.3.5 SDA Output ... 304

18.3.6 SDA Input ..... 304
18.3.7 Acknowledge ..... 304
18.3.8 Initialization of Transmit/Receive Operation ..... 304
18.4 Special Mode 2 ..... 305
18.4.1 $\overline{SSi}$ Input Pin Function (i = 0 to 6) ..... 307
18.4.2 Clock Phase Setting ..... 308
18.5 Notes on Serial Interface ..... 310
18.5.1 Changing the UiBRG Register (i = 0 to 8) ..... 310
18.5.2 Synchronous Serial Interface Mode ..... 310
18.5.3 Special Mode 1 ($I^2C$ Mode) ..... 310

19. A/D Converter 311
19.1 Mode Descriptions ..... 319
19.1.1 One-shot Mode ..... 319
19.1.2 Repeat Mode ..... 320
19.1.3 Single Sweep Mode ..... 321
19.1.4 Repeat Sweep Mode 0 ..... 322
19.1.5 Repeat Sweep Mode 1 ..... 323
19.1.6 Multi-port Single Sweep Mode ..... 324
19.1.7 Multi-port Repeat Sweep Mode 0 ..... 325
19.2 Functions ..... 326
19.2.1 Resolution Selection ..... 326
19.2.2 Sample and Hold Function ..... 326
19.2.3 Trigger Selection ..... 326
19.2.4 DMAC Operating Mode ..... 326
19.2.5 Function-extended Analog Input Pins ..... 327
19.2.6 External Operating Amplifier (Op-AMP) Connection Mode ..... 327
19.2.7 Power Saving ..... 328
19.2.8 Output Impedance of Sensor Equivalent Circuit under A/D Conversion ..... 328
19.3 Notes on A/D Converter ..... 330
19.3.1 Notes on Designing Boards ..... 330
19.3.2 Notes on Programming ..... 331

20. D/A Converter 332

21. CRC Calculator 334

22. X-Y Conversion 337
22.1 Data Conversion on Reading ..... 338
22.2 Data Conversion on Writing ..... 340

23. Intelligent I/O 341
23.1 Base Timer (for Groups 0 to 2) ..... 356

23.2 Time Measurement (for Groups 0 and 1) ... 362
23.3 Waveform Generation (for Groups 0 to 2) ... 366
23.3.1 Single-phase Waveform Output Mode (for Groups 0 to 2) ... 367
23.3.2 Inverted Waveform Output Mode (for Groups 0 to 2) ... 369
23.3.3 Set/Reset Waveform Output Mode (SR Waveform Output Mode) (for Groups 0 to 2) ... 371
23.3.4 Bit Modulation PWM Output Mode (for Group 2) ... 374
23.3.5 Real-Time Port Output Mode (RTP Output Mode) (for Group 2) ... 376
23.3.6 Parallel Real-Time Port Output Mode (RTP Output Mode) (for Group 2) ... 378
23.4 Group 2 Serial Interface ... 380
23.4.1 Variable Synchronous Serial Interface Mode (for Group 2) ... 385

24. Multi-master I2C-bus Interface 388
24.1 Multi-master I2C-bus Interface-associated Registers ... 390
24.1.1 I2C-bus Transmit/Receive Shift Register (I2CTRSR) ... 390
24.1.2 I2C-bus Slave Address Register (I2CSAR) ... 391
24.1.3 I2C-bus Control Register 0 (I2CCR0) ... 392
24.1.4 I2C-bus Clock Control Register (I2CCCR) ... 394
24.1.5 I2C-bus START Condition and STOP Condition Control Register (I2CSSCR) ... 396
24.1.6 I2C-bus Control Register 1 (I2CCR1) ... 397
24.1.7 I2C-bus Control Register 2 (I2CCR2) ... 400
24.1.8 I2C-bus Status Register (I2CSR) ... 402
24.1.9 I2C-bus Mode Register (I2CMR) ... 406
24.2 The Generating a START Condition ... 407
24.3 Generating a STOP Condition ... 409
24.4 START Condition Redundancy Prevention Function ... 410
24.5 Detecting START Condition and STOP Condition ... 411
24.6 Data Transmission and Reception ... 413
24.6.1 Master Transmission ... 414
24.6.2 Slave Reception ... 415
24.7 Notes on Using Multi-master I2C-bus Interface ... 416
24.7.1 Accessing Multi-master I2C-bus Interface-Associated Registers ... 416
24.7.2 Generating a Repeated START condition ... 418

25. CAN Module 419
25.1 CAN SFRs ... 422
25.1.1 CANi Control Register (CiCTLR Register) (i = 0, 1) ... 423
25.1.2 CANi Clock Select Register (CiCLKR Register) (i = 0, 1) ... 427
25.1.3 CANi Bit Configuration Register (CiBCR Register) (i = 0, 1) ... 428
25.1.4 CANi Mask Register k (CiMKRk Register) (i = 0, 1; k = 0 to 7) ... 430
25.1.5 CANi FIFO Received ID Compare Register n (Registers CiFIDCR0 and CiFIDCR1) (i = 0, 1; n = 0, 1) ... 431
25.1.6 CANi Mask Invalid Register (CiMKIVLR Register) (i = 0, 1) ... 433

25.1.7 CANi Mailbox (CiMBj Register) (i = 0, 1; j = 0 to 31) ..... 434
25.1.8 CANi Mailbox Interrupt Enable Register (CiMIER Register) (i = 0, 1) ..... 438
25.1.9 CANi Message Control Register (CiMCTLj Register) (i = 0, 1; j = 0 to 31) ..... 439
25.1.10 CANi Receive FIFO Control Register (CiRFCR Register) (i = 0, 1) ..... 443
25.1.11 CANi Receive FIFO Pointer Control Register (CiRFPCR Register) (i = 0, 1) ..... 446
25.1.12 CANi Transmit FIFO Control Register (CiTFCR Register) (i = 0, 1) ..... 447
25.1.13 CANi Transmit FIFO Pointer Control Register (CiTFPCR Register) (i = 0, 1) ..... 449
25.1.14 CANi Status Register (CiSTR Register) (i = 0, 1) ..... 450
25.1.15 CANi Mailbox Search Mode Register (CiMSMR Register) (i = 0, 1) ..... 453
25.1.16 CANi Mailbox Search Status Register (CiMSSR Register) (i = 0, 1) ..... 454
25.1.17 CANi Channel Search Support Register (CiCSSR Register) (i = 0, 1) ..... 456
25.1.18 CANi Acceptance Filter Support Register (CiAFSR Register) (i = 0, 1) ..... 457
25.1.19 CANi Error Interrupt Enable Register (CiEIER Register) (i = 0, 1) ..... 458
25.1.20 CANi Error Interrupt Factor Judge Register (CiEIFR Register) (i = 0, 1) ..... 460
25.1.21 CANi Receive Error Count Register (CiRECR Register) (i = 0, 1) ..... 463
25.1.22 CANi Transmit Error Count Register (CiTECR Register) (i = 0, 1) ..... 464
25.1.23 CANi Error Code Store Register (CiECSR Register) (i = 0, 1) ..... 465
25.1.24 CANi Time Stamp Register (CiTSR Register) (i = 0, 1) ..... 467
25.1.25 CANi Test Control Register (CiTCR Register) (i = 0, 1) ..... 468
25.2 Operating Mode ..... 471
25.2.1 CAN Reset Mode ..... 472
25.2.2 CAN Halt Mode ..... 473
25.2.3 CAN Sleep Mode ..... 474
25.2.4 CAN Operation Mode (Excluding Bus-Off State) ..... 475
25.2.5 CAN Operation Mode (Bus-Off State) ..... 476
25.3 CAN Communication Speed Configuration ..... 477
25.3.1 CAN Clock Configuration ..... 477
25.3.2 Bit Timing Configuration ..... 477
25.3.3 Bit rate ..... 478
25.4 Mailbox and Mask Register Structure ..... 479
25.5 Acceptance Filtering and Masking Function ..... 481
25.6 Reception and Transmission ..... 484
25.6.1 Reception ..... 485
25.6.2 Transmission ..... 487
25.7 CAN Interrupt ..... 488
26. I/O Pins 489
26.1 Port Pi Direction Register (PDi Register, i = 0 to 15) ..... 490
26.2 Output Function Select Register ..... 491
26.3 Input Function Select Register ..... 509
26.4 Pull-up Control Registers 0 to 4 (Registers PUR0 to PUR4) ..... 514

26.5 Port Control Register (PCR Register) ... 517
26.6 How To Configure Unused Pins ... 518

27. Flash Memory 521
27.1 Overview ... 521
27.2 Flash Memory Protection ... 523
27.2.1 Lock Bit Protection ... 523
27.2.2 ROM Code Protection ... 523
27.2.3 ID Code Protection ... 524
27.3 CPU Rewrite Mode ... 526
27.3.1 CPU Operating Mode and Flash Memory Rewrite ... 534
27.3.2 Flash Memory Rewrite Bus Timing ... 535
27.3.3 Software Commands ... 539
27.3.4 Mode Transition ... 540
27.3.5 How to Issue Software Commands ... 541
27.3.6 Status Check ... 547
27.4 Standard Serial I/O Mode ... 548
27.5 Parallel I/O mode ... 551
27.6 Notes on Flash Memory Rewriting ... 552
27.6.1 Note on Power Supply ... 552
27.6.2 Note on Hardware Reset ... 552
27.6.3 Note on Flash Memory Protection ... 552
27.6.4 Notes on Programming ... 552
27.6.5 Notes on Interrupts ... 552
27.6.6 Notes on Rewrite Control Program ... 553
27.6.7 Notes on Number of Programming/Erasure and Software Command Execution Time ... 553
27.6.8 Other Notes ... 553

28. Electrical Characteristics 554

29. Usage Notes 595
29.1 Notes on Board Designing ... 595
29.1.1 Power Supply Pins ... 595
29.1.2 Supply Voltage ... 595
29.2 Notes on Register Setting ... 596
29.2.1 Registers with Write-only Bits ... 596
29.3 Notes on Clock Generator ... 598
29.3.1 Sub Clock ... 598
29.3.2 Power Control ... 598
29.4 Notes on Bus ... 599
29.4.1 Notes on System Designing ... 599
29.4.2 Notes on Register Settings ... 599

29.5 Notes on Interrupts ..... 600
29.5.1 ISP Setting ..... 600
29.5.2 NMI ..... 600
29.5.3 External Interrupt ..... 600
29.6 Notes on DMAC ..... 601
29.6.1 DMAC-associated Register Settings ..... 601
29.6.2 Read from DMAC-associated Registers ..... 601
29.7 Notes on Timers ..... 602
29.7.1 Timer A and Timer B ..... 602
29.7.2 Timer A ..... 602
29.7.3 Timer B ..... 604
29.8 Notes on Three-phase Motor Control Timers ..... 605
29.8.1 Shutdown ..... 605
29.8.2 Register setting ..... 605
29.9 Notes on Serial Interface ..... 606
29.9.1 Changing the UiBRG Register (i = 0 to 8) ..... 606
29.9.2 Synchronous Serial Interface Mode ..... 606
29.9.3 Special Mode 1 ($I^2C$ Mode) ..... 606
29.10 Notes on A/D Converter ..... 607
29.10.1 Notes on Designing Boards ..... 607
29.10.2 Notes on Programming ..... 608
29.11 Notes on Flash Memory Rewriting ..... 609
29.11.1 Note on Power Supply ..... 609
29.11.2 Note on Hardware Reset ..... 609
29.11.3 Note on Flash Memory Protection ..... 609
29.11.4 Notes on Programming ..... 609
29.11.5 Notes on Interrupts ..... 609
29.11.6 Notes on Rewrite Control Program ..... 610
29.11.7 Notes on Number of Programming/Erasure and Software Command Execution Time ..... 610
29.11.8 Other Notes ..... 610

Appendix 1. Package Dimensions 611

INDEX 612

# 1. Overview

## 1.1 Features

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

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

The R32C/100 Series, in particular, provides the R32C/118 Group as a standard product. This product, provided as a 100/144-pin plastic molded LQFP package, configures nine channels of serial interface, one channel of multi-master $I^2C$-bus interface, and two channels of CAN module.

### 1.1.1 Applications

Car audio, audio, printer, office/industrial equipment etc.

### 1.1.2 Performance Overview

Table 1.1 to Table 1.4 show the performance overview of the R32C/118 Group.

**Table 1.1 R32C/118 Group Performance for the 144 pin-Package (1/2)**

| Unit | Function | Performance |
|---|---|---|
| CPU | Central processing unit | R32C/100 Series CPU Core<br>• Basic instructions: 108<br>• Minimum instruction execution time: 20 ns (f(CPU) = 50 MHz)<br>• Multiplier: 32-bit × 32-bit → 64-bit<br>• Multiply-accumulate unit: 32-bit × 32-bit + 64-bit → 64-bit<br>• IEEE-754 floating point standard: Single precision<br>• 32-bit barrel shifter<br>• Operating mode: Single-chip mode, memory expansion mode, microprocessor mode (optional (1)) |
| Memory | | Flash memory: 384 Kbytes to 1 Mbyte<br>RAM: 40 K/48 K/63 Kbytes<br>Data flash: 4 Kbytes × 2 blocks<br>Refer to Table 1.5 for memory size of each product group |
| Voltage Detector | Low voltage detector | Optional (1)<br>Low voltage detection interrupt |
| Clock | Clock generator | • 4 circuits (main clock, sub clock, PLL, on-chip oscillator)<br>• Oscillation stop detector: Main clock oscillator stop/re-oscillation detection<br>• Frequency divide circuit: Divide-by-2 to divide-by-24 selectable<br>• Low power modes: Wait mode, stop mode |
| External Bus Expansion | Bus and memory expansion | • Address space: 4 Gbytes (of which up to 64 Mbytes is user accessible)<br>• External bus Interface: Support for wait-state insertion, 4 chip select outputs<br>• Bus format: Separate bus/Multiplexed bus selectable, data bus width selectable (8/16/32 bits) |
| Interrupts | | Interrupt vectors: 261<br>External interrupt inputs: $\overline{\text{NMI}}$, $\overline{\text{INT}}$ × 9, key input × 4<br>Interrupt priority levels: 7 levels |
| Watchdog Timer | | 15 bits × 1 (selectable input frequency from prescaler output) |
| DMA | DMAC | 4 channels<br>• Cycle-steal transfer mode<br>• Request sources: 57<br>• 2 transfer modes: Single transfer, repeat transfer |
| | DMAC II | • Can be activated by any peripheral interrupt source<br>• 3 transfer functions: Immediate data transfer, calculation transfer, chained transfer |
| I/O Ports | Programmable I/O ports | • 2 input-only ports<br>• 120 CMOS inputs/outputs<br>• 32 ports are 5 V tolerant<br>• A pull-up resistor is selectable for every 4 input ports (except 5 V tolerant inputs) |

Note:
1. Please contact a Renesas sales office to use the optional feature.

**Table 1.2 R32C/118 Group Performance for the 144-pin Package (2/2)**

| Unit | Function | Performance |
|---|---|---|
| Timer | Timer A | 16-bit timer × 5<br>Timer mode, event counter mode, one-shot timer mode, pulse-width modulation (PWM) mode<br>Two-phase pulse signal processing in event counter mode (two-phase encoder input) × 3 |
| | Timer B | 16-bit timer × 6<br>Timer mode, event counter mode, pulse frequency measurement mode, pulse-width measurement mode |
| | Three-phase motor control timer | Three-phase motor control timer ×1 (timers A1, A2, A4, and B2 used)<br>8-bit programmable dead time timer |
| Serial Interface | UART0 to UART8 | Asynchronous/synchronous serial interface × 9 channels<br>• $I^2C$-bus (UART0 to UART6)<br>• Special mode 2 (UART0 to UART6)<br>• IEBus (1) (optional (2)) (UART0 to UART6) |
| A/D Converter | | 10-bit resolution × 34 channels<br>Sample and hold functionality integrated |
| D/A Converter | | 8-bit resolution × 2 |
| CRC Calculator | | CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) |
| X-Y Converter | | 16 bits × 16 bits |
| Intelligent I/O | | Time measurement (input capture): 16 bits × 16<br>Waveform generation (output compare): 16 bits × 24<br>Serial interface: Variable-length synchronous serial I/O mode, IEBus (1) mode (optional (2)) |
| Multi-master $I^2C$-bus Interface | | 1 channel |
| CAN Module | | 2 channels<br>CAN functionality compliant with ISO11898-1<br>32 mailboxes |
| Flash Memory | | Programming and erasure supply voltage:<br>VCC = 3.0 to 5.5 V<br>Minimum endurance: 1, 000 program/erase cycles<br>Security protection: ROM code protect, ID code protect<br>Debugging: On-chip debug, on-board flash programming |
| Operating Frequency/Supply Voltage | | 50 MHz/VCC = 3.0 to 5.5 V |
| Operating Temperature | | -20°C to 85°C (version N)<br>-40°C to 85°C (version D)<br>-40°C to 85°C (version P) |
| Current Consumption | | 35 mA (VCC = 5.0 V, f(CPU) = 50 MHz)<br>8 µA (VCC = 3.3 V, f(XCIN) = 32.768 kHz, in wait mode) |
| Package | | 144-pin plastic molded LQFP (PLQP0144KA-A) |

Notes:
1. IEBus is a trademark of NEC Electronics Corporation.
2. Please contact a Renesas sales office to use the optional feature.

**Table 1.3 R32C/118 Group Performance for the 100-pin Package (1/2)**

| Unit | Function | Performance |
|---|---|---|
| CPU | Central processing unit | R32C/100 Series CPU Core<br>• Basic instructions: 108<br>• Minimum instruction execution time: 20 ns (f(CPU) = 50 MHz)<br>• Multiplier: 32-bit × 32-bit → 64-bit<br>• Multiply-accumulate unit: 32-bit × 32-bit + 64-bit → 64-bit<br>• IEEE-754 floating point standard: Single precision<br>• 32-bit barrel shifter<br>• Operating mode: Single-chip mode, memory expansion mode, microprocessor mode (optional (1)) |
| Memory | | Flash memory: 384 Kbytes to 1 Mbyte<br>RAM: 40 K/48 K/63 Kbytes<br>Data flash: 4 Kbytes × 2 blocks<br>Refer to Table 1.5 for memory size of each product group |
| Voltage Detector | Low voltage detector | Optional (1)<br>Low voltage detection interrupt |
| Clock | Clock generator | • 4 circuits (main clock, sub clock, PLL, on-chip oscillator)<br>• Oscillation stop detector: Main clock oscillator stop/re-oscillation detection<br>• Frequency divide circuit: Divide-by-2 to divide-by-24 selectable<br>• Low power modes: Wait mode, stop mode |
| External Bus Expansion | Bus and memory expansion | • Address space: 4 Gbytes (of which up to 64 Mbytes is user accessible)<br>• External bus Interface: Support for wait-state insertion, 4 chip select outputs<br>• Bus format: Separate bus/Multiplexed bus selectable, data bus width selectable (8/16 bits) |
| Interrupts | | Interrupt vectors: 261<br>External interrupt inputs: $\overline{\text{NMI}}$, $\overline{\text{INT}}$ × 6, key input × 4<br>Interrupt priority levels: 7 levels |
| Watchdog Timer | | 15 bits × 1 (selectable input frequency from prescaler output) |
| DMA | DMAC | 4 channels<br>• Cycle-steal transfer mode<br>• Request sources: 51<br>• 2 transfer modes: Single transfer, repeat transfer |
| | DMAC II | • Can be activated by any peripheral interrupt source<br>• 3 transfer functions: Immediate data transfer, calculation transfer, chained transfer |
| I/O Ports | Programmable I/O ports | • 2 input-only ports<br>• 84 CMOS inputs/outputs<br>  • 32 ports are 5 V tolerant<br>• A pull-up resistor is selectable for every 4 input ports (except 5 V tolerant inputs) |

Note:
1. Please contact a Renesas sales office to use the optional feature.

**Table 1.4 R32C/118 Group Performance for the 100-pin Package (2/2)**

| Unit | Function | Performance |
|---|---|---|
| Timer | Timer A | 16-bit timer × 5<br>Timer mode, event counter mode, one-shot timer mode, pulse-width modulation (PWM) mode<br>Two-phase pulse signal processing in event counter mode (two-phase encoder input) × 3 |
| | Timer B | 16-bit timer × 6<br>Timer mode, event counter mode, pulse frequency measurement mode, pulse-width measurement mode |
| | Three-phase motor control timer | Three-phase motor control timer ×1 (timers A1, A2, A4, and B2 used)<br>8-bit programmable dead time timer |
| Serial Interface | UART0 to UART8 | Asynchronous/synchronous serial interface × 9 channels<br>• $I^2C$-bus (UART0 to UART6)<br>• Special mode 2 (UART0 to UART6)<br>• IEBus (1) (optional (2)) (UART0 to UART6) |
| A/D Converter | | 10-bit resolution × 26 channels<br>Sample and hold functionality integrated |
| D/A Converter | | 8-bit resolution × 2 |
| CRC Calculator | | CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) |
| X-Y Converter | | 16 bits × 16 bits |
| Intelligent I/O | | Time measurement (input capture): 16 bits × 16<br>Waveform generation (output compare): 16 bits × 19<br>Serial interface: Variable-length synchronous serial I/O mode, IEBus (1) mode (optional (2)) |
| Multi-master $I^2C$-bus Interface | | 1 channel |
| CAN Module | | 2 channels<br>CAN functionality compliant with ISO11898-1<br>32 mailboxes |
| Flash Memory | | Programming and erasure supply voltage:<br>VCC = 3.0 to 5.5 V<br>Minimum endurance: 1, 000 program/erase cycles<br>Security protection: ROM code protect, ID code protect<br>Debugging: On-chip debug, on-board flash programming |
| Operating Frequency/Supply Voltage | | 50 MHz/VCC = 3.0 to 5.5 V |
| Operating Temperature | | -20°C to 85°C (version N)<br>-40°C to 85°C (version D)<br>-40°C to 85°C (version P) |
| Current Consumption | | 35 mA (VCC = 5.0 V, f(CPU) = 50 MHz)<br>8 µA (VCC = 3.3 V, f(XCIN) = 32.768 kHz, in wait mode) |
| Package | | 100-pin plastic molded LQFP (PLQP0100KB-A) |

Notes:
1. IEBus is a trademark of NEC Electronics Corporation.
2. Please contact a Renesas sales office to use the optional feature.

## 1.2 Product Information

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

**Table 1.5 R32C/118 Group Product List As of November, 2009**

| Part Number | Package Code (1) | ROM Capacity (2) | RAM Capacity | Remarks |
|---|---|---|---|---|
| R5F64185NFD (P) | PLQP0144KA-A | 384 Kbytes + 8 Kbytes | 40 Kbytes | -20°C to 85°C (version N) |
| R5F64185DFD | | | | -40°C to 85°C (version D) |
| R5F64185PFD | | | | -40°C to 85°C (version P) |
| R5F64185NFB (P) | PLQP0100KB-A | | | -20°C to 85°C (version N) |
| R5F64185DFB | | | | -40°C to 85°C (version D) |
| R5F64185PFB | | | | -40°C to 85°C (version P) |
| R5F64186NFD (P) | PLQP0144KA-A | 512 Kbytes + 8 Kbytes | | -20°C to 85°C (version N) |
| R5F64186DFD | | | | -40°C to 85°C (version D) |
| R5F64186PFD | | | | -40°C to 85°C (version P) |
| R5F64186NFB (P) | PLQP0100KB-A | | | -20°C to 85°C (version N) |
| R5F64186DFB | | | | -40°C to 85°C (version D) |
| R5F64186PFB | | | | -40°C to 85°C (version P) |
| R5F64187NFD (P) | PLQP0144KA-A | 640 Kbytes + 8 Kbytes | 48Kbytes | -20°C to 85°C (version N) |
| R5F64187DFD | | | | -40°C to 85°C (version D) |
| R5F64187PFD | | | | -40°C to 85°C (version P) |
| R5F64187NFB (P) | PLQP0100KB-A | | | -20°C to 85°C (version N) |
| R5F64187DFB | | | | -40°C to 85°C (version D) |
| R5F64187PFB | | | | -40°C to 85°C (version P) |
| R5F64188NFD (P) | PLQP0144KA-A | 768 Kbytes + 8 Kbytes | 63 Kbytes | -20°C to 85°C (version N) |
| R5F64188DFD | | | | -40°C to 85°C (version D) |
| R5F64188PFD | | | | -40°C to 85°C (version P) |
| R5F64188NFB (P) | PLQP0100KB-A | | | -20°C to 85°C (version N) |
| R5F64188DFB | | | | -40°C to 85°C (version D) |
| R5F64188PFB | | | | -40°C to 85°C (version P) |
| R5F64189NFD (P) | PLQP0144KA-A | 1 Mbyte + 8 Kbytes | | -20°C to 85°C (version N) |
| R5F64189DFD | | | | -40°C to 85°C (version D) |
| R5F64189PFD | | | | -40°C to 85°C (version P) |
| R5F64189NFB (P) | PLQP0100KB-A | | | -20°C to 85°C (version N) |
| R5F64189DFB | | | | -40°C to 85°C (version D) |
| R5F64189PFB | | | | -40°C to 85°C (version P) |

(P): On planning phase

Notes:
1. The old package codes are as follows:PLQP0100KB-A: 100P6Q-A, PLQP0144KA-A: 144P6Q-A
2. Data flash memory provides an additional 8 Kbytes of ROM capacity.

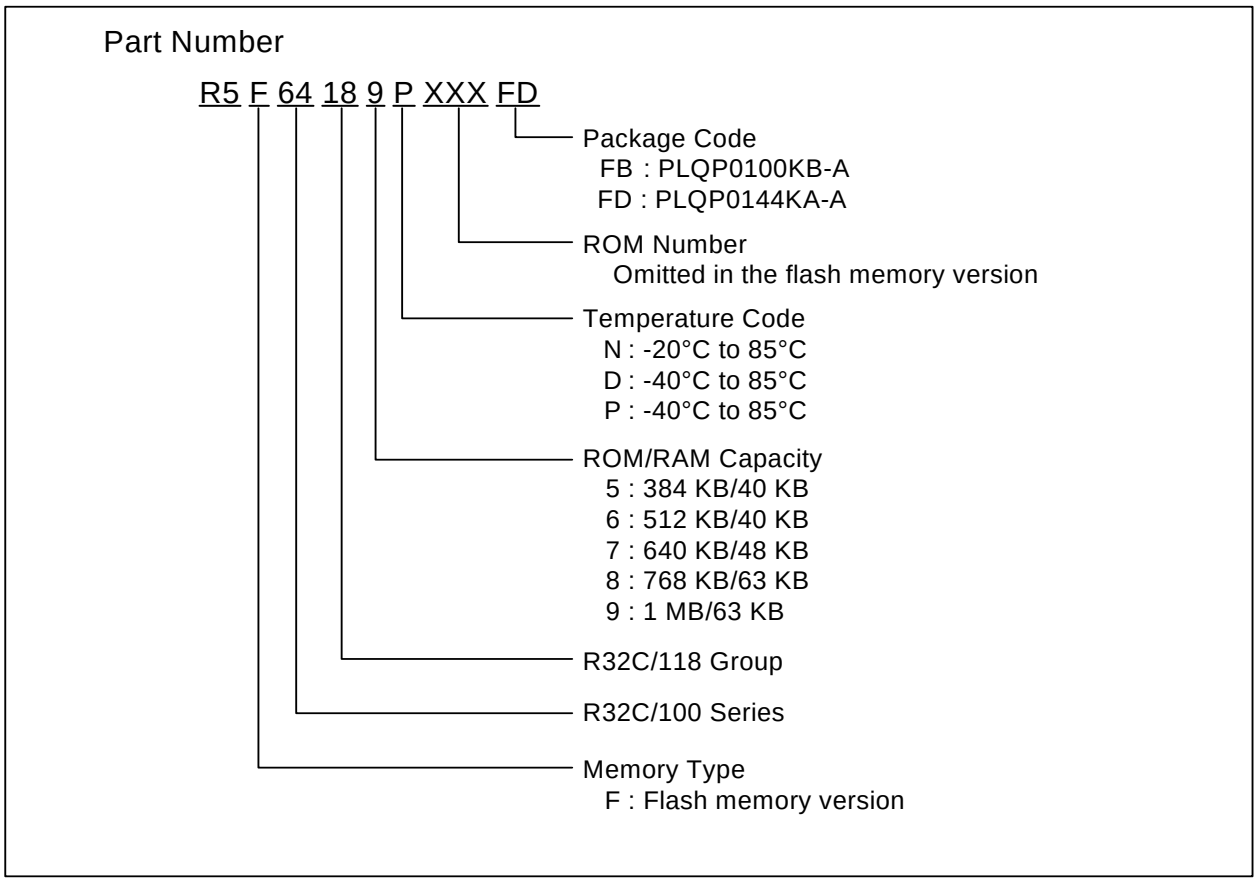
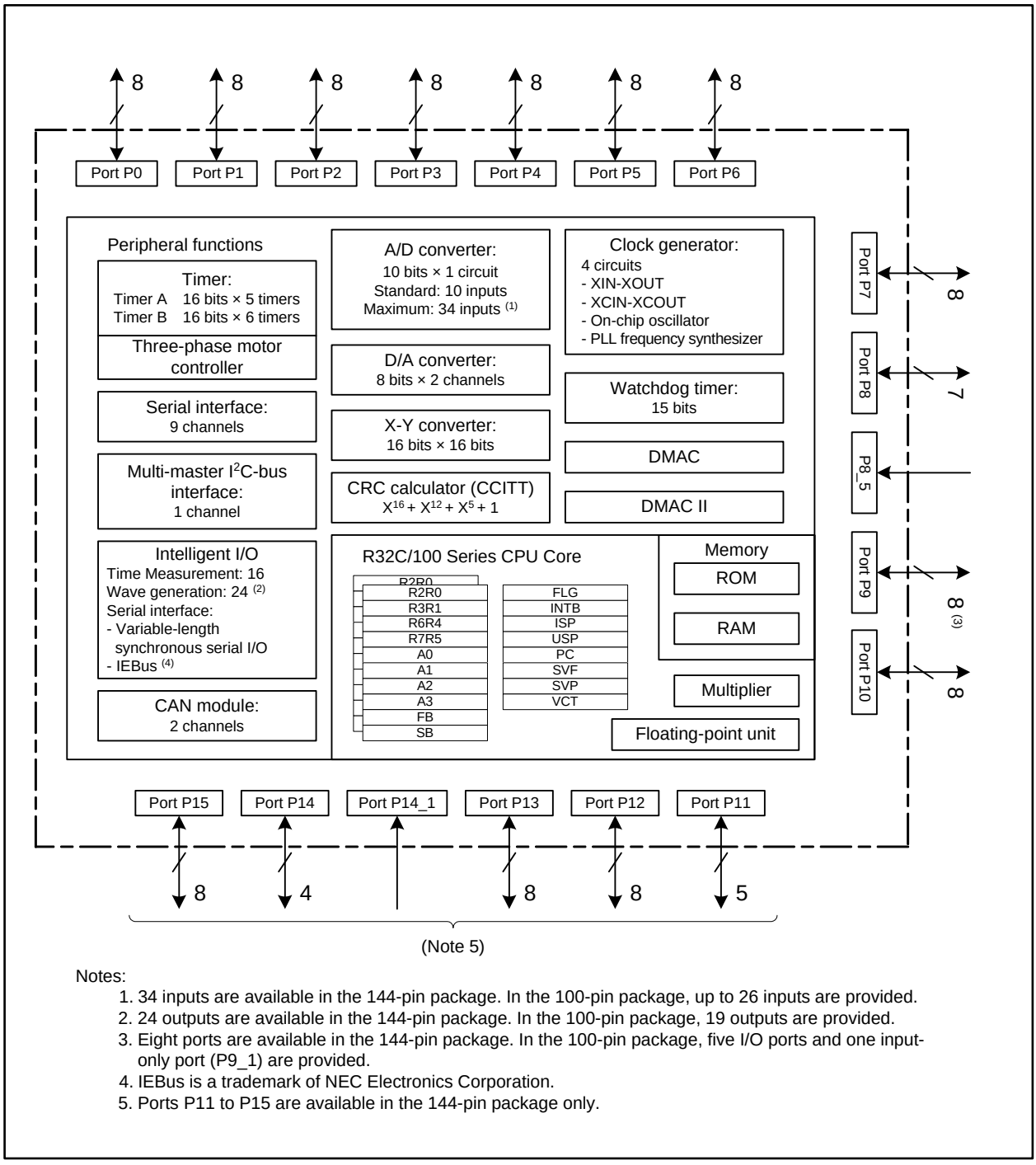



**Figure 1.1 Part Numbering**

## 1.3 Block Diagram

Figure 1.2 shows a block diagram of the R32C/118 Group.

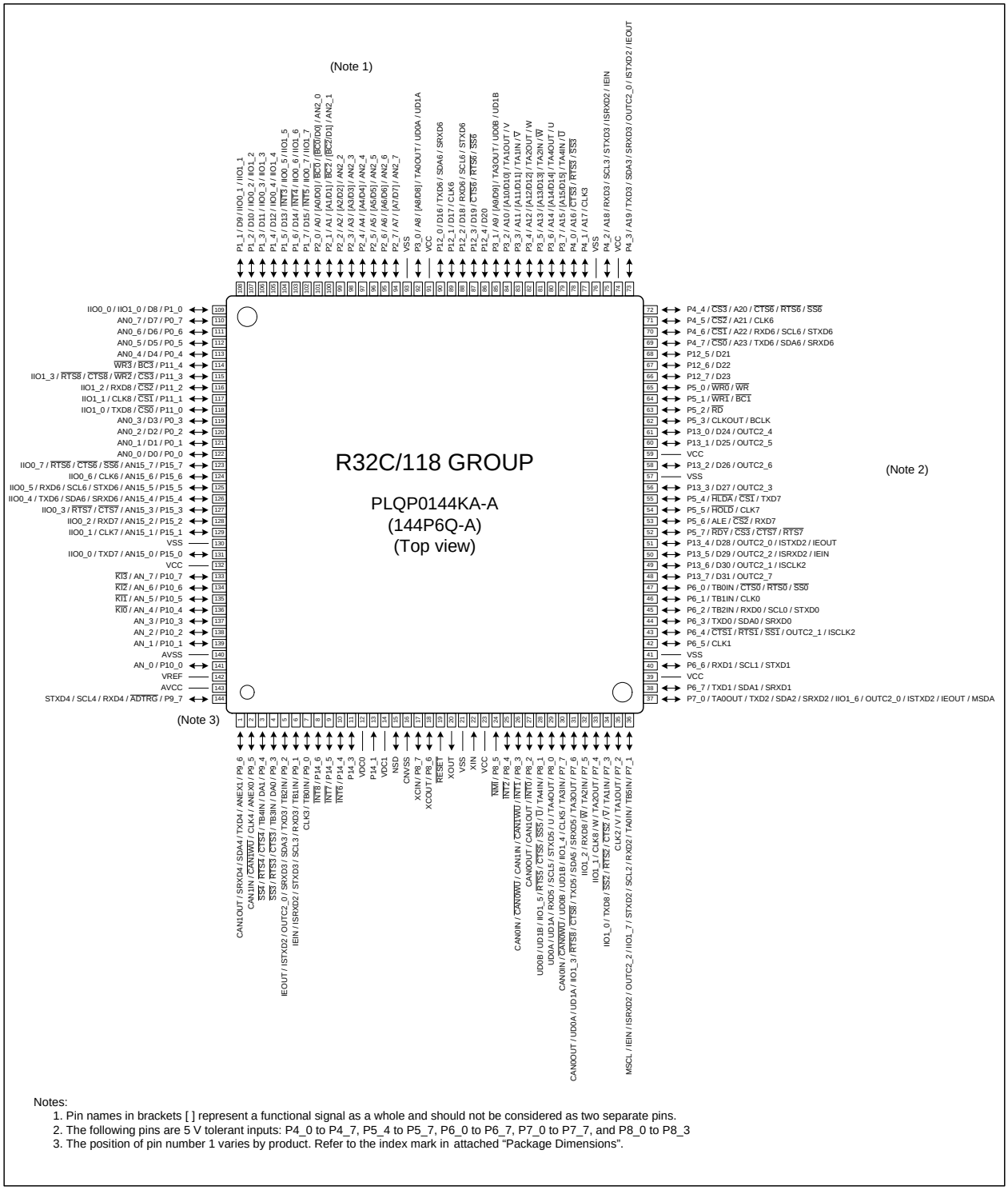



Notes:
1. 34 inputs are available in the 144-pin package. In the 100-pin package, up to 26 inputs are provided.
2. 24 outputs are available in the 144-pin package. In the 100-pin package, 19 outputs are provided.
3. Eight ports are available in the 144-pin package. In the 100-pin package, five I/O ports and one input-only port (P9_1) are provided.
4. IEBus is a trademark of NEC Electronics Corporation.
5. Ports P11 to P15 are available in the 144-pin package only.

**Figure 1.2 R32C/118 Group Block Diagram**

## 1.4 Pin Assignments

Figure 1.3 and Figure 1.4 show the pin assignments (top view) and Table 1.6 to Table 1.12 show the pin characteristics.




Notes:
1. Pin names in brackets [ ] represent a functional signal as a whole and should not be considered as two separate pins.
2. The following pins are 5 V tolerant inputs: P4_0 to P4_7, P5_4 to P5_7, P6_0 to P6_7, P7_0 to P7_7, and P8_0 to P8_3
3. The position of pin number 1 varies by product. Refer to the index mark in attached "Package Dimensions".

**Figure 1.3 Pin Assignment for the 144-pin Package (top view)**

**Table 1.6 Pin Characteristics for the 144-pin Package (1/4)**

| Pin No. | Control Pin | Port | Interrupt Pin | Timer Pin | UART/CAN Module Pin | Intelligent I/O Pin | Analog Pin | Bus Control Pin |
|---|---|---|---|---|---|---|---|---|
| 1 | | P9_6 | | | TXD4/SDA4/SRXD4/ CAN1OUT | | ANEX1 | |
| 2 | | P9_5 | | | CLK4/CAN1IN/ $\overline{\text{CAN1WU}}$ | | ANEX0 | |
| 3 | | P9_4 | | TB4IN | $\overline{\text{CTS4}}$/$\overline{\text{RTS4}}$/$\overline{\text{SS4}}$ | | DA1 | |
| 4 | | P9_3 | | TB3IN | $\overline{\text{CTS3}}$/$\overline{\text{RTS3}}$/$\overline{\text{SS3}}$ | | DA0 | |
| 5 | | P9_2 | | TB2IN | TXD3/SDA3/SRXD3 | OUTC2_0/ISTXD2/ IEOUT | | |
| 6 | | P9_1 | | TB1IN | RXD3/SCL3/STXD3 | ISRXD2/IEIN | | |
| 7 | | P9_0 | | TB0IN | CLK3 | | | |
| 8 | | P14_6 | $\overline{\text{INT8}}$ | | | | | |
| 9 | | P14_5 | $\overline{\text{INT7}}$ | | | | | |
| 10 | | P14_4 | $\overline{\text{INT6}}$ | | | | | |
| 11 | | P14_3 | | | | | | |
| 12 | VDC0 | | | | | | | |
| 13 | | P14_1 | | | | | | |
| 14 | VDC1 | | | | | | | |
| 15 | NSD | | | | | | | |
| 16 | CNVSS | | | | | | | |
| 17 | XCIN | P8_7 | | | | | | |
| 18 | XCOUT | P8_6 | | | | | | |
| 19 | $\overline{\text{RESET}}$ | | | | | | | |
| 20 | XOUT | | | | | | | |
| 21 | VSS | | | | | | | |
| 22 | XIN | | | | | | | |
| 23 | VCC | | | | | | | |
| 24 | | P8_5 | $\overline{\text{NMI}}$ | | | | | |
| 25 | | P8_4 | $\overline{\text{INT2}}$ | | | | | |
| 26 | | P8_3 | $\overline{\text{INT1}}$ | | CAN0IN/$\overline{\text{CAN0WU}}$/ CAN1IN/$\overline{\text{CAN1WU}}$ | | | |
| 27 | | P8_2 | $\overline{\text{INT0}}$ | | CAN0OUT/CAN1OUT | | | |
| 28 | | P8_1 | | TA4IN/$\overline{\text{U}}$ | $\overline{\text{CTS5}}$/$\overline{\text{RTS5}}$/$\overline{\text{SS5}}$ | IIO1_5/UD0B/UD1B | | |
| 29 | | P8_0 | | TA4OUT/U | RXD5/SCL5/STXD5 | UD0A/UD1A | | |
| 30 | | P7_7 | | TA3IN | CLK5/CAN0IN/ $\overline{\text{CAN0WU}}$ | IIO1_4/UD0B/UD1B | | |
| 31 | | P7_6 | | TA3OUT | TXD5/SDA5/SRXD5/ $\overline{\text{CTS8}}$/$\overline{\text{RTS8}}$/CAN0OUT | IIO1_3/UD0A/UD1A | | |
| 32 | | P7_5 | | TA2IN/$\overline{\text{W}}$ | RXD8 | IIO1_2 | | |
| 33 | | P7_4 | | TA2OUT/W | CLK8 | IIO1_1 | | |
| 34 | | P7_3 | | TA1IN/$\overline{\text{V}}$ | $\overline{\text{CTS2}}$/$\overline{\text{RTS2}}$/$\overline{\text{SS2}}$/TXD8 | IIO1_0 | | |
| 35 | | P7_2 | | TA1OUT/V | CLK2 | | | |
| 36 | | P7_1 | | TB5IN/ TA0IN | RXD2/SCL2/STXD2/ MSCL | IIO1_7/OUTC2_2/ ISRXD2/IEIN | | |

**Table 1.7 Pin Characteristics for the 144-pin Package (2/4)**

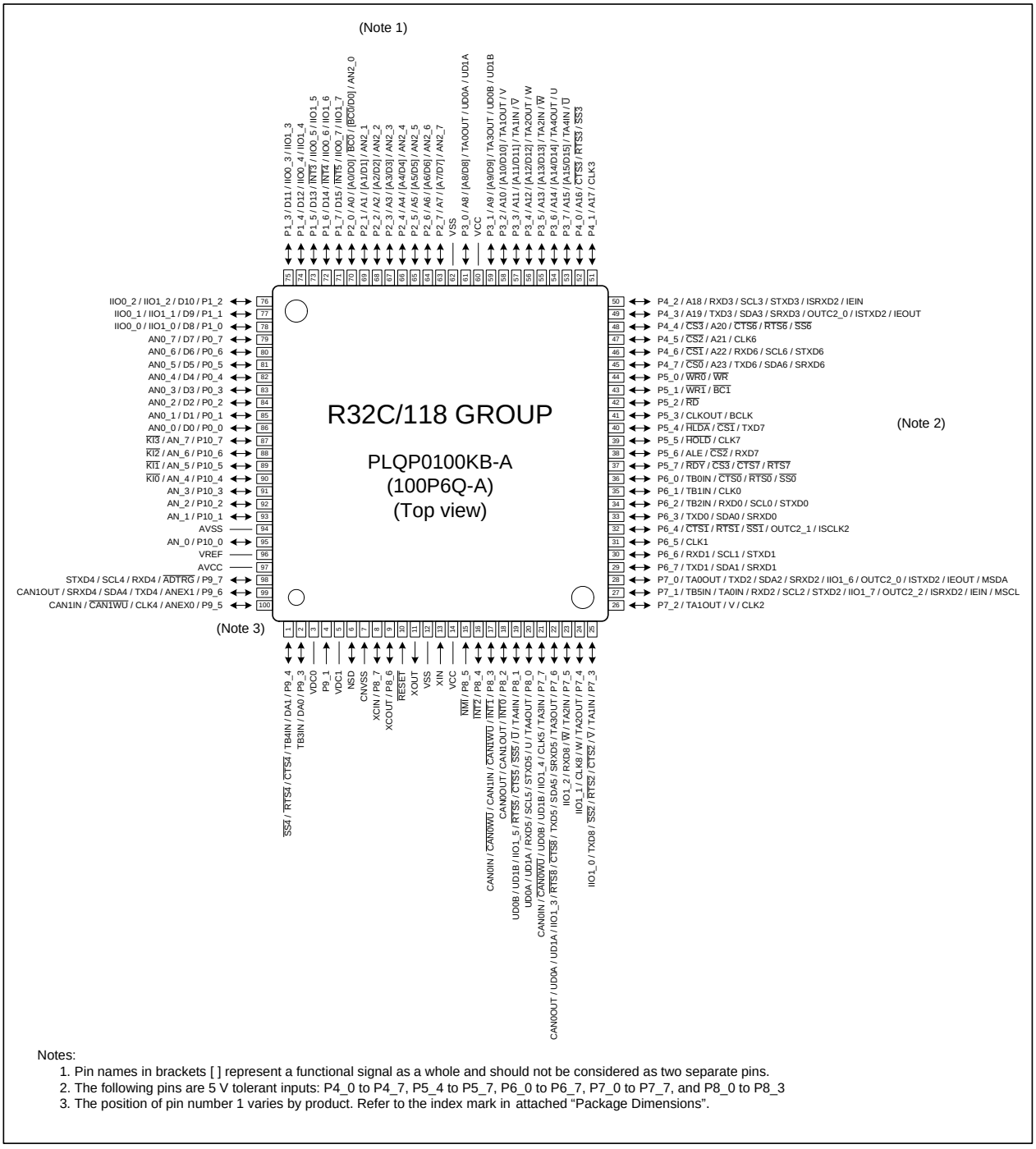
| Pin No. | Control Pin | Port | Interrupt Pin | Timer Pin | UART/CAN Module Pin | Intelligent I/O Pin | Analog Pin | Bus Control Pin |
|---|---|---|---|---|---|---|---|---|
| 37 | | P7_0 | | TA0OUT | TXD2/SDA2/SRXD2/ MSDA | IIO1_6/OUTC2_0/ ISTXD2/IEOUT | | |
| 38 | | P6_7 | | | TXD1/SDA1/SRXD1 | | | |
| 39 | VCC | | | | | | | |
| 40 | | P6_6 | | | RXD1/SCL1/STXD1 | | | |
| 41 | VSS | | | | | | | |
| 42 | | P6_5 | | | CLK1 | | | |
| 43 | | P6_4 | | | $\overline{\text{CTS1}}/\overline{\text{RTS1}}/\overline{\text{SS1}}$ | OUTC2_1/ISCLK2 | | |
| 44 | | P6_3 | | | TXD0/SDA0/SRXD0 | | | |
| 45 | | P6_2 | | TB2IN | RXD0/SCL0/STXD0 | | | |
| 46 | | P6_1 | | TB1IN | CLK0 | | | |
| 47 | | P6_0 | | TB0IN | $\overline{\text{CTS0}}/\overline{\text{RTS0}}/\overline{\text{SS0}}$ | | | |
| 48 | | P13_7 | | | | OUTC2_7 | | D31 |
| 49 | | P13_6 | | | | OUTC2_1/ISCLK2 | | D30 |
| 50 | | P13_5 | | | | OUTC2_2/ISRXD2/ IEIN | | D29 |
| 51 | | P13_4 | | | | OUTC2_0/ISTXD2/ IEOUT | | D28 |
| 52 | | P5_7 | | | $\overline{\text{CTS7}}/\overline{\text{RTS7}}$ | | | RDY/CS3 |
| 53 | | P5_6 | | | RXD7 | | | ALE/$\overline{\text{CS2}}$ |
| 54 | | P5_5 | | | CLK7 | | | $\overline{\text{HOLD}}$ |
| 55 | | P5_4 | | | TXD7 | | | $\overline{\text{HLDA}}/\overline{\text{CS1}}$ |
| 56 | | P13_3 | | | | OUTC2_3 | | D27 |
| 57 | VSS | | | | | | | |
| 58 | | P13_2 | | | | OUTC2_6 | | D26 |
| 59 | VCC | | | | | | | |
| 60 | | P13_1 | | | | OUTC2_5 | | D25 |
| 61 | | P13_0 | | | | OUTC2_4 | | D24 |
| 62 | | P5_3 | | | | | | CLKOUT/ BCLK |
| 63 | | P5_2 | | | | | | $\overline{\text{RD}}$ |
| 64 | | P5_1 | | | | | | $\overline{\text{WR1}}/\overline{\text{BC1}}$ |
| 65 | | P5_0 | | | | | | $\overline{\text{WR0}}/\overline{\text{WR}}$ |
| 66 | | P12_7 | | | | | | D23 |
| 67 | | P12_6 | | | | | | D22 |
| 68 | | P12_5 | | | | | | D21 |
| 69 | | P4_7 | | | TXD6/SDA6/SRXD6 | | | $\overline{\text{CS0}}$/A23 |
| 70 | | P4_6 | | | RXD6/SCL6/STXD6 | | | $\overline{\text{CS1}}$/A22 |
| 71 | | P4_5 | | | CLK6 | | | $\overline{\text{CS2}}$/A21 |
| 72 | | P4_4 | | | $\overline{\text{CTS6}}/\overline{\text{RTS6}}/\overline{\text{SS6}}$ | | | $\overline{\text{CS3}}$/A20 |
| 73 | | P4_3 | | | TXD3/SDA3/SRXD3 | OUTC2_0/ISTXD2/ IEOUT | | A19 |
| 74 | VCC | | | | | | | |

**Table 1.8 Pin Characteristics for the 144-pin Package (3/4)**

| Pin No. | Control Pin | Port | Interrupt Pin | Timer Pin | UART/CAN Module Pin | Intelligent I/O Pin | Analog Pin | Bus Control Pin |
|---|---|---|---|---|---|---|---|---|
| 75 | | P4_2 | | | RXD3/SCL3/STXD3 | ISRXD2/IEIN | | A18 |
| 76 | VSS | | | | | | | |
| 77 | | P4_1 | | | CLK3 | | | A17 |
| 78 | | P4_0 | | | $\overline{\text{CTS3}}$/$\overline{\text{RTS3}}$/$\overline{\text{SS3}}$ | | | A16 |
| 79 | | P3_7 | | TA4IN/$\overline{\text{U}}$ | | | | A15(/D15) |
| 80 | | P3_6 | | TA4OUT/U | | | | A14(/D14) |
| 81 | | P3_5 | | TA2IN/$\overline{\text{W}}$ | | | | A13(/D13) |
| 82 | | P3_4 | | TA2OUT/W | | | | A12(/D12) |
| 83 | | P3_3 | | TA1IN/$\overline{\text{V}}$ | | | | A11(/D11) |
| 84 | | P3_2 | | TA1OUT/V | | | | A10(/D10) |
| 85 | | P3_1 | | TA3OUT | | UD0B/UD1B | | A9(/D9) |
| 86 | | P12_4 | | | | | | D20 |
| 87 | | P12_3 | | | $\overline{\text{CTS6}}$/$\overline{\text{RTS6}}$/$\overline{\text{SS6}}$ | | | D19 |
| 88 | | P12_2 | | | RXD6/SCL6/STXD6 | | | D18 |
| 89 | | P12_1 | | | CLK6 | | | D17 |
| 90 | | P12_0 | | | TXD6/SDA6/SRXD6 | | | D16 |
| 91 | VCC | | | | | | | |
| 92 | | P3_0 | | TA0OUT | | UD0A/UD1A | | A8(/D8) |
| 93 | VSS | | | | | | | |
| 94 | | P2_7 | | | | | AN2_7 | A7(/D7) |
| 95 | | P2_6 | | | | | AN2_6 | A6(/D6) |
| 96 | | P2_5 | | | | | AN2_5 | A5(/D5) |
| 97 | | P2_4 | | | | | AN2_4 | A4(/D4) |
| 98 | | P2_3 | | | | | AN2_3 | A3(/D3) |
| 99 | | P2_2 | | | | | AN2_2 | A2(/D2) |
| 100 | | P2_1 | | | | | AN2_1 | A1(/D1)/ $\overline{\text{BC2}}$(/D1) |
| 101 | | P2_0 | | | | | AN2_0 | A0(/D0)/ $\overline{\text{BC0}}$(/D0) |
| 102 | | P1_7 | $\overline{\text{INT5}}$ | | | IIO0_7/IIO1_7 | | D15 |
| 103 | | P1_6 | $\overline{\text{INT4}}$ | | | IIO0_6/IIO1_6 | | D14 |
| 104 | | P1_5 | $\overline{\text{INT3}}$ | | | IIO0_5/IIO1_5 | | D13 |
| 105 | | P1_4 | | | | IIO0_4/IIO1_4 | | D12 |
| 106 | | P1_3 | | | | IIO0_3/IIO1_3 | | D11 |
| 107 | | P1_2 | | | | IIO0_2/IIO1_2 | | D10 |
| 108 | | P1_1 | | | | IIO0_1/IIO1_1 | | D9 |
| 109 | | P1_0 | | | | IIO0_0/IIO1_0 | | D8 |
| 110 | | P0_7 | | | | | AN0_7 | D7 |
| 111 | | P0_6 | | | | | AN0_6 | D6 |
| 112 | | P0_5 | | | | | AN0_5 | D5 |
| 113 | | P0_4 | | | | | AN0_4 | D4 |
| 114 | | P11_4 | | | | | | $\overline{\text{BC3}}$/$\overline{\text{WR3}}$ |

**Table 1.9 Pin Characteristics for the 144-pin Package (4/4)**

| Pin No. | Control Pin | Port | Interrupt Pin | Timer Pin | UART/CAN Module Pin | Intelligent I/O Pin | Analog Pin | Bus Control Pin |
|---|---|---|---|---|---|---|---|---|
| 115 | | P11_3 | | | CTS8/RTS8 | IIO1_3 | | CS3/WR2 |
| 116 | | P11_2 | | | RXD8 | IIO1_2 | | CS2 |
| 117 | | P11_1 | | | CLK8 | IIO1_1 | | CS1 |
| 118 | | P11_0 | | | TXD8 | IIO1_0 | | CS0 |
| 119 | | P0_3 | | | | | AN0_3 | D3 |
| 120 | | P0_2 | | | | | AN0_2 | D2 |
| 121 | | P0_1 | | | | | AN0_1 | D1 |
| 122 | | P0_0 | | | | | AN0_0 | D0 |
| 123 | | P15_7 | | | CTS6/RTS6/SS6 | IIO0_7 | AN15_7 | |
| 124 | | P15_6 | | | CLK6 | IIO0_6 | AN15_6 | |
| 125 | | P15_5 | | | RXD6/SCL6/STXD6 | IIO0_5 | AN15_5 | |
| 126 | | P15_4 | | | TXD6/SDA6/SRXD6 | IIO0_4 | AN15_4 | |
| 127 | | P15_3 | | | CTS7/RTS7 | IIO0_3 | AN15_3 | |
| 128 | | P15_2 | | | RXD7 | IIO0_2 | AN15_2 | |
| 129 | | P15_1 | | | CLK7 | IIO0_1 | AN15_1 | |
| 130 | VSS | | | | | | | |
| 131 | | P15_0 | | | TXD7 | IIO0_0 | AN15_0 | |
| 132 | VCC | | | | | | | |
| 133 | | P10_7 | KI3 | | | | AN_7 | |
| 134 | | P10_6 | KI2 | | | | AN_6 | |
| 135 | | P10_5 | KI1 | | | | AN_5 | |
| 136 | | P10_4 | KI0 | | | | AN_4 | |
| 137 | | P10_3 | | | | | AN_3 | |
| 138 | | P10_2 | | | | | AN_2 | |
| 139 | | P10_1 | | | | | AN_1 | |
| 140 | AVSS | | | | | | | |
| 141 | | P10_0 | | | | | AN_0 | |
| 142 | VREF | | | | | | | |
| 143 | AVCC | | | | | | | |
| 144 | | P9_7 | | | RXD4/SCL4/STXD4 | | ADTRG | |




Notes:
1. Pin names in brackets [ ] represent a functional signal as a whole and should not be considered as two separate pins.
2. The following pins are 5 V tolerant inputs: P4_0 to P4_7, P5_4 to P5_7, P6_0 to P6_7, P7_0 to P7_7, and P8_0 to P8_3
3. The position of pin number 1 varies by product. Refer to the index mark in attached “Package Dimensions”.

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

**Table 1.10 Pin Characteristics for the 100-pin Package (1/3)**

| Pin No. | Control Pin | Port | Interrupt Pin | Timer Pin | UART/CAN Module Pin | Intelligent I/O Pin | Analog Pin | Bus Control Pin |
|---|---|---|---|---|---|---|---|---|
| 1 | | P9_4 | | TB4IN | CTS4/RTS4/SS4 | | DA1 | |
| 2 | | P9_3 | | TB3IN | | | DA0 | |
| 3 | VDC0 | | | | | | | |
| 4 | | P9_1 | | | | | | |
| 5 | VDC1 | | | | | | | |
| 6 | NSD | | | | | | | |
| 7 | CNVSS | | | | | | | |
| 8 | XCIN | P8_7 | | | | | | |
| 9 | XCOUT | P8_6 | | | | | | |
| 10 | RESET | | | | | | | |
| 11 | XOUT | | | | | | | |
| 12 | VSS | | | | | | | |
| 13 | XIN | | | | | | | |
| 14 | VCC | | | | | | | |
| 15 | | P8_5 | NMI | | | | | |
| 16 | | P8_4 | INT2 | | | | | |
| 17 | | P8_3 | INT1 | | CAN0IN/CAN0WU/ CAN1IN/CAN1WU | | | |
| 18 | | P8_2 | INT0 | | CAN0OUT/CAN1OUT | | | |
| 19 | | P8_1 | | TA4IN/U | CTS5/RTS5/SS5 | IIO1_5/UD0B/UD1B | | |
| 20 | | P8_0 | | TA4OUT/U | RXD5/SCL5/STXD5 | UD0A/UD1A | | |
| 21 | | P7_7 | | TA3IN | CLK5/CAN0IN/ CAN0WU | IIO1_4/UD0B/UD1B | | |
| 22 | | P7_6 | | TA3OUT | TXD5/SDA5/SRXD5/ CTS8/RTS8/CAN0OUT | IIO1_3/UD0A/UD1A | | |
| 23 | | P7_5 | | TA2IN/W | RXD8 | IIO1_2 | | |
| 24 | | P7_4 | | TA2OUT/W | CLK8 | IIO1_1 | | |
| 25 | | P7_3 | | TA1IN/V | CTS2/RTS2/SS2/TXD8 | IIO1_0 | | |
| 26 | | P7_2 | | TA1OUT/V | CLK2 | | | |
| 27 | | P7_1 | | TB5IN/ TA0IN | RXD2/SCL2/STXD2/ MSCL | IIO1_7/OUTC2_2/ ISRXD2/IEIN | | |
| 28 | | P7_0 | | TA0OUT | TXD2/SDA2/SRXD2/ MSDA | IIO1_6/OUTC2_0/ ISTXD2/IEOUT | | |
| 29 | | P6_7 | | | TXD1/SDA1/SRXD1 | | | |
| 30 | | P6_6 | | | RXD1/SCL1/STXD1 | | | |
| 31 | | P6_5 | | | CLK1 | | | |
| 32 | | P6_4 | | | CTS1/RTS1/SS1 | OUTC2_1/ISCLK2 | | |
| 33 | | P6_3 | | | TXD0/SDA0/SRXD0 | | | |
| 34 | | P6_2 | | TB2IN | RXD0/SCL0/STXD0 | | | |
| 35 | | P6_1 | | TB1IN | CLK0 | | | |
| 36 | | P6_0 | | TB0IN | CTS0/RTS0/SS0 | | | |
| 37 | | P5_7 | | | CTS7/RTS7 | | | RDY/CS3 |
| 38 | | P5_6 | | | RXD7 | | | ALE/CS2 |

**Table 1.11 Pin Characteristics for the 100-pin Package (2/3)**

| Pin No. | Control Pin | Port | Interrupt Pin | Timer Pin | UART/CAN Module Pin | Intelligent I/O Pin | Analog Pin | Bus Control Pin |
|---|---|---|---|---|---|---|---|---|
| 39 | | P5_5 | | | CLK7 | | | $\overline{\text{HOLD}}$ |
| 40 | | P5_4 | | | TXD7 | | | $\overline{\text{HLDA}}$/$\overline{\text{CS1}}$ |
| 41 | | P5_3 | | | | | | CLKOUT/ BCLK |
| 42 | | P5_2 | | | | | | $\overline{\text{RD}}$ |
| 43 | | P5_1 | | | | | | $\overline{\text{WR1}}$/$\overline{\text{BC1}}$ |
| 44 | | P5_0 | | | | | | $\overline{\text{WR0}}$/$\overline{\text{WR}}$ |
| 45 | | P4_7 | | | TXD6/SDA6/SRXD6 | | | $\overline{\text{CS0}}$/A23 |
| 46 | | P4_6 | | | RXD6/SCL6/STXD6 | | | $\overline{\text{CS1}}$/A22 |
| 47 | | P4_5 | | | CLK6 | | | $\overline{\text{CS2}}$/A21 |
| 48 | | P4_4 | | | $\overline{\text{CTS6}}$/$\overline{\text{RTS6}}$/$\overline{\text{SS6}}$ | | | $\overline{\text{CS3}}$/A20 |
| 49 | | P4_3 | | | TXD3/SDA3/SRXD3 | OUTC2_0/ISTXD2/ IEOUT | | A19 |
| 50 | | P4_2 | | | RXD3/SCL3/STXD3 | ISRXD2/IEIN | | A18 |
| 51 | | P4_1 | | | CLK3 | | | A17 |
| 52 | | P4_0 | | | $\overline{\text{CTS3}}$/$\overline{\text{RTS3}}$/$\overline{\text{SS3}}$ | | | A16 |
| 53 | | P3_7 | | TA4IN/$\overline{\text{U}}$ | | | | A15(/D15) |
| 54 | | P3_6 | | TA4OUT/U | | | | A14(/D14) |
| 55 | | P3_5 | | TA2IN/$\overline{\text{W}}$ | | | | A13(/D13) |
| 56 | | P3_4 | | TA2OUT/W | | | | A12(/D12) |
| 57 | | P3_3 | | TA1IN/$\overline{\text{V}}$ | | | | A11(/D11) |
| 58 | | P3_2 | | TA1OUT/V | | | | A10(/D10) |
| 59 | | P3_1 | | TA3OUT | | UD0B/UD1B | | A9(/D9) |
| 60 | VCC | | | | | | | |
| 61 | | P3_0 | | TA0OUT | | UD0A/UD1A | | A8(/D8) |
| 62 | VSS | | | | | | | |
| 63 | | P2_7 | | | | | AN2_7 | A7(/D7) |
| 64 | | P2_6 | | | | | AN2_6 | A6(/D6) |
| 65 | | P2_5 | | | | | AN2_5 | A5(/D5) |
| 66 | | P2_4 | | | | | AN2_4 | A4(/D4) |
| 67 | | P2_3 | | | | | AN2_3 | A3(/D3) |
| 68 | | P2_2 | | | | | AN2_2 | A2(/D2) |
| 69 | | P2_1 | | | | | AN2_1 | A1(/D1) |
| 70 | | P2_0 | | | | | AN2_0 | A0(/D0)/ $\overline{\text{BC0}}$(/D0) |
| 71 | | P1_7 | $\overline{\text{INT5}}$ | | | IIO0_7/IIO1_7 | | D15 |
| 72 | | P1_6 | $\overline{\text{INT4}}$ | | | IIO0_6/IIO1_6 | | D14 |
| 73 | | P1_5 | $\overline{\text{INT3}}$ | | | IIO0_5/IIO1_5 | | D13 |
| 74 | | P1_4 | | | | IIO0_4/IIO1_4 | | D12 |
| 75 | | P1_3 | | | | IIO0_3/IIO1_3 | | D11 |

**Table 1.12 Pin Characteristics for the 100-pin Package (3/3)**

| Pin No. | Control Pin | Port | Interrupt Pin | Timer Pin | UART/CAN Module Pin | Intelligent I/O Pin | Analog Pin | Bus Control Pin |
|---|---|---|---|---|---|---|---|---|
| 76 | | P1_2 | | | | IIO0_2/IIO1_2 | | D10 |
| 77 | | P1_1 | | | | IIO0_1/IIO1_1 | | D9 |
| 78 | | P1_0 | | | | IIO0_0/IIO1_0 | | D8 |
| 79 | | P0_7 | | | | | AN0_7 | D7 |
| 80 | | P0_6 | | | | | AN0_6 | D6 |
| 81 | | P0_5 | | | | | AN0_5 | D5 |
| 82 | | P0_4 | | | | | AN0_4 | D4 |
| 83 | | P0_3 | | | | | AN0_3 | D3 |
| 84 | | P0_2 | | | | | AN0_2 | D2 |
| 85 | | P0_1 | | | | | AN0_1 | D1 |
| 86 | | P0_0 | | | | | AN0_0 | D0 |
| 87 | | P10_7 | $\overline{KI3}$ | | | | AN_7 | |
| 88 | | P10_6 | $\overline{KI2}$ | | | | AN_6 | |
| 89 | | P10_5 | $\overline{KI1}$ | | | | AN_5 | |
| 90 | | P10_4 | $\overline{KI0}$ | | | | AN_4 | |
| 91 | | P10_3 | | | | | AN_3 | |
| 92 | | P10_2 | | | | | AN_2 | |
| 93 | | P10_1 | | | | | AN_1 | |
| 94 | AVSS | | | | | | | |
| 95 | | P10_0 | | | | | AN_0 | |
| 96 | VREF | | | | | | | |
| 97 | AVCC | | | | | | | |
| 98 | | P9_7 | | | RXD4/SCL4/STXD4 | | $\overline{ADTRG}$ | |
| 99 | | P9_6 | | | TXD4/SDA4/SRXD4/ CAN1OUT | | ANEX1 | |
| 100 | | P9_5 | | | CLK4/CAN1IN/ $\overline{CAN1WU}$ | | ANEX0 | |

## 1.5 Pin Definitions and Functions

Table 1.13 to Table 1.17 show the pin definitions and functions.

**Table 1.13 Pin Definitions and Functions (1/4)**

| Function | Symbol | I/O | Description |
|---|---|---|---|
| Power supply | VCC, VSS | I | Applicable as follows: VCC = 3.0 to 5.5 V, VSS = 0 V |
| Connecting pins for decoupling capacitor | VDC0, VDC1 | — | A decoupling capacitor for internal voltage should be connected between VDC0 and VDC1 |
| Analog power supply | AVCC, AVSS | I | Power supply for the A/D converter. AVCC and AVSS should be connected to VCC and VSS, respectively |
| Reset input | $\overline{\text{RESET}}$ | I | The MCU is reset when this pin is driven low |
| CNVSS | CNVSS | I | This pin should be connected to VSS via a resistor |
| Debug port | NSD | I/O | This pin is to communicate with a debugger. It should be connected to VCC via a resistor of 1 to 4.7 kΩ |
| Main clock input | XIN | I | Input/output for the main clock oscillator. A crystal, or a ceramic resonator should be connected between pins XIN and XOUT. An external clock should be input at the XIN while leaving the XOUT open |
| Main clock output | XOUT | O | |
| Sub clock input | XCIN | I | Input/output for the sub clock oscillator. A crystal oscillator should be connected between pins XCIN and XCOUT. An external clock should be input at the XCIN while leaving the XCOUT open |
| Sub clock output | XCOUT | O | |
| BCLK output | BCLK | O | BCLK output |
| Clock output | CLKOUT | O | Output of the clock with the same frequency as fC, f8, or f32 |
| External interrupt input | $\overline{\text{INT0}}$ to $\overline{\text{INT8}}$ (1) | I | Input for external interrupts |
| NMI input | P8_5/$\overline{\text{NMI}}$ | I | Input for NMI |
| Key input interrupt | $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$ | I | Input for the key input interrupt |
| Bus control pins | D0 to D7 | I/O | Input/output of data (D0 to D7) while accessing an external memory space with a separate bus |
| | D8 to D15 | I/O | Input/output of data (D8 to D15) while accessing an external memory space with 16-bit or 32-bit separate bus |
| | D16 to D31 (2) | I/O | Input/output of data (D16 to D31) while accessing an external memory space with 32-bit separate bus |
| | A0 to A23 | O | Output of address bits A0 to A23 |
| | A0/D0 to A7/D7 | I/O | Output of address bits (A0 to A7) and input/output of data (D0 to D7) by time-division while accessing an external memory space with multiplexed bus |
| | A8/D8 to A15/D15 | I/O | Output of address bits (A8 to A15) and input/output of data (D8 to D15) by time-division while accessing an external memory space with 16-bit or 32-bit multiplexed bus |

Notes:
1. Pins $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ are available in the 144-pin package only.
2. Pins D16 to D31 are available in the 144-pin package only.

**Table 1.14 Pin Definitions and Functions (2/4)**

| Function | Symbol | I/O | Description |
|---|---|---|---|
| Bus control pins | $\overline{BC0}$/D0, $\overline{BC2}$/D1 (1) | I/O | Output of byte control ($\overline{BC0}$ and $\overline{BC2}$) and input/output of data (D0 and D1) by time-division while accessing an external memory space with multiplexed bus |
| | $\overline{CS0}$ to $\overline{CS3}$ | O | Chip select output |
| | $\overline{WR0}$/$\overline{WR1}$/$\overline{WR2}$/$\overline{WR3}$<br>$\overline{WR}$/$\overline{BC0}$/$\overline{BC1}$/$\overline{BC2}$/$\overline{BC3}$<br>$\overline{RD}$ (1) | O | Output of write, byte control, and read signals. Either $\overline{WRx}$ or $\overline{WR}$ and $\overline{BCx}$ can be selected by a program.<br>Data is read when $\overline{RD}$ is low.<br><br>• When $\overline{WR0}$, $\overline{WR1}$, $\overline{WR2}$, $\overline{WR3}$, and $\overline{RD}$ are selected, data is written to the following address:<br>4n+0, when $\overline{WR0}$ is low<br>4n+1, when $\overline{WR1}$ is low<br>4n+2, when $\overline{WR2}$ is low<br>4n+3, when $\overline{WR3}$ is low<br>on 32-bit external data bus<br>or<br>an even address, when $\overline{WR0}$ is low<br>an odd address, when $\overline{WR1}$ is low<br>on 16-bit external data bus<br><br>• When $\overline{WR}$, $\overline{BC0}$, $\overline{BC1}$, $\overline{BC2}$, $\overline{BC3}$, and $\overline{RD}$ are selected, data is written, when $\overline{WR}$ is low<br>and<br>the following address is accessed:<br>4n+0, when $\overline{BC0}$ is low<br>4n+1, when $\overline{BC1}$ is low<br>4n+2, when $\overline{BC2}$ is low<br>4n+3, when $\overline{BC3}$ is low<br>on 32-bit external data bus<br>or<br>an even address, when $\overline{BC0}$ is low<br>an odd address, when $\overline{BC1}$ is low<br>on 16-bit external data bus |
| | ALE | O | Latch enable signal in multiplexed bus format |
| | $\overline{HOLD}$ | I | The MCU is in a hold state while this pin is held low |
| | $\overline{HLDA}$ | O | This pin is driven low while the MCU is held in a hold state |
| | $\overline{RDY}$ | I | Bus cycle is extended by the CPU if this pin is low on the falling edge of the BCLK |

Note:
1. Pins $\overline{BC2}$/D1, $\overline{WR2}$, $\overline{WR3}$, $\overline{BC2}$, and $\overline{BC3}$ are available in the 144-pin package only.

**Table 1.15 Pin Definitions and Functions (3/4)**

| Function | Symbol | I/O | Description |
|---|---|---|---|
| I/O port (1, 2) | P0_0 to P0_7<br>P1_0 to P1_7<br>P2_0 to P2_7<br>P3_0 to P3_7<br>P4_0 to P4_7<br>P5_0 to P5_7<br>P6_0 to P6_7<br>P7_0 to P7_7<br>P8_0 to P8_4<br>P8_6, P8_7<br>P9_0 to P9_7<br>P10_0 to P10_7<br>P11_0 to P11_4<br>P12_0 to P12_7<br>P13_0 to P13_7<br>P14_3 to P14_6<br>P15_0 to P15_7 | I/O | I/O ports in CMOS. Each port can be programmed to input or output under the control of the direction register.<br>Some ports are 5 V tolerant inputs.<br>Pull-up resistors and N-channel open drain setting can be enabled on some ports. Refer to Table 1.17 “Pin Specifications” for details |
| Input port (2) | P9_1 (for 100-pin package)<br>P14_1 (for 144-pin package) | I | Input port in CMOS<br>Pull-up resistor is selectable.<br>Refer to Table 1.17 “Pin Specifications” for details |
| Timer A | TA0OUT to TA4OUT | I/O | Timers A0 to A4 input/output |
| | TA0IN to TA4IN | I | Timers A0 to A4 input |
| Timer B | TB0IN to TB5IN | I | Timers B0 to B5 input |
| Three-phase motor control timer output | U,$\overline{U}$,V,$\overline{V}$,W,$\overline{W}$ | O | Three-phase motor control timer output |
| Serial interface | $\overline{CTS0}$ to $\overline{CTS8}$ | I | Handshake input |
| | $\overline{RTS0}$ to $\overline{RTS8}$ | O | Handshake output |
| | CLK0 to CLK8 | I/O | Transmit/receive clock input/output |
| | RXD0 to RXD8 | I | Serial data input |
| | TXD0 to TXD8 | O | Serial data output |
| $I^2C$ bus (simplified) | SDA0 to SDA6 | I/O | Serial data input/output |
| | SCL0 to SCL6 | I/O | Transmit/receive clock input/output |
| Serial interface special functions | STXD0 to STXD6 | O | Serial data output in slave mode |
| | SRXD0 to SRXD6 | I | Serial data input in slave mode |
| | $\overline{SS0}$ to $\overline{SS6}$ | I | Input to control serial interface special functions |

Notes:
1. Port P9_1 in the 100-pin package is an input-only port.
2. Ports P9_0, P9_2, and P11 to P15 are available in the 144-pin package only.

**Table 1.16 Pin Definitions and Functions (4/4)**

| Function | Symbol | I/O | Description |
|---|---|---|---|
| A/D converter | AN_0 to AN_7<br>AN0_0 to AN0_7<br>AN2_0 to AN2_7<br>AN15_0 to AN15_7 (1) | I | Analog input for the A/D converter |
| | $\overline{\text{ADTRG}}$ | I | External trigger input for the A/D converter |
| | ANEX0 | I/O | Expanded analog input for the A/D converter and output in external op-amp connection mode |
| | ANEX1 | I | Expanded analog input for the A/D converter |
| D/A converter | DA0, DA1 | O | Output for the D/A converter |
| Reference voltage input | VREF | I | Reference voltage input for the A/D converter and D/A converter |
| Intelligent I/O | IIO0_0 to IIO0_7 | I/O | Input/output for the Intelligent I/O group 0. Either input capture or output compare is selectable |
| | IIO1_0 to IIO1_7 | I/O | Input/output for the Intelligent I/O group 1. Either input capture or output compare is selectable |
| | UD0A, UD0B,<br>UD1A, UD1B | I | Input for the two-phase encoder |
| | OUTC2_0 to OUTC2_7 (2) | O | Output for OC (output compare) of the Intelligent I/O group 2 |
| | ISCLK2 | I/O | Clock input/output for the serial interface |
| | ISRXD2 | I | Receive data input for the serial interface |
| | ISTXD2 | O | Transmit data output for the serial interface |
| | IEIN | I | Receive data input for the serial interface |
| | IEOUT | O | Transmit data output for the serial interface |
| Multi-master $I^2C$-bus | MSDA | I/O | Serial data input/output |
| | MSCL | I/O | Transmit/receive clock input/output |
| CAN Module | CAN0IN, CAN1IN | I | Receive data input for the CAN communications |
| | CAN0OUT,<br>CAN1OUT | O | Transmit data output for the CAN communications |
| | $\overline{\text{CAN0WU}}$,<br>$\overline{\text{CAN1WU}}$ | I | Input for the CAN wake-up interrupt |

Notes:
1. Pins AN15_0 to AN15_7 are available in the 144-pin package only.
2. Pins OUTC2_3 to OUTC2_7 are available in the 144-pin package only.

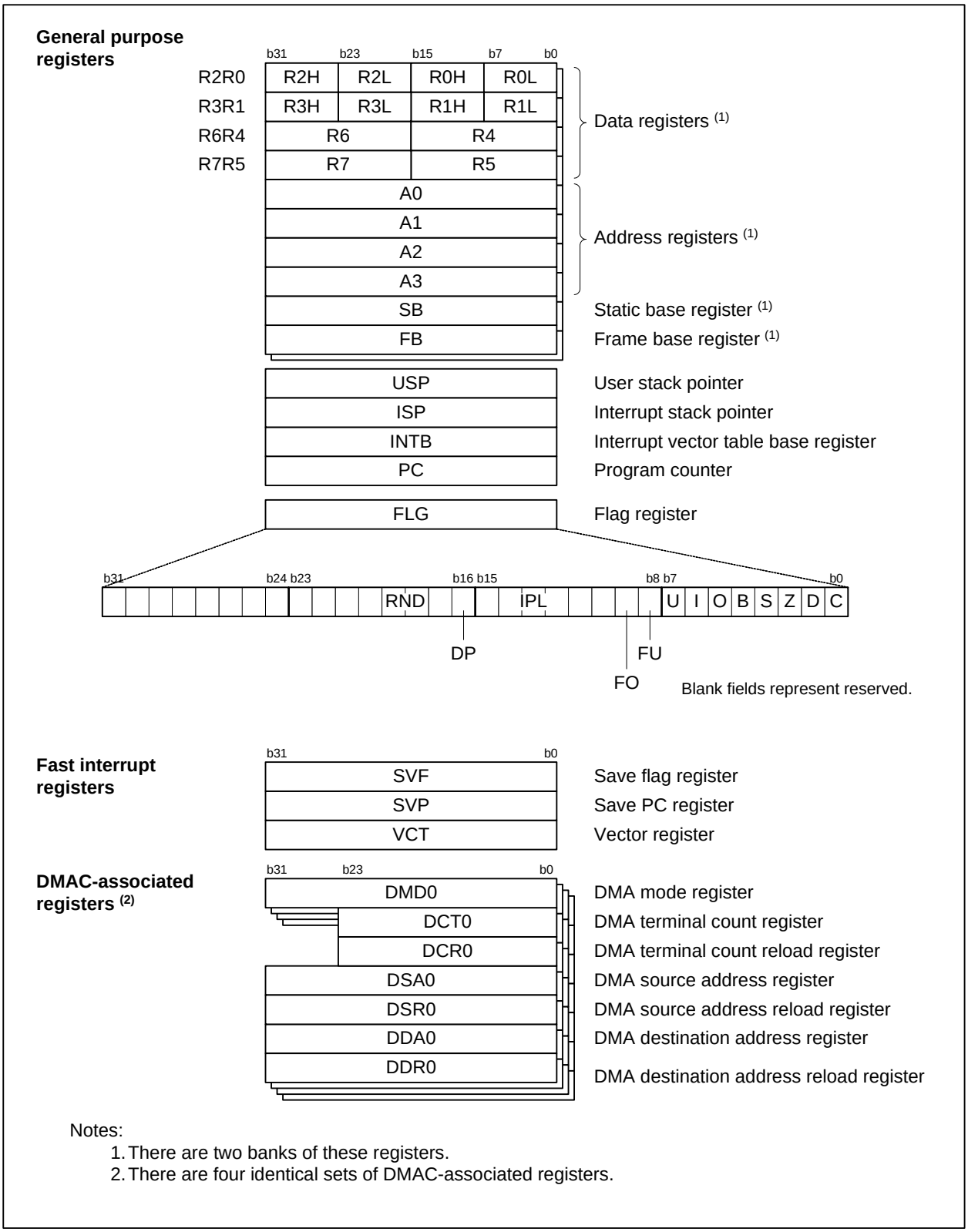
**Table 1.17 Pin Specifications**

| Pin names | Package | | Selectable Functions | | 5 V tolerant input (3) |
|---|---|---|---|---|---|
| | 144-pin | 100-pin | Pull-up resistor (1) | N-channel open drain (2) | |
| P0_0 to P0_7 | ✓ | ✓ | ✓ | | |
| P1_0 to P1_7 | ✓ | ✓ | ✓ | | |
| P2_0 to P2_7 | ✓ | ✓ | ✓ | | |
| P3_0 to P3_7 | ✓ | ✓ | ✓ | | |
| P4_0 to P4_7 | ✓ | ✓ | | ✓ | ✓ |
| P5_0 to P5_3 | ✓ | ✓ | ✓ | | |
| P5_4 to P5_7 | ✓ | ✓ | | ✓ | ✓ |
| P6_0 to P6_7 | ✓ | ✓ | | ✓ | ✓ |
| P7_0 to P7_7 | ✓ | ✓ | | ✓ | ✓ |
| P8_0 to P8_3 | ✓ | ✓ | | ✓ | ✓ |
| P8_4, P8_6, P8_7 | ✓ | ✓ | ✓ | | |
| P9_0 to P9_3 (144-pin) | ✓ | | ✓ | ✓ | |
| P9_1, P9_3 (100-pin) | | ✓ | ✓ | | |
| P9_4 to P9_7 | ✓ | ✓ | ✓ | ✓ | |
| P10_0 to P10_7 | ✓ | ✓ | ✓ | | |
| P11_0 to P11_3 | ✓ | | ✓ | ✓ | |
| P11_4 | ✓ | | ✓ | | |
| P12_0 to P12_3 | ✓ | | ✓ | ✓ | |
| P12_4 to P12_7 | ✓ | | ✓ | | |
| P13_0 to P13_7 | ✓ | | ✓ | | |
| P14_1 | ✓ | | ✓ | | |
| P14_3 to P14_6 | ✓ | | ✓ | | |
| P15_0 to P15_7 | ✓ | | ✓ | ✓ | |

Notes:
1. Pull-up resistors are selected in 4-pin units, but are only enabled for those pins set as input ports.
2. N-channel open drain output can be enabled on the applicable pins on a discrete pin basis.
3. 5 V tolerant input is enabled when an applicable pin is set as an input port. When it is set as an I/O port, to enable 5 V tolerant input, this pin should be set as N-channel open drain output.

# 2. Central Processing Unit (CPU)

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




Notes:
1. There are two banks of these registers.
2. There are four identical sets of DMAC-associated registers.

**Figure 2.1 CPU Registers**

## 2.1 General Purpose Registers

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

These 32-bit registers are primarily used for transfers and arithmetic/logic operations.
Each of the registers can be divided into upper and lower 16-bit registers, e.g. R2R0 can be divided into R2 and R0, R3R0 can be divided into R3 and R1, etc.
Moreover, data registers R2R0 and R3R1 can be divided into four 8-bit data registers: upper (R2H and R3H), mid-upper (R2L and R3L), mid-lower (R0H and R1H), and lower (R0L and R1L).

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

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

### 2.1.3 Static Base Register (SB)

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

### 2.1.4 Frame Base Register (FB)

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

### 2.1.5 Program Counter (PC)

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

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

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

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

Two types of 32-bit stack pointers (SPs) are provided: user stack pointer (USP) and interrupt stack pointer (ISP).
Use the stack pointer select flag (U flag) to select either the user stack pointer (USP) or the interrupt stack pointer (ISP). The U flag is bit 7 in the flag register (FLG). Refer to 2.1.8 “Flag Register (FLG)” for details.
To minimize the overhead of interrupt sequence due to less memory access, set the user stack pointer (USP) or the interrupt stack pointer (ISP) to a multiple of 4.

### 2.1.8 Flag Register (FLG)

This 32-bit register indicates the CPU status.

#### 2.1.8.1 Carry Flag (C flag)

This flag becomes 1 when any of the carry, borrow, shifted-out bit, etc. is generated in the arithmetic logic unit (ALU).

#### 2.1.8.2 Debug Flag (D flag)

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

#### 2.1.8.3 Zero Flag (Z flag)

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

#### 2.1.8.4 Sign Flag (S flag)

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

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

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

#### 2.1.8.6 Overflow Flag (O flag)

This flag becomes 1 if an overflow occurs in an operation; otherwise it is 0.

#### 2.1.8.7 Interrupt Enable Flag (I flag)

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

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

To select the interrupt stack pointer (ISP), set this flag to 0. To select the user stack pointer (USP), set this flag to 1.
It becomes 0 when a hardware interrupts is accepted or when an INT instruction designated by a software interrupt number from 0 to 127 is executed.

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

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

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

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

#### 2.1.8.11 Processor Interrupt Priority Level (IPL)

The processor interrupt priority level (IPL), consisting of three bits, selects a processor interrupt priority level from level 0 to 7. An interrupt is acceptable when the interrupt request level is higher than the selected IPL.
When the processor interrupt priority level (IPL) is set to 111b (level 7), all interrupts are disabled.

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

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

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

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

#### 2.1.8.14 Reserved

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

## 2.2 Fast Interrupt Registers

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

### 2.2.1 Save Flag Register (SVF)

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

### 2.2.2 Save PC Register (SVP)

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

### 2.2.3 Vector Register (VCT)

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

## 2.3 DMAC-associated Registers

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

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

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

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

These 24-bit registers are used to set DMA transfer counting.

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

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

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

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

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

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

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

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

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

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

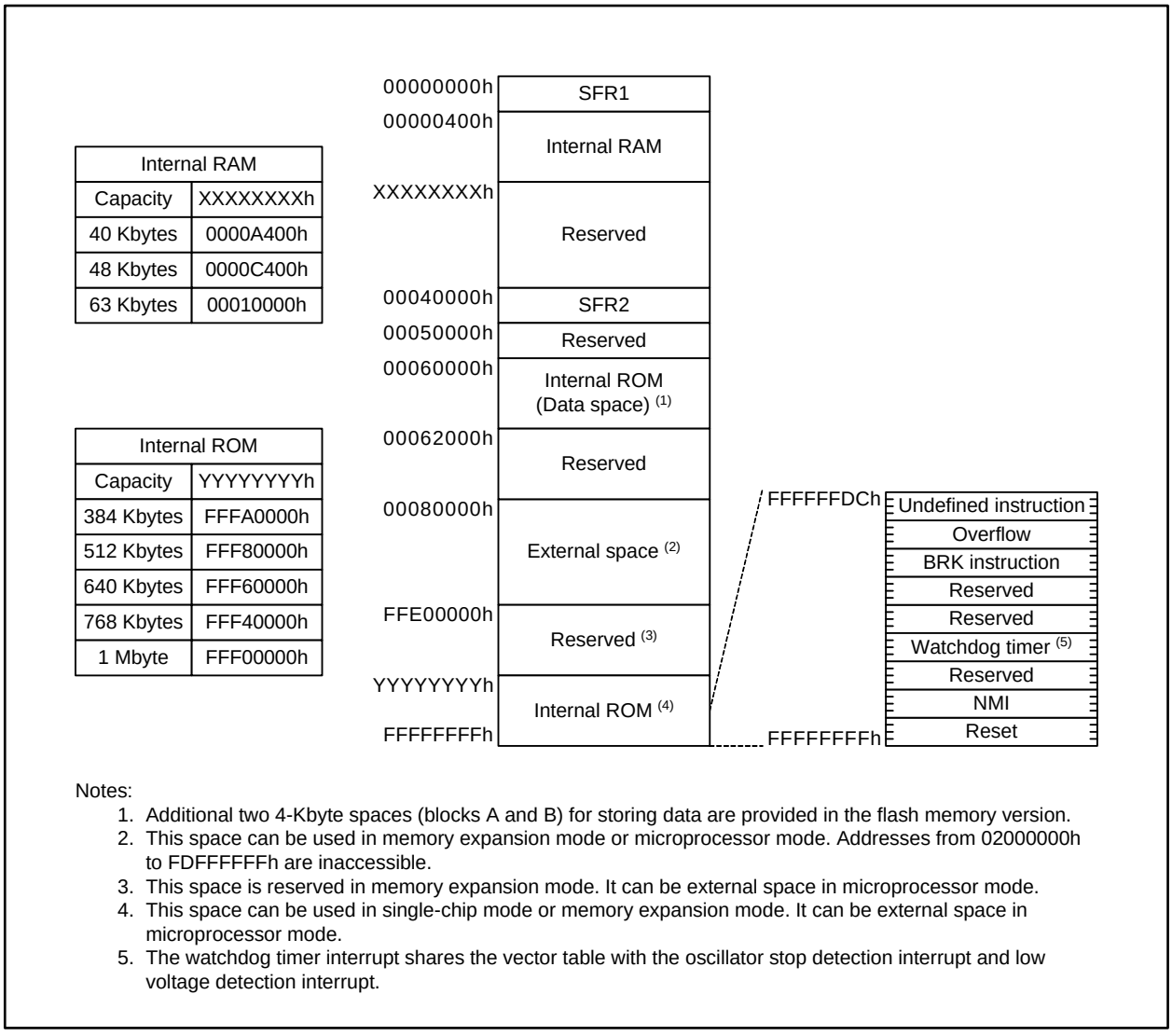
# 3. Memory

Figure 3.1 shows the memory map of the R32C/118 Group.
The R32C/118 Group provides a 4-Gbyte address space from 00000000h to FFFFFFFFh.

The internal ROM is mapped to the end of the memory map with the ending address fixed at FFFFFFFFh. Therefore, the 1-Mbyte internal ROM is mapped from FFF00000h to FFFFFFFFh.
The fixed interrupt vector table which contains each start address of interrupt handlers is mapped from FFFFFFDCh to FFFFFFFFh.
The internal RAM is mapped to the beginning of the memory map with the starting address fixed at 00000400h. Therefore, the 63-Kbyte internal RAM is mapped from 00000400h to 0000FFFFh. Besides being used for data storage, the internal RAM functions as a stack(s) for subroutines and/or interrupt handlers.

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

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




Notes:
1. Additional two 4-Kbyte spaces (blocks A and B) for storing data are provided in the flash memory version.
2. This space can be used in memory expansion mode or microprocessor mode. Addresses from 02000000h to FDFFFFFFh are inaccessible.
3. This space is reserved in memory expansion mode. It can be external space in microprocessor mode.
4. This space can be used in single-chip mode or memory expansion mode. It can be external space in microprocessor mode.
5. The watchdog timer interrupt shares the vector table with the oscillator stop detection interrupt and low voltage detection interrupt.

**Figure 3.1 Memory Map**

# 4. Special Function Registers (SFRs)

SFRs are memory-mapped peripheral registers that control the operation of peripherals. Table 4.1 SFR List (1) to Table 4.53 SFR List (53) list the SFR details.

**Table 4.1 SFR List (1)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.2 SFR List (2)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 000060h | | | |
| 000061h | Timer B5 Interrupt Control Register | TB5IC | XXXX X000b |
| 000062h | UART5 Transmit/NACK Interrupt Control Register | S5TIC | XXXX X000b |
| 000063h | UART2 Receive/ACK Interrupt Control Register/I2C Bus Line Interrupt Control Register | S2RIC/I2CLIC | XXXX X000b |
| 000064h | UART6 Transmit/NACK Interrupt Control Register | S6TIC | XXXX X000b |
| 000065h | UART3 Receive/ACK Interrupt Control Register | S3RIC | XXXX X000b |
| 000066h | UART5/6 Bus Collision, Start Condition/Stop Condition Detection Interrupt Control Register | BCN5IC/BCN6IC | XXXX X000b |
| 000067h | UART4 Receive/ACK Interrupt Control Register | S4RIC | XXXX X000b |
| 000068h | DMA0 Transfer Complete Interrupt Control Register | DM0IC | XXXX X000b |
| 000069h | UART0/3 Bus Collision, Start Condition/Stop Condition Detection Interrupt Control Register | BCN0IC/BCN3IC | XXXX X000b |
| 00006Ah | DMA2 Transfer Complete Interrupt Control Register | DM2IC | XXXX X000b |
| 00006Bh | A/D Converter 0 Convert Completion Interrupt Control Register | AD0IC | XXXX X000b |
| 00006Ch | Timer A0 Interrupt Control Register | TA0IC | XXXX X000b |
| 00006Dh | Intelligent I/O Interrupt Control Register 0 | IIO0IC | XXXX X000b |
| 00006Eh | Timer A2 Interrupt Control Register | TA2IC | XXXX X000b |
| 00006Fh | Intelligent I/O Interrupt Control Register 2 | IIO2IC | XXXX X000b |
| 000070h | Timer A4 Interrupt Control Register | TA4IC | XXXX X000b |
| 000071h | Intelligent I/O Interrupt Control Register 4 | IIO4IC | XXXX X000b |
| 000072h | UART0 Receive/ACK Interrupt Control Register | S0RIC | XXXX X000b |
| 000073h | Intelligent I/O Interrupt Control Register 6 | IIO6IC | XXXX X000b |
| 000074h | UART1 Receive/ACK Interrupt Control Register | S1RIC | XXXX X000b |
| 000075h | Intelligent I/O Interrupt Control Register 8 | IIO8IC | XXXX X000b |
| 000076h | Timer B1 Interrupt Control Register | TB1IC | XXXX X000b |
| 000077h | Intelligent I/O Interrupt Control Register 10 | IIO10IC | XXXX X000b |
| 000078h | Timer B3 Interrupt Control Register | TB3IC | XXXX X000b |
| 000079h | | | |
| 00007Ah | INT5 Interrupt Control Register | INT5IC | XX00 X000b |
| 00007Bh | CAN0 Wake-up Interrupt Control Register | C0WIC | XXXX X000b |
| 00007Ch | INT3 Interrupt Control Register | INT3IC | XX00 X000b |
| 00007Dh | | | |
| 00007Eh | INT1 Interrupt Control Register | INT1IC | XX00 X000b |
| 00007Fh | | | |
| 000080h | | | |
| 000081h | UART2 Transmit/NACK Interrupt Control Register/I2C-Bus Interrupt Control Register | S2TIC/I2CIC | XXXX X000b |
| 000082h | UART5 Receive/ACK Interrupt Control Register | S5RIC | XXXX X000b |
| 000083h | UART3 Transmit/NACK Interrupt Control Register | S3TIC | XXXX X000b |
| 000084h | UART6 Receive/ACK Interrupt Control Register | S6RIC | XXXX X000b |
| 000085h | UART4 Transmit/NACK Interrupt Control Register | S4TIC | XXXX X000b |
| 000086h | | | |
| 000087h | UART2 Bus Collision, Start Condition/Stop Condition Detection Interrupt Control Register | BCN2IC | XXXX X000b |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.3 SFR List (3)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 000088h | DMA1 Transfer Complete Interrupt Control Register | DM1IC | XXXX X000b |
| 000089h | UART1/4 Bus Collision, Start Condition/Stop Condition Detection Interrupt Control Register | BCN1IC/BCN4IC | XXXX X000b |
| 00008Ah | DMA3 Transfer Complete Interrupt Control Register | DM3IC | XXXX X000b |
| 00008Bh | Key Input Interrupt Control Register | KUPIC | XXXX X000b |
| 00008Ch | Timer A1 Interrupt Control Register | TA1IC | XXXX X000b |
| 00008Dh | Intelligent I/O Interrupt Control Register 1 | IIO1IC | XXXX X000b |
| 00008Eh | Timer A3 Interrupt Control Register | TA3IC | XXXX X000b |
| 00008Fh | Intelligent I/O Interrupt Control Register 3 | IIO3IC | XXXX X000b |
| 000090h | UART0 Transmit/NACK Interrupt Control Register | S0TIC | XXXX X000b |
| 000091h | Intelligent I/O Interrupt Control Register 5 | IIO5IC | XXXX X000b |
| 000092h | UART1 Transmit/NACK Interrupt Control Register | S1TIC | XXXX X000b |
| 000093h | Intelligent I/O Interrupt Control Register 7 | IIO7IC | XXXX X000b |
| 000094h | Timer B0 Interrupt Control Register | TB0IC | XXXX X000b |
| 000095h | Intelligent I/O Interrupt Control Register 9 | IIO9IC | XXXX X000b |
| 000096h | Timer B2 Interrupt Control Register | TB2IC | XXXX X000b |
| 000097h | Intelligent I/O Interrupt Control Register 11 | IIO11IC | XXXX X000b |
| 000098h | Timer B4 Interrupt Control Register | TB4IC | XXXX X000b |
| 000099h | | | |
| 00009Ah | INT4 Interrupt Control Register | INT4IC | XX00 X000b |
| 00009Bh | CAN1 Wake-up Interrupt Control Register | C1WIC | XXXX X000b |
| 00009Ch | INT2 Interrupt Control Register | INT2IC | XX00 X000b |
| 00009Dh | | | |
| 00009Eh | INT0 Interrupt Control Register | INT0IC | XX00 X000b |
| 00009Fh | | | |
| 0000A0h | Intelligent I/O Interrupt Request Register 0 | IIO0IR | 0000 0XX1b |
| 0000A1h | Intelligent I/O Interrupt Request Register 1 | IIO1IR | 0000 0XX1b |
| 0000A2h | Intelligent I/O Interrupt Request Register 2 | IIO2IR | 0000 0X01b |
| 0000A3h | Intelligent I/O Interrupt Request Register 3 | IIO3IR | 0000 XXX1b |
| 0000A4h | Intelligent I/O Interrupt Request Register 4 | IIO4IR | 000X 0XX1b |
| 0000A5h | Intelligent I/O Interrupt Request Register 5 | IIO5IR | 000X 0XX1b |
| 0000A6h | Intelligent I/O Interrupt Request Register 6 | IIO6IR | 000X 0XX1b |
| 0000A7h | Intelligent I/O Interrupt Request Register 7 | IIO7IR | X00X 0XX1b |
| 0000A8h | Intelligent I/O Interrupt Request Register 8 | IIO8IR | XX0X 0XX1b |
| 0000A9h | Intelligent I/O Interrupt Request Register 9 | IIO9IR | 0X00 0XX1b |
| 0000AAh | Intelligent I/O Interrupt Request Register 10 | IIO10IR | 0X00 0XX1b |
| 0000ABh | Intelligent I/O Interrupt Request Register 11 | IIO11IR | 0X00 0XX1b |
| 0000ACh | | | |
| 0000ADh | | | |
| 0000AEh | | | |
| 0000AFh | | | |

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.4 SFR List (4)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 0000B0h | Intelligent I/O Interrupt Enable Register 0 | IIO0IE | 00h |
| 0000B1h | Intelligent I/O Interrupt Enable Register 1 | IIO1IE | 00h |
| 0000B2h | Intelligent I/O Interrupt Enable Register 2 | IIO2IE | 00h |
| 0000B3h | Intelligent I/O Interrupt Enable Register 3 | IIO3IE | 00h |
| 0000B4h | Intelligent I/O Interrupt Enable Register 4 | IIO4IE | 00h |
| 0000B5h | Intelligent I/O Interrupt Enable Register 5 | IIO5IE | 00h |
| 0000B6h | Intelligent I/O Interrupt Enable Register 6 | IIO6IE | 00h |
| 0000B7h | Intelligent I/O Interrupt Enable Register 7 | IIO7IE | 00h |
| 0000B8h | Intelligent I/O Interrupt Enable Register 8 | IIO8IE | 00h |
| 0000B9h | Intelligent I/O Interrupt Enable Register 9 | IIO9IE | 00h |
| 0000BAh | Intelligent I/O Interrupt Enable Register 10 | IIO10IE | 00h |
| 0000BBh | Intelligent I/O Interrupt Enable Register 11 | IIO11IE | 00h |
| 0000BCh | | | |
| 0000BDh | | | |
| 0000BEh | | | |
| 0000BFh | | | |
| 0000C0h | | | |
| 0000C1h | CAN0 Transmit Interrupt Control Register | C0TIC | XXXX X000b |
| 0000C2h | | | |
| 0000C3h | CAN0 Error Interrupt Control Register | C0EIC | XXXX X000b |
| 0000C4h | | | |
| 0000C5h | CAN1 Receive Interrupt Control Register | C1RIC | XXXX X000b |
| 0000C6h | | | |
| 0000C7h | | | |
| 0000C8h | | | |
| 0000C9h | | | |
| 0000CAh | | | |
| 0000CBh | | | |
| 0000CCh | | | |
| 0000CDh | | | |
| 0000CEh | | | |
| 0000CFh | | | |
| 0000D0h | CAN0 Transmit FIFO Interrupt Control Register | C0FTIC | XXXX X000b |
| 0000D1h | | | |
| 0000D2h | CAN1 Transmit FIFO Interrupt Control Register | C1FTIC | XXXX X000b |
| 0000D3h | | | |
| 0000D4h | | | |
| 0000D5h | | | |
| 0000D6h | | | |
| 0000D7h | | | |
| 0000D8h | | | |
| 0000D9h | | | |
| 0000DAh | | | |
| 0000DBh | | | |
| 0000DCh | | | |
| 0000DDh | UART7 Transmit Interrupt Control Register | S7TIC | XXXX X000b |
| 0000DEh | INT7 Interrupt Control Register | INT7IC | XX00 X000b |
| 0000DFh | UART8 Transmit Interrupt Control Register | S8TIC | XXXX X000b |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.5 SFR List (5)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 0000E0h | | | |
| 0000E1h | CAN0 Receive Interrupt Control Register | C0RIC | XXXX X000b |
| 0000E2h | | | |
| 0000E3h | CAN1 Transmit Interrupt Control Register | C1TIC | XXXX X000b |
| 0000E4h | | | |
| 0000E5h | CAN1 Error Interrupt Control Register | C1EIC | XXXX X000b |
| 0000E6h | | | |
| 0000E7h | | | |
| 0000E8h | | | |
| 0000E9h | | | |
| 0000EAh | | | |
| 0000EBh | | | |
| 0000ECh | | | |
| 0000EDh | | | |
| 0000EEh | | | |
| 0000EFh | | | |
| 0000F0h | CAN0 Receive FIFO Interrupt Control Register | C0FRIC | XXXX X000b |
| 0000F1h | | | |
| 0000F2h | CAN1 Receive FIFO Interrupt Control Register | C1FRIC | XXXX X000b |
| 0000F3h | | | |
| 0000F4h | | | |
| 0000F5h | | | |
| 0000F6h | | | |
| 0000F7h | | | |
| 0000F8h | | | |
| 0000F9h | | | |
| 0000FAh | | | |
| 0000FBh | | | |
| 0000FCh | INT8 Interrupt Control Register | INT8IC | XX00 X000b |
| 0000FDh | UART7 Receive Interrupt Control Register | S7RIC | XXXX X000b |
| 0000FEh | INT6 Interrupt Control Register | INT6IC | XX00 X000b |
| 0000FFh | UART8 Receive Interrupt Control Register | S8RIC | XXXX X000b |
| 000100h | Group 1 Time Measurement/Waveform Generation Register 0 | G1TM0/G1PO0 | XXXXh |
| 000101h | | | |
| 000102h | Group 1 Time Measurement/Waveform Generation Register 1 | G1TM1/G1PO1 | XXXXh |
| 000103h | | | |
| 000104h | Group 1 Time Measurement/Waveform Generation Register 2 | G1TM2/G1PO2 | XXXXh |
| 000105h | | | |
| 000106h | Group 1 Time Measurement/Waveform Generation Register 3 | G1TM3/G1PO3 | XXXXh |
| 000107h | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.6 SFR List (6)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.7 SFR List (7)**

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

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.8 SFR List (8)**

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

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.9 SFR List (9)**

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

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.10 SFR List (10)**

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

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.11 SFR List (11)**

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

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.12 SFR List (12)**

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

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.13 SFR List (13)**

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

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.14 SFR List (14)**

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

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.15 SFR List (15)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 000380h | A/D0 Register 0 | AD00 | 00XXh |
| 000381h | | | |
| 000382h | A/D0 Register 1 | AD01 | 00XXh |
| 000383h | | | |
| 000384h | A/D0 Register 2 | AD02 | 00XXh |
| 000385h | | | |
| 000386h | A/D0 Register 3 | AD03 | 00XXh |
| 000387h | | | |
| 000388h | A/D0 Register 4 | AD04 | 00XXh |
| 000389h | | | |
| 00038Ah | A/D0 Register 5 | AD05 | 00XXh |
| 00038Bh | | | |
| 00038Ch | A/D0 Register 6 | AD06 | 00XXh |
| 00038Dh | | | |
| 00038Eh | A/D0 Register 7 | AD07 | 00XXh |
| 00038Fh | | | |
| 000390h | | | |
| 000391h | | | |
| 000392h | A/D0 Control Register 4 | AD0CON4 | XXXX 00XXb |
| 000393h | | | |
| 000394h | A/D0 Control Register 2 | AD0CON2 | X00X X000b |
| 000395h | A/D0 Control Register 3 | AD0CON3 | XXXX X000b |
| 000396h | A/D0 Control Register 0 | AD0CON0 | 00h |
| 000397h | A/D0 Control Register 1 | AD0CON1 | 00h |
| 000398h | D/A Register 0 | DA0 | XXh |
| 000399h | | | |
| 00039Ah | D/A Register 1 | DA1 | XXh |
| 00039Bh | | | |
| 00039Ch | D/A Control Register | DACON | XXXX XX00b |
| 00039Dh | | | |
| 00039Eh | | | |
| 00039Fh | | | |
| 0003A0h | | | |
| 0003A1h | | | |
| 0003A2h | | | |
| 0003A3h | | | |
| 0003A4h | | | |
| 0003A5h | | | |
| 0003A6h | | | |
| 0003A7h | | | |
| 0003A8h | | | |
| 0003A9h | | | |
| 0003AAh | | | |
| 0003ABh | | | |
| 0003ACh | | | |
| 0003ADh | | | |
| 0003AEh | | | |
| 0003AFh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.16 SFR List (16)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 0003B0h | | | |
| 0003B1h | | | |
| 0003B2h | | | |
| 0003B3h | | | |
| 0003B4h | | | |
| 0003B5h | | | |
| 0003B6h | | | |
| 0003B7h | | | |
| 0003B8h | | | |
| 0003B9h | | | |
| 0003BAh | | | |
| 0003BBh | | | |
| 0003BCh | | | |
| 0003BDh | | | |
| 0003BEh | | | |
| 0003BFh | | | |
| 0003C0h | Port P0 Register | P0 | XXh |
| 0003C1h | Port P1 Register | P1 | XXh |
| 0003C2h | Port P0 Direction Register | PD0 | 0000 0000b |
| 0003C3h | Port P1 Direction Register | PD1 | 0000 0000b |
| 0003C4h | Port P2 Register | P2 | XXh |
| 0003C5h | Port P3 Register | P3 | XXh |
| 0003C6h | Port P2 Direction Register | PD2 | 0000 0000b |
| 0003C7h | Port P3 Direction Register | PD3 | 0000 0000b |
| 0003C8h | Port P4 Register | P4 | XXh |
| 0003C9h | Port P5 Register | P5 | XXh |
| 0003CAh | Port P4 Direction Register | PD4 | 0000 0000b |
| 0003CBh | Port P5 Direction Register | PD5 | 0000 0000b |
| 0003CCh | Port P6 Register | P6 | XXh |
| 0003CDh | Port P7 Register | P7 | XXh |
| 0003CEh | Port P6 Direction Register | PD6 | 0000 0000b |
| 0003CFh | Port P7 Direction Register | PD7 | 0000 0000b |
| 0003D0h | Port P8 Register | P8 | XXh |
| 0003D1h | Port P9 Register | P9 | XXh |
| 0003D2h | Port P8 Direction Register | PD8 | 00X0 0000b |
| 0003D3h | Port P9 Direction Register | PD9 | 0000 0000b |
| 0003D4h | Port P10 Register | P10 | XXh |
| 0003D5h | Port P11 Register | P11 | XXh |
| 0003D6h | Port P10 Direction Register | PD10 | 0000 0000b |
| 0003D7h | Port P11 Direction Register | PD11 | XXX0 0000b |
| 0003D8h | Port P12 Register | P12 | XXh |
| 0003D9h | Port P13 Register | P13 | XXh |
| 0003DAh | Port P12 Direction Register | PD12 | 0000 0000b |
| 0003DBh | Port P13 Direction Register | PD13 | 0000 0000b |
| 0003DCh | Port P14 Register | P14 | XXh |
| 0003DDh | Port P15 Register | P15 | XXh |
| 0003DEh | Port P14 Direction Register | PD14 | X000 0000b |
| 0003DFh | Port P15 Direction Register | PD15 | 0000 0000b |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.17 SFR List (17)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 0003E0h | | | |
| 0003E1h | | | |
| 0003E2h | | | |
| 0003E3h | | | |
| 0003E4h | | | |
| 0003E5h | | | |
| 0003E6h | | | |
| 0003E7h | | | |
| 0003E8h | | | |
| 0003E9h | | | |
| 0003EAh | | | |
| 0003EBh | | | |
| 0003ECh | | | |
| 0003EDh | | | |
| 0003EEh | | | |
| 0003EFh | | | |
| 0003F0h | Pull-up Control Register 0 | PUR0 | 0000 0000b |
| 0003F1h | Pull-up Control Register 1 | PUR1 | XXXX X0XXb |
| 0003F2h | Pull-up Control Register 2 | PUR2 | 000X XXXXb |
| 0003F3h | Pull-up Control Register 3 | PUR3 | 0000 0000b |
| 0003F4h | Pull-up Control Register 4 | PUR4 | XXXX 0000b |
| 0003F5h | | | |
| 0003F6h | | | |
| 0003F7h | | | |
| 0003F8h | | | |
| 0003F9h | | | |
| 0003FAh | | | |
| 0003FBh | | | |
| 0003FCh | | | |
| 0003FDh | | | |
| 0003FEh | | | |
| 0003FFh | Port Control Register | PCR | 0XXX XXX0b |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.18 SFR List (18)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 040000h | Flash Memory Control Register 0 | FMR0 | 0X01 XX00b |
| 040001h | Flash Memory Status Register 0 | FMSR0 | 1000 0000b |
| 040002h | | | |
| 040003h | | | |
| 040004h | | | |
| 040005h | | | |
| 040006h | | | |
| 040007h | | | |
| 040008h | Flash Register Protection Unlock Register 0 | FPR0 | 00h |
| 040009h | Flash Memory Control Register 1 | FMR1 | 0000 0010b |
| 04000Ah | Block Protect Bit Monitor Register 0 | FBPM0 | ??X? ????b (1) |
| 04000Bh | Block Protect Bit Monitor Register 1 | FBPM1 | XXX? ????b (1) |
| 04000Ch | | | |
| 04000Dh | | | |
| 04000Eh | | | |
| 04000Fh | | | |
| 040010h | | | |
| 040011h | Block Protect Bit Monitor Register 2 | FBPM2 | ???? ????b (1) |
| 040012h | | | |
| 040013h | | | |
| 040014h | | | |
| 040015h | | | |
| 040016h | | | |
| 040017h | | | |
| 040018h | | | |
| 040019h | | | |
| 04001Ah | | | |
| 04001Bh | | | |
| 04001Ch | | | |
| 04001Dh | | | |
| 04001Eh | | | |
| 04001Fh | | | |
| 040020h | PLL Control Register 0 | PLC0 | 0000 0001b |
| 040021h | PLL Control Register 1 | PLC1 | 0001 1111b |
| 040022h | | | |
| 040023h | | | |
| 040024h | | | |
| 040025h | | | |
| 040026h | | | |
| 040027h | | | |
| 040028h | | | |
| 040029h | | | |
| 04002Ah | | | |
| 04002Bh | | | |
| 04002Ch | | | |
| 04002Dh | | | |
| 04002Eh | | | |
| 04002Fh | | | |

X: Undefined

Blanks are reserved. No access is allowed.

Note:

1. The status of protect bit of each block in flash memory is reflected.

**Table 4.19 SFR List (19)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 040030h to 04003Fh | | | |
| 040040h | | | |
| 040041h | | | |
| 040042h | | | |
| 040043h | | | |
| 040044h | Processor Mode Register 0 (1) | PM0 | 1000 0000b (CNVSS pin = Low)<br>0000 0011b (CNVSS pin = High) |
| 040045h | | | |
| 040046h | System Clock Control Register 0 | CM0 | 0000 1000b |
| 040047h | System Clock Control Register 1 | CM1 | 0010 0000b |
| 040048h | Processor Mode Register 3 | PM3 | 00h |
| 040049h | | | |
| 04004Ah | Protect Register | PRCR | XXXX X000b |
| 04004Bh | | | |
| 04004Ch | Protect Register 3 | PRCR3 | 0000 0000b |
| 04004Dh | Oscillator Stop Detection Register | CM2 | 00h |
| 04004Eh | | | |
| 04004Fh | | | |
| 040050h | | | |
| 040051h | | | |
| 040052h | | | |
| 040053h | Processor Mode Register 2 | PM2 | 00h |
| 040054h | Chip Select Output Pin Setting Register 0 | CSOP0 | 1000 XXXXb |
| 040055h | Chip Select Output Pin Setting Register 1 | CSOP1 | 01X0 XXXXb |
| 040056h | Chip Select Output Pin Setting Register 2 | CSOP2 | XXXX 0000b |
| 040057h | | | |
| 040058h | | | |
| 040059h | | | |
| 04005Ah | Low Speed Mode Clock Control Register | CM3 | XXXX XX00b |
| 04005Bh | | | |
| 04005Ch | | | |
| 04005Dh | | | |
| 04005Eh | | | |
| 04005Fh | | | |
| 040060h | Voltage Regulator Control Register | VRCR | 0000 0000b |
| 040061h | | | |
| 040062h | Low Voltage Detector Control Register | LVDC | 0000 XX00b |
| 040063h | | | |
| 040064h | Detection Voltage Configuration Register | DVCR | 0000 XXXXb |
| 040065h | | | |
| 040066h | | | |
| 040067h | | | |
| 040068h to 040093h | | | |

X: Undefined
Blanks are reserved. No access is allowed.

Note:
1. The value in the PM0 register remains unchanged even after a software reset or watchdog timer reset.

**Table 4.20 SFR List (20)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 040094h | | | |
| 040095h | | | |
| 040096h | | | |
| 040097h | Three-phase Output Buffer Control Register | IOBC | 0XXX XXXXb |
| 040098h | Input Function Select Register 0 | IFS0 | X000 0000b |
| 040099h | Input Function Select Register 1 | IFS1 | XXXX X0X0b |
| 04009Ah | Input Function Select Register 2 | IFS2 | 0000 00X0b |
| 04009Bh | Input Function Select Register 3 | IFS3 | XXXX XX00b |
| 04009Ch | | | |
| 04009Dh | | | |
| 04009Eh | | | |
| 04009Fh | | | |
| 0400A0h | Port P0_0 Function Select Register | P0_0S | 0XXX X000b |
| 0400A1h | Port P1_0 Function Select Register | P1_0S | XXXX X000b |
| 0400A2h | Port P0_1 Function Select Register | P0_1S | 0XXX X000b |
| 0400A3h | Port P1_1 Function Select Register | P1_1S | XXXX X000b |
| 0400A4h | Port P0_2 Function Select Register | P0_2S | 0XXX X000b |
| 0400A5h | Port P1_2 Function Select Register | P1_2S | XXXX X000b |
| 0400A6h | Port P0_3 Function Select Register | P0_3S | 0XXX X000b |
| 0400A7h | Port P1_3 Function Select Register | P1_3S | XXXX X000b |
| 0400A8h | Port P0_4 Function Select Register | P0_4S | 0XXX X000b |
| 0400A9h | Port P1_4 Function Select Register | P1_4S | XXXX X000b |
| 0400AAh | Port P0_5 Function Select Register | P0_5S | 0XXX X000b |
| 0400ABh | Port P1_5 Function Select Register | P1_5S | XXXX X000b |
| 0400ACh | Port P0_6 Function Select Register | P0_6S | 0XXX X000b |
| 0400ADh | Port P1_6 Function Select Register | P1_6S | XXXX X000b |
| 0400AEh | Port P0_7 Function Select Register | P0_7S | 0XXX X000b |
| 0400AFh | Port P1_7 Function Select Register | P1_7S | XXXX X000b |
| 0400B0h | Port P2_0 Function Select Register | P2_0S | 0XXX X000b |
| 0400B1h | Port P3_0 Function Select Register | P3_0S | XXXX X000b |
| 0400B2h | Port P2_1 Function Select Register | P2_1S | 0XXX X000b |
| 0400B3h | Port P3_1 Function Select Register | P3_1S | XXXX X000b |
| 0400B4h | Port P2_2 Function Select Register | P2_2S | 0XXX X000b |
| 0400B5h | Port P3_2 Function Select Register | P3_2S | XXXX X000b |
| 0400B6h | Port P2_3 Function Select Register | P2_3S | 0XXX X000b |
| 0400B7h | Port P3_3 Function Select Register | P3_3S | XXXX X000b |
| 0400B8h | Port P2_4 Function Select Register | P2_4S | 0XXX X000b |
| 0400B9h | Port P3_4 Function Select Register | P3_4S | XXXX X000b |
| 0400BAh | Port P2_5 Function Select Register | P2_5S | 0XXX X000b |
| 0400BBh | Port P3_5 Function Select Register | P3_5S | XXXX X000b |
| 0400BCh | Port P2_6 Function Select Register | P2_6S | 0XXX X000b |
| 0400BDh | Port P3_6 Function Select Register | P3_6S | XXXX X000b |
| 0400BEh | Port P2_7 Function Select Register | P2_7S | 0XXX X000b |
| 0400BFh | Port P3_7 Function Select Register | P3_7S | XXXX X000b |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.21 SFR List (21)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 0400C0h | Port P4_0 Function Select Register | P4_0S | X0XX X000b |
| 0400C1h | Port P5_0 Function Select Register | P5_0S | XXXX X000b |
| 0400C2h | Port P4_1 Function Select Register | P4_1S | X0XX X000b |
| 0400C3h | Port P5_1 Function Select Register | P5_1S | XXXX X000b |
| 0400C4h | Port P4_2 Function Select Register | P4_2S | X0XX X000b |
| 0400C5h | Port P5_2 Function Select Register | P5_2S | XXXX X000b |
| 0400C6h | Port P4_3 Function Select Register | P4_3S | X0XX X000b |
| 0400C7h | Port P5_3 Function Select Register | P5_3S | XXXX X000b |
| 0400C8h | Port P4_4 Function Select Register | P4_4S | X0XX X000b |
| 0400C9h | Port P5_4 Function Select Register | P5_4S | X0XX X000b |
| 0400CAh | Port P4_5 Function Select Register | P4_5S | X0XX X000b |
| 0400CBh | Port P5_5 Function Select Register | P5_5S | X0XX X000b |
| 0400CCh | Port P4_6 Function Select Register | P4_6S | X0XX X000b |
| 0400CDh | Port P5_6 Function Select Register | P5_6S | X0XX X000b |
| 0400CEh | Port P4_7 Function Select Register | P4_7S | X0XX X000b |
| 0400CFh | Port P5_7 Function Select Register | P5_7S | X0XX X000b |
| 0400D0h | Port P6_0 Function Select Register | P6_0S | X0XX X000b |
| 0400D1h | Port P7_0 Function Select Register | P7_0S | X0XX X000b |
| 0400D2h | Port P6_1 Function Select Register | P6_1S | X0XX X000b |
| 0400D3h | Port P7_1 Function Select Register | P7_1S | X0XX X000b |
| 0400D4h | Port P6_2 Function Select Register | P6_2S | X0XX X000b |
| 0400D5h | Port P7_2 Function Select Register | P7_2S | X0XX X000b |
| 0400D6h | Port P6_3 Function Select Register | P6_3S | X0XX X000b |
| 0400D7h | Port P7_3 Function Select Register | P7_3S | X0XX X000b |
| 0400D8h | Port P6_4 Function Select Register | P6_4S | X0XX X000b |
| 0400D9h | Port P7_4 Function Select Register | P7_4S | X0XX X000b |
| 0400DAh | Port P6_5 Function Select Register | P6_5S | X0XX X000b |
| 0400DBh | Port P7_5 Function Select Register | P7_5S | X0XX X000b |
| 0400DCh | Port P6_6 Function Select Register | P6_6S | X0XX X000b |
| 0400DDh | Port P7_6 Function Select Register | P7_6S | X0XX X000b |
| 0400DEh | Port P6_7 Function Select Register | P6_7S | X0XX X000b |
| 0400DFh | Port P7_7 Function Select Register | P7_7S | X0XX X000b |
| 0400E0h | Port P8_0 Function Select Register | P8_0S | X0XX X000b |
| 0400E1h | Port P9_0 Function Select Register | P9_0S | X0XX X000b |
| 0400E2h | Port P8_1 Function Select Register | P8_1S | X0XX X000b |
| 0400E3h | Port P9_1 Function Select Register | P9_1S | X0XX X000b |
| 0400E4h | Port P8_2 Function Select Register | P8_2S | X0XX X000b |
| 0400E5h | Port P9_2 Function Select Register | P9_2S | X0XX X000b |
| 0400E6h | Port P8_3 Function Select Register | P8_3S | X0XX X000b |
| 0400E7h | Port P9_3 Function Select Register | P9_3S | 00XX X000b |
| 0400E8h | Port P8_4 Function Select Register | P8_4S | XXXX X000b |
| 0400E9h | Port P9_4 Function Select Register | P9_4S | 00XX X000b |
| 0400EAh | | | |
| 0400EBh | Port P9_5 Function Select Register | P9_5S | 00XX X000b |
| 0400ECh | Port P8_6 Function Select Register | P8_6S | XXXX X000b |
| 0400EDh | Port P9_6 Function Select Register | P9_6S | 00XX X000b |
| 0400EEh | Port P8_7 Function Select Register | P8_7S | XXXX X000b |
| 0400EFh | Port P9_7 Function Select Register | P9_7S | X0XX X000b |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.22 SFR List (22)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 0400F0h | Port P10_0 Function Select Register | P10_0S | 0XXX X000b |
| 0400F1h | Port P11_0 Function Select Register | P11_0S | X0XX X000b |
| 0400F2h | Port P10_1 Function Select Register | P10_1S | 0XXX X000b |
| 0400F3h | Port P11_1 Function Select Register | P11_1S | X0XX X000b |
| 0400F4h | Port P10_2 Function Select Register | P10_2S | 0XXX X000b |
| 0400F5h | Port P11_2 Function Select Register | P11_2S | X0XX X000b |
| 0400F6h | Port P10_3 Function Select Register | P10_3S | 0XXX X000b |
| 0400F7h | Port P11_3 Function Select Register | P11_3S | X0XX X000b |
| 0400F8h | Port P10_4 Function Select Register | P10_4S | 0XXX X000b |
| 0400F9h | Port P11_4 Function Select Register | P11_4S | XXXX X000b |
| 0400FAh | Port P10_5 Function Select Register | P10_5S | 0XXX X000b |
| 0400FBh | | | |
| 0400FCh | Port P10_6 Function Select Register | P10_6S | 0XXX X000b |
| 0400FDh | | | |
| 0400FEh | Port P10_7 Function Select Register | P10_7S | 0XXX X000b |
| 0400FFh | | | |
| 040100h | Port P12_0 Function Select Register | P12_0S | X0XX X000b |
| 040101h | Port P13_0 Function Select Register | P13_0S | XXXX X000b |
| 040102h | Port P12_1 Function Select Register | P12_1S | X0XX X000b |
| 040103h | Port P13_1 Function Select Register | P13_1S | XXXX X000b |
| 040104h | Port P12_2 Function Select Register | P12_2S | X0XX X000b |
| 040105h | Port P13_2 Function Select Register | P13_2S | XXXX X000b |
| 040106h | Port P12_3 Function Select Register | P12_3S | X0XX X000b |
| 040107h | Port P13_3 Function Select Register | P13_3S | XXXX X000b |
| 040108h | Port P12_4 Function Select Register | P12_4S | XXXX X000b |
| 040109h | Port P13_4 Function Select Register | P13_4S | XXXX X000b |
| 04010Ah | Port P12_5 Function Select Register | P12_5S | XXXX X000b |
| 04010Bh | Port P13_5 Function Select Register | P13_5S | XXXX X000b |
| 04010Ch | Port P12_6 Function Select Register | P12_6S | XXXX X000b |
| 04010Dh | Port P13_6 Function Select Register | P13_6S | XXXX X000b |
| 04010Eh | Port P12_7 Function Select Register | P12_7S | XXXX X000b |
| 04010Fh | Port P13_7 Function Select Register | P13_7S | XXXX X000b |
| 040110h | | | |
| 040111h | Port P15_0 Function Select Register | P15_0S | 00XX X000b |
| 040112h | | | |
| 040113h | Port P15_1 Function Select Register | P15_1S | 00XX X000b |
| 040114h | | | |
| 040115h | Port P15_2 Function Select Register | P15_2S | 00XX X000b |
| 040116h | Port P14_3 Function Select Register | P14_3S | XXXX X000b |
| 040117h | Port P15_3 Function Select Register | P15_3S | 00XX X000b |
| 040118h | Port P14_4 Function Select Register | P14_4S | XXXX X000b |
| 040119h | Port P15_4 Function Select Register | P15_4S | 00XX X000b |
| 04011Ah | Port P14_5 Function Select Register | P14_5S | XXXX X000b |
| 04011Bh | Port P15_5 Function Select Register | P15_5S | 00XX X000b |
| 04011Ch | Port P14_6 Function Select Register | P14_6S | XXXX X000b |
| 04011Dh | Port P15_6 Function Select Register | P15_6S | 00XX X000b |
| 04011Eh | | | |
| 04011Fh | Port P15_7 Function Select Register | P15_7S | 00XX X000b |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.23 SFR List (23)**

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

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.24 SFR List (24)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 044060h | | | |
| 044061h | | | |
| 044062h | | | |
| 044063h | | | |
| 044064h | | | |
| 044065h | | | |
| 044066h | | | |
| 044067h | | | |
| 044068h | | | |
| 044069h | | | |
| 04406Ah | | | |
| 04406Bh | | | |
| 04406Ch | | | |
| 04406Dh | External Interrupt Source Select Register 1 | IFSR1 | X0XX X000b |
| 04406Eh | | | |
| 04406Fh | External Interrupt Source Select Register 0 | IFSR0 | 0000 0000b |
| 044070h | DMA0 Request Source Select Register 2 | DM0SL2 | XX00 0000b |
| 044071h | DMA1 Request Source Select Register 2 | DM1SL2 | XX00 0000b |
| 044072h | DMA2 Request Source Select Register 2 | DM2SL2 | XX00 0000b |
| 044073h | DMA3 Request Source Select Register 2 | DM3SL2 | XX00 0000b |
| 044074h | | | |
| 044075h | | | |
| 044076h | | | |
| 044077h | | | |
| 044078h | DMA0 Request Source Select Register | DM0SL | XXX0 0000b |
| 044079h | DMA1 Request Source Select Register | DM1SL | XXX0 0000b |
| 04407Ah | DMA2 Request Source Select Register | DM2SL | XXX0 0000b |
| 04407Bh | DMA3 Request Source Select Register | DM3SL | XXX0 0000b |
| 04407Ch | | | |
| 04407Dh | Wake-up IPL Setting Register 2 | RIPL2 | XX0X 0000b |
| 04407Eh | | | |
| 04407Fh | Wake-up IPL Setting Register 1 | RIPL1 | XX0X 0000b |
| 044080h | | | |
| 044081h | | | |
| 044082h | | | |
| 044083h | | | |
| 044084h | | | |
| 044085h | | | |
| 044086h | | | |
| 044087h | | | |
| 044088h | | | |
| 044089h | | | |
| 04408Ah | | | |
| 04408Bh | | | |
| 04408Ch | | | |
| 04408Dh | | | |
| 04408Eh | | | |
| 04408Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.25 SFR List (25)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 044090h to 0443FFh | | | |
| 044400h | I²C Bus Transmit/Receive Shift Register | I2CTRSR | XXh |
| 044401h | | | |
| 044402h | I²C Bus Slave Address Register | I2CSAR | 00h |
| 044403h | I²C Bus Control Register 0 | I2CCR0 | 0000 0000b |
| 044404h | I²C Bus Clock Control Register | I2CCCR | 0000 0000b |
| 044405h | I²C Bus START Condition/STOP Condition Control Register | I2CSSCR | 0000 0000b |
| 044406h | I²C Bus Control Register 1 | I2CCR1 | 0000 0000b |
| 044407h | I²C Bus Control Register 2 | I2CCR2 | 0000 0000b |
| 044408h | I²C Bus Status Register | I2CSR | 0000 0000b |
| 044409h | | | |
| 04440Ah | | | |
| 04440Bh | | | |
| 04440Ch | | | |
| 04440Dh | | | |
| 04440Eh | | | |
| 04440Fh | | | |
| 044410h | I²C Bus Mode Register | I2CMR | 0000 0000b |
| 044411h | | | |
| 044412h | | | |
| 044413h | | | |
| 044414h | | | |
| 044415h | | | |
| 044416h | | | |
| 044417h | | | |
| 044418h | | | |
| 044419h | | | |
| 04441Ah | | | |
| 04441Bh | | | |
| 04441Ch | | | |
| 04441Dh | | | |
| 04441Eh | | | |
| 04441Fh | | | |
| 044420h to 0467FFh | | | |

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.26 SFR List (26)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 046800h to 0477FFh | | | |
| 047800h | CAN1 Mailbox 0: Message Identifier | C1MB0 | XXXX XXXXh |
| 047801h | | | |
| 047802h | | | |
| 047803h | | | |
| 047804h | | | |
| 047805h | CAN1 Mailbox 0: Data Length | | XXh |
| 047806h | CAN1 Mailbox 0: Data Field | | XXXX XXXX |
| 047807h | | | XXXX XXXXh |
| 047808h | | | |
| 047809h | | | |
| 04780Ah | | | |
| 04780Bh | | | |
| 04780Ch | | | |
| 04780Dh | | | |
| 04780Eh | CAN1 Mailbox 0: Time Stamp | | XXXXh |
| 04780Fh | | | |
| 047810h | CAN1 Mailbox 1: Message Identifier | C1MB1 | XXXX XXXXh |
| 047811h | | | |
| 047812h | | | |
| 047813h | | | |
| 047814h | | | |
| 047815h | CAN1 Mailbox 1: Data Length | | XXh |
| 047816h | CAN1 Mailbox 1: Data Field | | XXXX XXXX |
| 047817h | | | XXXX XXXXh |
| 047818h | | | |
| 047819h | | | |
| 04781Ah | | | |
| 04781Bh | | | |
| 04781Ch | | | |
| 04781Dh | | | |
| 04781Eh | CAN1 Mailbox 1: Time Stamp | | XXXXh |
| 04781Fh | | | |
| 047820h | CAN1 Mailbox 2: Message Identifier | C1MB2 | XXXX XXXXh |
| 047821h | | | |
| 047822h | | | |
| 047823h | | | |
| 047824h | | | |
| 047825h | CAN1 Mailbox 2: Data Length | | XXh |
| 047826h | CAN1 Mailbox 2: Data Field | | XXXX XXXX |
| 047827h | | | XXXX XXXXh |
| 047828h | | | |
| 047829h | | | |
| 04782Ah | | | |
| 04782Bh | | | |
| 04782Ch | | | |
| 04782Dh | | | |
| 04782Eh | CAN1 Mailbox 2: Time Stamp | | XXXXh |
| 04782Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.27 SFR List (27)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047830h | CAN1 Mailbox 3: Message Identifier | C1MB3 | XXXX XXXXh |
| 047831h | | | |
| 047832h | | | |
| 047833h | | | |
| 047834h | | | |
| 047835h | CAN1 Mailbox 3: Data Length | | XXh |
| 047836h | CAN1 Mailbox 3: Data Field | | XXXX XXXX |
| 047837h | | | XXXX XXXXh |
| 047838h | | | |
| 047839h | | | |
| 04783Ah | | | |
| 04783Bh | | | |
| 04783Ch | | | |
| 04783Dh | | | |
| 04783Eh | CAN1 Mailbox 3: Time Stamp | | XXXXh |
| 04783Fh | | | |
| 047840h | CAN1 Mailbox 4: Message Identifier | C1MB4 | XXXX XXXXh |
| 047841h | | | |
| 047842h | | | |
| 047843h | | | |
| 047844h | | | |
| 047845h | CAN1 Mailbox 4: Data Length | | XXh |
| 047846h | CAN1 Mailbox 4: Data Field | | XXXX XXXX |
| 047847h | | | XXXX XXXXh |
| 047848h | | | |
| 047849h | | | |
| 04784Ah | | | |
| 04784Bh | | | |
| 04784Ch | | | |
| 04784Dh | | | |
| 04784Eh | CAN1 Mailbox 4: Time Stamp | | XXXXh |
| 04784Fh | | | |
| 047850h | CAN1 Mailbox 5: Message Identifier | C1MB5 | XXXX XXXXh |
| 047851h | | | |
| 047852h | | | |
| 047853h | | | |
| 047854h | | | |
| 047855h | CAN1 Mailbox 5: Data Length | | XXh |
| 047856h | CAN1 Mailbox 5: Data Field | | XXXX XXXX |
| 047857h | | | XXXX XXXXh |
| 047858h | | | |
| 047859h | | | |
| 04785Ah | | | |
| 04785Bh | | | |
| 04785Ch | | | |
| 04785Dh | | | |
| 04785Eh | CAN1 Mailbox 5: Time Stamp | | XXXXh |
| 04785Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.28 SFR List (28)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047860h | CAN1 Mailbox 6: Message Identifier | C1MB6 | XXXX XXXXh |
| 047861h | | | |
| 047862h | | | |
| 047863h | | | |
| 047864h | | | |
| 047865h | CAN1 Mailbox 6: Data Length | | XXh |
| 047866h | CAN1 Mailbox 6: Data Field | | XXXX XXXX |
| 047867h | | | XXXX XXXXh |
| 047868h | | | |
| 047869h | | | |
| 04786Ah | | | |
| 04786Bh | | | |
| 04786Ch | | | |
| 04786Dh | | | |
| 04786Eh | CAN1 Mailbox 6: Time Stamp | | XXXXh |
| 04786Fh | | | |
| 047870h | CAN1 Mailbox 7: Message Identifier | C1MB7 | XXXX XXXXh |
| 047871h | | | |
| 047872h | | | |
| 047873h | | | |
| 047874h | | | |
| 047875h | CAN1 Mailbox 7: Data Length | | XXh |
| 047876h | CAN1 Mailbox 7: Data Field | | XXXX XXXX |
| 047877h | | | XXXX XXXXh |
| 047878h | | | |
| 047879h | | | |
| 04787Ah | | | |
| 04787Bh | | | |
| 04787Ch | | | |
| 04787Dh | | | |
| 04787Eh | CAN1 Mailbox 7: Time Stamp | | XXXXh |
| 04787Fh | | | |
| 047880h | CAN1 Mailbox 8: Message Identifier | C1MB8 | XXXX XXXXh |
| 047881h | | | |
| 047882h | | | |
| 047883h | | | |
| 047884h | | | |
| 047885h | CAN1 Mailbox 8: Data Length | | XXh |
| 047886h | CAN1 Mailbox 8: Data Field | | XXXX XXXX |
| 047887h | | | XXXX XXXXh |
| 047888h | | | |
| 047889h | | | |
| 04788Ah | | | |
| 04788Bh | | | |
| 04788Ch | | | |
| 04788Dh | | | |
| 04788Eh | CAN1 Mailbox 8: Time Stamp | | XXXXh |
| 04788Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.29 SFR List (29)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047890h | CAN1 Mailbox 9: Message Identifier | C1MB9 | XXXX XXXXh |
| 047891h | | | |
| 047892h | | | |
| 047893h | | | |
| 047894h | | | |
| 047895h | CAN1 Mailbox 9: Data Length | | XXh |
| 047896h | CAN1 Mailbox 9: Data Field | | XXXX XXXX |
| 047897h | | | XXXX XXXXh |
| 047898h | | | |
| 047899h | | | |
| 04789Ah | | | |
| 04789Bh | | | |
| 04789Ch | | | |
| 04789Dh | | | |
| 04789Eh | CAN1 Mailbox 9: Time Stamp | | XXXXh |
| 04789Fh | | | |
| 0478A0h | CAN1 Mailbox 10: Message Identifier | C1MB10 | XXXX XXXXh |
| 0478A1h | | | |
| 0478A2h | | | |
| 0478A3h | | | |
| 0478A4h | | | |
| 0478A5h | CAN1 Mailbox 10: Data Length | | XXh |
| 0478A6h | CAN1 Mailbox 10: Data Field | | XXXX XXXX |
| 0478A7h | | | XXXX XXXXh |
| 0478A8h | | | |
| 0478A9h | | | |
| 0478AAh | | | |
| 0478ABh | | | |
| 0478ACh | | | |
| 0478ADh | | | |
| 0478AEh | CAN1 Mailbox 10: Time Stamp | | XXXXh |
| 0478AFh | | | |
| 0478B0h | CAN1 Mailbox 11: Message Identifier | C1MB11 | XXXX XXXXh |
| 0478B1h | | | |
| 0478B2h | | | |
| 0478B3h | | | |
| 0478B4h | | | |
| 0478B5h | CAN1 Mailbox 11: Data Length | | XXh |
| 0478B6h | CAN1 Mailbox 11: Data Field | | XXXX XXXX |
| 0478B7h | | | XXXX XXXXh |
| 0478B8h | | | |
| 0478B9h | | | |
| 0478BAh | | | |
| 0478BBh | | | |
| 0478BCh | | | |
| 0478BDh | | | |
| 0478BEh | CAN1 Mailbox 11: Time Stamp | | XXXXh |
| 0478BFh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.30 SFR List (30)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 0478C0h | CAN1 Mailbox 12: Message Identifier | C1MB12 | XXXX XXXXh |
| 0478C1h | | | |
| 0478C2h | | | |
| 0478C3h | | | |
| 0478C4h | | | |
| 0478C5h | CAN1 Mailbox 12: Data Length | | XXh |
| 0478C6h | CAN1 Mailbox 12: Data Field | | XXXX XXXX |
| 0478C7h | | | XXXX XXXXh |
| 0478C8h | | | |
| 0478C9h | | | |
| 0478CAh | | | |
| 0478CBh | | | |
| 0478CCh | | | |
| 0478CDh | | | |
| 0478CEh | CAN1 Mailbox 12: Time Stamp | | XXXXh |
| 0478CFh | | | |
| 0478D0h | CAN1 Mailbox 13: Message Identifier | C1MB13 | XXXX XXXXh |
| 0478D1h | | | |
| 0478D2h | | | |
| 0478D3h | | | |
| 0478D4h | | | |
| 0478D5h | CAN1 Mailbox 13: Data Length | | XXh |
| 0478D6h | CAN1 Mailbox 13: Data Field | | XXXX XXXX |
| 0478D7h | | | XXXX XXXXh |
| 0478D8h | | | |
| 0478D9h | | | |
| 0478DAh | | | |
| 0478DBh | | | |
| 0478DCh | | | |
| 0478DDh | | | |
| 0478DEh | CAN1 Mailbox 13: Time Stamp | | XXXXh |
| 0478DFh | | | |
| 0478E0h | CAN1 Mailbox 14: Message Identifier | C1MB14 | XXXX XXXXh |
| 0478E1h | | | |
| 0478E2h | | | |
| 0478E3h | | | |
| 0478E4h | | | |
| 0478E5h | CAN1 Mailbox 14: Data Length | | XXh |
| 0478E6h | CAN1 Mailbox 14: Data Field | | XXXX XXXX |
| 0478E7h | | | XXXX XXXXh |
| 0478E8h | | | |
| 0478E9h | | | |
| 0478EAh | | | |
| 0478EBh | | | |
| 0478ECh | | | |
| 0478EDh | | | |
| 0478EEh | CAN1 Mailbox 14: Time Stamp | | XXXXh |
| 0478EFh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.31 SFR List (31)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 0478F0h | CAN1 Mailbox 15: Message Identifier | C1MB15 | XXXX XXXXh |
| 0478F1h | | | |
| 0478F2h | | | |
| 0478F3h | | | |
| 0478F4h | | | |
| 0478F5h | CAN1 Mailbox 15: Data Length | | XXh |
| 0478F6h | CAN1 Mailbox 15: Data Field | | XXXX XXXX |
| 0478F7h | | | XXXX XXXXh |
| 0478F8h | | | |
| 0478F9h | | | |
| 0478FAh | | | |
| 0478FBh | | | |
| 0478FCh | | | |
| 0478FDh | | | |
| 0478FEh | CAN1 Mailbox 15: Time Stamp | | XXXXh |
| 0478FFh | | | |
| 047900h | CAN1 Mailbox 16: Message Identifier | C1MB16 | XXXX XXXXh |
| 047901h | | | |
| 047902h | | | |
| 047903h | | | |
| 047904h | | | |
| 047905h | CAN1 Mailbox 16: Data Length | | XXh |
| 047906h | CAN1 Mailbox 16: Data Field | | XXXX XXXX |
| 047907h | | | XXXX XXXXh |
| 047908h | | | |
| 047909h | | | |
| 04790Ah | | | |
| 04790Bh | | | |
| 04790Ch | | | |
| 04790Dh | | | |
| 04790Eh | CAN1 Mailbox 16: Time Stamp | | XXXXh |
| 04790Fh | | | |
| 047910h | CAN1 Mailbox 17: Message Identifier | C1MB17 | XXXX XXXXh |
| 047911h | | | |
| 047912h | | | |
| 047913h | | | |
| 047914h | | | |
| 047915h | CAN1 Mailbox 17: Data Length | | XXh |
| 047916h | CAN1 Mailbox 17: Data Field | | XXXX XXXX |
| 047917h | | | XXXX XXXXh |
| 047918h | | | |
| 047919h | | | |
| 04791Ah | | | |
| 04791Bh | | | |
| 04791Ch | | | |
| 04791Dh | | | |
| 04791Eh | CAN1 Mailbox 17: Time Stamp | | XXXXh |
| 04791Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.32 SFR List (32)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047920h | CAN1 Mailbox 18: Message Identifier | C1MB18 | XXXX XXXXh |
| 047921h | | | |
| 047922h | | | |
| 047923h | | | |
| 047924h | | | |
| 047925h | CAN1 Mailbox 18: Data Length | | XXh |
| 047926h | CAN1 Mailbox 18: Data Field | | XXXX XXXX |
| 047927h | | | XXXX XXXXh |
| 047928h | | | |
| 047929h | | | |
| 04792Ah | | | |
| 04792Bh | | | |
| 04792Ch | | | |
| 04792Dh | | | |
| 04792Eh | CAN1 Mailbox18: Time Stamp | | XXXXh |
| 04792Fh | | | |
| 047930h | CAN1 Mailbox 19: Message Identifier | C1MB19 | XXXX XXXXh |
| 047931h | | | |
| 047932h | | | |
| 047933h | | | |
| 047934h | | | |
| 047935h | CAN1 Mailbox 19: Data Length | | XXh |
| 047936h | CAN1 Mailbox 19: Data Field | | XXXX XXXX |
| 047937h | | | XXXX XXXXh |
| 047938h | | | |
| 047939h | | | |
| 04793Ah | | | |
| 04793Bh | | | |
| 04793Ch | | | |
| 04793Dh | | | |
| 04793Eh | CAN1 Mailbox 19: Time Stamp | | XXXXh |
| 04793Fh | | | |
| 047940h | CAN1 Mailbox 20: Message Identifier | C1MB20 | XXXX XXXXh |
| 047941h | | | |
| 047942h | | | |
| 047943h | | | |
| 047944h | | | |
| 047945h | CAN1 Mailbox 20: Data Length | | XXh |
| 047946h | CAN1 Mailbox 20: Data Field | | XXXX XXXX |
| 047947h | | | XXXX XXXXh |
| 047948h | | | |
| 047949h | | | |
| 04794Ah | | | |
| 04794Bh | | | |
| 04794Ch | | | |
| 04794Dh | | | |
| 04794Eh | CAN1 Mailbox 20: Time Stamp | | XXXXh |
| 04794Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.33 SFR List (33)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047950h | CAN1 Mailbox 21: Message Identifier | C1MB21 | XXXX XXXXh |
| 047951h | | | |
| 047952h | | | |
| 047953h | | | |
| 047954h | | | |
| 047955h | CAN1 Mailbox 21: Data Length | | XXh |
| 047956h | CAN1 Mailbox 21: Data Field | | XXXX XXXX |
| 047957h | | | XXXX XXXXh |
| 047958h | | | |
| 047959h | | | |
| 04795Ah | | | |
| 04795Bh | | | |
| 04795Ch | | | |
| 04795Dh | | | |
| 04795Eh | CAN1 Mailbox 21: Time Stamp | | XXXXh |
| 04795Fh | | | |
| 047960h | CAN1 Mailbox 22: Identifier | C1MB22 | XXXX XXXXh |
| 047961h | | | |
| 047962h | | | |
| 047963h | | | |
| 047964h | | | |
| 047965h | CAN1 Mailbox 22: Data Length | | XXh |
| 047966h | CAN1 Mailbox 22: Data Field | | XXXX XXXX |
| 047967h | | | XXXX XXXXh |
| 047968h | | | |
| 047969h | | | |
| 04796Ah | | | |
| 04796Bh | | | |
| 04796Ch | | | |
| 04796Dh | | | |
| 04796Eh | CAN1 Mailbox 22: Time Stamp | | XXXXh |
| 04796Fh | | | |
| 047970h | CAN1 Mailbox 23: Message Identifier | C1MB23 | XXXX XXXXh |
| 047971h | | | |
| 047972h | | | |
| 047973h | | | |
| 047974h | | | |
| 047975h | CAN1 Mailbox 23: Data Length | | XXh |
| 047976h | CAN1 Mailbox 23: Data Field | | XXXX XXXX |
| 047977h | | | XXXX XXXXh |
| 047978h | | | |
| 047979h | | | |
| 04797Ah | | | |
| 04797Bh | | | |
| 04797Ch | | | |
| 04797Dh | | | |
| 04797Eh | CAN1 Mailbox 23: Time Stamp | | XXXXh |
| 04797Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.34 SFR List (34)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047980h | CAN1 Mailbox 24: Message Identifier | C1MB24 | XXXX XXXXh |
| 047981h | | | |
| 047982h | | | |
| 047983h | | | |
| 047984h | | | |
| 047985h | CAN1 Mailbox 24: Data Length | | XXh |
| 047986h | CAN1 Mailbox 24: Data Field | | XXXX XXXX |
| 047987h | | | XXXX XXXXh |
| 047988h | | | |
| 047989h | | | |
| 04798Ah | | | |
| 04798Bh | | | |
| 04798Ch | | | |
| 04798Dh | | | |
| 04798Eh | CAN1 Mailbox 24: Time Stamp | | XXXXh |
| 04798Fh | | | |
| 047990h | CAN1 Mailbox 25: Message Identifier | C1MB25 | XXXX XXXXh |
| 047991h | | | |
| 047992h | | | |
| 047993h | | | |
| 047994h | | | |
| 047995h | CAN1 Mailbox 25: Data Length | | XXh |
| 047996h | CAN1 Mailbox 25: Data Field | | XXXX XXXX |
| 047997h | | | XXXX XXXXh |
| 047998h | | | |
| 047999h | | | |
| 04799Ah | | | |
| 04799Bh | | | |
| 04799Ch | | | |
| 04799Dh | | | |
| 04799Eh | CAN1 Mailbox 25: Time Stamp | | XXXXh |
| 04799Fh | | | |
| 0479A0h | CAN1 Mailbox 26: Message Identifier | C1MB26 | XXXX XXXXh |
| 0479A1h | | | |
| 0479A2h | | | |
| 0479A3h | | | |
| 0479A4h | | | |
| 0479A5h | CAN1 Mailbox 26: Data Length | | XXh |
| 0479A6h | CAN1 Mailbox 26: Data Field | | XXXX XXXX |
| 0479A7h | | | XXXX XXXXh |
| 0479A8h | | | |
| 0479A9h | | | |
| 0479AAh | | | |
| 0479ABh | | | |
| 0479ACh | | | |
| 0479ADh | | | |
| 0479AEh | CAN1 Mailbox 26: Time Stamp | | XXXXh |
| 0479AFh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.35 SFR List (35)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 0479B0h | CAN1 Mailbox 27: Message Identifier | C1MB27 | XXXX XXXXh |
| 0479B1h | | | |
| 0479B2h | | | |
| 0479B3h | | | |
| 0479B4h | | | |
| 0479B5h | CAN1 Mailbox 27: Data Length | | XXh |
| 0479B6h | CAN1 Mailbox 27: Data Field | | XXXX XXXX |
| 0479B7h | | | XXXX XXXXh |
| 0479B8h | | | |
| 0479B9h | | | |
| 0479BAh | | | |
| 0479BBh | | | |
| 0479BCh | | | |
| 0479BDh | | | |
| 0479BEh | CAN1 Mailbox 27: Time Stamp | | XXXXh |
| 0479BFh | | | |
| 0479C0h | CAN1 Mailbox 28: Message Identifier | C1MB28 | XXXX XXXXh |
| 0479C1h | | | |
| 0479C2h | | | |
| 0479C3h | | | |
| 0479C4h | | | |
| 0479C5h | CAN1 Mailbox 28: Data Length | | XXh |
| 0479C6h | CAN1 Mailbox 28: Data Field | | XXXX XXXX |
| 0479C7h | | | XXXX XXXXh |
| 0479C8h | | | |
| 0479C9h | | | |
| 0479CAh | | | |
| 0479CBh | | | |
| 0479CCh | | | |
| 0479CDh | | | |
| 0479CEh | CAN1 Mailbox 28: Time Stamp | | XXXXh |
| 0479CFh | | | |
| 0479D0h | CAN1 Mailbox 29: Message Identifier | C1MB29 | XXXX XXXXh |
| 0479D1h | | | |
| 0479D2h | | | |
| 0479D3h | | | |
| 0479D4h | | | |
| 0479D5h | CAN1 Mailbox 29: Data Length | | XXh |
| 0479D6h | CAN1 Mailbox 29: Data Field | | XXXX XXXX |
| 0479D7h | | | XXXX XXXXh |
| 0479D8h | | | |
| 0479D9h | | | |
| 0479DAh | | | |
| 0479DBh | | | |
| 0479DCh | | | |
| 0479DDh | | | |
| 0479DEh | CAN1 Mailbox 29: Time Stamp | | XXXXh |
| 0479DFh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.36 SFR List (36)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 0479E0h | CAN1 Mailbox 30: Message Identifier | C1MB30 | XXXX XXXXh |
| 0479E1h | | | |
| 0479E2h | | | |
| 0479E3h | | | |
| 0479E4h | | | |
| 0479E5h | CAN1 Mailbox 30: Data Length | | XXh |
| 0479E6h | CAN1 Mailbox 30: Data Field | | XXXX XXXX |
| 0479E7h | | | XXXX XXXXh |
| 0479E8h | | | |
| 0479E9h | | | |
| 0479EAh | | | |
| 0479EBh | | | |
| 0479ECh | | | |
| 0479EDh | | | |
| 0479EEh | CAN1 Mailbox 30: Time Stamp | | XXXXh |
| 0479EFh | | | |
| 0479F0h | CAN1 Mailbox 31: Message Identifier | C1MB31 | XXXX XXXXh |
| 0479F1h | | | |
| 0479F2h | | | |
| 0479F3h | | | |
| 0479F4h | | | |
| 0479F5h | CAN1 Mailbox 31: Data Length | | XXh |
| 0479F6h | CAN1 Mailbox 31: Data Field | | XXXX XXXX |
| 0479F7h | | | XXXX XXXXh |
| 0479F8h | | | |
| 0479F9h | | | |
| 0479FAh | | | |
| 0479FBh | | | |
| 0479FCh | | | |
| 0479FDh | | | |
| 0479FEh | CAN1 Mailbox 31: Time Stamp | | XXXXh |
| 0479FFh | | | |
| 047A00h | CAN1 Acceptance Mask Register 0 | C1MKR0 | XXXX XXXXh |
| 047A01h | | | |
| 047A02h | | | |
| 047A03h | | | |
| 047A04h | CAN1 Acceptance Mask Register 1 | C1MKR1 | XXXX XXXXh |
| 047A05h | | | |
| 047A06h | | | |
| 047A07h | | | |
| 047A08h | CAN1 Acceptance Mask Register 2 | C1MKR2 | XXXX XXXXh |
| 047A09h | | | |
| 047A0Ah | | | |
| 047A0Bh | | | |
| 047A0Ch | CAN1 Acceptance Mask Register 3 | C1MKR3 | XXXX XXXXh |
| 047A0Dh | | | |
| 047A0Eh | | | |
| 047A0Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.37 SFR List (37)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047A10h | CAN1 Acceptance Mask Register 4 | C1MKR4 | XXXX XXXXh |
| 047A11h | | | |
| 047A12h | | | |
| 047A13h | | | |
| 047A14h | CAN1 Acceptance Mask Register 5 | C1MKR5 | XXXX XXXXh |
| 047A15h | | | |
| 047A16h | | | |
| 047A17h | | | |
| 047A18h | CAN1 Acceptance Mask Register 6 | C1MKR6 | XXXX XXXXh |
| 047A19h | | | |
| 047A1Ah | | | |
| 047A1Bh | | | |
| 047A1Ch | CAN1 Acceptance Mask Register 7 | C1MKR7 | XXXX XXXXh |
| 047A1Dh | | | |
| 047A1Eh | | | |
| 047A1Fh | | | |
| 047A20h | CAN1 FIFO Received ID Compare Register 0 | C1FIDCR0 | XXXX XXXXh |
| 047A21h | | | |
| 047A22h | | | |
| 047A23h | | | |
| 047A24h | CAN1 FIFO Received ID Compare Register 1 | C1FIDCR1 | XXXX XXXXh |
| 047A25h | | | |
| 047A26h | | | |
| 047A27h | | | |
| 047A28h | CAN1 Mask Invalid Register | C1MKIVLR | XXXX XXXXh |
| 047A29h | | | |
| 047A2Ah | | | |
| 047A2Bh | | | |
| 047A2Ch | CAN1 Mailbox Interrupt Enable Register | C1MIER | XXXX XXXXh |
| 047A2Dh | | | |
| 047A2Eh | | | |
| 047A2Fh | | | |
| 047A30h | | | |
| 047A31h | | | |
| 047A32h | | | |
| 047A33h | | | |
| 047A34h | | | |
| 047A35h | | | |
| 047A36h | | | |
| 047A37h | | | |
| 047A38h | | | |
| 047A39h | | | |
| 047A3Ah | | | |
| 047A3Bh | | | |
| 047A3Ch | | | |
| 047A3Dh | | | |
| 047A3Eh | | | |
| 047A3Fh | | | |
| 047A40h to 047B1Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.38 SFR List (38)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047B20h | CAN1 Message Control Register 0 | C1MCTL0 | 00h |
| 047B21h | CAN1 Message Control Register 1 | C1MCTL1 | 00h |
| 047B22h | CAN1 Message Control Register 2 | C1MCTL2 | 00h |
| 047B23h | CAN1 Message Control Register 3 | C1MCTL3 | 00h |
| 047B24h | CAN1 Message Control Register 4 | C1MCTL4 | 00h |
| 047B25h | CAN1 Message Control Register 5 | C1MCTL5 | 00h |
| 047B26h | CAN1 Message Control Register 6 | C1MCTL6 | 00h |
| 047B27h | CAN1 Message Control Register 7 | C1MCTL7 | 00h |
| 047B28h | CAN1 Message Control Register 8 | C1MCTL8 | 00h |
| 047B29h | CAN1 Message Control Register 9 | C1MCTL9 | 00h |
| 047B2Ah | CAN1 Message Control Register 10 | C1MCTL10 | 00h |
| 047B2Bh | CAN1 Message Control Register 11 | C1MCTL11 | 00h |
| 047B2Ch | CAN1 Message Control Register 12 | C1MCTL12 | 00h |
| 047B2Dh | CAN1 Message Control Register 13 | C1MCTL13 | 00h |
| 047B2Eh | CAN1 Message Control Register 14 | C1MCTL14 | 00h |
| 047B2Fh | CAN1 Message Control Register 15 | C1MCTL15 | 00h |
| 047B30h | CAN1 Message Control Register 16 | C1MCTL16 | 00h |
| 047B31h | CAN1 Message Control Register 17 | C1MCTL17 | 00h |
| 047B32h | CAN1 Message Control Register 18 | C1MCTL18 | 00h |
| 047B33h | CAN1 Message Control Register 19 | C1MCTL19 | 00h |
| 047B34h | CAN1 Message Control Register 20 | C1MCTL20 | 00h |
| 047B35h | CAN1 Message Control Register 21 | C1MCTL21 | 00h |
| 047B36h | CAN1 Message Control Register 22 | C1MCTL22 | 00h |
| 047B37h | CAN1 Message Control Register 23 | C1MCTL23 | 00h |
| 047B38h | CAN1 Message Control Register 24 | C1MCTL24 | 00h |
| 047B39h | CAN1 Message Control Register 25 | C1MCTL25 | 00h |
| 047B3Ah | CAN1 Message Control Register 26 | C1MCTL26 | 00h |
| 047B3Bh | CAN1 Message Control Register 27 | C1MCTL27 | 00h |
| 047B3Ch | CAN1 Message Control Register 28 | C1MCTL28 | 00h |
| 047B3Dh | CAN1 Message Control Register 29 | C1MCTL29 | 00h |
| 047B3Eh | CAN1 Message Control Register 30 | C1MCTL30 | 00h |
| 047B3Fh | CAN1 Message Control Register 31 | C1MCTL31 | 00h |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.39 SFR List (39)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047B40h | CAN1 Control Register | C1CTLR | 0000 0101b |
| 047B41h | | | 0000 0000b |
| 047B42h | CAN1 Status Register | C1STR | 0000 0101b |
| 047B43h | | | 0000 0000b |
| 047B44h | CAN1 Bit Configuration Register | C1BCR | 00 0000h |
| 047B45h | | | |
| 047B46h | | | |
| 047B47h | CAN1 Clock Select Register | C1CLKR | 000X 0000b |
| 047B48h | CAN1 Receive FIFO Control Register | C1RFCR | 1000 0000b |
| 047B49h | CAN1 Receive FIFO Pointer Control Register | C1RFPCR | XXh |
| 047B4Ah | CAN1 Transmit FIFO Control Register | C1TFCR | 1000 0000b |
| 047B4Bh | CAN1 Transmit FIFO Pointer Control Register | C1TFPCR | XXh |
| 047B4Ch | CAN1 Error Interrupt Enable Register | C1EIER | 00h |
| 047B4Dh | CAN1 Error Interrupt Factor Judge Register | C1EIFR | 00h |
| 047B4Eh | CAN1 Reception Error Count Register | C1RECR | 00h |
| 047B4Fh | CAN1 Transmission Error Count Register | C1TECR | 00h |
| 047B50h | CAN1 Error Code Store Register | C1ECSR | 00h |
| 047B51h | CAN1 Channel Search Support Register | C1CSSR | XXh |
| 047B52h | CAN1 Mailbox Search Status Register | C1MSSR | 1000 0000b |
| 047B53h | CAN1 Mailbox Search Mode Register | C1MSMR | XXXX XX00b |
| 047B54h | CAN1 Time Stamp Register | C1TSR | 0000h |
| 047B55h | | | |
| 047B56h | CAN1 Acceptance Filter Support Register | C1AFSR | XXXXh |
| 047B57h | | | |
| 047B58h | CAN1 Test Control Register | C1TCR | 00h |
| 047B59h | | | |
| 047B5Ah | | | |
| 047B5Bh | | | |
| 047B5Ch | | | |
| 047B5Dh | | | |
| 047B5Eh | | | |
| 047B5Fh | | | |
| 047B60h to 047BFFh | | | |

X: Undefined

Blanks are reserved. No access is allowed.

**Table 4.40 SFR List (40)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047C00h | CAN0 Mailbox 0: Message Identifier | C0MB0 | XXXX XXXXh |
| 047C01h | | | |
| 047C02h | | | |
| 047C03h | | | |
| 047C04h | | | |
| 047C05h | CAN0 Mailbox 0: Data Length | | XXh |
| 047C06h | CAN0 Mailbox 0: Data Field | | XXXX XXXX |
| 047C07h | | | XXXX XXXXh |
| 047C08h | | | |
| 047C09h | | | |
| 047C0Ah | | | |
| 047C0Bh | | | |
| 047C0Ch | | | |
| 047C0Dh | | | |
| 047C0Eh | CAN0 Mailbox 0: Time Stamp | | XXXXh |
| 047C0Fh | | | |
| 047C10h | CAN0 Mailbox 1: Message Identifier | C0MB1 | XXXX XXXXh |
| 047C11h | | | |
| 047C12h | | | |
| 047C13h | | | |
| 047C14h | | | |
| 047C15h | CAN0 Mailbox 1: Data Length | | XXh |
| 047C16h | CAN0 Mailbox 1: Data Field | | XXXX XXXX |
| 047C17h | | | XXXX XXXXh |
| 047C18h | | | |
| 047C19h | | | |
| 047C1Ah | | | |
| 047C1Bh | | | |
| 047C1Ch | | | |
| 047C1Dh | | | |
| 047C1Eh | CAN0 Mailbox 1: Time Stamp | | XXXXh |
| 047C1Fh | | | |
| 047C20h | CAN0 Mailbox 2: Message Identifier | C0MB2 | XXXX XXXXh |
| 047C21h | | | |
| 047C22h | | | |
| 047C23h | | | |
| 047C24h | | | |
| 047C25h | CAN0 Mailbox 2: Data Length | | XXh |
| 047C26h | CAN0 Mailbox 2: Data Field | | XXXX XXXX |
| 047C27h | | | XXXX XXXXh |
| 047C28h | | | |
| 047C29h | | | |
| 047C2Ah | | | |
| 047C2Bh | | | |
| 047C2Ch | | | |
| 047C2Dh | | | |
| 047C2Eh | CAN0 Mailbox 2: Time Stamp | | XXXXh |
| 047C2Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.41 SFR List (41)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047C30h | CAN0 Mailbox 3: Message Identifier | C0MB3 | XXXX XXXXh |
| 047C31h | | | |
| 047C32h | | | |
| 047C33h | | | |
| 047C34h | | | |
| 047C35h | CAN0 Mailbox 3: Data Length | | XXh |
| 047C36h | CAN0 Mailbox 3: Data Field | | XXXX XXXX |
| 047C37h | | | XXXX XXXXh |
| 047C38h | | | |
| 047C39h | | | |
| 047C3Ah | | | |
| 047C3Bh | | | |
| 047C3Ch | | | |
| 047C3Dh | | | |
| 047C3Eh | CAN0 Mailbox 3: Time Stamp | | XXXXh |
| 047C3Fh | | | |
| 047C40h | CAN0 Mailbox 4: Message Identifier | C0MB4 | XXXX XXXXh |
| 047C41h | | | |
| 047C42h | | | |
| 047C43h | | | |
| 047C44h | | | |
| 047C45h | CAN0 Mailbox 4: Data Length | | XXh |
| 047C46h | CAN0 Mailbox 4: Data Field | | XXXX XXXX |
| 047C47h | | | XXXX XXXXh |
| 047C48h | | | |
| 047C49h | | | |
| 047C4Ah | | | |
| 047C4Bh | | | |
| 047C4Ch | | | |
| 047C4Dh | | | |
| 047C4Eh | CAN0 Mailbox 4: Time Stamp | | XXXXh |
| 047C4Fh | | | |
| 047C50h | CAN0 Mailbox 5: Message Identifier | C0MB5 | XXXX XXXXh |
| 047C51h | | | |
| 047C52h | | | |
| 047C53h | | | |
| 047C54h | | | |
| 047C55h | CAN0 Mailbox 5: Data Length | | XXh |
| 047C56h | CAN0 Mailbox 5: Data Field | | XXXX XXXX |
| 047C57h | | | XXXX XXXXh |
| 047C58h | | | |
| 047C59h | | | |
| 047C5Ah | | | |
| 047C5Bh | | | |
| 047C5Ch | | | |
| 047C5Dh | | | |
| 047C5Eh | CAN0 Mailbox 5: Time Stamp | | XXXXh |
| 047C5Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.42 SFR List (42)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047C60h | CAN0 Mailbox 6: Message Identifier | C0MB6 | XXXX XXXXh |
| 047C61h | | | |
| 047C62h | | | |
| 047C63h | | | |
| 047C64h | | | |
| 047C65h | CAN0 Mailbox 6: Data Length | | XXh |
| 047C66h | CAN0 Mailbox 6: Data Field | | XXXX XXXX |
| 047C67h | | | XXXX XXXXh |
| 047C68h | | | |
| 047C69h | | | |
| 047C6Ah | | | |
| 047C6Bh | | | |
| 047C6Ch | | | |
| 047C6Dh | | | |
| 047C6Eh | CAN0 Mailbox 6: Time Stamp | | XXXXh |
| 047C6Fh | | | |
| 047C70h | CAN0 Mailbox 7: Message Identifier | C0MB7 | XXXX XXXXh |
| 047C71h | | | |
| 047C72h | | | |
| 047C73h | | | |
| 047C74h | | | |
| 047C75h | CAN0 Mailbox 7: Data Length | | XXh |
| 047C76h | CAN0 Mailbox 7: Data Field | | XXXX XXXX |
| 047C77h | | | XXXX XXXXh |
| 047C78h | | | |
| 047C79h | | | |
| 047C7Ah | | | |
| 047C7Bh | | | |
| 047C7Ch | | | |
| 047C7Dh | | | |
| 047C7Eh | CAN0 Mailbox 7: Time Stamp | | XXXXh |
| 047C7Fh | | | |
| 047C80h | CAN0 Mailbox 8: Message Identifier | C0MB8 | XXXX XXXXh |
| 047C81h | | | |
| 047C82h | | | |
| 047C83h | | | |
| 047C84h | | | |
| 047C85h | CAN0 Mailbox 8: Data Length | | XXh |
| 047C86h | CAN0 Mailbox 8: Data Field | | XXXX XXXX |
| 047C87h | | | XXXX XXXXh |
| 047C88h | | | |
| 047C89h | | | |
| 047C8Ah | | | |
| 047C8Bh | | | |
| 047C8Ch | | | |
| 047C8Dh | | | |
| 047C8Eh | CAN0 Mailbox 8: Time Stamp | | XXXXh |
| 047C8Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.43 SFR List (43)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047C90h | CAN0 Mailbox 9: Message Identifier | C0MB9 | XXXX XXXXh |
| 047C91h | | | |
| 047C92h | | | |
| 047C93h | | | |
| 047C94h | | | |
| 047C95h | CAN0 Mailbox 9: Data Length | | XXh |
| 047C96h | CAN0 Mailbox 9: Data Field | | XXXX XXXX |
| 047C97h | | | XXXX XXXXh |
| 047C98h | | | |
| 047C99h | | | |
| 047C9Ah | | | |
| 047C9Bh | | | |
| 047C9Ch | | | |
| 047C9Dh | | | |
| 047C9Eh | CAN0 Mailbox 9: Time Stamp | | XXXXh |
| 047C9Fh | | | |
| 047CA0h | CAN0 Mailbox 10: Message Identifier | C0MB10 | XXXX XXXXh |
| 047CA1h | | | |
| 047CA2h | | | |
| 047CA3h | | | |
| 047CA4h | | | |
| 047CA5h | CAN0 Mailbox 10: Data Length | | XXh |
| 047CA6h | CAN0 Mailbox 10: Data Field | | XXXX XXXX |
| 047CA7h | | | XXXX XXXXh |
| 047CA8h | | | |
| 047CA9h | | | |
| 047CAAh | | | |
| 047CABh | | | |
| 047CACh | | | |
| 047CADh | | | |
| 047CAEh | CAN0 Mailbox 10: Time Stamp | | XXXXh |
| 047CAFh | | | |
| 047CB0h | CAN0 Mailbox 11: Message Identifier | C0MB11 | XXXX XXXXh |
| 047CB1h | | | |
| 047CB2h | | | |
| 047CB3h | | | |
| 047CB4h | | | |
| 047CB5h | CAN0 Mailbox 11: Data Length | | XXh |
| 047CB6h | CAN0 Mailbox 11: Data Field | | XXXX XXXX |
| 047CB7h | | | XXXX XXXXh |
| 047CB8h | | | |
| 047CB9h | | | |
| 047CBAh | | | |
| 047CBBh | | | |
| 047CBCh | | | |
| 047CBDh | | | |
| 047CBEh | CAN0 Mailbox 11: Time Stamp | | XXXXh |
| 047CBFh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.44 SFR List (44)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047CC0h | CAN0 Mailbox 12: Message Identifier | C0MB12 | XXXX XXXXh |
| 047CC1h | | | |
| 047CC2h | | | |
| 047CC3h | | | |
| 047CC4h | | | |
| 047CC5h | CAN0 Mailbox 12: Data Length | | XXh |
| 047CC6h | CAN0 Mailbox 12: Data Field | | XXXX XXXX |
| 047CC7h | | | XXXX XXXXh |
| 047CC8h | | | |
| 047CC9h | | | |
| 047CCAh | | | |
| 047CCBh | | | |
| 047CCCh | | | |
| 047CCDh | | | |
| 047CCEh | CAN0 Mailbox 12: Time Stamp | | XXXXh |
| 047CCFh | | | |
| 047CD0h | CAN0 Mailbox 13: Message Identifier | C0MB13 | XXXX XXXXh |
| 047CD1h | | | |
| 047CD2h | | | |
| 047CD3h | | | |
| 047CD4h | | | |
| 047CD5h | CAN0 Mailbox 13: Data Length | | XXh |
| 047CD6h | CAN0 Mailbox 13: Data Field | | XXXX XXXX |
| 047CD7h | | | XXXX XXXXh |
| 047CD8h | | | |
| 047CD9h | | | |
| 047CDAh | | | |
| 047CDBh | | | |
| 047CDCh | | | |
| 047CDDh | | | |
| 047CDEh | CAN0 Mailbox 13: Time Stamp | | XXXXh |
| 047CDFh | | | |
| 047CE0h | CAN0 Mailbox 14: Message Identifier | C0MB14 | XXXX XXXXh |
| 047CE1h | | | |
| 047CE2h | | | |
| 047CE3h | | | |
| 047CE4h | | | |
| 047CE5h | CAN0 Mailbox 14: Data Length | | XXh |
| 047CE6h | CAN0 Mailbox 14: Data Field | | XXXX XXXX |
| 047CE7h | | | XXXX XXXXh |
| 047CE8h | | | |
| 047CE9h | | | |
| 047CEAh | | | |
| 047CEBh | | | |
| 047CECh | | | |
| 047CEDh | | | |
| 047CEEh | CAN0 Mailbox 14: Time Stamp | | XXXXh |
| 047CEFh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.45 SFR List (45)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047CF0h | CAN0 Mailbox 15: Message Identifier | C0MB15 | XXXX XXXXh |
| 047CF1h | | | |
| 047CF2h | | | |
| 047CF3h | | | |
| 047CF4h | | | |
| 047CF5h | CAN0 Mailbox 15: Data Length | | XXh |
| 047CF6h | CAN0 Mailbox 15: Data Field | | XXXX XXXX |
| 047CF7h | | | XXXX XXXXh |
| 047CF8h | | | |
| 047CF9h | | | |
| 047CFAh | | | |
| 047CFBh | | | |
| 047CFCh | | | |
| 047CFDh | | | |
| 047CFEh | CAN0 Mailbox 15: Time Stamp | | XXXXh |
| 047CFFh | | | |
| 047D00h | CAN0 Mailbox 16: Message Identifier | C0MB16 | XXXX XXXXh |
| 047D01h | | | |
| 047D02h | | | |
| 047D03h | | | |
| 047D04h | | | |
| 047D05h | CAN0 Mailbox 16: Data Length | | XXh |
| 047D06h | CAN0 Mailbox 16: Data Field | | XXXX XXXX |
| 047D07h | | | XXXX XXXXh |
| 047D08h | | | |
| 047D09h | | | |
| 047D0Ah | | | |
| 047D0Bh | | | |
| 047D0Ch | | | |
| 047D0Dh | | | |
| 047D0Eh | CAN0 Mailbox 16: Time Stamp | | XXXXh |
| 047D0Fh | | | |
| 047D10h | CAN0 Mailbox 17: Message Identifier | C0MB17 | XXXX XXXXh |
| 047D11h | | | |
| 047D12h | | | |
| 047D13h | | | |
| 047D14h | | | |
| 047D15h | CAN0 Mailbox 17: Data Length | | XXh |
| 047D16h | CAN0 Mailbox 17: Data Field | | XXXX XXXX |
| 047D17h | | | XXXX XXXXh |
| 047D18h | | | |
| 047D19h | | | |
| 047D1Ah | | | |
| 047D1Bh | | | |
| 047D1Ch | | | |
| 047D1Dh | | | |
| 047D1Eh | CAN0 Mailbox 17: Time Stamp | | XXXXh |
| 047D1Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.46 SFR List (46)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047D20h | CAN0 Mailbox 18: Message Identifier | C0MB18 | XXXX XXXXh |
| 047D21h | | | |
| 047D22h | | | |
| 047D23h | | | |
| 047D24h | | | |
| 047D25h | CAN0 Mailbox 18: Data Length | | XXh |
| 047D26h | CAN0 Mailbox 18: Data Field | | XXXX XXXX |
| 047D27h | | | XXXX XXXXh |
| 047D28h | | | |
| 047D29h | | | |
| 047D2Ah | | | |
| 047D2Bh | | | |
| 047D2Ch | | | |
| 047D2Dh | | | |
| 047D2Eh | CAN0 Mailbox 18: Time Stamp | | XXXXh |
| 047D2Fh | | | |
| 047D30h | CAN0 Mailbox 19: Message Identifier | C0MB19 | XXXX XXXXh |
| 047D31h | | | |
| 047D32h | | | |
| 047D33h | | | |
| 047D34h | | | |
| 047D35h | CAN0 Mailbox 19: Data Length | | XXh |
| 047D36h | CAN0 Mailbox 19: Data Field | | XXXX XXXX |
| 047D37h | | | XXXX XXXXh |
| 047D38h | | | |
| 047D39h | | | |
| 047D3Ah | | | |
| 047D3Bh | | | |
| 047D3Ch | | | |
| 047D3Dh | | | |
| 047D3Eh | CAN0 Mailbox 19: Time Stamp | | XXXXh |
| 047D3Fh | | | |
| 047D40h | CAN0 Mailbox 20: Message Identifier | C0MB20 | XXXX XXXXh |
| 047D41h | | | |
| 047D42h | | | |
| 047D43h | | | |
| 047D44h | | | |
| 047D45h | CAN0 Mailbox 20: Data Length | | XXh |
| 047D46h | CAN0 Mailbox 20: Data Field | | XXXX XXXX |
| 047D47h | | | XXXX XXXXh |
| 047D48h | | | |
| 047D49h | | | |
| 047D4Ah | | | |
| 047D4Bh | | | |
| 047D4Ch | | | |
| 047D4Dh | | | |
| 047D4Eh | CAN0 Mailbox 20: Time Stamp | | XXXXh |
| 047D4Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.47 SFR List (47)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047D50h | CAN0 Mailbox 21: Message Identifier | C0MB21 | XXXX XXXXh |
| 047D51h | | | |
| 047D52h | | | |
| 047D53h | | | |
| 047D54h | | | |
| 047D55h | CAN0 Mailbox 21: Data Length | | XXh |
| 047D56h | CAN0 Mailbox 21: Data Field | | XXXX XXXX |
| 047D57h | | | XXXX XXXXh |
| 047D58h | | | |
| 047D59h | | | |
| 047D5Ah | | | |
| 047D5Bh | | | |
| 047D5Ch | | | |
| 047D5Dh | | | |
| 047D5Eh | CAN0 Mailbox 21: Time Stamp | | XXXXh |
| 047D5Fh | | | |
| 047D60h | CAN0 Mailbox 22: Message Identifier | C0MB22 | XXXX XXXXh |
| 047D61h | | | |
| 047D62h | | | |
| 047D63h | | | |
| 047D64h | | | |
| 047D65h | CAN0 Mailbox 22: Data Length | | XXh |
| 047D66h | CAN0 Mailbox 22: Data Field | | XXXX XXXX |
| 047D67h | | | XXXX XXXXh |
| 047D68h | | | |
| 047D69h | | | |
| 047D6Ah | | | |
| 047D6Bh | | | |
| 047D6Ch | | | |
| 047D6Dh | | | |
| 047D6Eh | CAN0 Mailbox 22: Time Stamp | | XXXXh |
| 047D6Fh | | | |
| 047D70h | CAN0 Mailbox 23: Message Identifier | C0MB23 | XXXX XXXXh |
| 047D71h | | | |
| 047D72h | | | |
| 047D73h | | | |
| 047D74h | | | |
| 047D75h | CAN0 Mailbox 23: Data Length | | XXh |
| 047D76h | CAN0 Mailbox 23: Data Field | | XXXX XXXX |
| 047D77h | | | XXXX XXXXh |
| 047D78h | | | |
| 047D79h | | | |
| 047D7Ah | | | |
| 047D7Bh | | | |
| 047D7Ch | | | |
| 047D7Dh | | | |
| 047D7Eh | CAN0 Mailbox 23: Time Stamp | | XXXXh |
| 047D7Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.48 SFR List (48)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047D80h | CAN0 Mailbox 24: Message Identifier | C0MB24 | XXXX XXXXh |
| 047D81h | | | |
| 047D82h | | | |
| 047D83h | | | |
| 047D84h | | | |
| 047D85h | CAN0 Mailbox 24: Data Length | | XXh |
| 047D86h | CAN0 Mailbox 24: Data Field | | XXXX XXXX |
| 047D87h | | | XXXX XXXXh |
| 047D88h | | | |
| 047D89h | | | |
| 047D8Ah | | | |
| 047D8Bh | | | |
| 047D8Ch | | | |
| 047D8Dh | | | |
| 047D8Eh | CAN0 Mailbox 24: Time Stamp | | XXXXh |
| 047D8Fh | | | |
| 047D90h | CAN0 Mailbox 25: Message Identifier | C0MB25 | XXXX XXXXh |
| 047D91h | | | |
| 047D92h | | | |
| 047D93h | | | |
| 047D94h | | | |
| 047D95h | CAN0 Mailbox 25: Data Length | | XXh |
| 047D96h | CAN0 Mailbox 25: Data Field | | XXXX XXXX |
| 047D97h | | | XXXX XXXXh |
| 047D98h | | | |
| 047D99h | | | |
| 047D9Ah | | | |
| 047D9Bh | | | |
| 047D9Ch | | | |
| 047D9Dh | | | |
| 047D9Eh | CAN0 Mailbox 25: Time Stamp | | XXXXh |
| 047D9Fh | | | |
| 047DA0h | CAN0 Mailbox 26: Message Identifier | C0MB26 | XXXX XXXXh |
| 047DA1h | | | |
| 047DA2h | | | |
| 047DA3h | | | |
| 047DA4h | | | |
| 047DA5h | CAN0 Mailbox 26: Data Length | | XXh |
| 047DA6h | CAN0 Mailbox 26: Data Field | | XXXX XXXX |
| 047DA7h | | | XXXX XXXXh |
| 047DA8h | | | |
| 047DA9h | | | |
| 047DAAh | | | |
| 047DABh | | | |
| 047DACh | | | |
| 047DADh | | | |
| 047DAEh | CAN0 Mailbox 26: Time Stamp | | XXXXh |
| 047DAFh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.49 SFR List (49)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047DB0h | CAN0 Mailbox 27: Message Identifier | C0MB27 | XXXX XXXXh |
| 047DB1h | | | |
| 047DB2h | | | |
| 047DB3h | | | |
| 047DB4h | | | |
| 047DB5h | CAN0 Mailbox 27: Data Length | | XXh |
| 047DB6h | CAN0 Mailbox 27: Data Field | | XXXX XXXX |
| 047DB7h | | | XXXX XXXXh |
| 047DB8h | | | |
| 047DB9h | | | |
| 047DBAh | | | |
| 047DBBh | | | |
| 047DBCh | | | |
| 047DBDh | | | |
| 047DBEh | CAN0 Mailbox 27: Time Stamp | | XXXXh |
| 047DBFh | | | |
| 047DC0h | CAN0 Mailbox 28: Message Identifier | C0MB28 | XXXX XXXXh |
| 047DC1h | | | |
| 047DC2h | | | |
| 047DC3h | | | |
| 047DC4h | | | |
| 047DC5h | CAN0 Mailbox 28: Data Length | | XXh |
| 047DC6h | CAN0 Mailbox 28: Data Field | | XXXX XXXX |
| 047DC7h | | | XXXX XXXXh |
| 047DC8h | | | |
| 047DC9h | | | |
| 047DCAh | | | |
| 047DCBh | | | |
| 047DCCh | | | |
| 047DCDh | | | |
| 047DCEh | CAN0 Mailbox 28: Time Stamp | | XXXXh |
| 047DCFh | | | |
| 047DD0h | CAN0 Mailbox 29: Message Identifier | C0MB29 | XXXX XXXXh |
| 047DD1h | | | |
| 047DD2h | | | |
| 047DD3h | | | |
| 047DD4h | | | |
| 047DD5h | CAN0 Mailbox 29: Data Length | | XXh |
| 047DD6h | CAN0 Mailbox 29: Data Field | | XXXX XXXX |
| 047DD7h | | | XXXX XXXXh |
| 047DD8h | | | |
| 047DD9h | | | |
| 047DDAh | | | |
| 047DDBh | | | |
| 047DDCh | | | |
| 047DDDh | | | |
| 047DDEh | CAN0 Mailbox 29: Time Stamp | | XXXXh |
| 047DDFh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.50 SFR List (50)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047DE0h | CAN0 Mailbox 30: Message Identifier | C0MB30 | XXXX XXXXh |
| 047DE1h | | | |
| 047DE2h | | | |
| 047DE3h | | | |
| 047DE4h | | | |
| 047DE5h | CAN0 Mailbox 30: Data Length | | XXh |
| 047DE6h | CAN0 Mailbox 30: Data Field | | XXXX XXXX |
| 047DE7h | | | XXXX XXXXh |
| 047DE8h | | | |
| 047DE9h | | | |
| 047DEAh | | | |
| 047DEBh | | | |
| 047DECh | | | |
| 047DEDh | | | |
| 047DEEh | CAN0 Mailbox 30: Time Stamp | | XXXXh |
| 047DEFh | | | |
| 047DF0h | CAN0 Mailbox 31: Message Identifier | C0MB31 | XXXX XXXXh |
| 047DF1h | | | |
| 047DF2h | | | |
| 047DF3h | | | |
| 047DF4h | | | |
| 047DF5h | CAN0 Mailbox 31: Data Length | | XXh |
| 047DF6h | CAN0 Mailbox 31: Data Field | | XXXX XXXX |
| 047DF7h | | | XXXX XXXXh |
| 047DF8h | | | |
| 047DF9h | | | |
| 047DFAh | | | |
| 047DFBh | | | |
| 047DFCh | | | |
| 047DFDh | | | |
| 047DFEh | CAN0 Mailbox 31: Time Stamp | | XXXXh |
| 047DFFh | | | |
| 047E00h | CAN0 Acceptance Mask Register 0 | C0MKR0 | XXXX XXXXh |
| 047E01h | | | |
| 047E02h | | | |
| 047E03h | | | |
| 047E04h | CAN0 Acceptance Mask Register 1 | C0MKR1 | XXXX XXXXh |
| 047E05h | | | |
| 047E06h | | | |
| 047E07h | | | |
| 047E08h | CAN0 Acceptance Mask Register 2 | C0MKR2 | XXXX XXXXh |
| 047E09h | | | |
| 047E0Ah | | | |
| 047E0Bh | | | |
| 047E0Ch | CAN0 Acceptance Mask Register 3 | C0MKR3 | XXXX XXXXh |
| 047E0Dh | | | |
| 047E0Eh | | | |
| 047E0Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.51 SFR List (51)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047E10h | CAN0 Acceptance Mask Register 4 | C0MKR4 | XXXX XXXXh |
| 047E11h | | | |
| 047E12h | | | |
| 047E13h | | | |
| 047E14h | CAN0 Acceptance Mask Register 5 | C0MKR5 | XXXX XXXXh |
| 047E15h | | | |
| 047E16h | | | |
| 047E17h | | | |
| 047E18h | CAN0 Acceptance Mask Register 6 | C0MKR6 | XXXX XXXXh |
| 047E19h | | | |
| 047E1Ah | | | |
| 047E1Bh | | | |
| 047E1Ch | CAN0 Acceptance Mask Register 7 | C0MKR7 | XXXX XXXXh |
| 047E1Dh | | | |
| 047E1Eh | | | |
| 047E1Fh | | | |
| 047E20h | CAN0 FIFO Receive ID Compare Register 0 | C0FIDCR0 | XXXX XXXXh |
| 047E21h | | | |
| 047E22h | | | |
| 047E23h | | | |
| 047E24h | CAN0 FIFO Receive ID Compare Register 1 | C0FIDCR1 | XXXX XXXXh |
| 047E25h | | | |
| 047E26h | | | |
| 047E27h | | | |
| 047E28h | CAN0 Mask Invalid Register | C0MKIVLR | XXXX XXXXh |
| 047E29h | | | |
| 047E2Ah | | | |
| 047E2Bh | | | |
| 047E2Ch | CAN0 Mailbox Interrupt Enable Register | C0MIER | XXXX XXXXh |
| 047E2Dh | | | |
| 047E2Eh | | | |
| 047E2Fh | | | |
| 047E30h | | | |
| 047E31h | | | |
| 047E32h | | | |
| 047E33h | | | |
| 047E34h | | | |
| 047E35h | | | |
| 047E36h | | | |
| 047E37h | | | |
| 047E38h | | | |
| 047E39h | | | |
| 047E3Ah | | | |
| 047E3Bh | | | |
| 047E3Ch | | | |
| 047E3Dh | | | |
| 047E3Eh | | | |
| 047E3Fh | | | |
| 047E40h to 047F1Fh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.52 SFR List (52)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047F20h | CAN0 Message Control Register 0 | C0MCTL0 | 00h |
| 047F21h | CAN0 Message Control Register 1 | C0MCTL1 | 00h |
| 047F22h | CAN0 Message Control Register 2 | C0MCTL2 | 00h |
| 047F23h | CAN0 Message Control Register 3 | C0MCTL3 | 00h |
| 047F24h | CAN0 Message Control Register 4 | C0MCTL4 | 00h |
| 047F25h | CAN0 Message Control Register 5 | C0MCTL5 | 00h |
| 047F26h | CAN0 Message Control Register 6 | C0MCTL6 | 00h |
| 047F27h | CAN0 Message Control Register 7 | C0MCTL7 | 00h |
| 047F28h | CAN0 Message Control Register 8 | C0MCTL8 | 00h |
| 047F29h | CAN0 Message Control Register 9 | C0MCTL9 | 00h |
| 047F2Ah | CAN0 Message Control Register 10 | C0MCTL10 | 00h |
| 047F2Bh | CAN0 Message Control Register 11 | C0MCTL11 | 00h |
| 047F2Ch | CAN0 Message Control Register 12 | C0MCTL12 | 00h |
| 047F2Dh | CAN0 Message Control Register 13 | C0MCTL13 | 00h |
| 047F2Eh | CAN0 Message Control Register 14 | C0MCTL14 | 00h |
| 047F2Fh | CAN0 Message Control Register 15 | C0MCTL15 | 00h |
| 047F30h | CAN0 Message Control Register 16 | C0MCTL16 | 00h |
| 047F31h | CAN0 Message Control Register 17 | C0MCTL17 | 00h |
| 047F32h | CAN0 Message Control Register 18 | C0MCTL18 | 00h |
| 047F33h | CAN0 Message Control Register 19 | C0MCTL19 | 00h |
| 047F34h | CAN0 Message Control Register 20 | C0MCTL20 | 00h |
| 047F35h | CAN0 Message Control Register 21 | C0MCTL21 | 00h |
| 047F36h | CAN0 Message Control Register 22 | C0MCTL22 | 00h |
| 047F37h | CAN0 Message Control Register 23 | C0MCTL23 | 00h |
| 047F38h | CAN0 Message Control Register 24 | C0MCTL24 | 00h |
| 047F39h | CAN0 Message Control Register 25 | C0MCTL25 | 00h |
| 047F3Ah | CAN0 Message Control Register 26 | C0MCTL26 | 00h |
| 047F3Bh | CAN0 Message Control Register 27 | C0MCTL27 | 00h |
| 047F3Ch | CAN0 Message Control Register 28 | C0MCTL28 | 00h |
| 047F3Dh | CAN0 Message Control Register 29 | C0MCTL29 | 00h |
| 047F3Eh | CAN0 Message Control Register 30 | C0MCTL30 | 00h |
| 047F3Fh | CAN0 Message Control Register 31 | C0MCTL31 | 00h |

X: Undefined
Blanks are reserved. No access is allowed.

**Table 4.53 SFR List (53)**

| Address | Register | Symbol | Reset Value |
|---|---|---|---|
| 047F40h | CAN0 Control Register | C0CTLR | 0000 0101b |
| 047F41h | | | 0000 0000b |
| 047F42h | CAN0 Status Register | C0STR | 0000 0101b |
| 047F43h | | | 0000 0000b |
| 047F44h | CAN0 Bit Configuration Register | C0BCR | 00 0000h |
| 047F45h | | | |
| 047F46h | | | |
| 047F47h | CAN0 Clock Select Register | C0CLKR | 000X 0000b |
| 047F48h | CAN0 Receive FIFO Control Register | C0RFCR | 1000 0000b |
| 047F49h | CAN0 Receive FIFO Pointer Control Register | C0RFPCR | XXh |
| 047F4Ah | CAN0 Transmit FIFO Control Register | C0TFCR | 1000 0000b |
| 047F4Bh | CAN0 Transmit FIFO Pointer Control Register | C0TFPCR | XXh |
| 047F4Ch | CAN0 Error Interrupt Enable Register | C0EIER | 00h |
| 047F4Dh | CAN0 Error Interrupt Factor Judge Register | C0EIFR | 00h |
| 047F4Eh | CAN0 Reception Error Count Register | C0RECR | 00h |
| 047F4Fh | CAN0 Transmission Error Count Register | C0TECR | 00h |
| 047F50h | CAN0 Error Code Store Register | C0ECSR | 00h |
| 047F51h | CAN0 Channel Search Support Register | C0CSSR | XXh |
| 047F52h | CAN0 Mailbox Search Status Register | C0MSSR | 1000 0000b |
| 047F53h | CAN0 Mailbox Search Mode Register | C0MSMR | XXXX XX00b |
| 047F54h | CAN0 Time Stamp Register | C0TSR | 0000h |
| 047F55h | | | |
| 047F56h | CAN0 Acceptance Filter Support Register | C0AFSR | XXXXh |
| 047F57h | | | |
| 047F58h | CAN0 Test Control Register | C0TCR | 00h |
| 047F59h | | | |
| 047F5Ah | | | |
| 047F5Bh | | | |
| 047F5Ch | | | |
| 047F5Dh | | | |
| 047F5Eh | | | |
| 047F5Fh | | | |
| 047F60h to 047FFFh | | | |
| 048000h to 04FFFFh | | | |

X: Undefined
Blanks are reserved. No access is allowed.

# 5. Resets

Three types of reset operations can be used to reset the MCU: hardware reset, software reset, and watchdog timer reset.

## 5.1 Hardware Reset

A hardware reset is generated when a low signal is applied to the $\overline{\text{RESET}}$ pin under the recommended operating conditions of supply voltage (refer to Table 5.1). When the $\overline{\text{RESET}}$ pin is driven low, all pins, and oscillators are initialized, and the main clock starts oscillating. The CPU and SFRs are initialized by a low-to-high transition on the $\overline{\text{RESET}}$ pin. Then, the CPU starts executing the program of the address indicated by the reset vector. The internal RAM is not affected by a hardware reset. However, if a hardware reset occurs during a write to the internal RAM, the content is undefined.
Figure 5.1 shows an example of the reset circuit. Figure 5.2 shows the reset sequence. Table 5.1 lists pin states while the $\overline{\text{RESET}}$ pin is held low. Figure 5.3 shows CPU register states after reset. For the SFR states after reset, refer to 4. “Special Function Registers (SFRs)”.

A. Reset on a stable supply voltage
   (1) Drive the $\overline{\text{RESET}}$ pin low
   (2) Provide 20 or more clock cycle inputs into the XIN pin
   (3) Drive the $\overline{\text{RESET}}$ pin high

B. Power-on reset
   (1) Drive the $\overline{\text{RESET}}$ pin low
   (2) Raise the supply voltage to the recommended operating level
   (3) Insert td(P-R) ms as wait time to stabilize the internal voltage
   (4) Provide 20 or more clock cycle inputs into the XIN pin
   (5) Drive the $\overline{\text{RESET}}$ pin high

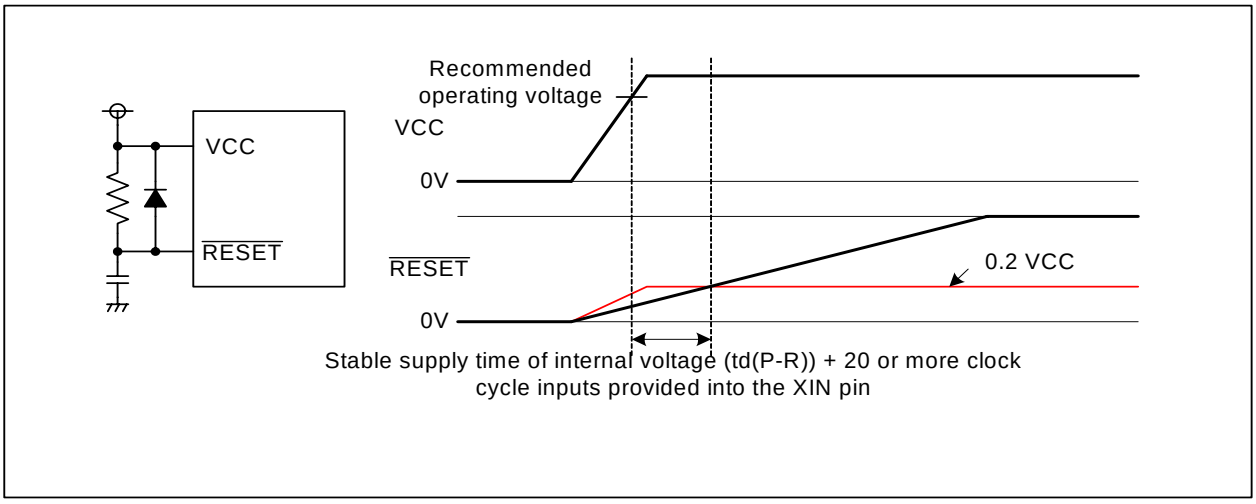
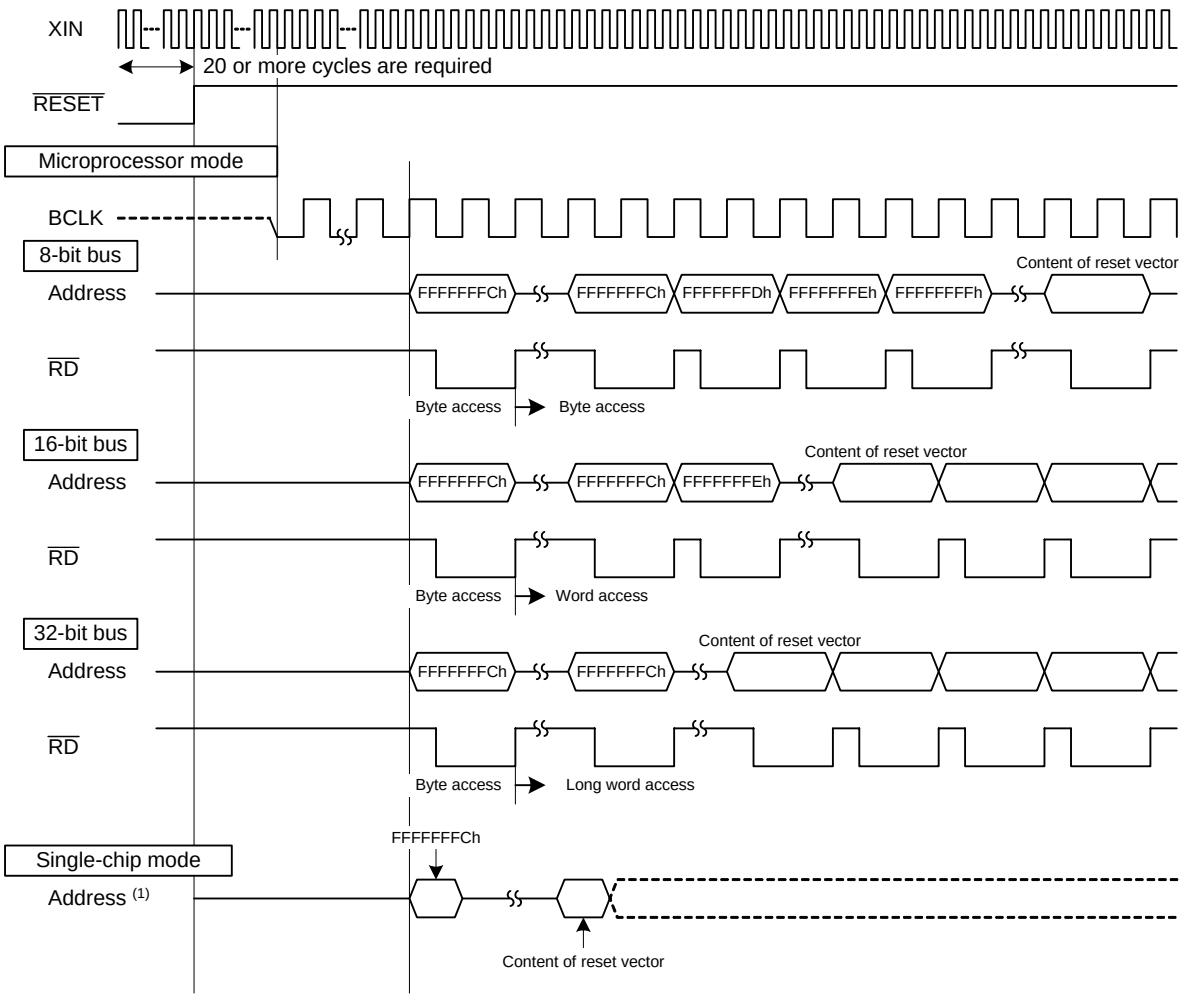



**Figure 5.1 Reset Circuitry**




Note:
1. Address data is not output from pins in single-chip mode.

**Figure 5.2 Reset Sequence**

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

| Pin Name | Pin States | |
|---|---|---|
| | CNVSS = VSS | CNVSS = VCC |
| P0 | Input port (high-impedance) | Inputs data |
| P1 | Input port (high-impedance) | Input port (high-impedance) |
| P2, P3 | Input port (high-impedance) | Output addresses (undefined) |
| P4_0 to P4_6 | Input port (high-impedance) | Output addresses (undefined) |
| P4_7 | Input port (high-impedance) | Outputs the $\overline{\text{CS0}}$ signal (high) |
| P5_0 | Input port (high-impedance) | Outputs the $\overline{\text{WR}}$ signal (high) |
| P5_1 | Input port (high-impedance) | Outputs the $\overline{\text{BC1}}$ signal (undefined) |
| P5_2 | Input port (high-impedance) | Outputs the $\overline{\text{RD}}$ signal (high) |
| P5_3 | Input port (high-impedance) | Outputs the BCLK (2) |
| P5_4 | Input port (high-impedance) | Outputs the $\overline{\text{HLDA}}$ signal (output signal depends on an input signal to the $\overline{\text{HOLD}}$ pin) (2) |
| P5_5 | Input port (high-impedance) | Inputs the $\overline{\text{HOLD}}$ signal (high-impedance) |
| P5_6 | Input port (high-impedance) | Outputs the $\overline{\text{CS2}}$ signal (high) |
| P5_7 | Input port (high-impedance) | Inputs the $\overline{\text{RDY}}$ signal (high-impedance) |
| P6 to P10 | Input port (high-impedance) | Input port (high-impedance) |
| P11 to P15 (3) | Input port (high-impedance) | Input port (high-impedance) |

Notes:
1. Whether a pull-up resistor is enabled or not is undefined until the internal voltage has stabilized.
2. State after power is on and the internal voltage has stabilized. It is undefined until the internal voltage has stabilized
3. Ports P11 to P15 are available in the 144-pin package only.

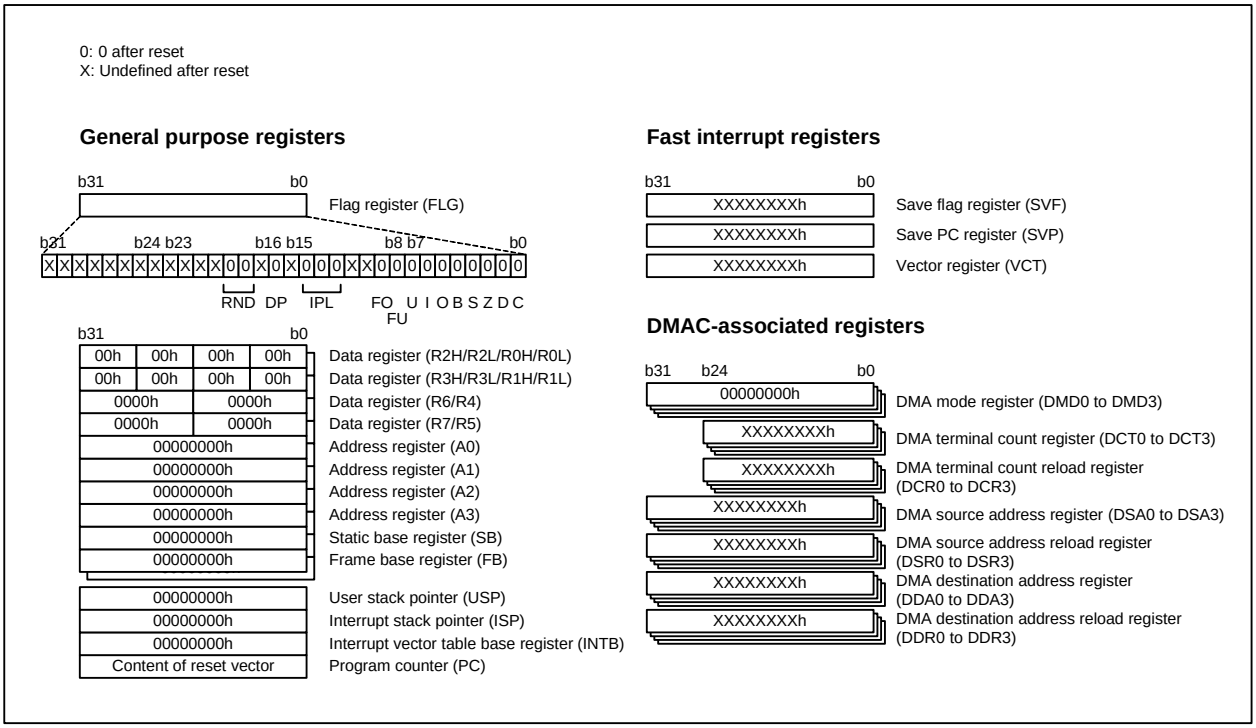



**Figure 5.3 CPU Registers after Reset**

## 5.2 Software Reset

A software reset is generated when the PM03 bit in the PM0 register is set to 1 (MCU is reset). When a software reset is released, the CPU, SFRs, and pins are initialized. Then, the CPU starts executing the program from the address indicated by the reset vector.

The PM03 bit should be set to 1 while the PLL clock is selected as the CPU clock source and the main clock oscillation is completely stable.

Processor mode remains unchanged since bits PM01 and PM00 in the PM0 register are not affected by a software reset.

## 5.3 Watchdog Timer Reset

A watchdog timer reset is generated when the watchdog timer underflows while the CM06 bit in the CM0 register is 1 (the MCU is reset if the watchdog timer underflows). When the watchdog timer reset is released, the CPU, SFRs, and pins are initialized. Then, the CPU starts executing the program from the address indicated by the reset vector.

Processor mode remains unchanged since bits PM01 and PM00 in the PM0 register are not affected by a watchdog timer reset.

## 5.4 Reset Vector

The reset vector in the R32C/100 Series is configured as shown in Figure 5.4.
The 32-bit start address of a program must be a multiple of 4. Because of this, the address always ends with two zero bits. The reset vector contains the upper 30 bits of the start address in bits 2 to 31. Bits 0 and 1 of the reset vector are used to select the external bus width in microprocessor mode.
In single-chip mode, these bits should be set to 00b.

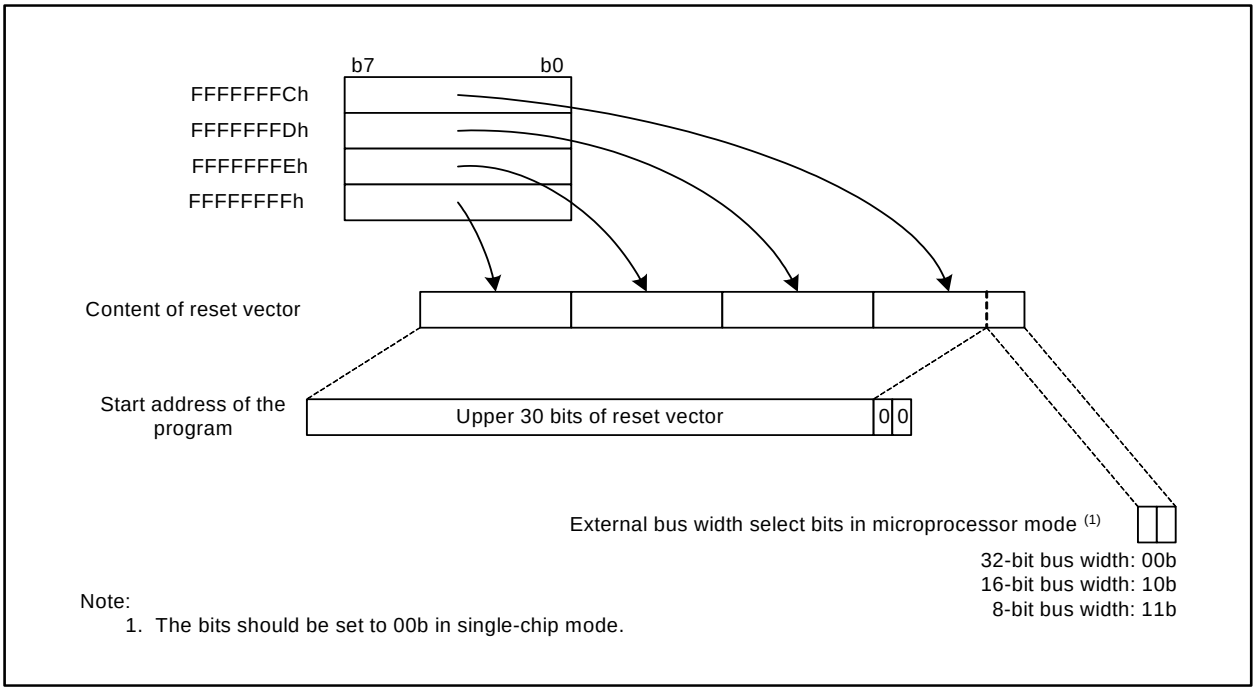



**Figure 5.4 Reset Vector Configuration**

# 6. Power Management

## 6.1 Voltage Regulators for Internal Logic

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

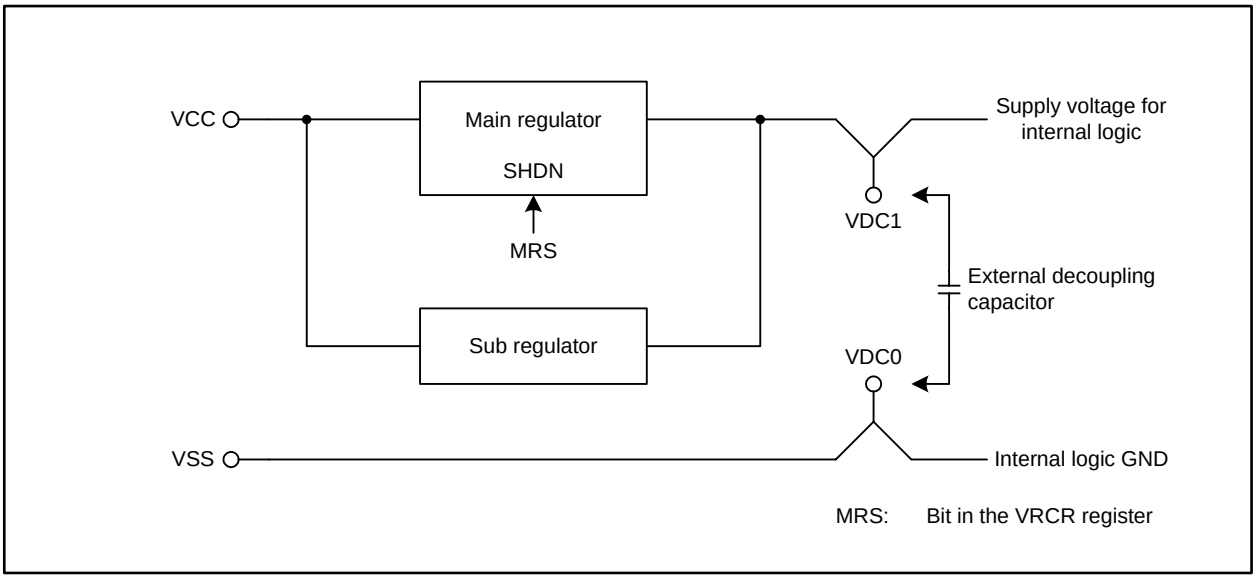



**Figure 6.1 Block Diagram of Voltage Regulators for Internal Logic**

Voltage Regulator Control Register (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | X | |

| Symbol | Address | Reset Value |
|---|---|---|
| VRCR | 040060h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| MRS | Main Regulator Shut-down Bit (2) | 0: Main regulator active<br>1: Main regulator stopped (3) | RW |
| —<br>(b7-b1) | No register bits; should be written with 0 and read as 0 | | — |

Notes:
1. Set the PRC31 bit in the PRCR3 register to 1 (write enabled) before rewriting this register.
2. This bit is fixed to 0 if the CM05 bit in the CM0 register is 0 (main clock oscillator active) or the CM10 bit in the CM1 register is 0 (PLL clock oscillator active) .
3. While the main regulator is stopped, do not rewrite the flash memory.

**Figure 6.2 VRCR Register**

### 6.1.1 Decoupling Capacitor

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

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

**Table 6.1 Recommended Capacitor Specifications**

| Temperature Characteristics | | | | Rated Voltage (V) | Nominal Capacitance (µF) | Capacitance Tolerance (%) |
|---|---|---|---|---|---|---|
| Applicable standard | | Operating temperature range (°C) | Capacitance change (%) | | | |
| B | JIS | -25 to 85 | ±10 | 6.3 or higher | 4.7 | ±20 or better |
| R | JIS | -55 to 125 | ±15 | 6.3 or higher | 4.7 | ±20 or better |
| X5R | EIA | -55 to 85 | ±15 | 6.3 or higher | 4.7 | ±20 or better |
| X7R | EIA | -55 to 125 | ±15 | 6.3 or higher | 4.7 | ±20 or better |
| X8R | EIA | -55 to 150 | ±15 | 6.3 or higher | 4.7 | ±20 or better |
| X6S | EIA | -55 to 105 | ±22 | 6.3 or higher | 4.7 | ±20 or better |
| X7S | EIA | -55 to 125 | ±22 | 6.3 or higher | 4.7 | ±20 or better |

## 6.2 Low Voltage Detector

The low voltage detector monitors the supply voltage of VCC pin.
This circuit is used to monitor the power supply upstream of the voltage regulators for internal logic and provide advanced warning that the power is about to fail. By providing a few milliseconds of advanced warning, the CPU can save any critical parameters to the flash memory and gracefully shut down.
Figure 6.3 shows a block diagram of the low voltage detector and Figure 6.4 and Figure 6.5 show registers associated with the circuit.

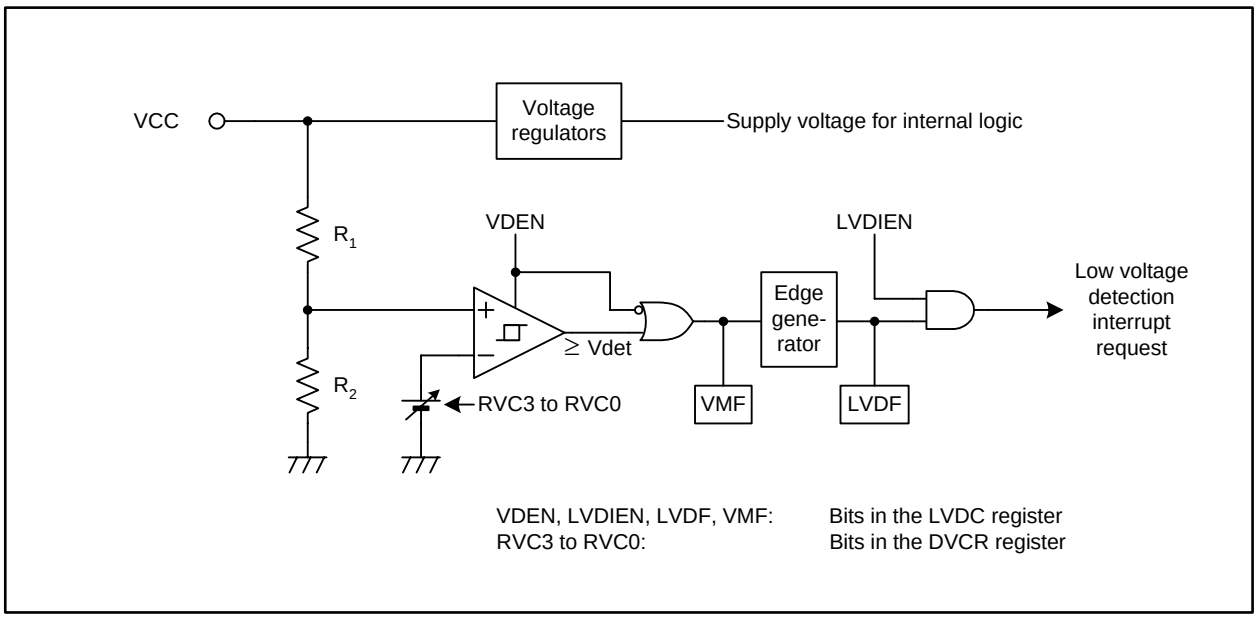



**Figure 6.3 Low Voltage Detector Block Diagram**

## Low Voltage Detector Control Register (1)

b7 b6 b5 b4 b3 b2 b1 b0

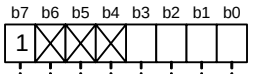
| Symbol | Address | Reset Value |
|---|---|---|
| LVDC | 40062h | 0000 XX00b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| VDEN | Low Voltage Detection Enable Bit | 0: Low voltage detection disabled<br>1: Low voltage detection enabled | RW |
| LVDIEN | Low Voltage Detection Interrupt Enable Bit (2) | 0: Interrupt disabled<br>1: Interrupt enabled | RW |
| LVDF | Low Voltage Detection Flag (3, 4) | 0: Low voltage undetected<br>1: Low voltage detected (Vdet passed) | RW |
| VMF | Voltage Monitor Flag (3) | 0: VCC < Vdet<br>1: VCC ≥ Vdet or low voltage detection disabled | RO |
| — (b7-b4) | No register bits; should be written with 0 and read as 0 | | — |

Notes:
1. Set the PRC31 bit in the PRCR3 register to 1 (write enabled) before rewriting this register.
2. Before setting this bit to 1, set the VDEN bit to 1 (low voltage detection enabled) first, and wait until the circuit has stabilized.
3. This bit is valid when the VDEN bit is set to 1 (low voltage detection enabled).
4. This bit becomes 0 if a 0 is written to (Writing a 1 to this bit has no effect).

**Figure 6.4 LVDC Register**

## Detection Voltage Configuration Register (1)




Symbol: DVCR
Address: 40064h
Reset Value: 0000 XXXXb

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| RVC0 | Reference Voltage Configuration Bit (2) | b3 b2 b1 b0<br>0 0 0 0 : 3.90 V<br>0 0 0 1 : 3.75 V<br>0 0 1 0 : 3.60 V<br>0 0 1 1 : 3.45 V<br>0 1 0 0 : 3.30 V<br>1 0 1 1 : 4.65 V<br>1 1 0 0 : 4.50 V<br>1 1 0 1 : 4.35 V<br>1 1 1 0 : 4.20 V<br>1 1 1 1 : 4.05 V<br>Only use the combinations listed above | RW |
| RVC1 | | | RW |
| RVC2 | | | RW |
| RVC3 | | | RW |
| — (b6-b4) | No register bits; should be written with 0 and read as 0 | | — |
| — (b7) | Reserved | Should be written with 1 | RW |

Notes:
1. Set the PRC31 bit in the PRCR3 register to 1 (write enabled) before rewriting this register. The rewriting should be performed when the VDEN bit in the LVDC register is set to 0 (low voltage detection disabled).
2. Refer to the table below for detected voltages Vdet(F) and Vdet(R).

| Reference Voltage | Low-detection Voltage Vdet(F) | Rise-detection Voltage Vdet(R) |
|---|---|---|
| 4.65 V | 4.55 V | 4.77 V |
| 4.50 V | 4.40 V | 4.62 V |
| 4.35 V | 4.24 V | 4.46 V |
| 4.20 V | 4.09 V | 4.31 V |
| 4.05 V | 3.95 V | 4.17 V |
| 3.90 V | 3.80 V | 4.02 V |
| 3.75 V | 3.65 V | 3.87 V |
| 3.60 V | 3.50 V | 3.72 V |
| 3.45 V | 3.35 V | 3.57 V |
| 3.30 V | 3.20 V | 3.42 V |

**Figure 6.5 DVCR register**

### 6.2.1 Operational State of Low Voltage Detector

The low voltage detector starts running after td(E-A) if the VDEN bit in the LVDC register is set to 1 (low voltage detection enabled).

When the input voltage to the VCC pin has dropped below Vdet(F), the VMF bit becomes 0 (VCC < Vdet) and the LVDF bit becomes 1 (low voltage detected (Vdet passed)). Then an interrupt request occurs if the LVDIEN bit is set to 1 (low voltage detection interrupt enabled). The LVDF bit should be set to 0 (low voltage undetected) by a program.

When the voltage has re-risen above Vdet(R), the VMF bit becomes to 1 (VCC ≥ Vdet) and the LVDF bit becomes 1 (low voltage detected (Vdet passed)). Then an interrupt request occurs if the LVDIEN bit is set to 1 (low voltage detection interrupt enabled).

Figure 6.6 shows the operational state of low voltage detector.

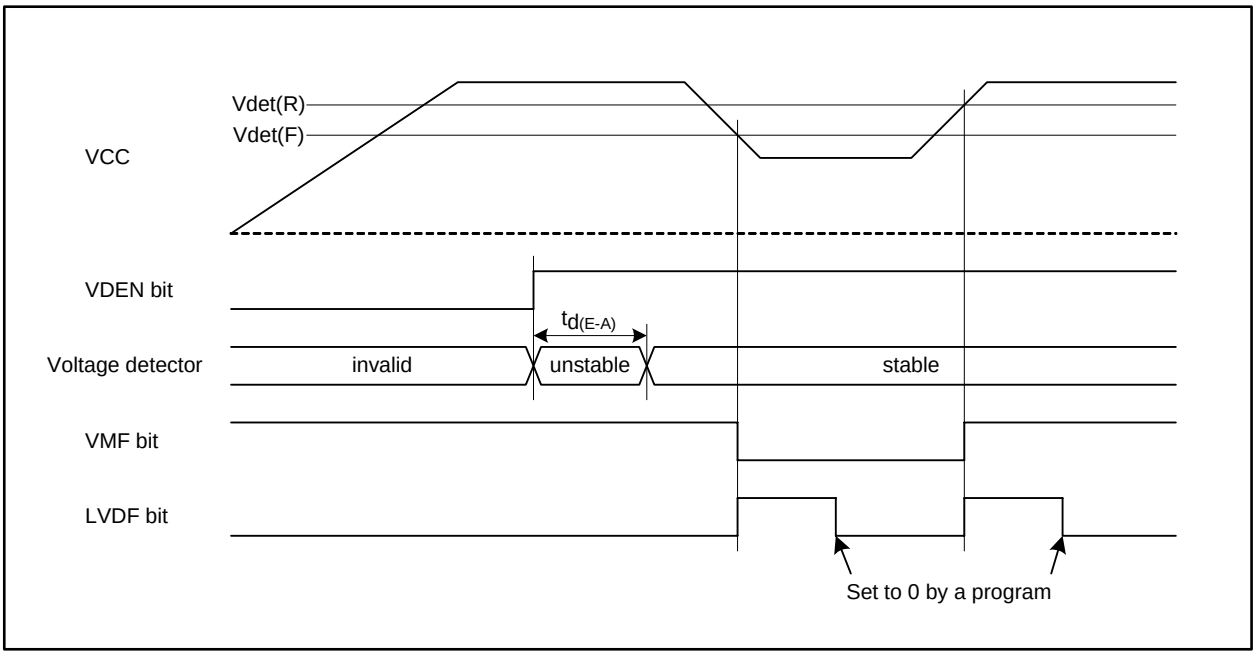



**Figure 6.6 Operational State of Low Voltage Detector**

### 6.2.2 Low Voltage Detection Interrupt

The low voltage detection interrupt occurs when the input voltage at the VCC pin rises to the Vdet(R) level and above, or falls below the Vdet(F) level if the LVDIEN bit in the LVDC register is set to 1 (low voltage detection interrupt enabled).

This interrupt shares the interrupt vector table with the watchdog timer interrupt and oscillator stop detection interrupt. In case of simultaneous use with other(s), it should be confirmed that the low voltage detection interrupt has occurred by reading the LVDF bit in the LVDC register in the interrupt handler.

The LVDF bit becomes 1 when the input voltage at the VCC pin has passed the Vdet(R) level or Vdet(F) level. When the LVDF bit changes from 0 to 1, a low voltage detection interrupt request occurs. This bit should be set to 0 (low voltage undetected) by a program.

### 6.2.3 An Application of Low Voltage Detector

Figure 6.7 shows an application of the low voltage detection interrupt.
The supply voltage for internal logic is generated by reducing the input voltage from the VCC pin with the voltage regulators. When the input voltage begins to fall, the internal voltage stays steady. Eventually, as the input voltage continues to fall, it begins to fall, which may affect the MCU operation. Consequently the system can be gracefully shut down from when the input voltage begins to fall until when the internal voltage begins to fall. The low voltage detection interrupt can be applied to detect the input voltage falling.

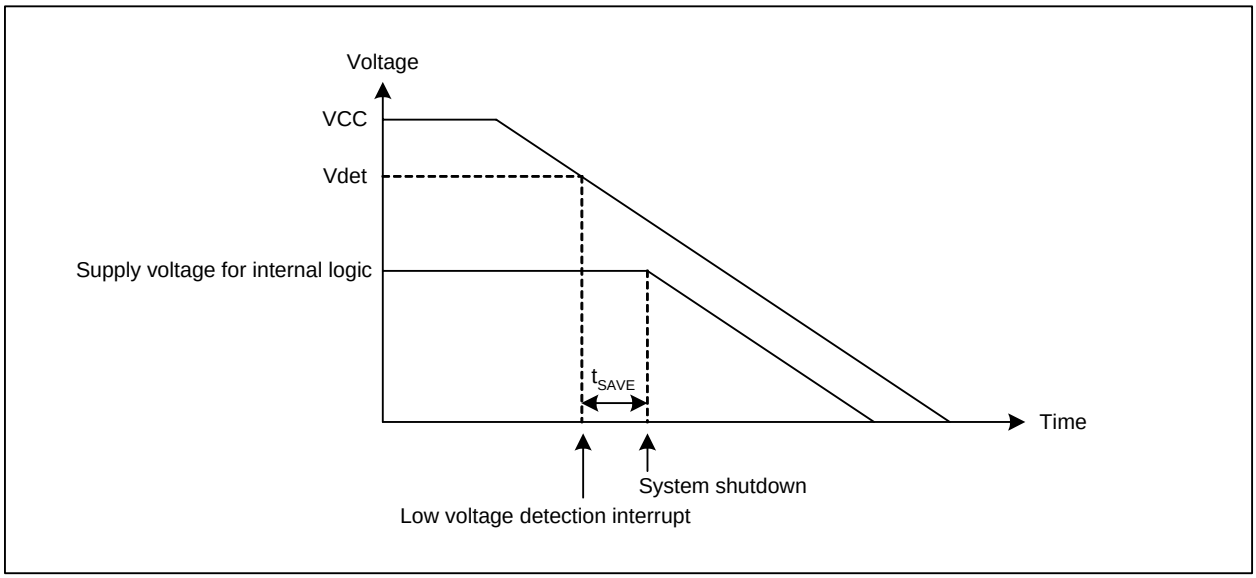



**Figure 6.7 Low Voltage Detection Interrupt**

# 7. Processor Mode

## 7.1 Types of Processor Mode

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

**Table 7.1 Processor Mode Characteristics**

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

Note:
1. Refer to 9. “Bus” for details.

The R32C/118 Group supports single-chip mode and memory expansion mode. The microprocessor mode is optional. Please contact a Renesas sales office to use this mode.

## 7.2 Processor Mode Setting

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

**Table 7.2 Processor Mode After Hardware Reset**

| Input Level into the CNVSS Pin (1) | Processor Mode |
|---|---|
| Low | Single-chip mode |
| High | Microprocessor mode |

Note:
1. The CNVSS pin should be connected to the VCC or VSS via a resistor.

To change to another processor mode after starting an operation in single-chip mode or microprocessor mode, bits PM01 and PM00 in the PM0 register should be rewritten. For example, to change the processor mode from single-chip mode to memory expansion mode, these bits should be set to 01b (memory expansion mode). Note that only when microprocessor mode is selected to start, the internal ROM is inaccessible irrespective of the setting of bits PM01 and PM00.

The notes on changing processor mode are as follows:
1. When bits PM01 and PM00 are rewritten with 01b (memory expansion mode) or 11b (microprocessor mode), bits PM07 to PM02 should not be rewritten simultaneously.
2. To rewrite bits PM02 to PM07, bits PM01 and PM00 should not be changed.
3. A processor mode should not be shifted to the microprocessor mode during the execution of a program in the internal ROM.
4. A processor mode should not be shifted to the single-chip mode during the execution of a program in an external space.
5. A processor mode should not be shifted to the memory expansion mode during the execution of a program in the same address as that assigned to the internal ROM.

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

## Processor Mode Register 0 (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | | | | |

Symbol: PM0

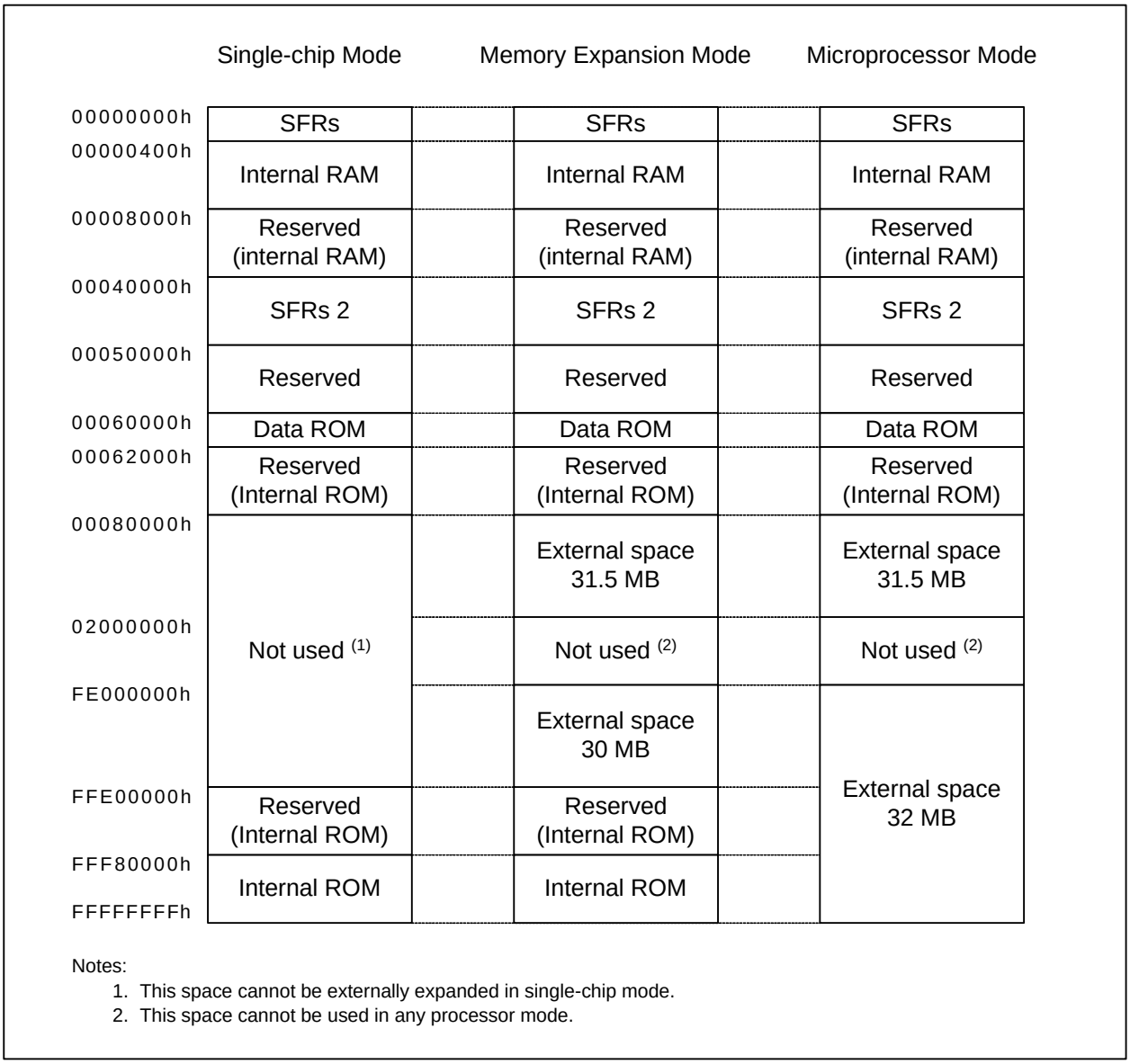
Address: 44044h

Reset Value: 1000 0000b (CNVSS pin is held low)
0000 0011b (CNVSS pin is held high)

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PM00 | Processor Mode Bit (2, 3) | b1 b0<br>0 0 : Single-chip mode<br>0 1 : Memory expansion mode<br>1 0 : Do not use this combination<br>1 1 : Microprocessor mode | RW |
| PM01 | | | RW |
| PM02 | R/W Mode Select Bit | 0: $\overline{RD}$ / $\overline{WR}$ / $\overline{BC0}$ / $\overline{BC1}$ / $\overline{BC2}$ / $\overline{BC3}$<br>1: $\overline{RD}$ / $\overline{WR0}$ / $\overline{WR1}$ / $\overline{WR2}$ / $\overline{WR3}$ | RW |
| PM03 | Software Reset Bit | The MCU is reset when this bit is set to 1. The bit is read as 0 | RW |
| — (b6-b4) | Reserved | Should be written with 0 | RW |
| PM07 | BCLK Output Function Select Bit (4) | 0: Output BCLK (5)<br>1: Do not output BCLK. Select the pin function using bits CM01 and CM00 in the CM0 register | RW |

Notes:
1. This register should be rewritten after the PRC1 bit in the PRCR register is set to 1 (write enabled).
2. The processor mode is not changed even when the PM03 bit is set to 1 (software reset).
3. To rewrite bits PM01 and PM00 with 01b or 11b, other bit(s) to be rewritten should be done prior to the rewrite of these bits. They should not be rewritten simultaneously.
4. In single-chip mode, the BCLK is not output even when the PM07 bit is set to 0.
   To stop clock output in memory expansion mode or microprocessor mode, the PM07 bit should be set to 1 and bits CM01 and CM00 in the CM0 register should be set to 00b (I/O port P5_3). The I/O port P5_3 outputs a low signal.
5. When the PM07 bit is set to 0 (output BCLK), bits CM01 and CM00 should be set to 00b.

**Figure 7.1 PM0 Register**




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

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

# 8. Clock Generator

## 8.1 Clock Generator Types

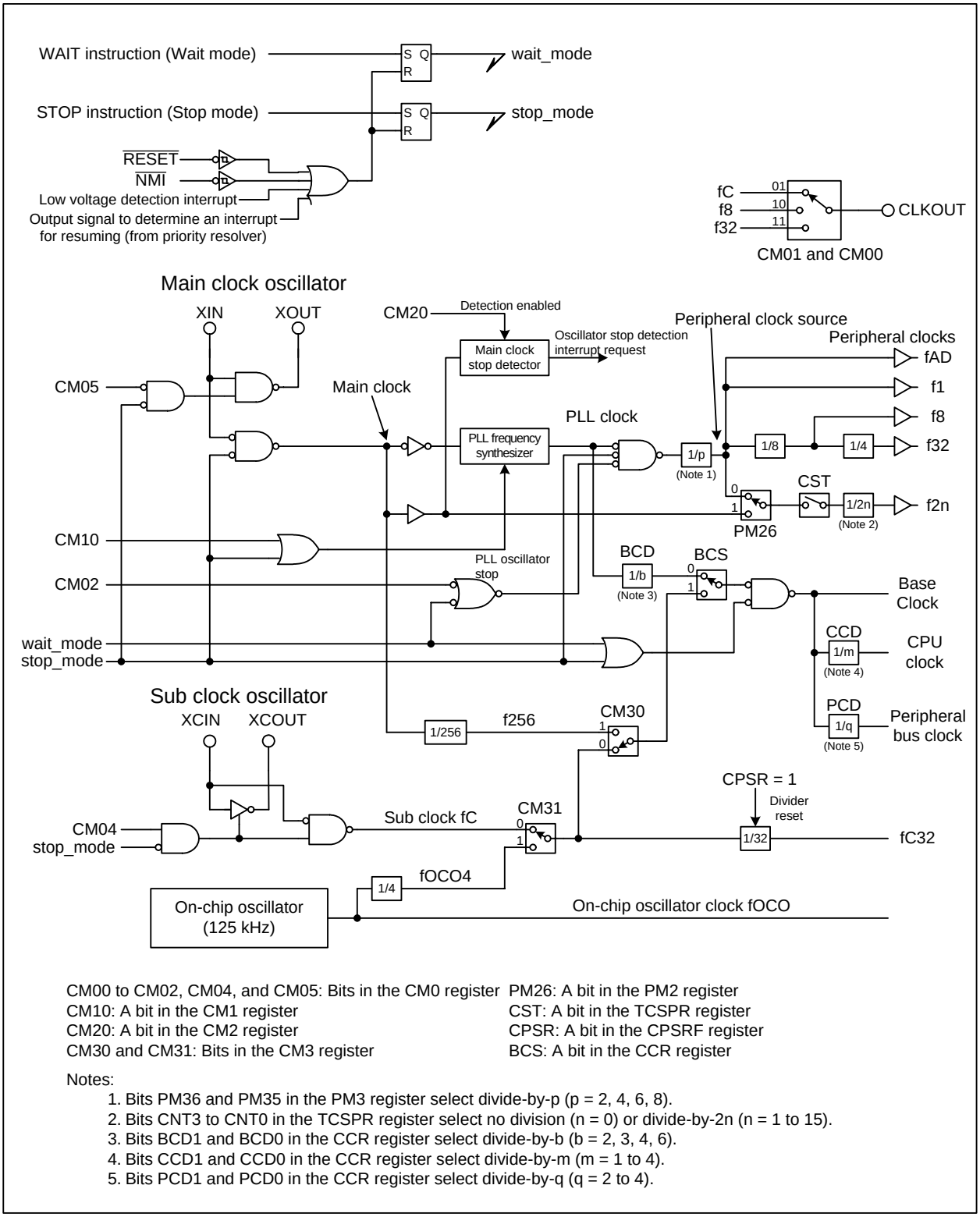
Four circuits are included to generate a system clock signal:

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

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

**Table 8.1 Clock Generator Specifications**

| Item | Main Clock Oscillator | Sub Clock Oscillator | PLL Frequency Synthesizer | On-chip Oscillator |
|---|---|---|---|---|
| Used as | • PLL reference clock source<br>• Peripheral clock source | • CPU clock source<br>• Clock source for timers A and B | • CPU clock source<br>• Peripheral clock source | • CPU clock source<br>• Clock source for timers A and B |
| Clock frequency | 4 to 16 MHz | 32.768 kHz | $f_{SO(PLL)}$ or $f_{(PLL)}$ | Approx. 125 kHz |
| Connectable oscillators or additional circuits | Ceramic resonator<br>Crystal oscillator | Crystal oscillator | — | — |
| Pins for oscillators or additional circuits | XIN, XOUT | XCIN, XCOUT | — | — |
| Oscillator stop, Restart | Available | Available | Available | Available |
| Oscillator state after reset | Running | Stopped | Running | Stopped |
| Note | Externally generated clock can be input | Externally generated clock can be input | When the main clock oscillator stops running, the PLL frequency synthesizer oscillates at its own frequency, $f_{SO(PLL)}$ | |




CM00 to CM02, CM04, and CM05: Bits in the CM0 register
CM10: A bit in the CM1 register
CM20: A bit in the CM2 register
CM30 and CM31: Bits in the CM3 register
PM26: A bit in the PM2 register
CST: A bit in the TCSPR register
CPSR: A bit in the CPSRF register
BCS: A bit in the CCR register

Notes:
1. Bits PM36 and PM35 in the PM3 register select divide-by-p (p = 2, 4, 6, 8).
2. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
3. Bits BCD1 and BCD0 in the CCR register select divide-by-b (b = 2, 3, 4, 6).
4. Bits CCD1 and CCD0 in the CCR register select divide-by-m (m = 1 to 4).
5. Bits PCD1 and PCD0 in the CCR register select divide-by-q (q = 2 to 4).

**Figure 8.1 Clock Generation Circuitry**

## Clock Control Register (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | 0 | | | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| CCR | 0004h | 0001 1000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| BCD0 | Base Clock Divide Ratio Select Bit (2) | b1 b0<br>0 0 : Divide-by-6<br>0 1 : Divide-by-4<br>1 0 : Divide-by-3<br>1 1 : Divide-by-2 | RW |
| BCD1 | | | RW |
| CCD0 | CPU Clock Divide Ratio Select Bit (3) | b3 b2<br>0 0 : Divide-by-4<br>0 1 : Divide-by-3<br>1 0 : Divide-by-2<br>1 1 : No division | RW |
| CCD1 | | | RW |
| PCD0 | Peripheral Bus Clock Divide Ratio Select Bit (2, 3, 4) | b5 b4<br>0 0 : Do not use this combination<br>0 1 : Divide-by-2<br>1 0 : Divide-by-3<br>1 1 : Divide-by-4 | RW |
| PCD1 | | | RW |
| — (b6) | Reserved | Should be written with 0 | RW |
| BCS | Base Clock Source Select Bit | 0: PLL clock<br>1: fC, fOCO4, or f256 (5, 6) | RW |

Notes:
1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. The divide ratios of the base clock and peripheral bus clock should not be changed simultaneously. Otherwise, the peripheral bus clock frequency may be over the operational maximum. To increase the base clock frequency, the divide ratio of the peripheral bus clock should be increased before reducing the divide ratio of base clock.
3. The divide ratio of the CPU clock should be equal to or lower than that of peripheral bus clock.
4. This bit should be set only once after reset and the setting should not be changed. To rewrite this bit, the PBC register should be rewritten first.
5. To set this bit to 1, a 32-bit write access to addresses 0004h to 0007h should be performed.
6. These low speed clocks are switched by setting bits CM31 and CM30 in the CM3 register.

**Figure 8.2 CCR Register**

## System Clock Control Register 0 (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | | | | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| CM0 | 40046h | 0000 1000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CM00 | Clock Output Function Select Bit (2) | b1 b0<br>0 0 : I/O port P5_3<br>0 1 : Output fC<br>1 0 : Output f8<br>1 1 : Output f32 | RW |
| CM01 | | | RW |
| CM02 | Peripheral Clock Source Stop Bit (3) | 0: Peripheral clock source not stopped in wait mode<br>1: Peripheral clock source stopped in wait mode (4) | RW |
| CM03 | XCIN-XCOUT Drive Power Select Bit (5) | 0: Low<br>1: High | RW |
| CM04 | Port XC Switch Bit | 0: I/O port<br>1: XCIN-XCOUT oscillator (6) | RW |
| CM05 | Main Clock Oscillator (XIN-XOUT) Stop Bit (3, 7) | 0: Start main clock oscillator running<br>1: Stop main clock oscillator running | RW |
| CM06 | Watchdog Timer Function Select Bit | 0: Watchdog timer interrupt<br>1: Reset (8) | RW |
| — (b7) | Reserved | Should be written with 0 | RW |

Notes:
1. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. When the PM07 bit in the PM0 register is set to 0 (BCLK output), bits CM01 and CM00 should be set to 00b. In memory expansion mode, when the PM07 bit is set to 1 (the pin function is selected using bits CM01 and CM00) and bits CM01 and CM00 are set to 00b, a low is output from the P5_3 pin (This pin does not function as Port P5_3).
3. When the PM21 bit in the PM2 register is 1 (clock change disabled), neither the CM02 bit nor the CM05 bit changes, even if written to.
4. fC32 and f2n whose clock source is the main clock do not stop.
5. When entering stop mode, the CM03 bit becomes 1.
6. To set the CM04 bit to 1 (XCIN-XCOUT oscillator), bits PD8_7 and PD8_6 in the PD8 register should be set to 0 (input) and the PU25 bit in the PUR2 register should be set to 0 (pull-up resistor disabled).
7. This bit stops the main clock when entering low power mode. It cannot detect whether or not the main clock oscillator stops. When the CM05 bit is set to 1, the clock applied to the XOUT pin becomes high. Since the on-chip feedback resistor remains connected, the XIN pin is connected to the XOUT pin via the feedback resistor.
8. Once this bit is set to 1, it cannot be set to 0 by a program.

**Figure 8.3 CM0 Register**

System Clock Control Register 1 (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | | | 0 | 0 | 0 | 0 | |

| Symbol | Address | Reset Value |
|---|---|---|
| CM1 | 40047h | 0010 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CM10 | PLL Clock Oscillator Stop Bit (2, 3) | 0: Start PLL clock oscillator running<br>1: Stop PLL clock oscillator running | RW |
| — (b4-b1) | Reserved | Should be written with 0 | RW |
| CM15 | XIN-XOUT Drive Power Select Bit | b6 b5<br>0 0 : Low<br>0 1 : High<br>1 0 : Super low (4)<br>1 1 : Do not use this combination | RW |
| CM16 | | | RW |
| — (b7) | Reserved | Should be written with 0 | RW |

Notes:
1. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. When the BCS bit in the CCR register is set to 0 (PLL clock selected as base clock source), the CM10 bit cannot be set to 1.
3. The PM21 bit in the PM2 register is set to 1 (clock change disabled), the CM10 bit is not changed by a write access.
4. The oscillator frequency should be 8 MHz or less to select super low mode.

**Figure 8.4 CM1 Register**

## Oscillator Stop Detection Register (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | | | 0 | |

Symbol: CM2 | Address: 4004Dh | Reset Value: 0000 0000b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CM20 | Oscillator Stop Detection Enable Bit (2, 3) | 0: Disabled<br>1: Enabled | RW |
| —<br>(b1) | Reserved | Should be written with 0 | RW |
| CM22 | Oscillator Stop Detection Flag (4) | 0: Main clock oscillator never stopped<br>1: Main clock oscillator stop detected | RW |
| CM23 | Main Clock Monitor Flag (5) | 0: Main clock oscillator active<br>1: Main clock oscillator stopped | RO |
| —<br>(b7-b4) | Reserved | Should be written with 0 | RW |

Notes:
1. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. This bit should be set to 0 (oscillator stop detection disabled) when f256 is selected as base clock source in low speed mode.
3. When the PM21 bit in the PM2 register is set to 1 (clock change disabled), the CM02 bit is not changed by a write access.
4. When a main clock oscillator stop is detected, this bit becomes 1. It can be set to 0 but not to 1. If it is set to 0 while the main clock oscillator is stopped, it can be changed to 1 after the next main clock oscillator stop is detected.
5. The main clock state should be determined by several read accesses of this bit after an oscillator stop detection interrupt is generated.

**Figure 8.5 CM2 Register**

## Low-Speed Mode Clock Control Register (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | | |

Symbol: CM3 | Address: 4005Ah | Reset Value: XXXX XX00b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CM30 | Low Speed Mode Base Clock Select Bit | b1 b0<br>0 0 : fC<br>0 1 : f256 (main clock divided by 256)<br>1 0 : fOCO4 (on-chip oscillator divided by 4) (2)<br>1 1 : Do not use this combination | RW |
| CM31 | | | RW |
| —<br>(b7-b2) | No register bits; should be written with 0 and read as undefined value | | — |

Notes:
1. Set the PRC27 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. The on-chip oscillator clock starts running when the CM31 bit is set to 1.

**Figure 8.6 CM3 Register**

## Count Source Prescaler Register

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | | | | |

Symbol: TCSPR
Address: 035Fh
Reset Value: 0000 0000b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CNT0 | Divide Ratio Select Bit (1) | f2n is either the main clock or peripheral clock source divided by 2n. If n = 0, the clock is not divided (n = setting value) | RW |
| CNT1 | | | RW |
| CNT2 | | | RW |
| CNT3 | | | RW |
| — (b6-b4) | Reserved | Should be written with 0 | RW |
| CST | Divider Operation Enable Bit | 0: Stop divider operation<br>1: Start divider operation | RW |

Note:
1. The CST bit should be set to 0 to rewrite bits CNT3 to CNT0.

**Figure 8.7 TCSPR Register**

## Clock Prescaler Reset Register

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

Symbol: CPSRF
Address: 0341h
Reset Value: 0XXX XXXXb

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b6-b0) | No register bits; should be written with 0 and read as undefined value | | — |
| CPSR | Clock Prescaler Reset Bit | When this bit is set to 1, fC divide-by-32 divider is initialized. The bit is read as 0 | RW |

**Figure 8.8 CPSRF Register**

## Processor Mode Register 2 (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | | 0 | | 0 | 0 | | 0 |

| Symbol | Address | Reset Value |
|---|---|---|
| PM2 | 40053h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b0) | Reserved | Should be written with 0 | RW |
| PM21 | System Clock Protect Bit (2, 3) | 0: Protect the clock by the PRCR register<br>1: Clock change disabled | RW |
| — (b3-b2) | Reserved | Should be written with 0 | RW |
| PM24 | NMI Enable Bit (2) | 0: NMI disabled (4)<br>1: NMI enabled | RW |
| — (b5) | Reserved | Should be written with 0 | RW |
| PM26 | f2n Clock Source Select Bit | 0: Peripheral clock source<br>1: Main clock | RW |
| — (b7) | Reserved | Should be written with 0 | RW |

Notes:
1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. Once this bit is set to 1, it cannot be set to 0 by a program.
3. When the PM21 bit is set to 1, the following bits are not changed by a write access:
   CM02 bit in the CM0 register (the peripheral clock source state in wait mode)
   CM05 bit in the CM0 register (start main clock oscillator running)
   CM10 bit in the CM1 register (start PLL clock oscillator running)
   CM20 bit in the CM2 register (oscillator stop detection enabled/disabled)
4. When the PM24 bit is set to 0 (NMI disabled), the forced cutoff of the three-phase motor control timers is not available.

**Figure 8.9 PM2 Register**

## Processor Mode Register 3 (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | | | 0 | 0 | 0 | 0 | 0 |

Symbol: PM3
Address: 40048h
Reset Value: 0000 0000b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b4-b0) | Reserved | Should be written with 0 | RW |
| PM35 | Peripheral Clock Source Divide Ratio Select Bit (2) | b6 b5<br>0 0 : Divide-by-8<br>0 1 : Divide-by-6<br>1 0 : Divide-by-4<br>1 1 : Divide-by-2 | RW |
| PM36 | | | RW |
| — (b7) | Reserved | Should be written with 0 | RW |

Notes:
1. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. The divide ratio should be selected so that the peripheral clock source has a frequency specified in the electrical characteristics or below.

**Figure 8.10 PM3 Register**

The following sections illustrate clocks generated in clock generators.

### 8.1.1 Main Clock

The main clock is generated by the main clock oscillator. This clock can be a clock source for the PLL reference clock or the peripheral clock. It also functions as an operating clock for the CAN module.

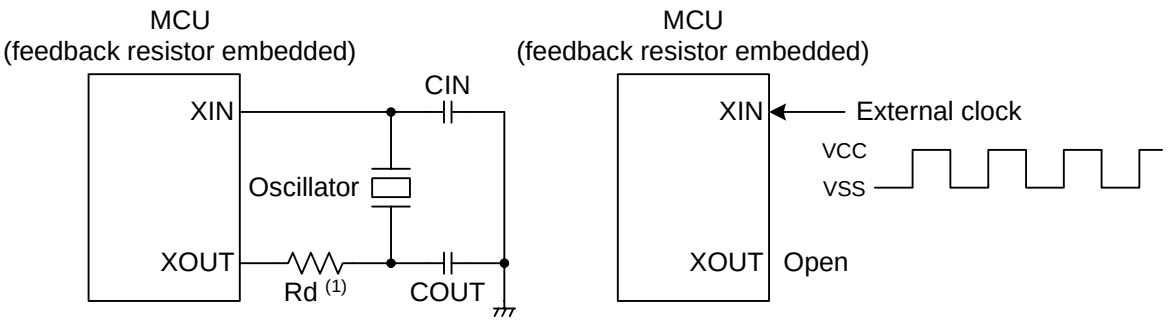
The main clock oscillator is configured with two pins, XIN and XOUT, connected by an oscillator or resonator. The circuit has an on-chip feedback resistor which is separated from the oscillator in stop mode to save power consumption. An external clock can be applied to the XIN pin in this circuit. Figure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

Circuit constants may vary depending on each oscillator. They should be applied by each manufacturer's recommendations.

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

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

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




Note:
1. A damping resistor should be placed if required. Resistance values may vary according to oscillator setting. Values recommended by the manufacturer should be set. A feedback resistor should be placed between XIN and XOUT if the manufacturer recommends placing a resistor externally.

**Figure 8.11 Main Clock Circuit Connection**

### 8.1.2 Sub Clock (fC)

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

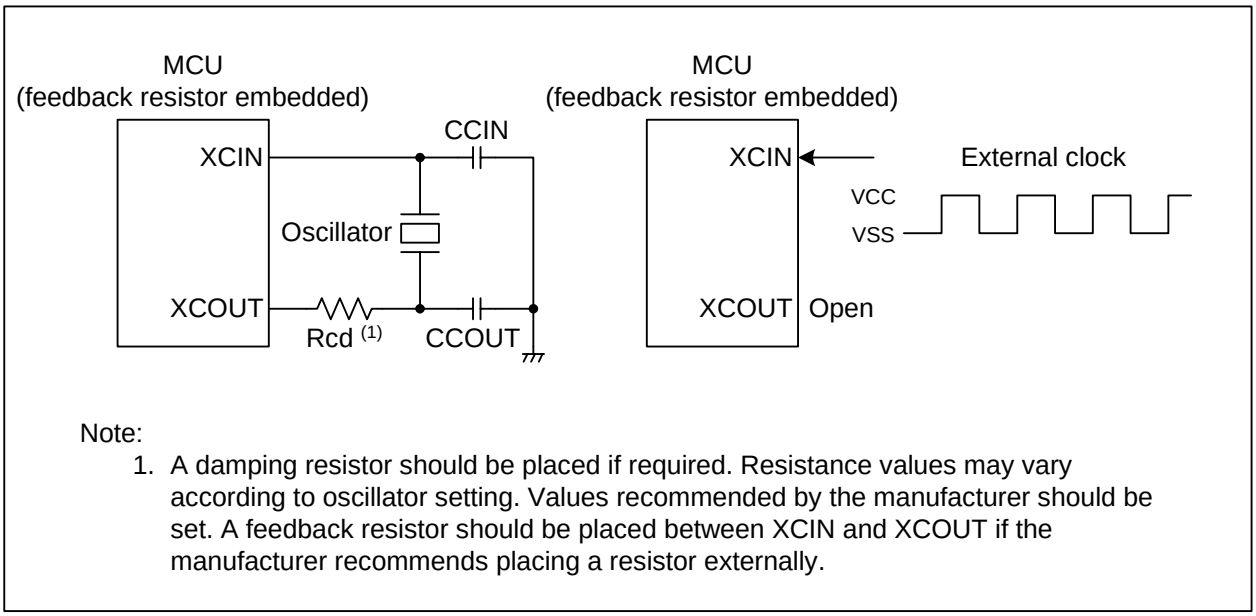
The sub clock oscillator is configured with pins XCIN and XCOUT connected by a crystal oscillator. The circuit has a on-chip feedback resistor which is separated from the oscillator in stop mode to save power consumption. An external clock can be applied to the XCIN pin. Figure 8.12 shows an example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

After a reset, the sub clock oscillator is stopped. The feedback resistor is separated from the oscillator. To resume running, first set bits PD8_6 and PD8_7 in the PD8 register to 0 (input mode), and the PU25 bit in the PUR2 register to 0 (pull-up resistor unused). Then, set the CM04 bit in the CM0 register to 1 (XCIN-XCOUT oscillator).

To input an external clock to the XCIN pin, bits PD8_7 and PU25 should be set to 0, then the CM04 bit should be set to 1. The clock applied to the XCIN pin becomes a clock source for the sub clock.

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

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




Note:
1. A damping resistor should be placed if required. Resistance values may vary according to oscillator setting. Values recommended by the manufacturer should be set. A feedback resistor should be placed between XCIN and XCOUT if the manufacturer recommends placing a resistor externally.

**Figure 8.12 Sub Clock Circuit Connection**

### 8.1.3 PLL Clock

The PLL clock is generated by the PLL frequency synthesizer based on the main clock. This clock can be a clock source for any clock including the CPU clock and the peripheral clock.
Figure 8.13 shows a block diagram of the PLL frequency synthesizer. Figure 8.14 and Figure 8.15 show registers PLC0 and PLC1, respectively.

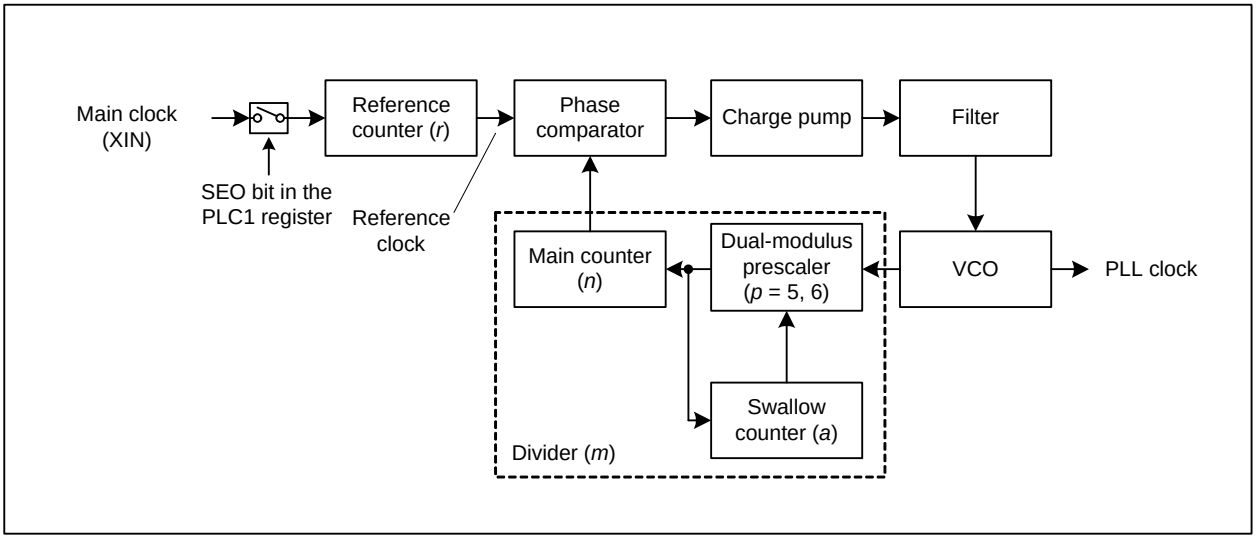



**Figure 8.13 PLL Frequency Synthesizer Block Diagram**

PLL Control Register 0 (1, 2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| PLC0 | 40020h | 0000 0001b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| MCV0 | Main Counter Divide Ratio Setting Bit | Set the bits to *n* - 1 (*n* = divide ratio of the main counter) | RW |
| MCV1 | | | RW |
| MCV2 | | | RW |
| MCV3 | | | RW |
| MCV4 | | | RW |
| SCV0 | Swallow Counter Divide Ratio Setting Bit | The divide ratio of dual-modulus prescaler is 6 (in *a* out of *n* times) or 5 (in other cases) (*a* = setting value) | RW |
| SCV1 | | | RW |
| SCV2 | | | RW |

Notes:
1. Set the PRC2 bit in the PRCR register to 1 (write enabled) just before rewriting this register. Any interrupt or DMA transfer should not be generated between these two instructions.
2. This register can be rewritten only once after a reset.

**Figure 8.14 PLC0 Register**

PLL Control Register 1 (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | | | | | |

Symbol: PLC1 | Address: 40021h | Reset Value: 0001 1111b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| RCV0 | Reference Counter Divide Ratio Setting Bit | Set the bits to r - 1 (r = divide ratio of the main clock) | RW |
| RCV1 | | | RW |
| RCV2 | | | RW |
| RCV3 | | | RW |
| SEO | Self-Oscillation Mode Setting Bit | 0: PLL mode 1: Self-oscillation mode | RW |
| — (b7-b5) | Reserved | Should be written with 0 | RW |

Note:
1. Set the PRC2 bit in the PRCR register to 1 (write enabled) just before rewriting this register. Any interrupt or DMA transfer should not be generated between these two instructions.

**Figure 8.15 PLC1 Register**

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

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

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

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

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

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

Each register should be set to meet the following conditions:

-The reference clock, which is the main clock divided by *r*, should be within 2 to 4 MHz

-The divide ratio *m* is $25 \le m \le 100$

For the setting of registers PLC1 and PLC0, Table 8.2 below should be applied. The waiting time of $t_{LOCK(PLL)}$ is required after changing the setting until the PLL clock oscillation has stabilized while the main clock oscillation is stable.

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

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

Note:
1. The setting of registers PLC1 and PLC0 should be done according to the list above.

### 8.1.4 On-chip Oscillator Clock

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

The on-chip oscillator clock is stopped after a reset. It starts running if the CM31 bit in the CM3 register is set to 1. The clock should be switched after the on-chip oscillator clock has stabilized.

## 8.2 Oscillator Stop Detection

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

- The CM22 bit: main clock oscillator stop detected
- The CM23 bit: main clock oscillator stopped
  (Refer to Figure 8.17 “State Transition (when the sub clock is used)”)

### 8.2.1 How to Use Oscillator Stop Detection

The oscillator stop detection interrupt shares vectors with the watchdog timer interrupt, and the low voltage detection interrupt. When using these interrupts simultaneously, read the CM22 bit with an interrupt handler to determine if an oscillator stop detection interrupt request has been generated.
When the main clock oscillator resumes running after an oscillator stop is detected, the PLL clock frequency may temporarily exceed the preset value before the PLL frequency synthesizer oscillation stabilizes. As soon as an oscillator stop is detected, the divide ratios of the base clock and peripheral clock source should be increased by a program. The respective divide ratio can be set by bits BCD1 and BCD0 in the CCR register and bits PM36 and PM35 in the PM3 register.
In low speed mode, when the main clock oscillator stops running, the oscillator stop detection interrupt request is generated, if the CM20 bit is set to 1 (oscillator stop detection enabled). The CPU clock remains running with a low speed clock source. Note that if the base clock is f256, which is the main clock divided by 256, the oscillator stop detection is disabled.
The oscillator stop detection is provided to handle main clock stop caused by external sources. To stop the main clock oscillator by a program, i.e., to enter stop mode or to set the CM05 bit to 1 (stop main clock oscillator running), the CM20 bit in the CM2 register should be set to 0 (oscillator stop detection disabled). To enter wait mode, this bit should be also set to 0.
The oscillator stop detection functions depending on the voltage of a capacitor which is being changed. More concrete terms, this function detects that the oscillator is stopped when the main clock goes lower than approximately 500 kHz. Note that if the CM22 bit is set to 0 by a program in an interrupt handler while the frequency is around 500 kHz, a stack overflow may occur caused by multiple interrupt requests.

## 8.3 Base Clock

Base clock is a reference clock for the CPU clock and peripheral bus clock. The base clock after a reset is the PLL clock divided by 6.
The base clock source is selected between the PLL clock and the low speed clocks which contains the sub clock (fC), on-chip oscillator clock divided by 4 (fOCO4), and main clock divided by 256 (f256).
If the PLL clock is selected, it is divided by a factor from 2, 3, 4, and 6 to become the base clock. If a low speed clock is selected, the clock itself can be the base clock.
The base clock source is set using the BCS bit in the CCR register and the divide ratio for the PLL clock is set using bits BCD1 and BCD0. Bits CM31 and CM30 in the CM3 register selects a low speed clock.

## 8.4 CPU Clock and Peripheral Bus Clock

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

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

In memory expansion mode or microprocessor mode, the peripheral bus clock as BCLK to be a reference clock for external timing generation is available as an output clock at the BCLK pin. Refer to 8.6 “Clock Output Function” for details.

When the CPU becomes out of control, to prevent the CPU clock whose clock source is the PLL clock from stopping, the CM05 bit in the CM0 register should be set to 0 (start main clock oscillator running) and the BCS bit in the CCR register should be set to 0 (PLL clock selected as base clock source). Then the following procedures should be performed.

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

## 8.5 Peripheral Clock

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

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

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

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

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

f8 and f32 are available as output clocks at the CLKOUT pin. Refer to 8.6 “Clock Output Function” for details.

### (2) fAD

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

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

### (3) fC32

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

## 8.6 Clock Output Function

fC, f8, and f32 are available to be output from the CLKOUT pin.
In memory expansion mode or microprocessor mode, the BCLK, that is, the peripheral bus clock which is the base clock divided by a factor from 2 to 4 is also available to be output from the BCLK pin.
Table 8.3 and Table 8.4 list the CLKOUT pin functions in single-chip mode, and memory expansion mode or microprocessor mode, respectively.

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

| PM0 Register (1) | CM0 Register (2) | | CLKOUT Pin Function |
|---|---|---|---|
| PM07 | CM01 | CM00 | |
| 0 or 1 | 0 | 0 | I/O port P5_3 |
| 1 | 0 | 1 | Output fC |
| 1 | 1 | 0 | Output f8 |
| 1 | 1 | 1 | Output f32 |

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

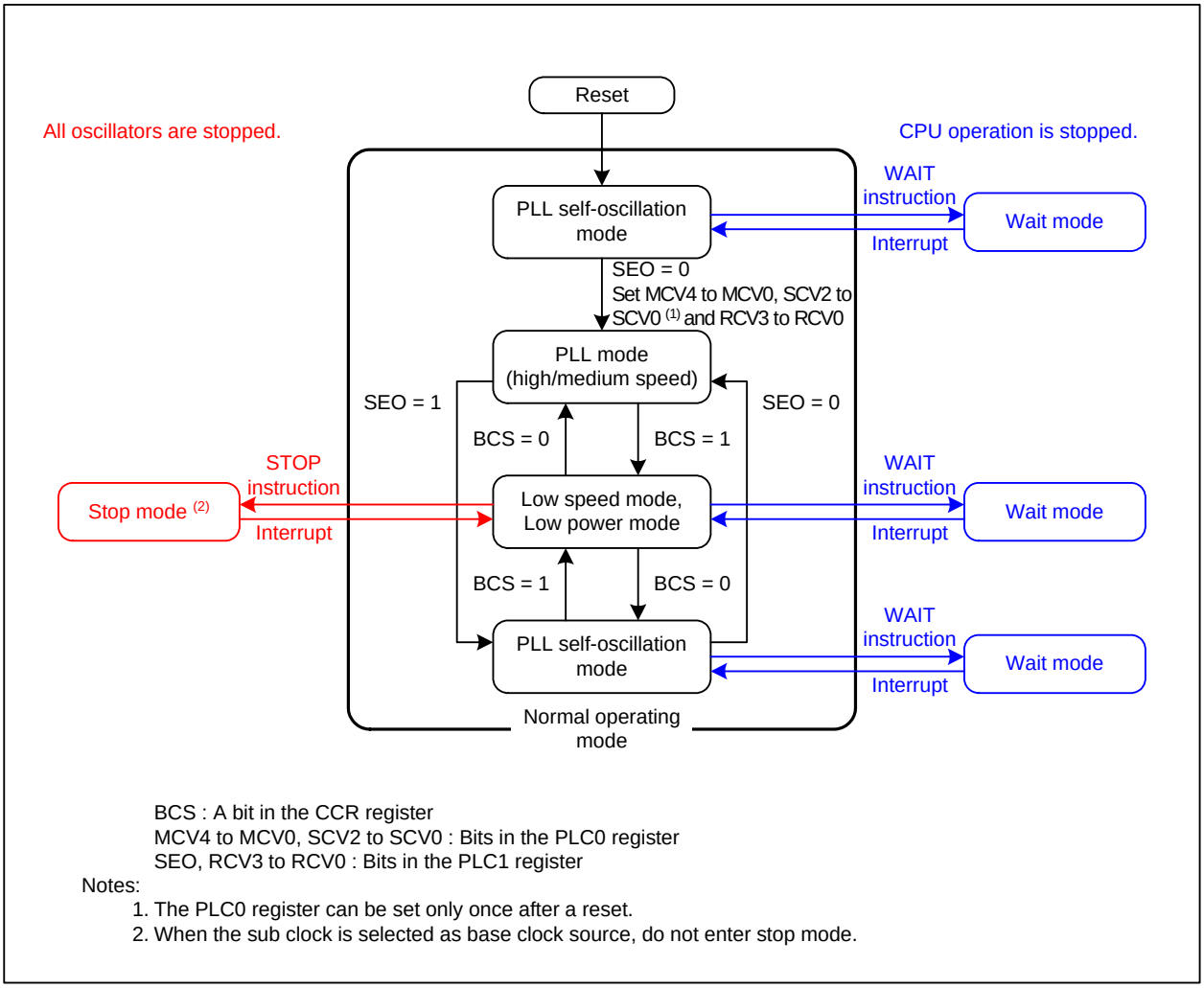
**Table 8.4 CLKOUT Pin Functions in Memory Expansion Mode or Microprocessor Mode**

| PM0 Register (1) | CM0 Register (2) | | CLKOUT Pin Function |
|---|---|---|---|
| PM07 | CM01 | CM00 | |
| 0 | 0 (3) | 0 (3) | Output BCLK |
| 1 | 0 | 0 | Output low (no function as P5_3) |
| 1 | 0 | 1 | Output fC |
| 1 | 1 | 0 | Output f8 |
| 1 | 1 | 1 | Output f32 |

Notes:
1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
3. When the PM07 bit is set to 0 (BCLK output), set bits CM01 and CM00 to 00b (I/O port P5_3).

## 8.7 Power Control

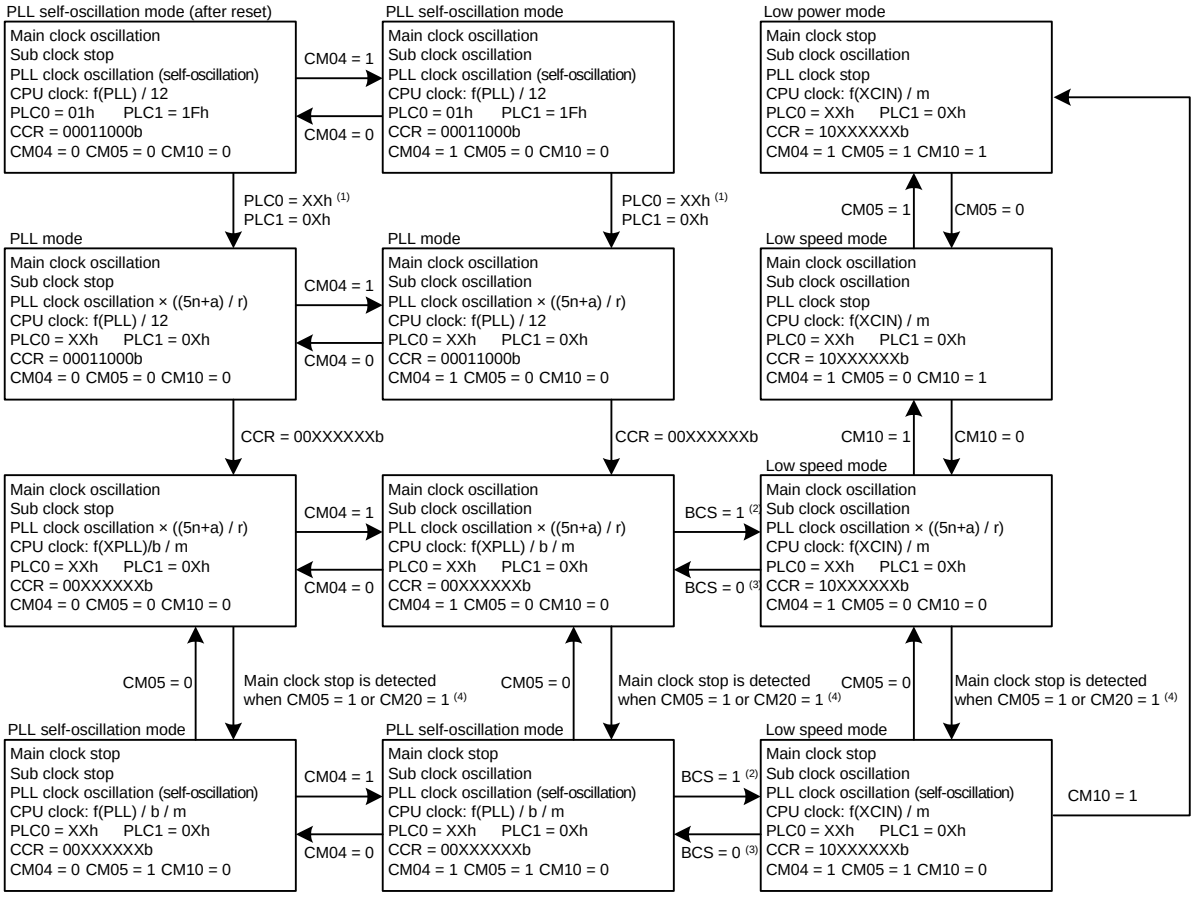
Power control contains three modes: wait mode, stop mode, and normal operating mode.
In this chapter, all mode states except wait mode and stop mode are called normal operating mode. Figure 8.16 to Figure 8.19 show block diagrams of the respective state transition: state in stop mode and wait mode, state when the sub clock is used, state when the main clock divided by 256 is used, and state when the on-chip oscillator clock is used.




BCS : A bit in the CCR register
MCV4 to MCV0, SCV2 to SCV0 : Bits in the PLC0 register
SEO, RCV3 to RCV0 : Bits in the PLC1 register

Notes:
1. The PLC0 register can be set only once after a reset.
2. When the sub clock is selected as base clock source, do not enter stop mode.

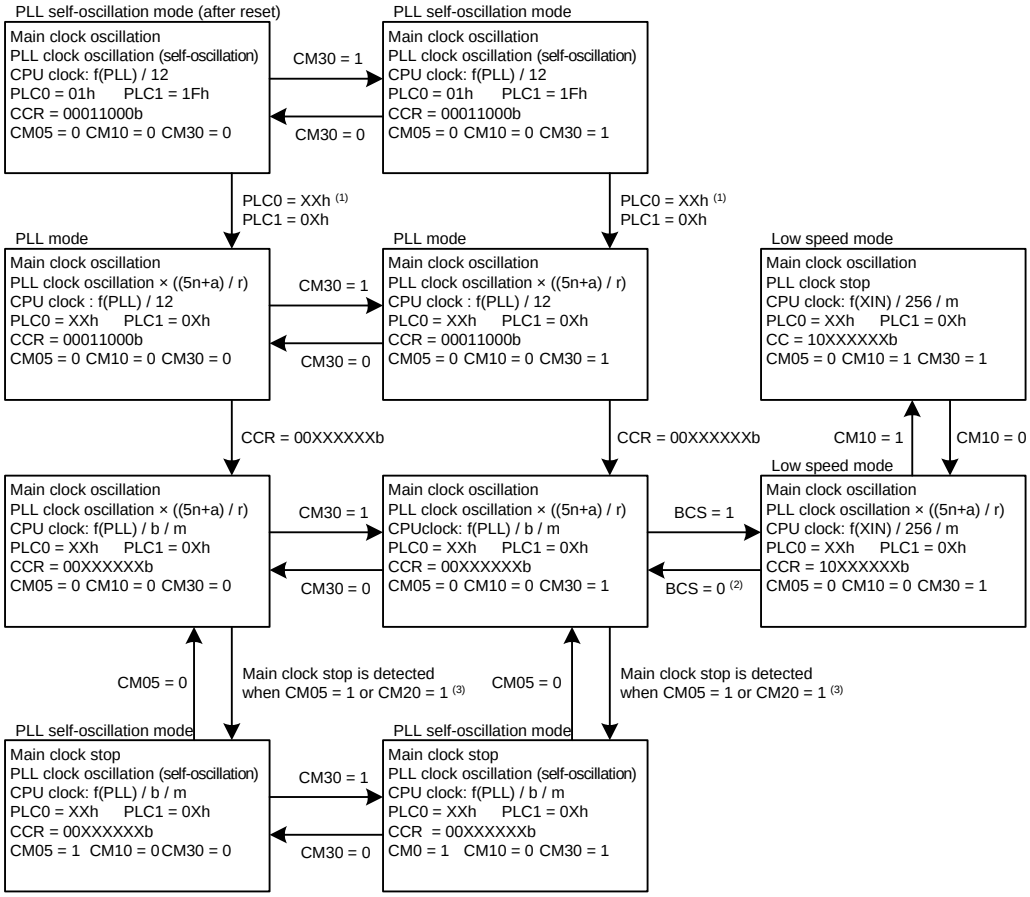
**Figure 8.16 State Transition in Stop Mode and Wait Mode**




→ : An arrow indicates a one-way transition between modes. No transition should be made unless indicated.
BCS: A bit in the CCR register
CM04 and CM05: Bits in the CM0 register
CM10: A bit in the CM1 register
CM20: A bit in the CM2 register
PLC0 = XXh: Multiplied ratio of the PLL clock should be set using bits MCV4 to MCV0 and bits SCV2 to SCV0 in the PLC0 register.
PLC1 = 0Xh: Divide ratio of the reference clock should be set using bits RCV3 to RCV0 in the PLC1 register. The PLL clock frequency should not exceed the maximum value specified in the electrical characteristics.
CCR = 00XXXXXXb: Divide ratio of each clock should be set using the CCR register. The CPU clock frequency and the peripheral bus clock frequency should not exceed the maximum value specified in the electrical characteristics, respectively.

Notes:
1. The PLC0 register can be set only once after a reset.
2. This clock should be switched after the sub clock oscillation is fully stabilized.
3. This clock should be switched after the PLL clock oscillation is fully stabilized.
4. The CM05 bit is not set to 1 when an oscillator stop is detected through the oscillation stop detector.

**Figure 8.17 State Transition (when the sub clock is used)**

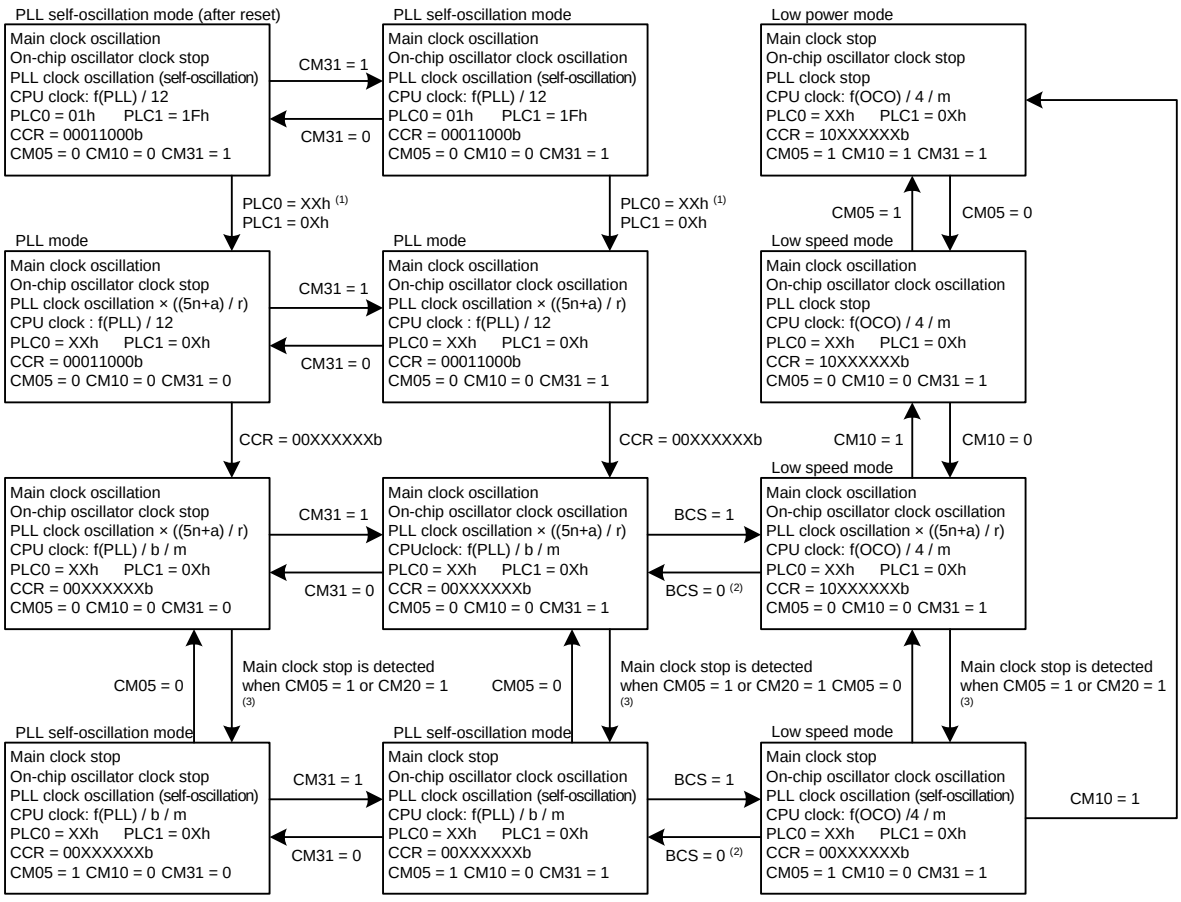



→ : An arrow indicates a one-way transition between modes. No transition should be made unless indicated.

BCS: A bit in the CCR register
CM05: A bit in the CM0 register
CM10: A bit in the CM1 register
CM20: A bit in the CM2 register
CM30: A bit in the CM3 register
PLC0 = XXh: Multiplied ratio of the PLL clock should be set using bits MCV4 to MCV0 and bits SCV2 to SCV0 in the PLC0 register.
PLC1 = 0Xh: Divide ratio of the reference clock should be set using bits RCV3 to RCV0 in the PLC1 register. The PLL clock frequency should not exceed the maximum value specified in the electrical characteristics.
CCR = 00XXXXXXb: Divide ratio of each clock should be set using the CCR register. The CPU clock frequency and the peripheral bus clock frequency should not exceed the maximum value specified in the electrical characteristics, respectively.

Notes:
1. The PLC0 register can be set only once after a reset.
2. This clock should be switched after the PLL clock oscillation is fully stabilized.
3. The CM05 bit is not set to 1 when an oscillator stop is detected through the oscillation stop detector.

**Figure 8.18 State Transition (when the main clock divided by 256 is used)**




→ : An arrow indicates a one-way transition between modes. No transition should be made unless indicated.

BCS: A bit in the CCR register
CM05: A bit in the CM0 register
CM10: A bit in the CM1 register
CM20: A bit in the CM2 register
CM31: A bit in the CM3 register
PLC0 = XXh: Multiplied ratio of the PLL clock should be set using bits MCV4 to MCV0 and bits SCV2 to SCV0 in the PLC0 register.
PLC1 = 0Xh: Divide ratio of the reference clock should be set using bits RCV3 to RCV0 in the PLC1 register. The PLL clock frequency should not exceed the maximum value specified in the electrical characteristics.
CCR = 00XXXXXXb: Divide ratio of each clock should be set using the CCR register. The CPU clock frequency and the peripheral bus clock frequency should not exceed the maximum value specified in the electrical characteristics, respectively.

Notes:
1. The PLC0 register can be set only once after a reset.
2. This clock should be switched after the PLL clock oscillation is fully stabilized.
3. The CM05 bit is not set to 1 when an oscillator stop is detected through the oscillation stop detector.

**Figure 8.19 State Transition (when the on-chip oscillator clock is used)**

### 8.7.1 Normal Operating Mode

The normal operating mode is classified into five modes as shown below.
In normal operating mode, the CPU clock and peripheral clock are provided to operate the CPU and peripheral functions. Power consumption is controlled by the CPU clock frequency. The higher the CPU clock frequency is, the more processing power increases. The lower the CPU clock frequency is, the less power consumption is required. Power consumption can be reduced further by stopping oscillators that are not being used.

(1) PLL Mode (high speed mode)

This mode enables the CPU to operate at the maximum operating frequency. The PLL clock divided by 2 becomes the base clock. The base clock frequency should be identical to that of the CPU clock. The PLL clock or the main clock is selected as the peripheral clock source. When the sub clock or the on-chip oscillator clock is provided, fC32 can be used as a count source for timers A and B.

(2) PLL Mode (medium speed mode)

The PLL clock divided by a factor from 2, 3, 4, and 6 becomes the base clock and the base clock divided by a factor from 1 to 4 becomes the CPU clock. The PLL clock or the main clock is selected as the peripheral clock source. When the sub clock or the on-chip oscillator clock is provided, fC32 can be used as a count source for timers A and B.

(3) Low Speed Mode

A low speed clock becomes the base clock and the base clock divided by a factor from 1 to 4 becomes the CPU clock. The PLL clock or the main clock is selected as the peripheral clock source. fC32 can be used as a count source for timers A and B when the sub clock or the on-chip oscillator clock is provided.

(4) Low Power Mode

In this mode, the main clock oscillator and the PLL frequency synthesizer in low speed mode are stopped. The sub clock or the on-chip oscillator clock divided by 4 becomes the base clock and the base clock divided by a factor from 1 to 4 becomes the CPU clock. fC32, which is the only peripheral clock available, can be used as a count source for timers A and B. By setting the MRS bit in the VRCR register to 1 (main VDC stopped), this mode consumes even less power than the modes above.

(5) PLL Self-oscillation Mode

In this mode, the main clock as reference clock source for the PLL clock is stopped. The PLL frequency synthesizer self-oscillates at its own frequency. The PLL clock divided by a factor from 2, 3, 4, and 6 becomes the base clock and the base clock divided by a factor from 1 to 4 becomes the CPU clock. The PLL clock or the main clock is selected as the peripheral clock source. When the sub clock or the on-chip oscillator clock is provided, fC32 can be used as a count source for timers A and B.

Before switching the CPU clock to another, that clock should be stabilized. In particular, the sub clock oscillator may require more time to stabilize (1), therefore, certain waiting time to switch should be taken by a program immediately after turning the MCU ON or exiting stop mode.

Note:
1. Contact the oscillator manufacturer for oscillator stabilization time.

### 8.7.2 Wait Mode

In wait mode, due to the base clock stop, the CPU clock and peripheral bus clock stop running as well. The CPU and watchdog timer, operated by the CPU clock, also stop.Since the main clock, sub clock, PLL clock, and on-chip oscillator clock continue running, peripheral functions using these clocks also continue operating.

#### 8.7.2.1 Peripheral Clock Source Stop Function

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

#### 8.7.2.2 Entering Wait Mode

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

- Initial setting
  Set the interrupt priority level to 7 for resuming (bits RLVL2 to RLVL0 in registers RIPL1 and RIPL2). Then set each interrupt request level.
- Before executing WAIT instruction
  (1) Set the I flag to 0
  (2) Set the interrupt request level for the interrupt to exit wait mode
  (3) Set the interrupt request levels other than number (2) above to 0
  (4) Set the IPL (processor interrupt priority level) in the flag register (FLG)
  (5) Set the interrupt priority level for resuming to the same level as the IPL
  Interrupt request level for the interrupt to exit wait mode > IPL = Interrupt priority level for resuming
  (6) Enter either PLL self-oscillation mode, low speed mode, or low power mode
  (7) Set the I flag to 1
  (8) Execute the WAIT instruction
- After exiting wait mode
  Set the interrupt priority level for resuming to 7 immediately after exiting wait mode.

### 8.7.2.3 Pin State in Wait Mode

Table 8.5 lists pin state in wait mode.

**Table 8.5 Pin State in Wait Mode**

| Pin | | Memory Expansion Mode/ Microprocessor Mode | Single-Chip Mode |
|---|---|---|---|
| Address bus, Data bus, $\overline{CS0}$ to $\overline{CS3}$, $\overline{BC0}$ to $\overline{BC3}$ | | The state immediately before entering wait mode is held | — |
| $\overline{RD}$, $\overline{WR}$, $\overline{WR0}$ to $\overline{WR3}$ | | High | — |
| $\overline{HLDA}$, BCLK | | High | — |
| ALE | | High | — |
| Ports | | The state immediately before entering wait mode is held | |
| DA0, DA1 | | The state immediately before entering wait mode is held | |
| CLKOUT | When fC is selected | The clock is output | |
| | When f8 or f32 is selected | The clock is output when the CM02 bit in the CM0 register is set to 0 (no peripheral clock source stopped in wait mode).<br>The state immediately before entering wait mode is held when the CM02 bit is set to 1 (peripheral clock source stopped in wait mode) | |

### 8.7.2.4 Exiting Wait Mode

Wait mode is exited by the hardware reset, an NMI, or peripheral interrupts designated by the software interrupt numbers 0 to 63.

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

The CM02 bit setting in the CM0 register affects the peripheral interrupts. When the CM02 bit is set to 0 (no peripheral clock source stopped in wait mode), all peripheral interrupts are available to exit wait mode. When this bit is set to 1 (peripheral clock source stopped in wait mode), peripheral functions using f1, f8, f32, f2n (when the clock source is the peripheral clock source), or fAD stop operating. Therefore, the peripheral interrupts cannot be used to exit wait mode. However, the peripheral interrupts with the software interrupt numbers 0 to 63 caused by an external clock, fC32, or f2n (when the clock source is the main clock) can be used to exit wait mode.

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

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

**Table 8.6 Interrupts to Exit Wait Mode and Usage Conditions**

| Interrupt | When the CM02 bit = 0 | When the CM02 bit = 1 |
|---|---|---|
| NMI | Available | Available |
| External interrupt | Available | Available |
| Key input interrupt | Available | Available |
| Low voltage detection interrupt | Available | Available |
| Timer A interrupt<br>Timer B interrupt | Available in any mode | Available in event counter mode, or when the count source is fC32 or f2n of which clock source is the main clock |
| Serial interface interrupt (1) | Available when the internal or external clock is used | Available when the external clock or f2n (when the clock source is the main clock) is used |
| A/D conversion interrupt | Available in single mode or single-sweep mode | Should not be used |
| Intelligent I/O interrupt | Available | Should not be used |
| $I^2C$-bus interface interrupt | Available | Should not be used |
| $I^2C$-bus line interrupt | Available | Available |
| CAN wake-up interrupt | Available | Available |

Note:
1. UART7 and UART8 are excluded.

### 8.7.3 Stop Mode

In stop mode, all oscillators and resonators stop running. That is, the CPU and peripheral functions, operated by the CPU clock and peripheral clock, also stop. This is the most power-saving mode.

#### 8.7.3.1 Entering Stop Mode

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

- Initial setting
  Set the interrupt priority level for resuming (bits RLVL2 to RLVL0 in registers RIPL1 and RIPL2) to 7. Then set each interrupt request level
- Before entering stop mode
  (1) Set the I flag to 0
  (2) Set the interrupt request level for the interrupt to exit stop mode
  (3) Set the interrupt request levels other than number (2) above to 0
  (4) Set the IPL (processor interrupt priority level) in the flag register (FLG)
  (5) Set the interrupt priority level for resuming to the same level as the IPL
  Interrupt request level for the interrupt to exit stop mode > IPL = Interrupt priority level for resuming
  (6) Change the base clock to either the main clock divided by 256 (f256) or the on-chip oscillator clock divided by 4 (fOCO4).
  (7) Set the CM20 bit in the CM2 register to 0 (oscillator stop detection disabled) when the oscillator stop detection is used.
  (8) Set the I flag to 1
  (9) Execute the STOP instruction
- After exiting stop mode
  Set the interrupt priority level for resuming to 7 immediately after exiting stop mode

### 8.7.3.2 Pin State in Stop Mode

Table 8.7 lists pin state in stop mode.

**Table 8.7 Pin State in Stop Mode**

| Pin | | Memory Expansion Mode/ Microprocessor Mode | Single-Chip Mode |
|---|---|---|---|
| Address bus, Data bus, $\overline{CS0}$ to $\overline{CS3}$, $\overline{BC0}$ to $\overline{BC3}$ | | The state immediately before entering stop mode is held | — |
| $\overline{RD}$, $\overline{WR}$, $\overline{WR0}$ to $\overline{WR3}$ | | High | — |
| $\overline{HLDA}$, BCLK | | High | — |
| ALE | | High | — |
| Ports | | The state immediately before entering stop mode is held | |
| DA0, DA1 | | The state immediately before entering stop mode is held | |
| CLKOUT | When fC is selected | High | |
| | When f8 or f32 is selected | The state immediately before entering stop mode is held | |
| XIN | | High-impedance | |
| XOUT | | High | |
| XCIN, XCOUT | | High-impedance | |

### 8.7.3.3 Exiting Stop Mode

Stop mode is exited by the hardware reset, an NMI, low voltage detection interrupt, or peripheral interrupts (refer to Table 8.8).

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

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

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

**Table 8.8 Interrupts to Exit Stop Mode and Usage Conditions**

| Interrupt | Usage Condition |
|---|---|
| NMI | |
| Low voltage detection interrupt | |
| External interrupt | |
| Key input interrupt | |
| Timer A interrupt<br>Timer B interrupt | Available when the timer counts external pulse, having its 100 Hz or less frequency, in event counter mode |
| Serial interface interrupt (1) | Available when external clock is used |
| $I^2C$-bus line interrupt | |
| CAN wake-up interrupt | |

Note:

1. UART7 and UART8 are excluded.

## 8.8 System Clock Protection

The system clock protection is a function to disable clock change when the PLL clock is selected as base clock source. This prevents the CPU clock, which is out of control, from stopping.
When the PM21 bit in the PM2 register is set to 1 (clock change disabled), the following bits cannot be written to:

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

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

(1) The PRC1 bit in the PRCR register should be set to 1 (write to the PM2 register enabled)
(2) The PM21 bit in the PM2 register should be set to 1 (clock change disabled)
(3) The PRC1 bit in the PRCR register should be set to 0 (write to the PM2 register disabled)

## 8.9 Notes on Clock Generator

### 8.9.1 Sub Clock

#### 8.9.1.1 Oscillation Parameter Matching

The constant matching of sub clock oscillator should be evaluated in both cases when the drive power is high and low.
Contact your oscillator manufacturer for details on the oscillation circuit constant matching.

### 8.9.2 Power Control

Do not switch the base clock source until the oscillation of the clock to be used has stabilized. However, this does not apply to the on-chip oscillator since the on-chip oscillator starts running immediately after the CM31 bit in the CM3 register is set to 1.
To switch the base clock source from PLL clock to a low speed clock, that is, to set the BCS bit in the CCR register to 1, use either the MOV.L or OR.L instruction.

- Program example in assembly language

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

- Program example in C language

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

#### 8.9.2.1 Stop Mode

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

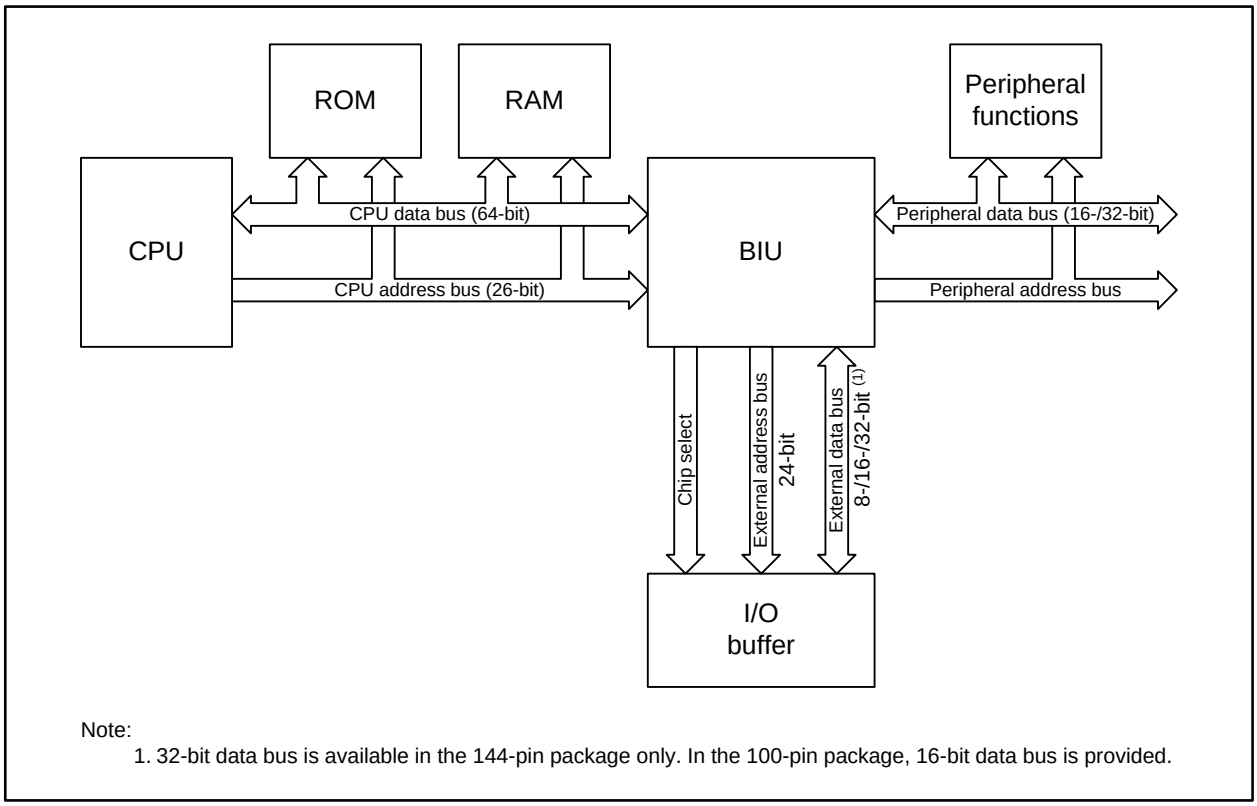
#### 8.9.2.2 Suggestions to Power Saving

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

- I/O pins:
  If inputs are floating, both transistors may be conducting. Set unassigned pins to input mode and connect each of them to VSS via a resistor, or set them to output mode and leave them open.
- A/D converter:
  When the A/D conversion is not performed, set the VCUT bit in the AD0CON1 register to 0 (VREF disconnected). To perform the A/D conversion, set the VCUT bit to 1 (VREF connected) and wait 1 µs or more for the operation.
- D/A converter:
  When the D/A conversion is not performed, set the DAiE bit in the DACON register (i = 0, 1) to 0 (output disabled) and the DAi register to 00h.
- Peripheral clock stop
  When entering wait mode, power consumption can be reduced by setting the CM02 bit in the CM0 register to 1 to stop peripheral clock source. However, the fC32 does not stop by the CM02 bit setting.

# 9. Bus

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




Note:
1. 32-bit data bus is available in the 144-pin package only. In the 100-pin package, 16-bit data bus is provided.

**Figure 9.1 Bus Block Diagram**

In memory expansion mode or microprocessor mode, some pins function as bus control pin to control the address bus and the data bus. The bus control pins are as follows: A0 to A23, D0 to D31, $\overline{CS0}$ to $\overline{CS3}$, $\overline{WR0}$/$\overline{WR}$, $\overline{BC0}$, $\overline{WR1}$/$\overline{BC1}$, $\overline{WR2}$/$\overline{BC2}$, $\overline{WR3}$/$\overline{BC3}$, $\overline{RD}$, BCLK, $\overline{HLDA}$, $\overline{HOLD}$, ALE, and $\overline{RDY}$.

## 9.1 Bus Setting

The bus setting is controlled by the two lowest bits of reset vector, the PBC register, registers EBC0 to EBC3, and CSOP0 to CSOP2.
Table 9.1 lists bus settings and their sources.

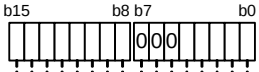
**Table 9.1 Bus Settings and Sources**

| Bus Setting | Sources |
|---|---|
| Internal SFR bus timing | PBC register |
| External bus timing | Registers EBC0 to EBC3 |
| External data bus width | PBC register, registers EBC0 to EBC3 |
| External data bus width after reset | Two lowest bits of reset vector |
| Separate bus/Multiplexed bus selection | PBC register, registers EBC0 to EBC3 |
| Pins to output chip select signals | Registers CSOP0 to CSOP2 |

## 9.2 Peripheral Bus Timing Setting

The peripheral bus of 16-/32-bit width operates at a frequency up to 32 MHz (the theoretical value and the maximum frequency of each product group are as defined by f(BCLK) in 28. “Electrical Characteristics”). The timing adjustment and bus-width conversion with the faster, 64-bit-wide CPU bus are controlled in the bus interface unit (BIU).
Figure 9.2 shows the PBC register which determines the peripheral bus timing.

Peripheral Bus Control Register (1, 2)




| Symbol | Address | Reset Value |
|---|---|---|
| PBC | 001Fh-001Eh | 0504h |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PRD0 | Read Timing Setting Bit | Select from the three options below according to the peripheral bus clock setting (bits PCD1 and PCD0 in the CCR register).<br>When bits PCD1 and PCD0 are set to:<br>1. 01b : 00100b<br>2. 10b : 01101b<br>3. 11b : 01111b | RW |
| PRD1 | | | |
| PRD2 | | | |
| PRD3 | | | |
| PRD4 | | | |
| — (b7-b5) | Reserved | Should be written with 0 | RW |
| PWR0 | Write Timing Setting Bit | Select from the three options below according to the peripheral bus clock setting (bits PCD1 and PCD0 in the CCR register).<br>When bits PCD1 and PCD0 are set to:<br>1. 01b : 00101b<br>2. 10b : 01010b<br>3. 11b : 01111b | RW |
| PWR1 | | | |
| PWR2 | | | |
| PWR3 | | | |
| PWR4 | | | |
| EXMPX | External Bus Format Select Bit (3) | 0: Separate bus in some spaces<br>1: Multiplexed bus in all spaces | RW |
| EXBW0 | External Bus Maximum Width Setting Bit (4) | b15b14<br>0 0 : 8-bit width<br>0 1 : 16-bit width<br>1 0 : 32-bit width (5)<br>1 1 : Do not use this combination | RW |
| EXBW1 | | | |

Notes:
1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. This register should be set only once after a reset. It should not be rewritten after the CCR register is set.
3. If this bit is set to 1 when the all MPX bits in registers EBC0 to EBC3 are set to 1, ports P0, P1, and P4_0 to P4_3 can be used as programmable I/O ports.
4. This bit should be the maximum bus width set in bits BW1 and BW0 in registers EBC0 to EBC3. The functions of ports P1, P12, and P13 vary with this bit setting.
5. This bit setting is applicable only in the 144-pin package.

**Figure 9.2 PBC Register**

## 9.3 External Bus Setting

External bus of 8-/16-/32-bit width operates at a frequency up to 32 MHz (the theoretical value and the maximum frequency of each product group are as defined by f(BCLK) in 28. “Electrical Characteristics”). The timing adjustment and bus-width conversion with the faster, 64-bit-wide CPU bus are controlled in the bus interface unit (BIU).

### 9.3.1 External Address Space Setting

In the R32C/100 Series, the CPU contains 26 address buses (A0 to A25) in the MCU. Since A26 to A31 are sign-extended of A25, it has 64 MB of accessible space in total in the addresses 00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh.

As address bus for external output, up to 24 buses (A0 to A23) are available. Decoded A18 to A25 function as 4 chip select signals ($\overline{CS3}$ to $\overline{CS0}$). If 16 MB space is assigned to every chip select signal, up to 63.5 MB is available for external address space. When the processing mode is changed from single-chip mode to memory expansion mode, the address bus status is undefined until an external space is accessed.

Chip select signals $\overline{CS3}$ to $\overline{CS0}$ share pins with A20 to A23, respectively. Other combinations of signal and output port are also available as follows: signals $\overline{CS0}$ to $\overline{CS3}$ with ports P11_0 to P11_3 and signals $\overline{CS1}$ to $\overline{CS3}$ with ports P5_4, P5_6, and P5_7.

In microprocessor mode, the $\overline{CS0}$ signal is output from port P4_7 after a reset. The maximum space per chip select signal is 8 MB since A23 is not available. Signals $\overline{CS1}$ to $\overline{CS3}$ are output only when being set.

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

A chip select signal to be used and an output pin are selected in registers CSOP0 to CSOP2. The space for each chip select signal is selected in registers CB01, CB12, and CB23.

Figure 9.4 to Figure 9.6 show registers CSOP0 to CSOP2. Figure 9.7, Figure 9.8, and Figure 9.9 show respectively registers CB01, CB12, and CB23. Figure 9.10 and Figure 9.11 show the chip select space.

A chip select signal should not be set for more than two output pins in registers CSOP0 to CSOP2.

Registers CB01, CB12, and CB23 should be set to meet the conditions below:

- In memory expansion mode

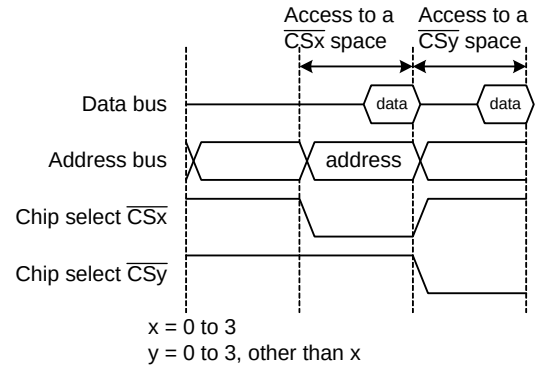
$$0080000\text{h} < (\text{CB23} \times 2^{18}) < (\text{CB12} \times 2^{18}) < (\text{CB01} \times 2^{18}) \leq 3\text{DC0000h}$$

- In microprocessor mode

$$0080000\text{h} < (\text{CB23} \times 2^{18}) < (\text{CB12} \times 2^{18}) < (\text{CB01} \times 2^{18}) \leq 3\text{FC0000h}$$

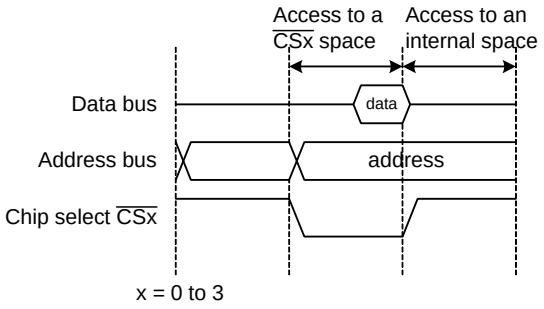
Pattern 1. Address bus: changed,
Chip select signal: changed,
at the first cycle after accessing an external space

When a $\overline{CSy}$ space is accessed in the first cycle after the access to a $\overline{CSx}$ space, both the address bus and the chip select signal are changed.



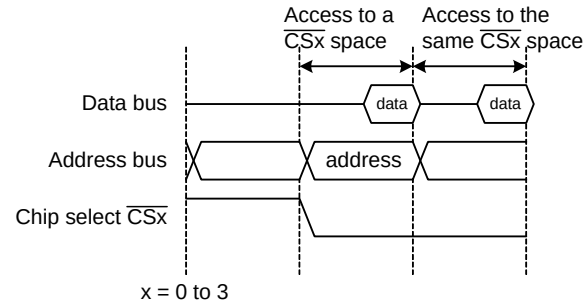

Pattern 2. Address bus: not changed,
Chip select signal: changed,
at the first cycle after accessing an external space

When an internal space is accessed in the first cycle after the access to a $\overline{CSx}$ space, only the chip select signal is changed.



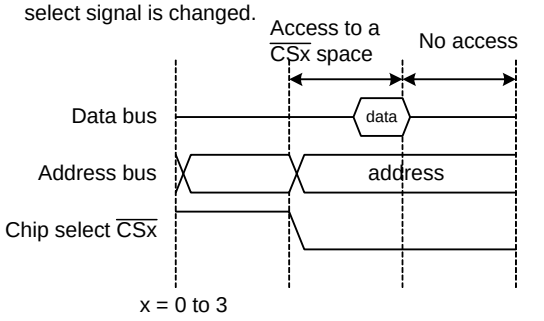

Pattern 3. Address bus: changed,
Chip select signal: not changed,
at the first cycle after accessing an external space

When the same $\overline{CSx}$ space is accessed in the first cycle after the access to a $\overline{CSx}$ space, only the address bus is changed.




Pattern 4. Address bus: not changed,
Chip select signal: not changed,
at the first cycle after accessing an external space

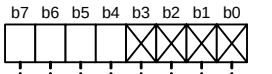
When no space is accessed in the first cycle after the access to a $\overline{CSx}$ space, (and no instruction prefetching is generated), neither the address bus nor the chip select signal is changed.




Note:
1. The patters above show combinations of an address bus and a chip select signal in sequential two cycles. A chip select signal may be extended to two-bus cycles or more according to the combination.

**Figure 9.3 Address Bus and Chip Select Signal Output Patterns (in Separate Bus Format)**

## Chip Select Output Pin Setting Register 0 (1, 2)



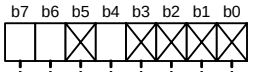

| Symbol | Address | Reset Value |
|---|---|---|
| CSOP0 | 40054h | 1000 XXXXb |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b3-b0) | No register bits; should be written with 0 and read as undefined value | | — |
| P4_4B | P4_4 Bus Function Setting Bit | 0: Output A20 from P4_4<br>1: Output $\overline{CS3}$ from P4_4 | RW |
| P4_5B | P4_5 Bus Function Setting Bit | 0: Output A21 from P4_5<br>1: Output $\overline{CS2}$ from P4_5 | RW |
| P4_6B | P4_6 Bus Function Setting Bit | 0: Output A22 from P4_6<br>1: Output $\overline{CS1}$ from P4_6 | RW |
| P4_7B | P4_7 Bus Function Setting Bit | 0: Output A23 from P4_7 (3)<br>1: Output $\overline{CS0}$ from P4_7 | RW |

Notes:
1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. This register should not be set in single-chip mode.
3. The P4_7B bit should not be set to 0 when starting an operation in microprocessor mode.

**Figure 9.4 CSOP0 Register**

## Chip Select Output Pin Setting Register 1 (1, 2)



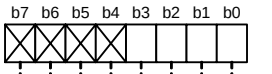

| Symbol | Address | Reset Value |
|---|---|---|
| CSOP1 | 40055h | 01X0 XXXXb |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b3-b0) | No register bits; should be written with 0 and read as undefined value | | — |
| P5_4B | P5_4 Bus Function Setting Bit | 0: Output $\overline{HLDA}$ from P5_4<br>1: Output $\overline{CS1}$ from P5_4 | RW |
| — (b5) | No register bit; should be written with 0 and read as undefined value | | — |
| P5_6B | P5_6 Bus Function Setting Bit | 0: Output ALE from P5_6<br>1: Output $\overline{CS2}$ from P5_6 | RW |
| P5_7B | P5_7 Bus Function Setting Bit | 0: $\overline{RDY}$ input pin<br>1: Output $\overline{CS3}$ from P5_7 | RW |

Notes:
1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. This register should not be set in single-chip mode.

**Figure 9.5 CSOP1 Register**

## Chip Select Output Pin Setting Register 2 (1, 2)



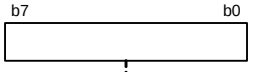

| Symbol | Address | Reset Value |
|---|---|---|
| CSOP2 | 40056h | XXXX 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| P11_0B | P11_0 Bus Function Setting Bit | 0: Use P11_0 for a peripheral function<br>1: Output $\overline{CS0}$ from P11_0 | RW |
| P11_1B | P11_1 Bus Function Setting Bit | 0: Use P11_1 for a peripheral function<br>1: Output $\overline{CS1}$ from P11_1 | RW |
| P11_2B | P11_2 Bus Function Setting Bit | 0: Use P11_2 for a peripheral function<br>1: Output $\overline{CS2}$ from P11_2 | RW |
| P11_3B | P11_3 Bus Function Setting Bit | 0: Use P11_3 for a peripheral function<br>1: Output $\overline{CS3}$ or $\overline{WR2}$ from P11_3 (3) | RW |
| — (b7-b4) | No register bits; should be written with 0 and read as undefined value | | — |

Notes:
1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. This register should not be set in single-chip mode.
3. $\overline{WR2}$ is output when the PM02 bit in the PM0 register is set to 1 ($\overline{RD}/\overline{WR0}/\overline{WR1}/\overline{WR2}/\overline{WR3}$) and bits EXBW1 and EXBW0 in the PBC register are set to 10b (32-bit width as the maximum width of external bus); otherwise, $\overline{CS3}$ is output.

**Figure 9.6 CSOP2 Register**

## Chip Selects 0 and 1 Boundary Setting Register (1)




| Symbol | Address | Reset Value |
|---|---|---|
| CB01 | 001Ah | 00h |

| Function | Setting Range | RW |
|---|---|---|
| Set this register to the value from A25 to A18 of the start address of $\overline{CS0}$ space.<br>The immediately preceding address mentioned above and lower is designated for $\overline{CS1}$ space | 04h to F8h (2) in memory expansion mode,<br>04h to FFh (2) in microprocessor mode | RW |

Notes:
1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. The setting value should be higher than that for the CB12 register.

**Figure 9.7 CB01 Register**

## Chip Selects 1 and 2 Boundary Setting Register (1)

b7 b0

| Symbol | Address | Reset Value |
|---|---|---|
| CB12 | 0016h | 00h |

| Function | Setting Range | RW |
|---|---|---|
| Set this register to the value from A25 to A18 of the start address of $\overline{CS1}$ space.<br>The immediately preceding address mentioned above and lower is designated for $\overline{CS2}$ space | 03h to F7h (2) in memory expansion mode,<br>03h to FEh (2) in microprocessor mode | RW |

Notes:
1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. The setting value should be higher than that for the CB23 register and lower than that for the CB01 register.

**Figure 9.8 CB12 Register**

## Chip Selects 2 and 3 Boundary Setting Register (1)

b7 b0

| Symbol | Address | Reset Value |
|---|---|---|
| CB23 | 0012h | 00h |

| Function | Setting Range | RW |
|---|---|---|
| Set this register to the value from A25 to A18 of the start address of $\overline{CS2}$ space.<br>The immediately preceding address mentioned above and lower is designated for $\overline{CS3}$ space | 02h to F6h (2) in memory expansion mode,<br>02h to FDh (2) in microprocessor mode | RW |

Notes:
1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. The setting value should be lower than that for the CB12 register.

**Figure 9.9 CB23 Register**

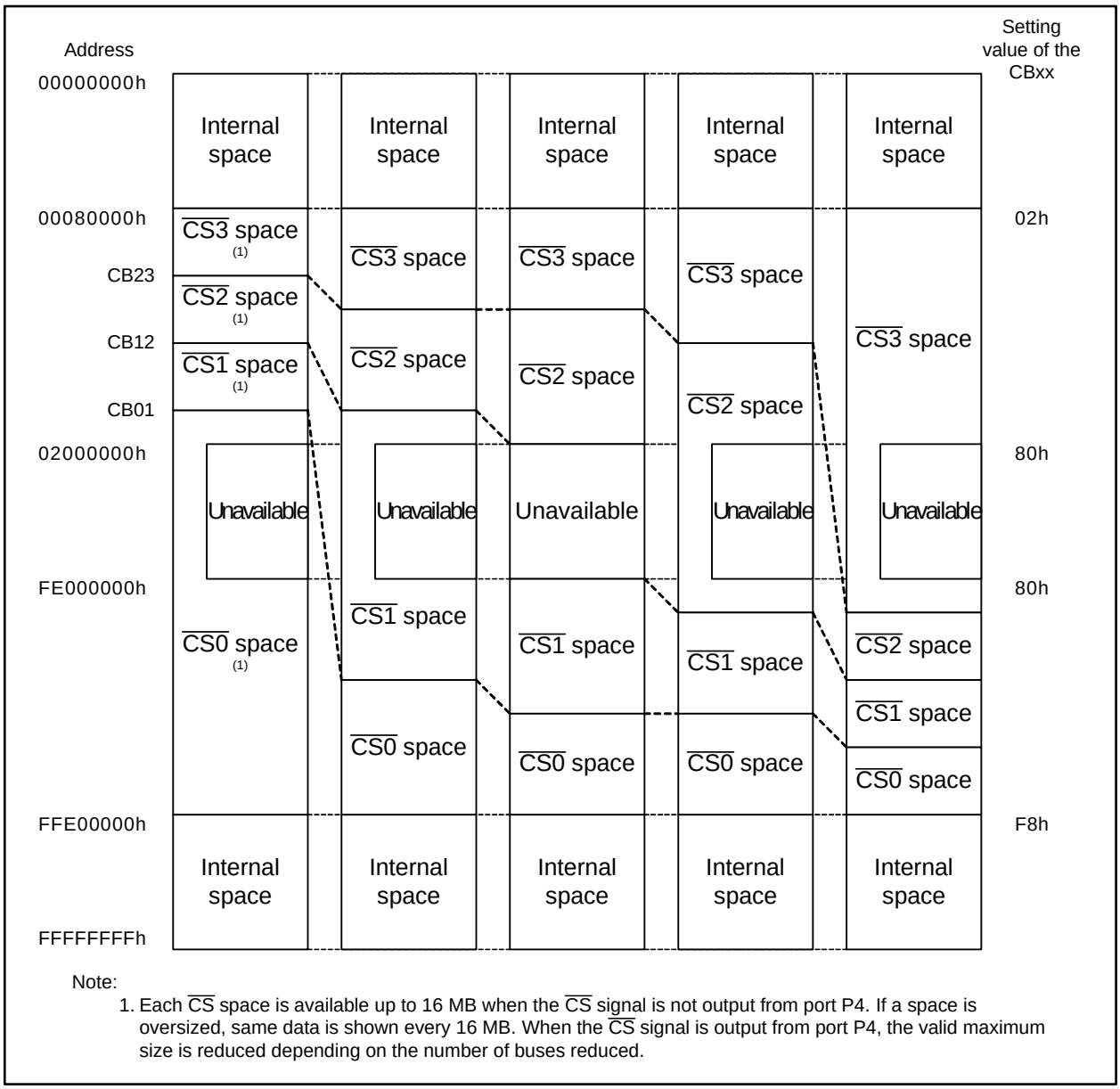
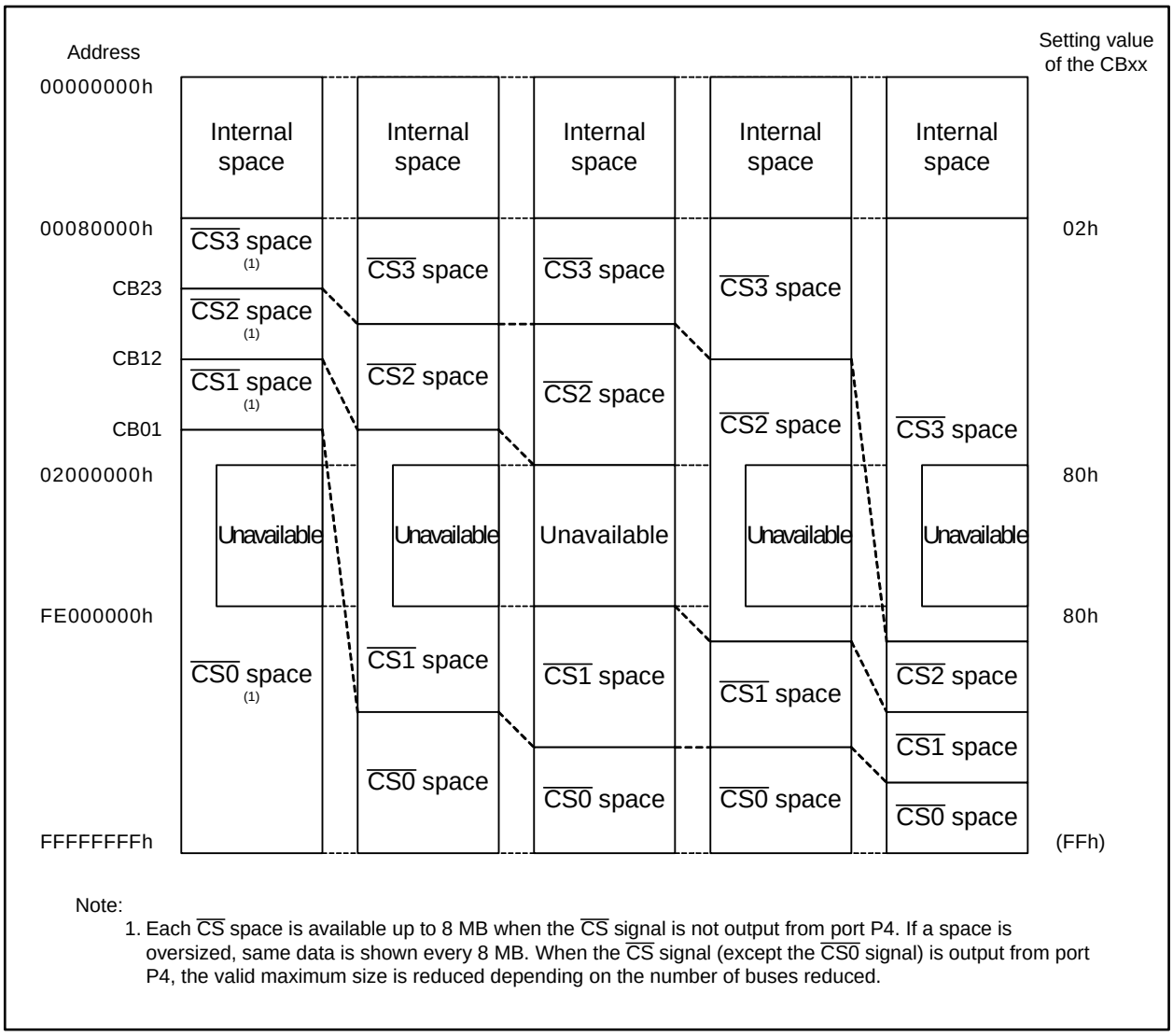



**Figure 9.10 Chip Select Spaces in Memory Expansion Mode**




Note:
1. Each $\overline{CS}$ space is available up to 8 MB when the $\overline{CS}$ signal is not output from port P4. If a space is oversized, same data is shown every 8 MB. When the $\overline{CS}$ signal (except the $\overline{CS0}$ signal) is output from port P4, the valid maximum size is reduced depending on the number of buses reduced.

**Figure 9.11 Chip Select Spaces in Microprocessor Mode**

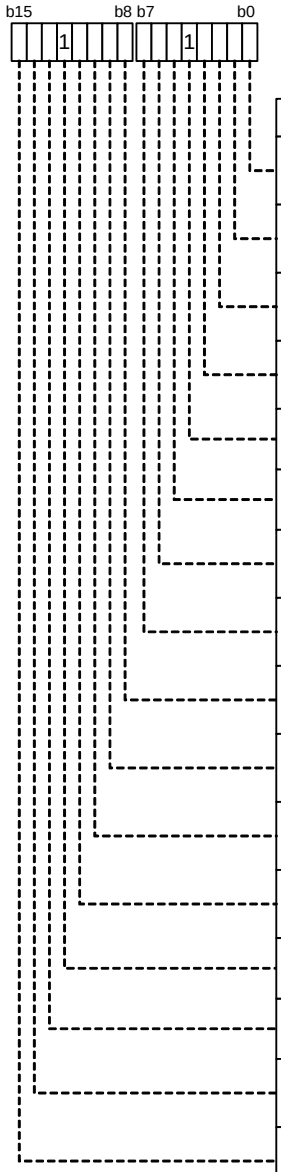
## 9.3.2 External Data Bus Width Setting

The external data bus width is selectable among 8 bits, 16 bits, and 32 bits. The bus width of each space is determined using bits BW1 and BW0 in registers EBC0 to EBC3. The maximum bus width for all spaces is set using bits EXBW1 and EXBW0 in the PBC register. The bus width specified in bits EXBW1 and EXBW0 should be equal to or greater than the value set using bits BW1 and BW0.

When an accessed space has a bus of less bit-width than that specified in bits EXBW1 and EXBW0, undefined value is output from the unused data output pins.

Figure 9.12 shows registers EBC0 to EBC3.

## External Bus Control Register i (i = 0 to 3) (1)




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

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| ESUR0 | Address Setup Before $\overline{RD}$ Setting Bit (2) | b1 b0<br>0 0 : *sur* = 0<br>0 1 : *sur* = 1<br>1 0 : *sur* = 2<br>1 1 : *sur* = 3 | RW |
| ESUR1 | | | |
| EWR0 | $\overline{RD}$ Pulse Width Setting Bit (2) | b3 b2<br>0 0 : *wr* = 1<br>0 1 : *wr* = 2<br>1 0 : *wr* = 3<br>1 1 : *wr* = 4 | RW |
| EWR1 | | | |
| — (b4) | Reserved | Should be written with 1 | RW |
| RDY | $\overline{RDY}$ Monitor Bit | 0: Ignore $\overline{RDY}$<br>1: Use $\overline{RDY}$ | RW |
| MPY0 | Multiplied Cycle Setting Bit (2) | b7 b6<br>0 0 : *mpy* = 1<br>0 1 : *mpy* = 2<br>1 0 : *mpy* = 3<br>1 1 : *mpy* = 4 | RW |
| MPY1 | | | |
| ESUW0 | Address Setup Before $\overline{WR}$ Setting Bit (2) | b9 b8<br>0 0 : *suw* = 0<br>0 1 : *suw* = 1<br>1 0 : *suw* = 2<br>1 1 : *suw* = 3 | RW |
| ESUW1 | | | |
| EWW0 | $\overline{WR}$ Pulse Width Setting Bit (2) | b11b10<br>0 0 : *ww* = 1<br>0 1 : *ww* = 2<br>1 0 : *ww* = 3<br>1 1 : *ww* = 4 | RW |
| EWW1 | | | |
| — (b12) | Reserved | Should be written with 1 | RW |
| MPX | External Bus Format Select Bit | 0: Separate bus<br>1: Multiplexed bus | RW |
| BW0 | External Bus Width Setting Bit (3) | b15b14<br>0 0 : 8-bit width<br>0 1 : 16-bit width<br>1 0 : 32-bit width (4)<br>1 1 : Do not use this combination | RW |
| BW1 | | | |

Notes:
1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. Refer to 9.3.5. “External Bus Timing” for the relation between register settings and practical timing.
3. The maximum value set here should be applied to bits EXBW1 and EXBW0 in the PBC register.
4. This bit setting is applicable only in the 144-pin package.

**Figure 9.12 Registers EBC0 to EBC3**

### 9.3.3 Separate Bus/Multiplexed Bus Selection

The bus format is selectable between separate bus format and multiplexed bus format. The bus format for each space is set using the MPX bit in registers EBC0 to EBC3. To specify multiplexed bus format for all spaces, the EXPMX bit in the PBC register should be set to 1 (multiplexed bus in all spaces). The ports P0 and P1, and P4_0 to P4_3 can be used as programmable I/O ports.

#### (1) Separate Bus

In this bus format, data and address have their own I/O pins.
To specify separate bus mode, the MPX bit in registers EBC0 to EBC3 should be set to 0. The data bus width is selectable among 8 bits, 16 bits, and 32 bits using bits BW1 and BW0 in registers EBC0 to EBC3.
According to the specified data bus width, pin functions vary as follows:
In 8-bit data bus format (bits EXBW1 and EXBW0 in the PBC register are set to 00b),
Port P0: data bus,
Ports P1, P12, and P13: programmable I/O ports.
In 16-bit data bus format (bits EXBW1 and EXBW0 are set to 01b),
Ports P0 and P1: data buses,
Ports P12 and P13: programmable I/O ports.
Note that port P1 (D8 to D15) becomes undefined if the MCU accesses an space where bits BW1 and BW0 are set to 00b (8-bit data bus).
In 32-bit data bus format (bits EXBW1 and EXBW0 are set to 10b),
Ports P0, P1, P12, and P13: data buses.
Note that ports P1, P12, and P13 (D8 to D31) become undefined if the MCU accesses an space where bits BW1 and BW0 are set to 00b (8-bit width data bus). In case of an access to an space set to 01b (16-bit data bus), ports P12 and P13 (D16 to D31) become undefined.

#### (2) Multiplexed Bus

In this bus format, data and address are input/output to/from a time-shared identical pin.
To specify multiplexed bus mode, the MPX bit in registers EBC0 to EBC3 should be set to 1.
According to the specified data bus width, pins are multiplexed as follows:
In 8-bit data bus format (bits BW1 and BW0 in registers EBC0 to EBC3 are set to 00b),
D0 to D7 are multiplexed with A0 to A7.
In 16-bit or 32-bit data bus format (bits BW1 and BW0 are set to 01b or 10b),
D0 to D15 are multiplexed with $\overline{BC0}$, A1/$\overline{BC2}$, and A2 to A15.
In microprocessor mode, an operation is started in separate bus format after a reset. Therefore the multiplexed bus format is available only for spaces $\overline{CS1}$ to $\overline{CS3}$ and is not available for $\overline{CS0}$ space.

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

**Table 9.2 Processor Mode and Pin Functions (1)**

| Process or Mode | Single-Chip Mode | Microprocessor Mode/Memory Expansion Mode | | | | | | Memory Expansion Mode | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Bus format | — | Separate bus only | | | Separate bus and multiplexed bus (mixed) | | | Multiplexed bus only | | |
| Data bus width | — | 8 bits only | 8/16 bits (mixed) | 8/16/32 bits (mixed) | 8 bits only | 8/16 bits (mixed) | 8/16/32 bits (mixed) | 8 bits only | 8/16 bits (mixed) | 8/16/32 bits (mixed) |
| P0_0 to P0_7 | I/O ports | D0 to D7 | | | | | | I/O ports | | |
| P1_0 to P1_7 | I/O ports | I/O ports | D8 to D15 | | I/O ports | D8 to D15 | | I/O ports | | |
| P2_0 | I/O port | A0 | A0 or $\overline{BC0}$ | | A0 or A0/D0 | A0, A0/D0, $\overline{BC0}$, or $\overline{BC0}$/D0 | | A0/D0 | A0/D0 or $\overline{BC0}$/D0 | |
| P2_1 | I/O port | A1 | A1 or $\overline{BC2}$ | | A1 or A1/D1 | | A1,A1/D1,$\overline{BC2}$, or $\overline{BC2}$/D1 | A1/D1 | | A1/D1 or $\overline{BC2}$/D1 |
| P2_2 to P2_7 | I/O ports | A2 to A7 | | | A2 to A7 or A2/D2 to A7/D7 | | | A2/D2 to A7/D7 | | |
| P3_0 to P3_7 | I/O ports | A8 to A15 | | | A8 to A15 | A8 to A15 or A8/D8 to A15/D15 | | A8 to A15 | A8/D8 to A15/D15 | |
| P4_0 to P4_3 | I/O ports | A16 to A19 | | | | | | A16 to A19 or I/O ports | | |
| P4_4 | I/O port | A20 or $\overline{CS3}$ | | | | | | | | |
| P4_5 | I/O port | A21 or $\overline{CS2}$ | | | | | | | | |
| P4_6 | I/O port | A22 or $\overline{CS1}$ | | | | | | | | |
| P4_7 | I/O port | A23 or $\overline{CS0}$ | | | | | | | | |
| P5_0 | I/O port | $\overline{WR}$ or $\overline{WR0}$ | | | | | | | | |
| P5_1 | I/O port | Undefined (2) | $\overline{BC1}$ or $\overline{WR1}$ | | Undefined (2) | $\overline{BC1}$ or $\overline{WR1}$ | | Undefined (2) | $\overline{BC1}$ or $\overline{WR1}$ | |
| P5_2 | I/O port | $\overline{RD}$ | | | | | | | | |
| P5_3 | I/O port | BCLK | | | | | | | | |
| P5_4 | I/O port | $\overline{HLDA}$ or $\overline{CS1}$ | | | | | | | | |
| P5_5 | I/O port | $\overline{HOLD}$ | | | | | | | | |
| P5_6 | I/O port | ALE or $\overline{CS2}$ | | | Set to ALE | | | | | |
| P5_7 | I/O port | $\overline{RDY}$ or $\overline{CS3}$ | | | | | | | | |
| P11_0 to P11_2 | I/O ports | $\overline{CS0}$ to $\overline{CS2}$ or I/O ports | | | | | | | | |
| P11_3 | I/O port | $\overline{CS3}$ or I/O port | | $\overline{CS3}$ or $\overline{WR2}$ | $\overline{CS3}$ or I/O port | | $\overline{CS3}$ to $\overline{WR2}$ | $\overline{CS3}$ or I/O port | | $\overline{CS3}$ or $\overline{WR2}$ |
| P11_4 | I/O port | I/O port | | $\overline{BC3}$ or $\overline{WR3}$ | I/O port | | $\overline{BC3}$ to $\overline{WR3}$ | I/O port | | $\overline{BC3}$ or $\overline{WR3}$ |
| P12_0 to P12_7 | I/O ports | I/O ports | | D16 to D23 | I/O ports | | D16 to D23 | I/O ports | | D16 to D23 |
| P13_0 to P13_7 | I/O ports | I/O ports | | D24 to D31 | I/O ports | | D24 to D31 | I/O ports | | D24 to D31 |

Notes:
1. Ports P11 to P15 are available in the 144-pin package only.
2. Undefined value is output.

**Table 9.3 Bus Format and Pin Functions (in Microprocessor Mode/Memory-Expansion Mode) (1)**

| Bus Format | Separate Bus | | | Multiplexed Bus | | |
|---|---|---|---|---|---|---|
| MPX bit | 0 | | | 1 | | |
| Bus width | 8 bits | 16 bits | 32 bits | 8 bits | 16 bits | 32 bits |
| Bits BW1 to BW0 | 00b | 01b | 10b | 00b | 01b | 10b |
| P0_0 to P0_7 | D0 to D7 | | | I/O ports | | |
| P1_0 to P1_7 | I/O ports | D8 to D15 | | I/O ports | | |
| P2_0 | A0 | $\overline{BC0}$ | | A0/D0 | $\overline{BC0}$/D0 | |
| P2_1 | A1 | | $\overline{BC2}$ | A1/D1 | | $\overline{BC2}$/D1 |
| P2_2 to P2_7 | A2 to A7 | | | A2/D2 to A7/D7 | | |
| P3_0 to P3_7 | A8 to A15 | | | A8/D8 to A15/D15 | | |
| P4_0 to P4_3 | A16 to A19 | | | A16 to A19 or I/O ports | | |
| P4_4 | A20 or $\overline{CS3}$ | | | | | |
| P4_5 | A21 or $\overline{CS2}$ | | | | | |
| P4_6 | A22 or $\overline{CS1}$ | | | | | |
| P4_7 | A23 or $\overline{CS0}$ ($\overline{CS0}$ fixed in microprocessor mode) | | | | | |
| P5_0 | $\overline{WR}$ or $\overline{WR0}$ | | | | | |
| P5_1 | Undefined (2) | $\overline{BC1}$ or $\overline{WR1}$ | | Undefined (2) | $\overline{BC1}$ or $\overline{WR1}$ | |
| P5_2 | $\overline{RD}$ | | | | | |
| P5_3 | BCLK | | | | | |
| P5_4 | $\overline{HLDA}$ or $\overline{CS1}$ | | | | | |
| P5_5 | $\overline{HOLD}$ | | | | | |
| P5_6 | ALE or $\overline{CS2}$ | | | Set to ALE | | |
| P5_7 | $\overline{RDY}$ or $\overline{CS3}$ | | | | | |
| P11_0 to P11_2 | $\overline{CS0}$ to $\overline{CS2}$ or I/O ports | | | | | |
| P11_3 | $\overline{CS3}$ or I/O port | | $\overline{CS3}$ or $\overline{WR2}$ | $\overline{CS3}$ or I/O port | | $\overline{CS3}$ or $\overline{WR2}$ |
| P11_4 | I/O port | | $\overline{BC3}$ or $\overline{WR3}$ | I/O port | | $\overline{BC3}$ or $\overline{WR3}$ |
| P12_0 to P12_7 | I/O ports | | D16 to D23 | I/O ports | | D16 to D23 |
| P13_0 to P13_7 | I/O ports | | D24 to D31 | I/O ports | | D24 to D31 |

Notes:
1. Ports P11 to P15 are available in the 144-pin package only.
2. Undefined value is output.

### 9.3.4 Read and Write Signals

In 16- or 32-bit data bus, the PM02 bit in the PM0 register selects a combination of $\overline{RD}$, $\overline{WR}$, $\overline{BC0}$, $\overline{BC1}$, $\overline{BC2}$, and $\overline{BC3}$ or $\overline{RD}$, $\overline{WR0}$, $\overline{WR1}$, $\overline{WR2}$, and $\overline{WR3}$ as read or write signals.

When bits EXBW1 and EXBW0 in the PBC register are set to 00b (8-bit data bus), the PM02 bit should be set to 0 ($\overline{RD}$/$\overline{WR}$/$\overline{BC0}$/$\overline{BC1}$/$\overline{BC2}$/$\overline{BC3}$). When bits EXBW1 and EXBW0 are set to 01b (16-bit data bus) or 10b (32-bit data bus) to access an 8-bit space, the combination of $\overline{RD}$, $\overline{WR}$, $\overline{BC0}$, $\overline{BC1}$, $\overline{BC2}$, and $\overline{BC3}$ is selected irrespective of the PM02 bit setting.

Table 9.4 and Table 9.5 list each signal operation.

The read and write signals after a reset are the following combination: $\overline{RD}$, $\overline{WR}$, $\overline{BC0}$, $\overline{BC1}$, $\overline{BC2}$, and $\overline{BC3}$. To shift to another combination, $\overline{RD}$, $\overline{WR0}$, $\overline{WR1}$, $\overline{WR2}$, and $\overline{WR3}$, the PM02 bit should be set first to write data to an external memory.

**Table 9.4 Signals $\overline{RD}$, $\overline{WR0}$, $\overline{WR1}$, $\overline{WR2}$, and $\overline{WR3}$ (1)**

| Data Bus Width | $\overline{RD}$ | $\overline{WR0}$ | $\overline{WR1}$ | $\overline{WR2}$ | $\overline{WR3}$ | External Data Bus Status |
|---|---|---|---|---|---|---|
| 32 bits (2) | L | H | H | H | H | Read 4-byte data |
| | H | L | H | H | H | Write 1-byte data to address 4n+0 |
| | H | H | L | H | H | Write 1-byte data to address 4n+1 |
| | H | H | H | L | H | Write 1-byte data to address 4n+2 |
| | H | H | H | H | L | Write 1-byte data to address 4n+3 |
| | H | L | L | H | H | Write 2-byte data to addresses 4n+0 to 4n+1 |
| | H | H | L | L | H | Write 2-byte data to addresses 4n+1 to 4n+2 |
| | H | H | H | L | L | Write 2-byte data to addresses 4n+2 to 4n+3 |
| | H | L | L | L | H | Write 3-byte data to addresses 4n+0 to 4n+2 |
| | H | H | L | L | L | Write 3-byte data to addresses 4n+1 to 4n+3 |
| | H | L | L | L | L | Write 4-byte data to addresses 4n+0 to 4n+3 |
| 16 bits | L | H | H | H/L (A1) | — | Read 2-byte data |
| | H | L | H | H/L (A1) | — | Write 1-byte data to even address |
| | H | H | L | H/L (A1) | — | Write 1-byte data to odd address |
| | H | L | L | H/L (A1) | — | Write 2-byte data to both even and odd addresses |
| 8 bits | L | H ($\overline{WR}$) | — | H/L (A1) | — | Read 1-byte data |
| | H | L ($\overline{WR}$) | — | H/L (A1) | — | Write 1-byte data |

Notes:
1. Signals $\overline{WR2}$ and $\overline{WR3}$ are available in the 144-pin package only.
2. Signals for 32-bit data bus width can be set in the 144-pin package only.

**Table 9.5 Signals $\overline{RD}$, $\overline{WR}$, $\overline{BC0}$, $\overline{BC1}$, $\overline{BC2}$, and $\overline{BC3}$ (1)**

| Data Bus Width | $\overline{RD}$ | $\overline{WR}$ | $\overline{BC0}$ | $\overline{BC1}$ | $\overline{BC2}$ | $\overline{BC3}$ | External Data Bus Status |
|---|---|---|---|---|---|---|---|
| 32 bits (2) | L | H | L | L | L | L | Read 4-byte data |
| | H | L | L | H | H | H | Write 1-byte data to address 4n+0 |
| | H | L | H | L | H | H | Write 1-byte data to address 4n+1 |
| | H | L | H | H | L | H | Write 1-byte data to address 4n+2 |
| | H | L | H | H | H | L | Write 1-byte data to address 4n+3 |
| | H | L | L | L | H | H | Write 2-byte data to addresses 4n+0 to 4n+1 |
| | H | L | H | L | L | H | Write 2-byte data to addresses 4n+1 to 4n+2 |
| | H | L | H | H | L | L | Write 2-byte data to addresses 4n+2 to 4n+3 |
| | H | L | L | L | L | H | Write 3-byte data to addresses 4n+0 to 4n+2 |
| | H | L | H | L | L | L | Write 3-byte data to addresses 4n+1 to 4n+3 |
| | H | L | L | L | L | L | Write 4-byte data to addresses 4n+0 to 4n+3 |
| 16 bits | L | H | L | L | H/L (A1) | — | Read 2-byte data |
| | H | L | L | H | H/L (A1) | — | Write 1-byte data to even address |
| | H | L | H | L | H/L (A1) | — | Write 1-byte data to odd address |
| | H | L | L | L | H/L (A1) | — | Write 2-byte data to both even and odd addresses |
| 8 bits | L | H | H/L (A0) | — | H/L (A1) | — | Read 1-byte data |
| | H | L | H/L (A0) | — | H/L (A1) | — | Write 1-byte data |

Notes:
1. Signals $\overline{BC2}$ and $\overline{BC3}$ are available in the 144-pin package only.
2. Signals for 32-bit data bus width can be set in the 144-pin package only.

### 9.3.5 External Bus Timing

The external bus timing is set using registers EBC0 to EBC3. The reference clock is the base clock set using bits BCD1 and BCD0 in the CCR register.

Table 9.6 lists the bit setting of MPY1, MPY0, ESUR1, and ESUR0 and the Tsu(A-R) (address setup before $\overline{RD}$), Table 9.7 lists the bit setting of MPY1, MPY0, EWR1, and EWR0 and the Tw(R) ($\overline{RD}$ pulse width), Table 9.8 lists the bit setting of MPY1, MPY0, ESUW1, and ESUW0 and the Tsu(A-W) (address setup before $\overline{WR}$), and Table 9.9 lists the bit setting of MPY1, MPY0, EWW1, and EWW0 and the Tw(W) ($\overline{WR}$ pulse width).

**Table 9.6 The Tsu(A-R) and Bit Settings: MPY1, MPY0, ESUR1, and ESUR0 (unit: cycles)**

| ESUR1 and ESUR0 Bit Settings | | Separate Bus | | | | Multiplexed Bus | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | MPY1 and MPY0 bit settings | | | | MPY1 and MPY0 bit settings | | | |
| | | 00b | 01b | 10b | 11b | 00b | 01b | 10b | 11b |
| | | *mpy = 1* | *mpy = 2* | *mpy = 3* | *mpy = 4* | *mpy = 1* | *mpy = 2* | *mpy = 3* | *mpy = 4* |
| 00b | *sur = 0* | 0.5 | 0.5 | 0.5 | 0.5 | 1 | 1 | 1 | 1 |
| 01b | *sur = 1* | 1.5 | 2.5 | 3.5 | 4.5 | 2 | 3 | 4 | 5 |
| 10b | *sur = 2* | 2.5 | 4.5 | 6.5 | 8.5 | 3 | 5 | 7 | 9 |
| 11b | *sur = 3* | 3.5 | 6.5 | 9.5 | 12.5 | 4 | 7 | 10 | 13 |
| Formula | | Tsu(A-R) = *sur* × *mpy* + 0.5 | | | | Tsu(A-R) = *sur* × *mpy* + 1 | | | |

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

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

Note:
1. Do not set this value.

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

| ESUW1 and ESUW0 Bit Settings | | MPY1 and MPY0 Bit Settings | | | |
|---|---|---|---|---|---|
| | | 00b | 01b | 10b | 11b |
| | | *mpy = 1* | *mpy = 2* | *mpy = 3* | *mpy = 4* |
| 00b | *suw = 0* | 1 | 1 | 1 | 1 |
| 01b | *suw = 1* | 2 | 3 | 4 | 5 |
| 10b | *suw = 2* | 3 | 5 | 7 | 9 |
| 11b | *suw = 3* | 4 | 7 | 10 | 13 |
| Formula | | Tsu(A-W) = *suw* × *mpy* + 1 | | | |

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

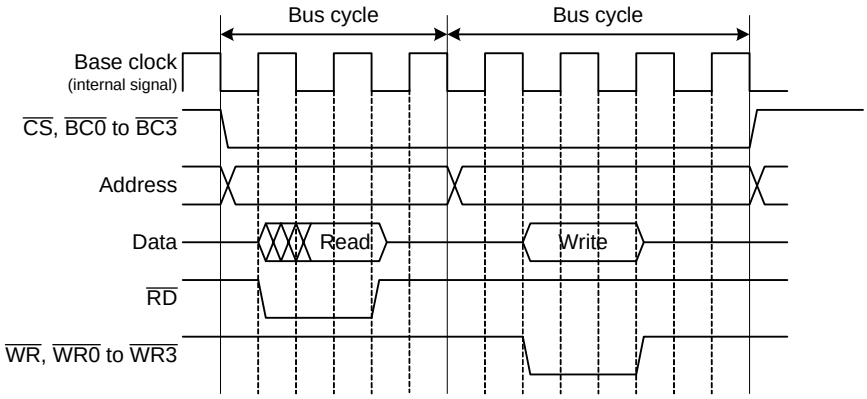
| EWW1 and EWW0 Bit Settings | | MPY1 and MPY0 Bit Settings | | | |
|---|---|---|---|---|---|
| | | 00b | 01b | 10b | 11b |
| | | *mpy = 1* | *mpy = 2* | *mpy = 3* | *mpy = 4* |
| 00b | *ww = 1* | 0.5 (1) | 1.5 | 2.5 | 3.5 |
| 01b | *ww = 2* | 1.5 | 3.5 | 5.5 | 7.5 |
| 10b | *ww = 3* | 2.5 | 5.5 | 8.5 | 11.5 |
| 11b | *ww = 4* | 3.5 | 7.5 | 11.5 | 15.5 |
| Formula | | Tw(W) = *ww* × *mpy* - 0.5 | | | |

Note:
1. Do not set this value.

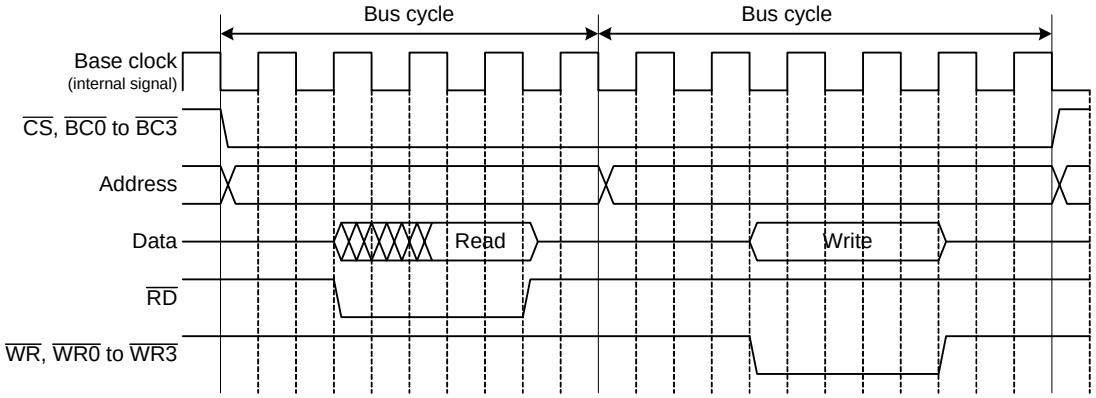
Figure 9.13 and Figure 9.14 show an example of external bus timing in separate bus format (the MPX bit is set to 0) and in multiplexed bus format (the MPX bit is set to 1), respectively.
Note that the actual bus cycles are adjusted to be the integral multiple of peripheral bus clock as follows:

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

(1) When the EBCi register is XX01 0100 0001 0000b




(2) When the EBCi register is XX01 1001 0001 0101b




(3) When the EBCi register is XX01 0101 0101 0101b

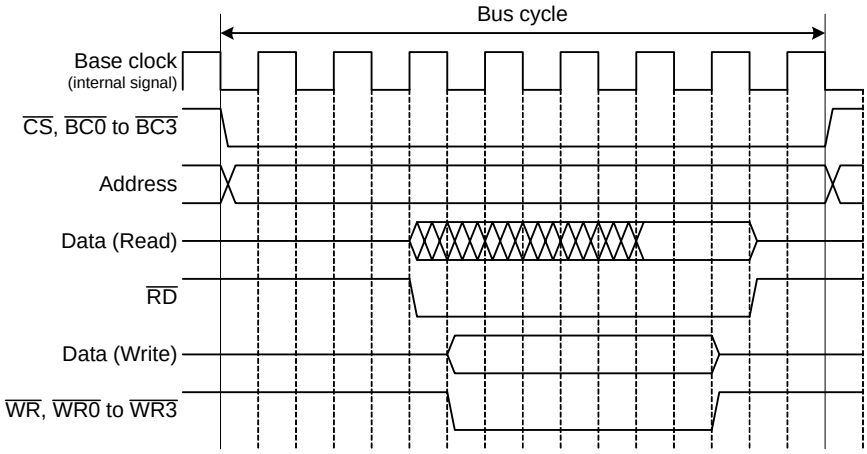
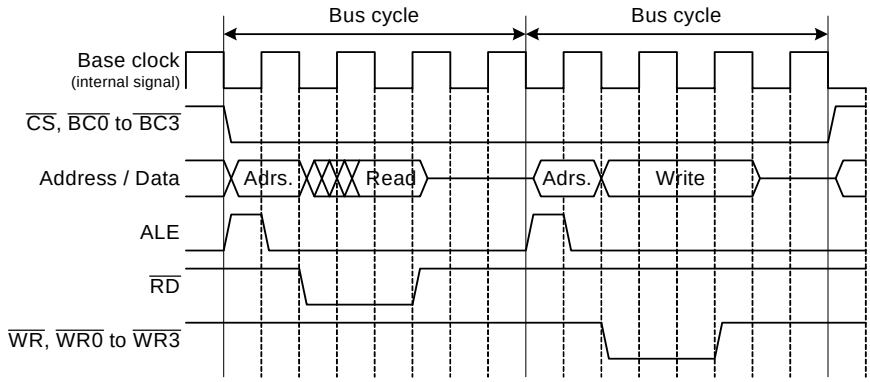
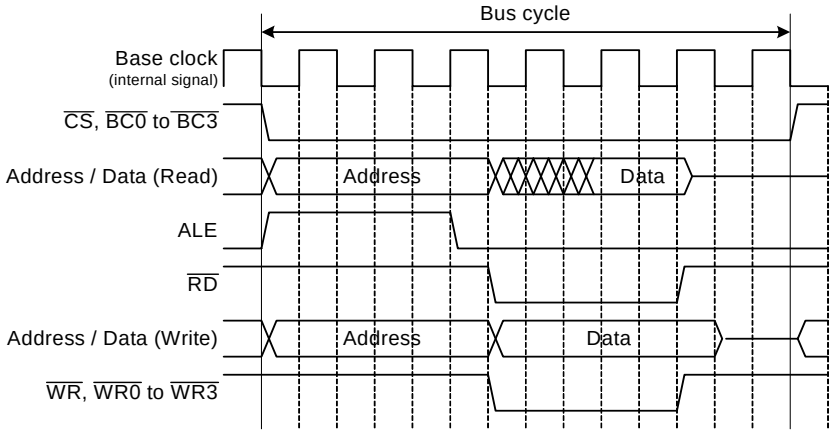



**Figure 9.13 External Bus Timing in Separate Bus Format (i = 0 to 3)**

(1) When the EBCi register is XX11 0100 0001 0100b




(2) When the EBCi register is XX11 1010 0001 1010b




(3) When the EBCi register is XX11 0101 0101 0101b

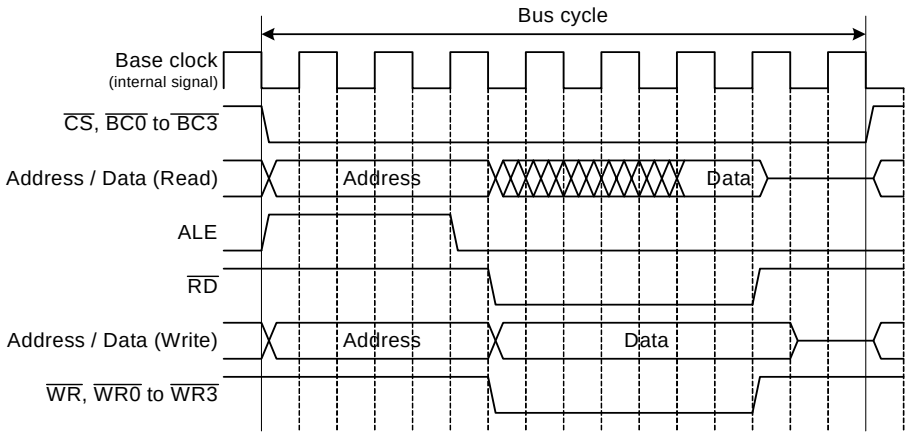



**Figure 9.14 External Bus Timing in Multiplexed Bus Format (i = 0 to 3)**

### 9.3.6 ALE Signal

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

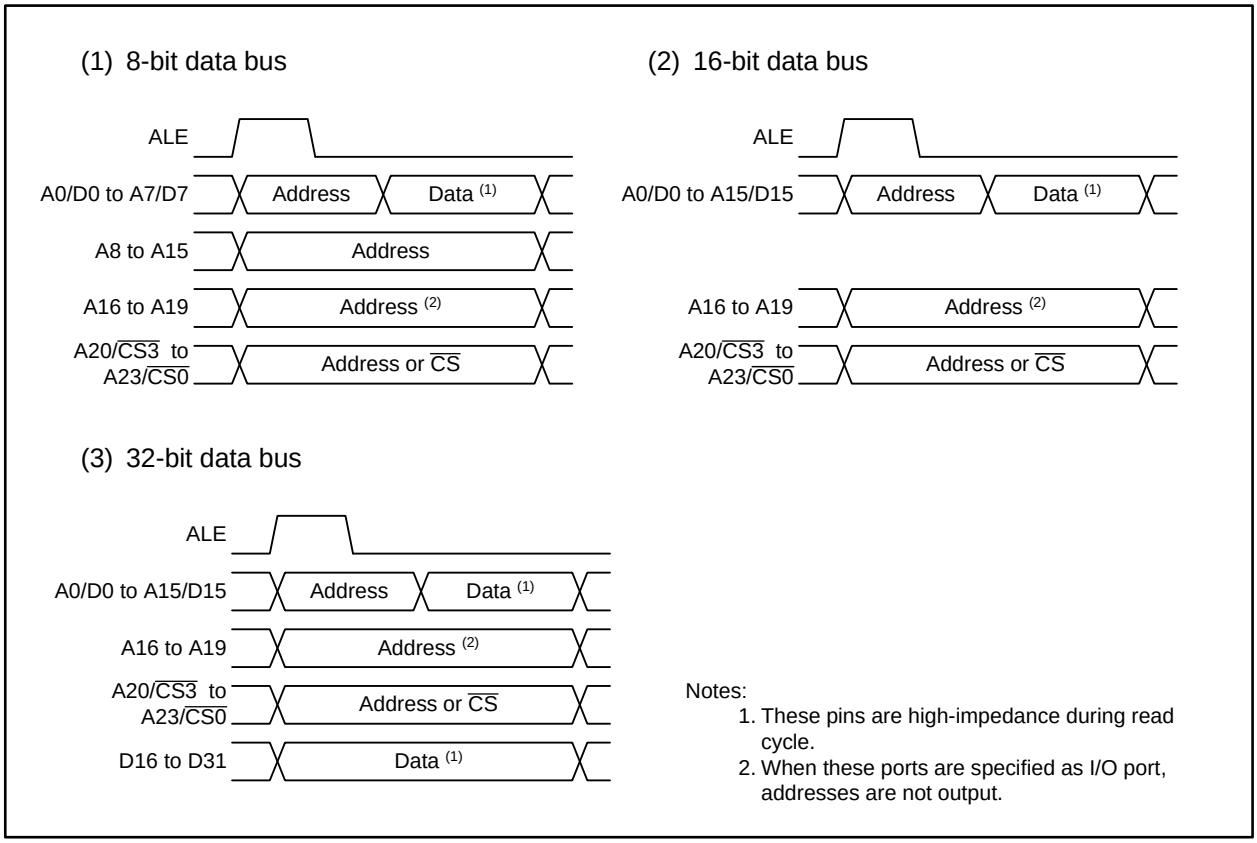



**Figure 9.15 ALE Signal and Address Bus/Data Bus**

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

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

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

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

Since the base clock is not output to external pins, practically, the $\overline{\text{RDY}}$ signal becomes low when the signals $\overline{\text{RD}}$, $\overline{\text{WR}}$, and $\overline{\text{WR0}}$ to $\overline{\text{WR3}}$ are held in a low level and it becomes high synchronizing the rise of the BCLK signal.

Figure 9.16 shows an example of $\overline{\text{RDY}}$ signal generator and Table 9.10 lists setting conditions of registers EBC0 to EBC3 to use this circuit. Figure 9.17 shows examples of bus cycle that is extended by the $\overline{\text{RDY}}$ signal.

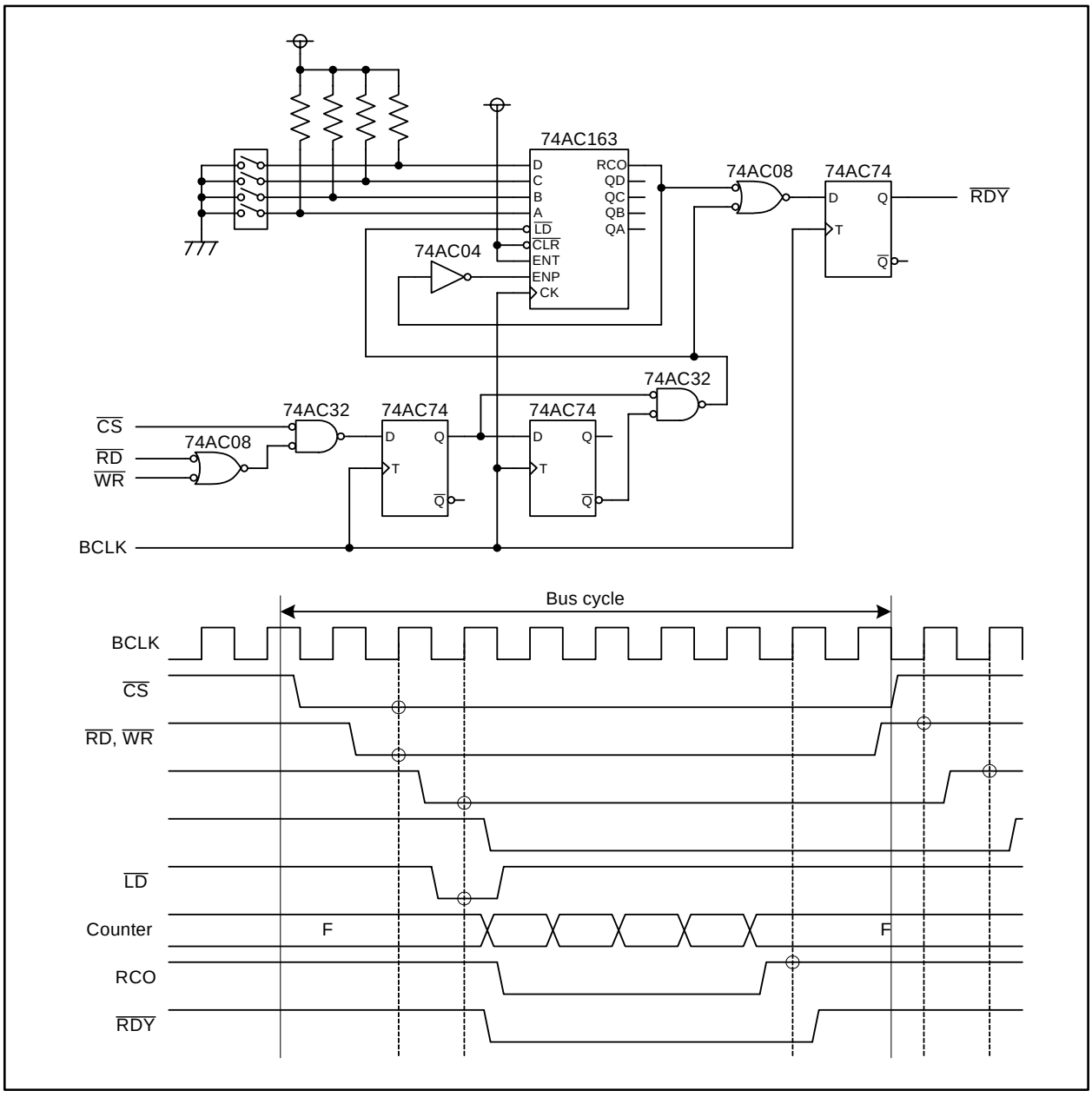



**Figure 9.16 $\overline{\text{RDY}}$ Signal Generation Circuitry**

**Table 9.10 Setting Conditions of the EBCi register when Using the Circuit in Figure 9.16 (i = 0 to 3)**

| Peripheral Bus Clock Frequency | Setting Condition | Setting Example |
|---|---|---|
| BCLK = 1/2 base clock | *mpy* = 3<br>In separate bus<br>$\overline{RD}$ pulse width ≥ 9.5<br>$\overline{WR}$ pulse width ≥ 11.5<br>$\overline{RD}$/$\overline{WR}$ high level width ≥ 2.5<br>In multiplexed bus<br>$\overline{RD}$ pulse width ≥ 11.5<br>$\overline{WR}$ pulse width ≥ 11.5 | In separate bus<br>EBCi = XX01 1101 1011 1001b<br>etc.<br>In multiplexed bus<br>EBCi = XX11 1101 1011 1101b<br>etc. |
| BCLK = 1/3 base clock | *mpy* = 3<br>In separate bus<br>$\overline{RD}$ pulse width ≥ 12.5<br>$\overline{WR}$ pulse width ≥ 11.5<br>$\overline{RD}$/$\overline{WR}$ high level width ≥ 3.5<br>In multiplexed bus<br>$\overline{RD}$ pulse width ≥ 11.5<br>$\overline{WR}$ pulse width ≥ 11.5 | In separate bus<br>EBCi = XX01 1101 1011 1101b<br>etc.<br>In multiplexed bus<br>EBCi = XX11 1101 1011 1101b<br>etc. |
| BCLK = 1/4 base clock | *mpy* = 4<br>In separate bus<br>$\overline{RD}$ pulse width ≥ 20.5<br>$\overline{WR}$ pulse width ≥ 19.5<br>$\overline{RD}$/$\overline{WR}$ high level width ≥ 4.5<br>In multiplexed bus<br>$\overline{RD}$ pulse width ≥ 19.5<br>$\overline{WR}$ pulse width ≥ 19.5 | In separate bus<br>unavailable<br>In multiplexed bus<br>unavailable |

X: given value

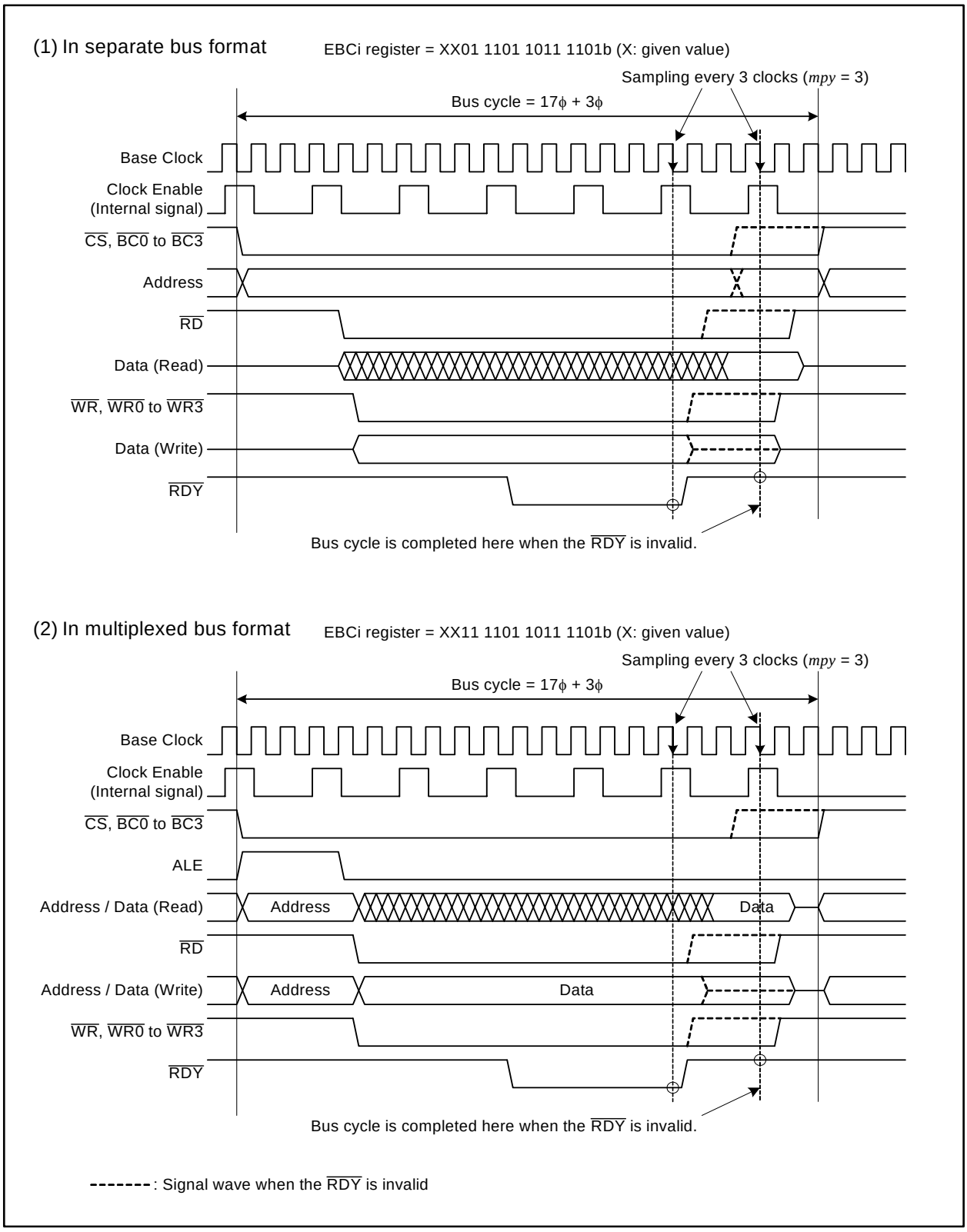



**Figure 9.17 An Example of Bus Cycle Extended by $\overline{\text{RDY}}$ Signal (f(BCLK) = 1/2 f(Base)) (i = 0 to 3)**

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

The $\overline{\text{HOLD}}$ signal is used when the external bus master requests the external bus from the CPU. When the external bus master drives the $\overline{\text{HOLD}}$ pin low, the CPU outputs a low signal from the $\overline{\text{HLDA}}$ pin after the ongoing bus access is completed. Then the external bus privilege is transferred to the external bus master. While the $\overline{\text{HOLD}}$ pin is held low, the CPU does not start the next bus cycle.
To return the bus privilege to the CPU, the external bus master should verify the $\overline{\text{HLDA}}$ pin is held low, and then drive the $\overline{\text{HOLD}}$ pin high.
Table 9.11 lists the MCU status in a hold state.
The bus is used in the following priority order: External bus master, DMAC, and CPU.

**Table 9.11 The MCU Status in Hold State**

| Item | State |
|---|---|
| Oscillation | ON |
| Address bus, data bus, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{BC0}}$ to $\overline{\text{BC3}}$ | High-impedance |
| $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WR0}}$ to $\overline{\text{WR3}}$ | High-impedance |
| Programmable I/O port | The state when $\overline{\text{HOLD}}$ was received is held |
| $\overline{\text{HLDA}}$ pin | Low is output |
| Internal peripheral circuit | ON (excluding the watchdog timer) |
| ALE pin | Low is output |

### 9.3.9 BCLK Output

The BCLK, which has the same frequency as peripheral bus clock, is a divided clock generated by PLL. In memory expansion mode or microprocessor mode, the BCLK is output from port P5_3 when the PM07 bit in the PM0 register is set to 0 (BCLK output) and bits CM01 and CM00 in the CM0 register are set to 00b (I/O port P5_3). In single-chip mode, it cannot be output. Refer to 8. “Clock Generator” for details.

## 9.4 External Bus Status when Accessing Internal Space

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

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

| Pin | | Pin State when Accessing SFR | Pin State when Accessing Internal Memory |
|---|---|---|---|
| Address bus | | Address is output | The address of SFR or external space last accessed is held |
| Data bus | Read Cycle | High-impedance | High-impedance |
| | Write Cycle | Data is output | Undefined |
| $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ | | High is output | High is output |
| $\overline{\text{BC0}}$ to $\overline{\text{BC3}}$ | | $\overline{\text{BC0}}$ to $\overline{\text{BC3}}$ are output | The address of SFR or external space last accessed is held |
| $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WR0}}$ to $\overline{\text{WR3}}$ | | $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WR0}}$ to $\overline{\text{WR3}}$ are output | High is output |
| ALE | | ALE is output | ALE is output |

## 9.5 Notes on Bus

### 9.5.1 Notes on System Designing

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

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

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

### 9.5.2 Notes on Register Settings

#### 9.5.2.1 Chip Select Boundary Select Registers

When using single-chip mode exclusively, do not change values after a reset for registers CB01, CB12, and CB23.

When the CPU operation is performed in memory expansion mode more than once, set a value within the specified range to all of these registers irrespective of the use of them.

#### 9.5.2.2 External Bus Control Registers

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

# 10. Protection

This function protects important registers from being easily overwritten when a program goes out of control. It contains the following registers: PRCR, PRCR2, PRCR3, and PRR.

## 10.1 PRCR Register

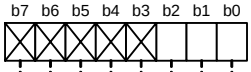
Figure 10.1 shows the PRCR register. Registers protected by the bits in the PRCR register are listed in Table 10.1.

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

| Bit | Protected Registers |
|---|---|
| PRC0 | CM0, CM1, CM2, and PM3 |
| PRC1 | PM0, PM2, CSOP0, CSOP1, CSOP2, INVC0, INVC1, IOBC, and I2CMR |
| PRC2 | PLC0, PLC1, PD9, and P9_iS (i = 0 to 7) |

The PRC2 bit becomes 0 (write disabled) when a write operation is performed in a given address after this bit is set to 1 (write enabled). In registers PD9, P9_iS (i = 0 to 7), PLC0, and PLC1, the write operation should be performed immediately after the instruction to set the PRC2 bit to 1. Any interrupt or DMA transfer should not be accepted between this instruction and the next one. Bits PRC0 and PRC1 are not set to 0 even if data is written to a given address. These bits should be set to 0 by a program.

Protect Register




| Symbol | Address | Reset Value |
|---|---|---|
| PRCR | 4004Ah | XXXX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PRC0 | Protect Bit 0 | Enable to write to registers CM0, CM1, CM2, and PM3<br>0: Write disabled<br>1: Write enabled | RW |
| PRC1 | Protect Bit 1 | Enable to write to registers PM0, PM2, CSOP0, CSOP1, CSOP2, INVC0, INVC1, IOBC, and I2CMR<br>0: Write disabled<br>1: Write enabled | RW |
| PRC2 | Protect Bit 2 (1) | Enable to write to registers PLC0, PLC1, PD9, and P9_iS (i = 0 to 7)<br>0: Write disabled<br>1: Write enabled | RW |
| —<br>(b7-b3) | No register bits; should be written with 0 and read as undefined value | | — |

Note:
1. The PRC2 bit becomes 0 if a write operation is performed in a given address after this bit is set to 1. Bits PRC0 and PRC1 do not automatically become 0. They should be set to 0 by a program.

**Figure 10.1 PRCR Register**

## 10.2 PRCR2 Register

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

Protect Register 2

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | X | X |

| Symbol | Address | Reset Value |
|---|---|---|
| PRCR2 | 4405Fh | 0XXX XXXXb |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b6-b0) | No register bits; should be written with 0 and read as undefined value | | — |
| PRC27 | CM3 Protect Bit | Enable to write to the CM3 register<br>0: Write disabled<br>1: Write enabled | RW |

**Figure 10.2 PRCR2 Register**

## 10.3 PRCR3 Register

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

**Table 10.2 Registers Protected by the PRCR3 Register**

| Bit | Registers to be protected |
|---|---|
| PRC31 | VRCR, LVDC, and DVCR |

Protect Register 3

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | | 0 |

| Symbol | Address | Reset Value |
|---|---|---|
| PRCR3 | 4004Ch | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b0) | Reserved | Should be written with 0 | RW |
| PRC31 | Protect Bit 31 | Enable to write to registers VRCR, LVDC, and DVCR<br>0: Write disabled<br>1: Write enabled | RW |
| — (b7-b2) | Reserved | Should be written with 0 | RW |

**Figure 10.3 PRCR3 Register**

## 10.4 PRR Register

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

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

Protect Release Register

b7 b0

| Symbol | Address | Reset Value |
|---|---|---|
| PRR | 0007h | 00h |

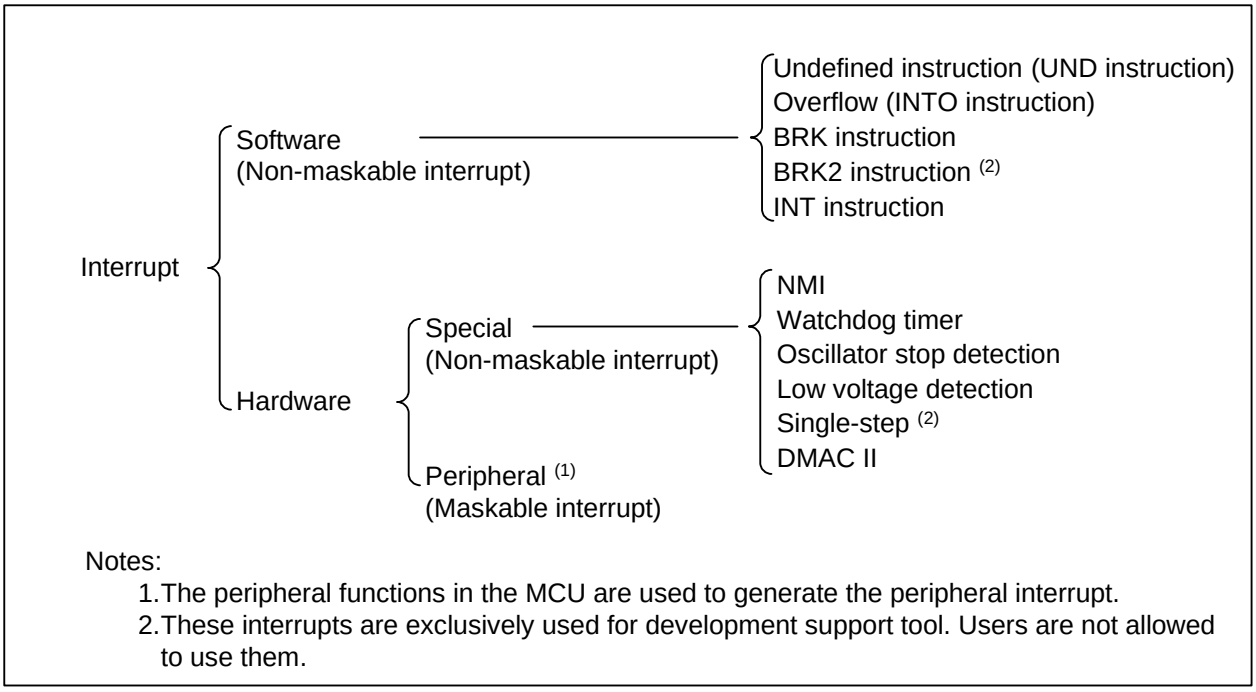
| Function | Setting Range | RW |
|---|---|---|
| Control the protection for registers CCR, FMCR, PBC, FEBC0, FEBC3, EBC0 to EBC3, CB01, CB12, and CB23.<br>AAh: Write enabled<br>Value other than AAh: Write disabled | 00h to FFh | RW |

**Figure 10.4 PRR Register**

# 11. Interrupts

## 11.1 Interrupt Types

Figure 11.1 shows types of interrupts.




Notes:
1. The peripheral functions in the MCU are used to generate the peripheral interrupt.
2. These interrupts are exclusively used for development support tool. Users are not allowed to use them.

**Figure 11.1 Interrupts**

The interrupts are also classified into maskable/non-maskable.

(1) Maskable Interrupt

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

(2) Non-maskable Interrupt

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

## 11.2 Software Interrupt

Software interrupts are non-maskable. A software interrupt is generated by executing an instruction. There are five types of software interrupts as follows:

### (1) Undefined Instruction Interrupt

This interrupt occurs when the UND instruction is executed.

### (2) Overflow Interrupt

This interrupt occurs when the INTO instruction is executed while the O flag is 1. The following instructions may change the O flag to 1, depending on the operation result:
ABS, ADC, ADCF, ADD, ADDF, ADSF, CMP, CMPF, CNVIF, DIV, DIVF, DIVU, DIVX, EDIV, EDIVU, EDIVX, MUL, MULF, MULU, MULX, NEG, RMPA, ROUND, SBB, SCMPU, SHA, SUB, SUBF, SUNTIL, and SWHILE

### (3) BRK Instruction Interrupt

This interrupt occurs when the BRK instruction is executed.

### (4) BRK2 Instruction Interrupt

This interrupt occurs when the BRK2 instruction is executed.
This interrupt is only meant for use with the development support tool, and users are not allowed to use it.

### (5) INT Instruction Interrupt

This interrupt occurs when the INT instruction is executed with a selected software interrupt number from 0 to 255. Numbers 0 to 127 are designated for peripheral interrupts. That is, the INT instruction with a number from 0 to 127 has the same interrupt handler as that for the peripheral interrupt.
The stack pointer (SP), which contains two types, is specified by the stack pointer select flag (U flag). For numbers 0 to 127, when an interrupt request is accepted, the U flag is saved to select the interrupt stack pointer (ISP) before the interrupt sequence is executed. The saved data of the U flag is restored upon returning from the interrupt handler. For numbers 128 to 255, the stack pointer used before the interrupt request acceptance remains unchanged for the interrupt sequence.

## 11.3 Hardware Interrupt

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

### 11.3.1 Special Interrupt

Special interrupts are non-maskable. There are five interrupts as follows:

(1) NMI (Non Maskable Interrupt)

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

(2) Watchdog Timer Interrupt

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

(3) Oscillator Stop Detection Interrupt

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

(4) Low Voltage Detection Interrupt

This interrupt occurs if the lowered voltage input to VCC is detected by the voltage detector. Refer to 6.2 “Low Voltage Detector” for details.

(5) Single-step Interrupt

This interrupt is only meant for use with the development support tool, and users are not allowed to use it.

### 11.3.2 Peripheral Interrupt

Peripheral interrupt is maskable, and is generated when an interrupt request from the peripheral functions in the MCU is accepted. It shares the interrupt vector table with software interrupt numbers 0 to 127 for the INT instruction.
Refer to Table 11.2 to Table 11.5 for details on the interrupt sources. Refer to the relevant description for details on each function.

## 11.4 Fast Interrupt

Fast interrupt enables the CPU to minimize the overhead of interrupt sequence. In peripheral interrupts, only one interrupt with the highest priority can be specified as the fast interrupt.

The procedure to set up a fast interrupt is as follows:

(1) Set both FSIT bits in registers RIPL1 and RIPL2 to 1 (interrupt request level 7 available for fast interrupt).

(2) Set both DMAII bits in registers RIPL1 and RIPL2 to 0 (interrupt request level 7 available for interrupts).

(3) Set the start address of the fast interrupt handler to the VCT register.

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

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

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

## 11.5 Interrupt Vectors

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

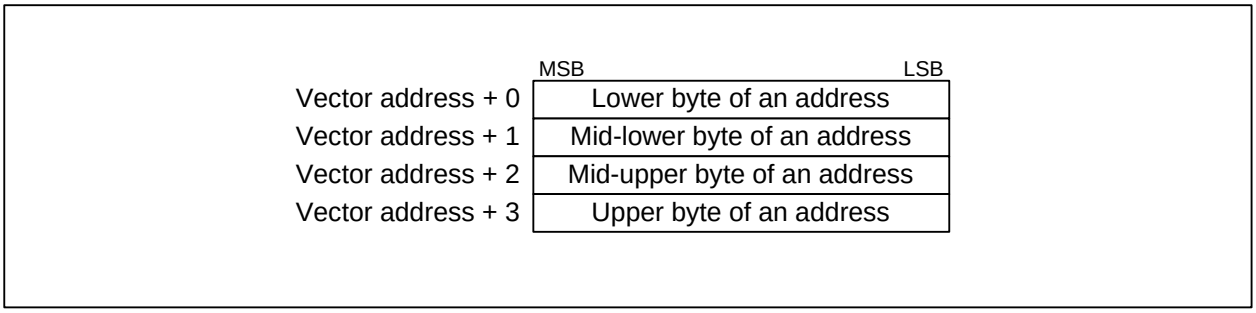



**Figure 11.2 Interrupt Vector**

### 11.5.1 Fixed Vector Table

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

**Table 11.1 Fixed Vector Table**

| Interrupt Source | Vector Table Addresses (Address (L) to Address (H)) | Remarks | Reference |
|---|---|---|---|
| Undefined instruction | FFFFFFDCh to FFFFFFDFh | Interrupt by the UND instruction | R32C/100 Series Software Manual |
| Overflow | FFFFFFE0h to FFFFFFE3h | Interrupt by the INTO instruction | |
| BRK instruction | FFFFFFE4h to FFFFFFE7h | If address FFFFFFE7h is FFh, the instruction jumps to the address stored into software interrupt 0 in the relocatable vector table | |
| — | FFFFFFE8h to FFFFFFEBh | Reserved | |
| — | FFFFFFECh to FFFFFFEFh | Reserved | |
| Watchdog timer<br>Oscillator stop detection<br>Low voltage detection | FFFFFFF0h to FFFFFFF3h | These addresses are shared by the watchdog timer interrupt, oscillator stop detection interrupt, and low voltage detection interrupt | 12. “Watchdog Timer”<br>8. “Clock Generator”<br>6.2 “Low Voltage Detector” |
| — | FFFFFFF4h to FFFFFFF7h | Reserved | |
| NMI | FFFFFFF8h to FFFFFFFBh | External interrupt by the $\overline{\text{NMI}}$ pin | |
| Reset | FFFFFFFCh to FFFFFFFFh | | 5. “Resets” |

### 11.5.2 Relocatable Vector Table

The relocatable vector table occupies a 1024-byte memory space from the start address set in the INTB register. Table 11.2 to Table 11.5. list the relocatable vector table entries.
An address in multiples of 4 should be set in the INTB register for faster interrupt sequence.

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

| Interrupt Source | Vector Table Relative Addresses (Address (L) to Address (H)) (1) | Software Interrupt Number | Reference |
|---|---|---|---|
| BRK instruction (2) | +0 to +3 (0000h to 0003h) | 0 | R32C/100 Series Software Manual |
| Reserved | +4 to +7 (0004h to 0007h) | 1 | |
| UART5 transmission, NACK (3) | +8 to +11 (0008h to 000Bh) | 2 | 18. “Serial Interface” |
| UART5 reception, ACK (3) | +12 to +15 (000Ch to 000Fh) | 3 | |
| UART6 transmission, NACK (3) | +16 to +19 (0010h to 0013h) | 4 | |
| UART6 reception, ACK (3) | +20 to +23 (0014h to 0017h) | 5 | |
| Bus collision detection, start condition detection, or stop condition detection (UART5 or UART6) (3, 4) | +24 to +27 (0018h to 001Bh) | 6 | |
| Reserved | +28 to +31 (001Ch to 001Fh) | 7 | |
| DMA0 Transfer Complete | +32 to +35 (0020h to 0023h) | 8 | 13. “DMAC” |
| DMA1 Transfer Complete | +36 to +39 (0024h to 0027h) | 9 | |
| DMA2 Transfer Complete | +40 to +43 (0028h to 002Bh) | 10 | |
| DMA3 Transfer Complete | +44 to +47 (002Ch to 002Fh) | 11 | |
| Timer A0 | +48 to +51 (0030h to 0033h) | 12 | 16.1 “Timer A” |
| Timer A1 | +52 to +55 (0034h to 0037h) | 13 | |
| Timer A2 | +56 to +59 (0038h to 003Bh) | 14 | |
| Timer A3 | +60 to +63 (003Ch to 003Fh) | 15 | |
| Timer A4 | +64 to +67 (0040h to 0043h) | 16 | |
| UART0 transmission, NACK (3) | +68 to +71 (0044h to 0047h) | 17 | 18. “Serial Interface” |
| UART0 reception, ACK (3) | +72 to +75 (0048h to 004Bh) | 18 | |
| UART1 transmission, NACK (3) | +76 to +79 (004Ch to 004Fh) | 19 | |
| UART1 reception, ACK (3) | +80 to +83 (0050h to 0053h) | 20 | |
| Timer B0 | +84 to +87 (0054h to 0057h) | 21 | 16.2 “Timer B” |
| Timer B1 | +88 to +91 (0058h to 005Bh) | 22 | |
| Timer B2 | +92 to +95 (005Ch to 005Fh) | 23 | |
| Timer B3 | +96 to +99 (0060h to 0063h) | 24 | |
| Timer B4 | +100 to +103 (0064h to 0067h) | 25 | |
| INT5 | +104 to +107 (0068h to 006Bh) | 26 | 11.10 “External Interrupt” |
| INT4 | +108 to +111 (006Ch to 006Fh) | 27 | |
| INT3 | +112 to +115 (0070h to 0073h) | 28 | |
| INT2 | +116 to +119 (0074h to 0077h) | 29 | |
| INT1 | +120 to +123 (0078h to 007Bh) | 30 | |
| INT0 | +124 to +127 (007Ch to 007Fh) | 31 | |
| Timer B5 | +128 to +131 (0080h to 0083h) | 32 | 16.2 “Timer B” |

Notes:
1. Each entry is relative to the base address in the INTB register.
2. Interrupts from this source cannot be disabled by the I flag.
3. In $I^2C$ mode, interrupts are generated by NACK, ACK, or detection of start condition/stop condition.
4. The IFSR16 bit in the IFSR1 register selects either the interrupt source in UART5 or that in UART6.

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

| Interrupt Source | Vector Table Relative Addresses (Address (L) to Address (H)) (1) | Software Interrupt Number | Reference |
|---|---|---|---|
| UART2 transmission, NACK (2)/I2C-bus interface (3) | +132 to +135 (0084h to 0087h) | 33 | 18. “Serial Interface”/24. “Multi-master I2C-bus Interface” |
| UART2 reception, ACK (2)/I2C-bus line (3) | +136 to +139 (0088h to 008Bh) | 34 | |
| UART3 transmission, NACK (2) | +140 to +143 (008Ch to 008Fh) | 35 | |
| UART3 reception, ACK (2) | +144 to +147 (0090h to 0093h) | 36 | |
| UART4 transmission, NACK (2) | +148 to +151 (0094h to 0097h) | 37 | |
| UART4 reception, ACK (2) | +152 to +155 (0098h to 009Bh) | 38 | |
| Bus collision detection, start condition detection, or stop condition detection (UART2) (2) | +156 to +159 (009Ch to 009Fh) | 39 | |
| Bus collision detection, start condition detection, or stop condition detection (UART3 or UART0) (2, 4) | +160 to +163 (00A0h to 00A3h) | 40 | |
| Bus collision detection, start condition detection, or stop condition detection (UART4 or UART1) (2, 4) | +164 to +167 (00A4h to 00A7h) | 41 | |
| A/D0 | +168 to +171 (00A8h to 00ABh) | 42 | 19. “A/D Converter” |
| Key input | +172 to +175 (00ACh to 00AFh) | 43 | 11.12 “Key Input Interrupt” |
| Intelligent I/O interrupt 0 | +176 to +179 (00B0h to 00B3h) | 44 | 11.13 “Intelligent I/O Interrupt”, 23. “Intelligent I/O” |
| Intelligent I/O interrupt 1 | +180 to +183 (00B4h to 00B7h) | 45 | |
| Intelligent I/O interrupt 2 | +184 to +187 (00B8h to 00BBh) | 46 | |
| Intelligent I/O interrupt 3 | +188 to +191 (00BCh to 00BFh) | 47 | |
| Intelligent I/O interrupt 4 | +192 to +195 (00C0h to 00C3h) | 48 | |
| Intelligent I/O interrupt 5 | +196 to +199 (00C4h to 00C7h) | 49 | |
| Intelligent I/O interrupt 6 | +200 to +203 (00C8h to 00CBh) | 50 | |
| Intelligent I/O interrupt 7 | +204 to +207 (00CCh to 00CFh) | 51 | |
| Intelligent I/O interrupt 8 | +208 to +211 (00D0h to 00D3h) | 52 | |
| Intelligent I/O interrupt 9 | +212 to +215 (00D4h to 00D7h) | 53 | |
| Intelligent I/O interrupt 10 | +216 to +219 (00D8h to 00DBh) | 54 | |
| Intelligent I/O interrupt 11 | +220 to +223 (00DCh to 00DFh) | 55 | |
| Reserved | +224 to +227 (00E0h to 00E3h) | 56 | |
| Reserved | +228 to +231 (00E4h to 00E7h) | 57 | |
| CAN0 wakeup | +232 to +235 (00E8h to 00EBh) | 58 | 25. “CAN Module” |
| CAN1 wakeup | +236 to +239 (00ECh to 00EFh) | 59 | |
| Reserved | +240 to +243 (00F0h to 00F3h) | 60 | |
| Reserved | +244 to +247 (00F4h to 00F7h) | 61 | |
| Reserved | +248 to +251 (00F8h to 00FBh) | 62 | |
| Reserved | +252 to +255 (00FCh to 00FFh) | 63 | |

Notes:
1. Each entry is relative to the base address in the INTB register.
2. In I2C mode, interrupts are generated by NACK, ACK, or detection of start condition/stop condition.
3. Select an interrupt source either of UART2 or I2C-bus interface by using the I2CEN bit in the I2CMR register.
4. The IFSR06 bit in the IFSR0 register selects either the interrupt source in UART0 or that in UART3. The IFSR07 bit selects either the interrupt source in UART1 or that in UART4.

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

| Interrupt Source | Vector Table Relative Addresses (Address (L) to Address (H)) (2) | Software Interrupt Number(s) | Reference |
|---|---|---|---|
| Reserved | +256 to +259 (0100h to 0103h) | 64 | |
| Reserved | +260 to +263 (0104h to 0107h) | 65 | |
| Reserved | +264 to +267 (0108h to 010Bh) | 66 | |
| Reserved | +268 to +271 (010Ch to 010Fh) | 67 | |
| Reserved | +272 to +275 (0110h to 0113h) | 68 | |
| Reserved | +276 to +279 (0114h to 0117h) | 69 | |
| Reserved | +280 to +283 (0118h to 011Bh) | 70 | |
| Reserved | +284 to +287 (011Ch to 011Fh) | 71 | |
| Reserved | +288 to +291 (0120h to 0123h) | 72 | |
| Reserved | +292 to +295 (0124h to 0127h) | 73 | |
| Reserved | +296 to +299 (0128h to 012Bh) | 74 | |
| Reserved | +300 to +303 (012Ch to 012Fh) | 75 | |
| Reserved | +304 to +307 (0130h to 0133h) | 76 | |
| Reserved | +308 to +311 (0134h to 0137h) | 77 | |
| Reserved | +312 to +315 (0138h to 013Bh) | 78 | |
| Reserved | +316 to +319 (013Ch to 013Fh) | 79 | |
| CAN0 transmit FIFO | +320 to +323 (0140h to 0143h) | 80 | 25. “CAN Module” |
| CAN0 receive FIFO | +324 to +327 (0144h to 0147h) | 81 | |
| CAN1 transmit FIFO | +328 to +331 (0148h to 014Bh) | 82 | |
| CAN1 receive FIFO | +332 to +335 (014Ch to 014Fh) | 83 | |
| Reserved | +336 to +339 (0150h to 0153h) | 84 | |
| Reserved | +340 to +343 (0154h to 0157h) | 85 | |
| Reserved | +344 to +347 (0158h to 015Bh) | 86 | |
| Reserved | +348 to +351 (015Ch to 015Fh) | 87 | |
| Reserved | +352 to +355 (0160h to 0163h) | 88 | |
| Reserved | +356 to +359 (0164h to 0167h) | 89 | |
| Reserved | +360 to +363 (0168h to 016Bh) | 90 | |
| Reserved | +364 to +367 (016Ch to 016Fh) | 91 | |
| Reserved | +368 to +371 (0170h to 0173h) | 92 | |
| INT8 | +372 to +375 (0174h to 0177h) | 93 | 11.10 “External Interrupt” |
| INT7 | +376 to +379 (0178h to 017Bh) | 94 | |
| INT6 | +380 to +383 (017Ch to 017Fh) | 95 | |

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

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

| Interrupt Source | Vector Table Relative Addresses (Address (L) to Address (H)) (2) | Software Interrupt Number(s) | Reference |
|---|---|---|---|
| CAN0 transmission | +384 to +387 (0180h to 0183h) | 96 | 25. “CAN Module” |
| CAN0 reception | +388 to +391 (0184h to 0187h) | 97 | |
| CAN0 error | +392 to +395 (0188h to 018Bh) | 98 | |
| CAN1 transmission | +396 to +399 (018Ch to 018Fh) | 99 | |
| CAN1 reception | +400 to +403 (0190h to 0193h) | 100 | |
| CAN1 error | +404 to +407 (0194h to 0197h) | 101 | |
| Reserved | +408 to +411 (0198h to 019Bh) | 102 | |
| Reserved | +412 to +415 (019Ch to 019Fh) | 103 | |
| Reserved | +416 to +419 (01A0h to 01A3h) | 104 | |
| Reserved | +420 to +423 (01A4h to 01A7h) | 105 | |
| Reserved | +424 to +427 (01A8h to 01ABh) | 106 | |
| Reserved | +428 to +431 (01ACh to 01AFh) | 107 | |
| Reserved | +432 to +435 (01B0h to 01B3h) | 108 | |
| Reserved | +436 to +439 (01B4h to 01B7h) | 109 | |
| Reserved | +440 to +443 (01B8h to 01BBh) | 110 | |
| Reserved | +444 to +447 (01BCh to 01BFh) | 111 | |
| Reserved | +448 to +451 (01C0h to 01C3h) | 112 | |
| Reserved | +452 to +455 (01C4h to 01C7h) | 113 | |
| Reserved | +456 to +459 (01C8h to 01CBh) | 114 | |
| Reserved | +460 to +463 (01CCh to 01CFh) | 115 | |
| Reserved | +464 to +467 (01D0h to 01D3h) | 116 | |
| Reserved | +468 to +471 (01D4h to 01D7h) | 117 | |
| Reserved | +472 to +475 (01D8h to 01DBh) | 118 | |
| Reserved | +476 to +479 (01DCh to 01DFh) | 119 | |
| Reserved | +480 to +483 (01E0h to 01E3h) | 120 | |
| Reserved | +484 to +487 (01E4h to 01E7h) | 121 | |
| Reserved | +488 to +491 (01E8h to 01EBh) | 122 | |
| Reserved | +492 to +495 (01ECh to 01EFh) | 123 | |
| UART7 transmission | +496 to +499 (01F0h to 01F3h) | 124 | 18. “Serial Interface” |
| UART7 reception | +500 to +503 (01F4h to 01F7h) | 125 | |
| UART8 transmission | +504 to +507 (01F8h to 01FBh) | 126 | |
| UART8 reception | +508 to +511 (01FCh to 01FFh) | 127 | |
| INT instruction (3) | +0 to +3 (0000h to 0003h) to +1020 to +1023 (03FCh to 03FFh) | 0 to 255 | 11.2 “Software Interrupt” |

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

## 11.6 Interrupt Request Acceptance

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

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

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

### 11.6.1 I Flag and IPL

The I flag (interrupt enable flag) enables or disables maskable interrupts. When the I flag is set to 1 (enabled), all maskable interrupts are enabled; when it is set to 0 (disabled), they are disabled. The I flag is automatically set to 0 after a reset.
The IPL (processor interrupt priority level), consisting of three bits, indicates eight interrupt priority levels from 0 to 7. An interrupt becomes acceptable when its interrupt request level is higher than the specified IPL (bits ILVL2 to ILVL0 > IPL).
Table 11.6 lists interrupt request levels classified by the IPL.

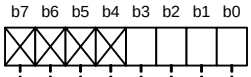
**Table 11.6 Acceptable Interrupt Request Levels and IPL**

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

### 11.6.2 Interrupt Control Register

The interrupt control registers control each peripheral interrupt.
Figure 11.3 and Figure 11.4 show the interrupt control registers.

Interrupt Control Register



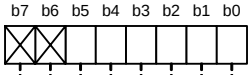

| Symbol | Address | Reset Value |
|---|---|---|
| TA0IC to TA4IC | 006Ch, 008Ch, 006Eh, 008Eh, 0070h | XXXX X000b |
| TB0IC to TB5IC | 0094h, 0076h, 0096h, 0078h, 0098h, 0061h | XXXX X000b |
| S0TIC to S4TIC | 0090h, 0092h, 0081h (1), 0083h, 0085h | XXXX X000b |
| S5TIC to S8TIC | 0062h, 0064h, 00DDh, 00DFh | XXXX X000b |
| S0RIC to S4RIC | 0072h, 0074h, 0063h (2), 0065h, 0067h | XXXX X000b |
| S5RIC to S8RIC | 0082h, 0084h, 00FDh, 00FFh | XXXX X000b |
| BCN0IC to BCN3IC | 0069h, 0089h, 0087h, 0069h (3) | XXXX X000b |
| BCN4IC to BCN6IC | 0089h (4), 0066h, 0066h (5) | XXXX X000b |
| DM0IC to DM3IC | 0068h, 0088h, 006Ah, 008Ah | XXXX X000b |
| AD0IC | 006Bh | XXXX X000b |
| KUPIC | 008Bh | XXXX X000b |
| IIO0IC to IIO5IC | 006Dh, 008Dh, 006Fh, 008Fh, 0071h, 0091h | XXXX X000b |
| IIO6IC to IIO11IC | 0073h, 0093h, 0075h, 0095h, 0077h, 0097h | XXXX X000b |
| I2CIC, I2CLIC | 0081h (1), 0063h (2) | XXXX X000b |
| C0FTIC, C1FTIC | 00D0h, 00D2h | XXXX X000b |
| C0FRIC, C1FRIC | 00F0h, 00F2h | XXXX X000b |
| C0TIC, C1TIC | 00C1h, 00E3h | XXXX X000b |
| C0RIC, C1RIC | 00E1h, 00C5h | XXXX X000b |
| C0EIC, C1EIC | 00C3h, 00E5h | XXXX X000b |
| C0WIC, C1WIC | 007Bh, 009Bh | XXXX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| ILVL0 | Interrupt Request Level Select Bit | b2 b1 b0<br>0 0 0 : Level 0 (interrupt disabled)<br>0 0 1 : Level 1<br>0 1 0 : Level 2<br>0 1 1 : Level 3<br>1 0 0 : Level 4<br>1 0 1 : Level 5<br>1 1 0 : Level 6<br>1 1 1 : Level 7 | RW |
| ILVL1 | | | RW |
| ILVL2 | | | RW |
| IR | Interrupt Request Flag | 0: No interrupt requested<br>1: Interrupt requested (6) | RW |
| —<br>(b7-b4) | No register bits; should be written with 0 and read as undefined value | | — |

Notes:
1. The S2TIC register shares an address with the I2CIC register.
2. The S2RIC register shares an address with the I2CLIC register.
3. The BCN0IC register shares an address with the BCN3IC register.
4. The BCN1IC register shares an address with the BCN4IC register.
5. The BCN5IC register shares an address with the BCN6IC register.
6. This bit can be set to 0 only (It should not be set to 1).

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

Interrupt Control Register




| Symbol | Address | Reset Value |
|---|---|---|
| INT0IC to INT2IC | 009Eh, 007Eh, 009Ch | XX00 X000b |
| INT3IC to INT5IC (1) | 007Ch, 009Ah, 007Ah | XX00 X000b |
| INT6IC to INT8IC | 00FEh, 00DEh, 00FCh | XX00 X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| ILVL0 | Interrupt Request Level Select Bit | b2 b1 b0<br>0 0 0 : Level 0 (interrupt disabled)<br>0 0 1 : Level 1<br>0 1 0 : Level 2<br>0 1 1 : Level 3<br>1 0 0 : Level 4<br>1 0 1 : Level 5<br>1 1 0 : Level 6<br>1 1 1 : Level 7 | RW |
| ILVL1 | | | RW |
| ILVL2 | | | RW |
| IR | Interrupt Request Flag | 0: No interrupt requested<br>1: Interrupt requested (2) | RW |
| POL | Polarity Select Bit | 0: Select the falling edge or a low<br>1: Select the rising edge or a high (3) | RW |
| LVS | Level/Edge Sensitive Select Bit | 0: Edge sensitive<br>1: Level sensitive (4) | RW |
| —<br>(b7-b6) | No register bits; should be written with 0 and read as undefined value | | — |

Notes:
1. When the 16- or 32-bit data bus is used in microprocessor mode or memory expansion mode, pins INT3 to INT5 function as data bus. Bits ILVL2 to ILVL0 in registers INT3IC to INT5IC should be set to 000b.
2. This bit can be set to 0 only (it should not be set to 1).
3. This bit should be set to 0 (the falling edge or low level) to set the corresponding bit in registers IFSR0 and IFSR1 to 1 (both edges).
4. To select the level sensitive, the corresponding bit in registers IFSR0 and IFSR1 should be set to 0 (one edge).

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

Bits ILVL2 to ILVL0

Bits ILVL2 to ILVL0 select the interrupt request level. The higher the level is, the higher interrupt priority is.

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

IR bit

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

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

When rewriting the interrupt control register, no corresponding interrupt request should be generated. If it may be generated, disable all the maskable interrupts before the rewrite.
When enabling the maskable interrupts immediately after the rewrite, there should be sufficient time for the rewrite to complete before the interrupt enable flag (I flag) becomes 1. To delay the execution of the second instruction, insert NOPs or perform a dummy read of the interrupt control register after the first instruction.
If an interrupt request is generated for a register being rewritten, the IR bit may not become 1 depending on the instruction being used. If this is not desired, use one of the following instructions to rewrite the register:

- AND
- OR
- BCLR
- BSET

When setting the IR bit to 0 by the AND or BCLR instruction, the IR bit may not become 0. This is because an interrupt request generated while the instruction above is being executed is kept pending. If this is not desired, the register should be reconfigured by the MOV instruction. To set just the IR bit to 0, first temporarily store the read value to memory or CPU-internal registers, then execute either the AND or BCLR instruction in the stored area. After that, write the value back to the register by the MOV instruction.

### 11.6.3 Wake-up IPL Setting Register

The wake-up IPL setting register (registers RIPL1 and RIPL2) is used for an interrupt to exit wait or stop mode, or for the fast interrupt.
Refer to 8.7.2 “Wait Mode”, 8.7.3 “Stop Mode”, or 11.4 “Fast Interrupt” for details.
Figure 11.5 shows registers RIPL1 and RIPL2.

Wake-up IPL Setting Register i (i = 1, 2) (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | X | | X | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| RIPL1, RIPL2 | 4407Fh, 4407Dh | XX0X 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| RLVL0 | Interrupt Priority Level for Wake-up Select Bit (2) | b2 b1 b0<br>0 0 0 : Level 0<br>0 0 1 : Level 1<br>0 1 0 : Level 2<br>0 1 1 : Level 3<br>1 0 0 : Level 4<br>1 0 1 : Level 5<br>1 1 0 : Level 6<br>1 1 1 : Level 7 | RW |
| RLVL1 | | | RW |
| RLVL2 | | | RW |
| FSIT | Fast Interrupt Select Bit (3) | 0: Use interrupt request level 7 for normal interrupt<br>1: Use interrupt request level 7 for fast interrupt (4) | RW |
| — (b4) | No register bit; should be written with 0 and read as undefined value | | — |
| DMAII | DMA II Select Bit (5) | 0: Use interrupt request level 7 for interrupt<br>1: Use interrupt request level 7 for DMA II transfer (4) | RW |
| — (b7-b6) | No register bits; should be written with 0 and read as undefined value | | — |

Notes:
1. Registers RIPL1 and RIPL2 should be identically set.
2. The MCU exits wait mode or stop mode if the request level of requested interrupt is higher than the level selected using bits RLVL2 to RLVL0. These bits should be set to the same value as the IPL in the flag register (FLG).
3. When the FSIT bit is set to 1, an interrupt with interrupt request level 7 becomes the fast interrupt. In this case only one interrupt should be set to the interrupt request level 7.
4. Either the FSIT or DMAII bit should be set to 1. Simultaneous use of the fast interrupt and the DMAC II is not available.
5. Bits ILVL2 to ILVL0 in the interrupt control register should be set after the DMAII bit is set. The DMA II transfer is not affected by the I flag or the IPL.

**Figure 11.5 Registers RIPL1 and RIPL2**

### 11.6.4 Interrupt Sequence

The interrupt sequence is performed from when an interrupt request has been accepted until the interrupt handler starts.
For most instructions, when an interrupt request is generated while an instruction is being executed, the requested interrupt is evaluated in the priority resolver after the current instruction is completed. If appropriate, the interrupt sequence starts from the next cycle.
For instructions RMPA, SCMPU, SIN, SMOVB, SMOVF, SMOVU, SOUT, SSTR, SUNTIL, and SWHILE, as soon as an interrupt request is generated, the requested interrupt is evaluated suspending the current instruction being executed. If appropriate, the interrupt sequence starts immediately.
The interrupt sequence is as follows:

(1) The CPU acknowledges the interrupt request to obtain the interrupt information (the interrupt number, and the interrupt request level) from the interrupt controller. Then the corresponding IR bit becomes 0 (no interrupt requested)
(2) The state of the flag register (FLG) before the interrupt sequence is stored to a temporary register (1) in the CPU.
(3) The following bits in the flag register (FLG) become 0:
- The I flag (interrupt enable flag): interrupt disabled
- The D flag (debug flag): single-step interrupt disabled
- The U flag (stack pointer select flag): ISP selected

(4) The contents of the temporary register (1) in the CPU is saved to the stack; or to the save flag register (SVF) in case of the fast interrupt.
(5) The contents of the program counter (PC) is saved to the stack; or to the save PC register (SVP) in case of the fast interrupt.
(6) The interrupt request level for the accepted interrupt is set in the IPL (processor interrupt priority level).
(7) The corresponding interrupt vector is read from the interrupt vector table.
(8) This interrupt vector is stored into the program counter (PC).

When the interrupt sequence completes, the interrupt handler is initiated.

Note:
1. This register is inaccessible to users.

### 11.6.5 Interrupt Response Time

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

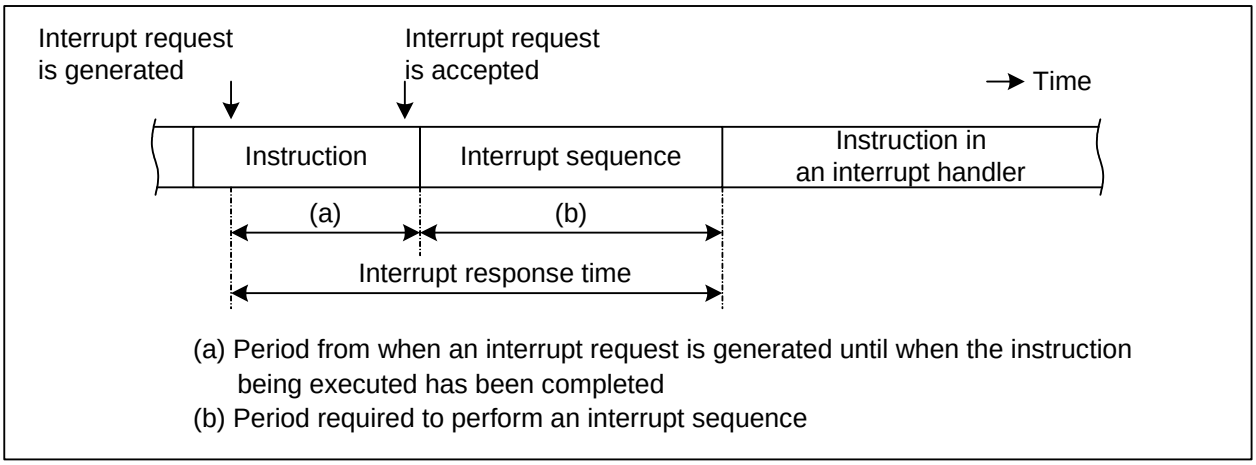



**Figure 11.6 Interrupt Response Time**

Period (a) varies depending on the instruction being executed. Instructions, such as LDCTX and STCTX in which registers are sequentially saved/restored into/from the stack, require the longest time. For example, the STCTX instruction requires at least 30 cycles for ten registers to be saved. It requires more time if the WAIT instruction is in the stack.
Period (b) is listed in Table 11.7.

**Table 11.7 Interrupt Sequence Execution Time (1)**

| Interrupt | Execution Time in Terms of CPU Clock |
|---|---|
| Peripheral | 13 + α cycles (2) |
| INT instruction | 11 cycles |
| NMI | 10 cycles |
| Watchdog timer<br>Oscillator stop detection<br>Low voltage detection | 11 cycles |
| Undefined instruction | 12 cycles |
| Overflow | 12 cycles |
| BRK instruction (relocatable vector table) | 16 cycles |
| BRK instruction (fixed vector table) | 19 cycles |
| BRK2 instruction | 19 cycles |
| Fast interrupt | 11 cycles |

Notes:
1. The interrupt vectors should be aligned in addresses in multiples of 4 of internal ROM. The fast interrupt is independent of this condition.
2. α is the number of waits to access SFR minus 2.

### 11.6.6 IPL After Interrupt Request Acceptance

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

Software interrupts and special interrupts have no interrupt request level. For these interrupt requests, if accepted, the value shown in Table 11.8 is set in the IPL as interrupt request level.

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

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

### 11.6.7 Register Saving

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

In the fast interrupt sequence, the flag register (FLG) and program counter (PC) values are saved to the save flag register (SVF) and save PC register (SVP), respectively.

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

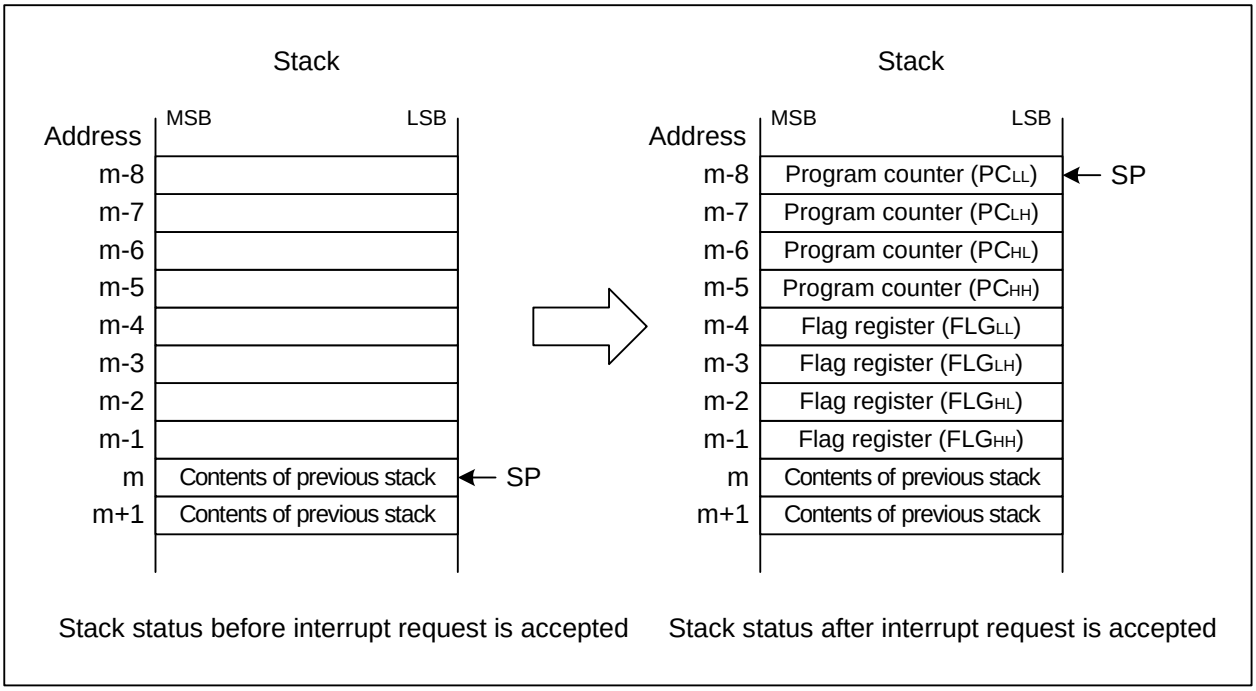



**Figure 11.7 Stack Status Before and After an interrupt Request is Accepted**

## 11.7 Register Restoring from Interrupt Handler

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

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

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

## 11.8 Interrupt Priority

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

For maskable interrupts (peripheral interrupts), the interrupt request level select bits (bits ILVL2 to ILVL0) select a request level. If there are more than two interrupts with the same level, they are accepted according to their relative priority predetermined by the hardware.

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

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

Software interrupts are not governed by priority. They always cause execution to jump to the interrupt handler whenever the relevant instruction is executed.

## 11.9 Priority Resolver

The priority resolver determines which interrupt request has a higher priority if two or more interrupt requests are detected at a sampling point.

Figure 11.8 shows the priority resolver.

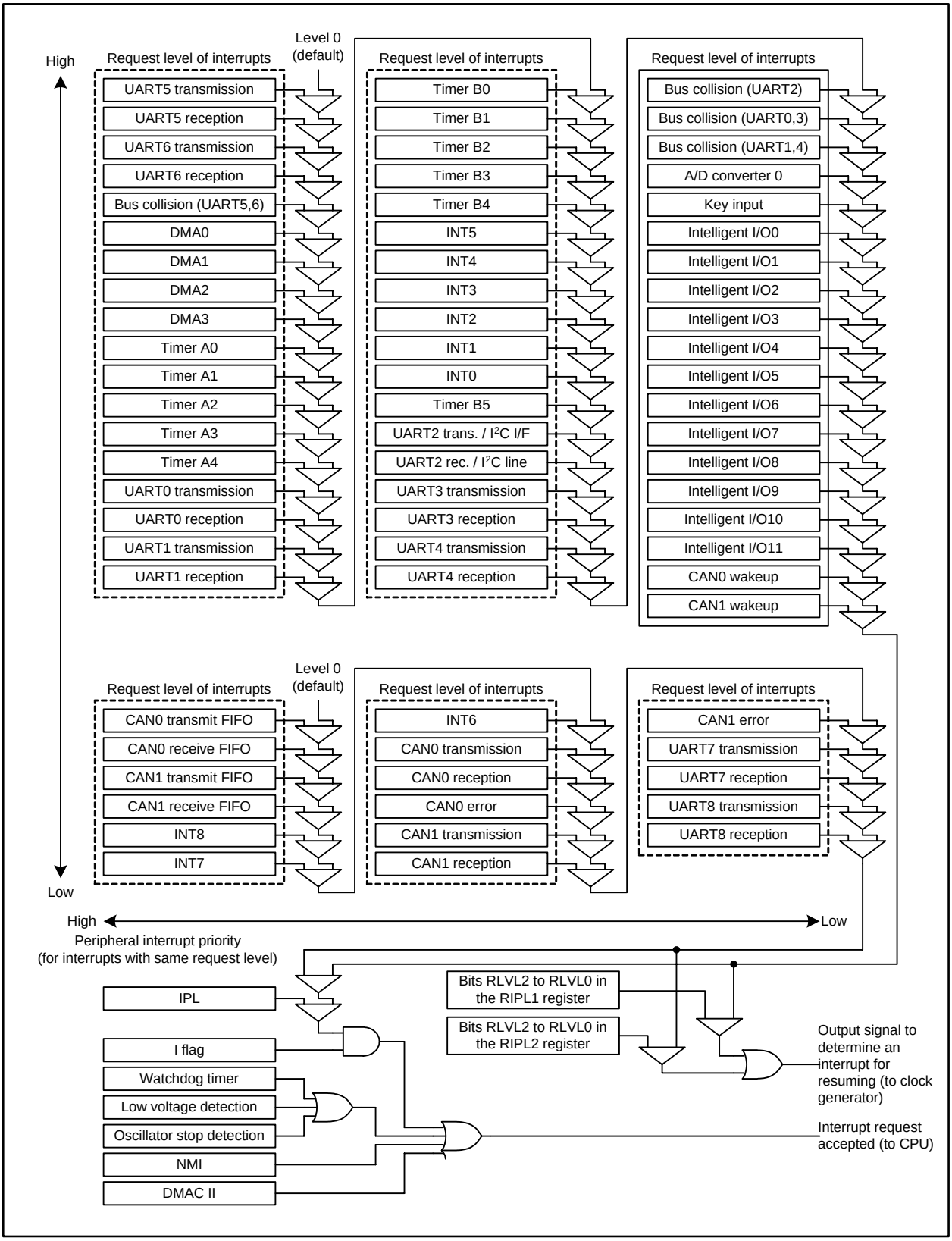



**Figure 11.8 Priority Resolver**

## 11.10 External Interrupt

An external interrupt is generated by an external input applied to the $\overline{INTi}$ pin (i = 0 to 8). The LVS bit in the INTiIC register selects whether an interrupt is triggered by the effective edge(s) (edge sensitive), or by the effective level (level sensitive) of the input signal. The polarity of the input signal is selected by the POL bit in the same register.

When using edge-triggered interrupts, setting the IFSR0j bit in the IFSR0 register to 1 (both edges) causes interrupt requests to be generated on both rising and falling edges of the external input applied to the $\overline{INTj}$ pin (j = 0 to 5). This also applies to setting the IFSR1n bit (n = m - 6) in the IFSR1 register to 1 (both edges) for the $\overline{INTm}$ pin (m = 6 to 8). When the IFSR0j bit or the IFSR1n bit is set to 1, the POL bit in the corresponding register should be set to 0 (falling edge).

When using level-triggered interrupts, set the IFSR0j or IFSR1n to 0 (one edge). When an effective level, which is selected by the POL bit, is detected on the $\overline{INTi}$ pin, the IR bit in the INTiIC register becomes 1. The IR bit remains unchanged until the INTi interrupt is accepted, or it is set to 0 by a program, even if the signal level at the $\overline{INTi}$ pin changes.

Figure 11.9 and Figure 11.10 show registers IFSR0 and IFSR1, respectively.

External Interrupt Request Source Select Register 0

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

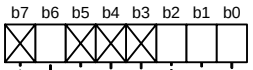
| Symbol | Address | Reset Value |
|---|---|---|
| IFSR0 | 4406Fh | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IFSR00 | $\overline{INT0}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR01 | $\overline{INT1}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR02 | $\overline{INT2}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR03 | $\overline{INT3}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR04 | $\overline{INT4}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR05 | $\overline{INT5}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR06 | UART0/UART3 Interrupt Source Select Bit | 0: Bus collision, start condition detection, stop condition detection in UART3<br>1: Bus collision, start condition detection, stop condition detection in UART0 | RW |
| IFSR07 | UART1/UART4 Interrupt Source Select Bit | 0: Bus collision, start condition detection, stop condition detection in UART4<br>1: Bus collision, start condition detection, stop condition detection in UART1 | RW |

Note:
1. This bit should be set to 0 to select the level sensitive input as trigger. To set this bit to 1, the POL bit in the corresponding INTiIC register (i = 0 to 5) should be set to 0 (falling edge).

**Figure 11.9 IFSR0 Register**

External Interrupt Request Source Select Register 1




Symbol: IFSR1 | Address: 4406Dh | Reset Value: X0XX X000b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IFSR10 | $\overline{\text{INT6}}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR11 | $\overline{\text{INT7}}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR12 | $\overline{\text{INT8}}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| —<br>(b5-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| IFSR16 | UART5/UART6 Interrupt Source Select Bit | 0: Bus collision, start condition detection, stop condition detection in UART5<br>1: Bus collision, start condition detection, stop condition detection in UART6 | RW |
| —<br>(b7) | No register bit; should be written with 0 and read as undefined value | | — |

Note:
1. This bit should be set to 0 to select the level sensitive input as trigger. To set this bit to 1, the POL bit in the corresponding INTiIC register (i = 6 to 8) should be set to 0 (falling edge).

**Figure 11.10 IFSR1 Register**

## 11.11 NMI

The NMI (Non Maskable Interrupt) occurs when an input signal at the $\overline{\text{NMI}}$ pin switches from high to low. This non maskable interrupt is disabled after a reset. To enable this interrupt, the PM24 bit in the PM2 register should be set to 1 after setting the interrupt stack pointer (ISP) at the beginning of the program. The $\overline{\text{NMI}}$ pin shares a pin with the port P8_5, which enables the P8_5 bit in the P8 register to indicate the input level at the $\overline{\text{NMI}}$ pin.

Note:
1. When not using the NMI, hold 0 as reset value of the PM24 bit in the PM2 register.

## 11.12 Key Input Interrupt

The key input interrupt is enabled by setting ports P10_4 to P10_7 as input ports.

The interrupt request is generated if any of the signals applied to ports P10_4 to P10_7 switches from high to low. This interrupt also functions as key wake-up to exit wait or stop mode. Figure 11.11 shows a block diagram of the key input interrupt. If any of the ports is held low, signals applied to other ports are not detected as interrupt request signals.

To use the key input interrupt, every register from P10_4S to P10_7S should be set to 00h (I/O port) and bits PD10_4 to PD10_7 should be set to 0 (input). This is the only setting available for the key input interrupt.

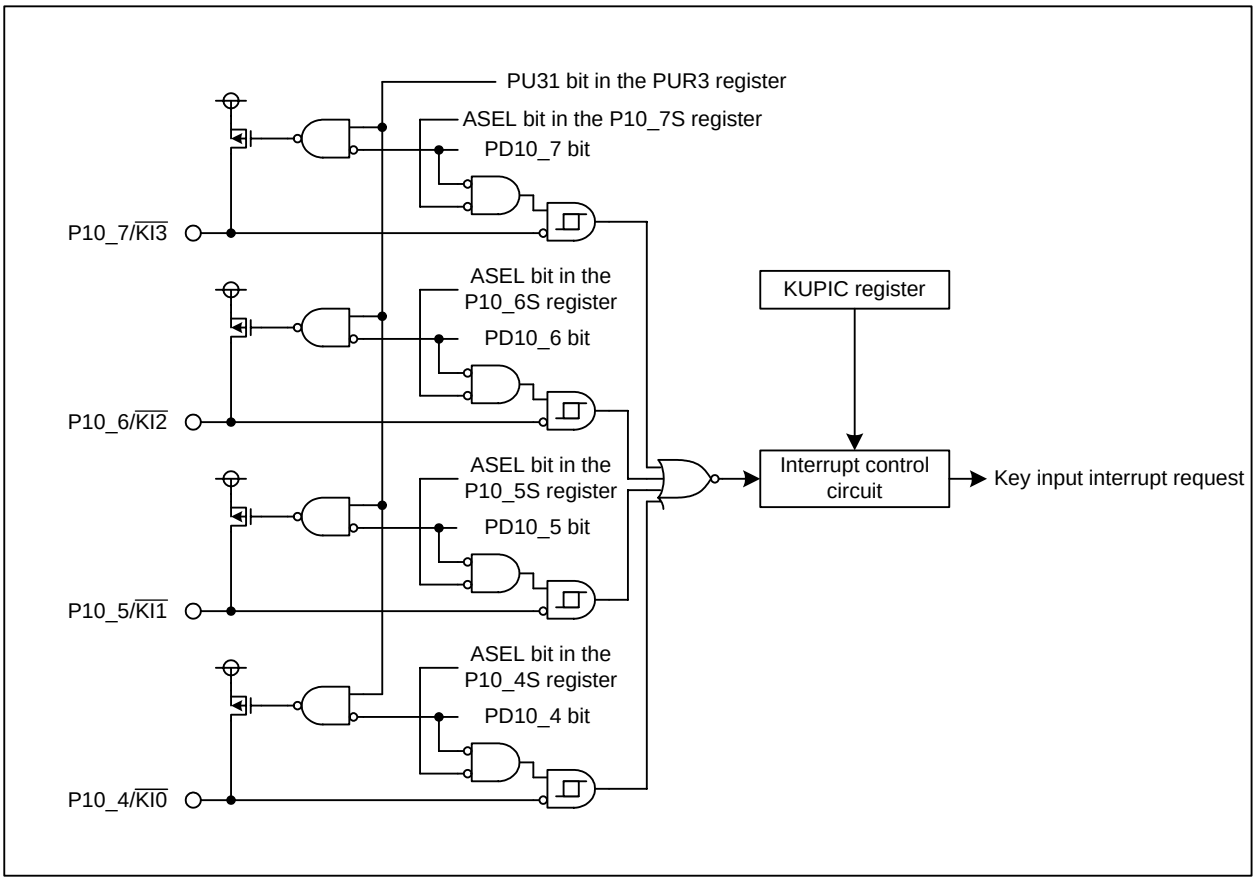



**Figure 11.11 Key Input Interrupt**

## 11.13 Intelligent I/O Interrupt

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

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

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

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

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

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

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

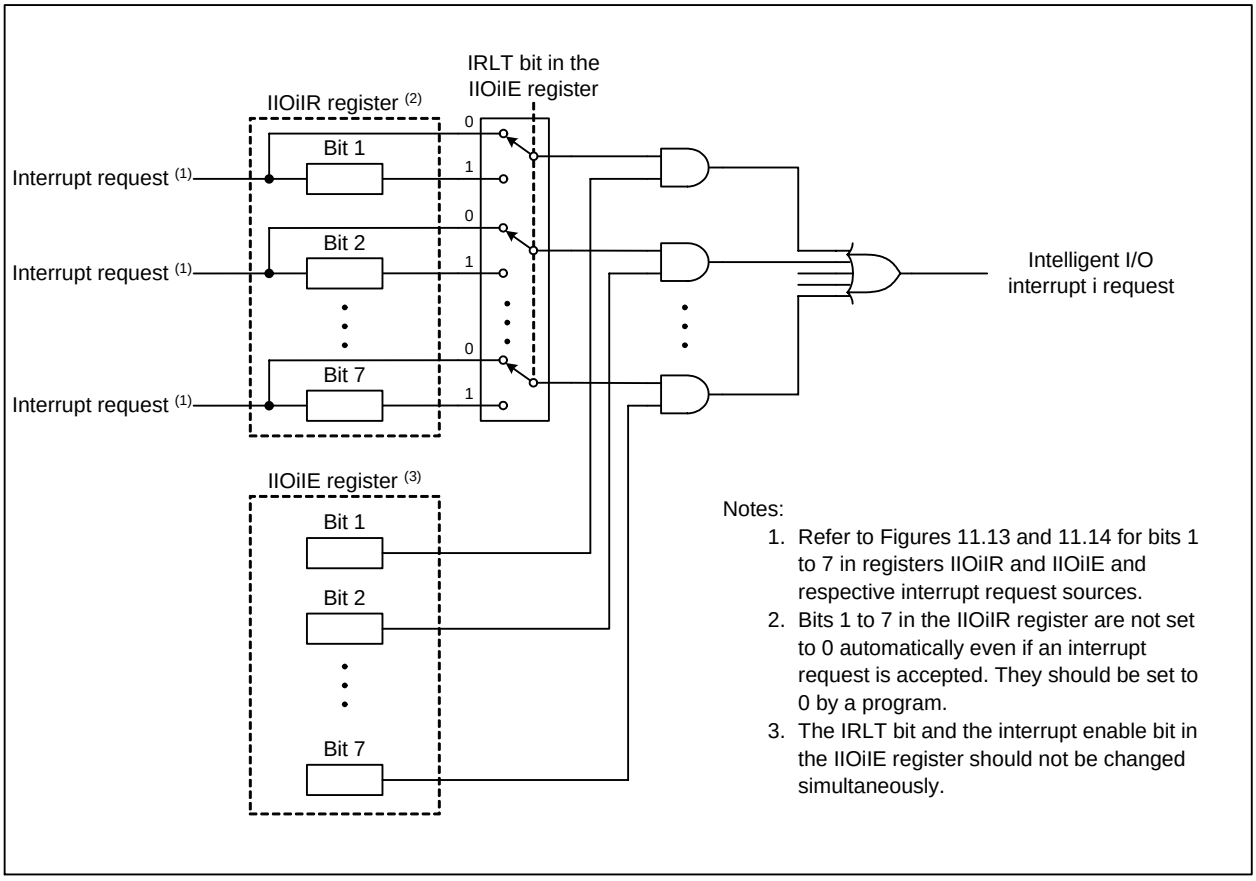
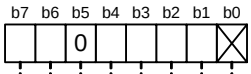



**Figure 11.12 Intelligent I/O Interrupt Block Diagram (i = 0 to 11)**

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




Symbol: IIO0IR to IIO11IR

Address: Refer to the table below

Reset Value: ??0? ???1b (1)

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b0) | No register bit; should be written with 0 and read as 1 | | — |
| (Note 2) | 0: No interrupt requested<br>1: Interrupt requested (3) | | RW |
| (Note 2) | 0: No interrupt requested<br>1: Interrupt requested (3) | | RW |
| (Note 2) | 0: No interrupt requested<br>1: Interrupt requested (3) | | RW |
| (Note 2) | 0: No interrupt requested<br>1: Interrupt requested (3) | | RW |
| — (b5) | Reserved | Should be written with 0 | RW |
| (Note 2) | 0: No interrupt requested<br>1: Interrupt requested (3) | | RW |
| (Note 2) | 0: No interrupt requested<br>1: Interrupt requested (3) | | RW |

Notes:

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

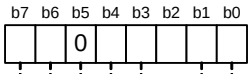
Bit Symbols for the Intelligent I/O Interrupt Request Register

| Symbol | Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| IIO0IR | 00A0h | — | — | — | — | — | TM13R/PO13R | TM02R/PO02R | — |
| IIO1IR | 00A1h | — | — | — | — | — | TM14R/PO14R | TM00R/PO00R | — |
| IIO2IR | 00A2h | — | — | — | — | — | TM12R/PO12R | — | — |
| IIO3IR | 00A3h | — | — | — | — | PO27R | TM10R/PO10R | TM03R/PO03R | — |
| IIO4IR | 00A4h | — | — | — | BT1R | — | TM17R/PO17R | TM04R/PO04R | — |
| IIO5IR | 00A5h | — | — | — | SIO2RR | — | PO21R | TM05R/PO05R | — |
| IIO6IR | 00A6h | — | — | — | SIO2TR | — | PO20R | TM06R/PO06R | — |
| IIO7IR | 00A7h | IE0R | — | — | BT0R | — | PO22R | TM07R/PO07R | — |
| IIO8IR | 00A8h | IE1R | IE2R | — | BT2R | — | PO23R | TM11R/PO11R | — |
| IIO9IR | 00A9h | — | INT6R | — | — | — | PO24R | TM15R/PO15R | — |
| IIO10IR | 00AAh | — | INT7R | — | — | — | PO25R | TM16R/PO16R | — |
| IIO11IR | 00ABh | — | INT8R | — | — | — | PO26R | TM01R/PO01R | — |

BTxR : Intelligent I/O group x base timer interrupt request (x = 0 to 2)
TMxyR : Intelligent I/O group x time measurement channel y interrupt request (x = 0, 1; y = 0 to 7)
POxyR : Intelligent I/O group x waveform generation channel y interrupt request (x = 0 to 2; y = 0 to 7)
IEzR : Intelligent I/O group 2 IEBus interrupt request (z = 0 to 2)
SIO2RR : Intelligent I/O group 2 receive interrupt request
SIO2TR : Intelligent I/O group 2 transmit interrupt request
INTmR : INTm interrupt request (m = 6 to 8)

**Figure 11.13 Registers IIO0IR to IIO11IR**

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




| Symbol | Address | Reset Value |
|---|---|---|
| IIO0IE to IIO11IE | Refer to the table below. | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IRLT | Interrupt Request Select Bit (2) | 0: Use interrupt requests for DMA or DMA II<br>1: Use interrupt requests for interrupt | RW |
| (Note 1) | 0: Disable the interrupt of bit 1 in the IIOiIR register<br>1: Enable the interrupt of bit 1 in the IIOiIR register | | RW |
| (Note 1) | 0: Disable the interrupt of bit 2 in the IIOiIR register<br>1: Enable the interrupt of bit 2 in the IIOiIR register | | RW |
| (Note 1) | 0: Disable the interrupt of bit 3 in the IIOiIR register<br>1: Enable the interrupt of bit 3 in the IIOiIR register | | RW |
| (Note 1) | 0: Disable the interrupt of bit 4 in the IIOiIR register<br>1: Enable the interrupt of bit 4 in the IIOiIR register | | RW |
| —<br>(b5) | Reserved | Should be written with 0 | RW |
| (Note 1) | 0: Disable the interrupt of bit 6 in the IIOiIR register<br>1: Enable the interrupt of bit 6 in the IIOiIR register | | RW |
| (Note 1) | 0: Disable the interrupt of bit 7 in the IIOiIR register<br>1: Enable the interrupt of bit 7 in the IIOiIR register | | RW |

Notes:
1. Refer to the table below for bit symbols.
2. To use interrupt requests for interrupt, the IRLT bit should be set to 1, then bits 1 to 4, 6, and 7 should be set to 1.

Bit Symbols for the Intelligent I/O Interrupt Enable Register

| Symbol | Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| IIO0IE | 00B0h | — | — | — | — | — | TM13E/PO13E | TM02E/PO02E | IRLT |
| IIO1IE | 00B1h | — | — | — | — | — | TM14E/PO14E | TM00E/PO00E | IRLT |
| IIO2IE | 00B2h | — | — | — | — | — | TM12E/PO12E | — | IRLT |
| IIO3IE | 00B3h | — | — | — | — | PO27E | TM10E/PO10E | TM03E/PO03E | IRLT |
| IIO4IE | 00B4h | — | — | — | BT1E | — | TM17E/PO17E | TM04E/PO04E | IRLT |
| IIO5IE | 00B5h | — | — | — | SIO2RE | — | PO21E | TM05E/PO05E | IRLT |
| IIO6IE | 00B6h | — | — | — | SIO2TE | — | PO20E | TM06E/PO06E | IRLT |
| IIO7IE | 00B7h | IE0E | — | — | BT0E | — | PO22E | TM07E/PO07E | IRLT |
| IIO8IE | 00B8h | IE1E | IE2E | — | BT2E | — | PO23E | TM11E/PO11E | IRLT |
| IIO9IE | 00B9h | — | INT6E | — | — | — | PO24E | TM15E/PO15E | IRLT |
| IIO10IE | 00BAh | — | INT7E | — | — | — | PO25E | TM16E/PO16E | IRLT |
| IIO11IE | 00BBh | — | INT8E | — | — | — | PO26E | TM01E/PO01E | IRLT |

BTxE : Intelligent I/O group x base timer interrupt enabled (x = 0 to 2)
TMxyE : Intelligent I/O group x time measurement channel y interrupt enabled (x = 0, 1; y = 0 to 7)
POxyE : Intelligent I/O group x waveform generation channel y interrupt enabled (x = 0 to 2; y = 0 to 7)
IEzE : Intelligent I/O group 2 IEBus interrupt enabled (z = 0 to 2)
SIO2RE : Intelligent I/O group 2 receive interrupt enabled
SIO2TE : Intelligent I/O group 2 transmit interrupt enabled
INTmE : INTm interrupt enabled (m = 6 to 8)

**Figure 11.14 Registers IIO0IE to IIO11IE**

## 11.14 Notes on Interrupts

### 11.14.1 ISP Setting

The interrupt stack pointer (ISP) is initialized to 00000000h after a reset. Set a value to the ISP before an interrupt is accepted, otherwise the program may go out of control. A multiple of 4 should be set to the ISP, which enables faster interrupt sequence due to less memory access.
For the use of NMI, in particular, since this interrupt cannot be disabled, the PM24 bit in the PM2 register should be set to 1 (NMI enabled) after the ISP is set at the beginning of program.

### 11.14.2 NMI

- The NMI cannot be disabled once the PM24 bit in the PM2 register is set to 1 (NMI enabled). This bit setting should be done only for the use of NMI.

- When the PM24 bit in the PM2 register is set to 1 (NMI enabled), the P8_5 bit in the P8 register is enabled just for monitoring the $\overline{\text{NMI}}$ pin state. It is not enabled as a general port.

### 11.14.3 External Interrupt

- The input signal to the $\overline{\text{INTi}}$ pin (i = 0 to 8) requires the pulse width specified by the electrical characteristics. If a pulse width is narrower than the specification, the external interrupt may not be accepted.

- When the effective level and/or edge of $\overline{\text{INTi}}$ pin (i = 0 to 8) are/is changed by the following bits: bits POL and/or LVS in the INTiIC register, the IFSR0i bit (i = 0 to 5) in the IFSR0 register, and/or the IFSR1j bit (j = i - 6; i = 6 to 8) in the IFSR1 register, the corresponding IR bit may become 1 (interrupt requested). When setting the above mentioned bits, preset bits ILVL2 to ILVL0 in the INTiIC register to 000b (interrupt disabled). After setting the above mentioned bits, set the corresponding IR bit to 0 (no interrupt requested), then set bits ILVL2 to ILVL0.

- When the effective level and/or edge of $\overline{\text{INTi}}$ pin (i = 6 to 8) are/is changed by the following bits: bits POL and/or LVS in the INTiIC register, and/or the IFSR1j bit (j = i - 6) in the IFSR1 register, the INTiR bit in the IIOkIR register (k = 9 to 11) may become 1 (ineterrupt requested). When setting the above mentioned bits, preset the INTiE bit in the IIOkIE register to 0 (interrupt disabled). After setting the above mentioned bits, set the corresponding INTiR bit to 0 (no interrupt requested), then set the INTiE bit to 1.

# 12. Watchdog Timer

The watchdog timer monitors program executions and detects defective programs. The 15-bit watchdog counter counts downward with the cycle which is the peripheral bus clock frequency divided by the prescaler.
When the watchdog timer underflows, the CM06 bit in CM0 register selects either a watchdog timer interrupt request or a reset. Once the CM06 bit is set to 1 (reset), it cannot be changed to 0 (watchdog timer interrupt) by a program. Only after a reset, it can be set to 0.
The watchdog timer contains a prescaler which is the peripheral bus clock divided by 16 or 128. The divide ratio is selected by setting the WDC7 bit in the WDC register.
The watchdog timer is stopped in wait mode, stop mode, or when the $\overline{\text{HOLD}}$ is driven low. It resumes counting from the value held when the mode or state is exited.

The general formula to calculate a watchdog timer period is:

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

For example, when the peripheral bus clock is 1/2 of 50 MHz-CPU clock and the prescaler has a divide-by-16 operation, the watchdog timer period is approximately 21 ms. Note that marginal errors within one prescaler output cycle may occur in the watchdog timer period.

The watchdog timer is initialized when a write to the WDTS register is performed or when a watchdog timer interrupt request is generated. The prescaler is initialized only when the MCU is reset.
After a reset, both the watchdog timer and the prescaler are stopped. They start counting when a write to the WDTS register is performed.

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

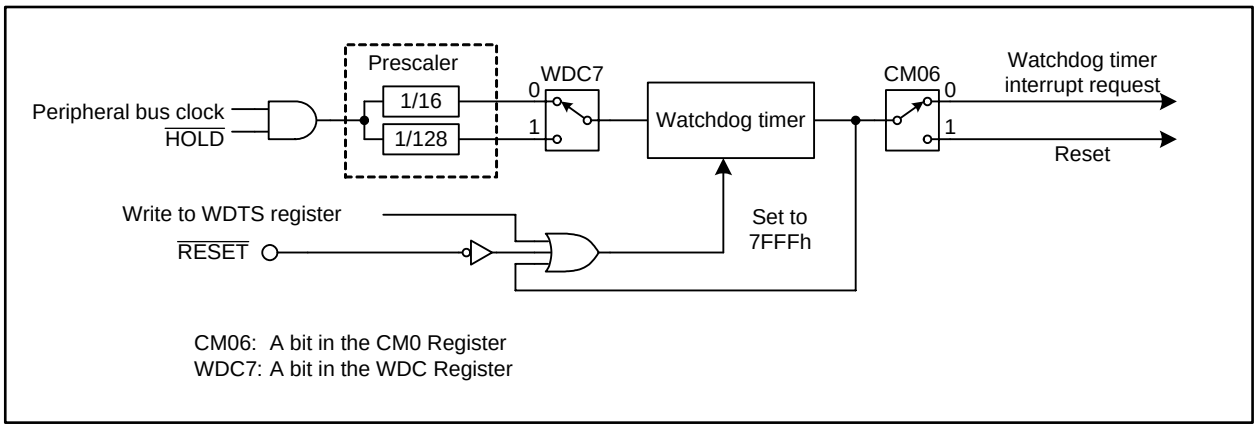



**Figure 12.1 Watchdog Timer Block Diagram**

## Watchdog Timer Control Register

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | 0 | 0 | | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| WDC | 4404Fh | 000X XXXXb |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b4-b0) | Upper five bits of the watchdog timer (b14 to b10) | | RO |
| — (b6-b5) | Reserved | Should be written with 0 | RW |
| WDC7 | Prescaler Select Bit | 0: Divide-by-16<br>1: Divide-by-128 | RW |

**Figure 12.2 WDC Register**

## Watchdog Timer Start Register

| b7 | b0 |
|---|---|
| | |

| Symbol | Address | Reset Value |
|---|---|---|
| WDTS | 4404Eh | Undefined |

| Function | RW |
|---|---|
| The watchdog timer is initialized by a write access. Then it starts counting downward. 7FFFh is set as the default value by writing a given value | WO |

**Figure 12.3 WDTS Register**

# 13. DMAC

Direct Memory Access (DMA) is a system that can control data transfer without using the CPU.
The R32C/100 Series' four channel DMA controller (DMAC) transmits 8-bit (byte), 16-bit (word), or 32-bit (long word) data in cycle-steal mode from a source address to a destination address every time a transfer request is generated.
The DMAC, which shares a data bus with the CPU, has a higher bus access priority than the CPU. This allows the DMAC to perform fast data transfer when a transfer request is generated.
Figure 13.1 shows a map of the CPU-internal registers associated with DMAC. Table 13.1 lists DMAC specifications. Figure 13.2 to Figure 13.10 show registers associated with DMAC. Since the registers shown in Figure 13.1 are allocated in the CPU, the LDC or STC instruction should be used to write to the registers.

DMAC-associated Registers

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

Note:
1. Registers are used for repeat transfer, not for single transfer.

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

**Table 13.1 DMAC Specifications**

| Item | | Specification |
|---|---|---|
| Channels | | 4 |
| Bus request mode | | Cycle-steal mode |
| Transfer memory spaces | | From a given address in a 64-Mbyte space (00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh) to another given address in the same space |
| Maximum transfer bytes | | 64-Mbytes (when 32-bit data is transferred), 32-Mbytes (when 16-bit data is transferred), 16-Mbytes (when 8-bit data is transferred) |
| DMA request sources (1) | | Falling edge or both edges of signals applied to pins INT0 to INT3 or pins $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$<br>Timers A0 to A4 interrupt requests<br>Timers B0 to B5 interrupt requests<br>UART0 to UART8 transmit/receive interrupt requests<br>A/D conversion interrupt requests<br>Intelligent I/O interrupt requests<br>Multi-master $I^2C$-bus interrupt requests<br>Software trigger |
| Channel priority | | DMA0 > DMA1 > DMA2 > DMA3 (DMA0 has the highest priority) |
| Transfer sizes | | 8 bits, 16 bits, or 32 bits |
| Addressing modes | | Incrementing addressing or non-incrementing addressing |
| Transfer modes | Single transfer | Transfer is completed when the DCTi register (i = 0 to 3) is set to 00000000h |
| | Repeat transfer | When the DCTi register is set to 00000000h, the value of the DCRi register is reloaded into the DCTi register to continue the DMA transfer |
| DMA transfer complete interrupt request generation timing | | When the DCTi register changes from 00000001h to 00000000h |
| DMA transfer startup | Single transfer | DMA starts when a DMA transfer request is generated after the DCTi register is set to a value more than 00000001h and bits MDi1 and MDi0 in the DMDi register are set to 01b (single transfer) |
| | Repeat transfer | DMA starts when a DMA transfer request is generated after the DCTi register is set to a value more than 00000001h and bits MDi1 and MDi0 are set to 11b (repeat transfer) |
| DMA transfer stop | Single transfer | DMA stops when bits MDi1 and MDi0 are set to 00b (DMA disabled) |
| | Repeat transfer | DMA stops when bits MDi1 and MDi0 are set to 00b (DMA disabled) |
| Reload timing to DCTi, DSAi or DDAi register | | When the DCTi register changes from 00000001h to 00000000h in repeat transfer mode |
| Minimum DMA transfer cycles | | 3 |

Note:
1. DMA transfer does not affect each interrupt.

The DMA transfer request is available by two different sources: software and hardware. More concretely, they are a write access to the DSR bit in the DMiSL2 register (i = 0 to 3) and an interrupt request output from a function specified in bits DSEL4 to DSEL0 in the DMiSL register, and in bits DSEL24 to DSEL20 in the DMiSL2 register. Unlike interrupt requests, the DMA transfer request is not affected by the I flag nor the interrupt control register. Therefore this request can be accepted even when any interrupt request cannot be because of “interrupt disabled”. Since the DMA transfer does not affect any interrupt, either, the IR bit in the interrupt control register is not changed by the DMA transfer.

DMAi Request Source Select Register (i = 0 to 3)

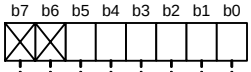

| Symbol | Address | Reset Value |
|---|---|---|
| DM0SL to DM3SL | 44078h, 44079h, 4407Ah, 4407Bh | XXX0 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| DSEL0 | DMA Request Source Select Bit (1) | Refer to Table 13.2 “DMiSL Register (i = 0 to 3) Function” | RW |
| DSEL1 | | | RW |
| DSEL2 | | | RW |
| DSEL3 | | | RW |
| DSEL4 | | | RW |
| — (b7-b5) | No register bits; should be written with 0 and read as undefined value | | — |

Note:
1. The bit settings of bits DSEL4 to DSEL0 should be changed while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

**Figure 13.2 Registers DM0SL to DM3SL**

## DMAi Request Source Select Register 2 (i = 0 to 3)




| Symbol | Address | Reset Value |
|---|---|---|
| DM0SL2 to DM3SL2 | 44070h, 44071h, 44072h, 44073h | XX00 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| DSEL20 | DMA Request Source Select Bit (1) | Refer to Table 13.3 "DMiSL2 Register (i = 0 to 3) Function" | RW |
| DSEL21 | | | RW |
| DSEL22 | | | RW |
| DSEL23 | | | RW |
| DSEL24 | | | RW |
| DSR | Software DMA Transfer Request Bit | When a software trigger is selected, a DMA transfer request is generated by setting this bit to 1 (The bit is read as 0) | RW |
| — (b7-b6) | No register bits; should be written with 0 and read as undefined value. | | — |

Note:
1. The bit settings of bits DSEL24 to DSEL20 should be changed while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

**Figure 13.3 Registers DM0SL2 to DM3SL2**

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

| Setting Value | DMA Request Source | | | |
|---|---|---|---|---|
| b4 b3 b2 b1 b0 | DMA0 | DMA1 | DMA2 | DMA3 |
| 0 0 0 0 0 | Select from DMiSL2 register | | | |
| 0 0 0 0 1 | Falling edge of $\overline{INT0}$ (1) | Falling edge of $\overline{INT1}$ (1) | Falling edge of $\overline{INT2}$ (1) | Falling edge of $\overline{INT3}$ (1, 2) |
| 0 0 0 1 0 | Both edges of $\overline{INT0}$ (1) | Both edges of $\overline{INT1}$ (1) | Both edges of $\overline{INT2}$ (1) | Both edges of $\overline{INT3}$ (1, 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 interrupt request or ACK interrupt request (3) | | | |
| 1 0 0 0 0 | UART1 transmit interrupt request | | | |
| 1 0 0 0 1 | UART1 receive interrupt request or ACK interrupt request (3) | | | |
| 1 0 0 1 0 | UART2 transmit interrupt request or $I^2C$-bus interface interrupt request (4) | | | |
| 1 0 0 1 1 | UART2 receive interrupt request, ACK interrupt request (3), or $I^2C$-bus line interrupt request (4) | | | |
| 1 0 1 0 0 | UART3 transmit interrupt request | | UART5 transmit interrupt request | |
| 1 0 1 0 1 | UART3 receive interrupt request or ACK interrupt request (3) | | UART5 receive interrupt request or ACK interrupt request (3) | |
| 1 0 1 1 0 | UART4 transmit interrupt request | | UART6 transmit interrupt request | |
| 1 0 1 1 1 | UART4 receive interrupt request or ACK interrupt request (3) | | UART6 receive interrupt request or ACK interrupt request (3) | |
| 1 1 0 0 0 | A/D0 interrupt request | | | |
| 1 1 0 0 1 | Intelligent I/O interrupt 0 request | Intelligent I/O interrupt 7 request | Intelligent I/O interrupt 2 request | Intelligent I/O interrupt 9 request |
| 1 1 0 1 0 | Intelligent I/O interrupt 1 request | Intelligent I/O interrupt 8 request | Intelligent I/O interrupt 3 request | Intelligent I/O interrupt 10 request |
| 1 1 0 1 1 | Intelligent I/O interrupt 2 request | Intelligent I/O interrupt 9 request | Intelligent I/O interrupt 4 request | Intelligent I/O interrupt 11 request |
| 1 1 1 0 0 | Intelligent I/O interrupt 3 request | Intelligent I/O interrupt 10 request | Intelligent I/O interrupt 5 request | Intelligent I/O interrupt 0 request |
| 1 1 1 0 1 | Intelligent I/O interrupt 4 request | Intelligent I/O interrupt 11 request | Intelligent I/O interrupt 6 request | Intelligent I/O interrupt 1 request |
| 1 1 1 1 0 | Intelligent I/O interrupt 5 request | Intelligent I/O interrupt 0 request | Intelligent I/O interrupt 7 request | Intelligent I/O interrupt 2 request |
| 1 1 1 1 1 | Intelligent I/O interrupt 6 request | Intelligent I/O interrupt 1 request | Intelligent I/O interrupt 8 request | Intelligent I/O interrupt 3 request |

Notes:

1. The falling edge and both edges of signals applied to the $\overline{INTi}$ pin (i = 0 to 3) triggers a DMA request. The external interrupts (bits POL and LVS in the INTiIC register, the IFSR0 register) are not affected and vice versa.
2. When the $\overline{INT3}$ pin is used for data bus in memory expansion mode or microprocessor mode, it cannot be used for a signal input of DMA3 request source.
3. Registers UiSMR and UiSMR2 (i = 0 to 6) are used to switch between the UARTi receive interrupt and ACK interrupt.
4. Select an interrupt source either of UART2 or $I^2C$-bus interface by using the I2CEN bit in the I2CMR register.

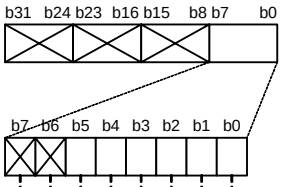
**Table 13.3 DMiSL2 Register (i=0 to 3) Functions**

| 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 $\overline{INT6}$ (1) | Falling edge of $\overline{INT7}$ (1) | Falling edge of $\overline{INT8}$ (1) | Reserved |
| 0 0 0 1 0 | Both edges of $\overline{INT6}$ (1) | Both edges of $\overline{INT7}$ (1) | Both edges of $\overline{INT8}$ (1) | Reserved |
| 0 0 0 1 1 | Reserved | | | |
| 0 0 1 0 0 | Reserved | | | |
| 0 0 1 0 1 | Reserved | | | |
| 0 0 1 1 0 | Reserved | | | |
| 0 0 1 1 1 | Reserved | | | |
| 0 1 0 0 0 | Reserved | | | |
| 0 1 0 0 1 | Reserved | | | |
| 0 1 0 1 0 | Reserved | | | |
| 0 1 0 1 1 | Reserved | | | |
| 0 1 1 0 0 | Reserved | | | |
| 0 1 1 0 1 | Reserved | | | |
| 0 1 1 1 0 | Reserved | | | |
| 0 1 1 1 1 | Reserved | | | |
| 1 0 0 0 0 | Reserved | | | |
| 1 0 0 0 1 | Reserved | | | |
| 1 0 0 1 0 | Reserved | | | |
| 1 0 0 1 1 | Reserved | | | |
| 1 0 1 0 0 | Reserved | | | |
| 1 0 1 0 1 | Reserved | | | |
| 1 0 1 1 0 | Reserved | | | |
| 1 0 1 1 1 | Reserved | | | |
| 1 1 0 0 0 | UART7 transmit interrupt request | | | |
| 1 1 0 0 1 | UART7 receive interrupt request | | | |
| 1 1 0 1 0 | UART8 transmit interrupt request | | | |
| 1 1 0 1 1 | UART8 receive interrupt request | | | |
| 1 1 1 0 0 | Reserved | | | |
| 1 1 1 0 1 | Reserved | | | |
| 1 1 1 1 0 | Reserved | | | |
| 1 1 1 1 1 | Reserved | | | |

Note:

1. The falling edge and both edges of signals applied to the $\overline{INTi}$ pin (i = 6 to 8) cause a DMA request generation. The external interrupts (bits POL and LVS in the INTiIC register and the IFSR1 register) are not affected by these DMA request sources, and vice versa.

## DMAi Mode Register (i = 0 to 3) (1)




Symbol: DMD0 to DMD3

Address: (CPU internal register)

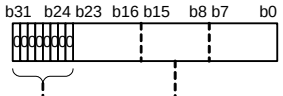
Reset Value: XXXX XXXX XXXX XXXX XXXX XXXX XX00 0000b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| MDi0 | Transfer Mode Select Bit (2) | b1 b0<br>0 0 : DMA transfer disabled<br>0 1 : Single transfer<br>1 0 : Do not use this combination<br>1 1 : Repeat transfer | RW |
| MDi1 | | | RW |
| BWi0 | Transfer Size Select Bit (3) | b3 b2<br>0 0 : 8 bits<br>0 1 : 16 bits<br>1 0 : 32 bits<br>1 1 : Do not use this combination | RW |
| BWi1 | | | RW |
| USAi | Source Addressing Mode Select Bit (3) | 0: Non-incrementing addressing<br>1: Incrementing addressing | RW |
| UDAi | Destination Addressing Mode Select Bit (3) | 0: Non-incrementing addressing<br>1: Incrementing addressing | RW |
| —<br>(b7-b6) | No register bits; should be written with 0 and read as undefined value | | — |
| —<br>(b31-b8) | No register bits; should be written with 0 and read as undefined value | | — |

Notes:
1. The LDC instruction should be used to write to this register.
2. This bit should be set after all other DMAC-associated registers are set.
3. Set bits MDi1 and MDi0 to 00b before rewriting these bits.

**Figure 13.4 Registers DMD0 to DMD3**

## DMAi Terminal Count Register (i = 0 to 3) (1)




Symbol: DCT0 to DCT3

Address: (CPU internal register)

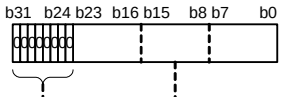
Reset Value: XXXX XXXXh

| Function | Setting Range | RW |
|---|---|---|
| Set the transfers to be performed | 000000h to FFFFFFh (2) | RW |
| Reserved | Should be set to 00h | RW |

Notes:
1. The LDC instruction should be used to write to this register. The register should be set while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).
2. When this register is set to 000000h, any new DMA transfer request is unacceptable.

**Figure 13.5 Registers DCT0 to DCT3**

## DMAi Terminal Count Reload Register (i = 0 to 3) (1)



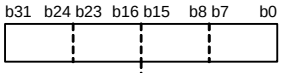

| Symbol | Address | Reset Value |
|---|---|---|
| DCR0 to DCR3 | (CPU internal register) | XXXX XXXXh |

| Function | Setting Range | RW |
|---|---|---|
| Set the transfers to be performed | 000000h to FFFFFFh | RW |
| Reserved | Should be set to 00h | RW |

Note:
1. The LDC instruction should be used to write to this register. The register should be set while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

**Figure 13.6 Registers DCR0 to DCR3**

## DMAi Source Address Register (i = 0 to 3) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| DSA0 to DSA3 | (CPU internal register) | XXXX XXXXh |

| Function | Setting Range | RW |
|---|---|---|
| Set a source address | 00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh (64-Mbyte space) | RW |

Note:
1. The LDC instruction should be used to write to this register. The register should be set while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

**Figure 13.7 Registers DSA0 to DSA3**

## DMAi Source Address Reload Register (i = 0 to 3) (1)

| b31 b24 b23 b16 b15 b8 b7 b0 | Symbol | Address | Reset Value |
|---|---|---|---|
| | DSR0 to DSR3 | (CPU internal register) | XXXX XXXXh |

| Function | Setting Range | RW |
|---|---|---|
| Set a source address | 00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh (64-Mbyte space) | RW |

Note:
1. The LDC instruction should be used to write to this register. The register should be set while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

**Figure 13.8 Registers DSR0 to DSR3**

## DMAi Destination Address Register (i = 0 to 3) (1)

| b31 b24 b23 b16 b15 b8 b7 b0 | Symbol | Address | Reset Value |
|---|---|---|---|
| | DDA0 to DDA3 | (CPU internal register) | XXXX XXXXh |

| Function | Setting Range | RW |
|---|---|---|
| Set a destination address | 00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh (64-Mbyte space) | RW |

Note:
1. The LDC instruction should be used to write to this register. The register should be set while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

**Figure 13.9 Registers DDA0 to DDA3**

## DMAi Destination Address Reload Register (i = 0 to 3) (1)

| b31 b24 b23 b16 b15 b8 b7 b0 | Symbol | Address | Reset Value |
|---|---|---|---|
| | DDR0 to DDR3 | (CPU internal register) | XXXX XXXXh |

| Function | Setting Range | RW |
|---|---|---|
| Set a destination address | 00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh (64-Mbyte space) | RW |

Note:
1. The LDC instruction should be used to write to this register. The register should be set while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

**Figure 13.10 Registers DDR0 to DDR3**

## 13.1 Transfer Cycle

The transfer cycle is composed of bus cycles to read data from memory or SFR (source read) and to write data to destination address (destination write).
The read and write bus cycles vary with the setting of registers DSAi (i = 0 to 3) and DDAi, the width of data bus connected to the relevant device and bus timing.

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

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

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

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

### 13.1.2 Effect of Bus Timing

In the R32C/100 Series, each device has its own bus addresses assigned. The bus width and bus timing vary with each device. Table 13.5 lists the bus width and access cycles for each device.

**Table 13.5 Bus Width and Bus Cycles**

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

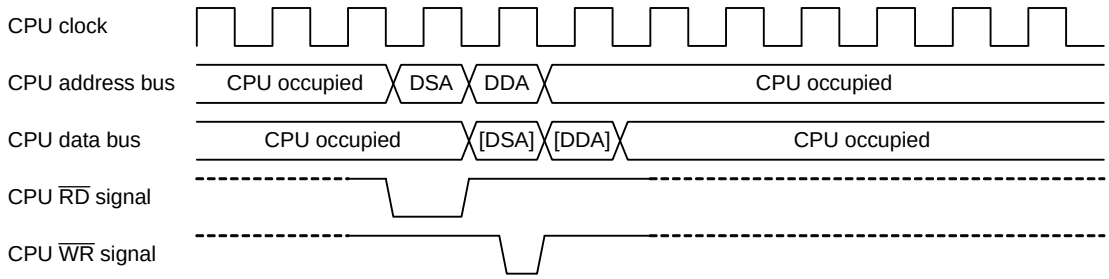
Notes:
1. Reserved spaces are included.
2. Access cycles are based on each bus clock.
3. An access to the same page as the previous time requires two cycles. Otherwise, three cycles are required.
4. If write cycles are generated sequentially, each write cycle except the initial one has two access cycles.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

Figure 13.11 shows an example of source-read bus cycles in a transfer cycle. In this figure, the number of source-read bus cycle is shown under different conditions, provided that the destination address is in an internal RAM with one bus cycle of destination-write. In real operation, the transfer cycles change according to conditions for destination-write bus cycles as well as for source-read bus cycles. To calculate a transfer cycle, therefore, respective conditions should be applied to both destination-write bus cycle and source-read bus cycle. In (2) of Figure 13.11, for example, if the destination-write bus cycle is generated twice, both bus cycles are two, respectively.

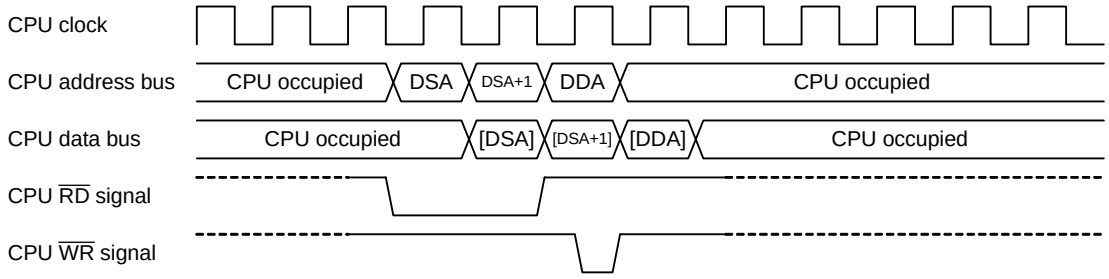
### 13.1.3 Effect of $\overline{\text{RDY}}$ Signal

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

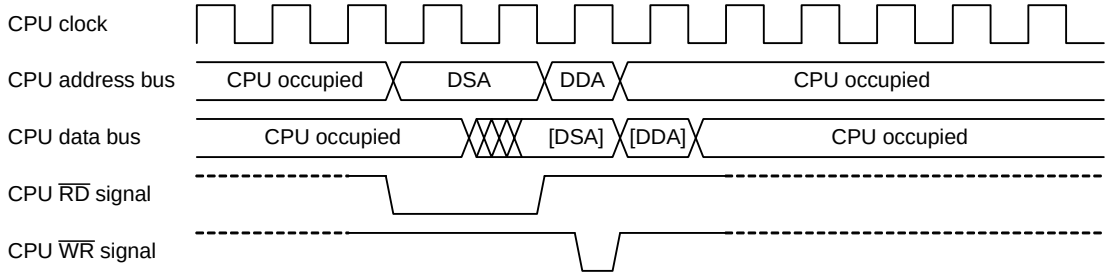
(1) One bus cycle of source-read is generated
Example: 16-bit data transfer from the address 8n of the RAM



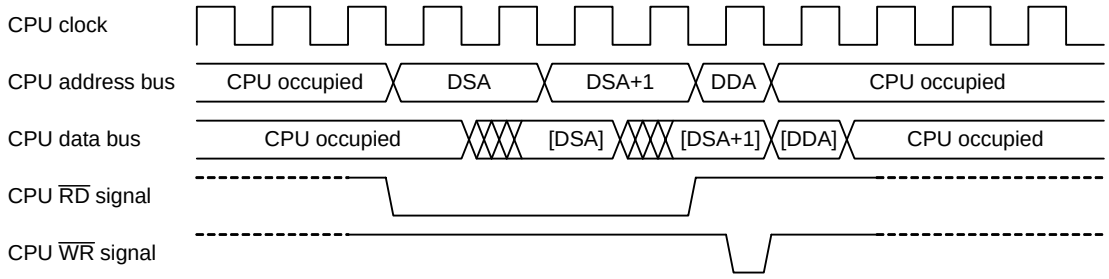

(2) Two bus cycles of source-read are generated
Example: 16-bit data transfer from the address 8n+7 of the RAM




(3) One bus cycle of source-read is generated with one wait cycle
Example: 16-bit data transfer from the address 16n of the ROM




(4) Two bus cycles of source-read is generated with one wait cycle
Example: 16-bit data transfer from the address 16n+7 of the ROM




Note:
1. The above applies under the condition of one bus cycle of destination-write. In real cases, the number of destination-write bus cycles should be considered according to the conditions such as above.

**Figure 13.11 Source-read Bus Cycles in a Transfer Cycle**

## 13.2 DMA Transfer Cycle

The DMA transfer cycles are calculated as follows:

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

where:

$j$ = access cycles for read,

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

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

"+1" in the formula above means a cycle required to decrement the value of DCTi register (i = 0 to 3).

The following are calculation examples:

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

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

Thus, there are three cycles.

To transfer 16-bit data from the AD00 register at address 380h to registers P1 and P0 at addresses 3C1h and 3C0h, respectively, with the peripheral bus clock (= 1/2 CPU clock),

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

Thus, there are nine cycles.

## 13.3 Channel Priority and DMA Transfer Timing

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

Figure 13.12 shows an example of the DMA transfer by external source, specifically when a DMA0 request and a DMA1 request are simultaneously generated. The DMA0 request having higher priority is received first to start a transfer. 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.

DMA transfer requests cannot be counted up. The transfer occurs only once even when an $\overline{\text{INTi}}$ interrupt is generated more than once before receiving the bus privilege, as the DMA1 shown in Figure 13.12.

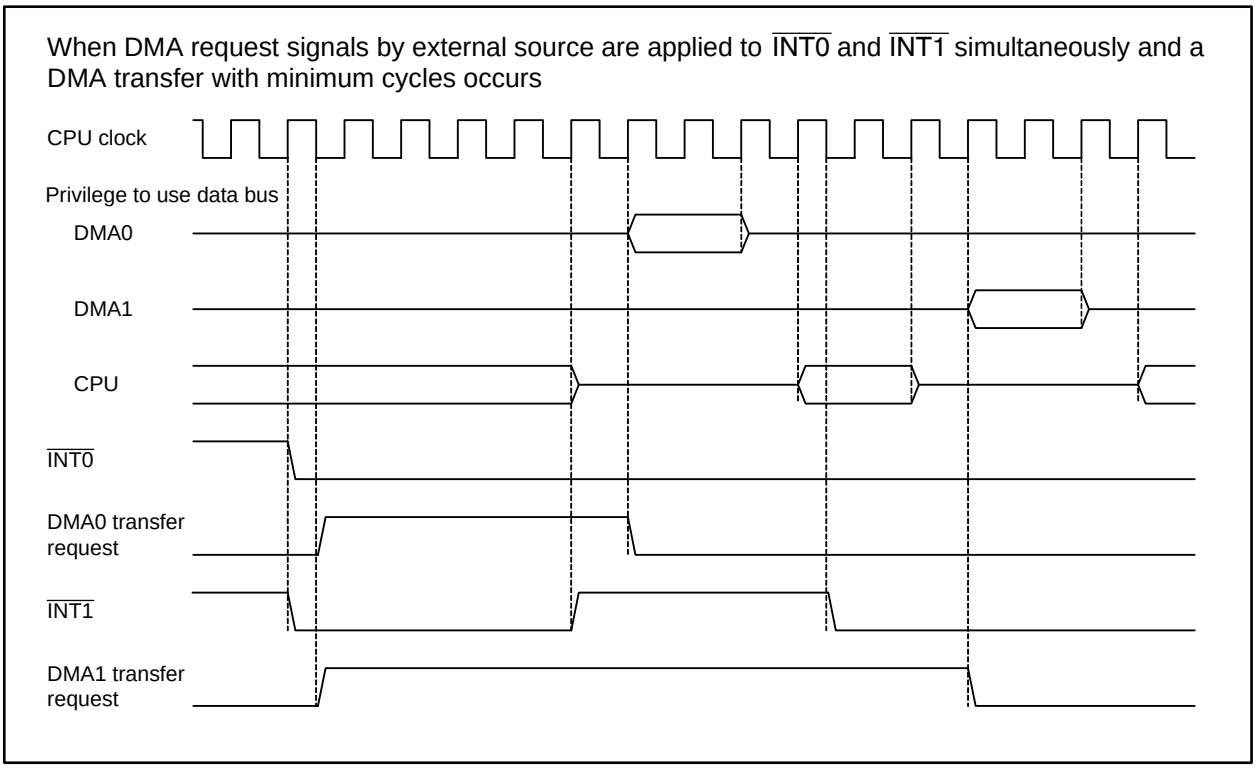



**Figure 13.12 DMA Transfer by External Source**

## 13.4 Notes on DMAC

### 13.4.1 DMAC-associated Register Settings

- Set the DMAC-associated registers while bits MDi1 and MDi0 (i = 0 to 3) in the DMDi register are 00b (DMA transfer disabled). Then, set bits MDi1 and MDi0 to 01b (single transfer) or 11b (repeat transfer) at the end of the setup procedure. This procedure is also applied to rewriting bits UDAi, USAi, and BWi1 and BWi0 in the DMDi register.

- In case the DMAC-associated registers are to be rewritten while DMA transfer is enabled, disable the peripheral function as DMA request source so that no DMA transfer request is generated, then set bits MDi1 and MDi0 in the DMDi register of the corresponding channel to 00b (DMA transfer disabled).

- Once a DMA transfer request is accepted, DMA transfer cannot be disabled even if setting bits MDi1 and MDi0 in the DMDi register to 00b (DMA transfer disabled). Do not change the settings of any DMAC-associated registers other than bits MDi1 and MDi0 until the DMA transfer is completed.

- Wait six or more peripheral clocks to set bits MDi1 and MDi0 in the DMDi register to 01b (single transfer) or 11b (repeat transfer) after setting registers DMiSL and DMiSL2.

### 13.4.2 Read from DMAC-associated Registers

- To sequentially read respective registers DMiSL and DMiSL2, follow the reading order as below:
  DM0SL, DM1SL, DM2SL, and DM3SL
  DM0SL2, DM1SL2, DM2SL2, and DM3SL2

# 14. DMAC II

DMAC II is activated by an interrupt request from any peripheral function, and performs data transfer without a CPU instruction. Transfer sources can be selected from memory, immediate data, memory + memory, and immediate data + memory.
Table 14.1 lists specifications of DMAC II.

**Table 14.1 DMAC II Specifications**

| Item | Specification |
|---|---|
| Triggers for DMAC II | Interrupt requests generated by any of peripheral functions when bits ILVL2 to ILVL0 in the corresponding interrupt control register are set to 111b (level 7) |
| Transfer types | • Data in memory is transferred to memory (memory-to-memory transfer)<br>• Immediate data is transferred to memory (immediate data transfer)<br>• Data in memory + data in memory are transferred to memory (calculation transfer)<br>• Immediate data + data in memory are transferred to memory (calculation transfer) |
| Transfer sizes | 8 bits or 16 bits |
| Transfer memory spaces | From a given address in a 64-Mbyte space (00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh) to another given address in the same space (1) |
| Addressing modes | Individually selectable for each source address and destination address from the following two modes:<br>• Non-incrementing addressing: Address is held constant throughout a data transfer/a DMA II transaction<br>• Incrementing addressing: Address increments by 1 (when an 8 bit-data is transferred) or 2 (when a 16 bit-data is transferred) after each data transfer |
| Transfer modes | • Single transfer: Only one data transfer is performed by one transfer request<br>• Burst transfer: Data transfers are continuously performed for the number of times set in the transfer counter<br>• Multiple transfer: Multiple memory-to-memory transfers are performed from different source addresses to different destination addresses by one transfer request |
| Chained transfer | Data transfer is sequentially performed according to a DMAC II Index (transfer information) linked with the previous transfer |
| DMA II transfer complete interrupt request | An interrupt request is generated when the transfer counter reaches 0000h |

Note:
1. When a 16-bit data is transferred to destination address at FFFFFFFFh, it is transferred to 00000000h as well as FFFFFFFFh. The same transfer is performed when the source address is FFFFFFFh.

## 14.1 DMAC II Settings

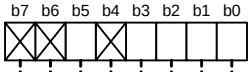
To activate DMAC II, set up the following items:
- Registers RIPL1 and RIPL2
- DMAC II index
- The interrupt control register of the peripheral function triggering DMAC II
- The relocatable vector of the peripheral function triggering DMAC II
- IIRLT bit in the IIOiIE register (i = 0 to 11) if the intelligent I/O interrupt is used. Refer to 11. “Interrupts” for details on the IIOiIE register.

### 14.1.1 Registers RIPL1 and RIPL2

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

Figure 14.1 shows registers RIPL1 and RIPL2.

Wake-up IPL Setting Register i (i = 1, 2) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| RIPL1, RIPL2 | 4407Fh, 4407Dh | XX0X 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| RLVL0 | Interrupt Priority Level for Wake-up Select Bit (2) | b2 b1 b0<br>0 0 0 : Level 0<br>0 0 1 : Level 1<br>0 1 0 : Level 2<br>0 1 1 : Level 3<br>1 0 0 : Level 4<br>1 0 1 : Level 5<br>1 1 0 : Level 6<br>1 1 1 : Level 7 | RW |
| RLVL1 | | | RW |
| RLVL2 | | | RW |
| FSIT | Fast Interrupt Select Bit (3) | 0: Use interrupt request level 7 for normal interrupt<br>1: Use interrupt request level 7 for fast interrupt (4) | RW |
| — (b4) | No register bit; should be written with 0 and read as undefined value | | — |
| DMAII | DMA II Select Bit (5) | 0: Use interrupt request level 7 for interrupt<br>1: Use interrupt request level 7 for DMA II transfer (4) | RW |
| — (b7-b6) | No register bits; should be written with 0 and read as undefined value | | — |

Notes:
1. Registers RIPL1 and RIPL2 should be identically set.
2. The MCU exits wait mode or stop mode if the request level of requested interrupt is higher than the level selected using bits RLVL2 to RLVL0. These bits should be set to the same value as the IPL in the flag register (FLG).
3. When the FSIT bit is set to 1, an interrupt with interrupt request level 7 becomes the fast interrupt. In this case only one interrupt should be set to the interrupt request level 7.
4. Either the FSIT or DMAII bit should be set to 1. Simultaneous use of the fast interrupt and the DMAC II is not available.
5. Bits ILVL2 to ILVL0 in the interrupt control register should be set after the DMAII bit is set. The DMA II transfer is not affected by the I flag or the IPL.

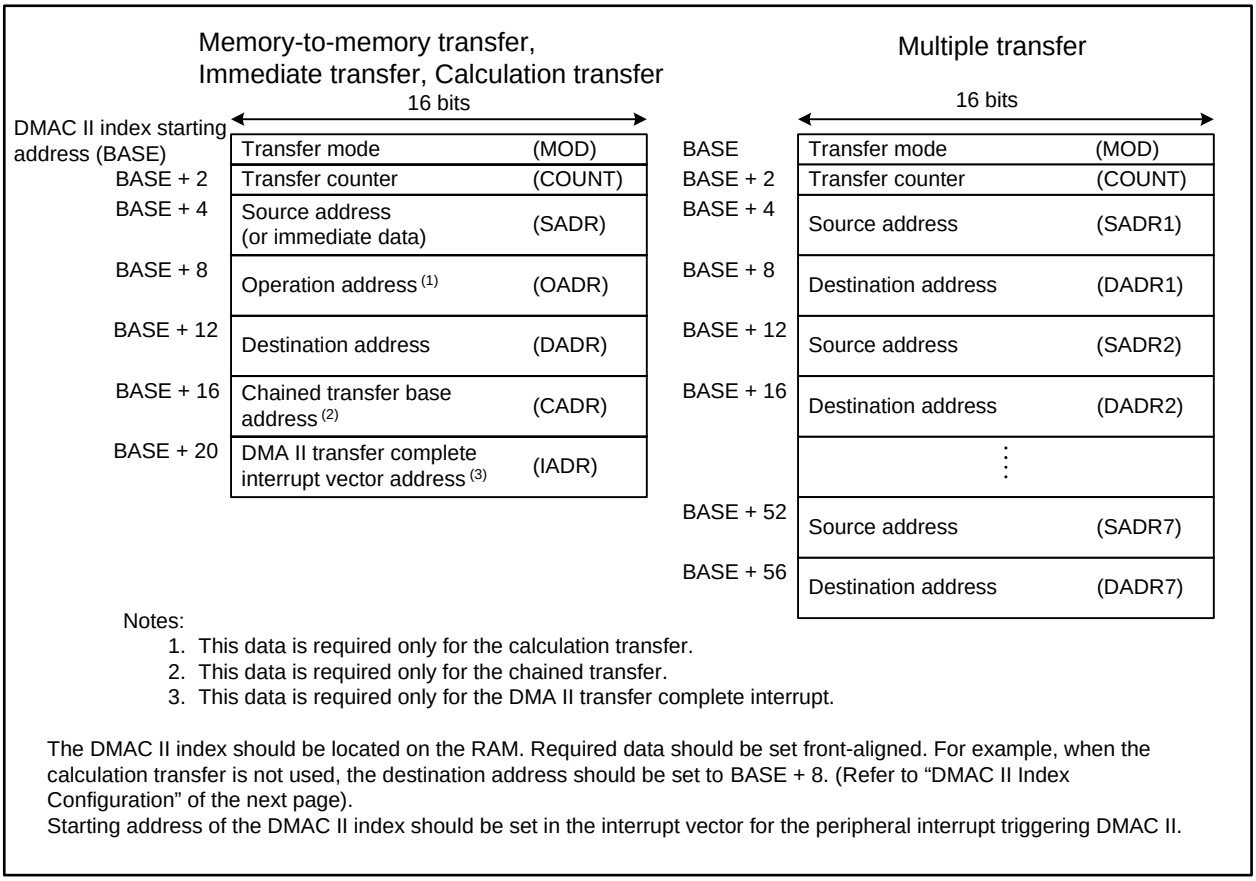
**Figure 14.1 Registers RIPL1 and RIPL2**

### 14.1.2 DMAC II Index

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

This DMAC II index should be located on the RAM.

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




Notes:
1. This data is required only for the calculation transfer.
2. This data is required only for the chained transfer.
3. This data is required only for the DMA II transfer complete interrupt.

The DMAC II index should be located on the RAM. Required data should be set front-aligned. For example, when the calculation transfer is not used, the destination address should be set to BASE + 8. (Refer to “DMAC II Index Configuration” of the next page).

Starting address of the DMAC II index should be set in the interrupt vector for the peripheral interrupt triggering DMAC II.

**Figure 14.2 DMAC II Index**

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

- Transfer mode (MOD)
  2-byte data is required to set transfer mode. Figure 14.3 shows a configuration for transfer mode.
- Transfer counter (COUNT)
  2-byte data is required to set the transfers to be performed.
- Source address (SADR)
  4-byte data is required to set a source address in a memory or an immediate data. However, the two upper bytes of immediate data are ignored.
- Operation address (OADR)
  4-byte data is required to set an address in a memory to be calculated. This data setting is required only for the calculation transfer.
- Destination address (DADR)
  4-byte data is required to set a destination address in a memory.
- Chained transfer base address (CADR)
  4-byte data is required to set BASE, the starting address of the DMAC II index for the next transfer. This data setting is required only for the chained transfer.
- DMA II transfer complete interrupt vector address (IADR)
  4-byte data is required to set a jump address for the DMA II transfer complete interrupt handler. This data setting is required only for the DMA II transfer complete interrupt.

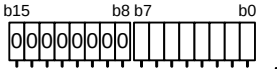
The symbols above are hereinafter used in place of their respective parameters.

**Table 14.2 DMAC II Index Configuration**

| Transfer Data | Memory-to-memory Transfer/ Immediate Data Transfer | | | | Calculation Transfer | | | | Multiple Transfer |
|---|---|---|---|---|---|---|---|---|---|
| Chained transfer | Not used | Used | Not used | Used | Not used | Used | Not used | Used | Not available |
| DMA II transfer complete interrupt | Not used | Not used | Used | Used | Not used | Not used | Used | Used | Not available |
| DMAC II index | MOD<br>COUNT<br>SADR<br>DADR<br>12 bytes | MOD<br>COUNT<br>SADR<br>DADR<br>CADR<br>16 bytes | MOD<br>COUNT<br>SADR<br>DADR<br>IADR<br>16 bytes | MOD<br>COUNT<br>SADR<br>DADR<br>CADR<br>IADR<br>20 bytes | MOD<br>COUNT<br>SADR<br>OADR<br>DADR<br>16 bytes | MOD<br>COUNT<br>SADR<br>OADR<br>DADR<br>CADR<br>20 bytes | MOD<br>COUNT<br>SADR<br>OADR<br>DADR<br>IADR<br>20 bytes | MOD<br>COUNT<br>SADR<br>OADR<br>DADR<br>CADR<br>IADR<br>24 bytes | MOD<br>COUNT<br>SADR1<br>DADR1<br><br>SADRi<br>DADRi<br>i = 1 to 7<br>max. 60 bytes<br>(when i = 7) |

## Transfer Mode (MOD) (1)

When multiple transfer is not selected (MULT=0)




| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| SIZE | Transfer Size Select Bit | 0: 8 bits<br>1: 16 bits | RW |
| IMM | Transfer Source Select Bit | 0: Immediate data<br>1: Memory | RW |
| UPDS | Source Addressing Select Bit | 0: Non-incrementing addressing<br>1: Incrementing addressing | RW |
| UPDD | Destination Addressing Select Bit | 0: Non-incrementing addressing<br>1: Incrementing addressing | RW |
| OPER | Calculation Transfer Select Bit | 0: Not used<br>1: Used | RW |
| BRST | Burst Transfer Select Bit | 0: Single transfer<br>1: Burst transfer | RW |
| INTE | DMA II Transfer Complete Interrupt Select Bit | 0: Not used<br>1: Used | RW |
| CHAIN | Chained Transfer Select Bit | 0: Not used<br>1: Used | RW |
| —<br>(b14-b8) | Reserved | Should be written with 0 | RW |
| MULT | Multiple Transfer Select Bit | 0: Not used | RW |

When multiple transfer is selected (MULT=1)


| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| SIZE | Transfer Size Select Bit | 0: 8 bits<br>1: 16 bits | RW |
| IMM | Reserved | Should be written with 1 | RW |
| UPDS | Source Addressing Select Bit | 0: Non-incrementing addressing<br>1: Incrementing addressing | RW |
| UPDD | Destination Addressing Select Bit | 0: Non-incrementing addressing<br>1: Incrementing addressing | RW |
| CNT0 | Transfer Number Set Bit | b6 b5 b4<br>0 0 0 : Do not use this combination<br>0 0 1 : Once<br>0 1 0 : Twice<br>:<br>:<br>1 1 1 : Seven times | RW |
| CNT1 | | | RW |
| CNT2 | | | RW |
| CHAIN | Reserved | Should be written with 0 | RW |
| —<br>(b14-b8) | Reserved | Should be written with 0 | RW |
| MULT | Multiple Transfer Select Bit | 1: Used | RW |

Note:
1. The MOD should be located on the RAM.

**Figure 14.3 MOD**

### 14.1.3 Interrupt Control Register of the Peripheral Function

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

### 14.1.4 Relocatable Vector Table of the Peripheral Function

Set the starting address of the DMAC II index to the interrupt vector for the peripheral interrupt triggering DMAC II.
To use the chained transfer, locate the relocatable vector table on the RAM.

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

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

## 14.2 DMAC II Performance

To perform a DMA II transfer, the DMAII bits in registers RIPL1 and RIPL2 should be set to 1 (interrupt request level 7 used for DMA II transfer). Any peripheral interrupts with bits ILVL2 to ILVL0 set to 111b (level 7) can be a request source to activate DMAC II. These peripheral interrupt requests are available only for DMA II transfer, that is, they cannot be used for CPU.
When an interrupt request is generated with interrupt request level 7, DMAC II is activated irrespective of the state of I flag or IPL.
When a peripheral interrupt request triggering DMAC II and a higher-priority request such as watchdog timer interrupt, low voltage detection interrupt, oscillator stop detection interrupt, and NMI are simultaneously generated, the higher-priority interrupt is accepted prior to the DMA II transfer, and the DMA II transfer starts after the higher-priority interrupt sequence.

## 14.3 Transfer Types

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

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

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

### 14.3.1 Memory-to-memory Transfer

Data transfer between any two memory locations can be:

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

When increment addressing mode is selected, SADR and DADR increment by one in a 8-bit transfer and by two in a 16-bit transfer after a data transfer for the next transfer. When SADR or DADR exceeds FFFFFFFFh as a result of address incrementation, it returns to 00000000h. Likewise, when SADR or DADR exceeds 001FFFFFh, it must become 00200000h, but an actual transfer is performed for 00000000h.

### 14.3.2 Immediate Data Transfer

DMAC II transfers immediate data to any memory location. Both incrementing or non-incrementing addressing modes are available for destination address. Store the immediate data to be transferred into SADR. To transfer an 8-bit immediate data, set the data to the one lower byte of SADR. For an 16-bit immediate data, set the data to the two lower bytes. The three upper bytes or the two upper bytes of respective case are ignored.

### 14.3.3 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. Set one address to be calculated or an immediate data to SADR and set the other address to be calculated to OADR. Both incrementing or non-incrementing addressing modes are available for source and destination addresses in case of a data in memory + a data in memory calculation transfer. If the source addressing is incrementing mode, the operation addressing should be also incrementing. In case of an immediate data + a data in memory calculation transfer, the addressing mode is selectable only for destination address.

## 14.4 Transfer Modes

DMAC II provides three types of basic transfer modes: single transfer, burst transfer, and multiple transfer. COUNT determines the number of transfers to be performed. No transfer is performed when COUNT is set to 0000h.

### 14.4.1 Single Transfer

Set the BRST bit in the MOD to 0.
One data transfer is performed by one transfer request.
When incrementing addressing mode is selected for the source and/or destination address, the address(es) increment(s) after a data transfer for the next transfer.
COUNT is decremented every time a data transfer is performed. When COUNT reaches 0000h, the DMA II transfer complete interrupt request is generated if the INTE bit in the MOD is 1 (the DMA II transfer complete interrupt used).

### 14.4.2 Burst Transfer

Set the BRST bit in the MOD to 1.
DMAC II continuously transfers data for the number of times determined by COUNT by one transfer request. COUNT is decremented every time a data transfer is performed. When COUNT reaches 0000h, the burst transfer is completed. The DMA II transfer complete interrupt request is generated if the INTE bit is 1 (the DMA II transfer complete interrupt used).
No interrupt is accepted during burst transfer being performed.

### 14.4.3 Multiple Transfer

Set the MULT bit in the MOD to 1.
Multiple memory-to-memory transfers are performed from different source addresses to different destination addresses by one transfer request.
Bits CNT2 to CNT0 in the MOD select the number of transfers to be performed from 001b (once) to 111b (seven times). These bits should not be set to 000b.
Allocate required number of SDARs and DADRs alternately following MOD and COUNT.
When the multiple transfer is selected, the following transfer functions are not available: the calculation transfer, burst transfer, chained transfer, and DMA II transfer complete interrupt.

## 14.5 Chained Transfer

The chained transfer is available when the CHAIN bit in the MOD is set to1.
The chained transfer is performed as follows:

(1) When a transfer request is generated, a data transfer is performed according to DMAC II index specified by the corresponding interrupt vector. Either single transfer (the BRST bit in the MOD is 0) or burst transfer (the BRST bit is 1) is performed according to the BRST bit setting.

(2) When COUNT reaches 0000h, the value in the interrupt vector in (1) above is overwritten with the value in CADR. Simultaneously, the DMA II transfer complete interrupt is generated when the INTE bit in the MOD is 1.

(3) When the next DMA II transfer request is generated, the data transfer is performed according to DMAC II index specified by the peripheral interrupt vector in (2) above.

Figure 14.4 shows the relocatable vector and DMAC II index in chained transfer.
To use the chained transfer, the relocatable vector table should be located on the RAM.

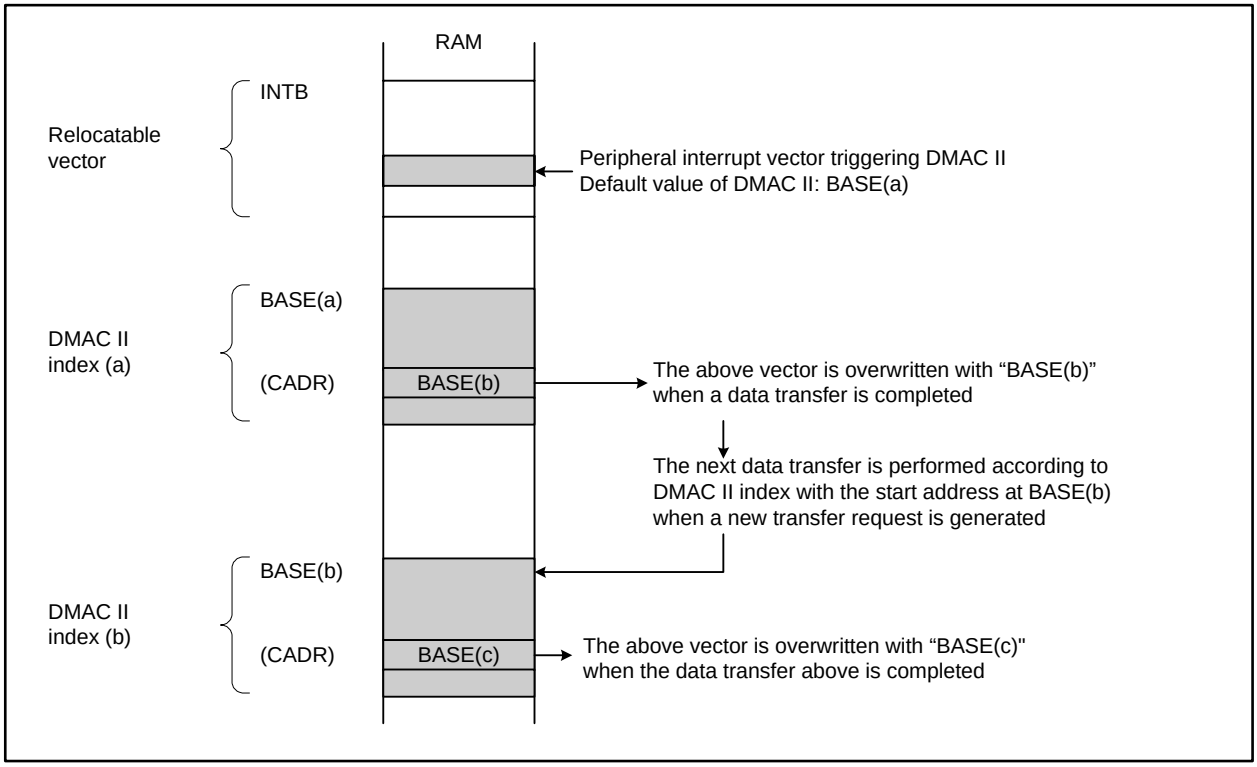



**Figure 14.4 Relocatable Vector and DMAC II Index in Chained Transfer**

## 14.6 DMA II Transfer Complete Interrupt

The DMA II transfer complete interrupt is available when the INTE bit in the MOD is set to1.
The starting address of the DMA II transfer complete interrupt handler should be set to IADR. The interrupt is generated when COUNT reaches 0000h.

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

## 14.7 Execution Time

DMAC II execution cycle is calculated by the following equations:

Other than multiple transfer: t = 6 + (26 + a + b + c + d) × m + (4 + e) × n cycles
Multiple transfer: t = 21 + (11 + b + c) × k cycles

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

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

When a DMA II transfer complete interrupt (transfer counter = 2) is generated with no chained transfer after a memory-to-memory transfer is performed twice with a incremented source address and a non-incremented destination address in single transfer mode

(a = -1, b = 0, c = 1, d = 0, e = 0, m = 1)

First DMA II transfer t = 6 + (26 - 1 + 0 + 1 + 0) × 1 + (4 + 0) × 1 = 36 cycles
Second DMA II transfer t = 6 + (26 - 1 + 0 + 1 + 0) × 1 + (4 + 0) × 0 = 32 cycles

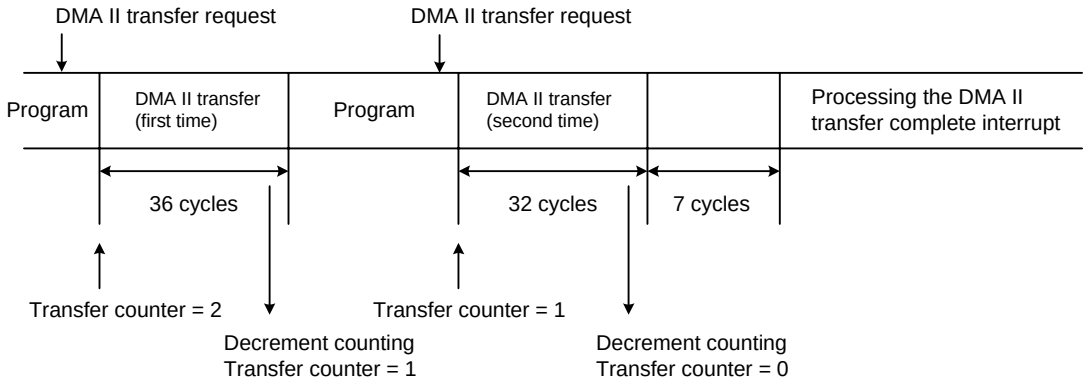
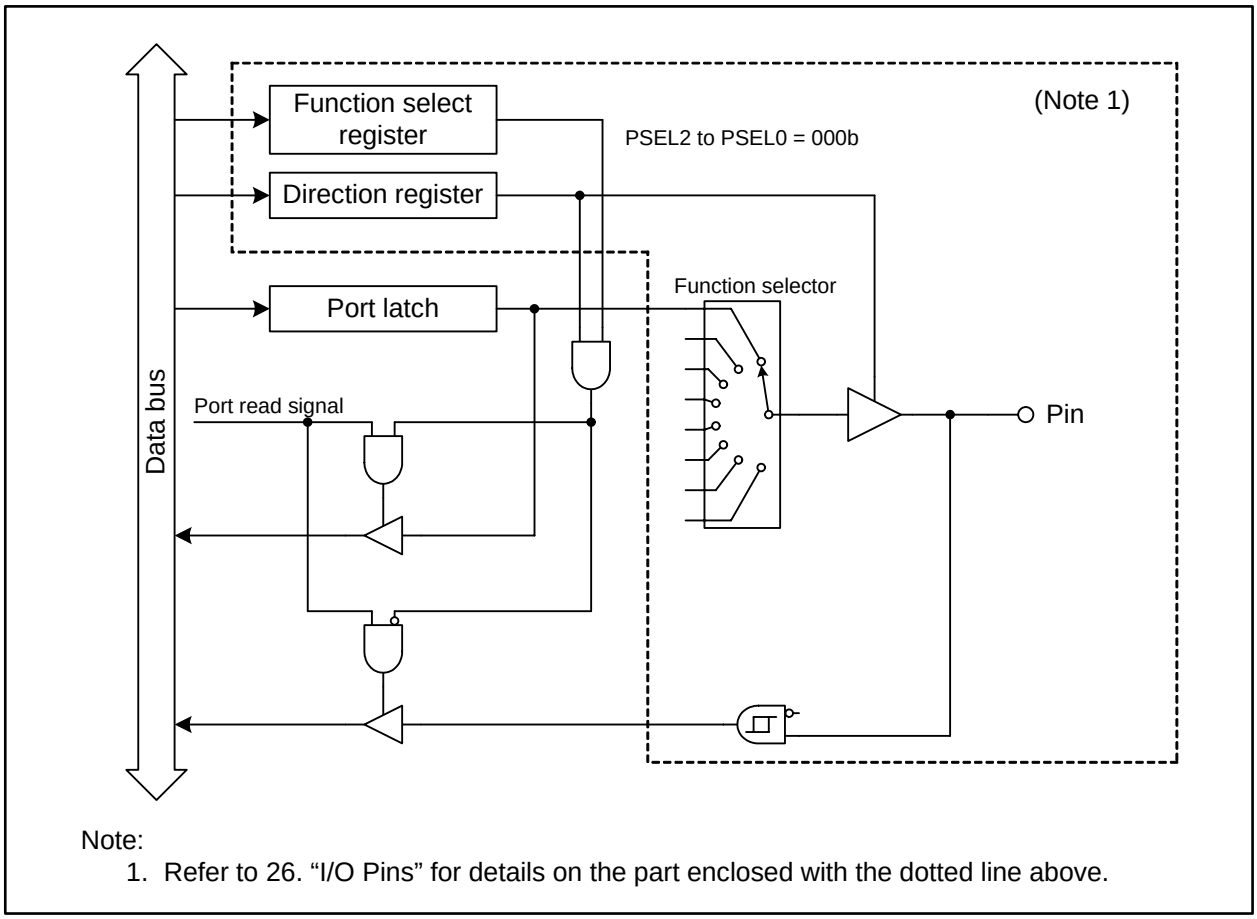



**Figure 14.5 Transfer Cycles**

# 15. Programmable I/O Ports

The programmable I/O ports in each pin package are designated as follows:
100-pin package: 84 ports from P0 to P10 (excluding P8_5 and P9_0 to P9_2), and
144-pin package: 120 ports from P0 to P15 (excluding P8_5 and P14_0 to P14_2).
Each port status, input or output, can be selected using the direction register except P8_5 and P9_1/P14_1 which are input only. The P8_5 bit in the P8 register indicates an $\overline{\text{NMI}}$ input level since the P8_5 shares a pin with the $\overline{\text{NMI}}$.
Figure 15.1 shows a configuration of programmable I/O ports and Figure 15.2 to Figure 15.4 show a configuration of each input-only port.




Note:
1. Refer to 26. “I/O Pins” for details on the part enclosed with the dotted line above.

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

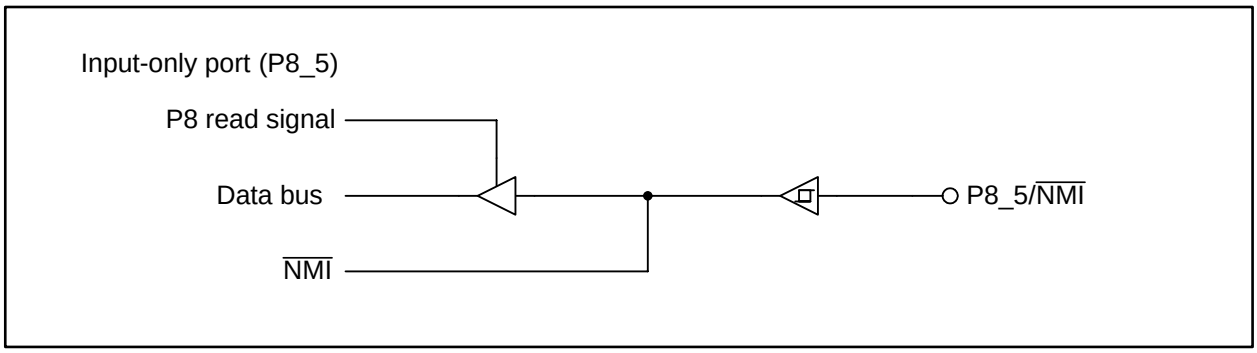



**Figure 15.2 Input-only Port Configuration (1/3)**

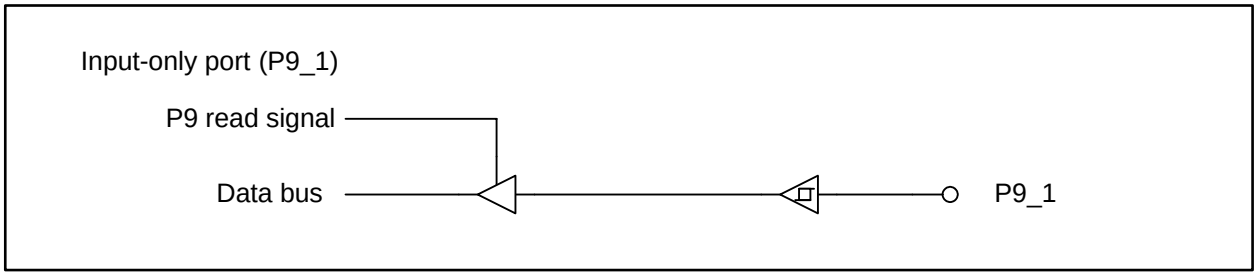



**Figure 15.3 Input-only Port Configuration (2/3) (in the 100-pin package only)**

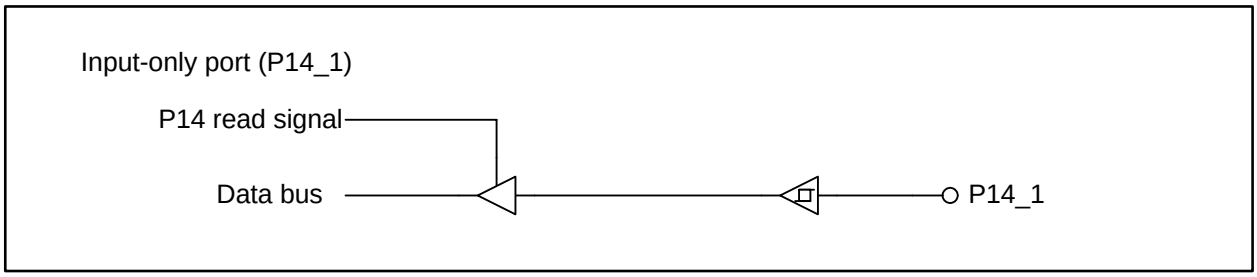



**Figure 15.4 Input-only Port Configuration (3/3) (in the 144-pin package only)**

## 15.1 Port Pi Register (Pi register, i = 0 to 15)

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

When a programmable I/O port is selected in the output function select register, the value in the port latch as output data and pin states as input data are respectively read.

In memory expansion mode or microprocessor mode, this register cannot control pins being used as the bus control pins (A0 to A23, D0 to D31, $\overline{CS0}$ to $\overline{CS3}$, $\overline{WR}$/$\overline{WR0}$, $\overline{BC0}$, $\overline{BC1}$/$\overline{WR1}$, $\overline{BC2}$/$\overline{WR2}$, $\overline{BC3}$/$\overline{WR3}$, $\overline{RD}$, CLKOUT/BCLK, $\overline{HLDA}$, $\overline{HOLD}$, ALE, and $\overline{RDY}$).

Figure 15.5 shows the Pi register.

Port Pi Register (i = 0 to 15) (1, 2)

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| P0 to P3 | 03C0h, 03C1h, 03C4h, 03C5h | Undefined |
| P4 to P7 | 03C8h, 03C9h, 03CCh, 03CDh | Undefined |
| P8 (3), P9 (3, 4) | 03D0h, 03D1h | Undefined |
| P10, P11 (2, 5) | 03D4h, 03D5h | Undefined |
| P12, P13 (2) | 03D8h, 03D9h | Undefined |
| P14 (2, 3, 4, 5), P15 (2) | 03DCh, 03DDh | Undefined |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| Pi_0 | Port Pi_0 bit (4) | When the direction bit = 0 (input)<br>A value is written to the corresponding bit. It is not output due to input mode selected. The read value is the corresponding pin state as follows:<br>0: Low<br>1: High<br><br>When the direction bit = 1 (output)<br>A value is written to the corresponding bit as follows:<br>0: Output low<br>1: Output high<br>The read value has the same output level as that written to the corresponding bit | RW |
| Pi_1 | Port Pi_1 bit (3) | | RW |
| Pi_2 | Port Pi_2 bit (4) | | RW |
| Pi_3 | Port Pi_3 bit | | RW |
| Pi_4 | Port Pi_4 bit | | RW |
| Pi_5 | Port Pi_5 bit (3, 5) | | RW |
| Pi_6 | Port Pi_6 bit (5) | | RW |
| Pi_7 | Port Pi_7 bit (5) | | RW |

Notes:
1. In memory expansion mode or microprocessor mode, the PDi register cannot control pins being used as bus control pins (A0 to A23, D0 to D31, $\overline{CS0}$ to $\overline{CS3}$, $\overline{WR}$/$\overline{WR0}$, $\overline{BC0}$, $\overline{BC1}$/$\overline{WR1}$, $\overline{BC2}$/$\overline{WR2}$, $\overline{BC3}$/$\overline{WR3}$, $\overline{RD}$, CLKOUT/BCLK, $\overline{HLDA}$, $\overline{HOLD}$, ALE, and $\overline{RDY}$).
2. Registers P11 to P15 are available in the 144-pin package only.
3. The P8_5 bit in the P8 register, the P9_1 bit in the P9 register (in the 100-pin package), and the P14_1 bit in the P14 register (in the 144-pin package) are read only.
4. Bits P9_0 and P9_2 in the P9 register (in the 100-pin package) and bits P14_0 and P14_2 in the P14 register (in the 144-pin package) are reserved. These bits should be written with 0 and read as undefined values.
5. Bits P11_5 to P11_7 in the P11 register and the P14_7 bit in the P14 register are unavailable on this MCU. These bits should be written with 0 and read as undefined values.

**Figure 15.5 Registers P0 to P15**

# 16. Timers

This MCU has eleven 16-bit timers which are divided into two groups according to functions: five timer As and six timer Bs. Each timer functions individually. The count source of each timer provides the clock for timer operations including counting, reloading and so on.
Figure 16.1 and Figure 16.2 show the configuration of the timers A and B, respectively.

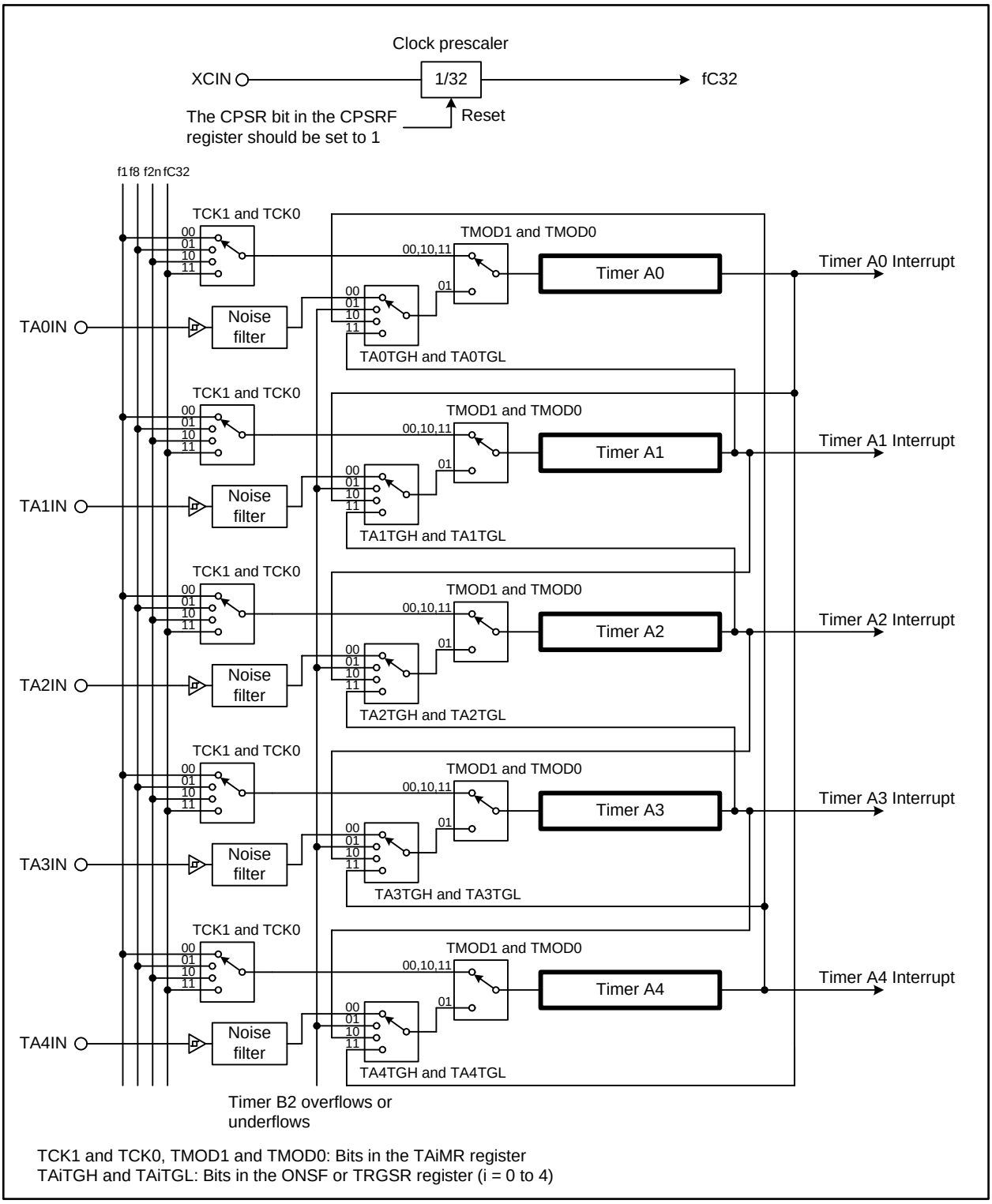



**Figure 16.1 Timer A Configuration**

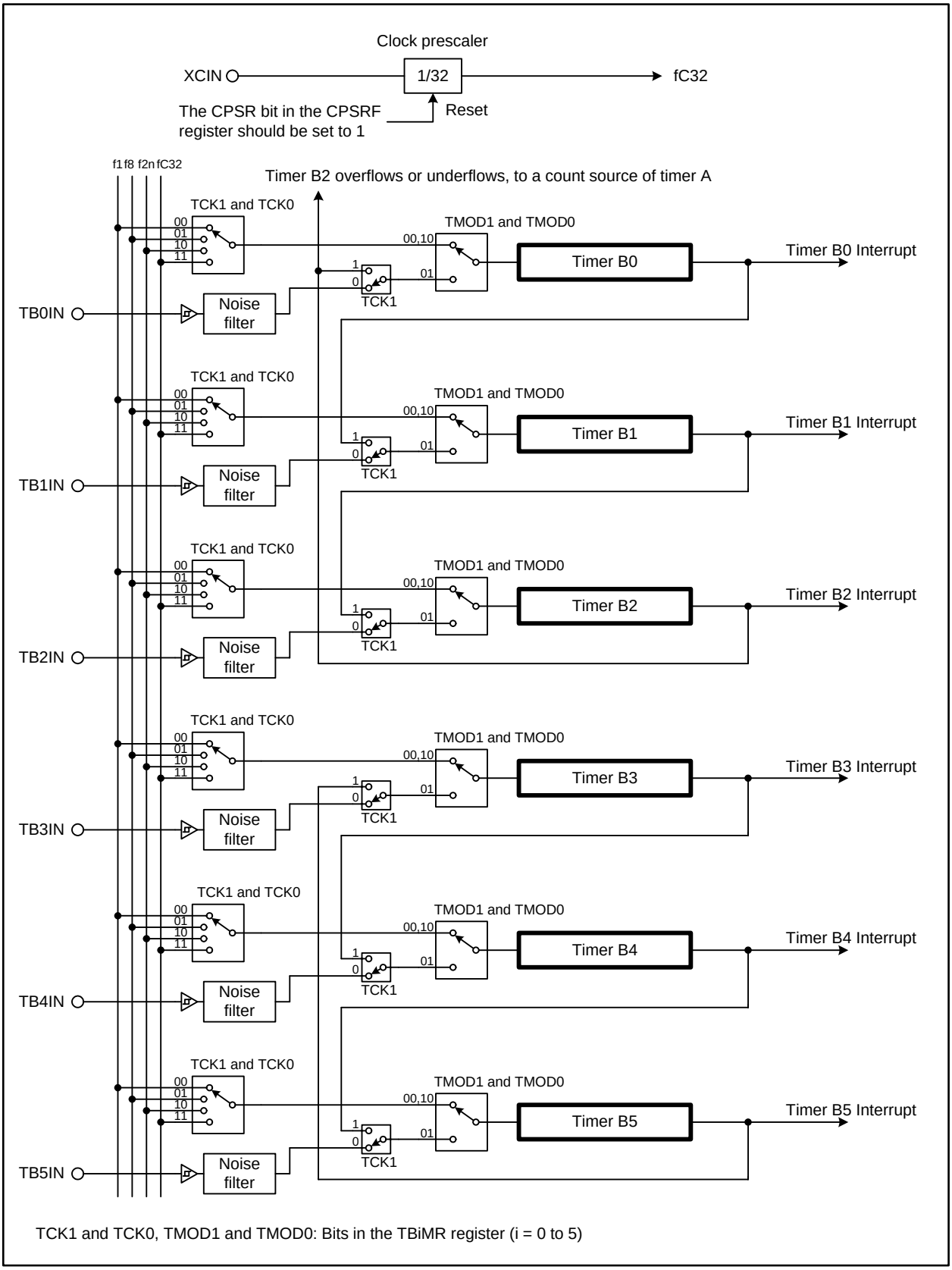


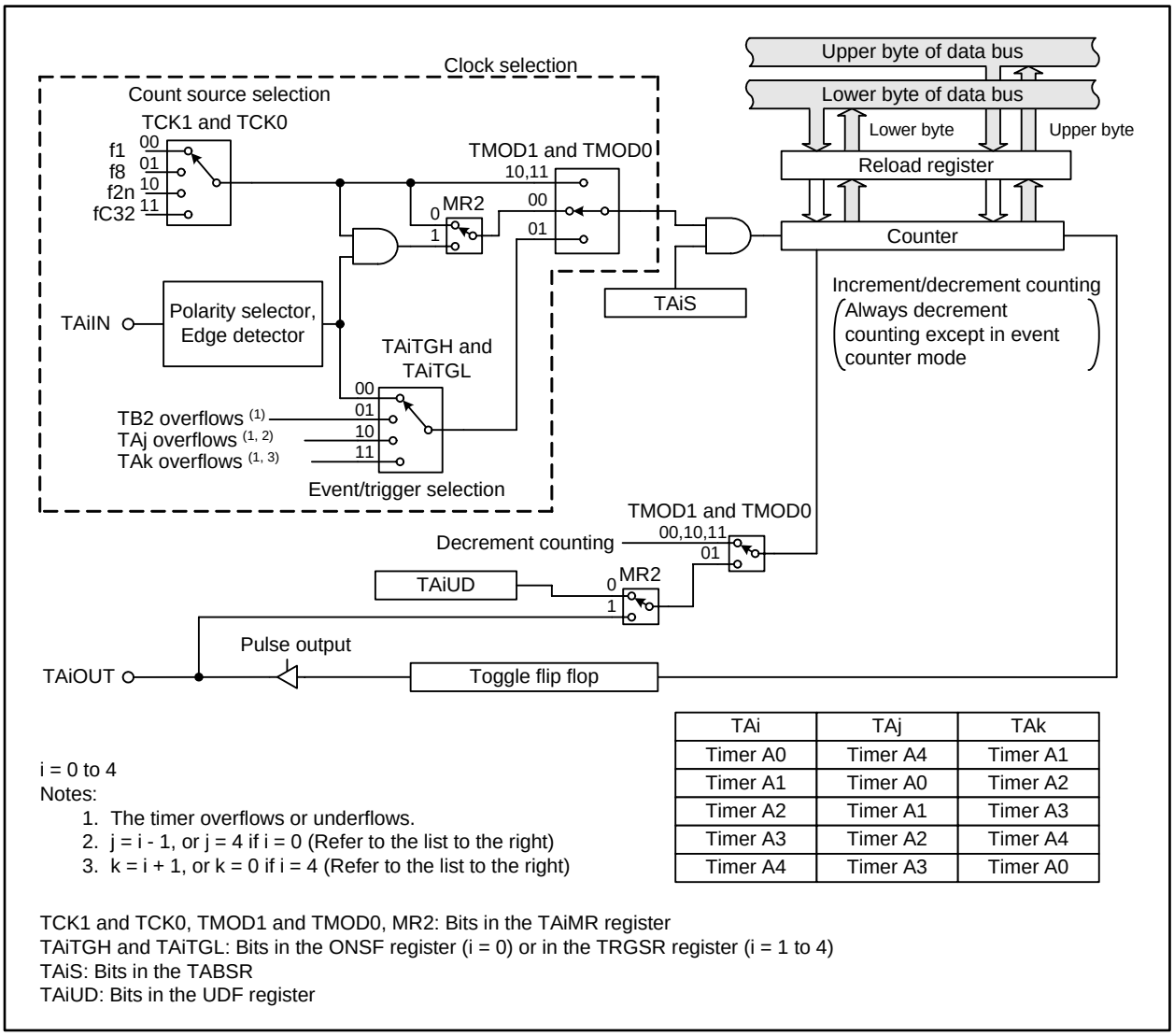

**Figure 16.2 Timer B Configuration**

## 16.1 Timer A

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

The timer A supports four modes shown as below. Timers A0 to A4 in any mode other than the event counter mode have the same function. A mode is selected using bits TMOD1 and TMOD0 in the TAiMR register (i = 0 to 4).

- 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 before the counter reaches 0000h
- Pulse-width modulation mode: The timer sequentially outputs pulses of given width




| TAi | TAj | TAk |
|---|---|---|
| Timer A0 | Timer A4 | Timer A1 |
| Timer A1 | Timer A0 | Timer A2 |
| Timer A2 | Timer A1 | Timer A3 |
| Timer A3 | Timer A2 | Timer A4 |
| Timer A4 | Timer A3 | Timer A0 |

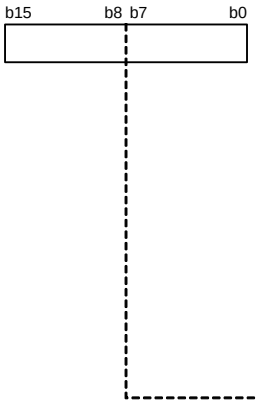
i = 0 to 4

Notes:
1. The timer overflows or underflows.
2. j = i - 1, or j = 4 if i = 0 (Refer to the list to the right)
3. k = i + 1, or k = 0 if i = 4 (Refer to the list to the right)

TCK1 and TCK0, TMOD1 and TMOD0, MR2: Bits in the TAiMR register
TAiTGH and TAiTGL: Bits in the ONSF register (i = 0) or in the TRGSR register (i = 1 to 4)
TAiS: Bits in the TABSR
TAiUD: Bits in the UDF register

**Figure 16.3 Timer A Block Diagram**

## Timer Ai Register (i = 0 to 4) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| TA0 to TA2 | 0347h-0346h, 0349h-0348h, 034Bh-034Ah | Undefined |
| TA3, TA4 | 034Dh-034Ch, 034Fh-034Eh | Undefined |

| Mode | Function | Setting Range | RW |
|---|---|---|---|
| Timer Mode | Divides the count source by n+1 (n = setting value) | 0000h to FFFFh | RW |
| Event Counter Mode | Divides the count source by FFFFh -n+1 (on increment counting) or by n+1 (on decrement counting) (n = setting value) (2) | 0000h to FFFFh | RW |
| One-Shot Timer Mode | Divides the count source by n, then stops (n = setting value) (3) | 0000h to FFFFh (4) | WO |
| Pulse-Width Modulation Mode (16-Bit PWM) | PWM period: ($2^{16}$ -1) / fj<br>High level width of PWM pulse: n / fj<br>(fj = count source frequency, n = setting value of the TAi register) (5) | 0000h to FFFEh (4) | WO |
| Pulse-Width Modulation Mode (8-Bit PWM) | PWM period: ($2^{8}$ -1) × (m+1) / fj<br>High level width of PWM pulse: (m+1)n / fj (fj = count source frequency, n = setting value of the upper byte in the TAi register, m = setting value of the lower byte in the TAi register ) (5) | 00h to FEh (upper byte)<br>00h to FFh (lower byte) (4) | WO |

fj: f1, f8, f2n, fC32

Notes:
1. A 16-bit read/write access to this register should be performed.
2. The timer counts an external input pulse or an overflow and underflow of other timers.
3. When the TAi register is set to 0000h, the timer counter does not start, nor the TAi interrupt request is generated.
4. The MOV instruction should be used to set the TAi register.
5. When the TAi register is set to 0000h, the pulse-width modulator does not operate, the TAiOUT pin is held low, and the TAi interrupt request is not generated. The same situation occurs in 8-bit pulse-width modulator mode if the upper byte in the TAi register are set to 00h.

**Figure 16.4 Registers TA0 to TA4**

## Timer Ai Mode Register (i = 0 to 4)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | 0 | | |

| Symbol | Address | Reset Value |
|---|---|---|
| TA0MR to TA4MR | 0356h, 0357h, 0358h, 0359h, 035Ah | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TMOD0 | Operating Mode Select Bit | b1 b0<br>0 0 : Timer mode<br>0 1 : Event counter mode<br>1 0 : One-shot timer mode<br>1 1 : Pulse-width modulation mode | RW |
| TMOD1 | | | RW |
| — (b2) | Reserved | Should be written with 0 | RW |
| MR1 | — | Function varies according to the operating mode | RW |
| MR2 | | | RW |
| MR3 | | | RW |
| TCK0 | Count Source Select Bit | Function varies according to the operating mode | RW |
| TCK1 | | | RW |

**Figure 16.5 Registers TA0MR to TA4MR**

## Count Start Register

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| TABSR | 0340h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TA0S | Timer A0 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TA1S | Timer A1 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TA2S | Timer A2 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TA3S | Timer A3 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TA4S | Timer A4 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TB0S | Timer B0 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TB1S | Timer B1 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TB2S | Timer B2 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |

**Figure 16.6 TABSR Register**

## Increment/Decrement Counting Select Register (1)

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| UDF | 0344h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TA0UD | Timer A0 Increment/Decrement Counting Select Bit | 0: Count decremented<br>1: Count incremented (2) | RW |
| TA1UD | Timer A1 Increment/Decrement Counting Select Bit | 0: Count decremented<br>1: Count incremented (2) | RW |
| TA2UD | Timer A2 Increment/Decrement Counting Select Bit | 0: Count decremented<br>1: Count incremented (2) | RW |
| TA3UD | Timer A3 Increment/Decrement Counting Select Bit | 0: Count decremented<br>1: Count incremented (2) | RW |
| TA4UD | Timer A4 Increment/Decrement Counting Select Bit | 0: Count decremented<br>1: Count incremented (2) | RW |
| TA2P | Timer A2 Two-Phase Pulse Signal Processing Select Bit | 0: Two-phase pulse signal processing disabled<br>1: Two-phase pulse signal processing enabled (3) | WO |
| TA3P | Timer A3 Two-Phase Pulse Signal Processing Select Bit | 0: Two-phase pulse signal processing disabled<br>1: Two-phase pulse signal processing enabled (3) | WO |
| TA4P | Timer A4 Two-Phase Pulse Signal Processing Select Bit | 0: Two-phase pulse signal processing disabled<br>1: Two-phase pulse signal processing enabled (3) | WO |

Notes:
1. The MOV instruction should be used to set this register.
2. This bit is enabled when the MR2 bit in the TAiMR register (i = 0 to 4) is set to 0 (the UDF register setting is the source of increment/decrement count switching) in event counter mode.
3. This bit should be set to 0 when the two-pulse signal processing is not in use.

**Figure 16.7 UDF Register**

## One-Shot Start Register

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| ONSF | 0342h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TA0OS | Timer A0 One-Shot Start Bit | 0: Timer in idle state<br>1: Start the timer (1) | RW |
| TA1OS | Timer A1 One-Shot Start Bit | 0: Timer in idle state<br>1: Start the timer (1) | RW |
| TA2OS | Timer A2 One-Shot Start Bit | 0: Timer in idle state<br>1: Start the timer (1) | RW |
| TA3OS | Timer A3 One-Shot Start Bit | 0: Timer in idle state<br>1: Start the timer (1) | RW |
| TA4OS | Timer A4 One-Shot Start Bit | 0: Timer in idle state<br>1: Start the timer (1) | RW |
| TAZIE | Z-Phase Input Enable Bit | 0: Z-phase input disabled<br>1: Z-phase input enabled | RW |
| TA0TGL | Timer A0 Event/Trigger Select Bit | b7 b6<br>0 0 : Select the input to the TA0IN pin<br>0 1 : Select the overflow of TB2 (2)<br>1 0 : Select the overflow of TA4 (2)<br>1 1 : Select the overflow of TA1 (2) | RW |
| TA0TGH | | | RW |

Notes:
1. This bit is read as 0.
2. The timer overflows or underflows.

**Figure 16.8 ONSF Register**

## Trigger Select Register

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| TRGSR | 0343h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TA1TGL | Timer A1 Event/Trigger Select Bit | b1 b0<br>0 0 : Select the input to the TA1IN pin<br>0 1 : Select the overflow of TB2 (1)<br>1 0 : Select the overflow of TA0 (1)<br>1 1 : Select the overflow of TA2 (1) | RW |
| TA1TGH | | | RW |
| TA2TGL | Timer A2 Event/Trigger Select Bit | b2 b3<br>0 0 : Select the input to the TA2IN pin<br>0 1 : Select the overflow of TB2 (1)<br>1 0 : Select the overflow of TA1 (1)<br>1 1 : Select the overflow of TA3 (1) | RW |
| TA2TGH | | | RW |
| TA3TGL | Timer A3 Event/Trigger Select Bit | b4 b5<br>0 0 : Select the input to the TA3IN pin<br>0 1 : Select the overflow of TB2 (1)<br>1 0 : Select the overflow of TA2 (1)<br>1 1 : Select the overflow of TA4 (1) | RW |
| TA3TGH | | | RW |
| TA4TGL | Timer A4 Event/Trigger Select Bit | b6 b7<br>0 0 : Select the input to the TA4IN pin<br>0 1 : Select the overflow of TB2 (1)<br>1 0 : Select the overflow of TA3 (1)<br>1 1 : Select the overflow of TA0 (1) | RW |
| TA4TGH | | | RW |

Note:
1. The timer overflows or underflows.

**Figure 16.9 TRGSR Register**

## Count Source Prescaler Register

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| TCSPR | 035Fh | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CNT0 | Divide Ratio Select Bit (1) | f2n is either the main clock or peripheral clock source divided by 2n. If n = 0, the clock is not divided (n = setting value) | RW |
| CNT1 | | | RW |
| CNT2 | | | RW |
| CNT3 | | | RW |
| — (b6-b4) | Reserved | Should be written with 0 | RW |
| CST | Divider Operation Enable Bit | 0: Stop divider operation<br>1: Start divider operation | RW |

Note:
1. The CST bit should be set to 0 to rewrite bits CNT3 to CNT0.

**Figure 16.10 TCSPR Register**

### 16.1.1 Timer Mode

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

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

| Item | Specification |
|---|---|
| Count sources | f1, f8, f2n, or fC32 |
| Count operations | • Decrement counting<br>• If the timer counter underflows, the reload register setting is reloaded into the counter to resume counting |
| Divide ratio | $\frac{1}{n+1}$ n: TAi register setting value, 0000h to FFFFh |
| Count start condition | The TAiS bit in the TABSR register is set to 1 (count starts) |
| Count stop condition | The TAiS bit in the TABSR register is set to 0 (count stops) |
| Interrupt request generating timing | When the timer counter underflows |
| TAiIN pin function | A programmable I/O port or a gate input |
| TAiOUT pin function | A programmable I/O port or a pulse output |
| Read from timer | The TAi register indicates a counter value |
| Write to timer | • While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both reload register and the counter<br>• While the timer counter is running, the value written to the TAi register is written to the reload register (It is transferred to the counter at the next reload timing) |
| Selectable functions | • Gate function<br>Input signal to the TAiIN pin can control to start/stop counting<br>• Pulse output function<br>The polarity of the TAiOUT pin is inverted whenever the timer counter underflows.<br>A low is output while the TAiS bit holds 0 (count stops) |

## Timer Ai Mode Register (i = 0 to 4) (timer mode)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | 0 | | | 0 | 0 | 0 |

| Symbol | Address | Reset Value |
|---|---|---|
| TA0MR to TA4MR | 0356h, 0357h, 0358h, 0359h, 035Ah | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TMOD0 | Operating Mode Select Bit | b1 b0<br>0 0 : Timer mode | RW |
| TMOD1 | | | RW |
| —<br>(b2) | Reserved | Should be written with 0 | RW |
| MR1 | Gate Function Select Bit | b4 b3<br>0 X : No gate function (1)<br>(TAiIN pin functions as programmable I/O port)<br>1 0 : Count only while the TAiIN pin is held low<br>1 1 : Count only while the TAiIN pin is held high | RW |
| MR2 | | | RW |
| MR3 | Should be written with 0 in timer mode | | RW |
| TCK0 | Count Source Select Bit | b7 b6<br>0 0 : f1<br>0 1 : f8<br>1 0 : f2n<br>1 1 : fC32 | RW |
| TCK1 | | | RW |

Note:
1. X can be set to either 0 or 1.

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

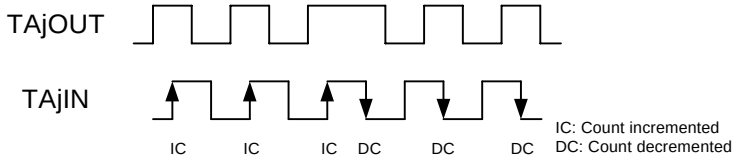
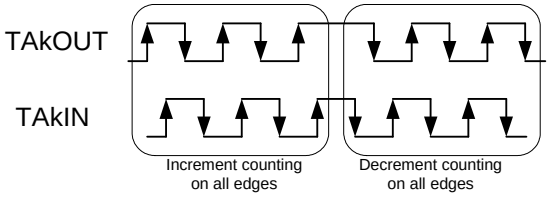
### 16.1.2 Event Counter Mode

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

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

| Item | Specification |
|---|---|
| Count sources | • External signal applied to the TAiIN pin (valid edge is selectable by a program)<br>• The overflow or underflow signal of timer B2, timer Aj (j = i - 1, or j = 4 if i = 0), and timer Ak (k = i + 1, or k = 0 if i = 4) |
| Count operations | • Increment/decrement counting can be switched by an external signal or program<br>• If the timer counter underflows or overflows, the reload register setting is reloaded into the counter to resume counting. In the free-running count operation, the timer counter continues counting without reloading |
| Divide ratio | • $\frac{1}{FFFFh - n + 1}$ for increment counting<br>• $\frac{1}{n + 1}$ for decrement counting<br>n: TAi register setting value, 0000h to FFFFh |
| Count start condition | The TAiS bit in the TABSR register is set to 1 (count starts) |
| Count stop condition | The TAiS bit in the TABSR register is set to 0 (count stops) |
| Interrupt request generating timing | When the timer counter overflows or underflows |
| TAiIN pin function | A programmable I/O port or a count source input |
| TAiOUT pin function | A programmable I/O port, a pulse output, or an input for increment/decrement count switching |
| Read from timer | The TAi register indicates a counter value |
| Write to timer | • While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both reload register and the counter<br>• While the timer counter is running, the value written to the TAi register is written to the reload register (It is transferred to the counter at the next reload timing) |
| Selectable functions | • Free-running count function<br>The reload register setting is not reloaded even if the timer counter overflows or underflows<br>• Pulse output function<br>The polarity of the TAiOUT pin is inverted whenever the timer counter overflows or underflows.<br>A low is output while the TAiS bit holds 0 (count stops) |

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

| Item | Specification |
|---|---|
| Count sources | Two-phase pulse signal applied to pins TAiIN and TAiOUT |
| Count operations | • Increment/decrement counting can be switched by a two-phase pulse signal<br>• If the timer counter underflows or overflows, the reload register setting is reloaded into the counter to resume counting. In the free-running count operation, the timer counter continues counting without reloading |
| Divide ratio | • $\frac{1}{FFFFh - n + 1}$ for increment counting<br>• $\frac{1}{n + 1}$ for decrement counting n: TAi register setting value, 0000h to FFFFh |
| Count start condition | The TAiS bit in the TABSR register is set to 1 (count starts) |
| Count stop condition | The TAiS bit in the TABSR register is set to 0 (count stops) |
| Interrupt request generating timing | When the timer counter overflows or underflows |
| TAiIN pin function | A two-phase pulse input |
| TAiOUT pin function | A two-phase pulse input |
| Read from timer | The TAi register indicates a counter value |
| Write to timer | • While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both reload register and the counter<br>• While the timer counter is running, the value written to the TAi register is written to the reload register (It is transferred to the counter at the next reload timing) |
| Selectable functions (1) | • Normal processing operation (timers A2 and A3)<br>While the input signal applied to the TAjOUT pin (j = 2 or 3) is held high, the timer increments the count on the rising edge of the TAjIN pin and decrements the count on the falling edge<br>  <br>• Quadrupled processing operation (timers A3 and A4)<br>When the input signal applied to the TAkOUT pin (k = 3 or 4) is held high on the rising edge of the TAkIN pin, the timer increments the count on both the rising and falling edges of pins TAkOUT and TAkIN.<br>When the signal is held high on the falling edge of the TAkIN pin, the timer decrements the count on both the rising and falling edges of pins TAkOUT and TAkIN<br>  <br>• Counter reset by Z-phase input (the timer A3)<br>The counter value is set to 0 by Z-phase input |

Note:
1. Only the timer A3 is available for any selectable functions. The timer A2 is exclusively for the normal processing operation and the timer A4 is for the quadrupled processing operation, respectively.

## Timer Ai Mode Register (i = 0 to 4) (event counter mode)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | 0 | | | 0 | 0 | 1 |

| Symbol | Address | Reset Value |
|---|---|---|
| TA0MR to TA4MR | 0356h, 0357h, 0358h, 0359h, 035Ah | 0000 0000b |

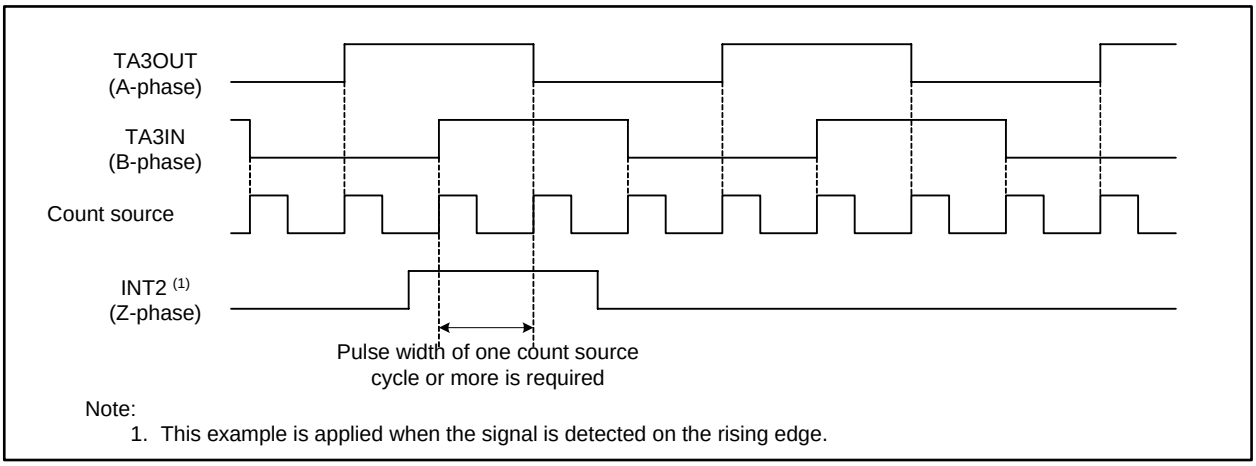
| Bit Symbol | Bit Name | Function (without two-phase pulse signal processing) | Function (with two-phase pulse signal processing) | RW |
|---|---|---|---|---|
| TMOD0 | Operating Mode Select Bit | b1 b0<br>0 1 : Event counter mode (1) | | RW |
| TMOD1 | | | | RW |
| — (b2) | Reserved | Should be written with 0 | | RW |
| MR1 | Count Polarity Select Bit (2) | 0: Count falling edges<br>1: Count rising edges | Should be written with 0 | RW |
| MR2 | Increment/Decrement Count Switching Source Select Bit | 0: UDF register setting<br>1: Input signal to the TAiOUT pin (3) | Should be written with 1 | RW |
| MR3 | Should be written with 0 in event counter mode | | | RW |
| TCK0 | Count Operation Type Select Bit | 0: Reloading<br>1: Free-running | | RW |
| TCK1 | Two-phase Pulse Processing Operation Select Bit (4, 5) | Should be written with 0 | 0: Normal processing operation<br>1: Quadrupled processing operation | RW |

Notes:
1. Bits TAiTGH and TAiTGL in the ONSF or TRGSR register select the count source in event counter mode.
2. This bit setting is enabled only when an external signal is counted.
3. The timer decrements the count when the input signal to the TAiOUT pin is held low and increments the count when the signal is held high.
4. The TCK1 bit is enabled only in the TA3MR register.
5. For two-phase pulse signal processing, the TAjP bit in the UDF register (j = 2 to 4) should be set to 1 (two-phase pulse signal processing enabled) and bits TAiTGH and TAiTGL should be set to 00b (the input to the TAjIN pin).

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

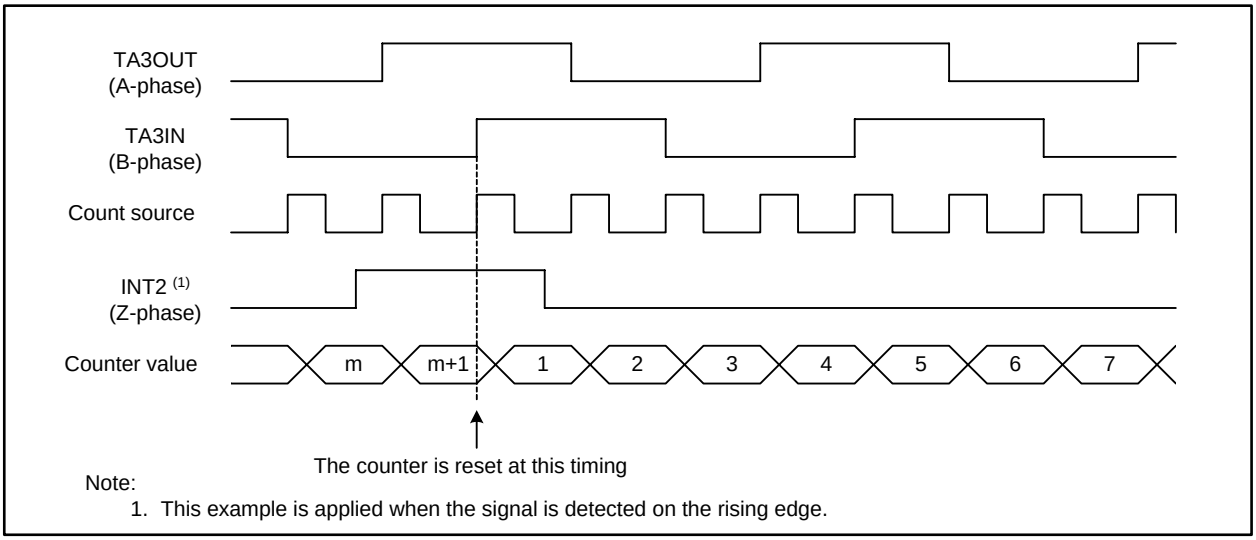
### 16.1.2.1 Counter Reset by Two-phase Pulse Signal Processing

A Z-phase input signal resets the timer counter when a two-phase pulse signal is being processed.
This function can be used under the following conditions: timer A3 event counter mode, two-phase pulse signal processing, free-running count operation type, and quadrupled processing. The Z-phase signal is applied to the $\overline{\text{INT2}}$ pin.
When the TAZIE bit in the ONSF register is set to 1 (Z-phase input enabled), the reset of timer counter by Z-phase input is enabled. To reset the counter, the TA3 register should be set to 0000h beforehand.
A Z-phase signal applied to the $\overline{\text{INT2}}$ pin is detected on a edge. The edge polarity is selected using the POL bit in the INT2IC register. The Z-phase signal should be input in order to have a pulse width of one count source cycle for timer A3 or more. Figure 16.13 shows the two-phase pulse (phases A and B) and the Z-phase.
The timer counter is reset at the initial count source input after a Z-phase input is detected. Figure 16.14 shows the counter reset timing.
If the timer A3 overflows or underflows during a reset processing by the Z-phase input, two timer A3 interrupt requests are sequentially generated. To avoid this situation, the timer A3 interrupt request should not be used when this function is in use.




Note:
1. This example is applied when the signal is detected on the rising edge.

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




Note:
1. This example is applied when the signal is detected on the rising edge.

**Figure 16.14 Counter Reset Timing**

### 16.1.3 One-shot Timer Mode

In one-shot timer mode, the timer operates only once for each trigger. Table 16.4 lists specifications of one-shot timer mode. Once a trigger occurs, the timer starts and operates for a given period. Figure 16.15 shows registers TA0MR to TA4NR in this mode.

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

| Item | Specification |
|---|---|
| Count sources | f1, f8, f2n, or fC32 |
| Count operations | • Decrement counting<br>• When the timer counter reaches 0000h, it stops running after the reload register setting is reloaded<br>• If a trigger occurs while counting, the reload register setting is reloaded into the counter to continue counting |
| Divide ratio | $\frac{1}{n}$ n: TAi register setting value, 0000h to FFFFh<br>(Note that the timer counter does not run if n = 0000h) |
| Count start conditions | The TAiS bit in the TABSR register is set to 1 (count starts) and any of following triggers occurs:<br>• An external trigger applied to the TAiIN pin<br>• The overflow or underflow signal of timer B2, timer Aj (j = i - 1, or j = 4 if i = 0), or timer Ak (k = i + 1, or k = 0 if i = 4)<br>• The TAiOS bit in the ONSF register is set to 1 (the timer started) |
| Count stop conditions | • The timer counter reaches 0000h and the reload register setting is reloaded<br>• The TAiS bit in the TABSR register is set to 0 (count stops) |
| Interrupt request generating timing | When the timer counter reaches 0000h |
| TAiIN pin function | A programmable I/O port or a trigger input |
| TAiOUT pin function | A programmable I/O port or a pulse output |
| Read from timer | The TAi register indicates undefined value |
| Write to timer | • While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both reload register and the counter<br>• While the timer counter is running, the value written to the TAi register is written to the reload register (It is transferred to the counter at the next reload timing) |
| Selectable function | • Pulse output function<br>A low is output while the timer counter is not running and a high is output while the timer counter is running |

## Timer Ai Mode Register (i = 0 to 4) (one-shot timer mode)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | 0 | | | 0 | 1 | 0 |

| Symbol | Address | Reset Value |
|---|---|---|
| TA0MR to TA4MR | 0356h, 0357h, 0358h, 0359h, 035Ah | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TMOD0 | Operating Mode Select Bit | b1 b0<br>1 0 : One-shot timer mode | RW |
| TMOD1 | | | RW |
| — (b2) | Reserved | Should be written with 0 | RW |
| MR1 | External Trigger Select Bit (1) | 0: Falling edge of input signal to the TAiIN pin<br>1: Rising edge of input signal to the TAiIN pin | RW |
| MR2 | Trigger Select Bit | 0: The TAiOS bit enabled<br>1: Selected using bits TAiTGH and TAiTGL | RW |
| MR3 | Should be written with 0 in one-shot timer mode | | RW |
| TCK0 | Count Source Select Bit | b7 b6<br>0 0 : f1<br>0 1 : f8<br>1 0 : f2n<br>1 1 : fC32 | RW |
| TCK1 | | | RW |

Note:
1. The MR1 bit setting is enabled only when bits TAiTGH and TAiTGL in the TRGSR register are set to 00b (the input to the TAiIN pin). This bit can be set to either 0 or 1 when bits TAiTGH and TAiTGL are set to 01b (the overflow or underflow of TB2), 10b (the overflow or underflow of TAi), or 11b (the overflow or underflow of TAi).

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

### 16.1.4 Pulse-width Modulation Mode

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

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

| Item | Specification |
|---|---|
| Count sources | f1, f8, f2n, or fC32 |
| Count operations | • Decrement counting (the timer counter functions as an 8-bit or a 16-bit pulse-width modulator)<br>• The reload register setting is reloaded on the rising edge of PWM pulse to resume counting<br>• The timer is not affected by a trigger that is generated while the counter is running |
| 16-bit PWM | • High level width: $\frac{n}{fj}$ n: TAi register setting value, 0000h to FFFEh<br>fj: Count source frequency<br>• Cycle: $\frac{2^{16}-1}{fj}$ fixed |
| 8-bit PWM | • High level width: $\frac{n \times (m+1)}{fj}$<br>• Cycle: $\frac{(2^8-1) \times (m+1)}{fj}$<br>n: TAi register (upper byte) setting value, 00h to FEh<br>m: TAi register (lower byte) setting value, 00h to FFh |
| Count start conditions | • The TAiS bit in the TABSR register is set to 1 (count starts)<br>• The TAiS bit is set to 1 and an external trigger applied to the TAiIN pin<br>• The TAiS bit is set to 1 and any of following triggers occurs:<br>The overflow or underflow signal of timer B2, timer Aj (j = i - 1, or j = 4 if i = 0), or timer Ak (k = i + 1, or k = 0 if i = 4) |
| Count stop condition | The TAiS bit in the TABSR register is set to 0 (count stops) |
| Interrupt request generating timing | On the falling edge of the PWM pulse |
| TAiIN pin function | A programmable I/O port or a trigger input |
| TAiOUT pin function | A pulse output |
| Read from timer | The TAi register indicates undefined value |
| Write to timer | • While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both reload register and the counter<br>• While the timer counter is running, the value written to the TAi register is written to the reload register (it is transferred to the counter at the next reload timing) |

## Timer Ai Mode Register (i = 0 to 4) (pulse-width modulation mode)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 1 |

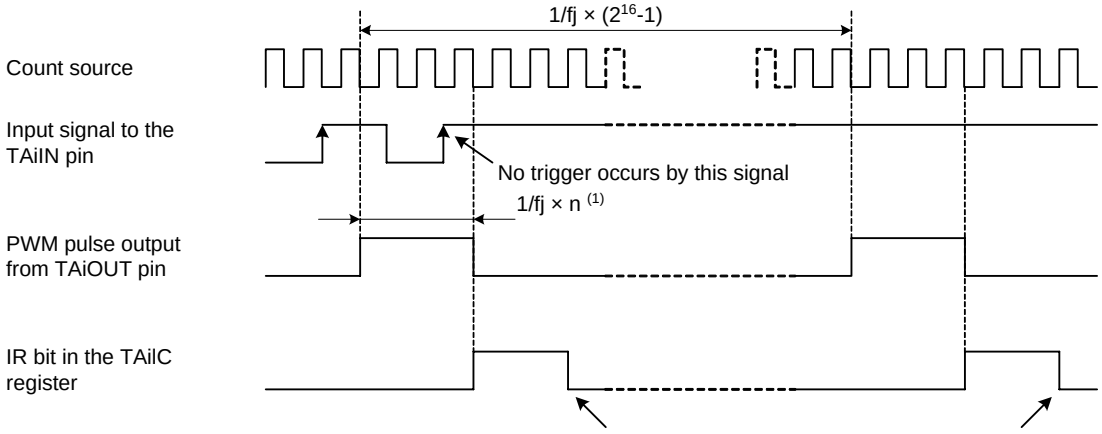
| Symbol | Address | Reset Value |
|---|---|---|
| TA0MR to TA4MR | 0356h, 0357h, 0358h, 0359h, 035Ah | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TMOD0 | Operating Mode Select Bit | b1 b0<br>1 1 : Pulse-width modulation (PWM) mode | RW |
| TMOD1 | | | RW |
| — (b2) | Reserved | Should be written with 0 | RW |
| MR1 | External Trigger Select Bit (1) | 0: Falling edge of input signal to the TAiIN pin<br>1: Rising edge of input signal to the TAiIN pin | RW |
| MR2 | Trigger Select Bit | 0: The TAiS bit enabled<br>1: Selected using bits TAiTGH and TAiTGL | RW |
| MR3 | 16-/8-Bit PWM Mode Select Bit | 0: 16-bit pulse-width modulator<br>1: 8-bit pulse-width modulator | RW |
| TCK0 | Count Source Select Bit | b7 b6<br>0 0 : f1<br>0 1 : f8<br>1 0 : f2n<br>1 1 : fC32 | RW |
| TCK1 | | | RW |

Note:
1. The MR1 bit setting is enabled only when bits TAiTGH and TAiTGL in the TRGSR register are set to 00b (input to the TAiIN pin). This bit can be set to either 0 or 1 when bits TAiTGH and TAiTGL are set to 01b (the overflow or underflow of TB2), 10b (the overflow or underflow of TAi), or 11b (the overflow or underflow of TAi).

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

When the reload register is set to 0003h and an external trigger (the rising edge of an input signal applied to the TAiIN pin) is selected

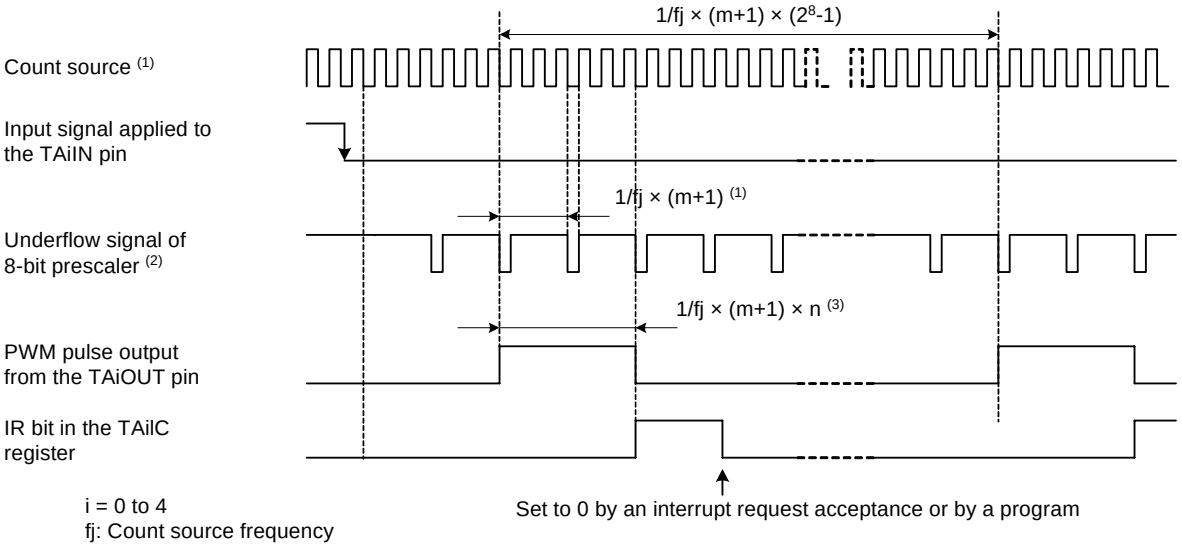



i = 0 to 4
fj: Count source frequency
(f1, f8, f2n, fC32)

Note:
1. n = 0000h to FFFEh

**Figure 16.17 16-bit Pulse-width Modulator Operation**

When the upper byte of the reload register is set to 02h, the lower byte is set to 02h and an external trigger (the falling edge of an input signal applied to the TAiIN pin) is selected




i = 0 to 4
fj: Count source frequency
(f1, f8, f2n, fC32)

Notes:
1. The 8-bit prescaler counts a count source.
2. The 8-bit pulse-width modulator counts underflow signals of the 8-bit prescaler.
3. m = 00h to FFh, n = 00h to FEh

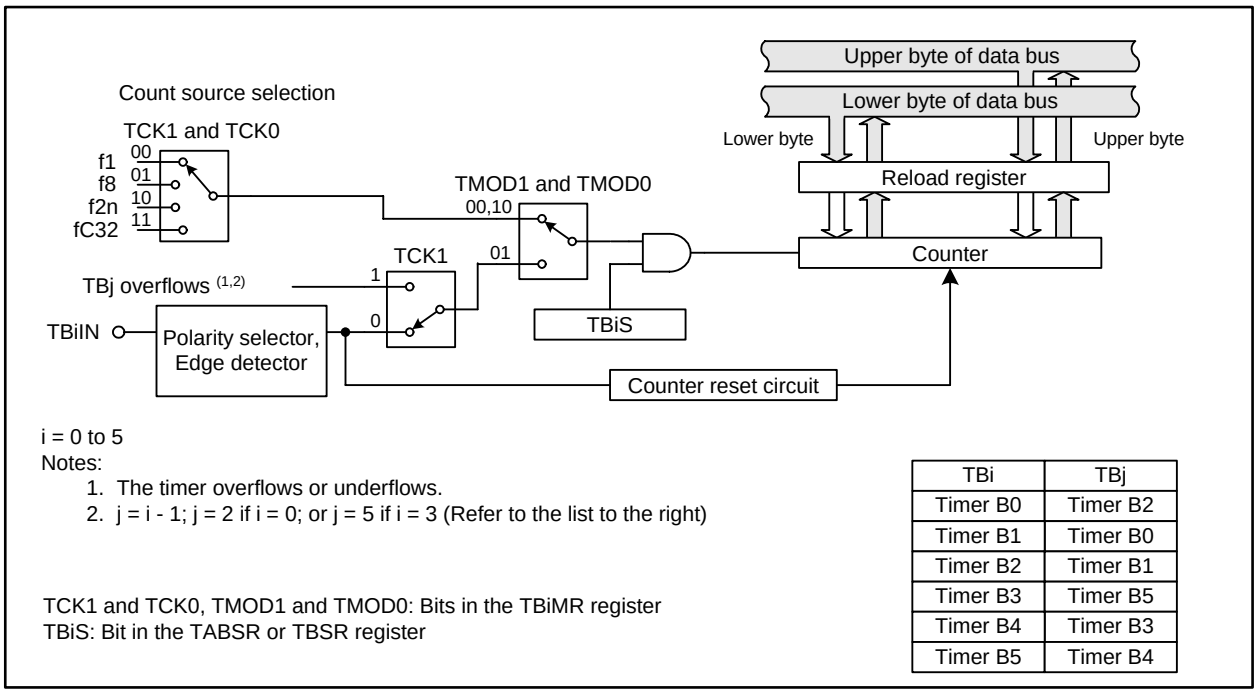
**Figure 16.18 8-bit Pulse-width Modulator Operation**

## 16.2 Timer B

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

The timer B supports three modes shown as below. A mode is selected using bits TMOD1 and TMOD0 in the TBiMR register (i = 0 to 5).

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




i = 0 to 5

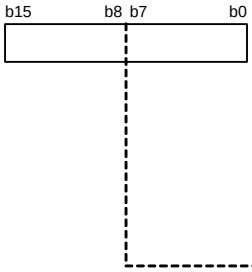
Notes:
1. The timer overflows or underflows.
2. j = i - 1; j = 2 if i = 0; or j = 5 if i = 3 (Refer to the list to the right)

TCK1 and TCK0, TMOD1 and TMOD0: Bits in the TBiMR register
TBiS: Bit in the TABSR or TBSR register

| TBi | TBj |
|---|---|
| Timer B0 | Timer B2 |
| Timer B1 | Timer B0 |
| Timer B2 | Timer B1 |
| Timer B3 | Timer B5 |
| Timer B4 | Timer B3 |
| Timer B5 | Timer B4 |

**Figure 16.19 Timer B Block Diagram**

## Timer Bi Register (i = 0 to 5) (1)



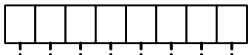

| Symbol | Address | Reset Value |
|---|---|---|
| TB0 to TB2 | 0351h-0350h, 0353h-0352h, 0355h-0354h | Undefined |
| TB3 to TB5 | 0311h-0310h, 0313h-0312h, 0315h-0314h | Undefined |

| Mode | Function | Setting Range | RW |
|---|---|---|---|
| Timer Mode | Divides the count source by n+1 (n = setting value) | 0000h to FFFFh | RW |
| Event Counter Mode | Divides the count source by n+1 (n = setting value) (2) | 0000h to FFFFh | RW |
| Pulse Period/Pulse-Width Measure Mode | Increments the count with the count source between one valid edge and another of TBiIN input pulse | — | RO |

Notes:
1. A 16-bit read/write access to this register should be performed.
2. The TBi register counts an external input pulse or an overflow and underflow of other timers.

**Figure 16.20 Registers TB0 to TB5**

## Timer Bi Mode Register (i = 0 to 5)



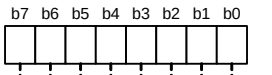

| Symbol | Address | Reset Value |
|---|---|---|
| TB0MR to TB2MR | 035Bh, 035Ch, 035Dh | 00XX 0000b |
| TB3MR to TB5MR | 031Bh, 031Ch, 031Dh | 00XX 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TMOD0 | Operating Mode Select Bit | b1 b0<br>0 0 : Timer mode<br>0 1 : Event counter mode<br>1 0 : Pulse period measure mode, pulse-width measure mode<br>1 1 : Do not use this combination | RW |
| TMOD1 | | | RW |
| MR0 | — | Function varies according to the operating mode (1, 2) | RW |
| MR1 | | | RW |
| MR2 | | | RW |
| MR3 | | | RW |
| TCK0 | Count Source Select Bit | Function varies according to the operating mode | RW |
| TCK1 | | | RW |

Notes:
1. The MR2 bit is available for registers TB0MR and TB3MR only.
2. The MR2 bit in registers TB1MR, TB2MR, TB4MR and TB5MR are unavailable on this MCU. This bit should be written with 0 and read as undefined value.

**Figure 16.21 Registers TB0MR to TB5MR**

## Count Start Register

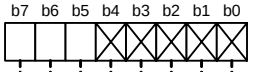



Symbol: TABSR
Address: 0340h
Reset Value: 0000 0000b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TA0S | Timer A0 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TA1S | Timer A1 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TA2S | Timer A2 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TA3S | Timer A3 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TA4S | Timer A4 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TB0S | Timer B0 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TB1S | Timer B1 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TB2S | Timer B2 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |

**Figure 16.22 TABSR Register**

## Count Start Register for Timers B3, B4 and B5




Symbol: TBSR
Address: 0300h
Reset Value: 000X XXXXb

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| —<br>(b4-b0) | No register bits; should be written with 0 and read as undefined value | | — |
| TB3S | Timer B3 Count Start Bit | 0: Stop counting<br>1: Start counting | RW |
| TB4S | Timer B4 Count Start Bit | 0: Stop counting<br>1: Start counting | RW |
| TB5S | Timer B5 Count Start Bit | 0: Stop counting<br>1: Start counting | RW |

**Figure 16.23 TBSR Register**

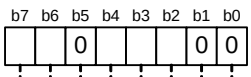
### 16.2.1 Timer Mode

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

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

| Item | Specification |
|---|---|
| Count sources | f1, f8, f2n, or fC32 |
| Count operations | • Decrement counting<br>• If the timer counter underflows, the reload register setting is reloaded into the counter to resume counting |
| Divide ratio | $\frac{1}{n+1}$ n: TBi register setting value, 0000h to FFFFh |
| Count start condition | The TBiS bit in the TABSR or TBSR register is set to 1 (count starts) |
| Count stop condition | The TBiS bit in the TABSR or TBSR register is set to 0 (count stops) |
| Interrupt request generating timing | When the timer counter underflows |
| TBiIN pin function | A programmable I/O port |
| Read from timer | The TBi register indicates a counter value |
| Write to timer | • While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TBi register is written to both reload register and the counter<br>• While the timer counter is running, the value written to the TBi register is written to the reload register (It is transferred to the counter at the next reload timing) |

## Timer Bi Mode Register (i = 0 to 5) (timer mode)




| Symbol | Address | Reset Value |
|---|---|---|
| TB0MR to TB2MR | 035Bh, 035Ch, 035Dh | 00XX 0000b |
| TB3MR to TB5MR | 031Bh, 031Ch, 031Dh | 00XX 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TMOD0 | Operating Mode Select Bit | b1 b0<br>0 0 : Timer mode | RW |
| TMOD1 | | | RW |
| MR0 | Disabled in timer mode.<br>Can be set to 0 or 1 | | RW |
| MR1 | | | RW |
| MR2 | In registers TB0MR and TB3MR;<br>Reserved; should be written with a 0 | | RW |
| | In registers TB1MR, TB2MR, TB4MR, and TB5MR;<br>No register bits; should be written with a 0 and read as undefined value | | — |
| MR3 | Disabled in timer mode. Should be written with a 0 and read as undefined value | | — |
| TCK0 | Count Source Select Bit | b7 b6<br>0 0 : f1<br>0 1 : f8<br>1 0 : f2n<br>1 1 : fC32 | RW |
| TCK1 | | | RW |

**Figure 16.24 Registers TB0MR to TB5MR in Timer Mode**

### 16.2.2 Event Counter Mode

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

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

| Item | Specification |
|---|---|
| Count sources | • External signal applied to the TBiIN pin (valid edge is selectable among the falling edge, the rising edge or the both by a program)<br>• The overflow or underflow signal of TBj (j = i - 1; j = 2 if i = 0; or j = 5 if i = 3) |
| Count operations | • Decrement counting<br>• If the timer counter underflows, the reload register setting is reloaded into the counter to resume counting |
| Divide ratio | $\frac{1}{n+1}$ n: TBi register setting value, 0000h to FFFFh |
| Count start condition | The TBiS bit in the TABSR or TBSR register is set to 1 (count starts) |
| Count stop condition | The TBiS bit in the TABSR or TBSR register is set to 0 (count stops) |
| Interrupt request generation timing | When the timer counter underflows |
| TBiIN pin function | A programmable I/O port or a count source input |
| Read from timer | The TBi register indicates a counter value |
| Write to timer | • While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TBi register is written to both reload register and the counter<br>• While the timer counter is running, the value written to the TBi register is written to the reload register (it is transferred to the counter at the next reload timing) |

## Timer Bi Mode Register (i = 0 to 5) (event counter mode)

| Symbol | Address | Reset Value |
|---|---|---|
| TB0MR to TB2MR | 035Bh, 035Ch, 035Dh | 00XX 0000b |
| TB3MR to TB5MR | 031Bh, 031Ch, 031Dh | 00XX 0000b |

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | 0 | | | | 0 | 1 |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TMOD0 | Operating Mode Select Bit | b1 b0<br>0 1 : Event counter mode | RW |
| TMOD1 | | | RW |
| MR0 | Count Polarity Select Bit (1) | b3 b2<br>0 0 : Count falling edges<br>0 1 : Count rising edges<br>1 0 : Count both edges<br>1 1 : Do not use this combination | RW |
| MR1 | | | RW |
| MR2 | In registers TB0MR and TB3MR; Reserved; should be written with 0 | | RW |
| | In registers TB1MR, TB2MR, TB4MR, and TB5MR; No register bit; should be written with 0 and read as undefined value | | — |
| MR3 | Disabled in event counter mode. Should be written with 0 and read as undefined value | | — |
| TCK0 | Disabled in event counter mode. Can be set to 0 or 1 | | RW |
| TCK1 | Event Clock Select Bit | 0: Input signal to the TBiIN pin<br>1: The overflow or underflow of TBj (2) | RW |

Notes:
1. These bit settings are enabled when the TCK1 bit is set to 0. When the TCK1 is set to 1, these bits can be set to either 0 or 1.
2. j = i - 1; j = 2 if i = 0; or j = 5 if i = 3.

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

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

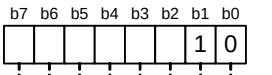
In pulse period/pulse-width measure mode, the timer measures pulse period or pulse width of an external signal. Table 16.8 lists specifications of pulse period/pulse-width measure mode. Figure 16.26 shows registers TB0MR to TB5MR in this mode. Figure 16.27 and Figure 16.28 respectively show an operation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

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

| Item | Specification |
|---|---|
| Count sources | f1, f8, f2n, or fC32 |
| Count operations | • Increment counting<br>• The counter value is transferred to the reload register on the valid edge of a pulse to be measured, then it is set to 0000h to resume counting |
| Count start condition | The TBiS bit in the TABSR or TBSR register is set to 1 (count starts) |
| Count stop condition | The TBiS bit in the TABSR or TBSR register is set to 0 (count stops) |
| Interrupt request generating timing | • On the valid edge of a pulse to be measured (1)<br>• When the timer counter overflows<br>(when the MR3 bit in the TBiMR register becomes 1 (overflow). (2)) |
| TBiIN pin function | A pulse input to be measured |
| Read from timer | The TBi register indicates a reload register value (measurement results) (3) |
| Write to timer | The value written to the TBi register is written to neither the reload register nor the counter |

Notes:
1. No interrupt request is generated when the pulse to be measured is applied on the initial valid edge after the timer counter starts.
2. To set the MR3 bit to 0 (no overflow), wait one or more count source cycles to write to the TBiMR register after the MR3 bit becomes 1 (overflow), while the TBiS bit is set to 1 (count starts).
3. The TBi register indicates undefined value until the pulse to be measured is applied on the second valid edge after the timer counter starts.

## Timer Bi Mode Register (i = 0 to 5) (pulse period/pulse-width measure mode)




| Symbol | Address | Reset Value |
|---|---|---|
| TB0MR to TB2MR | 035Bh, 035Ch, 035Dh | 00XX 0000b |
| TB3MR to TB5MR | 031Bh, 031Ch, 031Dh | 00XX 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TMOD0 | Operating Mode Select Bit | b1 b0<br>1 0 : Pulse period/pulse-width measure mode | RW |
| TMOD1 | | | RW |
| MR0 | Measure Mode Select Bit (1) | b3 b2<br>0 0 : Pulse period measurement 1<br>0 1 : Pulse period measurement 2<br>1 0 : Pulse-width measurement<br>1 1 : Do not use this combination | RW |
| MR1 | | | RW |
| MR2 | In registers TB0MR and TB3MR;<br>Reserved; should be written with 0 | | RW |
| | In registers TB1MR, TB2MR, TB4MR, and TB5MR;<br>No register bit; should be written with 0 and read as undefined value | | — |
| MR3 | Timer Bi Overflow Flag (2) | 0: No overflow<br>1: Overflow | RO |
| TCK0 | Count Source Select Bit | b7 b6<br>0 0 : f1<br>0 1 : f8<br>1 0 : f2n<br>1 1 : fC32 | RW |
| TCK1 | | | RW |

Notes:

1. Bits MR1 and MR0 select from the following measure modes:
   Pulse period measurement 1 (bits MR1 and MR0 = 00b):
   Measures between a falling edge and the next falling edge of a pulse
   Pulse period measurement 2 (bits MR1 and MR0 = 01b):
   Measures between a rising edge and the next rising edge of a pulse
   Pulse-width measurement (bits MR1 and MR0 = 10b):
   Measures between a falling edge and the next rising edge of a pulse and between the rising edge and the next falling edge of the pulse
2. The MR3 bit is undefined when the timer is reset.
   To set the MR3 bit to 0 (no overflow), wait one or more count source cycles to write to the TBiMR register after the MR3 bit becomes 1 (overflow), while the TBiS bit in the TABSR or TBSR register is set to 1 (count starts).
   The MR3 bit cannot be set to 1 by a program.

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

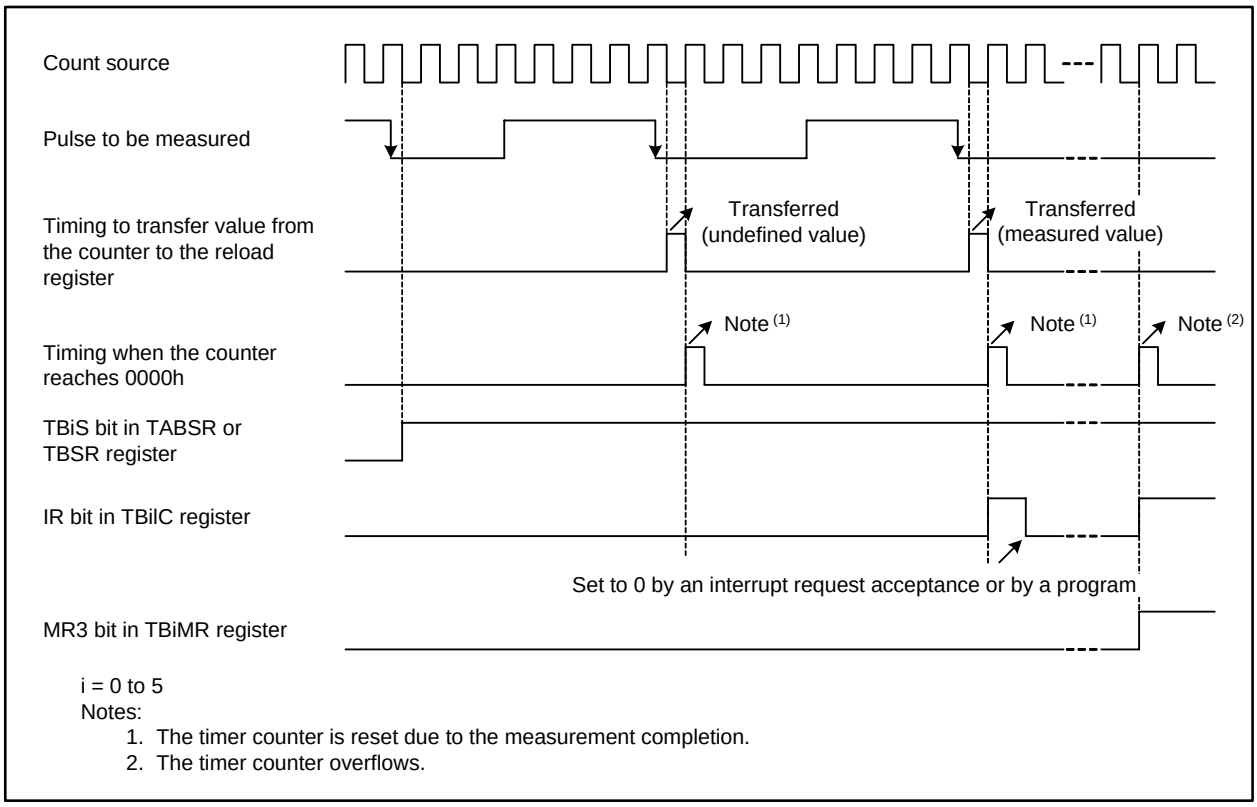



**Figure 16.27 Operation Example in Pulse Period Measurement**

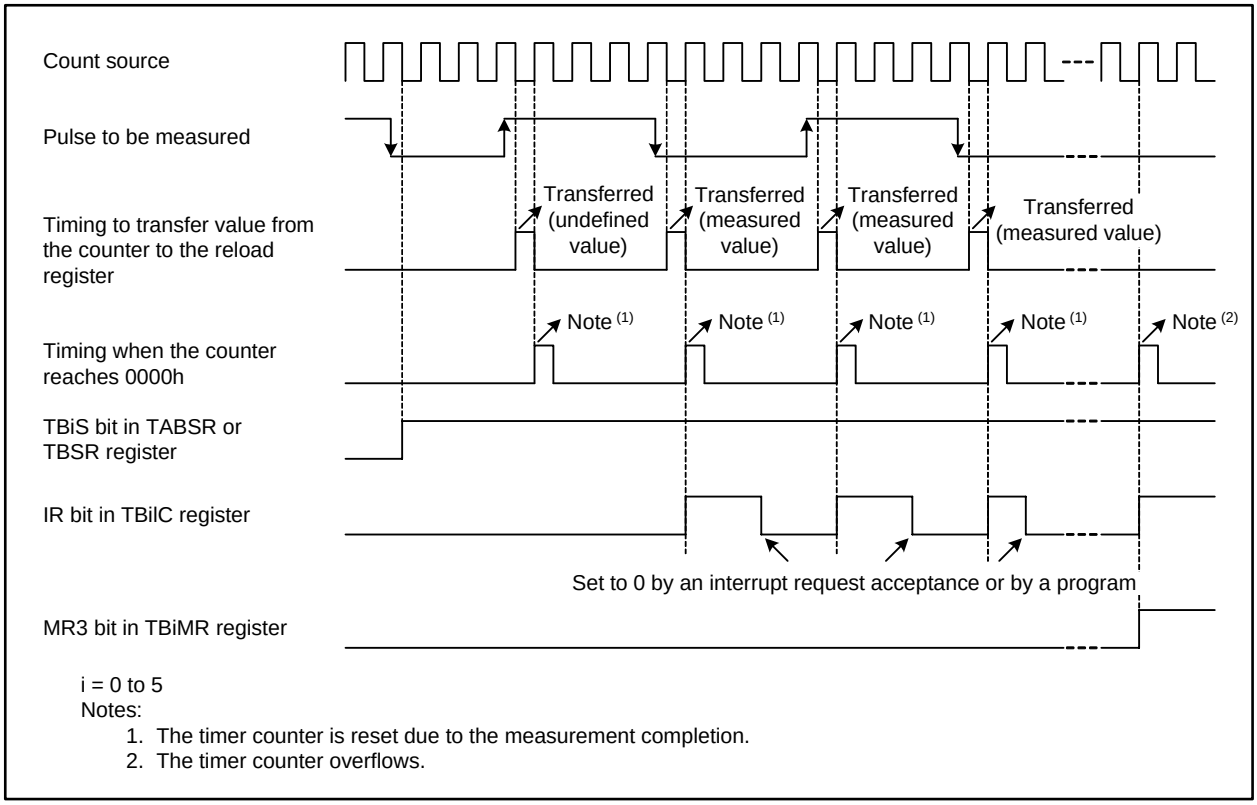



**Figure 16.28 Operation Example in Pulse-width Measurement**

## 16.3 Notes on Timers

### 16.3.1 Timer A and Timer B

All timers are stopped after a reset. To restart timers, configure parameters such as operating mode, count source, and counter value, then set the TAiS bit (i = 0 to 4) or TBjS bit (j = 0 to 5) in the TABSR or TBSR register to 1 (count starts).

The following registers and bits should be set while the TAiS bit or TBjS bit is 0 (count stops):

- Registers TAiMR and TBjMR
- The UDF register
- Bits TAZIE, TA0TGL, and TA0TGH in the ONSF register
- The TRGSR register

### 16.3.2 Timer A

#### 16.3.2.1 Timer Mode

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

#### 16.3.2.2 Event Counter Mode

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

#### 16.3.2.3 One-shot Timer Mode

- If the TAiS bit in the TABSR register is set to 0 (count stops) while the timer counter is running, the following operations are performed:
  - The timer counter stops and the setting value of the TAi register is reloaded.
  - A low signal is output at the TAiOUT pin.
  - The IR bit in the TAiIC register becomes 1 (interrupts requested) after one CPU clock cycle.

- One-shot timer is operated by an internal count source. When the trigger is an input to the TAiIN pin, the signal is output with a maximum of one count source clock delay after a trigger input to the TAiIN pin.

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt, set the IR bit to 0 after one of the settings below is done:
  - Select one-shot timer mode after a reset
  - Switch the operating mode from timer mode to one-shot timer mode, or
  - Switch the operating mode from event counter mode to one-shot timer mode

- If a retrigger occurs while counting, the timer counter decrements by one, reloads the setting value of the TAi register, and then continues counting. To generate a retrigger while counting, wait one or more count source cycles after the last trigger is generated.

- When an external trigger input is selected to start counting in timer A one-shot mode, do not provide an external retrigger for 300 ns before the timer counter reaches 0000h. Otherwise, it may stop counting.

#### 16.3.2.4 Pulse-width Modulation Mode

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt (i = 0 to 4), set the IR bit to 0 after one of the settings below is done:
  - Select pulse-width modulation mode after a reset
  - Switch the operating mode from timer mode to pulse-width modulation mode, or
  - Switch the operating mode from event counter mode to pulse-width modulation mode

- If the TAiS bit in the TABSR register is set to 0 (count stops) while PWM pulse is output, the following operations are performed:
  - The timer counter stops.
  - The output level at the TAiOUT pin changes from high to low. The IR bit becomes 1.
  - When a low signal is output at the TAiOUT pin, it remains unchanged. The IR bit does not change, either.

### 16.3.3 Timer B

#### 16.3.3.1 Timer Mode and Event Counter Mode

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

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

- To set the MR3 bit in the TBjMR register to 0 (no overflow), wait one or more count source cycles to write to the TBjMR register after the MR3 bit becomes 1 (overflow), while the TBjS bit is set to 1 (count starts).

- Use the IR bit in the TBjIC register to detect overflow. The MR3 bit is used only to determine an interrupt request source within the interrupt handler.

- The counter value is undefined when the timer counter starts. Therefore, the timer counter may overflow before a pulse to be measured is applied on the initial valid edge and cause a timer Bj interrupt request to be generated.

- When the pulse to be measured is applied on the initial valid edge after the timer counter starts, an undefined value is transferred to the reload register. At this time, the timer Bj interrupt request is not generated.

- The IR bit may become 1 (interrupt requested) by changing bits MR1 and MR0 in the TBjMR register after the timer counter starts. However, if the same value is rewritten to bits MR1 and MR0, the IR bit is not changed.

- Pulse width is repeatedly measured in pulse-width measure mode. Whether the measurement result is high-level width or not is determined by a program.

- If an overflow occurs simultaneously when a pulse is applied on the valid edge, this pulse is not recognized since an interrupt request is generated only once. Do not let an overflow occur in pulse period measure mode.

- In pulse-width measure mode, determine whether an interrupt source is a pulse applied on the valid edge or an overflow by reading the port level in the TBj interrupt handler.

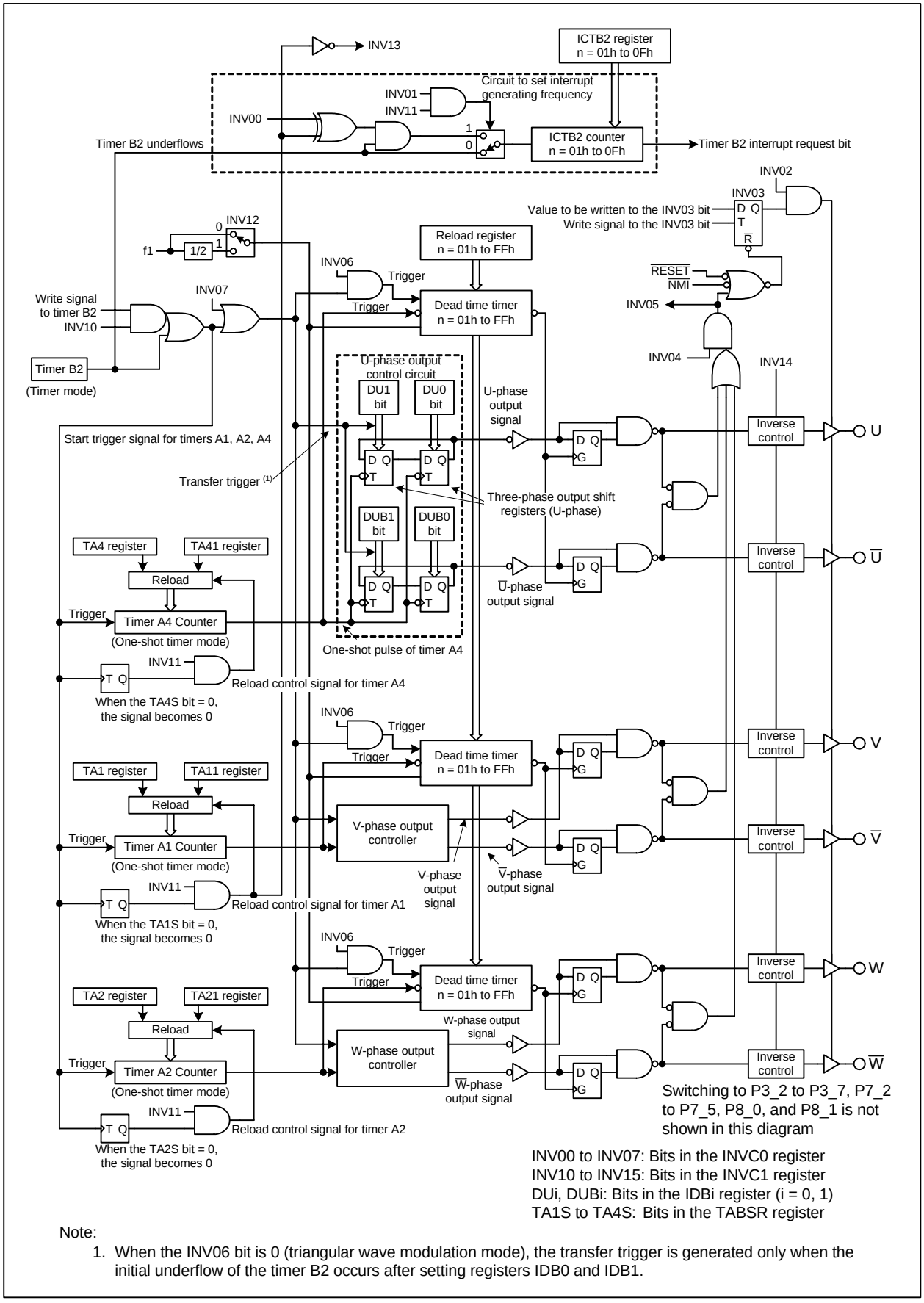
# 17. Three-phase Motor Control Timers

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

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

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

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

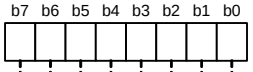



Note:

1. When the INV06 bit is 0 (triangular wave modulation mode), the transfer trigger is generated only when the initial underflow of the timer B2 occurs after setting registers IDB0 and IDB1.

**Figure 17.1 Three-phase Motor Control Timers Block Diagram**

## Three-Phase PWM Control Register 0 (1)




| Symbol | Address | Reset Value |
|---|---|---|
| INVC0 | 0308h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| INV00 | ICTB2 Count Condition Select Bit (2) | b1 b0<br>0 X : The underflow of timer B2<br>1 0 : The underflow of timer B2 when the reload control signal for timer A1 is set to 0 (3)<br>1 1 : The underflow of timer B2 when the reload control signal for timer A1 is set to 1 (3, 4) | RW |
| INV01 | | | RW |
| INV02 | Three-Phase Motor Control Timers Select Bit | 0: Do not use this function<br>1: Use this function (5, 6, 7) | RW |
| INV03 | Three-Phase Motor Control Timer Output Control Bit | 0: Disables the three-phase motor control timer output (7)<br>1: Enables the three-phase motor control timer output (8) | RW |
| INV04 | Simultaneous Conduction Prevention Bit | 0: Ignores simultaneous turn-on signal output<br>1: Disables simultaneous turn-on signal output | RW |
| INV05 | Simultaneous Conduction Detection Flag | 0: Not detected<br>1: Detected (9) | RO |
| INV06 | Modulation Mode Select Bit | 0: Triangular wave modulation mode<br>1: Sawtooth wave modulation mode (10) | RW |
| INV07 | Software Trigger Select Bit | A transfer trigger is generated when this bit is set to 1. When the INV06 bit is set to 1, another trigger to the dead time timer is also generated. This bit is read as 0 | RW |

Notes:
1. This register should be set after the PRC1 bit in the PRCR register is set to 1 (write enabled). Bits INV00 to INV02 and INV06 should be rewritten while timers A1, A2, A4, and B2 are stopped.
2. This bit is enabled when the INV11 bit in the INVC1 register is set to 1 (three-phase mode 1). When the INV11 bit is set to 0 (three-phase mode 0), the ICTB2 counter increments by one every time the timer B2 underflows irrespective of the INV00 and INV01 bit settings.
3. Set the ICTB2 register before setting the INV01 bit to 1. The timer A1 count start flag should be set to 1 before the initial timer B2 underflow occurs.
4. When the INV00 bit is set to 1, the first interrupt is generated if the timer B2 underflows n-1 times. (n is the value set in the ICTB2 counter). Subsequent interrupts are generated every n times the timer B2 underflows.
5. The INV02 bit should be set to 1 to operate the dead time timer, U-, V-, and W-phase output control circuits, and the ICTB2 counter.
6. After setting the INV02 bit to 1, pins should be configured first by the IOBC register then by the output function select register.
7. When the INV02 bit is set to 1 and the INV03 bit to 0, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$, including shared pins set by another output function, become high-impedance.
8. The INV03 bit becomes 0 when any of the following occurs:
   -Reset
   -Signals of both sides high and low are simultaneously switched active when the INV04 bit is set to 1
   -The INV03 bit is set to 0 by a program
   -The $\overline{NMI}$ pin goes from high to low when the PM24 bit in the PM2 register is set to 1 (NMI enabled)
9. This bit cannot be set to 1 by a program. The INV04 bit should be set to 0 to set this bit to 0.
10. When the INV06 bit is set to 1, the INV11 bit in the INVC1 register should be set to 0 (three-phase mode 0). In this case, the PWCON bit in the TB2SC register should be set to 0 (timer B2 register reloaded if the timer B2 underflows).

**Figure 17.2 INVC0 Register**

## Three-Phase PWM Control Register 1 (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | | | | | | | |

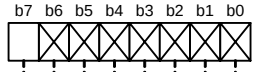
| Symbol | Address | Reset Value |
|---|---|---|
| INVC1 | 0309h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| INV10 | Timers A1, A2, and A4 Start Trigger Select Bit | 0: The underflow of timer B2<br>1: The underflow of timer B2 and a write to the TB2 register | RW |
| INV11 | Timers A1-1, A2-1, and A4-1 Control Bit | 0: Three-phase mode 0 (2, 3)<br>1: Three-phase mode 1 | RW |
| INV12 | Dead Time Timer Count Source Select Bit | 0: f1<br>1: f1 divided-by-2 | RW |
| INV13 | Carrier Wave Detection Flag (4) | 0: Timer A reload control signal is 0<br>1: Timer A reload control signal is 1 | RO |
| INV14 | Active Level Control Bit | 0: Active low output<br>1: Active high output | RW |
| INV15 | Dead Time Disable Bit | 0: Enables dead time<br>1: Disables dead time | RW |
| INV16 | Dead Time Timer Trigger Select Bit | 0: Falling edge of a one-shot pulse of timer (A4, A1, and A2) (5)<br>1: Rising edge of the three-phase output shift register (phases U, V, and W) | RW |
| — (b7) | Reserved | Should be written with 0 | RW |

Notes:
1. This register should be set after the PRC1 bit in the PRCR register is set to 1 (write enabled). Bits INV00 to INV02 and INV06 should be rewritten while timers A1, A2, A4, and B2 are stopped.
2. When the INV06 bit in the INVC0 register is set to 1 (sawtooth wave modulation mode), the INV11 bit should be set to 0 (three-phase mode 0).
3. When the INV11 bit is set to 0, the PWCON bit in the TB2SC register should be set to 0 (timer B2 register reloaded if the timer B2 underflows).
4. This bit setting is enabled when the INV06 bit is set to 0 (triangular wave modulation mode) and the INV11 bit is set to 1 (three-phase mode 1).
5. If the following conditions are all met, the INV16 bit should be set to 1:
   -The INV15 bit is set to 0 (dead time enabled)
   -The Dij bit (i = U, V, or W; j = 0 to 1) has a different value from the DiBj bit whenever the INV03 bit is set to 1 (three-phase motor control timer output enabled); the high- and low-side output signals have always inverse level on the period other than dead time.
   If any of the conditions above are not met, the INV16 bit should be set to 0.

**Figure 17.3 INVC1 Register**

## Three-Phase Output Buffer Control Register (1)




| Symbol | Address | Reset Value |
|---|---|---|
| IOBC | 40097h | 0XXX XXXXb |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b6-b0) | No register bits; should be written with 0 and read as undefined value | | — |
| TBSOUT | Three-Phase Output Pin Select Bit (2) | 0: Use pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ of ports P7 and P8<br>1: Use pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ of port P3 | RW |

Notes:
1. This register should be set after the PRC1 bit in the PRCR register is set to 1 (write enabled).
2. This bit should be set after the INV02 bit in the INVC0 register is set to 1. Then, the output function select register of corresponding port should be set. When the INV03 bit in the INVC0 register is set to 0, output pins for the three-phase motor control timers become high-impedance by the output enable control of output buffers. However, the output enable cannot be controlled only by the output function select register when more than two ports are assigned for output. Thus, a three-state output buffer should be selected using the TBSOUT bit.

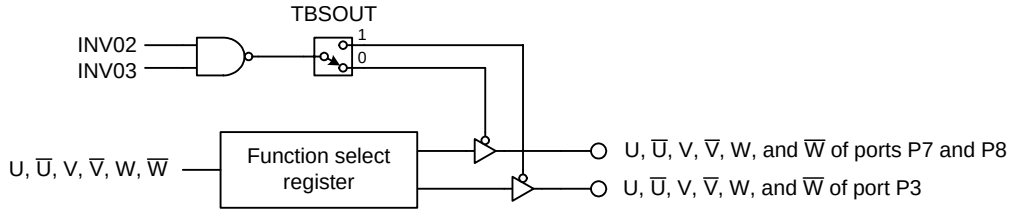
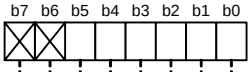



**Figure 17.4 IOBC Register**

## Three-Phase Output Buffer Register i (i = 0, 1) (1)



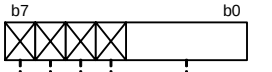

| Symbol | Address | Reset Value |
|---|---|---|
| IDB0, IDB1 | 030Ah, 030Bh | XX11 1111b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| DUi | U-Phase Output Buffer i | These bits should be written with an output level of the three-phase output shift register. The written value is reflected in each turn-on signal as follows:<br>0: Active (ON)<br>1: Inactive (OFF)<br>The bits are read as the value of the three-phase output shift register | RW |
| DUBi | $\overline{U}$-Phase Output Buffer i | | RW |
| DVi | V-Phase Output Buffer i | | RW |
| DVBi | $\overline{V}$-Phase Output Buffer i | | RW |
| DWi | W-Phase Output Buffer i | | RW |
| DWBi | $\overline{W}$-Phase Output Buffer i | | RW |
| — (b7-b6) | No register bits; should be written with 0 and read as undefined value | | — |

Note:
1. Values of registers IDB0 and IDB1 are transferred to the three-phase output shift register by a transfer trigger. The initial output signal level of each phase is determined by the value written in the IDB0 register after the transfer trigger occurs. After that, the output signal level is determined by the value written in the IDB1 register on the falling edge of a one-shot pulse of timers A1, A2, and A4.

**Figure 17.5 Registers IDB0 and IDB1**

## Timer B2 Interrupt Generating Frequency Set Counter (1, 2, 3)




| Symbol | Address | Reset Value |
|---|---|---|
| ICTB2 | 030Dh | Undefined |

| Function | Setting Range | RW |
|---|---|---|
| -When the INV01 bit = 0 (the ICTB2 counter increments every time the timer B2 underflows), the timer B2 interrupt request is generated every nth times the timer B2 underflows<br>-When the INV01 bit = 1 (the ICTB2 counter increments when the timer A1 reload control signal is set to 0 or 1 and the timer B2 underflows), the timer B2 interrupt request is generated every nth times the timer B2 underflows when the timer A1 reload control signal is 0 or 1<br>(n = setting value) | 01h to 0Fh | WO |
| No register bits; should be written with 0 | | — |

Notes:
1. The MOV instruction should be used to set the ICTB2 register.
2. When the INV01 bit in the INVC0 register is set to 1, set this register when the TB2S bit in the TABSR register is 0 (timer B2 count stops). When the INV01 bit is set to 0, this register can be set irrespective of the TB2S bit setting. However, do not set the register when the timer B2 underflows.
3. When the INV00 bit in the INVC0 register is set to 1, the first interrupt is generated if the timer B2 underflows n-1 times. Subsequent interrupts are generated every n times the timer B2 underflows. (n = setting value of the ICTB2 counter)

**Figure 17.6 ICTB2 Register**

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

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

**Table 17.2 Modulation Modes**

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

Note:

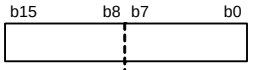
1. Transfer trigger: an underflow of timer B2 and a write to the INV07 bit, or a write to the TB2 register when the INV10 bit is set to 1.

## 17.2 Timer B2

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

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

Timer B2 Register (1)



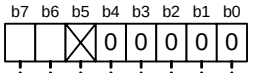

| Symbol | Address | Reset Value |
|---|---|---|
| TB2 | 0355h-0354h | Undefined |

| Function | Setting Range | RW |
|---|---|---|
| Divides the count source by n+1. Starts timers A1, A2, and A4 every time an underflow occurs (n = setting value) | 0000h to FFFFh | RW |

Note:
1. A 16-bit read/write access to this register should be performed.

**Figure 17.7 TB2 Register in Three-phase Motor Control Timers**

Timer B2 Mode Register

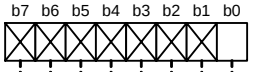



| Symbol | Address | Reset Value |
|---|---|---|
| TB2MR | 035Dh | 00XX 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TMOD0 | Operating Mode Select Bit | Should be written with 00b (timer mode) when using the three-phase motor control timers | RW |
| TMOD1 | | | RW |
| MR0 | Disabled in the three-phase motor control timers. Should be written with 0 and read as undefined value | | RW |
| MR1 | | | RW |
| MR2 | Should be written with 0 when using the three-phase motor control timers | | RW |
| MR3 | No register bit; should be written with 0 and read as undefined value | | — |
| TCK0 | Count Source Select Bit | b7 b6<br>0 0 : f1<br>0 1 : f8<br>1 0 : f2n<br>1 1 : fC32 | RW |
| TCK1 | | | RW |

**Figure 17.8 TB2MR Register in Three-phase Motor Control Timers**

Timer B2 Special Mode Register




| Symbol | Address | Reset Value |
|---|---|---|
| TB2SC | 035Eh | XXXX XXX0b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PWCON | Timer B2 Reload Timing Switching Bit (1) | 0: Reload the timer B2 register if the timer B2 underflows<br>1: Reload the timer B2 register every odd cycle of timer A output | RW |
| —<br>(b7-b1) | No register bits; should be written with 0 and read as undefined value | | — |

Note:
1. When the INV11 bit is set to 0 (three-phase mode 0) or the INV06 bit is set to 1 (sawtooth wave modulation mode), this bit should be set to 0.

**Figure 17.9 TB2SC Register**

## 17.3 Timers A4, A1, and A2

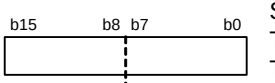
Timers A4, A1, and A2 are used for three-phase PWM output (U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$) control in the three-phase motor control timers.

These timers should operate in one-shot timer mode. Every time the timer B2 underflows, a trigger is input to timers A4, A1, and A2 to generate a one-shot pulse. If the values of registers TA4, TA1 and TA2 are rewritten every time a timer B2 interrupt is generated, the duty ratio of the PWM waveform can be varied.

In three-phase mode 1, the value of registers TAi and TAi-1 (i = 4, 1, 2) is alternately reloaded to the counter on each timer B2 interrupt, which halves timer B2 interrupt frequency. The sum of setting values for registers TAi and TAi1 should be identical to the setting value of the TB2 register in this mode.

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

Timer Ai/Timer Ai-1 Registers (i = 1, 2, 4) (1 to 6)




| Symbol | Address | Reset Value |
|---|---|---|
| TA1, TA2, TA4 | 0349h-0348h, 034Bh-034Ah, 034Fh-034Eh | Undefined |
| TA11, TA21, TA41 | 0303h-0302h, 0305h-0304h, 0307h-0306h | Undefined |

| Function | Setting Range | RW |
|---|---|---|
| The timer stops when the nth count source is counted after a start trigger is generated. Output signal for each phase is switched when timers A1, A2, and A4 stop (n = setting value) | 0000h to FFFFh | WO |

Notes:
1. A 16-bit write access to these registers should be performed.
2. When these registers are set to 0000h, the counter does not start, and no timer Ai interrupt request is generated.
3. The MOV instruction should be used to set these registers.
4. When the INV15 bit in the INVC1 register is set to 0 (dead time enabled), the turn-on output signal is switched to its active state with a delay. It is switched when the dead time timer stops.
5. When the INV11 bit in the INVC1 register is set to 0 (three-phase mode 0), the value of the TAi register is transferred to the reload register by a timer Ai start trigger. When the INV11 bit is set to 1 (three-phase mode 1), firstly the value of the TAi1 register is transferred to the reload register by a timer Ai start trigger. Then the value of the TAi register is transferred by the next timer Ai start trigger. After that the value of these registers TAi1 and TAi is alternately transferred to the reload register.
6. These registers should not be written when the timer B2 underflows.

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

## Timer Ai Mode Register (i = 1, 2, 4)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 0 | 0 | 1 | 0 |

| Symbol | Address | Reset Value |
|---|---|---|
| TA1MR, TA2MR, TA4MR | 0357h, 0358h, 035Ah | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TMOD0 | Operating Mode Select Bit | Should be written with 10b (one-shot timer mode) in three-phase motor control timers | RW |
| TMOD1 | | | RW |
| MR0 | Reserved | Should be written with 0 | RW |
| MR1 | External Trigger Select Bit | Should be written with 0 in three-phase motor control timers | RW |
| MR2 | Trigger Select Bit | Should be written with 1 (selected by the TRGSR register) in three-phase motor control timers | RW |
| MR3 | Should be written with 0 in three-phase motor control timers | | RW |
| TCK0 | Count Source Select Bit | b7 b6<br>0 0 : f1<br>0 1 : f8<br>1 0 : f2n<br>1 1 : fC32 | RW |
| TCK1 | | | RW |

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

## Trigger Select Register

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| TRGSR | 0343h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TA1TGL | Timer A1 Event/Trigger Select Bit | Should be set to 01b (the underflow of TB2) to use V-phase output control circuit | RW |
| TA1TGH | | | RW |
| TA2TGL | Timer A2 Event/Trigger Select Bit | Should be set to 01b (the underflow of TB2) to use W-phase output control circuit | RW |
| TA2TGH | | | RW |
| TA3TGL | Timer A3 Event/Trigger Select Bit | b5 b4<br>0 0 : Select the input to the TA3IN pin<br>0 1 : Select the overflow of TB2 (1)<br>1 0 : Select the overflow of TA2 (1)<br>1 1 : Select the overflow of TA4 (1) | RW |
| TA3TGH | | | RW |
| TA4TGL | Timer A4 Event/Trigger Select Bit | Should be set to 01b (the underflow of TB2) to use U-phase output control circuit | RW |
| TA4TGH | | | RW |

Note:
1. The timer overflows or underflows.

**Figure 17.12 TRGSR Register in Three-phase Motor Control Timers**

## Count Start Register

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| TABSR | 0340h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TA0S | Timer A0 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TA1S | Timer A1 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TA2S | Timer A2 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TA3S | Timer A3 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TA4S | Timer A4 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TB0S | Timer B0 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TB1S | Timer B1 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |
| TB2S | Timer B2 Count Start Bit | 0: Stop counter<br>1: Start counter | RW |

**Figure 17.13 TABSR Register**

## 17.4 Simultaneous Conduction Prevention and Dead Time Timer

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

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

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

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

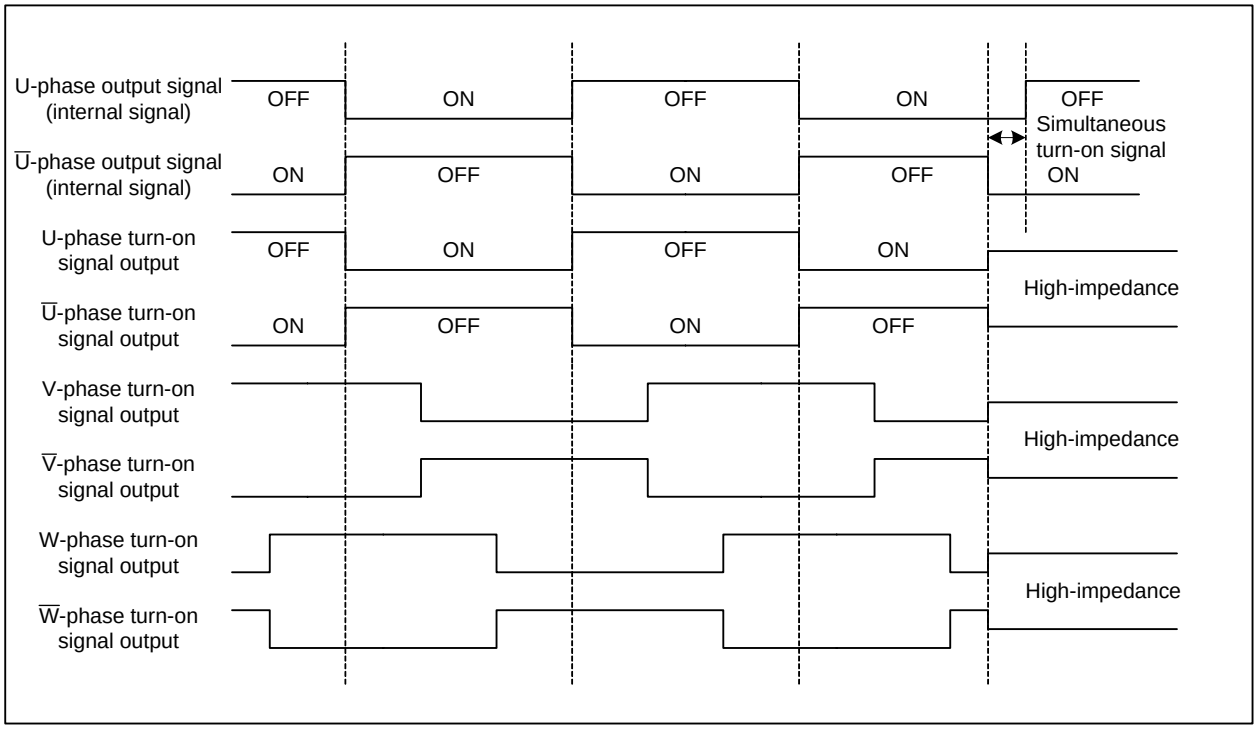



**Figure 17.14 Output Waveform When Simultaneous Turn-on Signal Output is Disabled**

Dead Time Timer (1, 2)

b7 b0

| Symbol | Address | Reset Value |
|---|---|---|
| DTT | 030Ch | Undefined |

| Function | Setting Range | RW |
|---|---|---|
| The dead time timer is a one-shot timer to delay the timing for a turn-on signal to be switched to its active state preventing a simultaneous conduction of high-side and low-side transistors.<br>The timer stops when counting a count source n times after a start trigger occurs (n = setting value) (3) | 01h to FFh | WO |

Notes:
1. The MOV instruction should be used to set this register.
2. This register setting is enabled when the INV15 bit in the INVC1 register is set to 0 (dead time enabled). No dead time can be set when the INV15 bit is set to 1 (dead time disabled).
3. The trigger and count source should be respectively selected using bits INV06 and INV12 in the INVC1 register.

**Figure 17.15 DTT Register**

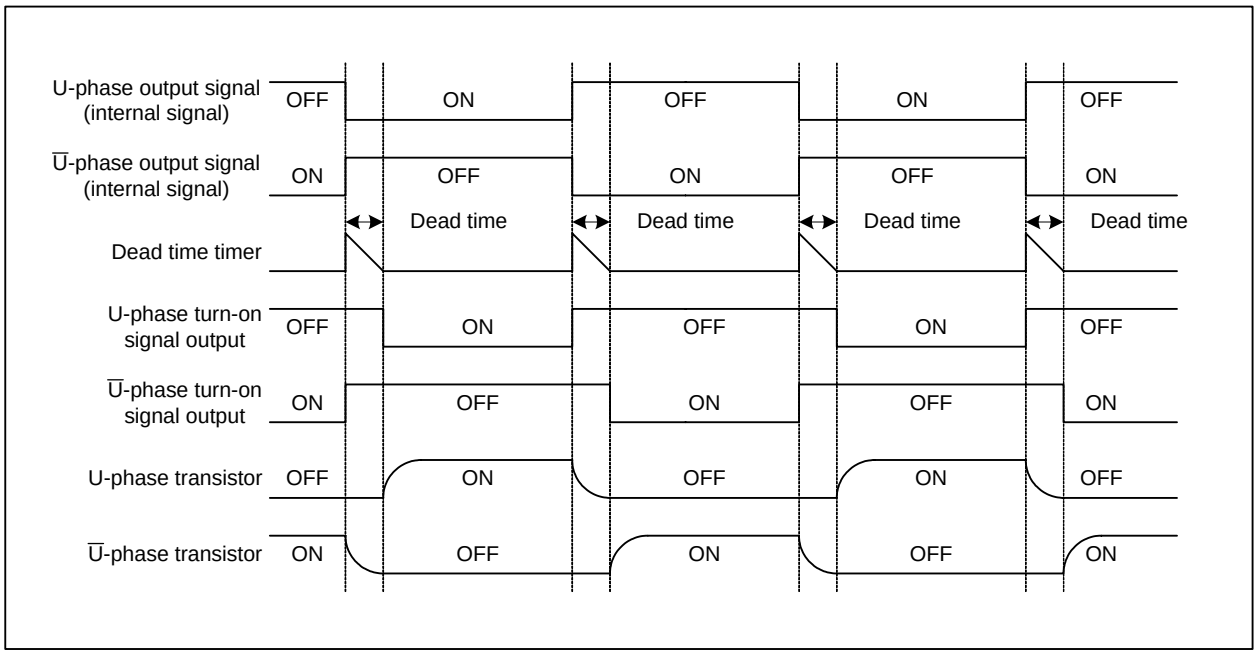
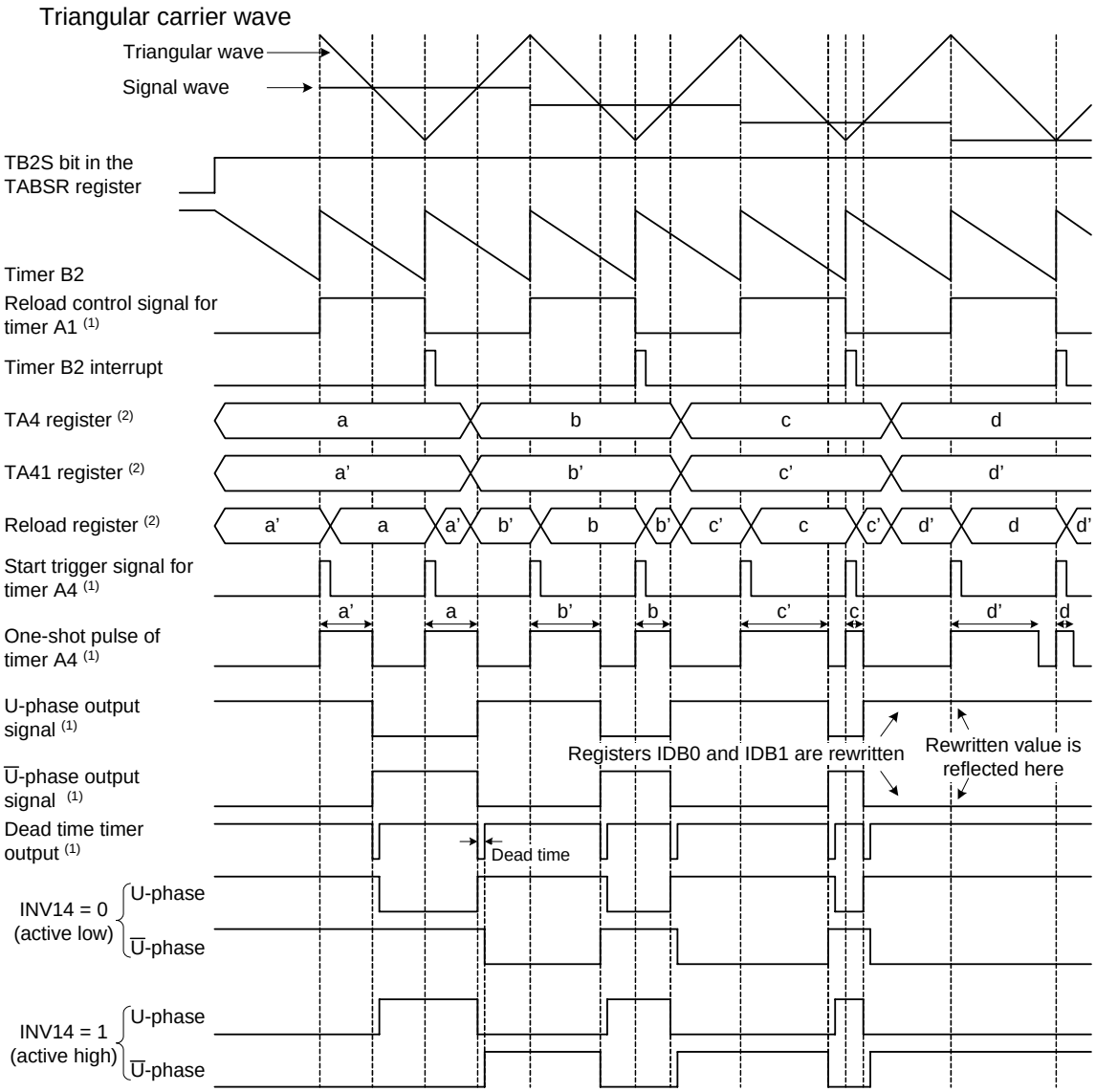



**Figure 17.16 Output Waveform on Using Dead Time Timer**

## 17.5 Three-phase Motor Control Timer Operation

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



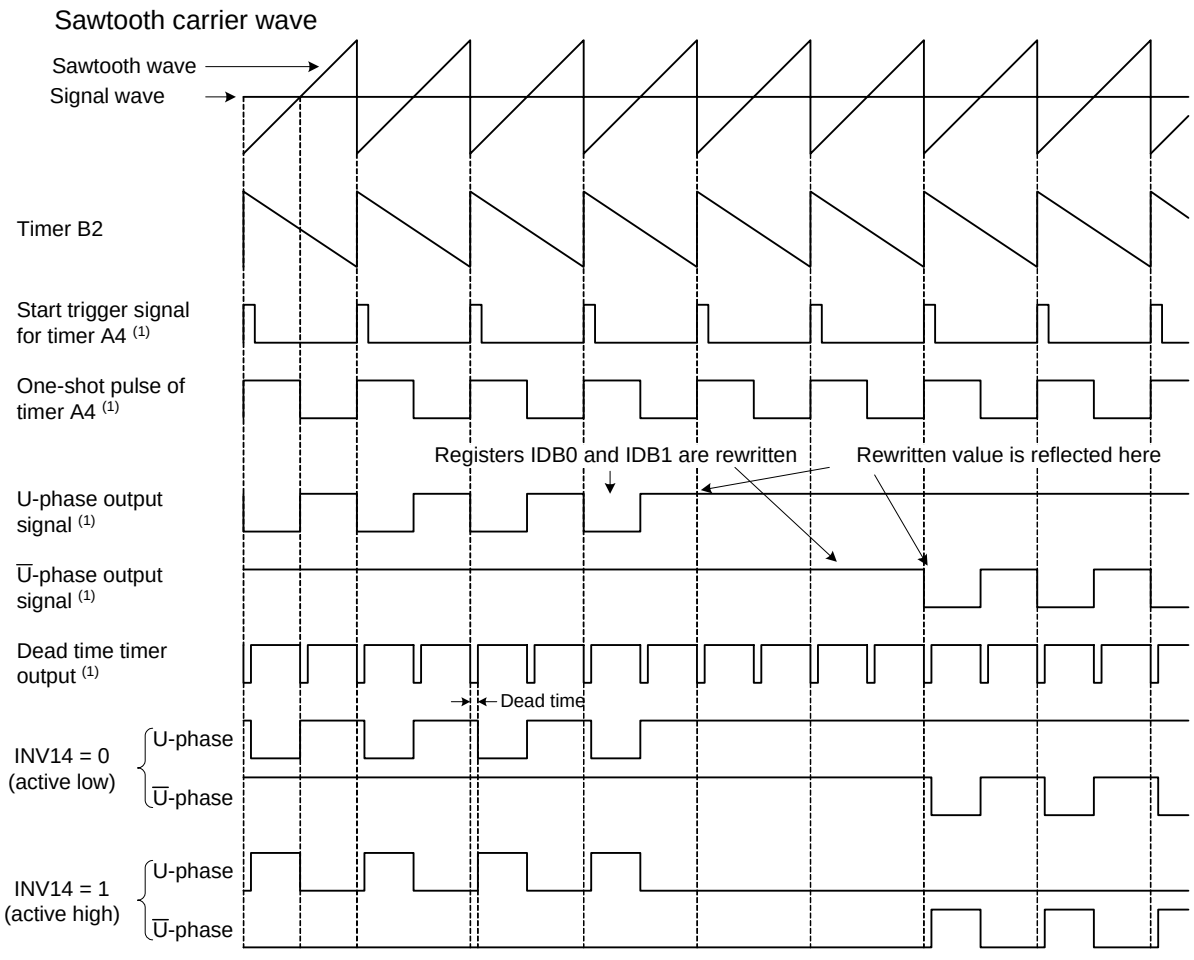

The above applies to INVC0 = 00XX11XXb (X varies depending on each system) and INVC1= 010XXXX0b.
The PWM output may vary as follows:

Case 1: When INV11 = 1 (three-phase mode 1)
- -INV01 = 0 and ICTB2 = 2h (timer B2 interrupt generated every 2nd time the timer B2 underflows), or INV01 = 1, INV00 = 1, and ICTB2 = 1h (timer B2 interrupt generated on the falling edge of the reload control signal for timer A)
- -The setting of registers TA4 and TA41 are varied every time a timer B2 interrupt is generated,
  Default value: TA41 = a', TA4 = a
  On the first timer B2 interrupt: TA41 = b', TA4 = b; the second time: TA41 = c', TA4 = c
- -Default value of registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1
  On the third time: DU0 = 1, DUB0 = 0, DU1 = 1, DUB1 = 0

Case 2: When INV11= 0 (three-phase mode 0)
- -INV01= 0 and ICTB2 = 1h (timer B2 interrupt generated every time timer B2 underflows)
- -TA4 register setting is varied every time a timer B2 interrupt is generated,
  Default value: TA4 = a'
  On the first timer B2 interrupt: TA4 = a; 2nd time: TA4 = b'; 3rd time: TA4 = b; 4th time: TA4 = c'; 5th time: TA4 = c
- -Default value of registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1
  On the sixth time: DU0 = 1, DUB0 = 0, DU1 = 1, DUB1 = 0

Notes:
1. Internal signal. Refer to the block diagram of three-phase motor control timers.
2. Applied when the INV11 bit in the INV1 register is set to 1 (three-phase mode 1).

**Figure 17.17 Triangular Wave Modulation Operation**




The above applies to INVC0= 01XX110Xb (X varies depending on each system) and INVC1= 000XXX00b.
This bit setting is applicable to turn-on control with a phase shift of 120 degrees.

The PWM output may vary as follows:
Default value of registers IDB0 and IDB1: DU0 = 0, DUB0 = 1, DU1 = 1, DUB1 = 1
On the third timer B2 interrupt: DU0 = 1, DUB0 = 1, DU1 = 1, DUB1 = 1
On the fifth timer B2 interrupt: DU0 = 1, DUB0 = 0, DU1 = 1, DUB1 = 1

Note:
1. Internal signal. Refer to the block diagram of three-phase motor control timers.

**Figure 17.18 Sawtooth Wave Modulation Operation**

## 17.6 Notes on Three-phase Motor Control Timers

### 17.6.1 Shutdown

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

### 17.6.2 Register setting

- Do not write to the TAi1 register (i = 1, 2, 4) in the timing that timer B2 overflows. Before writing to the TAi1 register, read the TB2 register to verify that sufficient time is left until timer B2 overflows. Then, immediately write to the TAi1 register so that no interrupt handler is performed during this write procedure. If the TB2 register indicates little time is left until the overflow, write to the TAi1 register after timer B2 overflows.

# 18. Serial Interface

Serial interface consists of nine channels (UART0 to UART8).
Each UARTi (i = 0 to 8) has an exclusive timer to generate the transmit/receive clock and operates independently.
Figure 18.1 and Figure 18.2 show respectively a block diagram of UART0 to UART6 and that of UART7 and UART8.
The UARTi supports following modes:

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

Figure 18.3 to Figure 18.19 show registers associated with the UARTi.
Refer to the tables listing each mode for registers and pin settings.

Note:
1. Please contact a Renesas sales office to use the optional feature.

**Table 18.1 Functions of UART0 to UART8**

| Mode/Function | UART0 to UART6 | UART7, UART8 |
|---|---|---|
| Synchronous serial interface mode | Available | Available |
| Serial data logical inversion | Available | Not available |
| UART mode | Available | Available |
| CTS/RTS selection | Available | Available |
| TXD and RXD I/O polarity selection | Available | Not available |
| Special mode 1 ($I^2C$ mode) | Available | Not available |
| Special mode 2 | Available | Not available |
| Special mode 4 (IE mode) (optional) (1) | Available | Not available |
| Pins TXD and RXD output mode | Push-pull output, N-channel open drain output programmable by port function select registers | Push-pull output, N-channel open drain output programmable by port function select registers |

Note:
1. Please contact a Renesas sales office to use the optional feature.

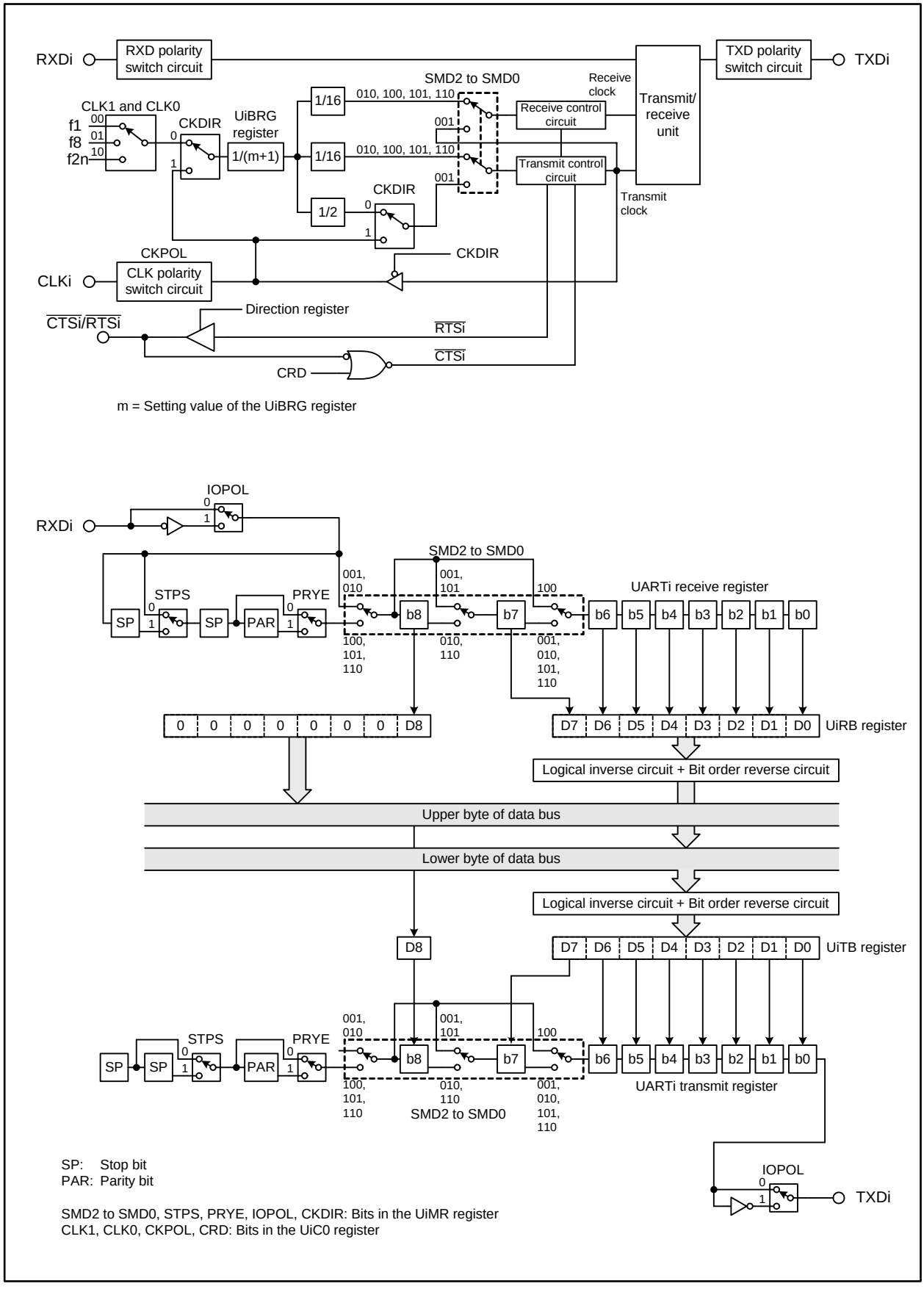
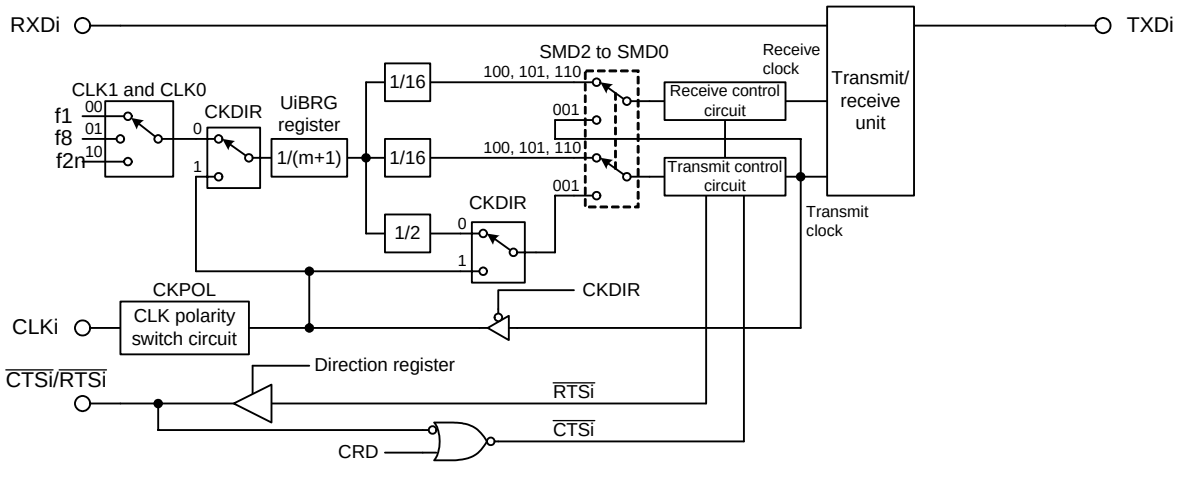
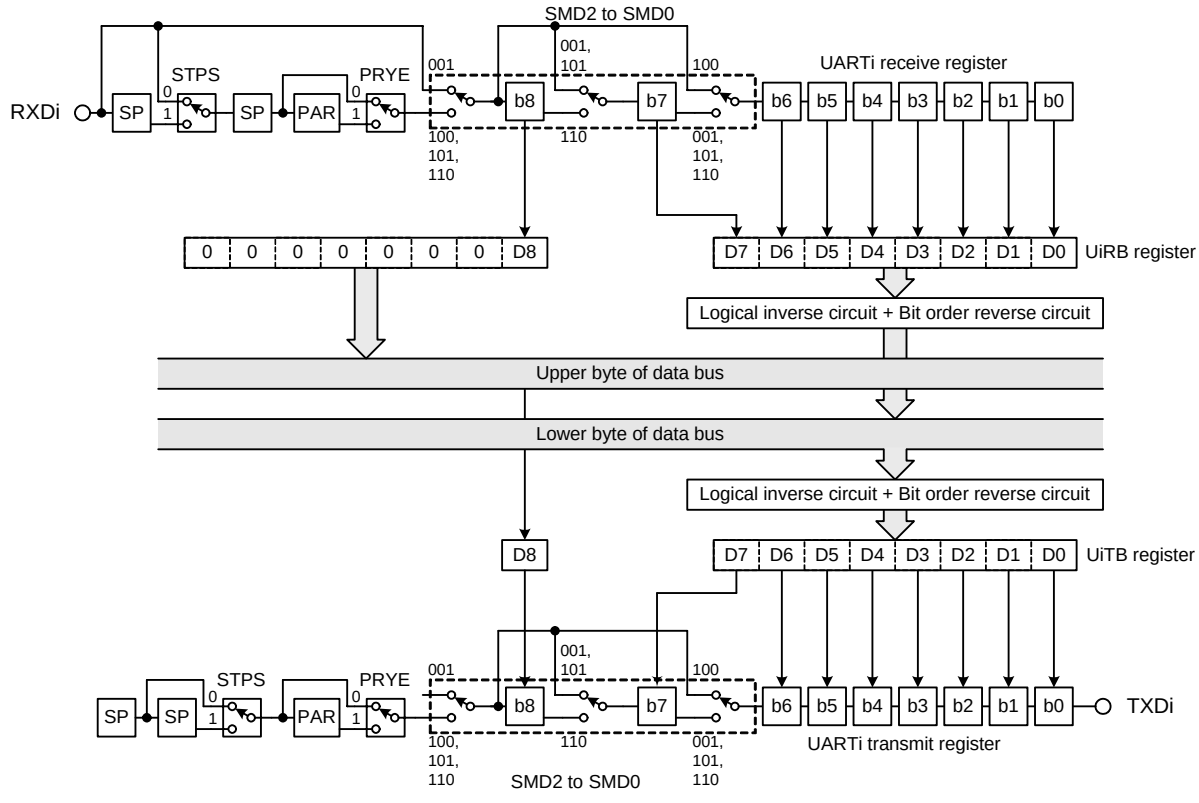



**Figure 18.1 UARTi Block Diagram (i = 0 to 6)**




m = Setting value of the UiBRG register

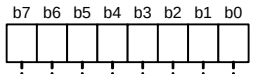



SP: Stop bit
PAR: Parity bit

SMD2 to SMD0, STPS, PRYE, IOPOL, CKDIR: Bits in the UiMR register
CLK1, CLK0, CKPOL, CRD: Bits in the UiC0 register

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

## UARTi Transmit/Receive Mode Register (i = 0 to 6)




| Symbol | Address | Reset Value |
|---|---|---|
| U0MR to U3MR | 0368h, 02E8h, 0338h, 0328h | 0000 0000b |
| U4MR to U6MR | 02F8h, 01C8h, 01D8h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| SMD0 | Serial Interface Mode Select Bit | b2 b1 b0<br>0 0 0 : Serial interface disabled<br>0 0 1 : Synchronous serial interface mode<br>0 1 0 : $I^2C$ mode<br>1 0 0 : UART mode, 7-bit character length<br>1 0 1 : UART mode, 8-bit character length<br>1 1 0 : UART mode, 9-bit character length<br>Only use the combinations listed above | RW |
| SMD1 | | | RW |
| SMD2 | | | RW |
| CKDIR | Internal/External Clock Select Bit | 0: Internal clock<br>1: External clock | RW |
| STPS | Stop Bit Length Select Bit | 0: 1 stop bit<br>1: 2 stop bits | RW |
| PRY | Odd/Even Parity Select Bit | Enabled when PRYE = 1<br>0: Odd parity<br>1: Even parity | RW |
| PRYE | Parity Enable Bit | 0: Parity disabled<br>1: Parity enabled | RW |
| IOPOL | TXD, RXD Input/Output Polarity Switch Bit | 0: Non inverted<br>1: Inverted | RW |

**Figure 18.3 Registers U0MR to U6MR**

## UARTi Transmit/Receive Mode Register (i = 7, 8)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | | | | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| U7MR, U8MR | 01E0h, 01E8h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| SMD0 | Serial Interface Mode Select Bit | b2 b1 b0<br>0 0 0 : Serial interface disabled<br>0 0 1 : Synchronous serial interface mode<br>0 1 0 : $I^2C$ mode<br>1 0 0 : UART mode, 7-bit character length<br>1 0 1 : UART mode, 8-bit character length<br>1 1 0 : UART mode, 9-bit character length<br>Only use the combinations listed above | RW |
| SMD1 | | | RW |
| SMD2 | | | RW |
| CKDIR | Internal/External Clock Select Bit | 0: Internal clock<br>1: External clock | RW |
| STPS | Stop Bit Length Select Bit | 0: 1 stop bit<br>1: 2 stop bits | RW |
| PRY | Odd/Even Parity Select Bit | Enabled when PRYE = 1<br>0: Odd parity<br>1: Even parity | RW |
| PRYE | Parity Enable Bit | 0: Parity disabled<br>1: Parity enabled | RW |
| —<br>(b7) | Reserved | Should be written with 0 | RW |

**Figure 18.4 Registers U7MR and U8MR**

## UARTi Transmit/Receive Control Register 0 (i = 0 to 6)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | 0 | | | 0 | | |

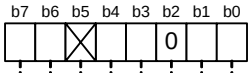
| Symbol | Address | Reset Value |
|---|---|---|
| U0C0 to U3C0 | 036Ch, 02ECh, 033Ch, 032Ch | 0000 1000b |
| U4C0 to U6C0 | 02FCh, 01CCh, 01DCh | 0000 1000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CLK0 | UiBRG Count Source Select Bit | b1 b0<br>0 0 : f1<br>0 1 : f8<br>1 0 : f2n<br>1 1 : Do not use this combination | RW |
| CLK1 | | | RW |
| — (b2) | Reserved | Should be written with 0 | RW |
| TXEPT | Transmit Shift Register Empty Flag | 0: Data held in the transmit shift register (transmission in progress)<br>1: No data held in the transmit shift register (transmission completed) | RO |
| CRD | CTS Disable Bit | 0: CTS enabled<br>1: CTS disabled | RW |
| — (b5) | Reserved | Should be written with 0 | RW |
| CKPOL | CLK Polarity Select Bit | 0: Output transmit data on the falling edge of the transmit/receive clock and input receive data on the rising edge<br>1: Output transmit data on the rising edge of the transmit/receive clock and input receive data on the falling edge | RW |
| UFORM | Bit Order Select Bit (1) | 0: LSB first<br>1: MSB first | RW |

Note:
1. This bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (synchronous serial interface mode selected) or 101b (UART mode, 8-bit character length selected). It should be set to 1 when bits SMD2 to SMD0 are set to 010b ($I^2C$ mode selected) and should be set to 0 when they are set to 100b (UART mode, 7-bit character length selected) or 110b (UART mode, 9-bit character length selected).

**Figure 18.5 Registers U0C0 to U6C0**

## UARTi Transmit/Receive Control Register 0 (i = 7, 8)



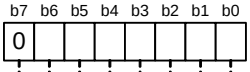

| Symbol | Address | Reset Value |
|---|---|---|
| U7C0, U8C0 | 01E4h, 01ECh | 00X0 1000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CLK0 | UiBRG Count Source Select Bit | b1 b0<br>0 0 : f1<br>0 1 : f8<br>1 0 : f2n<br>1 1 : Do not use this combination | RW |
| CLK1 | | | RW |
| — (b2) | Reserved | Should be written with 0 | RW |
| TXEPT | Transmit Shift Register Empty Flag | 0: Data held in the transmit shift register (transmission in progress)<br>1: No data held in the transmit shift register (transmission completed) | RO |
| CRD | CTS Disable Bit | 0: CTS enabled<br>1: CTS disabled | RW |
| — (b5) | No registers bit; should be written with 0 and read as undefined value | | — |
| CKPOL | CLK Polarity Select Bit | 0: Output transmit data on the falling edge of the transmit/receive clock and input receive data on the rising edge<br>1: Output transmit data on the rising edge of the transmit/receive clock and input receive data on the falling edge | RW |
| UFORM | Bit Order Select Bit (1) | 0: LSB first<br>1: MSB first | RW |

Note:
1. This bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (synchronous serial interface mode selected) or 101b (UART mode, 8-bit character length selected). It should be set to 0 when they are set to 100b (UART mode, 7-bit character length selected) or 110b (UART mode, 9-bit character length selected).

**Figure 18.6 Registers U7C0 and U8C0**

## UARTi Transmit/Receive Control Register 1 (i = 0 to 6)



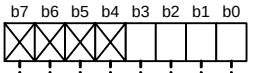

| Symbol | Address | Reset Value |
|---|---|---|
| U0C1 to U3C1 | 036Dh, 02EDh, 033Dh, 032Dh | 0000 0010b |
| U4C1 to U6C1 | 02FDh, 01CDh, 01DDh | 0000 0010b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TE | Transmit Enable Bit | 0: Transmission disabled<br>1: Transmission enabled | RW |
| TI | Transmit Buffer Empty Flag | 0: Data held in the UiTB register<br>1: No data held in the UiTB register | RO |
| RE | Receive Enable Bit | 0: Reception disabled<br>1: Reception enabled | RW |
| RI | Receive Complete Flag | 0: No data held in the UiRB register<br>1: Data held in the UiRB register | RO |
| UiIRS | UARTi Transmit Interrupt Source Select Bit | 0: UiTB register is empty (TI = 1)<br>1: Transmission is completed (TXEPT = 1) | RW |
| UiRRM | UARTi Continuous Receive Mode Enable Bit | 0: Continuous receive mode disabled<br>1: Continuous receive mode enabled | RW |
| UiLCH | Logical Inversion Select Bit (1) | 0: Data non logic-inverted<br>1: Data logic-inverted | RW |
| —<br>(b7) | Reserved | Should be written with 0 | RW |

Note:
1. This bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (synchronous serial interface mode selected), 100b (UART mode, 7-bit character length selected), or 101b (UART mode, 8-bit character length selected). It should be set to 0 when bits SMD2 to SMD0 are set to 010b ($I^2C$ mode selected) or 110b (UART mode, 9-bit character length selected).

**Figure 18.7 Registers U0C1 to U6C1**

## UARTi Transmit/Receive Control Register 1 (i = 7, 8)

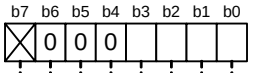



| Symbol | Address | Reset Value |
|---|---|---|
| U7C1, U8C1 | 01E5h, 01EDh | XXXX 0010b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TE | Transmit Enable Bit | 0: Transmission disabled<br>1: Transmission enabled | RW |
| TI | Transmit Buffer Empty Flag | 0: Data held in the UiTB register<br>1: No data held in the UiTB register | RO |
| RE | Receive Enable Bit | 0: Reception disabled<br>1: Reception enabled | RW |
| RI | Receive Complete Flag | 0: No data held in the UiRB register<br>1: Data held in the UiRB register | RO |
| —<br>(b7-b4) | No register bits; should be written with 0 and read as undefined value | | — |

**Figure 18.8 Registers U7C1 and U8C1**

## UART7, UART8 Transmit/Receive Control Register 2




| Symbol | Address | Reset Value |
|---|---|---|
| U78CON | 01F0h | X000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| U7IRS | UART7 Transmit Interrupt Source Select Bit | 0: Transmit buffer is empty (TI = 1)<br>1: Transmission is completed (TXEPT = 1) | RW |
| U8IRS | UART8 Transmit Interrupt Source Select Bit | 0: Transmit buffer is empty (TI = 1)<br>1: Transmission is completed (TXEPT = 1) | RW |
| U7RRM | UART7 Continuous Receive Mode Enable Bit | 0: Continuous receive mode disabled<br>1: Continuous receive mode enabled | RW |
| U8RRM | UART8 Continuous Receive Mode Enable Bit | 0: Continuous receive mode disabled<br>1: Continuous receive mode enabled | RW |
| —<br>(b6-b4) | Reserved | Should be written with 0 | RW |
| —<br>(b7) | No register bit; should be written with 0 and read as undefined value | | — |

**Figure 18.9 U78CON Register**

## UARTi Special Mode Register (i = 0 to 6)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | | | | 0 | | | |

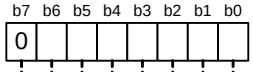
| Symbol | Address | Reset Value |
|---|---|---|
| U0SMR to U3SMR | 0367h, 02E7h, 0337h, 0327h | 0000 0000b |
| U4SMR to U6SMR | 02F7h, 01C7h, 01D7h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IICM | $I^2C$ Mode Select Bit (1) | 0: Non-$I^2C$ mode<br>1: $I^2C$ mode | RW |
| ABC | Arbitration Lost Detection Flag Control (1) | 0: Update every bit<br>1: Update every byte | RW |
| BBS | Bus Busy Flag (1, 2) | 0: Detect stop condition<br>1: Detect start condition (bus busy) | RW |
| — (b3) | Reserved | Should be written with 0 | RW |
| ABSCS | Bus Collision Detect Sampling Clock Select Bit (3) | 0: Rising edge of the transmit/receive clock<br>1: The underflow of timer Aj (j = 0, 3, 4) (4) | RW |
| ACSE | Transmit Enable Bit Auto-Reset to Zero Select Bit (3) | 0: Non auto-reset to zero<br>1: Auto-reset to zero at bus collision | RW |
| SSS | Transmit Start Condition Select Bit (3) | 0: No relation with RXDi<br>1: Synchronized with RXDi | RW |
| — (b7) | Reserved | Should be written with 0 | RW |

Notes:
1. This bit is used in $I^2C$ mode.
2. The BBS bit is only set to 0. This setting is unchanged even if it is set to 1.
3. This bit is used in IE mode.
4. UART0: timer A3 underflow signal, UART1: timer A4 underflow signal
   UART2: timer A0 underflow signal, UART3: timer A3 underflow signal
   UART4: timer A4 underflow signal, UART5: timer A3 underflow signal
   UART6: timer A4 underflow signal

**Figure 18.10 Registers U0SMR to U6SMR**

## UARTi Special Mode Register 2 (i = 0 to 6)



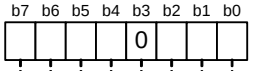

| Symbol | Address | Reset Value |
|---|---|---|
| U0SMR2 to U3SMR2 | 0366h, 02E6h, 0336h, 0326h | 0000 0000b |
| U4SMR2 to U6SMR2 | 02F6h, 01C6h, 01D6h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IICM2 | $I^2C$ Mode Select Bit 2 | 0: Use NACK/ACK interrupt<br>1: Use transmit/receive interrupt | RW |
| CSC | Clock Synchronous Bit (1) | 0: Clock synchronization disabled<br>1: Clock synchronization enabled | RW |
| SWC | SCL Wait Auto Insert Bit (2) | 0: No wait-state/wait-state cleared<br>1: Hold the SCLi pin low after the eighth bit of the SCLi is received | RW |
| ALS | SDA Output Auto Stop Bit (1) | If arbitration lost is detected,<br>0: Do not stop the SDAi output<br>1: Stop the SDAi output | RW |
| STC | UARTi Auto Initialize Bit (2) | If the start condition is detected,<br>0: Do not initialize the circuit<br>1: Initialize the circuit | RW |
| SWC2 | SCL Wait Output Bit 2 (1) | 0: Output the transmit/receive clock at the SCLi pin<br>1: Hold the SCLi pin low | RW |
| SDHI | SDA Output Stop Bit (2) | 0: Output data<br>1: Stop the output (high-impedance) | RW |
| —<br>(b7) | Reserved | Should be written with 0 | RW |

Notes:
1. This bit is used in master mode of $I^2C$ mode.
2. This bit is used in slave mode of $I^2C$ mode.

**Figure 18.11 Registers U0SMR2 to U6SMR2**

## UARTi Special Mode Register 3 (i = 0 to 6)



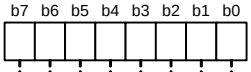

| Symbol | Address | Reset Value |
|---|---|---|
| U0SMR3 to U3SMR3 | 0365h, 02E5h, 0335h, 0325h | 0000 0000b |
| U4SMR3 to U6SMR3 | 02F5h, 01C5h, 01D5h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| SSE | $\overline{SS}$ Pin Function Enable Bit (1, 2) | 0: SS disabled<br>1: SS enabled | RW |
| CKPH | Clock-phase Set Bit | 0: Non clock delayed<br>1: Clock delayed | RW |
| DINC | Serial Input Pin Set Bit (1) | 0: Select the TXDi/RXDi pin (master mode)<br>1: Select the STXDi/SRXDi pin (slave mode) | RW |
| — (b3) | Reserved | Should be written with 0 | RW |
| ERR | Mode Fault Flag (1) | 0: No error detected<br>1: Error detected (3) | RW |
| DL0 | SDAi Digital Delay Time Set Bit (4, 5) | Based on the UiBRG count source, the output at the SDAi pin is delayed as below:<br>b7 b6 b5<br>0 0 0 : No delay<br>0 0 1 : 1 to 2 cycles<br>0 1 0 : 2 to 3 cycles<br>0 1 1 : 3 to 4 cycles<br>1 0 0 : 4 to 5 cycles<br>1 0 1 : 5 to 6 cycles<br>1 1 0 : 6 to 7 cycles<br>1 1 1 : 7 to 8 cycles | RW |
| DL1 | | | RW |
| DL2 | | | RW |

Notes:
1. This bit is used in special mode 2.
2. To use the SS function, the CRD bit in the UiC0 register should be set to 1 (CTS disabled).
3. The ERR bit is only set to 0. This setting is unchanged even if it is set to 1.
4. Bits DL2 to DL0 in $I^2C$ mode generate a digital delay for the output at the SDAi pin. These bits should be set to 000b (no delay) in all modes other than $I^2C$ mode.
5. When the external clock is selected, a delay of approximately 100 ns is added.

**Figure 18.12 Registers U0SMR3 to U6SMR3**

## UARTi Special Mode Register 4 (i = 0 to 6)



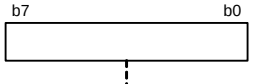

| Symbol | Address | Reset Value |
|---|---|---|
| U0SMR4 to U3SMR4 | 0364h, 02E4h, 0334h, 0324h | 0000 0000b |
| U4SMR4 to U6SMR4 | 02F4h, 01C4h, 01D4h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| STAREQ | Start Condition Generate Bit (1) | 0: Clear<br>1: Start (2) | RW |
| RSTAREQ | Restart Condition Generate Bit (1) | 0: Clear<br>1: Start (2) | RW |
| STPREQ | Stop Condition Generate Bit (1) | 0: Clear<br>1: Start (2) | RW |
| STSPSEL | SCL, SDA Output Select Bit (1) | 0: Select serial I/O circuit<br>1: Select start condition/stop condition generate circuit (3) | RW |
| ACKD | ACK Data Bit (4) | 0: ACK<br>1: NACK | RW |
| ACKC | ACK Data Output Enable Bit (4) | 0: Serial data output<br>1: ACK data output | RW |
| SCLHI | SCL Output Stop Bit (1) | If stop condition is detected,<br>0: Do not stop SCLi output<br>1: Stop SCLi output | RW |
| SWC9 | SCL Wait Auto Insert Bit 3 (4) | 0: No wait-state/wait-state cleared<br>1: Hold the SCLi pin low after the ninth bit of the SCLi is received | RW |

Notes:
1. This bit is used in master mode of $I^2C$ mode. To set this bit to 1, preset the IICM bit in the UiSMR register to 1 ($I^2C$ mode).
2. This bit becomes 0 when its condition is generated. The setting remains 1 when the condition is uncompleted.
3. The STSPSEL bit should be set to 1 after setting the STARREQ, RSTAREQ, or STPREQ bit to 1.
4. This bit is used in slave mode of $I^2C$ mode. To set this bit to 1, preset the IICM bit in the UiSMR register to 1 ($I^2C$ mode).

**Figure 18.13 Registers U0SMR4 to U6SMR4**

## UARTi Bit Rate Register (i = 0 to 8) (1, 2, 3)



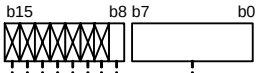

| Symbol | Address | Reset Value |
|---|---|---|
| U0BRG to U3BRG | 0369h, 02E9h, 0339h, 0329h | Undefined |
| U4BRG to U7BRG | 02F9h, 01C9h, 01D9h, 01E1h | Undefined |
| U8BRG | 01E9h | Undefined |

| Function | Setting Range | RW |
|---|---|---|
| The UiBRG register divides the count source by n+1 (n = setting value) | 00h to FFh | WO |

Notes:
1. Set bits CLK1 and CLK0 in the UiC0 register before rewriting this register.
2. The MOV instruction should be used to set this register.
3. The register should be set while no data is being transmitted/received.

**Figure 18.14 Registers U0BRG to U8BRG**

## UARTi Transmit Buffer Register (i = 0 to 8) (1)



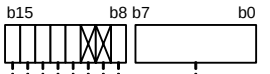

| Symbol | Address | Reset Value |
|---|---|---|
| U0TB to U2TB | 036Bh-036Ah, 02EBh-02EAh, 033Bh-033Ah | Undefined |
| U3TB to U5TB | 032Bh-032Ah, 02FBh-02FAh, 01CBh-01CAh | Undefined |
| U6TB to U8TB | 01DBh-01DAh, 01E3h-01E2h, 01EBh-01EAh | Undefined |

| Bit Symbol | Function | RW |
|---|---|---|
| — (b7-b0) | Data (D7 to D0) transmitted | WO |
| — (b8) | Data (D8) transmitted | WO |
| — (b15-b9) | No register bits; should be written with 0 | — |

Note:
1.The MOV instruction should be used to set the UARTi register.

**Figure 18.15 Registers U0TB to U8TB**

## UARTi Receive Buffer Register (i = 0 to 6)



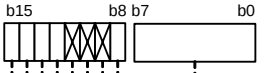

| Symbol | Address | Reset Value |
|---|---|---|
| U0RB to U2RB | 036Fh-036Eh, 02EFh-02EEh, 033Fh-033Eh | Undefined |
| U3RB to U5RB | 032Fh-032Eh, 02FFh-02FEh, 01CFh-01CEh | Undefined |
| U6RB | 01DFh-01DEh | Undefined |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b7-b0) | — | Data (D7 to D0) received | RO |
| — (b8) | — | Data (D8) received | RO |
| — (b10-b9) | No register bits; should be written with 0 and read as 0 | | — |
| ABT | Arbitration Lost Detection Flag (1) | 0: Not detected (win)<br>1: Detected (lose) | RW |
| OER | Overrun Error Flag (2) | 0: No overrun error occurred<br>1: Overrun error occurred | RO |
| FER | Framing Error Flag (2, 3) | 0: No framing error occured<br>1: Framing error occurred | RO |
| PER | Parity Error Flag (2, 3) | 0: No parity error occurred<br>1: Parity error occurred | RO |
| SUM | Error Sum Flag (2, 3) | 0: No error occurred<br>1: Error occurred | RO |

Notes:
1. The ABT bit can be set to 0 only.
2. Bits OER, FER, PER, and SUM are set to 0 when bits SMD2 to SMD0 in the UiMR register are set to 000b (serial interface disabled) or the RE bit in the UiC1 register is set to 0 (reception disabled). When bits OER, FER, and PER are all set to 0, the SUM bit is also set to 0. Bits FER and PER are set to 0 when the lower byte in the UiRB register is read.
3. When bits SMD2 to SMD0 are set to 001b (synchronous serial interface mode selected) or 010b ($I^2C$ mode selected), these error flags are disabled and read as an undefined value.

**Figure 18.16 Registers U0RB to U6RB**

## UARTi Receive Buffer Register (i = 7, 8)




| Symbol | Address | Reset Value |
|---|---|---|
| U7RB, U8RB | 01E7h-01E6h, 01EFh-01EEh | Undefined |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b7-b0) | — | Data (D7 to D0) received | RO |
| — (b8) | — | Data (D8) received | RO |
| — (b11-b9) | No register bits; should be written with 0 and read as 0 | | — |
| OER | Overrun Error Flag (1) | 0: No overrun error occurred<br>1: Overrun error occurred | RO |
| FER | Framing Error Flag (1, 2) | 0: No framing error occurred<br>1: Framing error occurred | RO |
| PER | Parity Error Flag (1, 2) | 0: No parity error occurred<br>1: Parity error occurred | RO |
| SUM | Error Sum Flag (1, 2) | 0: No error occurred<br>1: Error occurred | RO |

Notes:
1. Bits OER, FER, PER, and SUM are set to 0 when bits SMD2 to SMD0 in the UiMR register are set to 000b (serial interface disabled) or the RE bit in the UiC1 register is set to 0 (reception disabled). When bits OER, FER, and PER are all set to 0, the SUM bit is also set to 0. Bits FER and PER are set to 0 when the lower byte in the UiRB register is read.
2. When bits SMD2 to SMD0 are set to 001b (synchronous serial interface mode selected) or 010b ($I^2C$ mode selected), these error flags are disabled and read as an undefined value.

**Figure 18.17 Registers U7RB and U8RB**

## External Interrupt Request Source Select Register 0

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| IFSR0 | 4406Fh | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IFSR00 | $\overline{\text{INT0}}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR01 | $\overline{\text{INT1}}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR02 | $\overline{\text{INT2}}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR03 | $\overline{\text{INT3}}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR04 | $\overline{\text{INT4}}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR05 | $\overline{\text{INT5}}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR06 | UART0/UART3 Interrupt Source Select Bit | 0: Bus collision, start condition detection, stop condition detection in UART3<br>1: Bus collision, start condition detection, stop condition detection in UART0 | RW |
| IFSR07 | UART1/UART4 Interrupt Source Select Bit | 0: Bus collision, start condition detection, stop condition detection in UART4<br>1: Bus collision, start condition detection, stop condition detection in UART1 | RW |

Note:
1. This bit should be set to 0 to select the level sensitive input as trigger. To set this bit to 1, the POL bit in the corresponding INTiIC register (i = 0 to 5) should be set to 0 (falling edge).

**Figure 18.18 IFSR0 Register**

## External Interrupt Request Source Select Register 1

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| IFSR1 | 4406Dh | X0XX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IFSR10 | $\overline{\text{INT6}}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR11 | $\overline{\text{INT7}}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| IFSR12 | $\overline{\text{INT8}}$ Pin Polarity Select Bit (1) | 0: One edge<br>1: Both edges | RW |
| —<br>(b5-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| IFSR16 | UART5/UART6 Interrupt Source Select Bit | 0: Bus collision, start condition detection, stop condition detection in UART5<br>1: Bus collision, start condition detection, stop condition detection in UART6 | RW |
| —<br>(b7) | No register bit; should be written with 0 and read as undefined value | | — |

Note:
1. This bit should be set to 0 to select the level sensitive input as trigger. To set this bit to 1, the POL bit in the corresponding INTiIC register (i = 6 to 8) should be set to 0 (falling edge).

**Figure 18.19 IFSR1 Register**

## 18.1 Synchronous Serial Interface Mode

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

**Table 18.2 Synchronous Serial Interface Mode Specifications**

| Item | Specification |
|---|---|
| Data format | 8-bit character length |
| Transmit/receive clock | • The CKDIR bit in the UiMR register (i = 0 to 8) is set to 0 (internal clock): $\frac{fx}{2(m+1)}$ fx = f1, f8, f2n; m: UiBRG register setting value, 00h to FFh<br>• The CKDIR bit is set to 1 (external clock): input into the CLKi pin |
| Transmit/receive control | Selectable among CTS, RTS, and CTS/RTS disabled |
| Transmit start conditions | The conditions for starting data transmission are as follows (1):<br>• The TE bit in the UiC1 register is set to 1 (transmission enabled)<br>• The TI bit in the UiC1 register is set to 0 (data held in the UiTB register)<br>• Input level at the $\overline{\text{CTSi}}$ pin is L when the CTS is selected |
| Receive start conditions | The conditions for starting data reception are as follows (1):<br>• The RE bit in the UiC1 register is set to 1 (reception enabled)<br>• The TE bit in the UiC1 register is set to 1 (transmission enabled)<br>• The TI bit in the UiC1 register is set to 0 (data held in the UiTB register)<br>• Input level at the $\overline{\text{CTSi}}$ pin is low when the CTS is selected |
| Interrupt request generating timing | In transmit interrupt, one of the following conditions is selected to set the UiIRS bit in registers U0C1 to U6C1 and U78CON:<br>• The UiIRS bit is set to 0 (transmit buffer in the UiTB register is empty): when data is transferred from the UiTB register to the UARTi transmit register (when the transmission has started)<br>• The UiIRS bit is set to 1 (transmission is completed): when data transmission from the UARTi transmit register is completed<br>In receive interrupt,<br>• When data is transferred from the UARTi receive register to the UiRB register (when the reception is completed) |
| Error detection | Overrun error (2)<br>This error occurs when the seventh bit of the next data has been received before the UiRB register is read |
| Selectable functions | • CLK polarity<br>Selectable either rising or falling edge of the transmit/receive clock for output and input of transmit/receive data<br>• Bit order selection<br>Selectable either LSB first or MSB first<br>• Continuous receive mode<br>Data reception is enabled by a read access to the UiRB register<br>• Serial data logical inversion (UART0 to UART6)<br>This function logically inverses transmit/receive data |

Notes:

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

Table 18.3 and Table 18.4 list register settings. When UARTi (i = 0 to 8) operating mode is selected, a high is output at the TXDi pin until the transmission starts (the TXDi pin is high-impedance when the N-channel open drain output is selected).
Figure 18.20 and Figure 18.21 show respectively an example of transmit/receive operation in synchronous serial interface mode.

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

| Register | Bits | Function |
|---|---|---|
| UiMR | 7 to 4 | Set the bits to 0000b |
| | CKDIR | Select either the internal clock or the external clock |
| | SMD2 to SMD0 | Set the bits to 001b |
| UiC0 | UFORM | Select either LSB first or MSB first |
| | CKPOL | Select a transmit/receive clock polarity |
| | 5 | Set the bit to 0 |
| | CRD | Select the CTS enabled or disabled |
| | TXEPT | Transmit register empty flag |
| | 2 | Set the bit to 0 |
| | CLK1 and CLK0 | Select a count source for the UiBRG register |
| UiC1 | 7 | Set the bit to 0 |
| | UiLCH | Set the bit to 1 to use logical inversion |
| | UiRRM | Set the bit to 1 to use continuous receive mode |
| | UiIRS | Select a source for UARTi transmit interrupt |
| | RI | Receive complete flag |
| | RE | Set the bit to 1 to enable data reception |
| | TI | Transmit buffer empty flag |
| | TE | Set the bit to 1 to enable data transmission/reception |
| UiSMR | 7 to 0 | Set the bits to 00h |
| UiSMR2 | 7 to 0 | Set the bits to 00h |
| UiSMR3 | 7 to 0 | Set the bits to 00h |
| UiSMR4 | 7 to 0 | Set the bits to 00h |
| UiBRG | 7 to 0 | Set the bit rate |
| IFS0 | IFS06 | Select input pins for CLK3, RXD3, and $\overline{\text{CTS3}}$ |
| | IFS03 and IFS02 | Select input pins for CLK6, RXD6, and $\overline{\text{CTS6}}$ |
| UiTB | 7 to 0 | Set the data to be transmitted |
| UiRB | OER | Overrun error flag |
| | 7 to 0 | Received data is read |

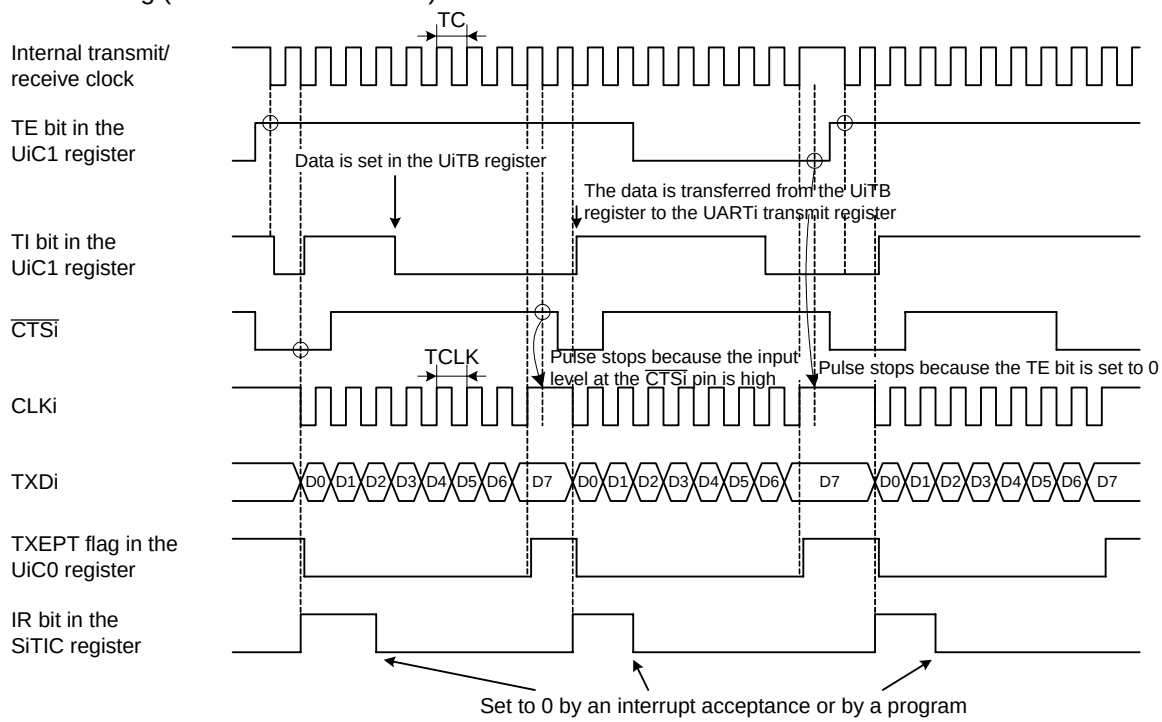
i = 0 to 6

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

| Register | Bits | Function |
|---|---|---|
| UiMR | 7 to 4 | Set the bits to 0000b |
| | CKDIR | Select the internal clock or the external clock |
| | SMD2 to SMD0 | Set the bits to 001b |
| UiC0 | UFORM | Select either LSB first or MSB first |
| | CKPOL | Select a transmit/receive clock polarity |
| | 5 | Set the bit to 0 |
| | CRD | Select the CTS enabled or disabled |
| | TXEPT | Transmit register empty flag |
| | 2 | Set the bit to 0 |
| | CLK1 and CLK0 | Select a count source for the UiBRG register |
| UiC1 | RI | Receive complete flag |
| | RE | Set the bit to 1 to enable data reception |
| | TI | Transmit buffer empty flag |
| | TE | Set the bit to 1 to enable data transmission/reception |
| U78CON | UiRRM | Set the bit to 1 to use continuous receive mode |
| | UiIRS | Select an interrupt source for UARTi transmit |
| IFS0 | IFS05 | Select input pins for CLK7, RXD7, and $\overline{\text{CTS7}}$ |
| | IFS04 | Select input pins for CLK8, RXD8, and $\overline{\text{CTS8}}$ |
| UiBRG | 7 to 0 | Set the bit rate |
| UiTB | 7 to 0 | Set the data to be transmitted |
| UiRB | OER | Overrun error flag |
| | 7 to 0 | Received data can be read |

i = 7, 8

Transmit timing (in internal clock mode)

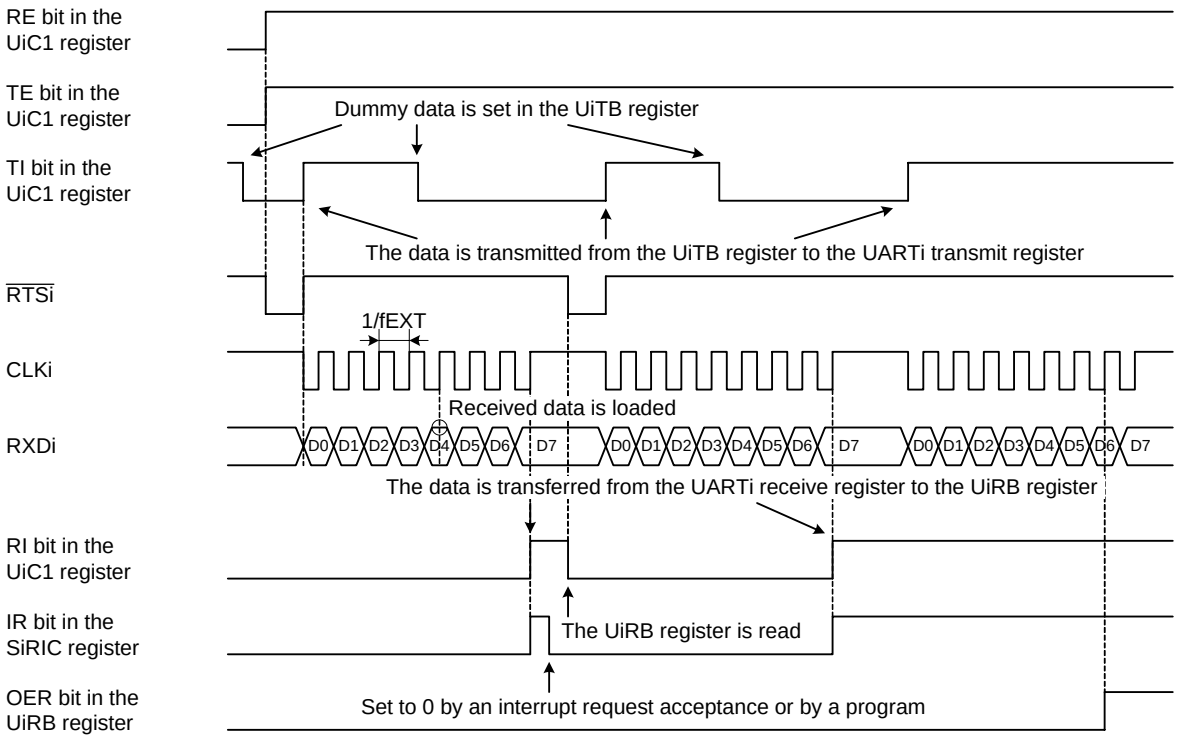



The figure above applies under the following conditions:
- -The CKDIR bit in the UiMR register = 0 (internal clock selected)
- -The CRD bit in the UiC0 register = 0 (CTS enabled)
- -The CKPOL bit in the UiC0 register = 0 (transmitted data output on the falling edge of the transmit/receive clock)
- -The UiRS bit in registers UiC1 and U78CON = 0 (an transmit interrupt request generated because the UiTB register is empty)

TC = TCLK = 2(m+1)/fx
fx: UiBRG count source frequency (f1, f8, or f2n)
m: Setting value of the UiBRG register

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

Receive timing (in external clock mode)




The figure above applies under the following conditions:
- -The CKDIR bit in the UiMR register = 1 (external clock selected)
- -The CKPOL bit in the UiC0 register = 0 (received data input on the rising edge of the transmit/receive clock)

fEXT: External clock frequency

The following conditions should be met while an input level at the CLKi pin before receiving data is high:
- -The TE bit in the UiC1 register = 1 (transmission enabled)
- -The RE bit in the UiC1 register = 1 (reception enabled)
- -A write of dummy data to the UiTB register

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

### 18.1.1 Reset Procedure on Transmit/Receive Error

When a transmit/receive error occurs in synchronous serial interface mode, a reset is required as the procedure below:

A. Reset procedure for the UiRB register (i = 0 to 8)

(1) Set the RE bit in the UiC1 register to 0 (reception disabled).
(2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(3) Set bits SMD2 to SMD0 in the UiMR register to 001b (synchronous serial interface mode selected).
(4) Set the RE bit in the UiC1 register to 1 (reception enabled).

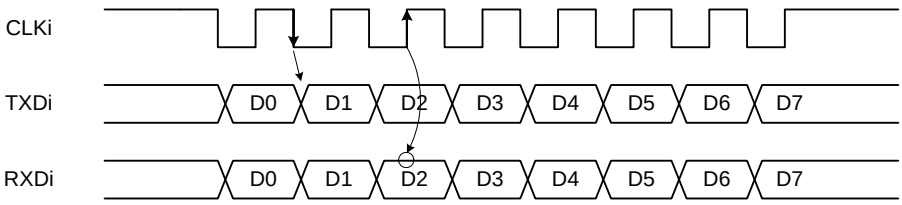
B. Reset procedure for the UiTB register (i = 0 to 8)

(1) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(2) Set bits SMD2 to SMD0 in the UiMR register to 001b (synchronous serial interface mode selected).
(3) Set the TE bit in the UiC1 register to 1 (transmission enabled) irrespective of the bit setting.

### 18.1.2 CLK Polarity

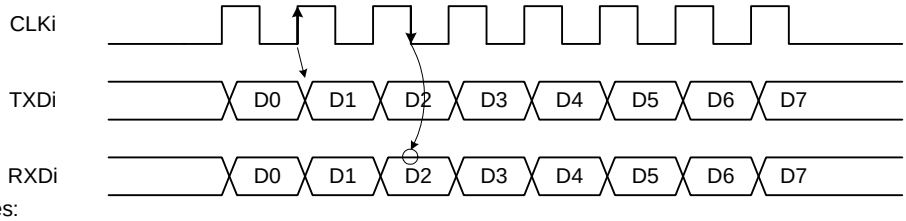
As shown in Figure 18.22, the polarity of the transmit/receive clock is selected using the CKPOL bit in the UiC0 register (i = 0 to 8).

Case 1: When the CKPOL bit in the UiC0 register is set to 0 (transmitted data output on the falling edge of the transmit/receive clock and received data input on the rising edge)




Notes:
1. The CLKi pin is held high when no data is transmitted/received.
2. The figure above applies under the following conditions:
-The UFORM bit in the UiC0 register = 0 (LSB first)
-The UiLCH bit in the UiC1 register = 0 (data is non logic-inverted)

Case 2: When the CKPOL bit in the UiC0 register is set to 1 (transmitted data output on the rising edge of the transmit/receive clock and received data input on the falling edge)



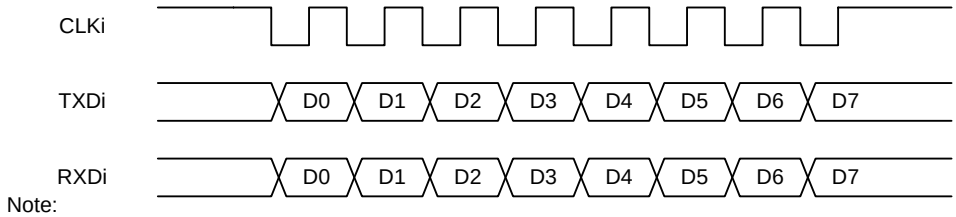

Notes:
3. The CLKi pin is held low when no data is transmitted/received.
4. The figure above applies under the following conditions:
-The UFORM bit in the UiC0 register = 0 (LSB first)
-The UiLCH bit in the UiC1 register = 0 (data is non logic-inverted)

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

### 18.1.3 LSB First and MSB First Selection

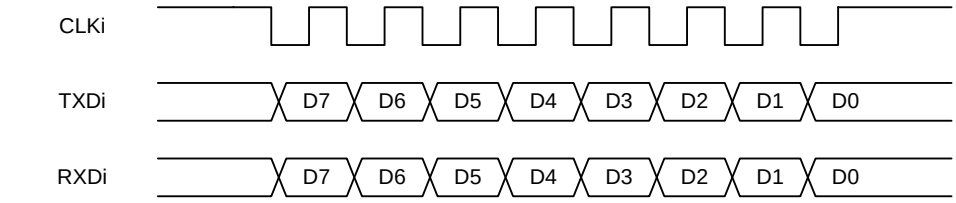
As shown in Figure 18.23, the bit order is selected using the UFORM bit in the UiC0 register (i = 0 to 8).

Case 1: When the UFORM bit in the UiC0 register is set to 0 (LSB first)




Note:
1. The figure above applies under the following conditions:
   -The CKPOL bit in the UiC0 register = 0 (transmitted data output on the falling edge of the transmit/ receive clock and received data input on the rising edge)
   -The UiLCH bit in the UiC1 register = 0 (data is non logic-inverted)

Case 2: When the UFORM bit in the UiC0 register is set to 1 (MSB first)




Note:
2. The figure above applies under the following conditions:
   -The CKPOL bit in the UiC0 register = 0 (transmitted data output on the falling edge of the transmit/ receive clock and received data input on the rising edge)
   -The UiLCH bit in the UiC1 register = 0 (data is non logic-inverted)

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

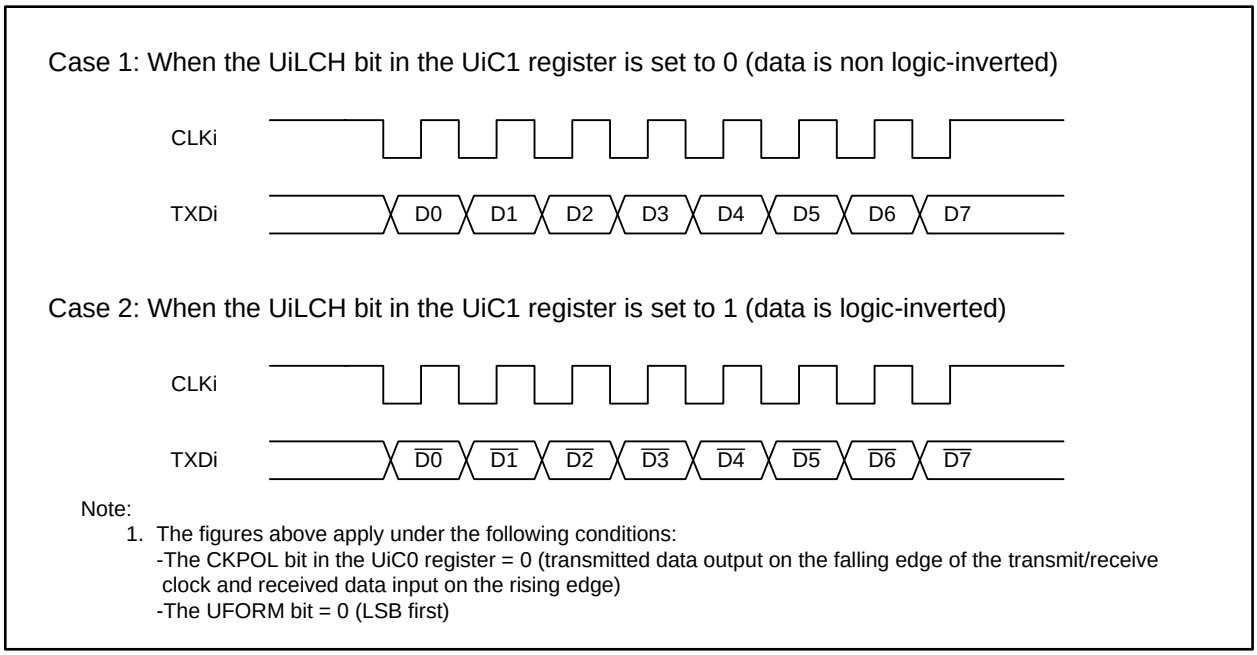
### 18.1.4 Continuous Receive Mode

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

When the UiRRM bit (i = 0 to 8) in registers U0C1 to U6C1 and U78CON is set to 1 (continuous receive mode enabled), the TI bit in the UiC1 register is set to 0 (data held in the UiTB register) by a read access to the UiRB register. In this UiRRM bit setting, any dummy data should not be written to the UiTB register.

### 18.1.5 Serial Data Logical Inversion

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




Note:
1. The figures above apply under the following conditions:
-The CKPOL bit in the UiC0 register = 0 (transmitted data output on the falling edge of the transmit/receive clock and received data input on the rising edge)
-The UFORM bit = 0 (LSB first)

**Figure 18.24 Serial Data Logical Inversion (i = 0 to 6)**

### 18.1.6 CTS/RTS Function

The CTS controls data transmission using the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ pin (i = 0 to 8). When an input level at the pin becomes low, data transmission is started. If the input level changes to high during transmit operation, the transmission of the next data is stopped.

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

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

## 18.2 Asynchronous Serial Interface Mode (UART Mode)

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

**Table 18.5 UART Mode Specifications**

| Item | Specification |
|---|---|
| Data format | • Start bit: 1 bit-length<br>• Data bit (data character) selectable among 7, 8, and 9 bit-length<br>• Parity bit: selectable among odd, even, and none<br>• Stop bit: selectable between 1 and 2 bit-length |
| Transmit/receive clock | • The CKDIR bit in the UiMR register (i = 0 to 8) is set to 0 (internal clock):<br>$\frac{fx}{16(m+1)}$ fx = f1, f8, f2n; m: UiBRG register setting value, 00h to FFh<br>• The CKDIR bit is set to 1 (external clock)<br>$\frac{fEXT}{16(m+1)}$ fEXT: Clock applied to the CLKi pin |
| Transmit/receive control | Selectable among CTS, RTS, and CTS/RTS disabled |
| Transmit start conditions | The conditions for starting data transmission are as follows:<br>• The TE bit in the UiC1 register is set to 1 (transmission enabled)<br>• The TI bit in the UiC1 register is set to 0 (data held in the UiTB register)<br>• Input level at the $\overline{\text{CTSi}}$ pin is low when the CTS is selected |
| Receive start conditions | The conditions for starting data reception are as follows:<br>• The RE bit in the UiC1 register is set to 1 (reception enabled)<br>• The start bit is detected |
| Interrupt request generating timing | In transmit interrupt, one of the following conditions is selected to set the UiIRS bit in registers U0C1 to U6C1 and U78CON:<br>• The UiIRS bit is set to 0 (transmit buffer in the UiTB register is empty): when data is transmitted from the UiTB register to the UARTi transmit register (when the transmission has started)<br>• The UiIRS bit is set to 1 (transmission is completed): when data transmission from the UARTi transmit register is completed<br>In receive interrupt,<br>• When data is transmitted from the UARTi receive register to the UiRB register (when reception is completed) |
| Error detection | • Overrun error (1)<br>This error occurs when one bit prior to the stop bit (when 1 stop bit length is selected) or the first stop bit (when 2 stop bit length is selected) of the next data has been received before the UiRB register is read<br>• Framing error<br>This error occurs when the required number of stop bits is not detected<br>• Parity error<br>This error occurs when an even number of 1's in parity and character bits is detected while the odd number is set, or vice versa. The parity should be enabled<br>• Error sum flag<br>This flag is set to 1 when any of overrun error, framing error, or parity error occurs |
| Selectable functions | • Bit order selection<br>Selectable either LSB first or MSB first<br>• Serial data logical inversion<br>This function logically inverses transmit/receive data. The start bit and stop bit are not inverted<br>• TXD/RXD I/O polarity switching<br>The output level from the TXD pin and the input level to the RXD pin are inverted. All I/O levels are inverted |

Note:

1. When an overrun error occurs, the UiRB register is undefined. The IR bit in the SiRIC register is not changed to 1 (interrupts requested).

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

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

| Register | Bits | Function |
|---|---|---|
| UiMR | IOPOL | Select I/O polarity of pins TXD and RXD |
| | PRY and PRYE | Select parity enabled or disabled, and odd or even |
| | STPS | Select a stop bit length |
| | CKDIR | Select the internal clock or the external clock |
| | SMD2 to SMD0 | Set the bits to 100b in 7-bit character length |
| | | Set the bits to 101b in 8-bit character length |
| | | Set the bits to 110b in 9-bit character length |
| UiC0 | UFORM | Selectable either LSB first or MSB first in 8-bit character length.<br>Set the bit to 0 in 7-bit or 9-bit character length |
| | CKPOL | Set the bit to 0 |
| | 5 | Set the bit to 0 |
| | CRD | Select the CTS enabled or disabled |
| | TXEPT | Transmit register empty flag |
| | 2 | Set the bit to 0 |
| | CLK1 and CLK0 | Select a count source for the UiBRG register |
| UiC1 | 7 | Set the bit to 0 |
| | UiLCH | Set the bit to 1 to use logical inversion |
| | UiRRM | Set the bit to 0 |
| | UiIRS | Select an interrupt source for UARTi transmit |
| | RI | Receive complete flag |
| | RE | Set the bit to 1 to enable data reception |
| | TI | Transmit buffer empty flag |
| | TE | Set the bit to 1 to enable data transmission |
| UiSMR | 7 to 0 | Set the bits to 00h |
| UiSMR2 | 7 to 0 | Set the bits to 00h |
| UiSMR3 | 7 to 0 | Set the bits to 00h |
| UiSMR4 | 7 to 0 | Set the bits to 00h |
| UiBRG | 7 to 0 | Set the bit rate |
| IFS0 | IFS06 | Select input pins for CLK3, RXD3, and $\overline{\text{CTS3}}$ |
| | IFS03 and IFS02 | Select input pins for CLK6, RXD6, and $\overline{\text{CTS6}}$ |
| UiTB | 8 to 0 | Set the data to be transmitted (1) |
| UiRB | OER, FER, PER, and SUM | Error flag |
| | 8 to 0 | Received data is read (1) |

i = 0 to 6

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

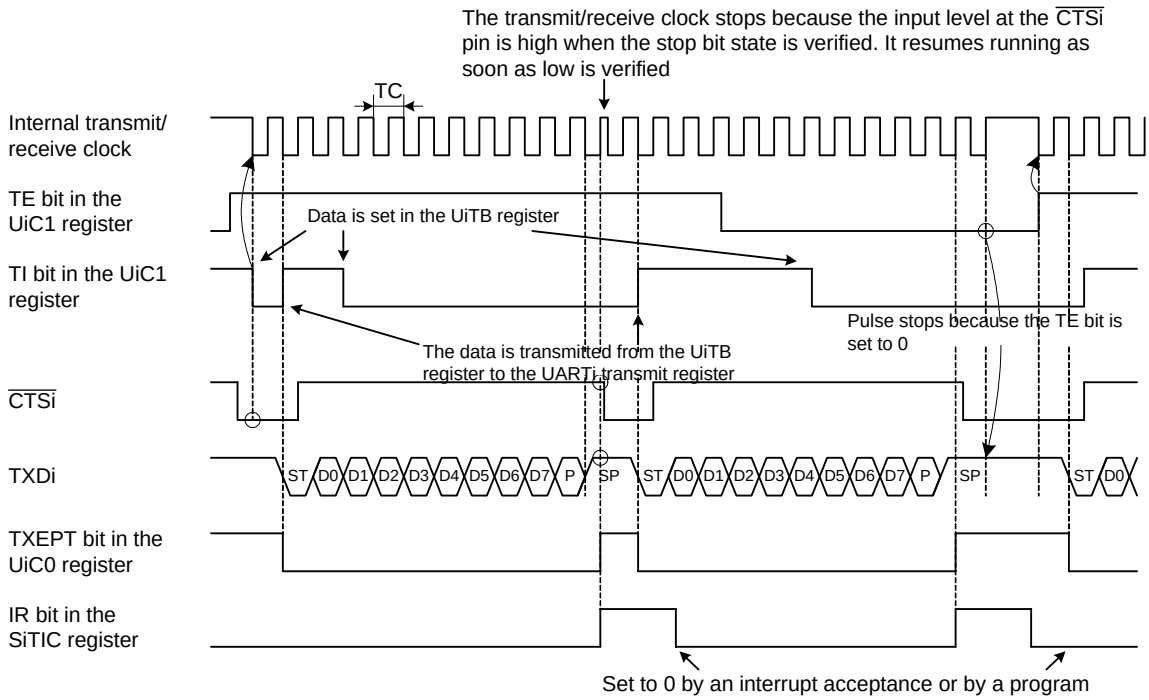
**Table 18.7 Register Settings in UART Mode (UART7, UART8)**

| Register | Bits | Function |
|---|---|---|
| UiMR | PRY and PRYE | Select parity enabled or disabled, and odd or even |
| | STPS | Select a stop bit length |
| | CKDIR | Select the internal clock or the external clock. |
| | SMD2 to SMD0 | Set the bits to 100b in 7-bit character length |
| | | Set the bits to 101b in 8-bit character length |
| | | Set the bits to 110b in 9-bit character length |
| UiC0 | UFORM | Selectable either LSB first or MSB first in 8-bit character length.<br>Set the bit to 0 in 7-bit or 9-bit character length |
| | CKPOL | Set the bit to 0 |
| | 5 | Set the bit to 0 |
| | CRD | Select the CTS enabled or disabled |
| | TXEPT | Transmit register empty flag |
| | 2 | Set the bit to 0 |
| | CLK1 and CLK0 | Select a count source for the UiBRG register |
| UiC1 | RI | Receive complete flag |
| | RE | Set the bit to 1 to enable data reception |
| | TI | Transmit buffer empty flag |
| | TE | Set the bit to 1 to enable data transmission |
| U78CON | UiRRM | Set the bit to 0 |
| | UiIRS | Select an interrupt source for UARTi transmit |
| UiBRG | 7 to 0 | Set the bit rate |
| IFS0 | IFS05 | Select input pins for CLK7, RXD7, and $\overline{\text{CTS7}}$ |
| | IFS04 | Select input pins for CLK8, RXD8, and $\overline{\text{CTS8}}$ |
| UiTB | 8 to 0 | Set the data to be transmitted (1) |
| UiRB | OER, FER, PER, and SUM0 | Error flag |
| | 8 to 0 | Received data is read (1) |

i = 7, 8

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

8-bit character long data transmit timing (parity enabled, 1 stop bit)



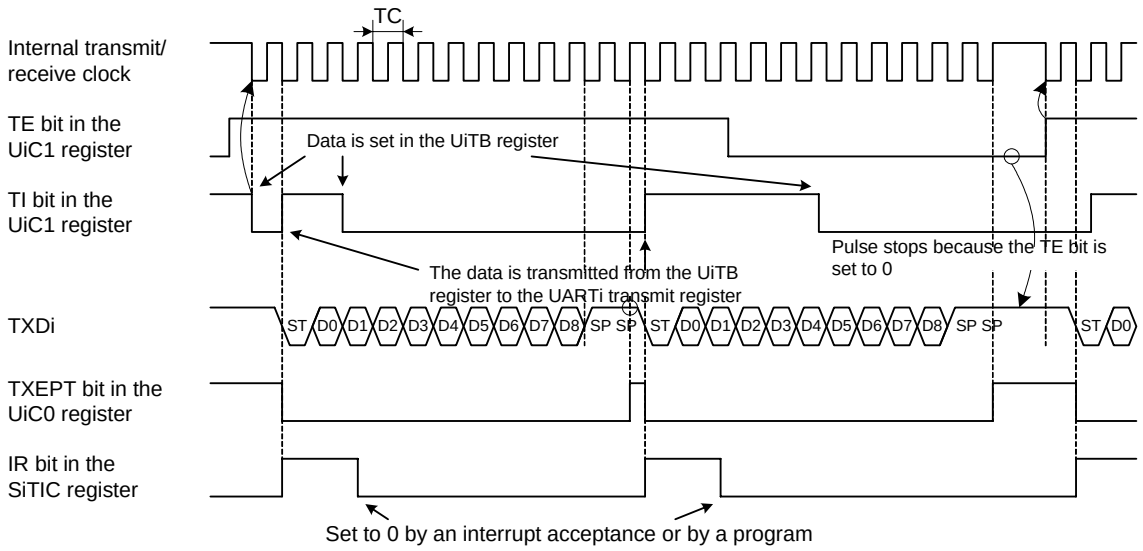

ST: Start bit
P: Parity bit
SP: Stop bit

The figure above applies under the following conditions:
- -The PRYE bit in the UiMR register = 1 (parity enabled)
- -The STPS bit in the UiMR register = 0 (1 stop bit)
- -The CRD bit in the UiC0 register = 0 (CTS enabled)
- -The UiIRS bit in the UiC1 register = 1 (transmit interrupt request generated because the UiTB register is empty)

TC = 16(m+1)/fx or 16(m+1)/fEXT
fx: UiBRG count source frequency (f1, f8, f2n)
fEXT: UiBRG count source frequency (external clock)
m: Setting value of the UiBRG register

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

9-bit character long data transmit timing (parity disabled, 2 stop bits)




ST: Start bit
P: Parity bit
SP: Stop bit

The figure above applies under the following conditions:
- -The PRYE bit in the UiMR register = 0 (parity disabled)
- -The STPS bit in the UiMR register = 1 (2 stop bits)
- -The CRD bit in the UiC0 register = 1 (CTS disabled)
- -The UiIRS bit in the UiC1 register = 0 (transmit interrupt request generated because the transmission is completed)

TC = 16(m+1)/fx or 16(m+1)/fEXT
fx: UiBRG count source frequency (f1, f8, f2n)
fEXT: UiBRG count source frequency (external clock)
m: Setting value of the UiBRG register

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

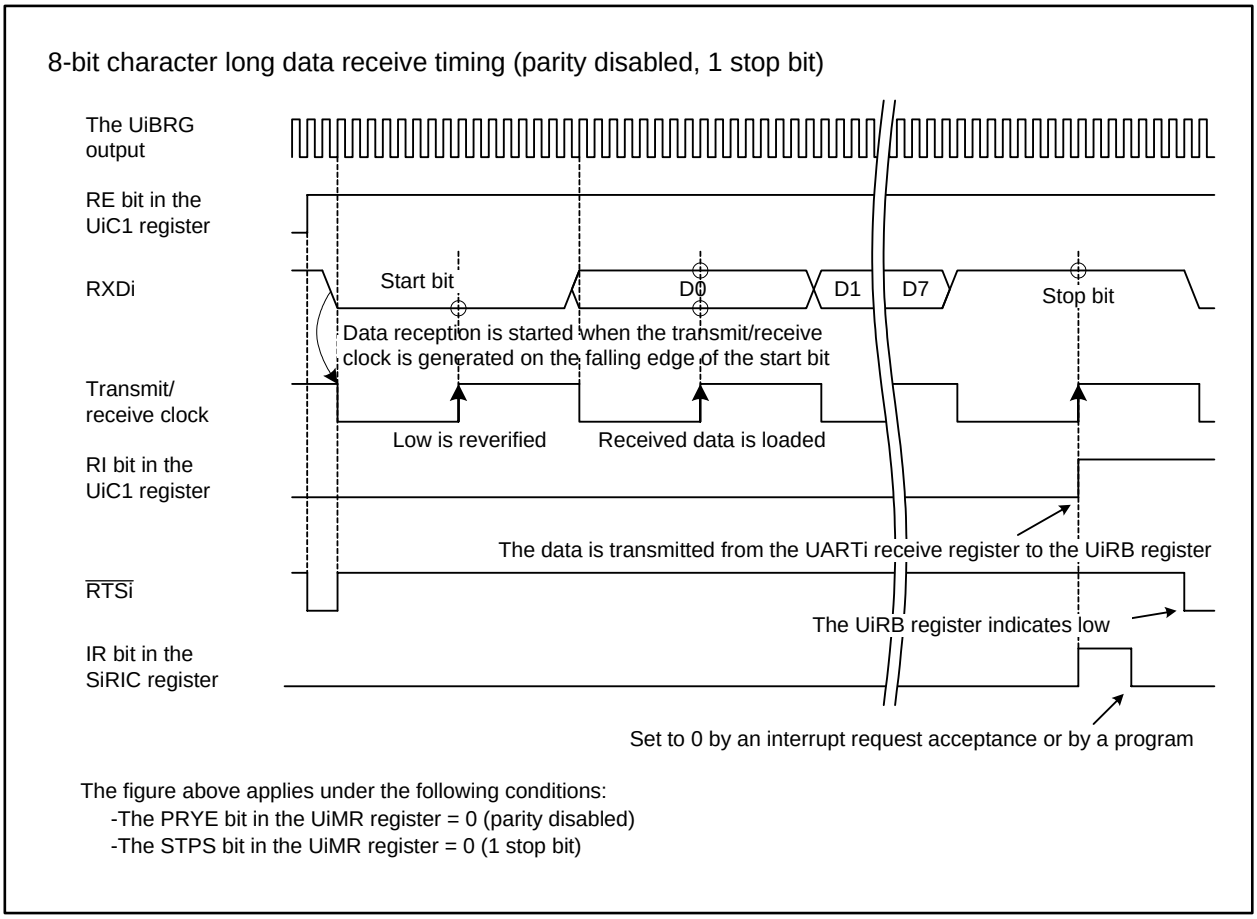



**Figure 18.27 Receive Operation in UART mode (i = 0 to 8)**

## 18.2.1 Bit Rate

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

**Table 18.8 Bit Rate Setting**

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

### 18.2.2 Reset Procedure on Transmit/Receive Error

When a transmit/receive error occurs in UART mode, a reset is required as the procedure below:

A. Reset procedure for the UiRB register (i = 0 to 8)

(1) Set the RE bit in the UiC1 register to 0 (reception disabled).
(2) Set the RE bit in the UiC1 register to 1 (reception enabled).

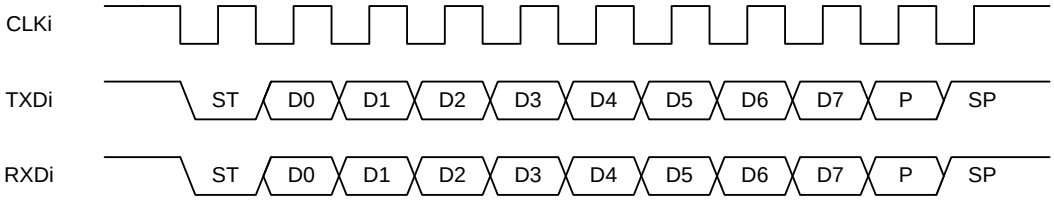
B. Reset procedure for the UiTB register

(1) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(2) Set again bits SMD2 to SMD0 to either of 001b, 101b, or 110b.
(3) Set the TE bit in the UiC1 register to 1 (transmission enabled) irrespective of the bit setting.

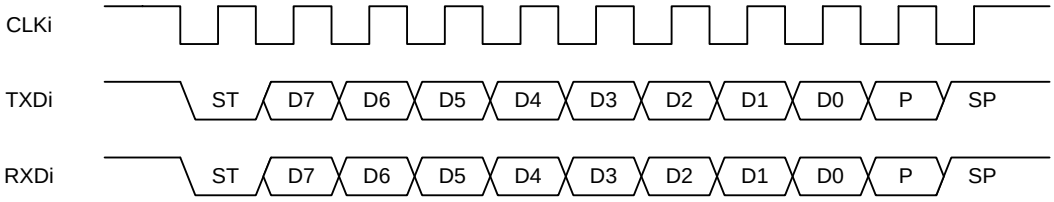
### 18.2.3 LSB First and MSB First Selection

As shown in Figure 18.28, the bit order is selected using the UFORM bit in the UiC0 register (i = 0 to 8). This function is available for the data format of 8-bit character length.

(1) When the UFORM bit in the UiC0 register is set to 0 (LSB first)




(2) When the UFORM bit in the UiC0 register is set to 1 (MSB first)




Note:
1. The figure above applies under the following conditions:
-The UiLCH bit in the UiC1 register = 0 (data is non logic-inverted)
-The STPS bit in the UiMR register = 0 (1 stop bit)
-The PRYE bit = 1 (parity enabled)

ST: Start bit
P: Parity bit
SP: Stop bit

**Figure 18.28 Bit Order (i = 0 to 8)**

### 18.2.4 Serial Data Logical Inversion

When the UiLCH bit in the UiC1 register (i = 0 to 6) is set to 1 (data logic-inverted), logical value written in the UiTB register is inverted to be transmitted. The UiRB register is read as logic-inverted receive data. The parity bit is not inverted. Figure 18.29 shows the logical inversion of serial data.

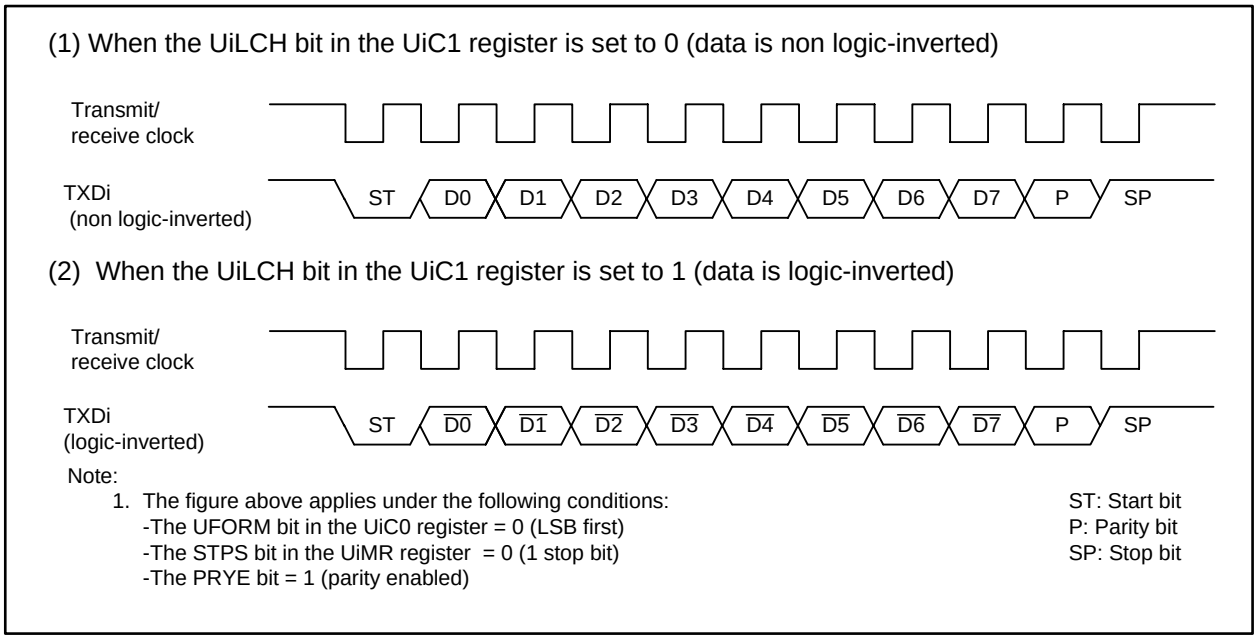



**Figure 18.29 Serial Data Logical Inversion (i = 0 to 6)**

### 18.2.5 TXD and RXD I/O Polarity Inversion

The output level at the TXD pin and the input level at the RXD pin are inverted by this function. All I/O data levels, including the start bit, stop bit, and parity bit are inverted by setting the IOPOL bit in the UiMR register (i = 0 to 6) to 1 (inverted). Figure 18.30 shows TXD and RXD I/O polarity inversion.

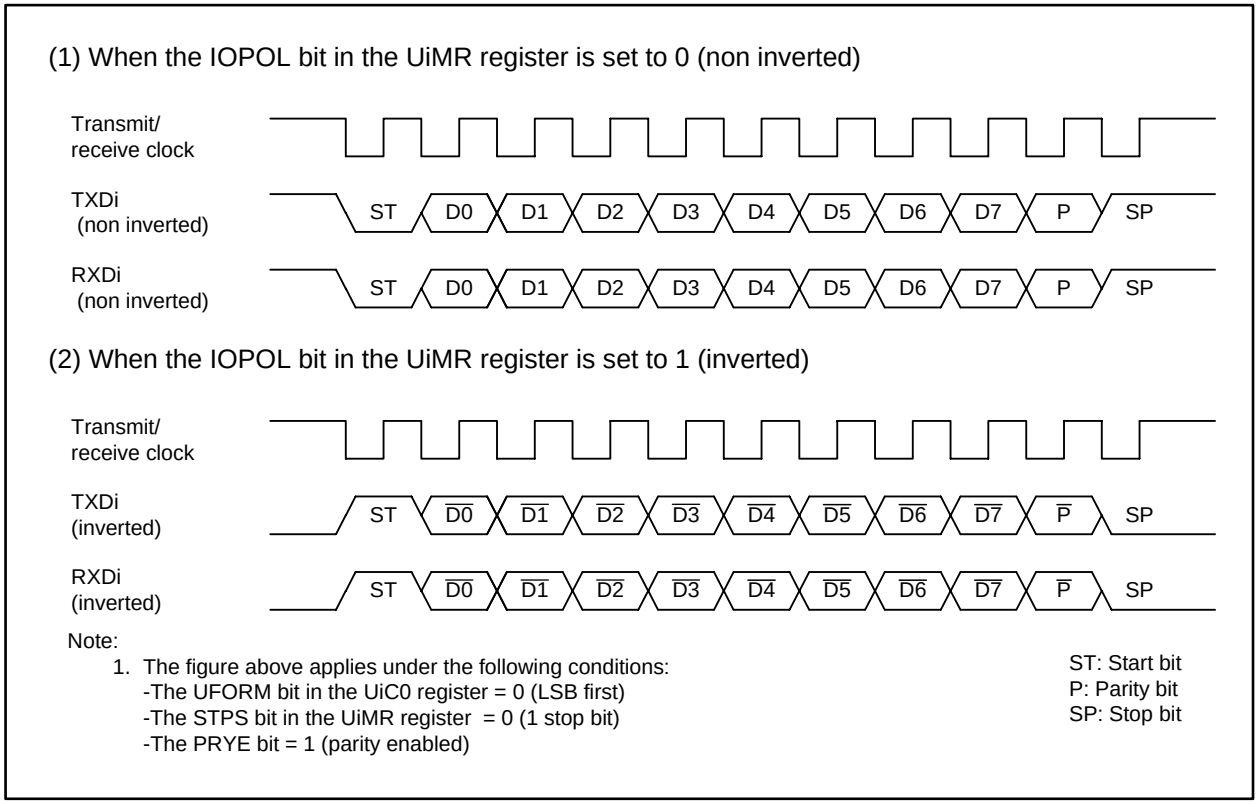



**Figure 18.30 TXD and RXD I/O Polarity Inversion (i = 0 to 6)**

### 18.2.6 CTS/RTS Function

The CTS controls data transmission using the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ pin (i = 0 to 8). When an input level at the pin becomes low, data transmission is started. If the input level changes to high during transmit operation, the transmission of the next data is stopped.

The RTS indicates receiver status using the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ pin. When the MCU is ready to receive data, the output level at the pin becomes low. It becomes high on the first falling edge of the CLKi pin.

## 18.3 Special Mode 1 ($I^2C$ Mode)

This mode uses an $I^2C$-typed interface for communication. Table 18.9 lists specifications of the $I^2C$ mode.

**Table 18.9 $I^2C$ Mode Specifications**

| Item | Specification |
|---|---|
| Data format | 8-bit character length |
| Transmit/receive clock | In master mode<br>• The CKDIR bit in the UiMR register (i = 0 to 6) is set to 0 (internal clock):<br>$\frac{fx}{2(m+1)}$ fx = f1, f8, f2n<br>m: UiBRG register setting value, 00h to FFh<br>In slave mode<br>• The CKDIR bit is set to 1 (external clock): input into the SCLi pin |
| Transmit start conditions | The conditions for starting data transmission are as follows (1):<br>• The TE bit in the UiC1 register is set to 1 (transmission enabled)<br>• The TI bit in the UiC1 register is set to 0 (data held in the UiTB register) |
| Receive start conditions | The conditions for starting data reception are as follows (1):<br>• The RE bit in the UiC1 register is set to 1 (reception enabled)<br>• The TE bit in the UiC1 register is set to 1 (transmission enabled)<br>• The TI bit in the UiC1 register is set to 0 (data held in the UiTB register) |
| Interrupt request generating timing | When any of the following is detected: start condition, stop condition, NACK (Not-Acknowledge), or ACK (Acknowledge) |
| Error detection | Overrun error (2)<br>This error occurs when the eighth bit of the next data has been received before the UiRB register is read |
| Selectable functions | • Arbitration lost<br>Selectable update timing of the ABT bit in the UiRB register<br>• SDAi digital delay<br>Selectable among non digital delay and two to eight cycles of digital delay of UiBRG count source<br>• Clock phase setting<br>Selectable either clock delayed or no clock delay |

Notes:
1. When an external clock is selected, the conditions should be met while the external clock signal is held high.
2. If an overrun error occurs, the UiRB register is undefined. The IR bit in the SiRIC register is not changed to 1 (interrupts requested).

Table 18.10 and Table 18.11 list respectively register settings and functions in $I^2C$ mode. Figure 18.31 shows a block diagram of $I^2C$ mode and Figure 18.32 shows timings for the transfer to the UiRB register (i = 0 to 6) and the interrupt.

As shown in Table 18.11, this mode is available when bits SMD2 to SMD0 in the UiMR register (i = 0 to 6) are set to 010b, and the IICM bit in the UiSMR register is set to 1. Since a transmit signal at the SDAi pin is output via the delay circuit, it changes after the SCLi pin is stably held low.

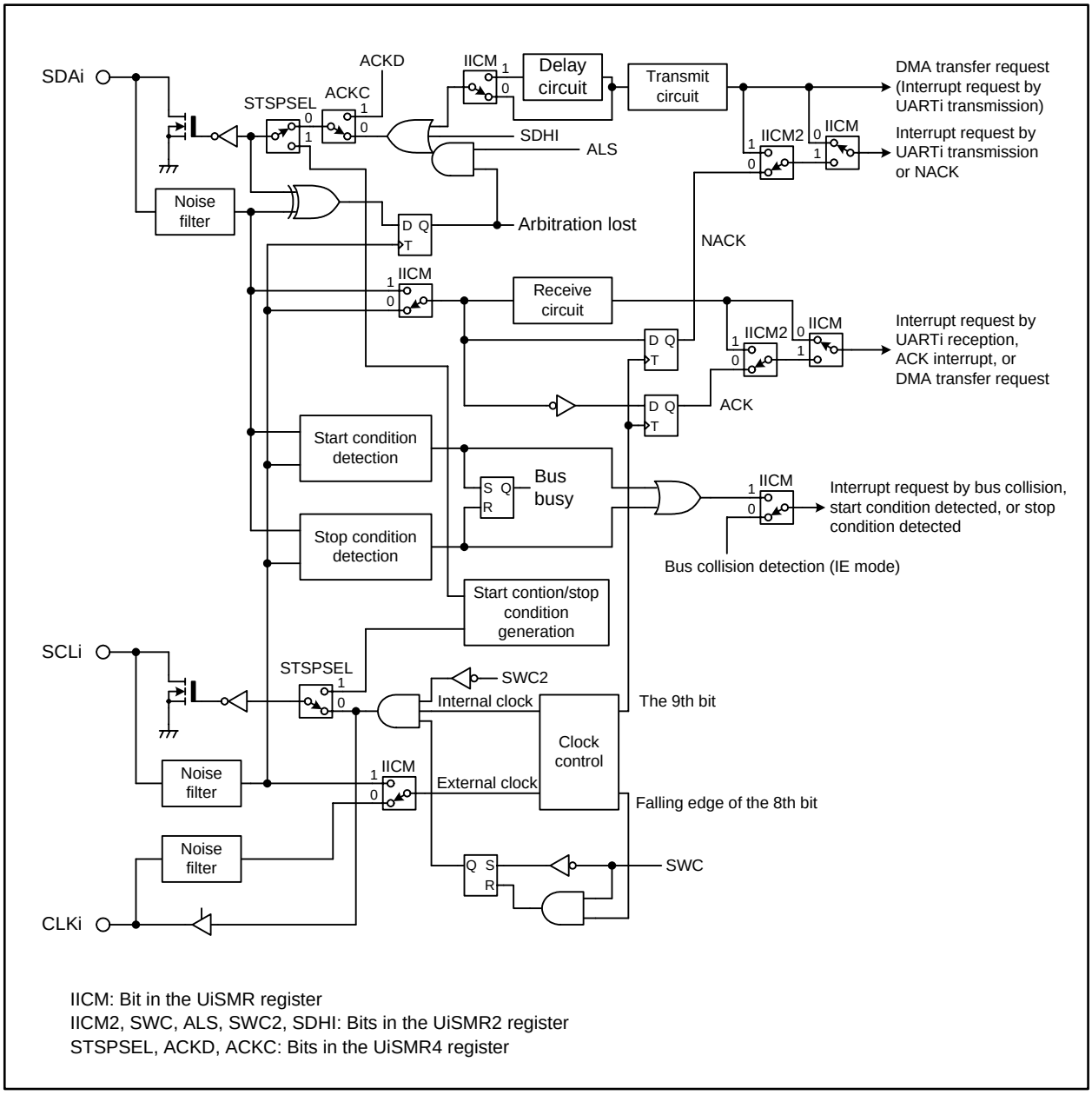



**Figure 18.31 I²C Mode Block Diagram (i = 0 to 6)**

**Table 18.10 Register Settings in I²C Mode (i = 0 to 6)**

| Register | Bits | Function | |
|---|---|---|---|
| | | Master | Slave |
| UiMR | IOPOL | Set the bit to 0 | |
| | CKDIR | Set the bit to 0 | Set the bit to 1 |
| | SMD2 to SMD0 | Set the bit to 010b | |
| UiC0 | 7 to 4 | Set the bits to 1001b | |
| | TXEPT | Transmit register empty flag | |
| | 2 | Set the bit to 0 | |
| | CLK1 and CLK0 | Select a count source for the UiBRG register | Disabled |
| UiC1 | 7 to 5 | Set the bits to 000b | |
| | UiIRS | Set the bit to 1 | |
| | RI | Receive complete flag | |
| | RE | Set the bit to 1 to enable data reception | |
| | TI | Transmit buffer empty flag | |
| | TE | Set the bit to 1 to enable data transmission/reception | |
| UiSMR | 7 to 3 | Set the bits to 00000b | |
| | BBS | Bus busy flag | |
| | ABC | Select an arbitration lost detection timing | Disabled |
| | IICM | Set the bit to 1 | |
| UiSMR2 | 7 | Set the bit to 0 | |
| | SDHI | Set the bit to 1 to disable the SDA output | |
| | SWC2 | Set the bit to 1 to hold the SCL output at a forcible low | |
| | STC | Set the bit to 0 | Set the bit to 1 to reset UARTi by detecting the start condition |
| | ALS | Set the bit to 1 to stop the output at the SDAi pin to detect an arbitration lost | Set the bit to 0 |
| | SWC | Set the bit to 1 to hold a low output at the SCLi pin after receiving the eighth bit of the clock | |
| | CSC | Set the bit to 1 to enable clock synchronization | Set the bit to 0 |
| | IICM2 | Refer to Table 18.11 | |
| UiSMR3 | DL2 to DL0 | Set the digital delay value of SDAi | |
| | 4 to 2 | Set the bit to 000b | |
| | CKPH | Refer to Table 18.11 | |
| | SSE | Set the bit to 0 | |
| UiSMR4 | SWC9 | Set the bit to 0 | Set the bit to 1 to hold a low output at the SCLi pin after receiving the ninth bit of the clock |
| | SCLHI | Set the bit to 1 to stop the SCL output to detect stop condition | Set the bit to 0 |
| | ACKC | Set the bit to 1 for ACK data output | |
| | ACKD | Select ACK or NACK | |
| | STSPSEL | Set the bit to 1 when any condition is output | Set the bit to 0 |
| | STPREQ | Set the bit to 1 to generate a stop condition | Set the bit to 0 |
| | RSTAREQ | Set the bit to 1 to generate a restart condition | Set the bit to 0 |
| | STAREQ | Set the bit to 1 to generate a start condition | Set the bit to 0 |
| UiBRG | 7 to 0 | Set the bit rate | Disabled |
| IFSR0 | IFSR06 and IFSR07 | Select a UART as interrupt source | |
| IFSR1 | IFSR16 | Select a UART as interrupt source | |
| IFS0 | IFS06 | Select input pins for SCL3 and SDA3 | |
| | IFS03 and IFS02 | Select input pins for SCL6 and SDA6 | |
| UiTB | 8 | Set the bit to 1 on transmission. Set the bit to the value of the ACK bit on reception | |
| | 7 to 0 | Set the data to be transmitted on transmission. Set the register to FFh on reception | |
| UiRB | OER | Overrun error flag | |
| | ABT | Arbitration lost detection flag | Disabled |
| | 8 | D0 is loaded immediately after a receive interrupt is generated. ACK or NACK is loaded after a transmit interrupt is generated | |
| | 7 to 0 | D7 to D1 are read immediately after a receive interrupt is generated. D7 to D0 are read after a transmit interrupt is generated | |

**Table 18.11 I²C Mode Functions (i = 0 to 6)**

| Function | Synchronous Serial Interface Mode (SMD2 to SMD0 = 001b, IICM = 0) | I²C Mode (SMD2 to SMD0 = 010b, IICM = 1) | | | |
|---|---|---|---|---|---|
| | | IICM2 = 0 (ACK/NACK interrupt) | | IICM2 = 1 (Transmit/receive interrupt) | |
| | | CKPH = 0 (Non clock delayed) | CKPH = 1 (Clock delayed) | CKPH = 0 (Non clock delayed) | CKPH = 1 (Clock delayed) |
| Source of software interrupt numbers 6, 39 to 41 (1) (Refer to Figure 18.32) | — | Start condition or stop condition detection (Refer to Table 18.12) | | | |
| Source of software interrupt numbers 2, 4, 17, 19, 33, 35, and 37 (1) (Refer to Figure 18.32) | UARTi transmission: Transmission started or completed (selected using the UiIRS register) | NACK detection: Rising edge of the ninth bit of SCLi | | UARTi transmission: Rising edge of the ninth bit of SCLi | UARTi transmission: Falling edge of the ninth bit of SCLi |
| Source of software interrupt numbers 3, 5, 18, 20, 34, 36, and 38 (1) (Refer to Figure 18.32) | UARTi reception: Receiving at eighth bit CKPOL= 0 (rising edge) CKPOL= 1 (falling edge) | ACK detection: Rising edge of the ninth bit of SCLi | | UARTi reception: Falling edge of the eighth bit of SCLi | |
| Data transfer timing from the UART receive register to the UiRB register | CKPOL = 0 (rising edge) CKPOL = 1 (falling edge) | Rising edge of the ninth bit of SCLi | | Falling edge of the eighth bit of SCLi | Falling edge of the eighth bit and rising edge of the ninth bit of SCLi |
| UARTi transmit output delay | Non delayed | Delayed | | | |
| Pins P6_3, P6_7, P7_0, P7_3, P7_6, P9_2, P9_6, P11_0, P12_0, P15_0, and P15_4 | TXDi output | SDAi I/O | | | |
| Pins P6_2, P6_6, P7_1, P7_5, P8_0, P9_1, P9_7, P11_2, P12_2, P15_2, and P15_5 | RXDi input | SCLi I/O | | | |
| Pins P6_1, P6_5, P7_2, P7_4, P7_7, P9_0, P9_5, P11_1, P12_1, P15_1, and P15_6 | Select CLKi input or output | — (Not used in I²C mode) | | | |
| Read level at pins RXDi and SCLi | Readable irrespective of the port direction bit | | | | |
| Default output value at the SDAi pin | — | High (Value set in the Port Pi register (i = 0 to 7) if the I/O port is selected by output function select registers) | | | |
| SCLi default and end values | — | High | Low | High | Low |
| DMA source (Refer to Figure 18.32) | UARTi reception | ACK detection | | UARTi reception: Falling edge of the eighth bit of SCLi | |

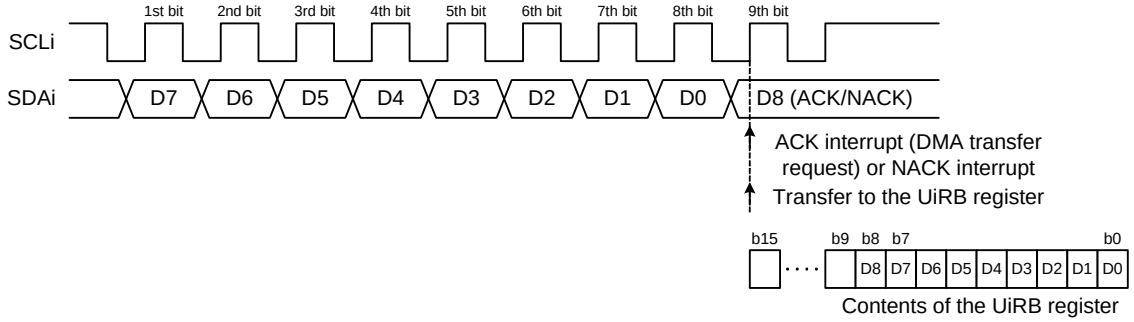
**Table 18.11 I²C Mode Functions (i = 0 to 6)**

| Function | Synchronous Serial Interface Mode (SMD2 to SMD0 = 001b, IICM = 0) | I²C Mode (SMD2 to SMD0 = 010b, IICM = 1) | | | |
|---|---|---|---|---|---|
| | | IICM2 = 0 (ACK/NACK interrupt) | | IICM2 = 1 (Transmit/receive interrupt) | |
| | | CKPH = 0 (Non clock delayed) | CKPH = 1 (Clock delayed) | CKPH = 0 (Non clock delayed) | CKPH = 1 (Clock delayed) |
| Store received data | The first to eighth bits of received data are stored into bits 0 to 7 in the UiRB register | The first to eighth bits of received data are stored into bits 7 to 0 in the UiRB register | | The first to seventh bits of received data are stored into bits 6 to 0 in the UiRB register and the eighth bit is stored into the bit 8 | Same as on the left column on the first data storing (2). The first to eighth bits of received data are stored into 7 to 0 bits in the UiRB register and the ninth bit is stored into the bit 8 on the second data storing (3) |
| Read received data | The UiRB register status is read as it is | | | Bits 6 to 0 in the UiRB register are read as bits 7 to 1 and the bit 8 is read as bit 0 | Same as on the left column on the first read (2). The UiRB register status is read as it is on the second read (3) |

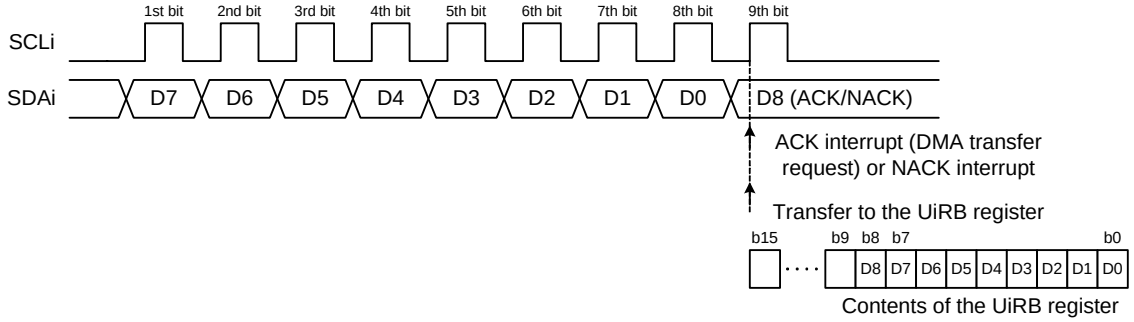
Notes:

1. The procedure to change interrupt sources is as follows:
   (1) Disable the interrupt of the corresponding software interrupt number.
   (2) Change the source of interrupt.
   (3) Set the IR bit of the corresponding software interrupt number to 0 (no interrupt requested).
   (4) Set bits ILVL2 to ILVL0 of the corresponding software interrupt number.
2. The first data transfer to the UiRB register starts on the rising edge of the eighth bit of SCLi.
3. The second data transfer to the UiRB register starts on the rising edge of the ninth bit of SCLi.

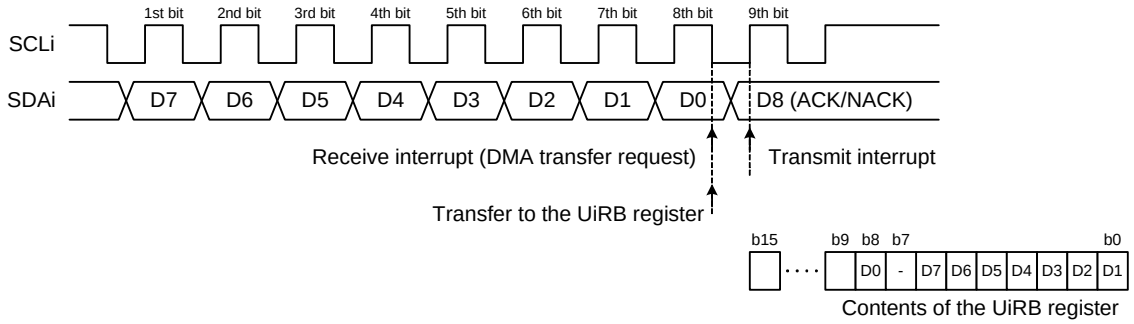
(1) When the IICM2 bit is set to 0 (ACK or NACK interrupt) and the CKPH bit is set to 0 (non clock delayed)



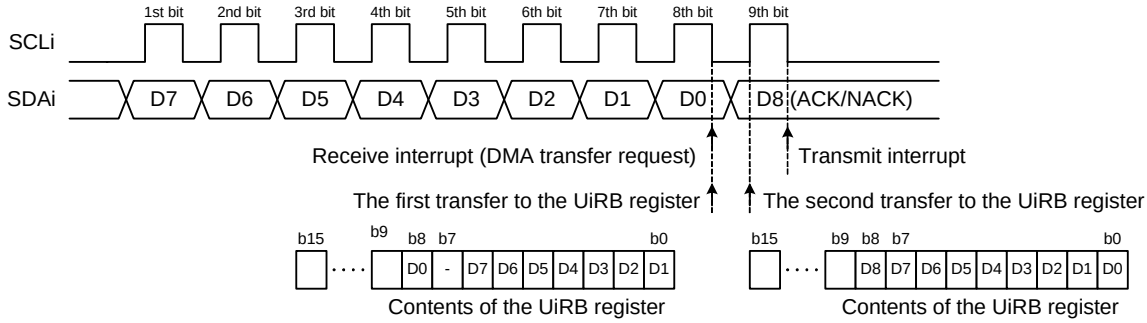

(2) When the IICM2 bit is set to 0 and the CKPH bit is set to 1 (clock delayed)




(3) When the IICM2 bit is set to 1 (transmit/receive interrupt) and the CKPH bit is set to 0




(4) When the IICM2 bit is set to 1 and the CKPH bit is set to 1




The figure above applies under the following condition:
-The CKDIR bit in the UiMR register = 0 (internal clock)

**Figure 18.32 Timings for the Transfer and Interrupt to the UiRB Register (i = 0 to 6)**

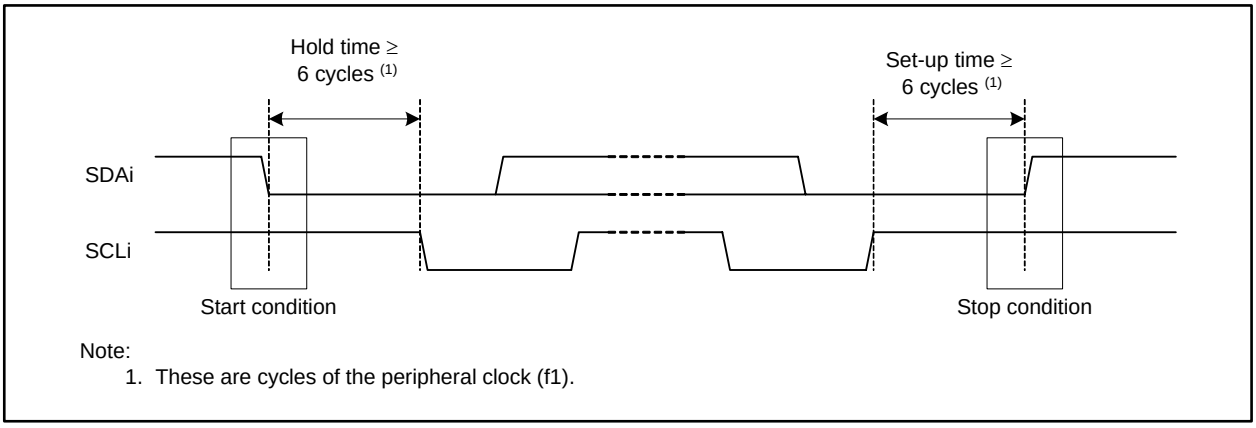
### 18.3.1 Start Condition and Stop Condition Detection

The start condition and stop condition are detected by their respective detectors.

The start condition detection interrupt request is generated by a high-to-low transition at the SDAi pin while the SCLi (i = 0 to 6) pin is held high. The stop condition detection interrupt request is generated by a low-to-high transition at the SDAi pin while the SCLi pin is held high.

The start condition detection interrupt shares interrupt control registers and vectors with the stop condition detection interrupt. The BBS bit in the UiSMR register determines which interrupt is requested.

To detect a start condition or stop condition, both set-up and hold times require six cycles or more of the peripheral clock (f1) as shown in Figure 18.33. To meet the condition for the Fast-mode specification, f1 is required to be 10 MHz or more.




Note:
1. These are cycles of the peripheral clock (f1).

**Figure 18.33 Start Condition and Stop Condition Detection Timing (i = 0 to 6)**

### 18.3.2 Start Condition and Stop Condition Generation

The start condition, restart condition, and stop condition are generated by bits STARREQ, RSTAREQ, and STPREQ in the UiSMR4 register (i = 0 to 6), respectively. To output the start condition, the STSPSEL bit in the UiSMR4 register should be set to 1 (start condition/stop condition generator selected) after setting the STAREQ bit to 1 (start). To output the restart condition and stop condition, the STSPSEL bit should be set to 1 after setting respective bits RSTAREQ and STPREQ to 1.

Table 18.12 and Figure 18.34 show the functions of the STSPSEL bit.

**Table 18.12 STSPSEL Bit Functions**

| Function | STSPSEL = 0 | STSPSEL = 1 |
|---|---|---|
| Start condition and stop condition generation | Output is provided by the program with port (no auto generation by hardware) | Start condition or stop condition is output according to the STAREQ, RSTAREQ, or STPREQ bit, respectively |
| Start condition and stop condition interrupt request generating timing | When start condition or stop condition is detected | When start condition or stop condition generation is completed |

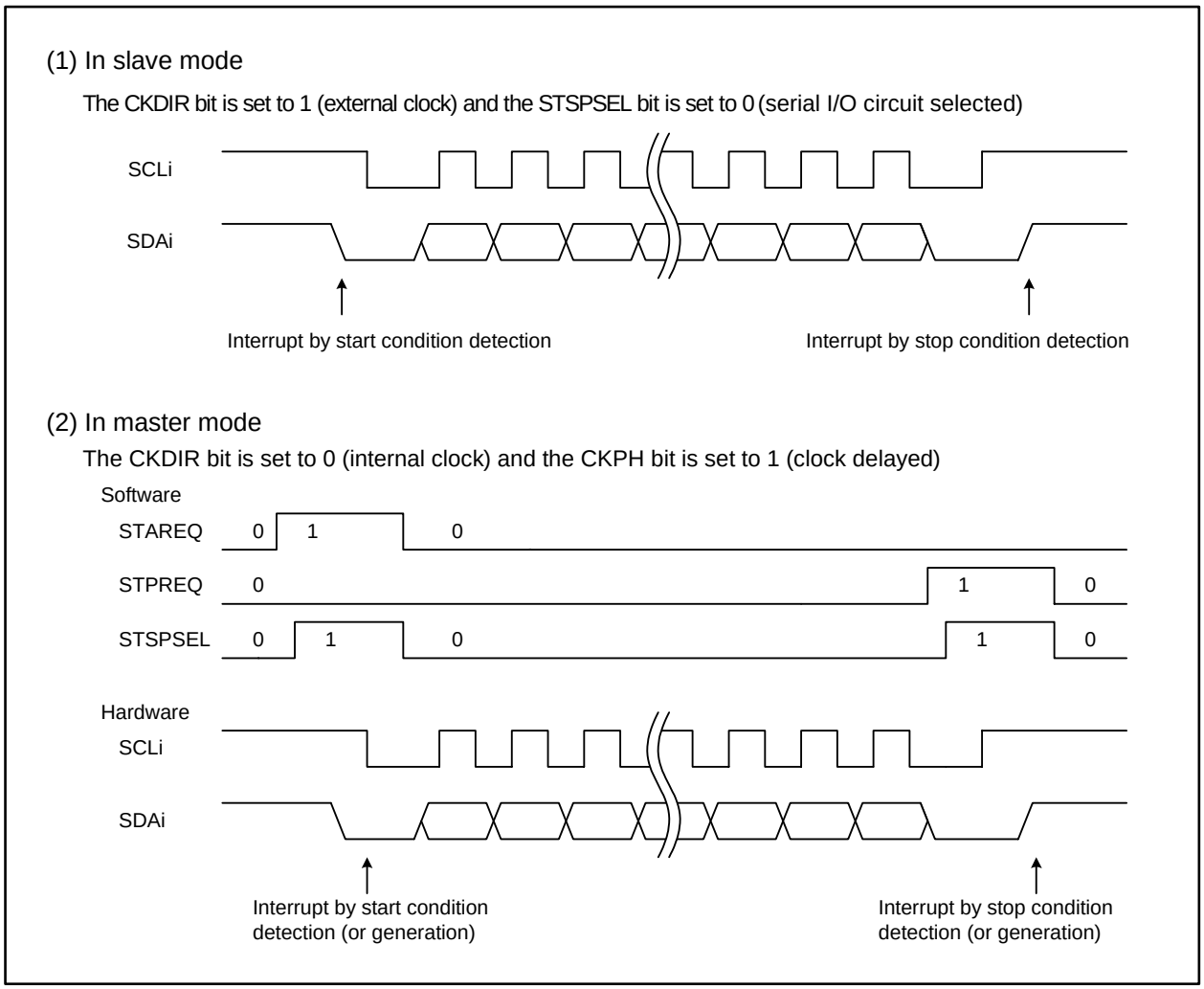



**Figure 18.34 STSPSEL Bit Functions (i = 0 to 6)**

### 18.3.3 Arbitration

The MCU determines whether the transmit data matches data input to the SDAi pin on the rising edge of the SCLi. If it does not match the input data the arbitration takes place at the SDAi pin by switching off the data output stage.

The ABC bit in the UiSMR register (i = 0 to 6) determines the update timing for the ABT bit in the UiRB register.

When the ABC bit is set to 0 (update per bit), the ABT bit is set to 1 (arbitration is lost) as soon as a data discrepancy is detected. If not detected, the ABT bit is set to 0 (arbitration is won). When the ABC bit is set to 1 (update per byte), the ABT bit is set to 1 on the falling edge of the eighth bit of the SCLi if any discrepancy is detected. In this ABC bit setting, the ABT bit should be set to 0 to start the next 1-byte transfer.

When the ALS bit in the UiSMR2 register is set to 1 (SDA output stop enabled), an arbitration lost occurs. As soon as the ABT bit is set to 1, the SDAi pin becomes high-impedance.

### 18.3.4 SCL Control and Clock Synchronization

Data transmission/reception in $I^2C$ mode uses the transmit/receive clock as shown in Figure 18.32. The clock speed increase makes it difficult to secure the required time for ACK generation and data transmit procedure. The $I^2C$ mode supports a function of wait-state insertion to secure this required time and a function of clock synchronization with a wait-state inserted by other devices.

The SWC bit in the UiSMR2 register (i = 0 to 6) is used to insert a wait-state for ACK generation. When the SWC bit is set to 1 (the SCLi pin is held low after the eighth bit of the SCLi is received), the SCLi pin is held low on the falling edge of the eighth bit of the SCLi. When the SWC bit is set to 0 (no wait-state/ wait-state cleared), the SCLi line is released.

When the SWC2 bit in the UiSMR2 register is set to 1 (the SCLi pin is held low), the SCLi pin is forced low even during transmission or reception in progress. When the SWC2 bit is set to 0 (transmit/receive clock is output at the SCLi pin), the SCLi line is released to output the transmit/receive clock.

The SWC9 bit in the UiSMR4 register is used to insert a wait-state for checking received acknowledge bits. While the CKPH bit in the UiSMR3 register is set to 1 (clock delayed), when the SWC9 bit is set to 1 (the SCLi pin is held low after the ninth bit of the SCLi is received), the SCLi pin is held low on the falling edge of the ninth bit of the SCLi. When the SWC9 bit is set to 0 (no wait-state/wait-state cleared), the SCLi line is released.

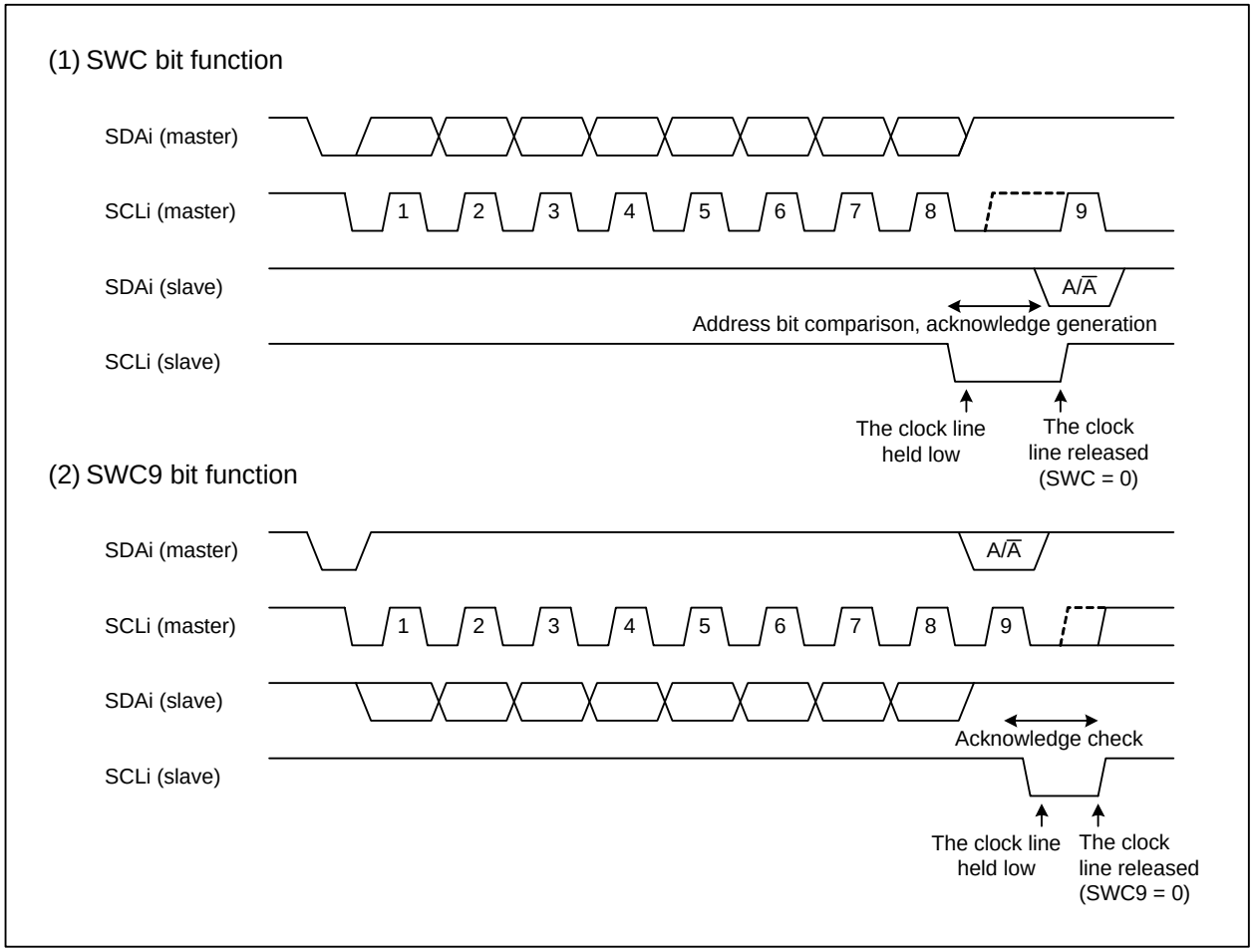



**Figure 18.35 Wait-state Insertion by Bits SWC or SWC9 (i = 0 to 6)**

The CSC bit in the UiSMR2 register is to synchronize an internally generated clock with the clock applied to the SCLi pin. For example, if a wait-state is inserted from other devices, the two clocks are not synchronized. While the CSC bit is set to 1 (clock synchronization enabled) and the internal clock is held high, when a high at the SCLi pin changes to low, the internal clock becomes low in order to reload the value of the UiBRG register and to resume counting. While the SCLi pin is held low, when the internal clock changes from low to high, the count is stopped until the SCLi pin becomes high. That is, the UARTi transmit/receive clock is the logical AND of the internal clock and the SCLi. The synchronized period starts from one clock prior to the first synchronized clock and ends when the ninth clock is completed. The CSC bit can be set to 1 only when the CKDIR bit in the UiMR register is set to 0 (internal clock selected).

The SCLHI bit in the UiSMR4 register is used to leave the SCLi pin open when other master generates a stop condition while the master is in transmit/receive operation. If the SCLHI bit is set to 1 (output stopped), the SCLi pin is open (the pin is high-impedance) when a stop condition is detected and the clock output is stopped.

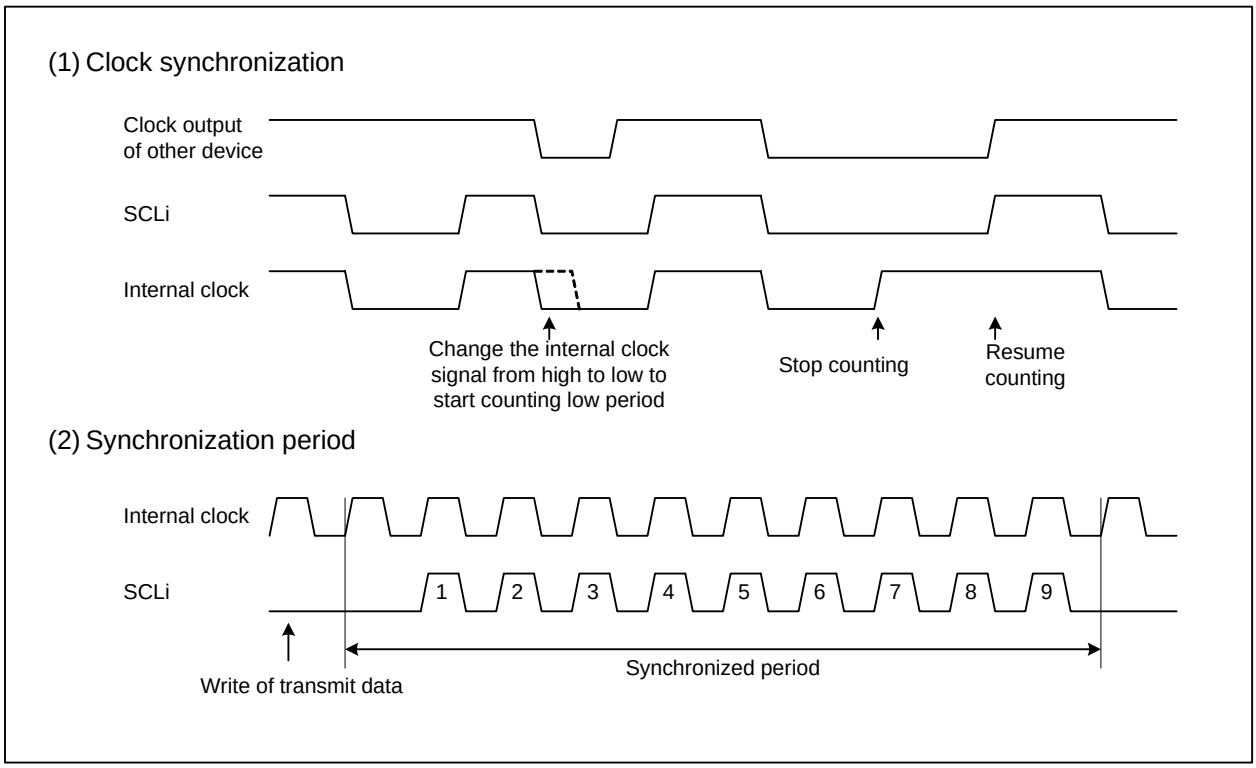



**Figure 18.36 Clock Synchronization (i = 0 to 6)**

### 18.3.5 SDA Output

Values set in bits 8 to 0 (D8 to D0) in the UiTB register (i = 0 to 6) are output starting from D7 to D0, and lastly D8, which is a bit for the acknowledge signal. When transmitting, D8 should be set to 1 to free the bus. When receiving, D8 should be set to ACK or NACK.

Bits DL2 to DL0 in the UiSMR3 register set a delay time of the SDAi on the falling edge of the SCLi. Based on the UiBRG count source, the delay time can be selected from zero cycles (no delay) and two to eight cycles.

When the SDHI bit in the UiSMR2 register is set to 1 (SDA output disabled), the SDAi pin forcibly becomes high-impedance. Output at the SDAi pin is low if the I/O port is selected for the SDAi and the pin is specified as the output port after selecting $I^2C$ mode. In this case, if the SDHI bit is set to 1, the SDAi pin becomes high-impedance.

When the SDHI bit is rewritten while the SCLi pin is held high, a start condition or stop condition is generated. When it is rewritten immediately before the rising edge of the SCLi, an arbitration lost may be accidently detected. Therefore, the SDHI bit should be rewritten so the SDAi pin level changes while the SCLi pin is low.

### 18.3.6 SDA Input

When the IICM2 bit in the UiSMR2 register (i = 0 to 6) is set to 0, the first eight bits of received data (D7 to D0) are stored into bits 7 to 0 in the UiRB register and the ninth bit (ACK/NACK) is stored into the bit 8.

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

If the IICM2 bit is set to 1 and the CKPH bit in the UiSMR3 register is set to 1 (clock delayed), the same data that is set when the IICM2 bit is 0 can be read. To read this data, read the UiRB register after data in the ninth bit is latched on the rising edge of the SCLi.

### 18.3.7 Acknowledge

When a data is to be received in master mode, the ACK is output after eight bits are received by setting the UiTB register to 00FFh as dummy data. When the STSPSEL bit in the UiSMR4 register (i = 0 to 6) is set to 0 (serial I/O circuit selected) and the ACKC bit is set to 1 (ACK data output), the value of the ACKD bit is output at the SDAi pin.

If the IICM2 bit is set to 0, the NACK interrupt request is generated when the SDAi pin is held high on the rising edge of the ninth bit of the SCLi. The ACK interrupt request is generated when the SDAi pin is held low.

If the DMA request source is “UARTi receive interrupt request or ACK interrupt request”, the DMA transfer is activated when an ACK is detected.

### 18.3.8 Initialization of Transmit/Receive Operation

When the CKDIR bit in the UiMR register (i = 0 to 6) is set to 1 (external clock selected) and the STC bit in the UiSMR2 register is set to 1 (the circuit is initialized), and a start condition is detected, the following three operations are performed:

- The transmit register is reset and the content of the UiTB register is transferred to the transmit register. The new data transmission starts on the falling edge of the first bit of the next SCLi as transmit clock. The content of the transmit register before the reset is output at the SDAi pin in the period from the falling edge of the SCLi until the first data output.
- The receive register is reset and the new data reception starts on the falling edge of the first bit of the next SCLi.
- The SWC bit in the UiSMR2 register is set to 1 (the SCL pin is held low after the eighth bit of the SCLi is received).

If UARTi transmission/reception is started with this function, the TI bit in the UiC1 register does not change.

## 18.4 Special Mode 2

Special mode 2 enables serial communication between one or multiple masters and multiple slaves. The $\overline{SSi}$ input pin (i = 0 to 6) controls the serial bus communication. Table 18.13 lists specifications of special mode 2.

**Table 18.13 Special Mode 2 Specifications**

| Item | Specification |
|---|---|
| Data format | 8-bit character length |
| Transmit/receive clock | • The CKDIR bit in the UiMR register (i = 0 to 6) is set to 0 (internal clock):<br>$\frac{fx}{2(m+1)}$ fx = f1, f8, f2n m: UiBRG register setting value, 00h to FFh<br>• The CKDIR bit is set to 1 (external clock): input into the CLKi pin |
| Transmit/receive control | SS function |
| Transmit start conditions | The conditions for starting data transmission are as follows (1):<br>• The TE bit in the UiC1 register is set to 1 (transmission enabled)<br>• The TI bit in the UiC1 register is set to 0 (data held in the UiTB register) |
| Receive start conditions | The conditions for starting data reception are as follows (1):<br>• The RE bit in the UiC1 register is set to 1 (reception enabled)<br>• The TE bit in the UiC1 register is set to 1 (transmission enabled)<br>• The TI bit in the UiC1 register is set to 0 (data held in the UiTB register) |
| Interrupt request generating timing | In transmit interrupt, one of the following conditions is selected to set the UiIRS bit in registers U0C1 to U6C1:<br>• The UiIRS bit is set to 0 (transmit buffer in the UiTB register is empty):<br>when data is transferred from the UiTB register to the UARTi transmit register (when the transmission has started)<br>• The UiIRS bit is set to 1 (transmission is completed):<br>when data transmission from the UARTi transmit register is completed<br>In receive interrupt,<br>• When data is transferred from the UARTi receive register to the UiRB register (when the reception is completed) |
| Error detection | Overrun error (2)<br>This error occurs when the seventh bit of the next data has been received before reading the UiRB register |
| Selectable functions | • CLK polarity<br>Selectable either rising or falling edge of the transmit/receive clock for transfer data input and output<br>• Bit order selection<br>Selectable either LSB first or MSB first<br>• Continuous receive mode<br>Data reception is enabled by a read access to the UiRB register<br>• Serial data logical inversion<br>This function logically inverses transmit/receive data<br>• Clock phase selection<br>Selectable from one of four combinations of transmit/receive clock polarity and phases<br>• $\overline{SSi}$ input pin function<br>Output pin can be high-impedance when the $\overline{SSi}$ pin is high |

Notes:
1. In case external clock is selected, the following preconditions should be met:
   • The CLKi pin is held high when the CKPOL bit in the UiC0 register is set to 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge)
   • The CLKi pin is held low when the CKPOL bit is set to 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge)
2. If an overrun error occurs, the UiRB register is undefined. The IR bit in the SiRIC register is not changed to 1 (interrupts requested).

Table 18.14 lists register settings in special mode 2.

**Table 18.14 Register Settings in Special Mode 2 (i = 0 to 6)**

| Register | Bits | Function |
|---|---|---|
| UiMR | 7 to 4 | Set the bits to 0000b |
| | CKDIR | Set the bit to 0 in master mode and set it to 1 in slave mode |
| | SMD2 to SMD0 | Set the bits to 001b |
| UiC0 | UFORM | Select either LSB first or MSB first |
| | CKPOL | Clock phase can be set by the combination of bits CKPOL and CKPH in the UiSMR3 register |
| | 5 | Set the bit to 0 |
| | CRD | Set the bit to 1 |
| | TXEPT | Transmit register empty flag |
| | 2 | Set the bit to 0 |
| | CLK1 and CLK0 | Select a count source for the UiBRG register |
| UiC1 | 7 and 6 | Set the bits to 00b |
| | UiRRM | Set the bit to 1 to use continuous receive mode |
| | UiIRS | Select a source for UARTi transmit interrupt |
| | RI | Receive complete flag |
| | RE | Set the bit to 1 to enable data reception |
| | TI | Transmit buffer empty flag |
| | TE | Set the bit to 1 to enable data transmission/reception |
| UiSMR | 7 to 0 | Set the bits to 00h |
| UiSMR2 | 7 to 0 | Set the bits to 00h |
| UiSMR3 | 7 to 5 | Set the bits to 000b |
| | ERR | Mode fault flag |
| | 3 | Set the bit to 0 |
| | DINC | Set to 0 in master mode and set to 1 in slave mode |
| | CKPH | Clock phase can be set by the combination of bits CKPH and CKPOL in the UiC0 register |
| | SSE | Set the bit to 1 |
| UiSMR4 | 7 to 0 | Set the bits to 00h |
| UiBRG | 7 to 0 | Set the bit rate |
| IFS0 | IFS06 | Select input pins for CLK3, RXD3, SRXD3, and $\overline{SS3}$ |
| | IFS03 and IFS02 | Select input pins for CLK6, RXD6, SRXD6, and $\overline{SS6}$ |
| UiTB | 7 to 0 | Set the data to be transmitted |
| UiRB | OER | Overrun error flag |
| | 7 to 0 | Received data is read |

### 18.4.1 $\overline{SSi}$ Input Pin Function (i = 0 to 6)

Special mode 2 is selected by setting the SSE bit in the UiSMR3 register to 1 (SS enabled). The $\overline{CTSi}$/$\overline{RTSi}$/$\overline{SSi}$ pin functions as $\overline{SSi}$ input.

The DINC bit in the UiSMR3 register determines which MCU performs as master or slave.

When multiple MCUs perform as master (multi-master system), the $\overline{SSi}$ pin setting determines which master MCU is active and when.

#### 18.4.1.1 SS Function in Slave Mode

When the DINC bit is set to 1 (slave mode selected) while an input at the $\overline{SSi}$ pin is high, the STXDi pin becomes high-impedance and the clock input at the CLKi pin is ignored. When an input at the $\overline{SSi}$ pin is low, the clock input is valid and serial data is output from the STXDi pin to enable serial communication.

#### 18.4.1.2 SS Function in Master Mode

When the DINC bit is set to 0 (master mode selected) while an input at the $\overline{SSi}$ pin is high, which means there is the only one master MCU or no other master MCU is active, the MCU as master starts communication. The master provides the transmit/receive clock output at the CLKi pin. When input at the $\overline{SSi}$ pin is low, which means that there are more masters, pins TXDi and CLKi become high-impedance. This error is called a mode fault. It can be verified using the ERR bit in the UiSMR3 register. The ongoing data transmission/reception is not stopped even if a mode fault occurs. To stop transmission/reception, bits SMD2 to SMD0 in the UiMR register should be set to 000b (serial interface disabled).

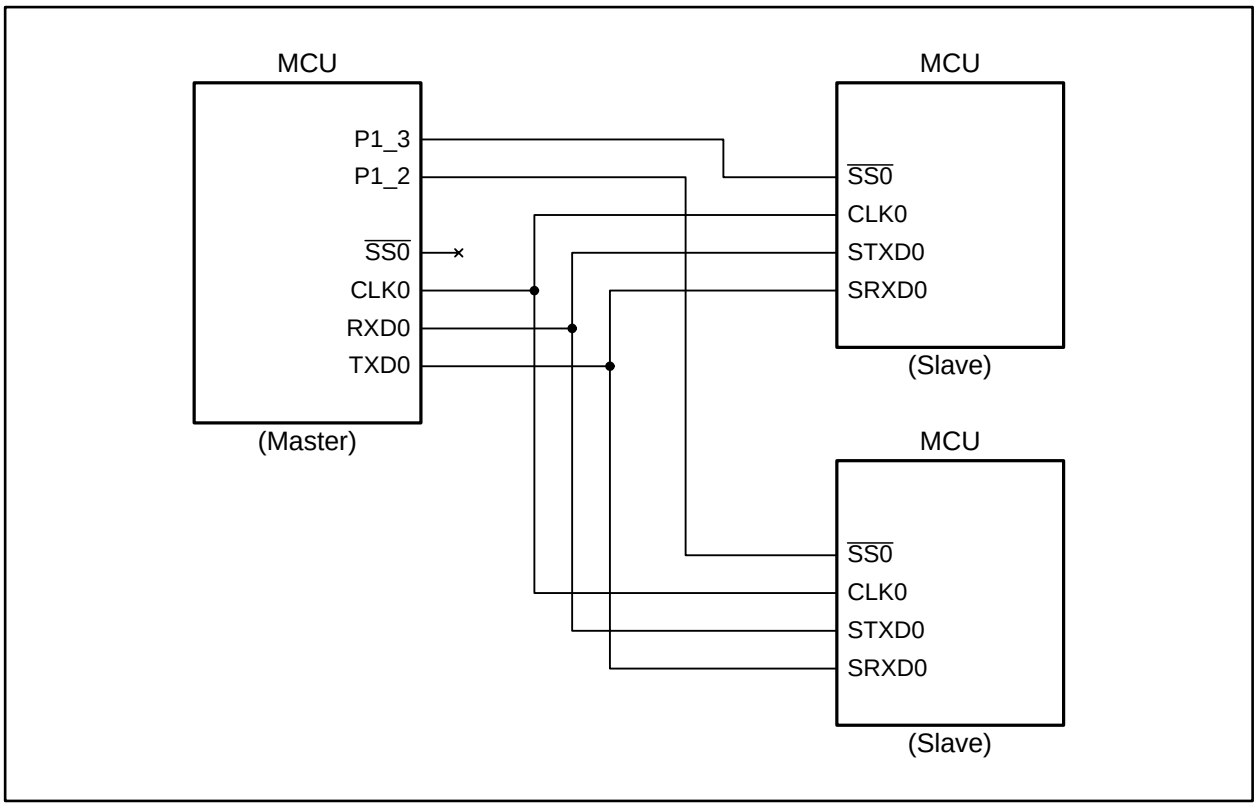



**Figure 18.37 Serial Bus Communication Control with $\overline{SSi}$ Pin**

### 18.4.2 Clock Phase Setting

The CKPH bit in the UiSMR3 register (i = 0 to 6) and the CKPOL bit in the UiC0 register select one of four combinations of transmit/receive clock polarity and serial clock phase.

The transmit/receive clock phase and polarity should be identical for the master device and the communicating slave device.

#### 18.4.2.1 Transmit/Receive Timing in Master Mode

When the DINC bit is set to 0 (master mode selected), the CKDIR bit in the UiMR register should be set to 0 (internal clock selected) to generate the clock. Figure 18.38 shows transmit/receive timing of each clock phase.

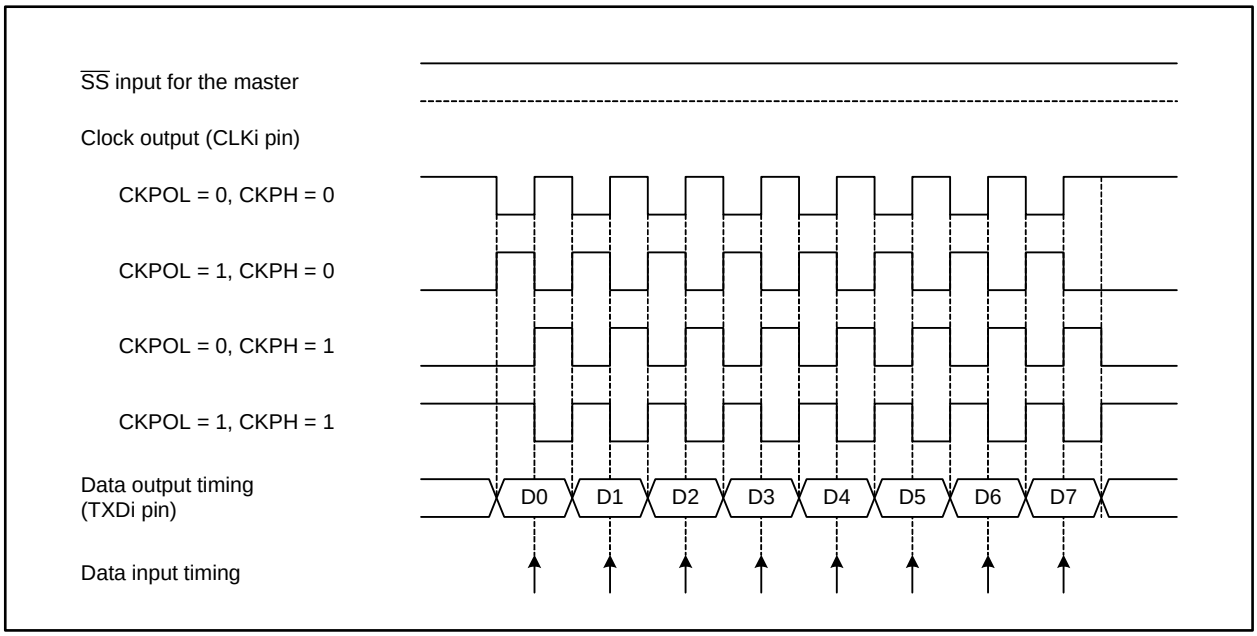



**Figure 18.38 Transmit/Receive Timing in Master Mode**

#### 18.4.2.2 Transmit/Receive Timing in Slave Mode

When the DINC bit is set to 1 (slave mode selected), the CKDIR bit in the UiMR register should be set to 1 (external clock selected).

When the CKPH bit is set to 0 (non clock delayed) while input at the $\overline{SSi}$ pin is high, the STXDi pin becomes high-impedance. When input at the $\overline{SSi}$ pin is low, the conditions for data transmission are all met, but output is undefined. Then the data transmission/reception starts synchronizing with the clock. Figure 18.39 shows the transmit/receive timing.

When the CKPH bit is set to 1 (clock delayed) while an input at the $\overline{SSi}$ pin is high, the STXDi pin becomes high-impedance. When an input at the $\overline{SSi}$ pin is low, the first data is output. Then the data transmission starts synchronizing with the clock. Figure 18.40 shows the transmit/receive timing.

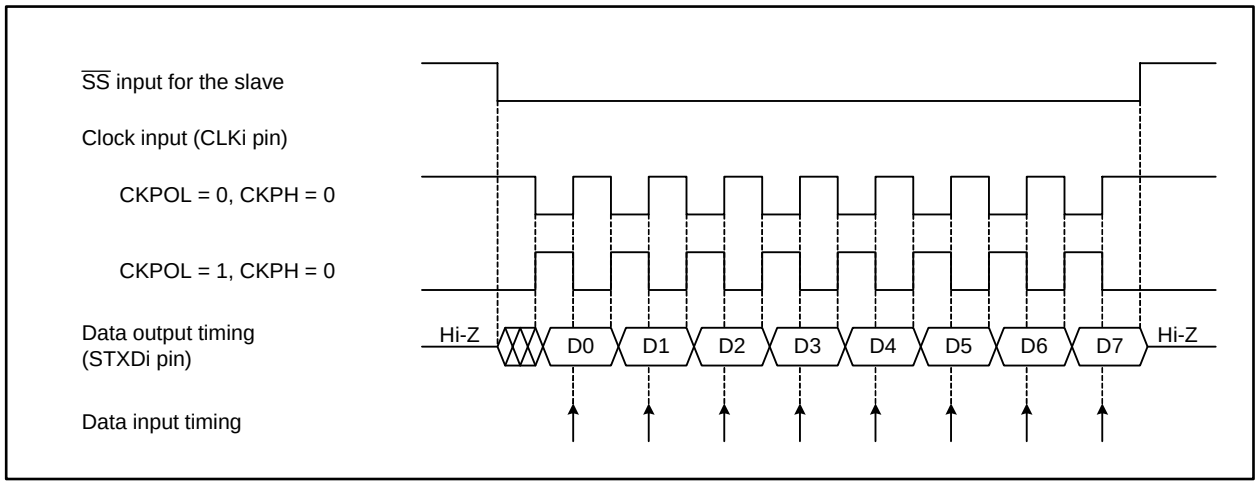



**Figure 18.39 Transmit/Receive Timing in Slave Mode (CKPH = 0)**

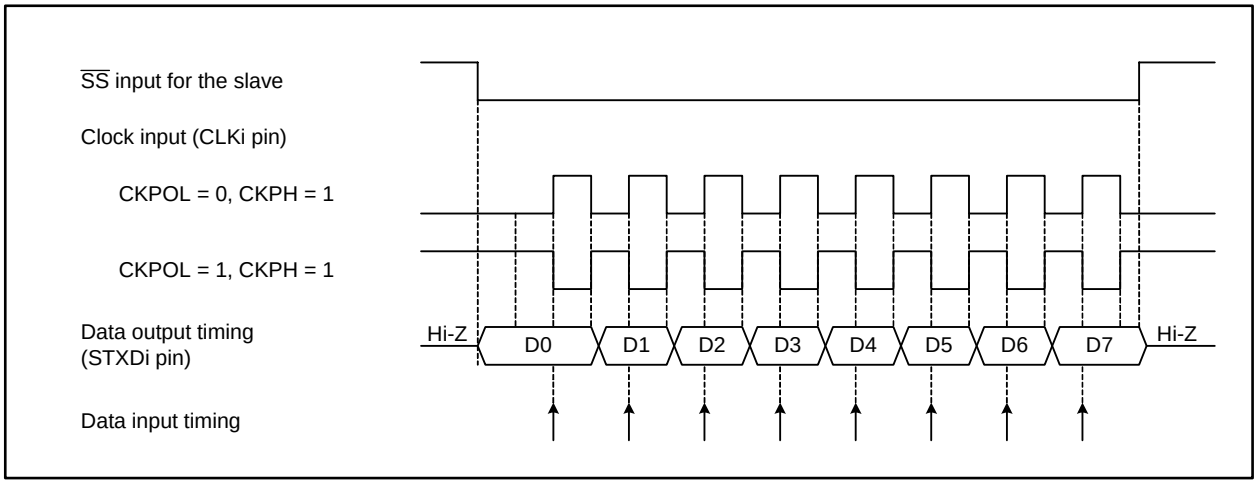



**Figure 18.40 Transmit/Receive Timing in Slave Mode (CKPH = 1)**

## 18.5 Notes on Serial Interface

### 18.5.1 Changing the UiBRG Register (i = 0 to 8)

- Set the UiBRG register after setting bits CLK1 and CLK0 in the UiC0 register. When these bits are changed, the UiBRG register must be set again.
- If a clock is input immediately after the UiBRG register is set to 00h, the counter reaches FFh. In this case, it requires an extra 256 clocks to reload 00h into the register. Once the 00h is reloaded, the counter performs the operation without dividing the count source according to the setting.

### 18.5.2 Synchronous Serial Interface Mode

#### 18.5.2.1 Selecting an External Clock

- If an external clock is selected, the following conditions must be met while the external clock is held high when the CKPOL bit in the UiC0 register (i = 0 to 8) is set to 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge), or while the external clock is held low when the CKPOL bit is set to 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge):
  - The TE bit in the UiC1 register is set to 1 (transmission enabled)
  - The RE bit in the UiC1 register is set to 1 (reception enabled)
  - The TI bit in the UiC1 register is set to 0 (data held in the UiTB register)
  - The RE bit setting is not required in transmit operation only.

#### 18.5.2.2 Receive Operation

- In synchronous serial interface mode, the transmit/receive clock is controlled by the transmit control circuit. Set the UARTi-associated registers (i = 0 to 8) for a transmit operation, even if the MCU is used only for receive operation. Dummy data is output from the TXDi pin while receiving if the TXDi pin is set to output mode.

- If data is received continuously, an overrun error occurs when the RI bit in the UiC1 register is 1 (data held in the UiRB register) and the seventh bit of the next data is received in the UARTi receive shift register. Then, the OER bit in the UiRB register becomes 1 (overrun error occurred). In this case, the UiRB register becomes undefined. If an overrun error occurs, the IR bit in the SiRIC register is not changed to 1.

### 18.5.3 Special Mode 1 ($I^2C$ Mode)

- To generate a start condition, stop condition, or restart condition, set the STSPSEL bit in the UiSMR4 register (i = 0 to 6) to 0. Then, wait a half or more clock cycles of the transmit/receive clock to change the respective condition generate bit (the STAREQ, RSTAREQ, or STPREQ bit) from 0 to 1.

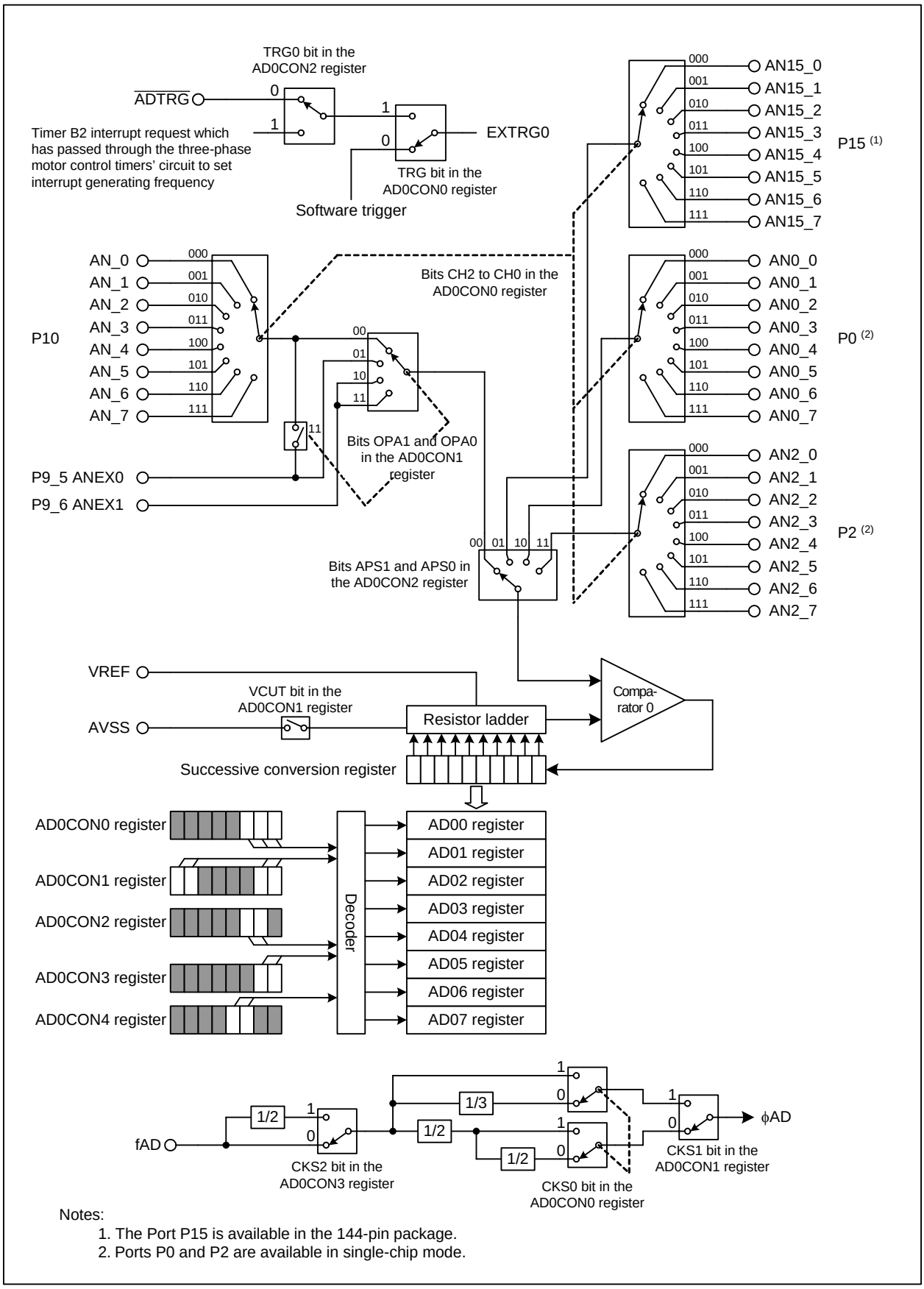
# 19. A/D Converter

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 in the A/D registers corresponding to selected pins. It is stored in the AD00 register only when DMAC operating mode is enabled.
When the A/D converter is not in use, power consumption can be reduced by setting the VCUT bit in the AD0CON1 register to 0 (VREF disconnected). This bit setting enables the power supply from VREF pin to the resistor ladder to stop.
Table 19.1 lists specifications of the A/D converter. Figure 19.1 shows a block diagram of the A/D converter. Figure 19.2 to Figure 19.7 show registers associated with the A/D converter.

**Table 19.1 A/D Converter Specifications**

| Item | Specification |
|---|---|
| A/D conversion method | Capacitance-based successive approximation |
| Analog input voltage (1) | 0 V to AVCC (VCC) |
| Operating clock, ϕAD (2) | fAD, fAD/2, fAD/3, fAD/4, fAD/6, or fAD/8 |
| Resolution | 8 bits or 10 bits |
| Operating modes | One-shot mode, repeat mode, single sweep mode, repeat sweet mode 0, repeat sweep mode 1, multi-port single sweep mode, multi-port repeat sweep mode 0 |
| Analog input pins (3) | 34 (4)<br>8 pins each for AN, AN0, AN2, and AN15 (5)<br>2 function-extended input pins (ANEX0 and ANEX1) |
| A/D conversion start conditions | • Software trigger<br>The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program<br>• External trigger (Retrigger is enabled)<br>An input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program<br>• Hardware trigger (Retrigger is enabled)<br>• Generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program |
| Conversion rates per pin | • Without sample and hold function<br>49 ϕAD cycles @ 8-bit resolution<br>59 ϕAD cycles @ 10-bit resolution<br>including 2 ϕAD cycles for sampling time<br>• With sample and hold function<br>28 ϕAD cycles @ 8-bit resolution<br>33 ϕAD cycles @ 10-bit resolution<br>including 3 ϕAD cycles for sampling time |

Notes:
1. Analog input voltage is not affected by with/without the sample and hold function.
2. The ϕAD frequency should be as follows:
   When VCC = 4.2 to 5.5 V, 16 MHz or below,
   When VCC = 3.0 to 4.2 V, 10 MHz or below
   Without the sample and hold function; 250 kHz or above
   With the sample and hold function; 1 MHz or above
3. When AVCC = VREF = VCC, A/D input voltage for pins AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, and ANEX1 should be VCC or lower.
4. Spec of the 144-pin package. In the 100-pin package, 26 channels are available.
5. Pins AN15_0 to AN15_7 are not available in the 100-pin package.




Notes:
1. The Port P15 is available in the 144-pin package.
2. Ports P0 and P2 are available in single-chip mode.

**Figure 19.1 A/D Converter Block Diagram**

## A/D0 Control Register 0 (1)

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| AD0CON0 | 0396h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CH0 | Analog Input Pin Select Bit (2, 3, 4) | b2 b1 b0<br>0 0 0 : ANi_0<br>0 0 1 : ANi_1<br>0 1 0 : ANi_2<br>0 1 1 : ANi_3<br>1 0 0 : ANi_4<br>1 0 1 : ANi_5<br>1 1 0 : ANi_6<br>1 1 1 : ANi_7 (i = none, 0, 2, 15) | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A/D Operating Mode Select Bit 0 (2, 5, 6) | b4 b3<br>0 0 : One-shot mode<br>0 1 : Repeat mode<br>1 0 : Single sweep mode<br>1 1 : Repeat sweep mode 0 or 1 | RW |
| MD1 | | | RW |
| TRG | Trigger Select Bit | 0: Software trigger<br>1: External trigger or hardware trigger (7) | RW |
| ADST | A/D Conversion Start Bit | 0: A/D conversion stopped<br>1: A/D conversion started (7) | RW |
| CKS0 | Frequency Select Bit | (Note 8) | RW |

Notes:

1. When this register is rewritten during an A/D conversion, the converted result is undefined.
2. Analog input pins should be set again after the A/D operating mode is changed.
3. This bit setting is enabled in one-shot mode or repeat mode.
4. Either AN, AN0, AN2, or AN15 port should be selected by using bits APS1 and APS0 in the AD0CON2 register.
5. When the MSS bit in the AD0CON3 register is set to 1 (multi-port sweep mode enabled), bits MD1 and MD0 should be set to 10b for multi-port single sweep mode and 11b for multi-port repeat sweep mode 0.
6. When the MSS bit in the AD0CON3 register is set to 1, bits MD1 and MD0 should be set to 10b or 11b.
7. To use the external trigger or the hardware trigger, a source of trigger should be selected using the TRG0 bit in the AD0CON2 register, the TRG bit should be set to 1, then the ADST bit should be set to 1.
8. The ϕAD frequency should be as follows: 16 MHz or below when VCC = 5V,
   10 MHz or below when VCC = 3.3 V
   It is selected from the combination of bits CKS0, CKS1, and CKS2 shown as below:

| The CKS2 bit in the AD0CON3 register | The CKS0 bit in the AD0CON0 register | The CKS1 bit in the AD0CON1 register | ϕAD |
|---|---|---|---|
| 0 | 0 | 0 | fAD divided by 4 |
| | | 1 | fAD divided by 3 |
| | 1 | 0 | fAD divided by 2 |
| | | 1 | fAD |
| 1 | 0 | 0 | fAD divided by 8 |
| | | 1 | fAD divided by 6 |

**Figure 19.2 AD0CON0 Register**

## A/D0 Control Register 1 (1)

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| AD0CON1 | 0397h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| SCAN0 | A/D Sweep Pin Select Bit (2, 3) | In single sweep mode or repeat sweep mode 0<br>b1 b0<br>0 0 : ANi_0, ANi_1<br>0 1 : ANi_0 to ANi_3<br>1 0 : ANi_0 to ANi_5<br>1 1 : ANi_0 to ANi_7<br>In repeat sweep mode 1 (4)<br>b1 b0<br>0 0 : ANi_0<br>0 1 : ANi_0, ANi_1<br>1 0 : ANi_0 to ANi_2<br>1 1 : ANi_0 to ANi_3<br>In multi-port single sweep mode or multi-port repeat sweep mode 0<br>b1 b0<br>1 1 : ANi_0 to ANi_7 (5)<br>(i = none, 0, 2, 15) | RW |
| SCAN1 | | | RW |
| MD2 | A/D Operating Mode Select Bit 1 | 0: Mode other than repeat sweep mode 1<br>1: Repeat sweep mode 1(6) | RW |
| BITS | 8/10-Bit Mode Select Bit | 0: 8-bit mode<br>1: 10-bit mode | RW |
| CKS1 | Frequency Select Bit | (Note 7) | RW |
| VCUT | VREF Connection Bit (8) | 0: VREF disconnected (9)<br>1: VREF connected (10) | RW |
| OPA0 | External Op-Amp Connect Mode Bit (11, 12) | b7 b6<br>0 0 : No use of ANEX0 or ANEX1 pin (Convert input at pins ANi_0 to ANi_7)<br>0 1 : Convert input at the ANEX0 pin<br>1 0 : Convert input at the ANEX1 pin<br>1 1 : External op-amp connected | RW |
| OPA1 | | | RW |

Notes:
1. When this register is rewritten during an A/D conversion, the converted result is undefined.
2. This bit setting is enabled in single sweep mode, repeat sweep mode 0, repeat sweep mode 1, multi-port single sweep mode, or multi-port repeat sweep mode 0.
3. Either AN, AN0, AN2, or AN15 port should be selected by using bits APS1 and APS0 in the AD0CON2 register.
4. These pins are commonly used in the AD conversion when the MD2 bit is set to 1.
5. Set bits SCAN0 and SCAN1 to 11b in multi-port single sweep mode or multi-port repeat sweep mode 0.
6. When the MSS bit in the AD0CON3 register is set to 1 (multi-port sweep mode enabled), the MD2 bit should be set to 0.
7. Refer to the note on the CKS0 bit in the AD0CON0 register.
8. This bit controls the reference voltage to the A/D converter. It does not affect the VREF performance of the D/A converter.
9. The VCUT bit should not be set to 0 during the A/D conversion.
10. When the VCUT bit is switched from 0 to 1, the A/D conversion should be started after 1 µs or more.
11. Bits OPA 0 and OPA1 can be set to 01b or 10b only in one-shot mode or repeat mode. They should be set to 00b or 11b in other modes.
12. When the MSS bit in the AD0CON3 register is set to 1 (multi-port sweep mode enabled), bits OPA0 and OPA 1 should be set to 00b.

**Figure 19.3 AD0CON1 Register**

## A/D0 Control Register 2 (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | | | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| AD0CON2 | 0394h | XX0X X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| SMP | A/D Conversion Method Select Bit | 0: Without sample & hold function<br>1: With sample & hold function | RW |
| APS0 | Analog Input Port Select Bit (2, 3, 4) | b2 b1<br>0 0 : AN_0 to AN_7, ANEX0, ANEX1<br>0 1 : AN15_0 to AN15_7<br>1 0 : AN0_0 to AN0_7<br>1 1 : AN2_0 to AN2_7 | RW |
| APS1 | | | RW |
| — (b4-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| TRG0 | External Trigger Request Source Select Bit | 0: Select $\overline{\text{ADTRG}}$ pin<br>1: Select a timer B2 interrupt request (after counting the ICTB2 register ) in the three-phase motor control timers | RW |
| — (b7-b6) | Reserved | Should be written with 0 and read as undefined value | RW |

Notes:
1. When this register is rewritten during an A/D conversion, the converted result is undefined.
2. Set bits APS1 and APS0 to 01b when the MSS bit in the AD0CON3 register is set to 1 (multi-port sweep mode enabled).
3. Do not set bits APS1 and APS0 to 01b In the 100-pin package when the MSS bit in the AD0CON3 register is set to 0 (multi-port sweep mode disabled).
4. These bits can be set to 10b or 11b in single-chip mode only.

**Figure 19.4 AD0CON2 Register**

## A/D0 Control Register 3 (1, 2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | | | | | |

Symbol: AD0CON3 | Address: 0395h | Reset Value: XXXX X000b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| DUS | DMAC Operating Mode Select Bit (3) | 0: DMAC operating mode disabled<br>1: DMAC operating mode enabled (4, 5) | RW |
| MSS | Multi-Port Sweep Mode Select Bit | 0: Multi-port sweep mode disabled<br>1: Multi-port sweep mode enabled (3, 6) | RW |
| CKS2 | Frequency Select Bit | (Note 7) | RW |
| MSF0 | Multi-Port Sweep Status Flag (8) | b4 b3<br>0 0 : AN_0 to AN_7<br>0 1 : AN15_0 to AN15_7<br>1 0 : AN0_0 to AN0_7<br>1 1 : AN2_0 to AN2_7 | RO |
| MSF1 | | | RO |
| — (b7-b5) | Reserved | Should be written with 0 and read as undefined value | RW |

Notes:
1. When this register is rewritten during A/D conversion, the converted result is undefined.
2. This register may be read incorrectly during A/D conversion. It should be read or written after the A/D converter stops operating.
3. To set the MSS bit to 1, the DUS bit should be also set to 1.
4. When the DUS bit is set to 1, all A/D conversion results are stored into the AD00 register.
5. To set the DUS bit to 1, configure the DMAC.
6. To set the MSS bit to 1, the following bit settings should be done:
   -the MD2 bit in the AD0CON1 register: 0 (mode other than repeat sweep mode 1)
   -bits APS1 and APS0 in the AD0CON2 register: 01b (AN15_0 to AN15_7)
   -bits OPA1 and OPA0 in the AD0CON1 register: 00b (no use of ANEX0 or ANEX1).
7. Refer to the note on the CKS0 bit in the AD0CON0 register.
8. This bit setting is enabled when the MSS bit is set to 1. The read value is undefined when the MSS bit is set to 0.

**Figure 19.5 AD0CON3 Register**

## A/D0 Control Register 4 (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | | | 0 | 0 |

| Symbol | Address | Reset Value |
|---|---|---|
| AD0CON4 | 0392h | XXXX 00XXb |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b1-b0) | Reserved | Should be written with 0 and read as undefined value | RW |
| MPS10 | Multi-port Sweep Port Select Bit (2, 3) | b3 b2<br>0 0 : (Note 4)<br>0 1 : AN_0 to AN_7, AN15_0 to AN15_7<br>1 0 : AN_0 to AN_7, AN0_0 to AN0_7<br>1 1 : AN_0 to AN_7, AN2_0 to AN2_7 | RW |
| MPS11 | | | RW |
| — (b7-b4) | Reserved | Should be written with 0 and read as undefined value | RW |

Notes:
1. When this register is rewritten during A/D conversion, the converted result is undefined.
2. Do not set bits MPS11 and MPS10 to 01b in the 100-pin package.
3. Bits MPS11 and MPS10 can be set to 10b or 11b in single-chip mode only.
4. When the MSS bit in the AD0CON3 register is set to 0 (multi-port sweep mode disabled), set bits MSP11 and MPS10 to 00b. When it is set to 1 (multi-port sweep mode enabled), set them to any value other than 00b.

**Figure 19.6 AD0CON4 Register**

## A/D0 Register i (i = 0 to 7) (1 to 4)

b15 b8 b7 b0

| Symbol | Address | Reset Value |
|---|---|---|
| AD00, AD01 | 0381h-0380h, 0383h-0382h | 0000 0000 XXXX XXXXb |
| AD02, AD03 | 0385h-0384h, 0387h-0386h | 0000 0000 XXXX XXXXb |
| AD04, AD05 | 0389h-0388h, 038Bh-038Ah | 0000 0000 XXXX XXXXb |
| AD06, AD07 | 038Dh-038Ch, 038Fh-038Eh | 0000 0000 XXXX XXXXb |

| Bit Symbol | Function | RW |
|---|---|---|
| — (b7-b0) | The lower byte in an A/D conversion result | RO |
| — (b9-b8) | In 10-bit mode: 2 upper bits in an A/D conversion result<br>In 8-bit mode: These bits are read as 0 | RO |
| — (b15-b10) | These bits are read as 0 | RO |

Notes:
1. In DMAC operating mode, register value read by a program is undefined.
2. Register value written while the A/D converter stops operating is undefined.
3. Only the AD00 register is available In DMAC operating mode. Other registers are undefined.
4. In DMAC operating mode at 10-bit mode, the DMAC should be set for a 16-bit transfer.

**Figure 19.7 Registers AD00 to AD07**

## 19.1 Mode Descriptions

### 19.1.1 One-shot Mode

In one-shot mode, the analog voltage applied to a selected pin is converted into a digital code only once. Table 19.2 lists specifications of one-shot mode.

**Table 19.2 One-shot Mode Specification**

| Item | Specification |
|---|---|
| Function | Converts only once the analog voltage applied to a pin into a digital code. The pin is selected using bits CH2 to CH0 in the AD0CON0 register, OPA1 and OPA0 in the AD0CON1 register, and APS1 and APS0 in the AD0CON2 register |
| Start conditions | In the TRG bit setting in the AD0CON0 register to 0 (software trigger)<br>• The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program<br>In the TRG bit setting to 1 (external trigger or hardware trigger)<br>• An input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program<br>• Generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program |
| Stop conditions | • An A/D conversion is completed (the ADST bit is set to 0 when the software trigger is selected)<br>• The ADST bit is set to 0 (A/D conversion stopped) by a program |
| Interrupt request generation timing | • When the A/D conversion is completed |
| Input pin to be selected | One pin is selected from among AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, and ANEX1 |
| Reading of A/D conversion result | In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled),<br>the AD0j register (j = 0 to 7) corresponding to the selected pin is read<br>In the DUS bit setting to 1 (DMAC operating mode enabled),<br>the AD00 register should not be read.<br>A/D conversion result is stored in the AD00 register after the conversion is completed. The DMAC transfers the conversion result to given memory space. Refer to 13. “DMAC” for DMAC settings |

### 19.1.2 Repeat Mode

In repeat mode, the analog voltage applied to a selected pin is repeatedly converted into a digital code. Table 19.3 lists specifications of repeat mode.

**Table 19.3 Repeat Mode Specification**

| Item | Specification |
|---|---|
| Function | Converts repeatedly the analog voltage input to a pin into a digital code. The pin is selected using bits CH2 to CH0 in the AD0CON0 register, OPA1 and OPA0 in the AD0CON1 register, and APS1 and APS0 in the AD0CON2 register |
| Start conditions | In the TRG bit setting in the AD0CON0 register to 0 (software trigger)<br>• The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program<br>In the TRG bit setting to 1 (external trigger or hardware trigger)<br>• An input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program<br>• Generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program |
| Stop conditions | • The ADST bit is set to 0 (A/D conversion stopped) by a program |
| Interrupt request generation timing | In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled),<br>• No interrupt request is generated<br>In the DUS bit setting to 1 (DMAC operating mode enabled),<br>• Every time an A/D conversion is completed |
| Analog voltage input pins | One pin is selected from among AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, and ANEX1 |
| Reading of A/D conversion result | In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled),<br>the AD0j register (j = 0 to 7) corresponding to the selected pin is read<br>In the DUS bit setting to 1 (DMAC operating mode enabled),<br>the AD00 register should not be read.<br>A/D conversion result is stored in the AD00 register after the conversion is completed. The DMAC transfers the conversion result to given memory space. Refer to 13. “DMAC” for DMAC settings |

### 19.1.3 Single Sweep Mode

In single sweep mode, the analog voltage applied to selected pins is converted one-by-one into a digital code. Table 19.4 lists specifications of single sweep mode.

**Table 19.4 Single sweep mode Specification**

| Item | Specification |
|---|---|
| Function | Converts one-by-one the analog voltage input to a set of pins into a digital code. The pins are selected using bits SCAN1 and SCAN0 in the AD0CON1 register and APS1 and APS0 in the AD0CON2 register |
| Start conditions | In the TRG bit setting in the AD0CON0 register to 0 (software trigger)<br>• The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program<br>In the TRG bit setting to 1 (external trigger or hardware trigger)<br>• An input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program<br>• Generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program |
| Stop conditions | • An A/D conversion is completed (the ADST bit is set to 0 when the software trigger is selected)<br>• The ADST bit is set to 0 (A/D conversion stopped) by a program |
| Interrupt request generation timing | In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled),<br>• When a sweep is completed<br>In the DUS bit setting to 1 (DMAC operating mode enabled),<br>• Every time an A/D conversion is completed |
| Analog voltage input pins | Selected from a group of 2 pins (ANi_0 and ANi_1) (i = none, 0, 2, 15), 4 pins (ANi_0 to ANi_3), 6 pins (ANi_0 to ANi_5), or 8 pins (ANi_0 to ANi_7) |
| Reading of A/D conversion result | In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled),<br>the AD0j register (j = 0 to 7) corresponding to the selected pin is read<br>In the DUS bit setting to 1 (DMAC operating mode enabled),<br>the AD00 register should not be read.<br>A/D conversion result is stored in the AD00 register after the conversion is completed. The DMAC transfers the conversion result to given memory space. Refer to 13. “DMAC” for DMAC settings |

### 19.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, the analog voltage applied to selected pins is repeatedly converted into a digital code. Table 19.5 lists specifications of repeat sweep mode 0.

**Table 19.5 Repeat Sweep Mode 0 Specification**

| Item | Specification |
|---|---|
| Function | Converts repeatedly the analog voltage input to a set of pins into a digital code. The pins are selected using bits SCAN1 and SCAN0 in the AD0CON1 register and APS1 and APS0 in the AD0CON2 register |
| Start conditions | In the TRG bit setting in the AD0CON0 register to 0 (software trigger)<br>• The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program<br>In the TRG bit setting to 1 (external trigger or hardware trigger)<br>• An input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program<br>• Generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program |
| Stop conditions | • The ADST bit is set to 0 (A/D conversion stopped) by a program |
| Interrupt request generation timing | In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled),<br>• No interrupt request is generated<br>In the DUS bit setting to 1 (DMAC operating mode enabled),<br>• Every time an A/D conversion is completed |
| Analog voltage input pins | Selected from a group of 2 pins (ANi_0 and ANi_1) (i = none, 0, 2, 15), 4 pins (ANi_0 to ANi_3), 6 pins (ANi_0 to ANi_5), or 8 pins (ANi_0 to ANi_7) |
| Reading of A/D conversion result | In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled),<br>the AD0j register (j = 0 to 7) corresponding to the selected pin is read<br>In the DUS bit setting to 1 (DMAC operating mode enabled),<br>the AD00 register should not be read.<br>A/D conversion result is stored in the AD00 register after the conversion is completed. The DMAC transfers the conversion result to given memory space. Refer to 13. “DMAC” for DMAC settings |

### 19.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, the analog voltage applied to eight selected pins including some prioritized pins is repeatedly converted into a digital code. Table 19.6 lists specifications of repeat sweep mode 1.

**Table 19.6 Repeat Sweep Mode 1 Specification**

| Item | Specification |
|---|---|
| Function | Converts repeatedly the analog voltage input to a set of eight pins into a digital code. A/some selected pin(s) is/are converted by priority<br>e.g. When AN_0 is prioritized, the analog voltage is converted into a digital code in the following order: AN_0→AN_1→AN_0→AN_2→AN_0→AN_3•••<br>The eight pins are selected using bits SCAN1 and SCAN0 in the AD0CON1 register and APS1 and APS0 in the AD0CON2 register |
| Start conditions | In the TRG bit setting in the AD0CON0 register to 0 (software trigger)<br>• The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program<br>In the TRG bit setting to 1 (external trigger or hardware trigger). Any external trigger generated during an A/D conversion (retrigger) is invalid.<br>• An input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program<br>• Generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program |
| Stop conditions | • The ADST bit is set to 0 (A/D conversion stopped) by a program |
| Interrupt request generation timing | In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled),<br>• No interrupt request is generated<br>In the DUS bit setting to 1 (DMAC operating mode enabled),<br>• Every time an A/D conversion is completed |
| Analog voltage input pins | 8 (ANi_0 to ANi_7) (i = none, 0, 2, 15) |
| Prioritized pin(s) | Selected from a group of 1 pin (ANi_0), 2 pins (ANi_0 and ANi_1), 3 pins (ANi_0 to ANi_2), or 4 pins (ANi_0 to ANi_3) |
| Reading of A/D conversion result | In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled),<br>the AD0j register (j = 0 to 7) corresponding to the selected pin is read<br>In the DUS bit setting to 1 (DMAC operating mode enabled),<br>the AD00 register should not be read.<br>A/D conversion result is stored in the AD00 register after the conversion is completed. The DMAC transfers the conversion result to given memory space. Refer to 13. “DMAC” for DMAC settings |

### 19.1.6 Multi-port Single Sweep Mode

In multi-port single sweep mode, the analog voltage applied to 16 selected pins is converted one-by-one into a digital code. The DUS bit in the AD0CON3 register should be set to 1 (DMAC operating mode enabled). Table 19.7 lists specifications of multi-port single sweep mode.

**Table 19.7 Multi-port Single Sweep Mode Specification**

| Item | Specification |
|---|---|
| Function | Converts one-by-one the analog voltage input to a set of 16 selected pins into a digital code in the following order: AN_0 to AN_7→ANi_0 to ANi_7 (i = 0, 2, 15)<br>The 16 pins are selected using bits MPS11 and MPS10 in the AD0CON4 register<br>e.g. When bits MPS11 and MPS10 are set to 10b (AN_0 to AN_7, AN0_0 to AN0_7),<br>the analog voltage is converted into a digital code in the following order: AN_0→AN_1→AN_2→AN_3→AN_4→AN_5→AN_6→AN_7→AN0_0→•••→AN0_6→AN0_7 |
| Start conditions | In the TRG bit setting in the AD0CON0 register to 0 (software trigger)<br>• The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program<br>In the TRG bit setting to 1 (external trigger or hardware trigger)<br>• An input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program<br>• Generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program |
| Stop conditions | • An A/D conversion is completed (the ADST bit is set to 0 when the software trigger is selected)<br>• The ADST bit is set to 0 (A/D conversion stopped) by a program |
| Interrupt request generation timing | • Every time an A/D conversion is completed (the DUS bit should be set to 1) |
| Analog voltage input pins | A combination of pin group is selected from AN_0 to AN_7→AN15_0 to AN15_7, AN_0 to AN_7→AN0_0 to AN0_7, or AN_0 to AN_7→AN2_0 to AN2_7 |
| Reading of A/D conversion result | The AD00 register should not be read.<br>The DUS bit should be set to 1.<br>A/D conversion result is stored in the AD00 register after the conversion is completed. The DMAC transfers the conversion result to given memory space. Refer to 13. “DMAC” for DMAC settings. |

### 19.1.7 Multi-port Repeat Sweep Mode 0

In multi-port repeat sweep mode 0, the analog voltage applied to 16 selected pins is repeatedly converted into a digital code. The DUS bit in the AD0CON3 register should be set to 1 (DMAC operating mode enabled). Table 19.8 lists specifications of multi-port repeat sweep mode 0.

**Table 19.8 Multi-port Repeat Sweep Mode 0 Specification**

| Item | Specification |
|---|---|
| Function | Converts repeatedly the analog voltage input to a set of 16 selected pins into a digital code in the following order: AN_0 to AN_7→ANi_0 to ANi_7 (i = 0, 2, 15)<br>The 16 pins are selected using bits MPS11 and MPS10 in the AD0CON4 register<br>e.g. When bits MPS11 and MPS10 are set to 10b (AN_0 to AN_7, AN0_0 to AN0_7),<br>the analog voltage is repeatedly converted into a digital code in the following order:<br>AN_0→AN_1→AN_2→AN_3→AN_4→AN_5→AN_6→AN_7→AN0_0→•••→AN0_6→AN0_7 |
| Start conditions | In the TRG bit setting in the AD0CON0 register to 0 (software trigger)<br>• The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program<br>In the TRG bit setting to 1 (external trigger or hardware trigger)<br>• An input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program<br>• Generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program |
| Stop conditions | • The ADST bit is set to 0 (A/D conversion stopped) by a program |
| Interrupt request generation timing | Every time an A/D conversion is completed (the DUS bit should be set to 1) |
| Analog voltage input pins | A combination of pin group is selected from AN_0 to AN_7→AN15_0 to AN15_7, AN_0 to AN_7→AN0_0 to AN0_7, or AN_0 to AN_7→AN2_0 to AN2_7 |
| Reading of A/D conversion result | The AD00 register should not be read.<br>The DUS bit should be set to 1.<br>A/D conversion result is stored in the AD00 register after the conversion is completed. The DMAC transfers the conversion result to given memory space. Refer to 13. “DMAC” for DMAC settings. |

## 19.2 Functions

### 19.2.1 Resolution Selection

The resolution is selected using the BITS bit in the AD0CON1 register. 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 AD0i register (i = 0 to 7). When the BITS bit is set to 0 (8-bit precision), the result is stored into bits 7 to 0 in the AD0i register.

### 19.2.2 Sample and Hold Function

This function improves the conversion rate per pin to 28 ϕAD cycles at 8-bit resolution and 33 ϕAD cycles for 10-bit resolution. To use this function, which is available in all operating modes, set the SMP bit in the AD0CON2 register to 1 (with sample and hold function). Start the A/D conversion after setting the SMP bit.

### 19.2.3 Trigger Selection

A trigger to start A/D conversion is specified by the combination of TRG bit in the AD0CON0 register and the TRG0 bit in the AD0CON2 register. Table 19.9 lists the settings of the trigger selection.

**Table 19.9 Trigger Selection Settings**

| Bit and Setting | | Trigger |
|---|---|---|
| AD0CON0 register | AD0CON2 register | |
| TRG = 0 | — | Software trigger<br>The ADST bit in the AD0CON0 register is set to 1 |
| TRG = 1 (1, 2) | TRG0 = 0 | External trigger<br>Falling edge of a signal applied to the $\overline{\text{ADTRG}}$ pin |
| | TRG0 = 1 | Hardware trigger<br>Generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers |

Notes:
1. The A/D converter starts operating if a trigger is generated while the ADST bit is set to 1 (A/D conversion started).
2. If an external trigger or a hardware trigger is generated during an A/D conversion, the A/D converter aborts the operation in progress. Then, it resumes the operation.

### 19.2.4 DMAC Operating Mode

The DMAC operating mode is available in 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 should be used definitely. 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. The DMAC transfers the data from the AD00 register to a given memory space every time an A/D conversion is completed at a pin. 8-bit DMA transfer should be selected for 8-bit resolution. For 10-bit resolution, 16-bit DMA transfer should be selected. Refer to 13. “DMAC” for details.

### 19.2.5 Function-extended Analog Input Pins

In one-shot mode and repeat mode, pins ANEX0 and ANEX1 are available as analog input by setting bits OPA1 and OPA0 in the AD0CON1 register (refer to Table 19.10). The A/D conversion result of pins ANEX0 and ANEX1 are respectively stored into registers AD00 and AD01. However, when the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode enabled), all results are stored into the AD00 register.

To use function-extended analog input pins, bits APS1 and APS0 in the AD0CON2 register should be set to 00b (AN0 to AN7, ANEX0, ANEX1 as analog input port) and the MSS bit in the AD0CON3 register to 0 (multi-port sweep mode disabled).

**Table 19.10 Function-extended Analog Input Pin Settings**

| AD0CON1 Register | | ANEX0 | ANEX1 |
|---|---|---|---|
| OPA1 | OPA0 | | |
| 0 | 0 | Not used | Not used |
| 0 | 1 | Analog input | Not used |
| 1 | 0 | Not used | Analog input |
| 1 | 1 | Output to an external op-amp | Input from an external op-amp |

### 19.2.6 External Operating Amplifier (Op-AMP) Connection Mode

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

When bits OPA1 and OPA0 in the AD0CON1 register are set to 11b (external op-amp connected), the voltage applied to pins AN0 to AN7 are output from the ANEX0 pin. This output signal should be amplified by an external op-amp and applied to the ANEX1 pin.

The analog voltage applied to the ANEX1 pin is converted into a digital code. The converted result is stored into the corresponding AD0i register (i = 0 to 7). The conversion rate varies with the response of the external op-amp. The ANEX0 pin should not be connected to the ANEX1 pin directly.

To use external op-amp connection mode, bits APS1 and APS0 in the AD0CON2 register should be set to 00b (AN0 to AN7, ANEX0, ANEX1 as analog input port).

Figure 19.8 shows an example of an external op-amp connection.

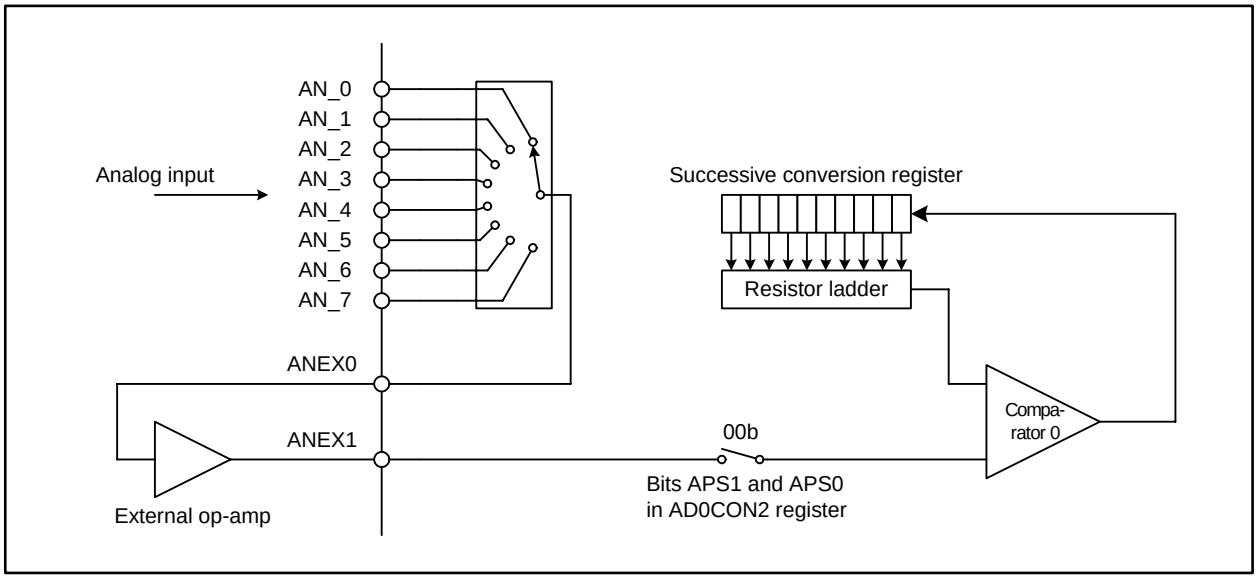



**Figure 19.8 External Op-Amp Connection**

### 19.2.7 Power Saving

When the A/D converter is not in use, power consumption can be reduced by setting the VCUT bit in the AD0CON1 to 0 (VREF disconnected). With this bit setting the reference voltage input pin (VREF) can be disconnected from the resistor ladder, which enables the power supply from the VREF to the resistor ladder to stop.

To use the A/D converter, the VCUT bit should be set to 1 (VREF connected) and 1 µs or more after, the ADST bit in the AD0CON0 register should be set to 1 (A/D conversion started). Bits ADST and VCUT should not be set to 1 simultaneously. The VCUT bit should not be set to 0 during the A/D conversion. The VCUT bit does not affect the VREF performance of the D/A converter (Refer to Figure 19.9).

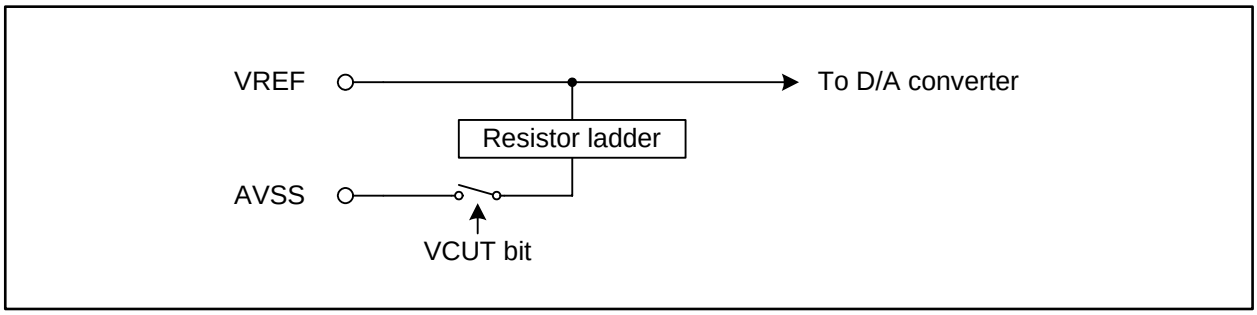



**Figure 19.9 Power Supply by VCUT Bit**

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

Figure 19.10 shows an analog input pin and external sensor equivalent circuit.

To perform A/D conversion correctly, internal capacitor (C) charging, shown in Figure 19.10, should be completed within the specified period. This period, called sampling time, is 2 ϕAD cycles for conversion without the sample and hold function and 3 ϕAD cycles for conversion with this function.

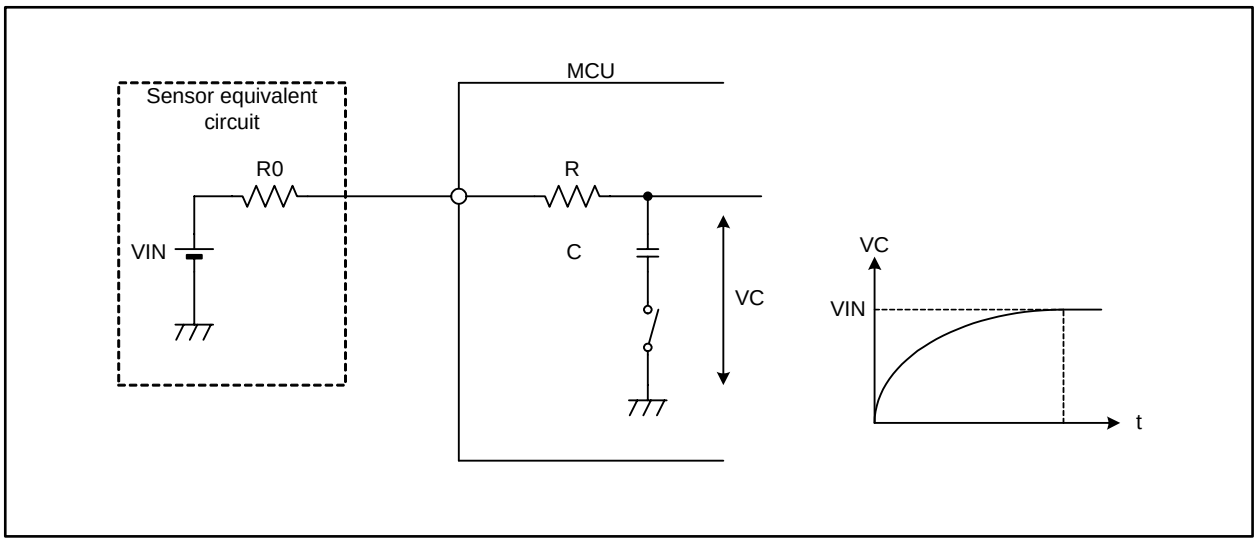



**Figure 19.10 Analog Input Pin and External Sensor Equivalent Circuitry**

The voltage between pins (VC) is expressed as follows:

$$VC = VIN\left\{1 - e^{-\frac{t}{C(R0+R)}}\right\}$$

When t = T and the precision (error) is x or less,

$$VC = VIN - \frac{x}{y}VIN = VIN\left(1 - \frac{x}{y}\right)$$

Thus, output impedance of the sensor equivalent circuit (R0) is determined by the following formulas:

$$e^{-\frac{T}{C(R0+R)}} = \frac{x}{y}$$

$$-\frac{T}{C(R0+R)} = \ln\frac{x}{y}$$

$$R0 = -\frac{T}{C\ln\frac{x}{y}} - R$$

where:
T[s] = Sampling time
R0[Ω] = Output Impedance of the sensor equivalent circuit
VC = Potential difference between edges of the capacitor C
R[Ω] = Internal resistance of the MCU
x[LSB] = Precision (error) of the A/D converter
y[step] = Resolution of the A/D converter (1024 steps @10-bit mode, 256 steps @ 8-bit mode)

When ϕAD = 10 MHz, the A/D conversion mode is 10-bit resolution with the sample and hold function, the output impedance (R0) with the precision (error) of 0.1 LSB or less is determined by the following formula:
Using T = 0.3 µs, R = 2.0 kΩ (reference value), C = 6.5 pF (reference value), x = 0.1, y = 1024,

$$R0 = -\frac{0.3\times10^{-6}}{6.5\times10^{-12}\times1\mathrm{n}\frac{0.1}{1024}} - 2.0\times10^{3}$$

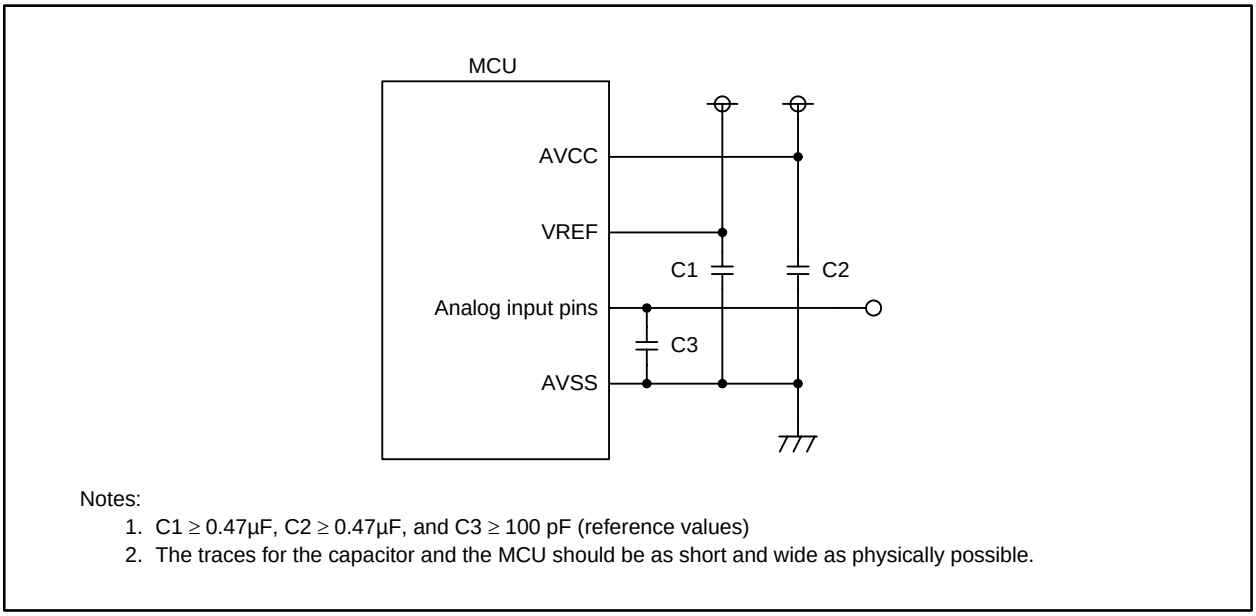
$$= 2998$$

Thus, the allowable output impedance of the sensor equivalent circuit (R0), making the precision (error) of 0.1 LSB or less, should be less than 3 kΩ.
Actual error, however, is the value of absolute precision added to 0.1 LSB mentioned above.

## 19.3 Notes on A/D Converter

### 19.3.1 Notes on Designing Boards

• Three capacitors should be respectively placed between the AVSS pin and such pins as AVCC, VREF, and analog inputs (AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, and AN15_0 to AN15_7) to avoid error operations caused by noise or latchup, and to reduce conversion errors. Figure 19.11 shows an example of pin configuration for A/D converter.




Notes:
1. C1 ≥ 0.47µF, C2 ≥ 0.47µF, and C3 ≥ 100 pF (reference values)
2. The traces for the capacitor and the MCU should be as short and wide as physically possible.

**Figure 19.11 Pin Configuration for A/D Converter**

• Do not use any of the four pins AN_4 to AN_7 for analog input if the key input interrupt is to be used. Otherwise, a key input interrupt request occurs when the A/D input voltage becomes VIL or lower.

• When AVCC = VREF = VCC, A/D input voltage for pins AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, and ANEX1 should be VCC or lower.

### 19.3.2 Notes on Programming

- The following registers should be written while the A/D conversion is stopped, that is, before a trigger occurs: AD0CON0 (except the ADST bit), AD0CON1, AD0CON2, AD0CON3, and AD0CON4.

- If the VCUT bit in the AD0CON1 register is switched from 0 (VREF connected) to 1 (VREF disconnected), the A/D conversion should be started after 1 µs or more. Set the VCUT bit to 0 when A/D conversion is not used to reduce power consumption.

- Set the port direction bit for the pin to be used as an analog input pin to 0 (input). Set the ASEL bit of the corresponding port function select register to 1 (the port is used as A/D input).

- If the TRG bit in the AD0CON0 register is set to 1 (external trigger or hardware trigger is selected), set the corresponding port direction bit (PD9_7 bit) for the ADTRG pin to 0 (input).

- The ϕAD frequency should be 16 MHz or below when VCC is 4.2 to 5.5 V, and 10 MHz or below when VCC is 3.0 to 4.2 V. It should be 1 MHz or above if the sample and hold function is enabled. If not, it should be 250 kHz or above.

- If A/D operating mode (bits MD1 and MD0 in the AD0CON0 register or the MD2 bit in the AD0CON1 register) has been changed, re-select analog input pins by using bits CH2 to CH0 in the AD0CON0 register or bits SCAN1 and SCAN0 in the AD0CON1 register.

- If the AD0i register (i = 0 to 7) is read when the A/D conversion result is stored to the register, the stored value may have an error. Read the AD0i register after the A/D conversion has been completed.
  In one-shot mode or single sweep mode, read the respective AD0i register after the IR bit in the AD0IC register has become 1 (interrupt requested).
  In repeat mode, repeat sweep mode 0, or repeat sweep mode 1, an interrupt request can be generated each time when an A/D conversion has been completed if the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode enabled). Similar to the other modes above, read the AD00 register after the IR bit in the AD0IC register has become 1 (interrupt requested).

- If the A/D conversion in progress is halted by setting the ADST bit in the AD0CON0 register to 0, the conversion result is undefined. In addition, the unconverted AD0i register may also become undefined. Consequently, the AD0i register should not be used just after A/D conversion is halted.

- The external trigger cannot be used in DMAC operating mode. Do not read the AD00 register by a program.

- If, in single sweep mode, the A/D conversion in progress is halted by setting the ADST bit in the AD0CON0 register to 0 (A/D conversion is stopped), an interrupt request may be generated even though the sweep is not completed. To halt the A/D conversion, first disable interrupts, then set the ADST bit to 0.

# 20. D/A Converter

The MCU has two separate 8-bit R-2R resistor ladder D/A converters.
Digital code is converted to an analog voltage when a value is written to the corresponding DAi register (i = 0,1). The DAiE bit in the DACON register determines whether the D/A conversion result is output or not. To output the converted value, the DAiE bit should be set to 1 (output enabled). This bit setting disables a pull-up resistor for the corresponding port.
Analog voltage to be output (V) is calculated based on the value (n) set in the DAi register (n = decimal).

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

*VREF*: reference voltage

Table 20.1 lists specifications of the D/A converter. Figure 20.1 shows a block diagram of the D/A converter. Figure 20.2 and Figure 20.3 show registers associated with the D/A. Figure 20.4 shows a D/A converter equivalent circuit.
When the D/A converter is not used, the DAi register should be set to 00h and the DAiE bit should be set to 0 (output disabled).

**Table 20.1 D/A Converter Specifications**

| Item | Specification |
|---|---|
| D/A conversion method | R-2R resistor ladder |
| Resolution | 8 bits |
| Analog output pins | 2 channels |

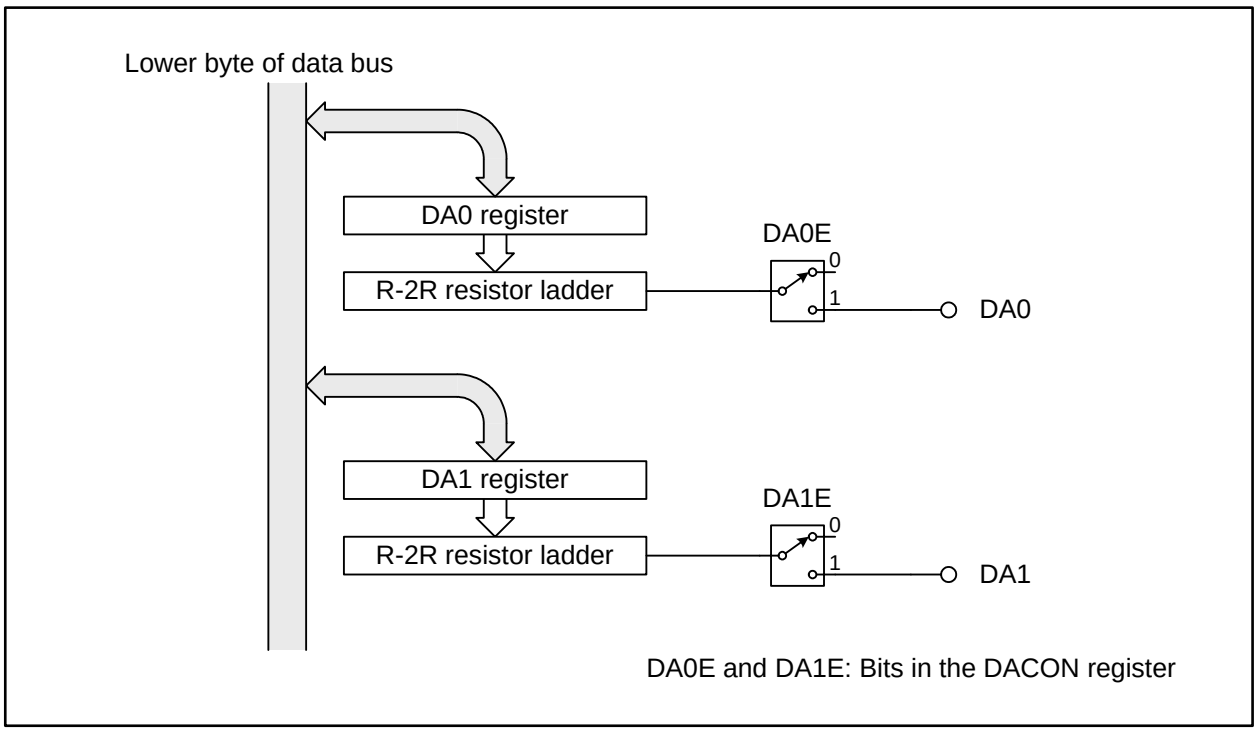



**Figure 20.1 D/A Converter Block Diagram**

D/A Control Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: DACON
Address: 039Ch
Reset Value: XXXX XX00b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| DA0E | D/A0 Output Enable Bit | 0: Output disabled<br>1: Output enabled | RW |
| DA1E | D/A1 Output Enable Bit | 0: Output disabled<br>1: Output enabled | RW |
| —<br>(b7-b2) | No register bits; should be written with 0 and read as undefined value | | — |

**Figure 20.2 DACON Register**

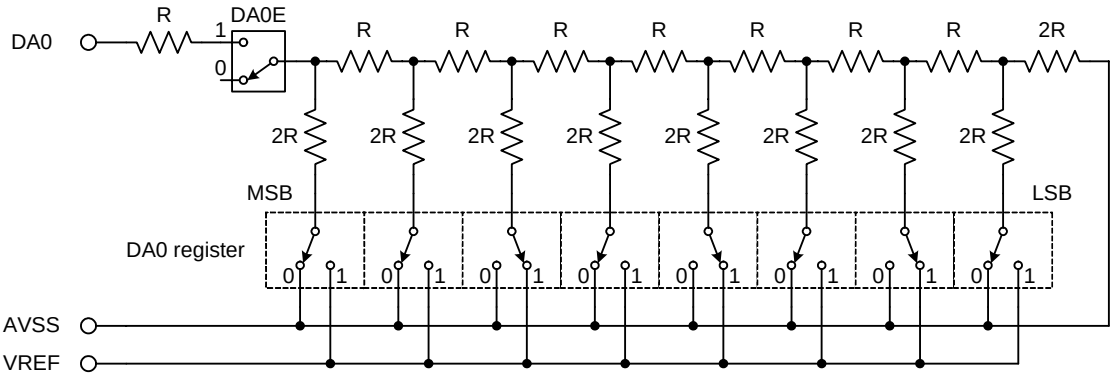
D/A Register i (i = 0, 1)

b7 b0

Symbol: DA0, DA1
Address: 0398h, 039Ah
Reset Value: Undefined

| Function | Setting Range | RW |
|---|---|---|
| Output value by the D/A conversion | 00h to FFh | RW |

**Figure 20.3 Registers DA0 and DA1**




Notes:
1. The figure above applies when the DA0 register is set to 2Ah.
2. This circuitry also applies to the D/A converter 1.
3. To reduce power consumption when the D/A converter is not in use, the DAiE bit (i = 0, 1) should be set to 0 (output disabled) and the DAi register should be set to 00h to prevent the current from flowing into the R-2R resistor ladder.

**Figure 20.4 D/A Converter Equivalent Circuitry**

# 21. CRC Calculator

The CRC (Cyclic Redundancy Check) calculator is used for error detection in data blocks. A generator polynomial of CRC_CCITT ($X^{16} + X^{12} + X^5 + 1$) generates a CRC.
The CRC is a 16-bit code generated for given blocks of 8-bit data. It is set in the CRCD register every time 1-byte data is written to the CRCIN register after a default value is set to the CRCD register.
Figure 21.1 shows a block diagram of the CRC calculator. Figure 21.2 and Figure 21.3 show registers associated with the CRC. Figure 21.4 shows an example of the CRC calculation.

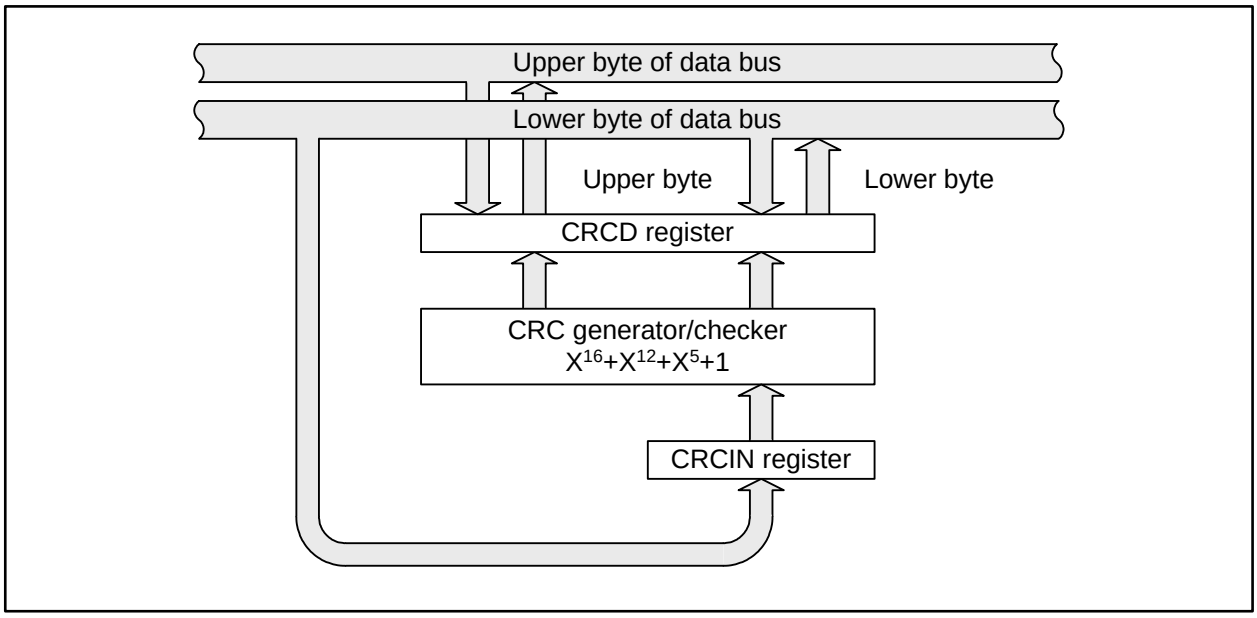



**Figure 21.1 CRC Calculator Block Diagram**

CRC Data Register

b15 b8 b7 b0

| Symbol | Address | Reset Value |
|---|---|---|
| CRCD | 037Dh-037Ch | Undefined |

| Function | Setting Range | RW |
|---|---|---|
| The CRC calculation result is stored in the CRCD register. When a default value in reversed bit position is set in this register and then a data in reversed bit position is written to the CRCIN register, the CRC in reversed bit position is read from this register | 0000h to FFFFh | RW |

**Figure 21.2 CRCD Register**

CRC Input Register

b7 b0

| Symbol | Address | Reset Value |
|---|---|---|
| CRCIN | 037Eh | Undefined |

| Function | Setting Range | RW |
|---|---|---|
| This register is to input data.<br>Data to be input should be in reversed bit position | 00h to FFh | RW |

**Figure 21.3 CRCIN Register**

## CRC Calculation and Setting Procedure to Generate CRC for 80C4h

- CRC Calculation for R32C

  CRC: a remainder of the division as follows: $\frac{\text{reversed-bit-position value in the CRCIN register}}{\text{generator polynominal}}$

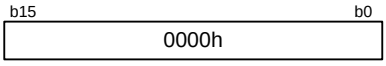
  Generator Polynomial: $X^{16}+X^{12}+X^5+1$(1 0001 0000 0010 0001b)

- Setting Steps

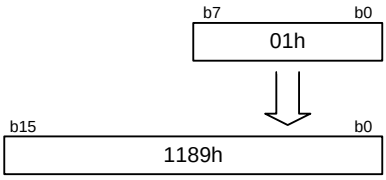
  (1) Reverse the bit position of 80C4h per byte by a program

  80h to 01h, C4h to 23h

  (2) Set 0000h (default value in reversed bit position) in CRCD register

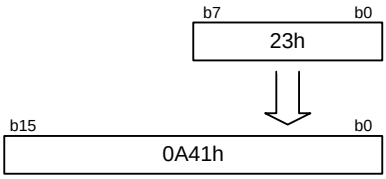
  
  

  (3) Set 01h (80h in reversed bit position) in CRCIN register

  1189h, CRC for 80h (9188h) in reversed bit position is stored into the CRCD register in the third cycle.

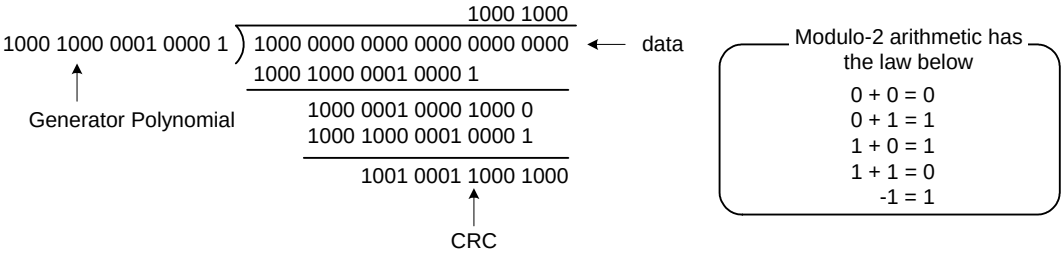
  (4) Set 23h (C4h in reversed bit position) in CRCIN register

  0A41h, CRC for 80C4h (8250h) in revered bit position is stored into the CRCD register in the third cycle.

- Details of CRC Calculation

As shown in (3) above, add 1000 0000 0000 0000 0000 0000b as 80h (1000 0000b) plus 16 digits to 0000 0000 0000 0000 0000 0000b as the default value of the CRCD register, 0000h plus eight digits to perform the modulo-2 division.




0001 0001 1000 1001b (1189h), the reversed-bit-position value of remainder 1001 0001 1000 1000b (9188h) can be read from the CRCD register.

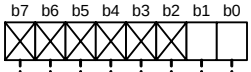
When going on (4) above, add 1100 0100 0000 0000 0000 0000b as C4h (1100 0100b) plus 16 digits to 1001 0001 1000 1000 0000 0000b as the remainder of (3) left in the CRCD register plus eight digits to perform the modulo-2 division.
0000 1010 0100 0001b (0A41h), the reversed-bit-position value of remainder 1000 0010 0101 0000b (8250h) can be read from the CRCD register.

**Figure 21.4 CRC Calculation**

# 22. X-Y Conversion

The X-Y conversion rotates a 16 × 16-bit matrix data 90 degrees or reverses the bit position of 16-bit data. The X-Y conversion is set using the XYC register shown in Figure 22.1.
Data is written in write-only XiR registers (i = 0 to 15) and converted data is read in read-only YjR register (j = 0 to 15). These registers are allocated to the same address. Figure 22.2 and Figure 22.3 show registers XiR and YjR, respectively. A write/read access from an even address to the XiR/YjR registers should be performed every 16 bits. 8-bit access operation results are undefined.

X-Y Control Register

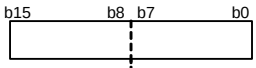



| Symbol | Address | Reset Value |
|---|---|---|
| XYC | 02E0h | XXXX XX00b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| XYC0 | Read Mode Set Bit | 0: Data rotation<br>1: No data rotation | RW |
| XYC1 | Write Mode Set Bit | 0: No bit position reverse<br>1: Bit position reverse | RW |
| —<br>(b7-b2) | No register bits; should be written with 0 and read as undefined value | | — |

**Figure 22.1 XYC Register**

Xi Register (i = 0 to 15) (1)



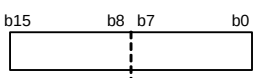

| Symbol | Address | Reset Value |
|---|---|---|
| X0R to X2R | 02C1h to 02C0h, 02C3h to 02C2h, 02C5h to 02C4h | Undefined |
| X3R to X5R | 02C7h to 02C6h, 02C9h to 02C8h, 02CBh to 02CAh | Undefined |
| X6R to X8R | 02CDh to 02CCh, 02CFh to 02CEh, 02D1h to 02D0h | Undefined |
| X9R to X11R | 02D3h to 02D2h, 02D5h to 02D4h, 02D7h to 02D6h | Undefined |
| X12Rto X14R | 02D9h to 02D8h, 02DBh to 02DAh, 02DDh to 02DCh | Undefined |
| X15R | 02DFh to 02DEh | Undefined |

| Function | Setting Range | RW |
|---|---|---|
| Input data for X-Y conversion | 0000h to FFFFh | WO |

Note:
1. A 16-bit write access to this register should be performed.

**Figure 22.2 XiR Register**

Yj Register (j = 0 to 15) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| Y0R to Y2R | 02C1h to 02C0h, 02C3h to 02C2h, 02C5h to 02C4h | Undefined |
| Y3R to Y5R | 02C7h to 02C6h, 02C9h to 02C8h, 02CBh to 02CAh | Undefined |
| Y6R to Y8R | 02CDh to 02CCh, 02CFh to 02CEh, 02D1h to 02D0h | Undefined |
| Y9R to Y11R | 02D3h to 02D2h, 02D5h to 02D4h, 02D7h to 02D6h | Undefined |
| Y12R to Y14R | 02D9h to 02D8h, 02DBh to 02DAh, 02DDh to 02DCh | Undefined |
| Y15R | 02DFh to 02DEh | Undefined |

| Function | RW |
|---|---|
| Result of X-Y conversion | RO |

Note:
1. A 16-bit read access to this register should be performed.

**Figure 22.3 YjR Register**

## 22.1 Data Conversion on Reading

The XYC0 bit in the XYC register selects a read mode for the YjR register. When the XYC0 bit is set to 0 (data rotation), bit j in the corresponding registers X0R to X15R is automatically read on reading the YjR register (j = 0 to 15).

More concretely, on reading bit i (i = 0 to 15) in the Y0R register, the data of each bit 0 in the XiR register is read. That is, a read data of bit 0 in the Y15R register means the data of bit 15 in the X0R register and the data of bit 15 in the Y0R register is identical to that of bit 0 in the X15R register.

Figure 22.4 shows the conversion table when the XYC0 bit is set to 0 and Figure 22.5 shows an example of X-Y conversion.

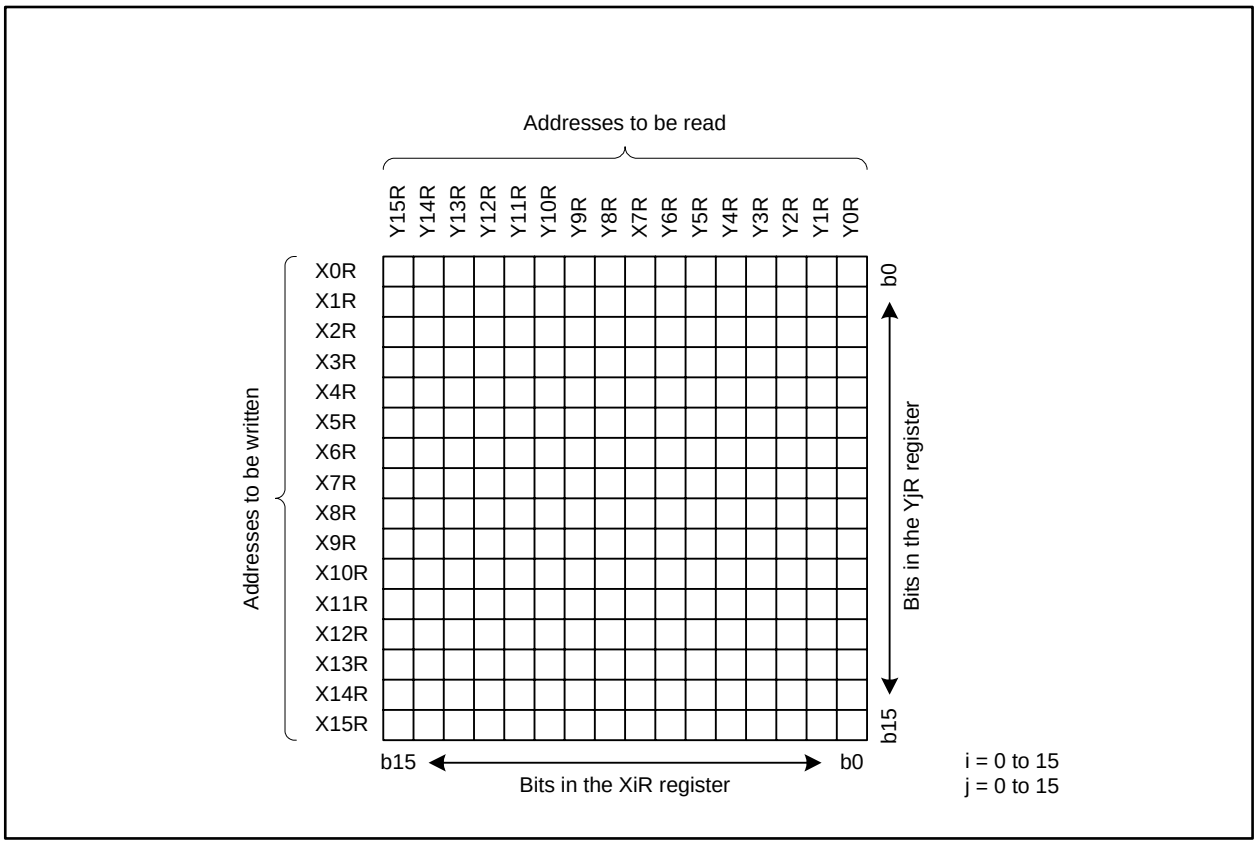



**Figure 22.4 Conversion Table (XYC0 Bit = 0)**

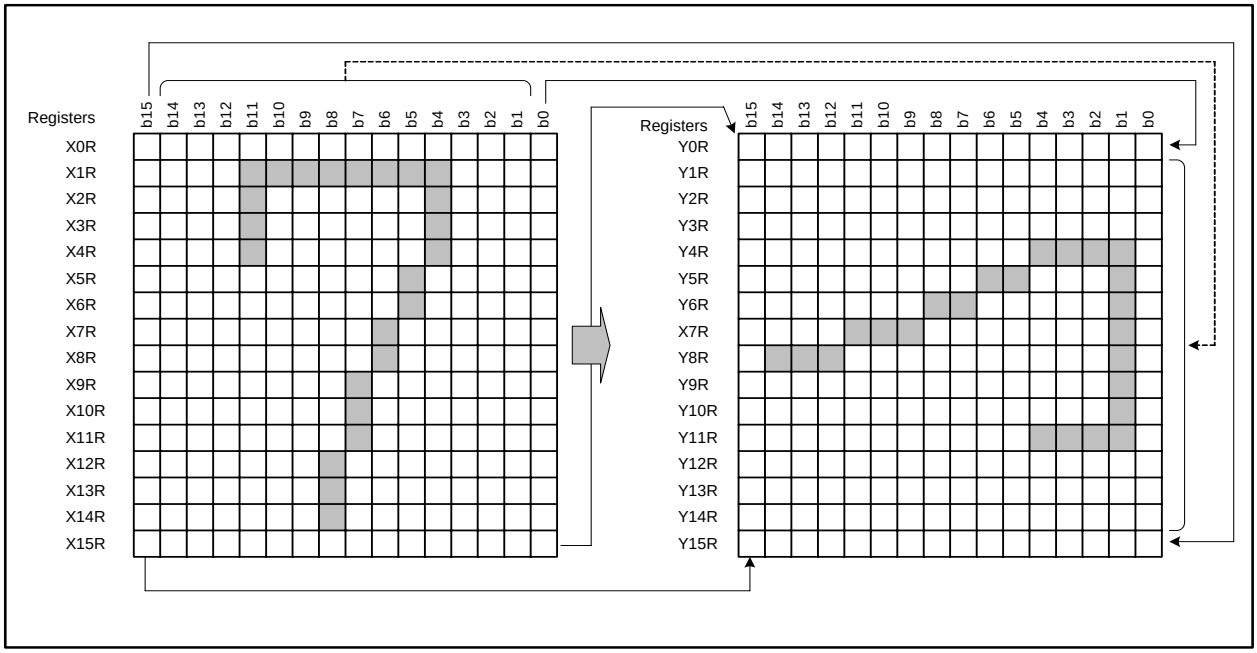



**Figure 22.5 X-Y Conversion**

When the XYC0 bit is set to 1 (no data rotation), the data of each bit in the YjR register is identical to that written in the XiR register. Figure 22.6 shows the conversion table when the XYC0 bit is set to 1.

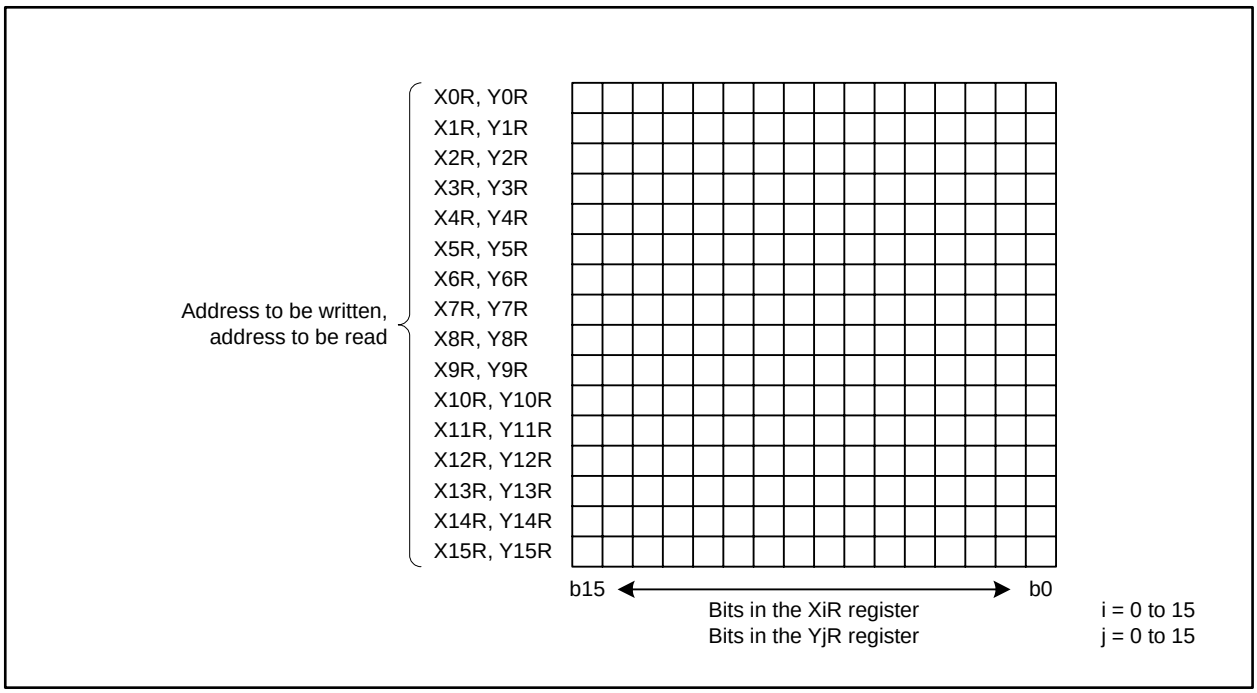



**Figure 22.6 Conversion Table (XYC0 Bit = 1)**

## 22.2 Data Conversion on Writing

The XYC1 bit in the XYC register selects a write mode for the XiR register.

When the XYC1 bit is set to 0 (no bit position reverse), the data is written in order. When it is set to 1 (bit position reverse), the data is written in reversed order. Figure 22.7 shows the conversion table when the XYC1 bit is set to 1.

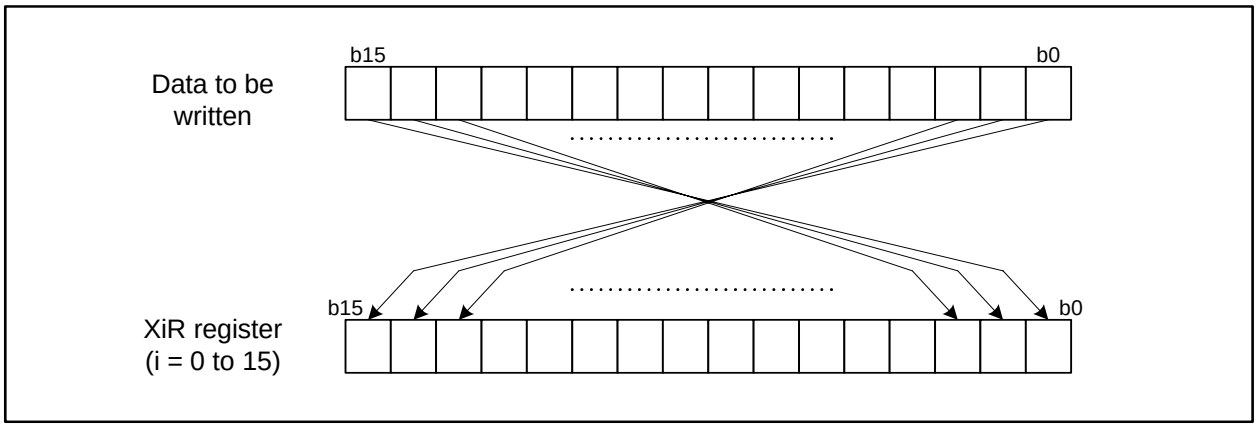



**Figure 22.7 Conversion Table (XYC1 Bit = 1)**

# 23. Intelligent I/O

The intelligent I/O is a multifunctional I/O port for time measurement, waveform generation, variable character length synchronous serial interface, and IEBus.
It consists of three groups each of which has one free-running 16-bit base timer and eight 16-bit registers for time measurement or waveform generation.
Table 23.1 lists functions and channels of the intelligent I/O.

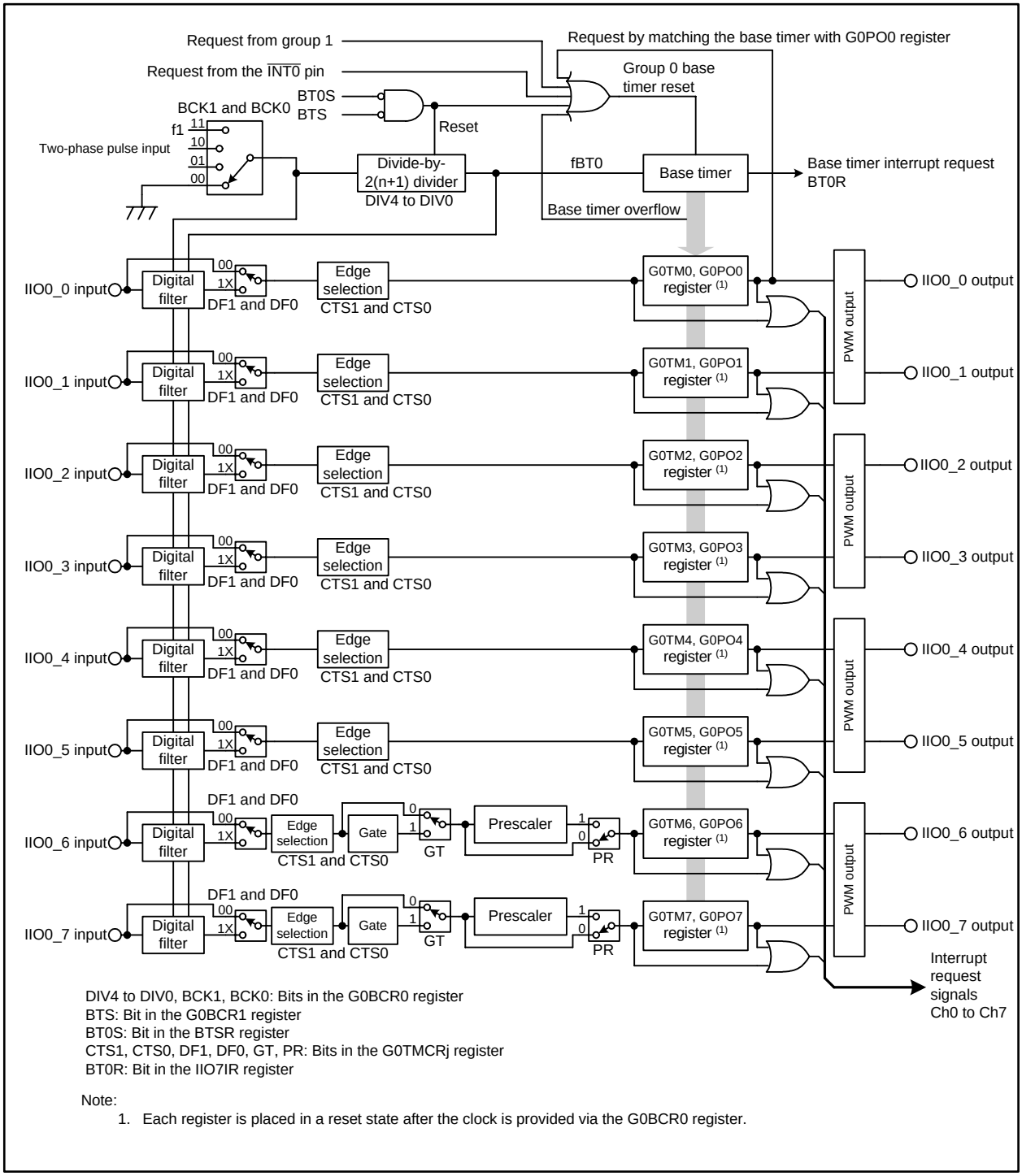
**Table 23.1 Intelligent I/O Functions and Channels**

| Functions | | Group 0 | Group 1 | Group 2 |
|---|---|---|---|---|
| Time measurement (1) | Digital filter | 8 channels | 8 channels | Not available |
| | Prescaler | 2 channels | 2 channels | |
| | Gating | 2 channels | 2 channels | |
| Waveform generation (1) | Single-phase waveform output mode | 8 channels | 8 channels | 8 channels |
| | Inverted waveform output mode | 8 channels | 8 channels | 8 channels |
| | SR waveform output mode | 8 channels | 8 channels | 8 channels |
| | Bit modulation PWM mode | Not available | Not available | 8 channels |
| | RTP mode | | | 8 channels |
| | Parallel RTP mode | | | 8 channels |
| Serial interface | Variable character length synchronous serial interface mode | Not available | Not available | Available |
| | IEBus mode (optional (2)) | | | |

Note:
1. Functions time measurement and waveform generation share a pin.
2. Please contact a Renesas sales office to use the optional feature.

Each channel individually selects a function from the time measurement and the waveform generation.

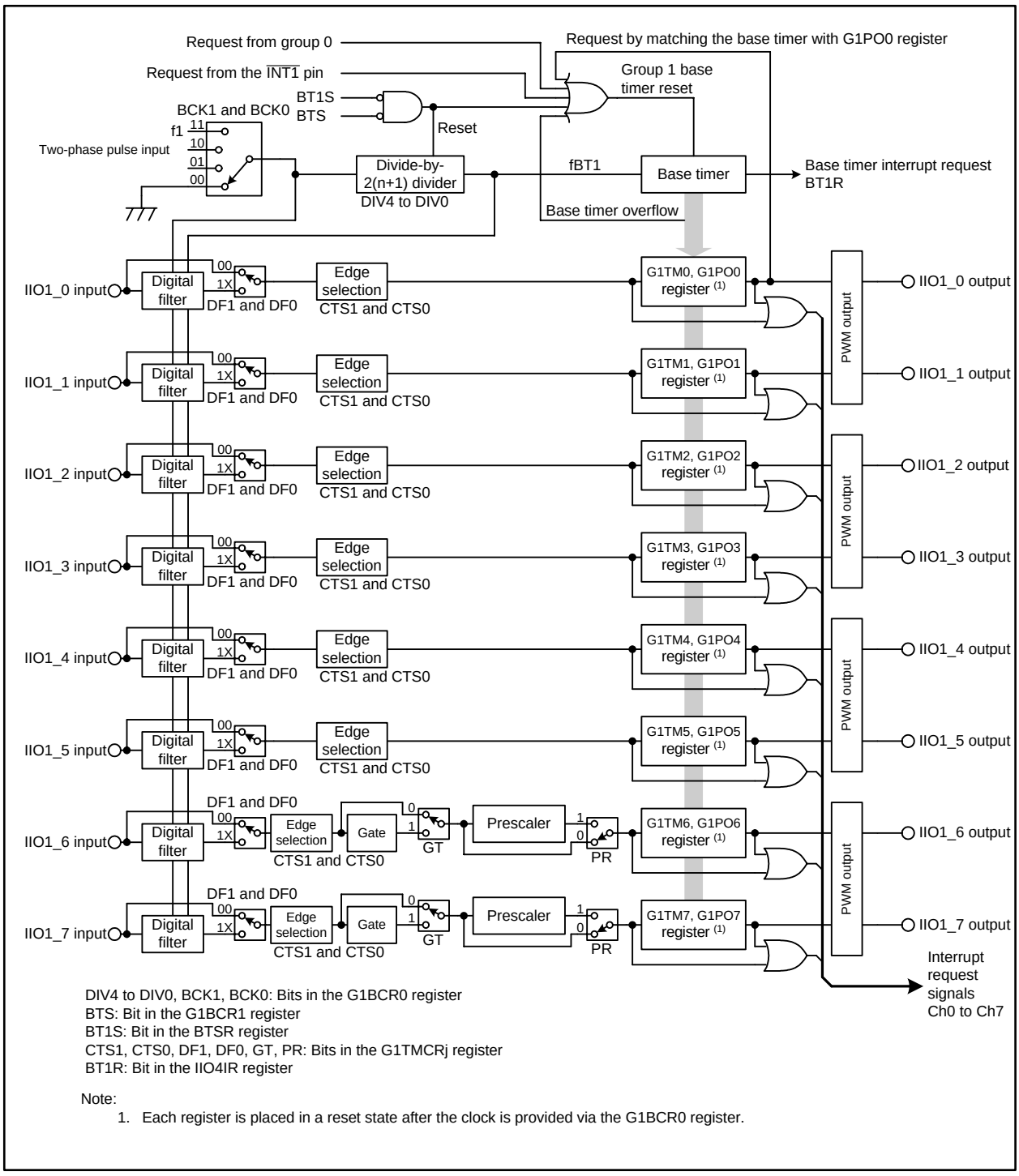
Figure 23.1 to Figure 23.3 show block diagrams of the intelligent I/O.




DIV4 to DIV0, BCK1, BCK0: Bits in the G0BCR0 register
BTS: Bit in the G0BCR1 register
BT0S: Bit in the BTSR register
CTS1, CTS0, DF1, DF0, GT, PR: Bits in the G0TMCRj register
BT0R: Bit in the IIO7IR register

Note:
1. Each register is placed in a reset state after the clock is provided via the G0BCR0 register.

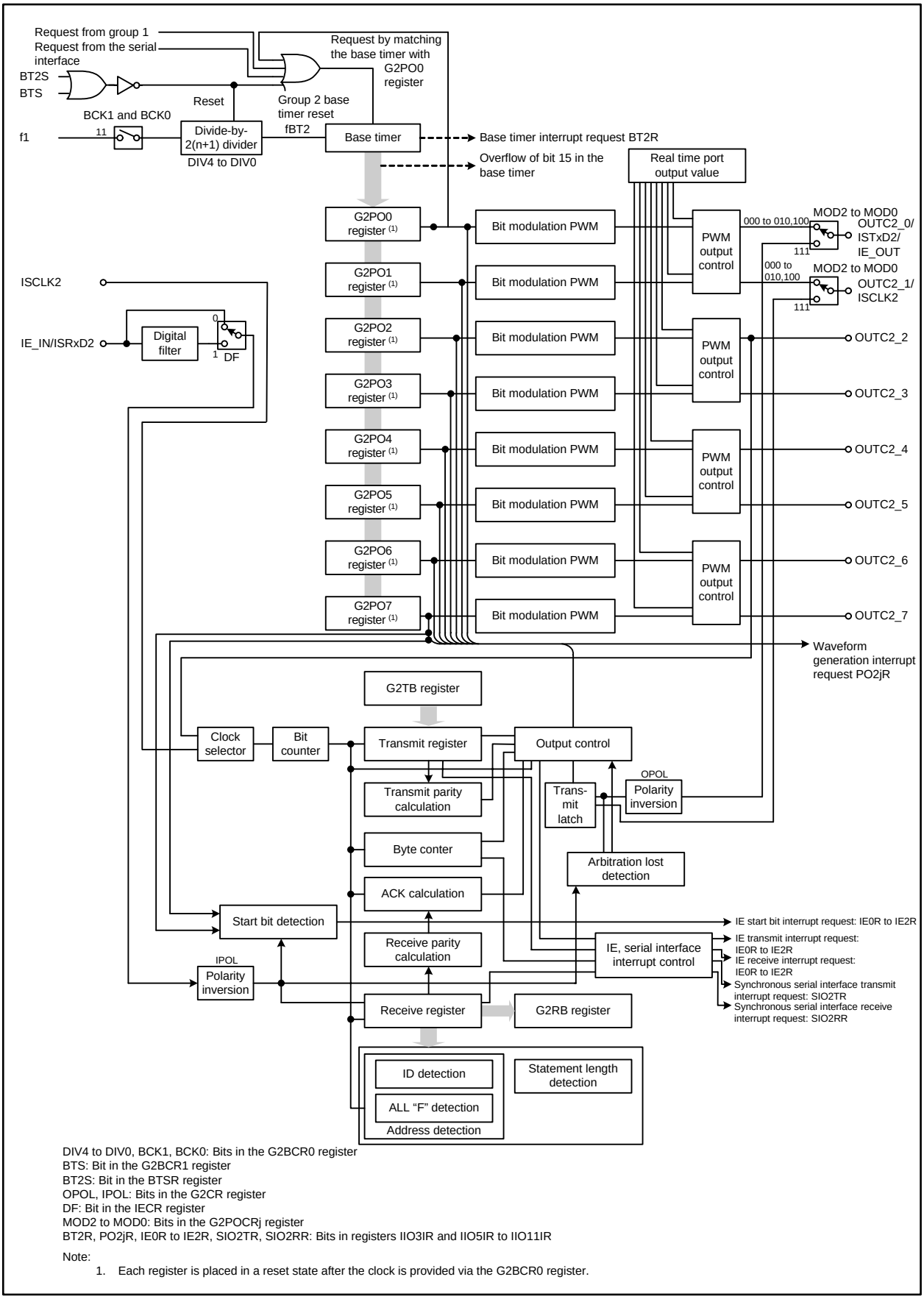
**Figure 23.1 Intelligent I/O Group 0 Block Diagram (j = 0 to 7)**




DIV4 to DIV0, BCK1, BCK0: Bits in the G1BCR0 register
BTS: Bit in the G1BCR1 register
BT1S: Bit in the BTSR register
CTS1, CTS0, DF1, DF0, GT, PR: Bits in the G1TMCRj register
BT1R: Bit in the IIO4IR register

Note:
1. Each register is placed in a reset state after the clock is provided via the G1BCR0 register.

**Figure 23.2 Intelligent I/O Group 1 Block Diagram (j = 0 to 7)**



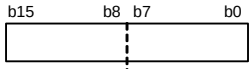

DIV4 to DIV0, BCK1, BCK0: Bits in the G2BCR0 register
BTS: Bit in the G2BCR1 register
BT2S: Bit in the BTSR register
OPOL, IPOL: Bits in the G2CR register
DF: Bit in the IECR register
MOD2 to MOD0: Bits in the G2POCRj register
BT2R, PO2jR, IE0R to IE2R, SIO2TR, SIO2RR: Bits in registers IIO3IR and IIO5IR to IIO11IR

Note:
1. Each register is placed in a reset state after the clock is provided via the G2BCR0 register.

**Figure 23.3 Intelligent I/O Group 2 Block Diagram (j = 0 to 7)**

Figure 23.4 to Figure 23.17 show registers associated with the intelligent I/O base timer, the time measurement, and the waveform generation (For registers associated with the serial interface, refer to Figure 23.33 to Figure 23.40).

Group i Base Timer Register (i = 0 to 2) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| G0BT, G1BT, G2BT | 01A1h-01A0h, 0121h-0120h, 0161h-0160h | Undefined Undefined |

| Function | Setting Range | RW |
|---|---|---|
| - While the base timer is running, this register indicates the value of base timer; when a value is written, the counter immediately starts counting from this value. The register is set to 0000h when the base timer is reset.<br>- While the base timer is being reset, this register is set to 0000h; the register is read as undefined value; no value can be set. (2) | 0000h to FFFFh | RW |

Notes:
1. The GiBT register reflects the base timer value after a delay of a half fBTi cycle.
2. The base timer stops only when bits BCK1 and BCK0 in the GiBCR0 register are set to 00b (clock stopped). However, if the BTiS bit in the BTSR register and the BTS bit in the GiBCR1 register are set to 0, the base timer is held in a reset state, holding the value 0000h. That is, the base timer is in a “no counting” state. If either of these bits is set to 1, this state is cleared and the base timer starts counting.

**Figure 23.4 Registers G0BT to G2BT**

## Group i Base Timer Control Register 0 (i = 0 to 2)

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| G0BCR0 to G2BCR0 | 01A2h, 0122h, 0162h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| BCK0 | Count Source Select Bit | b1 b2<br>0 0 : Clock stopped<br>0 1 : Do not use this combination<br>1 0 : Two-phase pulse signal input (1)<br>1 1 : f1 | RW |
| BCK1 | | | RW |
| DIV0 | Count Source Divide Ratio Select Bit | Divide the count source by 2(n+1). The count source is not divided when n = 31. (n = 0 to 31)<br>b6 b5 b4 b3 b2<br>0 0 0 0 0 : divide-by-2 (n = 0)<br>0 0 0 0 1 : divide-by-4 (n = 1)<br>0 0 0 1 0 : divide-by-6 (n = 2)<br>:<br>1 1 1 1 0 : divide-by-62 (n = 30)<br>1 1 1 1 1 : no division (n = 31) | RW |
| DIV1 | | | RW |
| DIV2 | | | RW |
| DIV3 | | | RW |
| DIV4 | | | RW |
| IT | Base Timer Interrupt Source Select Bit | 0: Overflow of bit 15 or the bit 9<br>1: Overflow of bit 14 | RW |

Note:
1. This bit setting is enabled only when bits UD1 and UD0 in the GjBCR1 register (j = 0, 1) are set to 10b (two-phase pulse signal processing mode). Bits BCK1 and BCK0 should not be set to 10b in other modes or in group 2.

**Figure 23.5 Registers G0BCR0 to G2BCR0**

## Group i Base Timer Control Register 1 (i = 0, 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | | | | 0 | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| G0BCR1, G1BCR1 | 01A3h, 0123h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| RST0 | Base Timer Reset Source Select Bit 0 | 0: No reset<br>1: Synchronization with another base timer reset (1) | RW |
| RST1 | Base Timer Reset Source Select Bit 1 | 0: No reset<br>1: Match with the GiPO0 register (2) | RW |
| RST2 | Base Timer Reset Source Select Bit 2 | 0: No reset<br>1: An low signal input into the $\overline{INTi}$ pin (3) | RW |
| — (b3) | Reserved | Should be written with 0 | RW |
| BTS | Base Timer Start Bit (4, 5) | 0: Reset the base timer<br>1: Start counting | RW |
| UD0 | Increment/Decrement Counting Control Bit | b6 b5<br>0 0 : Increment counting mode<br>0 1 : Increment/decrement counting mode<br>1 0 : Two-phase pulse signal processing mode (6)<br>1 1 : Do no use this combination | RW |
| UD1 | | | RW |
| — (b7) | Reserved | Should be written with 0 | RW |

Notes:
1. The group 0 base timer is reset by synchronizing with the reset of group 1 base timer, and vice versa.
2. The base timer is reset after two fBTi clock cycles if the base timer value has matched the GiPO0 register setting (Refer to Figure 23.14 for the details on the GiPO0 register). When the RST1 bit is set to 1, the value of GiPOj register (j = 1 to 7) to be used for the waveform generation should be smaller than that of the GiPO0 register.
3. The group 0 base timer is reset by an input of low signal into the $\overline{INT0}$ pin. The $\overline{INT1}$ pin is, accordingly, for the group 1 base timer reset.
4. To start base timers groups 0 and 1 individually, the BTS bit should be set to 1 after setting the BTkS bit (k = 0, 1) in the BTSR register to 0 (base timer is reset).
5. To start the base timers of two or all groups simultaneously, the BTSR register should be used. The BTS bit should be set to 0.
6. In two-phase pulse signal processing mode, the base timer is not reset, even though the RST1 bit is set to 1, if the timer counter decrements the count after two clock cycles when the base timer value matches the GiPO0 register.

**Figure 23.6 Registers G0BCR1 and G1BCR1**

## Group 2 Base Timer Control Register 1

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | 0 | 0 | | 0 | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| G2BCR1 | 0163h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| RST0 | Base Timer Reset Source Select Bit 0 | 0: No reset<br>1: Synchronization with group1 base timer reset | RW |
| RST1 | Base Timer Reset Source Select Bit 1 | 0: No reset<br>1: Match with the G2PO0 register (1) | RW |
| RST2 | Base Timer Reset Source Select Bit 2 | 0: No reset<br>1: A reset request from the serial interface | RW |
| — (b3) | Reserved | Should be written with 0 | RW |
| BTS | Base Timer Start Bit (2, 3) | 0: Reset the base timer<br>1: Start counting | RW |
| — (b6-b5) | Reserved | Should be written with 0 | RW |
| PRP | Parallel Real-Time Port Select Bit (4) | 0: RTP output mode<br>1: Parallel RTP output mode | RW |

Notes:
1. The base timer is reset after two fBT2 clock cycles if the base timer value matches the G2PO0 register setting. When the RST1 bit is set to 1, the value of G2POj register (j = 1 to 7) used for the waveform generation or the serial interface should be smaller than that of the G2PO0 register.
2. To start the group 2 base timer, the BTS bit should be set to 1 after setting the BT2S bit in the BTSR register to 0 (base timer is reset).
3. To start the base timers of two or all groups simultaneously, the BTSR register should be used. The BTS bit should be set to 0.
4. This bit setting is enabled when the RTP bit in the G2POCRi register is set to 1 (real-time port used).

**Figure 23.7 G2BCR1 Register**

## Base Timer Start Register (1, 2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | 0 | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| BTSR | 0164h | XXXX 0000b |

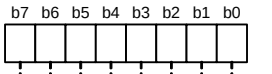
| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| BT0S | Group 0 Base Timer Start Bit | 0: Reset the base timer<br>1: Start counting | RW |
| BT1S | Group 1 Base Timer Start Bit | 0: Reset the base timer<br>1: Start counting | RW |
| BT2S | Group 2 Base Timer Start Bit | 0: Reset the base timer<br>1: Start counting | RW |
| —<br>(b3) | Reserved | Should be written with 0 | RW |
| —<br>(b7-b4) | No register bits; should be written with 0 and read as undefined value. | | — |

Notes:

1. The initial settings of bits and registers for the intelligent I/O are required as follows:
   (1) Set the G2BCR0 register to provide the clock to the group 2 base timer
   (2) Set all bits BT0S to BT3S to 0 (base timer is reset)
   (3) Set other registers associated with the intelligent I/O
   The BTiS bit (i = 0 to 2) allows the base timers of two or all groups to start counting simultaneously. To start counting individually, the BTiS bit should be set to 0 and the BTS bit in the GiBCR1 register should be used.
2. The following procedure is required to start the base timers of more than two groups simultaneously:
   -Bits BCK1 to BCK0 and DIV4 to DIV0 in the GiBCR0 registers (i = two among 0, 1, and 2 or all) to be used should be set identically
   -After bits BCK1 to BCK0 or DIV4 to DIV0 are changed, start the base timers twice as following procedure:
   (1) Set the BTiS bit to 1 (base timer starts counting)
   (2) Set the BTiS bit to 0 (base timer is reset) after one fBTi clock cycle or more
   (3) Set the BTiS bit to 1 again after another one fBTi clock cycle or more

**Figure 23.8 BTSR Register**

## Group i Time Measurement Control Register j (i = 0, 1; j = 0 to 7)



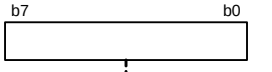

| Symbol | Address | Reset Value |
|---|---|---|
| G0TMCR0 to G0TMCR3 | 0198h, 0199h, 019Ah, 019Bh | 0000 0000b |
| G0TMCR4 to G0TMCR7 | 019Ch, 019Dh, 019Eh, 019Fh | 0000 0000b |
| G1TMCR0 to G1TMCR3 | 0118h, 0119h, 011Ah, 011Bh | 0000 0000b |
| G1TMCR4 to G1TMCR7 | 011Ch, 011Dh, 011Eh, 011Fh | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CTS0 | Time Measurement Trigger Select Bit | b1 b0<br>0 0 : No time measurement<br>0 1 : Rising edge<br>1 0 : Falling edge<br>1 1 : Both edges | RW |
| CTS1 | | | RW |
| DF0 | Digital Filter Select Bit | b3 b2<br>0 0 : No digital filter used<br>0 1 : Do not use this combination<br>1 0 : fBTi<br>1 1 : f1 | RW |
| DF1 | | | RW |
| GT | Gating Select Bit (1) | 0: Gating disabled<br>1: Gating enabled | RW |
| GOC | Gating Clear Select Bit (1, 2, 3) | 0: Gating not cleared<br>1: Gating cleared when the base timer matches the GiPOk register (k = j - 2) | RW |
| GSC | Gating Clear Bit (1, 2) | Gating is cleared by setting this bit to 1 | RW |
| PR | Prescaler Select Bit (1) | 0: No prescaler used<br>1: Use the prescaler | RW |

Notes:
1. These functions are available in registers GiTMCR6 and GiTMCR7. Bits 4 to 7 in registers GiTMCR0 to GiTMCR5 should be set to 0.
2. These bit settings are enabled when the GT bit is set to 1.
3. The GOC bit becomes 0 after gating is cleared.

**Figure 23.9 Registers G0TMCR0 to G0TMCR7 and G1TMCR0 to G1TMCR7**

## Group i Time Measurement Prescaler Register j (i = 0, 1; j = 6, 7)



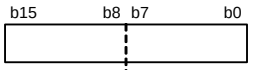

| Symbol | Address | Reset Value |
|---|---|---|
| G0TPR6 to G0TPR7 | 01A4h, 01A5h | 00h |
| G1TPR6 to G1TPR7 | 0124h, 0125h | 00h |

| Function | Setting Range | RW |
|---|---|---|
| Time measurement is executed whenever a trigger input is counted by n+1 (n = setting value) (1) | 00h to FFh | RW |

Note:
1. The first prescaler, after the PR bit in the GiTMCRj register is changed from 0 (no prescaler is used) to 1 (the prescaler is used), may be divided by n rather than n+1. The subsequent prescaler is divided by n+1.

**Figure 23.10 Registers G0TPR6, G0TPR7, G1TPR6 and G1TPR7**

Group i Time Measurement Register j (i = 0, 1; j = 0 to 7)

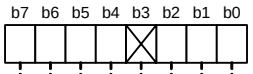



| Symbol | Address | Reset Value |
|---|---|---|
| G0TM0, G0TM1 | 0181h-0180h, 0183h-0182h | Undefined |
| G0TM2, G0TM3 | 0185h-0184h, 0187h-0186h | Undefined |
| G0TM4, G0TM5 | 0189h-0188h, 018Bh-018Ah | Undefined |
| G0TM6, G0TM7 | 018Dh-018Ch, 018Fh-018Eh | Undefined |
| G1TM0, G1TM1 | 0101h-0100h, 0103h-0102h | Undefined |
| G1TM2, G1TM3 | 0105h-0104h, 0107h-0106h | Undefined |
| G1TM4, G1TM5 | 0109h-0108h, 010Bh-010Ah | Undefined |
| G1TM6, G1TM7 | 010Dh-010Ch, 010Fh-010Eh | Undefined |

| Function | Setting Range | RW |
|---|---|---|
| The base timer value is stored every measurement timing | — | RO |

**Figure 23.11 Registers G0TM0 to G0TM7 and G1TM0 to G1TM7**

## Group i Waveform Generation Control Register j (i = 0, 1; j = 0 to 7)



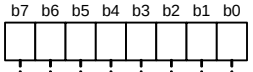

| Symbol | Address | Reset Value |
|---|---|---|
| G0POCR0 | 0190h | 0000 X000b |
| G0POCR1 to G0POCR3 | 0191h, 0192h, 0193h | 0X00 X000b |
| G0POCR4 to G0POCR7 | 0194h, 0195h, 0196h, 0197h | 0X00 X000b |
| G1POCR0 | 0110h | 0000 X000b |
| G1POCR1 to G1POCR3 | 0111h, 0112h, 0113h | 0X00 X000b |
| G1POCR4 to G1POCR7 | 0114h, 0115h, 0116h, 0117h | 0X00 X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| MOD0 | Operating Mode Select Bit | b2 b1 b0<br>0 0 0 : Single-phase waveform output mode<br>0 0 1 : SR waveform output mode (1)<br>0 1 0 : Inverted waveform output mode<br>0 1 1 : Do not use this combination<br>1 0 0 : Do not use this combination<br>1 0 1 : Do not use this combination<br>1 1 0 : Do not use this combination<br>1 1 1 : Do not use this combination | RW |
| MOD1 | | | RW |
| MOD2 | | | RW |
| — (b3) | No register bit; should be written with 0 and read as undefined value | | — |
| IVL | Default Output Value Select Bit (2) | 0: Output low as default value<br>1: Output high as default value | RW |
| RLD | GiPOj Register Value Reload Timing Select Bit | 0: Reload the value into the GiPOj register on a write access<br>1: Reload the value into the GiPOj register when the base timer is reset | RW |
| BTRE | Base Timer Reset Enable Bit (3) | 0: Reset the base timer when the bit 15 overflows<br>1: Reset the base timer when the bit 9 overflows (4) | RW |
| INV | Output Level Inversion Select Bit (5) | 0: Do not invert the output level<br>1: Invert the output level | RW |

Notes:
1. This bit setting is enabled only for even channels. In SR waveform output mode, the corresponding odd channel (the next channel after an even channel) setting is ignored. Waveforms are only output from even channels.
2. The setting value is output by a write to the IVL bit if the FSCj bit in the GiFS register is set to 0 (the waveform generation selected) and the IFEj bit in the GiFE register is set to 1 (the function for channel j enabled).
3. This bit is available only in the GiPOCR0 register. Bit 6 in registers GiPOCR1 to GiPOCR7 should be set to 0.
4. To set the BTRE bit to 1, bits BCK1 and BCK0 in the GiBCR0 register should be set to 11b (f1) and bits UD1 and UD0 in the GiBCR1 register should be set to 00b (increment counting mode).
5. The output level inversion is the final step in waveform generation process. When the INV bit is set to 1 (output level is inverted), high is output by setting the IVL bit to 0 (output low as default value), and vice versa.

**Figure 23.12 Registers G0POCR0 to G0POCR7 and G1POCR0 to G1POCR7**

## Group 2 Waveform Generation Control Register j (j = 0 to 7)



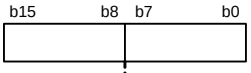

| Symbol | Address | Reset Value |
|---|---|---|
| G2POCR0 to G2POCR3 | 0150h, 0151h, 0152h, 0153h | 0000 0000b |
| G2POCR4 to G2POCR7 | 0154h, 0155h, 0156h, 0157h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| MOD0 | Operating Mode Select Bit (1) | b2 b1 b0<br>0 0 0 : Single waveform output mode<br>0 0 1 : SR waveform output mode (2)<br>0 1 0 : Inverted waveform output mode<br>0 1 1 : Do not use this combination<br>1 0 0 : Bit modulation PWM output mode<br>1 0 1 : Do not use this combination<br>1 1 0 : Do not use this combination<br>1 1 1 : Use an output for the serial interface (3) | RW |
| MOD1 | | | RW |
| MOD2 | | | RW |
| PRT | Parallel Real-Time Port Output Trigger Select Bit (4) | 0: Not triggered by matching the base timer with registers G2PO0 to G2PO7<br>1: Triggered by matching the base timer with registers G2PO0 to G2PO7 | RW |
| IVL | Default Output Value Select Bit | 0: Output low as default value<br>1: Output high as default value | RW |
| RLD | G2POj Register Value Reload Timing Select Bit | 0: Reload the value into the G2POj register on a write access<br>1: Reload the value into the G2POj register when the base timer is reset | RW |
| RTP | Real-Time Port Select Bit | 0: No real-time port function used<br>1: Use RTP output mode or parallel RTP output mode | RW |
| INV | Output Level Inversion Select Bit (5) | 0: Do not invert the output level<br>1: Invert the output level | RW |

Notes:
1. When the RTP bit is set to 1, the settings of bits MOD2 to MOD0 are disabled.
2. This bit setting is enabled only for even channels. In SR waveform output mode, the corresponding odd channel (the next channel after an even channel) setting is ignored. Waveforms are not output from odd channels but even channels.
3. This bit setting is enabled only for channels 0 and 1 of the group 2. To use the ISTxD2 or IEOUT pin as an output, bits MOD2 to MOD0 in the G2POCR0 register should be set to 111b. To use the ISCLK2 pin, the same bits in the G2POCR1 register should be set to 111b. This bit setting should not be done in other channels than 0 and 1.
4. This bit setting is enabled when the RTP bit is set to 1 (real-time port used) and the PRP bit in the G2BCR1 register is set to 1 (parallel RTP output mode).
5. The output level inversion is the final step in waveform generation process. When the INV bit is set to 1 (output level is inverted), high is output by setting the IVL bit to 0 (output low as default value), and vice versa.

**Figure 23.13 Registers G2POCR0 to G2POCR7**

## Group i Waveform Generation Register j (i = 0 to 2; j = 0 to 7)

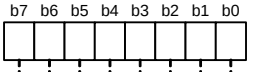



| Symbol | Address | Reset Value |
|---|---|---|
| G0PO0 to G0PO2 | 0181h-0180h, 0183h-0182h, 0185h-0184h | Undefined |
| G0PO3 to G0PO5 | 0187h-0186h, 0189h-0188h, 018Bh-018Ah | Undefined |
| G0PO6, G0PO7 | 018Dh-018Ch, 018Fh-018Eh | Undefined |
| G1PO0 to G1PO2 | 0101h-0100h, 0103h-0102h, 0105h-0104h | Undefined |
| G1PO3 to G1PO5 | 0107h-0106h, 0109h-0108h, 010Bh-010Ah | Undefined |
| G1PO6, G1PO7 | 010Dh-010Ch, 010Fh-010Eh | Undefined |
| G2PO0 to G2PO2 | 0141h-0140h, 0143h-0142h, 0145h-0144h | Undefined |
| G2PO3 to G2PO5 | 0147h-0146h, 0149h-0148h, 014Bh-014Ah | Undefined |
| G2PO6, G2PO7 | 014Dh-014Ch, 014Fh-014Eh | Undefined |

| Function | Setting Value | RW |
|---|---|---|
| -If the RLD bit in the GiPOCRj register is set to 0, the value is reloaded into the GiPOj register immediately after being written to be reflected in output waveform<br>-If the RLD bit is set to 1, the value is reloaded when the base timer is reset. The register indicates the written value until the value is reloaded | 0000h to FFFFh | RW |

**Figure 23.14 Registers G0PO0 to G0PO7, G1PO0 to G1PO7, and G2PO0 to G2PO7**

## Group i Function Select Register (i = 0 , 1)

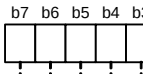



| Symbol | Address | Reset Value |
|---|---|---|
| G0FS, G1FS | 01A7h, 0127h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| FSC0 | Channel 0 Time Measurement/ Waveform Generation Select Bit | 0: Select the waveform generation<br>1: Select the time measurement | RW |
| FSC1 | Channel 1 Time Measurement/ Waveform Generation Select Bit | | RW |
| FSC2 | Channel 2 Time Measurement/ Waveform Generation Select Bit | | RW |
| FSC3 | Channel 3 Time Measurement/ Waveform Generation Select Bit | | RW |
| FSC4 | Channel 4 Time Measurement/ Waveform Generation Select Bit | | RW |
| FSC5 | Channel 5 Time Measurement/ Waveform Generation Select Bit | | RW |
| FSC6 | Channel 6 Time Measurement/ Waveform Generation Select Bit | | RW |
| FSC7 | Channel 7 Time Measurement/ Waveform Generation Select Bit | | RW |

**Figure 23.15 Registers G0FS and G1FS**

## Group i Function Enable Register (i = 0 to 2)

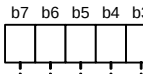



| Symbol | Address | Reset Value |
|---|---|---|
| G0FE to G2FE | 01A6h, 0126h, 0166h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IFE0 | Channel 0 Function Enable Bit | 0: Disable the function for channel j<br>1: Enable the function for channel j<br>(j = 0 to 7) | RW |
| IFE1 | Channel 1 Function Enable Bit | | RW |
| IFE2 | Channel 2 Function Enable Bit | | RW |
| IFE3 | Channel 3 Function Enable Bit | | RW |
| IFE4 | Channel 4 Function Enable Bit | | RW |
| IFE5 | Channel 5 Function Enable Bit | | RW |
| IFE6 | Channel 6 Function Enable Bit | | RW |
| IFE7 | Channel 7 Function Enable Bit | | RW |

**Figure 23.16 Registers G0FE to G2FE**

## Group 2 RTP Output Buffer Register




| Symbol | Address | Reset Value |
|---|---|---|
| G2RTP | 0167h | 0000 0000b |

| Bit symbol | Bit Name | Function | RW |
|---|---|---|---|
| RTP0 | Channel 0 RTP Output Buffer | 0: Output an low level<br>1: Output an high level | RW |
| RTP1 | Channel 1 RTP Output Buffer | | RW |
| RTP2 | Channel 2 RTP Output Buffer | | RW |
| RTP3 | Channel 3 RTP Output Buffer | | RW |
| RTP4 | Channel 4 RTP Output Buffer | | RW |
| RTP5 | Channel 5 RTP Output Buffer | | RW |
| RTP6 | Channel 6 RTP Output Buffer | | RW |
| RTP7 | Channel 7 RTP Output Buffer | | RW |

**Figure 23.17 G2RTP Register**

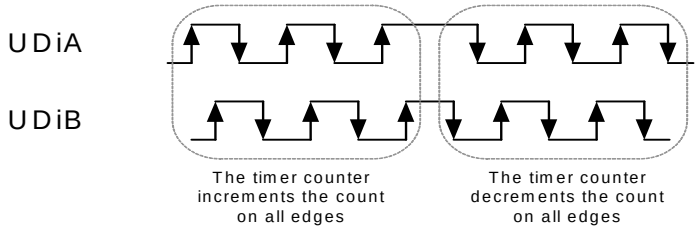
## 23.1 Base Timer (for Groups 0 to 2)

The base timer is a free-running counter that counts an internally generated count source. Table 23.2 lists specifications of the base timer. Figure 23.4 to Figure 23.17 show registers associated with the base timer. Figure 23.18 shows a block diagram of the base timer. Figure 23.19, Figure 23.20, and Figure 23.21 show respectively an operation example of the base timer (for groups 0 and 1) in increment counting mode, in increment/decrement counting mode, and in two-phase pulse signal processing mode.

**Table 23.2 Base Timer Specifications (i = 0 to 2)**

| Item | Specification |
|---|---|
| Count source (fBTi) | f1 divided by 2(n+1) (for groups 0 to 2), two-phase pulse input divided by 2(n+1) (for groups 0 and 1)<br>n: setting value using bits DIV4 to DIV0 in the GiBCR0 register<br>n = 0 to 31; however no division when n = 31 |
| Count operations | • Increment counting<br>• Increment/decrement counting<br>• Two-phase pulse signal processing |
| Count start conditions | • To start each base timer individually,<br>the BTS bit in the GiBCR1 register is set to 1 (count starts)<br>• To start base timers of two or all groups simultaneously,<br>the BTiS bit in the BTSR register is set to 1 (count starts) |
| Count stop condition | The BTiS bit in the BTSR register is set to 0 (base timer is reset) and the BTS bit in the GiBCR1 register is set to 0 (base timer is reset) |
| Reset conditions | • The base timer value matches the GiPO0 register setting<br>• An input of low signal into the external interrupt pin as follows:<br>for group 0: the $\overline{\text{INT0}}$ pin<br>for group 1: the $\overline{\text{INT1}}$ pin<br>• The overflow of bit 15 or 9 in the base timer<br>• The base timer reset request from the communication functions (group 2) |
| Reset value | 0000h |
| Interrupt request | When the BTiR bit in the interrupt request register is set to 1 (interrupts requested) by the overflow of bit 9, 14, or 15 in the base timer (refer to Figure 11.12) |
| Read from base timer | • The GiBT register indicates a counter value while the base timer is running<br>• The GiBT register is undefined while the base timer is being reset |
| Write to base timer | When a value is written while the base timer is running, the timer counter immediately starts counting from this value. No value can be written while the base timer is being reset |

**Table 23.2 Base Timer Specifications (i = 0 to 2)**

| Item | Specification |
|---|---|
| Selectable functions | • Increment/decrement counting mode (for groups 0 and 1)<br>The base timer starts counting when the BTS or BTiS bit is set to 1. On reaching FFFFh, it starts decrement counting. When the RST1 bit in the GiBCR1 register is set to 1 (reset by match with the GiPO0 register), the timer counter starts decrement counting as soon as the base timer value has matched the GiPO0 register setting. When the timer counter has reached 0000h, it starts increment counting again (Refer to Figure 23.20).<br>• Two-phase pulse signal processing mode (for groups 0 and 1)<br>Two-phase pulse signals at pins UDiA and UDiB are counted (Refer to Figure 23.21).<br>  |

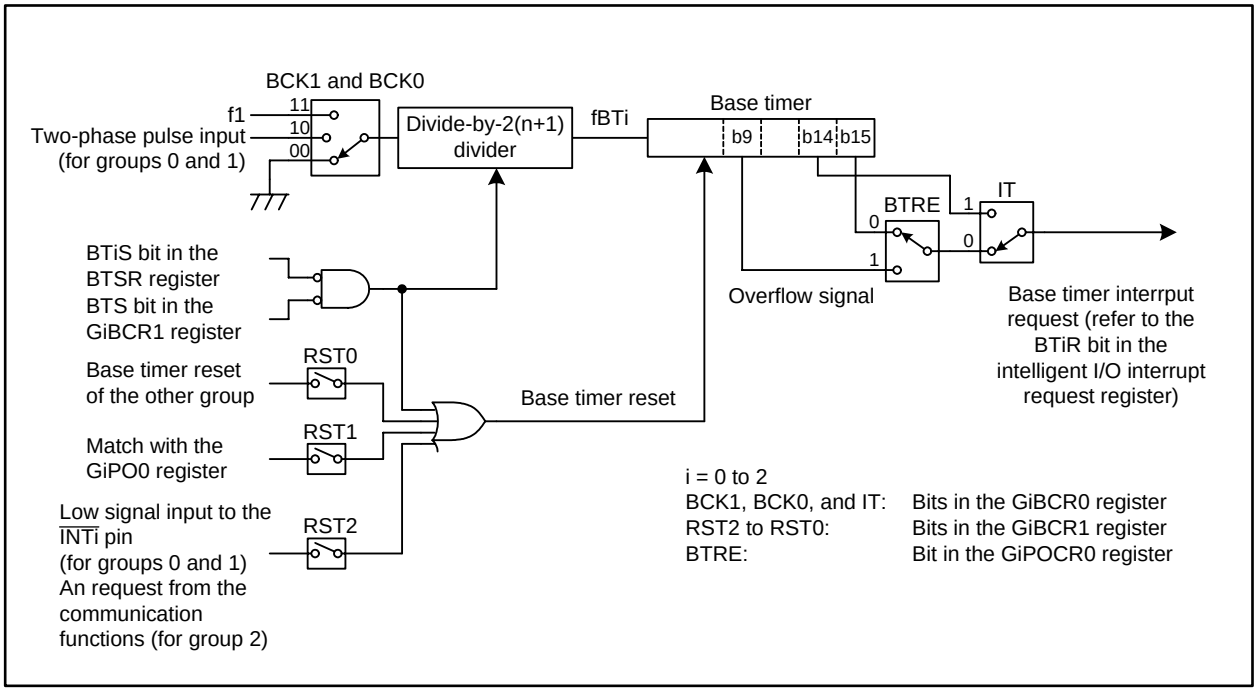



**Figure 23.18 Base Timer Block Diagram**

**Table 23.3 Base Timer Associated Register Settings (Common Settings for Time Measurement, Waveform Generation, and Serial Interface) (i = 0 to 2)**

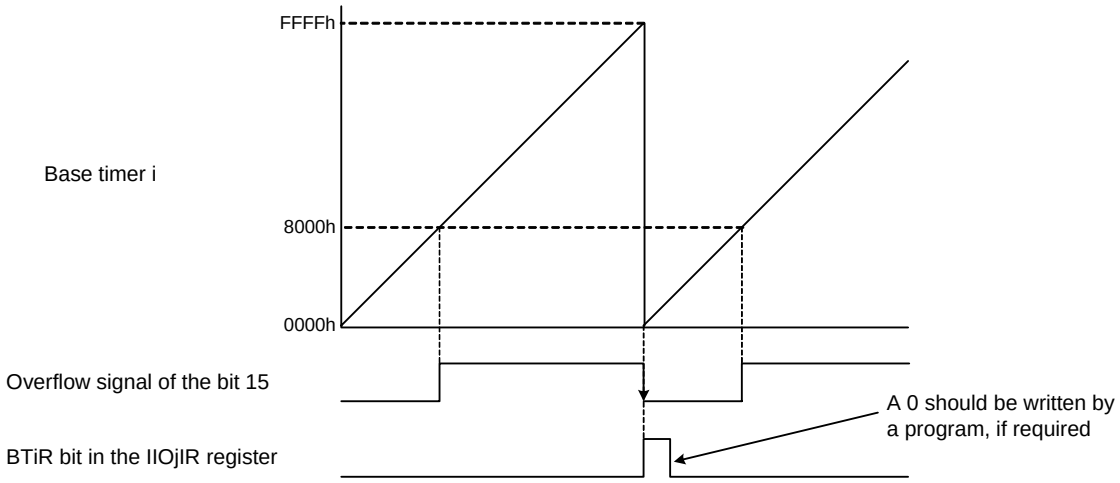
| Register | Bits | Function |
|---|---|---|
| G2BCR0 | — | Provide an operating clock to the BTSR register. Set to 0111 1111b |
| BTSR | — | Set to 0000 0000b |
| GiBCR0 | BCK1 and BCK0 | Select a count source |
| | DIV4 to DIV0 | Select a divide ratio of count source |
| | IT | Select a base timer interrupt source |
| GiBCR1 | RST2 to RST0 | Select a timing for base timer reset |
| | BTS | Use this bit when each base timer individually starts counting |
| | UD1 and UD0 | Select a count operation (in groups 0 and 1) |
| GiPOCR0 | BTRE | Select a source for base timer reset |
| GiBT | — | Read or write the base timer value |

The following register settings are required to set the RST1 bit to 1 (base timer is reset by match with the GiPO0 register).

| | | |
|---|---|---|
| GiPOCR0 | MOD2 to MOD0 | Set to 000b (single-phase waveform output mode) |
| GiPO0 | — | Set the reset cycle |
| GiFS | FSC0 | Set the bit to 0 (waveform generation) |
| GiFE | IFE0 | Set the bit to 1 (channel operation starts) |

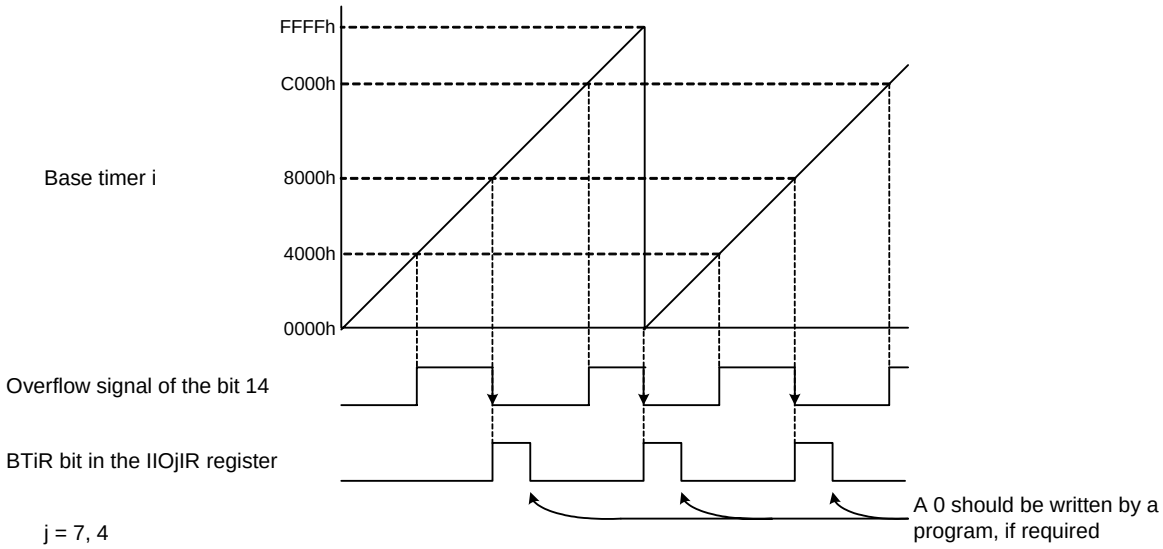
Bit configurations and functions vary with the groups.

(1) When the IT bit in the GiBCR0 register (i = 0, 1) is set to 0
(an interrupt is requested by the overflow of bit 15 in the base timer)




The figure above applies under the following conditions:
- The RST1 bit in the GiBCR1 register is set to 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)

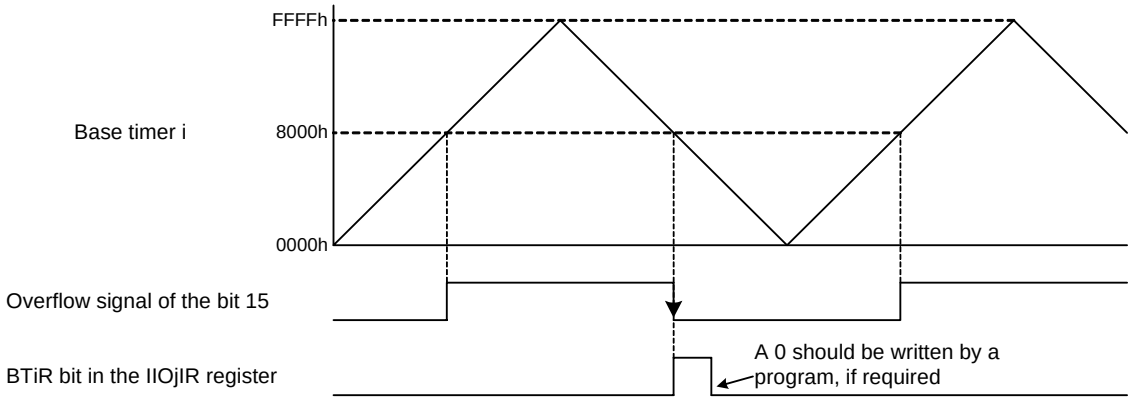
(2) When the IT bit in the GiBCR0 register (i = 0, 1) is set to 1
(an interrupt is requested by the overflow of bit 14 in the base timer)




The figure above applies under the following conditions:
- The RST1 bit in the GiBCR1 register is set to 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)

**Figure 23.19 Base Timer Increment Counting Mode (for Groups 0 and 1)**

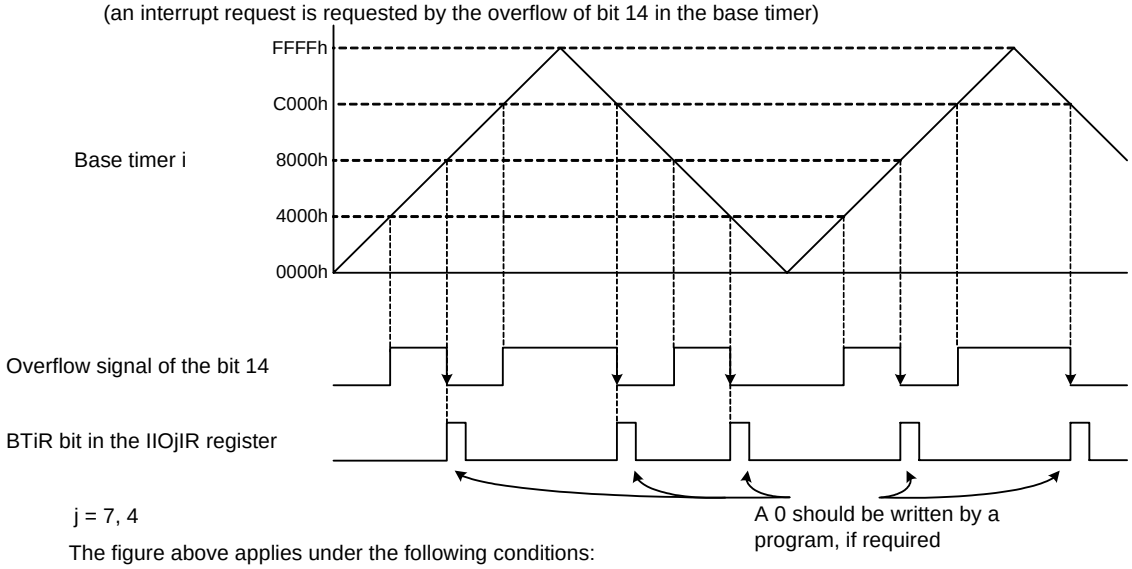
(1) When the IT bit in the GiBCR0 register (i = 0, 1) is set to 0
(an interrupt is requested by the overflow of bit 15 in the base timer)




j = 7, 4

The figure above applies under the following conditions:
- The RST1 bit in the GiBCR1 register is set to 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are set to 01b (increment/decrement counting mode)

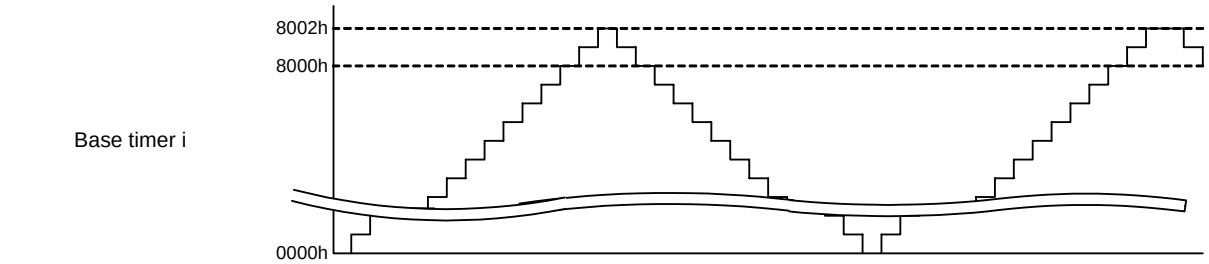
(2) When the IT bit in the GiBCR0 register (i = 0, 1) is set to 1
(an interrupt request is requested by the overflow of bit 14 in the base timer)




j = 7, 4

The figure above applies under the following conditions:
- The RST1 bit in the GiBCR1 register is set to 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are set to 01b (increment/decrement counting mode)

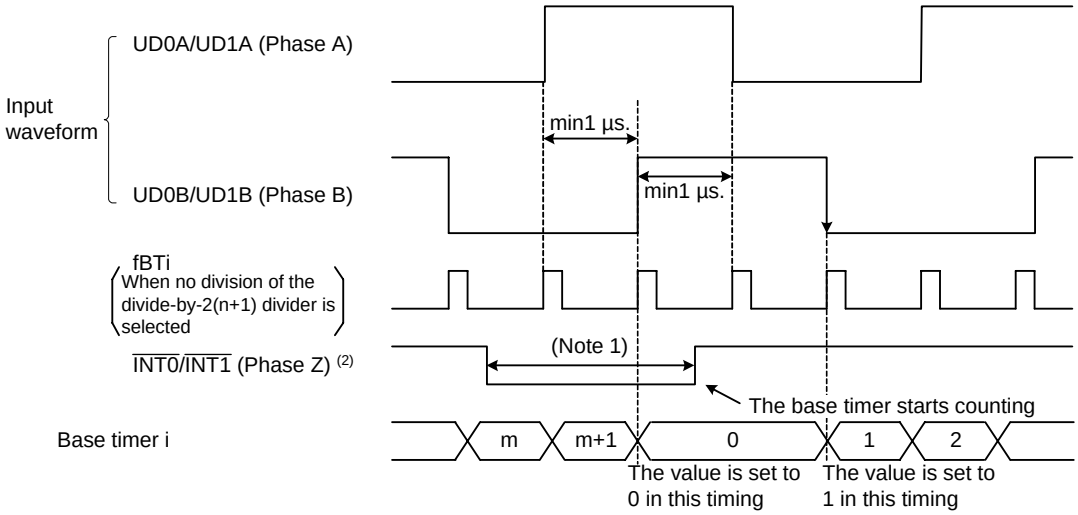
(3) When the RST bit in the GiBCR1 register is set to 1 (the base timer is reset by matching with the GiPO0 register)



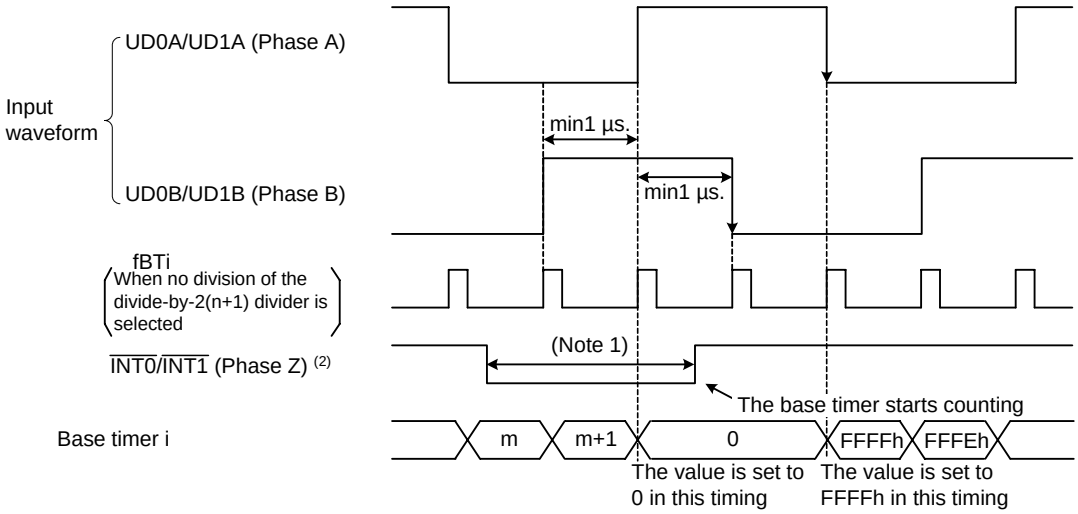

The figure above applies under the following conditions:
- The GiPO0 register value is set to 8000h
- Bits UD1 and UD0 in the GiBCR1 register are set to 01b (increment/decrement counting mode)

**Figure 23.20 Base Timer Increment/Decrement Counting (for Groups 0 and 1)**

(1) When the base timer is reset while it increments the count




(2) When the base timer is reset while it decrements the count




Notes:
1. 1.5 fBTi clock cycles or more are required.
2. The RST2 bit in the GiBCR1 register should be set to 1 in two-phase pulse signal processing mode.

**Figure 23.21 Base Timer Two-phase Pulse Signal Processing Mode (i = 0, 1) (for Groups 0 and 1)**

## 23.2 Time Measurement (for Groups 0 and 1)

Every time an external trigger is input, the base timer value is stored into the GiTMj register (i = 0, 1; j = 0 to 7). Table 23.4 lists specifications of the time measurement and Table 23.5 lists its register settings. Figure 23.22 and Figure 23.23 show operation examples of the time measurement and Figure 23.24 shows operation examples with the prescaler or gate function.

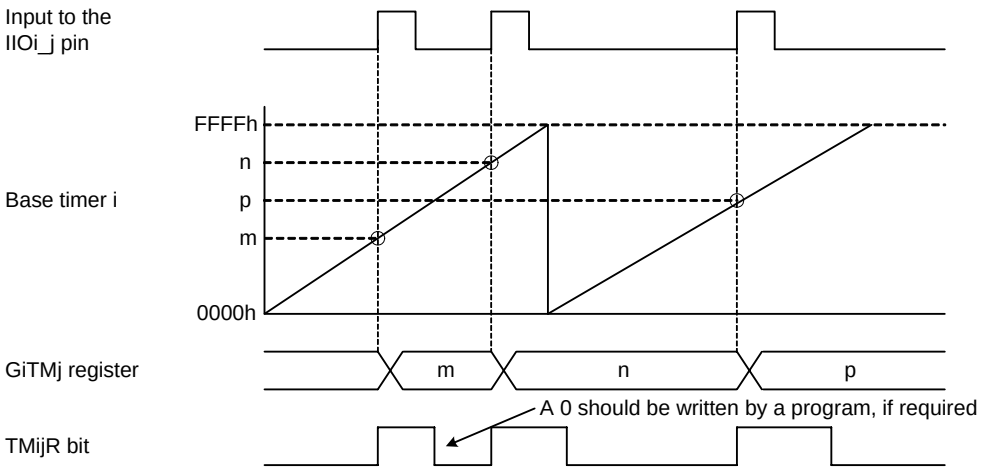
**Table 23.4 Time Measurement Specifications (i = 0, 1; j = 0 to 7)**

| Item | Specification |
|---|---|
| Time measurement channels | Group 0: Channels 0 to 7<br>Group 1: Channels 0 to 7 |
| Trigger input polarity | Rising edge, falling edge, or both edges of the IIOi_j pin |
| Time measurement start condition | The IFEj bit in the GiFE register is set to 1 (function for channel j enabled) while the FSCj bit in the GiFS register is set to 1 (time measurement selected) |
| Time measurement stop condition | The IFEj bit is set to 0 (function for channel j disabled) |
| Time measurement timing | • Without the prescaler: every time a trigger is input<br>• With the prescaler (for channels 6 and 7): every (GiTPRk register (k = 6, 7) value + 1) times a trigger is input |
| Interrupt request | When the TMijR bit in the interrupt request register is set to 1 (interrupts requested) (Refer to Figure 11.12) |
| IIOi_j input pin function | Trigger input |
| Selectable functions | • Digital filter<br>The digital filter determines a trigger input level every f1 or fBTi cycle and passes the signals holding the same level during three sequential cycles<br>• Prescaler (for channels 6 and 7)<br>Time measurement is executed every (GiTPRk register value + 1) times a trigger is input<br>• Gating (for channels 6 and 7)<br>This function disables any trigger input to be accepted after the time measurement by the first trigger input. However, the trigger input can be accepted again if any of following conditions are met while the GOC bit in the GiTMCRk register is set to 1 (the gating is cleared when the base timer matches the GiPOp register) (p = 4, 5; p = 4 when k = 6; p = 5 when k = 7):<br>• The base timer value matches the GiPOp register setting<br>• The GSC bit in the GiTMCRk register is set to 1 |

**Table 23.5 Time Measurement (for Groups 0 and 1) Associated Register Settings (i = 0, 1; j = 0 to 7; k = 6, 7)**

| Register | Bits | Function |
|---|---|---|
| GiTMCRj | CTS1 and CTS0 | Select a time measurement trigger |
| | DF1 and DF0 | Select a digital filter |
| | GT, GOC, GSC | Select if the gating is used |
| | PR | Select if the prescaler is used |
| GiTPRk | — | Set the prescaler value |
| GiFS | FSCj | Set the bit to 1 (the time measurement selected) |
| GiFE | IFEj | Set the bit to 1 (function for channel j enabled) |

Bit configurations and functions vary with the channels or groups.
Registers associated with the time measurement should be set after setting the base timer-associated registers.



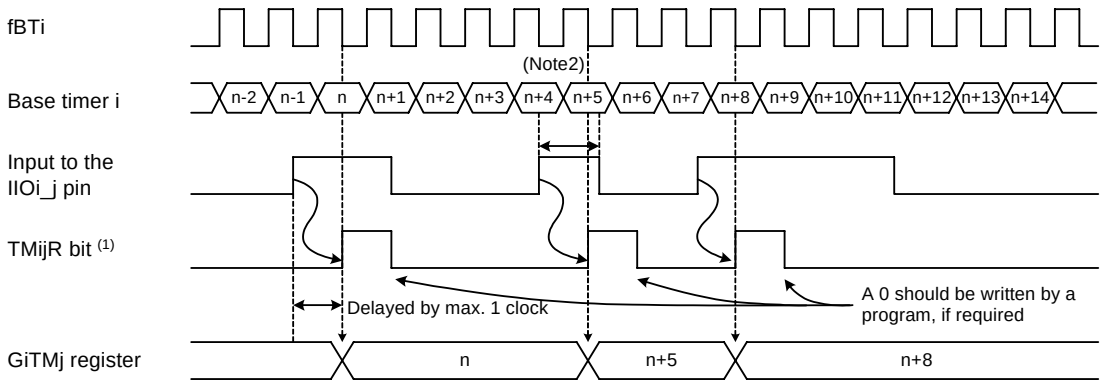

TMijR bit: Bit in registers IIO0IR to IIO11IR

This figure above applies under the following conditions:
- Bits CTS1 and CTS0 in the GiTMCRj register are set to 01b (rising edge as time measurement trigger), the PR bit is set to 0 (no prescaler is used) and the GT bit is set to 0 (no gating is used)
- Bits RST2 to RST0 in the GiBCR1 register are set to 000b (base timer is not reset) and bits UD1 and UD0 are set to 00b (increment counting mode)

The base timer reset by matching the GiPO0 register setting (bits RST2 to RST0 in the GiBCR1 register are set to 010b) is done after the base timer reaches the GiPO0 register setting value + 2.

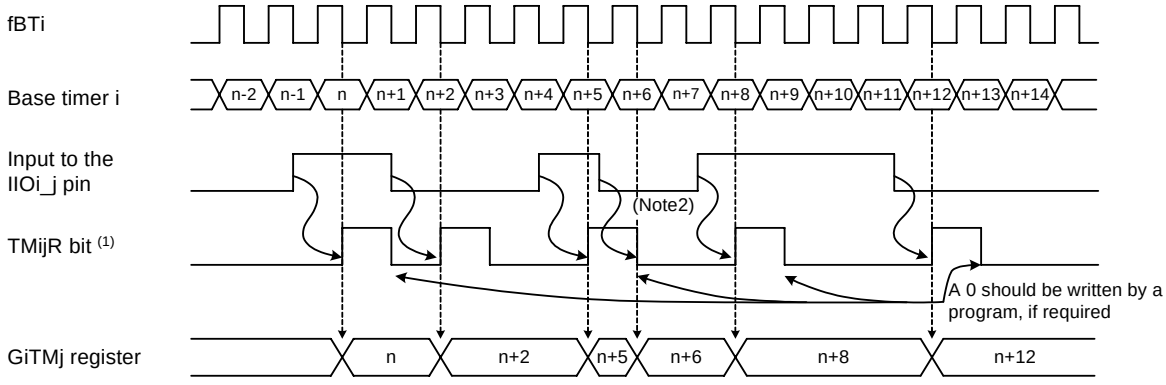
**Figure 23.22 Time Measurement Operation (1) (i = 0, 1; j = 0 to 7)**

Case 1: When selecting the rising edge as a time measurement trigger
(Bits CTS1 and CTS0 in the GiTMCRj register are set to 01b)



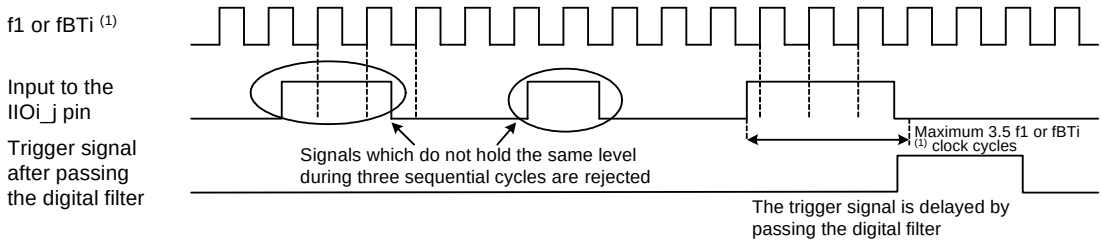

Notes:
1. Bit in registers IIO0IR to IIO11IR
2. Input pulse applied to the IIOi_j pin requires 1.5 fBTi clock cycles or more.

Case 2: When selecting both edges as a time measurement trigger
(Bits CTS1 and CTS0 in the GiTMCRj register are set to 11b)




Notes:
1. Bit in registers IIO0IR to IIO11IR
2. No interrupt is generated if the MCU receives a trigger signal when the TMijR bit is set to 1. However, the value of GiTMj register changes.

Case 3: Trigger signal when using the digital filter
(Bits DF1 and DF0 in the GiTMCRj register are set to 10b or 11b)

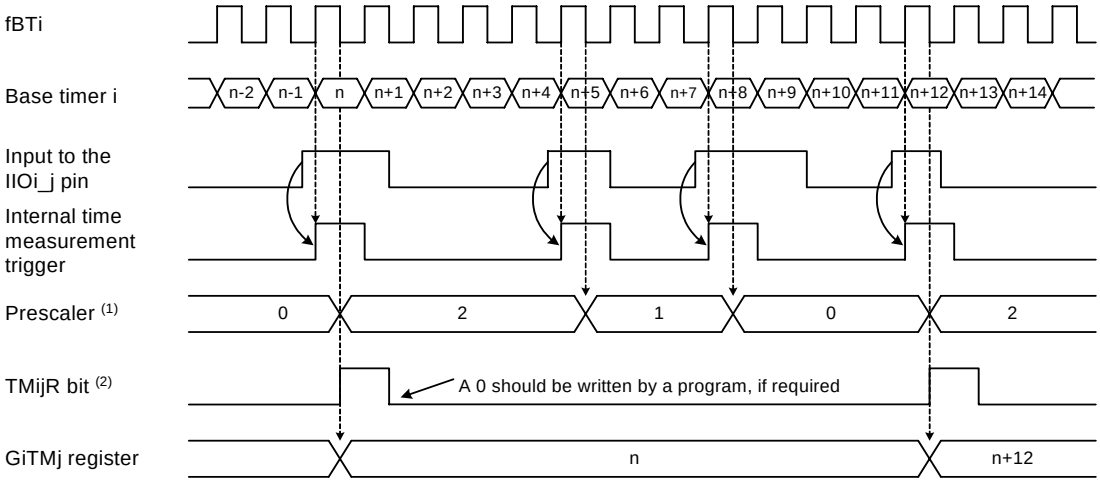



Note:
1. fBTi when bits DF1 and DF0 are set to 10b, f1 when the bits are set to 11b.

**Figure 23.23 Time Measurement Operation (2) (i = 0, 1; j = 0 to 7)**

Case 1: Operation with the prescaler

(The GiTPRj register is set to 02h and the PR bit in the GiTMCRj register is set to 1)

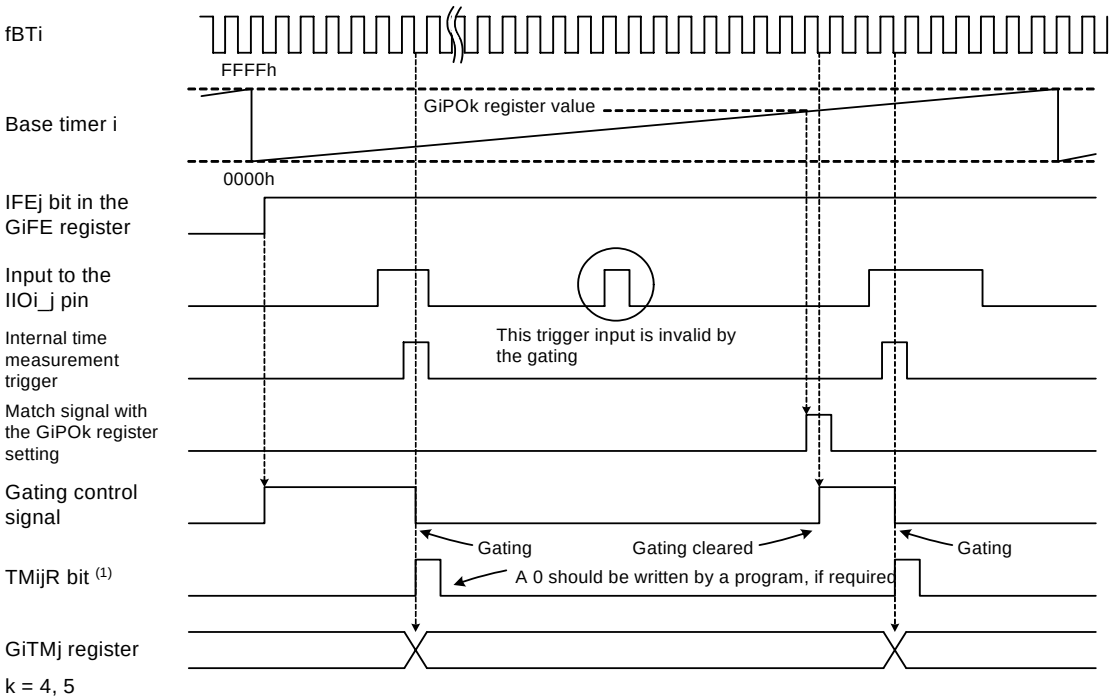



Notes:

1. This example applies to cycles following the first cycle after the PR bit in the GiTMCRj register is set to 1 (the prescaler is used)
2. Bit in registers IIO0IR to IIO11IR

Case 2: Operation with the gating

(The gating is cleared by matching the base timer value with the GiPOk register setting. Bits GT and GOC in the GiTMCRj register are respectively set to 1)




Note:

1. Bit in registers IIO0IR to IIO11IR

**Figure 23.24 Prescaler and Gate Operations (i = 0, 1; j = 6, 7)**

## 23.3 Waveform Generation (for Groups 0 to 2)

Waveforms are generated when the base timer value matches the GiPOj register setting (i = 0 to 2; j = 0 to 7).

Waveform generation has the following six modes:

- Single-phase waveform output mode (for groups 0 to 2)
- Inverted waveform output mode (for groups 0 to 2)
- Set/reset waveform output (SR waveform output) mode (for groups 0 to 2)
- Bit modulation PWM output mode (for group 2)
- Real-time port output (RTP output) mode (for group 2)
- Parallel real-time port output (parallel RTP output) mode (for group 2)

Table 23.6 lists registers associated with the waveform generation.

**Table 23.6 Waveform Generation Associated Register Settings (i = 0 to 2; j = 0 to 7)**

| Register | Bits | Function |
|---|---|---|
| GiPOCRj | MOD2 to MOD0 | Select a waveform output mode |
| | PRT (1) | Set the bit to 1 to use parallel RTP output mode |
| | IVL | Select a default value |
| | RLD | Select a timing to reload the value into the GiPOj register |
| | RTP (1) | Set the bit to 1 to use RTP output mode or parallel RTP output mode. The settings of bits MOD2 to MOD0 are disabled when this bit is set to 1 |
| | INV | Select if output level is inverted |
| G2BCR1 | PRP | Set the bit to 1 to use parallel RTP output mode |
| GiPOj | — | Set the timing to invert output waveform |
| GiFS | FSCj | Set the bit to 0 (the waveform generation selected) (for groups 0 and 1 only) |
| GiFE | IFEj | Set the bit to 1 (the function for channel j enabled) |
| G2RTP | RTP0 to RTP7 | Set the RTP output value in RTP output mode or parallel RTP output mode |

Bit configurations and functions vary with channels or groups.

Registers associated with the waveform generation should be set after setting the base timer-associated registers.

Note:

1. This bit is available in the G2POCRj register only. Neither the G0POCRj nor G1POCRj register has it.

### 23.3.1 Single-phase Waveform Output Mode (for Groups 0 to 2)

The output level at the IIOi_j pin (or OUTC2_j pin for Group 2) becomes high when the base timer value matches the GiPOj register (i = 0 to 2; j = 0 to 7). It switches to low when the base timer reaches 0000h. If the IVL bit in the GiPOCRj register is set to 1 (high level is output as default value), a high level output is provided when a waveform output starts. If the INV bit is set to 1 (output level is inverted), a waveform with inverted level is output. Refer to Figure 23.25 for details on single-phase waveform mode operation.

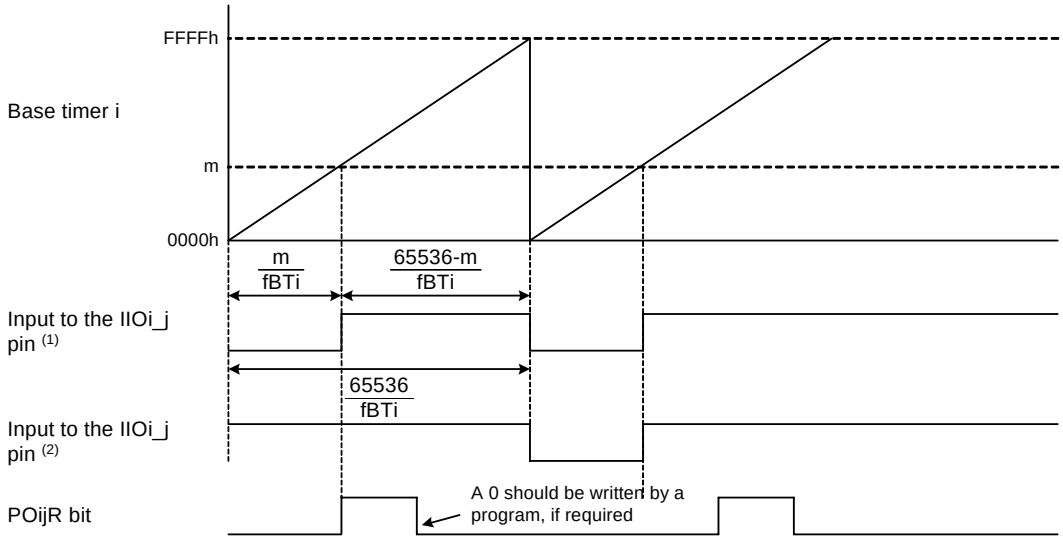
Table 23.7 lists specifications of single-phase waveform output mode.

**Table 23.7 Single-phase Waveform Output Mode Specifications (i= 0 to 2)**

| Item | Specification |
|---|---|
| Output waveform (1) | • Free-running operation (when bits RST2 to RST0 in the GiBCR1 register are set to 000b)<br>Cycle: $\frac{65536}{fBTi}$<br>Low level width: $\frac{m}{fBTi}$<br>High level width: $\frac{65536 - m}{fBTi}$<br>m: GiPOj register (j = 0 to 7) setting value, 0000h to FFFFh<br>• The base timer is reset by matching the base timer value with the GiPO0 register setting (when bits RST2 to RST0 are set to 010b)<br>Cycle: $\frac{n + 2}{fBTi}$<br>Low level width: $\frac{m}{fBTi}$<br>High level width: $\frac{n + 2 - m}{fBTi}$<br>m: GiPOj register (j = 1 to 7) setting value, 0000h to FFFFh<br>n: GiPO0 register setting value, 0001h to FFFDh<br>If m ≥ n + 2, the output level is fixed to low |
| Waveform output start condition (2) | The IFEj (j = 0 to 7) bit in the GiFE register is set to 1 (the function for channel j is enabled) |
| Waveform output stop condition | The IFEj bit is set to 0 (the function for channel j is disabled) |
| Interrupt request | When the POijR bit in the intelligent I/O interrupt request register is set to 1 (interrupts requested) by matching the base timer value with the GiPOj register setting (Refer to Figure 11.12) |
| IIOi_j output pin (or OUTC2_j pin for Group 2) function | Pulse signal output |
| Selectable functions | • Default value setting<br>This function determines the starting waveform output level<br>• Output level inversion<br>This function inverts the waveform output level and output the inverted signal from the IIOi_j pin (or OUTC2_j pin for Group 2) |

Notes:
1. When the INV bit in the GiPOCRj register is set to 1 (output level is inverted), widths low and high are inverted.
2. To use channels shared by time measurement and waveform generation, the FSCj bit in the GiFS register should be set to 0 (waveform generation is selected).

Case 1: Free-running operation (Bits RST2 to RST0 in the GiBCR register are set to 000b)

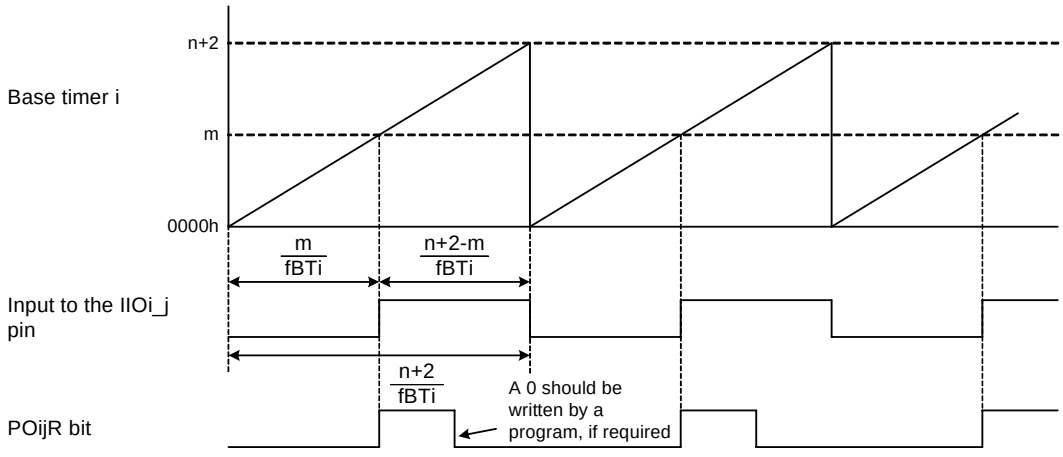



j = 0 to 7
m: GiPOj register setting value (0000h to FFFFh)
POijR bit: Bit in registers IIO0IR to IIO11IR

Notes:
1. Output waveform when the INV bit in the GiPOCRj register is set to 0 (output level is not inverted) and the IVL bit is set to 0 (low level is output as default value).
2. Output waveform when the INV bit is set to 0 (output level is not inverted) and the IVL bit is set to 1 (high level is output as default value).

This figure above applies under the following condition:
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)

Case 2: The base timer is reset by matching the base timer value with the GiPO0 register setting (Bits RST2 to RST0 in the GiBCR register are set to 010b)




j = 1 to 7
m: GiPOj register setting value (0000h to FFFFh)
n: GiPO0 register setting value (0001h to FFFDh)
POijR bit: Bit in registers IIO0IR to IIO11IR

This figure above applies under the following conditions:
- The IVL bit in the GiPOCRj register is set to 0 (low level is output as default value) and the INV bit is set to 0 (output is not inverted)
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)
- m < n+2

**Figure 23.25 Single-phase Waveform Output Mode Operation (i = 0 to 2)**

### 23.3.2 Inverted Waveform Output Mode (for Groups 0 to 2)

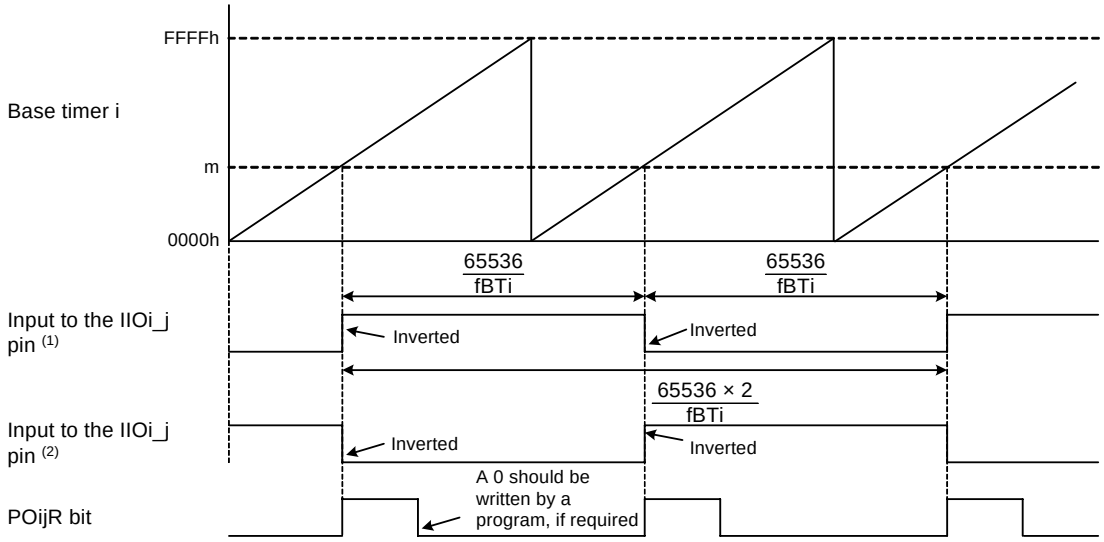
The output level at the IIOi_j pin (or OUTC2_j pin for Group 2) is inverted every time the base timer value matches the GiPOj register setting (i = 0 to 2; j = 0 to 7).
Table 23.8 lists specifications of the inverted waveform output mode. Figure 23.26 shows an example of the inverted waveform output mode operation.

**Table 23.8 Inverted Waveform Output Mode Specifications (i = 0 to 2)**

| Item | Specification |
|---|---|
| Output waveform | • Free-running operation (when bits RST2 to RST0 in the GiBCR1 register are set to 000b)<br>Cycle: $\frac{65536 \times 2}{fBTi}$<br>High or low level width: $\frac{65536}{fBTi}$<br>m: GiPOj register (j = 0 to 7) setting value, 0000h to FFFFh<br>• The base timer is reset by matching the base timer value with the GiPO0 register setting (when bits RST2 to RST0 are set to 010b)<br>Cycle: $\frac{2(n+2)}{fBTi}$<br>High or low level width: $\frac{n+2}{fBTi}$<br>n: GiPO0 register setting value, 0001h to FFFDh<br>GiPOj register (j = 1 to 7) setting value, 0000h to FFFFh<br>If the GiPOj register setting ≥ n+2, the output level is not inverted |
| Waveform output start condition (1) | The IFEj bit in the GiFE register (j = 0 to 7) is set to 1 (the function for channel j is enabled) |
| Waveform output stop condition | The IFEj bit is set to 0 (the function for channel j is disabled) |
| Interrupt request | When the POijR bit in the intelligent I/O interrupt request register is set to 1 (interrupts requested) by matching the base timer value with the GiPOj register setting (Refer to Figure 11.12) |
| IIOi_j output pin (or OUTC2_j pin for Group 2) function | Pulse signal output |
| Selectable functions | • Default value setting<br>This function determines the starting waveform output level<br>• Output level inversion<br>This function inverts the waveform output level and outputs the inverted signal from the IIOi_j pin (or OUTC2_j pin for Group 2) |

Note:
1. To use channels shared by time measurement and waveform generation, the FSCj bit in the GiFS register should be set to 0 (waveform generation is selected).

Case 1: Free-running operation (Bits RST2 to RST0 in the GiBCR1 register are set to 000b)

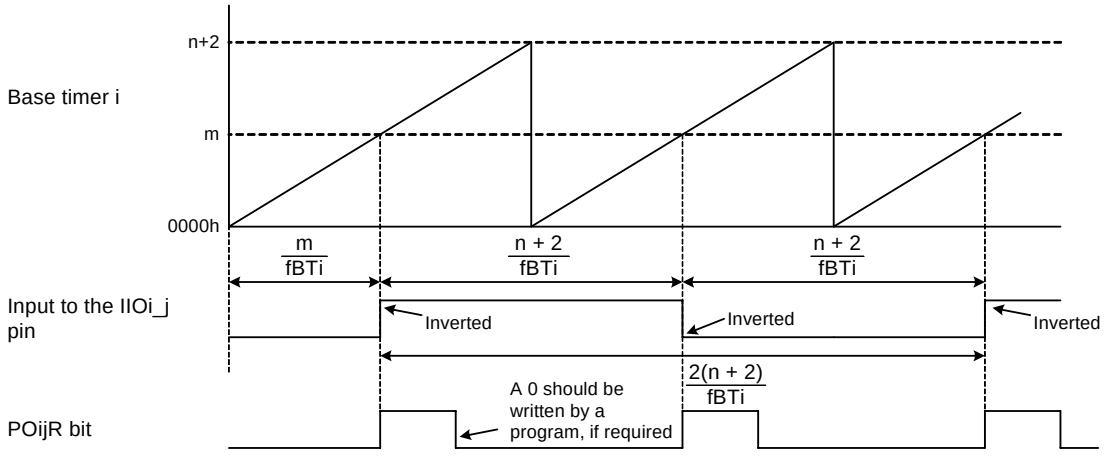



j = 0 to 7
m: GiPOj register setting value (0000h to FFFFh)
POijR bit: Bit in registers IIO0IR to IIO11IR

Notes:
1. Output waveform when the INV bit in the GiPOCRj register is set to 0 (output is not inverted) and the IVL bit is set to 0 (low level is output as default value).
2. Output waveform when the INV bit is set to 0 (output is not inverted) and the IVL bit is set to 1 (high level is output as default value).

This figure above applies under the following condition:
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)

Case 2: The base timer is reset by matching the base timer value with the GiPO0 register setting (Bits RST2 to RST0 are set to 010b)




j = 1 to 7
m: GiPOj register setting value (0000h to FFFFh)
n: GiPO0 register setting value (0001h to FFFDh)
POijR bit: Bit in registers IIO0IR to IIO11IR

This figure above applies under the following conditions:
- The IVL bit in the GiPOCRj register is set to 0 (low level is output as default value) and the INV bit is set to 0 (output is not inverted)
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)
- m < n+2

**Figure 23.26 Inverted Waveform Output Mode Operation (i = 0 to 2)**

### 23.3.3 Set/Reset Waveform Output Mode (SR Waveform Output Mode) (for Groups 0 to 2)

The output level at the IIOi_j pin (or OUTC2_j pin for Group 2) becomes high when the base timer value matches the GiPOj register setting (i = 0 to 2; j = 0, 2, 4, 6). It switches to low when the base timer value matches the GiPOk register setting (k = j + 1) or the base timer reaches 0000h. If the IVL bit in the GiPOCRj register (j = 0 to 7) is set to 1 (high level is output as default value), an high output level is provided when a waveform output starts. If the INV bit is set to 1 (output level is inverted), a waveform with inverted level is output. Refer to Figure 23.27 for details on SR waveform mode operation. Table 23.9 lists specifications of SR waveform output mode.

**Table 23.9 SR Waveform Output Mode Specifications (i = 0 to 2)**

| Item | Specification |
|---|---|
| Output waveform (1) | • Free-running operation (when bits RST2 to RST0 in the GiBCR1 register are set to 000b)<br>(1) m < n<br>High level width: $\frac{n-m}{fBTi}$<br>Low level width: $\frac{m}{fBTi}$ (2) + $\frac{65536-n}{fBTi}$ (3)<br>(2) m ≥ n<br>High level width: $\frac{65536-m}{fBTi}$<br>Low level width: $\frac{m}{fBTi}$<br>m: GiPOj register (j = 0, 2, 4, 6) setting value, 0000h to FFFFh<br>n: GiPOk register (k = j + 1) setting value, 0000h to FFFFh<br><br>• The base timer is reset by matching the base timer value with the GiPO0 register setting (when bits RST2 to RST0 are set to 010b) (4)<br>(1) m < n < p+2<br>High level width: $\frac{n+m}{fBTi}$<br>Low width: $\frac{m}{fBTi}$ (2) + $\frac{p+2-n}{fBTi}$ (3)<br>(2) m < p+2 ≤ n<br>High level width: $\frac{p+2-m}{fBTi}$<br>Low level width: $\frac{m}{fBTi}$<br>(3) m ≥ p+2, output level is fixed to low<br>p: GiPO0 register setting value, 0001h to FFFDh<br>m: GiPOj register (j = 2, 4, 6) setting value, 0000h to FFFFh<br>n: GiPOk register (k = j + 1) setting value, 0000h to FFFFh |
| Waveform output start Condition (5) | The IFEq bit (q = 0 to 7) in the GiFE register is set to 1 (the function for channel q is enabled) |
| Waveform output stop condition | The IFEq bit is set to 0 (the function for channel q is disabled) |

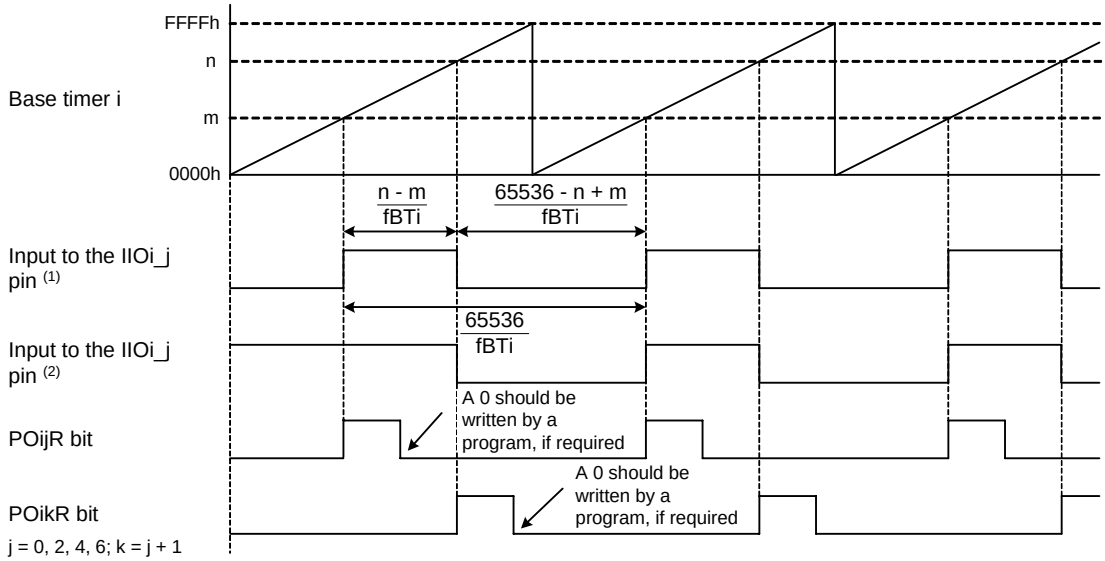
**Table 23.9 SR Waveform Output Mode Specifications (i = 0 to 2)**

| Item | Specification |
|---|---|
| Interrupt request | When the POijR bit in the intelligent I/O interrupt request register is set to 1 (interrupts requested) by matching the base timer value with the GiPOj register setting.<br>When the POikR bit is set to 1 (interrupts requested) by matching the base timer value with the GiPOk register setting (refer to Figure 11.12) |
| IIOi_j output pin (or OUTC2_j pin for Group 2) function | Pulse signal output |
| Selectable functions | • Default value setting<br>This function determines the starting waveform output level<br>• Output level inversion<br>This function inverts the waveform output level and output the inverted signal from the IIOi_j pin (or OUTC2_j pin for Group 2) |

Notes:

1. When the INV bit in the GiPOCRj register is set to 1 (output is inverted), widths low and high are inverted.
2. Output period from a base timer reset until when the output level becomes high.
3. Output period from when the output level becomes low until the next base timer reset.
4. When the GiPO0 register resets the base timer, channels 0 and 1 SR waveform generation functions are not available.
5. To use channels shared by time measurement and waveform generation, the FSCj bit in the GiFS register should be set to 0 (waveform generation is selected).

Case 1: Free-running operation (Bits RST2 to RST0 in the GiBCR1 register are set to 000b)

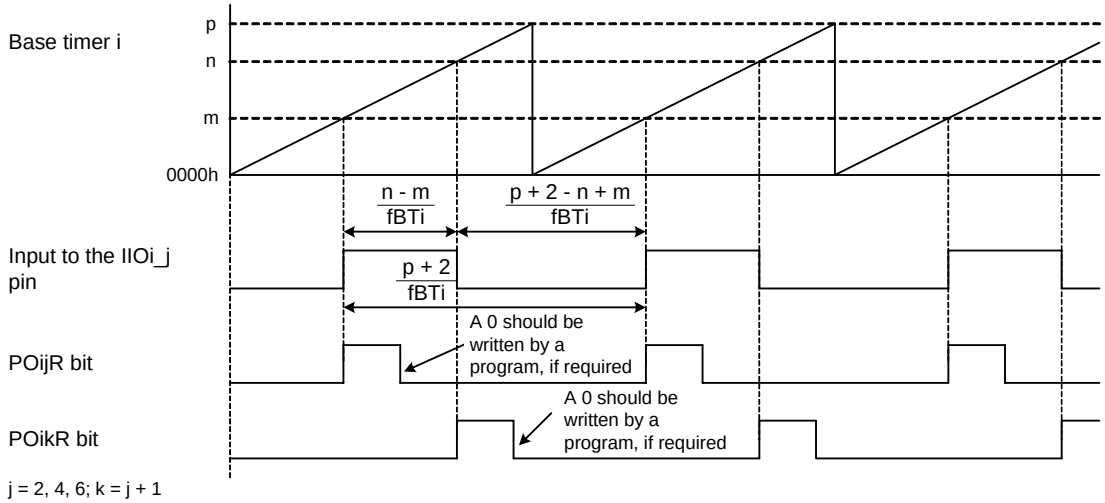



j = 0, 2, 4, 6; k = j + 1
m: GiPOj register setting value, 0000h to FFFFh
n: GiPOk register setting value, 0000h to FFFFh
POijR bit, POikR bit: Bits in registers IIO0IR to IIO11IR

Notes:
1. Output waveform when the INV bit in the GiPOCRj register is set to 0 (output is not inverted) and the IVL bit is set to 0 (low level is output as default value).
2. Output waveform when the INV bit is set to 0 (output is not inverted) and the IVL bit is set to 1 (high level is output as default value).

This figure above applies under the following conditions:
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)
- m < n

Case 2: The base timer is reset by matching the base timer value with the GiPO0 register setting (Bits RST2 to RST0 in the GiBCR1 register are set to 010b)




j = 2, 4, 6; k = j + 1
m: GiPOj register setting value, 0000h to FFFFh
n: GiPOk register setting value, 0000h to FFFDh
p: GiPO0 register setting value, 0001h to FFFDh
POijR bit, POikR bit: Bits in registers IIO0IR to IIO11IR

This figure above applies under the following conditions:
- The IVL bit is set to 0 (low level is output as default value) and the INV bit is set to 0 (output is not inverted).
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)
- m < n < p + 2

**Figure 23.27 SR Waveform Output Mode Operation (i = 0 to 2)**

### 23.3.4 Bit Modulation PWM Output Mode (for Group 2)

In bit modulation PWM output mode, a PWM output has a 16-bit resolution.
Pulses are output in repetitive cycles, each cycle consisting of span t repeated 1024 times. The span t has a cycle of $\frac{64}{fBT2}$. The six upper bits in the G2POj register (j = 0 to 7) determine the base low width.
The ten lower bits determine the number of span t, within a cycle, in which low width is extended by the minimum resolution bit width, that is, 1 clock cycle.
If the INV bit is set to 1 (output level is inverted), the waveform with inverted level is output.
Table 23.10 lists specifications of bit modulation PWM output mode. Table 23.11 lists the number of modulated spans and span ts to be extended with the minimum resolution bit width. Figure 23.28 shows an example of bit modulation PWM output mode operation.

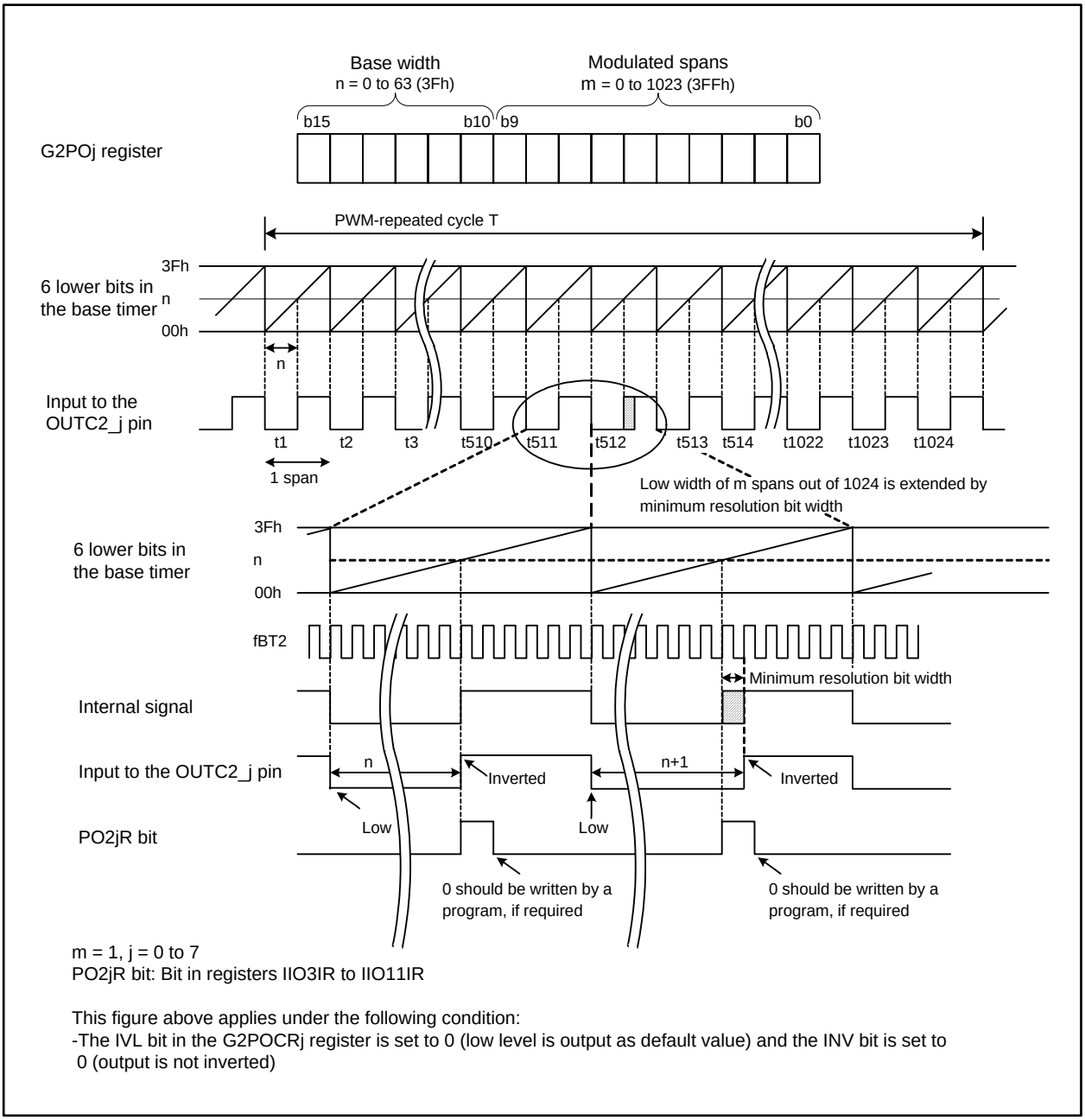
**Table 23.10 Bit Modulation PWM Output Mode Specifications**

| Item | Specification |
|---|---|
| Output waveform (1, 2) | PWM-repeated cycle T: $\frac{65536}{fBT2}$ (= $\frac{64}{fBT2}$ × 1024)<br>Cycle of span t: $\frac{64}{fBT2}$<br>Low width: $\frac{n+1}{fBT2}$ of m spans<br>$\frac{n}{fBT2}$ of (1024-m) spans<br>Mean low width: $\frac{1}{fBT2}$ × (n + $\frac{m}{1024}$)<br>n: G2POj register (j = 0 to 7) setting value (6 upper bits), 00h to 3Fh<br>m: G2POj register (j = 0 to 7) setting value (10 lower bits), 00h to 3FFh |
| Waveform output start condition | The IFEj bit in the G2FE register (j = 0 to 7) is set to 1 (the function for channel j is enabled) |
| Waveform output stop condition | The IFEj bit is set to 0 (the function for channel j is disabled) |
| Interrupt request | When the PO2jR bit in the interrupt request register is set to 1 (interrupts requested) by matching the 6 lower bits of the base timer value with the 6 upper bits of the G2POj register setting (Refer to Figure 11.12) |
| OUTC2_j pin function | Pulse signal output pin |
| Selectable functions | • Default value setting<br>This function determines the starting waveform output level<br>• Output level inversion<br>This function inverts the waveform output level and output the inverted signal from the OUTC2_j pin |

Notes:
1. Bits RST2 and RST0 in the G2BCR1 register should be set to 000b to use bit modulation PWM output mode.
2. When the INV bit in the G2POCRj register is set to 1 (output level is inverted), widths low and high are inverted.

**Table 23.11 Number of Modulated Spans and Span t Extended Minimum Resolution Bit Width**

| Modulated Spans | Span ts to be Extended with Minimum Resolution Bit Width |
|---|---|
| 00 0000 0000b | none |
| 00 0000 0001b | t512 |
| 00 0000 0010b | t256 and t768 |
| 00 0000 0100b | t128, t384, t640, and t896 |
| 00 0000 1000b | t64, t192, t320, t448, t576, t704, t832, and t960 |
| : | : |
| 10 0000 0000b | t1, t3, t5, t7, ••• t1019, t1021, and t1023 |




m = 1, j = 0 to 7
PO2jR bit: Bit in registers IIO3IR to IIO11IR

This figure above applies under the following condition:
-The IVL bit in the G2POCRj register is set to 0 (low level is output as default value) and the INV bit is set to 0 (output is not inverted)

**Figure 23.28 Bit Modulation PWM Output Mode Operation**

### 23.3.5 Real-Time Port Output Mode (RTP Output Mode) (for Group 2)

The OUTC2_j pin (j = 0 to 7) outputs the G2RTP register setting value in one-bit units when the base timer value matches the G2POj register setting. Table 23.12 lists specifications of RTP output mode. Figure 23.29 shows a block diagram of RTP output and Figure 23.30 shows an example of RTP output mode operation.

**Table 23.12 RTP Output Mode Specifications**

| Item | Specification |
|---|---|
| Waveform output start condition | The IFEj bit (j = 0 to 7) in the G2FE register is set to 1 (the function for channel j is enabled) |
| Waveform output stop condition | The IFEj bit is set to 0 (the function for channel j is disabled) |
| Interrupt request | When the PO2jR bit in the interrupt request register is set to 1 (interrupts requested) by matching the base timer value with the G2POj register setting (0000h to FFFFh (1)) (Refer to Figure 11.12) |
| OUTC2_j pin function | RTP output pin |
| Selectable functions | • Default value setting<br>This function determines the starting waveform output level<br>• Output level inversion<br>This function inverts the waveform output level and output the inverted signal from the OUTC2_j pin |

Note:
1. The G2PO0 register should be set to between 0001h and FFFDh to set the base timer value to 0000h (bits RST2 to RST0 are set to 010b) when the base timer value matches the G2PO0 register setting.

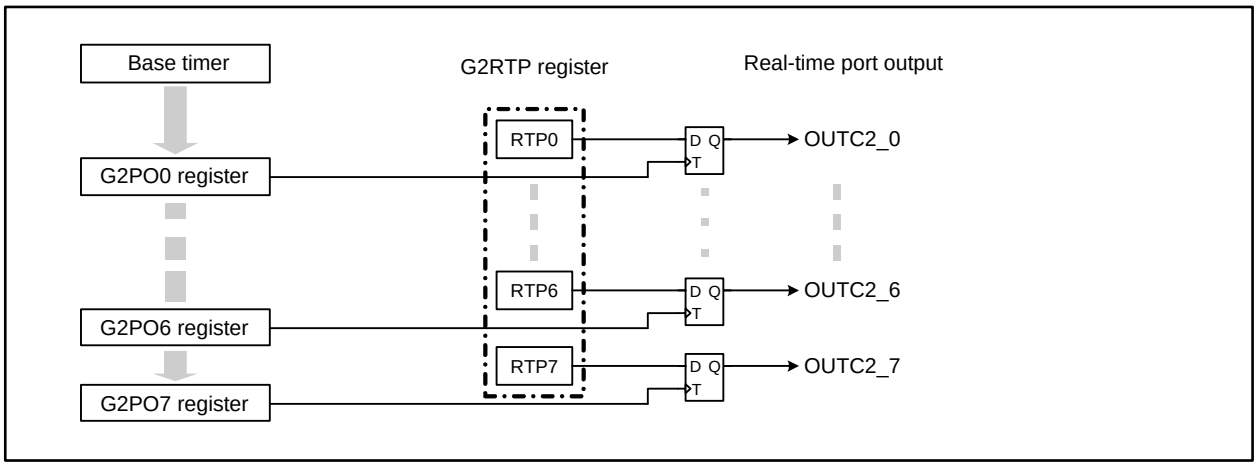
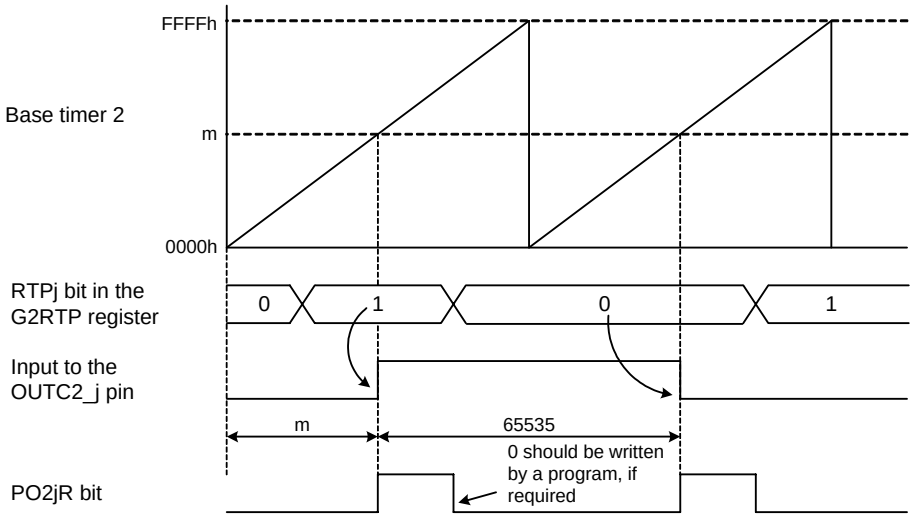



**Figure 23.29 RTP Output Block Diagram**

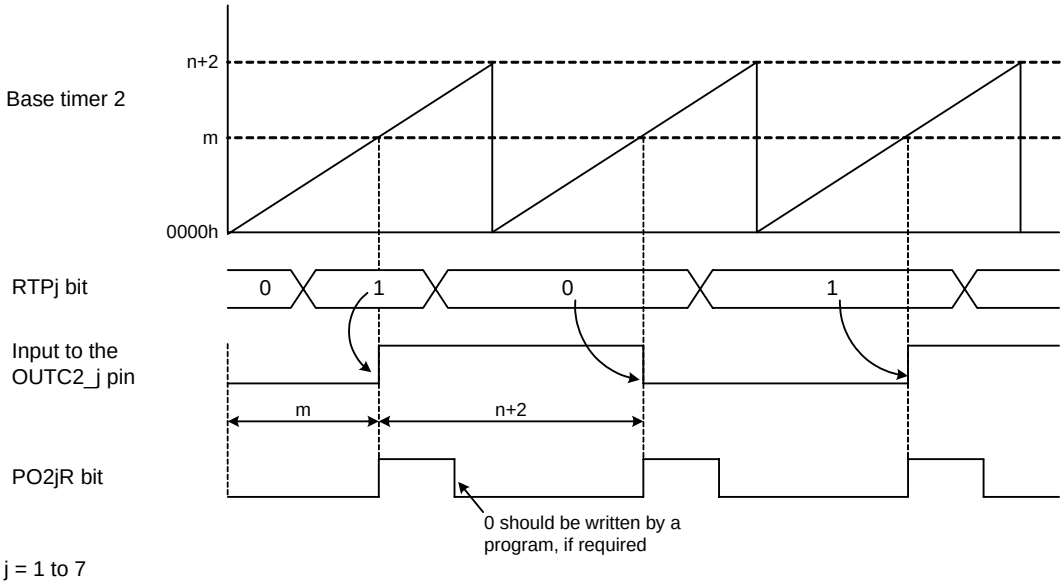
Case 1: Free-running operation (Bits RST2 to RST0 in the G2BCR1 register are set to 000b)




j = 0 to 7
m: G2POj register setting value, 0000h to FFFFh
PO2jR bit: Bit in registers IIO03R to IIO11IR

This figure above applies under the following conditions:
-The IVL bit in the G2POCRj register is set to 0 (low level is output as default value) and the INV bit is set to "0" (output is not inverted).
-Bits RST2 to RST0 in the G2BCR1 register are set to 000b (base timer is not reset)

Case 2: The base timer is reset by matching the base timer value with the G2PO0 register setting (Bits RST2 to RST0 are set to 010b)




j = 1 to 7
m: G2POj register setting value, 0000h to FFFFh
n: G2POj register setting value, 0001h to FFFDh
PO2jR bit: Bit in registers IIO03R to IIO11IR

This figure above applies under the following conditions:
-The IVL bit in the G2POCRj register is set to 0 (low level is output as default value) and the INV bit is set to 0 (output level is not inverted)
-m < n + 2

**Figure 23.30 RTP Output Mode Operation**

### 23.3.6 Parallel Real-Time Port Output Mode (RTP Output Mode) (for Group 2)

The OUTC2_j pin (j = 0 to 7) outputs all the G2RTP register setting values in one-byte units when the base timer value matches the G2POj register setting. Table 23.13 lists specifications of parallel RTP output mode. Figure 23.7 shows the G2BCR1 register. Figure 23.31 shows a block diagram of parallel RTP output and Figure 23.32 shows an example of parallel RTP output mode operation.

**Table 23.13 Parallel RTP Output Mode Specifications**

| Item | Specification |
|---|---|
| Waveform output start condition | The IFEj bit (j = 0 to 7) in the G2FE register is set to 1 (the function for channel j is enabled) |
| Waveform output stop Condition | The IFEj bit is set to 0 (the function for channel j is disabled) |
| Interrupt request | When the PO2jR bit in the interrupt request register is set to 1 (interrupts requested) by matching the base timer value with the G2POj register setting (0000h to FFFFh (1)) (Refer to Figure 11.12) |
| OUTC2_j pin function | RTP output pin |
| Selectable functions | • Default value setting<br>This function determines the starting waveform output level<br>• Output level inversion<br>This function inverts the waveform output level and output the inverted signal from the OUTC2_j pin |

Note:
1. The G2PO0 register should be set to between 0001h and FFFDh to set the base timer value to 0000h (bits RST2 to RST0 are set to 010b) when the base timer value matches the G2PO0 register setting.

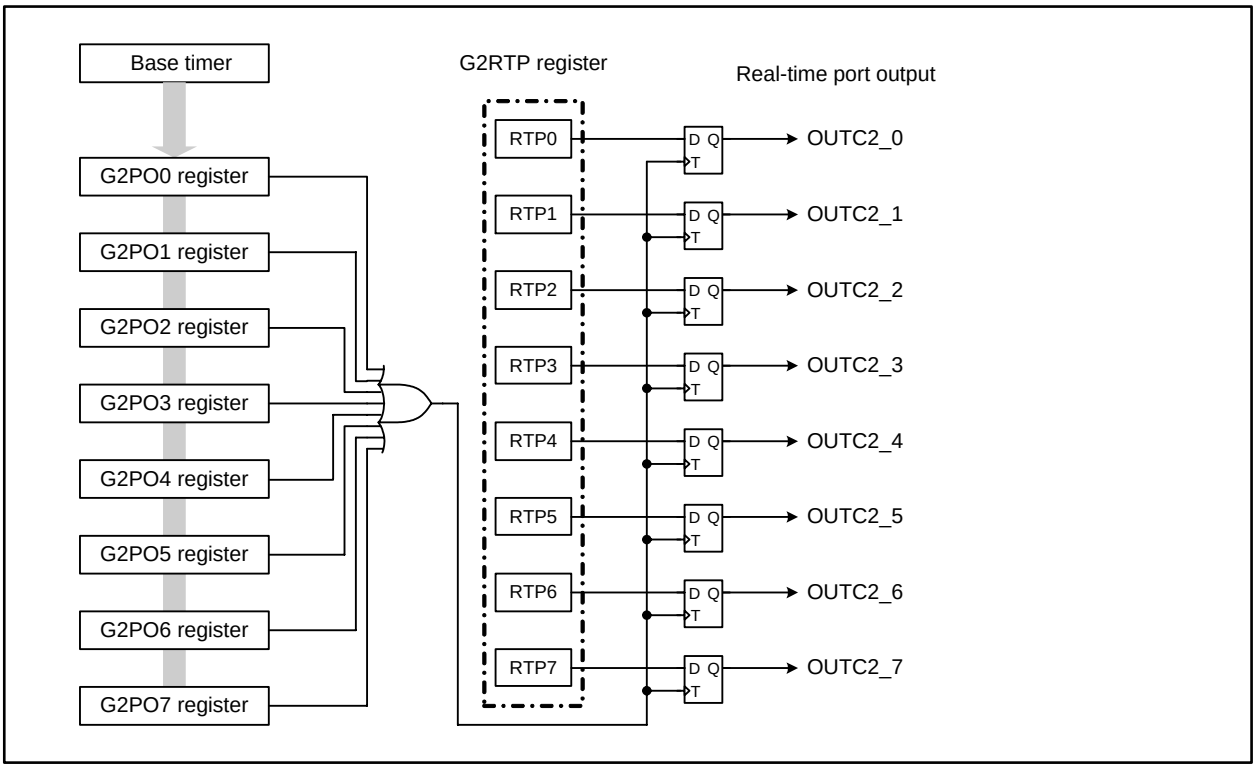
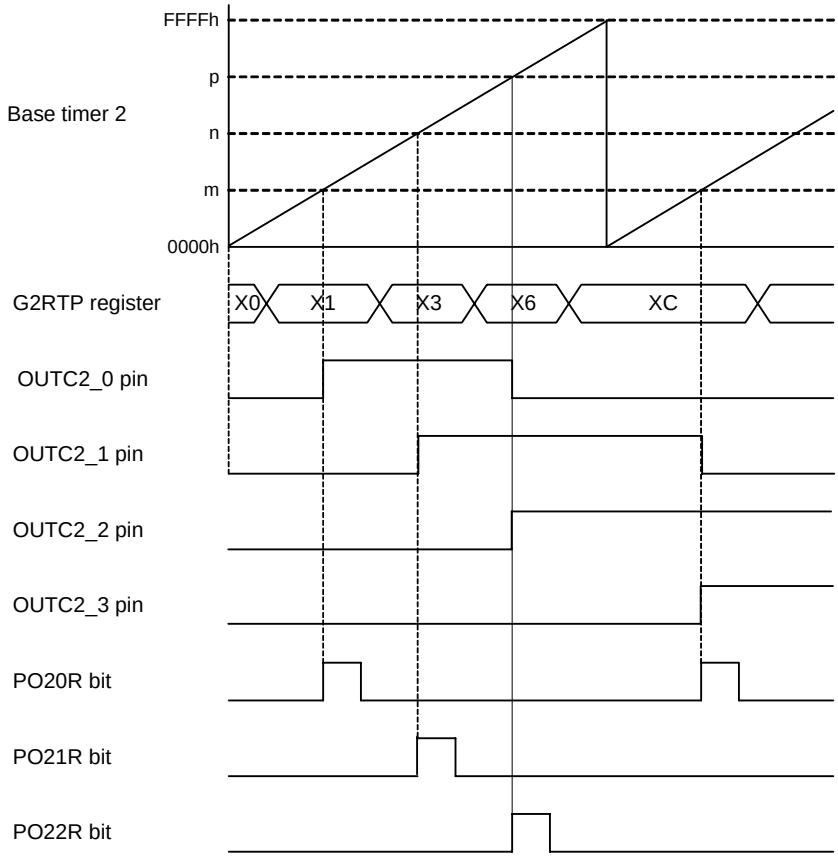



**Figure 23.31 Parallel RTP Output Mode Block Diagram**

Case 1: Free-running operation




m: G2PO0 register setting value, 0000h to FFFFh
n: G2PO1 register setting value, 0000h to FFFFh
p: G2PO2 register setting value, 0000h to FFFFh
Bits PO20R, PO21R, and PO22R: Bits in registers IIO3IR to IIO11IR

This figure above applies under the following conditions:
-The IVL bit in the G2POCRj register is set to 0 (low level is output as default value) and the IVL bit is set to 0 (output level is not inverted)
-Bits RST2 to RST0 in the G2BCR1 register are set to 000b (base timer is not reset)
-m < n < p

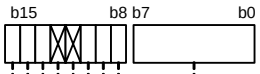
**Figure 23.32 Parallel RTP Output Mode Operation**

## 23.4 Group 2 Serial Interface

Two 8-bit shift registers and waveform generation enable the serial interface function. In group 2 of the intelligent I/O, the variable synchronous serial interface and IEBus (1) (optional (2)) are available. Figure 23.33 to Figure 23.40 show associated registers.

Notes:
1. IEBus is a trademark of NEC Electronics Corporation.
2. Please contact a Renesas sales office to use the optional feature.

### Group 2 SI/O Transmit Buffer Register



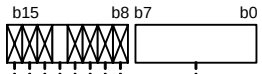

| Symbol | Address | Reset Value |
|---|---|---|
| G2TB | 016Dh-016Ch | Undefined |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b7-b0) | Transmit Buffer | Data transmitted | WO |
| SZ0 | Transmit/Receive Character Length Select Bit | b10 b9 b8<br>0 0 0 : 8 bits<br>0 0 1 : 1 bit<br>0 1 0 : 2 bits<br>0 1 1 : 3 bits<br>1 0 0 : 4 bits<br>1 0 1 : 5 bits<br>1 1 0 : 6 bits<br>1 1 1 : 7 bits | RW |
| SZ1 | | | RW |
| SZ2 | | | RW |
| — (b12-b11) | No register bits; should be written with 0 and read as undefined value | | — |
| A | ACK Function Select Bit | 0: Do not add the ACK bit<br>1: Add the ACK bit after last transmit bit | RW |
| PC | Parity Calculation Continuing Bit | 0: Add the parity bit after this transmit data<br>1: Carry over a parity to the data to be transmitted (1) | RW |
| P | Parity Select Bit | 0: No parity<br>1: Parity (even parity only) | RW |

Note:
1. The PC bit should be set to 1 after the P bit is set to 0.

**Figure 23.33 G2TB Register**

## Group 2 SI/O Receive Buffer Register



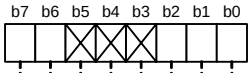

| Symbol | Address | Reset Value |
|---|---|---|
| G2RB | 016Fh-016Eh | Undefined |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b7-b0) | Receive Buffer | Data received | RO |
| — (b11-b8) | No register bits; should be written with 0 and read as undefined value | | — |
| OER | Overrun Error Flag (1) | 0: No overrun error<br>1: Overrun error | RO |
| — (b15-b13) | No register bits; should be written with 0 and read as undefined value | | — |

Note:
1. The OER bit is set to 0 when bits GMD1 and GMD0 in the G2MR register are set to 00b (the communication block is reset) or the RE bit in the G2CR register is set to 0 (reception disabled).

**Figure 23.34 G2RB Register**

## Group 2 Serial Interface Mode Register



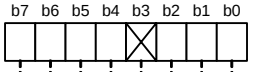

| Symbol | Address | Reset Value |
|---|---|---|
| G2MR | 016Ah | 00XX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| GMD0 | Serial Interface Mode Select Bit | b1 b0<br>0 0 : The communication block is reset (the OER bit is set to 0) (1)<br>0 1 : Variable synchronous serial interface mode (2)<br>1 0 : IEBus mode (2)<br>1 1 : Do not use this combination | RW |
| GMD1 | | | RW |
| CKDIR | Internal/External Clock Select Bit | 0: Internal clock<br>1: External clock | RW |
| — (b5-b3) | No register bits; should be written with 0 and read as undefined value. | | — |
| UFORM | Bit Order Select Bit | 0: LSB first<br>1: MSB first | RW |
| IRS | Transmit Interrupt Source Select Bit | 0: Transmit buffer is empty<br>1: Transmission is completed | RW |

Notes:
1. One base timer clock or more is required after bits GMD1 and GMD0 are set to 00b.
2. Bits GMD1 and GMD0 should be set to 01b or 10b while the base timer clock is stopped.

**Figure 23.35 G2MR Register**

## Group 2 Serial Interface Control Register



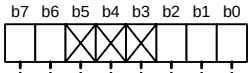

| Symbol | Address | Reset Value |
|---|---|---|
| G2CR | 016Bh | 0000 X110b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TE | Transmit Enable Bit | 0: Transmission disabled<br>1: Transmission enabled | RW |
| TXEPT | Transmit Shift Register Empty Flag | 0: Data in the transmit shift register (during transmission)<br>1: No data in the transmit shift register (transmission completed) | RO |
| TI | Transmit Buffer Empty Flag | 0: Data held in the G2TB register<br>1: No data held in the G2TB register | RO |
| — (b3) | No register bit; should be written with 0 and read as undefined value | | — |
| RE | Receive Enable Bit (1) | 0: Reception disabled<br>1: Reception enabled | RW |
| RI | Receive Complete Flag | 0: No data held in the G2RB register<br>1: Data held in the G2RB register | RO |
| OPOL | ISTxD Output Polarity Switching Bit | 0: Not inverted<br>1: Inverted | RW |
| IPOL | ISRxD Input Polarity Switching Bit (1) | 0: Not inverted<br>1: Inverted | RW |

Note:
1. The group 2 base timer may be reset when these bits are rewritten. To avoid unexpected resets, the RST2 bit in the G2BCR1 register should be set to 0 (base timer is not reset by a reset request from the serial interface).

**Figure 23.36 G2CR Register**

## Group 2 IEBus Control Register



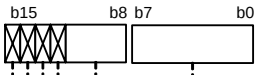

| Symbol | Address | Reset Value |
|---|---|---|
| IECR | 0172h | 00XX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IEB | IEBus Enable Bit (1) | 0: IEBus disabled (2)<br>1: IEBus enabled | RW |
| IETS | IEBus Transmit Start Request Bit | 0: Transmission completed<br>1: Transmission started | RW |
| IEBBS | IEBus Bus Busy Flag | 0: Idle state<br>1: Busy state (start condition detected) | RO |
| — (b5-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| DF | Digital Filter Select Bit | 0: No digital filter<br>1: Use the digital filter | RW |
| IEM | IEBus Mode Select Bit | 0: Mode 1<br>1: Mode 2 | RW |

Notes:
1. The IEB bit should be rewritten while the base timer clock is stopped.
2. One fBT2 cycle or more is required after the IEB bit is set to 0. To set this bit to 1, bits BCK1 and BCK0 in the G2BCR0 register should be set to 00b (clock is stopped).

**Figure 23.37 IECR Register**

## Group 2 IEBus Address Register

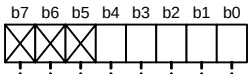



| Symbol | Address | Reset Value |
|---|---|---|
| IEAR | 0171h-0170h | Undefined |

| Function | RW |
|---|---|
| Address data | RW |
| Address data | RW |
| No register bits; should be written with 0 and read as undefined value. | — |

**Figure 23.38 IEAR Register**

## Group 2 IEBus Transmit Interrupt Source Detect Register



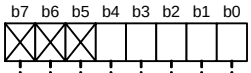

| Symbol | Address | Reset Value |
|---|---|---|
| IETIF | 0173h | XXX0 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IETNF | Normal Completion Flag | 0: Transmission is completed in error<br>1: Transmission is successfully completed (1) | RW |
| IEACK | ACK Error Flag | 0: No error detected<br>1: Error detected (1) | RW |
| IETMB | Maximum Transmit Byte Error Flag | 0: No error detected<br>1: Error detected (1) | RW |
| IETT | Timing Error Flag | 0: No error detected<br>1: Error detected (1) | RW |
| IEABL | Arbitration Lost Flag | 0: No error detected<br>1: Error detected (1) | RW |
| —<br>(b7-b5) | No register bits; should be written with 0 and read as undefined value | | — |

Note:
1. This bit can be set to 0 by a program but cannot be set to 1. To set this bit to 0, the IEB bit in the IECR register should be set to 0 (IEBus disabled).

**Figure 23.39 IETIF Register**

## Group 2 IEBus Receive Interrupt Source Detect Register




| Symbol | Address | Reset Value |
|---|---|---|
| IERIF | 0174h | XXX0 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IERNF | Normal Completion Flag | 0: Reception is completed in error<br>1: Reception is successfully completed (1) | RW |
| IEPAR | Parity Error Flag | 0: No error detected<br>1: Error detected (1) | RW |
| IERMB | Maximum Receive byte Error Flag | 0: No error detected<br>1: Error detected (1) | RW |
| IERT | Timing Error Flag | 0: No error detected<br>1: Error detected (1) | RW |
| IERETC | Error by Other Sources Flag | 0: No error detected<br>1: Error detected (1) | RW |
| —<br>(b7-b5) | No register bits; should be written with 0 and read as undefined value | | — |

Note:
1. This bit can be set to 0 by a program but cannot be set to 1. To set this bit to 0, the IEB bit in the IECR register should be set to 0 (IEBus disabled).

**Figure 23.40 IERIF Register**

### 23.4.1 Variable Synchronous Serial Interface Mode (for Group 2)

This mode allows 1-bit to 8-bit data transmission/reception synchronized with the transmit/receive clock. The character length is selectable from 1 to 8 bits. Table 23.14 lists specifications of the group2 variable synchronous serial interface mode and Table 23.15 lists its settings. Figure 23.41 shows an operation example of data transmission/reception.

**Table 23.14 Group 2 Variable Synchronous Serial Interface Mode Specifications**

| Item | Specification |
|---|---|
| Data format | 1- to 8-bit character length |
| Transmit/receive clock | • The CKDIR bit in the G2MR register is set to 0 (internal clock selected): $\frac{fBT2}{2(n+2)}$ n: G2PO0 register setting value, 0000h to FFFFh (1) The bit rate is set using the G2PO0 register. The clock is generated in the inverted waveform output mode of the channel 2 waveform generation<br>• The CKDIR bit is set to 1 (external clock selected): input into the ISCLK2 pin (2) |
| Transmit start conditions | The conditions for starting data transmission are as follows:<br>• The TE bit in the G2CR register is set to 1 (transmission enabled)<br>• The TI bit in the G2CR register is set to 0 (data held in the G2TB register) |
| Receive start conditions | The conditions for starting data reception are as follows:<br>• The RE bit in the G2CR register is set to 1 (reception enabled)<br>• The TE bit in the G2CR register is set to 1 (transmission enabled)<br>• The TI bit in the G2CR register is set to 0 (data held in the G2TB register) |
| Interrupt request | In transmit interrupt, either of the following conditions is selected to set the SIO2TR bit in the IIO6IR register to 1 (interrupts requested) (Refer to Figure 11.12):<br>• The IRS bit in the G2MR register is set to 0 (transmit buffer in the G2TB register is empty): when data is transferred from the G2TB register to the transmit shift register (when the transmission has started)<br>• The IRS bit is set to 1 (transmission is completed): when data transmission from the transmit shift register is completed<br>In receive interrupt,<br>When data is transferred from the receive shift register to the G2RB register (when the reception is completed), the SIO2PR bit in the IIO5IR register is set to 1 (interrupts requested) (Refer to Figure 11.12) |
| Error detection | Overrun error (3)<br>This error occurs when the last bit of the next data has been received before reading the G2RB register |
| Selectable functions | • Bit order selection<br>Selectable either LSB first or MSB first<br>• ISTXD2 and ISRXD2 I/O polarity<br>Output level from the ISTXD2 pin and input level to the ISRXD2 pin can be respectively inverted<br>• Character length for data transmission/reception<br>Selectable a character length from 1 to 8 bits |

Notes:
1. When using the serial interface, set 1 or above to the G2PO0 register.
2. The highest transmit/receive clock frequency should be fBT2 divided by 20.
3. If an overrun error occurs, the G2RB register is undefined.

**Table 23.15 Register Settings in Group2 Variable Synchronous Serial Interface Mode**

| Register | Bits | Function |
|---|---|---|
| G2BCR0 | BCK1 to BCK0 | Set the bits to 11b |
| | DIV4 to DIV0 | Select a divide ratio of count source |
| | IT | Set the bit to 0 |
| G2BCR1 | 7 to 0 | Set the bits to 0001 0010b |
| G2POCR0 | 7 to 0 | Set the bits to 0000 0111b |
| G2POCR1 | 7 to 0 | Set the bits to 0000 0111b |
| G2POCR2 | 7 to 0 | Set the bits to 0000 0010b |
| G2PO0 | 15 to 0 | Set a comparative value for waveform generation<br>$\frac{fBT2}{2 \times (\text{setting value} + 2)}$ = transmit/receive clock frequency |
| G2PO2 | 15 to 0 | Set to a value smaller than that in the G2PO0 register setting |
| G2FE | IFE2 to IFE0 | Set the bits to 111b |
| G2MR | GMD1 to GMD0 | Set the bits to 01b |
| | CKDIR | Select either the internal clock or the external clock |
| | UFORM | Select either LSB first or MSB first |
| | IRS | Select a source for transmit interrupt |
| G2CR | TE | Set the bit to 1 to enable data transmission/reception |
| | TXEPT | Transmit shift register empty flag |
| | TI | Transmit buffer empty flag |
| | RE | Set the bit to 1 to enable data reception |
| | RI | Receive complete flag |
| | OPOL | Select if the output level at the ISTXD2 pin is inverted (usually set the bit to 0) |
| | IPOL | Select if the input level at the ISRX2 pin is inverted (usually set the bit to 0) |
| G2TB | 15 to 0 | Set the data to be transmitted/received and its character length |
| G2RB | 15 to 0 | Store received data and error flag |

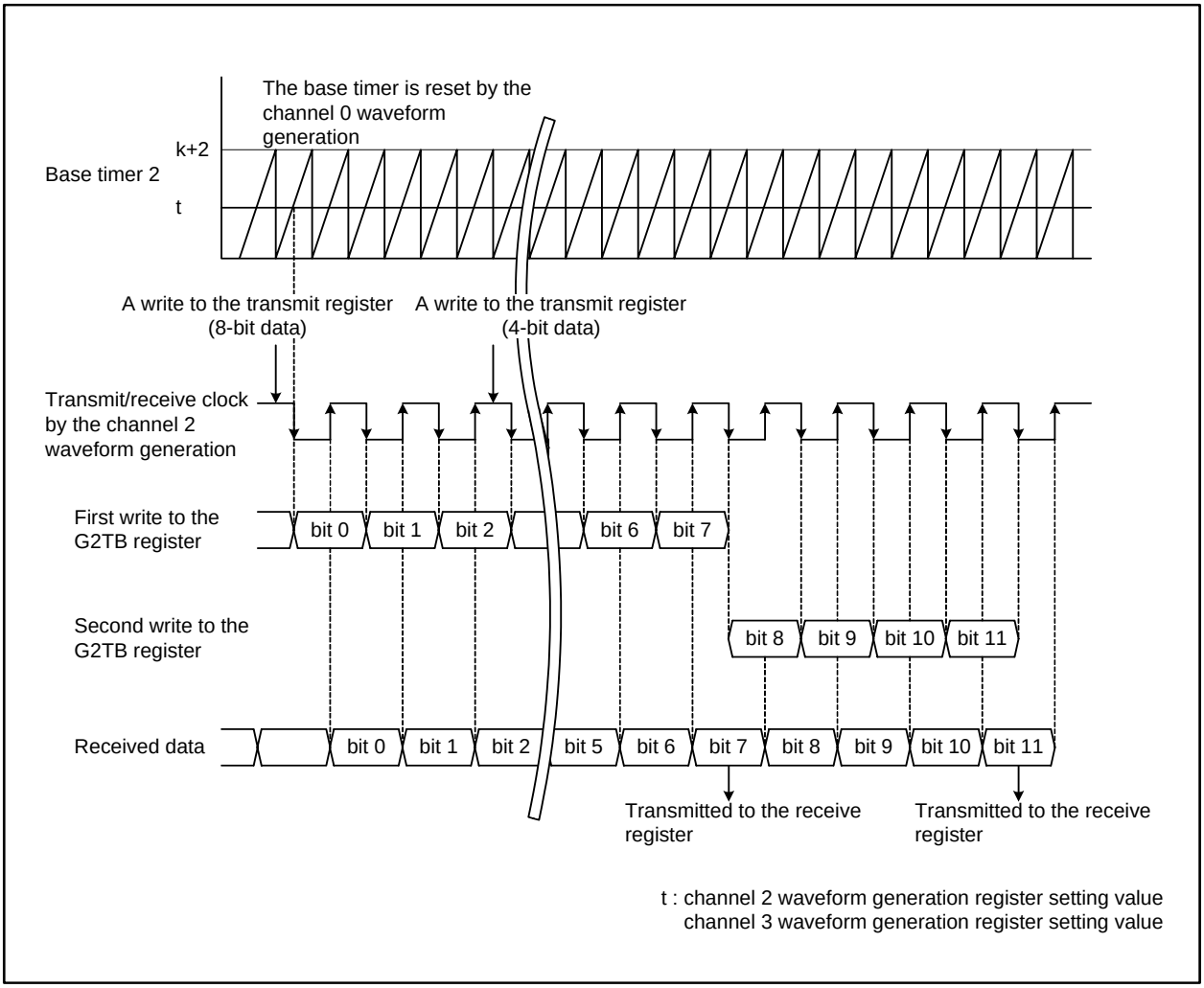



**Figure 23.41 Group 2 Variable Synchronous Serial Interface Mode Transmit/Receive Operation**

# 24. Multi-master I2C-bus Interface

The multi-master I2C-bus interface (MMI2C) is capable of serial, bi-directional data transfer in the I2C-bus data transmit and receive format. It contains an arbitration lost detector and a clock synchronization function. Table 24.1 lists specifications of the multi-master I2C-bus interface. Table 24.2 lists detectors of the multi-master I2C-bus interface. Figure 24.1 shows a block diagram of the multi-master I2C-bus interface.

**Table 24.1 Specifications of Multi-master I2C-bus Interface**

| Item | Specification |
|---|---|
| Data format | Compliant with the I2C-bus specification<br>• 7-bit addressing format<br>• Fast-mode<br>• Standard-mode |
| Master/Slave device | Selectable |
| I/O pins | Serial data line: MSDA (SDA)<br>Serial clock line: MSCL (SCL) |
| Transmit/Receive clock | 16.1 kbps to 400 kbps (ϕIIC = 4 MHz) ϕIIC: I2C-bus system clock |
| Transmit/Receive modes | Compliant with the I2C-bus specification<br>• Master-transmit mode<br>• Master-receive mode<br>• Slave-transmit mode<br>• Slave-receive mode |
| Interrupt request sources | • Six I2C-bus interface interrupts: Successful transmit, successful receive, slave address match detection, general call detection, STOP condition detection, and timeout detection<br>• Two I2C-bus line interrupts: Rising or falling edge of pins MSDA and MSCL |
| Selectable functions | • Timeout detector<br>This function detects the MSCL pin level is held high for longer than a specified time while the bus is busy.<br>• Free data format selector<br>This function selects the free data format to generate an interrupt request, regardless of the slave address value, when the first byte is received |

**Table 24.2 Detectors of Multi-master I2C-bus Interface**

| Item | Specification |
|---|---|
| Slave-address match detector | In slave transmit/receive mode, this detects whether the address sent from the master device matches the slave-address. When they match, ACK is automatically sent. When they don't, a NACK is automatically sent and communication is stopped |
| General call detector | This detects a general call when in slave-receive mode. |
| Arbitration lost detector | This detects that the master lost the arbitration and switches off the output to the MSDA pin |
| Bus is busy detector | This detects that the bus is busy, and sets the BBSY flag |

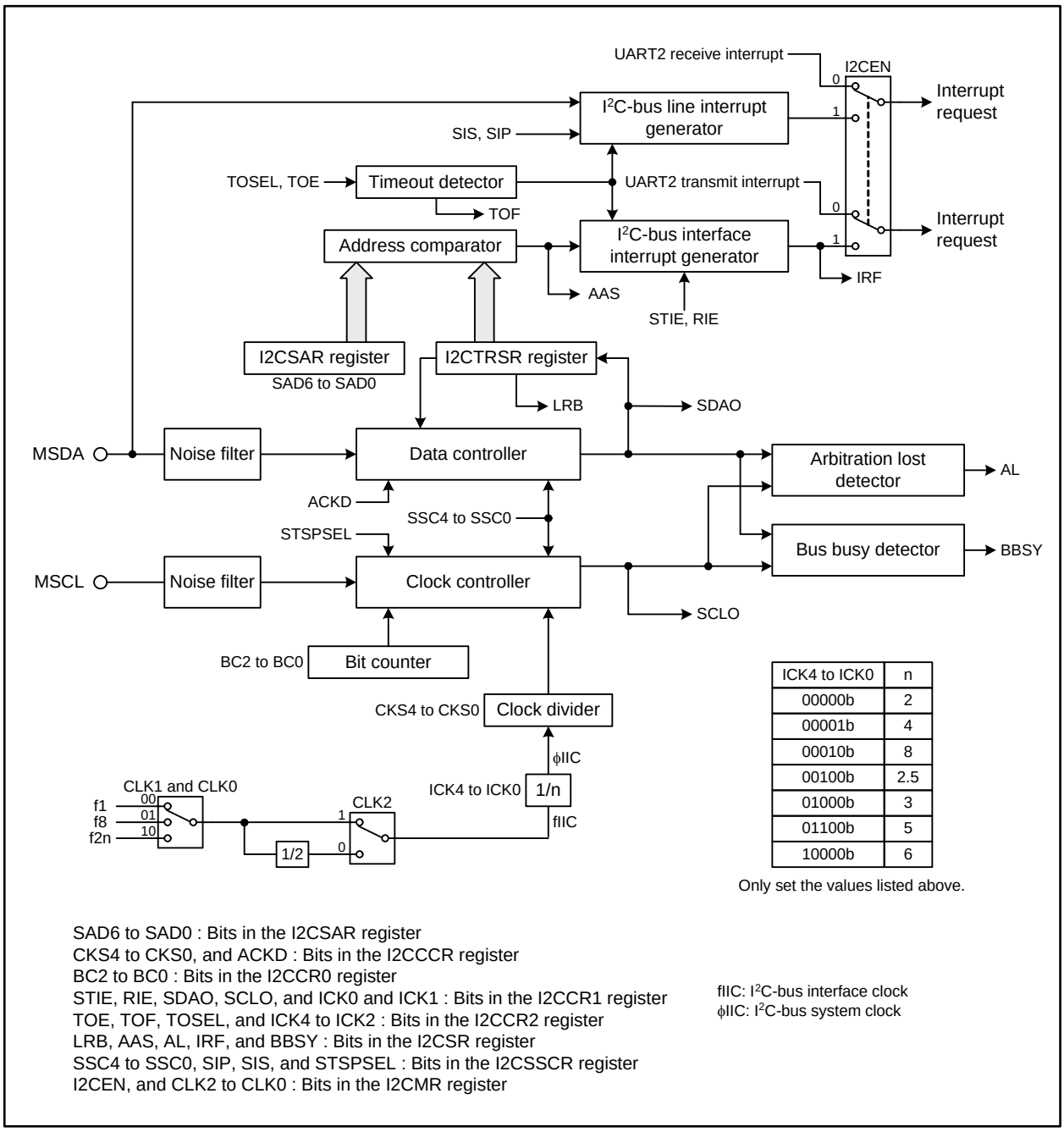



**Figure 24.1 Multi-master I2C-bus Interface Block Diagram**

## 24.1 Multi-master I2C-bus Interface-associated Registers

### 24.1.1 I2C-bus Transmit/Receive Shift Register (I2CTRSR)

I2C-bus Transmit/Receive Shift Register (1, 2, 3)

b7 b0

| Symbol | Address | Reset Value |
|---|---|---|
| I2CTRSR | 044400h | Undefined |

| Function | RW |
|---|---|
| Transmit data is written to this register in transmit mode.<br>In receive mode, dummy data is written to this register, then received data is read after and interrupt is generated.<br>The data currently being shifted can be read from this register, regardless of whether it is transmitting or receiving | RW |

Notes:
1. This register is writable when the ICE bit in the I2CCR0 register is 1 (I2C-bus interface enabled)
2. This register is used for transmitting and receiving. Read the received data before using this register for transmitting.
3. When data is written to this register, bits BC2 to BC0 in the I2CCR0 register become 000b, and bits LRB, AAS, and AL in the I2CSR register become 0.

**Figure 24.2 I2CTRSR Register**

The I2CTRSR register is an 8-bit shift register where received data is stored and transmit data is written. When transmit data is written to this register, the data is synchronized with the SCL clock and shifted out in order from bit 7. Every time a bit is shifted out, the data is shifted to the left by one bit. When in receive operation, the data is synchronized with the SCL clock and stored in order starting from bit 0. One bit of data is shifted (to the left) for every bit that is input. Figure 24.3 shows the timing when the received data is stored to the I2CTRSR register.

The I2CTRSR register is writable when the ICE bit in the I2CCR0 register is 1 (I2C-bus interface enabled-bus interface enabled). When the ICE bit is 1 and the MST bit in the I2CSR register is 1 (master mode), writing data to the I2CTRSR register resets the bit counter and the SCL clock is output.

Write to the I2CTRSR register when a START condition is generated or the MSCL pin is low. The register is always readable.

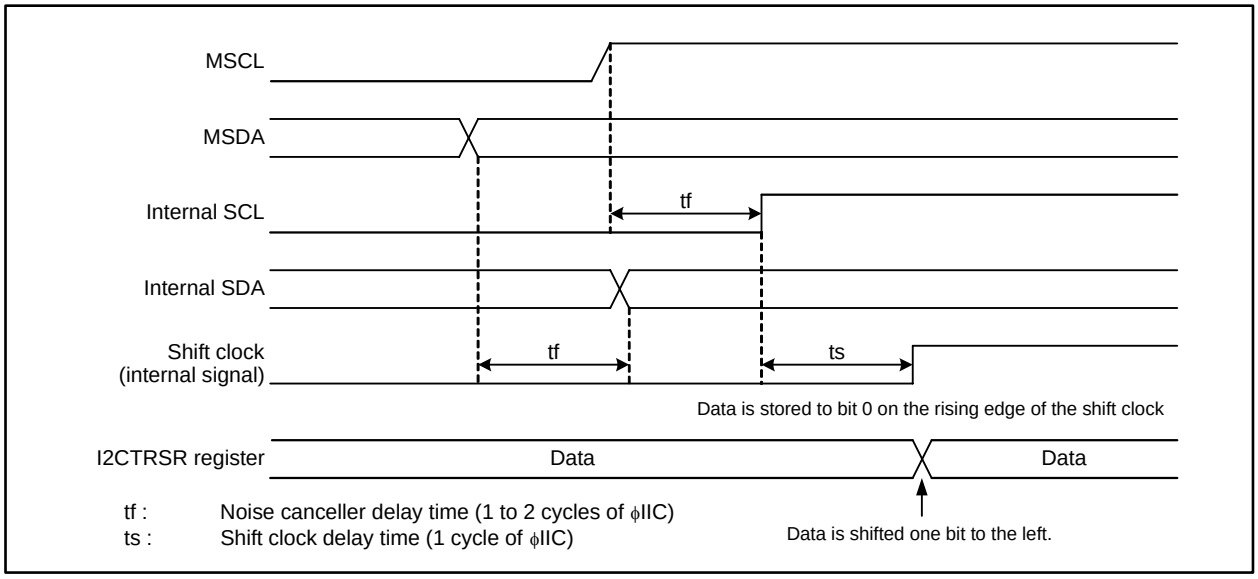



**Figure 24.3 Received Data Storing Timing to the I2CTRSR Register**

### 24.1.2 I2C-bus Slave Address Register (I2CSAR)

I2C-bus Slave Address Register

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | 0 |

Symbol: I2CSAR
Address: 044402h
Reset Value: 00h

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b0) | Reserved | Should be written with 0 | RW |
| SAD0 | Slave Address | The slave address must differ from other slave addresses of slave devices connected to the I2C-bus. In slave mode, the device becomes a slave device when the upper seven bits sent in the first frame after the START condition match bits SAD6 to SAD0 | RW |
| SAD1 | | | RW |
| SAD2 | | | RW |
| SAD3 | | | RW |
| SAD4 | | | RW |
| SAD5 | | | RW |
| SAD6 | | | RW |

**Figure 24.4 I2CSAR Register**

The I2CSAR register is used to store a slave address to automatically recognize itself as a slave device. When the received address matches the slave address, the device is activated as a slave device.

#### 24.1.2.1 SAD6 to SAD0

These seven bits are used to store a slave address. When the addressing format is enabled, the received 7-bit address and the slave address set in bits SAD6 to SAD0 are compared. When a match is detected, the device is activated as a slave device.

### 24.1.3 I2C-bus Control Register 0 (I2CCR0)

I2C-bus Control Register 0

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | | 0 | | | | | |

Symbol: I2CCR0 — Address: 044403h — Reset Value: 0000 0000b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| BC0 | Transmit/Receive Bit Number Set Bit (1) | b2 b1 b0<br>0 0 0 : 8 bits<br>0 0 1 : 7 bits<br>0 1 0 : 6 bits<br>0 1 1 : 5 bits<br>1 0 0 : 4 bits<br>1 0 1 : 3 bits<br>1 1 0 : 2 bits<br>1 1 1 : 1 bit | RW |
| BC1 | | | RW |
| BC2 | | | RW |
| ICE | I2C-bus Interface Enable Bit | 0: I2C-bus interface disabled<br>1: I2C-bus interface enabled | RW |
| DFS | Data Format Select Bit | 0: Addressing format<br>1: Free data format | RW |
| — (b5) | Reserved | Should be written with 0 | RW |
| RST | I2C-bus Interface Reset Bit | A write 1 to this bit resets the I2C-bus interface circuit | RW |
| — (b7) | Reserved | Should be written with 0 | RW |

Note:
1. These bits automatically become 000b in the following cases:
   - When a START or STOP condition is detected
   - When data transmission is completed, or
   - When data reception is completed

**Figure 24.5 I2CCR0 Register**

The I2CCR0 register is used to control data communication format.

#### 24.1.3.1 BC2 to BC0

Bits BC2 to BC0 are used to set the number of data bits to be sent or received next. When the number of data bits set by bits BC2 to BC0 (acknowledge clock pulse is included in the number, when the ACKCLK bit in the I2CCR register is 1) are sent or received, an I2C-bus interface interrupt request is generated. Consequently, bits BC2 to BC0 become 000b. Note that these bits also become 000b when a START condition is detected. Address data is sent or received in eight bits regardless of their setting.

#### 24.1.3.2 ICE bit

The ICE bit is used to enable the I2C-bus interface. Set this bit to 1 to enable the I2C-bus interface and 0 to disable it. When this bit is 0, pins MSDA and MSCL are fixed high (these pins are high-impedance when the corresponding NOD bits in registers P7_0S and P7_1S are 1), therefore the I2C-bus interface cannot be used.

When the ICE bit is set to 0, the following changes are made:

- Bits ADZ, AAS, AL, BBSY, TRS, and MST in the I2CSR register become 0, and the IRF bit becomes 1.
- Writing to the I2CTRSR register is disabled.
- The I2C-bus system clock (ϕIIC) is stopped, and the internal counter and the flag are reset.
- The TOF bit in the I2CCR2 register becomes 0 (timeout not detected).

#### 24.1.3.3 DFS bit

The DFS bit is used to enable the automatic recognition of slave address. When the DFS bit is set to 0, the addressing format is selected and the slave address data is automatically recognized. In this setting, data is received only when a general call is received or the slave address match is detected. When the DFS bit is set to 1, the free data format is selected. In this setting, the slave address is not recognized, so all data are received.

#### 24.1.3.4 RST bit

The RST bit is used to reset the I2C-bus interface when a communication error occurs. When the ICE bit in the I2CCR0 register is set to 1 (I2C-bus interface enabled), setting the RST bit to 1 (reset) has the following effect on the I2C-bus interface:

- Bits ADZ, AAS, AL, BBSY, TRS, and MST in the I2CSR register become 0 and the IRF bit becomes 1
- The TOF bit in the I2CCR2 register becomes 0 (timeout not detected)
- The internal counter and flag are reset

When the RST bit is set to 1, the multi-master I2C-bus interface is reset within a maximum of 2.5 ϕIIC cycles. Consequently, the RST bit automatically becomes 0.

Figure 24.6 shows the timing when the I2C-bus interface is reset.

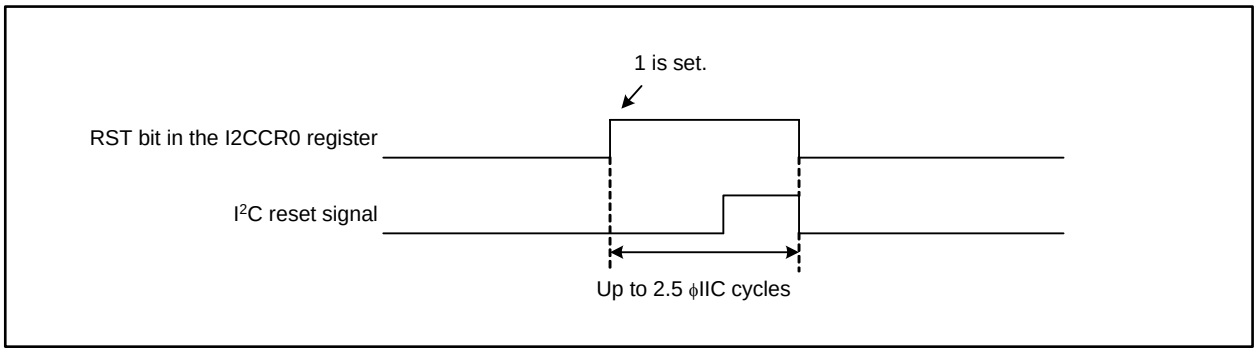



**Figure 24.6 I2C-bus Interface Reset Timing**

### 24.1.4 I2C-bus Clock Control Register (I2CCCR)

I2C-bus Clock Control Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: I2CCCR
Address: 044404h
Reset Value: 0000 0000b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CKS0 | Transmit/Receive Clock Frequency Control Bit | The transmit/receive clock frequency is given by ϕIIC/8n [Hz] in standard mode, or ϕIIC/4n [Hz] in fast-mode, where n is a setting value. However, when 00101b is set in fast-mode, the transmit/receive clock frequency becomes ϕIIC/2n [Hz]. Do not set to 00000b to 00010b | RW |
| CKS1 | | | RW |
| CKS2 | | | RW |
| CKS3 | | | RW |
| CKS4 | | | RW |
| CLKMD | Clock Mode Select Bit | 0: Standard-mode<br>1: Fast-mode | RW |
| ACKD | ACK Data Bit | 0: ACK sent<br>1: NACK sent | RW |
| ACKCLK | ACK Clock Generating Bit | 0: ACK clock not generated<br>1: ACK clock generated | RW |

**Figure 24.7 I2CCCR Register**

The register is used for ACK control and to set SCL mode and SCL clock frequency. While data is being transmitted or received, do not rewrite bits other than the ACKCLK bit.

#### 24.1.4.1 CKS4 to CKS0

Bits CKS4 to CKS0 are used to set the SCL clock frequency. The SCL clock frequency varies as shown in the Table 24.3, where n is a setting value of bits CKS4 to CKS0 (n = 3 to 31). Do not rewrite these bits while data is being transmitted or received.

**Table 24.3 I2CCCR Register Setting Values and SCL Frequencies**

| Bits CKS4 to CKS0 setting value (n) | SCL frequency (when ϕIIC = 4MHz) (1) | |
|---|---|---|
| | Standard-mode | Fast-mode |
| 0 to 2 | Do not set (2) | Do not set (2) |
| 3 | Do not set (3) | 333 kHz (ϕIIC/4n) |
| 4 | Do not set (3) | 250 kHz (ϕIIC/4n) |
| 5 | 100 kHz (ϕIIC/8n) | 400 kHz (ϕIIC/2n) (4) |
| 6 to 31 | 83 to 16 kHz (ϕIIC/8n) | 166 to 32 kHz (ϕIIC/4n) |

Notes:
1. The CKS value must be set so that the SCL clock frequency is below 100 kHz in standard-mode or below 400 kHz in fast-mode. The high period of the SCL clock has a margin of error of +2 to -4 ϕIIC in standard-mode, and +2 to -2 ϕIIC in fast-mode. Note that if the high period is shortened, the low period is lengthened, so the frequency remains unchanged.
2. Do not set the CKS value to 0 to 2 regardless of the ϕIIC frequency.
3. When ϕIIC is 4 MHz or above, do not set the CKS value to 3 or 4. The SCL clock frequency will extend beyond the specified range.
4. The normal duty cycle of SCL clock is 50%. When the CKS value is 5 in fast-mode, it varies from 35% to 45%.

### 24.1.4.2 CLKMD bit

The CLKMD bit is used to select the SCL mode. Set this bit to 0 to select standard-mode and to 1 for fast-mode. To use the device under the fast-mode I2C-bus specification (up to 400 kbit/s), set ϕIIO to be 4 MHz or higher.

### 24.1.4.3 ACKD bit

The ACKD bit is used to select the state of the MSDA pin when the ACK clock pulse is generated. When the ACKD bit is set to 0, the MSDA pin becomes low (acknowledged) by an ACK. When the ACKD bit is 1, the MSDA pin is held high even if an ACK clock pulse is generated.
Table 24.4 lists the MSDA pin level with the ACK clock.

**Table 24.4 MSDA Pin Levels with the ACK Clock**

| Received content | DFS bit | ACKD bit | Slave address | MSDA pin level |
|---|---|---|---|---|
| Slave Address | 0 | 0 | Match | Low (ACK) |
| | | | Mismatch | High (NACK) |
| | | 1 | — | High (NACK) |
| | 1 | 0 | — | Low (ACK) |
| | | 1 | — | High (NACK) |
| Data | — | 0 | — | Low (ACK) |
| | | 1 | — | High (NACK) |

### 24.1.4.4 ACKCLK bit

The ACKCLK bit is used to select whether to generate an ACK handshake. When this bit is 1 (ACK clock generated), the ACK clock pulse is generated after 1 byte of data is transmitted or received. When this bit is 0 (ACK clock not generated), the ACK clock is not generated after 1 byte of data is transmitted or received. In this case, the IR bit in the I2CIC register becomes 1 (I2C-bus interface interrupt requested) on the last falling edge of the clock for data transmission or reception.

### 24.1.5 I2C-bus START Condition and STOP Condition Control Register (I2CSSCR)

I2C-bus START Condition and STOP Condition Control Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: I2CSSCR
Address: 044405h
Reset Value: 00011010b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| SSC0 | START Condition and STOP Condition Detection Set Bit | The conditions of detecting START conditon and STOP condition (SCL open, set-up, and hold times) are set with these bits | RW |
| SSC1 | | | RW |
| SSC2 | | | RW |
| SSC3 | | | RW |
| SSC4 | | | RW |
| SIP | I2C Bus-line Interrupt Pin Edge Select Bit | 0: Falling edge<br>1: Rising edge | RW |
| SIS | I2C Bus-line Interrupt Pin Select Bit | 0: MSDA pin<br>1: MSCL pin | RW |
| STSPSEL | START Condition/STOP Condition Generating Mode Select Bit | 0: Short mode<br>1: Long mode | RW |

**Figure 24.8 I2CSSCR Register**

The I2CSSCR register is used to control the detection and generation of START condition and STOP condition.

#### 24.1.5.1 SSC4 to SSC0

Bits SSC4 to SSC0 are used to select the parameters of detecting the START condition and STOP condition by setting the high period of SCL pin, set-up, and hold times. This parameter is set by referencing the I2C-bus system clock (ϕIIC). Therefore, it will change according to the XIN frequency and the setting of the I2C-bus system clock select bits (i.e. bits ICK4 to ICK0 in registers I2CCR2 to I2CCR1). Do not set an odd number or 00000b to bits SSC4 to SSC0. To start the detection of START condition or STOP condition, set the ICE bit in the I2CCR0 register to 1 (I2C-bus interface enabled). Table 24.11 lists the recommended values for bits SSC4 to SSC0.

#### 24.1.5.2 SIP bit

The SIP bit is used to select which of the edges of MSCL or MSDA pin generates the I2C bus line interrupt. Set this bit to 0 to select the falling edge, and 1 to select the rising edge.

#### 24.1.5.3 SIS bit

The SIS bit is used to select the input signal to be used as an I2C bus line interrupt source. To select the MSDA pin as an I2C bus line interrupt source, set this bit to 0. To select the MSCL pin, set this bit to 1.

#### 24.1.5.4 STSPSEL bit

The STSPSEL bit is used to select the set-up and hold times when START condition and STOP condition are generated. Set this bit to 0 to select short mode and 1 to select long mode. The STSPSEL bit must be set to 1 (long mode) when the ϕIIC frequency is higher than 4 MHz. Figure 24.16 shows the START condition generation timing. Table 24.9 lists set-up and hold times when START condition and STOP condition are generated.

### 24.1.6 I2C-bus Control Register 1 (I2CCR1)

I2C-bus Control Register 1 (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | 0 | 0 | | |

Symbol: I2CCR1 | Address: 044406h | Reset Value: 00110000b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| STIE | STOP Condition Detection Interrupt Enable Bit | 0: Disabled<br>1: Enabled | RW |
| RIE | Successful Receive Interrupt Enable Bit | 0: Disabled<br>1: Enabled (3) | RW |
| —<br>(b3-b2) | Reserved | Should be written with 0 | RW |
| SDAO | Internal SDA Output Monitor Bit | 0: Low<br>1: High | RO |
| SCLO | Internal SCL Output Monitor Bit | 0: Low<br>1: High | RO |
| ICK0 | I2C-bus System Clock Select Bit (2) | b7 b6<br>0 0 : fIIC divided-by-2<br>0 1 : fIIC divided-by-4<br>1 0 : fIIC divided-by-8<br>1 1 : Do not use this combination | RW |
| ICK1 | | | RW |

Notes:
1. Do not use a 1-bit instruction to the I2CCR1 register.
2. These bits are enabled when bits ICK4 to ICK2 in the I2CCR2 register are 000b.
3. Set this bit to 0 when the ACKCLK is 0 (ACK clock is not generated).

**Figure 24.9 I2CCR1 Register**

The I2CCR1 register is used to control the I2C-bus interface.

#### 24.1.6.1 STIE bit

The STIE bit is used to enable the interrupt when a STOP condition is detected. Set this bit to 1 to enable the I2C-bus interface interrupt when a STOP condition is detected. Consequently, the STOP bit in the I2CCRR2 register becomes 1 (STOP condition detection interrupt requested) and the IR bit in the I2CIC register becomes 1 (I2C-bus interface interrupt requested).

#### 24.1.6.2 RIE bit

The RIE bit is used to enable the interrupt when the last bit of data is received, when the ACKCLK bit in the I2CCR register is 1 (ACK clock generated). When the RIE bit is 1, the I2C-bus interface interrupt is generated when the last bit (the eighth falling edge of the SCL) of data is received.

Note that the I2C-bus interface interrupt is always generated when the ACK bit (the ninth falling edge of the SCL) is received regardless of the RIE bit setting. Therefore, when the RIE bit is set to 1, two I2C-bus interface interrupts are generated per data. The source of the interrupt can be identified by reading the RIE bit. The read value indicates the internal WAIT flag state. When the read value is 1, the last bit of data is the interrupt source. When the read value is 0, the ACK bit is the interrupt source.

Set the RIE bit to 0 when the ACKCLK bit in the I2CCR register is 0 (ACK clock not generated). When the device is transmitting data or when it is receiving a slave address, the I2C-bus interface interrupt is generated only by the ACK bit (the ninth falling edge of the SCL) regardless of the RIE bit setting. In both cases, the internal WAIT flag is 0.

**Table 24.5 I2C-bus Interrupt Request Generating Timings when Data are Received and Resuming Communication**

| I2C-bus interface interrupt generating timing | Internal WAIT flag | Resuming transmission/reception |
|---|---|---|
| Last bit of data (on eighth clock) | 1 | Write to the ACKD bit in the I2CCR register |
| ACK bit (on ninth clock) | 0 | Write to the I2CTRSR register |

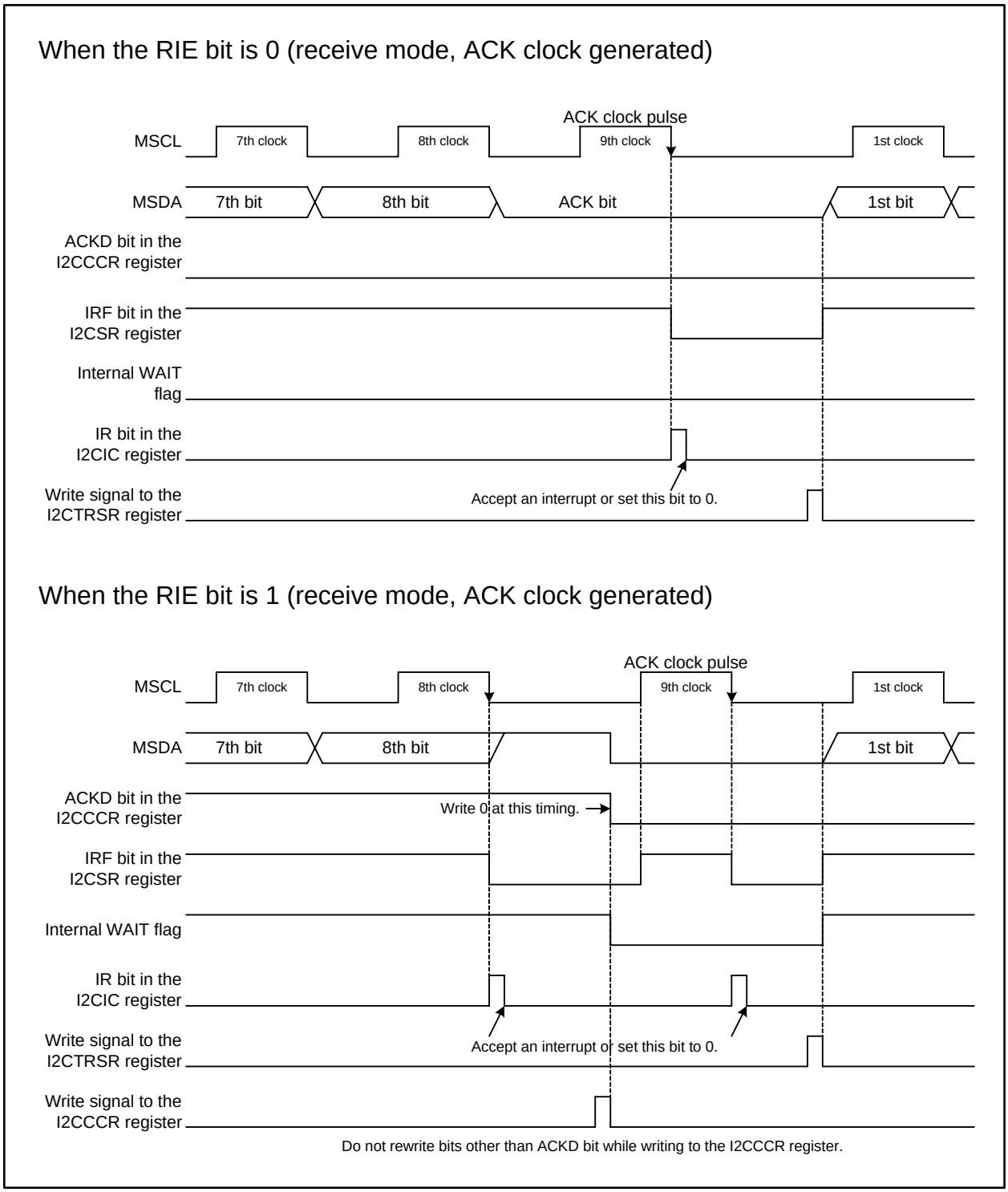



**Figure 24.10 Interrupt Request Generating Timing in Receive Mode**

### 24.1.6.3 SDAO and SCLO

Bits SDAO and SCLO are read-only bits, and are used to monitor the logical values of the internal SDA output signal and internal SCL output signal respectively. Only set these bits to 0. Note that the levels of internal SDA and SCL output signals read from bits SDO and SCLO are before influenced by the external devices, and are not the indications of MSDA and MSCL pin states.

### 24.1.6.4 ICK1 and ICK0

Bits ICK1 and ICK0 are used to select the I2C-bus system clock frequency (ϕIIC). These bits are enabled when bits ICK4 to ICK2 in the I2CCR2 register are 000b. Rewrite these bits when the ICE bit in the I2CCR0 register is 0 (I2C-bus interface disabled). The I2C-bus system clock frequency (ϕIIC) is selected from fIIC divided-by-2, -4, and -8 by setting these bits. fIIC divided-by-2.5, -3, -5, and -6 are also available by setting bits ICK4 to ICK2 in the I2CCR2 register. However, the settings for bits ICK1 and ICK0 become invalid in this case.

**Table 24.6 I2C-bus System Clock (ϕIIC) Select Bit Settings**

| I2CCR2 register | | | I2CCR1 register | | ϕIIC |
|---|---|---|---|---|---|
| ICK4 bit | ICK3 bit | ICK2 bit | ICK1 bit | ICK0 bit | |
| 0 | 0 | 0 | 0 | 0 | fIIC divided-by-2 |
| | | | 0 | 1 | fIIC divided-by-4 |
| | | | 1 | 0 | fIIC divided-by-8 |
| 0 | 0 | 1 | 0 | 0 | fIIC divided-by-2.5 |
| 0 | 1 | 0 | 0 | 0 | fIIC divided-by-3 |
| 0 | 1 | 1 | 0 | 0 | fIIC divided-by-5 |
| 1 | 0 | 0 | 0 | 0 | fIIC divided-by-6 |

Only set the values listed above.

### 24.1.7 I2C-bus Control Register 2 (I2CCR2)

I2C-bus Control Register 2

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | 0 | | | | | | |

Symbol: I2CCR2 | Address: 044407h | Reset Value: 0X000000b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TOE | Timeout Detector Enable Bit | 0: Timeout detector disabled<br>1: Timeout detector enabled | RW |
| TOF | Timeout Detect Flag | 0: Timeout not detected<br>1: Timeout detected | RO |
| TOSEL | Timeout Detect Period Select Bit | 0: Long<br>1: Short | RW |
| ICK2 | I2C-bus System Clock Select Bit | b5 b4 b3<br>0 0 0 : ϕIIC = set by bits ICK1 and ICK0 in the I2CCR1 register<br>0 0 1 : ϕIIC = fIIC / 2.5<br>0 1 0 : ϕIIC = fIIC / 3<br>0 1 1 : ϕIIC = fIIC / 5<br>1 0 0 : ϕIIC = fIIC / 6<br>Only set the values listed above | RW |
| ICK3 | | | RW |
| ICK4 | | | RW |
| — (b6) | Reserved | Should be written with 0 | RW |
| STOP | STOP Condition Detect Interrupt Request Monitor Bit | 0: I2C-bus interface interrupt not requested<br>1: I2C-bus interface interrupt requested | RW |

**Figure 24.11 I2CCR2 Register**

The I2CCR2 register is used to control the communication error detection. If the SCL clock stops, each device connected to the bus is halted suspending the communication. To avoid this, the multi-master I2C-bus interface support a function to generate an I2C-bus interface interrupt when the SCL clock is held high for a specified period of time during transmission or reception.

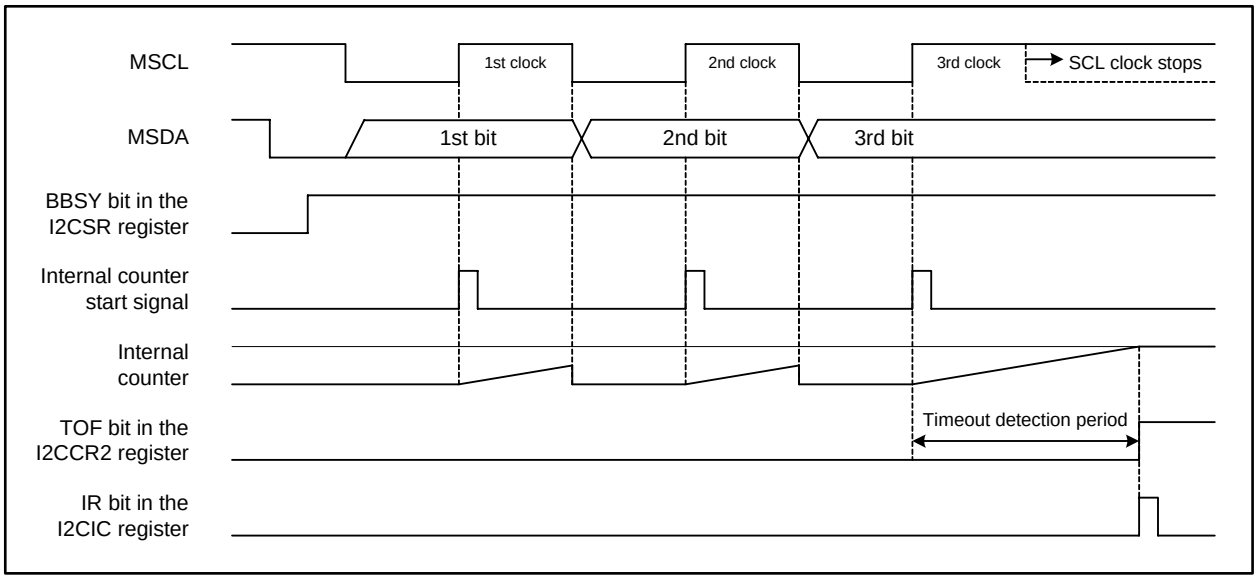



**Figure 24.12 Timeout Detecting Timing**

#### 24.1.7.1 TOE bit

The TOE bit is used to enable the timeout detector. When this bit is set to 1, the timeout detector is enabled, and when the SCL clock is held high for a specified period of time while the BBSY bit in the I2CSR register is 1 (bus is busy), an I2C-bus interface interrupt request is generated.

The timeout detection period is determined by a combination of 1) the internal counter that uses ϕIIC as a count source, and 2) the TOSEL bit setting (selects the timeout detection period to be either long or short). Refer to 24.1.7.3 “TOSEL bit”.

When a timeout is detected, set the ICE bit in the I2CCR0 register to 0 (I2C-bus interface disabled) and initialize the I2C-bus interface.

#### 24.1.7.2 TOF bit

The TOF bit is a flag to indicate the state of timeout detection. This bit is enabled when the TOE bit is 1. When the TOF bit becomes 1 (timeout detected), simultaneously the IR bit in the I2CIC register becomes 1 (I2C-bus interface interrupt requested).

#### 24.1.7.3 TOSEL bit

The TOSEL bit is used to select the length (long or short) of timeout detection period. This bit is enabled when the TOE bit is 1 (timeout detector enabled). Set this bit to 1 to select the long timeout period. In this setting, the internal counter functions as a 16-bit counter. Set this bit to 0 to select the short timeout period. In this setting, the internal counter functions as a 14-bit counter.

The internal counter increments using the I2C-bus system clock (ϕIIC) as a count source.

Table 24.7 lists timeout detection periods.

**Table 24.7 Example Timeout Detection Periods**

| ϕIIC | Long timeout detection period (TOSEL = 1) | Short timeout detection period (TOSEL = 1) |
|---|---|---|
| 4 MHz | 16.4 ms | 4.1 ms |
| 2 MHz | 32.8 ms | 8.2 ms |
| 1 MHz | 65.6 ms | 16.4 ms |

#### 24.1.7.4 ICK4 to ICK2

Bits ICK4 to ICK2 are used to select the I2C-bus system clock frequency (ϕIIC). Rewrite this bit when the ICE bit in the I2CCR0 register is 0 (I2C-bus interface disabled).

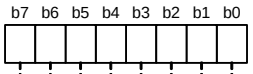
The I2C-bus system clock frequency (ϕIIC) can be selected from fIIC divided-by-2.5, -3, -5, and -6. Or, when bits ICK4 to ICK2 are 000b, the I2C-bus system clock frequency (ϕIIC) can be selected from fIIC divided-by-2, -4, and -8 by setting bits ICK1 and ICK0 in the I2CCR1 register. Refer to Table 24.6.

#### 24.1.7.5 STOP bit

The STOP bit is used to monitor the STOP condition detection interrupt. When the I2C-bus interface interrupt is generated by the detection of a STOP condition, the STOP bit becomes 1. This bit is enabled when the STIE bit in the I2CCR1 register is 1 (STOP condition detection interrupt is enabled). This bit can only be set to 0. Writing a 1 to this bit has no effect.

### 24.1.8 I2C-bus Status Register (I2CSR)

I2C-bus Status Register




Symbol: I2CSR | Address: 044408h | Reset Value: 0001 000Xb

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| LRB | Last Received Bit | 0: Last received bit is 0<br>1: Last received bit is 1 (1, 2) | RW |
| ADZ | General Call Detect Flag | 0: General call not detected<br>1: General call detected (1, 2) | RW |
| AAS | Slave Address Match Flag | 0: Address not matched<br>1: Address matched (1, 2) | RW |
| AL | Arbitration Lost Detect Flag | 0: Arbitration lost not detected<br>1: Arbitration lost detected (1, 2) | RW |
| IRF | I2C-bus Interface Interrupt Request Flag | 0: Requested<br>1: Not requested (3) | RO |
| BBSY | Bus Busy Flag | 0: Bus is free<br>1: Bus is busy (2) | RW |
| TRS | Transmit/Receive Switch Bit | 0: Receive mode<br>1: Transmit mode | RW |
| MST | Master/Slave Select Bit | 0: Slave mode<br>1: Master mode | RW |

Notes:
1. Write 1111b to the lower four bits of this register to set the TRS or MST bit to 1 without generating a START condition or STOP condition.
2. These bits are read-only when using them to check the status.
3. This bit is read-only. Only set this bit to 0.

**Figure 24.13 I2CSR Register**

The I2CSR register is used to monitor the state of the I2C-bus interface. Write to this register only when using the functions listed in Table 24.8, and only set the values that are listed. Note that the lower six bits are not rewritten even though a value from Table 24.8 is written.

**Table 24.8 I2CSR Register Settings and Functions**

| Values written to the I2CSR register | | | | | | | | Function |
|---|---|---|---|---|---|---|---|---|
| MST | TRS | BBSY | IRF | AL | AAS | ADZ | LRB | |
| 0 | 0 | X | 0 | 1 | 1 | 1 | 1 | Select slave-receive mode |
| 0 | 1 | | | | | | | Select slave-transmit mode |
| 1 | 0 | | | | | | | Select master-receive mode |
| 1 | 1 | | | | | | | Select master-transmit mode |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | Select master-transmit mode and set the device to be on STOP condition standby. |
| | | 1 | | | | | | Select master-transmit mode and set the device to be on START condition standby. |

#### 24.1.8.1 LRB bit

The LRB bit is used to store the data of last received bit. It is used to check whether the ACK is received. When the ACKCLK bit in the I2CCCR register is 1 (ACK clock generated), the LRB bit becomes 0 when the ACK is received, and 1 when the ACK is not received. When the ACKCLK bit in the I2CCCR register is 0 (ACK clock not generated), the last bit of data is stored to the LRB bit. When a value is written to the I2CTRSR register, the LRB bit becomes 0.

#### 24.1.8.2 ADZ bit

The ADZ bit is a flag to indicate that the general call (all the address data are 0) was received. When the DFS bit in the I2CCR0 register is 0 (addressing format) in slave-receive mode, the ADZ bit becomes 1 when the general call is received.
The ADZ bit becomes 0 in any of the following cases:

- When a STOP or START condition is detected
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled), or
- When the RST bit in the I2CCR0 register is set to 1 (I2C-bus interface reset)

#### 24.1.8.3 AAS bit

The AAS bit is a flag to indicate whether the received address data matches its own slave address. The AAS flag becomes 1 when the received address matches its own slave address in bits SAD6 to SAD0 in the I2CSAR register, when the DFS bit in the I2CCR0 register is 0 (addressing format) in slave-receive mode, or when the received address is the general call address.
The AAS flag becomes 0 in any of the following cases:

- When data is written to the I2CTRSR register
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled), or
- When the RST bit in the I2CCR0 register is set to 1 (I2C-bus interface reset)

#### 24.1.8.4 AL bit

The AL bit is a flag to indicate whether the arbitration lost is detected. In master transmit mode, if the MSDA pin is changed to low by other devices, then the AL bit becomes 1. Consequently, the TRS bit in the I2CSR register becomes 0 (receive mode).
When transmission of lost byte of data is completed, the MST bit in the I2CSR register becomes 0 (slave mode).
The AL bit becomes 0 in any of the following cases:

- When data is written to the I2CTRSR register
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled), or
- When the RST bit in the I2CCR0 register is set to 1 (I2C-bus interface reset)

#### 24.1.8.5 IRF bit

The IRF bit is used to generate the I2C-bus interface interrupt request signal. When the I2C-bus interface interrupt source is generated, first the IRF bit becomes 0, then the I2C-bus interface interrupt is generated on the falling edge of the IRF bit. Refer to Figure 24.10 for the timing.

The IRF bit becomes 0 in any of the following cases:

- When 1-byte data transmission is completed (including when the arbitration lost is detected)
- When 1-byte data reception is completed
- When the slave address is matched in addressing format in slave-receive mode
- When the general call address is received in addressing format in slave-receive mode, or
- When address data reception is completed in free data format in slave-receive mode

The IRF bit becomes 1 in any of the following cases:

- When data is written to the I2CTRSR register
- When data is written to the I2CCCR register (internal WAIT flag is 1)
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled), or
- When the RST bit in the I2CCR0 register is set to 1 (I2C-bus interface reset)

#### 24.1.8.6 BBSY bit

The BBSY bit is a flag to indicate the availability of I2C-bus. The BBSY flag becomes 1 when a START condition is detected, and 0 when a STOP condition is detected. When the BBSY flag is 0, the I2C-bus is not in use, and is available for the device to generate a START condition.

The detection of START condition and STOP condition are dependent on the setting of bits SSC4 to SSC0 in the I2CSSCR register.

The BBSY bit becomes 0 in any of the following cases:

- When a STOP condition is detected
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled), or
- When the RST bit in the I2CCR0 register is set to 1 (I2C-bus interface reset)

#### 24.1.8.7 TRS bit

The TRS bit is used to determine the direction of data communication. When this bit is set to 0, the device enters receive mode and waits for data to be sent from other devices. When this bit is set to 1, the device enters transmit mode and transmits data and address to the SDA line synchronized with the SCL clock.

The TRS bit automatically becomes 1 (transmit mode) when an address match is detected and the received R/W bit is 1 (data requested) in addressing format in slave-receive mode.

The TRS bit becomes 0 in any of the following cases:

- When this bit is set to 0
- When the arbitration lost is detected
- When a STOP condition is detected
- When the START condition redundancy prevention function is activated
- When a START condition is detected in slave mode
- When a NACK is received in slave mode
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled), or
- When the RST bit in the I2CCR0 register is set to 1 (I2C-bus interface reset)

#### 24.1.8.8 MST bit

The MST bit is used to select master or slave mode. To enter slave mode, set this bit to 0. Communication is initiated in synchronization with the SCL clock generated by the master device. Set this bit to 1 to enter master mode. The device generates the SCL clock to initiate communication. The MST bit becomes 0 in any of the following cases:

- When the MST bit is set to 0
- When the arbitration lost is detected, and transmission of the lost byte of data is completed
- When a STOP condition is detected
- When a START condition is detected
- When the START condition redundancy prevention function is enabled
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled), or
- When the RST bit in the I2CCR0 register is set to 1 (I2C-bus interface reset)

### 24.1.9 I2C-bus Mode Register (I2CMR)

I2C-bus Mode Register (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| I2CMR | 044410h | XXXX 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| I2CEN | I2C-bus Interface/UART2 Switch Bit | 0: UART2<br>1: I2C-bus Interface | RW |
| CLK0 | I2C-bus Interface Clock Source Select Bit | b3 b2 b1<br>0 0 0 : f1 divided-by-2<br>0 0 1 : f8 divided-by-2<br>0 1 0 : f2n divided-by-2<br>0 1 1 : Do not use this combination<br>1 0 0 : f1<br>1 0 1 : f8<br>1 1 0 : f2n<br>1 1 1 : Do not use this combination | RW |
| CLK1 | | | RW |
| CLK2 | | | RW |
| — (b7-b4) | No register bits; should be written with 0 and read as undefined value | | RW |

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

**Figure 24.14 I2CMR Register**

The I2CMR register is used to select signals for the I2C-bus interface and to select the clock source. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.

#### 24.1.9.1 I2CEN bit

The I2CEN bit is used to switch between signals for UART2 and for the I2C-bus interface. Set this bit to 1 to use the following signals: MSDA, MSCL, the I2C-bus interface interrupt, and the I2C-bus line interrupt. When this bit is set to 0, signals for UART 2 are available.

#### 24.1.9.2 CLK2 to CLK0

Bits CLK2 to CLK0 are used to select the clock source for the I2C-bus interface clock (fIIC). It is selected from f1, f8, f2n, the divide-by-2 of f1, f8, and f2n.

The clock source selected for the I2C-bus interface (fIIC) is used as the clock source for the I2C-bus system clock (ϕIIC).

## 24.2 The Generating a START Condition

To enter a START condition standby state, write E0h to the I2CSR register while the ICE bit in the I2CCR0 register is 1 (I2C-bus interface enabled) and the BBSY flag in the I2CSR register is 0 (bus is free). When in standby, write a slave address to the I2CTRSR register to generate a START condition. Consequently, the bit counter becomes 000b, 1 byte of SCL clock is output, and a slave address is transmitted. Figure 24.15 shows how to generate a START condition.

Note that writing to the I2CSR register is disabled for 1.5 cycles of ϕIIC after the BBSY flag becomes 0 (bus is free), following a STOP condition being generated. To generate a START condition immediately after generating a STOP condition, first write E0h to the I2CSR register, then confirm that bits STR and MST in the I2CSR register are 1. After that, write a slave address to the I2CTRSR register.

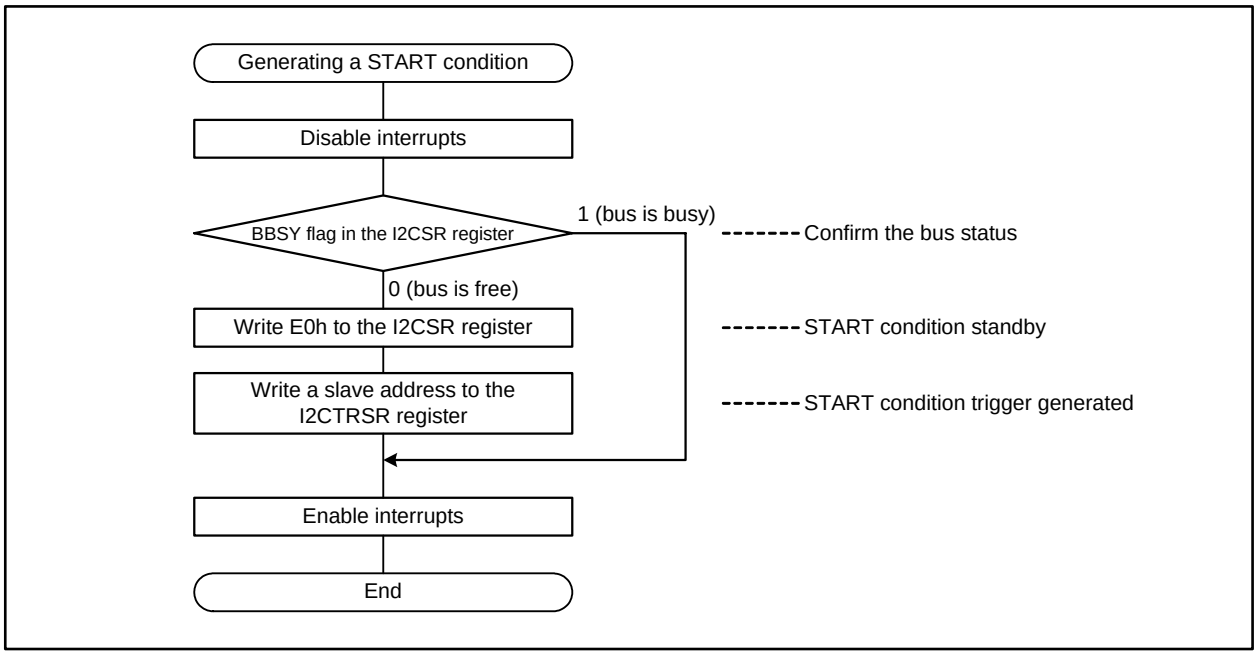



**Figure 24.15 Generating a START Condition**

The timing to generate a START condition differs between standard-mode and fast-mode. Figure 24.16 shows START condition generating timing. Table 24.9 lists the set-up and hold times when START condition or STOP condition is generated.

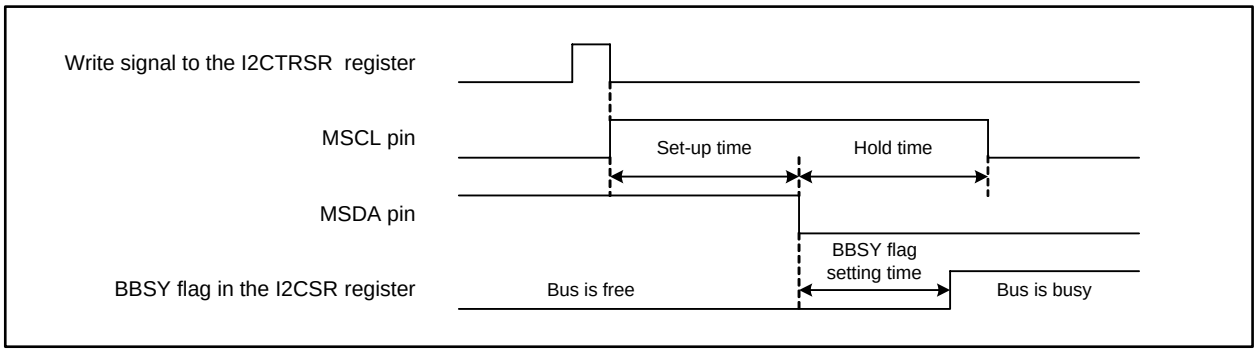



**Figure 24.16 START Condition Generating Timing**

**Table 24.9 Set-up and Hold Times When Generating a START or STOP Condition**

| Parameter | SCL mode | Short mode (STSPSEL = 0) | Long mode (STSPSEL = 1) |
|---|---|---|---|
| Set-up time | Standard-mode (CLKMD = 0) | 5.0 µs (20) | 13.0 µs (52) |
| | Fast-mode (CLKMD = 1) | 2.5 µs (10) | 6.5 µs (26) |
| Hold time | Standard-mode (CLKMD = 0) | 5.0 µs (20) | 13.0 µs (52) |
| | Fast-mode (CLKMD = 1) | 2.5 µs (10) | 6.5 µs (26) |

CLKMD: Bit in the I2CCCR register
STSPSEL: Bit in the I2CSSCR register
ϕIIC cycle numbers are in parentheses.

## 24.3 Generating a STOP Condition

To enter STOP condition standby state, write C0h to the I2CSR register while the ICE bit in the I2CCR0 register is 1 (I2C-bus interface enabled). Consequently, the MSDA pin becomes low. When in a standby state, write dummy data to the I2CTRSR register to generate a STOP condition. Figure 24.17 shows how to generate a STOP condition.

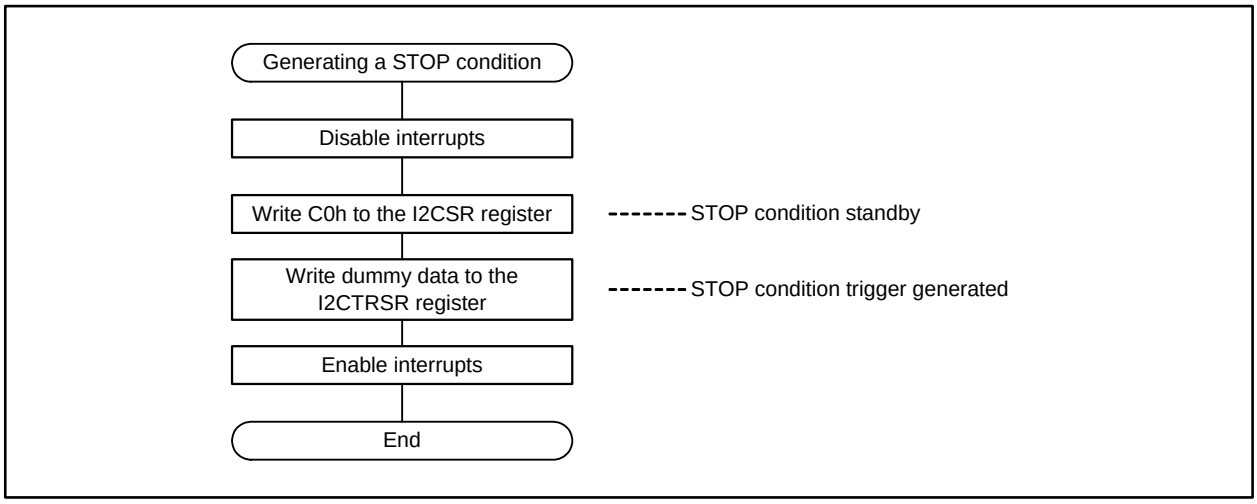



**Figure 24.17 Generating a STOP Condition**

The timing of generating a STOP condition differs between standard-mode and fast-mode. Figure 24.18 shows STOP condition generating timing. Table 24.9 list the set-up and hold times when a START or STOP condition is generated.

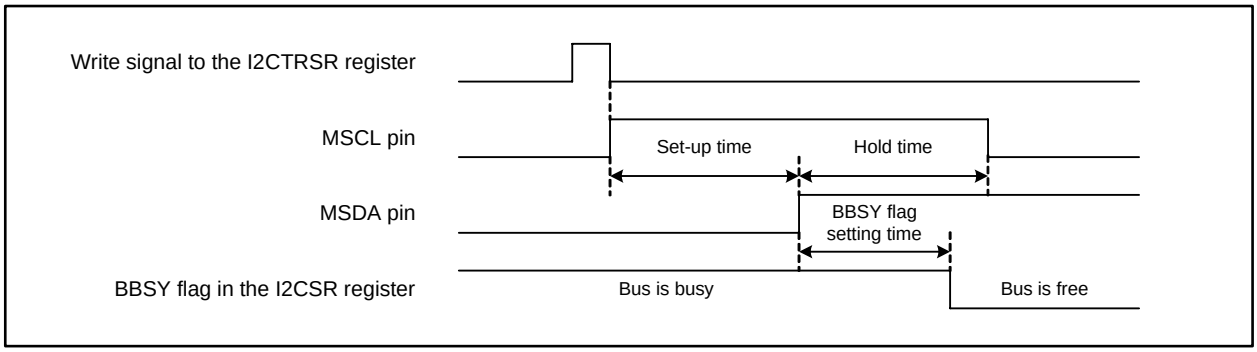



**Figure 24.18 STOP Condition Generating Timing**

To ensure the successful generation of a STOP condition, after the standby setting, do not write to the I2CSR or I2CTRSR register before the BBSY flag in the I2CSR register becomes 0 (bus is free), otherwise the STOP condition might not be generated successfully.
Furthermore, after the standby setting, if the MSCL pin input signal becomes low after the MSCL pin level becomes high, before the BBSY flag in the I2CSR register becomes 0 (bus is free), then the internal SCL output becomes low. In this case, low output from the MSCL pin is stopped (clock line released) by generating a STOP condition, by setting the ICE bit in the I2CCR0 register to 0 (I2C-bus interface enabled), or by setting the RST bit to 1 (I2C-bus interface reset)

## 24.4 START Condition Redundancy Prevention Function

A START condition is generated when the bus is free (confirmed with the BBSY flag in the I2CSR register). However, before a START condition is generated, if a different master device generates another START condition, the BBSY flag may become 1. In this case, the START condition redundancy prevention terminates the generation of its own START condition.

The START condition redundancy prevention functions as follows:

- Disable the START condition standby setting (exits from standby state)
- Disable a write to the I2CTRSR register (generation of the START condition trigger is disabled)
- Bits MST and TRS in the I2CSR register become 0 (enters into slave-receive mode)
- The AL flag in the I2CSR register becomes 1 (arbitration lost is detected)

Figure 24.19 shows the performance of the START condition redundancy prevention.

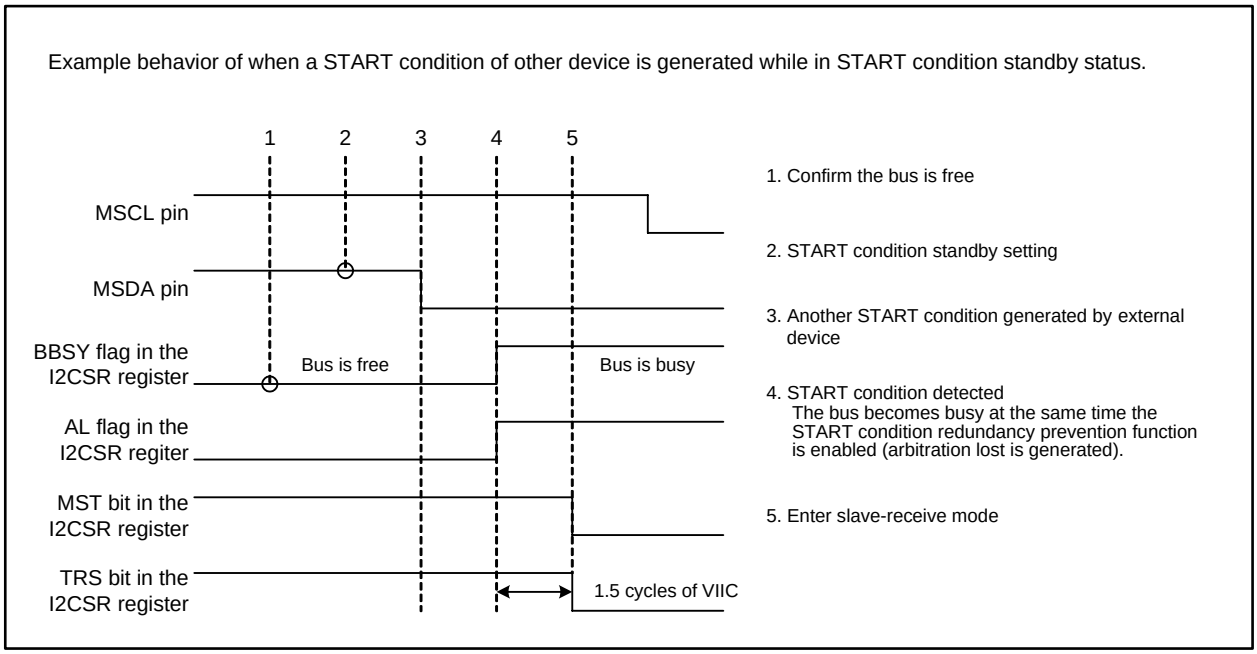



**Figure 24.19 Example Operation of the START Condition Redundancy Prevention**

The START condition redundancy prevention function is enabled from the SDA falling edge of a START condition until the slave address is completely received. This means, when registers I2CSR and I2CTRSR are written during this period, then the START condition redundancy prevention function is enabled. Figure 24.20 shows the duration.

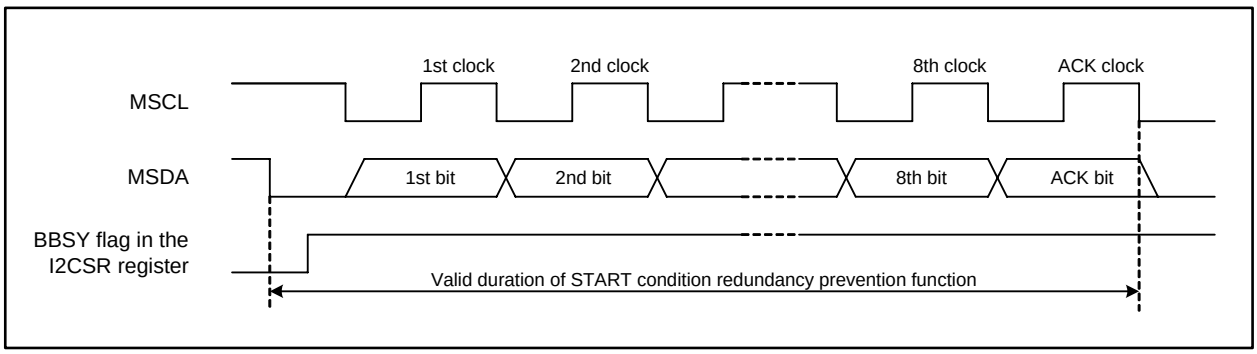



**Figure 24.20 Valid Duration of the START Condition Redundancy Prevention Function**

## 24.5 Detecting START Condition and STOP Condition

Figure 24.21 shows START condition detection. Figure 24.22 shows STOP condition detection. Table 24.10 lists the parameters for detecting START condition and STOP condition. The parameters to detect START condition and STOP condition are set by bits SSC4 to SSC0 in the I2CSSCR register. These parameters are detectable only when the input signals of pins MSCL and MSDA meet all the conditions of the high period of MSCL, set up, and hold times in Table 24.10.

The BBSY flag in the I2CSR register becomes 1 when a START condition is detected, and 0 when a STOP condition is detected. The timing for setting the BBSY flag differs between standard-mode and fast-mode. Refer to Table 24.11 for BBSY flag setting time. Table 24.11 lists the recommended settings for bits SSC4 to SSC0 in standard-mode.

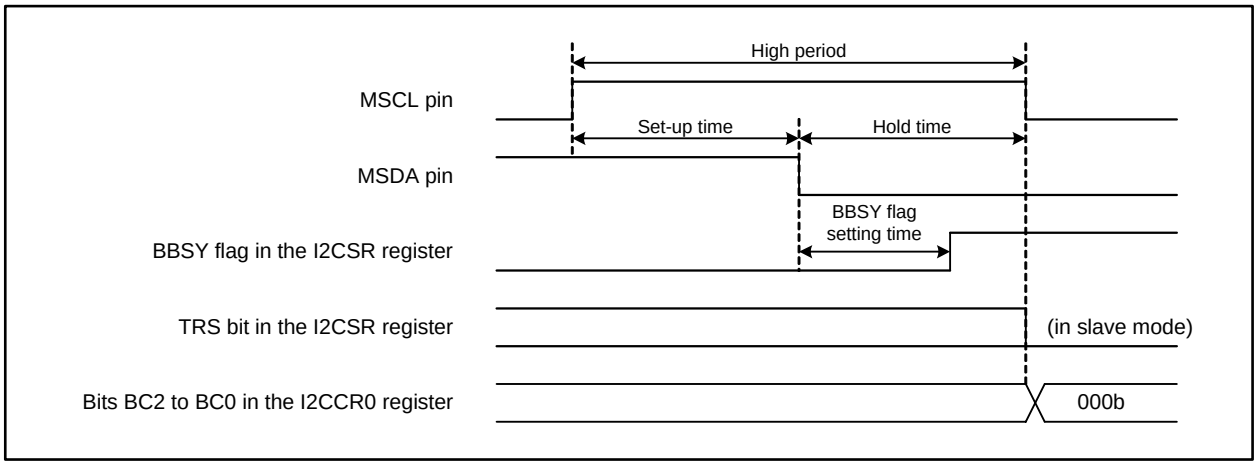



**Figure 24.21 Detecting a START Condition**

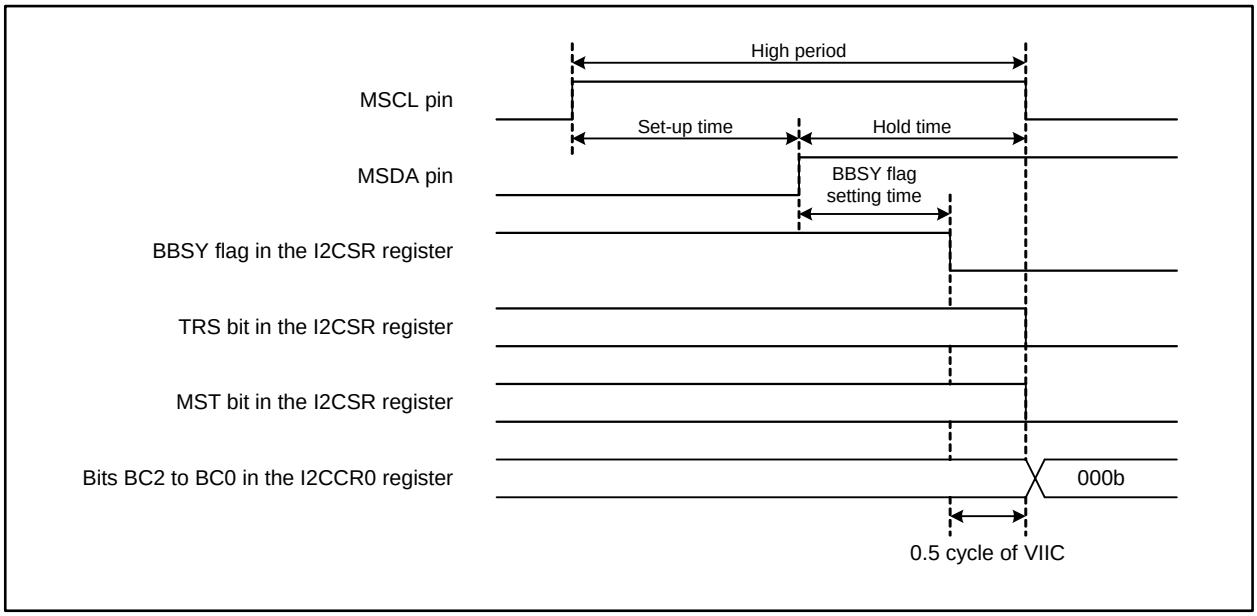



**Figure 24.22 Detecting a STOP Condition**

**Table 24.10 Parameters for Detecting START Condition and STOP Condition**

| Parameter | Standard Clock Mode | Fast Clock Mode |
|---|---|---|
| High period of MSCL pin | $\text{SSC value}$ + 1 cycle (6.25 µs) | 4 cycles (1.0 µs) |
| Set-up time | $\frac{\text{SSC value}}{2}$ + 1 cycle < 4.0 µs (3.25 µs) | 2 cycles (0.5 µs) |
| Hold time | $\frac{\text{SSC value}}{2}$ cycles < 4.0 µs (3.0 µs) | 2 cycles (0.5 µs) |
| BBSY flag setting time | $\frac{\text{SSC value - 1}}{2}$ + 2 cycles (3.375 µs) | 3.5 cycles (0.875 µs) |

Unit: ϕIIC cycle numbers
SSC value: Setting value of bits SSC4 to SSC0 in I2CSSCR register. Do not set these bits to 0 or any odd number.
Example times of when ϕIIC = 4 MHz and the I2CSSCR register = 18h are in parentheses.

**Table 24.11 Recommended Values for Bits SSC4 to SSC0 in Standard-mode**

| ϕIIC | SSC recommended value | Parameters for Detecting START Condition and STOP Condition | | | BBSY flag setting time |
|---|---|---|---|---|---|
| | | High period of MSCL pin | Set-up time | Hold time | |
| 5 MHz | 30 | 6.2 µs (31) | 3.2 µs (16) | 3.0 µs (15) | 4.125 µs (16.5) |
| 4 MHz | 26 | 6.75 µs (27) | 3.5 µs (14) | 3.25 µs (13) | 3.625 µs (14.5) |
| | 24 | 6.25 µs (25) | 3.25 µs (13) | 3.0 µs (12) | 3.375 µs (13.5) |
| 2 MHz | 12 | 6.5 µs (13) | 3.5 µs (7) | 3.0 µs (6) | 3.75 µs (7.5) |
| | 10 | 5.5 µs (11) | 3.0 µs (6) | 2.5 µs (5) | 3.25 µs (6.5) |
| 1 MHz | 4 | 5.0 µs (5) | 3.0 µs (3) | 2.0 µs (2) | 3.5 µs (3.5) |

The number of ϕIIC cycles are in parentheses.
SSC recommended values: Decimal setting value of bits SSC4 to SSC0 in the I2CSSCR register

## 24.6 Data Transmission and Reception

Examples of the data transmission and reception format for master-transmission or slave-reception in a 7-bit address format are shown in section 24.6.1 “Master Transmission” and 24.6.2 “Slave Reception”. These examples assume communication starts after initialization using the parameters set in Table 24.12.

**Table 24.12 Example of Initial Settings**

| Register | Setting value | Parameter | Initial setting |
|---|---|---|---|
| I2CSAR | 02h | Slave address | 1 |
| I2CCCR | 85h | SCL frequency | 100 kHz (ϕIIC = 4 MHz) |
| | | Clock mode | Standard-mode |
| | | ACK clock generation | ACK clock generated |
| I2CCR2 | 00h | Timeout Detector | Disabled |
| I2CCR1 | 13h | STOP condition detection interrupt | Enabled |
| | | Successful receive interrupt | Enabled |
| | | ϕIIC | fIIC divided-by-2 |
| I2CSR | 0Fh | Communication mode | Slave-receive mode |
| I2CSSCR | 98h | SSC value (see Table 24.11) | 24 |
| | | START condition/STOP condition generation mode | Long mode |
| I2CCR0 | 08h | Number of bits to be transmitted or received | 8 bits |
| | | I2C-bus interface | Enabled (communication enabled) |
| | | Data format | Addressing format |
| I2CMR | 09h | I2C-bus interface/UART2 | I2C-bus interface selected |
| | | I2C-bus interface clock source | fIIC = f2n |

### 24.6.1 Master Transmission

The behavior and procedures of master transmission are described in this section. Figure 24.23 shows an example of master transmission behavior. For (A) to (C) in the figure, see A to C in the descriptions and procedures below. (1) to (3) show the program's instructions. An arrow indicates that the procedure is performed by the MCU automatically.

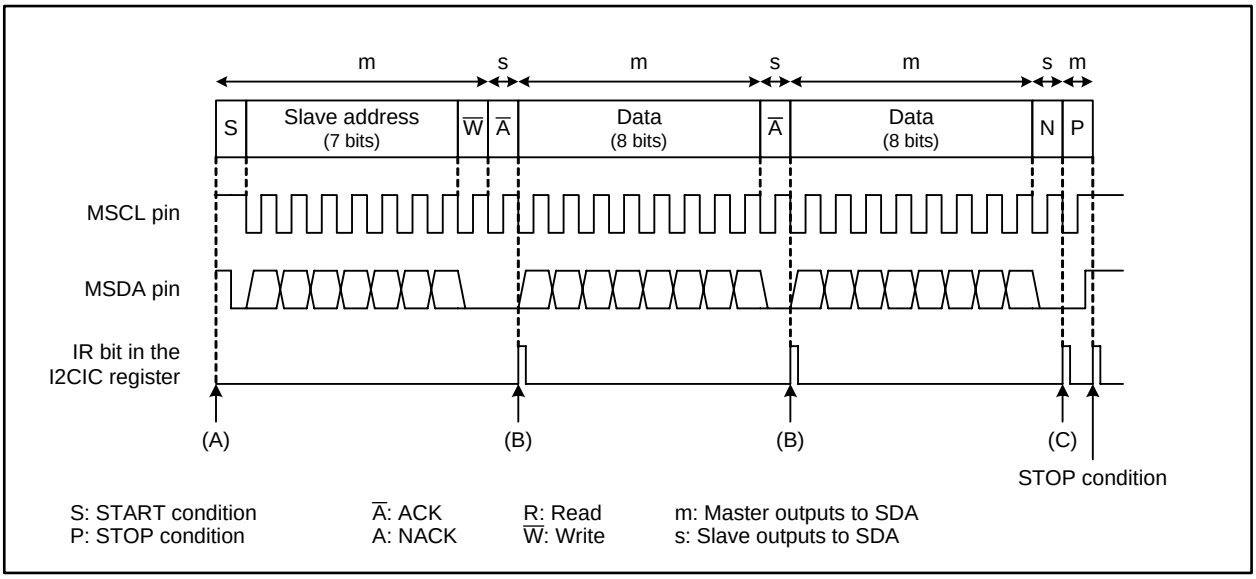



**Figure 24.23 Example Operation of Master Transmission**

A. Transmitting a slave address

(1) Confirm the BBSY flag in the I2CSR register is 0 (bus is free)
(2) Write E0h to the I2CSR register
→ The device enters the START condition standby state
(3) Write an address of a receiver (slave address) to the upper seven bits of the I2CTRSR register
→ A START condition is generated
→ The slave address is sent

B. Transmitting data (processed in the I2C-bus interrupt routine)

(1) Write transmit data to the I2CTRSR register
→ Data is sent
To send multiple bytes of data, write them to the I2CTRSR register in succession

C. Completing master transmission (processed in the I2C-bus interrupt routine)

(1) Write C0h to the I2CSR register
→ The device enters the STOP condition standby state
(2) Write dummy data to the I2CTRSR register
→ A STOP condition is generated

In addition to the case where transmission is completed, the procedure (C) is required when no ACK from the slave device is received (when a NACK is received as shown in Figure 24.23).

### 24.6.2 Slave Reception

The behavior and procedures of slave reception are described in this section. Figure 24.24 shows an example of slave reception behavior. For (A) to (C) in the figure, see A to C in the descriptions and procedures below. (1) to (3) show the program's instructions. An arrow indicates that the procedure is performed by the MCU automatically.

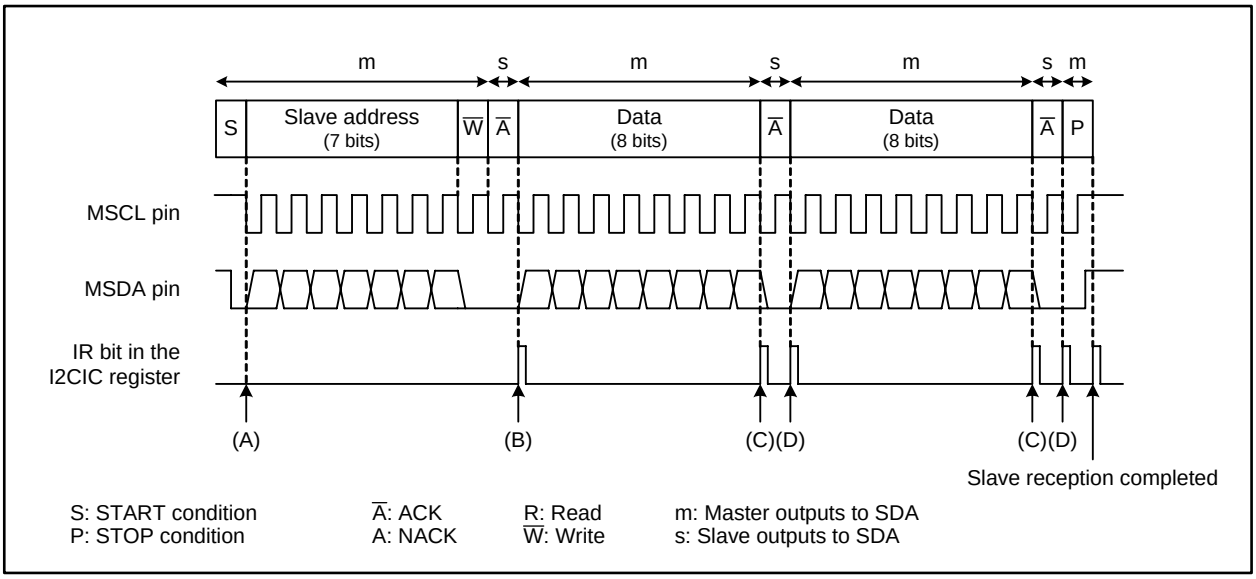



**Figure 24.24 Example Operation of Slave Reception**

A. Receiving a slave address (performed by the MCU automatically)
   → A START condition is detected
   → A slave address is received
   → An ACK is sent and the I2C-bus interface interrupt is generated in any of the following cases
   -When the general call address is received (the ADZ flag in the I2CSR register is 1), or
   -When an address match is detected (the AAS flag in the I2CSR register is 1)

B. Starting slave reception (performed inside the I2C-bus interrupt routine)
   (1) Confirm the content of the I2CSR register. When the TRS bit is 0, start the slave reception.
   (2) Write dummy data to the I2CTRSR register
   → Data reception starts

C. Completing slave reception (performed inside the I2C-bus interrupt routine)
   (1) Read the received data from the I2CTRSR register
   (2) Set the ACKD bit in the register to 1 (NACK) when the data is the last received data
   (3) Set the ACKD bit in the register to 0 (ACK) when the data is other than the last received data
   → An ACK or NACK is sent and the I2C-bus interface interrupt is generated

D. Completing ACK transmission (performed inside the I2C-bus interrupt routine)
   (1) Write dummy data to the I2CTRSR register
   → If the data is last received data, a STOP condition is detected
   → If not, data reception restarts

## 24.7 Notes on Using Multi-master I2C-bus Interface

### 24.7.1 Accessing Multi-master I2C-bus Interface-Associated Registers

Notes on reading from and writing to I2C-bus interface-associated registers.

- I2CTRSR register
  Do not write to this register during data transmission or reception. Otherwise the transmit/receive counter is reset and becomes unable to perform normal data transmission or reception.

- I2CCR0 register
  This register becomes 000b when a START condition is detected or 1 byte of data transmission or reception is completed. Do not write to or read from this register at these two timings. Otherwise the register may contain an unexpected value. Figure 24.26 and Figure 24.27 show the bit counter reset timings.

- I2CCCR register
  Only rewrite the ACKD bit during transmission or reception. Otherwise the I2C-bus clock is reset and the device becomes unable to perform normal transmission or reception.

- I2CCR1 register
  Rewrite bits ICK4 to ICK0 only when the ICE bit in the I2CCR0 register is 0 (I2C-bus interface disabled). When the I2CCR1 register is read, the internal WAIT flag status is read from this register. Therefore, do not use a bit processing instruction (read-modify-write instruction) to this register.

- I2CSR register
  Do not use a bit processing instruction (read-modify-write instruction) since the value of each bit in the I2CSR register changes depending on communication state. Also, do not write to or read from this register when the MST or TRS bits, which select communication mode, are rewritten. Otherwise the register may contain an unexpected value. Figure 24.25 to Figure 24.27 show the MST or TRS bit rewriting timing.

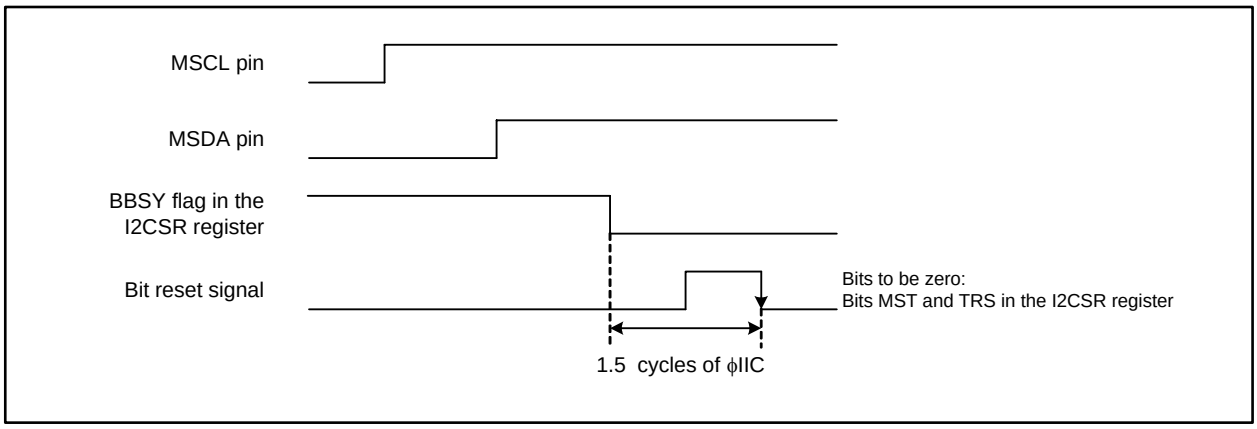



**Figure 24.25 Bit Resetting Timing (when a STOP condition is detected)**

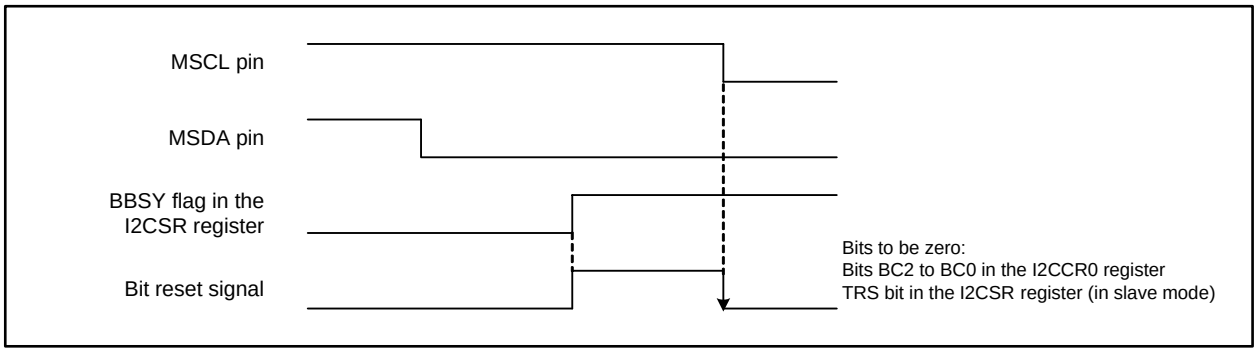



**Figure 24.26 Bit Resetting Timing (when a START condition is detected)**

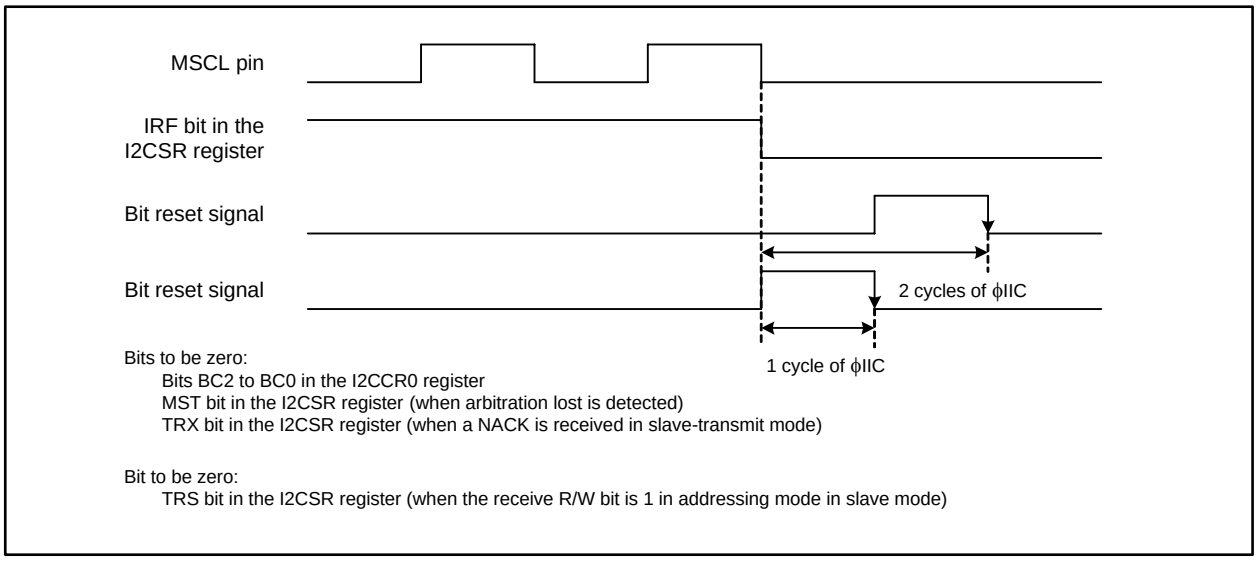



**Figure 24.27 Bit Setting/Resetting Timing (when data transmission/reception is completed)**

### 24.7.2 Generating a Repeated START condition

To generate a repeated START condition after transmitting 1-byte of data, follow the procedures below.

(1) Write E0h (the START condition standby state, and the MSDA pin is high) to the I2CSR register
(2) Wait until the MSDA pin becomes high
(3) Write a slave address to the I2CTRSR register to generate a START condition trigger

Figure 24.28 shows the repeated START condition generating timing.

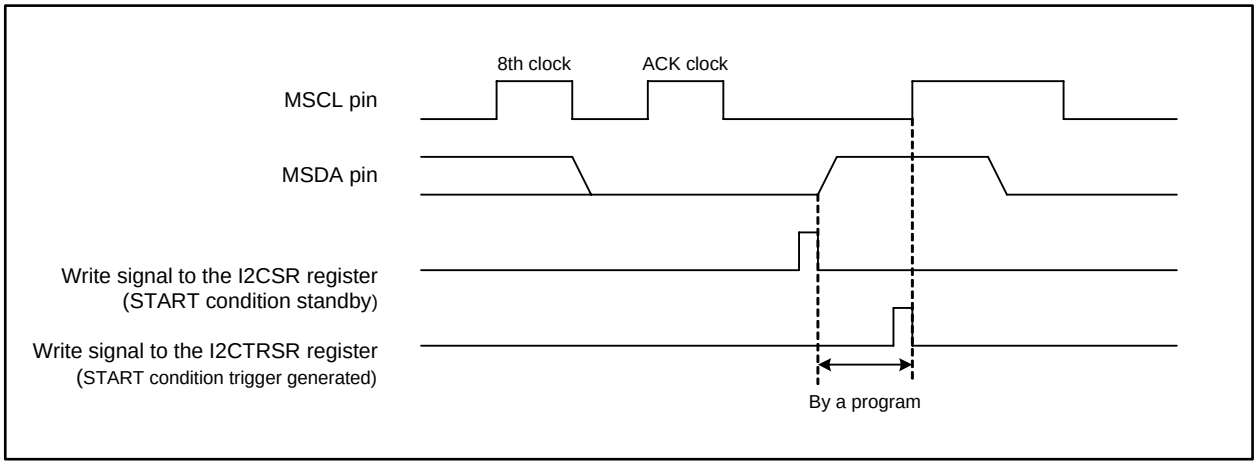



**Figure 24.28 Repeated START Condition Generating Timing**

# 25. CAN Module

The R32C/118 Group implements two channels (referred to as CAN0 and CAN1) of the Controller Area Network (CAN) module that complies with the ISO11898-1 Specifications. The CAN module transmits and receives both formats of messages, namely the standard identifier (11 bits) (identifier hereafter referred to as ID) and extended ID (29 bits).
Table 25.1 lists the CAN module specifications, and Figure 25.1 shows the CAN module block diagram.
Connect the CAN bus transceiver externally.

**Table 25.1 CAN Module Specifications (1)**

| Item | Specifications |
|---|---|
| Protocol | ISO11898-1 compliant |
| Bit rate | Up to 1 Mbps |
| Message boxes | 32 mailboxes:<br>Two selectable mailbox modes:<br>• Normal mailbox mode<br>All 32 mailboxes can be configured for transmission or reception.<br>• FIFO mailbox mode:<br>24 mailboxes can be configured for transmission or reception.<br>The remaining mailboxes can be configured as 4-stage FIFO for transmission and 4-stage FIFO for reception. |
| Reception | • Data frames and remote frames can be received.<br>• Selectable receiving ID format (only standard ID, only extended ID, or both ID)<br>• Programmable one-shot reception function<br>• Selectable overwrite mode (message overwritten) or overrun mode (message discarded)<br>• The reception complete interrupt can be individually enabled or disabled for each mailbox. |
| Acceptance filtering | 8 acceptance masks: one mask every 4 mailboxes<br>The mask can be individually enabled or disabled for each mailbox. |
| Transmission | • Data frame and remote frame can be transmitted.<br>• Selectable transmitting ID format (only standard ID, only extended ID, or both ID).<br>• Programmable one-shot transmission function<br>• Selectable ID priority transmit mode or mailbox number priority transmit mode<br>• Transmission request can be aborted. (The completion of abort can be confirmed with a flag.)<br>• The transmission complete interrupt can be individually enabled or disabled for each mailbox. |
| Mode transition for bus-off recovery | Mode transition for the recovery from the bus-off state can be selected:<br>• ISO11898-1 compliant<br>• Automatic entry to CAN halt mode at bus-off entry<br>• Automatic entry to CAN halt mode at bus-off end<br>• Entry to CAN halt mode by a program<br>• Transition to the error-active state by a program |

**Table 25.2 CAN Module Specifications (2)**

| Item | Specifications |
|---|---|
| Error status monitoring | • CAN bus errors (stuff error, form error, ACK error, CRC error, bit error, and ACK delimiter error) can be monitored.<br>• Transition to error states can be detected (error-warning, error-passive, bus-off entry, and bus-off recovery).<br>• The error counters can be read. |
| Time stamp function | Time stamp function using a 16-bit counter<br>The reference clock can be selected from either 1-, 2-, 4- or 8-bit time periods. |
| Interrupt sources | 6 types:<br>• Reception complete<br>• Transmission complete<br>• Receive FIFO<br>• Transmit FIFO<br>• Error<br>• Wake-up |
| CAN sleep mode | Current consumption can be reduced by stopping the CAN clock. |
| Software support units | 3 software support units:<br>• Acceptance filter support<br>• Mailbox search support (receive mailbox search, transmit mailbox search, and message lost search)<br>• Channel search support |
| CAN clock source | Selectable peripheral bus clock or main clock |
| Test mode | 3 test modes available for user evaluation:<br>• Listen-only mode<br>• Self-test mode 0 (external loop back)<br>• Self-test mode 1 (internal loop back) |

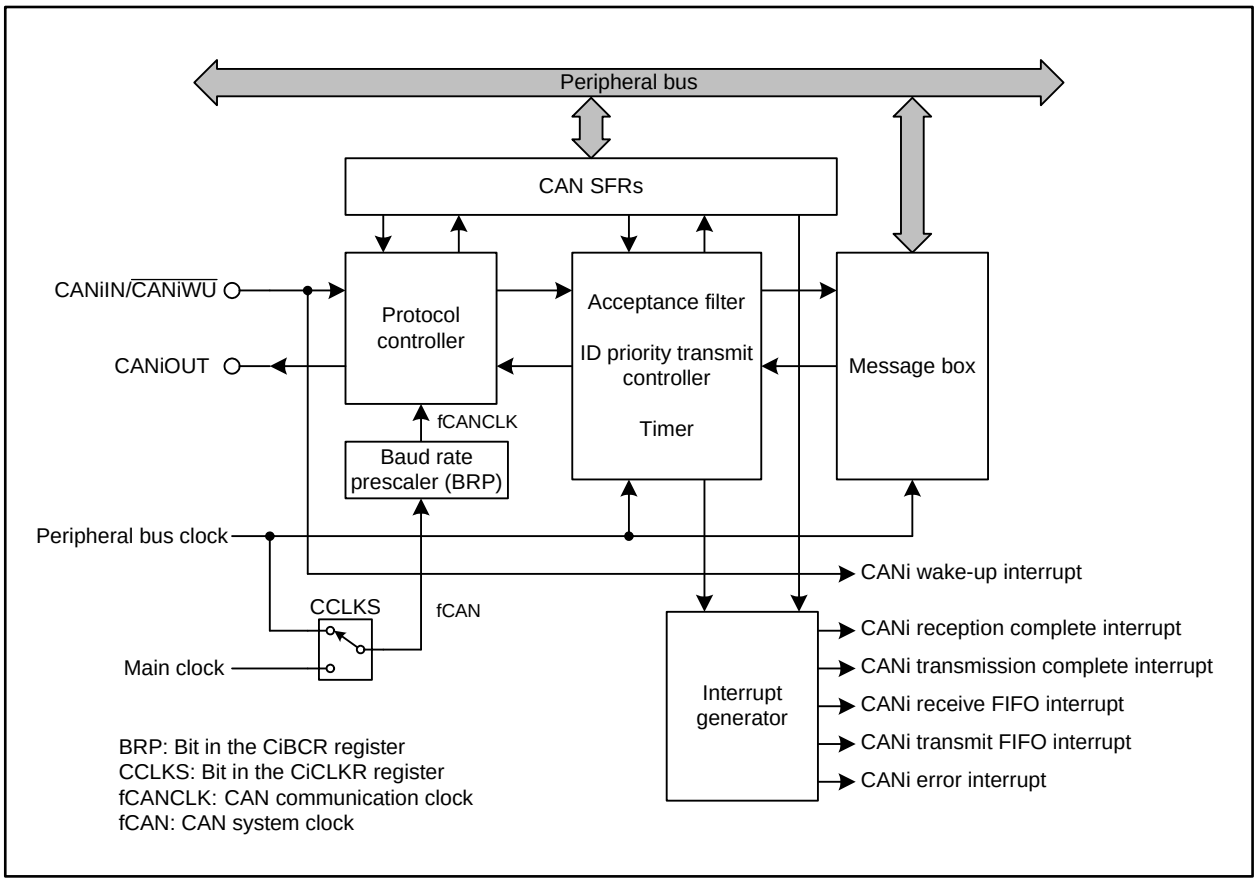



**Figure 25.1 CAN Module Block Diagram (i = 0, 1)**

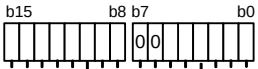
- CANiIN/CANiOUT (i = 0, 1): CAN input/output pins
- Protocol controller: Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, and error handling, etc.
- Message box: Consists of 32 mailboxes which can be configured as either transmit or receive mailboxes. Each mailbox has an individual ID, data length code, a data field (8 bytes), and a time stamp.
- Acceptance filter: Performs filtering of received messages. Registers CiMKR0 to CiMKR7 are used for the filtering process.
- Timer: Used for the time stamp function. The timer value when storing a message into the mailbox is written as the time stamp value.
- Wake-up function: Generates a CANi wake-up interrupt request when a message is detected on the CAN bus.
- Interrupt generator: Generates the following five types of interrupts:
  - CANi reception complete interrupt
  - CANi transmission complete interrupt
  - CANi receive FIFO interrupt
  - CANi transmit FIFO interrupt
  - CANi error interrupt
- CAN SFRs: CAN-associated registers. Refer to 25.1 “CAN SFRs” for details.

## 25.1 CAN SFRs

The CAN-associated registers are shown in Figures 25.2 to 25.11, 25.13, 25.14, 25.16 to 25.20, 25.22, and 25.24 to 25.30.

### 25.1.1 CANi Control Register (CiCTLR Register) (i = 0, 1)

CANi Control Register (i = 0, 1)




| Symbol | Address | Reset Value |
|---|---|---|
| C0CTLR | 47F41h-47F40h | 0000 0000 0000 0101b |
| C1CTLR | 47B41h-47B40h | 0000 0000 0000 0101b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CANM | CAN Operating Mode Select Bit (1) | b1 b0<br>0 0 : CAN operation mode<br>0 1 : CAN reset mode<br>1 0 : CAN halt mode<br>1 1 : Do not use this combination | RW |
| SLPM | CAN Sleep Mode Bit (1, 2) | 0: Other than CAN sleep mode<br>1: CAN sleep mode | RW |
| BOM | Bus-Off Recovery Mode Select Bit (3) | b4 b3<br>0 0 : Normal mode (ISO11898-1 compliant)<br>0 1: Entry to CAN halt mode automatically at bus-off entry<br>1 0 : Entry to CAN halt mode automatically at bus-off end<br>1 1 : Entry to CAN halt mode (during bus-off recovery period) by a program request | RW |
| RBOC | Forcible Return From Bus-Off Bit (4) | 0: Nothing occurred<br>1: Forcible return from bus-off (5) | RW |
| —<br>(b7-b6) | Reserved | Should be written with 0 | RW |
| MBM | CAN Mailbox Mode Select Bit (3) | 0: Normal mailbox mode<br>1: FIFO mailbox mode | RW |
| IDFM | ID Format Mode Select Bit (3) | b10b9<br>0 0 : Standard ID mode<br>0 1 : Extended ID mode<br>1 0 : Mixed ID mode<br>1 1 : Do not use this combination | RW |
| MLM | Message Lost Mode Select Bit (3) | 0: Overwrite mode<br>1: Overrun mode | RW |
| TPM | Transmit Priority Mode Select Bit (3) | 0: ID priority transmit mode<br>1: Mailbox number priority transmit mode | RW |
| TSRC | Time Stamp Counter Reset Bit (6) | 0: Nothing occurred<br>1: Reset (5) | RW |
| TSPS | Time Stamp Prescaler Select Bit (3) | b15b14<br>0 0 : Every bit time<br>0 1 : Every 2-bit time<br>1 0 : Every 4-bit time<br>1 1 : Every 8-bit time | RW |

Notes:
1. When bits CANM and SLPM are changed, check the CiSTR register to ensure that the mode has been switched.
   Do not change bits CANM and SLPM until the mode has been switched.
2. Write to the SLPM bit in CAN reset mode or CAN halt mode.
   When rewriting the SLPM bit, set only this bit to 0 or 1.
3. Write to bits BOM, MBM, IDFM, MLM, TPM, and TSPS in CAN reset mode.
4. Set the RBOC bit to 1 in bus-off state.
5. Bits RBOC and TSRC are automatically set back to 0 after being set to 1. They are read as 0.
6. Set the TSRC bit to 1 in CAN operation mode.

**Figure 25.2 Registers C0CTLR and C1CTLR**

#### 25.1.1.1 CANM Bit

The CANM bit selects one of the following modes for the CAN module: CAN operation mode, CAN reset mode, or CAN halt mode. Refer to 25.2 “Operating Mode” for detail.
CAN sleep mode is set by the SLPM bit.
Do not set the CANM bit to 11b.
When the CAN module enters CAN halt mode according to the setting of the BOM bit, the CANM bit is automatically set to 10b.

#### 25.1.1.2 SLPM Bit

When the SLPM bit is set to 1, the CAN module enters CAN sleep mode.
When this bit is set to 0, the CAN module exits CAN sleep mode.
Refer to 25.2 “Operating Mode” for detail.

#### 25.1.1.3 BOM Bit

The BOM bit is used to select bus-off recovery mode.

When the BOM bit is 00b, the recovery from bus-off is compliant with ISO11898-1, i.e. the CAN module re-enters CAN communication (error-active state) after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off.

When the BOM bit is 01b, as soon as the CAN module reaches the bus-off state, the CANM bit in the CiCTLR register (i = 0, 1) is set to 10b (CAN halt mode) and the CAN module enters CAN halt mode. No bus-off recovery interrupt request is generated when recovering from bus-off and registers CiTECR and CiRECR are set to 00h.

When the BOM bit is 10b, the CANM bit is set to 10b as soon as the CAN module reaches the bus-off state. The CAN module enters CAN halt mode after the recovery from the bus-off state, i.e. after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off and registers CiTECR and CiRECR are set to 00h.

When the BOM bit is 11b, the CAN module enters CAN halt mode by setting the CANM bit to 10b while the CAN module is still in bus-off state. No bus-off recovery interrupt request is generated when recovering from bus-off and registers CiTECR and CiRECR are set to 00h. However, if the CAN module recovers from bus-off after detecting 11 consecutive recessive bits 128 times before the CANM bit is set to10b, a bus-off recovery interrupt request is generated.

If the CPU requests an entry to CAN reset mode at the same time as the CAN module attempts to enter CAN halt mode (at bus-off entry when the BOM bit is 01b, or at bus-off end when the BOM bit is 10b), then the CPU request to enter CAN reset mode has higher priority.

#### 25.1.1.4 RBOC Bit

When the RBOC bit is set to 1 (forcible return from bus-off) in bus-off state, the CAN module forcibly returns from the bus-off state. This bit is automatically set to 0. The error state changes from bus-off to error-active.

When the RBOC bit is set to 1, registers CiRECR and CiTECR are set to 00h and the BOST bit in the CiSTR register is set to 0 (the CAN module is not in bus-off state). The other registers remain unchanged. No bus-off recovery interrupt request is generated by this recovery from the bus-off state.
Use the RBOC bit only when the BOM bit is 00b (normal mode).

### 25.1.1.5 MBM Bit

When the MBM bit is 0 (normal mailbox mode), mailboxes [0] to [31] are configured as transmit or receive mailboxes.
When this bit is 1 (FIFO mailbox mode), mailboxes [0] to [23] are configured as transmit or receive mailboxes. Mailboxes [24] to [27] are configured as a transmit FIFO and mailboxes [28] to [31] as a receive FIFO.
Transmit data is written into mailbox [24] (mailbox [24] is a window mailbox for the transmit FIFO).
Receive data is read from mailbox [28] (mailbox [28] is a window mailbox for the receive FIFO).
Table 25.3 lists the mailbox configuration.

**Table 25.3 Mailbox Configuration**

| Mailbox | MBM = 0 (Normal Mailbox Mode) | MBM = 1 (1) (FIFO Mailbox Mode) |
|---|---|---|
| Mailboxes [0] to [23] | Normal mailbox | Normal mailbox |
| Mailboxes [24] to [27] | | Transmit FIFO |
| Mailboxes [28] to [31] | | Receive FIFO |

Note:
1. When the MBM bit is set to 1, note the following:
   - Transmit FIFO is controlled by the CiTFCR register (i = 0, 1).
     The CiMCTLj register (j = 0 to 31) for mailboxes [24] to [27] is disabled.
     Registers CiMCTL24 to CiMCTL27 cannot be used.
   - Receive FIFO is controlled by the CiRFCR register.
     The CiMCTLj register for mailboxes [28] to [31] is disabled.
     Registers CiMCTL28 to CiMCTL31 cannot be used.
   - Refer to the CiMIER register about the FIFO interrupts.
   - The corresponding bits in the CiMKIVLR register for mailboxes [24] to [31] are disabled. Set 0 to these bits.
   - Transmit/receive FIFOs can be used for both data frames and remote frames.

### 25.1.1.6 IDFM Bit

The IDFM bit specifies the ID format.
When this bit is 00b, all mailboxes (including FIFO mailboxes) handle only standard IDs.
When this bit is 01b, all mailboxes (including FIFO mailboxes) handle only extended IDs.
When this bit is 10b, all mailboxes (including FIFO mailboxes) handle both standard IDs and extended IDs. Standard IDs or extended IDs are specified by using the IDE bit in the corresponding mailbox in normal mail box mode. In FIFO mailbox mode, the IDE bit in the corresponding mailbox is used for mailboxes [0] to [23], the IDE bit in registers CiFIDCR0 and CiFIDCR1 is used for the receive FIFO, and the IDE bit in mailbox [24] is used for the transmit FIFO.
Do not set 11b to the IDFM bit.

#### 25.1.1.7 MLM Bit

The MLM bit specifies the operation when a new message is captured in the unread mailbox.
Overwrite mode or overrun mode can be selected. All mailboxes (including the receive FIFO) are set to either overwrite mode or overrun mode.

When the MLM bit is 0, all mailboxes are set to overwrite mode and the new message is overwriting the old message.
When this bit is 1, all mailboxes are set to overrun mode and the new message is discarded.

#### 25.1.1.8 TPM Bit

The TPM bit specifies the priority of modes when transmitting messages.
ID priority transmit mode or mailbox number transmit mode can be selected.
All mailboxes are set for either ID priority transmission or mailbox number priority transmission.

When the TPM bit is 0, ID priority transmit mode is selected and transmission priority complies with the CAN bus arbitration rule, as defined in the ISO 11898-1 Specifications. In ID priority transmit mode, mailboxes [0] to [31] (in normal mailbox mode), and mailboxes [0] to [23] (in FIFO mailbox mode), and the transmit FIFO are compared for the IDs of mailboxes configured for transmission. If two or more mailbox IDs are the same, the mailbox with the smaller number has higher priority.

Only the next message to be transmitted from the transmit FIFO is included in the transmission arbitration. If a transmit FIFO message is being transmitted, the next pending message within the transmit FIFO is included in the transmission arbitration.

When the TPM bit is 1, mailbox number transmit mode is selected and the transmit mailbox with the smallest mailbox number has the highest priority. In FIFO mailbox mode, the transmit FIFO has lower priority than normal mailboxes (mailboxes [0] to [23]).

#### 25.1.1.9 TSRC Bit

The TSRC bit is used to reset the time stamp counter.
When this bit is set to 1, the CiTSR register (i = 0, 1) is set to 0000h. It is automatically set to 0.

#### 25.1.1.10 TSPS Bit

The TSPS bit selects the prescaler for the time stamp.
The reference clock for the time stamp can be selected from either 1-, 2-, 4- or 8-bit time periods.

### 25.1.2 CANi Clock Select Register (CiCLKR Register) (i = 0, 1)

CANi Clock Select Register (i = 0, 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | X | X | 0 | 0 | X | 0 | |

| Symbol | Address | Reset Value |
|---|---|---|
| C0CLKR | 47F47h | 000X 0000b |
| C1CLKR | 47B47h | 000X 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CCLKS | CAN Clock Source Select Bit (1) | 0: Peripheral bus clock<br>1: Main clock (2) | RW |
| — (b1) | Reserved | Should be written with 0 | RW |
| — (b2) | No register bit; should be written with 0 and read as 0 | | — |
| — (b3) | Reserved | Should be written with 0 | RW |
| — (b4) | Reserved | Should be written with 0 and read as undefined value | RW |
| — (b6-b5) | No register bits; should be written with 0 and read as 0 | | — |
| — (b7) | Reserved | Should be written with 0 | RW |

Notes:
1. Write to the CCLKS bit in CAN reset mode.
2. To set the CCLKS bit to 1, the frequency of the peripheral bus clock should be equal to or higher than the frequency of the main clock.

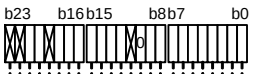
**Figure 25.3 Registers C0CLKR and C1CLKR**

#### 25.1.2.1 CCLKS Bit

When the CCLKS bit is set to 0, the CAN clock source (fCAN) originates from the PLL.
When this bit is set to 1, the fCAN originates directly from the external XIN pin bypassing the PLL.

### 25.1.3 CANi Bit Configuration Register (CiBCR Register) (i = 0, 1)

CANi Bit Configuration Register (i = 0, 1) (1, 2)




| Symbol | Address | Reset Value |
|---|---|---|
| C0BCR | 47F46h-47F44h | 00 0000h |
| C1BCR | 47B46h-47B44h | 00 0000h |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| BRP | Prescaler Division Ratio Set Bit (10 bits) | If the setting value is P (0 to 1023), the baud rate prescaler divides fCAN by P + 1 | RW |
| — (b10) | Reserved | Should be written with 0 | RW |
| — (b11) | No register bit; should be written with 0 and read as 0 | | — |
| TSEG1 | Time Segment 1 Control Bit | b15b14b13b12<br>0 0 0 0 : Do not use this combination<br>0 0 0 1 : Do not use this combination<br>0 0 1 0 : Do not use this combination<br>0 0 1 1 : 4 Tq<br>0 1 0 0 : 5 Tq<br>0 1 0 1 : 6 Tq<br>0 1 1 0 : 7 Tq<br>0 1 1 1 : 8 Tq<br>1 0 0 0 : 9 Tq<br>1 0 0 1 : 10 Tq<br>1 0 1 0 : 11 Tq<br>1 0 1 1 : 12 Tq<br>1 1 0 0 : 13 Tq<br>1 1 0 1 : 14 Tq<br>1 1 1 0 : 15 Tq<br>1 1 1 1 : 16 Tq | RW |
| TSEG2 | Time Segment 2 Control Bit | b18b17b16<br>0 0 0 : Do not use this combination<br>0 0 1 : 2 Tq<br>0 1 0 : 3 Tq<br>0 1 1 : 4 Tq<br>1 0 0 : 5 Tq<br>1 0 1 : 6 Tq<br>1 1 0 : 7 Tq<br>1 1 1 : 8 Tq | RW |
| — (b19) | No register bit; should be written with 0 and read as 0 | | — |
| SJW | Resynchronization Jump Width Control Bit | b21b20<br>0 0 : 1 Tq<br>0 1 : 2 Tq<br>1 0 : 3 Tq<br>1 1 : 4 Tq | RW |
| — (b23-b22) | No register bits; should be written with 0 and read as 0 | | — |

Notes:
1. Set the CiBCR register before entering CAN halt mode from CAN reset mode or CAN operation mode from CAN reset mode. After the setting is made once, this register can be written to in CAN reset mode or CAN halt mode.
2. The CiBCR register consists of 24 bits. A 32-bit read/write access should be performed carefully not to rewrite the CiCLKR register.

**Figure 25.4 Registers C0BCR and C1BCR**

Refer to 25.3 “CAN Communication Speed Configuration” about the bit timing configuration.

#### 25.1.3.1 BRP Bit

The BRP bit is used to set the frequency of the CAN communication clock (fCANCLK).
The cycle of the fCANCLK is set to be 1 Time Quantum (Tq).

#### 25.1.3.2 TSEG1 Bit

The TSEG1 bit is used to specify the total length of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) with the value of Tq.
A value from 4 to 16 time quanta can be set.

#### 25.1.3.3 TSEG2 Bit

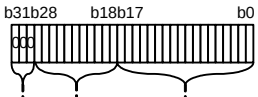
The TSEG2 bit is used to specify the length of phase buffer segment TSEG2 (PHASE_SEG2) with the value of Tq.
A value from 2 to 8 time quanta can be set.
Set the value smaller than that of the TSEG1 bit.

#### 25.1.3.4 SJW Bit

The SJW bit is used to specify the resynchronization jump width with the value of Tq.
A value from 1 to 4 time quanta can be set.
Set the value smaller than or equal to that of the TSEG2 bit.

### 25.1.4 CANi Mask Register k (CiMKRk Register) (i = 0, 1; k = 0 to 7)

CANi Mask Register k (i = 0, 1; k = 0 to 7) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| C0MKR0, C0MKR1 | 47E03h-47E00h, 47E07h-47E04h | Undefined |
| C0MKR2, C0MKR3 | 47E0Bh-47E08h, 47E0Fh-47E0Ch | Undefined |
| C0MKR4, C0MKR5 | 47E13h-47E10h, 47E17h-47E14h | Undefined |
| C0MKR6, C0MKR7 | 47E1Bh-47E18h, 47E1Fh-47E1Ch | Undefined |
| C1MKR0, C1MKR1 | 47A03h-47A00h, 47A07h-47A04h | Undefined |
| C1MKR2, C1MKR3 | 47A0Bh-47A08h, 47A0Fh-47A0Ch | Undefined |
| C1MKR4, C1MKR5 | 47A13h-47A10h, 47A17h-47A14h | Undefined |
| C1MKR6, C1MKR7 | 47A1Bh-47A18h, 47A1Fh-47A1Ch | Undefined |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| EID | Extended ID Bit | 0: Corresponding EID bit is not compared<br>1: Corresponding EID bit is compared | RW |
| SID | Standard ID Bit | 0: Corresponding SID bit is not compared<br>1: Corresponding SID bit is compared | RW |
| —<br>(b31-b29) | Reserved | Should be written with 0 | RW |

Note:
1. Write to registers CiMKR0 to CiMKR7 in CAN reset mode or CAN halt mode.

**Figure 25.5 Registers C0MKR0 to C1MKR7**

Refer to 25.5 “Acceptance Filtering and Masking Function” about the masking function in FIFO mailbox mode.

#### 25.1.4.1 EID Bit

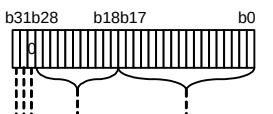
The EID bit is the filter mask bit corresponding to the CAN extended ID bit. This bit is used to receive extended ID messages.
When the EID bit is 0, the corresponding EID bit is not compared for the received ID and the mailbox ID.
When this bit is 1, the corresponding EID bit is compared for the received ID and the mailbox ID.

#### 25.1.4.2 SID Bit

The SID bit is the filter mask bit corresponding to the CAN standard ID bit. This bit is used to receive both standard ID and extended ID messages.
When the SID bit is 0, the corresponding SID bit is not compared for the received ID and the mailbox ID.
When this bit is 1, the corresponding SID bit is compared for the received ID and the mailbox ID.

### 25.1.5 CANi FIFO Received ID Compare Register n (Registers CiFIDCR0 and CiFIDCR1) (i = 0, 1; n = 0, 1)

CANi FIFO Received ID Compare Register n (i = 0, 1; n = 0, 1) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| C0FIDCR0 | 47E23h-47E20h | Undefined |
| C0FIDCR1 | 47E27h-47E24h | Undefined |
| C1FIDCR0 | 47A23h-47A20h | Undefined |
| C1FIDCR1 | 47A27h-47A24h | Undefined |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| EID | Extended ID Bit | 0: Corresponding EID bit is 0<br>1: Corresponding EID bit is 1 | RW |
| SID | Standard ID Bit | 0: Corresponding SID bit is 0<br>1: Corresponding SID bit is 1 | RW |
| —<br>(b29) | Reserved | Should be written with 0 | RW |
| RTR | Remote Frame Request Bit | 0: Data frame<br>1: Remote frame | RW |
| IDE | ID Extension Bit (2) | 0: Standard ID<br>1: Extended ID | RW |

Notes:
1. Write to registers CiFIDCR0 and CiFIDCR1 in CAN reset mode or CAN halt mode.
2. The IDE bit is enabled when the IDFM bit in the CiCTLR register is 10b (mixed ID mode). When the IDFM bit is not 10b, the IDE bit should be written with 0.

**Figure 25.6 Registers C0FIDCR0 to C1FIDCR1**

Registers CiFIDCR0 and CiFIDCR1 are enabled when the MBM bit in the CiCTLR register is set to 1 (FIFO mailbox mode). Bits EID, SID, RTR, and IDE in registers CiMB28 to CiMB31 are disabled. Refer to 25.5 “Acceptance Filtering and Masking Function” about the usage of these registers.

#### 25.1.5.1 EID Bit

The EID bit sets the extended ID of data frames and remote frames. This bit is used to receive extended ID messages.

#### 25.1.5.2 SID Bit

The SID bit sets the standard ID of data frames and remote frames. This bit is used to receive both standard ID and extended ID messages.

#### 25.1.5.3 RTR Bit

The RTR bit sets the specified frame format of data frames or remote frames.
This bit specifies the following operation:

- When both RTR bits in registers CiFIDCR0 and CiFIDCR1 (i = 0, 1) are set to 0, only data frames can be received.
- When both RTR bits in registers CiFIDCR0 and CiFIDCR1 are set to 1, only remote frames can be received.
- When the RTR bits in registers CiFIDCR0 and CiFIDCR1 are set to 0 or 1 individually, both data frames and remote frames can be received.

#### 25.1.5.4 IDE bit

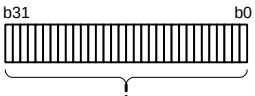
The IDE bit sets the ID format of standard ID or extended ID.
This bit is enabled when the IDFM bit in the CiCTLR register is 10b (mixed ID mode).
When the IDFM bit is 10b, the IDE bit specifies the following operation:

- When both IDE bits in registers CiFIDCR0 and CiFIDCR1 are set to 0, only standard ID frames can be received.
- When both IDE bits in registers CiFIDCR0 and CiFIDCR1 are set to 1, only extended ID frames can be received.
- When the IDE bits in registers CiFIDCR0 and CiFIDCR1 are set to 0 or 1 individually, both standard ID and extended ID frames can be received.

### 25.1.6 CANi Mask Invalid Register (CiMKIVLR Register) (i = 0, 1)

CANi Mask Invalid Register (i = 0, 1) (1)

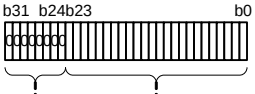
Normal mailbox mode




| Symbol | Address | Reset Value |
|---|---|---|
| C0MKIVLR | 47E2Bh-47E28h | Undefined |
| C1MKIVLR | 47A2Bh-47A28h | Undefined |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b31-b0) | Mask Invalid Bit | 0: Mask valid<br>1: Mask invalid | RW |

FIFO mailbox mode




| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b23-b0) | Mask Invalid Bit | 0: Mask valid<br>1: Mask invalid | RW |
| — (b31-b24) | Reserved | Should be written with 0 | RW |

Note:
1. Write to the CiMKIVLR register in CAN reset mode or CAN halt mode.

**Figure 25.7 Registers C0MKIVLR and C1MKIVLR**

Each bit corresponds to the mailbox with the same number. When each bit is 1, the acceptance mask for the mailbox corresponding to the bit number is disabled. In this case, a receiving message is stored into the mailbox only if its ID matches bits SID and EID in the CiMBj register (j = 0 to 31).

### 25.1.7 CANi Mailbox (CiMBj Register) (i = 0, 1; j = 0 to 31)

Table 25.4 lists the CANi mailbox memory mapping, and Table 25.5 lists the CAN data frame structure. The value after reset of CANi mailbox is undefined.

**Table 25.4 CANi Mailbox Memory Mapping (i = 0, 1)**

| Address | | Message Content |
|---|---|---|
| CAN0 | CAN1 | Memory Mapping |
| 47C00h + j * 16 + 0 | 47800h + j * 16 + 0 | EID7 to EID0 |
| 47C00h + j * 16 + 1 | 47800h + j * 16 + 1 | EID15 to EID8 |
| 47C00h + j * 16 + 2 | 47800h + j * 16 + 2 | SID5 to SID0, EID17, EID16 |
| 47C00h + j * 16 + 3 | 47800h + j * 16 + 3 | IDE, RTR, SID10 to SID6 |
| 47C00h + j * 16 + 4 | 47800h + j * 16 + 4 | — |
| 47C00h + j * 16 + 5 | 47800h + j * 16 + 5 | Data length code (DLC) |
| 47C00h + j * 16 + 6 | 47800h + j * 16 + 6 | Data byte 0 |
| 47C00h + j * 16 + 7<br>:<br>:<br>:<br>47C00h + j * 16 + 13 | 47800h + j * 16 + 7<br>:<br>:<br>:<br>47800h + j * 16 + 13 | Data byte 1<br>:<br>:<br>:<br>Data byte 7 |
| 47C00h + j * 16 + 14 | 47800h + j * 16 + 14 | Time stamp lower byte |
| 47C00h + j * 16 + 15 | 47800h + j * 16 + 15 | Time stamp upper byte |

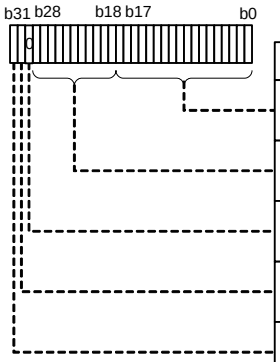
j: Mailbox number (j = 0 to 31)

**Table 25.5 CAN Data Frame Structure**

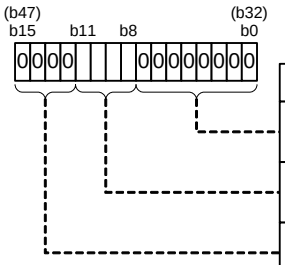
| SID10 to SID6 | SID5 to SID0 | EID17 to EID16 | EID15 to EID8 | EID7 to EID0 | DLC3 to DLC0 | DATA0 | DATA1 | ..... | DATA7 |
|---|---|---|---|---|---|---|---|---|---|

## CANi Mailbox Register j (i = 0, 1; j = 0 to 31) (1)

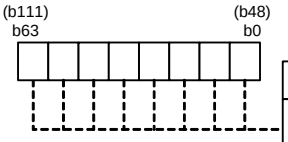
| Symbol | Address (2) | Reset Value |
|---|---|---|
| C0MB0 to C0MB31 | 47C00h to 47DFFh | Undefined |
| C1MB0 to C1MB31 | 47800h to 479FFh | Undefined |



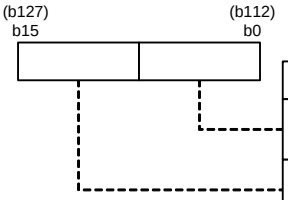

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| EID | Extended ID (3) | 0: Corresponding EID bit is 0<br>1: Corresponding EID bit is 1 | RW |
| SID | Standard ID | 0: Corresponding SID bit is 0<br>1: Corresponding SID bit is 1 | RW |
| —<br>(b29) | Reserved | Should be written with 0 | RW |
| RTR | Remote Frame Request Bit | 0: Data frame<br>1: Remote frame | RW |
| IDE | ID Extension Bit (4) | 0: Standard ID<br>1: Extended ID | RW |




| Bit Symbol | Bit Name | Setting Value | RW |
|---|---|---|---|
| —<br>(b7-b0) | Reserved | Should be written with 0 | RW |
| DLC | Data Length Code (5) | 0h to Fh | RW |
| —<br>(b15-b12) | Reserved | Should be written with 0 | RW |




| Symbol | Name | Setting Value | RW |
|---|---|---|---|
| DATA0 to DATA7 | Data Bytes 0 to 7 (5, 6) | 00h to FFh | RW |




| Symbol | Name | Setting Value | RW |
|---|---|---|---|
| TSL | Time Stamp Lower Byte | 00h to FFh | RW |
| TSH | Time Stamp Higher Byte | 00h to FFh | RW |

Notes:
1. Write to the CiMBj register only when the associated CiMCTLj register is 00h and the corresponding mailbox is not processing an abort request.
2. Refer to the memory mapping table for CANi mailbox on the previous page for detailed addresses.
3. If the mailbox has received a standard ID message, the EID bit in the mailbox is undefined.
4. The IDE bit is enabled when the IDFM bit in the CiCTLR register is 10b (mixed ID mode). When the IDFM bit is not 10b, it should be written with 0.
5. If the mailbox has received a message with n bytes less than 8 bytes, the values of DATAn to DATA7 in the mailbox are undefined.
6. If the mailbox has received a remote frame, the previous values of DATA0 to DATA7 in the mailbox are retained.

**Figure 25.8 Registers C0MBj and C1MBj**

The previous value of each mailbox is retained unless a new message is received.

#### 25.1.7.1 EID Bit

The EID bit sets the extended ID of data frames and remote frames. This bit is used to transmit or receive extended ID messages.

#### 25.1.7.2 SID Bit

The SID bit sets the standard ID of data frames and remote frames. This bit is used to transmit or receive both standard ID and extended ID messages.

#### 25.1.7.3 RTR Bit

The RTR bit sets the frame format of data frames or remote frames.
This bit specifies the following operation:

- Receive mailbox receives only frames with the format specified by the RTR bit.
- Transmit mailbox transmits according to the frame format specified by the RTR bit.
- Receive FIFO mailbox receives the data frame, remote frame, or both frames specified by the RTR bit in registers CiFIDCR0 and CiFIDCR1 (i = 0, 1).
- Transmit FIFO mailbox transmits the data frame or remote frame specified by the RTR bit in the relevant transmitting message.

#### 25.1.7.4 IDE Bit

The IDE bit sets the ID format of standard IDs or extended IDs.
This bit is enabled when the IDFM bit in the CiCTLR register is 10b (mixed ID mode).
When the IDFM bit is 10b, the IDE bit specifies the following operation:

- Receive mailbox receives only the ID format specified by the IDE bit.
- Transmit mailbox transmits according to the ID format specified by the IDE bit.
- Receive FIFO mailbox receives messages with the standard ID, extended ID, or both IDs specified by the IDE bit in registers CiFIDCR0 and CiFIDCR1.
- Transmit FIFO mailbox transmits messages with the standard ID or extended ID specified by the IDE bit in the relevant transmitting message.

### 25.1.7.5 DLC (Data Length Code)

The DLC is used to set the number of data bytes to be transmitted in a data frame. When data is requested using a remote frame, the number of data bytes to be requested is set.
When a data frame is received, the number of received data bytes is stored. When a remote frame is received, the number of requested data bytes is stored.
Table 25.6 lists the data length corresponding DLC.

**Table 25.6 Data Length Corresponding DLC**

| DLC[3] | DLC[2] | DLC[1] | DLC[0] | Data Length |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 byte |
| 0 | 0 | 0 | 1 | 1 byte |
| 0 | 0 | 1 | 0 | 2 bytes |
| 0 | 0 | 1 | 1 | 3 bytes |
| 0 | 1 | 0 | 0 | 4 bytes |
| 0 | 1 | 0 | 1 | 5 bytes |
| 0 | 1 | 1 | 0 | 6 bytes |
| 0 | 1 | 1 | 1 | 7 bytes |
| 1 | X | X | X | 8 bytes |

X: Any value

### 25.1.7.6 DATA0 to DATA7

DATA0 to DATA7 store the transmitted or received CAN message data. Transmission or reception starts from DATA0. The bit order on the CAN bus is MSB first, and transmission or reception starts from bit 7.

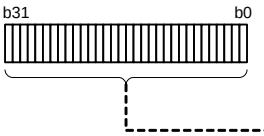
### 25.1.7.7 TSL and TSH

TSL and TSH store the counter value of the time stamp when received messages are stored in the mailbox.

### 25.1.8 CANi Mailbox Interrupt Enable Register (CiMIER Register) (i = 0, 1)

CANi Mailbox Interrupt Enable Register (i = 0, 1) (1, 2)

Normal mailbox mode




| Symbol | Address | Reset Value |
|---|---|---|
| C0MIER | 47E2Fh-47E2Ch | Undefined |
| C1MIER | 47A2Fh-47A2Ch | Undefined |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b31-b0) | Interrupt Enable Bit | 0: Interrupt disabled<br>1: Interrupt enabled | RW |

FIFO mailbox mode


| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b23-b0) | Interrupt Enable Bit | 0: Interrupt disabled<br>1: Interrupt enabled | RW |
| — (b24) | Transmit FIFO Interrupt Enable Bit | 0: Interrupt disabled<br>1: Interrupt enabled | RW |
| — (b25) | Transmit FIFO Interrupt Generation Timing Control Bit | Transmit FIFO interrupt request is generated<br>0: Every time transmission is completed<br>1: When transmit FIFO becomes empty due to completion of transmission | RW |
| — (b27-b26) | Reserved | Should be written with 0 | RW |
| — (b28) | Receive FIFO Interrupt Enable Bit | 0: Interrupt disabled<br>1: Interrupt enabled | RW |
| — (b29) | Receive FIFO Interrupt Generation Timing Control Bit | Receive FIFO interrupt request is generated<br>0: Every time reception is completed<br>1: When receive FIFO becomes buffer warning by completion of reception (3) | RW |
| — (b31-b30) | Reserved | Should be written with 0 | RW |

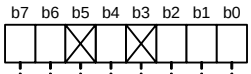
Notes:
1. Write to the CiMIER register only when the associated CiMCTLj register (j = 0 to 31) is 00h and the corresponding mailbox is not processing a transmission or reception abort request.
2. In FIFO mailbox mode, change the bits in the CiMIER register for the associated FIFO only when:
   - The TFE bit in the CiTFCR register is 0 and the TFEST bit is 1, and
   - The RFE bit in the CiRFCR register is 0 and the RFEST bit is 1.
3. No interrupt request is generated when the receive FIFO becomes buffer warning from full.

**Figure 25.9 Registers C0MIER and C1MIER**

Interrupts can enabled individually for each mailbox.
In normal mailbox mode (bits 0 to 31) and in FIFO mailbox mode (bits 0 to 23), each bit corresponds to the mailbox with the same number. These bits enable or disable transmission/reception complete interrupts for the corresponding mailboxes.
In FIFO mailbox mode, bits 24, 25, 28, and 29 specify whether transmit/receive FIFO interrupts are enabled/disabled and timing when interrupt requests are generated.
“Buffer warning” indicates a state in which the third unread message is stored in the receive FIFO.

### 25.1.9 CANi Message Control Register (CiMCTLj Register) (i = 0, 1; j = 0 to 31)

CANi Message Control Register j (i = 0, 1; j = 0 to 31) (1, 2)




| Symbol | Address | Reset Value |
|---|---|---|
| C0MCTL0 to C0MCTL31 | 47F20h to 47F3Fh | 00h |
| C1MCTL0 to C1MCTL31 | 47B20h to 47B3Fh | 00h |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| When the TRMREQ bit is 0 and the RECREQ bit is 1 | | | |
| NEWDATA | Reception Complete Flag (3, 4) | 0: No data has been received or 0 is written to the NEWDATA bit<br>1: A new message is being stored or has been stored to the mailbox | RW |
| INVALDATA | Reception-in-Progress Status Flag | 0: Message valid<br>1: Message being updated | RO |
| MSGLOST | Message Lost Flag (3, 4) | 0: Message is not overwritten or overrun<br>1: Message is overwritten or overrun | RW |
| When the TRMREQ bit is 1 and the RECREQ bit is 0 | | | |
| SENTDATA | Transmission Complete Flag (3, 4) | 0: Transmission is not completed (pending)<br>1: Transmission is completed (success) | RW |
| TRMACTIVE | Transmission-in-Progress Status Flag | 0: Transmission is pending or transmission is not requested<br>1: From acceptance of transmission request to completion of transmission, or error/arbitration lost | RO |
| TRMABT | Transmission Abort Complete Flag (3, 4) | 0: Transmission has started, transmission abort failed because transmission is completed, or transmission abort is not requested<br>1: Transmission abort is completed | RW |
| — (b3) | No register bit; should be written with 0 and read as 0 | | — |
| ONESHOT | One-shot Enable Bit (5) | 0: One-shot reception or one-shot transmission disabled<br>1: One-shot reception or one-shot transmission enabled | RW |
| — (b5) | No register bit; should be written with 0 and read as 0 | | — |
| RECREQ | Receive Mailbox Set Bit (4, 6, 7) | 0: Not configured for reception<br>1: Configured for reception | RW |
| TRMREQ | Transmit Mailbox Set Bit (4, 6) | 0: Not configured for transmission<br>1: Configured for transmission | RW |

Notes:
1. Write to the CiMCTLj register in CAN operation mode or CAN halt mode.
2. Do not use registers CiMCTL24 to CiMCTL31 in FIFO mailbox mode.
3. Write 0 only. Writing 1 has no effect.
4. When writing 0 to bits NEWDATA, SENTDATA, MSGLOST, TRMABT, RECREQ, and TRMREQ by a program, use the MOV instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1.
5. To enter one-shot receive mode, write 1 to the ONESHOT bit at the same time as setting the RECREQ bit to 1. To exit one-shot receive mode, write 0 to the ONESHOT bit after writing 0 to the RECREQ bit and confirming it has been set to 0.
   To enter one-shot transmit mode, write 1 to the ONESHOT bit at the same time as setting the TRMREQ bit to 1. To exit one-shot transmit mode, write 0 to the ONESHOT bit after the message has been transmitted or aborted.
6. Do not set both the RECREQ and TRMREQ bits to 1.
7. When setting the RECREQ bit to 0, set bits MSGLOST, NEWDATA, RECREQ to 0 simultaneously.

**Figure 25.10 Registers C0MCTLj and C1MCTLj**

#### 25.1.9.1 NEWDATA Bit

The NEWDATA bit is set to 1 when a new message is being stored or has been stored to the mailbox.
The timing for setting this bit to 1 is simultaneous with the INVALDATA bit.
The NEWDATA bit is set to 0 by writing 0 by a program.
This bit is not be set to 0 by writing 0 by a program while the related INVALDATA bit is 1.

#### 25.1.9.2 SENTDATA Bit

The SENTDATA bit is set to 1 when data transmission from the corresponding mailbox is completed.
This bit is set to 0 by writing 0 by a program.
To set the SENTDATA bit to 0, first set the TRMREQ bit to 0.
Bits SENTDATA and TRMREQ cannot be set to 0 simultaneously.
To transmit a new message from the corresponding mailbox, set the SENTDATA bit to 0.

#### 25.1.9.3 INVALDATA Bit

After the completion of a message reception, the INVALDATA bit is set to 1 while the received message is being updated into the corresponding mailbox.
This bit is set to 0 immediately after the message has been stored. If the mailbox is read while this bit is 1, the data is undefined.

#### 25.1.9.4 TRMACTIVE Bit

The TRMACTIVE bit is set to 1 when the corresponding mailbox of the CAN module begins transmitting a message.
This bit is set to 0 when the CAN module has lost CAN bus arbitration, a CAN bus error occurs, or data transmission is completed.

#### 25.1.9.5 MSGLOST Bit

The MSGLOST bit is set to 1 when the mailbox is overwritten or overrun by a new received message while the NEWDATA bit is 1. The MSGLOST bit is set to 1 at the end of the 6th bit of EOF.
This bit is set to 0 by writing 0 by a program.
In both overwrite and overrun modes, this bit is not set to 0 by writing 0 by a program during five cycles of fCAN (CAN system clock) following the 6th bit of EOF.

#### 25.1.9.6 TRMABT Bit

The TRMABT bit is set to 1 in the following cases:

- Following a transmission abort request, when the transmission abort is completed before starting transmission.
- Following a transmission abort request, when the CAN module detects CAN bus arbitration lost or a CAN bus error.
- In one-shot transmission mode (RECREQ bit = 0, TRMREQ bit = 1, and ONESHOT bit = 1), when the CAN module detects CAN bus arbitration lost or a CAN bus error.

The TRMABT bit is not set to 1 when data transmission is completed. In this case, the SENTDATA bit is set to 1.
The TRMABT bit is set to 0 by writing 0 by a program.

### 25.1.9.7 ONESHOT Bit

The ONESHOT bit can be used in the following two ways, receive mode and transmit mode:

(1) One-Shot Receive Mode

When the ONESHOT bit is set to 1 in receive mode (RECREQ bit = 1 and TRMREQ bit = 0), the mailbox receives a message only one time. The mailbox does not behave as a receive mailbox after having received a message one time. The behavior of bits NEWDATA and INVALDATA is the same as in normal reception mode. In one-shot receive mode, the MSGLOST bit is not set to 1.
To set the ONESHOT bit to 0, first write 0 to the RECREQ bit and ensure that it has been set to 0.

(2) One-Shot Transmit Mode

When the ONESHOT bit is set to 1 in transmit mode (RECREQ bit = 0 and TRMREQ bit = 1), the CAN module transmits a message only one time. The CAN module does not transmit the message again if a CAN bus error or CAN bus arbitration lost occurs. When transmission is completed, the SENTDATA bit is set to 1. If transmission is not completed due to a CAN bus error or CAN bus arbitration lost, the TRMABT bit is set to 1.
Set the ONESHOT bit to 0 after the SENTDATA or TRMABT bit is set to 1.

### 25.1.9.8 RECREQ Bit

The RECREQ bit selects receive modes shown in Table 25.11.
When the RECREQ bit is set to 1, the corresponding mailbox is configured for reception of a data frame or a remote frame.
When this bit is set to 0, the corresponding mailbox is not configured for reception of a data frame or a remote frame.
Due to HW protection, the RECREQ bit cannot be set to 0 by writing 0 by a program during the following period:
HW protection is started
- from the acceptance filter procedure (the beginning of the CRC field)

HW protection is released
- for the mailbox that is specified to receive the incoming message, after the received data is stored into the mailbox or a CAN bus error occurs (i.e. a maximum period of HW protection is from the beginning of the CRC field to the end of the 7th bit of EOF).
- for the other mailboxes, after the acceptance filter procedure.
- if no mailbox is specified to receive the message, after the acceptance filter procedure.

When setting the RECREQ bit to 1, do not set 1 to the TRMREQ bit.
To change the configuration of a mailbox from transmission to reception, first abort the transmission and then set bits SENTDATA and TRMABT to 0 before changing to reception.

#### 25.1.9.9 TRMREQ Bit

The TRMREQ bit selects transmit modes shown in Table 25.11.

When this bit is set to 1, the corresponding mailbox is configured for transmission of a data frame or a remote frame.

When this bit is set to 0, the corresponding mailbox is not configured for transmission of a data frame or a remote frame.

If the TRMREQ bit is changed from 1 to 0 to cancel the corresponding transmission request, either the TRMABT or SENTDATA bit is set to 1.

When setting the TRMREQ bit to 1, do not set the RECREQ bit to 1.

To change the configuration of a mailbox from reception to transmission, first abort the reception and then set bits NEWDATA and MSGLOST to 0 before changing to transmission.

### 25.1.10 CANi Receive FIFO Control Register (CiRFCR Register) (i = 0, 1)

CANi Receive FIFO Control Register (i = 0, 1) (1)


| Symbol | Address | Reset Value |
|---|---|---|
| C0RFCR | 47F48h | 1000 0000b |
| C1RFCR | 47B48h | 1000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| RFE | Receive FIFO Enable Bit (2) | 0: Receive FIFO disabled<br>1: Receive FIFO enabled | RW |
| RFUST | Receive FIFO Unread Message Number Status Bit | b3 b2 b1<br>0 0 0 : No unread message<br>0 0 1 : 1 unread message<br>0 1 0 : 2 unread messages<br>0 1 1 : 3 unread messages<br>1 0 0 : 4 unread messages<br>1 0 1 : Reserved<br>1 1 0 : Reserved<br>1 1 1 : Reserved | RO |
| RFMLF | Receive FIFO Message Lost Flag (3) | 0: No receive FIFO message lost has occurred<br>1: Receive FIFO message lost has occurred | RW |
| RFFST | Receive FIFO Full Status Bit | 0: Receive FIFO is not full<br>1: Receive FIFO is full (4 unread messages) | RO |
| RFWST | Receive FIFO Buffer Warning Status Bit | 0: Receive FIFO is not buffer warning<br>1: Receive FIFO is buffer warning (3 unread messages) | RO |
| RFEST | Receive FIFO Empty Status Bit | 0: Unread message in receive FIFO<br>1: No unread message in receive FIFO | RO |

Notes:
1. Write to the CiRFCR register in CAN operation mode or CAN halt mode.
2. Write 0 to the RFE bit simultaneously with the RFMLF bit.
3. Write 0 only. Writing 1 has no effect.

**Figure 25.11 Registers C0RFCR and C1RFCR**

#### 25.1.10.1 RFE Bit

When the RFE bit is set to 1, the receive FIFO is enabled.
When this bit is set to 0, the receive FIFO is disabled for reception and becomes empty (RFEST bit = 1).
Do not set this bit to 1 in normal mailbox mode (MBM bit in the CiCTLR register (i = 0, 1) = 0).
Due to HW protection, the RFE bit is not set to 0 by writing 0 by a program during the following period:
HW protection is started

- from the acceptance filter procedure (the beginning of the CRC field)

HW protection is released

- if the receive FIFO is specified to receive the incoming message, after the received data is stored into the receive FIFO or a CAN bus error occurs. (i.e. a maximum period of HW protection is from the beginning of the CRC field to the end of the 7th bit of EOF.)
- if the receive FIFO is not specified to receive the message, after the acceptance filter procedure.

#### 25.1.10.2 RFUST Bit

The RFUST bit indicates the number of unread messages in the receive FIFO.
The value of this bit is initialized to 000b when the RFE bit is set to 0.

#### 25.1.10.3 RFMLF Bit

The RFMLF bit is set to 1 (receive FIFO message lost has occurred) when the receive FIFO receives a new message and the receive FIFO is full. The timing for setting this bit to 1 is at the end of the 6th bit of EOF.
The RFMLF bit is set to 0 by writing 0 by a program.
In both overwrite and overrun modes, this bit cannot be set to 0 (receive FIFO message lost has not occurred) by writing 0 by a program due to HW protection during the five cycles of fCAN following the 6th bit of EOF, if the receive FIFO is full and determined to receive the message.

#### 25.1.10.4 RFFST Bit

The RFFST bit is set to 1 (receive FIFO is full) when the number of unread messages in the receive FIFO is 4. This bit is set to 0 (receive FIFO is not full) when the number of unread messages in the receive FIFO is less than 4. This bit is set to 0 when the RFE bit is 0.

#### 25.1.10.5 RFWST Bit

The RFWST bit is set to 1 (receive FIFO is buffer warning) when the number of unread messages in the receive FIFO is 3. This bit is set to 0 (receive FIFO is not buffer warning) when the number of unread messages in the receive FIFO is less than 3 or equal to 4. This bit is set to 0 when the RFE bit is 0.

#### 25.1.10.6 RFEST Bit

The RFEST bit is 1 (no unread message in receive FIFO) when the number of unread messages in the receive FIFO is 0. This bit is set to 1 when the RFE bit is set to 0. The RFEST bit is set to 0 (unread message in receive FIFO) when the number of unread messages in the receive FIFO is one or more.

Figure 25.12 shows the receive FIFO mailbox operation.

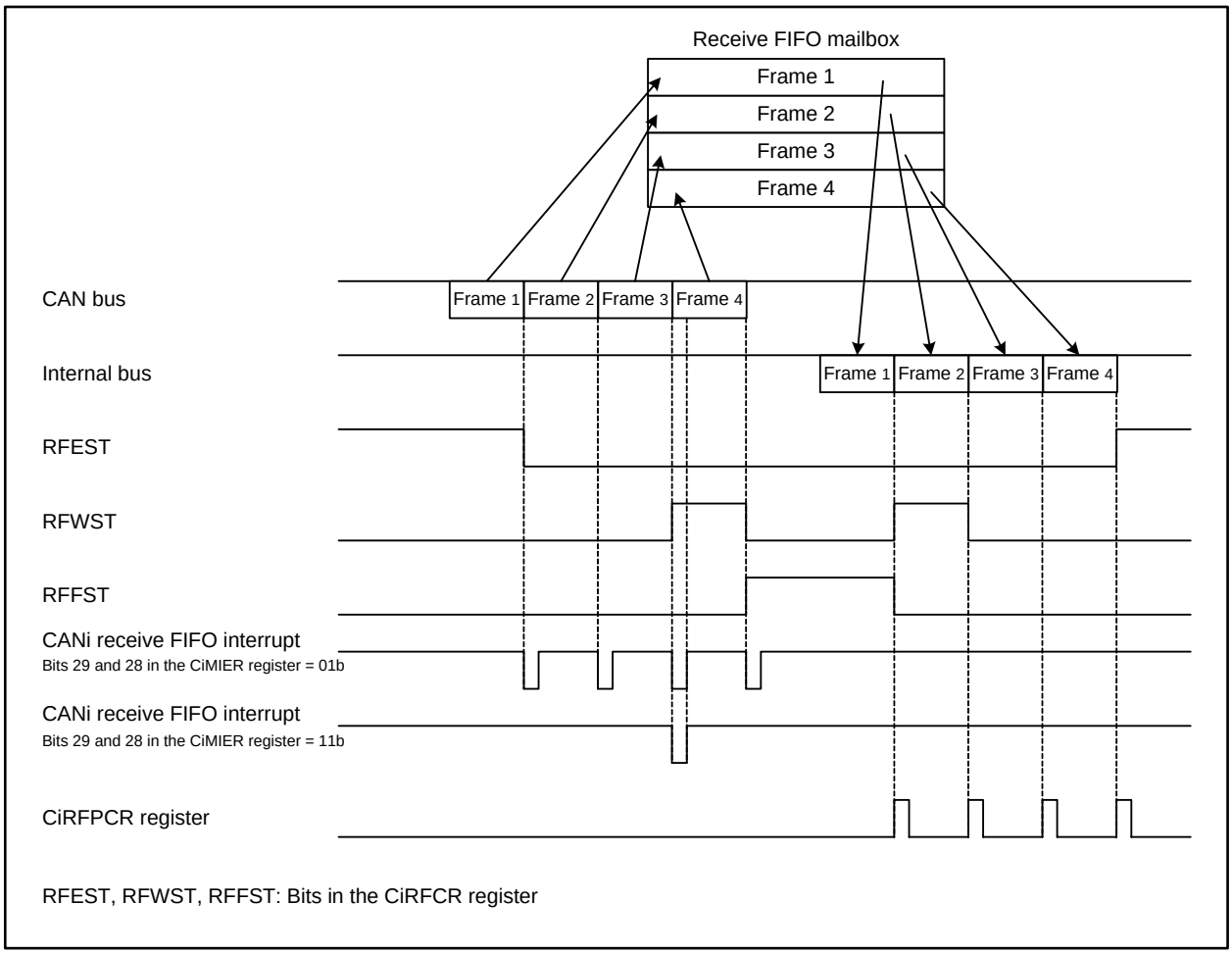
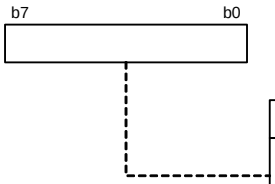



**Figure 25.12 Receive FIFO Mailbox Operation (Bits 29 and 28 in CiMIER Register = 01b and 11b) (i = 0, 1)**

### 25.1.11 CANi Receive FIFO Pointer Control Register (CiRFPCR Register) (i = 0, 1)

CANi Receive FIFO Pointer Control Register (i = 0, 1)




| Symbol | Address | Reset Value |
|---|---|---|
| C0RFPCR | 47F49h | Undefined |
| C1RFPCR | 47B49h | Undefined |

| Function | Setting Value | RW |
|---|---|---|
| The CPU-side pointer for the receive FIFO is incremented by writing FFh | FFh | WO |

**Figure 25.13 Registers C0RFPCR and C1RFPCR**

When the receive FIFO is not empty, write FFh to the CiRFPCR register by a program to increment the CPU-side pointer for the receive FIFO to the next mailbox location.
Do not write to the CiRFPCR register when the RFE bit in the CiRFCR register is 0 (receive FIFO disabled).
Both the CAN-side pointer and the CPU-side pointer are incremented when a new message is received and the RFFST bit is 1 (receive FIFO is full) in overwrite mode. When the RFMLF bit is 1 in this condition, the CPU-side pointer cannot be incremented by writing to the CiRFPCR register by a program.

### 25.1.12 CANi Transmit FIFO Control Register (CiTFCR Register) (i = 0, 1)

CANi Transmit FIFO Control Register (i = 0, 1) (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | 0 | X | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| C0TFCR | 47F4Ah | 1000 0000b |
| C1TFCR | 47B4Ah | 1000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TFE | Transmit FIFO Enable Bit | 0: Transmit FIFO disabled<br>1: Transmit FIFO enabled | RW |
| TFUST | Transmit FIFO Unsent Message Number Status Bit | b3 b2 b1<br>0 0 0 : No unsent message<br>0 0 1 : 1 unsent message<br>0 1 0 : 2 unsent messages<br>0 1 1 : 3 unsent messages<br>1 0 0 : 4 unsent messages<br>1 0 1 : Reserved<br>1 1 0 : Reserved<br>1 1 1 : Reserved | RO |
| — (b4) | No register bit; should be written with 0 and read as 0 | | — |
| — (b5) | Reserved | Should be written with 0 and read as undefined value | RO |
| TFFST | Transmit FIFO Full Status Bit | 0: Transmit FIFO is not full<br>1: Transmit FIFO is full (4 unsent messages) | RO |
| TFEST | Transmit FIFO Empty Status Bit | 0: Unsent message in transmit FIFO<br>1: No unsent message in transmit FIFO | RO |

Note:
1. Write to the CiTFCR register in CAN operation mode or CAN halt mode.

**Figure 25.14 Registers C0TFCR and C1TFCR**

#### 25.1.12.1 TFE Bit

When the TFE bit is set to 1, the transmit FIFO is enabled.
When this bit is set to 0, the transmit FIFO becomes empty (TFEST bit = 1) and then unsent messages from the transmit FIFO are lost as described below:

- If a message from the transmit FIFO is not scheduled for the next transmission or during transmission.
- Following the completion of transmission, a CAN bus error, CAN bus arbitration lost, or entry to CAN halt mode if a message from the transmit FIFO is scheduled for the next transmission or already during transmission.

Before setting the TFE bit to set to 1 again, ensure that the TFEST bit has been set to 1.
After setting the TFE bit to 1, write transmit data into the CiMB24 register.
Do not set this bit to 1 in normal mailbox mode (MBM bit in the CiCTLR register = 0).

#### 25.1.12.2 TFUST Bit

The TFUST bit indicates the number of unsent messages in the transmit FIFO.
After the TFE bit is set to 0, the value of the TFUST bit is initialized to 000b when transmission abort or transmission is completed.

### 25.1.12.3 TFFST Bit

The TFFST bit is set to 1 (transmit FIFO is full) when the number of unsent messages in the transmit FIFO is 4. This bit is set to 0 (transmit FIFO is not full) when the number of unsent messages in the transmit FIFO is less than 4. This bit is set to 0 when transmission from the transmit FIFO has been aborted.

### 25.1.12.4 TFEST Bit

The TFEST bit is set to 1 (no message in transmit FIFO) when the number of unsent messages in the transmit FIFO is 0. This bit is set to 1 when transmission from the transmit FIFO has been aborted.
The TFEST bit is set to 0 (message in transmit FIFO) when the number of unsent messages in the transmit FIFO is not 0.

Figure 25.15 shows the transmit FIFO mailbox operation.

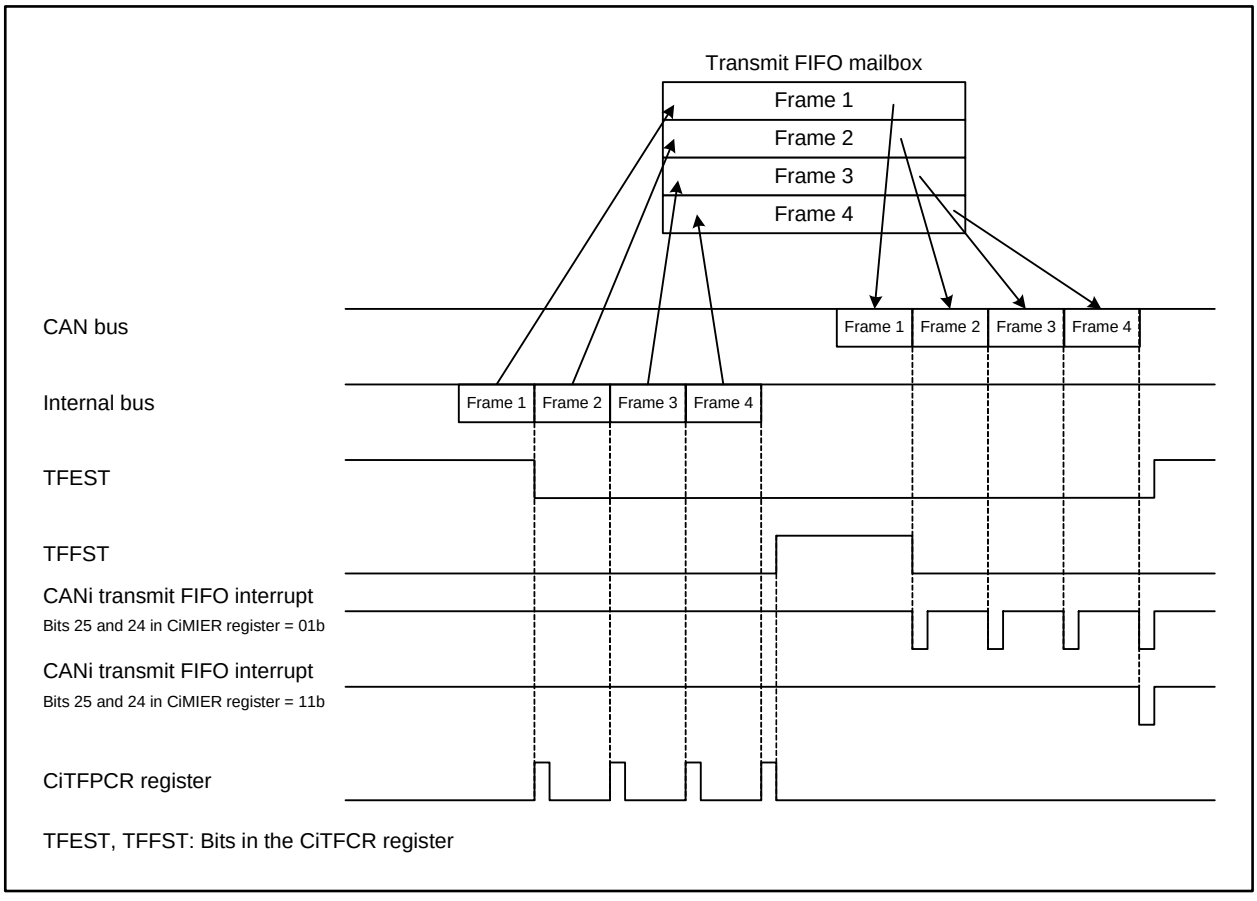
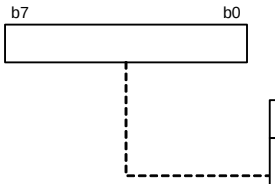



**Figure 25.15 Transmit FIFO Mailbox Operation (Bits 25 and 24 in CiMIER Register = 01b and 11b) (i = 0, 1)**

### 25.1.13 CANi Transmit FIFO Pointer Control Register (CiTFPCR Register) (i = 0, 1)

CANi Transmit FIFO Pointer Control Register (i = 0, 1)




| Symbol | Address | Reset Value |
|---|---|---|
| C0TFPCR | 47F4Bh | Undefined |
| C1TFPCR | 47B4Bh | Undefined |

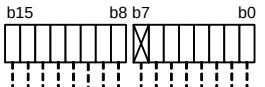
| Function | Setting Value | RW |
|---|---|---|
| The CPU-side pointer for the transmit FIFO is incremented by writing FFh | FFh | WO |

**Figure 25.16 Registers C0TFPCR and C1TFPCR**

When the transmit FIFO is not full, write FFh to the CiTFPCR register by a program to increment the CPU-side pointer for the transmit FIFO to the next mailbox location.
Do not write to the CiTFPCR register when the TFE bit in the CiTFPCR register is 0 (transmit FIFO disabled).

### 25.1.14 CANi Status Register (CiSTR Register) (i = 0, 1)

CANi Status Register (i = 0, 1)




| Symbol | Address | Reset Value |
|---|---|---|
| C0STR | 47F43h-47F42h | 0000 0000 0000 0101b |
| C1STR | 47B43h-47B42h | 0000 0000 0000 0101b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| RSTST | CAN Reset Status Flag | 0: Not in CAN reset mode<br>1: In CAN reset mode | RO |
| HLTST | CAN Halt Status Flag | 0: Not in CAN halt mode<br>1: In CAN halt mode | RO |
| SLPST | CAN Sleep Status Flag | 0: Not in CAN sleep mode<br>1: In CAN sleep mode | RO |
| EPST | Error-Passive Status Flag | 0: Not in error-passive state<br>1: In error-passive state | RO |
| BOST | Bus-Off Status Flag | 0: Not in bus-off state<br>1: In bus-off state | RO |
| TRMST | Transmit Status Flag (transmitter) | 0: Bus idle or reception in progress<br>1: Transmission in progress or in bus-off state | RO |
| RECST | Receive Status Flag (receiver) | 0: Bus idle or transmission in progress<br>1: Reception in progress | RO |
| — (b7) | No register bit; should be written with 0 and read as 0 | | — |
| NDST | NEWDATA Status Flag | 0: No mailbox with NEWDATA bit = 1<br>1: Mailbox(es) with NEWDATA bit = 1 | RO |
| SDST | SENTDATA Status Flag | 0: No mailbox with SENTDATA bit = 1<br>1: Mailbox(es) with SENTDATA bit = 1 | RO |
| RFST | Receive FIFO Status Flag | 0: No message in receive FIFO<br>1: Message in receive FIFO | RO |
| TFST | Transmit FIFO Status Flag | 0: Transmit FIFO is full<br>1: Transmit FIFO is not full | RO |
| NMLST | Normal Mailbox Message Lost Status Flag | 0: No mailbox with MSGLOST bit = 1<br>1: Mailbox(es) with MSGLOST bit = 1 | RO |
| FMLST | FIFO Mailbox Message Lost Status Flag | 0: RFMLF bit = 0<br>1: RFMLF bit = 1 | RO |
| TABST | Transmission Abort Status Flag | 0: No mailbox with TRMABT bit = 1<br>1: Mailbox(es) with TRMABT bit = 1 | RO |
| EST | Error Status Flag | 0: No error occurred<br>1: Error occurred | RO |

**Figure 25.17 Registers C0STR and C1STR**

#### 25.1.14.1 RSTST Bit

The RSTST bit is set to 1 when the CAN module is in CAN reset mode.
This bit is set to 0 when the CAN module is not in CAN reset mode.
Even when the state is changed from CAN reset mode to CAN sleep mode, the RSTST bit remains 1.

#### 25.1.14.2 HLTST Bit

The HLTST bit is set to 1 when the CAN module is in CAN halt mode.
This bit is set to 0 when the CAN module is not in CAN halt mode.
Even when the state is changed from CAN halt mode to CAN sleep mode, the HLTST bit remains 1.

#### 25.1.14.3 SLPST Bit

The SLPST bit is set to 1 when the CAN module is in CAN sleep mode.
This bit is set to 0 when the CAN module is not in CAN sleep mode.

#### 25.1.14.4 EPST Bit

The EPST bit is set to 1 when the value of the CiTECR or CiRECR register (i = 0, 1) exceeds 127 and the CAN module is in error-passive state ($128 \leq TEC < 256$ or $128 \leq REC < 256$). This bit is set to 0 when the CAN module is not in error-passive state.
TEC indicates the value of the transmit error counter (CiTECR register) and REC indicates the value of the receive error counter (CiRECR register).

#### 25.1.14.5 BOST Bit

The BOST bit is set to 1 when the value of the CiTECR register exceeds 255 and the CAN module is in bus-off state ($TEC \geq 256$). This bit is set to 0 when the CAN module is not in bus-off state.

#### 25.1.14.6 TRMST Bit

The TRMST bit is set to 1 when the CAN module performs as a transmitter node or is in bus-off state.
This bit is set to 0 when the CAN module performs as a receiver node or is in bus-idle state.

#### 25.1.14.7 RECST Bit

The RECST bit is set to 1 when the CAN module performs as a receiver node.
This bit is set to 0 when the CAN module performs as a transmitter node or is in bus-idle state.

#### 25.1.14.8 NDST Bit

The NDST bit is set to 1 when at least one NEWDATA bit in the CiMCTLj register (j = 0 to 31) is 1 regardless of the value of the CiMIER register.
The NDST bit is set to 0 when all NEWDATA bits are 0.

#### 25.1.14.9 SDST Bit

The SDST bit is set to 1 when at least one SENTDATA bit in the CiMCTLj register (i = 0, 1; j = 0 to 31) is 1 regardless of the value of the CiMIER register.
The SDST bit is set to 0 when all SENTDATA bits are 0.

#### 25.1.14.10 RFST Bit

The RFST bit is set to 1 when the receive FIFO is not empty.
This bit is set to 0 when the receive FIFO is empty.
This bit is set to 0 when normal mailbox mode is selected.

#### 25.1.14.11 TFST Bit

The TFST bit is set to 1 when the transmit FIFO is not full.
This bit is set to 0 when the transmit FIFO is full.
This bit is set to 0 when normal mailbox mode is selected.

#### 25.1.14.12 NMLST Bit

The NMLST bit is set to 1 when at least one MSGLOST bit in the CiMCTLj register is 1 regardless of the value of the CiMIER register.
The NMLST bit is set to 0 when all MSGLOST bits are 0.

#### 25.1.14.13 FMLST Bit

The FMLST bit is set to 1 when the RFMLF bit in the CiRFCR register is 1 regardless of the value of the CiMIER register.
The FMLST bit is set to 0 when the RFMLF bit is 0.

#### 25.1.14.14 TABST Bit

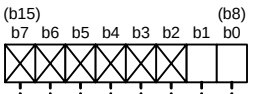
The TABST bit is set to 1 when at least one TRMABT bit in the CiMCTLj register is 1 regardless of the value of the CiMIER register.
The TABST bit is set to 0 when all TRMABT bits are 0.

#### 25.1.14.15 EST Bit

The EST bit is 1 when at least one error is detected by the CiEIFR register regardless of the value of the CiEIER register.
This bit is set to 0 when no error is detected by the CiEIFR register.

### 25.1.15 CANi Mailbox Search Mode Register (CiMSMR Register) (i = 0, 1)

CANi Mailbox Search Mode Register (i = 0, 1) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| C0MSMR | 47F53h | 0000 0000b |
| C1MSMR | 47B53h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| MBSM | Mailbox Search Mode Select Bit | b1 b0<br>0 0 : Receive mailbox search mode<br>0 1 : Transmit mailbox search mode<br>1 0 : Message lost search mode<br>1 1 : Channel search mode | RW |
| —<br>(b7-b2) | No register bits; should be written with 0 and read as 0 | | — |

Note:
1. Write to the CiMSMR register in CAN operation mode or CAN halt mode.

**Figure 25.18 Registers C0MSMR and C1MSMR**

#### 25.1.15.1 MBSM Bit

The MBSM bit selects the search mode for the mailbox search function.

When this bit is 00b, receive mailbox search mode is selected. In this mode, the search targets are the NEWDATA bit in the CiMCTLj register (j = 0 to 31) for the normal mailbox and the RFEST bit in the CiRFCR register.

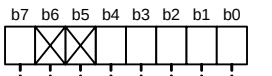
When the MBSM bit is 01b, transmit mailbox search mode is selected. In this mode, the search target is the SENTDATA bit in the CiMCTLj register.

When the MBSM bit is 10b, message lost search mode is selected. In this mode, the search targets are the MSGLOST bit in the CiMCTLj register for the normal mailbox and the RFMLF bit in the CiRFCR register.

When the MBSM bit is 11b, channel search mode is selected. In this mode, the search target is the CiCSSR register. Refer to 25.1.17 “CANi Channel Search Support Register (CiCSSR Register) (i = 0, 1)”.

### 25.1.16 CANi Mailbox Search Status Register (CiMSSR Register) (i = 0, 1)

CANi Mailbox Search Status Register (i = 0, 1)




| Symbol | Address | Reset Value |
|---|---|---|
| C0MSSR | 47F52h | 1000 0000b |
| C1MSSR | 47B52h | 1000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| MBNST | Search Result Mailbox Number Status Bit | Output of search result in each search mode<br>Output number: 0 to 31 | RO |
| —<br>(b6-b5) | No register bits; should be written with 0 and read as 0 | | — |
| SEST | Search Result Status Bit | 0: Search result found<br>1: No search result | RO |

**Figure 25.19 Registers C0MSSR and C1MSSR**

### 25.1.16.1 MBNST Bit

The MBNST bit outputs the smallest mailbox number that is searched in each mode of the CiMSMR register (i = 0, 1).
In receive mailbox, transmit mailbox, and message lost search modes, the value of the mailbox i.e.,the search result to be output, is updated as described below:

- When the NEWDATA, SENTDATA, or MSGLOST bit for the output mailbox is set to 0.
- When the NEWDATA, SENTDATA, or MSGLOST bit for a higher-priority mailbox is set to 1.

In receive mailbox search and message lost search modes, the receive FIFO (mailbox [28]) is output when the receive FIFO is not empty and there are no unread received messages or no lost messages in any of the normal mailboxes (mailboxes [0] to [23]).
In transmit mailbox search mode, the transmit FIFO (mailbox [24]) is not output.
Table 25.7 lists the behavior of MBNST bit in FIFO mailbox mode.

**Table 25.7 Behavior of MBNST Bit in FIFO Mailbox Mode**

| MBSM Bit | Mailbox [24] (Transmit FIFO) | Mailbox [28] (Receive FIFO) |
|---|---|---|
| 00b | Mailbox [24] is not output. | Mailbox [28] is output when no NEWDATA bit for the normal mailbox is set to 1 and the receive FIFO is not empty. |
| 01b | | Mailbox [28] is not output. |
| 10b | | Mailbox [28] is output when no MSGLOST bit for the normal mailbox is set to 1 and the RFMLF bit is set to 1 in the receive FIFO. |
| 11b | | Mailbox [28] is not output. |

In channel search mode, the MBNST bit outputs the corresponding channel number. After the CiMSSR register is read by a program, the next target channel number is output.

### 25.1.16.2 SEST Bit

The SEST bit is set to 1 when no corresponding mailbox is found after searching all mailboxes.
For example, in transmit mailbox search mode, the SEST bit is set to 1 when no SENTDATA bit for mailboxes is 1. The SEST bit is set to 0 when at lease one SENTDATA bit is 1.
When the SEST bit is 1, the value of the MBNST bit is undefined.

### 25.1.17 CANi Channel Search Support Register (CiCSSR Register) (i = 0, 1)

CANi Channel Search Support Register (i = 0, 1) (1, 2)

b7 b0

| Symbol | Address | Reset Value |
|---|---|---|
| C0CSSR | 47F51h | Undefined |
| C1CSSR | 47B51h | Undefined |

| Function | Setting Value | RW |
|---|---|---|
| When the value of the channel search is input, the channel number is output to the CiMSSR register | Channel value | RW |

Notes:
1. Write to the CiCSSR register only when the MBSM bit in the CiMSMR register is 11b (channel search mode).
2. Write to the CiCSSR register in CAN operation mode and CAN halt mode.

**Figure 25.20 Registers C0CSSR and C1CSSR**

The bits in the CiCSSR register, which are set to 1, are encoded by an 8/3 encoder (the lower bit position, the higer priority) and output to the MBNST bits in the CiMSSR register.
The CiMSSR register outputs the updated value whenever the CiMSSR register is read by a program.
Figure 25.21 shows the write and read of registers CiCSSR and CiMSSR.

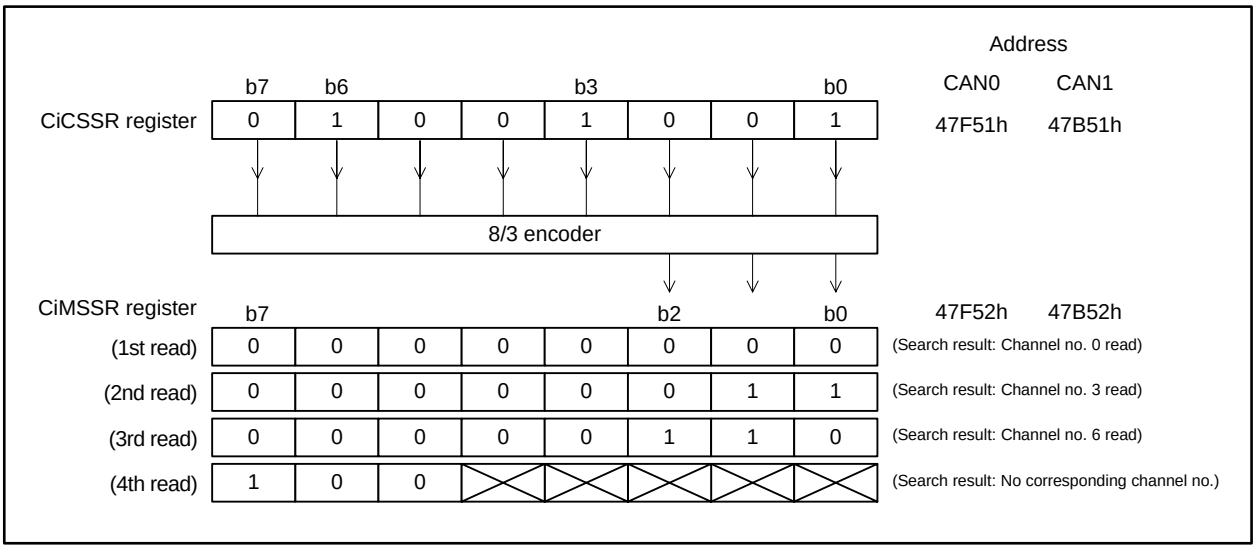
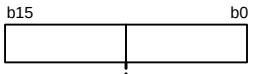



**Figure 25.21 Write and Read of Registers CiCSSR and CiMSSR (i = 0, 1)**

The value of the CiCSSR register is also updated whenever the CiMSSR register is read. When the CiCSSR register is read, the value before the 8/3 encoder conversion is read.

### 25.1.18 CANi Acceptance Filter Support Register (CiAFSR Register) (i = 0, 1)

CANi Acceptance Filter Support Register (i = 0, 1) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| C0AFSR | 47F57h-47F56h | Undefined |
| C1AFSR | 47B57h-47B56h | Undefined |

| Function | Setting Value | RW |
|---|---|---|
| After the standard ID of a received message is written, the value converted for data table search can be read | Standard ID/ converted value | RW |

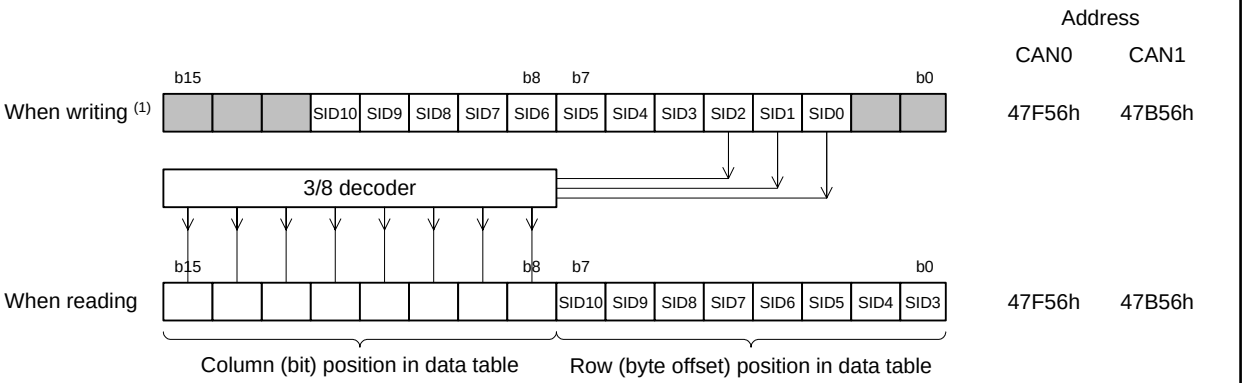
Note:
1. Write to the CiAFSR register in CAN operation mode or CAN halt mode.

**Figure 25.22 Registers C0AFSR and C1AFSR**

The acceptance filter support unit (ASU) can be used for data table (8 bits × 256) search. In the data table, all standard IDs created by the user are set to be valid/invalid in bit units. When the CiAFSR register is written with the 16-bit unit data including the SID bit in the CiMBj register (j = 0 to 31), in which a received ID is stored, a decoded row (byte offset) position and column (bit) position for data table search can be read. The ASU can be used for standard (11-bit) IDs only.
The ASU is enabled in the following cases:

- When the ID to receive cannot be masked by the acceptance filter.
  Example) IDs to receive: 078h, 087h, 111h
- When there are too many IDs to receive and software filtering time is expected to be shortened.

Figure 25.23 shows the write and read of CiAFSR register.




Note:
1. Write the same value as the 16-bit unit data including the SID bit in the CiMBj (j = 0 to 31) register.

**Figure 25.23 Write and Read of CiAFSR Register (i = 0, 1)**

### 25.1.19 CANi Error Interrupt Enable Register (CiEIER Register) (i = 0, 1)

CANi Error Interrupt Enable Register (i = 0, 1) (1)


| Symbol | Address | Reset Value |
|---|---|---|
| C0EIER | 47F4Ch | 00h |
| C1EIER | 47B4Ch | 00h |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| BEIE | Bus Error Interrupt Enable Bit | 0: Bus error interrupt disabled<br>1: Bus error interrupt enabled | RW |
| EWIE | Error Warning Interrupt Enable Bit | 0: Error warning interrupt disabled<br>1: Error warning interrupt enabled | RW |
| EPIE | Error Passive Interrupt Enable Bit | 0: Error passive interrupt disabled<br>1: Error passive interrupt enabled | RW |
| BOEIE | Bus-Off Entry Interrupt Enable Bit | 0: Bus-off entry interrupt disabled<br>1: Bus-off entry interrupt enabled | RW |
| BORIE | Bus-Off Recovery Interrupt Enable Bit | 0: Bus-off recovery interrupt disabled<br>1: Bus-off recovery interrupt enabled | RW |
| ORIE | Receive Overrun Interrupt Enable Bit | 0: Receive overrun interrupt disabled<br>1: Receive overrun interrupt enabled | RW |
| OLIE | Overload Frame Transmit Interrupt Enable Bit | 0: Overload frame transmit interrupt disabled<br>1: Overload frame transmit interrupt enabled | RW |
| BLIE | Bus Lock Interrupt Enable Bit | 0: Bus lock interrupt disabled<br>1: Bus lock interrupt enabled | RW |

Note:
1. Write to the CiEIER register in CAN reset mode.

**Figure 25.24 Registers C0EIER and C1EIER**

The CiEIER register is used to set the error interrupt enabled/disabled individually for each error interrupt source in the CiEIFR register.

#### 25.1.19.1 BEIE Bit

When the BEIE bit is 0, no error interrupt request is generated even if the BEIF bit in the CiEIFR register (i = 0, 1) is set to 1.
When the BEIE bit is 1, an error interrupt request is generated if the BEIF bit is set to 1.

#### 25.1.19.2 EWIE Bit

When the EWIE bit is 0, no error interrupt request is generated even if the EWIF bit in the CiEIFR register is set to 1.
When the EWIE bit is 1, an error interrupt request is generated if the EWIF bit is set to 1.

#### 25.1.19.3 EPIE Bit

When the EPIE bit is 0, no error interrupt request is generated even if the EPIF bit in the CiEIFR register is set to 1.
When the EPIE bit is 1, an error interrupt request is generated if the EPIF bit is set to 1.

#### 25.1.19.4 BOEIE Bit

When the BOEIE bit is 0, no error interrupt request is generated even if the BOEIF bit in the CiEIFR register is set to 1.
When the BOEIE bit is 1, an error interrupt request is generated if the BOEIF bit is set to 1.

#### 25.1.19.5 BORIE Bit

When the BORIE bit is 0, an error interrupt request is not generated even if the BORIF bit in the CiEIFR register is set to 1.
When the BORIE bit is 1, an error interrupt request is generated if the BORIF bit is set to 1.

#### 25.1.19.6 ORIE Bit

When the ORIE bit is 0, no error interrupt request is generated even if the ORIF bit in the CiEIFR register is set to 1.
When the ORIE bit is 1, an error interrupt request is generated if the ORIF bit is set to 1.

#### 25.1.19.7 OLIE Bit

When the OLIE bit is 0, no error interrupt request is generated even if the OLIF bit in the CiEIFR register is set to 1.
When the OLIE bit is 1, an error interrupt request is generated if the OLIF bit is set to 1.

#### 25.1.19.8 BLIE Bit

When the BLIE bit is 0, no error interrupt request is generated even if the BLIF bit in the CiEIFR register is set to 1.
When the BLIE bit is 1, an error interrupt request is generated if the BLIF bit is set to 1.

### 25.1.20 CANi Error Interrupt Factor Judge Register (CiEIFR Register) (i = 0, 1)

CANi Error Interrupt Factor Judge Register (i = 0, 1) (1)


| Symbol | Address | Reset Value |
|---|---|---|
| C0EIFR | 47F4Dh | 00h |
| C1EIFR | 47B4Dh | 00h |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| BEIF | Bus Error Detect Flag | 0: No bus error detected<br>1: Bus error detected | RW |
| EWIF | Error Warning Detect Flag | 0: No error warning detected<br>1: Error warning detected | RW |
| EPIF | Error Passive Detect Flag | 0: No error passive detected<br>1: Error passive detected | RW |
| BOEIF | Bus-Off Entry Detect Flag | 0: No bus-off entry detected<br>1: Bus-off entry detected | RW |
| BORIF | Bus-Off Recovery Detect Flag | 0: No bus-off recovery detected<br>1: Bus-off recovery detected | RW |
| ORIF | Receive Overrun Detect Flag | 0: No receive overrun detected<br>1: Receive overrun detected | RW |
| OLIF | Overload Frame Transmission Detect Flag | 0: No overload frame transmission detected<br>1: Overload frame transmission detected | RW |
| BLIF | Bus Lock Detect Flag | 0: No bus lock detected<br>1: Bus lock detected | RW |

Note:
1. When writing 0 to these bits by a program, use the MOV instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1. Writing 1 has no effect to these bit values.

**Figure 25.25 Registers C0EIFR and C1EIFR**

If an event corresponding to each bit occurs, the corresponding bit in the CiEIFR register is set to 1 regardless of the setting of the CiEIER register.
To set each bit to 0, write 0 by a program. If the set timing occurs simultaneously with the clear timing by the program, the bit becomes 1.

#### 25.1.20.1 BEIF Bit

The BEIF bit is set to 1 when a bus error is detected.

#### 25.1.20.2 EWIF Bit

The EWIF bit is set to 1 when the value of the receive error counter (REC) or transmit error counter (TEC) exceeds 95.
This bit is set to 1 only when the REC or TEC initially exceeds 95. Thus, if 0 is written to the EWIF bit by a program while the REC or TEC remains greater than 95, this bit is not set to 1 until the REC and the TEC go below 95 and then exceed 95 again.

### 25.1.20.3 EPIF Bit

The EPIF bit is set to 1 when the CAN error state becomes error-passive (the REC or TEC value exceeds 127).
This bit is set to 1 only when the REC or TEC initially exceeds 127. Thus, if 0 is written to the EPIF bit by a program while the REC or TEC remains greater than 127, this bit is not set to 1 until the REC and the TEC go below 127 and then exceed 127 again.

### 25.1.20.4 BOEIF Bit

The BOEIF bit is set to 1 when the CAN error state becomes bus-off (the TEC value exceeds 255).
This bit is also set to 1 when the BOM bit in the CiCTLR register (i = 0, 1) is 01b (entry to CAN halt mode automatically at bus-off entry) and the CAN module becomes the bus-off state.

### 25.1.20.5 BORIF Bit

The BORIF bit is set to 1 when the CAN module recovers from the bus-off state normally by detecting 11 consecutive bits 128 times in the following conditions:
(1) When the BOM bit in the CiCTLR register is 00b.
(2) When the BOM bit is 10b.
(3) When the BOM bit is 11b.

The BORIF bit is not set to 1 if the CAN module recovers from the bus-off state in the following conditions:
(1) When the CANM bit in the CiCTLR register is set to 01b (CAN reset mode).
(2) When the RBOC bit in the CiCTLR register is set to 1 (forcible return from bus-off).
(3) When the BOM bit is 01b.
(4) When the BOM bit is 11b and the CANM bit is set to 10b (CAN halt mode) before normal recovery occurs.

Table 25.8 lists the behavior of bits BOEIF and BORIF according to BOM bit setting value.

**Table 25.8 Behavior of Bits BOEIF and BORIF according to BOM Bit Setting Value**

| BOM Bit | BOEIF Bit | BORIF Bit |
|---|---|---|
| 00b | Set to 1 on entry to the bus-off state. | Set to 1 on exit from the bus-off state. |
| 01b | | Do not set to 1. |
| 10b | | Set to 1 on exit from the bus-off state. |
| 11b | | Set to 1 if normal bus-off recovery occurs before the CANM bit is set to 10b (CAN halt mode). |

### 25.1.20.6 ORIF Bit

The ORIF bit is set to 1 when a receive overrun occurs.
This bit is not to set to 1 in overwrite mode. In overwrite mode, a reception complete interrupt request is generated if an overwrite condition occurs and this bit is not set to 1.
In normal mailbox mode, if an overrun occurs in any of mailboxes [0] to [31] in overrun mode, this bit is set to 1.
In FIFO mailbox mode, if an overrun occurs in any of mailboxes [0] to [23] or the receive FIFO in overrun mode, this bit is set to 1.

### 25.1.20.7 OLIF Bit

The OLIF bit is set to 1 if the transmitting condition of an overload frame is detected when the CAN module performs transmission or reception.

### 25.1.20.8 BLIF Bit

The BLIF bit is set to 1 if 32 consecutive dominant bits are detected on the CAN bus while the CAN module is in CAN operation mode.

After the BLIF bit is set to 1, 32 consecutive dominant bits are detected again under either of the following conditions:

- After this bit is set to 0 from 1, recessive bits are detected.
- After this bit is set to 0 from 1, the CAN module enters CAN reset mode or CAN halt mode and then enters CAN operation mode again.

### 25.1.21 CANi Receive Error Count Register (CiRECR Register) (i = 0, 1)

CANi Receive Error Count Register (i = 0, 1)

b7 b0

| Symbol | Address | Reset Value |
|---|---|---|
| C0RECR | 47F4Eh | 00h |
| C1RECR | 47B4Eh | 00h |

| Function | Counter Value | RW |
|---|---|---|
| Receive error count function<br>The CiRECR register increments or decrements the counter value according to the error status of the CAN module during reception | 00h to FFh (1) | RO |

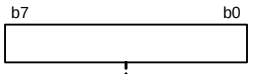
Note:
1. The value in bus-off state is undefined.

**Figure 25.26 Registers C0RECR and C1RECR**

The CiRECR register indicates the value of the receive error counter.
Refer to the CAN Specifications (ISO11898-1) about the increment/decrement conditions of the receive error counter.

### 25.1.22 CANi Transmit Error Count Register (CiTECR Register) (i = 0, 1)

CANi Transmit Error Count Register (i = 0, 1)




| Symbol | Address | Reset Value |
|---|---|---|
| C0TECR | 47F4Fh | 00h |
| C1TECR | 47B4Fh | 00h |

| Function | Counter Value | RW |
|---|---|---|
| Transmit error count function<br>The CiTECR register increments or decrements the counter value according to the error status of the CAN module during transmission | 00h to FFh (1) | RO |

Note:
1. The value in bus-off state is undefined.

**Figure 25.27 Registers C0TECR and C1TECR**

The CiTECR register indicates the value of the TEC error counter.
Refer to the CAN Specifications (ISO11898-1) about the increment/decrement conditions of the transmit error counter.

### 25.1.23 CANi Error Code Store Register (CiECSR Register) (i = 0, 1)

CANi Error Code Store Register (i = 0, 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| C0ECSR | 47F50h | 00h |
| C1ECSR | 47B50h | 00h |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| SEF | Stuff Error Flag (1, 2) | 0: No stuff error detected<br>1: Stuff error detected | RW |
| FEF | Form Error Flag (1, 2) | 0: No form error detected<br>1: Form error detected | RW |
| AEF | ACK Error Flag (1, 2) | 0: No ACK error detected<br>1: ACK error detected | RW |
| CEF | CRC Error Flag (1, 2) | 0: No CRC error detected<br>1: CRC error detected | RW |
| BE1F | Bit Error (recessive) Flag (1, 2) | 0: No bit error detected<br>1: Bit error (recessive) detected | RW |
| BE0F | Bit Error (dominant) Flag (1, 2) | 0: No bit error detected<br>1: Bit error (dominant) detected | RW |
| ADEF | ACK Delimiter Error Bit (1, 2) | 0: No ACK delimiter error detected<br>1: ACK delimiter error detected | RW |
| EDPM | Error Display Mode Select Bit (3, 4) | 0: Output of first detected error code<br>1: Output of accumulated error code | RW |

Notes:
1. Writing 1 has no effect to these bit values.
2. When writing 0 to bits SEF, FEF, AEF, CEF, BE1F, BE0F, and ADEF by a program, use the MOV instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1.
3. Write to the EDPM bit in CAN reset mode or CAN halt mode.
4. If more than one error condition is detected simultaneously, all related bits are set to 1.

**Figure 25.28 Registers C0ECSR and C1ECSR**

The CiECSR register can be used to monitor whether an error has occurred on the CAN bus. Refer to the CAN Specifications (ISO11898-1) to check the generation conditions of each error.
To set each bit except the EDPM bit to 0, write 0 by a program. If the timing at which each bit is set to 1 and the timing at which is written by a program are the same, the relevant bit is set to 1.

#### 25.1.23.1 SEF Bit

The SEF bit is set to 1 when a stuff error is detected.

#### 25.1.23.2 FEF Bit

The FEF bit is set to 1 when a form error is detected.

#### 25.1.23.3 AEF Bit

The AEF bit is set to 1 when an ACK error is detected.

#### 25.1.23.4 CEF Bit

The CEF bit is set to 1 when a CRC error is detected.

#### 25.1.23.5 BE1F Bit

The BE1F bit is set to 1 when a recessive bit error is detected.

#### 25.1.23.6 BE0F Bit

The BE0F bit is set to 1 when a dominant bit error is detected.

#### 25.1.23.7 ADEF Bit

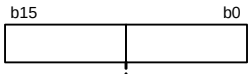
The ADEF bit is set to 1 when a form error is detected with the ACK delimiter during transmission.

#### 25.1.23.8 EDPM Bit

The EDPM bit selects the output mode of the CiECSR register (i = 0, 1).
When this bit is set to 0, the CiECSR register outputs the first error code.
When this bit is set to 1, the CiECSR register outputs the accumulated error code.

### 25.1.24 CANi Time Stamp Register (CiTSR Register) (i = 0, 1)

CANi Time Stamp Register (i = 0, 1) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| C0TSR | 47F55h-47F54h | 0000h |
| C1TSR | 47B55h-47B54h | 0000h |

| Function | Counter Value | RW |
|---|---|---|
| Free-running counter value for the time stamp function | 0000h to FFFFh | RO |

Note:
1. Read the CiTSR register in 16-bit units.

**Figure 25.29 Registers C0TSR and C1TSR**

When the CiTSR register is read, the value of the time stamp counter (16-bit free-running counter) at that moment is read.
The value of the time stamp counter reference clock is a multiple of 1 bit time, as configured by the TSPS bit in the CiCTLR register.
The time stamp counter stops in CAN sleep mode and CAN halt mode, and is initialized in CAN reset mode.
The time stamp counter value is stored to TSL and TSH in the CiMBj register (j = 0 to 31) when a received message is stored in a receive mailbox.

### 25.1.25 CANi Test Control Register (CiTCR Register) (i = 0, 1)

CANi Test Control Register (i = 0, 1) (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| C0TCR | 47F58h | 00h |
| C1TCR | 47B58h | 00h |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TSTE | CAN Test Mode Enable Bit | 0: CAN test mode disabled<br>1: CAN test mode enabled | RW |
| TSTM | CAN Test Mode Select Bit | b2 b1<br>0 0 : Other than CAN test mode<br>0 1 : Listen-only mode<br>1 0 : Self-test mode 0 (external loop back)<br>1 1 : Self-test mode 1 (internal loop back) | RW |
| —<br>(b7-b3) | Reserved | Should be written with 0 | RW |

Note:
1. Write to the CiTCR register only in CAN halt mode.

**Figure 25.30 Registers C0TCR and C1TCR**

#### 25.1.25.1 TSTE Bit

When the TSTE bit is set to 0, CAN test mode is disabled.
When this bit is set to 1, CAN test mode is enabled.

#### 25.1.25.2 TSTM Bit

The TSTM bit selects the CAN test mode.
The details of each CAN test mode is described below.

### 25.1.25.3 Listen-Only Mode

The ISO 11898-1 recommends an optional bus monitoring mode. In listen-only mode, the CAN node is able to receive valid data frames and valid remote frames. It sends only recessive bits on the CAN bus and the protocol controller is not required to send the ACK bit, overload flag, or active error flag.
Listen-only mode can be used for baud rate detection.
Do not request transmission from any mailboxes in this mode.
Figure 25.31 shows the connection when listen-only mode is selected.

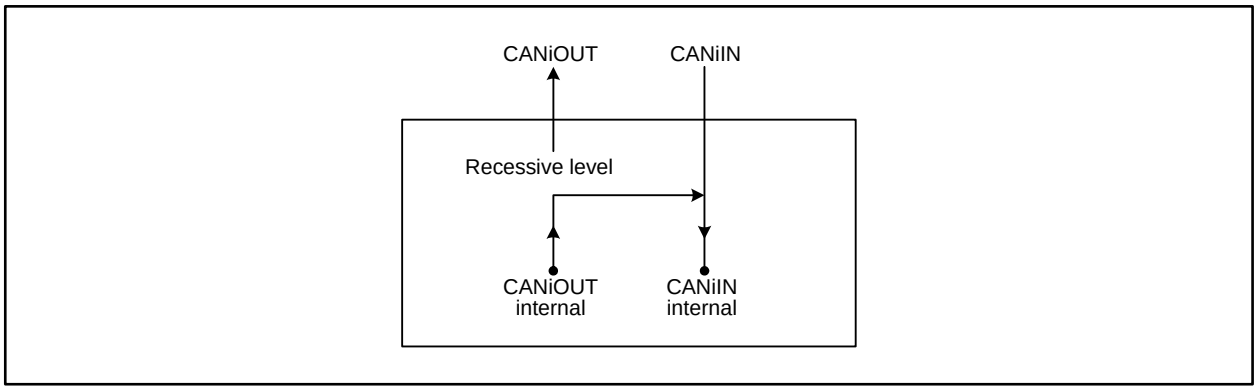



**Figure 25.31 Connection when Listen-Only Mode is Selected (i = 0, 1)**

### 25.1.25.4 Self-Test Mode 0 (External Loop Back)

Self-test mode 0 is provided for CAN transceiver tests.
In this mode, the protocol controller treats its own transmitted messages as messages received via the CAN transceiver and stores them into the receive mailbox. To be independent from external stimulation, the protocol controller generates the ACK bit.
Connect the CANiOUT/CANiIN pins (i = 0, 1) to the transceiver.
Figure 25.32 shows the connection when self-test mode 0 is selected.

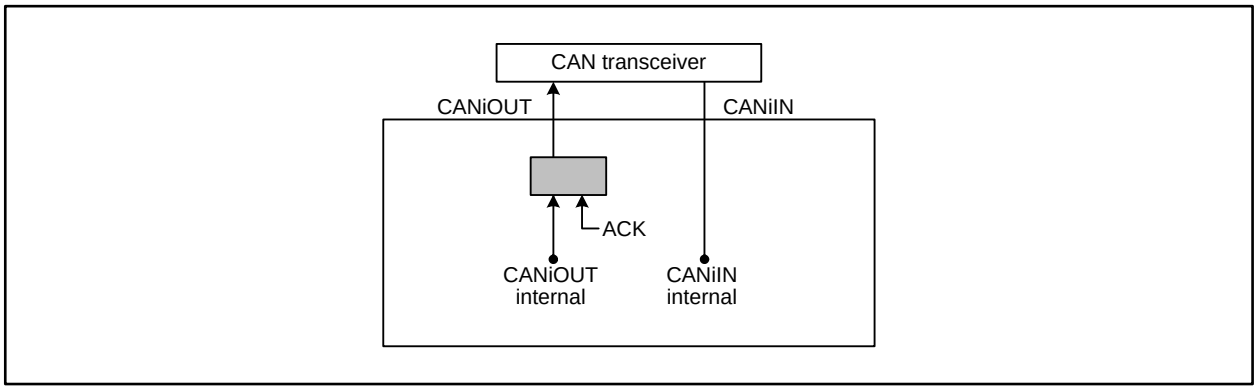



**Figure 25.32 Connection when Self-Test Mode 0 is Selected (i = 0, 1)**

### 25.1.25.5 Self-Test Mode 1 (Internal Loop Back)

Self-test mode 1 is provided for self-test functions.

In this mode, the protocol controller treats its transmitted messages as received messages and stores them into the receive mailbox. To be independent from external stimulation, the protocol controller generates the ACK bit.

In self-test mode 1, the protocol controller performs an internal feedback from the internal CANiOUT pin (i = 0, 1) to the internal CANiIN pin. The input value of the external CANiIN pin is ignored. The external CANiOUT pin outputs only recessive bits. The CANiOUT/CANiIN pins do not need to be connected to the CAN bus or any external device.

Figure 25.33 shows the connection when self-test mode 1 is selected.

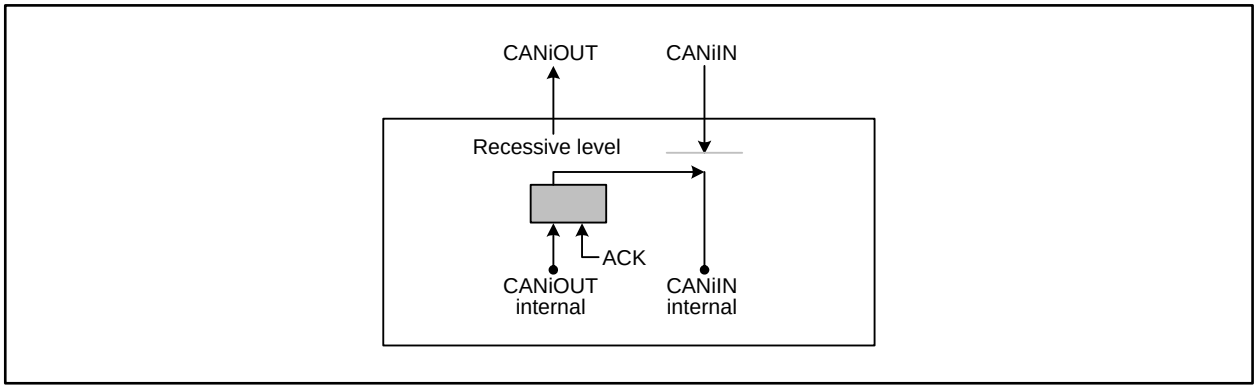


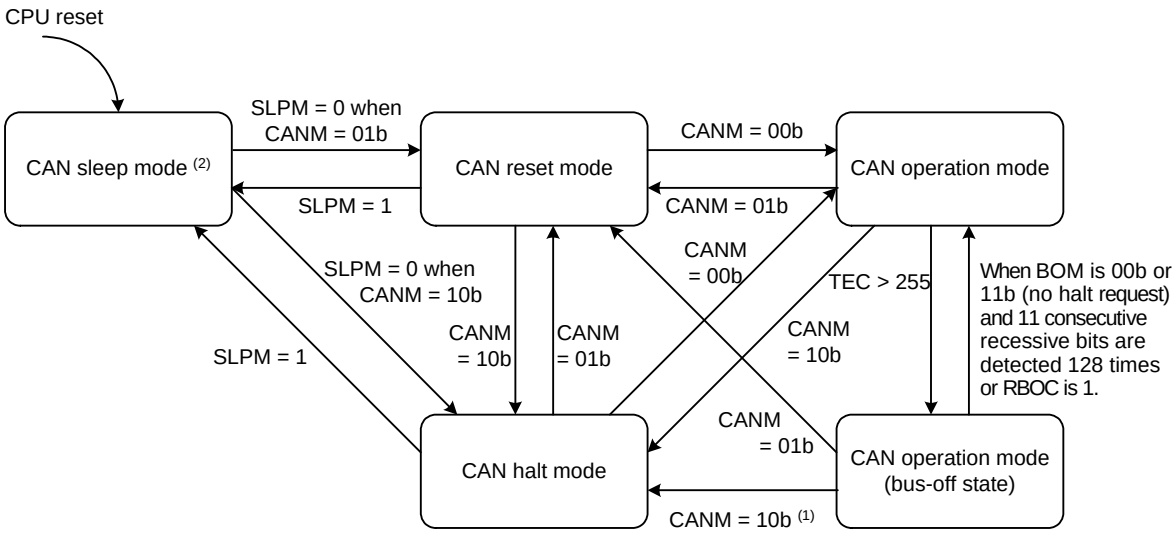

**Figure 25.33 Connection when Self-Test Mode 1 is Selected (i = 0, 1)**

## 25.2 Operating Mode

The CAN module has the following four operating modes:

- CAN reset mode
- CAN halt mode
- CAN operation mode
- CAN sleep mode

Figure 25.34 shows the transition between CAN operating modes.




CANM, SLPM, BOM, RBOC: Bits in the CiCTLR register

Notes:

1. The transition timing from the bus-off state to CAN halt mode depends on the setting of the BOM bit.
   - When the BOM bit is 01b, the state transition timing is immediately after entering the bus-off state.
   - When the BOM bit is 10b, the state transition timing is at the end of the bus-off state.
   - When the BOM bit is 11b, the state transition timing is at the setting of the CANM bit to 10b (CAN halt mode).
2. Write only to the SLPM bit to exit/set CAN sleep mode.

**Figure 25.34 Transition between CAN Operating Modes (i = 0, 1)**

### 25.2.1 CAN Reset Mode

CAN reset mode is provided for CAN communication configuration.
When the CANM bit in the CiCTLR register (i = 0, 1) is set to 01b, the CAN module enters CAN reset. Then the RSTST bit in the CiSTR register is set to 1. Do not change the CANM bit until the RSTST bit is set to 1.
Configure the CiBCR register before exiting CAN reset mode to any other modes.

The following registers are initialized to their reset values after entering CAN reset mode and their initialized values are retained during CAN reset mode:

- CiMCTLj register (j = 0 to 31)
- CiSTR register (except bits SLPST and TFST)
- CiEIFR register
- CiRECR register
- CiTECR register
- CiTSR register
- CiMSSR register
- CiMSMR register
- CiRFCR register
- CiTFCR register
- CiTCR register
- CiECSR register (except EDPM bit)

The previous values of the following registers are retained after entering CAN reset mode.

- CiCLKR register
- CiCTLR register
- CiSTR register (bits SLPST and TFST)
- CiMIER register
- CiEIER register
- CiBCR register
- CiCSSR register
- CiECSR register (EDPM bit only)
- CiMBj register
- Registers CiMKR0 to CiMKR7
- Registers CiFIDCR0 and CiFIDCR1
- CiMKIVLR register
- CiAFSR register
- CiRFPCR register
- CiTFPCR register

### 25.2.2 CAN Halt Mode

CAN halt mode is used for mailbox configuration and test mode setting.
When the CANM bit in the CiCTLR register (i = 0, 1) is set to 10b, CAN halt mode is selected. Then the HLTST bit in the CiSTR register is set to 1. Do not change the CANM bit until the HLTST bit is set to 1. Refer to Table 25.9 Operation in CAN Reset Mode and CAN Halt Mode regarding the state transition conditions when transmitting or receiving.

All registers except bits RSTST, HLTST, and SLPST in the CiSTR register remain unchanged when the CAN module enters CAN halt mode.
Do not change registers CiCLKR, CiCTLR (except bits CANM and SLPM,) and CiEIER in CAN halt mode. The CiBCR register can be changed in CAN halt mode only when listen-only mode is selected to use for automatic bit rate detection.

**Table 25.9 Operation in CAN Reset Mode and CAN Halt Mode**

| Mode | Receiver | Transmitter | Bus-Off |
|---|---|---|---|
| CAN reset mode | CAN module enters CAN reset mode without waiting for the end of message reception. | CAN module enters CAN reset mode after waiting for the end of message transmission. (1, 4) | CAN module enters CAN reset mode without waiting for the end of bus-off recovery. |
| CAN halt mode | CAN module enters CAN halt mode after waiting for the end of message reception. (2, 3) | CAN module enters CAN halt mode after waiting for the end of message transmission. (1, 4) | [When the BOM bit is 00b] A halt request from a program will be acknowledged only after bus-off recovery.<br>[When the BOM bit is 01b] CAN module enters automatically to CAN halt mode without waiting for the end of bus-off recovery (regardless of a halt request from a program).<br>[When the BOM bit is 10b] CAN module enters automatically to CAN halt mode after waiting for the end of bus-off recovery (regardless of a halt request from a program).<br>[When the BOM bit is 11b] CAN module enters CAN halt mode (without waiting for the end of bus-off recovery) if a halt is requested by a program during bus-off. |

BOM bit: Bit in the CiCTLR register (i = 0, 1)
Notes:
1. If several messages are requested to be transmitted, mode transition occurs after the completion of the first transmission. In a case that the CAN reset mode is being requested during suspend transmission, mode transition occurs when the bus is idle, the next transmission ends, or the CAN module becomes a receiver.
2. If the CAN bus is locked at the dominant level, the program can detect this state by monitoring the BLIF bit in the CiEIFR register.
3. If a CAN bus error occurs during reception after CAN halt mode is requested, the CAN mode transits to CAN halt mode.
4. If a CAN bus error or arbitration lost occurs during transmission after CAN reset mode or CAN halt mode is requested, the CAN mode transits to the requested CAN mode.

### 25.2.3 CAN Sleep Mode

CAN sleep mode is used for reducing current consumption by stopping the clock supply to the CAN module. After MCU hardware reset or software reset, the CAN module starts from CAN sleep mode.

When the SLPM bit in the CiCTLR register (i = 0, 1) is set to 1, the CAN module enters CAN sleep mode. Then the SLPST bit in the CiSTR register is set to 1. Do not change the value of the SLPM bit until the bit is set to 1. The other registers remain unchanged when the MCU enters CAN sleep mode.

Write to the SLPM bit in CAN reset mode and CAN halt mode. Do not change any other registers (except the SLPM bit) during CAN sleep mode. Read operation is still allowed.
When the SLPM bit is set to 0, the CAN module is released from CAN sleep mode. When the CAN module exits CAN sleep mode, the other registers remain unchanged.

### 25.2.4 CAN Operation Mode (Excluding Bus-Off State)

CAN operation mode is used for CAN communication.
When the CANM bit in the CiCTLR register (i = 0, 1) is set to 00b, the CAN module enters CAN operation mode.
Then bits RSTST and HLTST in the CiSTR register are set to 0. Do not change the value of the CANM bit until these bits are set to 0.

If 11 consecutive recessive bits are detected after entering CAN operation mode, the CAN module is in the following states:

- The CAN module becomes an active node on the network that enables transmission and reception of CAN messages.
- Error monitoring of the CAN bus, such as receive and transmit error counters, is performed.

During CAN operation mode, the CAN module may be in one of the following three sub-modes, depending on the status of the CAN bus:

- Idle mode: Transmission or reception is not being performed.
- Receive mode: A CAN message sent by another node is being received.
- Transmit mode: A CAN message is being transmitted. The CAN module may receive its own message simultaneously when self-test mode 0 (TSTM bit in the CiSTR register = 10b) or self-test mode 1 (TSTM bit = 11b) is selected.

Figure 25.35 shows the sub mode in CAN operation mode.

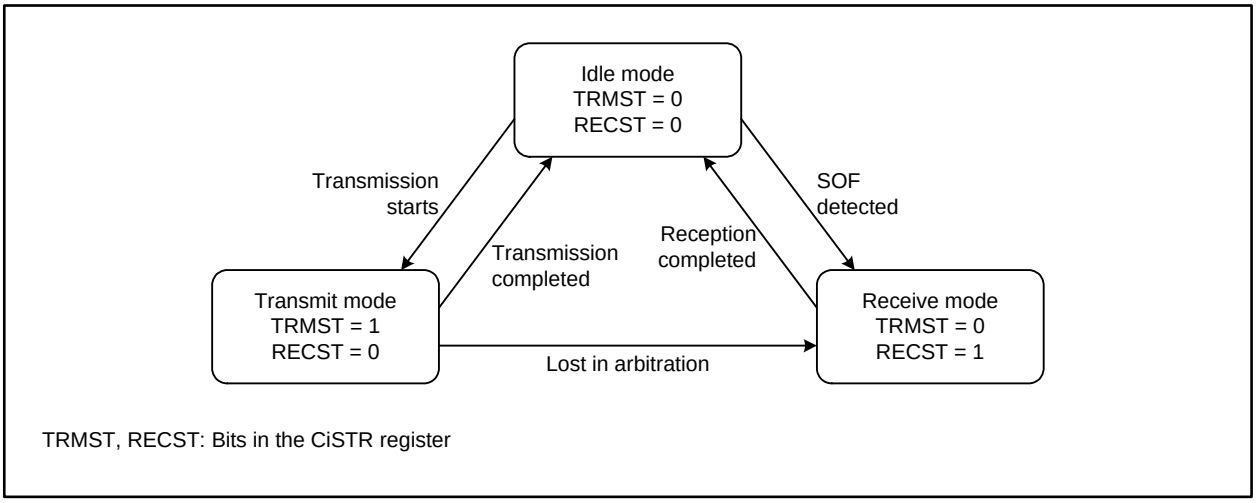



**Figure 25.35 Sub Mode in CAN Operation Mode (i = 0, 1)**

### 25.2.5 CAN Operation Mode (Bus-Off State)

The CAN module enters the bus-off state according to the increment/decrement rules for the transmit/error counters in the CAN Specifications.
The following cases apply when recovering from the bus-off state. When the CAN module is in bus-off state, the values of the associated registers, except registers CiSTR, CiEIFR, CiRECR, CiTECR and CiTSR (i = 0, 1), remain unchanged.

(1) When the BOM bit in the CiCTLR register is 00b (normal mode)

The CAN module enters the error-active state after it has completed the recovery from the bus-off state and CAN communication is enabled. The BORIF bit in the CiEIFR register is set to 1 (bus-off recovery detected) at this time.

(2) When the RBOC bit in the CiCTLR register is set to 1 (forcible return from bus-off)

The CAN module enters the error-active state when it is in bus-off state and the RBOC bit is set to 1. CAN communication is enabled again after 11 consecutive recessive bits are detected. The BORIF bit is not set to 1 at this time.

(3) When the BOM bit is 01b (entry to CAN halt mode automatically at bus-off entry)

The CAN module enters CAN halt mode when it reaches the bus-off state. The BORIF bit is not set to 1 at this time.

(4) When the BOM bit is 10b (entry to CAN halt mode automatically at bus-off end)

The CAN module enters CAN halt mode when it has completed the recovery from bus-off. The BORIF bit is set to 1 at this time.

(5) When the BOM bit is 11b (entry to CAN halt mode by a program) and the CANM bit in the CiCTLR register is set to 10b (CAN halt mode) during the bus-off state

The CAN module enters CAN halt mode when it is in bus-off state and the CANM bit is set to 10b (CAN halt mode). The BORIF bit is not set to 1 at this time.
If the CANM bit is not set to 10b during bus-off, the same behavior as (1) applies.

## 25.3 CAN Communication Speed Configuration

The following description explains about the CAN communication speed configuration.

### 25.3.1 CAN Clock Configuration

This group has a CAN clock selector.

The CAN clock can be configured by setting the CCLKS bit in the CiCLKR register (i = 0, 1) and the BRP bit in the CiBCR register.

Figure 25.36 shows the block diagram of CAN clock generator.

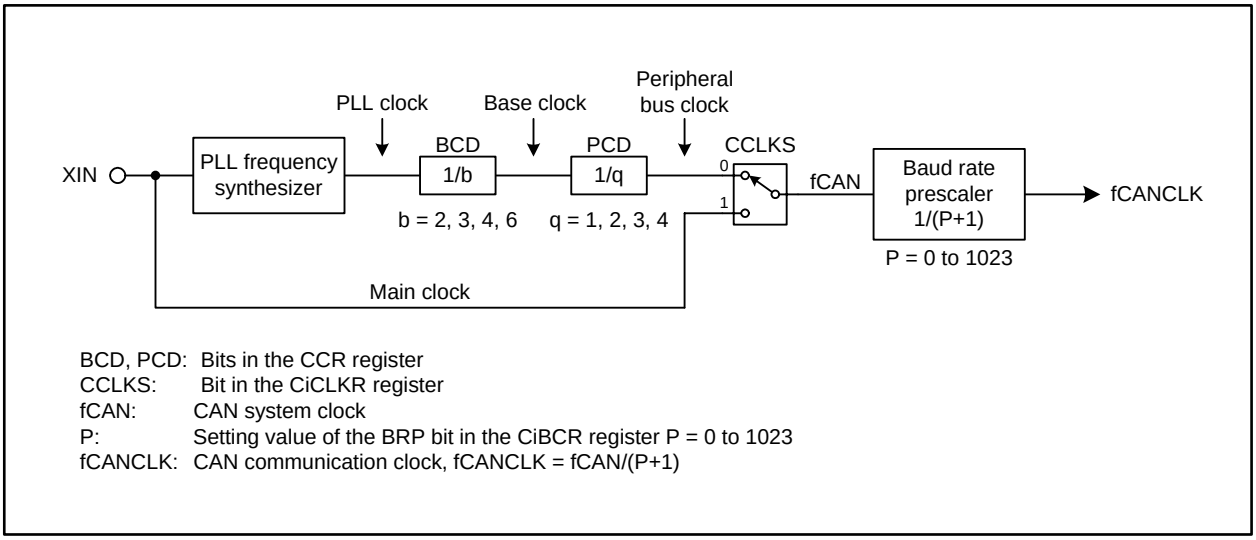



**Figure 25.36 Block Diagram of CAN Clock Generator (i = 0, 1)**

### 25.3.2 Bit Timing Configuration

The bit time is a single bit time for transmitting/receiving a message and consists of the following three segments.

Figure 25.37 shows the bit timing.

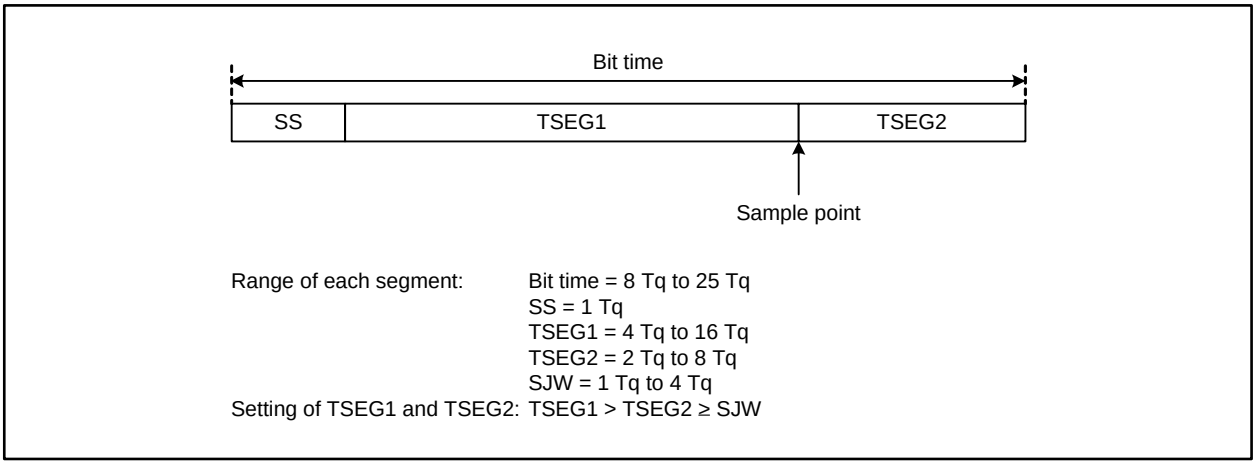



**Figure 25.37 Bit Timing**

### 25.3.3 Bit rate

The bit rate depends on the CAN clock (fCAN), the division value of the baud rate prescaler, and the number of Tq of one bit time.

$$\text{Bit rate[bps]} = \frac{fCAN}{\text{Baud rate prescaler division value}\ (1) \times \text{number of Tq of one bit time}} = \frac{fCANCLK}{\text{Number of Tq of one bit time}}$$

Note:
1. Division value of the baud rate prescaler = P + 1 (P = 0 to 1023)
   P: Setting value of the BRP bit in the CiBCR register (i = 0, 1)

Table 25.10 lists bit rate examples.

**Table 25.10 Bit Rate Examples**

| fCAN | 32 MHz | | 24 MHz | | 20 MHz | | 16 MHz | | 8 MHz | |
|---|---|---|---|---|---|---|---|---|---|---|
| Bit Rate | No. of Tq | P+1 | No. of Tq | P+1 | No. of Tq | P+1 | No. of Tq | P+1 | No. of Tq | P+1 |
| 1 Mbps | 8 Tq | 4 | 8 Tq | 3 | 10 Tq | 2 | 8 Tq | 2 | 8 Tq | 1 |
| | 16 Tq | 2 | | | 20 Tq | 1 | 16 Tq | 1 | | |
| 500 kbps | 8 Tq | 8 | 8 Tq | 6 | 10 Tq | 4 | 8 Tq | 4 | 8 Tq | 2 |
| | 16 Tq | 4 | 16 Tq | 3 | 20 Tq | 2 | 16 Tq | 2 | 16 Tq | 1 |
| 250 kbps | 8 Tq | 16 | 8 Tq | 12 | 10 Tq | 8 | 8 Tq | 8 | 8 Tq | 4 |
| | 16 Tq | 8 | 16 Tq | 6 | 20 Tq | 4 | 16 Tq | 4 | 16 Tq | 2 |
| 83.3 kbps | 8 Tq | 48 | 8 Tq | 36 | 8 Tq | 30 | 8 Tq | 24 | 8 Tq | 12 |
| | 16 Tq | 24 | 16 Tq | 18 | 10 Tq | 24 | 16 Tq | 12 | 16 Tq | 6 |
| | | | | | 16 Tq | 15 | | | | |
| | | | | | 20 Tq | 12 | | | | |
| 33.3 kbps | 8 Tq | 120 | 8 Tq | 90 | 8 Tq | 75 | 8 Tq | 60 | 8 Tq | 30 |
| | 10 Tq | 96 | 10 Tq | 72 | 10 Tq | 60 | 10 Tq | 48 | 10 Tq | 24 |
| | 16 Tq | 60 | 16 Tq | 45 | 20 Tq | 30 | 16 Tq | 30 | 16 Tq | 15 |
| | 20 Tq | 48 | 20 Tq | 36 | | | 20 Tq | 24 | 20 Tq | 12 |

## 25.4 Mailbox and Mask Register Structure

There are 32 mailboxes with the same structure.
Figure 25.38 shows the structure of registers C0MBj to C1MBj (j = 0 to 31).

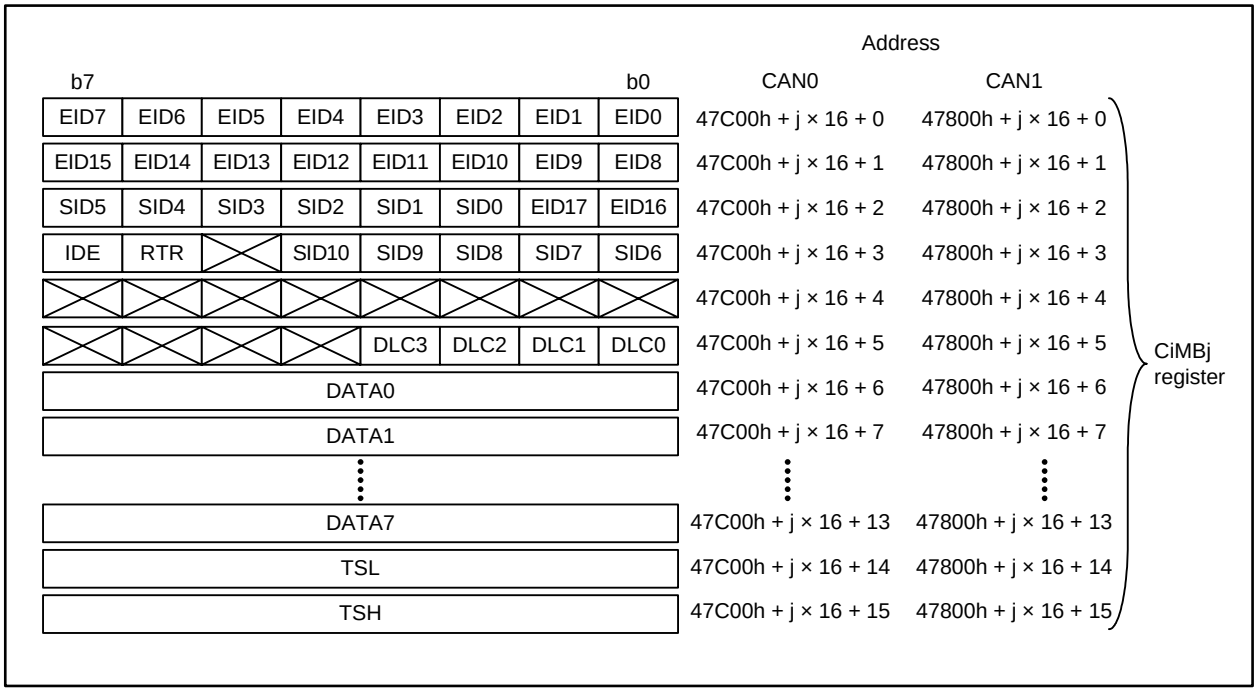



**Figure 25.38 Structure of Registers C0MBj to C1MBj (i = 0, 1; j = 0 to 31)**

There are 8 mask registers with the same structure.
Figure 25.39 shows the structure of registers C0MKRk to C1MKRk (k = 0 to 7).

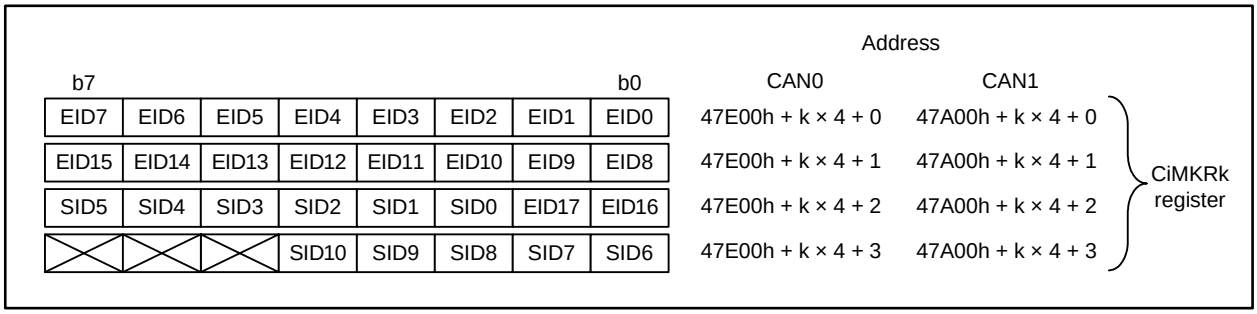



**Figure 25.39 Structure of Registers C0MKRk to C1MKRk (i = 0, 1; k = 0 to 7)**

There are 2 FIFO received ID compare registers with the same structure.
Figure 25.40 shows the structure of registers C0FIDCRn to C1FIDCRn (n = 0, 1).

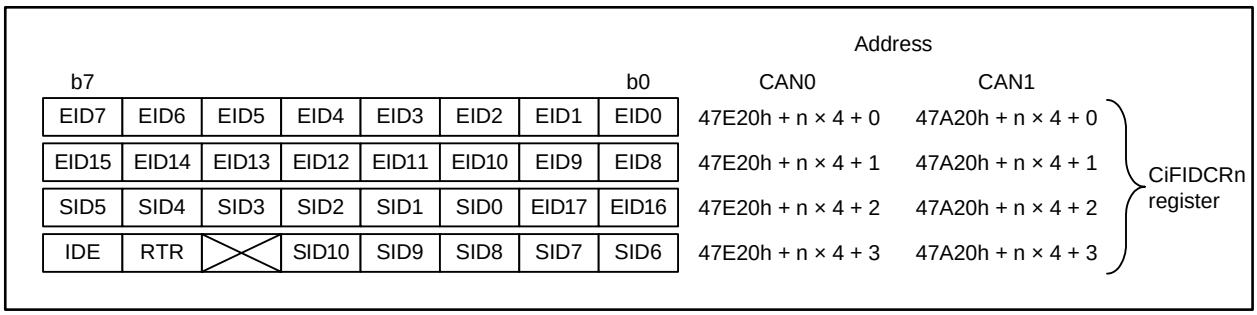



**Figure 25.40 Structure of Registers C0FIDCRn to C1FIDCRn (i = 0, 1; n = 0, 1)**

## 25.5 Acceptance Filtering and Masking Function

Acceptance filtering allows the user to receive messages with a specified range of multiple IDs for mailboxes.
Registers CiMKR0 to CiMKR7 (i = 0, 1) can perform masking of the standard ID and the extended ID of 29 bits.

- The CiMKR0 register corresponds to mailboxes [0] to [3].
- The CiMKR1 register corresponds to mailboxes [4] to [7].
- The CiMKR2 register corresponds to mailboxes [8] to [11].
- The CiMKR3 register corresponds to mailboxes [12] to [15].
- The CiMKR4 register corresponds to mailboxes [16] to [19].
- The CiMKR5 register corresponds to mailboxes [20] to [23].
- The CiMKR6 register corresponds to mailboxes [24] to [27] in normal mailbox mode, and receive FIFO mailboxes [28] to [31] in FIFO mailbox mode.
- The CiMKR7 register corresponds to mailboxes [28] to [31] in normal mailbox mode, and receive FIFO mailboxes [28] to [31] in FIFO mailbox mode.

The CiMKIVLR register disables acceptance filtering individually for each mailbox.

The IDE bit in the CiMBj register (j = 0 to 31) is enabled when the IDFM bit in the CiCTLR register is 10b (mixed ID mode).
The RTR bit in the CiMBj register selects a data frame or a remote frame.

In FIFO mailbox mode, normal mailboxes (mailboxes [0] to [23]) use the single corresponding register among registers CiMKR0 to CiMKR5 for acceptance filtering. Receive FIFO mailboxes (mailboxes [28] to [31]) use two registers CiMKR6 and CiMKR7 for the acceptance filtering.
Also, the receive FIFO uses two registers CiFIDCR0 and CiFIDCR1 for ID comparison. Bits EID, SID, RTR, and IDE in registers CiMB28 to CiMB31 for the receive FIFO are disabled. As acceptance filtering depends on the result of two ID-mask sets, two ranges of IDs can be received into the receive FIFO.
The CiMKIVLR register is disabled for the receive FIFO.

If both setting of standard ID and extended ID are set in the IDE bits in registers CiFIDCR0 and CiFIDCR1 individually, both ID formats are received.
If both setting of data frame and remote frame are set in the RTR bits in registers CiFIDCR0 and CiFIDCR1 individually, both data and remote frames are received.
When combination with two ranges of IDs is not necessary, set the same mask value and the same ID into both of the FIFO ID/mask register sets.

Figure 25.41 shows the correspondence of mask registers to mailboxes, and Figure 25.42 shows acceptance filtering.

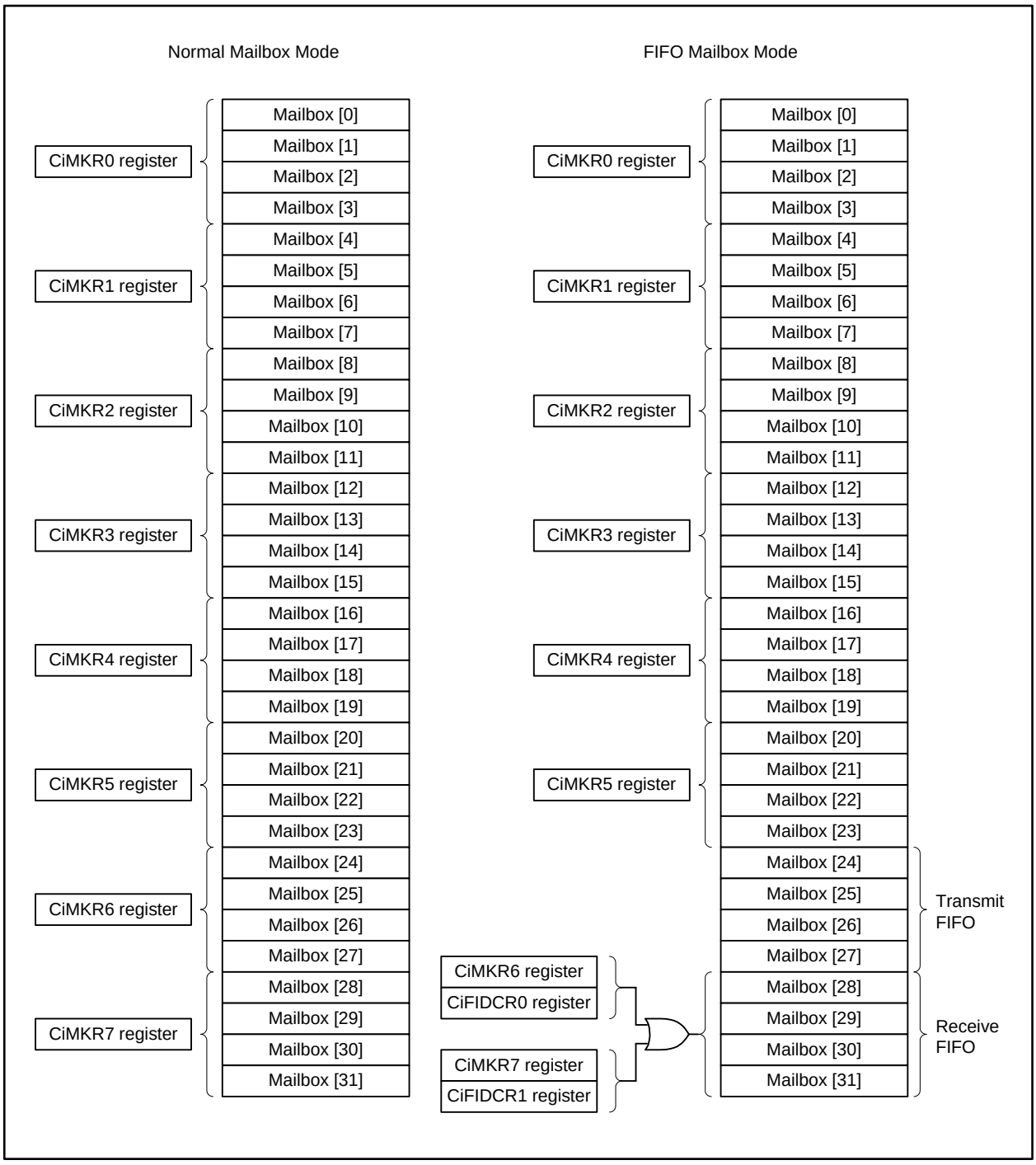
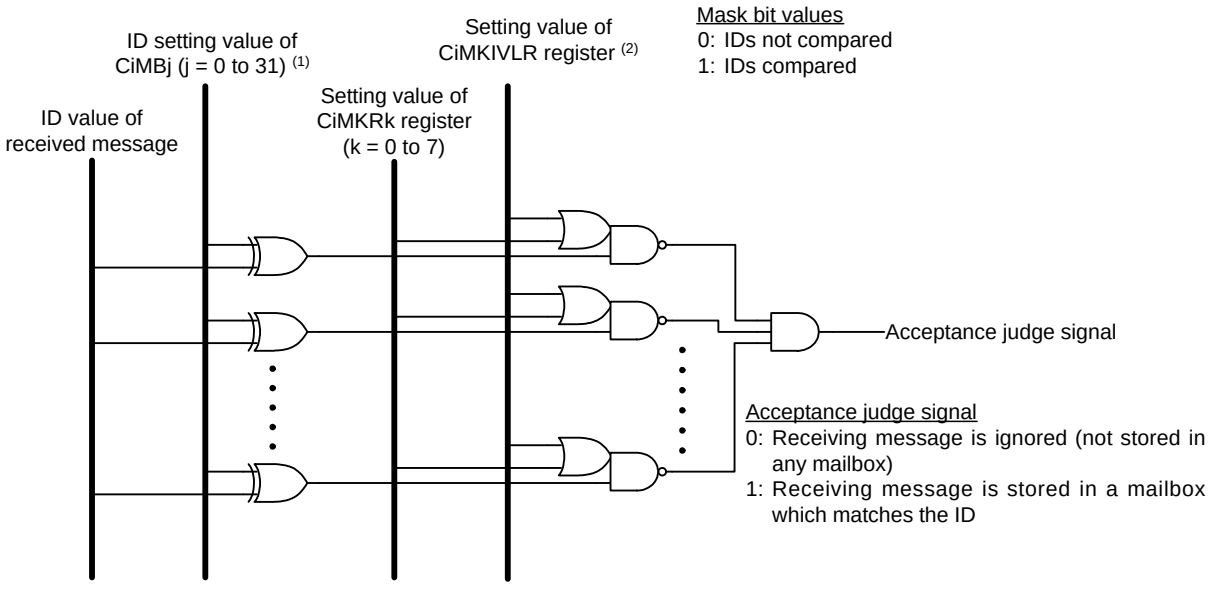



**Figure 25.41 Correspondence of Mask Registers to Mailboxes (i = 0, 1)**




Notes:
1. The values set in registers CiFIDCR0 and CiFIDCR1 are used in FIFO mailbox mode.
2. Invalid in FIFO mailboxes.

**Figure 25.42 Acceptance Filtering (i = 0, 1)**

## 25.6 Reception and Transmission

Table 25.11 list the CAN communication mode configuration.

**Table 25.11 Configuration for CAN Reception Mode and Transmission Mode**

| TRMREQ | RECREQ | ONESHOT | Communication Mode of Mailbox |
|---|---|---|---|
| 0 | 0 | 0 | Mailbox disabled or transmission being aborted. |
| 0 | 0 | 1 | Configurable only when transmission or reception from a mailbox (programmed in one-shot mode) is aborted. |
| 0 | 1 | 0 | Configured as a receive mailbox for a data frame or a remote frame. |
| 0 | 1 | 1 | Configured as a one-shot receive mailbox for a data frame or a remote frame. |
| 1 | 0 | 0 | Configured as a transmit mailbox for a data frame or a remote frame. |
| 1 | 0 | 1 | Configured as a one-shot transmit mailbox for a data frame or a remote frame. |
| 1 | 1 | 0 | Do not set. |
| 1 | 1 | 1 | Do not set. |

TRMREQ, RECREQ, ONESHOT: Bits in the CiMCTLj register (i = 0, 1; j = 0 to 31)

When a mailbox is configured as a receive mailbox or a one-shot receive mailbox, note the following:

(1) Before a mailbox is configured as a receive mailbox or a one-shot receive mailbox, set the CiMCTLj register (i = 0, 1; j = 0 to 31) to 00h.

(2) A received message is stored into the first mailbox that matches the condition according to the result of receive mode configuration and acceptance filtering. Upon deciding a mailbox which stores the received message, the mailbox with the smaller number has higher priority.

(3) In CAN operation mode, when a CAN module transmits a message whose ID matches with the ID/mask set of a mailbox configured to receive messages, the CAN module never receives the transmitted data. In self-test mode, however, the CAN module may receive its transmitted data. In this case, the CAN module sends an ACK.

When configuring a mailbox as a transmit mailbox or a one-shot transmit mailbox, note the following:

(1) Before a mailbox is configured as a transmit mailbox or one-shot transmit mailbox, ensure that the CiMCTLj register is 00h and that there is no pending abort process.

### 25.6.1 Reception

Figure 25.43 shows an operation example of data frame reception in overwrite mode.
This example shows the operation of overwriting the first message when the CAN module receives two consecutive CAN messages that matches the receiving conditions of the CiMCTL0 register (i = 0, 1).

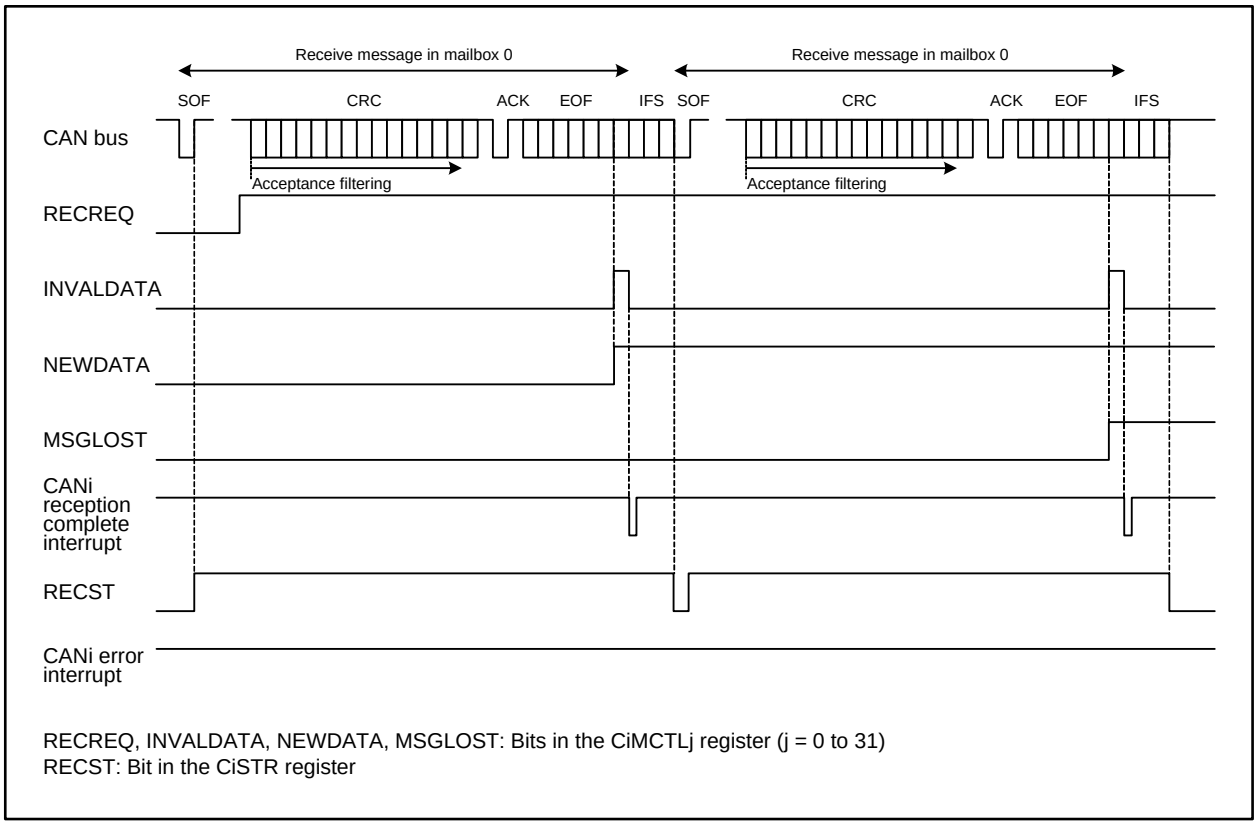



**Figure 25.43 Operation Example of Data Frame Reception in Overwrite Mode** (i = 0, 1)

(1) When a SOF is detected on the CAN bus, the RECST bit in the CiSTR register is set to 1 (reception in progress) if the CAN module has no message ready to start transmission.
(2) The acceptance filter procedure starts at the beginning of the CRC field to select the receive mailbox.
(3) After a message has been received, the NEWDATA bit in the CiMCTLj register (j = 0 to 31) for the receive mailbox is set to 1 (new data being updated/stored in the mailbox). The INVALDATA bit in the CiMCTLj register is set to 1 (message is being updated) at the same time, and then the INVALDATA bit is set to 0 (message valid) again after the complete message is transferred to the mailbox.
(4) When the interrupt enable bit in the CiMIER register for the receive mailbox is 1 (interrupt enabled), the CANi reception complete interrupt request is generated. This interrupt is generated when the INVALDATA bit is set to 0.
(5) After reading the message from the mailbox, the NEWDATA bit needs to be set to 0 by a program.
(6) In overwrite mode, if the next CAN message has been received into a mailbox whose NEWDATA bit is still set to 1, the MSGLOST bit in the CiMCTLj register is set to 1 (message has been overwritten). The new received message is transferred to the mailbox. The CANi reception complete interrupt request is generated the same as in (4).

Figure 25.44 shows the operational example of data frame reception in overrun mode.
This example shows the operation of overrunning the second message when the CAN module receives two consecutive CAN messages that matches the receiving conditions of the CiMCTL0 register (i = 0, 1).

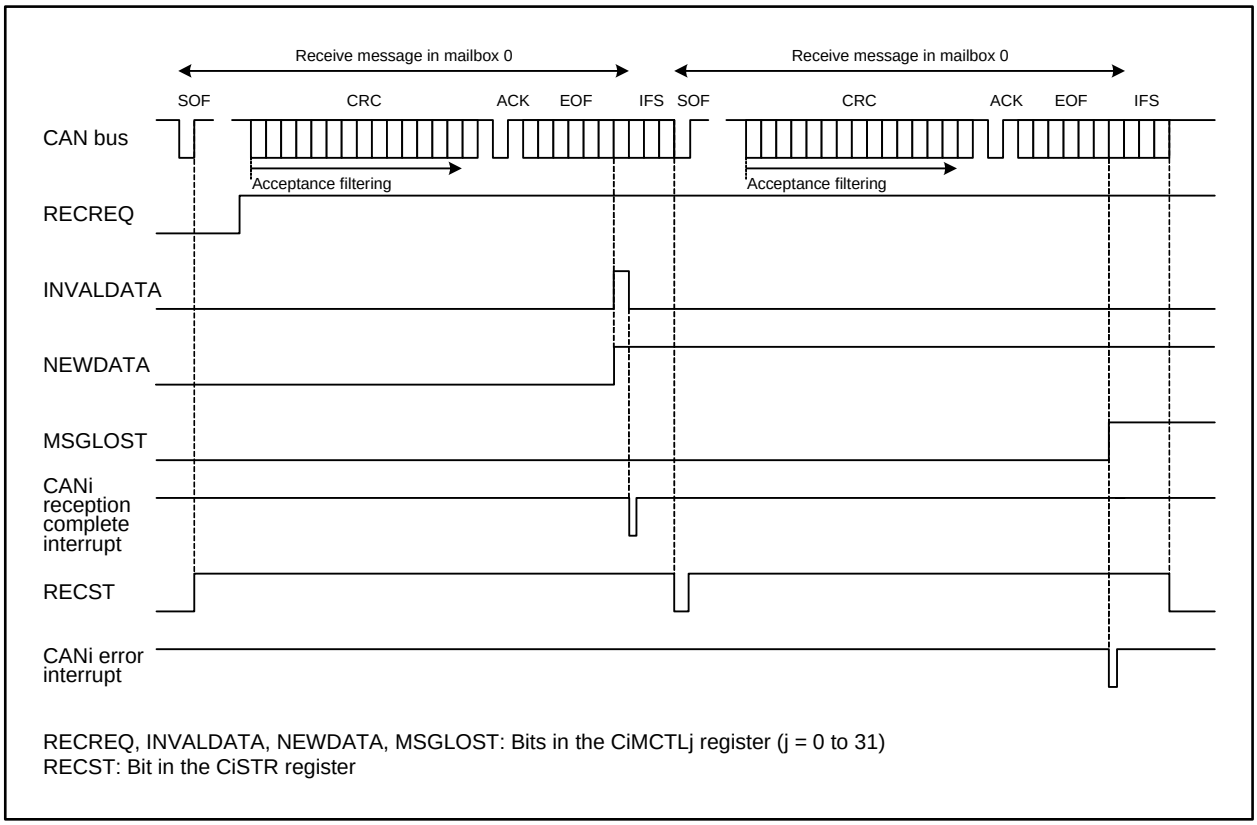



**Figure 25.44 Operation Example of Data Frame Reception in Overrun Mode** (i = 0, 1)

(1) to (5) are the same as overwrite mode.

(6) In overrun mode, if the next message has been received before the NEWDATA bit is set to 0, the MSGLOST bit in the CiMCTLj register (j = 0 to 31) is set to 1 (message has been overrun). The new received message is discarded and a CANi error interrupt request is generated if the corresponding interrupt enable bit in the CiEIER register is set to 1 (interrupt enabled).

## 25.6.2 Transmission

Figure 25.45 shows an operation example of data frame transmission. This example shows the operation of transmitting messages that has been set in registers CiMCTL0 and CiMCTL1 (i = 0, 1).

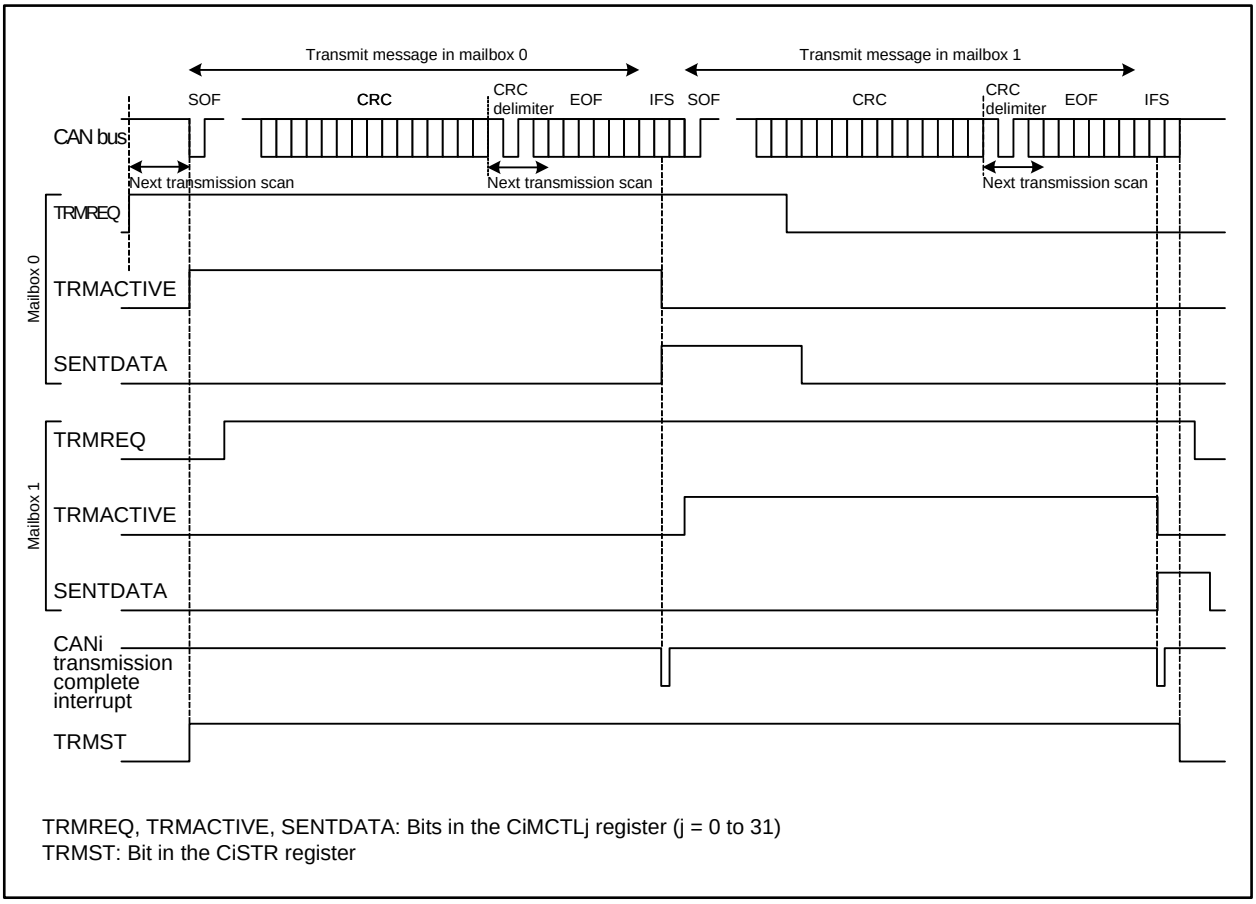



**Figure 25.45 Operation Example of Data Frame Transmission** (i = 0, 1)

(1) When a TRMREQ bit in the CiMCTLj register (j = 0 to 31) is set to 1 (transmit mailbox) in bus-idle state, the mailbox scan procedure starts to decide the highest-priority mailbox for transmission. Once the transmit mailbox is decided, the TRMACTIVE bit in the CiMCTLj register is set to 1 (from when a transmission request is received until transmission is completed, or an error/arbitration lost has occurred), the TRMST bit in the CiSTR register is set to 1 (transmission in progress), and the CAN module starts transmission. (1)

(2) If other TRMREQ bits are set, the transmission scan procedure starts with the CRC delimiter for the next transmission.

(3) If transmission is completed without losing arbitration, the SENTDATA bit in the CiMCTLj register is set to 1 (transmission completed) and the TRMACTIVE bit is set to 0 (transmission is pending, or no transmission request). If the interrupt enable bit in the CiMIER register is 1 (interrupt enabled), the CANi transmission complete interrupt request is generated.

(4) When requesting the next transmission from the same mailbox, set bits SENDTDATA and TRMREQ to 0, then set the TRMREQ bit to 1 after checking that bits SENDTDATA and TRMREQ have been set to 0.

Note:

1. If arbitration is lost after the CAN module starts transmission, the TRMACTIVE bit is set to 0. The transmission scan procedure is performed again to search for the highest-priority transmit mailbox from the beginning of the CRC delimiter. If an error occurs either during transmission or following the loss of arbitration, the transmission scan procedure is performed again from the start of the error delimiter to search for the highest-priority transmit mailbox.

## 25.7 CAN Interrupt

The CAN module provides the following CAN interrupts:

- CANi wakeup interrupt (i = 0, 1)
- CANi reception complete interrupt
- CANi transmission complete interrupt
- CANi receive FIFO interrupt
- CANi transmit FIFO interrupt
- CANi error interrupt

There are eight types of interrupt sources for the CANi error interrupts. These sources can be determined by checking the CiEIFR register.

- Bus error
- Error-warning
- Error-passive
- Bus-off entry
- Bus-off recovery
- Receive overrun
- Overload frame transmission
- Bus lock

# 26. I/O Pins

Each pin of the MCU functions as a programmable I/O port, an I/O pin for internal peripheral functions, or a bus control pin. These functions can be switched by the function select registers or the processor mode registers. This chapter particularly addresses the function select registers. For the use as the bus control pin, refer to 7. “Processor Mode” and 9. “Bus”.
The pull-up resistors are enabled for every group of four pins. However, a pull-up resistor is separated from other peripheral functions even if it is enabled, when a pin functions as output pin or an analog I/O pin.
Figure 26.1 shows a block diagram of typical I/O pin.

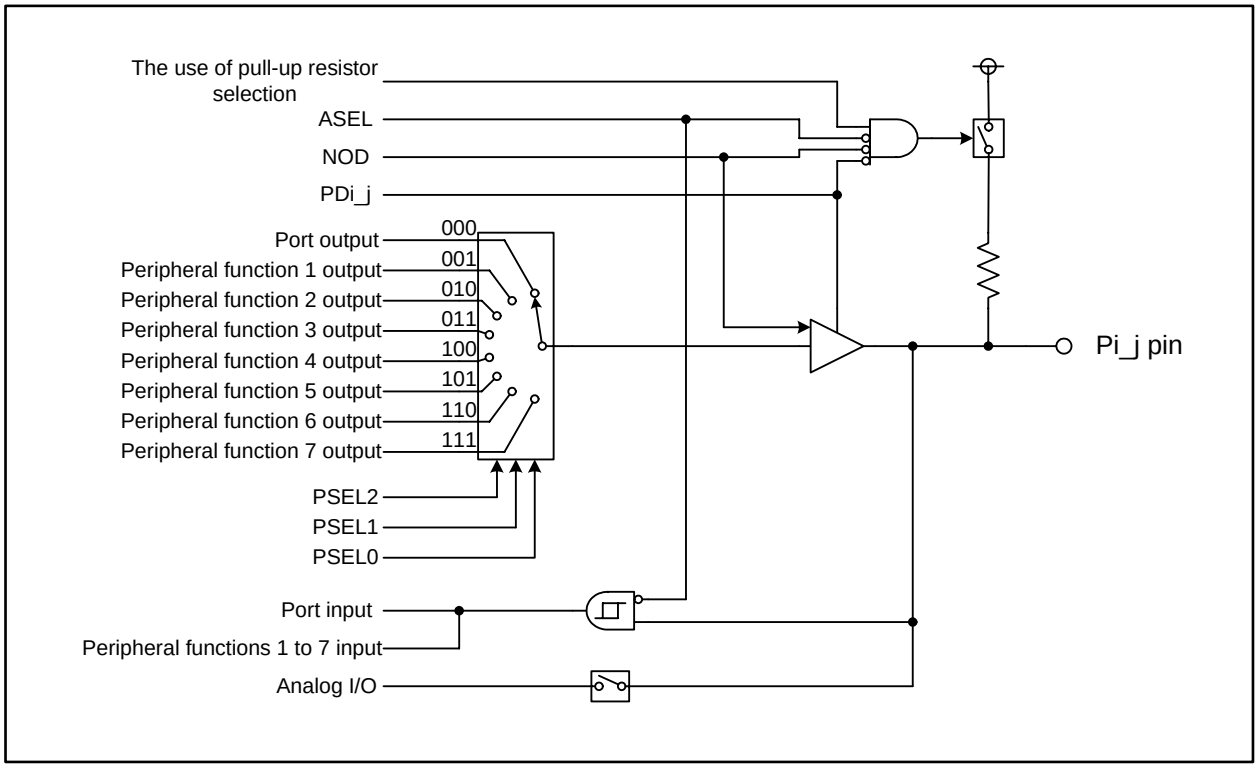



**Figure 26.1 Typical I/O Pin Block Diagram (i = 0 to 15; j = 0 to 7)**

The registers to control I/O pins are as follows: port Pi direction register (PDi register), output function select register and pull-up control register. The PDi register selects input or output state of pins. The output function select register which selects an output function consists of bits PSEL2 to PSEL0, NOD, and ASEL. Bits PSEL2 to PSEL0 are to select a function as programmable I/O or peripheral function output (except analog output). The NOD bit is to select the N-channel open drain output for a pin. The ASEL bit enables to prevent the increase in power consumption of input buffer due to an intermediate potential when a pin functions as an analog I/O pin. The pull-up control register enables/disables the pull-up resistors.
To use a pin as analog I/O pin, the PDi_j bit should be set to 0 (input) and bits PSEL2 to PSEL0 should be set to 000b and the ASEL bit should be set to 1.
The input-only port P8_5, which shares a pin with the $\overline{\text{NMI}}$ has neither the bit 5 of the function select register nor the PDi register. The port P14_1 (or P9_1 in the 100-pin package) also functions as input-only port. The bit 1 of the function select register and the PDi register is assigned for reserved bit. The port P9 is protected from unexpected write accesses by the PRC2 bit in the PRCR register. (Refer to 10. “Protection”)

## 26.1 Port Pi Direction Register (PDi Register, i = 0 to 15)

The PDi register selects input or output state of pins. Each bit in this register corresponds to a respective pin.

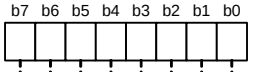
In memory expansion mode or microprocessor mode, this register cannot control pins being used as the bus control pins (A0 to A23, D0 to D31, $\overline{CS0}$ to $\overline{CS3}$, $\overline{WR}/\overline{WR0}$, $\overline{BC0}$, $\overline{BC1}/\overline{WR1}$, $\overline{BC2}/\overline{WR2}$, $\overline{BC3}/\overline{WR3}$, $\overline{RD}$, CLKOUT/BCLK, $\overline{HLDA}$, $\overline{HOLD}$, ALE, and $\overline{RDY}$).

Figure 26.2 shows the PDi register.

No register bit is provided for the P8_5. For the P14_1 (or P9_1 in the 100-pin package), a reserved bit is provided.

The PD9 register is protected from unexpected write accesses by setting the PRC2 bit in the PRCR register. (Refer to 10. “Protection”)

Port Pi Direction Register (i = 0 to 15) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| PD0 to PD3 | 03C2h, 03C3h, 03C6h, 03C7h | 0000 0000b |
| PD4 to PD7 | 03CAh, 03CBh, 03CEh, 03CFh | 0000 0000b |
| PD8 (2) | 03D2h | 00X0 0000b |
| PD9 (3, 4), PD10 | 03D3h, 03D6h | 0000 0000b |
| PD11 (2, 5) | 03D7h | XXX0 0000b |
| PD12, PD13 (5) | 03DAh, 03DBh | 0000 0000b |
| PD14 (2, 4, 5) | 03DEh | X000 0000b |
| PD15 (5) | 03DFh | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PDi_0 | Port Pi_0 Direction Bit (4) | 0: Input port<br>1: Output port | RW |
| PDi_1 | Port Pi_1 Direction Bit (4) | 0: Input port<br>1: Output port | RW |
| PDi_2 | Port Pi_2 Direction Bit (4) | 0: Input port<br>1: Output port | RW |
| PDi_3 | Port Pi_3 Direction Bit | 0: Input port<br>1: Output port | RW |
| PDi_4 | Port Pi_4 Direction Bit | 0: Input port<br>1: Output port | RW |
| PDi_5 | Port Pi_5 Direction Bit (2) | 0: Input port<br>1: Output port | RW |
| PDi_6 | Port Pi_6 Direction Bit (2) | 0: Input port<br>1: Output port | RW |
| PDi_7 | Port Pi_7 Direction Bit (2) | 0: Input port<br>1: Output port | RW |

Notes:
1. In memory expansion mode or microprocessor mode, the PDi register cannot control pins being used as bus control pins (A0 to A23, D0 to D31, $\overline{CS0}$ to $\overline{CS3}$, $\overline{WR}/\overline{WR0}$, $\overline{BC0}$, $\overline{BC1}/\overline{WR1}$, $\overline{BC2}/\overline{WR2}$, $\overline{BC3}$ $\overline{WR3}$, $\overline{RD}$, CLKOUT/BCLK, $\overline{HLDA}$, $\overline{HOLD}$, ALE, and $\overline{RDY}$).
2. The PD8_5 bit in the PD8 register, bits PD11_5 to PD11_7 in the PD11 register, the PD14_7 bit in the PD14 register are unavailable on this MCU. These bits should be written with 0 and read as undefined value.
3. The PD9 register should be written immediately after the instruction to set the PRC2 bit in the PRC2 register to 1 (write enabled). Any interrupt or DMA transfer should not be generated between these two instructions.
4. Bits PD9_0 to PD9_2 in the PD9 register in the 100-pin package and PD14_0 to PD14_2 in the PD14 register in the 144-pin package are reserved. These bits should be written with 0.
5. In the 100-pin package, enabled bits in registers PD11 to PD15 should be written with 1 (output port).

**Figure 26.2 Registers PD0 to PD15**

## 26.2 Output Function Select Register

This register selects an output function of either the programmable I/O port or a peripheral function if these two functions share a pin. Regarding input function, every connected peripheral functions obtain input signals irrespective of this register setting.

The output function select register consists of bits PSEL2 to PSEL0, NOD, and ASEL. Bits PSEL2 to PSEL0 select a function as programmable I/O or peripheral function output (except analog output). The NOD bit is to select the N-channel open drain output. The ASEL bit enables to prevent the increase in power consumption due to an intermediate potential generated when a pin functions as an analog I/O pin.

Table 26.1 shows the peripheral functions assigned for each combination of bits PSEL2 to PSEL0 and Figure 26.3 to Figure 26.19 show the function select registers.

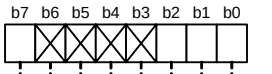
Note that ports P8_5 and P14_1 (or P9_1 in the 100-pin package) (input only) have no output function select registers.

The P9_iS register is protected from unexpected write accesses by setting the PRC2 bit in the PRCR register (Refer to 10. “Protection”)

**Table 26.1 Peripheral Function Assignment**

| Bits PSEL2 to PSEL0 | Peripheral Functions |
|---|---|
| 001b | Timer |
| 010b | Three-phase motor control timers |
| 011b | UART |
| 100b | UART special function |
| 101b | Intelligent I/O groups 0 and 2, CAN channel 0 |
| 110b | Intelligent I/O group 1, CAN channel 1 |
| 111b | UART8 |

Port P0_i Function Select Register (i = 0 to 7)



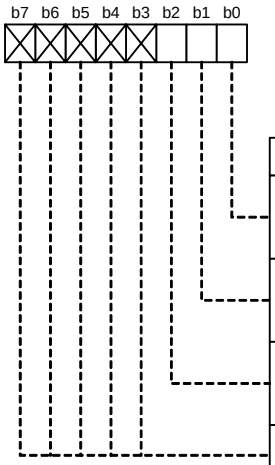

| Symbol | Address | Reset Value |
|---|---|---|
| P0_0S to P0_2S | 400A0h, 400A2h, 400A4h | 0XXX X000b |
| P0_3S to P0_5S | 400A6h, 400A8h, 400AAh | 0XXX X000b |
| P0_6S, P0_7S | 400ACh, 400AEh | 0XXX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P0_i Output Function Select Bit | b2 b1 b0<br>0 0 0 : I/O port P0_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : Do not use this combination<br>1 0 0 : Do not use this combination<br>1 0 1 : Do not use this combination<br>1 1 0 : Do not use this combination<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b6-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| ASEL | Port P0_i Analog Function Select Bit | 0: Function other than AN0_i<br>1: AN0_i | RW |

**Figure 26.3 Registers P0_0S to P0_7S**

The port P0_i (i = 0 to 7) shares a pin with the AN0_i input pin for the A/D converter.
To use as the programmable I/O port, the P0_iS register should be set to 00h. To use as the A/D converter input pin, this register should be set to 80h and the PD0_i bit should be set to 0 (Port P0_i functions as input port).

## Port P1_i Function Select Register (i = 0 to 7)




| Symbol | Address | Reset Value |
|---|---|---|
| P1_0S to P1_2S | 400A1h, 400A3h, 400A5h | XXXX X000b |
| P1_3S to P1_5S | 400A7h, 400A9h, 400ABh | XXXX X000b |
| P1_6S, P1_7S | 400ADh, 400AFh | XXXX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P1_i Output Function Select Bit (1) | b2 b1 b0<br>0 0 0 : I/O port P1_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : Do not use this combination<br>1 0 0 : Do not use this combination<br>1 0 1 : IIO0 output<br>1 1 0 : IIO1 output<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b7-b3) | No register bits; should be written with 0 and read as undefined value | | — |

Notes:
1. Refer to the table below for each pin setting.

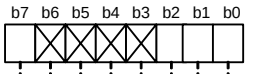
| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P1_0 | P1_0 | — (2) | — (2) | — (2) | — (2) | IIO0_0 output | IIO1_0 output | — (2) |
| P1_1 | P1_1 | — (2) | — (2) | — (2) | — (2) | IIO0_1 output | IIO1_1 output | — (2) |
| P1_2 | P1_2 | — (2) | — (2) | — (2) | — (2) | IIO0_2 output | IIO1_2 output | — (2) |
| P1_3 | P1_3 | — (2) | — (2) | — (2) | — (2) | IIO0_3 output | IIO1_3 output | — (2) |
| P1_4 | P1_4 | — (2) | — (2) | — (2) | — (2) | IIO0_4 output | IIO1_4 output | — (2) |
| P1_5 | P1_5 | — (2) | — (2) | — (2) | — (2) | IIO0_5 output | IIO1_5 output | — (2) |
| P1_6 | P1_6 | — (2) | — (2) | — (2) | — (2) | IIO0_6 output | IIO1_6 output | — (2) |
| P1_7 | P1_7 | — (2) | — (2) | — (2) | — (2) | IIO0_7 output | IIO1_7 output | — (2) |

2. Do not use this combination.

**Figure 26.4 Registers P1_0S to P1_7S**

The port P1_i (i = 0 to 7) shares a pin with the intelligent I/O groups 0 and 1 (IIO0 and IIO1), and the external interrupt input pin.
To use as an output pin, the PD1_i bit should be set to 1 (Port P1_i functions as output port) and a function should be selected according to the Figure 26.4. To use as an input pin, the PD1_i bit should be set to 0 (Port P1_i functions as input port).

## Port P2_i Function Select Register (i = 0 to 7)



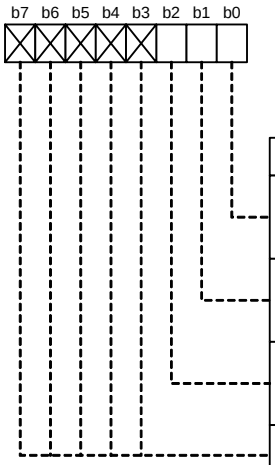

| Symbol | Address | Reset Value |
|---|---|---|
| P2_0S to P2_2S | 400B0h, 400B2h, 400B4h | 0XXX X000b |
| P2_3S to P2_5S | 400B6h, 400B8h, 400BAh | 0XXX X000b |
| P2_6S, P2_7S | 400BCh, 400BEh | 0XXX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P2_i Output Function Select Bit | b2 b1 b0<br>0 0 0 : I/O port P2_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : Do not use this combination<br>1 0 0 : Do not use this combination<br>1 0 1 : Do not use this combination<br>1 1 0 : Do not use this combination<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b6-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| ASEL | Port P2_i Analog Function Select Bit | 0: Function other than AN2_i<br>1: AN2_i | RW |

**Figure 26.5 Registers P2_0S to P2_7S**

The port P2_i (i = 0 to 7) shares a pin with the AN2_i pin for the A/D converter.
To use as the programmable I/O port, the P2_iS register should be set to 00h. To use as the A/D converter input pin, this register should be set to 80h and the PD2_i bit should be set to 0 (Port P2_i functions as input port).

## Port P3_i Function Select Register (i = 0 to 7)




| Symbol | Address | Reset Value |
|---|---|---|
| P3_0S to P3_2S | 400B1h, 400B3h, 400B5h | XXXX X000b |
| P3_3S to P3_5S | 400B7h, 400B9h, 400BBh | XXXX X000b |
| P3_6S, P3_7S | 400BDh, 400BFh | XXXX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P3_i Output Function Select Bit (1) | b2 b1 b0<br>0 0 0 : I/O port P3_i<br>0 0 1 : Timer output<br>0 1 0 : Three-phase motor control output<br>0 1 1 : Do not use this combination<br>1 0 0 : Do not use this combination<br>1 0 1 : Do not use this combination<br>1 1 0 : Do not use this combination<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b7-b3) | No register bits; should be written with 0 and read as undefined value | | — |

Notes:
1. Refer to the table below for each pin setting.

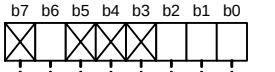
| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P3_0 | P3_0 | TA0OUT output | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P3_1 | P3_1 | TA3OUT output | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P3_2 | P3_2 | TA1OUT output | V | — (2) | — (2) | — (2) | — (2) | — (2) |
| P3_3 | P3_3 | — (2) | $\overline{V}$ | — (2) | — (2) | — (2) | — (2) | — (2) |
| P3_4 | P3_4 | TA2OUT output | W | — (2) | — (2) | — (2) | — (2) | — (2) |
| P3_5 | P3_5 | — (2) | $\overline{W}$ | — (2) | — (2) | — (2) | — (2) | — (2) |
| P3_6 | P3_6 | TA4OUT output | U | — (2) | — (2) | — (2) | — (2) | — (2) |
| P3_7 | P3_7 | — (2) | $\overline{U}$ | — (2) | — (2) | — (2) | — (2) | — (2) |

2. Do not use this combination.

**Figure 26.6 Registers P3_0S to P3_7S**

The port P3_i (i = 0 to 7) shares a pin with the timer output and the three-phase motor control output. To use as an output pin, the PD3_i bit should be set to 1 (Port P3_i functions as output port) and a function should be selected according to the Figure 26.6. To use as an input pin, the PD3_i register should be set to 0 (Port P3_i functions as input port).

## Port P4_i Function Select Register (i = 0 to 7)




| Symbol | Address | Reset Value |
|---|---|---|
| P4_0S to P4_2S | 400C0h, 400C2h, 400C4h | X0XX X000b |
| P4_3S to P4_5S | 400C6h, 400C8h, 400CAh | X0XX X000b |
| P4_6S, P4_7S | 400CCh, 400CEh | X0XX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P4_i Output Function Select Bit (1) | b2 b1 b0<br>0 0 0 : I/O port P4_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : UART3/UART6 output<br>1 0 0 : UART3/UART6 special function output<br>1 0 1 : IIO2 output<br>1 1 0 : Do not use this combination<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b5-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| NOD | N-channel Open Drain Output Select Bit | 0: Push-pull output<br>1: N-channel open drain output | RW |
| — (b7) | No register bit; should be written with 0 and read as undefined value | | — |

Notes:
1. Refer to the table below for each pin setting.

| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P4_0 | P4_0 | — (2) | — (2) | $\overline{\text{RTS3}}$ | — (2) | — (2) | — (2) | — (2) |
| P4_1 | P4_1 | — (2) | — (2) | CLK3 output | — (2) | — (2) | — (2) | — (2) |
| P4_2 | P4_2 | — (2) | — (2) | SCL3 output | STXD3 | — (2) | — (2) | — (2) |
| P4_3 | P4_3 | — (2) | — (2) | TXD3<br>SDA3 output | — (2) | OUTC2_0<br>ISTXD2<br>IEOUT | — (2) | — (2) |
| P4_4 | P4_4 | — (2) | — (2) | $\overline{\text{RTS6}}$ | — (2) | — (2) | — (2) | — (2) |
| P4_5 | P4_5 | — (2) | — (2) | CLK6 output | — (2) | — (2) | — (2) | — (2) |
| P4_6 | P4_6 | — (2) | — (2) | SCL6 output | STXD6 | — (2) | — (2) | — (2) |
| P4_7 | P4_7 | — (2) | — (2) | TXD6<br>SDA6 output | — (2) | — (2) | — (2) | — (2) |

2. Do not use this combination.

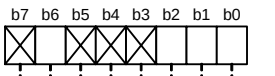
**Figure 26.7 Registers P4_0S to P4_7S**

The port P4_i (i = 0 to 7) shares a pin with the serial interface (UART3 and UART6) and the intelligent I/O group 2 (IIO2).

To use as an output pin, the PD4_i bit should be set to 1 (Port P4_i functions as output port) and a function should be selected according to the Figure 26.7. To use as an input pin, the PD4_i bit should be set to 0 (Port P4_i functions as input port).

Ports P4_0 to P4_7 are 5 V tolerant inputs. To use as an I/O pin with 5 V tolerant input enabled, the NOD bit should be set to 1.

## Port P5_i Function Select Register (i = 0 to 7)




| Symbol | Address | Reset Value |
|---|---|---|
| P5_0S, P5_1S | 400C1h, 400C3h | XXXX X000b |
| P5_2S, P5_3S | 400C5h, 400C7h | XXXX X000b |
| P5_4S, P5_5S | 400C9h, 400CBh | X0XX X000b |
| P5_6S, P5_7S | 400CDh, 400CFh | X0XX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P5_i Output Function Select Bit (1) | b2 b1 b0<br>0 0 0 : I/O port P5_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : UART7 output<br>1 0 0 : Do not use this combination<br>1 0 1 : Do not use this combination<br>1 1 0 : Do not use this combination<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b5-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| — (b6) (i = 0 to 3) NOD (i = 4 to 7) | No register bit; should be written with 0 and read as undefined value | | — |
| | N-channel Open Drain Output Select Bit | 0: Push-pull output<br>1: N-channel open drain output | RW |
| — (b7) | No register bit; should be written with 0 and read as undefined value | | — |

Notes:
1. Refer to the table below for each pin setting.

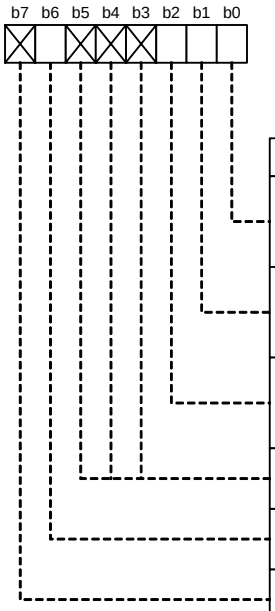
| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P5_0 | P5_0 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P5_1 | P5_1 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P5_2 | P5_2 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P5_3 | P5_3 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P5_4 | P5_4 | — (2) | — (2) | TXD7 | — (2) | — (2) | — (2) | — (2) |
| P5_5 | P5_5 | — (2) | — (2) | CLK7 output | — (2) | — (2) | — (2) | — (2) |
| P5_6 | P5_6 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P5_7 | P5_7 | — (2) | — (2) | $\overline{\text{RTS7}}$ | — (2) | — (2) | — (2) | — (2) |

2. Do not use this combination.

**Figure 26.8 Registers P5_0S to P5_7S**

The port P5_i (i = 0 to 7) shares a pin with the serial interface (UART7).
To use as an output pin, the PD5_i bit should be set to 1 (Port P5_i functions as output port) and a function should be selected according to the Figure 26.8. To use as an input pin, the PD5_i bit should be set to 0 (Port P5_i functions as input port).
Ports P5_4 to P5_7 are 5 V tolerant inputs. To use as an I/O pin with 5 V tolerant input enabled, the NOD bit should be set to 1.

## Port P6_i Function Select Register (i = 0 to 7)




| Symbol | Address | Reset Value |
|---|---|---|
| P6_0S to P6_2S | 400D0h, 400D2h, 400D4h | X0XX X000b |
| P6_3S to P6_5S | 400D6h, 400D8h, 400DAh | X0XX X000b |
| P6_6S, P6_7S | 400DCh, 400DEh | X0XX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P6_i Output Function Select Bit (1) | b2 b1 b0<br>0 0 0 : I/O port P6_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : UART0/UART1 output<br>1 0 0 : UART0/UART1 special function output<br>1 0 1 : IIO2 output<br>1 1 0 : Do not use this combination<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b5-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| NOD | N-channel Open Drain Output Select Bit | 0: Push-pull output<br>1: N-channel open drain output | RW |
| — (b7) | No register bit; should be written with 0 and read as undefined value | | — |

Notes:
1. Refer to the table below for each pin setting.

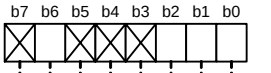
| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P6_0 | P6_0 | — (2) | — (2) | $\overline{\text{RTS0}}$ | — (2) | — (2) | — (2) | — (2) |
| P6_1 | P6_1 | — (2) | — (2) | CLK0 output | — (2) | — (2) | — (2) | — (2) |
| P6_2 | P6_2 | — (2) | — (2) | SCL0 output | STXD0 | — (2) | — (2) | — (2) |
| P6_3 | P6_3 | — (2) | — (2) | TXD0<br>SDA0 output | — (2) | — (2) | — (2) | — (2) |
| P6_4 | P6_4 | — (2) | — (2) | $\overline{\text{RTS1}}$ | — (2) | OUTC_1<br>ISCLK2 output | — (2) | — (2) |
| P6_5 | P6_5 | — (2) | — (2) | CLK1 output | — (2) | — (2) | — (2) | — (2) |
| P6_6 | P6_6 | — (2) | — (2) | SCL1 output | STXD1 | — (2) | — (2) | — (2) |
| P6_7 | P6_7 | — (2) | — (2) | TXD1<br>SDA1 output | — (2) | — (2) | — (2) | — (2) |

2. Do not use this combination.

**Figure 26.9 Registers P6_0S to P6_7S**

The port P6_i (i = 0 to 7) shares a pin with the serial interface (UART0 and UART1) and the intelligent I/O group 2 (IIO2).
To use as an output pin, the PD6_i bit should be set to 1 (Port P6_i functions as output port) and a function should be selected according to the Figure 26.9. To use as an input pin, the PD6_i bit should be set to 0 (Port P6_i functions as input port).
Ports P6_0 to P6_7 are 5 V tolerant inputs. To use as an I/O pin with 5 V tolerant input enabled, the NOD bit should be set to 1.

## Port P7_i Function Select Register (i = 0 to 7)




| Symbol | Address | Reset Value |
|---|---|---|
| P7_0S to P7_2S | 400D1h, 400D3h, 400D5h | X0XX X000b |
| P7_3S to P7_5S | 400D7h, 400D9h, 400DBh | X0XX X000b |
| P7_6S, P7_7S | 400DDh, 400DFh | X0XX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P7_i Output Function Select Bit (1) | b2 b1 b0<br>0 0 0 : I/O port P7_i<br>0 0 1 : Timer output<br>0 1 0 : Three-phase motor control output<br>0 1 1 : UART2/UART5/MMI2C output<br>1 0 0 : UART2 special function output<br>1 0 1 : IIO2/CAN0 output<br>1 1 0 : IIO1 output<br>1 1 1 : UART8 output | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b5-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| NOD | N-channel Open Drain Output Select Bit | 0: Push-pull output<br>1: N-channel open drain output | RW |
| — (b7) | No register bit; should be written with 0 and read as undefined value | | — |

Notes:
1. Refer to the table below for each pin setting.

| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P7_0 | P7_0 | TA0OUT output | — (2) | TXD2<br>SDA2 output<br>MSDA output | — (2) | OUTC2_0<br>ISTXD2<br>IEOUT | IIO1_6 output | — (2) |
| P7_1 | P7_1 | — (2) | — (2) | SCL2 output<br>MSCL output | STXD2 | OUTC2_2 | IIO1_7 output | — (2) |
| P7_2 | P7_2 | TA1OUT output | V | CLK2 output | — (2) | — (2) | — (2) | — (2) |
| P7_3 | P7_3 | — (2) | $\overline{V}$ | $\overline{RTS2}$ | — (2) | — (2) | IIO1_0 output | TXD8 |
| P7_4 | P7_4 | TA2OUT output | W | — (2) | — (2) | — (2) | IIO1_1 output | CLK8 output |
| P7_5 | P7_5 | — (2) | $\overline{W}$ | — (2) | — (2) | — (2) | IIO1_2 output | — (2) |
| P7_6 | P7_6 | TA3OUT output | — (2) | TXD5<br>SDA5 output | — (2) | CAN0OUT | IIO1_3 output | $\overline{RTS8}$ |
| P7_7 | P7_7 | — (2) | — (2) | CLK5 output | — (2) | — (2) | IIO1_4 output | — (2) |

2. Do not use this combination.

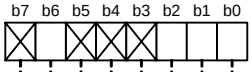
**Figure 26.10 Registers P7_0S to P7_7S**

The port P7_i (i=0 to 7) shares a pin with the timer, the three-phase motor control, the serial interface (UART2, UART5, and UART8), the multi-master I2C-bus interface, the intelligent I/O groups 1 and 2 (IIO1 and IIO2), and the CAN module.

To use as an output pin, the PD7_i bit should be set to 1 (Port P7_i functions as output port) and a function should be selected according to the Figure 26.10. To use as an input pin, the PD7_i bit should be set to 0 (Port P7_i functions as input port).

Ports P7_0 to P7_7 are 5 V tolerant inputs. To use as an I/O pin with 5 V tolerant input enabled, the NOD bit should be set to 1.

## Port P8_i Function Select Register (i = 0 to 4, 6, 7)




| Symbol | Address | Reset Value |
|---|---|---|
| P8_0S, P8_1S | 400E0h, 400E2h | X0XX X000b |
| P8_2S, P8_3S | 400E4h, 400E6h | X0XX X000b |
| P8_4S | 400E8h | XXXX X000b |
| P8_6S, P8_7S | 400ECh, 400EEh | XXXX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P8_i Output Function Select Bit (1) | b2 b1 b0<br>0 0 0 : I/O port P8_i<br>0 0 1 : Timer output<br>0 1 0 : Three-phase motor control output<br>0 1 1 : UART5 output<br>1 0 0 : UART5 special function output<br>1 0 1 : CAN0 output<br>1 1 0 : IIO1/CAN1 output<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b5-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| NOD (i = 0 to 3) | N-channel Open Drain Output Select Bit | 0: Push-pull output<br>1: N-channel open drain output | RW |
| — (b6) (i = 4, 6, 7) | No register bit; should be written with 0 and read as undefined value | | — |
| — (b7) | No register bit; should be written with 0 and read as undefined value | | — |

Notes:
1. Refer to the table below for each pin setting.

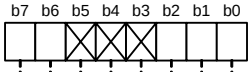
| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P8_0 | P8_0 | TA4OUT output | U | SCL5 output | STXD5 | — (2) | — (2) | — (2) |
| P8_1 | P8_1 | — (2) | $\overline{U}$ | $\overline{RTS5}$ | — (2) | — (2) | IIO1_5 output | — (2) |
| P8_2 | P8_2 | — (2) | — (2) | — (2) | — (2) | CAN0OUT | CAN1OUT | — (2) |
| P8_3 | P8_3 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P8_4 | P8_4 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P8_6 | P8_6 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P8_7 | P8_7 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |

2. Do not use this combination.

**Figure 26.11 Registers P8_0S to P8_4S, P8_6S, and P8_7S**

The port P8_i (i = 0 to 4, 6, and 7) shares a pin with the timer, the three-phase motor control, the serial interface (UART5), the intelligent I/O group 1 (IIO1), the CAN module, and the external interrupt input pin. To use as an output pin, the PD8_i bit should be set to 1 (Port P8_i functions as output port) and a function should be selected according to the Figure 26.11. To use as an input pin, the PD8_i bit should be set to 0 (Port P8_i functions as input port).
Ports P8_0 to P8_3 are 5 V tolerant inputs. To use as an I/O pin with 5 V tolerant input enabled, the NOD bit should be set to 1.

## Port P9_i Function Select Register (i = 0 to 7) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| P9_0S to P9_2S | 400E1h, 400E3h, 400E5h | X0XX X000b |
| P9_3S to P9_5S | 400E7h, 400E9h, 400EBh | 00XX X000b |
| P9_6S | 400EDh | 00XX X000b |
| P9_7S | 400EFh | X0XX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P9_i Output Function Select Bit (2) | b2 b1 b0<br>0 0 0 : I/O port P9_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : UART3/UART4 output<br>1 0 0 : UART3/UART4 special function output<br>1 0 1 : IIO2 output<br>1 1 0 : CAN1 output<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b5-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| NOD | N-channel Open Drain Output Select Bit | 0: Push-pull output<br>1: N-channel open drain output | RW |
| — (b7) (i = 0 to 2, 7) ASEL (i = 3 to 6) | No register bit; should be written with 0 and read as undefined value | | — |
| | Port P9_i (i = 3 to 6) Analog Functions Select Bit | 0: Function other than Analog pin<br>1: Analog pin | RW |

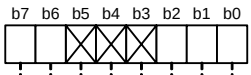
Notes:
1. The instruction to set this register should be written immediately after the instruction to set the PRC2 bit in the PRCR register to 1 (write enabled). Any interrupt or DMA transfer should not be generated between these two instructions.
2. Refer to the table below for each pin setting.

| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P9_0 | P9_0 | — (3) | — (3) | CLK3 output | — (3) | — (3) | — (3) | — (3) |
| P9_1 | P9_1 | — (3) | — (3) | SCL3 output | STXD3 | — (3) | — (3) | — (3) |
| P9_2 | P9_2 | — (3) | — (3) | TXD3<br>SDA3 output | — (3) | OUTC2_0<br>ISTXD2<br>IEOUT | — (3) | — (3) |
| P9_3 | P9_3 | — (3) | — (3) | $\overline{\text{RTS3}}$ | — (3) | — (3) | — (3) | — (3) |
| P9_4 | P9_4 | — (3) | — (3) | $\overline{\text{RTS4}}$ | — (3) | — (3) | — (3) | — (3) |
| P9_5 | P9_5 | — (3) | — (3) | CLK4 output | — (3) | — (3) | — (3) | — (3) |
| P9_6 | P9_6 | — (3) | — (3) | TXD4<br>SDA4 output | — (3) | — (3) | CAN1OUT | — (3) |
| P9_7 | P9_7 | — (3) | — (3) | SCL4 output | STXD4 | — (3) | — (3) | — (3) |

3. Do not use this combination.

**Figure 26.12 Registers P9_0S to P9_7S (144-pin package)**

## Port P9_i Function Select Register (i = 3 to 7) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| P9_3S | 400E7h | 0XXX X000b |
| P9_4S to P9_6S | 400E9h, 400EBh, 400EDh | 00XX X000b |
| P9_7S | 400EFh | X0XX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P9_i Output Function Select Bit (2) | b2 b1 b0<br>0 0 0 : I/O port P9_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : UART4 output<br>1 0 0 : UART4 special function output<br>1 0 1 : Do not use this combination<br>1 1 0 : CAN1 output<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b5-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| — (b6) (i = 0 to 3) NOD (i = 4 to 7) | No register bit; should be written with 0 and read as undefined value | | — |
| | N-channel Open Drain Output Select Bit | 0: Push-pull output<br>1: N-channel open drain output | RW |
| — (b7) (i = 0 to 2, 7) ASEL (i = 3 to 6) | No register bit; should be written with 0 and read as undefined value | | — |
| | Port P9_i (i = 3 to 6) Analog Functions Select Bit | 0: Function other than Analog pin<br>1: Analog pin | RW |

Notes:
1. The instruction to set this register should be written immediately after the instruction to set the PRC2 bit in the PRCR register to 1 (write enabled). Any interrupt or DMA transfer should not be generated between these two instructions.
2. Refer to the table below for each pin setting.

| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P9_3 | P9_3 | — (3) | — (3) | — (3) | — (3) | — (3) | — (3) | — (3) |
| P9_4 | P9_4 | — (3) | — (3) | $\overline{\text{RTS4}}$ | — (3) | — (3) | — (3) | — (3) |
| P9_5 | P9_5 | — (3) | — (3) | CLK4 output | — (3) | — (3) | — (3) | — (3) |
| P9_6 | P9_6 | — (3) | — (3) | TXD4<br>SDA4 output | — (3) | — (3) | CAN1OUT | — (3) |
| P9_7 | P9_7 | — (3) | — (3) | SCL4 output | STXD4 | — (3) | — (3) | — (3) |

3. Do not use this combination.

**Figure 26.13 Registers P9_3S to P9_7S (100-pin package)**

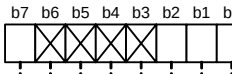
The port P9_i (i = 0 to 7) shares a pin with the serial interface (UART3 and UART4), the intelligent I/O group 2 (IIO2) and the CAN module. In particular, the port P9_i (i = 3 to 6) also shares a pin with the A/D converter I/O (ANEX0 and ANEX1) pin and the D/A converter output pin.

To use as the A/D converter pin or the D/A converter pin, the P9_iS register should be set to 80h and the PD9_i bit should be set to 0 (Port P9_i functions as input port) irrespective of the input/output state.

To use as an output pin of functions other than the A/D converter or the D/A converter, the PD9_i bit should be set to 1 (Port P9_i functions as output port) and a function should be selected according to the Figure 26.12. To use as an input pin of functions other than the A/D converter or the D/A converter, the PD_9i bit should be set to 0 (Port P9_i functions as input port).

When the NOD bit is set to 1, the corresponding pin functions as an N-channel open drain output.

## Port P10_i Function Select Register (i = 0 to 7)




| Symbol | Address | Reset Value |
|---|---|---|
| P10_0S to P10_2S | 400F0h, 400F2h, 400F4h | 0XXX X000b |
| P10_3S to P10_5S | 400F6h, 400F8h, 400FAh | 0XXX X000b |
| P10_6S, P10_7S | 400FCh, 400FEh | 0XXX X000b |

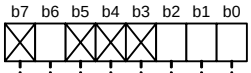
| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P10_i Output Function Select Bit | b2 b1 b0<br>0 0 0 : I/O port P10_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : Do not use this combination<br>1 0 0 : Do not use this combination<br>1 0 1 : Do not use this combination<br>1 1 0 : Do not use this combination<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b6-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| ASEL | Port P10_i Analog Functions Select Bit | 0: Function other than AN_i<br>1: AN_i | RW |

**Figure 26.14 Registers P10_0S to P10_7S**

The port P10_i (i = 0 to 7) shares a pin with the AN_i input pin for the A/D converter and the key input interrupt pin.

To use as the programmable I/O port, the P10_iS register should be set to 00h. To use as an input pin (except for the A/D converter), the PD10_i bit should be set to 0 (Port P10_i functions as input port). To use as an input pin for the A/D converter, the P10_iS register should be set to 80h and the PD10_i bit should be set to 0 (Port P10_i functions as input port).

## Port P11_i Function Select Register (i = 0 to 4)




| Symbol | Address | Reset Value |
|---|---|---|
| P11_0S to P11_2S | 400F1h, 400F3h, 400F5h | X0XX X000b |
| P11_3S | 400F7h | X0XX X000b |
| P11_4S | 400F9h | XXXX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P11_i Output Function Select Bit (1) | b2 b1 b0<br>0 0 0 : I/O port P11_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : UART8 output<br>1 0 0 : Do not use this combination<br>1 0 1 : Do not use this combination<br>1 1 0 : IIO1 output<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b5-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| NOD (i = 0 to 3) | N-channel Open Drain Output Select Bit | 0: Push-pull output<br>1: N-channel open drain output | RW |
| — (b6) (i = 4) | No register bit; should be written with 0 and read as undefined value | | — |
| — (b7) | No register bit; should be written with 0 and read as undefined value | | — |

Notes:
1. Refer to the table below for each pin setting.

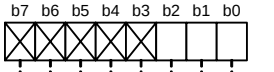
| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P11_0 | P11_0 | — (2) | — (2) | TXD8 | — (2) | — (2) | IIO1_0 output | — (2) |
| P11_1 | P11_1 | — (2) | — (2) | CLK8 output | — (2) | — (2) | IIO1_1 output | — (2) |
| P11_2 | P11_2 | — (2) | — (2) | — (2) | — (2) | — (2) | IIO1_2 output | — (2) |
| P11_3 | P11_3 | — (2) | — (2) | $\overline{\text{RTS8}}$ | — (2) | — (2) | IIO1_3 output | — (2) |
| P11_4 | P11_4 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |

2. Do not use this combination.

**Figure 26.15 Registers P11_0S to P11_4S**

The port P11_i (i = 0 to 4) shares a pin with the serial interface (UART8) and the intelligent I/O group 1 (IIO1).
To use as an output pin, the PD11_i bit should be set to 1 (Port P11_i functions as output port) and a function should be selected according to the Figure 26.15. To use as an input pin, the PD11_i bit should be set to 0 (Port P11_i functions as input port).
To use as an N-channel open drain output, the NOD bit should be set to 1.

## Port P12_i Function Select Register (i = 0 to 7)




| Symbol | Address | Reset Value |
|---|---|---|
| P12_0S to P12_2S | 40100h, 40102h, 40104h | XXXX X000b |
| P12_3S to P12_5S | 40106h, 40108h, 4010Ah | XXXX X000b |
| P12_6S, P12_7S | 4010Ch, 4010Eh | XXXX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P12_i Output Function Select Bit (1) | b2 b1 b0<br>0 0 0 : I/O port P12_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : UART6 output<br>1 0 0 : UART6 special function<br>1 0 1 : Do not use this combination<br>1 1 0 : Do not use this combination<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b7-b3) | No register bits; should be written with 0 and read as undefined value | | — |

Notes:
1. Refer to the table below for each pin setting.

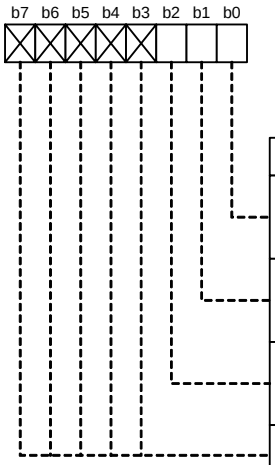
| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P12_0 | P12_0 | — (2) | — (2) | TXD6<br>SDA6 output | — (2) | — (2) | — (2) | — (2) |
| P12_1 | P12_1 | — (2) | — (2) | CLK6 output | — (2) | — (2) | — (2) | — (2) |
| P12_2 | P12_2 | — (2) | — (2) | SCL6 output | STXD6 | — (2) | — (2) | — (2) |
| P12_3 | P12_3 | — (2) | — (2) | $\overline{\text{RTS6}}$ | — (2) | — (2) | — (2) | — (2) |
| P12_4 | P12_4 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P12_5 | P12_5 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P12_6 | P12_6 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |
| P12_7 | P12_7 | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) | — (2) |

2. Do not use this combination.

**Figure 26.16 Registers P12_0S to P12_7S**

The port P12_i (i = 0 to 7) shares a pin with the serial interface (UART6).
To use as an output pin, the PD12_i bit should be set to 1 (Port P12_i functions as output port) and a function should be selected according to the Figure 26.16. To use as an input pin, the PD12_i bit should be set to 0 (Port P12_i functions as input port).
When the NOD bit is set to 1, the corresponding pin functions as an N-channel open drain output.

## Port P13_i Function Select Register (i = 0 to 7)




| Symbol | Address | Reset Value |
|---|---|---|
| P13_0S to P13_2S | 40101h, 40103h, 40105h | XXXX X000b |
| P13_3S to P13_5S | 40107h, 40109h, 4010Bh | XXXX X000b |
| P13_6S, P13_7S | 4010Dh, 4010Fh | XXXX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P13_i Output Function Select Bit (1) | b2 b1 b0<br>0 0 0 : I/O port P13_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : Do not use this combination<br>1 0 0 : Do not use this combination<br>1 0 1 : IIO2 output<br>1 1 0 : Do not use this combination<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b7-b3) | No register bits; should be written with 0 and read as undefined value | | — |

Notes:
1. Refer to the table below for each pin setting.

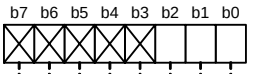
| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P13_0 | P13_0 | — (2) | — (2) | — (2) | — (2) | OUTC2_4 | — (2) | — (2) |
| P13_1 | P13_1 | — (2) | — (2) | — (2) | — (2) | OUTC2_5 | — (2) | — (2) |
| P13_2 | P13_2 | — (2) | — (2) | — (2) | — (2) | OUTC2_6 | — (2) | — (2) |
| P13_3 | P13_3 | — (2) | — (2) | — (2) | — (2) | OUTC2_3 | — (2) | — (2) |
| P13_4 | P13_4 | — (2) | — (2) | — (2) | — (2) | OUTC2_0<br>ISTXD2<br>IEOUT | — (2) | — (2) |
| P13_5 | P13_5 | — (2) | — (2) | — (2) | — (2) | OUTC2_2 | — (2) | — (2) |
| P13_6 | P13_6 | — (2) | — (2) | — (2) | — (2) | OUTC2_1<br>ISCLK2 output | — (2) | — (2) |
| P13_7 | P13_7 | — (2) | — (2) | — (2) | — (2) | OUTC2_7 | — (2) | — (2) |

2. Do not use this combination.

**Figure 26.17 Registers P13_0S to P13_7S**

The port P13_i (i = 0 to 7) shares a pin with the intelligent I/O group 2 (IIO2).
To use as an output pin, the PD13_i bit should be set to 1 (Port P13_i functions as output port) and a function should be selected according to the Figure 26.17. To use as an input pin, the PD13_i bit should be set to 0 (Port P13_i functions as input port).

## Port P14_i Function Select Register (i = 3 to 6)



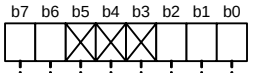

| Symbol | Address | Reset Value |
|---|---|---|
| P14_3S to P14_5S | 40116h, 40118h, 4011Ah | XXXX X000b |
| P14_6S | 4011Ch | XXXX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P14_i Output Function Select Bit | b2 b1 b0<br>0 0 0 : I/O port P14_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : Do not use this combination<br>1 0 0 : Do not use this combination<br>1 0 1 : Do not use this combination<br>1 1 0 : Do not use this combination<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b7-b3) | No register bits; should be written with 0 and read as undefined value | | — |

**Figure 26.18 Registers P14_3S to P14_6S**

The port P14_i (i = 3 to 6) shares a pin with the external interrupt input pin. The P14_iS register should be set to 00h (I/O port).

## Port P15_i Function Select Register (i = 0 to 7)




| Symbol | Address | Reset Value |
|---|---|---|
| P15_0S to P15_2S | 40111h, 40113h, 40115h | 00XX X000b |
| P15_3S to P15_5S | 40117h, 40119h, 4011Bh | 00XX X000b |
| P15_6S, P15_7S | 4011Dh, 4011Fh | 00XX X000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PSEL0 | Port P15_i Output Function Select Bit (1) | b2 b1 b0<br>0 0 0 : I/O port P15_i<br>0 0 1 : Do not use this combination<br>0 1 0 : Do not use this combination<br>0 1 1 : UART6/UART7 output<br>1 0 0 : UART6 special function<br>1 0 1 : IIO0 output<br>1 1 0 : Do not use this combination<br>1 1 1 : Do not use this combination | RW |
| PSEL1 | | | RW |
| PSEL2 | | | RW |
| — (b5-b3) | No register bits; should be written with 0 and read as undefined value | | — |
| NOD | N-channel Open Drain Output Select Bit | 0: Push-pull output<br>1: N-channel open drain output | RW |
| ASEL | Port P15_i Analog Function Select Bit | 0: Function other than AN15_i<br>1: AN15_i | RW |

Notes:
1. Refer to the table below for each pin setting.

| Port | Bits PSEL2 to PSEL0 Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| P15_0 | P15_0 | — (2) | — (2) | TXD7 | — (2) | IIO0_0 output | — (2) | — (2) |
| P15_1 | P15_1 | — (2) | — (2) | CLK7 output | — (2) | IIO0_1 output | — (2) | — (2) |
| P15_2 | P15_2 | — (2) | — (2) | — (2) | — (2) | IIO0_2 output | — (2) | — (2) |
| P15_3 | P15_3 | — (2) | — (2) | $\overline{\text{RTS7}}$ | — (2) | IIO0_3 output | — (2) | — (2) |
| P15_4 | P15_4 | — (2) | — (2) | TXD6<br>SDA6 output | — (2) | IIO0_4 output | — (2) | — (2) |
| P15_5 | P15_5 | — (2) | — (2) | SCL6 output | STXD6 | IIO0_5 output | — (2) | — (2) |
| P15_6 | P15_6 | — (2) | — (2) | CLK6 output | — (2) | IIO0_6 output | — (2) | — (2) |
| P15_7 | P15_7 | — (2) | — (2) | $\overline{\text{RTS6}}$ | — (2) | IIO0_7 output | — (2) | — (2) |

2. Do not use this combination.

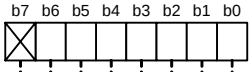
**Figure 26.19 Registers P15_0S to P15_7S**

The port P15_i (i = 0 to 7) shares a pin with the serial interface (UART6 and UART7), the intelligent I/O group 0 (IIO0), and the AN15_i input pin for the A/D converter.
To use as an output pin, the PD15_i bit should be set to 1 (Port P15_i functions as output port) and a function should be selected according to the Figure 26.19. To use as an input pin (except for the A/D converter), the PD15_i bit should be set to 0 (Port P15_i functions as input port). To use as an input pin for the A/D converter, the P15_iS register should be set to 80h and the PD15_i bit should be set to 0.
To use as an N-channel open drain output, the NOD bit should be set to 1.

## 26.3 Input Function Select Register

When a peripheral function input is assigned to multiple pins, this register selects which input pin should be connected to the peripheral function.
Figure 26.20 to Figure 26.23 show the input function select registers.

## Input Function Select Register 0




| Symbol | Address | Reset Value |
|---|---|---|
| IFS0 | 40098h | X000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IFS00 | Timer A Input Pin Switch Bit (1) | Assign timer A input to<br>0: Port P3<br>1: Port P7/port P8 | RW |
| IFS01 | Timer B Input Pin Switch Bit (2) | Assign timer B input to<br>0: Port P6<br>1: Port P7/port P9 | RW |
| IFS02 | UART6 Input Pin Switch Bit (3) | Assgin UART 6 input to<br>b3 b2<br>0 0 : Port P4<br>0 1 : Do not use this combination<br>1 0 : Port P15<br>1 1 : Port P12 | RW |
| IFS03 | | | RW |
| IFS04 | UART8 Input Pin Switch Bit (4) | Assign UART8 input to<br>0: Port P7<br>1: Port P11 | RW |
| IFS05 | UART7 Input Pin Switch Bit (5) | Assign UART7 input to<br>0: Port P5<br>1: Port P15 | RW |
| IFS06 | UART3 Input Pin Switch Bit (6) | Assign UART3 input to<br>0: Port P4<br>1: Port P9 | RW |
| —<br>(b7) | No register bit; should be written with 0 and read as undefined value | | — |

Notes:

1. Refer to the table below for each pin setting of the timer A.

| IFS00 | TA0OUT input | TA1OUT input | TA1IN | TA2OUT input | TA2IN | TA3OUT input | TA4OUT input | TA4IN |
|---|---|---|---|---|---|---|---|---|
| 0 | P3_0 | P3_2 | P3_3 | P3_4 | P3_5 | P3_1 | P3_6 | P3_7 |
| 1 | P7_0 | P7_2 | P7_3 | P7_4 | P7_5 | P7_6 | P8_0 | P8_1 |

2. Refer to the table below for each pin setting of the timer B. This bit should be set to 0 in the 100-pin package.

| IFS01 | TB0IN | TB1IN | TB2IN |
|---|---|---|---|
| 0 | P6_0 | P6_1 | P6_2 |
| 1 | P9_0 | P9_1 | P9_2 |

3. Refer to the table below for each pin setting of UART6. This bit should be set to 00b in the 100-pin package.

| IFS03 | IFS02 | SDA6 input/SRXD6 | RXD6/SCL6 input | CLK6 input | $\overline{\text{CTS6}}/\overline{\text{SS6}}$ |
|---|---|---|---|---|---|
| 0 | 0 | P4_7 | P4_6 | P4_5 | P4_4 |
| 1 | 0 | P15_4 | P15_5 | P15_6 | P15_7 |
| 1 | 1 | P12_0 | P12_2 | P12_1 | P12_3 |

4. Refer to the table below for each pin setting of UART8. This bit should be set to 00b in the 100-pin package.

| IFS04 | CLK8 input | RXD8 | $\overline{\text{CTS8}}$ |
|---|---|---|---|
| 0 | P7_4 | P7_5 | P7_6 |
| 1 | P11_1 | P11_2 | P11_3 |

5. Refer to the table below for each pin setting of UART7. This bit should be set to 00b in the 100-pin package.

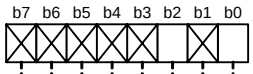
| IFS05 | CLK7 input | RXD7 | $\overline{\text{CTS7}}$ |
|---|---|---|---|
| 0 | P5_5 | P5_6 | P5_7 |
| 1 | P15_1 | P15_2 | P15_3 |

6. Refer to the table below for each pin setting of UART3. This bit should be set to 00b in the 100-pin package.

| IFS06 | SDA3 input/SRXD3 | RXD3/SCL3 input | CLK3 input | $\overline{\text{CTS3}}/\overline{\text{SS3}}$ |
|---|---|---|---|---|
| 0 | P4_3 | P4_2 | P4_1 | P4_0 |
| 1 | P9_2 | P9_1 | P9_0 | P9_3 |

**Figure 26.20 IFS0 Register**

Input Function Select Register 1




| Symbol | Address | Reset Value |
|---|---|---|
| IFS1 | 40099h | XXXX X0X0b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IFS10 | CAN0 Input Pin Switch Bit | Assign CAN0IN/$\overline{\text{CAN0WU}}$ input to<br>0: Port P7_7<br>1: Port P8_3 | RW |
| —<br>(b1) | No register bit; should be written with 0 and read as undefined value | | — |
| IFS12 | CAN1 Input Pin Switch Bit | Assign CAN1IN/$\overline{\text{CAN1WU}}$ input to<br>0: Port P9_5<br>1: Port P8_3 | RW |
| —<br>(b7-b3) | No register bits; should be written with 0 and read as undefined value | | — |

**Figure 26.21 IFS1 Register**

## Input Function Select Register 2

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| IFS2 | 4009Ah | 0000 00X0b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IFS20 | Intelligent I/O Group 0 Input Pin Switch Bit (1) | Assign IIO0 input to<br>0: Port P1<br>1: Port P15 | RW |
| —<br>(b1) | No register bit; should be written with 0 and read as undefined value | | — |
| IFS22 | Intelligent I/O Group 0 Two-Phase Pulse Input Pin Switch Bit (2) | Assign this input to<br>b3 b2<br>0 0 : Port P8 and $\overline{INT1}$<br>0 1 : Port P7 and $\overline{INT0}$<br>1 0 : Port P3 and $\overline{INT1}$<br>1 1 : Port P3 and $\overline{INT0}$ | RW |
| IFS23 | | | RW |
| IFS24 | Intelligent I/O Group 1 Input Pin Switch Bit (3) | Assign IIO1 input to<br>b5 b4<br>0 0 : Port P7/port P8<br>0 1 : Port P11<br>1 0 : Port P1<br>1 1 : Do not use this combination | RW |
| IFS25 | | | RW |
| IFS26 | Intelligent I/O Group 1 Two-Phase Pulse Input Pin Switch Bit (4) | Assign this input to<br>b7 b6<br>0 0 : Port P8 and $\overline{INT1}$<br>0 1 : Port P7 and $\overline{INT0}$<br>1 0 : Port P3 and $\overline{INT1}$<br>1 1 : Port P3 and $\overline{INT0}$ | RW |
| IFS27 | | | RW |

Notes:

1. Refer to the table below for each pin setting of the Intelligent I/O group 0. This bit should be set to 0 in the 100-pin package.

| IFS20 | IIO0_0 input | IIO0_1 input | IIO0_2 input | IIO0_3 input | IIO0_4 input | IIO0_5 input | IIO0_6 input | IIO0_7 input |
|---|---|---|---|---|---|---|---|---|
| 0 | P1_0 | P1_1 | P1_2 | P1_3 | P1_4 | P1_5 | P1_6 | P1_7 |
| 1 | P15_0 | P15_1 | P15_2 | P15_3 | P15_4 | P15_5 | P15_6 | P15_7 |

2. Refer to the table below for each pin setting of the Intelligent I/O group 0 in two-phase pulse signal processing mode.

| IFS23 | IFS22 | UD0A | UD0B | UD0Z |
|---|---|---|---|---|
| 0 | 0 | P8_0 | P8_1 | P8_3 ($\overline{INT1}$) |
| 0 | 1 | P7_6 | P7_7 | P8_2 ($\overline{INT0}$) |
| 1 | 0 | P3_0 | P3_1 | P8_3 ($\overline{INT1}$) |
| 1 | 1 | P3_0 | P3_1 | P8_2 ($\overline{INT0}$) |

3. Refer to the table below for each pin setting of the Intelligent I/O group 1. This bit should not be set to 01b in the 100-pin package.

| IFS25 | IFS24 | IIO1_0 input | IIO1_1 input | IIO1_2 input | IIO1_3 input | IIO1_4 input | IIO1_5 input | IIO1_6 input | IIO1_7 input |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | P7_3 | P7_4 | P7_5 | P7_6 | P7_7 | P8_1 | P7_0 | P7_1 |
| 0 | 1 | P11_0 | P11_1 | P11_2 | P11_3 | — | — | — | — |
| 1 | 0 | P1_0 | P1_1 | P1_2 | P1_3 | P1_4 | P1_5 | P1_6 | P1_7 |

4. Refer to the table below for each pin setting of the Intelligent I/O group 1 in two-phase pulse signal processing mode.

| IFS27 | IFS26 | UD1A | UD1B | UD1Z |
|---|---|---|---|---|
| 0 | 0 | P8_0 | P8_1 | P8_3 ($\overline{INT1}$) |
| 0 | 1 | P7_6 | P7_7 | P8_2 ($\overline{INT0}$) |
| 1 | 0 | P3_0 | P3_1 | P8_3 ($\overline{INT1}$) |
| 1 | 1 | P3_0 | P3_1 | P8_2 ($\overline{INT0}$) |

**Figure 26.22 IFS2 Register**

## Input Function Select Register 3

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| IFS3 | 4009Bh | XXXX XX00b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| IFS30 | Intelligent I/O Group 2 Input Pin Switch Bit (1) | Assign IIO2 input to<br>b1 b0<br>0 0 : Port P6/port P7<br>0 1 : Port P6/port P9<br>1 0 : Port P13<br>1 1 : Port P6/port P4 | RW |
| IFS31 | | | RW |
| — (b7-b2) | No register bits; should be written with 0 and read as undefined value | | — |

Note:
1. Refer to the table below for each pin setting of the Intelligent I/O group 2. This bit should be set to 00b or 11b in the 100-pin package.

| IFS31 | IFS30 | ISCLK2 input | ISRXD2/IEIN |
|---|---|---|---|
| 0 | 0 | P6_4 | P7_1 |
| 0 | 1 | P6_4 | P9_1 |
| 1 | 0 | P13_6 | P13_5 |
| 1 | 1 | P6_4 | P4_2 |

**Figure 26.23 IFS3 Register**

## 26.4 Pull-up Control Registers 0 to 4 (Registers PUR0 to PUR4)

Figure 26.24 to Figure 26.28 show registers PUR0 to PUR4.

These registers enable/disable the pull-up resistors for every group of four pins. To enable the pull-up resistors, the corresponding bits in registers PUR0 to PUR4 should be set to 1 (pull-up resistor enabled) and the respective bits in the direction register should be set to 0 (input).

In memory expansion mode or microprocessor mode, the pull-up control bits for ports P0 to P5 and P11 to P13, running as the bus control pins, should be set to 0 (pull-up resistor disabled). The pull-up resistors are enable for ports P0, P1, and P11 to P13 when these pins function as input ports in these modes.

Pull-Up Control Register 0 (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| PUR0 | 03F0h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PU00 | P0_0 to P0_3 Pull-Up Control Bit | Control pull-up setting for corresponding ports<br>0: Pull-up resistor disabled<br>1: Pull-up resistor enabled | RW |
| PU01 | P0_4 to P0_7 Pull-Up Control Bit | | RW |
| PU02 | P1_0 to P1_3 Pull-Up Control Bit | | RW |
| PU03 | P1_4 to P1_7 Pull-Up Control Bit | | RW |
| PU04 | P2_0 to P2_3 Pull-Up Control Bit | | RW |
| PU05 | P2_4 to P2_7 Pull-Up Control Bit | | RW |
| PU06 | P3_0 to P3_3 Pull-Up Control Bit | | RW |
| PU07 | P3_4 to P3_7 Pull-Up Control Bit | | RW |

Note:
1. In memory expansion mode or microprocessor mode, each bit of the PUR0 register should be set to 0 since ports P0 to P3 are used as the bus control pins. However, the pull-up resistors are enabled for ports P0 and P1 when these pins function as I/O ports with 8-bit bus or multiplexed bus format.

**Figure 26.24 PUR0 Register**

## Pull-Up Control Register 1 (1)

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| PUR1 | 03F1h | XXXX X0XXb |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b1-b0) | No register bits; should be written with 0 and read as undefined value | | — |
| PU12 | P5_0 to P5_3 Pull-Up Control Bit | Control pull-up setting for corresponding ports<br>0: Pull-up resistor disabled<br>1: Pull-up resistor enabled | RW |
| — (b7-b3) | No register bits; should be written with 0 and read as undefined value | | — |

Note:
1. In memory expansion mode or microprocessor mode, each bit of the PUR1 register should be set to 0 since the port P5 functions as the bus control pin.

**Figure 26.25 PUR1 Register**

## Pull-Up Control Register 2

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| PUR2 | 03F2h | 000X XXXXb |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b4-b0) | No register bits; should be written with 0 and read as undefined value | | — |
| PU25 | P8_4 to P8_7 Pull-Up Control Bit (1) | Control pull-up setting for corresponding ports<br>0: Pull-up resistor disabled<br>1: Pull-up resistor enabled | RW |
| PU26 | P9_1 to P9_3 Pull-Up Control Bit (2) | | RW |
| PU27 | P9_4 to P9_7 Pull-Up Control Bit | | RW |

Notes:
1. The port P8_5 has no pull-up resistor.
2. Ports P9_0 and P9_2 have no pull-up resistor in the 100-pin package.

**Figure 26.26 PUR2 Register**

## Pull-Up Control Register 3

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| PUR3 | 03F3h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PU30 | P10_0 to P10_3 Pull-Up Control Bit | Control pull-up setting for corresponding ports<br>0: Pull-up resistor disabled<br>1: Pull-up resistor enabled | RW |
| PU31 | P10_4 to P10_7 Pull-Up Control Bit | | RW |
| PU32 | P11_0 to P11_3 Pull-Up Control Bit (1, 2) | | RW |
| PU33 | P11_4 Pull-Up Control Bit (1, 2) | | RW |
| PU34 | P12_0 to P12_3 Pull-Up Control Bit (1, 2) | | RW |
| PU35 | P12_4 to P12_7 Pull-Up Control Bit (1, 2) | | RW |
| PU36 | P13_0 to P13_3 Pull-Up Control Bit (1, 2) | | RW |
| PU37 | P13_4 to P13_7 Pull-Up Control Bit (1, 2) | | RW |

Notes:
1. Ports P11 to P13 are not available in the 100-pin package. Bits PU32 to PU37 should be set to 0.
2. In memory expansion mode or microprocessor mode, bits PU32 to PU37 should be set to 0 since ports P11 to P13 function as the bus control pins. However, the pull-up resistors are enabled for ports P11 to P13 when these pins function as the I/O ports with 8-/16-bit bus or multiplexed bus format.

**Figure 26.27 PUR3 Register**

## Pull-Up Control Register 4 (1)

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| PUR4 | 03F4h | XXXX 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PU40 | P14_1 to P14_3 Pull-Up Control Bit | Control pull-up setting for corresponding ports<br>0: Pull-up resistor disabled<br>1: Pull-up resistor enabled | RW |
| PU41 | P14_4 to P14_6 Pull-Up Control Bit | | RW |
| PU42 | P15_0 to P15_3 Pull-Up Control Bit | | RW |
| PU43 | P15_4 to P15_7 Pull-Up Control Bit | | RW |
| — (b7-b4) | No register bits; should be written with 0 and read as undefined value | | — |

Note:
1. This register should be set to 00h in the 100-pin package.

**Figure 26.28 PUR4 Register**

## 26.5 Port Control Register (PCR Register)

Figure 26.29 shows the PCR register.

This register selects an output mode for the port P1 between push-pull output and pseudo-N-channel open drain output. When the PCR0 bit is set to 1, the P-channel transistor in the output buffer is turned off. Note that the port P1 cannot be a perfect open drain output due to remaining parasitic diode. The absolute maximum rating of the input voltage is, therefore, from -0.3 V to VCC + 0.3 V (Refer to Figure 26.30).

In memory expansion mode or microprocessor mode, when the port P1 is used for the data bus, the PCR0 bit should be set to 0. However, when the port P1 is used as the programmable I/O port or an I/O pin for the peripheral functions, the output mode can be selected using the PCR0 bit even in these operating modes.

Port Control Register

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| PCR | 03FFh | 0XXX XXX0b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PCR0 | Port P1 Output Format Control Bit (1) | 0: Push-pull output<br>1: Pseudo-N-channel open drain output (2) | RW |
| — (b6-b1) | No register bits; should be written with 0 and read as undefined value | | — |
| PCE | Ports P9_0, P9_2, P11 to P15 Enable Bit (3) | 0: Ports P9_0, P9_2, P11 to P15 disabled<br>1: Ports P9_0, P9_2, P11 to P15 enabled | RW |

Notes:
1. In memory expansion mode or microprocessor mode, this bit should be set to 0 since the port P1 is used for the data bus. However, when it is used as the I/O port or an I/O pin for the peripheral functions, the PCR0 bit can select an output format between push-pull output and pseudo-N-channel open drain output.
2. This function is designated not to make the port P1 a full open drain but to turn off the P-channel transistor in the CMOS output buffer. Therefore, the absolute maximum rating of the input voltage is from -0.3 V to VCC + 0.3 V.
3. This bit should not be set to 1 in the 100-pin package.

**Figure 26.29 PCR Register**

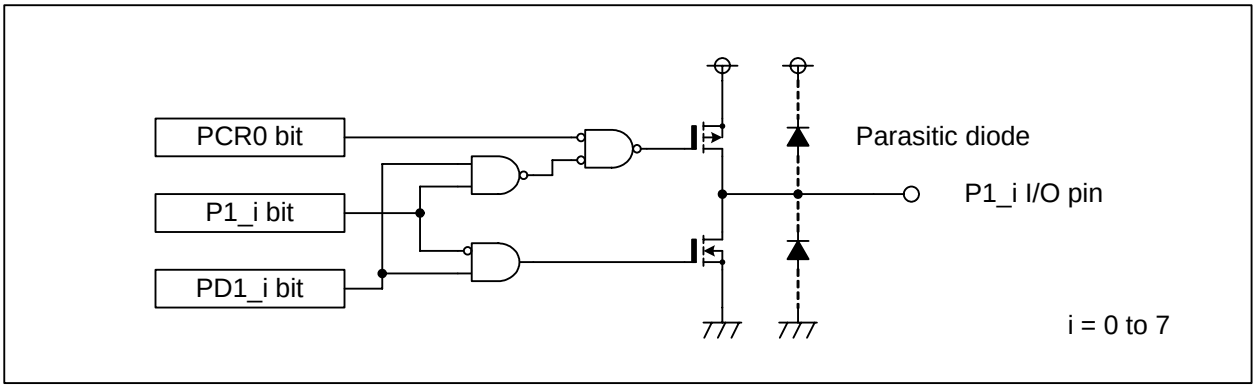



**Figure 26.30 Port P1 Output Buffer Configuration**

## 26.6 How To Configure Unused Pins

Table 26.2, Table 26.3, and Figure 26.32 show configuring examples of unused pins on the board.

**Table 26.2 Unused Pin Configuration in Single-chip Mode (1)**

| Pin Name | Setting |
|---|---|
| Ports P0 to P15 (excluding ports P8_5, and P9_1 (in the 100-pin package) or P14_1 (in the 144-pin package)) (2, 3, 4) | Configure as input ports so that each pin is connected to VSS via its own resistor (5); or configure as output ports to leave the pins open |
| P9_1 (in the 100-pin package) | Connect the pin to VSS via a resistor (5) |
| P14_1 (in the 144-pin package) | Connect the pin to VSS via a resistor (5) |
| XOUT (6) | Leave pin open |
| $\overline{\text{NMI}}$ (P8_5) | Connect the pin to VCC via a resistor (5) |
| AVCC | Connect the pin to VCC |
| AVSS, VREF | Connect the pin to VSS |
| NSD | Connect the pin to VCC via a resistor of 1 to 4.7 kΩ |

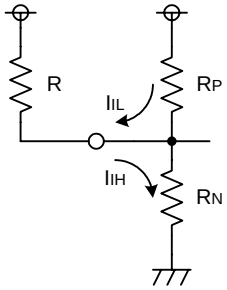
Notes:
1. Unused pins should be wired as closely as possible to the MCU (within 2 cm).
2. When configuring the pins as output ports to leave them open, note that the ports as inputs remain unchanged from when the reset is released until the mode transition is completed. During this transition, the power current may increase due to an undefined voltage level of the pins. In addition, the contents of the direction register may change because of noise or program runaway caused by the noise. To avoid these situations, reconfigure the direction register regularly by software, which may achieve the higher program reliability.
3. Ports P11 to P15 are available in the 144-pin package only.
4. In the 100-pin package, set FFh to the following addresses: 03D7h, 03DAh, 03DBh, 03DEh, and 03DFh.
5. The resistance value appropriate to the system should be designated. The range from 10 to 100 kΩ is recommended.
6. The setting is applicable when an external clock is applied to the XIN pin

**Table 26.3 Unused Pin Configuration in Memory Expansion Mode or Microprocessor Mode (1)**

| Pin Name | Setting |
|---|---|
| Ports P1, P6 to P15 (excluding ports P8_5, and P9_1 (in the 100-pin package) or P14_1 (in the 144-pin package)) (2, 3, 4) | Configure as input ports so that each pin is connected to VSS via its own resistor (5); or configure as output ports to leave the pins open |
| P9_1 (in the 100-pin package) | Connect the pin to VSS via a resistor (5) |
| P14_1 (in the 144-pin package) | Connect the pin to VSS via a resistor (5) |
| $\overline{BC0}$ to $\overline{BC3}$, $\overline{WR0}$ to $\overline{WR3}$, ALE, $\overline{HLDA}$, XOUT (6), BCLK | Leave pins open |
| $\overline{HOLD}$, $\overline{RDY}$ | Connect the pins to VCC via a resistor (5) |
| $\overline{NMI}$(P8_5) | Connect the pin to VCC via a resistor (5) |
| AVCC | Connect the pin to VCC |
| AVSS, VREF | Connect the pins to VSS |
| NSD | Connect the pin to VCC via a resistor of 1 to 4.7 kΩ |

Notes:
1. Unused pins should be wired as closely as possible to the MCU (within 2 cm).
2. In case of entering output mode to leave pins open, the ports remain input mode from when the reset is released until the ports become output mode. During this input mode, the power current may increase due to undefined voltage level of the pin. In addition to that, the contents of direction register may change because of noise which may lead to the out of control of program. Consequently, the higher program reliability may depend on the regular reconfiguration of the direction register by software.
3. Ports P11 to P15 are available in the 144-pin package only.
4. In the 100-pin package, set FFh to the following addresses: 03D7h, 03DAh, 03DBh, 03DEh, and 03DFh.
5. The resistance value appropriate to the system should be designated. The range from 10 to 100 kΩ is recommended.
6. The setting is applicable when an external clock is applied to the XIN pin.

Pull-up/pull-down resistors




The figure shows the equivalent circuit of an input pin.
The equivalent input resistors ($R_P$ and $R_N$) are calculated using input power current ($I_{IL}$ and $I_{IH}$).

(Example) When VCC = 5.0 V, $I_{IH} = I_{IL} = 5$ µA,

$$R_P = R_N = \frac{5.0}{5 \times 10^{-6}} = 1\ M\Omega$$

Since the voltage ($V_{IH}$) defined as high is more than 0.8 VCC, the resistance value R should satisfy the following expression:

$$R // R_P : R_N = 0.2 : 0.8$$

That is,

$$R = \frac{2R_PR_N}{8R_P - 2R_N}$$

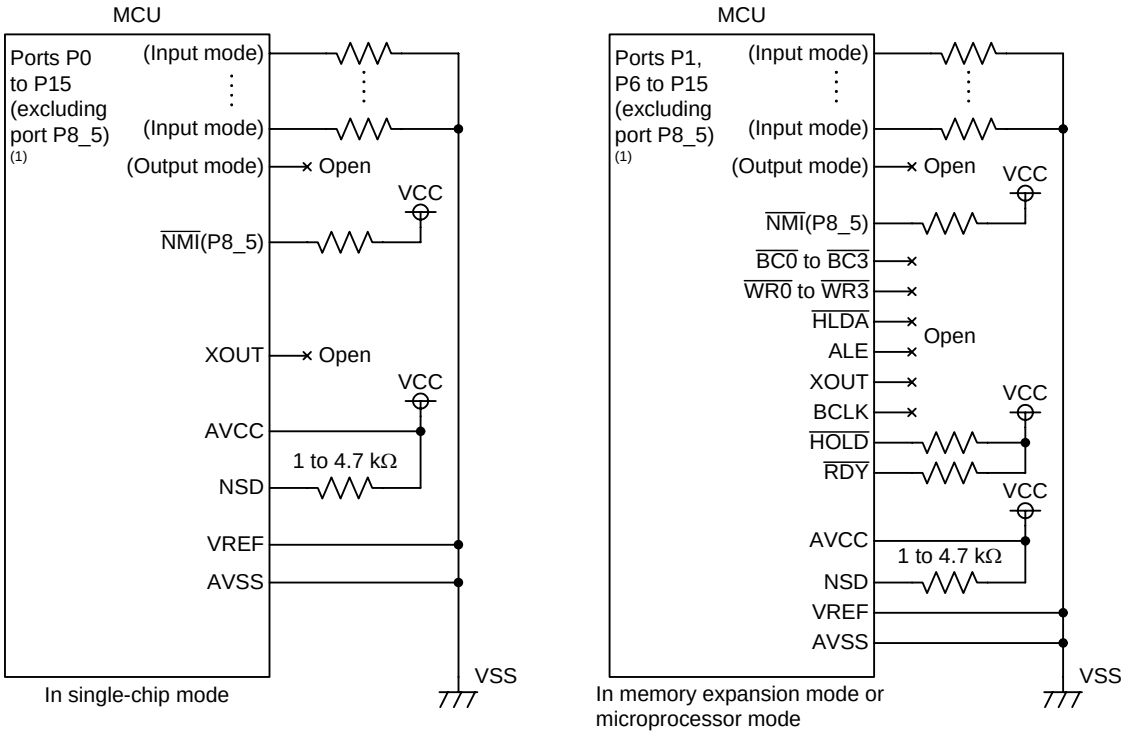
Specifically,

(Example) When VCC = 5.0 V, $I_{IH} = I_{IL} = 5$ µA,

$$R = \frac{2 \times 10^6 \times 10^6}{8 \times 10^6 - 2 \times 10^6} = 333333$$

The maximum pull-up resistor R is approx. 330 kΩ.
The actual resistance value is the calculated value with some margins.

**Figure 26.31 Pull-up/Pull-down Resistors**




Note:
1. Ports P11 to P15 are in the 144-pin package only.

**Figure 26.32 Unused Pin Configuration**

# 27. Flash Memory

## 27.1 Overview

Rewrite operation to the flash memory can be performed in the following three modes: CPU rewrite mode, standard serial I/O mode, and parallel I/O mode.
Table 27.1 lists specifications of the flash memory and Table 27.2 shows the overview of each rewrite mode.

**Table 27.1 Flash Memory Specifications**

| Item | Specification |
|---|---|
| Rewrite modes | CPU rewrite mode, standard serial I/O mode, parallel I/O mode |
| Structure | Block architecture. Refer to Figure 27.1 |
| Program operation | 8-byte basis |
| Erase operation | 1-block basis |
| Program/erase controlled by | Software commands |
| Protection types | Lock bit protect, ROM code protect, ID code protect |
| Software commands | 9 commands |

**Table 27.2 Flash Memory Rewrite Mode Overview**

| Rewrite Mode | CPU Rewrite Mode | Standard Serial I/O Mode | Parallel I/O Mode |
|---|---|---|---|
| Function | CPU executes a software command to rewrite the flash memory<br>EW0 mode:<br>Rewritable in areas other than the on-chip flash memory<br>EW1 mode:<br>Rewritable in areas other than specified blocks to be rewritten | A dedicated serial programmer rewrites the flash memory<br>Standard serial I/O mode 1:<br>Synchronous serial I/O selected<br>Standard serial I/O mode 2:<br>UART selected | A dedicated parallel programmer rewrites the flash memory |
| CPU operating mode | Single-chip mode, Memory expansion mode (EW0 mode) | Standard serial I/O mode | Parallel I/O mode |
| Programmer | — | Serial programmer | Parallel programmer |
| On-board rewriting | Supported | Supported | Not supported |

Figure 27.1 shows the on-chip flash memory structure.
The on-chip flash memory contains program area to store user programs, and data area/data flash to store the result of user programs. The program area consists of blocks 0 to 17, and data area/data flash consists of blocks A and B.
Each block can be individually protected (locked) from programing or erasing by setting the lock bit.

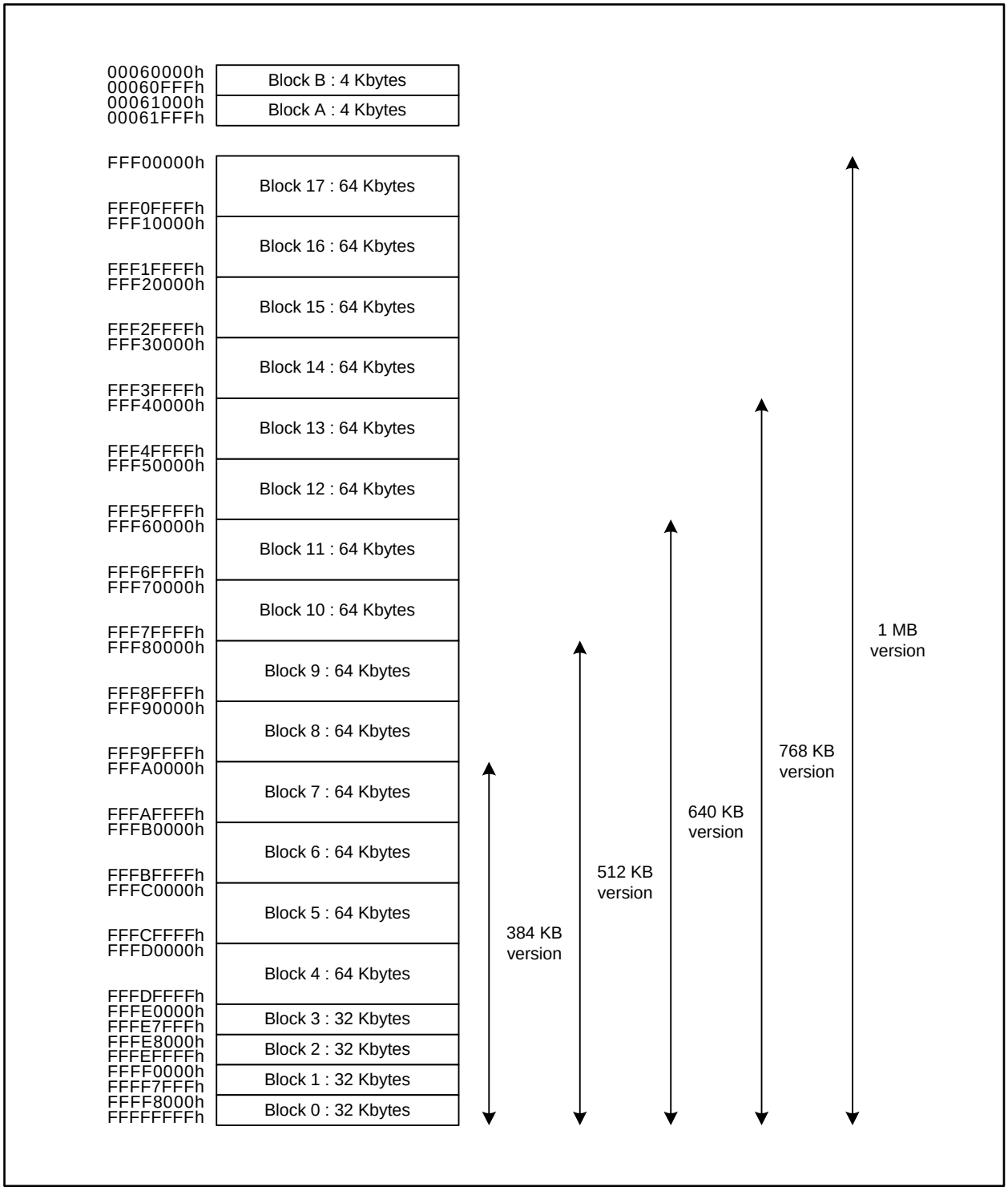



**Figure 27.1 Embedded Flash Memory Block Diagram**

## 27.2 Flash Memory Protection

There are three types of protections as shown in Table 27.3. Lock bit protection is intended to prevent accidental program or erase by program runaway. ROM code protection and ID code protection are intended to prevent read or write by a third party.

**Table 27.3 Protection Types and Characteristics**

| Protection Type | Lock Bit Protection | ROM Code Protection | ID Code Protection |
|---|---|---|---|
| Operations to be protected | Erase, write | Read, erase, write | Read, write |
| Protection available in | CPU rewrite mode<br>Standard serial I/O mode<br>Parallel I/O mode | Parallel I/O mode | Standard serial I/O mode |
| Protection available for | Individual blocks | The whole flash memory | The whole flash memory |
| Protection activated by | Setting 0 to the lock bit of block to be protected | Setting 0 to any protect bit of blocks | Writing the program which has set an ID code to specified address |
| Protection deactivated by | Setting the LBD bit in the FMR register to 1 (lock bit protection disabled). Or, by erasing the blocks whose lock bits are set to 0 to permanently diactivate the protection | Erasing all blocks whose protect bits are set to 0 by using the serial programmer | Inputting a proper ID code to the serial programmer |

### 27.2.1 Lock Bit Protection

This protection is available in all three rewrite modes. When the lock bit protection is activated, all the blocks whose lock bits are set to 0 (locked) are protected against programming and erasing.

To set the lock bit to 0, the lock bit program command must be issued.

To temporarily deactivate the protection of all protected blocks, disable the lock bit protection itself by setting the LBD bit in the FMR1 register to 1 (lock bit protection disabled). To permanently deactivate the protection of a protected block, erase the respective block to set the lock bit to 1 (unlocked).

### 27.2.2 ROM Code Protection

This protection is available only in parallel I/O mode. When the ROM code protection is activated, the whole flash memory is protected against reading and writing.

To deactivate the protection, use the serial programmer to erase all the blocks whose protect bits are set to 0 (protected).

Each block has two protect bits. Setting any protect bit to 0 by a software command activates the protection for the whole flash memory. Table 27.4 lists protect bit addresses.

**Table 27.4 Protect Bit Addresses**

| Block | Protect Bit 0 | Protect Bit 1 |
|---|---|---|
| Block B | 00060100h | 00060300h |
| Block A | 00061100h | 00061300h |
| Block 17 | FFF00100h | FFF00300h |
| Block 16 | FFF10100h | FFF10300h |
| Block 15 | FFF20100h | FFF20300h |
| Block 14 | FFF30100h | FFF30300h |
| Block 13 | FFF40100h | FFF40300h |
| Block 12 | FFF50100h | FFF50300h |
| Block 11 | FFF60100h | FFF60300h |
| Block 10 | FFF70100h | FFF70300h |
| Block 9 | FFF80100h | FFF80300h |
| Block 8 | FFF90100h | FFF90300h |
| Block 7 | FFFA0100h | FFFA0300h |
| Block 6 | FFFB0100h | FFFB0300h |
| Block 5 | FFFC0100h | FFFC0300h |
| Block 4 | FFFD0100h | FFFD0300h |
| Block 3 | FFFE0100h | FFFE0300h |
| Block 2 | FFFE8100h | FFFE8300h |
| Block 1 | FFFF0100h | FFFF0300h |
| Block 0 | FFFF8100h | FFFF8300h |

### 27.2.3 ID Code Protection

This protection is available only in standard serial I/O mode. When the ID code protection is activated, a command sent from the serial programmer is accepted only if the 7-byte ID code sent from the serial programmer is identical to the ID code programmed in the flash memory. However, if the reset vector is FFFFFFFFh, the ID code check is skipped because the flash memory is considered as “erase completed”. When the reset vector is FFFFFFFFh and the ROM code protection is activated, only the block erase command is accepted.

The ID codes sent from the serial programmer are consecutively numbered as ID1, ID2, ..., and ID7. On the other hand, the ID codes programmed in the flash memory, also numbered as ID1, ID2, ..., and ID7, are respectively assigned for addresses FFFFFE8h, FFFFFFE9h, ..., and FFFFFFEEh as shown in Figure 27.2. The ID code protection is activated when a program which has an ID code set in the corresponding address is written to the flash memory.

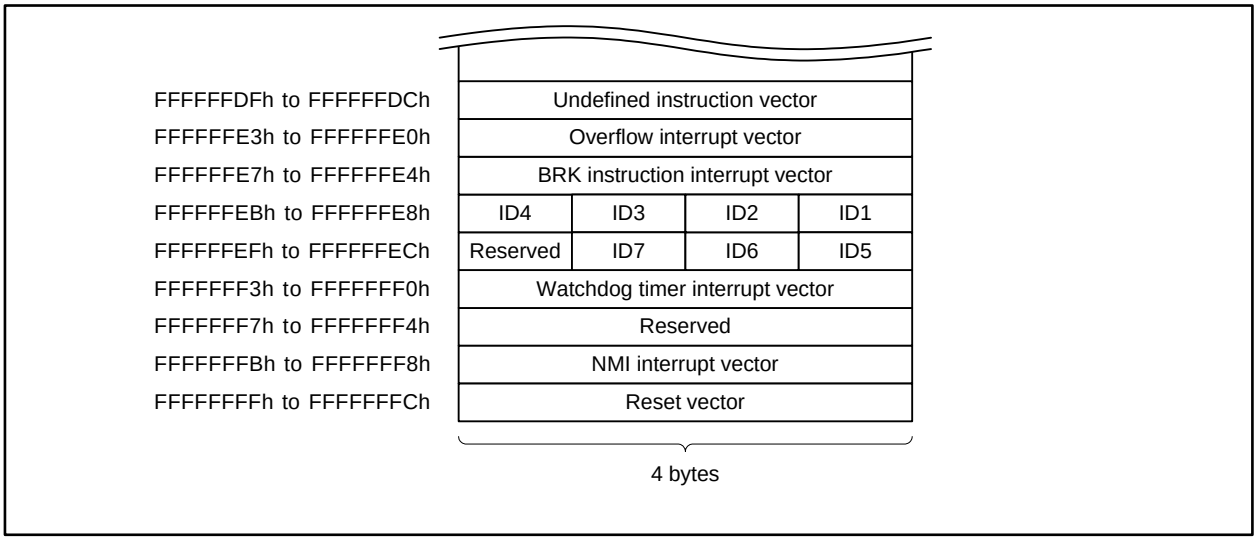



**Figure 27.2 Addresses for ID Code Stored**

## 27.3 CPU Rewrite Mode

In CPU rewrite mode, CPU executes software commands to rewrite the flash memory. The CPU accesses the flash memory not via the CPU buses but via the dedicated flash memory rewrite buses (refer to Figure 27.3).

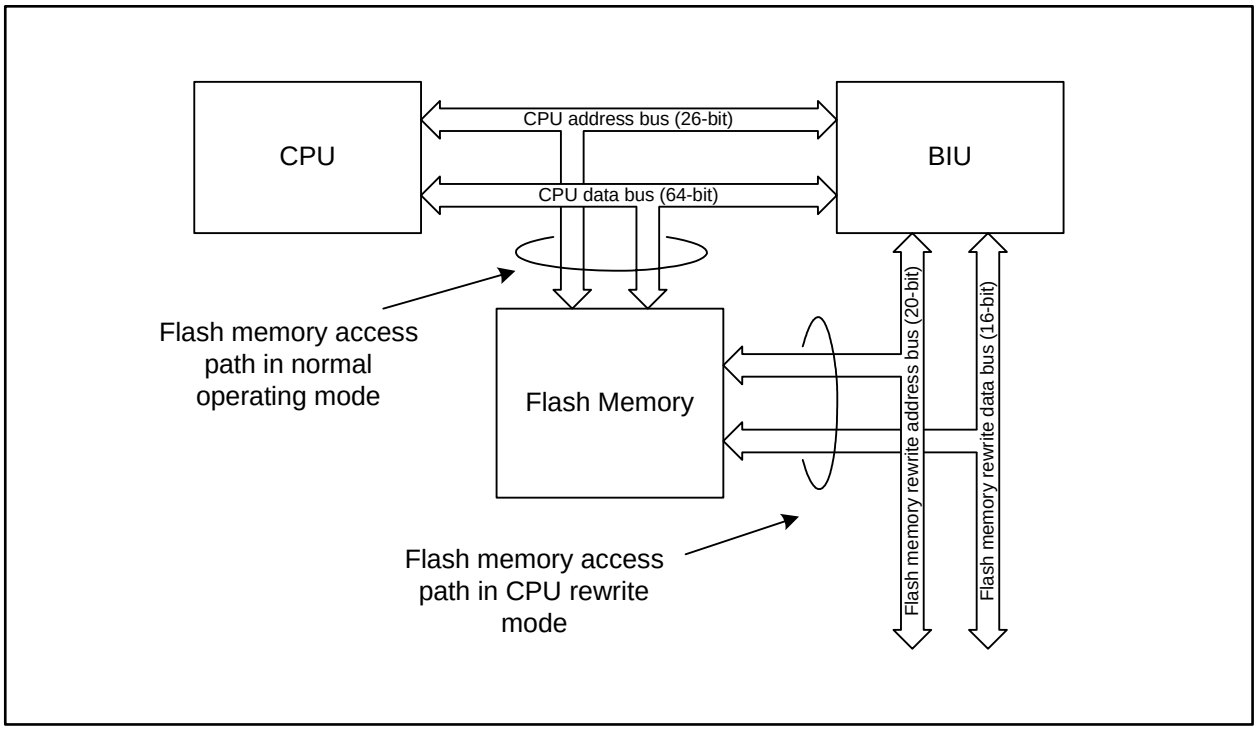



**Figure 27.3 Flash Memory Access Path in CPU Rewrite Mode**

Bus setting for flash memory rewrite should be performed by the FEBC0 and/or FEBC3 registers. Refer to 27.3.2 “Flash Memory Rewrite Bus Timing” and 28. “Electrical Characteristics” for the appropriate bus setting. Note that registers FEBC0 and FEBC3 share respective addresses with registers EBC0 and EBC3, that is, a rewrite of these registers affects the external bus setting. Set the EBC0 and/or EBC3 registers again after rewriting the FEBC0 and/or FEBC3 registers.

The CPU rewrite mode contains two sub modes: EW0 mode and EW1 mode as shown in Table 27.5.

**Table 27.5 Modes EW0 and EW1**

| Item | EW0 Mode | EW1 Mode |
|---|---|---|
| CPU operating modes | Single-chip mode<br>Memory expansion mode (1) | Single-chip mode |
| Rewrite program executable spaces | Spaces other than the on-chip flash memory | Internal spaces other than specified blocks to be rewritten, internal RAM |
| Restriction on software command | None | • Do not execute either the program or the block erase command for blocks where the rewrite control programs are written to<br>• Do not execute the read status register command<br>• Execute the enter read lock bit status mode command in RAM<br>• Execute the enter read protect bit status mode command in RAM |
| Mode after program/ erase operation | Read status register mode | Read array mode |
| CPU state during program/erase operation | Operating | In a hold state (I/O ports maintain the state before the command was executed) |
| Flash memory state detection by | • Reading the FMSR0 register by a program<br>• Executing the ready status register command to read data | • Reading the FMSR0 register by a program |
| Other restrictions | None | • Disable interrupts (except NMI) and DMA transfer during program/erase operation |

Note:
1. The $\overline{CS0}$ space and $\overline{CS3}$ space are conditionally available in memory expansion mode. Refer to 27.3.1 “CPU Operating Mode and Flash Memory Rewrite” for details.

To select CPU rewrite mode, the FEW bit in the FMCR register should be set to 1. Then, EW0 mode/EW1 mode can be selected by setting the EWM bit in the FMR0 register.
Registers FMCR and FMR0 are protected by registers PRR and FPR0, respectively.
Figure 27.4 to Figure 27.11 show associated registers.

Flash Memory Control Register (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

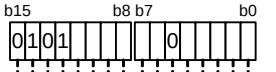
Symbol: FMCR
Address: 0006h
Reset Value: 0000 0001b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b0) | Reserved | Should be written with 1 | RW |
| — (b6-b1) | Reserved | Should be written with 0 | RW |
| FEW | CPU Rewrite Mode Setting Bit (2) | 0: Normal operating mode<br>1: CPU rewrite mode | RW |

Notes:
1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. When the MRS bit in the VRCR register is 1 (main regulator stopped), this bit should not be set to 1.

**Figure 27.4 FMCR Register**

## Flash Memory Rewrite Bus Control Register i (i = 0, 3) (1)




| Symbol | Address | Reset Value |
|---|---|---|
| FEBC0, FEBC3 | 001Dh-001Ch, 0011h-0010h | 0000h |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| FWR0 | $\overline{RD}$ Pulse Width Setting Bit | b3 b2 b1 b0<br>0 0 0 0 : *wr* = 1<br>0 0 0 1 : *wr* = 2<br>0 1 0 1 : *wr* = 3<br>0 1 1 0 : *wr* = 4<br>1 0 1 0 : *wr* = 5<br>1 0 1 1 : *wr* = 6<br>1 1 1 1 : *wr* = 7<br>Only use the combinations listed above | RW |
| FWR1 | | | RW |
| FWR2 | | | RW |
| FWR3 | | | RW |
| FWR4 | $\overline{RD}$ Pulse Width Extension Select Bit | 0: No pulse width extension<br>1: Pulse width extension selected | RW |
| — (b5) | Reserved | Should be written with 0 | RW |
| MPY0 | Multiplied Cycle Setting Bit | b7 b6<br>0 0 : Do not use this combination<br>0 1 : Do not use this combination<br>1 0 : *mpy* = 3<br>1 1 : *mpy* = 4 | RW |
| MPY1 | | | RW |
| FSUW0 | Address Setup Before $\overline{WR}$ Setting Bit | b9 b8<br>0 0 : *suw* = 0<br>0 1 : *suw* = 1<br>1 0 : *suw* = 2<br>1 1 : *suw* = 3 | RW |
| FSUW1 | | | |
| FWW0 | $\overline{WR}$ Pulse Width Setting Bit | b11 b10<br>0 0 : *ww* = 1<br>0 1 : *ww* = 2<br>1 0 : *ww* = 3<br>1 1 : *ww* = 4 | RW |
| FWW1 | | | |
| — (b12) | Reserved | Should be written with 1 | RW |
| — (b13) | Reserved | Should be written with 0 | RW |
| — (b14) | Reserved | Should be written with 1 | RW |
| — (b15) | Reserved | Should be written with 0 | RW |

Note:
1. Set the PRR register to AAh (write enabled) before rewriting this register.

**Figure 27.5 Registers FEBC0 and FEBC3**

## Flash Register Protection Unlock Register 0

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| Symbol | Address | Reset Value |
|---|---|---|
| FPR0 | 40008h | 0000 0000b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| PR0 | Protection Unlock Bit | To registers FMR0 and FMR1,<br>0: Write disabled<br>1: Write enabled | RW |
| —<br>(b7-b1) | Reserved | Should be written with 0 | RW |

**Figure 27.6 FPR0 Register**

## Flash Memory Control Register 0 (1, 2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | | 0 | | | | | |

| Symbol | Address | Reset Value |
|---|---|---|
| FMR0 | 40000h | 0X01 XX00b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| EWM | Rewrite Mode Select Bit | 0: EW0 mode<br>1: EW1 mode | RW |
| LBM | Lock Bit Read Mode Setting Bit | 0: Read via data bus (3)<br>1: Read by the LBS bit (4) | RW |
| LBS | Lock Bit Status Flag (4) | 0: Locked<br>1: Unlocked | RO |
| RRDY | Read Ready Flag | 0: Busy<br>1: Ready | RO |
| FCA | Final Command Accept Busy Flag | 0: Final command accept ready<br>1: Final command accept busy | RO |
| —<br>(b5) | Reserved | Should be written with 0 | RW |
| —<br>(b6) | Reserved | This bit is read as undefined value | RO |
| —<br>(b7) | Reserved | Should be written with 0 | RW |

Notes:
1. Set the PR0 bit in the FPR0 register to 1 (write enabled) before rewriting this register.
2. This register is reset after exiting wait mode or stop mode.
3. After entering read lock bit status mode, if any even address in the corresponding block is read, the lock bit status is indicated in the bit 6 of read data.
4. The LBS bit indicates the lock bit status by the read lock bit status command.

**Figure 27.7 FMR0 Register**

## Flash Memory Control Register 1 (1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | | 0 | | 0 |

Symbol: FMR1 Address: 40009h Reset Value: 0000 0010b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b0) | Reserved | Should be written with 0 | RW |
| RR | Reset Release Bit | 0: Reset<br>1: Reset released | RW |
| — (b2) | Reserved | Should be written with 0 | RW |
| LBD | Lock Bit Protect Disable Bit | 0: Lock bit protection enabled<br>1: Lock bit protection disabled | RW |
| — (b7-b4) | Reserved | Should be written with 0 | RW |

Note:
1. Set the PR0 bit in the FPR0 register to 1 (write enabled) before rewriting this register.

**Figure 27.8 FMR1 Register**

## Flash Memory Status Register 0

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

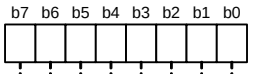
Symbol: FMSR0 Address: 40001h Reset Value: 1000 0000b

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — (b3-b0) | Reserved | These bits are read as undefined value | RO |
| WERR | Program Error Flag | 0: No program error<br>1: Program error occurred | RO |
| EERR | Erase Error Flag | 0: No erase error<br>1: Erase error occurred (1) | RO |
| — (b6) | Reserved | This bit is read as undefined value | RO |
| RDY | Ready Flag | 0: Busy<br>1: Ready | RO |

Note:
1. If an erase error has occurred, issue the clear status register command first, then re-issue the block erase command repeatedly until no more erase error occurs. After that, execute three more extra block erase operations.

**Figure 27.9 FMSR0 Register**

## Block Protect Bit Monitor Register 0 (1)



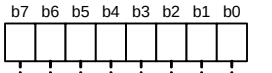

Symbol: FBPM0 | Address: 4000Ah | Reset Value: ??X? ????b (1)

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| BP0 | Block0 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP1 | Block1 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP2 | Block2 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP3 | Block3 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP4 | Block4 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| —<br>(b5) | Reserved | This bit is read as undefined value | RO |
| BP5 | Block5 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP6 | Block6 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |

Note:
1. This register is updated only once after exiting reset mode. The protect bit status at that time is applied as reset value.

**Figure 27.10 FBPM0 Register**

## Block Protect Bit Monitor Register 1 (1)




Symbol: FBPM1 | Address: 4000Bh | Reset Value: XXX? ????b (1)

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| BP7 | Block7 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP8 | Block8 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP9 | Block9 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BPB | BlockB Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BPA | BlockA Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| —<br>(b7-b5) | Reserved | These bits are read as undefined value | RO |

Note:
1. This register is updated only once after exiting reset mode. The protect bit status at that time is applied as reset value.

**Figure 27.11 FBPM1 Register**

## Block Protect Bit Monitor Register 2 (1)

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | Reset Value |
|---|---|---|
| FBPM2 | 40011h | ???? ????b (1) |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| BP10 | Block10 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP11 | Block11 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP12 | Block12 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP13 | Block13 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP14 | Block14 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP15 | Block15 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP16 | Block16 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |
| BP17 | Block17 Protect Bit Monitor Flag | 0: Protected<br>1: Protection unlocked | RO |

Note:
1. This register is updated only once after exiting reset mode. The protect bit status at that time is applied as reset value.

**Figure 27.12 FBPM2 Register**

### 27.3.1 CPU Operating Mode and Flash Memory Rewrite

To rewrite the flash memory, the bus setting using by the FEBC0 and/or FEBC3 registers is required. For exclusive use of single-chip mode, the FEBC3 register is not used. In this mode, do not change the reset value 00h of registers CB01, CB12, and CB23. The bus setting for both program area and data area can be performed by the FEBC0 register.

In other cases than the above, when the CPU operation is performed in memory expansion mode more than once, set registers CB01, CB12, and CB23 according to each setting range as shown in Table 27.6. The bus setting for program area and data area can be respectively performed by the FEBC0 register and FEBC3 register.

Note that registers FEBC0 and FEBC3 in memory expansion mode share respective addresses with registers EBC0 and EBC3, that is, when the FEBCi register (i = 0, 3) is set for the flash memory rewrite, the setting value for EBCi register is accordingly changed. This may cause external devices allocated for $\overline{CS0}$ space and/or $\overline{CS3}$ space in CPU rewrite mode to become inaccessible.

Table 27.6 lists the details of bus setting for the flash memory rewrite in each CPU operating mode.

**Table 27.6 CPU Operating Mode and Flash Memory Rewrite**

| Item | CPU Operating Mode | |
|---|---|---|
| | Single-chip mode | Memory expansion mode |
| CB01 register | Hold the reset value 00h | Setting range: 04h to F8h<br>Set an value higher than that for the CB12 register |
| CB12 register | Hold the reset value 00h | Setting range: 03h to F7h<br>Set an value higher than that for the CB23 register and lower than that for the CB01 register |
| CB23 register | Hold the reset value 00h | Setting range: 02h to F6h<br>Set an value lower than that for the CB12 register |
| Bus setting for program area | FEBC0 register | FEBC0 register |
| Bus setting for data area | FEBC0 register | FEBC3 register |
| State of $\overline{CS0}$ space and $\overline{CS3}$ space after the FEBCi register is set | N/A | • Separate bus format<br>• 16-bit bus width<br>• $\overline{RDY}$ ignored |
| Restrictions for the use of $\overline{CS0}$ space and $\overline{CS3}$ space | None | • $\overline{HOLD}$ is ignored<br>• In CPU rewrite mode, external devices become inaccessible to data with the bus format set for $\overline{CS0}$ space and/or $\overline{CS3}$ space as multiplexed bus<br>• The change of bus timing may cause external devices in $\overline{CS0}$ space and/or $\overline{CS3}$ space to become inaccessible |

### 27.3.2 Flash Memory Rewrite Bus Timing

As mentioned in 27.3.1, the bus setting for the flash memory rewrite is performed by using the FEBC0 and/or FEBC3registers. This section specifically describes the setting of registers FEBC0 and FEBC3. The reference clock is the base clock set using bits BCD1 and BCD0 in the CCR register. Time duration including tsu, tw, tc and th are specified by base clock cycles.

Table 27.7 to Table 27.9 show the correlation of read cycle and setting of following bits: MPY1, MPY0, and FWR4 to FWR0, according to respective peripheral bus clock divide ratio. Table 27.10 to Table 27.12 show the correlation of write cycle and setting of following bits: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0. Associated read/write timings are respectively illustrated in Figure 27.13 and Figure 27.14.

Read/write cycle timing is selected from these tables below to meet the timing requirements in CPU rewrite mode described in the electrical characteristics.

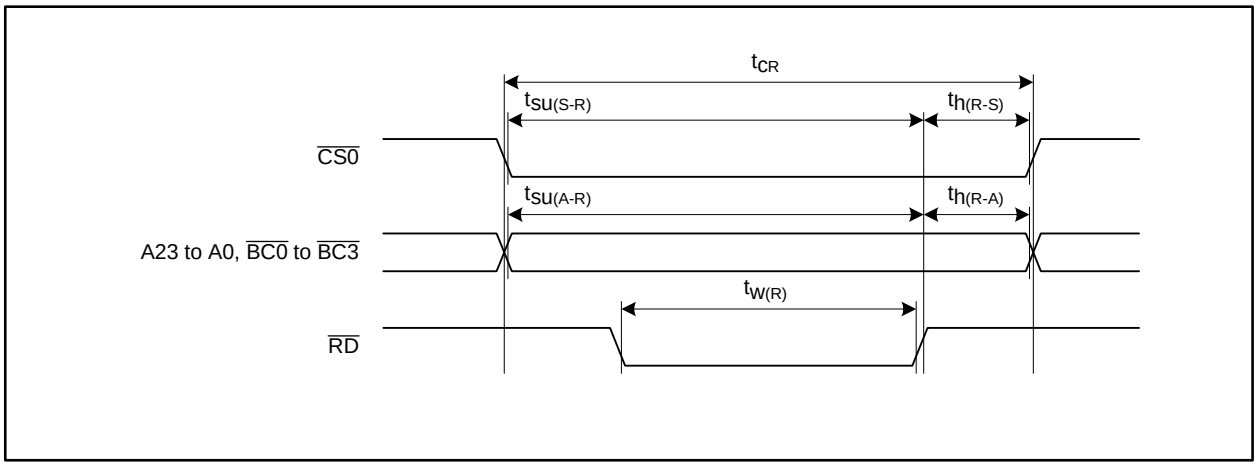



**Figure 27.13 Read Timing**

**Table 27.7 The Read Cycle and Bit Settings: MPY1, MPY0, and FWR4 to FWR0, When Peripheral Bus Clock is Divided by 2 (unit: cycles)**

| FWR3 to FWR0 Bit Settings | | FWR4 Bit Settings | MPY1 and MPY0 bit settings | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 10b | | | | 11b | | | |
| | | | *mpy* = 3 | | | | *mpy* = 4 | | | |
| | | | tsu(S-R), tsu(A-R) | tw(R) | tcR | th(R-S), th(R-A) | tsu(S-R), tsu(A-R) | tw(R) | tcR | th(R-S), th(R-A) |
| 0000b | *wr* = 1 | 0 | 4 | 3 | 4 | 0 | 6 | 5 | 6 | 0 |
| | | 1 | 6 | 5 | 6 | 0 | 6 | 5 | 6 | 0 |
| 0001b | *wr* = 2 | 0 | 8 | 7 | 8 | 0 | 10 | 9 | 10 | 0 |
| | | 1 | 8 | 7 | 8 | 0 | 10 | 9 | 10 | 0 |
| 0101b | *wr* = 3 | 0 | 10 | 9 | 10 | 0 | 14 | 13 | 14 | 0 |
| | | 1 | 12 | 11 | 12 | 0 | 14 | 13 | 14 | 0 |
| 0110b | *wr* = 4 | 0 | 14 | 13 | 14 | 0 | 18 | 17 | 18 | 0 |
| | | 1 | 14 | 13 | 14 | 0 | 18 | 17 | 18 | 0 |
| 1010b | *wr* = 5 | 0 | 16 | 15 | 16 | 0 | 22 | 21 | 22 | 0 |
| | | 1 | 18 | 17 | 18 | 0 | 22 | 21 | 22 | 0 |
| 1011b | *wr* = 6 | 0 | 20 | 19 | 20 | 0 | 26 | 25 | 26 | 0 |
| | | 1 | 20 | 19 | 20 | 0 | 26 | 25 | 26 | 0 |
| 1111b | *wr* = 7 | 0 | 22 | 21 | 22 | 0 | 30 | 29 | 30 | 0 |
| | | 1 | 24 | 23 | 24 | 0 | 30 | 29 | 30 | 0 |

**Table 27.8 The Read Cycle and Bit Settings: MPY1, MPY0, and FWR4 to FWR0, When Peripheral Bus Clock is Divided by 3 (unit: cycles)**

| FWR3 to FWR0 Bit Settings | | FWR4 Bit Settings | MPY1 and MPY0 bit settings | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 10b | | | | 11b | | | |
| | | | *mpy* = 3 | | | | *mpy* = 4 | | | |
| | | | $t_{su(S-R)}$, $t_{su(A-R)}$ | $t_{w(R)}$ | $t_{cR}$ | $t_{h(R-S)}$, $t_{h(R-A)}$ | $t_{su(S-R)}$, $t_{su(A-R)}$ | $t_{w(R)}$ | $t_{cR}$ | $t_{h(R-S)}$, $t_{h(R-A)}$ |
| 0000b | *wr* = 1 | 0 | 6 | 4.5 | 6 | 0 | 6 | 4.5 | 6 | 0 |
| | | 1 | 6 | 4.5 | 6 | 0 | 6 | 4.5 | 6 | 0 |
| 0001b | *wr* = 2 | 0 | 9 | 7.5 | 9 | 0 | 9 | 7.5 | 9 | 0 |
| | | 1 | 9 | 7.5 | 9 | 0 | 12 | 10.5 | 12 | 0 |
| 0101b | *wr* = 3 | 0 | 12 | 10.5 | 12 | 0 | 15 | 13.5 | 15 | 0 |
| | | 1 | 12 | 10.5 | 12 | 0 | 15 | 13.5 | 15 | 0 |
| 0110b | *wr* = 4 | 0 | 15 | 13.5 | 15 | 0 | 18 | 16.5 | 18 | 0 |
| | | 1 | 15 | 13.5 | 15 | 0 | 18 | 16.5 | 18 | 0 |
| 1010b | *wr* = 5 | 0 | 18 | 16.5 | 18 | 0 | 21 | 19.5 | 21 | 0 |
| | | 1 | 18 | 16.5 | 18 | 0 | 24 | 22.5 | 24 | 0 |
| 1011b | *wr* = 6 | 0 | 21 | 19.5 | 21 | 0 | 27 | 25.5 | 27 | 0 |
| | | 1 | 21 | 19.5 | 21 | 0 | 27 | 25.5 | 27 | 0 |
| 1111b | *wr* = 7 | 0 | 24 | 22.5 | 24 | 0 | 30 | 28.5 | 30 | 0 |
| | | 1 | 24 | 22.5 | 24 | 0 | 30 | 28.5 | 30 | 0 |

**Table 27.9 The Read Cycle and Bit Settings: MPY1, MPY0, and FWR4 to FWR0, When Peripheral Bus Clock is Divided by 4 (unit: cycles)**

| FWR3 to FWR0 Bit Settings | | FWR4 Bit Settings | MPY1 and MPY0 bit settings | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 10b | | | | 11b | | | |
| | | | *mpy* = 3 | | | | *mpy* = 4 | | | |
| | | | $t_{su(S-R)}$, $t_{su(A-R)}$ | $t_{w(R)}$ | $t_{cR}$ | $t_{h(R-S)}$, $t_{h(R-A)}$ | $t_{su(S-R)}$, $t_{su(A-R)}$ | $t_{w(R)}$ | $t_{cR}$ | $t_{h(R-S)}$, $t_{h(R-A)}$ |
| 0000b | *wr* = 1 | 0 | 4 | 2 | 4 | 0 | 8 | 6 | 8 | 0 |
| | | 1 | 8 | 6 | 8 | 0 | 8 | 6 | 8 | 0 |
| 0001b | *wr* = 2 | 0 | 8 | 6 | 8 | 0 | 12 | 10 | 12 | 0 |
| | | 1 | 8 | 6 | 8 | 0 | 12 | 10 | 12 | 0 |
| 0101b | *wr* = 3 | 0 | 12 | 10 | 12 | 0 | 16 | 14 | 16 | 0 |
| | | 1 | 12 | 10 | 12 | 0 | 16 | 14 | 16 | 0 |
| 0110b | *wr* = 4 | 0 | 16 | 14 | 16 | 0 | 20 | 18 | 20 | 0 |
| | | 1 | 16 | 14 | 16 | 0 | 20 | 18 | 20 | 0 |
| 1010b | *wr* = 5 | 0 | 16 | 14 | 16 | 0 | 24 | 22 | 24 | 0 |
| | | 1 | 20 | 18 | 20 | 0 | 24 | 22 | 24 | 0 |
| 1011b | *wr* = 6 | 0 | 20 | 18 | 20 | 0 | 28 | 26 | 28 | 0 |
| | | 1 | 20 | 18 | 20 | 0 | 28 | 26 | 28 | 0 |
| 1111b | *wr* = 7 | 0 | 24 | 22 | 24 | 0 | 32 | 30 | 32 | 0 |
| | | 1 | 24 | 22 | 24 | 0 | 32 | 30 | 32 | 0 |

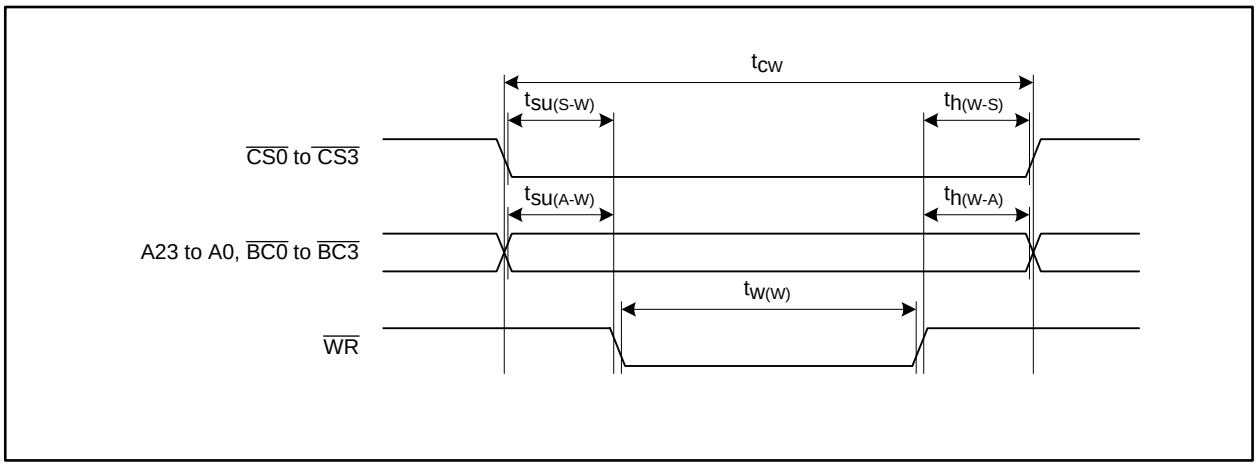



**Figure 27.14 Write Timing**

**Table 27.10 The Write Cycle and Bit Settings: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0, When Peripheral Bus Clock is Divided by 2 (unit: cycles)**

| FSUW1 and FSUW0 Bit Settings | | FWW1and FWW0 Bit Settings | | MPY1 and MPY0 Bit Settings | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 10b | | | | 11b | | | |
| | | | | mpy = 3 | | | | *mpy* = 4 | | | |
| | | | | tsu(S-W), tsu(A-W) | tw(W) | tcw | th(W-S), th(W-A) | tsu(S-W), tsu(A-W) | tw(W) | tcw | th(W-S), th(W-A) |
| 00b | *suw* = 0 | 00b | *ww* = 1 | 1 | 3 | 6 | 2 | 1 | 4 | 6 | 1 |
| | | 01b | *ww* = 2 | 1 | 6 | 8 | 1 | 1 | 8 | 10 | 1 |
| | | 10b | *ww* = 3 | 1 | 9 | 12 | 2 | 1 | 12 | 14 | 1 |
| | | 11b | *ww* = 4 | 1 | 12 | 14 | 1 | 1 | 16 | 18 | 1 |
| 01b | *suw* = 1 | 00b | *ww* = 1 | 4 | 3 | 8 | 1 | 5 | 4 | 10 | 1 |
| | | 01b | *ww* = 2 | 4 | 6 | 12 | 2 | 5 | 8 | 14 | 1 |
| | | 10b | *ww* = 3 | 4 | 9 | 14 | 1 | 5 | 12 | 18 | 1 |
| | | 11b | *ww* = 4 | 4 | 12 | 18 | 2 | 5 | 16 | 22 | 1 |
| 10b | *suw* = 2 | 00b | *ww* = 1 | 7 | 3 | 12 | 2 | 9 | 4 | 14 | 1 |
| | | 01b | *ww* = 2 | 7 | 6 | 14 | 1 | 9 | 8 | 18 | 1 |
| | | 10b | *ww* = 3 | 7 | 9 | 18 | 2 | 9 | 12 | 22 | 1 |
| | | 11b | *ww* = 4 | 7 | 12 | 20 | 1 | 9 | 16 | 26 | 1 |
| 11b | *suw* = 3 | 00b | *ww* = 1 | 10 | 3 | 14 | 1 | 13 | 4 | 18 | 1 |
| | | 01b | *ww* = 2 | 10 | 6 | 18 | 2 | 13 | 8 | 22 | 1 |
| | | 10b | *ww* = 3 | 10 | 9 | 20 | 1 | 13 | 12 | 26 | 1 |
| | | 11b | *ww* = 4 | 10 | 12 | 24 | 2 | 13 | 16 | 30 | 1 |

**Table 27.11 The Write Cycle and Bit Settings: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0, When Peripheral Bus Clock is Divided by 3 (unit: cycles)**

| FSUW1 and FSUW0 Bit Settings | | FWW1and FWW0 Bit Settings | | MPY1 and MPY0 Bit Settings | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 10b | | | | 11b | | | |
| | | | | mpy = 3 | | | | *mpy* = 4 | | | |
| | | | | tsu(S-W), tsu(A-W) | tw(W) | tcw | th(W-S), th(W-A) | tsu(S-W), tsu(A-W) | tw(W) | tcw | th(W-S), th(W-A) |
| 00b | *suw* = 0 | 00b | *ww* = 1 | 1 | 3 | 6 | 2 | 1 | 4 | 6 | 1 |
| | | 01b | *ww* = 2 | 1 | 6 | 9 | 2 | 1 | 8 | 12 | 3 |
| | | 10b | *ww* = 3 | 1 | 9 | 12 | 2 | 1 | 12 | 15 | 2 |
| | | 11b | *ww* = 4 | 1 | 12 | 15 | 2 | 1 | 16 | 18 | 1 |
| 01b | *suw* = 1 | 00b | *ww* = 1 | 4 | 3 | 9 | 2 | 6 | 3 | 12 | 3 |
| | | 01b | *ww* = 2 | 4 | 6 | 12 | 2 | 6 | 7 | 15 | 2 |
| | | 10b | *ww* = 3 | 4 | 9 | 15 | 2 | 6 | 11 | 18 | 1 |
| | | 11b | *ww* = 4 | 4 | 12 | 18 | 2 | 6 | 15 | 24 | 3 |
| 10b | *suw* = 2 | 00b | *ww* = 1 | 7 | 3 | 12 | 2 | 9 | 4 | 15 | 2 |
| | | 01b | *ww* = 2 | 7 | 6 | 15 | 2 | 9 | 8 | 18 | 1 |
| | | 10b | *ww* = 3 | 7 | 9 | 18 | 2 | 9 | 12 | 24 | 3 |
| | | 11b | *ww* = 4 | 7 | 12 | 21 | 2 | 9 | 16 | 27 | 2 |
| 11b | *suw* = 3 | 00b | *ww* = 1 | 10 | 3 | 15 | 2 | 13 | 4 | 18 | 1 |
| | | 01b | *ww* = 2 | 10 | 6 | 18 | 2 | 13 | 8 | 24 | 3 |
| | | 10b | *ww* = 3 | 10 | 9 | 21 | 2 | 13 | 12 | 27 | 2 |
| | | 11b | *ww* = 4 | 10 | 12 | 24 | 2 | 13 | 16 | 30 | 1 |

**Table 27.12 The Write Cycle and Bit Settings: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0, When Peripheral Bus Clock is Divided by 4 (unit: cycles)**

| FSUW1 and FSUW0 Bit Settings | | FWW1and FWW0 Bit Settings | | MPY1 and MPY0 Bit Settings | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 10b | | | | 11b | | | |
| | | | | mpy = 3 | | | | *mpy* = 4 | | | |
| | | | | tsu(S-W), tsu(A-W) | tw(W) | tcw | th(W-S), th(W-A) | tsu(S-W), tsu(A-W) | tw(W) | tcw | th(W-S), th(W-A) |
| 00b | *suw* = 0 | 00b | *ww* = 1 | 1 | 3 | 8 | 4 | 1 | 4 | 8 | 3 |
| | | 01b | *ww* = 2 | 1 | 6 | 8 | 1 | 1 | 8 | 12 | 3 |
| | | 10b | *ww* = 3 | 1 | 9 | 12 | 2 | 1 | 12 | 16 | 3 |
| | | 11b | *ww* = 4 | 1 | 12 | 16 | 3 | 1 | 16 | 20 | 3 |
| 01b | *suw* = 1 | 00b | *ww* = 1 | 4 | 3 | 8 | 1 | 5 | 4 | 12 | 3 |
| | | 01b | *ww* = 2 | 4 | 6 | 12 | 2 | 5 | 8 | 16 | 3 |
| | | 10b | *ww* = 3 | 4 | 9 | 16 | 3 | 5 | 12 | 20 | 3 |
| | | 11b | *ww* = 4 | 4 | 12 | 20 | 4 | 5 | 16 | 24 | 3 |
| 10b | *suw* = 2 | 00b | *ww* = 1 | 8 | 2 | 12 | 2 | 9 | 4 | 16 | 3 |
| | | 01b | *ww* = 2 | 8 | 5 | 16 | 3 | 9 | 8 | 20 | 3 |
| | | 10b | *ww* = 3 | 8 | 8 | 20 | 4 | 9 | 12 | 24 | 3 |
| | | 11b | *ww* = 4 | 8 | 11 | 20 | 1 | 9 | 16 | 28 | 3 |
| 11b | *suw* = 3 | 00b | *ww* = 1 | 10 | 3 | 16 | 3 | 13 | 4 | 20 | 3 |
| | | 01b | *ww* = 2 | 10 | 6 | 20 | 4 | 13 | 8 | 24 | 3 |
| | | 10b | *ww* = 3 | 10 | 9 | 20 | 1 | 13 | 12 | 28 | 3 |
| | | 11b | *ww* = 4 | 10 | 12 | 24 | 2 | 13 | 16 | 32 | 3 |

### 27.3.3 Software Commands

In CPU rewrite mode, software commands enable to rewrite or erase the flash memory. A write of command and read of data should be performed in 16-bit units.
Table 27.13 lists the software commands.

**Table 27.13 Software Commands**

| Command | First Command Cycle | | Second Command Cycle | |
|---|---|---|---|---|
| | Address | Data | Address | Data |
| Enter read array mode | FFFFF800h | 00FFh | — | — |
| Enter read status register mode (1) | FFFFF800h | 0070h | — | — |
| Clear status register | FFFFF800h | 0050h | — | — |
| Program (2) | FFFFF800h | 0043h | WA | WD |
| Block erase | FFFFF800h | 0020h | BA | 00D0h |
| Lock bit program | FFFFF800h | 0077h | BA | 00D0h |
| Read lock bit status | FFFFF800h | 0071h | BA | 00D0h |
| Enter read lock bit status mode (3) | FFFFF800h | 0071h | — | — |
| Protect bit program | FFFFF800h | 0067h | PBA | 00D0h |
| Enter read protect bit status mode (3) | FFFFF800h | 0061h | — | — |

WA: Even address to be written
WD: 16-bit data to be written
BA: An even address within a specific block
PBA: Protect bit address (Refer to Table 27.4)

Notes:
1. This command cannot be executed in EW1 mode.
2. A set of command consists of five words from the first command to the fifth. The program is performed in 64-bit (four-word) unit. The higher 29 bits of the address WA should be fixed and the lower three bits of respective commands from the second to fifth should be set to 000b, 010b, 100b, and 110b for the addresses 0h, 2h, 4h, and 6h, or 8h, Ah, Ch, and Eh.
3. This command should be executed in RAM.

### 27.3.4 Mode Transition

CPU rewrite mode supports four flash memory operating modes:

- Read array mode
- Read status register mode
- Read lock bit status mode
- Read protect bit status mode

When reading the flash memory in these modes, the content of memory, the content of status register, the state of lock bit of read block, and the state of protect bit are respectively read. The details are listed in Table 27.14 to Table 27.16.

**Table 27.14 Status Register**

| Bit | Bit Symbol | Bit Name | Definition | |
|---|---|---|---|---|
| | | | 0 | 1 |
| b15-b8 | — | Disabled bit | — | — |
| b7 | SR7 | Sequencer status | BUSY | READY |
| b6 | — | Reserved bit | — | — |
| b5 | SR5 | Erase status | Successfully completed | Error |
| b4 | SR4 | Program status | Successfully completed | Error |
| b3 | — | Reserved bit | — | — |
| b2 | — | Reserved bit | — | — |
| b1 | — | Reserved bit | — | — |
| b0 | — | Reserved bit | — | — |

**Table 27.15 Lock Bit Status**

| Bit | Bit Symbol | Bit Name | Definition | |
|---|---|---|---|---|
| | | | 0 | 1 |
| b15-b7 | — | Disabled bit | — | — |
| b6 | LBS | Lock bit status | Locked | Unlocked |
| b5-b0 | — | Disabled bit | — | — |

**Table 27.16 Protect Bit Status**

| Bit | Bit Symbol | Bit Name | Definition | |
|---|---|---|---|---|
| | | | 0 | 1 |
| b15-b7 | — | Disabled bit | — | — |
| b6 | PBS | Protect bit status | Protected | Unprotected |
| b5-b0 | — | Disabled bit | — | — |

In these operating mode, program/erase operation can be performed by software commands. After the operation is completed, the flash memory module automatically enters read array mode (in EW1 mode) or read status register mode (in EW0 mode).

### 27.3.5 How to Issue Software Commands

This section describes how to issue the software commands.
These commands should be issued while the RDY bit in the FMSR0 register is 1 (ready).

#### 27.3.5.1 Enter Read Array Mode Command

This command is executed to enter read array mode.
When 00FFh is written to address FFFFF800h, the flash memory enters read array mode. In this mode, data stored to a given address in memory can be read.
In EW1 mode, the flash memory is always in read array mode.

#### 27.3.5.2 Enter Read Status Register Mode

This command is executed to enter read status register mode.
When 0070h is written to address FFFFF800h, data of the status register is read in any address of the flash memory.
Do not execute this command in EW1 mode.

#### 27.3.5.3 Clear Status Register

This command is executed to reset the status register in the flash memory.
When 0050h is written to address FFFFF800h, bits SR5 and SR4 in the status register become 0 (successfully completed) (Refer to Table 27.14). Consequently, bits EERR and WERR in the FMSR0 register become 0 (no errors).

### 27.3.5.4 Program Command

This command is executed to program the flash memory in eight-byte (four-word) unit.
To start automatic programming (program and program-verify operation), write 0043h to address FFFFF800h, then write data to addresses 8n + 0 to 8n + 6. Verify that the FCA bit in the FMR0 register is 0 just before executing the final command.
To monitor the automatic program operation, read the RDY bit in the FMSR0 register. This bit indicates 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.
The operation result can be verified by the WERR bit in the FMSR0 register (Refer to 27.3.6 “Status Check”).
Do not write additional data to the address already programmed.

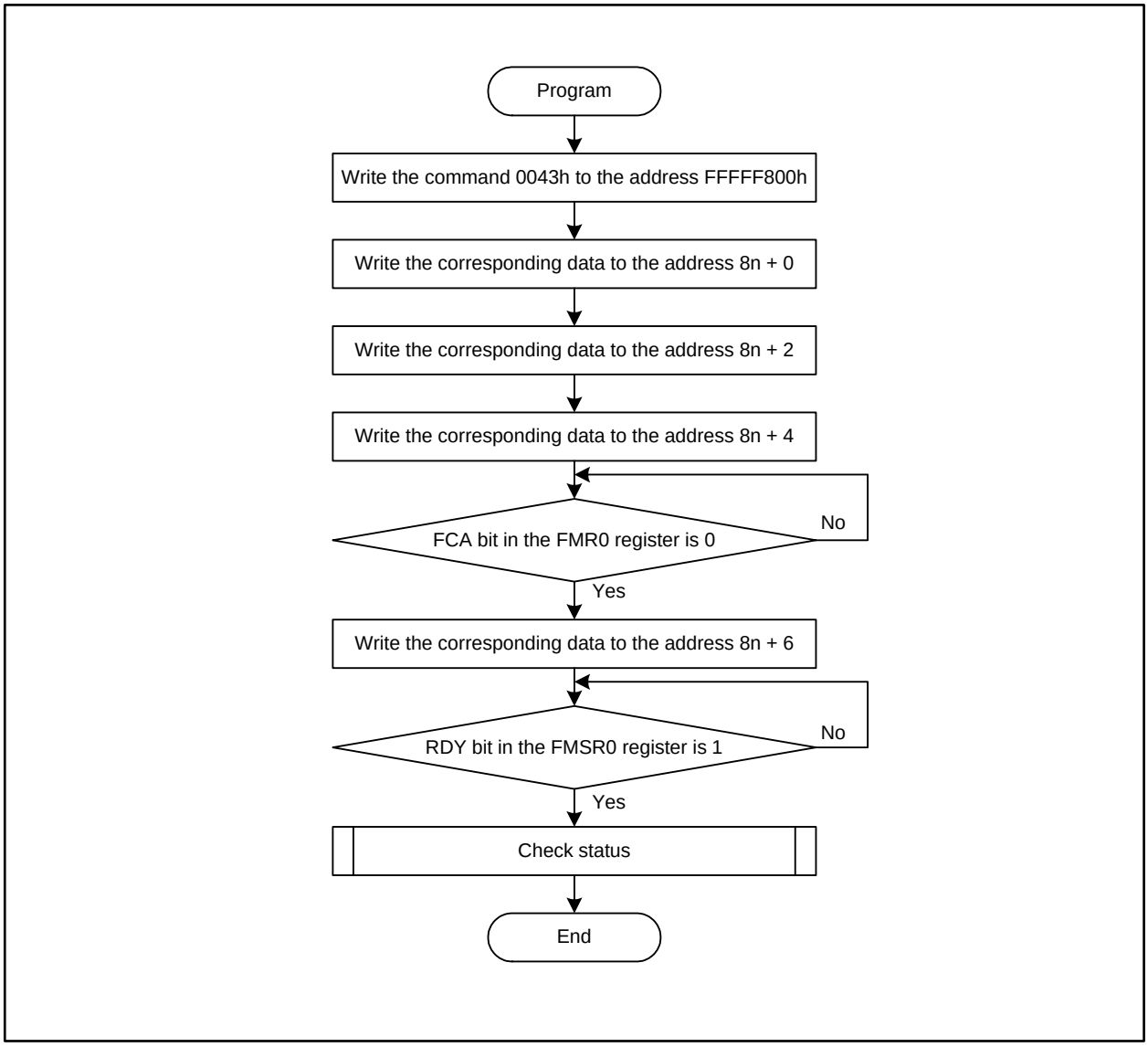



**Figure 27.15 Program Command Flow**

### 27.3.5.5 Block Erase Command

This command is executed to erase a specified block in the flash memory.

To start automatic erasing of the specified block (erase and erase-verify operation), write 0020h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, then write 00D0h to an even address of the corresponding block.

To monitor the automatic erase operation, read the RDY bit in the FMSR0 register. This bit indicates 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.

The operation result can be verified by the EERR bit in the FMSR0 register (Refer to 27.3.6 “Status Check”).

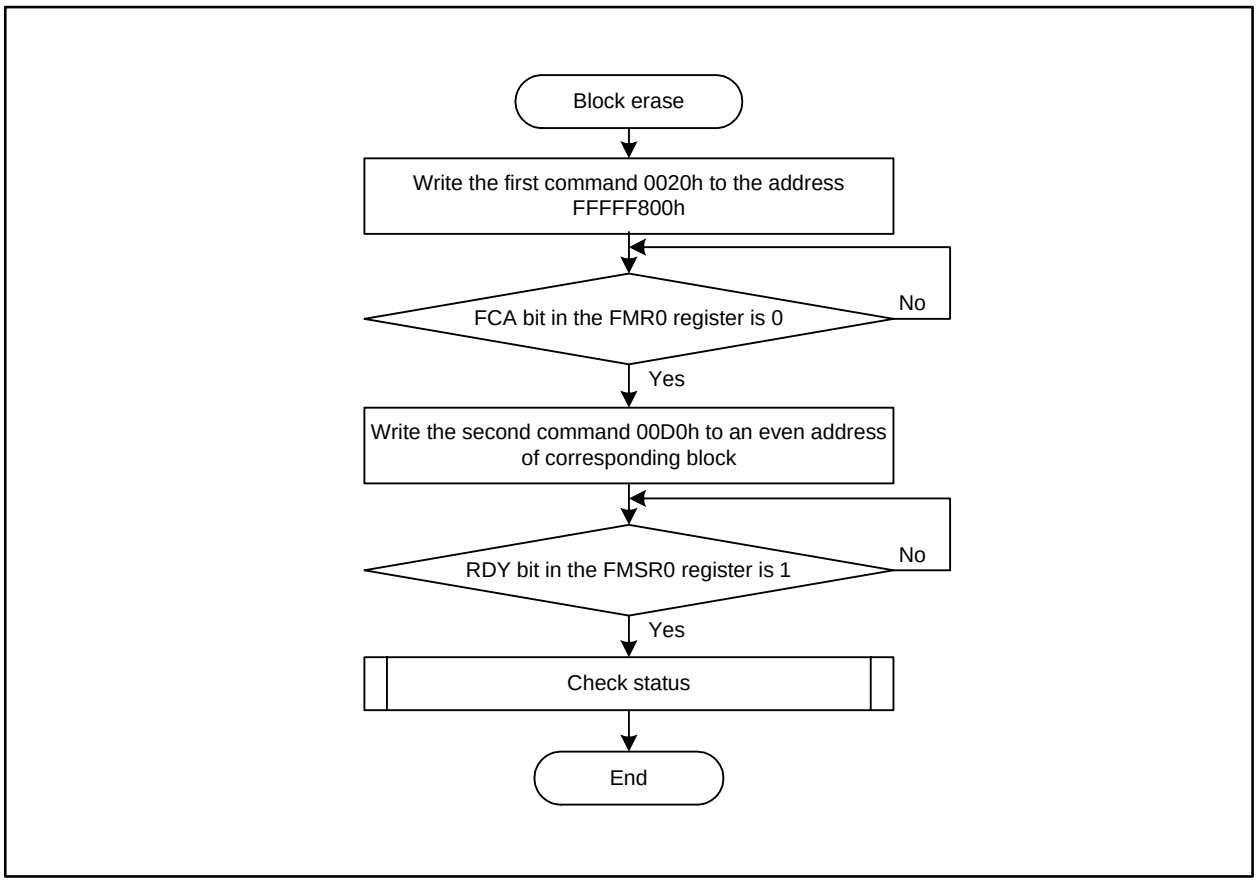



**Figure 27.16 Block Erase Command Flow**

### 27.3.5.6 Lock Bit Program Command

This command is executed to lock a specified block in the flash memory.
To lock the block, write 0077h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, then write 00D0h to an even address of the corresponding block. Then the lock bit of the block becomes 0 (locked).
To monitor the lock bit program, read the RDY bit in the FMSR0 register. This bit indicates 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.
The state of lock bit can be verified by the read lock bit status command if the LBM bit in the FMR0 register is 1 (read by the LBS bit) (Refer to 27.3.5.7 “Read Lock Bit Status Command”). If the LBM bit is 0 (read via data bus), enter read lock bit status mode (Refer to 27.3.5.8 “Enter Read Lock Bit Status Mode Command”).

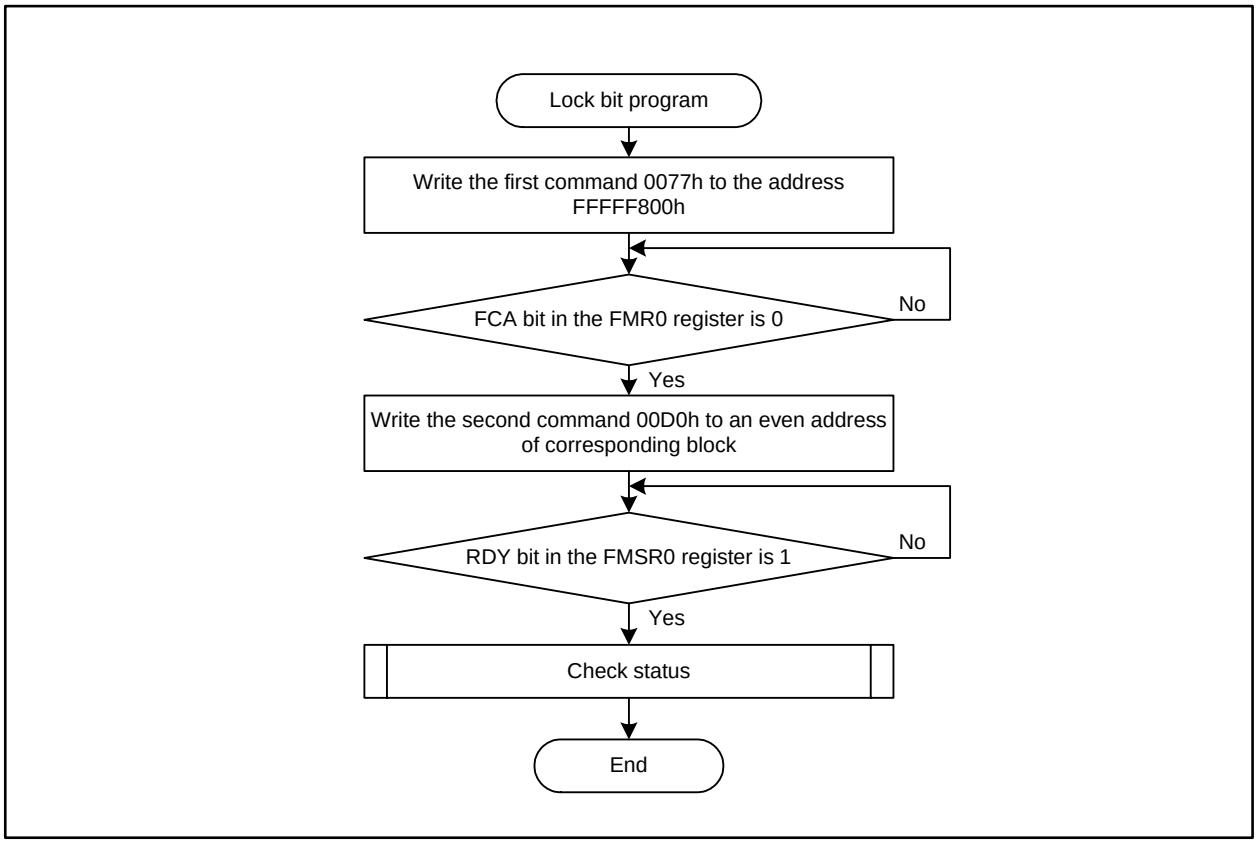



**Figure 27.17 Lock Bit Program Command Flow**

### 27.3.5.7 Read Lock Bit Status Command

This command is executed to verify if a specified block in the flash memory is locked. This command is available when the LBM bit in the FMR0 register is 1 (read by the LBS bit).
To read the LBS bit from the FMR0 register, write 0071h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, then write 00D0h to an even address of the corresponding block.
Read the LBS bit after the RDY bit in the FMSR0 register becomes 1 (ready).

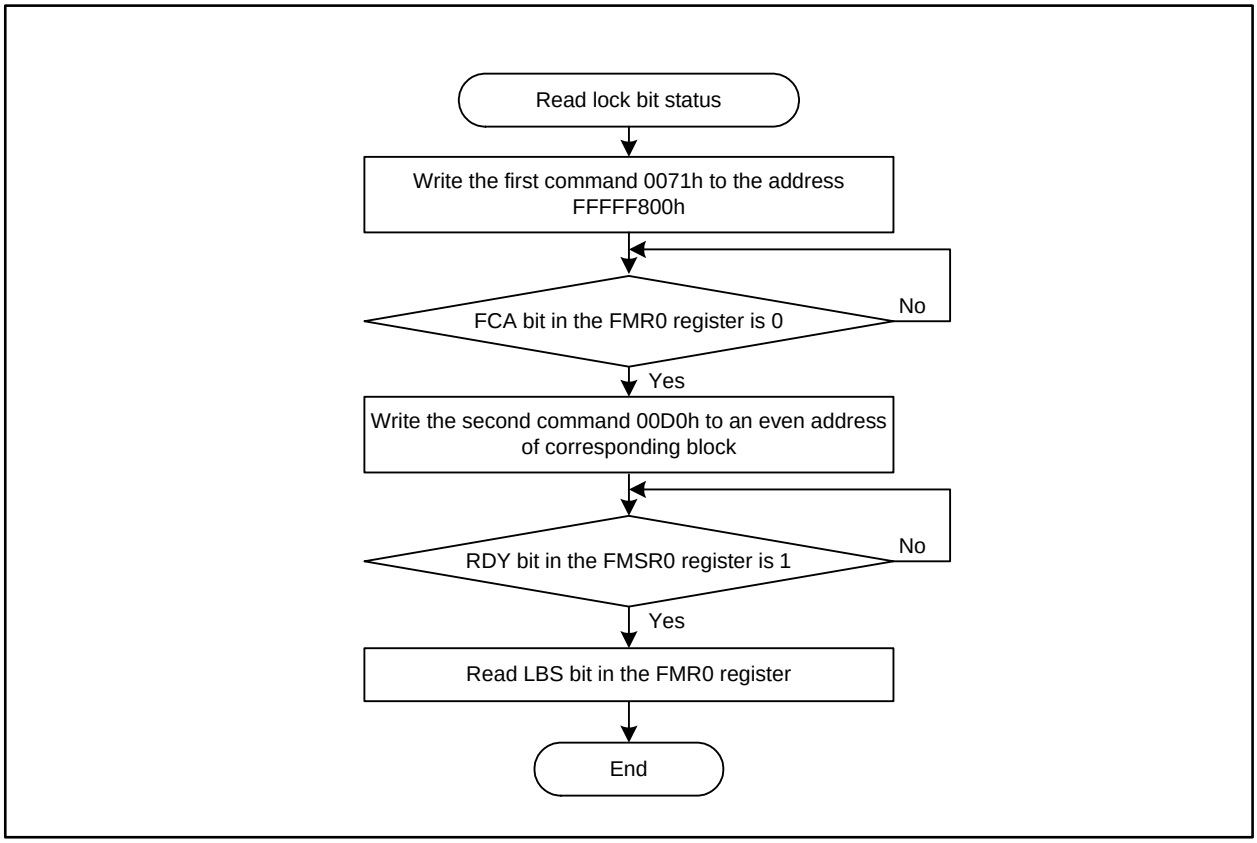



**Figure 27.18 Read Lock Bit Status Command Flow**

### 27.3.5.8 Enter Read Lock Bit Status Mode Command

This command is executed to enter read lock bit status mode. This command is enabled when the LBM bit in the FMR0 register is 0 (read via data bus).
To read the lock bit status of the read block, write 0071h to address FFFFF800h (Refer to Table 27.15).
The status is read in any address of the flash memory.
Execute this command in RAM.

### 27.3.5.9 Protect Bit Program Command

This command is executed to protect a block specified in the flash memory. ROM code protection is enabled by setting any protect bit of blocks to 0.

To program the protect bit of the designated block to 0 (protected), write 0067h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, then write 00D0h to the protect bit of the corresponding block (Refer to Table 27.4).

To monitor the protect bit program, read the RDY bit in the FMSR0 register. This bit shows 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.

To verify the state of protect bit, enter read protect bit status mode (Refer to 27.3.5.10 “Enter Read Protect Bit Status Mode Command”), then read the flash memory.

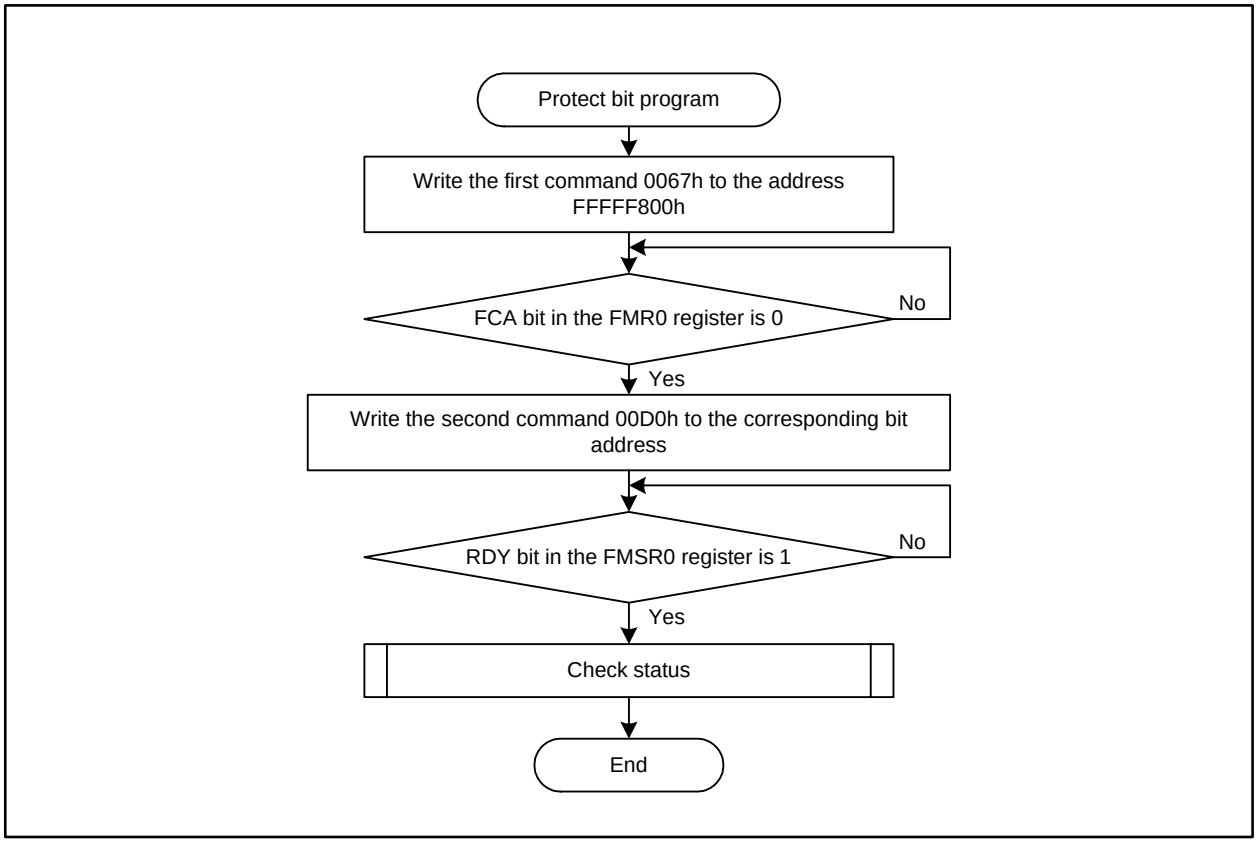



**Figure 27.19 Protect Bit Program Command Flow**

### 27.3.5.10 Enter Read Protect Bit Status Mode Command

This command is executed to enter read protect bit status mode.

To read the protect bit status of the read block, write 0061h to address FFFFF800h (Refer to Table 27.16). The status is read from any address of the flash memory.

Execute this command in RAM.

### 27.3.6 Status Check

To verify if a software command is successfully executed, read EERR or WERR bit in the FMSR0 register, or SR5 or SR4 bit in the status register.
Table 27.17 lists status and errors indicated by these bits and Figure 27.20 shows the flow of status check.

**Table 27.17 Status and Errors**

| FMSR0 register (Status register) | | Error | Causes for Error |
|---|---|---|---|
| EERR bit (SR5 bit) | WERR bit (SR4 bit) | | |
| 1 | 1 | Command sequence error | • Data other than 00D0h or 00FFh (command to cancel) was written as the last command of two commands<br>• An unavailable address was specified by an address specifying command |
| 1 | 0 | Erase error | • A locked block was tried to erase<br>• Corresponding block was not erased properly |
| 0 | 1 | Program error | • A locked block was tried to program<br>• Data was not programmed properly<br>• Lock bit was not programmed properly<br>• Protect bit was not programmed properly |
| 0 | 0 | No error | |

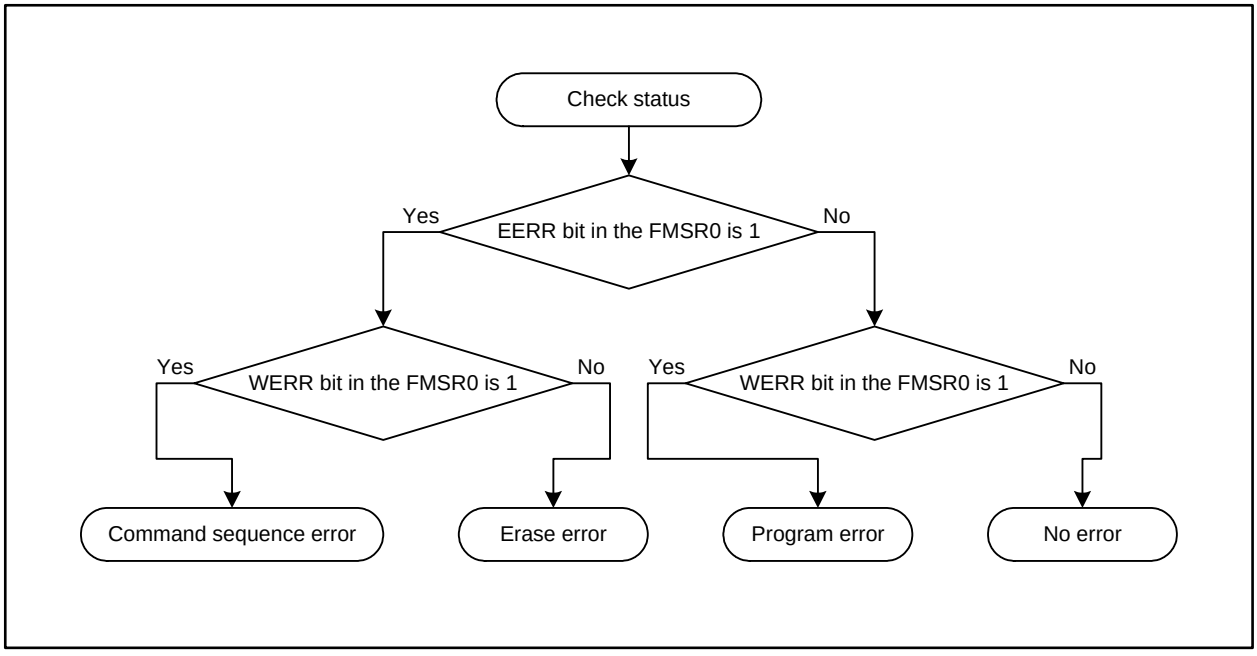



**Figure 27.20 Status Check Flow**

When an error occurred, execute clear status register command, then handle the error properly.
If erase errors or program errors occur frequently even though the program is correct, the corresponding block may be disabled.

## 27.4 Standard Serial I/O Mode

In standard serial I/O mode, the serial programmer supporting the R32C/118 Group can be used to rewrite the flash memory, while the MCU is mounted on a board.
For further information on the serial programmer, please contact your serial programmer manufacturer and refer to the user's manual included with your serial programmer for instructions.
This mode provides two types of transmit/receive mode: Standard serial I/O mode 1 which uses synchronous serial interface and standard serial I/O mode 2 which uses UART as shown in Table 27.18.

**Table 27.18 Standard Serial I/O Mode Specifications**

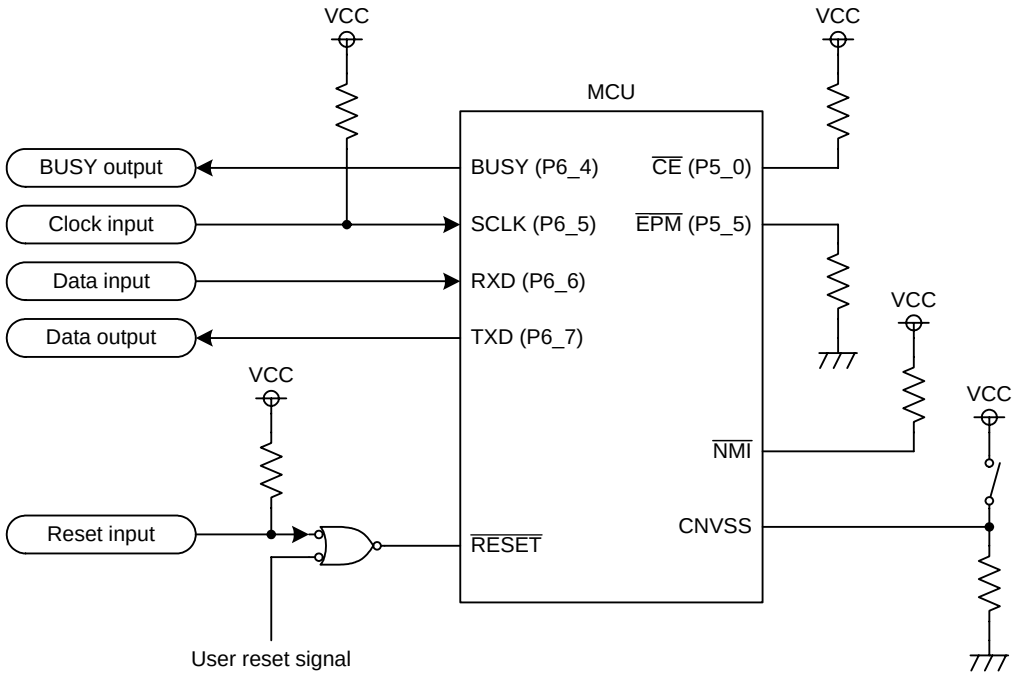
| Item | | Standard Serial I/O Mode 1 | Standard Serial I/O Mode 2 |
|---|---|---|---|
| Transmit/receive mode | | Synchronous serial I/O | UART |
| Transmit/receive bit rate | | High | Low |
| Serial interface to be used | | UART1 | UART1 |
| Pin setting | CNVSS | High | High |
| | $\overline{CE}$ (P5_0) | High | High |
| | $\overline{EPM}$ (P5_5) | Low | Low |
| | SCLK (P6_5) | In reset: Low<br>In transmission/reception: Transmit/receive clock | In reset: Low<br>In transmission/reception: Unused |
| Pin function | BUSY (P6_4) | BUSY signal | Monitor to check program operation |
| | RXD (P6_6) | Serial data input | Serial data input |
| | TXD (P6_7) | Serial data output | Serial data output |

Table 27.19 lists the pin definitions and functions in standard serial I/O mode. Figure 27.21 and Figure 27.22 show examples of a circuit application in standard serial I/O modes 1 and 2, respectively. Refer to the user's manual of your serial programmer to handle pins controlled by the serial programmer.

**Table 27.19 Pin Definitions and Functions in Standard Serial I/O Mode**

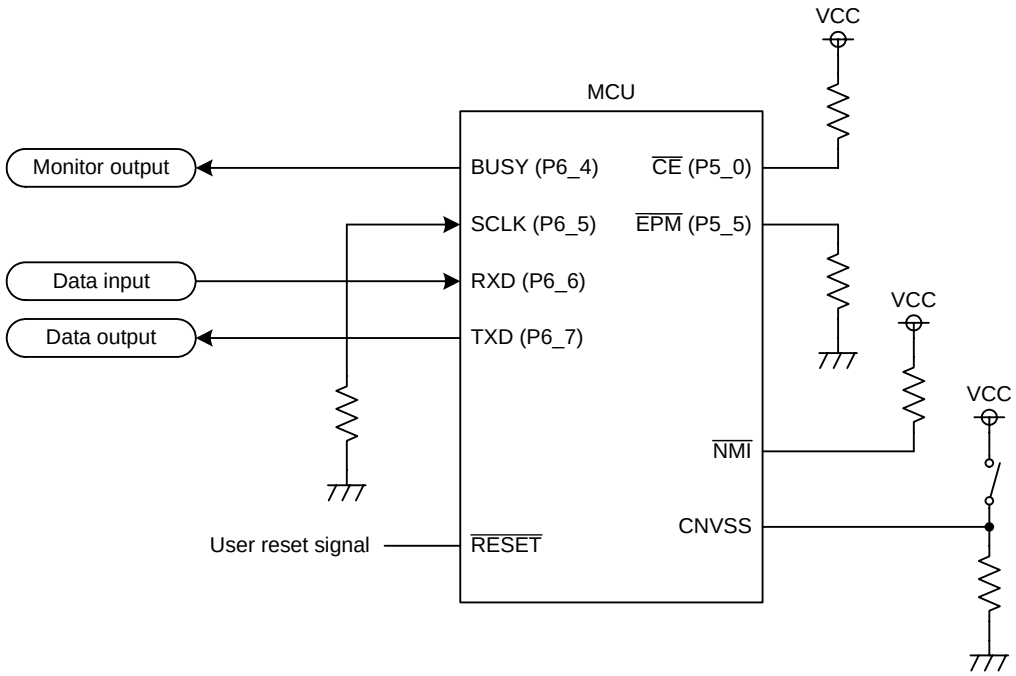
| Symbol | Function | I/O | Description |
|---|---|---|---|
| VCC, VSS | Power supply input | I | Applicable as follows: VCC = guaranteed voltage for program/ erase operation, VSS = 0 V |
| VDC1, VDC0 | Connecting pins for decoupling capacitor | — | A decoupling capacitor for internal voltage should be connected between VDC0 and VDC1 |
| CNVSS | CNVSS | I | This pin should be connected to VCC via a resistor |
| $\overline{\text{RESET}}$ | Reset input | I | Reset input pin. While the $\overline{\text{RESET}}$ pin is driven low, a clock of 20 cycles or more should be input at the XIN pin |
| XIN | Main clock input | I | A ceramic resonator or a crystal oscillator should be connected between pins XIN and XOUT. An external clock should be input at the XIN while leaving the XOUT open |
| XOUT | Main clock output | O | |
| NSD | Debug port | I/O | This pin should be connected to VCC via a resistor of 1 k to 4.7 kΩ |
| AVCC, AVSS | Analog power supply | I | AVCC and AVSS should be connected to VCC and VSS, respectively |
| VREF | Reference voltage input | I | Reference voltage input for the A/D converter and D/A converter |
| P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7 | Input port | I | High or low should be input, or the ports should be left open |
| P5_0 | $\overline{\text{CE}}$ input | I | High should be input |
| P5_1 to P5_4 | Input port | I | High or low should be input, or the ports should be left open |
| P5_5 | $\overline{\text{EPM}}$ input | I | Low should be input |
| P5_6, P5_7, P6_0 to P6_3 | Input port | I | High or low should be input, or the ports should be left open |
| P6_4 | BUSY output | O | Standard serial I/O mode 1: BUSY output pin<br>Standard serial I/O mode 2: Program operation monitor |
| P6_5 | SCLK input | I | Standard serial I/O mode 1: Serial clock input pin<br>Standard serial I/O mode 2: Low should be input |
| P6_6 | Data input RXD | I | Serial data input pin |
| P6_7 | Data output TXD | O | Serial data output pin |
| P7_0 to P7_7, P8_0 to P8_4 | Input port | I | High or low should be input, or the ports should be left open |
| P8_5 | $\overline{\text{NMI}}$ input | I | This pin should be connected to VCC via a resistor |
| P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7 (1) | Input port | I | High or low should be input, or the ports should be left open |
| P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7 (1) | Input port | I | High or low should be input, or the ports should be left open |
| P14_1, P14_3 to P14_6, P15_0 to P15_7 (1) | Input port | I | High or low should be input, or the ports should be left open |

Note:
1. Ports P9_0, P9_2, and P11 to P15 are available in the 144-pin package only.




Notes:
1. Control pins and external circuitry vary with the serial programmer. Refer to the user's manual included with the serial programmer.
2. In this example, a selector controls the voltage applied to the CNVSS pin to switch between in single-chip mode and in standard serial I/O mode 1.
3. If the user reset signal becomes low while the MCU is communicating with the serial programmer, the connection between the user reset signal and the $\overline{\text{RESET}}$ pin by, for example, a jumper selector.

**Figure 27.21 Circuit Application in Standard Serial I/O Mode 1**




Notes:
1. Control pins and external circuitry vary with the serial programmer. Refer to the user's manual included with the serial programmer.
2. In this example, a selector controls the voltage applied to the CNVSS pin to switch between in single-chip mode and in standard serial I/O mode 2.
3. If the user reset signal becomes low while the MCU is communicating with the serial programmer, the connection between the user reset signal and the $\overline{\text{RESET}}$ pin by, for example, a jumper selector.

**Figure 27.22 Circuit Application in Standard Serial I/O Mode 2**

## 27.5 Parallel I/O mode

In parallel I/O mode, the parallel programmer supporting the R32C/118 Group can be used to rewrite the flash memory.

For further information on the parallel programmer, please contact your parallel programmer manufacturer and refer to the user's manual included with your parallel programmer for instructions.

## 27.6 Notes on Flash Memory Rewriting

### 27.6.1 Note on Power Supply

- Keep the supply voltage constant within the range specified in the electrical characteristics while a rewrite operation on flash memory is in progress. If the supply voltage becomes beyond the guaranteed value, the device cannot be guaranteed.

### 27.6.2 Note on Hardware Reset

- Do not perform a hardware reset while a rewrite operation on flash memory is in progress.

### 27.6.3 Note on Flash Memory Protection

- If an ID code written in an assigned address has an error, any read/write operation of flash memory in standard serial I/O mode is disabled.

### 27.6.4 Notes on Programming

- Do not set the FEW bit in the FMCR register to 1 (CPU rewrite mode) in low speed mode or low power mode.
- Four software commands of program, block erase, lock bit program, and protect bit program are interrupted by an NMI, a watchdog timer interrupt, an oscillator stop detection interrupt, or a low voltage detection interrupt. If any of the software commands above is interrupted, erase the corresponding block and then execute the same command again. If the block erase command is interrupted, values of lock bits and protect bits become undefined. Therefore, disable the lock bit, and then execute the block erase command again.

### 27.6.5 Notes on Interrupts

- EW0 mode
  - To use interrupts assigned to the relocatable vector table, the vector table should be addressed in RAM space.
  - If either of an NMI, a watchdog timer interrupt, an oscillator stop detection interrupt, or a low voltage detection interrupt is generated, the flash memory module automatically enters read array mode. Therefore these interrupts are enabled even during a rewrite operation. On the other hand, the rewrite operation in progress is aborted by the interrupt and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.
  - Instructions BRK, INTO, and UND, which refer to data on the flash memory, are unavailable in this mode.

- EW1 mode
  - Interrupts assigned to the relocatable vector table should not be accepted during a program/erase operation.
  - The watchdog timer interrupt should not be generated, either.
  - If either of an NMI, a watchdog timer interrupt, an oscillator stop detection interrupt, or a low voltage detection interrupt is generated, the flash memory module automatically enters read array mode. Therefore this interrupt is enabled even during a rewrite operation. On the other hand, the rewrite operation in progress is aborted by the interrupt and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the EWM bit in the FMR0 register to 1 (set as EW1 mode) and the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.

### 27.6.6 Notes on Rewrite Control Program

- EW0 mode
  - If the supply voltage lowers during the rewrite operation of blocks having the rewrite control program, the rewrite control program may not be successfully rewritten, then the rewrite operation it self may not be performed. In this case perform the rewrite operation by serial programmer or parallel programmer.

- EW1 mode
  - Do not rewrite blocks having the rewrite control program.

### 27.6.7 Notes on Number of Programming/Erasure and Software Command Execution Time

- According to the increase of program/erase operation, the four software commands: program, block erase, lock bit program, and protect bit program require more time to be executed. If the number of programming/erasure exceeds the minimum endurance value specified in the electrical characteristics, it may take unpredictable time to execute the software commands. The waiting time for the execution of software commands should be set much longer than the execution time specified in the electrical characteristics.

### 27.6.8 Other Notes

- The required time to perform program/erase operation specified in the electrical characteristics can be guaranteed within the minimum values of programming/erasure endurance specified in the same table. Even if the number of programming/erasure exceeds the minimum endurance value, the program/erase operation may be unguaranteedly performed.
- Chips repeatedly programmed and erased for debugging are not allowed to be used for commercial products.

# 28. Electrical Characteristics

**Table 28.1 Absolute Maximum Ratings (1)**

| Symbol | Characteristic | | Condition | Value | Unit |
|---|---|---|---|---|---|
| $V_{CC}$ | Supply voltage | | $V_{CC} = AV_{CC}$ | -0.3 to 6.0 | V |
| $AV_{CC}$ | Analog supply voltage | | $V_{CC} = AV_{CC}$ | -0.3 to 6.0 | V |
| $V_I$ | Input voltage | XIN, $\overline{RESET}$, CNVSS, NSD, $V_{REF}$, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P5_0 to P5_3, P8_4 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_7 (2) | | -0.3 to $V_{CC}$ + 0.3 | V |
| | | P4_0 to P4_7, P5_4 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_3 | | -0.3 to 6.0 | V |
| $V_O$ | Output voltage | XOUT, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_7 (2) | | -0.3 to $V_{CC}$ + 0.3 | V |
| $P_d$ | Power consumption | | $T_a$ = 25°C | 500 | mW |
| — | Operating temperature range | | | -40 to 85 | °C |
| $T_{stg}$ | Storage temperature range | | | -65 to 150 | °C |

Notes:
1. Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
2. Ports P9_0, P9_2, and P11 to P15 are available in the 144-pin package only. Port P9_1 is designated as input pin in the 100-pin package.

**Table 28.2 Operating Conditions (1/5) (1)**

| Symbol | Characteristic | | Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $V_{CC}$ | Digital supply voltage | | 3.0 | 5.0 | 5.5 | V |
| $AV_{CC}$ | Analog supply voltage | | | $V_{CC}$ | | V |
| $V_{REF}$ | Reference voltage | | 3.0 | | $V_{CC}$ | V |
| $V_{SS}$ | Digital ground voltage | | | 0 | | V |
| $AV_{SS}$ | Analog ground voltage | | | 0 | | V |
| $dV_{CC}/dt$ | $V_{CC}$ ramp up rate ($V_{CC}$ < 2.0 V) | | 0.05 | | | V/ms |
| $V_{IH}$ | High level input voltage | XIN, $\overline{\text{RESET}}$, CNVSS, NSD, P2_0 to P2_7, P3_0 to P3_7, P5_0 to P5_3, P8_4 to P8_7 (2), P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P14_1, P14_3 to P14_6, P15_0 to P15_7 (3) | 0.8 × $V_{CC}$ | | $V_{CC}$ | V |
| | | P4_0 to P4_7, P5_4 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_3 | 0.8 × $V_{CC}$ | | 6.0 | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P12_0 to P12_7, P13_0 to P13_7 (in single-chip mode) (3) | 0.8 × $V_{CC}$ | | $V_{CC}$ | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P12_0 to P12_7, P13_0 to P13_7 (in memory expansion mode or microprocessor mode) (3) | 0.5 × $V_{CC}$ | | $V_{CC}$ | V |
| $V_{IL}$ | Low level input voltage | XIN, $\overline{\text{RESET}}$, CNVSS, NSD, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7 (2), P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P14_1, P14_3 to P14_6, P15_0 to P15_7 (3) | 0 | | 0.2 × $V_{CC}$ | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P12_0 to P12_7, P13_0 to P13_7 (in single-chip mode) (3) | 0 | | 0.2 × $V_{CC}$ | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P12_0 to P12_7, P13_0 to P13_7 (in memory expansion mode or microprocessor mode) (3) | 0 | | 0.16 × $V_{CC}$ | V |
| $T_{opr}$ | Operating temperature range | Version N | -20 | | 85 | °C |
| | | Version D | -40 | | 85 | °C |
| | | Version P | -40 | | 85 | °C |

Notes:
1. The device is operationally guaranteed under these operating conditions.
2. $V_{IH}$ and $V_{IL}$ for P8_7 are specified for P8_7 as a programmable port. These values are not applicable to P8_7 as XCIN.
3. Ports P9_0, P9_2, and P11 to P15 are available in the 144-pin package only. Port P9_1 is designated as input pin in the 100-pin package.

**Table 28.3 Operating Conditions (2/5)**
**($V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted) (1)**

| Symbol | Characteristic | | Value (2) | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $C_{VDC}$ | Decoupling capacitance for voltage regulator | Inter-pin voltage: 1.5 V | 2.4 | | 10.0 | µF |

Notes:
1. The device is operationally guaranteed under these operating conditions.
2. This value should be satisfied with due consideration of every condition as follows: operating temperature, DC bias, aging, etc.

**Table 28.4 Operating Conditions (3/5)**
**($V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0 V, and $T_a = T_{opr}$, unless otherwise noted) (1)**

| Symbol | Characteristic | | Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $I_{OH(peak)}$ | High level peak output current (2) | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_7 (3) | | | -10.0 | mA |
| $I_{OH(avg)}$ | High level average output current (4) | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_7 (3) | | | -5.0 | mA |
| $I_{OL(peak)}$ | Low level peak output current (2) | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_7 (3) | | | 10.0 | mA |
| $I_{OL(avg)}$ | Low level average output current (4) | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_7 (3) | | | 5.0 | mA |

Notes:
1. The device is operationally guaranteed under these operating conditions.
2. The following conditions should be satisfied:
   - The sum of $I_{OL(peak)}$ of ports P0, P1, P2, P8_6, P8_7, P9, P10, P11, P14, and P15 is 80 mA or less.
   - The sum of $I_{OL(peak)}$ of ports P3, P4, P5, P6, P7, P8_0 to P8_4, P12, and P13 is 80 mA or less.
   - The sum of $I_{OH(peak)}$ of ports P0, P1, P2, and P11 is -40 mA or less.
   - The sum of $I_{OH(peak)}$ of ports P8_6, P8_7, P9, P10, P14, and P15 is -40 mA or less.
   - The sum of $I_{OH(peak)}$ of ports P3, P4, P5, P12, and P13 is -40 mA or less.
   - The sum of $I_{OH(peak)}$ of ports P6, P7, and P8_0 to P8_4 is -40 mA or less.
3. Ports P9_0, P9_2, and P11 to P15 are available in the 144-pin package only. Port P9_1 is designated as input pin in the 100-pin package.
4. Average value within 100 ms.

**Table 28.5 Operating Conditions (4/5)**
**($V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted) (1)**

| Symbol | Characteristic | Value | | | Unit |
|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | |
| $f_{(XIN)}$ | Main clock oscillator frequency | 4 | | 16 | MHz |
| $f_{(XRef)}$ | Reference clock frequency | 2 | | 4 | MHz |
| $f_{(PLL)}$ | PLL clock oscillator frequency | 96 | | 128 | MHz |
| $f_{(Base)}$ | Base clock frequency | | | 50 | MHz |
| $t_{c(Base)}$ | Base clock cycle time | 20 | | | ns |
| $f_{(CPU)}$ | CPU operating frequency | | | 50 | MHz |
| $t_{c(CPU)}$ | CPU clock cycle time | 20 | | | ns |
| $f_{(BCLK)}$ | Peripheral bus clock operating frequency | | | 25 | MHz |
| $t_{c(BCLK)}$ | Peripheral bus clock cycle time | 40 | | | ns |
| $f_{(PER)}$ | Peripheral clock source frequency | | | 32 | MHz |
| $f_{(XCIN)}$ | Sub clock oscillator frequency | | 32.768 | 62.5 | kHz |

Note:
1. The device is operationally guaranteed under these operating conditions.

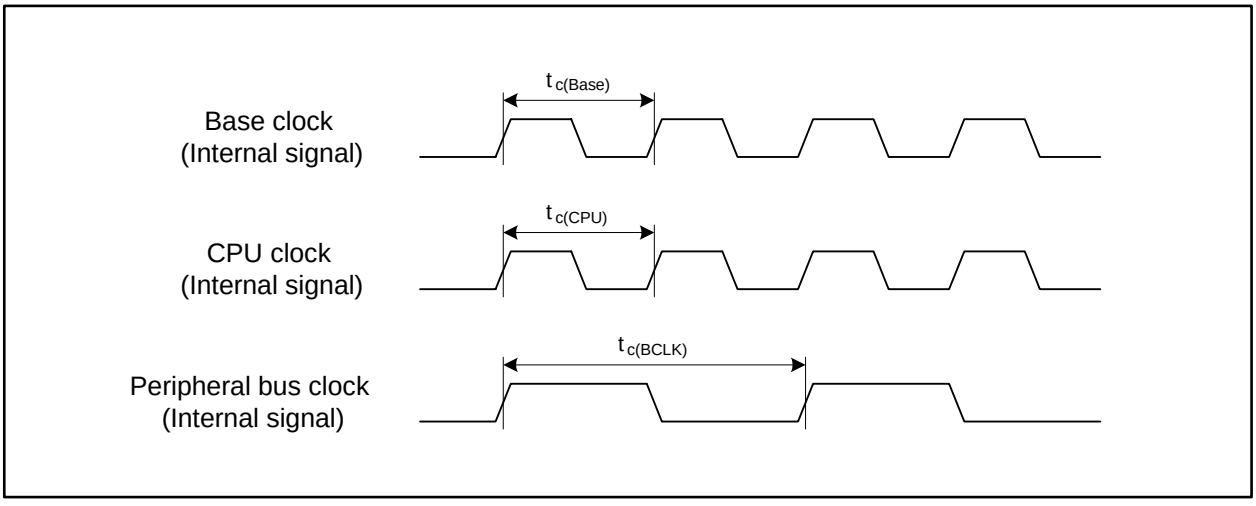



**Figure 28.1 Clock Cycle Time**

**Table 28.6 Operating Conditions (5/5)**
**($V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted) (1)**

| Symbol | Characteristic | | Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $V_{r(VCC)}$ | Allowable ripple voltage | $V_{CC}$ = 5.0 V | | | 0.5 | Vp-p |
| | | $V_{CC}$ = 3.0 V | | | 0.3 | Vp-p |
| $dV_{r(VCC)}/dt$ | Ripple voltage gradient | $V_{CC}$ = 5.0 V | | | ±0.3 | V/ms |
| | | $V_{CC}$ = 3.0 V | | | ±0.3 | V/ms |
| $f_{r(VCC)}$ | Allowable ripple frequency | | | | 10 | kHz |

Note:
1. The device is operationally guaranteed under these operating conditions.

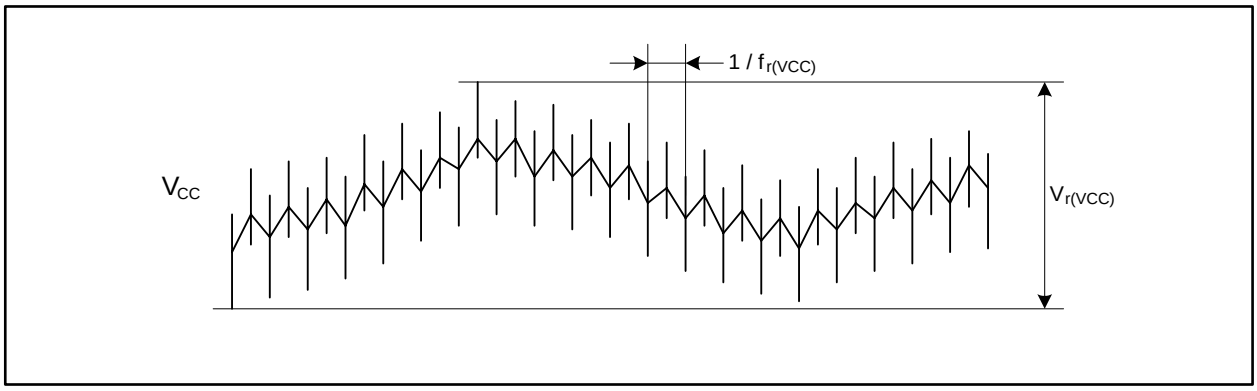



**Figure 28.2 Ripple Waveform**

**Table 28.7 RAM Electrical Characteristics**
**($V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

| Symbol | Characteristic | Measurement condition | Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $V_{RDR}$ | RAM data retention voltage | in stop mode | 2.0 | | | V |

**Table 28.8 Flash Memory Electrical Characteristics**
**($V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

| Symbol | Characteristic | | Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| — | Programming and erasure endurance of flash memory (1) | Program area | 1000 | | | times |
| | | Data area | 10000 | | | times |
| — | 4-word program time | Program area | | 150 | 900 | µs |
| | | Data area | | 300 | 1700 | µs |
| — | Lock bit-program time | Program area | | 70 | 500 | µs |
| | | Data area | | 140 | 1000 | µs |
| — | Block erasure time | 4 Kbyte block | | 0.12 | 3.0 | s |
| | | 32 Kbyte block | | 0.17 | 3.0 | s |
| | | 64 Kbyte block | | 0.20 | 3.0 | s |
| — | Data retention (2) | $T_a$ = 55°C (3) | 10 | | | years |

Notes:
1. Program/erase definition
   This value represents the number of erasures per block.
   If the flash memory is programmed/erased n times, each block can be erased n times.
   i.e. If 4-word write is performed in 512 different addresses in the block A of 4 Kbyte and then the block is erased, it is considered the programming/erasure is performed just once.
   However a write in the same address more than once for one erasure is disabled. (overwrite disabled).
2. The data retention time includes the periods when the supply voltage is not applied and no clock is provided.
3. Please contact a Renesas sales office regarding data retention time other than the above.

**Table 28.9 Power Supply Circuit Timing Characteristics ($V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0 V, and $T_a = T_{opr}$, unless otherwise noted)**

| Symbol | Characteristic | Measurement condition | Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $t_{d(P-R)}$ | Internal power supply start-up stabilization time after the main power supply is turned on | | | | 2 | ms |

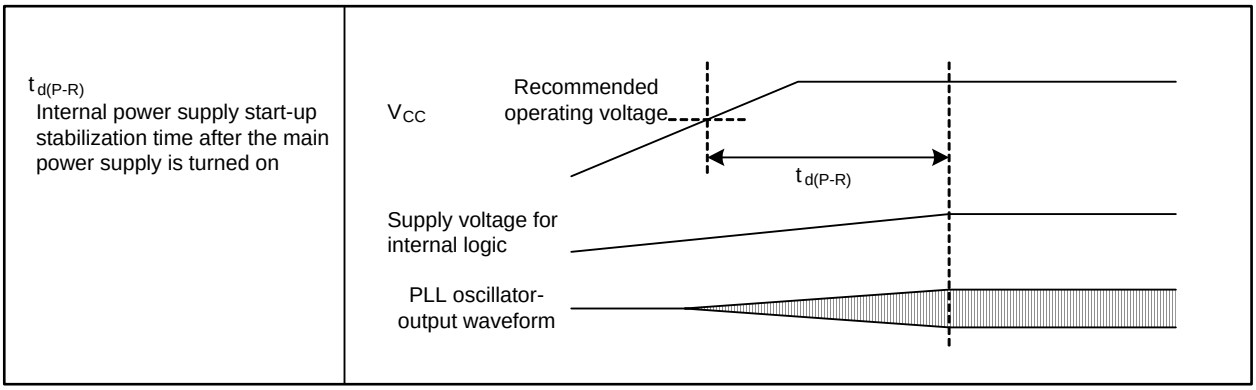



**Figure 28.3 Power Supply Circuit Timing**

**Table 28.10 Electrical Characteristics of Voltage Regulator for Internal Logic ($V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0 V, and $T_a = T_{opr}$, unless otherwise noted)**

| Symbol | Characteristics | Measurement condition | Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $V_{VDC1}$ | Output voltage | | | 1.5 | | V |

**Table 28.11 Electrical Characteristics of Low Voltage Detector ($V_{CC}$ = 4.2 to 5.5 V, $V_{SS}$ = 0 V, and $T_a = T_{opr}$, unless otherwise noted)**

| Symbol | Characteristics | Measurement condition | Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| ΔVdet | Detected voltage error | | | | ±0.3 | V |
| Vdet(R)-Vdet(F) | Hysteresis width | | 0 | | | V |
| — | Self-consuming current | $V_{CC}$ = 5.0 V, low voltage detector enabled | | 4 | | μA |
| $t_{d(E-A)}$ | Operation start time of low voltage detector | | | | 150 | μs |

**Table 28.12 Electrical Characteristics of Oscillator**
**($V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

| Symbol | Characteristics | Measurement condition | Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $f_{SO(PLL)}$ | PLL clock self-oscillation frequency | | 35 | 50 | 65 | MHz |
| $t_{LOCK(PLL)}$ | PLL lock time (1) | | | | 1 | ms |
| $t_{jitter(p-p)}$ | PLL jitter period (p-p) | | | | 2.0 | ns |
| $f_{(OCO)}$ | On-chip oscillator frequency | | 62.5 | 125 | 250 | kHz |

Note:
1. This value is applicable only when the main clock oscillation is stable.

**Table 28.13 Electrical Characteristics of Clock Circuitry**
**($V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

| Symbol | Characteristics | Measurement condition | Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $t_{rec(WAIT)}$ | Recovery time from wait mode to low power mode | | | | 225 | µs |
| $t_{rec(STOP)}$ | Recovery time from stop mode (1) | | | | 225 | µs |

Note:
1. This recovery time does not include the period until the main clock oscillator is stabilized. The CPU starts operating before the oscillator is stabilized.

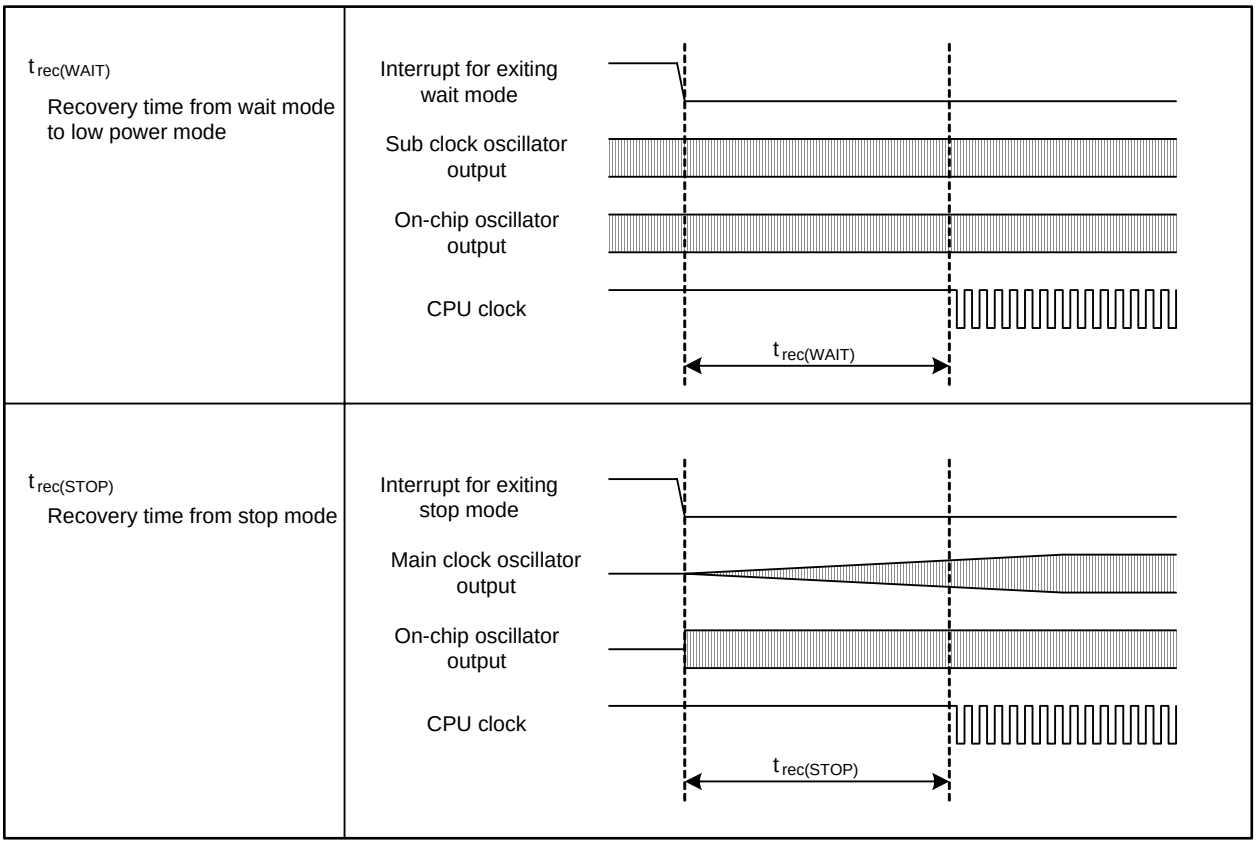



**Figure 28.4 Clock Circuit Timing**

**Timing Requirements ($V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.14 Flash Memory CPU Rewrite Mode Timing**

| Symbol | Characteristics | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{cR}$ | Read cycle time | 200 | | ns |
| $t_{su(S-R)}$ | Chip-select setup time for read | 200 | | ns |
| $t_{h(R-S)}$ | Chip-select hold time after read | 0 | | ns |
| $t_{su(A-R)}$ | Address setup time for read | 200 | | ns |
| $t_{h(R-A)}$ | Address hold time after read | 0 | | ns |
| $t_{w(R)}$ | Read pulse width | 100 | | ns |
| $t_{cW}$ | Write cycle time | 200 | | ns |
| $t_{su(S-W)}$ | Chip-select setup time for write | 0 | | ns |
| $t_{h(W-S)}$ | Chip-select hold time after write | 30 | | ns |
| $t_{su(A-W)}$ | Address setup time for write | 0 | | ns |
| $t_{h(W-A)}$ | Address hold time after write | 30 | | ns |
| $t_{w(W)}$ | Write pulse width | 50 | | ns |

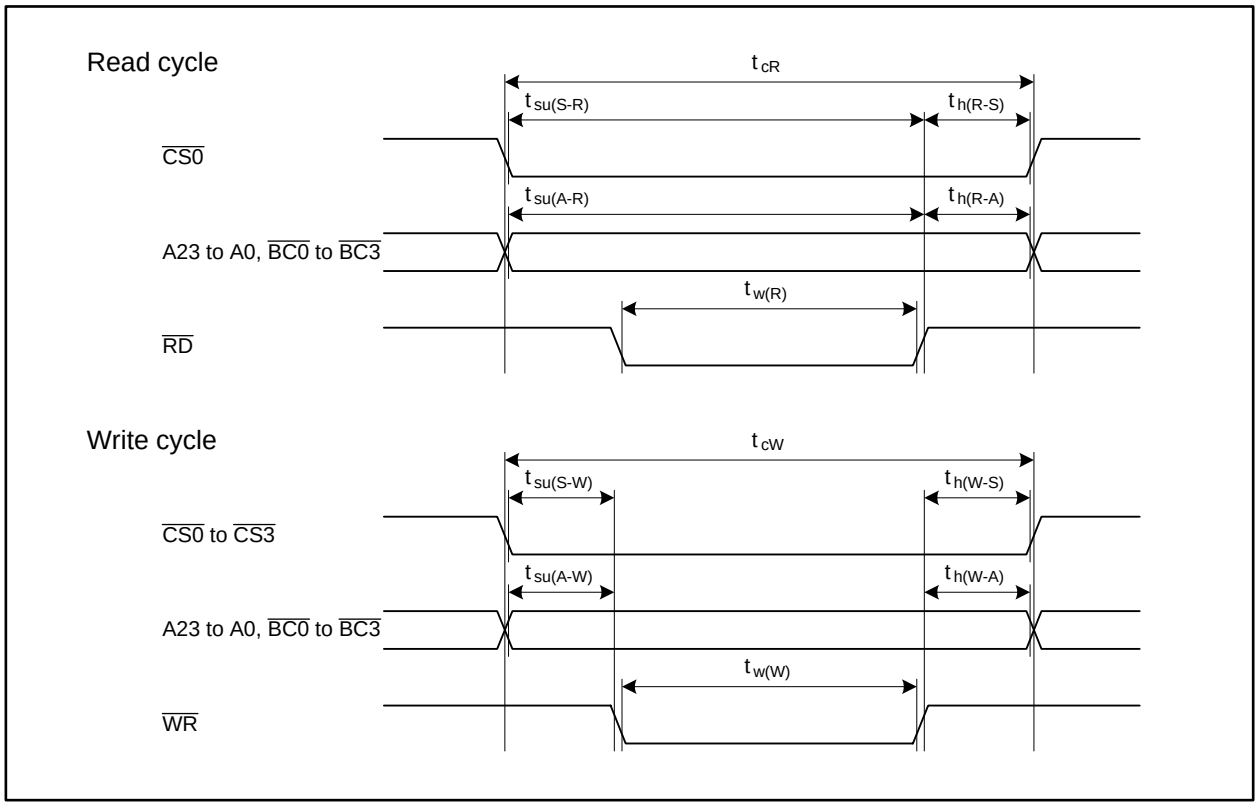



**Figure 28.5 Flash Memory CPU Rewrite Mode Timing**

$V_{CC}$ = 5 V

**Table 28.15 Electrical Characteristics (1/3)**
**($V_{CC}$ = 4.2 to 5.5 V, $V_{SS}$ = 0 V, $T_a$ = $T_{opr}$, and $f_{(CPU)}$ = 50 MHz, unless otherwise noted)**

| Symbol | Characteristic | | Measurement condition | Value | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| $V_{OH}$ | High level output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_7 (1) | $I_{OH}$ = -5 mA | $V_{CC}$ - 2.0 | | $V_{CC}$ | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_7 (1) | $I_{OH}$ = -200 µA | $V_{CC}$ - 0.3 | | $V_{CC}$ | V |
| $V_{OL}$ | Low level output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_7 (1) | $I_{OL}$ = 5 mA | | | 2.0 | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_7 (1) | $I_{OL}$ = 200 µA | | | 0.45 | V |

Note:
1. Ports P9_0, P9_2, and P11 to P15 are available in the 144-pin package only. Port P9_1 is designated as input pin in the 100-pin package.

$V_{CC}$ = 5 V

**Table 28.16 Electrical Characteristics (2/3) ($V_{CC}$ = 4.2 to 5.5 V, $V_{SS}$ = 0 V, $T_a$ = $T_{opr}$, and $f_{(CPU)}$ = 50 MHz, unless otherwise noted)**

| Symbol | Characteristic | | Measurement condition | Value | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| $V_{T+} - V_{T-}$ | Hysteresis | $\overline{HOLD}$, $\overline{RDY}$, $\overline{NMI}$, $\overline{INT0}$ to $\overline{INT8}$, $\overline{KI0}$ to $\overline{KI3}$, TA0IN to TA4IN, TA0OUT to TA4OUT, TB0IN to TB5IN, $\overline{CTS0}$ to $\overline{CTS8}$, CLK0 to CLK8, RXD0 to RXD8, SCL0 to SCL6, SDA0 to SDA6, $\overline{SS0}$ to $\overline{SS6}$, SRXD0 to SRXD6, $\overline{ADTRG}$, IIO0_0 to IIO0_7, IIO1_0 to IIO1_7, UD0A, UD0B, UD1A, UD1B, ISCLK2, ISRXD2, IEIN, CAN0IN, CAN1IN, $\overline{CAN0WU}$, $\overline{CAN1WU}$ (1) | | 0.2 | | 1.0 | V |
| | | $\overline{RESET}$ | | 0.2 | | 1.8 | V |
| $I_{IH}$ | High level input current | XIN, $\overline{RESET}$, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_7 (2) | $V_I$ = 5 V | | | 5.0 | µA |
| $I_{IL}$ | Low level input current | XIN, $\overline{RESET}$, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_7 (2) | $V_I$ = 0 V | | | -5.0 | µA |
| $R_{PULLUP}$ | Pull-up resistor | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P5_0 to P5_3, P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_7 (2) | $V_I$ = 0 V | 30 | 50 | 170 | kΩ |
| $R_{fXIN}$ | Feedback resistor | XIN | | | 1.5 | | MΩ |
| $R_{fXCIN}$ | Feedback resistor | XCIN | | | 15 | | MΩ |

Notes:
1. Pins $\overline{INT6}$ to $\overline{INT8}$ are available in the 144-pin package only.
2. Ports P9_0, P9_2, and P11 to P15 are available in the 144-pin package only. Port P9_1 is designated as input pin in the 100-pin package.

$V_{CC}$ = 5 V

**Table 28.17 Electrical Characteristics (3/3)**
**($V_{CC}$ = 4.2 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

| Symbol | Characteristic | Measurement condition | | Value | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| $I_{CC}$ | Power supply current | In single-chip mode, output pins are left open and others are connected to $V_{SS}$<br><br>XIN-XOUT<br>Drive power: low<br><br>XCIN-XCOUT<br>Drive power: low | $f_{(CPU)}$ = 50 MHz, $f_{(BCLK)}$ = 25 MHz,<br>$f_{(XIN)}$ = 8 MHz,<br>Active: XIN, PLL,<br>Stopped: XCIN, OCO | | 35 | 50 | mA |
| | | | $f_{(CPU)}$ = $f_{SO(PLL)}$/24 MHz,<br>Active: PLL (self-oscillation),<br>Stopped: XIN, XCIN, OCO | | 12 | | mA |
| | | | $f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(XIN)}$/256 MHz,<br>$f_{(XIN)}$ = 8 MHz,<br>Active: XIN,<br>Stopped: PLL, XCIN, OCO | | 1.2 | | mA |
| | | | $f_{(CPU)}$ = $f_{(BCLK)}$ = 32.768 kHz,<br>Active: XCIN,<br>Stopped: XIN, PLL, OCO,<br>Main regulator: shutdown | | 220 | | µA |
| | | | $f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(OCO)}$/4 kHz,<br>Active: OCO,<br>Stopped: XIN, PLL, XCIN,<br>Main regulator: shutdown | | 230 | | µA |
| | | | $f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(XIN)}$/256 MHz,<br>$f_{(XIN)}$ = 8 MHz,<br>Active: XIN,<br>Stopped: PLL, XCIN, OCO,<br>$T_a$ = 25°C, Wait mode | | 960 | 1600 | µA |
| | | | $f_{(CPU)}$ = $f_{(BCLK)}$ = 32.768 kHz,<br>Active: XCIN,<br>Stopped: XIN, PLL, OCO,<br>Main regulator: shutdown,<br>$T_a$ = 25°C, Wait mode | | 8 | 140 | µA |
| | | | $f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(OCO)}$/4 kHz,<br>Active: OCO,<br>Stopped: XIN, PLL, XCIN,<br>Main regulator: shutdown,<br>$T_a$ = 25°C, Wait mode | | 10 | 150 | µA |
| | | | Stopped: all clocks,<br>Main regulator: shutdown,<br>$T_a$ = 25°C | | 5 | 70 | µA |

$V_{CC}$ = 5 V

**Table 28.18 A/D Conversion Characteristics ($V_{CC}$ = $AV_{CC}$ = $V_{REF}$ = 4.2 to 5.5 V, $V_{SS}$ = $AV_{SS}$ = 0 V, $T_a$ = $T_{opr}$, and $f_{(BCLK)}$ = 25 MHz, unless otherwise noted)**

| Symbol | Characteristic | Measurement condition | | Value | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| — | Resolution | $V_{REF}$ = $V_{CC}$ | | | | 10 | Bits |
| — | Absolute error | $V_{REF}$ = $V_{CC}$ = 5 V | AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, ANEX1 (1) | | | ±3 | LSB |
| | | | External op-amp connection mode | | | ±7 | LSB |
| INL | Integral non-linearity error | $V_{REF}$ = $V_{CC}$ = 5 V | AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, ANEX1 (1) | | | ±3 | LSB |
| | | | External op-amp connection mode | | | ±7 | LSB |
| DNL | Differential non-linearity error | | | | | ±1 | LSB |
| — | Offset error | | | | | ±3 | LSB |
| — | Gain error | | | | | ±3 | LSB |
| $R_{LADDER}$ | Resistor ladder | $V_{REF}$ = $V_{CC}$ | | 4 | | 20 | kΩ |
| $t_{CONV}$ | Conversion time (10 bits) | $\phi_{AD}$ = 16 MHz, with sample and hold function | | 2.06 | | | µs |
| | | $\phi_{AD}$ = 16 MHz, without sample and hold function | | 3.69 | | | µs |
| $t_{CONV}$ | Conversion time (8 bits) | $\phi_{AD}$ = 16 MHz, with sample and hold function | | 1.75 | | | µs |
| | | $\phi_{AD}$ = 16 MHz, without sample and hold function | | 3.06 | | | µs |
| $t_{SAMP}$ | Sampling time | $\phi_{AD}$ = 16 MHz | | 0.188 | | | µs |
| $V_{IA}$ | Analog input voltage | | | 0 | | $V_{REF}$ | V |
| $\phi_{AD}$ | Operating clock frequency | without sample and hold function | | 0.25 | | 16 | MHz |
| | | with sample and hold function | | 1 | | 16 | MHz |

Note:
1. Pins AN15_0 to AN15_7 are available in the 144-pin package only.

$V_{CC}$ = 5 V

**Table 28.19 D/A Conversion Characteristics ($V_{CC}$ = $AV_{CC}$ = $V_{REF}$ = 4.2 to 5.5 V, $V_{SS}$ = $AV_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

| Symbol | Characteristic | Measurement condition | Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| — | Resolution | | | | 8 | Bits |
| — | Absolute precision | | | | 1.0 | % |
| $t_S$ | Settling time | | | | 3 | µs |
| $R_O$ | Output resistance | | 4 | 10 | 20 | kΩ |
| $I_{VREF}$ | Reference input current | (1) | | | 1.5 | mA |

Note:

1. One D/A converter is used. The DAi register (i = 0, 1) of the other unused converter is set to 00h. The resistor ladder for A/D converter is not considered.
   Even when the VCUT bit in the AD0CON1 register is set to 0 ($V_{REF}$ disconnected), $I_{VREF}$ is supplied.

$V_{CC}$ = 5 V

**Timing Requirements ($V_{CC}$ = 4.2 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.20 External Clock Input**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(X)}$ | External clock input period | 62.5 | 250 | ns |
| $t_{w(XH)}$ | External clock input high level pulse width | 25 | | ns |
| $t_{w(XL)}$ | External clock input low level pulse width | 25 | | ns |
| $t_{r(X)}$ | External clock input rise time | | 5 | ns |
| $t_{f(X)}$ | External clock input fall time | | 5 | ns |
| $t_w$ / $t_c$ | External clock input duty | 40 | 60 | % |

**Table 28.21 External Bus Timing**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{su(D-R)}$ | Data setup time for read | 40 | | ns |
| $t_{h(R-D)}$ | Data hold time after read | 0 | | ns |
| $t_{dis(R-D)}$ | Data disable time after read | | 0.5 × $t_{c(Base)}$ + 10 | ns |

$V_{CC}$ = 5 V

**Timing Requirements ($V_{CC}$ = 4.2 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.22 Timer A Input (Counting input in event counter mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TAiIN input clock period | 200 | | ns |
| $t_{w(TAH)}$ | TAiIN input high level pulse width | 80 | | ns |
| $t_{w(TAL)}$ | TAiIN input low level pulse width | 80 | | ns |

**Table 28.23 Timer A Input (Gating input in timer mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TAiIN input clock period | 400 | | ns |
| $t_{w(TAH)}$ | TAiIN input high level pulse width | 180 | | ns |
| $t_{w(TAL)}$ | TAiIN input low level pulse width | 180 | | ns |

**Table 28.24 Timer A Input (External trigger input in one-shot timer mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TAiIN input clock period | 200 | | ns |
| $t_{w(TAH)}$ | TAiIN input high level pulse width | 80 | | ns |
| $t_{w(TAL)}$ | TAiIN input low level pulse width | 80 | | ns |

**Table 28.25 Timer A Input (External trigger input in pulse-width modulation mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{w(TAH)}$ | TAiIN input high level pulse width | 80 | | ns |
| $t_{w(TAL)}$ | TAiIN input low level pulse width | 80 | | ns |

**Table 28.26 Timer A Input (Increment/decrement count switching input in event counter mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(UP)}$ | TAiOUT input clock period | 2000 | | ns |
| $t_{w(UPH)}$ | TAiOUT input high level pulse width | 1000 | | ns |
| $t_{w(UPL)}$ | TAiOUT input low level pulse width | 1000 | | ns |
| $t_{su(UP-TIN)}$ | TAiOUT input setup time | 400 | | ns |
| $t_{h(TIN-UP)}$ | TAiOUT input hold time | 400 | | ns |

$V_{CC}$ = 5 V

**Timing Requirements ($V_{CC}$ = 4.2 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.27 Timer B Input (Counting input in event counter mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input clock period (one edge counting) | 200 | | ns |
| $t_{W(TBH)}$ | TBiIN input high level pulse width (one edge counting) | 80 | | ns |
| $t_{W(TBL)}$ | TBiIN input low level pulse width (one edge counting) | 80 | | ns |
| $t_{C(TB)}$ | TBiIN input clock period (both edges counting) | 200 | | ns |
| $t_{W(TBH)}$ | TBiIN input high level pulse width (both edges counting) | 80 | | ns |
| $t_{W(TBL)}$ | TBiIN input low level pulse width (both edges counting) | 80 | | ns |

**Table 28.28 Timer B Input (Pulse period measure mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input clock period | 400 | | ns |
| $t_{W(TBH)}$ | TBiIN input high level pulse width | 180 | | ns |
| $t_{W(TBL)}$ | TBiIN input low level pulse width | 180 | | ns |

**Table 28.29 Timer B Input (Pulse-width measure mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input clock period | 400 | | ns |
| $t_{W(TBH)}$ | TBiIN input high level pulse width | 180 | | ns |
| $t_{W(TBL)}$ | TBiIN input low level pulse width | 180 | | ns |

$V_{CC}$ = 5 V

**Timing Requirements ($V_{CC}$ = 4.2 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.30 Serial Interface**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input clock period | 200 | | ns |
| $t_{W(CKH)}$ | CLKi input high level pulse width | 80 | | ns |
| $t_{W(CKL)}$ | CLKi input low level pulse width | 80 | | ns |
| $t_{su(D-C)}$ | RXDi input setup time | 80 | | ns |
| $t_{h(C-D)}$ | RXDi input hold time | 90 | | ns |

**Table 28.31 A/D Trigger Input**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{W(ADH)}$ | $\overline{ADTRG}$ input high level pulse width<br>Hardware trigger input high level pulse width | $\frac{3}{\phi_{AD}}$ | | ns |
| $t_{W(ADL)}$ | $\overline{ADTRG}$ input low level pulse width<br>Hardware trigger input high level pulse width | 125 | | ns |

**Table 28.32 External Interrupt INTi Input**

| Symbol | Characteristic | | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{W(INH)}$ | $\overline{INTi}$ input high level pulse width | Edge sensitive | 250 | | ns |
| | | Level sensitive | $t_{c(CPU)}$ + 200 | | ns |
| $t_{W(INL)}$ | $\overline{INTi}$ input low level pulse width | Edge sensitive | 250 | | ns |
| | | Level sensitive | $t_{c(CPU)}$ + 200 | | ns |

**Table 28.33 Intelligent I/O**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(ISCLK2)}$ | ISCLK2 input clock period | 600 | | ns |
| $t_{W(ISCLK2H)}$ | ISCLK2 input high level pulse width | 270 | | ns |
| $t_{W(ISCLK2L)}$ | ISCLK2 input low level pulse width | 270 | | ns |
| $t_{su(RXD-ISCLK2)}$ | ISRXD2 input setup time | 150 | | ns |
| $t_{h(ISCLK2-RXD)}$ | ISRXD2 input hold time | 100 | | ns |

$V_{CC}$ = 5 V

**Timing Requirements ($V_{CC}$ = 4.2 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.34 Multi-master I²C-bus Interface**

| Symbol | Characteristic | Value | | | | Unit |
|---|---|---|---|---|---|---|
| | | Standard-mode | | Fast-mode | | |
| | | Min. | Max. | Min. | Max. | |
| $t_{w(SCLH)}$ | MSCL input high level pulse width | 600 | | 600 | | ns |
| $t_{w(SCLL)}$ | MSCL input low level pulse width | 600 | | 600 | | ns |
| $t_{r(SCL)}$ | MSCL input rise time | | 1000 | | 300 | ns |
| $t_{f(SCL)}$ | MSCL input fall time | | 300 | | 300 | ns |
| $t_{r(SDA)}$ | MSDA input rise time | | 1000 | | 300 | ns |
| $t_{f(SDA)}$ | MSDA input fall time | | 300 | | 300 | ns |
| $t_{h(SDA-SCL)S}$ | MSCL high level hold time after start condition/restart condition | (1) | | $2 \times t_{c(\phi IIC)} + 40$ | | ns |
| $t_{su(SCL-SDA)P}$ | MSCL high level setup time for restart condition/stop condition | (1) | | $2 \times t_{c(\phi IIC)} + 40$ | | ns |
| $t_{w(SDAH)P}$ | MSDA high level pulse width after stop condition | (1) | | $4 \times t_{c(\phi IIC)} + 40$ | | ns |
| $t_{su(SDA-SCL)}$ | MSDA input setup time | 100 | | 100 | | ns |
| $t_{h(SCL-SDA)}$ | MSDA input hold time | 0 | | 0 | | ns |

Note:

1. The value is calculated by the following formulas based on a value SSC set by bits SSC4 to SSC0 in the I2CSSCR register:

   $t_{h(SDA-SCL)S} = SSC \div 2 \times t_{c(\phi IIC)} + 40$ [ns]

   $t_{su(SCL-SDA)P} = (SSC \div 2 + 1) \times t_{c(\phi IIC)} + 40$ [ns]

   $t_{w(SDAH)P} = (SSC + 1) \times t_{c(\phi IIC)} + 40$ [ns]

$V_{CC}$ = 5 V

**Switching Characteristics ($V_{CC}$ = 4.2 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.35 External Bus Timing (Separate bus)**

| Symbol | Characteristic | Measurement condition | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{su(S-R)}$ | Chip-select setup time for read | Refer to Figure 28.6 | (1) | | ns |
| $t_{h(R-S)}$ | Chip-select hold time after read | | $t_{c(Base)}$ - 15 | | ns |
| $t_{su(A-R)}$ | Address setup time for read | | (1) | | ns |
| $t_{h(R-A)}$ | Address hold time after read | | $t_{c(Base)}$ - 15 | | ns |
| $t_{W(R)}$ | Read pulse width | | (1) | | ns |
| $t_{su(S-W)}$ | Chip-select setup time for write | | (1) | | ns |
| $t_{h(W-S)}$ | Chip-select hold time after write | | 1.5 × $t_{c(Base)}$ - 15 | | ns |
| $t_{su(A-W)}$ | Address setup time for write | | (1) | | ns |
| $t_{h(W-A)}$ | Address hold time after write | | 1.5 × $t_{c(Base)}$ - 15 | | ns |
| $t_{W(W)}$ | Write pulse width | | (1) | | ns |
| $t_{su(D-W)}$ | Data setup time for write | | (1) | | ns |
| $t_{h(W-D)}$ | Data hold time after write | | 0 | | ns |

Note:

1. The value is calculated by the following formulas based on the base clock cycles ($t_{c(Base)}$) and respective cycles of Tsu(A-R), Tw(R), Tsu(A-W), and Tw(W) set by registers EBC0 to EBC3. If the calculation results in a negative value, modify the value to be set. For the details of how to set values, refer to 9.3.5 “External Bus Timing”.

   $t_{su(S-R)} = t_{su(A-R)}$ = Tsu(A-R) × $t_{c(Base)}$ - 15 [ns]

   $t_{W(R)}$ = Tw(R) × $t_{c(Base)}$ - 10 [ns]

   $t_{su(S-W)} = t_{su(A-W)}$ = Tsu(A-W) × $t_{c(Base)}$ - 15 [ns]

   $t_{W(W)} = t_{su(D-W)}$ = Tw(W) × $t_{c(Base)}$ - 10 [ns]

$V_{CC}$ = 5 V

**Switching Characteristics ($V_{CC}$ = 4.2 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.36 External Bus Timing (Multiplexed bus)**

| Symbol | Characteristic | Measurement condition | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{su(S-ALE)}$ | Chip-select setup time for ALE | Refer to Figure 28.6 | (1) | | ns |
| $t_{h(R-S)}$ | Chip-select hold time after read | | 1.5 × $t_{c(Base)}$ - 15 | | ns |
| $t_{su(A-ALE)}$ | Address setup time for ALE | | (1) | | ns |
| $t_{h(ALE-A)}$ | Address hold time after ALE | | 0.5 × $t_{c(Base)}$ - 5 | | ns |
| $t_{h(R-A)}$ | Address hold time after read | | 1.5 × $t_{c(Base)}$ - 15 | | ns |
| $t_{d(ALE-R)}$ | ALE-read delay time | | 0.5 × $t_{c(Base)}$ - 5 | 0.5 × $t_{c(Base)}$ + 10 | ns |
| $t_{w(ALE)}$ | ALE pulse width | | (1) | | ns |
| $t_{dis(R-A)}$ | Address disable time after read | | | 8 | ns |
| $t_{w(R)}$ | Read pulse width | | (1) | | ns |
| $t_{h(W-S)}$ | Chip-select hold time after write | | 1.5 × $t_{c(Base)}$ - 15 | | ns |
| $t_{h(W-A)}$ | Address hold time after write | | 1.5 × $t_{c(Base)}$ - 15 | | ns |
| $t_{d(ALE-W)}$ | ALE-write delay time | | 0.5 × $t_{c(Base)}$ - 5 | 0.5 × $t_{c(Base)}$ + 10 | ns |
| $t_{w(W)}$ | Write pulse width | | (1) | | ns |
| $t_{su(D-W)}$ | Data setup time for write | | (1) | | ns |
| $t_{h(W-D)}$ | Data hold time after write | | 0.5 × $t_{c(Base)}$ | | ns |

Note:

1. The value is calculated by the following formulas based on the base clock cycles ($t_{c(Base)}$) and respective cycles of Tsu(A-R), Tw(R), Tsu(A-W), and Tw(W) set by registers EBC0 to EBC3. If the calculation results in a negative value, modify the value to be set. For the details of how to set values, refer to 9.3.5 “External Bus Timing”.

   $t_{su(S-ALE)} = t_{su(A-ALE)} = t_{w(ALE)}$ = (Tsu(A-R) - 0.5) × $t_{c(Base)}$ -15 [ns]

   $t_{w(R)}$ = Tw(R) × $t_{c(Base)}$ -10 [ns]

   $t_{w(W)} = t_{su(D-W)}$ = Tw(W) × $t_{c(Base)}$ -10 [ns]

$V_{CC}$ = 5 V

**Switching Characteristics ($V_{CC}$ = 4.2 to 5.5 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.37 Serial Interface**

| Symbol | Characteristic | Measurement condition | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{d(C-Q)}$ | TXDi output delay time | Refer to Figure 28.6 | | 80 | ns |
| $t_{h(C-Q)}$ | TXDi output hold time | | 0 | | ns |

**Table 28.38 Intelligent I/O**

| Symbol | Characteristic | Measurement condition | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{d(ISCLK2-TXD)}$ | ISTXD2 output delay time | Refer to Figure 28.6 | | 180 | ns |
| $t_{h(ISCLK2-RXD)}$ | ISTXD2 output hold time | | 0 | | ns |

**Table 28.39 Multi-master I²C-bus Interface (Standard-mode)**

| Symbol | Characteristic | Measurement condition | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{f(SCL)}$ | MSCL output fall time | Refer to Figure 28.6 | 2 | | ns |
| $t_{f(SDA)}$ | MSDA output fall time | | 2 | | ns |
| $t_{d(SDA-SCL)S}$ | MSCL output delay time after start condition/restart condition | | 20 × $t_{c(\phi IIC)}$ - 120 | 52 × $t_{c(\phi IIC)}$ - 40 | ns |
| $t_{d(SCL-SDA)P}$ | Restart condition/stop condition output delay time after MSCL becomes high | | 20 × $t_{c(\phi IIC)}$ + 40 | 52 × $t_{c(\phi IIC)}$ + 120 | ns |
| $t_{d(SCL-SDA)}$ | MSDA output delay time | | 2 × $t_{c(\phi IIC)}$ + 40 | 3 × $t_{c(\phi IIC)}$ + 120 | ns |

**Table 28.40 Multi-master I²C-bus Interface (Fast-mode)**

| Symbol | Characteristic | Measurement condition | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{f(SCL)}$ | MSCL output fall time | Refer to Figure 28.6 | 2 (1) | | ns |
| $t_{f(SDA)}$ | MSDA output fall time | | 2 (1) | | ns |
| $t_{d(SDA-SCL)S}$ | MSCL output delay time after start condition/restart condition | | 10 × $t_{c(\phi IIC)}$ - 120 | 26 × $t_{c(\phi IIC)}$ - 40 | ns |
| $t_{d(SCL-SDA)P}$ | Restart condition/stop condition output delay time after MSCL becomes high | | 10 × $t_{c(\phi IIC)}$ + 40 | 26 × $t_{c(\phi IIC)}$ + 120 | ns |
| $t_{d(SCL-SDA)}$ | MSDA output delay time | | 2 × $t_{c(\phi IIC)}$ + 40 | 3 × $t_{c(\phi IIC)}$ + 120 | ns |

Note:

1. External circuits are required to satisfy the I²C-bus specification.

$V_{CC}$ = 3.3 V

**Table 28.41 Electrical Characteristics (1/3) ($V_{CC}$ = 3.0 to 3.6 V, $V_{SS}$ = 0 V, $T_a$ = $T_{opr}$, and $f_{(CPU)}$ = 50 MHz, unless otherwise noted)**

| Symbol | Characteristic | | Measurement condition | Value | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| $V_{OH}$ | High level output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_7 (1) | $I_{OH}$ = -1 mA | $V_{CC}$ - 0.6 | | $V_{CC}$ | V |
| $V_{OL}$ | Low level output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_7 (1) | $I_{OL}$ = 1 mA | | | 0.5 | V |

Notes:
1. Ports P9_0, P9_2, and P11 to P15 are available in the 144-pin package only. Port P9_1 is designated as input pin in the 100-pin package.

$V_{CC}$ = 3.3 V

**Table 28.42 Electrical Characteristics (2/3) ($V_{CC}$ = 3.0 to 3.6 V, $V_{SS}$ = 0 V, $T_a$ = $T_{opr}$, and $f_{(CPU)}$ = 50 MHz, unless otherwise noted)**

| Symbol | Characteristic | | Measurement condition | Value | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| $V_{T+} - V_{T-}$ | Hysteresis | $\overline{HOLD}$, $\overline{RDY}$, $\overline{NMI}$, $\overline{INT0}$ to $\overline{INT8}$, $\overline{KI0}$ to $\overline{KI3}$, TA0IN to TA4IN, TA0OUT to TA4OUT, TB0IN to TB5IN, $\overline{CTS0}$ to $\overline{CTS8}$, CLK0 to CLK8, RXD0 to RXD8, SCL0 to SCL6, SDA0 to SDA6, $\overline{SS0}$ to $\overline{SS6}$, SRXD0 to SRXD6, $\overline{ADTRG}$, IIO0_0 to IIO0_7, IIO1_0 to IIO1_7, UD0A, UD0B, UD1A, UD1B, ISCLK2, ISRXD2, IEIN, CAN0IN, CAN1IN, $\overline{CAN0WU}$, $\overline{CAN1WU}$ (1) | | 0.2 | | 1.0 | V |
| | | $\overline{RESET}$ | | 0.2 | | 1.8 | V |
| $I_{IH}$ | High level input current | XIN, $\overline{RESET}$, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_7 (2) | $V_I$ = 3.3 V | | | 4.0 | µA |
| $I_{IL}$ | Low level input current | XIN, $\overline{RESET}$, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_7 (2) | $V_I$ = 0 V | | | -4.0 | µA |
| $R_{PULLUP}$ | Pull-up resistor | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P5_0 to P5_3, P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_7 (2) | $V_I$ = 0 V | 50 | 100 | 500 | kΩ |
| $R_{fXIN}$ | Feedback resistor | XIN | | | 3 | | MΩ |
| $R_{fXCIN}$ | Feedback resistor | XCIN | | | 25 | | MΩ |

Notes:
1. Pins $\overline{INT6}$ to $\overline{INT8}$ are available in the 144-pin package only.
2. Ports P9_0, P9_2, and P11 to P15 are available in the 144-pin package only. Port P9_1 is designated as input pin in the 100-pin package.

$V_{CC}$ = 3.3 V

**Table 28.43 Electrical Characteristics (3/3)**
**($V_{CC}$ = 3.0 to 3.6 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

| Symbol | Characteristic | Measurement condition | | Value | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| $I_{CC}$ | Power supply current | In single-chip mode, output pins are left open and others are connected to $V_{SS}$<br><br>XIN-XOUT<br>Drive power: low<br><br>XCIN-XCOUT<br>Drive power: low | $f_{(CPU)}$ = 50 MHz, $f_{(BCLK)}$ = 25 MHz,<br>$f_{(XIN)}$ = 8 MHz,<br>Active: XIN, PLL,<br>Stopped: XCIN, OCO | | 32 | 45 | mA |
| | | | $f_{(CPU)}$ = $f_{SO(PLL)}$/24 MHz,<br>Active: PLL (self-oscillation),<br>Stopped: XIN, XCIN, OCO | | 9 | | mA |
| | | | $f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(XIN)}$/256 MHz,<br>$f_{(XIN)}$ = 8 MHz,<br>Active: XIN,<br>Stopped: PLL, XCIN, OCO | | 670 | | µA |
| | | | $f_{(CPU)}$ = $f_{(BCLK)}$ = 32.768 kHz,<br>Active: XCIN,<br>Stopped: XIN, PLL, OCO,<br>Main regulator: shutdown | | 180 | | µA |
| | | | $f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(OCO)}$/4 kHz,<br>Active: OCO,<br>Stopped: XIN, PLL, XCIN,<br>Main regulator: shutdown | | 190 | | µA |
| | | | $f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(XIN)}$/256 MHz,<br>$f_{(XIN)}$ = 8 MHz,<br>Active: XIN,<br>Stopped: PLL, XCIN, OCO,<br>$T_a$ = 25°C, Wait mode | | 500 | 900 | µA |
| | | | $f_{(CPU)}$ = $f_{(BCLK)}$ = 32.768 kHz,<br>Active: XCIN,<br>Stopped: XIN, PLL, OCO,<br>Main regulator: shutdown,<br>$T_a$ = 25°C, Wait mode | | 8 | 140 | µA |
| | | | $f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(OCO)}$/4 kHz,<br>Active: OCO,<br>Stopped: XIN, PLL, XCIN,<br>Main regulator: shutdown,<br>$T_a$ = 25°C, Wait mode | | 10 | 150 | µA |
| | | | Stopped: all clocks,<br>Main regulator: shutdown,<br>$T_a$ = 25°C | | 5 | 70 | µA |

$V_{CC}$ = 3.3 V

**Table 28.44 A/D Conversion Characteristics ($V_{CC}$ = $AV_{CC}$ = $V_{REF}$ = 3.0 to 3.6 V, $V_{SS}$ = $AV_{SS}$ = 0 V, $T_a$ = $T_{opr}$, and $f_{(BCLK)}$ = 25 MHz, unless otherwise noted)**

| Symbol | Characteristic | Measurement condition | | Value | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| — | Resolution | $V_{REF}$ = $V_{CC}$ | | | | 10 | Bits |
| — | Absolute error | $V_{REF}$ = $V_{CC}$ = 3.3 V | AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, ANEX1 (1) | | | ±5 | LSB |
| | | | External op-amp connection mode | | | ±7 | LSB |
| INL | Integral non-linearity error | $V_{REF}$ = $V_{CC}$ = 3.3 V | AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, ANEX1 (1) | | | ±5 | LSB |
| | | | External op-amp connection mode | | | ±7 | LSB |
| DNL | Differential non-linearity error | $V_{REF}$ = $V_{CC}$ = 3.3 V | | | | ±1 | LSB |
| — | Offset error | | | | | ±3 | LSB |
| — | Gain error | | | | | ±3 | LSB |
| $R_{LADDER}$ | Resistor ladder | $V_{REF}$ = $V_{CC}$ | | 4 | | 20 | kΩ |
| $t_{CONV}$ | Conversion time (10 bits) | $\phi_{AD}$ = 10 MHz, with sample and hold function | | 3.3 | | | µs |
| $t_{CONV}$ | Conversion time (8 bits) | $\phi_{AD}$ = 10 MHz, with sample and hold function | | 2.8 | | | µs |
| $t_{SAMP}$ | Sampling time | $\phi_{AD}$ = 10 MHz | | 0.3 | | | µs |
| $V_{IA}$ | Analog input voltage | | | 0 | | $V_{REF}$ | V |
| $\phi_{AD}$ | Operating clock frequency | without sample and hold function | | 0.25 | | 10 | MHz |
| | | with sample and hold function | | 1 | | 10 | MHz |

Note:
1. Pins AN15_0 to AN15_7 are available in the 144-pin package only.

$V_{CC}$ = 3.3 V

**Table 28.45 D/A Conversion Characteristics ($V_{CC}$ = $AV_{CC}$ = $V_{REF}$ = 3.0 to 3.6 V, $V_{SS}$ = $AV_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

| Symbol | Characteristic | Measurement condition | Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| — | Resolution | | | | 8 | Bits |
| — | Absolute precision | | | | 1.0 | % |
| $t_S$ | Settling time | | | | 3 | µs |
| $R_O$ | Output resistance | | 4 | 10 | 20 | kΩ |
| $I_{VREF}$ | Reference input current | (1) | | | 1.0 | mA |

Note:

1. One D/A converter is used. The DAi register (i = 0, 1) of the other unused converter is set to 00h. The resistor ladder for A/D converter is not considered.
   Even when the VCUT bit in the AD0CON1 register is set to 0 ($V_{REF}$ disconnected), $I_{VREF}$ is supplied.

$V_{CC}$ = 3.3 V

**Timing Requirements ($V_{CC}$ = 3.0 to 3.6 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.46 External Clock Input**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(X)}$ | External clock input period | 62.5 | 250 | ns |
| $t_{w(XH)}$ | External clock input high level pulse width | 25 | | ns |
| $t_{w(XL)}$ | External clock input low level pulse width | 25 | | ns |
| $t_{r(X)}$ | External clock input rise time | | 5 | ns |
| $t_{f(X)}$ | External clock input fall time | | 5 | ns |
| $t_w$ / $t_c$ | External clock input duty | 40 | 60 | % |

**Table 28.47 External Bus Timing**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{su(D-R)}$ | Data setup time for read | 40 | | ns |
| $t_{h(R-D)}$ | Data hold time after read | 0 | | ns |
| $t_{dis(R-D)}$ | Data disable time after read | | 0.5 × $t_{c(Base)}$ + 10 | ns |

$V_{CC}$ = 3.3 V

**Timing Requirements ($V_{CC}$ = 3.0 to 3.6 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.48 Timer A Input (Counting input in event counter mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TAiIN input clock period | 200 | | ns |
| $t_{w(TAH)}$ | TAiIN input high level pulse width | 80 | | ns |
| $t_{w(TAL)}$ | TAiIN input low level pulse width | 80 | | ns |

**Table 28.49 Timer A Input (Gating input in timer mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TAiIN input clock period | 400 | | ns |
| $t_{w(TAH)}$ | TAiIN input high level pulse width | 180 | | ns |
| $t_{w(TAL)}$ | TAiIN input low level pulse width | 180 | | ns |

**Table 28.50 Timer A Input (External trigger input in one-shot timer mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TAiIN input clock period | 200 | | ns |
| $t_{w(TAH)}$ | TAiIN input high level pulse width | 80 | | ns |
| $t_{w(TAL)}$ | TAiIN input low level pulse width | 80 | | ns |

**Table 28.51 Timer A Input (External trigger input in pulse-width modulation mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{w(TAH)}$ | TAiIN input high level pulse width | 80 | | ns |
| $t_{W(TAL)}$ | TAiIN input low level pulse width | 80 | | ns |

**Table 28.52 Timer A Input (Increment/decrement count switching input in event counter mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(UP)}$ | TAiOUT input clock period | 2000 | | ns |
| $t_{w(UPH)}$ | TAiOUT input high level pulse width | 1000 | | ns |
| $t_{w(UPL)}$ | TAiOUT input low level pulse width | 1000 | | ns |
| $t_{su(UP-TIN)}$ | TAiOUT input setup time | 400 | | ns |
| $t_{h(TIN-UP)}$ | TAiOUT input hold time | 400 | | ns |

$V_{CC}$ = 3.3 V

**Timing Requirements ($V_{CC}$ = 3.0 to 3.6 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.53 Timer B Input (Counting input in event counter mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input clock period (one edge counting) | 200 | | ns |
| $t_{w(TBH)}$ | TBiIN input high level pulse width (one edge counting) | 80 | | ns |
| $t_{w(TBL)}$ | TBiIN input low level pulse width (one edge counting) | 80 | | ns |
| $t_{c(TB)}$ | TBiIN input clock period (both edges counting) | 200 | | ns |
| $t_{w(TBH)}$ | TBiIN input high level pulse width (both edges counting) | 80 | | ns |
| $t_{w(TBL)}$ | TBiIN input low level pulse width (both edges counting) | 80 | | ns |

**Table 28.54 Timer B Input (Pulse period measure mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input clock period | 400 | | ns |
| $t_{w(TBH)}$ | TBiIN input high level pulse width | 180 | | ns |
| $t_{w(TBL)}$ | TBiIN input low level pulse width | 180 | | ns |

**Table 28.55 Timer B Input (Pulse-width measure mode)**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input clock period | 400 | | ns |
| $t_{w(TBH)}$ | TBiIN input high level pulse width | 180 | | ns |
| $t_{w(TBL)}$ | TBiIN input low level pulse width | 180 | | ns |

## $V_{CC}$ = 3.3 V

**Timing Requirements ($V_{CC}$ = 3.0 to 3.6 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.56 Serial Interface**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input clock period | 200 | | ns |
| $t_{w(CKH)}$ | CLKi input high level pulse width | 80 | | ns |
| $t_{w(CKL)}$ | CLKi input low level pulse width | 80 | | ns |
| $t_{su(D-C)}$ | RXDi input setup time | 80 | | ns |
| $t_{h(C-D)}$ | RXDi input hold time | 90 | | ns |

**Table 28.57 A/D Trigger Input**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{w(ADH)}$ | $\overline{ADTRG}$ input high level pulse width<br>Hardware trigger input high level pulse width | $\frac{3}{\phi_{AD}}$ | | ns |
| $t_{w(ADL)}$ | $\overline{ADTRG}$ input low level pulse width<br>Hardware trigger input high level pulse width | 125 | | ns |

**Table 28.58 External Interrupt INTi Input**

| Symbol | Characteristic | | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{w(INH)}$ | $\overline{INTi}$ input high level pulse width | Edge sensitive | 250 | | ns |
| | | Level sensitive | $t_{c(CPU)}$ + 200 | | ns |
| $t_{W(INL)}$ | $\overline{INTi}$ input low level pulse width | Edge sensitive | 250 | | ns |
| | | Level sensitive | $t_{c(CPU)}$ + 200 | | ns |

**Table 28.59 Intelligent I/O**

| Symbol | Characteristic | Value | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(ISCLK2)}$ | ISCLK2 input clock period | 600 | | ns |
| $t_{W(ISCLK2H)}$ | ISCLK2 input high level pulse width | 270 | | ns |
| $t_{W(ISCLK2L)}$ | ISCLK2 input low level pulse width | 270 | | ns |
| $t_{su(RXD-ISCLK2)}$ | ISRXD2 input setup time | 150 | | ns |
| $t_{h(ISCLK2-RXD)}$ | ISRXD2 input hold time | 100 | | ns |

$V_{CC}$ = 3.3 V

**Timing Requirements ($V_{CC}$ = 3.0 to 3.6 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.60 Multi-master I²C-bus Interface**

| Symbol | Characteristic | Value | | | | Unit |
|---|---|---|---|---|---|---|
| | | Standard-mode | | Fast-mode | | |
| | | Min. | Max. | Min. | Max. | |
| $t_{w(SCLH)}$ | MSCL input high level pulse width | 600 | | 600 | | ns |
| $t_{w(SCLL)}$ | MSCL input low level pulse width | 600 | | 600 | | ns |
| $t_{r(SCL)}$ | MSCL input rise time | | 1000 | | 300 | ns |
| $t_{f(SCL)}$ | MSCL input fall time | | 300 | | 300 | ns |
| $t_{r(SDA)}$ | MSDA input rise time | | 1000 | | 300 | ns |
| $t_{f(SDA)}$ | MSDA input fall time | | 300 | | 300 | ns |
| $t_{h(SDA-SCL)S}$ | MSCL high level hold time after start condition/restart condition | (1) | | $2 \times t_{c(\phi IIC)} + 40$ | | ns |
| $t_{su(SCL-SDA)P}$ | MSCL high level setup time for restart condition/stop condition | (1) | | $2 \times t_{c(\phi IIC)} + 40$ | | ns |
| $t_{w(SDAH)P}$ | MSDA high level pulse width after stop condition | (1) | | $4 \times t_{c(\phi IIC)} + 40$ | | ns |
| $t_{su(SDA-SCL)}$ | MSDA input setup time | 100 | | 100 | | ns |
| $t_{h(SCL-SDA)}$ | MSDA input hold time | 0 | | 0 | | ns |

Note:

1. The value is calculated by the following formulas based on a value SSC set by bits SSC4 to SSC0 in the I2CSSCR register:

   $t_{h(SDA-SCL)S} = SSC \div 2 \times t_{c(\phi IIC)} + 40$ [ns]

   $t_{su(SCL-SDA)P} = (SSC \div 2 + 1) \times t_{c(\phi IIC)} + 40$ [ns]

   $t_{w(SDAH)P} = (SSC + 1) \times t_{c(\phi IIC)} + 40$ [ns]

$V_{CC}$ = 3.3 V

**Switching Characteristics ($V_{CC}$ = 3.0 to 3.6 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.61 External Bus Timing (Separate bus)**

| Symbol | Characteristic | Measurement condition | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{su(S-R)}$ | Chip-select setup time for read | Refer to Figure 28.6 | (1) | | ns |
| $t_{h(R-S)}$ | Chip-select hold time after read | | $t_{c(Base)}$ - 15 | | ns |
| $t_{su(A-R)}$ | Address setup time for read | | (1) | | ns |
| $t_{h(R-A)}$ | Address hold time after read | | $t_{c(Base)}$ - 15 | | ns |
| $t_{w(R)}$ | Read pulse width | | (1) | | ns |
| $t_{su(S-W)}$ | Chip-select setup time for write | | (1) | | ns |
| $t_{h(W-S)}$ | Chip-select hold time after write | | 1.5 × $t_{c(Base)}$ - 15 | | ns |
| $t_{su(A-W)}$ | Address setup time for write | | (1) | | ns |
| $t_{h(W-A)}$ | Address hold time after write | | 1.5 × $t_{c(Base)}$ - 15 | | ns |
| $t_{w(W)}$ | Write pulse width | | (1) | | ns |
| $t_{su(D-W)}$ | Data setup time for write | | (1) | | ns |
| $t_{h(W-D)}$ | Data hold time after write | | 0 | | ns |

Note:

1. The value is calculated by the following formulas based on the base clock cycles ($t_{c(Base)}$) and respective cycles of Tsu(A-R), Tw(R), Tsu(A-W), and Tw(W) set by registers EBC0 to EBC3. If the calculation results in a negative value, modify the value to be set. For the details of how to set values, refer to 9.3.5 “External Bus Timing”.

   $t_{su(S-R)} = t_{su(A-R)}$ = Tsu(A-R) × $t_{c(Base)}$ - 15 [ns]

   $t_{w(R)}$ = Tw(R) × $t_{c(Base)}$ - 10 [ns]

   $t_{su(S-W)} = t_{su(A-W)}$ = Tsu(A-W) × $t_{c(Base)}$ - 15 [ns]

   $t_{w(W)} = t_{su(D-W)}$ = Tw(W) × $t_{c(Base)}$ - 10 [ns]

$V_{CC}$ = 3.3 V

**Switching Characteristics ($V_{CC}$ = 3.0 to 3.6 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.62 External Bus Timing (Multiplexed bus)**

| Symbol | Characteristic | Measurement condition | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{su(S-ALE)}$ | Chip-select setup time for ALE | Refer to Figure 28.6 | (1) | | ns |
| $t_{h(R-S)}$ | Chip-select hold time after read | | 1.5 × $t_{c(Base)}$ - 15 | | ns |
| $t_{su(A-ALE)}$ | Address setup time for ALE | | (1) | | ns |
| $t_{h(ALE-A)}$ | Address hold time after ALE | | 0.5 × $t_{c(Base)}$ - 5 | | ns |
| $t_{h(R-A)}$ | Address hold time after read | | 1.5 × $t_{c(Base)}$ - 15 | | ns |
| $t_{d(ALE-R)}$ | ALE-read delay time | | 0.5 × $t_{c(Base)}$ - 5 | 0.5 × $t_{c(Base)}$ + 10 | ns |
| $t_{W(ALE)}$ | ALE pulse width | | (1) | | ns |
| $t_{dis(R-A)}$ | Address disable time after read | | | 8 | ns |
| $t_{w(R)}$ | Read pulse width | | (1) | | ns |
| $t_{h(W-S)}$ | Chip-select hold time after write | | 1.5 × $t_{c(Base)}$ - 15 | | ns |
| $t_{h(W-A)}$ | Address hold time after write | | 1.5 × $t_{c(Base)}$ - 15 | | ns |
| $t_{d(ALE-W)}$ | ALE-write delay time | | 0.5 × $t_{c(Base)}$ - 5 | 0.5 × $t_{c(Base)}$ + 10 | ns |
| $t_{w(W)}$ | Write pulse width | | (1) | | ns |
| $t_{su(D-W)}$ | Data setup time for write | | (1) | | ns |
| $t_{h(W-D)}$ | Data hold time after write | | 0.5 × $t_{c(Base)}$ | | ns |

Note:

1. The value is calculated by the following formulas based on the base clock cycles ($t_{c(Base)}$) and respective cycles of Tsu(A-R), Tw(R), Tsu(A-W), and Tw(W) set by registers EBC0 to EBC3. If the calculation results in a negative value, modify the value to be set. For the details of how to set values, refer to 9.3.5 “External Bus Timing”.

   $t_{su(S-ALE)}$ = $t_{su(A-ALE)}$ = (Tsu(A-R) - 0.5) × $t_{c(Base)}$ -15 [ns]

   $t_{W(ALE)}$ = (Tsu(A-R) - 0.5) × $t_{c(Base)}$ - 20 [ns]

   $t_{w(R)}$ = Tw(R) × $t_{c(Base)}$ -10 [ns]

   $t_{w(W)}$ = $t_{su(D-W)}$ = Tw(W) × $t_{c(Base)}$ -10 [ns]

$V_{CC}$ = 3.3 V

**Switching Characteristics ($V_{CC}$ = 3.0 to 3.6 V, $V_{SS}$ = 0 V, and $T_a$ = $T_{opr}$, unless otherwise noted)**

**Table 28.63 Serial Interface**

| Symbol | Characteristic | Measurement condition | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{d(C-Q)}$ | TXDi output delay time | Refer to Figure 28.6 | | 80 | ns |
| $t_{h(C-Q)}$ | TXDi output hold time | | 0 | | ns |

**Table 28.64 Intelligent I/O**

| Symbol | Characteristic | Measurement condition | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{d(ISCLK2-TXD)}$ | ISTXD2 output delay time | Refer to Figure 28.6 | | 180 | ns |
| $t_{h(ISCLK2-RXD)}$ | ISTXD2 output hold time | | 0 | | ns |

**Table 28.65 Multi-master I²C-bus Interface (Standard-mode)**

| Symbol | Characteristic | Measurement condition | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{f(SCL)}$ | MSCL output fall time | Refer to Figure 28.6 | 2 | | ns |
| $t_{f(SDA)}$ | MSDA output fall time | | 2 | | ns |
| $t_{d(SDA-SCL)S}$ | MSCL output delay time after start condition/restart condition | | 20 × $t_{c(\phi IIC)}$ - 120 | 52 × $t_{c(\phi IIC)}$ - 40 | ns |
| $t_{d(SCL-SDA)P}$ | Restart condition/stop condition output delay time after MSCL becomes high | | 20 × $t_{c(\phi IIC)}$ + 40 | 52 × $t_{c(\phi IIC)}$ + 120 | ns |
| $t_{d(SCL-SDA)}$ | MSDA output delay time | | 2 × $t_{c(\phi IIC)}$ + 40 | 3 × $t_{c(\phi IIC)}$ + 120 | ns |

**Table 28.66 Multi-master I²C-bus Interface (Fast-mode)**

| Symbol | Characteristic | Measurement condition | Value | | Unit |
|---|---|---|---|---|---|
| | | | Min. | Max. | |
| $t_{f(SCL)}$ | MSCL output fall time | Refer to Figure 28.6 | 2 (1) | | ns |
| $t_{f(SDA)}$ | MSDA output fall time | | 2 (1) | | ns |
| $t_{d(SDA-SCL)S}$ | MSCL output delay time after start condition/restart condition | | 10 × $t_{c(\phi IIC)}$ - 120 | 26 × $t_{c(\phi IIC)}$ - 40 | ns |
| $t_{d(SCL-SDA)P}$ | Restart condition/stop condition output delay time after MSCL becomes high | | 10 × $t_{c(\phi IIC)}$ + 40 | 26 × $t_{c(\phi IIC)}$ + 120 | ns |
| $t_{d(SCL-SDA)}$ | MSDA output delay time | | 2 × $t_{c(\phi IIC)}$ + 40 | 3 × $t_{c(\phi IIC)}$ + 120 | ns |

Note:
1. External circuits are required to satisfy the I²C-bus specification.

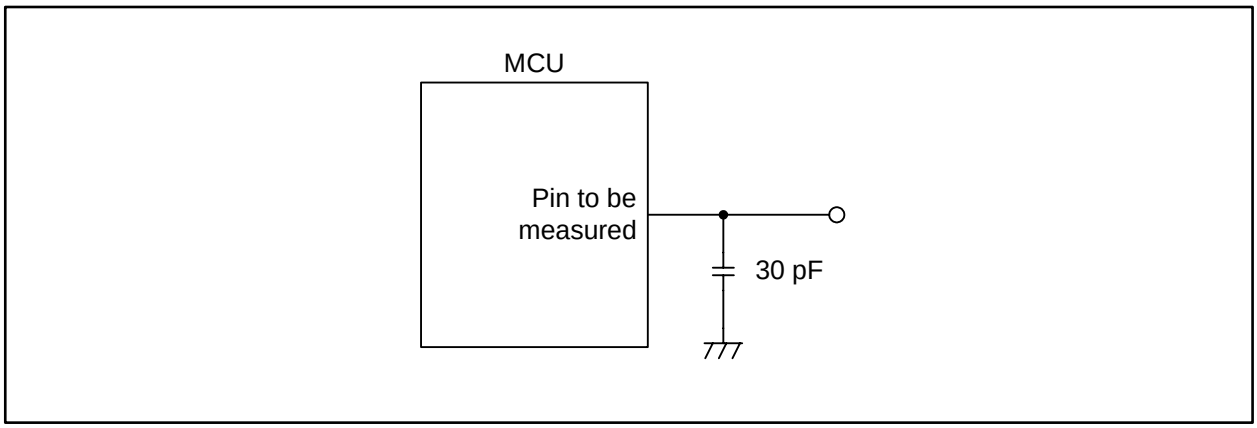



**Figure 28.6 Switching Characteristic Measurement Circuit**

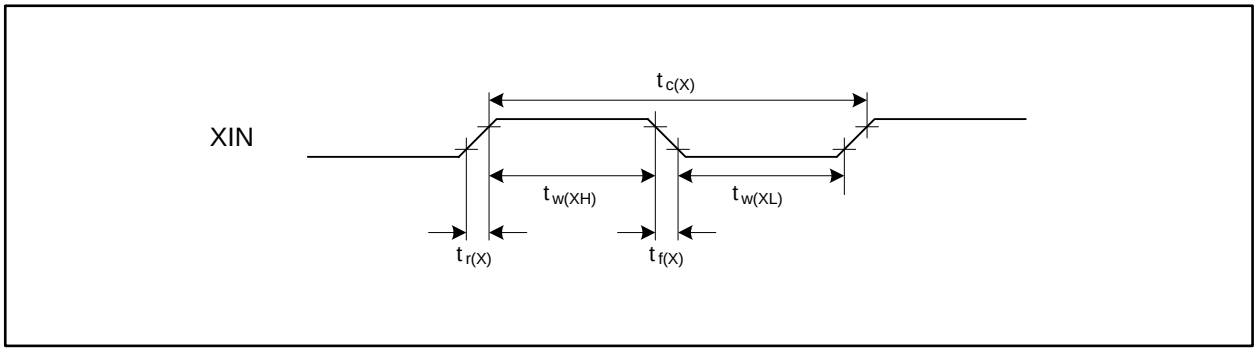
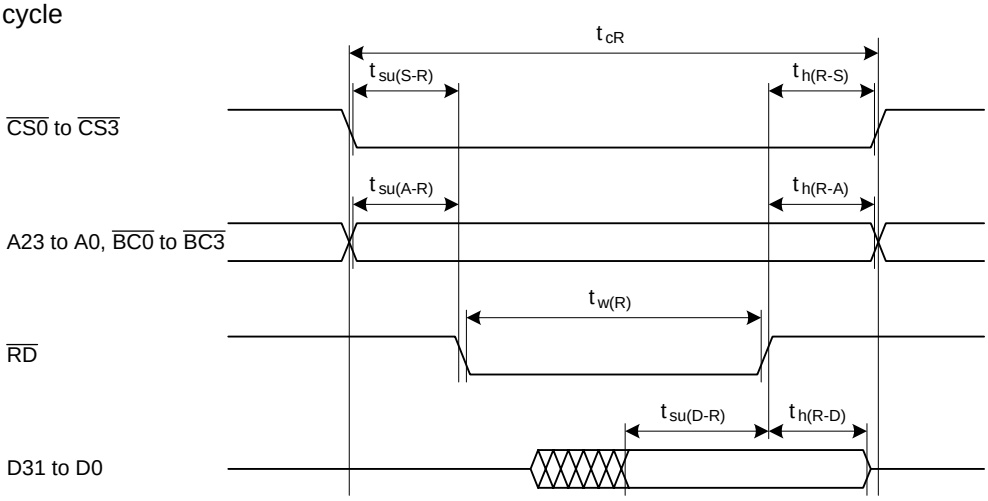


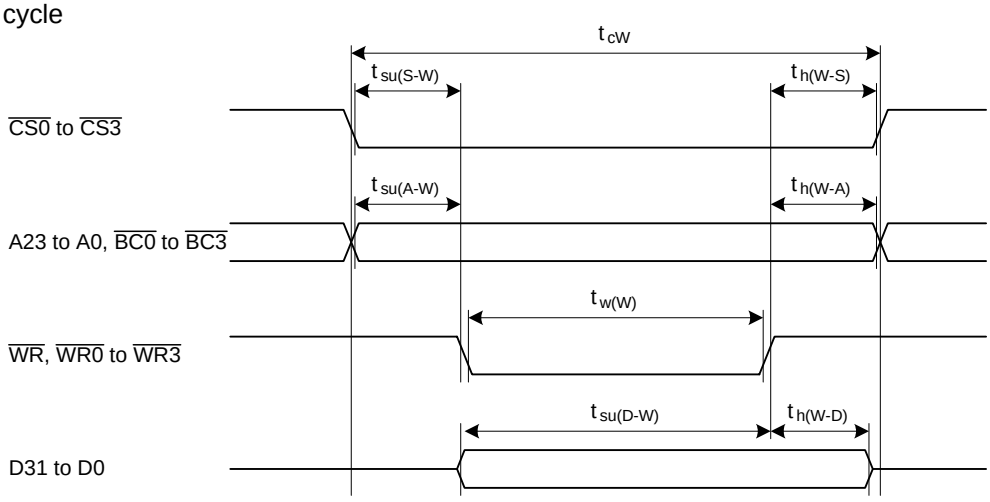

**Figure 28.7 External Clock Input Timing**

External bus timing (Separate bus)

Read cycle




Write cycle



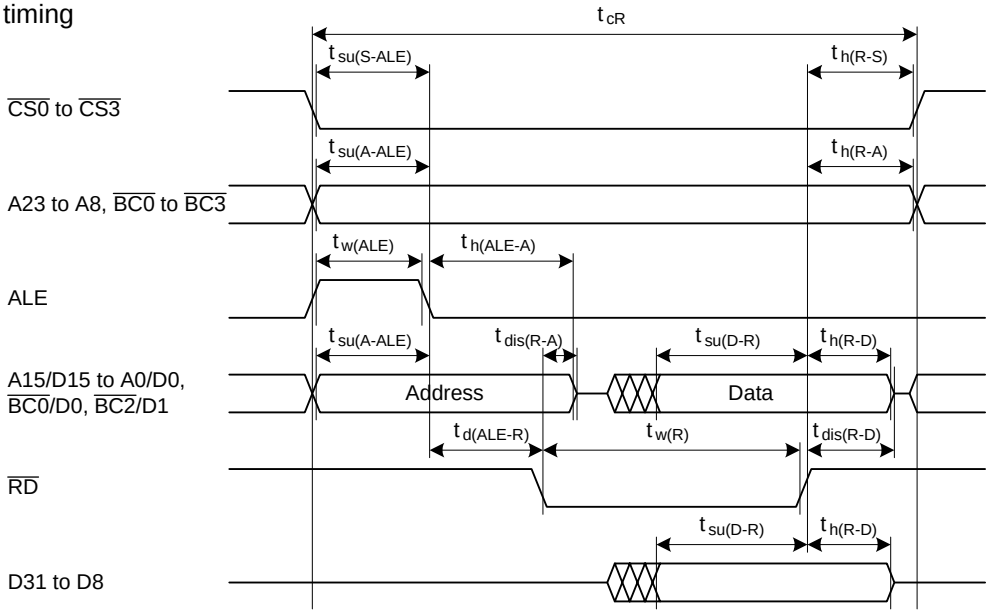

Measurement conditions

| Item | | $V_{CC}$ = 4.2 to 5.5 V | $V_{CC}$ = 3.0 to 3.6 V |
|---|---|---|---|
| Criterion for input voltage | $V_{IH}$ | 2.5 V | 1.5 V |
| | $V_{IL}$ | 0.8 V | 0.5 V |
| Criterion for output voltage | $V_{OH}$ | 2.0 V | 2.4 V |
| | $V_{OL}$ | 0.8 V | 0.5 V |

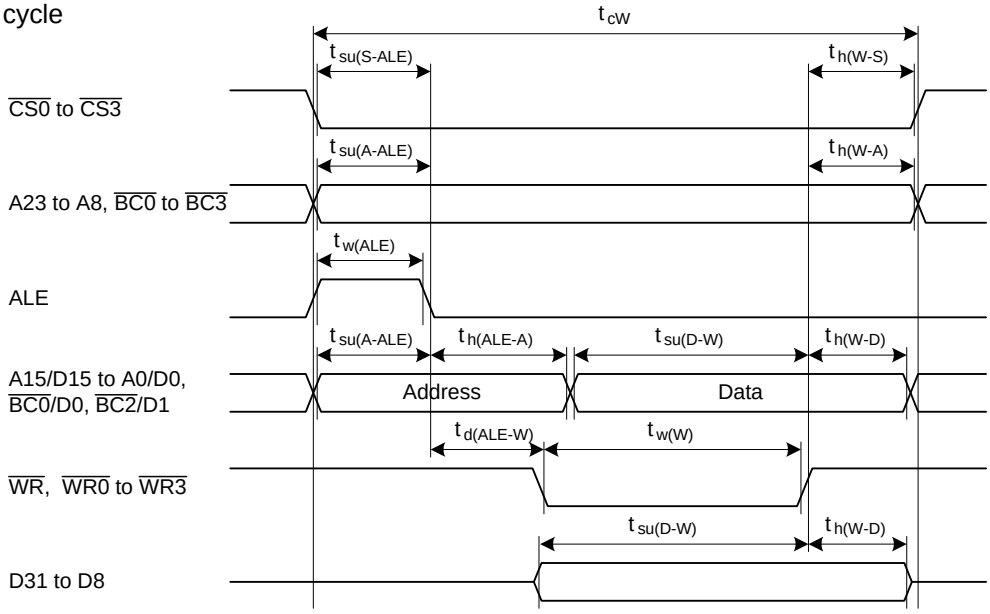
**Figure 28.8 External Bus Timing (Separate Bus)**

External bus timing (Multiplexed bus)

Read timing




Write cycle




Measurement conditions

| Item | | $V_{CC}$ = 4.2 to 5.5 V | $V_{CC}$ = 3.0 to 3.6 V |
|---|---|---|---|
| Criterion for input voltage | $V_{IH}$ | 2.5 V | 1.5 V |
| | $V_{IL}$ | 0.8 V | 0.5 V |
| Criterion for output voltage | $V_{OH}$ | 2.0 V | 2.4 V |
| | $V_{OL}$ | 0.8 V | 0.5 V |

**Figure 28.9 External Bus Timing (Multiplexed Bus)**

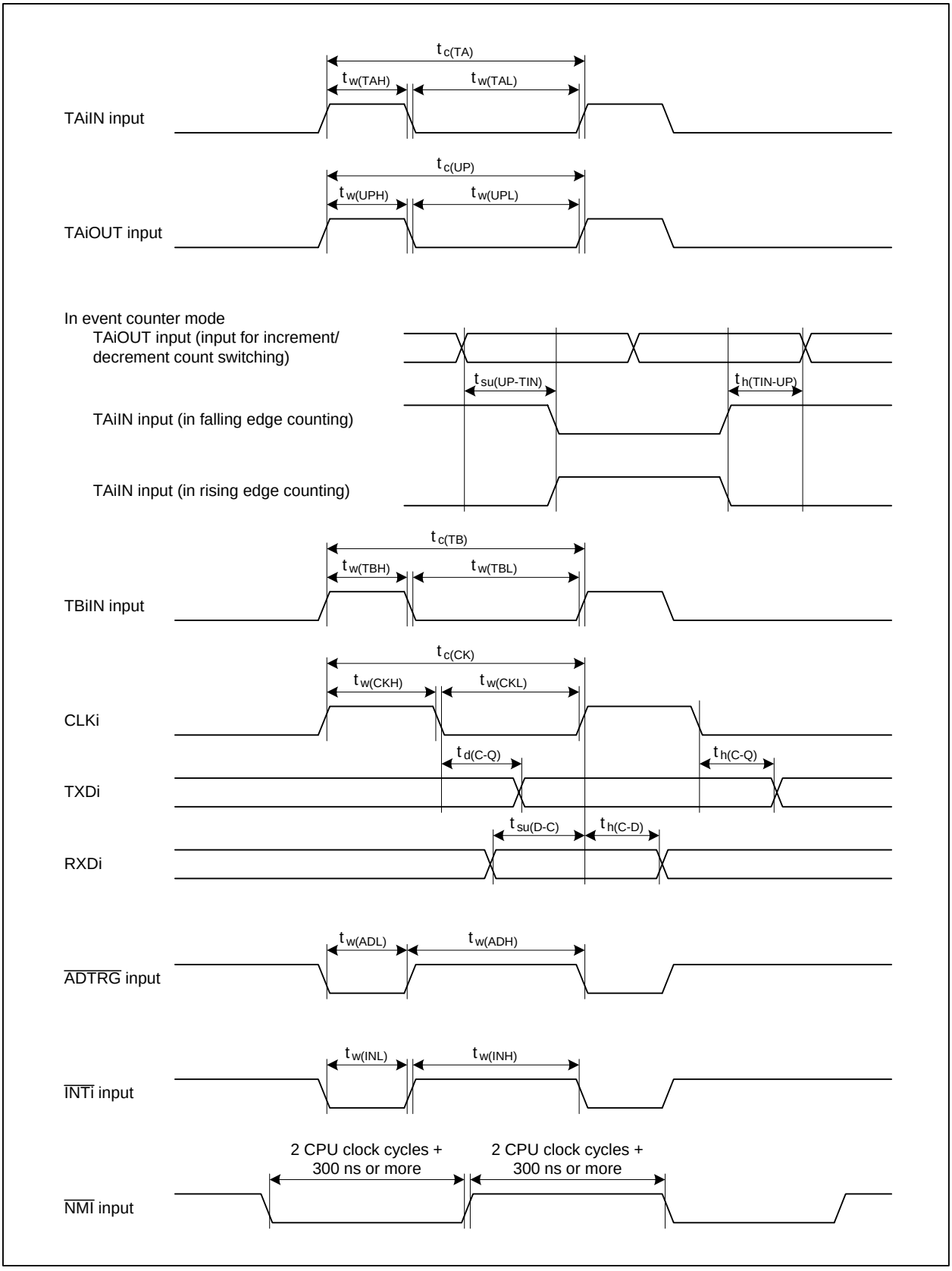



**Figure 28.10 Timing of Peripheral Functions**

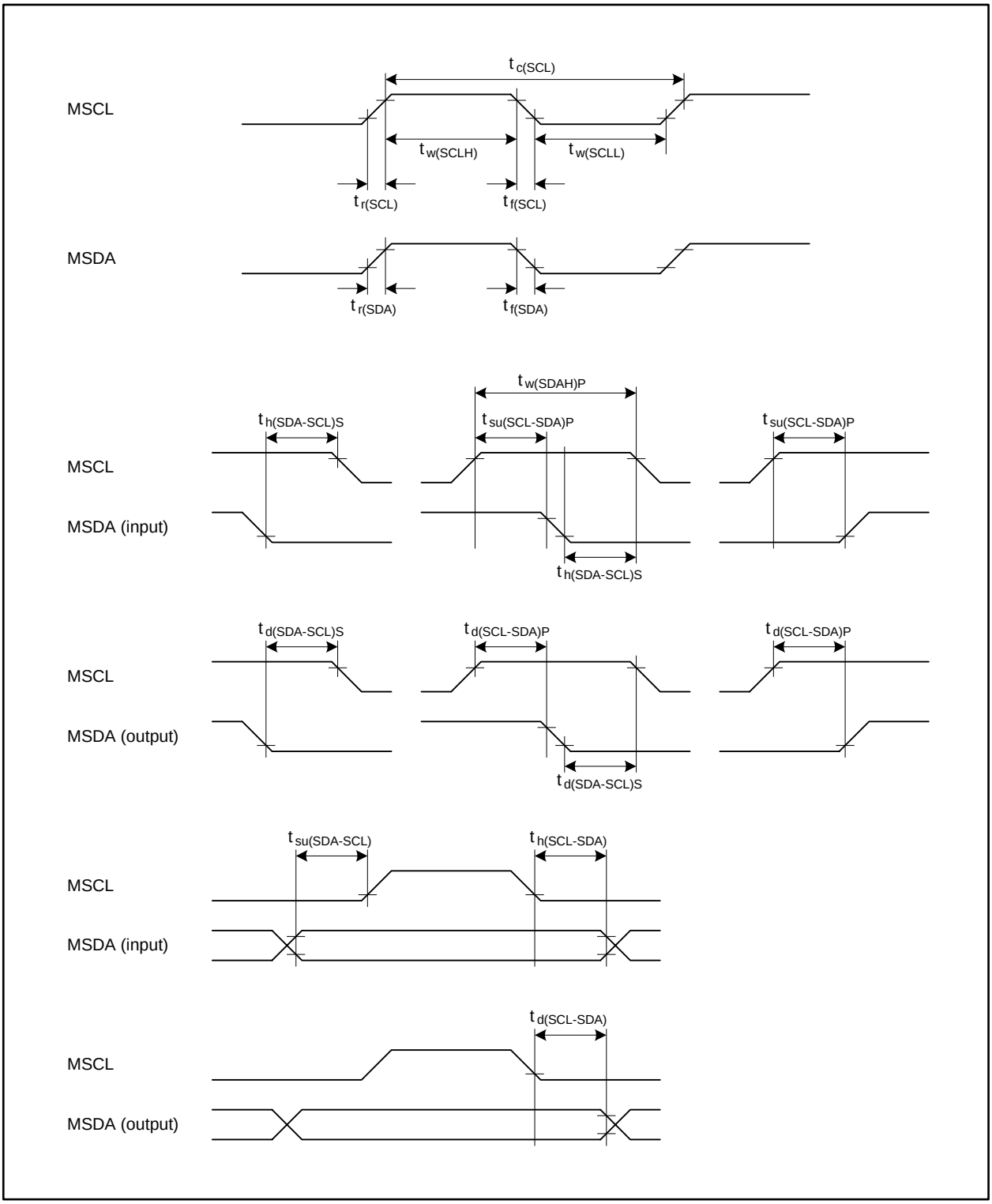



**Figure 28.11 Timing of Multi-master I²C-bus Interface**

# 29. Usage Notes

## 29.1 Notes on Board Designing

### 29.1.1 Power Supply Pins

The board should be designed so that there is no potential difference between pins with the same name. Note the following points:

- Connect all VSS pins to an identical GND. The trances for the pins should be as wide as physically possible so that the same voltage can be applied to every VSS pin.
- Connect all VCC pins to an identical power supply. The traces for the pins should be as wide as physically possible so that the same voltage can be applied to every VCC pin.

Insert a capacitor between each VCC pin and the VSS pin to ensure the noise tolerance. The capacitor should be beneficially effective at high/low frequencies and should have around 0.1 µF of capacitance. The traces for the capacitor and the power supply pins should be short and wide as much as physically possible.

### 29.1.2 Supply Voltage

The device is operationally guaranteed under operating conditions specified in electrical characteristics.
Drive the $\overline{\text{RESET}}$ pin low before the supply voltage becomes lower than the recommended value.

## 29.2 Notes on Register Setting

### 29.2.1 Registers with Write-only Bits

Table 29.1 lists registers containing write-only bits. For the setting of these registers, read-modify-write instructions listed in Table 29.2 cannot be used since each of these instructions reads the value of an address, modifies the value, and writes to the same address. To set a new value by modifying the previous one, write the previous value into RAM as well as to the register, change the contents of the RAM and then transfer the new value to the register by the MOV instruction.

**Table 29.1 Registers with Write-only Bit(s)**

| Module | Register | Symbol | Address(es) |
|---|---|---|---|
| Watchdog timer | Watchdog timer start register | WDTS | 04404Eh |
| Timer A | Timer A0 register (1) | TA0 | 0347h-0346h |
| | Timer A1 register (1) | TA1 | 0349h-0348h |
| | Timer A2 register (1) | TA2 | 034Bh-034Ah |
| | Timer A3 register (1) | TA3 | 034Dh-034Ch |
| | Timer A4 register (1) | TA4 | 034Fh-034Eh |
| | Increment/decrement counting select register | UDF | 0344h |
| Three-phase motor control timers | Timer B2 interrupt generating frequency set counter | ICTB2 | 030Dh |
| | Timer A1-1 register | TA11 | 0303h-0302h |
| | Timer A2-1 register | TA21 | 0305h-0304h |
| | Timer A4-1 register | TA41 | 0307h-0306h |
| | Dead time timer | DTT | 030Ch |
| Serial interface | UART0 bit rate register | U0BRG | 0369h |
| | UART1 bit rate register | U1BRG | 02E9h |
| | UART2 bit rate register | U2BRG | 0339h |
| | UART3 bit rate register | U3BRG | 0329h |
| | UART4 bit rate register | U4BRG | 02F9h |
| | UART5 bit rate register | U5BRG | 01C9h |
| | UART6 bit rate register | U6BRG | 01D9h |
| | UART7 bit rate register | U7BRG | 01E1h |
| | UART8 bit rate register | U8BRG | 01E9h |
| | UART0 transmit buffer register | U0TB | 036Bh-036Ah |
| | UART1 transmit buffer register | U1TB | 02EBh-02EAh |
| | UART2 transmit buffer register | U2TB | 033Bh-033Ah |
| | UART3 transmit buffer register | U3TB | 032Bh-032Ah |
| | UART4 transmit buffer register | U4TB | 02FBh-02FAh |
| | UART5 transmit buffer register | U5TB | 01CBh-01CAh |
| | UART6 transmit buffer register | U6TB | 01DBh-01DAh |
| | UART7 transmit buffer register | U7TB | 01E3h-01E2h |
| | UART8 transmit buffer register | U8TB | 01EBh-01EAh |
| Intelligent I/O | Group 2 SIO transmit buffer register | G2TB | 016Dh-016Ch |
| CAN module | CAN0 receive FIFO pointer control register | C0RFPCR | 047F49h |
| | CAN0 transmit FIFO pointer control register | C0TFPCR | 047F4Bh |
| | CAN1 receive FIFO pointer control register | C1RFPCR | 047B49h |
| | CAN1 transmit FIFO pointer control register | C1TFPCR | 047B4Bh |

Note:
1. The register has write-only bits in one-shot timer mode and pulse-width modulation mode.

**Table 29.2 Read-Modify-Write Instructions**

| Function | Mnemonic |
|---|---|
| Transfer | MOV*Dir* |
| Bit processing | BCLR, BM*Cnd*, BNOT, BSET, BTSTC, and BTSTS |
| Shifting | ROLC, RORC, ROT, SHA, and SHL |
| Arithmetic operation | ABS, ADC, ADCF, ADD, ADSF, DEC, DIV, DIVU, DIVX, EXTS, EXTZ, INC, MUL, MULU, NEG, SBB, and SUB |
| Decimal operation | DADC, DADD, DSBB, and DSUB |
| Floating-point operation | ADDF, DIVF, MULF, and SUBF |
| Logical operation | AND, NOT, OR, and XOR |

## 29.3 Notes on Clock Generator

### 29.3.1 Sub Clock

#### 29.3.1.1 Oscillation Parameter Matching

The constant matching of sub clock oscillator should be evaluated in both cases when the drive power is high and low.
Contact your oscillator manufacturer for details on the oscillation circuit constant matching.

### 29.3.2 Power Control

Do not switch the base clock source until the oscillation of the clock to be used has stabilized. However, this does not apply to the on-chip oscillator since the on-chip oscillator starts running immediately after the CM31 bit in the CM3 register is set to 1.
To switch the base clock source from PLL clock to a low speed clock, that is, to set the BCS bit in the CCR register to 1, use either the MOV.L or OR.L instruction.

- Program example in assembly language

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

- Program example in C language

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

#### 29.3.2.1 Stop Mode

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

#### 29.3.2.2 Suggestions to Power Saving

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

- I/O pins:
  If inputs are floating, both transistors may be conducting. Set unassigned pins to input mode and connect each of them to VSS via a resistor, or set them to output mode and leave them open.
- A/D converter:
  When the A/D conversion is not performed, set the VCUT bit in the AD0CON1 register to 0 (VREF disconnected). To perform the A/D conversion, set the VCUT bit to 1 (VREF connected) and wait 1 µs or more for the operation.
- D/A converter:
  When the D/A conversion is not performed, set the DAiE bit in the DACON register (i = 0, 1) to 0 (output disabled) and the DAi register to 00h.
- Peripheral clock stop
  When entering wait mode, power consumption can be reduced by setting the CM02 bit in the CM0 register to 1 to stop peripheral clock source. However, the fC32 does not stop by the CM02 bit setting.

## 29.4 Notes on Bus

### 29.4.1 Notes on System Designing

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

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

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

### 29.4.2 Notes on Register Settings

#### 29.4.2.1 Chip Select Boundary Select Registers

When using single-chip mode exclusively, do not change values after a reset for registers CB01, CB12, and CB23.

When the CPU operation is performed in memory expansion mode more than once, set a value within the specified range to all of these registers irrespective of the use of them.

#### 29.4.2.2 External Bus Control Registers

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

## 29.5 Notes on Interrupts

### 29.5.1 ISP Setting

The interrupt stack pointer (ISP) is initialized to 00000000h after a reset. Set a value to the ISP before an interrupt is accepted, otherwise the program may go out of control. A multiple of 4 should be set to the ISP, which enables faster interrupt sequence due to less memory access.
For the use of NMI, in particular, since this interrupt cannot be disabled, the PM24 bit in the PM2 register should be set to 1 (NMI enabled) after the ISP is set at the beginning of program.

### 29.5.2 NMI

- The NMI cannot be disabled once the PM24 bit in the PM2 register is set to 1 (NMI enabled). This bit setting should be done only for the use of NMI.

- When the PM24 bit in the PM2 register is set to 1 (NMI enabled), the P8_5 bit in the P8 register is enabled just for monitoring the $\overline{\text{NMI}}$ pin state. It is not enabled as a general port.

### 29.5.3 External Interrupt

- The input signal to the $\overline{\text{INTi}}$ pin (i = 0 to 8) requires the pulse width specified by the electrical characteristics. If a pulse width is narrower than the specification, the external interrupt may not be accepted.

- When the effective level and/or edge of $\overline{\text{INTi}}$ pin (i = 0 to 8) are/is changed by the following bits: bits POL and/or LVS in the INTiIC register, the IFSR0i bit (i = 0 to 5) in the IFSR0 register, and/or the IFSR1j bit (j = i - 6; i = 6 to 8) in the IFSR1 register, the corresponding IR bit may become 1 (interrupt requested). When setting the above mentioned bits, preset bits ILVL2 to ILVL0 in the INTiIC register to 000b (interrupt disabled). After setting the above mentioned bits, set the corresponding IR bit to 0 (no interrupt requested), then set bits ILVL2 to ILVL0.

- When the effective level and/or edge of $\overline{\text{INTi}}$ pin (i = 6 to 8) are/is changed by the following bits: bits POL and/or LVS in the INTiIC register, and/or the IFSR1j bit (j = i - 6) in the IFSR1 register, the INTiR bit in the IIOkIR register (k = 9 to 11) may become 1 (ineterrupt requested). When setting the above mentioned bits, preset the INTiE bit in the IIOkIE register to 0 (interrupt disabled). After setting the above mentioned bits, set the corresponding INTiR bit to 0 (no interrupt requested), then set the INTiE bit to 1.

## 29.6 Notes on DMAC

### 29.6.1 DMAC-associated Register Settings

- Set the DMAC-associated registers while bits MDi1 and MDi0 (i = 0 to 3) in the DMDi register are 00b (DMA transfer disabled). Then, set bits MDi1 and MDi0 to 01b (single transfer) or 11b (repeat transfer) at the end of the setup procedure. This procedure is also applied to rewriting bits UDAi, USAi, and BWi1 and BWi0 in the DMDi register.

- In case the DMAC-associated registers are to be rewritten while DMA transfer is enabled, disable the peripheral function as DMA request source so that no DMA transfer request is generated, then set bits MDi1 and MDi0 in the DMDi register of the corresponding channel to 00b (DMA transfer disabled).

- Once a DMA transfer request is accepted, DMA transfer cannot be disabled even if setting bits MDi1 and MDi0 in the DMDi register to 00b (DMA transfer disabled). Do not change the settings of any DMAC-associated registers other than bits MDi1 and MDi0 until the DMA transfer is completed.

- Wait six or more peripheral clocks to set bits MDi1 and MDi0 in the DMDi register to 01b (single transfer) or 11b (repeat transfer) after setting registers DMiSL and DMiSL2.

### 29.6.2 Read from DMAC-associated Registers

- To sequentially read respective registers DMiSL and DMiSL2, follow the reading order as below:
  DM0SL, DM1SL, DM2SL, and DM3SL
  DM0SL2, DM1SL2, DM2SL2, and DM3SL2

## 29.7 Notes on Timers

### 29.7.1 Timer A and Timer B

All timers are stopped after a reset. To restart timers, configure parameters such as operating mode, count source, and counter value, then set the TAiS bit (i = 0 to 4) or TBjS bit (j = 0 to 5) in the TABSR or TBSR register to 1 (count starts).

The following registers and bits should be set while the TAiS bit or TBjS bit is 0 (count stops):

- Registers TAiMR and TBjMR
- The UDF register
- Bits TAZIE, TA0TGL, and TA0TGH in the ONSF register
- The TRGSR register

### 29.7.2 Timer A

#### 29.7.2.1 Timer Mode

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

#### 29.7.2.2 Event Counter Mode

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

#### 29.7.2.3 One-shot Timer Mode

- If the TAiS bit in the TABSR register is set to 0 (count stops) while the timer counter is running, the following operations are performed:
  - The timer counter stops and the setting value of the TAi register is reloaded.
  - A low signal is output at the TAiOUT pin.
  - The IR bit in the TAiIC register becomes 1 (interrupts requested) after one CPU clock cycle.

- One-shot timer is operated by an internal count source. When the trigger is an input to the TAiIN pin, the signal is output with a maximum of one count source clock delay after a trigger input to the TAiIN pin.

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt, set the IR bit to 0 after one of the settings below is done:
  - Select one-shot timer mode after a reset
  - Switch the operating mode from timer mode to one-shot timer mode, or
  - Switch the operating mode from event counter mode to one-shot timer mode

- If a retrigger occurs while counting, the timer counter decrements by one, reloads the setting value of the TAi register, and then continues counting. To generate a retrigger while counting, wait one or more count source cycles after the last trigger is generated.

- When an external trigger input is selected to start counting in timer A one-shot mode, do not provide an external retrigger for 300 ns before the timer counter reaches 0000h. Otherwise, it may stop counting.

#### 29.7.2.4 Pulse-width Modulation Mode

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt (i = 0 to 4), set the IR bit to 0 after one of the settings below is done:
  - Select pulse-width modulation mode after a reset
  - Switch the operating mode from timer mode to pulse-width modulation mode, or
  - Switch the operating mode from event counter mode to pulse-width modulation mode

- If the TAiS bit in the TABSR register is set to 0 (count stops) while PWM pulse is output, the following operations are performed:
  - The timer counter stops.
  - The output level at the TAiOUT pin changes from high to low. The IR bit becomes 1.
  - When a low signal is output at the TAiOUT pin, it remains unchanged. The IR bit does not change, either.

## 29.7.3 Timer B

### 29.7.3.1 Timer Mode and Event Counter Mode

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

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

- To set the MR3 bit in the TBjMR register to 0 (no overflow), wait one or more count source cycles to write to the TBjMR register after the MR3 bit becomes 1 (overflow), while the TBjS bit is set to 1 (count starts).

- Use the IR bit in the TBjIC register to detect overflow. The MR3 bit is used only to determine an interrupt request source within the interrupt handler.

- The counter value is undefined when the timer counter starts. Therefore, the timer counter may overflow before a pulse to be measured is applied on the initial valid edge and cause a timer Bj interrupt request to be generated.

- When the pulse to be measured is applied on the initial valid edge after the timer counter starts, an undefined value is transferred to the reload register. At this time, the timer Bj interrupt request is not generated.

- The IR bit may become 1 (interrupt requested) by changing bits MR1 and MR0 in the TBjMR register after the timer counter starts. However, if the same value is rewritten to bits MR1 and MR0, the IR bit is not changed.

- Pulse width is repeatedly measured in pulse-width measure mode. Whether the measurement result is high-level width or not is determined by a program.

- If an overflow occurs simultaneously when a pulse is applied on the valid edge, this pulse is not recognized since an interrupt request is generated only once. Do not let an overflow occur in pulse period measure mode.

- In pulse-width measure mode, determine whether an interrupt source is a pulse applied on the valid edge or an overflow by reading the port level in the TBj interrupt handler.

## 29.8 Notes on Three-phase Motor Control Timers

### 29.8.1 Shutdown

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

### 29.8.2 Register setting

- Do not write to the TAi1 register (i = 1, 2, 4) in the timing that timer B2 overflows. Before writing to the TAi1 register, read the TB2 register to verify that sufficient time is left until timer B2 overflows. Then, immediately write to the TAi1 register so that no interrupt handler is performed during this write procedure. If the TB2 register indicates little time is left until the overflow, write to the TAi1 register after timer B2 overflows.

## 29.9 Notes on Serial Interface

### 29.9.1 Changing the UiBRG Register (i = 0 to 8)

- Set the UiBRG register after setting bits CLK1 and CLK0 in the UiC0 register. When these bits are changed, the UiBRG register must be set again.
- If a clock is input immediately after the UiBRG register is set to 00h, the counter reaches FFh. In this case, it requires an extra 256 clocks to reload 00h into the register. Once the 00h is reloaded, the counter performs the operation without dividing the count source according to the setting.

### 29.9.2 Synchronous Serial Interface Mode

#### 29.9.2.1 Selecting an External Clock

- If an external clock is selected, the following conditions must be met while the external clock is held high when the CKPOL bit in the UiC0 register (i = 0 to 8) is set to 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge), or while the external clock is held low when the CKPOL bit is set to 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge):
  - The TE bit in the UiC1 register is set to 1 (transmission enabled)
  - The RE bit in the UiC1 register is set to 1 (reception enabled)
  - The TI bit in the UiC1 register is set to 0 (data held in the UiTB register)
  - The RE bit setting is not required in transmit operation only.

#### 29.9.2.2 Receive Operation

- In synchronous serial interface mode, the transmit/receive clock is controlled by the transmit control circuit. Set the UARTi-associated registers (i = 0 to 8) for a transmit operation, even if the MCU is used only for receive operation. Dummy data is output from the TXDi pin while receiving if the TXDi pin is set to output mode.

- If data is received continuously, an overrun error occurs when the RI bit in the UiC1 register is 1 (data held in the UiRB register) and the seventh bit of the next data is received in the UARTi receive shift register. Then, the OER bit in the UiRB register becomes 1 (overrun error occurred). In this case, the UiRB register becomes undefined. If an overrun error occurs, the IR bit in the SiRIC register is not changed to 1.

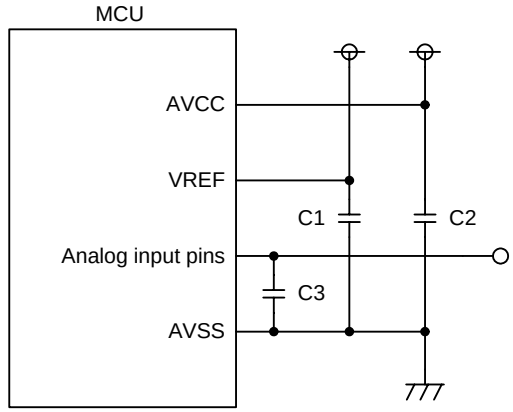
### 29.9.3 Special Mode 1 ($I^2C$ Mode)

- To generate a start condition, stop condition, or restart condition, set the STSPSEL bit in the UiSMR4 register (i = 0 to 6) to 0. Then, wait a half or more clock cycles of the transmit/receive clock to change the respective condition generate bit (the STAREQ, RSTAREQ, or STPREQ bit) from 0 to 1.

## 29.10 Notes on A/D Converter

### 29.10.1 Notes on Designing Boards

• Three capacitors should be respectively placed between the AVSS pin and such pins as AVCC, VREF, and analog inputs (AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, and AN15_0 to AN15_7) to avoid error operations caused by noise or latchup, and to reduce conversion errors. Figure 29.1 shows an example of pin configuration for A/D converter.




Notes:
1. C1 ≥ 0.47µF, C2 ≥ 0.47µF, and C3 ≥ 100 pF (reference values)
2. The traces for the capacitor and the MCU should be as short and wide as physically possible.

**Figure 29.1 Pin Configuration for A/D Converter**

• Do not use any of the four pins AN_4 to AN_7 for analog input if the key input interrupt is to be used. Otherwise, a key input interrupt request occurs when the A/D input voltage becomes VIL or lower.

• When AVCC = VREF = VCC, A/D input voltage for pins AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, and ANEX1 should be VCC or lower.

### 29.10.2 Notes on Programming

- The following registers should be written while the A/D conversion is stopped, that is, before a trigger occurs: AD0CON0 (except the ADST bit), AD0CON1, AD0CON2, AD0CON3, and AD0CON4.

- If the VCUT bit in the AD0CON1 register is switched from 0 (VREF connected) to 1 (VREF disconnected), the A/D conversion should be started after 1 µs or more. Set the VCUT bit to 0 when A/D conversion is not used to reduce power consumption.

- Set the port direction bit for the pin to be used as an analog input pin to 0 (input). Set the ASEL bit of the corresponding port function select register to 1 (the port is used as A/D input).

- If the TRG bit in the AD0CON0 register is set to 1 (external trigger or hardware trigger is selected), set the corresponding port direction bit (PD9_7 bit) for the ADTRG pin to 0 (input).

- The ϕAD frequency should be 16 MHz or below when VCC is 4.2 to 5.5 V, and 10 MHz or below when VCC is 3.0 to 4.2 V. It should be 1 MHz or above if the sample and hold function is enabled. If not, it should be 250 kHz or above.

- If A/D operating mode (bits MD1 and MD0 in the AD0CON0 register or the MD2 bit in the AD0CON1 register) has been changed, re-select analog input pins by using bits CH2 to CH0 in the AD0CON0 register or bits SCAN1 and SCAN0 in the AD0CON1 register.

- If the AD0i register (i = 0 to 7) is read when the A/D conversion result is stored to the register, the stored value may have an error. Read the AD0i register after the A/D conversion has been completed.
  In one-shot mode or single sweep mode, read the respective AD0i register after the IR bit in the AD0IC register has become 1 (interrupt requested).
  In repeat mode, repeat sweep mode 0, or repeat sweep mode 1, an interrupt request can be generated each time when an A/D conversion has been completed if the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode enabled). Similar to the other modes above, read the AD00 register after the IR bit in the AD0IC register has become 1 (interrupt requested).

- If the A/D conversion in progress is halted by setting the ADST bit in the AD0CON0 register to 0, the conversion result is undefined. In addition, the unconverted AD0i register may also become undefined. Consequently, the AD0i register should not be used just after A/D conversion is halted.

- The external trigger cannot be used in DMAC operating mode. Do not read the AD00 register by a program.

- If, in single sweep mode, the A/D conversion in progress is halted by setting the ADST bit in the AD0CON0 register to 0 (A/D conversion is stopped), an interrupt request may be generated even though the sweep is not completed. To halt the A/D conversion, first disable interrupts, then set the ADST bit to 0.

## 29.11 Notes on Flash Memory Rewriting

### 29.11.1 Note on Power Supply

- Keep the supply voltage constant within the range specified in the electrical characteristics while a rewrite operation on flash memory is in progress. If the supply voltage becomes beyond the guaranteed value, the device cannot be guaranteed.

### 29.11.2 Note on Hardware Reset

- Do not perform a hardware reset while a rewrite operation on flash memory is in progress.

### 29.11.3 Note on Flash Memory Protection

- If an ID code written in an assigned address has an error, any read/write operation of flash memory in standard serial I/O mode is disabled.

### 29.11.4 Notes on Programming

- Do not set the FEW bit in the FMCR register to 1 (CPU rewrite mode) in low speed mode or low power mode.
- Four software commands of program, block erase, lock bit program, and protect bit program are interrupted by an NMI, a watchdog timer interrupt, an oscillator stop detection interrupt, or a low voltage detection interrupt. If any of the software commands above is interrupted, erase the corresponding block and then execute the same command again. If the block erase command is interrupted, values of lock bits and protect bits become undefined. Therefore, disable the lock bit, and then execute the block erase command again.

### 29.11.5 Notes on Interrupts

- EW0 mode
  - To use interrupts assigned to the relocatable vector table, the vector table should be addressed in RAM space.
  - If either of an NMI, a watchdog timer interrupt, an oscillator stop detection interrupt, or a low voltage detection interrupt is generated, the flash memory module automatically enters read array mode. Therefore these interrupts are enabled even during a rewrite operation. On the other hand, the rewrite operation in progress is aborted by the interrupt and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.
  - Instructions BRK, INTO, and UND, which refer to data on the flash memory, are unavailable in this mode.

- EW1 mode
  - Interrupts assigned to the relocatable vector table should not be accepted during a program/erase operation.
  - The watchdog timer interrupt should not be generated, either.
  - If either of an NMI, a watchdog timer interrupt, an oscillator stop detection interrupt, or a low voltage detection interrupt is generated, the flash memory module automatically enters read array mode. Therefore this interrupt is enabled even during a rewrite operation. On the other hand, the rewrite operation in progress is aborted by the interrupt and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the EWM bit in the FMR0 register to 1 (set as EW1 mode) and the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.

### 29.11.6 Notes on Rewrite Control Program

- EW0 mode
  - If the supply voltage lowers during the rewrite operation of blocks having the rewrite control program, the rewrite control program may not be successfully rewritten, then the rewrite operation it self may not be performed. In this case perform the rewrite operation by serial programmer or parallel programmer.

- EW1 mode
  - Do not rewrite blocks having the rewrite control program.

### 29.11.7 Notes on Number of Programming/Erasure and Software Command Execution Time

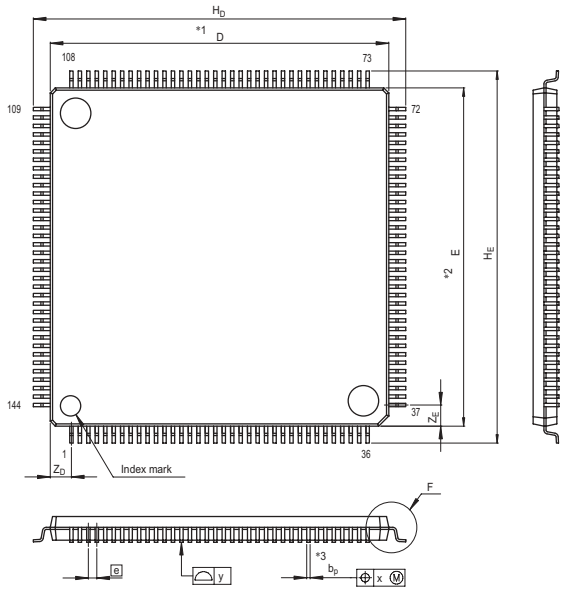
- According to the increase of program/erase operation, the four software commands: program, block erase, lock bit program, and protect bit program require more time to be executed. If the number of programming/erasure exceeds the minimum endurance value specified in the electrical characteristics, it may take unpredictable time to execute the software commands. The waiting time for the execution of software commands should be set much longer than the execution time specified in the electrical characteristics.

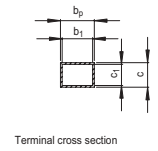
### 29.11.8 Other Notes

- The required time to perform program/erase operation specified in the electrical characteristics can be guaranteed within the minimum values of programming/erasure endurance specified in the same table. Even if the number of programming/erasure exceeds the minimum endurance value, the program/erase operation may be unguaranteedly performed.
- Chips repeatedly programmed and erased for debugging are not allowed to be used for commercial products.

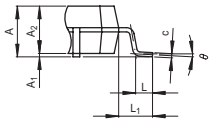
# Appendix 1. Package Dimensions

| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-LQFP144-20x20-0.50 | PLQP0144KA-A | 144P6Q-A / FP-144L / FP-144LV | 1.2g |




Terminal cross section

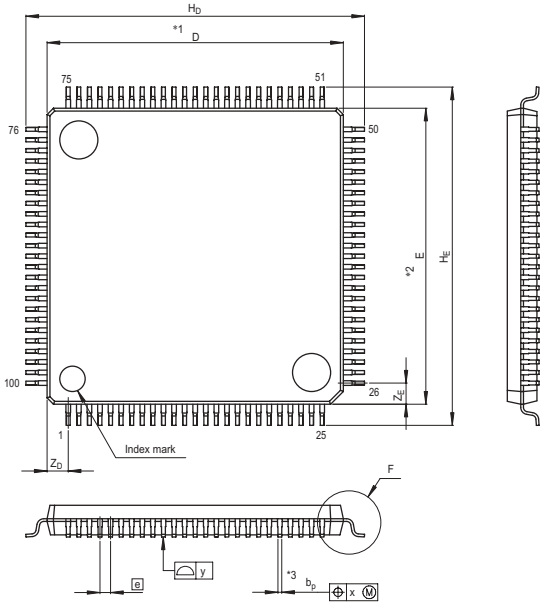


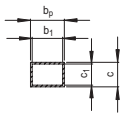

Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

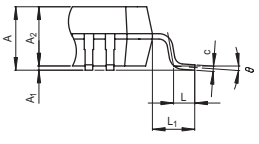
| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 19.9 | 20.0 | 20.1 |
| E | 19.9 | 20.0 | 20.1 |
| $A_2$ | — | 1.4 | — |
| $H_D$ | 21.8 | 22.0 | 22.2 |
| $H_E$ | 21.8 | 22.0 | 22.2 |
| A | — | — | 1.7 |
| $A_1$ | 0.05 | 0.1 | 0.15 |
| $b_p$ | 0.17 | 0.22 | 0.27 |
| $b_1$ | — | 0.20 | — |
| c | 0.09 | 0.145 | 0.20 |
| $c_1$ | — | 0.125 | — |
| θ | 0° | — | 8° |
| e | — | 0.5 | — |
| x | — | — | 0.08 |
| y | — | — | 0.10 |
| $Z_D$ | — | 1.25 | — |
| $Z_E$ | — | 1.25 | — |
| L | 0.35 | 0.5 | 0.65 |
| $L_1$ | — | 1.0 | — |

| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-LQFP100-14x14-0.50 | PLQP0100KB-A | 100P6Q-A / FP-100U / FP-100UV | 0.6g |




Terminal cross section




Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 13.9 | 14.0 | 14.1 |
| E | 13.9 | 14.0 | 14.1 |
| $A_2$ | — | 1.4 | — |
| $H_D$ | 15.8 | 16.0 | 16.2 |
| $H_E$ | 15.8 | 16.0 | 16.2 |
| A | — | — | 1.7 |
| $A_1$ | 0.05 | 0.1 | 0.15 |
| $b_p$ | 0.15 | 0.20 | 0.25 |
| $b_1$ | — | 0.18 | — |
| c | 0.09 | 0.145 | 0.20 |
| $c_1$ | — | 0.125 | — |
| θ | 0° | — | 8° |
| e | — | 0.5 | — |
| x | — | — | 0.08 |
| y | — | — | 0.08 |
| $Z_D$ | — | 1.0 | — |
| $Z_E$ | — | 1.0 | — |
| L | 0.35 | 0.5 | 0.65 |
| $L_1$ | — | 1.0 | — |

# INDEX

## A

A0 ..... 24
A1 ..... 24
A2 ..... 24
A3 ..... 24
AD00 to AD07 ..... 318
AD0CON0 ..... 314
AD0CON1 ..... 315
AD0CON2 ..... 316
AD0CON3 ..... 317
AD0CON4 ..... 318
AD0IC ..... 164
Address Register ..... 24

## B

B Flag ..... 25
BCN0IC to BCN6IC ..... 164
BRK Instruction Interrupt ..... 155
BRK2 Instruction Interrupt ..... 155
BTSR ..... 349

## C

C Flag ..... 24
C0AFSR, C1AFSR ..... 457
C0BCR, C1BCR ..... 428
C0CLKR, C1CLKR ..... 427
C0CSSR, C1CSSR ..... 456
C0CTLR, C1CTLR ..... 423
C0ECSR, C1ECSR ..... 465
C0EIC, C1EIC ..... 164
C0EIER, C1EIER ..... 458
C0EIFR, C1EIFR ..... 460
C0FIDCR0, C0FIDCR1 ..... 431
C0FRIC, C1FRIC ..... 164
C0FTIC, C1FTIC ..... 164
C0MB0 to C0MB31 ..... 435
C0MCTL0 to C0MCTL31 ..... 439
C0MIER, C1MIER ..... 438
C0MKIVLR, C1MKIVLR ..... 433
C0MKR0 to C0MKR7 ..... 430
C0MSMR, C1MSMR ..... 453
C0MSSR, C1MSSR ..... 454
C0RECR, C1RECR ..... 463
C0RFCR, C1RFCR ..... 443
C0RFPCR, C1RFPCR ..... 446
C0RIC, C1RIC ..... 164
C0STR, C1STR ..... 450
C0TCR, C1TCR ..... 468
C0TECR, C1TECR ..... 464
C0TFCR, C1TFCR ..... 447
C0TFPCR, C1TFPCR ..... 449
C0TIC, C1TIC ..... 164
C0TSR, C1TSR ..... 467
C0WIC, C1WIC ..... 164
C1FIDCR0, C1FIDCR1 ..... 431
C1MB0 to C1MB31 ..... 435
C1MCTL0 to C1MCTL31 ..... 439
C1MKR0 to C1MKR7 ..... 430
Carry Flag ..... 24
CB01 ..... 131
CB12 ..... 132
CB23 ..... 132
CCR ..... 98
CM0 ..... 99
CM1 ..... 100
CM2 ..... 101
CM3 ..... 101
CPSRF ..... 102
CRCD ..... 334
CRCIN ..... 335
CSOP0 ..... 130
CSOP1 ..... 130
CSOP2 ..... 131

## D

D Flag ..... 24
DA0, DA1 ..... 333
DACON ..... 333
Data Register ..... 24
DCR0 to DCR3 ..... 26, 189
DCT0 to DCT3 ..... 26, 188
DDA0 to DDA3 ..... 26, 190
DDR0 to DDR3 ..... 26, 190
Debug Flag ..... 24
DM0IC to DM3IC ..... 164
DM0SL to DM3SL ..... 184
DM0SL2 to DM3SL2 ..... 185
DMA Destination Address Register ..... 26
DMA Destination Address Reload Register 26
DMA Mode Register ..... 26
DMA Source Address Register ..... 26
DMA Source Address Reload Register .... 26
DMA Terminal Count Register ..... 26
DMA Terminal Count Reload Register ..... 26
DMD0 to DMD3 ..... 26, 188
DP Bit ..... 25
DSA0 to DSA3 ..... 26, 189

DSR0 to DSR3 ..... 26, 190
DTT ..... 256
DVCR ..... 90

## E

EBC0 to EBC3 ..... 135

## F

Fast Interrupt ..... 157
FB ..... 24
FBPM0 ..... 532
FBPM1 ..... 532
FBPM2 ..... 533
FEBC0, FEBC3 ..... 529
Fixed-Point Designation Bit ..... 25
Flag Register ..... 24
FLG ..... 24
Floating-Point Overflow Flag ..... 25
Floating-Point Rounding Mode ..... 25
Floating-Point Underflow Flag ..... 25
FMCR ..... 528
FMR0 ..... 530
FMR1 ..... 531
FMSR0 ..... 531
FO Flag ..... 25
FPR0 ..... 530
Frame Base Register ..... 24
FU Flag ..... 25

## G

G0BCR0 to G2BCR0 ..... 346
G0BCR1, G1BCR1 ..... 347
G0BT to G2BT ..... 345
G0FE to G2FE ..... 355
G0FS, G1FS ..... 354
G0PO0 to G0PO7 ..... 354
G0POCR0 to G0POCR7 ..... 352
G0TM0 to G0TM7 ..... 351
G0TMCR0 to G0TMCR7 ..... 350
G0TPR6, G0TPR7 ..... 350
G1PO0 to G1PO7 ..... 354
G1POCR0 to G1POCR7 ..... 352
G1TM0 to G1TM7 ..... 351
G1TMCR0 to G1TMCR7 ..... 350
G1TPR6, G1TPR7 ..... 350
G2BCR1 ..... 348
G2CR ..... 382
G2MR ..... 381
G2PO0 to G2PO7 ..... 354
G2POCR0 to G2POCR7 ..... 353
G2RB ..... 381
G2RTP ..... 355
G2TB ..... 380

## H

Hardware Interrupt ..... 156

## I

I Flag ..... 25
I2CCCR ..... 394
I2CCR0 ..... 392
I2CCR1 ..... 397
I2CCR2 ..... 400
I2CIC ..... 164
I2CLIC ..... 164
I2CMR ..... 406
I2CSAR ..... 391
I2CSR ..... 402
I2CSSCR ..... 396
I2CTRSR ..... 390
ICTB2 ..... 248
IDB0, IDB1 ..... 247
IEAR ..... 383
IECR ..... 382
IERIF ..... 384
IETIF ..... 383
IFS0 ..... 510
IFS1 ..... 511
IFS2 ..... 512
IFS3 ..... 513
IFSR0 ..... 173, 275
IFSR1 ..... 174, 276
IIO0IC to IIO11IC ..... 164
IIO0IE to IIO11IE ..... 178
IIO0IR to IIO11IR ..... 177
INT Instruction Interrupt ..... 155
INT0IC to INT8IC ..... 165
INTB ..... 24
Interrupt Control Register ..... 164
Interrupt Enable Flag ..... 25
Interrupt request level ..... 165
Interrupt Response Time ..... 169
Interrupt Sequence ..... 168
Interrupt Stack Pointer ..... 24
Interrupt Types ..... 154
Interrupt Vector Table Base Register ..... 24
INVC0 ..... 244
INVC1 ..... 245

IOBC ..... 246
IPL ..... 25, 163
ISP ..... 24

## K

KUPIC ..... 164

## L

Low Voltage Detection Interrupt ..... 91, 156
Low Voltage Detector ..... 88
LVDC ..... 89

## M

Maskable Interrupt ..... 154

## N

NMI (Non Maskable Interrupt) ..... 156
Non-maskable Interrupt ..... 154

## O

O Flag ..... 25
ONSF ..... 215
Oscillator Stop Detection Interrupt ..... 156
Overflow Flag ..... 25
Overflow Interrupt ..... 155

## P

P0 to P15 ..... 208
P0_0S to P0_7S ..... 492
P10_0S to P10_7S ..... 503
P1_0S to P1_7S ..... 493
P11_0S to P11_4S ..... 504
P12_0S to P12_7S ..... 505
P13_0S to P13_7S ..... 506
P14_3S to P14_6S ..... 507
P15_0S to P15_7S ..... 508
P2_0S to P2_7S ..... 494
P3_0S to P3_7S ..... 495
P4_0S to P4_7S ..... 496
P5_0S to P5_7S ..... 497
P6_0S to P6_7S ..... 498
P7_0S to P7_7S ..... 499
P8_0S to P8_4S, P8_6S, P8_7S ..... 500
P9_0S to P9_7S (144-pin package) ..... 501
P9_3S to P9_7S (100-pin package) ..... 502
PBC ..... 127
PC ..... 24
PCR ..... 517
PD0 to PD15 ..... 490
Peripheral Interrupt ..... 156
PLC0 ..... 107
PLC1 ..... 108
PM0 ..... 94
PM2 ..... 103
PM3 ..... 104
PRCR ..... 151
PRCR2 ..... 152
PRCR3 ..... 152
Processor Interrupt Priority Level ..... 25, 163
Program Counter ..... 24
PRR ..... 153
PUR0 ..... 514
PUR1 ..... 515
PUR2 ..... 515
PUR3 ..... 516
PUR4 ..... 516

## R

R2R0 ..... 24
R3R1 ..... 24
R6R4 ..... 24
R7R5 ..... 24
Register Bank Select Flag ..... 25
Register Saving ..... 170
RIPL1, RIPL2 ..... 167, 198
RND ..... 25

## S

S Flag ..... 24
S0RIC to S8RIC ..... 164
S0TIC to S8TIC ..... 164
Save Flag Register ..... 26
Save PC Register ..... 26
SB ..... 24
Sign Flag ..... 24
Single-step Interrupt ..... 156
Software Interrupt ..... 155
SP ..... 24
Special Interrupt ..... 156
Stack Pointer ..... 24
Stack Pointer Select Flag ..... 25
Static Base Register ..... 24
SVF ..... 26
SVP ..... 26

## T

TA0 to TA4 ..... 212
TA0IC to TA4IC ..... 164
TA0MR to TA4MR .. 213, 219, 222, 225, 227
TA1, TA2, TA4, TA11, TA21, TA41 ..... 252
TA1MR, TA2MR, TA4MR ..... 253
TABSR ..... 213, 231, 254
TB0 to TB5 ..... 230
TB0IC to TB5IC ..... 164
TB0MR to TB5MR ..... 230, 233, 235, 237
TB2 ..... 250
TB2MR ..... 250
TB2SC ..... 251
TBSR ..... 231
TCSPR ..... 102, 217
TRGSR ..... 216, 254

## U

U Flag ..... 25
U0BRG to U8BRG ..... 272
U0C0 to U6C0 ..... 265
U0C1 to U6C1 ..... 267
U0MR to U6MR ..... 263
U0RB to U6RB ..... 273
U0SMR to U6SMR ..... 269
U0SMR2 to U6SMR2 ..... 270
U0SMR3 to U6SMR3 ..... 271
U0SMR4 to U6SMR4 ..... 272
U0TB to U8TB ..... 273
U78CON ..... 268
U7C0, U8C0 ..... 266
U7C1, U8C1 ..... 267
U7MR, U8MR ..... 264
U7RB, U8RB ..... 274
UDF ..... 214
Undefined Instruction Interrupt ..... 155
User Stack Pointer ..... 24
USP ..... 24

## V

VCT ..... 26
Vector Register ..... 26
VRCR ..... 86

## W

Watchdog Timer Interrupt ..... 156
WDC ..... 181
WDTS ..... 181

## X

X0R to X15R ..... 337
XYC ..... 337

## Y

Y0R to Y15R ..... 338

## Z

Z Flag ..... 24
Zero Flag ..... 24

# REVISION HISTORY — R32C/118 Group Hardware Manual

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 0.62 | Apr 08, 2009 | — | Initial release |
| 1.00 | Nov 24, 2009 | — | Second edition released |
| | | — | This manual in general<br>• Changed the following expressions: “Multi-master I²C-bus interface”, to “I²C-bus interface”, “Multi-master I²C-bus line”, to “I²C-bus line” (under Chapters 11, 13, and 24); “start/stop condition”, to “start condition/stop condition” (under Chapters 4, 11, 13, 18, and 24); “Pins/ports/bits/registers xxx, xxx, and xxx are provided in the xx-pin package”, to “Pins/ports/bits/registers xxx, xxx, and xxx are available in the xx-pin package” (under Chapters 5, 9, 15, 16, 19, 23, 26, and 28); "reset operation”, to “reset” (under Chapters 4, 7-9, 11, 12, 25, and 27)<br>• Modified the following descriptions: “multimaster I²C-bus interface”, to “multi-master I²C-bus interface” (under Chapters 1 and 26)“; “This register should be rewritten after (the xxx bit in) the xxx register is set to 1/AAh/00b ((re)write enabled).”, to “Set (the xxx bit in) the xxx register to 1/AAh/00b (write enabled) before rewriting this register.” (under Chapters 6, 8, 9, 13, 18, 24, and 27) |
| | | | **About This Manual** |
| | | — | • Corrected typos “Hardware Manual” and “characteristics)” in **1. Purpose and Target User**, to “Hardware” and “characteristics”, respectively<br>• Made major text modifications to **2. Numbers and Symbols**<br>• Revised the illustration in **3. Registers** |
| | | | **Chapter 1. Overview** |
| | | 1 | • Modified description for **1.1.1** |
| | | 2, 4 | • Modified description for “External Bus Expansion” in **Tables 1.1 and 1.3**; Moved this unit below “Clock” |
| | | 3, 5 | • Modified description for “Flash memory” in **Tables 1.2 and 1.4** |
| | | 5 | • Modified the position of note symbol (1) in **Table 1.4**<br>• Modified description “32-slot message buffer” for “CAN Module” in **Table 1.4**, to “32 mailboxes” |
| | | 6 | • Completed all “under development” products in **Table 1.5** |
| | | 9, 14 | • Corrected a typo “R5_3” for pin No. 62 in **Figure 1.3** and for pin No. 41 in **Figure 1.4**, to “P5_3” |
| | | | **Chapter 2. CPU** |
| | | 25 | • Modified the second sentence of **2.1.8.8** descriptively |
| | | | **Chapter 4. SFRs** |
| | | 28 | • Changed hexadecimal format of reset values for registers CCR and FMCR in **Table 4.1**, to binary |
| | | 37 | • Changed reset values “XXXX XXXXb” and “XXXX 000Xb” for registers U7RB and U8RB in **Table 4.10**, to “XXXXh” |
| | | 38 | • Changed expression of register name “Xi Register Yi Register” (i = 0 to 15) and register symbol “XiR, YiR” in **Table 4.11**, to “Xi Register/ Yi Register” and “XiR/YiR”, respectively |

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| | | 66, 80 | • Changed reset values for CiCLKR in **Tables 4.39 and 4.53** from "00h", to "000X 0000b" |
| | | | **Chapter 5. Resets** |
| | | 85 | • Corrected a typo "pultiple" in line 2 of **5.4**, to "multiple" |
| | | | **Chapter 6. Power Management** |
| | | — | • Made minor text modifications to this chapter |
| | | | **Chapter 7. Processor Mode** |
| | | — | • Made minor text modification to this chapter |
| | | | **Chapter 8. Clock** |
| | | — | • Made minor text modifications to this chapter |
| | | 107 | • Added "in the PLC1 register" to "SEO bit" in **Figure 8.13** |
| | | 107, 108 | • Added description to Note 1 for registers PLC0 and PLC1 in **Figures 8.14 and 8.15**, respectively |
| | | 108, 111, 112 | • Deleted description associated with frequency from line 14 below **Figure 8.15**, line 2 of **8.3**, and line 2 of **8.4** |
| | | 115-117 | • Added description for the following bits: BCS, CM04, CM05, CM10, CM20, CM30, and CM31, to **Figures 8.17 to 8.19** |
| | | 119 | • Added description for procedure (6) to **8.7.2.2** |
| | | 121 | • Added $I^2C$-bus interface interrupt and $I^2C$-bus line interrupt to **Table 8.6** |
| | | 123 | • Moved previous Table 8.7 with one sentence above the table to 8.7.3.3 as **Table 8.8** |
| | | | • Added $I^2C$-bus line interrupt to **Table 8.8** |
| | | | **Chapter 9. Bus** |
| | | — | • Made minor text modifications to this chapter |
| | | 126 | • Deleted description for frequency and Note 1 in **Figure 9.1**; Modified description for peripheral data bus "16-bit", to "16-/32-bit" |
| | | 127 | • Modified peripheral bus width in line 1 of **9.2**, from "16-bit width" to "16-/32-bit width" |
| | | | • Deleted description for 00b of PRD4 to PRD0 and PWR4 to PWR0 in **Figure 9.2** |
| | | 130 | • Modified description for setting the P5_7B bit to 0 in **Figure 9.5**: "Output $\overline{RDY}$ from P5_7", to "$\overline{RDY}$ input pin" |
| | | 148 | • Deleted "(i = 0 to 3)" from **Figure 9.17** |
| | | | **Chapter 10. Protection** |
| | | — | • Made minor text modifications to this chapter |
| | | 151 | • Added "I2CMR" as a protected register for PRC1 bit, to **Table 10.1 and Figure 10.1**; Changed the order of registers for PRC1 and PRC2 |
| | | 152 | • Deleted "(i = 0 to 7)" from the title of **Table 10.2** |
| | | | **Chapter 11. Interrupts** |
| | | — | • Made minor text modifications to this chapter |
| | | 159-162 | • Added details to "Reference" in **Tables 11.2 to 11.5** |
| | | 172 | • Modified "Bits RLVL02 to RLVL00" and "Bits RLVL12 to RLVL10" in **Figure 11.8**, to "Bits RLVL2 to RLVL0 in the RIPL1 register" and "Bits RLVL2 to RLVL0 in the RIPL2 register", respectively |

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| | | 174<br>176<br>177, 178 | • Modified Note 1 for **11.11** descriptively<br>• Moved “(i = 0 to 11)” in **Figure 11.12** to the title<br>• Modified the following register names: ”Intelligent I/O Interrupt Request Register” in **Figure 11.13**, and “Intelligent I/O Interrupt Enable Register” in **Figure 11.14**, to “Intelligent I/O Interrupt Request Register i (i = 0 to 11)”, and “Intelligent I/O Interrupt Enable Register i (i = 0 to 11)”, respectively<br>• Changed variables “i”s, “j”s, and “k”s for description of bits in **Figures 11.13 and 11.14**, to “x”s, “y”s, and “z”s, respectively; Added expression “channel”, to descriptions for BTxR, TMxyR, POxyR, IEzR, BTxE, TMxyE, POxyE, and IEzE |
| | | | **Chapter 12. Watchdog Timer** |
| | | — | • Revised this chapter entirely<br>• Modified description “bus clock”s, to “peripheral bus clock”s |
| | | | **Chapter 13. DMAC** |
| | | —<br>184<br>187<br>188<br><br><br>196 | • Changed the following principle expressions: “transfer unit” to “transfer size”, “destination address” to “addressing mode”, “fixed” to “non-incrementing addressing”, “forward” to “incrementing addressing”<br>• Modified the following description: “registers DMiSL and DMiSL2” in line 3 of the paragraph above **Figure 13.2**, to “the DMiSL register, and in bits DSEL24 to DSEL20 in the DMiSL2 register”<br>• Modified description “the INTiIC register, IFSR0 register)” in Note 1 of **Table 13.3**, to “the INTiIC register and the IFSR1 register)”<br>• Changed bit names USAi and UDAi for DMDi register in **Figure 13.4** and their function descriptively<br>• Deleted the second sentence of Note 2 for DMDi register in **Figure 13.4**; Added Note 3<br>• Modified description for Note 2 in **Figure 13.5**; Deleted Note 3<br>• Modified description “channel i” in line 1 of the first bullet point of **13.4.1**, to “the DMDi register”; Added one sentence to the same bullet point; Deleted whole description of the second and third bullet points; Added two new paragraphs |
| | | | **Chapter 14. DMAC II** |
| | | —<br><br><br><br><br><br><br><br>197 | • Revised this chapter entirely<br>• Changed the following principle expressions: “transfer data” to “transfer type”, “transfer data unit” to “transfer size”, “transfer space” to “transfer memory space”, “transfer direction” to “addressing mode”, “fixed address” to “non-incrementing/constant address”, “forward address” to “incrementing address”, “end-of-transfer interrupt” to “DMA II transfer complete interrupt”, “transfer source address” to “source addressing”, and “transfer destination address” to “destination addressing”<br>• Corrected a typo “64 Kbyte-space” in **Table 14.1**, to “64-Mbyte space”<br>• Modified description “The relocatable vector table” in the forth bullet point of **14.1**, to “The relocatable vector” |

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| | | 201 | • Modified description for MOD in **Figure 14.3**; Divided the figure into two according to the MULT bit setting; Modified function of b14 to b8 from “No register bits”, to “Reserved” |
| | | 202 | • Moved and modified description below previous Figure 14.5, to lines 7 to 10 of **14.2** |
| | | 204 | • Modified description “CADR1 to CADR0” in **Figure 14.4**, to “CADR”; Changed “(1)”, “(2)”, and “(3)”, to “(a)”, “(b)”, and “(c)”, respectively<br>• Modified description “IADR1 and IADR0” in line 2 of **14.6** (previous 14.4.5), to “IADR”<br>• Moved a sentence from previous 14.5, to lines 5 and 6 of **14.6** |
| | | 205 | • Modified formulas in **Figure 14.5** |
| | | | **Chapter 16. Timers** |
| | | 209 | • Corrected the following typos: “TTA0TGL” and “TAiGH and TAiGL” in **Figure 16.1**, to “TA0TGL” and “TAiTGH and TAiTGL”, respectively |
| | | 211 | • Corrected a typo “TBiS bit” in **Figure 16.3**, to “TAiS” |
| | | 226 | • Corrected a typo “FEh” as value of m for “8-bit PWM” in **Table 16.5**, to “FFh” |
| | | 227 | • Modified reset value for TAiMR register in **Figure 16.16** from “0000 000b” to “0000 0000b” |
| | | 229 | • Changed expression “TBiS bit” in **Figure 16.19**, to “TBiS” |
| | | 230 | • Changed description for Note 1 of TBiMR in **Figure 16.21** descriptively |
| | | 241 | • Deleted “(j = 0 to 5)” from the eighth bullet point of **16.3.3.2** |
| | | | **Chapter 17. Three-phase Motor Control Timers** |
| | | — | • Made minor text modifications to the this chapter |
| | | 243 | • Added description “P3_2 to P3_7” to paragraph below “Inverse control” unit in **Figure 17.1** |
| | | 244 | • Modified the expression of Note 8 for INVC0 in **Figure 17.2** descriptively |
| | | 250 | • Modified reset value for TB2MR in **Figure 17.8** |
| | | 257 | • Corrected a typo “TA4-1 register” in **Figure 17.17**, to “TA41 register” |
| | | | **Chapter 18. Serial Interface** |
| | | — | • Made minor text modifications to the this chapter |
| | | 264 | • Modified expression “7 (, 8, and 9)-bit transfer data” for “Function” of UiMR register in **Figure 18.4**, to “7(, 8, and 9)-bit character length” |
| | | 270 | • Modified “SCL pin” for SWC bit of UiSMR2 in **Figure 18.11**, to “SCLi pin” |
| | | 271 | • Modified description “To set the SS” in Note 2 for UiSMR3 register in **Figure 18.12**, to “To use the SS function”; Corrected a typo “UiCO register” in Note 2, to “UiC0 register” |
| | | 272 | • Modified description for the SWC9 bit of UiSMR4 in **Figure 18.13** |
| | | 273 | • Deleted description “and read as undefined value” from “Function” of b15-b9 for UiTB register in **Figure 18.15** |
| | | 277 | • Modified description “(i = 0 to 6)” for “Transmit/receive clock” in **Table 18.2**, to “(i = 0 to 8)” |

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| | | 283 | • Deleted a ")" from description for Note 1 in **Figure 18.23** |
| | | 285 | • Modified expressions "1 stop bit" and "2 stop bits" in the first bullet point of "Error detection" in **Table 18.5**, to "1 stop bit length" and "2 stop bit length", descriptively |
| | | 291 | • Deleted "(i = 0 to 8)" from "B" of **18.2.2** |
| | | 297 | • Modified the following descriptions in **Table 18.11**: "UART transmit/ UART receive interrupt" in "IICM2 = 1", to "Transmit/receive interrupt"; "the Pi_jS register (i, j = 0 to 7) if the I/O port is selected)" in "Default output value at the SDAi pin", to "the Port Pi register (i = 0 to 7) if the I/O port is selected by output function select registers)" |
| | | 299 | • Modified description "UART transmit/UART receive interrupt" in (3) of **Figure 18.32**, to "transmit/receive interrupt" |
| | | 307 | • Corrected a typo "$\overline{SS}$ pin" in title of **Figure 18.37**, to "$\overline{SSi}$ pin" |
| | | 310 | • Deleted whole description from the third bullet point of **18.5.2.2** |
| | | | **Chapter 19. A/D Converter** |
| | | — | • Made minor text modifications to the this chapter |
| | | 316, 318 | • Modified Notes 2 to 4 for AD0CON2 in **Figure 19.4** and AD0CON4 in **Figure 19.6** descriptively |
| | | 326 | • Modified description "AD0j register" in line 3 of **19.2.1**, to "AD0i register" |
| | | 331 | • Deleted "(j = 0 to 7)" from the eighth bullet point of **19.3.2** |
| | | | **Chapter 20. X-Y Conversion** |
| | | — | • Made minor text modifications to the this chapter |
| | | | **Chapter 23. Intelligent I/O** |
| | | — | • Made minor text modifications to the this chapter |
| | | 342-344 | • Moved "(j = 0 to 7)" in **Figures 23.1 to 23.3** to respective figure titles |
| | | 342 | • Added description for BT0R to **Figure 23.1** |
| | | 343 | • Added description for BT1R to **Figure 23.2** |
| | | 344 | • Added description for bits BT2R, PO2jR, IE0R to IE2R, SIO2TR, and SIO2RR to **Figure 23.3**; Deleted note symbol "(3)" |
| | | 362 | • Changed "IIOi_j pin function" in **Table 23.4**, to "IIOi_j input pin function"; Moved "(j = 0 to 7)" for "Trigger input polarity", to the table title |
| | | 363 | • Moved "(j = 0 to 7; k = 6, 7)" below **Table 23.5** to the title |
| | | 363, 364 | • Moved "(j = 0 to 7)" in **Figures 23.22 and 23.23** to the titles |
| | | 365 | • Moved "(j = 6, 7)" in **Figure 23.24** to the title |
| | | 366 | • Moved "(j = 0 to 7)" below **Table 23.6** to the title |
| | | 367, 369, 371, 372 | • Added "(or OUTC2_j pin for Group 2)" after "IIOi_j pin", to respective line 1 of **23.3.1**, **23.3.2**, and **23.3.3** and description for "Specification" in **Tables 23.7 to 23.9** |
| | | 367, 369, 372 | • Modified "IIOi_j pin function" in "Item" in **Table 23.7**, and "IIOi_j pin function (output)" in "Item" in **Tables 23.8 and 23.9**, to "IIOi_j output pin (or OUTC2_j pin for Group 2) function" |
| | | 374 | • Modified description "G2PO0 register" for "Output waveform" in **Table 23.10**, to "G2POj register (j = 0 to 7)" |

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| | | 375 | • Corrected following typos: “fBTi” in **Figure 23.28**, to “fBT2”; “G2POCR register”, to “G2POCRj register” |
| | | 377 | • Added description “in the G2RTP register” to “RTPj bit” in **Figure 23.30** |
| | | | **Chapter 24. Multi-master $I^2C$-bus Interface** |
| | | — | • Made minor text modifications to the this chapter |
| | | 396 | • Modified description “SCL/SDA Interrupt”s for bits SIP and SIS in **Figure 24.8**, to “$I^2C$bus-line Interrupt” |
| | | 397 | • Modified description “(b2-b3)” for I2CCR1 register in **Figure 24.9**, to “(b3-b2)” |
| | | 400 | • Modified description “(b5)” for I2CCR2 register in **Figure 24.11**, to “(b6)” |
| | | 406 | • Modified description “b2 b1 b0” for bits CLK2 to CLK0 in **Figure 24.14**, to “b3 b2 b1” |
| | | | • Modified description below **Figure 24.14 and 24.1.9.1** |
| | | | **Chapter 25. CAN Module** |
| | | — | • Made minor text modifications to the this chapter |
| | | 421 | • Modified description “XIN” in **Figure 25.1**, to “Main clock” |
| | | 427 | • Modified read value of b4 for CiCLKR in **Figure 25.3**, to be as undefined |
| | | 428 | • Added description “from CAN reset mode” to Note 1 for CiBCR in **Figure 25.4** |
| | | | • Corrected a typo “(b23-22)” for CiBCR register in **Figure 25.4**, to “(b23-b22)” |
| | | 430, 431 | • Modified function of b31 to b29 for CiMKRk in **Figure 25.5** and b29 for CiFIDCRn in **Figure 25.6**, to “Reserved” |
| | | 430 | • Corrected a typo “47E10h” in **Figure 25.5**, to “47A10h” |
| | | 431 | • Deleted description “and read as 0” from Note 2 for CiFIDCRn in **Figure 25.6** |
| | | 435 | • Modified function of b29, b39 to b32, and b47 to b44 for CiMBj in **Figure 25.8**, to “Reserved; Changed description for Note 2; Deleted description “and read as 0” from Note 4 |
| | | 437 | • Changed expression “-”s in **Table 25.6**, to “X”s |
| | | 439 | • Modified Note 4 for CiMCTLj register in **Figure 25.10** descriptively |
| | | 442 | • Deleted description of a maximum delay from lines 7 to 8 of **25.1.9.9** |
| | | 447 | • Modified description for **25.1.12.2** |
| | | 456 | • Modified description for Note 2 of CiCSSR in **Figure 25.20** |
| | | | • Modified “0”s for b3 and b4 of 4th read in **Figure 25.21**, to “X”s |
| | | 457 | • Modified description for Note 1 of CiAFSR in **Figure 25.22** |
| | | | • Modified description “CAFSR” in line 2 of **25.1.18**, to “CiAFSR” |
| | | 463, 464 | • Deleted “(8 bits)” from “Function” for CiRECR in **Figure 25.26** and CiTECR in **Figure 25.27** |
| | | 471 | • Modified description of Note 2 for **Figure 25.34** |
| | | 477 | • Deleted description of BRP from **Figure 25.36** |
| | | 478 | • Deleted description “division value of the” for fCAN from line 1 of **25.3.3** |

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| | | 488 | • Moved description for “CANi wake-up interrupt” in **25.7**, to an upper line |
| | | | **Chapter 26. I/O Pins** |
| | | — | • Made minor text modifications to the this chapter |
| | | 491 | • Added “b” to binary form in **Table 26.1** |
| | | 493 | • Changed “IIO0_i output” and “IIO1_i output” in “Function” of P1_iS register in **Figure 26.4**, to “IIO0 output” and “IIO1 output”, respectively |
| | | 498, 504 | • Modified description “b7-b3” for P6_iS in **Figure 26.9** and P11_iS in **Figure 26.15**, to “b7” |
| | | 501 | • Modified bit symbol for b6 of registers P9_3S to P9_0S in **Figure 26.12**, to be exclusively as NOD |
| | | 504 | • Modified the explanation about the usage of an N-channel open drain output in the paragraphs below **Figure 26.15** |
| | | 510 | • Modified expression “TAiIN input” in Note 1 for IFS0 in **Figure 26.20**, to “TAiIN” |
| | | 513 | • Modified description for IFS30 and IFS31 in **Figure 26.23** from “port P9”, to “port P6/port P9” |
| | | | **Chapter 27. Flash Memory** |
| | | — | • Revised this chapter entirely<br>• Changed expressions “write” and “rewrite”, to “program” when this word is used in combination with “erase” |
| | | 521 | • Revised **Table 27.1** |
| | | 531 | • Corrected a typo “(b7-4)” for FMR1 register in **Figure 27.8**, to “(b7-b4)” |
| | | 533 | • Corrected address and “Function” of BP15 bit in **Figure 27.12** |
| | | 540 | • Corrected a typo “b5-0” in **Tables 27.15 and 27.16**, to “b5-b0” |
| | | 547 | • Modified expression “Status/Error” in **Table 27.17**, to “Error” |
| | | 552 | • Modified description for the third bullet point of EW1 mode in **27.6.5** |
| | | | **Chapter 28. Electrical Characteristics** |
| | | — | • Made minor text modifications to the this chapter |
| | | 557 | • Corrected a typo “pots” in line 2 of Note 2 for **Table 28.4**, to “ports” |
| | | 562 | • Changed the order of description of trec(STOP) and trec(WAIT) in **Table 28.13 and Figure 28.4** |
| | | 572, 585 | • Changed the minimum value for “$t_{w(ADH)}$” in **Tables 28.31 and 28.57** from “$2/\phi_{AD}$”, to “$3/\phi_{AD}$” |
| | | 573, 576, 586, 589 | • Newly Added characteristics for multi-master $I^2C$-bus to **Tables 28.34, 28.39, 28.40, 28.60, 28.65, and 28.66** |
| | | 575, 588 | • Modified “Characteristics” for $t_{su(S-ALE)}$ in **Tables 28.36 and 28.62**, from “Chip-select hold time for ALE” to “Chip-select setup time for ALE” |
| | | 576, 589 | • Modified “Characteristics” for $t_{h(C-Q)}$ in **Tables 28.37 and 28.63**, from “TXDi hold time” to “TXDi output hold time”<br>• Added “Measurement condition” to **Tables 28.38 and 28.64** |

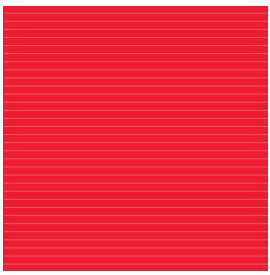
| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| | | 582 | • Corrected typos "$t_{w(H)}$," "$t_{w(L)}$", "$t_r$", and "$t_f$" in **Table 28.46**, to "$t_{w(XH)}$", "$t_{w(XL)}$", "$t_{r(X)}$", and "$t_{f(X)}$", respectively |
| | | 594 | • Newly Added timing diagram for multi-master $I^2C$-bus to **Figure 28.11** |
| | | | **Chapter 29. Usage Notes** |
| | | — | • Made minor text modifications to the this chapter |
| | | 601 | • Modified description "channel i" in line 1 of the first bullet point of **29.6.1**, to "the DMDi register"; Added one sentence to the same bullet point; Deleted whole description of the second and third bullet points; Added two new paragraphs |
| | | 604 | • Deleted "(j = 0 to 5)" from the eighth bullet point of **29.7.3.2** |
| | | 606 | • Deleted whole description from the third bullet point of **29.9.2.2** |
| | | 608 | • Deleted "(i = 0 to 7)" from the eighth bullet point of **29.10.2** |
| | | 609 | • Modified description for the third bullet point of EW1 mode in **29.11.5** |

R32C/118 Group Hardware Manual

Publication Date: Apr 08, 2009 Rev.0.62
Nov 24, 2009 Rev.1.00

Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
2-6-2, Otemachi, Chiyoda-ku, Tokyo 100-0004

# R32C/118 Group
# Hardware Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan

# X-ON Electronics

Largest Supplier of Electrical and Electronic Components

***Click to view similar products for*** Microprocessors - MPU ***category:***

***Click to view products by*** Renesas ***manufacturer:***

Other Similar products are found below :

MC68302EH20C MC7457RX1000LC MC7457RX1267LC A2C00010998 A A2C52004004 R5F117BCGNA#20 R5F52106BDLA#U0 R5S72690W266BG#U0 ADJ3400IAA5DOE MPC8245TVV266D MPC8245TZU300D MPC8260ACVVMHBB MPC8323ECVRAFDCA MPC8536ECVJAVLA BOXNUC5PGYH0AJ 20-668-0024 P1010NSN5DFB P2020NXE2HHC P5020NSE7TNB P5020NSE7VNB LS1020ASN7KQB LS1020AXN7HNB LS1020AXN7KQB A2C00010729 A A2C00039344 T1022NSE7MQB T1022NXN7PQB T1023NSE7MQA T1024NXE7PQA T1042NSN7MQB T1042NXN7WQB T2080NSN8PTB T2080NXE8TTB T2081NXN8TTB R5F101AFASP#V0 MC68302CEH20C TS68040MF33A MPC8260ACVVMIBB MPC8280CZUUPEA MPC8313ECVRAFFC MPC8313ECVRAGDC MPC8313EVRADDC MPC8313EVRAFFC MPC8313VRADDC MPC8323EVRAFDCA BOXSTCK1A8LFCL UPD78F0503AMCA-CAB-G UPD78F0513AGA-8EU-AT UPD78F0730MC-CAB-AX DF2134BFA20V