REJ09B0047-0060Z

16

# M16C/28 Group

## Hardware Manual

RENESAS 16-BIT CMOS SINGLE-CHIP MICROCOMPUTER

M16C FAMILY / M16C/Tiny SERIES

Hardware Manual

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

Rev. 0.60
Revision date: February. 01. 2004

Renesas Technology
www.renesas.com

## Keep safety first in your circuit designs!

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

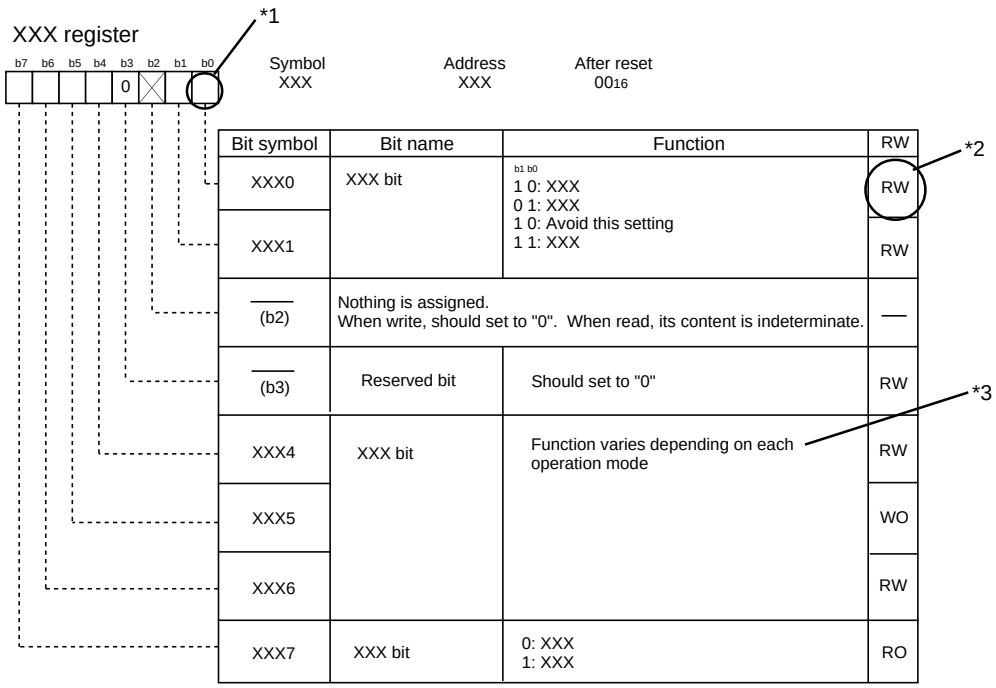
## Notes regarding these materials

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

# How to Use This Manual

This hardware manual provides detailed information on features in the M16C/28 Group microcomputer.
Users are expected to have basic knowledge of electric circuits, logical circuits and micro-computer.

Each register diagram contains bit functions with the following symbols and descriptions.




*1

Blank: Set to "0" or "1" according to your intended use
0: Set to "0"
1: Set to "1"
X: Nothing is assigned

*2

RW: Read and write
RO: Read only
WO: Write only
–: Nothing is assigned

*3

Terms to use here are explained as follows.

• Nothing is assigned
Nothing is assigned to the bit concerned. When write, set to "0" for new function in future plan.

• Reserved bit
Reserved bit. Set the specified value.

• Avoid this setting
The operation at having selected is not guaranteed.

• Function varies depending on each operation mode
Bit function varies depending on peripheral function mode.
Refer to register diagrams in each mode.

# M16C Family Documents

| Document | Contents |
|---|---|
| Short Sheet | Hardware overview |
| Data Sheet | Hardware overview and electrical characteristics |
| Hardware Manual | Hardware specifications (pin assignments, memory maps, specifications of peripheral functions, electrical characteristics, timing charts) |
| Software Manual | Detailed description about instructions and microcomputer performance by each instruction |
| Application Note | • Application examples of peripheral functions<br>• Sample programs<br>• Introductory description about basic functions in M16C family<br>• Programming method with the assembly and C languages |

# Table of Contents


**Quick Reference to Pages Classified by Address ..... B-1**

**1. Overview ..... 1**

**1.1 Applications ..... 1**

**1.2 Performance Outline ..... 2**

**1.3 Block Diagram ..... 4**

**1.4 Product List ..... 6**

**1.5 Pin Configuration ..... 8**

**1.6 Pin Description ..... 10**

**2. Central Processing Unit (CPU) ..... 12**

**2.1 Data Registers (R0, R1, R2 and R3) ..... 12**

**2.2 Address Registers (A0 and A1) ..... 12**

**2.3 Frame Base Register (FB) ..... 13**

**2.4 Interrupt Table Register (INTB) ..... 13**

**2.5 Program Counter (PC) ..... 13**

**2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP) ..... 13**

**2.7 Static Base Register (SB) ..... 13**

**2.8 Flag Register (FLG) ..... 13**

**2.8.1 Carry Flag (C Flag) ..... 13**

**2.8.2 Debug Flag (D Flag) ..... 13**

**2.8.3 Zero Flag (Z Flag) ..... 13**

**2.8.4 Sign Flag (S Flag) ..... 13**

**2.8.5 Register Bank Select Flag (B Flag) ..... 13**

**2.8.6 Overflow Flag (O Flag) ..... 13**

**2.8.7 Interrupt Enable Flag (I Flag) ..... 13**

**2.8.8 Stack Pointer Select Flag (U Flag) ..... 13**

**2.8.9 Processor Interrupt Priority Level (IPL) ..... 13**

**2.8.10 Reserved Area ..... 13**

3. Memory ..... 14

4. Special Function Register (SFR) Map ..... 15

5. Reset ..... 22

5.1 Hardware Reset ..... 22

5.1.1 Hardware Reset 1 ..... 22

5.1.2 Hardware Reset 2 ..... 22

5.2 Software Reset ..... 23

5.3 Watchdog Timer Reset ..... 23

5.4 Oscillation Stop Detection Reset ..... 23

5.5 Voltage Detection Circuit ..... 25

5.5.1 Voltage Detection Interrupt ..... 28

5.5.1.1 Precautions ..... 28

5.5.1.1.1. Limitations on Stop Mode ..... 28

5.5.1.1.2. Limitations on WAIT Instruction ..... 29

6. Processor Mode ..... 30

7. Clock Generation Circuit ..... 31

7.1 Main Clock ..... 38

7.2 Sub Clock ..... 39

7.3 Ring Oscillator Clock ..... 40

7.4 PLL Clock ..... 40

7.5 CPU Clock and Peripheral Function Clock ..... 42

7.5.1 CPU Clock ..... 42

7.5.2 Peripheral Function Clock
(f1, f2, f8, f32, f1SIO, f2SIO, f8SIO, f32SIO, fAD, fC32) ..... 42

7.6 Power Control ..... 43

7.6.1 Normal Operation Mode ..... 43

7.6.1.1 High-speed Mode ..... 43

7.6.1.2 PLL Operation Mode ..... 43

7.6.1.3 Medium-speed Mode ..... 43

7.6.1.4 Low-speed Mode ..... 43

7.6.1.5 Low Power Dissipation Mode ..... 43

7.6.1.6 Ring Oscillator Mode ..... 44
7.6.1.7 Ring Oscillator Low Power Dissipation Mode ..... 44
7.6.2 Wait Mode ..... 44
7.6.2.1 Peripheral Function Clock Stop Function ..... 44
7.6.2.2 Entering Wait Mode ..... 44
7.6.2.3 Pin Status During Wait Mode ..... 44
7.6.2.4 Exiting Wait Mode ..... 45
7.6.3 Stop Mode ..... 46
7.6.3.1 Entering Stop Mode ..... 46
7.6.3.2 Pin Status during Stop Mode ..... 46
7.6.3.3 Exiting Stop Mode ..... 46
7.7 System Clock Protective Function ..... 50
7.8 Oscillation Stop and Re-oscillation Detect Function ..... 50
7.8.1 Operation When CM27 bit = 0 (Oscillation Stop Detection Reset) ..... 51
7.8.2 Operation When CM27 bit = 1 (Oscillation Stop and Re-oscillation Detect Interrupt) ..... 51
7.8.3 How to Use Oscillation Stop and Re-oscillation Detect Function ..... 52
8. Protection ..... 53
9. Interrupts ..... 54
9.1 Type of Interrupts ..... 54
9.1.1 Software Interrupts ..... 55
9.1.1.1 Undefined Instruction Interrupt ..... 55
9.1.1.2 Overflow Interrupt ..... 55
9.1.1.3 BRK Interrupt ..... 55
9.1.1.4 INT Instruction Interrupt ..... 55
9.1.2 Hardware Interrupts ..... 56
9.1.2.1 Special Interrupts ..... 56
9.1.2.1.1 $\overline{\text{NMI}}$ Interrupt ..... 56
9.1.2.1.2 $\overline{\text{DBC}}$ Interrupt ..... 56
9.1.2.1.3 Watchdog Timer Interrupt ..... 56
9.1.2.1.4 Oscillation Stop and Re-oscillation Detection Interrupt ..... 56
9.1.2.1.5 Voltage Down Detection Interrupt ..... 56
9.1.2.1.6 Single-step Interrupt ..... 56
9.1.2.1.7 Address Match Interrupt ..... 56
9.1.2.2 Peripheral Function Interrupts ..... 56
9.2 Interrupts and Interrupt Vector ..... 57

9.2.1 Fixed Vector Tables ..... 57
9.2.2 Relocatable Vector Tables ..... 58
9.3 Interrupt Control ..... 59
9.3.1 I Flag ..... 62
9.3.2 IR Bit ..... 62
9.3.3 ILVL2 to ILVL0 Bits and IPL ..... 62
9.4 Interrupt Sequence ..... 63
9.4.1 Interrupt Response Time ..... 64
9.4.2 Variation of IPL when Interrupt Request is Accepted ..... 64
9.4.3 Saving Registers ..... 65
9.4.4 Returning from an Interrupt Routine ..... 67
9.5 Interrupt Priority ..... 67
9.5.1 Interrupt Priority Resolution Circuit ..... 67
9.6 $\overline{\text{INT}}$ Interrupt ..... 69
9.7 $\overline{\text{NMI}}$ Interrupt ..... 70
9.8 Key Input Interrupt ..... 70
9.9 Address Match Interrupt ..... 71
10. Watchdog Timer ..... 73
11. DMAC ..... 75
11.1 Transfer Cycles ..... 80
11.1.1 Effect of Source and Destination Addresses ..... 80
11.1.2 Effect of Software Wait ..... 80
11.2. DMA Transfer Cycles ..... 82
11.3 DMA Enable ..... 83
11.4 DMA Request ..... 83
11.5 Channel Priority and DMA Transfer Timing ..... 84
12. Timers ..... 85
12.1 Timer A ..... 87
12.1.1. Timer Mode ..... 90
12.1.2. Event Counter Mode ..... 91
12.1.2.1 Counter Initialization by Two-Phase Pulse Signal Processing ..... 95

12.1.3. One-shot Timer Mode ..... 96
12.1.4. Pulse Width Modulation (PWM) Mode ..... 98
12.2 Timer B ..... 101
12.2.1 Timer Mode ..... 103
12.2.2 Event Counter Mode ..... 104
12.2.3 Pulse Period and Pulse Width Measurement Mode ..... 105
12.2.4 A-D Trigger Mode ..... 107
12.3 Three-phase Motor Control Timer Function ..... 109
12.3.1 Position-data-retain Function ..... 120
12.3.1.1 Operation of the Position-data-retain Function ..... 120
12.3.1.2 Position-data-retain Function Control Register ..... 121
12.3.1.2.1 W-phase Position Data Retain Bit (PDRW) ..... 121
12.3.1.2.2 V-phase Position Data Retain Bit (PDRV) ..... 121
12.3.1.2.3 U-phase Position Data Retain Bit (PDRU) ..... 121
12.3.1.2.4 Retain-trigger Polarity Select Bit (PDRT) ..... 121
13. Timer S (Input Capture/Output Compare) ..... 122
13.1 Base Timer ..... 134
13.1.1 Base Timer Reset Register ..... 138
13.2 Interrupt Operation ..... 139
13.3 DMA Support ..... 139
13.4 Time Measurement Function ..... 140
13.5 Waveform Generation Function ..... 144
13.5.1 Single-Phase Waveform Output Mode ..... 145
13.5.2 Phase-Delayed Waveform Output Mode ..... 147
13.5.3 Set/Reset Waveform Output (SR Waveform Output) Mode ..... 149
13.6 I/O Port Function Select ..... 151
13.6.1 INPC17 Alternate Input Pin Selection ..... 152
13.6.2 Digital Debounce Function for Pin P17/$\overline{\text{INT5}}$/INPC17 ..... 152
14. Serial I/O ..... 153
14.1. UARTi (i=0 to 2) ..... 153
14.1.1. Clock Synchronous serial I/O Mode ..... 163
14.1.1.1 CLK Polarity Select Function ..... 167
14.1.1.2 LSB First/MSB First Select Function ..... 167
14.1.1.3 Continuous receive mode ..... 168

14.1.1.4 Serial data logic switch function (UART2) ..... 168
14.1.1.5 Transfer clock output from multiple pins function (UART1) ..... 168
14.1.1.6 $\overline{CTS}$/$\overline{RTS}$ separate function (UART0) ..... 169
14.1.2. Clock Asynchronous Serial I/O (UART) Mode ..... 170
14.1.2.1. LSB First/MSB First Select Function ..... 174
14.1.2.2. Serial Data Logic Switching Function (UART2) ..... 175
14.1.2.3. TxD and RxD I/O Polarity Inverse Function (UART2) ..... 175
14.1.2.4. $\overline{CTS}$/$\overline{RTS}$ Separate Function (UART0) ..... 176
14.1.3 Special Mode 1 (I2C Bus mode)(UART2) ..... 177
14.1.3.1 Detection of Start and Stop Condition ..... 183
14.1.3.2 Output of Start and Stop Condition ..... 183
14.1.3.3 Arbitration ..... 184
14.1.3.4 Transfer Clock ..... 185
14.1.3.5 SDA Output ..... 185
14.1.3.6 SDA Input ..... 185
14.1.3.7 ACK and NACK ..... 186
14.1.3.8 Initialization of Transmission/Reception ..... 186
14.1.4 Special Mode 2 (UART2) ..... 187
14.1.4.1 Clock Phase Setting Function ..... 190
14.1.4.1.1 Master (Internal Clock) ..... 190
14.1.4.1.2 Slave (External Clock) ..... 190
14.1.5 Special Mode 3 (IE Bus mode)(UART2) ..... 192
14.1.6 Special Mode 4 (SIM Mode) (UART2) ..... 194
14.1.6.1 Parity Error Signal Output ..... 197
14.1.6.2 Format ..... 198
14.2 SI/O3 and SI/O4 ..... 199
14.2.1 SI/Oi Operation Timing ..... 202
14.2.2 CLK Polarity Selection ..... 202
14.2.3 Functions for Setting an SOUTi Initial Value ..... 203
15. A-D Converter ..... 204
15.1 Operation Modes ..... 210
15.1.1 One-Shot Mode ..... 210
15.1.2 Repeat mode ..... 212
15.1.3 Single Sweep Mode ..... 214
15.1.4 Repeat Sweep Mode 0 ..... 216
15.1.5 Repeat Sweep Mode 1 ..... 218
15.1.6 Simultaneous Sample Sweep Mode ..... 220
15.1.7 Delayed Trigger Mode 0 ..... 223
15.1.8 Delayed Trigger Mode 1 ..... 229

15.2 Resolution Select Function ..... 235
15.3 Sample and Hold ..... 235
15.4 Current Consumption Reducing Function ..... 235
15.5 Analog Input Pin and External Sensor Equivalent Circuit Example 235
15.6 Precautions of Using A-D Converter ..... 236
16. Multi-master I²C bus Interface ..... 237
16.1 I²C0 Data Shift Register (S00 register) ..... 246
16.2 I²C0 Address Register (S0D0 register) ..... 246
16.3 I²C0 Clock Control Register (S20 register) ..... 247
16.3.1 Bits 0 to 4: SCL frequency control bits (CCR0–CCR4) ..... 247
16.3.2 Bit 5: SCL mode specification bit (FAST MODE) ..... 247
16.3.3 Bit 6: ACK bit (ACK BIT) ..... 247
16.3.4 Bit 7: ACK clock bit (ACK) ..... 247
16.4 I²C0 Control Register 0 (S1D0 register) ..... 249
16.4.1 Bits 0 to 2: Bit counter (BC0–BC2) ..... 249
16.4.2 Bit 3: I2C interface enable bit (ES0) ..... 249
16.4.3 Bit 4: Data format select bit (ALS) ..... 249
16.4.4 Bit 6: I²C bus interface reset bit (IHR) ..... 249
16.4.5 Bit 7: I²C bus interface pin input level select bit (TISS) ..... 250
16.5 I²C0 Status Register (S10 register) ..... 251
16.5.1 Bit 0: Last receive bit (LRB) ..... 251
16.5.2 Bit 1: General call detection flag (ADR0) ..... 251
16.5.3 Bit 2: Slave address comparison flag (AAS) ..... 251
16.5.4 Bit 3: Arbitration lost detection flag (AL)(Note 1) ..... 251
16.5.5 Bit 4: I²C bus interface interrupt request bit (PIN) ..... 252
16.5.6 Bit 5: Bus busy flag (BB) ..... 252
16.5.7 Bit 6: Communication mode select bit (transfer direction select bit: TRX) . 253
16.5.8 Bit 7: Communication mode select bit (master/slave select bit: MST) ..... 253
16.6 I²C0 control register 1 (S3D0 register) ..... 254
16.6.1 Bit 0 : Interrupt enable bit by STOP condition (SIM ) ..... 254
16.6.2 Bit 1: Interrupt enable bit at the completion of data receive (WIT) ..... 254
16.6.3 Bits 2,3 : Port function select bits PED, PEC ..... 255
16.6.4 Bits 4,5 : SDA/SCL logic output value monitor bits SDAM/SCLM ..... 256

16.6.5 Bits 6,7 : I²C system clock select bits ICK0, ICK1 ..... 256
16.6.6 The address receive in STOP mode/WAIT mode ..... 256
16.7 I²C0 control register 2 (S3D0 register) ..... 257
16.7.1 Bit0: Time out detection function enable bit (TOE) ..... 258
16.7.2 Bit1: Time out detection flag (TOF ) ..... 258
16.7.3 Bit2: time out detection period select bit (TOSEL) ..... 258
16.7.4 Bits 3,4,5: I²C system clock select bits (ICK2-4) ..... 258
16.7.5 Bit7: STOP condition detection interrupt request bit (SCPIN) ..... 258
16.8 I²C0 START/STOP condition control registers (S2D0 register) ..... 259
16.8.1 Bit0-Bit4: START/STOP condition setting bits (SSC0-SSC4) ..... 259
16.8.2 Bit5: SCL/SDA interrupt pin polarity select bit (SIP) ..... 259
16.8.3 Bit6 : SCL/SDA interrupt pin select bit (SIS) ..... 259
16.8.4 Bit7: START/STOP condition generation select bit (STSPSEL) ..... 259
16.9 START Condition Generation Method ..... 260
16.10 START condition duplicate protect function ..... 261
16.11 STOP Condition Generation Method ..... 261
16.12 START/STOP Condition Detect Operation ..... 263
16.13 Address Data Communication ..... 264
16.13.1 Example of Master Transmit ..... 264
16.13.2 Example of Slave Receive ..... 265
16.14 Usage precautions ..... 267
17. Programmable I/O Ports ..... 270
17.1 Port Pi Direction Register (PDi Register, i = 0 to 3, 6 to 10) ..... 270
17.2 Port Pi Register (Pi Register, i = 0 to 3, 6 to 10) ..... 270
17.3 Pull-up Control Register 0 to Pull-up Control Register 2 (PUR0 to PUR2 Registers) ..... 270
17.4 Port Control Register ..... 270
17.5 Pin Assignment Control register (PACR) ..... 271
17.6 Digital Debounce function ..... 271
18. Electrical Characteristics ..... 284
18.1. Normal version ..... 284

18.2. T version .....305
19. Flash Memory Version ..... 326
19.1 Flash Memory Performance .....326
19.2 Memory Map .....328
19.3 Functions To Prevent Flash Memory from Rewriting ..... 331
19.3.1 ROM Code Protect Function ..... 331
19.3.2 ID Code Check Function ..... 331
19.4 CPU Rewrite Mode .....333
19.4.1 EW0 Mode ..... 334
19.4.2 EW1 Mode ..... 334
19.5 Register Description .....335
19.5.1 Flash memory control register 0 (FMR0): ..... 335
•FMR 00 Bit ..... 335
•FMR01 Bit ..... 335
•FMR02 Bit ..... 335
•FMSTP Bit ..... 335
•FMR06 Bit ..... 335
•FMR07 Bit ..... 335
19.5.2 Flash memory control register 1 (FMR1): ..... 336
•FMR11 Bit ..... 336
•FMR16 Bit ..... 336
•FMR17 Bit ..... 336
19.5.3 Flash memory control register 4 (FMR4): ..... 336
•FMR40 Bit ..... 336
•FMR41 Bit ..... 336
•FMR46 Bit ..... 336
19.6 Precautions in CPU Rewrite Mode ..... 341
19.6.1 Operation Speed ..... 341
19.6.2 Prohibited Instructions ..... 341
19.6.3 Interrupts ..... 341
19.6.4 How to Access ..... 341
19.6.5 Writing in the User ROM Space ..... 341
19.6.5.1 EW0 Mode ..... 341
19.6.5.2 EW1 Mode ..... 341
19.6.6 DMA Transfer ..... 342
19.6.7 Writing Command and Data ..... 342

19.6.8 Wait Mode ..... 342
19.6.9 Stop Mode ..... 342
19.6.10 Low Power Consumption Mode and Ring Oscillator-Low Power Consumption Mode ..... 342
19.7 Software Commands ..... 343
19.7.1 Read Array Command ($FF_{16}$) ..... 343
19.7.2 Read Status Register Command ($70_{16}$) ..... 343
19.7.3 Clear Status Register Command ($50_{16}$) ..... 344
19.7.4 Program Command ($40_{16}$) ..... 344
19.7.5 Block Erase ..... 345
19.8 Status Register ..... 347
19.8.1 Sequence Status (SR7 and FMR00 Bits ) ..... 347
19.8.2 Erase Status (SR5 and FMR07 Bits) ..... 347
19.8.3 Program Status (SR4 and FMR06 Bits) ..... 347
19.8.4 Full Status Check ..... 348
19.9 Standard Serial I/O Mode ..... 350
19.9.1 ID Code Check Function ..... 350
19.9.2 Example of Circuit Application in Standard Serial I/O Mode ..... 354
19.10 Parallel I/O Mode ..... 356
19.10.1 ROM Code Protect Function ..... 356
20. Package ..... 357
Register Index ..... 358

# Quick Reference to Pages Classified by Address

| Address | Register | Symbol | Page |
|---|---|---|---|
| 0000₁₆ | | | |
| 0001₁₆ | | | |
| 0002₁₆ | | | |
| 0003₁₆ | | | |
| 0004₁₆ | Processor mode register 0 | PM0 | 30 |
| 0005₁₆ | Processor mode register 1 | PM1 | 30 |
| 0006₁₆ | System clock control register 0 | CM0 | 33 |
| 0007₁₆ | System clock control register 1 | CM1 | 34 |
| 0008₁₆ | | | |
| 0009₁₆ | Address match interrupt enable register | AIER | 72 |
| 000A₁₆ | Protect register | PRCR | 53 |
| 000B₁₆ | | | |
| 000C₁₆ | Oscillation stop detection register | CM2 | 35 |
| 000D₁₆ | | | |
| 000E₁₆ | Watchdog timer start register | WDTS | 74 |
| 000F₁₆ | Watchdog timer control register | WDC | 25, 74 |
| 0010₁₆<br>0011₁₆<br>0012₁₆ | Address match interrupt register 0 | RMAD0 | 72 |
| 0013₁₆ | | | |
| 0014₁₆<br>0015₁₆<br>0016₁₆ | Address match interrupt register 1 | RMAD1 | 72 |
| 0017₁₆ | | | |
| 0018₁₆ | | | |
| 0019₁₆ | Voltage detection register 1 | VCR1 | 26 |
| 001A₁₆ | Voltage detection register 2 | VCR2 | 26 |
| 001B₁₆ | | | |
| 001C₁₆ | PLL control register 0 | PLC0 | 37 |
| 001D₁₆ | | | |
| 001E₁₆ | Processor mode register 2 | PM2 | 36 |
| 001F₁₆ | Voltage down detection interrupt register | D4INT | 26 |
| 0020₁₆<br>0021₁₆<br>0022₁₆ | DMA0 source pointer | SAR0 | 79 |
| 0023₁₆ | | | |
| 0024₁₆<br>0025₁₆<br>0026₁₆ | DMA0 destination pointer | DAR0 | 79 |
| 0027₁₆ | | | |
| 0028₁₆<br>0029₁₆ | DMA0 transfer counter | TCR0 | 79 |
| 002A₁₆ | | | |
| 002B₁₆ | | | |
| 002C₁₆ | DMA0 control register | DM0CON | 78 |
| 002D₁₆ | | | |
| 002E₁₆ | | | |
| 002F₁₆ | | | |
| 0030₁₆<br>0031₁₆<br>0032₁₆ | DMA1 source pointer | SAR1 | 79 |
| 0033₁₆ | | | |
| 0034₁₆<br>0035₁₆<br>0036₁₆ | DMA1 destination pointer | DAR1 | 79 |
| 0037₁₆ | | | |
| 0038₁₆<br>0039₁₆ | DMA1 transfer counter | TCR1 | 79 |
| 003A₁₆ | | | |
| 003B₁₆ | | | |
| 003C₁₆ | DMA1 control register | DM1CON | 78 |
| 003D₁₆ | | | |
| 003E₁₆ | | | |
| 003F₁₆ | | | |

Note: The blank areas are reserved and cannot be accessed by users.

| Address | Register | Symbol | Page |
|---|---|---|---|
| 0040₁₆ | | | |
| 0041₁₆ | | | |
| 0042₁₆ | | | |
| 0043₁₆ | | | |
| 0044₁₆ | INT3 interrupt control register | INT3IC | 60 |
| 0045₁₆ | IC/OC 0 interrupt control register | ICOC0IC | 60 |
| 0046₁₆ | IC/OC 1 interrupt control register<br>$I^2C$-BUS interface interrupt control register | ICOC1IC<br>IICIC | 60 |
| 0047₁₆ | IC/OC base timer interrupt control register<br>SCLSDA interrupt control register | BTIC<br>SCLDAIC | 60 |
| 0048₁₆ | SI/O4 interrupt control register<br>INT5 interrupt control register | S4IC,<br>INT5IC | 60 |
| 0049₁₆ | SI/O3 interrupt control register,<br>INT4 interrupt control register | S3IC,<br>INT4IC | 60 |
| 004A₁₆ | UART2 Bus collision detection interrupt control register | BCNIC | 60 |
| 004B₁₆ | DMA0 interrupt control register | DM0IC | 60 |
| 004C₁₆ | DMA1 interrupt control register | DM1IC | 60 |
| 004D₁₆ | Key input interrupt control register | KUPIC | 60 |
| 004E₁₆ | A-D conversion interrupt control register | ADIC | 60 |
| 004F₁₆ | UART2 transmit interrupt control register | S2TIC | 60 |
| 0050₁₆ | UART2 receive interrupt control register | S2RIC | 60 |
| 0051₁₆ | UART0 transmit interrupt control register | S0TIC | 60 |
| 0052₁₆ | UART0 receive interrupt control register | S0RIC | 60 |
| 0053₁₆ | UART1 transmit interrupt control register | S1TIC | 60 |
| 0054₁₆ | UART1 receive interrupt control register | S1RIC | 60 |
| 0055₁₆ | Timer A0 interrupt control register | TA0IC | 60 |
| 0056₁₆ | Timer A1 interrupt control register | TA1IC | 60 |
| 0057₁₆ | Timer A2 interrupt control register | TA2IC | 60 |
| 0058₁₆ | Timer A3 interrupt control register | TA3IC | 60 |
| 0059₁₆ | Timer A4 interrupt control register | TA4IC | 60 |
| 005A₁₆ | Timer B0 interrupt control register | TB0IC | 60 |
| 005B₁₆ | Timer B1 interrupt control register | TB1IC | 60 |
| 005C₁₆ | Timer B2 interrupt control register | TB2IC | 60 |
| 005D₁₆ | INT0 interrupt control register | INT0IC | 60 |
| 005E₁₆ | INT1 interrupt control register | INT1IC | 60 |
| 005F₁₆ | INT2 interrupt control register | INT2IC | 60 |
| 0060₁₆ | | | |
| 0061₁₆ | | | |
| 0062₁₆ | | | |
| 0063₁₆ | | | |
| 0064₁₆ | | | |
| 0065₁₆ | | | |
| 0066₁₆ | | | |
| 0067₁₆ | | | |
| 0068₁₆ | | | |
| 0069₁₆ | | | |
| 006A₁₆ | | | |
| 006B₁₆ | | | |
| 006C₁₆ | | | |
| 006D₁₆ | | | |
| 006E₁₆ | | | |
| 006F₁₆ | | | |
| 0070₁₆ | | | |
| 0071₁₆ | | | |
| 0072₁₆ | | | |
| 0073₁₆ | | | |
| 0074₁₆ | | | |
| 0075₁₆ | | | |
| 0076₁₆ | | | |
| 0077₁₆ | | | |
| 0078₁₆ | | | |
| 0079₁₆ | | | |
| 007A₁₆ | | | |
| 007B₁₆ | | | |
| 007C₁₆ | | | |
| 007D₁₆ | | | |
| 007E₁₆ | | | |
| 007F₁₆ | | | |

# Quick Reference to Pages Classified by Address

| Address | Register | Symbol | Page |
|---|---|---|---|
| 0080₁₆ | | | |
| 0081₁₆ | | | |
| 0082₁₆ | | | |
| 0083₁₆ | | | |
| 0084₁₆ | | | |
| 0085₁₆ | | | |
| 0086₁₆ | | | |
| | | | |
| | | | |
| 01B0₁₆ | | | |
| 01B1₁₆ | | | |
| 01B2₁₆ | | | |
| 01B3₁₆ | Flash memory control register 4 (Note 2) | FMR4 | 352 |
| 01B4₁₆ | | | |
| 01B5₁₆ | Flash memory control register 1 (Note 2) | FMR1 | 351 |
| 01B6₁₆ | | | |
| 01B7₁₆ | Flash memory control register 0 (Note 2) | FMR0 | 351 |
| 01B8₁₆ | | | |
| 01B9₁₆ | | | |
| 01BA₁₆ | | | |
| 01BB₁₆ | | | |
| 01BC₁₆<br>01BD₁₆<br>01BE₁₆ | | | |
| 01BF₁₆ | | | |
| | | | |
| | | | |
| 0250₁₆ | | | |
| 0251₁₆ | | | |
| 0252₁₆ | | | |
| 0253₁₆ | | | |
| 0254₁₆ | | | |
| 0255₁₆ | | | |
| 0256₁₆ | | | |
| 0257₁₆ | | | |
| 0258₁₆ | | | |
| 0259₁₆ | | | |
| 025A₁₆ | | | |
| 025B₁₆ | | | |
| 025C₁₆ | Ring oscillator control register | ROCR | 34 |
| 025D₁₆ | Pin assignment control register | PACR | 281 |
| 025E₁₆ | Peripheral clock select register | PCLKR | 36 |
| 025F₁₆ | | | |
| | | | |
| | | | |
| 02E0₁₆ | $I^2C0$ data shift register | S00 | 241 |
| 02E1₁₆ | | | |
| 02E2₁₆ | $I^2C0$ address register | S0D0 | 240 |
| 02E3₁₆ | $I^2C0$ control register 0 | S1D0 | 242 |
| 02E4₁₆ | $I^2C0$ clock control register | S20 | 241 |
| 02E5₁₆ | $I^2C0$ start/stop condition control register | S2D0 | 246 |
| 02E6₁₆ | $I^2C0$ control register 1 | S3D0 | 244 |
| 02E7₁₆ | $I^2C0$ control register 2 | S4D0 | 245 |
| 02E8₁₆ | $I^2C0$ status register | S10 | 243 |
| 02E9₁₆ | | | |
| 02EA₁₆ | | | |
| | | | |
| | | | |
| 02FE₁₆ | | | |
| 02FF₁₆ | | | |

| Address | Register | Symbol | Page |
|---|---|---|---|
| 0300₁₆<br>0301₁₆ | TM/WG register 0 | G1TM0/G1PO0 | 129,130 |
| 0302₁₆<br>0303₁₆ | TM/WG register 1 | G1TM1/G1PO1 | 129,130 |
| 0304₁₆<br>0305₁₆ | TM/WG register 2 | G1TM2/G1PO2 | 129,130 |
| 0306₁₆<br>0307₁₆ | TM/WG register 3 | G1TM3/G1PO3 | 129,130 |
| 0308₁₆<br>0309₁₆ | TM/WG register 4 | G1TM4/G1PO4 | 129,130 |
| 030A₁₆<br>030B₁₆ | TM/WG register 5 | G1TM5/G1PO5 | 129,130 |
| 030C₁₆<br>030D₁₆ | TM/WG register 6 | G1TM6/G1PO6 | 129,130 |
| 030E₁₆<br>030F₁₆ | TM/WG register 7 | G1TM7/G1PO7 | 129,130 |
| 0310₁₆ | WG control register 0 | G1POCR0 | 129 |
| 0311₁₆ | WG control register 1 | G1POCR1 | 129 |
| 0312₁₆ | WG control register 2 | G1POCR2 | 129 |
| 0313₁₆ | WG control register 3 | G1POCR3 | 129 |
| 0314₁₆ | WG control register 4 | G1POCR4 | 129 |
| 0315₁₆ | WG control register 5 | G1POCR5 | 129 |
| 0316₁₆ | WG control register 6 | G1POCR6 | 129 |
| 0317₁₆ | WG control register 7 | G1POCR7 | 129 |
| 0318₁₆ | TM control register 0 | G1TMCR0 | 128 |
| 0319₁₆ | TM control register 1 | G1TMCR1 | 128 |
| 031A₁₆ | TM control register 2 | G1TMCR2 | 128 |
| 031B₁₆ | TM control register 3 | G1TMCR3 | 128 |
| 031C₁₆ | TM control register 4 | G1TMCR4 | 128 |
| 031D₁₆ | TM control register 5 | G1TMCR5 | 128 |
| 031E₁₆ | TM control register 6 | G1TMCR6 | 128 |
| 031F₁₆ | TM control register 7 | G1TMCR7 | 128 |
| 0320₁₆<br>0321₁₆ | Base timer register | G1BT | 124 |
| 0322₁₆ | Base timer control register 0 | G1BCR0 | 124 |
| 0323₁₆ | Base timer control register 1 | G1BCR1 | 126 |
| 0324₁₆ | TM prescale register 6 | G1TPR6 | 128 |
| 0325₁₆ | TM prescale register 7 | G1TPR7 | 128 |
| 0326₁₆ | Function enable register | G1FE | 131 |
| 0327₁₆ | Function select register | G1FS | 131 |
| 0328₁₆<br>0329₁₆ | Base timer reset register | G1BTRR | 127 |
| 032A₁₆ | Divider register | G1DV | 125 |
| 032B₁₆ | | | |
| 032C₁₆ | | | |
| 032D₁₆ | | | |
| 032E₁₆ | | | |
| 032F₁₆ | | | |
| 0330₁₆ | Interrupt request register | G1IR | 132 |
| 0331₁₆ | Interrupt enable register 0 | G1IE0 | 133 |
| 0332₁₆ | Interrupt enable register 1 | G1IE1 | 133 |
| 0333₁₆ | | | |
| 0334₁₆ | | | |
| 0335₁₆ | | | |
| 0336₁₆ | | | |
| 0337₁₆ | | | |
| 0338₁₆ | | | |
| 0339₁₆ | | | |
| 033A₁₆ | | | |
| 033B₁₆ | | | |
| 033C₁₆ | | | |
| 033D₁₆ | | | |
| 033E₁₆ | NMI digital debounce register | NDDR | 282 |
| 033F₁₆ | P17 digital debounce register | P17DDR | 282 |

Note 1: The blank areas are reserved and cannot be accessed by users.
Note 2: This register is included in the flash memory version.

# Quick Reference to Pages Classified by Address

| Address | Register | Symbol | Page |
|---|---|---|---|
| $0340_{16}$ | | | |
| $0341_{16}$ | | | |
| $0342_{16}$ $0343_{16}$ | Timer A1-1 register | TA11 | 114 |
| $0344_{16}$ $0345_{16}$ | Timer A2-1 register | TA21 | 114 |
| $0346_{16}$ $0347_{16}$ | Timer A4-1 register | TA41 | 114 |
| $0348_{16}$ | Three-phase PWM control register 0 | INVC0 | 111 |
| $0349_{16}$ | Three-phase PWM control register 1 | INVC1 | 112 |
| $034A_{16}$ | Three-phase output buffer register 0 | IDB0 | 113 |
| $034B_{16}$ | Three-phase output buffer register 1 | IDB1 | 113 |
| $034C_{16}$ | Dead time timer | DTT | 113 |
| $034D_{16}$ | Timer B2 interrupt occurrence frequency set counter | ICTB2 | 113 |
| $034E_{16}$ | Position-data-retain function contol register | PDRF | 121 |
| $034F_{16}$ | | | |
| $0350_{16}$ | | | |
| $0351_{16}$ | | | |
| $0352_{16}$ | | | |
| $0353_{16}$ | | | |
| $0354_{16}$ | | | |
| $0355_{16}$ | | | |
| $0356_{16}$ | | | |
| $0357_{16}$ | | | |
| $0358_{16}$ | | | |
| $0359_{16}$ | | | |
| $035A_{16}$ | | | |
| $035B_{16}$ | | | |
| $035C_{16}$ | | | |
| $035D_{16}$ | | | |
| $035E_{16}$ | Interrupt request cause select register 2 | IFSR2A | 61 |
| $035F_{16}$ | Interrupt request cause select register | IFSR | 61, 69 |
| $0360_{16}$ | SI/O3 transmit/receive register | S3TRR | 200 |
| $0361_{16}$ | | | |
| $0362_{16}$ | SI/O3 control register | S3C | 200 |
| $0363_{16}$ | SI/O3 bit rate generator | S3BRG | 200 |
| $0364_{16}$ | SI/O4 transmit/receive register | S4TRR | 200 |
| $0365_{16}$ | | | |
| $0366_{16}$ | SI/O4 control register | S4C | 200 |
| $0367_{16}$ | SI/O4 bit rate generator | S4BRG | 200 |
| $0368_{16}$ | | | |
| $0369_{16}$ | | | |
| $036A_{16}$ | | | |
| $036B_{16}$ | | | |
| $036C_{16}$ | | | |
| $036D_{16}$ | | | |
| $036E_{16}$ | | | |
| $036F_{16}$ | | | |
| $0370_{16}$ | | | |
| $0371_{16}$ | | | |
| $0372_{16}$ | | | |
| $0373_{16}$ | | | |
| $0374_{16}$ | UART2 special mode register 4 | U2SMR4 | 162 |
| $0375_{16}$ | UART2 special mode register 3 | U2SMR3 | 162 |
| $0376_{16}$ | UART2 special mode register 2 | U2SMR2 | 161 |
| $0377_{16}$ | UART2 special mode register | U2SMR | 161 |
| $0378_{16}$ | UART2 transmit/receive mode register | U2MR | 158 |
| $0379_{16}$ | UART2 bit rate generator | U2BRG | 157 |
| $037A_{16}$ $037B_{16}$ | UART2 transmit buffer register | U2TB | 157 |
| $037C_{16}$ | UART2 transmit/receive control register 0 | U2C0 | 159 |
| $037D_{16}$ | UART2 transmit/receive control register 1 | U2C1 | 160 |
| $037E_{16}$ $037F_{16}$ | UART2 receive buffer register | U2RB | 157 |

| Address | Register | Symbol | Page |
|---|---|---|---|
| $0380_{16}$ | Count start flag | TABSR | 88, 102, 116 |
| $0381_{16}$ | Clock prescaler reset flag | CPSRF | 89, 102 |
| $0382_{16}$ | One-shot start flag | ONSF | 89 |
| $0383_{16}$ | Trigger select register | TRGSR | 89, 116 |
| $0384_{16}$ | Up-down flag | UDF | 88 |
| $0385_{16}$ | | | |
| $0386_{16}$ $0387_{16}$ | Timer A0 register | TA0 | 88 |
| $0388_{16}$ $0389_{16}$ | Timer A1 register | TA1 | 88, 114 |
| $038A_{16}$ $038B_{16}$ | Timer A2 register | TA2 | 88, 114 |
| $038C_{16}$ $038D_{16}$ | Timer A3 register | TA3 | 88 |
| $038E_{16}$ $038F_{16}$ | Timer A4 register | TA4 | 88, 114 |
| $0390_{16}$ $0391_{16}$ | Timer B0 register | TB0 | 102 |
| $0392_{16}$ $0393_{16}$ | Timer B1 register | TB1 | 102 |
| $0394_{16}$ $0395_{16}$ | Timer B2 register | TB2 | 102, 116 |
| $0396_{16}$ | Timer A0 mode register | TA0MR | 87 |
| $0397_{16}$ | Timer A1 mode register | TA1MR | 87, 117 |
| $0398_{16}$ | Timer A2 mode register | TA2MR | 87, 117 |
| $0399_{16}$ | Timer A3 mode register | TA3MR | 87 |
| $039A_{16}$ | Timer A4 mode register | TA4MR | 87, 117 |
| $039B_{16}$ | Timer B0 mode register | TB0MR | 101 |
| $039C_{16}$ | Timer B1 mode register | TB1MR | 101 |
| $039D_{16}$ | Timer B2 mode register | TB2MR | 101, 117 |
| $039E_{16}$ | Timer B2 special mode register | TB2SC | 108, 115 |
| $039F_{16}$ | | | |
| $03A0_{16}$ | UART0 transmit/receive mode register | U0MR | 158 |
| $03A1_{16}$ | UART0 bit rate generator | U0BRG | 157 |
| $03A2_{16}$ $03A3_{16}$ | UART0 transmit buffer register | U0TB | 157 |
| $03A4_{16}$ | UART0 transmit/receive control register 0 | U0C0 | 159 |
| $03A5_{16}$ | UART0 transmit/receive control register 1 | U0C1 | 160 |
| $03A6_{16}$ $03A7_{16}$ | UART0 receive buffer register | U0RB | 157 |
| $03A8_{16}$ | UART1 transmit/receive mode register | U1MR | 158 |
| $03A9_{16}$ | UART1 bit rate generator | U1BRG | 157 |
| $03AA_{16}$ $03AB_{16}$ | UART1 transmit buffer register | U1TB | 157 |
| $03AC_{16}$ | UART1 transmit/receive control register 0 | U1C0 | 159 |
| $03AD_{16}$ | UART1 transmit/receive control register 1 | U1C1 | 160 |
| $03AE_{16}$ $03AF_{16}$ | UART1 receive buffer register | U1RB | 157 |
| $03B0_{16}$ | UART transmit/receive control register 2 | UCON | 159 |
| $03B1_{16}$ | | | |
| $03B2_{16}$ | | | |
| $03B3_{16}$ | | | |
| $03B4_{16}$ | | | |
| $03B5_{16}$ | | | |
| $03B6_{16}$ | | | |
| $03B7_{16}$ | | | |
| $03B8_{16}$ | DMA0 request cause select register | DM0SL | 77 |
| $03B9_{16}$ | | | |
| $03BA_{16}$ | DMA1 request cause select register | DM1SL | 78 |
| $03BB_{16}$ | | | |
| $03BC_{16}$ | | | |
| $03BD_{16}$ | | | |
| $03BE_{16}$ | | | |
| $03BF_{16}$ | | | |

Note : The blank areas are reserved and cannot be accessed by users.

# Quick Reference to Pages Classified by Address

| Address | Register | Symbol | Page |
|---|---|---|---|
| $03C0_{16}$<br>$03C1_{16}$ | A-D register 0 | AD0 | 208 |
| $03C2_{16}$<br>$03C3_{16}$ | A-D register 1 | AD1 | 208 |
| $03C4_{16}$<br>$03C5_{16}$ | A-D register 2 | AD2 | 208 |
| $03C6_{16}$<br>$03C7_{16}$ | A-D register 3 | AD3 | 208 |
| $03C8_{16}$<br>$03C9_{16}$ | A-D register 4 | AD4 | 208 |
| $03CA_{16}$<br>$03CB_{16}$ | A-D register 5 | AD5 | 208 |
| $03CC_{16}$<br>$03CD_{16}$ | A-D register 6 | AD6 | 208 |
| $03CE_{16}$<br>$03CF_{16}$ | A-D register 7 | AD7 | 208 |
| $03D0_{16}$ | | | |
| $03D1_{16}$ | | | |
| $03D2_{16}$ | A-D trigger control register | ADTRGCON | 207 |
| $03D3_{16}$ | A-D convert status register 0 | ADSTAT0 | 208 |
| $03D4_{16}$ | A-D control register 2 | ADCON2 | 206 |
| $03D5_{16}$ | | | |
| $03D6_{16}$ | A-D control register 0 | ADCON0 | 206 |
| $03D7_{16}$ | A-D control register 1 | ADCON1 | 206 |
| $03D8_{16}$ | | | |
| $03D9_{16}$ | | | |
| $03DA_{16}$ | | | |
| $03DB_{16}$ | | | |
| $03DC_{16}$ | | | |
| $03DD_{16}$ | | | |
| $03DE_{16}$ | | | |
| $03DF_{16}$ | | | |
| $03E0_{16}$ | Port P0 register | P0 | 279 |
| $03E1_{16}$ | Port P1 register | P1 | 279 |
| $03E2_{16}$ | Port P0 direction register | PD0 | 278 |
| $03E3_{16}$ | Port P1 direction register | PD1 | 278 |
| $03E4_{16}$ | Port P2 register | P2 | 279 |
| $03E5_{16}$ | Port P3 register | P3 | 279 |
| $03E6_{16}$ | Port P2 direction register | PD2 | 278 |
| $03E7_{16}$ | Port P3 direction register | PD3 | 278 |
| $03E8_{16}$ | | | |
| $03E9_{16}$ | | | |
| $03EA_{16}$ | | | |
| $03EB_{16}$ | | | |
| $03EC_{16}$ | Port P6 register | P6 | 279 |
| $03ED_{16}$ | Port P7 register | P7 | 279 |
| $03EE_{16}$ | Port P6 direction register | PD6 | 278 |
| $03EF_{16}$ | Port P7 direction register | PD7 | 278 |
| $03F0_{16}$ | Port P8 register | P8 | 279 |
| $03F1_{16}$ | Port P9 register | P9 | 279 |
| $03F2_{16}$ | Port P8 direction register | PD8 | 278 |
| $03F3_{16}$ | Port P9 direction register | PD9 | 278 |
| $03F4_{16}$ | Port P10 register | P10 | 279 |
| $03F5_{16}$ | | | |
| $03F6_{16}$ | Port P10 direction register | PD10 | 278 |
| $03F7_{16}$ | | | |
| $03F8_{16}$ | | | |
| $03F9_{16}$ | | | |
| $03FA_{16}$ | | | |
| $03FB_{16}$ | | | |
| $03FC_{16}$ | Pull-up control register 0 | PUR0 | 280 |
| $03FD_{16}$ | Pull-up control register 1 | PUR1 | 280 |
| $03FE_{16}$ | Pull-up control register 2 | PUR2 | 280 |
| $03FF_{16}$ | Port control register | PCR | 281 |

Note : The blank areas are reserved and cannot be accessed by users.

REJ09B0047-0060Z
Rev.0.60
2004.02.01

# 1. Overview

The M16C/28 group of single-chip microcomputers is built using the high-performance silicon gate CMOS process using a M16C/60 Series CPU core and is packaged in a 64-pin and 80-pin plastic molded QFP. These single-chip microcomputers operate using sophisticated instructions featuring a high level of instruction efficiency. With 1M bytes of address space, they are capable of executing instructions at high speed. In addition, this microcomputer contains a multiplier and a DMAC which combined with fast instruction processing capability, makes it suitable for control of various OA, communication, and industrial equipment which requires high-speed arithmetic/logic operations.

## 1.1 Applications

Audio, cameras, office/communications/portable/industrial equipment,
home appliances (inverter solution), etc

Specifications written in this manual are believed to be accurate, but are not guaranteed to be entirely free of error. Specifications in this manual may be changed for functional or performance improvements. Please make sure your manual is the latest edition.

## 1.2 Performance Outline

Table 1.2.1 lists performance outline of M16C/28 group 80-pin device.

Table 1.2.2 lists performance outline of M16C/28 group 64-pin device.

**Table 1.2.1. Performance outline of M16C/28 group (80-pin device)**

| Item | | Performance |
|---|---|---|
| Number of basic instructions | | 91 instructions |
| Shortest instruction execution time | | 50 ns (f(BCLK)= 20MHz, Vcc= 3.0V to 5.5V) (Normal-ver./T-ver.)<br>100 ns (f(BCLK)= 10MHz, Vcc= 2.7V to 5.5V) (Normal-ver.)<br>50 ns (f(BCLK)= 20MHz, Vcc= 4.2V to 5.5V -40 to 105°C) (V-ver.)<br>62.5 ns (f(BCLK)= 16MHz, Vcc= 4.2V to 5.5V -40 to 125°C) (V-ver.) |
| Memory capacity | ROM | (See the product list) |
| | RAM | (See the product list) |
| I/O port | | 71 lines |
| Multifunction timer | | TimerA:16 bits x 5 channels, TimerB:16 bits x 3 channels<br>Three-phase Motor Control Timer<br>TimerS (Input Capture/Output Compare)<br>: 16bit base timer x 1 channel (Input/Output x 8 channels) |
| Serial I/O | | 2 channels (UART0, UART1)<br>UART, clock synchronous<br>1 channel (UART2)<br>UART, clock synchronous, I2C bus[1], or IEBus[2]<br>2 channels (SI/O3, SI/O4)<br>Clock synchronous<br>1 channel (Multi-Master I2C bus[1]) |
| A-D converter | | 10 bits x 24 channels |
| DMAC | | 2 channels (trigger: 31 sources) |
| Watchdog timer | | 15 bits x 1 (with prescaler) |
| Interrupt | | 25 internal and 8 external sources, 4 software sources, 7 levels |
| Clock generation circuit | | 4 circuits<br>• Main clock, • Sub-clock (These circuits contain a built-in feedback resistor and external ceramic/quartz oscillator)<br>• Ring oscillator(main-clock oscillation stop detect function)<br>• PLL frequency synthesizer |
| Low voltage detection circuit | | Present |
| Power supply voltage | | Vcc=3.0V to 5.5V (f(BCLK)=20MHz) (Normal-ver.)<br>Vcc=2.7V to 5.5V (f(BCLK)=10MHz) |
| | | Vcc=3.0V to 5.5V (T-ver.) |
| | | Vcc=4.2V to 5.5V (V-ver.) |
| Flash memory | Program/erase voltage | 2.7V to 5.5V (Normal-ver.) 3.0V to 5.5V (T-ver.) 4.2V to 5.5V (V-ver.) |
| | Number of program/erase | 100 times ( Block A ,Block B : 10,000 times (option[3]) ) |
| Power consumption | | 16mA (Vcc=5V, f(BCLK)=20MHz)<br>25 µA (Vcc=3V, f(BCLK)=f(XCIN)=32KHz on RAM)<br>1.8 µA (Vcc=3V, f(BCLK)=f(XCIN)=32KHz, in wait mode)<br>0.7 µA (Vcc=3V, when stop mode) |
| Operating ambient temperature | | -20 to 85°C / -40 to 85°C (option[3]) (Normal-ver.) |
| | | -40 to 85°C (T-ver.) |
| | | -40 to 105°C / -40 to 125°C (V-ver.) |
| Package | | 80-pin plastic mold QFP |

Notes:
1. I2C Bus is a registered trademark of PHILIPS.
2. IEBus is a trademark of NEC Electronics Corporation.
3. If you desire this option, please so specify.

**Table 1.2.2. Performance outline of M16C/28 group (64-pin device)**

| Item | | Performance |
|---|---|---|
| Number of basic instructions | | 91 instructions |
| Shortest instruction execution time | | 50 ns (f(BCLK)= 20MHz, Vcc= 3.0V to 5.5V) (Normal-ver./T-ver.)<br>100 ns (f(BCLK)= 10MHz, Vcc= 2.7V to 5.5V) (Normal-ver.)<br>50 ns (f(BCLK)= 20MHz, Vcc= 4.2V to 5.5V -40 to 105°C) (V-ver.)<br>62.5 ns (f(BCLK)= 16MHz, Vcc= 4.2V to 5.5V -40 to 125°C) (V-ver.) |
| Memory capacity | ROM | (See the product list) |
| | RAM | (See the product list) |
| I/O port | | 55 lines |
| Multifunction timer | | TimerA:16 bits x 5 channels, TimerB:16 bits x 3 channels<br>Three-phase Motor Control Timer<br>TimerS (Input Capture/Output Compare)<br>: 16bit base timer x 1 channel (Input/Output x 8 channels) |
| Serial I/O | | 2 channels (UART0, UART1)<br>UART, clock synchronous<br>1 channel (UART2)<br>UART, clock synchronous, I2C bus1, or IEBus2<br>1 channel (SI/O3)<br>Clock synchronous<br>1 channel (Multi-Master I2C bus1) |
| A-D converter | | 10 bits x 13 channels |
| DMAC | | 2 channels (trigger: 30 sources) |
| Watchdog timer | | 15 bits x 1 (with prescaler) |
| Interrupt | | 24 internal and 8 external sources, 4 software sources, 7 levels |
| Clock generation circuit | | 4 circuits<br>• Main clock, • Sub-clock (These circuits contain a built-in feedback resistor and external ceramic/quartz oscillator)<br>• Ring oscillator(main-clock oscillation stop detect function)<br>• PLL frequency synthesizer |
| Low voltage detection circuit | | Present |
| Power supply voltage | | Vcc=3.0V to 5.5V (f(BCLK)=20MHz) (Normal-ver.)<br>Vcc=2.7V to 5.5V (f(BCLK)=10MHz) |
| | | Vcc=3.0V to 5.5V (T-ver.) |
| | | Vcc=4.2V to 5.5V (V-ver.) |
| Flash memory | Program/erase voltage | 2.7V to 5.5V (Normal-ver.) 3.0V to 5.5V (T-ver.) 4.2V to 5.5V (V-ver.) |
| | Number of program/erase | 100 times ( Block A ,Block B : 10,000 times (option3) ) |
| Power consumption | | 16mA (Vcc=5V, f(BCLK)=20MHz)<br>25 µA (Vcc=3V, f(BCLK)=f(XCIN)=32KHz on RAM)<br>1.8 µA (Vcc=3V, f(BCLK)=f(XCIN)=32KHz, in wait mode)<br>0.7 µA (Vcc=3V, when stop mode) |
| Operating ambient temperature | | -20 to 85°C / -40 to 85°C (option3) (Normal-ver.) |
| | | -40 to 85°C (T-ver.) |
| | | -40 to 105°C / -40 to 125°C (V-ver.) |
| Package | | 64-pin plastic mold QFP |

Notes:
1. I2C Bus is a registered trademark of PHILIPS.
2. IEBus is a trademark of NEC Electronics Corporation.
3. If you desire this option, please so specify.

## 1.3 Block Diagram

Figure 1.3.1 is a block diagram of the M16C/28 group, 80-pin device.

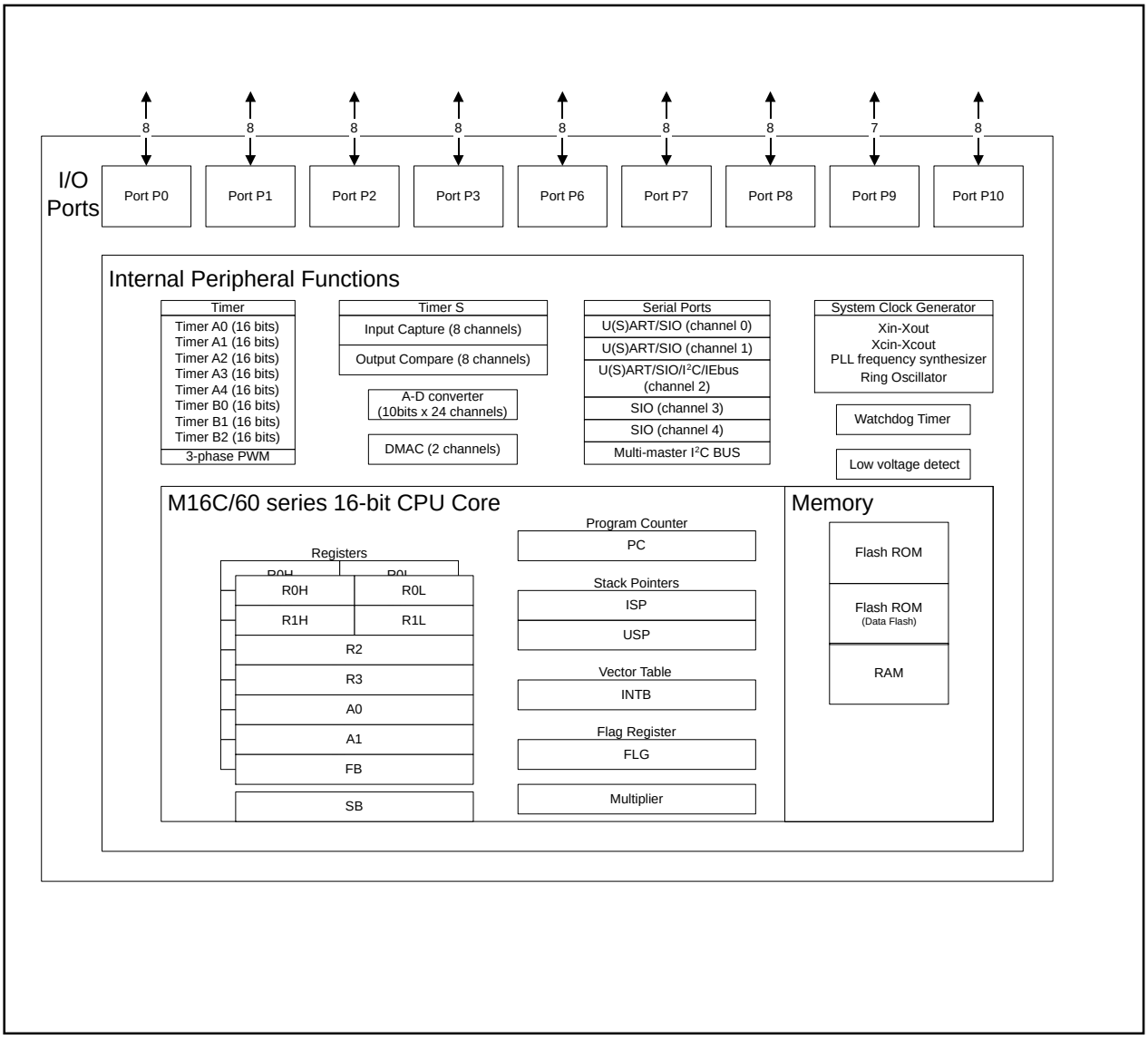



**Figure 1.3.1. M16C/28 Group, 80-pin Block Diagram**

Figure 1.3.2 is a block diagram of the M16C/28 group, 64-pin device.

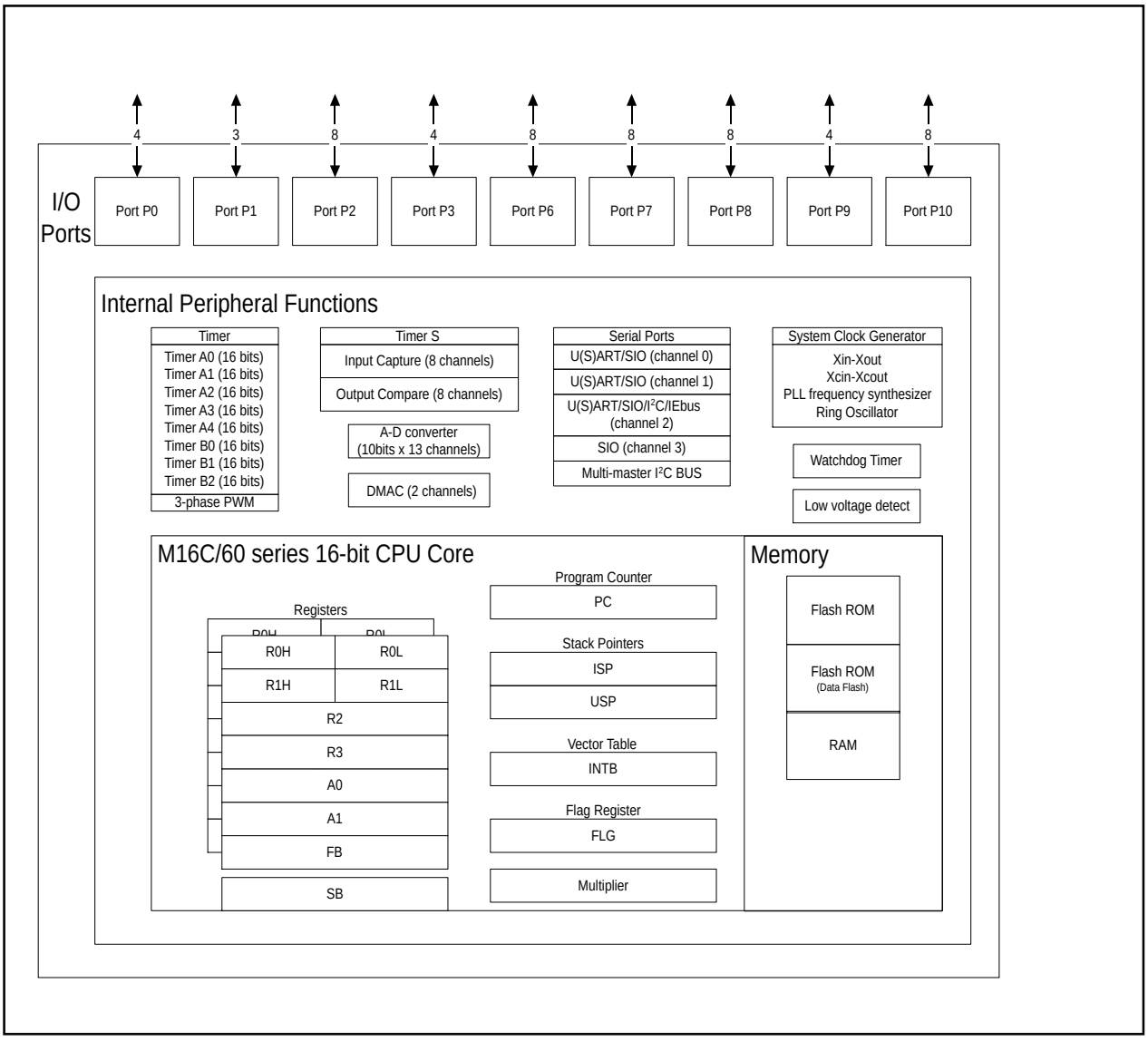



**Figure 1.3.2. M16C/28 Group, 64-pin Block Diagram**

## 1.4 Product List

Tables 1.4.1 to 1.4.3 list the M16C/28 group products and Figure 1.4.1 shows the type numbers, memory sizes and packages.

**Table 1.4.1. Product List (1) -Normal Version**

**As of November 2003**

| Type No. | | ROM capacity | RAM capacity | Package type | Remarks |
|---|---|---|---|---|---|
| M30280F6HP | ** | 48K + 4K byte | 4K byte | 80P6Q-A | Flash ROM Version |
| M30280F8HP | ** | 64K + 4K byte | 4K byte | | |
| M30280FAHP | ** | 96K + 4K byte | 8K byte | | |
| M30281F6HP | ** | 48K + 4K byte | 4K byte | 64P6Q-A | |
| M30281F8HP | ** | 64K + 4K byte | 4K byte | | |
| M30281FAHP | ** | 96K + 4K byte | 8K byte | | |
| M30280M4-XXXHP | * | 32K byte | 2K byte | 80P6Q-A | Mask ROM Version |
| M30280M6-XXXHP | * | 48K byte | 4K byte | | |
| M30280M8-XXXHP | * | 64K byte | 4K byte | | |
| M30281M4-XXXHP | * | 32K byte | 2K byte | 64P6Q-A | |
| M30281M6-XXXHP | * | 48K byte | 4K byte | | |
| M30281M8-XXXHP | * | 64K byte | 4K byte | | |

* : under planning

** : under development

**Table 1.4.2. Product List (2) -T Version**

**As of November 2003**

| Type No. | | ROM capacity | RAM capacity | Package type | Remarks |
|---|---|---|---|---|---|
| M30280F6THP | ** | 48K + 4K byte | 4K byte | 80P6Q-A | Flash ROM Version (T-version) |
| M30280F8THP | ** | 64K + 4K byte | 4K byte | | |
| M30280FATHP | ** | 96K + 4K byte | 8K byte | | |
| M30281F6THP | ** | 48K + 4K byte | 4K byte | 64P6Q-A | |
| M30281F8THP | ** | 64K + 4K byte | 4K byte | | |
| M30281FATHP | ** | 96K + 4K byte | 8K byte | | |
| M30280M4T-XXXHP | * | 32K byte | 2K byte | 80P6Q-A | Mask ROM Version (T-version) |
| M30280M6T-XXXHP | * | 48K byte | 4K byte | | |
| M30280M8T-XXXHP | * | 64K byte | 4K byte | | |
| M30281M4T-XXXHP | * | 32K byte | 2K byte | 64P6Q-A | |
| M30281M6T-XXXHP | * | 48K byte | 4K byte | | |
| M30281M8T-XXXHP | * | 64K byte | 4K byte | | |

* : under planning

** : under development

**Table 1.4.2. Product List (3) -V Version**

**As of November 2003**

| Type No. | ROM capacity | RAM capacity | Package type | Remarks |
|---|---|---|---|---|
| M30280F6VHP ** | 48K + 4K byte | 4K byte | 80P6Q-A | Flash ROM Version (V-version) |
| M30280F8VHP ** | 64K + 4K byte | 4K byte | | |
| M30280FAVHP ** | 96K + 4K byte | 8K byte | | |
| M30281F6VHP ** | 48K + 4K byte | 4K byte | 64P6Q-A | |
| M30281F8VHP ** | 64K + 4K byte | 4K byte | | |
| M30281FAVHP ** | 96K + 4K byte | 8K byte | | |
| M30280M4V-XXXHP * | 32K byte | 2K byte | 80P6Q-A | Mask ROM Version (V-version) |
| M30280M6V-XXXHP * | 48K byte | 4K byte | | |
| M30280M8V-XXXHP * | 64K byte | 4K byte | | |
| M30281M4V-XXXHP * | 32K byte | 2K byte | 64P6Q-A | |
| M30281M6V-XXXHP * | 48K byte | 4K byte | | |
| M30281M8V-XXXHP * | 64K byte | 4K byte | | |

* : under planning

** : under development

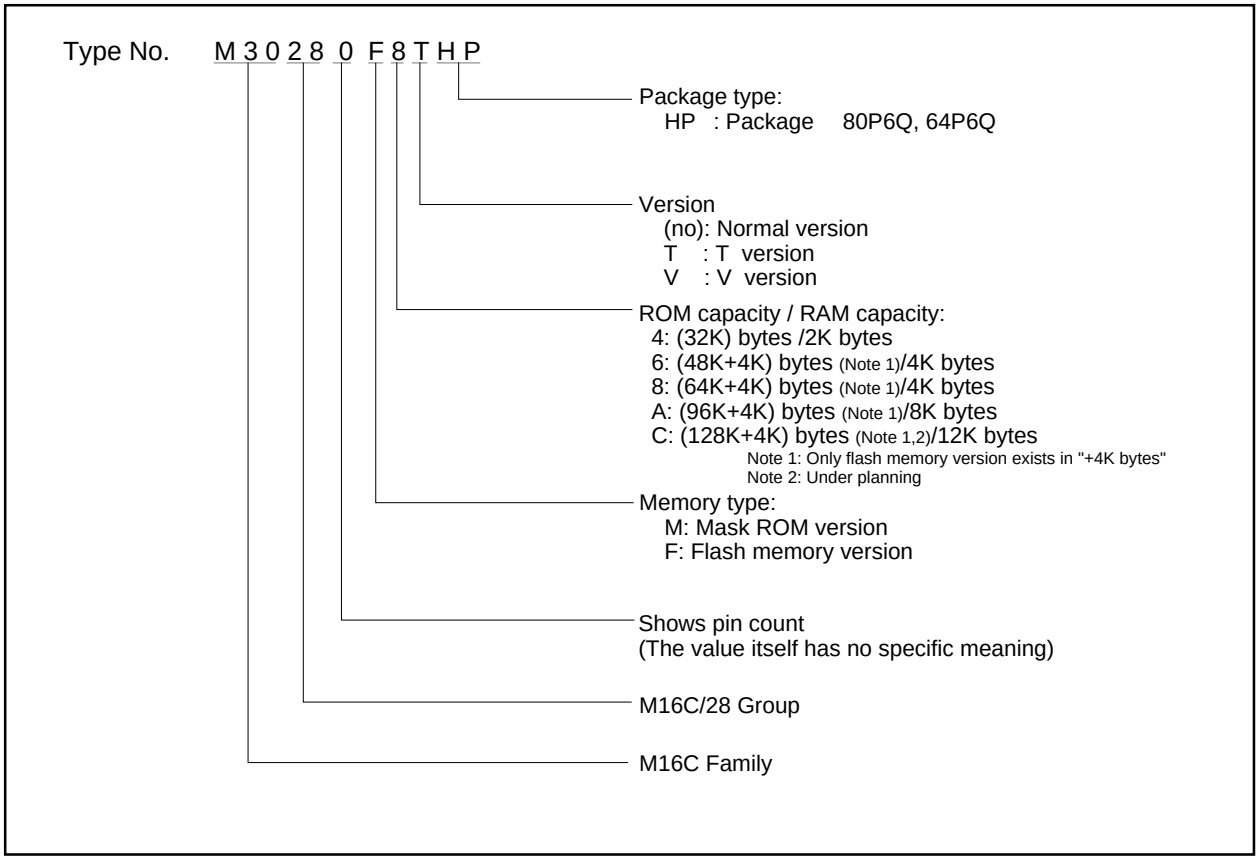
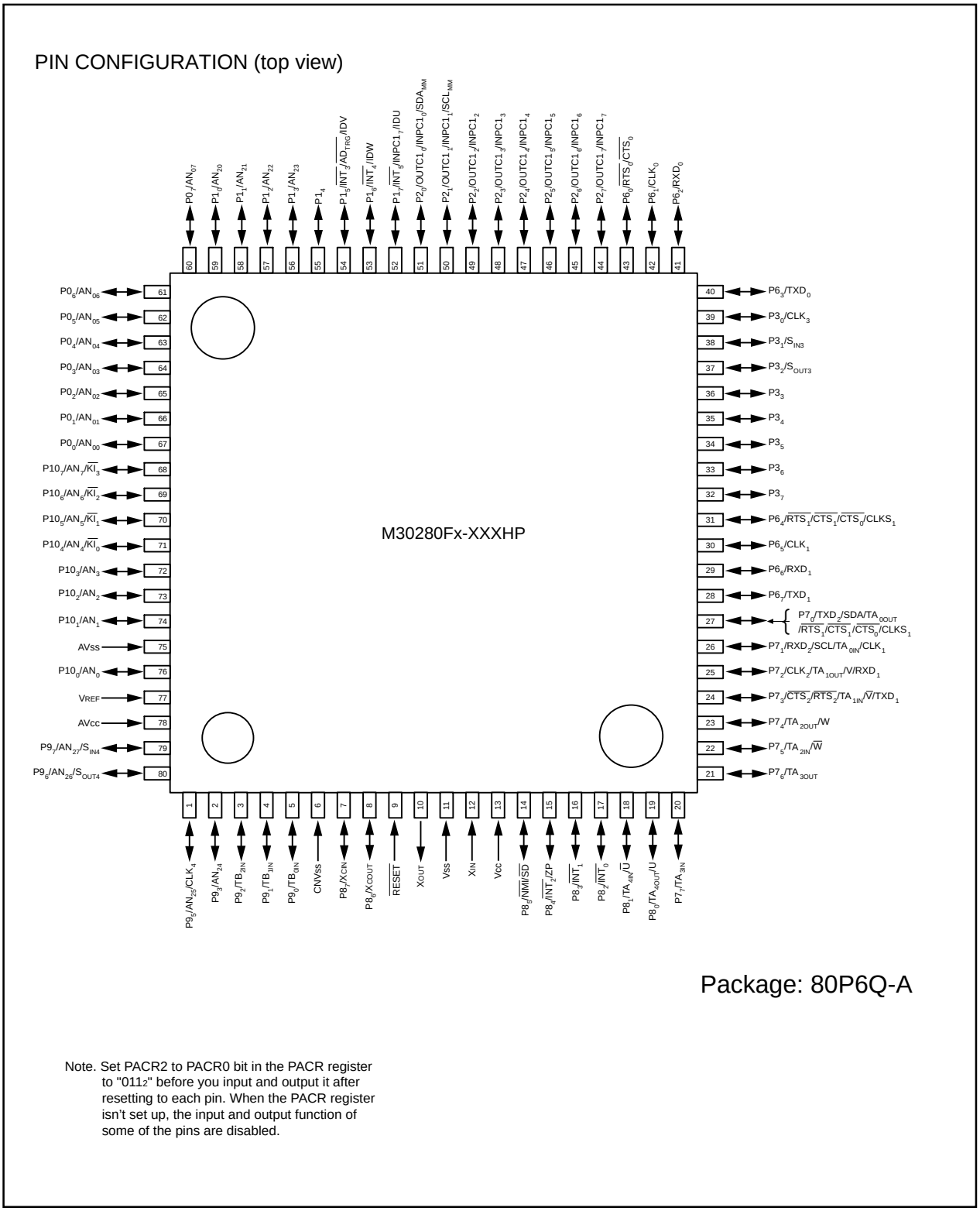



**Figure 1.4.1. Type No., Memory Size, and Package**

## 1.5 Pin Configuration

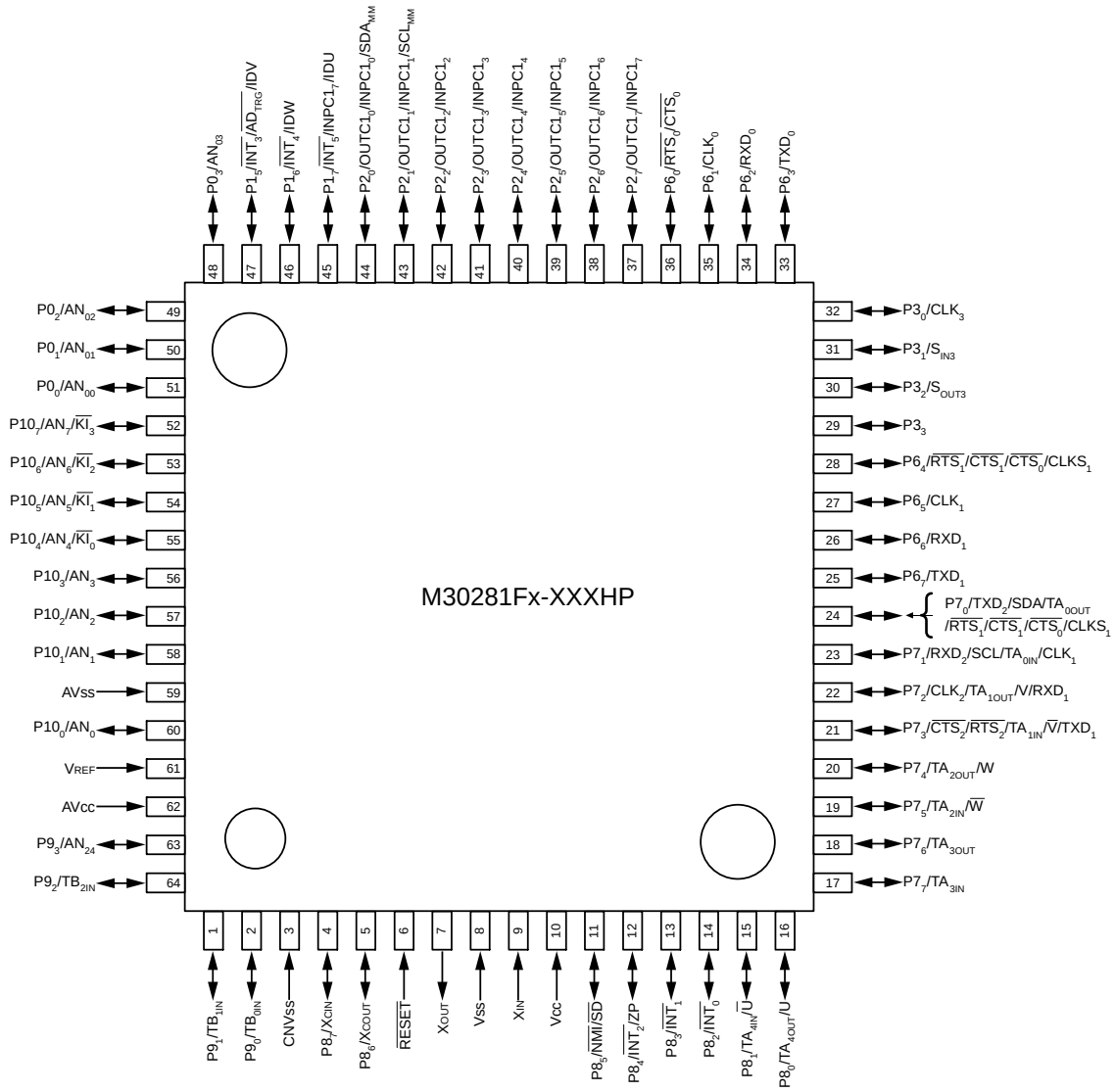
Figures 1.5.1 and 1.5.2 show the pin configurations (top view).




Note. Set PACR2 to PACR0 bit in the PACR register to "011₂" before you input and output it after resetting to each pin. When the PACR register isn't set up, the input and output function of some of the pins are disabled.

**Figure 1.5.1. Pin Configuration (Top View) of M16C/28 Group, 80-pin Package**

PIN CONFIGURATION (top view)




Package: 64P6Q-A

Note. Set PACR2 to PACR0 bit in the PACR register to "010₂" before you input and output it after resetting to each pin. When the PACR register isn't set up, the input and output function of some of the pins are disabled.

**Figure 1.5.2. Pin Configuration (Top View) of M16C/28 Group, 64-pin Package**

## 1.6 Pin Description

Table 1.6.1 and 1.6.2 describes the available pins.

**Table 1.6.1 Pin Description(1)**

| Pin name | Signal name | I/O type | Function |
|---|---|---|---|
| $V_{CC}$,$V_{SS}$ | Power supply input | | Apply 0V to the Vss pin, and the following voltage to the Vcc pin.<br>2.7 to 5.5V (Normal-ver.)<br>3.0 to 5.5V (T-ver.)<br>4.2 to 5.5V (V-ver.) |
| $CNV_{SS}$ | $CNV_{SS}$ | Input | Connect this pin to Vss. |
| $\overline{RESET}$ | Reset input | Input | "L" on this input resets the microcomputer. |
| $X_{IN}$<br>$X_{OUT}$ | Clock input<br>Clock output | Input<br>Output | These pins are provided for the main clock generating circuit input/output. Connect a ceramic resonator or crystal between the $X_{IN}$ and the $X_{OUT}$ pins. To use an externally derived clock, input it to the $X_{IN}$ pin and leave the $X_{OUT}$ pin open. If $X_{IN}$ is not used (for external oscillator or external clock) connect $X_{IN}$ pin to $V_{CC}$ and leave $X_{OUT}$ pin open. |
| $AV_{CC}$ | Analog power supply input | | This pin is a power supply input for the A-D converter. Connect this pin to $V_{CC}$. |
| $AV_{SS}$ | Analog power supply input | | This pin is a power supply input for the A-D converter. Connect this pin to $V_{SS}$. |
| $V_{REF}$ | Reference Voltage input | Input | This pin is a reference voltage input for the A-D converter. |
| $P0_0$~$P0_7$ | I/O port P0 | Input/output | This is an 8-bit CMOS I/O port. It has an input/output port direction register that allows the user to set each pin for input or output individually. When used for input, a pull-up resister option can be selected for the entire group of four pins.Software can also select this port to function as A-D converter input pins. $P0_4$~$P0_7$ is not in 64 pin version. |
| $P1_0$~$P1_7$ | I/O port P1 | Input/output | This is an 8-bit I/O port equivalent to P0. Additional software-select able secondary functions are: 1) $P1_0$ to $P1_3$ can act as A-D converter input pins; 2) $P1_5$ to $P1_7$ can be configured as external interrupt pins; 3) $P1_5$ to $P1_7$ can be configured as position-data-retain function input pins,and; 4) $P1_5$ can input a trigger for the A-D converter. $P1_0$~$P1_4$ is not in 64 pin version. |
| $P2_0$~$P2_7$ | I/O port P2 | Input/output | This is an 8-bit I/O port equivalent to P0. Software can also select this port to perform as I/O for the Timer S (all pins), and MultiMaster $I^2C$ Bus ($P2_0$ and $P2_1$ only) |
| $P3_0$~$P3_7$ | I/O port P3 | Input/output | This is an 8-bit I/O port equivalent to P0. $P3_0$ to $P3_2$ also function as SIO3 I/O, as selected by software. $P3_4$~$P3_7$ is not in 64 pin version. |
| $P6_0$~$P6_7$ | I/O port P6 | Input/output | This is an 8-bit I/O port equivalent to P0. Pins in this port also function as UART0 and UART1 I/O, as selected by software. |
| $P7_0$~$P7_7$ | I/O port P7 | Input/output | This is an 8-bit I/O port equivalent to P0. P7 can also function as I/O for timer A0-A3, as selected by software. Additional programming options are: $P7_0$ to $P7_3$ can assume UART1 and UART2 I/O capabilities, and $P7_2$ to $P7_5$ can function as output pins for the three-phase motor control timer. |

**Table 1.6.2 Pin Description(2)**

| Pin name | Signal name | I/O type | Function |
|---|---|---|---|
| $P8_0$~$P8_7$ | I/O port P8 | Input/output | This is an 8-bit I/O port equivalent to P0. Additional software-select able secondary functions are: 1) $P8_0$ and $P8_1$ can act as either I/O for Timer A4, or as output pins for the three-phase motor control timer; 2) $P8_2$ to $P8_4$ can be configured as external interrupt pins. $P8_4$ can be used for Timer A Zphase function; 3) $P8_5$ can be used as $\overline{NMI}/\overline{SD}$. $P8_5$ can not be used as I/O port while the three-phase motor control is enabled. Apply a stable "H" to $P8_5$ after setting the direction register for $P8_5$ to "0" when the three-phase motor control is enabled, and; 4) $P8_6$ and $P8_7$ can serve as I/O pins for the sub-clock generation circuit. In this latter case, a quartz oscillator must be connented between $P8_6$ ($X_{COUT}$ pin) and $P8_7$ ($X_{CIN}$ pin). |
| $P9_0$~$P9_3$, $P9_5$~$P9_7$ | I/O port P9 | Input/output | This is an 7-bit I/O port equivalent to P0. Additional software-select able secondary functions are: 1) $P9_0$ to $P9_2$ can act as Timer B0~B2 input pins; 2) $P9_3$, $P9_5$ to $P9_7$ can act as A-D converter input pins, and; 3) $P9_6$ to $P9_7$ can assume SI/O4 I/O. $P9_5$ to $P9_7$ is not in 64 pin version. |
| $P10_0$~$P10_7$ | I/O port P10 | Input/output | This is an 8-bit I/O port equivalent to P0. This port can also function as A-D converter input pins, as selected by software. Furthermore, $P10_4$-$P10_7$ can also function as input pins for the key input interrupt function. |

# 2. Central Processing Unit (CPU)

Figure 2.1 shows the CPU registers. The CPU has 13 registers. Of these, R0, R1, R2, R3, A0, A1 and FB comprise a register bank. There are two register banks.

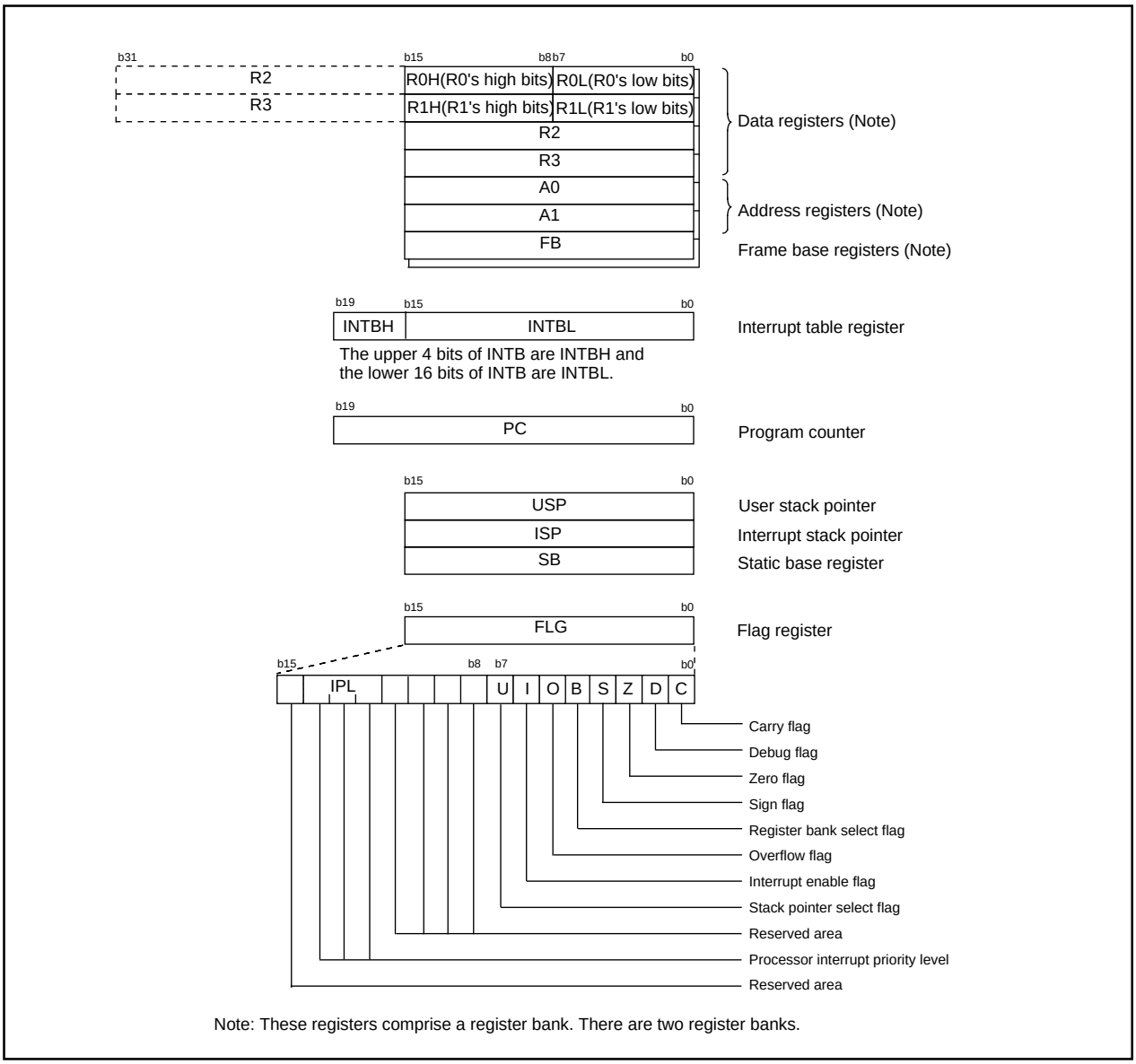



**Figure 2.1. Central Processing Unit Register**

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

The R0 register consists of 16 bits, and is used mainly for transfers and arithmetic/logic operations. R1 to R3 are the same as R0.

The R0 register can be separated between high (R0H) and low (R0L) for use as two 8-bit data registers. R1H and R1L are the same as R0H and R0L. Conversely, R2 and R0 can be combined for use as a 32-bit data register (R2R0). R3R1 is the same as R2R0.

## 2.2 Address Registers (A0 and A1)

The register A0 consists of 16 bits, and is used for address register indirect addressing and address register relative addressing. They also are used for transfers and arithmetic/logic operations. A1 is the same as A0.

In some instructions, registers A1 and A0 can be combined for use as a 32-bit address register (A1A0).

## 2.3 Frame Base Register (FB)

FB is configured with 16 bits, and is used for FB relative addressing.

## 2.4 Interrupt Table Register (INTB)

INTB is configured with 20 bits, indicating the start address of an interrupt vector table.

## 2.5 Program Counter (PC)

PC is configured with 20 bits, indicating the address of an instruction to be executed.

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

Stack pointer (SP) comes in two types: USP and ISP, each configured with 16 bits.

Your desired type of stack pointer (USP or ISP) can be selected by the U flag of FLG.

## 2.7 Static Base Register (SB)

SB is configured with 16 bits, and is used for SB relative addressing.

## 2.8 Flag Register (FLG)

FLG consists of 11 bits, indicating the CPU status.

### 2.8.1 Carry Flag (C Flag)

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

### 2.8.2 Debug Flag (D Flag)

The D flag is used exclusively for debugging purpose. During normal use, it must be set to “0”.

### 2.8.3 Zero Flag (Z Flag)

This flag is set to “1” when an arithmetic operation resulted in 0; otherwise, it is “0”.

### 2.8.4 Sign Flag (S Flag)

This flag is set to “1” when an arithmetic operation resulted in a negative value; otherwise, it is “0”.

### 2.8.5 Register Bank Select Flag (B Flag)

Register bank 0 is selected when this flag is “0” ; register bank 1 is selected when this flag is “1”.

### 2.8.6 Overflow Flag (O Flag)

This flag is set to “1” when the operation resulted in an overflow; otherwise, it is “0”.

### 2.8.7 Interrupt Enable Flag (I Flag)

This flag enables a maskable interrupt.

Maskable interrupts are disabled when the I flag is “0”, and are enabled when the I flag is “1”. The I flag is cleared to “0” when the interrupt request is accepted.

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

ISP is selected when the U flag is “0”; USP is selected when the U flag is “1”.

The U flag is cleared to “0” when a hardware interrupt request is accepted or an INT instruction for software interrupt Nos. 0 to 31 is executed.

### 2.8.9 Processor Interrupt Priority Level (IPL)

IPL is configured with three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.

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

### 2.8.10 Reserved Area

When write to this bit, write "0". When read, its content is indeterminate.

# 3. Memory

Figure 3.1 is a memory map of the M16C/28 group. The linear address space of 1M bytes extends from address 00000$_{16}$ to FFFFF$_{16}$. From FFFFF$_{16}$ down is ROM. For example, in the M30280F8HP,there are 64 Kbytes of internal ROM from F0000$_{16}$ to FFFFF$_{16}$.

The vector table for fixed interrupts, such as Reset and NMI, is mapped from FFFDC$_{16}$ to FFFFF$_{16}$. The starting address of the interrupt routine is stored here.

The address of the vector table for timer interrupts,etc.,can be set as desired using the interrupt table register(INTB). See the section on interrupts for details.

From 00400$_{16}$ up is RAM. For example, in the M30280FAHP, 8K bytes of internal RAM is mapped to the space from 00400$_{16}$ to 023FF$_{16}$. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.

These devices also contain two blocks of Flash ROM as Data Flash memory to store data. These two blocks of 2K bytes are located from 0F000$_{16}$ to 0FFFF$_{16}$ on all versions.

The SFR area is mapped from 00000$_{16}$ to 003FF$_{16}$. This area accommodates the control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers, etc. Any part of the SFR area that is not occupied is reserved and cannot be used for other purposes.

The special page vector table is allocated to the address from FFE00$_{16}$ to FFFDB$_{16}$. This vector is used by the JMPS or JSRS instruction. For details, refer to the "M16C/60 and M16C/20 Series Software Manual".

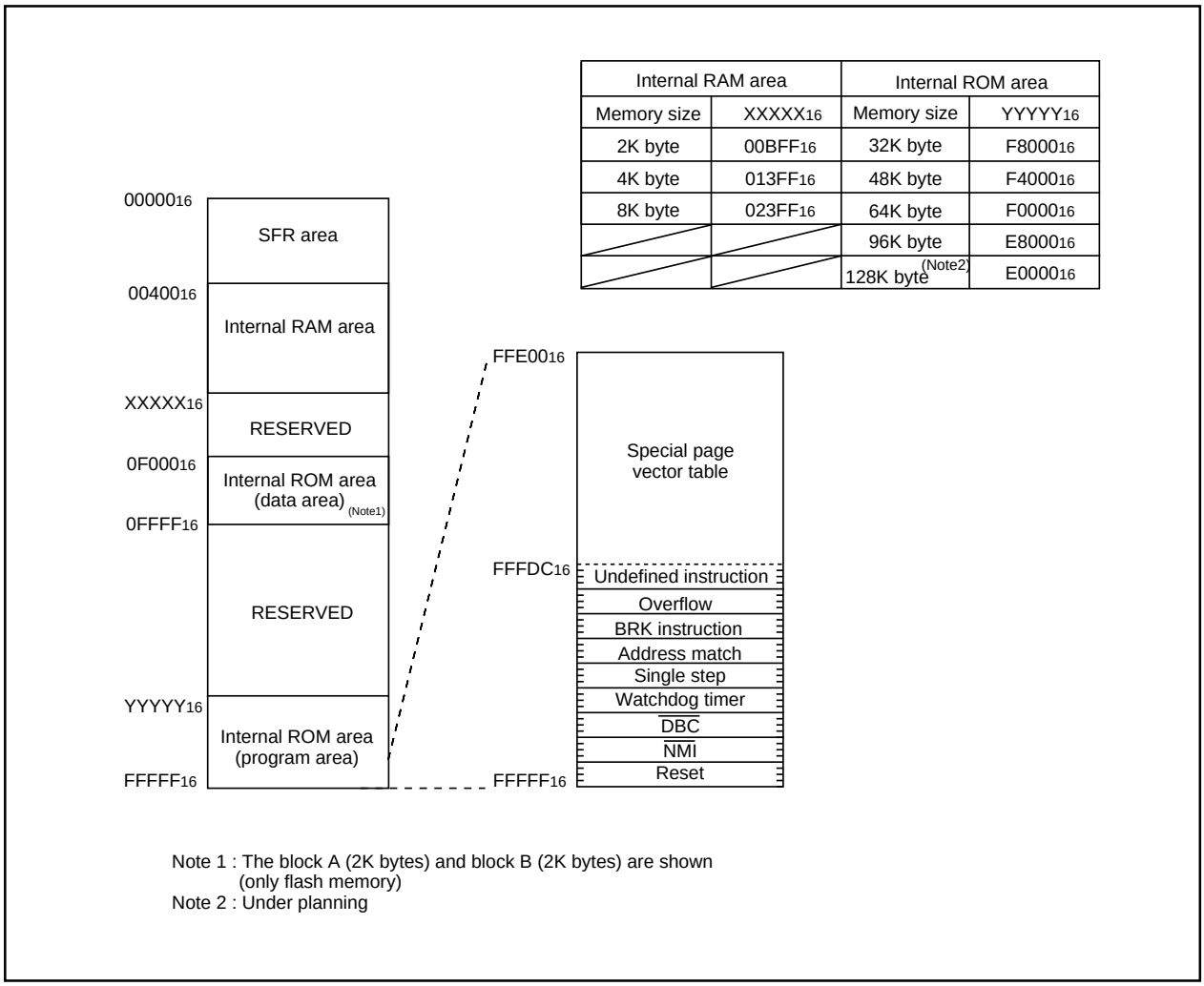



**Figure 3.1. Memory Map**

# 4. Special Function Register (SFR) Map

| Address | Register Name | Acronym | Value after Reset |
|---|---|---|---|
| 0000₁₆ | | | |
| 0001₁₆ | | | |
| 0002₁₆ | | | |
| 0003₁₆ | | | |
| 0004₁₆ | Processor mode register 0 | PM0 | 00₁₆ |
| 0005₁₆ | Processor mode register 1 | PM1 | 00001000₂ |
| 0006₁₆ | System clock control register 0 | CM0 | 01001000₂ |
| 0007₁₆ | System clock control register 1 | CM1 | 00100000₂ |
| 0008₁₆ | | | |
| 0009₁₆ | Address match interrupt enable register | AIER | XXXXXX00₂ |
| 000A₁₆ | Protect register | PRCR | XX000000₂ |
| 000B₁₆ | | | |
| 000C₁₆ | Oscillation stop detection register (Note 2) | CM2 | 0X000010₂ |
| 000D₁₆ | | | |
| 000E₁₆ | Watchdog timer start register | WDTS | ??₁₆ |
| 000F₁₆ | Watchdog timer control register | WDC | 00??????₂(Note 3) |
| 0010₁₆ | Address match interrupt register 0 | RMAD0 | 00₁₆ |
| 0011₁₆ | | | 00₁₆ |
| 0012₁₆ | | | X0₁₆ |
| 0013₁₆ | | | |
| 0014₁₆ | Address match interrupt register 1 | RMAD1 | 00₁₆ |
| 0015₁₆ | | | 00₁₆ |
| 0016₁₆ | | | X0₁₆ |
| 0017₁₆ | | | |
| 0018₁₆ | | | |
| 0019₁₆ | Voltage detection register 1 (Note 4) | VCR1 | 00001000₂ |
| 001A₁₆ | Voltage detection register 2 (Note 4) | VCR2 | 00₁₆ |
| 001B₁₆ | | | |
| 001C₁₆ | PLL control register 0 | PLC0 | 0001X010₂ |
| 001D₁₆ | | | |
| 001E₁₆ | Processor mode register 2 | PM2 | XXX00000₂ |
| 001F₁₆ | Voltage down detection interrupt register | D4INT | 00₁₆ |
| 0020₁₆ | DMA0 source pointer | SAR0 | ??₁₆ |
| 0021₁₆ | | | ??₁₆ |
| 0022₁₆ | | | X?₁₆ |
| 0023₁₆ | | | |
| 0024₁₆ | DMA0 destination pointer | DAR0 | ??₁₆ |
| 0025₁₆ | | | ??₁₆ |
| 0026₁₆ | | | X?₁₆ |
| 0027₁₆ | | | |
| 0028₁₆ | DMA0 transfer counter | TCR0 | ??₁₆ |
| 0029₁₆ | | | ??₁₆ |
| 002A₁₆ | | | |
| 002B₁₆ | | | |
| 002C₁₆ | DMA0 control register | DM0CON | 00000?00₂ |
| 002D₁₆ | | | |
| 002E₁₆ | | | |
| 002F₁₆ | | | |
| 0030₁₆ | DMA1 source pointer | SAR1 | ??₁₆ |
| 0031₁₆ | | | ??₁₆ |
| 0032₁₆ | | | X?₁₆ |
| 0033₁₆ | | | |
| 0034₁₆ | DMA1 destination pointer | DAR1 | ??₁₆ |
| 0035₁₆ | | | ??₁₆ |
| 0036₁₆ | | | X?₁₆ |
| 0037₁₆ | | | |
| 0038₁₆ | DMA1 transfer counter | TCR1 | ??₁₆ |
| 0039₁₆ | | | ??₁₆ |
| 003A₁₆ | | | |
| 003B₁₆ | | | |
| 003C₁₆ | DMA1 control register | DM1CON | 00000?00₂ |
| 003D₁₆ | | | |
| 003E₁₆ | | | |
| 003F₁₆ | | | |

Note 1: The blank areas are reserved and cannot be accessed by users.
Note 2: The CM20, CM21, and CM27 bits do not change at oscillation stop detection reset.
Note 3: Tjhe WDC5 bit is "0" (cold start) immediately after power-on. It can only be set to "1" in a program. It is set to "0" when the input voltage at the Vcc pin drops to Vdet2 or less while the VC25 bit in the VCR2 register is set to "1"(RAM retention limit detection circuit enable).
Note 4: This register does not change at software reset, watchdog timer reset and oscillation stop detection reset.

X : Noting is mapped to this bit
? : Value indeterminate at reset

**Figure 4.1. SFR Map (1 of 7)**

| Address | Register Name | Acronym | Value after Reset |
|---|---|---|---|
| 0040₁₆ | | | |
| 0041₁₆ | | | |
| 0042₁₆ | | | |
| 0043₁₆ | | | |
| 0044₁₆ | INT3 interrupt control register | INT3IC | XX00?000₂ |
| 0045₁₆ | IC/OC 0 interrupt control register | ICOC0IC | XXXX?000₂ |
| 0046₁₆ | IC/OC 1 interrupt control register, I²C-BUS interface interrupt control register | ICOC1IC, IICIC | XXXX?000₂ |
| 0047₁₆ | IC/OC base timer interrupt control register, SCLSDA interrupt control register | BTIC, SCLDAIC | XXXX?000₂ |
| 0048₁₆ | SI/O4 interrupt control register, INT5 interrupt control register | S4IC, INT5IC | XX00?000₂ |
| 0049₁₆ | SI/O3 interrupt control register, INT4 interrupt control register | S3IC, INT4IC | XX00?000₂ |
| 004A₁₆ | UART2 Bus collision detection interrupt control register | BCNIC | XXXX?000₂ |
| 004B₁₆ | DMA0 interrupt control register | DM0IC | XXXX?000₂ |
| 004C₁₆ | DMA1 interrupt control register | DM1IC | XXXX?000₂ |
| 004D₁₆ | Key input interrupt control register | KUPIC | XXXX?000₂ |
| 004E₁₆ | A-D conversion interrupt control register | ADIC | XXXX?000₂ |
| 004F₁₆ | UART2 transmit interrupt control register | S2TIC | XXXX?000₂ |
| 0050₁₆ | UART2 receive interrupt control register | S2RIC | XXXX?000₂ |
| 0051₁₆ | UART0 transmit interrupt control register | S0TIC | XXXX?000₂ |
| 0052₁₆ | UART0 receive interrupt control register | S0RIC | XXXX?000₂ |
| 0053₁₆ | UART1 transmit interrupt control register | S1TIC | XXXX?000₂ |
| 0054₁₆ | UART1 receive interrupt control register | S1RIC | XXXX?000₂ |
| 0055₁₆ | Timer A0 interrupt control register | TA0IC | XXXX?000₂ |
| 0056₁₆ | Timer A1 interrupt control register | TA1IC | XXXX?000₂ |
| 0057₁₆ | Timer A2 interrupt control register | TA2IC | XXXX?000₂ |
| 0058₁₆ | Timer A3 interrupt control register | TA3IC | XXXX?000₂ |
| 0059₁₆ | Timer A4 interrupt control register | TA4IC | XXXX?000₂ |
| 005A₁₆ | Timer B0 interrupt control register | TB0IC | XXXX?000₂ |
| 005B₁₆ | Timer B1 interrupt control register | TB1IC | XXXX?000₂ |
| 005C₁₆ | Timer B2 interrupt control register | TB2IC | XXXX?000₂ |
| 005D₁₆ | INT0 interrupt control register | INT0IC | XX00?000₂ |
| 005E₁₆ | INT1 interrupt control register | INT1IC | XX00?000₂ |
| 005F₁₆ | INT2 interrupt control register | INT2IC | XX00?000₂ |
| 0060₁₆ | | | |
| 0061₁₆ | | | |
| 0062₁₆ | | | |
| 0063₁₆ | | | |
| 0064₁₆ | | | |
| 0065₁₆ | | | |
| 0066₁₆ | | | |
| 0067₁₆ | | | |
| 0068₁₆ | | | |
| 0069₁₆ | | | |
| 006A₁₆ | | | |
| 006B₁₆ | | | |
| 006C₁₆ | | | |
| 006D₁₆ | | | |
| 006E₁₆ | | | |
| 006F₁₆ | | | |
| 0070₁₆ | | | |
| 0071₁₆ | | | |
| 0072₁₆ | | | |
| 0073₁₆ | | | |
| 0074₁₆ | | | |
| 0075₁₆ | | | |
| 0076₁₆ | | | |
| 0077₁₆ | | | |
| 0078₁₆ | | | |
| 0079₁₆ | | | |
| 007A₁₆ | | | |
| 007B₁₆ | | | |
| 007C₁₆ | | | |
| 007D₁₆ | | | |
| 007E₁₆ | | | |
| 007F₁₆ | | | |

Note 1: The blank areas are reserved and cannot be accessed by users.

X : Noting is mapped to this bit
? : Value indeterminate at reset

**Figure 4.2. SFR Map (2 of 7)**

| Address | Register Name | Acronym | Value after Reset |
|---|---|---|---|
| 0080₁₆ | | | |
| 0081₁₆ | | | |
| 0082₁₆ | | | |
| 0083₁₆ | | | |
| 0084₁₆ | | | |
| 0085₁₆ | | | |
| 0086₁₆ | | | |
| ≈ | | | ≈ |
| 01B0₁₆ | | | |
| 01B1₁₆ | | | |
| 01B2₁₆ | | | |
| 01B3₁₆ | Flash memory control register 4 (Note 2) | FMR4 | 01000000₂ |
| 01B4₁₆ | | | |
| 01B5₁₆ | Flash memory control register 1 (Note 2) | FMR1 | 000???0?₂ |
| 01B6₁₆ | | | |
| 01B7₁₆ | Flash memory control register 0 (Note 2) | FMR0 | ??000001₂ |
| 01B8₁₆ | | | |
| 01B9₁₆ | | | |
| 01BA₁₆ | | | |
| 01BB₁₆ | | | |
| 01BC₁₆ | | | |
| 01BD₁₆ | | | |
| 01BE₁₆ | | | |
| 01BF₁₆ | | | |
| ≈ | | | ≈ |
| 0250₁₆ | | | |
| 0251₁₆ | | | |
| 0252₁₆ | | | |
| 0253₁₆ | | | |
| 0254₁₆ | | | |
| 0255₁₆ | | | |
| 0256₁₆ | | | |
| 0257₁₆ | | | |
| 0258₁₆ | | | |
| 0259₁₆ | | | |
| 025A₁₆ | | | |
| 025B₁₆ | | | |
| 025C₁₆ | Ring oscillator control register | ROCR | 00000101₂ |
| 025D₁₆ | Pin assignment control register | PACR | 00₁₆ |
| 025E₁₆ | Peripheral clock select register | PCLKR | 00000011₂ |
| 025F₁₆ | | | |
| ≈ | | | ≈ |
| 02E0₁₆ | $I^2C0$ data shift register | S00 | ??₁₆ |
| 02E1₁₆ | | | |
| 02E2₁₆ | $I^2C0$ address register | S0D0 | 00₁₆ |
| 02E3₁₆ | $I^2C0$ control register 0 | S1D0 | 00₁₆ |
| 02E4₁₆ | $I^2C0$ clock control register | S20 | 00₁₆ |
| 02E5₁₆ | $I^2C0$ start/stop condition control register | S2D0 | 00011010₂ |
| 02E6₁₆ | $I^2C0$ control register 1 | S3D0 | 00110000₂ |
| 02E7₁₆ | $I^2C0$ control register 2 | S4D0 | 00₁₆ |
| 02E8₁₆ | $I^2C0$ status register | S10 | 0001000X₂ |
| 02E9₁₆ | | | |
| 02EA₁₆ | | | |
| ≈ | | | ≈ |
| 02FE₁₆ | | | |
| 02FF₁₆ | | | |

Note 1:The blank areas are reserved and cannot be accessed by users.
Note 2:This register is included in the flash memory version.

X : Noting is mapped to this bit
? : Value indeterminate at reset

**Figure 4.3. SFR Map (3 of 7)**

| Address | Register Name | Acronym | Value after Reset |
|---|---|---|---|
| 0300₁₆<br>0301₁₆ | TM/WG register 0 | G1TM0/G1PO0 | ??₁₆<br>??₁₆ |
| 0302₁₆<br>0303₁₆ | TM/WG register 1 | G1TM1/G1PO1 | ??₁₆<br>??₁₆ |
| 0304₁₆<br>0305₁₆ | TM/WG register 2 | G1TM2/G1PO2 | ??₁₆<br>??₁₆ |
| 0306₁₆<br>0307₁₆ | TM/WG register 3 | G1TM3/G1PO3 | ??₁₆<br>??₁₆ |
| 0308₁₆<br>0309₁₆ | TM/WG register 4 | G1TM4/G1PO4 | ??₁₆<br>??₁₆ |
| 030A₁₆<br>030B₁₆ | TM/WG register 5 | G1TM5/G1PO5 | ??₁₆<br>??₁₆ |
| 030C₁₆<br>030D₁₆ | TM/WG register 6 | G1TM6/G1PO6 | ??₁₆<br>??₁₆ |
| 030E₁₆<br>030F₁₆ | TM/WG register 7 | G1TM7/G1PO7 | ??₁₆<br>??₁₆ |
| 0310₁₆ | WG control register 0 | G1POCR0 | 0X00X000₂ |
| 0311₁₆ | WG control register 1 | G1POCR1 | 0X00X000₂ |
| 0312₁₆ | WG control register 2 | G1POCR2 | 0X00X000₂ |
| 0313₁₆ | WG control register 3 | G1POCR3 | 0X00X000₂ |
| 0314₁₆ | WG control register 4 | G1POCR4 | 0X00X000₂ |
| 0315₁₆ | WG control register 5 | G1POCR5 | 0X00X000₂ |
| 0316₁₆ | WG control register 6 | G1POCR6 | 0X00X000₂ |
| 0317₁₆ | WG control register 7 | G1POCR7 | 0X00X000₂ |
| 0318₁₆ | TM control register 0 | G1TMCR0 | 00₁₆ |
| 0319₁₆ | TM control register 1 | G1TMCR1 | 00₁₆ |
| 031A₁₆ | TM control register 2 | G1TMCR2 | 00₁₆ |
| 031B₁₆ | TM control register 3 | G1TMCR3 | 00₁₆ |
| 031C₁₆ | TM control register 4 | G1TMCR4 | 00₁₆ |
| 031D₁₆ | TM control register 5 | G1TMCR5 | 00₁₆ |
| 031E₁₆ | TM control register 6 | G1TMCR6 | 00₁₆ |
| 031F₁₆ | TM control register 7 | G1TMCR7 | 00₁₆ |
| 0320₁₆<br>0321₁₆ | Base timer register | G1BT | ??₁₆<br>??₁₆ |
| 0322₁₆ | Base timer control register 0 | G1BCR0 | 00₁₆ |
| 0323₁₆ | Base timer control register 1 | G1BCR1 | 00₁₆ |
| 0324₁₆ | TM prescale register 6 | G1TPR6 | 00₁₆ |
| 0325₁₆ | TM prescale register 7 | G1TPR7 | 00₁₆ |
| 0326₁₆ | Function enable register | G1FE | 00₁₆ |
| 0327₁₆ | Function select register | G1FS | 00₁₆ |
| 0328₁₆<br>0329₁₆ | Base timer reset register | G1BTRR | ??₁₆<br>??₁₆ |
| 032A₁₆ | Divider register | G1DV | 00₁₆ |
| 032B₁₆ | | | |
| 032C₁₆ | | | |
| 032D₁₆ | | | |
| 032E₁₆ | | | |
| 032F₁₆ | | | |
| 0330₁₆ | Interrupt request register | G1IR | ??₁₆ |
| 0331₁₆ | Interrupt enable register 0 | G1IE0 | 00₁₆ |
| 0332₁₆ | Interrupt enable register 1 | G1IE1 | 00₁₆ |
| 0333₁₆ | | | |
| 0334₁₆ | | | |
| 0335₁₆ | | | |
| 0336₁₆ | | | |
| 0337₁₆ | | | |
| 0338₁₆ | | | |
| 0339₁₆ | | | |
| 033A₁₆ | | | |
| 033B₁₆ | | | |
| 033C₁₆ | | | |
| 033D₁₆ | | | |
| 033E₁₆ | NMI digital debounce register | NDDR | FF₁₆ |
| 033F₁₆ | P17 digital debounce register | P17DDR | FF₁₆ |

Note 1:The blank areas are reserved and cannot be accessed by users.

X : Noting is mapped to this bit
? : Value indeterminate at reset

**Figure 4.4. SFR Map (4 of 7)**

| Address | Register Name | Acronym | Value after Reset |
|---|---|---|---|
| 0340₁₆ | | | |
| 0341₁₆ | | | |
| 0342₁₆<br>0343₁₆ | Timer A1-1 register | TA11 | ??₁₆<br>??₁₆ |
| 0344₁₆<br>0345₁₆ | Timer A2-1 register | TA21 | ??₁₆<br>??₁₆ |
| 0346₁₆<br>0347₁₆ | Timer A4-1 register | TA41 | ??₁₆<br>??₁₆ |
| 0348₁₆ | Three-phase PWM control register 0 | INVC0 | 00₁₆ |
| 0349₁₆ | Three-phase PWM control register 1 | INVC1 | 00₁₆ |
| 034A₁₆ | Three-phase output buffer register 0 | IDB0 | 00₁₆ |
| 034B₁₆ | Three-phase output buffer register 1 | IDB1 | 00₁₆ |
| 034C₁₆ | Dead time timer | DTT | ??₁₆ |
| 034D₁₆ | Timer B2 interrupt occurrence frequency set counter | ICTB2 | X?₁₆ |
| 034E₁₆ | Position-data-retain function contol register | PDRF | XXXX0000₂ |
| 034F₁₆ | | | |
| 0350₁₆ | | | |
| 0351₁₆ | | | |
| 0352₁₆ | | | |
| 0353₁₆ | | | |
| 0354₁₆ | | | |
| 0355₁₆ | | | |
| 0356₁₆ | | | |
| 0357₁₆ | | | |
| 0358₁₆ | | | |
| 0359₁₆ | | | |
| 035A₁₆ | | | |
| 035B₁₆ | | | |
| 035C₁₆ | | | |
| 035D₁₆ | | | |
| 035E₁₆ | Interrupt request cause select register 2 | IFSR2A | 00XXXXXX₂(Note 2) |
| 035F₁₆ | Interrupt request cause select register | IFSR | 00₁₆ |
| 0360₁₆ | SI/O3 transmit/receive register | S3TRR | ??₁₆ |
| 0361₁₆ | | | |
| 0362₁₆ | SI/O3 control register | S3C | 01000000₂ |
| 0363₁₆ | SI/O3 bit rate generator | S3BRG | ??₁₆ |
| 0364₁₆ | SI/O4 transmit/receive register | S4TRR | ??₁₆ |
| 0365₁₆ | | | |
| 0366₁₆ | SI/O4 control register | S4C | 01000000₂ |
| 0367₁₆ | SI/O4 bit rate generator | S4BRG | ??₁₆ |
| 0368₁₆ | | | |
| 0369₁₆ | | | |
| 036A₁₆ | | | |
| 036B₁₆ | | | |
| 036C₁₆ | | | |
| 036D₁₆ | | | |
| 036E₁₆ | | | |
| 036F₁₆ | | | |
| 0370₁₆ | | | |
| 0371₁₆ | | | |
| 0372₁₆ | | | |
| 0373₁₆ | | | |
| 0374₁₆ | UART2 special mode register 4 | U2SMR4 | 00₁₆ |
| 0375₁₆ | UART2 special mode register 3 | U2SMR3 | 000X0X0X₂ |
| 0376₁₆ | UART2 special mode register 2 | U2SMR2 | X0000000₂ |
| 0377₁₆ | UART2 special mode register | U2SMR | X0000000₂ |
| 0378₁₆ | UART2 transmit/receive mode register | U2MR | 00₁₆ |
| 0379₁₆ | UART2 bit rate generator | U2BRG | ??₁₆ |
| 037A₁₆<br>037B₁₆ | UART2 transmit buffer register | U2TB | ????????₂<br>XXXXXXX?₂ |
| 037C₁₆ | UART2 transmit/receive control register 0 | U2C0 | 00001000₂ |
| 037D₁₆ | UART2 transmit/receive control register 1 | U2C1 | 00000010₂ |
| 037E₁₆<br>037F₁₆ | UART2 receive buffer register | U2RB | ????????₂<br>?????XX?₂ |

Note 1: The blank areas are reserved and cannot be accessed by users.
Note 2: Write "1" to bit 0 after reset.

X : Noting is mapped to this bit
? : Value indeterminate at reset

**Figure 4.5. SFR Map (5 of 7)**

| Address | Register Name | Acronym | Value after Reset |
|---|---|---|---|
| 038016 | Count start flag | TABSR | 0016 |
| 038116 | Clock prescaler reset flag | CPSRF | 0XXXXXXX2 |
| 038216 | One-shot start flag | ONSF | 0016 |
| 038316 | Trigger select register | TRGSR | 0016 |
| 038416 | Up-down flag | UDF | 0016 |
| 038516 | | | |
| 038616 | Timer A0 register | TA0 | ??16 |
| 038716 | | | ??16 |
| 038816 | Timer A1 register | TA1 | ??16 |
| 038916 | | | ??16 |
| 038A16 | Timer A2 register | TA2 | ??16 |
| 038B16 | | | ??16 |
| 038C16 | Timer A3 register | TA3 | ??16 |
| 038D16 | | | ??16 |
| 038E16 | Timer A4 register | TA4 | ??16 |
| 038F16 | | | ??16 |
| 039016 | Timer B0 register | TB0 | ??16 |
| 039116 | | | ??16 |
| 039216 | Timer B1 register | TB1 | ??16 |
| 039316 | | | ??16 |
| 039416 | Timer B2 register | TB2 | ??16 |
| 039516 | | | ??16 |
| 039616 | Timer A0 mode register | TA0MR | 0016 |
| 039716 | Timer A1 mode register | TA1MR | 0016 |
| 039816 | Timer A2 mode register | TA2MR | 0016 |
| 039916 | Timer A3 mode register | TA3MR | 0016 |
| 039A16 | Timer A4 mode register | TA4MR | 0016 |
| 039B16 | Timer B0 mode register | TB0MR | 00??00002 |
| 039C16 | Timer B1 mode register | TB1MR | 00?X00002 |
| 039D16 | Timer B2 mode register | TB2MR | 00?X00002 |
| 039E16 | Timer B2 special mode register | TB2SC | X00000002 |
| 039F16 | | | |
| 03A016 | UART0 transmit/receive mode register | U0MR | 0016 |
| 03A116 | UART0 bit rate generator | U0BRG | ??16 |
| 03A216 | UART0 transmit buffer register | U0TB | ????????2 |
| 03A316 | | | XXXXXXX?2 |
| 03A416 | UART0 transmit/receive control register 0 | U0C0 | 000010002 |
| 03A516 | UART0 transmit/receive control register 1 | U0C1 | 000000102 |
| 03A616 | UART0 receive buffer register | U0RB | ????????2 |
| 03A716 | | | ?????XX?2 |
| 03A816 | UART1 transmit/receive mode register | U1MR | 0016 |
| 03A916 | UART1 bit rate generator | U1BRG | ??16 |
| 03AA16 | UART1 transmit buffer register | U1TB | ????????2 |
| 03AB16 | | | XXXXXXX?2 |
| 03AC16 | UART1 transmit/receive control register 0 | U1C0 | 000010002 |
| 03AD16 | UART1 transmit/receive control register 1 | U1C1 | 000000102 |
| 03AE16 | UART1 receive buffer register | U1RB | ????????2 |
| 03AF16 | | | ?????XX?2 |
| 03B016 | UART transmit/receive control register 2 | UCON | X00000002 |
| 03B116 | | | |
| 03B216 | | | |
| 03B316 | | | |
| 03B416 | | | |
| 03B516 | | | |
| 03B616 | | | |
| 03B716 | | | |
| 03B816 | DMA0 request cause select register | DM0SL | 0016 |
| 03B916 | | | |
| 03BA16 | DMA1 request cause select register | DM1SL | 0016 |
| 03BB16 | | | |
| 03BC16 | | | |
| 03BD16 | | | |
| 03BE16 | | | |
| 03BF16 | | | |

Note 1:The blank areas are reserved and cannot be accessed by users.

X : Noting is mapped to this bit
? : Value indeterminate at reset

**Figure 4.6. SFR Map (6 of 7)**

| Address | Register Name | Acronym | Value after Reset |
|---|---|---|---|
| 03C0₁₆ | A-D register 0 | AD0 | ????????₂ |
| 03C1₁₆ | | | XXXXXX??₂ |
| 03C2₁₆ | A-D register 1 | AD1 | ????????₂ |
| 03C3₁₆ | | | XXXXXX??₂ |
| 03C4₁₆ | A-D register 2 | AD2 | ????????₂ |
| 03C5₁₆ | | | XXXXXX??₂ |
| 03C6₁₆ | A-D register 3 | AD3 | ????????₂ |
| 03C7₁₆ | | | XXXXXX??₂ |
| 03C8₁₆ | A-D register 4 | AD4 | ????????₂ |
| 03C9₁₆ | | | XXXXXX??₂ |
| 03CA₁₆ | A-D register 5 | AD5 | ????????₂ |
| 03CB₁₆ | | | XXXXXX??₂ |
| 03CC₁₆ | A-D register 6 | AD6 | ????????₂ |
| 03CD₁₆ | | | XXXXXX??₂ |
| 03CE₁₆ | A-D register 7 | AD7 | ????????₂ |
| 03CF₁₆ | | | XXXXXX??₂ |
| 03D0₁₆ | | | |
| 03D1₁₆ | | | |
| 03D2₁₆ | A-D trigger control register | ADTRGCON | XXXX0000₂ |
| 03D3₁₆ | A-D convert status register 0 | ADSTAT0 | 00000X00₂ |
| 03D4₁₆ | A-D control register 2 | ADCON2 | 00₁₆ |
| 03D5₁₆ | | | |
| 03D6₁₆ | A-D control register 0 | ADCON0 | 00000???₂ |
| 03D7₁₆ | A-D control register 1 | ADCON1 | 00₁₆ |
| 03D8₁₆ | | | |
| 03D9₁₆ | | | |
| 03DA₁₆ | | | |
| 03DB₁₆ | | | |
| 03DC₁₆ | | | |
| 03DD₁₆ | | | |
| 03DE₁₆ | | | |
| 03DF₁₆ | | | |
| 03E0₁₆ | Port P0 register | P0 | ??₁₆ |
| 03E1₁₆ | Port P1 register | P1 | ??₁₆ |
| 03E2₁₆ | Port P0 direction register | PD0 | 00₁₆ |
| 03E3₁₆ | Port P1 direction register | PD1 | 00₁₆ |
| 03E4₁₆ | Port P2 register | P2 | ??₁₆ |
| 03E5₁₆ | Port P3 register | P3 | ??₁₆ |
| 03E6₁₆ | Port P2 direction register | PD2 | 00₁₆ |
| 03E7₁₆ | Port P3 direction register | PD3 | 00₁₆ |
| 03E8₁₆ | | | |
| 03E9₁₆ | | | |
| 03EA₁₆ | | | |
| 03EB₁₆ | | | |
| 03EC₁₆ | Port P6 register | P6 | ??₁₆ |
| 03ED₁₆ | Port P7 register | P7 | ??₁₆ |
| 03EE₁₆ | Port P6 direction register | PD6 | 00₁₆ |
| 03EF₁₆ | Port P7 direction register | PD7 | 00₁₆ |
| 03F0₁₆ | Port P8 register | P8 | ??₁₆ |
| 03F1₁₆ | Port P9 register | P9 | ???X????₂ |
| 03F2₁₆ | Port P8 direction register | PD8 | 00₁₆ |
| 03F3₁₆ | Port P9 direction register | PD9 | 000X0000₂ |
| 03F4₁₆ | Port P10 register | P10 | ??₁₆ |
| 03F5₁₆ | | | |
| 03F6₁₆ | Port P10 direction register | PD10 | 00₁₆ |
| 03F7₁₆ | | | |
| 03F8₁₆ | | | |
| 03F9₁₆ | | | |
| 03FA₁₆ | | | |
| 03FB₁₆ | | | |
| 03FC₁₆ | Pull-up control register 0 | PUR0 | 00₁₆ |
| 03FD₁₆ | Pull-up control register 1 | PUR1 | 00₁₆ |
| 03FE₁₆ | Pull-up control register 2 | PUR2 | 00₁₆ |
| 03FF₁₆ | Port control register | PCR | 00₁₆ |

Note 1:The blank areas are reserved and cannot be accessed by users.

X : Noting is mapped to this bit
? : Value indeterminate at reset

**Figure 4.7. SFR Map (7 of 7)**

# 5. Reset

There are four types of resets: a hardware reset, a software reset, an watchdog timer reset, and an oscillation stop detection reset.

## 5.1 Hardware Reset

There are two types of hardware resets: a hardware reset 1 and a hardware reset 2.

### 5.1.1 Hardware Reset 1

A reset is applied using the $\overline{\text{RESET}}$ pin. When an “L” signal is applied to the $\overline{\text{RESET}}$ pin while the power supply voltage is within the recommended operating condition, the pins are initialized (see Table 5.1.1.1 Pin Status When $\overline{\text{RESET}}$ Pin Level is “L”). The internal ring oscillator is initialized and used as sysem clock.

When the input level at the $\overline{\text{RESET}}$ pin is released from “L” to “H”, the CPU and SFR are initialized, and the program is executed starting from the address indicated by the reset vector. The internal RAM is not initialized. If the $\overline{\text{RESET}}$ pin is pulled “L” while writing to the internal RAM, the internal RAM becomes indeterminate.

Figure 5.1.1.1 shows the example reset circuit. Figure 5.1.1.2 shows the reset sequence. Table 5.1.1.1 shows the status of the other pins while the $\overline{\text{RESET}}$ pin is “L”. Figure 5.1.1.3 shows the CPU register status after reset. Refer to “SFR Map” for SFR status after reset.

1. When the power supply is stable
(1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin.
(2) Wait td(ROC) or more.
(3) Apply an “H” signal to the $\overline{\text{RESET}}$ pin.

2. Power on
(1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin.
(2) Let the power supply voltage increase until it meets the recommended operating condition.
(3) Wait td(P-R) or more until the internal power supply stabilizes.
(4) Wait td(ROC) or more.
(5) Apply an “H” signal to the $\overline{\text{RESET}}$ pin.

### 5.1.2 Hardware Reset 2

This reset is generated by the microcomputer’s internal voltage detection circuit. The voltage detection circuit monitors the voltage supplied to the $V_{CC}$ pin.

If the VC26 bit in the VCR2 register is set to “1” (reset level detection circuit enabled), the microcomputer is reset when the voltage at the $V_{CC}$ input pin drops below Vdet3.

Similarly, if the VC25 bit in the VCR2 register is set to “1” (RAM retention limit detection circuit enabled), the microcomputer is reset when the voltage at the VCC input pin drops below Vdet2.

Conversely, when the input voltage at the $V_{CC}$ pin rises to Vdet3 or more, the pins and the CPU and SFR are initialized, and the program is executed starting from the address indicated by the reset vector. It takes about td(S-R) before the program starts running after Vdet3 is detected. The initialized pins and registers and the status thereof are the same as in hardware reset 1.

Set the CM10 bit in the CM1 register to “1” (stop mode) after setting the VC25 bit to “1” (RAM retention limit detection circuit enabled), and the microcomputer will be reset when the voltage at the VCC input pin drops below Vdet2 and comes out of reset when the voltage at the $V_{CC}$ input pin rises above Vdet3. During stop mode, the value set in the VC26 bit has no effect. Therefore, no reset is generated even when the input voltage at the $V_{CC}$ pin drops to Vdet3 or less.

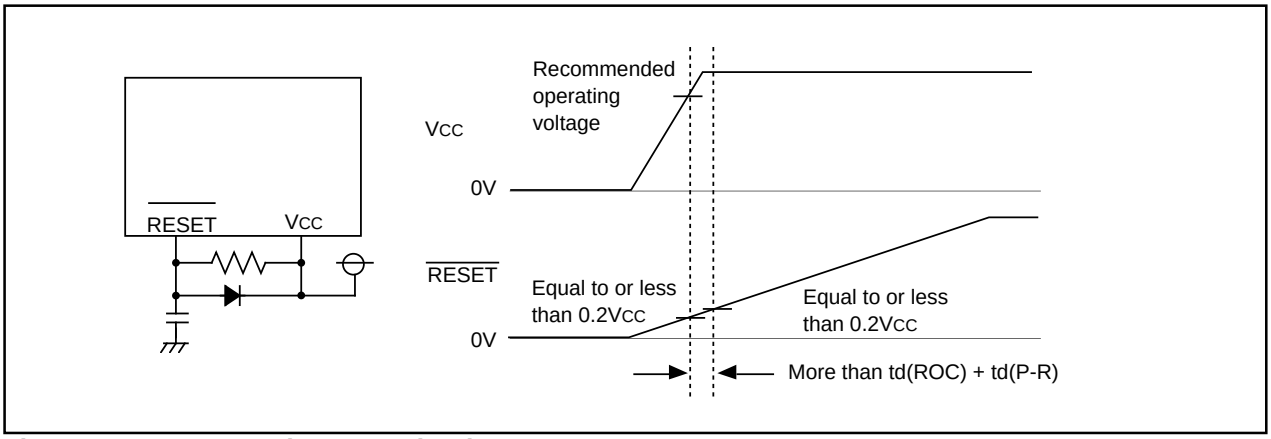



**Figure 5.1.1.1. Example Reset Circuit**

## 5.2 Software Reset

When the PM03 bit in the PM0 register is set to “1” (microcomputer reset), the microcomputer has its pins, CPU, and SFR initialized. Then the program is executed starting from the address indicated by the reset vector. The device will reset using internal ring oscillator as the system clock.

At software reset, some SFR’s are not initialized. Refer to “SFR”.

## 5.3 Watchdog Timer Reset

When the PM12 bit in the PM1 register is “1” (reset when watchdog timer underflows), the microcomputer initializes its pins, CPU and SFR if the watchdog timer underflows. The device will reset using internal ring oscillator as the system clock. Then the program is executed starting from the address indicated by the reset vector.

At watchdog timer reset, some SFR’s are not initialized. Refer to “SFR”.

## 5.4 Oscillation Stop Detection Reset

When the CM27 bit in the CM2 register is “0” (reset at oscillation stop detection), the microcomputer initializes its pins, CPU and SFR, coming to a halt if it detects main clock oscillation circuit stop. Refer to the section “oscillation stop, re-oscillation detection function”.

At oscillation stop detection reset, some SFR’s are not initialized. Refer to the section “SFR”.

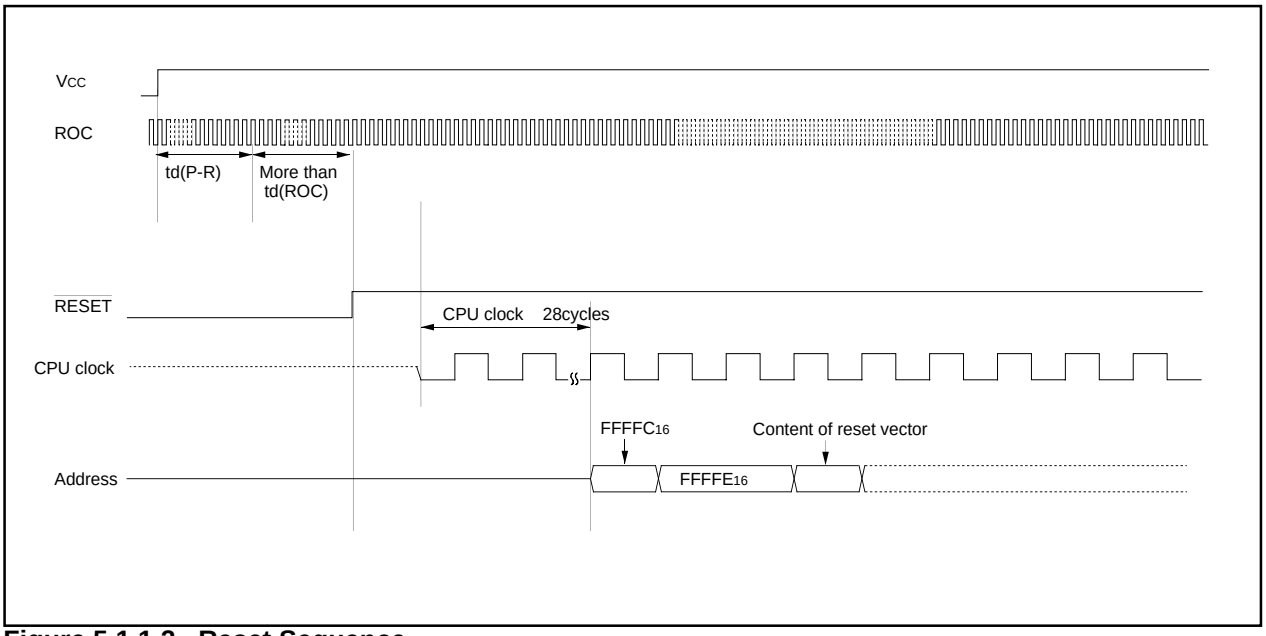



**Figure 5.1.1.2. Reset Sequence**

**Table 5.1.1.1. Pin Status When $\overline{\text{RESET}}$ Pin Level is "L"**

| Pin name | Status |
|---|---|
| P0 to P3,<br>P6 to P10 | Input port (high impedance) |

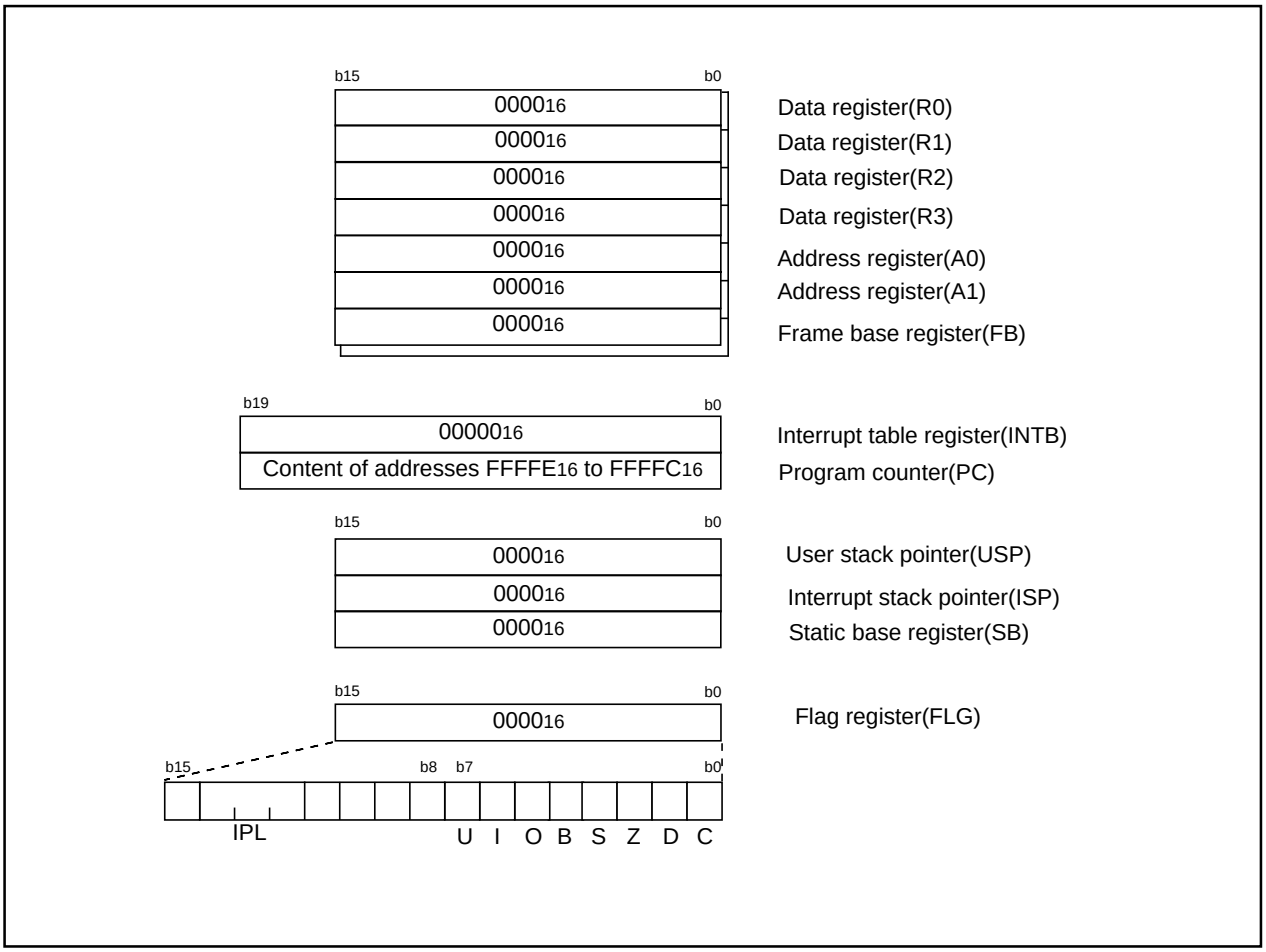



**Figure 5.1.1.3. CPU Register Status After Reset**

## 5.5 Voltage Detection Circuit

The voltage detection circuit has circuits to monitor the input voltage at the Vcc pin, each checking the input voltage with respect to Vdet2, Vdet3, and Vdet4, respectively. Use the VC25 to VC27 bits in the VCR2 register to select whether or not to enable these circuits.

The VC25 bit in the VCR2 register needs to be set to “1” (the internal RAM retention limit detection circuit enable) when using hardware reset 2 in stop mode, or when using the WDC5 bit. WDC5 bit =“1” shows state of the internal RAM retention.Use the reset level detection circuit for hardware reset 2.

The voltage down detection circuit can be set to detect whether the input voltage is equal to or greater than Vdet4 or less than Vdet4 by monitoring the VC13 bit in the VCR1 register. Furthermore, a voltage down detection interrupt can be generated.

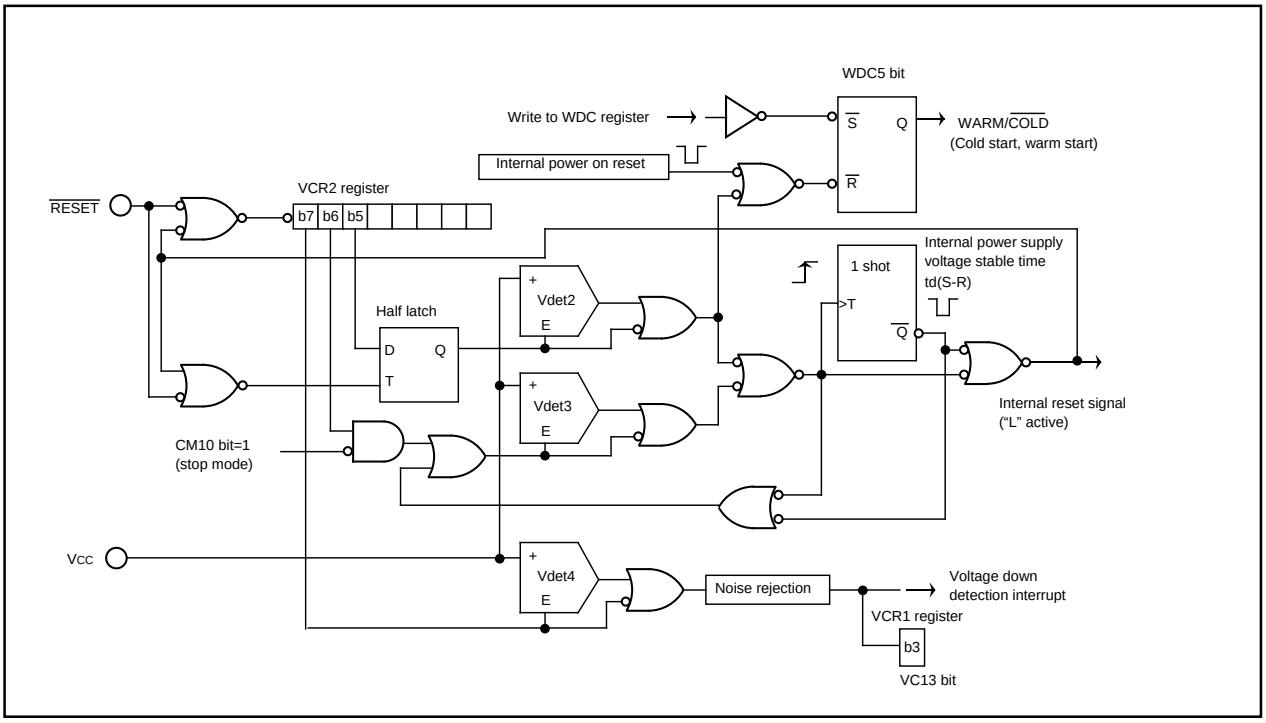



**Figure 5.5.1. Voltage Detection Circuit Block**

Watchdog timer control register

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

Symbol: WDC
Address: 000F16
After reset: 00XXXXXX2(Note 2)

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| $\overline{\text{(b4-b0)}}$ | High-order bit of watchdog timer | | RO |
| WDC5 | Cold start / warm start discrimination flag (Note 1, 2) | 0 : Cold start<br>1 : Warm start | RW |
| $\overline{\text{(b6)}}$ | Reserved bit | Must set to “0” | RW |
| WDC7 | Prescaler select bit | 0 : Divided by 16<br>1 : Divided by 128 | RW |

Note 1: Writing to the WDC register causes the WDC5 bit to be set to “1” (warm start).
Note 2: The WDC5 bit is “0” (cold start) immediately after power-on. It can only be set to “1” in a program. It is set to “0” when the input voltage at the Vcc pin drops to Vdet2 or less while the VC25 bit in the VCR2 register is set to “1” (RAM retention limit detection circuit enable).

**Figure 5.5.2. WDC Register**

Voltage detection register 1

| b7 | b | b | b4 | b | b | b1 | b |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

Symbol VCR1 — Address 0019₁₆ — After reset (Note 2) 00001000₂

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| (b2-b0) | Reserved bit | Must set to “0” | RW |
| VC13 | Voltage down monitor flag (Note 1) | 0:Vcc < Vdet4<br>1:Vcc > Vdet4 | RO |
| (b7-b4) | Reserved bit | Must set to “0” | RW |

Note 1: The VC13 bit is useful when the VC27 bit of VCR2 register is set to “1” (voltage down detection circuit enable). The VC13 bit is always “1” (Vcc≥ Vdet4) when the VC27 bit in the VCR2 register is set to “0” (voltage down detection circuit disable).
Note 2: This register does not change at software reset, watchdog timer reset and oscillation stop detection reset.

Voltage detection register 2 (Note 1)

| b7 | b | b | b4 | b | b | b1 | b |
|---|---|---|---|---|---|---|---|
| | | | 0 | 0 | 0 | 0 | 0 |

Symbol VCR2 — Address 001A₁₆ — After reset (Note 6) 00₁₆

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| (b4-b0) | Reserved bit | Must set to “0” | RW |
| VC25 | RAM retention limit detection monitor bit (Notes 3, 4, 7) | 0: Disable RAM retention limit detection circuit<br>1: Enable RAM retention limit detection circuit | RW |
| VC26 | Reset level monitor bit (Notes 2, 3, 7) | 0: Disable reset level detection circuit<br>1: Enable reset level detection circuit | RW |
| VC27 | Voltage down monitor bit (Note 5) | 0: Disable voltage down detection circuit<br>1: Enable voltage down detection circuit | RW |

Note 1: Write to this register after setting the PRC3 bit in the PRCR register to “1” (write enable).
Note 2: When not in stop mode, to use hardware reset 2, set the VC26 bit to “1” (reset level detection circuit enable).
Note 3: To use hardware reset 2 in stop mode, set the VC25 bit to “1” (RAM retention limit detection circuit enable). VC26 bit is disabled in stop mode. (The microcomputer is not reset even if the voltage input to Vcc1 pin becomes lower than Vdet3.)
Note 4: To use the WDC5 bit in the WDC register, set the VC25 bit to “1” (RAM retention limit detection circuit enable).
Note 5: When the VC13 bit in the VCR1 register and D42 bit in the D4INT register are used or the D40 bit is set to “1” (voltage down detection interrupt enable), set the VC27 bit to “1” (voltage down detection circuit enable).
Note 6: This register does not change at software reset, watchdog timer reset and oscillation stop detection reset.
Note 7: The detection circuit does not start operation until td(E-A) elapses after the VC25 bit, VC26 bit, or VC27 bit are

Voltage down detection interrupt register (Note 1)

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

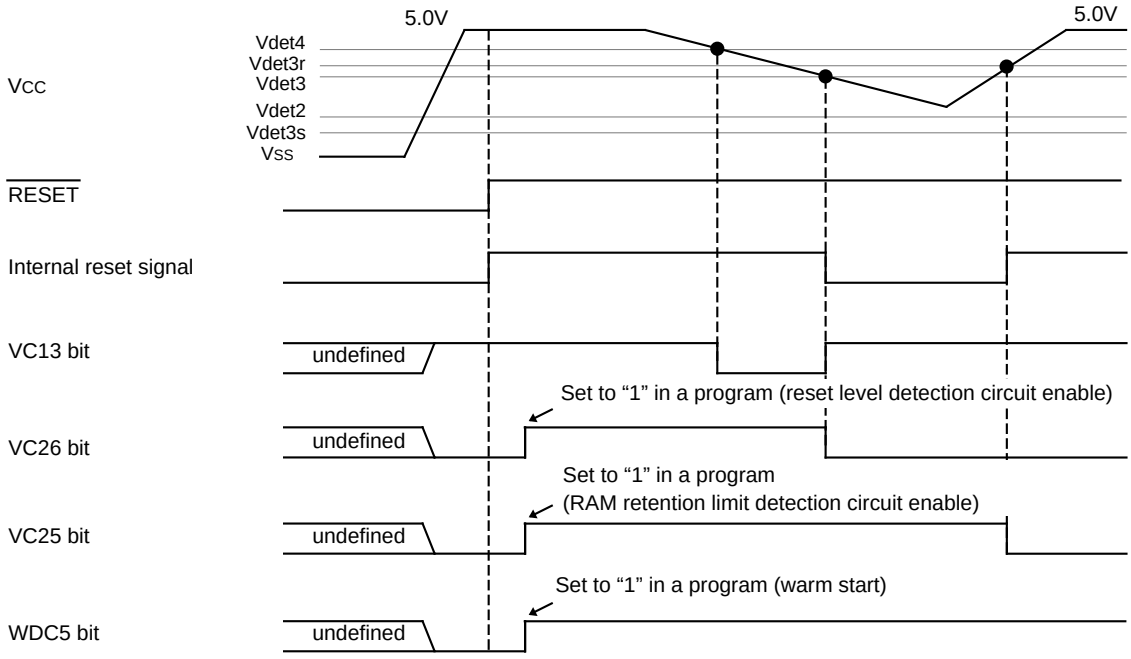
Symbol D4INT — Address 001F₁₆ — After reset 00₁₆

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| D40 | Voltage down detection interrupt enable bit (Note 5) | 0 : Disable<br>1 : Enable | RW |
| D41 | STOP mode deactivation control bit (Note 4) | 0: Disable (do not use the voltage down detection interrupt to get out of stop mode)<br>1: Enable (use the voltage down detection interrupt to get out of stop mode) | RW |
| D42 | Voltage change detection flag (Note 2) | 0: Not detected<br>1: Vdet4 passing detection | RW (Note 3) |
| D43 | WDT overflow detect flag | 0: Not detected<br>1: Detected | RW (Note 3) |
| DF0 | Sampling clock select bit | b5b4<br>00 : CPU clock divided by 8<br>01 : CPU clock divided by 16<br>10 : CPU clock divided by 32<br>11 : CPU clock divided by 64 | RW |
| DF1 | | | RW |
| — (b7-b6) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: Write to this register after setting the PRC3 bit in the PRCR register to “1” (write enable).
Note 2: Useful when the VC27 bit in the VCR2 register is set to “1” (voltage down detection circuit enabled). If the VC27 bit is set to “0” (voltage down detection circuit disable), the D42 bit is set to “0” (Not detect).
Note 3: This bit is set to “0” by writing a “0” in a program. (Writing a “1” has no effect.)
Note 4: If the voltage down detection interrupt needs to be used to get out of stop mode again after once used for that purpose, reset the D41 bit by writing a “0” and then a “1”.
Note 5: The D40 bit is effective when the VCR2 register VC27 bit = 1. To set the D40 bit to “1”, follow the procedure described below.
(1) Set the VC27 bit to “1”.
(2) Wait for td(E-A) until the detection circuit is actuated.
(3) Wait for the sampling time (refer to “Table 5.5.1.2 Sampling Clock Periods”).

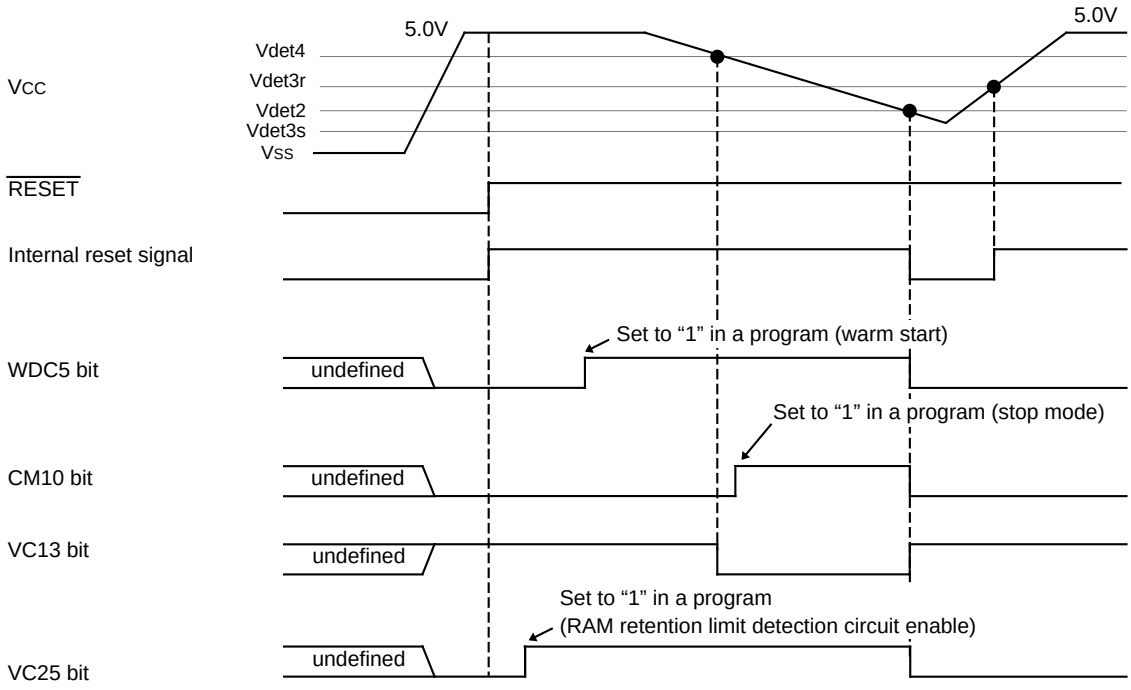
**Figure 5.5.3. VCR1 Register, VCR2 Register, and D4INT Register**

(1) When stop mode is not used




The above diagram applies to the following case:
• The CM10 bit in the CM1 register =0 (clock oscillating, not in stop mode)
• The VC27 bit in the VCR2 register is set to “1” after reset (voltage down detection circuit enabled)(to use VC13)

(1) When stop mode is used




The above diagram applies to the following case:
• The VC27 bit in the VCR2 register is set to “1” after reset (voltage down detection circuit enabled)(to use VC13)

**Figure 5.5.4. Typical Operation of Hardware Reset 2**

### 5.5.1 Voltage Detection Interrupt

A voltage down detection interrupt request is generated when the input voltage at the VCC pin rises to Vdet4 or more or drops below Vdet4 while the D40 bit in the D4INT register is set to “1” (voltage down detection interrupt enable). The voltage down detection interrupt shares the interrupt vector with the watchdog timer interrupt and oscillation stop, re-oscillation detection interrupt.

To use the voltage down detection interrupt to get out of stop mode, set the D41 bit in the D4INT register to “1” (enable).

The D42 bit in the D4INT register becomes “1” when passing through Vdet4 is detected after the voltage inputted to the VCC pin is up or down.

A voltage down detection interrupt is generated when the D42 bit changes state from “0” to “1”. The D42 bit needs to be cleared to “0” by software. However, when D41 bit is “1” and the microcontroller is in stop mode, if the voltage down detection interrupt occurs (due to voltage applied at VCC increases, passing through Vdet4), the microcontroller awakes from stop mode with no regard to the status of the D42 bit.

Table 5.5.1.1 shows the voltage down detection interrupt request generation conditions.

It is possible to set the sampling clock detecting that the voltage applied to the VCC pin has passed through Vdet4 with the DF1 to DF0 bits of D4INT register. Table 5.5.1.2 shows sampling clock periods.

**Table 5.5.1.1. Voltage Detection Interrupt Request Generation Conditions**

| operation mode | VC27 bit | D40 bit | D41 bit | D42 bit | CM02 bit | VC13 bit |
|---|---|---|---|---|---|---|
| Normal operation mode(Note 1) | 1 | 1 | — | 0 to 1 | — | 0 to 1 (Note 3) |
| | | | | | | 1 to 0 (Note 3) |
| Wait mode (Note 2) | | | — | 0 to 1 | 0 | 0 to 1 (Note 3) |
| | | | | | | 1 to 0 (Note 3) |
| | | | | — | 1 | 0 to 1 |
| Stop mode (Note 2) | | | 1 | — | 0 | 0 to 1 |

– : “0”or “1”

Note 1: The status except the wait mode and stop mode is handled as the normal mode.(Refer to “Clock generating circuit”)
Note 2: Refer to “Limitations on stop mode”, “Limitations on wait mode”.
Note 3: An interrupt request for voltage reduction is generated a sampling time after the value of the VC13 bit has changed. Refer to the “Figure 5.5.1.1.2.2.Voltage Down Detection Interrupt Generation Circuit Operation Example” for details.

**Table 5.5.1.2. Sampling Clock Periods**

| CPU clock (MHz) | Sampling clock (µs) | | | |
|---|---|---|---|---|
| | DF1 to DF0=00 (CPU clock divided by 8) | DF1 to DF0=01 (CPU clock divided by 16) | DF1 to DF0=10 (CPU clock divided by 32) | DF1 to DF0=11 (CPU clock divided by 64) |
| 16 | 3.0 | 6.0 | 12.0 | 24.0 |

#### 5.5.1.1 Precautions

#### 5.5.1.1.1. Limitations on Stop Mode

Before setting the CM10 bit in the CM1 register to “1”(stop mode), be sure to clear the CM02 bit in the CM0 register to “0” (do not stop the peripheral function clock). If the CM10 bit in the CM1 register is set to “1” (stop mode) when the VC13 bit in the VCR1 register is “1” (VCC ≥ Vdet4) while the VC27 bit in the VCR2 register is “1” (voltage down detection circuit enable) and the D40 bit in the D4INT register is “1” (voltage down detection interrupt enable) and D41 bit in the D4INT register is “1” (voltage down detection interrupt is used to get out of stop mode), a voltage down detection interrupt is immediately generated, causing the microcomputer to exit stop mode.

In systems where the microcomputer enters stop mode when the input voltage at the VCC pin drops below Vdet4 and exits stop mode when the input voltage rises to Vdet4 or more, make sure the CM10 bit is set to “1” when VC13 bit is “0” (VCC < Vdet4).

### 5.5.1.1.2. Limitations on WAIT Instruction

If the WAIT instruction is executed when the VC13 bit in the VCR1 register is “1” ($V_{CC} \geq V_{det4}$) while the VC27 bit in the VCR2 register is “1” (voltage down detection circuit enable) and the D40 bit in the D4INT register is “1” (voltage down detection interrupt enable), a voltage down detection interrupt is immediately generated, causing the microcomputer to exit wait mode.

In systems where the microcomputer enters wait mode when the input voltage at the $V_{CC}$ pin drops below Vdet4 and exits wait mode when the input voltage rises to Vdet4 or more, make sure the D42 bit is cleared to "0" when VC13 bit is “0” ($V_{CC} < V_{det4}$) before executing the WAIT instruction.

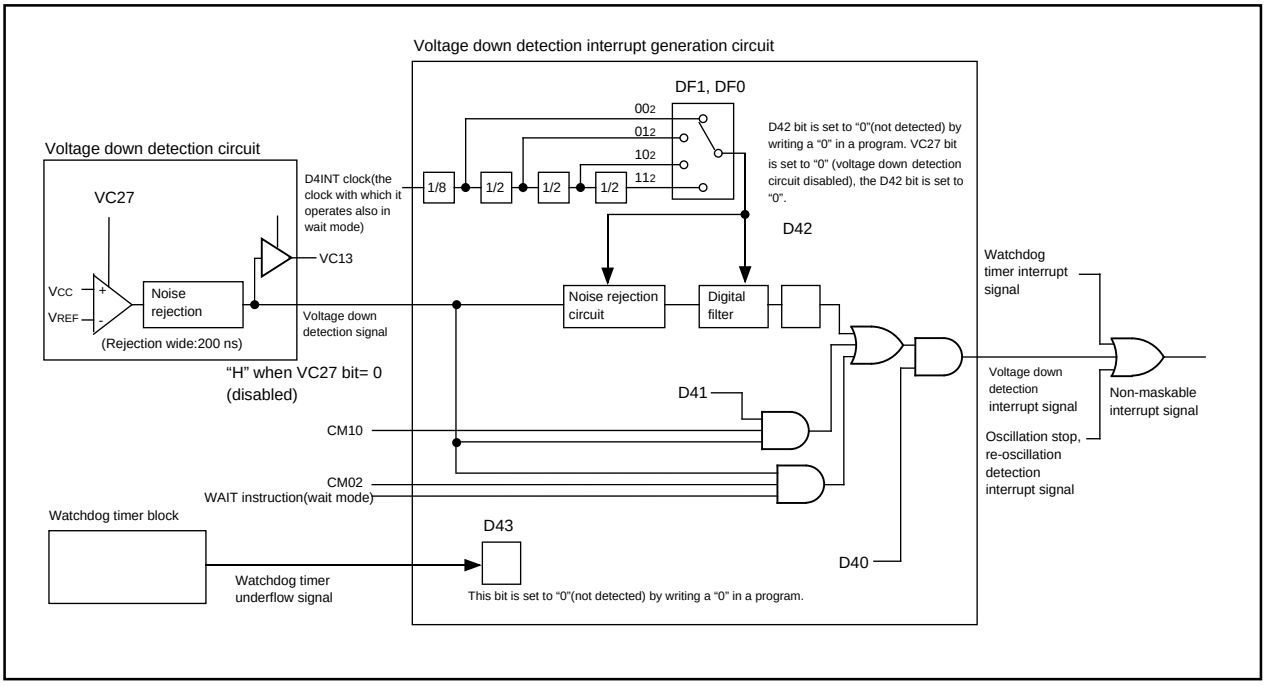
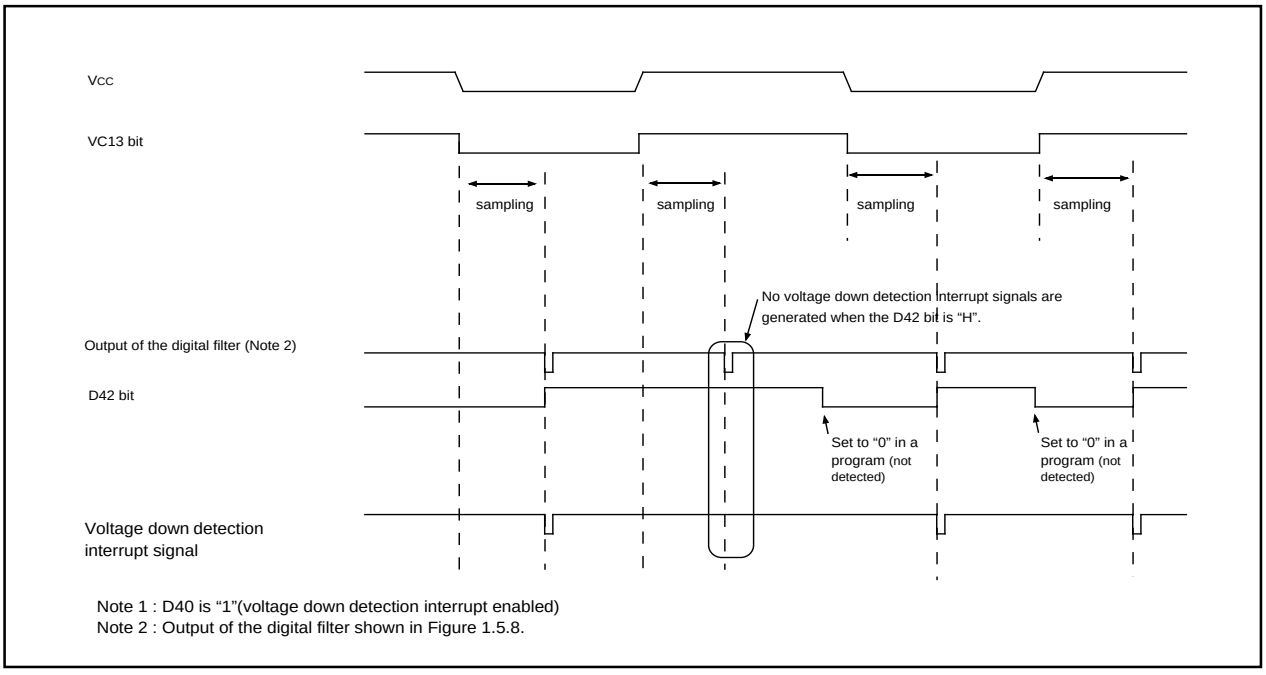



**Figure 5.5.1.1.2.1. Voltage Down Detection Interrupt Generation Block**




Note 1 : D40 is “1”(voltage down detection interrupt enabled)
Note 2 : Output of the digital filter shown in Figure 1.5.8.

**Figure 5.5.1.1.2.2. Voltage Down Detection Interrupt Generation Circuit Operation Example**

# 6. Processor Mode

This device functions in single-chip mode only. Figures 6.1 and 6.2 detail the associated registers.

Processor mode register 0 (Note)

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

Symbol: PM0 | Address: 0004$_{16}$ | After reset: 0000000 0$_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| $\overline{\text{(b2-b0)}}$ | Reserved bit | Should be set to "0". | RW |
| PM03 | Software reset bit | Setting this bit to "1" resets the microcomputer. When read, its content is "0". | RW |
| $\overline{\text{(b7-b4)}}$ | Reserved bit | Should be set to "0". | RW |

Note: Write to this register after setting the PRC1 bit in the PRCR register to "1" (write enable).

**Figure 6.1. PM0 Register**

Processor mode register 1 (Note 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | 0 | | | 1 | | 0 | |

Symbol: PM1 | Address: 0005$_{16}$ | After reset: 0000100 0$_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PM10 | Flash data block access bit (Note 2) | 0: Disabled<br>1: Enabled (Note 3) | RW |
| $\overline{\text{(b1)}}$ | Reserved bit | Should be set to "0". | RW |
| PM12 | Watchdog timer function select bit | 0 : Watchdog timer interrupt<br>1 : Watchdog timer reset (Note 4) | RW |
| $\overline{\text{(b3)}}$ | Reserved bit | Should be set to "1". | RW |
| $\overline{\text{(b6-b4)}}$ | Reserved bit | Should be set to "0". | RW |
| PM17 | Wait bit (Note 5) | 0 : No wait state<br>1 : With wait state (1 wait) | RW |

Note 1: Write to this register after setting the PRC1 bit in the PRCR register to "1" (write enable).
Note 2: To access the two 2K-byte data areas in data block A and data block B, this bit must be set to "1".
Note 3: When CPU rewrite mode (FMR01="1"), this bit is automatically set to "1" during that time.
Note 4: PM12 bit is set to 1 by writing a 1 in a program. (Writing a 0 has no effect.)
Note 5: When PM17 bit is set to "1" (with wait state), one wait state is inserted when accessing the internal RAM or the internal ROM.

**Figure 6.2. PM1 Register**

# 7. Clock Generation Circuit

The clock generation circuit contains four oscillator circuits as follows:

(1) Main clock oscillation circuit
(2) Sub clock oscillation circuit
(3) Variable ring oscillator (available at reset, oscillation stop detect function)
(4) PLL frequency synthesizer

Table 7.1 lists the clock generation circuit specifications. Figure 7.1 shows the clock generation circuit. Figures 7.2 to 7.6 show the clock-related registers.

**Table 7.1. Clock Generation Circuit Specifications**

| Item | Main clock oscillation circuit | Sub clock oscillation circuit | Variable ring oscillator | PLL frequency synthesizer |
|---|---|---|---|---|
| Use of clock | CPU clock source<br>Peripheral function clock source | CPU clock source<br>Timer A, B's clock source | CPU clock source<br>Peripheral function clock source<br>CPU and peripheral function clock sources when the main clock stops oscillating | CPU clock source<br>Peripheral function clock source |
| Clock frequency | 0 to 20 MHz | 32.768 kHz | Selectable source frequency:<br>$f_{1(ROC)}$, $f_{2(ROC)}$, $f_{3(ROC)}$<br>Selectable divider:<br>by 2, by 4, by 8 | 10 to 20 MHz |
| Usable oscillator | Ceramic oscillator<br>Crystal oscillator | Crystal oscillator | ——— | ——— |
| Pins to connect oscillator | XIN, XOUT | XCIN, XCOUT | ——— | ——— |
| Oscillation stop, restart function | Presence | Presence | Presence | Presence |
| Oscillator status after reset | Oscillating | Stopped | Oscillating<br>(CPU clock source) | Stopped |
| Other | Externally derived clock can be input | | ——— | ——— |

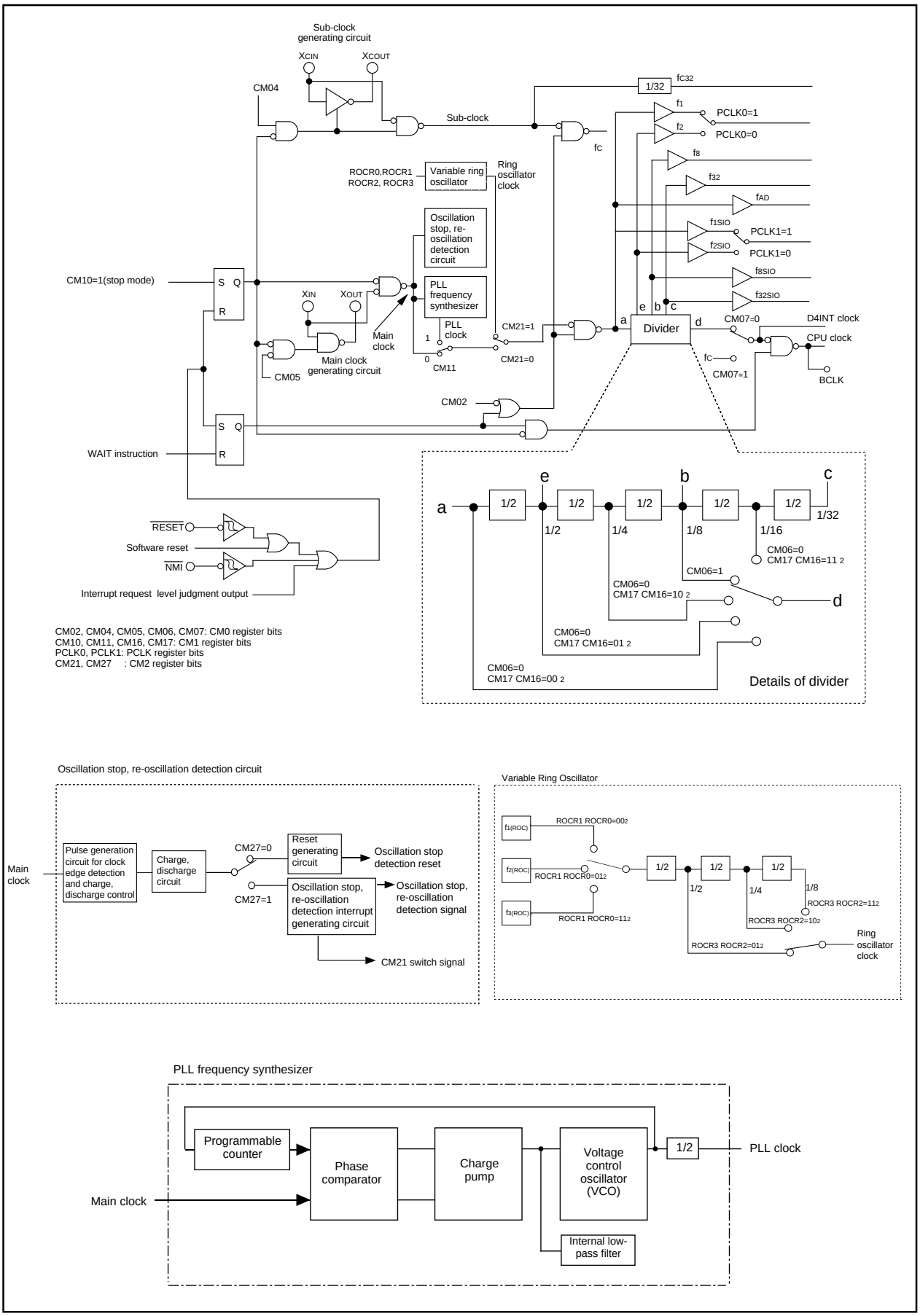
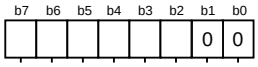



**Figure 7.1. Clock Generation Circuit**

System clock control register 0 (Note 1)



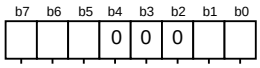

| Symbol | Address | After reset |
|---|---|---|
| CM0 | 0006$_{16}$ | 01001000$_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| —— (b1-b0) | Reserved bits | Must set to "0" | RW |
| CM02 | Wait Mode peripheral function clock stop bit (Note 10) | 0 : Do not stop peripheral function clock in wait mode<br>1 : Stop peripheral function clock in wait mode (Note 8) | RW |
| CM03 | XCIN-XCOUT drive capacity select bit (Note 2) | 0 : LOW<br>1 : HIGH | RW |
| CM04 | Port XC select bit (Note 2) | 0 : I/O port P8$_6$, P8$_7$<br>1 : XCIN-XCOUT generation function(Note 9) | RW |
| CM05 | Main clock stop bit (Notes 3, 10, 12, 13) | 0 : On (Note 4)<br>1 : Off (Note5) | RW |
| CM06 | Main clock division select bit 0 (Notes 7, 13, 14) | 0 : CM16 and CM17 valid<br>1 : Division by 8 mode | RW |
| CM07 | System clock select bit (Notes 6, 10, 11, 12) | 0 : Main clock, PLL clock, or ring oscillator clock<br>1 : Sub-clock | RW |

Note 1: Write to this register after setting the PRC0 bit of PRCR register to ì1î (write enable).
Note 2: The CM03 bit is set to ì1î (high) when the CM04 bit is set to ì0î (I/O port) or the microcomputer goes to a stop mode.
Note 3: This bit is provided to stop the main clock when the low power dissipation mode or ring oscillator low power dissipation mode is selected. This bit cannot be used for detection as to whether the main clock stopped or not. To stop the main clock, the following setting is required:
(1) Set the CM07 bit to ì1î (Sub-clock select) or the CM21 bit of CM2 register to ì1î (Ring oscillator select) with the sub-cl ock stably oscillating.
(2) Set the CM20 bit of CM2 register to ì0î (Oscillation stop, re-oscillation detection function disabled).
(3) Set the CM05 bit to ì1î (Stop).
Note 4: During external clock input, set to "0"(On).
Note 5: When CM05 bit is set to ì1, the XOUT pin goes ìHî. Furthermore, because the internal feedback resistor remains connected, the XIN pin is pulled ìHî to the same level as XOUT via the feedback resistor.
Note 6: After setting the CM04 bit to ì1î (XCIN-XCOUT oscillator function), wait until the sub-clock oscillates stably before switching the CM07 bit from ì0î to ì1î (sub-clock).
Note 7: When entering stop mode from high or middle speed mode, ring oscillator mode or ring oscillator low power mode, the CM06 bit is set to ì1î (divide-by-8 mode).
Note 8: The fC32 clock does not stop. During low speed or low power dissipation mode, do not set this bit to ì1î (peripheral clock turned off when in wait mode).
Note 9: To use a sub-clock, set this bit to ì1î. Also make sure ports P8$_6$ and P8$_7$ are directed for input, with no pull-ups.
Note 10: When the PM21 bit of PM2 register is set to ì1î (clock modification disable), writing to the CM02, CM05, and CM07 bits has no effect.
Note 11: If the PM21 bit needs to be set to ì1î, set the CM07 bit to ì0î(main clock) before setting it.
Note 12: To use the main clock as the clock source for the CPU clock, follow the procedure below.
(1) Set the CM05 bit to ì0î (oscillate).
(2) Wait until td(M-L) elapses or the main clock oscillation stabilizes, whichever is longer.
(3) Set the CM11, CM21 and CM07 bits all to ì0î.
Note 13: When the CM21 bit = 0 (ring oscillaor turned off) and the CM05 bit = 1 (main clock turned off), the CM06 bit is fixed to "1" (divide-by-8 mode) and the CM15 bit is fixed to "1" (drive capability High).
Note 14: To return from ring oscillator mode to high-speed or middle-speed mode set the CM06 and CM15 bits both to "1".

**Figure 7.2. CM0 Register**

System clock control register 1 (Note 1)



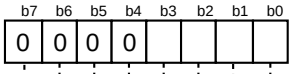

Symbol: CM1 — Address: 0007$_{16}$ — After reset: 00100000$_2$

| Bit symbol | Bit | Function | RW |
|---|---|---|---|
| CM10 | All clock stop control bit (Notes 4, 6) | 0 : Clock on<br>1 : All clocks off (stop mode) | RW |
| CM11 | System clock select bit 1 (Notes 6, 7) | 0 : Main clock<br>1 : PLL clock (Note 5) | RW |
| (b4-b2) | Reserved bit | Must set to "0" | RW |
| CM15 | XIN-XOUT drive capacity select bit (Note 2) | 0 : LOW<br>1 : HIGH | RW |
| CM16 | Main clock division select bits (Note 3) | b7 b6<br>0 0 : No division mode<br>0 1 : Division by 2 mode<br>1 0 : Division by 4 mode<br>1 1 : Division by 16 mode | RW |
| CM17 | | | RW |

Note 1: Write to this register after setting the PRC0 bit of PRCR register to "1" (write enable).
Note 2: When entering stop mode from high or middle speed mode, or when the CM05 bit is set to "1" (main clock turned off) in low speed mode, the CM15 bit is set to "1" (drive capability high).
Note 3: Effective when the CM06 bit is "0" (CM16 and CM17 bits enable).
Note 4: If the CM10 bit is "1" (stop mode), XOUT goes "H" and the internal feedback resistor is disconnected. The XCIN and XCOUT pins are placed in the high-impedance state. When the CM11 bit is set to "1" (PLL clock), or the CM20 bit of CM2 register is set to "1" (oscillation stop, re-oscillation detection function enabled), do not set the CM10 bit to "1".
Note 5: After setting the PLC07 bit in PLC0 register to "1" (PLL operation), wait until Tsu (PLL) elapses before setting the CM11 bit to "1" (PLL clock).
Note 6: When the PM21 bit of PM2 register is set to "1" (clock modification disable), writing to the CM10, CM011 bits has no effect. When the PM22 bit of PM2 register is set to "1" (watchdog timer count source is ring oscillator clock), writing to the CM10 bit has no effect.
Note 7: Effective when CM07 bit is "0" and CM21 bit is "0" .

**Figure 7.3. CM1 Register**

Ring Oscillator Control register (Note 1)



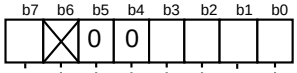

Symbol: ROCR — Address: 025C$_{16}$ — After reset: 00000101$_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| ROCR0 | Frequency select bits | b1 b0<br>0 0 : $f_1$ (ROC)<br>0 1 : $f_2$ (ROC)<br>1 0 : not supported<br>1 1 : $f_3$ (ROC) | RW |
| ROCR1 | | | RW |
| ROCR2 | Divider select bits | b3 b2<br>0 0 : not supported<br>0 1 : divide by 2<br>1 0 : divide by 4<br>1 1 : divide by 8 | RW |
| ROCR3 | | | RW |
| (b7-b4) | Reserved bit | Must set to "0" | RW |

Note 1 : Write to this register after setting the PRC0 bit of PRCR register to "1" (write enable).

**Figure 7.4. ROCR Register**

## Oscillation stop detection register (Note 1)




Symbol: CM2 | Address: 000C16 | After reset: 0X000010$_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| CM20 | Oscillation stop, re-oscillation detection bit (Notes 7, 9, 10, 11) | 0: Oscillation stop, re-oscillation detection function disabled<br>1: Oscillation stop, re-oscillation detection function enabled | RW |
| CM21 | System clock select bit 2 (Notes 2, 3, 6, 8, 11, 12 ) | 0: Main clock or PLL clock<br>1: Ring oscillator clock (Ring oscillator oscillating) | RW |
| CM22 | Oscillation stop, re-oscillation detection flag (Note 4) | 0: Main clock stop,or re-oscillation not detected<br>1: Main clock stop,or re-oscillation detected | RW |
| CM23 | XIN monitor flag (Note 5) | 0: Main clock oscillating<br>1: Main clock not oscillating | RO |
| — (b5-b4) | Reserved bit | Must set to “0” | RW |
| — (b6) | Nothing is assigned. When write, set to “0”. When read, its content is indeterminate. | | — |
| CM27 | Operation select bit (when an oscillation stop, re-oscillation is detected) (Note 11) | 0: Oscillation stop detection reset<br>1: Oscillation stop, re-oscillation detection interrupt | RW |

Note 1: Write to this register after setting the PRC0 bit of PRCR register to “1” (write enable).
Note 2: When the CM20 bit is “1” (oscillation stop, re-oscillation detection function enabled), the CM27 bit is “1” (oscillation stop, re-oscillation detection interrupt), and the CPU clock source is the main clock, the CM21 bit is automatically set to “1” (ring oscillator clock) if the main clock stop is detected.
Note 3: If the CM20 bit is “1” and the CM23 bit is “1” (main clock not oscillating), do not set the CM21 bit to “0”.
Note 4: This flag is set to “1” when the main clock is detected to have stopped or when the main clock is detected to have restarted oscillating. When this flag changes state from “0” to “1”, an oscillation stop, reoscillation restart detection interrupt is generated. Use this flag in an interrupt routine to discriminate the causes of interrupts between the oscillation stop, reoscillation detection interrupts and the watchdog timer interrupt. The flag is cleared to “0” by writing a “0” in a program. (Writing a “1” has no effect. Nor is it cleared to “0” by an oscillation stop or an oscillation restart detection interrupt request acknowledged.) If when the CM22 bit = 1 an oscillation stoppage or an oscillation restart is detected, no oscillation stop, reoscillation restart detection interrupts are generated.
Note 5: Read the CM23 bit in an oscillation stop, re-oscillation detection interrupt handling routine to determine the main clock status.
Note 6: Effective when the CM07 bit of CM0 register is “0”.
Note 7: When the PM21 bit of PM2 register is “1” (clock modification disabled), writing to the CM20 bit has no effect.
Note 8: When the CM20 bit is “1” (oscillation stop, re-oscillation detection function enabled), the CM27 bit is “1” (oscillation stop, re-oscillation detection interrupt), and the CM11 bit is “1” (the CPU clock source is PLL clock), the CM21 bit remains unchanged even when main clock stop is detected. If the CM22 bit is “0” under these conditions, oscillation stop, re-oscillation detection interrupt occur at main clock stop detection; it is, therefore, necessary to set the CM21 bit to “1” (ring oscillator clock) inside the interrupt routine.
Note 9: Set the CM20 bit to “0” (disable) before entering stop mode. After exiting stop mode, set the CM20 bit back to “1” (enable).
Note 10: Set the CM20 bit to “0” (disable) before setting the CM05 bit of CM0 register.
Note 11: The CM20, CM21 and CM27 bits do not change at oscillation stop detection reset.
Note 12: When the CM21 bit = 0 (ring oscillator turned off) and the CM05 bit = 1 (main clock turned off), the CM06 bit is fixed to “1” (divide-by-8 mode) and the CM15 bit is fixed to “1” (drive capability High).

**Figure 7.5. CM2 Register**

## Peripheral clock select register (Note)

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

Symbol: PCLKR — Address: $025E_{16}$ — When reset: $00000011_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PCLK0 | Timers A, B clock select bit (Clock source for the timers A, B, the timer S, and the dead timer) | 0 : $f_2$<br>1 : $f_1$ | RW |
| PCLK1 | SI/O clock select bit (Clock source for UART0 to UART2, SI/O3, SI/O4) | 0 : $f_{2SIO}$<br>1 : $f_{1SIO}$ | RW |
| —— (b7-b2) | Reserved bit | Must set to “0” | RW |

Note: Write to this register after setting the PRC0 bit of PRCR register to “1” (write enable).

## Processeor mode register 2 (Note 1)

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

Symbol: PM2 — Address: $001E_{16}$ — When reset: $XXX00000_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PM20 | Specifying wait when accessing SFR during PLL operation (Note 2) | 0 : 2 wait<br>1 : 1 wait | RW |
| PM21 | System clock protective bit (Note 3,4) | 0 : Clock is protected by PRCR register<br>1 : Clock modification disabled | RW |
| PM22 | WDT count source protective bit (Note 3,5) | 0 : CPU clock is used for the watchdog timer count source<br>1 : Ring oscillator clock is used for the watchdog timer count source | RW |
| —— (b3) | Reserved bit | Must set to “0” | RW |
| PM24 | $P8_5$/$\overline{NMI}$ configuration bit (Note 6,7) | 0 : $P8_5$ function ($\overline{NMI}$ disable)<br>1 : $\overline{NMI}$ function | RW |
| —— (b7-b5) | Nothing is assigned. When write, set to “0”. When read, its content is indeterminate. | | — |

Note 1: Write to this register after setting the PRC1 bit of PRCR register to “1” (write enable).
Note 2: This bit can only be rewritten while the PLC07 bit is “0” (PLL turned off). Also, to select a 16MHz or higher PLL clock or sytem clock, set this bit to “0” (2 wait).
Note 3: Once this bit is set to “1”, it cannot be cleared to “0” in a program.
Note 4: Setting the PM21 bit to “1” results in the following conditions:
- The BCLK is not halted by executing the WAIT instruction.
- Writting to the following bits has no effect.
  - CM02 bit of CM0 register
  - CM05 bit of CM0 register (main clock is not halted)
  - CM07 bit of CM0 register (CPU clock source does not change)
  - CM10 bit of CM1 register (stop mode is not entered)
  - CM11 bit of CM1 register (CPU clock source does not change)
  - CM20 bit of CM2 register (oscillation stop, re-oscillation detection function settings do not change)
  - All bit of PLC0 register (PLL frequency synthesizer setting do not change)

Note 5: Setting the PM22 bit to “1” results in the following conditions:
- The ring oscillator starts oscillating, and the ring oscillator clock becomes the watchdog timer count source.
- The CM10 bit of CM1 register is disabled against write. (Writing a “1” has no effect, nor is stop mode entered.)
- The watchdog timer does not stop when in wait mode.

Note 6: For $\overline{NMI}$ function, the PM24 bit must be set to “1”($\overline{NMI}$ function) in first instruction after rest. Once this bit is set to “1”, it cannot be cleared to “0” in a program. When the PM24 bit is set to “1”, the $P8_5$ direction register must be “0”.
Note 7: $\overline{SD}$ input is valid regardless of the PM24 setting.

**Figure 7.6. PCLKR Register and PM2 Register**

## PLL control register 0 (Note 1, Note 2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | 0 | 0 | 1 | X | | | |

| Symbol | Address | After reset |
|---|---|---|
| PLC0 | $001C_{16}$ | $0001\ X010_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PLC00 | PLL multiplying factor select bit (Note 3) | b2 b1 b0<br>0 0 0: Do not set<br>0 0 1: Multiply by 2<br>0 1 0: Multiply by 4<br>0 1 1: } Do not set<br>1 0 0: } Do not set<br>1 0 1: } Do not set<br>1 1 0: } Do not set<br>1 1 1: } Do not set | RW |
| PLC01 | | | RW |
| PLC02 | | | RW |
| —— (b3) | Nothing is assigned. When write, set to "0". When read, its content is indeterminate. | | — |
| —— (b4) | Reserved bit | Must set to "1" | RW |
| —— (b6-b5) | Reserved bit | Must set to "0" | RW |
| PLC07 | Operation enable bit (Note 4) | 0: PLL Off<br>1: PLL On | RW |

Note 1: Write to this register after setting the PRC0 bit of PRCR register to "1" (write enable).
Note 2: When the PM21 bit of PM2 register is "1" (clock modification disable), writing to this register has no effect.
Note 3: These three bits can only be modified when the PLC07 bit = "0" (PLL turned off). The value once written to this bit cannot be modified.
Note 4: Before setting this bit to "1" , set the CM07 bit to "0" (main clock), set the CM17 to CM16 bits to "$00_2$" (main clock undivided mode), and set the CM06 bit to "0" (CM16 and CM17 bits enable).

**Figure 7.7. PLC0 Register**

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

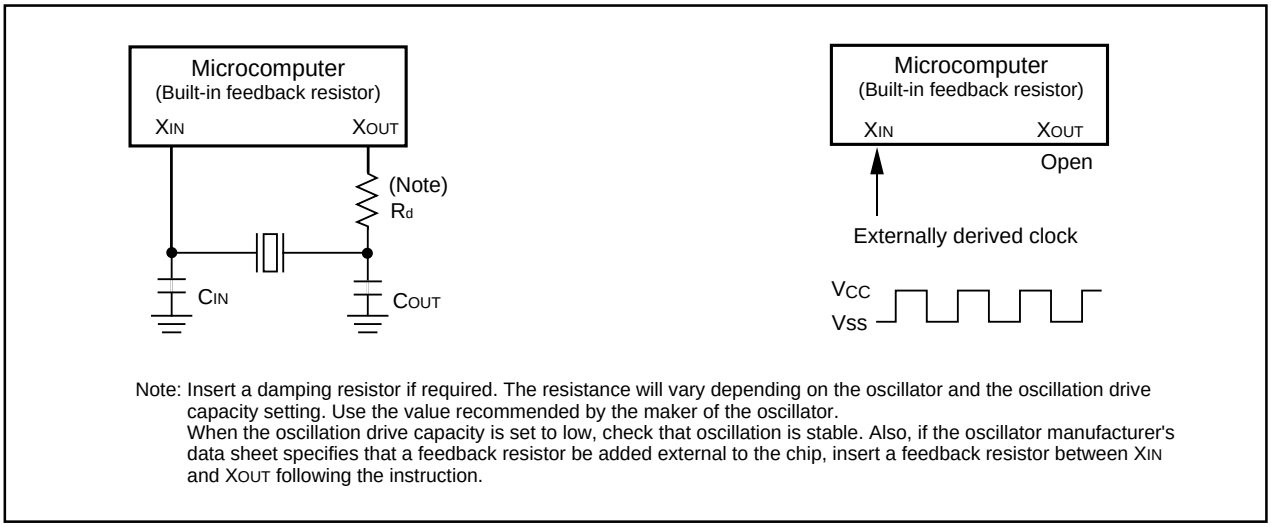
## 7.1 Main Clock

The main clock is generated by the main clock oscillation circuit. This clock is used as the clock source for the CPU and peripheral function clocks. The main clock oscillator circuit is configured by connecting a resonator between the XIN and XOUT pins. The main clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The main clock oscillator circuit may also be configured by feeding an externally generated clock to the XIN pin. Figure 7.1.1 shows the examples of main clock connection circuit.

The power consumption in the chip can be reduced by setting the CM05 bit of CM0 register to “1” (main clock oscillator circuit turned off) after switching the clock source for the CPU clock to a sub clock or ring oscillator clock. In this case, XOUT goes “H”. Furthermore, because the internal feedback resistor remains on, XIN is pulled “H” to XOUT via the feedback resistor.

During stop mode, all clocks including the main clock are turned off. Refer to “power control”.

If the main clock is not used, it is recommended to connect the XIN pin to VCC to reduce power consumption during reset.




Note: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
When the oscillation drive capacity is set to low, check that oscillation is stable. Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip, insert a feedback resistor between XIN and XOUT following the instruction.

**Figure 7.1.1. Examples of Main Clock Connection Circuit**

## 7.2 Sub Clock

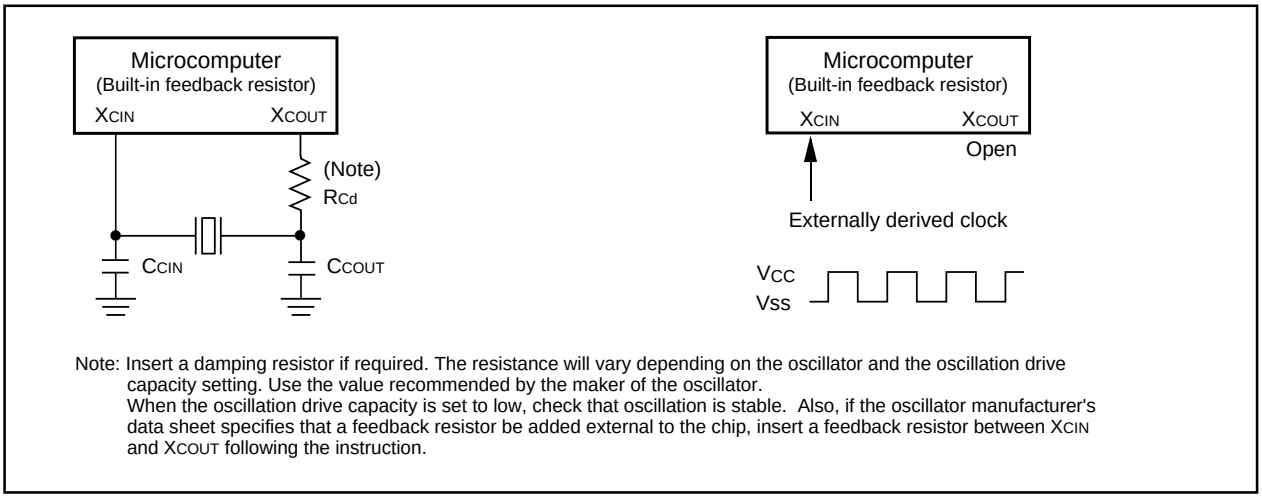
The sub clock is generated by the sub clock oscillation circuit. This clock is used as the clock source for the CPU clock, as well as the timer A and timer B count sources.

The sub clock oscillator circuit is configured by connecting a crystal resonator between the XCIN and XCOUT pins. The sub clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The sub clock oscillator circuit may also be configured by feeding an externally generated clock to the XCIN pin. Figure 7.2.1 shows the examples of sub clock connection circuit.

After reset, the sub clock is turned off. At this time, the feedback resistor is disconnected from the oscillator circuit.

To use the sub clock for the CPU clock, set the CM07 bit of CM0 register to “1 ” (sub clock) after the sub clock becomes oscillating stably.

During stop mode, all clocks including the sub clock are turned off. Refer to “power control”.




Note: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
When the oscillation drive capacity is set to low, check that oscillation is stable. Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip, insert a feedback resistor between XCIN and XCOUT following the instruction.

**Figure 7.2.1. Examples of Sub Clock Connection Circuit**

## 7.3 Ring Oscillator Clock

This clock is supplied by a variable ring oscillator. This clock is used as the clock source for the CPU and peripheral function clocks. In addition, if the PM22 bit of PM2 register is “1” (ring oscillator clock for the watchdog timer count source), this clock is used as the count source for the watchdog timer (Refer to “10. Watchdog Timer • Count source protective mode”).

After reset, the ring oscillator clock divided by 16 is used for the CPU clock. It can also be turned on by setting the CM21 bit of CM2 register to “1” (ring oscillator clock), and is used as the clock source for the CPU and peripheral function clocks. If the main clock stops oscillating when the CM20 bit of CM2 register is “1” (oscillation stop, re-oscillation detection function enabled) and the CM27 bit is “1” (oscillation stop, re-oscillation detection interrupt), the ring oscillator automatically starts operating, supplying the necessary clock for the microcomputer.

## 7.4 PLL Clock

The PLL clock is generated from the main clock by a PLL frequency synthesizer. This clock is used as the clock source for the CPU and peripheral function clocks. After reset, the PLL clock is turned off. The PLL frequency synthesizer is activated by setting the PLC07 bit to “1” (PLL operation). When the PLL clock is used as the clock source for the CPU clock, wait $t_{su}$(PLL) for the PLL clock to be stable, and then set the CM11 bit in the CM1 register to “1”.

Before entering wait mode or stop mode, be sure to set the CM11 bit to “0” (CPU clock source is the main clock). Furthermore, before entering stop mode, be sure to set the PLC07 bit in the PLC0 register to “0” (PLL stops). Figure 7.4.1 shows the procedure for using the PLL clock as the clock source for the CPU.

The PLL clock frequency is determined by the equation below.

PLL clock frequency=f($X_{IN}$) X (multiplying factor set by the PLC02 to PLC00 bits PLC0 register
(However, 10 MHz ≤ PLL clock frequency ≤ 20 MHz)

The PLC02 to PLC00 bits can be set only once after reset. Table 7.4.1 shows the example for setting PLL clock frequencies.

**Table 7.4.1. Example for Setting PLL Clock Frequencies**

| $X_{IN}$ (MHz) | PLC02 | PLC01 | PLC00 | Multiplying factor | PLL clock (MHz)(Note) |
|---|---|---|---|---|---|
| 10 | 0 | 0 | 1 | 2 | 20 |
| 5 | 0 | 1 | 0 | 4 | |

Note: 10MHz ≤ PLL clock frequency ≤ 20MHz.

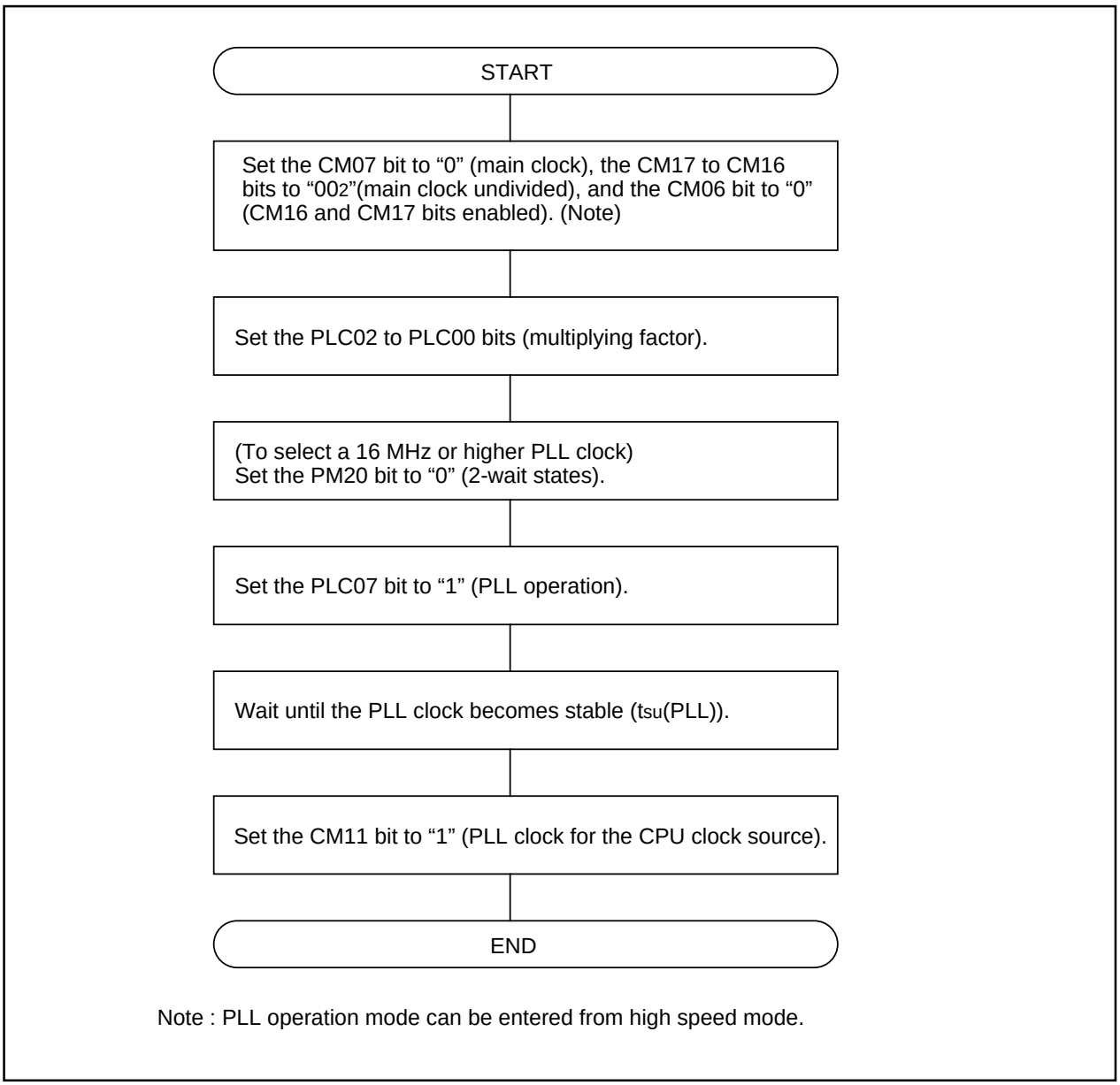



**Figure 7.4.1. Procedure to Use PLL Clock as CPU Clock Source**

## 7.5 CPU Clock and Peripheral Function Clock

The CPU clock is used to operate the CPU and peripheral function clocks are used to operate the peripheral functions.

### 7.5.1 CPU Clock

This is the operating clock for the CPU and watchdog timer.

The clock source for the CPU clock can be chosen to be the main clock, sub clock, ring oscillator clock or the PLL clock.

If the main clock or ring oscillator clock is selected as the clock source for the CPU clock, the selected clock source can be divided by 1 (undivided), 2, 4, 8 or 16 to produce the CPU clock. Use the CM06 bit in CM0 register and the CM17 to CM16 bits in CM1 register to select the divide-by-n value.

When the PLL clock is selected as the clock source for the CPU clock, the CM06 bit should be set to “0” and the CM17 to CM16 bits to “$00_2$” (undivided).

After reset, the ring oscillator clock divided by 16 provides the CPU clock.

Note that when entering stop mode from high or middle speed mode, ring oscillator mode or ring oscillator low power dissipation mode, or when the CM05 bit of CM0 register is set to “1” (main clock turned off) in low-speed mode, the CM06 bit of CM0 register is set to “1” (divide-by-8 mode).

### 7.5.2 Peripheral Function Clock($f_1$, $f_2$, $f_8$, $f_{32}$, $f_{1SIO}$, $f_{2SIO}$, $f_{8SIO}$, $f_{32SIO}$, $f_{AD}$, $f_{C32}$)

These are operating clocks for the peripheral functions.

Of these, fi (i = 1, 2, 8, 32) and $f_{iSIO}$ are derived from the main clock, PLL clock or ring oscillator clock by dividing them by i. The clock fi is used for timers A and B, and $f_{iSIO}$ is used for serial I/O.

The $f_{AD}$ clock is produced from the main clock, PLL clock or ring oscillator clock, and is used for the A-D converter.

When the WAIT instruction is executed after setting the CM02 bit of CM0 register to “1” (peripheral function clock turned off during wait mode), or when the microcomputer is in low power dissipation mode, the fi, $f_{iSIO}$ and $f_{AD}$ clocks are turned off.

The $f_{C32}$ clock is produced from the sub clock, and is used for timers A and B. This clock can only be used when the sub clock is on.

## 7.6 Power Control

There are three power control modes. For convenience' sake, all modes other than wait and stop modes are referred to as normal operation mode here.

### 7.6.1 Normal Operation Mode

Normal operation mode is further classified into seven modes.

In normal operation mode, because the CPU clock and the peripheral function clocks both are on, the CPU and the peripheral functions are operating. Power control is exercised by controlling the CPU clock frequency. The higher the CPU clock frequency, the greater the processing capability. The lower the CPU clock frequency, the smaller the power consumption in the chip. If the unnecessary oscillator circuits are turned off, the power consumption is further reduced.

Before the clock sources for the CPU clock can be switched over, the new clock source to which switched must be oscillating stably. If the new clock source is the main clock, sub clock or PLL clock, allow a sufficient wait time in a program until it becomes oscillating stably.

Note that operation modes cannot be changed directly from low speed or low power dissipation mode to ring oscillator or ring oscillator low power dissipation mode. Nor can operation modes be changed directly from ring oscillator or ring oscillator low power dissipation mode to low speed or low power dissipation mode. When the CPU clock source is changed from the ring oscillator to the main clock, change the operation mode to the medium speed mode (divided by 8 mode) after the clock was divided by 8 (the CM06 bit of CM0 register was set to "1") in the ring oscillator mode.

#### 7.6.1.1 High-speed Mode

The main clock divided by 1 provides the CPU clock. If the sub clock is on, $f_{C32}$ can be used as the count source for timers A and B.

#### 7.6.1.2 PLL Operation Mode

The main clock multiplied by 2 or 4 provides the PLL clock, and this PLL clock serves as the CPU clock. If the sub clock is on, $f_{C32}$ can be used as the count source for timers A and B. PLL operation mode can be entered from high speed mode. If PLL operation mode is to be changed to wait or stop mode, first go to high speed mode before changing.

#### 7.6.1.3 Medium-speed Mode

The main clock divided by 2, 4, 8 or 16 provides the CPU clock. If the sub clock is on, $f_{C32}$ can be used as the count source for timers A and B.

#### 7.6.1.4 Low-speed Mode

The sub clock provides the CPU clock. The main clock is used as the clock source for the peripheral function clock when the CM21 bit is set to "0" (ring oscillator turned off), and the ring oscillator clock is used when the CM21 bit is set to "1" (ring oscillator oscillating).

The $f_{C32}$ clock can be used as the count source for timers A and B.

#### 7.6.1.5 Low Power Dissipation Mode

In this mode, the main clock is turned off after being placed in low speed mode. The sub clock provides the CPU clock. The $f_{C32}$ clock can be used as the count source for timers A and B. Peripheral function clock can use only $f_{C32}$.

Simultaneously when this mode is selected, the CM06 bit of CM0 register becomes "1" (divided by 8 mode). In the low power dissipation mode, do not change the CM06 bit. Consequently, the medium speed (divided by 8) mode is to be selected when the main clock is operated next.

#### 7.6.1.6 Ring Oscillator Mode

The selected ring oscillator clock divided by 1 (undivided), 2, 4, 8 or 16 provides the CPU clock. The ring oscillator clock is also the clock source for the peripheral function clocks. If the sub clock is on, $f_{C32}$ can be used as the count source for timers A and B. The ring oscillator frequency can be selected using the ring oscillator control register (ROCR:025C16) bits 0 to 3. See Figure.7.4 for details. When the operation mode is returned to the high and medium speed modes, set the CM06 bit to “1” (divided by 8 mode).

#### 7.6.1.7 Ring Oscillator Low Power Dissipation Mode

The main clock is turned off after being placed in ring oscillator mode. The CPU clock can be selected as in the ring oscillator mode. The ring oscillator clock is the clock source for the peripheral function clocks. If the sub clock is on, $f_{C32}$ can be used as the count source for timers A and B.

**Table 7.6.1.1. Setting Clock Related Bit and Modes**

| Modes | | CM2 register | CM1 register | | CM0 register | | | |
|---|---|---|---|---|---|---|---|---|
| | | CM21 | CM11 | CM17, CM16 | CM07 | CM06 | CM05 | CM04 |
| PLL operation mode | | 0 | 1 | 00₂ | 0 | 0 | 0 | — |
| High-speed mode | | 0 | 0 | 00₂ | 0 | 0 | 0 | — |
| Medium-speed mode | divided by 2 | 0 | 0 | 01₂ | 0 | 0 | 0 | — |
| | divided by 4 | 0 | 0 | 10₂ | 0 | 0 | 0 | — |
| | divided by 8 | 0 | 0 | — | 0 | 1 | 0 | — |
| | divided by 16 | 0 | 0 | 11₂ | 0 | 0 | 0 | — |
| Low-speed mode | | — | — | — | 1 | — | 0 | 1 |
| Low power dissipation mode | | — | — | — | 1 | 1(Note 1) | 1(Note 1) | 1 |
| Ring oscillator mode (Note 3) | divided by 1 | 1 | — | 00₂ | 0 | 0 | 0 | — |
| | divided by 2 | 1 | — | 01₂ | 0 | 0 | 0 | — |
| | divided by 4 | 1 | — | 10₂ | 0 | 0 | 0 | — |
| | divided by 8 | 1 | — | 0— | 0 | 1 | 0 | — |
| | divided by 16 | 1 | — | 11₂ | 0 | 0 | 0 | — |
| Ring oscillator low power dissipation mode | | 1 | — | (Note 2) | 0 | (Note 2) | 1 | — |

Note 1: When the CM05 bit is set to ì1î (main clock turned off) in low-speed mode, the mode goes to low power dissipation mode and CM06 bit is set to ì1î (divided by 8 mode) simultaneously.
Note 2: The divide-by-n value can be selected the same way as in ring oscillator mode.
Note 3: Variable ring oscillator frequency can be any of those described in the section "Variable Ring Oscillator Mode".

### 7.6.2 Wait Mode

In wait mode, the CPU clock is turned off, so are the CPU (because operated by the CPU clock) and the watchdog timer. However, if the PM22 bit of PM2 register is “1” (ring oscillator clock for the watchdog timer count source), the watchdog timer remains active. Because the main clock, sub clock, ring oscillator clock and PLL clock all are on, the peripheral functions using these clocks keep operating.

#### 7.6.2.1 Peripheral Function Clock Stop Function

If the CM02 bit is “1” (peripheral function clocks turned off during wait mode), the $f_1$, $f_2$, $f_8$, $f_{32}$, $f_{1SIO}$, $f_{8SIO}$, $f_{32SIO}$ and $f_{AD}$ clocks are turned off when in wait mode, with the power consumption reduced that much. However, $f_{C32}$ remains on.

#### 7.6.2.2 Entering Wait Mode

The microcomputer is placed into wait mode by executing the WAIT instruction.

When the CM11 bit = “1” (CPU clock source is the PLL clock), be sure to clear the CM11 bit to “0” (CPU clock source is the main clock) before going to wait mode. The power consumption of the chip can be reduced by clearing the PLC07 bit to “0” (PLL stops).

#### 7.6.2.3 Pin Status During Wait Mode

The I/O port pins retain their status held just prior to wait mode.

### 7.6.2.4 Exiting Wait Mode

The microcomputer is moved out of wait mode by a hardware reset, $\overline{\text{NMI}}$ interrupt or peripheral function interrupt.

If the microcomputer is to be moved out of exit wait mode by a hardware reset or $\overline{\text{NMI}}$ interrupt, set the peripheral function interrupt priority ILVL2 to ILVL0 bits to "$000_2$" (interrupts disabled) before executing the WAIT instruction.

The peripheral function interrupts are affected by the CM02 bit. If CM02 bit is "0" (peripheral function clocks not turned off during wait mode), all peripheral function interrupts can be used to exit wait mode. If CM02 bit is "1" (peripheral function clocks turned off during wait mode), the peripheral functions using the peripheral function clocks stop operating, so that only the peripheral functions clocked by external signals can be used to exit wait mode.

Table 7.6.2.4.1 lists the interrupts to exit wait mode.

**Table 7.6.2.4.1. Interrupts to Exit Wait Mode**

| Interrupt | CM02=0 | CM02=1 |
|---|---|---|
| $\overline{\text{NMI}}$ interrupt | Can be used | Can be used |
| Serial I/O interrupt | Can be used when operating with internal or external clock | Can be used when operating with external clock |
| Multi-Master $I^2C$ interrupt | Can be used | — (Do not use) |
| key input interrupt | Can be used | Can be used |
| A-D conversion interrupt | Can be used in one-shot mode or single sweep mode | — (Do not use) |
| Timer A interrupt<br>Timer B interrupt | Can be used in all modes | Can be used in event counter mode or when the count source is fC32 |
| $\overline{\text{INT}}$ interrupt | Can be used | Can be used |

If the microcomputer is to be moved out of wait mode by a peripheral function interrupt, set up the following before executing the WAIT instruction.

1. In the ILVL2 to ILVL0 bits of interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit wait mode.
   Also, for all of the peripheral function interrupts not used to exit wait mode, set the ILVL2 to ILVL0 bits to "$000_2$" (interrupt disable).
2. Set the I flag to "1".
3. Enable the peripheral function whose interrupt is to be used to exit wait mode.
   In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt routine is executed.

The CPU clock turned on when exiting wait mode by a peripheral function interrupt is the same CPU clock that was on when the WAIT instruction was executed.

## 7.6.3 Stop Mode

In stop mode, all oscillator circuits are turned off, so are the CPU clock and the peripheral function clocks. Therefore, the CPU and the peripheral functions clocked by these clocks stop operating. The least amount of power is consumed in this mode. If the voltage applied to Vcc pin is VRAM or more, the internal RAM is retained. When applying 2.7 or less voltage to Vcc pin, make sure Vcc≥VRAM.

However, the peripheral functions clocked by external signals keep operating. The following interrupts can be used to exit stop mode.

- $\overline{\text{NMI}}$ interrupt
- Key interrupt
- $\overline{\text{INT}}$ interrupt
- Timer A, Timer B interrupt (when counting external pulses in event counter mode)
- Serial I/O interrupt (when external clock is selected)
- Voltage down detection interrupt (refer to “voltage down detection interrupt” for an operating condition)

### 7.6.3.1 Entering Stop Mode

The microcomputer is placed into stop mode by setting the CM10 bit of CM1 register to “1” (all clocks turned off). At the same time, the CM06 bit of CM0 register is set to “1” (divide-by-8 mode) and the CM15 bit of CM10 register is set to “1” (main clock oscillator circuit drive capability high).

Before entering stop mode, set the CM20 bit to “0” (oscillation stop, re-oscillation detection function disable).

Also, if the CM11 bit is “1” (PLL clock for the CPU clock source), set the CM11 bit to “0” (main clock for the CPU clock source) and the PLC07 bit to “0” (PLL turned off) before entering stop mode.

### 7.6.3.2 Pin Status during Stop Mode

The I/O pins retain their status held just prior to entering stop mode.

### 7.6.3.3 Exiting Stop Mode

The microcomputer is moved out of stop mode by a hardware reset, $\overline{\text{NMI}}$ interrupt or peripheral function interrupt.

If the microcomputer is to be moved out of stop mode by a hardware reset or $\overline{\text{NMI}}$ interrupt, set the peripheral function interrupt priority ILVL2 to ILVL0 bits to “$000_2$” (interrupts disable) before setting the CM10 bit to “1”.

If the microcomputer is to be moved out of stop mode by a peripheral function interrupt, set up the following before setting the CM10 bit to “1”.

1. In the ILVL2 to ILVL0 bits of interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit stop mode.
   Also, for all of the peripheral function interrupts not used to exit stop mode, set the ILVL2 to ILVL0 bits to “$000_2$”.
2. Set the I flag to “1”.
3. Enable the peripheral function whose interrupt is to be used to exit stop mode.
   In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt service routine is executed.

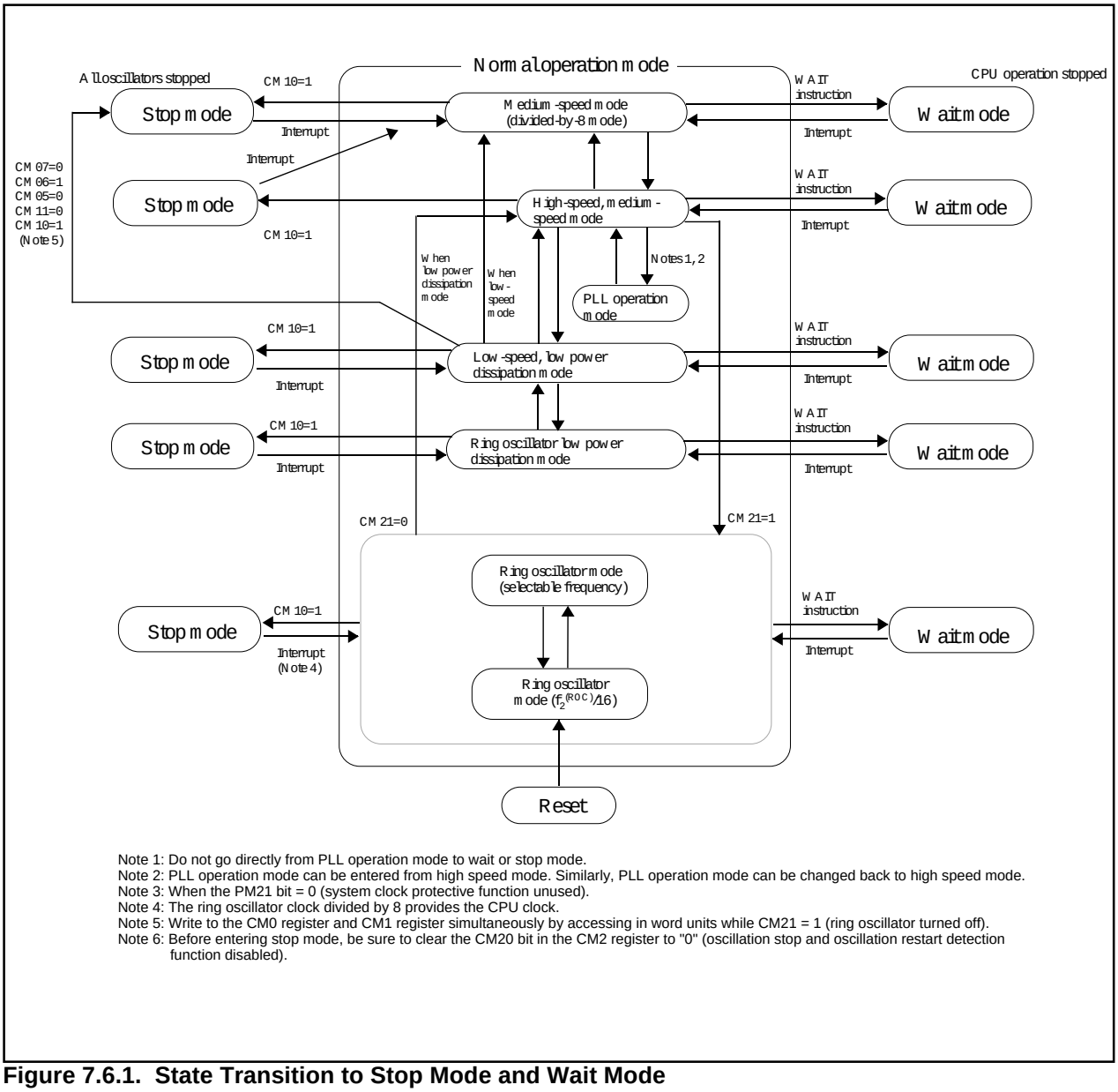
Which CPU clock will be used after exiting stop mode by a peripheral function or $\overline{\text{NMI}}$ interrupt is determined by the CPU clock that was on when the microcomputer was placed into stop mode as follows:

If the CPU clock before entering stop mode was derived from the sub clock: sub clock

If the CPU clock before entering stop mode was derived from the main clock: main clock divide-by-8

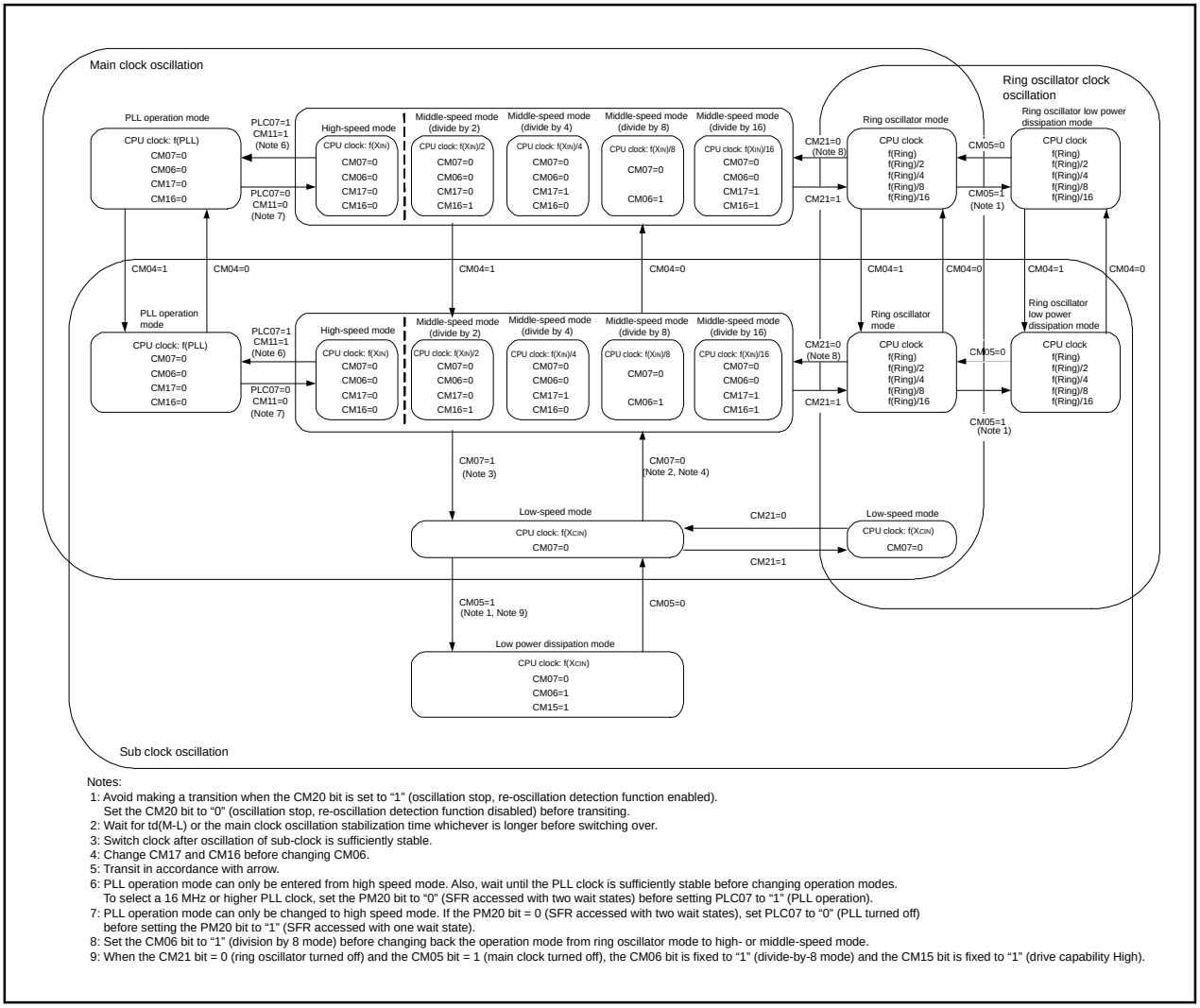
If the CPU clock before entering stop mode was derived from the ring oscillator clock: ring oscillator clock divide-by-8

Figure 7.6.1 shows the state transition from normal operation mode to stop mode and wait mode. Figure 7.6.1.1 shows the state transition in normal operation mode.
Table 7.6.1 shows a state transition matrix describing allowed transition and setting. The vertical line shows current state and horizontal line shows state after transition.




Note 1: Do not go directly from PLL operation mode to wait or stop mode.
Note 2: PLL operation mode can be entered from high speed mode. Similarly, PLL operation mode can be changed back to high speed mode.
Note 3: When the PM21 bit = 0 (system clock protective function unused).
Note 4: The ring oscillator clock divided by 8 provides the CPU clock.
Note 5: Write to the CM0 register and CM1 register simultaneously by accessing in word units while CM21 = 1 (ring oscillator turned off).
Note 6: Before entering stop mode, be sure to clear the CM20 bit in the CM2 register to "0" (oscillation stop and oscillation restart detection function disabled).

**Figure 7.6.1. State Transition to Stop Mode and Wait Mode**




Notes:
1: Avoid making a transition when the CM20 bit is set to “1” (oscillation stop, re-oscillation detection function enabled).
Set the CM20 bit to “0” (oscillation stop, re-oscillation detection function disabled) before transiting.
2: Wait for td(M-L) or the main clock oscillation stabilization time whichever is longer before switching over.
3: Switch clock after oscillation of sub-clock is sufficiently stable.
4: Change CM17 and CM16 before changing CM06.
5: Transit in accordance with arrow.
6: PLL operation mode can only be entered from high speed mode. Also, wait until the PLL clock is sufficiently stable before changing operation modes.
To select a 16 MHz or higher PLL clock, set the PM20 bit to “0” (SFR accessed with two wait states) before setting PLC07 to “1” (PLL operation).
7: PLL operation mode can only be changed to high speed mode. If the PM20 bit = 0 (SFR accessed with two wait states), set PLC07 to “0” (PLL turned off) before setting the PM20 bit to “1” (SFR accessed with one wait state).
8: Set the CM06 bit to “1” (division by 8 mode) before changing back the operation mode from ring oscillator mode to high- or middle-speed mode.
9: When the CM21 bit = 0 (ring oscillator turned off) and the CM05 bit = 1 (main clock turned off), the CM06 bit is fixed to “1” (divide-by-8 mode) and the CM15 bit is fixed to “1” (drive capability High).

**Figure 7.6.1.1. State Transition in Normal Mode**

## Table 7.6.1. Allowed Transition and Setting

| Current state | State after transition: High-speed mode, middle-speed mode | Low-speed mode[2] | Low power dissipation mode | PLL operation mode[2] | Ring oscillator mode | Ring oscillator low power dissipation mode | Stop mode | Wait mode |
|---|---|---|---|---|---|---|---|---|
| High-speed mode, middle-speed mode | See Table A[8] | (9)[7] | -- | (13)[3] | (15) | -- | (16)[1] | (17) |
| Low-speed mode[2] | (8) | | (11)[1, 6] | -- | -- | -- | (16)[1] | (17) |
| Low power dissipation mode | -- | (10) | | -- | -- | -- | (16)[1] | (17) |
| PLL operation mode[2] | (12)[3] | -- | -- | | -- | -- | -- | -- |
| Ring oscillator mode | (14)[4] | -- | -- | -- | See Table A[8] | (11)[1] | (16)[1] | (17) |
| Ring oscillator low power dissipation mode | -- | -- | -- | -- | (10) | See Table A[8] | (16)[1] | (17) |
| Stop mode | (18)[5] | (18) | (18) | -- | (18)[5] | (18)[5] | | -- |
| Wait mode | (18) | (18) | (18) | -- | (18) | (18) | -- | |

--: Cannot transit

Notes:
1. Avoid making a transition when the CM21 bit is set to "1" (oscillation stop, re-oscillation detection function enabled). Set the CM21 bit to "0" (oscillation stop, re-oscillation detection function disabled) before transiting.
2. Ring oscillator clock oscillates and stops in low-speed mode. In this mode, the ring oscillator can be used as peripheral function clock. Sub clock oscillates and stops in PLL operation mode. In this mode, sub clock can be used as peripheral function clock.
3. PLL operation mode can only be entered from and changed to high-speed mode.
4. Set the CM06 bit to "1" (division by 8 mode) before transiting from ring oscillator mode to high- or middle-speed mode.
5. When exiting stop mode, the CM06 bit is set to "1" (division by 8 mode).
6. If the CM05 bit is set to "1" (main clock stop), then the CM06 bit is set to "1" (division by 8 mode).
7. A transition can be made only when sub clock is oscillating.
8. State transitions within the same mode (divide-by-n values changed or subclock oscillation turned on or off) are shown in the table below.

| | | Sub clock oscillating: No division | Divided by 2 | Divided by 4 | Divided by 8 | Divided by 16 | Sub clock turned off: No division | Divided by 2 | Divided by 4 | Divided by 8 | Divided by 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Sub clock oscillating | No division | | (4) | (5) | (7) | (6) | (1) | -- | -- | -- | -- |
| | Divided by 2 | (3) | | (5) | (7) | (6) | -- | (1) | -- | -- | -- |
| | Divided by 4 | (3) | (4) | | (7) | (6) | -- | -- | (1) | -- | -- |
| | Divided by 8 | (3) | (4) | (5) | | (6) | -- | -- | -- | (1) | -- |
| | Divided by 16 | (3) | (4) | (5) | (7) | | -- | -- | -- | -- | (1) |
| Sub clock turned off | No division | (2) | -- | -- | -- | -- | | (4) | (5) | (7) | (6) |
| | Divided by 2 | -- | (2) | -- | -- | -- | (3) | | (5) | (7) | (6) |
| | Divided by 4 | -- | -- | (2) | -- | -- | (3) | (4) | | (7) | (6) |
| | Divided by 8 | -- | -- | -- | (2) | -- | (3) | (4) | (5) | | (6) |
| | Divided by 16 | -- | -- | -- | -- | (2) | (3) | (4) | (5) | (7) | |

9. ( ) : setting method. Refer to following table.

--: Cannot transit

| | Setting | Operation |
|---|---|---|
| (1) | CM04 = 0 | Sub clock turned off |
| (2) | CM04 = 1 | Sub clock oscillating |
| (3) | CM06 = 0, CM17 = 0 , CM16 = 0 | CPU clock no division mode |
| (4) | CM06 = 0, CM17 = 0 , CM16 = 1 | CPU clock division by 2 mode |
| (5) | CM06 = 0, CM17 = 1 , CM16 = 0 | CPU clock division by 4 mode |
| (6) | CM06 = 0, CM17 = 1 , CM16 = 1 | CPU clock division by 16 mode |
| (7) | CM06 = 1 | CPU clock division by 8 mode |
| (8) | CM07 = 0 | Main clock, PLL clock, or ring oscillator clock selected |
| (9) | CM07 = 1 | Sub clock selected |
| (10) | CM05 = 0 | Main clock oscillating |
| (11) | CM05 = 1 | Main clock turned off |
| (12) | PLC07 = 0, CM11 = 0 | Main clock selected |
| (13) | PLC07 = 1, CM11 = 1 | PLL clock selected |
| (14) | CM21 = 0 | Main clock or PLL clock selected |
| (15) | CM21 = 1 | Ring oscillator clock selected |
| (16) | CM10 = 1 | Transition to stop mode |
| (17) | wait instruction | Transition to wait mode |
| (18) | Hardware interrupt | Exit stop mode or wait mode |

CM04, CM05, CM06, CM07 : bit of CM0 register
CM10, CM11, CM16, CM17 : bit of CM1 register
CM20, CM21 : bit of CM2 register
PLC07 : bit of PLC0 register

## 7.7 System Clock Protective Function

When the main clock is selected for the CPU clock source, this function protects the clock from modifications in order to prevent the CPU clock from becoming halted by run-away.

If the PM21 bit of PM2 register is set to “1” (clock modification disabled), the following bits are protected against writes:

- CM02, CM05, and CM07 bits in CM0 register
- CM10, CM11 bits in CM1 register
- CM20 bit in CM2 register
- All bits in PLC0 register

Before the system clock protective function can be used, the following register settings must be made while the CM05 bit of CM0 register is “0” (main clock oscillating) and CM07 bit is “0” (main clock selected for the CPU clock source):

(1) Set the PRC1 bit of PRCR register to “1” (enable writes to PM2 register).
(2) Set the PM21 bit of PM2 register to “1” (disable clock modification).
(3) Set the PRC1 bit of PRCR register to “0” (disable writes to PM2 register).

Do not execute the WAIT instruction when the PM21 bit is “1”.

## 7.8 Oscillation Stop and Re-oscillation Detect Function

The oscillation stop and re-oscillation detect function allows the detection of main clock oscillation stop and reoscillation. At oscillation stop or re-oscillation detection, reset or oscillation stop, re-oscillation detection interrupt are generated. Depending on the CM27 bit of CM2 register. The oscillation stop detection function can be enabled and disabled by the CM20 bit in the CM2 register. Table 7.8.1 lists a specification overview of the oscillation stop and re-oscillation detect function.

**Table 7.8.1. Specification Overview of Oscillation Stop and Re-oscillation Detect Function**

| Item | Specification |
|---|---|
| Oscillation stop detectable clock and frequency bandwidth | $f(X_{IN}) \geq 2$ MHz |
| Enabling condition for oscillation stop, re-oscillation detection function | Set CM20 bit to “1”(enable) |
| Operation at oscillation stop, re-oscillation detection | •Reset occurs (when CM27 bit =0)<br>•Oscillation stop, re-oscillation detection interrupt occurs(when CM27 bit =1) |

### 7.8.1 Operation When CM27 bit = 0 (Oscillation Stop Detection Reset)

When main clock stop is detected when the CM20 bit is “1” (oscillation stop, re-oscillation detection function enabled), the microcomputer is initialized, coming to a halt (oscillation stop reset; refer to “SFR”, “Reset”).

This status is reset with hardware reset 1 or hardware reset 2. Also, even when re-oscillation is detected, the microcomputer can be initialized and stopped; it is, however, necessary to avoid such usage. (During main clock stop, do not set the CM20 bit to “1” and the CM27 bit to “0”.)

### 7.8.2 Operation When CM27 bit = 1 (Oscillation Stop and Re-oscillation Detect Interrupt)

When the main clock corresponds to the CPU clock source and the CM20 bit is “1” (oscillation stop and re-oscillation detect function enabled), the system is placed in the following state if the main clock comes to a halt:

- Oscillation stop and re-oscillation detect interrupt request occurs.
- The ring oscillator starts oscillation, and the ring oscillator clock becomes the CPU clock and clock source for peripheral functions in place of the main clock.
- CM21 bit = 1 (ring oscillator clock for CPU clock source)
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)

When the PLL clock corresponds to the CPU clock source and the CM20 bit is “1”, the system is placed in the following state if the main clock comes to a halt: Since the CM21 bit remains unchanged, set it to “1” (ring oscillator clock) inside the interrupt routine.

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)
- CM21 bit remains unchanged

When the CM20 bit is “1”, the system is placed in the following state if the main clock re-oscillates from the stop condition:

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit = 1 (main clock re-oscillation detected)
- CM23 bit = 0 (main clock oscillation)
- CM21 bit remains unchanged

### 7.8.3 How to Use Oscillation Stop and Re-oscillation Detect Function

- The oscillation stop and re-oscillation detect interrupt shares the vector with the watchdog timer interrupt. If the oscillation stop, re-oscillation detection and watchdog timer interrupts both are used, read the CM22 bit in an interrupt routine to determine which interrupt source is requesting the interrupt.
- Where the main clock re-oscillated after oscillation stop, return the main clock to the CPU clock and peripheral function clock source in the program. Figure 7.8.3.1 shows the procedure for switching the clock source from the ring oscillator to the main clock.
- Simultaneously with oscillation stop, re-oscillation detection interrupt occurrence, the CM22 bit becomes “1”. When the CM22 bit is set at “1”, oscillation stop, re-oscillation detection interrupt are disabled. By setting the CM22 bit to “0” in the program, oscillation stop, re-oscillation detection interrupt are enabled.
- If the main clock stops during low speed mode where the CM20 bit is “1”, an oscillation stop, re-oscillation detection interrupt request is generated. At the same time, the ring oscillator starts oscillating. In this case, although the CPU clock is derived from the sub clock as it was before the interrupt occurred, the peripheral function clocks now are derived from the ring oscillator clock.
- To enter wait mode while using the oscillation stop, re-oscillation detection function, set the CM02 bit to “0” (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop, re-oscillation detection function is provided in preparation for main clock stop due to external factors, set the CM20 bit to “0” (Oscillation stop, re-oscillation detection function disabled) where the main clock is stopped or oscillated in the program, that is where the stop mode is selected or the CM05 bit is altered.
- This function cannot be used if the main clock frequency is 2 MHz or less. In that case, set the CM20 bit to “0”.

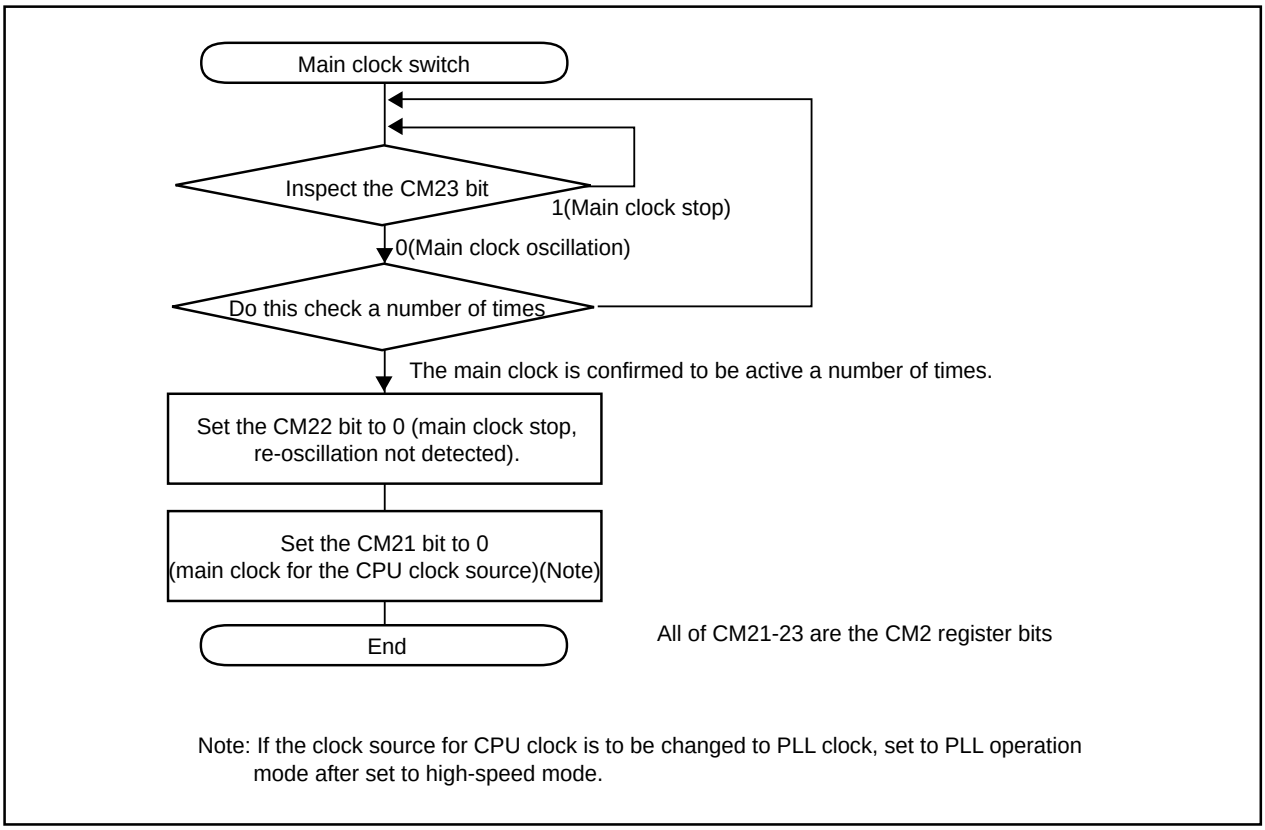



**Figure 7.8.3.1. Procedure to Switch Clock Source From Ring Oscillator to Main Clock**

# 8. Protection

In the event that a program runs out of control, this function protects the important registers so that they will not be rewritten easily. Figure 8.1 shows the PRCR register. The following lists the registers protected by the PRCR register.

- Registers protected by PRC0 bit: CM0, CM1, CM2, PLC0, ROCR and PCLKR registers
- Registers protected by PRC1 bit: PM0, PM1, PM2, TB2SC, INVC0 and INVC1 registers
- Registers protected by PRC2 bit: PD9 , PACR and S4C registers
- Registers protected by PRC3 bit: VCR2 and D4INT registers

Set the PRC2 bit to “1” (write enabled) and then write to any address, and the PRC2 bit will be cleared to “0” (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to “1”. Make sure no interrupts or DMA transfers will occur between the instruction in which the PRC2 bit is set to “1” and the next instruction. The PRC0, PRC1 and PRC3 bits are not automatically cleared to “0” by writing to any address. They can only be cleared in a program.

Protect register

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

| Symbol | Address | After reset |
|---|---|---|
| PRCR | $000A_{16}$ | $XX000000_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PRC0 | Protect bit 0 | Enable write to CM0, CM1, CM2, ROCR, PLC0 and PCLKR registers<br>0 : Write protected<br>1 : Write enabled | RW |
| PRC1 | Protect bit 1 | Enable write to PM0, PM1, PM2, TB2SC, INVC0 and INVC1 registers<br>0 : Write protected<br>1 : Write enabled | RW |
| PRC2 | Protect bit 2 | Enable write to PD9, PACR and S4C registers<br>0 : Write protected<br>1 : Write enabled | RW |
| PRC3 | Protect bit 3 | Enable write to VCR2 and D4INT registers<br>0 : Write protected<br>1 : Write enabled | RW |
| $\overline{\text{(b5-b4)}}$ | Reserved bit | Must set to 0 | RW |
| $\overline{\text{(b7-b6)}}$ | Nothing is assigned. When write, set to 0 . When read, its content is indeterminate. | | — |

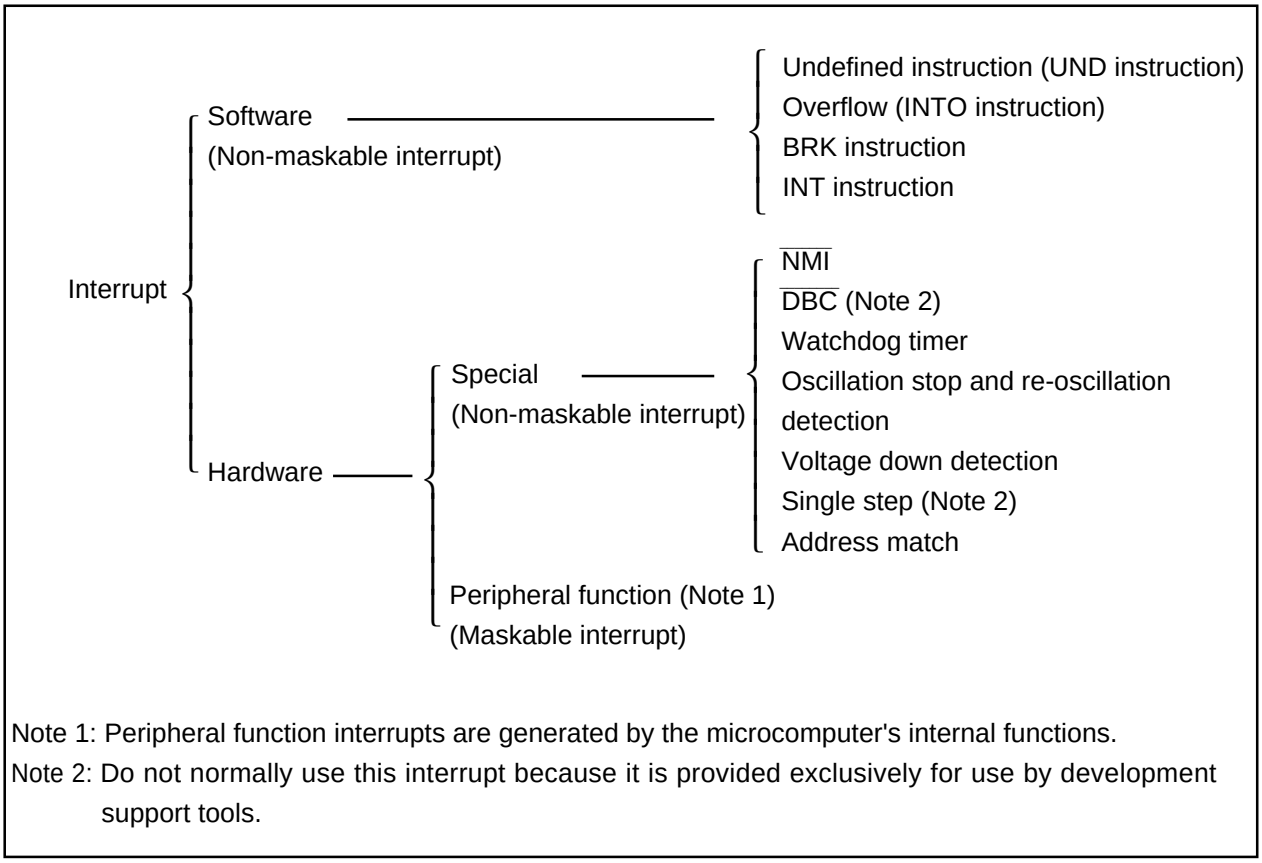
Note: The PRC2 bit is set to 0 by writing to any address after setting it to 1 . Other bits are not set to 0 by writing to any address, and must therefore be set in a program.

**Figure 8.1. PRCR Register**

# 9. Interrupts

## 9.1 Type of Interrupts

Figure 9.1.1 shows types of interrupts.




Note 1: Peripheral function interrupts are generated by the microcomputer's internal functions.
Note 2: Do not normally use this interrupt because it is provided exclusively for use by development support tools.

**Figure 9.1.1. Interrupts**

- Maskable Interrupt: An interrupt which can be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **can be changed** by priority level.
- Non-maskable Interrupt: An interrupt which cannot be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **cannot be changed** by priority level.

### 9.1.1 Software Interrupts

A software interrupt occurs when executing certain instructions. Software interrupts are non-maskable interrupts.

#### 9.1.1.1 Undefined Instruction Interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

#### 9.1.1.2 Overflow Interrupt

An overflow interrupt occurs when executing the INTO instruction with the O flag set to “1” (the operation resulted in an overflow). The following are instructions whose O flag changes by arithmetic: ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB

#### 9.1.1.3 BRK Interrupt

A BRK interrupt occurs when executing the BRK instruction.

#### 9.1.1.4 INT Instruction Interrupt

An INT instruction interrupt occurs when executing the INT instruction. Software interrupt Nos. 0 to 63 can be specified for the INT instruction. Because software interrupt Nos. 4 to 31 are assigned to peripheral function interrupts, the same interrupt routine as for peripheral function interrupts can be executed by executing the INT instruction.

In software interrupt Nos. 0 to 31, the U flag is saved to the stack during instruction execution and is cleared to “0” (ISP selected) before executing an interrupt sequence. The U flag is restored from the stack when returning from the interrupt routine. In software interrupt Nos. 32 to 63, the U flag does not change state during instruction execution, and the SP then selected is used.

### 9.1.2 Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral function interrupts.

#### 9.1.2.1 Special Interrupts

Special interrupts are non-maskable interrupts.

##### 9.1.2.1.1 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low. For details about the $\overline{\text{NMI}}$ interrupt, refer to the section "NMI interrupt".

##### 9.1.2.1.2 $\overline{\text{DBC}}$ Interrupt

This interrupt is exclusively for debugger, do not use in any other circumstances.

##### 9.1.2.1.3 Watchdog Timer Interrupt

Generated by the watchdog timer. Once a watchdog timer interrupt is generated, be sure to initialize the watchdog timer. For details about the watchdog timer, refer to the section "watchdog timer".

##### 9.1.2.1.4 Oscillation Stop and Re-oscillation Detection Interrupt

Generated by the oscillation stop and re-oscillation detection function. For details about the oscillation stop and re-oscillation detection function, refer to the section "clock generating circuit".

##### 9.1.2.1.5 Voltage Down Detection Interrupt

Generated by the voltage detection circuit. For details about the voltage detection circuit, refer to the section "voltage detection circuit".

##### 9.1.2.1.6 Single-step Interrupt

Do not normally use this interrupt because it is provided exclusively for use by development support tools.

##### 9.1.2.1.7 Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by the RMAD0 or RMAD1 register, if the corresponding enable bit (AIER register’s AIER0 or AIER1bit) is set to “1”. For details about the address match interrupt, refer to the section “address match interrupt”.

#### 9.1.2.2 Peripheral Function Interrupts

Peripheral function interrupts are maskable interrupts and generated by the microcomputer's internal functions. The interrupt sources for peripheral function interrupts are listed in “Table 1.11.2. Relocatable Vector Tables”. For details about the peripheral functions, refer to the description of each peripheral function in this manual.

## 9.2 Interrupts and Interrupt Vector

One interrupt vector consists of 4 bytes. Set the start address of each interrupt routine in the respective interrupt vectors. When an interrupt request is accepted, the CPU branches to the address set in the corresponding interrupt vector. Figure 9.2.1 shows the interrupt vector.

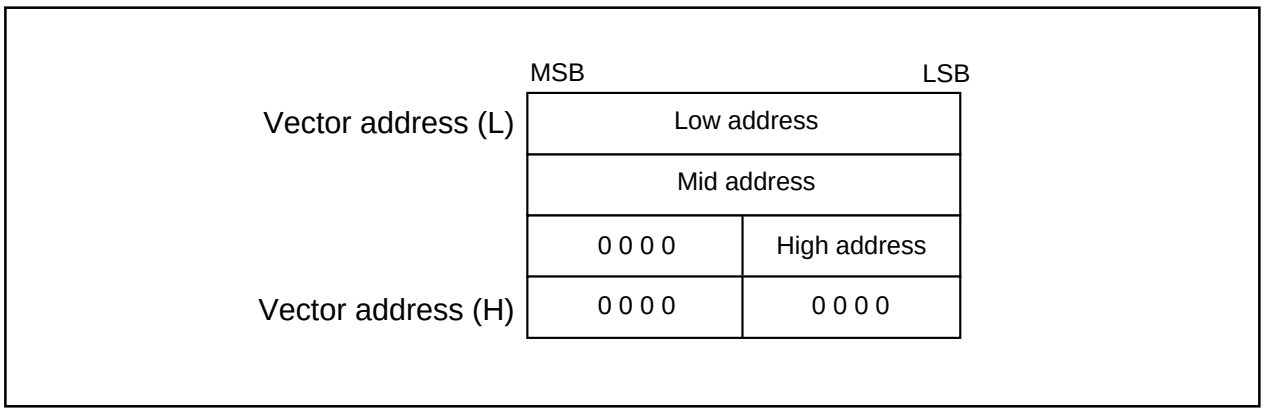



**Figure 9.2.1. Interrupt Vector**

### 9.2.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDC16 to FFFFF16. Table 9.2.1.1 lists the fixed vector tables. In the flash memory version of microcomputer, the vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to the section "flash memory rewrite disabling function".

**Table 9.2.1.1. Fixed Vector Tables**

| Interrupt source | Vector table addresses Address (L) to address (H) | Remarks | Reference |
|---|---|---|---|
| Undefined instruction | FFFDC16 to FFFDF16 | Interrupt on UND instruction | M16C/60, M16C/20 serise software maual |
| Overflow | FFFE016 to FFFE316 | Interrupt on INTO instruction | |
| BRK instruction | FFFE416 to FFFE716 | If the contents of address FFFE716 is FF16, program execution starts from the address shown by the vector in the relocatable vector table. | |
| Address match | FFFE816 to FFFEB16 | | Address match interrupt |
| Single step (Note) | FFFEC16 to FFFEF16 | | |
| Watchdog timer<br>Oscillation stop and re-oscillation detection<br>Voltage down detection | FFFF016 to FFFF316 | | Watchdog timer<br>Clock generating circuit<br>Voltage detection circuit |
| $\overline{\text{DBC}}$ (Note) | FFFF416 to FFFF716 | | |
| $\overline{\text{NMI}}$ | FFFF816 to FFFFB16 | | $\overline{\text{NMI}}$ interrupt |
| Reset | FFFFC16 to FFFFF16 | | Reset |

Note: Do not normally use this interrupt because it is provided exclusively for use by development support tools.

### 9.2.2 Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a reloacatable vector table area. Table 9.2.2.1 lists the relocatable vector tables. Setting an even address in the INTB register results in the interrupt sequence being executed faster than in the case of odd addresses.

**Table 9.2.2.1. Relocatable Vector Tables**

| Interrupt source | Vector address (Note 1) Address (L) to address (H) | Software interrupt number | Reference |
|---|---|---|---|
| BRK instruction (Note 5) | +0 to +3 ($0000_{16}$ to $0003_{16}$) | 0 | M16C/60, M16C/20 series software manual |
| ——— (Reserved) | | 1 to 3 | |
| $\overline{\text{INT3}}$ | +16 to +19 ($0010_{16}$ to $0013_{16}$) | 4 | $\overline{\text{INT}}$ interrupt |
| ICOC interrupt 0 | +20 to +23 ($0014_{16}$ to $0017_{16}$) | 5 | Timer S |
| ICOC interrupt 1, I²C-BUS interface (Note 4) | +24 to +27 ($0018_{16}$ to $001B_{16}$) | 6 | Timer S Multi-Master I²C-BUS interface |
| ICOC base timer, SCL/SDA (Note 4) | +28 to +31 ($001C_{16}$ to $001F_{16}$) | 7 | |
| SI/O4, $\overline{\text{INT5}}$ (Note 2) | +32 to +35 ($0020_{16}$ to $0023_{16}$) | 8 | $\overline{\text{INT}}$ interrupt Serial I/O |
| SI/O3, $\overline{\text{INT4}}$ (Note 2) | +36 to +39 ($0024_{16}$ to $0027_{16}$) | 9 | |
| UART 2 bus collision detection (Note 6) | +40 to +43 ($0028_{16}$ to $002B_{16}$) | 10 | Serial I/O |
| DMA0 | +44 to +47 ($002C_{16}$ to $002F_{16}$) | 11 | DMAC |
| DMA1 | +48 to +51 ($0030_{16}$ to $0033_{16}$) | 12 | |
| Key input interrupt | +52 to +55 ($0034_{16}$ to $0037_{16}$) | 13 | Key input interrupt |
| A-D | +56 to +59 ($0038_{16}$ to $003B_{16}$) | 14 | A-D convertor |
| UART2 transmit, NACK2 (Note 3) | +60 to +63 ($003C_{16}$ to $003F_{16}$) | 15 | Serial I/O |
| UART2 receive, ACK2 (Note 3) | +64 to +67 ($0040_{16}$ to $0043_{16}$) | 16 | |
| UART0 transmit | +68 to +71 ($0044_{16}$ to $0047_{16}$) | 17 | |
| UART0 receive | +72 to +75 ($0048_{16}$ to $004B_{16}$) | 18 | |
| UART1 transmit | +76 to +79 ($004C_{16}$ to $004F_{16}$) | 19 | |
| UART1 receive | +80 to +83 ($0050_{16}$ to $0053_{16}$) | 20 | |
| Timer A0 | +84 to +87 ($0054_{16}$ to $0057_{16}$) | 21 | Timer |
| Timer A1 | +88 to +91 ($0058_{16}$ to $005B_{16}$) | 22 | |
| Timer A2 | +92 to +95 ($005C_{16}$ to $005F_{16}$) | 23 | |
| Timer A3 | +96 to +99 ($0060_{16}$ to $0063_{16}$) | 24 | |
| Timer A4 | +100 to +103 ($0064_{16}$ to $0067_{16}$) | 25 | |
| Timer B0 | +104 to +107 ($0068_{16}$ to $006B_{16}$) | 26 | |
| Timer B1 | +108 to +111 ($006C_{16}$ to $006F_{16}$) | 27 | |
| Timer B2 | +112 to +115 ($0070_{16}$ to $0073_{16}$) | 28 | |
| $\overline{\text{INT0}}$ | +116 to +119 ($0074_{16}$ to $0077_{16}$) | 29 | $\overline{\text{INT}}$ interrupt |
| $\overline{\text{INT1}}$ | +120 to +123 ($0078_{16}$ to $007B_{16}$) | 30 | |
| $\overline{\text{INT2}}$ | +124 to +127 ($007C_{16}$ to $007F_{16}$) | 31 | |
| Software interrupt (Note 5) | +128 to +131 ($0080_{16}$ to $0083_{16}$) to +252 to +255 ($00FC_{16}$ to $00FF_{16}$) | 32 to 63 | M16C/60, M16C/20 series software manual |

Note 1: Address relative to address in INTB.
Note 2: Use the IFSR register's IFSR6 and IFSR7 bits to select.
Note 3: During I²C Bus mode, NACK and ACK interrupts comprise the interrupt source.
Note 4: Use the IFSR2A register's IFSR26 and IFSR27 bits to select.
Note 5: These interrupts cannot be disabled using the I flag.
Note 6: Bus collision detection : During IE Bus mode, this bus collision detection constitutes the cause of an interrupt. During I²C Bus mode, however, a start condition or a stop condition detection constitutes the cause of an interrupt.

## 9.3 Interrupt Control

The following describes how to enable/disable the maskable interrupts, and how to set the priority in which order they are accepted. What is explained here does not apply to nonmaskable interrupts.
Use the FLG register's I flag, IPL, and each interrupt control register's ILVL2 to ILVL0 bits to enable/disable the maskable interrupts. Whether an interrupt is requested is indicated by the IR bit in each interrupt control register.
Figure 9.3.1 shows the interrupt control registers.

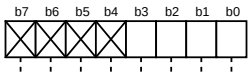
Also, the following interrupts share a vector and an interrupt control register.

- $\overline{\text{INT4}}$ and SIO3
- $\overline{\text{INT5}}$ and SIO4
- ICOC base timer and SCL/SDA
- ICOC interrupt 1 and $I^2C$-BUS interface

An interrupt request is set by the IFSR6, IFSR7 bits in the IFSR register and the IFSR26 and IFSR27 bits in the IFSR2A register. Figure 9.3.2 shows the IFSR, IFSR2A registers.

Interrupt control register (Note 2)

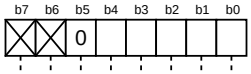
| Symbol | Address | After reset |
|---|---|---|
| ICOC0IC | $0045_{16}$ | $XXXXX000_2$ |
| ICOC1IC, IICIC (Note 3) | $0046_{16}$ | $XXXXX000_2$ |
| BTIC, SCLDAIC (Note 3) | $0047_{16}$ | $XXXXX000_2$ |
| BCNIC | $004A_{16}$ | $XXXXX000_2$ |
| DM0IC, DM1IC | $004B_{16}$, $004C_{16}$ | $XXXXX000_2$ |
| KUPIC | $004D_{16}$ | $XXXXX000_2$ |
| ADIC | $004E_{16}$ | $XXXXX000_2$ |
| S0TIC to S2TIC | $0051_{16}$, $0053_{16}$, $004F_{16}$ | $XXXXX000_2$ |
| S0RIC to S2RIC | $0052_{16}$, $0054_{16}$, $0050_{16}$ | $XXXXX000_2$ |
| TA0IC to TA4IC | $0055_{16}$ to $0059_{16}$ | $XXXXX000_2$ |
| TB0IC to TB2IC | $005A_{16}$ to $005C_{16}$ | $XXXXX000_2$ |




| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| ILVL0 | Interrupt priority 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 bit | 0 : Interrupt not requested<br>1 : Interrupt requested | RW<br>(Note 1) |
| —<br>(b7-b4) | No functions are assigned.<br>When writing to these bits, write “0”. The values in these bits when read are indeterminate. | | — |

Note 1: This bit can only be reset by writing “0” (Do not write “1”).
Note 2: To rewrite the interrupt control registers, do so at a point that does not generate the interrupt request for that register. For details, see the “precautions for interrupts” of the Usage Notes Reference Book.
Note 3: Use the IFSR2A register to select.

| Symbol | Address | After reset |
|---|---|---|
| INT3IC | $0044_{16}$ | $XX00X000_2$ |
| S4IC/INT5IC | $0048_{16}$ | $XX00X000_2$ |
| S3IC/INT4IC | $0049_{16}$ | $XX00X000_2$ |
| INT0IC to INT2IC | $005D_{16}$ to $005F_{16}$ | $XX00X000_2$ |

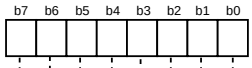



| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| ILVL0 | Interrupt priority level select bit | b2 b1 b0<br>0 0 0 : Level 0 (interrupt disabled)<br>0 0 1 : Level 1<br>0 1 0 : Level 2<br>0 1 1 : Level 3<br>1 0 0 : Level 4<br>1 0 1 : Level 5<br>1 1 0 : Level 6<br>1 1 1 : Level 7 | RW |
| ILVL1 | | | RW |
| ILVL2 | | | RW |
| IR | Interrupt request bit | 0: Interrupt not requested<br>1: Interrupt requested | RW<br>(Note 1) |
| POL | Polarity select bit | 0 : Selects falling edge (Notes 3, 4)<br>1 : Selects rising edge | RW |
| —<br>(b5) | Reserved bit | Must always be set to “0” | RW |
| —<br>(b7-b6) | No functions are assigned.<br>When writing to these bits, write “0”. The values in these bits when read are indeterminate. | | RW |

Note 1: This bit can only be reset by writing “0” (Do not write “1”).
Note 2: To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. For details, see the “precautions for interrupts” of the Usage Notes Reference Book.
Note 3: If the IFSR register’s IFSRi bit (i = 0 to 5) is “1” (both edges), set the INTiIC register’s POL bit to “0” (falling edge).
Note 4: Set the S3IC or S4IC register’s POL bit to “0” (falling edge) when the IFSR register’s IFSR6 bit = 0 (SI/O3 selected) or IFSR7 bit = 0 (SI/O4 selected), respectively.

**Figure 9.3.1. Interrupt Control Registers**

Interrupt request cause select register

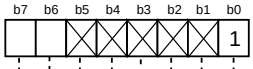



Symbol: IFSR | Address: 035F16 | After reset: 0016

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| IFSR0 | INT0 interrupt polarity switching bit | 0 : One edge<br>1 : Both edges (Note 1) | RW |
| IFSR1 | INT1 interrupt polarity switching bit | 0 : One edge<br>1 : Both edges (Note 1) | RW |
| IFSR2 | INT2 interrupt polarity switching bit | 0 : One edge<br>1 : Both edges (Note 1) | RW |
| IFSR3 | INT3 interrupt polarity switching bit | 0 : One edge<br>1 : Both edges (Note 1) | RW |
| IFSR4 | INT4 interrupt polarity switching bit | 0 : One edge<br>1 : Both edges (Note 1) | RW |
| IFSR5 | INT5 interrupt polarity switching bit | 0 : One edge<br>1 : Both edges (Note 1) | RW |
| IFSR6 | Interrupt request cause select bit | 0 : SI/O3 (Note 2)<br>1 : $\overline{\text{INT4}}$ | RW |
| IFSR7 | Interrupt request cause select bit | 0 : SI/O4 (Note 2)<br>1 : $\overline{\text{INT5}}$ | RW |

Note 1: When setting this bit to “1” (= both edges), make sure the INT0IC to INT5IC register’s POL bit is set to “0” (= falling edge).
Note 2: When setting this bit to “0” (= SI/O3, SI/O4), make sure the S3IC and S4IC registers’ POL bit is set to “0” (= falling edge).

Interrupt request cause select register 2




Symbol: IFSR2A | Address: 035E16 | After reset: 00XXXXX02

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| IFSR20 | Reserved bit (Note 1) | Must be set to “1”. | RW |
| — (b5-b1) | Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate. | | — |
| IFSR26 | Interrupt request cause select bit | 0 : ICOC base timer<br>1 : SCL/SDA | RW |
| IFSR27 | Interrupt request cause select bit | 0 : ICOC interrupt 1<br>1 : I2C-BUS interface | RW |

Note 1: Set this bit to "1" befor you enable interrupt after resetting.

**Figure 9.3.2. IFSR Register and IFSR2A Register**

### 9.3.1 I Flag

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

### 9.3.2 IR Bit

The IR bit is set to “1” (= interrupt requested) when an interrupt request is generated. Then, when the interrupt request is accepted and the CPU branches to the corresponding interrupt vector, the IR bit is cleared to “0” (= interrupt not requested).
The IR bit can be cleared to “0” in a program. Note that do not write “1” to this bit.

### 9.3.3 ILVL2 to ILVL0 Bits and IPL

Interrupt priority levels can be set using the ILVL2 to ILVL0 bits.
Table 1.11.3 shows the settings of interrupt priority levels and Table 1.11.4 shows the interrupt priority levels enabled by the IPL.

The following are conditions under which an interrupt is accepted:
· I flag = “1”
· IR bit = “1”
· interrupt priority level > IPL

The I flag, IR bit, ILVL2 to ILVL0 bits and IPL are independent of each other. In no case do they affect one another.

**Table 9.3.3.1. Settings of Interrupt Priority Levels**

| ILVL2 to ILVL0 bits | Interrupt priority level | Priority order |
|---|---|---|
| $000_2$ | Level 0 (interrupt disabled) | ——— |
| $001_2$ | Level 1 | Low ↓ |
| $010_2$ | Level 2 | |
| $011_2$ | Level 3 | |
| $100_2$ | Level 4 | |
| $101_2$ | Level 5 | |
| $110_2$ | Level 6 | |
| $111_2$ | Level 7 | High |

**Table 9.3.3.2. Interrupt Priority Levels Enabled by IPL**

| IPL | Enabled interrupt priority levels |
|---|---|
| $000_2$ | Interrupt levels 1 and above are enabled |
| $001_2$ | Interrupt levels 2 and above are enabled |
| $010_2$ | Interrupt levels 3 and above are enabled |
| $011_2$ | Interrupt levels 4 and above are enabled |
| $100_2$ | Interrupt levels 5 and above are enabled |
| $101_2$ | Interrupt levels 6 and above are enabled |
| $110_2$ | Interrupt levels 7 and above are enabled |
| $111_2$ | All maskable interrupts are disabled |

## 9.4 Interrupt Sequence

An interrupt sequence (the device behavior from the instant an interrupt is accepted to the instant the interrupt routine is executed) is described here.

If an interrupt occurs during execution of an instruction, the processor determines its priority when the execution of the instruction is completed, and transfers control to the interrupt sequence from the next cycle. If an interrupt occurs during execution of either the SMOVB, SMOVF, SSTR or RMPA instruction, the processor temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

The CPU behavior during the interrupt sequence is described below. Figure 9.4.1 shows time required for executing the interrupt sequence.

(1) The CPU gets interrupt information (interrupt number and interrupt request priority level) by reading the address $00000_{16}$. Then it clears the IR bit for the corresponding interrupt to “0” (interrupt not requested).
(2) The FLG register immediately before entering the interrupt sequence is saved to the CPU’s internal temporary register(Note).
(3) The I, D and U flags in the FLG register become as follows:
The I flag is cleared to “0” (interrupts disabled).
The D flag is cleared to “0” (single-step interrupt disabled).
The U flag is cleared to “0” (ISP selected).
However, the U flag does not change state if an INT instruction for software interrupt Nos. 32 to 63 is executed.
(4) The CPU’s internal temporary register (Note) is saved to the stack.
(5) The PC is saved to the stack.
(6) The interrupt priority level of the accepted interrupt is set in the IPL.
(7) The start address of the relevant interrupt routine set in the interrupt vector is stored in the PC.

After the interrupt sequence is completed, the processor resumes executing instructions from the start address of the interrupt routine.

Note: This register cannot be used by user.

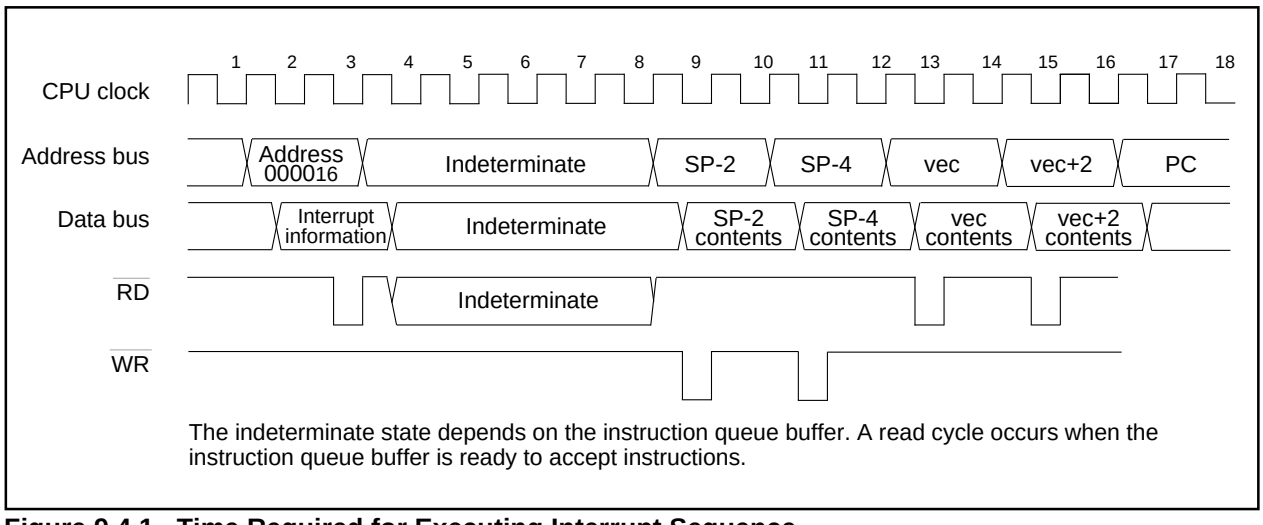
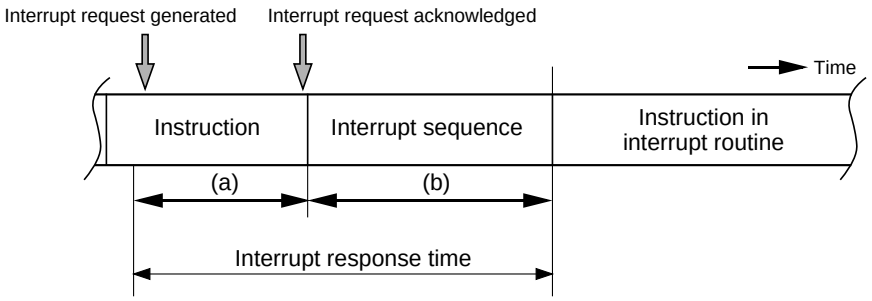



**Figure 9.4.1. Time Required for Executing Interrupt Sequence**

### 9.4.1 Interrupt Response Time

Figure 9.4.1.1 shows the interrupt response time. The interrupt response or interrupt acknowledge time denotes time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of the time from when an interrupt request is generated till when the instruction then executing is completed ((a) in Figure 9.4.1.1) and the time during which the interrupt sequence is executed ((b) in Figure 9.4.1.1).




(a) The time from when an interrupt request is generated till when the instruction then executing is completed. The length of this time varies with the instruction being executed. The DIVX instruction requires the longest time, which is equal to 30 cycles (without wait state, the divisor being a register).

(b) The time during which the interrupt sequence is executed. For details, see the table below. Note, however, that the values in this table must be increased 2 cycles for the $\overline{\text{DBC}}$ interrupt and 1 cycle for the address match and single-step interrupts.

| Interrupt vector address | SP value | Without wait |
|---|---|---|
| Even | Even | 18 cycles |
| Even | Odd | 19 cycles |
| Odd | Even | 19 cycles |
| Odd | Odd | 20 cycles |

**Figure 9.4.1.1. Interrupt response time**

### 9.4.2 Variation of IPL when Interrupt Request is Accepted

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

When a software interrupt or special interrupt request is accepted, one of the interrupt priority levels listed in Table 9.4.2.1 is set in the IPL. Shown in Table 9.4.2.1 are the IPL values of software and special interrupts when they are accepted.

**Table 9.4.2.1. IPL Level That is Set to IPL When A Software or Special Interrupt Is Accepted**

| Interrupt sources | Level that is set to IPL |
|---|---|
| Watchdog timer, $\overline{\text{NMI}}$, Oscillation stop and re-oscillation detection, voltage down detection | 7 |
| Software, address match, $\overline{\text{DBC}}$, single-step | Not changed |

### 9.4.3 Saving Registers

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

At this time, the 4 high-order bits of the PC and the 4 high-order (IPL) and 8 low-order bits of the FLG register, 16 bits in total, are saved to the stack first. Next, the 16 low-order bits of the PC are saved. Figure 9.4.3.1 shows the stack status before and after an interrupt request is accepted.

The other necessary registers must be saved in a program at the beginning of the interrupt routine. Use the PUSHM instruction, and all registers except SP can be saved with a single instruction.

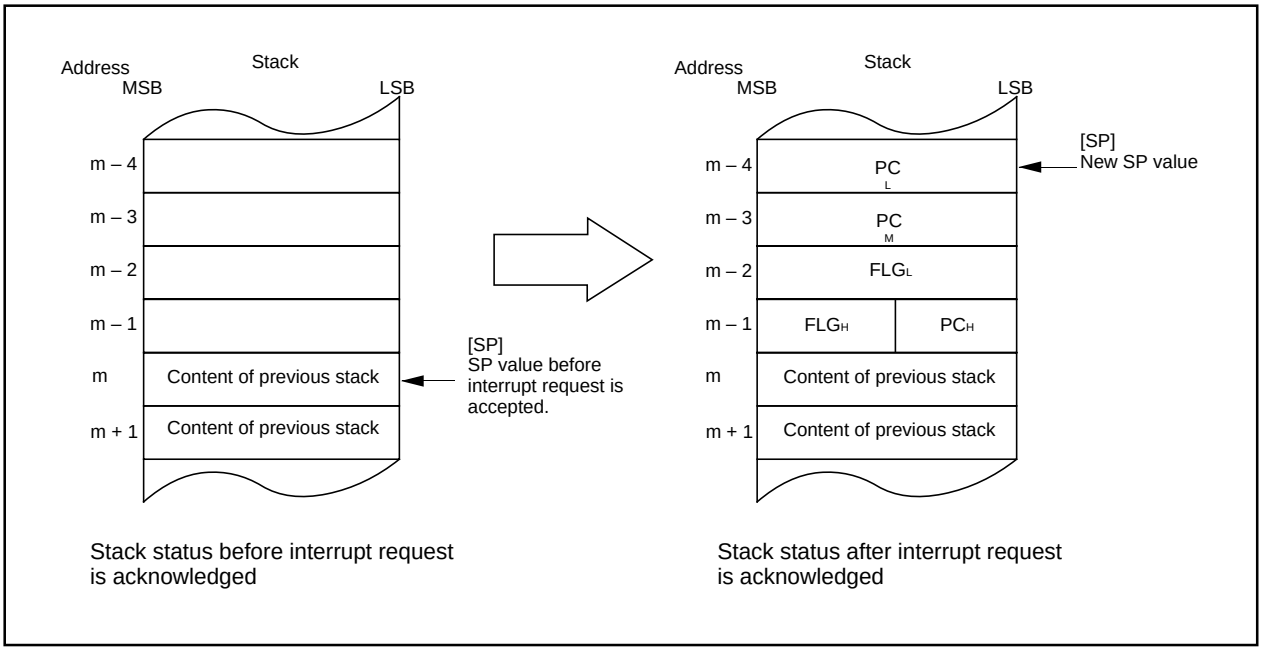
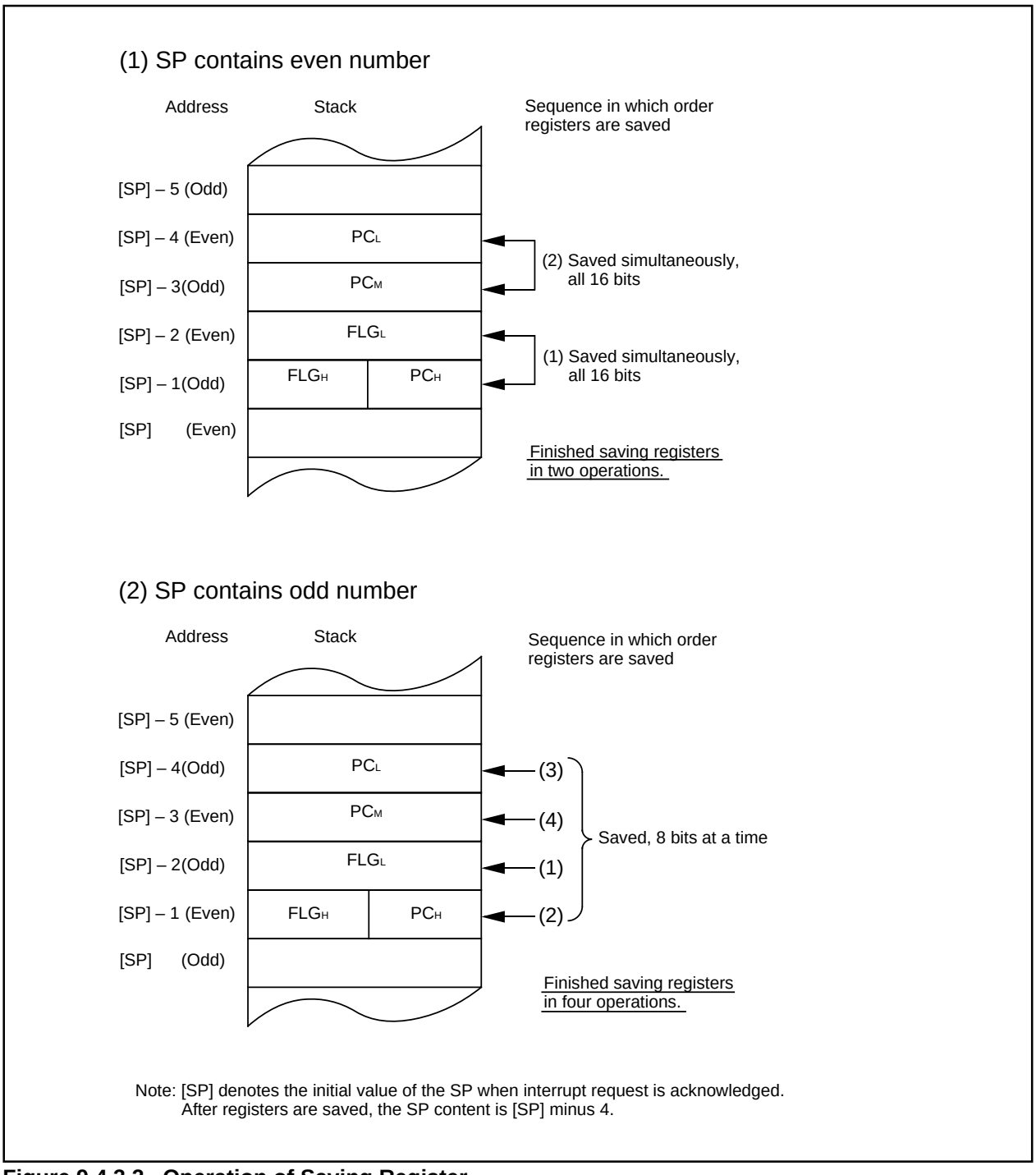



**Figure 9.4.3.1. Stack Status Before and After Acceptance of Interrupt Request**

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP(Note), at the time of acceptance of an interrupt request, is even or odd. If the stack pointer (Note) is even, the FLG register and the PC are saved, 16 bits at a time. If odd, they are saved in two steps, 8 bits at a time. Figure 9.4.3.2 shows the operation of the saving registers.

Note: When any INT instruction in software numbers 32 to 63 has been executed, this is the SP indicated by the U flag. Otherwise, it is the ISP.




Note: [SP] denotes the initial value of the SP when interrupt request is acknowledged.
After registers are saved, the SP content is [SP] minus 4.

**Figure 9.4.3.2. Operation of Saving Register**

### 9.4.4 Returning from an Interrupt Routine

The FLG register and PC in the state in which they were immediately before entering the interrupt sequence are restored from the stack by executing the REIT instruction at the end of the interrupt routine. Thereafter the CPU returns to the program which was being executed before accepting the interrupt request.

Return the other registers saved by a program within the interrupt routine using the POPM or similar instruction before executing the REIT instruction.

## 9.5 Interrupt Priority

If two or more interrupt requests are generated while executing one instruction, the interrupt request that has the highest priority is accepted.

For maskable interrupts (peripheral functions), any desired priority level can be selected using the ILVL2 to ILVL0 bits. However, if two or more maskable interrupts have the same priority level, their interrupt priority is resolved by hardware, with the highest priority interrupt accepted.

The watchdog timer and other special interrupts have their priority levels set in hardware. Figure 9.5.1 shows the priorities of hardware interrupts.

Software interrupts are not affected by the interrupt priority. If an instruction is executed, control branches invariably to the interrupt routine.

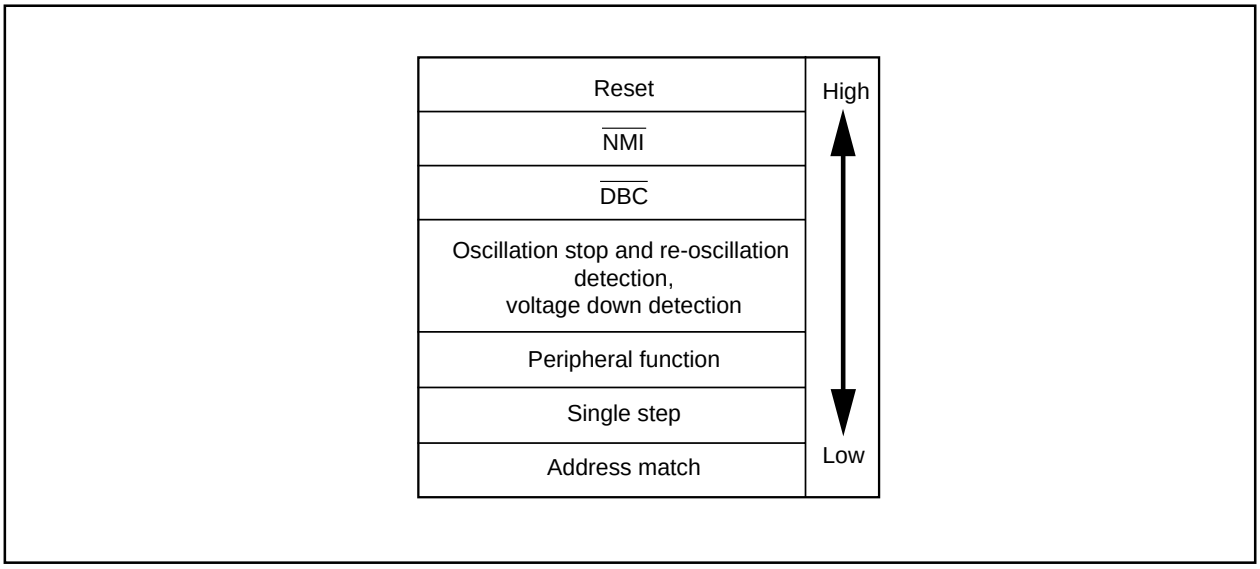



**Figure 9.5.1. Hardware Interrupt Priority**

### 9.5.1 Interrupt Priority Resolution Circuit

The interrupt priority resolution circuit is used to select the interrupt with the highest priority among those requested.

Figure 9.5.1.1 shows the circuit that judges the interrupt priority level.

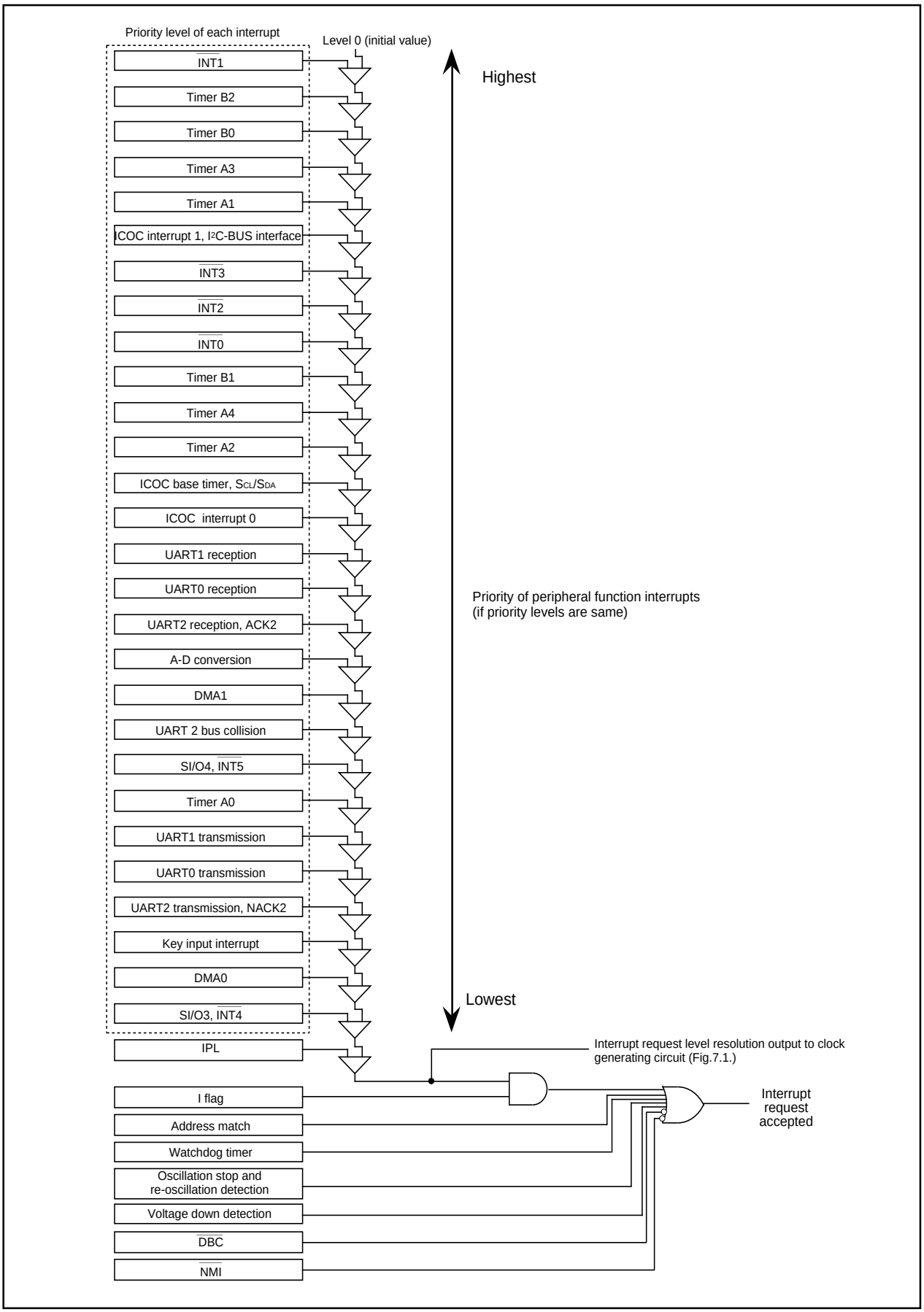



**Figure 9.5.1.1. Interrupts Priority Select Circuit**

## 9.6 $\overline{\text{INT}}$ Interrupt

$\overline{\text{INT}}$i interrupt (i=0 to 5) is triggered by the edges of external inputs. The edge polarity is selected using the IFSR register's IFSRi bit.

To use the $\overline{\text{INT4}}$ interrupt, set the IFSR register's IFSR6 bit to "1" (=$\overline{\text{INT4}}$). To use the $\overline{\text{INT5}}$ interrupt, set the IFSR register's IFSR7 bit to "1" (=$\overline{\text{INT5}}$).

After modifiying the IFSR6 or IFSR7 bit, clear the corresponding IR bit to "0" (=interrupt not requested) before enabling the interrupt.

The $\overline{\text{INT5}}$ input has an effective digital debounce function for a noise rejection. Refer to "**17.6 Digital Debounce function**" for this detail.

Figure 9.6.1 shows the IFSR registers.

Interrupt request cause select register

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

| Symbol | Address | After reset |
|---|---|---|
| IFSR | $035F_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| IFSR0 | INT0 interrupt polarity switching bit | 0 : One edge<br>1 : Both edges (Note 1) | RW |
| IFSR1 | INT1 interrupt polarity switching bit | 0 : One edge<br>1 : Both edges (Note 1) | RW |
| IFSR2 | INT2 interrupt polarity switching bit | 0 : One edge<br>1 : Both edges (Note 1) | RW |
| IFSR3 | INT3 interrupt polarity switching bit | 0 : One edge<br>1 : Both edges (Note 1) | RW |
| IFSR4 | INT4 interrupt polarity switching bit | 0 : One edge<br>1 : Both edges (Note 1) | RW |
| IFSR5 | INT5 interrupt polarity switching bit | 0 : One edge<br>1 : Both edges (Note 1) | RW |
| IFSR6 | Interrupt request cause select bit | 0 : SI/O3 (Note 2)<br>1 : $\overline{\text{INT4}}$ | RW |
| IFSR7 | Interrupt request cause select bit | 0 : SI/O4 (Note 2)<br>1 : $\overline{\text{INT5}}$ | RW |

Note 1: When setting this bit to “1” (= both edges), make sure the INT0IC to INT5IC register’s POL bit is set to “0” (= falling edge).
Note 2: When setting this bit to “0” (= SI/O3, SI/O4), make sure the S3IC and S4IC registers’ POL bit is set to “0” (= falling edge).

**Figure 9.6.1. IFSR Register**

## 9.7 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt request is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low, after the $\overline{\text{NMI}}$ interrupt was enabled by writing a “1” to bit 4 of register PM2. The $\overline{\text{NMI}}$ interrupt is a non-maskable interrupt, once it is enabled.

The input level of this $\overline{\text{NMI}}$ interrupt input pin can be read by accessing the P8 register’s P8_5 bit.

$\overline{\text{NMI}}$ is disabled by default after reset (the pin is a GPIO pin, $P8_5$) and can be enabled using bit 4 of PM2 register. Once enabled, it can only be disabled by a reset signal.

The $\overline{\text{NMI}}$ input has an effective digital debounce function for a noise rejection. Refer to "**17.6 Digital Debounce function**" for this detail.

## 9.8 Key Input Interrupt

Of $P10_4$ to $P10_7$, a key input interrupt is generated when input on any of the $P10_4$ to $P10_7$ pins which has had the PD10 register’s PD10_4 to PD10_7 bits set to “0” (= input) goes low. Key input interrupts can be used as a key-on wakeup function, the function which gets the microcomputer out of wait or stop mode. However, if you intend to use the key input interrupt, do not use $P10_4$ to $P10_7$ as analog input ports. Figure 9.8.1 shows the block diagram of the key input interrupt. Note, however, that while input on any pin which has had the PD10_4 to PD10_7 bits set to “0” (= input mode) is pulled low, inputs on all other pins of the port are not detected as interrupts.

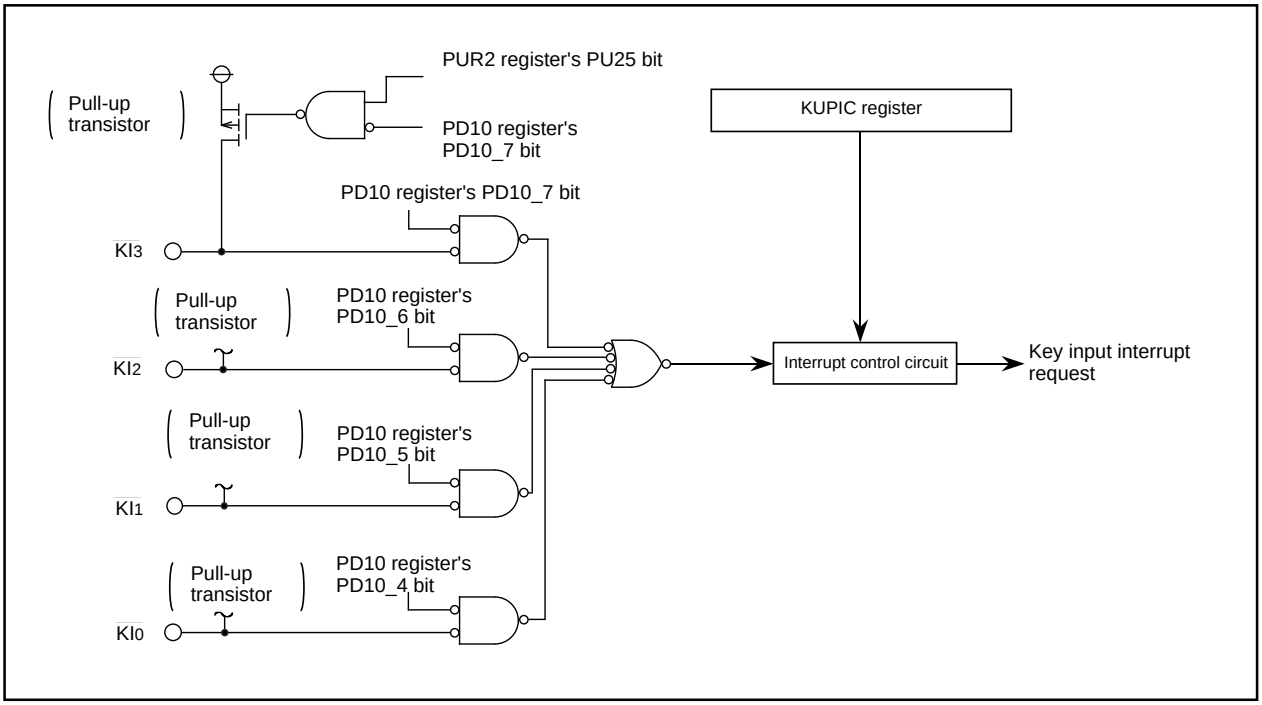



**Figure 9.8.1. Key Input Interrupt**

## 9.9 Address Match Interrupt

An address match interrupt request is generated immediately before executing the instruction at the address indicated by the RMADi register (i=0 to 1). Set the start address of any instruction in the RMADi register. Use the AIER register's AIER0 and AIER1 bits to enable or disable the interrupt. Note that the address match interrupt is unaffected by the I flag and IPL. For address match interrupts, the value of the PC that is saved to the stack area varies depending on the instruction being executed (refer to "Saving Registers").

(The value of the PC that is saved to the stack area is not the correct return address.) Therefore, follow one of the methods described below to return from the address match interrupt.

• Rewrite the content of the stack and then use the REIT instruction to return.

• Restore the stack to its previous state before the interrupt request was accepted by using the POP or similar other instruction and then use a jump instruction to return.

Table 9.9.1 shows the value of the PC that is saved to the stack area when an address match interrupt request is accepted.

Figure 9.9.1 shows the AIER, RMAD0 and RMAD1 registers.

**Table 9.9.1. Value of the PC that is saved to the stack area when an address match interrupt request is accepted.**

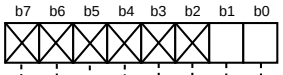
| Instruction at the address indicated by the RMADi register | Value of the PC that is saved to the stack area |
|---|---|
| • 16-bit op-code instruction<br>• Instruction shown below among 8-bit operation code instructions<br>ADD.B:S #IMM8,dest SUB.B:S #IMM8,dest AND.B:S #IMM8,dest<br>OR.B:S #IMM8,dest MOV.B:S #IMM8,dest STZ.B:S #IMM8,dest<br>STNZ.B:S #IMM8,dest STZX.B:S #IMM81,#IMM82,dest<br>CMP.B:S #IMM8,dest PUSHM src POPM dest<br>JMPS #IMM8 JSRS #IMM8<br>MOV.B:S #IMM,dest (However, dest=A0 or A1) | The address indicated by the RMADi register +2 |
| Instructions other than the above | The address indicated by the RMADi register +1 |

Value of the PC that is saved to the stack area : Refer to "Saving Registers".

**Table 9.9.2. Relationship Between Address Match Interrupt Sources and Associated Registers**

| Address match interrupt sources | Address match interrupt enable bit | Address match interrupt register |
|---|---|---|
| Address match interrupt 0 | AIER0 | RMAD0 |
| Address match interrupt 1 | AIER1 | RMAD1 |

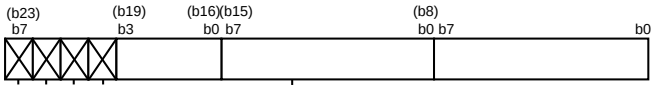
Address match interrupt enable register




| Symbol | Address | After reset |
|---|---|---|
| AIER | $0009_{16}$ | $XXXXXX00_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| AIER0 | Address match interrupt 0 enable bit | 0 : Interrupt disabled<br>1 : Interrupt enabled | RW |
| AIER1 | Address match interrupt 1 enable bit | 0 : Interrupt disabled<br>1 : Interrupt enabled | RW |
| ——<br>(b7-b2) | Nothing is assigned.<br>When write, set to “0”.<br>When read, their contents are indeterminate. | | — |

Address match interrupt register i (i = 0 to 1)




| Symbol | Address | After reset |
|---|---|---|
| RMAD0 | $0012_{16}$ to $0010_{16}$ | $X00000_{16}$ |
| RMAD1 | $0016_{16}$ to $0014_{16}$ | $X00000_{16}$ |

| Function | Setting range | RW |
|---|---|---|
| Address setting register for address match interrupt | $00000_{16}$ to $FFFFF_{16}$ | RW |
| Nothing is assigned.<br>When write, set to “0”.<br>When read, their contents are indeterminate. | | — |

**Figure 9.9.1. AIER Register, RMAD0 and RMAD1 Registers**

# 10. Watchdog Timer

The watchdog timer is the function of detecting when the program is out of control. Therefore, we recommend using the watchdog timer to improve reliability of a system. The watchdog timer contains a 15-bit counter which counts down the clock derived by dividing the CPU clock using the prescaler. Whether to generate a watchdog timer interrupt request or apply a watchdog timer reset as an operation to be performed when the watchdog timer underflows after reaching the terminal count can be selected using the PM12 bit of PM1 register. The PM12 bit can only be set to “1” (reset). Once this bit is set to “1”, it cannot be set to “0” (watchdog timer interrupt) in a program. Refer to “5.3 Watchdog Timer Reset” for the details of watchdog timer reset.
When the main clock source is selected for CPU clock, ring oscillator clock, PLL clock,the WDC register's the WDC7 bit value for prescaler can be chosen to be 16 or 128. If a sub-clock is selected for CPU clock, the prescaler is always 2 no matter how the WDC7 bit is set. The period of watchdog timer can be calculated as given below. The period of watchdog timer is, however, subject to an error due to the prescaler.

With main clock source chosen for CPU clock, ring oscillator clock, PLL clock

$$\text{Watchdog timer period} = \frac{\text{Prescaler dividing (16 or 128) X Watchdog timer count (32768)}}{\text{CPU clock}}$$

With sub-clock chosen for CPU clock

$$\text{Watchdog timer period} = \frac{\text{Prescaler dividing (2) X Watchdog timer count (32768)}}{\text{CPU clock}}$$

For example, when CPU clock = 16 MHz and the divide-by-N value for the prescaler= 16, the watchdog timer period is approx. 32.8 ms.

The watchdog timer is initialized by writing to the WDTS register. The prescaler is initialized after reset. Note that the watchdog timer and the prescaler both are inactive after reset, so that the watchdog timer is activated to start counting by writing to the WDTS register.
In stop mode, wait mode and hold state, the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes or state are released.
Figure 10.1 shows the block diagram of the watchdog timer. Figure 10.2 shows the watchdog timer-related registers.

• Count source protective mode
In this mode, a ring oscillator clock is used for the watchdog timer count source. The watchdog timer can be kept being clocked even when CPU clock stops as a result of run-away.
Before this mode can be used, the following register settings are required:
(1) Set the PRC1 bit of PRCR register to “1” (enable writes to PM1 and PM2 registers).
(2) Set the PM12 bit of PM1 register to “1” (reset when the watchdog timer underflows).
(3) Set the PM22 bit of PM2 register to “1” (ring oscillator clock used for the watchdog timer count source).
(4) Set the PRC1 bit of PRCR register to “0” (disable writes to PM1 and PM2 registers).
(5) Write to the WDTS register (watchdog timer starts counting).

Setting the PM22 bit to “1” results in the following conditions

• The ring oscillator starts oscillating, and the ring oscillator clock becomes the watchdog timer count source.

$$\text{Watchdog timer period} = \frac{\text{Watchdog timer count (32768)}}{\text{ring oscillator clock}}$$

• The CM10 bit of CM1 register is disabled against write. (Writing a “1” has no effect, nor is stop mode entered.)

• The watchdog timer does not stop when in wait mode.

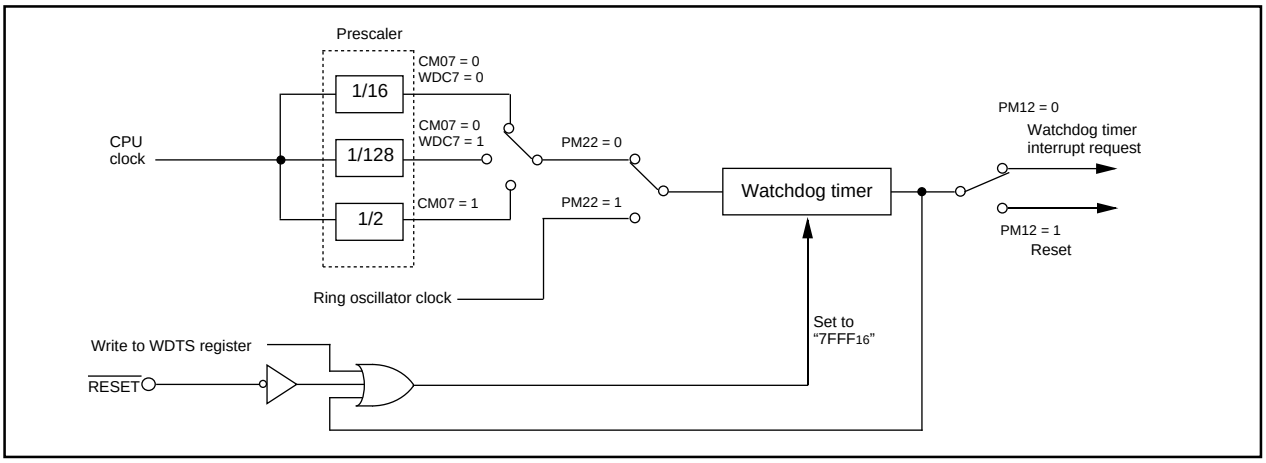



**Figure 10.1. Watchdog Timer Block Diagram**

Watchdog timer control register

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

Symbol: WDC — Address: 000F16 — After reset: 00XXXXX2(Note 2)

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| $\overline{\text{(b4-b0)}}$ | High-order bit of watchdog timer | | RO |
| WDC5 | Cold start / warm start discrimination flag (Note 1, 2) | 0 : Cold start<br>1 : Warm start | RW |
| $\overline{\text{(b6)}}$ | Reserved bit | Must set to “0” | RW |
| WDC7 | Prescaler select bit | 0 : Divided by 16<br>1 : Divided by 128 | RW |

Note 1: Writing to the WDC register causes the WDC5 bit to be set to “1” (warm start).
Note 2: The WDC5 bit is “0” (cold start) immediately after power-on. It can only be set to “1” in a program. It is set to “0” when the input voltage at the VCC pin drops to Vdet2 or less while the VC25 bit in the VCR2 register is set to “1” (RAM retention limit detection circuit enable).

Watchdog timer start register (Note)

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

Symbol: WDTS — Address: 000E16 — After reset: Indeterminate

| Function | RW |
|---|---|
| The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to “7FFF16” regardless of whatever value is written. | WO |

Note : Write to the WDTS register after the watchdog timer interrupt occurs.

**Figure 10.2. WDC Register and WDTS Register**

# 11. DMAC

The DMAC (Direct Memory Access Controller) allows data to be transferred without the CPU intervention. Two DMAC channels are included. Each time a DMA request occurs, the DMAC transfers one (8 or 16-bit) data from the source address to the destination address. The DMAC uses the same data bus as used by the CPU. Because the DMAC has higher priority of bus control than the CPU and because it makes use of a cycle steal method, it can transfer one word (16 bits) or one byte (8 bits) of data within a very short time after a DMA request is generated. Figure 11.1 shows the block diagram of the DMAC. Table 11.1 shows the DMAC specifications. Figures 11.2 to 11.4 show the DMAC-related registers.

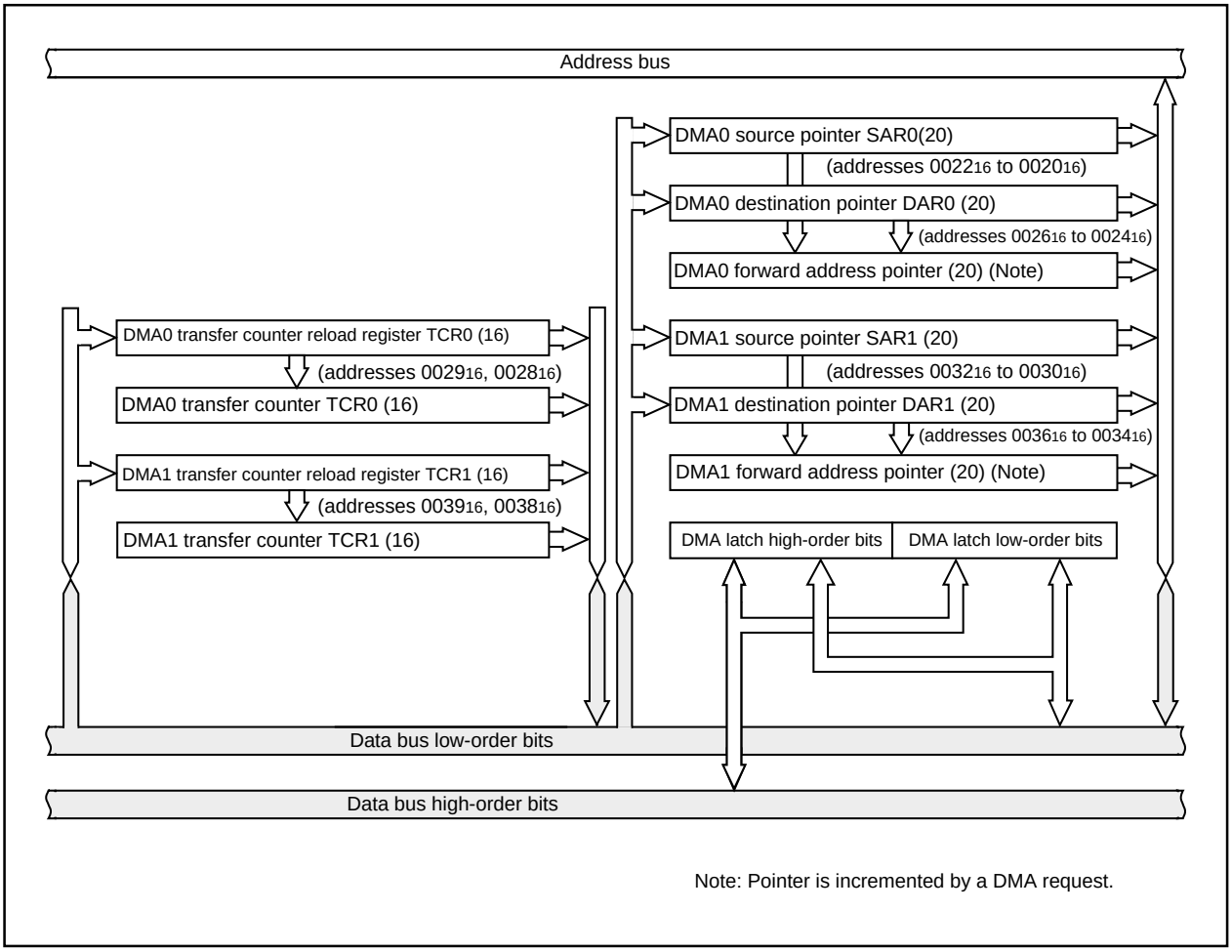



**Figure 11.1 DMAC Block Diagram**

A DMA request is generated by a write to the DMiSL register (i = 0,1)'s DSR bit, as well as by an interrupt request which is generated by any function specified by the DMiSL register's DMS and DSEL3 to DSEL0 bits. However, unlike in the case of interrupt requests, DMA requests are not affected by the I flag and the interrupt control register, so that even when interrupt requests are disabled and no interrupt request can be accepted, DMA requests are always accepted. Furthermore, because the DMAC does not affect interrupts, the interrupt control register's IR bit does not change state due to a DMA transfer.

A data transfer is initiated each time a DMA request is generated when the DMiCON register's DMAE bit = "1" (DMA enabled). However, if the cycle in which a DMA request is generated is faster than the DMA transfer cycle, the number of transfer requests generated and the number of times data is transferred may not match. For details, refer to "DMA Requests".

**Table 11.1 DMAC Specifications**

| Item | | Specification |
|---|---|---|
| No. of channels | | 2 (cycle steal method) |
| Transfer memory space | | • From any address in the 1M bytes space to a fixed address<br>• From a fixed address to any address in the 1M bytes space<br>• From a fixed address to a fixed address |
| Maximum No. of bytes transferred | | 128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers) |
| DMA request factors (Note 1, Note 2) | | Falling edge of $\overline{INT0}$ or $\overline{INT1}$<br>Both edge of $\overline{INT0}$ or $\overline{INT1}$<br>Timer A0 to timer A4 interrupt requests<br>Timer B0 to timer B2 interrupt requests<br>UART0 transfer, UART0 reception interrupt requests<br>UART1 transfer, UART1 reception interrupt requests<br>UART2 transfer, UART2 reception interrupt requests<br>SI/O3, SI/O4 interrupt requests<br>A-D conversion interrupt requests<br>Timer S(ICOC) requests<br>Software triggers |
| Channel priority | | DMA0 > DMA1 (DMA0 takes precedence) |
| Transfer unit | | 8 bits or 16 bits |
| Transfer address direction | | forward or fixed (The source and destination addresses cannot both be in the forward direction.) |
| Transfer mode | Single transfer | Transfer is completed when the DMAi transfer counter (i = 0,1) underflows after reaching the terminal count. |
| | Repeat transfer | When the DMAi transfer counter underflows, it is reloaded with the value of the DMAi transfer counter reload register and a DMA transfer is con tinued with it. |
| DMA interrupt request generation timing | | When the DMAi transfer counter underflowed |
| DMA startup | | Data transfer is initiated each time a DMA request is generated when the DMAiCON register's DMAE bit = "1" (enabled). |
| DMA shutdown | Single transfer | • When the DMAE bit is set to "0" (disabled)<br>• After the DMAi transfer counter underflows |
| | Repeat transfer | When the DMAE bit is set to "0" (disabled) |
| Reload timing for forward address pointer and transfer counter | | When a data transfer is started after setting the DMAE bit to "1" (en abled), the forward address pointer is reloaded with the value of the SARi or the DARi pointer whichever is specified to be in the forward direction and the DMAi transfer counter is reloaded with the value of the DMAi transfer counter reload register. |

Notes:

1. DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the I flag nor by the interrupt control register.
2. The selectable causes of DMA requests differ with each channel.
3. Make sure that no DMAC-related registers (addresses $0020_{16}$ to $003F_{16}$) are accessed by the DMAC.

DMA0 request cause select register

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

| Symbol | Address | After reset |
|---|---|---|
| DM0SL | $03B8_{16}$ | $00_{16}$ |

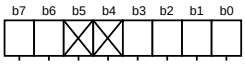
| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| DSEL0 | DMA request cause select bit | Refer to note | RW |
| DSEL1 | | | RW |
| DSEL2 | | | RW |
| DSEL3 | | | RW |
| — (b5-b4) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |
| DMS | DMA request cause expansion select bit | 0: Basic cause of request<br>1: Extended cause of request | RW |
| DSR | Software DMA request bit | A DMA request is generated by setting this bit to “1” when the DMS bit is “0” (basic cause) and the DSEL3 to DSEL0 bits are “$0001_2$” (software trigger).<br>The value of this bit when read is “0” . | RW |

Note: The causes of DMA0 requests can be selected by a combination of DMS bit and DSEL3 to DSEL0 bits in the manner described below.

| DSEL3 to DSEL0 | DMS=0(basic cause of request) | DMS=1(extended cause of request) |
|---|---|---|
| $0\,0\,0\,0_2$ | Falling edge of INT0 pin | ICOC base timer |
| $0\,0\,0\,1_2$ | Software trigger | – |
| $0\,0\,1\,0_2$ | Timer A0 | ICOC channel 0 |
| $0\,0\,1\,1_2$ | Timer A1 | ICOC channel 1 |
| $0\,1\,0\,0_2$ | Timer A2 | – |
| $0\,1\,0\,1_2$ | Timer A3 | – |
| $0\,1\,1\,0_2$ | Timer A4 | Two edges of INT0 pin |
| $0\,1\,1\,1_2$ | Timer B0 | – |
| $1\,0\,0\,0_2$ | Timer B1 | – |
| $1\,0\,0\,1_2$ | Timer B2 | – |
| $1\,0\,1\,0_2$ | UART0 transmit | ICOC channel 2 |
| $1\,0\,1\,1_2$ | UART0 receive | ICOC channel 3 |
| $1\,1\,0\,0_2$ | UART2 transmit | ICOC channel 4 |
| $1\,1\,0\,1_2$ | UART2 receive | ICOC channel 5 |
| $1\,1\,1\,0_2$ | A-D conversion | ICOC channel 6 |
| $1\,1\,1\,1_2$ | UART1 transmit | ICOC channel 7 |

**Figure 11.2 DM0SL Register**

DMA1 request cause select register



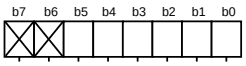

| Symbol | Address | After reset |
|---|---|---|
| DM1SL | 03BA16 | 0016 |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| DSEL0 | DMA request cause select bit | Refer to note | RW |
| DSEL1 | | | RW |
| DSEL2 | | | RW |
| DSEL3 | | | RW |
| —— (b5-b4) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | —— |
| DMS | DMA request cause expansion select bit | 0: Basic cause of request<br>1: Extended cause of request | RW |
| DSR | Software DMA request bit | A DMA request is generated by setting this bit to “1” when the DMS bit is “0” (basic cause) and the DSEL3 to DSEL0 bits are “00012” (software trigger).<br>The value of this bit when read is “0” . | RW |

Note: The causes of DMA1 requests can be selected by a combination of DMS bit and DSEL3 to DSEL0 bits in the manner described below.

| DSEL3 to DSEL0 | DMS=0(basic cause of request) | DMS=1(extended cause of request) |
|---|---|---|
| 0 0 0 02 | Falling edge of INT1 pin | ICOC base timer |
| 0 0 0 12 | Software trigger | – |
| 0 0 1 02 | Timer A0 | ICOC channel 0 |
| 0 0 1 12 | Timer A1 | ICOC channel 1 |
| 0 1 0 02 | Timer A2 | – |
| 0 1 0 12 | Timer A3 | SI/O3 |
| 0 1 1 02 | Timer A4 | SI/O4 |
| 0 1 1 12 | Timer B0 | Two edges of INT1 |
| 1 0 0 02 | Timer B1 | – |
| 1 0 0 12 | Timer B2 | – |
| 1 0 1 02 | UART0 transmit | ICOC channel 2 |
| 1 0 1 12 | UART0 receive | ICOC channel 3 |
| 1 1 0 02 | UART2 transmit | ICOC channel 4 |
| 1 1 0 12 | UART2 receive/ACK2 | ICOC channel 5 |
| 1 1 1 02 | A-D conversion | ICOC channel 6 |
| 1 1 1 12 | UART1 receive | ICOC channel 7 |

DMAi control register(i=0,1)



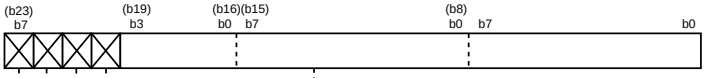

| Symbol | Address | After reset |
|---|---|---|
| DM0CON | 002C16 | 00000X002 |
| DM1CON | 003C16 | 00000X002 |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| DMBIT | Transfer unit bit select bit | 0 : 16 bits<br>1 : 8 bits | RW |
| DMASL | Repeat transfer mode select bit | 0 : Single transfer<br>1 : Repeat transfer | RW |
| DMAS | DMA request bit | 0 : DMA not requested<br>1 : DMA requested | RW<br>(Note 1) |
| DMAE | DMA enable bit | 0 : Disabled<br>1 : Enabled | RW |
| DSD | Source address direction select bit (Note 2) | 0 : Fixed<br>1 : Forward | RW |
| DAD | Destination address direction select bit (Note 2) | 0 : Fixed<br>1 : Forward | RW |
| —— (b7-b6) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | —— |

Note 1: The DMAS bit can be set to “0” by writing “0” in a program (This bit remains unchanged even if “1” is written).
Note 2: At least one of the DAD and DSD bits must be “0” (address direction fixed).

**Figure 11.3 DM1SL Register, DM0CON Register, and DM1CON Registers**

DMAi source pointer (i = 0, 1) (Note)

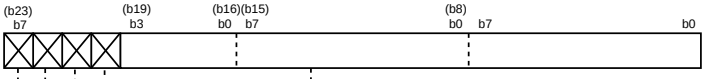



| Symbol | Address | After reset |
|---|---|---|
| SAR0 | $0022_{16}$ to $0020_{16}$ | Indeterminate |
| SAR1 | $0032_{16}$ to $0030_{16}$ | Indeterminate |

| Function | Setting range | RW |
|---|---|---|
| Set the source address of transfer | $00000_{16}$ to $FFFFF_{16}$ | RW |
| Nothing is assigned. When write, set "0". When read, these contents are "0". | | — |

Note: If the DSD bit of DMiCON register is "0" (fixed), this register can only be written to when the DMAE bit of DMiCON register is "0" (DMA disabled).
If the DSD bit is "1" (forward direction), this register can be written to at any time.
If the DSD bit is "1" and the DMAE bit is "1" (DMA enabled), the DMAi forward address pointer can be read from this register. Otherwise, the value written to it can be read.

DMAi destination pointer (i = 0, 1)(Note)

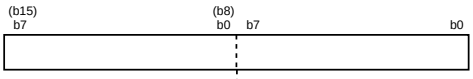



| Symbol | Address | After reset |
|---|---|---|
| DAR0 | $0026_{16}$ to $0024_{16}$ | Indeterminate |
| DAR1 | $0036_{16}$ to $0034_{16}$ | Indeterminate |

| Function | Setting range | RW |
|---|---|---|
| Set the destination address of transfer | $00000_{16}$ to $FFFFF_{16}$ | RW |
| Nothing is assigned. When write, set "0". When read, these contents are "0". | | — |

Note: If the DAD bit of DMiCON register is "0" (fixed), this register can only be written to when the DMAE bit of DMiCON register is "0"(DMA disabled).
If the DAD bit is "1" (forward direction), this register can be written to at any time.
If the DAD bit is "1" and the DMAE bit is "1" (DMA enabled), the DMAi forward address pointer can be read from this register. Otherwise, the value written to it can be read.

DMAi transfer counter (i = 0, 1)




| Symbol | Address | After reset |
|---|---|---|
| TCR0 | $0029_{16}$, $0028_{16}$ | Indeterminate |
| TCR1 | $0039_{16}$, $0038_{16}$ | Indeterminate |

| Function | Setting range | RW |
|---|---|---|
| Set the transfer count minus 1. The written value is stored in the DMAi transfer counter reload register, and when the DMAE bit of DMiCON register is set to "1" (DMA enabled) or the DMAi transfer counter underflows when the DMASL bit of DMiCON register is "1" (repeat transfer), the value of the DMAi transfer counter reload register is transferred to the DMAi transfer counter. When read, the DMAi transfer counter is read. | $0000_{16}$ to $FFFF_{16}$ | RW |

**Figure 11.4 SAR0, SAR1, DAR0, DAR1, TCR0, and TCR1 Registers**

## 11.1 Transfer Cycles

The transfer cycle consists of a memory or SFR read (source read) bus cycle and a write (destination write) bus cycle. The number of read and write bus cycles is affected by the source and destination addresses of transfer. Furthermore, the bus cycle itself is extended by a software wait.

### 11.1.1 Effect of Source and Destination Addresses

If the transfer unit is 16 bits and the source address of transfer begins with an odd address, the source read cycle consists of one more bus cycle than when the source address of transfer begins with an even address.

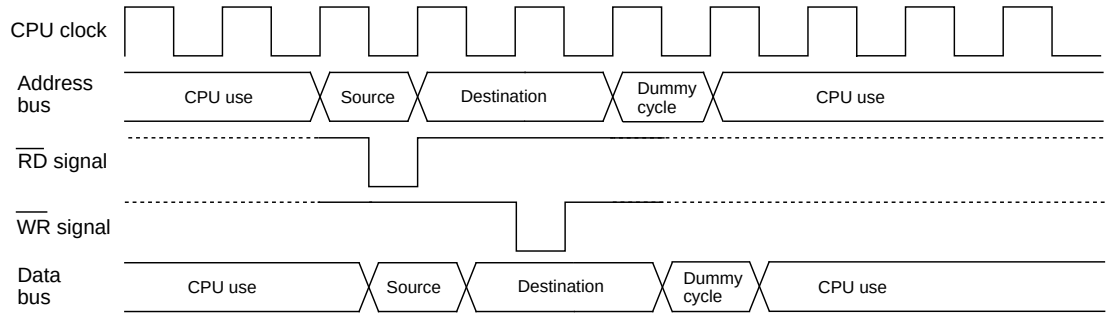
Similarly, if the transfer unit is 16 bits and the destination address of transfer begins with an odd address, the destination write cycle consists of one more bus cycle than when the destination address of transfer begins with an even address.

### 11.1.2 Effect of Software Wait

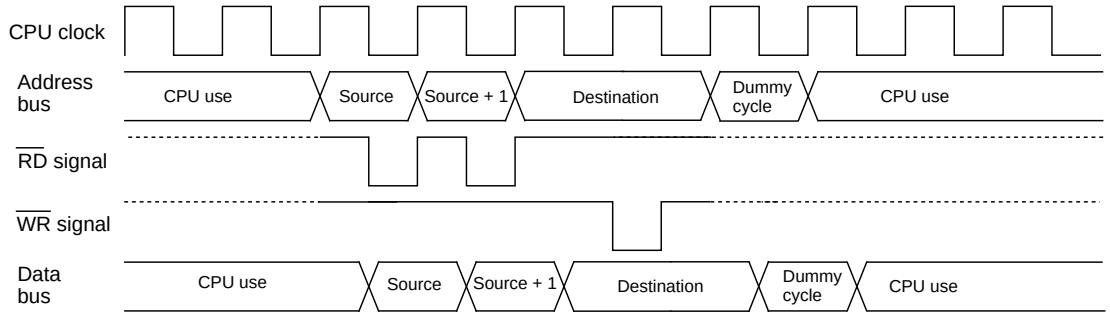
For memory or SFR accesses in which one or more software wait states are inserted, the number of bus cycles required for that access increases by an amount equal to software wait states.

Figure 11.1.1 shows the example of the cycles for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating transfer cycles, take into consideration each condition for the source read and the destination write cycle, respectively. For example, when data is transferred in 16 bit units and when both the source address and destination address are an odd address ((2) in Figure 11.1.1), two source read bus cycles and two destination write bus cycles are required.

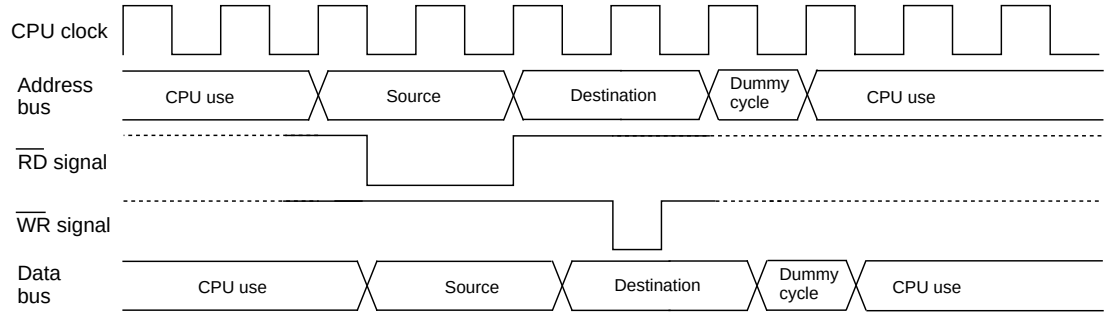
(1) When the transfer unit is 8 or 16 bits and the source of transfer is an even address



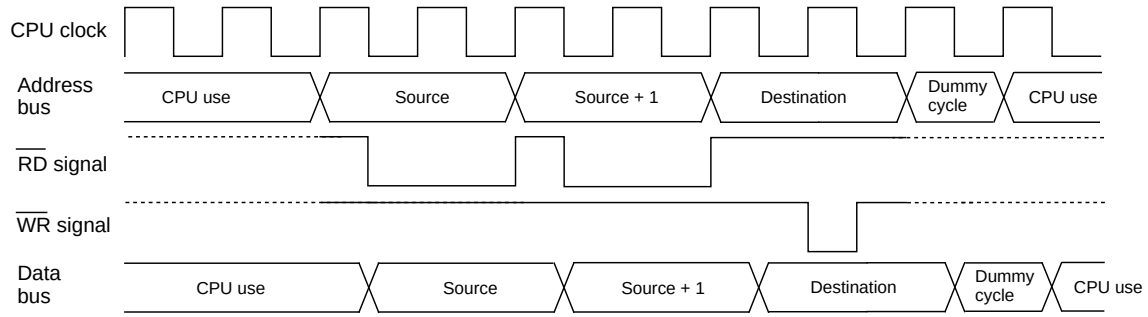

(2) When the transfer unit is 16 bits and the source address of transfer is an odd address, or when the transfer unit is 16 bits and an 8-bit bus is used




(3) When the source read cycle under condition (1) has one wait state inserted




(4) When the source read cycle under condition (2) has one wait state inserted




Note: The same timing changes occur with the respective conditions at the destination as at the source.

**Figure 11.1.1 Transfer Cycles for Source Read**

## 11.2. DMA Transfer Cycles

Any combination of even or odd transfer read and write adresses is possible. Table 11.2.1 shows the number of DMA transfer cycles. Table 11.2.2 shows the Coefficient j, k.
The number of DMAC transfer cycles can be calculated as follows:

No. of transfer cycles per transfer unit = No. of read cycles x j + No. of write cycles x k

**Table 11.2.1 DMA Transfer Cycles**

| Transfer unit | Access address | No. of read cycles | No. of write cycles |
|---|---|---|---|
| 8-bit transfers (DMBIT= “1”) | Even | 1 | 1 |
| | Odd | 1 | 1 |
| 16-bit transfers (DMBIT= “0”) | Even | 1 | 1 |
| | Odd | 2 | 2 |

**Table 11.2.2 Coefficient j, k**

| | Internal area | | | |
|---|---|---|---|---|
| | Internal ROM, RAM | | SFR | |
| | No wait | With wait | 1 wait (Note) | 2 wait (Note) |
| j | 1 | 2 | 2 | 3 |
| k | 1 | 2 | 2 | 3 |

Note : Depends on the set value of PM20 bit in PM2 register

## 11.3 DMA Enable

When a data transfer starts after setting the DMAE bit in DMiCON register (i = 0, 1) to "1" (enabled), the DMAC operates as follows:

(1) Reload the forward address pointer with the SARi register value when the DSD bit in DMiCON register is "1" (forward) or the DARi register value when the DAD bit of DMiCON register is "1" (forward).

(2) Reload the DMAi transfer counter with the DMAi transfer counter reload register value.

If the DMAE bit is set to "1" again while it remains set, the DMAC performs the above operation. However, if a DMA request may occur simultaneously when the DMAE bit is being written, follow the steps below.

Step 1: Write "1" to the DMAE bit and DMAS bit in DMiCON register simultaneously.

Step 2: Make sure that the DMAi is in an initial state as described above (1) and (2) in a program.

If the DMAi is not in an initial state, the above steps should be repeated.

## 11.4 DMA Request

The DMAC can generate a DMA request as triggered by the cause of request that is selected with the DMS and DSEL3 to DSEL0 bits of DMiSL register (i = 0, 1) on either channel. Table 11.4.1 shows the timing at which the DMAS bit changes state.

Whenever a DMA request is generated, the DMAS bit is set to "1" (DMA requested) regardless of whether or not the DMAE bit is set. If the DMAE bit was set to "1" (enabled) when this occurred, the DMAS bit is set to "0" (DMA not requested) immediately before a data transfer starts. This bit cannot be set to "1" in a program (it can only be set to "0").

The DMAS bit may be set to "1" when the DMS or the DSEL3 to DSEL0 bits change state. Therefore, always be sure to set the DMAS bit to "0" after changing the DMS or the DSEL3 to DSEL0 bits.

Because if the DMAE bit is "1", a data transfer starts immediately after a DMA request is generated, the DMAS bit in almost all cases is "0" when read in a program. Read the DMAE bit to determine whether the DMAC is enabled.

**Table 11.4.1 Timing at Which the DMAS Bit Changes State**

| DMA factor | DMAS bit of the DMiCON register | |
|---|---|---|
| | Timing at which the bit is set to "1" | Timing at which the bit is set to "0" |
| Software trigger | When the DSR bit of DMiSL register is set to "1" | • Immediately before a data transfer starts<br>• When set by writing "0" in a program |
| Peripheral function | When the interrupt control register for the peripheral function that is selected by the DSEL3 to DSEL0 and DMS bits of DMiSL register has its IR bit set to "1" | |

## 11.5 Channel Priority and DMA Transfer Timing

If both DMA0 and DMA1 are enabled and DMA transfer request signals from DMA0 and DMA1 are detected active in the same sampling period (one period from a falling edge to the next falling edge of CPU clock), the DMAS bit on each channel is set to “1” (DMA requested) at the same time. In this case, the DMA requests are arbitrated according to the channel priority, DMA0 > DMA1. The following describes DMAC operation when DMA0 and DMA1 requests are detected active in the same sampling period. Figure 11.5.1 shows an example of DMA transfer effected by external factors.

DMA0 request having priority is received first to start a transfer when a DMA0 request and DMA1 request are generated simultanelously. After one DMA0 transfer is completed, a bus arbitration is returned to the CPU. When the CPU has completed one bus access, a DMA1 transfer starts. After one DMA1 transfer is completed, the bus arbitration is again returned to the CPU.

In addition, DMA requsts cannot be counted up since each channel has one DMAS bit. Therefore, when DMA requests, as DMA1 in Figure 11.5.1, occurs more than one time, the DAMS bit is set to "0" as soon as getting the bus arbitration. The bus arbitration is returned to the CPU when one transfer is completed.

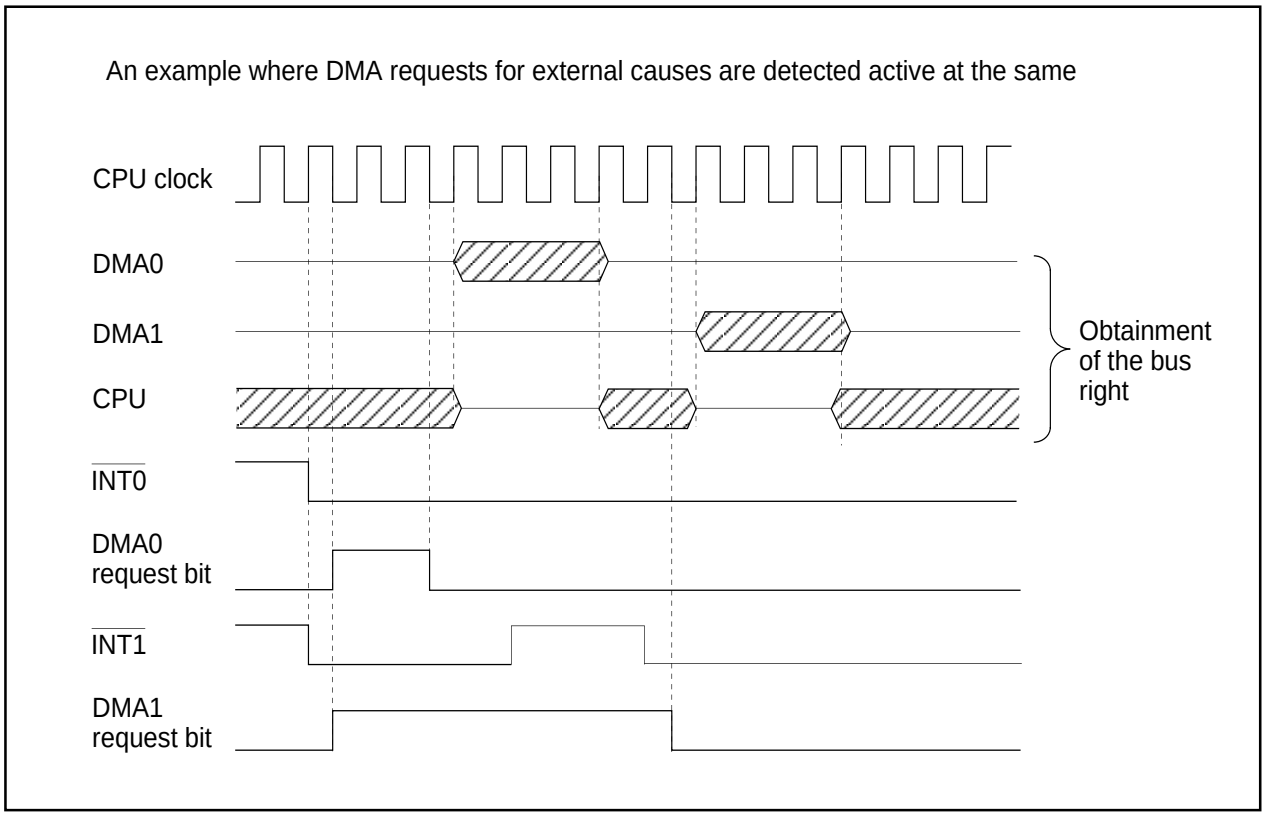



**Figure 11.5.1 DMA Transfer by External Factors**

# 12. Timers

Eight 16-bit timers, each capable of operating independently of the others, can be classified by function as either timer A (five) and timer B (three). The count source for each timer acts as a clock, to control such timer operations as counting, reloading, etc. Figures 12.1 and 12.2 show block diagrams of timer A and timer B configuration, respectively.

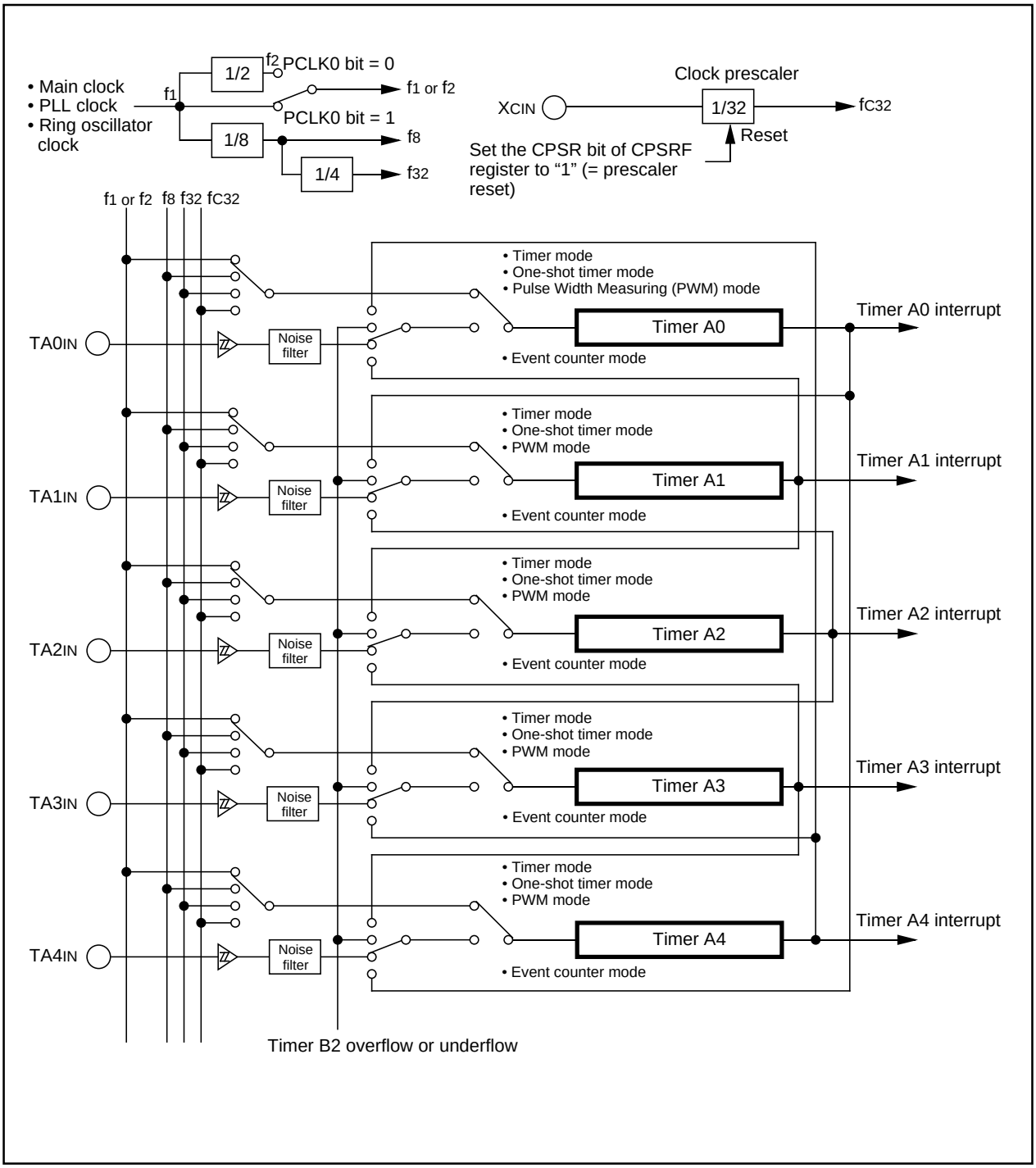



**Figure 12.1. Timer A Configuration**

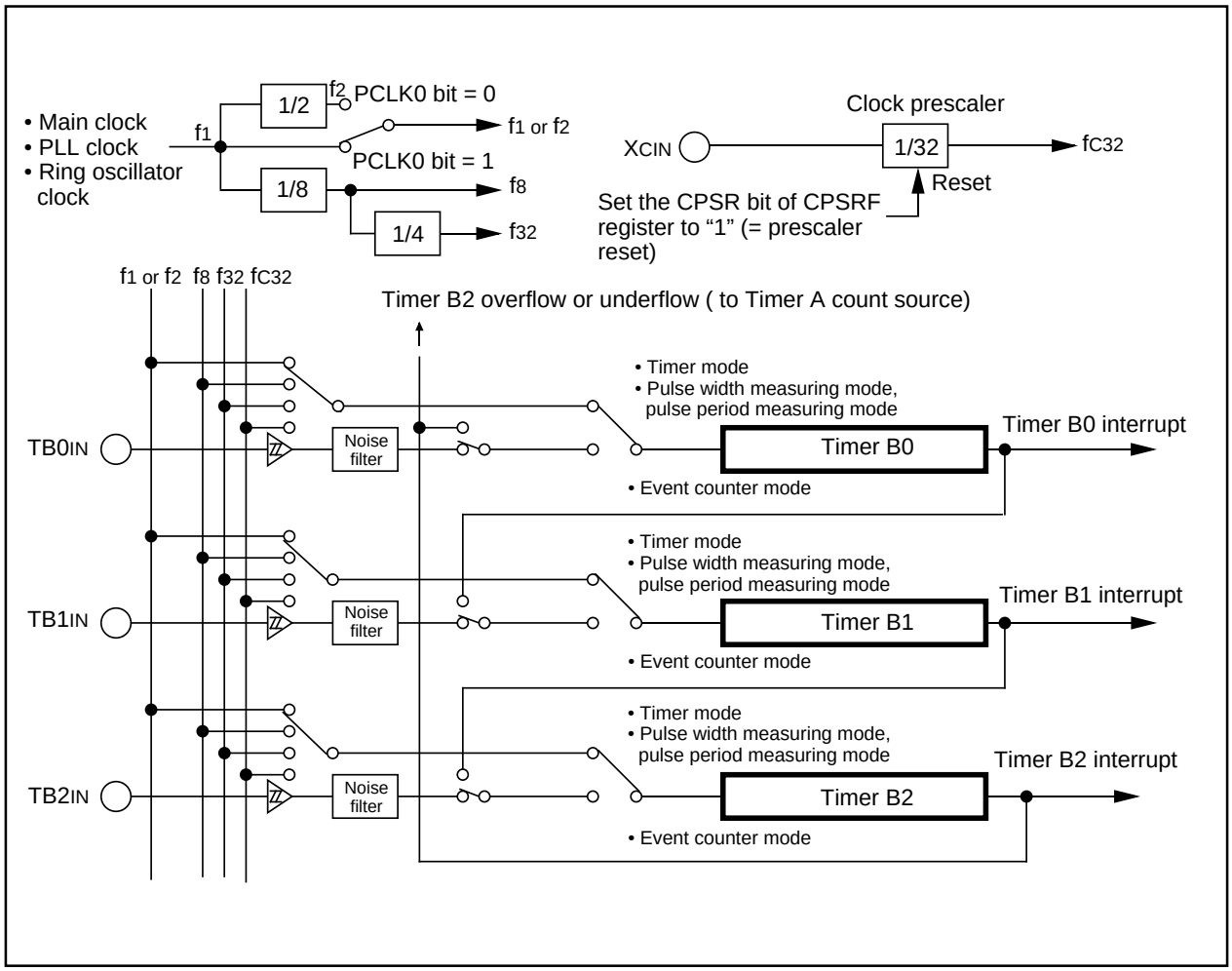



**Figure 12.2. Timer B Configuration**

## 12.1 Timer A

Figure 12.1.1 shows a block diagram of the timer A. Figures 12.1.2 to 12.1.4 show registers related to the timer A.

The timer A supports the following four modes. Except in event counter mode, timers A0 to A4 all have the same function. Use the TMOD1 to TMOD0 bits of TAiMR register (i = 0 to 4) to select the desired mode.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external device or overflows and underflows of other timers.
- One-shot timer mode: The timer outputs a pulse only once before it reaches the minimum count "$0000_{16}$."
- Pulse width modulation (PWM) mode: The timer outputs pulses in a given width successively.

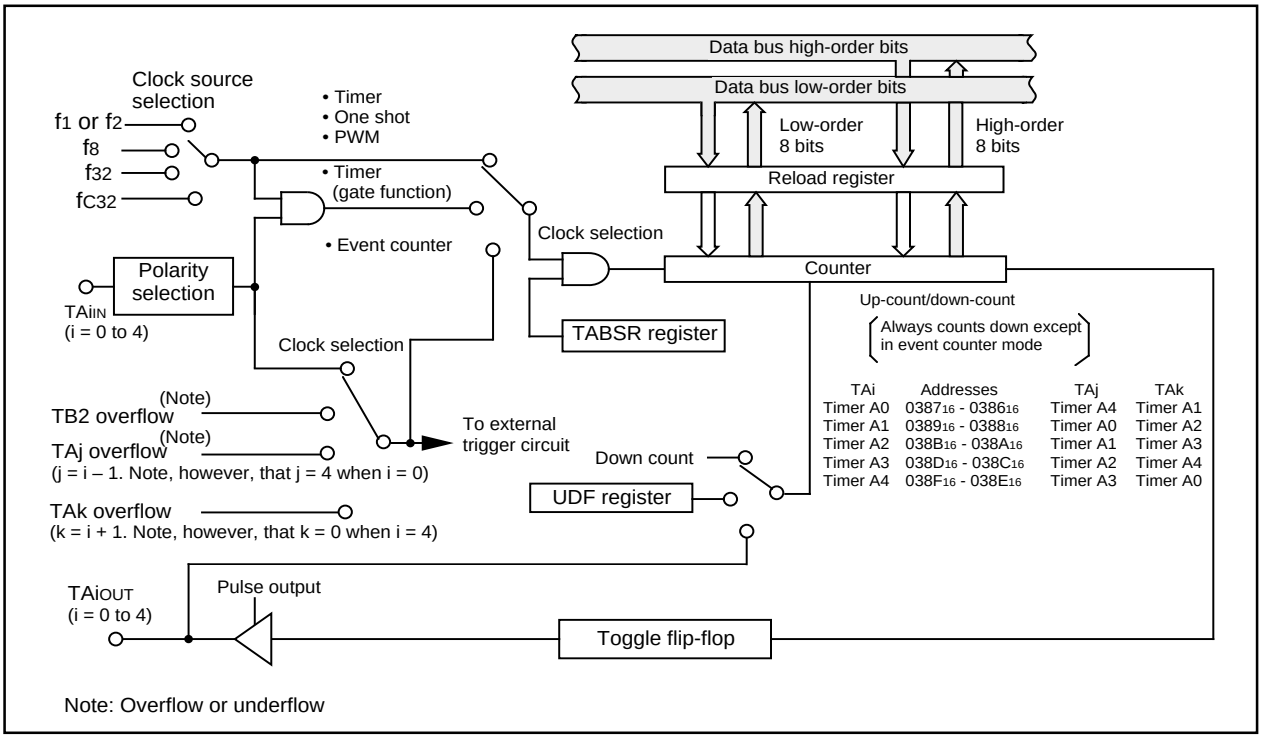



**Figure 12.1.1. Timer A Block Diagram**

Timer Ai mode register (i=0 to 4)

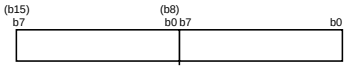
| Symbol | Address | After reset |
|---|---|---|
| TA0MR to TA4MR | $0396_{16}$ to $039A_{16}$ | $00_{16}$ |

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

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation 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 (PWM) mode | RW |
| TMOD1 | | | RW |
| MR0 | ——— | Function varies with each operation mode | RW |
| MR1 | | | RW |
| MR2 | | | RW |
| MR3 | | | RW |
| TCK0 | Count source select bit | Function varies with each operation mode | RW |
| TCK1 | | | RW |

**Figure 12.1.2. TA0MR to TA4MR Registers**

Timer Ai register (i= 0 to 4) (Note 1)

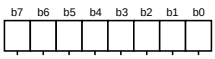



| Symbol | Address | After reset |
|---|---|---|
| TA0 | $0387_{16}$, $0386_{16}$ | Indeterminate |
| TA1 | $0389_{16}$, $0388_{16}$ | Indeterminate |
| TA2 | $038B_{16}$, $038A_{16}$ | Indeterminate |
| TA3 | $038D_{16}$, $038C_{16}$ | Indeterminate |
| TA4 | $038F_{16}$, $038E_{16}$ | Indeterminate |

| Mode | Function | Setting range | RW |
|---|---|---|---|
| Timer mode | Divide the count source by n + 1 where n = set value | $0000_{16}$ to $FFFF_{16}$ | RW |
| Event counter mode | Divide the count source by $FFFF_{16}$ – n + 1 where n = set value when counting up or by n + 1 when counting down (Note 5) | $0000_{16}$ to $FFFF_{16}$ | RW |
| One-shot timer mode | Divide the count source by n where n = set value and cause the timer to stop | $0000_{16}$ to $FFFF_{16}$ (Notes 2, 4) | WO |
| Pulse width modulation mode (16-bit PWM) | Modify the pulse width as follows:<br>PWM period: $(2^{16} – 1)$ / fj<br>High level PWM pulse width: n / fj<br>where n = set value, fj = count source frequency | $0000_{16}$ to $FFFE_{16}$ (Note 3, 4) | WO |
| Pulse width modulation mode (8-bit PWM) | Modify the pulse width as follows:<br>PWM period: $(2^8 – 1)$ x (m + 1)/ fj<br>High level PWM pulse width: (m + 1)n / fj<br>where n = high-order address set value, m = low-order address set value, fj = count source frequency | $00_{16}$ to $FE_{16}$ (High-order address)<br>$00_{16}$ to $FF_{16}$ (Low-order address)<br>(Note 3, 4) | WO |

Note 1: The register must be accessed in 16 bit units.
Note 2: If the TAi register is set to '$0000_{16}$,' the counter does not work and timer Ai interrupt requests are not generated either. Furthermore, if "pulse output" is selected, no pulses are output from the TAiOUT pin.
Note 3: If the TAi register is set to '$0000_{16}$,' the pulse width modulator does not work, the output level on the TAiOUT pin remains low, and timer Ai interrupt requests are not generated either. The same applies when the 8 high-order bits of the timer TAi register are set to '001 6' while operating as an 8-bit pulse width modulator.
Note 4: Use the MOV instruction to write to the TAi register.
Note 5: The timer counts pulses from an external device or overflows or underflows in other timers.

Count start flag




| Symbol | Address | After reset |
|---|---|---|
| TABSR | $0380_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TA0S | Timer A0 count start flag | 0 : Stops counting<br>1 : Starts counting | RW |
| TA1S | Timer A1 count start flag | | RW |
| TA2S | Timer A2 count start flag | | RW |
| TA3S | Timer A3 count start flag | | RW |
| TA4S | Timer A4 count start flag | | RW |
| TB0S | Timer B0 count start flag | | RW |
| TB1S | Timer B1 count start flag | | RW |
| TB2S | Timer B2 count start flag | | RW |

Up/down flag (Note 1)

| Symbol | Address | After reset |
|---|---|---|
| UDF | $0384_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TA0UD | Timer A0 up/down flag | 0 : Down count<br>1 : Up count<br>Enabled by setting the TAiMR register's MR2 bit to "0" (= switching source in UDF register) during event counter mode. | RW |
| TA1UD | Timer A1 up/down flag | | RW |
| TA2UD | Timer A2 up/down flag | | RW |
| TA3UD | Timer A3 up/down flag | | RW |
| TA4UD | Timer A4 up/down flag | | 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<br>(Notes 2, 3) | WO |
| TA3P | Timer A3 two-phase pulse signal processing select bit | | WO |
| TA4P | Timer A4 two-phase pulse signal processing select bit | | WO |

Note 1: Use MOV instruction to write to this register.
Note 2: Make sure the port direction bits for the TA2IN to TA4IN and TA2OUT to TA4OUT pins are set to "0" (input mode).
Note 3: When not using the two-phase pulse signal processing function, set the corresponding bit to "0".

**Figure 12.1.3. TA0 to TA4 Registers, TABSR Register, and UDF Register**

### One-shot start flag

| Symbol | Address | After reset |
|---|---|---|
| ONSF | $0382_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TA0OS | Timer A0 one-shot start flag | The timer starts counting by setting this bit to "1" while the TMOD1 to TMOD0 bits of TAiMR register (i = 0 to 4) = '$10_2$' (= one-shot timer mode) and the MR2 bit of TAiMR register = "0" (=TAiOS bit enabled). When read, its content is "0". | RW |
| TA1OS | Timer A1 one-shot start flag | | RW |
| TA2OS | Timer A2 one-shot start flag | | RW |
| TA3OS | Timer A3 one-shot start flag | | RW |
| TA4OS | Timer A4 one-shot start flag | | 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 : Input on $TA0_{IN}$ is selected (Note 1)<br>0 1 : TB2 overflow is selected (Note 2)<br>1 0 : TA4 overflow is selected (Note 2)<br>1 1 : TA1 overflow is selected (Note 2) | RW |
| TA0TGH | | | RW |

Note 1: Make sure the PD7_1 bit of PD7 register is set to "0" (= input mode).
Note 2: Overflow or underflow

### Trigger select register

| Symbol | Address | After reset |
|---|---|---|
| TRGSR | $0383_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TA1TGL | Timer A1 event/trigger select bit | b1 b0<br>0 0 : Input on $TA1_{IN}$ is selected (Note)<br>0 1 : TB2 is selected<br>1 0 : TA0 is selected<br>1 1 : TA2 is selected | RW |
| TA1TGH | | | RW |
| TA2TGL | Timer A2 event/trigger select bit | b3 b2<br>0 0 : Input on $TA2_{IN}$ is selected (Note)<br>0 1 : TB2 is selected<br>1 0 : TA1 is selected<br>1 1 : TA3 is selected | RW |
| TA2TGH | | | RW |
| TA3TGL | Timer A3 event/trigger select bit | b5 b4<br>0 0 : Input on $TA3_{IN}$ is selected (Note)<br>0 1 : TB2 is selected<br>1 0 : TA2 is selected<br>1 1 : TA4 is selected | RW |
| TA3TGH | | | RW |
| TA4TGL | Timer A4 event/trigger select bit | b7 b6<br>0 0 : Input on $TA4_{IN}$ is selected (Note)<br>0 1 : TB2 is selected<br>1 0 : TA3 is selected<br>1 1 : TA0 is selected | RW |
| TA4TGH | | | RW |

Note : Make sure the port direction bits for the $TA1_{IN}$ to $TA4_{IN}$ pins are set to "0" (= input mode).

### Clock prescaler reset flag

| Symbol | Address | After reset |
|---|---|---|
| CPSRF | $0381_{16}$ | $0XXXXXXX_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| — (b6-b0) | Nothing is assigned. When write, set to "0". When read, their contents are indeterminate. | | — |
| CPSR | Clock prescaler reset flag | Setting this bit to "1" initializes the prescaler for the timekeeping clock. ( When read, its content is "0".) | RW |

**Figure 12.1.4. ONSF Register, TRGSR Register, and CPSRF Register**

### 12.1.1. Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 12.1.1.1). Figure 1.2.1.1.1 shows TAiMR register in timer mode.

**Table 12.1.1.1. Specifications in Timer Mode**

| Item | Specification |
|---|---|
| Count source | $f_1$, $f_2$, $f_8$, $f_{32}$, $f_{C32}$ |
| Count operation | • Down-count<br>• When the timer underflows, it reloads the reload register contents and continues counting |
| Divide ratio | 1/(n+1) n: set value of TAi register (i= 0 to 4) $0000_{16}$ to $FFFF_{16}$ |
| Count start condition | Set TAiS bit of TABSR register to "1" (= start counting) |
| Count stop condition | Set TAiS bit to "0" (= stop counting) |
| Interrupt request generation timing | Timer underflow |
| TAiIN pin function | I/O port or gate input |
| TAiOUT pin function | I/O port or pulse output |
| Read from timer | Count value can be read by reading TAi register |
| Write to timer | • When not counting and until the 1st count source is input after counting start<br>Value written to TAi register is written to both reload register and counter<br>• When counting (after 1st count source input)<br>Value written to TAi register is written to only reload register<br>(Transferred to counter when reloaded next) |
| Select function | • Gate function<br>Counting can be started and stopped by an input signal to TAiIN pin<br>• Pulse output function<br>Whenever the timer underflows, the output polarity of TAiOUT pin is inverted.<br>When not counting, the pin outputs a low. |

Timer Ai mode register (i=0 to 4)

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

| Symbol | Address | After reset |
|---|---|---|
| TA0MR to TA4MR | $0396_{16}$ to $039A_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | b1 b0<br>0 0 : Timer mode | RW |
| TMOD1 | | | RW |
| MR0 | Pulse output function select bit | 0 : Pulse is not output (TAiOUT pin is a normal port pin)<br>1 : Pulse is output (TAiOUT pin is a pulse output pin) | RW |
| MR1 | Gate function select bit | b4 b3<br>0 0 :<br>0 1 : } Gate function not available (TAiIN pin functions as I/O port)<br>1 0 : Counts while input on the TAiIN pin is low (Note 1)<br>1 1 : Counts while input on the TAiIN pin is high (Note 1) | RW |
| MR2 | | | RW |
| MR3 | Must be set to "0" in timer mode | | RW |
| TCK0 | Count source select bit | b7 b6<br>0 0 : $f_1$ or $f_2$<br>0 1 : $f_8$<br>1 0 : $f_{32}$<br>1 1 : $f_{C32}$ | RW |
| TCK1 | | | RW |

Note 1: The port direction bit for the TAiIN pin must be set to "0" (= input mode).

**Figure 12.1.1.1. Timer Ai Mode Register in Timer Mode**

## 12.1.2. Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Timers A2, A3 and A4 can count two-phase external signals. Table 12.1.2.1 lists specifications in event counter mode (when not processing two-phase pulse signal). Table 12.1.2.2 lists specifications in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4). Figure 12.1.2.1 shows TAiMR register in event counter mode (when not processing two-phase pulse signal). Figure 12.1.2.2 shows TA2MR to TA4MR registers in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4).

**Table 12.1.2.1. Specifications in Event Counter Mode (when not processing two-phase pulse signal)**

| Item | Specification |
|---|---|
| Count source | • External signals input to TAiIN pin (i=0 to 4) (effective edge can be selected in program)<br>• Timer B2 overflows or underflows,<br>timer Aj (j=i-1, except j=4 if i=0) overflows or underflows,<br>timer Ak (k=i+1, except k=0 if i=4) overflows or underflows |
| Count operation | • Up-count or down-count can be selected by external signal or program<br>• When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading. |
| Divided ratio | 1/ (FFFF16 - n + 1) for up-count<br>1/ (n + 1) for down-count n : set value of TAi register 000016 to FFFF16 |
| Count start condition | Set TAiS bit of TABSR register to “1” (= start counting) |
| Count stop condition | Set TAiS bit to “0” (= stop counting) |
| Interrupt request generation timing | Timer overflow or underflow |
| TAiIN pin function | I/O port or count source input |
| TAiOUT pin function | I/O port, pulse output, or up/down-count select input |
| Read from timer | Count value can be read by reading TAi register |
| Write to timer | • When not counting and until the 1st count source is input after counting start<br>Value written to TAi register is written to both reload register and counter<br>• When counting (after 1st count source input)<br>Value written to TAi register is written to only reload register<br>(Transferred to counter when reloaded next) |
| Select function | • Free-run count function<br>Even when the timer overflows or underflows, the reload register content is not reloaded to it<br>• Pulse output function<br>Whenever the timer underflows or underflows, the output polarity of TAiOUT pin is inverted . When not counting, the pin outputs a low. |

Timer Ai mode register (i=0 to 4)
(When not using two-phase pulse signal processing)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | 0 | | | | 0 | 1 |

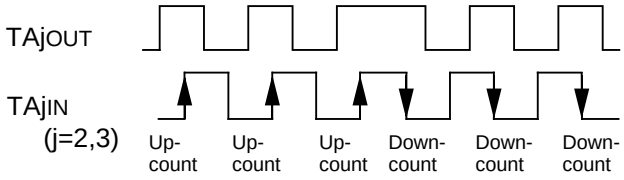
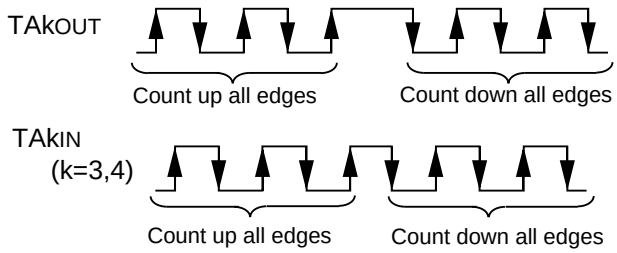
| Symbol | Address | After reset |
|---|---|---|
| TA0MR to TA4MR | $0396_{16}$ to $039A_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | b1 b0<br>0 1 : Event counter mode (Note 1) | RW |
| TMOD1 | | | RW |
| MR0 | Pulse output function select bit | 0 : Pulse is not output (TAiOUT pin functions as I/O port)<br>1 : Pulse is output (TAiOUT pin functions as pulse output pin) | RW |
| MR1 | Count polarity select bit (Note 2) | 0 : Counts external signal's falling edge<br>1 : Counts external signal's rising edge | RW |
| MR2 | Up/down switching cause select bit | 0 : UDF register<br>1 : Input signal to TAiOUT pin (Note 3) | RW |
| MR3 | Must be set to “0” in event counter mode | | RW |
| TCK0 | Count operation type select bit | 0 : Reload type<br>1 : Free-run type | RW |
| TCK1 | Can be “0” or “1” when not using two-phase pulse signal processing | | RW |

Note 1: During event counter mode, the count source can be selected using the ONSF and TRGSR registers.
Note 2: Effective when the TAiTGH and TAiTGL bits of ONSF or TRGSR register are ‘$00_2$’ (TAiIN pin input).
Note 3: Count down when input on TAiOUT pin is low or count up when input on that pin is high. The port direction bit for TAiOUT pin must be set to “0” (= input mode).

**Figure 12.1.2.1. TAiMR Register in Event Counter Mode (when not using two-phase pulse signal processing)**

**Table 12.1.2.2. Specifications in Event Counter Mode (when processing two-phase pulse signal with timers A2, A3 and A4)**

| Item | Specification |
|---|---|
| Count source | • Two-phase pulse signals input to TAiIN or TAiOUT pins (i = 2 to 4) |
| Count operation | • Up-count or down-count can be selected by two-phase pulse signal<br>• When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading. |
| Divide ratio | 1/ (FFFF16 - n + 1) for up-count<br>1/ (n + 1) for down-count n : set value of TAi register 000016 to FFFF16 |
| Count start condition | Set TAiS bit of TABSR register to “1” (= start counting) |
| Count stop condition | Set TAiS bit to “0” (= stop counting) |
| Interrupt request generation timing | Timer overflow or underflow |
| TAiIN pin function | Two-phase pulse input |
| TAiOUT pin function | Two-phase pulse input |
| Read from timer | Count value can be read by reading timer A2, A3 or A4 register |
| Write to timer | • When not counting and until the 1st count source is input after counting start<br>Value written to TAi register is written to both reload register and counter<br>• When counting (after 1st count source input)<br>Value written to TAi register is written to reload register<br>(Transferred to counter when reloaded next) |
| Select function (Note) | • Normal processing operation (timer A2 and timer A3)<br>The timer counts up rising edges or counts down falling edges on TAjIN pin when input signals on TAjOUT pin is “H”.<br>  <br>• Multiply-by-4 processing operation (timer A3 and timer A4)<br>If the phase relationship is such that TAkIN(k=3, 4) pin goes “H” when the input signal on TAkOUT pin is “H”, the timer counts up rising and falling edges on TAkOUT and TAkIN pins. If the phase relationship is such that TAkIN pin goes “L” when the input signal on TAkOUT pin is “H”, the timer counts down rising and falling edges on TAkOUT and TAkIN pins.<br>  <br>• Counter initialization by Z-phase input (timer A3)<br>The timer count value is initialized to 0 by Z-phase input. |

Notes:
1. Only timer A3 is selectable. Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.

Timer Ai mode register (i=2 to 4)
(When using two-phase pulse signal processing)

| b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|
| | | 0 | 1 | 0 | 0 | 0 | 1 |

| Symbol | Address | After reset |
|---|---|---|
| TA2MR to TA4MR | $0398_{16}$ to $039A_{16}$ | $00_{16}$ |

| | Bit name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | b1 b0<br>0 1 : Event counter mode | RW |
| TMOD1 | | | RW |
| MR0 | To use two-phase pulse signal processing, set this bit to “0”. | | RW |
| MR1 | To use two-phase pulse signal processing, set this bit to “0”. | | RW |
| MR2 | To use two-phase pulse signal processing, set this bit to “1”. | | RW |
| MR3 | To use two-phase pulse signal processing, set this bit to “0”. | | RW |
| TCK0 | Count operation type select bit | 0 : Reload type<br>1 : Free-run type | RW |
| TCK1 | Two-phase pulse signal processing operation select bit (Note 1)(Note 2) | 0 : Normal processing operation<br>1 : Multiply-by-4 processing operation | RW |

Note 1: TCK1 bit is valid for timer A3 mode register. No matter how this bit is set, timers A2 and A4 always operate in normal processing mode and x4 processing mode, respectively.
Note 2: If two-phase pulse signal processing is desired, following register settings are required:
- Set the UDF register’s TAiP bit to “1” (two-phase pulse signal processing function enabled).
- Set the TRGSR register’s TAiTGH and TAiTGL bits to ‘$00_2$’ (TAiIN pin input).
- Set the port direction bits for TAiIN and TAiOUT to “0” (input mode).

**Figure 12.1.2.2. TA2MR to TA4MR Registers in Event Counter Mode (when using two-phase pulse signal processing with timer A2, A3 or A4)**

#### 12.1.2.1 Counter Initialization by Two-Phase Pulse Signal Processing

This function initializes the timer count value to “0” by Z-phase (counter initialization) input during two-phase pulse signal processing.

This function can only be used in timer A3 event counter mode during two-phase pulse signal processing, free-running type, x4 processing, with Z-phase entered from the $\overline{INT2}$ pin.

Counter initialization by Z-phase input is enabled by writing “$0000_{16}$” to the TA3 register and setting the TAZIE bit in ONSF register to “1” (= Z-phase input enabled).

Counter initialization is accomplished by detecting Z-phase input edge. The active edge can be chosen to be the rising or falling edge by using the POL bit of INT2IC register. The Z-phase pulse width applied to the $\overline{INT2}$ pin must be equal to or greater than one clock cycle of the timer A3 count source.

The counter is initialized at the next count timing after recognizing Z-phase input. Figure 12.1.2.1.1 shows the relationship between the two-phase pulse (A phase and B phase) and the Z phase.

If timer A3 overflow or underflow coincides with the counter initialization by Z-phase input, a timer A3 interrupt request is generated twice in succession. Do not use the timer A3 interrupt when using this function.

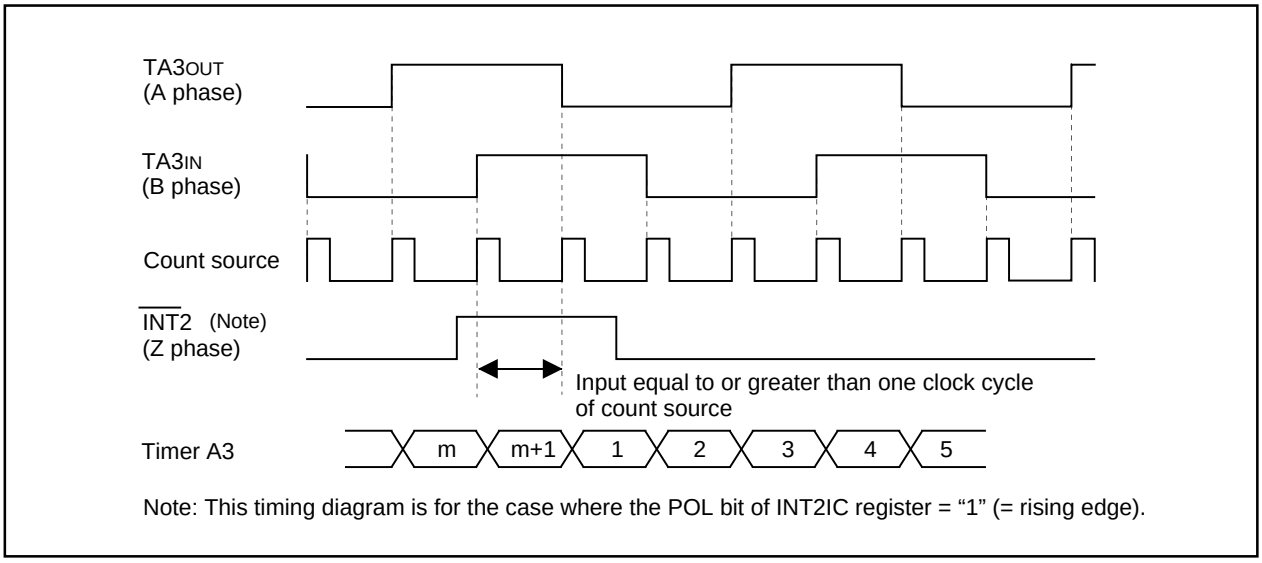



**Figure 12.1.2.1.1. Two-phase Pulse (A phase and B phase) and the Z Phase**

### 12.1.3. One-shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger. (See Table 12.1.3.1.) When the trigger occurs, the timer starts up and continues operating for a given period. Figure 12.1.3.1 shows the TAiMR register in one-shot timer mode.

**Table 12.1.3.1. Specifications in One-shot Timer Mode**

| Item | Specification |
|---|---|
| Count source | $f_1$, $f_2$, $f_8$, $f_{32}$, $f_{C32}$ |
| Count operation | • Down-count<br>• When the counter reaches $0000_{16}$, it stops counting after reloading a new value<br>• If a trigger occurs when counting, the timer reloads a new count and restarts counting |
| Divide ratio | 1/n n : set value of TAi register $0000_{16}$ to $FFFF_{16}$<br>However, the counter does not work if the divide-by-n value is set to $0000_{16}$. |
| Count start condition | TAiS bit of TABSR register = "1" (start counting) and one of the following triggers occurs.<br>• External trigger input from the TAiIN pin<br>• Timer B2 overflow or underflow,<br>timer Aj (j=i-1, except j=4 if i=0) overflow or underflow,<br>timer Ak (k=i+1, except k=0 if i=4) overflow or underflow<br>• The TAiOS bit of ONSF register is set to "1" (= timer starts) |
| Count stop condition | • When the counter is reloaded after reaching "$0000_{16}$"<br>• TAiS bit is set to "0" (= stop counting) |
| Interrupt request generation timing | When the counter reaches "$0000_{16}$" |
| TAiIN pin function | I/O port or trigger input |
| TAiOUT pin function | I/O port or pulse output |
| Read from timer | An indeterminate value is read by reading TAi register |
| Write to timer | • When not counting and until the 1st count source is input after counting start<br>Value written to TAi register is written to both reload register and counter<br>• When counting (after 1st count source input)<br>Value written to TAi register is written to only reload register<br>(Transferred to counter when reloaded next) |
| Select function | • Pulse output function<br>The timer outputs a low when not counting and a high when counting. |

Timer Ai mode register (i=0 to 4)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | 0 | | | | 1 | 0 |

| Symbol | Address | After reset |
|---|---|---|
| TA0MR to TA4MR | $396_{16}$ to $039A_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | b1 b0<br>1 0 : One-shot timer mode | RW |
| TMOD1 | | | RW |
| MR0 | Pulse output function select bit | 0 : Pulse is not output (TAiOUT pin functions as I/O port)<br>1 : Pulse is output (TAiOUT pin functions as a pulse output pin) | RW |
| MR1 | External trigger select bit (Note 1) | 0 : Falling edge of input signal to TAiIN pin (Note 2)<br>1 : Rising edge of input signal to TAiIN pin (Note 2) | RW |
| MR2 | Trigger select bit | 0 : TAiOS bit is enabled<br>1 : Selected by TAiTGH to TAiTGL bits | RW |
| MR3 | Must be set to “0” in one-shot timer mode | | RW |
| TCK0 | Count source select bit | b7 b6<br>0 0 : $f_1$ or $f_2$<br>0 1 : $f_8$<br>1 0 : $f_{32}$<br>1 1 : $f_{C32}$ | RW |
| TCK1 | | | RW |

Note 1: Effective when the TAiTGH and TAiTGL bits of ONSF or TRGSR register are ‘$00_2$’ (TAiIN pin input).
Note 2: The port direction bit for the TAiIN pin must be set to “0” (= input mode).

**Figure 12.1.3.1. TAiMR Register in One-shot Timer Mode**

## 12.1.4. Pulse Width Modulation (PWM) Mode

In PWM mode, the timer outputs pulses of a given width in succession (see Table 12.1.4.1). The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator. Figure 12.1.4.1 shows TAiMR register in pulse width modulation mode. Figures 12.1.4.2 and 12.1.4.3 show examples of how a 16-bit pulse width modulator operates and how an 8-bit pulse width modulator operates.

**Table 12.1.4.1. Specifications in Pulse Width Modulation Mode**

| Item | Specification |
|---|---|
| Count source | $f_1$, $f_2$, $f_8$, $f_{32}$, $f_{C32}$ |
| Count operation | • Down-count (operating as an 8-bit or a 16-bit pulse width modulator)<br>• The timer reloads a new value at a rising edge of PWM pulse and continues counting<br>• The timer is not affected by a trigger that occurs during counting |
| 16-bit PWM | • High level width n / fj n : set value of TAi register (i=o to 4)<br>• Cycle time $(2^{16}-1)$ / fj fixed fj: count source frequency ($f_1$, $f_2$, $f_8$, $f_{32}$, $f_{C32}$) |
| 8-bit PWM | • High level width n x (m+1) / fj n : set value of TAi register high-order address<br>• Cycle time $(2^8-1)$ x (m+1) / fj m : set value of TAi register low-order address |
| Count start condition | • TAiS bit of TABSR register is set to “1” (= start counting)<br>• The TAiS bit = 1 and external trigger input from the $TAi_{IN}$ pin<br>• The TAiS bit = 1 and one of the following external triggers occurs<br>• Timer B2 overflow or underflow,<br>timer Aj (j=i-1, except j=4 if i=0) overflow or underflow,<br>timer Ak (k=i+1, except k=0 if i=4) overflow or underflow |
| Count stop condition | TAiS bit is set to “0” (= stop counting) |
| Interrupt request generation timing | PWM pulse goes “L” |
| $TAi_{IN}$ pin function | I/O port or trigger input |
| $TAi_{OUT}$ pin function | Pulse output |
| Read from timer | An indeterminate value is read by reading TAi register |
| Write to timer | • When not counting and until the 1st count source is input after counting start<br>Value written to TAi register is written to both reload register and counter<br>• When counting (after 1st count source input)<br>Value written to TAi register is written to only reload register<br>(Transferred to counter when reloaded next) |

Timer Ai mode register (i= 0 to 4)

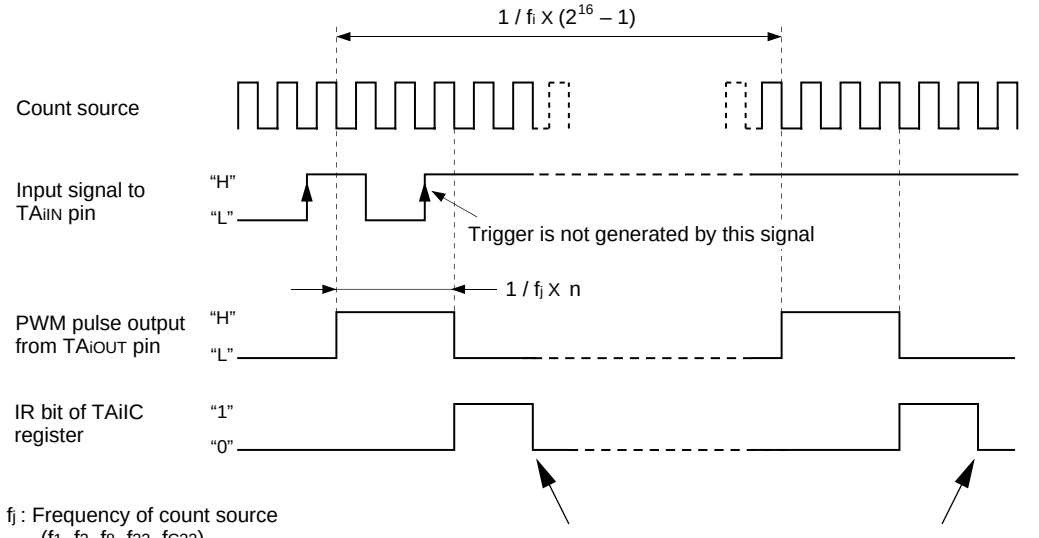
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | 1 | 1 | 1 |

| Symbol | Address | After reset |
|---|---|---|
| TA0MR to TA4MR | $0396_{16}$ to $039A_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | b1 b0<br>1 1 : PWM mode | RW |
| TMOD1 | | | RW |
| MR0 | Must be set to “1” in PWM mode | | RW |
| MR1 | External trigger select bit (Note 1) | 0: Falling edge of input signal to TAiIN pin(Note 2)<br>1: Rising edge of input signal to TAiIN pin(Note 2) | RW |
| MR2 | Trigger select bit | 0 : Write “1” to TAiS bit in the TASF register<br>1 : Selected by TAiTGH to TAiTGL bits | RW |
| MR3 | 16/8-bit PWM mode select bit | 0: Functions as a 16-bit pulse width modulator<br>1: Functions as an 8-bit pulse width modulator | RW |
| TCK0 | Count source select bit | b7 b6<br>0 0 : $f_1$ or $f_2$<br>0 1 : $f_8$<br>1 0 : $f_{32}$<br>1 1 : $f_{C32}$ | RW |
| TCK1 | | | RW |

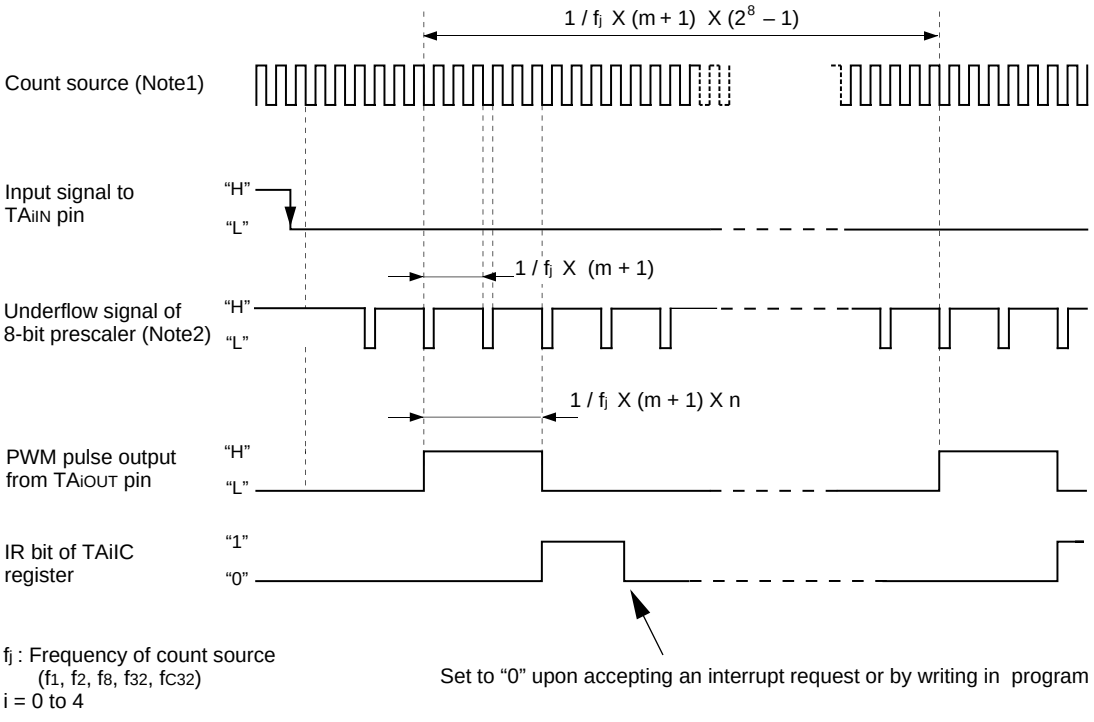
Note 1: Effective when the TAiTGH and TAiTGL bits of ONSF or TRGSR register are ‘$00_2$’ (TAiIN pin input).
Note 2: The port direction bit for the TAiIN pin must be set to “0” (= input mode).

**Figure 12.1.4.1. TAiMR Register in Pulse Width Modulation Mode**




Note 1: n = 0000$_{16}$ to FFFE$_{16}$.
Note 2: This timing diagram is for the case where the TAi register is '0003$_{16}$,' the TAiTGH and TAiTGL bits of ONSF or TRGSR register = '00$_2$' (TAiIN pin input), the MR1 bit of TAiMR register = 1 (rising edge), and the MR2 bit of TAiMR register = 1 (trigger selected by TAiTGH and TAiTGL bits).

**Figure 12.1.4.2. Example of 16-bit Pulse Width Modulator Operation**




Note 1: The 8-bit prescaler counts the count source.
Note 2: The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
Note 3: m = 00$_{16}$ to FF$_{16}$; n = 00$_{16}$ to FE$_{16}$.
Note 4: This timing diagram is for the case where the TAi register is '0202$_{16}$,' the TAiTGH and TAiTGL bits of ONSF or TRGSR register = '00$_2$' (TAiIN pin input), the MR1 bit of TAiMR register = 0 (falling edge), and the MR2 bit of TAiMR register = 1 (trigger selected by TAiTGH and TAiTGL bits).

**Figure 12.1.4.3. Example of 8-bit Pulse Width Modulator Operation**

## 12.2 Timer B

Figure 12.2.1 shows a block diagram of the timer B. Figures 12.2.2 and 12.2.3 show registers related to the timer B.

Timer B supports the following four modes. Use the TMOD1 and TMOD0 bits of TBiMR register (i = 0 to 2) to select the desired mode.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external device or overflows or underflows of other timers.
- Pulse period/pulse width measuring mode: The timer measures an external signal's pulse period or pulse width.
- A-D trigger mode: The timer counts only once before it reaches the minimum count "0000$_{16}$"

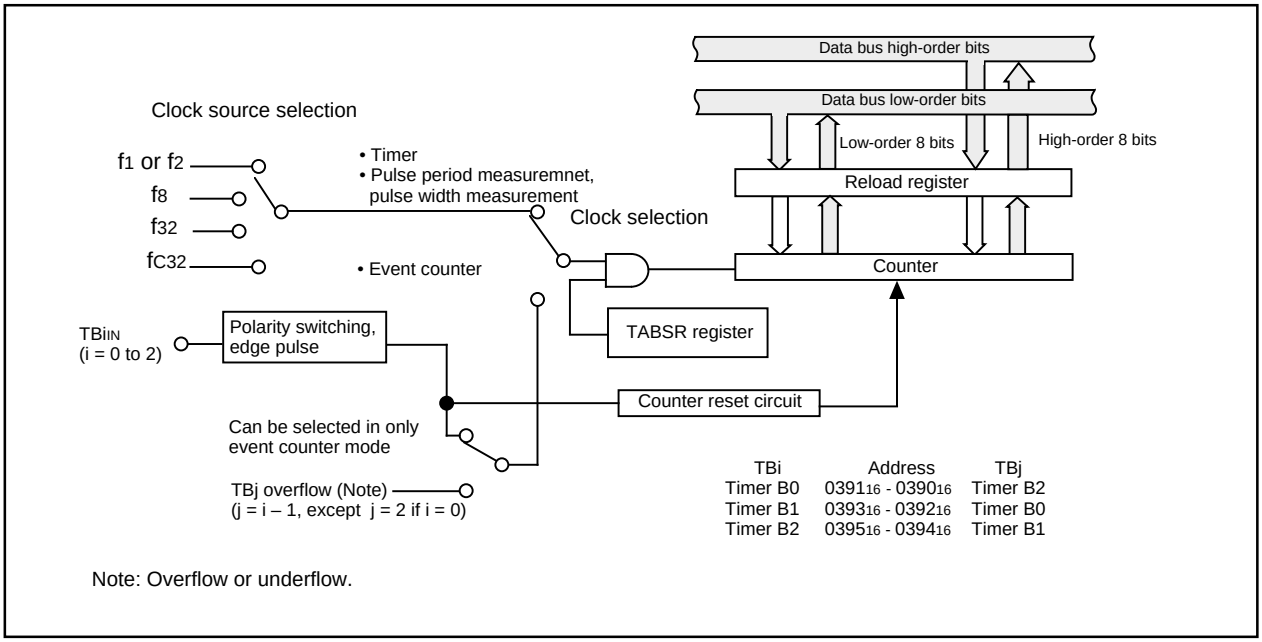



**Figure 12.2.1. Timer B Block Diagram**

Timer Bi mode register (i=0 to 2)

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

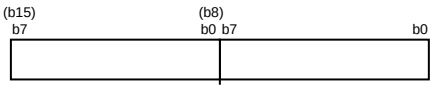
| Symbol | Address | After reset |
|---|---|---|
| TB0MR to TB2MR | 039B$_{16}$ to 039D$_{16}$ | 00XX0000$_2$ |

| Bit symbol | Bit | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | b1 b0<br>0 0 : Timer mode or A-D trigger mode<br>0 1 : Event counter mode<br>1 0 : Pulse period measurement mode, pulse width measurement mode<br>1 1 : Must not be set | RW |
| TMOD1 | | | RW |
| MR0 | ——— | Function varies with each operation mode | RW |
| MR1 | | | RW |
| MR2 | | | RW (Note 1)<br>——— (Note 2) |
| MR3 | | | RO |
| TCK0 | Count source select bit | Function varies with each operation mode | RW |
| TCK1 | | | RW |

Note 1: Timer B0.
Note 2: Timer B1, Timer B2.

**Figure 12.2.2. TB0MR to TB2MR Registers**

### Timer Bi register (i=0 to 2)(Note 1)

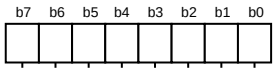



| Symbol | Address | After reset |
|---|---|---|
| TB0 | 039116, 039016 | Indeterminate |
| TB1 | 039316, 039216 | Indeterminate |
| TB2 | 039516, 039416 | Indeterminate |

| Mode | Function | Setting range | RW |
|---|---|---|---|
| Timer mode | Divide the count source by n + 1 where n = set value | 000016 to FFFF16 | RW |
| Event counter mode | Divide the count source by n + 1 where n = set value (Note 2) | 000016 to FFFF16 | RW |
| Pulse period modulation mode, Pulse width modulation mode | Measures a pulse period or width | —— | RO |
| A-D trigger mode (Note 3) | Divide the count source by n + 1 where n = set value and cause the timer stop | 000016 to FFFF16 | RW |

Note 1: The register must be accessed in 16 bit units.
Note 2: The timer counts pulses from an external device or overflows or underflows of other timers.
Note 3: When this mode is used combining delayed trigger mode 0, set the larger value than the value of the timer B0 register to the timer B1 register.

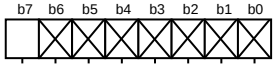
### Count start flag




| Symbol | Address | After reset |
|---|---|---|
| TABSR | 038016 | 0016 |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TA0S | Timer A0 count start flag | 0 : Stops counting<br>1 : Starts counting | RW |
| TA1S | Timer A1 count start flag | | RW |
| TA2S | Timer A2 count start flag | | RW |
| TA3S | Timer A3 count start flag | | RW |
| TA4S | Timer A4 count start flag | | RW |
| TB0S | Timer B0 count start flag | | RW |
| TB1S | Timer B1 count start flag | | RW |
| TB2S | Timer B2 count start flag | | RW |

### Clock prescaler reset flag




| Symbol | Address | After reset |
|---|---|---|
| CPSRF | 038116 | 0XXXXXXX2 |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| —— (b6-b0) | Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate. | | — |
| CPSR | Clock prescaler reset flag | Setting this bit to “1” initializes the prescaler for the timekeeping clock. (When read, the value of this bit is “0”.) | RW |

**Figure 12.2.3. TB0 to TB2 Registers, TABSR Register, CPSRF Register**

## 12.2.1 Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 12.2.1.1). Figure 12.2.1.1 shows TBiMR register in timer mode.

**Table 12.2.1.1 Specifications in Timer Mode**

| Item | Specification |
|---|---|
| Count source | $f_1$, $f_2$, $f_8$, $f_{32}$, $f_{C32}$ |
| Count operation | • Down-count<br>• When the timer underflows, it reloads the reload register contents and continues counting |
| Divide ratio | 1/(n+1) n: set value of TBi register (i= 0 to 2) $0000_{16}$ to $FFFF_{16}$ |
| Count start condition | Set TBiS bit(Note) to "1" (= start counting) |
| Count stop condition | Set TBiS bit to "0" (= stop counting) |
| Interrupt request generation timing | Timer underflow |
| $TBi_{IN}$ pin function | I/O port |
| Read from timer | Count value can be read by reading TBi register |
| Write to timer | • When not counting and until the 1st count source is input after counting start<br>Value written to TBi register is written to both reload register and counter<br>• When counting (after 1st count source input)<br>Value written to TBi register is written to only reload register<br>(Transferred to counter when reloaded next) |

Note : The TB0S to TB2S bits are assigned to the TABSR register bit 5 to bit 7.

Timer Bi mode register (i= 0 to 2)

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

Symbol: TB0MR to TB2MR — Address: $039B_{16}$ to $039D_{16}$ — After reset: $00XX0000_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | b1 b0<br>0 0 : Timer mode or A-D trigger mode | RW |
| TMOD1 | | | RW |
| MR0 | Has no effect in timer mode<br>Can be set to "0" or "1" | | RW |
| MR1 | | | RW |
| MR2 | TB0MR register<br>Must be set to "0" in timer mode | | RW |
| | TB1MR, TB2MR registers<br>Nothing is assigned. When write, set to "0". When read, its content is indeterminate | | — |
| MR3 | When write in timer mode, set to "0". When read in timer mode, its content is indeterminate. | | RO |
| TCK0 | Count source select bit | b7 b6<br>0 0 : $f_1$ or $f_2$<br>0 1 : $f_8$<br>1 0 : $f_{32}$<br>1 1 : $f_{C32}$ | RW |
| TCK1 | | | RW |

**Figure 12.2.1.1 TBiMR Register in Timer Mode**

## 12.2.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers (see Table 12.2.2.1) . Figure 12.2.2.1 shows TBiMR register in event counter mode.

**Table 12.2.2.1 Specifications in Event Counter Mode**

| Item | Specification |
|---|---|
| Count source | • External signals input to TBiIN pin (i=0 to 2) (effective edge can be selected in program)<br>• Timer Bj overflow or underflow (j=i-1, except j=2 if i=0) |
| Count operation | • Down-count<br>• When the timer underflows, it reloads the reload register contents and continues counting |
| Divide ratio | 1/(n+1) n: set value of TBi register 0000$_{16}$ to FFFF$_{16}$ |
| Count start condition | Set TBiS bit[1] to “1” (= start counting) |
| Count stop condition | Set TBiS bit to “0” (= stop counting) |
| Interrupt request generation timing | Timer underflow |
| TBiIN pin function | Count source input |
| Read from timer | Count value can be read by reading TBi register |
| Write to timer | • When not counting and until the 1st count source is input after counting start<br>Value written to TBi register is written to both reload register and counter<br>• When counting (after 1st count source input)<br>Value written to TBi register is written to only reload register<br>(Transferred to counter when reloaded next) |

Notes:
1. The TB0S to TB2S bits are assigned to the TABSR register bit 5 to bit 7.

Timer Bi mode register (i=0 to 2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | | 0 | 1 |

| Symbol | Address | After reset |
|---|---|---|
| TB0MR to TB2MR | 039B$_{16}$ to 039D$_{16}$ | 00XX0000$_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | b1 b0<br>0 1 : Event counter mode | RW |
| TMOD1 | | | RW |
| MR0 | Count polarity select bit (Note 1) | b3 b2<br>0 0 : Counts external signal's falling edges<br>0 1 : Counts external signal's rising edges<br>1 0 : Counts external signal's falling and rising edges<br>1 1 : Must not be set | RW |
| MR1 | | | RW |
| MR2 | TB0MR register<br>Must be set to “0” in timer mode | | RW |
| | TB1MR, TB2MR registers<br>Nothing is assigned. When write, set to “0”. When read, its content is indeterminate. | | — |
| MR3 | When write in event counter mode, set to “0”. When read in event counter mode, its content is indeterminate. | | RO |
| TCK0 | Has no effect in event counter mode.<br>Can be set to “0” or “1”. | | RW |
| TCK1 | Event clock select | 0 : Input from TBiIN pin (Note 2)<br>1 : TBj overflow or underflow<br>(j = i – 1, except j = 2 if i = 0) | RW |

Note 1: Effective when the TCK1 bit = “0” (input from TBiIN pin). If the TCK1 bit = “1” (TBj overflow or underflow), these bits can be set to “0” or “1”.
Note 2: The port direction bit for the TBiIN pin must be set to “0” (= input mode).

**Figure 12.2.2.1 TBiMR Register in Event Counter Mode**

### 12.2.3 Pulse Period and Pulse Width Measurement Mode

In pulse period and pulse width measurement mode, the timer measures pulse period or pulse width of an external signal (see Table 12.2.3.1). Figure 12.2.3.1 shows TBiMR register in pulse period and pulse width measurement mode. Figure 12.2.3.2 shows the operation timing when measuring a pulse period. Figure 12.2.3.3 shows the operation timing when measuring a pulse width.

**Table 12.2.3.1 Specifications in Pulse Period and Pulse Width Measurement Mode**

| Item | Specification |
|---|---|
| Count source | $f_1$, $f_2$, $f_8$, $f_{32}$, $f_{C32}$ |
| Count operation | • Up-count<br>• Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to "$0000_{16}$" to continue counting. |
| Count start condition | Set TBiS (i=0 to 2) bit[3] to "1" (= start counting) |
| Count stop condition | Set TBiS bit to "0" (= stop counting) |
| Interrupt request generation timing | • When an effective edge of measurement pulse is input[1]<br>• Timer overflow. When an overflow occurs, MR3 bit of TBiMR register is set to "1" (overflowed) simultaneously. MR3 bit is cleared to "0" (no overflow) by writing to TBiMR register at the next count timing or later after MR3 bit was set to "1". At this time, make sure TBiS bit is set to "1" (start counting). |
| $TBi_{IN}$ pin function | Measurement pulse input |
| Read from timer | Contents of the reload register (measurement result) can be read by reading TBi register[2] |
| Write to timer | Value written to TBi register is written to neither reload register nor counter |

Notes:
1. Interrupt request is not generated when the first effective edge is input after the timer started counting.
2. Value read from TBi register is indeterminate until the second valid edge is input after the timer starts counting.
3. The TB0S to TB2S bits are assigned to the TABSR register bit 5 to bit 7.

Timer Bi mode register (i=0 to 2)

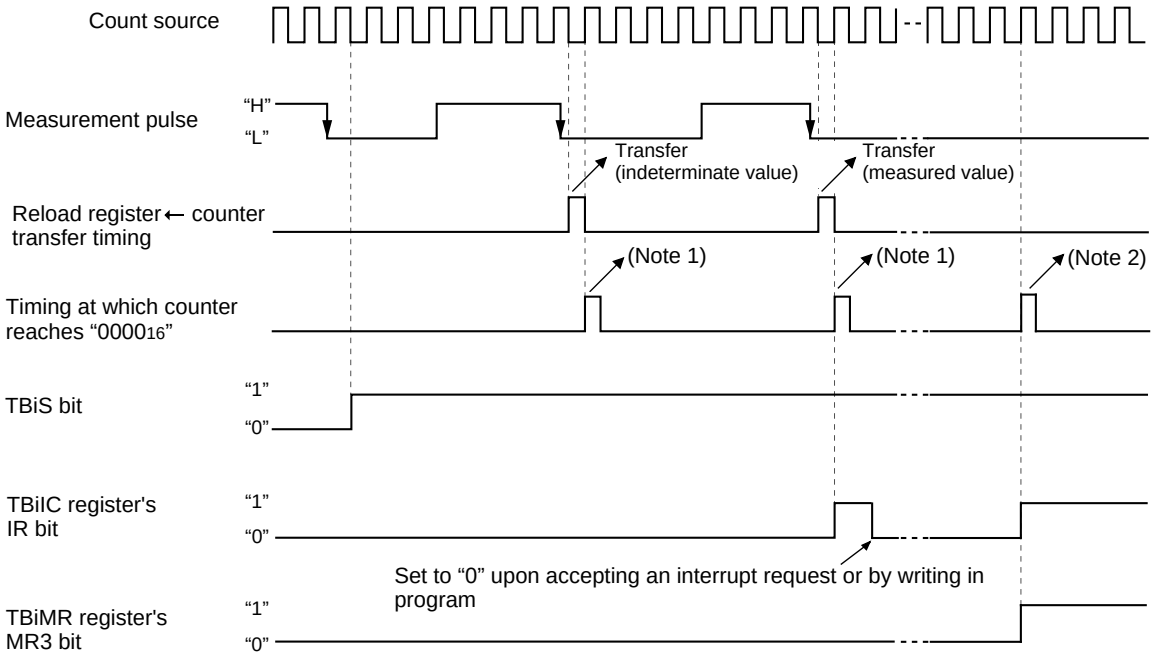
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | | 1 | 0 |

| Symbol | Address | After reset |
|---|---|---|
| TB0MR to TB2MR | $039B_{16}$ to $039D_{16}$ | $00XX0000_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | b1 b0<br>1 0 : Pulse period / pulse width measurement mode | RW |
| TMOD1 | | | RW |
| MR0 | Measurement mode select bit | b3 b2<br>0 0 : Pulse period measurement (Measurement between a falling edge and the next falling edge of measured pulse)<br>0 1 : Pulse period measurement (Measurement between a rising edge and the next rising edge of measured pulse)<br>1 0 : Pulse width measurement (Measurement between a falling edge and the next rising edge of measured pulse and between a rising edge and the next falling edge)<br>1 1 : Must not be set. | RW |
| MR1 | | | RW |
| MR2 | TB0MR register<br>Must be set to "0" in pulse period and pulse width measurement mode | | RW |
| | TB1MR, TB2MR registers<br>Nothing is assigned. When write, set to "0". When read, its content turns out to be indeterminate. | | — |
| MR3 | Timer Bi overflow flag ( Note) | 0 : Timer did not overflow<br>1 : Timer has overflowed | RO |
| TCK0 | Count source select bit | b7 b6<br>0 0 : $f_1$ or $f_2$<br>0 1 : $f_8$<br>1 0 : $f_{32}$<br>1 1 : $f_{C32}$ | RW |
| TCK1 | | | RW |

Note: This flag is indeterminate after reset. When the TBiS bit = 1 (start counting), the MR3 bit is cleared to "0" (no overflow) by writing to the TBiMR register at the next count timing or later after the MR3 bit was set to "1" (overflowed). The MR3 bit cannot be set to "1" in a program. The TB0S to TB2S bits are assigned to the TABSR register's bit 5 to bit 7.

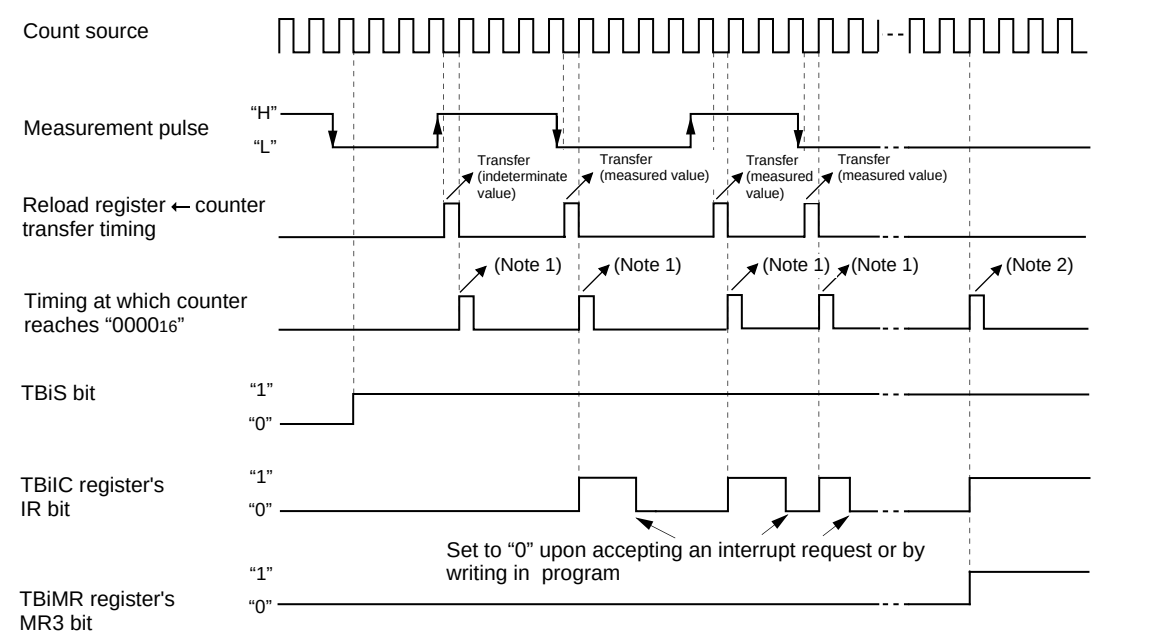
**Figure 12.2.3.1 TBiMR Register in Pulse Period and Pulse Width Measurement Mode**




The TB0S to TB2S bits are assigned to the TABSR register's bit 5 to bit 7.

i = 0 to 2
Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.
Note 3: This timing diagram is for the case where the TBiMR register's MR1 to MR0 bits are "00₂" (measure the interval from falling edge to falling edge of the measurement pulse).

**Figure 12.2.3.2 Operation timing when measuring a pulse period**




The TB0S to TB2S bits are assigned to the TABSR register's bit 5 to bit 7.

i = 0 to 2
Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.
Note 3: This timing diagram is for the case where the TBiMR register's MR1 to MR0 bits are "10₂" (measure the interval from a falling edge to the next rising edge and the interval from a rising edge to the next falling edge of the measurement pulse).

**Figure 12.2.3.3 Operation timing when measuring a pulse width**

### 12.2.4 A-D Trigger Mode

A-D trigger mode is used as conversion start trigger for A-D converter in simultaneous sample sweep mode of A-D conversion or delayed trigger mode 0. This mode is used as conversion start trigger of A-D converter. A-D trigger mode is used in timer B0 and timer B1. In this mode, the timer is activated only by one trigger. A-D trigger mode is available only for timer B0 and time B1. Figure 12.2.4.1 shows the TBiMR register in A-D trigger mode and figure 12.2.4.2 shows the TB2SC register.

**Table 12.2.4.1 Specifications in A-D Trigger Mode**

| Item | Specification |
|---|---|
| Count Source | $f_1$, $f_2$, $f_8$, $f_{32}$, and $f_{C32}$ |
| Count Operation | • Down count<br>• When the timer underflows, reload register contents are reloaded before stopping counting<br>• When a trigger is generated during the count operation, the count is not affected |
| Divide Ratio | 1/(n+1) n: Setting value of TBi register (i=0,1)<br>$0000_{16}$-$FFFF_{16}$ |
| Count Start Condition | When the TBiS (i=0,1) bit in the TABSR register is "1"(count started), TBiEN(i=0,1) in TB2SC register is "1" and the following trigger is generated. (Selection based on TB2SEL, TBiTRIG (i=0,1) bits of TB2SC)<br>• Timer B2 overflow or underflow<br>• Underflow of Timer B2 interrupt generation frequency counter setting |
| Count Stop Condition | • After the count value is $0000_{16}$ and reload register contents are reloaded<br>• Set the TBiS bit to "0"(count stopped) |
| Interrupt Request Generation Timing | Timer underflows (Note 1) |
| TBiIN Pin Function | I/O port |
| Read From Timer | Count value can be read by reading TBi register |
| Write To Timer (Note 2) | • When writing in the TBi register during count stopped.<br>Value is written to both reload register and counter<br>• When writing in the TBi register during count.<br>Value is written to only reload register (Transfered to counter when reloaded next) |

Note 1: A-D conversion is started by the timer underflow.
For details refer to **Section 14. A-D Converter**.

Note 2: When using in delayed trigger mode 0, set the larger value than the value of the timer B0 register to the timer B1 register.

Timer Bi mode register (i= 0 to 1)

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

Symbol: TB0MR to TB1MR
Address: 039B16 to 039C16
After reset: 00XX00002

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | b1 b0<br>0 0 : Timer mode or A-D trigger mode | RW |
| TMOD1 | | | RW |
| MR0 | Invalid in A-D trigger mode<br>Either "0" or "1" is enabled | | RW |
| MR1 | | | RW |
| MR2 | TB0MR register<br>Set to “0” in A-D trigger mode | | RW |
| | TB1MR register<br>Nothing is assigned. When write, set to “0”. When read, its content is indeterminate | | — |
| MR3 | When write in A-D trigger mode, set to “0”. When read in A-D trigger mode, its content is indeterminate. | | RO |
| TCK0 | Count source select bit | b7 b6<br>0 0 : f1 or f2<br>0 1 : f8<br>1 0 : f32<br>1 1 : fC32 | RW |
| TCK1 | | | RW |

**Figure 12.2.4.1 TBiMR Register in Delayed Trigger Mode**

Timer B2 special mode register (Note 1)

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

Symbol: TB2SC
Address: 039E16
After reset: X00000002

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PWCOM | Timer B2 Reload Timing Switch Bit (Note 2) | 0 : Timer B2 underflow<br>1 : Timer A output at odd-numbered | RW |
| IVPCR1 | Three-Phase Output Port $\overline{SD}$ Control Bit 1 (Note 3, 4, 7) | 0 : Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) disabled<br>1 : Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) enabled | RW |
| TB0EN | Timer B0 Operation Mode Select Bit | 0 : Other than A-D trigger mode<br>1 : A-D trigger mode (Note 5) | RW |
| TB1EN | Timer B1 Operation Mode Select Bit | 0 : Other than A-D trigger mode<br>1 : A-D trigger mode (Note 5) | RW |
| TB2SEL | Trigger Select Bit (Note 6) | 0 : TB2 interrupt<br>1 : Underflow of TB2 interrupt generation frequency setting counter [ICTB2] | RW |
| $\overline{\text{(b6-b5)}}$ | Reserved bits | Must set to "0" | RW |
| — (b7) | Nothing is assigned. When write, set to 0 .<br>When read, its content is 0 . | | — |

Note 1. Write to this register after setting the PRC1 bit in the PRCR register to "1" (write enabled).
Note 2. If the INV11 bit is "0" (three-phase mode 0) or the INV06 bit is "1" (triangular wave modulation mode), set this bit to "0" (timer B2 underflow).
Note 3. When setting the IVPCR1 bit to "1" (three-phase output forcible cutoff by $\overline{SD}$ pin input enabled), Set the PD8_5 bit to "0" (= input mode).
Note 4. Related pins are U(P80), $\overline{U}$(P81), V(P72), $\overline{V}$(P73), W(P74), $\overline{W}$(P75). After forcible cutoff, input "H" to the P85/$\overline{NMI}$/$\overline{SD}$ pin. Set the IVPCR1 bit to "0", and this forcible cutoff will be reset. If L is input to the P85/$\overline{NMI}$/$\overline{SD}$ pin, a three-phase motor control timer output will be disabled (INV03=0). At this time, when the IVPCR1 bit is "0", the target pins changes to programmable I/O port. When the IVPCR1 bit is "1", the target pins changes to high-impedance state regardless of which functions of those pins are used.
Note 5. When this bit is used in delayed trigger mode 0, set the TB0EN and TB1EN bits to "1" (A-D trigger mode).
Note 6. When setting the TB2SEL bit to "1" (underflow of TB2 interrupt generation frequency setting counter[ICTB2]), Set the INV02 bit to "1" (three-phase motor control timer function).
Note 7. Refer to "**17.6 Digital Debounce function**" for the $\overline{SD}$ input

**Figure 12.2.4.2 TB2SC Register**

## 12.3 Three-phase Motor Control Timer Function

Timers A1, A2, A4 and B2 can be used to output three-phase motor drive waveforms. Table 12.3.1 lists the specifications of the three-phase motor control timer function. Figure 12.3.1 shows the block diagram for three-phase motor control timer function. Also, the related registers are shown on Figure 12.3.2 to Figure 12.3.8.

**Table 12.3.1. Three-phase Motor Control Timer Function Specifications**

| Item | Specification |
|---|---|
| Three-phase waveform output pin | Six pins (U, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$) |
| Forced cutoff input (Note 1) | Input "L" to $\overline{SD}$ pin |
| Used Timers | Timer A4, A1, A2 (used in the one-shot timer mode)<br>Timer A4: U- and $\overline{U}$-phase waveform control<br>Timer A1: V- and $\overline{V}$-phase waveform control<br>Timer A2: W- and $\overline{W}$-phase waveform control<br>Timer B2 (used in the timer mode)<br>Carrier wave cycle control<br>Dead timer timer (3 eight-bit timer and shared reload register)<br>Dead time control |
| Output waveform | Triangular wave modulation, Sawtooth wave modification<br>Enable to output "H" or "L" for one cycle<br>Enable to set positive-phase level and negative-phase level respectively |
| Carrier wave cycle | Triangular wave modulation: count source x (m+1) x 2<br>Sawtooth wave modulation: count source x (m+1)<br>m: Setting value of TB2 register, 0 to 65535<br>Count source: $f_1$, $f_2$, $f_8$, $f_{32}$, $f_{C32}$ |
| Three-phase PWM output width | Triangular wave modulation: count source x n x 2<br>Sawtooth wave modulation: count source x n<br>n: Setting value of TA4, TA1 and TA2 register (of TA4, TA41, TA1, TA11, TA2 and TA21 registers when setting the INV11 bit to "1"), 1 to 65535<br>Count source: $f_1$, $f_2$, $f_8$, $f_{32}$, $f_{C32}$ |
| Dead time active disable function | Count source x p, or no dead time<br>p: Setting value of DTT register, 1 to 255<br>Count source: $f_1$, $f_2$, $f_1$ divided by 2, $f_2$ divided by 2 |
| Active level | Eable to select "H" or "L" |
| Positive and negative-phase concurrent | Positive and negative-phases concurrent active disable function<br>Positive and negative-phases concurrent active detect function |
| Interrupt frequency | For Timer B2 interrupt, select a carrier wave cycle-to-cycle basis through 15 times carrier wave cycle-to-cycle basis |

Note 1: When three phase motor control function is enabled (INV02=1) $P8_5$ becomes $\overline{SD}$. Do not use $P8_5$ for GPIO. If the $\overline{SD}$ fueature is not needed then $P8_5/\overline{SD}$ must always be driven high.
When $\overline{SD}$ is driven low, INV03 (three phase output control bit) is cleared, pins U($P8_0$), $\overline{U}$($P8_1$), V($P7_2$), $\overline{V}$($P7_3$), W($P7_4$), $\overline{W}$($P7_5$) pins go back to GPIO mode and are controlled by their corresponding Port Direction and Data registers. In addition if bit IVPRC1 is set to 1 when $\overline{SD}$ is driven low pin $P8_0$, $P8_1$, $P7_2$, $P7_3$, $P7_4$, $P7_5$ tri-state regardless of when function (3 phase, GPIO, or UART) is assigned to them.

Related pins
$P7_2$/CLK2/$TA1_{OUT}$/V/$RXD_1$
$P7_3/\overline{CTS_2}/\overline{RTS_2}$/$TA1_{IN}/\overline{V}$/$TXD_1$
$P7_4$/$TA2_{OUT}$/W
$P7_5$/$TA2_{IN}/\overline{W}$
$P8_0$/$TA4_{OUT}$/U
$P8_1$/$TA4_{IN}/\overline{U}$

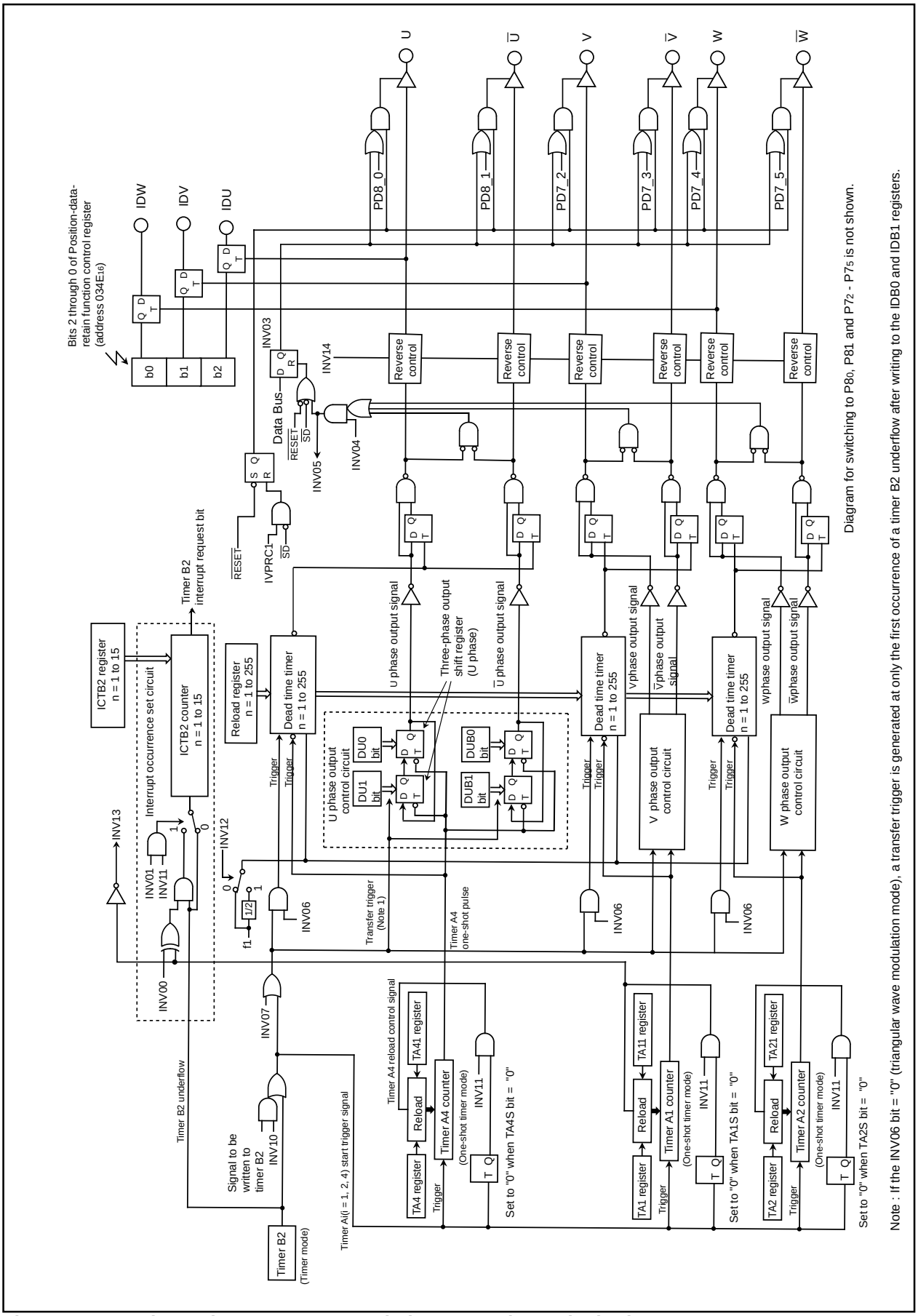



**Figure 12.3.1. Three-phase Motor Control Timer Functions Block Diagram**

Three-phase PWM control register 0 (Note 1)

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

| Symbol | Address | After reset |
|---|---|---|
| INVC0 | 0348$_{16}$ | 00$_{16}$ |

| Bit symbol | Bit name | Description | RW |
|---|---|---|---|
| INV00 | Effective interrupt output polarity select bit (Note 3) | 0: ICTB2 counter incremented by 1 at odd-numbered occurrences of a timer B2 underflow<br>1: ICTB2 counter incremented by 1 at even-numbered occurrences of a timer B2 underflow | RW |
| INV01 | Effective interrupt output specification bit (Note 2, Note 3) | 0: ICTB2 counter incremented by 1 at a timer B2 underflow<br>1: Selected by INV00 bit | RW |
| INV02 | Mode select bit (Note 4) | 0: Three-phase motor control timer function unused<br>1: Three-phase motor control timer function | RW |
| INV03 | Output control bit (Note 6) | 0: Three-phase motor control timer output disabled (Note 5)<br>1: Three-phase motor control timer output enabled (Note 10) | RW |
| INV04 | Positive and negative phases concurrent output disable bit | 0: Simultaneous active output enabled<br>1: Simultaneous active output disabled | RW |
| INV05 | Positive and negative phases concurrent output detect flag | 0: Not detected yet<br>1: Already detected (Note 7) | RW |
| INV06 | Modulation mode select bit (Note 8) | 0: Triangular wave modulation mode (Note 9)<br>1: Sawtooth wave modulation mode | RW |
| INV07 | Software trigger select bit | Setting this bit to “1” generates a transfer trigger. If the INV06 bit is “1”, a trigger for the dead time timer is also generated. The value of this bit when read is “0”. | RW |

Note 1: Write to this register after setting the PRC1 bit of PRCR register to “1” (write enable). Note also that this register can only be rewritten when timers A1, A2, A4 and B2 are idle.
Note 2: If this bit needs to be set to “1”, set any value in the ICTB2 register before writing to it.
Note 3: Effective when the INV11 bit is “1” (three-phase mode 1). If INV11 is “0” (three-phase mode 0), the ICTB2 counter is incremented by “1” each time the timer B2 underflows, regardless of whether the INV00 and INV01 bits are set.
Note 4: Setting the INV02 bit to “1” activates the dead time timer, U/V/W-phase output control circuits and ICTB2 counter.
Note 5: When INV03="1"(three-phase motor control timer output enabled) $P8_0$, $P8_1$, $P7_2$, $P7_3$, $P7_4$, and $P7_5$ functionas U, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$
Note 6: The INV03 bit is set to “0” in the following cases:
- When reset
- When positive and negative go active (INV05="1") simultaneously while INV04 bit is “1”
- When set to “0” in a program
- When input on the $\overline{SD}$ pin changes state from “H” to “L” (The INV03 bit cannot be set to “1” when $\overline{SD}$ input is “L”.)

INV03 is set to “0” when the $\overline{SD}$ pin changes from “H” to “L” regardless of the value of the INVCR1 bit.
INV03 is set to "0" when both INV04 bit and INV05 bit are "1".
Note 7: Can only be set by writing “0” in a program, and cannot be set to “1”.
Note 8: The effects of the INV06 bit are described in the table below.

| Item | INV06=0 | INV06=1 |
|---|---|---|
| Mode | Triangular wave modulation mode | Sawtooth wave modulation mode |
| Timing at which transferred from IDB0 to IDB1 registers to three-phase output shift register | Transferred only once synchronously with the transfer trigger after writing to the IDB0 to IDB1 registers | Transferred every transfer trigger |
| Timing at which dead time timer trigger is generated when INV16 bit is “0” | Synchronous with the falling edge of timer A1, A2, or A4 one-shot pulse | Synchronous with the transfer trigger and the falling edge of timer A1, A2, or A4 one-shot pulse |
| INV13 bit | Effective when INV11 is “1” and INV06 is “0” | Has no effect |

Transfer trigger: Timer B2 underflow, write to the INV07 bit or write to the TB2 register when INV10 is “1”

Note 9: If the INV06 bit is “1”, set the INV11 bit to “0” (three-phase mode 0) and set the PWCON bit to “0” (timer B2 reloaded by a timer B2 underflow).
Note 10: Individual pins can be disabled using PFCR register.

**Figure 12.3.2. INVC0 Register**

Three-phase PWM control register 1 (Note 1)

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

| Symbol | Address | After reset |
|---|---|---|
| INVC1 | $0349_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Description | RW |
|---|---|---|---|
| INV10 | Timer A1, A2, A4 start trigger signal select bit | 0: Timer B2 underflow<br>1: Timer B2 underflow and write to the TB2 register | RW |
| INV11 | Timer A1-1, A2-1, A4-1 control bit (Note 2) | 0: Three-phase mode 0<br>1: Three-phase mode 1 (Note 3) | RW |
| INV12 | Dead time timer count source select bit | 0 : $f_1$ or $f_2$<br>1 : $f_1$ divided by 2 or $f_2$ divided by 2 | RW |
| INV13 | Carrier wave detect flag (Note 4) | 0: Timer A output at even-numbered occurrences (TA11, TA21, TA41 register value counted)<br>1: Timer A output at odd-numbered occurrences (TA1, TA2, TA4 register value counted) | RO |
| INV14 | Output polarity control bit | 0 : Output waveform “L” active<br>1 : Output waveform “H” active | RW |
| INV15 | Dead time invalid bit | 0: Dead time timer enabled<br>1: Dead time timer disabled | RW |
| INV16 | Dead time timer trigger select bit | 0: Falling edge of timer A4, A1 or A2 one-shot pulse<br>1: Rising edge of three-phase output shift register (U, V or W phase) output (Note 5) | RW |
| ——— (b7) | Reserved bit | This bit should be set to “0” | RW |

Note 1: Write to this register after setting the PRC1 bit of PRCR register to “1” (write enable). Note also that this register can only be rewritten when timers A1, A2, A4 and B2 are idle.
Note 2: The effects of the INV11 bit are described in the table below.

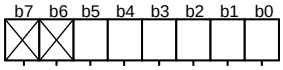
| Item | INV11=0 | INV11=1 |
|---|---|---|
| Mode | Three-phase mode 0 | Three-phase mode 1 |
| TA11, TA21, TA41 registers | Not used | Used |
| INV00 bit, INV01 bit | Has no effect. ICTB2 counted every time timer B2 underflows regardless of whether the INV00 to INV01 bits are set. | Effect |
| INV13 bit | Has no effect | Effective when INV11 bit is “1” and INV06 bit is “0” |

Note 3: If the INV06 bit is “1” (sawtooth wave modulation mode), set this bit to “0” (three-phase mode 0). Also, if the INV11 bit is “0”, set the PWCON bit to “0” (timer B2 reloaded by a timer B2 underflow).
Note 4: The INV13 bit is effective only when the INV06 bit is “0” (triangular wave modulation mode) and the INV11 bit is “1” (three-phase mode 1).
Note 5: If all of the following conditions hold true, set the INV16 bit to “1” (dead time timer triggered by the rising edge of three-phase output shift register output)
- The INV15 bit is “0” (dead time timer enabled)
- When the INV03 bit is set to “1” (three-phase motor control timer output enabled), the Dij bit and DiBj bit (i:U, V, or W, j: 0 to 1) have always different values (the positive-phase and negative-phase always output different levels during the period other than dead time).

Conversely, if either one of the above conditions holds false, set the INV16 bit to “0” (dead time timer triggered

**Figure 12.3.3. INVC1 Register**

### Three-phase output buffer register i (i=0, 1) (Note)

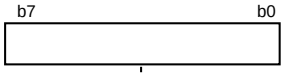



| Symbol | Address | After reset |
|---|---|---|
| IDB0 | 034A16 | 0016 |
| IDB1 | 034B16 | 0016 |

| Bit Symbol | Bit name | Function | RW |
|---|---|---|---|
| DUi | U phase output buffer i | Write the output level<br>0: Active level<br>1: Inactive level<br><br>When read, these bits show the three-phase output shift register value. | 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) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | —— |

Note: The IDB0 and IDB1 register values are transferred to the three-phase shift register by a transfer trigger. The value written to the IDB0 register after a transfer trigger represents the output signal of each phase, and the next value written to the IDB1 register at the falling edge of the timer A1, A2 or A4 one-shot pulse represents the output signal of each phase.

### Dead time timer (Note 1, Note 2)

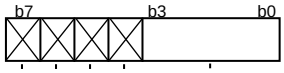



| Symbol | Address | After reset |
|---|---|---|
| DTT | 034C16 | ??16 |

| Function | Setting range | RW |
|---|---|---|
| Assuming the set value = n, upon a start trigger the timer starts counting the count source selected by the INV12 bit and stops after counting it n times. The positive or negative phase whichever is going from an inactive to an active level changes at the same time the dead time timer stops. | 1 to 255 | WO |

Note 1: Use MOV instruction to write to this register.
Note 2: Effective when the INV15 bit is “0” (dead time timer enable). If the INV15 bit is “1”, the dead time timer is disabled and has no effect.

### Timer B2 interrupt occurrences frequency set counter




| Symbol | Address | After reset |
|---|---|---|
| ICTB2 | 034D16 | X?16 |

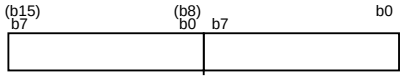
| Function | Setting range | RW |
|---|---|---|
| If the INV01 bit is ì0î (ICTB2 counter counted every time timer B2 underflows), assuming the set value = n, a timer B2 interrupt is generated at every níth occurrence of a timer B2 underflow.<br>If the INV01 bit is ì1î (ICTB2 counter count timing selected by the INV00 bit), assuming the set value = n, a timer B2 interrupt is generated at every níth occurrence of a timer B2 underflow that meets the condition selected by the INV00 bit. (Note) | 1 to 15 | WO |
| Nothing is assigned. When write, set to "0". When read, its content is indeterminate. | | —— |

Note : Use MOV instruction to write to this register.
If the INV01 bit = ì1î, make sure the TB2S bit also = ì0î (timer B2 count stopped) when writing to this register.
If the INV01 bit = ì0î, although this register can be written even when the TB2S bit = ì1î (timer B2 count start), do not write synchronously with a timer B2 underflow.

**Figure 12.3.4. IDB0 Register, IDB1Register, DTT Register, and ICCTB2 Register**

Timer Ai, Ai-1 register (i=1, 2, 4) (Note 1, Note 2, Note 3, Note 4, Note 5)

| Symbol | Address | After reset |
|---|---|---|
| TA1 | $0389_{16}$-$0388_{16}$ | $????_{16}$ |
| TA2 | $038B_{16}$-$038A_{16}$ | $????_{16}$ |
| TA4 | $038F_{16}$-$038E_{16}$ | $????_{16}$ |
| TA11 (Note6,7) | $0343_{16}$-$0342_{16}$ | $????_{16}$ |
| TA21 (Note6,7) | $0345_{16}$-$0344_{16}$ | $????_{16}$ |
| TA41 (Note6,7) | $0347_{16}$-$0346_{16}$ | $????_{16}$ |

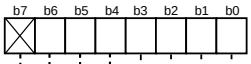



| Function | Setting range | RW |
|---|---|---|
| Assuming the set value = n, upon a start trigger the timer starts counting the count source and stops after counting it n times. The positive and negative phases change at the same time timer A, A2 or A4 stops. | $0000_{16}$ to $FFFF_{16}$ | WO |

Note 1: The register must be accessed in 16 bit units.
Note 2: When the timer Ai register is set to "$0000_{16}$", the counter does not operate and a timer Ai interrupt does not occur.
Note 3: Use MOV instruction to write to these registers.
Note 4: If the INV15 bit is "0" (dead time timer enable), the positive or negative phase whichever is going from an inactive to an active level changes at the same time the dead time timer stops.
Note 5: If the INV11 bit is "0" (three-phase mode 0), the TAi register value is transferred to the reload register by a timer Ai (i = 1, 2 or 4) start trigger.
If the INV11 bit is "1" (three-phase mode 1), the TAi1 register value is transferred to the reload register by a timer Ai start trigger first and then the TAi register value is transferred to the reload register by the next timer Ai start trigger. Thereafter, the TAi1 register and TAi register values are transferred to the reload register alternately.
Note 6: Do not write to TAi1 registers synchronously with a timer B2 underflow In three-phase mode 1, do not set TAi1 register while the timer B underflows.
Note 7: Write to the TAi1 register as follows:
(1) Write a value to the TAi1 register
(2) Wait for one cycle of timer Ai count source.
(3) Write the same value to the TAi1 register again.

**Figure 12.3.5. TA1, TA2, TA4, TA11, TA21 and TA41 Registers**

Timer B2 special mode register (Note 1)




Symbol TB2SC | Address 039E16 | After reset X0000000₂

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PWCOM | Timer B2 Reload Timing Switch Bit (Note 2) | 0 : Timer B2 underflow<br>1 : Timer A output at odd-numbered | RW |
| IVPCR1 | Three-Phase Output Port $\overline{SD}$ Control Bit 1 (Note 3, 4, 7) | 0 : Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) disabled<br>1 : Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) enabled | RW |
| TB0EN | Timer B0 Operation Mode Select Bit | 0 : Other than A-D trigger mode<br>1 : A-D trigger mode (Note 5) | RW |
| TB1EN | Timer B1 Operation Mode Select Bit | 0 : Other than A-D trigger mode<br>1 : A-D trigger mode (Note 5) | RW |
| TB2SEL | Trigger Select Bit (Note 6) | 0 : TB2 interrupt<br>1 : Underflow of TB2 interrupt generation frequency setting counter [ICTB2] | RW |
| $\overline{\text{(b6-b5)}}$ | Reserved bits | Must set to "0" | RW |
| — (b7) | Nothing is assigned. When write, set to 0 . When read, its content is 0 . | | — |

Note 1. Write to this register after setting the PRC1 bit in the PRCR register to "1" (write enabled).
Note 2. If the INV11 bit is "0" (three-phase mode 0) or the INV06 bit is "1" (triangular wave modulation mode), set this bit to "0" (timer B2 underflow).
Note 3. When setting the IVPCR1 bit to "1" (three-phase output forcible cutoff by $\overline{SD}$ pin input enabled), Set the PD8_5 bit to "0" (= input mode).
Note 4. Related pins are U(P8₀), $\overline{U}$(P8₁), V(P7₂), $\overline{V}$(P7₃), W(P7₄), $\overline{W}$(P7₅). After forcible cutoff, input "H" to the P8₅/$\overline{NMI}$/$\overline{SD}$ pin. Set the IVPCR1 bit to "0", and this forcible cutoff will be reset. If L is input to the P8₅/$\overline{NMI}$/$\overline{SD}$ pin, a three-phase motor control timer output will be disabled (INV03=0). At this time, when the IVPCR1 bit is "0", the target pins changes to programmable I/O port. When the IVPCR1 bit is "1", the target pins changes to high-impedance state regardless of which functions of those pins are used.
Note 5. When this bit is used in delayed trigger mode 0, set the TB0EN and TB1EN bits to "1" (A-D trigger mode).
Note 6. When setting the TB2SEL bit to "1" (underflow of TB2 interrupt generation frequency setting counter[ICTB2]), Set the INV02 bit to "1" (three-phase motor control timer function).
Note 7. Refer to "**17.6 Digital Debounce function**" for the $\overline{SD}$ input

The effect of P8₅/$\overline{NMI}$/$\overline{SD}$ pin input is below.
1.Case of INV03 = "1"(Three-phase motor control timer output enabled)

| IVPCR1 bit | P8₅/$\overline{NMT}$/$\overline{SD}$ pin inputs (Note 3) | status of U/V/W pins | Remarks |
|---|---|---|---|
| "1" (Three-phase output forcrible cutoff enable) | H | Three-phase PWM output | |
| | L(Note 1) | High impedance | Three-phase output forcrible cutoff |
| "0" (Three-phase output forcrible cutoff disable) | H | Three-phase PWM output | |
| | L(Note 1) | Input/output port(Note 2) | |

Note 1: When "L" is input to the P8₅/$\overline{NMI}$/$\overline{SD}$ pin, INV03 bit changes in "0" at the same time.
Note 2: The value of the port register and the port direction register becomes effective.
Note 3: When $\overline{SD}$ function isn't used, set to "0"(Input) in PD8₅ and pullup to "H" in P8₅/$\overline{NMI}$/$\overline{SD}$ pin from outside.

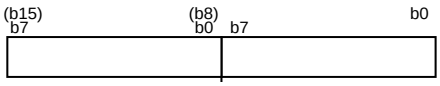
2.Case of INV03 = "0"(Three-phase motor control timer output disabled)

| IVPCR1 bit | P8₅/$\overline{NMT}$/$\overline{SD}$ pin inputs | status of U/V/W pins | Remarks |
|---|---|---|---|
| "1" (Three-phase output forcrible cutoff enable) | H | peripheral input/output or input/output port | |
| | L | High impedance | Three-phase output forcrible cutoff(Note 1) |
| "0" (Three-phase output forcrible cutoff disable) | H | peripheral input/output or input/output port | |
| | L | peripheral input/output or input/output port | |

Note 1: The three-phase output forcrible cutoff function becomes effective if the INPCR1 bit is "1" (three-phase output forcrible cutoff function enable) even when INV03 bit is "0"(three-phase motor control timer output disalbe)

**Figure 12.3.6. TB2SC Registers**

Timer B2 register (Note )

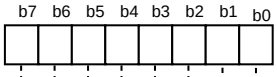



| Symbol | Address | After reset |
|---|---|---|
| TB2 | 0395$_{16}$-0394$_{16}$ | Indeterminate |

| Function | Setting range | RW |
|---|---|---|
| Divide the count source by n + 1 where n = set value. Timer A1, A2 and A4 are started at every occurrence of underflow. | 0000$_{16}$ to FFFF$_{16}$ | RW |

Note : The register must be accessed in 16 bit units.

Trigger select register

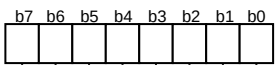



| Symbol | Address | After reset |
|---|---|---|
| TRGSR | 0383$_{16}$ | 00$_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TA1TGL | Timer A1 event/trigger select bit | To use the V-phase output control circuit, set these bits to “01$_2$”(TB2 underflow). | RW |
| TA1TGH | | | RW |
| TA2TGL | Timer A2 event/trigger select bit | To use the W-phase output control circuit, set these bits to “01$_2$”(TB2 underflow). | RW |
| TA2TGH | | | RW |
| TA3TGL | Timer A3 event/trigger select bit | b5 b4<br>0 0 : Input on TA3$_{IN}$ is selected (Note 1)<br>0 1 : TB2 overflow is selected (Note 2)<br>1 0 : TA2 overflow is selected (Note 2)<br>1 1 : TA4 overflow is selected (Note 2) | RW |
| TA3TGH | | | RW |
| TA4TGL | Timer A4 event/trigger select bit | To use the U-phase output control circuit, set these bits to “01$_2$”(TB2 underflow). | RW |
| TA4TGH | | | RW |

Note 1: Set the corresponding port direction bit to “0” (input mode).
Note 2: Overflow or underflow.

Count start flag




| Symbol | Address | After reset |
|---|---|---|
| TABSR | 0380$_{16}$ | 00$_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TA0S | Timer A0 count start flag | 0 : Stops counting<br>1 : Starts counting | RW |
| TA1S | Timer A1 count start flag | | RW |
| TA2S | Timer A2 count start flag | | RW |
| TA3S | Timer A3 count start flag | | RW |
| TA4S | Timer A4 count start flag | | RW |
| TB0S | Timer B0 count start flag | | RW |
| TB1S | Timer B1 count start flag | | RW |
| TB2S | Timer B2 count start flag | | RW |

**Figure 12.3.7. TB2 Register, TRGSR Register, and TABSR Register**

Timer Ai mode register

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | 0 | 1 | | 0 | 1 | 0 |

| Symbol | Address | After reset |
|---|---|---|
| TA1MR | $039716$ | $0016$ |
| TA2MR | $039816$ | $0016$ |
| TA4MR | $039A16$ | $0016$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | Must set to "$10_2$" (one-shot timer mode) for the three-phase motor control timer function | RW |
| TMOD1 | | | RW |
| MR0 | Pulse output function select bit | Must set to "0" for the three-phase motor control timer function | — |
| MR1 | External trigger select bit | Has no effect for the three-phase motor control timer function | RW |
| MR2 | Trigger select bit | Must set to "1" (selected by event/trigger select register) for the three-phase motor control timer function | RW |
| MR3 | Must set to "0" for the three-phase motor control timer function | | RW |
| TCK0 | Count source select bit | b7 b6<br>0 0 : $f_1$ or $f_2$<br>0 1 : $f_8$<br>1 0 : $f_{32}$<br>1 1 : $fc_{32}$ | RW |
| TCK1 | | | RW |

Timer B2 mode register

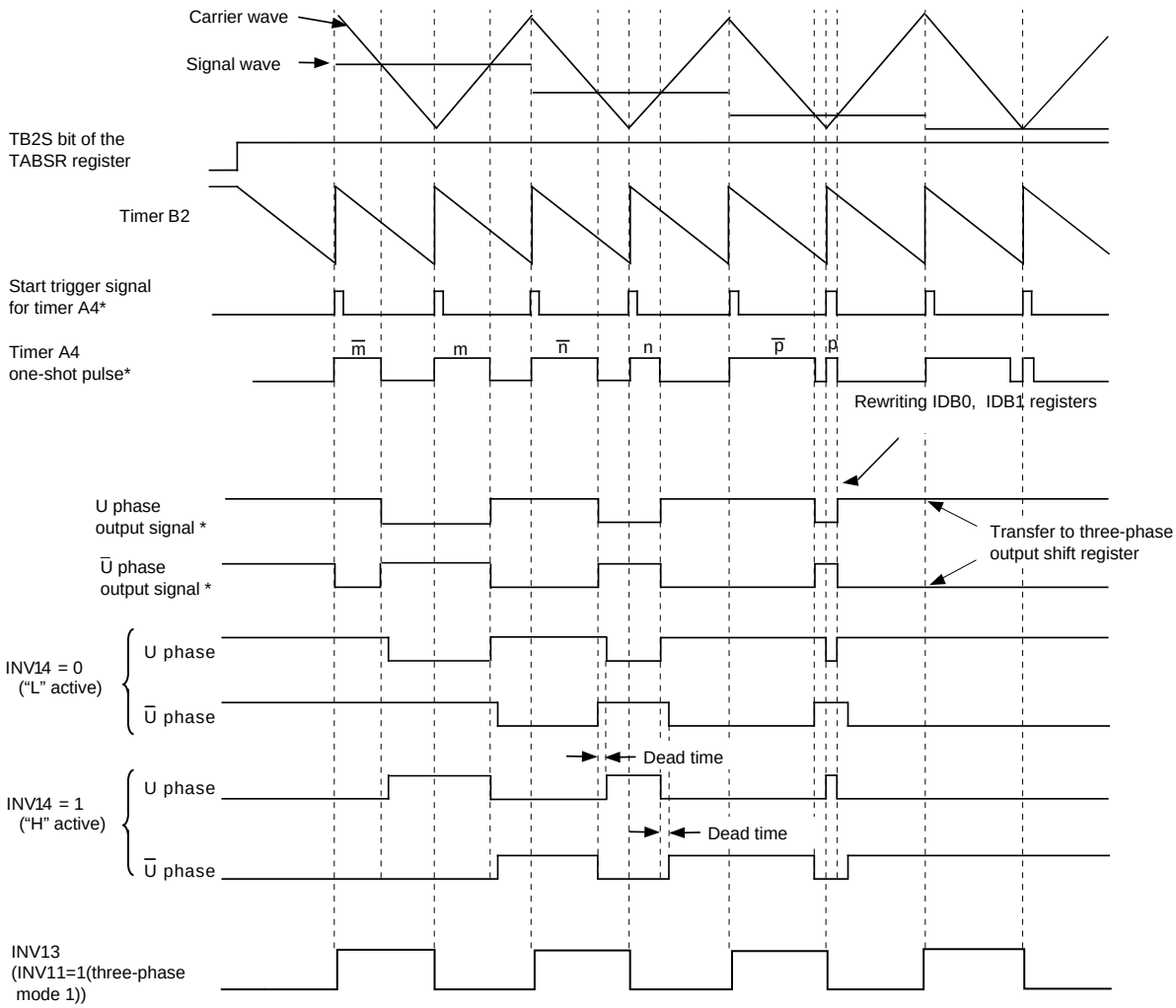
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | 0 | | | 0 | 0 |

| Symbol | Address | After reset |
|---|---|---|
| TB2MR | $039D16$ | $00XX0000_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | Set to "$00_2$" (timer mode) for the three-phase motor control timer function | RW |
| TMOD1 | | | RW |
| MR0 | Has no effect for the three-phase motor control timer function. When write, set to "0". When read, its content is indeterminate. | | RW |
| MR1 | | | RW |
| MR2 | Must set to "0" for the three-phase motor control timer function | | RW |
| MR3 | When write in three-phase motor control timer function, write "0". When read, its content is indeterminate. | | RO |
| TCK0 | Count source select bit | b7 b6<br>0 0 : $f_1$ or $f_2$<br>0 1 : $f_8$<br>1 0 : $f_{32}$<br>1 1 : $fc_{32}$ | RW |
| TCK1 | | | RW |

**Figure 12.3.8. TA1MR, TA2MR, TA4MR, and TB2MR Registers**

The three-phase motor control timer function is enabled by setting the INV02 bit of INVC0 register to “1”. When this function is on, timer B2 is used to control the carrier wave, and timers A4, A1 and A2 are used to control three-phase PWM outputs (U, $\overline{U}$, V, $\overline{V}$, W and $\overline{W}$). The dead time is controlled by a dedicated dead-time timer. Figure 12.3.9 shows the example of triangular modulation waveform, and Figure 12.3.10 shows the example of sawtooth modulation waveform.




* Internal signals. See the block diagram of the three-phase motor control timer function.

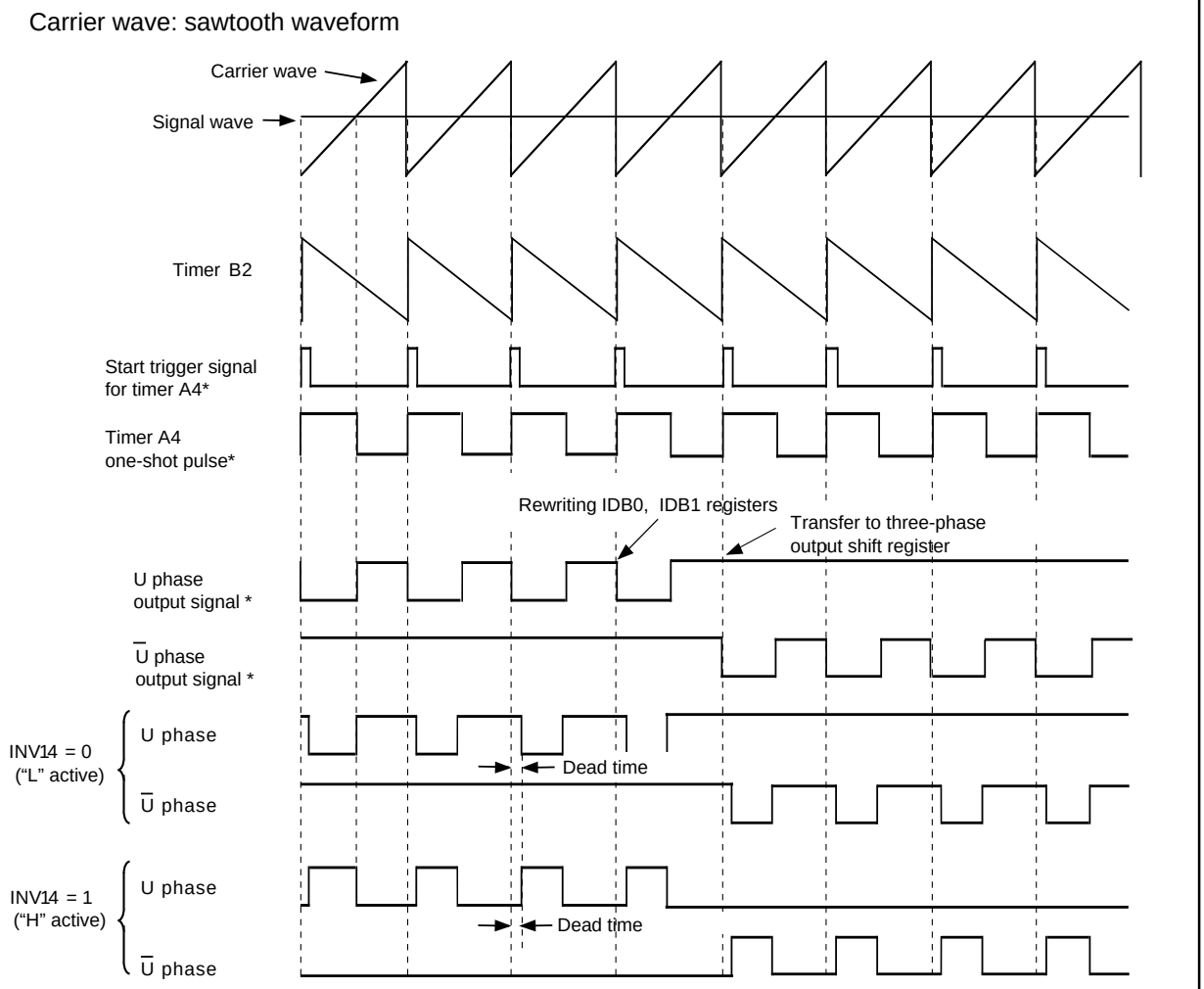
Shown here is a typical waveform for the case where INVC0 = 00XX11XX$_2$ (X = set as suitable for the system) and INVC1 = 010XXXX0$_2$.
An example for changing PWM outputs is shown below.

(1)When INV11=1(three-phase mode 1)
- INV01=0, ICTB2=2$_{16}$(timer B2 interrupt is generated at every 2'th occurrence of a timer B2 underflow), or INV01=1, INV00=1, ICTB2=1$_{16}$(timer B2 interrupt is generated at even-numbered occurrences of a timer B2 underflow).
- Initial timer value: TA41=$\overline{m}$, TA4=m. The TA4 and TA41 registers are modified every time a timer B2 interrupt occurs. First time, TA41= $\overline{n}$, TA4 = n. Second time, TA41 = $\overline{p}$, TA4 = p.
- Initial values of IDB0 and IDB1 registers: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1.The register values are changed to DU0 = 1, DUB0 = 0, DU1= 1 and DUB1 = 0 the third time a timer B2 interrupt occurs.

(2)When INV11=0(three-phase mode 0)
- INV01=0, ICTB2=1$_{16}$(timer B2 interrupt is generated at every occurrence of a timer B2 underflow)
- Initial timer value: TA4 = $\overline{m}$. The TA4 register is modified each time a timer B2 interrupt occurs. First time, TA4 = m. Second time, TA4 = $\overline{n}$. Third time, TA4 = n. Fourth time, TA4 = $\overline{p}$. Fifth time, TA4 = p.
- Initial values of IDB0 and IDB1 registers: DU0=1, DUB0=0, DU1=0, DUB1=1.The register values are changed to DU0 = 1, DUB0 = 0, DU1= 1 and DUB1 = 0 the sixth time a timer B2 interrupt occurs.

The value written to the TA4 register and TA41 register are inverted at odd-numbered timer A outputs.

**Figure 12.3.9. Triangular Wave Modulation Operation**




* Internal signals. See the block diagram of the three-phase motor control timer function.

Shown here is a typical waveform for the case where INVC0= 01XX110X2 (X = set as suitable for the system) and INVC1 = 010XXX002. An example for changing PWM outputs is shown below.
• Initial values of IDB0 and IDB1 registers: DU0=0, DUB0=1, DU1=1, DUB1=1. The register values are changed to DU0=1, DUB0=0, DU1=1, DUB1=1 a timer B2 interrupt occurs.

**Figure 12.3.10. Sawtooth Wave Modulation Operation**

## 12.3.1 Position-data-retain Function

This function is used to retain the position data synchronously with the three-phase waveform output.There are three position-data input pins for U, V, and W phases.

A trigger to retain the position data (hereafter, this trigger is referred to as "retain trigger") can be selected by the retain-trigger polarity select bit(bit 3 of the position-data-retain function control register, at address 034E16). This bit selects the retain trigger to be the falling edge of each positive phase, or the rising edge of each positive phase.

### 12.3.1.1 Operation of the Position-data-retain Function

Figure 12.3.1.1.1 shows a usage example of the position-data-retain function (U phase) when the retain trigger is selected as the falling edge of the positive signal.

(1) At the falling edge of the U-phase waveform ouput, the state at pin IDU is transferred to the U-phase position data retain bit ( bit2 at address 034E16 ).

(2) Until the next falling edge of the Uphase waveform output,the above value is retained.

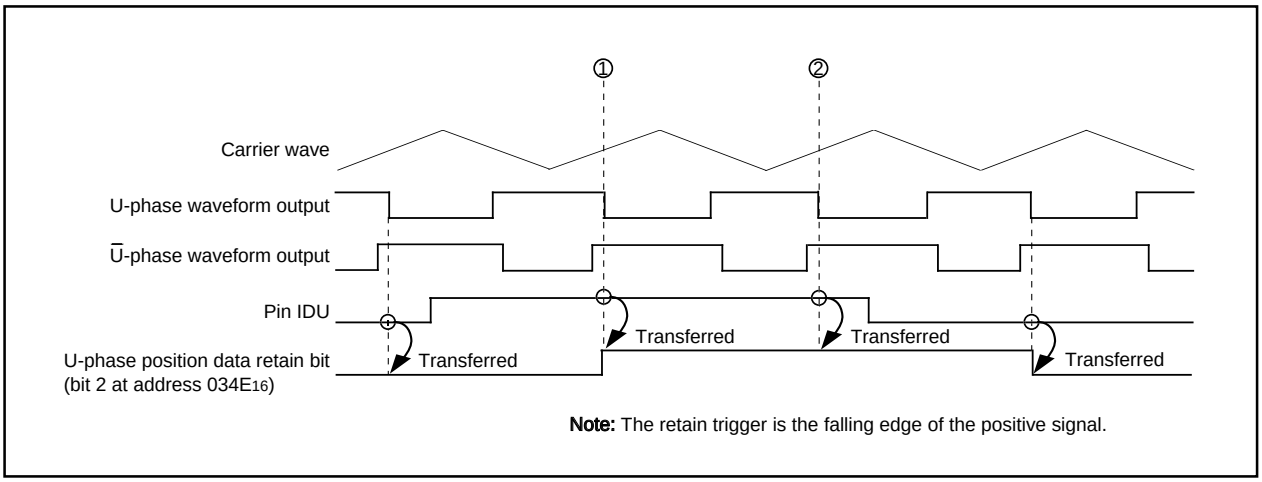



**Figure 12.3.1.1.1 Usage Example of Position-data-retain Function ( U phase )**

### 12.3.1.2 Position-data-retain Function Control Register

Figure 12.3.1.2.1 shows the structure of the position-data-retain function contol register.

Position-data-retain function control register (Note)

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

| Symbol | Address | When reset |
|---|---|---|
| PDRF | $034E_{16}$ | $XXXX\ 0000_2$ |

| Bit | Bit name | Function | RW |
|---|---|---|---|
| PDRW | W-phase position data retain bit | Input level at pin IDW is read out.<br>0: "L" level<br>1: "H" level | RO |
| PDRV | V-phase position data retain bit | Input level at pin IDV is read out.<br>0: "L" level<br>1: "H" level | RO |
| PDRU | U-phase position data retain bit | Input level at pin IDU is read out.<br>0: "L" level<br>1: "H" level | RO |
| PDRT | Retain-trigger polarity select bit | 0: Rising edge of positive phase<br>1: Falling edge of positive phase | RW |
| (b7-b4) | Nothing is assigned. When write, set to ì0î. When read, contents are indeterminate. | | — |

Note: This register is valid only in the three-phase mode.

**Figure 12.3.1.2.1. Structure of Position-data-retain Function Control Register**

#### 12.3.1.2.1 W-phase Position Data Retain Bit (PDRW)

**This bit is used to retain the input level at pin IDW.**

#### 12.3.1.2.2 V-phase Position Data Retain Bit (PDRV)

**This bit is used to retain the input level at pin IDV.**

#### 12.3.1.2.3 U-phase Position Data Retain Bit (PDRU)

**This bit is used to retain the input level at pin IDU.**

#### 12.3.1.2.4 Retain-trigger Polarity Select Bit (PDRT)

This bit is used to select the trigger polarity to retain the position data.

When this bit = "0", the rising edge of each positive phase selected.

When this bit = "1", the falling edge of each pocitive phase selected.

# 13. Timer S (Input Capture/Output Compare)

The Timer S (Input Capture/Output Compare : here after, Timer S is referred to as "IC/OC".) is a multi-functional I/O port for time measurement and waveform generation. Each channel of the IC/OC module provides the capability for time measurement, by input capture, and also provides the capability for waveform generation, by output comparison.

The IC/OC consists of one 16-bit base timer for free-running operation, as well as eight 16-bit registers for time measurement and waveform generation.

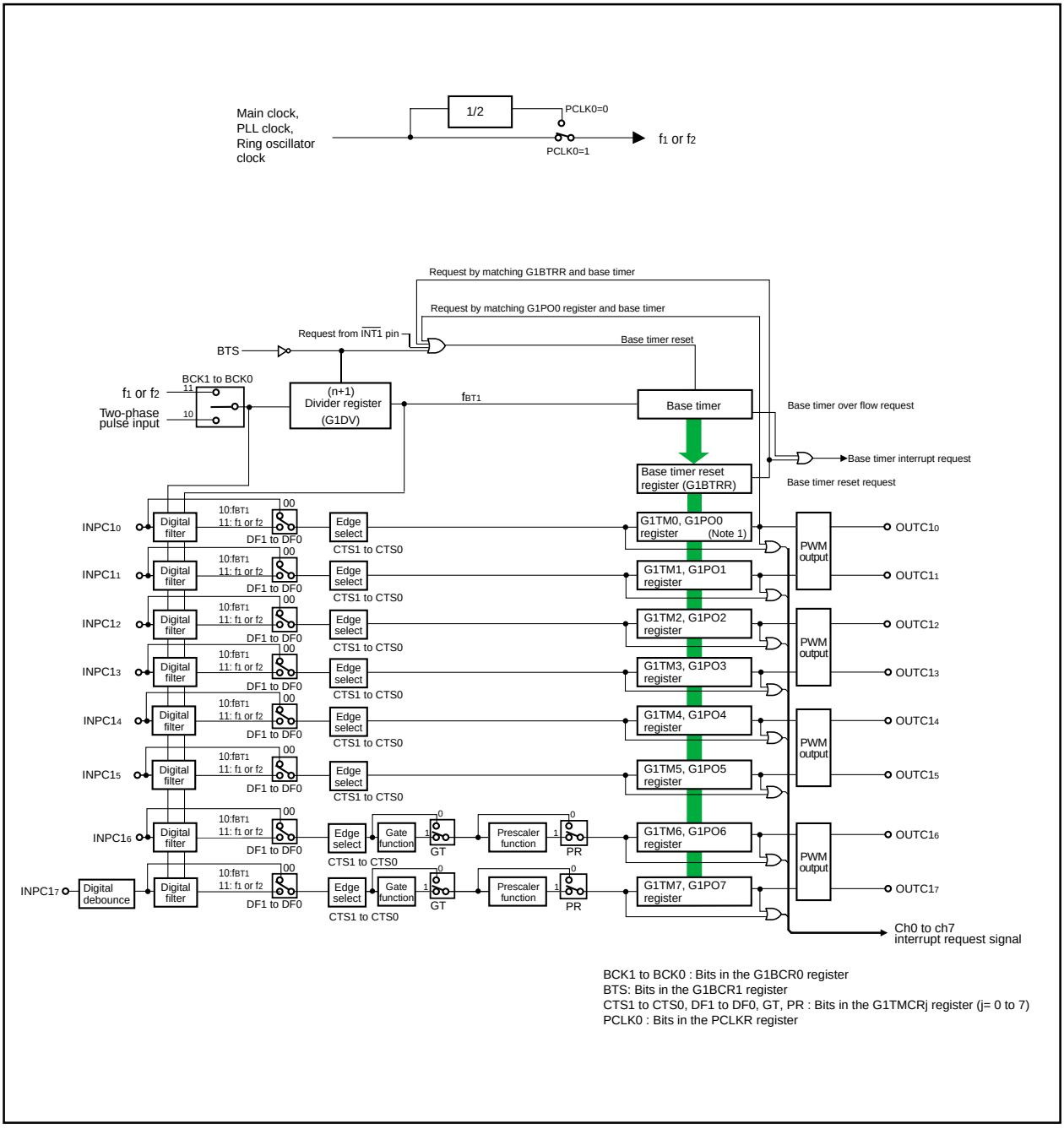
Table 13.1 lists functions and channels of the IC/OC.

**Table 13.1. IC/OC Functions and Channels**

| Function | |
|---|---|
| Time measurement (Note 1) | 8 channels |
| Digital filter | 8 channels |
| Trigger input prescaler | 2 channels |
| Trigger input gate | 2 channels |
| Waveform generation (Note 1) | 8 channels |
| Single-phase waveform output | Available |
| Phase-delayed waveform output | Available |
| Set/Reset waveform output | Available |

Notes 1 : The time measurement function shares pins with the waveform generation function.
The time measurement function or waveform generation function can be selected for each channel.

Figure 13.1 shows the block diagram of the IC/OC.




Figures 13.1. IC/OC Block Diagram

Figures 13.2 to 13.11 show registers associated with the IC/OC base timer, the time measurement function, and the waveform generation function.

## Base timer register (Note 1)

b15 (b7) b8 (b0) b7 b0

| Symbol | Address | When reset |
|---|---|---|
| G1BT | $0321_{16}$ - $0320_{16}$ | $????_{16}$ |

| Function | Setting range | RW |
|---|---|---|
| • When the base timer is operating: When read, value of the counter can be read. When write, the counter starts counting from the value written. When the base timer is reset, this register is set to "$0000_{16}$". (Note 2)<br>• When the base timer is reset: This register is set to "$0000_{16}$" but a value read is indeterminate. No value is written. (Note 2) | $0000_{16}$ to $FFFF_{16}$ | RW |

Note 1: The value which is written in this register is reflected synchronizing with the base timer count source $f_{BT1}$.
Note 2: This base timer stops only when the BCK1 to BCK0 bits in the G1BCR0 register are set to "$00_2$" (count source clock stop). This base timer operates when the BCK1 to BCK0 bits are set to other than "$00_2$". When the BTS bit in the G1BCR1 register is set to "0", the base timer continues to be held in reset, and remains in a no counting state with a value of "$0000_{16}$". When the BTS bit in the G1BCR1 register is set to "1", this state is cleared and the timer starts counting.

## Base timer control register 0

b7 b6 b5 b4 b3 b2 b1 b0 (b5 = 0, b4 = 0, b3 = 0)

| Symbol | Address | When reset |
|---|---|---|
| G1BCR0 | $0322_{16}$ | 0000 $0000_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| BCK0 | Count source select bit | b1b0<br>0 0: Clock stop<br>0 1: Avoid this setting<br>1 0: Two-phase input (Note 1)<br>1 1: $f_1$ or $f_2$ (Note 2) | RW |
| BCK1 | | | RW |
| RST4 | Base timer reset cause select bit 4 | 0: Base timer not reset by matching G1BTRR<br>1: Base timer reset by matching G1BTRR | RW |
| (b5-b3) | Reserved bit | Should be set to "0" | RW |
| CH7INSEL | Channel 7 Input select bit | 0: $P2_7$/OUTC$1_7$/INPC$1_7$ pin<br>1: $P1_7$/$\overline{INT5}$/INPC$1_7$/IDU pin | RW |
| IT | Base timer overflow select bit | 0: Bit 15 overflow<br>1: Bit 14 overflow | RW |

Note 1: This setting can be used when the UD1 to UD0 bits in the G1BCR1 register are set to "$10_2$" (two-phase signal processing mode). Avoid setting the BCK1 to BCK0 bits to "$10_2$" in other modes.
Note 2: When the PCLK0 bit in the PCLKR register is set to "0", the Count source is $f_2$. And when this bit is set to "1", the Count source is $f_1$.

**Figure 13.2. G1BT and G1BCR0 Registers**

Divider register

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

| Symbol | Address | When reset |
|---|---|---|
| G1DV | $032A_{16}$ | $00_{16}$ |

| Function | Setting range | RW |
|---|---|---|
| Divide ratio of $f_1$, $f_2$ or two pulse input by (n+1) for $f_{BT1}$.<br>n: this register detemines | $00_{16}$ to $FF_{16}$ | RW |

**Figure 13.3. G1DV Register**

## Base timer control register 1

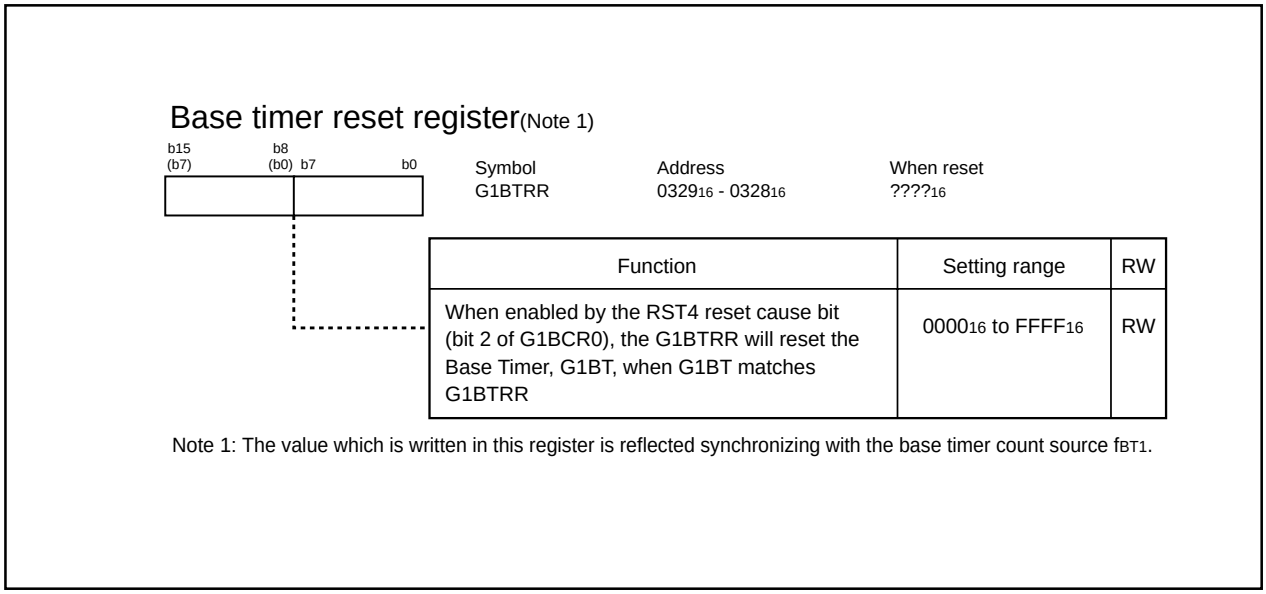
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | | | | 0 | | | 0 |

| Symbol | Address | When reset |
|---|---|---|
| G1BCR1 | $0323_{16}$ | $0000\ 0000_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| ——— (b0) | Reserved bit | Should be set to "0". | RW |
| RST1 | Base timer reset cause select bit 1 | 0: The base timer is not reset by matching the G1PO0 register<br>1: The base timer is reset by matching the G1PO0 register (Note 1) | RW |
| RST2 | Base timer reset cause select bit 2 | 0: The base timer is not reset when an input to the $\overline{INT1}$ pin is "L" level<br>1: The base timer is reset when an input to the $\overline{INT1}$ pin is "L" level | RW |
| ——— (b3) | Reserved bit | Should set to "0". | RW |
| BTS | Base timer start bit | 0: Base timer is reset<br>1: Base timer starts counting | RW |
| UD0 | Counter increment/ decrement control bit | b6b5<br>0 0 : Counter increment mode<br>0 1 : Counter increment/decrement mode<br>1 0 : Two-phase pulse signal processing mode<br>1 1 : Avoid this setting | RW |
| UD1 | | | RW |
| ——— (b7) | Reserved bit | Should set to "0". | RW |

Note 1: The base timer is reset after two clock cycles of $f_{BT1}$ when it matches the G1PO0 register. (See Figure 13.8 about the G1PO0 register.) When setting the RST1 bit to "1", values of the G1POj register (j=1 to7) to use for the waveform generation function should be set to smaller value than values of the G1PO0 register.

**Figure 13.4. G1BCR1 Register**



Base timer reset register(Note 1)

b15 (b7) b8 (b0) b7 b0

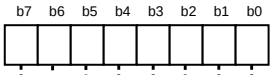
| Symbol | Address | When reset |
|---|---|---|
| G1BTRR | $0329_{16}$ - $0328_{16}$ | $????_{16}$ |

| Function | Setting range | RW |
|---|---|---|
| When enabled by the RST4 reset cause bit (bit 2 of G1BCR0), the G1BTRR will reset the Base Timer, G1BT, when G1BT matches G1BTRR | $0000_{16}$ to $FFFF_{16}$ | RW |

Note 1: The value which is written in this register is reflected synchronizing with the base timer count source $f_{BT1}$.

**Figure 13.5. G1BTRR Register**

## Time measurement control register j (j=0 to 7)

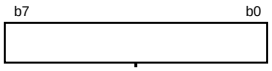



| Symbol | Address | When reset |
|---|---|---|
| G1TMCR0 to G1TMCR3 | $0318_{16}$, $0319_{16}$, $031A_{16}$, $031B_{16}$ | $0000\ 0000_2$ |
| G1TMCR4 to G1TMCR7 | $031C_{16}$, $031D_{16}$, $031E_{16}$, $031F_{16}$ | $0000\ 0000_2$ |

| 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 function select bit | b3 b2<br>0 0 : No digital filter<br>0 1 : Avoid this setting<br>1 0 : $f_{BT1}$<br>1 1 : $f_1$ or $f_2$ (Note 1) | RW |
| DF1 | | | RW |
| GT | Gate function select bit (Note 2) | 0 : Gate function not used<br>1 : Gate function used | RW |
| GOC | Gate function clear select bit (Notes 2, 3, 4) | 0 : Not cleared<br>1 : The gate is cleared when the base timer matches the G1POk register | RW |
| GSC | Gate function clear bit (Notes 2, 3) | When setting the GSC bit to "1", the gate is cleared. | RW |
| PR | Prescaler function select bit (Note 2) | 0 : Not used<br>1 : Used | RW |

Notes :
1. When the PCLK0 bit in the PCLKR register is set to "0", the Count source is $f_2$. And when this bit is set to "1", the Count source is $f_1$.
2. This bit is in the G1TMCR6 and G1TMCR7 registers.
   All bits 4 to 7 in the G1TMCR0 to G1TMCR5 registers should be set to "0".
3. These bits are available when setting the GT bit to "1".
3. The GOC bit is set to "0" after the gate function is cleared. See Figure 13.8 about the G1POk register (k=4 when j=6 and k=5 when j=7).

## Time measurement prescale register j (j=6,7)(Note 1)



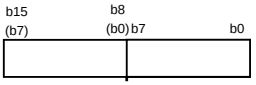

| Symbol | Address | When reset |
|---|---|---|
| G1TPR6 to G1TPR7 | $0324_{16}$, $0325_{16}$ | $0000\ 0000_2$ |

| Function | Setting range | RW |
|---|---|---|
| As the setting value is n, time is measured whenever a trigger input is counted by n+1 (Note 2) | $00_{16}$ to $FF_{16}$ | RW |

Note 1: The value which is written in this register is reflected synchronizing with the base timer count source $f_{BT1}$.
Note 2: The first prescaler, after the PR bit in the G1TMCRj register changes "0" (prescaler function unused) to "1" (prescaler function used), may be divided by n, not by n+1. Subsequent prescalers

**Figure 13.6. G1TMCR0 to G1TMCR7 Registers, and G1TPR6 to G1TPR7 Registers**

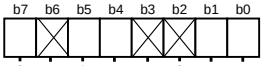
## Time measurement register j (j=0 to 7)




| Symbol | Address | When reset |
|---|---|---|
| G1TM0 to G1TM2 | $0301_{16}$ - $0300_{16}$, $0303_{16}$ - $0302_{16}$, $0305_{16}$ - $0304_{16}$ | $????_{16}$ |
| G1TM3 to G1TM5 | $0307_{16}$ - $0306_{16}$, $0309_{16}$ - $0308_{16}$, $030B_{16}$ - $030A_{16}$ | $????_{16}$ |
| G1TM6 to G1TM7 | $030D_{16}$ - $030C_{16}$, $030F_{16}$ - $030E_{16}$ | $????_{16}$ |

| Function | Setting range | RW |
|---|---|---|
| Value of the base timer is stored every time measurement. | ——— | RO |

## Waveform generation control register j (j=0 to 7)



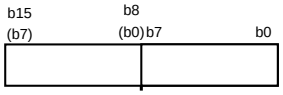

| Symbol | Address | When reset |
|---|---|---|
| G1POCR0 to G1POCR3 | $0310_{16}$, $0311_{16}$, $0312_{16}$, $0313_{16}$ | $0X00\ XX00_2$ |
| G1POCR4 to G1POCR7 | $0314_{16}$, $0315_{16}$, $0316_{16}$, $0317_{16}$ | $0X00\ XX00_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| MOD0 | Operation mode select bit | b1b0<br>00: Single waveform output mode<br>01: SR waveform output mode (Note 1)<br>10: Phase-delayed waveform output mode<br>11: Avoid this setting | RW |
| MOD1 | | | RW |
| ——— (b3-b2) | Nothing is assigned. When write, should set to "0". When read, its content is indeterminate. | | — |
| IVL | Output initial value select bit | 0: Outputs "L" as an initial value<br>1: Outputs "H" as an initial value | RW |
| RLD | GiPOj register value reload timing select bit | 0: Reloads the G1POj register when the CPU writes to a counter<br>1: Reloads the G1POj register when the base timer is reset | RW |
| ——— (b6) | Nothing is assigned. When write, should set to "0". When read, its content is indeterminate. | | — |
| INV | Inverse output function select bit (Note 2) | 0: Output is not inversed<br>1: Output is inversed | RW |

Notes :
1. This setting is enabled only on even channels. In SR waveform output mode, the values written to the corresponding odd channel (next channel after an even channel) are ignored. An even channel outputs waveform. An odd channel outputs no waveform.
2. The inverse output function is performed as a final step on a process of waveform generation. When setting the INV bit to "1" (output inverse), "H" is output with setting the IVL bit to "0" (output "L" as an initial value) and "L" with setting the IVL bit to "1" (output "H" as an initial value).

**Figure 13.7. G1TM0 to G1TM7 Registers, and G1POCR0 to G1POCR7 Registers**

## Waveform generation register j (j=0 to 7)

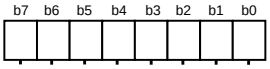



| Symbol | Address | When reset |
|---|---|---|
| G1PO0 to G1PO2 | $0301_{16}$-$0300_{16}$, $0303_{16}$-$0302_{16}$, $0305_{16}$-$0304_{16}$ | $????_{16}$ |
| G1PO3 to G1PO5 | $0307_{16}$-$0306_{16}$, $0309_{16}$-$0308_{16}$, $030B_{16}$-$030A_{16}$ | $????_{16}$ |
| G1PO6 to G1PO7 | $030D_{16}$-$030C_{16}$, $030F_{16}$-$030E_{16}$ | $????_{16}$ |

| Function | Setting range | RW |
|---|---|---|
| • When the RLD bit in the G1POCRj register should be set to "0"<br>Value is reloaded into the G1POj register to include it in output waveform, etc.<br>• When the RLD bit should be set to "1"<br>Value is reloaded when the base timer is reset.<br>Value written can be read between writing the value and reloaded. | $0000_{16}$ to $FFFF_{16}$ | RW |

**Figure 13.8. G1PO0 to G1PO7 Registers**

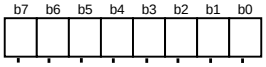
## Function select register




| Symbol | Address | When reset |
|---|---|---|
| G1FS | $0327_{16}$ | $0000\ 0000_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| FSC0 | Channel 0 time measurement/waveform generation function select bit | 0 : Select the waveform generation function<br>1 : Select the time measurement function | RW |
| FSC1 | Channel 1 time measurement/waveform generation function select bit | | RW |
| FSC2 | Channel 2 time measurement/waveform generation function select bit | | RW |
| FSC3 | Channel 3 time measurement/waveform generation function select bit | | RW |
| FSC4 | Channel 4 time measurement/waveform generation function select bit | | RW |
| FSC5 | Channel 5 time measurement/waveform generation function select bit | | RW |
| FSC6 | Channel 6 time measurement/waveform generation function select bit | | RW |
| FSC7 | Channel 7 time measurement/waveform generation function select bit | | RW |

## Function enable register(Note 1)




| Symbol | Address | When reset |
|---|---|---|
| G1FE | $0326_{16}$ | $0000\ 0000_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| IFE0 | Channel 0 function enable bit | 0 : Channel functions disabled<br>1 : Channel functions enabled | 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 |

Note 1: The value which is written in this register is reflected synchronizing with the base timer count source $f_{BT1}$.

**Figure 13.9. G1FS and G1FE Registers**

## Interrupt request register (Notes 1, 2)

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

| Symbol | Address | When reset |
|---|---|---|
| G1IR | $0330_{16}$ | ???? ????$_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| G1IR0 | Interrupt Request, Ch0 | 0 : No Request<br>1 : Interrupt Requested | RW |
| G1IR1 | Interrupt Request, Ch1 | | RW |
| G1IR2 | Interrupt Request, Ch2 | | RW |
| G1IR3 | Interrupt Request, Ch3 | | RW |
| G1IR4 | Interrupt Request, Ch4 | | RW |
| G1IR5 | Interrupt Request, Ch5 | | RW |
| G1IR6 | Interrupt Request, Ch6 | | RW |
| G1IR7 | Interrupt Request, Ch7 | | RW |

Notes:
1. Interrupt Request for Base Timer is latched by the soft interrupt 7 ICR at address 0047h.
2. Each G1IRi (i = 0 to 7) is ANDed with its corresponding G1IEx (x = 0 to 1) to generate soft interrupt 5 or soft interrupt 6.

**Figure 13.10. G1IR Register**

## Interrupt enable register 0

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

| Symbol | Address | When reset |
|---|---|---|
| G1IE0 | $0331_{16}$ | $0000\ 0000_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| G1IE00 | Interrupt Enable 0, CH0 | 0 : IC/OC interrupt 0 request disable<br>1 : IC/OC interrupt 0 request enable | RW |
| G1IE01 | Interrupt Enable 0, CH1 | | RW |
| G1IE02 | Interrupt Enable 0, CH2 | | RW |
| G1IE03 | Interrupt Enable 0, CH3 | | RW |
| G1IE04 | Interrupt Enable 0, CH4 | | RW |
| G1IE05 | Interrupt Enable 0, CH5 | | RW |
| G1IE06 | Interrupt Enable 0, CH6 | | RW |
| G1IE07 | Interrupt Enable 0, CH7 | | RW |

## Interrupt enable register 1

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

| Symbol | Address | When reset |
|---|---|---|
| G1IE1 | $0332_{16}$ | $0000\ 0000_2$ |

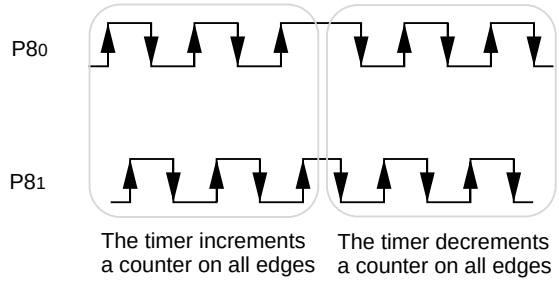
| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| G1IE10 | Interrupt Enable 1, CH0 | 0 : IC/OC interrupt 1 request disable<br>1 : IC/OC interrupt 1 request enable | RW |
| G1IE11 | Interrupt Enable 1, CH1 | | RW |
| G1IE12 | Interrupt Enable 1, CH2 | | RW |
| G1IE13 | Interrupt Enable 1, CH3 | | RW |
| G1IE14 | Interrupt Enable 1, CH4 | | RW |
| G1IE15 | Interrupt Enable 1, CH5 | | RW |
| G1IE16 | Interrupt Enable 1, CH6 | | RW |
| G1IE17 | Interrupt Enable 1, CH7 | | RW |

**Figure 13.11. G1IE0 and G1IE1 Registers**

## 13.1 Base Timer

The base timer counts an internally generated count source with free-running.
Table 13.1.1 lists specifications of the base timer. Table 13.1.2 shows registers associated with the base timer. Figure 13.1.1 shows a block diagram of the base timer. Figure 13.1.2 shows an example of the base timer in counter increment mode. Figure 13.1.3 shows an example of the base timer in counter increment/decrement mode. Figure 13.1.4 shows an example of two-phase pulse signal processing mode.

**Table 13.1.1. Base Timer Specifications**

| Item | Specification |
|---|---|
| Count source($f_{BT1}$) | $f_1$ or $f_2$ divided by (n+1) , two pulse input divided by (n+1)<br>n: The DIV7 to DIV0 bits in the G1DV register determines. n=0 to 255<br>In $f_1$ and two pulse input when n=0, a count source is not divided |
| Counting operation | The base timer increments the counter<br>The base timer increments/decrements the counter (see the selectable function)<br>Two-phase pulse processing (see the selectable function) |
| Count start condition | • The base timer starts counting when the BTS bit in the G1BCR1 register is set to "1" |
| Count stop condition | The BTS bit in the G1BCR1 register is set to "0" (base timer reset) |
| Base timer reset condition | (1) Value of the base timer matches value of the G1BTRR register<br>(2) Value of the base timer matches value of G1PO0 register.<br>(3) Apply a low-level signal ("L") to external interrupt pin,$\overline{INT1}$ pin |
| Value for base timer reset | "$0000_{16}$" |
| Interrupt request | The base timer interrupt request is asserten:<br>(1) At bit 14 or bit 15 is overflow of the base timer<br>(2) Base timer value matches the base timer reset register, and the base timer reset is enable (See Figure 13.1.1.) |
| Read from timer | • While the base timer is running,the G1BT register indicates a counter value<br>• When the base timer is reset, a counter value is indeterminate |
| Write to timer | When a value is written while the base timer is running, the value written is counted first. No value can be written while the base timer is reset. |
| Selectable function | • Counter increment/decrement mode<br>The base timer starts counting in increment mode until reaching the maximum count value. Then the base timer starts in decrement mode until reaching next $0000_{16}$. (See Figure 13.1.3.)<br>• Two-phase pulse processing mode.<br>Two-phase pulses from $P8_0$ and $P8_1$ pins are counted (See Figure 13.1.4.)<br>  |

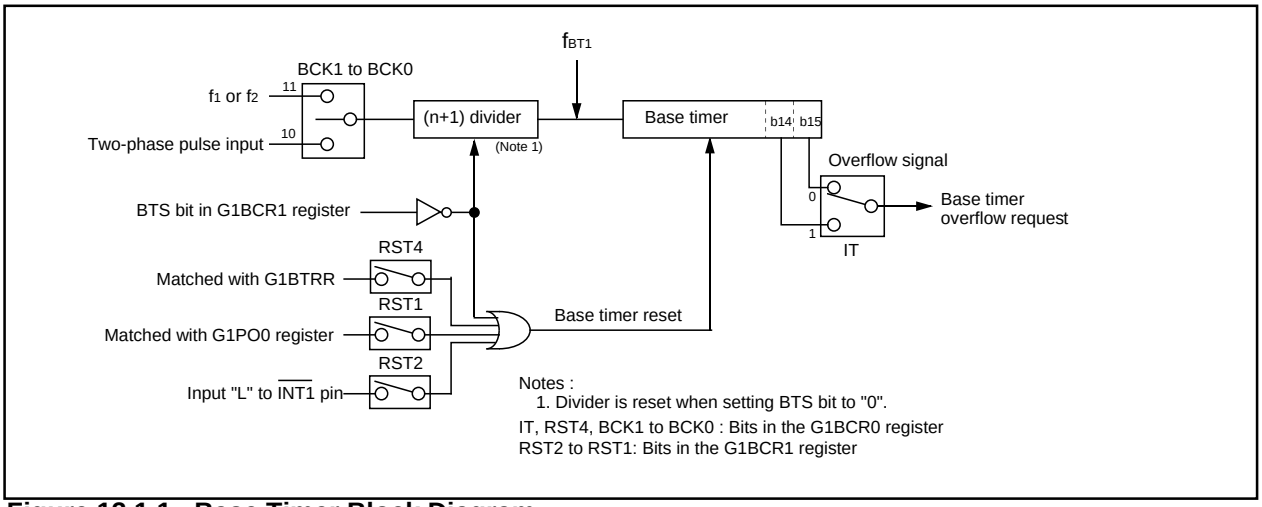


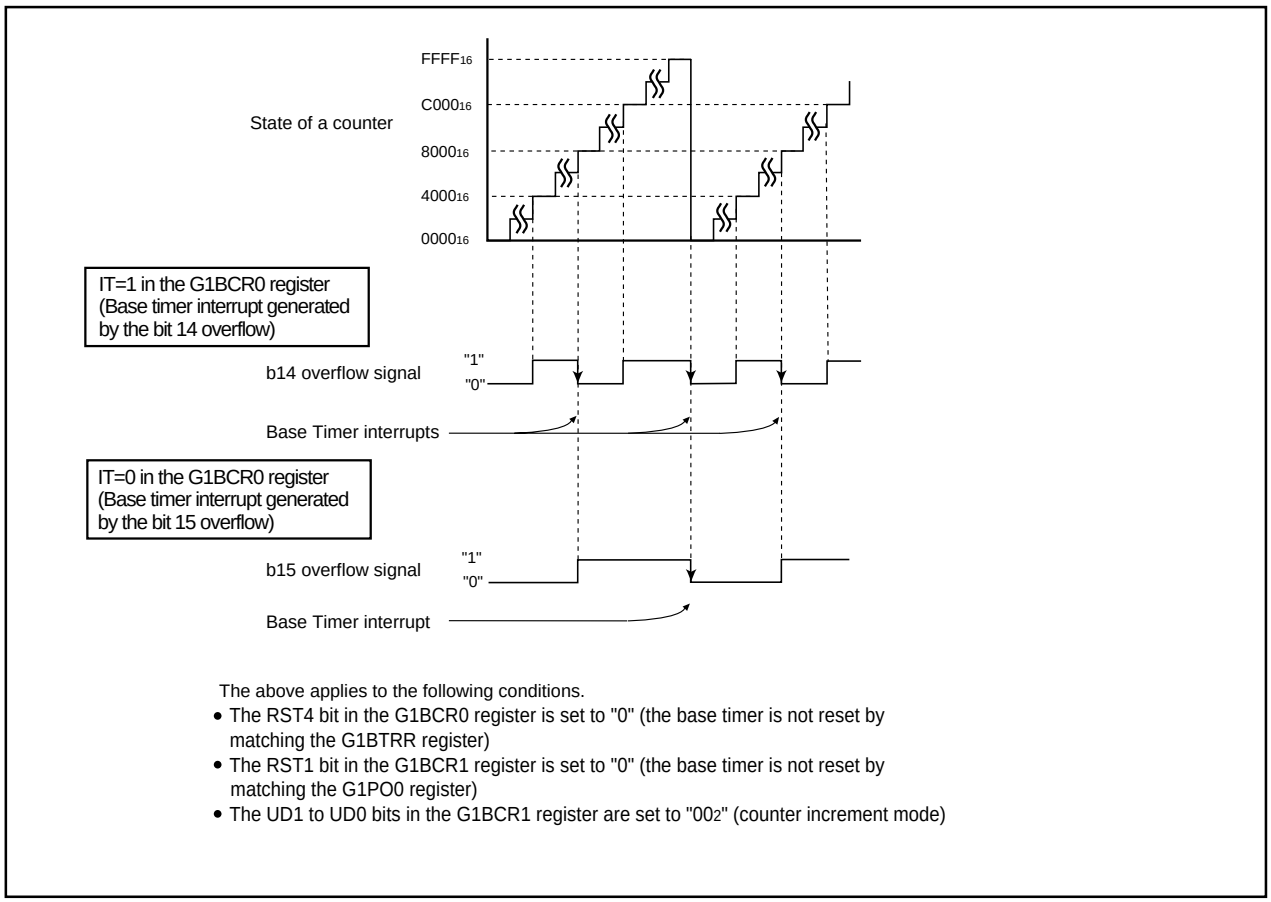

**Figure 13.1.1. Base Timer Block Diagram**

**Table 13.1.2. Base Timer Associated Register Settings (Time Measurement Function, Waveform Generation Function, Communication Function)**

| Register | Bit | Function |
|---|---|---|
| G1BCR0 | BCK1 to BCK0 | Select a count source |
| | RST4 | Select base timer reset timing |
| | IT | Select the base timer overflow |
| G1BCR1 | RST2 to RST1 | Select base timer reset timing |
| | BTS | Used when starting the base timer |
| | UD1 to UD0 | Select how to count |
| G1BT | - | Base timer value to read or to write |
| G1DV | - | Divide ratio of a count source |

When setting the RST1 bit to "1" (base timer reset when base timer matches G1PO0), the following registers require to be setup.

| | | |
|---|---|---|
| G1POCR0 | MOD1 to MOD0 | Set to "00$_2$" (single-phase waveform output mode) |
| G1PO0 | - | Set reset cycle |
| G1FS | FSC0 | Set to "0" (waveform generation function) |
| G1FE | IFE0 | Set to "1" (channel operation start) |




The above applies to the following conditions.

- The RST4 bit in the G1BCR0 register is set to "0" (the base timer is not reset by matching the G1BTRR register)
- The RST1 bit in the G1BCR1 register is set to "0" (the base timer is not reset by matching the G1PO0 register)
- The UD1 to UD0 bits in the G1BCR1 register are set to "00$_2$" (counter increment mode)

**Figure 13.1.2. Counter Increment Mode**

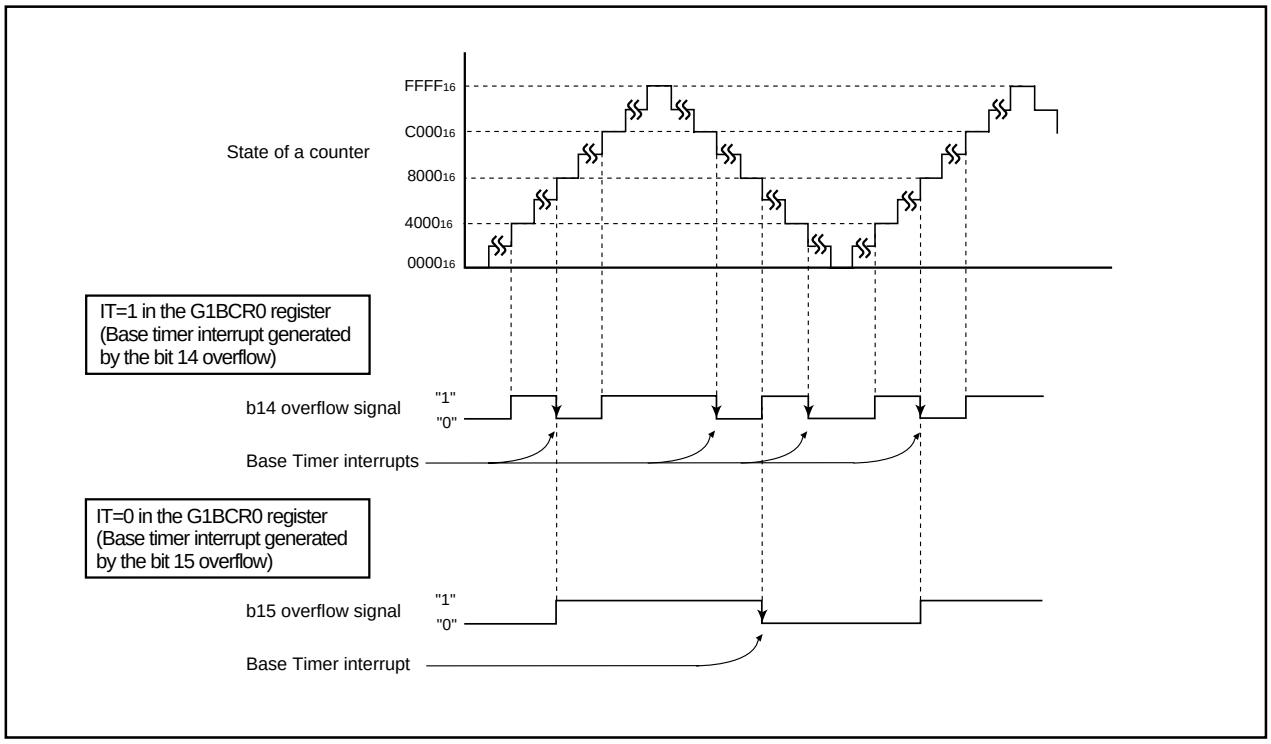



**Figure 13.1.3. Counter Increment/Decrement Mode**

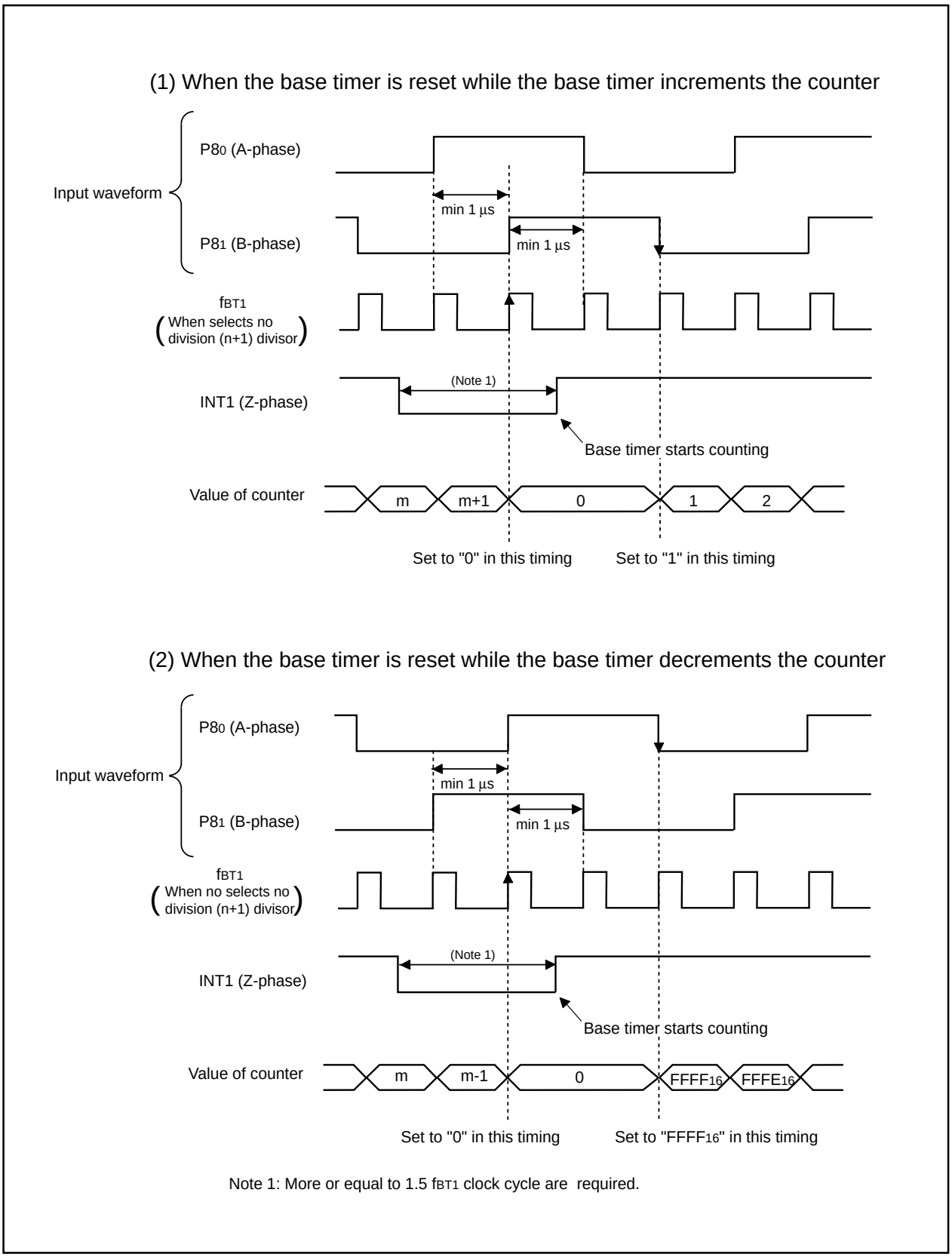



**Figure 13.1.4. Base Timer Operation in Two-phase Pulse Signal Processing Mode**

### 13.1.1 Base Timer Reset Register

The Base Timer Reset Register(G1BTRR) provides the capability to reset the Base Timer(BT) when the base timer count value matches the value stored in the G1BTRR. The G1BTRR is enabled by the RST4 reset cause select bit,G1BCR0(2). This function is identical in operation to the G1PO0 base timer reset that is enabled by RST1.The Base Timer Reset feature is included to allow all eight channels to be used for waveform generation while providing a base timer reset on match function.

It is possible to simultaneously enable both RST1 and RST4, G1PO0 and G1BTRR base timer resets, although operation of both base timer reset on match functions may cause unexpected behavior. It is recommended that only one of RST1 or RST4 be enabled.

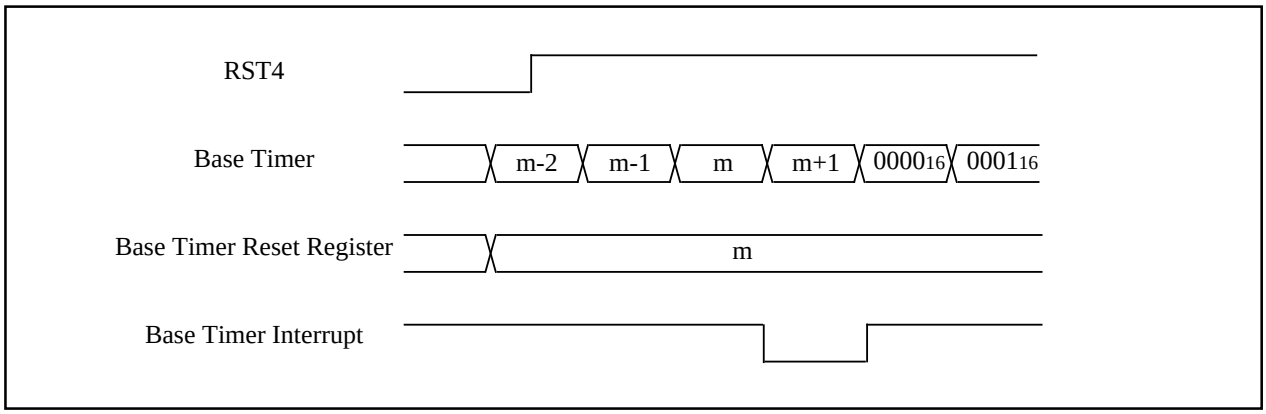



**Figure 13.1.1.1. Base Timer Reset operation by Base Timer Reset Register**

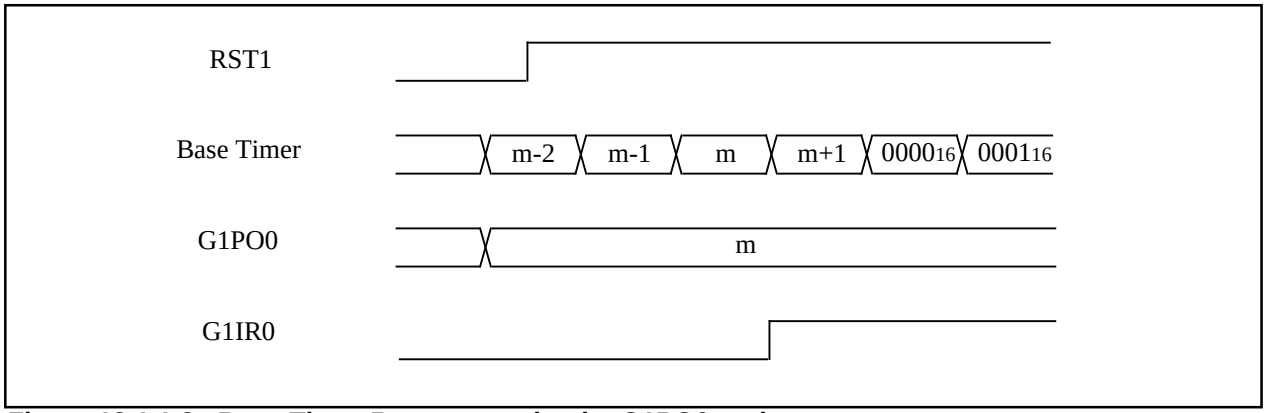



**Figure 13.1.1.2. Base Timer Reset operation by G1PO0 register**

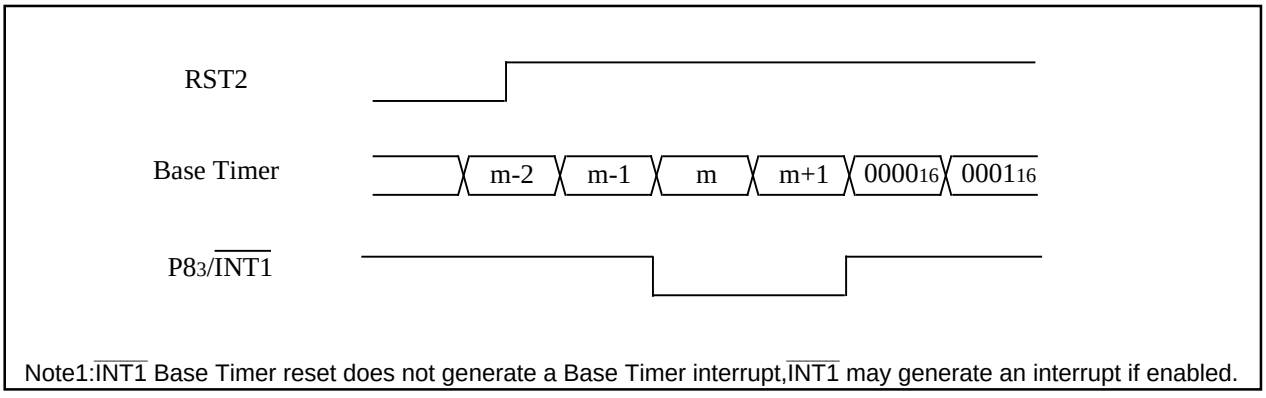



**Figure 13.1.1.3. Base Timer Reset operation by $\overline{INT1}$**

## 13.2 Interrupt Operation

The IC/OC interrupt contains several request causes. Figure 13.2.1 shows the IC/OC interrupt block diagram and Table 13.2.1 shows the IC/OC interrupt assignation.

When either the base timer reset request or base timer overflow request is generated, the IR bit (bit 3 in the BTIC register) corresponding to the IC/OC base timer interrupt is set to "1" (with an interrupt request). Also when an interrupt request of each eight channels (channel i) is generated, the bit i in the G1IR register is set to "1" (with an interrupt request). At this time, if the bit i in the G1IE0 register is "1" (IC/OC interrupt 0 request enabled), the IR bit (bit 3 in the ICOC0IC register) corresponding to the IC/OC interrupt 0 is set to "1" (with an interrupt request). And if the bit i in the G1IE1 register is "1" (IC/OC interrupt 1 request enabled), the IR bit (bit 3 in the ICOC1IC register) corresponding to the IC/OC interrupt 1 is set to "1"(with an interrupt request).

Additionally, because each bit in the G1IR register is not automatically set to "0" even if the interrupt is acknowledged, set to "0" using a program. If these bits are left "1", all IC/OC channel interrupt causes, which are generated after setting the IR bit to "1", will be disabled.

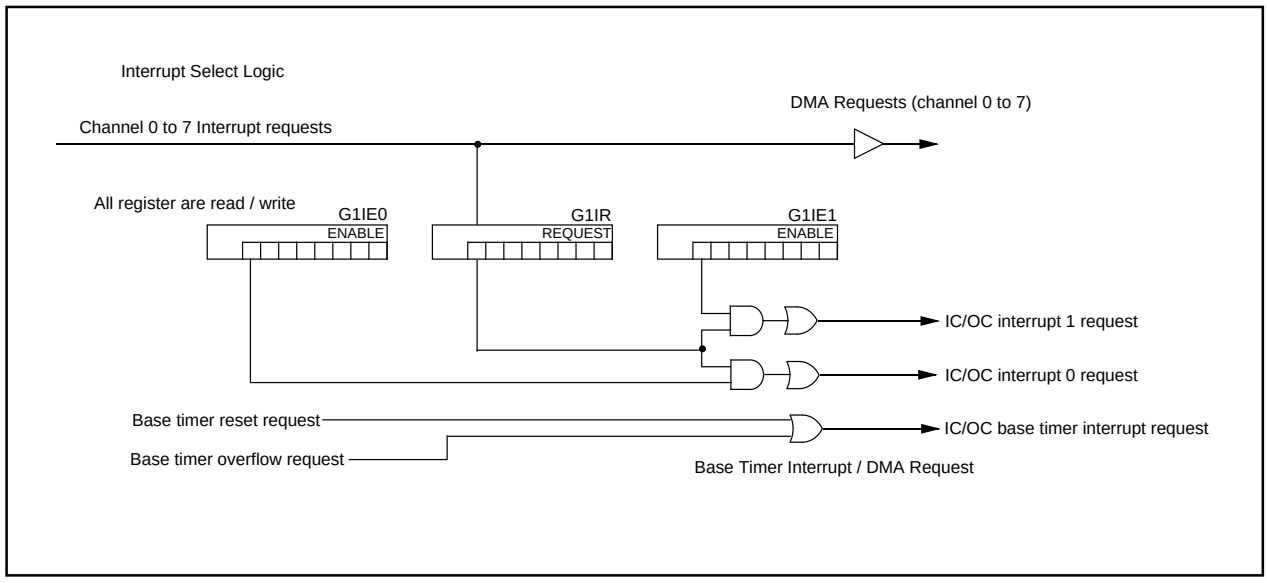



**Figure 13.2.1. IC/OC Interrupt and DMA request generation**

**Table 13.2.1. Interrupt Assignment**

| Interrupt | Interrupt control register |
|---|---|
| IC/OC base timer interrupt | BTIC($0047_{16}$) |
| IC/OC interrupt 0 | ICOC0IC($0045_{16}$) |
| IC/OC interrupt 1 | ICOC0IC($0046_{16}$) |

## 13.3 DMA Support

Each of the interrupt sources - the eight IC/OC channel interrupts and the one Base Timer interrupt - are capable of generating a DMA request.

## 13.4 Time Measurement Function

Synchronizing with an external trigger input, the value of the base timer is stored into the G1TMj register (j=0 to 7). Table 13.4.1 shows specifications of the time measurement function. Table 13.4.2 shows register settings associated with the time measurement function. Figures 13.4.1 and 13.4.2 display operational timing of the time measurement function. Figure 13.4.3 shows operational timing of the prescaler function and the gate function.

**Table 13.4.1. Time Measurement Function Specifications**

| Item | Specification |
|---|---|
| Measurement channel | Channels 0 to 7 |
| Selecting trigger input polarity | Rising edge, falling edge, both edges of the INPC1j pin(Note 1) |
| Measurement start condition | The IFEj bit in the G1FE register should be set to "1" (channels j function enabled) when the FSCj bit (j=0 to 7) in the G1FS register is set to "1" (time measurement function selected). |
| Measurement stop condition | The IFEj bit should be set to "0" (channel j function disabled) |
| Time measurement timing | •No prescaler :every input is a trigger<br>•Prescaler (for channel 6 and channel 7):<br>every [G1TPRk (k=6,7) +1]$^{th}$ input is a trigger |
| Interrupt request generation timing | The G1IRi bit (i=0 to 7) in the interrupt request register (See Figure 13.10) is set to "1" at time measurement timing |
| INPC1j pin function(Note 1) | Trigger input pin |
| Selectable function | • Digital filter function<br>The digital filter samples a trigger input level every $f_1$, $f_2$ or $f_{BT1}$ to pass pulses matching a trigger input level three times<br>• Prescaler function (for channel 6 and channel 7)<br>Trigger inputs are counted to perform time measurement whenever value of the G1TPRk(k=6,7) register + 1 trigger is input<br>• Gate function (for channel 6 and channel 7)<br>When a trigger input is inhibited with setting the GOC bit in the G1TMCRk (k=6,7) register to "1" (gate cleared by matching the G1POp register (p=4 when k=6, p=5 when k=7)) after time measurement by first trigger input, a trigger input is enabled to receive again by matching the base timer with the G1POp register<br>• Digital Debounce function (for channel7)<br>See section 13.6.2 and 17.6 for details |

Note1: The $INPC1_0$ to $INPC1_7$ pins

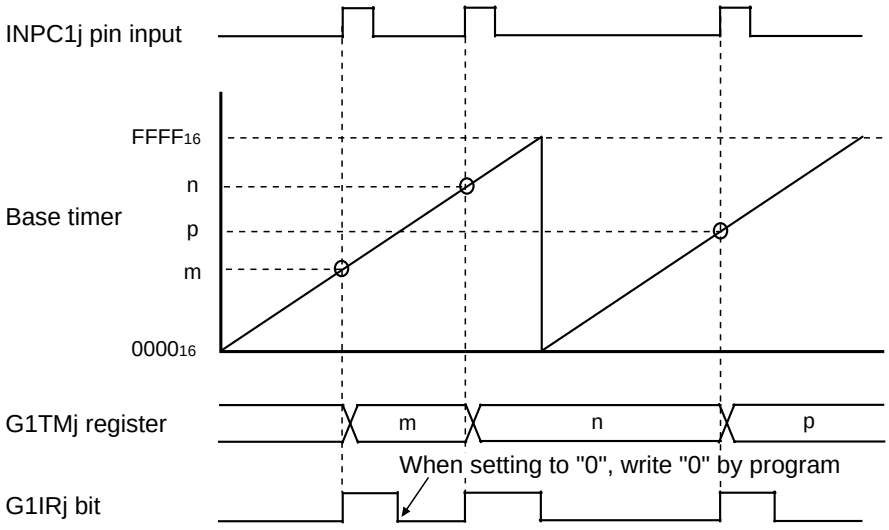
**Table 13.4.2. Register Settings Associated with the Time Measurement Function**

| Register | Bit | Function |
|---|---|---|
| G1TMCRj | CTS1 to CTS0 | Select time measurement trigger |
| | DF1 to DF0 | Select the digital filter function |
| | GT, GOC, GSC | Select the gate function |
| | PR | Select the prescaler function |
| G1TPRk | - | Setting value of prescaler |
| G1FS | FSCj | Set to "1" (time measurement function) |
| G1FE | IFEj | Set to "1" (channel j function enabled) |

j = 0 to 7 k = 6, 7

Bit configuration and function vary depending on which channel is used.

Registers associated with the time measurement function should be set after setting registers associated with the base time.




j=0 to 7
G1IRj bit : Bits in the G1IR register

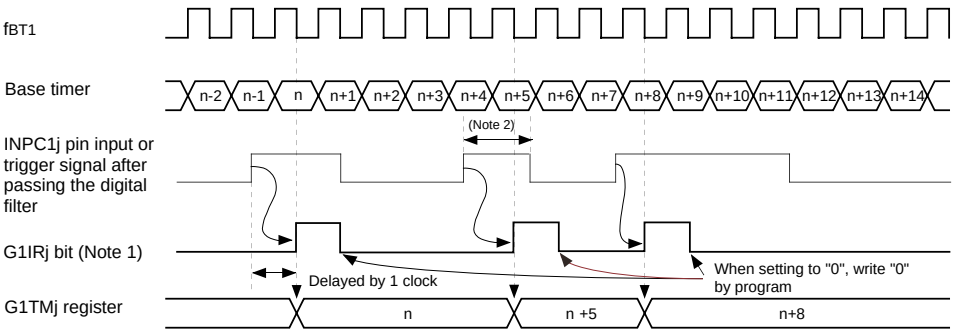
The above applies to the following condition.

- The CTS1 to CTS0 bits in the G1TMCRj registers are set to "01$_2$" (time measurement is trigger on the rising edge). The PR bit is set to "0" (no prescaler used) and the GT bit is set to"0" (no gate function used).
- Bits RTS4, RTS2, and RTS1 of the G1BCR0 and G1BCR1 registers are set to "0" (no base timer reset). The UD1 to UD0 bits are set to "00$_2$" (counter increment mode).

When the base timer matches the G1PO0 register and is set to "0000$_{16}$" (setting the RST1 bit to "1", and the RST4 and RST2 bits to "0"), the base timer is set to "0000$_{16}$" after it reaches a setting value of the G1PO0 register+2.

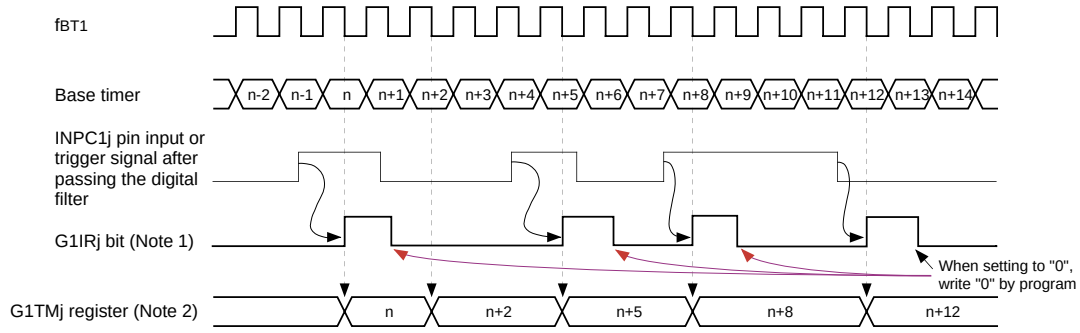
**Figure 13.4.1. Time Measurement Function (1)**

(a) When selecting the rising edge for timer measurement trigger
(The CTS1 to CTS0 bits in the G1TMCR register (j=0 to 7)=01$_2$)



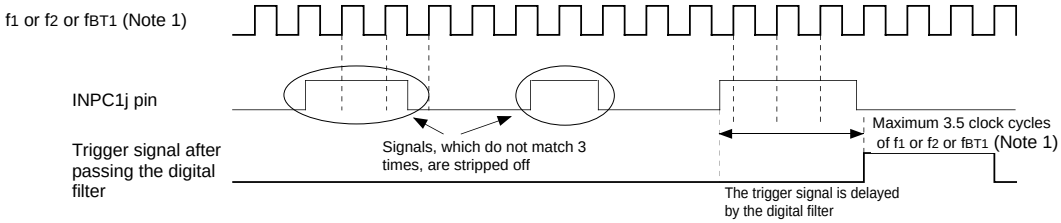

Notes :
1. Bits in the G1IR register.
2. For an input pulse to the INPC1j pin, more or equal to 1.5 fBT1 clock periods are required.

(b) When selecting both edges for timer measurement trigger
(The CTS1 to CTS0 bits=11$_2$)




Notes :
1. Bits in the G1IR register.
2. No interrupt is generated when the microcomputer receives input when the G1IRj bit is in "H". Value of the base timer is stored into the G1TMj register

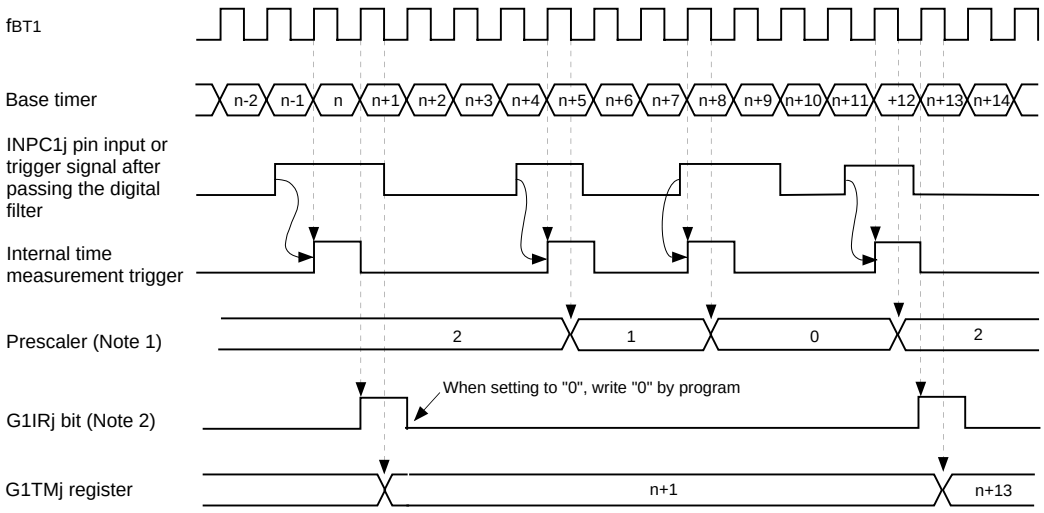
(c) Trigger signal when using digital filter
(The DF1 to DF0 bits in the G1TMCR register =10$_2$ or 11$_2$)




Note 1: fBT1 when the DF1 to DF0 bits are set to "10$_2$", and f1 or f2 when set to "11$_2$".

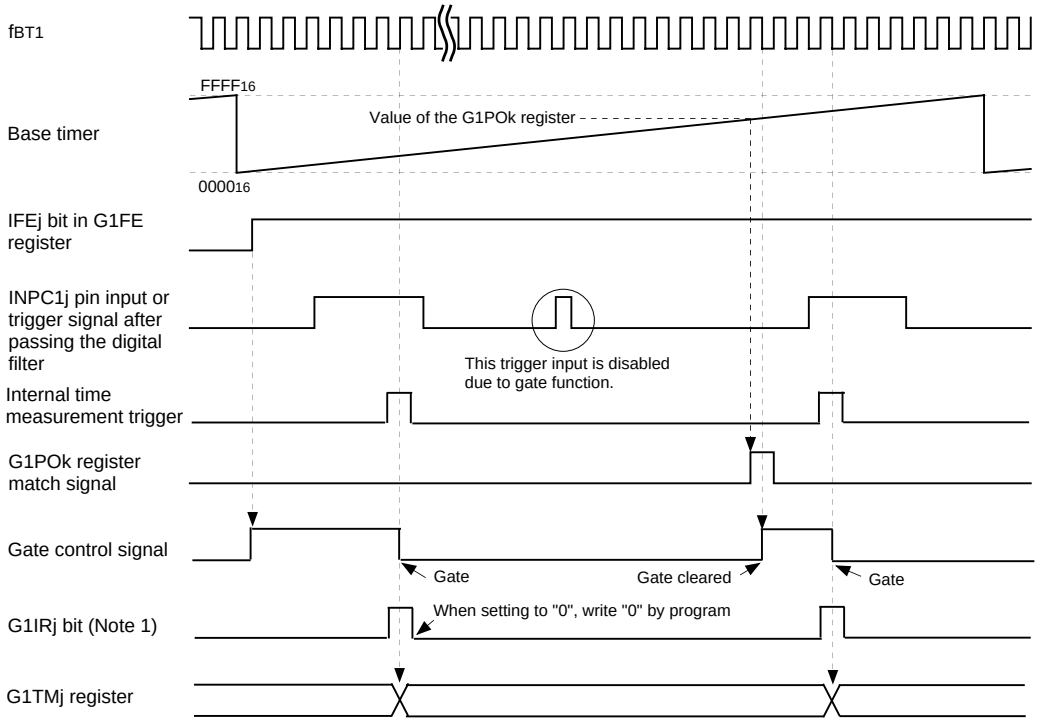
**Figure 13.4.2. Time Measurement Function (2)**

(a) When using the prescaler function
(When the G1TPRj register (j=6, 7) =02$_{16}$, PR bit in the G1TMCRj register (j=6, 7) =1)




Notes :
1. This applies to the second period that the G1TPRj register decrements after setting the PR bit in the G1TMCRj register to "1" (prescaler used).
2. Bits in the G1IR register.

(b) When using the gate function
(Gate function is cleared by matching the G1POk register and base timer, the GT bit in the G1TMCRj register=1, the GOC bit=1)




Note 1: Bits in the G1IR register.

**Figure 13.4.3. Prescaler Function and Gate Function**

## 13.5 Waveform Generation Function

Waveforms are generated when value of the base timer matches G1POj register (j=0 to 7).

The waveform generation function has the following three modes :

- Single-phase waveform output mode
- Phase-delayed waveform output mode
- Set/Reset waveform output (SR waveform output) mode

Table 13.5.1 lists registers associated with the waveform generation function.

**Table 13.5.1. Registers Related to the Waveform Generation Function Settings**

| Register | Bit | Function |
|---|---|---|
| G1POCRj | MOD1 to MOD0 | Select output waveform mode |
| | IVL | Select default value |
| | RLD | Select G1POj register value reload timing |
| | INV | Select inverse output |
| G1POj | - | Select timing to output waveform inverted |
| G1FS | FSCj | Set to "0" (waveform generation function) |
| G1FE | IFEj | Set to "1" (enables function on channel j) |

j = 0 to 7

Bit configuration and function vary depending on which channel is used.

Registers associated with the waveform generation function should be set after setting registers associated with the base time.

### 13.5.1 Single-Phase Waveform Output Mode

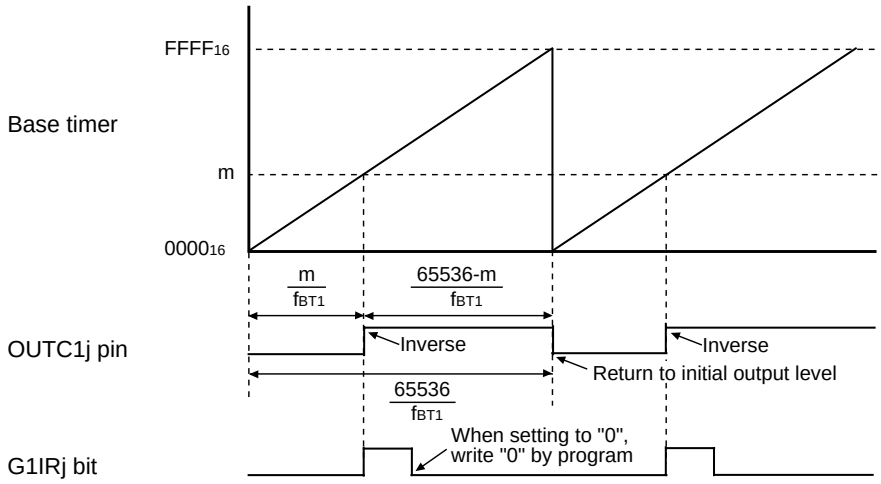
Output level of the OUTC1j pin is inverted when value of the base timer matches that of the G1POj register (j=0 to 7). The inverted output level is returned to a default output level when the base timer reaches "$0000_{16}$". Table 13.5.1.1 lists specifications of single-phase waveform mode. Figure 13.5.1.1 lists an example of single-phase waveform mode operation.

**Table 13.5.1.1. Single-phase Waveform Output Mode Specifications**

| Item | Specification |
|---|---|
| Output waveform | • Free-running operation<br>(the RST1, RST2, and RST4 bits of the G1BCR1 and G1BCR0 registers are set to "0" (no reset))<br>Cycle : $\frac{65536}{f_{BT1}}$<br>Default output level : $\frac{m}{f_{BT1}}$<br>Inverse level : $\frac{65536-m}{f_{BT1}}$<br>• The base timer is reset when its value matches that of either register<br>(a) G1PO0 (enabled by setting bit RST1 to "1", and bits RST4 and RST2 to "0"), or<br>(b) G1BTRR (enabled by setting bit RST4 to "1", and bits RST2 and RST1 to "0")<br>Cycle : $\frac{n+2}{f_{BT1}}$<br>Default output level : $\frac{m}{f_{BT1}}$<br>Inverse level : $\frac{n+2-m}{f_{BT1}}$<br>m : setting value of the G1POj register (j=0 to 7), $0001_{16}$ to $FFFD_{16}$<br>n : setting value of the G1PO0 register or the G1BTRR register, $0001_{16}$ to $FFFD_{16}$ |
| Waveform output start condition | The IFEj bit in the G1FE register should be set to "1" (channel j function enabled) |
| Waveform output stop condition | The IFEj bit should be set to "0" (channel j function disabled) |
| Interrupt request | The G1IRj bit in the interrupt request register is set to "1" when value of the base timer matches one of the G1POj registers. (See Figure 13.10.) |
| OUTC1j pin(Note 1) | Pulse output |
| Selectable function | • Default value set function : Output level is set when waveform output starts<br>• Inverse output function : Waveform level is inverted to output waveform from the OUTC1j pin |

Note 1: The $OUTC1_0$ to $OUTC1_7$ pins .

(1) Free-running operation
(Bits RST4, RST2, and RST1 of the G1BCR0 and G1BCR1 registers are set to "0")

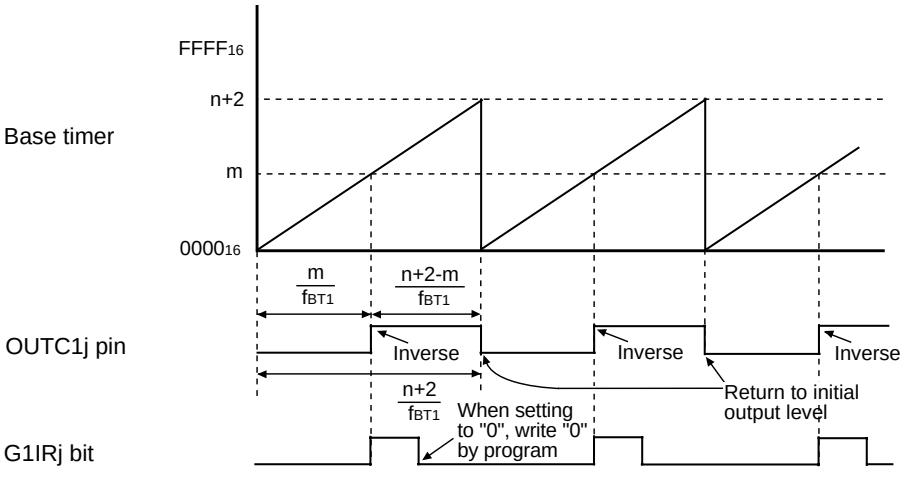



j=0 to 7
m : Setting value of the G1POj register
G1IRj bit : Bits in the G1IR register

The above applies to the following conditions.

- The IVL bit in the G1POCRj register is set to "0" (output "L" as an initial value) and the INV bit is set to "0" (no output inverted).
- Bits RST4, RST2, and RST1 of the G1BCR0 and G1BCR1 registers are set to "0" (no base timer reset), and the UD1 to UD0 bits are set to "00$_2$" (counter increment mode).

(2) The base timer is reset when its value matches that of either register
(a) G1PO0 (enabled by setting bit RST1 to "1", and bits RST4 and RST2 to "0"), or
(b) G1BTRR (enabled by setting bit RST4 to "1", and bits RST2 and RST1 to "0")




j=1 to 7
m : Setting value of the G1POk register
n: Setting value of either G1PO0 register or G1BTRR register
G1IRj bit : Bits in the G1IR register

The above applies to the following conditions.

- The IVL bit in the G1POCRj register is set to "0" (output "L" as an initial value) and the INV bit be set to "0" (no output inverted).
- The UD1 to UD0 bits are set to "00$_2$" (counter increment mode).

**Figure 13.5.1.1. Single-phase Waveform Output Mode**

### 13.5.2 Phase-Delayed Waveform Output Mode

Output level of the OUTC1j pin is inverted whenever the value of the base timer matches that of the G1POj register value ( j=0 to 7). Table 13.5.2.1 lists specifications of phase-delayed waveform mode. Figure 13.5.2.1 lists an example of phase-delayed waveform mode operation.

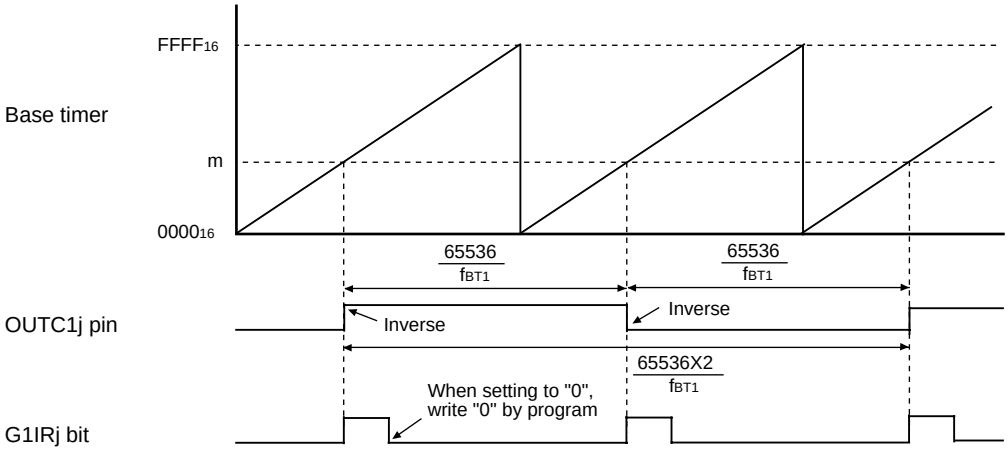
**Table 13.5.2.1. Phase-delayed Waveform Output Mode Specifications**

| Item | Specification |
|---|---|
| Output waveform | • Free-running operation<br>(the RST1, RST2, and RST4 bits of the G1BCR1 and G1BCR0 registers are set to "0" (no reset))<br>Cycle : $\frac{65536 \times 2}{f_{BT1}}$<br>"H" and "L" width : $\frac{65536}{f_{BT1}}$<br>• Setting bit RST1 to "1", and bits RST4 and RST2 to "0" enables the base timer to be reset when its value matches the G1PO0 register. Likewise, setting bit RST4 to "1",and bits RST2 and RST1 to "0" enables the base timer to be reset when its value matches the G1BTRR register.<br>Cycle : $\frac{2(n+2)}{f_{BT1}}$<br>"H" and "L" width : $\frac{n+2}{f_{BT1}}$<br>n : setting value of either G1PO0 register or G1BTRR register, $0001_{16}$ to $FFFD_{16}$ |
| Waveform output start condition(Note 1) | The IFEj bit in the G1FE register should be set to "1" (channel j function enabled) |
| Waveform output stop condition | The IFEj bit should be set to "0" (channel j function disabled) |
| Interrupt request | The G1IRj bit in the interrupt request register is set to "1" when value of the base timer matches one of the G1POj registers. (See Figure 13.10.) |
| OUTC1j pin(Note 2) | Pulse output |
| Selectable function | • Default value set function : Output level is set when waveform output starts<br>• Inverse output function : Waveform level is inverted to output waveform from the OUTC1j pin |

Note 1 : The FSCj bit in the G1FS register should be set to "0" (waveform generation function selected) in the channels shared by the time measurement function and waveform generation function.

Note 2 : The $OUTC1_0$ to $OUTC1_7$ pins.

(1) Free-running operation
(Bits RST4, RST2, and RST1 of the G1BCR0 and G1BCR1 registers are set to "0")

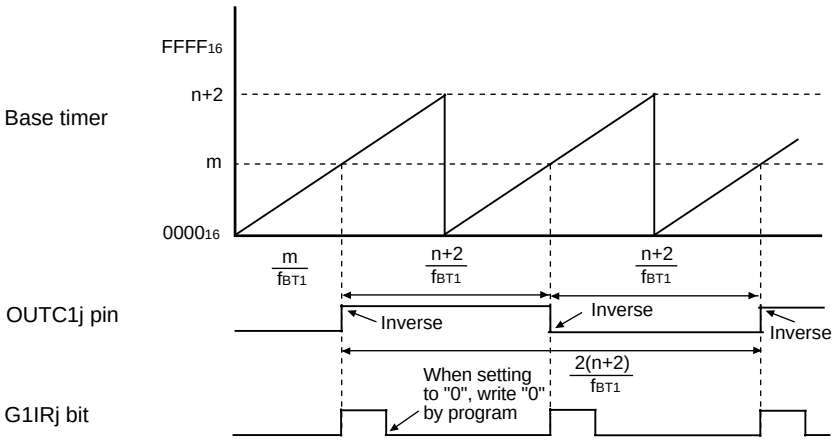



j=0 to 7
m : Setting value of the G1POj register
G1IRj bit : Bits in the G1IR register

The above applies to the following conditions.

- The IVL bit in the G1POCRj register is set to "0" (output "L" as an initial value). The INV bit is set to "0" (no output inverted).
- Bits RST4, RST2, and RST1 of the G1BCR0 and G1BCR1 registers are set to "0" (no base timer reset). The UD1 to UD0 bits are set to "00$_2$" (counter increment mode).

(2) Base timer is reset when its value matches that of either register (a) G1PO0 (enabled by setting bit RST1 to "1", and bits RST4 and RST2 to "0"), or (b) G1BTRR (enabled by setting bit RST4 to "1", and bits RST2 and RST1 to "0")




j=1 to 7
m : Setting value of the G1POj register n: Setting value of either register G1PO0 or G1BTRR
G1IRj bit : Bits in the G1IR register

The above applies to the following conditions.

- The IVL bit in the G1POCRj register is set to "0" (output "L" as an initial value). The INV bit is set to "0" (no output inverted).
- The UD1 to UD0 bits are set to "00$_2$" (counter increment mode).

**Figure 13.5.2.1. Phase-delayed Waveform Output Mode**

### 13.5.3 Set/Reset Waveform Output (SR Waveform Output) Mode

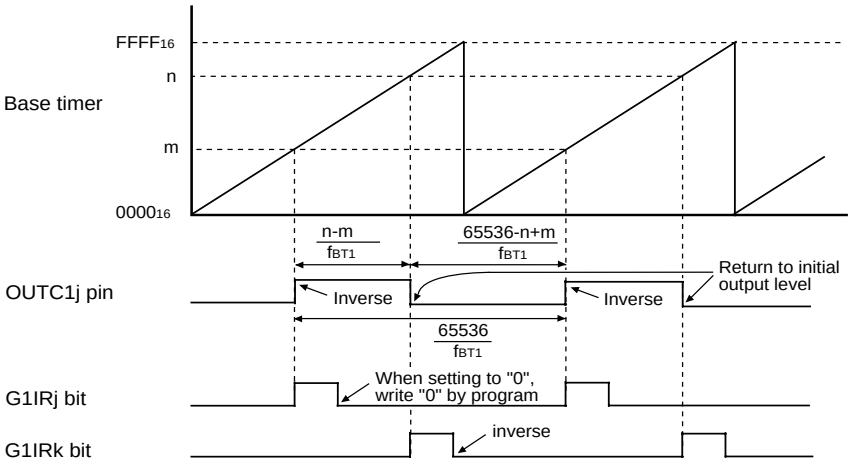
Output level of the OUTC1j pin is inverted when the base timer value matches that of the G1POj register value (j=0, 2, 4, 6). It is returned to default output level when the base timer value matches that of the G1POk register (k=j+1). Table 13.5.3.1 lists specifications of SR waveform mode. Figure 13.5.3.1 lists an example of the SR waveform mode operation.

**Table 13.5.3.1. SR Waveform Output Mode Specifications**

| Item | Specification |
|---|---|
| Output waveform | • Free-running operation<br>(the RST1, RTS2, and RST4 bits of the G1BCR1 and G1BCR0 registers are set to "0" (no reset))<br>Cycle : $\frac{65536}{f_{BT1}}$<br>Inverse level(Note 1) : $\frac{m-n}{f_{BT1}}$<br>• Setting bit RST1 to "1", and bits RST4 and RST2 to "0" enables the base timer to be reset when its value matches the G1PO0 register(Note 2). Likewise, setting bit RST4 to "1", and bits RST2 and RST1 to "0" enables the base timer to be reset when its value matches the G1BTRR register.<br>Cycle : $\frac{p+2}{f_{BT1}}$<br>Inverse level(Note 1) : $\frac{m-n}{f_{BT1}}$<br>m : setting value of the G1POj register (j=0, 2, 4, 6 )<br>n : setting value of the G1POk register (k=j+1)<br>p : setting value of either G1PO0 register or G1BTRR register<br>all m, n, p: $0001_{16}$ to $FFFD_{16}$ |
| Waveform output start condition(Note 3) | The IFEj bit in the G1FE register should be set to "1" (channel j function enabled) |
| Waveform output stop condition | The IFEj bit should be set to "0" (channel j function disabled) |
| Interrupt request | The G1IRj bit in the interrupt request register is set to "1" when value of the base timer matches one of the G1POj registers.<br>The G1IRk bit in the interrupt request register is set to "1 " when value of the base timer matches one of the G1POk registers (See Figure 13.10.) |
| OUTC1j pin(Note 3) | Pulse output |
| Selectable function | • Default value set function : Output level is set when waveform output starts<br>• Inverse output function : Waveform level is inverted to output waveform from the OUTC1j pin |

Note 1 : The waveform generation register of odd channel should have greater value than the one of even channel has.
Note 2 : When the G1PO0 register resets the base timer, the SR waveform generation function with channels 0 and 1 cannot be used.
Note 3 : The $OUTC1_0$, $OUTC1_2$, OUTC14, $OUTC1_6$ pins.

(1) Free-running operation
(The RST1 bit is set to "1", and both RST4 and RST2 bits are set to "0")

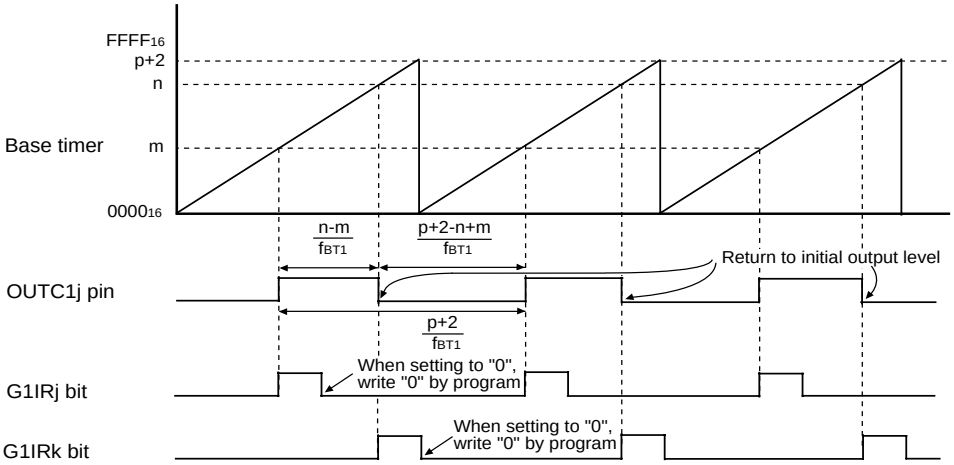



j=0, 2, 4, 6 k=j+1
m : Setting value of the G1POk register n: Setting value of the G1POj register
G1IRj, G1IRk bits: Bits in the G1IR register

The above applies to the following conditions.

- The IVL bit in the G1POCRj register is set to "0" (output "L" as an initial value). The INV bit is set to "0" (no output inverted).
- Bits RST4, RST2, and RST1 of the G1BCR0 and G1BCR1 registers are set to "0" (no base timer reset). The UD1 to UD0 bits are set to "00$_2$" (counter increment mode).

(2) Base timer is reset when its value matches that of either register (a) G1PO0 (enabled by setting bit RST1 to "1", and bits RST4 and RST2 to "0"), or (b) G1BTRR (enabled by setting bit RST4 to "1", and bits RST2 and RST1 to "0")




j=2, 4, 6 k=j+1
m : Setting value of the G1POk register n: Setting value of the G1POj register
p: Setting value of either register G1PO0 or G1BTRR
G1IRj, G1IRk bits: Bits in the G1IR register

The above applies to the following conditions.

- The IVL bit in the G1POCRj register is set to "0" (output "L" as an initial value). The INV bit is set to "0" (no output inverted).
- The UD1 to UD0 bits are set to "00$_2$" (counter increment mode).

**Figure 13.5.3.1. Set/Reset Waveform Output Mode**

## 13.6 I/O Port Function Select

The M16C/28 will automatically configure the port package pins to be IC/OC inputs or outputs based on the values in the Function Enable (G1FE) and Function Select (G1FS) registers. When using PWM S-R mode, two channels are enabled and selected as output, but only one output, the output corresponding to the even numbered channel, is generated.

The port package pin corresponding to the odd numbered channel is available for use as General Purpose Input / Output.

**Table 13.6.1. Pin setting for Time Measurement and Waveform Generation Functions**

| Pin | IFE | FSC | MOD1 | MOD0 | Port Direction | Port Data |
|---|---|---|---|---|---|---|
| $P2_7$/INPC1$_7$/ OUTC1$_7$ | 0 | X | X | X | Determined by $PD2_7$ | $P2_7$ |
| | 1 | 1 | X | X | Determined by $PD2_7$, Input to INPC1$_7$ is always active | $P2_7$ or INPC1$_7$ |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC1$_7$ |
| | 1 | 0 | 0 | 1 | Determined by $PD2_7$, S-R PWM mode | $P2_7$ |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC1$_7$ |
| $P2_6$/INPC1$_6$/ OUTC1$_6$ | 0 | X | X | X | Determined by $PD2_6$ | $P2_6$ |
| | 1 | 1 | X | X | Determined by $PD2_6$, Input to INPC1$_6$ is always active | $P2_6$ or INPC1$_6$ |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC1$_6$ |
| | 1 | 0 | 0 | 1 | SR Waveform Output | OUTC1$_6$ |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC1$_6$ |
| $P2_5$/INPC1$_5$/ OUTC1$_5$ | 0 | X | X | X | Determined by $PD2_5$ | $P2_5$ |
| | 1 | 1 | X | X | Determined by $PD2_5$, Input to INPC1$_5$ is always active | $P2_5$ or INPC1$_5$ |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC1$_5$ |
| | 1 | 0 | 0 | 1 | Determined by $PD2_5$, S-R PWM mode | $P2_5$ |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC1$_5$ |
| $P2_4$/INPC1$_4$/ OUTC1$_4$ | 0 | X | X | X | Determined by $PD2_4$ | $P2_4$ |
| | 1 | 1 | X | X | Determined by $PD2_4$, Input to INPC1$_4$ is always active | $P2_4$ or INPC1$_4$ |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC1$_4$ |
| | 1 | 0 | 0 | 1 | SR Waveform Output | OUTC1$_4$ |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC1$_4$ |
| $P2_3$/INPC1$_3$/ OUTC1$_3$ | 0 | X | X | X | Determined by $PD2_3$ | $P2_3$ |
| | 1 | 1 | X | X | Determined by $PD2_3$, Input to INPC1$_3$ is always active | $P2_3$ or INPC1$_3$ |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC1$_3$ |
| | 1 | 0 | 0 | 1 | Determined by $PD2_3$, S-R PWM mode | $P2_3$ |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC1$_3$ |
| $P2_2$/INPC1$_2$/ OUTC1$_2$ | 0 | X | X | X | Determined by $PD2_2$ | $P2_2$ |
| | 1 | 1 | X | X | Determined by $PD2_2$, Input to INPC1$_2$ is always active | $P2_2$ or INPC1$_2$ |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC1$_2$ |
| | 1 | 0 | 0 | 1 | SR Waveform Output | OUTC1$_2$ |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC1$_2$ |
| $P2_1$/INPC1$_1$/ OUTC1$_1$ | 0 | X | X | X | Determined by $PD2_1$ | $P2_1$ |
| | 1 | 1 | X | X | Determined by $PD2_1$, Input to INPC1$_1$ is always active | $P2_1$ or INPC1$_1$ |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC1$_1$ |
| | 1 | 0 | 0 | 1 | Determined by $PD2_1$, S-R PWM mode | $P2_1$ |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC1$_1$ |
| $P2_0$/INPC1$_0$/ OUTC1$_0$ | 0 | X | X | X | Determined by $PD2_0$ | $P2_0$ |
| | 1 | 1 | X | X | Determined by $PD2_0$, Input to INPC1$_0$ is always active | $P2_0$ or INPC1$_0$ |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC1$_0$ |
| | 1 | 0 | 0 | 1 | SR Waveform Output | OUTC1$_0$ |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC1$_0$ |

IFE: IFEj (j=0 to 7) bits in the G1FE register.

FSC: FSCj (j=0 to 7) bits in the G1FS register.

MOD2 to MOD1: Bits in the G1POCRj (j=0 to 7) register.

### 13.6.1 INPC17 Alternate Input Pin Selection

The input capture pin for IC/OC channel 7 can be assigned to one of two package pins.

Control bit, G1BCR0(6) CH7INSEL, Channel 7 input select, selects IC/OC INPC17 to come from P27/OUTC17/INPC17 or P17/$\overline{\text{INT5}}$/INPC17/IDU.

### 13.6.2 Digital Debounce Function for Pin P17/$\overline{\text{INT5}}$/INPC17

The $\overline{\text{INT5}}$/INPC17 input from the P17/$\overline{\text{INT5}}$/INPC17/IDU pin has an effective digital debounce function for a noise rejection. Refer to "**17.6 Digital Debounce function**" for this detail.

# 14. Serial I/O

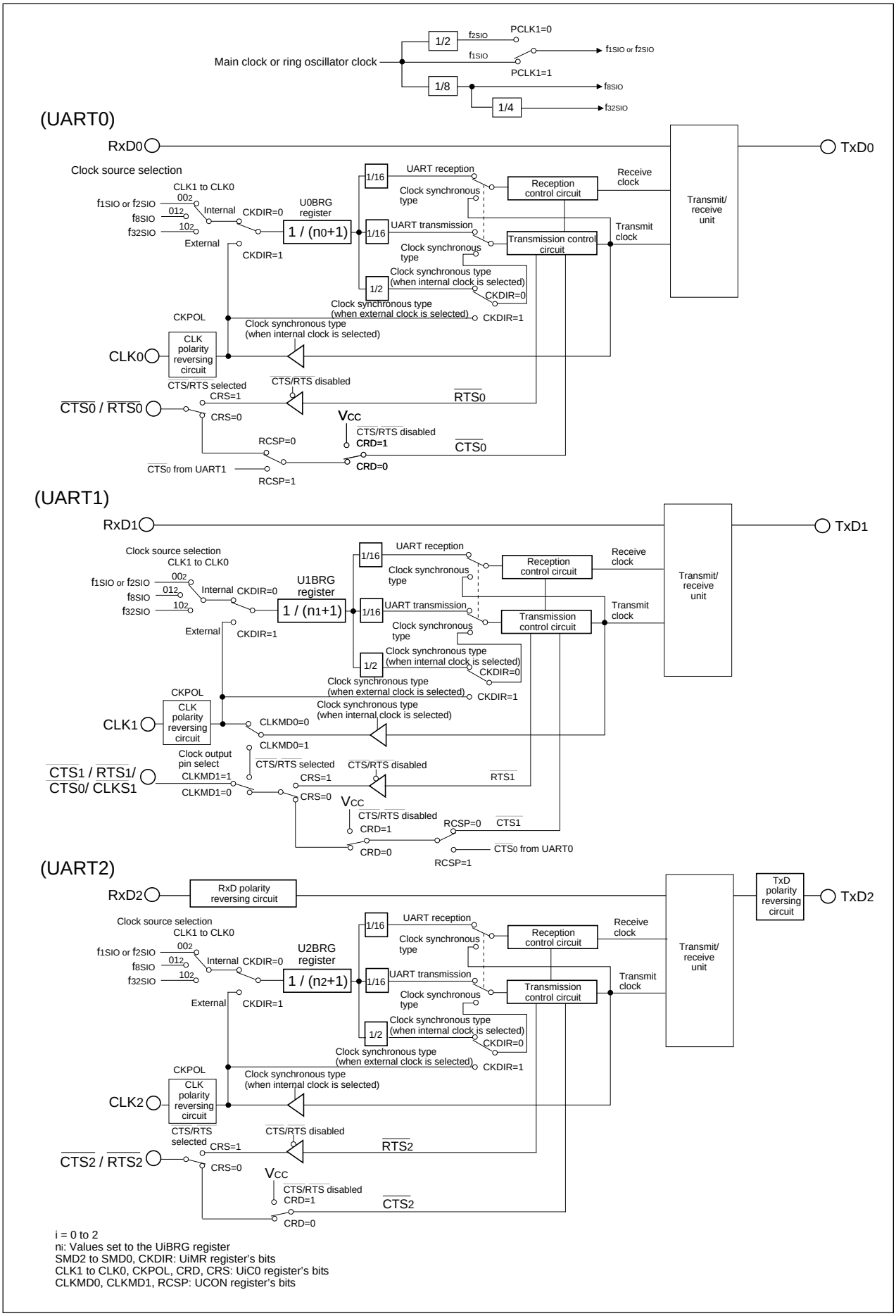
Serial I/O is configured with five channels: UART0 to UART2, SI/O3 and SI/O4.
SI/O4 is not in 64 pin version.

## 14.1. UARTi (i=0 to 2)

UARTi each have an exclusive timer to generate a transfer clock, so they operate independently of each other.
Figure 14.1.1 shows the block diagram of UARTi. Figures 14.1.2 and 14.1.3 shows the block diagram of the UARTi transmit/receive.

UARTi has the following modes:
- Clock synchronous serial I/O mode
- Clock asynchronous serial I/O mode (UART mode).
- Special mode 1 ($I^2C$ mode) : UART2
- Special mode 2 : UART2
- Special mode 3 (Bus collision detection function, IE mode) : UART2
- Special mode 4 (SIM mode) : UART2

Figures 14.1.4 to 14.1.9 show the UARTi-related registers.
Refer to tables listing each mode for register setting.




i = 0 to 2
ni: Values set to the UiBRG register
SMD2 to SMD0, CKDIR: UiMR register's bits
CLK1 to CLK0, CKPOL, CRD, CRS: UiC0 register's bits
CLKMD0, CLKMD1, RCSP: UCON register's bits

**Figure 14.1.1. Block diagram of UARTi (i = 0 to 2)**

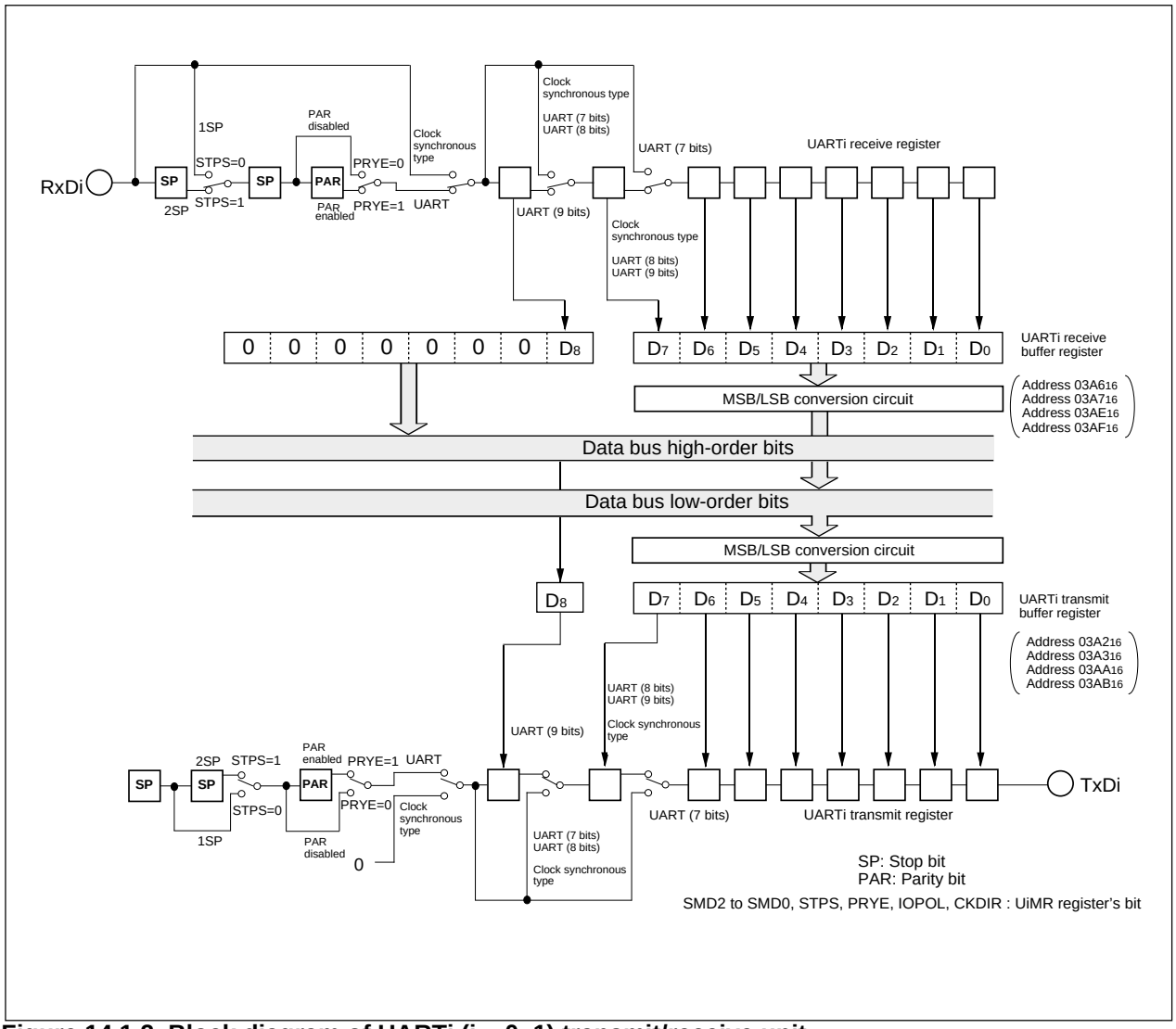



**Figure 14.1.2. Block diagram of UARTi (i = 0, 1) transmit/receive unit**

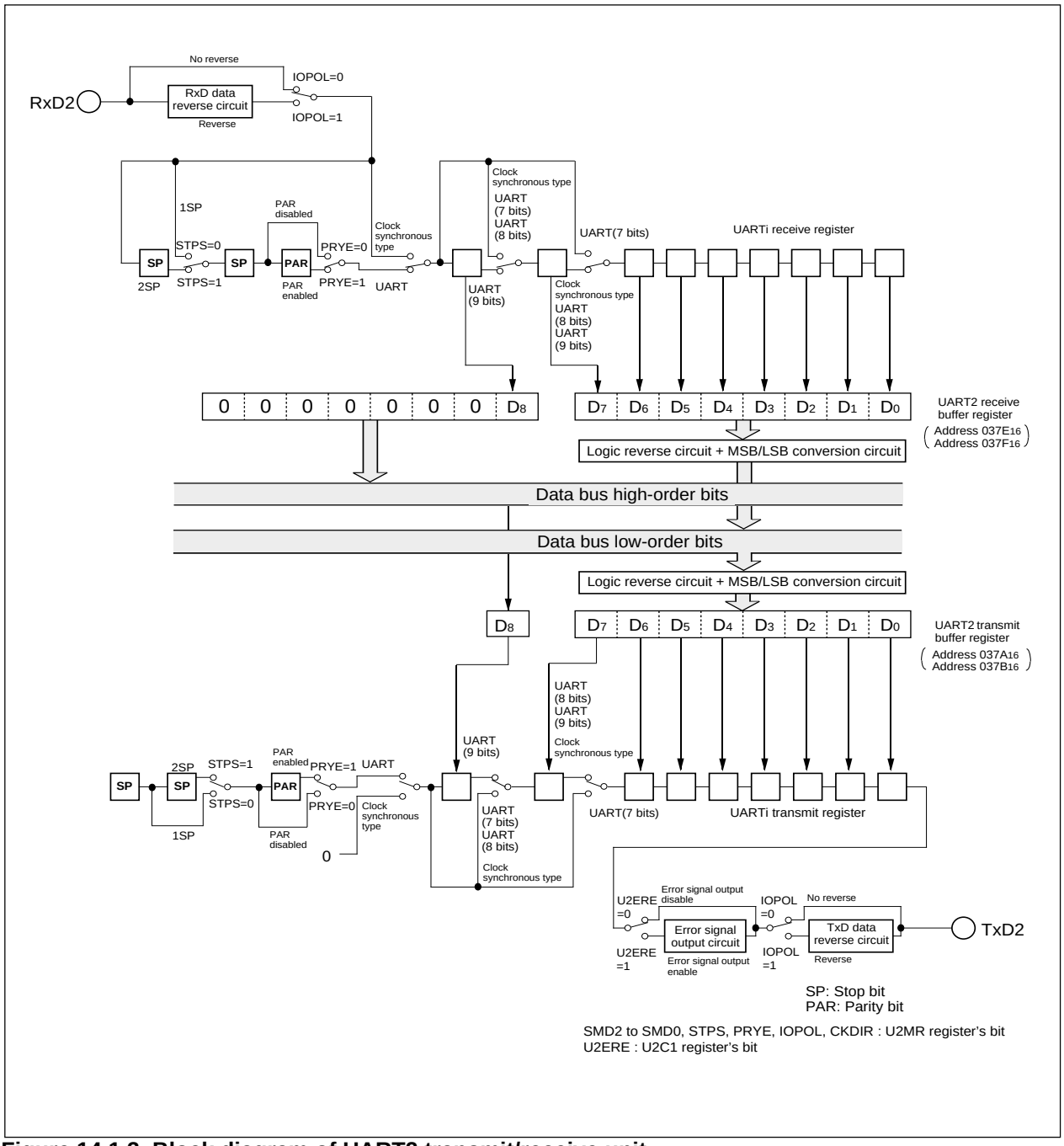
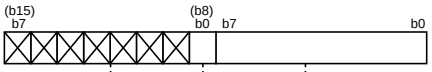



**Figure 14.1.3. Block diagram of UART2 transmit/receive unit**

UARTi transmit buffer register (i=0 to 2)(Note)

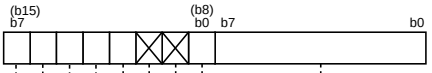



| Symbol | Address | After reset |
|---|---|---|
| U0TB | $03A3_{16}$-$03A2_{16}$ | Indeterminate |
| U1TB | $03AB_{16}$-$03AA_{16}$ | Indeterminate |
| U2TB | $037B_{16}$-$037A_{16}$ | Indeterminate |

| Function | RW |
|---|---|
| Transmit data | WO |
| Nothing is assigned.<br>In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate. | — |

Note: Use MOV instruction to write to this register.

UARTi receive buffer register (i=0 to 2)

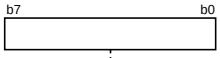



| Symbol | Address | After reset |
|---|---|---|
| U0RB | $03A7_{16}$-$03A6_{16}$ | Indeterminate |
| U1RB | $03AF_{16}$-$03AE_{16}$ | Indeterminate |
| U2RB | $037F_{16}$-$037E_{16}$ | Indeterminate |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| — (b7-b0) | ——— | Receive data ($D_7$ to $D_0$) | RO |
| — (b8) | ——— | Receive data ($D_8$) | RO |
| — (b10-b9) | Nothing is assigned.<br>In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”. | | — |
| ABT | Arbitration lost detecting flag (Note 2) | 0 : Not detected<br>1 : Detected | RW |
| OER | Overrun error flag (Note 1) | 0 : No overrun error<br>1 : Overrun error found | RO |
| FER | Framing error flag (Note 1) | 0 : No framing error<br>1 : Framing error found | RO |
| PER | Parity error flag (Note 1) | 0 : No parity error<br>1 : Parity error found | RO |
| SUM | Error sum flag (Note 1) | 0 : No error<br>1 : Error found | RO |

Note 1: When the UiMR register’s SMD2 to SMD0 bits = “$000_2$” (serial I/O disabled) or the UiC1 register’s RE bit = “0” (reception disabled), all of the SUM, PER, FER and OER bits are set to “0” (no error). The SUM bit is set to “0” (no error) when all of the PER, FER and OER bits = “0” (no error). Also, the PER and FER bits are set to “0” by reading the lower byte of the UiRB register.
Note 2: The ABT bit is set to “0” by writing “0” in a program. (Writing “1” has no effect.)

UARTi baud rate generation register (i=0 to 2)(Notes 1, 2)



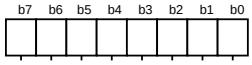

| Symbol | Address | After reset |
|---|---|---|
| U0BRG | $03A1_{16}$ | Indeterminate |
| U1BRG | $03A9_{16}$ | Indeterminate |
| U2BRG | $0379_{16}$ | Indeterminate |

| Function | Setting range | RW |
|---|---|---|
| Assuming that set value = n, UiBRG divides the count source by n + 1 | $00_{16}$ to $FF_{16}$ | WO |

Note 1: Write to this register while serial I/O is neither transmitting nor receiving.
Note 2: Use MOV instruction to write to this register.

**Figure 14.1.4. Serial I/O-related registers (1)**

UARTi transmit/receive mode register (i=0, 1)

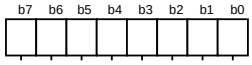



Symbol: U0MR, U1MR — Address: 03A0$_{16}$, 03A8$_{16}$ — After reset: 00$_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SMD0 | Serial I/O mode select bit (Note 2) | b2 b1 b0<br>0 0 0 : Serial I/O disabled<br>0 0 1 : Clock synchronous serial I/O mode<br>1 0 0 : UART mode transfer data 7 bits long<br>1 0 1 : UART mode transfer data 8 bits long<br>1 1 0 : UART mode transfer data 9 bits long<br>Must not be set except above | RW |
| SMD1 | | | RW |
| SMD2 | | | RW |
| CKDIR | Internal/external clock select bit | 0 : Internal clock<br>1 : External clock (Note 1) | RW |
| STPS | Stop bit length select bit | 0 : One stop bit<br>1 : Two stop bits | RW |
| PRY | Odd/even parity select bit | Effective when PRYE = 1<br>0 : Odd parity<br>1 : Even parity | RW |
| PRYE | Parity enable bit | 0 : Parity disabled<br>1 : Parity enabled | RW |
| $\overline{\text{(b7)}}$ | Reserve bit | Write to "0" | RW |

Note 1: Set the corresponding port direction bit for each CLKi pin to “0” (input mode).
Note 2: To receive data, set the corresponding port direction bit for each RxDi pin to “0” (input mode).

UART2 transmit/receive mode register



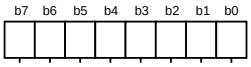

Symbol: U2MR — Address: 037816 — After reset: 00$_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SMD0 | Serial I/O mode select bit (Note 2) | b2 b1 b0<br>0 0 0 : Serial I/O disabled<br>0 0 1 : Clock synchronous serial I/O mode<br>0 1 0 : I$^2$C mode (Note 3)<br>1 0 0 : UART mode transfer data 7 bits long<br>1 0 1 : UART mode transfer data 8 bits long<br>1 1 0 : UART mode transfer data 9 bits long<br>Must not be set except above | RW |
| SMD1 | | | RW |
| SMD2 | | | RW |
| CKDIR | Internal/external clock select bit | 0 : Internal clock<br>1 : External clock (Note 1) | RW |
| STPS | Stop bit length select bit | 0 : One stop bit<br>1 : Two stop bits | RW |
| PRY | Odd/even parity select bit | Effective 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 I/O polarity reverse bit | 0 : No reverse<br>1 : Reverse | RW |

Note 1: Set the corresponding port direction bit for each CLK2 pin to “0” (input mode).
Note 2: To receive data, set the corresponding port direction bit for each RxD2 pin to “0” (input mode).
Note 3: Set the corresponding port direction bit for SCL and SDA pins to “0” (input mode).

**Figure 14.1.5. Serial I/O-related registers (2)**

UARTi transmit/receive control register 0 (i=0 to 2)

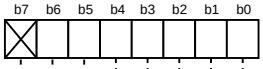



Symbol: U0C0 to U2C0 | Address: 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$ | After reset: 00001000$_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| CLK0 | BRG count source select bit | b1 b0<br>0 0 : f1SIO or f2SIO is selected<br>0 1 : f8SIO is selected<br>1 0 : f32SIO is selected<br>1 1 : Must not be set | RW |
| CLK1 | | | RW |
| CRS | $\overline{CTS}$/$\overline{RTS}$ function select bit (Note 3) | Effective when CRD = 0<br>0 : $\overline{CTS}$ function is selected (Note 1)<br>1 : RTS function is selected | RW |
| TXEPT | Transmit register empty flag | 0 : Data present in transmit register (during transmission)<br>1 : No data present in transmit register (transmission completed) | RO |
| CRD | $\overline{CTS}$/$\overline{RTS}$ disable bit | 0 : $\overline{CTS}$/$\overline{RTS}$ function enabled<br>1 : $\overline{CTS}$/$\overline{RTS}$ function disabled (P60, P64 and P73 can be used as I/O ports) | RW |
| NCH | Data output select bit | 0 : TxDi/SDAi and SCLi pins are CMOS output<br>1 : TxDi/SDAi and SCLi pins are N-channel open-drain output | RW |
| CKPOL | CLK polarity select bit | 0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge<br>1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge | RW |
| UFORM | Transfer format select bit (Note 2) | 0 : LSB first<br>1 : MSB first | RW |

Note 1: Set the corresponding port direction bit for each $\overline{CTS}$i pin to "0" (input mode).
Note 2: Effective for clock synchronous serial I/O mode, UART mode transfer data 8 bits long and special mode 2.
Note 3: $\overline{CTS}$1/$\overline{RTS}$1 can be used when the UCON register's CLKMD1 bit = "0" (only CLK1 output) and the UCON register's RCSP bit = "0" ($\overline{CTS}$0/$\overline{RTS}$0 not separated).

UART transmit/receive control register 2



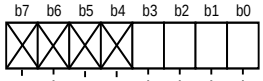

Symbol: UCON | Address: 03B0$_{16}$ | After reset: X0000000$_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| U0IRS | UART0 transmit interrupt cause select bit | 0 : Transmit buffer empty (TI = 1)<br>1 : Transmission completed (TXEPT = 1) | RW |
| U1IRS | UART1 transmit interrupt cause select bit | 0 : Transmit buffer empty (TI = 1)<br>1 : Transmission completed (TXEPT = 1) | RW |
| U0RRM | UART0 continuous receive mode enable bit | 0 : Continuous receive mode disabled<br>1 : Continuous receive mode enable | RW |
| U1RRM | UART1 continuous receive mode enable bit | 0 : Continuous receive mode disabled<br>1 : Continuous receive mode enabled | RW |
| CLKMD0 | UART1 CLK/CLKS select bit 0 | Effective when CLKMD1 = "1"<br>0 : Clock output from CLK1<br>1 : Clock output from CLKS1 | RW |
| CLKMD1 | UART1 CLK/CLKS select bit 1 (Note) | 0 : CLK output is only CLK1<br>1 : Transfer clock output from multiple pins function selected | RW |
| RCSP | Separate UART0 CTS/RTS bit | 0 : $\overline{CTS}$/$\overline{RTS}$ shared pin<br>1 : $\overline{CTS}$/$\overline{RTS}$ separated ($\overline{CTS}$0 supplied from the P64 pin) | RW |
| $\overline{\quad}$ (b7) | Nothing is assigned. When write, set "0". When read, its content is indeterminate. | | — |

Note: When using multiple transfer clock output pins, make sure the following conditions are met:
U1MR register's CKDIR bit = "0" (internal clock)

**Figure 14.1.6. Serial I/O-related registers (3)**

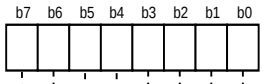
### UARTi transmit/receive control register 1 (i=0, 1)




Symbol: U0C1, U1C1 — Address: 03A5$_{16}$,03AD$_{16}$ — After reset: 00000010$_2$

| 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 present in UiTB register<br>1 : No data present in UiTB register | RO |
| RE | Receive enable bit | 0 : Reception disabled<br>1 : Reception enabled | RW |
| RI | Receive complete flag | 0 : No data present in UiRB register<br>1 : Data present in UiRB register | RO |
| — (b7-b4) | Nothing is assigned.<br>When write, set “0”. When read, these contents are “0”. | | — |

### UART2 transmit/receive control register 1

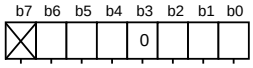



Symbol: U2C1 — Address: 037D$_{16}$ — After reset: 00000010$_2$

| 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 present in U2TB register<br>1 : No data present in U2TB register | RO |
| RE | Receive enable bit | 0 : Reception disabled<br>1 : Reception enabled | RW |
| RI | Receive complete flag | 0 : No data present in U2RB register<br>1 : Data present in U2RB register | RO |
| U2IRS | UART2 transmit interrupt cause select bit | 0 : Transmit buffer empty (TI = 1)<br>1 : Transmit is completed (TXEPT = 1) | RW |
| U2RRM | UART2 continuous receive mode enable bit | 0 : Continuous receive mode disabled<br>1 : Continuous receive mode enabled | RW |
| U2LCH | Data logic select bit | 0 : No reverse<br>1 : Reverse | RW |
| U2ERE | Error signal output enable bit | 0 : Output disabled<br>1 : Output enabled | RW |

**Figure 14.1.7. Serial I/O-related registers (4)**

UART2 special mode register

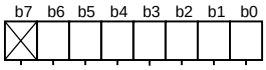



Symbol U2SMR | Address $037716$ | After reset $X0000000_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| IICM | $I^2C$ mode select bit | 0 : Other than $I^2C$ mode<br>1 : $I^2C$ mode | RW |
| ABC | Arbitration lost detecting flag control bit | 0 : Update per bit<br>1 : Update per byte | RW |
| BBS | Bus busy flag | 0 : STOP condition detected<br>1 : START condition detected (busy) | RW (Note1) |
| — (b3) | Reserved bit | Set to “0” | RW |
| ABSCS | Bus collision detect sampling clock select bit | 0 : Rising edge of transfer clock<br>1 : Underflow signal of timer A0 | RW |
| ACSE | Auto clear function select bit of transmit enable bit | 0 : No auto clear function<br>1 : Auto clear at occurrence of bus collision | RW |
| SSS | Transmit start condition select bit | 0 : Not synchronized to RxDi<br>1 : Synchronized to RxDi (Note 2) | RW |
| — (b7) | Nothing is assigned. When write, set “0”. When read, its content is indeterminate. | | — |

Note 1: The BBS bit is set to “0” by writing “0” in a program. (Writing “1” has no effect.).
Note 2: When a transfer begins, the SSS bit is set to “0” (Not synchronized to RxDi).

UART2 special mode register 2

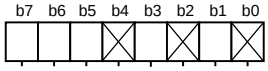



Symbol U2SMR2 | Address $037616$ | After reset $X0000000_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| IICM2 | $I^2C$ mode select bit 2 | Refer to “Table 141.3.4. $I^2C$ Mode Functions” | RW |
| CSC | Clock-synchronous bit | 0 : Disabled<br>1 : Enabled | RW |
| SWC | SCL wait output bit | 0 : Disabled<br>1 : Enabled | RW |
| ALS | SDA output stop bit | 0 : Disabled<br>1 : Enabled | RW |
| STAC | UART initialization bit | 0 : Disabled<br>1 : Enabled | RW |
| SWC2 | SCL wait output bit 2 | 0: Transfer clock<br>1: “L” output | RW |
| SDHI | SDA output disable bit | 0: Enabled<br>1: Disabled (high impedance) | RW |
| — (b7) | Nothing is assigned. When write, set “0”. When read, its content is indeterminate. | | — |

**Figure 14.1.8. Serial I/O-related registers (5)**

UART2 special mode register 3

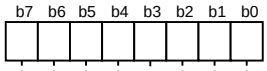



Symbol: U2SMR3 | Address: 037516 | After reset: 000X0X0X2

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| — (b0) | Nothing is assigned. When write, set "0". When read, its content is indeterminate. | | — |
| CKPH | Clock phase set bit | 0 : Without clock delay<br>1 : With clock delay | RW |
| — (b2) | Nothing is assigned. When write, set "0". When read, its content is indeterminate. | | — |
| NODC | Clock output select bit | 0 : CLKi is CMOS output<br>1 : CLKi is N-channel open drain output | RW |
| — (b4) | Nothing is assigned. When write, set "0". When read, its content is indeterminate. | | — |
| DL0 | SDA digital delay setup bit (Note 1, Note 2) | b7 b6 b5<br>0 0 0 : Without delay<br>0 0 1 : 1 to 2 cycle(s) of UiBRG count source<br>0 1 0 : 2 to 3 cycles of UiBRG count source<br>0 1 1 : 3 to 4 cycles of UiBRG count source<br>1 0 0 : 4 to 5 cycles of UiBRG count source<br>1 0 1 : 5 to 6 cycles of UiBRG count source<br>1 1 0 : 6 to 7 cycles of UiBRG count source<br>1 1 1 : 7 to 8 cycles of UiBRG count source | RW |
| DL1 | | | RW |
| DL2 | | | RW |

Note 1 : The DL2 to DL0 bits are used to generate a delay in SDAi output by digital means during I2C mode. In other than I2C mode, set these bits to "0002" (no delay).
Note 2 : The amount of delay varies with the load on SCL2 and SDA2 pins. Also, when using an external clock, the amount of delay increases by about 100 ns.

UART2 special mode register 4




Symbol: U2SMR4 | Address: 037416 | After reset: 0016

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| STAREQ | Start condition generate bit (Note) | 0 : Clear<br>1 : Start | RW |
| RSTAREQ | Restart condition generate bit (Note) | 0 : Clear<br>1 : Start | RW |
| STPREQ | Stop condition generate bit (Note) | 0 : Clear<br>1 : Start | RW |
| STSPSEL | SCL,SDA output select bit | 0 : Start and stop conditions not output<br>1 : Start and stop conditions output | RW |
| ACKD | ACK data bit | 0 : ACK<br>1 : NACK | RW |
| ACKC | ACK data output enable bit | 0 : Serial I/O data output<br>1 : ACK data output | RW |
| SCLHI | SCL output stop enable bit | 0 : Disabled<br>1 : Enabled | RW |
| SWC9 | SCL wait bit 3 | 0 : SCL "L" hold disabled<br>1 : SCL "L" hold enabled | RW |

Note: Set to "0" when each condition is generated.

**Figure 14.1.9. Serial I/O-related registers (6)**

### 14.1.1. Clock Synchronous serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Table 14.1.1.1 lists the specifications of the clock synchronous serial I/O mode. Table 14.1.1.2 lists the registers used in clock synchronous serial I/O mode and the register values set.

**Table 14.1.1.1. Clock Synchronous Serial I/O Mode Specifications**

| Item | Specification |
|---|---|
| Transfer data format | • Transfer data length: 8 bits |
| Transfer clock | • UiMR(i=0 to 2) register's CKDIR bit = "0" (internal clock) : fj/ 2(n+1)<br>fj = $f_{1SIO}$, $f_{2SIO}$, $f_{8SIO}$, $f_{32SIO}$. n: Setting value of UiBRG register $00_{16}$ to $FF_{16}$<br>• CKDIR bit = "1" (external clock) : Input from CLKi pin |
| Transmission, reception control | • Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS}/\overline{RTS}$ function disable |
| Transmission start condition | • Before transmission can start, the following requirements must be met (Note 1)<br>– The TE bit of UiC1 register= 1 (transmission enabled)<br>– The TI bit of UiC1 register = 0 (data present in UiTB register)<br>– If $\overline{CTS}$ function is selected, input on the $\overline{CTS}$i pin = "L" |
| Reception start condition | • Before reception can start, the following requirements must be met (Note 1)<br>– The RE bit of UiC1 register= 1 (reception enabled)<br>– The TE bit of UiC1 register= 1 (transmission enabled)<br>– The TI bit of UiC1 register= 0 (data present in the UiTB register) |
| Interrupt request generation timing | • For transmission, one of the following conditions can be selected<br>– The UiIRS bit (Note 3) = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission)<br>– The UiIRS bit =1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register<br>• For reception<br>When transferring data from the UARTi receive register to the UiRB register (at completion of reception) |
| Error detection | • Overrun error (Note 2)<br>This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 7th bit of the next data |
| Select function | • CLK polarity selection<br>Transfer data input/output can be chosen to occur synchronously with the rising or the falling edge of the transfer clock<br>• LSB first, MSB first selection<br>Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected<br>• Continuous receive mode selection<br>Reception is enabled immediately by reading the UiRB register<br>• Switching serial data logic (UART2)<br>This function reverses the logic value of the transmit/receive data<br>• Transfer clock output from multiple pins selection (UART1)<br>The output pin can be selected in a program from two UART1 transfer clock pins that have been set<br>• Separate $\overline{CTS}/\overline{RTS}$ pins (UART0)<br>$\overline{CTS_0}$ and $\overline{RTS_0}$ are input/output from separate pins |

Note 1: When an external clock is selected, the conditions must be met while if the UiC0 register's CKPOL bit = "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the UiC0 register's CKPOL bit = "1" (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
Note 2: If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit of SiRIC register does not change.
Note 3: The U0IRS and U1IRS bits respectively are the UCON register bits 0 and 1; the U2IRS bit is the U2C1 register bit 4.

**Table 14.1.1. 2. Registers to Be Used and Settings in Clock Synchronous Serial I/O Mode**

| Register | Bit | Function |
|---|---|---|
| UiTB(Note3) | 0 to 7 | Set transmission data |
| UiRB(Note3) | 0 to 7 | Reception data can be read |
| | OER | Overrun error flag |
| UiBRG | 0 to 7 | Set a transfer rate |
| UiMR(Note3) | SMD2 to SMD0 | Set to “$001_2$” |
| | CKDIR | Select the internal clock or external clock |
| | IOPOL(i=2)(Note 4) | Set to “0” |
| UiC0 | CLK1 to CLK0 | Select the count source for the UiBRG register |
| | CRS | Select $\overline{CTS}$ or $\overline{RTS}$ to use |
| | TXEPT | Transmit register empty flag |
| | CRD | Enable or disable the $\overline{CTS}$ or $\overline{RTS}$ function |
| | NCH | Select TxDi pin output mode |
| | CKPOL | Select the transfer clock polarity |
| | UFORM | Select the LSB first or MSB first |
| UiC1 | TE | Set this bit to “1” to enable transmission/reception |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to “1” to enable reception |
| | RI | Reception complete flag |
| | U2IRS (Note 1) | Select the source of UART2 transmit interrupt |
| | U2RRM (Note 1) | Set this bit to “1” to use UART2 continuous receive mode |
| | U2LCH(Note 3) | Set this bit to “1” to use UART2 inverted data logic |
| | U2ERE(Note 3) | Set to “0” |
| UiSMR | 0 to 7 | Set to “0” |
| UiSMR2 | 0 to 7 | Set to “0” |
| UiSMR3 | 0 to 2 | Set to “0” |
| | NODC | Select clock output mode |
| | 4 to 7 | Set to “0” |
| UiSMR4 | 0 to 7 | Set to “0” |
| UCON | U0IRS, U1IRS | Select the source of UART0/UART1 transmit interrupt |
| | U0RRM, U1RRM | Set this bit to “1” to use continuous receive mode |
| | CLKMD0 | Select the transfer clock output pin when CLKMD1 = 1 |
| | CLKMD1 | Set this bit to “1” to output UART1 transfer clock from two pins |
| | RCSP | Set this bit to “1” to accept as input the UART0 $\overline{CTS_0}$ signal from the $P6_4$ pin |
| | 7 | Set to “0” |

Note 1: Set the U0C1 and U1C1 register bit 4 and bit 5 to “0”. The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.

Note 2: Not all register bits are described above. Set those bits to “0” when writing to the registers in clock synchronous serial I/O mode.

Note 3: Set the U0C1 and U1C1 register bit 6 and bit 7 to "0".

Note 4: Set the U0MR and U1MR register bit 7 to "0".

i=0 to 2

Table 14.1.1.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. Table 14.3 shows pin functions for the case where the multiple transfer clock output pin select function is deselected. Table 14.1.1.4 lists the P6$_4$ pin functions during clock synchronous serial I/O mode.
Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an "H". (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

**Table 14.1.1.3. Pin Functions (When Not Select Multiple Transfer Clock Output Pin Function)**

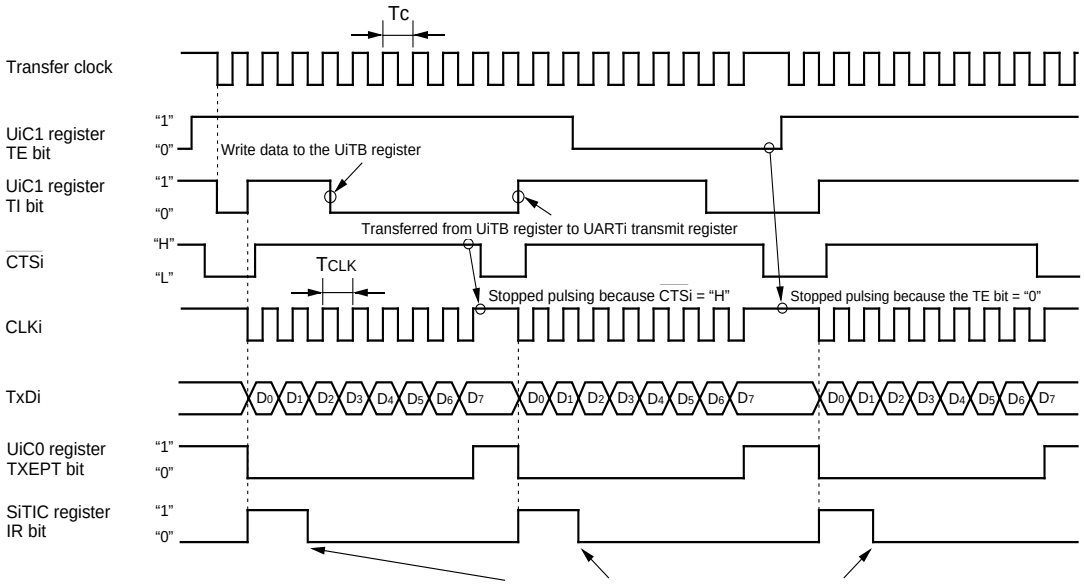
| Pin name | Function | Method of selection |
|---|---|---|
| TxDi (i = 0 to 2) (P6$_3$, P6$_7$, P7$_0$) | Serial data output | (Outputs dummy data when performing reception only) |
| RxDi (P6$_2$, P6$_6$, P7$_1$) | Serial data input | PD6 register's PD6_2 bit=0, PD6_6 bit=0, PD7 register's PD7_1 bit=0<br>(Can be used as an input/output port when performing transmission only) |
| CLKi (P6$_1$, P6$_5$, P7$_2$) | Transfer clock output | UiMR register's CKDIR bit=0 |
| | Transfer clock input | UiMR register's CKDIR bit=1<br>PD6 register's PD6_1 bit=0, PD6_5 bit=0, PD7 register's PD7_2 bit=0 |
| $\overline{\text{CTS}}$i/$\overline{\text{RTS}}$i (P6$_0$, P6$_4$, P7$_3$) | $\overline{\text{CTS}}$ input | UiC0 register's CRD bit=0<br>UiC0 register's CRS bit=0<br>PD6 register's PD6_0 bit=0, PD6_4 bit=0, PD7 register's PD7_3 bit=0 |
| | $\overline{\text{RTS}}$ output | UiC0 register's CRD bit=0<br>UiC0 register's CRS bit=1 |
| | I/O port | UiC0 register's CRD bit=1 |

**Table 14.1.1.4. P6$_4$ Pin Functions**

| Pin function | Bit set value | | | | | |
|---|---|---|---|---|---|---|
| | U1C0 register | | UCON register | | | PD6 register |
| | CRD | CRS | RCSP | CLKMD1 | CLKMD0 | PD6_4 |
| P6$_4$ | 1 | — | 0 | 0 | — | Input: 0, Output: 1 |
| $\overline{\text{CTS}}_1$ | 0 | 0 | 0 | 0 | — | 0 |
| $\overline{\text{RTS}}_1$ | 0 | 1 | 0 | 0 | — | — |
| $\overline{\text{CTS}}_0$(Note1) | 0 | 0 | 1 | 0 | — | 0 |
| CLKS$_1$ | — | — | — | 1(Note 2) | 1 | — |

Note 1: In addition to this, set the U0C0 register's CRD bit to "0" ($\overline{\text{CTS}}_0$/$\overline{\text{RTS}}_0$ enabled) and the U0C0 register's CRS bit to "1" ($\overline{\text{RTS}}_0$ selected).
Note 2: When the CLKMD1 bit = 1 and the CLKMD0 bit = 0, the following logic levels are output:
- High if the U1C0 register's CLKPOL bit = 0
- Low if the U1C0 register's CLKPOL bit = 1

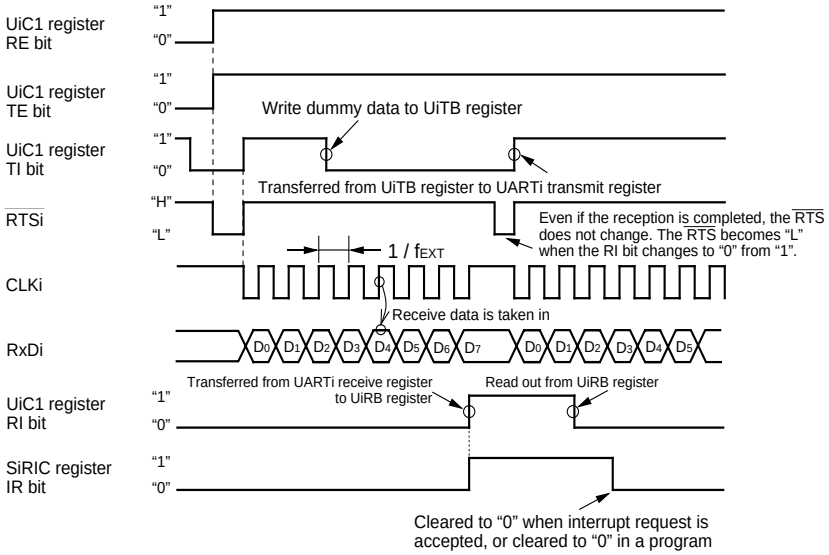
(1) Example of transmit timing




Tc = TCLK = 2(n + 1) / fj
fj: frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
n: value set to UiBRG register
i: 0 to 2

The above timing diagram applies to the case where the register bits are set as follows:
- UiMR register CKDIR bit = 0 (internal clock)
- UiC0 register CRD bit = 0 (CTS/RTS enabled), CRS bit = 0 (CTS selected)
- UiC0 register CKPOL bit = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)
- UiIRS bit = 0 (an interrupt request occurs when the transmit buffer becomes empty): U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

(2) Example of receive timing




The above timing diagram applies to the case where the register bits are set as follows:
- UiMR register CKDIR bit = 1 (external clock)
- UiC0 register CRD bit = 0 (CTS/RTS enabled), CRS bit = 1 (RTS selected)
- UiC0 register CKPOL bit = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)

Make sure the following conditions are met when input to the CLKi pin before receiving data is high:
- UiC0 register TE bit = 1 (transmit enabled)
- UiC0 register RE bit = 1 (Receive enabled)
- Write dummy data to the UiTB register

**Figure 14.1.1.1. Typical transmit/receive timings in clock synchronous serial I/O mode**

#### 14.1.1.1 CLK Polarity Select Function

Use the UiC0 register (i = 0 to 2)'s CKPOL bit to select the transfer clock polarity. Figure 14.1.1.1.1 shows the polarity of the transfer clock.

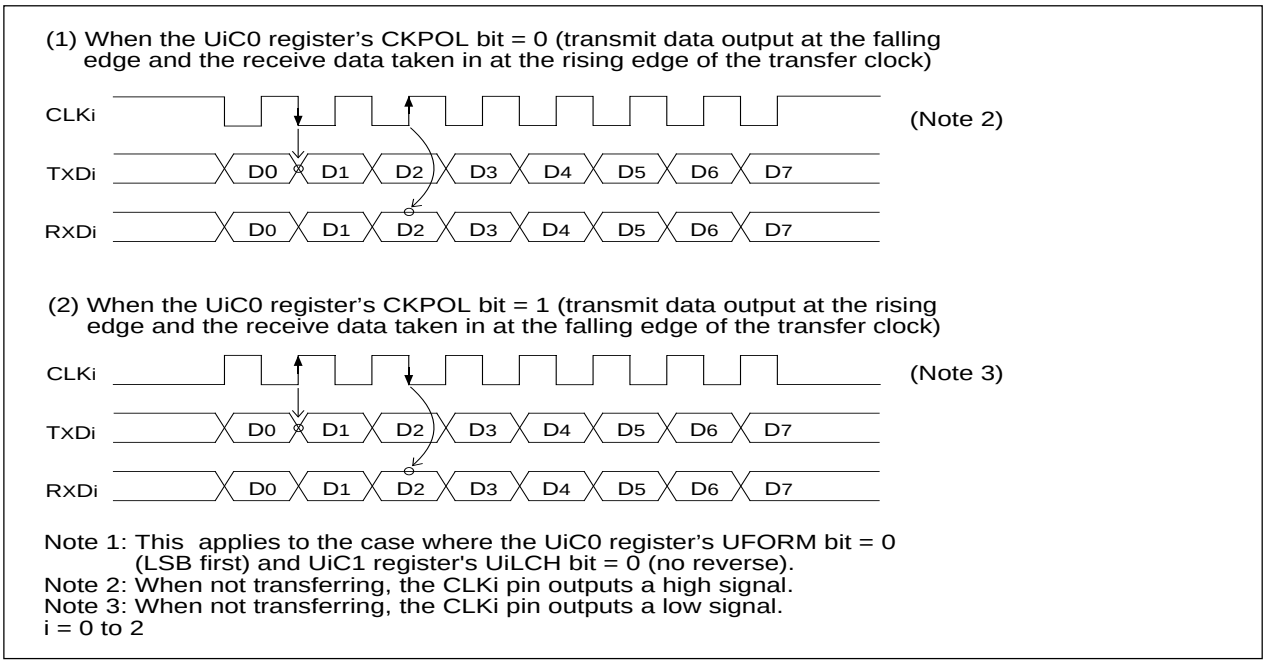



**Figure 14.1.1.1.1. Polarity of transfer clock**

#### 14.1.1.2 LSB First/MSB First Select Function

Use the UiC0 register (i = 0 to 2)'s UFORM bit to select the transfer format. Figure 14.1.1.2.1 shows the transfer format.

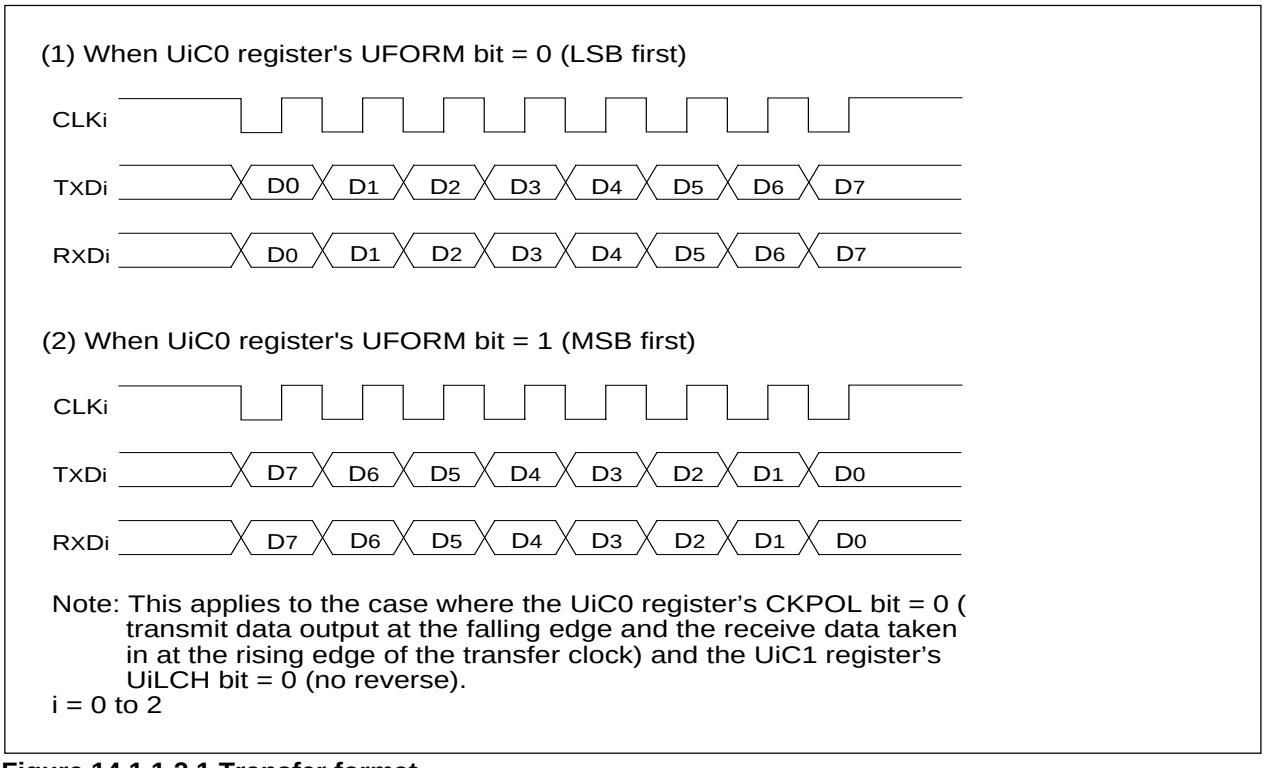


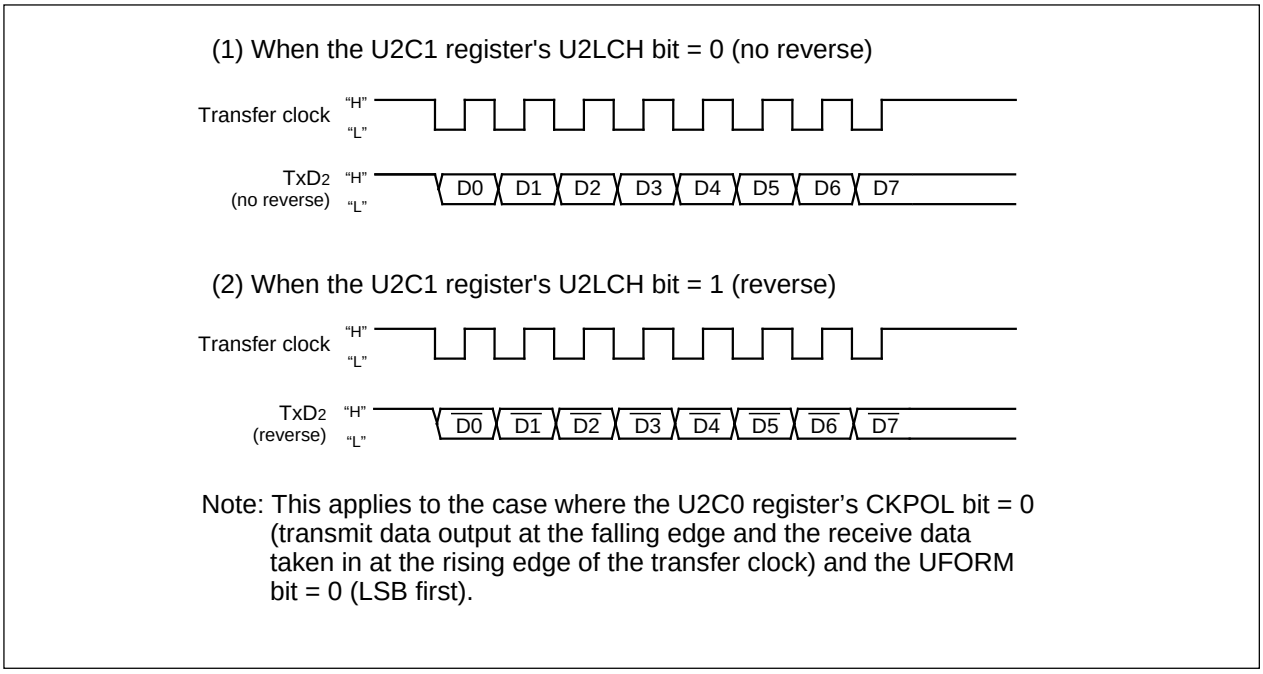

**Figure 14.1.1.2.1 Transfer format**

#### 14.1.1.3 Continuous receive mode

When the UiRRM bit (i = 0 to 2) = 1 (continuous receive mode), the UiC1 register's TI bit is set to "0" (data present in the UiTB register) by reading the UiRB register. In this case, i.e., UiRRM bit = 1, do not write dummy data to the UiTB register in a program. The U0RRM and U1RRM bits are the UCON register bit 2 and bit 3, respectively, and the U2RRM bit is the U2C1 register bit 5.

#### 14.1.1.4 Serial data logic switch function (UART2)

When the U2C1 register's U2LCH bit = 1 (reverse), the data written to the U2TB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the U2RB register. Figure 14.1.1.4.1 shows serial data logic.

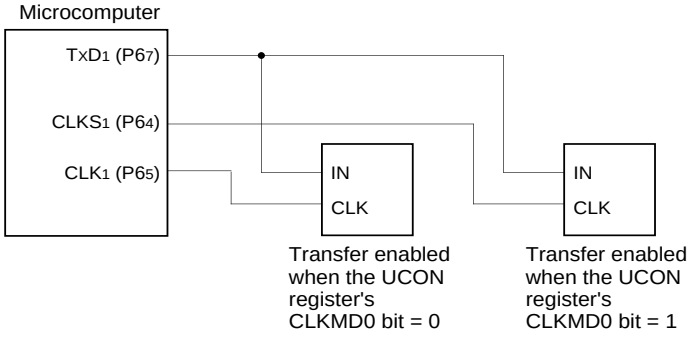



Note: This applies to the case where the U2C0 register's CKPOL bit = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock) and the UFORM bit = 0 (LSB first).

**Figure 14.1.1.4.1. Serial data logic switch timing**

#### 14.1.1.5 Transfer clock output from multiple pins function (UART1)

This function allows the setting two transfer clock output pins and choosing one of the two to output a clock by using the CLK and CLKS select bit (bits 4 and 5 at address $03B0_{16}$). (See Figure 14.1.1.5.1.) The multiple pins function is valid only when the internal clock is selected for UART1.




Note: This applies to the case where the U1MRregister's CKDIR bit = 0 (internal clock) and the UCON register's CLKMD1 bit = 1 ( transfer clock output from multiple pins).

**Figure 14.1.1.5.1 Transfer Clock Output From Multiple Pins**

### 14.1.1.6 $\overline{CTS}/\overline{RTS}$ separate function (UART0)

This function separates $\overline{CTS}_0/\overline{RTS}_0$, outputs $\overline{RTS}_0$ from the $P6_0$ pin, and accepts as input the $\overline{CTS}_0$ from the $P6_4$ pin. To use this function, set the register bits as shown below.

- U0C0 register's CRD bit = 0 (enables UART0 $\overline{CTS}/\overline{RTS}$)
- U0C0 register's CRS bit = 1 (outputs UART0 $\overline{RTS}$)
- U1C0 register's CRD bit = 0 (enables UART1 $\overline{CTS}/\overline{RTS}$)
- U1C0 register's CRS bit = 0 (inputs UART1 $\overline{CTS}$)
- UCON register's RCSP bit = 1 (inputs $\overline{CTS}_0$ from the $P6_4$ pin)
- UCON register's CLKMD1 bit = 0 ($CLKS_1$ not used)

Note that when using the $\overline{CTS}/\overline{RTS}$ separate function, UART1 $\overline{CTS}/\overline{RTS}$ separate function cannot be used.

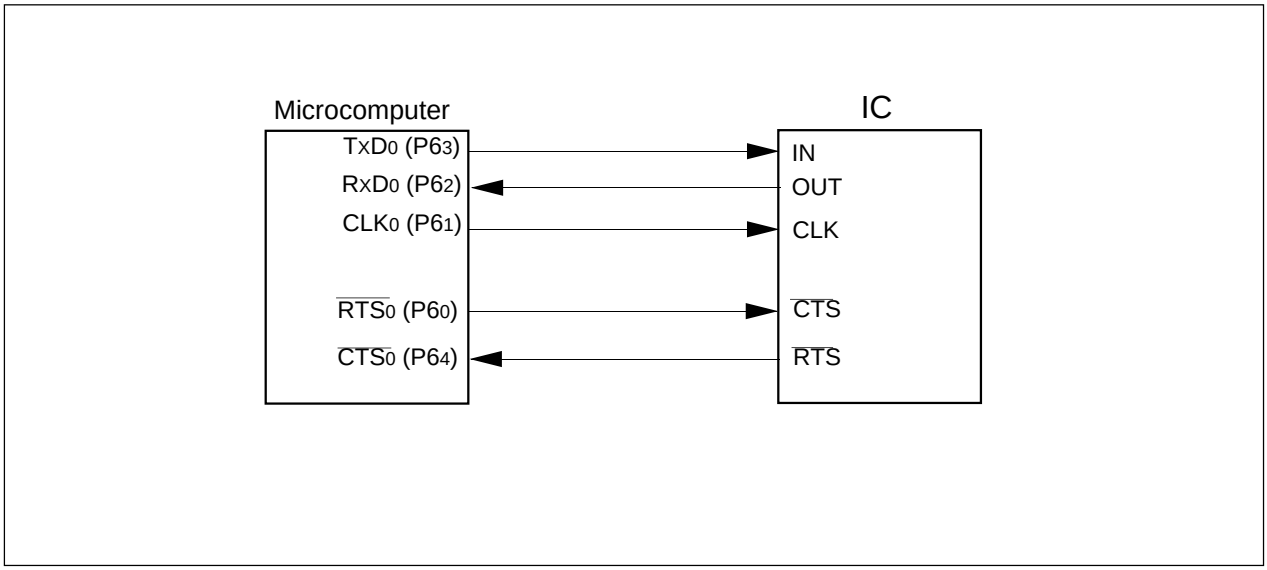



**Figure 14.1.1.6.1. CTS/RTS separate function usage**

## 14.1.2. Clock Asynchronous Serial I/O (UART) Mode

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. Tables 14.1.2.1 lists the specifications of the UART mode.

**Table 14.1.2.1. UART Mode Specifications**

| Item | Specification |
|---|---|
| Transfer data format | • Character bit (transfer data): Selectable from 7, 8 or 9 bits<br>• Start bit: 1 bit<br>• Parity bit: Selectable from odd, even, or none<br>• Stop bit: Selectable from 1 or 2 bits |
| Transfer clock | • UiMR(i=0 to 2) register's CKDIR bit = 0 (internal clock) : fj/ 16(n+1)<br>fj = $f_{1SIO}$, $f_{2SIO}$, $f_{8SIO}$, $f_{32SIO}$. n: Setting value of UiBRG register $00_{16}$ to $FF_{16}$<br>• CKDIR bit = "1" (external clock) : $f_{EXT}$/16(n+1)<br>$f_{EXT}$: Input from CLKi pin. n :Setting value of UiBRG register $00_{16}$ to $FF_{16}$ |
| Transmission, reception control | • Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS}/\overline{RTS}$ function disable |
| Transmission start condition | • Before transmission can start, the following requirements must be met<br>– The TE bit of UiC1 register= 1 (transmission enabled)<br>– The TI bit of UiC1 register = 0 (data present in UiTB register)<br>– If $\overline{CTS}$ function is selected, input on the $\overline{CTS}$i pin = "L" |
| Reception start condition | • Before reception can start, the following requirements must be met<br>– The RE bit of UiC1 register= 1 (reception enabled)<br>– Start bit detection |
| Interrupt request generation timing | • For transmission, one of the following conditions can be selected<br>– The UiIRS bit (Note 2) = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission)<br>– The UiIRS bit =1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register<br>• For reception<br>When transferring data from the UARTi receive register to the UiRB register (at completion of reception) |
| Error detection | • Overrun error (Note 1)<br>This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the bit one before the last stop bit of the next data<br>• Framing error<br>This error occurs when the number of stop bits set is not detected<br>• Parity error<br>This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set<br>• Error sum flag<br>This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered |
| Select function | • LSB first, MSB first selection<br>Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected<br>• Serial data logic switch (UART2)<br>This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed.<br>• TxD, RxD I/O polarity switch (UART2)<br>This function reverses the polarities of hte TxD pin output and RxD pin input. The logic levels of all I/O data is reversed.<br>• Separate $\overline{CTS}/\overline{RTS}$ pins (UART0)<br>$\overline{CTS_0}$ and $\overline{RTS_0}$ are input/output from separate pins |

Note 1: If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit of SiRIC register does not change.
Note 2: The U0IRS and U1IRS bits respectively are the UCON register bits 0 and 1; the U2IRS bit is the U2C1 register bit 4.

**Table 14.1.2.2. Registers to Be Used and Settings in UART Mode**

| Register | Bit | Function |
|---|---|---|
| UiTB | 0 to 8 | Set transmission data (Note 1) |
| UiRB | 0 to 8 | Reception data can be read (Note 1) |
| | OER,FER,PER,SUM | Error flag |
| UiBRG | 0 to 7 | Set a transfer rate |
| UiMR | SMD2 to SMD0 | Set these bits to '$100_2$' when transfer data is 7 bits long<br>Set these bits to '$101_2$' when transfer data is 8 bits long<br>Set these bits to '$110_2$' when transfer data is 9 bits long |
| | CKDIR | Select the internal clock or external clock |
| | STPS | Select the stop bit |
| | PRY, PRYE | Select whether parity is included and whether odd or even |
| | IOPOL(i=2)(Note 4) | Select the TxD/RxD input/output polarity |
| UiC0 | CLK0, CLK1 | Select the count source for the UiBRG register |
| | CRS | Select $\overline{CTS}$ or $\overline{RTS}$ to use |
| | TXEPT | Transmit register empty flag |
| | CRD | Enable or disable the $\overline{CTS}$ or $\overline{RTS}$ function |
| | NCH | Select TxDi pin output mode |
| | CKPOL | Set to "0" |
| | UFORM | LSB first or MSB first can be selected when transfer data is 8 bits long. Set this bit to "0" when transfer data is 7 or 9 bits long. |
| UiC1 | TE | Set this bit to "1" to enable transmission |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to "1" to enable reception |
| | RI | Reception complete flag |
| | U2IRS (Note 2) | Select the source of UART2 transmit interrupt |
| | U2RRM (Note 2) | Set to "0" |
| | UiLCH (Note 3) | Set this bit to "1" to use UART2 inverted data logic |
| | UiERE (Note 3) | Set to "0" |
| UiSMR | 0 to 7 | Set to "0" |
| UiSMR2 | 0 to 7 | Set to "0" |
| UiSMR3 | 0 to 7 | Set to "0" |
| UiSMR4 | 0 to 7 | Set to "0" |
| UCON | U0IRS, U1IRS | Select the source of UART0/UART1 transmit interrupt |
| | U0RRM, U1RRM | Set to "0" |
| | CLKMD0 | Invalid because CLKMD1 = 0 |
| | CLKMD1 | Set to "0" |
| | RCSP | Set this bit to "1" to accept as input the UART0 $\overline{CTS_0}$ signal from the $P6_4$ pin |
| | 7 | Set to "0" |

Note 1: The bits used for transmit/receive data are as follows: Bit 0 to bit 6 when transfer data is 7 bits long; bit 0 to bit 7 when transfer data is 8 bits long; bit 0 to bit 8 when transfer data is 9 bits long.

Note 2: Set the U0C1 and U1C1 registers bit 4 to bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are included in the UCON register.

Note 3: Set the U0C1 and U1C1 registers bit 6 to bit 7 to "0".

Note 4: Set the U0MR and U1MR registers bit 7 to "0".

i=0 to 2

Table 14.1.2.3 lists the functions of the input/output pins during UART mode. Table 14.1.2.4 lists the P6$_4$ pin functions during UART mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an “H”. (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

**Table 14.1.2.3. I/O Pin Functions in UART mode**

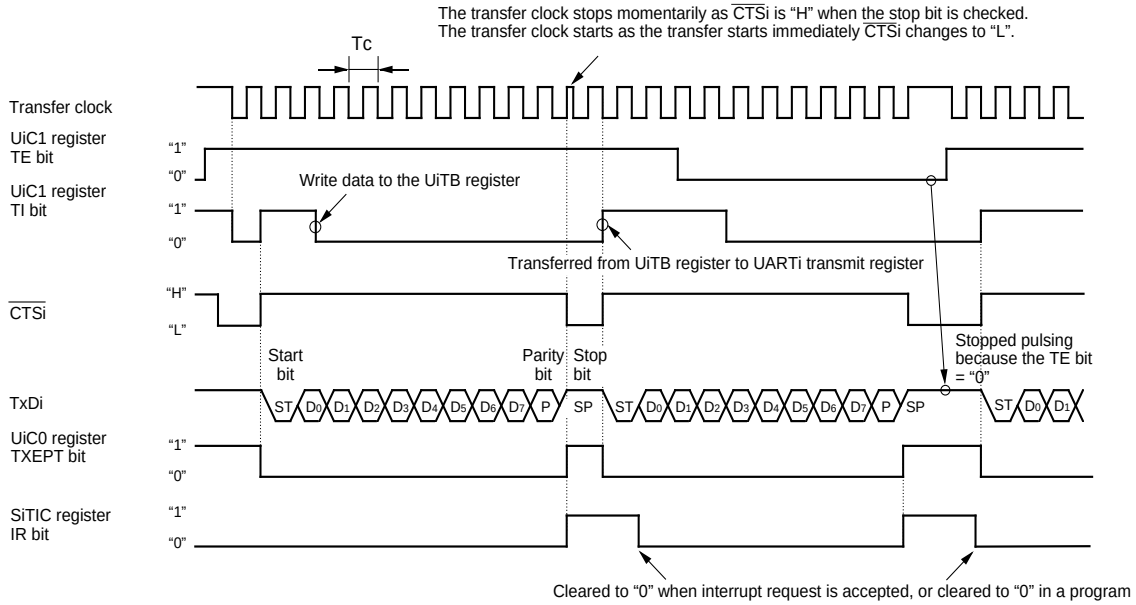
| Pin name | Function | Method of selection |
|---|---|---|
| TxDi (i = 0 to 2) (P6$_3$, P6$_7$, P7$_0$) | Serial data output | (Outputs "H" when performing reception only) |
| RxDi (P6$_2$, P6$_6$, P7$_1$) | Serial data input | PD6 register’s PD6_2 bit=0, PD6_6 bit=0, PD7 register’s PD7_1 bit=0<br>(Can be used as an input/output port when performing transmission only) |
| CLKi (P6$_1$, P6$_5$, P7$_2$) | Input/output port | UiMR register’s CKDIR bit=0 |
| | Transfer clock input | UiMR register’s CKDIR bit=1<br>PD6 register’s PD6_1 bit=0, PD6_5 bit=0, PD7 register’s PD7_2 bit=0 |
| $\overline{\text{CTS}}$i/$\overline{\text{RTS}}$i (P6$_0$, P6$_4$, P7$_3$) | $\overline{\text{CTS}}$ input | UiC0 register’s CRD bit=0<br>UiC0 register’s CRS bit=0<br>PD6 register’s PD6_0 bit=0, PD6_4 bit=0, PD7 register’s PD7_3 bit=0 |
| | $\overline{\text{RTS}}$ output | UiC0 register’s CRD bit=0<br>UiC0 register’s CRS bit=1 |
| | Input/output port | UiC0 register’s CRD bit=1 |

**Table 14.1.2.4. P6$_4$ Pin Functions in UART mode**

| Pin function | Bit set value | | | | |
|---|---|---|---|---|---|
| | U1C0 register | | UCON register | | PD6 register |
| | CRD | CRS | RCSP | CLKMD1 | PD6_4 |
| P6$_4$ | 1 | — | 0 | 0 | Input: 0, Output: 1 |
| $\overline{\text{CTS}}_1$ | 0 | 0 | 0 | 0 | 0 |
| $\overline{\text{RTS}}_1$ | 0 | 1 | 0 | 0 | — |
| $\overline{\text{CTS}}_0$ (Note) | 0 | 0 | 1 | 0 | 0 |

Note: In addition to this, set the U0C0 register’s CRD bit to “0” ($\overline{\text{CTS}}_0$/$\overline{\text{RTS}}_0$ enabled) and the U0C0 register’s CRS bit to “1” ($\overline{\text{RTS}}_0$ selected).

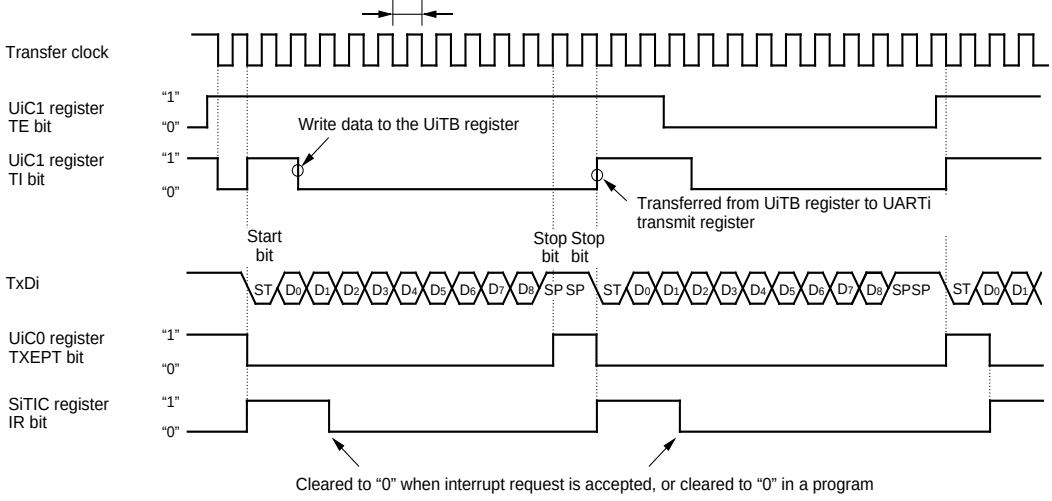
• Example of transmit timing when transfer data is 8 bits long (parity enabled, one stop bit)




The above timing diagram applies to the case where the register bits are set as follows:
- UiMR register PRYE bit = 1 (parity enabled)
- UiMR register STPS bit = 0 (1 stop bit)
- UiC0 register CRD bit = 0 (CTS/RTS enabled), CRS bit = 0 (CTS selected)
- UiIRS bit = 1 (an interrupt request occurs when transmit completed):
  U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

Tc = 16 (n + 1) / fj or 16 (n + 1) / $f_{EXT}$
fj : frequency of UiBRG count source ($f_{1SIO}$, $f_{2SIO}$, $f_{8SIO}$, $f_{32SIO}$)
$f_{EXT}$ : frequency of UiBRG count source (external clock)
n : value set to UiBRG
i: 0 to 2

• Example of transmit timing when transfer data is 9 bits long (parity disabled, two stop bits)

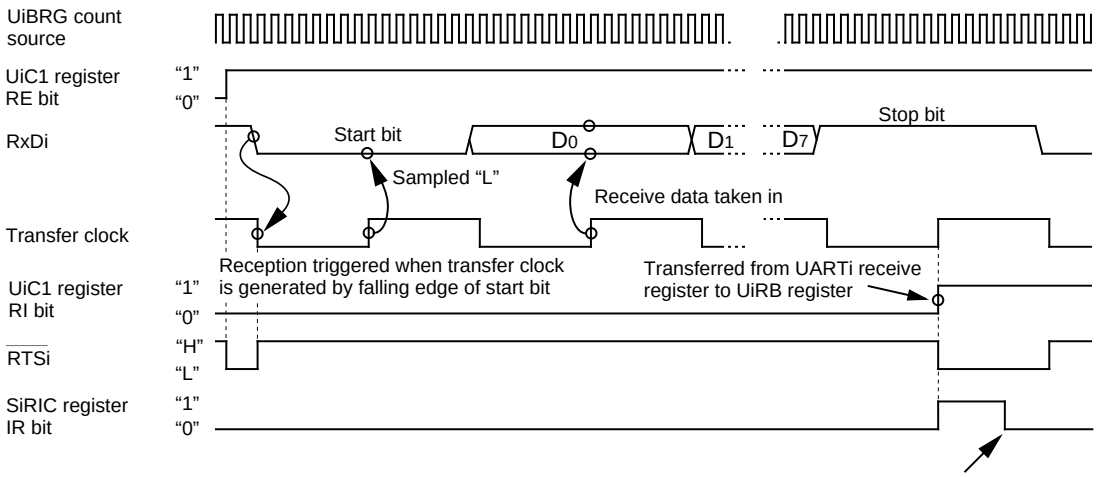



The above timing diagram applies to the case where the register bits are set as follows:
- UiMR register PRYE bit = 0 (parity disabled)
- UiMR register STPS bit = 1 (2 stop bits)
- UiC0 register CRD bit = 1 (CTS/RTS disabled)
- UiIRS bit = 0 (an interrupt request occurs when transmit buffer becomes empty):
  U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

Tc = 16 (n + 1) / fj or 16 (n + 1) / $f_{EXT}$
fj : frequency of UiBRG count source ($f_{1SIO}$, $f_{2SIO}$, $f_{8SIO}$, $f_{32SIO}$)
$f_{EXT}$ : frequency of UiBRG count source (external clock)
n : value set to UiBRG
i: 0 to 2

**Figure 14.1.2.1. Typical transmit timing in UART mode (UART0, UART1)**

• Example of receive timing when transfer data is 8 bits long (parity disabled, one stop bit)



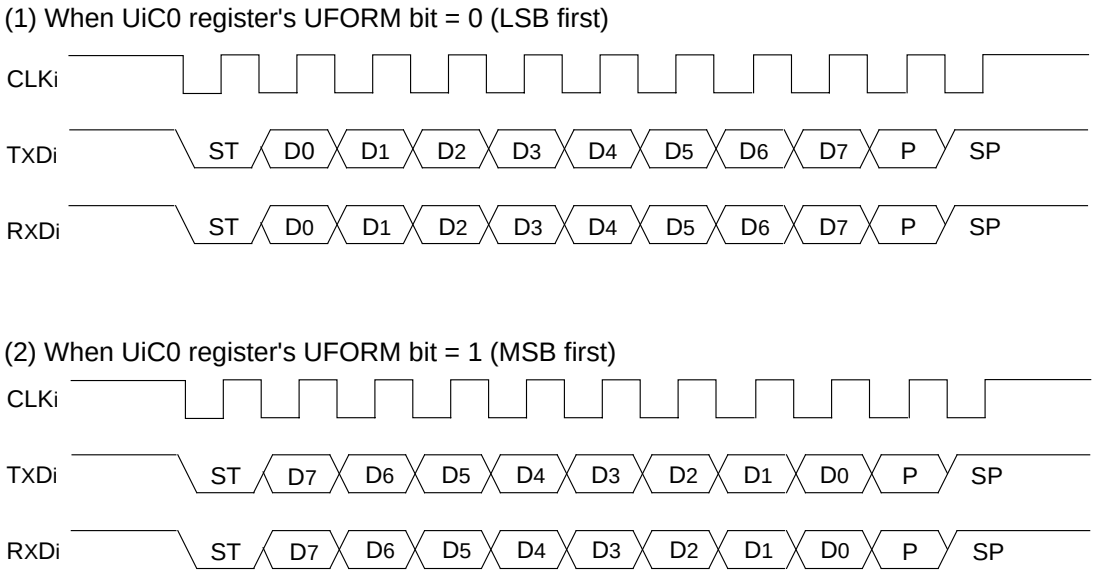

The above timing diagram applies to the case where the register bits are set as follows:
- UiMR register PRYE bit = 0 (parity disabled)
- UiMR register STPS bit = 0 (1 stop bit)
- UiC0 register CRD bit = 0 (CTSi/RTSi enabled), CRS bit = 1 (RTSi selected)

i = 0 to 2

**Figure 14.1.2.2. Receive Operation**

### 14.1.2.1. LSB First/MSB First Select Function

As shown in Figure 14.1.2.1.1, use the UiC0 register's UFORM bit to select the transfer format. This function is valid when transfer data is 8 bits long.




Note: This applies to the case where the UiC0 register's CKPOL bit = 0 ( transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the UiC1 register's UiLCH bit = 0 (no reverse), UiMR register's STPS bit = 0 (1 stop bit) and UiMR register's PRYE bit = 1 (parity enabled).

ST : Start bit
P : Parity bit
SP : Stop bit
i = 0 to 2

**Figure 14.1.2.1.1. Transfer Format**

### 14.1.2.2. Serial Data Logic Switching Function (UART2)

The data written to the U2TB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the U2RB register. Figure 14.1.2.2.1 shows serial data logic.

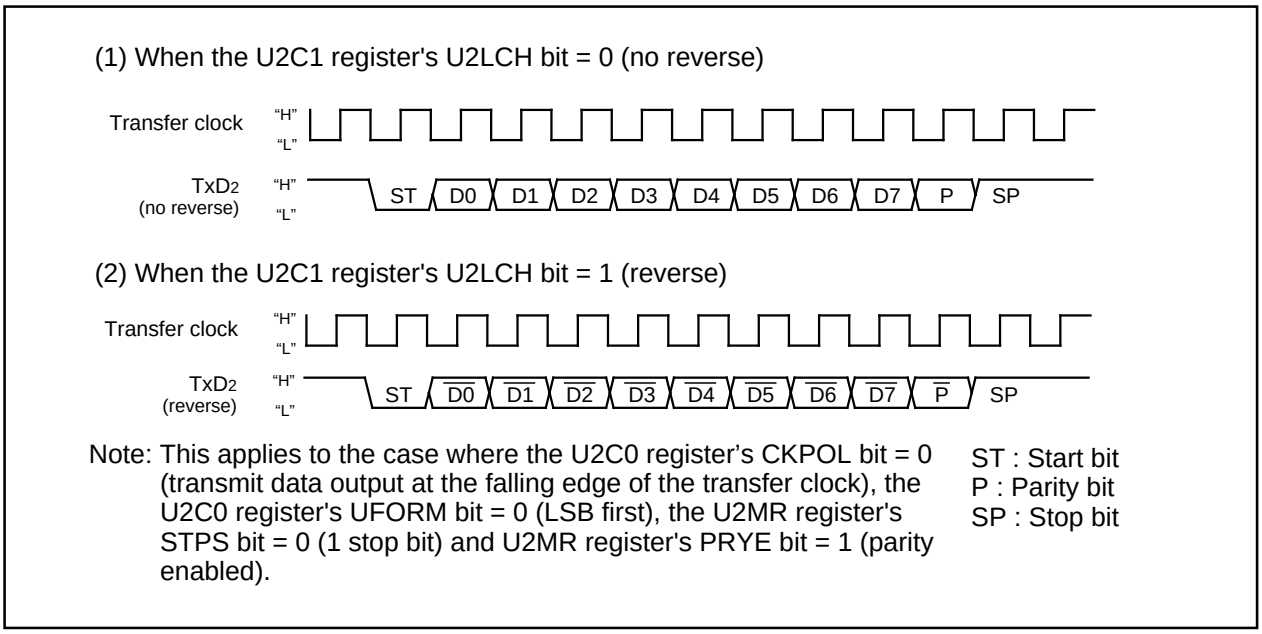



**Figure 14.1.2.2.1. Serial Data Logic Switching**

### 14.1.2.3. TxD and RxD I/O Polarity Inverse Function (UART2)

This function inverses the polarities of the TxD2 pin output and RxD2 pin input. The logic levels of all input/output data (including the start, stop and parity bits) are inversed. Figure 14.1.2.3.1 shows the TxD pin output and RxD pin input polarity inverse.

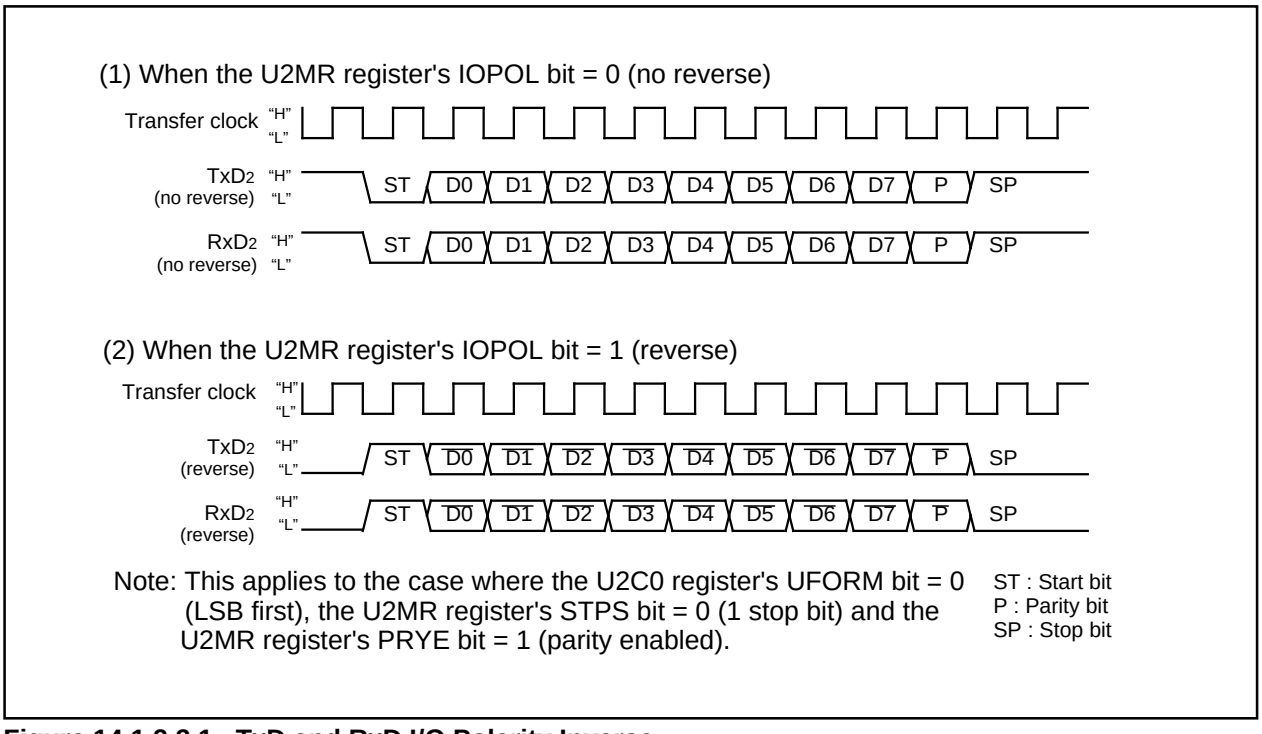



**Figure 14.1.2.3.1. TxD and RxD I/O Polarity Inverse**

#### 14.1.2.4. $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function (UART0)

This function separates $\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$, outputs $\overline{\text{RTS}}_0$ from the $P6_0$ pin, and accepts as input the $\overline{\text{CTS}}_0$ from the $P6_4$ pin. To use this function, set the register bits as shown below.

- U0C0 register's CRD bit = 0 (enables UART0 $\overline{\text{CTS}}/\overline{\text{RTS}}$)
- U0C0 register's CRS bit = 1 (outputs UART0 $\overline{\text{RTS}}$)
- U1C0 register's CRD bit = 0 (enables UART1 $\overline{\text{CTS}}/\overline{\text{RTS}}$)
- U1C0 register's CRS bit = 0 (inputs UART1 $\overline{\text{CTS}}$)
- UCON register's RCSP bit = 1 (inputs $\overline{\text{CTS}}_0$ from the $P6_4$ pin)
- UCON register's CLKMD1 bit = 0 ($CLKS_1$ not used)

Note that when using the $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function, UART1 $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function cannot be used.

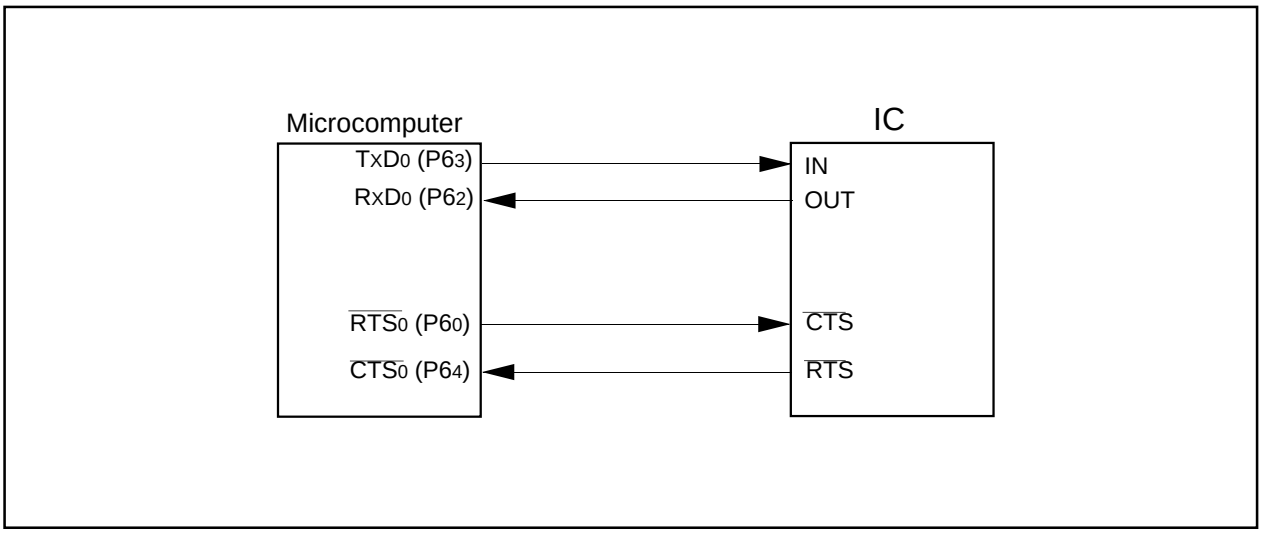



**Figure 1.19.6. $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function**

## 14.1.3 Special Mode 1 (I²C Bus mode)(UART2)

II²C mode is provided for use as a simplified I²C interface compatible mode. Table 14.1.3.1 lists the specifications of the I²C mode. Table 14.1.3.2 and 14.1.3.3 list the registers used in the I²C mode and the register values set. Table 14.1.3.4 lists the I²C mode fuctions. Figure 14.1.3.1 shows the block diagram for I²C mode. Figure 14.1.3.2 shows $SCL_2$ timing.

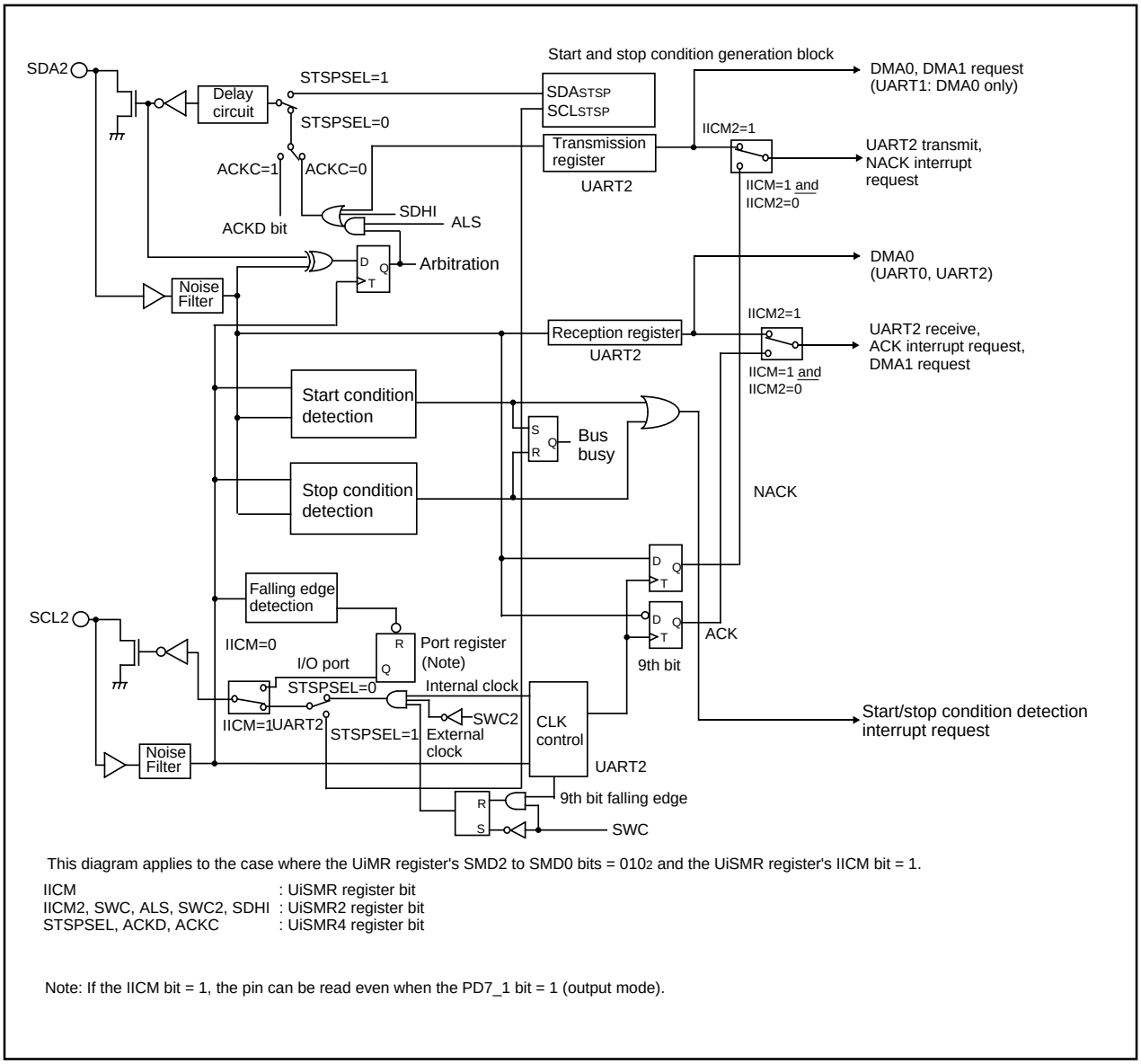
As shown in Table 14.1.3.2, the microcomputer is placed in I²C mode by setting the SMD2 to SMD0 bits to '$010_2$' and the IICM bit to "1". Because $SDA_2$ transmit output has a delay circuit attached, SDA output does not change state until $SCL_2$ goes low and remains stably low.

**Table 14.1.3.1. I²C Bus Mode Specifications**

| Item | Specification |
|---|---|
| Transfer data format | • Transfer data length: 8 bits |
| Transfer clock | • During master<br>U2MR register's CKDIR bit = "0" (internal clock) : fj/ 2(n+1)<br>fj = $f_{1SIO}$, $f_{2SIO}$, $f_{8SIO}$, $f_{32SIO}$. n: Setting value of U2BRG register $00_{16}$ to $FF_{16}$<br>• During slave<br>CKDIR bit = "1" (external clock) : Input from SCL pin |
| Transmission start condition | • Before transmission can start, the following requirements must be met (Note 1)<br>– The TE bit of U2C1 register= 1 (transmission enabled)<br>– The TI bit of U2C1 register = 0 (data present in U2TB register) |
| Reception start condition | • Before reception can start, the following requirements must be met (Note 1)<br>– The RE bit of U2C1 register= 1 (reception enabled)<br>– The TE bit of U2C1 register= 1 (transmission enabled)<br>– The TI bit of U2C1 register= 0 (data present in the UiTB register) |
| Interrupt request generation timing | When start or stop condition is detected, acknowledge undetected, and acknowledge detected |
| Error detection | • Overrun error (Note 2)<br>This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the 8th bit of the next data |
| Select function | • Arbitration lost<br>Timing at which the U2RB register's ABT bit is updated can be selected<br>• SDA digital delay<br>No digital delay or a delay of 2 to 8 U2BRG count source clock cycles selectable<br>• Clock phase setting<br>With or without clock delay selectable |

Note 1: When an external clock is selected, the conditions must be met while the external clock is in the high state.

Note 2: If an overrun error occurs, the value of U2RB register will be indeterminate. The IR bit of S2RIC register does not change.




This diagram applies to the case where the UiMR register's SMD2 to SMD0 bits = 010₂ and the UiSMR register's IICM bit = 1.

IICM : UiSMR register bit
IICM2, SWC, ALS, SWC2, SDHI : UiSMR2 register bit
STSPSEL, ACKD, ACKC : UiSMR4 register bit

Note: If the IICM bit = 1, the pin can be read even when the PD7_1 bit = 1 (output mode).

**Figure 14.1.3.1. I²C Bus Mode Block Diagram**

**Table 14.1.3.2. Registers to Be Used and Settings in I²C Bus Mode (1) (Continued)**

| Register | Bit | Function | |
|---|---|---|---|
| | | Master | Slave |
| U2TB (Note 1) | 0 to 7 | Set transmission data | Set transmission data |
| U2RB (Note 1) | 0 to 7 | Reception data can be read | Reception data can be read |
| | 8 | ACK or NACK is set in this bit | ACK or NACK is set in this bit |
| | ABT | Arbitration lost detection flag | Invalid |
| | OER | Overrun error flag | Overrun error flag |
| U2BRG | 0 to 7 | Set a transfer rate | Invalid |
| U2MR (Note 1) | SMD2 to SMD0 | Set to '010$_2$' | Set to '010$_2$' |
| | CKDIR | Set to "0" | Set to "1" |
| | IOPOL | Set to "0" | Set to "0" |
| U2C0 | CLK1, CLK0 | Select the count source for the U2BRG register | Invalid |
| | CRS | Invalid because CRD = 1 | Invalid because CRD = 1 |
| | TXEPT | Transmit buffer empty flag | Transmit buffer empty flag |
| | CRD | Set to "1" | Set to "1" |
| | NCH | Set to "1" | Set to "1" |
| | CKPOL | Set to "0" | Set to "0" |
| | UFORM | Set to "1" | Set to "1" |
| U2C1 | TE | Set this bit to "1" to enable transmission | Set this bit to "1" to enable transmission |
| | TI | Transmit buffer empty flag | Transmit buffer empty flag |
| | RE | Set this bit to "1" to enable reception | Set this bit to "1" to enable reception |
| | RI | Reception complete flag | Reception complete flag |
| | U2IRS | Invalid | Invalid |
| | U2RRM, U2LCH, U2ERE | Set to "0" | Set to "0" |
| U2SMR | IICM | Set to "1" | Set to "1" |
| | ABC | Select the timing at which arbitration-lost is detected | Invalid |
| | BBS | Bus busy flag | Bus busy flag |
| | 3 to 7 | Set to "0" | Set to "0" |
| U2SMR2 | IICM2 | Refer to Table "I²C Mode Functions" | Refer to Table " I²C Mode Functions" |
| | CSC | Set this bit to "1" to enable clock synchronization | Set to "0" |
| | SWC | Set this bit to "1" to have SCL$_2$ output fixed to "L" at the falling edge of the 9th bit of clock | Set this bit to "1" to have SCL$_2$ output fixed to "L" at the falling edge of the 9$^{th}$ bit of clock |
| | ALS | Set this bit to "1" to have SDA$_2$ output stopped when arbitration-lost is detected | Set to "0" |
| | STAC | Set to "0" | Set this bit to "1" to initialize UART2 at start condition detection |
| | SWC2 | Set this bit to "1" to have SCL$_2$ output forcibly pulled low | Set this bit to "1" to have SCL$_2$ output forcibly pulled low |
| | SDHI | Set this bit to "1" to disable SDA$_2$ output | Set this bit to "1" to disable SDA$_2$ output |
| | 7 | Set to "0" | Set to "0" |
| U2SMR3 | 0, 2, 4 and NODC | Set to "0" | Set to "0" |
| | CKPH | Refer to Table "I²C Mode Functions" | Refer to Table "I²C Mode Functions" |
| | DL2 to DL0 | Set the amount of SDA$_2$ digital delay | Set the amount of SDA$_2$ digital delay |

Note 1: Not all register bits are described above. Set those bits to "0" when writing to the registers in I²C Bus mode.

**Table 14.1.3.3. Registers to Be Used and Settings in I²C Bus Mode (2) (Continued)**

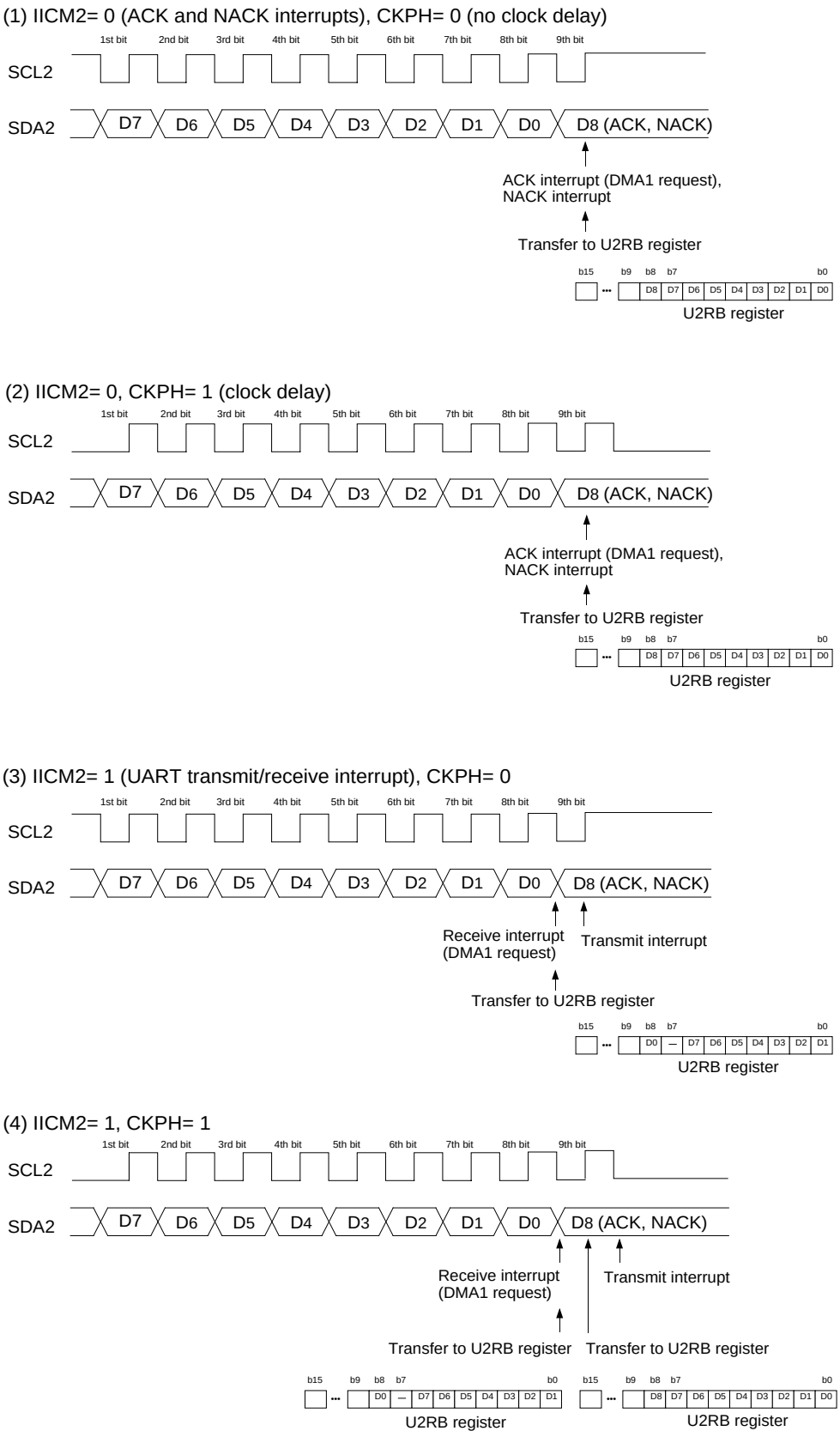
| Register | Bit | Function | |
|---|---|---|---|
| | | Master | Slave |
| U2SMR4 | STAREQ | Set this bit to “1” to generate start condition | Set to “0” |
| | RSTAREQ | Set this bit to “1” to generate restart condition | Set to “0” |
| | STPREQ | Set this bit to “1” to generate stop condition | Set to “0” |
| | STSPSEL | Set this bit to “1” to output each condition | Set to “0” |
| | ACKD | Select ACK or NACK | Select ACK or NACK |
| | ACKC | Set this bit to “1” to output ACK data | Set this bit to “1” to output ACK data |
| | SCLHI | Set this bit to “1” to have SCL2 output stopped when stop condition is detected | Set to “0” |
| | SWC9 | Set to “0” | Set this bit to “1” to set the SCL2 to “L” hold at the falling edge of the 9th bit of clock |

Note 1: Not all register bits are described above. Set those bits to “0” when writing to the registers in I²C Bus mode.

### Table 14.1.3.4. $I^2C$ Bus Mode Functions

| Function | Clock synchronous serial I/O mode (SMD2 to SMD0 = $001_2$, IICM = 0) | $I^2C$ mode (SMD2 to SMD0 = $010_2$, IICM = 1) | | | |
|---|---|---|---|---|---|
| | | IICM2 = 0 (NACK/ACK interrupt) | | IICM2 = 1 (UART transmit/ receive interrupt) | |
| | | CKPH = 0 (No clock delay) | CKPH = 1 (Clock delay) | CKPH = 0 (No clock delay) | CKPH = 1 (Clock delay) |
| Factor of interrupt number 10 (Note 1) (Refer to Fig. 14.1.3.2.) | ——— | Start condition detection or stop condition detection (Refer to Fig 14.1.3.4.) | | | |
| Factor of interrupt number 15 (Note 1) (Refer to Fig. 14.1.3.2.) | UART2 transmission Transmission started or completed (selected by U2IRS) | No acknowledgment detection (NACK) Rising edge of $SCL_2$ 9th bit | | UART2 transmission Rising edge of $SCL_2$ 9th bit | UART2 transmission Falling edge of $SCL_2$ next to the 9th bit |
| Factor of interrupt number 16 (Note 1) (Refer to Fig. 14.1.3.2.) | UART2 reception When 8th bit received CKPOL = 0 (rising edge) CKPOL = 1 (falling edge) | Acknowledgment detection (ACK) Rising edge of $SCL_2$ 9th bit | | UART2 transmission Falling edge of $SCL_2$ 9th bit | |
| Timing for transferring data from the UART reception shift register to the U2RB register | CKPOL = 0 (rising edge) CKPOL = 1 (falling edge) | Rising edge of $SCL_2$ 9th bit | | Falling edge of $SCL_2$ 9th bit | Falling and rising edges of $SCL_2$ 9th bit |
| UART2 transmission output delay | Not delayed | Delayed | | | |
| Functions of $P7_0$ pin | TxD2 output | $SDA_2$ input/output | | | |
| Functions of $P7_1$ pin | RxD2 input | $SCL_2$ input/output | | | |
| Functions of $P7_2$ pin | CLK2 input or output selected | ——— (Cannot be used in $I^2C$ mode) | | | |
| Noise filter width | 15ns | 200ns | | | |
| Read RxD2 and $SCL_2$ pin levels | Possible when the corresponding port direction bit = 0 | Always possible no matter how the corresponding port direction bit is set | | | |
| Initial value of TxD2 and $SDA_2$ outputs | CKPOL = 0 (H) CKPOL = 1 (L) | The value set in the port register before setting $I^2C$ mode (Note 2) | | | |
| Initial and end values of $SCL_2$ | ——— | H | L | H | L |
| DMA1 factor (Refer to Fig. 14.1.3.2.) | UART2 reception | Acknowledgment detection (ACK) | | UART2 reception Falling edge of $SCL_2$ 9th bit | |
| Store received data | 1st to 8th bits are stored in U2RB register bit 0 to bit 7 | 1st to 8th bits are stored in U2RB register bit 7 to bit 0 | | 1st to 7th bits are stored in U2RB register bit 6 to bit 0, with 8th bit stored in U2RB register bit 8 | 1st to 8th bits are stored in U2RB register bit 7 to bit 0 (Note 3) |
| Read received data | U2RB register status is read directly as is | | | | Read U2RB register Bit 6 to bit 0 as bit 7 to bit 1, and bit 8 as bit 0 (Note 4) |

Note 1: If the source or cause of any interrupt is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to 1 (interrupt requested). (Refer to Notes on interrupts in Precautions.)
If one of the bits shown below is changed, the interrupt source, the interrupt timing, etc. change. Therefore, always be sure to clear the IR bit to 0 (interrupt not requested) after changing those bits.
SMD2—SMD0 bits in the U2MR register, IICM bit in the U2SMR register,
IICM2 bit in the U2SMR2 register, CKPH bit in the U2SMR3 register
Note 2: Set the initial value of $SDA_2$ output while the U2MR register s SMD2 to SMD0 bits = $000_2$ (serial I/O disabled).
Note 3: Second data transfer to U2RB register (Rising edge of $SCL_2$ 9th bit)
Note 4. First data transfer to U2RB register (Falling edge of $SCL_2$ 9th bit)




This diagram applies to the case where the following condition is met.
• U2MR register CKDIR bit = 0 (Slave selected)

**Figure 14.1.3.2. Transfer to U2RB Register and Interrupt Timing**

### 14.1.3.1 Detection of Start and Stop Condition

Whether a start or a stop condition has been detected is determined.

A start condition-detected interrupt request is generated when the $SDA_2$ pin changes state from high to low while the $SCL_2$ pin is in the high state. A stop condition-detected interrupt request is generated when the $SDA_2$ pin changes state from low to high while the $SCL_2$ pin is in the high state.

Because the start and stop condition-detected interrupts share the interrupt control register and vector, check the U2SMR register's BBS bit to determine which interrupt source is requesting the interrupt.

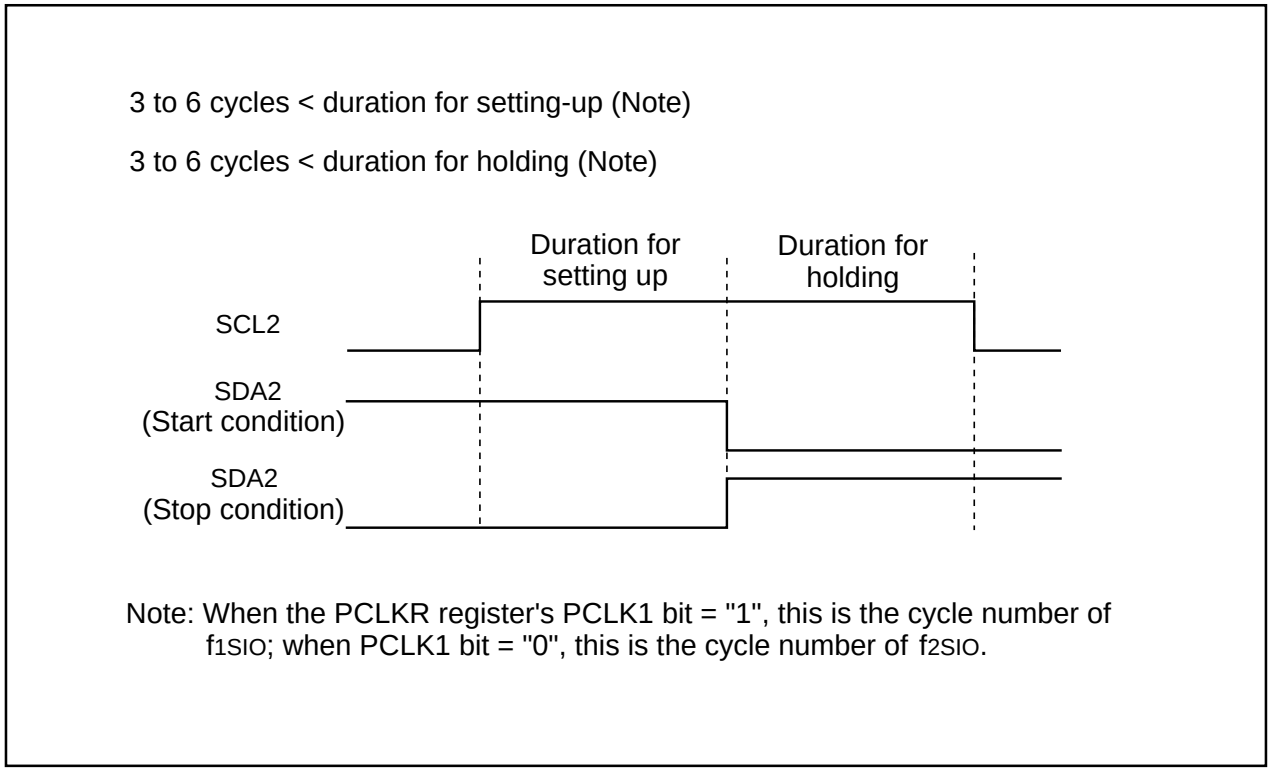



**Figure 14.1.3.1.1. Detection of Start and Stop Condition**

### 14.1.3.2 Output of Start and Stop Condition

A start condition is generated by setting the U2SMR4 register's STAREQ bit to "1" (start).

A restart condition is generated by setting the U2SMR4 register's RSTAREQ bit to "1" (start).

A stop condition is generated by setting the U2SMR4 register's STPREQ bit to "1" (start).

The output procedure is described below.

(1) Set the STAREQ bit, RSTAREQ bit or STPREQ bit to "1" (start).

(2) Set the STSPSEL bit in the U2SMR4 register to "1" (output).

Make sure that no interrupts or DMA transfers will occur between (1) and (2).

The function of the STSPSEL bit is shown in Table 14.1.3.2.1 and Figure 14.1.3.2.1.

**Table 14.1.3.2.1. STSPSEL Bit Functions**

| Function | STSPSEL = 0 | STSPSEL = 1 |
|---|---|---|
| Output of SCL2 and SDA2 pins | Output of transfer clock and data<br>Output of start/stop condition is accomplished by a program using ports (not automatically generated in hardware) | Output of a start/stop condition according to the STAREQ, RSTAREQ and STPREQ bit |
| Start/stop condition interrupt request generation timing | Start/stop condition detection | Finish generating start/stop condition |

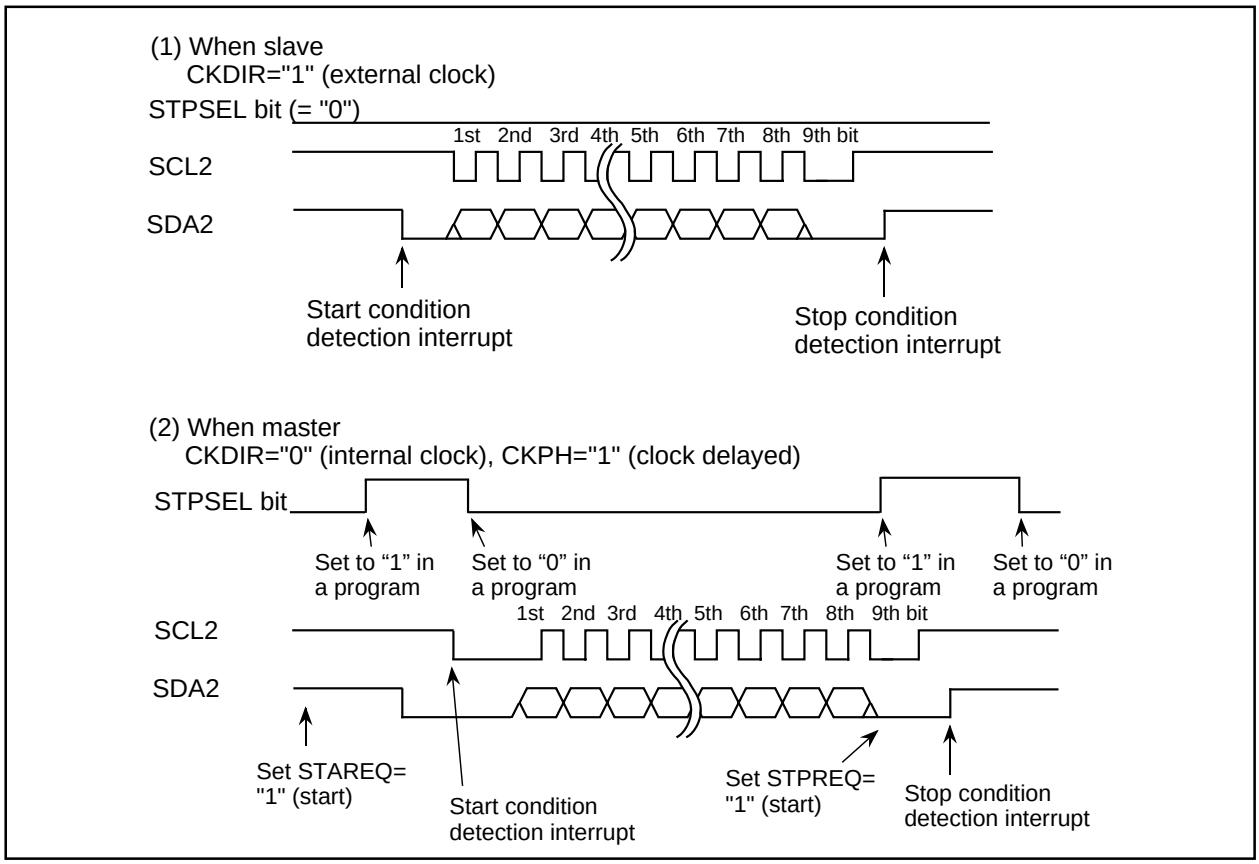



**Figure 14.1.3.2.1. STSPSEL Bit Functions**

### 14.1.3.3 Arbitration

Unmatching of the transmit data and SDA2 pin input data is checked synchronously with the rising edge of SCL2. Use the U2SMR register's ABC bit to select the timing at which the U2RB register's ABT bit is updated. If the ABC bit = 0 (updated bitwise), the ABT bit is set to "1" at the same time unmatching is detected during check, and is cleared to "0" when not detected. In cases when the ABC bit is set to "1", if unmatching is detected even once during check, the ABT bit is set to "1" (unmatching detected) at the falling edge of the clock pulse of 9th bit. If the ABT bit needs to be updated bytewise, clear the ABT bit to "0" (undetected) after detecting acknowledge in the first byte, before transferring the next byte.

Setting the U2SMR2 register's ALS bit to "1" (SDA output stop enabled) causes arbitration-lost to occur, in which case the SDA2 pin is placed in the high-impedance state at the same time the ABT bit is set to "1" (unmatching detected).

#### 14.1.3.4 Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in Figure 14.1.3.2.1.

The U2SMR2 register's CSC bit is used to synchronize the internally generated clock (internal SCL2) and an external clock supplied to the $SCL_2$ pin. In cases when the CSC bit is set to "1" (clock synchronization enabled), if a falling edge on the $SCL_2$ pin is detected while the internal $SCL_2$ is high, the internal $SCL_2$ goes low, at which time the U2BRG register value is reloaded with and starts counting in the low-level interval. If the internal $SCL_2$ changes state from low to high while the $SCL_2$ pin is low, counting stops, and when the $SCL_2$ pin goes high, counting restarts.

In this way, the UART2 transfer clock is comprised of the logical product of the internal $SCL_2$ and $SCL_2$ pin signal. The transfer clock works from a half period before the falling edge of the internal $SCL_2$ 1st bit to the rising edge of the $9^{th}$ bit. To use this function, select an internal clock for the transfer clock.

The U2SMR2 register's SWC bit allows to select whether the $SCL_2$ pin should be fixed to or freed from low-level output at the falling edge of the 9th clock pulse.

If the U2SMR4 register's SCLHI bit is set to "1" (enabled), $SCL_2$ output is turned off (placed in the high-impedance state) when a stop condition is detected.

Setting the U2SMR2 register's SWC2 bit = 1 (0 output) makes it possible to forcibly output a low-level signal from the $SCL_2$ pin even while sending or receiving data. Clearing the SWC2 bit to "0" (transfer clock) allows the transfer clock to be output from or supplied to the $SCL_2$ pin, instead of outputting a low-level signal.

If the U2SMR4 register's SWC9 bit is set to "1" (SCL hold low enabled) when the U2SMR3 register's CKPH bit = 1, the $SCL_2$ pin is fixed to low-level output at the falling edge of the clock pulse next to the ninth. Setting the SWC9 bit = 0 (SCL hold low disabled) frees the $SCL_2$ pin from low-level output.

#### 14.1.3.5 SDA Output

The data written to the U2TB register bit 7 to bit 0 ($D_7$ to $D_0$) is sequentially output beginning with $D_7$. The ninth bit ($D_8$) is ACK or NACK.

The initial value of $SDA_2$ transmit output can only be set when IICM = 1 ($I^2C$ Bus mode) and the U2MR register's SMD2 to SMD0 bits = '$000_2$' (serial I/O disabled).

The U2SMR3 register's DL2 to DL0 bits allow to add no delays or a delay of 2 to 8 U2BRG count source clock cycles to $SDA_2$ output.

Setting the U2SMR2 register's SDHI bit = 1 (SDA output disabled) forcibly places the $SDA_2$ pin in the high-impedance state. Do not write to the SDHI bit synchronously with the rising edge of the UART2 transfer clock. This is because the ABT bit may inadvertently be set to "1" (detected).

#### 14.1.3.6 SDA Input

When the IICM2 bit = 0, the 1st to 8th bits ($D_7$ to $D_0$) of received data are stored in the U2RB register bit 7 to bit 0. The 9th bit ($D_8$) is ACK or NACK.

When the IICM2 bit = 1, the 1st to 7th bits ($D_7$ to $D_1$) of received data are stored in the U2RB register bit 6 to bit 0 and the 8th bit ($D_0$) is stored in the U2RB register bit 8. Even when the IICM2 bit = 1, providing the CKPH bit = 1, the same data as when the IICM2 bit = 0 can be read out by reading the U2RB register after the rising edge of the corresponding clock pulse of 9th bit.

#### 14.1.3.7 ACK and NACK

If the STSPSEL bit in the U2SMR4 register is set to “0” (start and stop conditions not generated) and the ACKC bit in the U2SMR4 register is set to “1” (ACK data output), the value of the ACKD bit in the U2SMR4 register is output from the $SDA_2$ pin.

If the IICM2 bit = 0, a NACK interrupt request is generated if the $SDA_2$ pin remains high at the rising edge of the 9th bit of transmit clock pulse. An ACK interrupt request is generated if the $SDA_2$ pin is low at the rising edge of the 9th bit of transmit clock pulse.

If ACK2 is selected for the cause of DMA1 request, a DMA transfer can be activated by detection of an acknowledge.

#### 14.1.3.8 Initialization of Transmission/Reception

If a start condition is detected while the STAC bit = 1 (UART2 initialization enabled), the serial I/O operates as described below.

- The transmit shift register is initialized, and the content of the U2TB register is transferred to the transmit shift register. In this way, the serial I/O starts sending data synchronously with the next clock pulse applied. However, the UART2 output value does not change state and remains the same as when a start condition was detected until the first bit of data is output synchronously with the input clock.
- The receive shift register is initialized, and the serial I/O starts receiving data synchronously with the next clock pulse applied.
- The SWC bit is set to “1” (SCL wait output enabled). Consequently, the $SCL_2$ pin is pulled low at the falling edge of the ninth clock pulse.

Note that when UART2 transmission/reception is started using this function, the TI does not change state. Note also that when using this function, the selected transfer clock should be an external clock.

## 14.1.4 Special Mode 2 (UART2)

Multiple slaves can be serially communicated from one master. Transfer clock polarity and phase are selectable. Table 14.1.4.1 lists the specifications of Special Mode 2. Table 14.1.4.2 lists the registers used in Special Mode 2 and the register values set. Figure 14.1.4.1 shows communication control example for Special Mode 2.

**Table 14.1.4.1. Special Mode 2 Specifications**

| Item | Specification |
|---|---|
| Transfer data format | • Transfer data length: 8 bits |
| Transfer clock | • Master mode<br>U2MR register's CKDIR bit = "0" (internal clock) : fj/ 2(n+1)<br>fj = $f_{1SIO}$, $f_{2SIO}$, $f_{8SIO}$, $f_{32SIO}$. n: Setting value of U2BRG register $00_{16}$ to $FF_{16}$<br>• Slave mode<br>CKDIR bit = "1" (external clock selected) : Input from CLK2 pin |
| Transmit/receive control | Controlled by input/output ports |
| Transmission start condition | • Before transmission can start, the following requirements must be met (Note 1)<br>– The TE bit of U2C1 register= 1 (transmission enabled)<br>– The TI bit of U2C1 register = 0 (data present in U2TB register) |
| Reception start condition | • Before reception can start, the following requirements must be met (Note 1)<br>– The RE bit of U2C1 register= 1 (reception enabled)<br>– The TE bit of U2C1 register= 1 (transmission enabled)<br>– The TI bit of U2C1 register= 0 (data present in the U2TB register) |
| Interrupt request generation timing | • For transmission, one of the following conditions can be selected<br>– The U2IRS bit of U2C1 register = 0 (transmit buffer empty): when transferring data from the U2TB register to the UART2 transmit register (at start of transmission)<br>– The U2IRS bit =1 (transfer completed): when the serial I/O finished sending data from the UART2 transmit register<br>• For reception<br>When transferring data from the UART2 receive register to the U2RB register (at completion of reception) |
| Error detection | • Overrun error (Note 2)<br>This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the 7th bit of the next data |
| Select function | • Clock phase setting<br>Selectable from four combinations of transfer clock polarities and phases |

Note 1: When an external clock is selected, the conditions must be met while if the U2C0 register's CKPOL bit = "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the U2C0 register's CKPOL bit = "1" (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.

Note 2: If an overrun error occurs, the value of U2RB register will be indeterminate. The IR bit of S2RIC register does not change.

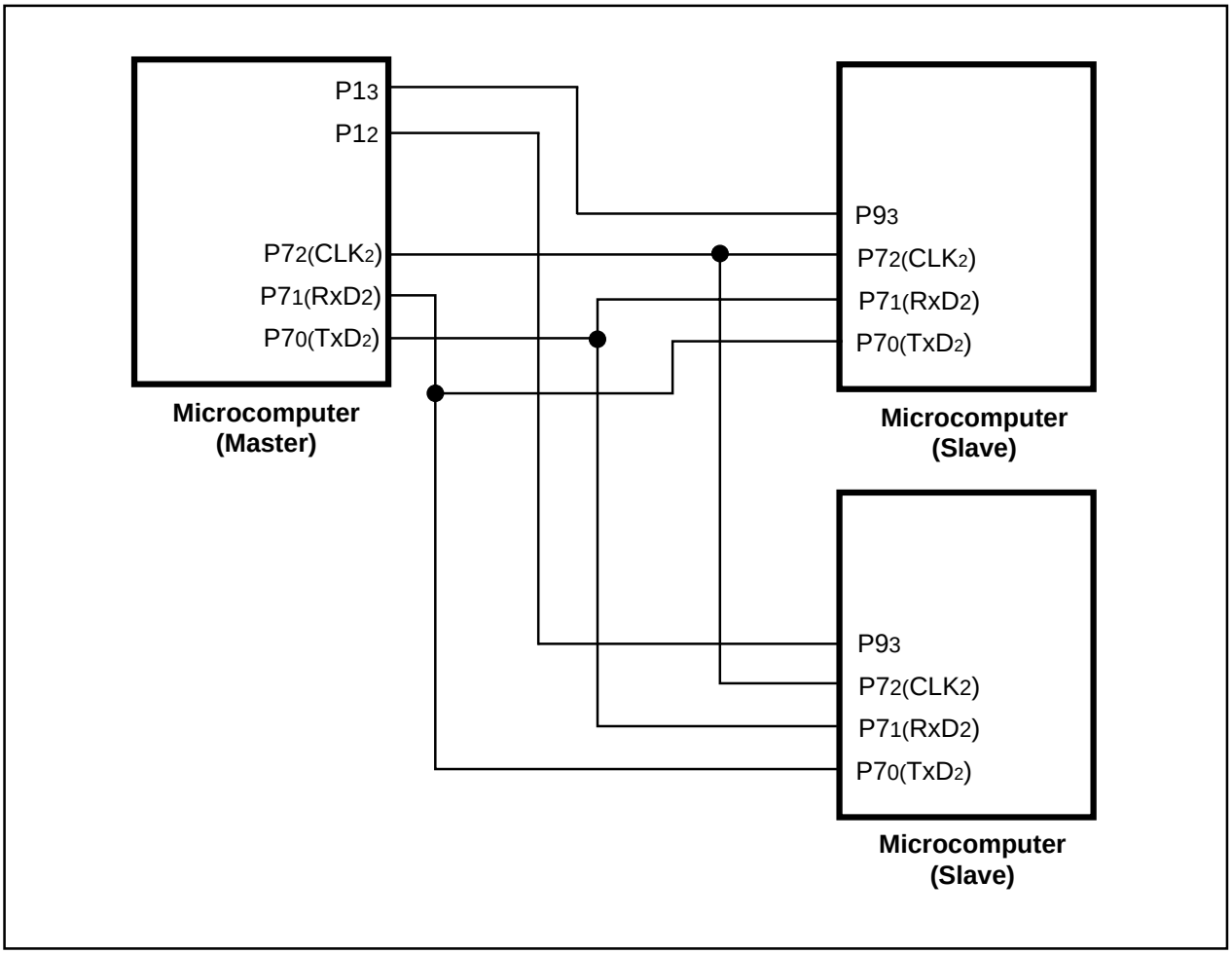



**Figure 14.1.4.1. Serial Bus Communication Control Example (UART2)**

**Table 14.1.4.2. Registers to Be Used and Settings in Special Mode 2**

| Register | Bit | Function |
|---|---|---|
| U2TB(Note) | 0 to 7 | Set transmission data |
| U2RB(Note) | 0 to 7 | Reception data can be read |
| | OER | Overrun error flag |
| U2BRG | 0 to 7 | Set a transfer rate |
| U2MR(Note) | SMD2 to SMD0 | Set to '001$_2$' |
| | CKDIR | Set this bit to "0" for master mode or "1" for slave mode |
| | IOPOL | Set to "0" |
| U2C0 | CLK1, CLK0 | Select the count source for the U2BRG register |
| | CRS | Invalid because CRD = 1 |
| | TXEPT | Transmit register empty flag |
| | CRD | Set to "1" |
| | NCH | Select TxD2 pin output format |
| | CKPOL | Clock phases can be set in combination with the U2SMR3 register's CKPH bit |
| | UFORM | Select the LSB first or MSB first |
| U2C1 | TE | Set this bit to "1" to enable transmission |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to "1" to enable reception |
| | RI | Reception complete flag |
| | U2IRS | Select UART2 transmit interrupt cause |
| | U2RRM, U2LCH, U2ERE | Set to "0" |
| U2SMR | 0 to 7 | Set to "0" |
| U2SMR2 | 0 to 7 | Set to "0" |
| U2SMR3 | CKPH | Clock phases can be set in combination with the U2C0 register's CKPOL bit |
| | NODC | Set to "0" |
| | 0, 2, 4 to 7 | Set to "0" |
| U2SMR4 | 0 to 7 | Set to "0" |

Note : Not all register bits are described above. Set those bits to "0" when writing to the registers in Special Mode 2.

#### 14.1.4.1 Clock Phase Setting Function

One of four combinations of transfer clock phases and polarities can be selected using the U2SMR3 register's CKPH bit and the U2C0 register's CKPOL bit.

Make sure the transfer clock polarity and phase are the same for the master and slave to communicate.

##### 14.1.4.1.1 Master (Internal Clock)

Figure 14.1.4.1.1.1 shows the transmission and reception timing in master (internal clock).

##### 14.1.4.1.2 Slave (External Clock)

Figure 14.1.4.1.2.1 shows the transmission and reception timing (CKPH=0) in slave (external clock) while Figure 14.1.4.1.2.2 shows the transmission and reception timing (CKPH=1) in slave (external clock).

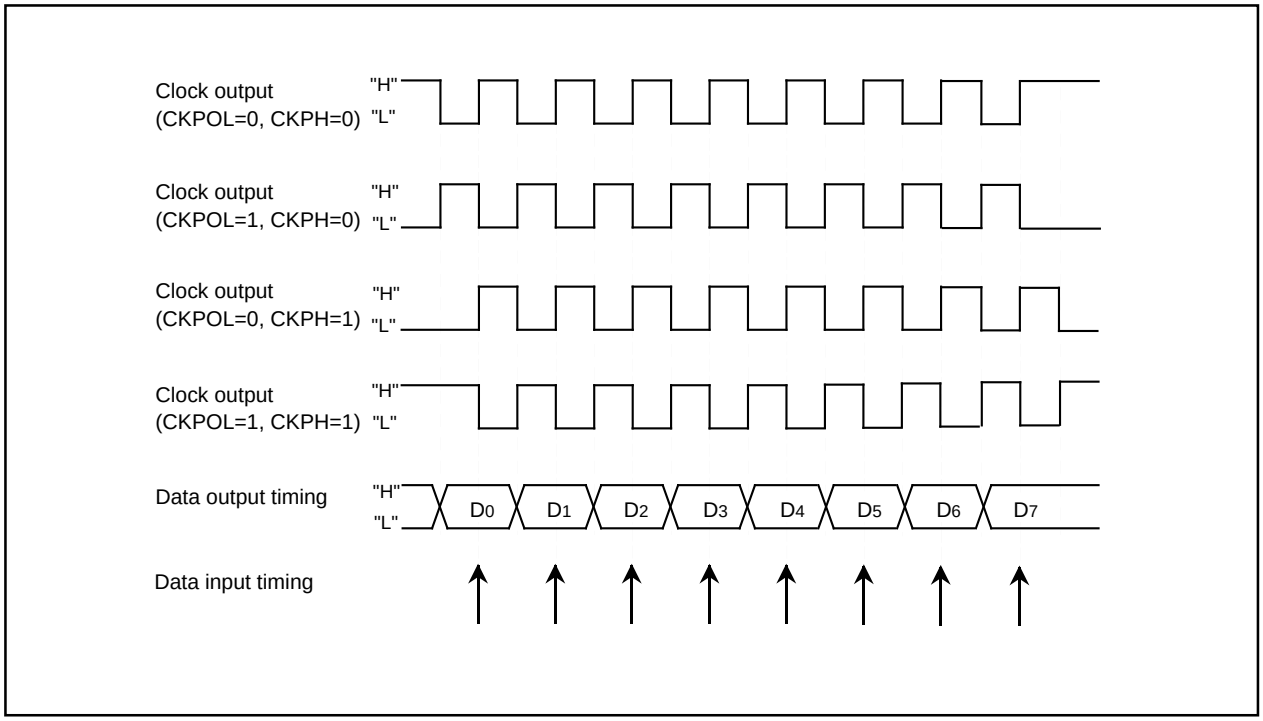



**Figure 14.1.4.1.1.1. Transmission and Reception Timing in Master Mode (Internal Clock)**

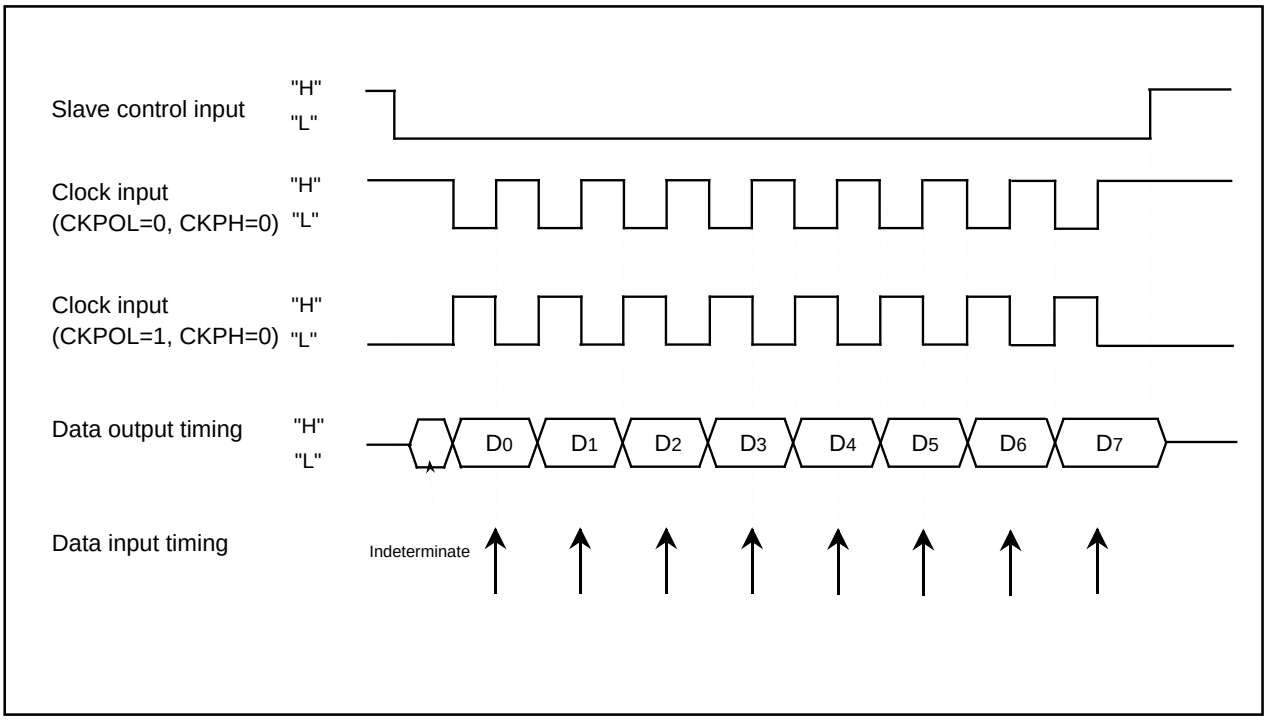



**Figure 14.1.4.1.2.1. Transmission and Reception Timing (CKPH=0) in Slave Mode (External Clock)**

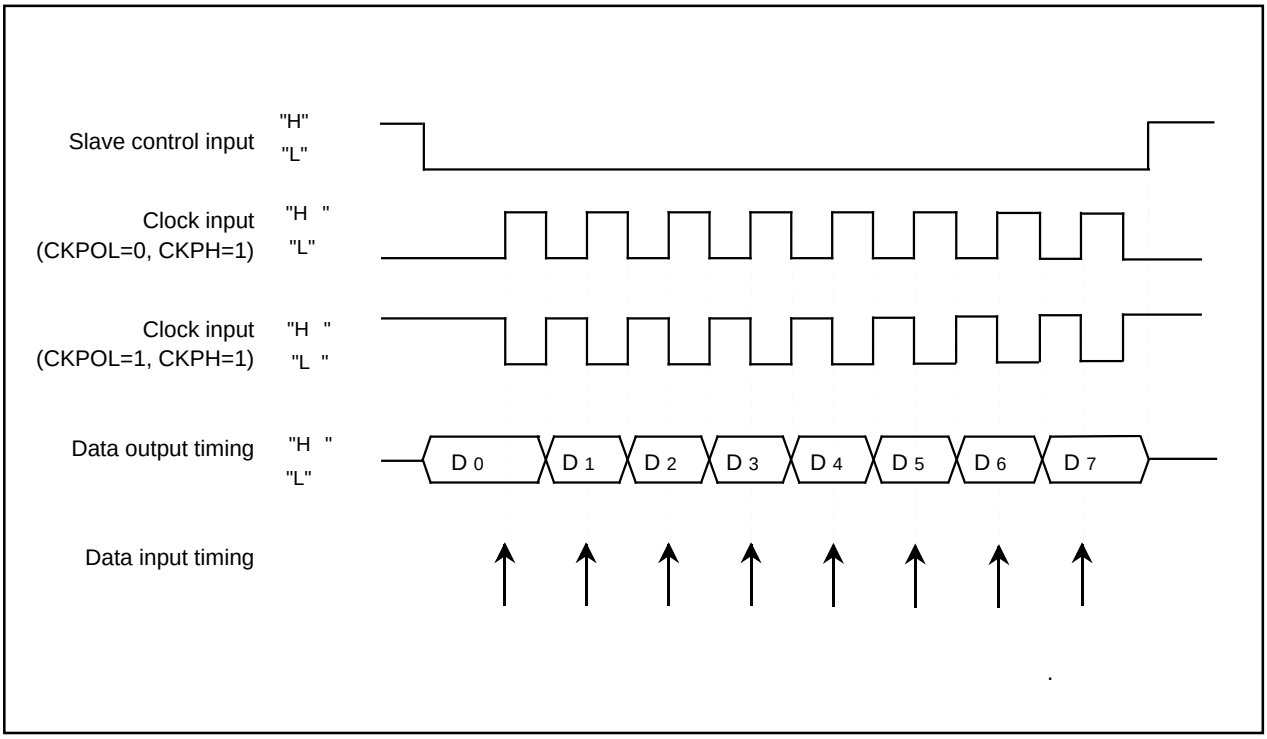



**Figure 14.1.4.1.2.2. Transmission and Reception Timing (CKPH=1) in Slave Mode (External Clock)**

## 14.1.5 Special Mode 3 (IE Bus mode)(UART2)

In this mode, one bit of IE Bus is approximated with one byte of UART mode waveform.

Table 14.1.5.1 lists the registers used in IE Bus mode and the register values set. Figure 14.1.5.1 shows the functions of bus collision detect function related bits.

If the TxD2 pin output level and RxD2 pin input level do not match, a UART2 bus collision detect interrupt request is generated.

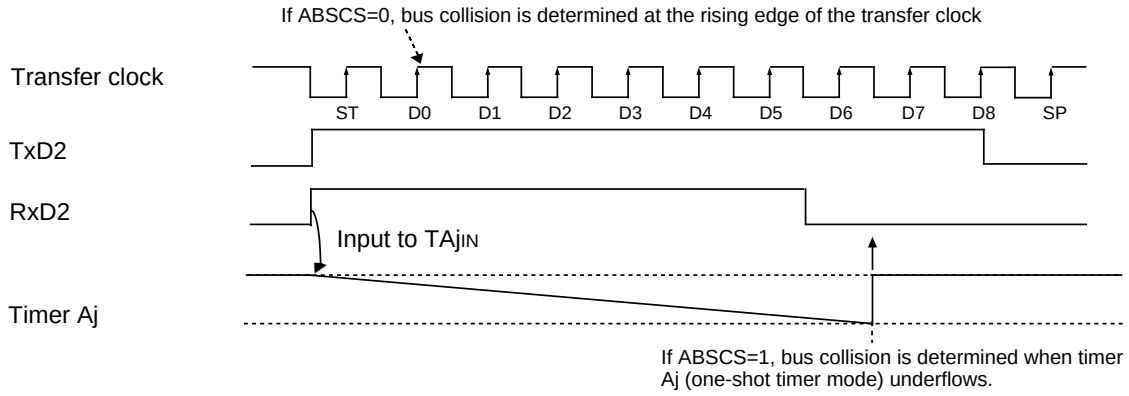
Use the IFSR2A register's IFSR26 and IFSR27 bits to enable the UART0/UART1 bus collision detect function.

**Table 14.1.5.1. Registers to Be Used and Settings in IE Bus Mode**

| Register | Bit | Function |
|---|---|---|
| U2TB | 0 to 8 | Set transmission data |
| U2RB(Note) | 0 to 8 | Reception data can be read |
| | OER,FER,PER,SUM | Error flag |
| U2BRG | 0 to 7 | Set a transfer rate |
| U2MR | SMD2 to SMD0 | Set to '110$_2$' |
| | CKDIR | Select the internal clock or external clock |
| | STPS | Set to "0" |
| | PRY | Invalid because PRYE=0 |
| | PRYE | Set to "0" |
| | IOPOL | Select the TxD/RxD input/output polarity |
| U2C0 | CLK1, CLK0 | Select the count source for the U2BRG register |
| | CRS | Invalid because CRD=1 |
| | TXEPT | Transmit register empty flag |
| | CRD | Set to "1" |
| | NCH | Select TxD2 pin output mode |
| | CKPOL | Set to "0" |
| | UFORM | Set to "0" |
| U2C1 | TE | Set this bit to "1" to enable transmission |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to "1" to enable reception |
| | RI | Reception complete flag |
| | U2IRS | Select the source of UART2 transmit interrupt |
| | U2RRM, U2LCH, U2ERE | Set to "0" |
| U2SMR | 0 to 3, 7 | Set to "0" |
| | ABSCS | Select the sampling timing at which to detect a bus collision |
| | ACSE | Set this bit to "1" to use the auto clear function of transmit enable bit |
| | SSS | Select the transmit start condition |
| U2SMR2 | 0 to 7 | Set to "0" |
| U2SMR3 | 0 to 7 | Set to "0" |
| U2SMR4 | 0 to 7 | Set to "0" |

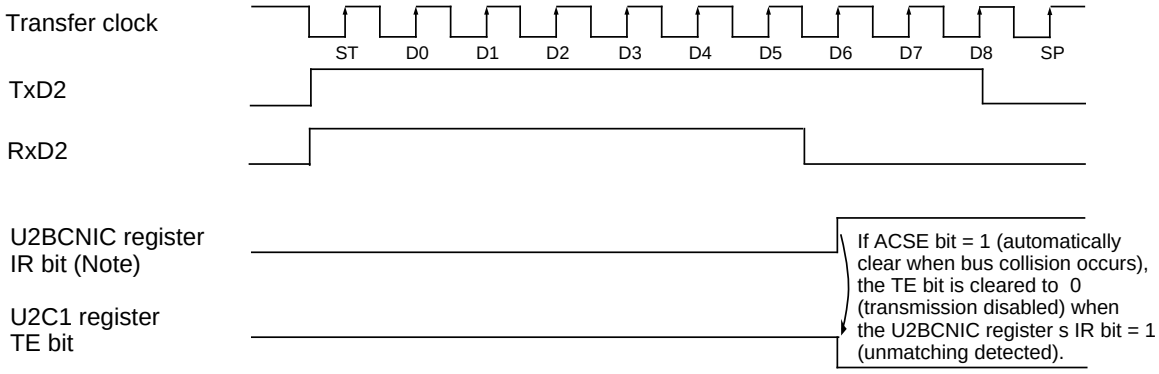
Note : Not all register bits are described above. Set those bits to "0" when writing to the registers in IE Bus mode.

**(1) U2SMR register ABSCS bit (bus collision detect sampling clock select)**




Timer Aj: timer A0 when UART2

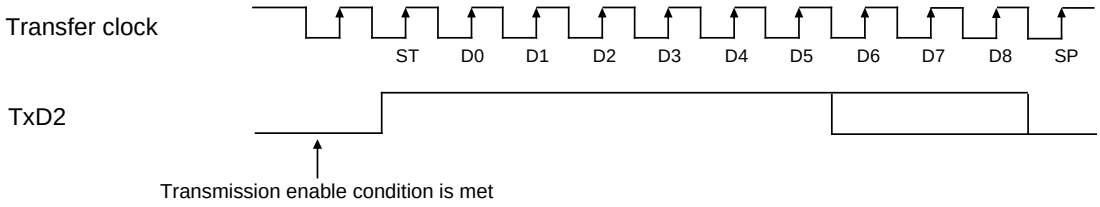
**(2) U2SMR register ACSE bit (auto clear of transmit enable bit)**



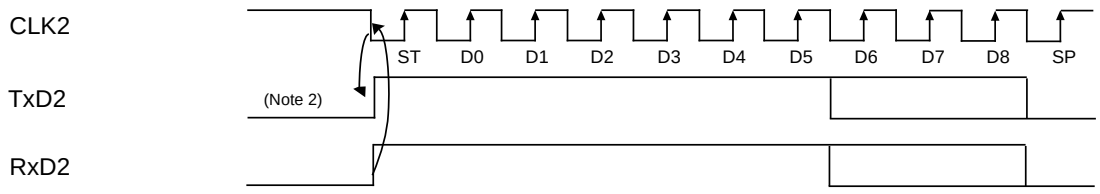

Note: BCNIC register when UART2.

**(3) U2SMR register SSS bit (Transmit start condition select)**

If SSS bit = 0, the serial I/O starts sending data one transfer clock cycle after the transmission enable condition is met.




If SSS bit = 1, the serial I/O starts sending data at the rising edge (Note 1) of RxD2




Note 1: The falling edge of RxD2 when IOPOL=0; the rising edge of RxD2 when IOPOL = 1.
Note 2: The transmit condition must be met before the falling edge (Note 1) of RxD.

This diagram applies to the case where IOPOL=1 (reversed).

**Figure 14.1.5.1. Bus Collision Detect Function-Related Bits**

## 14.1.6 Special Mode 4 (SIM Mode) (UART2)

Based on UART mode, this is an SIM interface compatible mode. Direct and inverse formats can be implemented, and this mode allows output of a low from the TxD2 pin when a parity error is detected. Tables 14.1.6.1 lists the specifications of SIM mode. Table 14.1.6.2 lists the registers used in the SIM mode and the register values set.

**Table 14.1.6.1. SIM Mode Specifications**

| Item | Specification |
|---|---|
| Transfer data format | • Direct format<br>• Inverse format |
| Transfer clock | • U2MR register's CKDIR bit = "0" (internal clock) : fi/ 16(n+1)<br>fi = $f_{1SIO}$, $f_{2SIO}$, $f_{8SIO}$, $f_{32SIO}$. n: Setting value of U2BRG register $00_{16}$ to $FF_{16}$<br>• CKDIR bit = "1" (external clock) : $f_{EXT}$/16(n+1)<br>$f_{EXT}$: Input from $CLK_2$ pin. n: Setting value of U2BRG register $00_{16}$ to $FF_{16}$ |
| Transmission start condition | • Before transmission can start, the following requirements must be met<br>– The TE bit of U2C1 register= 1 (transmission enabled)<br>– The TI bit of U2C1 register = 0 (data present in U2TB register) |
| Reception start condition | • Before reception can start, the following requirements must be met<br>– The RE bit of U2C1 register= 1 (reception enabled)<br>– Start bit detection |
| Interrupt request generation timing (Note 2) | • For transmission<br>When the serial I/O finished sending data from the U2TB transfer register (U2IRS bit =1)<br>• For reception<br>When transferring data from the UART2 receive register to the U2RB register (at completion of reception) |
| Error detection | • Overrun error (Note 1)<br>This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the bit one before the last stop bit of the next data<br>• Framing error<br>This error occurs when the number of stop bits set is not detected<br>• Parity error<br>During reception, if a parity error is detected, parity error signal is output from the $TxD_2$ pin.<br>During transmission, a parity error is detected by the level of input to the $RxD_2$ pin when a transmission interrupt occurs<br>• Error sum flag<br>This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered |

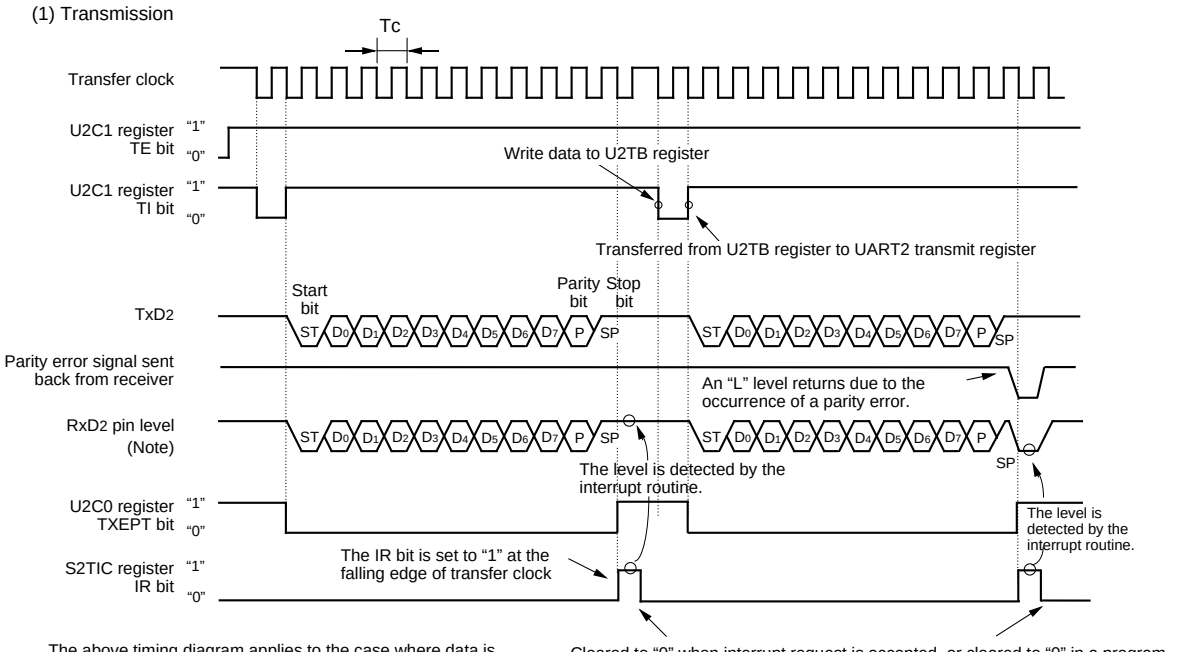
Note 1: If an overrun error occurs, the value of U2RB register will be indeterminate. The IR bit of S2RIC register does not change.

Note 2: A transmit interrupt request is generated by setting the U2C1 register U2IRS bit to "1" (transmission complete) and U2ERE bit to "1" (error signal output) after reset. Therefore, when using SIM mode, be sure to clear the IR bit to "0" (no interrupt request) after setting these bits.

**Table 14.1.6.2. Registers to Be Used and Settings in SIM Mode**

| Register | Bit | Function |
|---|---|---|
| U2TB(Note) | 0 to 7 | Set transmission data |
| U2RB(Note) | 0 to 7 | Reception data can be read |
| | OER,FER,PER,SUM | Error flag |
| U2BRG | 0 to 7 | Set a transfer rate |
| U2MR | SMD2 to SMD0 | Set to '101₂' |
| | CKDIR | Select the internal clock or external clock |
| | STPS | Set to "0" |
| | PRY | Set this bit to "1" for direct format or "0" for inverse format |
| | PRYE | Set to "1" |
| | IOPOL | Set to "0" |
| U2C0 | CLK1, CLK0 | Select the count source for the U2BRG register |
| | CRS | Invalid because CRD=1 |
| | TXEPT | Transmit register empty flag |
| | CRD | Set to "1" |
| | NCH | Set to "0" |
| | CKPOL | Set to "0" |
| | UFORM | Set this bit to "0" for direct format or "1" for inverse format |
| U2C1 | TE | Set this bit to "1" to enable transmission |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to "1" to enable reception |
| | RI | Reception complete flag |
| | U2IRS | Set to "1" |
| | U2RRM | Set to "0" |
| | U2LCH | Set this bit to "0" for direct format or "1" for inverse format |
| | U2ERE | Set to "1" |
| U2SMR(Note) | 0 to 3 | Set to "0" |
| U2SMR2 | 0 to 7 | Set to "0" |
| U2SMR3 | 0 to 7 | Set to "0" |
| U2SMR4 | 0 to 7 | Set to "0" |

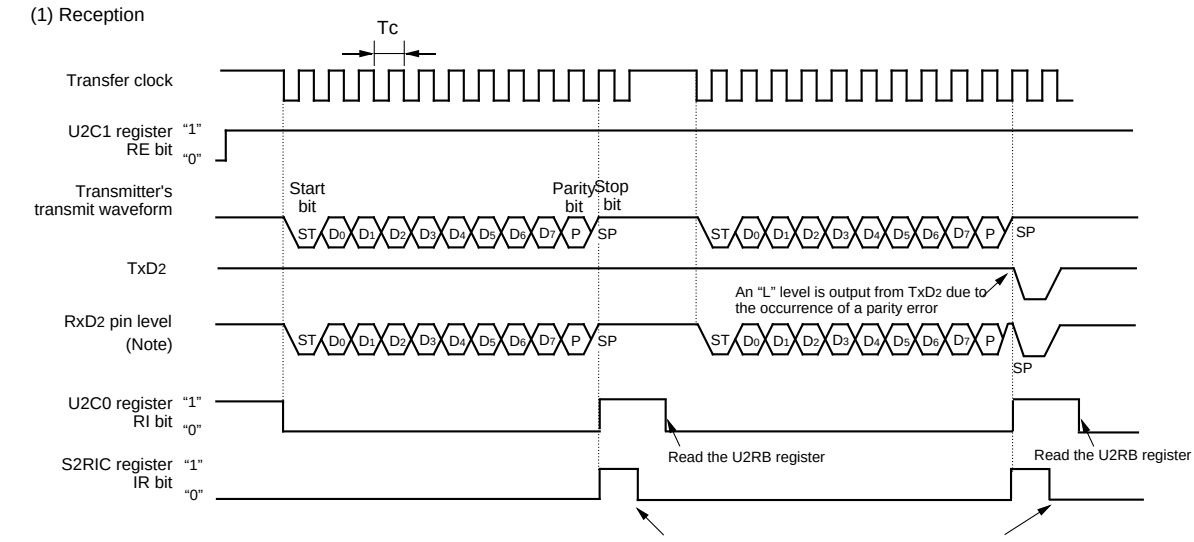
Note: Not all register bits are described above. Set those bits to "0" when writing to the registers in SIM mode.




The above timing diagram applies to the case where data is transferred in the direct format.
- U2MR register STPS bit = 0 (1 stop bit)
- U2MR register PRY bit = 1 (even)
- U2C0 register UFORM bit = 0 (LSB first)
- U2C1 register U2LCH bit = 0 (no reverse)
- U2C1 register U2IRSCH bit = 1 (transmit is completed)

Tc = 16 (n + 1) / fi or 16 (n + 1) / $f_{EXT}$
fi : frequency of U2BRG count source ($f_{1SIO}$, $f_{2SIO}$, $f_{8SIO}$, $f_{32SIO}$)
$f_{EXT}$ : frequency of U2BRG count source (external clock)
n : value set to U2BRG

Note : Because TxD2 and RxD2 are connected, this is composite waveform consisting of the TxD2 output and the parity error signal sent back from receiver.




The above timing diagram applies to the case where data is transferred in the direct format.
- U2MR register STPS bit = 0 (1 stop bit)
- U2MR register PRY bit = 1 (even)
- U2C0 register UFORM bit = 0 (LSB first)
- U2C1 register U2LCH bit = 0 (no reverse)
- U2C1 register U2IRSCH bit = 1 (transmit is completed)

Tc = 16 (n + 1) / fi or 16 (n + 1) / $f_{EXT}$
fi : frequency of U2BRG count source ($f_{1SIO}$, $f_{2SIO}$, $f_{8SIO}$, $f_{32SIO}$)
$f_{EXT}$ : frequency of U2BRG count source (external clock)
n : value set to U2BRG

Note : Because TxD2 and RxD2 are connected, this is composite waveform consisting of the transmitter's transmit waveform and the parity error signal received.

**Figure 14.1.6.1. Transmit and Receive Timing in SIM Mode**

Figure 14.1.6.2 shows the example of connecting the SIM interface. Connect TxD2 and RxD2 and apply pull-up.

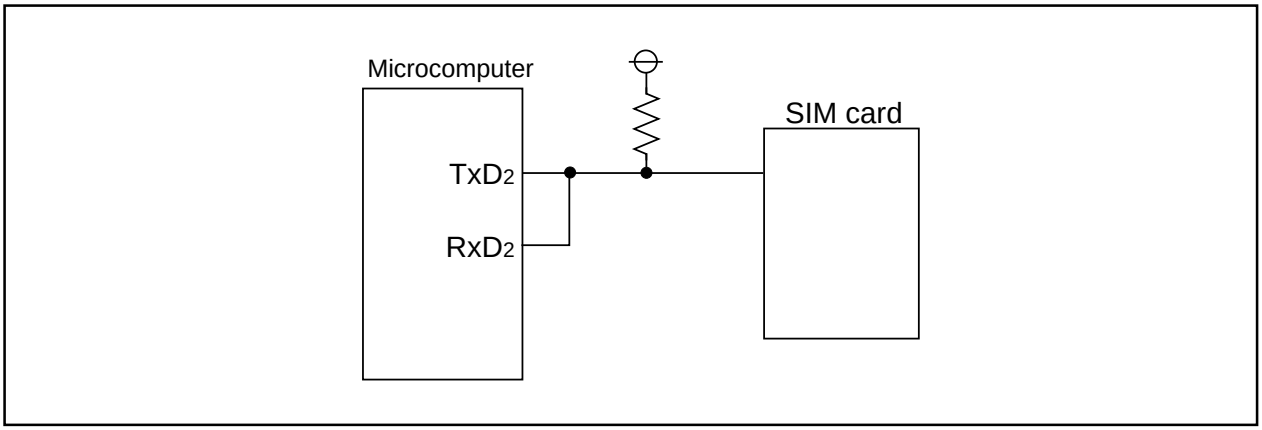



**Figure 14.1.6.2. SIM Interface Connection**

### 14.1.6.1 Parity Error Signal Output

The parity error signal is enabled by setting the U2C1 register's U2ERE bit to "1".

• When receiving

The parity error signal is output when a parity error is detected while receiving data. This is achieved by pulling the TxD2 output low with the timing shown in Figure 14.1.6.1.1. If the R2RB register is read while outputting a parity error signal, the PER bit is cleared to "0" and at the same time the TxD2 output is returned high.

• When transmitting

A transmission-finished interrupt request is generated at the falling edge of the transfer clock pulse that immediately follows the stop bit. Therefore, whether a parity signal has been returned can be determined by reading the port that shares the RxD2 pin in a transmission-finished interrupt service routine.

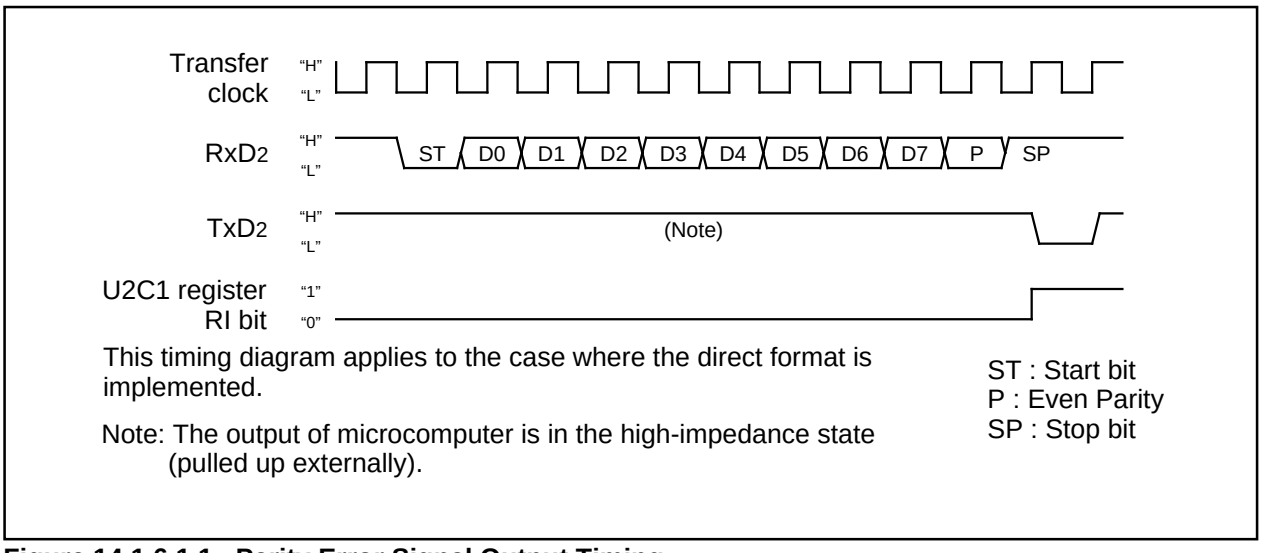



**Figure 14.1.6.1.1. Parity Error Signal Output Timing**

### 14.1.6.2 Format

• Direct Format

Set the U2MR register's PRY bit to “1”, U2C0 register's UFORM bit to “0” and U2C1 register's U2LCH bit to “0”.

• Inverse Format

Set the PRY bit to “0”, UFORM bit to “1” and U2LCH bit to “1”.

Figure 14.1.6.2.1 shows the SIM interface format.

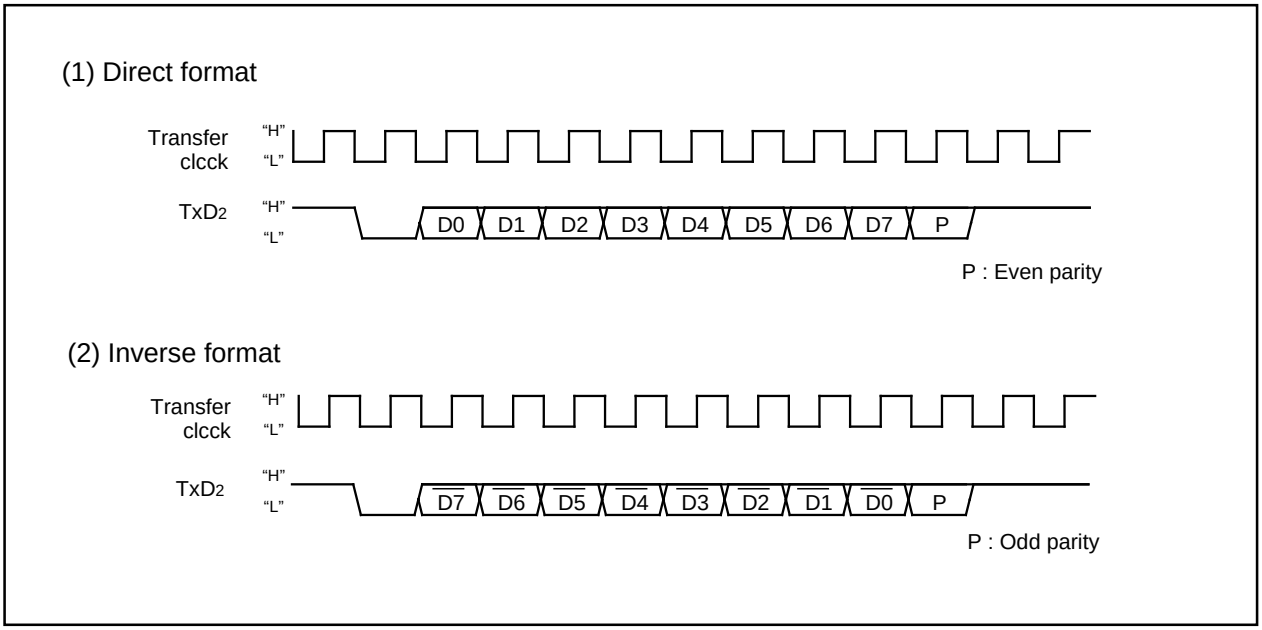



**Figure 14.1.6.2.1. SIM Interface Format**

## 14.2 SI/O3 and SI/O4

SI/O3 and SI/O4 are exclusive clock-synchronous serial I/Os.

Figure 14.2.1 shows the block diagram of SI/O3 and SI/O4, and Figure 14.2.2 shows the SI/O3 and SI/O4-related registers.

Table 14.2.1 shows the specifications of SI/O3 and SI/O4.

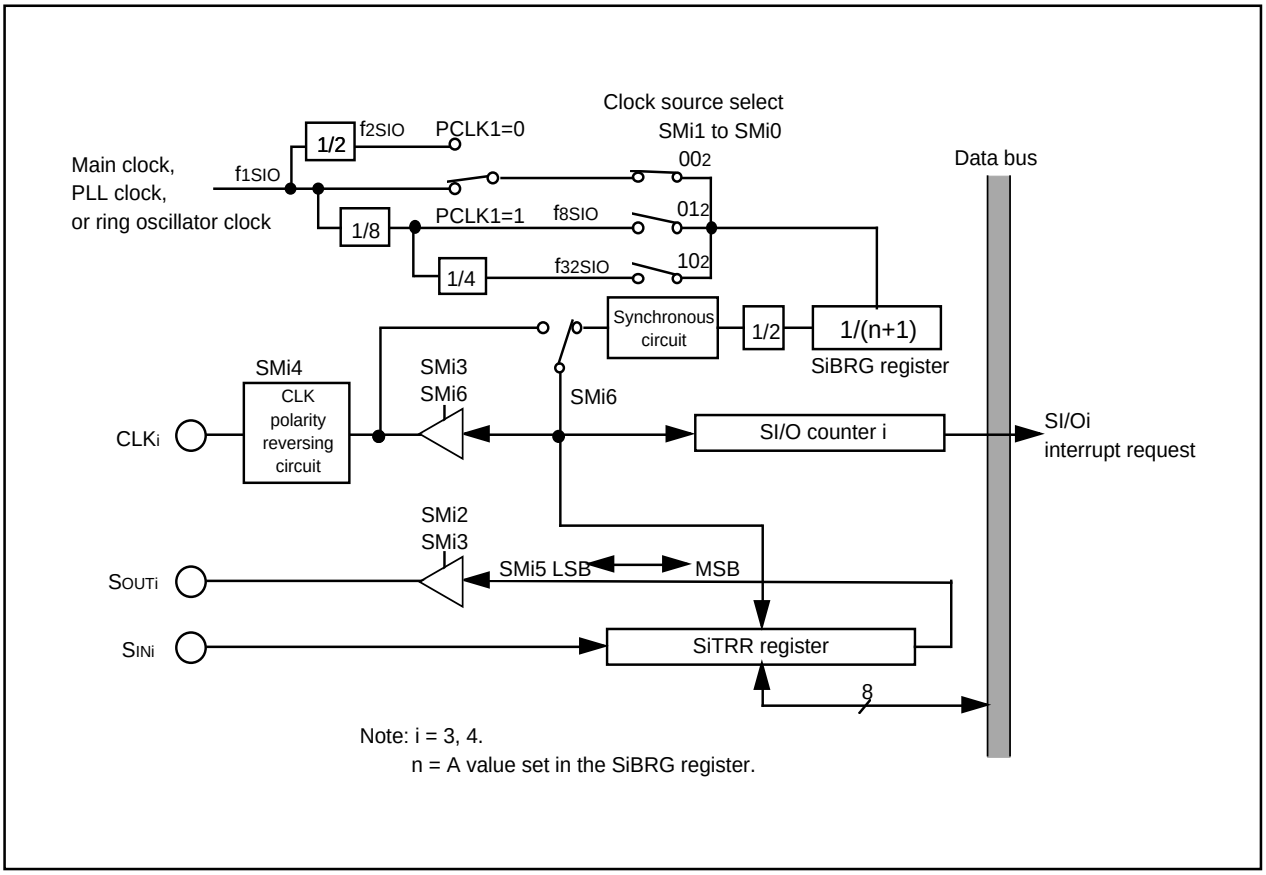



**Figure 14.2.1. SI/O3 and SI/O4 Block Diagram**

S I/Oi control register (i = 3, 4) (Note 1)

b7 b6 b5 b4 b3 b2 b1 b0

| Symbol | Address | After reset |
|---|---|---|
| S3C | 036216 | 01000002 |
| S4C | 036616 | 01000002 |

| Bit symbol | Bit name | Description | RW |
|---|---|---|---|
| SMi0 | Internal synchronous clock select bit | b1 b0<br>0 0 : Selecting f1SIO or f2SIO<br>0 1 : Selecting f8SIO<br>1 0 : Selecting f32SIO<br>1 1 : Must not be set. | RW |
| SMi1 | | | RW |
| SMi2 | SOUTi output disable bit (Note 4) | 0 : SOUTi output<br>1 : SOUTi output disable(high impedance) | RW |
| SMi3 | S I/Oi port select bit | 0 : Input/output port<br>1 : SOUTi output, CLKi function | RW |
| SMi4 | CLK polarity select bit | 0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge<br>1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge | RW |
| SMi5 | Transfer direction select bit | 0 : LSB first<br>1 : MSB first | RW |
| SMi6 | Synchronous clock select bit | 0 : External clock (Note 2)<br>1 : Internal clock (Note 3) | RW |
| SMi7 | SOUTi initial value set bit | Effective when SMi3 = 0<br>0 : "L" output<br>1 : "H" output | RW |

Note 1: Make sure register S4C is written to by the next instruction after setting the PRCR register's PRC2 bit to "1" (write enable).
Note 2: Set the SMi3 bit to "1" (SOUTi output, CLKi function).
Note 3: Set the SMi3 bit to "1" and the corresponding port direction bit to "0" (input mode).
Note 4: Effective when SMi3 bit = 1.

SI/Oi bit rate generator (i = 3, 4) (Notes 1, 2)

b7 b0

| Symbol | Address | After reset |
|---|---|---|
| S3BRG | 036316 | ??16 |
| S4BRG | 036716 | ??16 |

| Description | Setting range | RW |
|---|---|---|
| Assuming that set value = n, BRGi divides the count source by n + 1 | 0016 to FF16 | WO |

Note 1: Write to this register while serial I/O is neither transmitting nor receiving.
Note 2: Use MOV instruction to write to this register.

SI/Oi transmit/receive register (i = 3, 4) (Note 1, 2)

b7 b0

| Symbol | Address | After reset |
|---|---|---|
| S3TRR | 036016 | ??16 |
| S4TRR | 036416 | ??16 |

| Description | RW |
|---|---|
| Transmission/reception starts by writing transmit data to this register. After transmission/reception finishes, reception data can be read by reading this register. | RW |

Note 1: Write to this register while serial I/O is neither transmitting nor receiving.
Note 2: To receive data, set the corresponding port direction bit for SINi to "0" (input mode).

**Figure 14.2.2. S3C and S4C Registers, S3BRG and S4BRG Registers, and S3TRR and S4TRR Registers**

**Table 14.2.1. SI/O3 and SI/O4 Specifications**

| Item | Specification |
|---|---|
| Transfer data format | • Transfer data length: 8 bits |
| Transfer clock | • SiC (i=3, 4) register's SMi6 bit = "1" (internal clock) : fj/ 2(n+1)<br>fj = f1SIO, f2SIO, f8SIO, f32SIO. n=Setting value of SiBRG register $00_{16}$ to $FF_{16}$.<br>• SMi6 bit = "0" (external clock) : Input from CLKi pin (Note 1) |
| Transmission/reception start condition | • Before transmission/reception can start, the following requirements must be met<br>Write transmit data to the SiTRR register (Notes 2, 3) |
| Interrupt request generation timing | • When SiC register's SMi4 bit = 0<br>The rising edge of the last transfer clock pulse (Note 4)<br>• When SMi4 = 1<br>The falling edge of the last transfer clock pulse (Note 4) |
| CLKi pin fucntion | I/O port, transfer clock input, transfer clock output |
| SOUTi pin function | I/O port, transmit data output, high-impedance |
| SINi pin function | I/O port, receive data input |
| Select function | • LSB first or MSB first selection<br>Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected<br>• Function for setting an SOUTi initial value set function<br>When the SiC register's SMi6 bit = 0 (external clock), the SOUTi pin output level while not tranmitting can be selected.<br>• CLK polarity selection<br>Whether transmit data is output/input timing at the rising edge or falling edge of transfer clock can be selected. |

Note 1: To set the SiC register's SMi6 bit to "0" (external clock), follow the procedure described below.
• If the SiC register's SMi4 bit = 0, write transmit data to the SiTRR register while input on the CLKi pin is high. The same applies when rewriting the SiC register's SMi7 bit.
• If the SMi4 bit = 1, write transmit data to the SiTRR register while input on the CLKi pin is low. The same applies when rewriting the SMi7 bit.
• Because shift operation continues as long as the transfer clock is supplied to the SI/Oi circuit, stop the transfer clock after supplying eight pulses. If the SMi6 bit = 1 (internal clock), the transfer clock automatically stops.

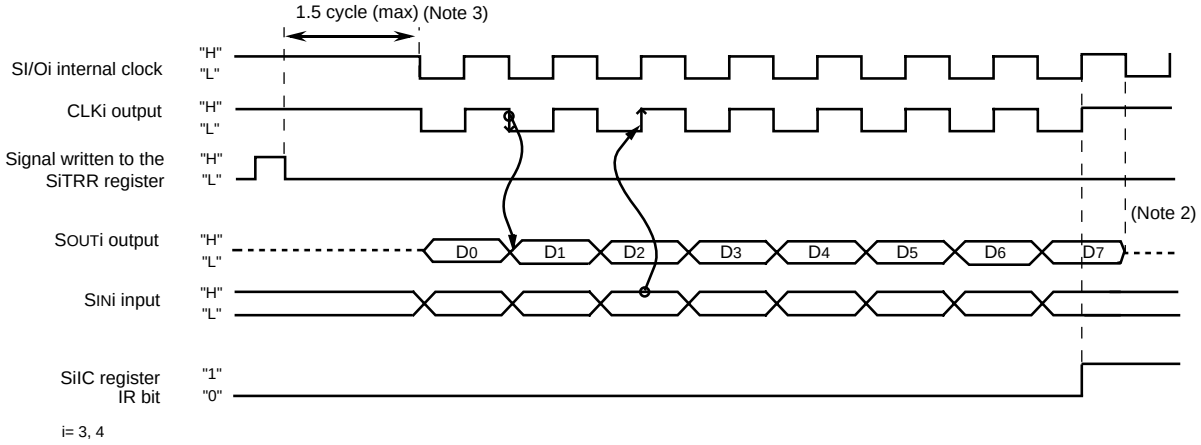
Note 2: Unlike UART0 to UART2, SI/Oi (i = 3 to 4) is not separated between the transfer register and buffer. Therefore, do not write the next transmit data to the SiTRR register during transmission.

Note 3: When the SiC register's SMi6 bit = 1 (internal clock), SOUTi retains the last data for a 1/2 transfer clock period after completion of transfer and, thereafter, goes to a high-impedance state. However, if transmit data is written to the SiTRR register during this period, SOUTi immediately goes to a high-impedance state, with the data hold time thereby reduced.

Note 4: When the SiC register's SMi6 bit = 1 (internal clock), the transfer clock stops in the high state if the SMi4 bit = 0, or stops in the low state if the SMi4 bit = 1.

### 14.2.1 SI/Oi Operation Timing

Figure 14.2.1.1 shows the SI/Oi operation timing

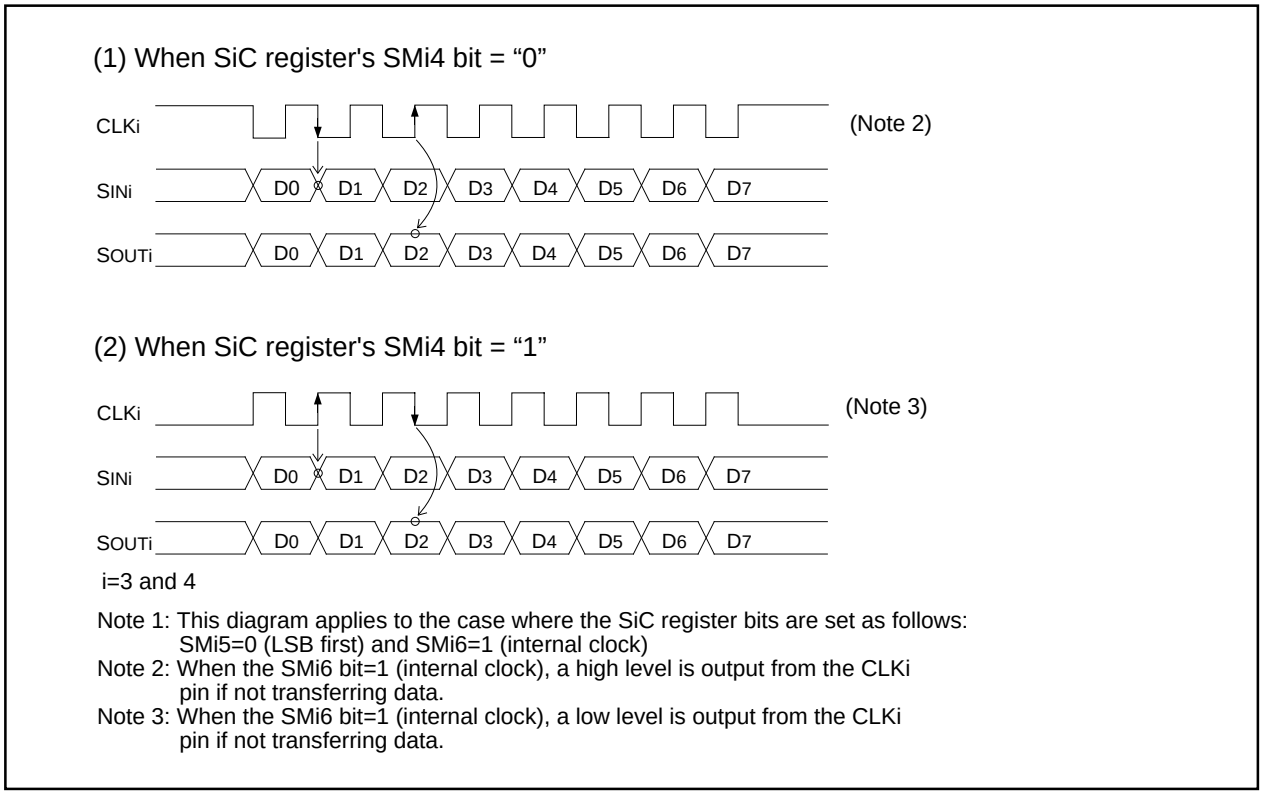



Note 1: This diagram applies to the case where the SiC register bits are set as follows:
SMi2=0 (SOUTi output), SMi3=1 (SOUTi output, CLKi function), SMi4=0 (transmit data output at the falling edge and receive data input at the rising edge of the transfer clock), SMi5=0 (LSB first) and SMi6=1 (internal clock)
Note 2: When the SMi6 bit = 1 (internal clock), the SOUTi pin is placed in the high-impedance state after the transfer finishes.
Note 3: If the SMi6 bit=0 (internal clock), the serial I/O starts sending or receiving data a maximum of 1.5 transfer clock cycles after writing to the SiTRR register.

**Figure 14.2.1.1. SI/Oi Operation Timing**

### 14.2.2 CLK Polarity Selection

The SiC register's SMi4 bit allows selection of the polarity of the transfer clock. Figure 14.2.2.1 shows the polarity of the transfer clock.

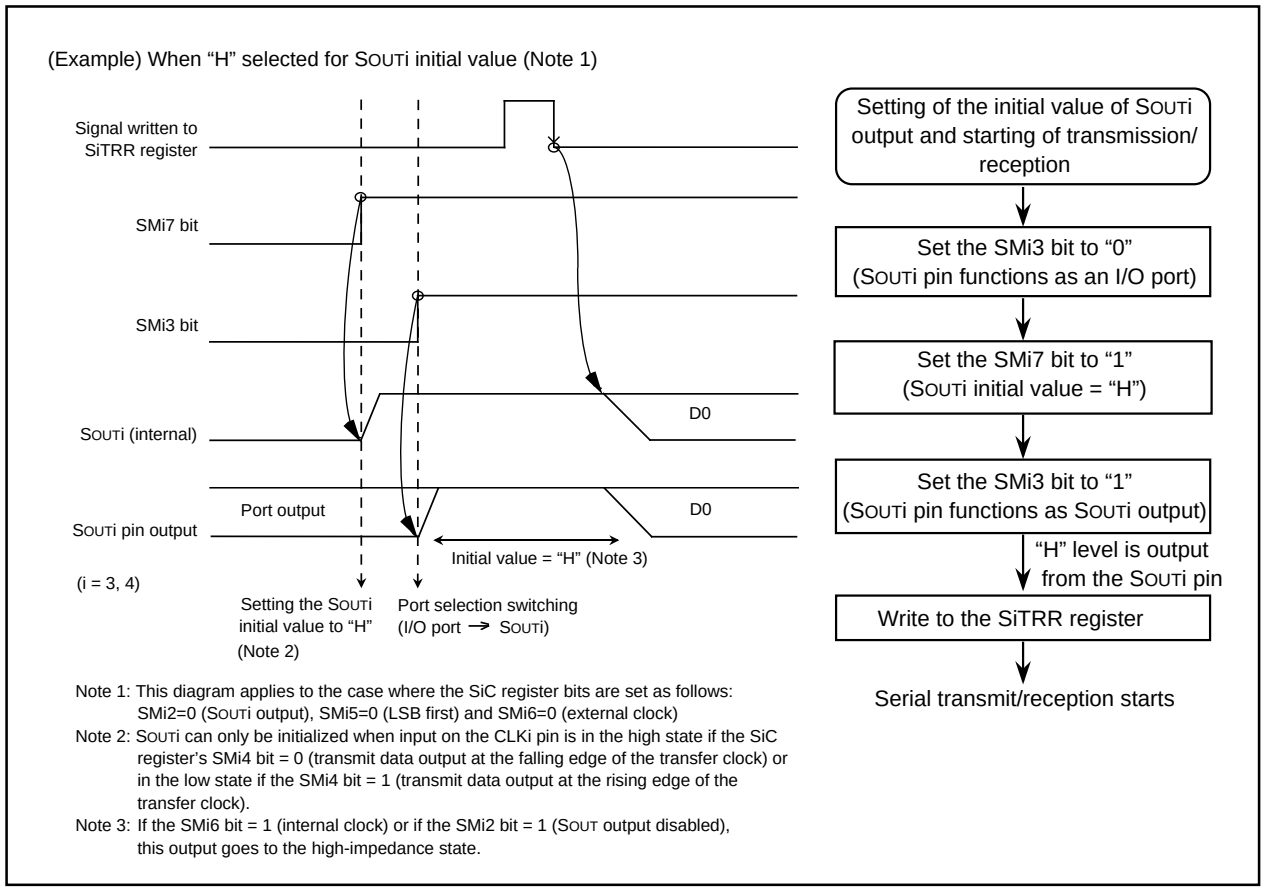



Note 1: This diagram applies to the case where the SiC register bits are set as follows:
SMi5=0 (LSB first) and SMi6=1 (internal clock)
Note 2: When the SMi6 bit=1 (internal clock), a high level is output from the CLKi pin if not transferring data.
Note 3: When the SMi6 bit=1 (internal clock), a low level is output from the CLKi pin if not transferring data.

**Figure 14.2.2.1. Polarity of Transfer Clock**

### 14.2.3 Functions for Setting an SOUTi Initial Value

If the SiC register's SMi6 bit = 0 (external clock), the SOUTi pin output can be fixed high or low when not transferring. Figure 14.2.3.1 shows the timing chart for setting an SOUTi initial value and how to set it.




Note 1: This diagram applies to the case where the SiC register bits are set as follows: SMi2=0 (SOUTi output), SMi5=0 (LSB first) and SMi6=0 (external clock)
Note 2: SOUTi can only be initialized when input on the CLKi pin is in the high state if the SiC register's SMi4 bit = 0 (transmit data output at the falling edge of the transfer clock) or in the low state if the SMi4 bit = 1 (transmit data output at the rising edge of the transfer clock).
Note 3: If the SMi6 bit = 1 (internal clock) or if the SMi2 bit = 1 (SOUT output disabled), this output goes to the high-impedance state.

**Figure 14.2.3.1. SOUTi's Initial Value Setting**

# 15. A-D Converter

The microcomputer contains one A-D converter circuit based on 10-bit successive approximation method configured with a capacitive-coupling amplifier. The analog inputs share the pins with P10$_0$ to P10$_7$ (AN$_0$ to AN$_7$), P0$_0$ to P0$_7$ (AN$_{00}$ to AN$_{07}$), and P1$_0$ to P1$_3$, P9$_3$, P9$_5$ to P9$_7$ (AN$_{20}$ to AN$_{27}$). Similarly, $\overline{\text{ADTRG}}$ input shares the pin with P1$_5$. Therefore, when using these inputs, make sure the corresponding port direction bits are set to "0" (= input mode). Note that P1$_0$ to P1$_3$, P9$_3$, P9$_5$ to P9$_7$ (AN$_{20}$ to AN$_{27}$) are available only in the 80-pin package.

When not using the A-D converter, set the VCUT bit to "0" (= $V_{REF}$ unconnected), so that no current will flow from the $V_{REF}$ pin into the resistor ladder, helping to reduce the power consumption of the chip.

The A-D conversion result is stored in the ADi register bits for AN$_i$, AN$_{0i}$, and AN$_{2i}$ pins (i = 0 to 7).

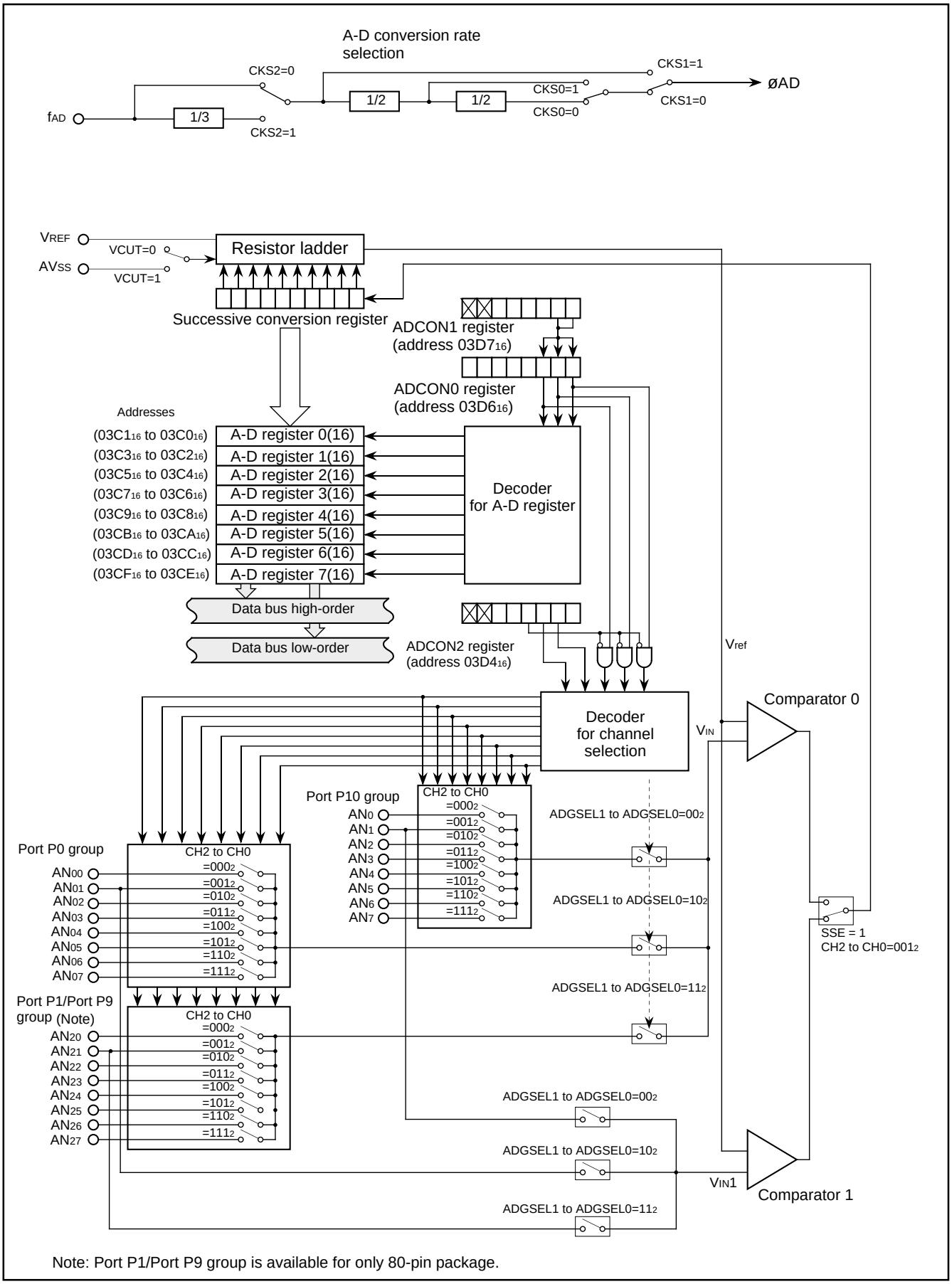
Table 15.1 shows the A-D converter performance. Figure 15.1 shows the A-D converter block diagram and Figures 15.2 to 15.4 show the A-D converter associated with registers.

**Table 15.1 A-D Converter Performance**

| Item | Performance |
|---|---|
| A-D Conversion Method | Successive approximation (capacitive coupling amplifier) |
| Analog Input Voltage (Note 1) | 0V to $AV_{CC}$ ($V_{CC}$) |
| Operating Clock $f_{AD}$ (Note 2) | $f_{AD}$/divided-by-2 or $f_{AD}$/divided-by-3 or $f_{AD}$/divided-by-4 or $f_{AD}$/divided-by-6 or $f_{AD}$/divided-by-12 or $f_{AD}$ |
| Resolution | 8-bit or 10-bit (selectable) |
| Integral Nonlinearity Error | When $AV_{CC}$ = $V_{REF}$ = 5V<br>• With 8-bit resolution: ±2LSB<br>• With 10-bit resolution<br>- AN$_0$ to AN$_7$ input : ±3LSB<br>- AN$_{00}$ to AN$_{07}$ input and AN$_{20}$ to AN$_{27}$ input : ±7LSB<br>When $AV_{CC}$ = $V_{REF}$ = 3.3V<br>• With 8-bit resolution: ±2LSB<br>• With 10-bit resolution<br>- AN$_0$ to AN$_7$ input : ±5LSB<br>- AN$_{00}$ to AN$_{07}$ input and AN$_{20}$ to AN$_{27}$ input : ±7LSB |
| Operating Modes | One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, repeat sweep mode 1, simultaneous sample sweep mode and delayed trigger mode 0,1 |
| Analog Input Pins | 8 pins (AN$_0$ to AN$_7$) + 8 pins (AN$_{00}$ to AN$_{07}$) + 8 pins (AN$_{20}$ to AN$_{27}$) (80pin-ver.)<br>8 pins (AN$_0$ to AN$_7$) + 4 pins (AN$_{00}$ to AN$_{03}$) + 1 pin (AN$_{24}$) (64pin-ver.) |
| Conversion Speed Per Pin | • Without sample and hold function<br>8-bit resolution: 49 $f_{AD}$ cycles, 10-bit resolution: 59 $f_{AD}$ cycles<br>• With sample and hold function<br>8-bit resolution: 28 $f_{AD}$ cycles, 10-bit resolution: 33 $f_{AD}$ cycles |

Note 1: Not dependent on use of sample and hold function.

Note 2: Set the fAD frequency to 10 MHz or less.
Without sample-and-hold function, set the fAD frequency to 250kHz or more.
With the sample and hold function, set the fAD frequency to 1MHz or more.




Note: Port P1/Port P9 group is available for only 80-pin package.

**Figure 15.1 A-D Converter Block Diagram**

A-D control register 0 (Note)

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

Symbol: ADCON0 — Address: $03D6_{16}$ — After reset: $00000XXX_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| CH0 | Analog Input Pin Select Bit | Function varies with each operation mode | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A-D Operation Mode Select Bit 0 | b4 b3<br>0 0 : One-shot mode or Delayed trigger mode 0,1<br>0 1 : Repeat mode<br>1 0 : Single sweep mode or Simultaneous sample sweep mode<br>1 1 : Repeat sweep mode 0 or Repeat sweep mode 1 | RW |
| MD1 | | | RW |
| TRG | Trigger Select Bit | 0 : Software trigger<br>1 : Hardware trigger | RW |
| ADST | A-D Conversion Start Flag | 0 : A-D conversion disabled<br>1 : A-D conversion started | RW |
| CKS0 | Frequency Select Bit 0 | See **Table 15.2 A-D Conversion Frequency Select** | RW |

Note: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.

A-D control register 1 (Note 1)

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

Symbol: ADCON1 — Address: $03D7_{16}$ — After reset: $00_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SCAN0 | A-D Sweep Pin Select Bit | Function varies with each operation mode | RW |
| SCAN1 | | | RW |
| MD2 | A-D Operation Mode Select Bit 1 | 0 : Other than repeat sweep mode 1<br>1 : Repeat sweep mode 1 | RW |
| BITS | 8/10-Bit Mode Select Bit | 0 : 8-bit mode<br>1 : 10-bit mode | RW |
| CKS1 | Frequency Select Bit 1 | See **Table 15.2 A-D Conversion Frequency Select** | RW |
| VCUT | VREF Connect Bit (Note 2) | 0 : VREF not connected<br>1 : VREF connected | RW |
| — (b7-b6) | Nothing is assigned. When write, set to "0". When read, its content is "0". | | — |

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: If the VCUT bit is reset from "0" (VREF unconnected) to "1" (VREF connected), wait for 1 µs or more before startingA-D conversion.

A-D control register 2 (Note 1)

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

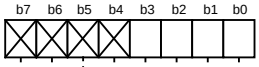
Symbol: ADCON2 — Address: $03D4_{16}$ — After reset: $00_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SMP | A-D Conversion Method Select Bit | 0 : Without sample and hold<br>1 : With sample and hold | RW |
| ADGSEL0 | A-D Input Group Select Bit | b2 b1<br>0 0 : Select port P10 group (ANi)<br>0 1 : Do not set<br>1 0 : Select port P0 group (AN0i)<br>1 1 : Select port P1/P9 group (AN2i) | RW |
| ADGSEL1 | | | RW |
| — (b3) | Reserved Bit | Set to "0" | RW |
| CKS2 | Frequency Select Bit 2 | See **Table 15.2 A-D Conversion Frequency Select** | RW |
| TRG1 | Trigger Select Bit | Function varies with each operation mode | RW |
| — (b7-b6) | Nothing is assigned. When write, set to "0". When read, its content is "0". | | — |

Note 1: If the ADCON2 register is rewritten during A-D conversion, the conversion result will be indeterminate.

**Figure 15.2 ADCON0 to ADCON2 Registers**

A-D trigger control register (Note 1, 2)




| Symbol | Address | After reset |
|---|---|---|
| ADTRGCON | 03D2$_{16}$ | 00$_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SSE | A-D Operation Mode Select Bit 2 | 0 : Other than simultaneous sample sweep mode or delayed trigger mode 0,1<br>1 : Simultaneous sample sweep mode or delayed trigger mode 0,1 | RW |
| DTE | A-D Operation Mode Select Bit 3 | 0 : Other than delayed trigger mode 0,1<br>1 : Delayed trigger mode 0,1 | RW |
| HPTRG0 | AN0 Trigger Select Bit | Function varies with each operation mode | RW |
| HPTRG1 | AN1 Trigger Select Bit | Function varies with each operation mode | RW |
| —— (b7-b4) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: If the ADTRGCON register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: Set “00$_{16}$” in this register in one-shot mode, repeat mode, single sweep mode, repeat sweep mode 0 and repeat sweep mode 1.

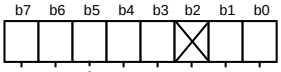
**Figure 15.3 ADTRGCON Register**

**Table 15.2 A-D Conversion Frequency Select**

| CKS2 | CKS1 | CKS0 | ØAD |
|---|---|---|---|
| 0 | 0 | 0 | Divided-by-4 of fAD |
| 0 | 0 | 1 | Divided-by-2 of fAD |
| 0 | 1 | 0 | fAD |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | Divided-by-12 of fAD |
| 1 | 0 | 1 | Divided-by-6 of fAD |
| 1 | 1 | 0 | Divided-by-3 of fAD |
| 1 | 1 | 1 | |

Note: Set the ØAD frequency to 10 MHz or less. The selected ØAD frequency is determined by a combination of the CKS0 bit in the ADCON0 register, CKS1 bit in the ADCON1 register and the CKS2 bit in the ADCON2 register.

A-D conversion status register 0 (Note 1)



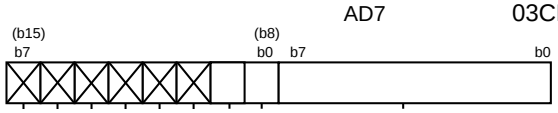

| Symbol | Address | After reset |
|---|---|---|
| ADSTAT0 | 03D3₁₆ | 00₁₆ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| ADERR0 | AN1 Trigger Status Flag | 0 : Without AN1 trigger during AN0 conversion<br>1 : With AN1 trigger during AN0 conversion | RW |
| ADERR1 | Conversion Termination Flag | 0 : Conversion not terminated<br>1 : Conversion terminated by Timer B0 underflow | RW |
| — (b2) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |
| ADTCSF | Delayed Trigger Sweep Status Flag | 0 : Sweep not in progress<br>1 : Sweep in progress | RO |
| ADSTT0 | AN0 Conversion Status Flag | 0 : AN0 conversion not in progress<br>1 : AN0 conversion in progress | RO |
| ADSTT1 | AN1 Conversion Status Flag | 0 : AN1 conversion not in progress<br>1 : AN1 conversion in progress | RO |
| ADSTRT0 | AN0 Conversion Completion Status Flag | 0 : AN0 conversion not completed<br>1 : AN0 conversion completed | RW |
| ADSTRT1 | AN1 Conversion Completion Status Flag | 0 : AN1 conversion not completed<br>1 : AN1 conversion completed | RW |

Note 1: ADSTAT0 register is valid only when the DTE bit in the ADTRGCON register is set to “1”.

A-D Register i (i=0 to 7)

| Symbol | Address | After reset |
|---|---|---|
| AD0 | 03C1₁₆ to 03C0₁₆ | Indeterminate |
| AD1 | 03C3₁₆ to 03C2₁₆ | Indeterminate |
| AD2 | 03C5₁₆ to 03C4₁₆ | Indeterminate |
| AD3 | 03C7₁₆ to 03C6₁₆ | Indeterminate |
| AD4 | 03C9₁₆ to 03C8₁₆ | Indeterminate |
| AD5 | 03CB₁₆ to 03CA₁₆ | Indeterminate |
| AD6 | 03CD₁₆ to 03CC₁₆ | Indeterminate |
| AD7 | 03CF₁₆ to 03CE₁₆ | Indeterminate |




| Function | | RW |
|---|---|---|
| When the BITS bit in the ADCON1 register is “1” (10-bit mode) | When the BITS bit in the ADCON1 register is “0” (8-bit mode) | |
| Eight low-order bits of A-D conversion result | A-D conversion result | RO |
| Two high-order bits of A-D conversion result | When read, its content is indeterminate | RO |
| Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

**Figure 15.4 ADSTAT0 Register and AD0 to AD7 Registers**

Timer B2 special mode register (Note 1)

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

| Symbol | Address | After reset |
|---|---|---|
| TB2SC | $039E_{16}$ | $X0000000_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PWCOM | Timer B2 Reload Timing Switch Bit (Note 2) | 0 : Timer B2 underflow<br>1 : Timer A output at odd-numbered | RW |
| IVPCR1 | Three-Phase Output Port $\overline{SD}$ Control Bit 1 (Note 3, 4, 7) | 0 : Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) disabled<br>1 : Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) enabled | RW |
| TB0EN | Timer B0 Operation Mode Select Bit | 0 : Other than A-D trigger mode<br>1 : A-D trigger mode (Note 5) | RW |
| TB1EN | Timer B1 Operation Mode Select Bit | 0 : Other than A-D trigger mode<br>1 : A-D trigger mode (Note 5) | RW |
| TB2SEL | Trigger Select Bit (Note 6) | 0 : TB2 interrupt<br>1 : Underflow of TB2 interrupt generation frequency setting counter [ICTB2] | RW |
| $\overline{\text{(b6-b5)}}$ | Reserved bits | Must set to "0" | RW |
| $\overline{\text{(b7)}}$ | Nothing is assigned. When write, set to 0 . When read, its content is | 0 . | — |

Note 1. Write to this register after setting the PRC1 bit in the PRCR register to "1" (write enabled).
Note 2. If the INV11 bit is "0" (three-phase mode 0) or the INV06 bit is "1" (triangular wave modulation mode), set this bit to "0" (timer B2 underflow).
Note 3. When setting the IVPCR1 bit to "1" (three-phase output forcible cutoff by $\overline{SD}$ pin input enabled), Set the PD8_5 bit to "0" (= input mode).
Note 4. Related pins are U($P8_0$), $\overline{U}$($P8_1$), V($P7_2$), $\overline{V}$($P7_3$), W($P7_4$), $\overline{W}$($P7_5$). After forcible cutoff, input "H" to the $P8_5$/$\overline{NMI}$/$\overline{SD}$ pin. Set the IVPCR1 bit to "0", and this forcible cutoff will be reset. If L is input to the $P8_5$/$\overline{NMI}$/$\overline{SD}$ pin, a three-phase motor control timer output will be disabled (INV03=0). At this time, when the IVPCR1 bit is "0", the target pins changes to programmable I/O port. When the IVPCR1 bit is "1", the target pins changes to high-impedance state regardless of which functions of those pins are used.
Note 5. When this bit is used in delayed trigger mode 0, set the TB0EN and TB1EN bits to "1" (A-D trigger mode).
Note 6. When setting the TB2SEL bit to "1" (underflow of TB2 interrupt generation frequency setting counter[ICTB2]), Set the INV02 bit to "1" (three-phase motor control timer function).
Note 7. Refer to "**17.6 Digital Debounce function**" for the $\overline{SD}$ input

**Figure 15.5 TB2SC Register**

# 15.1 Operation Modes

## 15.1.1 One-Shot Mode

In one-shot mode, analog voltage applied to a selected pin is once converted to a digital code. Table 15.1.1.1 shows the one-shot mode specifications. Figure 15.1.1.1 shows the operation example in one-shot mode. Figure 15.1.1.2 shows the ADCON0 to ADCON2 registers in one-shot mode.

**Table 15.1.1.1 One-shot Mode Specifications**

| Item | Specification |
|---|---|
| Function | The CH2 to CH0 bits in the ADCON0 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to a selected pin is once converted to a digital code |
| A-D Conversion Start Condition | • When the TRG bit in the ADCON0 register is “0” (software trigger)<br>Set the ADST bit in the ADCON0 register to “1” (A-D conversion started)<br>• When the TRG bit in the ADCON0 register is “1” (hardware trigger)<br>The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to “1” (A-D conversion started) |
| A-D Conversion Stop Condition | • A-D conversion completed (If a software trigger is selected, the ADST bit is set to “0” (A-D conversion halted)).<br>• Set the ADST bit to “0” |
| Interrupt Request Generation Timing | A-D conversion completed |
| Analog Input Pin | Select one pin from $AN_0$ to $AN_7$, $AN_{00}$ to $AN_{07}$, $AN_{20}$ to $AN_{27}$ |
| Readout of A-D Conversion Result | Readout one of the AD0 to AD7 registers that corresponds to the selected pin |

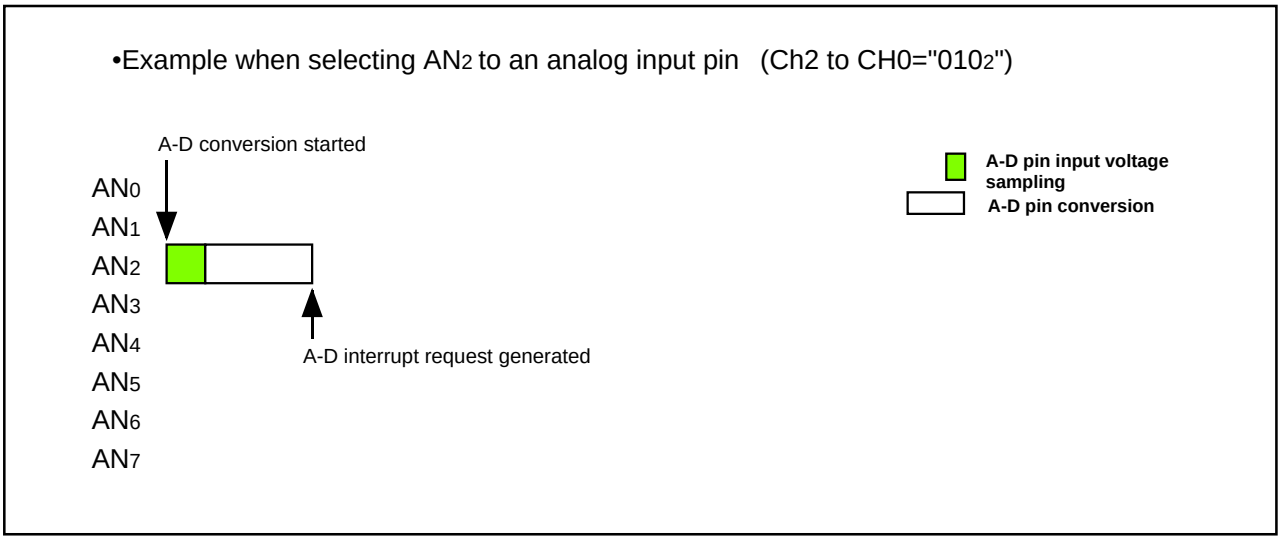
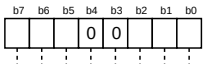



**Figure 15.1.1.1 Operation Example in One-Shot Mode**

A-D control register 0 (Note 1)

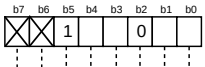



Symbol ADCON0 — Address 03D6₁₆ — After reset 00000XXX₂

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| CH0 | Analog Input Pin Select Bit (Note 2, 3) | b2 b1 b0<br>0 0 0 : Select AN0<br>0 0 1 : Select AN1<br>0 1 0 : Select AN2<br>0 1 1 : Select AN3<br>1 0 0 : Select AN4<br>1 0 1 : Select AN5<br>1 1 0 : Select AN6<br>1 1 1 : Select AN7 | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A-D Operation Mode Select Bit 0 (Note 3) | b4 b3<br>0 0 : One-shot mode or delayed trigger mode 0,1 | RW |
| MD1 | | | RW |
| TRG | Trigger Select Bit | 0 : Software trigger<br>1 : Hardware trigger ($\overline{\text{ADTRG}}$ trigger) | RW |
| ADST | A-D Conversion Start Flag | 0 : A-D conversion disabled<br>1 : A-D conversion started | RW |
| CKS0 | Frequency Select Bit 0 | See **Table 15.2 A-D Conversion Frequency Select** | RW |

Note 1: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: AN00 to AN07 and AN20 to AN27 can be used in the same way as AN0 to AN7. Use the ADGSEL1 to ADGSEL 0 bits in the ADCON2 register to select the desired pin.
Note 3: After rewriting the MD1 to MD0 bits, set the CH2 to CH0 bits over again using an another instruction.

A-D control register 1 (Note 1)

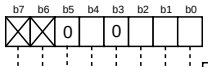



Symbol ADCON1 — Address 03D7₁₆ — After reset 00₁₆

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SCAN0 | A-D Sweep Pin Select Bit | Invalid in one-shot mode | RW |
| SCAN1 | | | RW |
| MD2 | A-D Operation Mode Select Bit 1 | 0 : Any mode other than repeat sweep mode 1 | RW |
| BITS | 8/10-Bit Mode Select Bit | 0 : 8-bit mode<br>1 : 10-bit mode | RW |
| CKS1 | Frequency Select Bit 1 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| VCUT | VREF Connect Bit (Note 2) | 1 : VREF connected | RW |
| — (b7-b6) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: If the VCUT bit is reset from “0” (VREF unconnected) to “1” (VREF connected), wait for 1 µs or more before starting A-D conversion.

A-D control register 2 (Note 1)




Symbol ADCON2 — Address 03D4₁₆ — After reset 00₁₆

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SMP | A-D Conversion Method Select Bit | 0 : Without sample and hold<br>1 : With sample and hold | RW |
| ADGSEL0 | A-D Input Group Select Bit | b2 b1<br>0 0 : Select port P10 group (ANi)<br>0 1 : Do not set<br>1 0 : Select port P0 group (AN0i)<br>1 1 : Select port P1/P9 group (AN2i) | RW |
| ADGSEL1 | | | RW |
| — (b3) | Reserved Bit | Set to “0” | RW |
| CKS2 | Frequency Select Bit 2 | See **Table 15.2 A-D Conversion Frequency Select** | RW |
| TRG1 | Trigger Select Bit 1 | Set to "0" in one-shot mode | RW |
| — (b7-b6) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: If the ADCON2 register is rewritten during A-D conversion, the conversion result will be indeterminate.

**Figure 15.1.1.2 ADCON0 to ADCON2 Registers in One-Shot Mode**

### 15.1.2 Repeat mode

In repeat mode, analog voltage applied to a selected pin is repeatedly converted to a digital code. Table 15.1.2.1 shows the repeat mode specifications. Figure 15.1.2.1 shows the operation example in repeat mode. Figure 15.1.2.2 shows the ADCON0 to ADCON2 registers in repeat mode.

**Table 15.1.2.1 Repeat Mode Specifications**

| Item | Specification |
|---|---|
| Function | The CH2 to CH0 bits in the ADCON0 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to a selected pin is repeatedly converted to a digital code |
| A-D Conversion Start Condition | • When the TRG bit in the ADCON0 register is “0” (software trigger)<br>Set the ADST bit in the ADCON0 register to “1” (A-D conversion started)<br>• When the TRG bit in the ADCON0 register is “1” (hardware trigger)<br>The $\overline{\text{AD}}_{\text{TRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to “1” (A-D conversion started) |
| A-D Conversion Stop Condition | Set the ADST bit to “0” (A-D conversion halted) |
| Interrupt Request Generation Timing | None generated |
| Analog Input Pin | Select one pin from $AN_0$ to $AN_7$, $AN_{00}$ to $AN_{07}$ and $AN_{20}$ to $AN_{27}$ |
| Readout of A-D Conversion Result | Readout one of the AD0 to AD7 registers that corresponds to the selected pin |

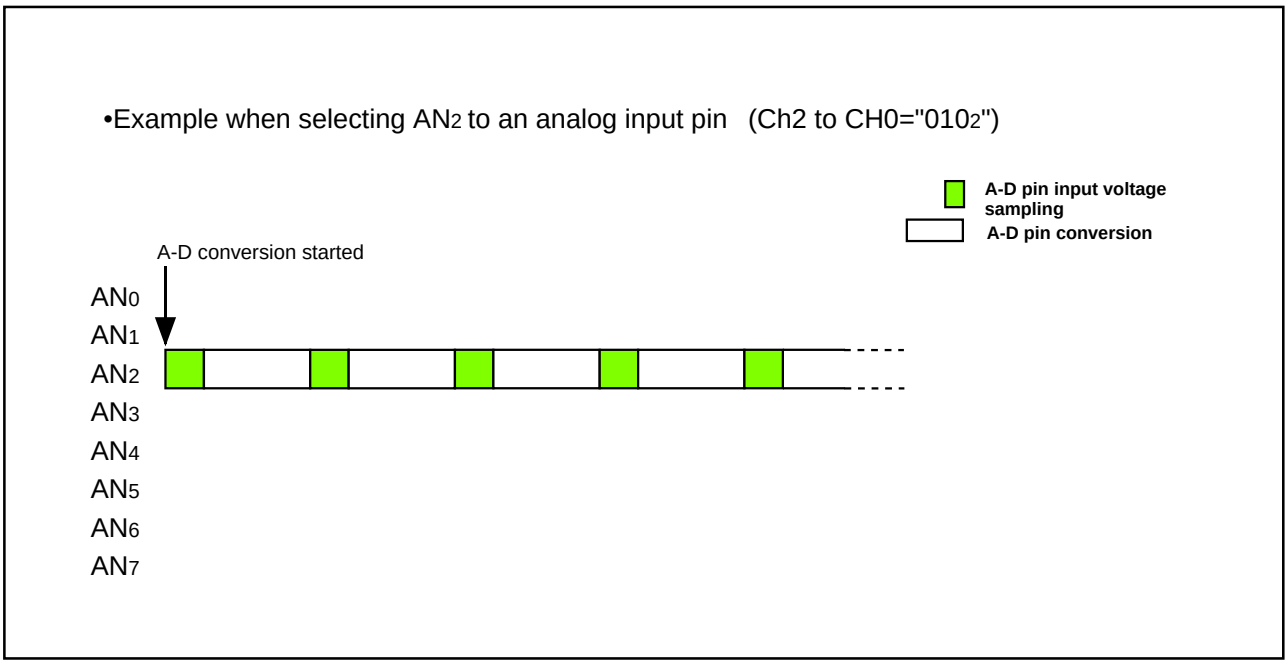
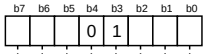



**Figure 15.1.2.1 Operation Example in Repeat Mode**

A-D control register 0 (Note 1)




Symbol: ADCON0 | Address: $03D6_{16}$ | After reset: $00000XXX_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| CH0 | Analog Input Pin Select Bit (Note 2, 3) | b2 b1 b0<br>0 0 0 : Select $AN_0$<br>0 0 1 : Select $AN_1$<br>0 1 0 : Select $AN_2$<br>0 1 1 : Select $AN_3$<br>1 0 0 : Select $AN_4$<br>1 0 1 : Select $AN_5$<br>1 1 0 : Select $AN_6$<br>1 1 1 : Select $AN_7$ | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A-D Operation Mode Select Bit 0 (Note 3) | b4 b3<br>0 1 : Repeat mode | RW |
| MD1 | | | RW |
| TRG | Trigger Select Bit | 0 : Software trigger<br>1 : Hardware trigger ($\overline{ADTRG}$ trigger) | RW |
| ADST | A-D Conversion Start Flag | 0 : A-D conversion disabled<br>1 : A-D conversion started | RW |
| CKS0 | Frequency Select Bit 0 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |

Note 1: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: $AN_{00}$ to $AN_{07}$ and $AN_{20}$ to $AN_{27}$ can be used in the same way as $AN_0$ to $AN_7$. Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
Note 3: After rewriting the MD1 to MD0 bits, set the CH2 to CH0 bits over again using an another instruction.

A-D control register 1 (Note 1)

Symbol: ADCON1 | Address: $03D7_{16}$ | After reset: $00_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SCAN0 | A-D Sweep Pin Select Bit | Invalid in repeat mode | RW |
| SCAN1 | | | RW |
| MD2 | A-D Operation Mode Select Bit 1 | 0 : Any mode other than repeat sweep mode 1 | RW |
| BITS | 8/10-Bit Mode Select Bit | 0 : 8-bit mode<br>1 : 10-bit mode | RW |
| CKS1 | Frequency Select Bit 1 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| VCUT | $V_{REF}$ connect bit (Note 2) | 1 : $V_{REF}$ connected | RW |
| — (b7-b6) | Nothing is assigned. When write, set to "0". When read, its content is "0". | | — |

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: If the VCUT bit is reset from "0" ($V_{REF}$ unconnected) to "1" ($V_{REF}$ connected), wait for 1 µs or more before starting A-D conversion.

A-D control register 2 (Note 1)

Symbol: ADCON2 | Address: $03D4_{16}$ | After reset: $00_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SMP | A-D Conversion Method Select Bit | 0 : Without sample and hold<br>1 : With sample and hold | RW |
| ADGSEL0 | A-D Input Group Select Bit | b2 b1<br>0 0 : Select port P10 group ($AN_i$)<br>0 1 : Do not set<br>1 0 : Select port P0 group ($AN_{0i}$)<br>1 1 : Select port P1/P9 group ($AN_{2i}$) | RW |
| ADGSEL1 | | | RW |
| — (b3) | Reserved Bit | Set to "0" | RW |
| CKS2 | Frequency Select Bit 2 | See **Table 15.2 A-D Conversion Frequency Select** | RW |
| TRG1 | Trigger Select Bit 1 | Set to "0" in repeat mode | RW |
| — (b7-b6) | Nothing is assigned. When write, set to "0". When read, its content is "0". | | — |

Note 1: If the ADCON2 register is rewritten during A-D conversion, the conversion result will be indeterminate.

**Figure 15.1.2.2 ADCON0 to ADCON2 Registers in Repeat Mode**

### 15.1.3 Single Sweep Mode

In single sweep mode, analog voltage is applied to the selected pins are converted one-by-one to a digital code. Table 15.1.3.1 shows the single sweep mode specifications. Figure 15.1.3.1 shows the operation example in single sweep mode. Figure 15.1.3.2 shows the ADCON0 to ADCON2 registers in single sweep mode.

**Table 15.1.3.1 Single Sweep Mode Specifications**

| Item | Specification |
|---|---|
| Function | The SCAN1 to SCAN0 bits in the ADCON1 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to the selected pins is converted one-by-one to a digital code |
| A-D Conversion Start Condition | • When the TRG bit in the ADCON0 register is “0” (software trigger)<br>Set the ADST bit in the ADCON0 register to “1” (A-D conversion started)<br>• When the TRG bit in the ADCON0 register is “1” (hardware trigger)<br>The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to “1” (A-D conversion started) |
| A-D Conversion Stop Condition | • A-D conversion completed(When selecting a software trigger, the ADST bit is set to “0” (A-D conversion halted)).<br>• Set the ADST bit to “0” |
| Interrupt Request Generation Timing | A-D conversion completed |
| Analog Input Pin | Select from $AN_0$ to $AN_1$ (2 pins), $AN_0$ to $AN_3$ (4 pins), $AN_0$ to $AN_5$ (6 pins), $AN_0$ to $AN_7$ (8 pins) (Note 1) |
| Readout of A-D Conversion Result | Readout one of the AD0 to AD7 registers that corresponds to the selected pin |

Note 1. $AN_{00}$ to $AN_{07}$ and $AN_{20}$ to $AN_{27}$ can be used in the same way as $AN_0$ to $AN_7$. However, all input pins need to belong to the same group.

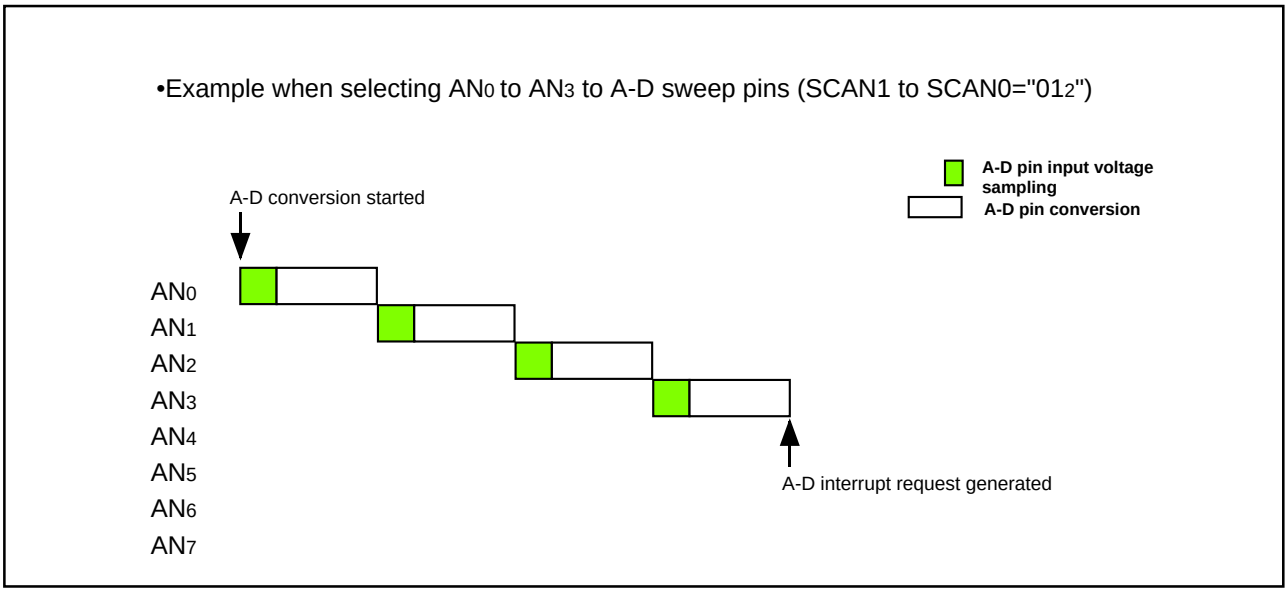
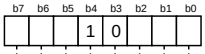



**Figure 15.1.3.1 Operation Example in Single Sweep Mode**

A-D control register 0 (Note 1)




Symbol ADCON0 — Address 03D6₁₆ — After reset 00000XXX₂

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| CH0 | Analog Input Pin Select Bit | Invalid in single sweep mode | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A-D Operation Mode Select Bit 0 | b4 b3<br>1 0 : Single sweep mode or simultaneous sample sweep mode | RW |
| MD1 | | | RW |
| TRG | Trigger Select Bit | 0 : Software trigger<br>1 : Hardware trigger ($\overline{\text{ADTRG}}$ trigger) | RW |
| ADST | A-D Conversion Start Flag | 0 : A-D conversion disabled<br>1 : A-D conversion started | RW |
| CKS0 | Frequency Select Bit 0 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |

Note 1: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.

A-D control register 1 (Note 1)

Symbol ADCON1 — Address 03D7₁₆ — After reset 00₁₆

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SCAN0 | A-D Sweep Pin Select Bit (Note 2) | When selecting single sweep mode<br>b1 b0<br>0 0 : AN0 to AN1 (2 pins)<br>0 1 : AN0 to AN3 (4 pins)<br>1 0 : AN0 to AN5 (6 pins)<br>1 1 : AN0 to AN7 (8 pins) | RW |
| SCAN1 | | | RW |
| MD2 | A-D Operation Mode Select Bit 1 | 0 : Any mode other than repeat sweep mode 1 | RW |
| BITS | 8/10-Bit Mode Select Bit | 0 : 8-bit mode<br>1 : 10-bit mode | RW |
| CKS1 | Frequency Select Bit 1 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| VCUT | VREF Connect Bit (Note 3) | 1 : VREF connected | RW |
| — (b7-b6) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: AN00 to AN07 and AN20 to AN27 can be used in the same way as AN0 to AN7. Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
Note 3: If the VCUT bit is reset from “0” (VREF unconnected) to “1” (VREF connected), wait for 1 µs or more before starting A-D conversion.

A-D control register 2 (Note 1)

Symbol ADCON2 — Address 03D4₁₆ — After reset 00₁₆

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SMP | A-D Conversion Method Select Bit | 0 : Without sample and hold<br>1 : With sample and hold | RW |
| ADGSEL0 | A-D Input Group Select Bit | b2 b1<br>0 0 : Select port P10 group (ANi)<br>0 1 : Do not set<br>1 0 : Select port P0 group (AN0i)<br>1 1 : Select port P1/P9 group (AN2i) | RW |
| ADGSEL1 | | | RW |
| — (b3) | Reserved Bit | Set to “0” | RW |
| CKS2 | Frequency Select Bit 2 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| TRG1 | Trigger Select Bit 1 | Set to "0" in single sweep mode | RW |
| — (b7-b6) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: If the ADCON2 register is rewritten during A-D conversion, the conversion result will be indeterminate.

**Figure 15.1.3.2 ADCON0 Register to ADCON2 Registers in Single Sweep Mode**

### 15.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, analog voltage is applied to the selected pins are repeatedly converted to a digital code. Table 15.1.4.1 shows the repeat sweep mode 0 specifications. Figure 15.1.4.1 shows the operation example in repeat sweep mode 0. Figure 15.1.4.2 shows the ADCON0 to ADCON2 registers in repeat sweep mode 0.

**Table 15.1.4.1 Repeat Sweep Mode 0 Specifications**

| Item | Specification |
|---|---|
| Function | The SCAN1 to SCAN0 bits in the ADCON1 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to the selected pins is repeatedly converted to a digital code |
| A-D Conversion Start Condition | • When the TRG bit in the ADCON0 register is "0" (software trigger)<br>Set the ADST bit in the ADCON0 register to "1" (A-D conversion started)<br>• When the TRG bit in the ADCON0 register is "1" (Hardware trigger)<br>The $\overline{\text{ADTRG}}$ pin input changes state from "H" to "L" after setting the ADST bit to "1" (A-D conversion started) |
| A-D Conversion Stop Condition | Set the ADST bit to "0" (A-D conversion halted) |
| Interrupt Request Generation Timing | None generated |
| Analog Input Pin | Select from $AN_0$ to $AN_1$ (2 pins), $AN_0$ to $AN_3$ (4 pins), $AN_0$ to $AN_5$ (6 pins), $AN_0$ to $AN_7$ (8 pins) (Note 1) |
| Readout of A-D Conversion Result | Readout one of the AD0 to AD7 registers that corresponds to the selected pin |

Note 1. $AN_{00}$ to $AN_{07}$ and $AN_{20}$ to $AN_{27}$ can be used in the same way as $AN_0$ to $AN_7$. However, all input pins need to belong to the same group.

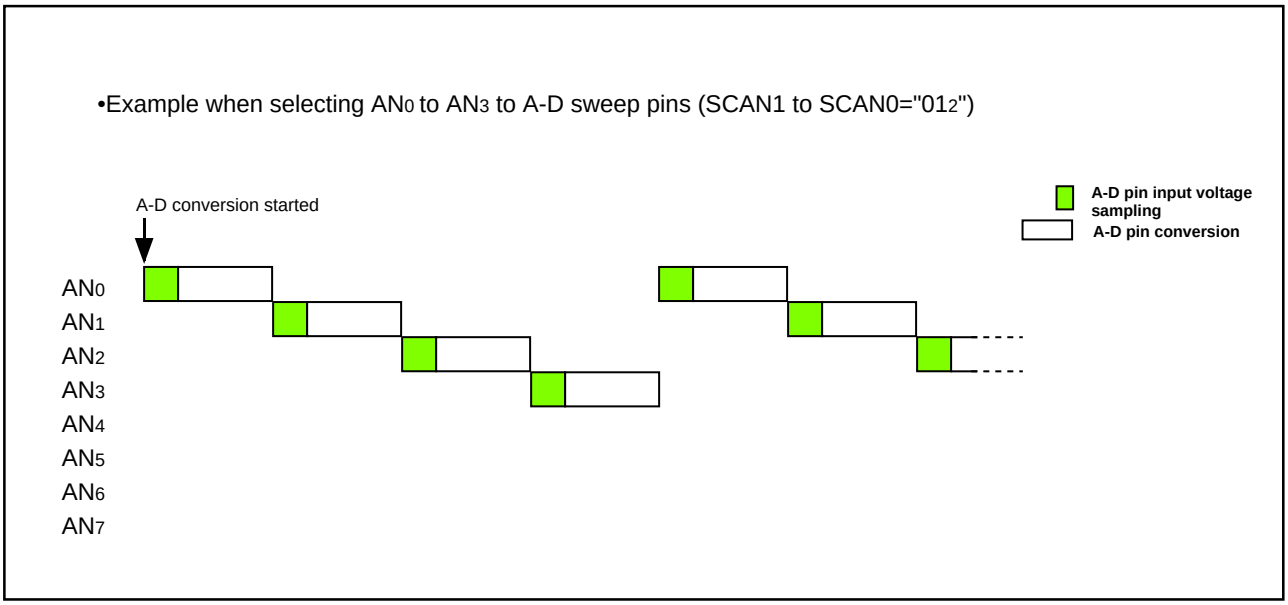
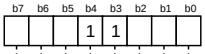



**Figure 15.1.4.1 Operation Example in Repeat Sweep Mode 0**

A-D control register 0 (Note 1)




Symbol: ADCON0 | Address: $03D6_{16}$ | After reset: $00000XXX_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| CH0 | Analog Input Pin Select Bit | Invalid in repeat sweep mode 0 | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A-D Operation Mode Select Bit 0 | b4 b3<br>1 1 : Repeat sweep mode 0 or Repeat sweep mode 1 | RW |
| MD1 | | | RW |
| TRG | Trigger Select Bit | 0 : Software trigger<br>1 : Hardware trigger ($\overline{\text{ADTRG}}$ trigger) | RW |
| ADST | A-D Conversion Start Flag | 0 : A-D conversion disabled<br>1 : A-D conversion started | RW |
| CKS0 | Frequency Select Bit 0 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |

Note 1: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.

A-D control register 1 (Note 1)

Symbol: ADCON1 | Address: $03D7_{16}$ | After reset: $00_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SCAN0 | A-D Sweep Pin Select Bit (Note 2) | When selecting repeat sweep mode 0<br>b1 b0<br>0 0 : $AN_0$ to $AN_1$ (2 pins)<br>0 1 : $AN_0$ to $AN_3$ (4 pins)<br>1 0 : $AN_0$ to $AN_5$ (6 pins)<br>1 1 : $AN_0$ to $AN_7$ (8 pins) | RW |
| SCAN1 | | | RW |
| MD2 | A-D Operation Mode Select Bit 1 | 0 : Any mode other than repeat sweep mode 1 | RW |
| BITS | 8/10-Bit Mode Select Bit | 0 : 8-bit mode<br>1 : 10-bit mode | RW |
| CKS1 | Frequency Select Bit 1 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| VCUT | $V_{REF}$ Connect Bit (Note 3) | 1 : $V_{REF}$ connected | RW |
| $\overline{\text{(b7-b6)}}$ | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: $AN_{00}$ to $AN_{07}$ and $AN_{20}$ to $AN_{27}$ can be used in the same way as $AN_0$ to $AN_7$ . Use the ADGSEL1 to ADGSET0 bits in the ADCON2 register to select the desired pin.
Note 3: If the VCUT bit is reset from “0” ($V_{REF}$ unconnected) to “1” ($V_{REF}$ connected), wait for 1 µs or more before starting A-D conversion.

A-D control register 2 (Note 1)

Symbol: ADCON2 | Address: $03D4_{16}$ | After reset: $00_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SMP | A-D Conversion Method Select Bit | 0 : Without sample and hold<br>1 : With sample and hold | RW |
| ADGSEL0 | A-D Input Group Select Bit | b2 b1<br>0 0 : Select port P10 group ($AN_i$)<br>0 1 : Do not set<br>1 0 : Select port P0 group ($AN_{0i}$)<br>1 1 : Select port P1/P9 group ($AN_{2i}$) | RW |
| ADGSEL1 | | | RW |
| $\overline{\text{(b3)}}$ | Reserved Bit | Set to “0” | RW |
| CKS2 | Frequency Select Bit 2 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| TRG1 | Trigger Select Bit 1 | Set to "0" in repeat sweep mode 0 | RW |
| $\overline{\text{(b7-b6)}}$ | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: If the ADCON2 register is rewritten during A-D conversion, the conversion result will be indeterminate.

**Figure 15.1.4.2 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 0**

### 15.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltage is applied to the all selected pins are converted to a digital code, with mainly used in the selected pins. Table 15.1.5.1 shows the repeat sweep mode 1 specifications. Figure 15.1.5.1 shows the operation example in repeat sweep mode 1. Figure 15.1.5.2 shows the ADCON0 to ADCON2 registers in repeat sweep mode 1.

**Table 15.1.5.1 Repeat Sweep Mode 1 Specifications**

| Item | Specification |
|---|---|
| Function | The SCAN1 to SCAN0 bits in the ADCON1 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register mainly select pins. Analog voltage applied to the all selected pins is repeatedly converted to a digital code<br>Example : When selecting $AN_0$<br>Analog voltage is converted to a digital code in the following order<br>$AN_0 \rightarrow AN_1 \rightarrow AN_0 \rightarrow AN_2 \rightarrow AN_0 \rightarrow AN_3$, and so on. |
| A-D Conversion Start Condition | • When the TRG bit in the ADCON0 register is “0” (software trigger)<br>Set the ADST bit in the ADCON0 register to “1” (A-D conversion started)<br>• When the TRG bit in the ADCON0 register is “1” (hardware trigger)<br>The $\overline{ADTRG}$ pin input changes state from “H” to “L” after setting the ADST bit to “1” (A-D conversion started) |
| A-D Conversion Stop Condition | Set the ADST bit to “0” (A-D conversion halted) |
| Interrupt Request Generation Timing | None generated |
| Analog Input Pins Mainly Used in A-D Conversions | Select from $AN_0$ (1 pins), $AN_0$ to $AN_1$ (2 pins), $AN_0$ to $AN_2$ (3 pins), $AN_0$ to $AN_3$ (4 pins) (Note 1) |
| Readout of A-D Conversion Result | Readout one of the AD0 to AD7 registers that corresponds to the selected pin |

Note1. $AN_{00}$ to $AN_{07}$ and $AN_{20}$ to $AN_{27}$ can be used in the same way as $AN_0$ to $AN_7$. However, all input pins need to belong to the same group.

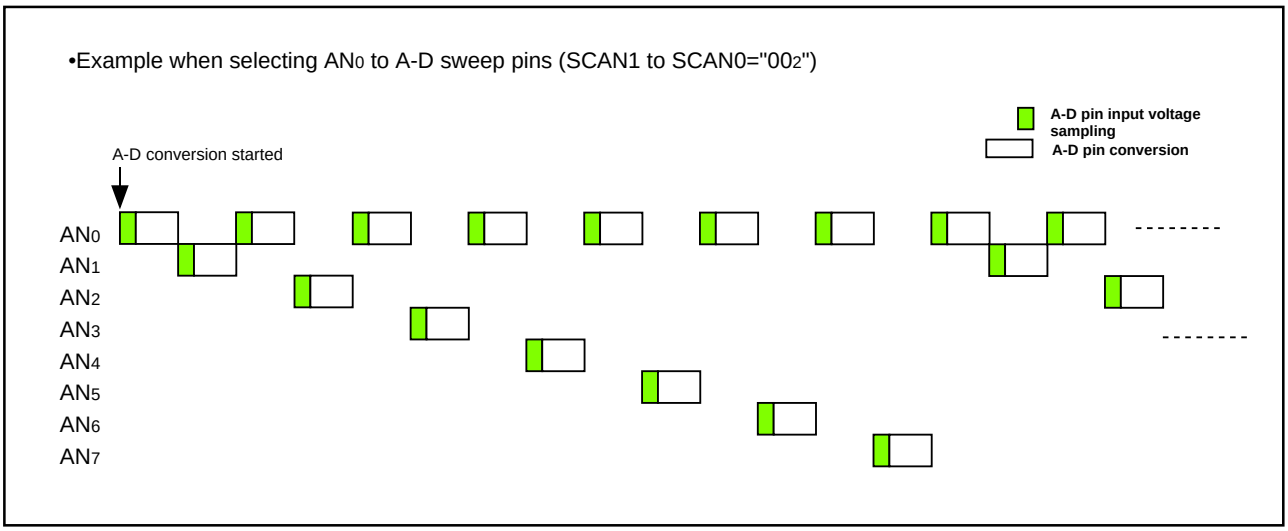
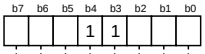



**Figure 15.1.5.1 Operation Example in Repeat Sweep Mode 1**

A-D control register 0 (Note 1)




Symbol ADCON0 — Address 03D6₁₆ — After reset 00000XXX₂

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| CH0 | Analog Input Pin Select Bit | Invalid in repeat sweep mode 1 | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A-D Operation Mode Select Bit 0 | b4 b3<br>1 1 : Repeat sweep mode 0 or Repeat sweep mode 1 | RW |
| MD1 | | | RW |
| TRG | Trigger Select Bit | 0 : Software trigger<br>1 : Hardware trigger (ADTRG trigger) | RW |
| ADST | A-D Conversion Start Flag | 0 : A-D conversion disabled<br>1 : A-D conversion started | RW |
| CKS0 | Frequency Select Bit 0 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |

Note 1: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.

A-D control register 1 (Note 1)

Symbol ADCON1 — Address 03D7₁₆ — After reset 00₁₆

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SCAN0 | A-D Sweep Pin Select Bit (Note2) | When selecting repeat sweep mode 1<br>b1 b0<br>0 0 : AN0 (1 pin)<br>0 1 : AN0 to AN1 (2 pins)<br>1 0 : AN0 to AN2 (3 pins)<br>1 1 : AN0 to AN3 (4 pins) | RW |
| SCAN1 | | | RW |
| MD2 | A-D Operation Mode Select Bit 1 | 1 : Repeat sweep mode 1 | RW |
| BITS | 8/10-Bit Mode Select Bit | 0 : 8-bit mode<br>1 : 10-bit mode | RW |
| CKS1 | Frequency Select Bit 1 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| VCUT | VREF Connect Bit (Note 3) | 1 : VREF connected | RW |
| (b7-b6) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: AN00 to AN07 and AN20 to AN27 can be used in the same way as AN0 to AN7 .
Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
Note 3: If the VCUT bit is reset from “0” (VREF unconnected) to “1” (VREF connected), wait for 1 µs or more before starting A-D conversion.

A-D control register 2 (Note 1)

Symbol ADCON2 — Address 03D4₁₆ — After reset 00₁₆

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SMP | A-D Conversion Method Select Bit | 0 : Without sample and hold<br>1 : With sample and hold | RW |
| ADGSEL0 | A-D Input Group Select Bit | b2 b1<br>0 0 : Select port P10 group (ANi)<br>0 1 : Do not set<br>1 0 : Select port P0 group (AN0i)<br>1 1 : Select port P1/P9 group (AN2i) | RW |
| ADGSEL1 | | | RW |
| (b3) | Reserved Bit | Set to “0” | RW |
| CKS2 | Frequency Select Bit 2 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| TRG1 | Trigger Select Bit 1 | Set to "0" in repeat sweep mode 1 | RW |
| (b7-b6) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: If the ADCON2 register is rewritten during A-D conversion, the conversion result will be indeterminate.

**Figure 15.1.5.2 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 1**

### 15.1.6 Simultaneous Sample Sweep Mode

In simultaneous sample sweep mode, analog voltage is applied to the selected pins are converted one-by-one to a digital code. At this time, the input voltage of AN0 and AN1 are sampled simultaneously using two circuits of sample and hold circuit. Table 15.1.6.1 shows the simultaneous sample sweep mode specifications. Figure 15.1.6.1 shows the operation example in simultaneous sample sweep mode. Figure 15.1.6.2 shows ADCON0 to ADCON2 registers and Figure 15.1.6.3 shows ADTRGCON registers in simultaneous sample sweep mode. Table 15.1.6.2 shows the trigger select bit setting in simultaneous sample sweep mode. In simultaneous sample sweep mode, Timer B0 underflow can be selected as a trigger by combining software trigger, $\overline{\text{ADTRG}}$ trigger, Timer B2 underflow, Timer B2 interrupt generation frequency setting counter underflow or A-D trigger mode of Timer B.

**Table 15.1.6.1 Simultaneous Sample Sweep Mode Specifications**

| Item | Specification |
|---|---|
| Function | The SCAN1 to SCAN0 bits in the ADCON1 register and ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to the selected pins is converted one-by-one to a digital code. At this time, the input voltage of $AN_0$ and $AN_1$ are sampled simultaneously. |
| A-D Conversion Start Condition | When the TRG bit in the ADCON0 register is "0" (software trigger)<br>Set the ADST bit in the ADCON0 register to “1” (A-D conversion started)<br>When the TRG bit in the ADCON0 register is "1" (hardware trigger)<br>The trigger is selected by TRG1 and HPTRG0 bits (See **Table 15.1.6.2**)<br>The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to “1” (A-D conversion started)<br>Timer B0, B2 or Timer B2 interrupt generation frequency setting counter underflow after setting the ADST bit to “1” (A-D conversion started) |
| A-D Conversion Stop Condition | A-D conversion completed (If selecting software trigger, the ADST bit is automatically set to "0" ).<br>Set the ADST bit to "0" (A-D conversion halted) |
| Interrupt Generation Timing | A-D conversion completed |
| Analog Input Pin | Select from $AN_0$ to $AN_1$ (2 pins), $AN_0$ to $AN_3$ (4 pins),$AN_0$ to $AN_5$ (6 pins), or $AN_0$ to $AN_7$ (8 pins) (Note 1) |
| Readout of A-D conversion result | Readout one of the AN0 to AN7 registers that corresponds to the selected pin |

Note 1. $AN_{00}$ to $AN_{07}$ and $AN_{20}$ to $AN_{27}$ can be used in the same way as $AN_0$ to $AN_{07}$. However, all input pins need to belong to the same group.

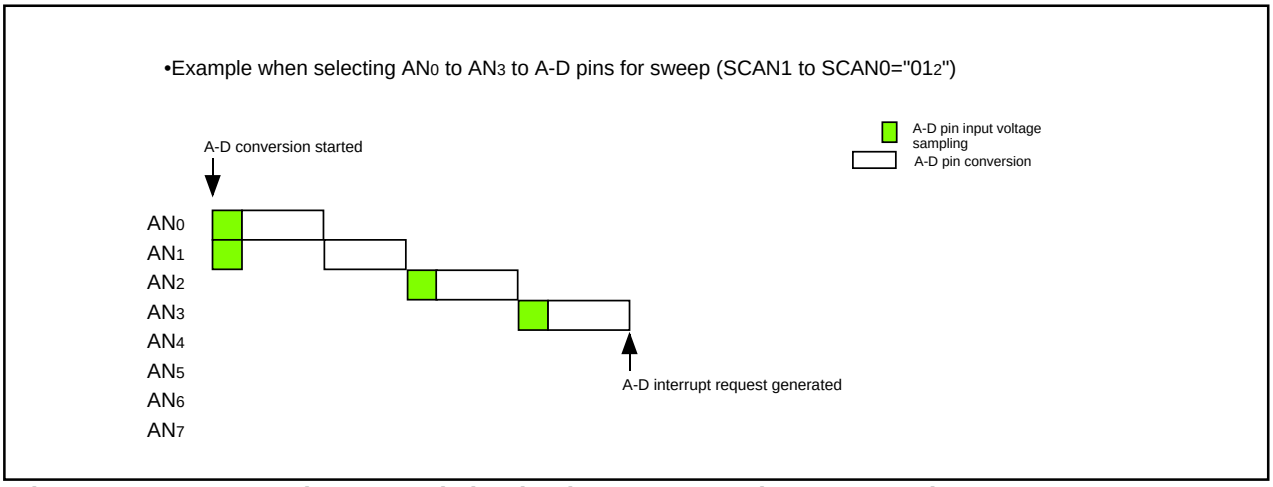
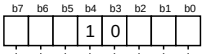



**Figure 15.1.6.1 Operation Example in Simultaneous Sample Sweep Mode**

A-D control register 0 (Note 1)

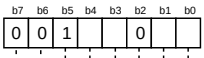



Symbol ADCON0 — Address 03D6$_{16}$ — After reset 00000XXX$_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| CH0 | Analog Input Pin Select Bit | Invalid in simultaneous sample sweep mode | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A-D Operation Mode Select Bit 0 | b4 b3<br>1 0 : Single sweep mode or simultaneous sample sweep mode | RW |
| MD1 | | | RW |
| TRG | Trigger Select Bit | Refer to **Table 15.1.6.2 Trigger Select Bit Setting in Simultaneous Sample Sweep Mode** | RW |
| ADST | A-D Conversion Start Fag | 0 : A-D conversion disabled<br>1 : A-D conversion started | RW |
| CKS0 | Frequency Select Bit 0 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |

Note 1: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.

A-D control register 1 (Note 1)




Symbol ADCON1 — Address 03D7$_{16}$ — After reset 00$_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SCAN0 | A-D Sweep Pin Select Bit (Note2) | When selecting simultaneous sample sweep mode<br>b1 b0<br>0 0 : AN$_0$ to AN$_1$ (2 pins)<br>0 1 : AN$_0$ to AN$_3$ (4 pins)<br>1 0 : AN$_0$ to AN$_5$ (6 pins)<br>1 1 : AN$_0$ to AN$_7$ (8 pins) | RW |
| SCAN1 | | | RW |
| MD2 | A-D Operation Mode Select Bit 1 | 0 : Any mode other than repeat sweep mode 1 | RW |
| BITS | 8/10-Bit Mode Select Bit | 0 : 8-bit mode<br>1 : 10-bit mode | RW |
| CKS1 | Frequency Select Bit 1 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| VCUT | V$_{REF}$ Connect Bit (Note 3) | 1 : V$_{REF}$ connected | RW |
| — (b7-b6) | Reserved Bit | Set to "0" | RW |

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: AN$_{00}$ to AN$_{07}$ and AN$_{20}$ to AN$_{27}$ can be used in the same way as AN$_0$ to AN$_7$. Use the ADGSEL1 to ADGSET0 bits in the ADCON2 register to select the desired pin.
Note 3: If the VCUT bit is reset from "0" (V$_{REF}$ unconnected) to "1" (V$_{REF}$ connected), wait for 1 µs or more before starting A-D conversion.

A-D control register 2 (Note 1)

Symbol ADCON2 — Address 03D4$_{16}$ — After reset 00$_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SMP | A-D Conversion Method Select Bit | Set to "1" in simultaneous sample sweep mode | RW |
| ADGSEL0 | A-D Input Group Select Bit | b2 b1<br>0 0 : Select port P10 group (AN$_i$)<br>0 1 : Do not set<br>1 0 : Select port P0 group (AN$_{0i}$)<br>1 1 : Select port P1/P9 group (AN$_{2i}$) | RW |
| ADGSEL1 | | | RW |
| — (b3) | Reserved Bit | Set to "0" | RW |
| CKS2 | Frequency Select Bit 2 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| TRG1 | Trigger select bit 1 | Refer to **Table 15.1.6.2 Trigger Select Bit Setting in Simultaneous Sample Sweep Mode** | RW |
| — (b7-b6) | Nothing is assigned. When write, set to "0". When read, its content is "0". | | — |

Note 1: If the ADCON2 register is rewritten during A-D conversion, the conversion result will be indeterminate.

**Figure 15.1.6.2 ADCON0 to ADCON2 Registers for Simultaneous Sample Sweep Mode**

A-D trigger control register (Note 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | 0 | | 0 | 1 |

| Symbol | Address | After reset |
|---|---|---|
| ADTRGCON | $03D2_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SSE | A-D Operation Mode Select Bit 2 | 1 : Simultaneous sample sweep mode or delayed trigger mode 0, 1 | RW |
| DTE | A-D Operation Mode Select Bit 3 | 0 : Any mode other than delayed trigger mode 0,1 | RW |
| HPTRG0 | AN0 Trigger Select Bit | Refer to **Table 15.1.6.2 Trigger Select Bit Setting in Simultaneous Sample Sweep Mode** | RW |
| HPTRG1 | AN1 Trigger Select Bit | Set to "0" in simultaneous sample sweep mode | RW |
| ($\overline{b7\text{-}b4}$) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: If ADTRGCON register is rewritten during A-D conversion, the conversion result will be indeterminate.

**Figure 15.1.6.3 ADTRGCON Register in Simultaneous Sample Sweep Mode**

| TRG | TRG1 | HPTRG0 | TRIGGER |
|---|---|---|---|
| 0 | - | - | Software trigger |
| 1 | - | 1 | Timer B0 underflow (Note 1) |
| 1 | 0 | 0 | $\overline{\text{ADTRG}}$ |
| 1 | 1 | 0 | Timer B2 or Timer B2 interrupt generation frequency setting counter underflow (Note 2) |

**Note 1. A count can be started for Timer B2, Timer B2 interrupt generation frequency setting counter underflow or the INT5 pin falling edge as count start conditions of Timer B0.**
**Note 2. Select Timer B2 or Timer B2 interrupt generation frequency setting counter using the TB2SEL bit in the TB2SC register.**

**Table 15.1.6.2 Trigger Select Bit Setting in Simultaneous Sample Sweep Mode**

### 15.1.7 Delayed Trigger Mode 0

In delayed trigger mode 0, analog voltage applied to the selected pins are converted one-by-one to a digital code. The delayed trigger mode 0 used in combination with A-D trigger mode of Timer B. The Timer B0 underflow starts a single sweep conversion. After completing the $AN_0$ pin conversion, the $AN_1$ pin is not sampled and converted until the Timer B1 underflow is generated. When the Timer B1 underflow is generated, the single sweep conversion is restarted after the $AN_1$ pin. Table 15.1.7.1 shows the delayed trigger mode 0 specifications. Figure 15.1.7.1 shows the operation example in delayed trigger mode 0. Figure 15.1.7.2 to Figure 15.1.7.3 show each flag operation in the ADSTAT0 register that corresponds to the operation example. Figure 15.1.7.4 shows the ADCON0 to ADCON2 registers in delayed trigger mode 0. Figure 15.1.7.5 shows the ADTRGCON register in delayed trigger mode 0 and Table 15.1.7.2 shows the trigger select bit setting in delayed trigger mode 0.

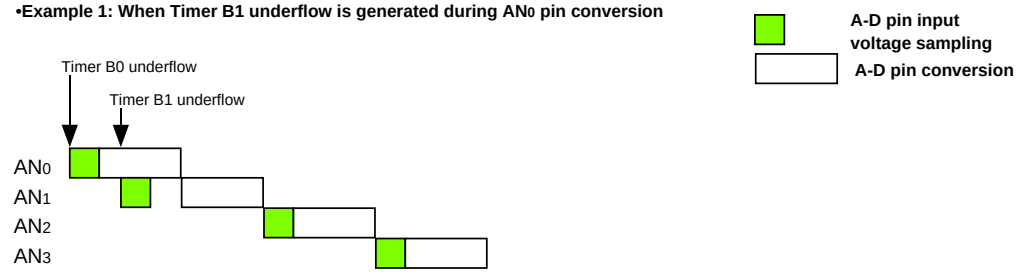
**Table 15.1.7.1 Delayed Trigger Mode 0 Specifications**

| Item | Specification |
|---|---|
| Function | The SCAN1 to SCAN0 bits in the ADCON1 register and ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to the input voltage of the selected pins are converted one-by-one to the digital code. At this time, Timer B0 under flow generation starts $AN_0$ pin conversion. Timer B1 underflow generation starts conversion after the $AN_1$ pin. (Note 1) |
| A-D Conversion Start | $AN_0$ pin conversion start condition<br>•After Timer B0 underflow is generated if Timer B0 underflow is generated again before Timer B1 underflow is generated , the conversion is not affected<br>•When Timer B0 underflow is generated during A-D conversion of pins after the $AN_1$ pin, conversion is halted and the sweep is restarted from the $AN_0$ pin again<br>$AN_1$ pin conversion start condition<br>•When Timer B1 underflow is generated during A-D conversion of the $AN_0$ pin, the input voltage of the $AN_1$ pin is sampled. The $AN_1$ conversion and the rest of the sweep start when $AN_0$ conversion is completed. |
| A-D Conversion Stop Condition | •When single sweep conversion from the $AN_0$ pin is completed<br>•Set the ADST bit to "0" (A-D conversion halted)(Note 2) |
| Interrupt Request Generation Timing | A-D conversion completed |
| Analog Input Pin | Select from $AN_0$ to $AN_1$ (2 pins), $AN_0$ to $AN_3$ (4 pins), $AN_0$ to $AN_5$ (6 pins) and $AN_0$ to $AN_7$ (8 pins)(Note 3) |
| Readout of A-D Conversion Result | Readout one of the AN0 to AN7 registers that corresponds to the selected pins |

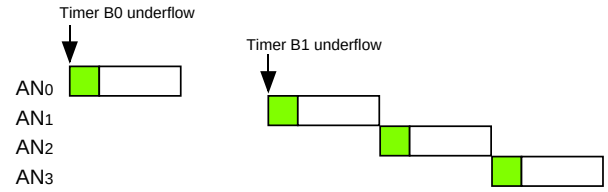
Note 1. Set the larger value than the value of the timer B0 register to the timer B1 register.
Note 2. Do not write “1” (A-D conversion started) to the ADST bit in delayed trigger mode 0. When write “1”, unexpected interrupts may be generated.
Note 3. $AN_{00}$ to $AN_{07}$ and $AN_{20}$ to $AN_{27}$ can be used in the same way as $AN_0$ to $AN_7$. However, all input pins need to belong to the same group.

•Example when selecting $AN_0$ to $AN_3$ to A-D sweep pins (SCAN1 to SCAN0="$01_2$")

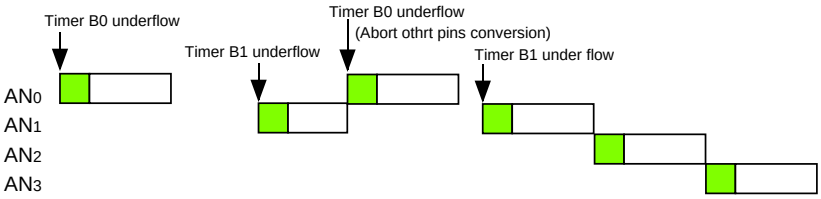
**•Example 1: When Timer B1 underflow is generated during $AN_0$ pin conversion**




**•Example 2: When Timer B1 underflow is generated after $AN_0$ pin conversion**




**•Example 3: When Timer B0 underflow is generated during A-D conversion of any pins except $AN_0$ pin**




**•Example 4: When Timer B0 underflow is generated again before Timer B1 underflow is generated after Timer B0 underflow generation**

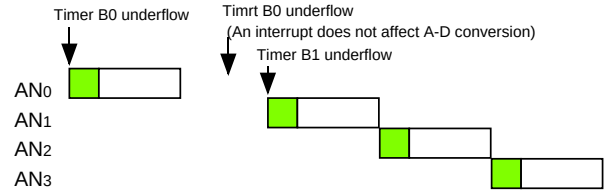
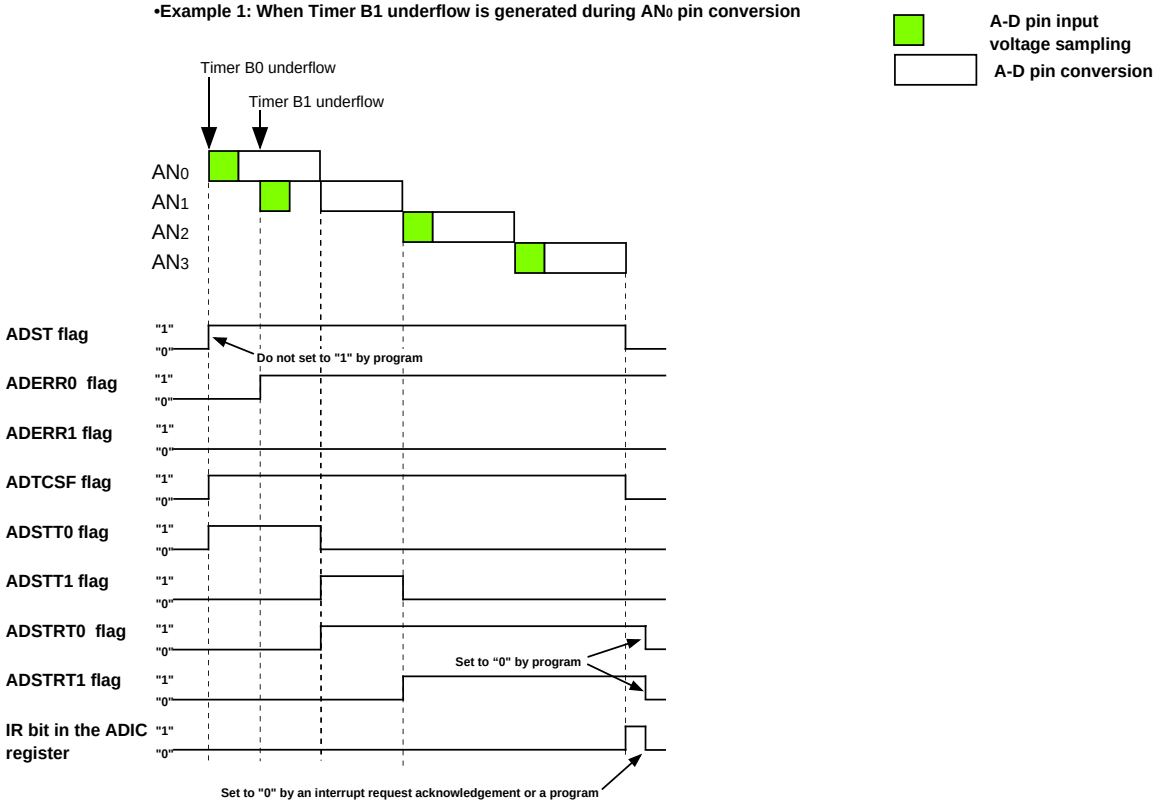


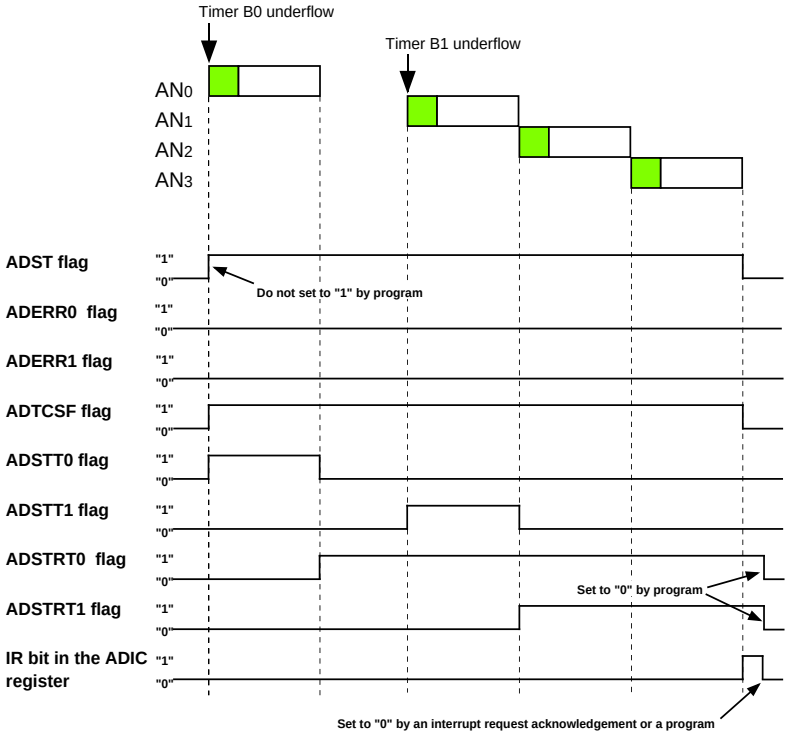

**Figure 15.1.7.1 Operation Example in Delayed Trigger Mode 0**

•Example when selecting $AN_0$ to $AN_3$ to A-D sweep pins (SCAN1 to SCAN0="$01_2$")

**•Example 1: When Timer B1 underflow is generated during $AN_0$ pin conversion**



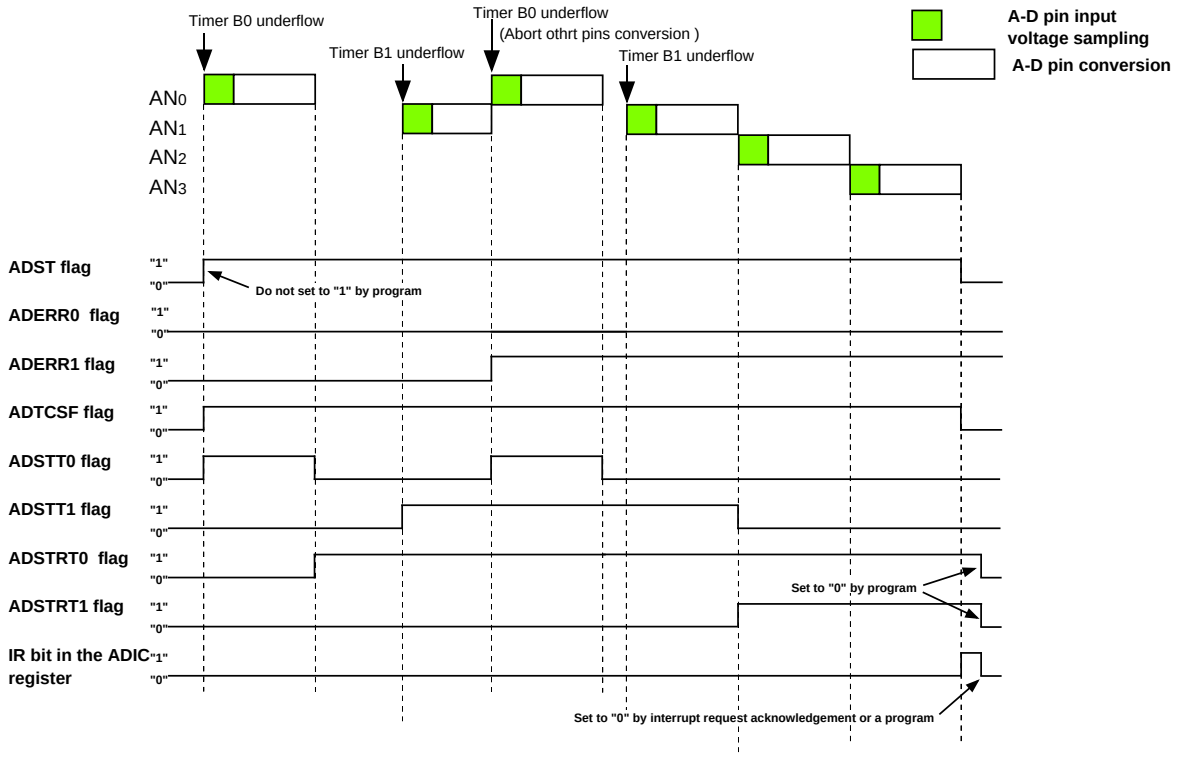

**•Example 2: When Timer B1 underflow is generated after $AN_0$ pin conversion**




**ADST flag: Bit 6 in the ADCON0 register**

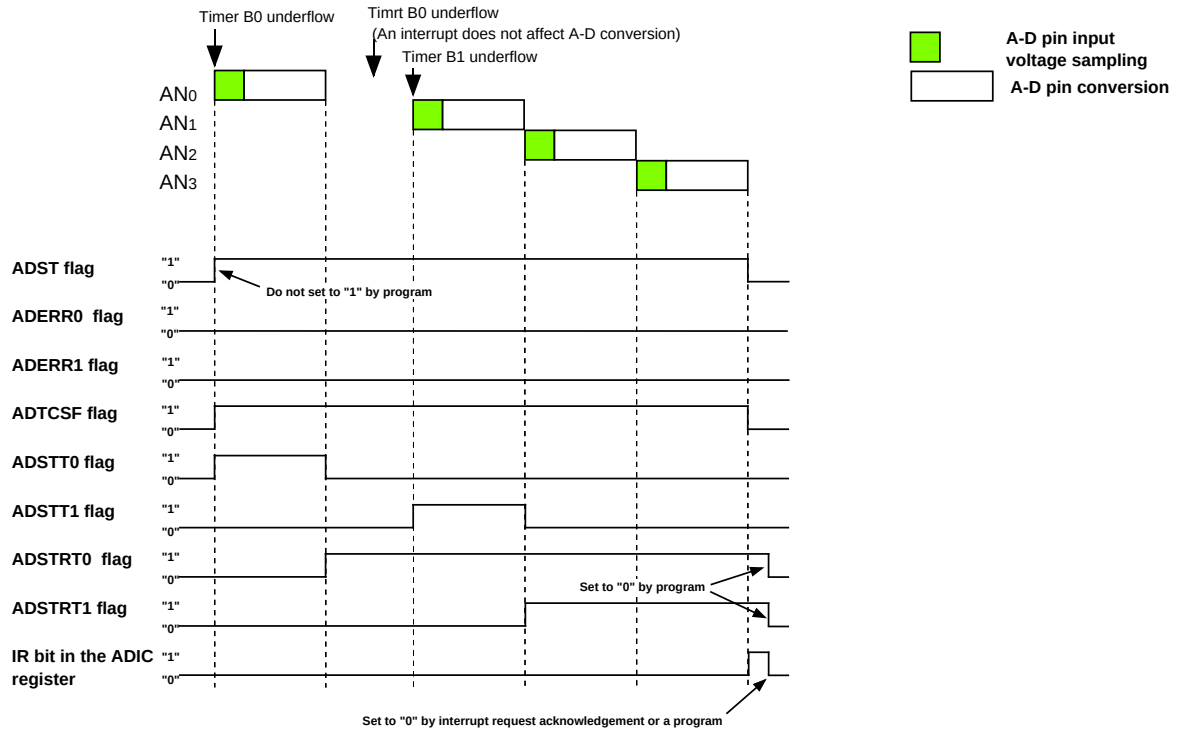
**Figure 15.1.7.2 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 0 (1)**

•Example 3: When Timer B0 underflow is generated during A-D pin conversion of any pins except AN0 pin




ADST flag: Bit 6 in the ADCON0 register
ADERR0, ADERR1, ADTCSF, ADSTT0, ADSTT1, ADSTRT0 and ADSTRT1 flag: bits 0, 1, 3, 4, 5, 6 and 7 in the ADSTAT0 register

•Example 4: After Timer B0 underflow is generated and when Timer B0 underflow is generated again before Timer B1 underflow is genetaed




ADST flag: Bit 6 in the ADCON0 register
ADERR0, ADERR1, ADTCSF, ADSTT0, ADSTT1, ADSTRT0 and ADSTRT1 flag: bits 0, 1, 3, 4, 5, 6 and 7 in the ADSTAT0 register

**Figure 15.1.7.3 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 0 (2)**

A-D control register 0 (Note 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | 0 | 0 | 1 | 1 | 1 |

Symbol: ADCON0 — Address: $03D6_{16}$ — After reset: $00000XXX_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| CH0 | Analog Input Pin Select Bit | b2 b1 b0<br>1 1 1 : Set to "111b" in delayed trigger mode 0 | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A-D Operation Mode Select Bit 0 | b4 b3<br>0 0 : One-shot mode or delayed trigger mode 0,1 | RW |
| MD1 | | | RW |
| TRG | Trigger Select Bit | Refer to **Table 15.1.7.2 Trigger Select Bit Setting in Delayed Trigger Mode 0** | RW |
| ADST | A-D Conversion Start Flag (Note 2) | 0 : A-D conversion disabled<br>1 : A-D conversion started | RW |
| CKS0 | Frequency Select Bit 0 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |

Note 1: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: Do not write "1" in delayed trigger mode 0. When write, set to "0".

A-D control register 1 (Note 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | | | 0 | | |

Symbol: ADCON1 — Address: $03D7_{16}$ — After reset: $00_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SCAN0 | A-D Sweep Pin Select Bit (Note2) | When selecting delayed trigger sweep mode 0<br>b1 b0<br>0 0: $AN_0$ to $AN_1$ (2 pins)<br>0 1: $AN_0$ to $AN_3$ (4 pins)<br>1 0: $AN_0$ to $AN_5$ (6 pins)<br>1 1: $AN_0$ to $AN_7$ (8 pins) | RW |
| SCAN1 | | | RW |
| MD2 | A-D Operation Mode Select Bit 1 | 0 : Any mode other than repeat sweep mode 1 | RW |
| BITS | 8/10-Bit Mode Select Bit | 0 : 8-bit mode<br>1 : 10-bit mode | RW |
| CKS1 | Frequency Select Bit 1 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| VCUT | $V_{REF}$ Connect Bit (Note 3) | 1 : $V_{REF}$ connected | RW |
| $\overline{\text{(b7-b6)}}$ | Reserved Bit | Set to "0" | RW |

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: $AN_{00}$ to $AN_{07}$ and $AN_{20}$ to $AN_{27}$ can be used in the same way as $AN_0$ to $AN_7$. Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
Note 3: If the VCUT bit is reset from "0" ($V_{REF}$ unconnected) to "1" ($V_{REF}$ connected), wait for 1 µs or more before starting A-D conversion.

A-D control register 2 (Note 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | X | | | 0 | | | 1 |

Symbol: ADCON2 — Address: $03D4_{16}$ — After reset: $00_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SMP | A-D Conversion Method Select Bit (Note 2) | 1 : With sample and hold | RW |
| ADGSEL0 | A-D Input Group Select Bit | b2 b1<br>0 0 : Select port P10 group ($AN_i$)<br>0 1 : Do not set<br>1 0 : Select port P0 group ($AN_{0i}$)<br>1 1 : Select port P1/P9 group ($AN_{2i}$) | RW |
| ADGSEL1 | | | RW |
| $\overline{\text{(b3)}}$ | Reserved Bit | Set to "0" | RW |
| CKS2 | Frequency Select Bit 2 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| TRG1 | Trigger Select Bit 1 | Refer to **Table 15.1.7.2 Trigger Select Bit Setting in Delayed Trigger Mode 0** | RW |
| $\overline{\text{(b7-b6)}}$ | Nothing is assigned. When write, set to "0". When read, its content is "0". | | — |

Note 1: If the ADCON2 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: Set to "1" in delayed trigger mode 0.

**Figure 15.1.7.4 ADCON0 to ADCON2 Registers in Delayed Trigger Mode 0**

A-D trigger control register (Note 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | 0 | | 1 | 1 |

| Symbol | Address | After reset |
|---|---|---|
| ADTRGCON | $03D2_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SSE | A-D Operation Mode Select Bit 2 | Simultaneous sample sweep mode or delayed trigger mode 0,1 | RW |
| DTE | A-D Operation Mode Select Bit 3 | Delayed trigger mode 0, 1 | RW |
| HPTRG0 | AN0 Trigger Select Bit | Refer to **Table 15.1.7.2 Trigger Select Bit Setting in Delayed Trigger Mode 0** | RW |
| HPTRG1 | AN1 Trigger Select Bit | Refer to **Table 15.1.7.2 Trigger Select Bit Setting in Delayed Trigger Mode 0** | RW |
| — (b7-b4) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: If ADTRGCON reigster is rewritten during A-D conversion, the conversion result will be indeterminate.

**Figure 15.1.7.5 ADTRGCON Register in Delayed Trigger Mode 0**

**Table 15.1.7.2 Trigger Select Bit Setting in Delayed Trigger Mode 0**

| TRG | TRG1 | HPTRG0 | HPTRG1 | Trigger |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | Timer B0, B1 underflow |

### 15.1.8 Delayed Trigger Mode 1

In delayed trigger mode 1, analog voltages applied to the selected pins are converted one-by-one to a digital code. When the input of the $\overline{\text{ADTRG}}$ pin (falling edge) changes state from “H” to “L”, a single sweep conversion is started. After completing the AN0 pin conversion, the AN1 pin is not sampled and converted until the second $\overline{\text{ADTRG}}$ pin falling edge is generated. When the second $\overline{\text{ADTRG}}$ falling edge is generated, The single sweep conversion of the pins after the AN1 pin is restarted. Table 15.1.8.1 shows the delayed trigger mode 1 specifications. Figure 15.1.8.1 shows the operation example of delayed trigger mode 1. Figure 15.1.8.2 to Figure 15.1.8.3 show each flag operation in the ADSTAT0 register that corresponds to the operation example. Figure 15.1.8.4 shows the ADCON0 to ADCON2 registers in delayed trigger mode 1. Figure 15.1.8.5 shows the ADTRGCON register in delayed trigger mode 1 and Table 15.1.8.2 shows the trigger select bit setting in delayed trigger mode 1.

**Table 15.1.8.1 Delayed Trigger Mode 1 Specifications**

| Item | Specification |
|---|---|
| Function | The SCAN1 to SCAN0 bits in the ADCON1 register and ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltages applied to the selected pins are converted one-by-one to a digital code. At this time, the $\overline{\text{ADTRG}}$ pin falling edge starts AN0 pin conversion and the second $\overline{\text{ADTRG}}$ pin falling edge starts conversion of the pins after AN1 pin |
| A-D Conversion Start Condition | AN0 pin conversion start condition<br>The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” (falling edge)(Note 1)<br>AN1 pin conversion start condition (Note 2)<br>The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” (falling edge)<br>•When the second $\overline{\text{ADTRG}}$ pin falling edge is generated during A-D conversion of the AN0 pin, input voltage of AN1 pin is sampled or after at the time of $\overline{\text{ADTRG}}$ falling edge. The conversion of AN1 and the rest of the sweep starts when AN0 conversion is completed.<br>•When the $\overline{\text{ADTRG}}$ pin falling edge is generated again during single sweep conver sion of pins after the AN1 pin, the conversion is not affected |
| A-D Conversion Stop Condition | •A-D conversion completed<br>•Set the ADST bit to "0" (A-D conversion halted)(Note 3) |
| Interrupt Request Generation Timing | Single sweep conversion completed |
| Analog Input Pin | Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins) and AN0 to AN7 (8 pins)(Note 4) |
| Readout of A-D Conversion Result | Readout one of the AN0 to AN7 registers that corresponds to the selected pins |

Note 1. When a third $\overline{\text{ADTRG}}$ pin falling edge is generated again during A-D conversion, its trigger is ignored.

Note 2. The $\overline{\text{ADTRG}}$ pin falling edge is detected synchronized with the operation clock ϕAD. Therefore, when the $\overline{\text{ADTRG}}$ pin falling edge is generated in shorter periods than ϕAD, the second $\overline{\text{ADTRG}}$ pin falling edge may not be detected. Do not generate the $\overline{\text{ADTRG}}$ pin falling edge in shorter periods than ϕAD.

Note 3. Do not write “1” (A-D conversion started) to the ADST bit in delayed trigger mode 1. When write “1”,unexpected interrupts may be generated.

Note 4. AN00 to AN07 and AN20 to AN27 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.

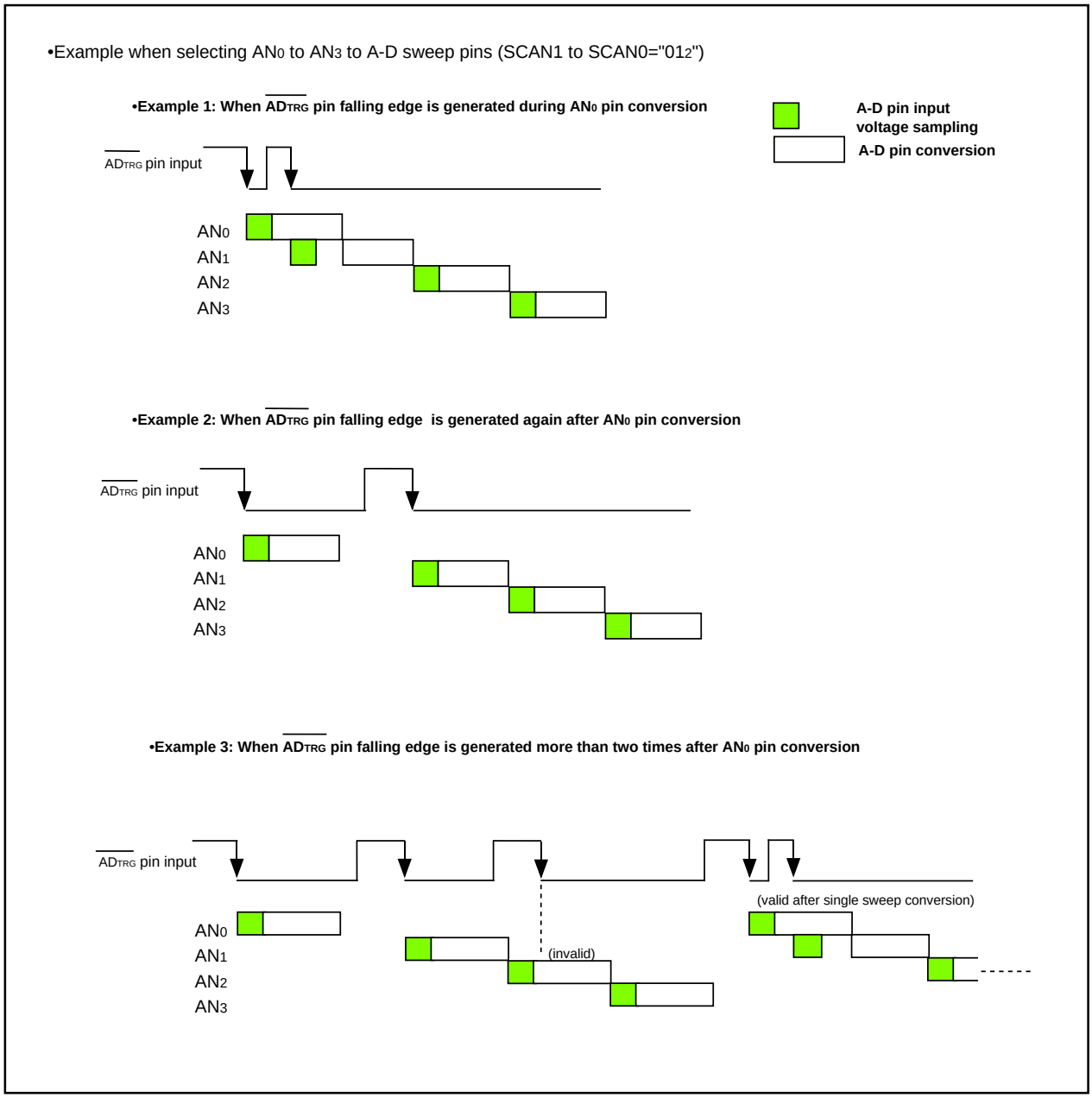
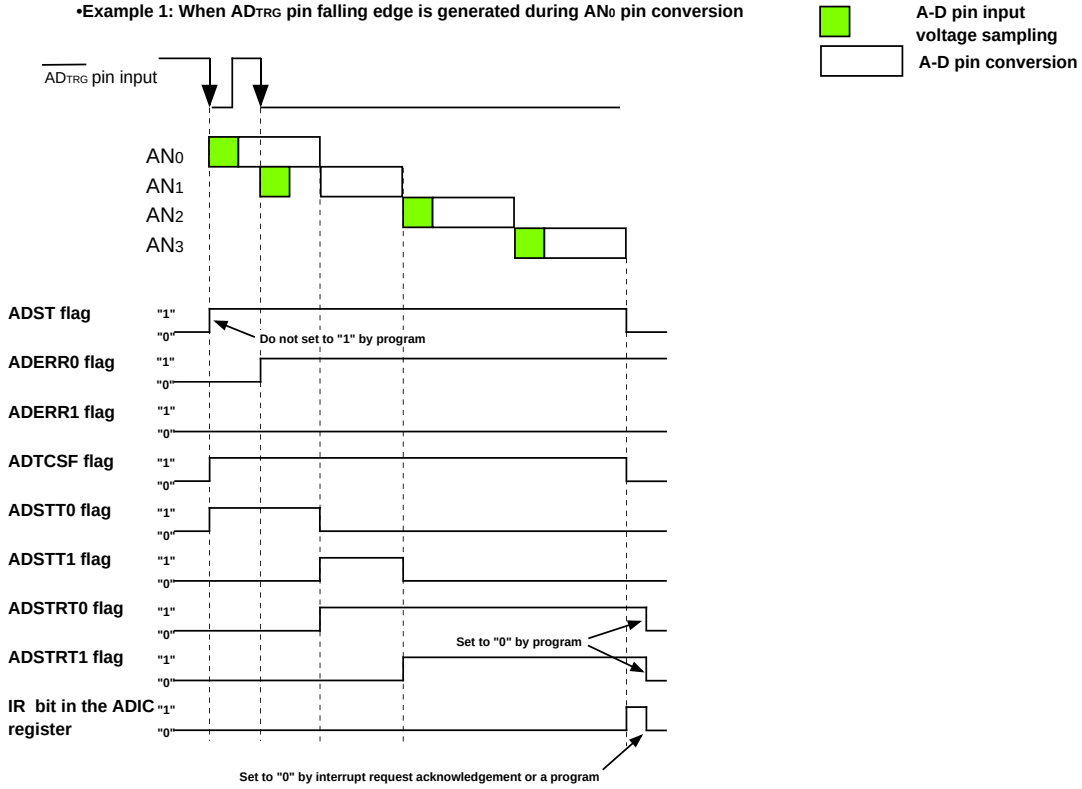


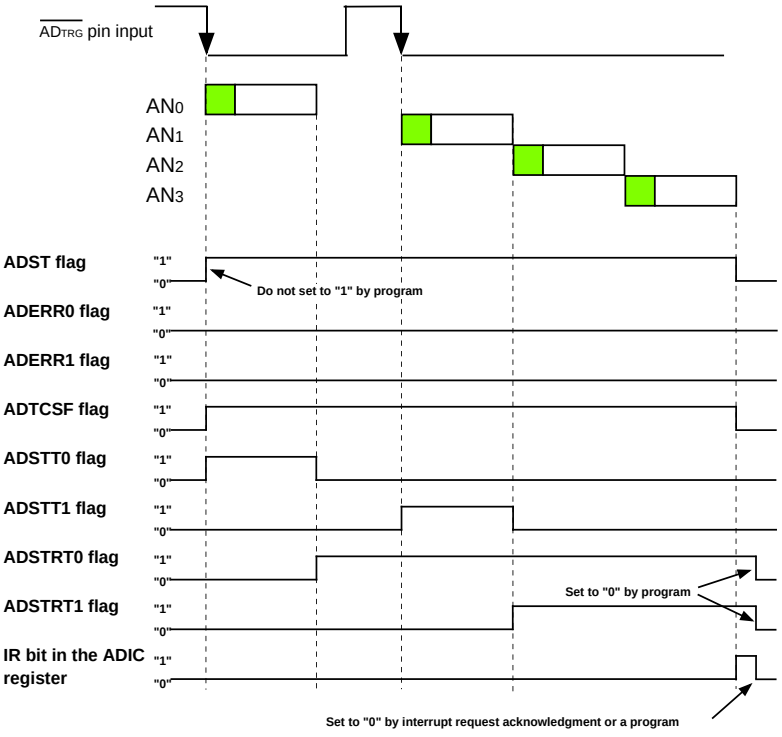

**Figure 15.1.8.1 Operation Example in Delayed Trigger Mode1**

•Example when selecting $AN_0$ to $AN_3$ to A-D sweep pins (SCAN1 to SCAN0="$01_2$")

**•Example 1: When $\overline{ADTRG}$ pin falling edge is generated during $AN_0$ pin conversion**




**•Example 2: When $\overline{ADTRG}$ pin falling edge is generated again after $AN_0$ pin conversion**




**ADST flag: Bit 6 in the ADCON0 register**
**ADERR0, ADERR1, ADTCSF, ADSTT0, ADSTT1, ADSTRT0 and ADSTRT1 flag: bits 0, 1, 3, 4, 5, 6 and 7 in the ADSTAT0 register**

**Figure 15.1.8.2 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 1 (1)**

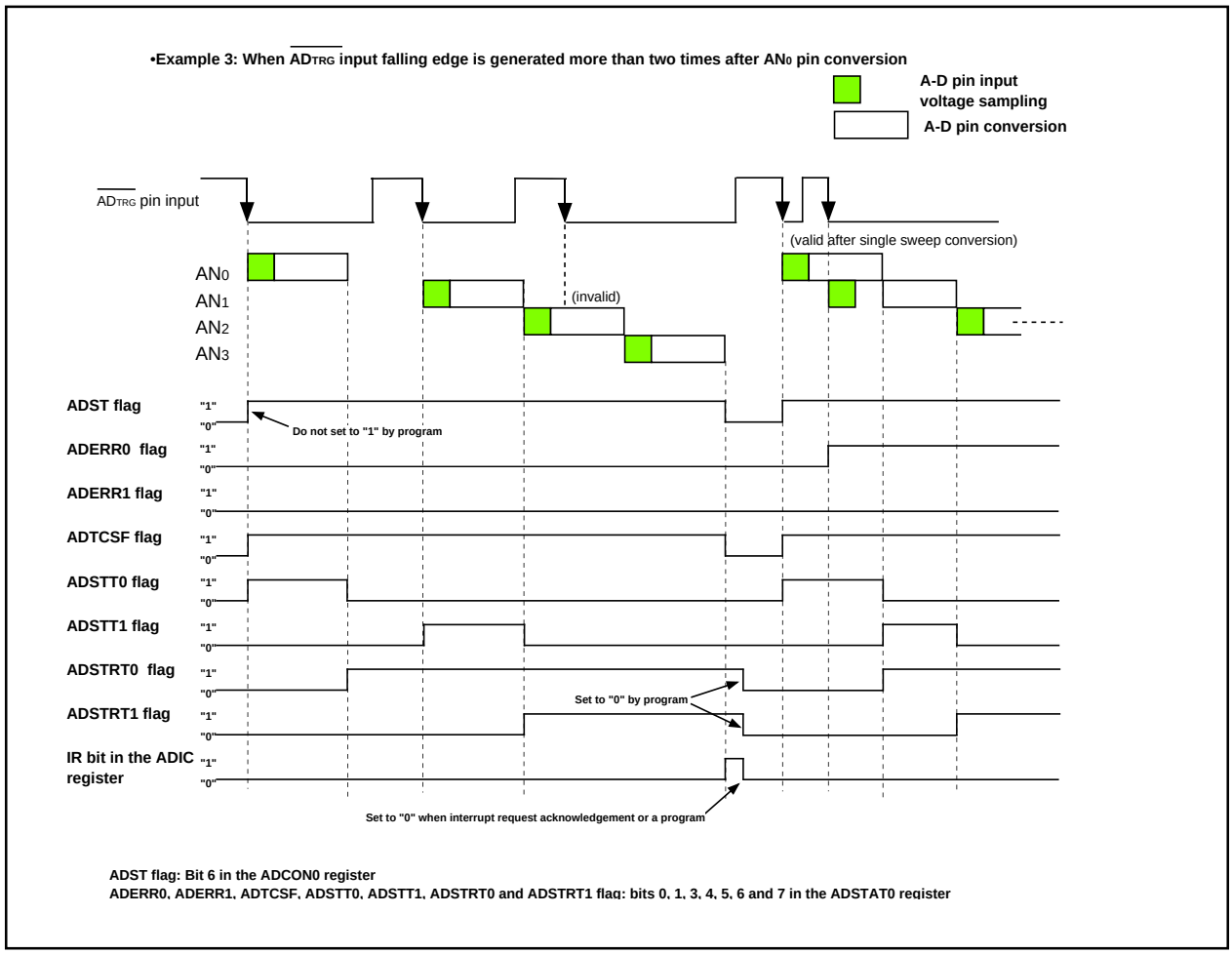



**Figure 15.1.8.2 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 1 (2)**

A-D control register 0 (Note 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | 0 | 0 | 1 | 1 | 1 |

Symbol ADCON0 — Address 03D6$_{16}$ — After reset 00000XXX$_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| CH0 | Analog Input Pin Select Bit | b2 b1 b0<br>1 1 1 : Set to "111b" in delayed trigger mode 1 | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A-D Operation Mode Select Bit 0 | b4 b3<br>0 0 : One-shot mode or delayed trigger mode 0,1 | RW |
| MD1 | | | RW |
| TRG | Trigger Select Bit | Refer to **Table 15.1.8.2 Trigger Select Bit Setting in Delayed Trigger Mode 1** | RW |
| ADST | A-D Conversion Start Flag (Note 2) | 0 : A-D conversion disabled<br>1 : A-D conversion started | RW |
| CKS0 | Frequency Select Bit 0 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |

Note 1: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: Do not write "1" in delayed trigger mode 1. When write, set to "0".

A-D control register 1 (Note 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | | | 0 | | |

Symbol ADCON1 — Address 03D7$_{16}$ — After reset 00$_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SCAN0 | A-D Sweep Pin Select Bit (Note 2) | When selecting delayed trigger mode 1<br>b1 b0<br>0 0: AN$_0$ to AN$_1$ (2 pins)<br>0 1: AN$_0$ to AN$_3$ (4 pins)<br>1 0: AN$_0$ to AN$_5$ (6 pins)<br>1 1: AN$_0$ to AN$_7$ (8 pins) | RW |
| SCAN1 | | | RW |
| MD2 | A-D Operation Mode Select Bit 1 | 0 : Any mode other than repeat sweep mode 1 | RW |
| BITS | 8/10-Bit Mode Select Bit | 0 : 8-bit mode<br>1 : 10-bit mode | RW |
| CKS1 | Frequency Select Bit 1 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| VCUT | V$_{REF}$ Connect Bit (Note 3) | 1 : V$_{REF}$ connected | RW |
| (b7-b6) | Reserved Bit | Set to "0" | RW |

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: AN$_{00}$ to AN$_{07}$ and AN$_{20}$ to AN$_{27}$ can be used in the same way as AN$_0$ to AN$_7$. Use the ADGSEL1 to ADGSET0 bits in the ADCON2 register to select the desired pin.
Note 3: If the VCUT bit is reset from "0" (V$_{REF}$ unconnected) to "1" (V$_{REF}$ connected), wait for 1 µs or more before starting A-D conversion.

A-D control register 2 (Note 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | X | | | 0 | | | 1 |

Symbol ADCON2 — Address 03D4$_{16}$ — After reset 00$_{16}$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SMP | A-D Conversion Method Select Bit (Note 2) | 1 : With sample and hold | RW |
| ADGSEL0 | A-D Input Group Select Bit | b2 b1<br>0 0 : Select port P10 group (AN$_i$)<br>0 1 : Do not set<br>1 0 : Select port P0 group (AN$_{0i}$)<br>1 1 : Select port P1/P9 group (AN$_{2i}$) | RW |
| ADGSEL1 | | | RW |
| (b3) | Reserved Bit | Set to "0" | RW |
| CKS2 | Frequency Select Bit 2 | Refer to **Table 15.2 A-D Conversion Frequency Select** | RW |
| TRG1 | Trigger Select Bit 1 | Refer to **Table 15.1.8.2 Trigger Select Bit Setting in Delayed Trigger Mode 1** | RW |
| (b7-b6) | Nothing is assigned. When write, set to "0". When read, its content is "0". | | — |

Note 1: If the ADCON2 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: Set to "1" in delayed trigger mode 1.

**Figure 15.1.8.4 ADCON0 to ADCON2 Registers in Delayed Trigger Mode 1**

A-D trigger control register (Note 1)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | 0 | | 1 | 1 |

| Symbol | Address | After reset |
|---|---|---|
| ADTRGCON | 03D2h | 00h |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| SSE | A-D Operation Mode Select Bit 2 | Simultaneous sample sweep mode or delayed trigger mode 0,1 | RW |
| DTE | A-D Operation Mode Select Bit 3 | Delayed trigger mode 0, 1 | RW |
| HPTRG0 | AN0 Trigger Select Bit | Refer to **Table 15.1.8.2 Trigger Select Bit Setting in Delayed Trigger Mode 1** | RW |
| HPTRG1 | AN1 Trigger Select Bit | Refer to **Table 15.1.8.2 Trigger Select Bit Setting in Delayed Trigger Mode 1** | RW |
| — (b7-b4) | Nothing is assigned. When write, set to “0”. When read, its content is “0”. | | — |

Note 1: If ADTRGCON is rewritten during A-D conversion, the conversion result will be indeterminate.

**Figure 15.1.8.5 ADTRGCON Register in Delayed Trigger Mode 1**

**Table 15.1.8.2 Trigger Select Bit Setting in Delayed Trigger Mode 1**

| TRG | TRG1 | HPTRG0 | HPTRG1 | Trigger |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | $\overline{\text{AD}_{\text{TRG}}}$ |

## 15.2 Resolution Select Function

The BITS bit in the ADCON1 register determines the resolution. When the BITS bit is set to “1” (10-bit precision), the A-D conversion result is stored into bits 0 to 9 in the ADi register (i=0 to 7). When the BITS bit is set to “0” (8-bit precision), the A-D conversion result is stored into bits 0 to 7 in the ADi register.

## 15.3 Sample and Hold

When the SMP bit in the ADCON 2 register is set to “1” (with the sample and hold function), A-D conversion rate per pin increases to 28 ϕAD cycles for 8-bit resolution or 33 ϕAD cycles for 10-bit resolution. The sample and hold function is available in one-shot mode, repeat mode, single sweep mode, repeat sweep mode 0 and repeat sweep mode 1. In these modes, start A-D conversion after selecting whether the sample and hold circuit is to be used or not.

## 15.4 Current Consumption Reducing Function

When the A-D converter is not used, the VCUT bit in the ADCON1 register isolates the resistor ladder of the A-D converter from the reference voltage input pin (VREF). Power consumption is reduced by shutting off any current flow into the resistor ladder from the VREF pin.

When using the A-D converter, set the VCUT bit to “1” (VREF connected) before setting the ADST bit in the ADCON0 register to “1” (A-D conversion started). Do not set the ADST bit and VCUT bit to “1” simultaneously, nor set the VCUT bit to “0” (VREF unconnected) during A-D conversion.

## 15.5 Analog Input Pin and External Sensor Equivalent Circuit Example

Figure 15.5.1 shows an example of the analog input pin and external sensor equivalent circuit.

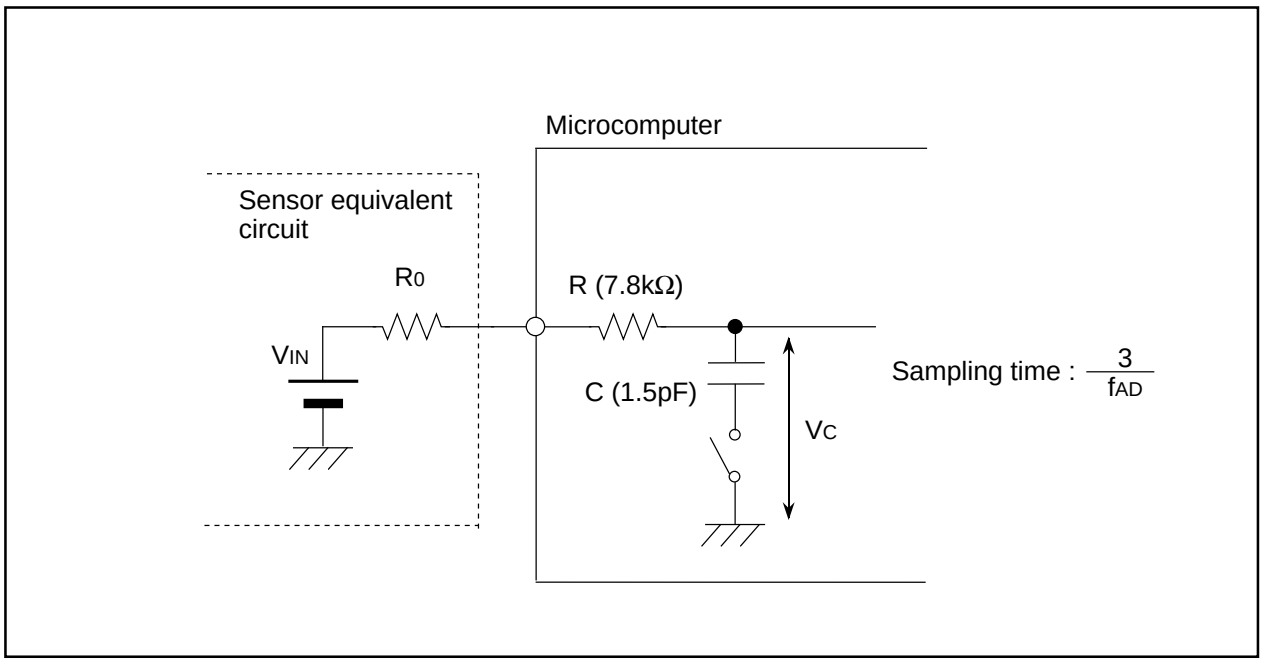



**Figure 15.5.1 Analog Input Pin and External Sensor Equivalent Circuit**

## 15.6 Precautions of Using A-D Converter

(1) Set the bit in the port direction register, which corresponds to the pin being used as the analog input, to “0” (input mode) Set the bit in the port direction register, which corresponds to pin $\overline{\text{ADTRG}}$, to “0” (input mode) if the external trigger is used.

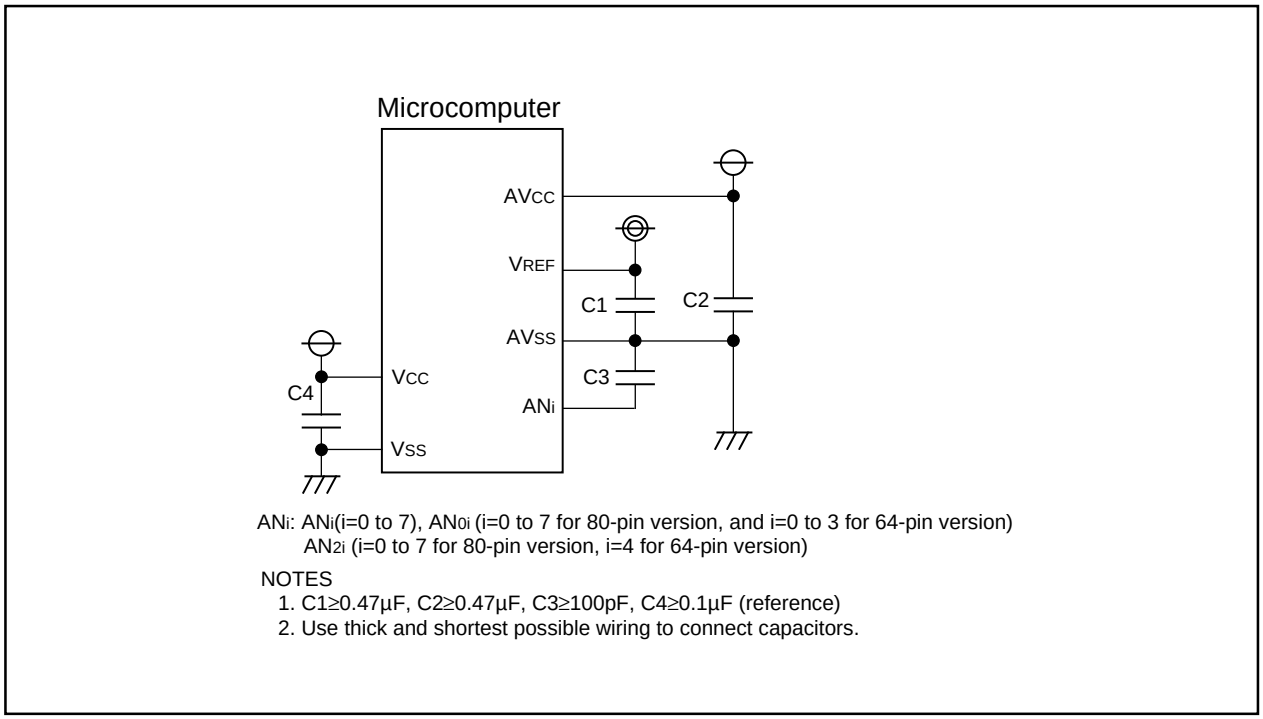
(2) When using a key input interrupt, do not use pins $AN_4$ to $AN_7$ as analog input pins (key input interrupt request is generated when the A-D input voltage is “L”).

(3) Insert capacitors between pins $AV_{CC}$, $V_{REF}$, analog input pin ($AN_i$ (i=0 to 7), $AN_{0i}$ and $AN_{2i}$) and $AV_{SS}$ to prevent latch-ups and malfunctions due to noise, and to minimize conversion errors. The same applies to pins $V_{CC}$ and $V_{SS}$. Figure 15.6.1 shows the procedure of each pin.

(4) Incorrect values are stored in the ADi register (i=0 to 7) if the CPU reads the ADi register while the ADi register is storing results from a completed A-D conversion. This occurs when a divided main clock or a sub clock is selected as the CPU clock.

- In one-shot mode or single sweep mode, simultaneous sample sweep mode and delayed trigger mode 0, 1 , read the corresponding ADi register after verifying that the A-D conversion has been completed. (The completion of the A-D conversion can be determined by the IR bit in the ADIC register).
- In repeat mode, repeat sweep mode 0 and repeat sweep mode 1, use an undivided main clock as the CPU clock.

(5) Conversion results of the A-D converter are indeterminate, if the ADST bit in the ADCON0 register is set to “0” (A-D conversion halted) and the conversion is forcibly terminated, by program during A-D conversion. ADi registers not operating A-D conversion may also be indeterminate. If the ADST bit is changed to “0” by program, during the A-D conversion, do not use any values obtained from the ADi registers.




$AN_i$: $AN_i$(i=0 to 7), $AN_{0i}$ (i=0 to 7 for 80-pin version, and i=0 to 3 for 64-pin version)
$AN_{2i}$ (i=0 to 7 for 80-pin version, i=4 for 64-pin version)

NOTES
1. C1≥0.47µF, C2≥0.47µF, C3≥100pF, C4≥0.1µF (reference)
2. Use thick and shortest possible wiring to connect capacitors.

**Figure 15.6.1 VCC, VSS, AVCC, AVSS, VREF and ANi Connections**

# 16. Multi-master I2C bus Interface

The multi-master I2C bus interface is a serial communication circuit based on Philips I2C bus data transfer format. 2 independent channels, with both arbitration lost detection and synchronous functions, are built in for the multi-master serial communication. Figure 16.1 shows a block diagram of the multi-master I2C bus interface and Table 16.1 lists the multi-master I2C bus interface functions.

The multi-master I2C bus interface consists of the I2C0 address register, the I2C0 data shift register, the I2C0 clock control register, the I2C0 control register 1, I2C0 control register 2, the I2C0 status register, the I2C0 start/stop condition control register and other control circuits.

Figure 16.2 to 16.8 show the registers associated with the multi-master I2C bus.

**Table 16.1 Multi-master I2C bus interface functions**

| Item | Function |
|---|---|
| Format | Based on Philips I2C bus standard:<br>7-bit addressing format<br>High-speed clock mode<br>Standard clock mode |
| Communication mode | Based on Philips I2C bus standard:<br>Master transmit<br>Master receive<br>Slave transmit<br>Slave receive |
| SCL clock frequency | 16.1kHz to 400kHz (at VIIC (Note 1)= 4MHz) |

Note 1. VIIC=I2C system clock

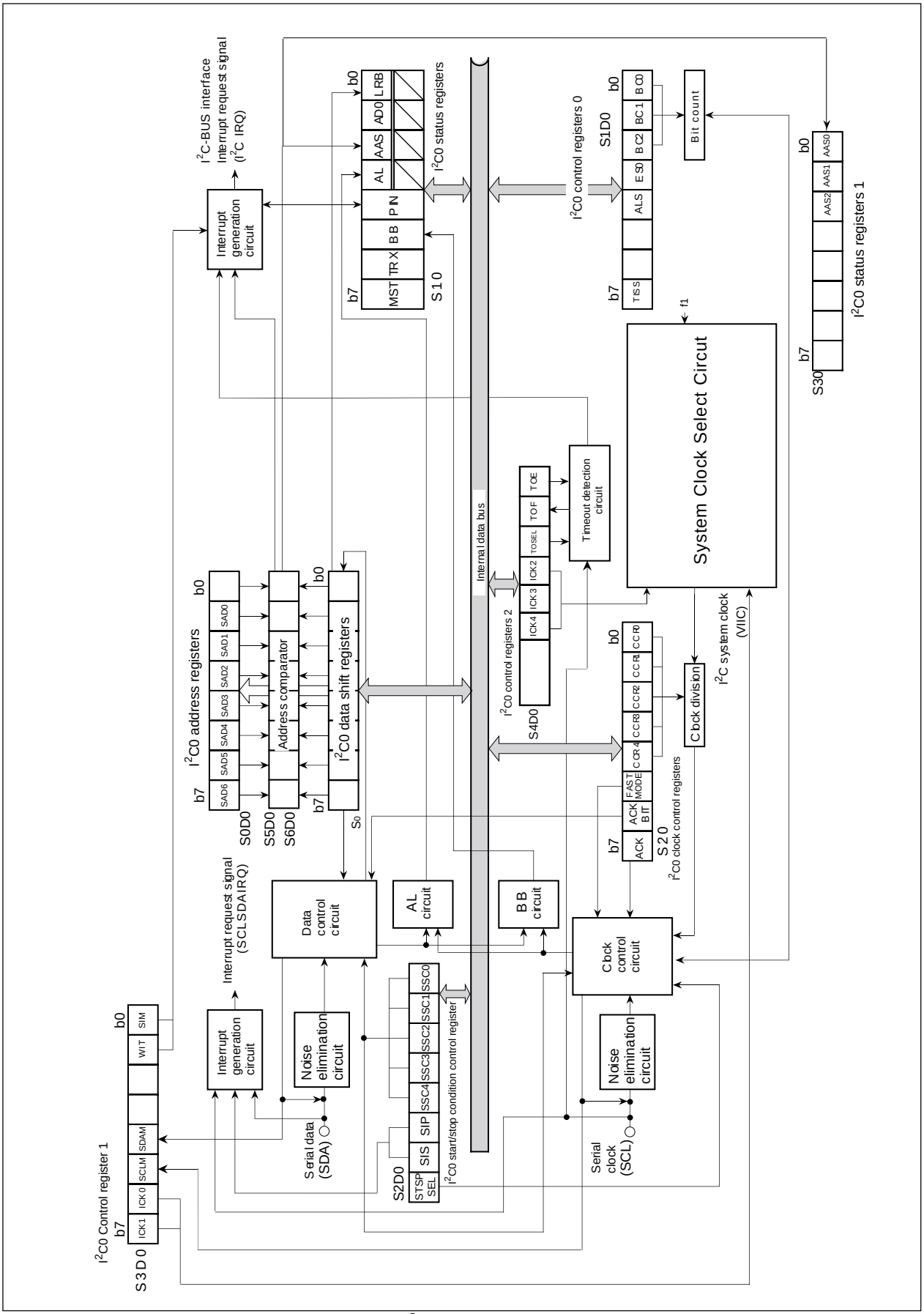
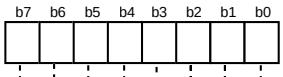



**Figure 16.1 Block diagram of multi-master I²C bus interface**

I2C0 address register

| Symbol | Address | After reset |
|---|---|---|
| S0D0 | 02E216 | 0016 |




| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| — | Reserved bit | Set to “0” | RW |
| SAD0 | Slave address | Comparing with received address data | RW |
| SAD1 | | | RW |
| SAD2 | | | RW |
| SAD3 | | | RW |
| SAD4 | | | RW |
| SAD5 | | | RW |
| SAD6 | | | RW |

**Figure 16.2 I2C0 address register, I2C0 address register 2**

## I²C0 data shift register

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

| Symbol | Address | When reset |
|---|---|---|
| S00 | 02E0₁₆ | XX₁₆ |

| Function | RW |
|---|---|
| Transmit/receive data are stored.<br>In the master transmit mode, the start condition/stop condition are triggered by writing data to the register (refer to **Section 16.9 START Condition Generation Method and Section16.11 STOP Condition Generation Method)**. The transmit/receive are started synchronized with SCL. | RW (Note 1) |

Note 1: The write is only enabled when the bus interface enable bit (ES0 bit) is "1". Because the register is used both for storing transmit/receive data, Write the transmit data after the receive data is read out when transmitting.

## I²C0 clock control register

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

| Symbol | Address | After reset |
|---|---|---|
| S20 | 02E4₁₆ | 00₁₆ |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| CCR0 | SCL frequency control bits | See **table 16.3 Set values of I²C0 clock control register and SCL frequency** | RW |
| CCR1 | | | RW |
| CCR2 | | | RW |
| CCR3 | | | RW |
| CCR4 | | | RW |
| FAST MODE | SCL mode specification bit | 0: Standard clock mode<br>1: High-speed clock mode | RW |
| ACK BIT | ACK bit | 0: ACK is returned<br>1: ACK is not returned | RW |
| ACK | ACK clock bit | 0: No ACK clock<br>1: With ACK clock | RW |

**Figure 16.3 I²C0 data shift register, I²C0 clock control register**

I²C0 control register 0

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

| Symbol | Address | After reset |
|---|---|---|
| S1D0 | 02E3₁₆ | 00₁₆ |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| BC0 | Bit counter (Number of transmit/receive bits) (Note 1) | b2 b1 b0<br>0 0 0 : 8<br>0 0 1 : 7<br>0 1 0 : 6<br>0 1 1 : 5<br>1 0 0 : 4<br>1 0 1 : 3<br>1 1 0 : 2<br>1 1 1 : 1 | RW |
| BC1 | | | RW |
| BC2 | | | RW |
| ES0 | I²C bus interface enable bit | 0: Disabled<br>1: Enabled | RW |
| ALS | Data format select bit | 0: Addressing format<br>1: Free data format | RW |
| $\overline{(b5)}$ | Reserved bit | Set to "0" | RW |
| IHR | I²C bus interface reset bit | 0: Reset release (auto)<br>1: Reset | RW |
| TISS | I²C bus interface pin input level select bit | 0: I²C bus input<br>1: SMBUS input | RW |

Note 1: In the following status, the bit counter is cleared automatically
- Start condition/stop condition are detected
- Immediately after the completion of 1-byte data transmit
- Immediately after the completion of 1-byte data receive

**Figure 16.4 I²C0 control register 0**

I²C0 status register

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

| Symbol | Address | After reset |
|---|---|---|
| S10 | $02E8_{16}$ | $0001000X_2$ |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| LRB | Last receive bit | 0: Last bit = 0<br>1: Last bit = 1 | RO (Note 1) |
| ADR0 | General call detecting flag | 0: No general call detected<br>1: General call detected | RO (Note 1) |
| AAS | Slave address comparison flag | 0: No address matched<br>1: Address matched | RO (Note 1) |
| AL | Arbitration lost detection flag | 0: Not detected<br>1: Detected | RO (Note 2) |
| PIN | I²C bus interface interrupt request bit | 0: Interrupt request issued<br>1: No interrupt request issued | RO (Note 2) |
| BB | Bus busy flag | 0: Bus free<br>1: Bus busy | RO (Note 1) |
| TRX | Communication mode specification bits | b7 b6<br>0 0: Slave receive mode<br>0 1: Slave transmit mode<br>1 0: Master receive mode<br>1 1: Master transmit mode | RO (Note 3) |
| MST | | | RO (Note 3) |

Note 1: This bit is read only if it is used for the status check.
To write to this bit, refer to **Sections 16.9 START Condition Generation Method and 16.11 STOP Condition Generation Method**.
Note 2: This bit is read only, When write, set to “0”.
Note 3: To write to these bits, refer to **Sections 16.9 START Condition Generation Method and 16.11 STOP Condition Generation Method.**

**Figure 16.5 I²C0 status register**

## I²C0 control register 1

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

| Symbol | Address | When reset |
|---|---|---|
| S3D0 | 02E6₁₆ | 00110000₂ |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| SIM | The interrupt enable bit for STOP condition detection | 0: Disable the I²C bus interface interrupt of STOP condition detection<br>1: Enable the I²C bus interface interrupt of STOP condition detection | RW |
| WIT | The interrupt enable bit for data receive completion | 0: Disable the I²C bus interface interrupt of data receive completion<br>1: Enable the I²C bus interface interrupt of data receive completion<br><br>When setting NACK (ACK bit = 0), write "0" | RW |
| PED | SDAi/Port function switch bit | 0: SDA I/O pin (enable ES0 = 1)<br>1: Port output pin (enable ES0 = 1) | RW |
| PEC | SCLi/Port function switch bit | 0: SCL I/O pin (enable ES0 = 1)<br>1: Port output pin (enable ES0 = 1) | RW |
| SDAM | The logic value monitor bit of SDA output | 0: SDA output logic value = 0<br>1: SDA output logic value = 1 | RO |
| SCLM | The logic value monitor bit of SCL output | 0: SCL output logic value = 0<br>1: SCL output logic value = 1 | RO |
| ICK0 | I²C system clock selection bits, if ICK4 to ICK2 bits in the S4D0 register is "000₂" | b7 b6<br>0 0 : VIIC =1/2 f1<br>0 1 : VIIC =1/4f1 ✻ f1=f(XIN)<br>1 0 : VIIC =1/8f1<br>1 1 : Reserved | RW |
| ICK1 | | | RW |

**Figure 16.6 I²C0 control register 1**

I2C0 control register 2

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

| Symbol | Address | When reset |
|---|---|---|
| S4D0 | 02E716 | 0016 |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TOE | Time out detection function enable bit | 0 : Disabled<br>1 : Enabled | RW |
| TOF | Time out detection flag | 0 : Not detected<br>1 : Detected | RO |
| TOSEL | Time out detection time select bit | 0 : Long time<br>1 : Short time | RW |
| ICK2 | I2C system clock select bits | b5 b4 b3<br>0 0 0 VIIC set by ICK1 and ICK0 bits in S3D0 register<br>0 0 1 VIIC = 1/2.5 f1<br>0 1 0 VIIC = 1/3 f1<br>0 1 1 VIIC = 1/5 f1<br>1 0 0 VIIC = 1/6 f1 | RW |
| ICK3 | | | RW |
| ICK4 | | | RW |
| $\overline{\text{(b6)}}$ | Reserved bit | Set to "0" | RW |
| SCPIN | STOP condition detection interrupt request bit | 0 : No I2C bus interface interrupt request<br>1 : I2C bus interface interrupt request | RW |

**Figure 16.7 I2C0 control register 2**

I²C0 start/stop condition control register

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

Symbol: S2D0 | Address: 02E5₁₆ | When reset: 00011010₂

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| SSC0 | START/STOP condition setting bits(Note 1) | Setting for detection condition of START/STOP condition. See **Table 16.2 Recommended setting value (SSC4 - SSC0) start/stop condition at each oscillation frequency**. | RW |
| SSC1 | | | RW |
| SSC2 | | | RW |
| SSC3 | | | RW |
| SSC4 | | | RW |
| SIP | SCL/SDA interrupt pin polarity select bit | 0: Active in falling edge<br>1: Active in rising edge | RW |
| SIS | SCL/SDA interrupt pin select bit | 0: SDA enabled<br>1: SCL enabled | RW |
| STSP SEL | START/STOP condition generation select bit | 0: Setup/hold time short mode<br>1: Setup/hold time long mode | RW |

Note 1: Disable the setting of "00000₂" and odd values.

**Figure 16.8 I²C0 start/stop condition control register**

**Table 16.2 Recommended setting value (SSC4 - SSC0) start/stop condition at each oscillation frequency**

| Oscillation f1 (MHz) | I²C Bus system clock select | I²C Bus system clock(MHz) | SSC4-SSC0 | SCL release time(cycle) | Setup time (cycle) | Hold time (cycle) |
|---|---|---|---|---|---|---|
| 10 | 1 / 2f1 | 5 | XXX11110 | 6.2 µs (31) | 3.2 µs (16) | 3.0 µs (15) |
| 8 | 1 / 2f1 | 4 | XXX11010 | 6.75 µs(27) | 3.5 µs (14) | 3.25 µs(13) |
| | | | XXX11000 | 6.25 µs(25) | 3.25 µs (13) | 3.0 µs (12) |
| 8 | 1 / 8f1 | 1 | XXX00100 | 5.0 µs (5) | 3.0 µs (3) | 2.0 µs (2) |
| 4 | 1 / 2f1 | 2 | XXX01100 | 6.5 µs (13) | 3.5 µs (7) | 3.0 µs (6) |
| | | | XXX01010 | 5.5 µs (11) | 3.0 µs (6) | 2.5 µs (5) |
| 2 | 1 / 2f1 | 1 | XXX00100 | 5.0 µs (5) | 3.0 µs (3) | 2.0 µs (2) |

Note: Do not set odd values or “00000₂” to START/STOP condition setting bits(SSC4 to SSC0)

## 16.1 I2C0 Data Shift Register (S00 register)

The I2C0 data shift register (address 02E016) is the 8-bit shift register to store the receive data and the write transmit data. When the transmit data is written into this register, it is transferred to the outside from bit 7 in synchronization with the SCL clock, and each time one-bit data is output, the data of this register is shifted by one bit to the left. When the data is received, it is input to this register from the bit 0 in synchronization with the SCL clock, and each time the one-bit data is input, the data of this register is shifted by one bit to the left. Figure 16.9 shows the timing which stores the receive data to this register. The I2C0 data shift register is in a write enable status only when the I2C bus interface enable bit (ES0 bit : bit 3 of address 02E316) of the I2C0 control register 0 is "1". The bit counter is reset by a write instruction to the I2C0 data shift register. When both the ES0 bit and the MST bit in the I2C0 status register (address 02E816) are "1", the SCL is output by a write instruction to the I2C0 data shift register. Reading data from the I2C0 data shift register is always enabled regardless of the ES0 bit value.

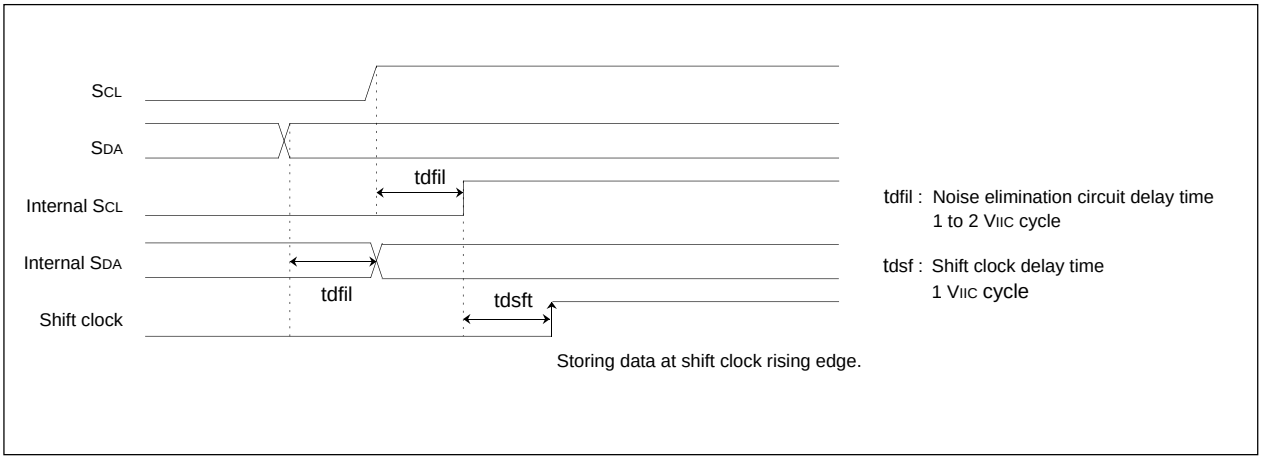



Figure 16.9 The timing of receiving data stored to I2C0 data shift register

## 16.2 I2C0 Address Register (S0D0 register)

This register consists of 7 bits of SAD6 to SAD0. At the addressing format which detects the slave address automatically, the contents of SAD6 to SAD0 are compared with the address data to be received.

## 16.3 I2C0 Clock Control Register (S20 register)

The I2C0 clock control register (address 02E416) is used to set theACK control, SCL mode and the SCL frequency.

### 16.3.1 Bits 0 to 4: SCL frequency control bits (CCR0–CCR4)

These bits control the SCL frequency. See **Table 16.3 Set values of I2C0 clock control register and SCL frequency**.

### 16.3.2 Bit 5: SCL mode specification bit (FAST MODE)

This bit specifies SCL mode. When this bit is set to “0”, Standard clock mode is selected. When the bit is set to “1” , high-speed clock mode is selected. When connecting to the bus with high-speed mode I2C bus standard (maximum 400 kbits/s), set 4 MHz or more to the I2C system clock(VIIC).

### 16.3.3 Bit 6: ACK bit (ACK BIT)

This bit sets the SDA status when an ACK clock(Note 1) is generated. When this bit is set to “0”, ACK return mode is selected and the SDA goes to “L” at the ACK clock generation. When the bit is set to “1”, ACK non-return mode is selected. The SDA is held in the “H” status at the ACK clock generation. However, when the address data is received at the ACK BIT=0 and the slave address matches with the address data, the SDA is automatically set to “L” (ACK is returned). If the slave address does not match with the address data, the SDA is automatically set to “H” (ACK is not returned).

Note 1. ACK clock: Clock for acknowledgment

### 16.3.4 Bit 7: ACK clock bit (ACK)

This bit specifies mode of acknowledgment for responses to transfer data. When this bit is set to “0”, no ACK clock mode is selected. In this case, the ACK clock is not generated after the data transmit. When the bit is set to “1”, ACK clock mode is selected and the master generates an ACK clock at the completion of each 1-byte data transfer. The device for transmitting the address data and the control data releases the SDA at the ACK clock generation (set the SDA to “H”) and receives the ACK bit generated by the data receive device.

**Note .** Do not rewrite the data into the I2C0 clock control register other than the ACK bit (ACKBIT) during the transfer. If data is written during the transfer, the I2C Bus clock circuit is reset and the data can not be transferred normally.

**Table 16.3 Set values of I2C0 clock control register and SCL frequency**

| Setting value of CCR4 to CCR0 | | | | | SCL frequency (at VIIC=4MHz, unit : kHz) (Note 1) | |
|---|---|---|---|---|---|---|
| CCR4 | CCR3 | CCR2 | CCR1 | CCR0 | Standard clock mode | High-speed clock mode |
| 0 | 0 | 0 | 0 | 0 | Setting disabled | Setting disabled |
| 0 | 0 | 0 | 0 | 1 | Setting disabled | Setting disabled |
| 0 | 0 | 0 | 1 | 0 | Setting disabled | Setting disabled |
| 0 | 0 | 0 | 1 | 1 | - (Note 2) | 333 |
| 0 | 0 | 1 | 0 | 0 | - (Note 2) | 250 |
| 0 | 0 | 1 | 0 | 1 | 100 | 400 (Note 3) |
| 0 | 0 | 1 | 1 | 0 | 83.3 | 166 |
| ↓ | ↓ | ↓ | ↓ | ↓ | 500 / CCR value (Note 3) | 1000 / CCR value (Note 3) |
| 1 | 1 | 1 | 0 | 1 | 17.2 | 34.5 |
| 1 | 1 | 1 | 1 | 0 | 16.6 | 33.3 |
| 1 | 1 | 1 | 1 | 1 | 16.1 | 32.3 |

**Note 1:** The duty of the SCL clock output is 50 %. The duty becomes 35 to 45 % only when high-speed clock mode is selected and the CCR value = 5 (400 kHz, at VIIC = 4 MHz). “H” duration of the clock fluctuates from –4 to +2 I2C system clock cycles in standard clock mode, and fluctuates from –2 to +2 I2C system clock cycles in high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because the “L” is extended instead of “H” reduc tion. These are the values when the SCL clock synchronization by the synchronous function is not performed. The CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.

**Note 2:** Each value of the SCL frequency exceeds the limit at VIIC = 4 MHz or more. When using these setting values, use VIIC = 4 MHz or less. Refer to **Figure 16.6 I2C system clock select bits (bit 6 and 7 of I2C control register 1) on VIIC**.

**Note 3:** The data formula of SCL frequency is described below:
VIIC/(8 × CCR value) Standard clock mode
VIIC/(4 × CCR value) High-speed clock mode (CCR value ≠ 5)
VIIC/(2 × CCR value) High-speed clock mode (CCR value = 5)
Do not set 0 to 2 as the CCR value regardless of the VIIC frequency.
Set 100 kHz (max.) in standard clock mode and 400 kHz (max.) in high-speed clock mode to the SCL frequency by setting the SCL frequency control bits CCR4 to CCR0.

## 16.4 I2C0 Control Register 0 (S1D0 register)

The I2C0 control register 0 (address 02E316) controls the data communication format.

### 16.4.1 Bits 0 to 2: Bit counter (BC0–BC2)

These bits decide the number of bits for the next 1-byte data to be transmitted. The I2C Bus interface interrupt request signal is generated immediately after the number of count specified with these bits (the ACK clock is added to the number of count when the ACK clock is selected by the ACK bit (bit 7 of address 02E416)) have been transferred, and the BC0 to BC2 are returned to "0002".

Also when a START condition is detected, these bits become "0002" and the address data is always transmitted and received in 8 bits.

### 16.4.2 Bit 3: I2C interface enable bit (ES0)

This bit enables to use the multi-master I2C bus interface. When this bit is set to "0", the interface is disabled and the SDA and the SCL become high-impedance. When the bit is set to "1", the interface is enabled.

When the ES0 bit is set to "0", the following is performed.

1)Set MST = 1, TRX = 0, PIN = 1, BB = 0, AL = 0, AAS = 0, and ADR0 = 0, of the I2C0 status register (Address : 02E816)

2)Writing the data into the I2C0 data shift register (Address : 02E016) is disabled.

3)The TOF bit in the I2C0 control register (Address : 02E716) is cleared to "0"

4)The I2C system clock (VIIC) is stopped and the internal counter, flags are initialized.

### 16.4.3 Bit 4: Data format select bit (ALS)

This bit decides if the recognition of the slave address is processed or not. When this bit is set to "0", the addressing format is selected and the address data is recognized. The transfer will be processed only when a comparison is matched between the slave address and the address data or a general call is received (Refer to **Figure 16.5 I2C0 status register: the item of bit 1, general call detection flag**). When this bit is set to "1", the free data format is selected and the slave address is not recognized.

### 16.4.4 Bit 6: I2C bus interface reset bit (IHR)

The bit is used to reset the I2C bus interface circuit when the abnormal communication occurs.

When the ES0 bit is "1" (I2C bus interface is enabled), writing "1" to the IHR bit resets H/W.

Flags are processed as follows:

1)Set MST = 0, TRX = 0, PIN = 1, BB = 0, AL = 0, AAS = 0, and ADR0 = 0, of I2C0 status register (Address : 02E816)

2)The TOF bit of the I2C0 control register 2 (Address : 02E716) is cleared to "0"

3)The internal counter, flags are initialized.

After writing"1" to the IHR bit, the circuit reset processing is finished in Max. 2.5 VIIC cycles and the IHR bit is automatically cleared to "0". Figure 16.10 shows the reset timing.

### 16.4.5 Bit 7: I2C bus interface pin input level select bit (TISS)

This bit selects the input level of the SCL and SDA pins of the multi-master I2C bus interface. When this bit is set to “1”, the P20 and P21 become the SMBus input level.

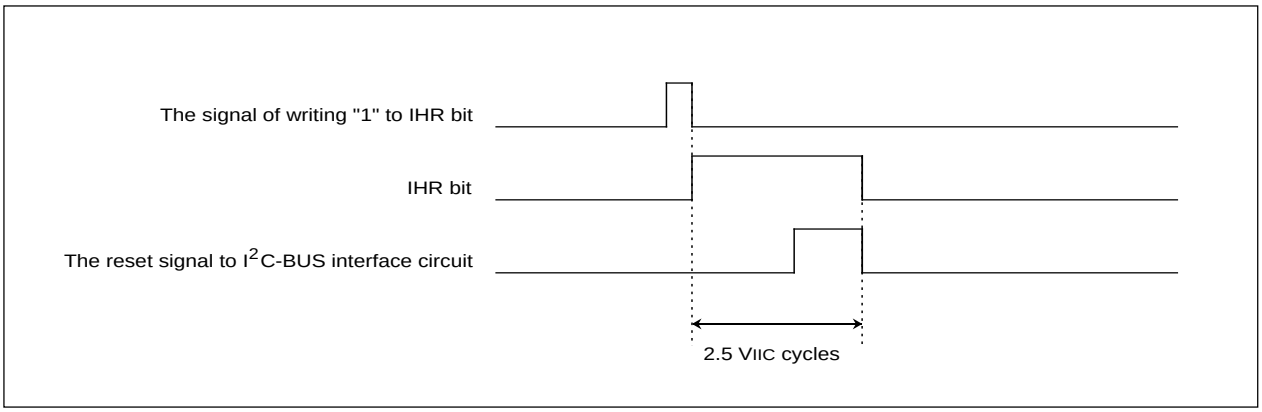



**Figure 16.10 The timing of reset to the I2C bus interface circuit**

## 16.5 I2C0 Status Register (S10 register)

The I2C0 status register (address 02E816) controls the I2C bus interface status. Use the lower-6 bit as read only if it is used for a status check.

### 16.5.1 Bit 0: Last receive bit (LRB)

This bit stores the last bit value of received data and can also be used for an ACK receive confirmation. If the ACK is returned when the ACK clock is generated, the LRB bit is set to “0”. If the ACK is not returned, this bit is set to “1”. Except in ACK mode, the last bit value of the received data is input. The bit is “0” by executing a write instruction to the I2C0 data shift register (address 02E016).

### 16.5.2 Bit 1: General call detection flag (ADR0)

When the ALS bit is “0”, this bit is set to “1” when a general call(Note 1),whose address data is all “0”, is received in slave mode. By a general call of the master device, every slave device receives control data after the general call. The ADR0 bit is set to “0” by detecting the STOP condition, START condition and when the ES0 is “0”, or reset.

Note 1. General call: The master transmits the general call address “0016” to all slaves.

### 16.5.3 Bit 2: Slave address comparison flag (AAS)

This flag indicates a comparison result of the address data when the ALS bit in the S1D0 register is “0”.
In slave receive mode, this bit is set to “1” in one of the following conditions:
- 7 bit of the address data matches the slave address stored in the S0D0 register.
- A general call is received.

The AAS flag is set to “0” in one of the following conditions:
- When the ES0 bit is set to “1”, excute to write an instruction to the S00 register
- When the ES0 bit is set to “0”.
- Excute to reset by the IHR bit in the S1D0 register.

### 16.5.4 Bit 3: Arbitration lost detection flag (AL)(Note 1)

When devices other than the microcomputer set the SDA to “L” in master transmit mode, the arbitration is judged to be lost and the AL bit is set to “1”. At the same time, the TRX bit is set to “0”. Immediately after the bute transmist, whose arbitration is lost, is completed, the MST bit is set to “0”. The arbitration lost can be detected only in master transmit mode. When the arbitration is lost during the slave address transmit, the TRX bit is set to “0” and the receive mode is set. Consequently, it is possible to detect the match between its own slave address and address data transmitted by another master devices. The bit becomes “0” if writing to the I2C0 data shift register (address 02E016) when the ES0 bit is “1”.
The bit also becomes “0” when the ES0 bit is set to “0” or when reset.

Note 1. Arbitration lost: The status is that communication as a master is disabled.

### 16.5.5 Bit 4: I2C bus interface interrupt request bit (PIN)

This bit generates an I2C bus interface interrupt request signal. After each byte data is transmitted, the PIN bit is changed from “1” to “0”. At the same time, an I2C bus interface interrupt request signal is generated to the CPU. The PIN bit is set to “0” synchronized with the falling edge of the last internal transmit clock (the ACK clock in ACK clock enable mode, the 8th clock in ACK clock disable mode) and an interrupt request signal is generated synchronized with the falling edge of the PIN bit. When the PIN bit is “0”, SCL is kept in the “0” state and the clock generation is disabled. In ACK clock enable mode, and when the WIT bit in the S3D0 register is set to “1”, synchronized with the falling edge of the last bit clock and the ACK clock, the PIN bit becomes to “0” and the I2C bus interface interrupt request is generated (Refer to **Section 16.6.2 Bit1: Interrupt enable bit at the completion of data receive (WIT)**. Figure 16.11 shows the timing of the I2C bus interface interrupt request generation.

The PIN bit is set to “1” in one of the following conditions:

- •Executing a write instruction to the S00 register (address 02E016).
- •Executing a write instruction to the S20 register (Address : 02E416)
  (only when the WIT is “1” and the internal WAIT flag is “1”)
- •When the ES0 bit is “0”
- •At reset

The PIN bit is set to “0” in one of the following conditions:

- •Immediately after the completion of the 1-byte data transmit (including arbitration lost is detected)
- •Immediately after the completion of the 1-byte data receive
- •In slave receive mode, with the ALS = 0 and immediately after the completion of the slave address match or the general call address receive
- •In slave receive mode, with the ALS = 1 and immediately after the completion of the address data receive

### 16.5.6 Bit 5: Bus busy flag (BB)

This bit indicates the operating conditions of the bus system. When this bit is set to “0”, the bus system is not used and a START condition can be generated. The BB flag is set/reset by the SCL and the SDA pins input the signal regardless of master or slave mode. This flag is set to “1” by detecting the start condition, and is set to “0” by detecting the stop condition. The condition of these detections is followed by the start/stop condition setting bits (SSC4–SSC0) of the S2D0 register (address 02E516). When the ES0 bit of the S1D0 register (address 02E316) is “0” or reset, the BB flag is set to “0”. For the writing function to the BB flag, refer to **Section 16.9 START Condition Generation Method and 16.11 STOP Condition Generation Method** as described later.

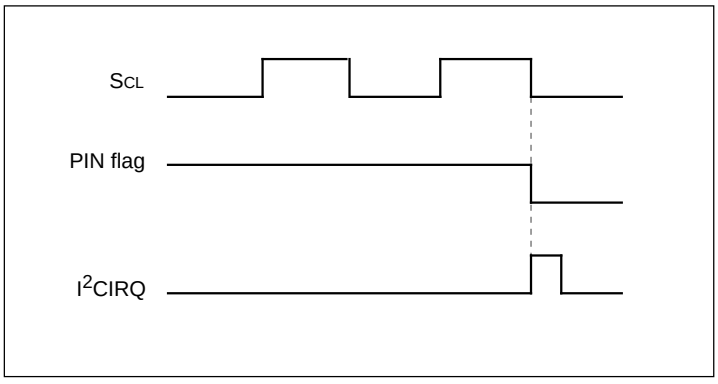



**Figure 16.11 Interrupt request signal generation timing**

### 16.5.7 Bit 6: Communication mode select bit (transfer direction select bit: TRX)

This bit decides a transfer direction for the data communication. When this bit is “0”, receive mode is selected and the data from a transmit device is received. When the bit is “1”, transmit mode is selected and the address data and the control data are output onto the SDA synchronized with the clock generated on the SCL. This bit can be set/reset by software or hardware. This bit is set to “1” by hardware in the following condition:

•In slave mode with the ALS = 0, if the AAS flag is set to “1” after the address data receive and the received R/$\overline{W}$ bit is “1”.

This bit is set to “0” by hardware in one of the following conditions:

•When an arbitration lost is detected.
•When a STOP condition is detected.
•When a START condition is detected.
•When a start condition is disabled by the START condition duplicate protect function (1).
•When a start condition is detected with MST = 0.
•When ACK non-return is detected with MST = 0.
•ES0 = 0.
•At reset

### 16.5.8 Bit 7: Communication mode select bit (master/slave select bit: MST)

This bit is used for the master/slave select bit for the data communication. When this bit is “0”, the slave is specified, so that a START condition and a STOP condition are generated by the master are received. The data communication is performed synchronized with the clock generated by the master. When this bit is “1”, the master is specified and a START condition and a STOP condition are generated.

Additionally, the clocks required for the data communication are generated on the SCL.

This bit is set to “0” by hardware in one of the following conditions.

•Immediately after the completion of 1-byte data transfer,which lost the arbitration, when arbitration lost is detected.
•When a STOP condition is detected.
•When a START condition is detected.
•Writing a start condition is disabled by the start condition duplicate protect function(Note 1).
•At reset

Note 1. START condition duplicate protect function

The MST, TRX, and BB bits are set to “1” at the same time after confirming that the BB flag is “0” in the procedure of a START condition generation. However, when a START condition generation by other master devices and the BB flag is set to “1” immediately after the contents of the BB flag are confirmed, the START condition duplicate protect function makes the writing to the MST and TRX bits invalid. The duplicate protect function becomes valid from the rising of the BB flag to receive completion of the slave address. Refer to **Section 16.9 START Condition Generation Method** for details.

## 16.6 I2C0 control register 1 (S3D0 register)

I2C0 control register 1 (address 02E616) controls I2C bus interface circuit.

### 16.6.1 Bit 0 : Interrupt enable bit by STOP condition (SIM )

This bit enables the I2C bus interface to request an I2C bus interface interrupt by detecting a STOP condition. If the bit set to “1”, an interrupt request from the I2C bus interface is generated by detecting a STOP condition ( There is no change for the PIN flag)

### 16.6.2 Bit 1: Interrupt enable bit at the completion of data receive (WIT)

When with ACK mode (ACK bit = 1) is specified, by the interrupt enable (WIT bit = 1) at the completion of data receive, the I2C bus interface interrupt request is generated and the PIN bit becomes “0” synchronized with the falling edge of the last data bit clock. The SCL becomes “L” and the ACK clock generation is suppressed.

Table 16.4 and Figure 16.12 show the I2C Bus interrupt request timing and the communication restart method. After the communication restart, synchronized with the falling edge of ACK clock, the PIN bit becomes “0” again and the I2C bus interface interrupt request is generated.

**Table16.4 Timing of interrupt generation in data receive**

| I2C Bus interrupt generation timing | Communication restart method |
|---|---|
| 1) Synchronized with the falling edge of the last data bit clock | The execution of writing to ACK bit of I2C0 clock control register. Follow this by a register write to set PIN bit = 1. (Do not write to the I2C0 data shift register. The ACK clock operation can be incorrect.) |
| 2) Synchronized with the falling edge of the ACK clock | The execution of writing to the I2C0 data shift register |

The state of the internal WAIT flag can be read out by reading the WIT bit. The internal WAIT flag is set after writing to the I2C0 data shift register, and it is reset after writing to the I2C0 clock control register. Consequently, the I2C bus interface interrupt request generated by the timing 1) or 2) can be determined. (See **Figure 16.12 The timing of the interrupt generation at the competion of data receive**.) In the cases of transmit and the address data receive immediately after the START condition, the I2C bus interface interrupt request is only generated at the falling edge of the ACK clock regardless of the value of the WIT bit and the WAIT flag remains the reset state. Write “0” to the WIT bit when in NACK is specified. (ACK bit = 0)

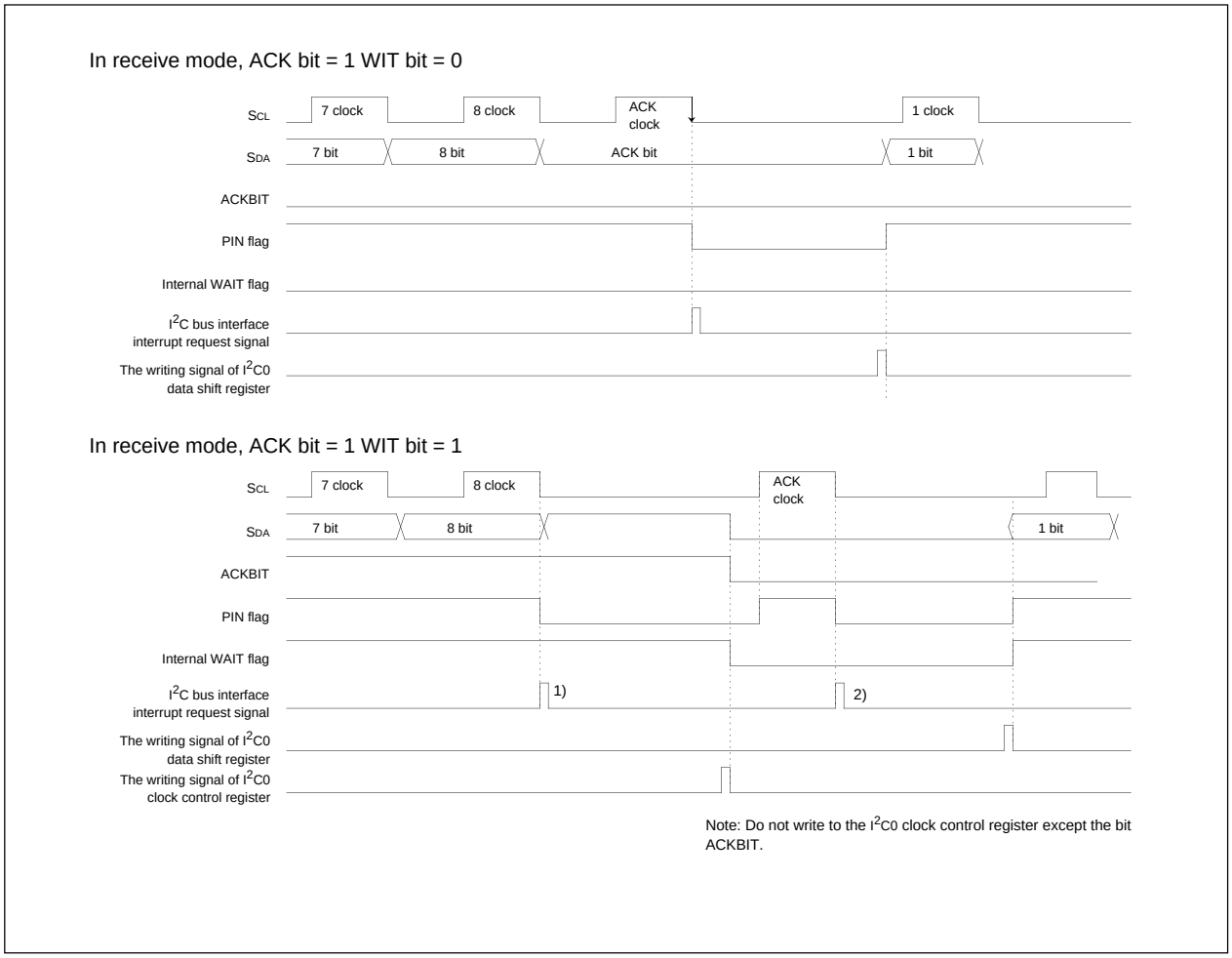



**Figure 16.12 The timing of the interrupt generation at the completion of the data receive**

## 16.6.3 Bits 2,3 : Port function select bits PED, PEC

When the ES0 bit of the I²C0 control register 0 is set to "1", P21 and P20 functions as SCL and SDA pins respectively. However, if the PED is set to "1", the SDA functions as the output port so as to the SCL if the PEC is set to "1". In this case, if "0" or "1" is written to the port register, the data can be output onto the I²C bus regardless of the internal SCL/SDA output signals. The functions of SCL/SDA are returned back by setting the PED to "1" again.

If the ports are set in input mode, the values on the I²C bus can be known by reading the port register regardless of the values of the PED and PEC.Table 16.5 shows the port specification.

**Table 16.5 Port specifications**

| Pin name | ES0 bit | PED bit | P20 port direction register | Function |
|---|---|---|---|---|
| P20 | 0 | - | 0/1 | Port I/O function |
| | 1 | 0 | - | SDA I/O function |
| | 1 | 1 | - | SDA input function, port output function |
| P21 | ES0 bit | PEC bit | P21 port direction register | Function |
| | 0 | - | 0/1 | Port I/O function |
| | 1 | 0 | - | SCL I/O function |
| | 1 | 1 | - | SCL input function, port output function |

### 16.6.4 Bits 4,5 : SDA/SCL logic output value monitor bits SDAM/SCLM

These bits enableto monitor the logic value of the SDA and SCL output signals from the I2C bus interface circuit. The SDAM bit monitors the SDA output logic value. The SCLM bit monitors the SCL output logic value. The bits are read-only. When write, set to “0”.

### 16.6.5 Bits 6,7 : I2C system clock select bits ICK0, ICK1

These bits and ICK4 to ICK2 bits in the S4D0 register select the system clock (VIIC) of the I2C bus interface circuit. These bits enable to select the I2C bus system clock VIIC among divisions by 2, 2.5, 3, 4, 5, 6 or 8 of the main clock f(XIN).

**Table 16.6 I2C system clock select bits**

| I3CK4[S4D0] | ICK3[S4D0] | ICK2[S4D0] | ICK1[S3D0] | ICK0[S3D0] | I2C system clock |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | VIIC = 1/2 f(XIN) |
| 0 | 0 | 0 | 0 | 1 | VIIC = 1/4 f(XIN) |
| 0 | 0 | 0 | 1 | 0 | VIIC = 1/8 f(XIN) |
| 0 | 0 | 1 | X | X | VIIC = 1/2.5 f(XIN) |
| 0 | 1 | 0 | X | X | VIIC = 1/3 f(XIN) |
| 0 | 1 | 1 | X | X | VIIC = 1/5 f(XIN) |
| 1 | 0 | 0 | X | X | VIIC = 1/6 f(XIN) |

( Do not set the combination which is not indicated here)

### 16.6.6 The address receive in STOP mode/WAIT mode

The I2C bus interface circuit enables to receive the address data in WAIT mode when setting the CM02 bit in the CM0 register to "0" (do not stop the peripheral function clock in wait mode) and entering WAIT mode. However, the I2C bus interface circuit is not operated in STOP mode or in low power consumption mode, because the I2C bus system clock VIIC is not supplied.

## 16.7 I2C0 control register 2 (S3D0 register)

I2C0 control register 2 (address: 02E716) controls the abnormal communication detection. In the I2C bus communication, the data transfer is controlled by the SCL clock signal. The devices are stoped in the communication state if the SCL clock is stopped during the transfer. To avoid that, if the SCL clock is stopped in “H” state for a period of time, the I2C bus interface circuit has the function to detect the time out and generate an I2C bus interface interrupt request. Please see **Figure 16.13 The timing of time out detection**.

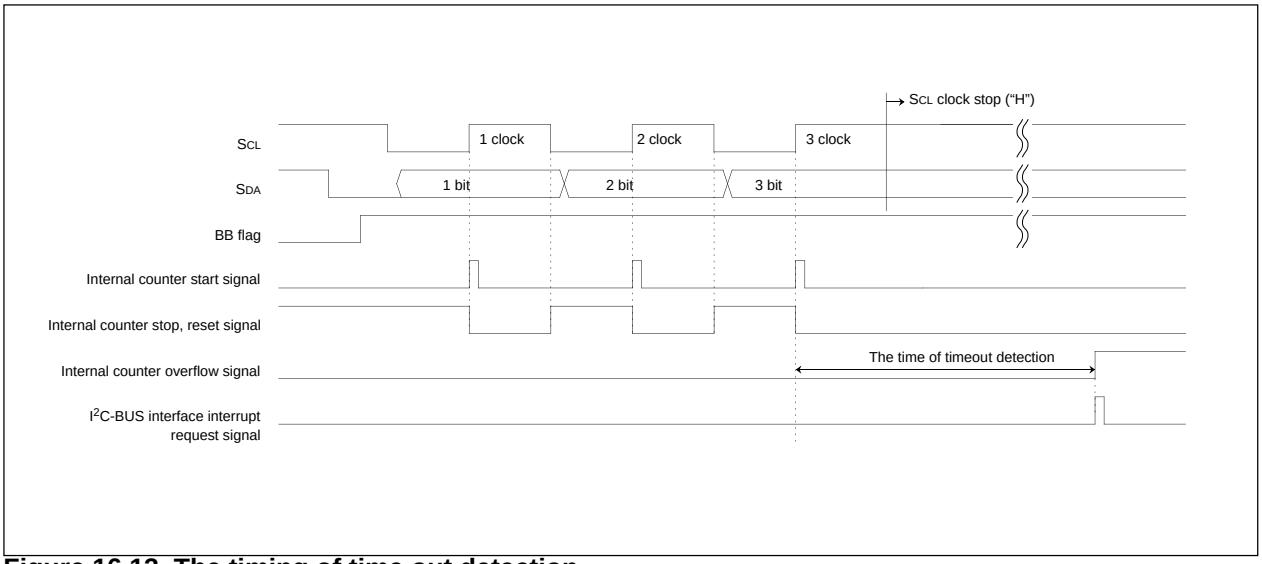



**Figure 16.13 The timing of time out detection**

### 16.7.1 Bit0: Time out detection function enable bit (TOE)

The bit enables a time out detection function. When setting this bit to “1”, the I2C bus interface interrupt request signal is generated if the SCL clock is stopped in “H” state for a period of time during the bus busy (BB flag =1).

The time out detection period is measured by the internal counter and selected from long time mode or short time mode by the time out detection period select bit (TOSEL). When time out is detected, set the ES0 bit to “0” and then process initialization.

### 16.7.2 Bit1: Time out detection flag (TOF )

The bit is the flag showing the time out detection status. If the internal counter which measures the time out period overflows, the time out detection flag (TOF) becomes to “1”, and at the same time the I2C bus interface interrupt request signal is generated.

### 16.7.3 Bit2: time out detection period select bit (TOSEL)

The bit selects time out detection period from long time and short time mode. If the TOSEL = 0, the long time mode and TOSEL = 1, the short time mode is selected respectively. The long time is counted by 16-bit counters and the short time is counted by 14-bit counters based on the I2C system clock (VIIC). Table 16.7 shows examples of the time out detection period.

**Table 16.7 Examples of time out detection period (Unit: ms)**

| VIIC(MHz) | Long time mode | Short time mode |
|---|---|---|
| 4 | 16.4 | 4.1 |
| 2 | 32.8 | 8.2 |
| 1 | 65.6 | 16.4 |

### 16.7.4 Bits 3,4,5: I2C system clock select bits (ICK2-4)

ICK4 to 2 bits, ICK1 and ICK0 bits of the S3D0 register select the system clock (VIIC) of the I2C bus interface circuit.Table 16.6 shows the I2C system clock setting for the setting values.

### 16.7.5 Bit7: STOP condition detection interrupt request bit (SCPIN)

The bit monitors the stop condition detection interrupt. The bit becomes to “1” when the I2C bus interface interrupt is generated by detecting of the STOP condition. Writing “0” clears the bit and “1” can not be written.

## 16.8 I2C0 START/STOP condition control registers (S2D0 register)

The I2C0 START/STOP condition control register(address 02E516) controls the detection of the START/STOP condition.

### 16.8.1 Bit0-Bit4: START/STOP condition setting bits (SSC0-SSC4)

Because the release time, the set up time and the hold time of the SCL are measured on the base of the I2C bus system clock(VIIC). The detecting condition changes depending on the oscillation frequency (XIN) and the I2C bus system clock select bits. It is necessary to set the appropriate value of START/STOP condition setting bits (SSC4-SSC0) and set the release time, the set up time and the hold time by the system clock frequency. Refer to **Table 16.10 Start/Stop condition detect conditions**. Do not set odd numbers or "000002" to START/STOP condition setting bits. Table 16.2 shows the recommended setting value to START/STOP condition setting bits (SSC4-SSC0) at each oscillation frequency under standard clock mode. The detection of the START/STOP condition starts immediately after setting the ES0 bit to "1".

### 16.8.2 Bit5: SCL/SDA interrupt pin polarity select bit (SIP)

The SCL/SDA interrupt can be generated by detecting the rising edge or the falling edge of the SCL pin or the SDA pin. The SCL/SDA interrupt pin polarity select bit selects the polarity of the SCL pin or the SDA pin for interrupt.

### 16.8.3 Bit6 : SCL/SDA interrupt pin select bit (SIS)

The SCL/SDA interrupt pin select bit selects either the SCL pin or the SDA pin as the SCL/SDA interrupt enable pin.

NOTES:

The SCL/SDA interrupt request may be set when the setting of the SCL/SDA interrupt pin polarity se lect bit, SCL/SDA interrupt pin select bit and I2C bus interface enable bit ES0 are changed. When using the SCL/SDA interrupt, write "0" to the SCL/SDA interrupt request bit after setting the above bits, and enable the SCL/SDA interrupt.

### 16.8.4 Bit7: START/STOP condition generation select bit (STSPSEL)

The bit selects the length of the set up and the hold time when the START/STOP condition is generated. The length of the set up and hold time is based on the I2C system clock cycles. Refer to **Table 16.8 Start/Stop generation timing table**. Set the bit to "1" if the I2C bus system clock frequency is over 4MHz.

## 16.9 START Condition Generation Method

When the ES0 bit of the I2C0 control register is “1” and the BB flag of the I2C0 status register is “0”, writing “1” to the MST, TRX, and BB bits and “0” to the PIN and low-order bits of the I2C0 status register (S10 register) simultaneously enters the standby status to generate the start condition. The start condition is generated after writing the slave address data to the I2C0 data shift register. After that, the bit counter becomes “$000_2$” and 1-byte SCL are output. The start condition generation timing is different in standard clock mode and high-speed clock mode. Refer to **Figure 16.16 Start condition generation timing diagram, and Table 16.8 Start/Stop generation timing table**.

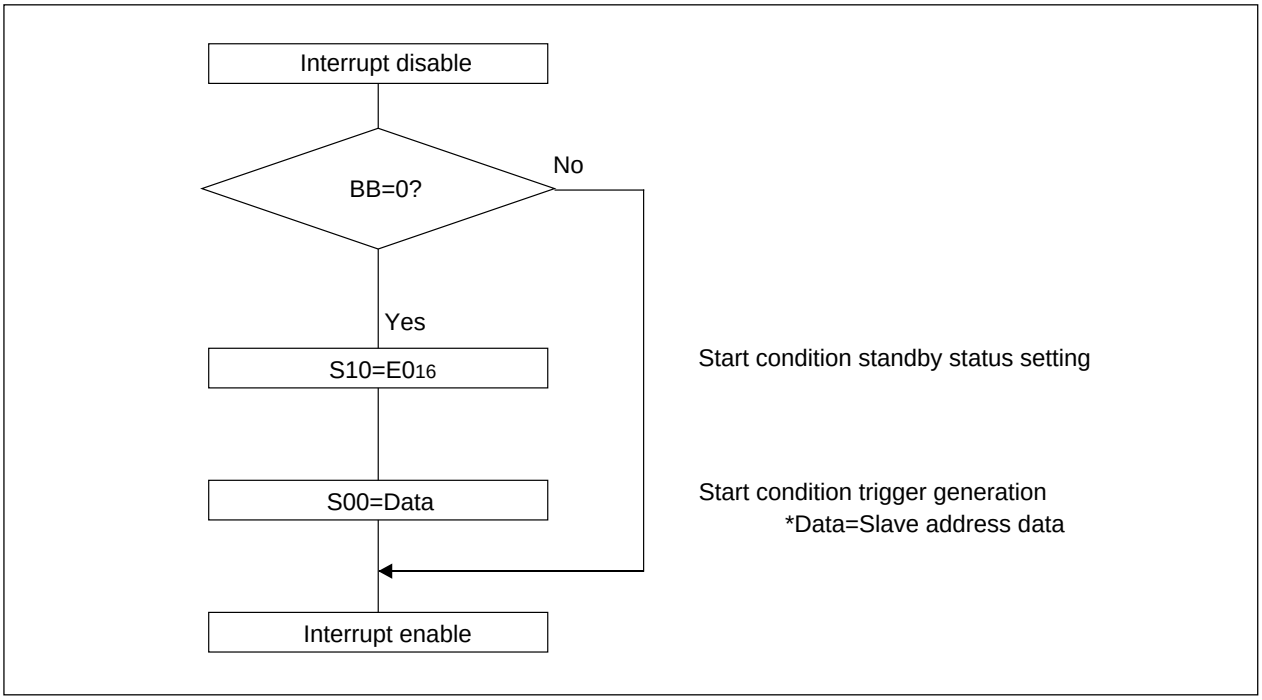



**Figure 16.14 Start condition generation flow chart**

## 16.10 START condition duplicate protect function

It is necessary to verify that the bus is not in use via the BB flag before the start condition is generated. However,when the BB flag is set to “1” because a start condition is generated by another master devices immediately after the BB flag is verified, the start condition is suspended by the start condition duplicate protect function. When the function starts, it works as follows:

•The start condition standby setting is disabled.

If the start condition standby has been set, release it and resets the MST and TRX bits.

•Writing to the I2C0 data shift register is disabled. (The start condition trigger generation is disabled)

•When the start condition generation is interrupted, sets the AL flag.

The start condition duplicate protect function is valid from the SDA falling edge of the start condition to the slave receive completion. Figure16.15 shows the duration of the start condition duplicate protect function.

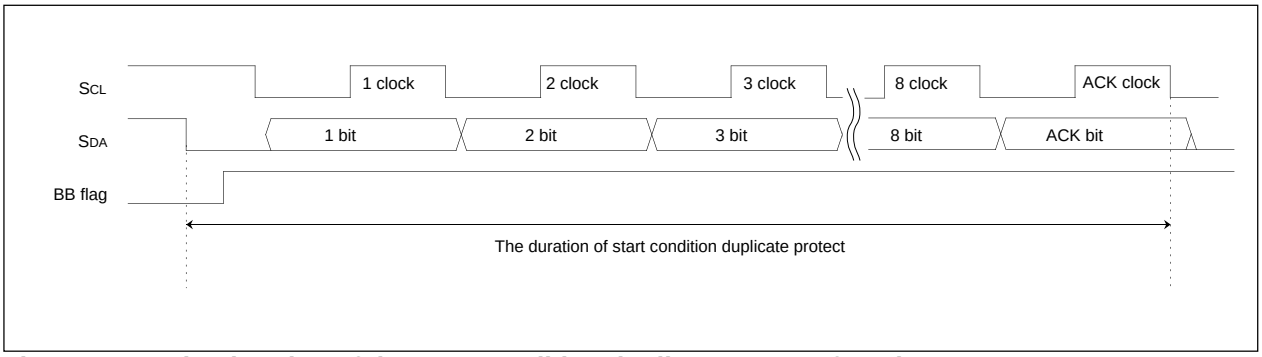



**Figure 16.15 The duration of the start condition duplicate protect function**

## 16.11 STOP Condition Generation Method

When the ES0 bit in the I2C0 control register is “1”, writing “1” to the MST and the TRX bits in the I2C0 status register, and “0” to the BB, PIN and low-order 4 bits in the I2C0 status register simultaneously enters the standby status to generate the stop condition. The stop condition is generated after writing the dummy data to the I2C0 data shift register. The stop condition generation timing is different in standard clock mode and high-speed clock mode. Refer to **Figure 16.17 STOP condition generation timing diagram**, and **Table 16.8 Start/Stop generation timing table**. Do not write data to the I2C0 status register and the I2C0 data shift register, before the BB flag becomes “0” after executing the instruction to generate the stop condition. Otherwise, the stop condition waveform may not be operated normally.

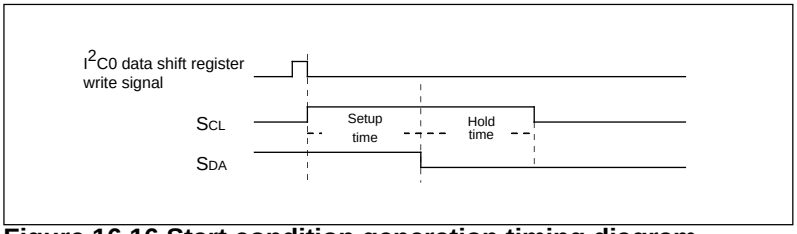



**Figure 16.16 Start condition generation timing diagram**

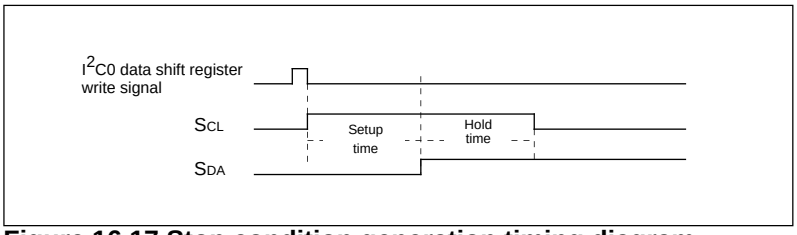



**Figure 16.17 Stop condition generation timing diagram**

**Table 16.8 Start/Stop generation timing table**

| Item | Start/Stop condition generation select bit | Standard clock mode | High-speed clock mode |
|---|---|---|---|
| Setup time | "0" | 5.0µs (20 cycles) | 2.5µs (10 cycles) |
| | "1" | 13.0µs (52 cycles) | 6.5µs (26 cycles) |
| hold time | "0" | 5.0µs (20 cycles) | 2.5µs (10 cycles) |
| | "1" | 13.0µs (52 cycles) | 6.5µs (26 cycles) |

Note 1. Actual time at the time of $V_{IIC}$ = 4MHz, The contents in () denote cycle numbers.

As mentioned above, Writing "1" to MST and TRX bits.
Writing "1" or "0" to the BB bit, writing "0" to the PIN and low-order 4 bits, simultaneously set up the START or STOP condition standby. It releases the SDA in the START condition standby, sets the SDA to "L" in the STOP condition standby. The signal writing to data shift register triggers the generation of START/STOP conditions. In the case of setting the MST, and the TRX to "1" without generating a START/STOP condition. Write "1" to the low-order 4 bits simultaneously. Table16.9 shows the function of writing to the status register.

**Table 16.9 The function of writing to status register**

| The value of the data writing to status register | | | | | | | | Function |
|---|---|---|---|---|---|---|---|---|
| MST | TRX | BB | PIN | AL | AAS | AS0 | LRB | |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | Setting up the START condition stand by in master transmit mode |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | Setting up the STOP condition stand by in master transmit mode |
| 0/1 | 0/1 | - | 0 | 1 | 1 | 1 | 1 | Setting up each communication mode (refer to **Chapter 16.5 I²C status register**) |

## 16.12 START/STOP Condition Detect Operation

Figure 16.18, Figure 16.19 and Table 16.10 show START/STOP condition detect operations. The START/STOP condition is set by the START/STOP condition set bit. The START/STOP condition can be detected only when the input signal of the SCL and SDA pins satisfied with three conditions: the SCL release time, the setup time, and the hold time (see **Table.16.10 Start/Stop condition detect conditions**). The BB flag is set to "1" by detecting the start condition and is set to "0" by detecting the stop condition. The BB flag set and reset timing are different in standard clock mode and high-speed clock mode. See **Table.16.10 Start/Stop condition detect conditions**.

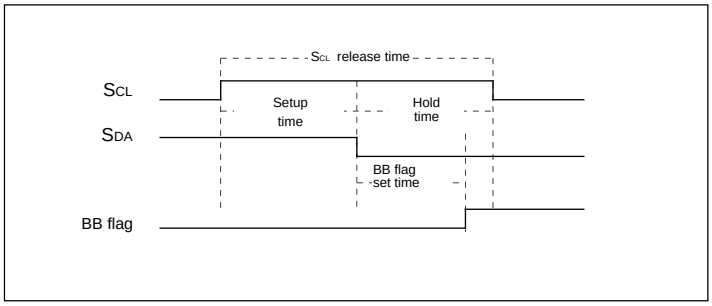



**Figure 16.18 Start condition detection timing diagram**

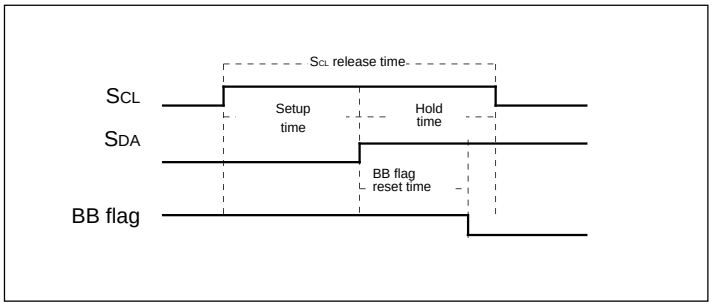



**Figure 16.19 Stop condition detection timing diagram**

**Table 16.10 Start/Stop detection timing table**

| | Standard clock mode | High-speed clock mode |
|---|---|---|
| SCL release time | SSC value + 1 cycle (6.25µs) | 4 cycles (1.0µs) |
| Setup time | $\frac{\text{SSC value}}{2}$ + 1 cycle < 4.0µs (3.25µs) | 2 cycles (0.5µs) |
| Hold time | $\frac{\text{SSC value}}{2}$ cycle < 4.0µs (3.0µs) | 2 cycles (0.5µs) |
| BB flag set/reset time | $\frac{\text{SSC value - 1}}{2}$ +2 cycles (3.375µs) | 3.5 cycles (0.875µs) |

Note 1. Unit : Cycle numbers of I2C system clock VIIC
The SSC value is the decimal notation value of the START/STOP condition set bits SSC4 to SSC0. Do not set "0" or odd numbers to the SSC value. The values in () are examples when the I2C0 start/stop condition control register is set to "1816" at VIIC = 4 MHz.

## 16.13 Address Data Communication

### 16.13.1 Example of Master Transmit

An example of master transmit in standard clock mode, at the SCL frequency of 100 kHz and in ACK return mode is shown below.

1)Set the slave address in the upper 7-bit of I2C0 address registers (S0D0).
2)Set ACK return mode and the SCL = 100 kHz by setting “0016” in the I2C0 control register 1(S3D0), “0002” in the ICK4 to ICK2 bits of the I2C0 control register 2(S4D0) and “8516” in the I2C0 clock control register (S20) respectively. (f1=8MHz)
3)Set “0016” in the I2C0 status register (S10) so that transmit/receive mode is initialized.
4)Set a communication enable status by setting “0816” in the I2C0 control register 0 (S1D0).
5)Confirm the bus free condition by the BB flag of the I2C0 status register (S10).
6)Set “E016” in the I2C0 status register (S10) to set the start condition standby.
7 )Set the destination address data for transmit in high-order 7 bit in the I2C0 data shift register (S00) and set “0” in the least significant bit. And then a start condition is generated. At this time, SCL for 1 byte and an ACK clock are automatically generated.
8)Set transmit data in the I2C0 data shift register (S00). At this time, an SCL and an ACK clock are automaticall generated.
9)When transmitting more than 1-byte control data, repeat step 7).
10)Set “C016” in the I2C0 status register (S10) to set a stop condition if ACK is not returned from the slave receive side or the transmit end.
11)A stop condition is generated when writing the dummy data to the I2C0 data shift register (S00).

Figure 16.20 (1) shows the master transmit format.

### 16.13.2 Example of Slave Receive

An example of the slave receive in high-speed clock mode, at the SCL frequency of 400 kHz, in ACK return mode and using the addressing format is shown below.

1)Set a slave address in the high-order 7 bits in the I2C0 address register (S0D0).
2)Set ACK clock mode and SCL = 400 kHz by setting "0016" in the I2C0 control register 1 (S3D0), "0002" in the ICK4 to ICK2 bits in the I2C0 control register 2 (S4D0) and "A516" in the I2C0 clock control register (S20) respectively. (f1=8MHz)
3)Set "0016" in the I2C0 status register (S10) so that transmit/receive mode is initialized.
4)Set a communication enable status by setting "0816" in the I2C0 control register 0 (S1D0).
5)When a start condition is received, an address comparison is performed.
6)•When all transmitted addresses are "0" (general call):
ADR0 in the I2C0 status register (S10) is set to "1" and an I2C bus interface interrupt request signal is generated.
•When the transmitted addresses match with the address set in 2):
ASS in the I2C0 status register (S10) is set to "1" and an I2C bus interface interrupt request signal occurs.
•In the cases other than the above ADR0 and AAS of the I2C0 status register are set to "0" and no I2C bus interface interrupt request signal occurs.
7)Set dummy data in the I2C0 data shift register (S00).
8)After receiving 1-byte data, an ACK is automatically returned and an I2C bus interface interrupt request signal is generated.
9)After receiving 1-byte data, an I2C bus interface interrupt request signal is generated, set the ACKBIT to "1" or "0" by reading the contents in the data shift register (S00). and the ACK bit is returned or not returned.
10)When receiving more than 1-byte control data, repeat step 7) 8) or 7) 9).
11)When a STOP condition is detected, the communication ends.
Refer to **Figure 16.20 Address data communication format, (2)**.

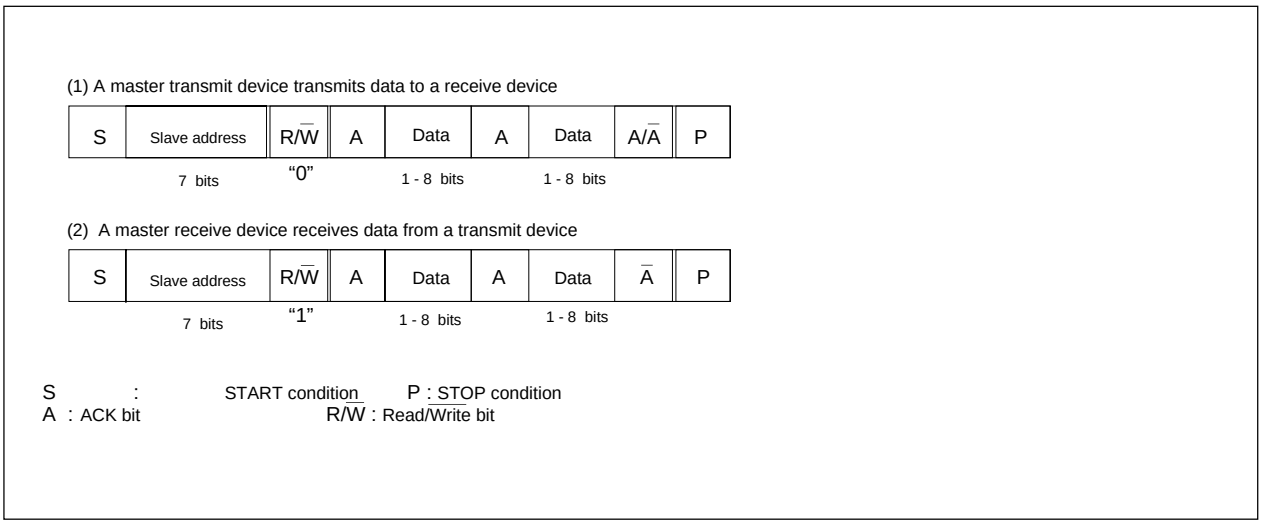



**Figure 16.20 Address data communication format**

## 16.14 Usage precautions

(1) Access to the registers of I2C bus interface circuit

The precaution of read/write to the control registers of I2C bus interface circuit is as follows.

•I2C0 data shift register (S00 : 02E016)

Do not write the register during the data transfer. The transfer bit counter is reset and the data may not be transfered normally.

•I2C0 control register 0 (S1D0 : address 02E316).

After the start condition detection or the 1-byte transfer completion, the bit counter (bits BC2 to BC0) is reset by Hardware. Do not read/write the register at this time, because the data may be undetermined.

Figure 16.22 and Figure 16.23 show the bit counter reset timing by Hardware.

•I2C0 clock control register (S20 : address 02E416)

Do not write to this register except the ACKBIT during the transfer. The I2C clock generator is reset and the data may not be transfered normally.

•I2C0 control register 1 (S3D0 : address 02E616)

Write I2C system clock select bits when I2C bus interface enable bit (ES0)is disabled. When the data receive completion interrupt enable bit (WIT) reads out, the internal WAIT flag is read.Do not use the bit managing instrustion (read-modify-write instruction) to access the register.

•I2C0 status register (S10 : address 02E816)

Do not use the bit managing instruction (read-modify-write instruction) to access the register be cause all bits of this register are changed by H/W. Do not read/write during the timing when the MST and the TRX bits for the communication mode setting are changed. The data may be undetermined. Figure16.21 to Figure 16.23 show the timing when the MST and the TRX bits are changed by H/W.

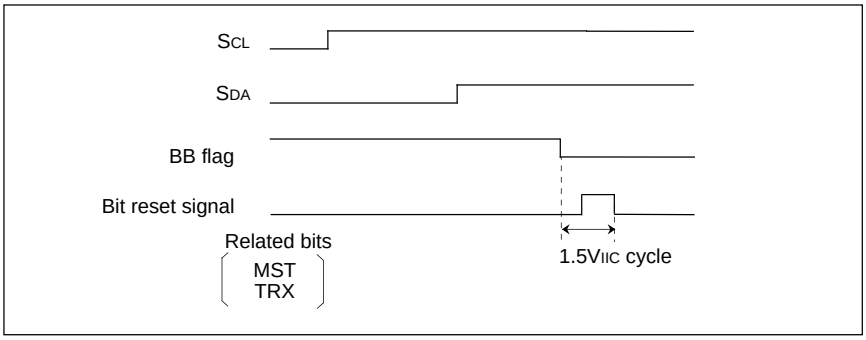



**Figure 16.21 The bit reset timing (The STOP condition detection)**

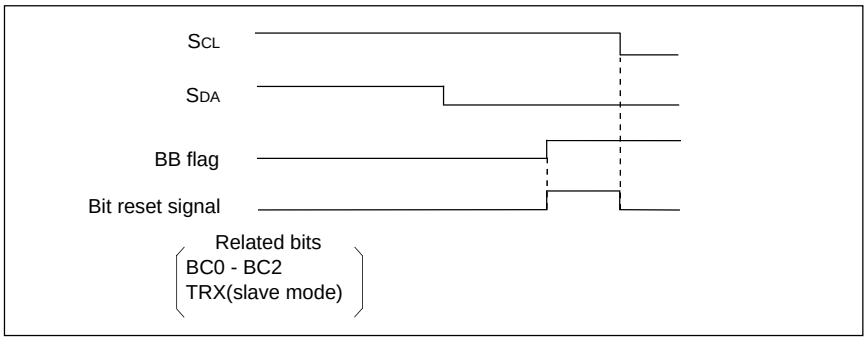



**Figure 16.22 The bit reset timing (The START condition detection)**

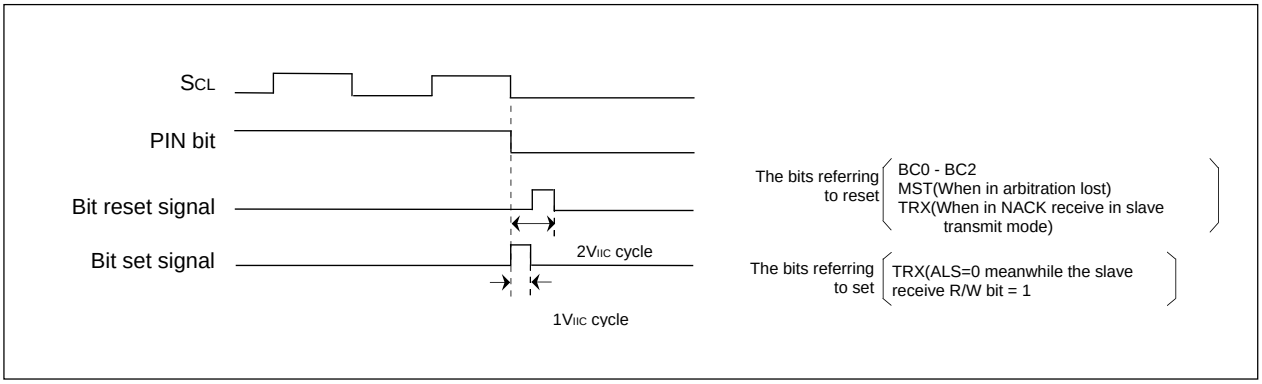



**Figure 16.23 Bit set/reset timing ( at the completion of data transfer)**

(2) Generation of RESTART condition

After 1-byte data transfer and a restart condition is generated, write"E016" to I2C0 status register, set the start condition standby and the SDA pin will be released. Writing to the I2C0 data shift register generates the start condition trigger after waiting in software until the SDA becomes "H". Figure 16.24 shows the restart condition generation timing.

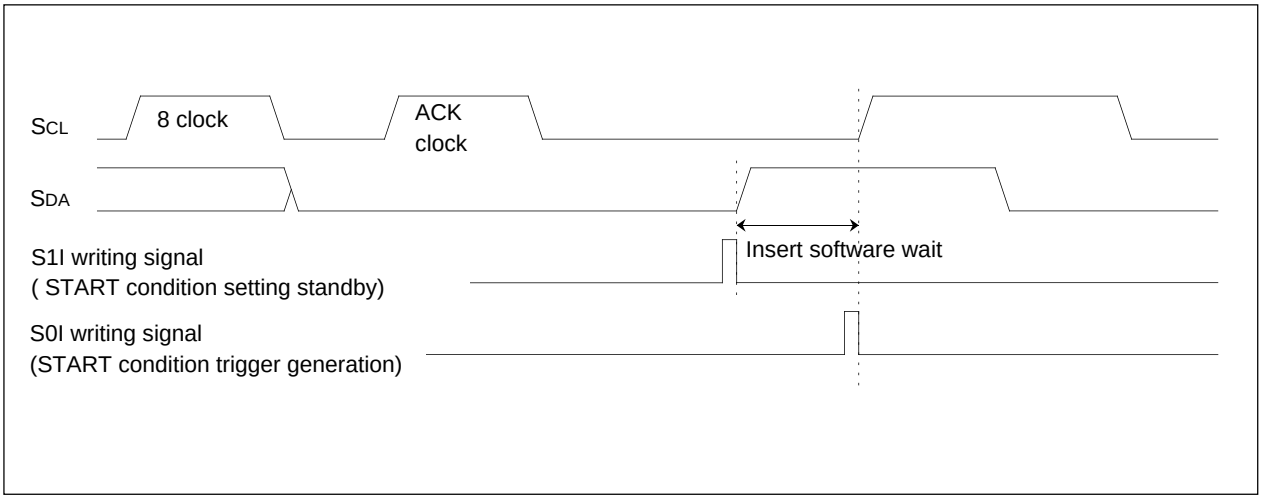



**Figure 16.24 The time of generation of RESTART condition**

(3) Iimitation of CPU clock

The registers of I2C bus interface circuit can not be read from or written to if the CPU clock is selected to the sub clock (XCIN, XCOUT) by the system clock select bit (system clock control register 0, address 0006h, CM07 bit). Select the main clock (XIN, XOUT) or the ring oscillator clock in read/write.

# 17. Programmable I/O Ports

The programmable input/output ports (hereafter referred to simply as “I/O ports”) consist of 71 lines P0, P1,P2, P3, P6, P7, P8, P9, P10 (except $P9_4$) for the 80-pin version, or 55 lines $P0_0$ to $P0_3$, $P1_5$ to $P1_7$, P2, $P3_0$ to $P3_3$, P6, P7, P8, $P9_0$ to $P9_3$, P10 for the 64-pin version. Each port can be set for input or output every line by using a direction register, and can also be chosen to be or not be pulled high in sets of 4 lines.
Figures 17.1 to 17.5 show the I/O ports. Figure 17.6 shows the I/O pins.
Each pin functions as an I/O port, a peripheral function input/output.
For details on how to set peripheral functions, refer to each functional description in this manual. If any pin is used as a peripheral function input, set the direction bit for that pin to “0” (input mode). Any pin used as an output pin for peripheral functions is directed for output no matter how the corresponding direction bit is set.

## 17.1 Port Pi Direction Register (PDi Register, i = 0 to 3, 6 to 10)

Figure 17.1.1 shows the direction registers.
This register selects whether the I/O port is to be used for input or output. The bits in this register correspond one for one to each port.

## 17.2 Port Pi Register (Pi Register, i = 0 to 3, 6 to 10)

Figure 17.2.1 shows the Pi registers.
Data input/output to and from external devices are accomplished by reading and writing to the Pi register. The Pi register consists of a port latch to hold the output data and a circuit to read the pin status. For ports set for input mode, the input level of the pin can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register.
For ports set for output mode, the port latch can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register. The data written to the port latch is output from the pin. The bits in the Pi register correspond one for one to each port.

## 17.3 Pull-up Control Register 0 to Pull-up Control Register 2 (PUR0 to PUR2 Registers)

Figure 17.3.1 shows the PUR0 to PUR2 registers.
The PUR0 to PUR2 register bits can be used to select whether or not to pull the corresponding port high in 4 bit units. The port chosen to be pulled high has a pull-up resistor connected to it when the direction bit is set for input mode.

## 17.4 Port Control Register

Figure 17.4.1 shows the port control register.
When the P1 register is read after setting the PCR register’s PCR0 bit to “1”, the corresponding port latch can be read no matter how the PD1 register is set.

## 17.5 Pin Assignment Control register (PACR)

Figure 17.5.1 shows the PACR. After reset set PACR2 to PACR0 bit in the PACR register before you input and output to each pin. When the PACR register isn't set up, the input and output function of some of the pins doesn't work.

PACR2 to PACR0 : control the number of pins enabled for use.

At reset these bits equal "$000_2$" .
When using the 80 pin version of the M16C/28 set these bits to "$011_2$".
When using the 64 pin version of the M16C/28 set these bits to "$010_2$".

U1MAP : controls the assignment of UART1 pins.

If U1MAP = "0" (default at reset) the UART1 functions are assigned to $P6_4$/$CTS_1$/$RTS_1$, $P6_5$/$CLK_1$, $P6_6$/$RxD_1$, and $P6_7$/$TxD_1$.
If U1MAP = "1" the UART1 functions are assigned to $P7_0$/$CTS_1$/$RTS_1$, $P7_1$/$CLK_1$, $P7_2$/$RxD_1$, and $P7_3$/$TxD_1$.

PACR is write protected by PRC2 bit of PRCR (protect register). PRC2 bit of PRCR must be set immediately before the write to PACR.

## 17.6 Digital Debounce function

Two digital debounce function circuits are provided. Level is determined when level is held, after applying either a falling edge or rising edge to the pin, longer than the programmed filter width time. This enables noise reduction.
This function is assigned to $\overline{INT5}$/INPC17 and $\overline{NMI}$/$\overline{SD}$. Digital filter width is set in the NDDR register and the P17DDR register respectively.Figure 17.6.1 shows the NDDR register and the P17DDR register. Additionally, a digital debounce function is disabled to the port $P1_7$ input and the port $P8_5$ input.

Filter width : $f8 \times 1 / (n+1)$ n: count value set in the NDDR register and P17DDR register

The NDDR register and the P17DDR register decrement count value with f8 as the count source. The NDDR register and the P17DDR register indicate count time. Count value is reloaded if a falling edge or a rising edge is applied to the pin.
The NDDR register and the P17DDR register can be set $00_{16}$ to $FF_{16}$ when using the digital debounce function. Setting to $FF_{16}$ disables the digital filter. See Figure 17.6.2 for details.

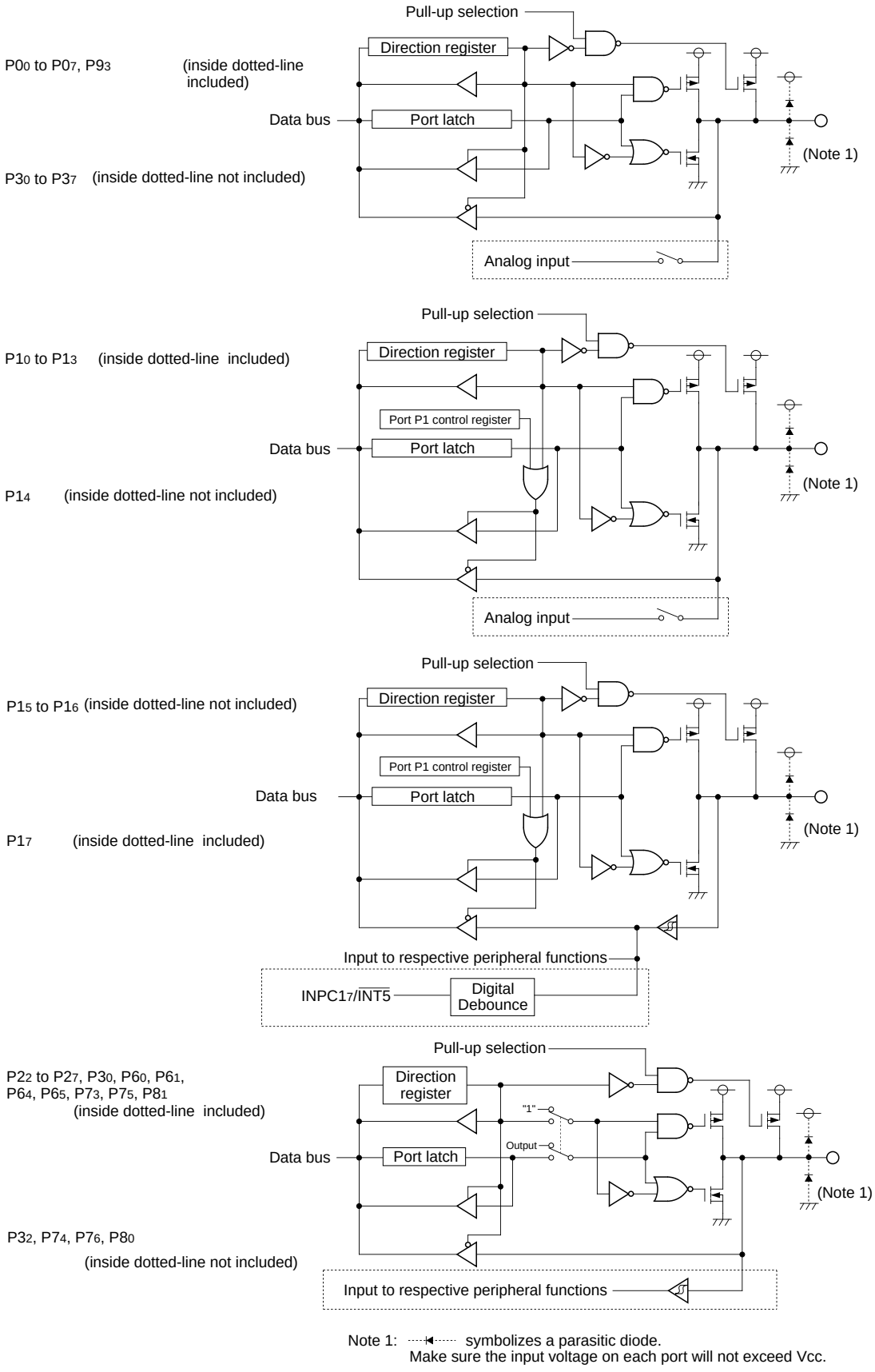
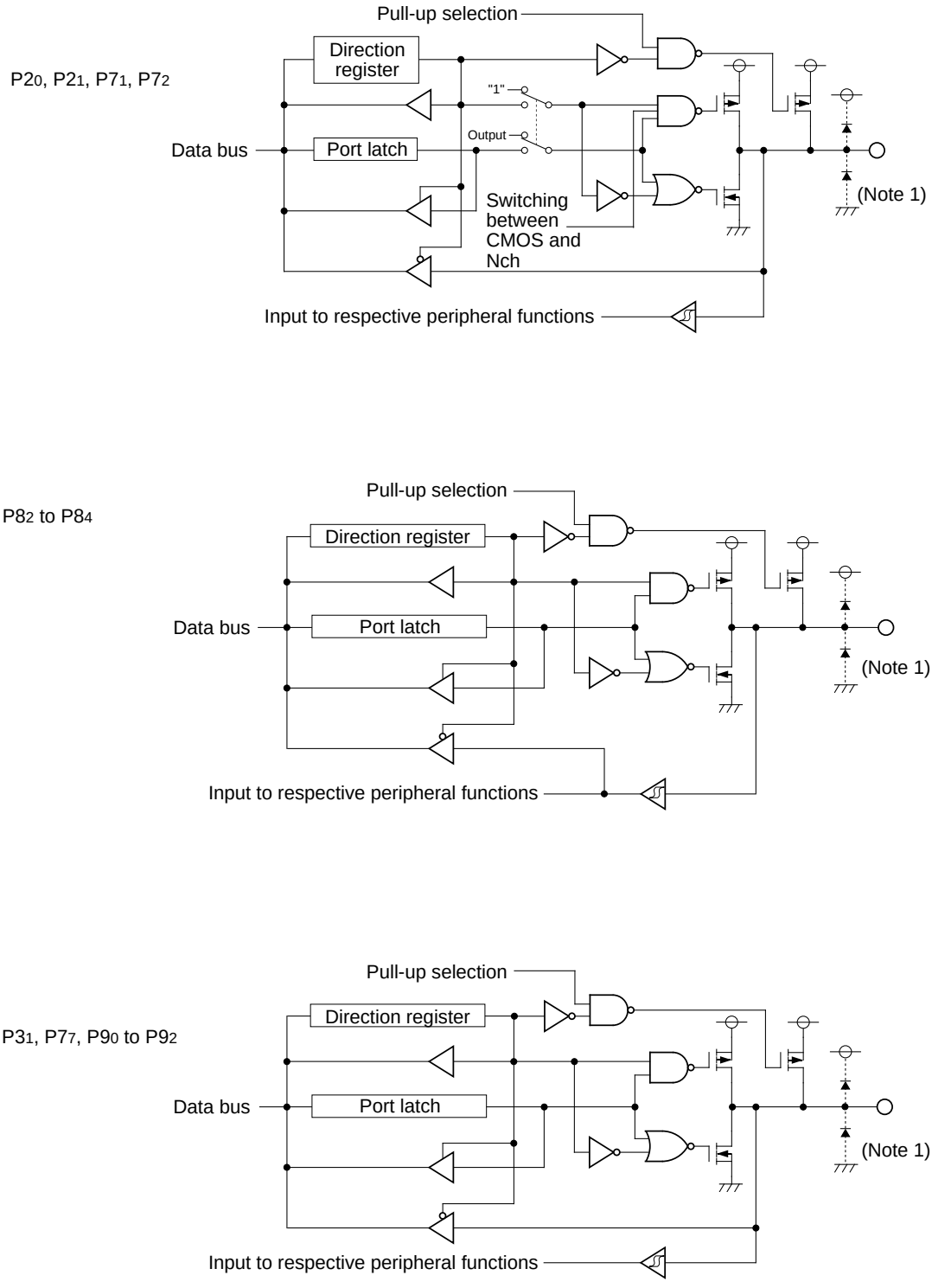


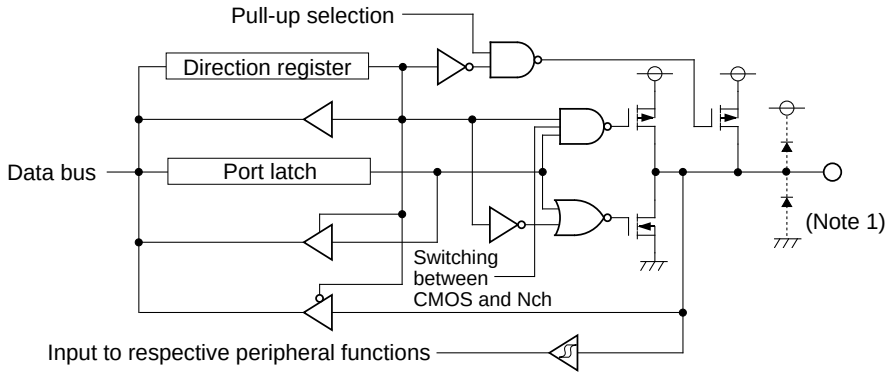

**Figure 17.1. I/O Ports (1)**



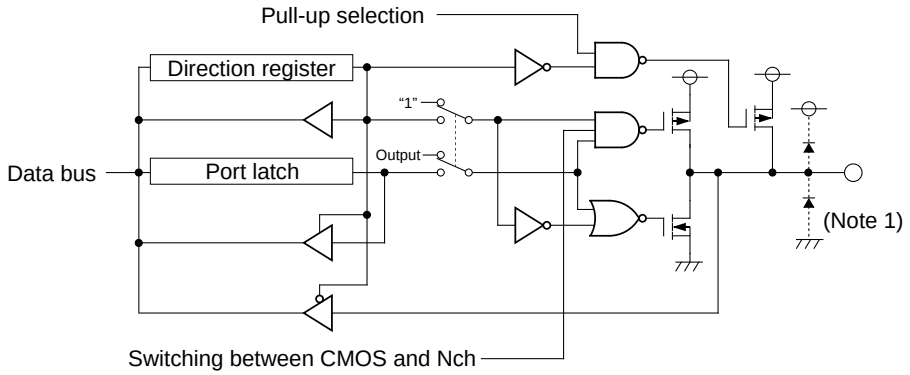

Note 1: ····|◄···· symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

**Figure 17.2. I/O Ports (2)**

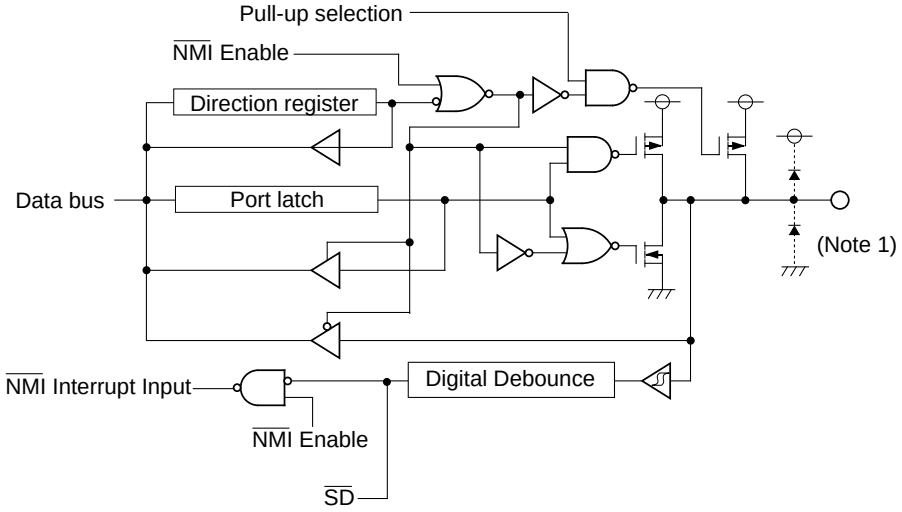
P62, P66




P63, P67

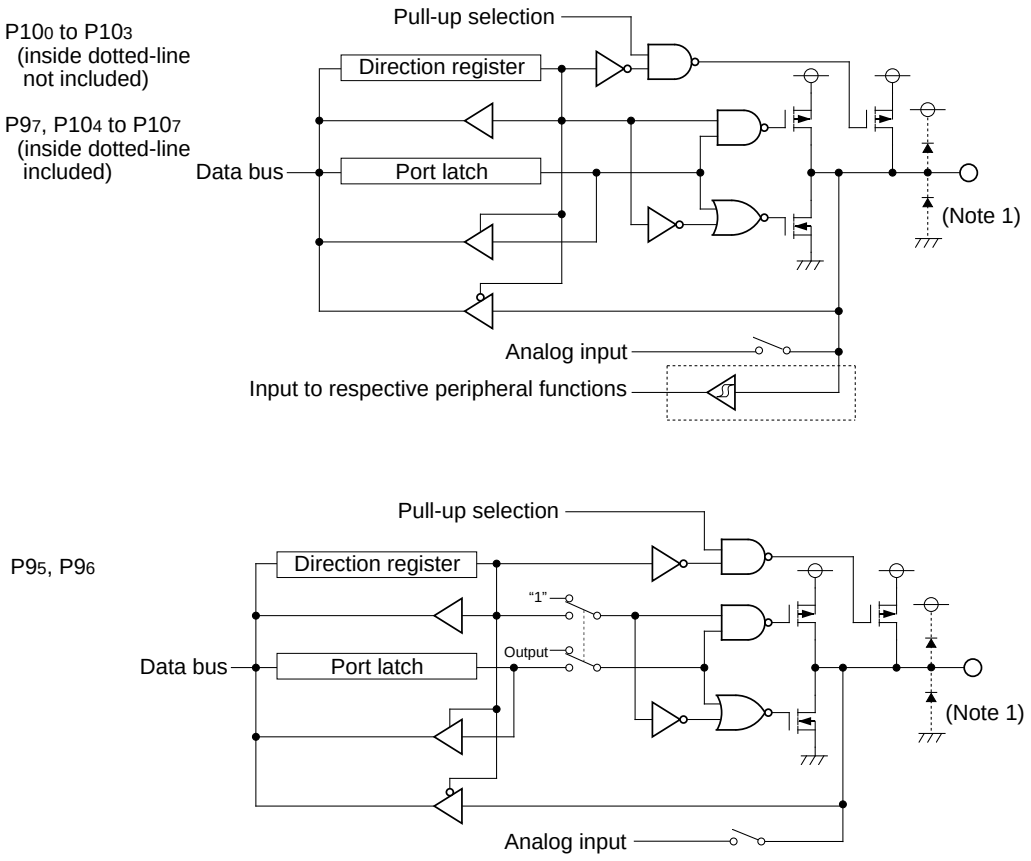



P85




Note 1: ····|◄····· symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

**Figure 17.3. I/O Ports (3)**




Note 1: ⋯◂⋯ symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

**Figure 17.4. I/O Ports (4)**

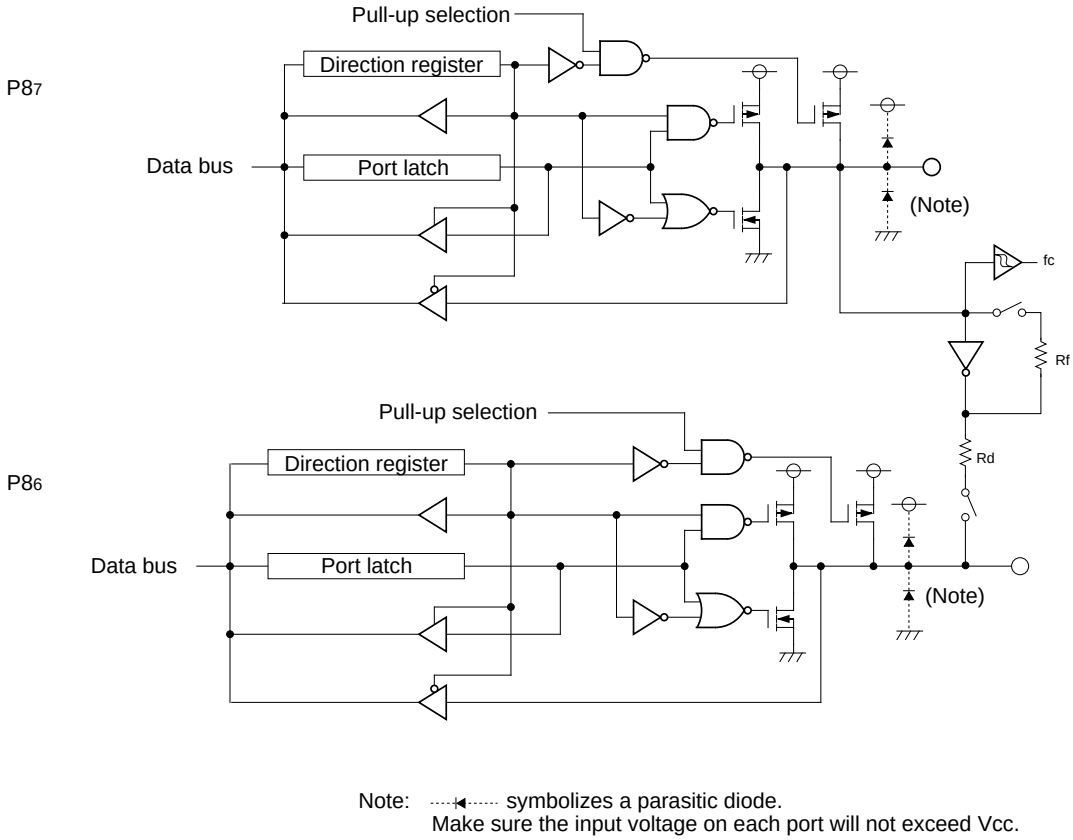



**Figure 17.5. I/O Ports (5)**

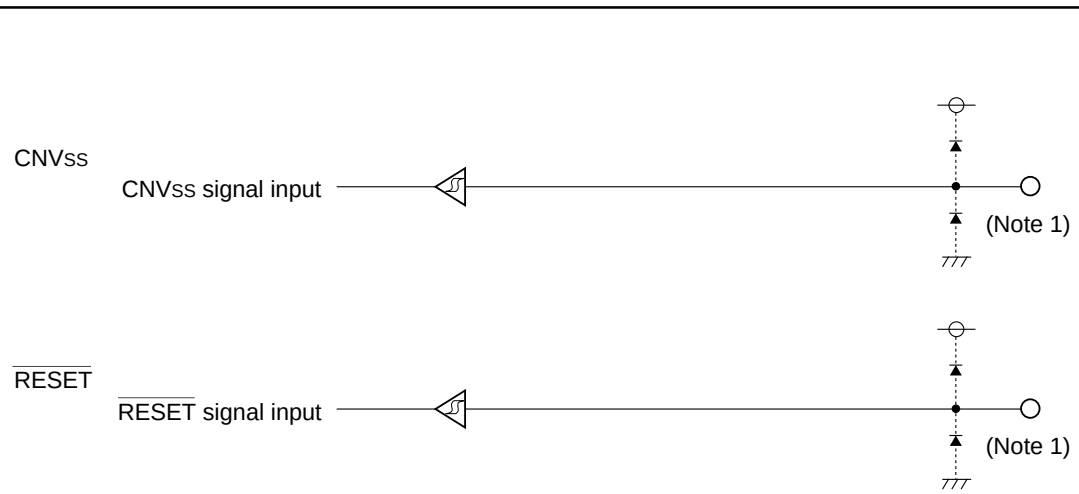
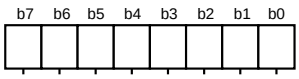



**Figure 17.6. I/O Pins**

## Port Pi direction register (i=0 to 3, 6 to 8, and 10) (Note)

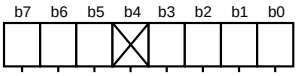



| Symbol | Address | After reset |
|---|---|---|
| PD0 to PD3 | $03E2_{16}$, $03E3_{16}$, $03E6_{16}$, $03E7_{16}$ | $00_{16}$ |
| PD6 to PD8 | $03EE_{16}$, $03EF_{16}$, $03F2_{16}$ | $00_{16}$ |
| PD10 | $03F6_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PDi_0 | Port $Pi_0$ direction bit | 0 : Input mode (Functions as an input port)<br>1 : Output mode (Functions as an output port)<br>(i = 0 to 3, 6 to 8, and 10) | RW |
| PDi_1 | Port $Pi_1$ direction bit | | RW |
| PDi_2 | Port $Pi_2$ direction bit | | RW |
| PDi_3 | Port $Pi_3$ direction bit | | RW |
| PDi_4 | Port $Pi_4$ direction bit | | RW |
| PDi_5 | Port $Pi_5$ direction bit | | RW |
| PDi_6 | Port $Pi_6$ direction bit | | RW |
| PDi_7 | Port $Pi_7$ direction bit | | RW |

Note: Ports must be enabled using the PACR
In 80 pin version set PACR2, PACR1, PACR0 to "$011_2$"
In 64 pin version set PACR2, PACR1, PACR0 to "$010_2$"

## Port P9 direction register (Note 1,2)



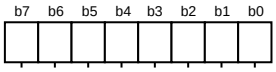

| Symbol | Address | After reset |
|---|---|---|
| PD9 | $03F3_{16}$ | $00000000_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PD9_0 | Port $P9_0$ direction bit | 0 : Input mode (Functions as an input port)<br>1 : Output mode (Functions as an output port) | RW |
| PD9_1 | Port $P9_1$ direction bit | | RW |
| PD9_2 | Port $P9_2$ direction bit | | RW |
| PD9_3 | Port $P9_3$ direction bit | | RW |
| $\overline{\text{(b4)}}$ | Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate. | | — |
| PD9_5 | Port $P9_5$ direction bit | 0 : Input mode (Functions as an input port)<br>1 : Output mode (Functions as an output port) | RW |
| PD9_6 | Port $P9_6$ direction bit | | RW |
| PD9_7 | Port $P9_7$ direction bit | | RW |

Note 1: Make sure the PD9 register is written to by the next instruction after setting the PRCR register's PRC2 bit to "1"(write enabled).
Note 2: Ports must be enabled using the PACR
In 80 pin version set PACR2, PACR1, PACR0 to "$011_2$"
In 64 pin version set PACR2, PACR1, PACR0 to "$010_2$"

**Figure 17.1.1. PD0, PD1, PD2, PD3, PD6, PD7, PD8, PD9, and PD10 Registers**

Port Pi register (i=0 to 3, 6 to 8 and 10) (Note)

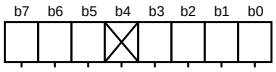



| Symbol | Address | After reset |
|---|---|---|
| P0 to P3 | $03E0_{16}$, $03E1_{16}$, $03E4_{16}$, $03E5_{16}$ | Indeterminate |
| P6 to P8 | $03EC_{16}$, $03ED_{16}$, $03F0_{16}$ | Indeterminate |
| P10 | $03F4_{16}$ | Indeterminate |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| Pi_0 | Port $Pi_0$ bit | The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register.<br>The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register<br>0 : "L" level<br>1 : "H" level (Note 1)<br>(i = 0 to 3, 6 to 8 and 10) | RW |
| Pi_1 | Port $Pi_1$ bit | | RW |
| Pi_2 | Port $Pi_2$ bit | | RW |
| Pi_3 | Port $Pi_3$ bit | | RW |
| Pi_4 | Port $Pi_4$ bit | | RW |
| Pi_5 | Port $Pi_5$ bit | | RW |
| Pi_6 | Port $Pi_6$ bit | | RW |
| Pi_7 | Port $Pi_7$ bit | | RW |

Note: Ports must be enabled using the PACR
In 80 pin version set PACR2, PACR1, PACR0 to "$011_2$"
In 64 pin version set PACR2, PACR1, PACR0 to "$010_2$"

Port P9 register (Note1)



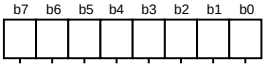

| Symbol | Address | After reset |
|---|---|---|
| P9 | $03F1_{16}$ | Indeterminate |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| P9_0 | Port $P9_0$ bit | The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register.<br>The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register (except for $P8_5$)<br>0 : "L" level<br>1 : "H" level | RW |
| P9_1 | Port $P9_1$ bit | | RW |
| P9_2 | Port $P9_2$ bit | | RW |
| P9_3 | Port $P9_3$ bit | | RW |
| (b4) | Nothing is assigned (Note 2) | | - |
| P9_5 | Port $P9_5$ bit | | RW |
| P9_6 | Port $P9_6$ bit | | RW |
| P9_7 | Port $P9_7$ bit | | RW |

Note1: Ports must be enabled using the PACR
In 80 pin version set PACR2, PACR1, PACR0 to "$011_2$"
In 64 pin version set PACR2, PACR1, PACR0 to "$010_2$"
Note2: Nothing is assigned. In an attempt to write t o this bit, write "0".
The value if read turns out to be "0".

**Figure 17.2.1. P0, P1, P2, P3, P6, P7, P8, P9, and P10 Registers**

Pull-up control register 0 (Note)

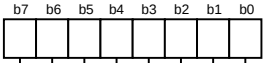



| Symbol | Address | After reset |
|---|---|---|
| PUR0 | 03FC16 | 0016 |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PU00 | P00 to P03 pull-up | 0 : Not pulled high<br>1 : Pulled high (Note) | RW |
| PU01 | P04 to P07 pull-up | | RW |
| PU02 | P10 to P13 pull-up | | RW |
| PU03 | P14 to P17 pull-up | | RW |
| PU04 | P20 to P23 pull-up | | RW |
| PU05 | P24 to P27 pull-up | | RW |
| PU06 | P30 to P33 pull-up | | RW |
| PU07 | P34 to P37 pull-up | | RW |

Note : The pin for which this bit is “1” (pulled high) and the direction bit is “0” (input mode) is pulled high.

Pull-up control register 1

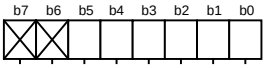



| Symbol | Address | After reset(Note 5) |
|---|---|---|
| PUR1 | 03FD16 | 000000002 |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| ——<br>(b3-b0) | Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”. | | — |
| PU14 | P60 to P63 pull-up | 0 : Not pulled high<br>1 : Pulled high (Note) | RW |
| PU15 | P64 to P67 pull-up | | RW |
| PU16 | P70 to P73 pull-up | | RW |
| PU17 | P74 to P77 pull-up | | RW |

Note : The pin for which this bit is “1” (pulled high) and the direction bit is “0” (input mode) is pulled high.

Pull-up control register 2



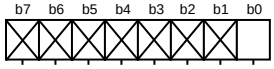

| Symbol | Address | After reset |
|---|---|---|
| PUR2 | 03FE16 | 0016 |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PU20 | P80 to P83 pull-up | 0 : Not pulled high<br>1 : Pulled high (Note) | RW |
| PU21 | P84 to P87 pull-up | | RW |
| PU22 | P90 to P93 pull-up | | RW |
| PU23 | P95 to P97 pull-up | | RW |
| PU24 | P100 to P103 pull-up | | RW |
| PU25 | P104 to P107 pull-up | | RW |
| ——<br>(b7-b6) | Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”. | | — |

Note : The pin for which this bit is “1” (pulled high) and the direction bit is “0” (input mode) is pulled high.

**Figure 17.3.1. PUR0 to PUR2 Registers**

Port control register

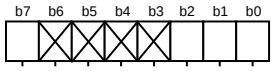



| Symbpl | Address | After reset |
|---|---|---|
| PCR | $03FF_{16}$ | $00_{16}$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PCR0 | Port P1 control bit | Operation performed when the P1 register is read<br>0: When the port is set for input, the input levels of P10 to P17 pins are read. When set for output, the port latch is read.<br>1: The port latch is read regardless of whether the port is set for input or output. | RW |
| —— (b7-b1) | Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”. | | — |

**Figure 17.4.1. PCR Register**

Pin assignment control register (Note)




| Symbpl | Address | After reset |
|---|---|---|
| PACR | $025D_{16}$ | $00000000_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| PACR0 | Pin enabling bit | 010 : 64 pin<br>011 : 80 pin<br>All other values are reserved. Do not use. | RW |
| PACR1 | | | RW |
| PACR2 | | | RW |
| —— (b6-b3) | Reserved bits | Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”. | — |
| U1MAP | UART1 pin remapping bit | UART1 pins assigned to<br>0 : $P6_7$ to $P6_4$<br>1 : $P7_3$ to $P7_0$ | RW |

Note : Set bit 2 of protect register (address $000A_{16}$) to "1" before writing to PACR.

**Figure 17.5.1. PACR Register**

NMI digital debounce register (Note)

| b7 | b0 | Symbol | Address | After reset |
|---|---|---|---|---|
| | | NDDR | $033E_{16}$ | $FF_{16}$ |

| Function | Setting range | RW |
|---|---|---|
| Assuming that set value =n,<br>for n = 0 to FEh, $\overline{\text{NMI}}$ / $\overline{\text{SD}}$ pulse whose width is greater than (V1/8) / ( n + 1) will be input.<br>For n = FFh, the digital debounce filter is disabled.<br>All signals are input | $00_{16}$~$FF_{16}$ | RW |

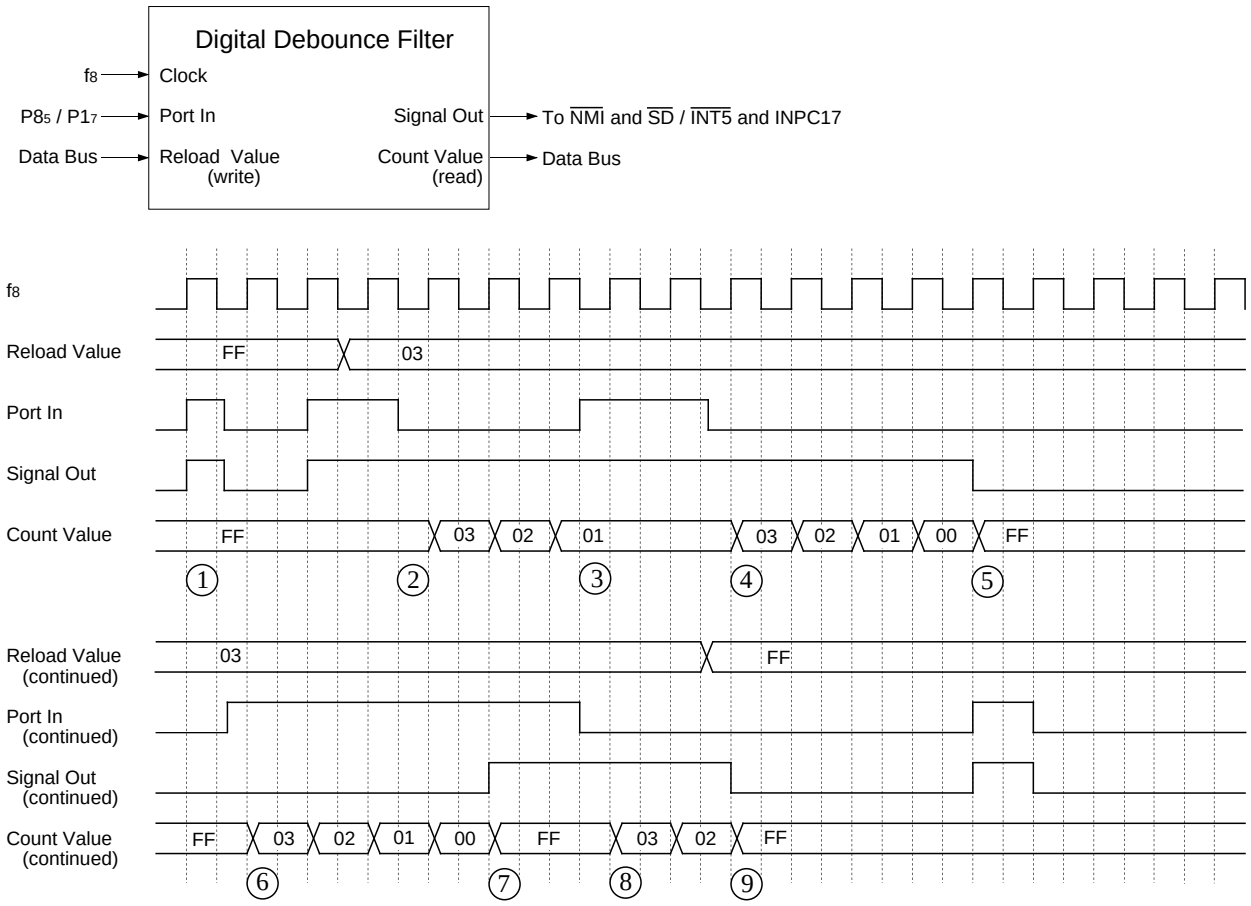
Note : Set bit 2 of protect register (Address $000A_{16}$) to "1" before writing to NDDR.

$P1_7$ digital debounce register

| b7 | b0 | Symbol | Address | After reset |
|---|---|---|---|---|
| | | P17DDR | $033F_{16}$ | $FF_{16}$ |

| Function | Setting range | RW |
|---|---|---|
| Assuming that set value =n,<br>for n = 0 to FEh, INPC17 / $\overline{\text{INT5}}$ pulse whose width is greater than (V1/8) / ( n + 1) will be input.<br>For n = FFh, the digital debounce filter is disabled.<br>All signals are input | $00_{16}$~$FF_{16}$ | RW |

**Figure 17.6.1. NDDR and P17DDR Registers**




1. (Condition after reset). Reload = FF, Port In = signal Out continuosly.
2. Reload = 03. At edge of Port In != Signal Out, Counter gets Reload Value and stats counting down.
3. Port In = Signal Out, counting stops.
4. At edge of Port In != Signal Out, Counter gets Reload Value and starts counting.

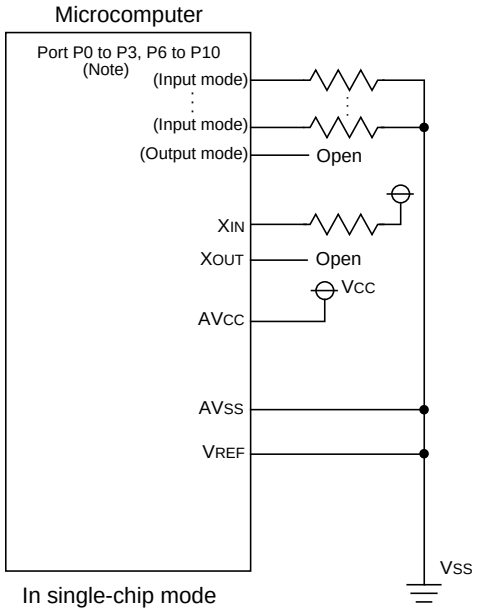
5. Counter underflows, stops, and Port In is driven to Signal Out.
6. At edge of Port In != Signal Out, counter gets Reload Value and starts counting.
7. Counter underflows, stops, and Port In is driven to Signal Out.
8. At edge of Port In != Signal Out, counter gets Reload Value and starts counting.
9. FF is written to Reload Value. Counter is stopped and loaded with FF.
   Port In = Signal Out continuously.

**Figure 17.6.2. Functioning of Digital Debounce Filter**

**Table 17.1. Unassigned Pin Handling in Single-chip Mode**

| Pin name | Connection |
|---|---|
| Ports P0 to P3, P6 to P10 | After setting for input mode, connect every pin to VSS via a resistor(pull-down); or after setting for output mode, leave these pins open. (Note 1, Note 2, Note 4) |
| XOUT (Note 3) | Open |
| Xin | Connect via resistor to VCC (pull-up) |
| AVCC | Connect to VCC |
| AVSS, VREF | Connect to VSS |

Note 1: When setting the port for output mode and leave it open, be aware that the port remains in input mode until it is switched to output mode in a program after reset. For this reason, the voltage level on the pin becomes indeterminate, causing the power supply current to increase while the port remains in input mode.
Futhermore, by considering a possibility that the contents of the direction registers could be changed by noise or noise-induced runaway, it is recommended that the contents of the directionregisters be periodically reset in software, for the increased reliability of the program.
Note 2: Make sure the unused pins are processed with the shortest possible wiring from the microcomputer pins (within 2 cm).
Note 3: With external clock or VCC input to XIN pin.
Note 4: When using the 80pin version, set PACR2, PACR1, PACR0 to "011$_2$".
When using the 64pin version, set PACR2, PACR1, PACR0 to "010$_2$".




Note : when using the 64pin version, set PACR2, PACR1, PACR0 to "010$_2$".

**Figure 17.7. Unassigned Pins Handling**

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

# 18. Electrical Characteristics

## 18.1. Normal version

**Table 18.1. Absolute Maximum Ratings**

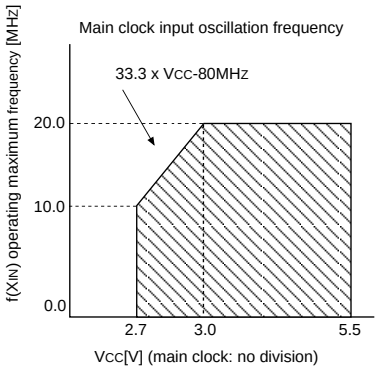
| Symbol | Parameter | | Condition | Rated value | Unit |
|---|---|---|---|---|---|
| $V_{CC}$ | Supply voltage | | $V_{CC}$=AV$_{CC}$ | -0.3 to 6.5 | V |
| AV$_{CC}$ | Analog supply voltage | | $V_{CC}$=AV$_{CC}$ | -0.3 to 6.5 | V |
| $V_I$ | Input voltage | P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$, $X_{IN}$, $V_{REF}$, $\overline{RESET}$, $CNV_{SS}$ | | -0.3 to $V_{CC}$+0.3 | V |
| $V_O$ | Output voltage | P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$, $X_{OUT}$ | | -0.3 to $V_{CC}$+0.3 | V |
| $P_d$ | Power dissipation | | Topr=25 °C | 300 | mW |
| $T_{opr}$ | Operating ambient temperature | | | -20 to 85 / -40 to 85 | °C |
| $T_{stg}$ | Storage temperature | | | -65 to 150 | °C |

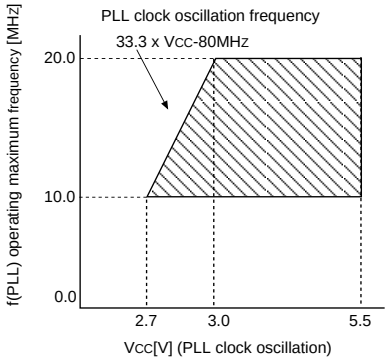
**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

## Table 18.2. Recommended Operating Conditions (Note 1)

| Symbol | Parameter | | Standard | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $V_{CC}$ | Supply voltage | | 2.7 | | 5.5 | V |
| AVcc | Analog supply voltage | | | $V_{CC}$ | | V |
| Vss | Supply voltage | | | 0 | | V |
| AVss | Analog supply voltage | | | 0 | | V |
| $V_{IH}$ | HIGH input voltage | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, $X_{IN}$, $\overline{RESET}$, $CNV_{SS}$ | 0.7$V_{CC}$ | | $V_{CC}$ | V |
| $V_{IL}$ | LOW input voltage | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, $X_{IN}$, $\overline{RESET}$, $CNV_{SS}$ | 0 | | 0.3$V_{CC}$ | V |
| $I_{OH\ (peak)}$ | HIGH peak output current | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | | -10.0 | mA |
| $I_{OH\ (avg)}$ | HIGH average output current | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | | -5.0 | mA |
| $I_{OL\ (peak)}$ | LOW peak output current | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | | 10.0 | mA |
| $I_{OL\ (avg)}$ | LOW average output current | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | | 5.0 | mA |
| f ($X_{IN}$) | Main clock input oscillation frequency (Note 4) | $V_{CC}$=3.0 to 5.5V | 0 | | 20 | MHz |
| | | $V_{CC}$=2.7 to 3.0V | 0 | | 33 X $V_{CC}$-80 | MHz |
| f ($X_{CIN}$) | Sub-clock oscillation frequency | | | 32.768 | 50 | kHz |
| $f_1$ (ROC) | Ring oscillation frequency 1 | | | 1 | | MHz |
| $f_2$ (ROC) | Ring oscillation frequency 2 | | | 2 | | MHz |
| $f_3$ (ROC) | Ring oscillation frequency 3 | | | 16 | | MHz |
| f (PLL) | PLL clock oscillation frequency (Note 4) | $V_{CC}$=3.0 to 5.5V | 10 | | 20 | MHz |
| | | $V_{CC}$=2.7 to 3.0V | 10 | | 33 X $V_{CC}$-80 | MHz |
| f (BCLK) | CPU operation clock | | 0 | | 20 | MHz |
| $T_{SU}$(PLL) | PLL frequency synthesizer stabilization wait time | $V_{CC}$=5.0V | | | 20 | ms |
| | | $V_{CC}$=3.0V | | | 50 | ms |

Note 1: Referenced to $V_{CC}$ = 2.7 to 5.5V at Topr = -20 to 85 °C / -40 to 85 °C unless otherwise specified.
Note 2: The mean output current is the mean value within 100ms.
Note 3: The total $I_{OL(peak)}$ for all ports must be 80mA max. The total $I_{OH(peak)}$ for all ports must be -80mA max.
Note 4: Relationship between main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.



**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

### Table 18.3. A-D Conversion Characteristics (Note 1)

| Symbol | Parameter | | Measuring condition | | Standard Min. | Standard Typ. | Standard Max. | Unit |
|---|---|---|---|---|---|---|---|---|
| – | Resolution | | $V_{REF}$ =$V_{CC}$ | | | | 10 | Bits |
| INL | Integral non-linearity error | 10 bit | $V_{REF}$ =$V_{CC}$=5V | $AN_0$ to $AN_7$ input | | | ±3 | LSB |
| | | | | $AN_{00}$ to $AN_{07}$, $AN_{20}$ to $AN_{27}$ input | | | ±7 | LSB |
| | | | $V_{REF}$ =$V_{CC}$=3.3V | $AN_0$ to $AN_7$ input | | | ±5 | LSB |
| | | | | $AN_{00}$ to $AN_{07}$, $AN_{20}$ to $AN_{27}$ input | | | ±7 | LSB |
| | | 8 bit | $V_{REF}$ =$V_{CC}$=3.3V | $AN_0$ to $AN_7$ input | | | ±2 | LSB |
| – | Absolute accuracy | 10 bit | $V_{REF}$ =$V_{CC}$=5V | $AN_0$ to $AN_7$ input | | | ±3 | LSB |
| | | | | $AN_{00}$ to $AN_{07}$, $AN_{20}$ to $AN_{27}$ input | | | ±7 | LSB |
| | | | $V_{REF}$ =$V_{CC}$=3.3V | $AN_0$ to $AN_7$ input | | | ±5 | LSB |
| | | | | $AN_{00}$ to $AN_{07}$, $AN_{20}$ to $AN_{27}$ input | | | ±7 | LSB |
| | | 8 bit | $V_{REF}$ =$V_{CC}$=3.3V | $AN_0$ to $AN_7$ input | | | ±2 | LSB |
| DNL | Differential non-linearity error | | | | | | ±1 | LSB |
| – | Offset error | | | | | | ±3 | LSB |
| – | Gain error | | | | | | ±3 | LSB |
| $R_{LADDER}$ | Ladder resistance | | $V_{REF}$ =$V_{CC}$ | | 10 | | 40 | kΩ |
| $t_{CONV}$ | Conversion time(10bit), Sample & hold function available | | $V_{REF}$ =$V_{CC}$=5V, $ø_{AD}$=10MHz | | 3.3 | | | µs |
| $t_{CONV}$ | Conversion time(8bit), Sample & hold function available | | $V_{REF}$ =$V_{CC}$=5V, $ø_{AD}$=10MHz | | 2.8 | | | µs |
| $t_{SAMP}$ | Sampling time | | | | 0.3 | | | µs |
| $V_{REF}$ | Reference voltage | | | | 2.0 | | $V_{CC}$ | V |
| $V_{IA}$ | Analog input voltage | | | | 0 | | $V_{REF}$ | V |

Note 1: Referenced to $V_{CC}$=$AV_{CC}$=$V_{REF}$=3.3 to 5.5V, $V_{SS}$=$AV_{SS}$=0V at Topr = -20 to 85 °C / -40 to 85 °C unless otherwise specified.
Note 2: AD operation clock frequency ($ø_{AD}$ frequency) must be 10 MHz or less. And divide the $f_{AD}$ if $V_{CC}$ is less than 4.2V, and make $ø_{AD}$ frequency equal to or lower than $f_{AD}$/2.
Note 3: A case without sample & hold function turn $ø_{AD}$ frequency into 250 kHz or more in addition to a limit of Note 2.
A case with sample & hold function turn $ø_{AD}$ frequency into 1MHz or more in addition to a limit of Note 2.

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

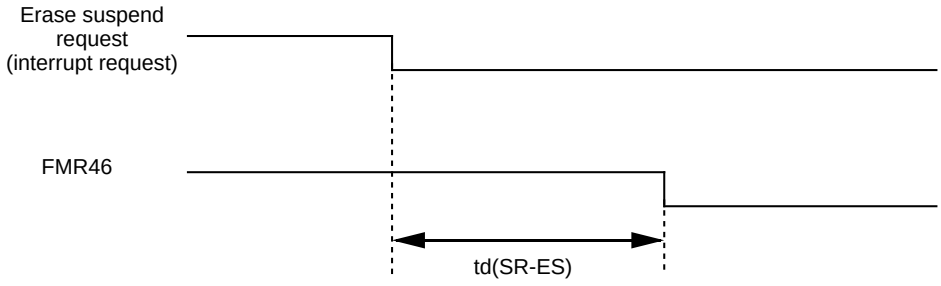
**Table 18.4. Flash Memory Version Electrical Characteristics (Note 1)**

| Symbol | Parameter | | Standard | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. (Note 2) | Max | |
| – | Erase/Write cycle (Note 3) | | 100(Note 4) | | | cycle |
| – | Word program time (Vcc=5.0V, Topr=25°C) | | | 75 | 600 | µs |
| – | Block erase time | 2Kbyte block | | 0.2 | 9 | s |
| | | 8Kbyte block | | 0.4 | 9 | s |
| | | 16Kbyte block | | 0.7 | 9 | s |
| | | 32Kbyte block | | 1.2 | 9 | s |
| td(SR-ES) | Time delay from Suspend Request until Erase Suspend | | | | 20 | ms |
| – | Data retention time (Note 5) | | 20 | | | year |

**Table 18.5. Flash Memory Version Electrical Characteristics (Note 6) 10000 E/W cycle products (Option) [blockA and block B(Note 7)]**

| Symbol | Parameter | Standard | | | Unit |
|---|---|---|---|---|---|
| | | Min. | Typ. (Note 2) | Max | |
| – | Erase/Write cycle (Note 3, 8, 9) | 10000(Note 4,10) | | | cycle |
| – | Word program time (Vcc=5.0V, Topr=25°C) | | 100 | | µs |
| – | Block erase time(Vcc=5.0V, Topr=25°C) (2Kbyte block) | | 0.3 | 9 | s |
| td(SR-ES) | Time delay from Suspend Request until Erase Suspend | | | 20 | ms |

Note 1: When not otherwise specified, Vcc = 2.7 to5.5V; Topr = 0 to 60 °C.
Note 2: VCC = 5V; TOPR = 25 °C.
Note 3: Definition of E/W cycle: Each block may be written to a variable number of times - up to a maximum of the total number of distinct word addresses - for every block erase. Performing multiple writes to the same address before an erase operation is prohibited.
Note 4: Maximum number of E/W cycles for which opration is guaranteed.
Note 5: Topr = 55°C.
Note 6: When not otherwise specified, Vcc = 2.7 to 5.5V; Topr = -20 to 85°C / -40 to 85°C (Option).
Note 7: Table18.5 applies for Block A or B E/W cycles > 1000. Otherwise, use Table 18.4.
Note 8: To reduce the number of E/W cycles, a block erase should ideally be performed after writing as many different word addresses (only one time each) as possible. It is important to track the total number of block erases.
Note 9: Should erase error occur during block erase, attempt to execute clear status register command, then clock erase command at least three times until erase error disappears.
Note 10: When Block A or B E/W cycles exceed 100 (Option), select one wait state per block access. When FMR17 is set to "1", one wait state is inserted per access to Block A or B - regardless of the value of PM17. Wait state insertion during access to all other blocks, as well as to internal RAM, is controlled by PM17 - regardless of the setting of



**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

### Table 18.6. Low Voltage Detection Circuit Electrical Characteristics (Note 1, Note 3)

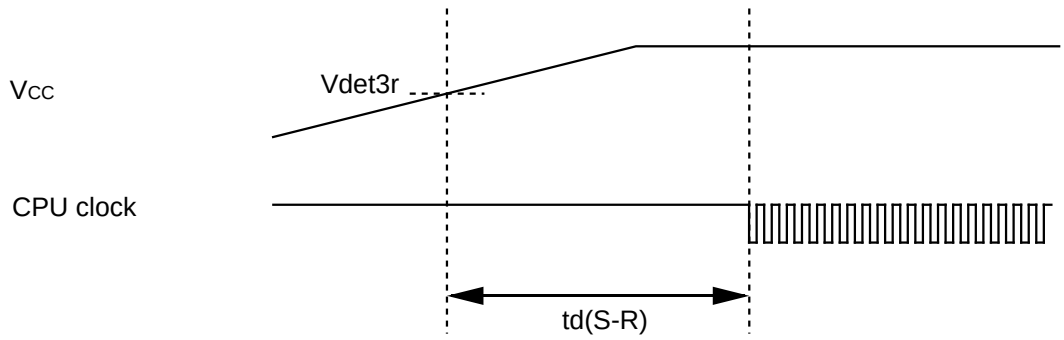
| Symbol | Parameter | Measuring condition | Standard | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| Vdet4 | Voltage down detection voltage (Note 1) | Vcc=0.8 to 5.5V | T.B.D | T.B.D | T.B.D | V |
| Vdet3 | Reset level detection voltage (Notes 1) | | T.B.D | T.B.D | T.B.D | V |
| Vdet3s | Low voltage reset retention voltage (Note 2) | | T.B.D | T.B.D | T.B.D | V |
| Vdet3r | Low voltage reset release voltage | | T.B.D | T.B.D | T.B.D | V |

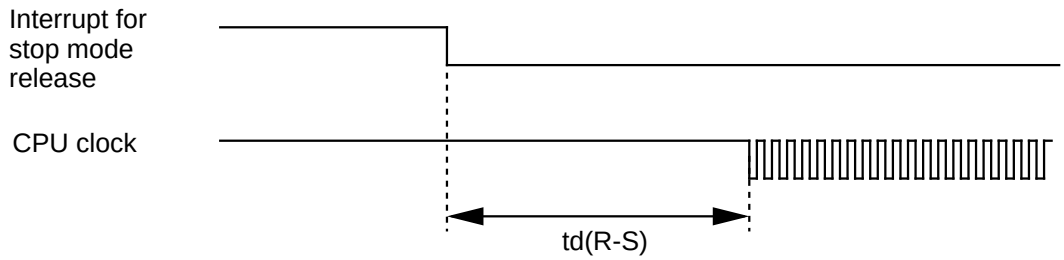
Note 1: Vdet4 > Vdet3
Note 2: Vdet3s is the min voltage at which "hardware reset 2" is maintained.
Note 3: The low voltage detection circuit is designed to use when Vcc is set to 5V.

### Table 18.7. Power Supply Circuit Timing Characteristics

| Symbol | Parameter | Measuring condition | Standard | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| td(P-R) | Time for internal power supply stabilization during powering-on | Vcc=2.7 to 5.5V | | | 2 | ms |
| td(ROC) | Time for internal ring oscillator stabilization during powering-on | | T.B.D | T.B.D | T.B.D | ms |
| td(R-S) | STOP release time (Note 2) | | | | 2 | ms |
| td(W-S) | Low power dissipation mode wait mode release time (Note 2) | | | | 150 | µs |
| td(M-L) | Time for internal power supply stabilization when main clock oscillation starts | | | | 50 | µs |
| td(S-R) | Hardware reset 2 release wait time | Vcc=Vdet3r to 5.5V | | 6 (Note 1) | 20 | ms |
| td(E-A) | Low voltage detection circuit operation start time (Note 3) | Vcc=2.7 to 5.5V | | | 20 | µs |

Note 1: When Vcc = 5V
Note 2: This is the time between interrupt for (STOP/WAIT) mode release and resumption of CPU clock operation.
Note 3: After enabling low voltage detection, this time is required before proper detection can occur.



**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 5V

## Table 18.8. Electrical Characteristics (Note 1)

| Symbol | Parameter | | Measuring condition | Standard | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| $V_{OH}$ | HIGH output voltage | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | $I_{OH}$=-5mA | $V_{CC}$-2.0 | | $V_{CC}$ | V |
| $V_{OH}$ | HIGH output voltage | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | $I_{OH}$=-200µA | $V_{CC}$-0.3 | | $V_{CC}$ | V |
| $V_{OH}$ | HIGH output voltage $X_{OUT}$ | HIGHPOWER | $I_{OH}$=-1mA | $V_{CC}$-2.0 | | $V_{CC}$ | V |
| | | LOWPOWER | $I_{OH}$=-0.5mA | $V_{CC}$-2.0 | | $V_{CC}$ | |
| | HIGH output voltage $X_{COUT}$ | HIGHPOWER | With no load applied | | 2.5 | | V |
| | | LOWPOWER | With no load applied | | 1.6 | | |
| $V_{OL}$ | LOW output voltage | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | $I_{OL}$=5mA | | | 2.0 | V |
| $V_{OL}$ | LOW output voltage | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | $I_{OL}$=200µA | | | 0.45 | V |
| $V_{OL}$ | LOW output voltage $X_{OUT}$ | HIGHPOWER | $I_{OL}$=1mA | | | 2.0 | V |
| | | LOWPOWER | $I_{OL}$=0.5mA | | | 2.0 | |
| | LOW output voltage $X_{COUT}$ | HIGHPOWER | With no load applied | | 0 | | V |
| | | LOWPOWER | With no load applied | | 0 | | |
| $V_{T+}$-$V_{T-}$ | Hysteresis | $TA0_{IN}$ to $TA4_{IN}$, $TB0_{IN}$ to $TB2_{IN}$, $\overline{INT0}$ to $\overline{INT5}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS0}$ to $\overline{CTS2}$, SCL, SDA, $CLK0$ to $CLK2$, $TA2_{OUT}$ to $TA4_{OUT}$, $\overline{KI0}$ to $\overline{KI3}$, $RxD0$ to $RxD2$, $SIN3$, $SIN4$ | | 0.2 | | 1.0 | V |
| $V_{T+}$-$V_{T-}$ | Hysteresis | $\overline{RESET}$ | | 0.2 | | 2.5 | V |
| $I_{IH}$ | HIGH input current | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, $X_{IN}$, $\overline{RESET}$, CNVss | $V_I$=5V | | | 5.0 | µA |
| $I_{IL}$ | LOW input current | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, $X_{IN}$, $\overline{RESET}$, CNVss | $V_I$=0V | | | -5.0 | µA |
| $R_{PULLUP}$ | Pull-up resistance | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | $V_I$=0V | 30 | 50 | 170 | kΩ |
| $R_{fXIN}$ | Feedback resistance $X_{IN}$ | | | | 1.5 | | MΩ |
| $R_{fXCIN}$ | Feedback resistance $X_{CIN}$ | | | | 15 | | MΩ |
| $V_{RAM}$ | RAM retention voltage | | At stop mode | 2.0 | | | V |

Note 1: Referenced to $V_{CC}$=4.2 to 5.5V, $V_{SS}$=0V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK)=20MHz unless otherwise specified.

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 5V

### Table 18.9. Electrical Characteristics (2) (Note 1)

| Symbol | Parameter | | Measuring condition | | Standard Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|---|
| $I_{CC}$ | Power supply current ($V_{CC}$=3.0 to 5.5V) | The output pins are open and other pins are $V_{SS}$ | Flash memory | $f(X_{IN})$=20MHz, No division | | 16 | 19 | mA |
| | | | | No division, Ring oscillation | | 1 | | mA |
| | | | Flash memory Program | f(BCLK)=10MHz, $V_{CC}$=5.0V | | T.B.D | | mA |
| | | | Flash memory Erase | f(BCLK)=10MHz, $V_{CC}$=5.0V | | T.B.D | | mA |
| | | | Flash memory | f(BCLK)=32kHz, Low power dissipation mode, RAM(Note 3) | | 25 | | µA |
| | | | | f(BCLK)=32kHz Low power dissipation mode, Flash memory(Note 3) | | 500 | | µA |
| | | | | Ring oscillation, Wait mode | | T.B.D | | µA |
| | | | Flash memory | f(BCLK)=32kHz, Wait mode (Note 2), Oscillation capacity High | | 12 | | µA |
| | | | | f(BCLK)=32kHz, Wait mode(Note 2), Oscillation capacity Low | | T.B.D | | µA |
| | | | | Stop mode, $T_{opr}$=25°C | | 0.8 | 3 | µA |
| Idet4 | Voltage down detection dissipation current (Note 4) | | | | | T.B.D | T.B.D | µA |
| Idet3 | Reset area detection dissipation current (Note 4) | | | | | T.B.D | T.B.D | µA |
| Idet2 | RAM retention limit detection dissipation current (Note 4) | | | | | T.B.D | T.B.D | µA |

Note 1: Referenced to $V_{CC}$=3.0 to 5.5V, $V_{SS}$=0V at Topr = -20 to 85 °C / -40 to 85 °C, $f(X_{IN})$=20MHz unless otherwise specified.
Note 2: With one timer operated using $f_{C32}$.
Note 3: This indicates the memory in which the program to be executed exists.
Note 4: Idet is dissipation current when the following bit is set to "1" (detection circuit enabled).
Idet4: VC27 bit of VCR2 register
Idet3: VC26 bit of VCR2 register
Idet2: VC25 bit of VCR2 register

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 5V

### Timing Requirements

**($V_{CC}$ = 5V, $V_{SS}$ = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)**

**Table 18.10. External Clock Input ($X_{IN}$ input)**

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_c$ | External clock input cycle time | 50 | | ns |
| $t_{w(H)}$ | External clock input HIGH pulse width | 20 | | ns |
| $t_{w(L)}$ | External clock input LOW pulse width | 20 | | ns |
| $t_r$ | External clock rise time | | 9 | ns |
| $t_f$ | External clock fall time | | 9 | ns |

**These standards are not final and focused only normal-version.
Should be used as a reference.**

$V_{CC}$ = 5V

**Timing Requirements**

**($V_{CC}$ = 5V, $V_{SS}$ = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)**

**Table 18.11. Timer A Input (Counter Input in Event Counter Mode)**

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TAi_{IN}$ input cycle time | 100 | | ns |
| $t_{w(TAH)}$ | $TAi_{IN}$ input HIGH pulse width | 40 | | ns |
| $t_{w(TAL)}$ | $TAi_{IN}$ input LOW pulse width | 40 | | ns |

**Table 18.12. Timer A Input (Gating Input in Timer Mode)**

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TAi_{IN}$ input cycle time | 400 | | ns |
| $t_{w(TAH)}$ | $TAi_{IN}$ input HIGH pulse width | 200 | | ns |
| $t_{w(TAL)}$ | $TAi_{IN}$ input LOW pulse width | 200 | | ns |

**Table 18.13. Timer A Input (External Trigger Input in One-shot Timer Mode)**

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TAi_{IN}$ input cycle time | 200 | | ns |
| $t_{w(TAH)}$ | $TAi_{IN}$ input HIGH pulse width | 100 | | ns |
| $t_{w(TAL)}$ | $TAi_{IN}$ input LOW pulse width | 100 | | ns |

**Table 18.14. Timer A Input (External Trigger Input in Pulse Width Modulation Mode)**

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{w(TAH)}$ | $TAi_{IN}$ input HIGH pulse width | 100 | | ns |
| $t_{w(TAL)}$ | $TAi_{IN}$ input LOW pulse width | 100 | | ns |

**Table 18.15. Timer A Input (Counter Increment/decrement Input in Event Counter Mode)**

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(UP)}$ | $TAi_{OUT}$ input cycle time | 2000 | | ns |
| $t_{w(UPH)}$ | $TAi_{OUT}$ input HIGH pulse width | 1000 | | ns |
| $t_{w(UPL)}$ | $TAi_{OUT}$ input LOW pulse width | 1000 | | ns |
| $t_{su(UP-TIN)}$ | $TAi_{OUT}$ input setup time | 400 | | ns |
| $t_{h(TIN-UP)}$ | $TAi_{OUT}$ input hold time | 400 | | ns |

**Table 18.16. Timer A Input (Two-phase Pulse Input in Event Counter Mode)**

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TAi_{IN}$ input cycle time | 800 | | ns |
| $t_{su(TAIN-TAOUT)}$ | $TAi_{OUT}$ input setup time | 200 | | ns |
| $t_{su(TAOUT-TAIN)}$ | $TAi_{IN}$ input setup time | 200 | | ns |

**These standards are not final and focused only normal-version.
Should be used as a reference.**

$V_{CC}$ = 5V

## Timing Requirements

**($V_{CC}$ = 5V, $V_{SS}$ = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)**

### Table 18.17. Timer B Input (Counter Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time (counted on one edge) | 100 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width (counted on one edge) | 40 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width (counted on one edge) | 40 | | ns |
| $t_{c(TB)}$ | TBiIN input cycle time (counted on both edges) | 200 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width (counted on both edges) | 80 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width (counted on both edges) | 80 | | ns |

### Table 18.18. Timer B Input (Pulse Period Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time | 400 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width | 200 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width | 200 | | ns |

### Table 18.19. Timer B Input (Pulse Width Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time | 400 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width | 200 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width | 200 | | ns |

### Table 18.20. A-D Trigger Input

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(AD)}$ | $\overline{ADTRG}$ input cycle time (trigger able minimum) | 1000 | | ns |
| $t_{w(ADL)}$ | $\overline{ADTRG}$ input LOW pulse width | 125 | | ns |

### Table 18.21. Serial I/O

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input cycle time | 200 | | ns |
| $t_{w(CKH)}$ | CLKi input HIGH pulse width | 100 | | ns |
| $t_{w(CKL)}$ | CLKi input LOW pulse width | 100 | | ns |
| $t_{d(C-Q)}$ | TxDi output delay time | | 80 | ns |
| $t_{h(C-Q)}$ | TxDi hold time | 0 | | ns |
| $t_{su(D-C)}$ | RxDi input setup time | 30 | | ns |
| $t_{h(C-D)}$ | RxDi input hold time | 90 | | ns |

### Table 18.22. External Interrupt $\overline{INTi}$ Input

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{w(INH)}$ | $\overline{INTi}$ input HIGH pulse width | 250 | | ns |
| $t_{w(INL)}$ | $\overline{INTi}$ input LOW pulse width | 250 | | ns |

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 5V

### Timing Requirements

**($V_{CC}$ = 5V, $V_{SS}$ = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)**

**Table 18.23. Multi-master $I^2C$-Bus Line**

| Symbol | Parameter | Standard clock mode | | High-speed clock mode | | Unit |
|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | |
| tBUF | Bus free time | 4.7 | | 1.3 | | µs |
| tHD;STA | The hold time in start condition | 4.0 | | 0.6 | | µs |
| tLOW | The hold time in SCL clock "0" status | 4.7 | | 1.3 | | µs |
| tR | SCL, SDA signals' rising time | | 1000 | 20+0.1Cb | 300 | ns |
| tHD;DAT | Data hold time | 0 | | 0 | 0.9 | µs |
| tHIGH | The hold time in SCL clock "1" status | 4.0 | | 0.6 | | µs |
| tF | SCL, SDA signals' falling time | | 300 | 20+0.1Cb | 300 | ns |
| tSU;DAT | Data setup time | 250 | | 100 | | ns |
| tSU;STA | The setup time in restart condition | 4.7 | | 0.6 | | µs |
| tSU;STO | Stop condition setup time | 4.0 | | 0.6 | | µs |

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

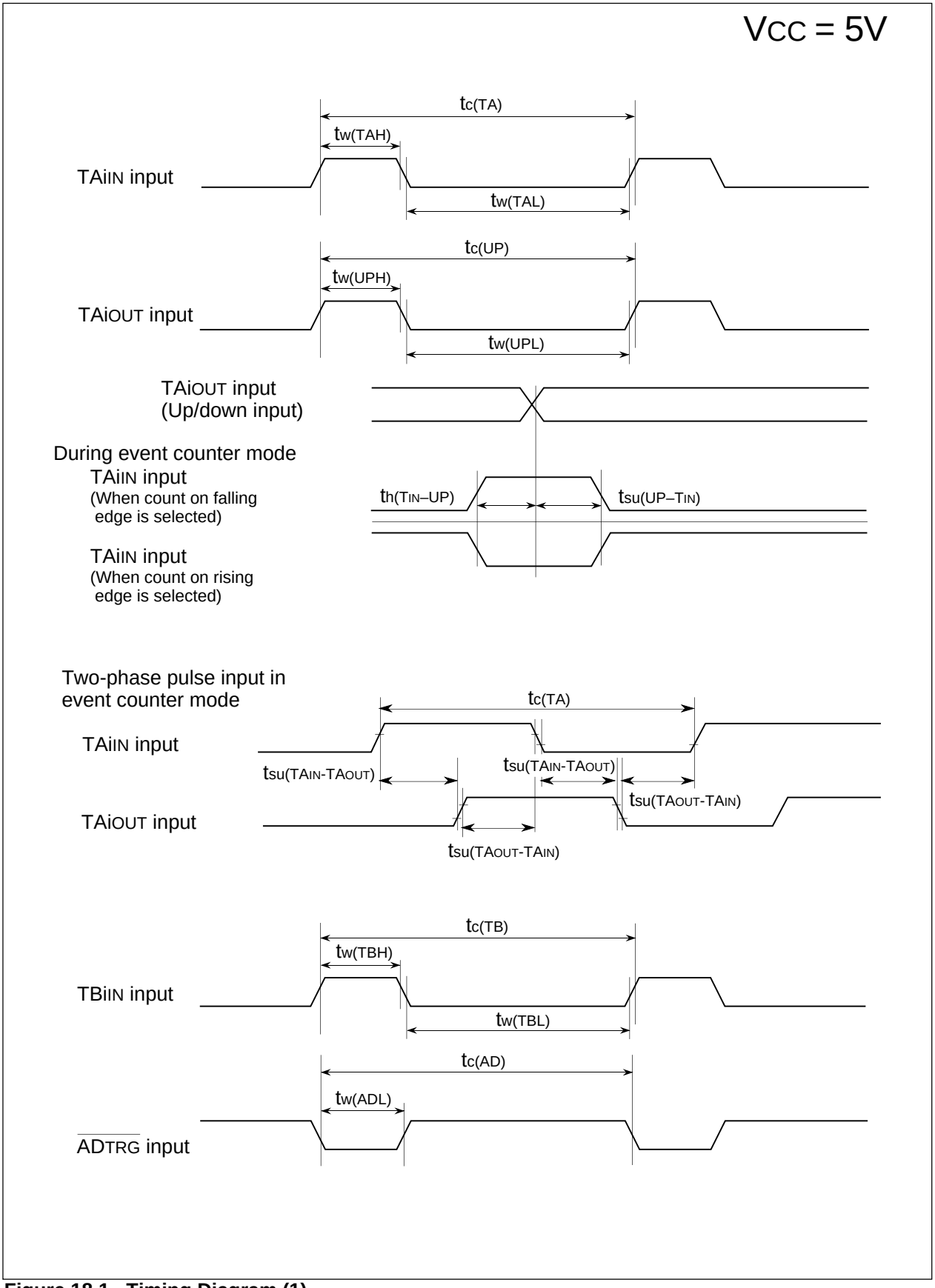



**Figure 18.1. Timing Diagram (1)**

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

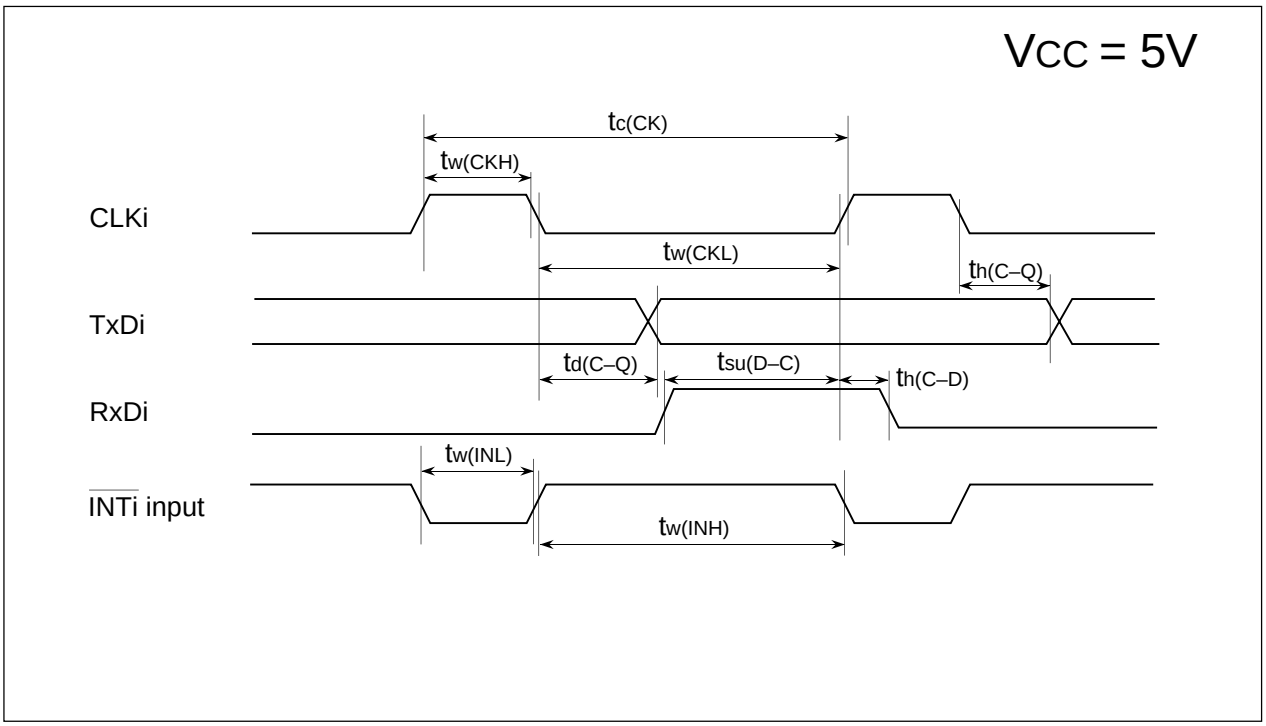



**Figure 18.2. Timing Diagram (2)**

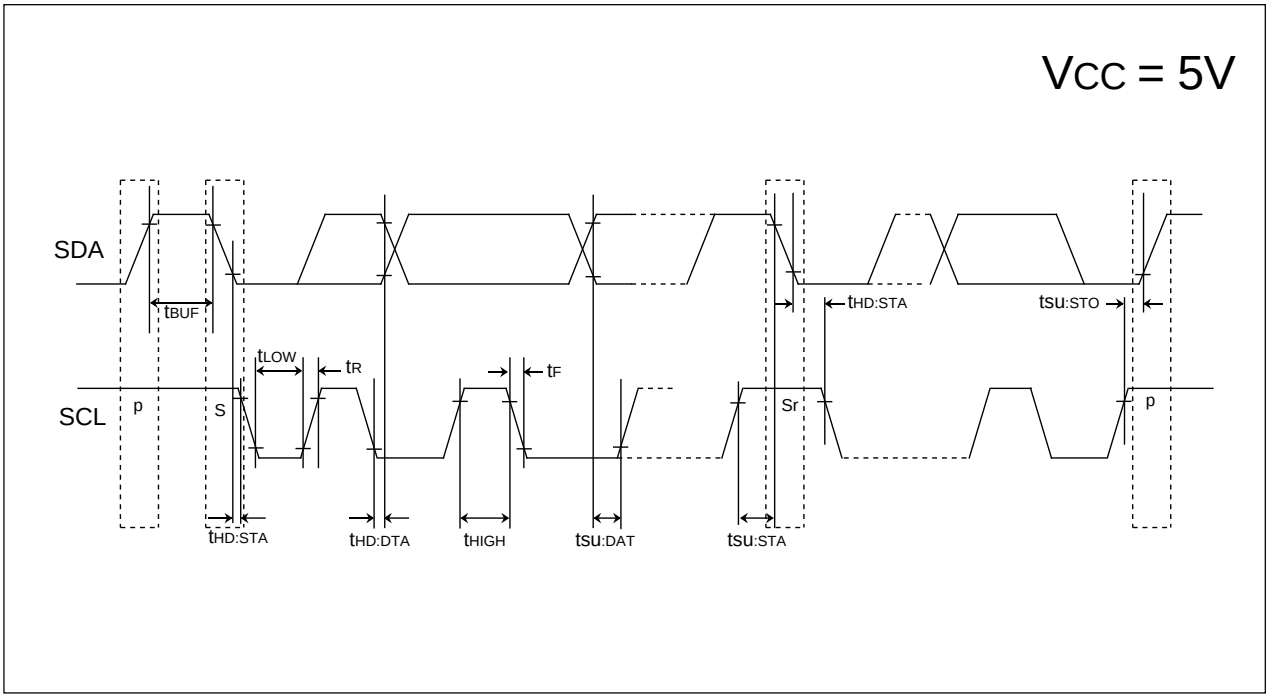



**Figure 18.3. Timing Diagram (3)**

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 3V

## Table 18.24. Electrical Characteristics (Note 1)

| Symbol | Parameter | | Measuring condition | Standard | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| $V_{OH}$ | HIGH output voltage $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | $I_{OH}$=-1mA | $V_{CC}$-0.5 | | $V_{CC}$ | V |
| $V_{OH}$ | HIGH output voltage $X_{OUT}$ | HIGHPOWER | $I_{OH}$=-0.1mA | $V_{CC}$-0.5 | | $V_{CC}$ | V |
| | | LOWPOWER | $I_{OH}$=-50µA | $V_{CC}$-0.5 | | $V_{CC}$ | |
| | HIGH output voltage $X_{COUT}$ | HIGHPOWER | With no load applied | | 2.5 | | V |
| | | LOWPOWER | With no load applied | | 1.6 | | |
| $V_{OL}$ | LOW output voltage $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | $I_{OL}$=1mA | | | 0.5 | V |
| $V_{OL}$ | LOW output voltage $X_{OUT}$ | HIGHPOWER | $I_{OL}$=0.1mA | | | 0.5 | V |
| | | LOWPOWER | $I_{OL}$=50µA | | | 0.5 | |
| | LOW output voltage $X_{COUT}$ | HIGHPOWER | With no load applied | | 0 | | V |
| | | LOWPOWER | With no load applied | | 0 | | |
| $V_{T+}$-$V_{T-}$ | Hysteresis $TA0_{IN}$ to $TA4_{IN}$, $TB0_{IN}$ to $TB2_{IN}$, $\overline{INT0}$ to $\overline{INT5}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS0}$ to $\overline{CTS2}$, SCL, SDA, $CLK0$ to $CLK2$, $TA2_{OUT}$ to $TA4_{OUT}$, $\overline{KI0}$ to $\overline{KI3}$, $RxD0$ to $RxD2$, $S_{IN3}$,$S_{IN4}$ | | | 0.2 | | 0.8 | V |
| $V_{T+}$-$V_{T-}$ | Hysteresis $\overline{RESET}$ | | | 0.2 | 0.7 | 1.8 | V |
| $I_{IH}$ | HIGH input current $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ $X_{IN}$, $\overline{RESET}$, CNVss | | $V_I$=3V | | | 4.0 | µA |
| $I_{IL}$ | LOW input current $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ $X_{IN}$, $\overline{RESET}$, CNVss | | $V_I$=0V | | | -4.0 | µA |
| $R_{PULLUP}$ | Pull-up resistance $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | $V_I$=0V | 50 | 100 | 500 | kΩ |
| $R_{fXIN}$ | Feedback resistance $X_{IN}$ | | | | 3.0 | | MΩ |
| $R_{fXCIN}$ | Feedback resistance $X_{CIN}$ | | | | 25 | | MΩ |
| $V_{RAM}$ | RAM retention voltage | | At stop mode | 2.0 | | | V |

Note 1 : Referenced to $V_{CC}$=2.7 to 3.3V, $V_{SS}$=0V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK)=10MHz unless otherwise specified.

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

# VCC = 3V

## Table 18.25. Electrical Characteristics (2) (Note 1)

| Symbol | Parameter | | Measuring condition | | Standard Min. | Standard Typ. | Standard Max. | Unit |
|---|---|---|---|---|---|---|---|---|
| ICC | Power supply current (VCC=2.7 to 3.6V) | The output pins are open and other pins are VSS | Flash memory | f(BCLK)=10MHz, No division | | 8 | T.B.D | mA |
| | | | Flash memory Program | f(BCLK)=10MHz, Vcc=3.0V | | T.B.D | | mA |
| | | | Flash memory Erase | f(BCLK)=10MHz, Vcc=3.0V | | T.B.D | | mA |
| | | | Flash memory | f(BCLK)=32kHz, Low power dissipation mode, RAM(Note 3) | | T.B.D | | µA |
| | | | | f(BCLK)=32kHz, Low power dissipation mode, Flash memory(Note 3) | | T.B.D | | µA |
| | | | | Ring oscillation, Wait mode | | T.B.D | | µA |
| | | | Flash memory | f(BCLK)=32kHz, Wait mode (Note 2), Oscillation capacity High | | T.B.D | | µA |
| | | | | f(BCLK)=32kHz, Wait mode (Note 2), Oscillation capacity Low | | T.B.D | | µA |
| | | | | Stop mode, Topr=25°C | | 0.7 | T.B.D | µA |
| Idet4 | Voltage down detection dissipation current (Note 4) | | | | | T.B.D | T.B.D | µA |
| Idet3 | Reset level detection dissipation current (Note 4) | | | | | T.B.D | T.B.D | µA |
| Idet2 | RAM retention limit detection dissipation current (Note 4) | | | | | T.B.D | T.B.D | µA |

Note 1: Referenced to VCC=2.7 to 3.3V, VSS=0V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK)=10MHz unless otherwise specified.
Note 2: With one timer operated using fC32.
Note 3: This indicates the memory in which the program to be executed exists.
Note 4: Idet is dissipation current when the following bit is set to "1" (detection circuit enabled).
Idet4: VC27 bit of VCR2 register
Idet3: VC26 bit of VCR2 register
Idet2: VC25 bit of VCR2 register

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 3V

### Timing Requirements

**($V_{CC}$ = 3V, $V_{SS}$ = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)**

**Table 18.26. External Clock Input ($X_{IN}$ input)**

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_c$ | External clock input cycle time | 100 | | ns |
| $t_{w(H)}$ | External clock input HIGH pulse width | 40 | | ns |
| $t_{w(L)}$ | External clock input LOW pulse width | 40 | | ns |
| $t_r$ | External clock rise time | | 18 | ns |
| $t_f$ | External clock fall time | | 18 | ns |

**These standards are not final and focused only normal-version.
Should be used as a reference.**

$V_{CC}$ = 3V

## Timing Requirements

**($V_{CC}$ = 3V, $V_{SS}$ = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)**

### Table 18.27. Timer A Input (Counter Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TA_{iIN}$ input cycle time | 150 | | ns |
| $t_{w(TAH)}$ | $TA_{iIN}$ input HIGH pulse width | 60 | | ns |
| $t_{w(TAL)}$ | $TA_{iIN}$ input LOW pulse width | 60 | | ns |

### Table 18.28. Timer A Input (Gating Input in Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TA_{iIN}$ input cycle time | 600 | | ns |
| $t_{w(TAH)}$ | $TA_{iIN}$ input HIGH pulse width | 300 | | ns |
| $t_{w(TAL)}$ | $TA_{iIN}$ input LOW pulse width | 300 | | ns |

### Table 18.29. Timer A Input (External Trigger Input in One-shot Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TA_{iIN}$ input cycle time | 300 | | ns |
| $t_{w(TAH)}$ | $TA_{iIN}$ input HIGH pulse width | 150 | | ns |
| $t_{w(TAL)}$ | $TA_{iIN}$ input LOW pulse width | 150 | | ns |

### Table 18.30. Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{w(TAH)}$ | $TA_{iIN}$ input HIGH pulse width | 150 | | ns |
| $t_{w(TAL)}$ | $TA_{iIN}$ input LOW pulse width | 150 | | ns |

### Table 18.31. Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(UP)}$ | $TA_{iOUT}$ input cycle time | 3000 | | ns |
| $t_{w(UPH)}$ | $TA_{iOUT}$ input HIGH pulse width | 1500 | | ns |
| $t_{w(UPL)}$ | $TA_{iOUT}$ input LOW pulse width | 1500 | | ns |
| $t_{su(UP-TIN)}$ | $TA_{iOUT}$ input setup time | 600 | | ns |
| $t_{h(TIN-UP)}$ | $TA_{iOUT}$ input hold time | 600 | | ns |

### Table 18.32. Timer A Input (Two-phase Pulse Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TA_{iIN}$ input cycle time | 2 | | µs |
| $t_{su(TAIN-TAOUT)}$ | $TA_{iOUT}$ input setup time | 500 | | ns |
| $t_{su(TAOUT-TAIN)}$ | $TA_{iIN}$ input setup time | 500 | | ns |

**These standards are not final and focused only normal-version.
Should be used as a reference.**

$V_{CC}$ = 3V

## Timing Requirements

**($V_{CC}$ = 3V, $V_{SS}$ = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)**

### Table 18.33. Timer B Input (Counter Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time (counted on one edge) | 150 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width (counted on one edge) | 60 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width (counted on one edge) | 60 | | ns |
| $t_{c(TB)}$ | TBiIN input cycle time (counted on both edges) | 300 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width (counted on both edges) | 120 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width (counted on both edges) | 120 | | ns |

### Table 18.34. Timer B Input (Pulse Period Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time | 600 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width | 300 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width | 300 | | ns |

### Table 18.35. Timer B Input (Pulse Width Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time | 600 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width | 300 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width | 300 | | ns |

### Table 18.36. A-D Trigger Input

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(AD)}$ | $\overline{ADTRG}$ input cycle time (trigger able minimum) | 1500 | | ns |
| $t_{w(ADL)}$ | $\overline{ADTRG}$ input LOW pulse width | 200 | | ns |

### Table 18.37. Serial I/O

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input cycle time | 300 | | ns |
| $t_{w(CKH)}$ | CLKi input HIGH pulse width | 150 | | ns |
| $t_{w(CKL)}$ | CLKi input LOW pulse width | 150 | | ns |
| $t_{d(C-Q)}$ | TxDi output delay time | | 160 | ns |
| $t_{h(C-Q)}$ | TxDi hold time | 0 | | ns |
| $t_{su(D-C)}$ | RxDi input setup time | 50 | | ns |
| $t_{h(C-D)}$ | RxDi input hold time | 90 | | ns |

### Table 18.38. External Interrupt $\overline{INTi}$ Input

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{w(INH)}$ | $\overline{INTi}$ input HIGH pulse width | 380 | | ns |
| $t_{w(INL)}$ | $\overline{INTi}$ input LOW pulse width | 380 | | ns |

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 3V

### Timing Requirements

**($V_{CC}$ = 3V, $V_{SS}$ = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)**

**Table 18.39. Multi-master I²C-Bus Line**

| Symbol | Parameter | Standard clock mode | | High-speed clock mode | | Unit |
|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | |
| tBUF | Bus free time | 4.7 | | 1.3 | | µs |
| tHD;STA | The hold time in start condition | 4.0 | | 0.6 | | µs |
| tLOW | The hold time in SCL clock "0" status | 4.7 | | 1.3 | | µs |
| tR | SCL, SDA signals' rising time | | 1000 | 20+0.1Cb | 300 | ns |
| tHD;DAT | Data hold time | 0 | | 0 | 0.9 | µs |
| tHIGH | The hold time in SCL clock "1" status | 4.0 | | 0.6 | | µs |
| tF | SCL, SDA signals' falling time | | 300 | 20+0.1Cb | 300 | ns |
| tSU;DAT | Data setup time | 250 | | 100 | | ns |
| tSU;STA | The setup time in restart condition | 4.7 | | 0.6 | | µs |
| tSU;STO | Stop condition setup time | 4.0 | | 0.6 | | µs |

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

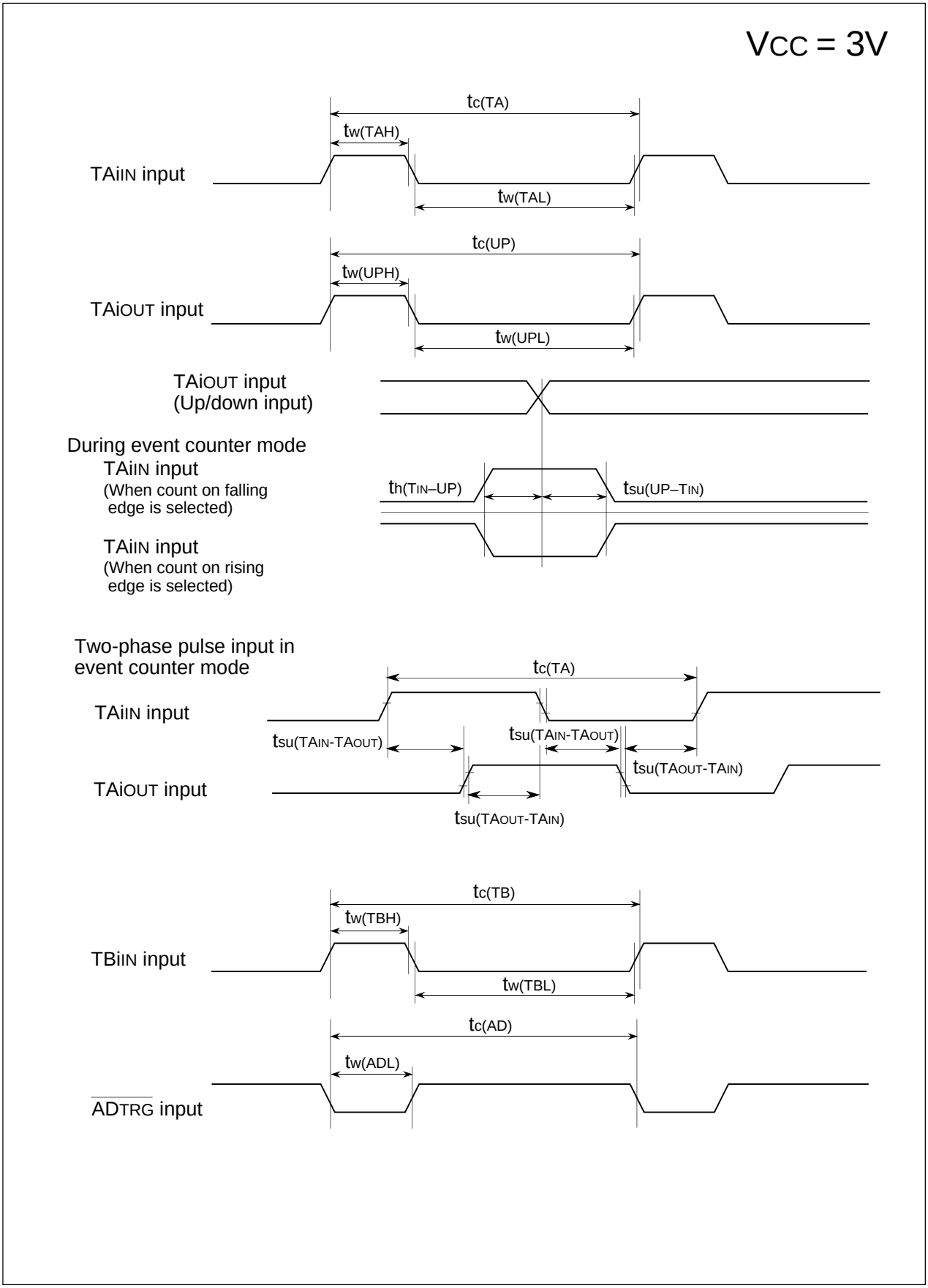



**Figure 18.4. Timing Diagram (1)**

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

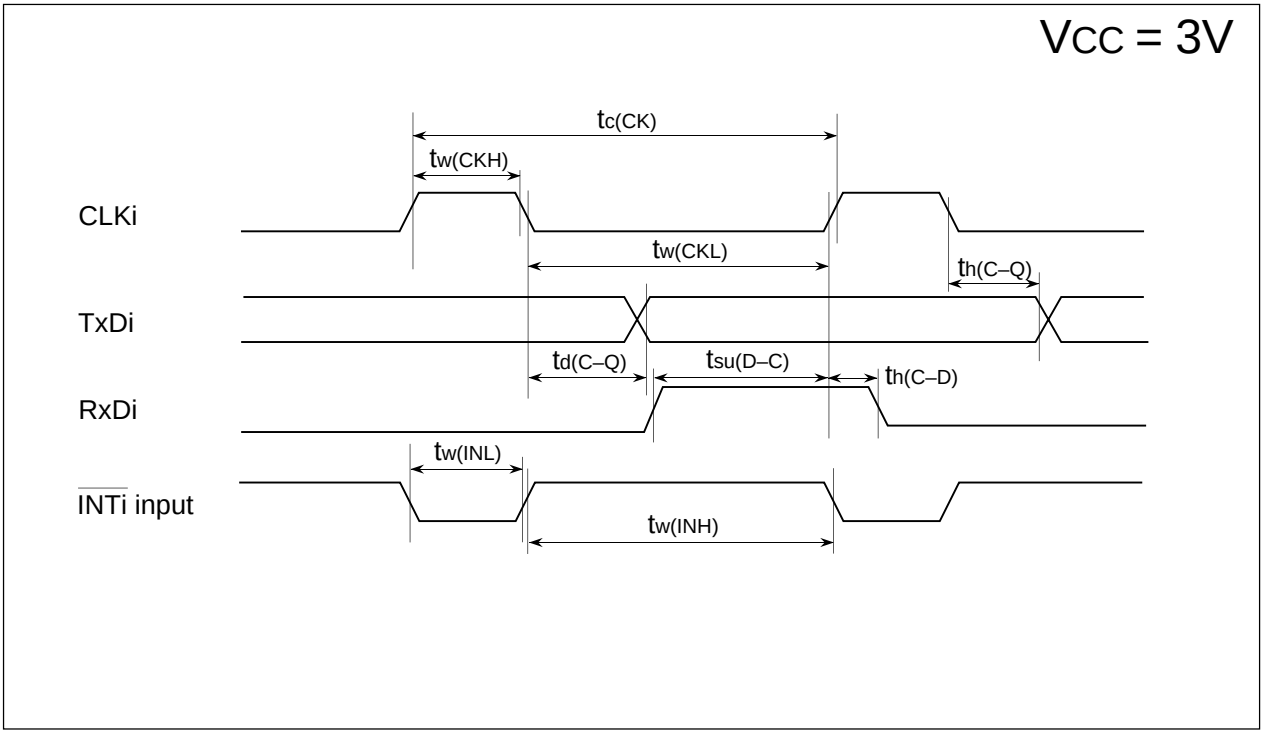



**Figure 18.5. Timing Diagram (2)**

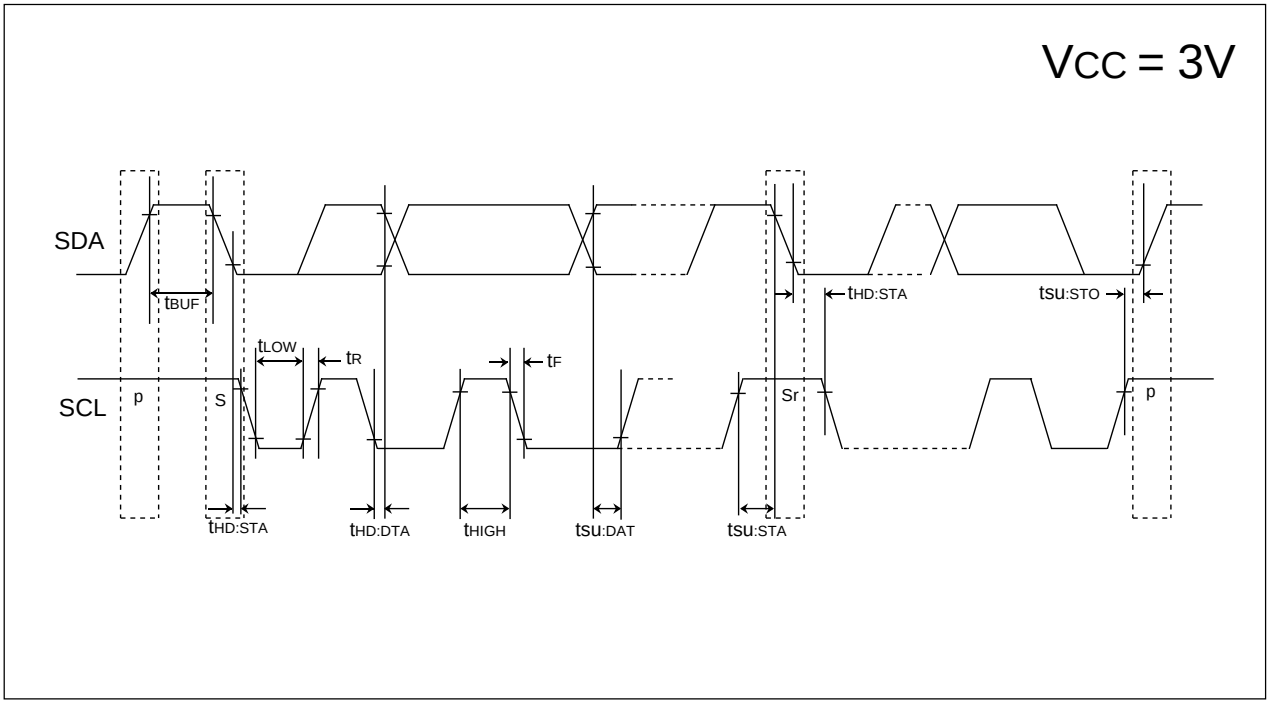



**Figure 18.6. Timing Diagram (3)**

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

## 18.2. T version

**Table 18.40. Absolute Maximum Ratings**

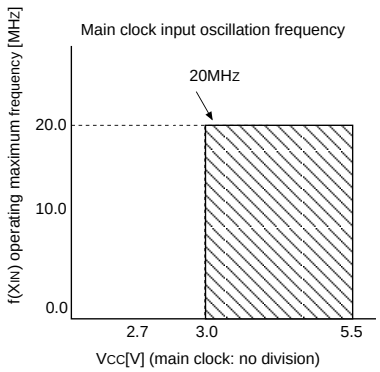
| Symbol | Parameter | | Condition | Rated value | Unit |
|---|---|---|---|---|---|
| $V_{CC}$ | Supply voltage | | $V_{CC}$=AV$_{CC}$ | -0.3 to 6.5 | V |
| AV$_{CC}$ | Analog supply voltage | | $V_{CC}$=AV$_{CC}$ | -0.3 to 6.5 | V |
| $V_I$ | Input voltage | P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$, $X_{IN}$, $V_{REF}$, $\overline{RESET}$, $CNV_{SS}$ | | -0.3 to $V_{CC}$+0.3 | V |
| $V_O$ | Output voltage | P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$, $X_{OUT}$ | | -0.3 to $V_{CC}$+0.3 | V |
| $P_d$ | Power dissipation | | Topr=25 °C | 300 | mW |
| $T_{opr}$ | Operating ambient temperature | | | -40 to 85 | °C |
| $T_{stg}$ | Storage temperature | | | -65 to 150 | °C |

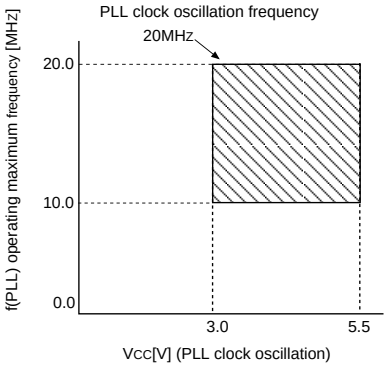
**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

## Table 18.41. Recommended Operating Conditions (Note 1)

| Symbol | Parameter | | Standard Min. | Standard Typ. | Standard Max. | Unit |
|---|---|---|---|---|---|---|
| $V_{CC}$ | Supply voltage | | 3.0 | | 5.5 | V |
| AVcc | Analog supply voltage | | | $V_{CC}$ | | V |
| Vss | Supply voltage | | | 0 | | V |
| AVss | Analog supply voltage | | | 0 | | V |
| $V_{IH}$ | HIGH input voltage | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, $X_{IN}$, $\overline{RESET}$, $CNV_{SS}$ | 0.7$V_{CC}$ | | $V_{CC}$ | V |
| $V_{IL}$ | LOW input voltage | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, $X_{IN}$, $\overline{RESET}$, $CNV_{SS}$ | 0 | | 0.3$V_{CC}$ | V |
| $I_{OH\ (peak)}$ | HIGH peak output current | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | | -10.0 | mA |
| $I_{OH\ (avg)}$ | HIGH average output current | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | | -5.0 | mA |
| $I_{OL\ (peak)}$ | LOW peak output current | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | | 10.0 | mA |
| $I_{OL\ (avg)}$ | LOW average output current | $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | | 5.0 | mA |
| f ($X_{IN}$) | Main clock input oscillation frequency (Note 3) | | 0 | | 20 | MHz |
| f ($X_{CIN}$) | Sub-clock oscillation frequency | | | 32.768 | 50 | kHz |
| $f_1$ (ROC) | Ring oscillation frequency 1 | | | 1 | | MHz |
| $f_2$ (ROC) | Ring oscillation frequency 2 | | | 2 | | MHz |
| $f_3$ (ROC) | Ring oscillation frequency 3 | | | 16 | | MHz |
| f (PLL) | PLL clock oscillation frequency (Note 3) | | 10 | | 20 | MHz |
| f (BCLK) | CPU operation clock | | 0 | | 20 | MHz |
| $T_{SU}$(PLL) | PLL frequency synthesizer stabilization wait time | $V_{CC}$=5.0V | | | 20 | ms |
| | | $V_{CC}$=3.0V | | | 50 | ms |

Note 1: Referenced to $V_{CC}$ = 3.0 to 5.5V at Topr = -40 to 85 °C unless otherwise specified.
Note 2: The mean output current is the mean value within 100ms.
Note 3: Relationship between main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.
Note 4: The total IOL(peak) for all ports must be 80mA max. The total IOH(peak) for all ports must be -80mA max.



**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

**Table 18.42. A-D Conversion Characteristics (Note 1)**

| Symbol | Parameter | | Measuring condition | | Standard | | | Unit |
|---|---|---|---|---|---|---|---|---|
| | | | | | Min. | Typ. | Max. | |
| – | Resolution | | $V_{REF}$ =$V_{CC}$ | | | | 10 | Bits |
| INL | Integral non-linearity error | 10 bit | $V_{REF}$ =$V_{CC}$=5V | $AN_0$ to $AN_7$ input | | | ±3 | LSB |
| | | | | $AN_{00}$ to $AN_{07}$, $AN_{20}$ to $AN_{27}$ input | | | ±7 | LSB |
| | | | $V_{REF}$ =$V_{CC}$=3.3V | $AN_0$ to $AN_7$ input | | | ±5 | LSB |
| | | | | $AN_{00}$ to $AN_{07}$, $AN_{20}$ to $AN_{27}$ input | | | ±7 | LSB |
| | | 8 bit | $V_{REF}$ =$V_{CC}$=3.3V | $AN_0$ to $AN_7$ input | | | ±2 | LSB |
| – | Absolute accuracy | 10 bit | $V_{REF}$ =$V_{CC}$=5V | $AN_0$ to $AN_7$ input | | | ±3 | LSB |
| | | | | $AN_{00}$ to $AN_{07}$, $AN_{20}$ to $AN_{27}$ input | | | ±7 | LSB |
| | | | $V_{REF}$ =$V_{CC}$=3.3V | $AN_0$ to $AN_7$ input | | | ±5 | LSB |
| | | | | $AN_{00}$ to $AN_{07}$, $AN_{20}$ to $AN_{27}$ input | | | ±7 | LSB |
| | | 8 bit | $V_{REF}$ =$V_{CC}$=3.3V | $AN_0$ to $AN_7$ input | | | ±2 | LSB |
| DNL | Differential non-linearity error | | | | | | ±1 | LSB |
| – | Offset error | | | | | | ±3 | LSB |
| – | Gain error | | | | | | ±3 | LSB |
| $R_{LADDER}$ | Ladder resistance | | $V_{REF}$ =$V_{CC}$ | | 10 | | 40 | kΩ |
| $t_{CONV}$ | Conversion time(10bit), Sample & hold function available | | $V_{REF}$ =$V_{CC}$=5V, $ø_{AD}$=10MHz | | 3.3 | | | µs |
| $t_{CONV}$ | Conversion time(8bit), Sample & hold function available | | $V_{REF}$ =$V_{CC}$=5V, $ø_{AD}$=10MHz | | 2.8 | | | µs |
| $t_{SAMP}$ | Sampling time | | | | 0.3 | | | µs |
| $V_{REF}$ | Reference voltage | | | | 2.0 | | $V_{CC}$ | V |
| $V_{IA}$ | Analog input voltage | | | | 0 | | $V_{REF}$ | V |

Note 1: Referenced to $V_{CC}$=$AV_{CC}$=$V_{REF}$=3.3 to 5.5V, $V_{SS}$=$AV_{SS}$=0V at Topr = -40 to 85 °C unless otherwise specified.
Note 2: AD operation clock frequency ($Ø_{AD}$ frequency) must be 10 MHz or less. And divide the $f_{AD}$ if $V_{CC}$ is less than 4.2V, and make $Ø_{AD}$ frequency equal to or lower than $f_{AD}$/2.
Note 3: A case without sample & hold function turn $Ø_{AD}$ frequency into 250 kHz or more in addition to a limit of Note 3.
A case with sample & hold function turn $Ø_{AD}$ frequency into 1MHz or more in addition to a limit of Note 3.

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

**Table 18.43. Flash Memory Version Electrical Characteristics (Note 1) for 100 E/W cycle products**

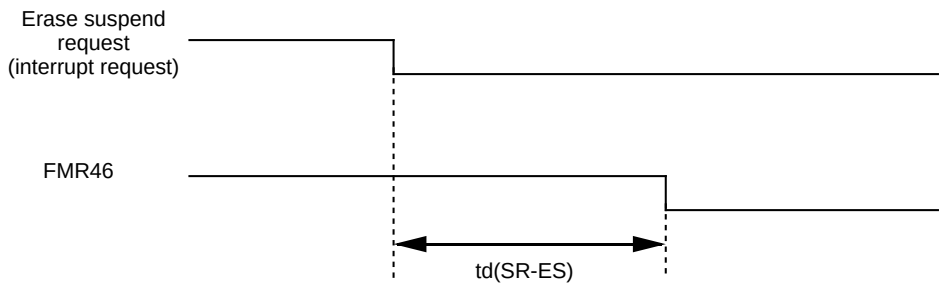
| Symbol | Parameter | | Standard | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. (Note 2) | Max | |
| – | Erase/Write cycle (Note 3) | | 100(Note 4) | | | cycle |
| – | Word program time (Vcc=5.0V, Topr=25°C) | | | 75 | 600 | µs |
| – | Block erase time | 2Kbyte block | | 0.2 | 9 | s |
| | | 8Kbyte block | | 0.4 | 9 | s |
| | | 16Kbyte block | | 0.7 | 9 | s |
| | | 32Kbyte block | | 1.2 | 9 | s |
| td(SR-ES) | Time delay from Suspend Request until Erase Suspend | | | | 20 | ms |
| – | Data retention time (Note 5) | | 20 | | | year |

**Table 18.44. Flash Memory Version Electrical Characteristics (Note 6) for 10000 E/W cycle products (Option)**

**[Block A and Block B (Note 7)]**

| Symbol | Parameter | Standard | | | Unit |
|---|---|---|---|---|---|
| | | Min. | Typ. (Note 2) | Max | |
| – | Erase/Write cycle (Note 3, 8, 9) | 10000(Note 4,10) | | | cycle |
| – | Word program time (Vcc=5.0V, Topr=25°C) | | 100 | | µs |
| – | Block erase time(Vcc=5.0V, Topr=25°C) (2Kbyte block) | | 0.3 | 9 | s |
| td(SR-ES) | Time delay from Suspend Request until Erase Suspend | | | 20 | ms |

Note 1: When not otherwise specified, Vcc = 2.7 to5.5V; Topr = 0 to 60 °C.
Note 2: VCC = 5V; TOPR = 25 °C.
Note 3: Definition of E/W cycle: Each block may be written to a variable number of times - up to a maximum of the total number of distinct word addresses - for every block erase. Performing multiple writes to the same address before an erase operation is prohibited.
Note 4: Maximum number of E/W cycles for which opration is guaranteed.
Note 5: Topr = 55°C.
Note 6: When not otherwise specified, Vcc = 2.7 to 5.5V; Topr = -40 to 85°C.
Note 7: Table18.44 applies for Block A or B E/W cycles > 1000. Otherwise, use Table 18.43.
Note 8: To reduce the number of E/W cycles, a block erase should ideally be performed after writing as many different word addresses (only one time each) as possible. It is important to track the total number of block erases.
Note 9: Should erase error occur during block erase, attempt to execute clear status register command, then clock erase command at least three times until erase error disappears.
Note 10: When Block A or B E/W cycles exceed 100 (Option), select one wait state per block access. When FMR17 is set to "1", one wait state is inserted per access to Block A or B - regardless of the value of PM17. Wait state insertion during access to all other blocks, as well as to internal RAM, is controlled by PM17 - regardless of



**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

### Table 18.45. Low Voltage Detection Circuit Electrical Characteristics (Note 1, Note 3)

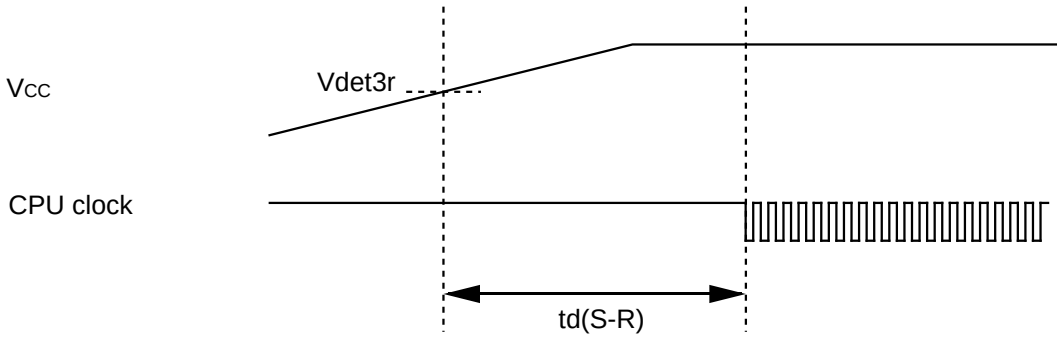
| Symbol | Parameter | Measuring condition | Standard | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| Vdet4 | Voltage down detection voltage (Note 1) | Vcc=0.8 to 5.5V | T.B.D | T.B.D | T.B.D | V |
| Vdet3 | Reset level detection voltage (Notes 1) | | T.B.D | T.B.D | T.B.D | V |
| Vdet3s | Low voltage reset retention voltage (Note 2) | | T.B.D | T.B.D | T.B.D | V |
| Vdet3r | Low voltage reset release voltage | | T.B.D | T.B.D | T.B.D | V |

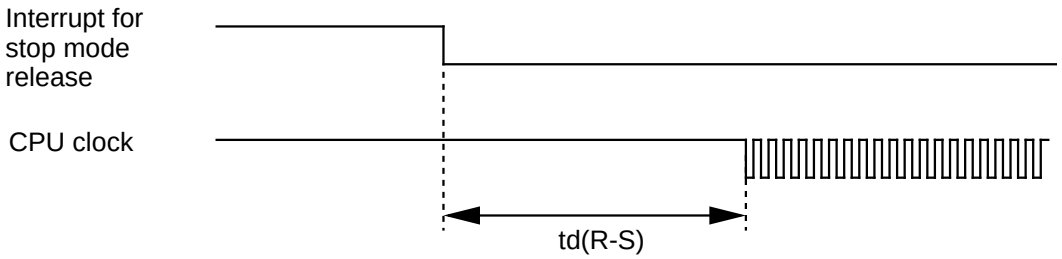
Note 1: Vdet4 > Vdet3
Note 2: Vdet3s is the min voltage at which "hardware reset 2" is maintained. Below this voltage.
Note 3: The low voltage detection circuit is designed to use when Vcc is set to 5V.

### Table 18.46. Power Supply Circuit Timing Characteristics

| Symbol | Parameter | Measuring condition | Standard | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| td(P-R) | Time for internal power supply stabilization during powering-on | Vcc=3.0 to 5.5V | | | 2 | ms |
| td(ROC) | Time for internal ring oscillator stabilization during powering-on | | T.B.D | T.B.D | T.B.D | ms |
| td(R-S) | STOP release time (Note 2) | | | | 2 | ms |
| td(W-S) | Low power dissipation mode wait mode release time (Note 2) | | | | 150 | µs |
| td(M-L) | Time for internal power supply stabilization when main clock oscillation starts | | | | 50 | µs |
| td(S-R) | Hardware reset 2 release wait time | Vcc=Vdet3r to 5.5V | | 6 (Note 1) | 20 | ms |
| td(E-A) | Low voltage detection circuit operation start time (Note 3) | Vcc=3.0 to 5.5V | | | 20 | µs |

Note 1: When Vcc = 5V
Note 2: This is the time between interrupt for (STOP/WAIT) mode release and resumption of CPU clock operation.
Note 3: After enabling low voltage detection, this time is required before proper detection can occur.



**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 5V

## Table 18.47. Electrical Characteristics (Note 1)

| Symbol | Parameter | | Measuring condition | Standard Min. | Standard Typ. | Standard Max. | Unit |
|---|---|---|---|---|---|---|---|
| $V_{OH}$ | HIGH output voltage | P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$ | $I_{OH}$=-5mA | $V_{CC}$-2.0 | | $V_{CC}$ | V |
| $V_{OH}$ | HIGH output voltage | P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$ | $I_{OH}$=-200µA | $V_{CC}$-0.3 | | $V_{CC}$ | V |
| $V_{OH}$ | HIGH output voltage $X_{OUT}$ | HIGHPOWER | $I_{OH}$=-1mA | $V_{CC}$-2.0 | | $V_{CC}$ | V |
| | | LOWPOWER | $I_{OH}$=-0.5mA | $V_{CC}$-2.0 | | $V_{CC}$ | |
| | HIGH output voltage $X_{COUT}$ | HIGHPOWER | With no load applied | | 2.5 | | V |
| | | LOWPOWER | With no load applied | | 1.6 | | |
| $V_{OL}$ | LOW output voltage | P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$ | $I_{OL}$=5mA | | | 2.0 | V |
| $V_{OL}$ | LOW output voltage | P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$ | $I_{OL}$=200µA | | | 0.45 | V |
| $V_{OL}$ | LOW output voltage $X_{OUT}$ | HIGHPOWER | $I_{OL}$=1mA | | | 2.0 | V |
| | | LOWPOWER | $I_{OL}$=0.5mA | | | 2.0 | |
| | LOW output voltage $X_{COUT}$ | HIGHPOWER | With no load applied | | 0 | | V |
| | | LOWPOWER | With no load applied | | 0 | | |
| $V_{T+}$-$V_{T-}$ | Hysteresis | TA0$_{IN}$ to TA4$_{IN}$, TB0$_{IN}$ to TB2$_{IN}$, INT$_0$ to INT$_5$, NMI, ADTRG, CTS$_0$ to CTS$_2$, SCL, SDA, CLK$_0$ to CLK$_2$, TA2$_{OUT}$ to TA4$_{OUT}$, KI$_0$ to KI$_3$, RxD$_0$ to RxD$_2$, SIN$_3$, SIN$_4$ | | 0.2 | | 1.0 | V |
| $V_{T+}$-$V_{T-}$ | Hysteresis | $\overline{\text{RESET}}$ | | 0.2 | | 2.5 | V |
| $I_{IH}$ | HIGH input current | P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$, $X_{IN}$, $\overline{\text{RESET}}$, CNVss | $V_I$=5V | | | 5.0 | µA |
| $I_{IL}$ | LOW input current | P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$, $X_{IN}$, $\overline{\text{RESET}}$, CNVss | $V_I$=0V | | | -5.0 | µA |
| $R_{PULLUP}$ | Pull-up resistance | P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$ | $V_I$=0V | 30 | 50 | 170 | k |
| $R_{fXIN}$ | Feedback resistance $X_{IN}$ | | | | 1.5 | | M |
| $R_{fXCIN}$ | Feedback resistance $X_{CIN}$ | | | | 15 | | M |
| $V_{RAM}$ | RAM retention voltage | | At stop mode | 2.0 | | | V |

Note 1: Referenced to $V_{CC}$=4.2 to 5.5V, $V_{SS}$=0V at Topr = -40 to 85 °C, f(BCLK)=20MHz unless otherwise specified.

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 5V

## Table 18.48. Electrical Characteristics (2) (Note 1)

| Symbol | Parameter | | Measuring condition | | Standard Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|---|
| $I_{CC}$ | Power supply current ($V_{CC}$=3.0 to 5.5V) | The output pins are open and other pins are $V_{SS}$ | Flash memory | $f(X_{IN})$=20MHz, No division | | 16 | 19 | mA |
| | | | | No division, Ring oscillation | | 1 | | mA |
| | | | Flash memory Program | f(BCLK)=10MHz, $V_{CC1}$=5.0V | | T.B.D | | mA |
| | | | Flash memory Erase | f(BCLK)=10MHz, $V_{CC1}$=5.0V | | T.B.D | | mA |
| | | | Flash memory | f(BCLK)=32kHz, Low power dissipation mode, RAM(Note 3) | | 25 | | µA |
| | | | | f(BCLK)=32kHz Low power dissipation mode, Flash memory(Note 3) | | 500 | | µA |
| | | | | Ring oscillation, Wait mode | | T.B.D | | µA |
| | | | Flash memory | f(BCLK)=32kHz, Wait mode (Note 2), Oscillation capacity High | | 12 | | µA |
| | | | | f(BCLK)=32kHz, Wait mode(Note 2), Oscillation capacity Low | | T.B.D | | µA |
| | | | | Stop mode, $T_{opr}$=25°C | | 0.8 | 3 | µA |
| Idet4 | Voltage down detection dissipation current (Note 4) | | | | | T.B.D | T.B.D | µA |
| Idet3 | Reset area detection dissipation current (Note 4) | | | | | T.B.D | T.B.D | µA |
| Idet2 | RAM retention limit detection dissipation current (Note 4) | | | | | T.B.D | T.B.D | µA |

Note 1: Referenced to $V_{CC}$=3.0 to 5.5V, $V_{SS}$=0V at Topr = -40 to 85 °C, $f(X_{IN})$=20MHz unless otherwise specified.
Note 2: With one timer operated using $f_{C32}$.
Note 3: This indicates the memory in which the program to be executed exists.
Note 4: Idet is dissipation current when the following bit is set to "1" (detection circuit enabled).
Idet4: VC27 bit of VCR2 register
Idet3: VC26 bit of VCR2 register
Idet2: VC25 bit of VCR2 register

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 5V

### Timing Requirements

**($V_{CC}$ = 5V, $V_{SS}$ = 0V, at Topr = – 40 to 85°C unless otherwise specified)**

**Table 18.49. External Clock Input ($X_{IN}$ input)**

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_c$ | External clock input cycle time | 50 | | ns |
| $t_{w(H)}$ | External clock input HIGH pulse width | 20 | | ns |
| $t_{w(L)}$ | External clock input LOW pulse width | 20 | | ns |
| $t_r$ | External clock rise time | | 9 | ns |
| $t_f$ | External clock fall time | | 9 | ns |

**These standards are not final and focused only normal-version.
Should be used as a reference.**

$V_{CC}$ = 5V

## Timing Requirements

**($V_{CC}$ = 5V, $V_{SS}$ = 0V, at Topr = – 40 to 85°C unless otherwise specified)**

### Table 18.50. Timer A Input (Counter Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TAi_{IN}$ input cycle time | 100 | | ns |
| $t_{w(TAH)}$ | $TAi_{IN}$ input HIGH pulse width | 40 | | ns |
| $t_{w(TAL)}$ | $TAi_{IN}$ input LOW pulse width | 40 | | ns |

### Table 18.51. Timer A Input (Gating Input in Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TAi_{IN}$ input cycle time | 400 | | ns |
| $t_{w(TAH)}$ | $TAi_{IN}$ input HIGH pulse width | 200 | | ns |
| $t_{w(TAL)}$ | $TAi_{IN}$ input LOW pulse width | 200 | | ns |

### Table 18.52. Timer A Input (External Trigger Input in One-shot Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TAi_{IN}$ input cycle time | 200 | | ns |
| $t_{w(TAH)}$ | $TAi_{IN}$ input HIGH pulse width | 100 | | ns |
| $t_{w(TAL)}$ | $TAi_{IN}$ input LOW pulse width | 100 | | ns |

### Table 18.53. Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{w(TAH)}$ | $TAi_{IN}$ input HIGH pulse width | 100 | | ns |
| $t_{w(TAL)}$ | $TAi_{IN}$ input LOW pulse width | 100 | | ns |

### Table 18.54. Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(UP)}$ | $TAi_{OUT}$ input cycle time | 2000 | | ns |
| $t_{w(UPH)}$ | $TAi_{OUT}$ input HIGH pulse width | 1000 | | ns |
| $t_{w(UPL)}$ | $TAi_{OUT}$ input LOW pulse width | 1000 | | ns |
| $t_{su(UP-TIN)}$ | $TAi_{OUT}$ input setup time | 400 | | ns |
| $t_{h(TIN-UP)}$ | $TAi_{OUT}$ input hold time | 400 | | ns |

### Table 18.55. Timer A Input (Two-phase Pulse Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TAi_{IN}$ input cycle time | 800 | | ns |
| $t_{su(TAIN-TAOUT)}$ | $TAi_{OUT}$ input setup time | 200 | | ns |
| $t_{su(TAOUT-TAIN)}$ | $TAi_{IN}$ input setup time | 200 | | ns |

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 5V

## Timing Requirements

**($V_{CC}$ = 5V, $V_{SS}$ = 0V, at Topr = – 40 to 85°C unless otherwise specified)**

### Table 18.56. Timer B Input (Counter Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time (counted on one edge) | 100 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width (counted on one edge) | 40 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width (counted on one edge) | 40 | | ns |
| $t_{c(TB)}$ | TBiIN input cycle time (counted on both edges) | 200 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width (counted on both edges) | 80 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width (counted on both edges) | 80 | | ns |

### Table 18.57. Timer B Input (Pulse Period Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time | 400 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width | 200 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width | 200 | | ns |

### Table 18.58. Timer B Input (Pulse Width Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time | 400 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width | 200 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width | 200 | | ns |

### Table 18.59. A-D Trigger Input

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(AD)}$ | $\overline{ADTRG}$ input cycle time (trigger able minimum) | 1000 | | ns |
| $t_{w(ADL)}$ | $\overline{ADTRG}$ input LOW pulse width | 125 | | ns |

### Table 18.60. Serial I/O

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input cycle time | 200 | | ns |
| $t_{w(CKH)}$ | CLKi input HIGH pulse width | 100 | | ns |
| $t_{w(CKL)}$ | CLKi input LOW pulse width | 100 | | ns |
| $t_{d(C-Q)}$ | TxDi output delay time | | 80 | ns |
| $t_{h(C-Q)}$ | TxDi hold time | 0 | | ns |
| $t_{su(D-C)}$ | RxDi input setup time | 30 | | ns |
| $t_{h(C-D)}$ | RxDi input hold time | 90 | | ns |

### Table 18.61. External Interrupt $\overline{INTi}$ Input

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{w(INH)}$ | $\overline{INTi}$ input HIGH pulse width | 250 | | ns |
| $t_{w(INL)}$ | $\overline{INTi}$ input LOW pulse width | 250 | | ns |

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 5V

## Timing Requirements

**($V_{CC}$ = 5V, $V_{SS}$ = 0V, at Topr = – 40 to 85°C unless otherwise specified)**

### Table 18.62. Multi-master $I^2C$-Bus Line

| Symbol | Parameter | Standard clock mode | | High-speed clock mode | | Unit |
|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | |
| tBUF | Bus free time | 4.7 | | 1.3 | | µs |
| tHD;STA | The hold time in start condition | 4.0 | | 0.6 | | µs |
| tLOW | The hold time in SCL clock "0" status | 4.7 | | 1.3 | | µs |
| tR | SCL, SDA signals' rising time | | 1000 | 20+0.1Cb | 300 | ns |
| tHD;DAT | Data hold time | 0 | | 0 | 0.9 | µs |
| tHIGH | The hold time in SCL clock "1" status | 4.0 | | 0.6 | | µs |
| tF | SCL, SDA signals' falling time | | 300 | 20+0.1Cb | 300 | ns |
| tSU;DAT | Data setup time | 250 | | 100 | | ns |
| tSU;STA | The setup time in restart condition | 4.7 | | 0.6 | | µs |
| tSU;STO | Stop condition setup time | 4.0 | | 0.6 | | µs |

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

Vcc = 5V

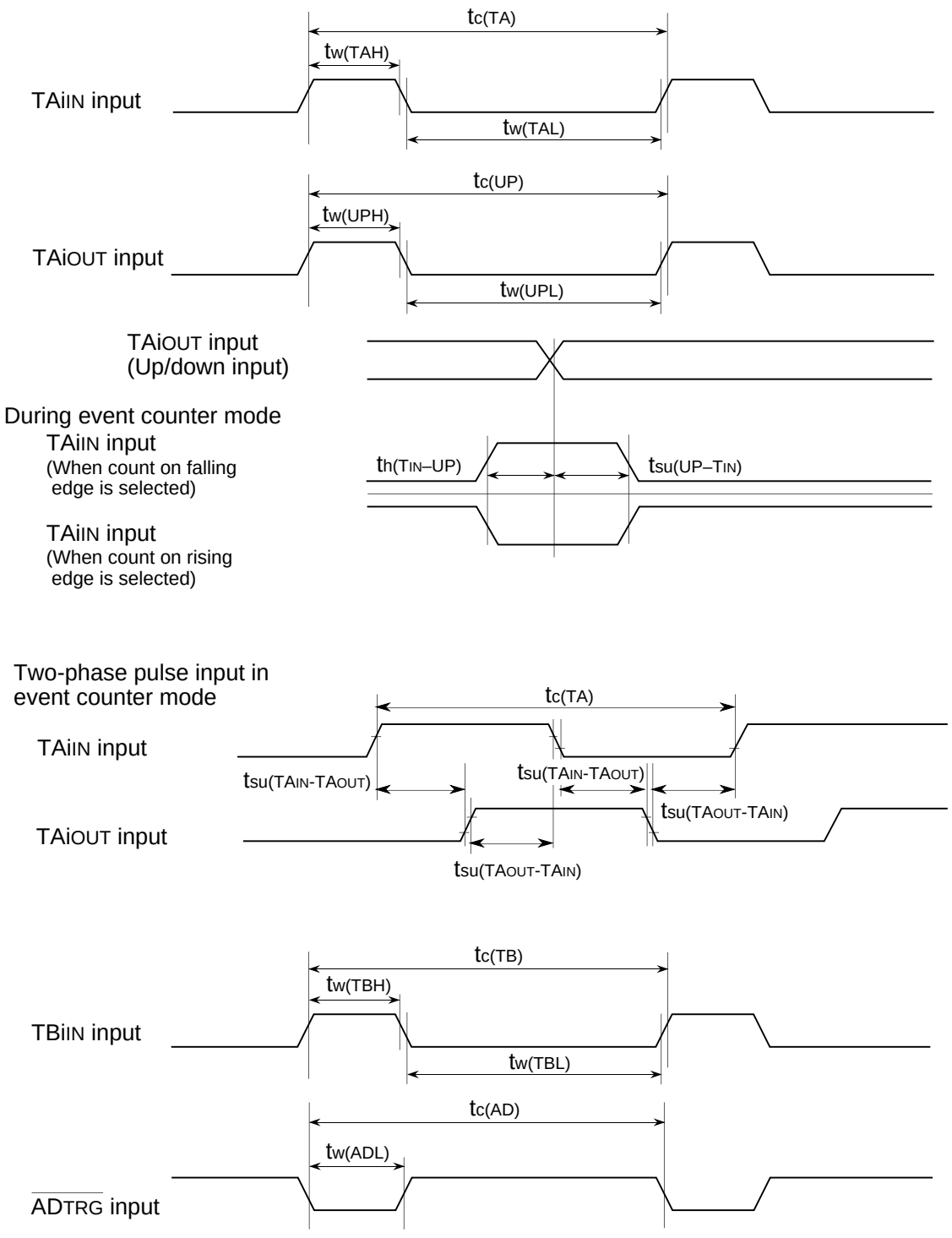



**Figure 18.7. Timing Diagram (1)**

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

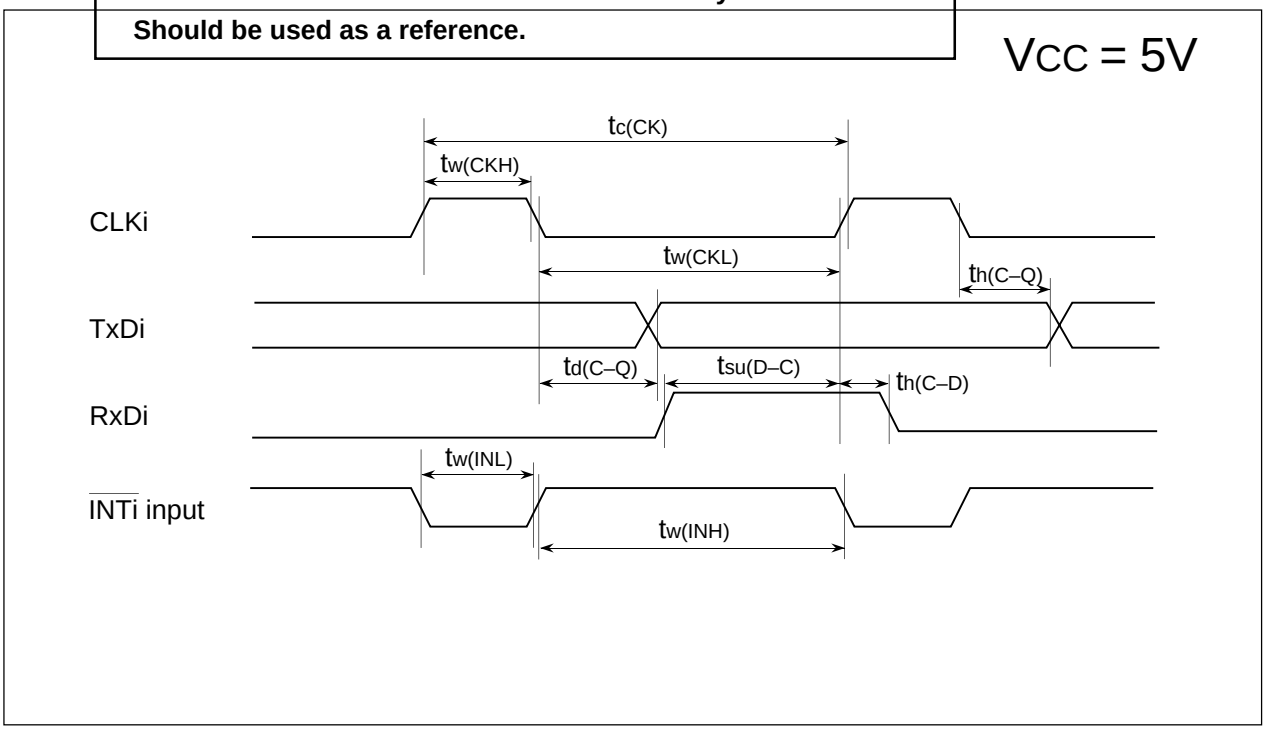



**Figure 18.8. Timing Diagram (2)**

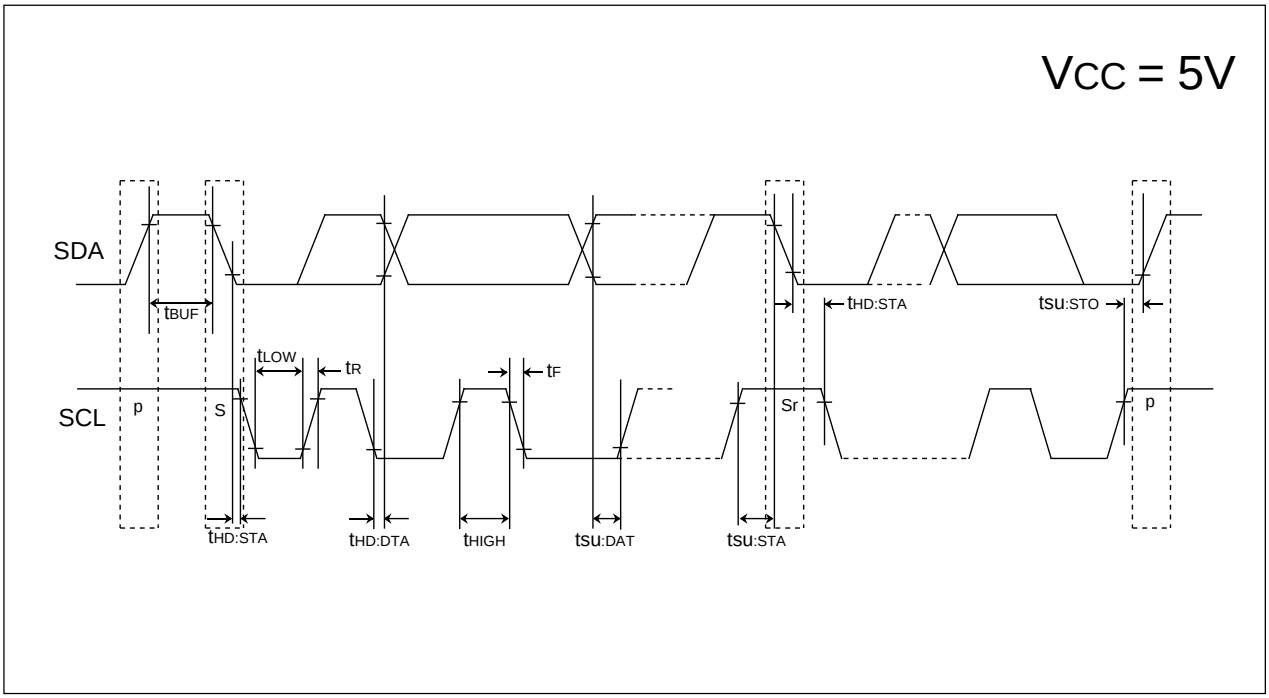



**Figure 18.9. Timing Diagram (3)**

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 3V

### Table 18.63. Electrical Characteristics (Note)

| Symbol | Parameter | | Measuring condition | Standard | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| $V_{OH}$ | HIGH output voltage $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | $I_{OH}$=-1mA | $V_{CC}$-0.5 | | $V_{CC}$ | V |
| $V_{OH}$ | HIGH output voltage $X_{OUT}$ | HIGHPOWER | $I_{OH}$=-0.1mA | $V_{CC}$-0.5 | | $V_{CC}$ | V |
| | | LOWPOWER | $I_{OH}$=-50µA | $V_{CC}$-0.5 | | $V_{CC}$ | |
| | HIGH output voltage $X_{COUT}$ | HIGHPOWER | With no load applied | | 2.5 | | V |
| | | LOWPOWER | With no load applied | | 1.6 | | |
| $V_{OL}$ | LOW output voltage $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | $I_{OL}$=1mA | | | 0.5 | V |
| $V_{OL}$ | LOW output voltage $X_{OUT}$ | HIGHPOWER | $I_{OL}$=0.1mA | | | 0.5 | V |
| | | LOWPOWER | $I_{OL}$=50µA | | | 0.5 | |
| | LOW output voltage $X_{COUT}$ | HIGHPOWER | With no load applied | | 0 | | V |
| | | LOWPOWER | With no load applied | | 0 | | |
| $V_{T+}$-$V_{T-}$ | Hysteresis $TA0_{IN}$ to $TA4_{IN}$, $TB0_{IN}$ to $TB2_{IN}$, $\overline{INT0}$ to $\overline{INT5}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS0}$ to $\overline{CTS2}$, SCL, SDA, $CLK0$ to $CLK2$, $TA2_{OUT}$ to $TA4_{OUT}$, $\overline{KI0}$ to $\overline{KI3}$, $RxD0$ to $RxD2$, $S_{IN3}$,$S_{IN4}$ | | | 0.2 | | 0.8 | V |
| $V_{T+}$-$V_{T-}$ | Hysteresis $\overline{RESET}$ | | | 0.2 | 0.7 | 1.8 | V |
| $I_{IH}$ | HIGH input current $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ $X_{IN}$, $\overline{RESET}$, CNVss | | $V_I$=3V | | | 4.0 | µA |
| $I_{IL}$ | LOW input current $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ $X_{IN}$, $\overline{RESET}$, CNVss | | $V_I$=0V | | | -4.0 | µA |
| $R_{PULLUP}$ | Pull-up resistance $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ | | $V_I$=0V | 50 | 100 | 500 | kΩ |
| $R_{fXIN}$ | Feedback resistance $X_{IN}$ | | | | 3.0 | | MΩ |
| $R_{fXCIN}$ | Feedback resistance $X_{CIN}$ | | | | 25 | | MΩ |
| $V_{RAM}$ | RAM retention voltage | | At stop mode | 2.0 | | | V |

Note 1 : Referenced to $V_{CC}$=3.0 to 3.3V, $V_{SS}$=0V at Topr = -40 to 85 °C, f(BCLK)=20MHz unless otherwise specified.

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 3V

## Table 18.64. Electrical Characteristics (2) (Note 1)

| Symbol | Parameter | | Measuring condition | | Standard Min. | Standard Typ. | Standard Max. | Unit |
|---|---|---|---|---|---|---|---|---|
| $I_{CC}$ | Power supply current ($V_{CC}$=2.7 to 3.6V) | The output pins are open and other pins are $V_{SS}$ | Flash memory | f(BCLK)=10MHz, No division | | 8 | T.B.D | mA |
| | | | Flash memory Program | f(BCLK)=10MHz, Vcc=3.0V | | T.B.D | | mA |
| | | | Flash memory Erase | f(BCLK)=10MHz, Vcc=3.0V | | T.B.D | | mA |
| | | | Flash memory | f(BCLK)=32kHz, Low power dissipation mode, RAM(Note 3) | | T.B.D | | µA |
| | | | | f(BCLK)=32kHz, Low power dissipation mode, Flash memory(Note 3) | | T.B.D | | µA |
| | | | | Ring oscillation, Wait mode | | T.B.D | | µA |
| | | | Flash memory | f(BCLK)=32kHz, Wait mode (Note 2), Oscillation capacity High | | T.B.D | | µA |
| | | | | f(BCLK)=32kHz, Wait mode (Note 2), Oscillation capacity Low | | T.B.D | | µA |
| | | | | Stop mode, $T_{opr}$=25°C | | 0.7 | T.B.D | µA |
| Idet4 | Voltage down detection dissipation current (Note 4) | | | | | T.B.D | T.B.D | µA |
| Idet3 | Reset level detection dissipation current (Note 4) | | | | | T.B.D | T.B.D | µA |
| Idet2 | RAM retention limit detection dissipation current (Note 4) | | | | | T.B.D | T.B.D | µA |

Note 1: Referenced to $V_{CC}$=3.0 to 3.3V, $V_{SS}$=0V at Topr = -40 to 85 °C, f(BCLK)=20MHz unless otherwise specified.
Note 2: With one timer operated using $f_{C32}$.
Note 3: This indicates the memory in which the program to be executed exists.
Note 4: Idet is dissipation current when the following bit is set to "1" (detection circuit enabled).
Idet4: VC27 bit of VCR2 register
Idet3: VC26 bit of VCR2 register
Idet2: VC25 bit of VCR2 register

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

## Timing Requirements

**($V_{CC}$ = 3V, $V_{SS}$ = 0V, at Topr = – 40 to 85°C unless otherwise specified)**

### Table 18.65. External Clock Input ($X_{IN}$ input)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_c$ | External clock input cycle time | 100 | | ns |
| $t_{w(H)}$ | External clock input HIGH pulse width | 40 | | ns |
| $t_{w(L)}$ | External clock input LOW pulse width | 40 | | ns |
| $t_r$ | External clock rise time | | 18 | ns |
| $t_f$ | External clock fall time | | 18 | ns |

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 3V

## Timing Requirements

**($V_{CC}$ = 3V, $V_{SS}$ = 0V, at Topr = – 40 to 85°C unless otherwise specified)**

### Table 18.66. Timer A Input (Counter Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TAi_{IN}$ input cycle time | 150 | | ns |
| $t_{w(TAH)}$ | $TAi_{IN}$ input HIGH pulse width | 60 | | ns |
| $t_{w(TAL)}$ | $TAi_{IN}$ input LOW pulse width | 60 | | ns |

### Table 18.67. Timer A Input (Gating Input in Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TAi_{IN}$ input cycle time | 600 | | ns |
| $t_{w(TAH)}$ | $TAi_{IN}$ input HIGH pulse width | 300 | | ns |
| $t_{w(TAL)}$ | $TAi_{IN}$ input LOW pulse width | 300 | | ns |

### Table 18.68. Timer A Input (External Trigger Input in One-shot Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TAi_{IN}$ input cycle time | 300 | | ns |
| $t_{w(TAH)}$ | $TAi_{IN}$ input HIGH pulse width | 150 | | ns |
| $t_{w(TAL)}$ | $TAi_{IN}$ input LOW pulse width | 150 | | ns |

### Table 18.69. Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{w(TAH)}$ | $TAi_{IN}$ input HIGH pulse width | 150 | | ns |
| $t_{w(TAL)}$ | $TAi_{IN}$ input LOW pulse width | 150 | | ns |

### Table 18.70. Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(UP)}$ | $TAi_{OUT}$ input cycle time | 3000 | | ns |
| $t_{w(UPH)}$ | $TAi_{OUT}$ input HIGH pulse width | 1500 | | ns |
| $t_{w(UPL)}$ | $TAi_{OUT}$ input LOW pulse width | 1500 | | ns |
| $t_{su(UP-TIN)}$ | $TAi_{OUT}$ input setup time | 600 | | ns |
| $t_{h(TIN-UP)}$ | $TAi_{OUT}$ input hold time | 600 | | ns |

### Table 18.71. Timer A Input (Two-phase Pulse Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TA)}$ | $TAi_{IN}$ input cycle time | 2 | | µs |
| $t_{su(TAIN-TAOUT)}$ | $TAi_{OUT}$ input setup time | 500 | | ns |
| $t_{su(TAOUT-TAIN)}$ | $TAi_{IN}$ input setup time | 500 | | ns |

**These standards are not final and focused only normal-version.
Should be used as a reference.**

$V_{CC}$ = 3V

## Timing Requirements

**($V_{CC}$ = 3V, $V_{SS}$ = 0V, at Topr = – 40 to 85°C unless otherwise specified)**

### Table 18.72. Timer B Input (Counter Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | $TBi_{IN}$ input cycle time (counted on one edge) | 150 | | ns |
| $t_{w(TBH)}$ | $TBi_{IN}$ input HIGH pulse width (counted on one edge) | 60 | | ns |
| $t_{w(TBL)}$ | $TBi_{IN}$ input LOW pulse width (counted on one edge) | 60 | | ns |
| $t_{c(TB)}$ | $TBi_{IN}$ input cycle time (counted on both edges) | 300 | | ns |
| $t_{w(TBH)}$ | $TBi_{IN}$ input HIGH pulse width (counted on both edges) | 120 | | ns |
| $t_{w(TBL)}$ | $TBi_{IN}$ input LOW pulse width (counted on both edges) | 120 | | ns |

### Table 18.73. Timer B Input (Pulse Period Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | $TBi_{IN}$ input cycle time | 600 | | ns |
| $t_{w(TBH)}$ | $TBi_{IN}$ input HIGH pulse width | 300 | | ns |
| $t_{w(TBL)}$ | $TBi_{IN}$ input LOW pulse width | 300 | | ns |

### Table 18.74. Timer B Input (Pulse Width Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(TB)}$ | $TBi_{IN}$ input cycle time | 600 | | ns |
| $t_{w(TBH)}$ | $TBi_{IN}$ input HIGH pulse width | 300 | | ns |
| $t_{w(TBL)}$ | $TBi_{IN}$ input LOW pulse width | 300 | | ns |

### Table 18.75. A-D Trigger Input

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(AD)}$ | $\overline{ADTRG}$ input cycle time (trigger able minimum) | 1500 | | ns |
| $t_{w(ADL)}$ | $\overline{ADTRG}$ input LOW pulse width | 200 | | ns |

### Table 18.76. Serial I/O

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input cycle time | 300 | | ns |
| $t_{w(CKH)}$ | CLKi input HIGH pulse width | 150 | | ns |
| $t_{w(CKL)}$ | CLKi input LOW pulse width | 150 | | ns |
| $t_{d(C-Q)}$ | TxDi output delay time | | 160 | ns |
| $t_{h(C-Q)}$ | TxDi hold time | 0 | | ns |
| $t_{su(D-C)}$ | RxDi input setup time | 50 | | ns |
| $t_{h(C-D)}$ | RxDi input hold time | 90 | | ns |

### Table 18.77. External Interrupt $\overline{INTi}$ Input

| Symbol | Parameter | Standard | | Unit |
|---|---|---|---|---|
| | | Min. | Max. | |
| $t_{w(INH)}$ | $\overline{INTi}$ input HIGH pulse width | 380 | | ns |
| $t_{w(INL)}$ | $\overline{INTi}$ input LOW pulse width | 380 | | ns |

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

$V_{CC}$ = 3V

### Timing Requirements

**($V_{CC}$ = 3V, $V_{SS}$ = 0V, at Topr = – 40 to 85°C unless otherwise specified)**

**Table 18.78. Multi-master I²C-Bus Line**

| Symbol | Parameter | Standard clock mode | | High-speed clock mode | | Unit |
|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | |
| tBUF | Bus free time | 4.7 | | 1.3 | | µs |
| tHD;STA | The hold time in start condition | 4.0 | | 0.6 | | µs |
| tLOW | The hold time in SCL clock "0" status | 4.7 | | 1.3 | | µs |
| tR | SCL, SDA signals' rising time | | 1000 | 20+0.1Cb | 300 | ns |
| tHD;DAT | Data hold time | 0 | | 0 | 0.9 | µs |
| tHIGH | The hold time in SCL clock "1" status | 4.0 | | 0.6 | | µs |
| tF | SCL, SDA signals' falling time | | 300 | 20+0.1Cb | 300 | ns |
| tsu;DAT | Data setup time | 250 | | 100 | | ns |
| tsu;STA | The setup time in restart condition | 4.7 | | 0.6 | | µs |
| tsu;STO | Stop condition setup time | 4.0 | | 0.6 | | µs |

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

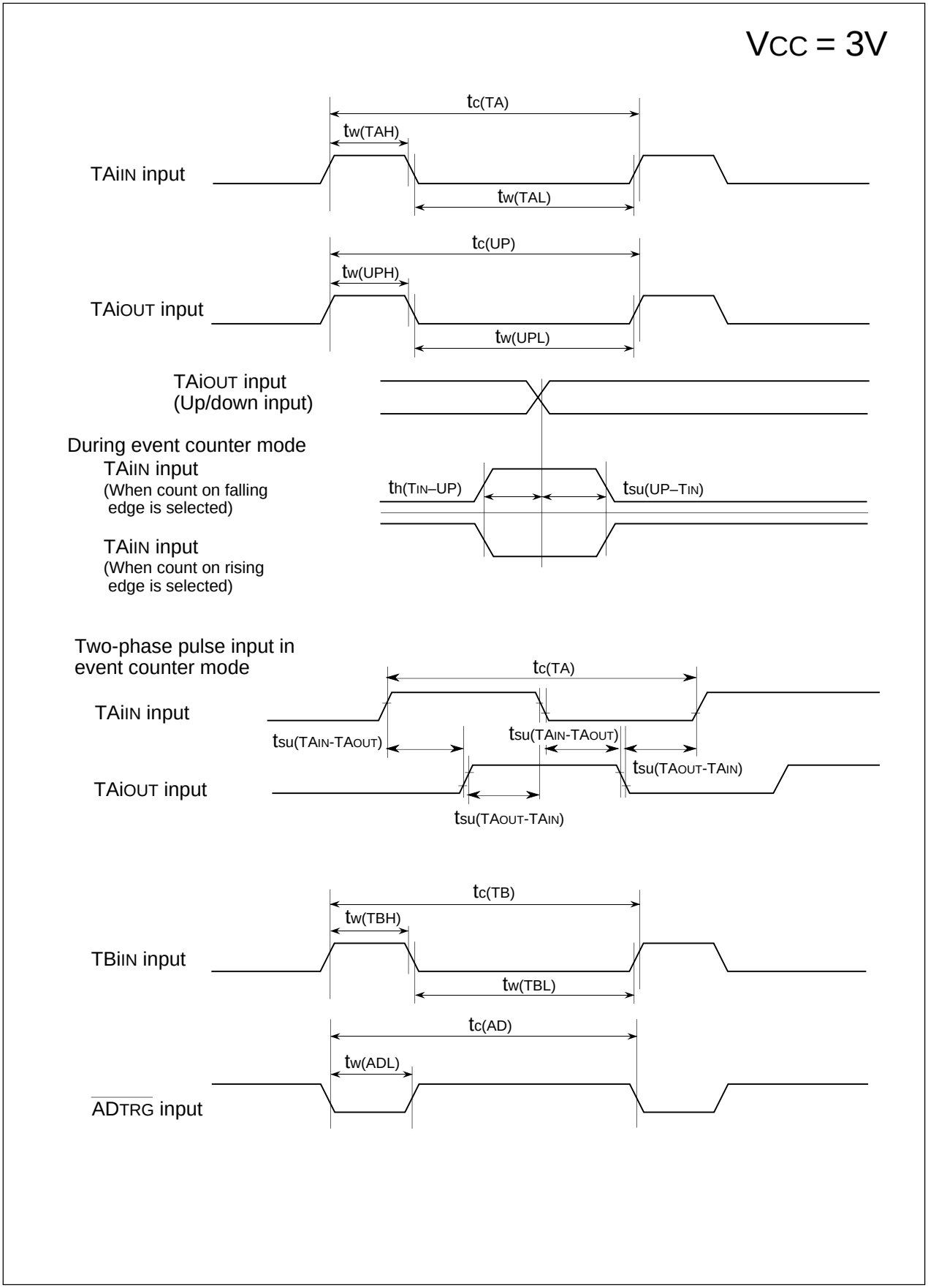



**Figure 18.10. Timing Diagram (1)**

**These standards are not final and focused only normal-version.**
**Should be used as a reference.**

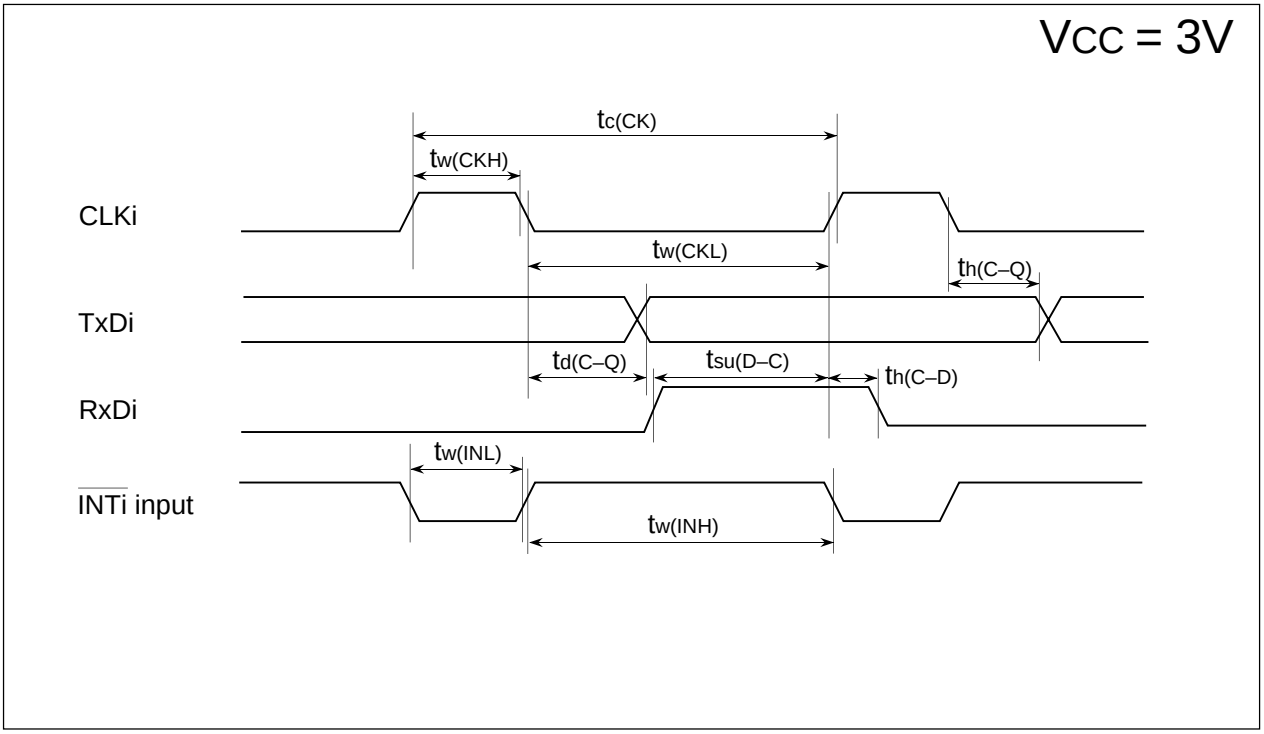



**Figure 18.11. Timing Diagram (2)**

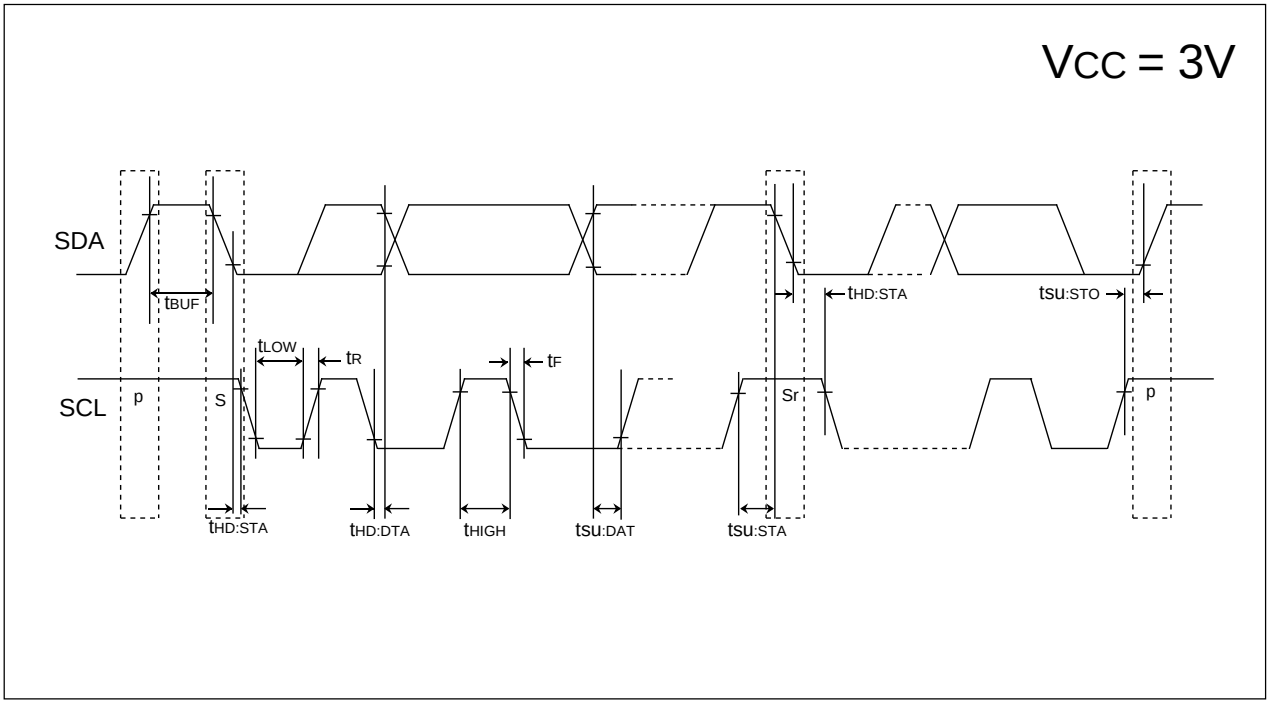



**Figure 18.12. Timing Diagram (3)**

# 19. Flash Memory Version

## 19.1 Flash Memory Performance

The flash memory version is functionally the same as the mask ROM version except that it internally contains flash memory.

In the flash memory version, the flash memory can perform in three mode : CPU rewrite mode, standard serial I/O mode and parallel I/O mode.

Table 19.1 shows the flash memory version specifications. (Refer to “Table 1.2.1 Performance Outline of M16C/28 Group (80-pin device)” for the items not listed in Table 19.1.” or “Table 1.2.2 Performance Outline of M16C/28 Group (64-pin device)”).

**Table 19.1. Flash Memory Version Specifications**

| Item | | Specification |
|---|---|---|
| Flash memory operating mode | | 3 modes (CPU rewrite, standard serial I/O, parallel I/O) |
| Erase block | | See Figure 19.2.1 to19.2.3 Flash Memory Block Diagram |
| Program method | | In units of word |
| Erase method | | Block erase |
| Program, erase control method | | Program and erase controlled by software command |
| Protect method | | All user blocks are write protected by bit FMR16.<br>In addition, the block 0 and block 1 are write protected by bit FMR02. |
| Number of commands | | 5 commands |
| Program/Erase Endurance(Note1) | Block 0 to 4 (program area) | 100 times 1,000 times (Option) |
| | Block A and B (data are) (Note2) | 100 times 10,000 times (Option) |
| Data Retention | | 10 years |
| ROM code protection | | Parallel I/O and standard serial I/O modes are supported. |

Note 1: Program and erase endurance definition
Program and erase endurance are the erase endurance of each block. If the program and erase endurance are n times (n=100,1,000,10,000), each block can be erased n times. For example, if a 2-Kbyte block A is erased after writing 1 word data 1024 times, each to different addresses, this is counted as one program and erasure. However, data cannot be written to the same address more than once without erasing the block. (Rewrite disabled)

Note 2: To use the limited number of erasure efficiently, write to unused address within the block instead of rewrite. Erase block only after all possible address are used. For example, an 8-word program can be written 128 times before erase is necessary. Maintaining an equal number of erasure between Block A and B will also improve efficiency. We recommend keeping track of the number of times erasure is used.

**Table 19.2. Flash Memory Rewrite Modes Overview**

| Flash memory rewrite mode | CPU rewrite mode | Standard serial I/O mode | Parallel I/O mode |
|---|---|---|---|
| Function | The user ROM area is rewritten when the CPU executes software command.<br>EW0 mode:<br>Rewrite in area other than flash memory<br>EW1 mode:<br>Rewrite in flash memory | The user ROM area is rewritten using a dedicated serial programmer.<br>Standard serial I/O mode 1:<br>Clock synchronous serial I/O<br>Standard serial I/O mode 2:<br>UART | The user ROM areas are rewritten using a dedicated parallel programmer. |
| Areas which can be rewritten | User ROM area | User ROM area | User ROM area |
| Operation mode | Single chip mode | Boot mode | Parallel I/O mode |
| ROM programmer | None | Serial programmer | Parallel programmer |

## 19.2 Memory Map

The flash memory contains the user ROM area and the boot ROM area (reserved area). Figures 19.2.1 to 19.2.3 show the flash memory block diagram. The user ROM area has space to store the microcomputer operation program in single-chip mode and a separate 2-Kbyte space as the block A and B.

The user ROM area is divided into several blocks. The user ROM area can be rewritten in CPU rewrite, standard serial input/output, and parallel input/output modes. However, if block 0 and 1 are rewritten in CPU rewrite mode, setting the FMR02 bit in the FMR0 register to “1” (block 0, 1 rewrite enabled) and the FMR16 bit in the FMR1 register to “1”(blocks 0 to 4 rewrite enabled) enable rewriting. Also, if blocks 2 to 4 are rewritten in CPU rewrite mode, setting the FMR16 bit in the FMR1 register to “1” (blocks 0 to 4 rewrite enabled) enables writing. Setting the PM10 bit in the PM1 register to “1”(data area access enabled) for block A and B enables to use.

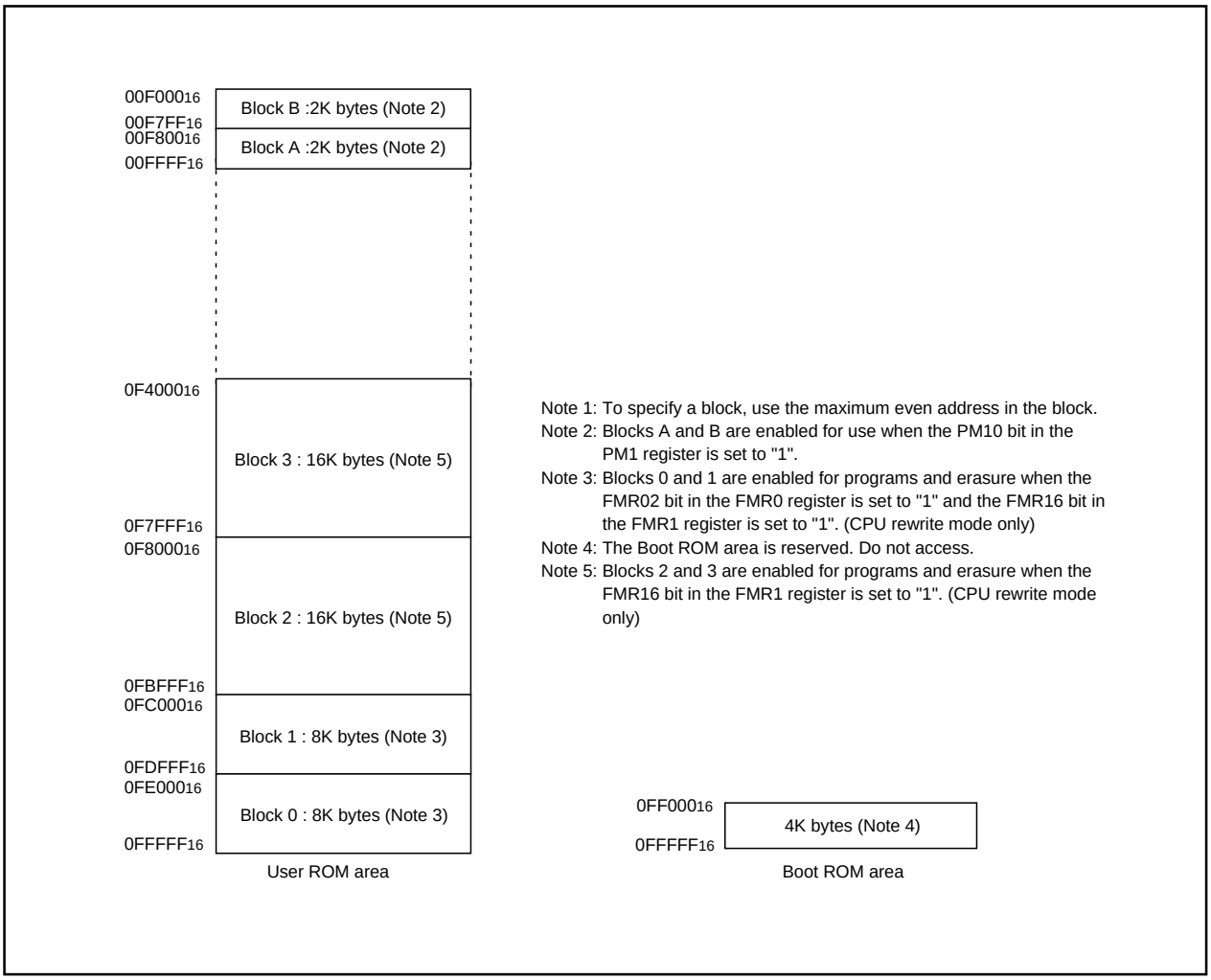



**Figure 19.2.1. Flash Memory Block Diagram (ROM capacity 48K byte)**

The M16C/28 (flash memory version) contains the flash memory that can be rewritten with a single voltage. For this flash memory, three flash memory modes area available in which to read, program, and erase: parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and a CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU). Each mode is detailed in the follwing sections.
The flash memory is divided into several blocks as shown in Figures 19.2.1 to 19.2.3, so that memory can be erased one block at time.
In addition to the user ROM area to store a microcomputer operation control program, the flash memory has a boot ROM area that is used to store a program to control rewriting in CPU rewrite and standard serial I/O mode. This boot ROM area has a standard serial I/O mode control program stored in it when shipped from the factory, which can be rewritten with a rewrite control program, to suit the user's application system.
When the CPU is shifted to the stop or wait modes, power to the internal flash memory is automatically shut off. It is reconnected automatically when CPU operation is restored.

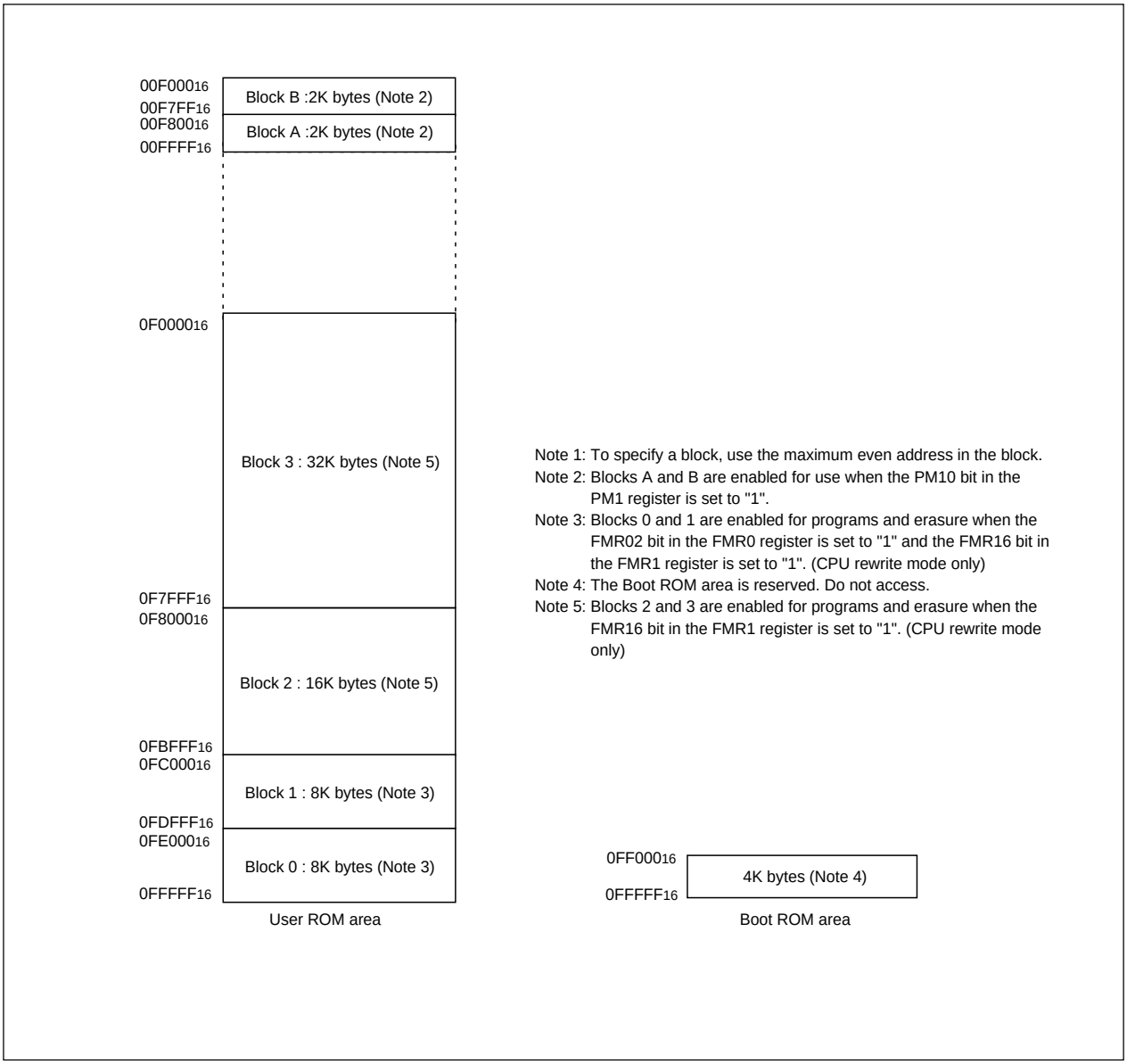



**Figure 19.2.2. Flash Memory Block Diagram (ROM capacity 64K byte)**

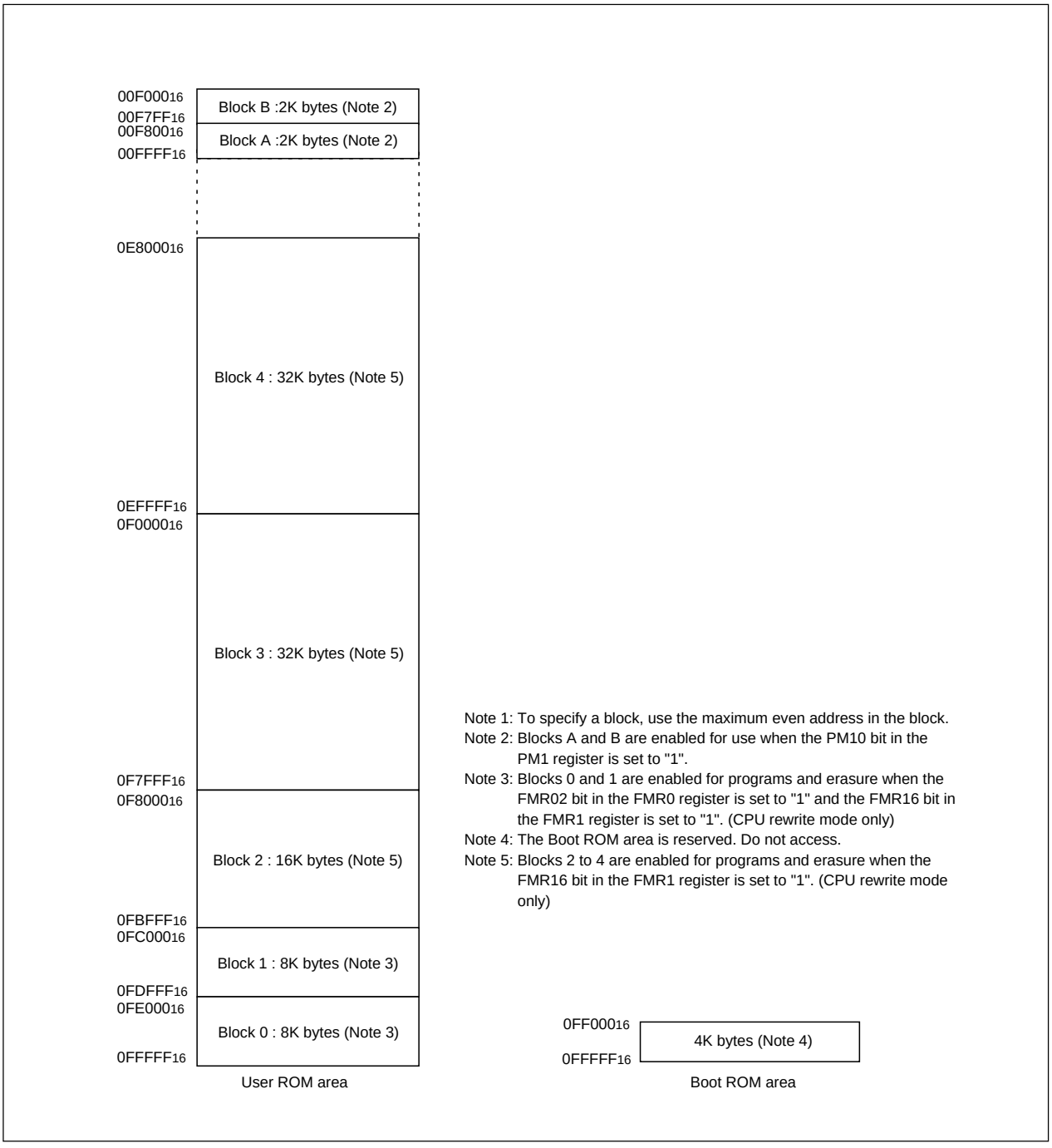



**Figure 19.2.3. Flash Memory Block Diagram (ROM capacity 96K byte)**

## 19.3 Functions To Prevent Flash Memory from Rewriting

The flash memory has a built-in ROM code protect function for parallel I/O mode and a built-in ID code check function for standard input/output mode to prevent the flash memory from reading or rewriting.

### 19.3.1 ROM Code Protect Function

The ROM code protect function prevents the flash memory from reading and rewriting in parallel input/output mode. Figure 19.3.1.1 shows the ROMCP register. The ROMCP register is located in the user ROM area. The ROMCP1 bit consists of two bits. The ROM code protect function is enabled and reading and rewriting flash memory is disabled when setting either or both of two ROMCP1 bits to “0” other than the ROMCR bit is ‘$00_2$’. However, when setting the ROMCR bit to ‘$00_2$’, the flash memory can be read or rewritten. Once the ROM code protect function is enabled, the ROMCR bits can not be changed in parallel input/output mode. Therefore, use the standard serial input/output or other modes to rewrite the flash memory.

### 19.3.2 ID Code Check Function

Use the ID code check function in standard serial input/output mode. Unless the flash memory is blank, the ID codes sent from the programmer and the seven bytes ID codes written in the flash memory are compared to see if they match. If the ID codes do not match, the commands sent from the programmer are not acknowledged. The ID code consists of 8-bit data, starting with the first byte, into addresses, $0FFFDF_{16}$, $0FFFE3_{16}$, $0FFFEB_{16}$, $0FFFEF_{16}$, $0FFFF3_{16}$, $0FFFF7_{16}$, and $0FFFFB_{16}$. The flash memory has a program with the ID code set in these addresses.

ROM code protect control address

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | Symbol | Address | Factory Setting |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 1 | 1 | 1 | 1 | ROMCP | 0FFFFF16 | FF16 (Note 4) |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| ——— | Reserved bit | Set this bit to “1” | RW |
| ——— | Reserved bit | Set this bit to “1” | RW |
| ——— | Reserved bit | Set this bit to “1” | RW |
| ——— | Reserved bit | Set this bit to “1” | RW |
| ROMCR | ROM code protect reset bit (Note 2, Note 4) | b5 b4<br>00: Disables protect<br>01:<br>10: } Enables ROMCP1 bit<br>11: | RW<br>RW |
| ROMCP1 | ROM code protect level 1 set bit (Note 1, Note 3, Note 4) | b7 b6<br>00:<br>01: } Enables protect<br>10:<br>11: Disables protect | RW<br>RW |

Note 1: When the ROMCR bits are set to other than ‘00$_2$’ and the ROMCP1 bits are set to other than ‘11$_2$’ (ROM code protect enabled), the flash memory is disabled against reading and rewriting in parallel input/output mode.
Note 2: When the ROMCR bits are set to ‘00$_2$’, the ROM code protect level 1 is reset. Because the ROMCR bits can not be modified in parallel input/output mode, modify in standard serial input/output mode.
Note 3: The ROMCP1 bits are valid when the ROMCR bits are ‘01$_2$’, ‘10$_2$’ or ‘11$_2$’.
Note 4: This bit can not be set to “1” once it is set to “0”. The ROMCP register is set to ‘FF$_{16}$’ when a block, including the ROMCP register, is erased.

**Figure 19.3.1.1. ROMCP Register**

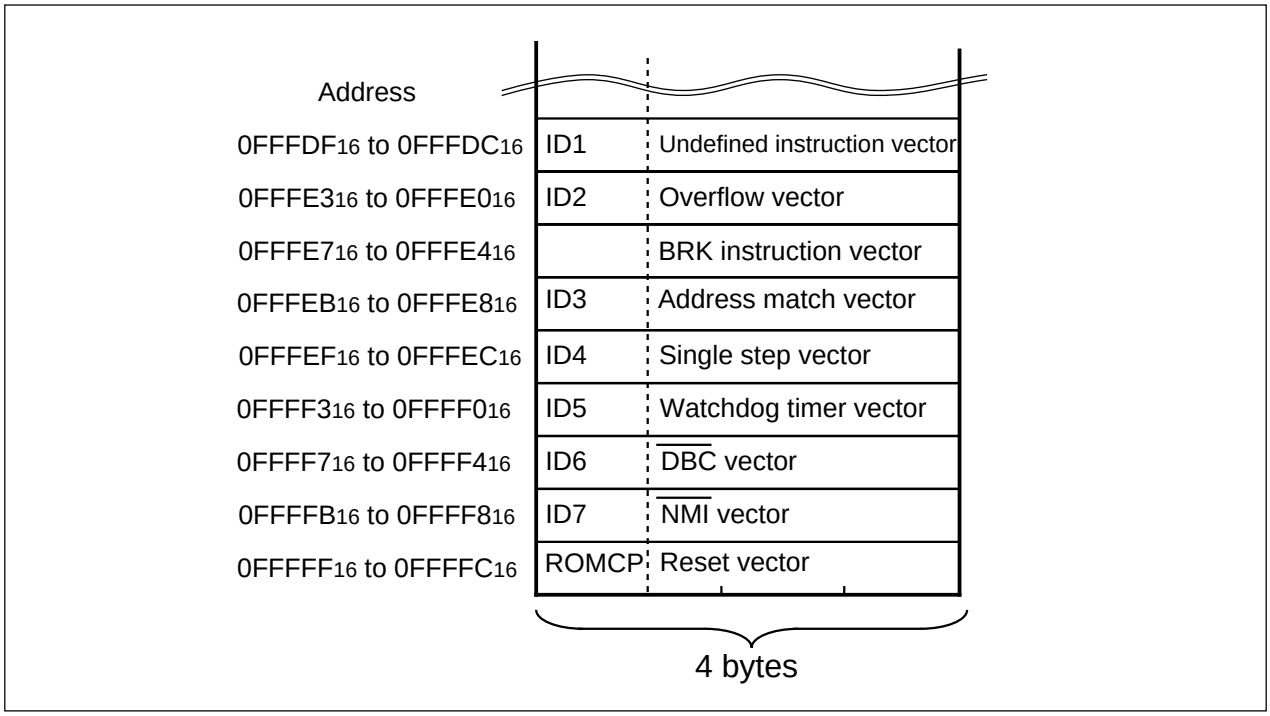



**Figure 19.3.2.1. Address for ID Code Stored**

## 19.4 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten when the CPU executes software commands. In CPU rewrite mode, only the user ROM area shown in Figure 19.2.1 to 19.2.3 can be rewritten and the boot ROM area cannot be rewritten. Verify the Program and the Block Erase commands are executed only on blocks in the user ROM area. Therefore, the user ROM area can be rewritten directly while the microcomputer is mounted on-board without using a ROM programmer, etc.

For interrupts requested during an erasing operation in CPU rewrite mode, the M16C/28 flash module offers an erase-suspend function which the erasing operation to be suspended, and access made available to the flash. Erase-write 0 (EW0) mode and erase-write 1 (EW1) mode are provided as CPU rewrite mode. Table 19.4.1 shows the differences between erase-write 0 (EW0) and erase-write 1 (EW1) modes. 1 wait is required for the CPU erase-write control.

Table 19.4.1. EW0 Mode and EW1 Mode

| Item | EW0 mode | EW1 mode(Note 2) |
|---|---|---|
| Operation mode | Single chip mode | Single chip mode |
| Areas in which a rewrite control program can be located | User ROM area | User ROM area |
| Areas where rewrite control program can be executed | The rewrite control program must be transferred to any other than the flash memory (e.g., RAM) before being executed | The rewrite control program can be excuted in the user ROM area |
| Areas which can be rewritten | User ROM area | User ROM area<br>However, this excludes blocks with the rewrite control program |
| Software command Restrictions | None | • Program, block erase command<br>Cannot be executed in a block having the rewite control program<br>• Read Status Register command<br>Cannot be executed |
| Mode after programming or erasing | Read Status Register mode | Read Array mode |
| CPU state during auto-write and auto-erase | Operating | Hold state (I/O ports retain the state before the command is excuted<br>(Note 1) |
| Flash memory status detection(Note 2) | • Read the FMR0 register's FMR00, FMR06, and FMR07 bits in the FMR0 register by program<br>• Execute the read status register command to read the SR7, SR5, and SR4 bits. | Read the FMR0 register's FMR00, FMR06, and FMR07 bits in a program |
| Condition for transferring to erase-suspend(Note 3) | Set the FMR40 and FMR41 bits in the FMR4 register to "1" by program. | The FMR40 bit in the FMR4 register is set to "1" and the interruput request of |

Note 1: Do not generate a DMA transfer.
Note 2: Block 1 and Block 0 are enabled for rewrite by setting FMR02 bit in the FMR0 register to "1" and setting FMR16 bit in the FMR1 register to "1". Block 2 to Block 4 are enabled for rewrite by setting FMR16 bit in the FMR1 register to "1".
Note 3: The time, until entering erase suspend and reading flash is enabled, is maximum td(SR-ES) after satisfying the conditions

### 19.4.1 EW0 Mode

The microcomputer enters CPU rewrite mode by setting the FMR01 bit in the FMR0 register to “1” (CPU rewrite mode enabled) and is ready to acknowledge the software commands. EW0 mode is selected by setting the FMR11 bit in the FMR1 register to “0”. When setting the FMR01 bit to “1”, set to “1” after first writing “0”. The software commands control programming and erasing. The FMR0 register or the status register indicates whether a programming or erasing operations is completed. When entering the erase-suspend during the auto-erasing, set the FMR40 bit to “1” (erase-suspend enabled) and the FMR41 bit to “1” (suspend request). And wait for td(SR-ES). After verifying the FMR46 bit is set to “1” (auto-erase stop), access to the user ROM area. When setting the FMR41 bit to “0” (erase restart), auto-erasing is restarted.

### 19.4.2 EW1 Mode

EW1 mode is selected by setting the FMR11 bit to “1” after the FMR01 bit is set to “1”. (set to “1” after first writing “0”). The FMR0 register indicates whether or not a programming or an erasing operation is completed. Do not execute the software commands of read status register in EW1 mode. When enabling an erase suspend function, set the FMR40 bit to “1” (erase suspend enabled) and execute block erase commands. Also, preliminarily set an interrupt to enter the erase-suspend to an interrupt enabled status. After td(SR-ES) from an interrupt request and entering erase suspend, an interrupt can be acknowledged. When an interrupt request is generated, the FMR41 bit is automatically set to “1” (suspend request) and an auto-erasing is halted. If an auto-erasing is not completed (the FMR00 bit is “0”) after an interrupt process completed, set the FMR41 bit to “0” (erase restart) and execute block erase commands again.

## 19.5 Register Description

Figure 19.5.1 shows the flash memory control register 0 and flash memory control register 1. Figure 19.5.2 shows the flash memory control register 4.

### 19.5.1 Flash memory control register 0 (FMR0):

**•FMR 00 Bit**

This bit indicates the operation status of the flash memory. The bit is “0” during programming, erasing, or erase-suspend mode; otherwise, the bit is “1”.

**•FMR01 Bit**

The microcomputer enables to acknowledge commands by setting the FMR01 bit to “1” (CPU rewrite mode). To set this bit to “1”, it is necessary to set to “0” after first setting to “1”. To set this bit to “0” by only writing “0”.

**•FMR02 Bit**

The combined setting of the FMR02 bit and the FMR16 bit enable to program and erase in the user ROM area. See Table 19.5.2.1 for setting details. To set this bit to “1”, it is necessary to set to “0” after first setting to “1”. To set this bit to “0” by only writing “0”. This bit is enabled only when the FMR01 bit is “1” (CPU rewrite mode enable).

**•FMSTP Bit**

This bit resets the flash memory control circuits and minimizes power consumption in the flash memory. Access to the flash memory is disabled when the FMSTP bit is set to “1”. Set the FMSTP bit by a program in a space other than the flash memory.

Set the FMSTP bit to “1” if one of the following occurs:

•A flash memory access error occurs during erasing or programming in EW0 mode (FMR00 bit does not switch back to “1” (ready)).

•Low-power consumption mode or ring oscillator low-power consumption mode is entered. Figure 19.5.1.3 shows a flow chart illustrating how to start and stop the flash memory before and after entering low power mode. Follow the procedure on this flow chart.

**•FMR06 Bit**

This is a read-only bit indicating an auto-program operation status. This bit is set to “1” when a program error occurs; otherwise, it is set to “0”. For details, refer to the description of the full status check.

**•FMR07 Bit**

This is a read-only bit indicating an auto-erase operation status. The bit is set to “1” when an erase error occurs; otherwise, it is set to “0”. For details, refer to the description of the full status check.

Figure 19.5.1.1 shows a EW0 mode set/reset flowchart, figure 19.5.1.2 shows a EW1 mode set/reset flowchart.

## 19.5.2 Flash memory control register 1 (FMR1):

**•FMR11 Bit**

EW1 mode is entered by setting the FMR11 bit to “1” (EW1 mode). This bit is enabled only when the FMR01 bit is “1”.

**•FMR16 Bit**

The combined setting of the FMR02 bit and the FMR16 bit enables to program and erase in the user ROM area. To set this bit to “1”, it is necessary to set to “0” after first setting to “1”. To set this bit to “0” by only writing “0”. This bit is enabled only when the FMR01 bit is “1”.

**•FMR17 Bit**

FMR17 bit is “1” (with wait state), regardless of the content of the PM17 bit, 1 wait is inserted at the access to block A and block B. Regardless of the content of the FMR17 bit, access to other block and the internal RAM becomes the PM17 bit setting.

Set this bit to “1” (with wait state) when rewriting more than 100 times (Option).

**Table 19.5.2.1. Protection using FMR16 and FMR02**

| FMR16 | FMR02 | Block A, Block B | Block 0, Block 1 | other user block |
|---|---|---|---|---|
| 0 | 0 | write allowed | write protected | write protected |
| 0 | 1 | write allowed | write protected | write protected |
| 1 | 0 | write allowed | write protected | write allowed |
| 1 | 1 | write allowed | write allowed | write allowed |

## 19.5.3 Flash memory control register 4 (FMR4):

**•FMR40 Bit**

The erase-suspend function is enabled by setting the FMR40 bit is set to “1” (enabled).

**•FMR41 Bit**

When setting the FMR41 bit to “1” in a program during auto-erasing in EW0 mode the flash module enters erase suspend mode. In EW1 mode, the FMR41 bit is automatically set to “1” (suspend request) when an interrupt request of an enabled interrupt is generated, the FMR41 bit is automatically set to “1” (suspend request) and when an auto-erasing operation is restarted, set the FMR41 bit to “0” (erase restart).

**•FMR46 Bit**

The FMR46 bit is set to “0” during auto-erasing execution and set to “1” during erase-suspend mode. Do not access to flash memory while this bit is “0”.

Flash memory control register 0

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

| Symbol | Address | After reset |
|---|---|---|
| FMR0 | $01B7_{16}$ | $XX000001_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| FMR00 | RY/$\overline{BY}$ status flag | 0: Busy (during writing or erasing)<br>1: Ready | RO |
| FMR01 | CPU rewrite mode select bit (Note1) | 0: Disables CPU rewrite mode (Disables software command)<br>1: Enables CPU rewrite mode (Enables software commands) | RW |
| FMR02 | Block 0, 1 rewrite enable bit (Note 2) | Set write protection for user ROM area (see Table 19.5.2.1) | RW |
| FMSTP | Flash memory stop bit (Note 3, 5) | 0: Starts flash memory operation<br>1: Stops flash memory operation (Enters low-power consumption state and flash memory reset) | RW |
| — (b5-b4) | Reserved bit | Set to “0” | RW |
| FMR06 | Program status flag (Note 4) | 0: Terminated normally<br>1: Terminated in error | RO |
| FMR07 | Erase status flag (Note 4) | 0: Terminated normally<br>1: Terminated in error | RO |

Note 1: When setting this bit to “1”, set to “1” immdediately after setting it first to “0”. Do not generate an interrupt or a DMA transfer between setting the bit to “0” and setting it to “1”. Set this bit while the P8₅/NMI/SD pin is “H” when selecting the NMI function. Set by program in a space other than the flash memory in EW0 mode. Set this bit to read alley mode and “0”
Note 2: Set this bit to “1” immediately after setting it first to “0” while the FMR01 bit is set to “1”. Do not generate an interrupt or a DMA transfer between setting this bit to “0” and setting it to “1”.
Note 3: Set this bit by a program in a space other than the flash memory.
Note 4: This bit is set to “0” by executing the clear status command.
Note 5: This bit is enabled when the FMR01 bit is set to “1” (CPU rewrite mode). This bit can be set to “1” when the FMR01 bit is set to “1”. However, the flash memory does not enter low-power consumption status and it is not initialized.

Flash memory control register 1

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

| Symbol | Address | After reset |
|---|---|---|
| FMR1 | $01B5_{16}$ | $000XXX0X_2$ |

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| — (b0) | Reserved bit | When read, its content is indeterminate | RO |
| FMR11 | EW1 mode select bit (Note1) | 0: EW0 mode<br>1: EW1 mode | RW |
| — (b3-b2) | Reserved bit | When read, its content is indeterminate | RO |
| — (b4) | Nothing is assigned. When write, set to “0”. When read, its contect is indeterminate. | | — |
| — (b5) | Reserved bit | Set to “0” | RW |
| FMR16 | Block 0 to 3 rewrite enable bit (Note2) | Set write protection for user ROM area (see Table 19.5.2.1)<br>0: Disable<br>1: Enable | RW |
| FMR17 | Block A, B access wait bit ( Note 3) | 0: PM17 enabled<br>1: With wait state (1 wait) | RW |

Note 1: Set this bit to “1” immediately after setting it first to “0”. Do not generate an interrupt or a DMA transfer between setting the bit to “0” and setting it to “1”. Set this bit while the P8₅/NMI/SD pin is “H” when the NMI function is selected. If the FMR01 bit is set to “0”, the FMR01 bit and FMR11 bit are both set to “0”
Note 2: Set this bit to “1” immediately after setting it first to “0”. Do not generate an interrupt or a DMA transfer after setting to “0”.
Note 3: When rewriting more than 100 times, set this bit to “1” (with wait state). When the FMR17 bit is “1” (with wait state), regardless of the content of the PM17 bit, 1 wait is inserted at the access to the block A and B.

**Figure 19.5.1. Flash memory control register 0,1**

Flash memory control register 4

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

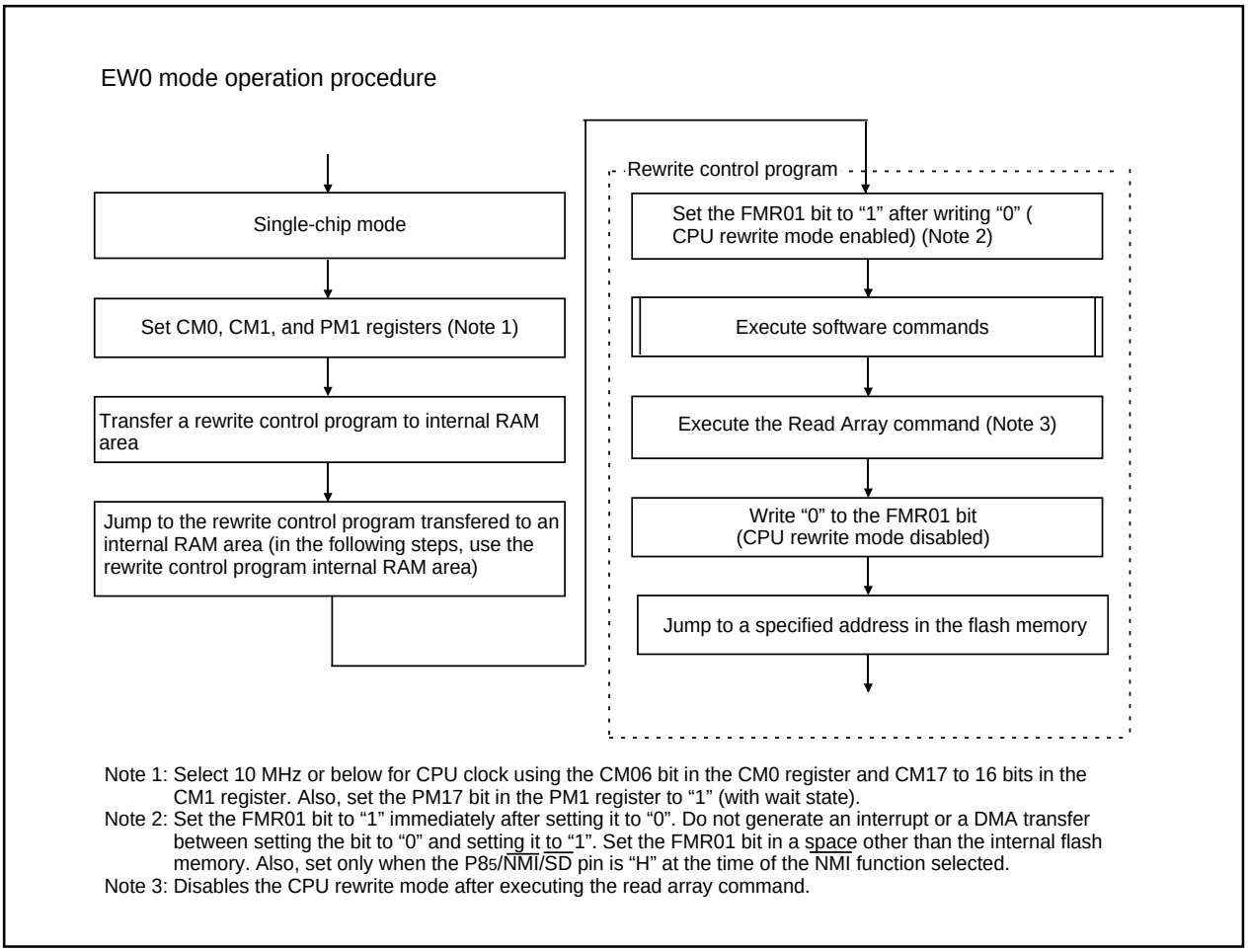
Symbol: FMR4 | Address: $01B3_{16}$ | After reset: $01000000_2$

| Bit symbol | Bit name | Function | RW |
|---|---|---|---|
| FMR40 | Erase suspend function enable bit (Note 1) | 0: Disabled<br>1: Enabled | RW |
| FMR41 | Erase suspend request bit (Note 2) | 0: Erase restart<br>1: Suspend request | RW |
| $\overline{(b5\text{-}b2)}$ | Reserved bit | Set to “0” | RO |
| FMR46 | Erase status | 0: During auto-erase operation<br>1: Auto-erase stop<br>(erase suspend mode) | RO |
| $\overline{(b7)}$ | Reserved bit | Set to “0” | RW |

Note 1: When setting this bit to “1”, set to “1” immediately after setting it first to “0”. Do not generate an interrupt or a DMA transfer between setting the bit to “0” and setting it to “1”. Set by a program in a space other than the flash memory in EW0 mode.

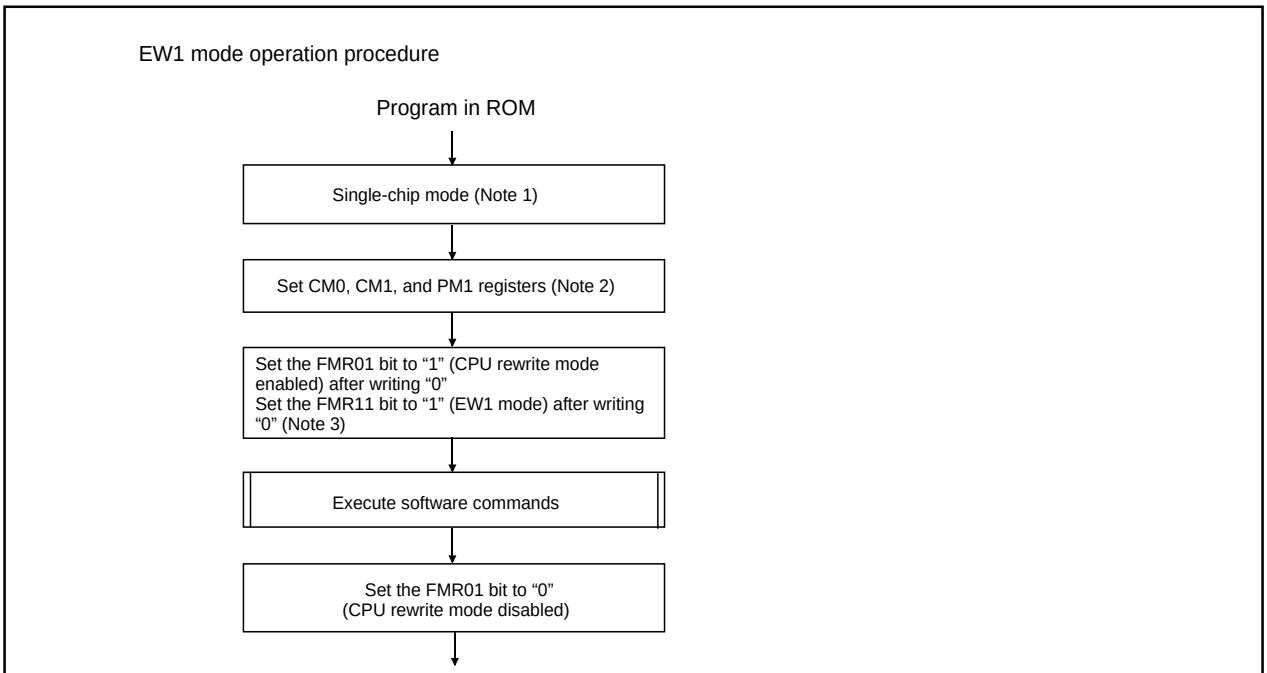
Note 2: This bit is valid only when the erase-suspend enable bit (FMR40) is “1”. Writing is enabled only between executing an erase command and completing erase (this bit is set to “1” other than the above duration). This bit can be set to “0” or “1” by a program in EW0 mode. In EW1 mode, this bit is automatically set to “1” when the FMR40 bit is “1” and a maskable interrupt is generated during erasing. Do not write to “1” by a program (writing “0” is enabled).

**Figure 19.5.2. Flash memory control register 4**



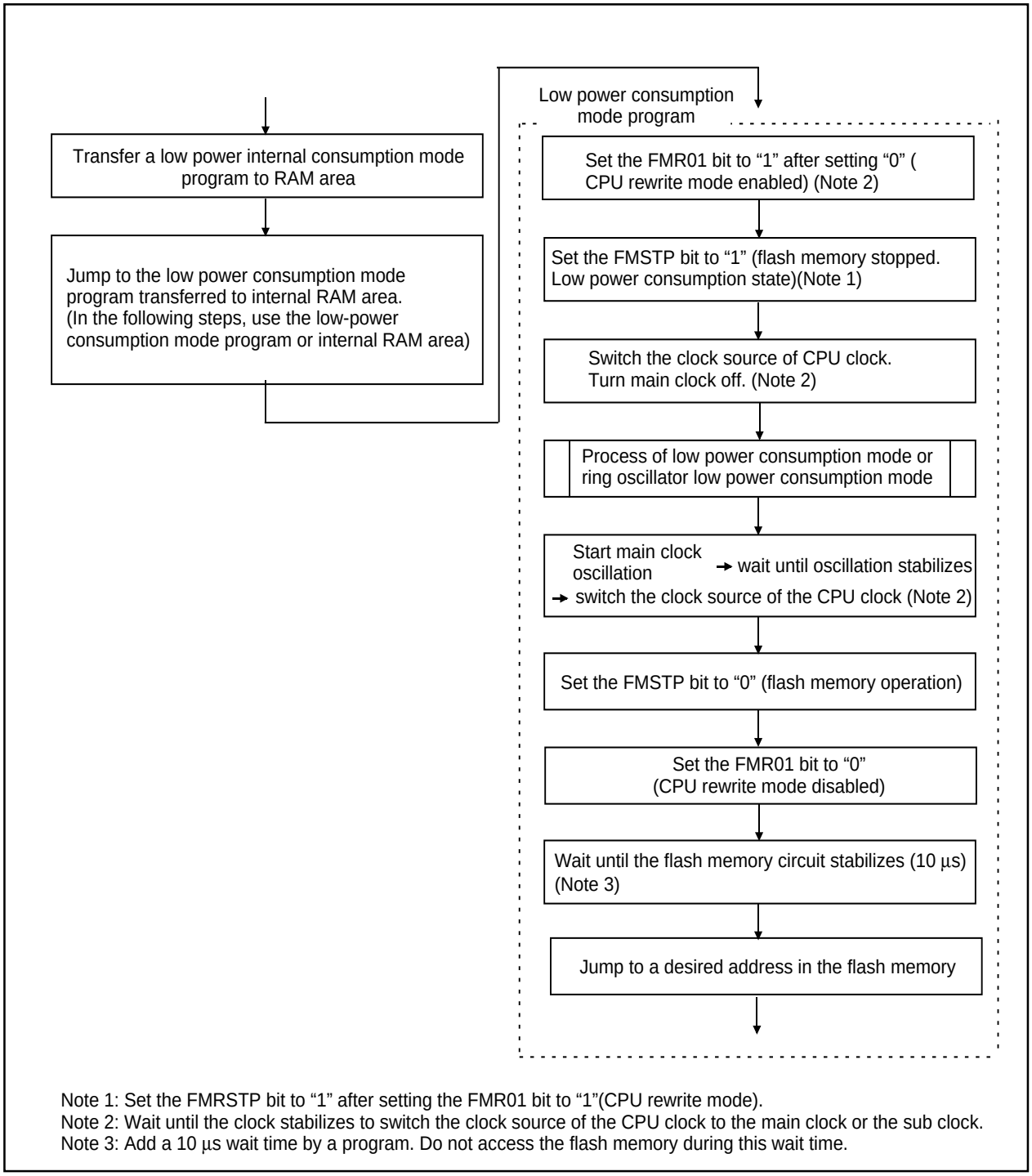

Note 1: Select 10 MHz or below for CPU clock using the CM06 bit in the CM0 register and CM17 to 16 bits in the CM1 register. Also, set the PM17 bit in the PM1 register to “1” (with wait state).
Note 2: Set the FMR01 bit to “1” immediately after setting it to “0”. Do not generate an interrupt or a DMA transfer between setting the bit to “0” and setting it to “1”. Set the FMR01 bit in a space other than the internal flash memory. Also, set only when the P85/NMI/SD pin is “H” at the time of the NMI function selected.
Note 3: Disables the CPU rewrite mode after executing the read array command.

**Figure 19.5.1.1. Setting and Resetting of EW0 Mode**




Note 1: In EW1 mode, do not set boot mode.
Note 2: Select 10 MHz or below for CPU clock using the CM06 bit in the CM0 register and CM17 to 16 bits. in the CM1 register. Also, set the PM17 bit in the PM1 register to “1” (with wait state).
Note 3: Set the FMR01 bits to “1” immediately after setting it to “0”. Do not generate an interrupt or a DMA transfer between setting the bit to “0” and setting the bit to “1”. Set the FMR01 bit in a space other than the internal flash memory. Set only when the P85/NMI/SD pin is “H” at the time of the NMI function selected.

**Figure 19.5.1.2. Setting and Resetting of EW1 Mode**




Note 1: Set the FMRSTP bit to “1” after setting the FMR01 bit to “1”(CPU rewrite mode).
Note 2: Wait until the clock stabilizes to switch the clock source of the CPU clock to the main clock or the sub clock.
Note 3: Add a 10 µs wait time by a program. Do not access the flash memory during this wait time.

**Figure 19.5.1.3. Processing Before and After Low Power Dissipation Mode**

## 19.6 Precautions in CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

### 19.6.1 Operation Speed

When CPU clock source is the main clock, before entering CPU rewrite mode (EW0 or EW1 mode), select 10 MHz or below for CPU clock using the CM06 bit in the CM0 register and the CM17 to CM16 bits in the CM1 register. Also, when selecting $f_3$(ROC) of a ring oscillator as a CPU clock source, before entering CPU rewrite mode (EW0 or EW1 mode), the ROCR3 to ROCR2 bits in the ROCR register set the CPU clock division rate to “divide-by-4” or “divide-by-8”.

On both cases, set the PM17 bit in the PM1 register to “1” (with wait state).

### 19.6.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because the CPU tries to read data in the flash memory: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

### 19.6.3 Interrupts

EW0 Mode

- To use interrupts having vectors in a relocatable vector table, the vectors must be relocated to the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts can be used since the FMR0 and FMR1 registers are forcibly reset when either interrupt is generated. However, the jump addresses for each interrupt service routines to the fixed vector table are set and interrupt programs are required. Flash memory rewrite operation is halted when the NMI or watchdog timer interrupt is generated. Set the FMR01 bit to “1” and execute the rewrite and erase program again after exiting the interrupt routine.
- The address match interrupt can not be used since the CPU tries to read data in the flash memory.

EW1 Mode

- Do not acknowledge any interrupts with vectors in the relocatable vector table or the address match interrupt during the auto-program or erase-suspend function.

### 19.6.4 How to Access

To set the FMR01, FMR02, or FMR11 bit to “1”, write “1” after first setting the bit to “0”. Do not generate an interrupt or a DMA transfer between the instruction to set the bit to “0” and the instruction to set it to “1”. Set the bit while an “H” signal is applied to the NMI pin.

### 19.6.5 Writing in the User ROM Space

#### 19.6.5.1 EW0 Mode

- If the supply voltage drops while rewriting the block where the rewrite control program is stored, the flash memory can not be rewritten, because the rewrite control program is not correctly rewritten. If this error occurs, rewrite the user ROM area in standard serial I/O mode or parallel I/O mode.

#### 19.6.5.2 EW1 Mode

- Do not rewrite the block where the rewrite control program is stored.

### 19.6.6 DMA Transfer

In EW1 mode, do not perform a DMA transfer while the FMR00 bit in the FMR0 register is set to “0”. (the auto-programming or auto-erasing duration ).

### 19.6.7 Writing Command and Data

Write the command code and data to even addresses in the user ROM area.

### 19.6.8 Wait Mode

When entering wait mode, set the FMR01 bit to “0” (CPU rewrite mode disabled) before executing the WAIT instruction.

### 19.6.9 Stop Mode

When entering stop mode, the following settings are required:

• Set the FMR01 bit to “0” (CPU rewrite mode disabled) and disable the DMA transfer before setting the CM10 bit to “1” (stop mode).

• Execute the instruction to set the CM10 bit to “1” (stop mode) and the JMP.B instruction.

```
Program example     BSET    0, CM1      ; Stop mode
                    JMP.B   L1
                L1:
                    Program after exiting stop mode
```

### 19.6.10 Low Power Consumption Mode and Ring Oscillator-Low Power Consumption Mode

If the CM05 bit is set to “1” (main clock stopped), do not execute the following commands.

• Program

• Block erase

## 19.7 Software Commands

Read or write 16-bit commands and data from or to even addresses in the user ROM area. When writing a command code, 8 high-order bits ($D_{15}$–$D_8$) are ignored.

**Table 19.7.1. Software Commands**

| Command | First bus cycle | | | Second bus cycle | | |
|---|---|---|---|---|---|---|
| | Mode | Address | Data ($D_{15}$ to $D_0$) | Mode | Address | Data ($D_{15}$ to $D_0$) |
| Read array | Write | X | xxFF16 | | | |
| Read status register | Write | X | xx7016 | Read | X | SRD |
| Clear status register | Write | X | xx5016 | | | |
| Program | Write | WA | xx4016 | Write | WA | WD |
| Block erase | Write | X | xx2016 | Write | BA | xxD016 |

SRD: Status register data ($D_7$ to $D_0$)
WA : Write address (However,even address)
WD : Write data (16 bits)
BA : Highest-order block address (However,even address)
X : Any even address in the user ROM area
xx : 8 high-order bits of command code (ignored)

### 19.7.1 Read Array Command (FF16)

This command reads the flash memory.

By writing command code 'xxFF16' in the first bus cycle, read array mode is entered. Content of a specified address can be read in 16-bit unit after the next bus cycle. The microcomputer remains in read array mode until an another command is written. Therefore, contents of multiple addresses can be read consecutively.

### 19.7.2 Read Status Register Command (7016)

This command reads the status register.

By writing command code 'xx7016' in the first bus cycle, the status register can be read in the second bus cycle (Refer to "Status Register"). Read an even address in the user ROM area. Do not execute this command in EW1 mode.

### 19.7.3 Clear Status Register Command ($50_{16}$)

This command clears the status register to “0”.

By writing ‘xx$50_{16}$’ in the first bus cycle, and the FMR06 to FMR07 bits in the FMR0 register and SR4 to SR5 bits in the status register are set to “0”.

### 19.7.4 Program Command ($40_{16}$)

The program command writes 2-byte data to the flash memory. By writing ‘xx$40_{16}$’ in the first bus cycle and data to the write address specified in the second bus cycle, the auto-programming/erasing (data prorgramming and verify) start. Set the address value specified in the first bus cycle to same and even address as the write address specified in the second bus cycle. The FMR00 bit in the FMR0 register indicates whether an auto-programming operation has been completed. The FMR00 bit is set to “0” during the auto-programming and “1” when the auto-programming operation is completed. After the auto-programming operation is completed, the FMR06 bit in the FMR0 register indicates whether or not the auto-programming operation has been completed as expected. (Refer to “Full Status Check”). Also, each block disables writing (Refer to “Table 19.5.2.1”). Do not write additions to the address which is already programmed. When commands other than a program command are executed immediately after a program command, set the same address as the write address specified in the second bus cycle of the program command, to the specified address value in the first bus cycle of the following command. In EW1 mode, do not execute this command on the blocks where the rewrite control program is allocated. In EW0 mode, the microcomputer enters read status register mode as soon as the auto-programming operation starts and the status register can be read. The SR7 bit in the status register is set to “0” as soon as the auto-programming operation starts. This bit is set to “1” when the auto-programming operation is completed. The microcomputer remains in read status register mode until the read array command is written. After completion of the auto-programming operation, the status register indicates whether or not the auto-programming operation has been completed as expected.

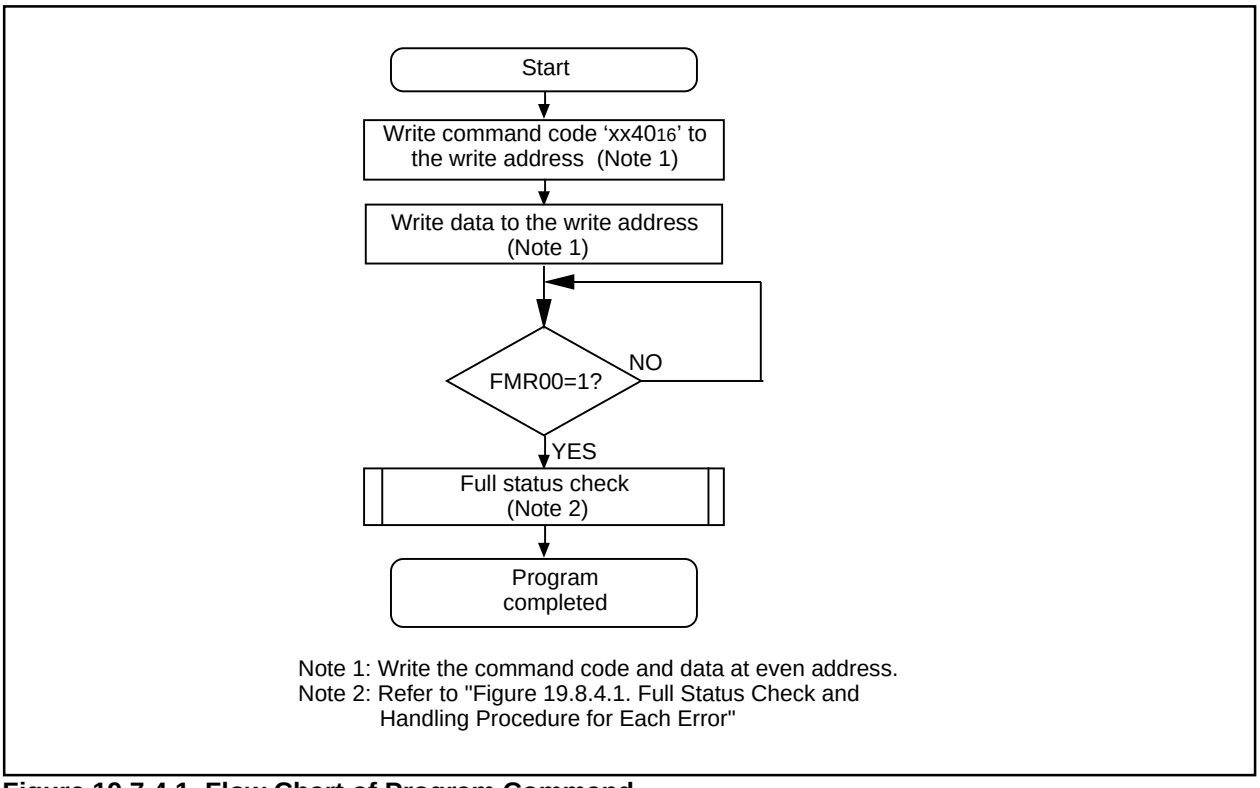
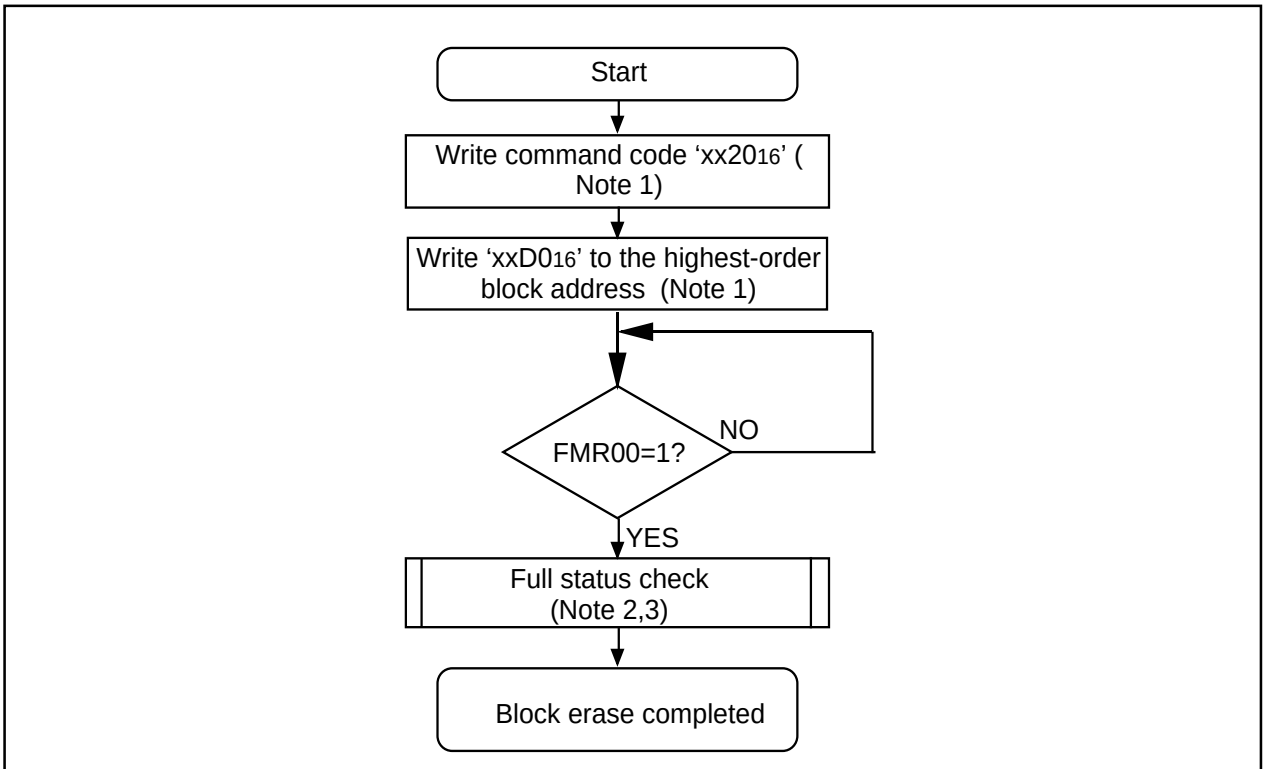



**Figure 19.7.4.1. Flow Chart of Program Command**

### 19.7.5 Block Erase

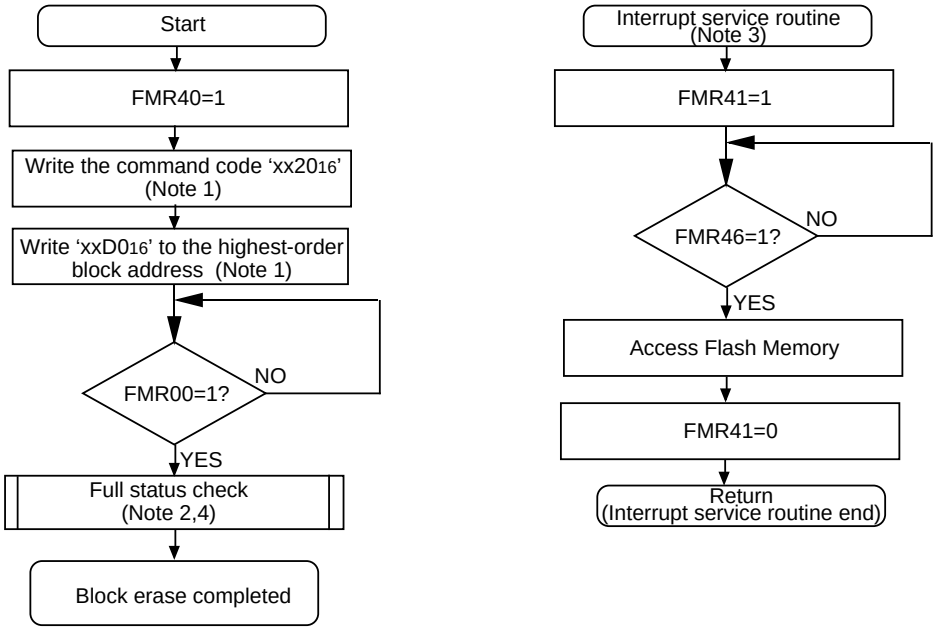
By writing 'xx20$_{16}$' in the first bus cycle and 'xxD0$_{16}$' in the second bus cycle to the highest-order (even addresse of a block) and the auto-programming/erasing (erase and erase verify) start. The FMR00 bit in the FMR0 register indicates whether the auto-programming operation has been completed. The FMR00 bit is set to "0" during the auto-erasing operation and "1" when the auto-erasing operation is completed. When using the erase-suspend function in EW0 mode, the FMR46 bit in the FMR4 register indicates whether a flash memory has entered erase-suspend mode. The FMR46 bit is set to "0" during auto-erasing operation and "1" when the auto-erasing operation is completed (entering erase-suspend). After the completion of an auto-erasing operation, the FMR07 bit in the FMR0 register indicates whether or not the auto erasing-operation has been completed as expected. (Refer to "Full Status Check"). Also, each block disables erasing. (Refer to "Table 19.5.2.1"). Figure 19.7.5.1 shows a flow chart of the block erase command programming when not using the erase-suspend function. Figure 19.7.5.2 shows a flow chart of the block erase command programming when using an erase-suspend function. In EW1 mode, do not execute this command on the block where the rewrite control program is allocated. In EW0 mode, the microcomputer enters read status register mode as soon as the auto-erasing operation starts and the status register can be read. The SR7 bit in the status register is set to "0" as soon as the auto-erasing operation starts. This bit is set to "1" when the auto-erasing operation is completed. The microcomputer remains in read status register mode until the read array command is written. Also, execute the clear status register command and the block erase command at least 3 times until the erase error is not generated when an erase error is not generated.



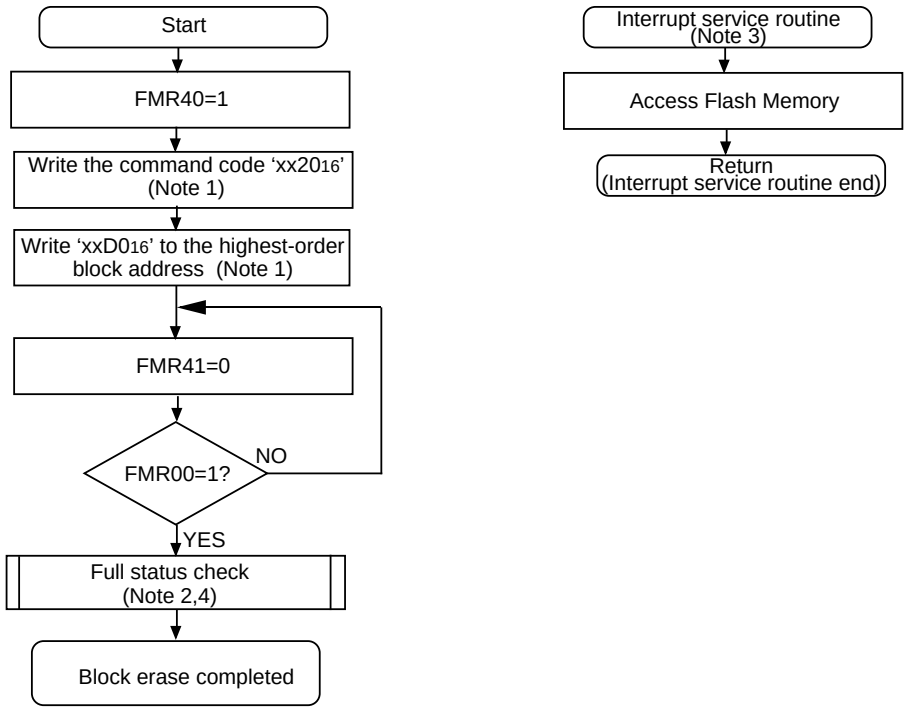

Note 1: Write the command code and data at even address.
Note 2: Refer to "Figure 19.8.4.1. Full Status Check and Handling Porcedure for Each Error".
Note 3: Execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.

**Figure 19.7.5.1. Flow Chart of Block Erase Command (when not using erase suspend function)**

(EW0 mode)




(EW1 mode)




Note 1: Write the command code and data to even address.
Note 2: Execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.
Note 3: In EW0 mode, allocate an interrupt vector table of an interrupt, to be used, to a RAM area
Note 4: Refer to "Figure 19.8.4.1. Full Status Check and Handling Porcedure for Each Error".

**Figure 19.7.5.2. Block Erase Command (at use erase suspend)**

## 19.8 Status Register

The status register indicates the operating status of the flash memory and whether an erasing or a programming operates normally and an error ends. The FMR00, FMR06, and FMR07 bits in the FMR0 register indicate the status of the status register.

Table 19.8.1 shows the status register.

In EW0 mode, the status register can be read in the following cases:

(1) When a given even address in the user ROM area is read after writing the read status register command

(2) When a given even address in the user ROM area is read after executing the program or block erase command but before executing the read a rray command.

### 19.8.1 Sequence Status (SR7 and FMR00 Bits )

The sequence status indicates the operating status of the flash memory. This bit is set to “0” (busy) during an auto-programming and auto-erasing and “1” (ready) as soon as these operations are completed. This bit indicates “0” (busy) in erase-suspend mode.

### 19.8.2 Erase Status (SR5 and FMR07 Bits)

Refer to “Full Status Check.”

### 19.8.3 Program Status (SR4 and FMR06 Bits)

Refer to “Full Status Check.”

**Table 19.8.1. Status Register**

| Bits in the SRD register | Bits in the FMR0 register | Status name | Contents | | Value after reset |
|---|---|---|---|---|---|
| | | | "0" | "1" | |
| SR7 ($D_7$) | FMR00 | Sequence status | Busy | Ready | 1 |
| SR6 ($D_6$) | —— | Reserved | - | - | —— |
| SR5 ($D_5$) | FMR07 | Erase status | Completed normally | Terminated by error | 0 |
| SR4 ($D_4$) | FMR06 | Program status | Completed normally | Terminated by error | 0 |
| SR3 ($D_3$) | —— | Reserved | - | - | —— |
| SR2 ($D_2$) | —— | Reserved | - | - | —— |
| SR1 ($D_1$) | —— | Reserved | - | - | —— |
| SR0 ($D_0$) | —— | Reserved | - | - | —— |

- $D_7$ to $D_0$: Indicates the data bus which is read out when executing the read status register command.
- The FMR07 bit (SR5) and FMR06 bit (SR4) are set to “0” by executing the clear status register command.
- When the FMR07 bit (SR5) or FMR06 bit (SR4) is 1, the program, and block erase command are not acknowledged.

### 19.8.4 Full Status Check

When an error occurs, the FMR06 to FMR07 bits in the FMR0 register are set to “1”, indicating occurrence of each specific error. Therefore, execution results can be verified by checking these status bits (full status check). Table 19.8.4.1 shows errors and the status of FMR0 register. Figure 19.8.4.1 shows a flow chart of the full status check and handling procedure for each error.

**Table 19.8.4.1. Errors and FMR0 Register Status**

| FMR00 register (SRD register) status | | Error | Error occurrence condition |
|---|---|---|---|
| FMR07 (SR5) | FMR06 (SR4) | | |
| 1 | 1 | Command sequence error | • When any commands are not written correctly<br>• A value other than ‘xxD0$_{16}$’ or ‘xxFF$_{16}$’ is written in the second bus cycle of the block erase command (Note 1)<br>• When the block erase command is executed on protected blocks<br>• When the program command is executed on protected blocks |
| 1 | 0 | Erase error | • When the block erase command is executed on unprotected blocks but the blocks are not automatically erased correctly |
| 0 | 1 | Program error | • When the program command is executed on unprotected blocks but the blocks are not automatically programmed correctly. |

Note 1: The flash memory enters read array mode by writing command code ‘xxFF$_{16}$’ in the second bus cycle of these commands. The command code written in the first bus cycle becomes invalid.

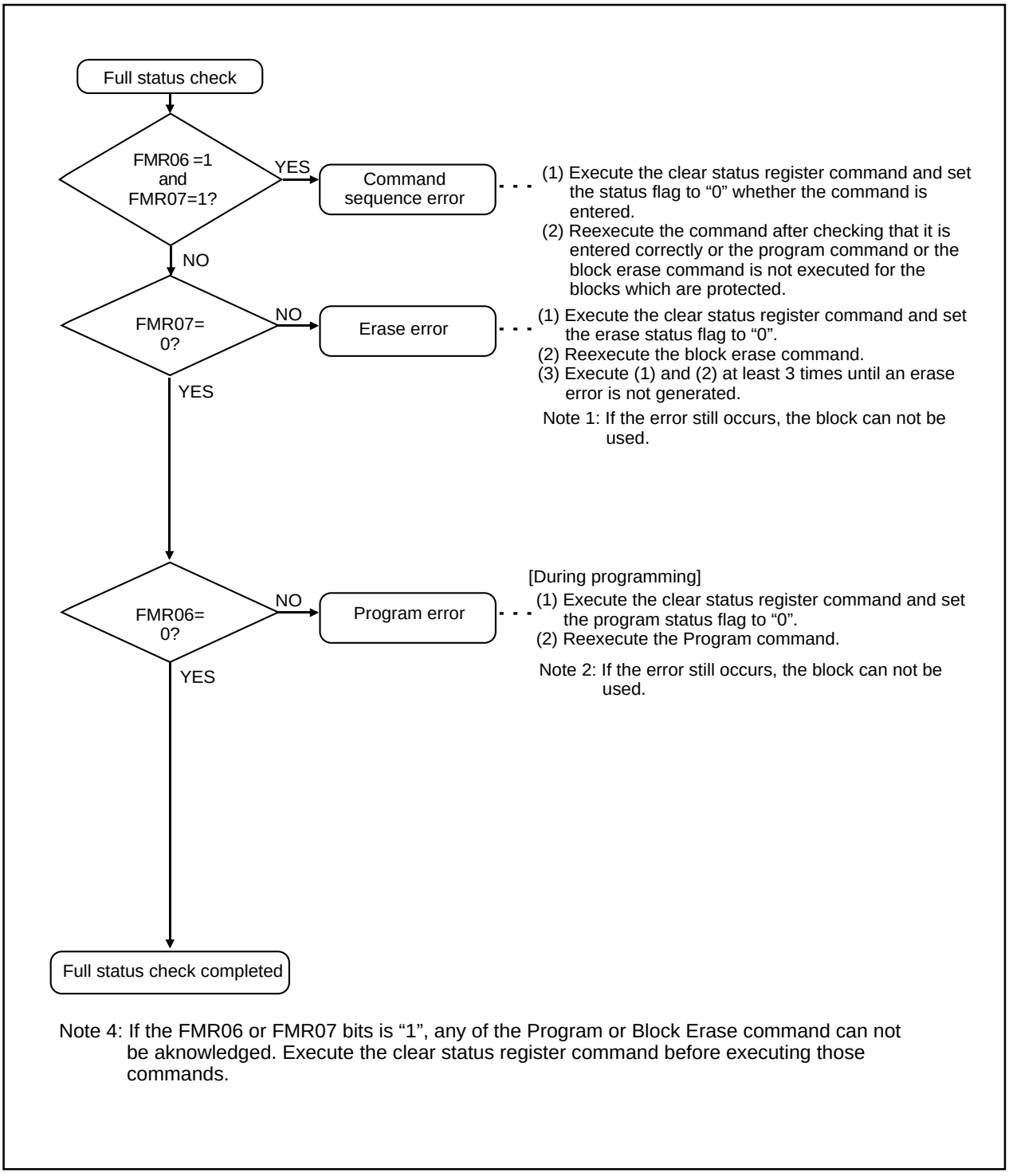



**Figure 19.8.4.1. Full Status Check and Handling Procedure for Each Error**

## 19.9 Standard Serial I/O Mode

In standard serial input/output mode, the user ROM area can be rewritten while the microcomputer is mounted on-board by using a serial programmer which is applicable for the M16C/28 group. For more information about serial programmers, contact the manufacturer of your serial programmer. For details on how to use the serial programmer, refer to the user's manual included with your serial programmer.
Table 19.9.1 shows pin functions (flash memory standard serial input/output mode). Figures 19.9.1 and 19.9.2 show pin connections for standard serial input/output mode.

### 19.9.1 ID Code Check Function

This function determines whether the ID codes sent from the serial programmer and those written in the flash memory match. (Refer to "19.3 Functions To Prevent Flash Memory from Rewriting".)

### Table 19.9.1. Pin Functions (Flash Memory Standard Serial I/O Mode)

| Pin | Name | I/O | Description |
|---|---|---|---|
| VCC,VSS | Power input | | Apply the voltage guaranteed for Program and Erase to Vcc pin and 0 V to Vss pin. |
| CNVSS | CNVSS | I | Connect to Vcc pin. |
| $\overline{\text{RESET}}$ | Reset input | I | Reset input pin. While $\overline{\text{RESET}}$ pin is "L" level, wait for td(ROC). |
| XIN | Clock input | I | Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin. |
| XOUT | Clock output | O | |
| AVCC, AVSS | Analog power supply input | | Connect AVss to Vss and AVcc to Vcc, respectively. |
| VREF | Reference voltage input | I | Enter the reference voltage for AD conversion. |
| P00 to P07 | Input port P0 | I | Input "H" or "L" level signal or open. |
| P10 to P17 | Input port P1 | I | Input "H" or "L" level signal or open. |
| P20 to P27 | Input port P2 | I | Input "H" or "L" level signal or open. |
| P30 to P37 | Input port P3 | I | Input "H" or "L" level signal or open. |
| P60 to P63 | Input port P6 | I | Input "H" or "L" level signal or open. |
| P64 | BUSY output | O | Standard serial I/O mode 1: BUSY signal output pin<br>Standard serial I/O mode 2: Monitor signal output pin for boot program operation check |
| P65 | SCLK input | I | Standard serial I/O mode 1: Serial clock input pin<br>Standard serial I/O mode 2: Input "L". |
| P66 | RxD input | I | Serial data input pin |
| P67 | TxD output | O | Serial data output pin (Note 1) |
| P70 to P77 | Input port P7 | I | Input "H" or "L" level signal or open. |
| P80 to P84, P87 | Input port P8 | I | Input "H" or "L" level signal or open. |
| P85 | $\overline{\text{RP}}$ input | I | Connect this pin to Vss while $\overline{\text{RESET}}$ pin is “L”. (Note 2) |
| P86 | $\overline{\text{CE}}$ input | I | Connect this pin to Vcc while $\overline{\text{RESET}}$ pin is “L”. (Note 2) |
| P90 to P93, P95 to P97 | Input port P9 | I | Input "H" or "L" level signal or open. |
| P100 to P107 | Input port P10 | I | Input "H" or "L" level signal or open. |

Note 1: When using standard serial input/output mode 1, to input “H” to the TxD pin is necessary while the $\overline{\text{RESET}}$ pin is “L”. Therefore, connect this pin to VCC via a resistor. Adjust the pull-up resistor value on a system not to affect a data transfer after reset, because this pin changes to a data-output pin

Note 2: Set either the $\overline{\text{RP}}$ pin or the $\overline{\text{CE}}$ pin.

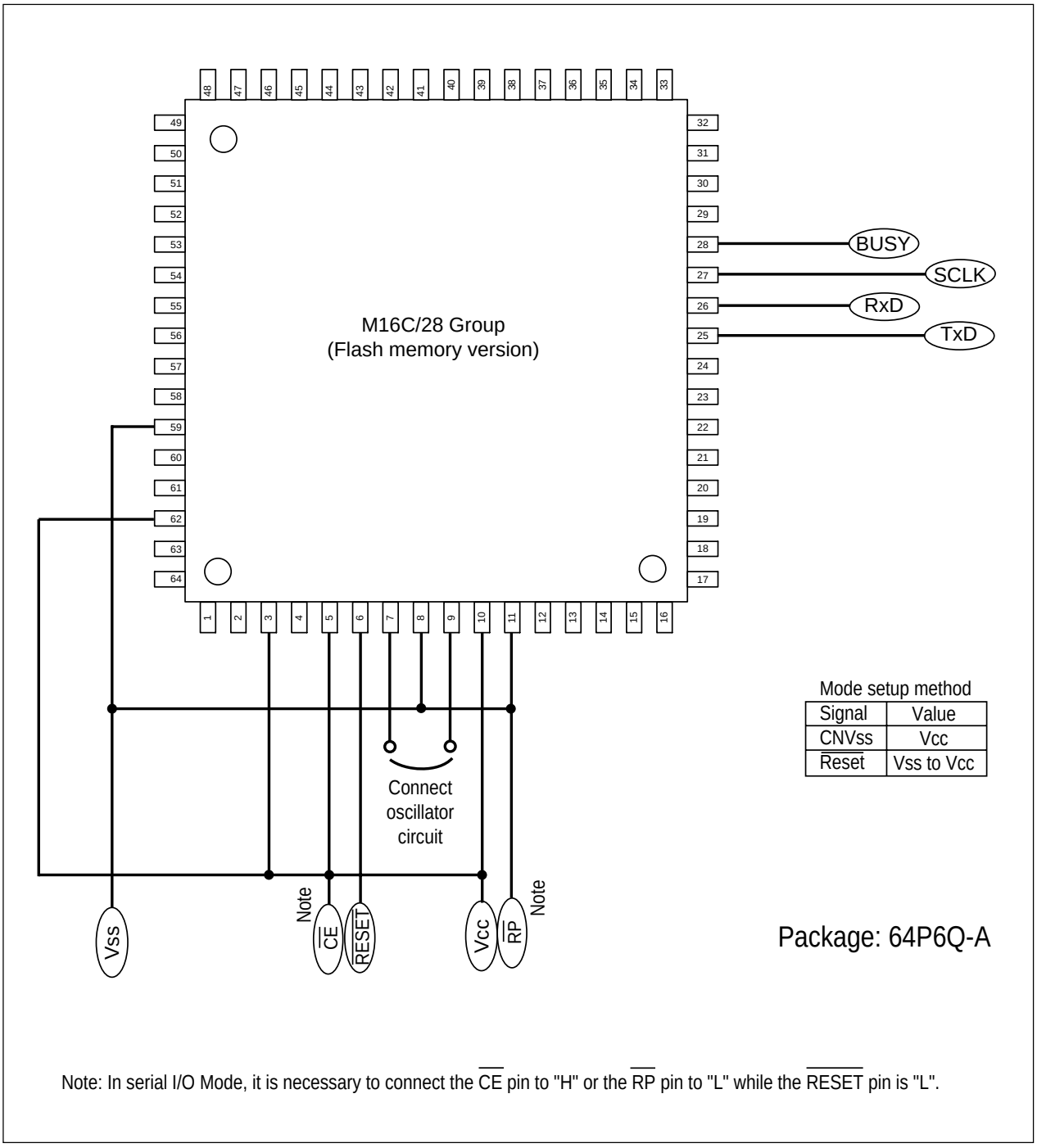



**Figure 19.9.1. Pin Connections for Serial I/O Mode (1)**

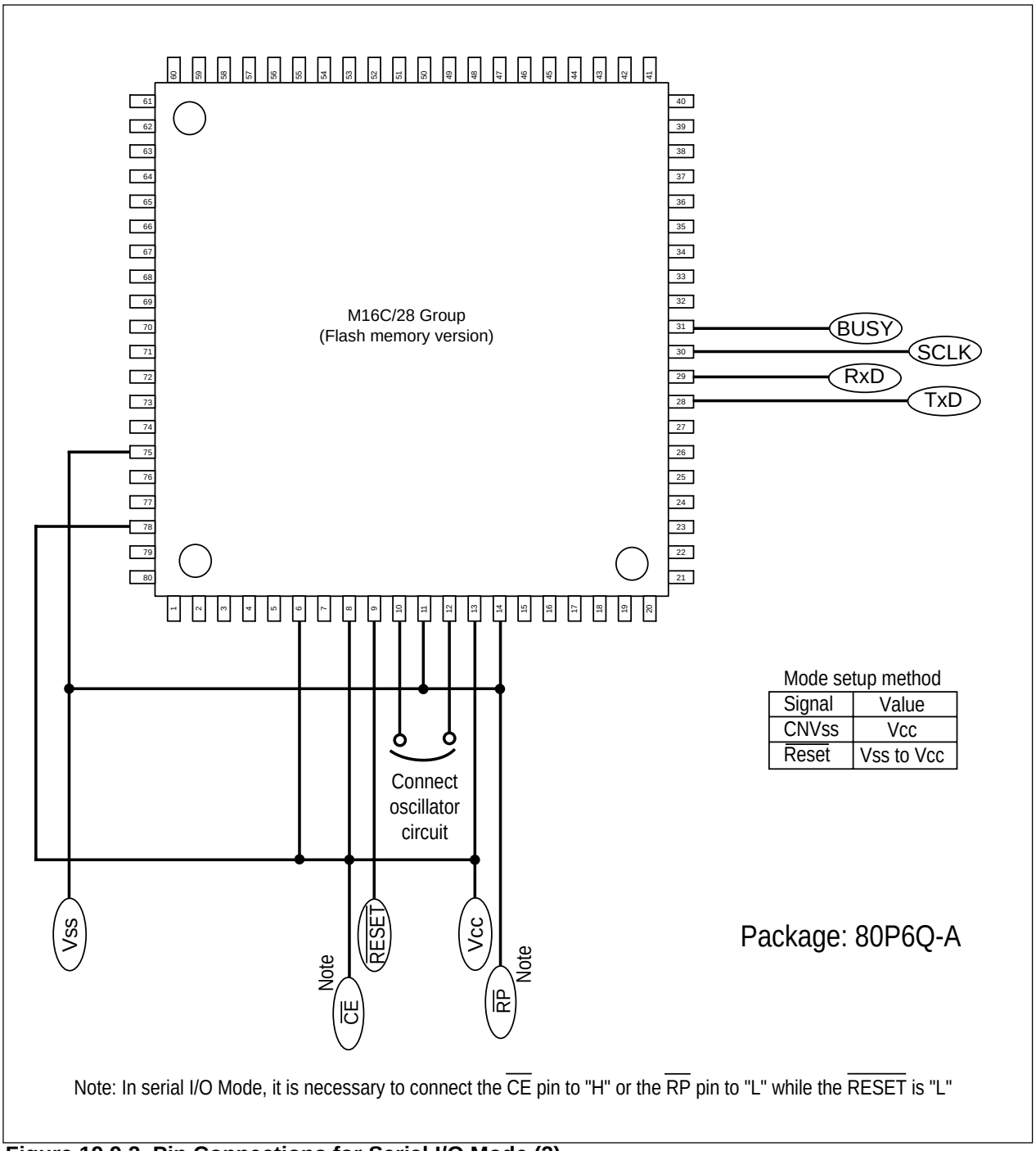
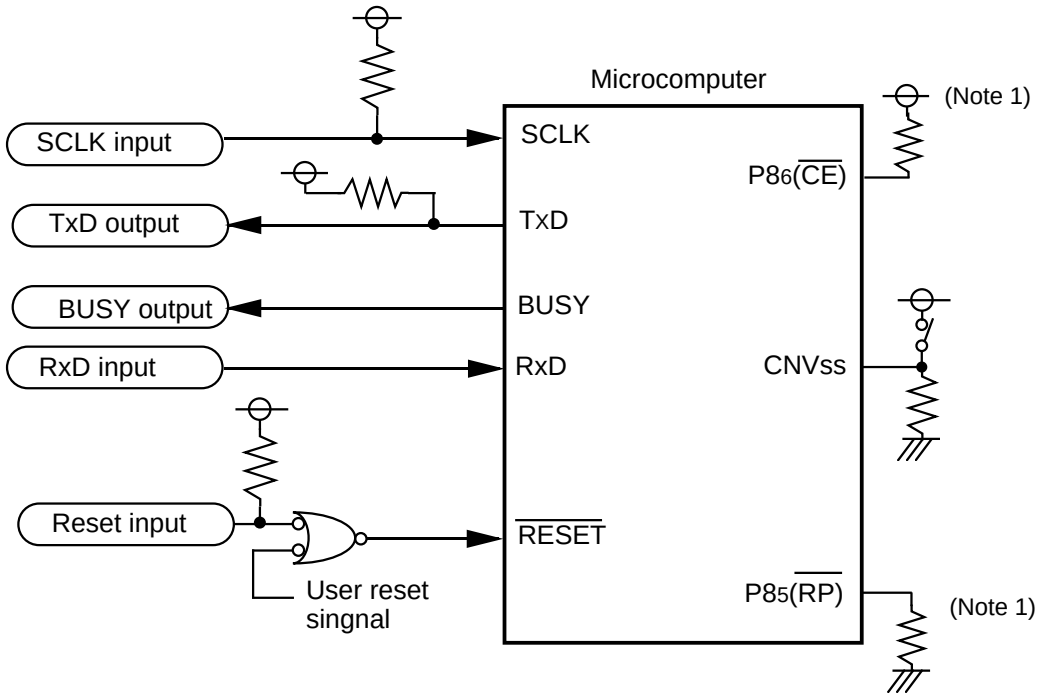



**Figure 19.9.2. Pin Connections for Serial I/O Mode (2)**

### 19.9.2 Example of Circuit Application in Standard Serial I/O Mode

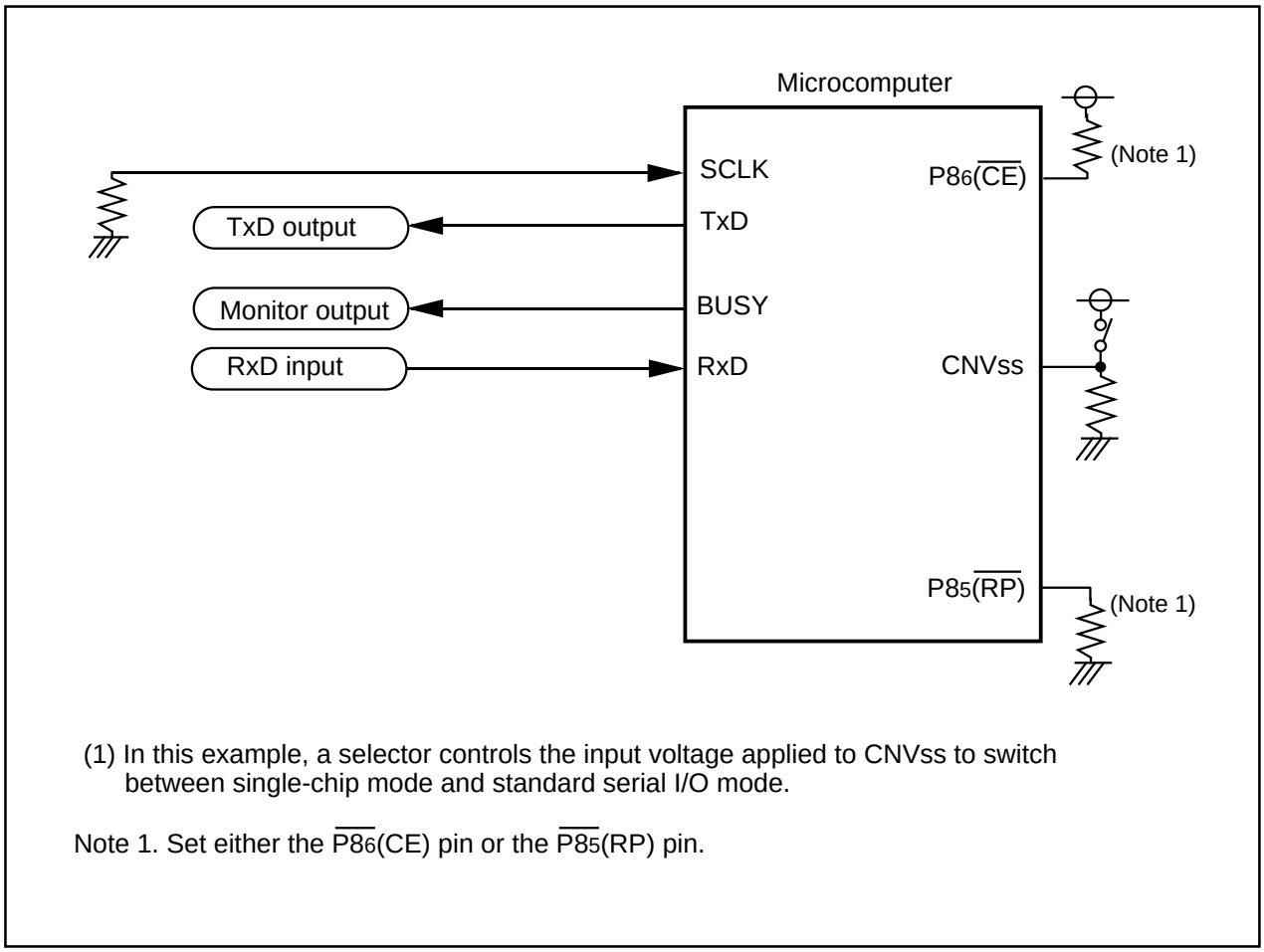
Figure 19.9.2.1 shows an example of a circuit application in standard serial I/O mode 1 and Figure 19.9.2.2 shows an example of a circuit application in standard serial I/O mode 2. Refer to the user's manual for a serial writer to handle pins controlled by the serial writer.




(1) Controlling pins and external circuits vary with the serial programmer. For more information, refer to the user's manual included with the serial programmer.
(2) In this example, a selector controls the input voltage applied to CNVss to switch between single-chip mode and standard serial I/O mode.
(3) In standard serial input/output mode 1, if the user reset signal becomes “L” while the microcomputer is communicating with the serial programmer, break the connection between the user reset signal and the RESET pin using a jumper switch.

Note 1. Set either the P86(CE) pin or the P85(RP) pin.

**Figure 19.9.2.1. Circuit Application in Standard Serial I/O Mode 1**




(1) In this example, a selector controls the input voltage applied to CNVss to switch between single-chip mode and standard serial I/O mode.

Note 1. Set either the $\overline{P86}$(CE) pin or the $\overline{P85}$(RP) pin.

**Figure 19.9.2.2. Circuit Application in Standard Serial I/o Mode 2**

## 19.10 Parallel I/O Mode

In parallel input/output mode, the user ROM can be rewritten using a parallel programmer which is applicable for the M16C/28 group. For more information about the parallel programmer, contact your parallel programmer manufacturer. For details on how to use the parallel programmer, refer to the user's manual of the parallel programmer.

### 19.10.1 ROM Code Protect Function

The ROM code protect function prevents the flash memory from being read or rewritten. (Refer to the description of the functions to inhibit rewriting flash memory version.)

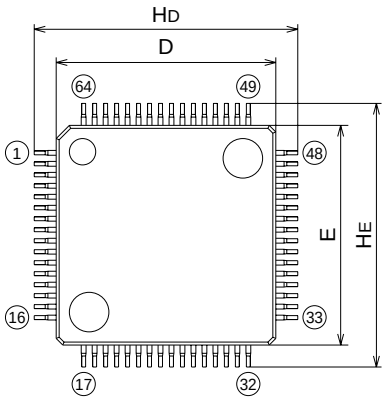
# 20. Package

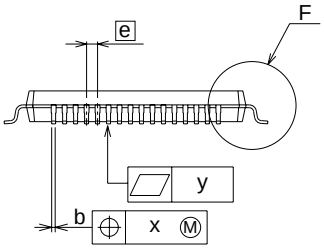
## 64P6Q-A

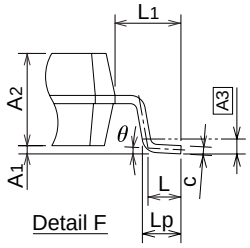
Recommended

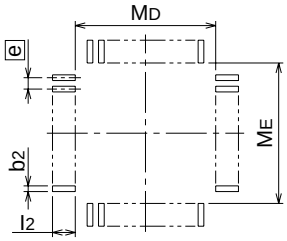
**Plastic 64pin 10×10mm body LQFP**

| EIAJ Package Code | JEDEC Code | Weight(g) | Lead Material |
|---|---|---|---|
| LQFP64-P-1010-0.50 | – | – | Cu Alloy |






Recommended Mount Pad

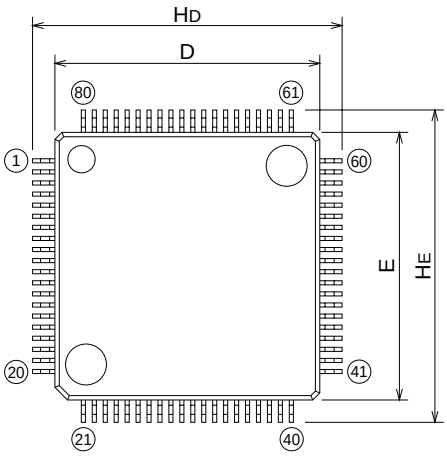
| Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| A | – | – | 1.7 |
| A1 | 0 | 0.1 | 0.2 |
| A2 | – | 1.4 | – |
| b | 0.13 | 0.18 | 0.28 |
| c | 0.105 | 0.125 | 0.175 |
| D | 9.9 | 10.0 | 10.1 |
| E | 9.9 | 10.0 | 10.1 |
| e | – | 0.5 | – |
| HD | 11.8 | 12.0 | 12.2 |
| HE | 11.8 | 12.0 | 12.2 |
| L | 0.3 | 0.5 | 0.7 |
| L1 | – | 1.0 | – |
| Lp | 0.45 | 0.6 | 0.75 |
| A3 | – | 0.25 | – |
| x | – | – | 0.08 |
| y | – | – | 0.1 |
| θ | 0° | – | 10° |
| b2 | – | 0.225 | – |
| l2 | 1.0 | – | – |
| MD | – | 10.4 | – |
| ME | – | 10.4 | – |

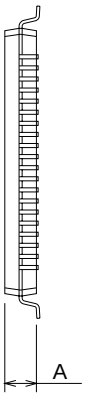
## 80P6Q-A

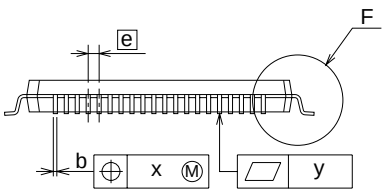
Recommended

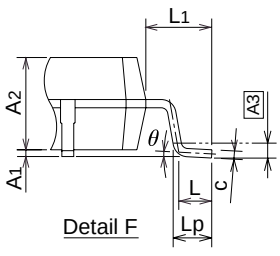
**Plastic 80pin 12×12mm body LQFP**

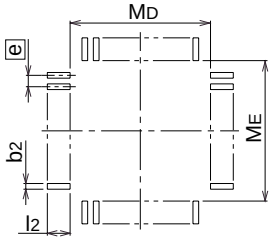
| EIAJ Package Code | JEDEC Code | Weight(g) | Lead Material |
|---|---|---|---|
| LQFP80-P-1212-0.5 | – | 0.47 | Cu Alloy |




Recommended Mount Pad

| Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| A | – | – | 1.7 |
| A1 | 0 | 0.1 | 0.2 |
| A2 | – | 1.4 | – |
| b | 0.13 | 0.18 | 0.28 |
| c | 0.105 | 0.125 | 0.175 |
| D | 11.9 | 12.0 | 12.1 |
| E | 11.9 | 12.0 | 12.1 |
| e | – | 0.5 | – |
| HD | 13.8 | 14.0 | 14.2 |
| HE | 13.8 | 14.0 | 14.2 |
| L | 0.3 | 0.5 | 0.7 |
| L1 | – | 1.0 | – |
| Lp | 0.45 | 0.6 | 0.75 |
| A3 | – | 0.25 | – |
| x | – | – | 0.08 |
| y | – | – | 0.1 |
| θ | 0° | – | 10° |
| b2 | – | 0.225 | – |
| l2 | 0.9 | – | – |
| MD | – | 12.4 | – |
| ME | – | 12.4 | – |

# Register Index


## A

AD0 to AD7 **208**
ADCON0 **206**
ADCON1 **206**
ADCON2 **206**
ADIC **60**
ADSTAT0 **208**
ADTRGCON **207**
AIER **72**

## B

BCNIC 60
BTIC 60

## C

CM0 **33**
CM1 **34**
CM2 **35**
CPSRF **89,102**

## D

D4INT **26**
DAR0 **79**
DAR1 **79**
DM0CON **78**
DM0IC **60**
DM0SL **78**
DM1CON **78**
DM1IC **60**
DM1SL **78**
DTT **113**

## F

FMR0 **351**
FMR1 **351**
FMR4 **352**

## G

G1BCR0 **124**
G1BCR1 **126**
G1BT **124**
G1BTRR **127**
G1DV **125**
G1FE **131**
G1FS **131**
G1IE0 **133**
G1IE1 **133**
G1IR **132**
G1PO0 to G1PO7 **130**
G1POCR0 to G1POCR7 **129**
G1TM0 to G1TM7 **129**
G1TMCR0 to G1TMCR7 **128**
G1TPR6 **128**
G1TPR7 **128**

## I

I2CIC **60**
ICOC0IC **60**
ICOC1IC **60**
ICTB2 **113**
IDB0 **113**
IDB1 **113**
IFSR **61, 69**
IFSR2A **61**
INT0IC to INT5IC **60**
INVC0 **111**
INVC1 **112**

## K

KUPIC **60**

## M

## N

NDDR **282**

## O

ONSF **89**

## P

P0 to P3,P6 to P10 **279**
P17DDR **282**
PACR **281**
PCLKR **36**
PCR **281**
PD0 to PD3,PD6 to PD10 **278**
PDRF **121**
PLC0 **37**
PM0 **30**
PM1 **30**
PM2 **36**

PRCR **53**
PUR0 to PUR2 **280**

## R

RMAD0 **72**
RMAD1 **72**
ROCR **34**

## S

S00 **241**
S0D0 **240**
S0TIC **60**
S0RIC **60**
S10 **243**
S1D0 **242**
S1TIC **60**
S1RIC **60**
S20 **241**
S2D0 **246**
S2TIC **60**
S2RIC **60**
S3BRG **200**
S3C **200**
S3D0 **244**
S3IC **60**
S3TRR **200**
S4BRG **200**
S4C **200**
S4D0 **245**
S4IC **60**
S4TRR **200**
SAR0 **79**
SAR1 **79**
SCLDAIC **60**

## T

TA0 **88**
TA0IC **60**
TA0MR **87**
TA1 **88,114**
TA11 **114**
TA1IC **60**
TA1MR **87,117**
TA2 **88,114**
TA21 **114**
TA2IC **60**
TA2MR **87,117**
TA3 **88**
TA3IC **60**
TA3MR **87**
TA4 **88,114**
TA41 **114**
TA4IC **60**
TA4MR **87,117**
TABSR **88,102,116**
TB0 **102**
TB0IC **60**
TB0MR **101**
TB1 **102**
TB1IC **60**
TB1MR **101**
TB2 **102,116**
TB2IC **60**
TB2MR **101,117**
TB2SC **108,115**
TCR0 **79**
TCR1 **79**
TRGSR **89,116**

## U

U0BRG **157**
U0C0 **159**
U0C1 **160**
U0MR **158**
U0RB **157**
U0TB **157**
U1BRG **157**
U1C0 **159**
U1C1 **160**
U1MR **158**
U1RB **157**
U1TB **157**
U2BRG **157**
U2C0 **159**
U2C1 **160**
U2MR **158**
U2RB **157**
U2SMR **161**

U2SMR2 **161**
U2SMR3 **162**
U2SMR4 **162**
U2TB **157**
UCON **159**
UDF **88**

## V

VCR1 **26**
VCR2 **26**

## W

WDC **25,74**
WDTS **74**

**RENESAS 16-BIT CMOS SINGLE-CHIP MICROCOMPUTER**
**HARDWARE MANUAL**
**M16C/28 Group Rev.0.60**

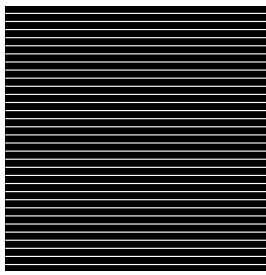
---

Editioned by
Committee of editing of RENESAS Semiconductor Hardware Manual

---

This book, or parts thereof, may not be reproduced in any form without permission of Renesas Technology Corporation.
Copyright © 2003. Renesas Technology Corporation, All rights reserved.

# M16C/28 Group
# Hardware Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan

REJ09B0170-0060Z

16

# M16C/28 Group

## Usage Notes Reference Book

RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER

M16C FAMILY / M16C/Tiny SERIES

For the most current **Usage Notes Reference Book**, please visit our website.

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

Rev. 0.60
Revision date: February. 01. 2004

Renesas Technology
www.renesas.com

Usage Notes Reference Book

## Keep safety first in your circuit designs!

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

## Notes regarding these materials

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

# Preface

The “Usage Notes Reference Book” is a compilation of usage notes from the Hardware Manual as well as technical news related to this product.

# Table of Contents


1. Usage Precaution ..... 1

1.1 Precautions for SFR ..... 1

1.1.1 Precaution for 80 pin version ..... 1

1.1.2 Precaution for 64 pin version ..... 1

1.2 Precautions for PLL Frequency Synthesizer ..... 2

1.3 Precautions for Power Control ..... 3

1.4 Precautions for Protect ..... 4

1.5 Precautions for Interrupts ..... 5

1.5.1 Reading address $00000_{16}$ ..... 5

1.5.2 Setting the SP ..... 5

1.5.3 The $\overline{\text{NMI}}$ Interrupt ..... 5

1.5.4 Changing the Interrupt Generate Factor ..... 6

1.5.5 $\overline{\text{INT}}$ Interrupt ..... 6

1.5.6 Rewrite the Interrupt Control Register ..... 7

1.5.7 Watchdog Timer Interrupt ..... 8

1.6 Precautions for DMAC ..... 9

1.6.1 Write to DMAE Bit in DMiCON Register ..... 9

1.7 Precautions for Timers ..... 10

1.7.1 Timer A ..... 10

1.7.1.1 Timer A (Timer Mode) ..... 10

1.7.1.2 Timer A (Event Counter Mode) ..... 11

1.7.1.3 Timer A (One-shot Timer Mode) ..... 12

1.7.1.4 Timer A (Pulse Width Modulation Mode) ..... 13

1.7.2 Timer B ..... 14

1.7.2.1 Timer B (Timer Mode) ..... 14

1.7.2.2 Timer B (Event Counter Mode) ..... 15

1.7.2.3 Timer B (Pulse Period/pulse Width Measurement Mode) ..... 16

1.7.3 Timer S ..... 17

1.7.3.1 Rewrite the G1IR register ..... 17

1.8 Precautions for Serial I/O (Clock-synchronous Serial I/O) ..... 18
1.8.1 Transmission/reception ..... 18
1.8.2 Transmission ..... 19
1.8.3 Reception ..... 20
1.9 Precautions for Serial I/O (UART Mode) ..... 21
1.9.1 Special Mode 2 ..... 21
1.9.2 Special Mode 4 (SIM Mode) ..... 21
1.10 Precautions for A-D Converter ..... 22
1.11 Precautions for Programmable I/O Ports ..... 24
1.12 Electric Characteristic Differences Between Mask ROM and Flash Memory Version Microcomputers ..... 25
1.13 Precautions for Flash Memory Version ..... 26
1.13.1 Precautions for Functions to Inhibit Rewriting Flash Memory Rewrite ..... 26
1.13.2 Precautions for Stop mode ..... 26
1.13.3 Precautions for Wait mode ..... 26
1.13.4 Precautions for Low power dissipation mode, ring oscillator low power dissipation mode ..... 26
1.13.5 Writing command and data ..... 26
1.13.6 Precautions for Program Command ..... 26
1.13.7 Operation speed ..... 27
1.13.8 Instructions inhibited against use ..... 27
1.13.9 Interrupts ..... 27
1.13.10 How to access ..... 27
1.13.11 Writing in the user ROM area ..... 28
1.13.12 DMA transfer ..... 28
1.13.13 Regarding Programming/Erasure Times and Execution Time ..... 28

# 1. Usage Precaution

## 1.1 Precautions for SFR

### 1.1.1 Precaution for 80 pin version

Set the IFSR20 bit in the IFSR2A register to "1" after reset and set the PACR2 to PACR0 bits in the PACR register to "$011_2$".

### 1.1.2 Precaution for 64 pin version

Set the IFSR20 bit in the IFSR2A register to "1" after reset and set the PACR2 to PACR0 bits in the PACR register to "$010_2$".

## 1.2 Precautions for PLL Frequency Synthesizer

Make the supply voltage stable to use the PLL frequency synthesizer.

For ripple with the supply voltage 5V, keep below 10kHz as frequency, below 0.5V (peak to peak) as voltage fluctuation band and below 1V/mS as voltage fluctuation rate.

For ripple with the supply voltage 3V, keep below 10kHz as frequency, below 0.3V (peak to peak) as voltage fluctuation band and below 0.6V/mS as voltage fluctuation rate.

## 1.3 Precautions for Power Control

1. When exiting stop mode by hardware reset, the device will startup using the ring oscillator.

2. Insert more than four NOP instructions after an WAIT instruction or a instruction to set the CM10 bit of CM1 register to “1”. When shifting to wait mode or stop mode, an instruction queue reads ahead to the next instruction to halt a program by an WAIT instruction and an instruction to set the CM10 bit to “1” (all clocks stopped). The next instruction may be executed before entering wait mode or stop mode, depending on a combination of instruction and an execution timing.

3. Wait until the $t_{d(M-L)}$ elapses or main clock oscillation stabilization time, whichever is longer, before switching the clock source for CPU clock to the main clock.
Similarly, wait until the sub clock oscillates stably before switching the clock source for CPU clock to the sub clock.

4. Suggestions to reduce power consumption

**(a) Ports**

The processor retains the state of each I/O port even when it goes to wait mode or to stop mode. A current flows in active I/O ports. A pass current flows in input ports that high-impedance state. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

**(b) A-D converter**

When A-D conversion is not performed, set the VCUT bit of ADiCON1 register to “0” (no $V_{REF}$ connection). When A-D conversion is performed, start the A-D conversion at least 1 µs or longer after setting the VCUT bit to “1” ($V_{REF}$ connection).

**(c) Stopping peripheral functions**

Use the CM0 register CM02 bit to stop the unnecessary peripheral functions during wait mode.
However, because the peripheral function clock ($f_{C32}$) generated from the sub-clock does not stop, this measure is not conducive to reducing the power consumption of the chip. If low speed mode or low power dissipation mode is to be changed to wait mode, set the CM02 bit to “0” (do not peripheral function clock stopped when in wait mode), before changing wait mode.

**(d) Switching the oscillation-driving capacity**

Set the driving capacity to “LOW” when oscillation is stable.

## 1.4 Precautions for Protect

Set the PRC2 bit to “1” (write enabled) and then write to any address, and the PRC2 bit will be cleared to “0” (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to “1”. Make sure no interrupts or DMA transfers will occur between the instruction in which the PRC2 bit is set to “1” and the next instruction.

## 1.5 Precautions for Interrupts

### 1.5.1 Reading address 00000$_{16}$

Do not read the address 00000$_{16}$ in a program. When a maskable interrupt request is accepted, the CPU reads interrupt information (interrupt number and interrupt request priority level) from the address 00000$_{16}$ during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to “0”. If the address 00000$_{16}$ is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to “0”. This causes a problem that the interrupt is canceled, or an unexpected interrupt request is generated.

### 1.5.2 Setting the SP

Set any value in the SP(USP, ISP) before accepting an interrupt. The SP(USP, ISP) is cleared to ‘0000$_{16}$’ after reset. Therefore, if an interrupt is accepted before setting any value in the SP(USP, ISP), the program may go out of control.
Especially when using $\overline{\text{NMI}}$ interrupt, set a value in the ISP at the beginning of the program. For the first and only the first instruction after reset, all interrupts including $\overline{\text{NMI}}$ interrupt are disabled.

### 1.5.3 The $\overline{\text{NMI}}$ Interrupt

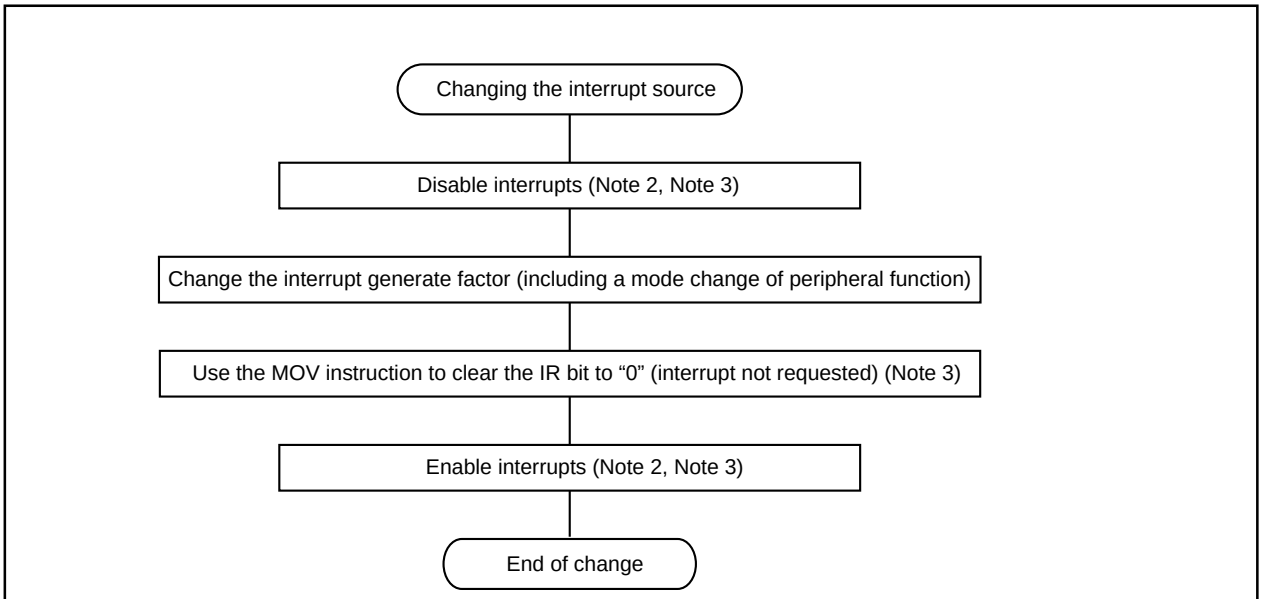
1. The $\overline{\text{NMI}}$ interrupt is invalid after reset. The $\overline{\text{NMI}}$ interrupt becomes effective by setting to “1” the PM24 bit of the PM2 register. Once enabled, it stays enabled until a reset is applid.
2. The input level of the $\overline{\text{NMI}}$ pin can be read by accessing the P8 register’s P8_5 bit. Note that the P8_5 bit can only be read when determining the pin level in $\overline{\text{NMI}}$ interrupt routine.
3. When selecting $\overline{\text{NMI}}$ function, stop mode cannot be entered into while input on the $\overline{\text{NMI}}$ pin is low. This is because while input on the $\overline{\text{NMI}}$ pin is low the CM1 register’s CM10 bit is fixed to “0”.
4. When selecting $\overline{\text{NMI}}$ function, do not go to wait mode while input on the $\overline{\text{NMI}}$ pin is low. This is because when input on the $\overline{\text{NMI}}$ pin goes low, the CPU stops but CPU clock remains active; therefore, the current consumption in the chip does not drop. In this case, normal condition is restored by an interrupt generated thereafter.
5. When selecting $\overline{\text{NMI}}$ function, the low and high level durations of the input signal to the $\overline{\text{NMI}}$ pin must each be 2 CPU clock cycles + 300 ns or more.

### 1.5.4 Changing the Interrupt Generate Factor

If the interrupt generate factor is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to “1” (interrupt requested). If you changed the interrupt generate factor for an interrupt that needs to be used, be sure to clear the IR bit for that interrupt to “0” (interrupt not requested).

“Changing the interrupt generate factor” referred to here means any act of changing the source, polarity or timing of the interrupt assigned to each software interrupt number. Therefore, if a mode change of any peripheral function involves changing the generate factor, polarity or timing of an interrupt, be sure to clear the IR bit for that interrupt to “0” (interrupt not requested) after making such changes. Refer to the description of each peripheral function for details about the interrupts from peripheral functions.

Figure 1.5.1 shows the procedure for changing the interrupt generate factor.




IR bit: A bit in the interrupt control register for the interrupt whose interrupt generate factor is to be changed

Note 1: The above settings must be executed individually. Do not execute two or more settings simultaneously (using one instruction).
Note 2: Use the I flag for the $\overline{\text{INT}}$i interrupt (i = 0 to 5).
For the interrupts from peripheral functions other than the $\overline{\text{INT}}$i interrupt, turn off the peripheral function that is the source of the interrupt in order not to generate an interrupt request before changing the interrupt generate factor. In this case, if the maskable interrupts can all be disabled without causing a problem, use the I flag. Otherwise, use the corresponding ILVL2 to ILVL0 bit for the interrupt whose interrupt generate factor is to be changed.
Note 3: Refer to Section 1.1.6, “Rewrite the Interrupt Control Register” for details about the instructions to use and the notes to be taken for instruction execution.

**Figure 1.5.1. Procedure for Changing the Interrupt Generate Factor**

### 1.5.5 $\overline{\text{INT}}$ Interrupt

1. Either an “L” level of at least $t_{W(INH)}$ or an “H” level of at least $t_{W(INL)}$ width is necessary for the signal input to pins $INT_0$ through $INT_5$ regardless of the CPU operation clock.
2. If the POL bit in the INT0IC to INT5IC registers or the IFSR7 to IFSR0 bits in the IFSR register are changed, the IR bit may inadvertently set to 1 (interrupt requested). Be sure to clear the IR bit to 0 (interrupt not requested) after changing any of those register bits.

### 1.5.6 Rewrite the Interrupt Control Register

(1) The interrupt control register for any interrupt should be modified in places where no requests for that interrupt may occur. Otherwise, disable the interrupt before rewriting the interrupt control register.

(2) To rewrite the interrupt control register for any interrupt after disabling that interrupt, be careful with the instruction to be used.

**Changing any bit other than the IR bit**

If while executing an instruction, a request for an interrupt controlled by the register being modified occurs, the IR bit in the register may not be set to “1” (interrupt requested), with the result that the interrupt request is ignored. If such a situation presents a problem, use the instructions shown below to modify the register.

Usable instructions: AND, OR, BCLR, BSET

**Changing the IR bit**

Depending on the instruction used, the IR bit may not always be cleared to “0” (interrupt not requested). Therefore, be sure to use the MOV instruction to clear the IR bit.

(3) When using the I flag to disable an interrupt, refer to the sample program fragments shown below as you set the I flag. (Refer to (2) for details about rewrite the interrupt control registers in the sample program fragments.)

Examples 1 through 3 show how to prevent the I flag from being set to “1” (interrupts enabled) before the interrupt control register is rewrited, owing to the effects of the internal bus and the instruction queue buffer.

**Example 1:Using the NOP instruction to keep the program waiting until the interrupt control register is modified**

```
INT_SWITCH1:
        FCLR    I               ; Disable interrupts.
        AND.B   #00h, 0055h     ; Set the TA0IC register to “00₁₆”.
        NOP                     ;
        NOP
        FSET    I               ; Enable interrupts.
```

**Example 2:Using the dummy read to keep the FSET instruction waiting**

```
INT_SWITCH2:
        FCLR    I               ; Disable interrupts.
        AND.B   #00h, 0055h     ; Set the TA0IC register to “00₁₆”.
        MOV.W   MEM, R0         ; Dummy read.
        FSET    I               ; Enable interrupts.
```

**Example 3:Using the POPC instruction to changing the I flag**

```
INT_SWITCH3:
        PUSHC   FLG
        FCLR    I               ; Disable interrupts.
        AND.B   #00h, 0055h     ; Set the TA0IC register to “00₁₆”.
        POPC    FLG             ; Enable interrupts.
```

### 1.5.7 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt occurs.

## 1.6 Precautions for DMAC

### 1.6.1 Write to DMAE Bit in DMiCON Register

When both of the conditions below are met, follow the steps below.

Conditions

- The DMAE bit is set to “1” again while it remains set (DMAi is in an active state).
- A DMA request may occur simultaneously when the DMAE bit is being written.

Step 1: Write “1” to the DMAE bit and DMAS bit in DMiCON register simultaneously(*1).
Step 2: Make sure that the DMAi is in an initial state(*2) in a program.
If the DMAi is not in an initial state, the above steps should be repeated.

Notes:

*1. The DMAS bit remains unchanged even if “1” is written. However, if “0” is written to this bit, it is set to “0” (DMA not requested). In order to prevent the DMAS bit from being modified to “0”, “1” should be written to the DMAS bit when “1” is written to the DMAE bit. In this way the state of the DMAS bit immediately before being written can be maintained.
Similarly, when writing to the DMAE bit with a read-modify-write instruction, “1” should be written to the DMAS bit in order to maintain a DMA request which is generated during execution.

*2. Read the TCRi register to verify whether the DMAi is in an initial state. If the read value is equal to a value which was written to the TCRi register before DMA transfer start, the DMAi is in an initial state. (If a DMA request occurs after writing to the DMAE bit, the value written to the TCRi register is “1”.) If the read value is a value in the middle of transfer, the DMAi is not in an initial state.

## 1.7 Precautions for Timers

### 1.7.1 Timer A

#### 1.7.1.1 Timer A (Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register and the TAi register before setting the TAiS bit in the TABSR register to “1” (count starts).
   Always make sure the TAiMR register is modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

2. While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the counter is read at the same time it is reloaded, the value “FFFF$_{16}$” is read. Also, if the counter is read before it starts counting after a value is set in the TAi register while not counting, the set value is read.

3. If a low-level signal is applied to the $\overline{SD}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1$_{OUT}$, TA2$_{OUT}$ and TA4$_{OUT}$ pins go to a high-impedance state.

#### 1.7.1.2 Timer A (Event Counter Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, the UDF register, the ONSF register TAZIE, TA0TGL and TA0TGH bits and the TRGSR register before setting the TAiS bit in the TABSR register to “1” (count starts).
   Always make sure the TAiMR register, the UDF register, the ONSF register TAZIE, TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

2. While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, “FFFF$_{16}$” can be read in underflow, while reloading, and “0000$_{16}$” in overflow. When setting TAi register to a value during a counter stop, the setting value can be read before a counter starts counting. Also, if the counter is read before it starts counting after a value is set in the TAi register while not counting, the set value is read.

3. If a low-level signal is applied to the $\overline{SD}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1$_{OUT}$, TA2$_{OUT}$ and TA4$_{OUT}$ pins go to a high-impedance state.

#### 1.7.1.3 Timer A (One-shot Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, the ONSF register TA0TGL and TA0TGH bits and the TRGSR register before setting the TAiS bit in the TABSR register to “1” (count starts).
   Always make sure the TAiMR register, the ONSF register TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

2. When setting TAiS bit to “0” (count stop), the followings occur:
   • A counter stops counting and a content of reload register is reloaded.
   • TAi$_{OUT}$ pin outputs “L”.
   • After one cycle of the CPU clock, the IR bit of TAiIC register is set to “1” (interrupt request).

3. Output in one-shot timer mode synchronizes with a count source internally generated. When an external trigger has been selected, one-cycle delay of a count source as maximum occurs between a trigger input to TAi$_{IN}$ pin and output in one-shot timer mode.

4. The IR bit is set to “1” when timer operation mode is set with any of the following procedures:
   • Select one-shot timer mode after reset.
   • Change an operation mode from timer mode to one-shot timer mode.
   • Change an operation mode from event counter mode to one-shot timer mode.
   To use the timer Ai interrupt (the IR bit), set the IR bit to “0” after the changes listed above have been made.

5. When a trigger occurs, while counting, a counter reloads the reload register to continue counting after generating a re-trigger and counting down once. To generate a trigger while counting, generate a second trigger between occurring the previous trigger and operating longer than one cycle of a timer count source.

6. If a low-level signal is applied to the $\overline{SD}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1$_{OUT}$, TA2$_{OUT}$ and TA4$_{OUT}$ pins go to a high-impedance state.

#### 1.7.1.4 Timer A (Pulse Width Modulation Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, the ONSF register TA0TGL and TA0TGH bits and the TRGSR register before setting the TAiS bit in the TABSR register to “1” (count starts).
   Always make sure the TAiMR register, the ONSF register TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

2. The IR bit is set to “1” when setting a timer operation mode with any of the following procedures:
   • Select the PWM mode after reset.
   • Change an operation mode from timer mode to PWM mode.
   • Change an operation mode from event counter mode to PWM mode.
   To use the timer Ai interrupt (interrupt request bit), set the IR bit to “0” by program after the above listed changes have been made.

3. When setting TAiS register to “0” (count stop) during PWM pulse output, the following action occurs:
   • Stop counting.
   • When TAiOUT pin is output “H”, output level is set to “L” and the IR bit is set to “1”.
   • When TAiOUT pin is output “L”, both output level and the IR bit remains unchanged.

4. If a low-level signal is applied to the $\overline{SD}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

## 1.7.2 Timer B

### 1.7.2.1 Timer B (Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 2) register and TBi register before setting the TBiS bit in the TABSR or the TBSR register to “1” (count starts).
   Always make sure the TBiMR register is modified while the TBiS bit remains “0” (count stops) regardless whether after reset or not.

2. A value of a counter, while counting, can be read in TBi register at any time. “$FFFF_{16}$” is read while reloading. Setting value is read between setting values in TBi register at count stop and starting a counter.

#### 1.7.2.2 Timer B (Event Counter Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 2) register and TBi register before setting the TBiS bit in the TABSR or the TBSR register to “1” (count starts).
   Always make sure the TBiMR register is modified while the TBiS bit remains “0” (count stops) regardless whether after reset or not.

2. The counter value can be read out on-the-fly at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always “FFFF16.” If the TBi register is read after setting a value in it while not counting but before the counter starts counting, the read value is the one that has been set in the register.

#### 1.7.2.3 Timer B (Pulse Period/pulse Width Measurement Mode)

1. The timer remains idle after reset. Set the mode, count source, etc. using the TBiMR (i = 0 to 2) register before setting the TBiS bit in the TABSR or the TBSR register to “1” (count starts). Always make sure the TBiMR register is modified while the TBiS bit remains “0” (count stops) regardless whether after reset or not. To clear the MR3 bit to “0” by writing to the TBiMR register while the TBiS bit = “1” (count starts), be sure to write the same value as previously written to the TM0D0, TM0D1, MR0, MR1, TCK0 and TCK1 bits and a 0 to the MR2 bit.

2. The IR bit of TBiIC register (i=0 to 2) goes to “1” (interrupt request), when an effective edge of a measurement pulse is input or timer Bi is overflowed. The factor of interrupt request can be determined by use of the MR3 bit of TBiMR register within the interrupt routine.

3. If the source of interrupt cannot be identified by the MR3 bit such as when the measurement pulse input and a timer overflow occur at the same time, use another timer to count the number of times timer B has overflowed.

4. To set the MR3 bit to “0” (no overflow), set TBiMR register with setting the TBiS bit to “1” and counting the next count source after setting the MR3 bit to “1” (overflow).

5. Use the IR bit of TBiIC register to detect only overflows. Use the MR3 bit only to determine the interrupt factor within the interrupt routine.

6. When a count is started and the first effective edge is input, an indeterminate value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.

7. A value of the counter is indeterminate at the beginning of a count. MR3 may be set to “1” and timer Bi interrupt request may be generated between a count start and an effective edge input.

8. For pulse width measurement, pulse widths are successively measured. Use program to check whether the measurement result is an “H” level width or an “L” level width.

## 1.7.3 Timer S

### 1.7.3.1 Rewrite the G1IR register

When write "0" (without interrupt request) to each bit in the G1IR register, use the following instructions.

Usable instructions: AND, BCLR

## 1.8 Precautions for Serial I/O (Clock-synchronous Serial I/O)

### 1.8.1 Transmission/reception

1. With an external clock selected, and choosing the $\overline{\text{RTS}}$ function, the output level of the $\overline{\text{RTSi}}$ pin goes to “L” when the data-receivable status becomes ready, which informs the transmission side that the reception has become ready. The output level of the $\overline{\text{RTSi}}$ pin goes to “H” when reception starts. So if the $\overline{\text{RTSi}}$ pin is connected to the $\overline{\text{CTSi}}$ pin on the transmission side, the circuit can transmission and reception data with consistent timing. With the internal clock, the $\overline{\text{RTS}}$ function has no effect.

2. If a low-level signal is applied to the $\overline{\text{SD}}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{\text{SD}}$ pin enabled), the $\text{RTS}_2$ and $\text{CLK}_2$ pins go to a high-impedance state.

### 1.8.2 Transmission

When an external clock is selected, the conditions must be met while if the UiC0 register's CKPOL bit = "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the UiC0 register's CKPOL bit = "1" (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.

- The TE bit of UiC1 register= "1" (transmission enabled)
- The TI bit of UiC1 register = "0" (data present in UiTB register)
- If $\overline{\text{CTS}}$ function is selected, input on the $\overline{\text{CTS}}$i pin = "L"

### 1.8.3 Reception

1. In operating the clock-synchronous serial I/O, operating a transmitter generates a shift clock. Fix settings for transmission even when using the device only for reception. Dummy data is output to the outside from the TxDi pin when receiving data.

2. When an internal clock is selected, set the UiC1 register (i = 0 to 2)'s TE bit to 1 (transmission enabled) and write dummy data to the UiTB register, and the shift clock will thereby be generated. When an external clock is selected, set the UiC1 register (i = 0 to 2)'s TE bit to 1 and write dummy data to the UiTB register, and the shift clock will be generated when the external clock is fed to the CLKi input pin.

3. When successively receiving data, if all bits of the next receive data are prepared in the UARTi receive register while the UiC1 register (i = 0 to 2)'s RE bit = "1" (data present in the UiRB register), an overrun error occurs and the UiRB register OER bit is set to "1" (overrun error occurred). In this case, because the content of the UiRB register is indeterminate, a corrective measure must be taken by programs on the transmit and receive sides so that the valid data before the overrun error occurred will be retransmitted. Note that when an overrun error occurred, the SiRIC register IR bit does not change state.

4. To receive data in succession, set dummy data in the lower-order byte of the UiTB register every time reception is made.

5. When an external clock is selected, the conditions must be met while if the CKPOL bit = "0", the external clock is in the high state; if the CKPOL bit = "1", the external clock is in the low state.
   - The RE bit of UiC1 register= "1" (reception enabled)
   - The TE bit of UiC1 register= "1" (transmission enabled)
   - The TI bit of UiC1 register= "0" (data present in the UiTB register)

## 1.9 Precautions for Serial I/O (UART Mode)

### 1.9.1 Special Mode 2

If a low-level signal is applied to the $\overline{SD}$ pin when the TB2SC register IVPCR1 bit = 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the $RTS_2$ and $CLK_2$ pins go to a high-impedance state.

### 1.9.2 Special Mode 4 (SIM Mode)

A transmit interrupt request is generated by setting the U2C1 register U2IRS bit to “1” (transmission complete) and U2ERE bit to “1” (error signal output) after reset. Therefore, when using SIM mode, be sure to clear the IR bit to “0” (no interrupt request) after setting these bits.

## 1.10 Precautions for A-D Converter

1. Set ADCON0 (except bit 6), ADCON1, ADCON2 and ADTRGCON registers when A-D conversion is stopped (before a trigger occurs).

2. When the VCUT bit of ADCON1 register is changed from “0” (Vref not connected) to “1” (Vref connected), start A-D conversion after passing 1 µs or longer.

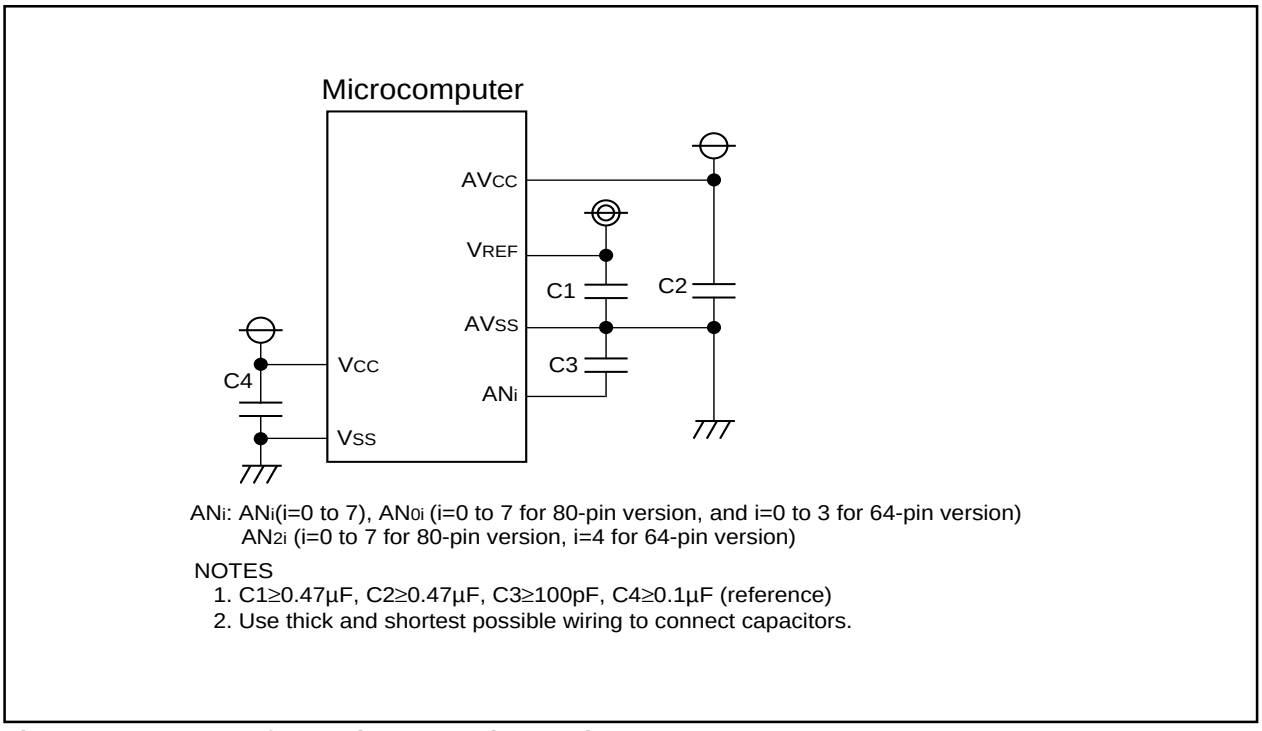
3. To prevent noise-induced device malfunction or latchup, as well as to reduce conversion errors, insert capacitors between the AVCC, VREF, and analog input pins (ANi, AN0i, AN2i(i=0 to 7)) each and the AVSS pin. Similarly, insert a capacitor between the VCC1 pin and the VSS pin. Figure 1.10.1 is an example connection of each pin.

4. Make sure the port direction bits for those pins that are used as analog inputs are set to “0” (input mode). Also, if the ADCON0 register’s TGR bit = 1 (external trigger), make sure the port direction bit for the $\overline{\text{ADTRG}}$ pin is set to “0” (input mode).

**5.** When using key input interrupts, do not use any of the four AN4 to AN7 pins as analog inputs. (A key input interrupt request is generated when the A-D input voltage goes low.)

6. The ϕAD frequency must be 10 MHz or less. Without sample-and-hold function, limit the ϕAD frequency to 250kHz or more. With the sample and hold function, limit the ϕAD frequency to 1MHz or more.

7. When changing an A-D operation mode, select analog input pin again in the CH2 to CH0 bits of ADCON0 register and the SCAN1 to SCAN0 bits of ADCON1 register.




ANi: ANi(i=0 to 7), AN0i (i=0 to 7 for 80-pin version, and i=0 to 3 for 64-pin version)
AN2i (i=0 to 7 for 80-pin version, i=4 for 64-pin version)

NOTES
1. C1≥0.47µF, C2≥0.47µF, C3≥100pF, C4≥0.1µF (reference)
2. Use thick and shortest possible wiring to connect capacitors.

**Figure 1.10.1. Use of capacitors to reduce noise**

8. If the CPU reads the ADi register (i = 0 to 7) at the same time the conversion result is stored in the ADi register after completion of A-D conversion, an incorrect value may be stored in the ADi register. This problem occurs when a divide-by-n clock derived from the main clock or a subclock is selected for CPU clock.
   - When operating in one-shot, single-sweep mode, simultaneous sample sweep mode, delayed trigger mode 0 or delayed trigger mode 1
     Check to see that A-D conversion is completed before reading the target ADi register. (Check the ADIC register's IR bit to see if A-D conversion is completed.)
   - When operating in repeat mode or repeat sweep mode 0 or 1
     Use the main clock for CPU clock directly without dividing it.

9. If A-D conversion is forcibly terminated while in progress by setting the ADCON0 register's ADST bit to "0" (A-D conversion halted), the conversion result of the A-D converter is indeterminate. The contents of ADi registers irrelevant to A-D conversion may also become indeterminate. If while A-D conversion is underway the ADST bit is cleared to "0" in a program, ignore the values of all ADi registers.

10. When setting the ADST bit in the ADCON register to "0" and terminating forcefully by a program in single sweep conversion mode, A-D delayed trigger mode 0 and A-D delayed trigger mode 1 during A-D converting operation, the A-D interrupt request may be generated. If this causes a problemm, set the ADST bit to "0" after an interrupt is disabled.

## 1.11 Precautions for Programmable I/O Ports

1. If a low-level signal is applied to the $\overline{SD}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the $P7_2$ to $P7_5$, $P8_0$ and $P8_1$ pins go to a high-impedance state.

2. Setting the SM32 bit in the S3C register to “1” causes the $P3_2$ pin to go to a high-impedance state. Similarly, setting the SM42 bit in the S4C register to “1” causes the $P9_6$ pin to go to a high-impedance state.

3. When the INV03 bit of the INVC0 register is "1"(three-phase motor control timer output enabled), it becomes the following by the $\overline{SD}$ function when "L" is input to the $P8_5$ /$\overline{NMI}$/$\overline{SD}$ pin.

   - When the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the U/ $\overline{U}$/ V/ $\overline{V}$/ W/ $\overline{W}$ pins go to a high-impedance state.
   - When the TB2SC register IVPCR1 bit = “0” (three-phase output forcible cutoff by input on $\overline{SD}$ pin disabled), the U/ $\overline{U}$/ V/ $\overline{V}$/ W/ $\overline{W}$ pins go to a normal port.

   When the $\overline{SD}$ function isn't used, set to "0" (Input) in $PD8_5$ and pullup to "H" in the $P8_5$ /$\overline{NMI}$/$\overline{SD}$ pin from outside.

## 1.12 Electric Characteristic Differences Between Mask ROM and Flash Memory Version Microcomputers

Flash memory version and mask ROM version may have different characteristics, operating margin, noise tolerated dose, noise width dose in electrical characteristics due to internal ROM, different layout pattern, etc. When switching to the mask ROM version, conduct equivalent tests as system evaluation tests conducted in the flush memory version.

## 1.13 Precautions for Flash Memory Version

### 1.13.1 Precautions for Functions to Inhibit Rewriting Flash Memory Rewrite

ID codes are stored in addresses 0FFFDF16, 0FFFE316, 0FFFEB16, 0FFFEF16, 0FFFF316, 0FFFF716, and 0FFFFB16. If wrong data are written to theses addresses, the flash memory cannot be read or written in standard serial I/O mode.

The ROMCP register is mapped in address 0FFFFF16. If wrong data is written to this address, the flash memory cannot be read or written in parallel I/O mode.

In the flash memory version of microcomputer, these addresses are allocated to the vector addresses (H) of fixed vectors.

### 1.13.2 Precautions for Stop mode

When shifting to stop mode, the following settings are required:

• Set the FMR01 bit to “0” (CPU rewrite mode disabled) and disable DMA transfers before setting the CM10 bit to “1” (stop mode).

• Execute the JMP.B instruction subsequent to the instruction which sets the CM10 bit to “1” (stop mode)

```
Example program     BSET    0, CM1      ; Stop mode
                    JMP.B   L1
                L1:
                    Program after returning from stop mode
```

### 1.13.3 Precautions for Wait mode

When shifting to wait mode, set the FMR01 bit to “0” (CPU rewrite mode diabled) before executing the WAIT instruction.

### 1.13.4 Precautions for Low power dissipation mode, ring oscillator low power dissipation mode

If the CM05 bit is set to “1” (main clock stop), the following commands must not be executed.

• Program
• Block erase
• Erase all unlocked blocks
• Lock bit program

### 1.13.5 Writing command and data

Write the command code and data at even addresses.

### 1.13.6 Precautions for Program Command

Write ‘xx4016’ in the first bus cycle and write data to the write address in the second bus cycle, and an auto program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same even address as the write address specified in the second bus cycle.

### 1.13.7 Operation speed

Before entering CPU rewrite mode (EW0 or EW1 mode), select 10 MHz or less for CPU clock using the CM0 register's CM06 bit and CM1 register's CM17–6 bits. Also, set the PM1 register's PM17 bit to 1 (with wait state).

### 1.13.8 Instructions inhibited against use

The following instructions cannot be used in EW0 mode because the flash memory's internal data is referenced: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

### 1.13.9 Interrupts

EW0 Mode

- Any interrupt which has a vector in the variable vector table can be used providing that its vector is transferred into the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts can be used because the FMR0 register and FMR1 register are initialized when one of those interrupts occurs. The jump addresses for those interrupt service routines should be set in the fixed vector table.

  Because the rewrite operation is halted when a $\overline{\text{NMI}}$ or watchdog timer interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.
- The address match interrupt cannot be used because the flash memory's internal data is referenced.

EW1 Mode

- Make sure that any interrupt which has a vector in the variable vector table or address match interrupt will not be accepted during the auto program or auto erase period.
- Avoid using watchdog timer interrupts.
- The $\overline{\text{NMI}}$ interrupt can be used because the FMR0 register and FMR1 register are initialized when this interrupt occurs. The jump address for the interrupt service routine should be set in the fixed vector table.

  Because the rewrite operation is halted when a $\overline{\text{NMI}}$ interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

### 1.13.10 How to access

To set the FMR01, FMR02, or FMR11 bit to "1", write "0" and then "1" in succession. This is necessary to ensure that no interrupts or DMA transfers will occur before writing "1" after writing "0". Also only when $\overline{\text{NMI}}$ pin is "H" level.

### 1.13.11 Writing in the user ROM area

EW0 Mode

- If the power supply voltage drops while rewriting any block in which the rewrite control program is stored, a problem may occur that the rewrite control program is not correctly rewritten and, consequently, the flash memory becomes unable to be rewritten thereafter. In this case, standard serial I/O or parallel I/O mode should be used.

EW1 Mode

- Avoid rewriting any block in which the rewrite control program is stored.

### 1.13.12 DMA transfer

In EW1 mode, make sure that no DMA transfers will occur while the FMR0 register's FMR00 bit = 0 (during the auto program or auto erase period).

### 1.13.13 Regarding Programming/Erasure Times and Execution Time

As the number of programming/erasure times increases, so does the execution time for software commands (Program, Block Erase, Erase All Unlock Blocks, and Lock Bit Program). Especially when the number of programming/erasure times exceeds 1,000, the software command execution time is noticeably extended. Therefore, the software command wait time that is set must be greater than the maximum rated value of electrical characteristics.

The software commands are aborted by hardware reset 1, hardware reset 2, $\overline{\text{NMI}}$ interrupt, and watchdog timer interrupt. If a software command is aborted by such reset or interrupt, the block that was in process must be erased before reexecuting the aborted command.

**RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER**
**USAGE NOTES REFERENCE BOOK**
**M16C/28 Group Rev.0.60**

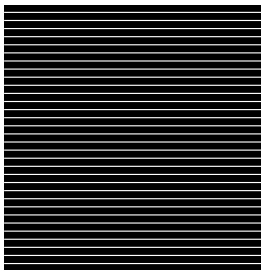
---

Editioned by
Committee of editing of RENESAS Semiconductor Usage Notes Reference Book

---

This book, or parts thereof, may not be reproduced in any form without permission of Renesas Technology Corporation.
Copyright © 2003. Renesas Technology Corporation, All rights reserved.

# M16C/28 Group
# Usage Notes Reference Book



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan