

**16**

# M16C/64 Group

## Hardware Manual

RENESAS MCU  
M16C FAMILY / M16C/60 SERIES

PRELIMINARY

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

Rev.0.64  
Revision Date: Oct 12, 2007

Renesas Technology  
[www.renesas.com](http://www.renesas.com)

Hardware Manual

## Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com> )
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
  - (1) artificial life support devices or systems
  - (2) surgical implantations
  - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
  - (4) any other purposes that pose a direct threat to human lifeRenesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

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

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

### 1. Handling of Unused Pins

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

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

### 2. Processing at Power-on

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

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

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

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

### 3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

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

### 4. Clock Signals

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

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

### 5. Differences between Products

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

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

# HOW TO USE THIS MANUAL

## 1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual.

The manual comprises an overview of the product; descriptions of the CPU, system control functions, peripheral functions, and electrical characteristics; and usage notes.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the M16C/64 Group. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Technology Web site.

Document Type	Description	Document Title	Document No.
Datasheet	Hardware overview and electrical characteristics	M16C/64 Group Datasheet	REJ03B0216
Hardware manual	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	M16C/64 Group Hardware Manual	This hardware manual
Application note	Information on using peripheral functions and application examples Sample programs Information on writing programs in assembly language and C	Available from Renesas Technology Web site.	
Renesas technical update	Product specifications, updates on documents, etc.		

## 2. Notation of Numbers and Symbols

The notation conventions for register names, bit names, numbers, and symbols used in this manual are described below.

### (1) Register Names, Bit Names, and Pin Names

Registers, bits, and pins are referred to in the text by symbols. The symbol is accompanied by the word “register,” “bit,” or “pin” to distinguish the three categories.

Examples    the PM03 bit in the PM0 register  
              P3\_5 pin, VCC pin

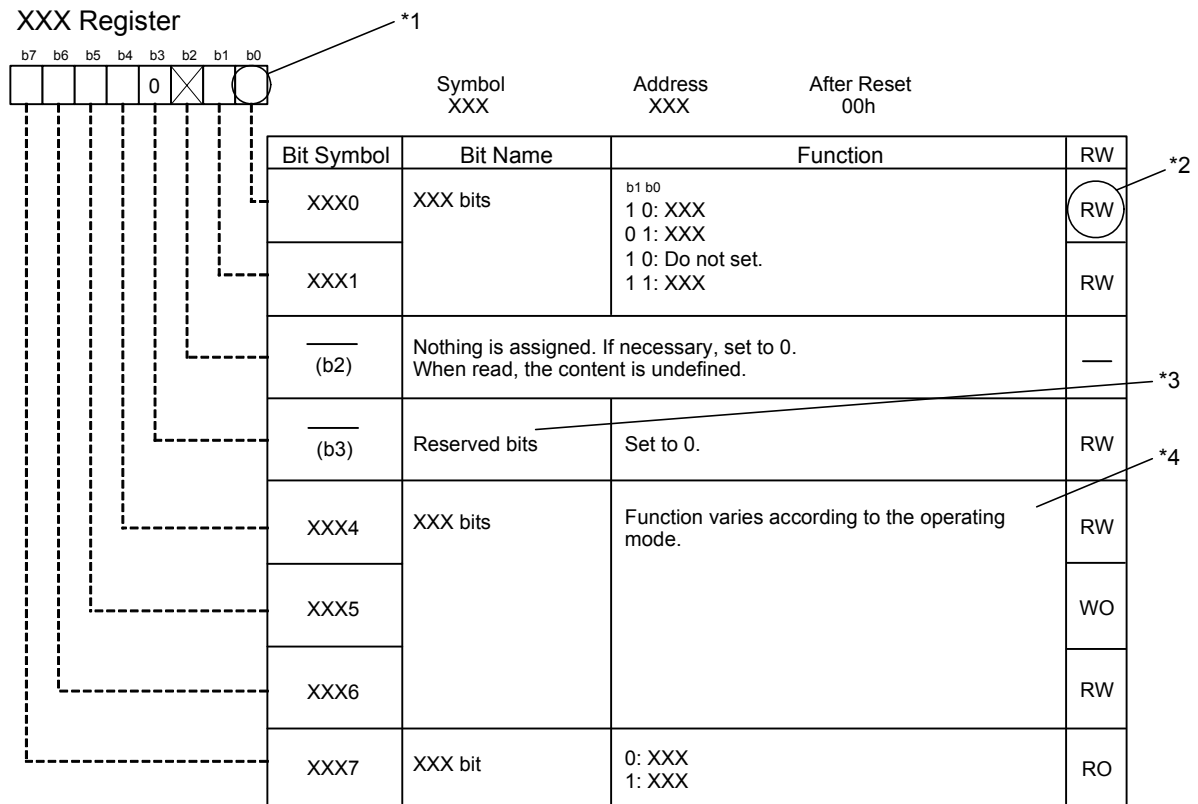
### (2) Notation of Numbers

The indication “b” is appended to numeric values given in binary format. However, nothing is appended to the values of single bits. The indication “h” is appended to numeric values given in hexadecimal format. Nothing is appended to numeric values given in decimal format.

Examples    Binary: 11b  
              Hexadecimal: EFA0h  
              Decimal: 1234

### 3. Register Notation

The symbols and terms used in register diagrams are described below.



\*1  
Blank: Set to 0 or 1 according to the application.  
0: Set to 0.  
1: Set to 1.  
X: Nothing is assigned.

\*2  
RW: Read and write.  
RO: Read only.  
WO: Write only.  
—: Nothing is assigned.

\*3  
• Reserved bit  
Reserved bit. Set to specified value.

\*4  
• Nothing is assigned  
Nothing is assigned to the bit. As the bit may be used for future functions, if necessary, set to 0.  
• Do not set to a value  
Operation is not guaranteed when a value is set.  
• Function varies according to the operating mode.  
The function of the bit varies with the peripheral function mode. Refer to the register diagram for information on the individual modes.

#### 4. List of Abbreviations and Acronyms

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

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

IEBus is a registered trademark of NEC Electronics Corporation.

# SFR Page Reference

Address	Register	Symbol	Page
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	48
0005h	Processor Mode Register 1	PM1	49
0006h	System Clock Control Register 0	CM0	76
0007h	System Clock Control Register 1	CM1	77
0008h	Chip Select Control Register	CSR	55
0009h			
000Ah	Protect Register	PRCR	98
000Bh	Data Bank Register	DBR	67
000Ch	Oscillation Stop Detection Register	CM2	78
000Dh			
000Eh			
000Fh			
0010h	Program 2 Area Control Register	PRG2C	50
0011h			
0012h	Peripheral Clock Select Register	PCLKR	79
0013h			
0014h			
0015h	Clock Prescaler Reset Flag	CPSRF	141
0016h			
0017h			
0018h	Reset Source Determine Flag	RSTFR	46
0019h	Voltage Detection 2 Circuit Flag Register	VCR1	38
001Ah	Voltage Detection Circuit Operation Enable Register	VCR2	38
001Bh	Chip Select Expansion Control Register	CSE	62
001Ch	PLL Control Register 0	PLC0	80
001Dh			
001Eh	Processor Mode Register 2	PM2	79
001Fh	Low Voltage Detection Interrupt Register	D4INT	39
0020h			
0021h			
0022h			
0023h			
0024h			
0025h			
0026h			
0027h			
0028h			
0029h			
002Ah	Voltage Monitor 0 Circuit Control Register	VW0C	40
002Bh			
002Ch			
002Dh			
002Eh			
002Fh			
0030h			
0031h			
0032h			
0033h			
0034h			
0035h			
0036h			
0037h			
0038h			
0039h			
003Ah			
003Bh			
003Ch			
003Dh			
003Eh			
003Fh			
0040h			
0041h			

NOTE: 1. Blank columns are all reserved space. No access is allowed.

Address	Register	Symbol	Page
0042h	INT7 Interrupt Control Register	INT7IC	106
0043h	INT6 Interrupt Control Register	INT6IC	106
0044h	INT3 Interrupt Control Register	INT3IC	106
0045h	Timer B5 Interrupt Control Register	TB5IC	105
0046h	Timer B4 Interrupt Control Register, UART1 BUS Collision Detection Interrupt Control Register	TB4IC, U1BCNIC	105
0047h	Timer B3 Interrupt Control Register, UART0 BUS Collision Detection Interrupt Control Register	TB3IC, U0BCNIC	105
0048h	SI/O4 Interrupt Control Register, INT5 Interrupt Control Register	S4IC, INT5IC	106
0049h	SI/O3 Interrupt Control Register, INT4 Interrupt Control Register	S3IC, INT4IC	106
004Ah	UART2 BUS Collision Detection Interrupt Control Register	BCNIC	106
004Bh	DMA0 Interrupt Control Register	DM0IC	105
004Ch	DMA1 Interrupt Control Register	DM1IC	105
004Dh	Key Input Interrupt Control Register	KUPIC	105
004Eh	A/D Conversion Interrupt Control Register	ADIC	105
004Fh	UART2 Transmit Interrupt Control Register	S2TIC	105
0050h	UART2 Receive Interrupt Control Register	S2RIC	105
0051h	UART0 Transmit Interrupt Control Register	S0TIC	105
0052h	UART0 Receive Interrupt Control Register	S0RIC	105
0053h	UART1 Transmit Interrupt Control Register	S1TIC	105
0054h	UART1 Receive Interrupt Control Register	S1RIC	105
0055h	Timer A0 Interrupt Control Register	TA0IC	105
0056h	Timer A1 Interrupt Control Register	TA1IC	105
0057h	Timer A2 Interrupt Control Register	TA2IC	105
0058h	Timer A3 Interrupt Control Register	TA3IC	105
0059h	Timer A4 Interrupt Control Register	TA4IC	105
005Ah	Timer B0 Interrupt Control Register	TB0IC	105
005Bh	Timer B1 Interrupt Control Register	TB1IC	105
005Ch	Timer B2 Interrupt Control Register	TB2IC	105
005Dh	INT0 Interrupt Control Register	INT0IC	106
005Eh	INT1 Interrupt Control Register	INT1IC	106
005Fh	INT2 Interrupt Control Register	INT2IC	106
0060h			
0061h			
0062h			
0063h			
0064h			
0065h			
0066h			
0067h			
0068h			
0069h	DMA2 Interrupt Control Register	DM2IC	105
006Ah	DMA3 Interrupt Control Register	DM3IC	105
006Bh	UART5 BUS Collision Detection Interrupt Control Register	U5BCNIC	105
006Ch	UART5 Transmit Interrupt Control Register	S5TIC	105
006Dh	UART5 Receive Interrupt Control Register	S5RIC	105
006Eh	UART6 BUS Collision Detection Interrupt Control Register	U6BCNIC	105
006Fh	UART6 Transmit Interrupt Control Register	S6TIC	105
0070h	UART6 Receive Interrupt Control Register	S6RIC	105
0071h	UART7 BUS Collision Detection Interrupt Control Register	U7BCNIC	105
0072h	UART7 Transmit Interrupt Control Register	S7TIC	105
0073h	UART7 Receive Interrupt Control Register	S7RIC	105
0074h			
0075h			
0076h			
0077h			
0078h			
0079h			
007Ah			
007Bh			
007Ch			
007Dh			
007Eh			
007Fh			
D080h to D17Fh			



Address	Register	Symbol	Page
0180h	DMA0 Source Pointer	SAR0	128
0181h			
0182h			
0183h			
0184h	DMA0 Destination Pointer	DAR0	128
0185h			
0186h			
0187h			
0188h	DMA0 Transfer Counter	TCR0	128
0189h			
018Ah			
018Bh			
018Ch	DMA0 Control Register	DM0CON	128
018Dh			
018Eh			
018Fh			
0190h	DMA1 Source Pointer	SAR1	128
0191h			
0192h			
0193h			
0194h	DMA1 Destination Pointer	DAR1	128
0195h			
0196h			
0197h			
0198h	DMA1 Transfer Counter	TCR1	128
0199h			
019Ah			
019Bh			
019Ch	DMA1 Control Register	DM1CON	127
019Dh			
019Eh			
019Fh			
01A0h	DMA2 Source Pointer	SAR2	128
01A1h			
01A2h			
01A3h			
01A4h	DMA2 Destination Pointer	DAR2	128
01A5h			
01A6h			
01A7h			
01A8h	DMA2 Transfer Counter	TCR2	128
01A9h			
01AAh			
01ABh			
01ACh	DMA2 Control Register	DM2CON	127
01ADh			
01AEh			
01AFh			
01B0h	DMA3 Source Pointer	SAR3	128
01B1h			
01B2h			
01B3h			
01B4h	DMA3 Destination Pointer	DAR3	128
01B5h			
01B6h			
01B7h			
01B8h	DMA3 Transfer Counter	TCR3	128
01B9h			
01BAh			
01BBh			
01BCh	DMA3 Control Register	DM3CON	128
01BDh			
01BEh			
01BFh			
01C0h			
01C1h			
01C2h			

Address	Register	Symbol	Page
01C3h			
01C4h			
01C5h			
01C6h			
01C7h			
01C8h	Timer B Count Source Select Register 0	TBCS0	157
01C9h	Timer B Count Source Select Register 1	TBCS1	157
01CAh			
01CBh			
01CCh			
01CDh			
01CEh			
01CFh			
01D0h	Timer A Count Source Select Register 0	TACS0	141
01D1h	Timer A Count Source Select Register 1	TACS1	141
01D2h	Timer A Count Source Select Register 2	TACS2	142
01D3h			
01D4h			
01D5h	Timer A Waveform Output Function Select Register	TAPOFS	142
01D6h			
01D7h			
01D8h			
01D9h			
01DAh			
01DBh			
01DEh			
01DCh			
01DDh			
01DFh			
01E0h			
01E1h			
01E2h			
01E3h			
01E4h			
01E5h			
01E6h			
01E7h			
01E8h	Timer B Count Source Select Register 2	TBCS2	157
01E9h	Timer B Count Source Select Register 3	TBCS3	157
01EAh			
01EBh			
01ECh			
01EDh			
01EEh			
01EFh			
01F0h			
01F1h			
01F2h			
01F3h			
01F4h			
01F5h			
01F6h			
01F7h			
01F8h			
01F9h			
01FAh			
01FBh			
01FCh			
01FDh			
01FEh			
01FFh			
0200h			
0201h			
0202h			
0203h			
0204h			

NOTE: 1. Blank columns are all reserved space. No access is allowed.

Address	Register	Symbol	Page
0205h	Interrupt Source Select Register 3	IFSR3A	114
0206h	Interrupt Source Select Register 2	IFSR2A	114
0207h	Interrupt Source Select Register	IFSR	113
0208h			
0209h			
020Ah			
020Bh			
020Ch			
020Dh			
020Eh	Address Match Interrupt Enable Register	AIER	117
020Fh	Address Match Interrupt Enable Register 2	AIER2	117
0210h	Address Match Interrupt Register 0	RMAD0	117
0211h			
0212h			
0213h			
0214h	Address Match Interrupt Register 1	RMAD1	117
0215h			
0216h			
0217h			
0218h	Address Match Interrupt Register 2	RMAD2	117
0219h			
021Ah			
021Bh			
021Ch	Address Match Interrupt Register 3	RMAD3	117
021Dh			
021Eh			
021Fh			
0220h	Flash Memory Control Register 0	FMR0	272
0221h	Flash Memory Control Register 1	FMR1	273
0222h	Flash Memory Control Register 2	FMR2	274
0223h			
0224h			
0225h			
0226h			
0227h			
0228h			
0229h			
022Ah			
022Bh			
022Ch			
022Dh			
022Eh			
022Fh			
0230h	Flash Memory Control Register 6	FMR6	275
0231h			
0232h			
0233h			
0234h			
0235h			
0236h			
0237h			
0238h			
0239h			
023Ah			
023Bh			
023Ch			
023Dh			
023Eh			
023Fh			
0240h			
0241h			
0242h			
0243h			
0244h	UART0 Special Mode Register 4	U0SMR4	185
0245h	UART0 Special Mode Register 3	U0SMR3	184
0246h	UART0 Special Mode Register 2	U0SMR2	184
0247h	UART0 Special Mode Register	U0SMR	183
0248h	UART0 Transmit/Receive Mode Register	U0MR	180

Address	Register	Symbol	Page
0249h	UART0 Bit Rate Register	U0BRG	180
024Ah	UART0 Transmit Buffer Register	U0TB	179
024Bh			
024Ch	UART0 Transmit/Receive Control Register 0	U0C0	181
024Dh	UART0 Transmit/Receive Control Register 1	U0C1	182
024Eh	UART0 Receive Buffer Register	U0RB	179
024Fh			
0250h	UART Transmit/Receive Control Register 2	U0CON	183
0251h			
0252h			
0253h			
0254h	UART1 Special Mode Register 4	U1SMR4	185
0255h	UART1 Special Mode Register 3	U1SMR3	184
0256h	UART1 Special Mode Register 2	U1SMR2	184
0257h	UART1 Special Mode Register	U1SMR	183
0258h	UART1 Transmit/Receive Mode Register	U1MR	180
0259h	UART1 Bit Rate Register	U1BRG	180
025Ah	UART1 Transmit Buffer Register	U1TB	170
025Bh			
025Ch	UART1 Transmit/Receive Control Register 0	U1C0	180
025Dh	UART1 Transmit/Receive Control Register 1	U1C1	182
025Eh	UART1 Receive Buffer Register	U1RB	170
025Fh			
0260h			
0261h			
0262h			
0263h			
0264h	UART2 Special Mode Register 4	U2SMR4	185
0265h	UART2 Special Mode Register 3	U2SMR3	184
0266h	UART2 Special Mode Register 2	U2SMR2	184
0267h	UART2 Special Mode Register	U2SMR	183
0268h	UART2 Transmit/Receive Mode Register	U2MR	180
0269h	UART2 Bit Rate Register	U2BRG	180
026Ah	UART2 Transmit Buffer Register	U2TB	170
026Bh			
026Ch	UART2 Transmit/Receive Control Register 0	U2C0	181
026Dh	UART2 Transmit/Receive Control Register 1	U2C1	182
026Eh	UART2 Receive Buffer Register	U2RB	179
026Fh			
0270h	SI/O3 Transmit/Receive Register	S3TRR	224
0271h			
0272h	SI/O3 Control Register	S3C	224
0273h	SI/O3 Bit Rate Register	S3BRG	224
0274h	SI/O4 Transmit/Receive Register	S4TRR	224
0275h			
0276h	SI/O4 Control Register	S4C	224
0277h	SI/O4 Bit Rate Register	S4BRG	224
0278h	SI/O34 Control Register 2	S34C2	225
0279h			
027Ah			
027Bh			
027Ch			
027Dh			
027Eh			
027Fh			
0280h			
0281h			
0282h			
0283h			
0284h	UART5 Special Mode Register 4	U5SMR4	185
0285h	UART5 Special Mode Register 3	U5SMR3	184
0286h	UART5 Special Mode Register 2	U5SMR2	184
0287h	UART5 Special Mode Register	U5SMR	183
0288h	UART5 Transmit/Receive Mode Register	U5MR	180
0289h	UART5 Bit Rate Register	U5BRG	180

NOTE: 1. Blank columns are all reserved space. No access is allowed.

Address	Register	Symbol	Page
028Ah 028Bh	UART5 Transmit Buffer Register	U5TB	179
028Ch	UART5 Transmit/Receive Control Register 0	U5C0	181
028Dh	UART5 Transmit/Receive Control Register 1	U5C1	182
028Eh 028Fh	UART5 Receive Buffer Register	U5RB	170
0290h			
0291h			
0292h			
0293h			
0294h	UART6 Special Mode Register 4	U6SMR4	185
0295h	UART6 Special Mode Register 3	U6SMR3	184
0296h	UART6 Special Mode Register 2	U6SMR2	184
0297h	UART6 Special Mode Register	U6SMR	183
0298h	UART6 Transmit/Receive Mode Register	U6MR	180
0299h	UART6 Bit Rate Register	U6BRG	180
029Ah 029Bh	UART6 Transmit Buffer Register	U6TB	179
029Ch	UART6 Transmit/Receive Control Register 0	U6C0	181
029Dh	UART6 Transmit/Receive Control Register 1	U6C1	182
029Eh	UART6 Receive Buffer Register	U6RB	179
029Fh			
02A0h			
02A1h			
02A2h			
02A3h			
02A4h	UART7 Special Mode Register 4	U7SMR4	185
02A5h	UART7 Special Mode Register 3	U7SMR3	184
02A6h	UART7 Special Mode Register 2	U7SMR2	184
02A7h	UART7 Special Mode Register	U7SMR	183
02A8h	UART7 Transmit/Receive Mode Register	U7MR	180
02A9h	UART7 Bit Rate Register	U7BRG	180
02AAh 02ABh	UART7 Transmit Buffer Register	U7TB	179
02ACh	UART7 Transmit/Receive Control Register 0	U7C0	181
02ADh	UART7 Transmit/Receive Control Register 1	U7C1	182
02AEh 02AFh	UART7 Receive Buffer Register	U7RB	179
02B0h to 02FFh			
0300h	Timer B3,4,5 Count Start Flag	TBSR	156
0301h			
0302h 0303h	Timer A1-1 Register	TA11	170
0304h 0305h	Timer A2-1 Register	TA21	170
0306h 0307h	Timer A4-1 Register	TA41	170
0308h	Three-Phase PWM Control Register 0	INVC0	167
0309h	Three-Phase PWM Control Register 1	INVC1	168
030Ah	Three-Phase Output Buffer Register 0	IDB0	169
030Bh	Three-Phase Output Buffer Register 1	IDB1	169
030Ch	Dead Time Timer	DTT	169
030Dh	Timer B2 Interrupt Generation Frequency Set Counter	ICTB2	169
030Eh			
030Fh			
0310h 0311h	Timer B3 Register	TB3	155
0312h 0313h	Timer B4 Register	TB4	155
0314h 0315h	Timer B5 Register	TB5	155
0316h			
0317h			
0318h			
0319h			

Address	Register	Symbol	Page
031Ah			
031Bh	Timer B3 Mode Register	TB3MR	155
031Ch	Timer B4 Mode Register	TB4MR	155
031Dh	Timer B5 Mode Register	TB5MR	155
031Eh			
031Fh			
0320h	Count Start Flag	TABSR	156
0321h			
0322h	One-Shot Start Flag	ONSF	140
0323h	Trigger Select Register	TRGSR	140
0324h	Up/Down Flag	UDF	139
0325h			
0326h 0327h	Timer A0 Register	TA0	152
0328h 0329h	Timer A1 Register	TA1	152
032Ah 032Bh	Timer A2 Register	TA2	152
032Ch 032Dh	Timer A3 Register	TA3	152
032Eh 032Fh	Timer A4 Register	TA4	152
0330h 0331h	Timer B0 Register	TB0	155
0332h 0333h	Timer B1 Register	TB1	155
0334h 0335h	Timer B2 Register	TB2	155
0336h	Timer A0 Mode Register	TA0MR	138
0337h	Timer A1 Mode Register	TA1MR	138
0338h	Timer A2 Mode Register	TA2MR	138
0339h	Timer A3 Mode Register	TA3MR	138
033Ah	Timer A4 Mode Register	TA4MR	138
033Bh	Timer B0 Mode Register	TB0MR	155
033Ch	Timer B1 Mode Register	TB1MR	155
033Dh	Timer B2 Mode Register	TB2MR	155
033Eh	Timer B2 Special Mode Register	TB2SC	170
033Fh			
0340h			
0341h			
0342h			
0343h			
0344h			
0345h			
0346h			
0347h			
0348h			
0349h			
034Ah			
034Bh			
034Ch			
034Dh			
034Eh			
034Fh			
0350h			
0351h			
0352h			
0353h			
0354h			
0355h			
0356h			
0357h			
0358h			
0359h			
035Ah			
035Bh			

NOTE: 1. Blank columns are all reserved space. No access is allowed.

Address	Register	Symbol	Page
035Ch			
035Dh			
035Eh			
035Fh			
0360h	Pull-Up Control Register 0	PUR0	258
0361h	Pull-Up Control Register 1	PUR1	258
0362h	Pull-Up Control Register 2	PUR2	259
0363h			
0364h			
0365h			
0366h	Port Control Register	PCR	259
0367h			
0368h			
0369h			
036Ah			
036Bh			
036Ch			
036Dh			
036Eh			
036Fh			
0370h			
0371h			
0372h			
0373h			
0374h			
0375h			
0376h			
0377h			
0378h			
0379h			
037Ah			
037Bh			
037Ch	Count Source Protection Mode Register	CSPR	120
037Dh	Watchdog Timer Reset Register	WDTR	119
037Eh	Watchdog Timer Start Register	WDTS	119
037Fh	Watchdog Timer Control Register	WDC	119
0380h			
0381h			
0382h			
0383h			
0384h			
0385h			
0386h			
0387h			
0388h			
0389h			
038Ah			
038Bh			
038Ch			
038Dh			
038Eh			
038Fh			
0390h	DMA2 Source Select Register	DM2SL	125
0391h			
0392h	DMA3 Source Select Register	DM3SL	125
0393h			
0394h			
0395h			
0396h			
0397h			
0398h	DMA0 Source Select Register	DM0SL	125
0399h			
039Ah	DMA1 Source Select Register	DM1SL	125
039Bh			
039Ch			
039Dh			
039Eh			
039Fh			

Address	Register	Symbol	Page
03A0h			
03A1h			
03A2h			
03A3h			
03A4h			
03A5h			
03A6h			
03A7h			
03A8h			
03A9h			
03AAh			
03ABh			
03ACh			
03ADh			
03AEh			
03AFh			
03B0h			
03B1h			
03B2h			
03B3h			
03B4h			
03B5h			
03B6h			
03B7h			
03B8h			
03B9h			
03BAh			
03BBh			
03BCh	CRC Data Register	CRCD	248
03BDh			
03BEh	CRC Input Register	CRCIN	248
03BFh			
03C0h	A/D Register 0	AD0	233
03C1h			
03C2h	A/D Register 1	AD1	233
03C3h			
03C4h	A/D Register 2	AD2	233
03C5h			
03C6h	A/D Register 3	AD3	233
03C7h			
03C8h	A/D Register 4	AD4	233
03C9h			
03CAh	A/D Register 5	AD5	233
03CBh			
03CCh	A/D Register 6	AD6	233
03CDh			
03CEh	A/D Register 7	AD7	233
03CFh			
03D0h			
03D1h			
03D2h			
03D3h			
03D4h	A/D Control Register 2	ADCON2	233
03D5h			
03D6h	A/D Control Register 0	ADCON0	232
03D7h	A/D Control Register 1	ADCON1	232
03D8h	D/A0 Register	DA0	247
03D9h			
03DAh	D/A1 Register	DA1	247
03DBh			
03DCh	D/A Control Register	DACON	247
03DDh			
03DEh			
03DFh			
03E0h	Port P0 Register	P0	257
03E1h	Port P1 Register	P1	257
03E2h	Port P0 Direction Register	PD0	256
03E3h	Port P1 Direction Register	PD1	256

NOTE: 1. Blank columns are all reserved space. No access is allowed.

Address	Register	Symbol	Page
03E4h	Port P2 Register	P2	257
03E5h	Port P3 Register	P3	257
03E6h	Port P2 Direction Register	PD2	256
03E7h	Port P3 Direction Register	PD3	256
03E8h	Port P4 Register	P4	257
03E9h	Port P5 Register	P5	257
03EAh	Port P4 Direction Register	PD4	256
03EBh	Port P5 Direction Register	PD5	256
03ECh	Port P6 Register	P6	257
03EDh	Port P7 Register	P7	257
03EEh	Port P6 Direction Register	PD6	256
03EFh	Port P7 Direction Register	PD7	256
03F0h	Port P8 Register	P8	257
03F1h	Port P9 Register	P9	257
03F2h	Port P8 Direction Register	PD8	256
03F3h	Port P9 Direction Register	PD9	256
03F4h	Port P10 Register	P10	257
03F5h			
03F6h	Port P10 Direction Register	PD10	256
03F7h			
03F8h			
03F9h			
03FAh			
03FBh			
03FCh			
03FDh			
03FEh			
03FFh			
D000h to D7FFh			

OFS1	Optional Feature Select Address	OFS1	269
------	---------------------------------	------	-----

NOTE: 1. Blank columns are all reserved space. No access is allowed.

# TABLE OF CONTENTS

SFR Page Reference .....	B-1
1. Overview .....	1
1.1 Features .....	1
1.1.1 Applications .....	1
1.2 Specifications .....	2
1.3 Product List .....	4
1.4 Block Diagram .....	6
1.5 Pin Assignments.....	7
1.6 Pin Functions.....	11
2. Central Processing Unit (CPU).....	14
2.1 Data Registers (R0, R1, R2 and R3).....	14
2.2 Address Registers (A0 and A1).....	15
2.3 Frame Base Registers (FB).....	15
2.4 Interrupt Table Register (INTB) .....	15
2.5 Program Counter (PC) .....	15
2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP).....	15
2.7 Static Base Register (SB).....	15
2.8 Flag Register (FLG).....	15
2.8.1 Carry Flag (C Flag).....	15
2.8.2 Debug Flag (D Flag).....	15
2.8.3 Zero Flag (Z Flag).....	15
2.8.4 Sign Flag (S Flag).....	15
2.8.5 Register Bank Select Flag (B Flag) .....	15
2.8.6 Overflow Flag (O Flag) .....	15
2.8.7 Interrupt Enable Flag (I Flag).....	15
2.8.8 Stack Pointer Select Flag (U Flag) .....	16
2.8.9 Processor Interrupt Priority Level (IPL) .....	16
2.8.10 Reserved Space .....	16

3.	Memory .....	17
4.	Special Function Registers (SFRs) .....	18
5.	Reset.....	32
5.1	Hardware Reset 1 .....	32
5.1.1	Reset on a Stable Supply Voltage .....	32
5.1.2	Power-on Reset.....	32
5.2	Brown-out Reset.....	35
5.3	Software Reset.....	35
5.4	Watchdog Timer Reset.....	35
5.5	Oscillation Stop Detection Reset.....	35
5.6	Internal Space .....	36
6.	Voltage Detection Circuit .....	37
6.1	Brown-out Reset.....	41
6.2	Low Voltage Detection Interrupt .....	42
6.3	Limitations on Exiting Stop Mode .....	44
6.4	Limitations on Exiting Wait Mode .....	44
6.5	Cold Start-up / Warm Start-up Discrimination.....	45
7.	Processor Mode .....	47
7.1	Types of Processor Mode.....	47
7.2	Setting Processor Modes .....	47
7.3	Internal Memory .....	51
8.	Bus .....	53
8.1	Bus Mode .....	53
8.1.1	Separate Bus.....	53
8.1.2	Multiplexed Bus .....	53
8.2	Bus Control.....	54
8.2.1	Address Bus .....	54
8.2.2	Data Bus.....	54
8.2.3	Chip Select Signal .....	54
8.2.4	Read and Write Signals.....	57
8.2.5	ALE Signal.....	57
8.2.6	$\overline{\text{RDY}}$ Signal.....	58

8.2.7	$\overline{\text{HOLD}}$ Signal.....	59
8.2.8	BCLK Output .....	59
8.2.9	External Bus Status When Internal Area IS Accessed .....	61
8.2.10	Software Wait .....	61
9.	Memory Space Expansion Function.....	66
9.2	4-Mbyte Mode .....	66
9.2.1	Addresses 04000h to 3FFFFh, C0000h to FFFFFh .....	66
9.2.2	Addresses 40000h to BFFFFh .....	66
10.	Clock Generation Circuit .....	74
10.1	Type of the Clock Generation Circuit.....	74
10.1.1	Main Clock.....	81
10.1.2	Sub Clock .....	82
10.1.3	125 kHz On-Chip Oscillator Clock (fOCO-S).....	83
10.1.4	PLL Clock .....	83
10.2	CPU Clock and Peripheral Function Clock.....	86
10.2.1	CPU Clock and BCLK.....	86
10.2.2	Peripheral Function Clock (f1, fC32) .....	87
10.3	Clock Output Function.....	87
10.4	Power Control.....	88
10.4.1	Normal Operating Mode .....	88
10.4.2	Wait Mode .....	90
10.4.3	Stop Mode .....	92
10.5	System Clock Protection Function .....	95
10.6	Oscillation Stop and Re-Oscillation Detect Function .....	96
10.6.1	Operation When CM27 bit = 0 (Oscillation Stop Detection Reset).....	96
10.6.2	Operation When CM27 bit = 1 (Oscillation Stop and Re-oscillation Detect Interrupt).....	96
10.6.3	How to Use Oscillation Stop and Re-Oscillation Detect Function.....	97
11.	Protection .....	98
12.	Interrupt.....	99
12.1	Type of Interrupts .....	99
12.2	Software Interrupts .....	100
12.2.1	Undefined Instruction Interrupt .....	100



12.2.2	Overflow Interrupt.....	100
12.2.4	INT Instruction Interrupt.....	100
12.3	Hardware Interrupts.....	101
12.3.1	Special Interrupts.....	101
12.3.2	Peripheral Function Interrupts .....	101
12.4	Interrupts and Interrupt Vector.....	102
12.4.1	Fixed Vector Tables .....	102
12.4.2	Relocatable Vector Tables .....	103
12.5	Interrupt Control .....	105
12.5.1	I Flag.....	107
12.5.2	IR Bit.....	107
12.5.3	Bits ILVL2 to ILVL0 and IPL.....	107
12.5.4	Interrupt Sequence .....	108
12.5.5	Interrupt Response Time .....	109
12.5.6	Variation of IPL when Interrupt Request IS Accepted .....	109
12.5.7	Saving Registers .....	110
12.5.8	Returning from an Interrupt Routine .....	111
12.5.9	Interrupt Priority.....	111
12.5.10	Interrupt Priority Level Select Circuit .....	111
12.6	$\overline{\text{INT}}$ Interrupt .....	113
12.7	$\overline{\text{NMI}}$ Interrupt .....	115
12.8	Key Input Interrupt.....	115
12.9	Address Match Interrupt.....	116
13.	Watchdog Timer .....	118
13.1	Count Source Protection Mode Disabled .....	121
13.2	Count Source Protection Mode Enabled .....	122
14.	DMAC.....	123
14.1	Transfer Cycles .....	129
14.1.1	Effect of Source and Destination Addresses .....	129
14.1.2	Effect of BYTE Pin Level .....	129
14.1.3	Effect of Software Wait .....	129
14.1.4	Effect of $\overline{\text{RDY}}$ Signal.....	129
14.2	DMA Transfer Cycles .....	131
14.3	DMA Enabled .....	132

14.4	DMA Request.....	132
14.5	Channel Priority and DMA Transfer Timing.....	133
15.	Timers.....	134
15.1	Timer A.....	137
15.1.1	Timer Mode .....	143
15.1.2	Event Counter Mode.....	144
15.1.3	One-Shot Timer Mode .....	149
15.1.4	Pulse Width Modulation (PWM) Mode.....	151
15.2	Timer B.....	154
15.2.1	Timer Mode .....	158
15.2.2	Event Counter Mode.....	160
15.2.3	Pulse Period and Pulse Width Measurement Modes .....	162
16.	Three-Phase Motor Control Timer Function .....	165
17.	Serial Interface .....	175
17.1	UARTi (i = 0 to 2, 5 to 7).....	175
17.1.1	Clock Synchronous Serial I/O Mode.....	186
17.1.2	Clock Asynchronous Serial I/O (UART) Mode.....	194
17.1.3	Special Mode 1 (I <sup>2</sup> C mode) .....	202
17.1.4	Special Mode 2.....	212
17.1.5	Special Mode 3 (IE mode).....	216
17.1.6	Special Mode 4 (SIM Mode) (UART2).....	218
17.2	SI/O3 and SI/O4.....	223
17.2.1	SI/Oi Operation Timing.....	227
17.2.2	CLK Polarity Selection.....	227
17.2.3	Functions for Setting an SOUTi Initial Value.....	228
17.2.4	Functions for Selecting SOUTi State after Transmission.....	229
18.	A/D Converter.....	230
18.1	Mode Description .....	234
18.1.1	One-Shot Mode .....	234
18.1.2	Repeat Mode .....	236
18.1.3	Single Sweep Mode.....	238
18.1.4	Repeat Sweep Mode 0.....	240
18.1.5	Repeat Sweep Mode 1 .....	242

18.2	Conversion Rate.....	244
18.3	Extended Analog Input Pins .....	244
18.4	Current Consumption Reducing Function .....	244
18.5	Output Impedance of Sensor under A/D Conversion .....	245
19.	D/A Converter.....	246
19.1	Summary .....	246
20.	CRC Operation.....	248
21.	Programmable I/O Ports.....	250
21.1	Port Pi Direction Register (PDi Register, i = 0 to 10).....	250
21.2	Port Pi Register (Pi Register, i = 0 to 10) .....	250
21.3	Pull-up Control Register 0 to Pull-up Control Register 2 (Registers PUR0 to PUR2).....	250
21.4	Port Control Register (PCR Register) .....	250
22.	Flash Memory Version.....	263
22.1	Memory Map .....	264
22.1.1	Boot Mode .....	265
22.1.2	User Boot Function.....	265
22.2	Functions to Prevent Flash Memory from Rewriting .....	267
22.2.1	ROM Code Protect Function .....	267
22.2.2	ID Code Check Function .....	267
22.2.3	Forced Erase Function .....	268
22.2.4	Standard Serial I/O Mode Disable Function .....	268
22.3	CPU Rewrite Mode.....	270
22.3.1	EW0 Mode.....	271
22.3.2	EW1 Mode.....	271
22.3.3	Flash Memory Control Register (Registers FMR0, FMR1, FMR2 and FMR6).....	271
22.3.4	Precautions on CPU Rewrite Mode.....	282
22.3.5	Software Commands .....	284
22.3.6	Data Protect Function.....	290
22.3.7	Status Register .....	290
22.3.8	Full Status Check.....	292
22.4	Standard Serial I/O Mode .....	294

22.4.1	ID Code Check Function .....	294
22.4.2	Example of Circuit Application in the Standard Serial I/O Mode.....	298
22.5	Parallel I/O Mode.....	300
22.5.1	ROM Code Protect Function .....	300
23.	Electrical Characteristics .....	301
23.1	Electrical Characteristics .....	301
24.	Precautions .....	341
24.1	SFR .....	341
24.1.1	Register Settings .....	341
24.2	Reset.....	342
24.2.1	VCC1 .....	342
24.2.2	CNVSS .....	342
24.3	Bus .....	343
24.4	PLL Frequency Synthesizer .....	344
24.5	Power Control.....	345
24.6	Protect.....	347
24.7	Interrupt.....	348
24.7.1	Reading address 00000h .....	348
24.7.2	SP Setting.....	348
24.7.3	$\overline{\text{NMI}}$ Interrupt.....	348
24.7.4	Changing an Interrupt Generate Factor.....	349
24.7.5	$\overline{\text{INT}}$ Interrupt .....	349
24.7.6	Rewriting the Interrupt Control Register .....	350
24.7.7	Watchdog Timer Interrupt.....	351
24.8	DMAC.....	352
24.8.1	Write to the DMAE Bit in the DMiCON Register (i = 0 to 3).....	352
24.9	Timers .....	353
24.9.1	Timer A .....	353
24.9.2	Timer B .....	357
24.10	Serial Interface .....	360
24.10.1	Clock Synchronous Serial I/O.....	360
24.10.2	UART (Clock Asynchronous Serial I/O) Mode.....	362
24.10.3	Special Mode 1 (I2C Mode).....	362
24.10.4	Special Mode 4 (SIM Mode) .....	362

24.10.5	SI/O3, SI/O4 .....	363
24.11	A/D Converter.....	364
24.12	Programmable I/O Ports.....	366
24.13	Flash Memory Version.....	367
24.13.1	Functions to Inhibit Rewriting Flash Memory.....	367
24.13.2	Stop Mode .....	367
24.13.3	Wait Mode .....	367
24.13.4	Low Power Consumption Mode, On-Chip Oscillator Low Power Consumption Mode .....	367
24.13.5	Writing Command and Data .....	367
24.13.6	Program Command .....	367
24.13.7	Lock Bit Program Command .....	367
24.13.8	Operation Speed.....	368
24.13.9	Instructions Inhibited against Use.....	368
24.13.10	Interrupts .....	368
24.13.11	How to Access.....	368
24.13.12	Writing in the User ROM Area .....	368
24.13.13	DMA transfer .....	368
24.13.14	Programming / Erasing Endurance and Execution Time.....	369
24.14	Noise .....	370
Appendix 1.	Package Dimensions .....	371
Register Index		Appendix 1-372

**M16C/64 Group**RENESAS MCU

---

**1. Overview****1.1 Features**

The M16C/64 Group MCUs incorporate the M16C/60 Series CPU core and flash memory, employing sophisticated instructions for a high level of efficiency. With 1 Mbyte of address space (expandable to 4 Mbyte), this MCU is capable of executing instructions at high speed. In addition, the CPU core boasts a multiplier for high-speed operation processing.

Power consumption is low, and the M16C/64 Group supports operating modes that allow additional power control. The MCU also uses an anti-noise configuration to reduce emissions of electromagnetic noise and is designed to withstand EMI. Integration of many peripheral functions, including multifunction timer and serial interface, reduces the number of system components.

**1.1.1 Applications**

Audio, cameras, television, home appliance, office equipment, communication equipment, portable equipment, industrial equipment, etc.

## 1.2 Specifications

Tables 1.1 and 1.2 list Specifications Outline.

**Table 1.1 Specifications (1)**

Item	Function	Specification
CPU	Central processing unit	M16C/60 core (multiplier: 16-bit × 16-bit → 32 bits, multiply and accumulate instruction: 16 × 16 + 32 → 32 bits) <ul style="list-style-type: none"> <li>• Number of basic instructions: 91</li> <li>• Minimum instruction execution time: 40.0 ns ( f(BCLK) = 25 MHz, VCC1 = VCC2 = 2.7 to 5.5 V)</li> <li>• Operating modes: Single-chip, memory expansion, and microprocessor</li> </ul>
Memory	ROM, RAM, data flash	See Table 1.3 Product List.
Voltage Detection	Voltage detection circuit	Low-voltage detection unit
Clock	Clock generation circuit	<ul style="list-style-type: none"> <li>• 4 circuits: Main clock, sub clock, on-chip oscillator (125 kHz), PLL</li> <li>• Oscillation stop detection: Main clock oscillation stop detection and re-oscillation detection function</li> <li>• Frequency divider circuit: Divide ratio selectable from 1, 2, 4, 8 and 16</li> <li>• Low-power consumption modes: Wait mode, stop mode</li> </ul>
External Bus Expansion	Bus and memory expansion	<ul style="list-style-type: none"> <li>• Address space: 1 Mbyte</li> <li>• External bus interface: 0 to 3 wait states, chip select 4 outputs, memory area expansion function (up to 4 Mbytes), 3 V, 5 V interface</li> <li>• Bus format: Separate bus or multiplexed bus selectable, data bus width selectable (8 or 16 bits), number of address buses selectable (12, 16, or 20 buses)</li> </ul>
I/O Ports	Programmable I/O ports	<ul style="list-style-type: none"> <li>• CMOS I/O ports: 85, selectable pull-up resistor</li> <li>• Nch open drain ports: 3</li> </ul>
Interrupts		<ul style="list-style-type: none"> <li>• Interrupt vectors: 70</li> <li>• External interrupt input: 13 (<math>\overline{\text{NMI}}</math>, <math>\overline{\text{INT}} \times 8</math>, key input × 4)</li> <li>• Interrupt priority levels: 7</li> </ul>
Watchdog Timer		15 bits × 1 (with prescaler) Automatic reset start function selectable
DMA	DMAC	<ul style="list-style-type: none"> <li>• 4 channels, cycle steal mode</li> <li>• Trigger sources: 43</li> <li>• Transfer modes: 2 (single transfer, repeat transfer)</li> </ul>
Timer	Timer A	16-bit timer × 5 Timer mode, event counter mode, one shot timer mode, pulse width modulation (PWM) mode Event counter two-phase pulse signal processing (two-phase encoder input) × 3 channels
	Timer B	16-bit timer × 6 Timer mode, event counter mode, pulse period measurement mode, pulse width measurement mode
	Timer functions for three-phase motor control	Three-phase inverter control (timer A1, timer A2, timer A4, timer B2), On-chip dead time timer
Serial Interface	UART0 to UART2, UART5 to UART7	Clock synchronous/asynchronous × 6 channels I <sup>2</sup> C-bus, IEBus (1), special mode 2, SIM (UART2)
	SI/O3, SI/O4	Clock synchronization only × 2 channels

NOTE:

1. IEBus is a registered trademark of NEC Electronics Corporation.

**Table 1.2 Specifications (2)**

Item	Function	Specification
A/D Converter		10-bit resolution × 26 channels, including sample and hold function, Conversion time: 1.72 μs
D/A Converter		8-bit resolution × 2
CRC Calculation Circuit		CRC-CCITT ( $X^{16} + X^{12} + X^5 + 1$ ) compliant
Flash Memory		Programming and erasure power supply voltage: 2.7 V to 5.5 V Programming and erasure endurance: 100 times Program security: ROM code protect, ID code check
Debug Function		Functions on-chip debug, on-board flash rewrite function, address match × 4
Operation Frequency/Supply Voltage		25 MHz/VCC1 = VCC2 = 2.7 to 5.5 V
Power Consumption		20 mA (25 MHz/VCC1 = VCC2 = 3 V) 3.0 μA(VCC1 = VCC2 = 3 V, in stop mode)
Operating Temperature		-20°C to 85°C, -40°C to 85°C
Package		100-pin QFP: PRQP0100JD-B (Previous package code: 100P6F-A) 100-pin LQFP: PLQP0100KB-A (Previous package code: 100P6Q-A)



### 1.3 Product List

Table 1.3 lists product information. Figure 1.1 shows part numbers, memory sizes, and packages.

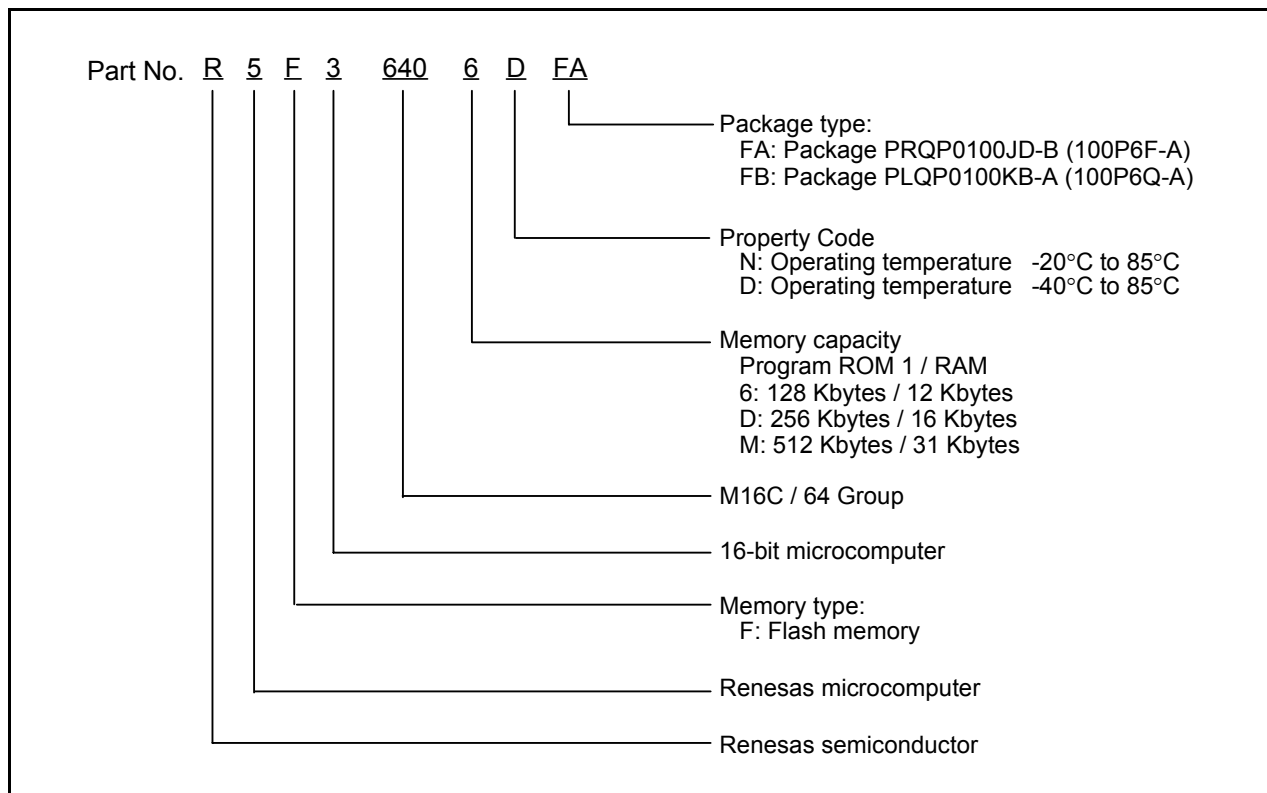
**Table 1.3 Product List**

Part No.	ROM Capacity			RAM Capacity	Package Code	Remarks
	Program ROM 1	Program ROM 2	Data Flash			
R5F36406NFA (D)	128 Kbytes	16 Kbytes	4 Kbytes × 2 blocks	12 Kbytes	PRQP0100JD-B	Operating temperature -20°C to 85°C
R5F36406NFB (D)					PLQP0100KB-A	
R5F3640DNFA (D)	256 Kbytes	16 Kbytes	4 Kbytes × 2 blocks	16 Kbytes	PRQP0100JD-B	
R5F3640DNFB (D)					PLQP0100KB-A	
R5F3640MNFA (D)	512 Kbytes	16 Kbytes	4 Kbytes × 2 blocks	31 Kbytes	PRQP0100JD-B	
R5F3640MNFB (D)					PLQP0100KB-A	
R5F36406DFA (D)	128 Kbytes	16 Kbytes	4 Kbytes × 2 blocks	12 Kbytes	PRQP0100JD-B	Operating temperature -40°C to 85°C
R5F36406DFB (D)					PLQP0100KB-A	
R5F3640DDFA (D)	256 Kbytes	16 Kbytes	4 Kbytes × 2 blocks	16 Kbytes	PRQP0100JD-B	
R5F3640DDFB (D)					PLQP0100KB-A	
R5F3640MDFA (D)	512 Kbytes	16 Kbytes	4 Kbytes × 2 blocks	31 Kbytes	PRQP0100JD-B	
R5F3640MDFB (D)					PLQP0100KB-A	

(D) : Under development

**NOTE:**

1. Previous package codes are as follows.  
 PRQP0100JD-B: 100P6F-A, PLQP0100KB-A: 100P6Q-A



**Figure 1.1 Correspondence of Part No., with Memory Size and Package**

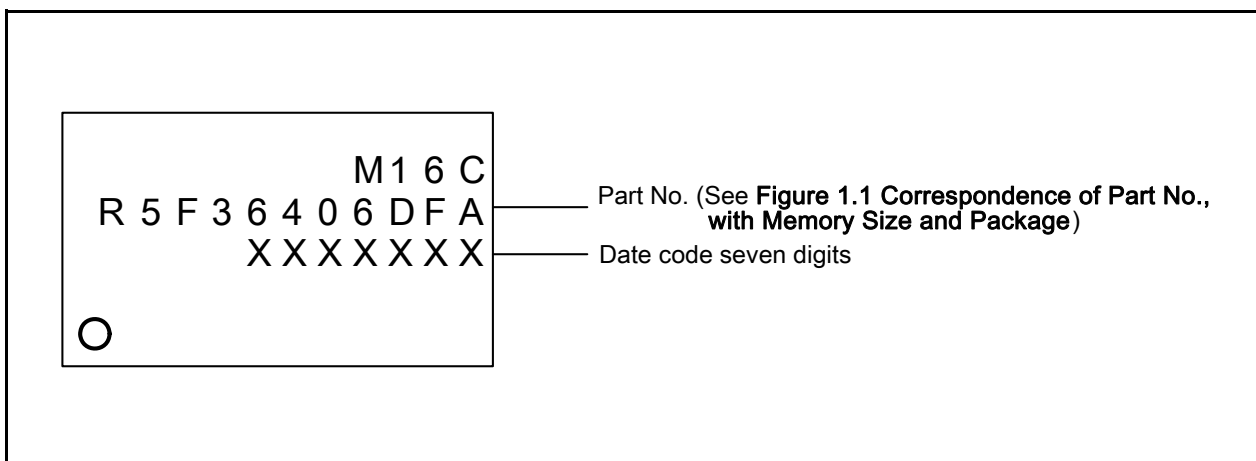


Figure 1.2 Marking Diagram of Flash Memory Version (Top View)

### 1.4 Block Diagram

Figure 1.3 is a M16C/64 Group Block Diagram.

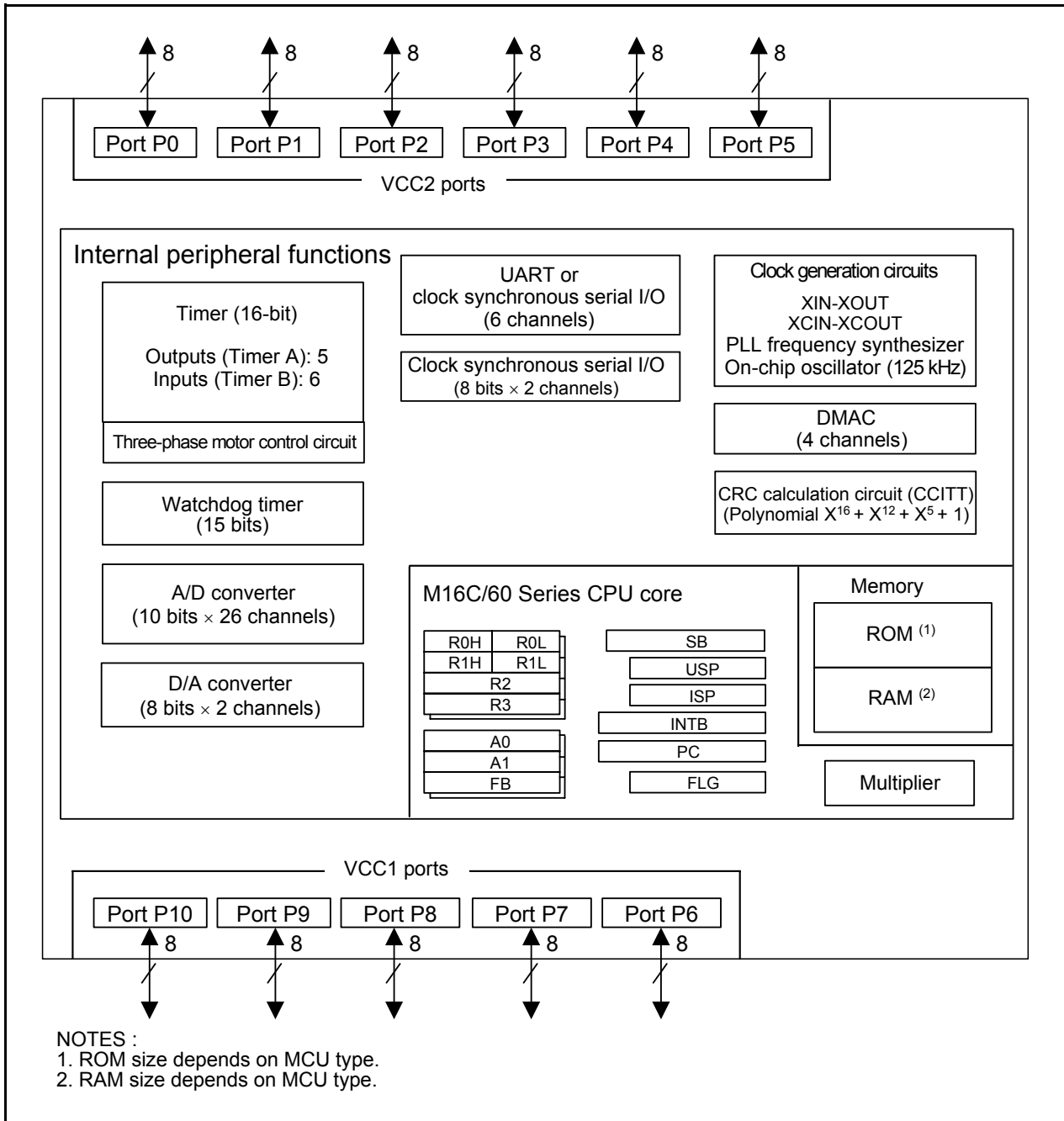


Figure 1.3 Block Diagram

### 1.5 Pin Assignments

Figures 1.4 and 1.5 show pin assignments (top view).

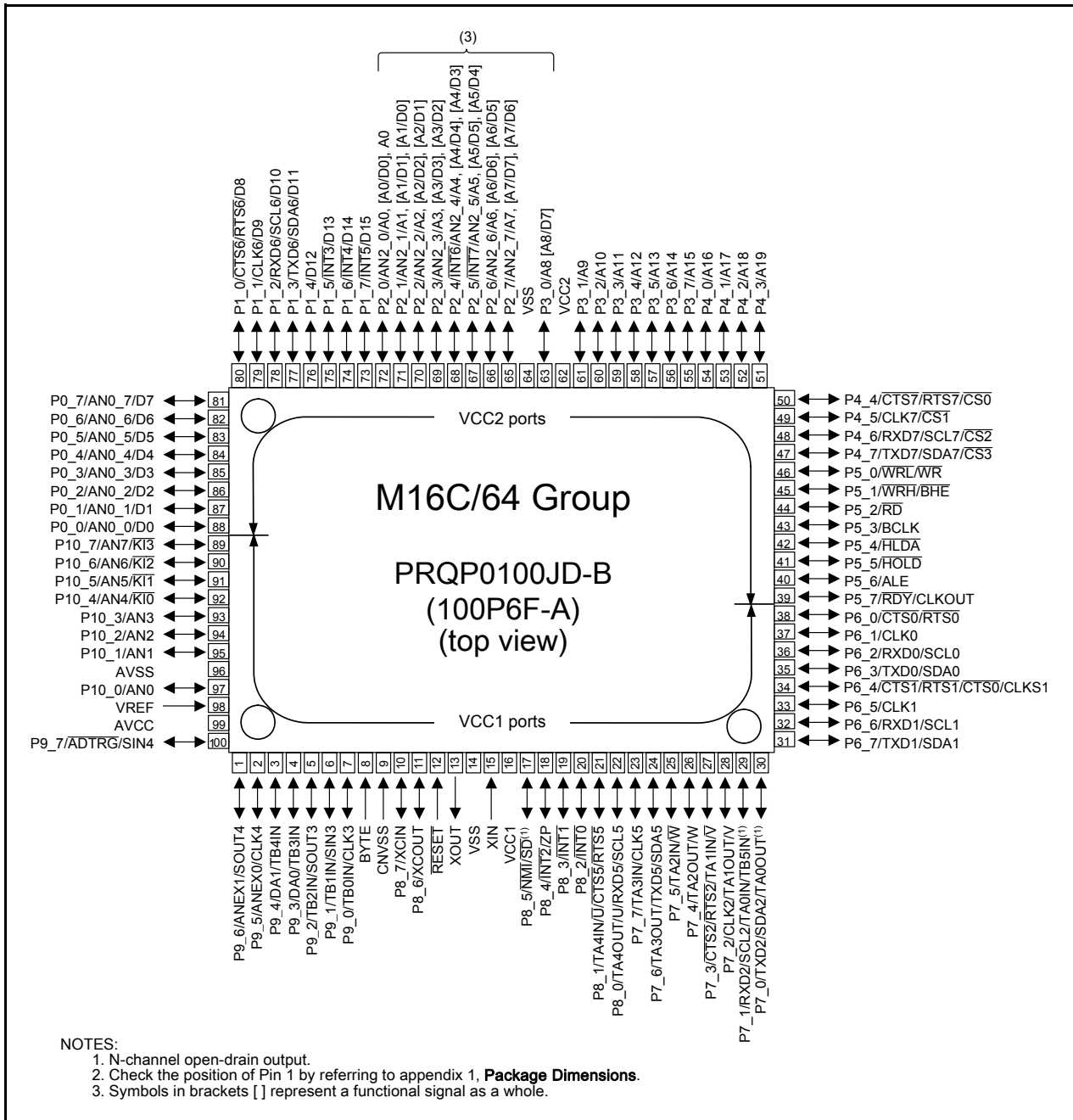


Figure 1.4 Pin Assignment (Top View)

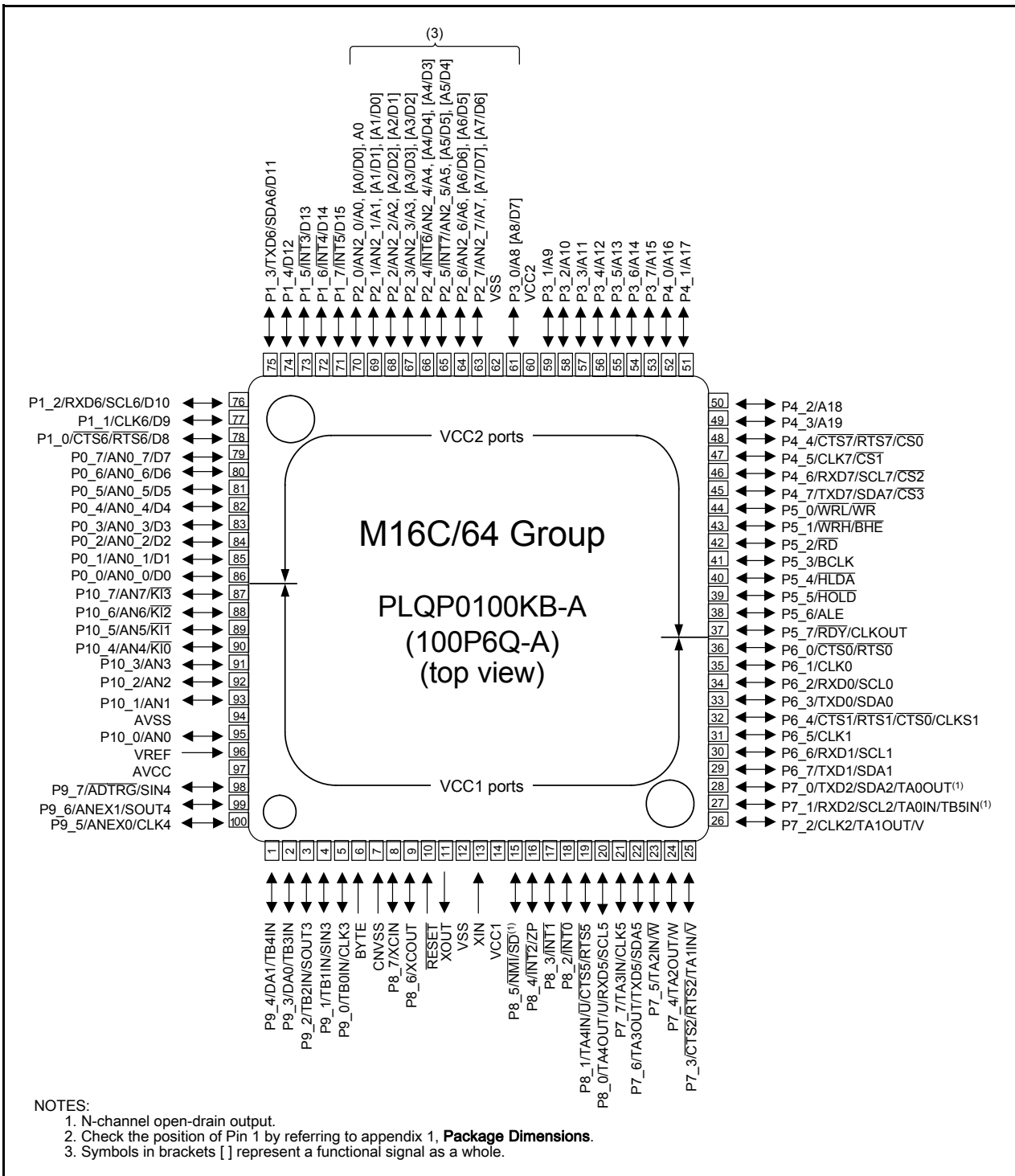


Figure 1.5 Pin Assignment (Top View)

**Table 1.4 Pin Names, for 100-Pin Package(1)**

Pin No.		Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin	Bus Control Pin
FA	FB							
1	99		P9_6			SOUT4	ANEX1	
2	100		P9_5			CLK4	ANEX0	
3	1		P9_4		TB4IN		DA1	
4	2		P9_3		TB3IN		DA0	
5	3		P9_2		TB2IN	SOUT3		
6	4		P9_1		TB1IN	SIN3		
7	5		P9_0		TB0IN	CLK3		
8	6	BYTE						
9	7	CNVSS						
10	8	XCIN	P8_7					
11	9	XCOUT	P8_6					
12	10	RESET						
13	11	XOUT						
14	12	VSS						
15	13	XIN						
16	14	VCC1						
17	15		P8_5	NMI	SD			
18	16		P8_4	INT2	ZP			
19	17		P8_3	INT1				
20	18		P8_2	INT0				
21	19		P8_1		TA4IN/U	CTS5/RTS5		
22	20		P8_0		TA4OUT/U	RXD5/SCL5		
23	21		P7_7		TA3IN	CLK5		
24	22		P7_6		TA3OUT	TXD5/SDA5		
25	23		P7_5		TA2IN/W			
26	24		P7_4		TA2OUT/W			
27	25		P7_3		TA1IN/V	CTS2/RTS2		
28	26		P7_2		TA1OUT/V	CLK2		
29	27		P7_1		TA0IN/TB5IN	RXD2/SCL2		
30	28		P7_0		TA0OUT	TXD2/SDA2		
31	29		P6_7			TXD1/SDA1		
32	30		P6_6			RXD1/SCL1		
33	31		P6_5			CLK1		
34	32		P6_4			CTS1/RTS1/CTS0/ CLKS1		
35	33		P6_3			TXD0/SDA0		
36	34		P6_2			RXD0/SCL0		
37	35		P6_1			CLK0		
38	36		P6_0			CTS0/RTS0		
39	37		P5_7					RDY/CLKOUT
40	38		P5_6					ALE
41	39		P5_5					HOLD
42	40		P5_4					HLDA
43	41		P5_3					BCLK
44	42		P5_2					RD
45	43		P5_1					WRH/BHE
46	44		P5_0					WRL/WR
47	45		P4_7			TXD7/SDA7		CS3
48	46		P4_6			RXD7/SCL7		CS2
49	47		P4_5			CLK7		CS1
50	48		P4_4			CTS7/RTS7		CS0

**Table 1.5 Pin Names for 100-Pin Package(2)**

Pin No.		Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin	Bus Control Pin
FA	FB							
51	49		P4_3					A19
52	50		P4_2					A18
53	51		P4_1					A17
54	52		P4_0					A16
55	53		P3_7					A15
56	54		P3_6					A14
57	55		P3_5					A13
58	56		P3_4					A12
59	57		P3_3					A11
60	58		P3_2					A10
61	59		P3_1					A9
62	60	VCC2						
63	61		P3_0					A8, [A8/D7]
64	62	VSS						
65	63		P2_7				AN2_7	A7, [A7/D7], [A7/D6]
66	64		P2_6				AN2_6	A6, [A6/D6], [A6/D5]
67	65		P2_5	INT7			AN2_5	A5, [A5/D5], [A5/D4]
68	66		P2_4	INT6			AN2_4	A4, [A4/D4], [A4/D3]
69	67		P2_3				AN2_3	A3, [A3/D3], [A3/D2]
70	68		P2_2				AN2_2	A2, [A2/D2], [A2/D1]
71	69		P2_1				AN2_1	A1, [A1/D1], [A1/D0]
72	70		P2_0				AN2_0	A0, [A0/D0], A0
73	71		P1_7	INT5				D15
74	72		P1_6	INT4				D14
75	73		P1_5	INT3				D13
76	74		P1_4					D12
77	75		P1_3			TXD6/SDA6		D11
78	76		P1_2			RXD6/SCL6		D10
79	77		P1_1			CLK6		D9
80	78		P1_0			CTS6/RTS6		D8
81	79		P0_7				AN0_7	D7
82	80		P0_6				AN0_6	D6
83	81		P0_5				AN0_5	D5
84	82		P0_4				AN0_4	D4
85	83		P0_3				AN0_3	D3
86	84		P0_2				AN0_2	D2
87	85		P0_1				AN0_1	D1
88	86		P0_0				AN0_0	D0
89	87		P10_7	KI3			AN7	
90	88		P10_6	KI2			AN6	
91	89		P10_5	KI1			AN5	
92	90		P10_4	KI0			AN4	
93	91		P10_3				AN3	
94	92		P10_2				AN2	
95	93		P10_1				AN1	
96	94	AVSS						
97	95		P10_0				AN0	
98	96	VREF						
99	97	AVCC						
100	98		P9_7			SIN4	ADTRG	

## 1.6 Pin Functions

**Table 1.6 Pin Functions (1)**

Signal Name	Pin Name	I/O	Power Supply	Description
Power supply input	VCC1 VCC2 VSS	I	-	Apply 2.7 to 5.5 V to pins VCC1 and VCC2 (VCC1 =VCC2) and 0 V to the VSS pin (1)
Analog power supply input	AVCC AVSS	I	VCC1	Apply the power supply for the A/D converter. Connect the AVCC pin to VCC1. Connect the AVSS pin to VSS.
Reset input	RESET	I	VCC1	Low active input pin. Driving this pin Low resets the MCU.
CNVSS	CNVSS	I	VCC1	Input pin to switch processor mode. To start up in single-chip mode after a reset, connect the CNVSS pin to VSS via resistor. To start up in microprocessor mode, connect the CNVSS pin to VSS1.
External data bus width select input	BYTE	I	VCC1	Input pin to select the data bus of the external memory area. The data bus is 16-bit when it is Low and 8-bit when it is High. This pin must be fixed either High or Low. Connect the BYTE pin to VSS in single-chip mode
Bus control pins	D0 to D7	I/O	VCC2	Inputs or outputs data (D0 to D7) while accessing an external memory area with separate bus
	D8 to D15	I/O	VCC2	Inputs or outputs data (D8 to D15) while accessing an external memory area with 16-bit separate bus
	A0 to A19	O	VCC2	Outputs address bits A0 to A19
	A0/D0 to A7/D7	I/O	VCC2	Inputs or outputs data (D0 to D7) and outputs address bits (A0 to A7) by timesharing, while accessing an external memory area with 8-bit multiplexed bus
	A1/D0 to A8/D7	I/O	VCC2	Inputs or outputs data (D0 to D7) and outputs address bits (A1 to A8) by timesharing, while accessing an external memory area with 16-bit multiplexed bus
	CS0 to CS3	O	VCC2	Outputs chip-select signals CS0 to CS3 to specify an external memory area
	WRL/WR WRH/BHE RD	O	VCC2	Low active output pins. Outputs WRL, WRH, (WR, BHE), RD signals. WRL and WRH can be switched with or BHE and WR can be selected by a program. WRL, WRH and RD selected If the external data bus is 16-bit, data is written to an even address in external memory area when WRL is driven low. Data is written to an odd address when WRH is driven low. Data is read when RD is driven low. WR, BHE and RD are selected Data is written to external memory area when WR is driven low. Data in external memory area is read when RD is driven low. An odd address is accessed when BHE is driven low. Select WR, BHE, and RD for external 8-bit data bus
	ALE	O	VCC2	Output ALE signal to latch address.
	HOLD	I	VCC2	Low active input pin. The MCU is placed in hold state while the HOLD pin is driven low.
	HLDA	O	VCC2	Low active output pin. In a hold state, HLDA outputs a low-level signal.
RDY	I	VCC2	Low active input pin. The MCU is placed in wait state while the RDY pin is driven low.	

Power supply: VCC2 is used to supply power to external bus related pins.

NOTE:

1. VCC1 is hereinafter referred to as VCC unless otherwise noted.



**Table 1.7 Pin Functions (2)**

Signal Name	Pin Name	I/O	Power Supply	Description
Main clock input	XIN	I	VCC1	I/O pins for the main clock oscillation circuit. Connect a ceramic resonator or crystal oscillator between XIN and XOUT <sup>(1)</sup> . To apply an external clock, connect it to XIN and leave XOUT open.
Main clock output	XOUT	O	VCC1	
Sub clock input	XCIN	I	VCC1	I/O pins for a sub clock oscillation circuit. Connect a crystal oscillator between XCIN and XCOUT <sup>(1)</sup> . To apply an external clock, connect it to XCIN and leave XCOUT open
Sub clock output	XCOUT	O	VCC1	
BCLK output	BCLK	O	VCC2	Output pin for BCLK signal
Clock output	CLKOUT	O	VCC2	This pin outputs the clock having the same frequency as f <sub>C</sub> , f <sub>8</sub> , or f <sub>32</sub>
INT interrupt input	INT0 to INT2	I	VCC1	Low active input pins for INT interrupt
	INT3 to INT7	I	VCC2	
NMI interrupt input	NMI	I	VCC1	Low active input pin for NMI interrupt
Key input interrupt input	KI0 to KI3	I	VCC1	Low active input pins for key input interrupt
Timer A	TA0OUT to TA4OUT	I/O	VCC1	Timer A0 to A4 I/O pins (TA0OUT as an output pin is N-channel open drain output)
	TA0IN to TA4IN	I	VCC1	Timer A0 to A4 input pins
	ZP	I	VCC1	Input pin for Z-phase
Timer B	TB0IN to TB5IN	I	VCC1	Timer B0 to B5 input pins
Three-phase motor control timer	U, $\bar{U}$ , V, $\bar{V}$ , W, $\bar{W}$	O	VCC1	Output pins for three-phase motor control timer output
	SD	I	VCC1	Input pin for three-phase motor control timer input
Serial interface UART0 to UART2, UART5 to UART7	CTS0 to CTS2, CTS5	I	VCC1	Input pins to control data transmission
	CTS6, CTS7	I	VCC2	
	RTS0 to RTS2, RTS5	O	VCC1	Output pins to control data reception
	RTS6, RTS7	O	VCC2	
	CLK0 to CLK2, CLK5	I/O	VCC1	Transfer clock I/O pins
	CLK6, CLK7	I/O	VCC2	
	RXD0 to RXD2, RXD5	I	VCC1	Serial data input pins
	RXD6, RXD7	I	VCC2	
	TXD0 to TXD2, TXD5	O	VCC1	Serial data output pins <sup>(2)</sup>
	TXD6, TXD7	O	VCC2	
	CLKS1	O	VCC1	Output pin for transfer clock multiple-pin output function

## NOTES:

1. Consult the oscillator manufacturer regarding the oscillation characteristics.
2. TXD2, SDA2, and SCL2 are N-channel open-drain output pins. TXDi (i = 0, 1, 5 to 7), SDAi, and SCLi can be selected as CMOS output pins or N-channel open-drain output pins by a program.

**Table 1.8 Pin Functions (3)**

Signal Name	Pin Name	I/O	Power Supply	Description
UART0 to UART2, UART5 to UART7 I <sup>2</sup> C mode	SDA0 to SDA2, SDA5	I/O	VCC1	Serial data I/O pins <sup>(1)</sup>
	SDA6, SDA7	I/O	VCC2	
	SCL0 to SCL2, SCL5	I/O	VCC1	Transfer clock I/O pins <sup>(1)</sup>
	SCL6, SCL7	I/O	VCC2	
Serial interface SI/03, SI/04	CLKS3, CLKS4	I/O	VCC1	Transfer clock I/O pins
	SIN3, SIN4	I	VCC1	Serial data input pins
	SOUT3, SOUT4	O	VCC1	Serial data output pins
Reference voltage input	VREF	I	VCC1	Reference voltage input pins for the A/D converter and D/A converter. Connect to VCC1.
A/D converter	AN0 to AN7	I	VCC1	Analog input pins for the A/D converter
	AN0_0 to AN0_7 AN2_0 to AN2_7	I	VCC2	
	ADTRG	I	VCC1	Input pin for an external A/D trigger
	ANEX0, ANEX1	I	VCC1	Extended analog input pin for the A/D converter
D/A converter	DA0, DA1	O	VCC1	Output pin for the D/A converter
I/O port	P0_0 to P0_7 P1_0 to P1_7 P2_0 to P2_7 P3_0 to P3_7 P4_0 to P4_7 P5_0 to P5_7	I/O	VCC2	8-bit CMOS I/O ports. A direction register determines whether each pin is used as an input port or an output port. A pull-up resistor may be enabled or disabled for input ports in 4-bit units.
	P6_0 to P6_7 P7_0 to P7_7 P8_0 to P8_7 P9_0 to P9_7 P10_0 to P10_7		VCC1	8-bit I/O ports having equivalent functions to P0. However, P7_0, P7_1, and P8_5 are N-channel open-drain output ports. No pull-up resistor is provided. P8_5 is an input port for verifying the NMI pin level and shares a pin with NMI.

NOTE:

1. TXD2, SDA2, and SCL2 are N-channel open drain output pins. TXDi (i = 0, 1, 5 to 7), SDAi, SCLi can be selected as CMOS output pin or N-channel open drain output pin by program.

## 2. Central Processing Unit (CPU)

Figure 2.1 shows the CPU registers. Seven registers (R0, R1, R2, R3, A0, A1, and FB) out of thirteen registers configure a register bank. There are two sets of register banks.

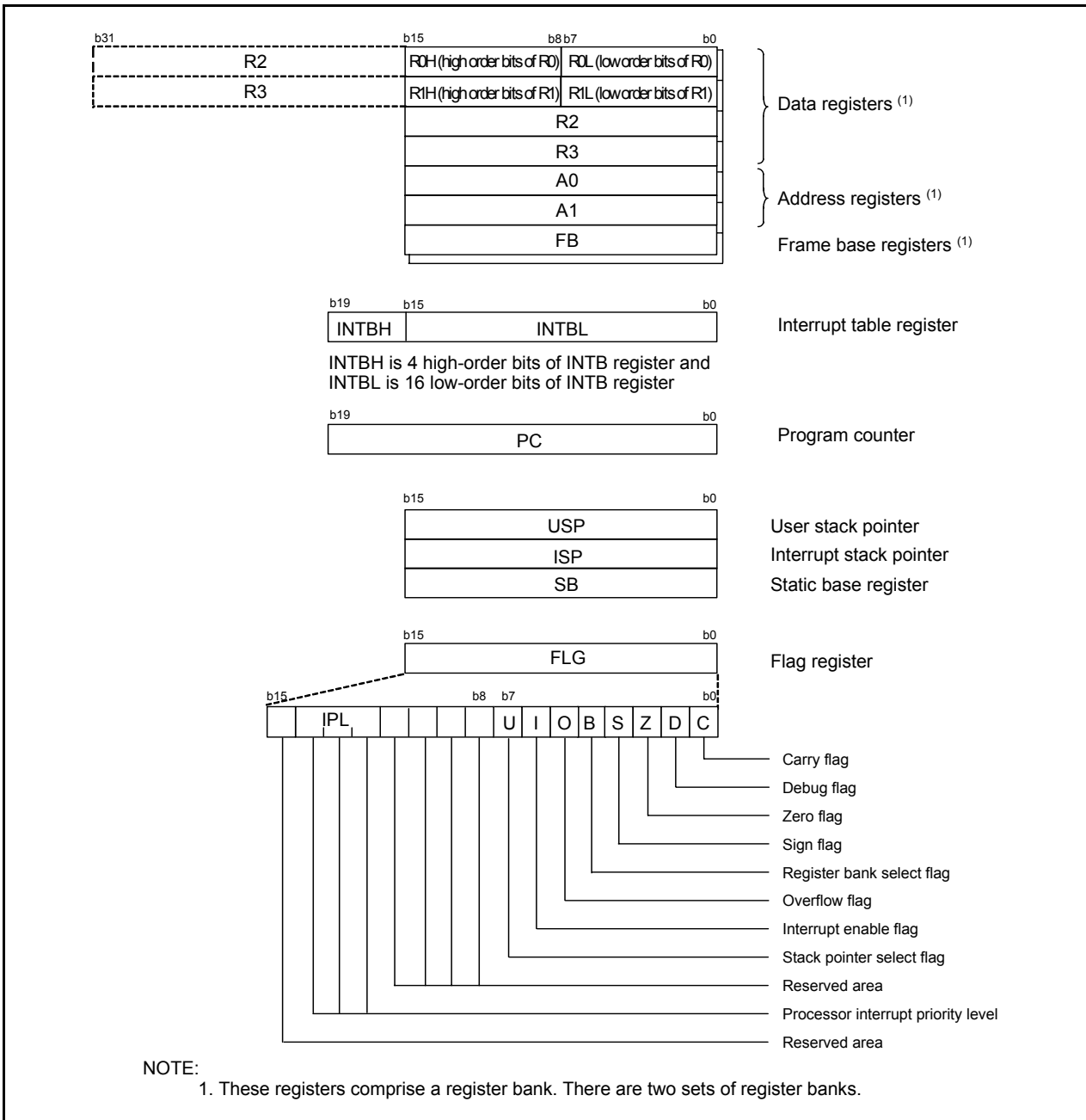


Figure 2.1 Central Processing Unit Register

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

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

## 2.2 Address Registers (A0 and A1)

A0 and A1 are 16-bit registers used for A0-/A1-indirect addressing, A0-/A1-relative addressing, transfer, arithmetic and logic operations. A0 can be combined with A1 and used as a 32-bit address register (A1A0).

## 2.3 Frame Base Registers (FB)

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

## 2.4 Interrupt Table Register (INTB)

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

## 2.5 Program Counter (PC)

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

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

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

## 2.7 Static Base Register (SB)

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

## 2.8 Flag Register (FLG)

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

### 2.8.1 Carry Flag (C Flag)

The C flag retains a carry, borrow, or shift-out bit that has been generated by the arithmetic/logic unit.

### 2.8.2 Debug Flag (D Flag)

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

### 2.8.3 Zero Flag (Z Flag)

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

### 2.8.4 Sign Flag (S Flag)

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

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

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

### 2.8.6 Overflow Flag (O Flag)

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

### 2.8.7 Interrupt Enable Flag (I Flag)

The I flag enables maskable interrupts.

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

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

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

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

### 2.8.9 Processor Interrupt Priority Level (IPL)

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

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

### 2.8.10 Reserved Space

Only write 0 to bits assigned as reserved bits. The read value is undefined.

### 3. Memory

Figure 3.1 is a memory map of the M16C/64 Group. The M16C/64 Group has 1 Mbyte address space from address 00000h to FFFFFh.

The internal ROM is flash memory. Program ROM 1 is allocated from address FFFFFh to lower.

For example, a 64-Kbyte program ROM 1 is addressed from F0000h to FFFFFh. An 8-Kbyte data flash is addressed from 0E000h to 0FFFFh. This data flash space is used not only for data storage but also for program storage. Program ROM 2 is allocated addresses 10000h to 13FFFh. The user boot code area is assigned addresses 13FF0h to 13FFFh in the program ROM 2.

The fixed interrupt vectors are addressed from FFFDCh to FFFFFh. They store the starting address of each interrupt routine.

The internal RAM is allocated from address 00400h to higher. For example, a 10-Kbyte internal RAM is addressed from 00400h to 02BFFh. The internal RAM is used not only for data storage but also for stack area when subroutines are called or when interrupt request are acknowledged.

SFRs are allocated from address 00000h to 003FFh and from 0D000h to 0D7FFh. Peripheral function control registers are located here. All blank spaces within SFRs are reserved and cannot be accessed by users. The special page vectors are addressed from FFE00h to FFFD7h. They are used for the JMPS instruction and JSRS instruction. Refer to the **M16C/60, M16C/20 Series Software Manual** for details.

In memory expansion mode or microprocessor mode, some spaces are reserved and cannot be accessed by users.

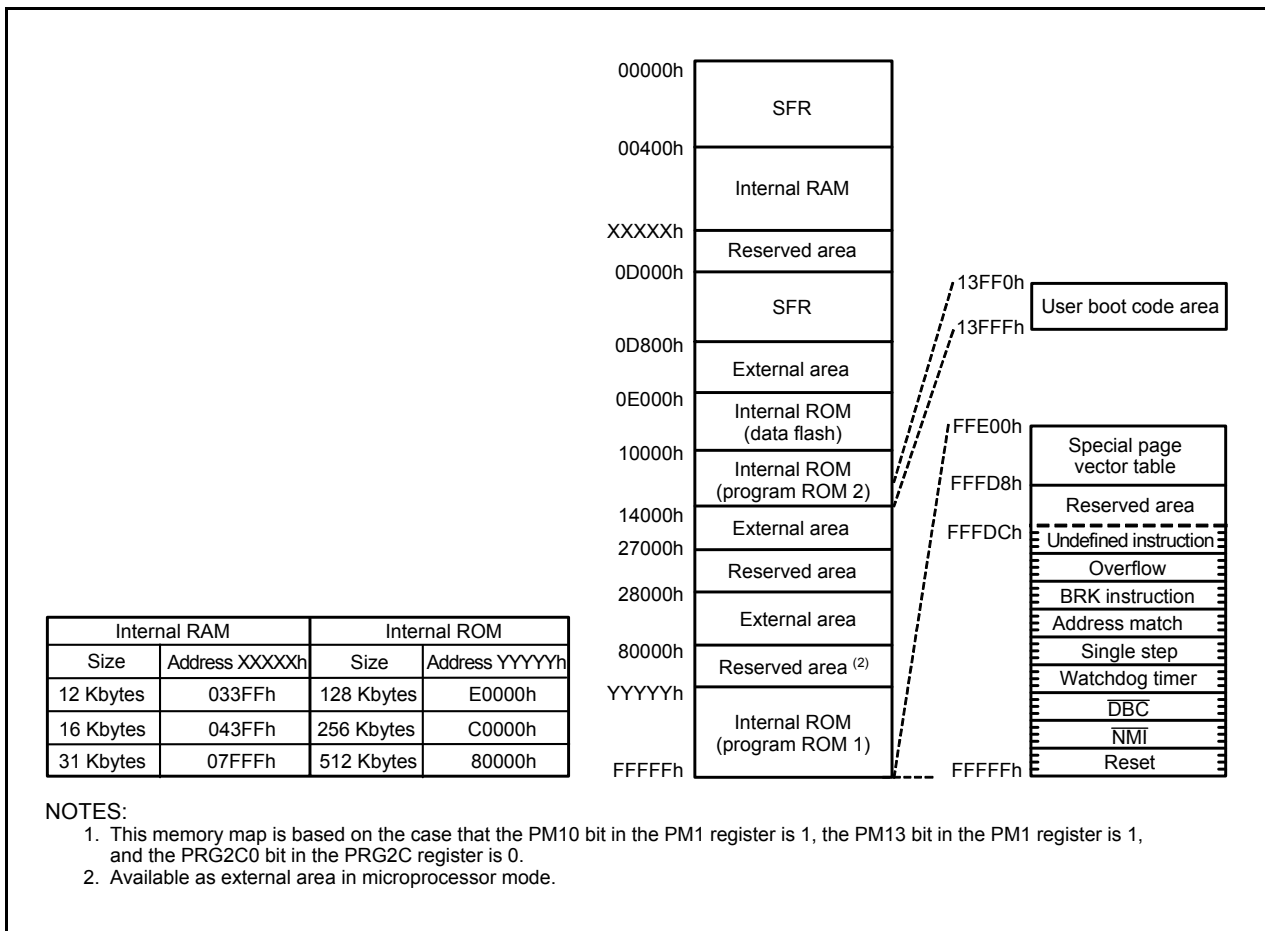


Figure 3.1 Memory Map

## 4. Special Function Registers (SFRs)

An SFR (Special Function Register) is a control register for a peripheral function. Tables 4.1 to 4.14 list SFR information.

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

Address	Register	Symbol	After Reset
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	0000000b (CNVSS pin is "L") 00000011b(CNVSS pin is "H") (2)
0005h	Processor Mode Register 1	PM1	00001000b
0006h	System Clock Control Register 0	CM0	01001000b
0007h	System Clock Control Register 1	CM1	00100000b
0008h	Chip Select Control Register	CSR	00000001b
0009h			
000Ah	Protect Register	PRCR	00h
000Bh	Data Bank Register	DBR	00h
000Ch	Oscillation Stop Detection Register	CM2	0X000010b (3)
000Dh			
000Eh			
000Fh			
0010h	Program 2 Area Control Register	PRG2C	XXXXXXXX0b
0011h			
0012h	Peripheral Clock Select Register	PCLKR	00000011b
0013h			
0014h			
0015h	Clock Prescaler Reset Flag	CPSRF	0XXXXXXXXb
0016h			
0017h			
0018h	Reset Source Determine Flag	RSTFR	0XXXXXXXXb (4)
0019h	Voltage Detection 2 Circuit Flag Register	VCR1	00001000b (2)
001Ah	Voltage Detection Circuit Operation Enable Register	VCR2	000X0000b (Hardware reset 1) 001X0000b (Brown-out reset) (2)
001Bh	Chip Select Expansion Control Register	CSE	00h
001Ch	PLL Control Register 0	PLC0	0X01X010b
001Dh			
001Eh	Processor Mode Register 2	PM2	XXX00X01b (2)
001Fh	Low Voltage Detection Interrupt Register	D4INT	00h
0020h			
0021h			
0022h			
0023h			
0024h			
0025h			
0026h			
0027h			
0028h			
0029h			
002Ah	Voltage Monitor 0 Circuit Control Register	VW0C	10001X10b (Hardware Reset 1) 11001X11b (Brown-out reset) (2)
002Bh			
002Ch			
002Dh			
002Eh			
002Fh			

NOTES:

X: Undefined

- The blank areas are reserved and cannot be accessed by users.
- Software reset, watchdog timer reset, and oscillation stop detection reset do not affect bits PM01 and PM00 in the PM0 register, and registers VCR1, VCR2, and VW0C.
- Oscillation stop detection reset do not affect bits CM20, CM21, and CM27.
- The CWR bit in the RSTFR register is set to 0 after brown-out reset. This bit does not change by any other reset.

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

Address	Register	Symbol	After Reset
0030h			
0031h			
0032h			
0033h			
0034h			
0035h			
0036h			
0037h			
0038h			
0039h			
003Ah			
003Bh			
003Ch			
003Dh			
003Eh			
003Fh			
0040h			
0041h			
0042h	INT7 Interrupt Control Register	INT7IC	XX00X000b
0043h	INT6 Interrupt Control Register	INT6IC	XX00X000b
0044h	INT3 Interrupt Control Register	INT3IC	XX00X000b
0045h	Timer B5 Interrupt Control Register	TB5IC	XXXXX000b
0046h	Timer B4 Interrupt Control Register, UART1 BUS Collision Detection Interrupt Control Register	TB4IC, U1BCNIC	XXXXX000b
0047h	Timer B3 Interrupt Control Register, UART0 BUS Collision Detection Interrupt Control Register	TB3IC, U0BCNIC	XXXXX000b
0048h	SI/O4 Interrupt Control Register, INT5 Interrupt Control Register	S4IC, INT5IC	XX00X000b
0049h	SI/O3 Interrupt Control Register, INT4 Interrupt Control Register	S3IC, INT4IC	XX00X000b
004Ah	UART2 BUS Collision Detection Interrupt Control Register	BCNIC	XXXXX000b
004Bh	DMA0 Interrupt Control Register	DM0IC	XXXXX000b
004Ch	DMA1 Interrupt Control Register	DM1IC	XXXXX000b
004Dh	Key Input Interrupt Control Register	KUPIC	XXXXX000b
004Eh	A/D Conversion Interrupt Control Register	ADIC	XXXXX000b
004Fh	UART2 Transmit Interrupt Control Register	S2TIC	XXXXX000b
0050h	UART2 Receive Interrupt Control Register	S2RIC	XXXXX000b
0051h	UART0 Transmit Interrupt Control Register	S0TIC	XXXXX000b
0052h	UART0 Receive Interrupt Control Register	S0RIC	XXXXX000b
0053h	UART1 Transmit Interrupt Control Register	S1TIC	XXXXX000b
0054h	UART1 Receive Interrupt Control Register	S1RIC	XXXXX000b
0055h	Timer A0 Interrupt Control Register	TA0IC	XXXXX000b
0056h	Timer A1 Interrupt Control Register	TA1IC	XXXXX000b
0057h	Timer A2 Interrupt Control Register	TA2IC	XXXXX000b
0058h	Timer A3 Interrupt Control Register	TA3IC	XXXXX000b
0059h	Timer A4 Interrupt Control Register	TA4IC	XXXXX000b
005Ah	Timer B0 Interrupt Control Register	TB0IC	XXXXX000b
005Bh	Timer B1 Interrupt Control Register	TB1IC	XXXXX000b
005Ch	Timer B2 Interrupt Control Register	TB2IC	XXXXX000b
005Dh	INT0 Interrupt Control Register	INT0IC	XX00X000b
005Eh	INT1 Interrupt Control Register	INT1IC	XX00X000b
005Fh	INT2 Interrupt Control Register	INT2IC	XX00X000b

NOTE:

X: Undefined

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



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

Address	Register	Symbol	After Reset
0060h			
0061h			
0062h			
0063h			
0064h			
0065h			
0066h			
0067h			
0068h			
0069h	DMA2 Interrupt Control Register	DM2IC	XXXXX000b
006Ah	DMA3 Interrupt Control Register	DM3IC	XXXXX000b
006Bh	UART5 BUS Collision Detection Interrupt Control Register	U5BCNIC	XXXXX000b
006Ch	UART5 Transmit Interrupt Control Register	S5TIC	XXXXX000b
006Dh	UART5 Receive Interrupt Control Register	S5RIC	XXXXX000b
006Eh	UART6 BUS Collision Detection Interrupt Control Register	U6BCNIC	XXXXX000b
006Fh	UART6 Transmit Interrupt Control Register	S6TIC	XXXXX000b
0070h	UART6 Receive Interrupt Control Register	S6RIC	XXXXX000b
0071h	UART7 BUS Collision Detection Interrupt Control Register	U7BCNIC	XXXXX000b
0072h	UART7 Transmit Interrupt Control Register	S7TIC	XXXXX000b
0073h	UART7 Receive Interrupt Control Register	S7RIC	XXXXX000b
0074h			
0075h			
0076h			
0077h			
0078h			
0079h			
007Ah			
007Bh			
007Ch			
007Dh			
007Eh			
007Fh			
0080h			
0081h			
0082h			
0083h			
0084h			
0085h			
0086h			
0087h			
0088h			
0089h			
008Ah			
008Bh			
008Ch			
008Dh			
008Eh			
008Fh			
0090h			
0091h			
0092h			
0093h			
0094h			
0095h			
0096h			
0097h			
0098h			
0099h			
009Ah			
009Bh			
009Ch			
009Dh			
009Eh			
009Fh to 015Fh			

NOTE:

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

X: Undefined

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

Address	Register	Symbol	After Reset
0160h			
0161h			
0162h			
0163h			
0164h			
0165h			
0166h			
0167h			
0168h			
0169h			
016Ah			
016Bh			
016Ch			
016Dh			
016Eh			
016Fh			
0170h			
0171h			
0172h			
0173h			
0174h			
0175h			
0176h			
0177h			
0178h			
0179h			
017Ah			
017Bh			
017Ch			
017Dh			
017Eh			
017Fh			
0180h	DMA0 Source Pointer	SAR0	XXh
0181h			XXh
0182h			0Xh
0183h			
0184h	DMA0 Destination Pointer	DAR0	XXh
0185h			XXh
0186h			0Xh
0187h			
0188h	DMA0 Transfer Counter	TCR0	XXh
0189h			XXh
018Ah			
018Bh			
018Ch	DMA0 Control Register	DM0CON	00000X00b
018Dh			
018Eh			
018Fh			
0190h	DMA1 Source Pointer	SAR1	XXh
0191h			XXh
0192h			0Xh
0193h			
0194h	DMA1 Destination Pointer	DAR1	XXh
0195h			XXh
0196h			0Xh
0197h			
0198h	DMA1 Transfer Counter	TCR1	XXh
0199h			XXh
019Ah			
019Bh			
019Ch	DMA1 Control Register	DM1CON	00000X00b
019Dh			
019Eh			
019Fh			

NOTE:

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

X: Undefined

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

Address	Register	Symbol	After Reset
01A0h 01A1h 01A2h	DMA2 Source Pointer	SAR2	XXh XXh 0Xh
01A3h			
01A4h 01A5h 01A6h	DMA2 Destination Pointer	DAR2	XXh XXh 0Xh
01A7h			
01A8h 01A9h	DMA2 Transfer Counter	TCR2	XXh XXh
01AAh			
01ABh			
01ACh	DMA2 Control Register	DM2CON	00000X00b
01ADh			
01AEh			
01AFh			
01B0h 01B1h 01B2h	DMA3 Source Pointer	SAR3	XXh XXh 0Xh
01B3h			
01B4h 01B5h 01B6h	DMA3 Destination Pointer	DAR3	XXh XXh 0Xh
01B7h			
01B8h 01B9h	DMA3 Transfer Counter	TCR3	XXh XXh
01BAh			
01BBh			
01BCh	DMA3 Control Register	DM3CON	00000X00b
01BDh			
01BEh			
01BFh			
01C0h			
01C1h			
01C2h			
01C3h			
01C4h			
01C5h			
01C6h			
01C7h			
01C8h	Timer B Count Source Select Register 0	TBCS0	00h
01C9h	Timer B Count Source Select Register 1	TBCS1	X0h
01CAh			
01CBh			
01CCh			
01CDh			
01CEh			
01CFh			
01D0h	Timer A Count Source Select Register 0	TACS0	00h
01D1h	Timer A Count Source Select Register 1	TACS1	00h
01D2h	Timer A Count Source Select Register 2	TACS2	X0h
01D3h			
01D4h			
01D5h	Timer A Waveform Output Function Select Register	TAPOFS	XXX00000b
01D6h			
01D7h			
01D8h			
01D9h			
01DAh			
01DBh			
01DCh			
01DDh			
01DEh			
01DFh			

NOTE:

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

X: Undefined

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

Address	Register	Symbol	After Reset
01E0h			
01E1h			
01E2h			
01E3h			
01E4h			
01E5h			
01E6h			
01E7h			
01E8h	Timer B Count Source Select Register 2	TBCS2	00h
01E9h	Timer B Count Source Select Register 3	TBCS3	X0h
01EAh			
01EBh			
01ECh			
01EDh			
01EEh			
01EFh			
01F0h			
01F1h			
01F2h			
01F3h			
01F4h			
01F5h			
01F6h			
01F7h			
01F8h			
01F9h			
01FAh			
01FBh			
01FCh			
01FDh			
01FEh			
01FFh			
0200h			
0201h			
0202h			
0203h			
0204h			
0205h	Interrupt Source Select Register 3	IFSR3A	00h
0206h	Interrupt Source Select Register 2	IFSR2A	00h
0207h	Interrupt Source Select Register	IFSR	00h
0208h			
0209h			
020Ah			
020Bh			
020Ch			
020Dh			
020Eh	Address Match Interrupt Enable Register	AIER	XXXXXX00b
020Fh	Address Match Interrupt Enable Register 2	AIER2	XXXXXX00b
0210h	Address Match Interrupt Register 0	RMAD0	00h
0211h			00h
0212h			X0h
0213h			
0214h	Address Match Interrupt Register 1	RMAD1	00h
0215h			00h
0216h			X0h
0217h			
0218h	Address Match Interrupt Register 2	RMAD2	00h
0219h			00h
021Ah			X0h
021Bh			
021Ch	Address Match Interrupt Register 3	RMAD3	00h
021Dh			00h
021Eh			X0h
021Fh			

NOTE:

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

X: Undefined

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

Address	Register	Symbol	After Reset
0220h	Flash Memory Control Register 0	FMR0	0000001b
0221h	Flash Memory Control Register 1	FMR1	00X0XX0Xb
0222h	Flash Memory Control Register 2	FMR2	XXXX0000b
0223h			
0224h			
0225h			
0226h			
0227h			
0228h			
0229h			
022Ah			
022Bh			
022Ch			
022Dh			
022Eh			
022Fh			
0230h	Flash Memory Control Register 6	FMR6	XX0XX00b
0231h			
0232h			
0233h			
0234h			
0235h			
0236h			
0237h			
0238h			
0239h			
023Ah			
023Bh			
023Ch			
023Dh			
023Eh			
023Fh			
0240h			
0241h			
0242h			
0243h			
0244h	UART0 Special Mode Register 4	U0SMR4	00h
0245h	UART0 Special Mode Register 3	U0SMR3	000X0X0Xb
0246h	UART0 Special Mode Register 2	U0SMR2	X0000000b
0247h	UART0 Special Mode Register	U0SMR	X0000000b
0248h	UART0 Transmit/Receive Mode Register	U0MR	00h
0249h	UART0 Bit Rate Register	U0BRG	XXh
024Ah	UART0 Transmit Buffer Register	U0TB	XXh
024Bh			XXh
024Ch	UART0 Transmit/Receive Control Register 0	U0C0	00001000b
024Dh	UART0 Transmit/Receive Control Register 1	U0C1	00XX0010b
024Eh	UART0 Receive Buffer Register	U0RB	XXh
024Fh			XXh
0250h	UART Transmit/Receive Control Register 2	UCON	X0000000b
0251h			
0252h			
0253h			
0254h	UART1 Special Mode Register 4	U1SMR4	00h
0255h	UART1 Special Mode Register 3	U1SMR3	000X0X0Xb
0256h	UART1 Special Mode Register 2	U1SMR2	X0000000b
0257h	UART1 Special Mode Register	U1SMR	X0000000b
0258h	UART1 Transmit/Receive Mode Register	U1MR	00h
0259h	UART1 Bit Rate Register	U1BRG	XXh
025Ah	UART1 Transmit Buffer Register	U1TB	XXh
025Bh			XXh
025Ch	UART1 Transmit/Receive Control Register 0	U1C0	00001000b
025Dh	UART1 Transmit/Receive Control Register 1	U1C1	00XX0010b
025Eh	UART1 Receive Buffer Register	U1RB	XXh
025Fh			XXh

NOTE:

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

X: Undefined

**Table 4.8 SFR Information (8) (1)**

Address	Register	Symbol	After Reset
0260h			
0261h			
0262h			
0263h			
0264h	UART2 Special Mode Register 4	U2SMR4	00h
0265h	UART2 Special Mode Register 3	U2SMR3	000X0X0Xb
0266h	UART2 Special Mode Register 2	U2SMR2	X0000000b
0267h	UART2 Special Mode Register	U2SMR	X0000000b
0268h	UART2 Transmit/Receive Mode Register	U2MR	00h
0269h	UART2 Bit Rate Register	U2BRG	XXh
026Ah	UART2 Transmit Buffer Register	U2TB	XXh
026Bh			XXh
026Ch	UART2 Transmit/Receive Control Register 0	U2C0	00001000b
026Dh	UART2 Transmit/Receive Control Register 1	U2C1	00000010b
026Eh	UART2 Receive Buffer Register	U2RB	XXh
026Fh			XXh
0270h	SI/O3 Transmit/Receive Register	S3TRR	XXh
0271h			
0272h	SI/O3 Control Register	S3C	01000000b
0273h	SI/O3 Bit Rate Register	S3BRG	XXh
0274h	SI/O4 Transmit/Receive Register	S4TRR	XXh
0275h			
0276h	SI/O4 Control Register	S4C	01000000b
0277h	SI/O4 Bit Rate Register	S4BRG	XXh
0278h	SI/O34 Control Register 2	S34C2	00XXX0X0b
0279h			
027Ah			
027Bh			
027Ch			
027Dh			
027Eh			
027Fh			
0280h			
0281h			
0282h			
0283h			
0284h	UART5 Special Mode Register 4	U5SMR4	00h
0285h	UART5 Special Mode Register 3	U5SMR3	000X0X0Xb
0286h	UART5 Special Mode Register 2	U5SMR2	X0000000b
0287h	UART5 Special Mode Register	U5SMR	X0000000b
0288h	UART5 Transmit/Receive Mode Register	U5MR	00h
0289h	UART5 Bit Rate Register	U5BRG	XXh
028Ah	UART5 Transmit Buffer Register	U5TB	XXh
028Bh			XXh
028Ch	UART5 Transmit/Receive Control Register 0	U5C0	00001000b
028Dh	UART5 Transmit/Receive Control Register 1	U5C1	00000010b
028Eh	UART5 Receive Buffer Register	U5RB	XXh
028Fh			XXh
0290h			
0291h			
0292h			
0293h			
0294h	UART6 Special Mode Register 4	U6SMR4	00h
0295h	UART6 Special Mode Register 3	U6SMR3	000X0X0Xb
0296h	UART6 Special Mode Register 2	U6SMR2	X0000000b
0297h	UART6 Special Mode Register	U6SMR	X0000000b
0298h	UART6 Transmit/Receive Mode Register	U6MR	00h
0299h	UART6 Bit Rate Register	U6BRG	XXh
029Ah	UART6 Transmit Buffer Register	U6TB	XXh
029Bh			XXh
029Ch	UART6 Transmit/Receive Control Register 0	U6C0	00001000b
029Dh	UART6 Transmit/Receive Control Register 1	U6C1	00000010b
029Eh	UART6 Receive Buffer Register	U6RB	XXh
029Fh			XXh

NOTE: X: Undefined  
 1. The blank areas are reserved and cannot be accessed by users.

**Table 4.9 SFR Information (9) <sup>(1)</sup>**

Address	Register	Symbol	After Reset
02A0h			
02A1h			
02A2h			
02A3h			
02A4h	UART7 Special Mode Register 4	U7SMR4	00h
02A5h	UART7 Special Mode Register 3	U7SMR3	000X0X0Xb
02A6h	UART7 Special Mode Register 2	U7SMR2	X0000000b
02A7h	UART7 Special Mode Register	U7SMR	X0000000b
02A8h	UART7 Transmit/Receive Mode Register	U7MR	00h
02A9h	UART7 Bit Rate Register	U7BRG	XXh
02AAh	UART7 Transmit Buffer Register	U7TB	XXh
02ABh			XXh
02ACh	UART7 Transmit/Receive Control Register 0	U7C0	00001000b
02ADh	UART7 Transmit/Receive Control Register 1	U7C1	00000010b
02AEh	UART7 Receive Buffer Register	U7RB	XXh
02AFh			XXh
02B0h			
02B1h			
02B2h			
02B3h			
02B4h			
02B5h			
02B6h			
02B7h			
02B8h			
02B9h			
02BAh			
02BBh			
02BCh			
02BDh			
02BEh			
02BFh			
02C0h			
02C1h			
02C2h			
02C3h			
02C4h			
02C5h			
02C6h			
02C7h			
02C8h			
02C9h			
02CAh			
02CBh			
02CCh			
02CDh			
02CEh			
02CFh			
02D0h			
02D1h			
02D2h			
02D3h			
02D4h			
02D5h			
02D6h			
02D7h			
02D8h			
02D9h			
02DAh			
02DBh			
02DCh			
02DDh			
02DEh			
02DFh			

NOTE:

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

X: Undefined

Table 4.10 SFR Information (10) <sup>(1)</sup>

Address	Register	Symbol	After Reset
02E0h			
02E1h			
02E2h			
02E3h			
02E4h			
02E5h			
02E6h			
02E7h			
02E8h			
02E9h			
02EAh			
02EBh			
02ECh			
02EDh			
02EEh			
02EFh			
02F0h			
02F1h			
02F2h			
02F3h			
02F4h			
02F5h			
02F6h			
02F7h			
02F8h			
02F9h			
02FAh			
02FBh			
02FCh			
02FDh			
02FEh			
02FFh			
0300h	Timer B3,4,5 Count Start Flag	TBSR	000XXXXXb
0301h			
0302h	Timer A1-1 Register	TA11	XXh
0303h			XXh
0304h	Timer A2-1 Register	TA21	XXh
0305h			XXh
0306h	Timer A4-1 Register	TA41	XXh
0307h			XXh
0308h	Three-Phase PWM Control Register 0	INVC0	00h
0309h	Three-Phase PWM Control Register 1	INVC1	00h
030Ah	Three-Phase Output Buffer Register 0	IDB0	XX111111b
030Bh	Three-Phase Output Buffer Register 1	IDB1	XX111111b
030Ch	Dead Time Timer	DTT	XXh
030Dh	Timer B2 Interrupt Generation Frequency Set Counter	ICTB2	XXh
030Eh			
030Fh			
0310h	Timer B3 Register	TB3	XXh
0311h			XXh
0312h	Timer B4 Register	TB4	XXh
0313h			XXh
0314h	Timer B5 Register	TB5	XXh
0315h			XXh
0316h			
0317h			
0318h			
0319h			
031Ah			
031Bh	Timer B3 Mode Register	TB3MR	00XX0000b
031Ch	Timer B4 Mode Register	TB4MR	00XX0000b
031Dh	Timer B5 Mode Register	TB5MR	00XX0000b
031Eh			
031Fh			

NOTE:

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

X: Undefined



**Table 4.11 SFR Information (11) <sup>(1)</sup>**

Address	Register	Symbol	After Reset
0320h	Count Start Flag	TABSR	00h
0321h			
0322h	One-Shot Start Flag	ONSF	00h
0323h	Trigger Select Register	TRGSR	00h
0324h	Up/Down Flag	UDF	00h
0325h			
0326h	Timer A0 Register	TA0	XXh
0327h			XXh
0328h	Timer A1 Register	TA1	XXh
0329h			XXh
032Ah	Timer A2 Register	TA2	XXh
032Bh			XXh
032Ch	Timer A3 Register	TA3	XXh
032Dh			XXh
032Eh	Timer A4 Register	TA4	XXh
032Fh			XXh
0330h	Timer B0 Register	TB0	XXh
0331h			XXh
0332h	Timer B1 Register	TB1	XXh
0333h			XXh
0334h	Timer B2 Register	TB2	XXh
0335h			XXh
0336h	Timer A0 Mode Register	TA0MR	00h
0337h	Timer A1 Mode Register	TA1MR	00h
0338h	Timer A2 Mode Register	TA2MR	00h
0339h	Timer A3 Mode Register	TA3MR	00h
033Ah	Timer A4 Mode Register	TA4MR	00h
033Bh	Timer B0 Mode Register	TB0MR	00XX0000b
033Ch	Timer B1 Mode Register	TB1MR	00XX0000b
033Dh	Timer B2 Mode Register	TB2MR	00XX0000b
033Eh	Timer B2 Special Mode Register	TB2SC	XXXXXX00b
033Fh			
0340h			
0341h			
0342h			
0343h			
0344h			
0345h			
0346h			
0347h			
0348h			
0349h			
034Ah			
034Bh			
034Ch			
034Dh			
034Eh			
034Fh			
0350h			
0351h			
0352h			
0353h			
0354h			
0355h			
0356h			
0357h			
0358h			
0359h			
035Ah			
035Bh			
035Ch			
035Dh			
035Eh			
035Fh			

NOTE: X: Undefined  
 1. The blank areas are reserved and cannot be accessed by users.

**Table 4.12 SFR Information (12)** <sup>(1)</sup>

Address	Register	Symbol	After Reset
0360h	Pull-Up Control Register 0	PUR0	00h
0361h	Pull-Up Control Register 1	PUR1	00000000b <sup>(2)</sup> 00000010b
0362h	Pull-Up Control Register 2	PUR2	00h
0363h			
0364h			
0365h			
0366h	Port Control Register	PCR	0000XX0bh
0367h			
0368h			
0369h			
036Ah			
036Bh			
036Ch			
036Dh			
036Eh			
036Fh			
0370h			
0371h			
0372h			
0373h			
0374h			
0375h			
0376h			
0377h			
0378h			
0379h			
037Ah			
037Bh			
037Ch	Count Source Protection Mode Register	CSPR	00h <sup>(3)</sup>
037Dh	Watchdog Timer Reset Register	WDTR	XXh
037Eh	Watchdog Timer Start Register	WDTS	XXh
037Fh	Watchdog Timer Control Register	WDC	00XXXXXXb
0380h			
0381h			
0382h			
0383h			
0384h			
0385h			
0386h			
0387h			
0388h			
0389h			
038Ah			
038Bh			
038Ch			
038Dh			
038Eh			
038Fh			

NOTES:

X: Undefined

- The blank areas are reserved and cannot be accessed by users.
- Values after hardware reset 1 or brown-out reset are as follows:
  - 00000000b when "L" is input to the CNVSS pin
  - 00000010b when "H" is input to the CNVSS pin
 Values after software reset, watchdog timer reset, and oscillation stop detection reset are as follows:
  - 00000000b when bits PM01 and PM00 in the PM0 register are set to 00b (single-chip mode).
  - 00000010b when bits PM01 and PM00 in the PM0 register are set to 01b (memory expansion mode) or 11b (microprocessor mode).
- When the CSPROINT bit in the OFS1 address is set to 0, value after reset is 10000000b

**Table 4.13 SFR Information (13)** <sup>(1)</sup>

Address	Register	Symbol	After Reset
0390h	DMA2 Source Select Register	DM2SL	00h
0391h			
0392h	DMA3 Source Select Register	DM3SL	00h
0393h			
0394h			
0395h			
0396h			
0397h			
0398h	DMA0 Source Select Register	DM0SL	00h
0399h			
039Ah	DMA1 Source Select Register	DM1SL	00h
039Bh			
039Ch			
039Dh			
039Eh			
039Fh			
03A0h			
03A1h			
03A2h			
03A3h			
03A4h			
03A5h			
03A6h			
03A7h			
03A8h			
03A9h			
03AAh			
03ABh			
03ACh			
03ADh			
03AEh			
03AFh			
03B0h			
03B1h			
03B2h			
03B3h			
03B4h			
03B5h			
03B6h			
03B7h			
03B8h			
03B9h			
03BAh			
03BBh			
03BCh	CRC Data Register	CRCD	XXh
03BDh			XXh
03BEh	CRC Input Register	CRCIN	XXh
03BFh			
03C0h	A/D Register 0	AD0	XXXXXXXXb
03C1h			000000XXb
03C2h	A/D Register 1	AD1	XXXXXXXXb
03C3h			000000XXb
03C4h	A/D Register 2	AD2	XXXXXXXXb
03C5h			000000XXb
03C6h	A/D Register 3	AD3	XXXXXXXXb
03C7h			000000XXb
03C8h	A/D Register 4	AD4	XXXXXXXXb
03C9h			000000XXb

NOTE:

X: Undefined

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

**Table 4.14 SFR Information (14)** <sup>(1)</sup>

Address	Register	Symbol	After Reset
03CAh 03CBh	A/D Register 5	AD5	XXXXXXXXb 000000XXb
03CCh 03CDh	A/D Register 6	AD6	XXXXXXXXb 000000XXb
03CEh 03CFh	A/D Register 7	AD7	XXXXXXXXb 000000XXb
03D0h			
03D1h			
03D2h			
03D3h			
03D4h	A/D Control Register 2	ADCON2	0000X00Xb
03D5h			
03D6h	A/D Control Register 0	ADCON0	00000XXXb
03D7h	A/D Control Register 1	ADCON1	0000X000b
03D8h	D/A0 Register	DA0	00h
03D9h			
03DAh	D/A1 Register	DA1	00h
03DBh			
03DCh	D/A Control Register	DACON	00h
03DDh			
03DEh			
03DFh			
03E0h	Port P0 Register	P0	XXh
03E1h	Port P1 Register	P1	XXh
03E2h	Port P0 Direction Register	PD0	00h
03E3h	Port P1 Direction Register	PD1	00h
03E4h	Port P2 Register	P2	XXh
03E5h	Port P3 Register	P3	XXh
03E6h	Port P2 Direction Register	PD2	00h
03E7h	Port P3 Direction Register	PD3	00h
03E8h	Port P4 Register	P4	XXh
03E9h	Port P5 Register	P5	XXh
03EAh	Port P4 Direction Register	PD4	00h
03EBh	Port P5 Direction Register	PD5	00h
03ECh	Port P6 Register	P6	XXh
03EDh	Port P7 Register	P7	XXh
03EEh	Port P6 Direction Register	PD6	00h
03EFh	Port P7 Direction Register	PD7	00h
03F0h	Port P8 Register	P8	XXh
03F1h	Port P9 Register	P9	XXh
03F2h	Port P8 Direction Register	PD8	00h
03F3h	Port P9 Direction Register	PD9	00h
03F4h	Port P10 Register	P10	XXh
03F5h			
03F6h	Port P10 Direction Register	PD10	00h
03F7h			
03F8h			
03F9h			
03FAh			
03FBh			
03FCh			
03FDh			
03FEh			
03FFh			
D000h to D7FFh			

NOTE:

X: Undefined

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

## 5. Reset

Hardware reset 1, brown-out reset, software reset, watchdog timer reset and oscillation stop detection reset are available to reset the microcomputer.

### 5.1 Hardware Reset 1

The microcomputer resets pins, the CPU, and SFR by setting the  $\overline{\text{RESET}}$  pin. If the supply voltage meets the recommended operating conditions, the microcomputer resets all pins, the CPU, and SFR when an “L” signal is applied to the  $\overline{\text{RESET}}$  pin (see **Table 5.1 “Pin Status When  $\overline{\text{RESET}}$  Pin Level is “L”**”).

When the signal applied to the  $\overline{\text{RESET}}$  pin changes low (“L”) to high (“H”), the microcomputer executes the program in an address indicated by the reset vector. The 125 kHz on-chip oscillator clock divided by 8 is automatically selected as a CPU clock after reset.

Refer to **4. “Special Function Registers (SFRs)”** for SFR states after reset.

The internal RAM is not reset. When an “L” signal is applied to the  $\overline{\text{RESET}}$  pin while writing data to the internal RAM, the internal RAM is in an indeterminate state.

Figure 5.1 shows an Example Reset Circuit. Figure 5.2 shows a Reset Sequence. Table 5.1 lists Pin Status When  $\overline{\text{RESET}}$  Pin Level is “L”.

#### 5.1.1 Reset on a Stable Supply Voltage

- (1) Apply “L” to the  $\overline{\text{RESET}}$  pin
- (2) Wait for  $1/f_{\text{OCO-S}} \times 20$
- (3) Apply an “H” signal to the  $\overline{\text{RESET}}$  pin

#### 5.1.2 Power-on Reset

- (1) Apply “L” to the  $\overline{\text{RESET}}$  pin
- (2) Raise the supply voltage to the recommended operating level
- (3) Insert  $t_{\text{d(P-R)}}$  ms as wait time for the internal voltage to stabilize
- (4) Wait for  $1/f_{\text{OCO-S}} \times 20$
- (5) Apply “H” to the  $\overline{\text{RESET}}$  pin

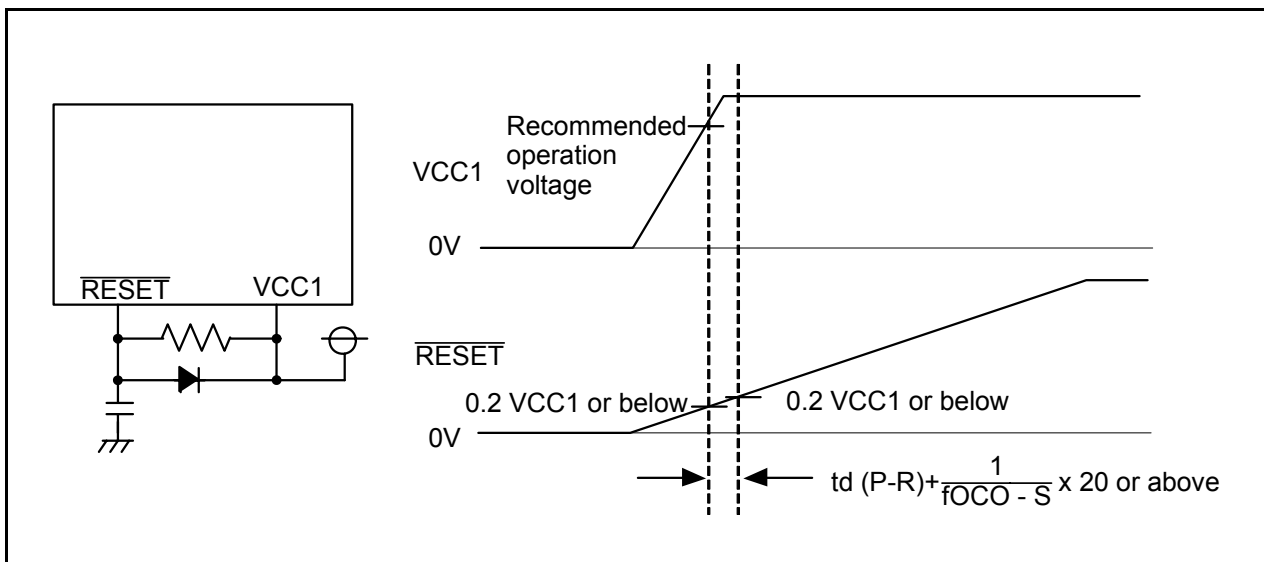


Figure 5.1 Example Reset Circuit

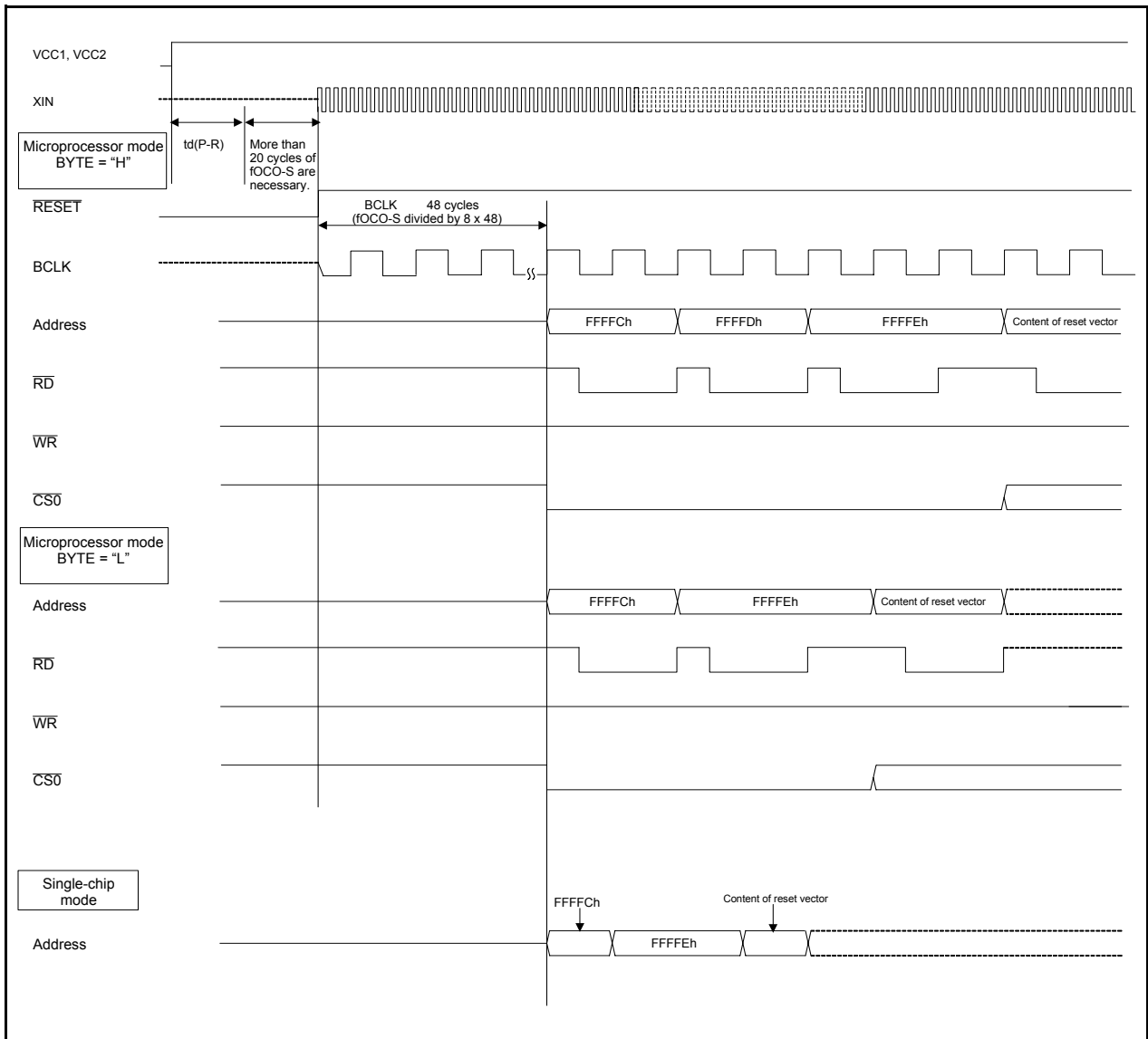


Figure 5.2 Reset Sequence

**Table 5.1 Pin Status When  $\overline{\text{RESET}}$  Pin Level is “L”**

Pin Name	Status		
	CNVSS = VSS	CNVSS = VCC1 (1)	
		BYTE = VSS	BYTE = VCC1
P0	Input port	Data input	Data input
P1	Input port	Data input	Input port
P2, P3, P4_0 to P4_3	Input port	Address output (undefined)	Address output (undefined)
P4_4	Input port	$\overline{\text{CS0}}$ output (“H” is output)	$\overline{\text{CS0}}$ output (“H” is output)
P4_5 to P4_7	Input port	Input port (pulled high)	Input port (pulled high)
P5_0	Input port	$\overline{\text{WR}}$ output (“H” is output)	$\overline{\text{WR}}$ output (“H” is output)
P5_1	Input port	$\overline{\text{BHE}}$ output (undefined)	$\overline{\text{BHE}}$ output (undefined)
P5_2	Input port	$\overline{\text{RD}}$ output (“H” is output)	$\overline{\text{RD}}$ output (“H” is output)
P5_3	Input port	BCLK output	BCLK output
P5_4	Input port	$\overline{\text{HLDA}}$ output (The output value depends on the input to the $\overline{\text{HOLD}}$ pin)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the $\overline{\text{HOLD}}$ pin)
P5_5	Input port	$\overline{\text{HOLD}}$ input (2)	$\overline{\text{HOLD}}$ input (2)
P5_6	Input port	ALE output (“L” is output)	ALE output (“L” is output)
P5_7	Input port	$\overline{\text{RDY}}$ input	$\overline{\text{RDY}}$ input
P6, P7, P8, P9, P10	Input port	Input port	Input port

## NOTE:

1. Shown here is the valid pin state when the internal power supply voltage has stabilized after power on. When CNVSS = VCC1, the pin state is indeterminate until the internal power supply voltage stabilizes.
2. Apply a “H” signal.

## 5.2 Brown-out Reset

The microcomputer resets pins, the CPU, or SFRs by setting the built-in voltage detection 0 circuit. The voltage detection 0 circuit monitors the voltage applied to the VCC1 pin (Vdet0).

The microcomputer resets pins, the CPU, and SFR as soon as the voltage that is applied to the VCC1 pin drops to Vdet0 or below.

Then, 125 kHz on-chip oscillator clock starts counting when the voltage that is applied to the VCC1 pin goes up to Vdet0 or above. The internal reset signal becomes “H” after the 125 kHz on-chip oscillator clock is counted 32 times, and then reset sequence starts (see Figure 5.2). The 125 kHz on-chip oscillator clock divided by 8 is automatically selected as a CPU clock after reset.

Refer to 4. “**Special Function Registers (SFRs)**” for the SFR status after brown-out reset.

The internal RAM is not reset. When the voltage that is applied to the VCC1 pin drops to Vdet0 or below while writing data to the internal RAM, the internal RAM is in an indeterminate state.

Refer to 6. “**Voltage Detection Circuit**” for details of the voltage detection 0 circuit.

## 5.3 Software Reset

The microcomputer resets pins, the CPU, and SFRs when the PM03 bit in the PM0 register is set to 1 (microcomputer reset). Then the microcomputer executes the program in an address determined by the reset vector. The 125 kHz on-chip oscillator clock divided by 8 is automatically selected as a CPU clock after reset.

In the software reset, the microcomputer does not reset a part of the SFRs. Refer to 4. “**Special Function Registers (SFRs)**” for details.

The internal RAM is not reset.

## 5.4 Watchdog Timer Reset

The microcomputer resets pins, the CPU, and SFRs when the PM12 bit in the PM1 register is set to 1 (reset when watchdog timer underflows) and the watchdog timer underflows. Then the microcomputer executes the program in an address determined by the reset vector. The 125 kHz on-chip oscillator clock divided by 8 is automatically selected as a CPU clock after reset.

In the watchdog timer reset, the microcomputer does not reset a part of the SFRs. Refer to 4. “**Special Function Registers (SFRs)**” for details.

The internal RAM is not reset. When the watchdog timer underflows while writing data to the internal RAM, the internal RAM is in an indeterminate state.

Refer to 13. “**Watchdog Timer**” for details.

## 5.5 Oscillation Stop Detection Reset

The microcomputer resets and stops pins, the CPU, and SFRs when the CM27 bit in the CM2 register is 0 (reset when oscillation stop detected), if it detects main clock oscillation circuit stop. Refer to 10.6 “**Oscillation Stop and Re-Oscillation Detect Function**” for details.

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



## 5.6 Internal Space

Figure 5.3 shows CPU Register Status After Reset. Refer to 4. “Special Function Registers (SFRs)” for SFR states after reset.

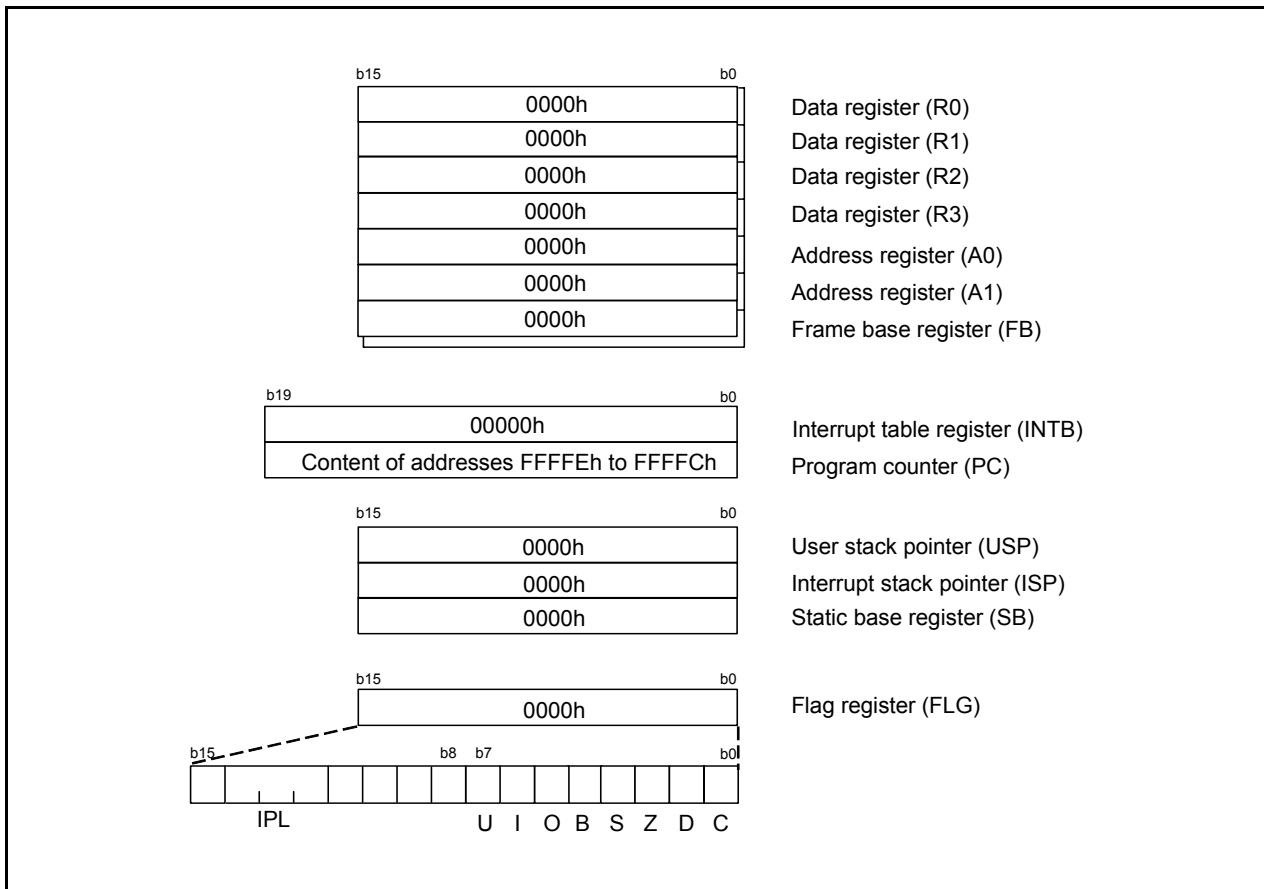


Figure 5.3 CPU Register Status After Reset

## 6. Voltage Detection Circuit

The voltage detection circuit consists of the voltage detection 0 circuit and the low voltage detection circuit. The voltage detection 0 circuit monitors the voltage applied to the VCC1 pin. The microcomputer is reset if the voltage detection 0 circuit detects VCC1 is Vdet0 or below.

The low voltage detection circuit also monitors the voltage applied to the VCC1 pin. The low voltage detection signal is generated when the low voltage detection circuit detects that VCC1 passes through Vdet2. This signal generates the low voltage detection interrupt. The VC13 bit in the VCR1 register determines whether VCC1 is Vdet2 and above or below Vdet2.

The voltage detection circuit is available when VCC1 = 5 V.

Figure 6.1 shows a Voltage Detection Circuit Block Diagram.

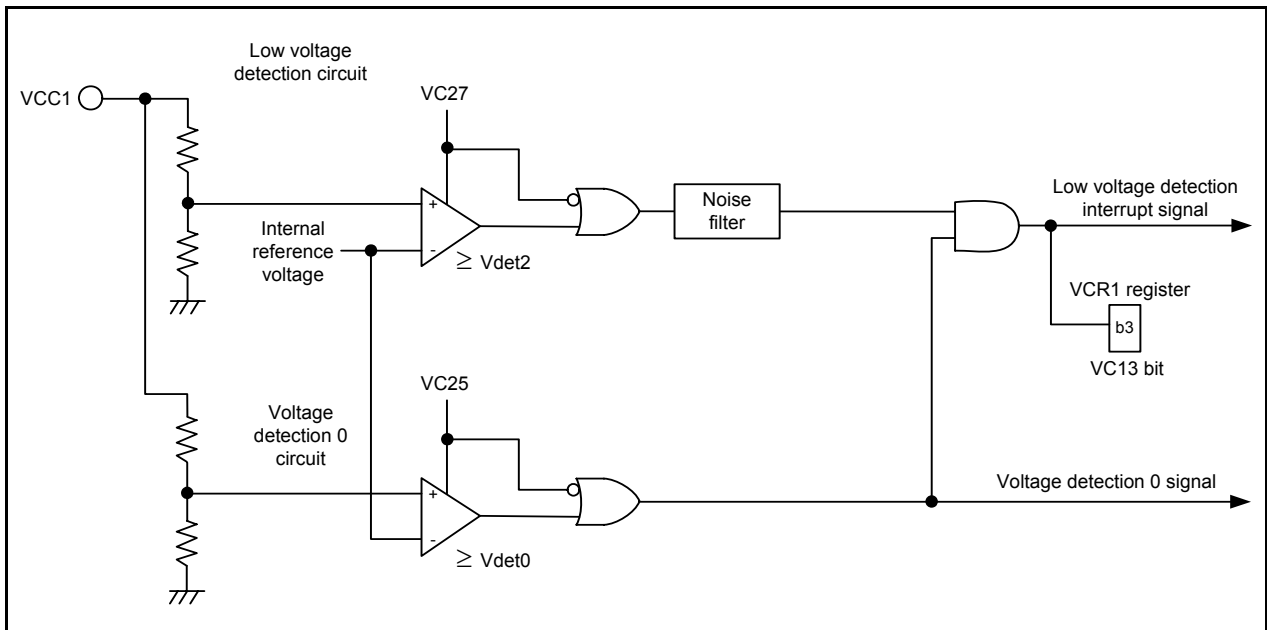
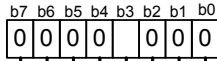


Figure 6.1 Voltage Detection Circuit Block Diagram

### Voltage Detection 2 Circuit Flag Register



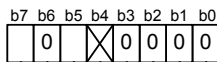
Symbol: VCR1  
 Address: 0019h  
 After Reset <sup>(2)</sup>: 00001000b

Bit Symbol	Bit Name	Function	RW
— (b2-b0)	Reserved bits	Set to 0	RW
VC13	Low voltage monitor flag <sup>(1)</sup>	0 : VCC1 < Vdet2 1 : VCC1 ≥ Vdet2	RO
— (b7-b4)	Reserved bits	Set to 0	RW

NOTES :

1. The VC13 bit is enabled when the VC27 bit in the VCR2 register is set to 1 (low voltage detection circuit enabled). The VC13 bit is always 1 (VCC1 ≥ Vdet2) when the VC27 bit is set to 0 (low voltage detection circuit disabled).
2. This register dose not change at software reset, watchdog timer reset, and oscillation stop detection reset.

### Voltage Detection Circuit Operation Enable Register <sup>(1)</sup>



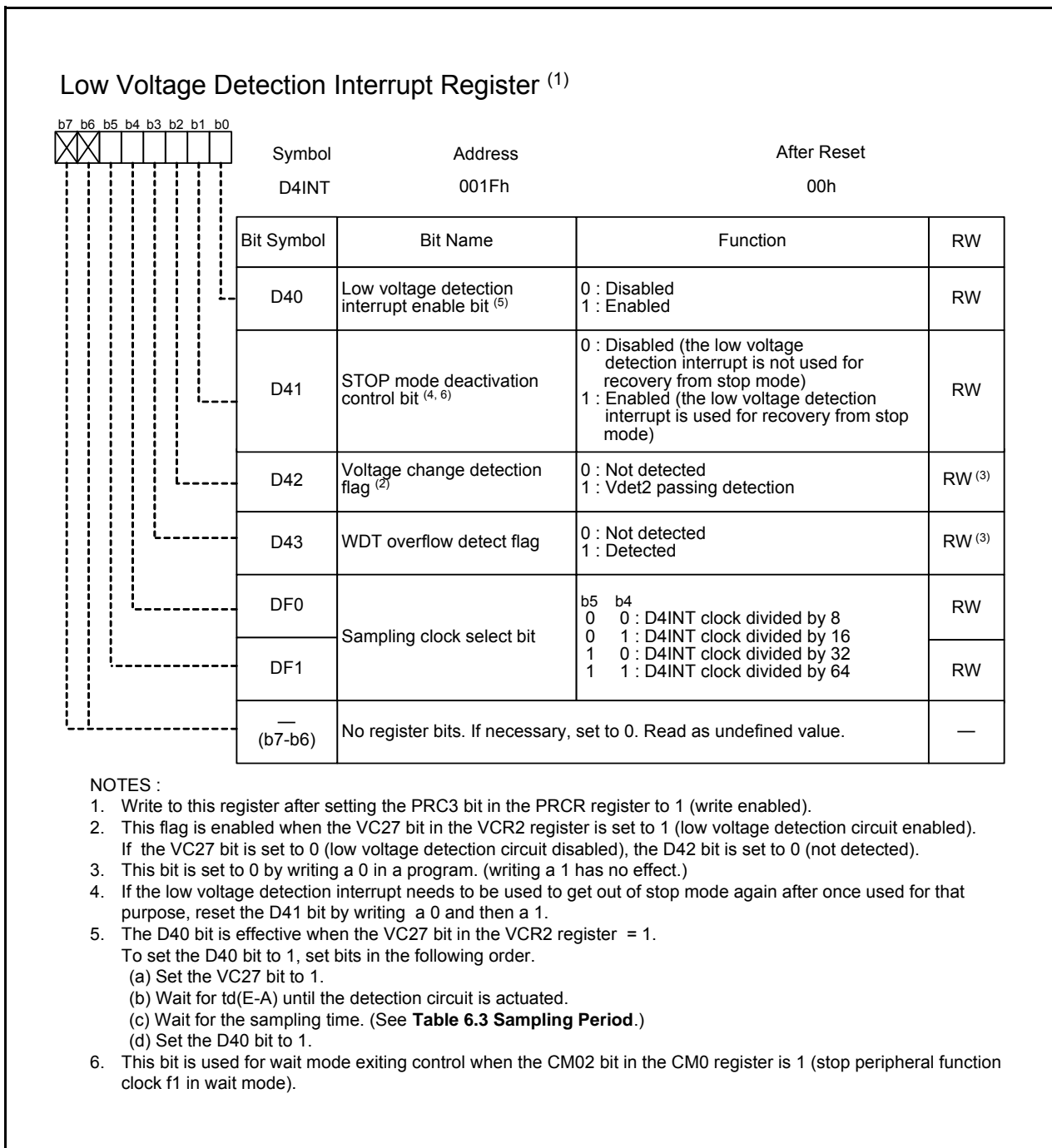
Symbol: VCR2  
 Address: 001Ah  
 After Reset <sup>(4)</sup>: 000X0000b (Hardware reset 1)  
 001X0000b (Brown-out reset)

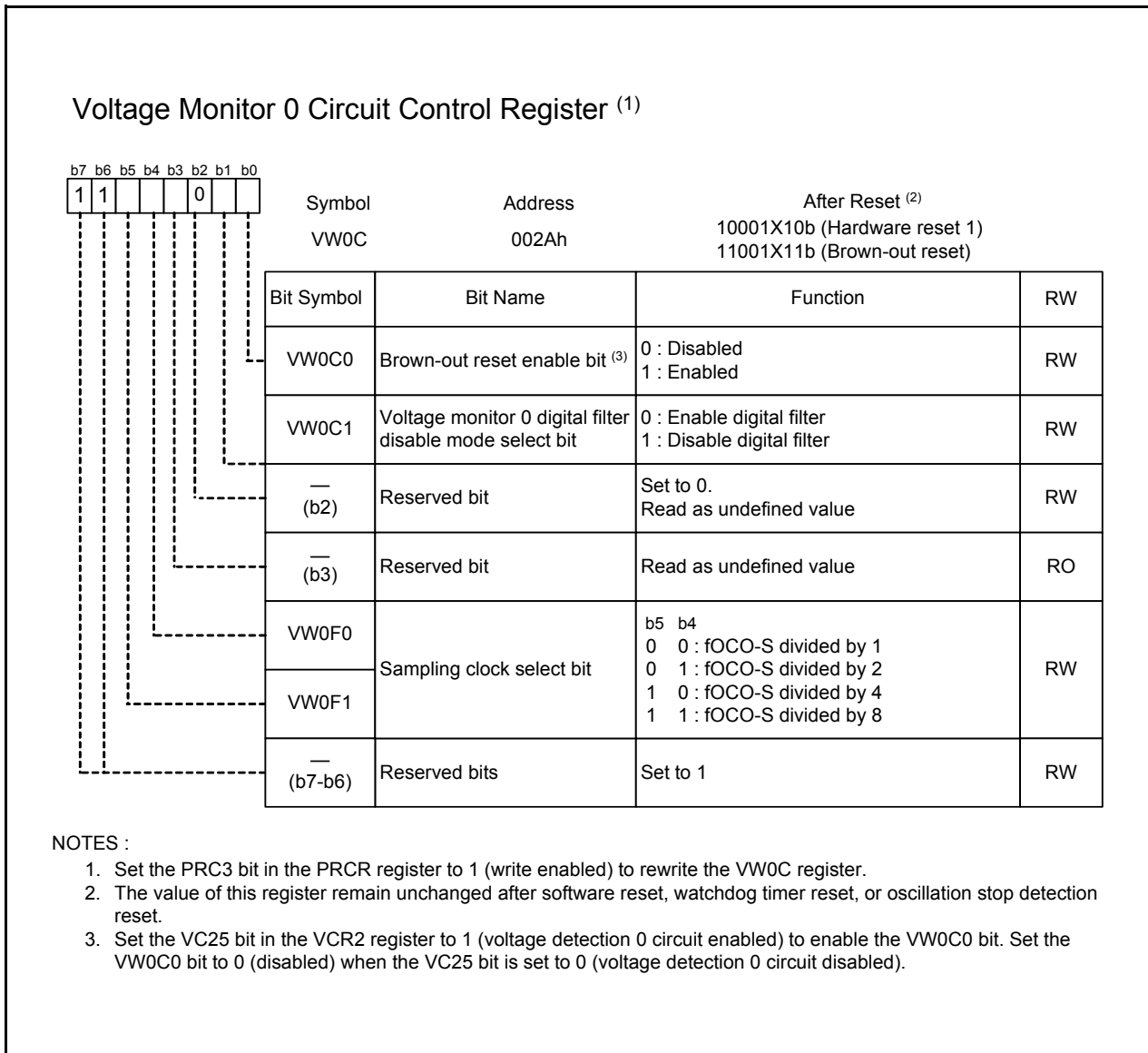
Bit Symbol	Bit Name	Function	RW
— (b3-b0)	Reserved bits	Set to 0	RW
— (b4)	No register bit. If necessary, set to 0. Read as undefined value	—	—
VC25	Voltage detection 0 enable bit <sup>(2, 5)</sup>	0 : Disable voltage detection 0 circuit 1 : Enable voltage detection 0 circuit	RW
— (b6)	Reserved bit	Set to 0	RW
VC27	Low voltage monitor bit <sup>(3, 5)</sup>	0 : Disable low voltage detection circuit 1 : Enable low voltage detection circuit	RW

NOTES :

1. Write to this register after setting the PRC3 bit in the PRCR register to 1 (write enabled).
2. To use brown-out reset, set the VC25 bit to 1 (voltage detection 0 circuit enabled).
3. 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 (low voltage detection interrupt enabled), set the VC27 bit to 1 (low voltage detection circuit enabled).
4. This register does not change at software reset, watchdog timer reset, and oscillation stop detection reset.
5. The detection circuit does not start operation until td(E-A) elapses after the VC25 bit or VC27 bit is set to 1.

Figure 6.2 Registers VCR1 and VCR2

**Figure 6.3 D4INT Register**



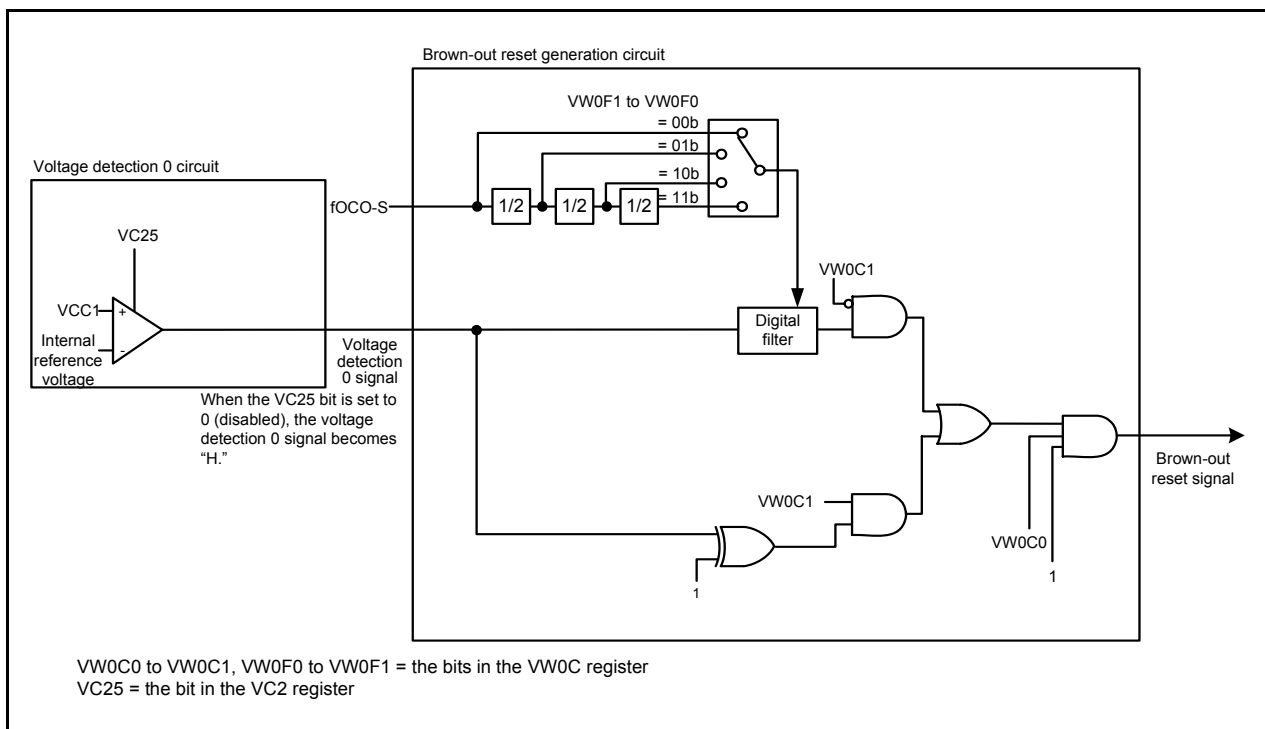
**Figure 6.4 VW0C Register**

### 6.1 Brown-out Reset

Figure 6.5 is a block diagram illustrating brown-out reset generation circuit. Table 6.1 shows a setting procedure of the bits for brown-out reset. Figure 6.6 provides an example of brown-out reset operation. When using brown-out reset to exit stop mode, set the VW0C1 bit in the VW0C register to 1 (digital filter disabled).

**Table 6.1 Setting Procedures of the Bits for Brown-out Reset**

Procedure	When using the digital filter	When not using the digital filter
1	Set the VC25 bit in the VCR2 register to 1 (voltage detection 0 circuit enabled)	
2	Wait for $t_d$ (E-A)	
3	Use bits VW0F0 to VW0F1 in the VW0C register to select the digital filter sampling clock. Set the VW0C1 bit to 0 (digital filter enabled), bits 6 and 7 to 1	Set the VW0C1 bit in the VW0C register to 1 (digital filter disabled), and bits 6 and 7 to 1
4	Set bit 2 in the VW0C register to 0 (setting bit 2 to 0 once again after procedure 3 is necessary)	
5	Set the CM14 bit in the CM1 register to 0 (125 kHz on-chip oscillator oscillates)	-
6	Wait for digital filter sampling clock x 4 cycles	- (no wait time)
7	Set the VW0C0 bit in the VW0C register to 1 (brown-out reset enabled)	



**Figure 6.5 Brown-out Reset Generation Circuit**

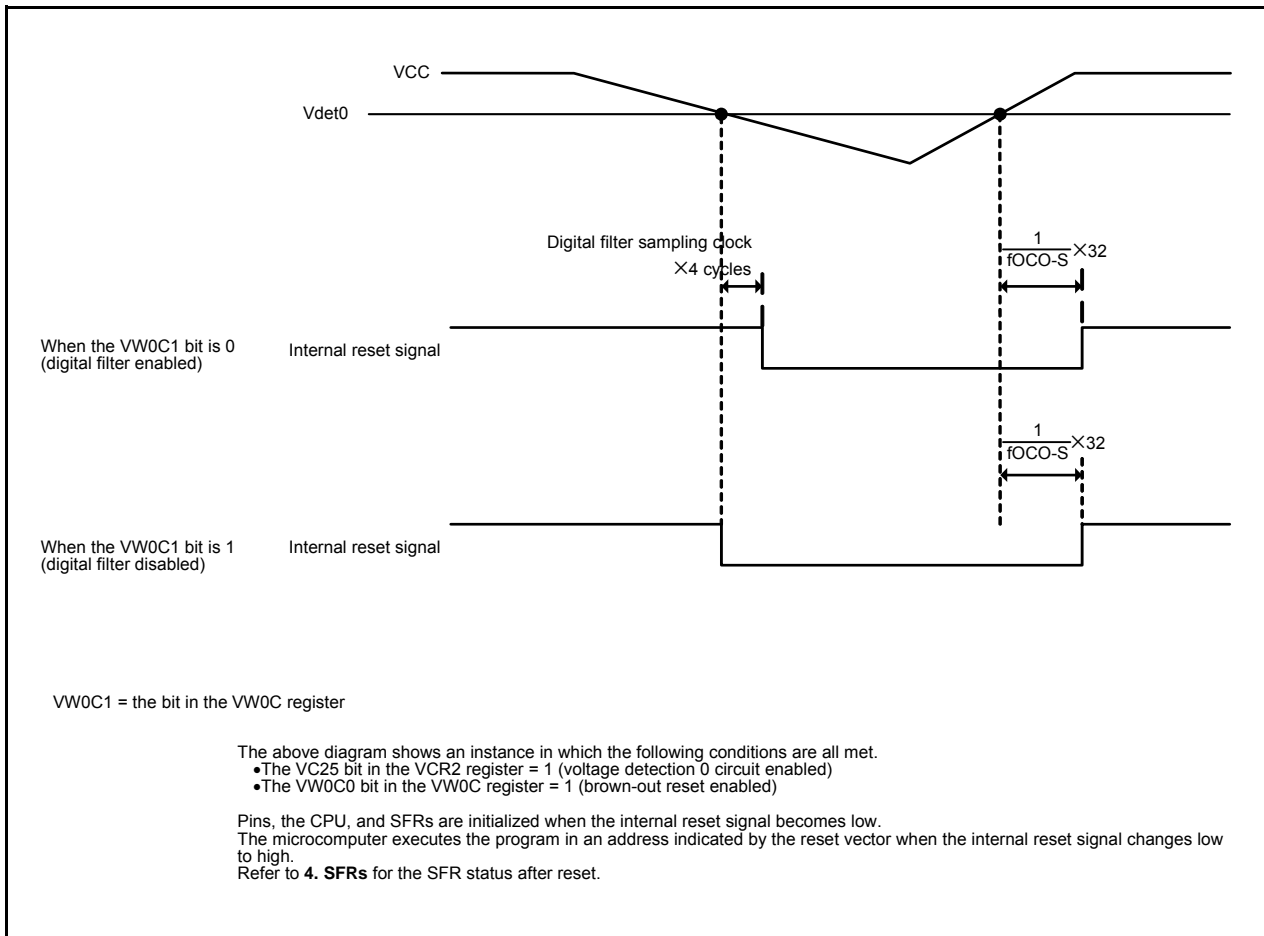


Figure 6.6 Brown-out Reset Operation Example

## 6.2 Low Voltage Detection Interrupt

If the D40 bit in the D4INT register is set to 1 (low voltage detection interrupt enabled), the low voltage detection interrupt request is generated when the voltage applied to the VCC1 pin is above or below Vdet2. The low voltage detection interrupt shares the same interrupt vector with the watchdog timer interrupt, oscillation stop, and re-oscillation detection interrupt.

Set the D41 bit in the D4INT register to 1 (enabled) to use the low voltage detection interrupt to exit stop mode.

The D42 bit in the D4INT register is set to 1 as soon as the voltage applied to the VCC1 pin reaches Vdet2 due to the voltage rise and voltage drop. When the D42 bit changes 0 to 1, the low voltage detection interrupt request is generated. Set the D42 bit to 0 by program. However, when the D41 bit is set to 1 and the microcomputer is in stop mode, the low voltage detection interrupt request is generated regardless of the D42 bit state if the voltage applied to the VCC1 pin is detected to be above Vdet2. The microcomputer then exits stop mode.

Table 6.2 shows Low Voltage Detection Interrupt Request Generation Conditions.

Bits DF1 to DF0 in the D4INT register determine the sampling period that detects the voltage applied to the VCC1 pin reaches Vdet2. Table 6.3 shows the Sampling Periods.

**Table 6.2 Low Voltage Detection Interrupt Request Generation Conditions**

Operating Mode	VC27 Bit	D40 Bit	D41 Bit	D42 Bit	CM02 Bit	VC13 Bit	
Normal Operating Mode (1)	1	1	-	0 to 1	-	0 to 1 (3)	
						1 to 0 (3)	
Wait Mode (2)			-	0 to 1	0	0 to 1 (3)	
					1	1 to 0 (3)	
Stop Mode (2)			-	1	-	0	0 to 1
							0 to 1

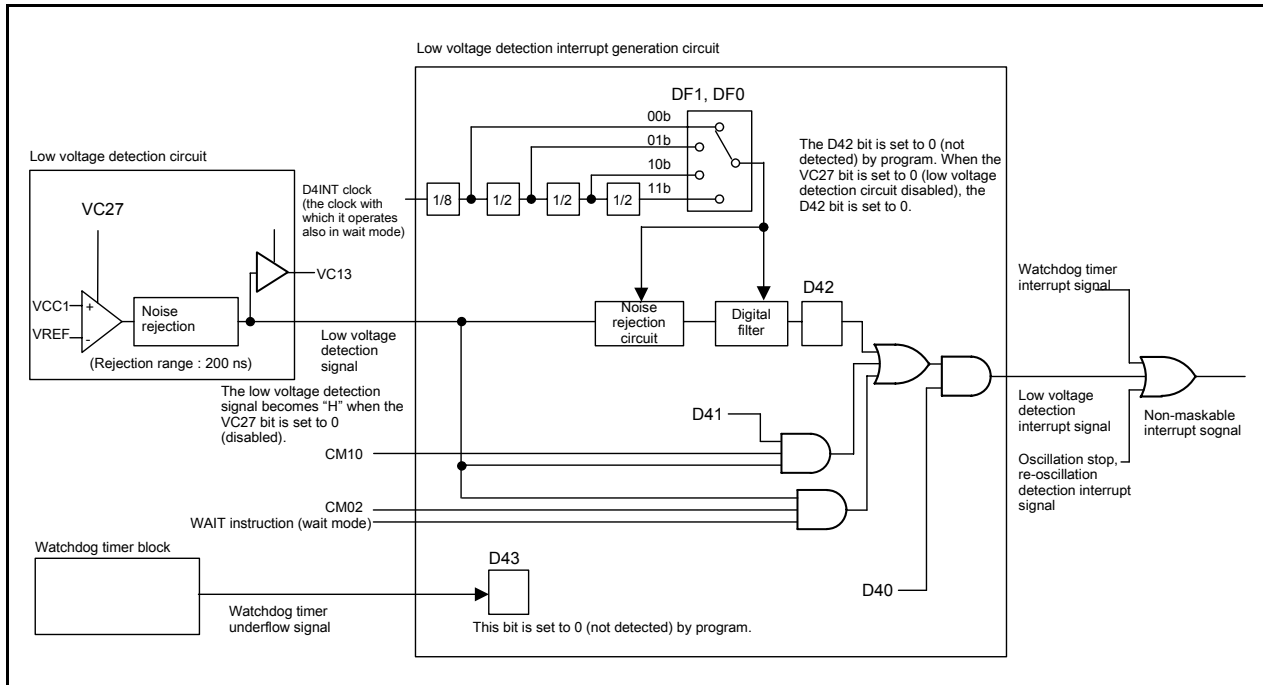
- indicates either 0 or 1 is settable.

**NOTE:**

1. The status except wait mode and stop mode is handled as normal mode. (Refer to 10. "Clock Generation Circuit")
2. Refer to 6.3 "Limitations on Exiting Stop Mode" and 6.4 "Limitations on Exiting Wait Mode".
3. An interrupt request for voltage reduction is generated after the value of the VC13 bit changes and a sampling time elapses. See Figure 6.8 "Low Voltage Detection Interrupt Generation Circuit Operation Example" for details.

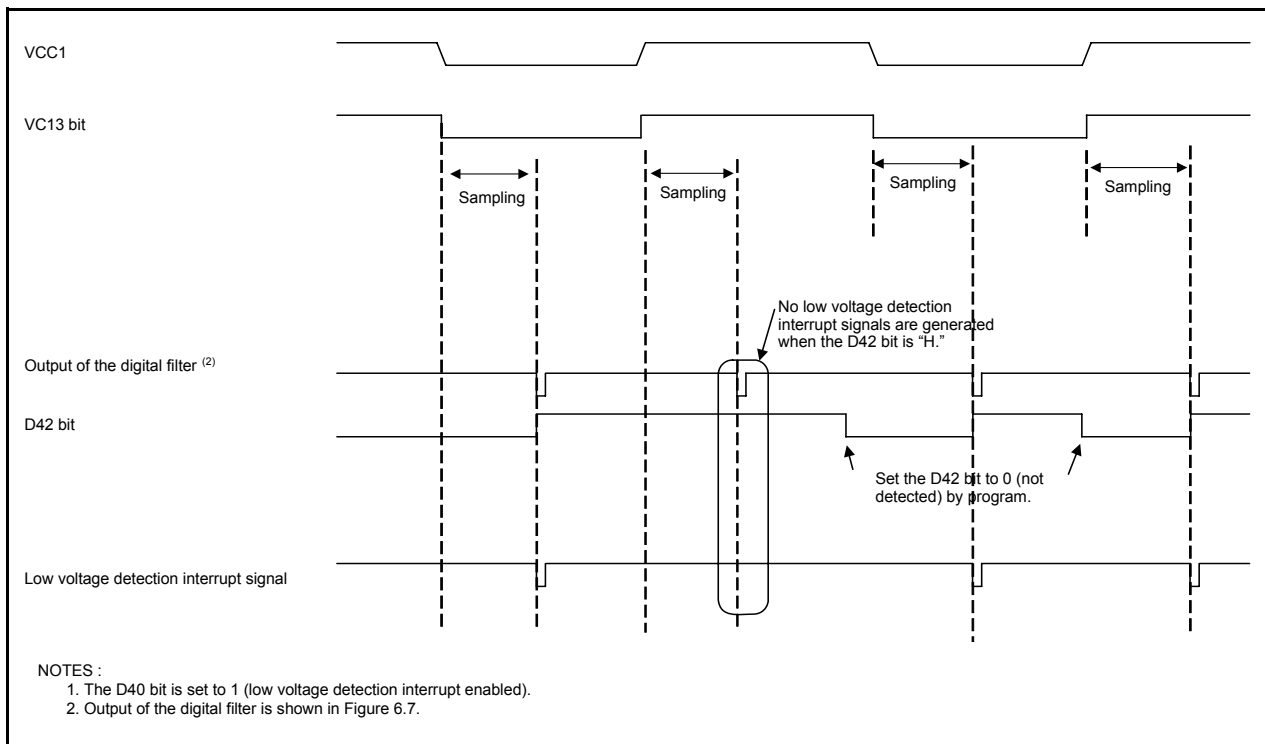
**Table 6.3 Sampling Periods**

CPU Clock (D4INT 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



**Figure 6.7 Low Voltage detection Interrupt Generation Block Diagram**





**Figure 6.8 Low Voltage Detection Interrupt Generation Circuit Operation Example**

### 6.3 Limitations on Exiting Stop Mode

The low voltage detection interrupt is immediately generated and the microcomputer exits stop mode if the CM10 bit in the CM1 register is set to 1 (stop mode) under the conditions below.

- the VC27 bit in the VCR2 register is set to 1 (low voltage detection circuit enabled)
- the D40 bit in the D4INT register is set to 1 (low voltage detection interrupt enabled)
- the D41 bit in the D4INT register is set to 1 (low voltage detection interrupt is used to exit stop mode)
- the voltage applied to the VCC1 pin is higher than Vdet2 (the VC13 bit in the VCR1 register is 1)

If the microcomputer is set to enter stop mode when the voltage applied to the VCC1 pin drops below Vdet2 and to exit stop mode when the voltage applied rises to Vdet2 or above, set the CM10 bit to 1 when VC13 bit is 0 ( $VCC1 < Vdet2$ ).

### 6.4 Limitations on Exiting Wait Mode

The low voltage detection interrupt is immediately generated and the microcomputer exits wait mode if WAIT instruction is executed under the conditions below.

- the CM02 bit in the CM0 register is set to 1 (stop peripheral function clock)
- the VC27 bit in the VCR2 register is set to 1 (low voltage detection circuit enabled)
- the D40 bit in the D4INT register is set to 1 (low voltage detection interrupt enabled)
- the D41 bit in the D4INT register is set to 1 (low voltage detection interrupt is used to exit wait mode)
- the voltage applied to the VCC1 pin is higher than Vdet2 (the VC13 bit in the VCR1 register is 1)

If the microcomputer is set to enter wait mode when the voltage applied to the VCC1 pin drops below Vdet2 and to exit wait mode when the voltage applied rises to Vdet2 or above, perform WAIT instruction when the VC13 bit is 0 ( $VCC1 < Vdet2$ ).

### 6.5 Cold Start-up / Warm Start-up Discrimination

As for the cold start-up / warm start-up discrimination, the CWR bit in the RSTFR register determines either cold start-up (reset process) when power-on or warm start-up (reset process) when reset signal is applied during the microcomputer running.

The value of the CWR bit is 0 when power is applied. The CWR bit is also set to 0 after brown-out reset. The CWR bit is set to 1 by writing a 1 in a program and does not change at hardware reset 1, software reset, watchdog timer reset, and oscillation stop detection reset.

Use brown-out reset for cold start-up / warm start-up discrimination.

Follow Table 6.1 Setting Procedures of the Bits for Brown-out Reset to set the bits for brown-out reset.

Figure 6.9 shows Cold Start-up / Warm Start-up Discrimination Example. Figure 6.10 shows RSTFR Register.

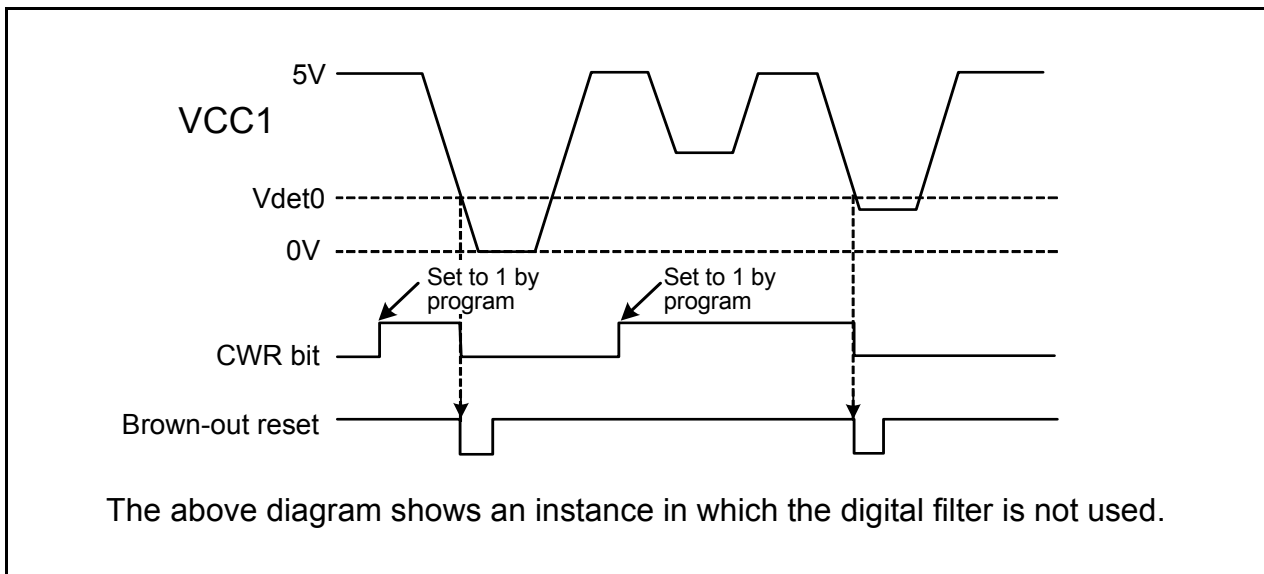


Figure 6.9 Cold Start-up / Warm Start-up Discrimination Example

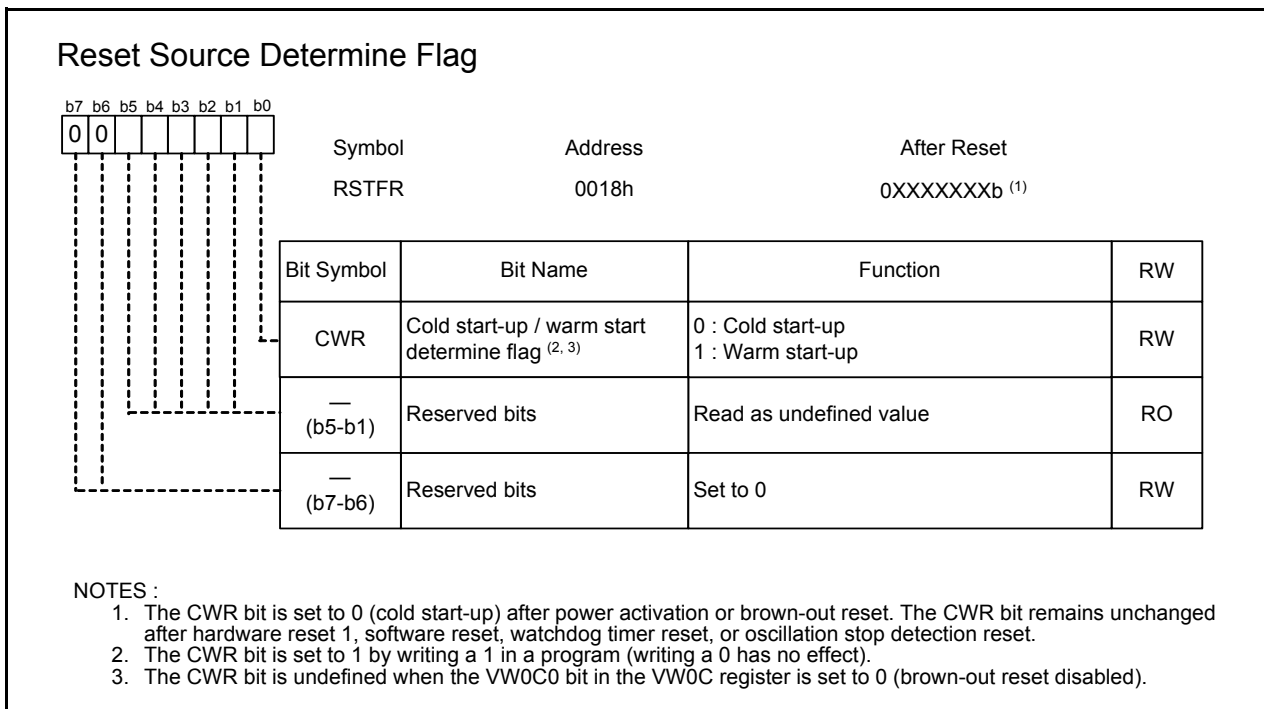


Figure 6.10 RSTFR Register

## 7. Processor Mode

### 7.1 Types of Processor Mode

Three processor modes are available to choose from: single-chip mode, memory expansion mode, and microprocessor mode. Table 7.1 shows the Features of Processor Modes.

**Table 7.1 Features of Processor Modes**

Processor Modes	Access Space	Pins Which Are Assigned I/O Ports
Single-chip mode	SFR, internal RAM, internal ROM	All pins are I/O ports or peripheral function I/O pins
Memory expansion mode	SFR, internal RAM, internal ROM, external area <sup>(1)</sup>	Some pins serve as bus control pins <sup>(1)</sup>
Microprocessor mode	SFR, internal RAM, external area <sup>(1)</sup>	Some pins serve as bus control pins <sup>(1)</sup>

NOTE:

1. Refer to 8. "Bus" for details.

### 7.2 Setting Processor Modes

Processor mode is set by using the CNVSS pin and bits PM01 to PM00 in the PM0 register.

Table 7.2 shows the Processor Mode After Hardware Reset. Table 7.3 shows Bits PM01 to PM00 Set Values and Processor Modes

**Table 7.2 Processor Mode After Hardware Reset**

CNVSS Pin Input Level	Processor Modes
VSS	Single-chip mode
VCC1 <sup>(1, 2)</sup>	Microprocessor mode

NOTES:

1. If the microcomputer is reset in hardware by applying VCC1 to the CNVSS pin (hardware reset 1 or brown-out reset), the internal ROM cannot be accessed regardless of the status of bits PM10 to PM00.
2. The multiplexed bus cannot be assigned to the entire  $\overline{CS}$  space.

**Table 7.3 Bits PM01 to PM00 Set Values and Processor Modes**

Bits PM01 to PM00	Processor Modes
00b	Single-chip mode
01b	Memory expansion mode
10b	Do not set
11b	Microprocessor mode

Rewriting bits PM01 to PM00 places the microcomputer in the corresponding processor mode regardless of whether the input level on the CNVSS pin is "H" or "L". Note, however, that bits PM01 to PM00 cannot be rewritten to 01b (memory expansion mode) or 11b (microprocessor mode) at the same time bits PM07 to PM02 are rewritten. Note also that these bits cannot be rewritten to enter microprocessor mode in the internal ROM, nor can they be rewritten to exit microprocessor mode in areas overlapping the internal ROM.

If the microcomputer is reset in hardware by applying VCC1 to the CNVSS pin (hardware reset 1 or brown-out reset), the internal ROM cannot be accessed regardless of bits PM01 to PM00.

Figures 7.1 to 7.3 show the PM0 Register and PM1 Register. Figure 7.4 show the Memory Map in Single-Chip Mode.

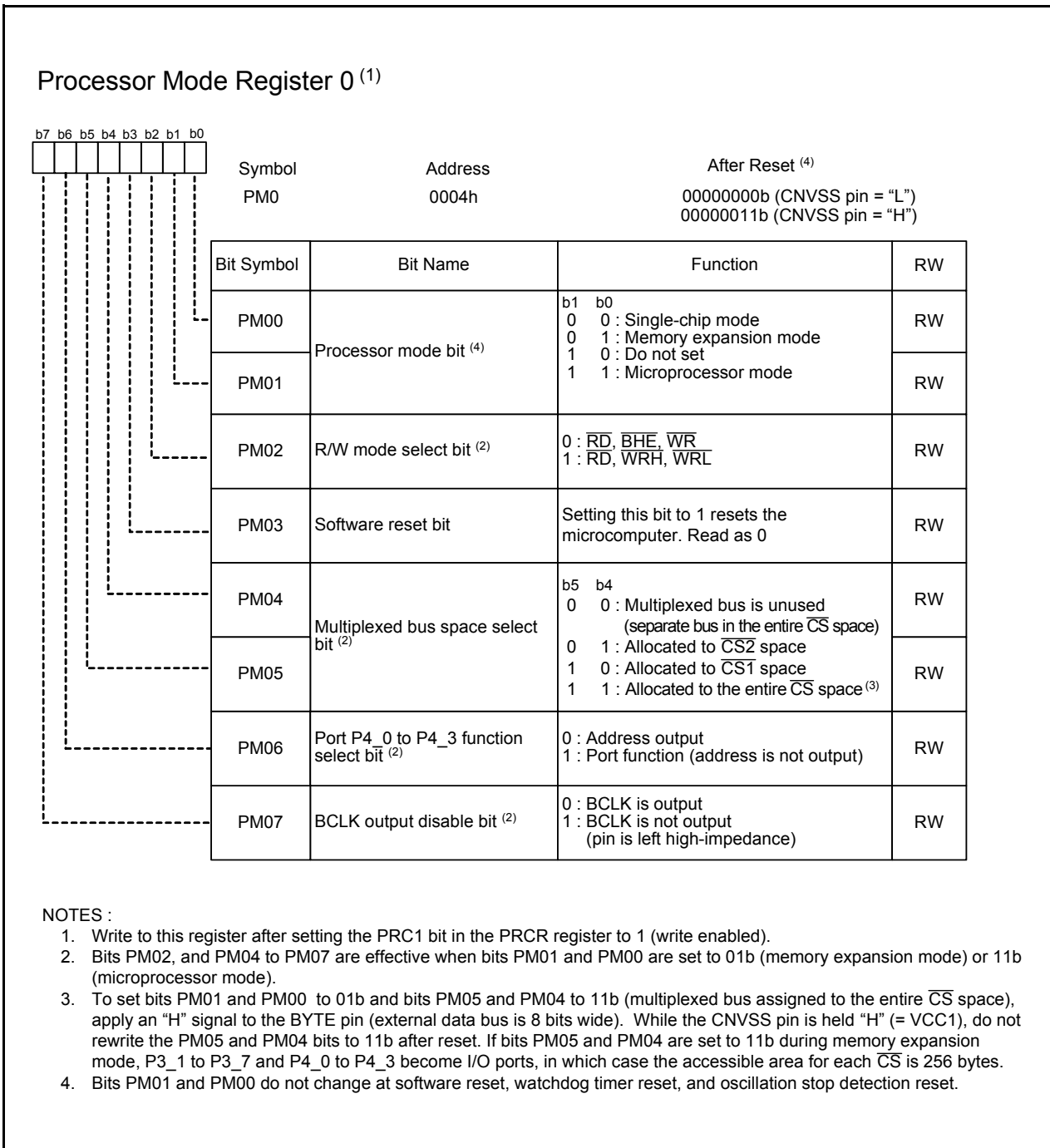


Figure 7.1 PM0 Register

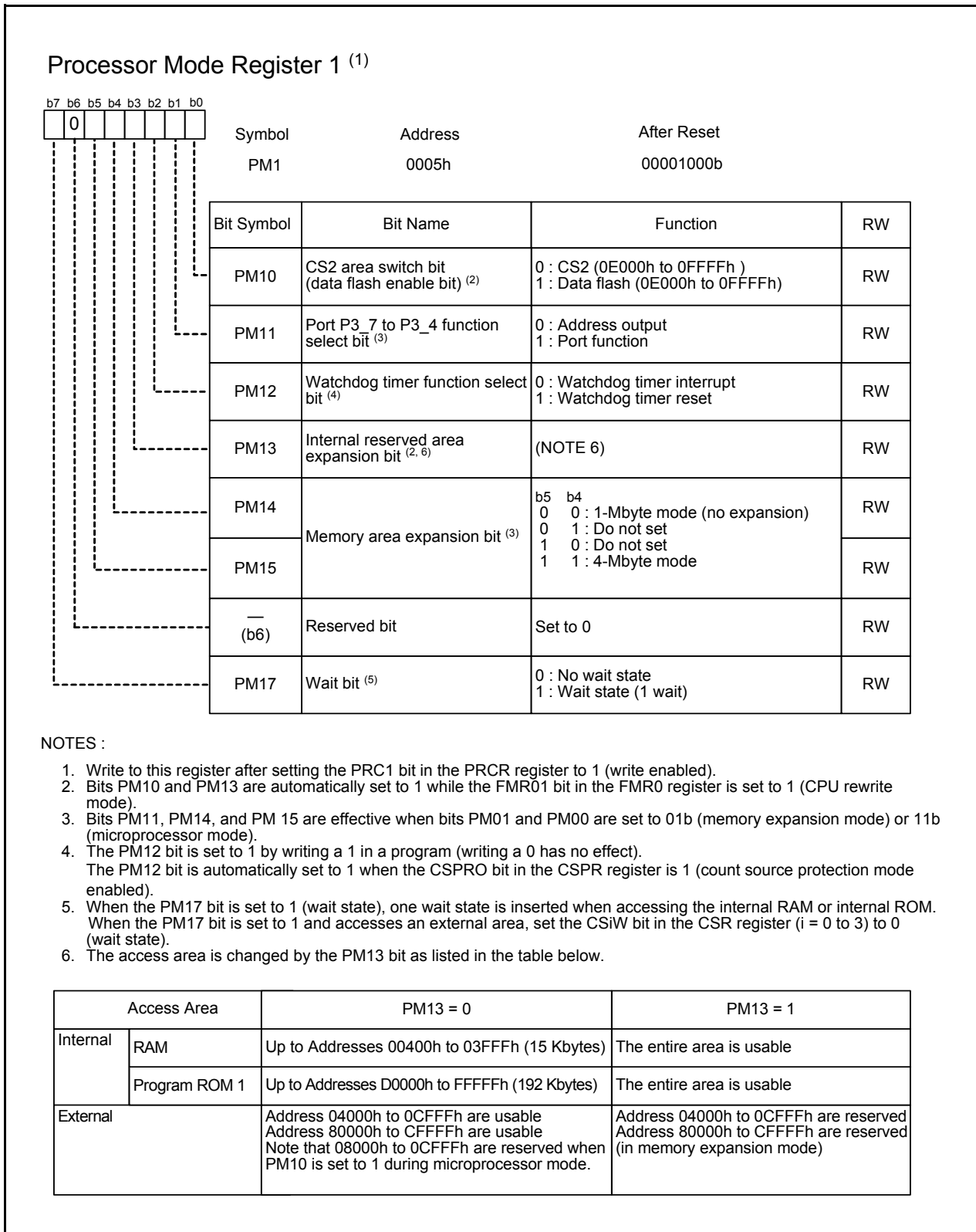


Figure 7.2 PM1 Register

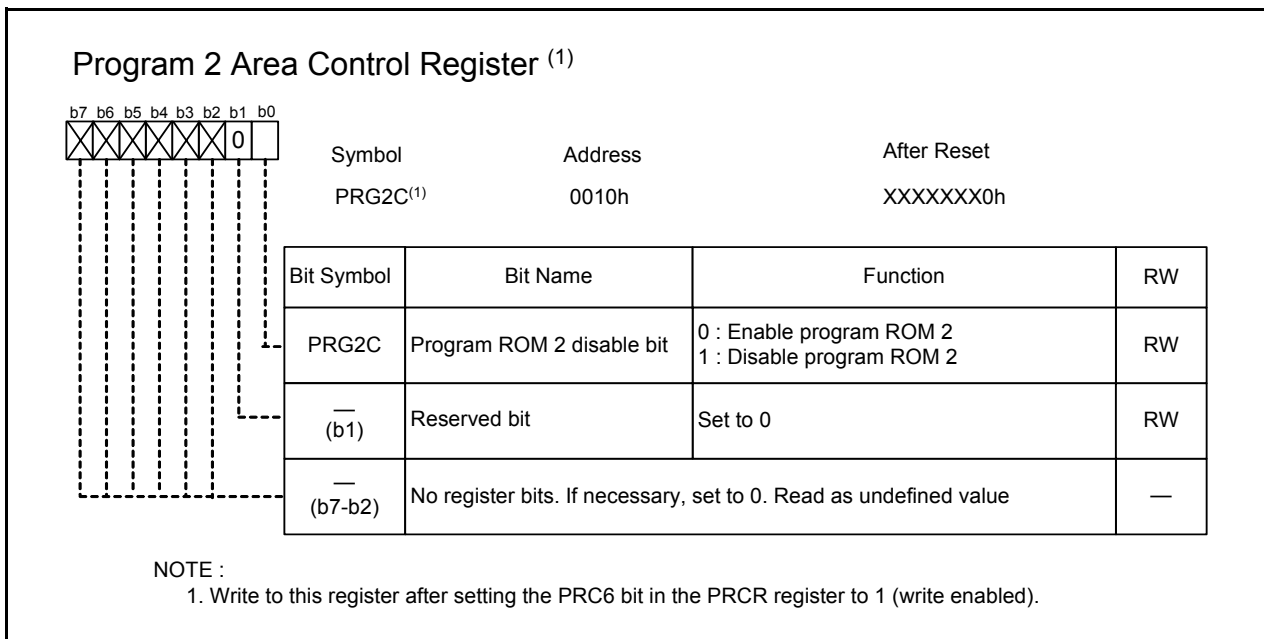


Figure 7.3 PRG2C Register

### 7.3 Internal Memory

The internal RAM can be used in all processor modes. The range of the internal RAM depends on the setting of the PM 13 bit in the PM1 register.

The internal ROM is used in single-chip mode and memory expansion mode. Three internal ROMs are available: data flash, program ROM 2, and program ROM 1.

Data flash includes block A (addresses 0E000h to 0EFFFh) and block B (addresses 0F000h to 0FFFFh). When data flash is selected by the setting of the PM10 bit in the PM1 register, both block A and block B can be used. Table 7.4 shows Data Flash (addresses 0E000h to 0FFFFh).

**Table 7.4 Data Flash (addresses 0E000h to 0FFFFh)**

PM10 Bit in PM1 Register		0	1
Processor Modes	Single-chip mode	Unusable	Data flash
	Memory expansion mode	External area	Data flash
	Microprocessor mode	External area	Reserved area

Set the PRG2C0 bit in the PRG2C register to select program ROM 2. Table 7.5 shows Program ROM 2 (addresses 10000h to 13FFFh).

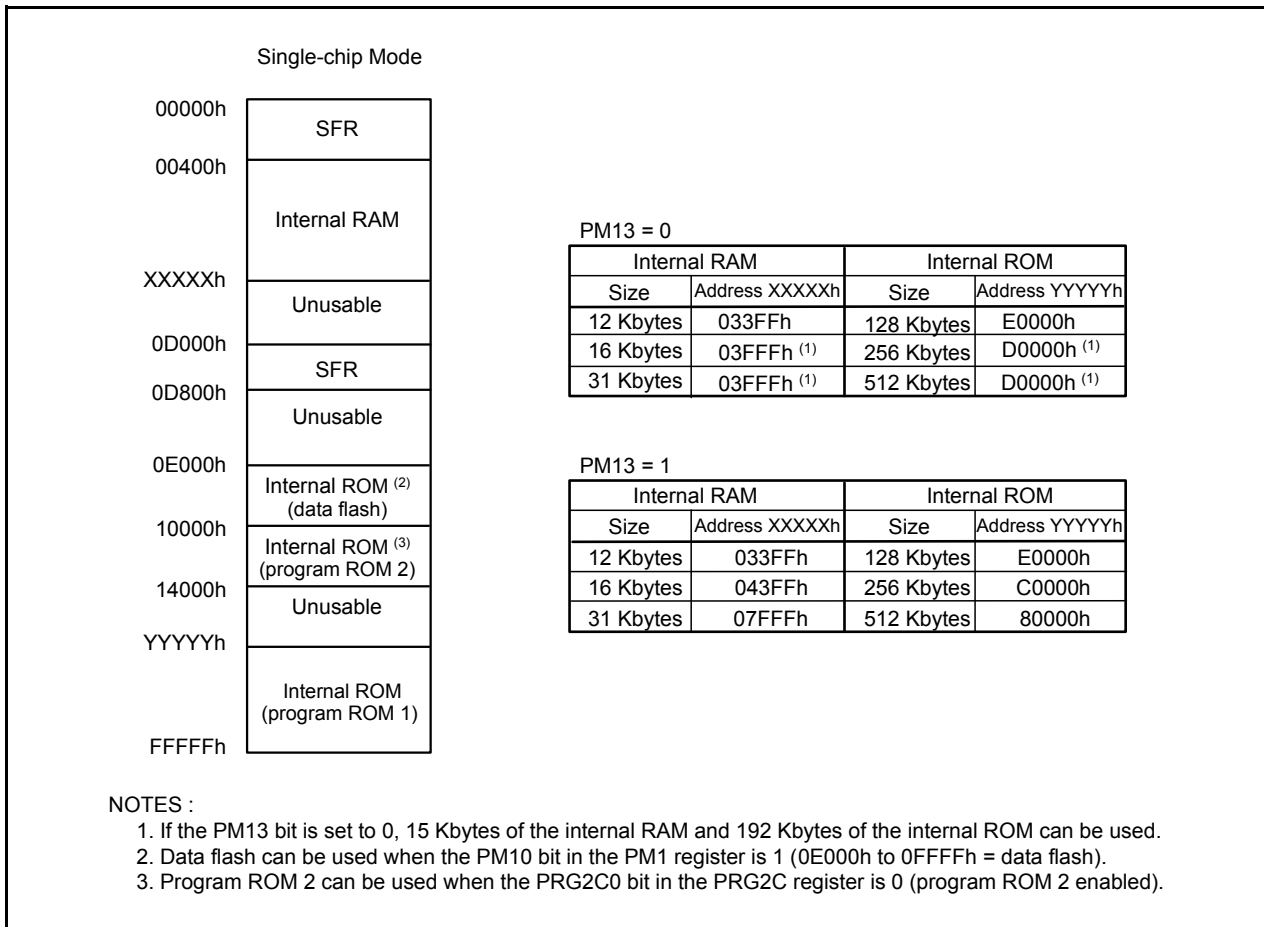
Do not use the last 16 bytes (addresses 13FF0h to 13FFFh) when using program ROM 2 in single-chip mode or memory expansion mode. These bytes are assigned as the user boot code area (refer to **22.1.2 “User Boot Function”**).

**Table 7.5 Program ROM 2 (addresses 10000h to 13FFFh)**

PRG2C0 bit in PRG2C Register		0	1
Processor Modes	Single-chip mode	Program ROM 2	Unusable
	Memory expansion mode	Program ROM 2	External area
	Microprocessor mode	Reserved area	External area

The range of program ROM 1 depends on the setting of the PM13 bit in the PM1 register. Figure 7.4 indicates the Memory Map in Single-Chip Mode.





**Figure 7.4 Memory Map in Single-Chip Mode**

## 8. Bus

During memory expansion or microprocessor mode, some pins serve as the bus control pins to perform data input /output to and from external devices. These bus control pins include A0 to A19, D0 to D15,  $\overline{CS0}$  to  $\overline{CS3}$ ,  $\overline{RD}$ ,  $\overline{WRL}$  /  $\overline{WR}$ ,  $\overline{WRH}$  /  $\overline{BHE}$ ,  $\overline{ALE}$ ,  $\overline{RDY}$ ,  $\overline{HOLD}$ ,  $\overline{HLDA}$ , and  $\overline{BCLK}$ .

### 8.1 Bus Mode

Bus mode, either multiplexed or separate, can be selected using bits PM05 and PM04 in the PM0 register. Table 8.1 shows the Difference between Separate Bus and Multiplexed Bus.

#### 8.1.1 Separate Bus

In this bus mode, data and address are separate.

#### 8.1.2 Multiplexed Bus

In this bus mode, data and address are multiplexed.

##### 8.1.2.1 When the Input Level on BYTE Pin is High (8-Bit Data Bus)

D0 to D7 and A0 to A7 are multiplexed.

##### 8.1.2.2 When the Input Level on BYTE Pin is Low (16-Bit Data Bus)

D0 to D7 and A1 to A8 are multiplexed. D8 to D15 are not multiplexed. Do not use D8 to D15. External devices connecting to a multiplexed bus are allocated to only the even addresses of the micro-computer. Odd addresses cannot be accessed.

**Table 8.1 Difference between Separate Bus and Multiplexed Bus**

Pin Name (1)	Separate Bus	Multiplexed Bus	
		BYTE = "H"	BYTE = "L"
P0_0 to P0_7 / D0 to D7		(NOTE 2)	(NOTE 2)
P1_0 to P1_7 / D8 to D15		I/O port P1_0 to P1_7	(NOTE 2)
P2_0 / A0 (/ D0 / -)			
P2_1 to P2_7 / A1 to A7 (/ D1 to D7 / D0 to D6)			
P3_0 / A8 (/ - / D7)			

NOTES:

1. See **Table 8.6 "Pin Functions for Each Processor Mode"** for bus control signals other than the above.
2. It changes with a setting of PM05 and PM04, and area to access.  
See **Table 8.6 "Pin Functions for Each Processor Mode"** for details.

## 8.2 Bus Control

The following describes the signals needed for accessing external devices and the functionality of software wait.

### 8.2.1 Address Bus

The address bus consists of 20 lines: A0 to A19. The address bus width can be chosen to be 12, 16, or 20 bits by using the PM06 bit in the PM0 register and the PM11 bit in the PM1 register. Table 8.2 shows the Set Value of Bits PM06 and PM11, and Address Bus Width.

**Table 8.2 Set Value of Bits PM06 and PM11, and Address Bus Width**

Bits Set Value (1)	Pin Function	Address Bus Width
PM11 = 1	P3_4 to P3_7	12 bits
PM06 = 1	P4_0 to P4_3	
PM11 = 0	A12 to A15	16 bits
PM06 = 1	P4_0 to P4_3	
PM11 = 0	A12 to A15	20 bits
PM06 = 0	A16 to A19	

NOTE:

1. No values other than those shown above can be set.

When processor mode is changed from single-chip mode to memory expansion mode, the address bus is indeterminate until any external area is accessed.

### 8.2.2 Data Bus

When input on the BYTE pin is high (data bus is 8 bits wide), 8 lines D0 to D7 comprise the data bus; when input on the BYTE pin is low (data bus is 16 bits wide), 16 lines D0 to D15 comprise the data bus. Do not change the input level on the BYTE pin while in operation.

### 8.2.3 Chip Select Signal

The chip select (hereafter referred to as the  $\overline{CS}$ ) signals are output from the  $\overline{CS}_i$  ( $i = 0$  to 3) pins. These pins can be chosen to function as I/O ports or as  $\overline{CS}$  by using the  $\overline{CS}_i$  bit in the CSR register. Figure 8.1 shows the CSR Register.

During 1-Mbyte mode, the external area can be separated into up to 4 by the  $\overline{CS}_i$  signal which is output from the  $\overline{CS}_i$  pin. During 4-Mbyte mode,  $\overline{CS}_i$  signal or bank number is output from the  $\overline{CS}_i$  pin. Refer to **9. "Memory Space Expansion Function"**. Figure 8.2 shows Examples of Address Bus and  $\overline{CS}_i$  Signal Output in 1-Mbyte mode.

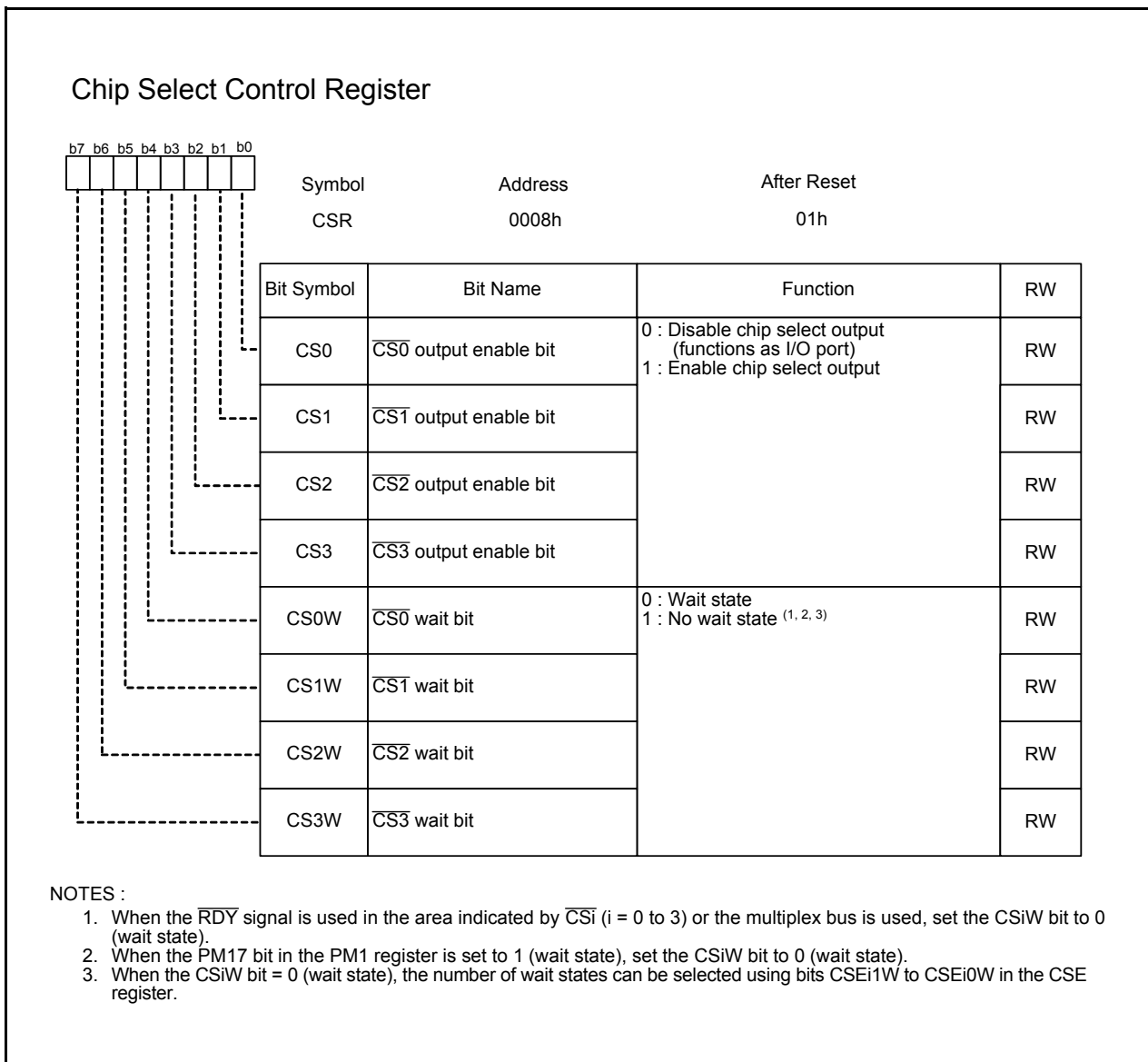
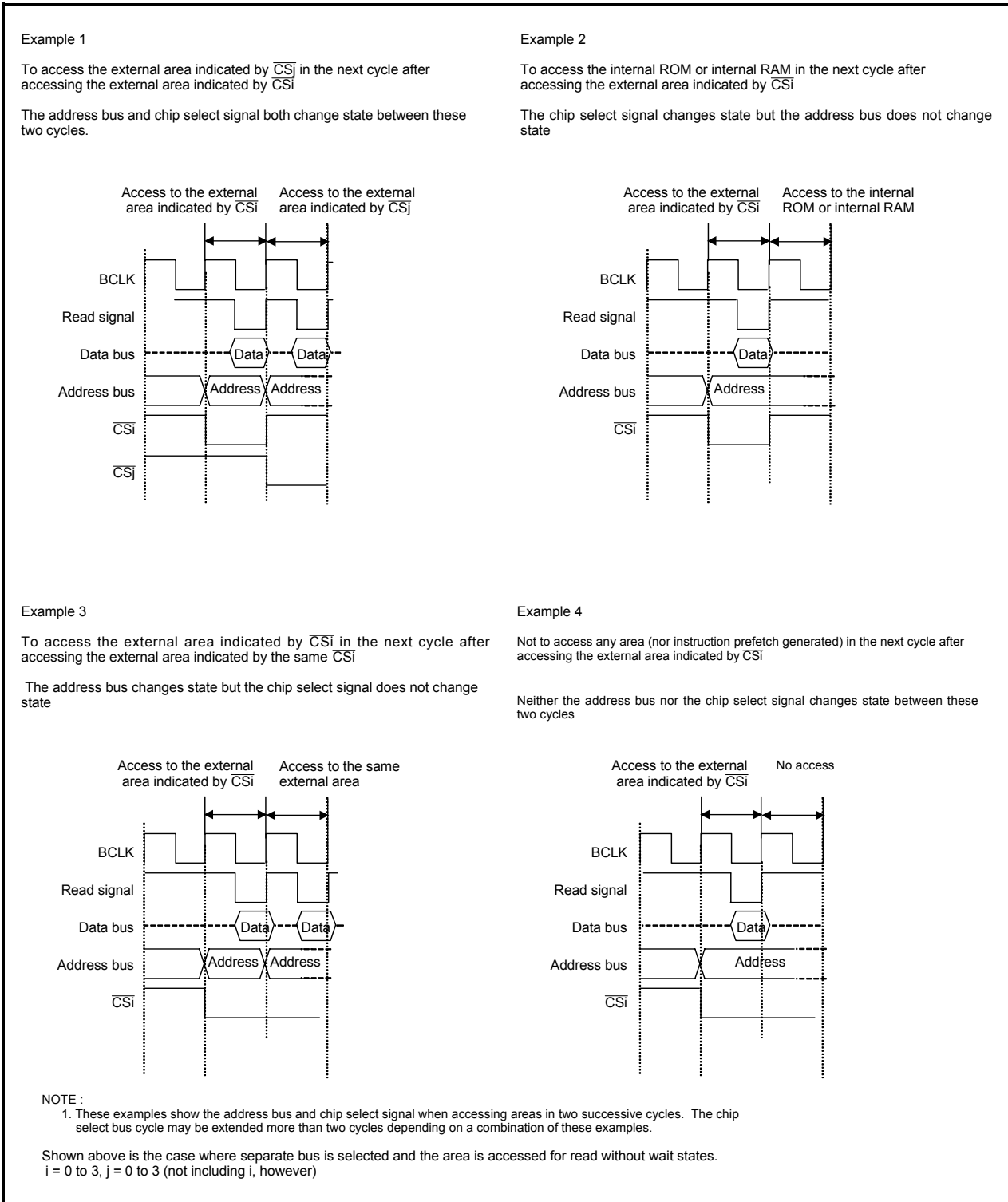


Figure 8.1 CSR Register



**Figure 8.2 Examples of Address Bus and  $\overline{CS}_i$  Signal Output in 1-Mbyte Mode**

### 8.2.4 Read and Write Signals

When the data bus is 16 bits wide, the read and write signals can be chosen to be a combination of  $\overline{RD}$ ,  $\overline{BHE}$ , and  $\overline{WR}$  or a combination of  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  by using the PM02 bit in the PM0 register. When the data bus is 8 bits wide, use a combination of  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{BHE}$ .

Table 8.3 shows the Operation of  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  Signals. Table 8.4 shows the Operation of  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{BHE}$  Signals.

**Table 8.3 Operation of  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  Signals**

Data Bus Width	$\overline{RD}$	$\overline{WRL}$	$\overline{WRH}$	Status of External Data Bus
16-bit (BYTE pin input = "L")	L	H	H	Read data
	H	L	H	Write 1 byte of data to an even address
	H	H	L	Write 1 byte of data to an odd address
	H	L	L	Write data to both even and odd addresses

**Table 8.4 Operation of  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{BHE}$  Signals**

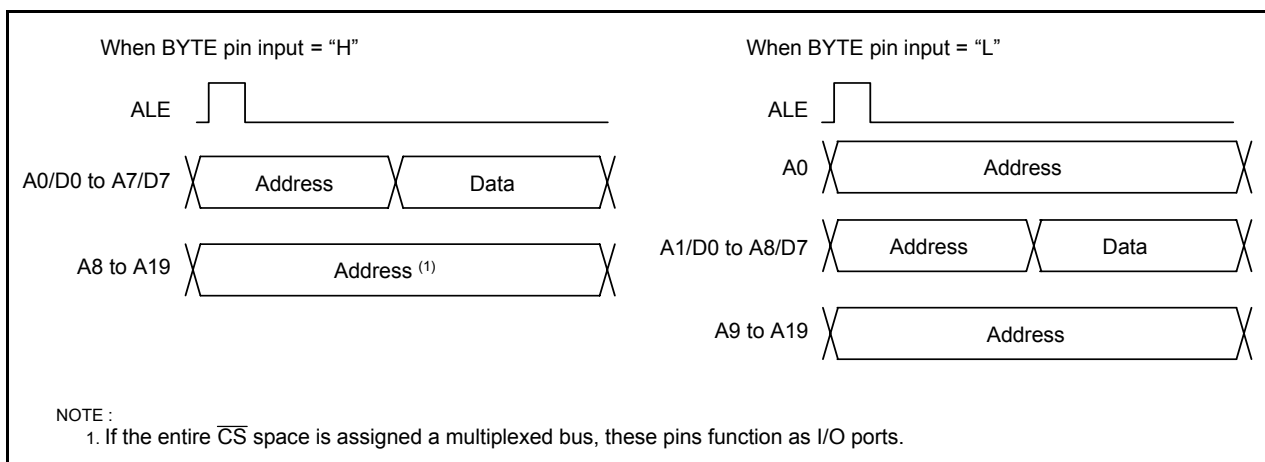
Data Bus Width	$\overline{RD}$	$\overline{WR}$	$\overline{BHE}$	A0	Status of External Data Bus
16-bit (BYTE pin input = "L")	H	L	L	H	Write 1 byte of data to an odd address
	L	H	L	H	Read 1 byte of data from an odd address
	H	L	H	L	Write 1 byte of data to an even address
	L	H	H	L	Read 1 byte of data from an even address
	H	L	L	L	Write data to both even and odd addresses
	L	H	L	L	Read data from both even and odd addresses
8-bit (BYTE pin input = "H")	H	L	– (1)	H or L	Write 1 byte of data
	L	H	– (1)	H or L	Read 1 byte of data

NOTE:

1. Do not use.

### 8.2.5 ALE Signal

The ALE signal latches the address when accessing the multiplexed bus space. Latch the address when the ALE signal falls.



NOTE :

1. If the entire  $\overline{CS}$  space is assigned a multiplexed bus, these pins function as I/O ports.

**Figure 8.3 ALE Signal, Address Bus, Data Bus**

### 8.2.6 $\overline{\text{RDY}}$ Signal

This signal is provided for accessing external devices which need to be accessed at low speed. If input on the  $\overline{\text{RDY}}$  pin is low at the last falling edge of BCLK of the bus cycle, one wait state is inserted in the bus cycle. While in a wait state, the following signals retain the state in which they were when the  $\overline{\text{RDY}}$  signal was acknowledged.

A0 to A19, D0 to D15,  $\overline{\text{CS}}_0$  to  $\overline{\text{CS}}_3$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{WRL}}$ ,  $\overline{\text{WRH}}$ ,  $\overline{\text{WR}}$ ,  $\overline{\text{BHE}}$ , ALE,  $\overline{\text{HLDA}}$

Then, when the input on the  $\overline{\text{RDY}}$  pin is detected high at the falling edge of BCLK, the remaining bus cycle is executed. Figure 8.4 shows Examples in which the Wait State was Inserted into Read Cycle by  $\overline{\text{RDY}}$  Signal. To use the  $\overline{\text{RDY}}$  signal, set the corresponding bit (bits CS3W to CS0W) in the CSR register to 0 (with wait state). When not using the  $\overline{\text{RDY}}$  signal, the  $\overline{\text{RDY}}$  pin need to be pulled-up.

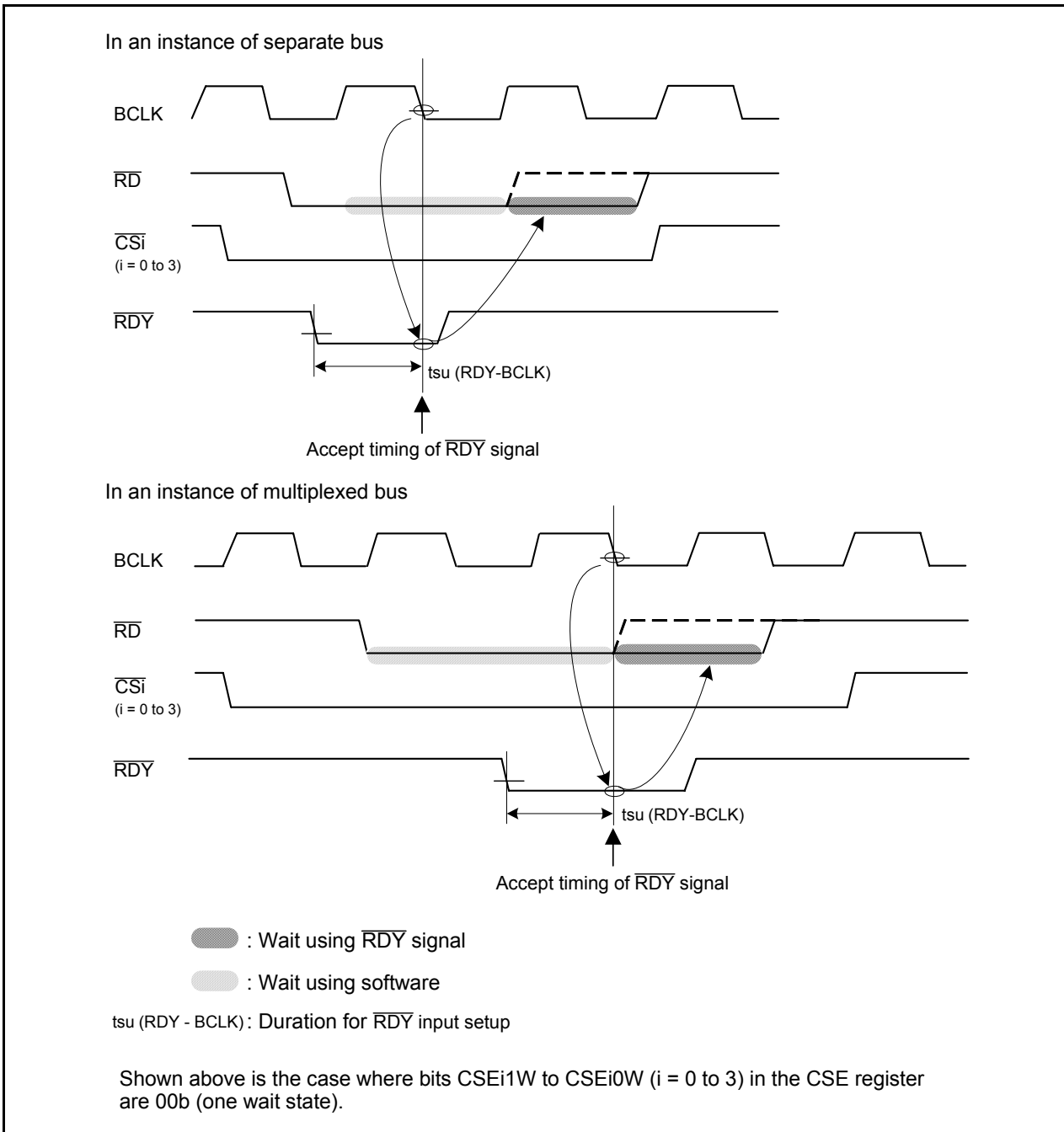


Figure 8.4 Examples in Which Wait State Was Inserted into Read Cycle by  $\overline{\text{RDY}}$  Signal

### 8.2.7 $\overline{\text{HOLD}}$ Signal

This signal is used to transfer control of the bus from the CPU or DMAC to an external circuit. When the input on the  $\overline{\text{HOLD}}$  pin is pulled low, the microcomputer is placed in a hold state after the bus access at that time finishes. The microcomputer remains in the hold state while the  $\overline{\text{HOLD}}$  pin is held low, during which time the  $\overline{\text{HLDA}}$  pin outputs a low-level signal.

Table 8.5 shows the Microcomputer Status in Hold State.

Bus-using priorities are given to  $\overline{\text{HOLD}}$ , DMAC, and CPU in descending order. However, if the CPU is accessing an odd address in word units, the DMAC cannot gain control of the bus during two separate accesses.

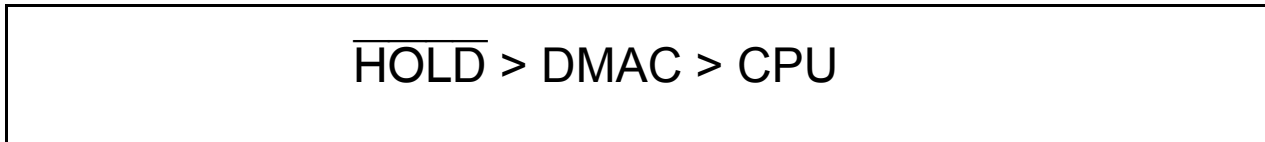


Figure 8.5 Bus-Using Priorities

Table 8.5 Microcomputer Status in Hold State

Item		Status
BCLK		Output
A0 to A19, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ , $\overline{\text{RD}}$ , $\overline{\text{WRL}}$ , $\overline{\text{WRH}}$ , $\overline{\text{WR}}$ , $\overline{\text{BHE}}$		High-impedance
I/O ports	P0, P1, P3, P4 (1)	High-impedance
	P6 to P10	Maintains status when $\overline{\text{HOLD}}$ signal is received
$\overline{\text{HLDA}}$		Output "L"
Internal Peripheral Circuits		On (but watchdog timer stops) (2)
ALE		Undefined

NOTES:

- When I/O port function is selected.
- The watchdog timer does not stop when the CSPRO bit in the CSPR register is set to 1 (count source protection mode enabled).

### 8.2.8 BCLK Output

If the PM07 bit in the PM0 register is set to 0 (output enabled), a clock with the same frequency as that of the CPU clock is output as BCLK from the BCLK pin. Refer to 10.2 "CPU Clock and Peripheral Function Clock".



**Table 8.6 Pin Functions for Each Processor Mode**

Processor Mode	Memory Expansion Mode or Microprocessor Mode				Memory Expansion Mode
Bits PM05 and PM04	00b (separate bus)		01b (CS <sub>2</sub> is for multiplexed bus and the others are for separate bus) 10b (CS <sub>1</sub> is for multiplexed bus and the others are for separate bus)		11b (multiplexed bus for the entire space) (1)
Data Bus Width BYTE Pin	8 bits "H"	16 bits "L"	8 bits "H"	16 bits "L"	8 bits "H"
P0_0 to P0_7	D0 to D7	D0 to D7	D0 to D7 (4)	D0 to D7 (4)	I/O ports
P1_0 to P1_7	I/O ports	D8 to D15	I/O ports	D8 to D15(4)	I/O ports
P2_0	A0	A0	A0/D0 (2)	A0	A0/D0
P2_1 to P2_7	A1 to A7	A1 to A7	A1 to A7 /D1 to D7 (2)	A1 to A7 /D0 to D6 (2)	A1 to A7 /D1 to D7
P3_0	A8	A8	A8	A8/D7 (2)	A8
P3_1 to P3_3	A9 to A11				I/O ports
P3_4 to P3_7	PM11 = 0	A12 to A15			I/O ports
	PM11 = 1	I/O ports			
P4_0 to P4_3	PM06 = 0	A16 to A19			I/O ports
	PM06 = 1	I/O ports			
P4_4	CS0 = 0	I/O ports			
	CS0 = 1	$\overline{CS0}$			
P4_5	CS1 = 0	I/O ports			
	CS1 = 1	$\overline{CS1}$			
P4_6	CS2 = 0	I/O ports			
	CS2 = 1	$\overline{CS2}$			
P4_7	CS3 = 0	I/O ports			
	CS3 = 1	$\overline{CS3}$			
P5_0	PM02 = 0	$\overline{WR}$			
	PM02 = 1	- (3)	$\overline{WRL}$	- (3)	$\overline{WRL}$ - (3)
P5_1	PM02 = 0	$\overline{BHE}$			
	PM02 = 1	- (3)	$\overline{WRH}$	- (3)	$\overline{WRH}$ - (3)
P5_2	$\overline{RD}$				
P5_3	BCLK				
P5_4	$\overline{HLDA}$				
P5_5	$\overline{HOLD}$				
P5_6	ALE				
P5_7	$\overline{RDY}$				

I/O ports : Function as I/O ports or peripheral function I/O pins.

NOTES:

- When bits PM01 and PM00 are set to 01b (memory expansion mode) and bits PM05 and PM04 are set to 11b (multiplexed bus assigned to the entire  $\overline{CS}$  space), apply "H" to the BYTE pin (external data bus 8 bits wide). While the CNVSS pin is held "H" (= VCC1), do not rewrite bits PM05 and PM04 to 11b after reset. If bits PM05 and PM04 are set to 11b during memory expansion mode, P3\_1 to P3\_7 and P4\_0 to P4\_3 become I/O ports, in which case the accessible area for each  $\overline{CS}$  is 256 bytes.
- In separate bus mode, these pins serve as the address bus.
- If the data bus is 8 bits wide, make sure the PM02 bit is set to 0 ( $\overline{RD}$ ,  $\overline{BHE}$ ,  $\overline{WR}$ ).
- When accessing the area that uses a multiplexed bus, these pins output an indeterminate value during a write.

### 8.2.9 External Bus Status When Internal Area IS Accessed

Table 8.7 shows the External Bus Status When Internal Area Accessed.

**Table 8.7 External Bus Status When Internal Area Accessed**

Item	SFR Accessed	Internal ROM, RAM Accessed
A0 to A19	Address output	Maintain the last accessed address of external area or SFR
D0 to D15	When Read	High-impedance
	When Write	Output data
$\overline{RD}$ , $\overline{WR}$ , $\overline{WRL}$ , $\overline{WRH}$	$\overline{RD}$ , $\overline{WR}$ , $\overline{WRL}$ , $\overline{WRH}$ output	Output "H"
BHE	$\overline{BHE}$ output	Maintain the last accessed status of external area or SFR
$\overline{CS0}$ to $\overline{CS3}$	Output "H"	Output "H"
ALE	Output "L"	Output "L"

### 8.2.10 Software Wait

The PM17 bit in the PM1 register, which is a software-wait-related bit, has an influence over the internal memory and the external area.

The SFR area is accessed in 2 BCLK or 3 BCLK cycles as determined by the PM20 bit in the PM2 register. Data flash, one of the internal ROMs, is affected by the PM17 bit and the FMR17 bit in the FMR1 register. See **Table 8.8 "Bits and Bus Cycles Related to Software Wait (SFR, Internal Memory)"** for details.

Software waits can be inserted to the external area by setting the PM17 bit or setting the CSiW bit in the CSR register or bits CSEi1W to CSEi0W in the CSE register for each CSi (i = 0 to 3). To use the  $\overline{RDY}$  signal, set the corresponding CSiW bit to 0 (with wait state). Refer to **Table 8.9 "Bits and Bus Cycles Related to Software Wait (External Area)"** for details.

Table 8.8 shows the Bits and Bus Cycles Related to Software Wait (SFR, Internal Memory), Figure 8.6 shows the CSE register, and Table 8.9 shows the Bits and Bus Cycles Related to Software Wait (External Area).

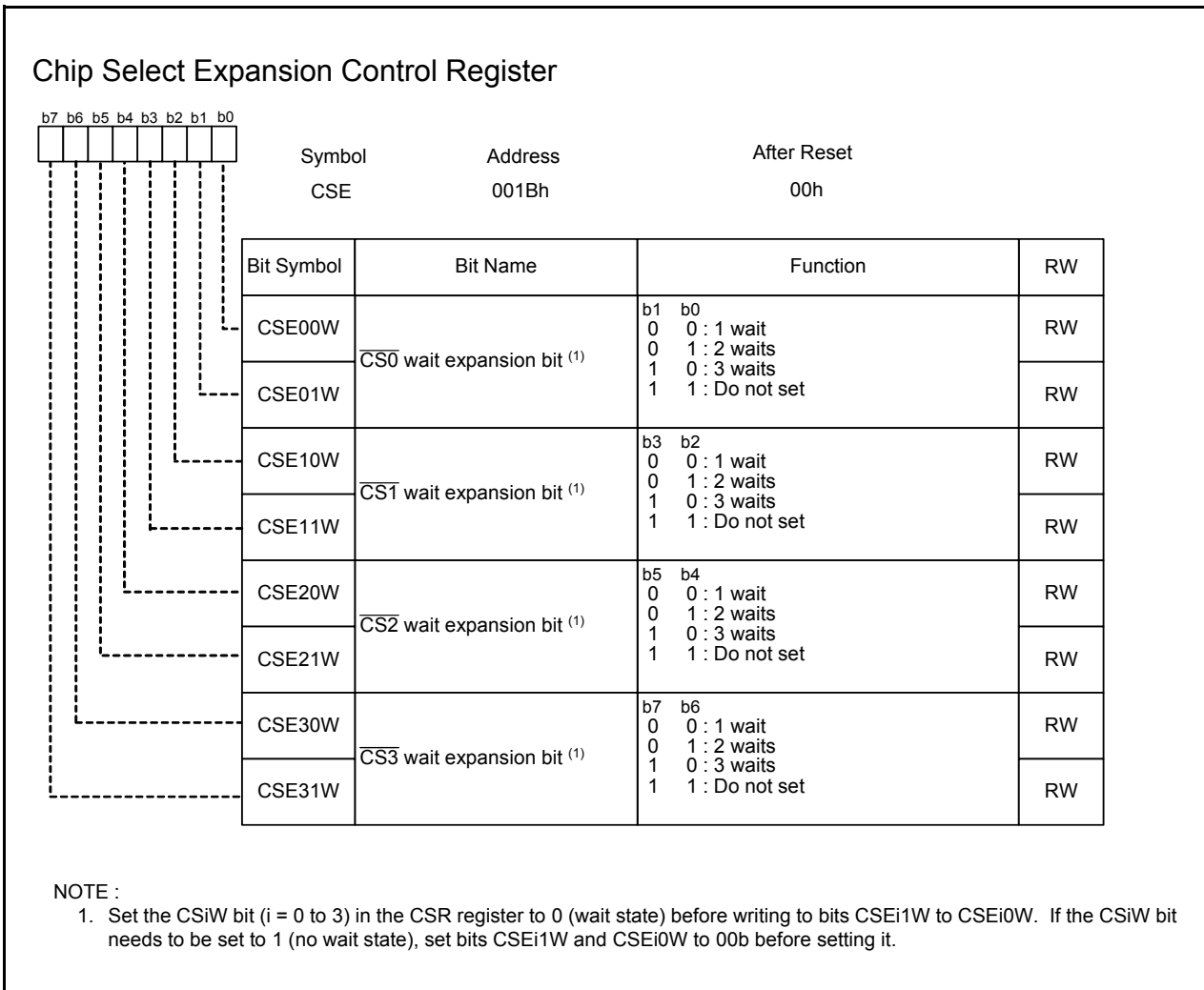
**Table 8.8 Bits and Bus Cycles Related to Software Wait (SFR, Internal Memory)**

Area	Setting of Software-Wait-related Bits			Software Wait	Bus cycle
	PM2 Register PM20 Bit (1)	FMR1 Register FMR17 Bit	PM1 Register PM17 Bit		
SFR	1	-	-	1 wait	2 BCLK cycles <sup>(3)</sup>
	0	-	-	2 waits	3 BCLK cycles
Internal RAM	-	-	0	No wait	1 BCLK cycle <sup>(3)</sup>
			1	1 wait	2 BCLK cycles
Internal ROM	Program ROM 1 Program ROM 2	-	0	No wait	1 BCLK cycle <sup>(3)</sup>
			1	1 wait	2 BCLK cycles
	Data Flash (2)	-	0	1 wait	2 BCLK cycles <sup>(3)</sup>
			1	0	No wait
			1	1 wait	2 BCLK cycle

- indicates that either 0 or 1 can be set.

**NOTES:**

- The PM 20 bit is valid when the PLC07 bit in the PLC0 register is set to 1 (PLL operation).
- When  $2.7\text{ V} \leq VCC1 \leq 3.0\text{ V}$  and  $f(\text{BCLK}) \geq 16\text{ MHz}$ , or when  $3.0\text{ V} < VCC1 \leq 5.5\text{ V}$  and  $f(\text{BCLK}) \geq 20\text{ MHz}$ , 1 wait is necessary to read data flash. Use the PM17 bit or the FMR 17 bit to set 1 wait.
- Status after reset.



**Figure 8.6 CSE Register**

**Table 8.9 Bits and Bus Cycles Related to Software Wait (External Area)**

Area	Bus Mode	Setting of Software-Wait-related Bits			Software Wait	Bus Cycle
		PM1 Register PM17 Bit	CSR Register CSiW Bit (1)	CSE Register Bits CSEi1W to CSEi0W		
External Area	Separate Bus	0	1	00b	No wait	1 BCLK cycle (Read) 2 BCLK cycles (Write)
		-	0	00b	1 wait	2 BCLK cycles (4)
		-	0	01b	2 waits	3 BCLK cycles
		-	0	10b	3 waits	4 BCLK cycles
		1	0 (3)	00b	1 wait	2 BCLK cycles
	Multiplexed Bus	-	0 (2)	00b	1 wait	3 BCLK cycles
		-	0 (2)	01b	2 waits	3 BCLK cycles
		-	0 (2)	10b	3 waits	4 BCLK cycles
		1	0 (2, 3)	00b	1 wait	3 BCLK cycles

i = 0 to 3

- indicates that either 0 or 1 can be set.

**NOTES:**

1. To use the  $\overline{\text{RDY}}$  signal, set the CSiW bit to 0 (with wait state).
2. To access in multiplexed bus mode, set the CSiW bit to 0 (with wait state).
3. When the PM17 bit is set to 1 and accesses an external area, set the CSiW bit to 0 (with wait state).
4. After reset, the PM17 bit is set to 0 (without wait state), bits CS0W to CS3W are set to 0 (with wait state), and the CSE register is set to 00h (one wait state for  $\overline{\text{CS0}}$  to  $\overline{\text{CS3}}$ ). Therefore, all external areas are accessed with one wait state.

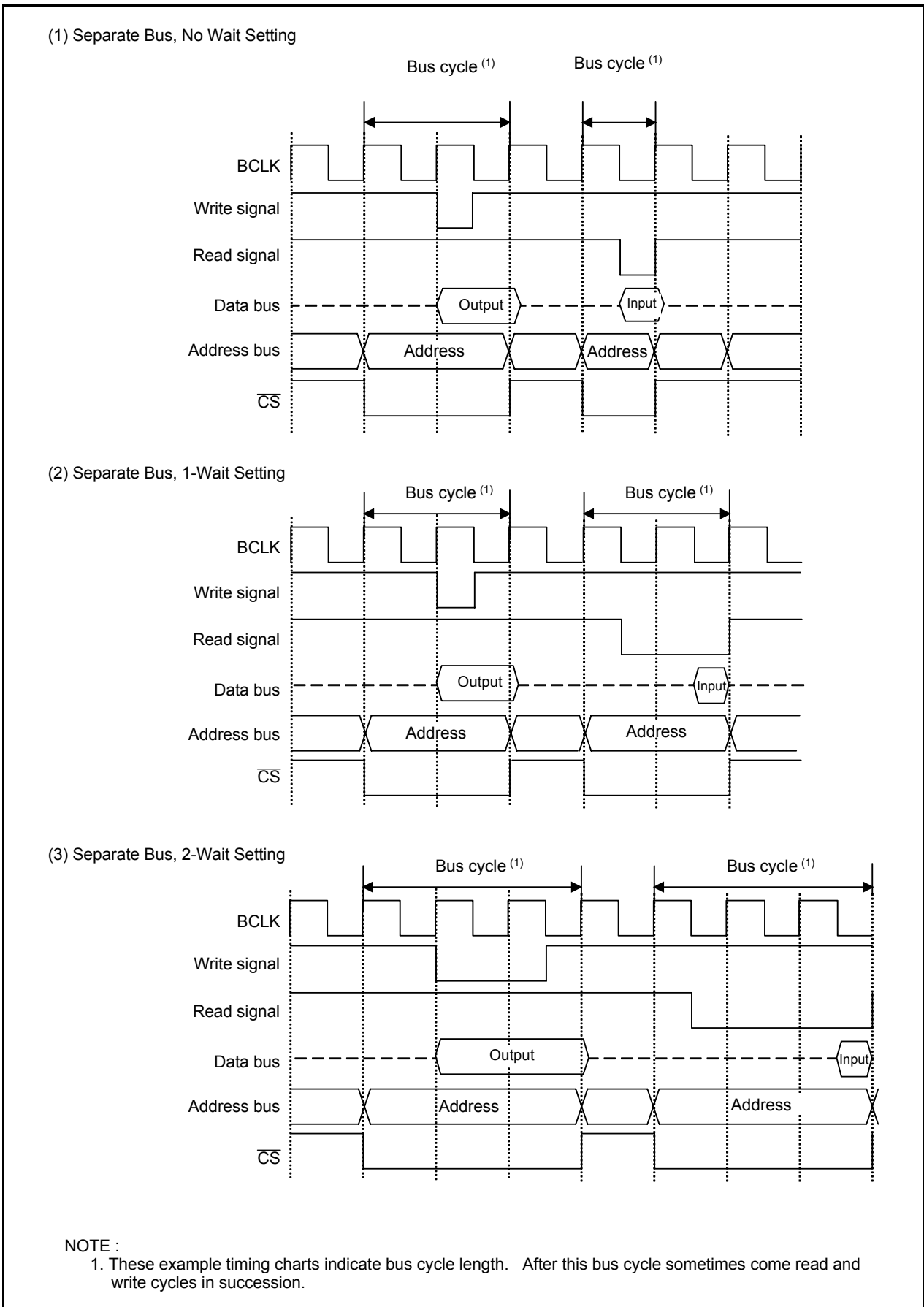
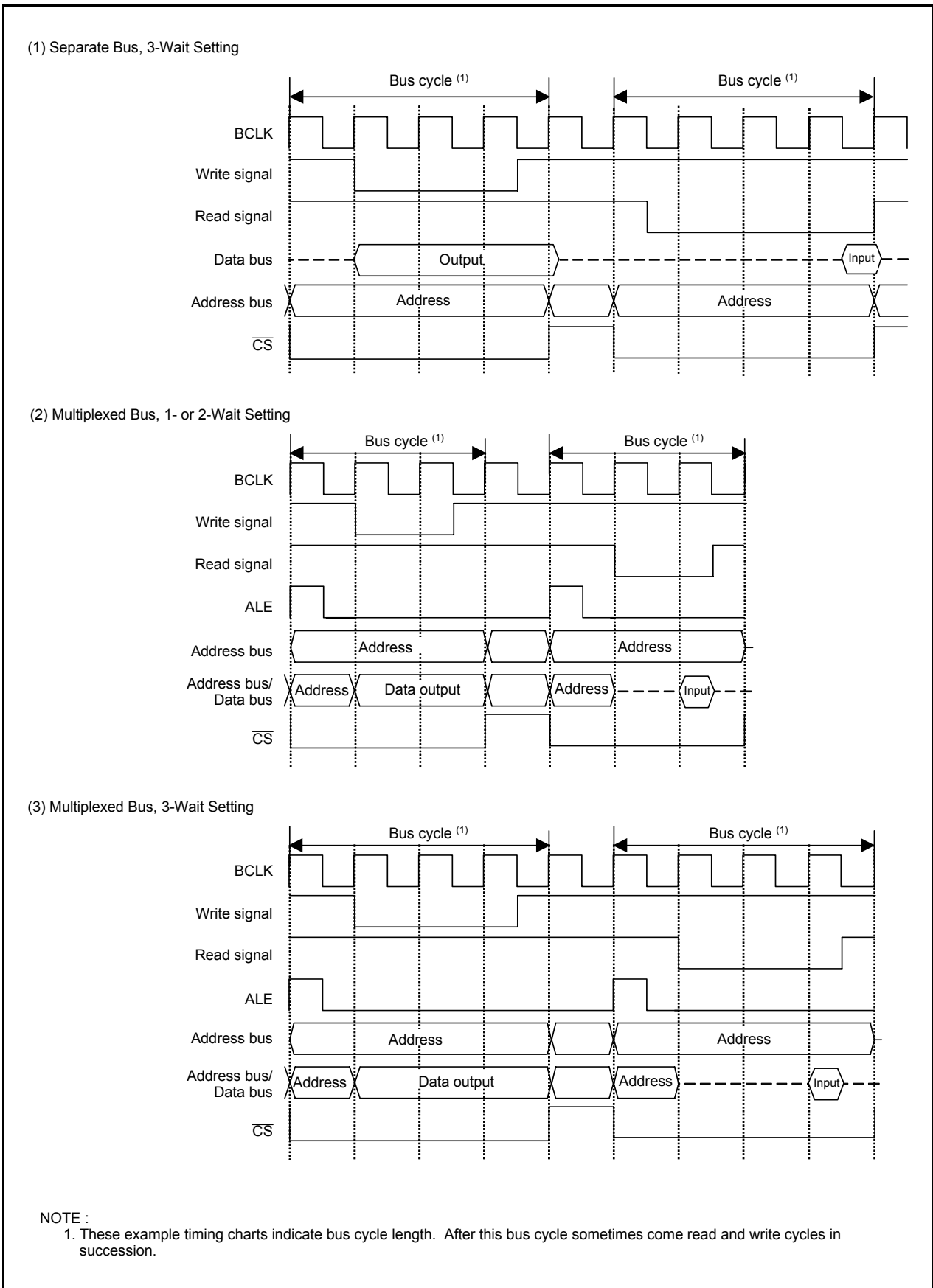


Figure 8.7 Typical Bus Timings Using Software Wait (1)



NOTE :

1. These example timing charts indicate bus cycle length. After this bus cycle sometimes come read and write cycles in succession.

Figure 8.8 Typical Bus Timings Using Software Wait (2)

## 9. Memory Space Expansion Function

The following describes a memory space expansion function.

During memory expansion or microprocessor mode, the memory space expansion function allows the access space to be expanded using the appropriate register bits.

Table 9.1 shows Setting of Memory Space Expansion Function, Memory Space.

**Table 9.1 Setting of Memory Space Expansion Function, Memory Space**

Memory Space Expansion Function	Setting (PM15 to PM14)	Memory Space
1-Mbyte Mode	00b	1 Mbyte (no expansion)
4-Mbyte Mode	11b	4 Mbytes

### 9.1 1-Mbyte Mode

In this mode, the memory space is 1 Mbyte. In 1-Mbyte mode, the external area to be accessed is specified using the  $\overline{CS}_i$  ( $i = 0$  to  $3$ ) signals (hereafter referred to as the  $\overline{CS}_i$  area). Figures 9.2 and 9.3 show the Memory Mapping and  $\overline{CS}$  Area in 1-Mbyte mode.

### 9.2 4-Mbyte Mode

In this mode, the memory space is 4 Mbytes. Figure 9.1 shows the DBR Register. Bits BSR2 to BSR0 in the DBR register select a bank number which is to be accessed to read or write data. Setting the OFS bit to 1 (with offset) allows the accessed address to be offset by 40000h.

In 4-Mbyte mode, the  $\overline{CS}_i$  ( $i = 0$  to  $3$ ) pin function differs depending on an area to be accessed.

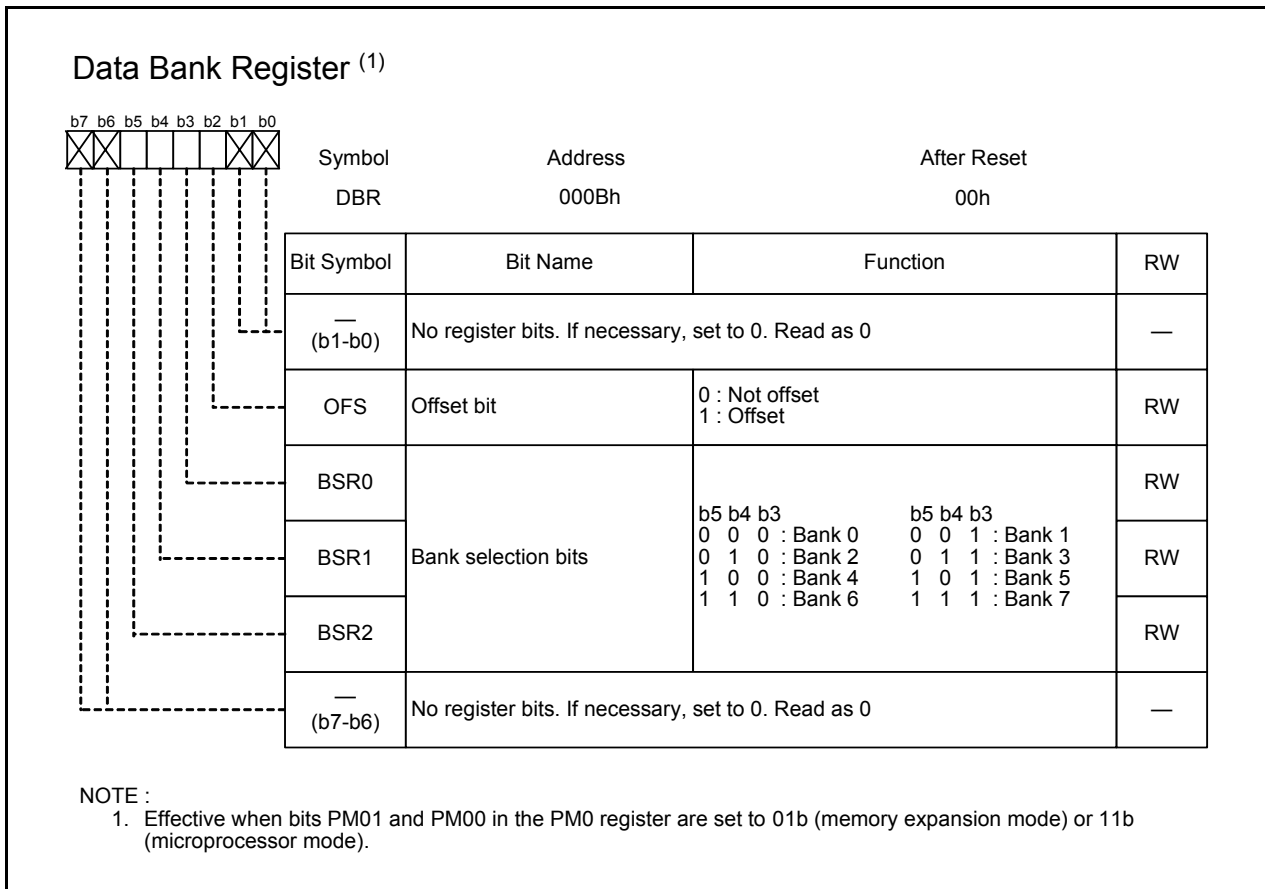
#### 9.2.1 Addresses 04000h to 3FFFFh, C0000h to FFFFFh

- The  $\overline{CS}_i$  signal is output from the  $\overline{CS}_i$  pin (same operation as 1-Mbyte mode; however, the last address of the  $\overline{CS}_1$  area is 3FFFFh).

#### 9.2.2 Addresses 40000h to BFFFFh

- The  $\overline{CS}_0$  pin outputs "L"
- Pins  $\overline{CS}_1$  to  $\overline{CS}_3$  output the setting values of bits BSR2 to BSR0 (bank number)

Figures 9.4 and 9.5 show the Memory Mapping and  $\overline{CS}$  Area in 4-Mbyte mode. Note that banks 0 to 6 are data-only areas. Locate the program in bank 7 or the  $\overline{CS}_i$  area.



**Figure 9.1 DBR Register**



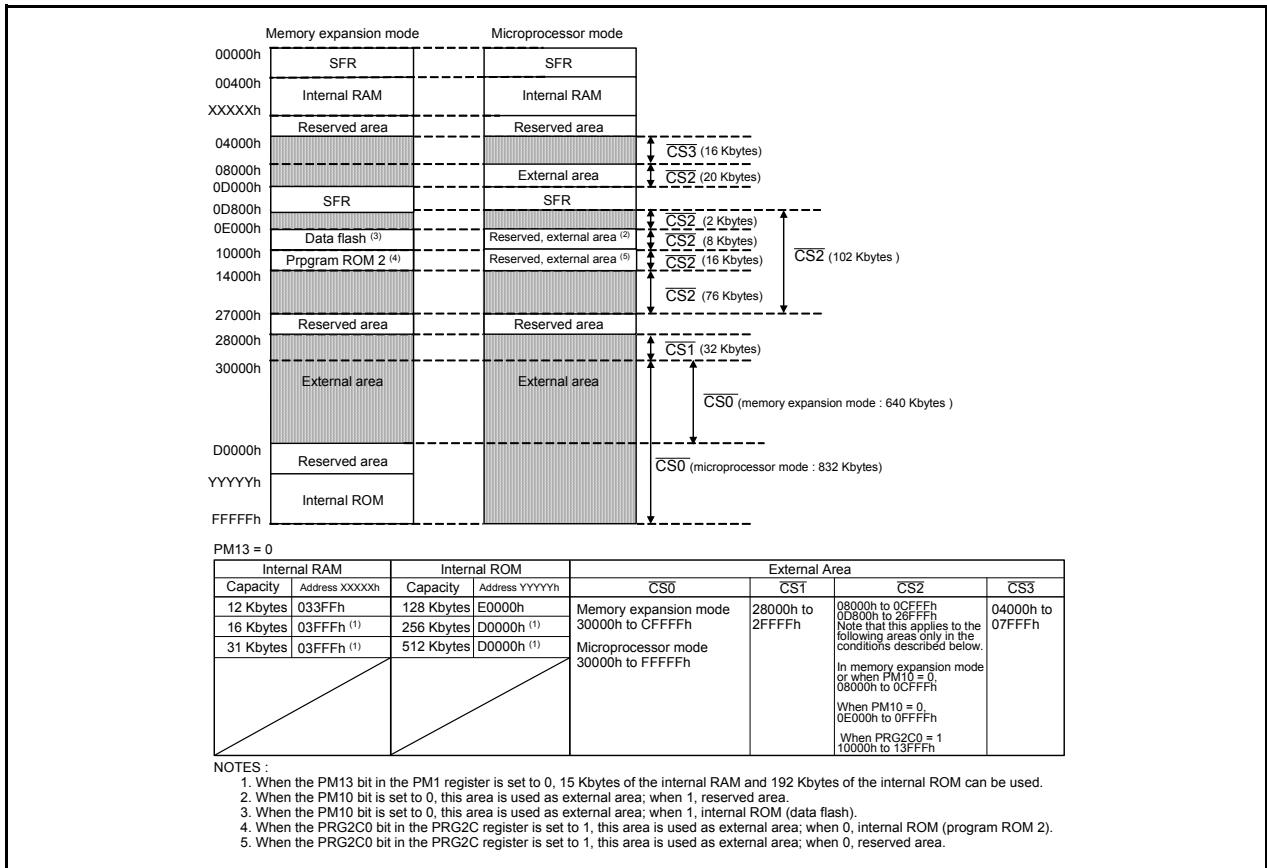


Figure 9.2 Memory Mapping and CS Area in 1-Mbyte Mode (PM13 = 0)

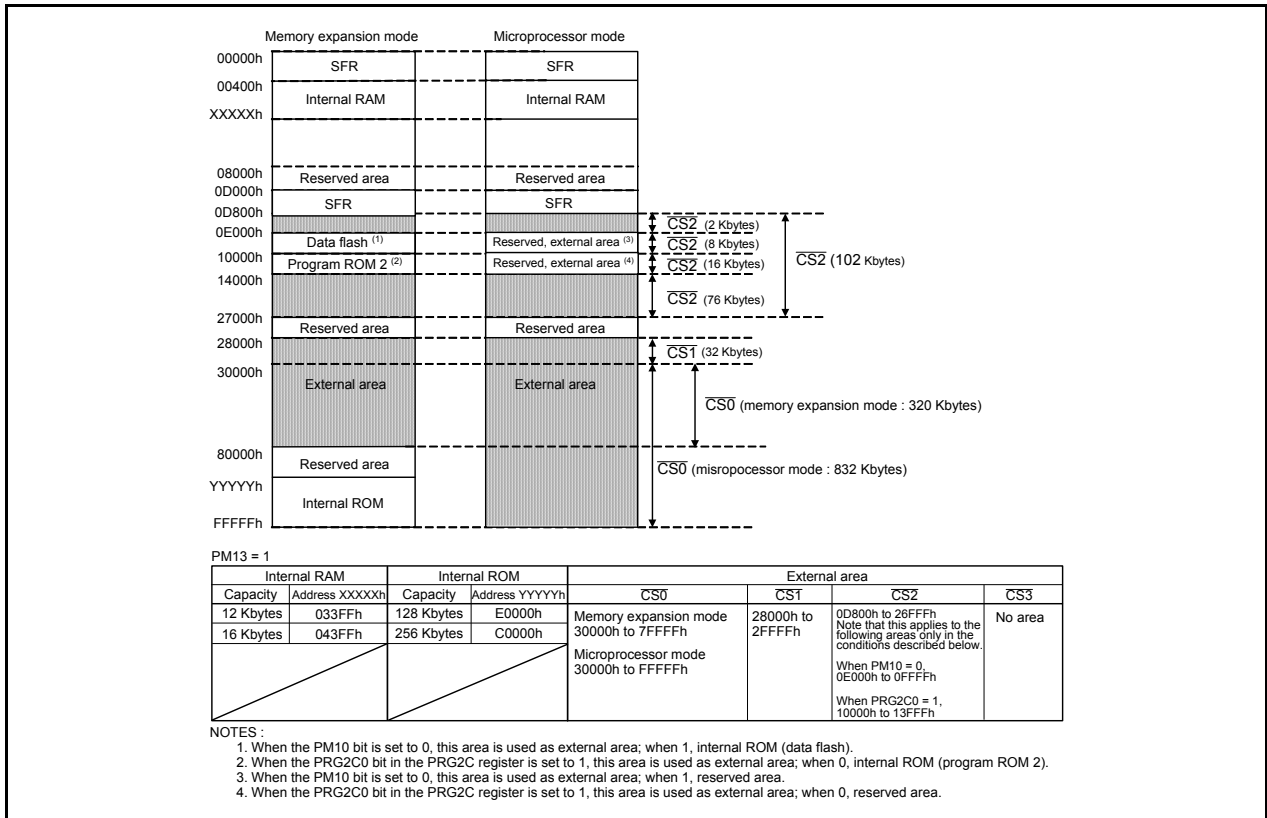


Figure 9.3 Memory Mapping and CS Area in 1-Mbyte Mode (PM13 = 1)

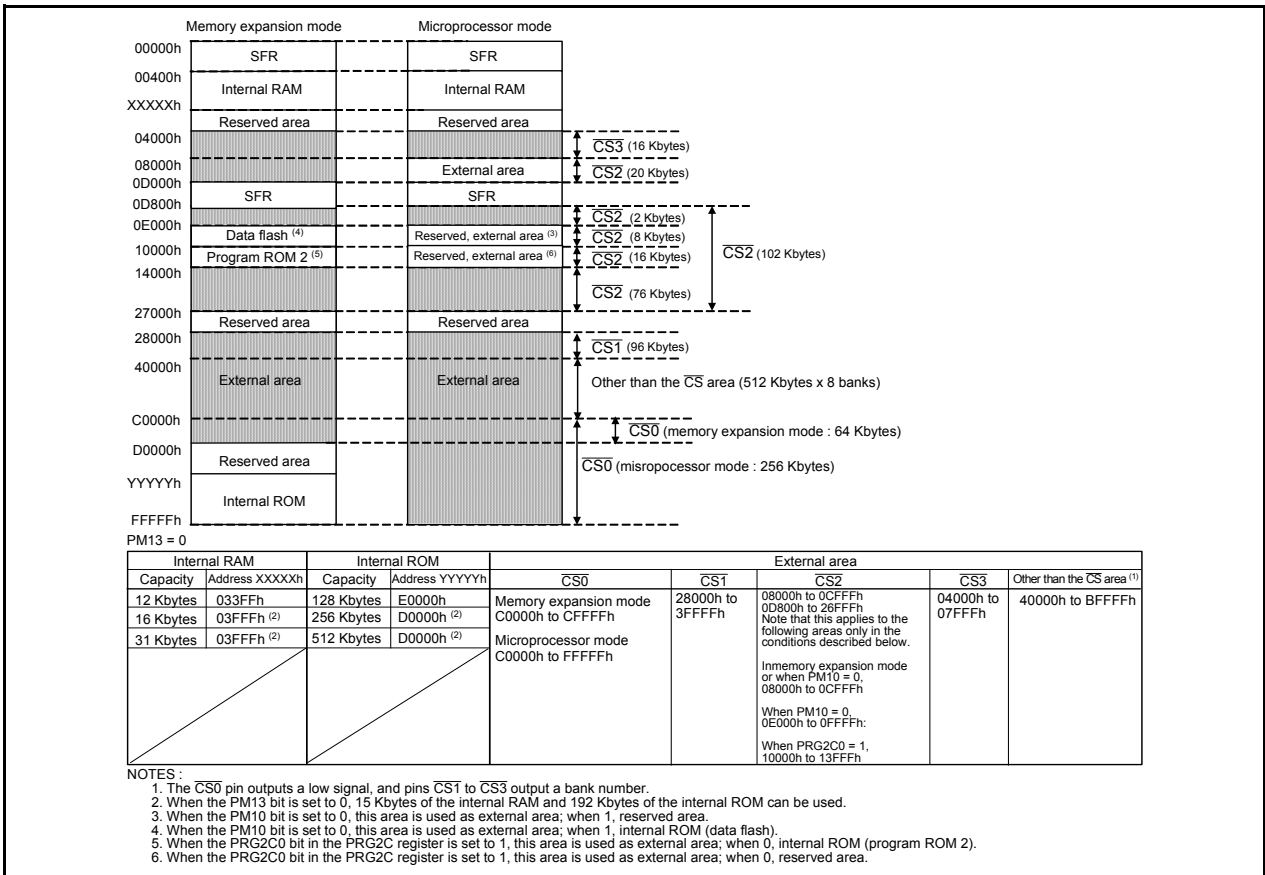


Figure 9.4 Memory Mapping and CS Area in 4-Mbyte Mode (PM13 = 0)

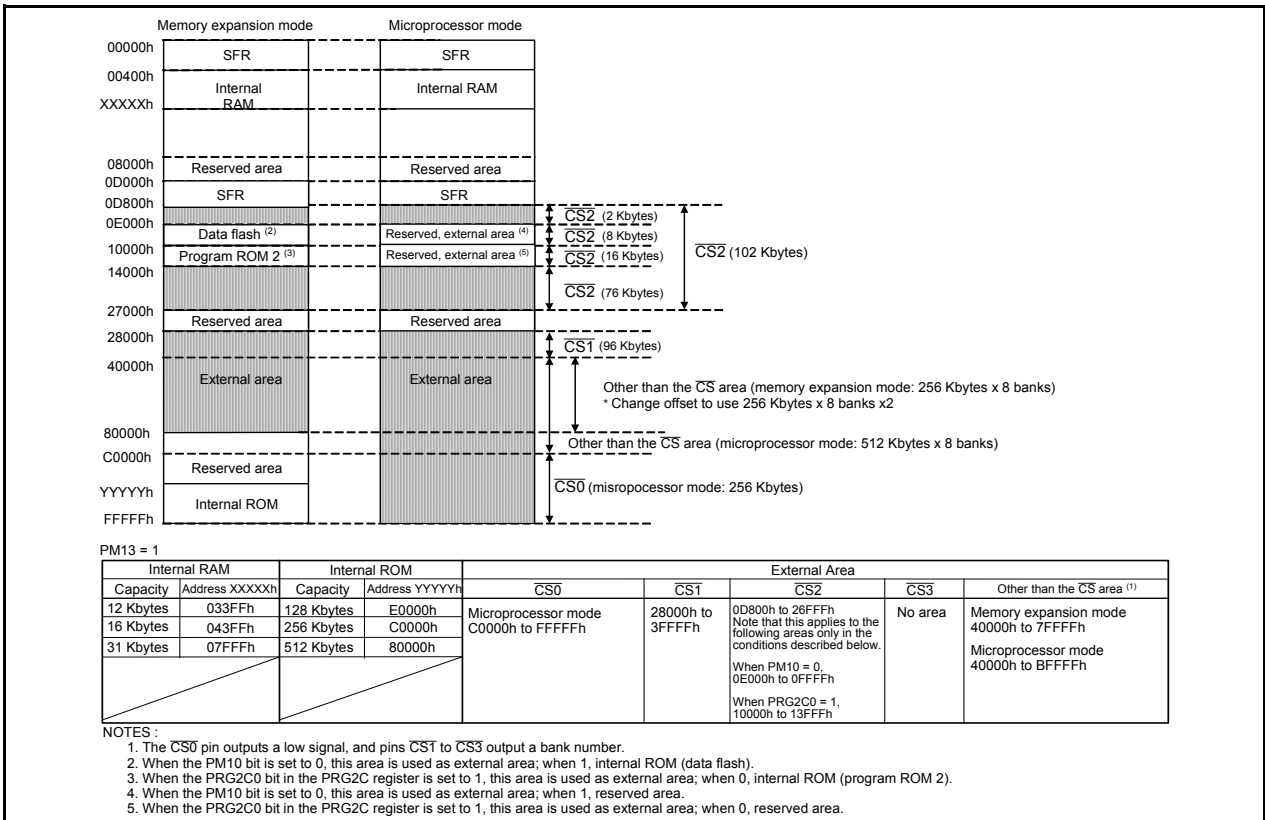


Figure 9.5 Memory Mapping and CS Area in 4-Mbyte Mode (PM13 = 1)

Figure 9.6 shows the External Memory Connect Example in 4-Mbyte Mode.

In this example, the  $\overline{CS}$  pin of 4-Mbyte ROM is connected to the  $\overline{CS0}$  pin of the microcomputer. The 4-Mbyte ROM address input pins AD21, AD20, and AD19 are connected to pins  $\overline{CS3}$ ,  $\overline{CS2}$ , and  $\overline{CS1}$  of the microcomputer, respectively. The address input AD18 pin is connected to the A19 pin of microcomputer. Figures 9.7 to 9.9 show the relationship of addresses between the 4-Mbyte ROM and the microcomputer for the case of a connection example in Figure 9.6.

In microprocessor mode or in memory expansion mode where the PM13 bit in the PM1 register is 0, banks are located every 512 Kbytes. Setting the OFS bit in the DBR register to 1 (offset) allows the accessed address to be offset by 40000h, so that even the data overlapping at a bank boundary can be accessed in succession.

In memory expansion mode where the PM13 bit is 1, each 512-Kbyte bank can be accessed in 256 Kbyte units by switching them over with the OFS bit.

Because the SRAM can be accessed on condition that the chip select signals  $S2 = "H"$  and  $\overline{S1} = "L"$ ,  $\overline{CS0}$  and  $\overline{CS2}$  can be connected to  $S2$  and  $\overline{S1}$ , respectively. If the SRAM does not have the input pins which accept "H" active and "L" active chip select signals ( $\overline{S1}$ ,  $S2$ ),  $\overline{CS0}$  and  $\overline{CS2}$  should be decoded external to the chip.

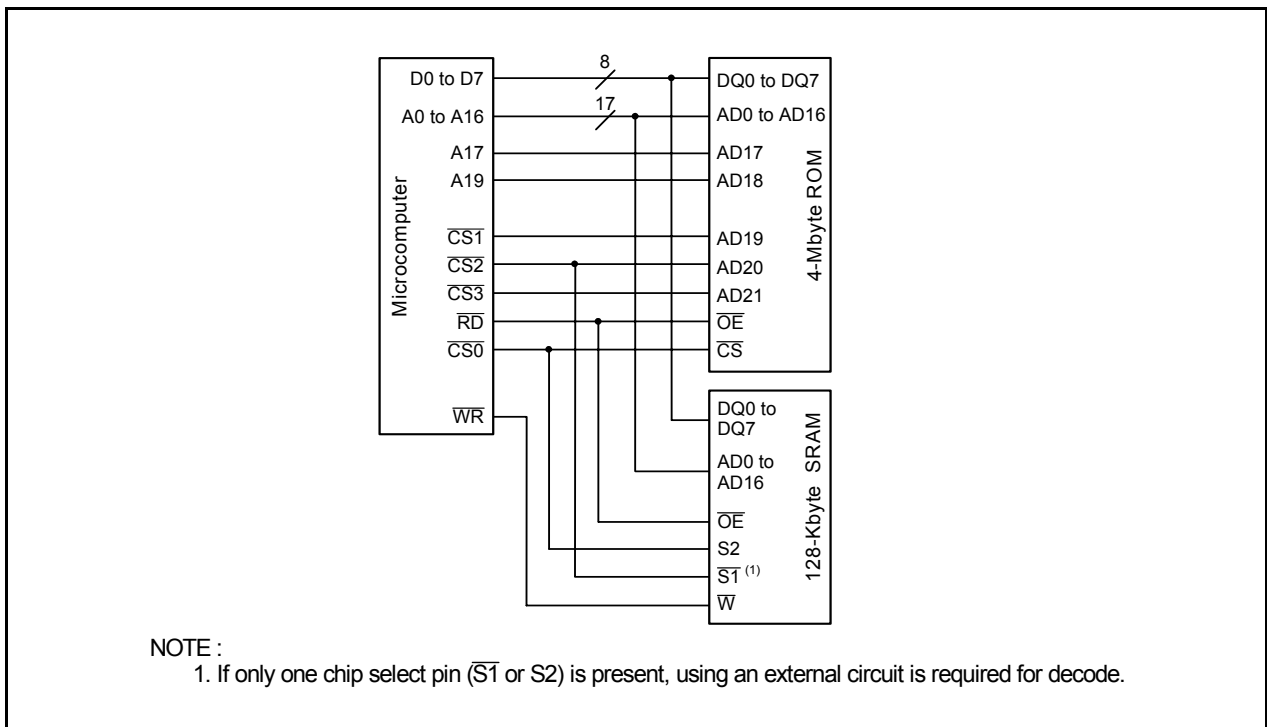


Figure 9.6 External Memory Connect Example in 4-Mbyte Mode

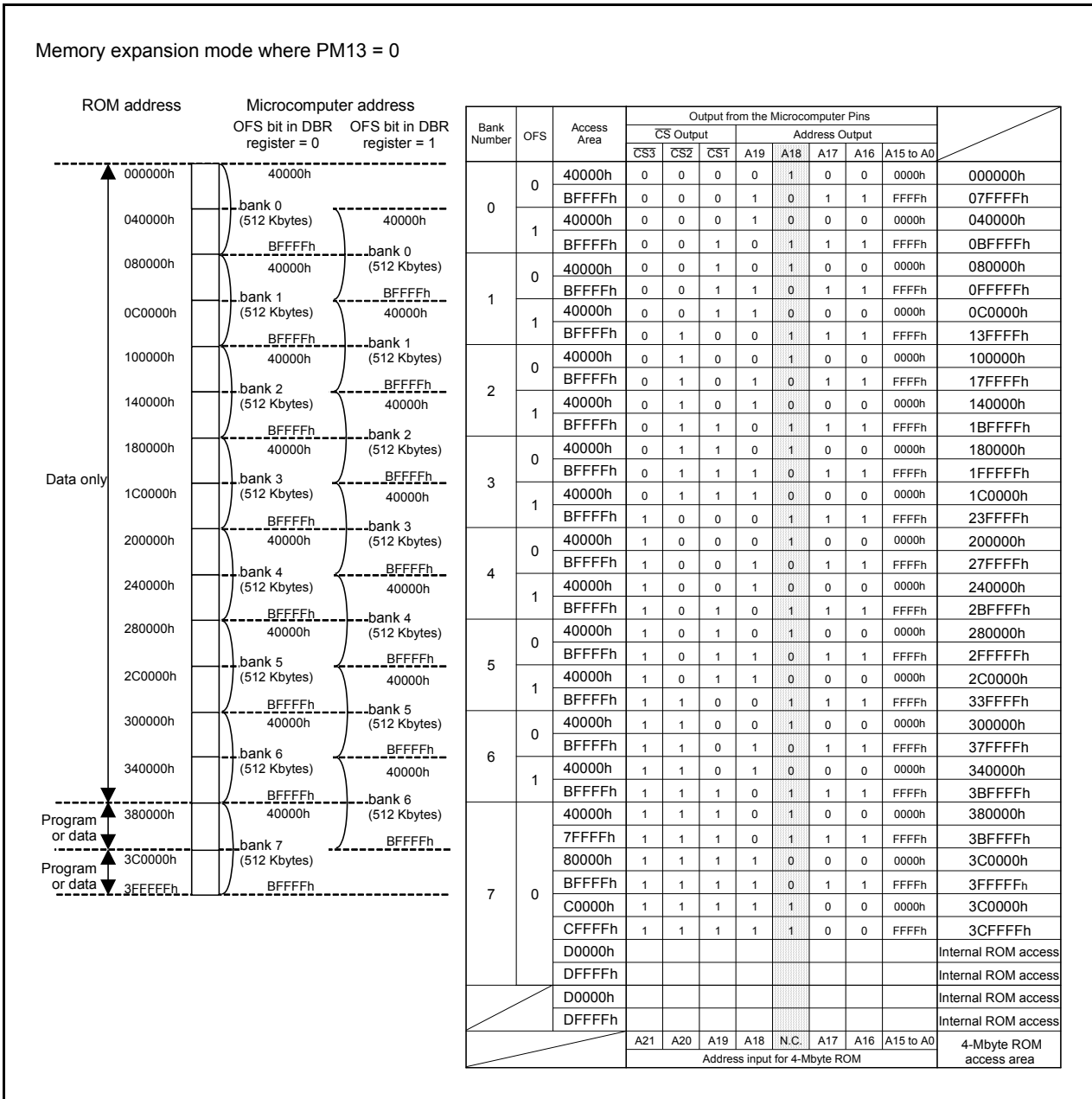


Figure 9.7 Relationship Between Addresses on 4-Mbyte ROM and Those on Microcomputer (1)

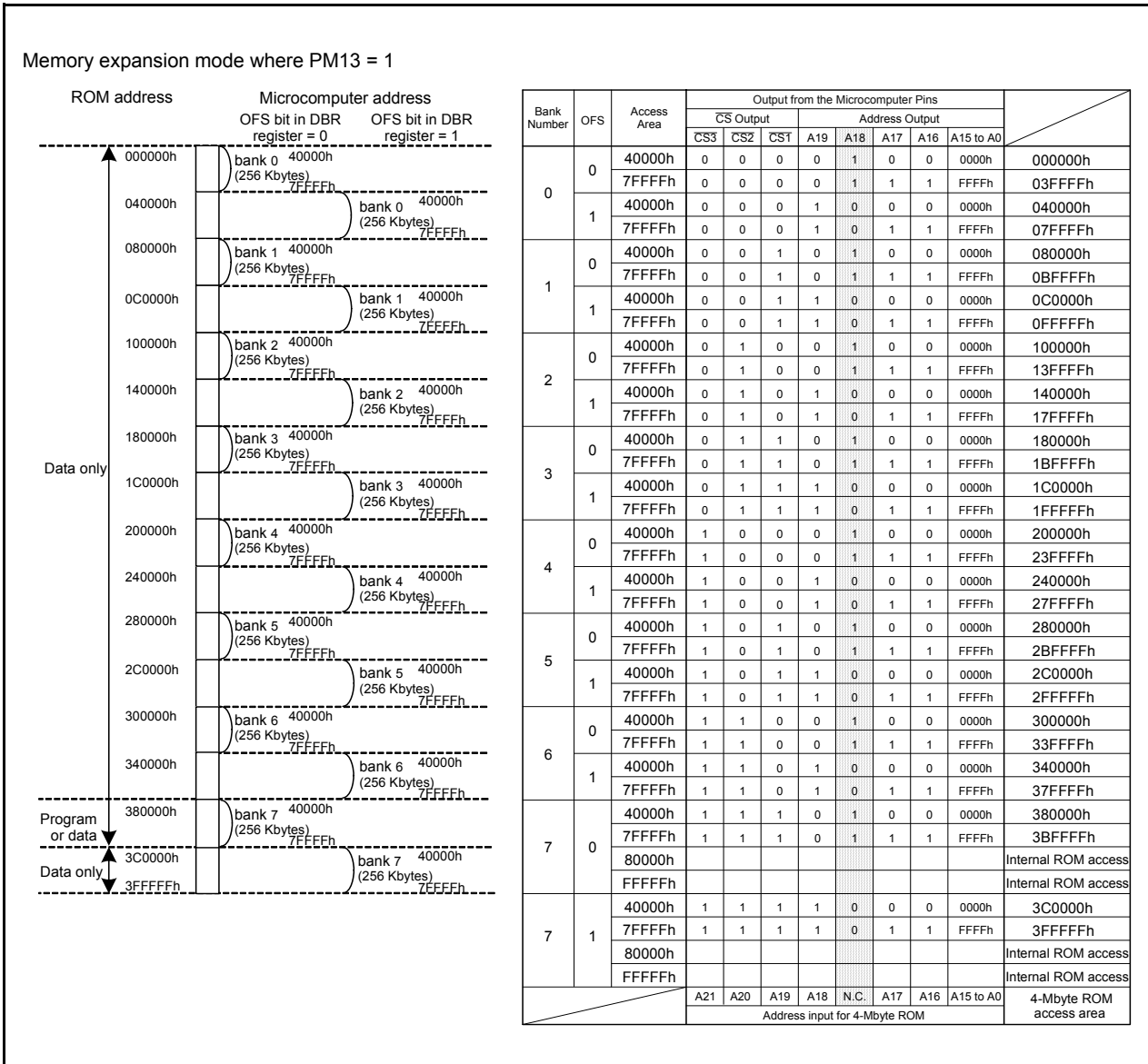


Figure 9.8 Relationship Between Addresses on 4-Mbyte ROM and Those on Microcomputer (2)

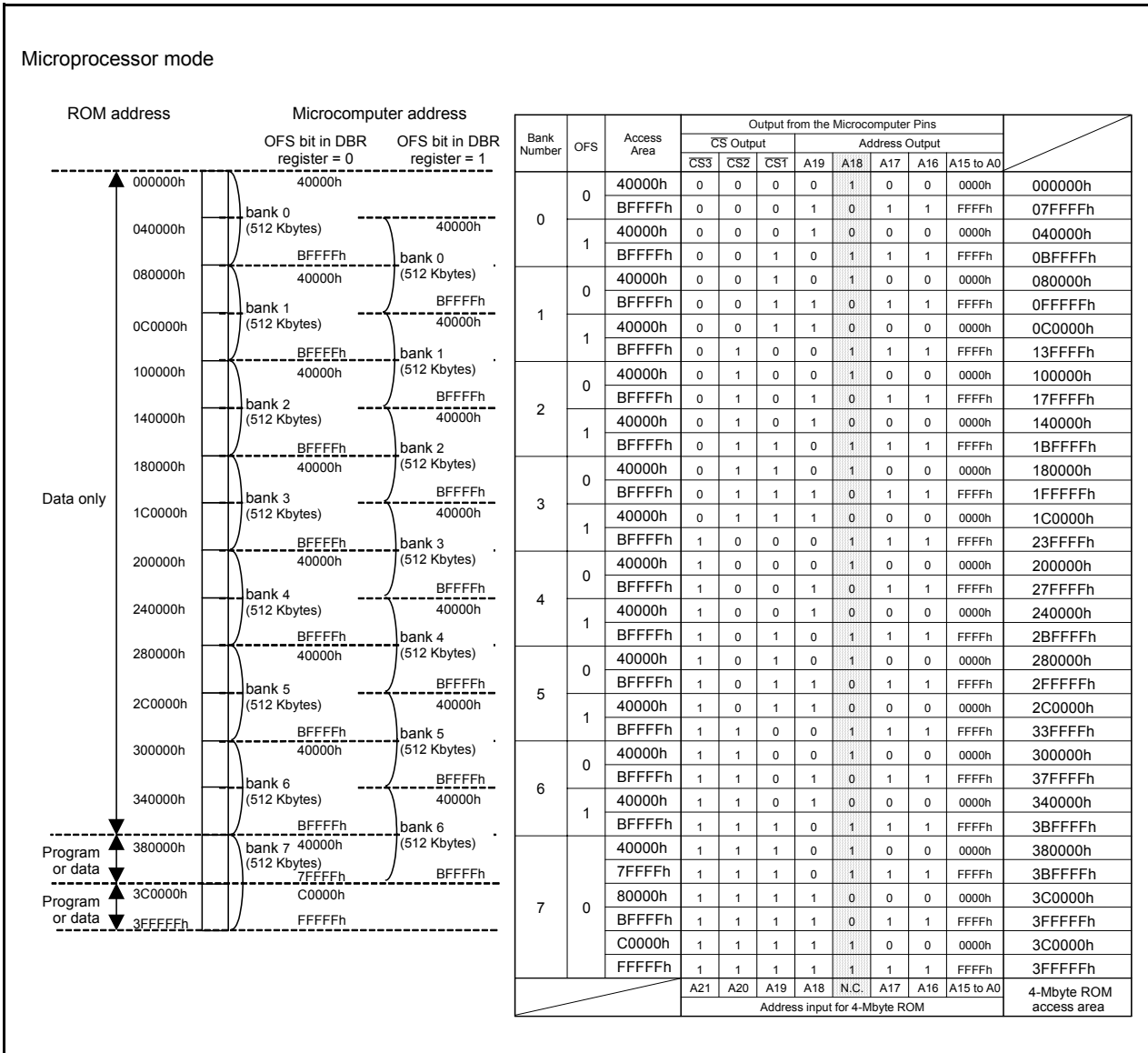


Figure 9.9 Relationship Between Addresses on 4-Mbyte ROM and Those on Microcomputer (3)

## 10. Clock Generation Circuit

### 10.1 Type of the Clock Generation Circuit

4 circuits are incorporated to generate the system clock signal:

- Main clock oscillation circuit
- Sub clock oscillation circuit
- 125 kHz on-chip oscillator
- PLL frequency synthesizer

Table 10.1 lists the Clock Generation Circuit Specifications. Figure 10.1 shows the System Clock Generation Circuit.

Figures 10.2 to 10.6 show the clock-related registers.

**Table 10.1 Clock Generation Circuit Specifications**

Item	Main Clock Oscillation Circuit	Sub Clock Oscillation Circuit	125 kHz On-Chip Oscillator	PLL Frequency Synthesizer
Use of Clock	CPU clock source Peripheral function clock source	CPU clock source Clock source of timer A and B.	CPU clock source Peripheral function clock source CPU and peripheral function clock sources when the main clock stops oscillating	CPU clock source Peripheral function clock source
Clock Frequency	0 to 20 MHz	32.768 kHz	About 125 kHz	10 to 25 MHz
Usable Oscillator	Ceramic oscillator Crystal oscillator	Crystal oscillator	-	- (1)
Pins to Connect Oscillator	XIN, XOUT	XCIN, XCOU	-	- (1)
Oscillation Stop, Restart Function	Presence	Presence	Presence	Presence
Oscillator Status After Reset	Oscillating	Stopped	Oscillating	Stopped
Other	Externally derived clock can be input		-	- (1)

NOTE:

1. The PLL frequency synthesizer uses the main clock oscillation circuit as a reference clock source. The items above are based on those of the main clock oscillation circuit.

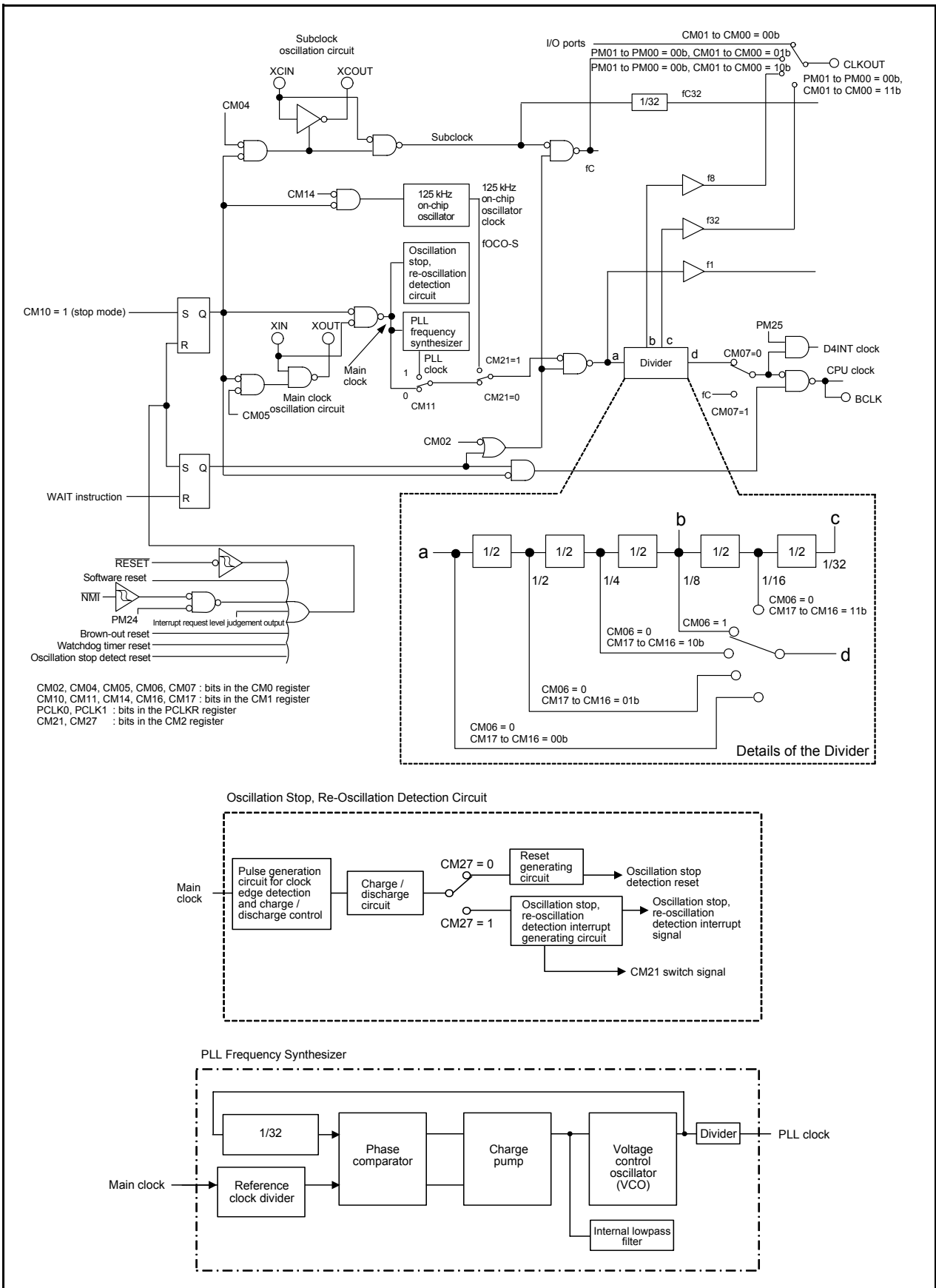


Figure 10.1 System Clock Generation Circuit



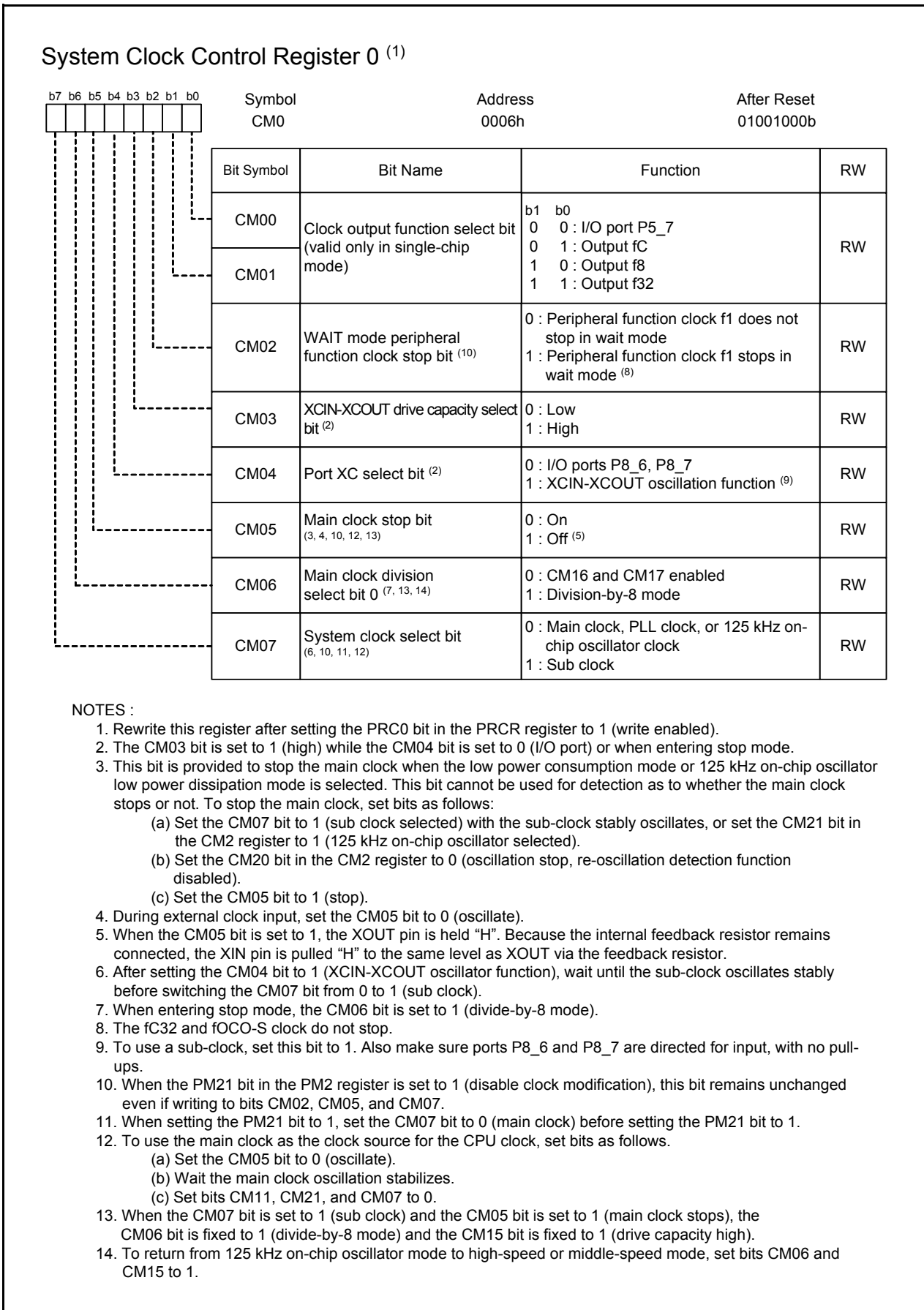


Figure 10.2 CM0 Register

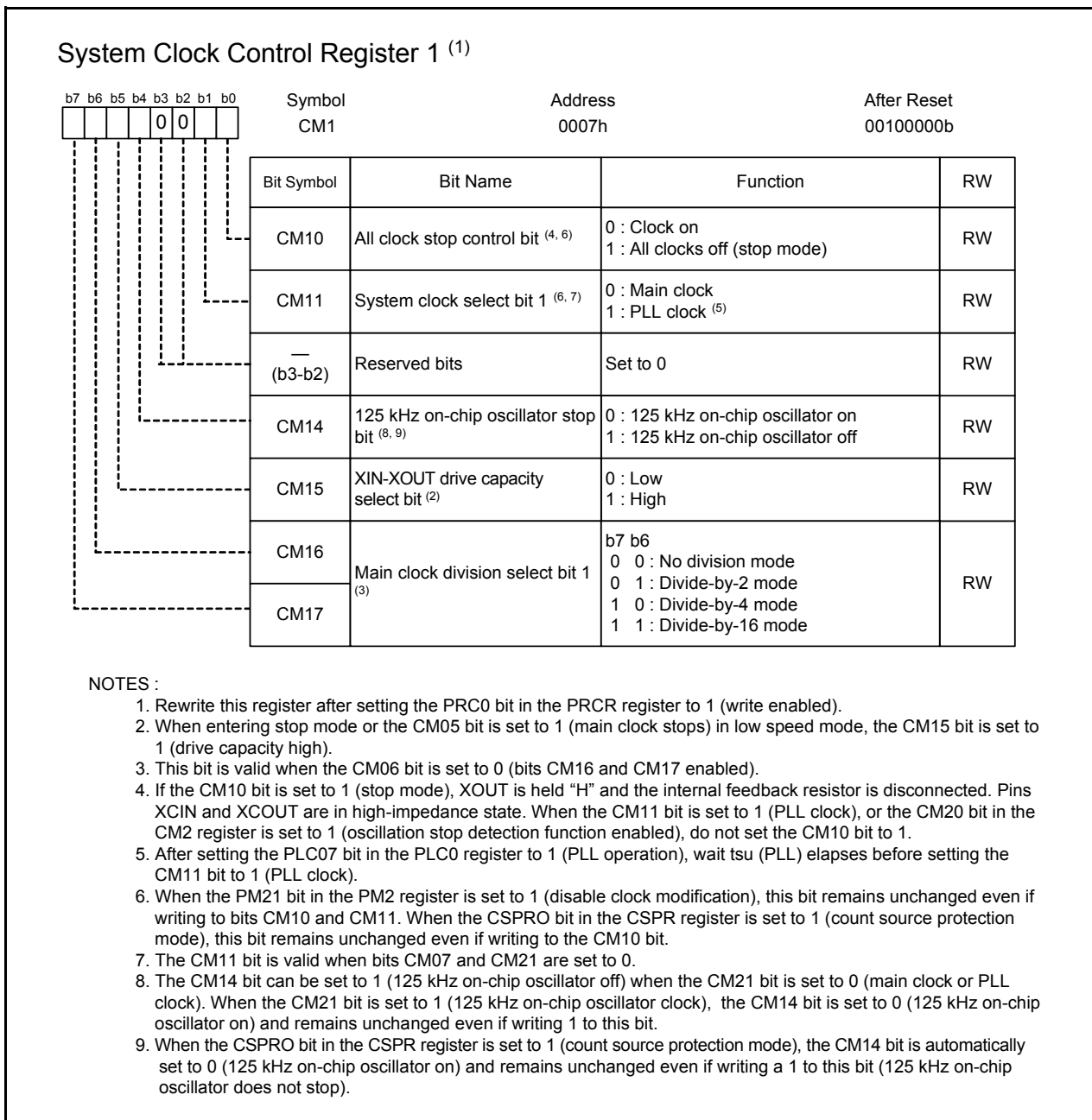


Figure 10.3 CM1 Register

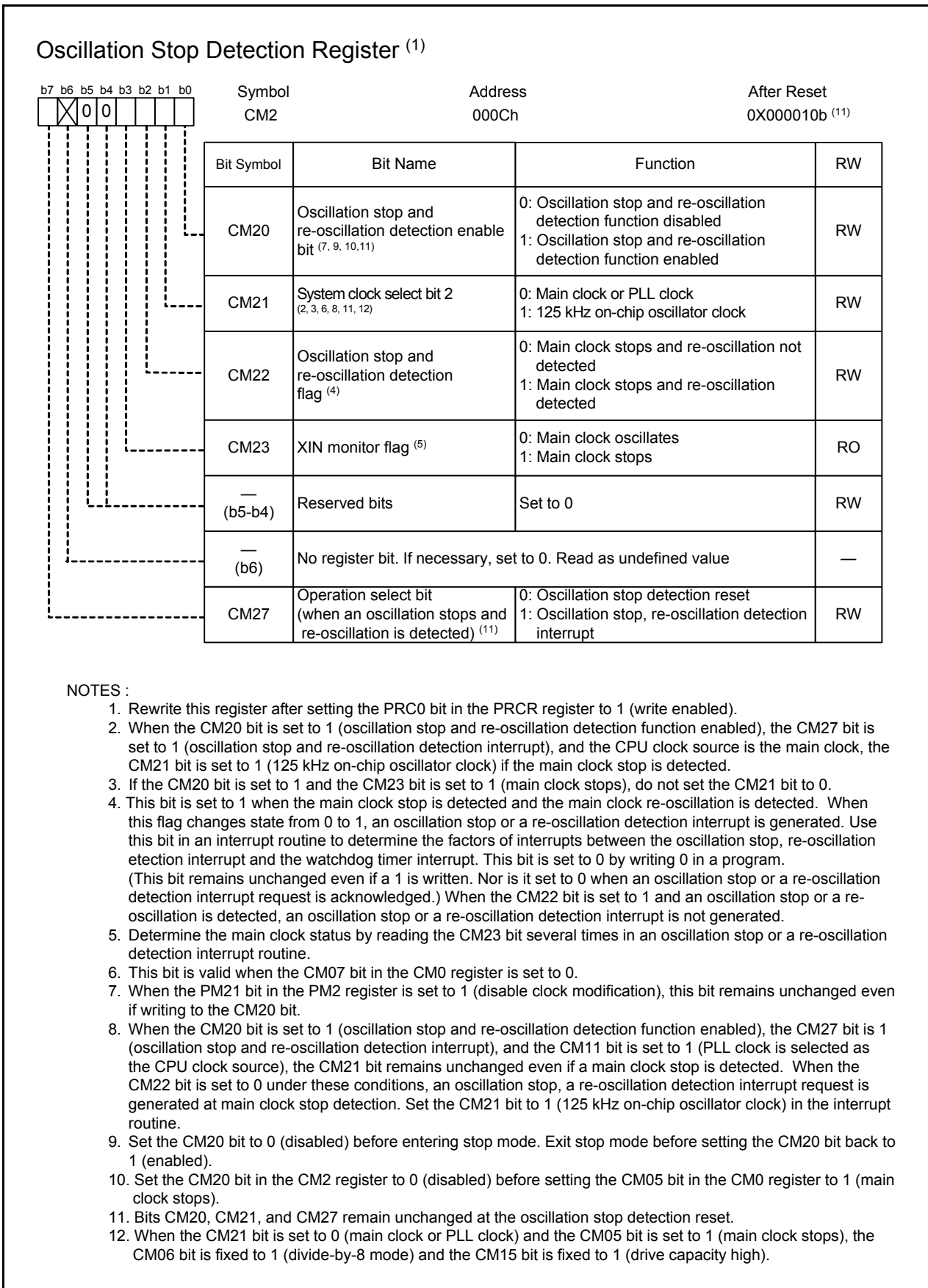


Figure 10.4 CM2 Register

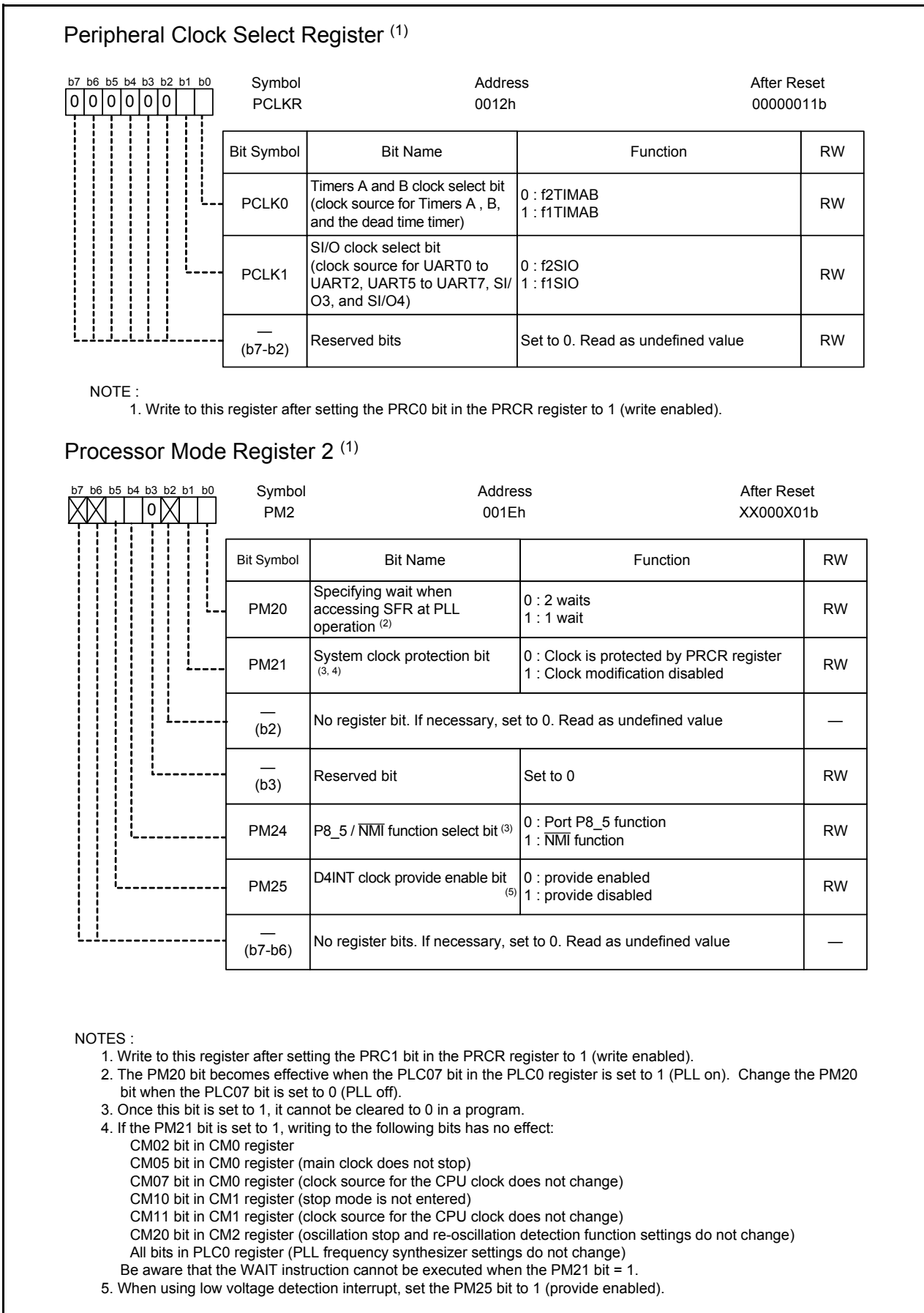


Figure 10.5 PCLKR Register and PM2 Register

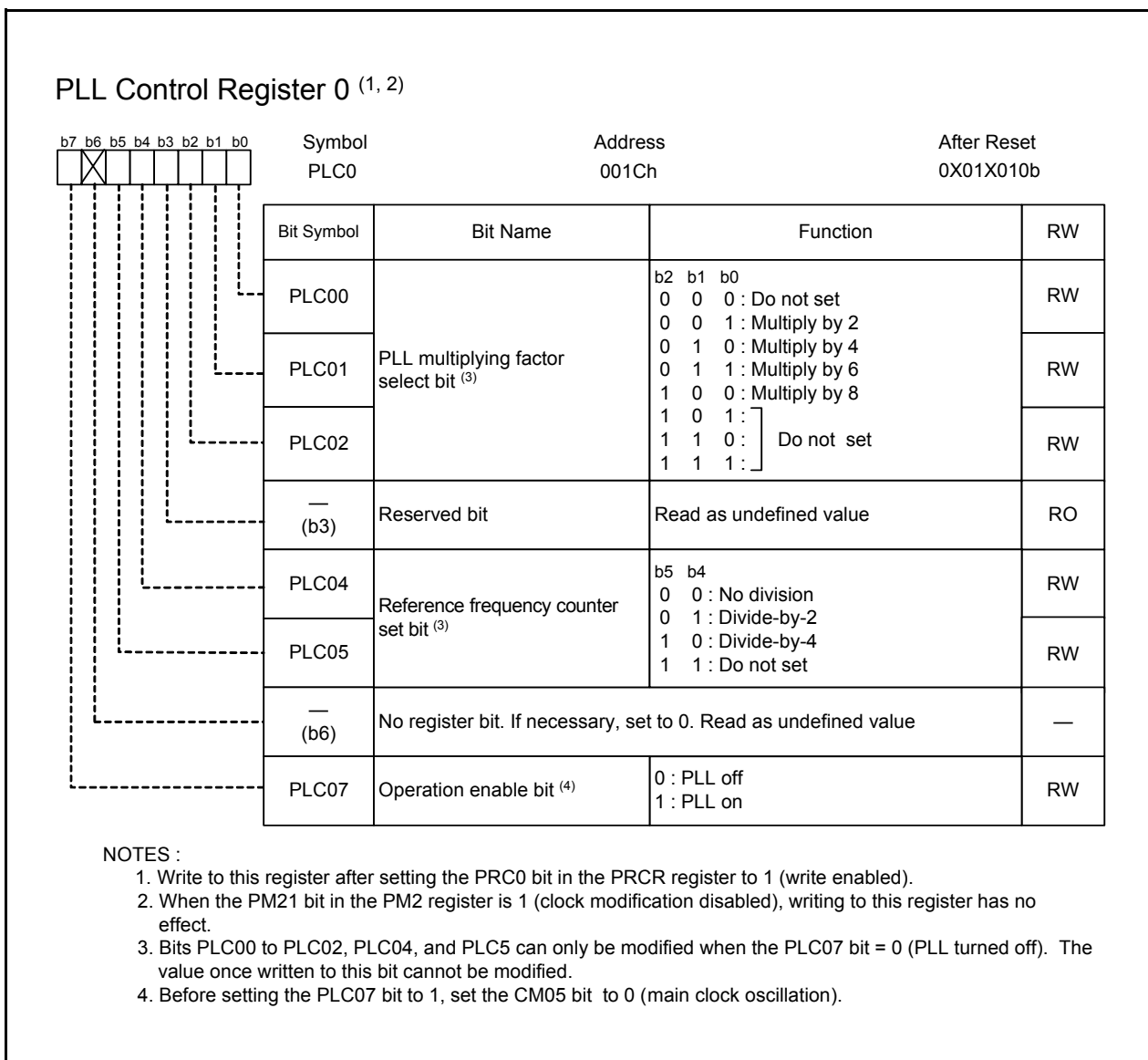


Figure 10.6 PLC0 Register

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

### 10.1.1 Main Clock

This clock is provided by the main clock oscillation circuit and used as the clock source for the CPU and peripheral function clocks. The main clock oscillation circuit is configured by connecting a resonator between pins XIN and XOUT. The main clock oscillation circuit contains a feedback resistor, which is disconnected from the oscillation circuit during stop mode in order to reduce the amount of power consumed in the chip. The main clock oscillation circuit may also be configured by feeding an externally generated clock to the XIN pin. Figure 10.7 shows the Examples of Main Clock Connection Circuit.

The power consumption in the chip can be reduced by setting the CM05 bit in the CM0 register to 1 (main clock oscillation circuit turned off) after switching the clock source for the CPU clock to a sub clock or 125 kHz on-chip oscillation 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 **10.4 "Power Control"** for details.

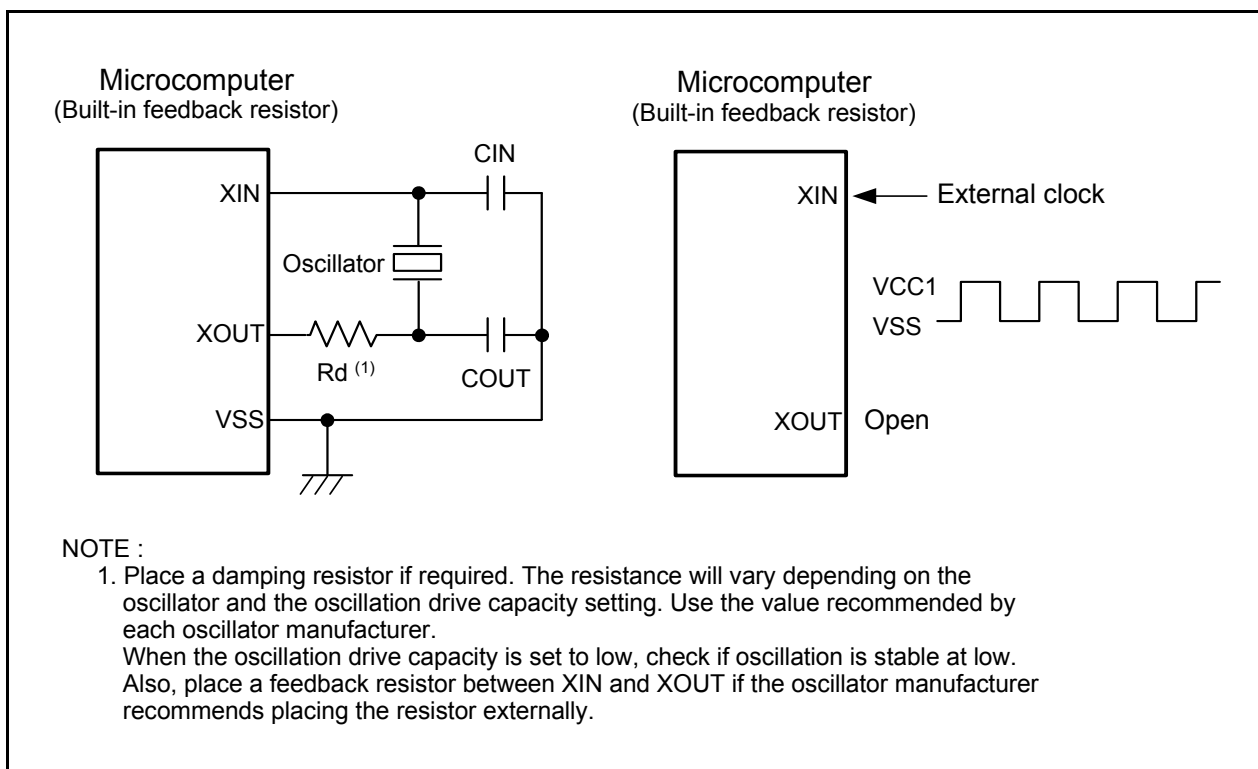


Figure 10.7 Examples of Main Clock Connection Circuit

### 10.1.2 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. In addition, an fC clock with the same frequency as that of the sub clock can be output from the CLKOUT pin.

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

Figure 10.8 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 oscillation circuit.

To use the sub clock for the CPU clock, set the CM07 bit in the 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 **10.4 “Power Control”** for details.

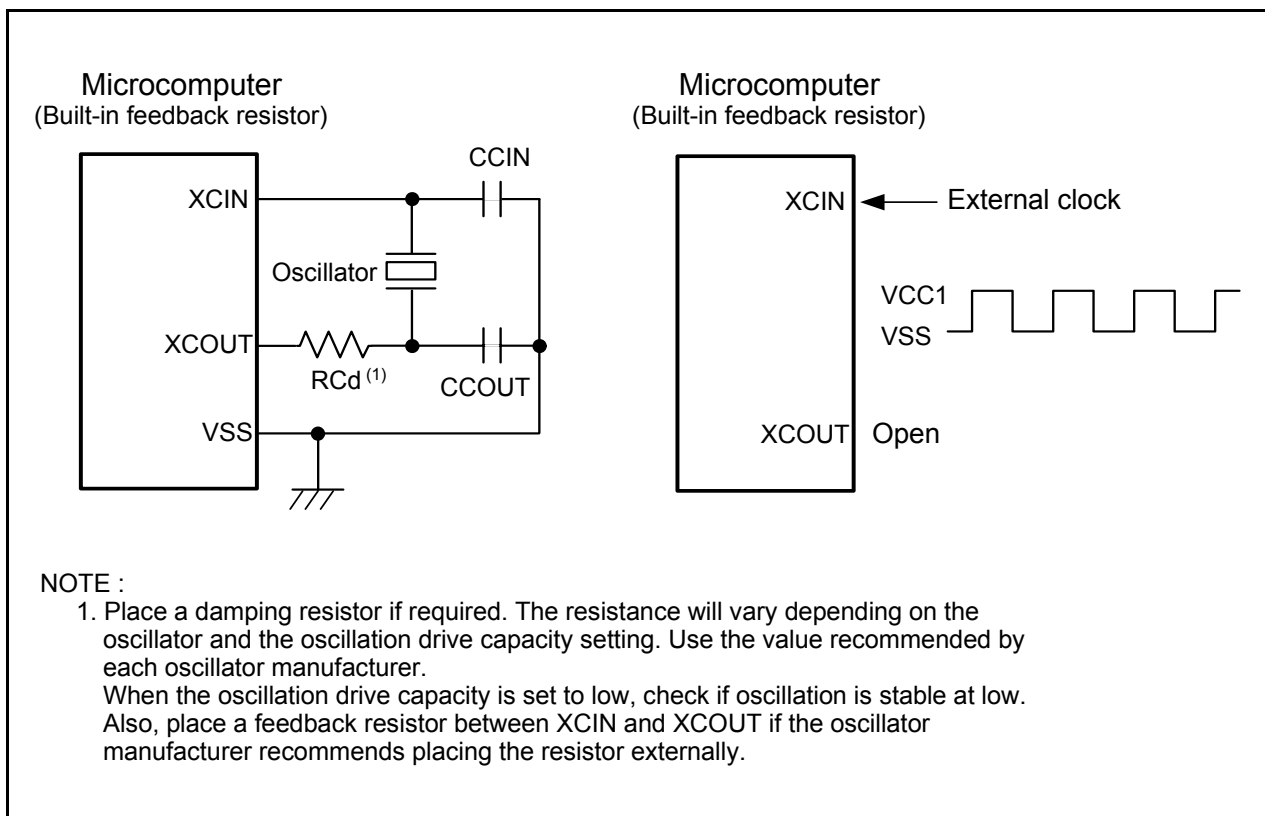


Figure 10.8 Examples of Sub Clock Connection Circuit

### 10.1.3 125 kHz On-Chip Oscillator Clock (fOCO-S)

This clock, approximately 125 kHz, is supplied by 125 kHz on-chip oscillator. This clock is used as the clock source for the CPU and peripheral function clocks. In addition, if the CSPRO bit in the CSPR register is 1 (count source protection mode enabled), this clock is used as the count source for the watchdog timer (Refer to 13.2 “Count Source Protection Mode Enabled”).

After reset, the 125 kHz on-chip oscillator divided by 8 provides the CPU clock. It stops when the CM14 bit in the CM1 register is set to 0 (125 kHz on-chip oscillator stops). If the main clock stops oscillating when the CM20 bit in the CM2 register is 1 (oscillation stop, re-oscillation detection function enabled) and the CM27 bit is 1 (oscillation stop, re-oscillation detection interrupt), the 125 kHz on-chip oscillator automatically starts operating and supplying the necessary clock for the microcomputer.

### 10.1.4 PLL Clock

The PLL clock is generated by the PLL frequency synthesizer. This clock is used as the clock source for the CPU and peripheral function clocks. After reset, the PLL frequency synthesizer 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 for  $t_{su}(PLL)$  until 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 10.10 shows the Procedure to Use PLL Clock as CPU Clock Source.

The PLL clock is the main clock divided by the selected values of bits PLC05 and PLC04 in the PLC0 register, and then multiplied by the selected values of bits PLC02 to PLC00. Set bits PLC05 and PLC04 to fit divided frequency between 2 MHz and 5 MHz. Figure 10.9 shows the Relation between the Main Clock and the PLL Clock.

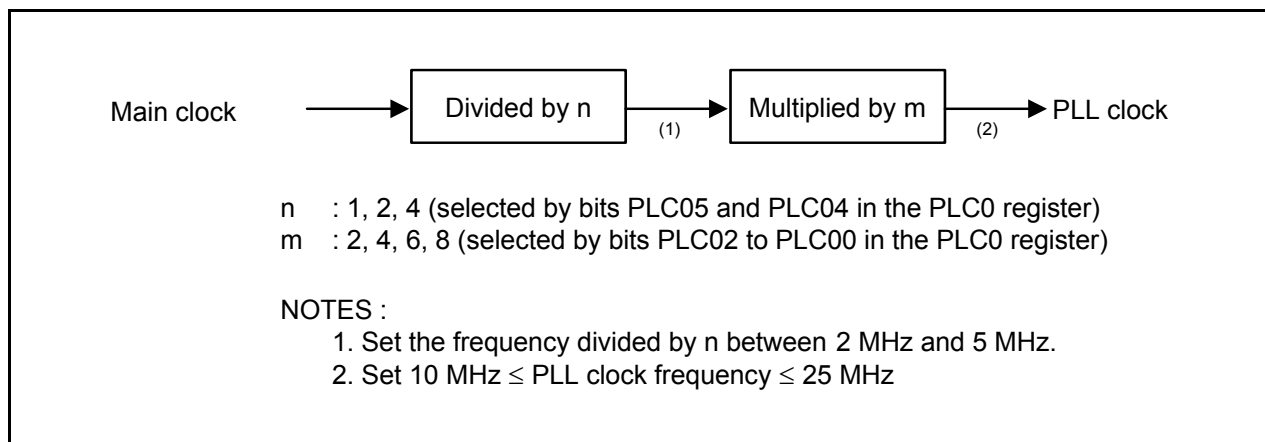


Figure 10.9 Relation between the Main Clock and the PLL Clock



Bits PLC05 and PLC04 and bits PLC02 to PLC00 can be set only once after reset. Table 10.2 shows the Example for Setting PLL Clock Frequencies.

**Table 10.2 Example for Setting PLL Clock Frequencies**

Main Clock	Setting Value		PLL Clock
	Bits PLC05 and PLC04	Bits PLC02 to PLC00	
10 MHz	01b (divided by 2)	010b (multiplied by 4)	20 MHz
5 MHz	00b (not divided)	010b (multiplied by 4)	
12 MHz	10b (divided by 4)	100b (multiplied by 8)	24 MHz
6 MHz	01b (divided by 2)	100b (multiplied by 8)	

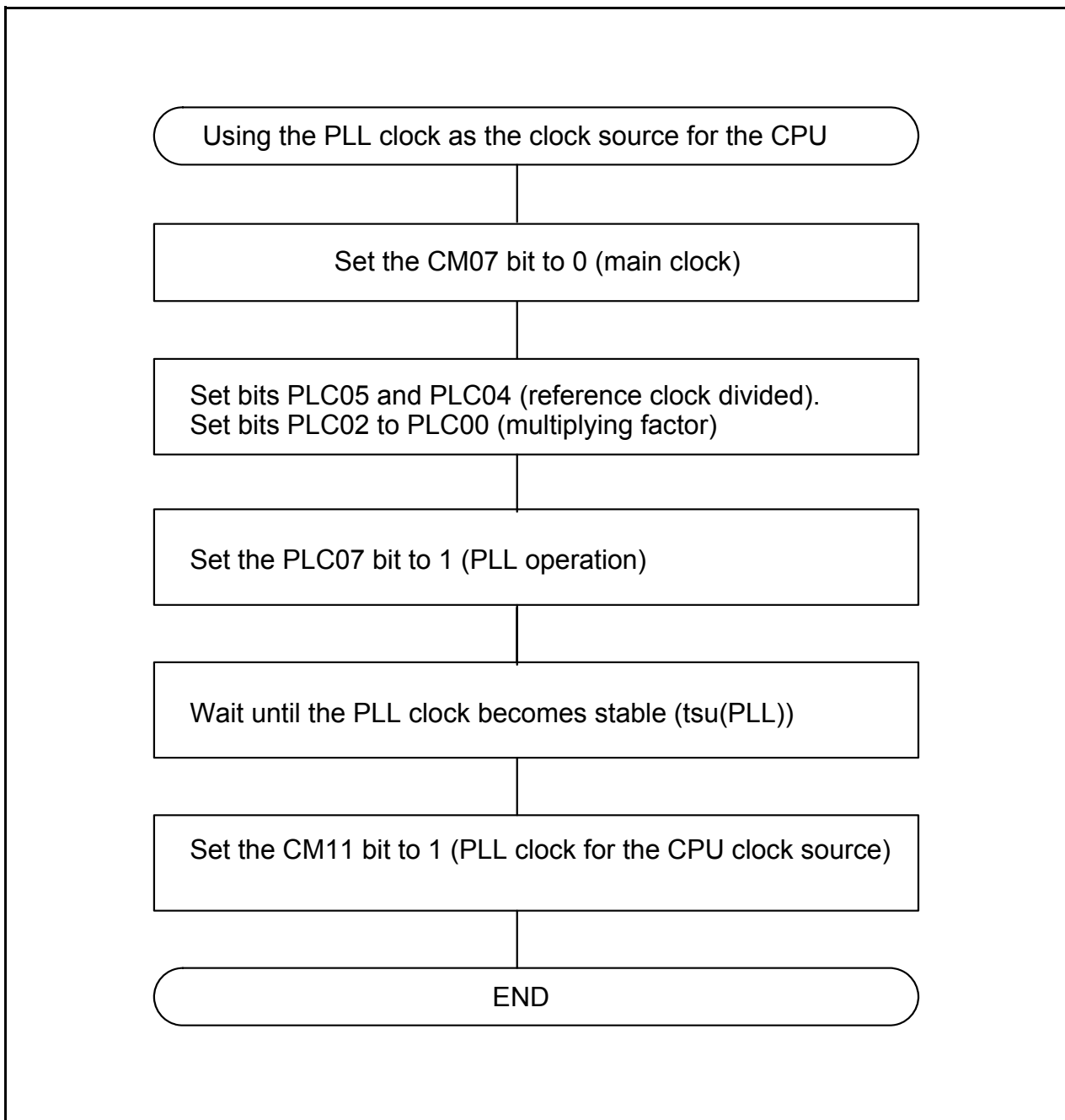


Figure 10.10 Procedure to Use PLL Clock as CPU Clock Source

## 10.2 CPU Clock and Peripheral Function Clock

Two types of clock exists: CPU clock to operate the CPU

Peripheral function clocks to operate the peripheral functions.

### 10.2.1 CPU Clock and BCLK

These are operating clocks for the CPU and watchdog timer.

The main clock, sub clock, 125 kHz on-chip oscillator clock, or the PLL clock can be selected as the clock source for the CPU clock.

When the main clock, PLL clock, or 125 kHz on-chip 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 the CM0 register and bits CM17 and CM16 in the CM1 register to select a divide-by-n value.

After reset, the 125 kHz on-chip oscillator clock divided by 8 provides the CPU clock.

During memory expansion or microprocessor mode, a BCLK signal with the same frequency as the CPU clock can be output from the BCLK pin by setting the PM07 bit in the PM0 register to 0 (output enabled).

Note that when entering stop mode or when the CM05 bit in the CM0 register is set to 1 (stop) in low-speed mode, the CM06 bit in the CM0 register is set to 1 (divide-by-8 mode).

### 10.2.2 Peripheral Function Clock (f1, fC32)

These are operating clocks for the peripheral functions.

f1 is produced from the main clock, the PLL clock, or the 125 kHz on-chip oscillator clock, and is used for timers A and B, UART0 to UART2, UART5 to UART7, SI/O3, SI/O4, and A/D converter.

When the WAIT instruction is executed after setting the CM02 bit in the CM0 register to 1 (peripheral function clock f1 turned off during wait mode), or when the microcomputer is in low power consumption mode, the f1 clock is turned off.

The fC32 clock is produced from the sub clock, and is used for timers A and B. This clock can be used when the sub clock is on.

fOCO-S is used for timers A and B. fOCO-S can be used when the CM14 bit in the CM1 register is set to 0 (125 kHz on-chip oscillator oscillates).

Figure 10.11 shows the Peripheral Function Clock.

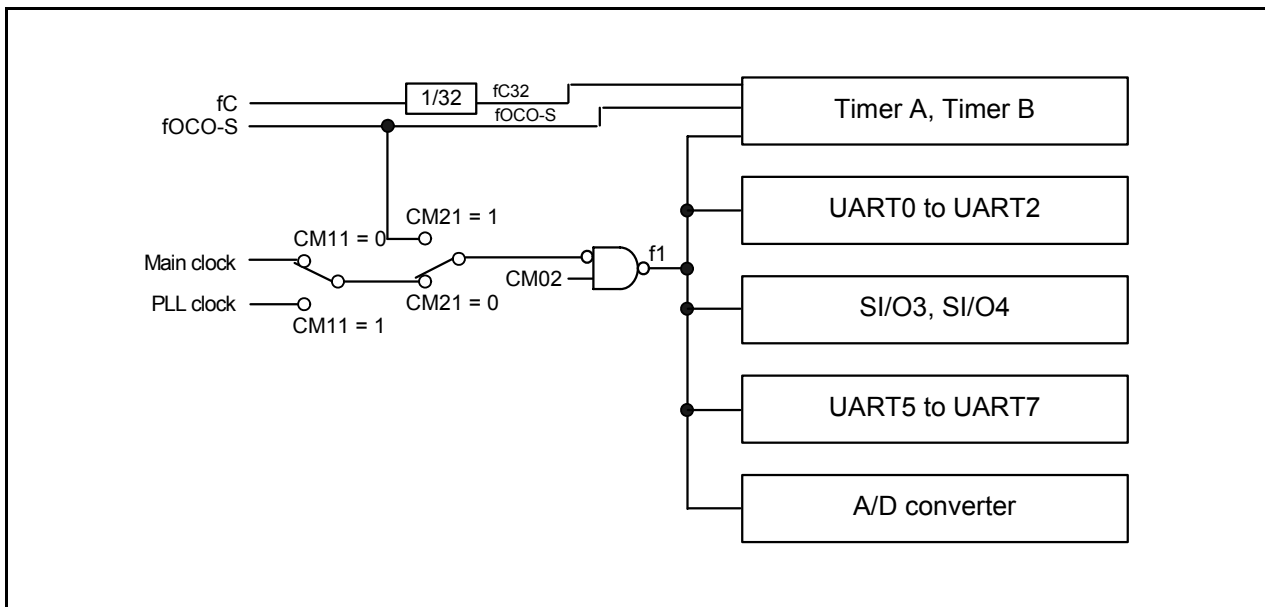


Figure 10.11 Peripheral Function Clock

### 10.3 Clock Output Function

During single-chip mode, the f8, f32, or fC clock can be output from the CLKOUT pin. Use bits CM01 and CM00 in the CM0 register to select.

## 10.4 Power Control

Normal operating mode, wait mode, and stop mode are provided as the power consumption control. All mode states, except wait mode and stop mode, are called normal operating mode in this document.

### 10.4.1 Normal Operating Mode

Normal operating mode is further classified into seven modes.

In normal operating 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.

When the CPU clock source is changed from the 125 kHz on-chip oscillator to the main clock, change the operating mode to the medium speed mode (divided by 8 mode) after the clock was divided by 8 (the CM06 bit in the CM0 register was set to 1) in the 125 kHz on-chip oscillator mode.

#### 10.4.1.1 High-speed Mode

The main clock divided by 1 provides the CPU clock. If the sub clock is on, fC32 can be used as the count source for timers A and B.

#### 10.4.1.2 PLL Operating Mode

The PLL clock serves as the CPU clock. If the sub clock is on, fC32 can be used as the count source for timers A and B. If fOCO-S is oscillating, fOCO-S can be used as the count source for timers A and B. PLL operating mode can be entered from high-speed mode or medium-speed mode. If PLL operating mode is to be changed to wait or stop mode, first go to high-speed mode or medium-speed mode before changing.

#### 10.4.1.3 Medium-Speed Mode

The main clock divided by 2, 4, 8, or 16 provides the CPU clock. If the sub clock is on, fC32 can be used as the count source for timers A and B. If fOCO-S is oscillating, fOCO-S can be used as the count source for timers A and B.

#### 10.4.1.4 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 in the CM2 register is set to 0 (main clock or PLL clock), and the 125 kHz on-chip oscillator clock is used when the CM21 bit is set to 1 (125 kHz on-chip oscillator clock).

The fC32 clock can be used as the count source for timers A and B.

#### 10.4.1.5 Low Power Consumption 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 fC32 clock can be used as the count source for timers A and B. If fOCO-S is oscillating, fOCO-S can be used as the count source for timers A and B.

Simultaneously when this mode is selected, the CM06 bit in the CM0 register becomes 1 (divided by 8 mode). In the low power consumption 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.

### 10.4.1.6 125 kHz On-Chip Oscillator Mode

The 125 kHz on-chip oscillator clock divided by 1 (undivided), 2, 4, 8, or 16 provides the CPU clock. The 125 kHz on-chip oscillator clock is also the clock source for the peripheral function clocks. If the sub clock is on, fC32 can be used as the count source for timers A and B. When the operating mode is returned to the high- and medium-speed modes, set the CM06 bit in the CM0 register to 1 (divided by 8 mode).

### 10.4.1.7 125 kHz On-Chip Oscillator Low Power Consumption Mode

The main clock is turned off after being placed in 125 kHz on-chip oscillator mode. The CPU clock can be selected as in the 125 kHz on-chip oscillator mode. The 125 kHz on-chip oscillator clock is the clock source for the peripheral function clocks. If the sub clock is on, fC32 can be used as the count source for timers A and B.

**Table 10.3 Setting Clock Related Bit and Modes**

Mode		CM2 Register	CM1 Register			CM0 Register			
		CM21	CM11	CM14	CM17, CM16	CM07	CM06	CM05	CM04
PLL Operating Mode	divided by 1	0	1	-	00b	0	0	0	-
	divided by 2	0	1	-	01b	0	0	0	-
	divided by 4	0	1	-	10b	0	0	0	-
	divided by 8	0	1	-	-	0	1	0	-
	divided by 16	0	1	-	11b	0	0	0	-
High-Speed Mode		0	0	-	00b	0	0	0	-
Medium-Speed Mode	divided by 2	0	0	-	01b	0	0	0	-
	divided by 4	0	0	-	10b	0	0	0	-
	divided by 8	0	0	-	-	0	1	0	-
	divided by 16	0	0	-	11b	0	0	0	-
Low-Speed Mode		-	0	-	-	1	-	0	1
Low Power Consumption Mode		0	0	-	-	1	1 (1)	1 (1)	1
125 kHz On-chip Oscillator Mode	divided by 1	1	0	0	00b	0	0	0	-
	divided by 2	1	0	0	01b	0	0	0	-
	divided by 4	1	0	0	10b	0	0	0	-
	divided by 8	1	0	0	-	0	1	0	-
	divided by 16	1	0	0	11b	0	0	0	-
125 kHz On-Chip Oscillator Low Power Consumption Mode		1	0	0	(2)	0	(2)	1	-

- indicates that either 0 or 1 is set.

**NOTES:**

- When the CM05 bit is set to 1 (main clock turned off) in low-speed mode, the mode goes to low power consumption mode and the CM06 bit is set to 1 (divided by 8 mode) simultaneously.
- The divide-by-n value can be selected the same way as in 125 kHz on-chip oscillator mode.

## 10.4.2 Wait Mode

In wait mode, the CPU clock is turned off, so are the CPU and the watchdog timer because they are operated by the CPU clock. However, if the CSPRO bit in the CSPR register is 1 (count source protection enabled), the watchdog timer remains active. Because the main clock, sub clock, and 125 kHz on-chip oscillator clock all are on, the peripheral functions using these clocks keep operating.

### 10.4.2.1 Peripheral Function Clock Stop Function

If the CM02 bit in the CM0 register is 1 (peripheral function clock f1 turned off during wait mode), the f1 clock is turned off while in wait mode, with the power consumption reduced that much. However, fC32 and fOCO-S (clock source of Timers A and B) remain on for the CM02 bit.

### 10.4.2.2 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 in the CM1 register 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 in the PLC0 register to 0 (PLL stops).

### 10.4.2.3 Pin Status during Wait Mode

Table 10.4 lists Pin Status during Wait Mode.

**Table 10.4 Pin Status during Wait Mode**

Pin		Memory Expansion Mode Microprocessor Mode	Single-Chip Mode
A0 to A19, D0 to D15, CS $\bar{0}$ to CS $\bar{3}$ , BHE $\bar{}$		Retains status just prior to entering wait mode	Cannot be used as a bus control pin
RD $\bar{}$ , WR $\bar{}$ , WRL $\bar{}$ , WRH $\bar{}$		"H"	
HLDA $\bar{}$ , BCLK		"H"	
ALE		"L"	
I/O ports		Retains status just prior to entering wait mode	Retains status before wait mode
CLKOUT	When fC selected	Cannot be used as a CLKOUT pin	Does not stop
	When f8, f32 selected		Does not stop when the CM02 bit is 0. When the CM02 bit is 1, the status immediately prior to entering wait mode is maintained

### 10.4.2.4 Exiting Wait Mode

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

If the microcomputer is to exit wait mode by a hardware reset,  $\overline{\text{NMI}}$  interrupt, or low voltage detection interrupt, set the peripheral function interrupt bits ILVL2 to ILVL0 to 000b (interrupts disabled) before executing the WAIT instruction.

The peripheral function interrupts are affected by the CM02 bit. If the CM02 bit is 0 (peripheral function clocks not turned off during wait mode), peripheral function interrupts can be used to exit wait mode. If the 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 activated by external signals can be used to exit wait mode.

**Table 10.5 Resets and Interrupts to Exit Wait Mode and Use Conditions**

Reset, Interrupt	CM02 = 0	CM02 = 1
NMI Interrupt	Usable	Usable
Serial Interface Interrupt	Usable when operating with internal or external clock	Usable when operating with external clock
Key Input Interrupt	Usable	Usable
A/D Conversion Interrupt	Usable in one-shot mode or single sweep mode	Do not use
Timer A Interrupt Timer B Interrupt	Usable in all modes	Usable in event counter mode or when the count source is fC32
$\overline{\text{INT}}$ Interrupt	Usable	Usable
Low Voltage Detection Interrupt	Usable	Usable
Hardware Reset 1	Usable	
Brown-out Reset	Usable (See 6.1 Brown-out Reset)	
Watchdog Timer Reset	Usable when count source protection mode is enabled (CSPRO = 1)	

Table 10.5 lists the Resets and Interrupts to Exit Wait Mode and Use Conditions.

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) Set bits ILVL2 to ILVL0 in the interrupt control register, for peripheral function interrupts used to exit wait mode.  
 Bits ILVL2 to ILVL0 in all other interrupt control registers, for peripheral function interrupts not used to exit wait mode, are set to 000b (interrupt disabled).
- (2) Set the I flag to 1.
- (3) Start operating the peripheral functions used to exit wait mode.  
 When the peripheral function interrupt is used, an interrupt routine is performed after an interrupt request is generated and then the CPU clock is supplied again.

When the microcomputer exits wait mode by the peripheral function interrupt, the CPU clock is the same clock as the CPU clock executing the WAIT instruction.



### 10.4.3 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 pins VCC1 and VCC2 is VRAM or greater, the internal RAM is retained. When applying 2.7 or less voltage to pins VCC1 and VCC2, make sure  $VCC1 = VCC2 \geq VRAM$ .

However, the peripheral functions activated by external signals keep operating. The following resets and interrupts can be used to exit stop mode. Table 10.6 lists Resets and Interrupts to Stop Mode and Use Conditions

**Table 10.6 Resets and Interrupts to Stop Mode and Use Conditions**

Reset, Interrupt	Condition
NMI Interrupt	Usable
Key Input Interrupt	Usable
INT Interrupt	Usable
Timer A Interrupt Timer B Interrupt	Usable when counting external pulses in event counter mode
Serial Interface Interrupt	Usable when external clock is selected
Low Voltage Detection Interrupt	Usable (See <b>6.2 Low Voltage Detection Interrupt</b> )
Hardware Reset 1	Usable
Brown-out Reset	Usable when digital filter is disabled (VW0C = 1)

#### 10.4.3.1 Entering Stop Mode

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

Before entering stop mode, set the CM20 bit in the CM2 register to 0 (oscillation stop, re-oscillation detection function disabled).

Also, if the CM11 bit in the CM1 register 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 in the PLC0 register to 0 (PLL turned off) before entering stop mode.

#### 10.4.3.2 Pin Status in Stop Mode

Table 10.7 lists Pin Status in Stop Mode.

**Table 10.7 Pin Status in Stop Mode**

Pin	Memory Expansion Mode Microprocessor Mode	Single-Chip Mode
A0 to A19, D0 to D15, CS0 to CS3, BHE	Retains status just prior to stop mode	Cannot be used as a bus control pin
RD, WR, WRL, WRH	"H"	
HLDA, BCLK	"H"	
ALE	indeterminate	
I/O ports	Retains status just prior to stop mode	Retains status just prior to stop mode
CLKOUT	Cannot be used as a CLKOUT pin	"H"

### 10.4.3.3 Exiting Stop Mode

Stop mode is exited by a hardware reset,  $\overline{\text{NMI}}$  interrupt, low voltage detection interrupt, or peripheral function interrupt.

When the hardware reset,  $\overline{\text{NMI}}$  interrupt, or low voltage detection interrupt is used to exit stop mode, set bits ILVL2 to ILVL0 in the interrupt control registers for the peripheral function interrupt to 000b (interrupt disabled) before setting the CM10 bit to 1.

When the peripheral function interrupt is used to exit stop mode, set the CM10 bit to 1 after the following settings are completed.

(1) Set bits ILVL2 to ILVL0 in the interrupt control registers to decide the peripheral priority level of the peripheral function interrupt.

Set the interrupt priority levels of the interrupts, not being used to exit stop mode, to 0 by setting bits ILVL2 to ILVL0 to 000b (interrupt disabled).

(2) Set the I flag to 1.

(3) Start operation of peripheral function being used to exit stop mode.

When exiting stop mode by the peripheral function interrupt, the interrupt routine is performed after an interrupt request is generated and then the CPU clock is supplied again.

When stop mode is exited by the peripheral function interrupt, low voltage detection interrupt, or  $\overline{\text{NMI}}$  interrupt, the CPU clock source is as follows, in accordance with the CPU clock source setting before the microcomputer had entered stop mode.

- When the sub clock is the CPU clock before entering stop mode: sub clock
- When the main clock is the CPU clock source before entering stop mode: main clock divided by 8
- When the 125 kHz on-chip oscillator clock is the CPU clock source before entering stop mode: 125 kHz on-chip oscillator clock divided by 8

Figure 10.12 shows the power Control Transition

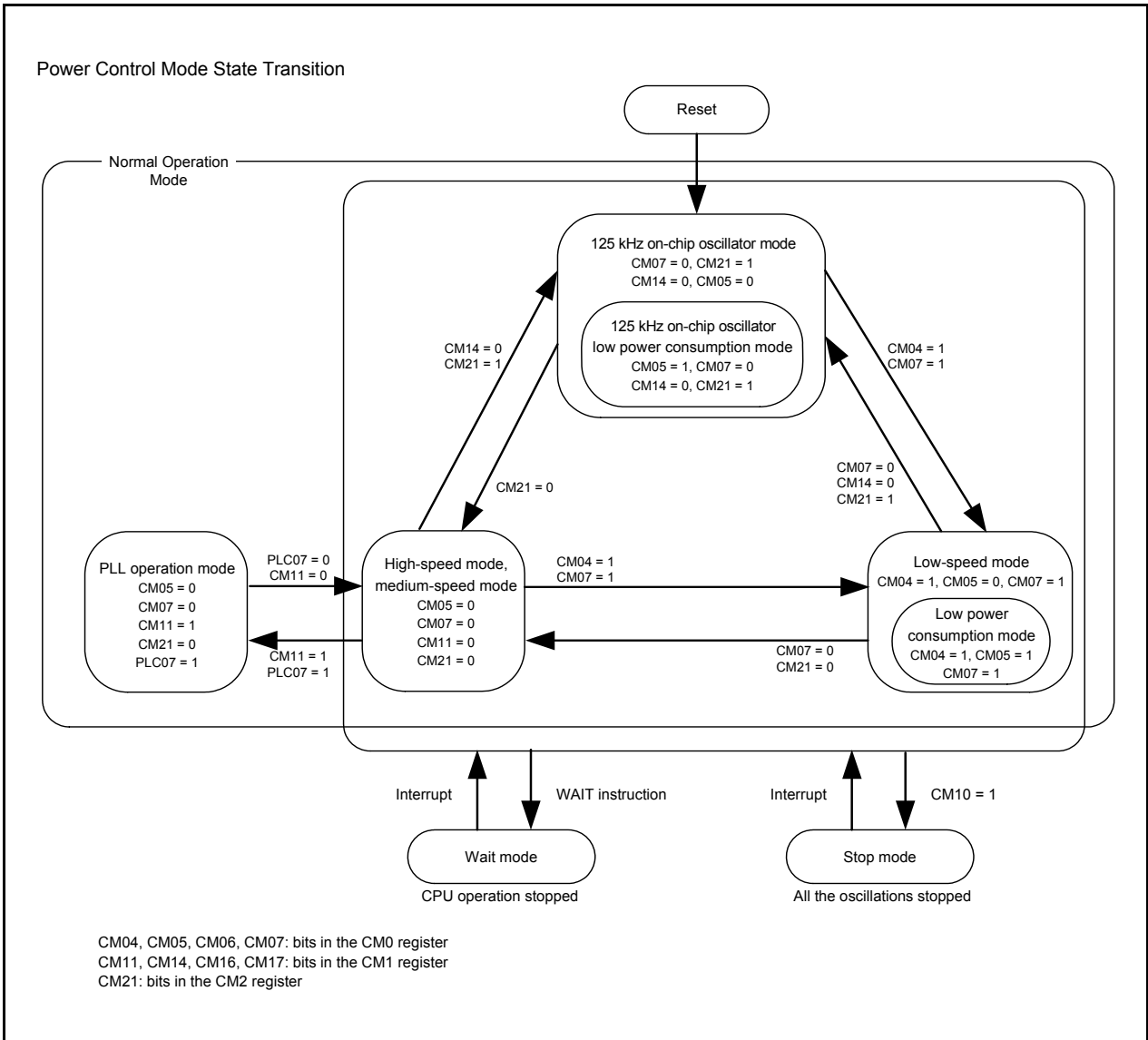


Figure 10.12 Power Control Transition

## 10.5 System Clock Protection Function

The system clock protection function prohibits the CPU clock from changing clock sources when the main clock is selected as the CPU clock source. This is to prevent the CPU clock from stopping by an unexpected program operation.

When the PM21 bit in the PM2 register is set to 1 (clock change disabled), the following bits cannot be written to:

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

When using the system clock protection function, set the CM05 bit in the CM0 register to 0 (main clock oscillation) and CM07 bit to 0 (main clock as CPU clock source) and follow the procedure below.

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

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

## 10.6 Oscillation Stop and Re-Oscillation Detect Function

The oscillation stop and re-oscillation detect function is such that main clock oscillation circuit stop and re-oscillation are detected. At oscillation stop or re-oscillation detection, reset oscillation stop or re-oscillation detection interrupt are generated. Which is to be generated can be selected using the CM27 bit in the CM2 register. The oscillation stop and re-oscillation detect function can be enabled and disabled by the CM20 bit in the CM2 register. Table 10.8 lists a Specification Overview of Oscillation Stop and Re-Oscillation Detect Function.

**Table 10.8 Specification Overview of Oscillation Stop and Re-Oscillation Detect Function**

Item	Specification
Oscillation Stop Detectable Clock and Frequency Bandwidth	$f(XIN) \geq 2 \text{ MHz}$
Enabling Condition for Oscillation Stop, Re-Oscillation Detect Function	Set CM20 bit to 1 (enabled)
Operation at Oscillation Stop, Re-Oscillation Detection	Reset occurs (when CM27 bit = 0) Oscillation stop, re-oscillation detection interrupt generated (when CM27 bit = 1)

### 10.6.1 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 4. “Special Function Registers (SFRs)”, 5. “Reset”).

This status is reset with hardware reset 1 or brown-out reset. 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).

### 10.6.2 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.
- CM14 bit = 0 (125 kHz on-chip oscillator clock oscillates)
- CM21 bit = 1 (125 kHz on-chip oscillator clock for CPU clock source and clock source of peripheral function.)
- 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 (125 kHz on-chip oscillator clock) inside the interrupt routine.

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM14 bit = 0 (125 kHz on-chip oscillator clock oscillates)
- 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.
- CM14 bit = 0 (125 kHz on-chip oscillator clock oscillates)
- CM22 bit = 1 (main clock re-oscillation detected)
- CM23 bit = 0 (main clock oscillation)
- CM21 bit remains unchanged

### 10.6.3 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 and low voltage detection 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.
- When the main clock re-oscillated after oscillation stop, the clock source for the CPU clock and peripheral functions must be switched to the main clock in a program. Figure 10.13 shows the Procedure to Switch Clock Source from 125 kHz On-chip Oscillator to Main Clock.
- Simultaneously with oscillation stop and re-oscillation detection interrupt occurrence, the CM22 bit becomes 1. When the CM22 bit is set to 1, oscillation stop and re-oscillation detection interrupt are disabled. By setting the CM22 bit to 0 in a program, oscillation stop and re-oscillation detection interrupt are enabled.
- If the main clock stops during low speed mode where the CM20 bit is 1, an oscillation stop and re-oscillation detection interrupt request is generated. At the same time, the 125 kHz on-chip 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 125 kHz on-chip oscillator clock.
- To enter wait mode while using the oscillation stop and re-oscillation detection function, set the CM02 bit to 0 (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop and re-oscillation detection function is provided in preparation for main clock stop due to external factors, set the CM20 bit to 0 (oscillation stop and re-oscillation detection function disabled) where the main clock is stopped or oscillated in a 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.

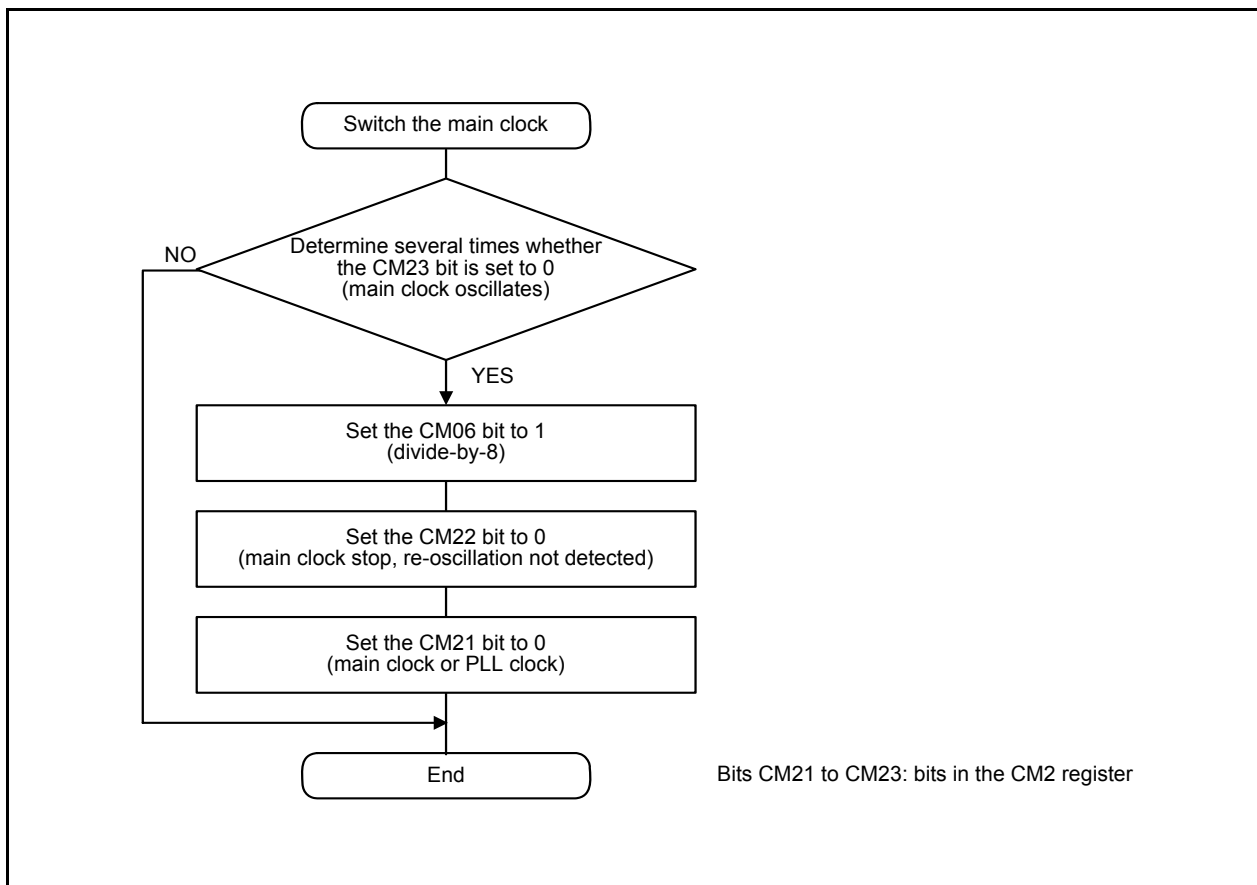


Figure 10.13 Procedure to Switch Clock Source From 125 kHz On-chip Oscillator to Main Clock

## 11. 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 11.1 shows the PRCR Register. The following lists the registers protected by the PRCR register.

- The PRC0 bit protects registers CM0, CM1, CM2, PLC0, and PCLKR.
- The PRC1 bit protects registers PM0, PM1, PM2, TB2SC, INVC0, and INVC1.
- The PRC2 bit protects registers PD9, S3C, and S4C.
- The PRC3 bit protects registers VCR2, D4INT, and VW0C.
- The PRC6 bit protects the PRG2C register.

Set the PRC2 bit to 1 (write enabled) and then write to given SFR 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. Bits PRC0, PRC1, PRC3, and PRC6 are not automatically cleared to 0 by writing to given SFR address. They can only be cleared in a program.

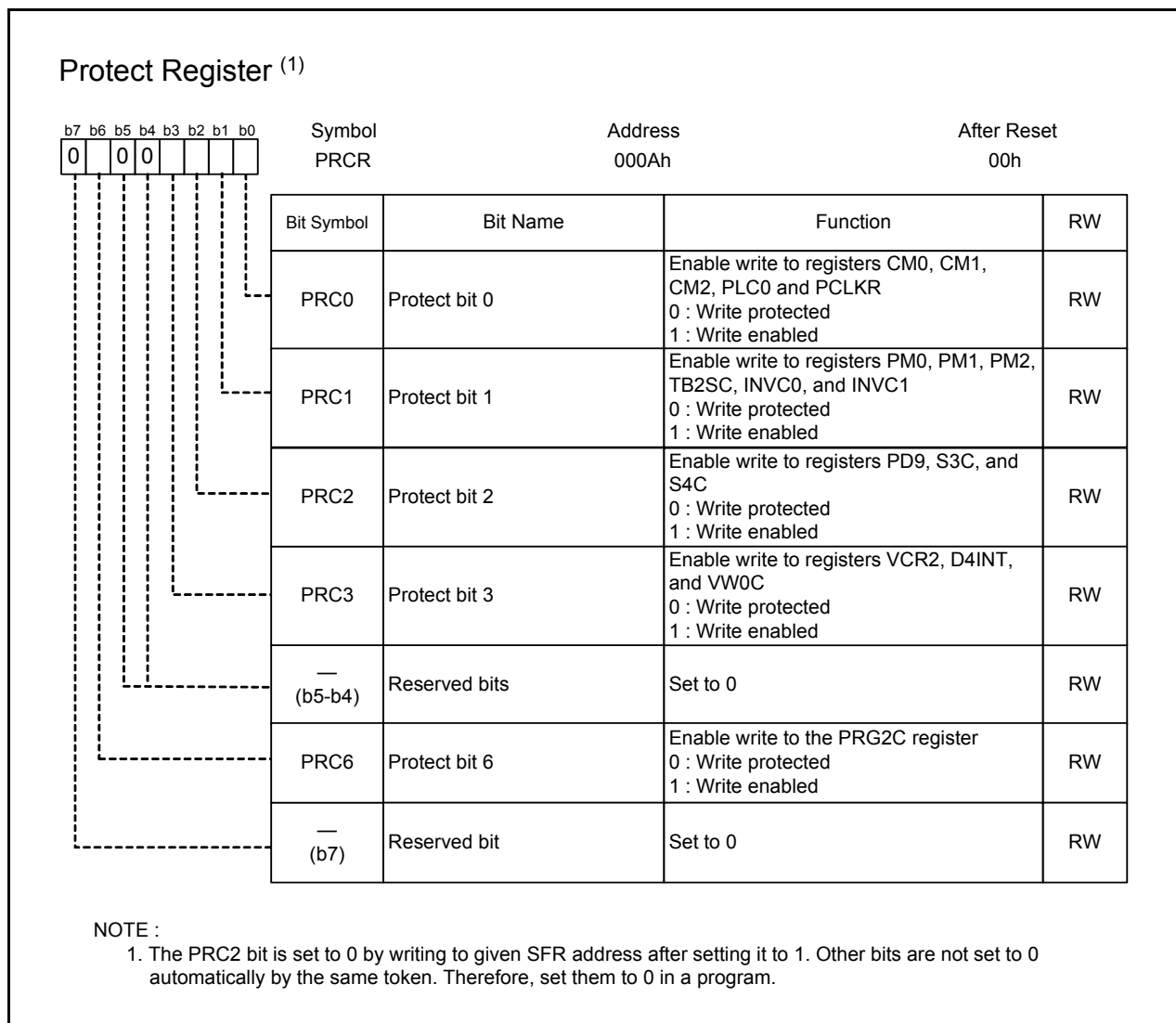


Figure 11.1 PRCR Register

## 12. Interrupt

### 12.1 Type of Interrupts

Figure 12.1 shows Type of Interrupts.

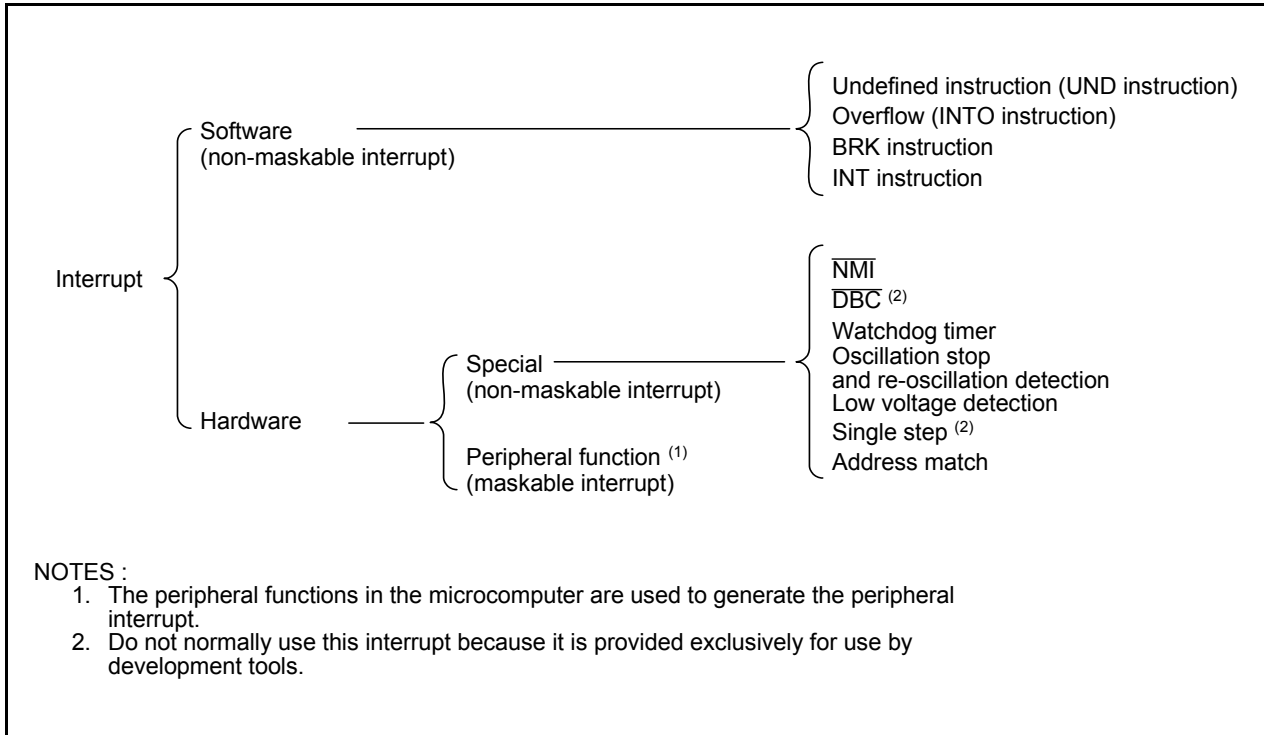


Figure 12.1 Type of Interrupts

- Maskable Interrupt : The interrupt priority **can be changed** by enabling (disabling) an interrupt with the interrupt enable flag (I flag) or by using interrupt priority levels.
- Non-Maskable Interrupt : The interrupt priority **cannot be changed** by enabling (disabling) an interrupt with the interrupt enable flag (I flag) or by using interrupt priority levels.



## 12.2 Software Interrupts

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

### 12.2.1 Undefined Instruction Interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

### 12.2.2 Overflow Interrupt

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

### 12.2.3 BRK Interrupt

A BRK interrupt occurs when executing the BRK instruction.

### 12.2.4 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. 2 to 31 and 41 to 51 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 selected at the time is used.

## 12.3 Hardware Interrupts

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

### 12.3.1 Special Interrupts

Special interrupts are non-maskable interrupts.

#### 12.3.1.1 $\overline{\text{NMI}}$ 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 **12.7 “ $\overline{\text{NMI}}$  Interrupt”**.

#### 12.3.1.2 $\overline{\text{DBC}}$ Interrupt

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

#### 12.3.1.3 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 **13. “Watchdog Timer”**.

#### 12.3.1.4 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 **10. “Clock Generation Circuit”**.

#### 12.3.1.5 Low Voltage Detection Interrupt

Generated by the voltage detection circuit. For details about the voltage detection circuit, refer to **6. “Voltage Detection Circuit”**.

#### 12.3.1.6 Single-Step Interrupt

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

#### 12.3.1.7 Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by registers RMAD0 to RMAD3 that correspond to one of the AIER0 or AIER1 bit in the AIER register or the AIER20 or AIER21 bit in the AIER2 register which is 1 (address match interrupt enabled). For details about the address match interrupt, refer to **12.9 “Address Match Interrupt”**.

### 12.3.2 Peripheral Function Interrupts

The peripheral function interrupt occurs when a request from the peripheral functions in the microcomputer is acknowledged. The peripheral function interrupt is a maskable interrupt. See Tables 12.2 and 12.3 Relocatable Vector Tables. Refer to the descriptions of each function for details about how the peripheral function interrupt occurs.

## 12.4 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 12.2 shows the Interrupt Vector.

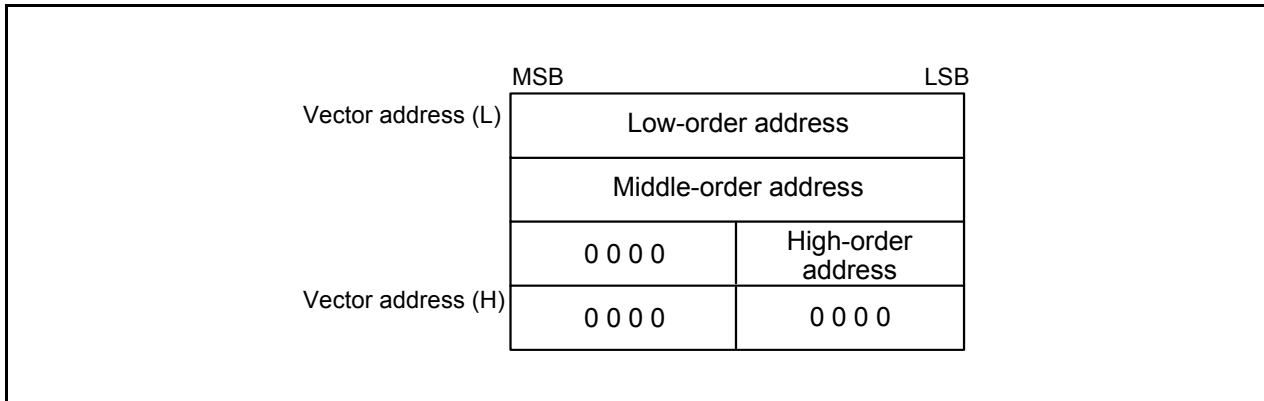


Figure 12.2 Interrupt Vector

### 12.4.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDCh to FFFFFh. Table 12.1 lists the Fixed Vector Table. 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 22.2 “Functions to Prevent Flash Memory from Rewriting”.

Table 12.1 Fixed Vector Table

Interrupt Source	Vector Table Addresses Address (L) to Address (H)	Reference
Undefined Instruction (UND instruction)	FFFDCh to FFFDFh	M16C/60, M16C/20 series software manual
Overflow (INTO instruction)	FFFE0h to FFFE3h	
BRK Instruction (2)	FFFE4h to FFFE7h	
Address Match	FFFE8h to FFFEBh	<b>12.9 “Address Match Interrupt”</b>
Single Step (1)	FFFECh to FFFEFh	-
Watchdog Timer, Oscillation Stop and Re-Oscillation Detection, Low Voltage Detection	FFFF0h to FFFF3h	<b>13. “Watchdog Timer”</b> <b>10. “Clock Generation Circuit”</b> <b>6. “Voltage Detection Circuit”</b>
$\overline{DBC}$ (1)	FFFF4h to FFFF7h	-
NMI	FFFF8h to FFFFBh	<b>12.7 “NMI Interrupt”</b>
Reset	FFFFCh to FFFFFh	<b>5. “Reset”</b>

NOTES:

1. Do not normally use this interrupt because it is provided exclusively for use by development tools.
2. If the contents of address FFFE7h is FFh, program execution starts from the address shown by the vector in the relocatable vector table.

## 12.4.2 Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a relocatable vector table area. Tables 12.3 and 12.3 list the Relocatable Vector Tables. Setting an even address in the INTB register results in the interrupt sequence being executed faster than setting an odd address.

**Table 12.2 Relocatable Vector Table (1)**

Interrupt Source	Vector Address (1) Address (L) to Address (H)	Software Interrupt Number	Reference
BRK Instruction (5)	+0 to +3 (0000h to 0003h)	0	M16C/60, M16C/20 series software manual
– (Reserved)		1	
INT7	+8 to +11 (0008h to 000Bh)	2	12.6 “INT Interrupt”
INT6	+12 to +15 (000Ch to 000Fh)	3	
INT3	+16 to +19 (0010h to 0013h)	4	
Timer B5	+20 to +23 (0014h to 0017h)	5	15. “Timers”
Timer B4, UART1 Bus Collision Detect (4, 6)	+24 to +27 (0018h to 001Bh)	6	15. “Timers” 17. “Serial Interface”
Timer B3, UART0 Bus Collision Detect (4, 6)	+28 to +31 (001Ch to 001Fh)	7	
SI/O4, INT5 (2)	+32 to +35 (0020h to 0023h)	8	12.6 “INT Interrupt” 17. “Serial Interface”
SI/O3, INT4 (2)	+36 to +39 (0024h to 0027h)	9	
UART2 Bus Collision Detection (6)	+40 to +43 (0028h to 002Bh)	10	17. “Serial Interface”
DMA0	+44 to +47 (002Ch to 002Fh)	11	14. “DMAC”
DMA1	+48 to +51 (0030h to 0033h)	12	
Key Input Interrupt	+52 to +55 (0034h to 0037h)	13	12.8 “Key Input Interrupt”
A/D	+56 to +59 (0038h to 003Bh)	14	18. “A/D Converter”
UART2 Transmit, NACK2 (3)	+60 to +63 (003Ch to 003Fh)	15	17. “Serial Interface”
UART2 Receive, ACK2 (3)	+64 to +67 (0040h to 0043h)	16	
UART0 Transmit, NACK0 (3)	+68 to +71 (0044h to 0047h)	17	
UART0 Receive, ACK0 (3)	+72 to +75 (0048h to 004Bh)	18	
UART1 Transmit, NACK1 (3)	+76 to +79 (004Ch to 004Fh)	19	
UART1 Receive, ACK1 (3)	+80 to +83 (0050h to 0053h)	20	
Timer A0	+84 to +87 (0054h to 0057h)	21	15. “Timers”
Timer A1	+88 to +91 (0058h to 005Bh)	22	
Timer A2	+92 to +95 (005Ch to 005Fh)	23	
Timer A3	+96 to +99 (0060h to 0063h)	24	
Timer A4	+100 to +103 (0064h to 0067h)	25	
Timer B0	+104 to +107 (0068h to 006Bh)	26	
Timer B1	+108 to +111 (006Ch to 006Fh)	27	
Timer B2	+112 to +115 (0070h to 0073h)	28	

NOTES:

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

Table 12.3 Relocatable Vector Table (2)

Interrupt Source	Vector Address (1) Address (L) to Address (H)	Software Interrupt Number	Reference
$\overline{\text{INT0}}$	+116 to +119 (0074h to 0077h)	29	<b>12.6 “<math>\overline{\text{INT}}</math> Interrupt”</b>
$\overline{\text{INT1}}$	+120 to +123 (0078h to 007Bh)	30	
$\overline{\text{INT2}}$	+124 to +127 (007Ch to 007Fh)	31	
INT Instruction Interrupt (3)	+128 to +131 (0080h to 0083h) to +160 to +163 (00A0h to 00A3h)	32 to 40	M16C/60, M16C/20 series software manual
DMA2	+164 to +167 (00A4h to 00A7h)	41	<b>14. “DMAC”</b>
DMA3	+168 to +171 (00A8h to 00ABh)	42	
UART5 Bus Collision Detec- tion(4)	+172 to +175 (00ACh to 0AFh)	43	<b>17. “Serial Interface”</b>
UART5 Transmit, NACK5(2)	+176 to +179 (00B0h to 00B3h)	44	
UART5 Receive, ACK5 (2)	+180 to +183 (00B4h to 00B7h)	45	
UART6 Bus Collision Detec- tion (4)	+184 to +187 (00B8h to 00BBh)	46	
UART6 Transmit, NACK6 (2)	+188 to +191 (00BCh to 00BFh)	47	
UART6 Receive, ACK6 (2)	+192 to +195 (00C0h to 00C3h)	48	
UART7 Bus Collision Detec- tion (4)	+196 to +199 (00C4h to 00C7h)	49	
UART7 Transmit, NACK7 (2)	+200 to +203 (00C8h to 00CBh)	50	
UART7 Receive, ACK7 (2)	+204 to +207 (00CCh to 00CFh)	51	
- (Reserved)		52 to 63	

## NOTES:

1. Address relative to address in INTB.
2. During I<sup>2</sup>C mode, interrupts NACK and ACK comprise the interrupt source.
3. These interrupts cannot be disabled using the I flag.
4. Bus collision detection: During IE mode, this bus collision detection constitutes the factor of an interrupt.
5. Bus collision detection: During IE mode, this bus collision detection constitutes the interrupt source. During I<sup>2</sup>C mode, however, a start condition or a stop condition detection constitutes the interrupt source.

### 12.5 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 I flag in the FLG register, IPL, and bits ILVL2 to ILVL0 in each interrupt control register to enable / disable the maskable interrupts. Whether an interrupt is requested or not is indicated by the IR bit in each interrupt control register.

Figures 12.3 and 12.4 show the Interrupt Control Registers.

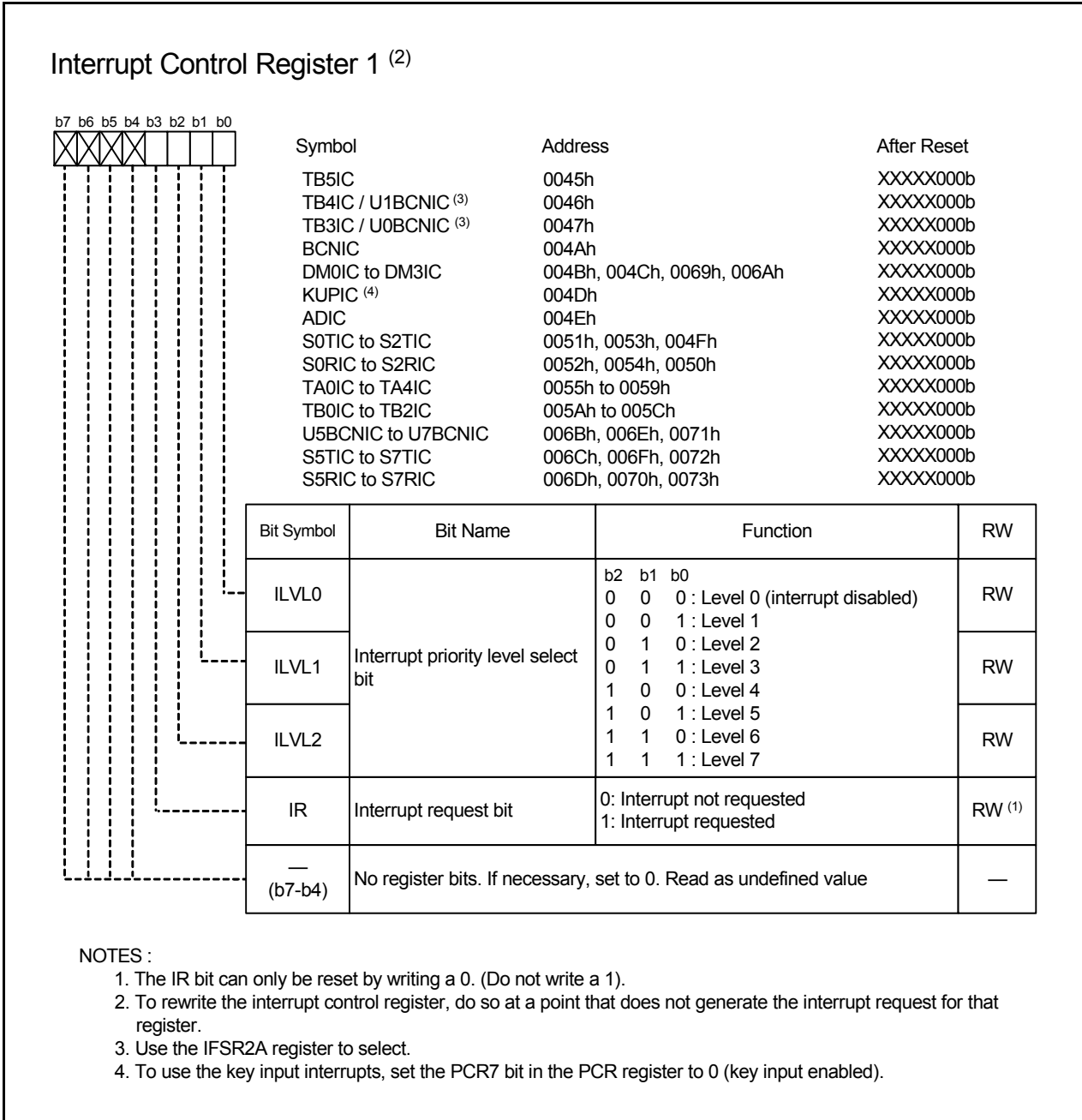


Figure 12.3 Interrupt Control Register (1)

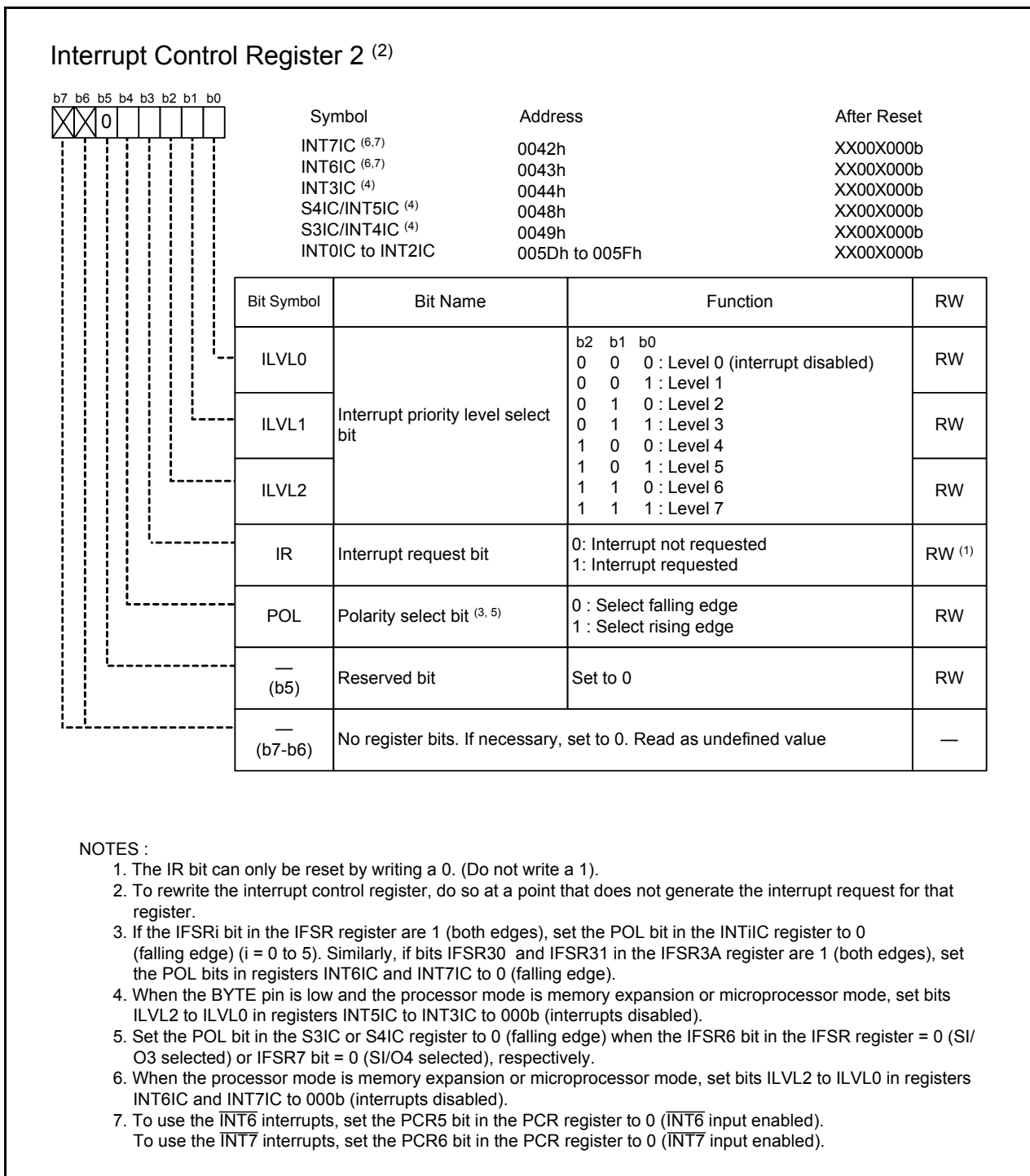


Figure 12.4 Interrupt Control Register (2)

### 12.5.1 I Flag

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

### 12.5.2 IR Bit

The IR bit is set to 1 (interrupt requested) when an interrupt request is generated. Then, when the interrupt request is accepted, the IR bit is cleared to 0 (interrupt not requested).

The IR bit can be cleared to 0 in a program. Do not write a 1 to this bit.

### 12.5.3 Bits ILVL2 to ILVL0 and IPL

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


Table 12.4 shows the Settings of Interrupt Priority Levels and Table 12.5 shows the Interrupt Priority Levels Enabled by IPL.

The followings are conditions under which an interrupt is accepted:

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

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

**Table 12.4 Settings of Interrupt Priority Levels**

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

**Table 12.5 Interrupt Priority Levels Enabled by IPL**

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



### 12.5.4 Interrupt Sequence

An interrupt sequence – what are performed over a period from the instant an interrupt request is accepted to the instant the interrupt routine is executed – is described here.

If an interrupt request 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.

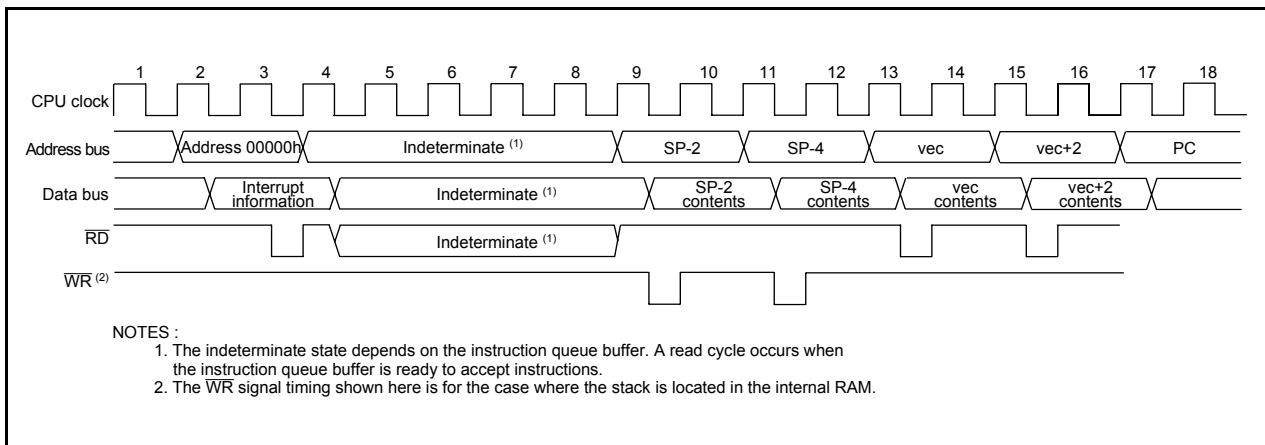
The CPU behavior during the interrupt sequence is described below. Figure 12.5 shows Time Required for Executing Interrupt Sequence.

- (1) The CPU obtains interrupt information (interrupt number and interrupt request level) by reading address 00000h. Then, the IR bit applicable to the interrupt information is set to 0 (interrupt not requested).
- (2) The FLG register, prior to an interrupt sequence, is saved to a temporary register (1) within the CPU.
- (3) Flags I, D, and U in the FLG register become as follows:
  - The I flag is set to 0 (interrupt disabled)
  - The D flag is set to 0 (single-step interrupt disabled)
  - The U flag is set to 0 (ISP selected)
 Note that the U flag does not change states if an INT instruction for software interrupt Nos. 32 to 63 is executed.
- (4) The temporary register (1) within the CPU is saved to the stack.
- (5) The PC is saved to the stack.
- (6) The interrupt priority level of the acknowledged interrupt in IPL is set.
- (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, an instruction is executed from the starting address of the interrupt routine.

**NOTE:**

1. Temporary register cannot be modified by users.



**Figure 12.5 Time Required for Executing Interrupt Sequence**

### 12.5.5 Interrupt Response Time

Figure 12.6 shows the Interrupt Response Time. The interrupt response or interrupt acknowledge time denotes a time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of a time from when an interrupt request is generated till when the executing instruction is completed ((a) on Figure 12.6) and a time during which the interrupt sequence is executed ((b) on Figure 12.6).

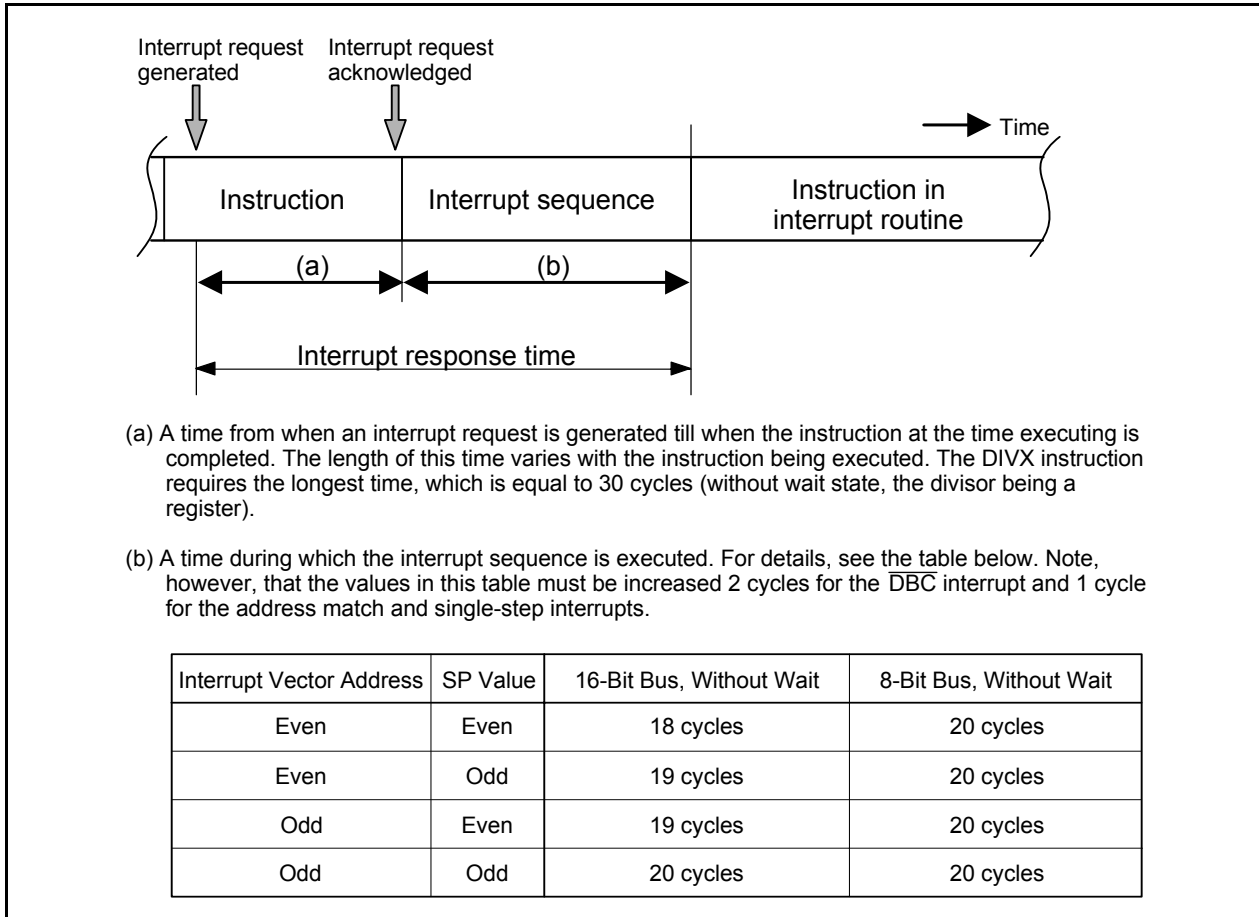


Figure 12.6 Interrupt Response Time

### 12.5.6 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 12.6 is set in the IPL. Table 12.6 lists the IPL Level That is Set to IPL When a Software or Special Interrupt is Accepted.

Table 12.6 IPL Level That is Set to IPL When a Software or Special Interrupt is Accepted

Interrupt Sources	Level Set to IPL
Watchdog Timer, NMI, Oscillation Stop and Re-Oscillation Detection, Low Voltage Detection	7
Software, Address Match, DBC, Single-Step	Not changed

### 12.5.7 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 in 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 12.7 shows the Stack Status Before and After Acceptance of Interrupt Request. 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.

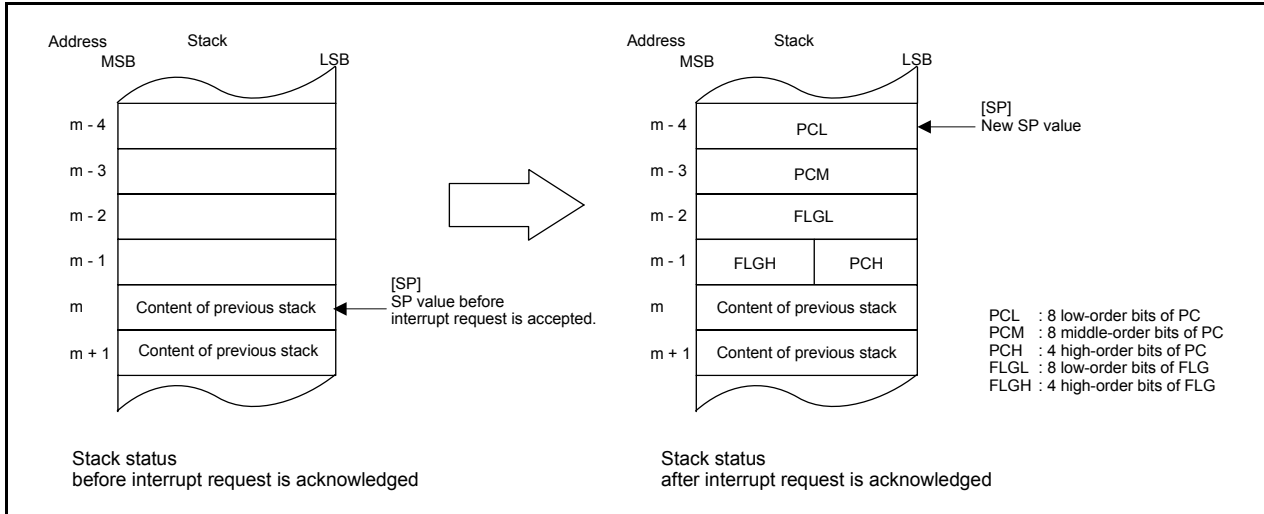


Figure 12.7 Stack Status Before and After Acceptance of Interrupt Request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP (1), at the time of acceptance of an interrupt request, is even or odd. If the SP (1) 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 12.8 shows the Operation of Saving Register.

NOTE:

1. 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.

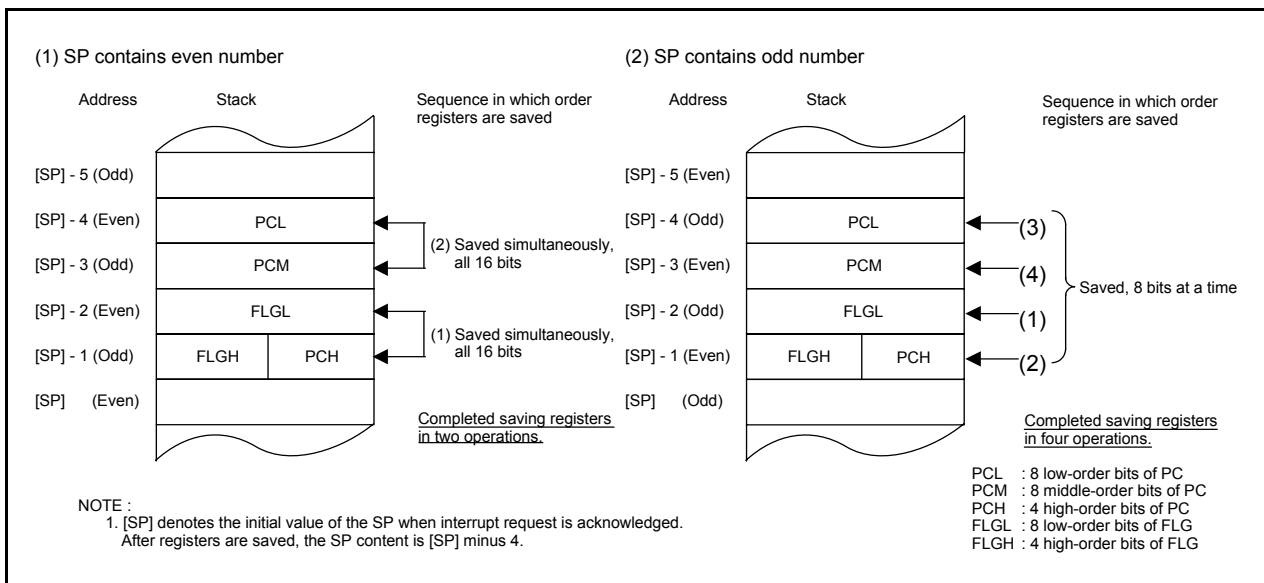


Figure 12.8 Operation of Saving Register

### 12.5.8 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.

Register bank is switched back to the bank used prior to the interrupt sequence by the REIT instruction.

### 12.5.9 Interrupt Priority

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

For maskable interrupts (peripheral functions interrupt), any desired priority level can be selected using bits ILVL2 to ILVL0. 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 12.9 shows the Hardware Interrupt Priority.

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

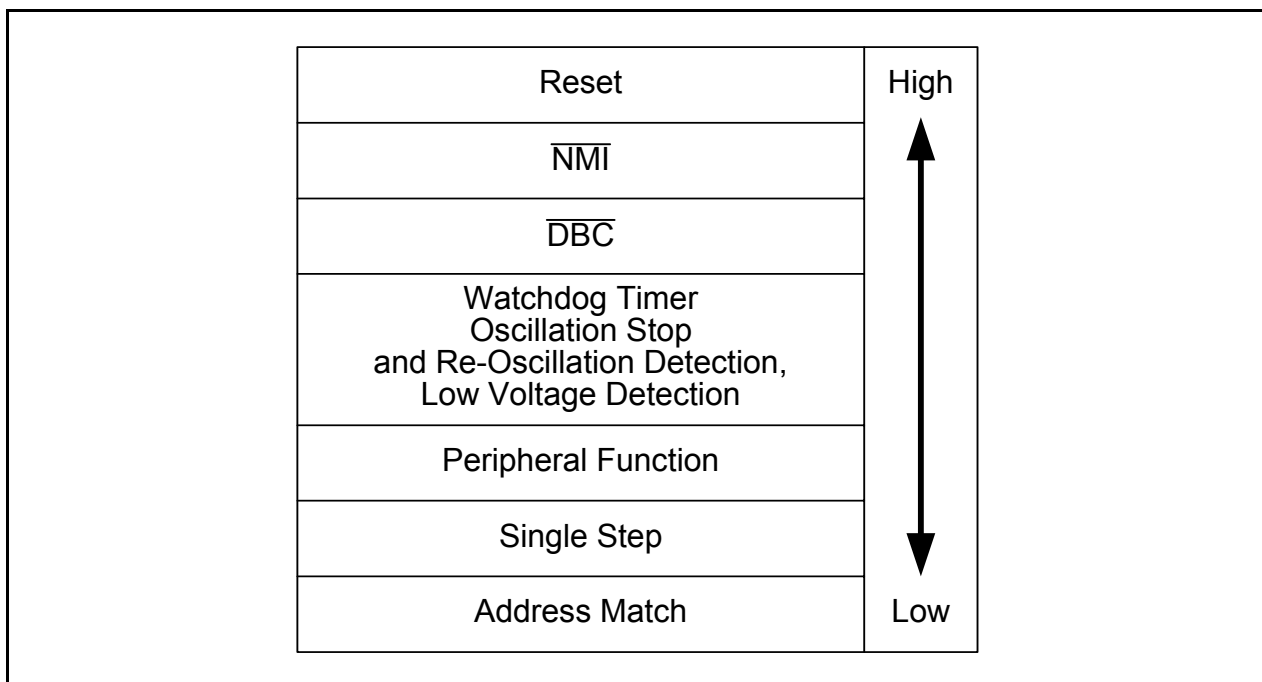


Figure 12.9 Hardware Interrupt Priority

### 12.5.10 Interrupt Priority Level Select Circuit

The interrupt priority level select circuit selects the highest priority interrupt in a sampled interrupt request(s) at the same sampling point.

Figure 12.10 shows the Interrupts Priority Select Circuit.

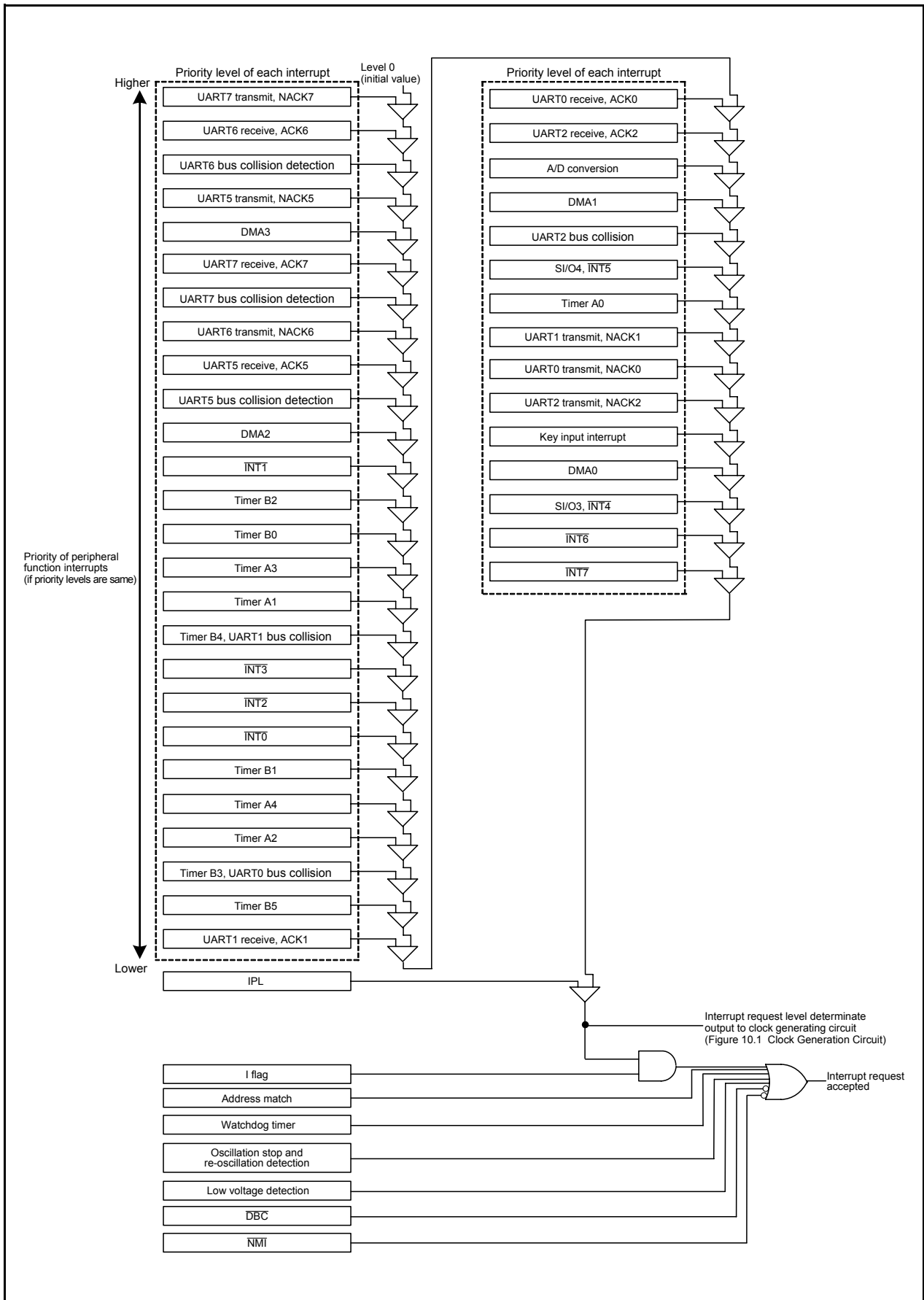


Figure 12.10 Interrupts Priority Select Circuit

## 12.6 INT Interrupt

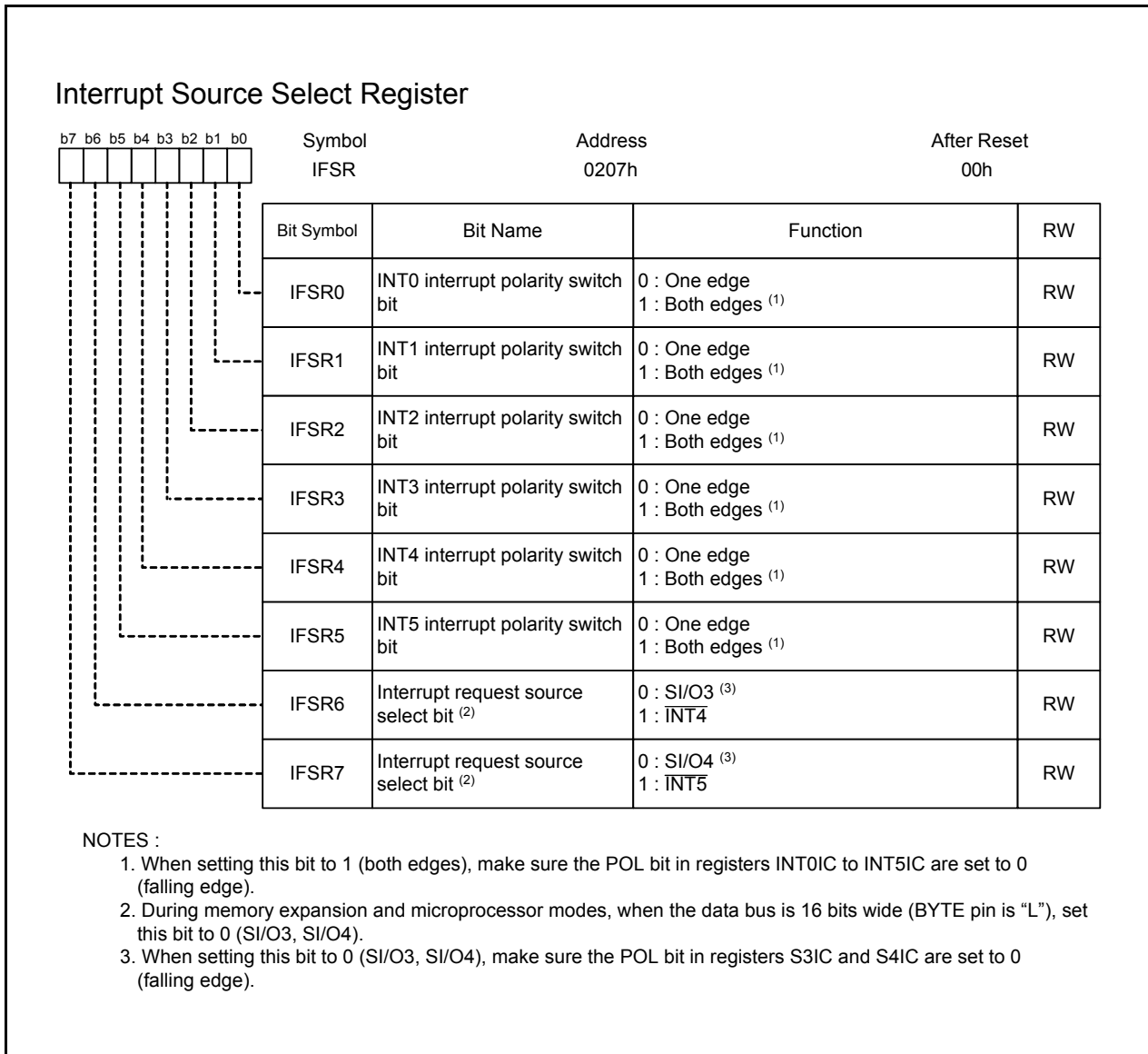
$\overline{\text{INT}}_i$  interrupt ( $i = 0$  to  $7$ ) is triggered by the edges of external inputs. The edge polarity is selected using the IFSR $_i$  bit in the IFSR register, or the IFSR30 or IFSR31 bit in the IFSR3A register.

INT4 and INT5 share the interrupt vector and interrupt control register with SI/O3 and SI/O4, respectively. To use the  $\overline{\text{INT}}_4$  interrupt, set the IFSR6 bit in the IFSR register to 1 ( $\overline{\text{INT}}_4$ ). To use the  $\overline{\text{INT}}_5$  interrupt, set the IFSR7 bit in the IFSR register to 1 ( $\overline{\text{INT}}_5$ ).

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

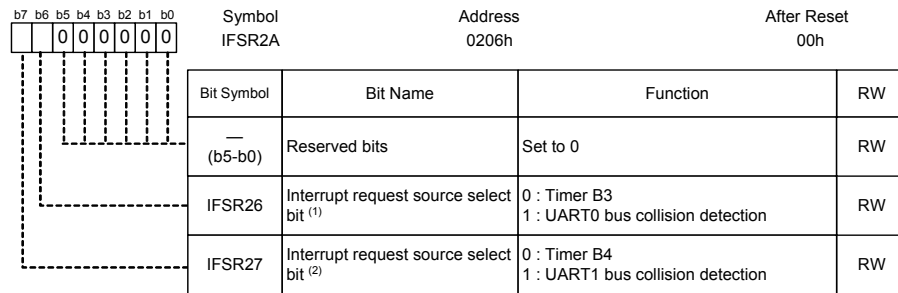
To use the  $\overline{\text{INT}}_6$  interrupt, set the PCR5 bit in the PCR register to 0 ( $\overline{\text{INT}}_6$  input enabled). To use the  $\overline{\text{INT}}_7$  interrupt, set the PCR6 bit in the PCR register to 0 ( $\overline{\text{INT}}_7$  input enabled).

Figure 12.11 shows the IFSR Register, and Figure 12.12 shows Registers IFSR2A, IFSR3A, and PCR.



**Figure 12.11 IFSR Register**

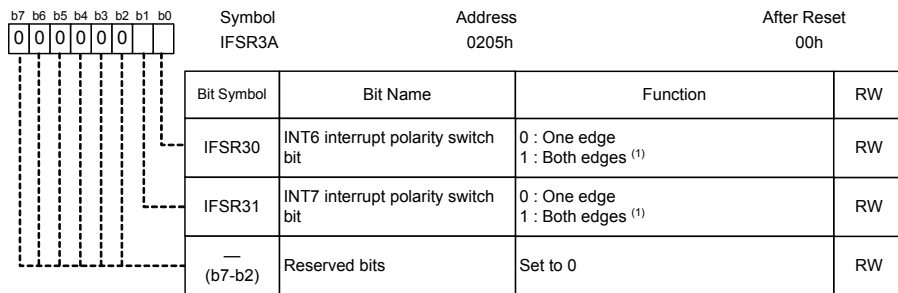
### Interrupt Source Select Register 2



NOTES :

1. Timer B3 and UART0 bus collision detection share the vector and interrupt control register. When using Timer B3 interrupt, clear the IFSR26 bit to 0 (Timer B3). When using UART0 bus collision detection, set the IFSR26 bit to 1.
2. Timer B4 and UART1 bus collision detection share the vector and interrupt control register. When using Timer B4 interrupt, clear the IFSR27 bit to 0 (Timer B4). When using UART1 bus collision detection, set the IFSR27 bit to 1.

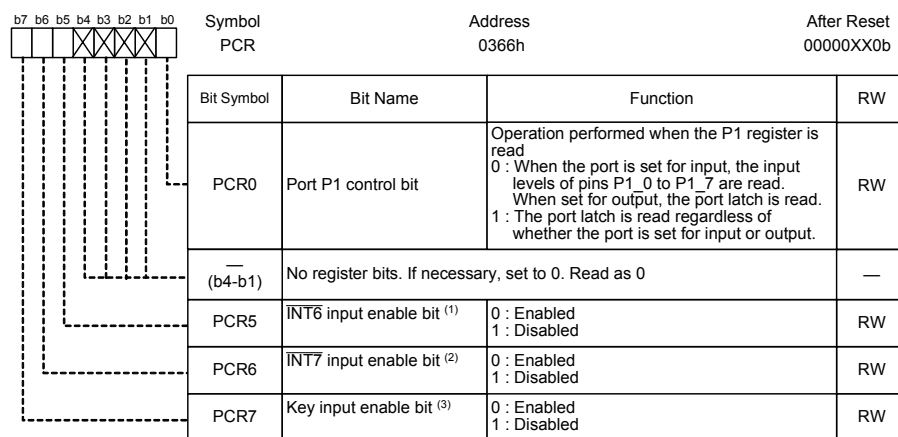
### Interrupt Source Select Register 3



NOTE :

1. When setting this bit to 1 (both edges), make sure the POL bit in registers INT6IC and INT7IC are set to 0 (falling edge).

### Port Control Register



NOTES :

1. To use the AN2\_4 pin as an analog input pin, set the PCR5 bit to 1 (INT6 input disabled).
2. To use the AN2\_5 pin as an analog input pin, set the PCR6 bit to 1 (INT7 input disabled).
3. To use pins AN4 to AN7 as analog input pins, set the PCR7 bit to 1 (key input disabled).

Figure 12.12 Registers IFSR2A, IFSR3A, and PCR

### 12.7 $\overline{\text{NMI}}$ Interrupt

An  $\overline{\text{NMI}}$  interrupt is generated when input on the  $\overline{\text{NMI}}$  pin changes state from high to low. The  $\overline{\text{NMI}}$  interrupt is a non-maskable interrupt. To use the  $\overline{\text{NMI}}$  interrupt, set the PM24 bit in the PM2 register to 1 (NMI function).

### 12.8 Key Input Interrupt

Of P10\_4 to P10\_7, a key input interrupt is generated when input on any of pins P10\_4 to P10\_7 which has had bits PD10\_4 to PD10\_7 in the PD10 register set to 0 (input) goes low. Key input interrupts can be used as a key-on wake up function, the function which gets the microcomputer out of wait or stop mode. However, if using the key input interrupt, do not use P10\_4 to P10\_7 as analog input pins. Figure 12.13 shows the block diagram of the Key Input Interrupt. Note, however, that while input on any pin which has had bits PD10\_4 to PD10\_7 set to 0 (input mode) is pulled low, inputs on all other pins of the port are not detected as interrupts. Set the PCR7 bit in the PCR register to 0 (key input enabled) to use key input interrupts.

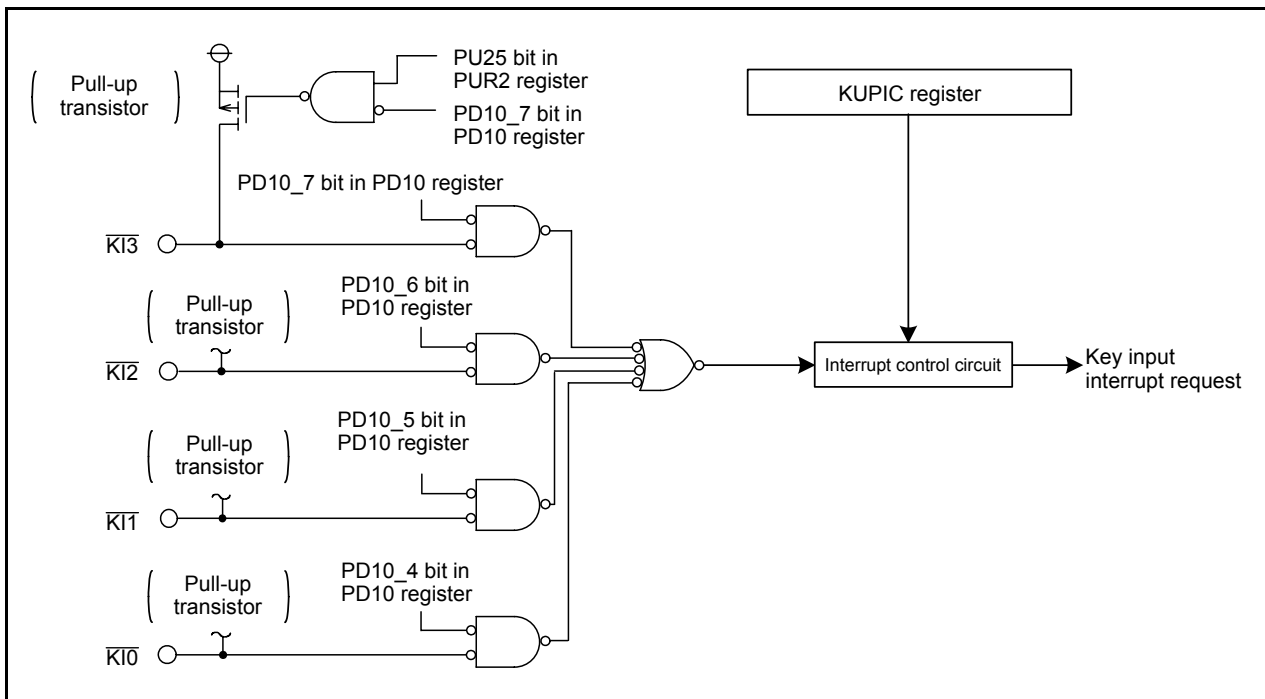


Figure 12.13 Key Input Interrupt



## 12.9 Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by the RMADi register (i = 0 to 3). Set the start address of any instruction in the RMADi register. Use bits AIER0 and AIER1 in the AIER register and bits AIER20 and AIER21 in the AIER2 register to enable or disable the interrupt. Note that the address match interrupt is unaffected by the I flag and IPL. When address match interrupt requests are acknowledged, the value of the PC that is saved to the stack area (refer to **12.5.7 “Saving Registers”**) varies depending on the instruction at the address indicated by the RMADi register (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 12.7 shows the Value of the PC That Is Saved to the Stack Area When an Address Match Interrupt Request is Accepted. Note that when using the external bus in 8 bits width, no address match interrupts can be used for external areas.

Figure 12.14 shows Registers AIER, AIER2, and RMAD0 to RMAD3.

**Table 12.7 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 Instruction shown below among 8-bit operation code instructions ADD.B:S #IMM8, dest SUB.B:S #IMM8, dest AND.B:S #IMM8, dest OR.B:S #IMM8, dest MOV.B:S #IMM8, dest STZ.B:S #IMM8, dest STNZ.B:S #IMM8, dest STZX.B:S #IMM81, #IMM82,dest CMP.B:S #IMM8, dest PUSHM src POPM dest JMPS #IMM8 JSRS #IMM8 MOV.B:S #IMM, dest (However, dest = A0 or A1)	The address indicated by the RMADi register +2
Instructions other than the above	The address indicated by the RMADi register +1

NOTE:

Value of the PC that is saved to the stack area: Refer to **12.5.7 “Saving Registers”**.

**Table 12.8 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 2	AIER20	RMAD2
Address Match Interrupt 3	AIER21	RMAD3

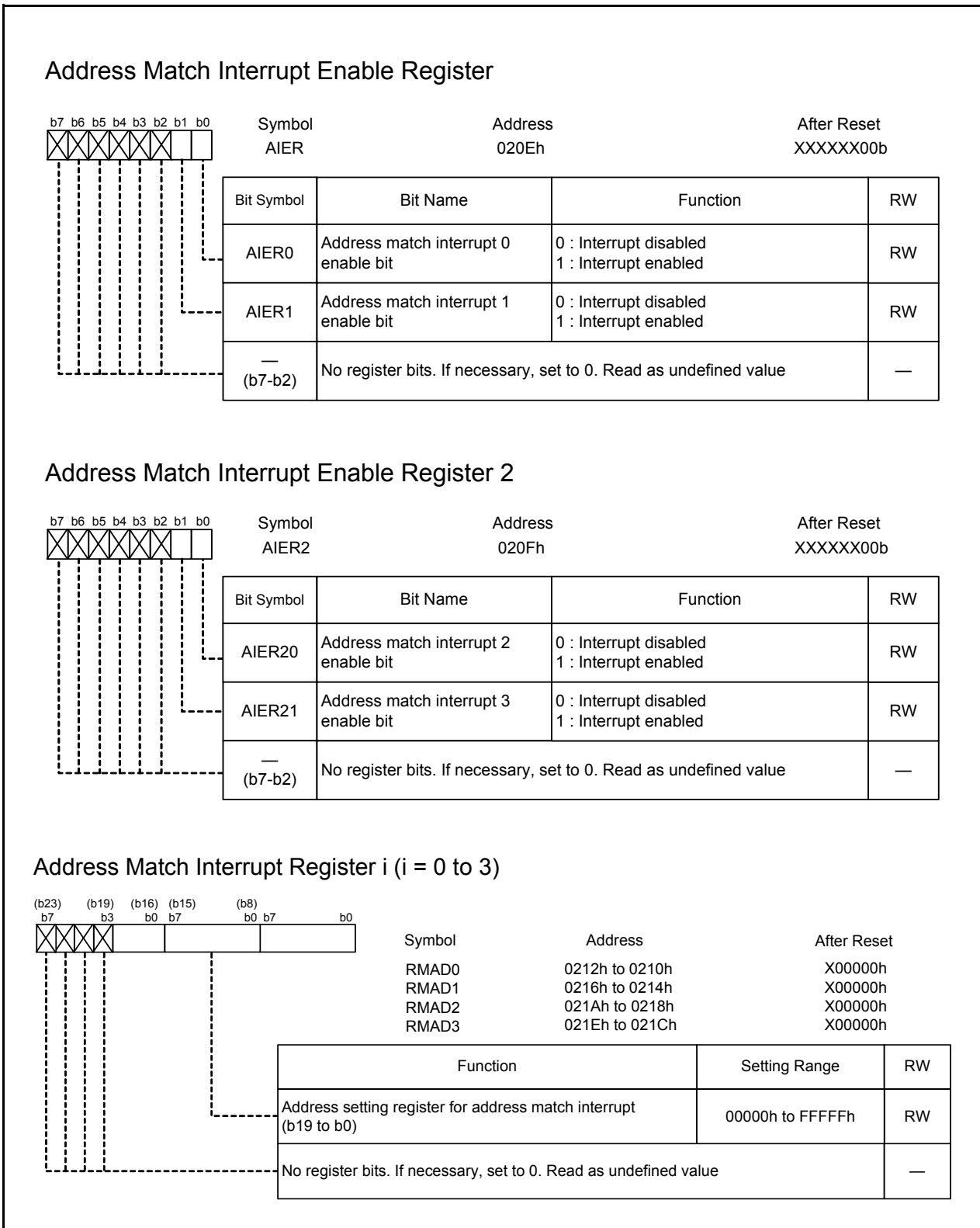


Figure 12.14 Registers AIER, AIER2, and RMAD0 to RMAD3

### 13. Watchdog Timer

The watchdog timer detects whether 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, and count source protection mode (enabled / disabled) is set here.

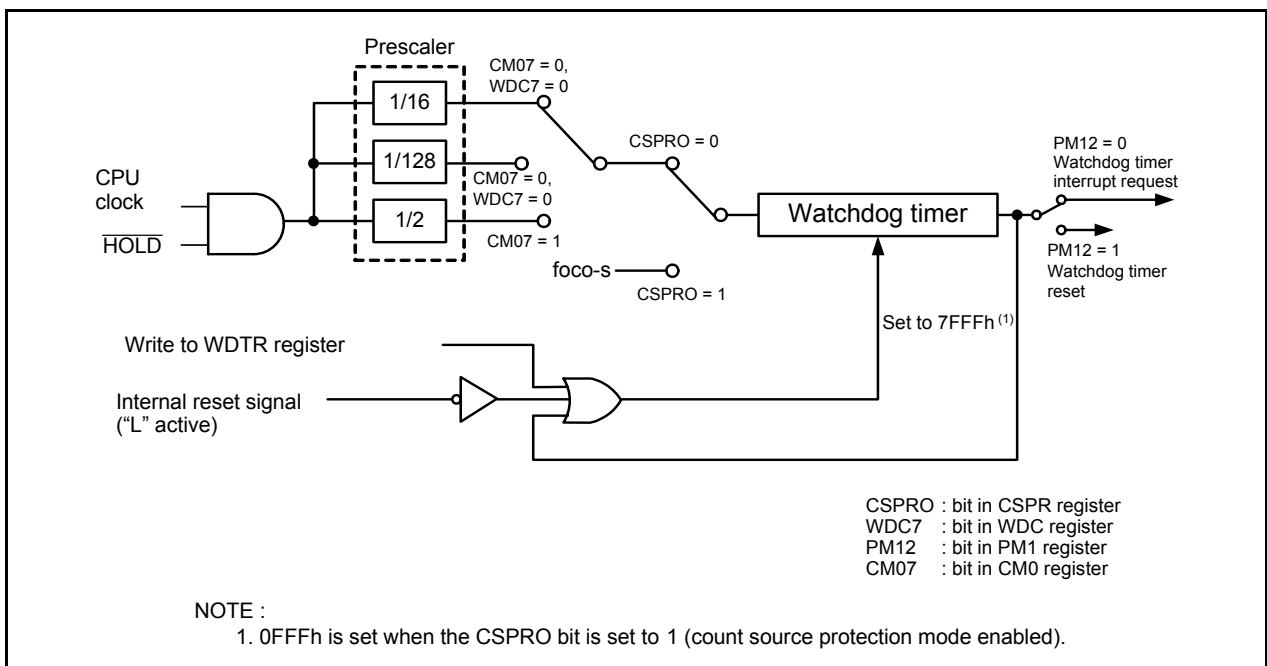
Table 13.1 shows the Watchdog Timer Specification.

Refer to 5.4 "Watchdog Timer Reset" for details of watchdog timer reset.

Figure 13.1 shows the Watchdog Timer Block Diagram. Figure 13.2 shows the Registers WDTR, WDTS, and WDC. Figure 13.3 shows the CSPR Register and OFS1 Address.

**Table 13.1 Watchdog Timer Specification**

Item	When count source protection mode is disabled	When count source protection mode is enabled
Count Source	CPU clock	125kHz on-chip oscillator clock
Count Operation	Decrement	
Count Start Condition	Either of the followings can be selected. Count automatically starts after reset. Count starts by writing to the WDTS register.	
Count Stop Condition	Stop mode, wait mode, hold state	None
Watchdog Timer Reset Condition	Reset Write 00h, and then FFh to the WDTR register. Underflow	
Operation When the Timer Underflows	Watchdog timer interrupt or watchdog timer reset	Watchdog timer reset
Select Function	Prescaler divide ratio Set the WDC7 bit in the WDC register to select this mode. Count source protection mode Set the CSPROINI bit (flash memory) in the OFS1 address to select whether this mode is enabled or disabled after reset. If this mode is set to disabled after reset, set the CSPRO bit (program) in the CSPR register. Start up or stop watchdog timer after reset Set the WDTON bit in the OFS1 address to select startup or stop.	



**Figure 13.1 Watchdog Timer Block Diagram**

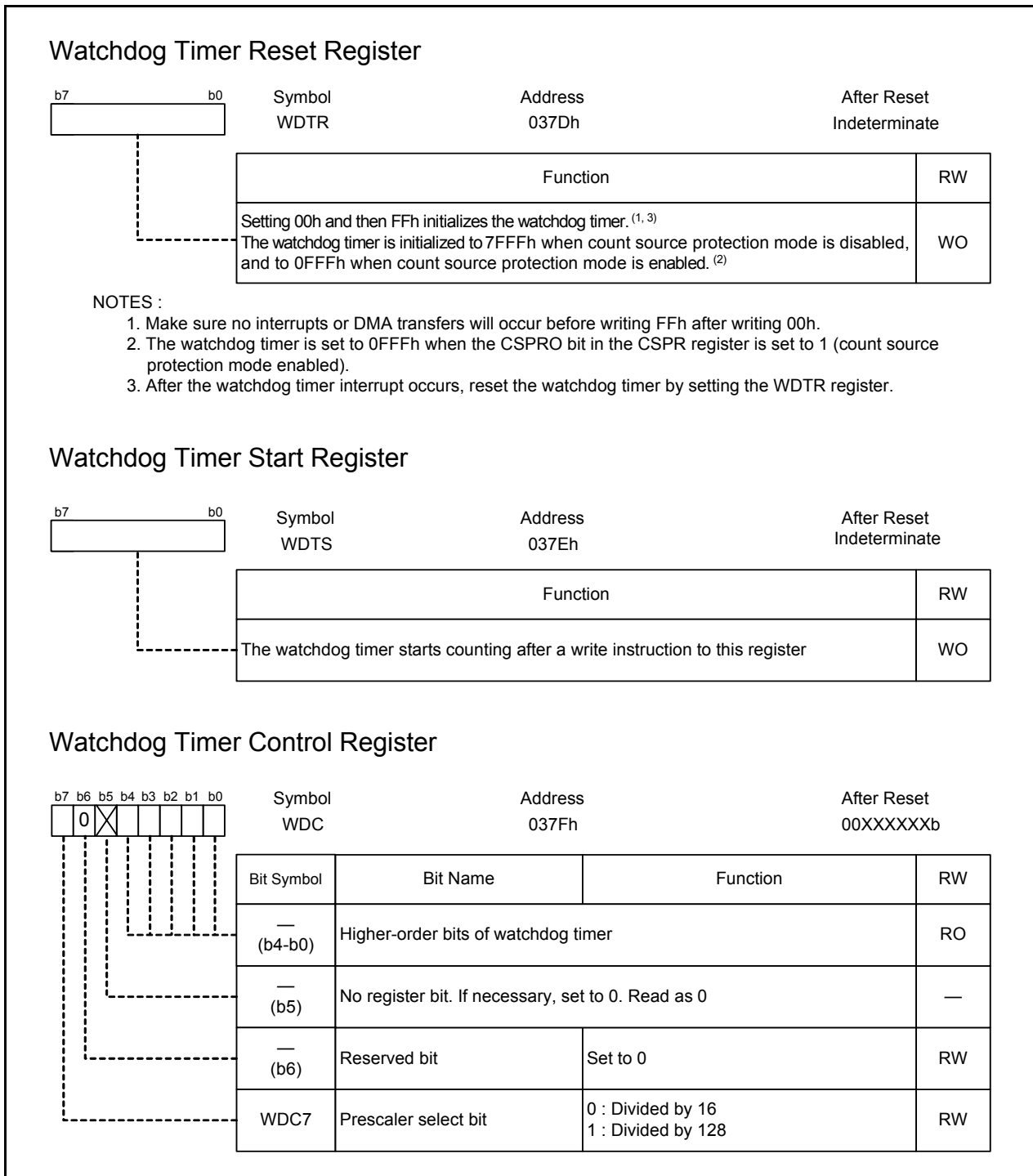
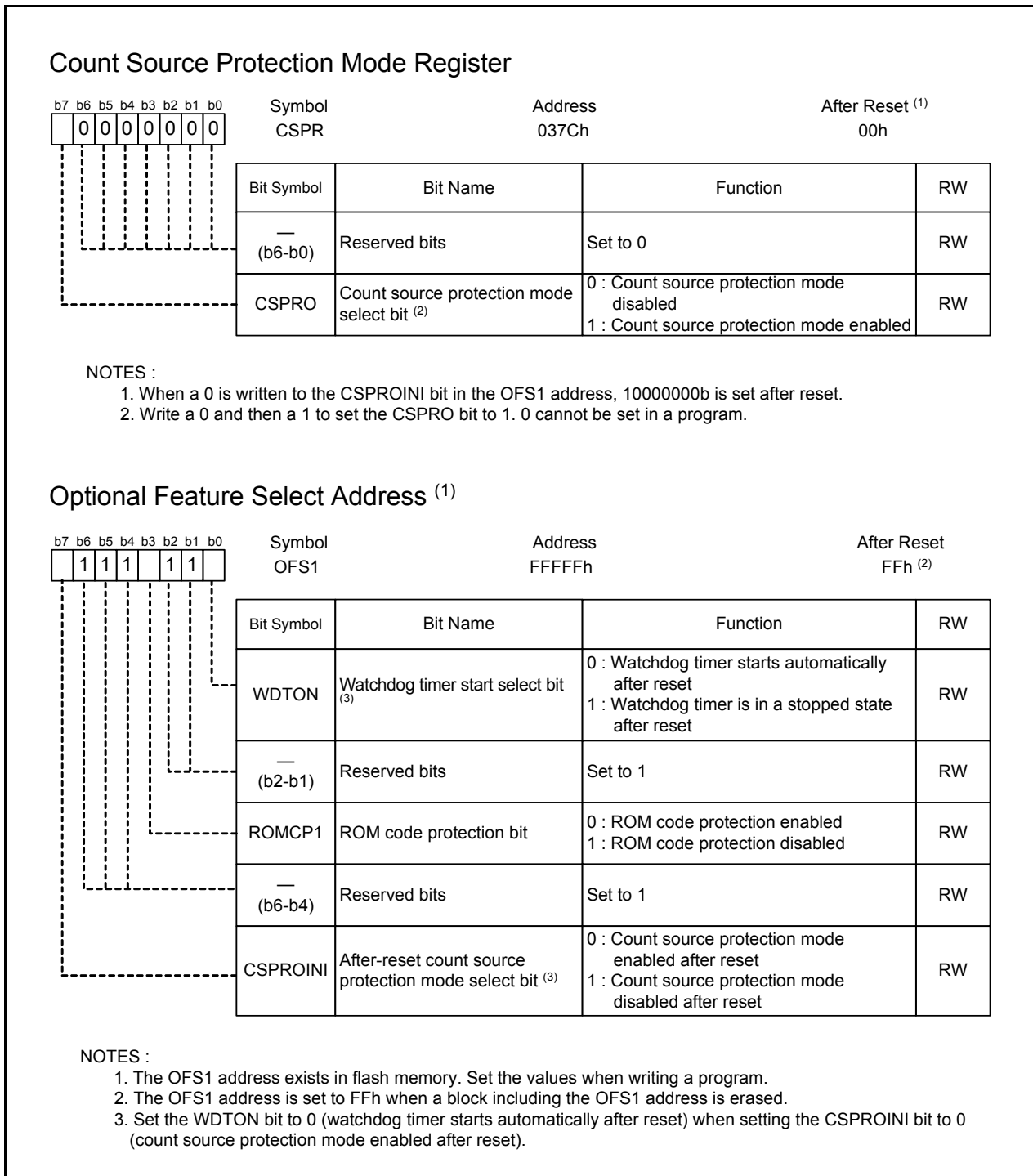


Figure 13.2 Registers WDTR, WDTS, and WDC



**Figure 13.3 CSPR Register and OFS1 Address**

### 13.1 Count Source Protection Mode Disabled

The CPU clock is used for the watchdog timer count source when count source protection mode is disabled.

Table 13.2 lists the Watchdog Timer Specifications (When Count Source Protection Mode is Disabled).

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

Item	Specification
Count Source	CPU Clock
Count Operation	Decrement
Period	$\text{Prescaler divide ratio (n) x watchdog timer count value (32768)}^{(1)}$ CPU clock n: 16 or 128 (selected by the WDC7 bit in the WDC register) example: When CPU clock frequency = 16 MHz and prescaler divided by 16, period = approximately 32.8ms
Watchdog Timer Reset Condition	<ul style="list-style-type: none"> <li>• Reset</li> <li>• Write 00h, and then FFh to the WDTR register.</li> <li>• Underflow</li> </ul>
Count Start Condition	Set the WDTON bit <sup>(2)</sup> in the OFS1 address (FFFFh) to select the watchdog timer operation after reset. <ul style="list-style-type: none"> <li>• When the WDTON bit is set to 1 (watchdog timer is in stop state after reset)                The watchdog timer and prescaler stop after reset and count starts by writing to the WDTS register.</li> <li>• When the WDTON bit is set to 0 (watchdog timer starts automatically after reset)                The watchdog timer and prescaler start counting automatically after reset.</li> </ul>
Count Stop Condition	Stop mode, wait mode, hold state (count resumes from the hold value after exiting.)
Operation when the Timer Underflows	<ul style="list-style-type: none"> <li>• When the PM 12 bit in the PM1 register is set to 0                Watchdog timer interrupt</li> <li>• When the PM 12 bit in the PM1 register is set to 1                Watchdog timer reset (see 5.4 “Watchdog Timer Reset”)</li> </ul>

**NOTES:**

1. Write 00h, and then FFh to the WDTR register to initialize the watchdog timer. The prescaler is initialized after reset. Some errors in the period of the watchdog timer may be caused by the prescaler.
2. The WDTON bit cannot be changed by a program. Write a 0 to bit 0 of address FFFFFh with a flash programmer to set the WDTON bit.

### 13.2 Count Source Protection Mode Enabled

The 125 kHz on-chip oscillator clock is used for the watchdog timer count source when count source protection mode is enabled. If the CPU clock stops when a program is out of control, the clock can still be supplied to the watchdog timer.

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

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

Item	Specification
Count Source	125 kHz on-chip oscillator clock
Count Operation	Decrement
Period	<u>Watchdog timer count value (4096)</u> 125 kHz on-chip oscillator clock  example: When 125 kHz on-chip oscillator clock = 125 kHz, period = approximately 32.8ms
Watchdog Timer Reset Condition	<ul style="list-style-type: none"> <li>• Reset</li> <li>• Write 00h, and then FFh to the WDTR register.</li> <li>• Underflow</li> </ul>
Count Start Condition	Set the WDTON bit <sup>(1)</sup> in the OFS1 address (FFFFh) to select the watchdog timer operation after reset. <ul style="list-style-type: none"> <li>• When the WDTON bit is set to 1 (watchdog timer is in stop state after reset) The watchdog timer and prescaler stop after reset and count starts by writing to the WDTS register.</li> <li>• When the WDTON bit is set to 0 (watchdog timer starts automatically after reset) The watchdog timer and prescaler start counting automatically after reset.</li> </ul>
Count Stop Condition	None (Count does not stop in wait mode or in hold state once count starts. The MCU does not enter stop mode.)
Operation when the Timer Underflows	Watchdog timer reset (see 5.4 “Watchdog Timer Reset” )
Registers, Bits	<ul style="list-style-type: none"> <li>• When the CSPRO bit in the CSPR register is set to 1 (count source protection mode enabled) <sup>(2)</sup>, the followings are set automatically.                             <ul style="list-style-type: none"> <li>-Set 0FFFh to the watchdog timer.</li> <li>-Set the CM14 bit in the CM1 register to 0 (125 kHz on-chip oscillator on).</li> <li>-Set the PM12 bit in the PM1 register to 1 (The watchdog timer reset is generated when watchdog timer underflows.).</li> </ul> </li> <li>• The following conditions apply in count source protection mode.                             <ul style="list-style-type: none"> <li>-Writing to the CM10 bit in the CM1 register is disabled (It remains unchanged even if it is set to 1. The MCU does not enter stop mode.).</li> <li>-Writing to the CM14 bit in the CM1 register is disabled (It remains unchanged even if it is set to 1. The 125 kHz on-chip oscillator does not stop.).</li> </ul> </li> </ul>

NOTES:

1. The WDTON bit cannot be changed by a program. Write 0 to bit 0 of address FFFFh with a flash programmer to set the WDTON bit.
2. Even if 0 is written to the CSPROINI bit in the OFS1 address, the CSPRO is set to 1. The CSPROINI bit cannot be changed by a program. Write 0 to bit 7 of address FFFFh with a flash programmer to set the CSPROINI bit.

## 14. DMAC

The DMAC (Direct Memory Access Controller) allows data to be transferred without the CPU intervention. Four 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 14.1 shows the DMAC Block Diagram. Table 14.1 lists the DMAC Specifications. Figures 14.2 to 14.5 show the DMAC-related registers.

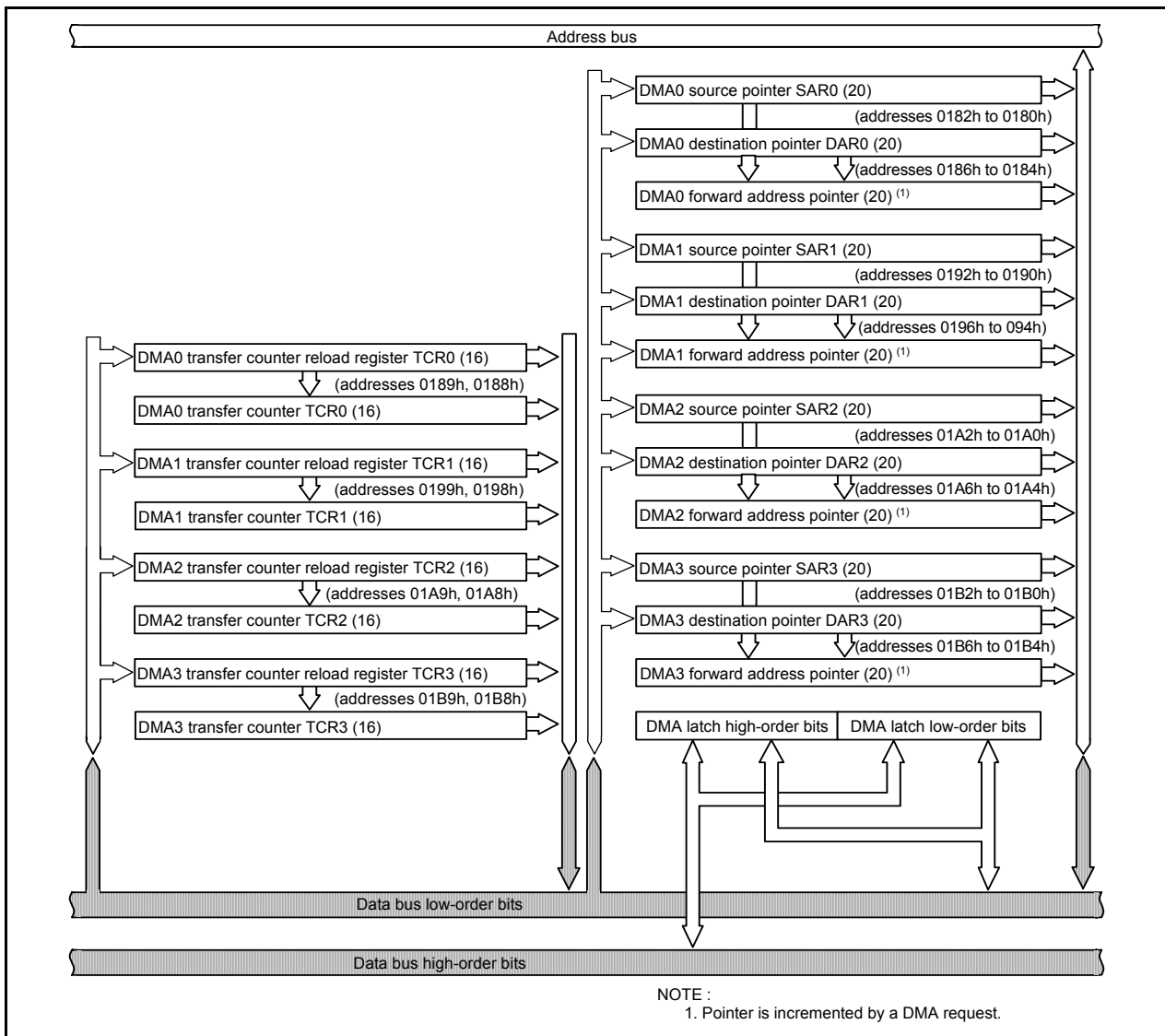


Figure 14.1 DMAC Block Diagram

A DMA request is generated by a write to the DSR bit in the DMiSL register ( $i = 0$  to  $3$ ), as well as by an interrupt request which is generated by any function specified by bits DMS and DSEL4 to DSEL0 in the DMiSL register. 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 IR bit in the interrupt control register does not change state due to a DMA transfer.

A data transfer is initiated each time a DMA request is generated when the DMAE bit in the DMiCON register = 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. Refer to 14.4 “DMA Request” for details.

**Table 14.1 DMAC Specifications (3)**

Item		Specification
No. of Channels		4 (cycle steal method)
Transfer Memory Space		<ul style="list-style-type: none"> <li>• From given address in the 1-Mbyte space to a fixed address</li> <li>• From a fixed address to given address in the 1-Mbyte space</li> <li>• From a fixed address to a fixed address</li> </ul>
Maximum No. of Bytes Transferred		128 Kbytes (with 16-bit transfers) or 64 Kbytes (with 8-bit transfers)
DMA Request Factors (1, 2)		Falling edge of $\overline{INT0}$ to $\overline{INT7}$ Both edges of $\overline{INT0}$ to $\overline{INT7}$ Timer A0 to timer A4 interrupt requests Timer B0 to timer B5 interrupt requests UART0 to 2, UART5 to 7 transmission interrupt requests UART0 to 2, UART5 to 7 reception / ACK interrupt requests SI/O3, SI/O4 interrupt requests A/D conversion interrupt requests Software triggers
Channel Priority		DMA0 > DMA1 > DMA2 > DMA3 (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 DMA <sub>i</sub> transfer counter underflows.
	Repeat Transfer	When the DMA <sub>i</sub> transfer counter underflows, it is reloaded with the value of the DMA <sub>i</sub> transfer counter reload register and a DMA transfer is continued with it.
DMA Interrupt Request Generation Timing		When the DMA <sub>i</sub> transfer counter underflowed
DMA Transfer Start		Data transfer is initiated each time a DMA request is generated when the DMAE bit in the DMA <sub>i</sub> CON register = 1 (enabled).
DMA Transfer Stop	Single Transfer	<ul style="list-style-type: none"> <li>• When the DMAE bit is set to 0 (disabled)</li> <li>• After the DMA<sub>i</sub> transfer counter underflows</li> </ul>
	Repeat Transfer	When the DMAE bit is set to 0 (disabled)
Reload Timing for Forward Address Pointer and DMA <sub>i</sub> Transfer Counter		When a data transfer is started after setting the DMAE bit to 1 (enabled), the forward address pointer is reloaded with the value of the SAR <sub>i</sub> or DAR <sub>i</sub> pointer whichever is specified to be in the forward direction and the DMA <sub>i</sub> transfer counter is reloaded with the value of the DMA <sub>i</sub> transfer counter reload register.
DMA Transfer Cycles		Minimum 3 cycles between SFR and internal RAM

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 factors of DMA requests differ with each channel.
3. Make sure that no DMAC-related registers (addresses 0180h to 01BFh) are accessed by the DMAC.  
i = 0 to 3

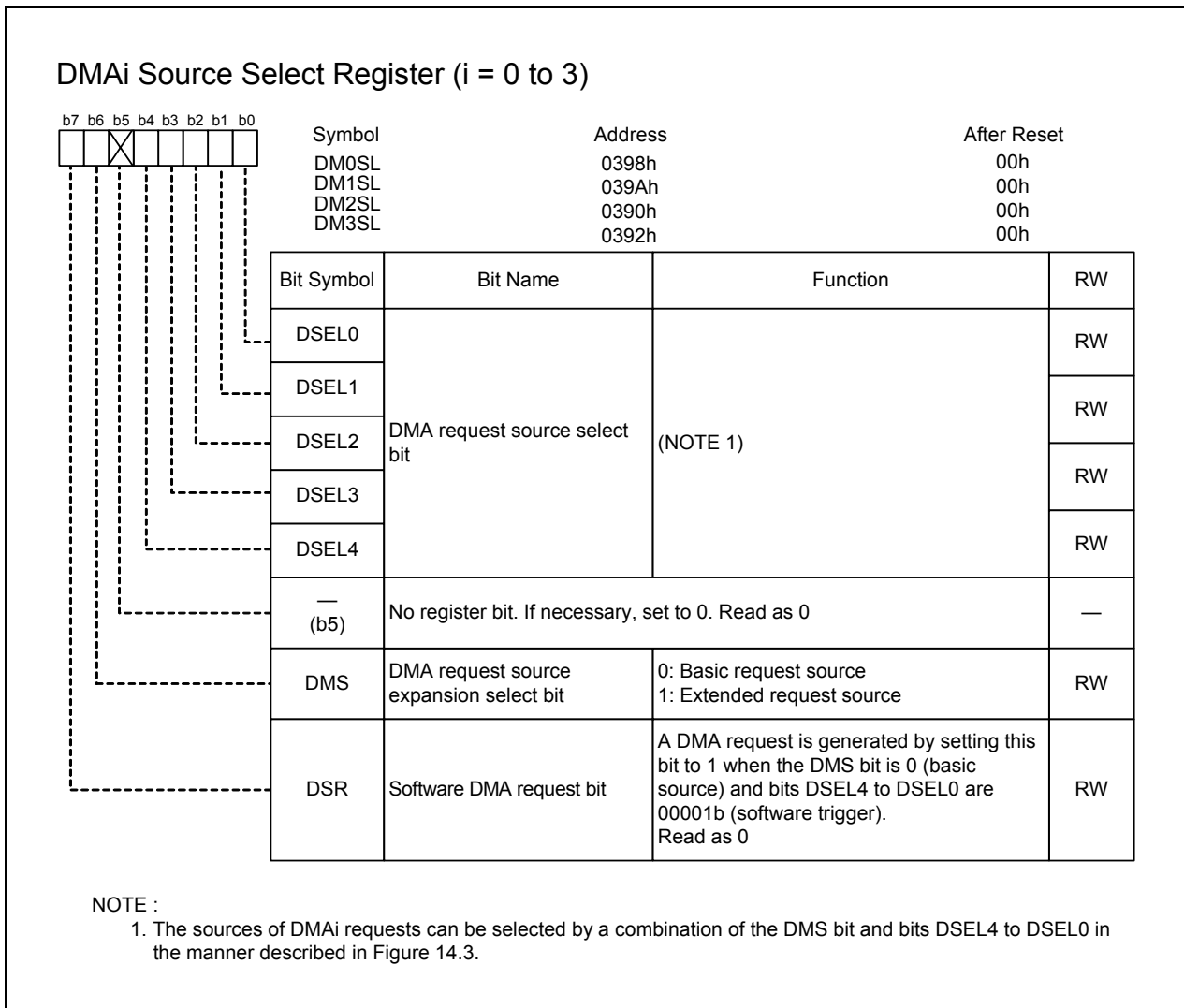


Figure 14.2 Registers DM0SL, DM1SL, DM2SL, and DM3SL (1)

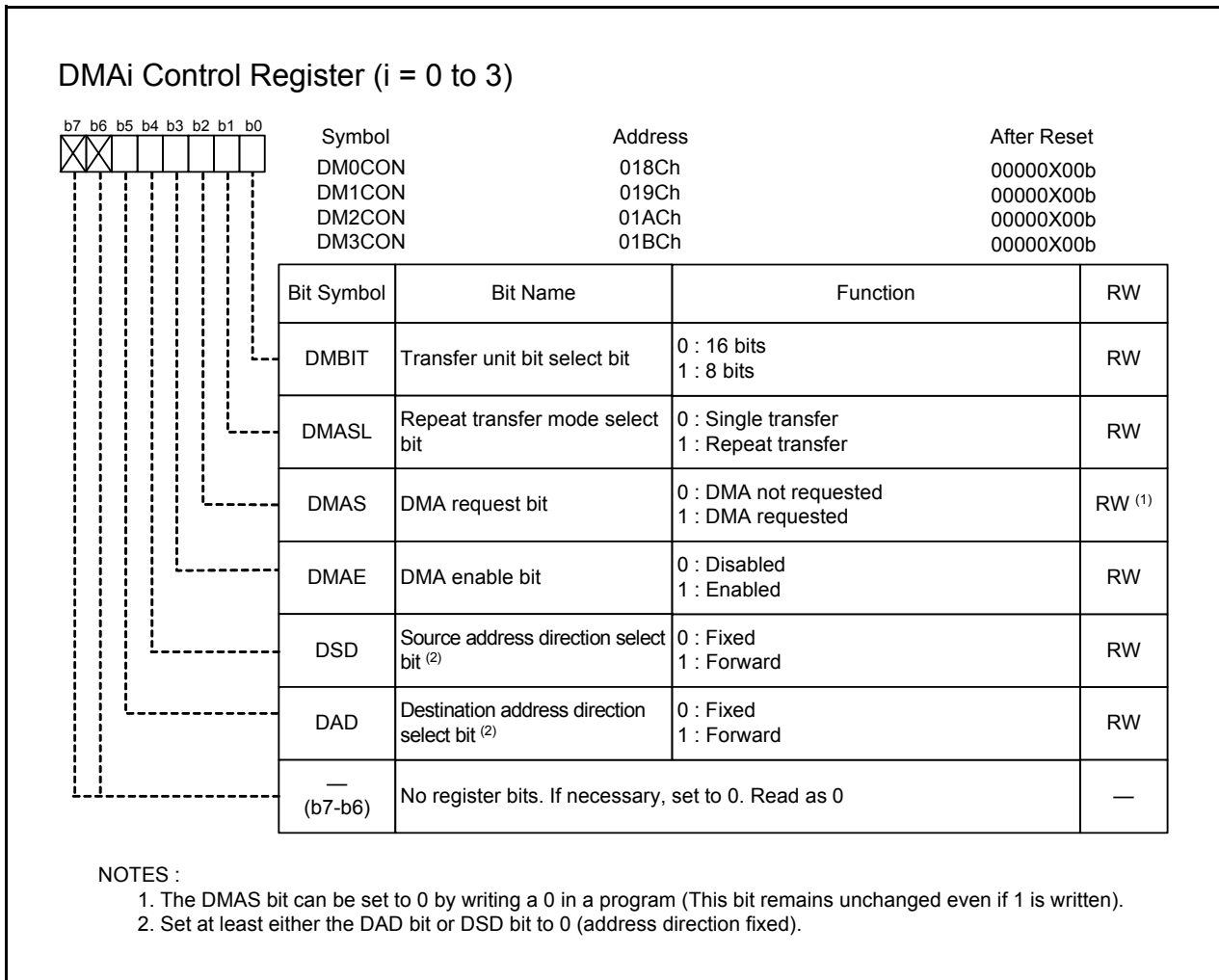
DMA0			DMA2		
DSEL4 to DSEL0	DMS = 0 (Basic Factor of Request)	DMS = 1 (Extended Factor of Request)	DSEL4 to DSEL0	DMS = 0 (Basic Factor of Request)	DMS = 1 (Extended Factor of Request)
0 0 0 0 0 b	Falling edge of INT0 pin	-	0 0 0 0 0 b	Falling edge of INT2 pin	-
0 0 0 0 1 b	Software trigger	-	0 0 0 0 1 b	Software trigger	-
0 0 0 1 0 b	Timer A0	-	0 0 0 1 0 b	Timer A0	-
0 0 0 1 1 b	Timer A1	-	0 0 0 1 1 b	Timer A1	-
0 0 1 0 0 b	Timer A2	-	0 0 1 0 0 b	Timer A2	-
0 0 1 0 1 b	Timer A3	-	0 0 1 0 1 b	Timer A3	-
0 0 1 1 0 b	Timer A4	Both edges of INT0 pin	0 0 1 1 0 b	Timer A4	Both edges of INT2 pin
0 0 1 1 1 b	Timer B0	Timer B3	0 0 1 1 1 b	Timer B0	Timer B3
0 1 0 0 0 b	Timer B1	Timer B4	0 1 0 0 0 b	Timer B1	Timer B4
0 1 0 0 1 b	Timer B2	Timer B5	0 1 0 0 1 b	Timer B2	Timer B5
0 1 0 1 0 b	UART0 transmission	-	0 1 0 1 0 b	UART0 transmission	-
0 1 0 1 1 b	UART0 reception	-	0 1 0 1 1 b	UART0 reception	-
0 1 1 0 0 b	UART2 transmission	-	0 1 1 0 0 b	UART2 transmission	-
0 1 1 0 1 b	UART2 reception	-	0 1 1 0 1 b	UART2 reception	-
0 1 1 1 0 b	A/D conversion	-	0 1 1 1 0 b	A/D conversion	-
0 1 1 1 1 b	UART1 transmission	-	0 1 1 1 1 b	UART1 transmission	-
1 0 0 0 0 b	UART1 reception	Falling edge of INT4 pin	1 0 0 0 0 b	UART1 reception	Falling edge of INT6 pin
1 0 0 0 1 b	UART5 transmission	Both edges of INT4 pin	1 0 0 0 1 b	UART5 transmission	Both edges of INT6 pin
1 0 0 1 0 b	UART5 reception	-	1 0 0 1 0 b	UART5 reception	-
1 0 0 1 1 b	UART6 transmission	-	1 0 0 1 1 b	UART6 transmission	-
1 0 1 0 0 b	UART6 reception	-	1 0 1 0 0 b	UART6 reception	-
1 0 1 0 1 b	UART7 transmission	-	1 0 1 0 1 b	UART7 transmission	-
1 0 1 1 0 b	UART7 reception	-	1 0 1 1 0 b	UART7 reception	-
1 0 1 1 1 b	-	-	1 0 1 1 1 b	-	-
1 1 X X X b	-	-	1 1 X X X b	-	-

X indicates 0 or 1. - indicates no setting.

DMA1			DMA3		
DSEL4 to DSEL0	DMS = 0 (Basic Factor of Request)	DMS = 1 (Extended Factor of Request)	DSEL4 to DSEL0	DMS = 0 (Basic Factor of Request)	DMS = 1 (Extended Factor of Request)
0 0 0 0 0 b	Falling edge of INT1 pin	-	0 0 0 0 0 b	Falling edge of INT3 pin	-
0 0 0 0 1 b	Software trigger	-	0 0 0 0 1 b	Software trigger	-
0 0 0 1 0 b	Timer A0	-	0 0 0 1 0 b	Timer A0	-
0 0 0 1 1 b	Timer A1	-	0 0 0 1 1 b	Timer A1	-
0 0 1 0 0 b	Timer A2	-	0 0 1 0 0 b	Timer A2	-
0 0 1 0 1 b	Timer A3	SI / O3	0 0 1 0 1 b	Timer A3	SI / O3
0 0 1 1 0 b	Timer A4	SI / O4	0 0 1 1 0 b	Timer A4	SI / O4
0 0 1 1 1 b	Timer B0	Both edges of INT1 pin	0 0 1 1 1 b	Timer B0	Both edges of INT3 pin
0 1 0 0 0 b	Timer B1	-	0 1 0 0 0 b	Timer B1	-
0 1 0 0 1 b	Timer B2	-	0 1 0 0 1 b	Timer B2	-
0 1 0 1 0 b	UART0 transmission	-	0 1 0 1 0 b	UART0 transmission	-
0 1 0 1 1 b	UART0 reception / ACK0	-	0 1 0 1 1 b	UART0 reception / ACK0	-
0 1 1 0 0 b	UART2 transmission	-	0 1 1 0 0 b	UART2 transmission	-
0 1 1 0 1 b	UART2 reception / ACK2	-	0 1 1 0 1 b	UART2 reception / ACK2	-
0 1 1 1 0 b	A/D conversion	-	0 1 1 1 0 b	A/D conversion	-
0 1 1 1 1 b	UART1 reception / ACK1	-	0 1 1 1 1 b	UART1 reception / ACK1	-
1 0 0 0 0 b	UART1 transmission	Falling edge of INT5 pin	1 0 0 0 0 b	UART1 transmission	Falling edge of INT7 pin
1 0 0 0 1 b	UART5 transmission	Both edges of INT5 pin	1 0 0 0 1 b	UART5 transmission	Both edges of INT7 pin
1 0 0 1 0 b	UART5 reception / ACK5	-	1 0 0 1 0 b	UART5 reception / ACK5	-
1 0 0 1 1 b	UART6 transmission	-	1 0 0 1 1 b	UART6 transmission	-
1 0 1 0 0 b	UART6 reception / ACK6	-	1 0 1 0 0 b	UART6 reception / ACK6	-
1 0 1 0 1 b	UART7 transmission	-	1 0 1 0 1 b	UART7 transmission	-
1 0 1 1 0 b	UART7 reception / ACK7	-	1 0 1 1 0 b	UART7 reception / ACK7	-
1 0 1 1 1 b	-	-	1 0 1 1 1 b	-	-
1 1 X X X b	-	-	1 1 X X X b	-	-

X indicates 0 or 1. - indicates no setting.

Figure 14.3 Registers DM0SL, DM1SL, DM2SL, and DM3SL (2)



**Figure 14.4 Registers DM0CON, DM1CON, DM2CON, and DM3CON**

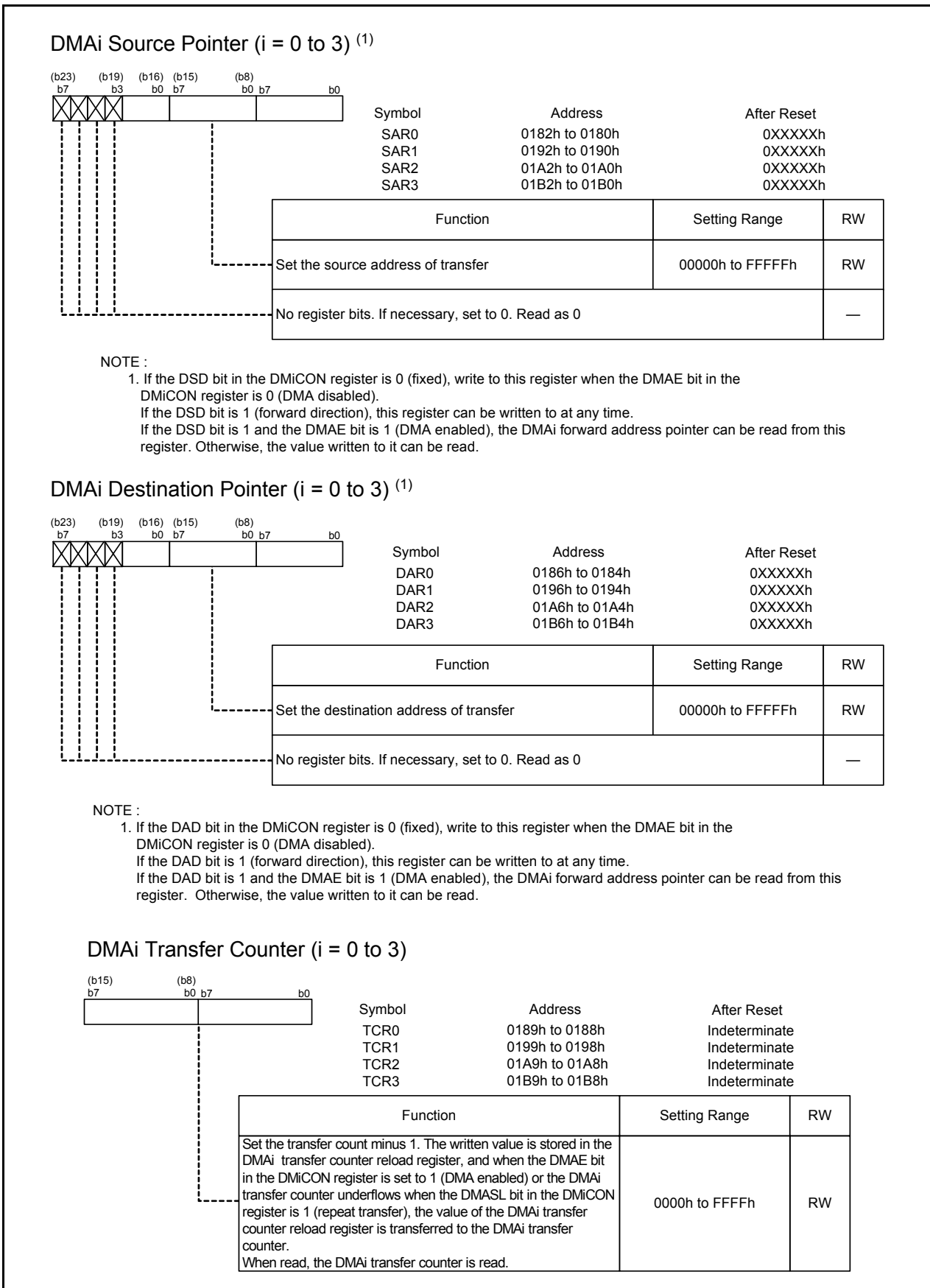


Figure 14.5 Registers SAR0, SAR1, SAR2, SAR3, DAR0, DAR1, DAR2, DAR3, TCR0, TCR1, TCR2, and TCR3

## 14.1 Transfer Cycles

Transfer cycle is composed of a bus cycle to read data from source address (source read) and a bus cycle to write data to destination address (destination write). The number of read and write bus cycles depends on source and destination addresses. During memory extension and microprocessor modes, it is also affected by the BYTE pin level. Furthermore, the bus cycle itself is extended by a software wait or  $\overline{\text{RDY}}$  signal.

### 14.1.1 Effect of Source and Destination Addresses

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

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

### 14.1.2 Effect of BYTE Pin Level

During memory extension and microprocessor modes, if 16 bits of data are to be transferred on an 8-bit data bus (input on the BYTE pin = high), the operation is accomplished by transferring 8 bits of data twice. Therefore, this operation requires two bus cycles to read data and two bus cycles to write data. Furthermore, if the DMAC is to access the internal area (internal ROM, internal RAM, or SFR), unlike in the case of the CPU, the DMAC does it through the data bus width selected by the BYTE pin.

### 14.1.3 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.

### 14.1.4 Effect of $\overline{\text{RDY}}$ Signal

During memory extension and microprocessor modes, DMA transfers to and from an external area are affected by the  $\overline{\text{RDY}}$  signal. Refer to 8.2.6 “ $\overline{\text{RDY}}$  Signal”.

Figure 14.6 shows the example of the Transfer Cycles for 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 using an 8-bit bus ((2) on Figure 14.6), two source read bus cycles and two destination write bus cycles are required.

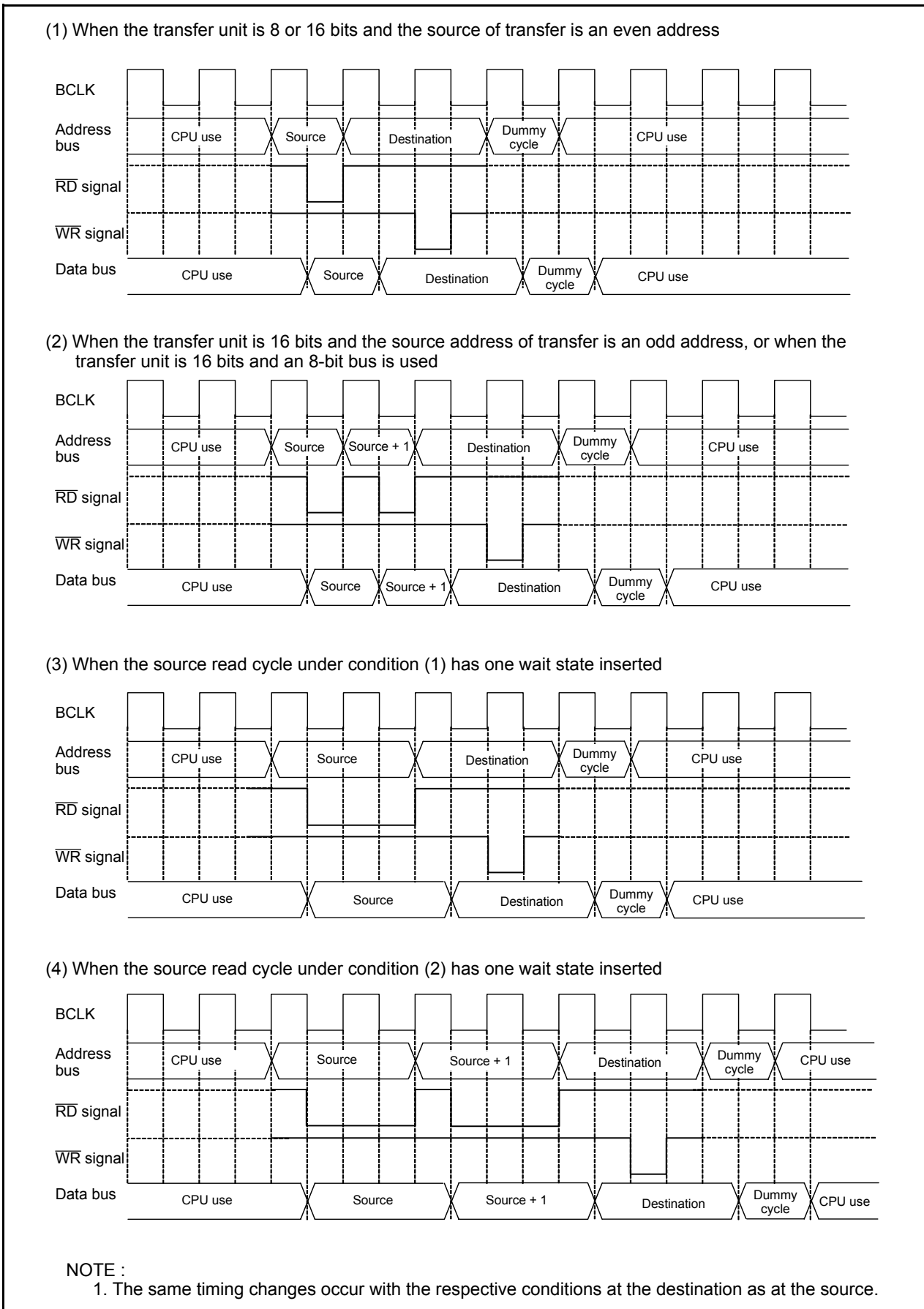


Figure 14.6 Transfer Cycles for Source Read

## 14.2 DMA Transfer Cycles

The number of DMA transfer cycles can be calculated as follows.

Table 14.2 lists the DMAC Transfer Cycles. Table 14.3 lists the Coefficients j and k.

Number of transfer cycles per transfer unit = Number of read cycles × j + Number of write cycles × k

**Table 14.2 DMAC Transfer Cycles**

Transfer Unit	Bus Width	Access Address	Single-Chip Mode		Memory Expansion Mode Microprocessor Mode	
			No. of Read Cycles	No. of Write Cycles	No. of Read Cycles	No. of Write Cycles
8-bit Transfers (DMBIT= 1)	16-bit (BYTE = "L")	Even	1	1	1	1
		Odd	1	1	1	1
	8-bit (BYTE = "H")	Even	-	-	1	1
		Odd	-	-	1	1
16-bit Transfers (DMBIT= 0)	16-bit (BYTE = "L")	Even	1	1	1	1
		Odd	2	2	2	2
	8-bit (BYTE = "H")	Even	-	-	2	2
		Odd	-	-	2	2

- indicates that no condition exists.

**Table 14.3 Coefficients j and k**

	Internal Area				External Area						
	Internal ROM, RAM		SFR		Separate Bus			Multiplex Bus			
	No Wait	With Wait	1-Wait (2)	2-Wait (2)	No Wait	With Wait (1)			With Wait (1)		
						1-Wait	2-Wait	3-Wait	1-Wait	2-Wait	3-Wait
j	1	2	2	3	1	2	3	4	3	3	4
k	1	2	2	3	2	2	3	4	3	3	4

NOTES:

1. It depends on the set value of the CSE register.
2. It depends on the set value of the PM20 bit in the PM2 register.



### 14.3 DMA Enabled

When a data transfer starts after setting the DMAE bit in the DMiCON register ( $i = 0$  to  $3$ ) to 1 (enabled), the DMAC operates as follows:

- (1) Reload the forward address pointer with the SAR $i$  register value when the DSD bit in the DMiCON register is 1 (forward) or the DAR $i$  register value when the DAD bit in the DMiCON register is 1 (forward).
- (2) Reload the DMA $i$  transfer counter with the DMA $i$  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 the DMiCON register simultaneously.

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

If the DMA $i$  is not in an initial state, the above steps should be repeated.

### 14.4 DMA Request

The DMAC can generate a DMA request as triggered by the factor of request that is selected with the DMS bit and bits DSEL4 to DSEL0 in the DMiSL register ( $i = 0$  to  $3$ ) on either channel. Table 14.1 lists 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 is set to 1 (enabled) when this occurs, 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 bit or bits DSEL4 to DSEL0 change state. Therefore, always be sure to set the DMAS bit to 0 after changing the DMS bit or bits DSEL4 to DSEL0.

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 14.4 Timing at Which the DMAS Bit Changes State**

DMA Factor	DMAS Bit in 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 in the DMiSL register is set to 1	• Immediately before a data transfer starts
Peripheral Function	When the interrupt control register for the peripheral function that is selected by bits DSEL4 to DSEL0 and DMS in the DMiSL register has its IR bit set to 1	• When set by writing a 0 in a program

$i = 0$  to  $3$

### 14.5 Channel Priority and DMA Transfer Timing

If several channels of DMA0 to DMA3 are enabled and DMA transfer request signals are detected active in the same sampling period (one period from a falling edge to the next falling edge of BCLK), 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 > DMA2 > DMA3. The following describes DMAC operation when DMA0 and DMA1 requests are detected active in the same sampling period. Figure 14.7 shows an example of DMA Transfer by External Factors.

In Figure 14.7, DMA0 request having priority is received first to start a transfer when DMA0 and DMA1 requests are generated simultaneously. After one DMA0 transfer is completed, a bus access privilege 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 access privilege is again returned to the CPU.

In addition, DMA requests cannot be incremented since each channel has one DMAS bit. Therefore, when DMA requests, as DMA1 in Figure 14.7, occurs more than one time, the DMAS bit is set to 0 after getting the bus access privilege. The bus access privilege is returned to the CPU when one transfer is completed.

Refer to 8.2.7 "HOLD Signal" for details about bus access privilege.

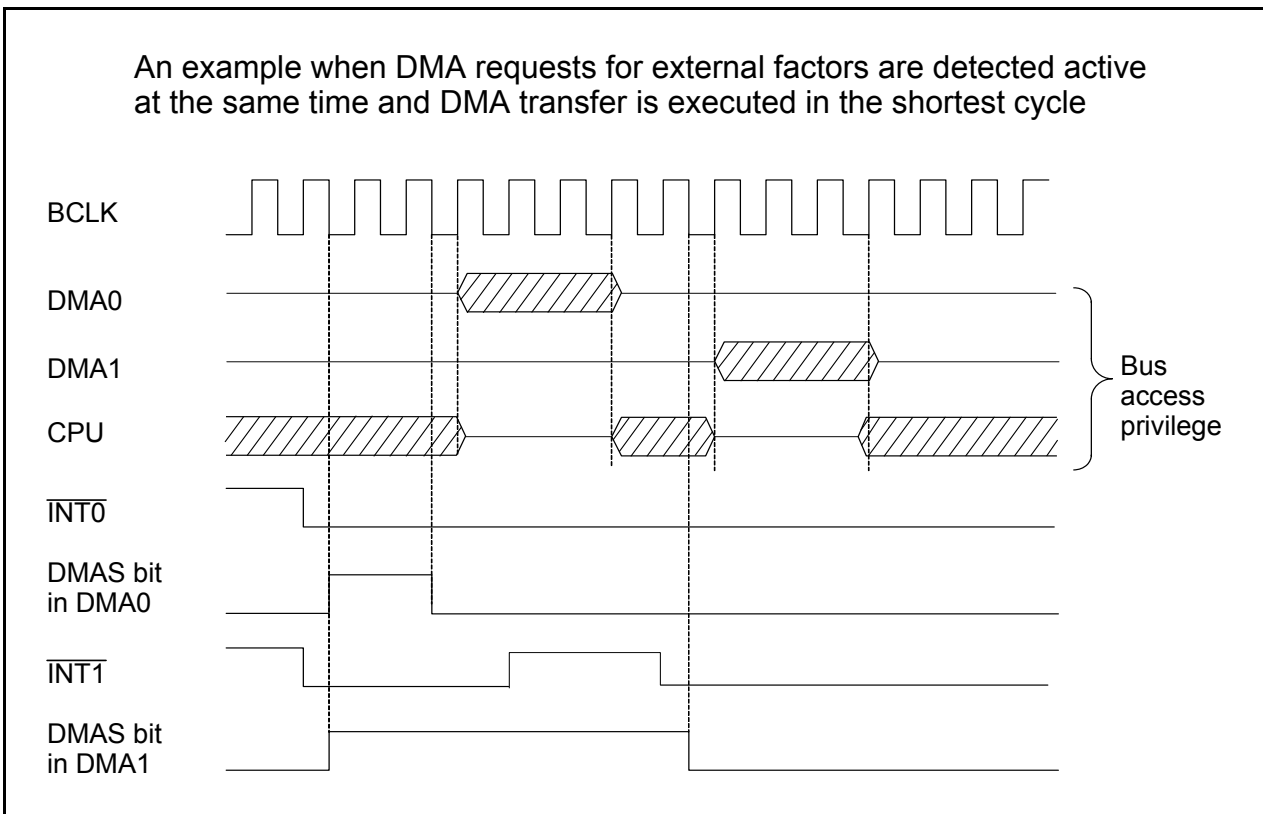


Figure 14.7 DMA Transfer by External Factors

## 15. Timers

Eleven 16-bit timers, each capable of operating independently of the others, can be classified by function as either Timer A (five) and Timer B (six). The count source for each timer acts as a clock, to control such timer operations as counting, reloading, etc. Figure 15.1 shows Timers A and B count source, and Figures 15.2 and 15.3 show block diagrams of Timer A and Timer B configuration, respectively.

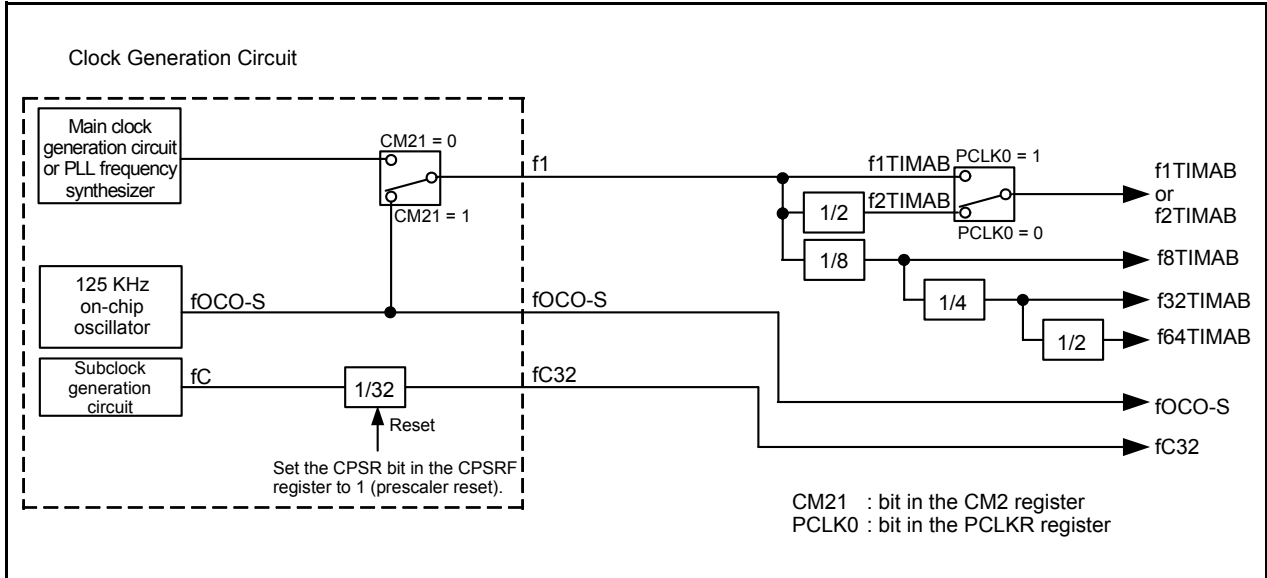


Figure 15.1 Timers A and B Count Source

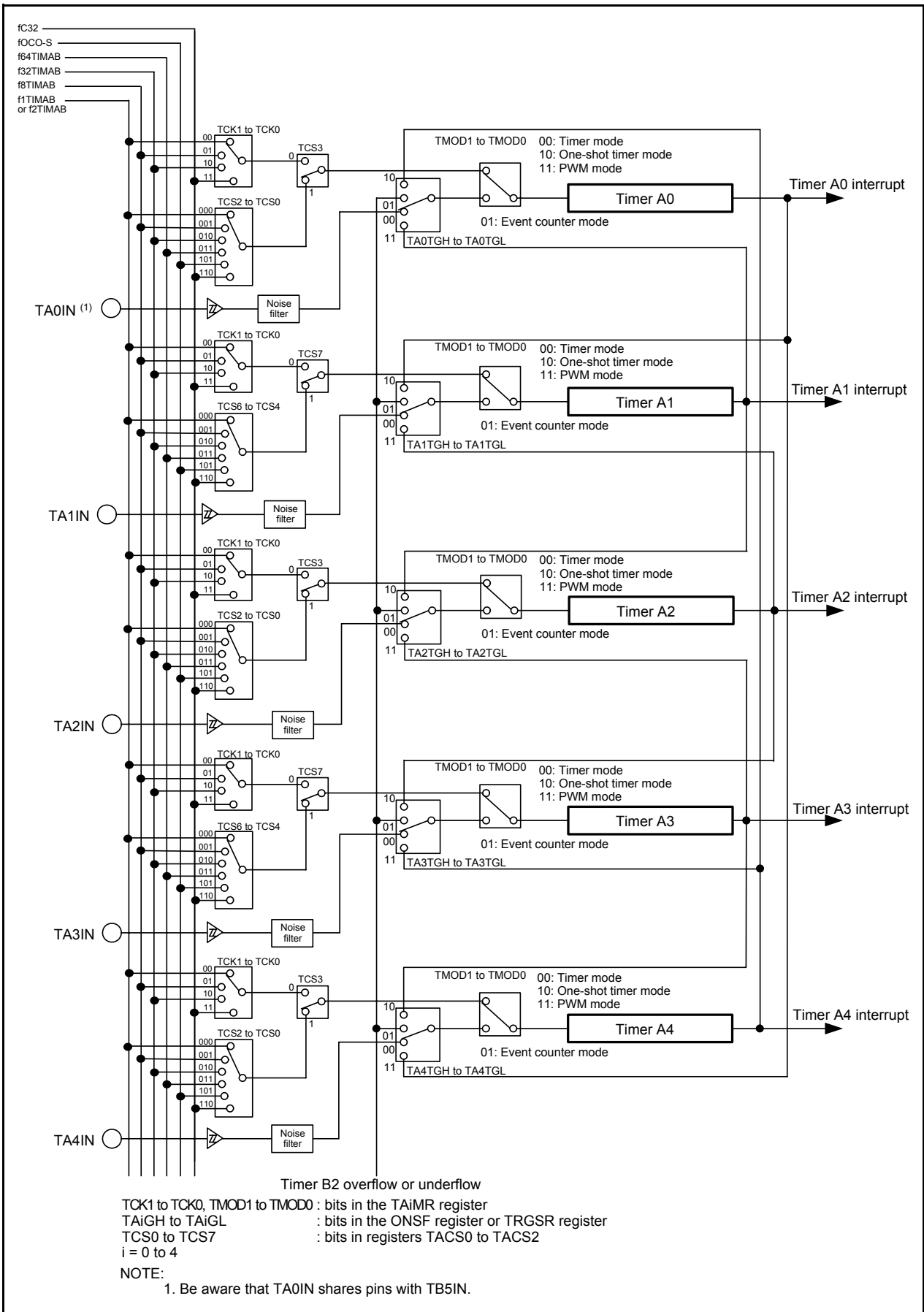


Figure 15.2 Timer A Configuration

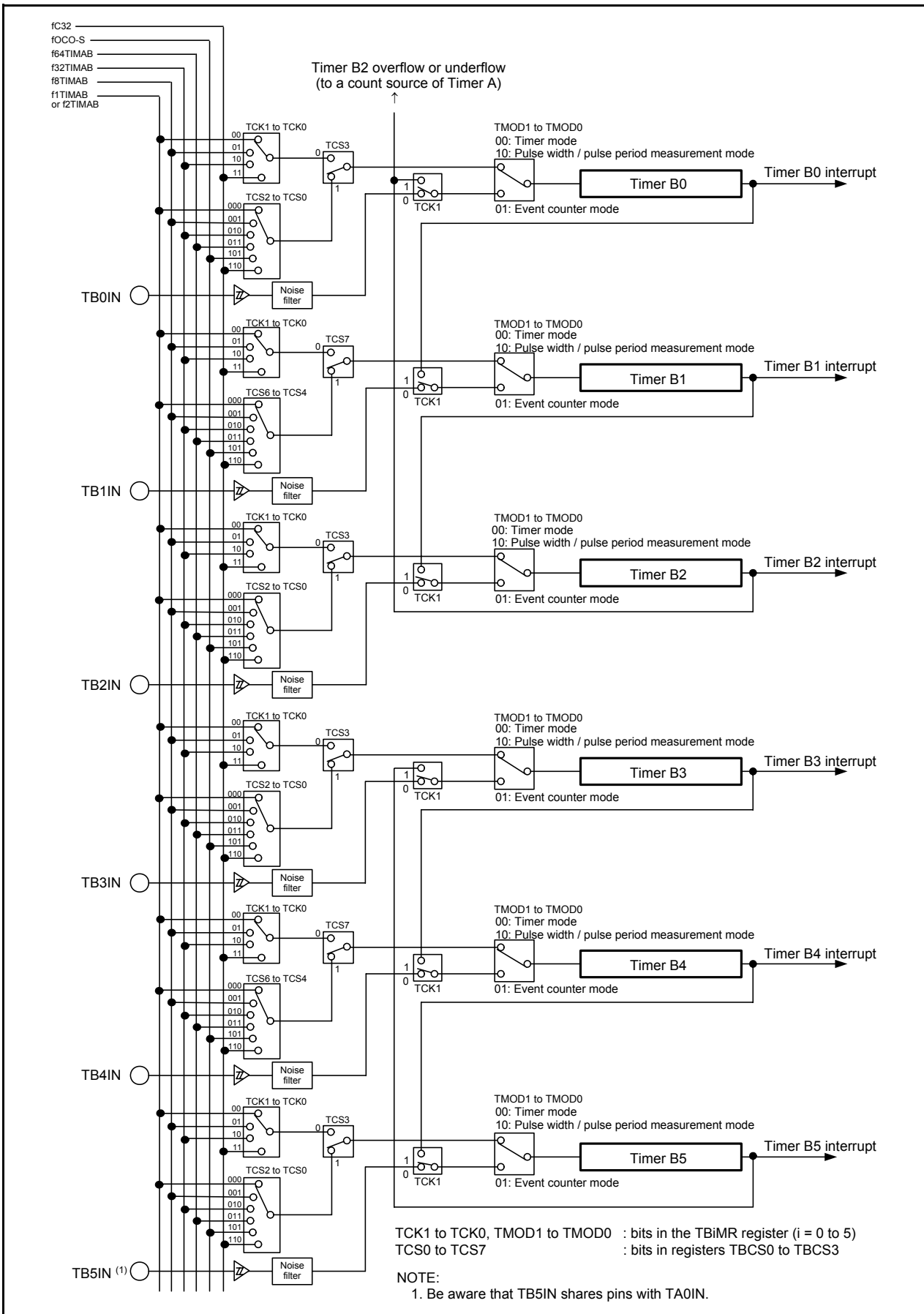


Figure 15.3 Timer B Configuration

### 15.1 Timer A

Figure 15.4 shows a Timer A Block Diagram. Figures 15.5 to 15.9 show registers related to Timer A. Timer A supports the following four modes. Except in event counter mode, Timers A0 to A4 all have the same function. Use bits TMOD1 to TMOD0 in the TAI<sub>i</sub>MR 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 0000h.
- Pulse Width Modulation (PWM) Mode        The timer outputs pulses in a given width successively.

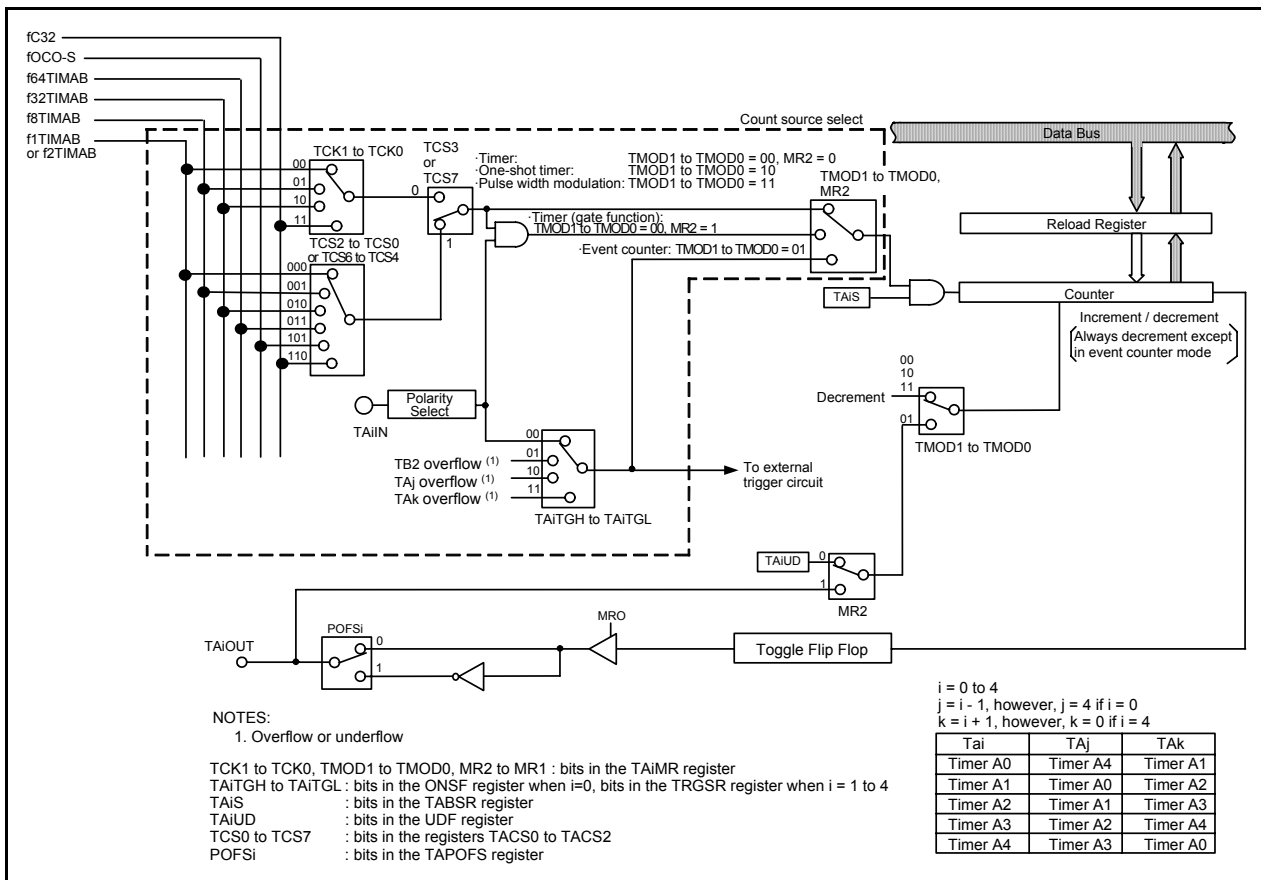


Figure 15.4 Timer A Block Diagram

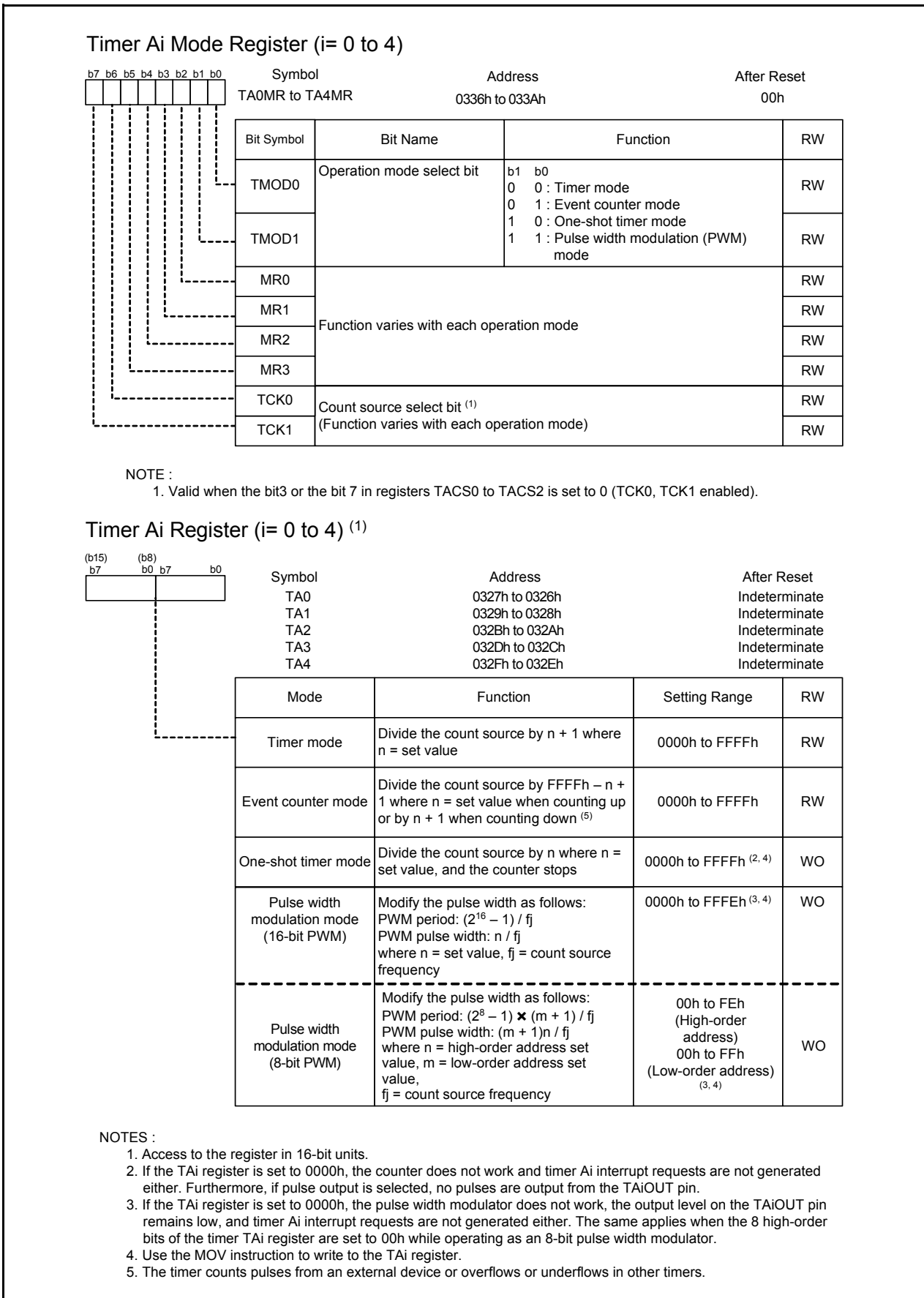


Figure 15.5 Registers TA0MR to TA4MR and TA0 to TA4

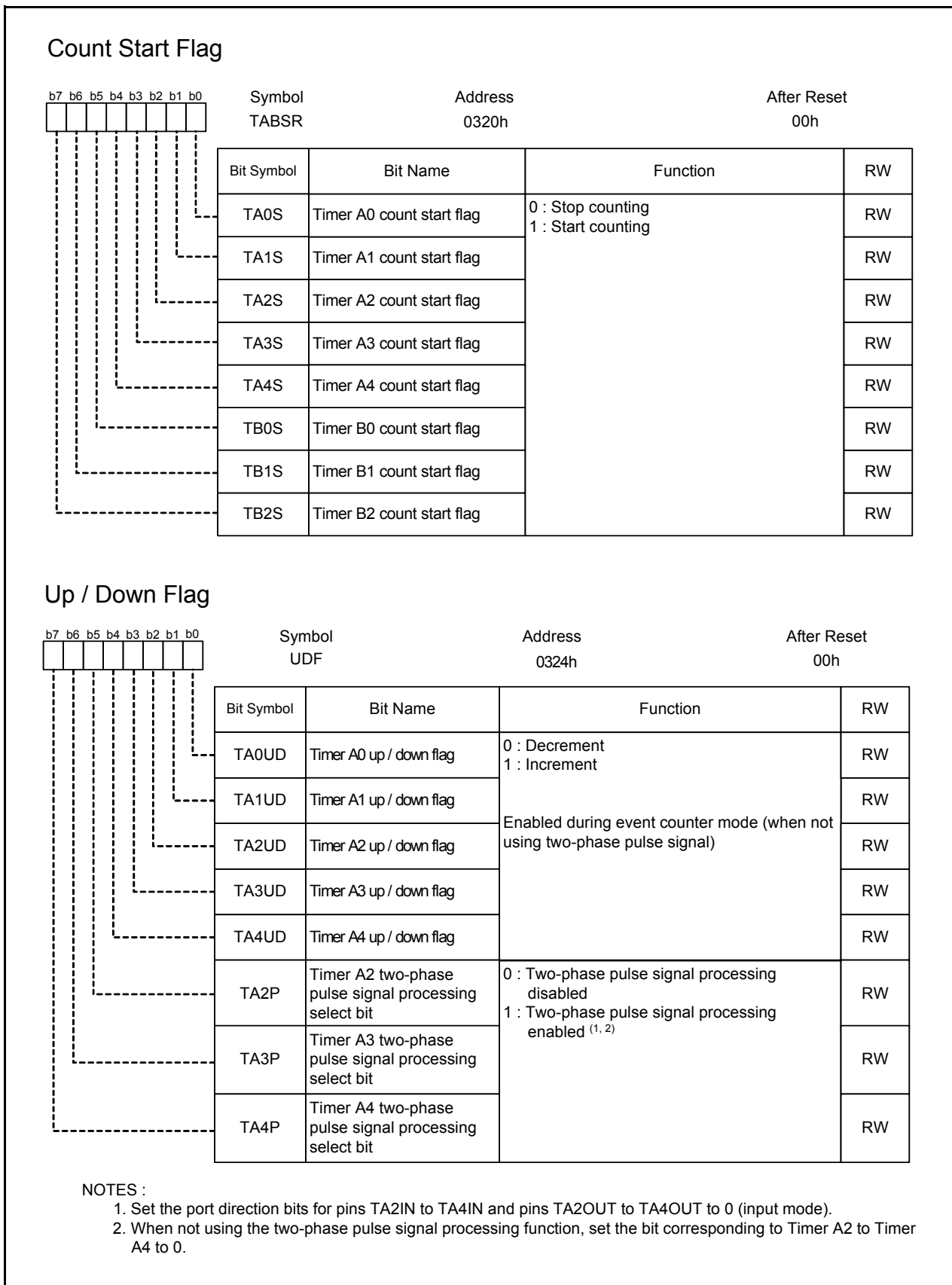


Figure 15.6 Registers TABSR and UDF



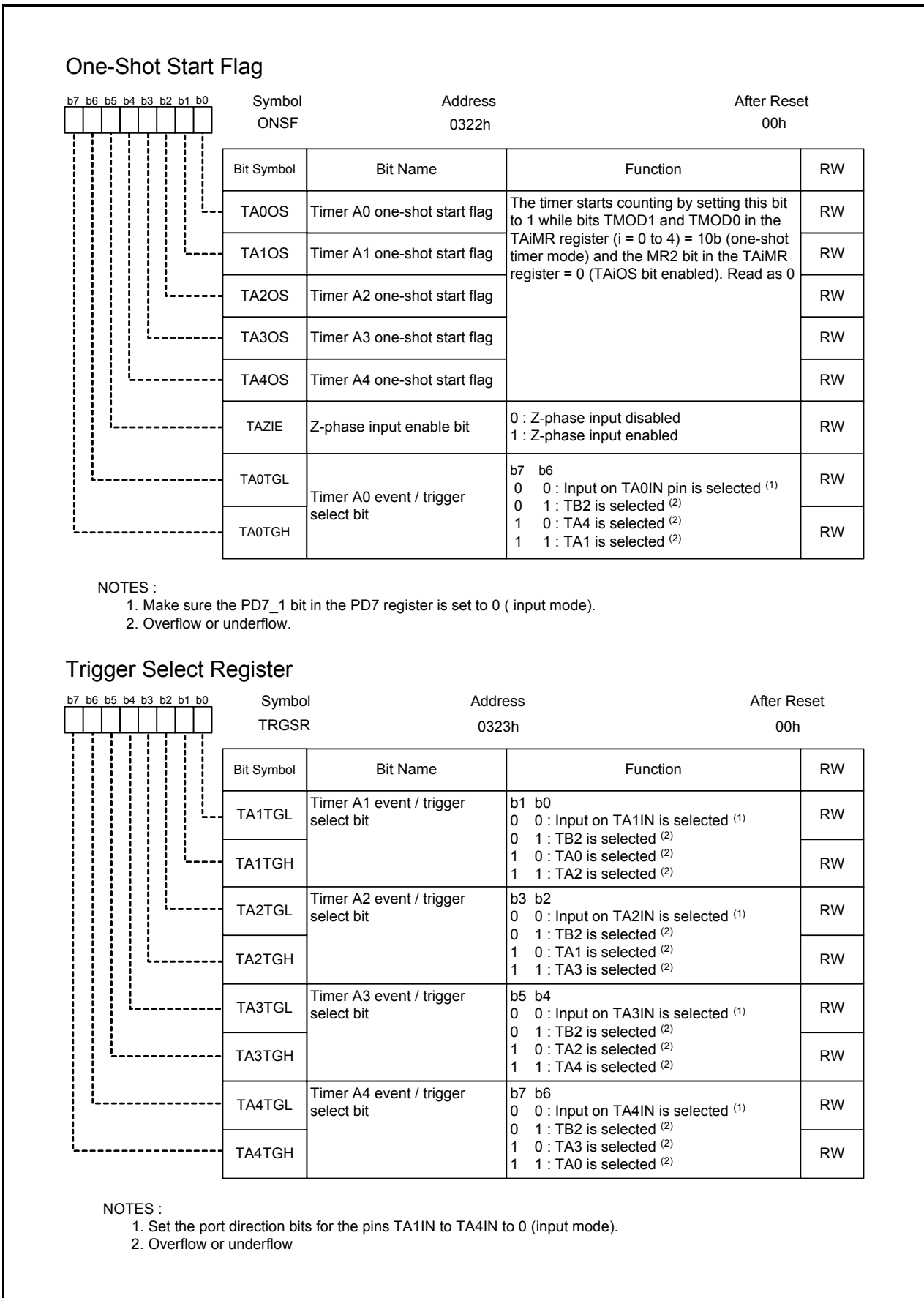


Figure 15.7 Registers ONSF and TRGSR

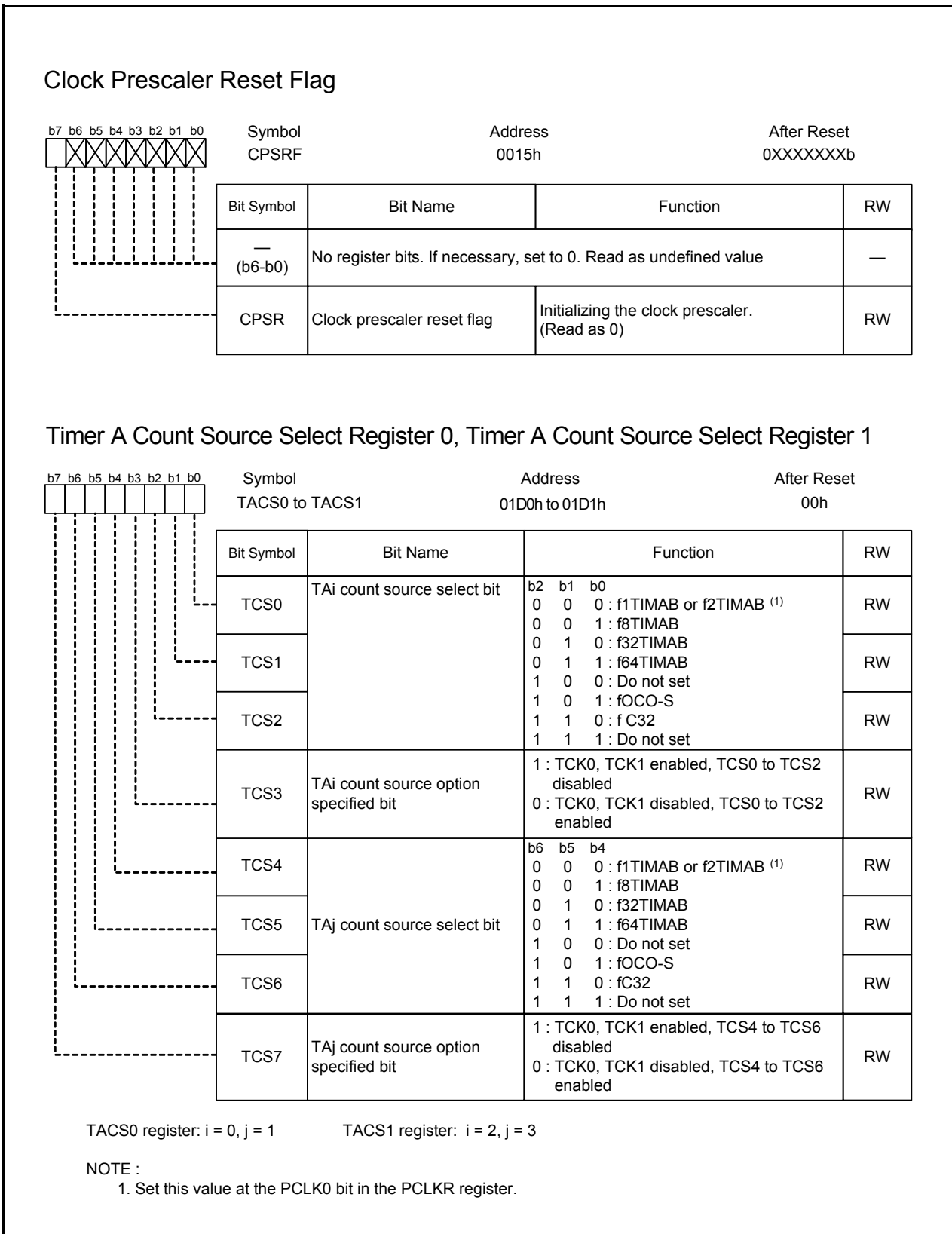
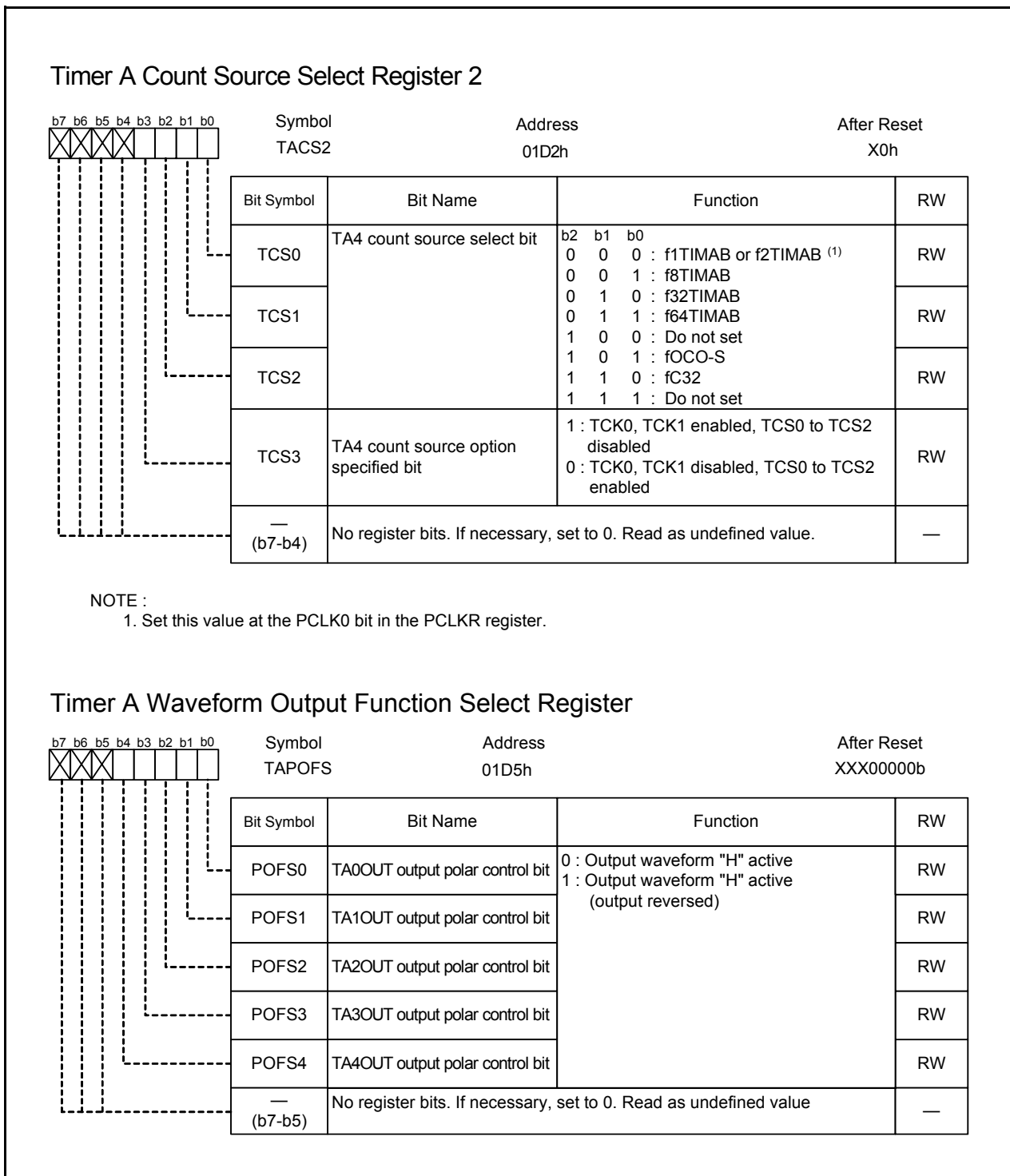


Figure 15.8 Registers CPSRF, TACS0, and TACS1



**Figure 15.9 Registers TACS2 and TAPOFS**

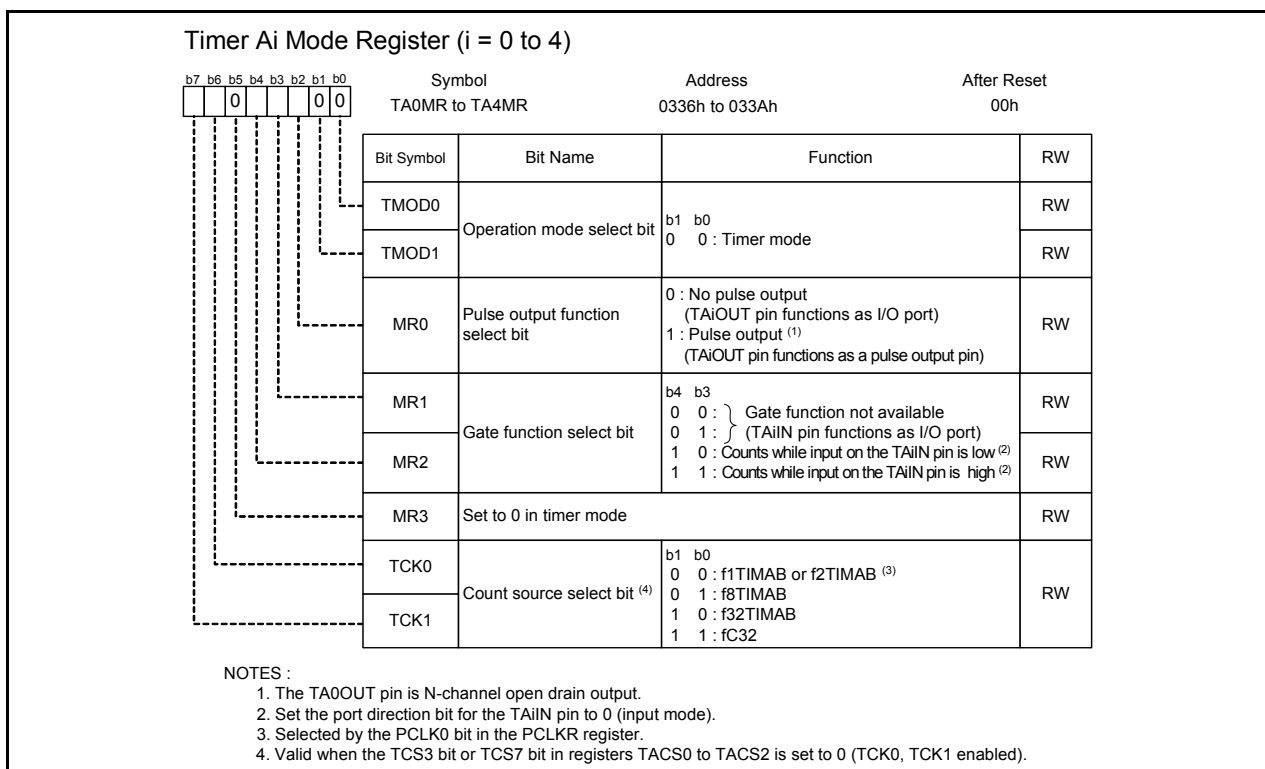
### 15.1.1 Timer Mode

In timer mode, the timer counts a count source generated internally (see **Table 15.1**). Figure 15.10 shows TAIiMR Register in Timer Mode.

**Table 15.1 Specifications in Timer Mode**

Item	Specification
Count Source	f1TIMAB, f2TIMAB, f8TIMAB, f32TIMAB, f64TIMAB, fOCO-S, fC32
Count Operation	<ul style="list-style-type: none"> <li>Decrement</li> <li>When the timer underflows, it reloads the reload register contents and continues counting</li> </ul>
Divide Ratio	1 / (n+1)      n: set value of TAI register 0000h to FFFFh
Count Start Condition	Set the TAI <i>S</i> bit in the TABSR register to 1 (start counting)
Count Stop Condition	Set the TAI <i>S</i> 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 the TAI register
Write to Timer	<ul style="list-style-type: none"> <li>When not counting Value written to the TAI register is written to both reload register and counter</li> <li>When counting Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>
Select Function	<ul style="list-style-type: none"> <li>Gate function Counting can be started and stopped by an input signal to the TAIiN pin</li> <li>Pulse output function Whenever the timer underflows, the output polarity of TAIiOUT pin is inverted. When the TAI<i>S</i> bit is set to 0 (stop counting), the pin outputs "L."</li> <li>Output polarity control While the output polarity of the TAIiOUT pin is inverted (the TAI<i>S</i> bit is set to 0 (stop counting)), the pin outputs "H."</li> </ul>

i = 0 to 4



**Figure 15.10 TAIiMR Register in Timer Mode**

### 15.1.2 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 15.2 lists Specifications in Event Counter Mode (When Not Processing Two-Phase Pulse Signal). Figure 15.11 shows the TAI<sub>MR</sub> Register in Event Counter Mode (when not using two-phase pulse signal processing).

**Table 15.2 Specifications in Event Counter Mode (When Not Processing Two-Phase Pulse Signal)**

Item	Specification
Count Source	<ul style="list-style-type: none"> <li>External signals input to the TAI<sub>IN</sub> pin (effective edge can be selected in a program)</li> <li>Timer B2 overflows or underflows,</li> <li>Timer A<sub>j</sub> (<math>j = i - 1</math>, except <math>j = 4</math> if <math>i = 0</math>) overflows or underflows,</li> <li>Timer A<sub>k</sub> (<math>k = i + 1</math>, except <math>k=0</math> if <math>i = 4</math>) overflows or underflows</li> </ul>
Count Operation	<ul style="list-style-type: none"> <li>Increment or decrement can be selected by program.</li> <li>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.</li> </ul>
Divide Ratio	<ul style="list-style-type: none"> <li><math>1 / (FFFFh - n + 1)</math> for increment</li> <li><math>1 / (n + 1)</math> for decrement</li> <li>n: set value of the TAI register 0000h to FFFFh</li> </ul>
Count Start Condition	Set the TAI <sub>S</sub> bit in the TABSR register to 1 (start counting)
Count Stop Condition	Set the TAI <sub>S</sub> bit to 0 (stop counting)
Interrupt Request Generation Timing	Timer overflow or underflow
TAI <sub>IN</sub> Pin Function	I/O port or count source input
TAI <sub>OUT</sub> Pin Function	I/O port, pulse output
Read from Timer	Count value can be read by reading the TAI register
Write to Timer	<ul style="list-style-type: none"> <li>When not counting Value written to the TAI register is written to both reload register and counter</li> <li>When counting Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>
Select Function	<ul style="list-style-type: none"> <li>Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it</li> <li>Pulse output function Whenever the timer underflows or underflows, the output polarity of the TAI<sub>OUT</sub> pin is inverted. When the TAI<sub>S</sub> bit is set to 0 (stop counting), the pin outputs low.</li> <li>Output polarity control While the output polarity of the TAI<sub>OUT</sub> pin is inverted (the TAI<sub>S</sub> bit is set to 0 (stop counting)), the pin outputs high.</li> </ul>

$i = 0$  to 4

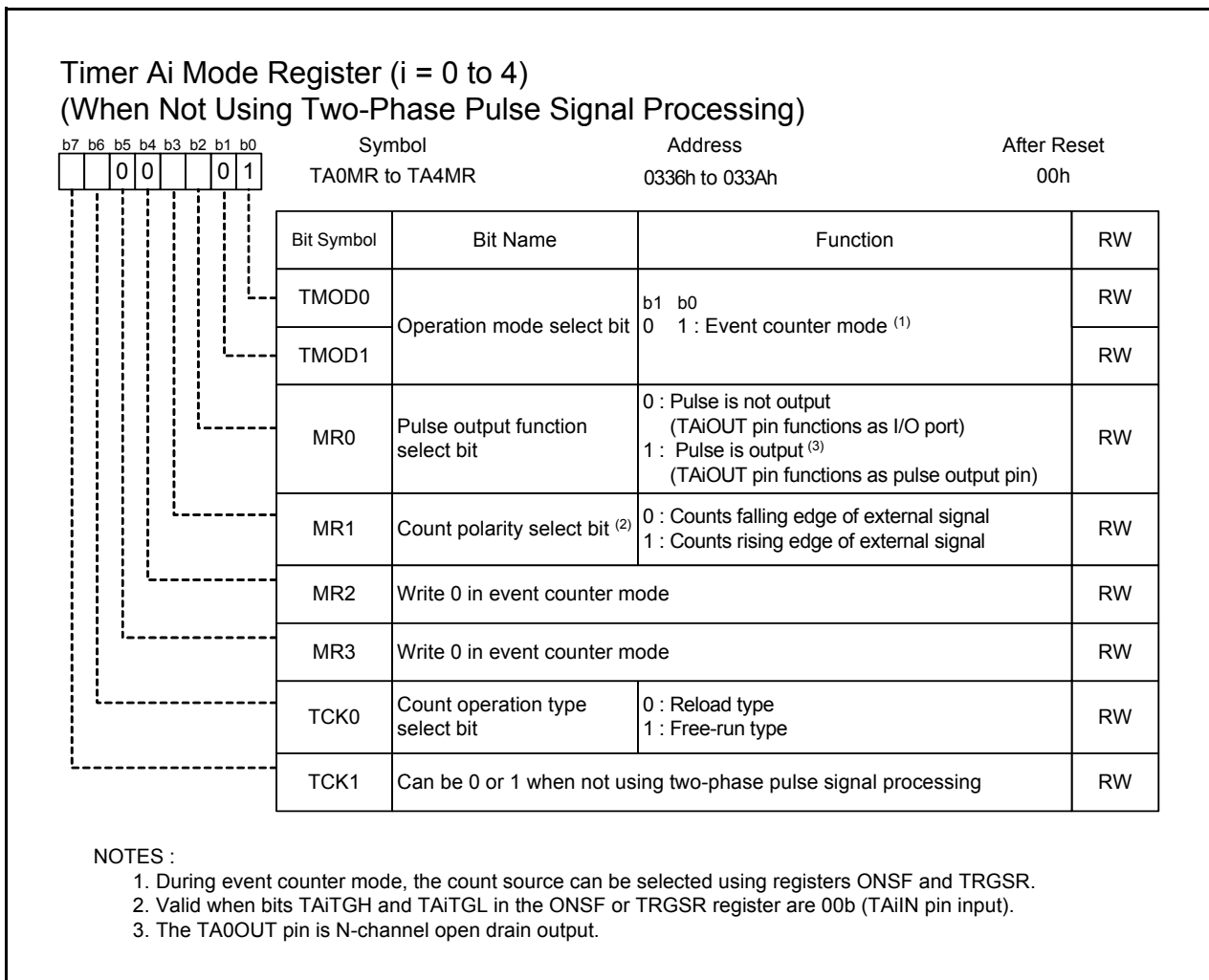
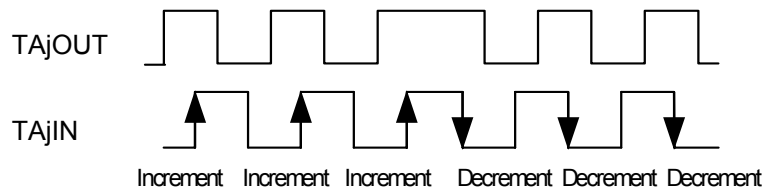
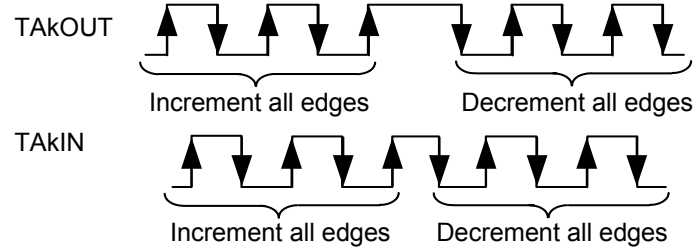


Figure 15.11 TAIMR Register in Event Counter Mode (when not using two-phase pulse signal processing)

Table 15.3 lists Specifications in Event Counter Mode (when processing two-phase pulse signal with Timers A2, A3, and A4). Figure 15.12 shows Registers TA2MR to TA4MR in Event Counter Mode (when using two-phase pulse signal processing with Timers A2, A3, and A4).

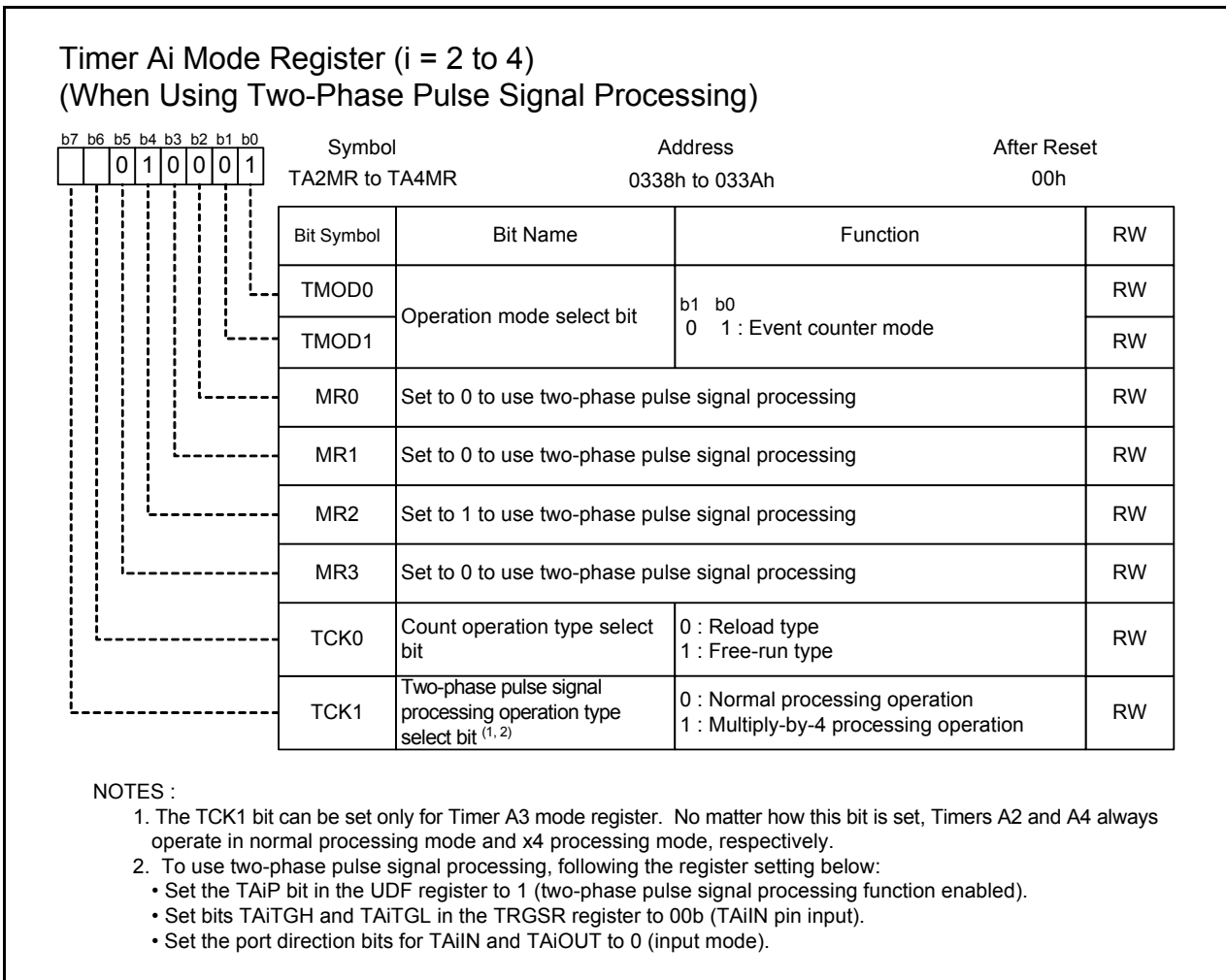
**Table 15.3 Specifications in Event Counter Mode (when processing two-phase pulse signal with Timers A2, A3, and A4)**

Item	Specification
Count Source	Two-phase pulse signals input to TAIiN or TAIiOUT pin
Count Operation	<ul style="list-style-type: none"> <li>Increment or decrement can be selected by two-phase pulse signal</li> <li>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.</li> </ul>
Divide Ratio	<ul style="list-style-type: none"> <li>1/ (FFFFh - n + 1) for increment</li> <li>1/ (n + 1) for decrement</li> </ul> n: set value of the TAI register 0000h to FFFFh
Count Start Condition	Set the TAIiS bit in the TABSR register to 1 (start counting)
Count Stop Condition	Set the TAIiS 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 Value written to the TAI register is written to both reload register and counter When counting Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)
Select Function (1)	<p>Normal processing operation (Timer A2 and Timer A3)                      The timer increments rising edges or decrements falling edges on the TAJiN pin when input signals on the TAJiOUT pin is "H".</p>  <p>Multiply-by-4 processing operation (Timer A3 and Timer A4)                      If the phase relationship is such that TAKIN pin goes "H" when the input signal on the TAKOUT pin is "H," the timer increments rising and falling edges on pins TAKOUT and TAKIN.                      If the phase relationship is such that the TAKIN pin goes "L" when the input signal on the TAKOUT pin is "H," the timer counts down rising and falling edges on pins TAKOUT and TAKIN.</p>  <p>Counter initialization by Z-phase input (Timer A3)                      The timer count value is initialized to 0 by Z-phase input.</p>

i = 2 to 4, j = 2, 3, k = 3, 4

NOTE:

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.



**Figure 15.12 Registers TA2MR to TA4MR in Event Counter Mode (when using two-phase pulse signal processing with Timers A2, A3, and A4)**



### 15.1.2.1 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, multiply-by-4 processing, with Z phase entered from the ZP pin.

Counter initialization by Z-phase input is enabled by writing 0000h to the TA3 register and setting the TAZIE bit in the 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 in the INT2IC register. The Z-phase pulse width applied to the  $\bar{Z}P$  pin must be equal to or greater than one clock cycle of Timer A3 count source.

The counter is initialized at the next count timing after recognizing Z-phase input. Figure 15.13 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 Timer A3 interrupt when using this function.

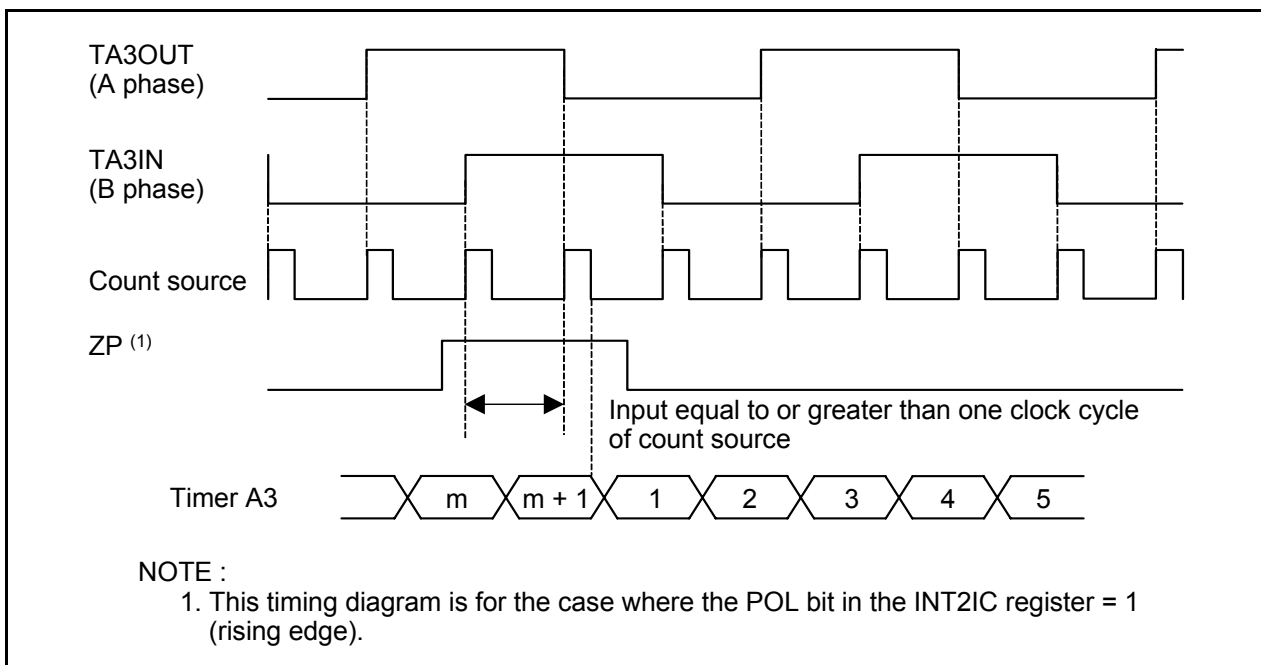


Figure 15.13 Relationship between the Two-Phase Pulse (A Phase and B Phase) and the Z Phase

### 15.1.3 One-Shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger (see Table 15.4). When the trigger occurs, the timer starts up and continues operating for a given period. Figure 15.14 shows the TAI<sub>M</sub>R Register in One-Shot Timer Mode.

**Table 15.4 Specifications in One-shot Timer Mode**

Item	Specification
Count Source	f1TIMAB, f2TIMAB, f8TIMAB, f32TIMAB, f64TIMAB, fOCO-S, fC32
Count Operation	<ul style="list-style-type: none"> <li>• Decrement</li> <li>• When the counter reaches 0000h, it stops counting after reloading a new value.</li> <li>• If a trigger occurs when counting, the timer reloads a new count and restarts counting.</li> </ul>
Divide Ratio	1/n      n: set value of the TAI register    0000h to FFFFh However, the counter does not work if the divide-by-n value is set to 0000h.
Count Start Condition	The TAI <sub>S</sub> bit in the TABSR register = 1 (start counting) and one of the following triggers occurs. <ul style="list-style-type: none"> <li>• External trigger input from the TAI<sub>IN</sub> pin</li> <li>• Timer B2 overflow or underflow, Timer A<sub>j</sub> (j = i - 1, except j = 4 if i = 0) overflow or underflow, Timer A<sub>k</sub> (k = i + 1, except k = 0 if i = 4) overflow or underflow</li> <li>• The TAI<sub>OS</sub> bit in the ONSF register is set to 1 (timer starts)</li> </ul>
Count Stop Condition	<ul style="list-style-type: none"> <li>• When the counter is reloaded after reaching 0000h</li> <li>• The TAI<sub>S</sub> bit is set to 0 (stop counting)</li> </ul>
Interrupt Request Generation Timing	When the counter reaches 0000h
TAI <sub>IN</sub> Pin Function	I/O port or trigger input
TAI <sub>OUT</sub> Pin Function	I/O port or pulse output
Read from Timer	An indeterminate value is read by reading the TAI register
Write to Timer	<ul style="list-style-type: none"> <li>• When not counting and until the 1st count source is input after counting starts Value written to the TAI register is written to both reload register and counter</li> <li>• When counting (after 1st count source input) Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>
Select Function	<ul style="list-style-type: none"> <li>• Pulse output function The timer outputs low when not counting and high when counting.</li> <li>• Output polarity control While the output polarity of TAI<sub>OUT</sub> pin is inverted (the TAI<sub>S</sub> bit is set to 0 (stop counting)), the pin outputs high.</li> </ul>

i = 0 to 4

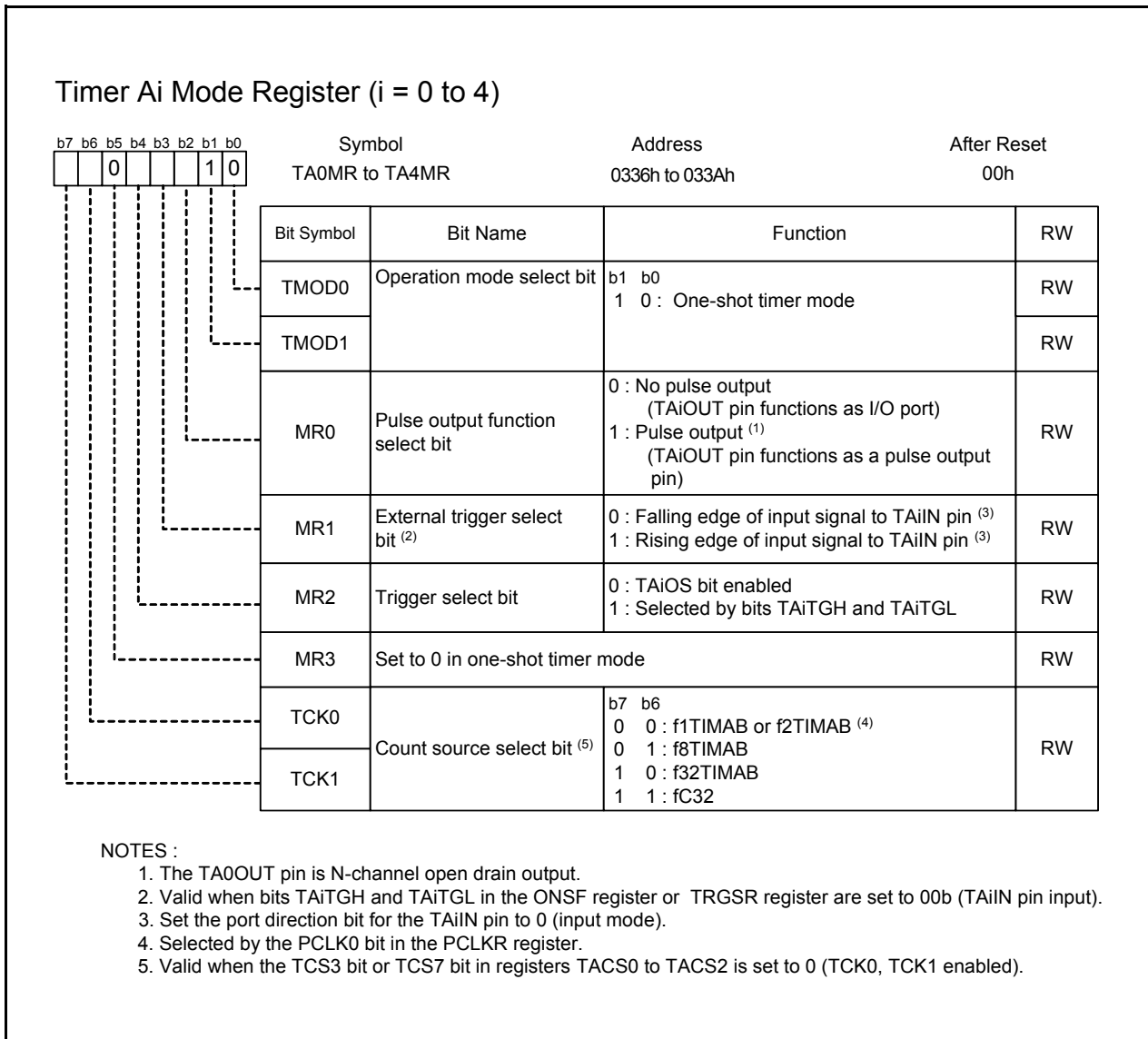


Figure 15.14 TAIiMR Register in One-Shot Timer Mode

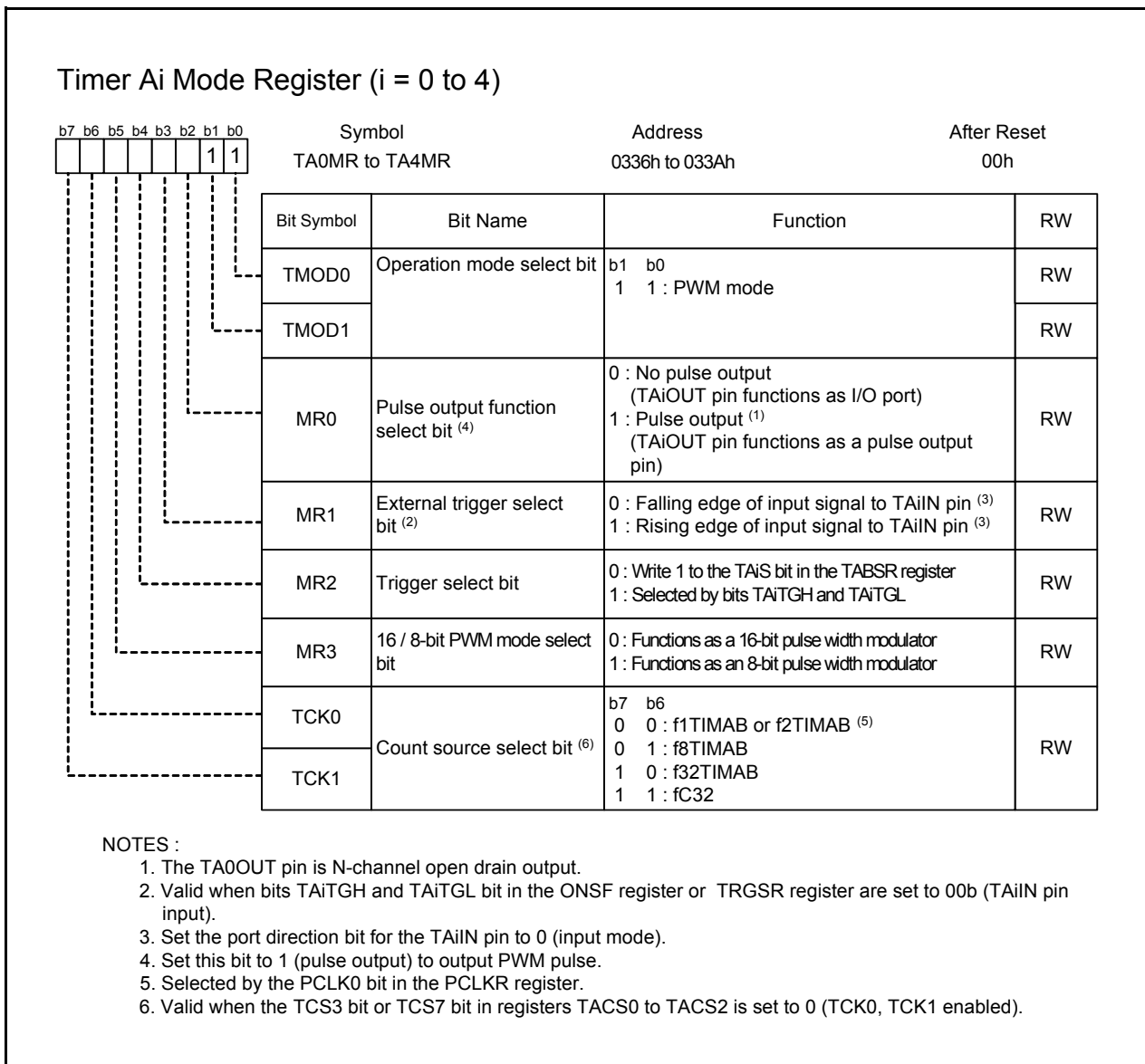
### 15.1.4 Pulse Width Modulation (PWM) Mode

In PWM mode, the timer outputs pulses of a given width in succession (see Table 15.5). The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator. Figure 15.15 shows TAI MR Register in PWM Mode. Figures 15.16 and 15.17 show an Example of 16-Bit Pulse Width Modulator Operation and 8-bit Pulse Width Modulator Operation, respectively.

**Table 15.5 Specifications in PWM Mode**

Item	Specification
Count Source	f1TIMAB, f2TIMAB, f8TIMAB, f32TIMAB, f64TIMAB, fOCO-S, fC32
Count Operation	<ul style="list-style-type: none"> <li>Decrement (operating as an 8-bit or a 16-bit pulse width modulator)</li> <li>The timer reloads a new value at a rising edge of PWM pulse and continues counting.</li> <li>The timer is not affected by a trigger that occurs during counting.</li> </ul>
16-bit PWM	<ul style="list-style-type: none"> <li>Pulse width <math>n / f_j</math>                                      <math>n</math>: set value of the TAI register</li> <li>Cycle time <math>(2^{16} - 1) / f_j</math> fixed                      <math>f_j</math>: count source frequency (1TIMAB, f2TIMAB, f8TIMAB, f32TIMAB, f64TIMAB, fOCO-S, fC32)</li> </ul>
8-bit PWM	<ul style="list-style-type: none"> <li>Pulse width <math>n \times (m+1) / f_j</math>                              <math>n</math>: set value of the TAI register high-order address</li> <li>Cycle time <math>(2^8-1) \times (m+1) / f_j</math>                      <math>m</math>: set value of the TAI register low-order address</li> </ul>
Count Start Condition	<ul style="list-style-type: none"> <li>The TAI S bit of the TABSR register is set to 1 (start counting)</li> <li>The TAI S bit = 1 and external trigger input from the TAI IN pin</li> <li>The TAI S bit = 1 and one of the following external triggers occurs                      Timer B2 overflow or underflow,                      Timer Aj (<math>j = i - 1</math>, except <math>j = 4</math> if <math>i = 0</math>) overflow or underflow,                      Timer Ak (<math>k = i + 1</math>, except <math>k = 0</math> if <math>i = 4</math>) overflow or underflow</li> </ul>
Count Stop Condition	The TAI S bit is set to 0 (stop counting)
Interrupt Request Generation Timing	On the falling edge of PWM pulse
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 the TAI register
Write to Timer	<ul style="list-style-type: none"> <li>When not counting                      Value written to the TAI register is written to both reload register and counter</li> <li>When counting                      Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>
Select Function	<ul style="list-style-type: none"> <li>Output polarity control                      While the output polarity of TAI OUT pin is inverted (the TAI S bit is set to 0 (stop counting)), the pin outputs high.</li> </ul>

$i = 0$  to 4



**Figure 15.15 TAIiMR Register in PWM Mode**

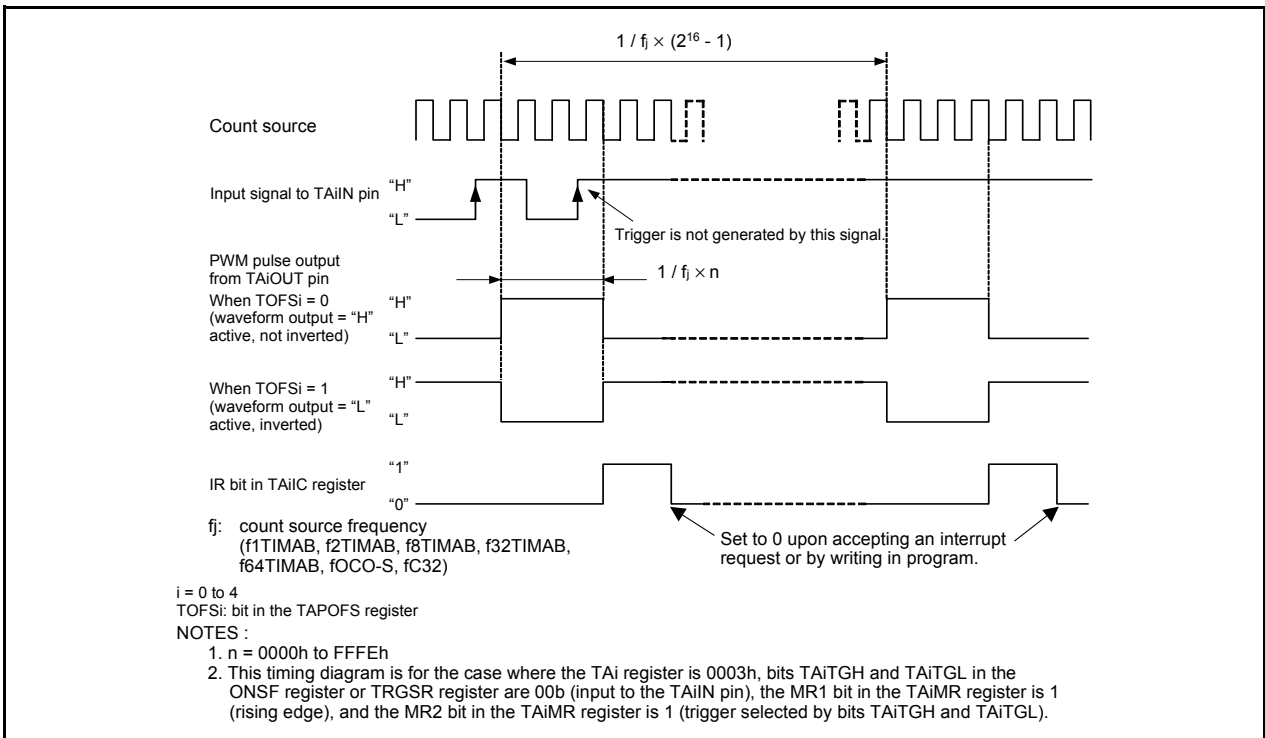


Figure 15.16 Example of 16-Bit Pulse Width Modulator Operation

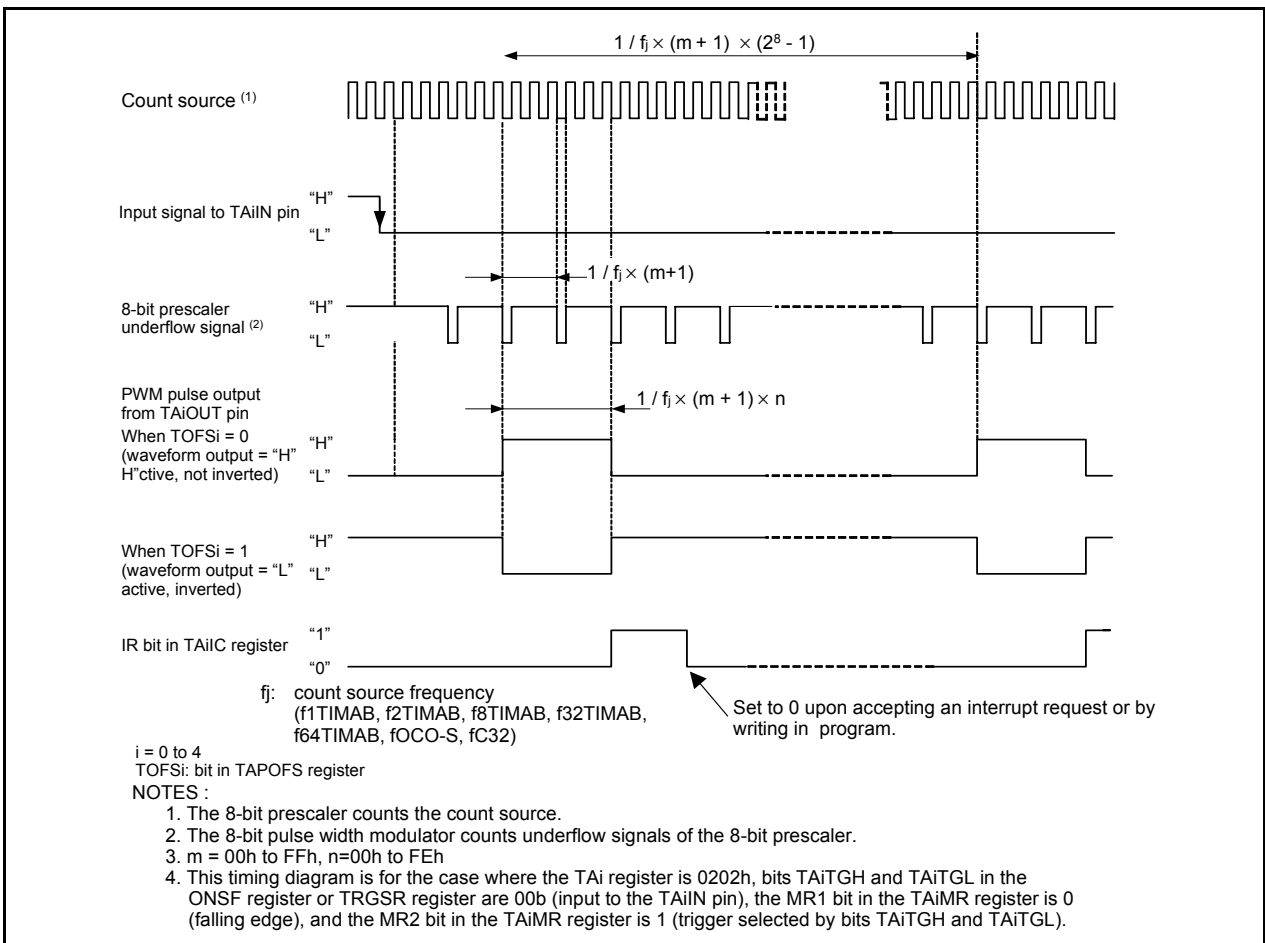


Figure 15.17 Example of 8-Bit Pulse Width Modulator Operation

## 15.2 Timer B

Figure 15.18 shows Timer B Block Diagram. Figures 15.19 to 15.21 show registers related to Timer B. Timer B supports the following three modes. Use bits TMOD1 and TMOD0 in the TBiMR register (i = 0 to 5) 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 Measurement Mode: The timer measures pulse period or pulse width of an external signal.

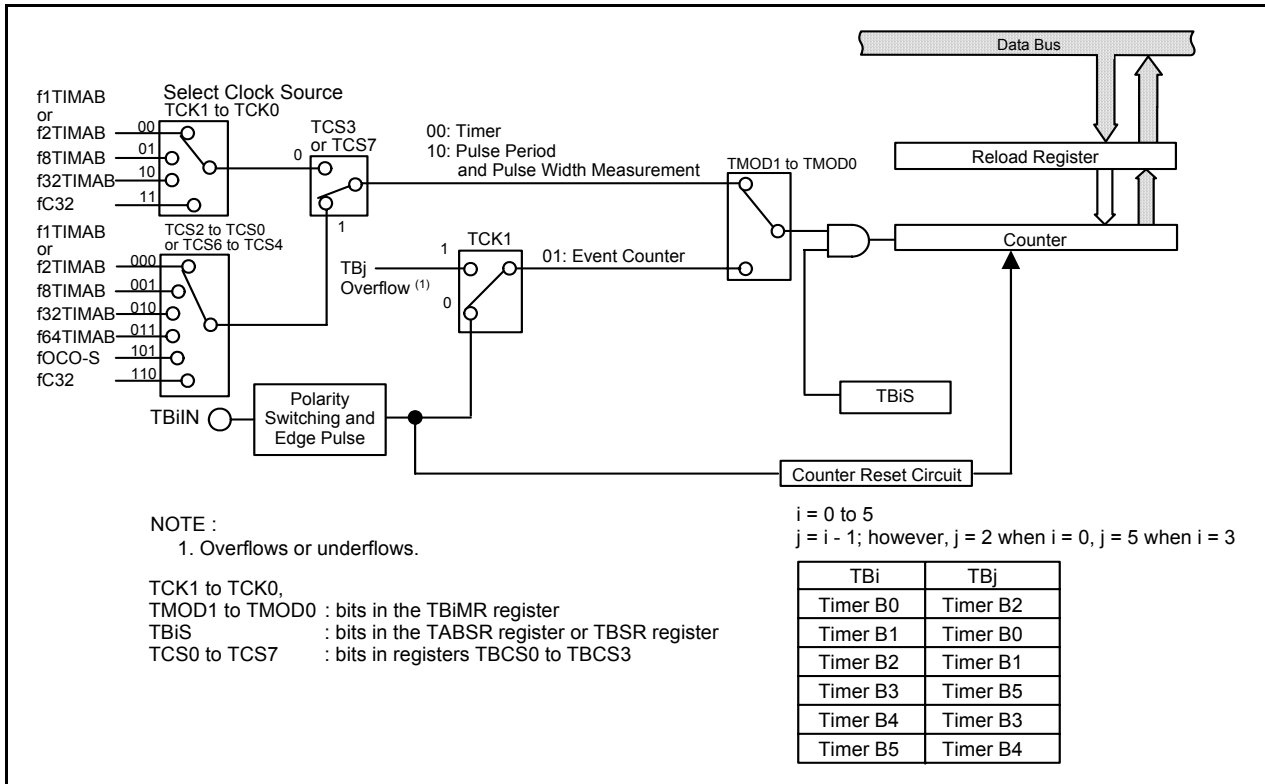


Figure 15.18 Timer B Block Diagram

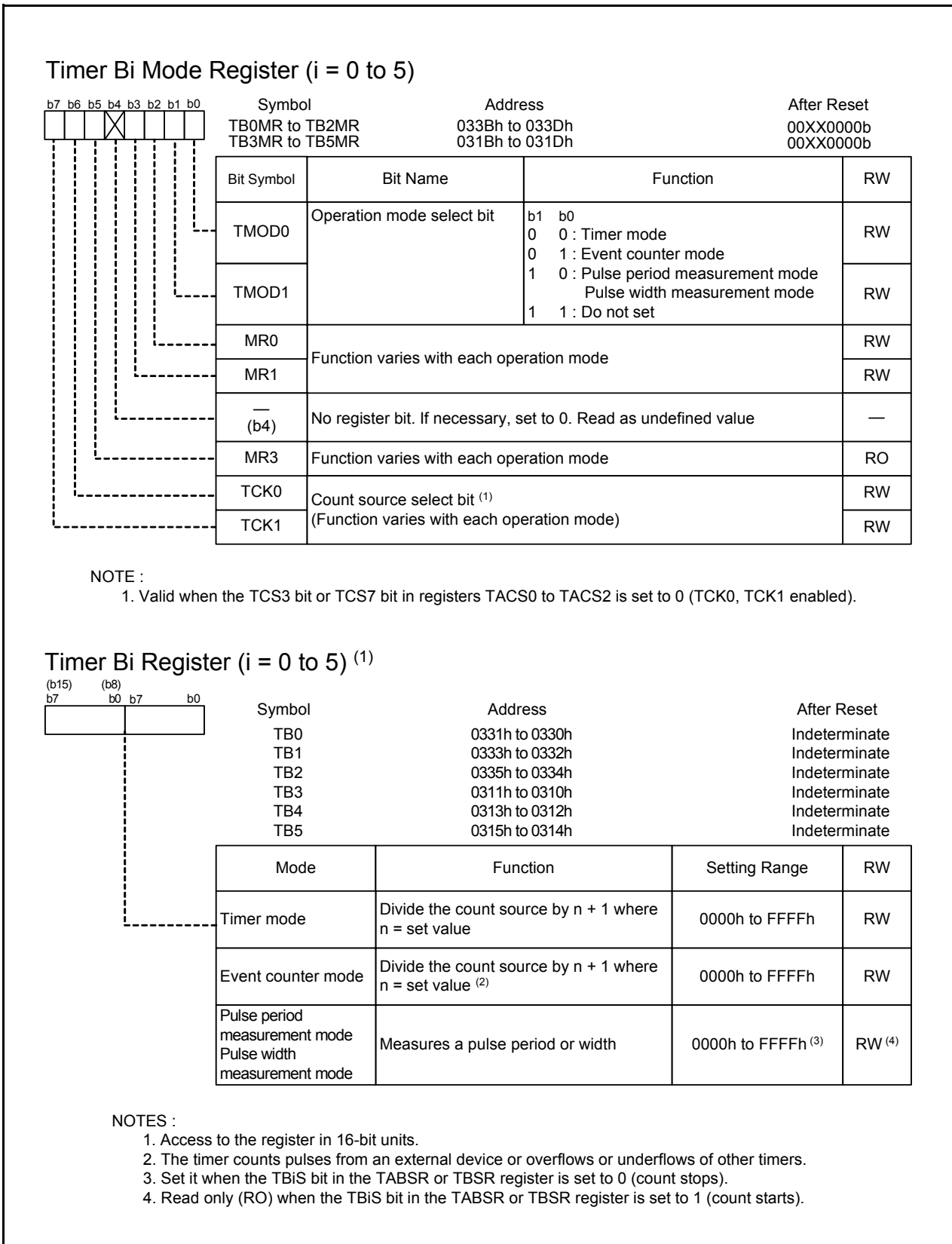


Figure 15.19 Register TB0MR to TB5MR and TB0 to TB5



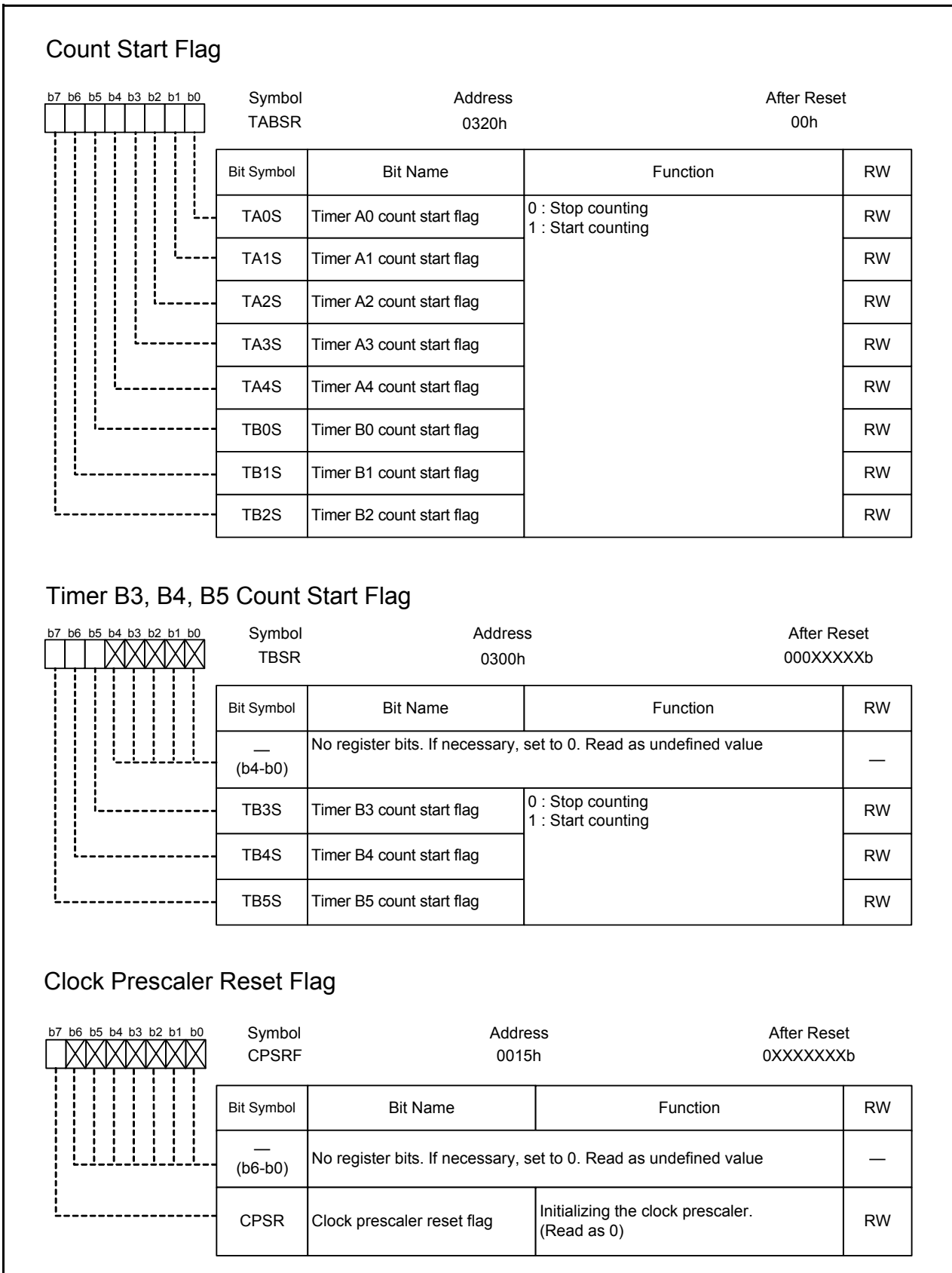


Figure 15.20 Register TABSR, TBSR, and CPSRF

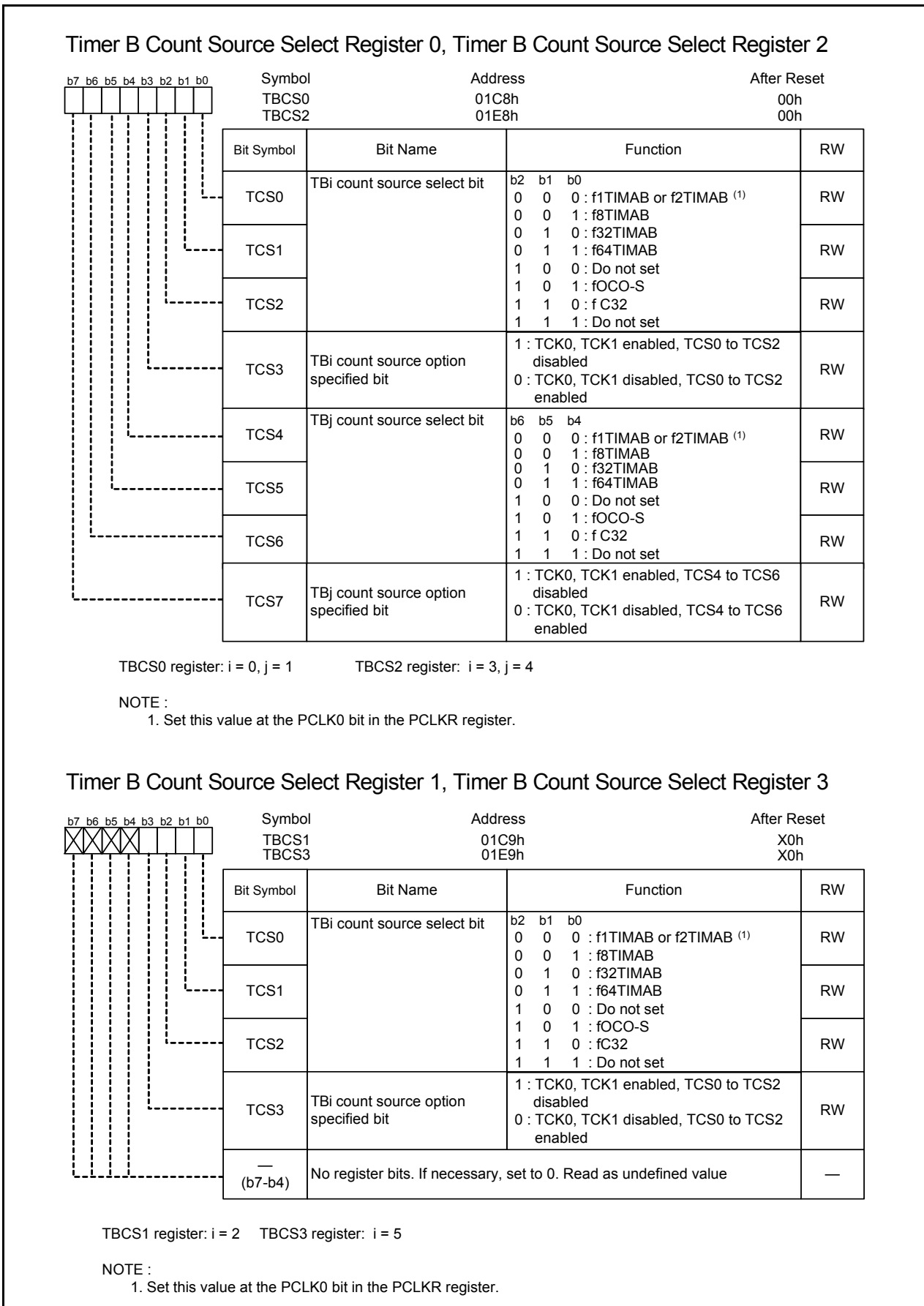


Figure 15.21 Registers TBCS0, TBCS1, TBCS2, and TBCS3

### 15.2.1 Timer Mode

In timer mode, the timer counts a count source generated internally (see **Table 15.6**). Figure 15.22 shows the TBIMR Register in Timer Mode.

**Table 15.6 Specifications in Timer Mode**

Item	Specification
Count Source	f1TIMAB, f2TIMAB, f8TIMAB, f32TIMAB, f64TIMAB, fOCO-S, fC32
Count Operation	<ul style="list-style-type: none"> <li>• Decrement</li> <li>• When the timer underflows, it reloads the reload register contents and continues counting</li> </ul>
Divide Ratio	$1 / (n + 1)$ n: set value of the TBi register      0000h to FFFFh
Count Start Condition	Set the TBiS bit <sup>(1)</sup> to 1 (start counting)
Count Stop Condition	Set the TBiS bit to 0 (stop counting)
Interrupt Request Generation Timing	Timer underflow
TBiIN Pin Function	I/O port
Read from Timer	Count value can be read by reading the TBi register
Write to Timer	<ul style="list-style-type: none"> <li>• When not counting Value written to the TBi register is written to both reload register and counter</li> <li>• When counting Value written to the TBi register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>

i = 0 to 5

NOTE:

1. Bits TB0S to TB2S are assigned to bits 5 to 7 in the TABSR register, and bits TB3S to TB5S are assigned to bits 5 to 7 in the TBSR register.

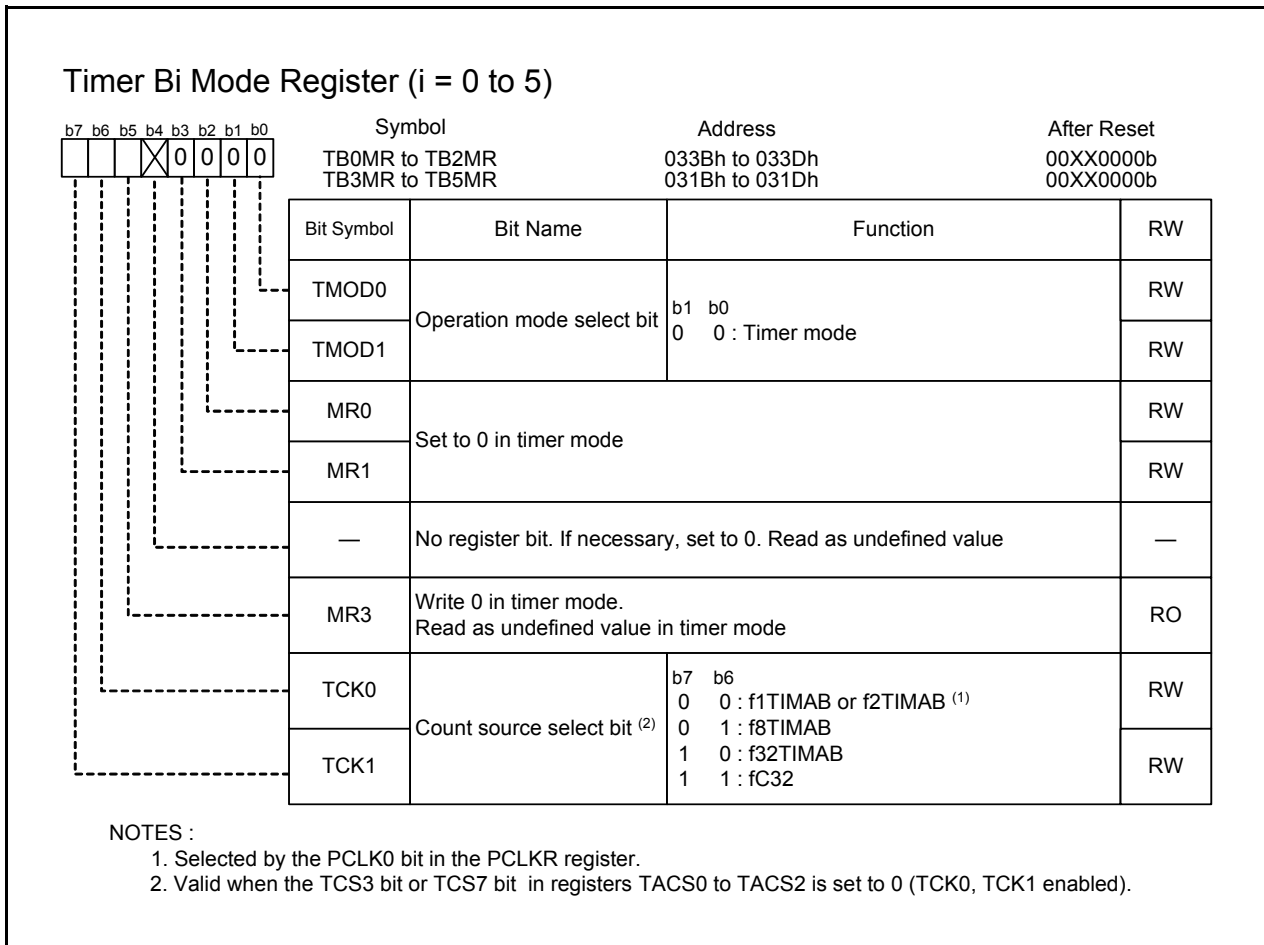


Figure 15.22 TBiMR Register in Timer Mode

### 15.2.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers (see **Table 15.7**). Figure 15.23 shows the TBiMR Register in Event Counter Mode.

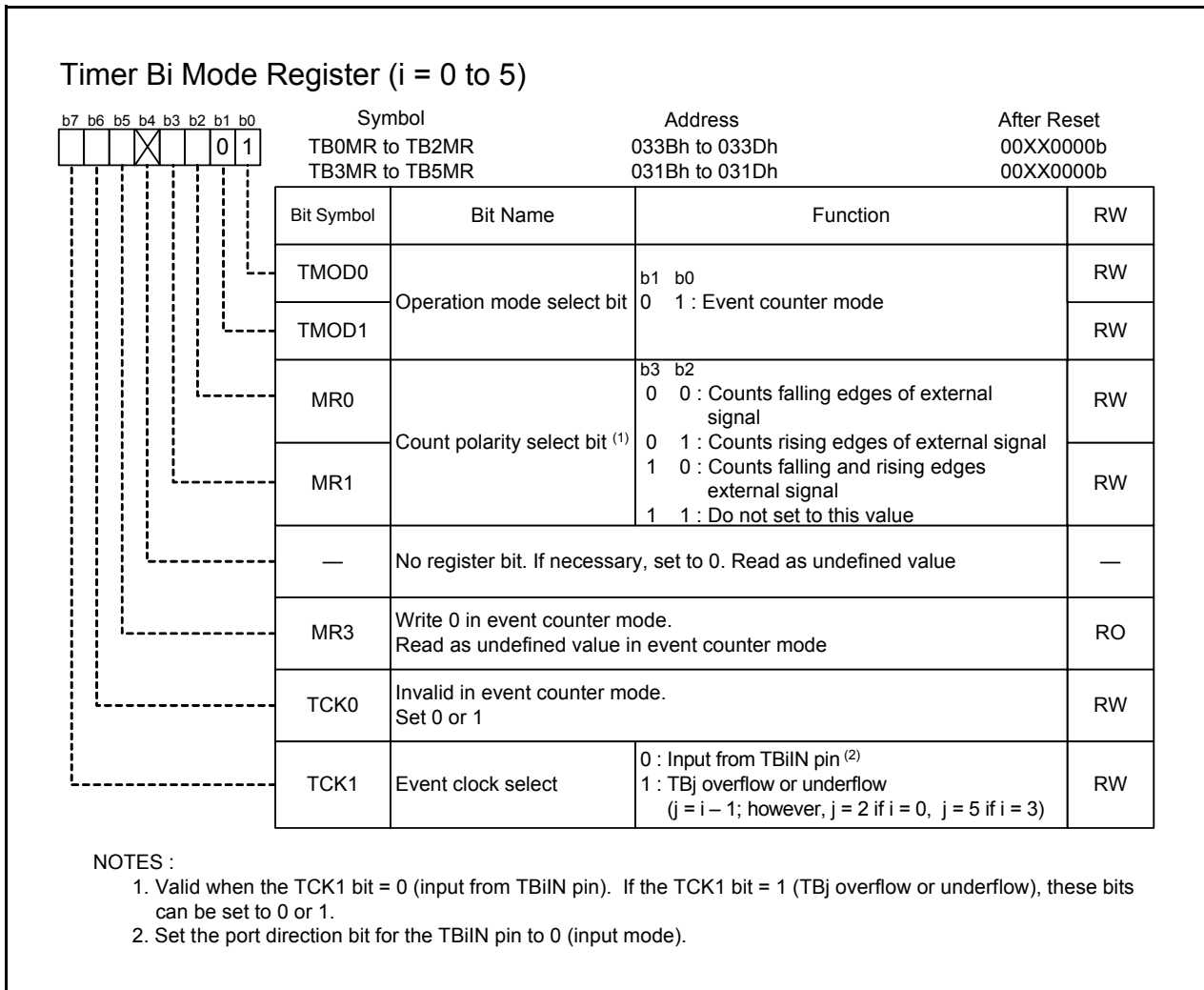
**Table 15.7 Specifications in Event Counter Mode**

Item	Specification
Count Source	<ul style="list-style-type: none"> <li>External signals input to TBiIN pin (effective edge rising edge, falling edge, or both rising and falling edges) can be selected in a program)</li> <li>Timer Bj overflow or underflow (<math>j = i - 1</math>, except <math>j = 2</math> if <math>i = 0</math>, <math>j = 5</math> if <math>i = 3</math>)</li> </ul>
Count Operation	<ul style="list-style-type: none"> <li>Decrement</li> <li>When the timer underflows, it reloads the reload register contents and continues counting.</li> </ul>
Divide Ratio	$1 / (n + 1)$ n: set value of the TBi register    0000h to FFFFh
Count Start Condition	Set the TBiS bit <sup>(1)</sup> to 1 (start counting)
Count Stop Condition	Set the 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 the TBi register.
Write to Timer	<ul style="list-style-type: none"> <li>When not counting Value written to the TBi register is written to both reload register and counter</li> <li>When counting Value written to the TBi register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>

i = 0 to 5

NOTE:

- Bits TB0S to TB2S are assigned to bits 5 to 7 in the TABSR register, and bits TB3S to TB5S are assigned to bits 5 to 7 in the TBSR register.



**Figure 15.23 TBiMR Register in Event Counter Mode**

### 15.2.3 Pulse Period and Pulse Width Measurement Modes

In pulse period and pulse width measurement mode, the timer measures pulse period or pulse width of an external signal (see **Table 15.8**). Figure 15.24 shows the TBiMR Register in Pulse Period and Pulse Width Measurement Mode. Figure 15.25 shows the Operation Timing when Measuring a Pulse Period. Figure 15.26 shows the Operation Timing when Measuring a Pulse Width.

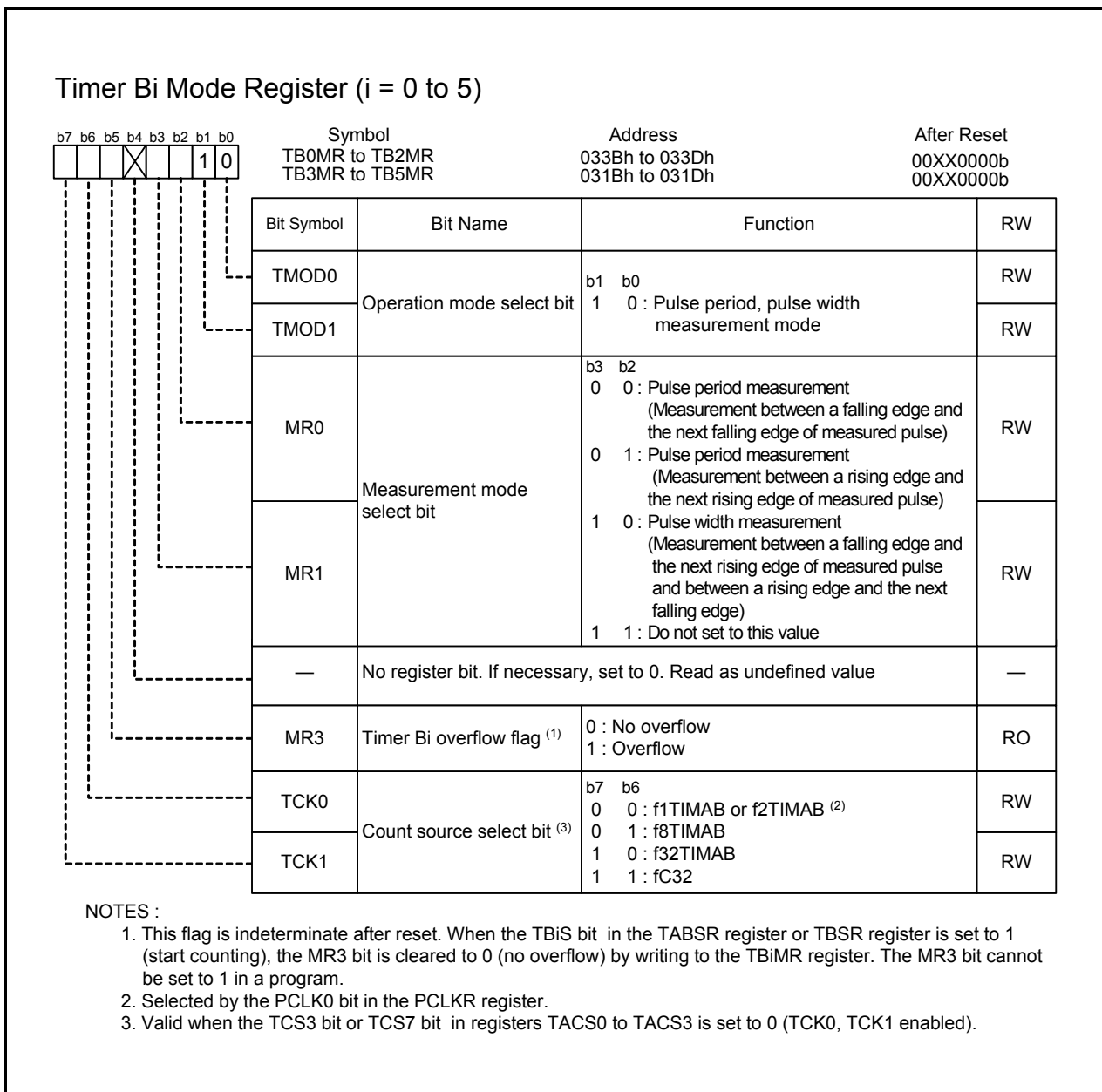
**Table 15.8 Specifications in Pulse Period and Pulse Width Measurement Mode**

Item	Specification
Count Source	f1TIMAB, f2TIMAB, f8TIMAB, f32TIMAB, f64TIMAB, fOCO-S, fC32
Count Operation	<ul style="list-style-type: none"> <li>Increment</li> <li>Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to 0000h to continue counting.</li> </ul>
Count Start Condition	Set the TBiS bit <sup>(3)</sup> to 1 (start counting)
Count Stop Condition	Set the TBiS bit to 0 (stop counting)
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>When an effective edge of measurement pulse is input <sup>(1)</sup></li> </ul> Timer overflow. When an overflow occurs, the MR3 bit in the TBiMR register is set to 1 (overflowed) simultaneously.
TBiIN Pin Function	Measurement pulse input
Read from Timer	Contents of the reload register (measurement result) can be read by reading the TBi register <sup>(2)</sup>
Write to Timer	Value written to the TBi register is written to neither reload register nor counter

i = 0 to 5

NOTES:

1. Interrupt request is not generated when the first effective edge is input after the timer started counting.
2. Value read from the TBi register is indeterminate until the second valid edge is input after the timer starts counting.
3. Bits TB0S to TB2S are assigned to bits 5 to 7 in the TABSR register, and bits TB3S to TB5S are assigned to bits 5 to 7 in the TBSR register.



**Figure 15.24 TBiMR Register in Pulse Period and Pulse Width Measurement Mode**



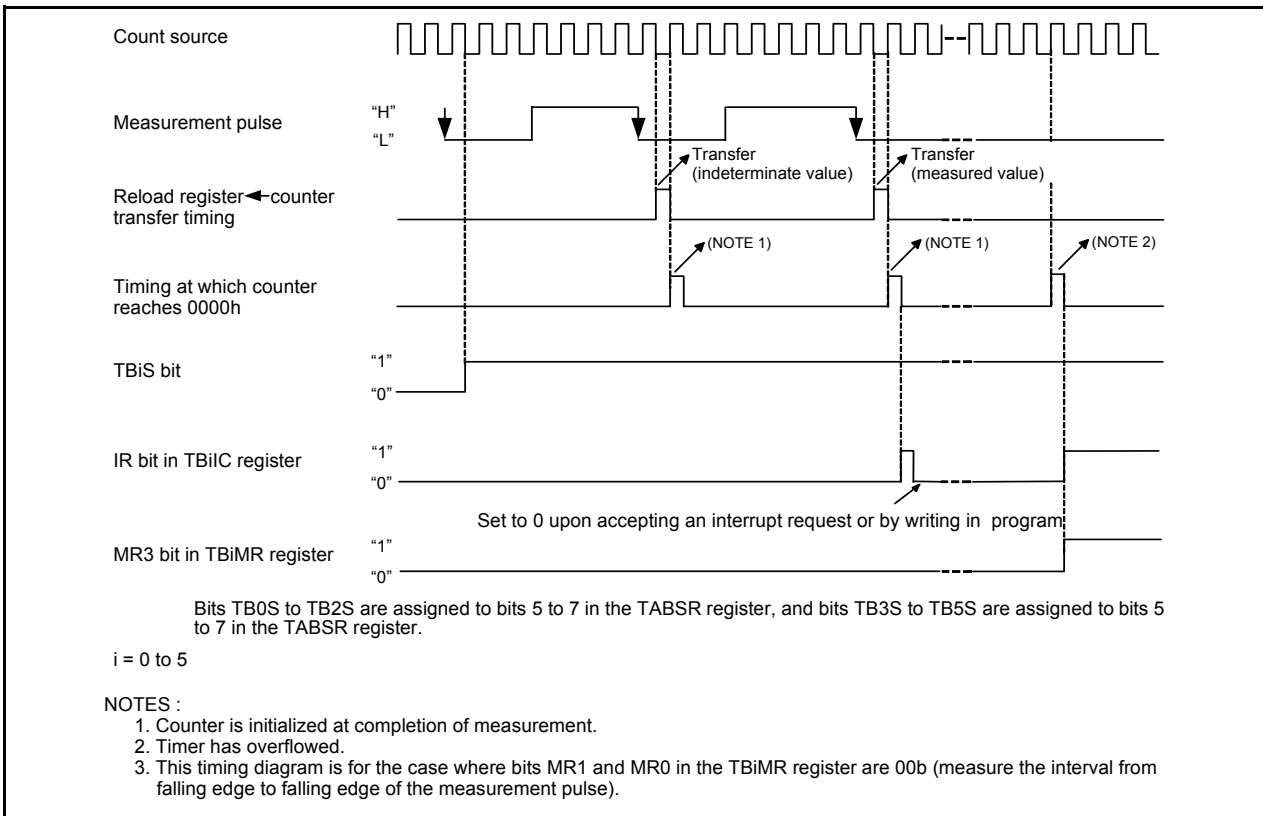


Figure 15.25 Operation Timing when Measuring a Pulse Period

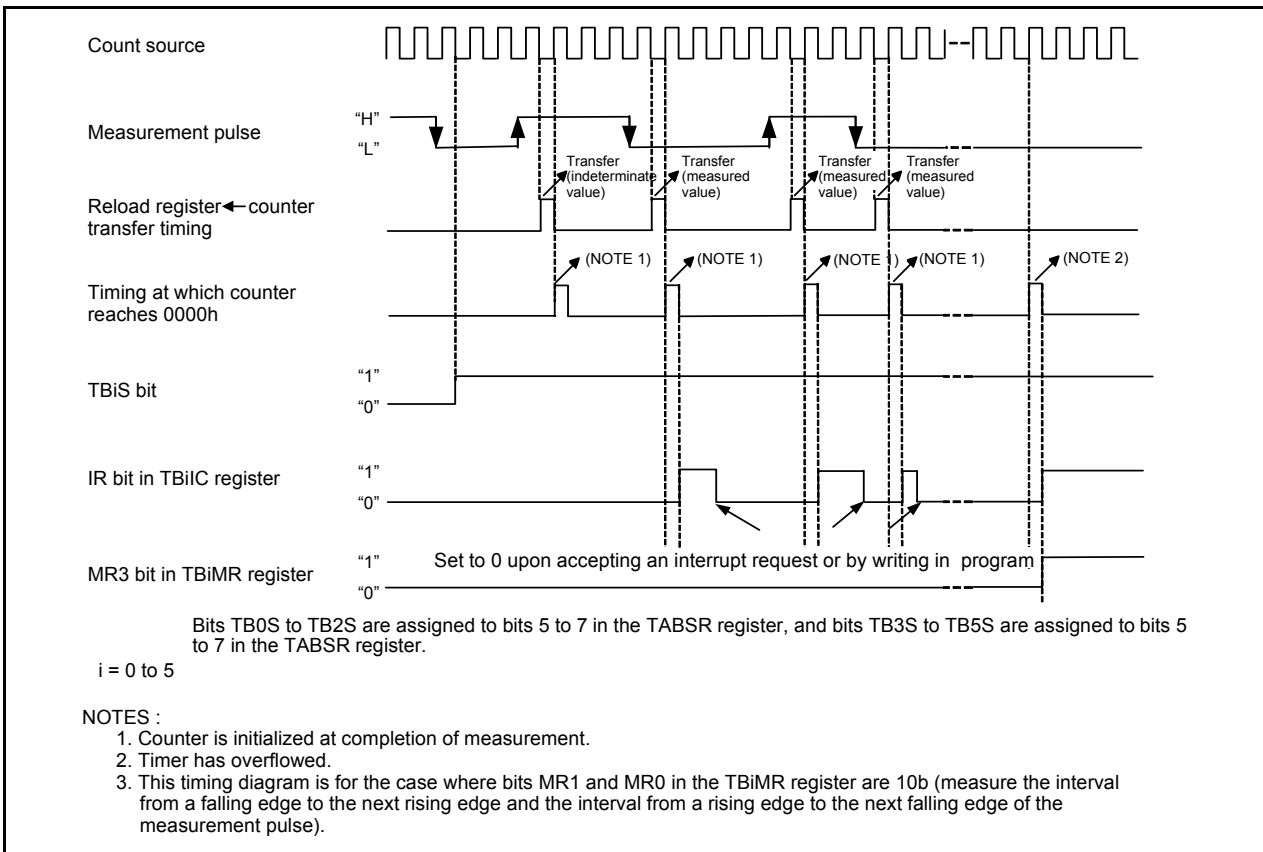


Figure 15.26 Operation Timing when Measuring a Pulse Width

## 16. Three-Phase Motor Control Timer Function

Timers A1, A2, A4, and B2 can be used to output three-phase motor drive waveforms. Table 16.1 lists the Three-phase Motor Control Timer Functions Specifications. Figure 16.1 shows the Three-phase Motor Control Timer Functions Block Diagram. Also, the related registers are shown on Figures 16.2 to 16.7.

**Table 16.1 Three-phase Motor Control Timer Functions Specifications**

Item	Specification
Three-Phase Waveform Output Pin	Six pins (U, $\bar{U}$ , V, $\bar{V}$ , W, $\bar{W}$ )
Forced Cutoff Input (1)	Input "L" to the $\bar{SD}$ pin
Used Timers	Timer A4, A1, A2 (used in one-shot timer mode) Timer A4: U- and $\bar{U}$ -phase waveform control Timer A1: V- and $\bar{V}$ -phase waveform control Timer A2: W- and $\bar{W}$ -phase waveform control Timer B2 (used in timer mode) Carrier wave cycle control Dead time timer (3 eight-bit timers and shared reload register) Dead time control
Output Waveform	Triangular wave modulation, sawtooth wave modulation • Enable to output "H" or "L" for one cycle • Enable to set positive-phase level and negative-phase level independently
Carrier Wave Cycle	Triangular wave modulation : count source x (m + 1) x 2 Sawtooth wave modulation : count source x (m + 1) m: setting value of the TB2 register, 0000h to FFFFh Count source: f1TIMAB, f2TIMAB, f8TIMAB, f32TIMAB, f64TIMAB, fOCO-S, fC32
Three-Phase PWM Output Width	Triangular wave modulation: count source x n x 2 Sawtooth wave modulation: count source x n n: setting value of registers TA4, TA1, and TA2 (of registers TA4, TA41, TA1, TA11, TA2, and TA21 when setting the INV11 bit to 1), 0001h to FFFFh Count source: f1TIMAB, f2TIMAB, f8TIMAB, f32TIMAB, f64TIMAB, fOCO-S, fC32
Dead Time	Count source x p, or no dead time p: setting value of the DTT register, 01h to FFh Count source: f1TIMAB, f2TIMAB, f1TIMAB divided by 2, f2TIMAB divided by 2
Active Level	Enable to select "H" or "L"
Positive and Negative-Phase Concurrent Active Disable Function	Positive-and negative-phases concurrent active disable function Positive-and negative-phases concurrent active detect function
Interrupt Frequency	Timer B2 interrupt is generated every q times q: carrier wave cycle-to-cycle basis, 1 to 15

NOTES:

1. Forced cutoff with  $\bar{SD}$  input is effective when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by  $\bar{SD}$  input enabled). If an "L" signal is applied to the  $\bar{SD}$  pin when the IVPCR1 bit is 1, the related pins go to a high-impedance state regardless of which functions of those pins are being used.
2. Related pins: P7\_2/CLK2/TA1OUT/ $\bar{V}$ , P7\_3/ $\bar{CTS2}$ / $\bar{RTS2}$ /TA1IN/ $\bar{V}$ , P7\_4/TA2OUT/W, P7\_5/TA2IN/ $\bar{W}$ , P8\_0/TA4OUT/RXD5/SCL5/U, P8\_1/TA4IN/ $\bar{CTS5}$ / $\bar{RTS5}$ / $\bar{U}$

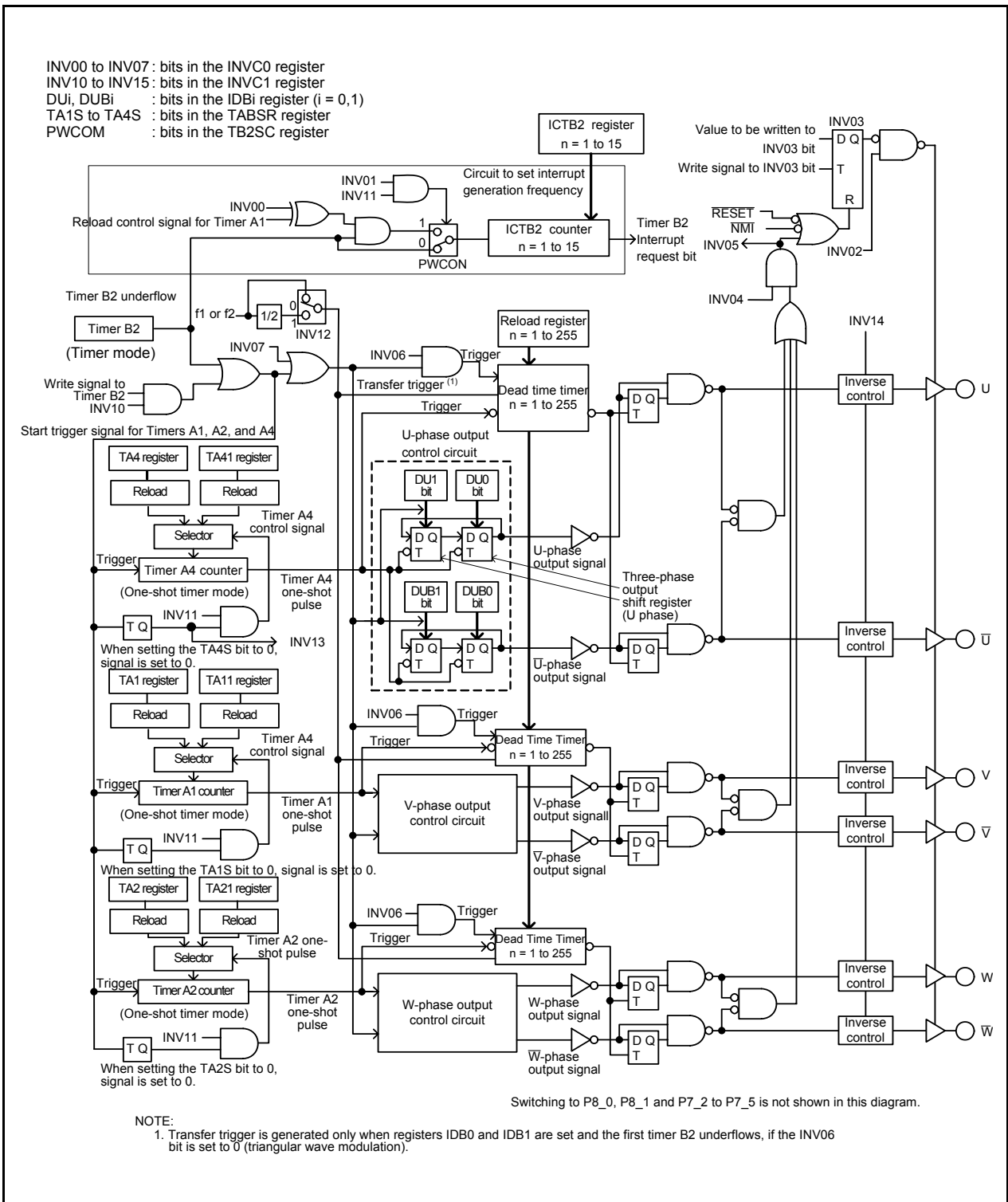


Figure 16.1 Three-phase Motor Control Timer Functions Block Diagram

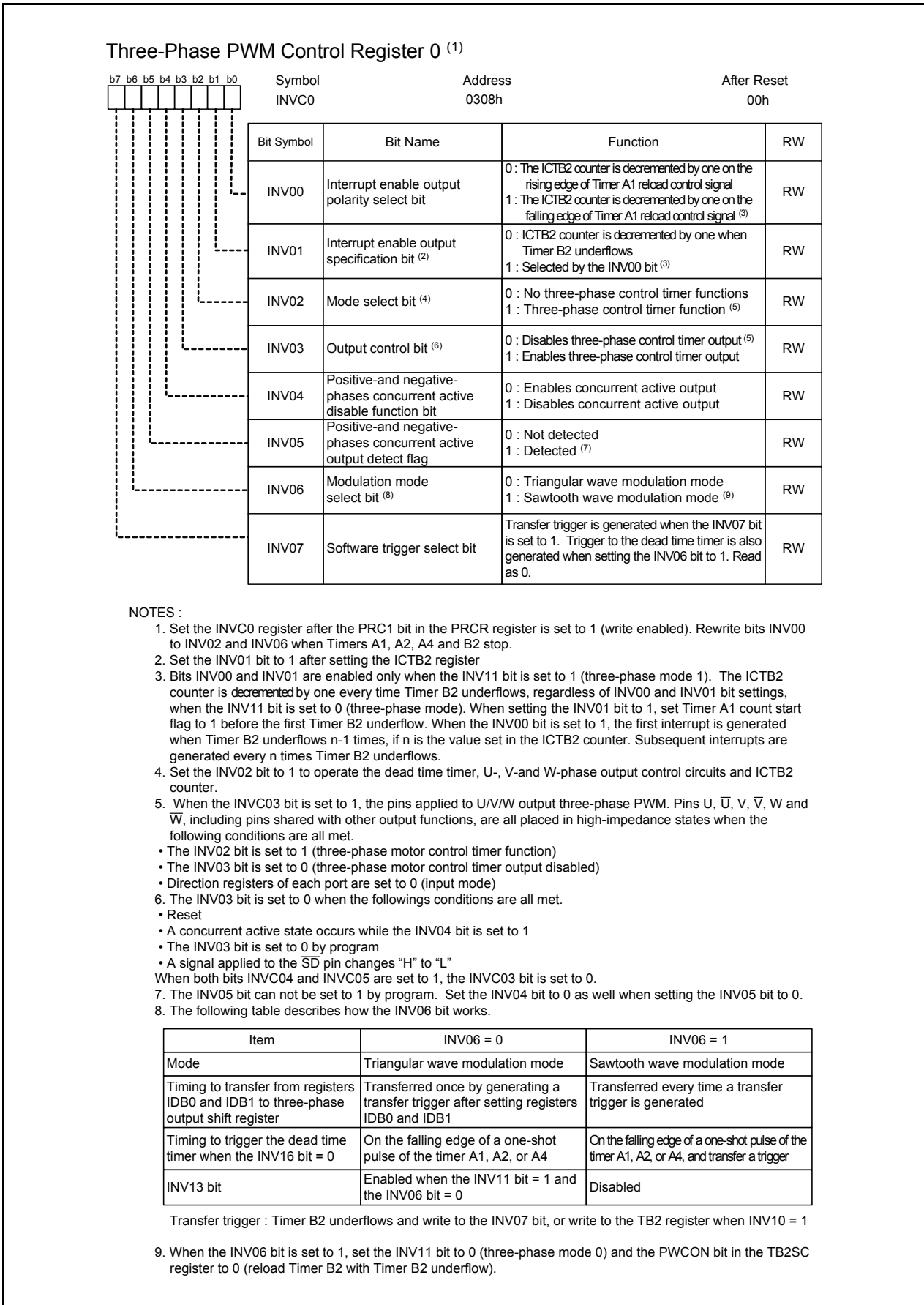
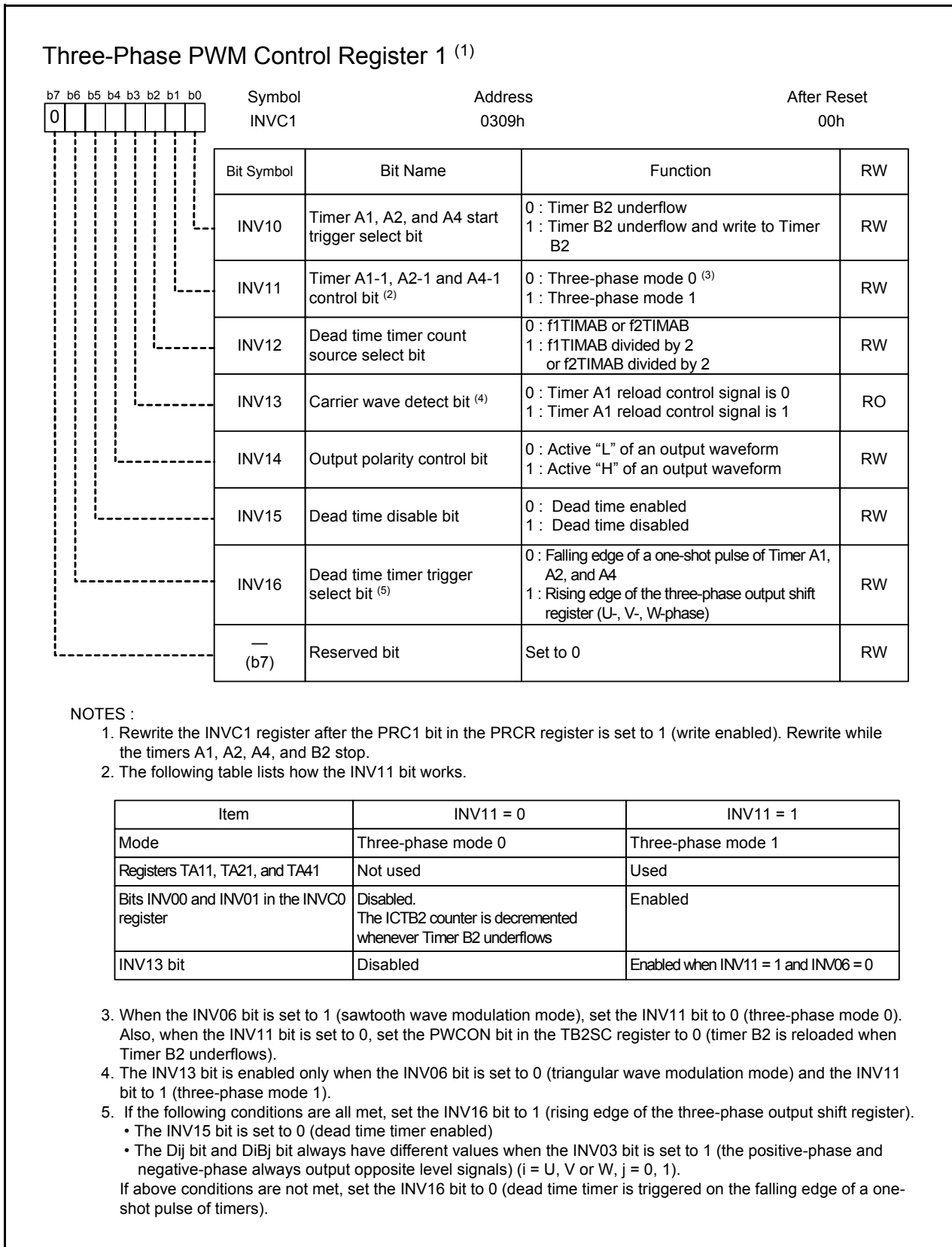


Figure 16.2 INVC0 Register



**Figure 16.3 INVC1 Register**

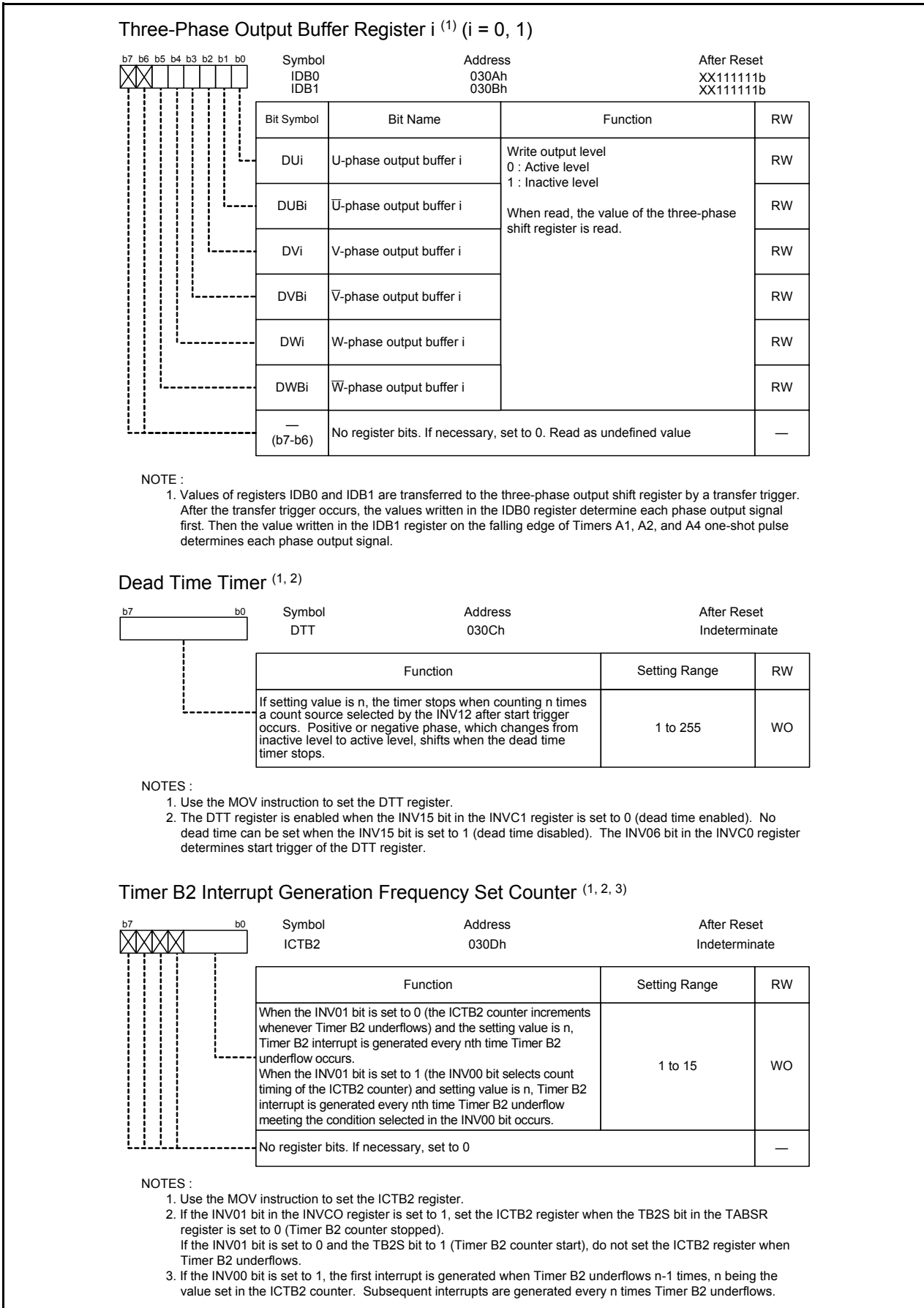


Figure 16.4 Registers IDB0, IDB1, DTT, and ICTB2

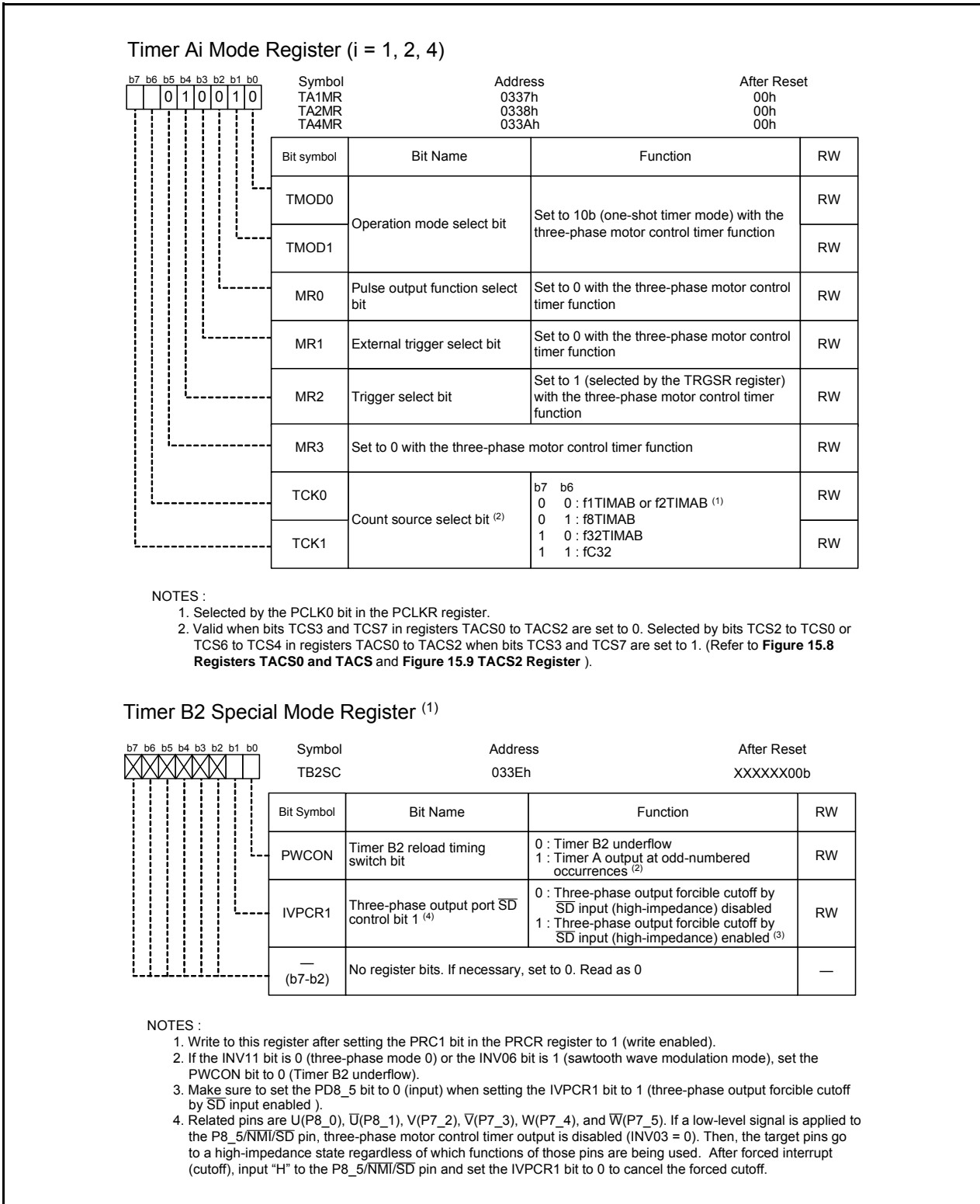


Figure 16.5 Registers TA1, TA2, TA4, TA11, TA21, TA41, and TB2SC

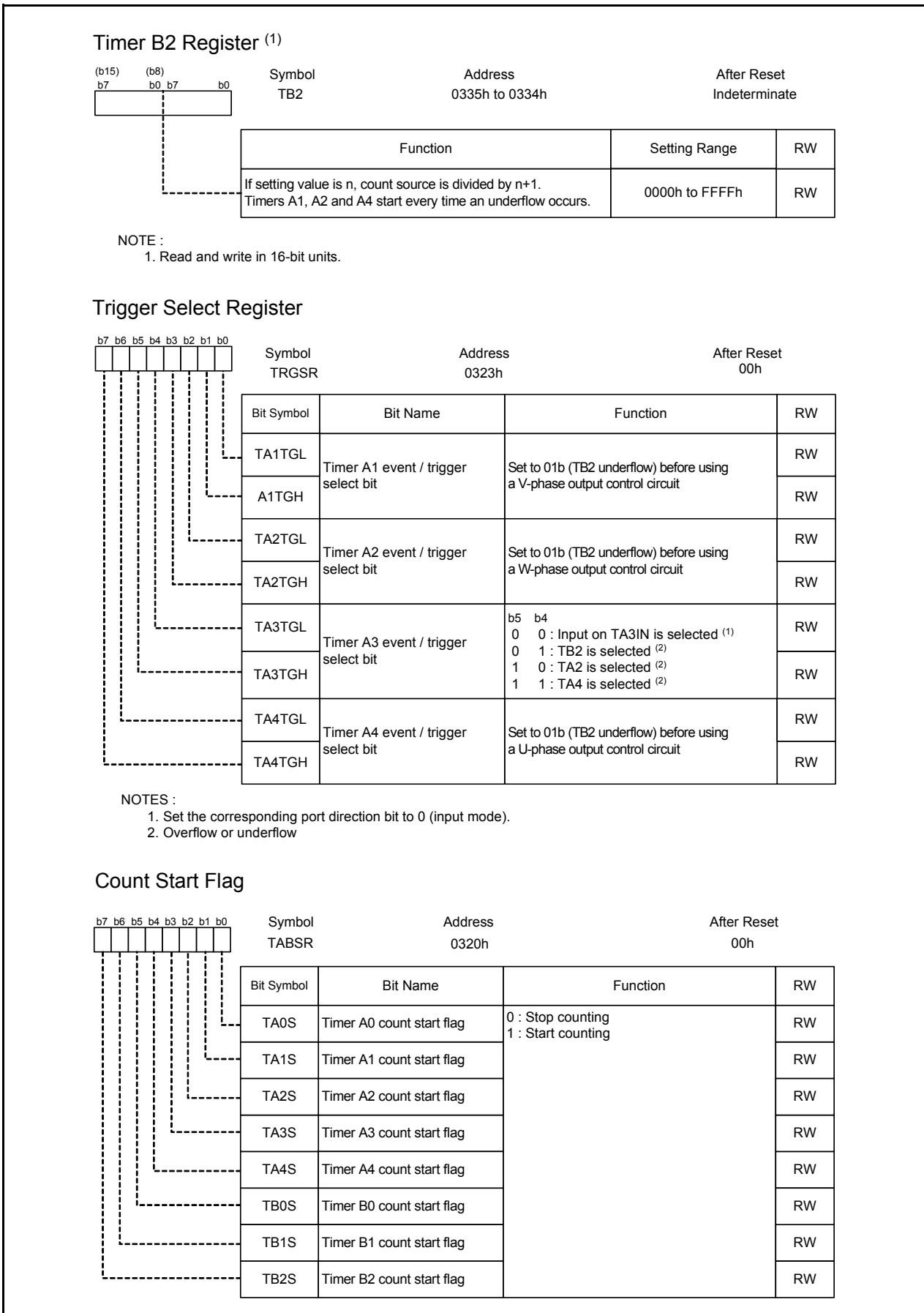


Figure 16.6 Registers TB2, TRGSR, and TABSR



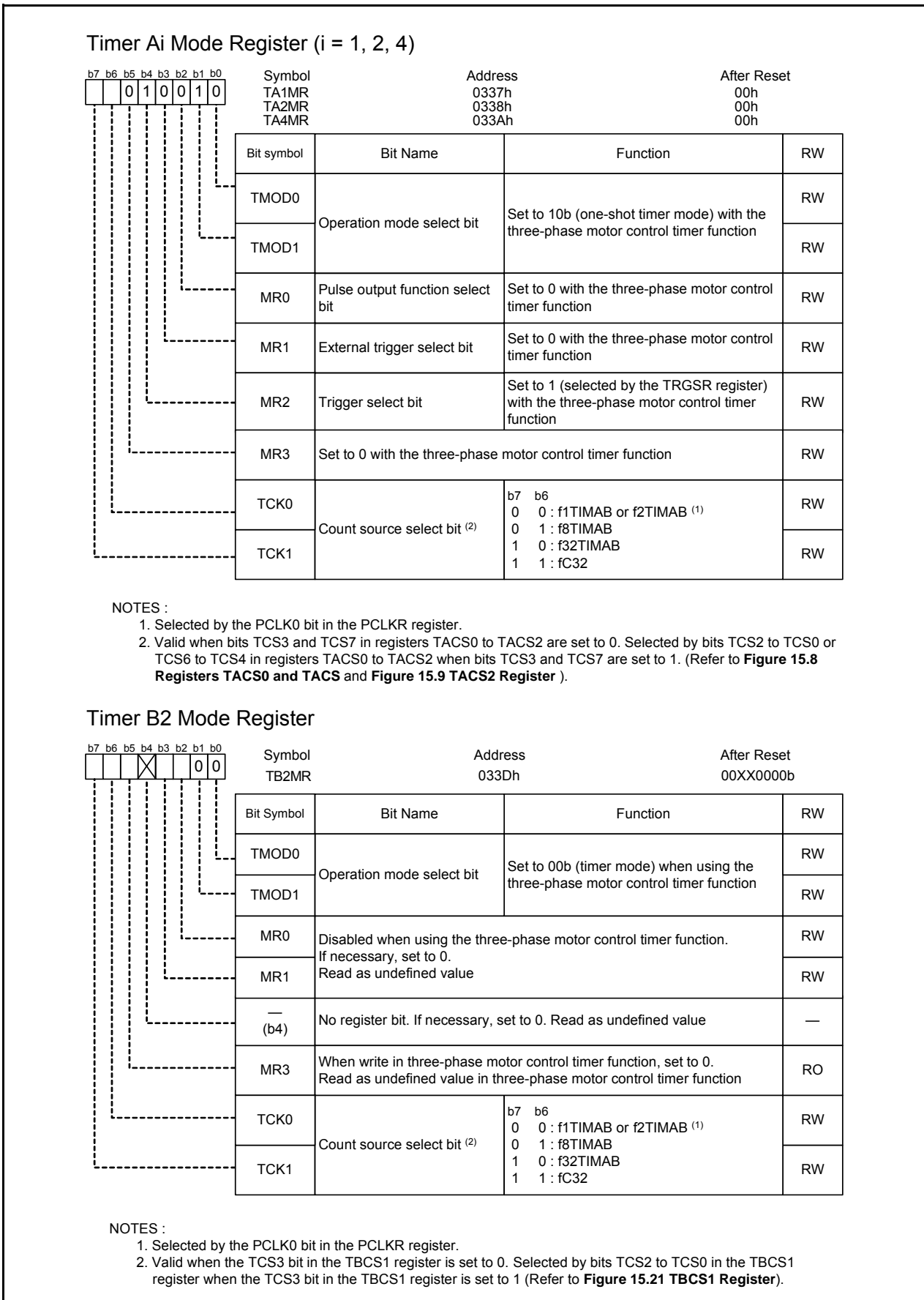


Figure 16.7 Registers TA1MR, TA2MR, TA4MR, and TB2MR

The three-phase motor control timer function is enabled by setting the INV02 bit in the 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,  $\bar{U}$ , V,  $\bar{V}$ , W, and  $\bar{W}$ ). The dead time is controlled by a dedicated dead time timer. Figure 16.8 shows an Example of Triangular Wave Modulation Operation and Figure 16.9 shows an Example of Sawtooth Wave Modulation Operation.

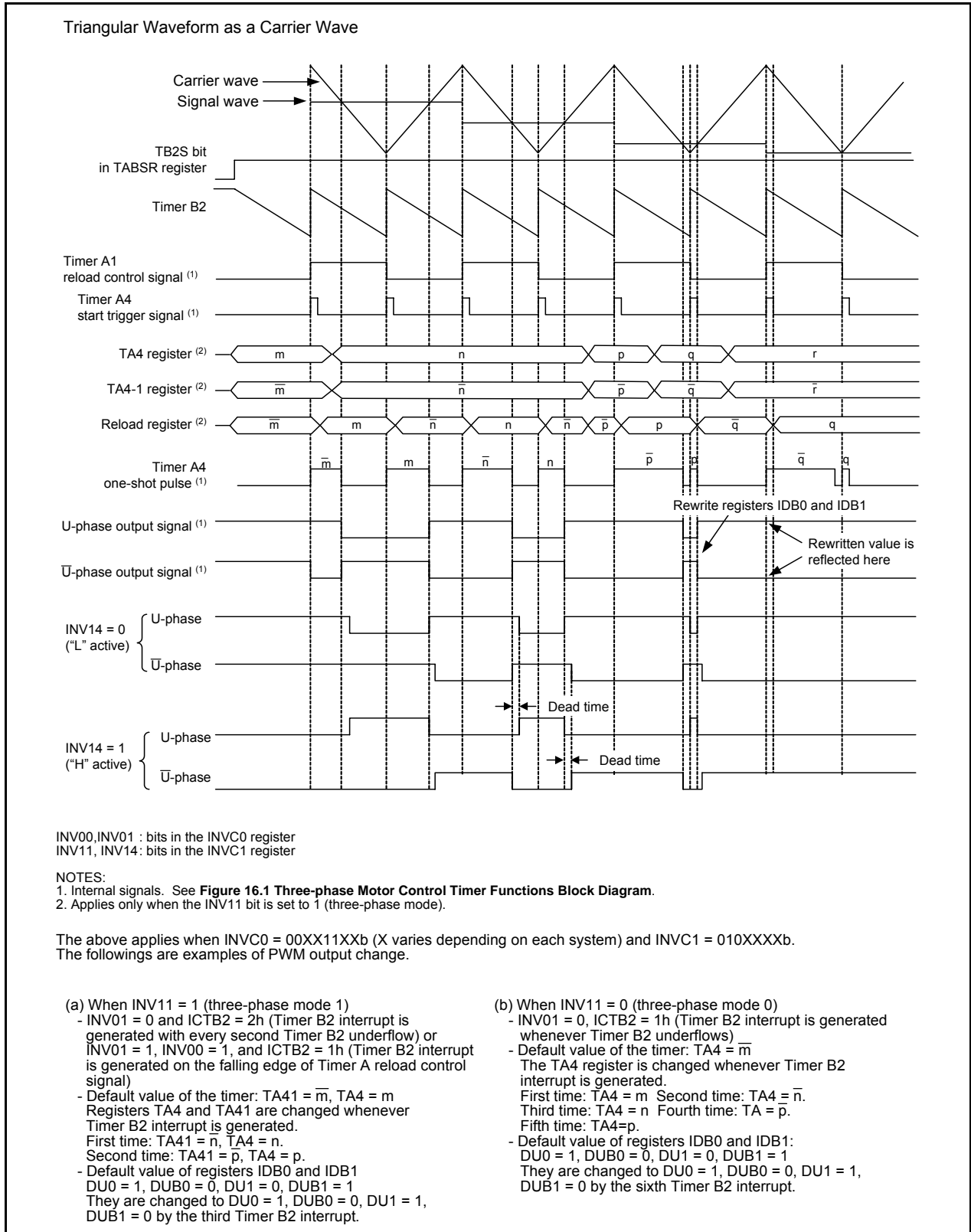


Figure 16.8 Example of Triangular Wave Modulation Operation

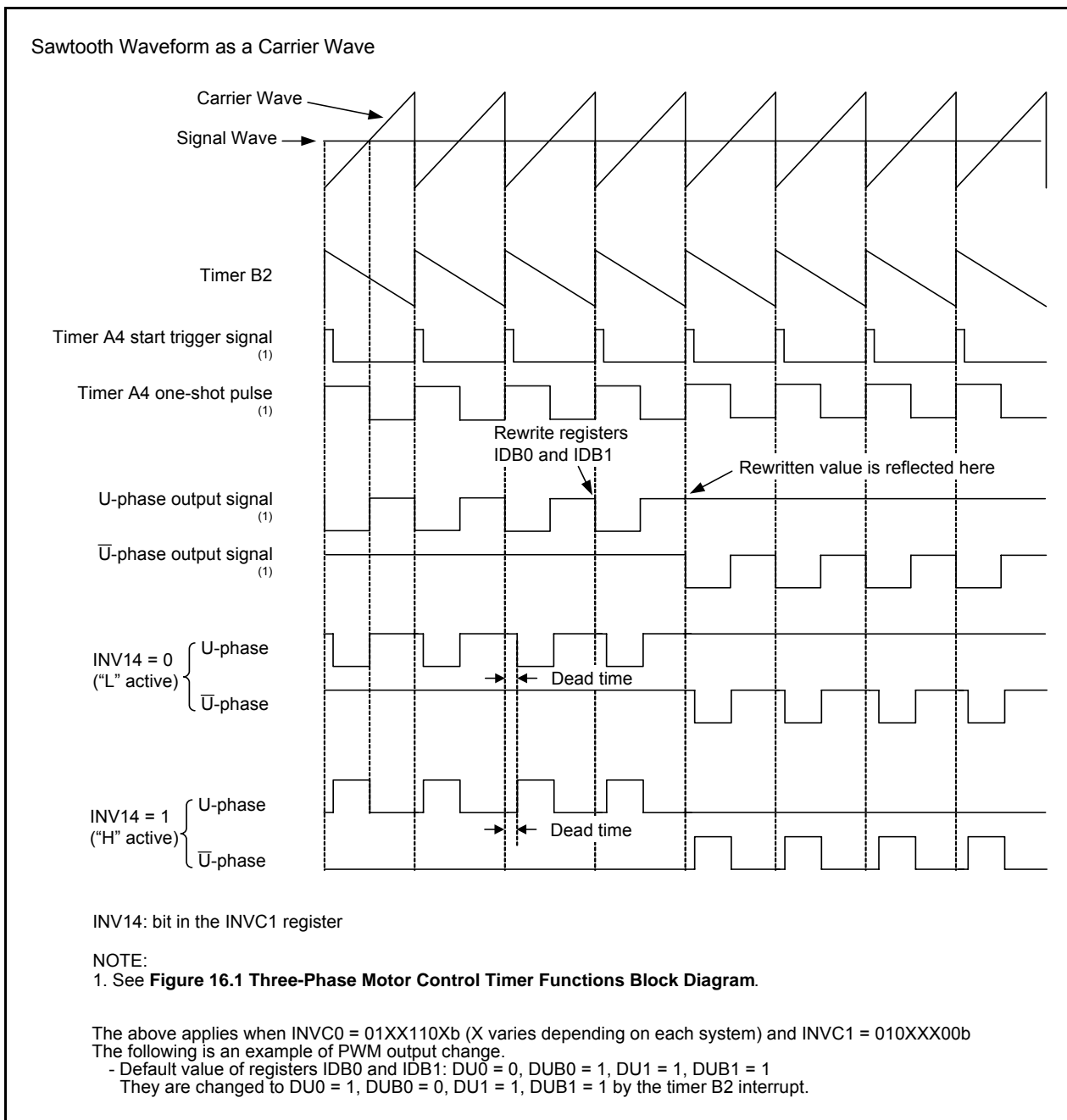


Figure 16.9 Example of Sawtooth Wave Modulation Operation

## 17. Serial Interface

Serial interfaces consist of eight channels: UART0 to UART2, UART5 to UART7, SI/O3, and SI/O4.

### 17.1 UARTi (i = 0 to 2, 5 to 7)

Each UARTi has an exclusive timer to generate a transfer clock, so it operates independently of each other.

Figures 17.1 to 17.3 show the block diagrams of UARTi. Figure 17.4 shows the UARTi Transmit / Receive Unit.

UARTi has the following modes:

- Clock synchronous serial I/O mode
- Clock asynchronous serial I/O mode (UART mode)
- Special mode 1 (I<sup>2</sup>C mode)
- Special mode 2
- Special mode 3 (Bus collision detection function, IE mode)
- Special mode 4 (SIM mode) : UART2

Figures 17.5 to 17.11 show the UARTi-related registers.

Refer to tables for each mode for register setting.

UART6 and UART7 cannot be used in memory expansion mode or microprocessor mode.

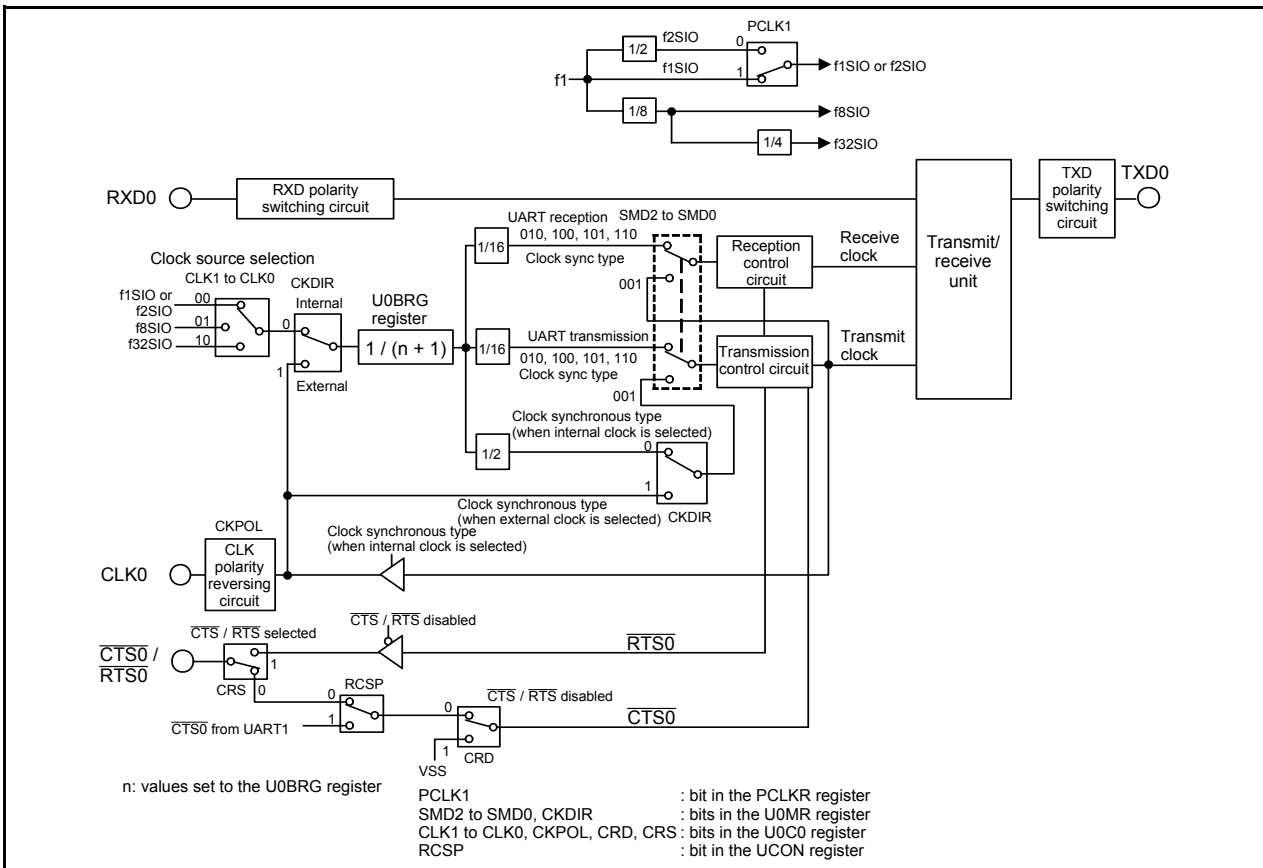


Figure 17.1 UART0 Block Diagram

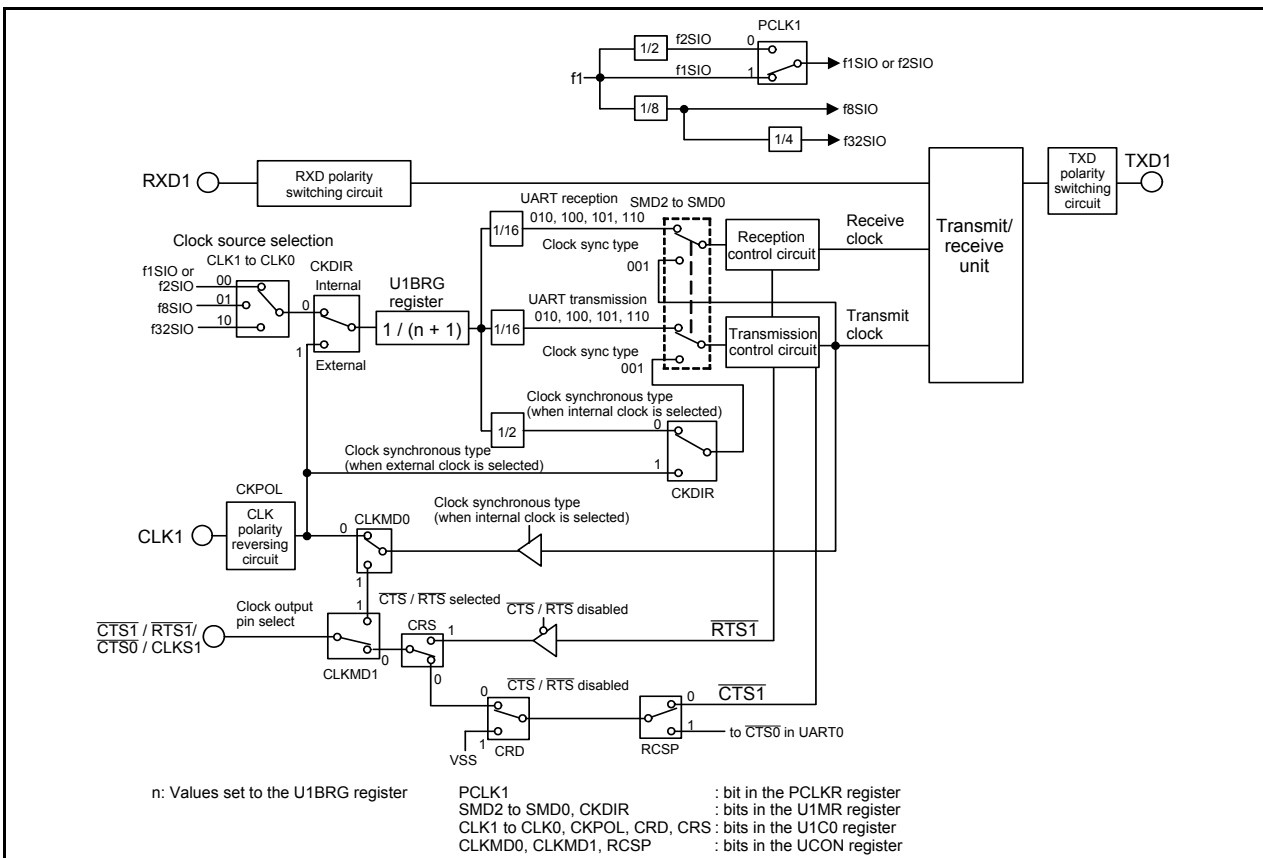


Figure 17.2 UART1 Block Diagram

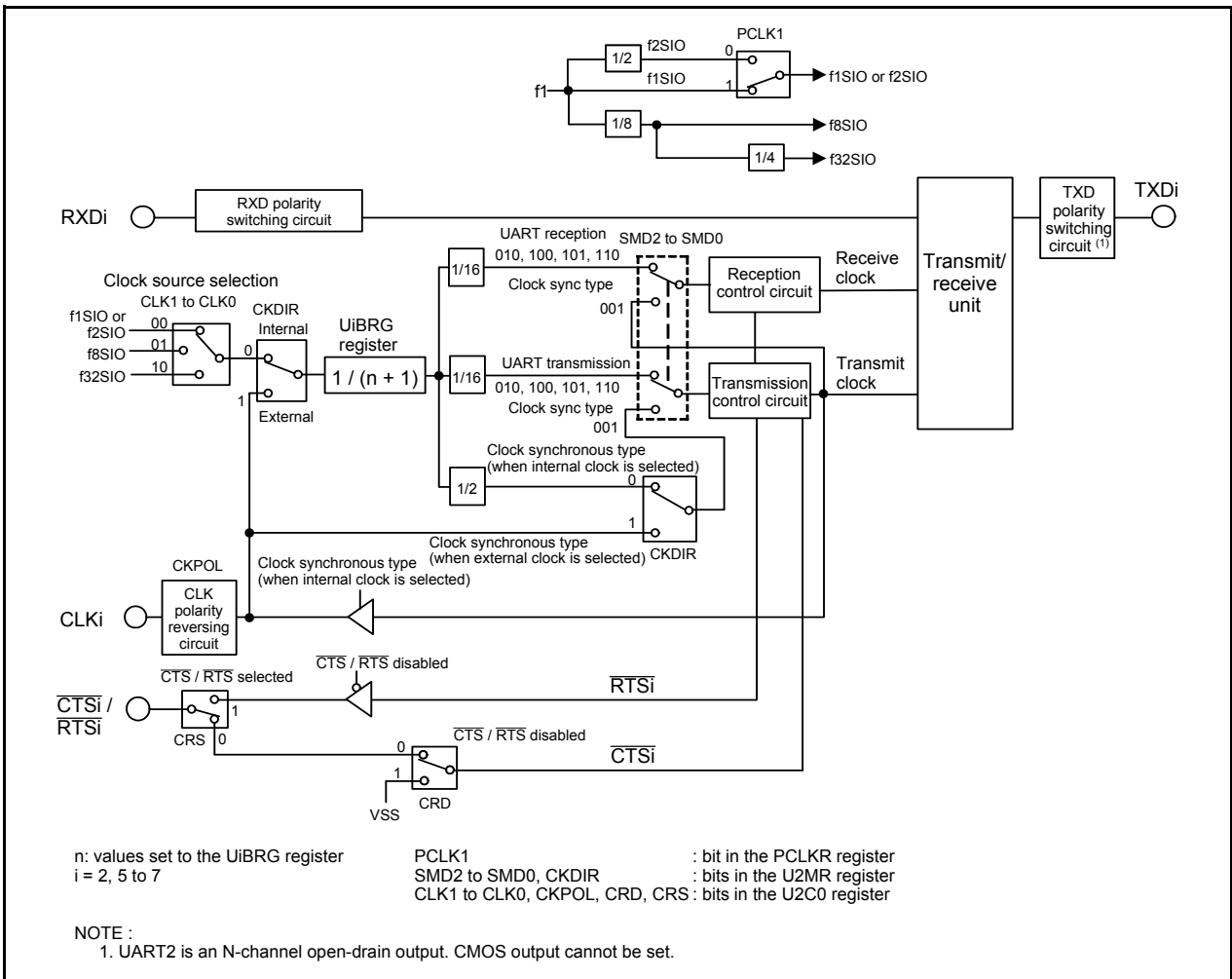


Figure 17.3 UART2, and UART5 to UART7 Block Diagram

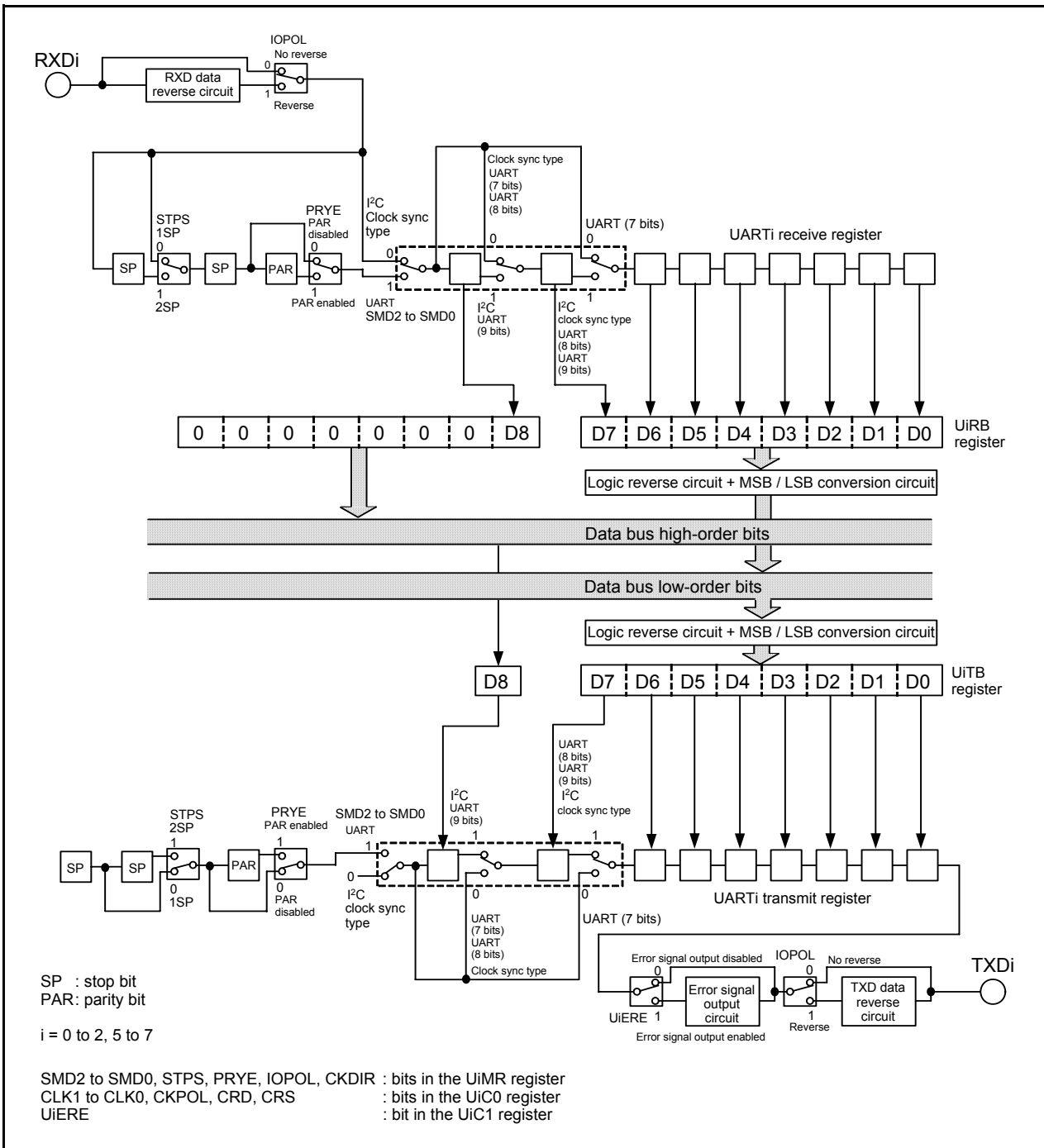


Figure 17.4 UARTi Transmit / Receive Unit

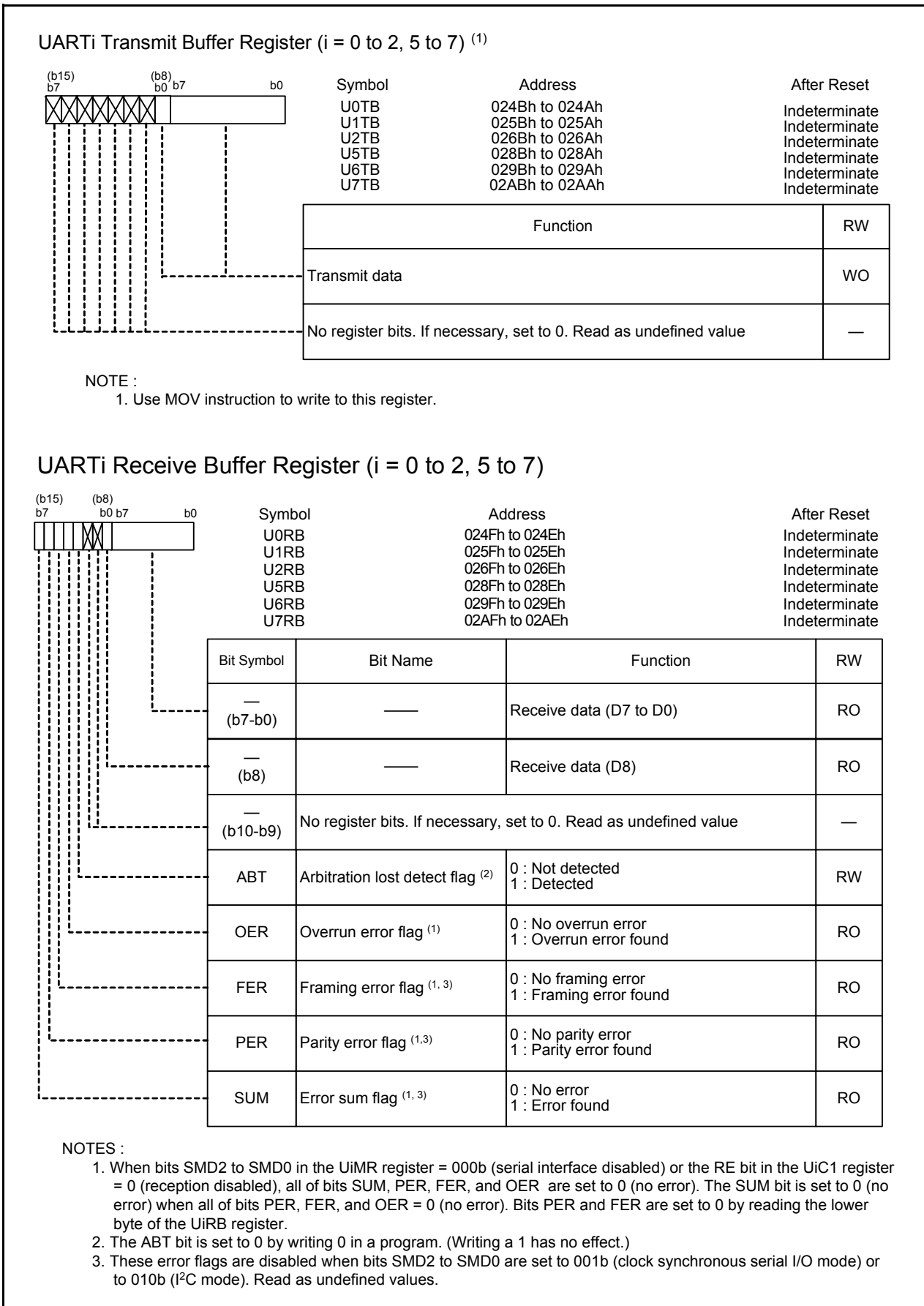


Figure 17.5 Registers U0TB to U2TB, U5TB to U7TB, U0RB to U2RB, and U5RB to U7RB



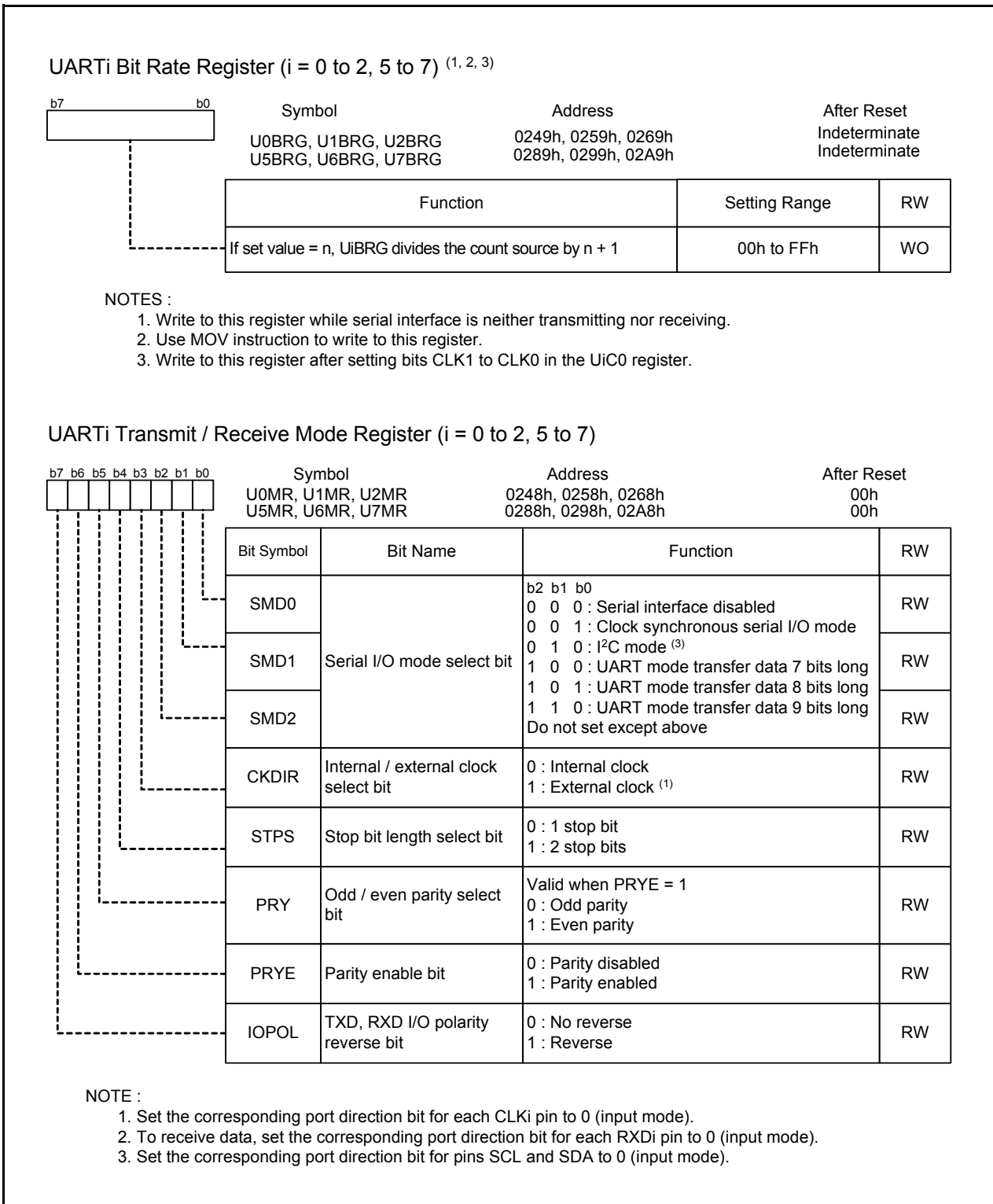


Figure 17.6 Registers U0BRG to U2BRG, U5BRG to U7BRG, U0MR to U2MR, and U5MR to U7MR

## UARTi Transmit / Receive Control Register 0 (i = 0 to 2, 5 to 7)

Bit	Symbol	Address	After Reset
b7	U0C0, U1C0, U2C0 U5C0, U6C0, U7C0	024Ch, 025Ch, 026Ch 028Ch, 029Ch, 02ACh	00001000b 00001000b
b6			
b5			
b4			
b3			
b2			
b1			
b0			

Bit Symbol	Bit Name	Function	RW
CLK0	UiBRG count source select bit <sup>(6)</sup>	b1 b0 0 0 : f1SIO or f2SIO is selected <sup>(5)</sup> 0 1 : f8SIO is selected 1 0 : f32SIO is selected 1 1 : Do not set to this value	RW
CLK1		RW	
CRS	$\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ function select bit <sup>(4)</sup>	Valid when CRD = 0 0 : $\overline{\text{CTS}}$ function selected <sup>(1)</sup> 1 : $\overline{\text{RTS}}$ function selected	RW
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	RO
CRD	$\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ disable bit	0 : $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ function enabled 1 : $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ function disabled (P6_0, P6_4, P7_3, P8_1, P1_0, and P4_4 can be used as I/O ports)	RW
NCH	Data output select bit <sup>(2)</sup>	0 : Pins TXDi / SDAi and SCLi are CMOS output 1 : Pins TXDi / SDAi and SCLi are N-channel open-drain output	RW
CKPOL	CLK polarity select bit	0 : Transmit data is output at the falling edge of transfer clock and receive data is input at the rising edge 1 : Transmit data is output at the rising edge of transfer clock and receive data is input at the falling edge	RW
UFORM	Transfer format select bit <sup>(3)</sup>	0 : LSB first 1 : MSB first	RW

## NOTES :

- Set the corresponding port direction bit for each  $\overline{\text{CTS}}$ i pin to 0 (input mode).
- TXD2 / SDA2 and SCL2 are N-channel open-drain output. Cannot be set to the CMOS output. No NCH bit in the U2C0 register is assigned. If necessary, set to 0.
- The UFORM bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (clock synchronous serial I/O mode), or 101b (UART mode, 8-bit transfer data). Set this bit to 1 when bits SMD2 to SMD0 are set to 010b (I<sup>2</sup>C mode), and to 0 when bits SMD2 to SMD0 are set to 100b (UART mode, 7-bit transfer data) or 110b (UART mode, 9-bit transfer data).
- $\overline{\text{CTS}}$ 1 /  $\overline{\text{RTS}}$ 1 can be used when the CLKMD1 bit in the UCON register = 0 (only CLK1 output) and the RCSP bit in the UCON register = 0 ( $\overline{\text{CTS}}$ 0 /  $\overline{\text{RTS}}$ 0 not separated).
- Selected by the PCLK1 bit in the PCLKR register.
- When changing bits CLK1 and CLK0, set the UiBRG register.

Figure 17.7 Registers U0C0 to U2C0 and U5C0 to U7C0

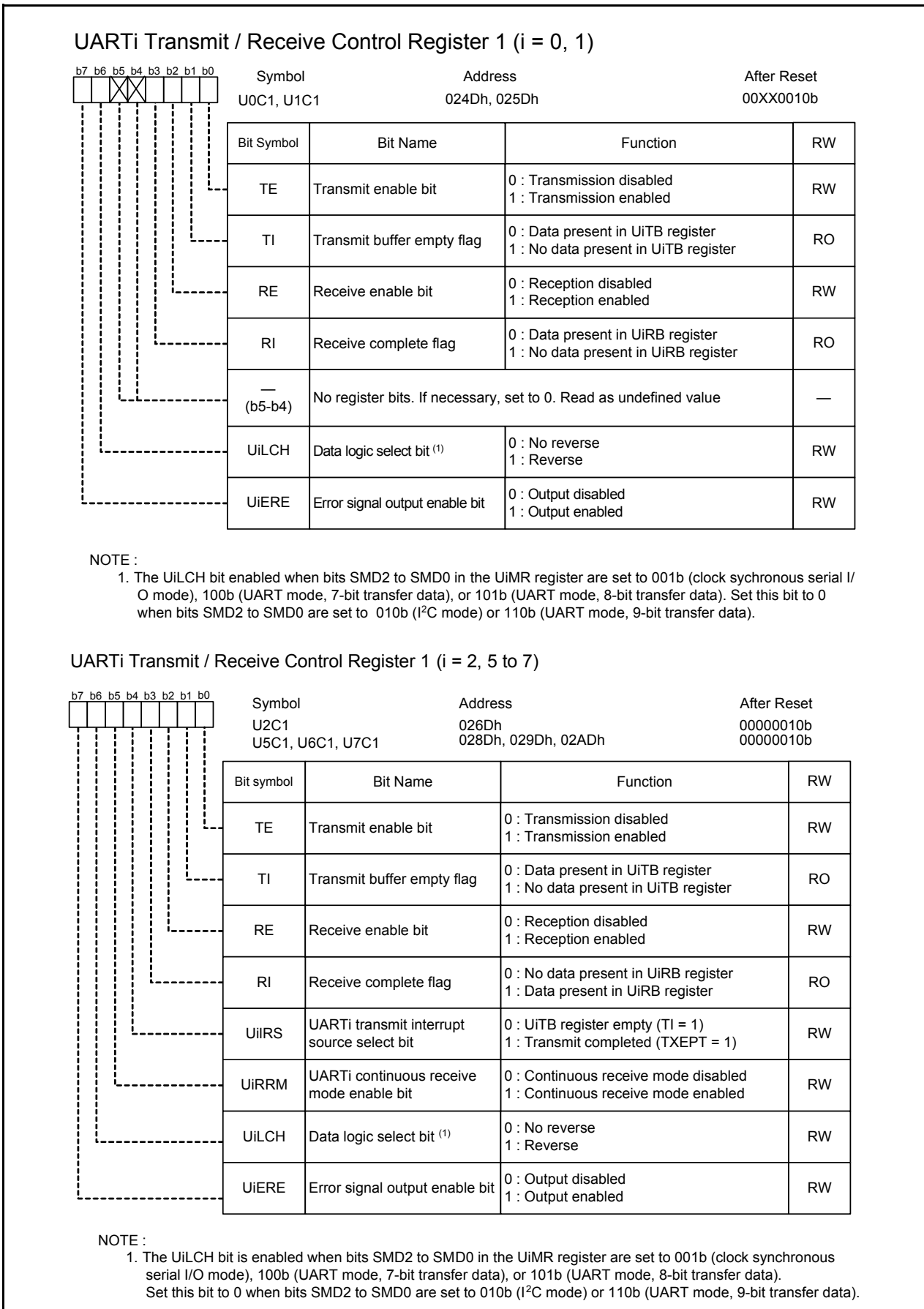
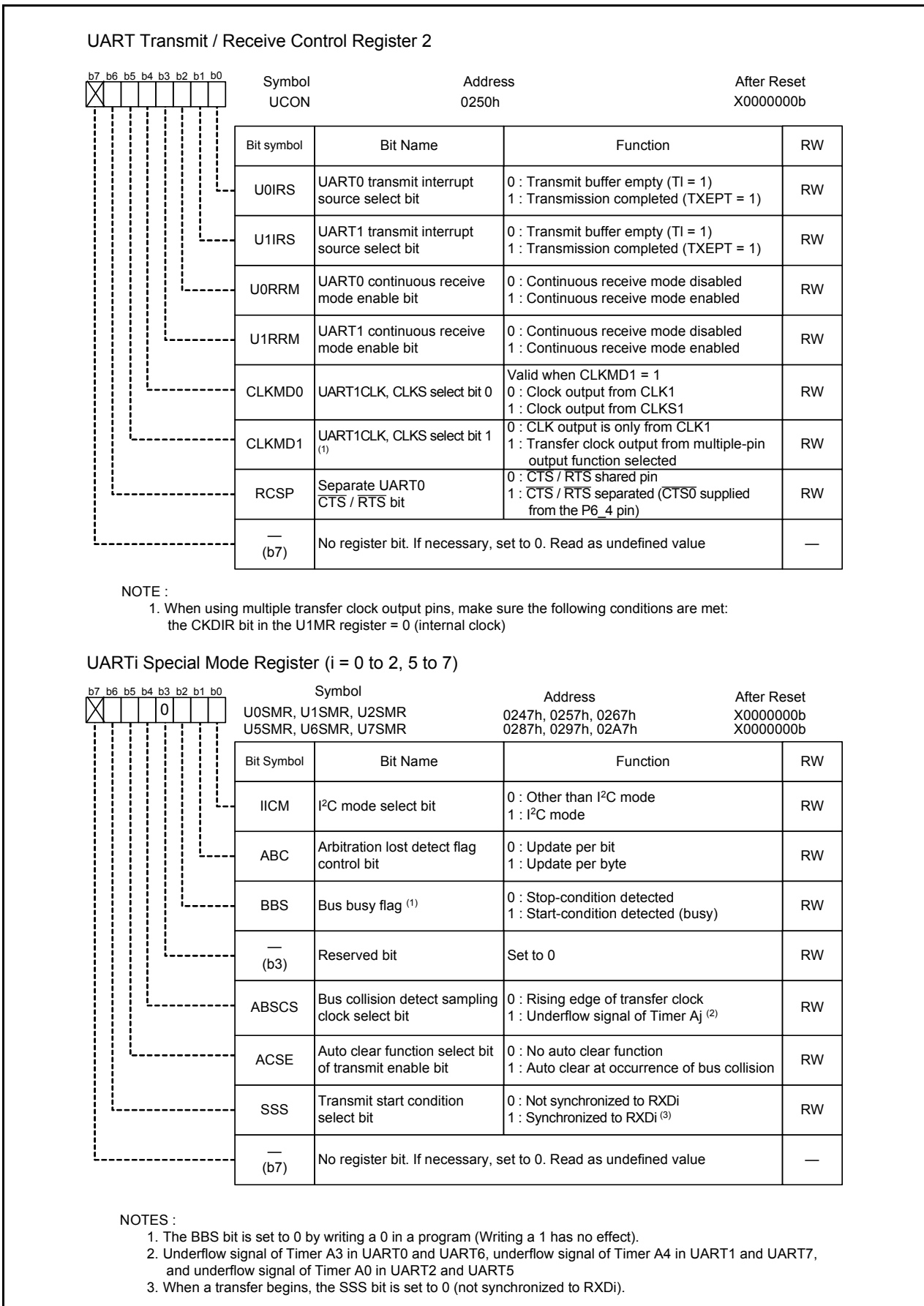
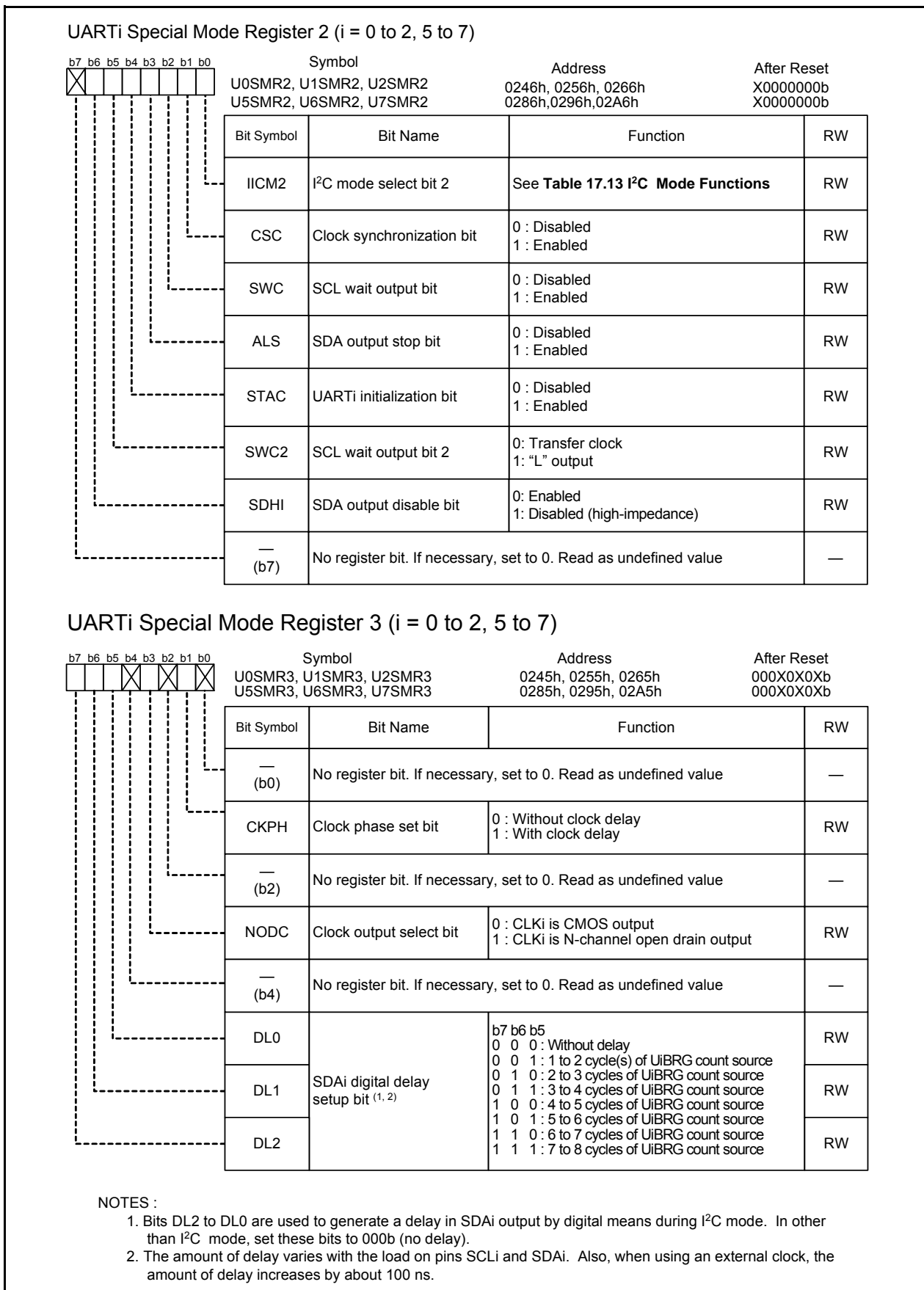


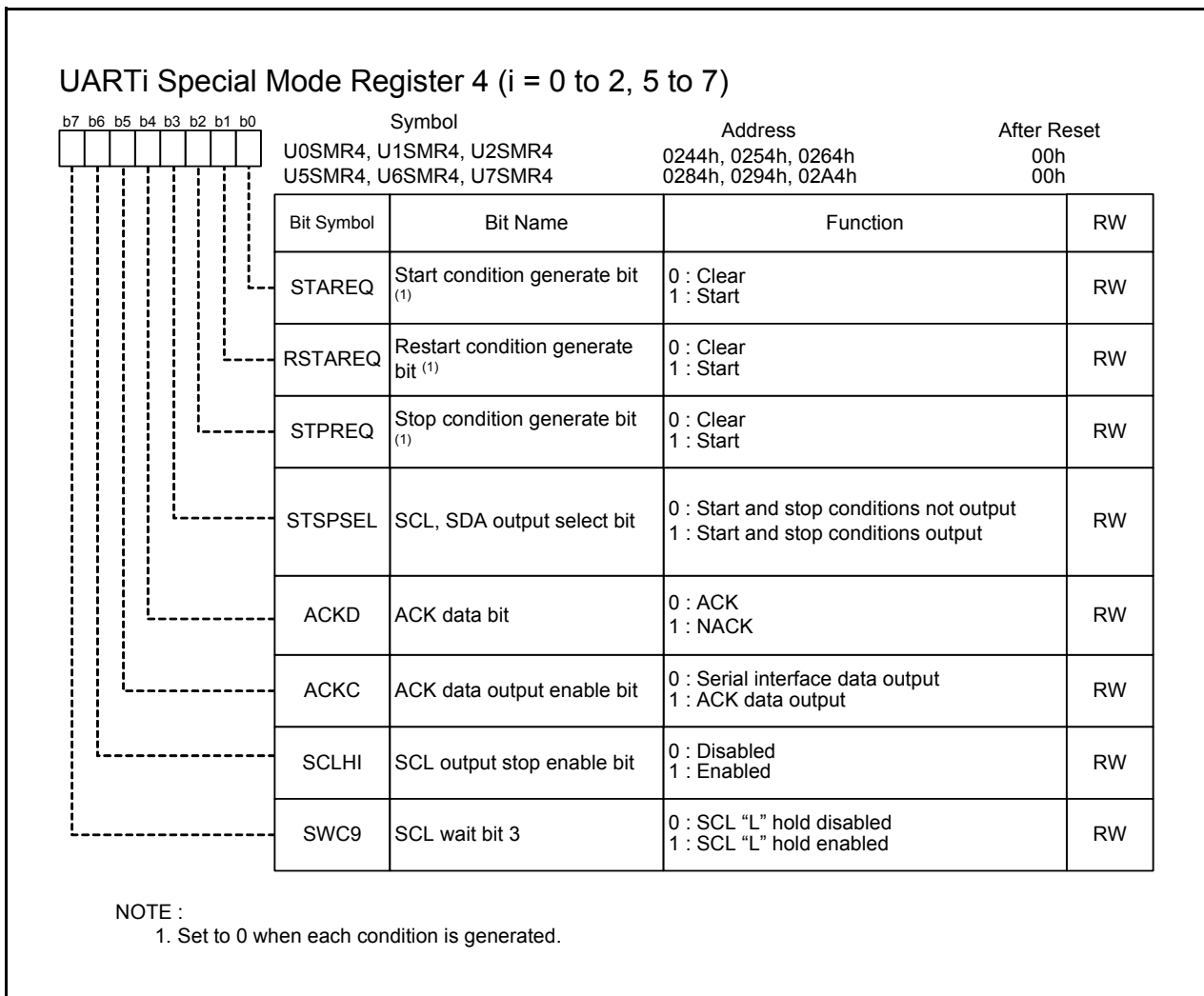
Figure 17.8 Registers U0C1 to U2C1 and U5C1 to U7C1



**Figure 17.9 Registers UCON, U0SMR to U2SMR, and U5SMR to U7SMR**



**Figure 17.10 Registers U0SMR2 to U2SMR2, U5SMR2 to U7SMR2, U0SMR3 to U2SMR3, and U5SMR3 to U7SMR3**



**Figure 17.11 Registers U0SMR4 to U2SMR4 and U5SMR4 to U7SMR4**

### 17.1.1 Clock Synchronous Serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Table 17.1 lists the Clock Synchronous Serial I/O Mode Specifications. Table 17.2 lists Registers to Be Used and Settings in Clock Synchronous Serial I/O Mode.

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

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer Clock	<ul style="list-style-type: none"> <li>CKDIR bit in the UiMR register = 0 (internal clock): <math>f_j / (2(n+1))</math>  <math>f_j = f1SIO, f2SIO, f8SIO, f32SIO</math>    <math>n =</math> setting value of UiBRG register    00h to FFh</li> <li>CKDIR bit = 1 (external clock) : input from CLKi pin</li> </ul>
Transmission, Reception Control	Selectable from CTS function, $\overline{RTS}$ function or CTS / $\overline{RTS}$ function disable
Transmission Start Condition	Before transmission starts, satisfy the following requirements <sup>(1)</sup> <ul style="list-style-type: none"> <li>The TE bit in the UiC1 register = 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register = 0 (data present in UiTB register)</li> <li>If <math>\overline{CTS}</math> function is selected, input on the <math>\overline{CTS}_i</math> pin = "L"</li> </ul>
Reception Start Condition	Before reception starts, satisfy the following requirements <sup>(1)</sup> <ul style="list-style-type: none"> <li>The RE bit in the UiC1 register = 1 (reception enabled)</li> <li>The TE bit in the UiC1 register = 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register = 0 (data present and dummy written in the UiTB register)</li> </ul>
Interrupt Request Generation Timing	For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> <li>The UiIRS bit <sup>(3)</sup> = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission)</li> <li>The UiIRS bit = 1 (transfer completed): when the serial interface finished sending data from the UARTi transmit register</li> </ul> For reception <ul style="list-style-type: none"> <li>When transferring data from the UARTi receive register to the UiRB register (at completion of reception)</li> </ul>
Error Detection	Overrun error <sup>(2)</sup> This error occurs if the serial interface started receiving the next data before reading the UiRB register and received the 7th bit of the next data
Select Function	<ul style="list-style-type: none"> <li>CLK polarity selection Transfer data input / output can be chosen to occur synchronously with the rising or the falling edge of the transfer clock</li> <li>LSB first, MSB first selection Whether to start sending / receiving data beginning with bit 0 or beginning with bit 7 can be selected</li> <li>Continuous receive mode selection Reception is enabled immediately by reading the UiRB register</li> <li>Switching serial data logic This function reverses the logic value of the transmit / receive data</li> <li>Transfer clock output from multiple pins selection (UART1) The output pin can be selected in a program from two UART1 transfer clock pins that have been set</li> <li>Separate <math>\overline{CTS}</math> / <math>\overline{RTS}</math> pins (UART0) <math>\overline{CTS}_0</math> and <math>\overline{RTS}_0</math> are input / output from separate pins</li> </ul>

i = 0 to 2, 5 to 7

**NOTES:**

- When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 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 CKPOL bit in the UiC0 register = 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.
- If an overrun error occurs, the receive data of the UiRB register will be indeterminate. The IR bit in the SiRIC register does not change to 1 (interrupt requested).
- Bits U0IRS and U1IRS correspond to bits 0 and 1 in the UCON register respectively. Bits U2IRS,

U5IRS, U6IRS, and U7IRS are in registers U2C1, U5C1, U6C1, and U7C1 respectively.

**Table 17.2 Registers to Be Used and Settings in Clock Synchronous Serial I/O Mode**

Register	Bit	Function
UiTB (3)	0 to 7	Set transmission data
UiRB (3)	0 to 7	Reception data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a bit rate
UiMR (3)	SMD2 to SMD0	Set to 001b
	CKDIR	Select the internal clock or external clock
	IOPOL	Set to 0
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Select either CTS or RTS to use functions
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ function
	NCH	Select TXDi pin output mode (2)
	CKPOL	Select the transfer clock polarity
	UFORM	Select the LSB first or MSB first
UiC1	TE	Set this bit to 1 to enable transmission / reception
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	UiIRS (1)	Select the source of UARTi transmit interrupt
	UiRRM (1)	Set this bit to 1 to use continuous reception mode
	UiLCH	Set this bit to 1 to use inverted data logic
	UiERE	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 reception 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 $\overline{\text{CTS0}}$ signal of UART0 from the P6_4 pin
	7	Set to 0

i = 0 to 2, 5 to 7

NOTES:

1. Set bits 4 and 5 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.
2. The TXD2 pin is N channel open-drain output. Set the NCH bit in the U2C0 register to 0.
3. Set bits not listed above to 0 when writing to the registers in clock synchronous serial I/O mode.



Table 17.3 lists the functions of the input / output pins during clock synchronous serial I/O mode. Table 17.3 shows pin functions for the case where the multiple transfer clock output pin select function is not selected. Table 17.4 lists the P6\_4 Pin Functions during clock synchronous serial I/O mode.

Note that for a period from when UARTi operating mode is selected to when transfer starts, the TXDi pin outputs “H” (If the N-channel open-drain output is selected, this pin is in high-impedance state).

**Table 17.3 Pin Functions during Clock Synchronous Serial I/O Mode (Multiple Transfer Clock Output Pin Function Not Selected)**

Pin Name	Function	Method of Selection
TXDi	Serial data output	(Outputs dummy data when performing reception only)
RXDi	Serial data input	Set the port direction bit corresponding to the RXDi pin = 0 (can be used as an input port when performing transmission only)
CLKi	Transfer clock output	The CKDIR bit in the UiMR register = 0
	Transfer clock input	The CKDIR bit in the UiMR register = 1 Set the port direction bit corresponding to the CLKi pin = 0
$\overline{\text{CTS}}_i / \overline{\text{RTS}}_i$	$\overline{\text{CTS}}$ input	The CRD bit in the UiC0 register = 0 The CRS bit in the UiC0 register = 0 Set the port direction bit corresponding to the $\overline{\text{CTS}}_i$ pin = 0
	$\overline{\text{RTS}}$ output	The CRD bit in the UiC0 register = 0 The CRS bit in the UiC0 register = 1
	I/O port	The CRD bit in the UiC0 register = 1

i = 0 to 2, 5 to 7

**Table 17.4 P6\_4 Pin Functions during Clock Synchronous Serial I/O Mode**

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	-	-
$\overline{\text{CTS}}_0$ (1)	0	0	1	0	-	0
CLKS1	-	-	-	1 (2)	1	-

- indicates either 0 or 1

NOTES:

- In addition to this, set the CRD bit in the U0C0 register to 0 ( $\overline{\text{CTS}}_0 / \overline{\text{RTS}}_0$  enabled) and the CRS bit in the U0C0 register to 1 ( $\overline{\text{RTS}}_0$  selected).
- When the CLKMD1 bit = 1 and the CLKMD0 bit = 0, the following logic levels are output:
  - High if the CLKPOL bit in the U1C0 register = 0
  - Low if the CLKPOL bit in the U1C0 register = 1

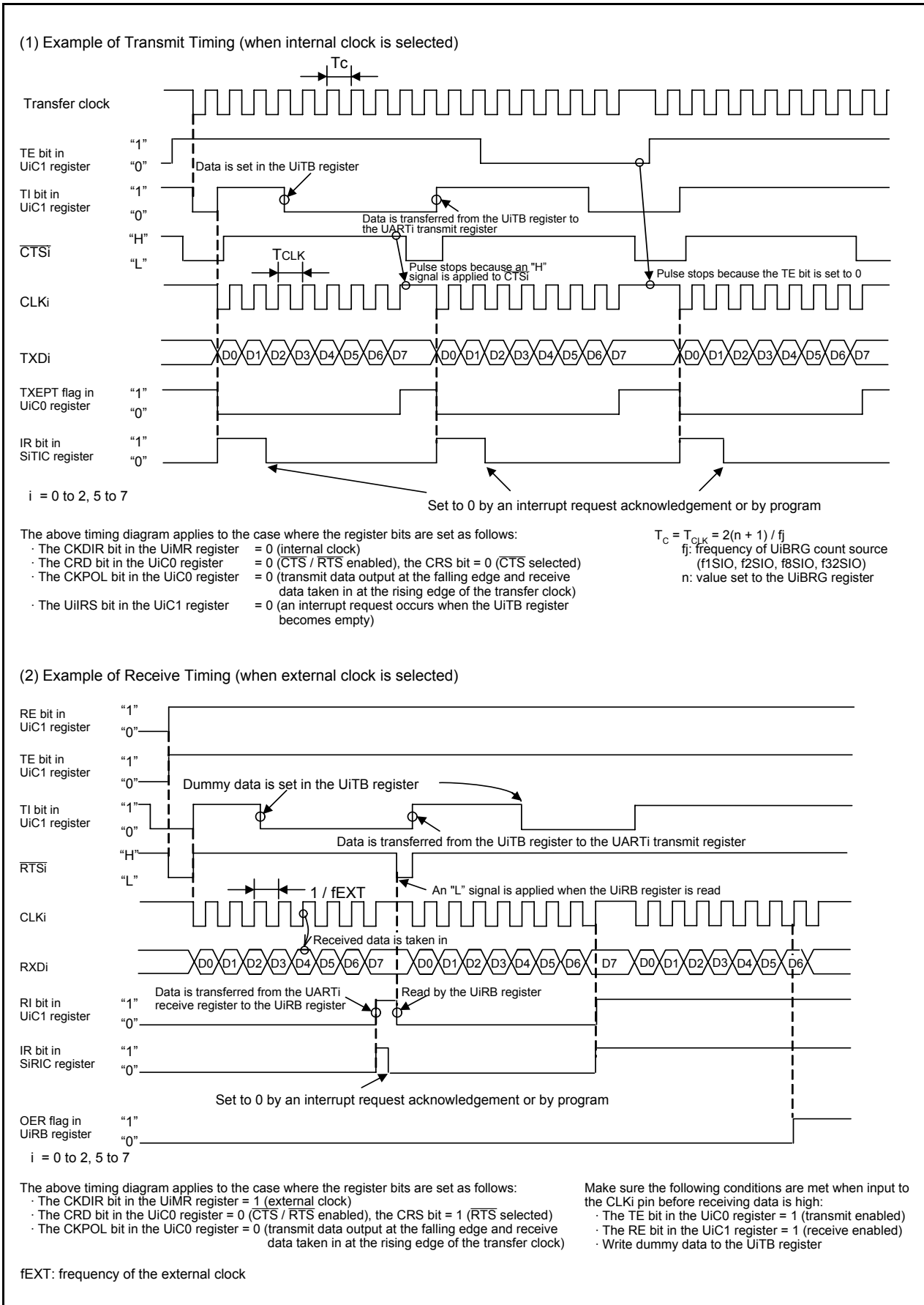


Figure 17.12 Transmit and Receive Operation during Clock Synchronous Serial I/O Mode

### 17.1.1.1 Counter Measure for Communication Error

If a communication error occurs while transmitting or receiving in clock synchronous serial I/O mode, follow the procedures below.

- Resetting the UiRB register (i = 0 to 2, 5 to 7)
  - (1) Set the RE bit in the UiC1 register to 0 (reception disabled)
  - (2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled)
  - (3) Set bits SMD2 to SMD0 in the UiMR register to 001b (clock synchronous serial I/O mode)
  - (4) Set the RE bit in the UiC1 register to 1 (reception enabled)
  
- Resetting the UiTB register (i = 0 to 2, 5 to 7)
  - (1) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled)
  - (2) Set bits SMD2 to SMD0 in the UiMR register to 001b (clock synchronous serial I/O mode)
  - (3) A 1 is written to the RE bit in the UiC1 register (transmission enabled), regardless of the value of the TE bit in the UiCi register

### 17.1.1.2 CLK Polarity Select Function

Use the CKPOL bit in the UiC0 register (i = 0 to 2, 5 to 7) to select the transfer clock polarity. Figure 17.13 shows the Transfer Clock Polarity.

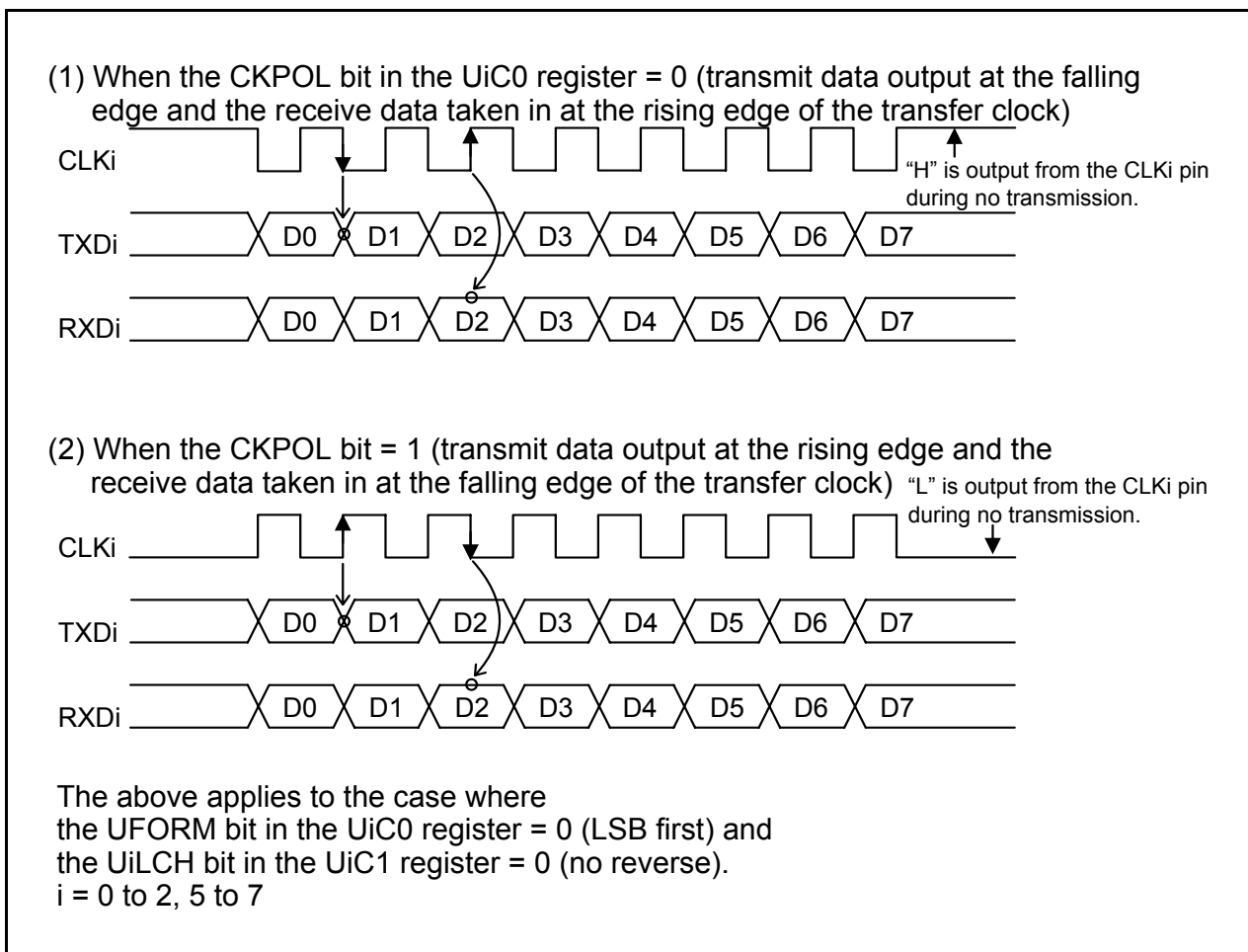


Figure 17.13 Transfer Clock Polarity

### 17.1.1.3 LSB First / MSB First Select Function

Use the UFORM bit in the UiC0 register (i = 0 to 2, 5 to 7) to select the transfer format. Figure 17.14 shows the Transfer Format.

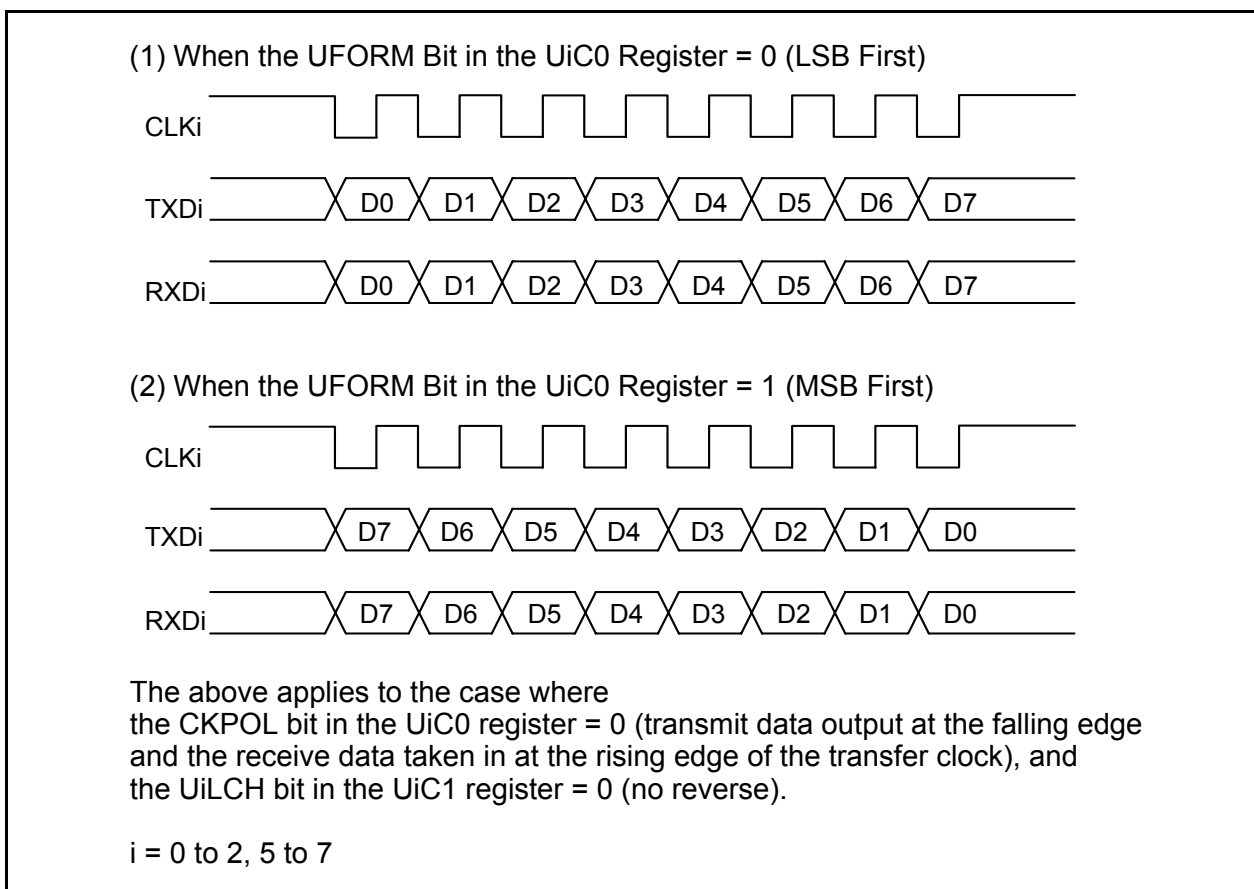


Figure 17.14 Transfer Format

### 17.1.1.4 Continuous Reception Mode

In continuous reception mode, receive operation becomes enabled when the receive buffer register is read. It is not necessary to write dummy data into the transmit buffer register to enable receive operation in this mode. However, a dummy read of the receive buffer register is required when starting the operating mode.

When the UiRRM bit (i = 0 to 2, 5 to 7) = 1 (continuous reception mode), the TI bit in the UiC1 register 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. Bits U0RRM and U1RRM correspond to bits 2 and 3 in the UCON register, respectively. Bits U2RRM, U5RRM, U6RRM, and U7RRM are in registers U2C1, U5C1, U6C1, and U7C1.

### 17.1.1.5 Serial Data Logic Switching Function

When the UiLCH bit in the UiC1 register ( $i = 0$  to  $2, 5$  to  $7$ ) = 1 (reverse), the data written to the UiTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the UiRB register. Figure 17.15 shows Serial Data Logic Switching.

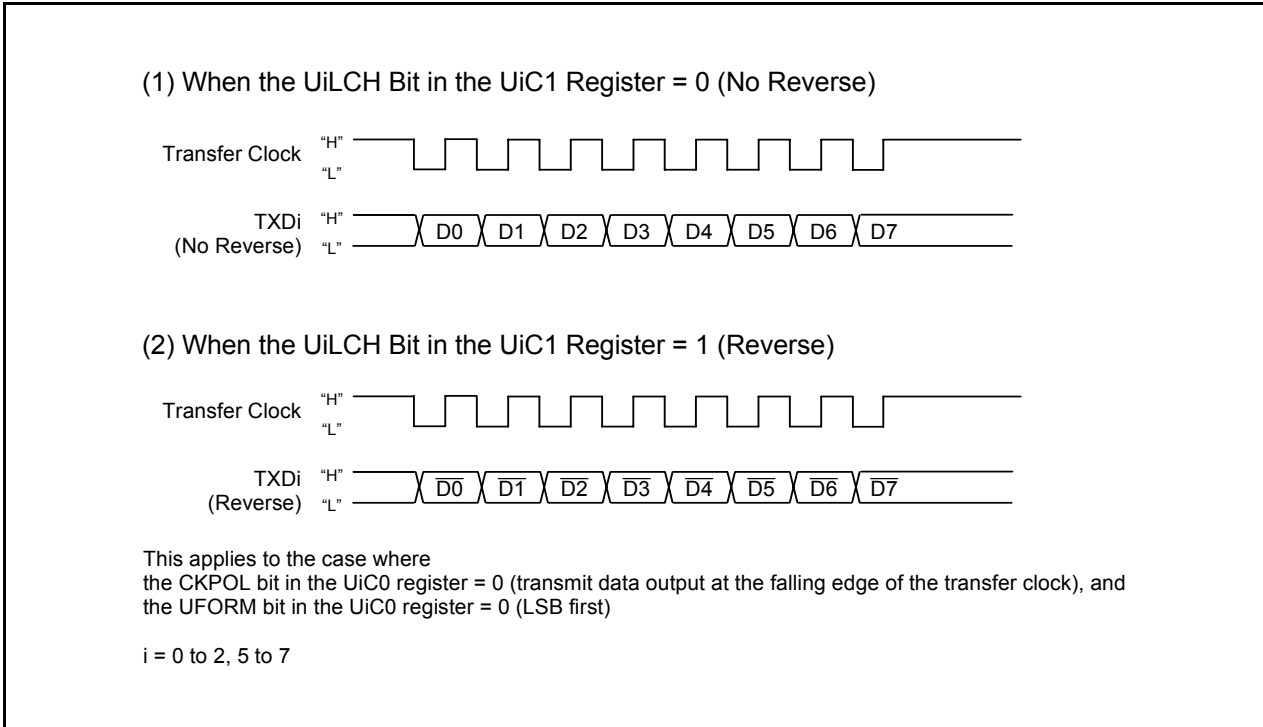


Figure 17.15 Serial Data Logic Switching

### 17.1.1.6 Transfer Clock Output from Multiple Pins (UART1)

Use bits CLKMD1 to CLKMD0 in the UCON register to select one of the two transfer clock output pins (see Figure 17.16). This function can be used when the selected transfer clock for UART1 is an internal clock.

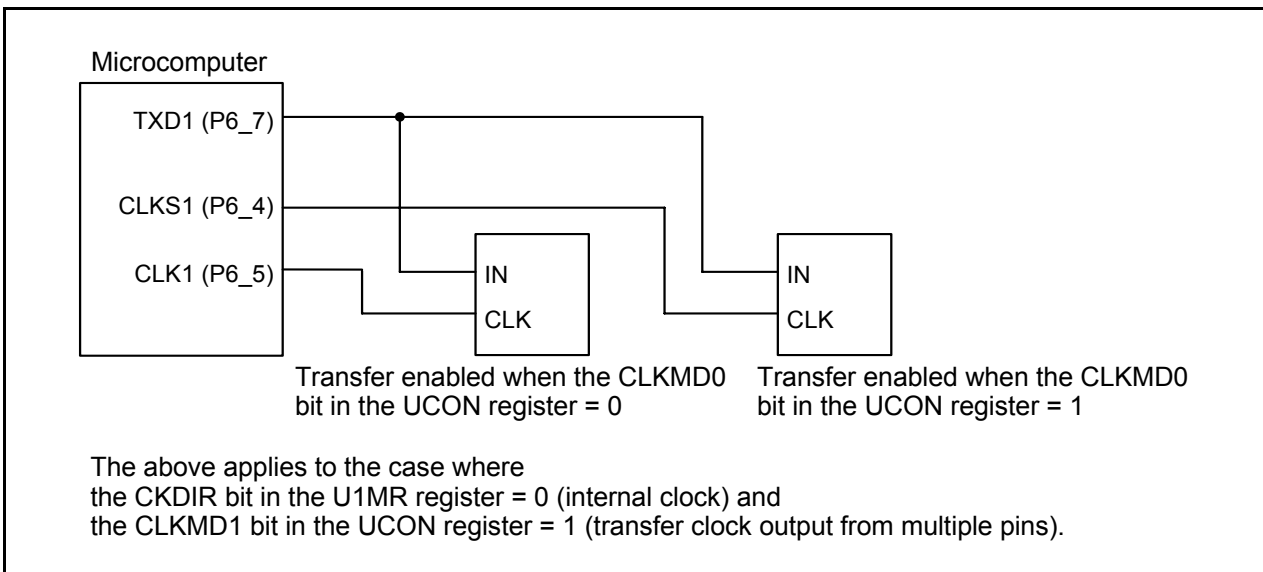


Figure 17.16 Transfer Clock Output from Multiple Pins

### 17.1.1.7 $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ Function

The  $\overline{\text{CTS}}$  function is used to start transmit and receive operation when “L” is applied to the  $\overline{\text{CTS}}_i$  /  $\overline{\text{RTS}}_i$  ( $i = 0$  to 2, 5 to 7) pin. Transmit and receive operation begins when the  $\overline{\text{CTS}}_i$  /  $\overline{\text{RTS}}_i$  pin is held “L”. If the “L” signal is switched to “H” during a transmit or receive operation, the operation stops before the next data.

For the  $\overline{\text{RTS}}$  function, the  $\overline{\text{CTS}}_i$  /  $\overline{\text{RTS}}_i$  pin outputs “L” when the microcomputer is ready to receive. The output level becomes “H” on the first falling edge of the CLK $_i$  pin.

- The CRD bit in the UIC0 register = 1 (disable  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  function)  
 $\overline{\text{CTS}}_i$  /  $\overline{\text{RTS}}_i$  pin is programmable I/O function
- The CRD bit = 0, CRS bit = 0 ( $\overline{\text{CTS}}$  function selected)  $\overline{\text{CTS}}_i$  /  $\overline{\text{RTS}}_i$  pin is  $\overline{\text{CTS}}$  function
- The CRD bit = 0, CRS bit = 1 ( $\overline{\text{RTS}}$  function selected)  $\overline{\text{CTS}}_i$  /  $\overline{\text{RTS}}_i$  pin is  $\overline{\text{RTS}}$  function

### 17.1.1.8 $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ 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 inputs  $\overline{\text{CTS}}_0$  from the P6\_4 pin. To use this function, set the register bits as shown below.

- The CRD bit in the UOC0 register = 0 (enable  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  of UART0)
- The CRS bit in the UOC0 register = 1 (output  $\overline{\text{RTS}}$  of UART0)
- The CRD bit in the U1C0 register = 0 (enable  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  of UART1)
- The CRS bit in the U1C0 register = 0 (input  $\overline{\text{CTS}}$  of UART1)
- The RCSP bit in the UCON register = 1 (inputs  $\overline{\text{CTS}}_0$  from the P6\_4 pin)
- The CLKMD1 bit in the UCON register = 0 (CLKS1 not used)

Note that when using the  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  separate function,  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  of UART1 function cannot be used.

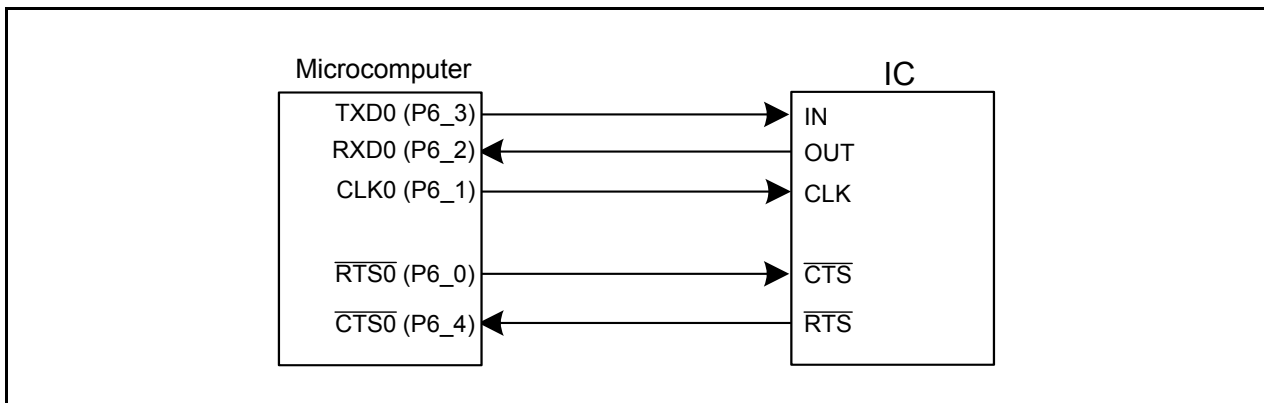


Figure 17.17  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  Separate Function

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

The UART mode allows transmitting and receiving data after setting the desired bit rate and transfer data format. Table 17.5 lists the UART Mode Specifications.

**Table 17.5 UART Mode Specifications**

Item	Specification
Transfer Data Format	<ul style="list-style-type: none"> <li>Character bit (transfer data): selectable from 7, 8, or 9 bits</li> <li>Start bit : 1 bit</li> <li>Parity bit : selectable from odd, even, or none</li> <li>Stop bit : selectable from 1 bit or 2 bits</li> </ul>
Transfer Clock	<ul style="list-style-type: none"> <li>The CKDIR bit in the UiMR register = 0 (internal clock): <math>f_j / (16(n + 1))</math>  <math>f_j = f1SIO, f2SIO, f8SIO, f32SIO</math> n: setting value of UiBRG register 00h to FFh</li> <li>CKDIR bit = 1 (external clock): <math>fEXT / (16(n + 1))</math>                      fEXT: input from CLKi pin n: setting value of UiBRG register 00h to FFh</li> </ul>
Transmission, Reception Control	Selectable from CTS function, RTS function or CTS / RTS function disabled
Transmission Start Condition	Before transmission starts, satisfy the following requirements <ul style="list-style-type: none"> <li>The TE bit in the UiC1 register = 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register = 0 (data present in the UiTB register)</li> <li>If CTS function is selected, input on the CTSi pin = "L"</li> </ul>
Reception Start Condition	Before reception starts, satisfy the following requirements <ul style="list-style-type: none"> <li>The RE bit in the UiC1 register = 1 (reception enabled)</li> <li>Start bit detection</li> </ul>
Interrupt Request Generation Timing	For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> <li>The UiIRS bit (2) = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission)</li> <li>The UiIRS bit = 1 (transfer completed): when the serial interface completes sending data from the UARTi transmit register</li> </ul> For reception <ul style="list-style-type: none"> <li>When transferring data from the UARTi receive register to the UiRB register (at completion of reception)</li> </ul>
Error Detection	<ul style="list-style-type: none"> <li>Overrun error (1) This error occurs if the serial interface started receiving the next data before reading the UiRB register and received the bit one before the last stop bit of the next data</li> <li>Framing error (3) This error occurs when the number of stop bits set is not detected</li> <li>Parity error (3) This error occurs when if parity is enabled, the number of 1 in parity and character bits does not match the number of 1 set</li> <li>Error sum flag This flag is set to 1 when any of the overrun, framing, or parity errors occur</li> </ul>
Select Function	<ul style="list-style-type: none"> <li>LSB first, MSB first selection Whether to start sending / receiving data beginning with bit 0 or beginning with bit 7 can be selected</li> <li>Serial data logic switch This function reverses the logic of the transmit / receive data. The start and stop bits are not reversed.</li> <li>TXD, RXD I/O polarity switch This function reverses the polarities of the TXD pin output and RXD pin input. The logic levels of all I/O data are reversed.</li> <li>Separate CTS / RTS pins (UART0) CTS0 and RTS0 are input / output from separate pins.</li> </ul>

i = 0 to 2, 5 to 7

**NOTES:**

- If an overrun error occurs, the receive data of the UiRB register will be indeterminate. The IR bit in the SiRIC register does not change.
- Bits U0IRS and U1IRS are bits 0 and 1 in the UCON register. Bits U2IRS, U5IRS, U6IRS, and U7IRS are in registers U2C1, U5C1, U6C1, and U7C1.
- The timing at which the framing error flag and the parity error flag are set is detected when data is transferred from the UARTi receive register to the UiRB register.

**Table 17.6 Registers to Be Used and Settings in UART Mode**

Register	Bit	Function
UiTB	0 to 8	Set transmission data <sup>(1)</sup>
UiRB	0 to 8	Reception data can be read <sup>(1)</sup>
	OER, FER, PER, SUM	Error flag
UiBRG	0 to 7	Set a bit rate
UiMR	SMD2 to SMD0	Set these bits to 100b when transfer data is 7 bits long
		Set these bits to 101b when transfer data is 8 bits long
		Set these bits to 110b when transfer data is 9 bits long
	CKDIR	Select the internal clock or external clock
	STPS	Select the stop bit
	PRY, PRYE	Select whether parity is included and whether odd or even
	IOPOL	Select the TXD / RXD input / output polarity
UiC0	CLK0, CLK1	Select the count source for the UiBRG register
	CRS	Select CTS or RTS to use functions
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the CTS or RTS function
	NCH	Select TXDi pin output mode <sup>(3)</sup>
	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
	UiIRS <sup>(2)</sup>	Select the source of UARTi transmit interrupt
	UiRRM <sup>(2)</sup>	Set to 0
	UiLCH	Set this bit to 1 to use reversed data logic
	UiERE	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 $\overline{\text{CTS0}}$ signal of UART0 from the P6_4 pin
	7	Set to 0

i = 0 to 2, 5 to 7

NOTES:

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.
2. Set the bit 4 and bit 5 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are included in the UCON register.
3. TXD2 pin is N channel open-drain output. Set the NCH bit in the U2C0 register to 0.



Table 17.7 lists the functions of the input / output pins during UART mode. Table 17.8 lists the P6\_4 Pin Functions in UART Mode. Note that for a period from when the UARTi operating mode is selected to when transfer starts, the TXDi pin outputs “H” (If the N-channel open-drain output is selected, this pin is in high-impedance state).

**Table 17.7 I/O Pin Functions in UART Mode**

Pin Name	Function	Method of Selection
TXDi	Serial data output	(“H” output when performing reception only)
RXDi	Serial data input	Set the port direction bit corresponding to the RXDi pin to 0 (can be used as an input port when performing transmission only)
CLKi	Input / output port	The CKDIR bit in the UiMR register = 0
	Transfer clock input	The CKDIR bit in the UiMR register = 1 Set the port direction bit corresponding to the CLKi pin to 0
CTS $\bar{i}$ / RTS $\bar{i}$	CTS input	The CRD bit in the UiC0 register = 0 The CRS bit in the UiC0 register = 0 Set the port direction bit corresponding to the CTS $\bar{i}$ pin to 0
	RTS input	The CRD bit in the UiC0 register = 0 The CRS bit in the UiC0 register = 1
	Input / output port	The CRD bit in the UiC0 register = 1

i = 0 to 2 , 5 to 7

**Table 17.8 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
CTS $\bar{1}$	0	0	0	0	0
RTS $\bar{1}$	0	1	0	0	-
CTS $\bar{0}$ (1)	0	0	1	0	0

- indicates either 0 or 1.

**NOTE:**

1. In addition to this, set the CRD bit in the U0C0 register to 0 (CTS $\bar{0}$  / RTS $\bar{0}$  enabled) and the CRS bit in the U0C0 register to 1 (RTS $\bar{0}$  selected).

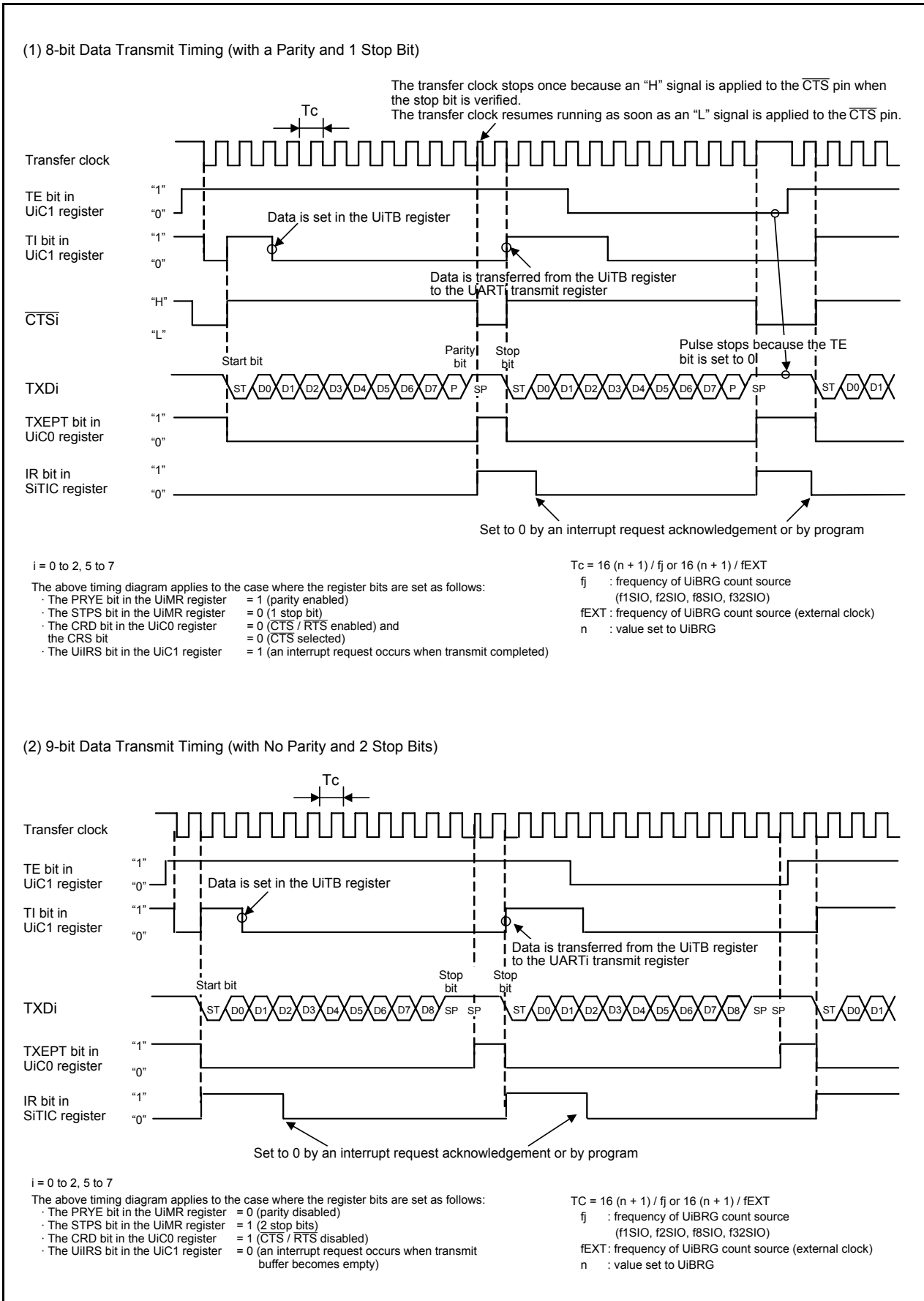
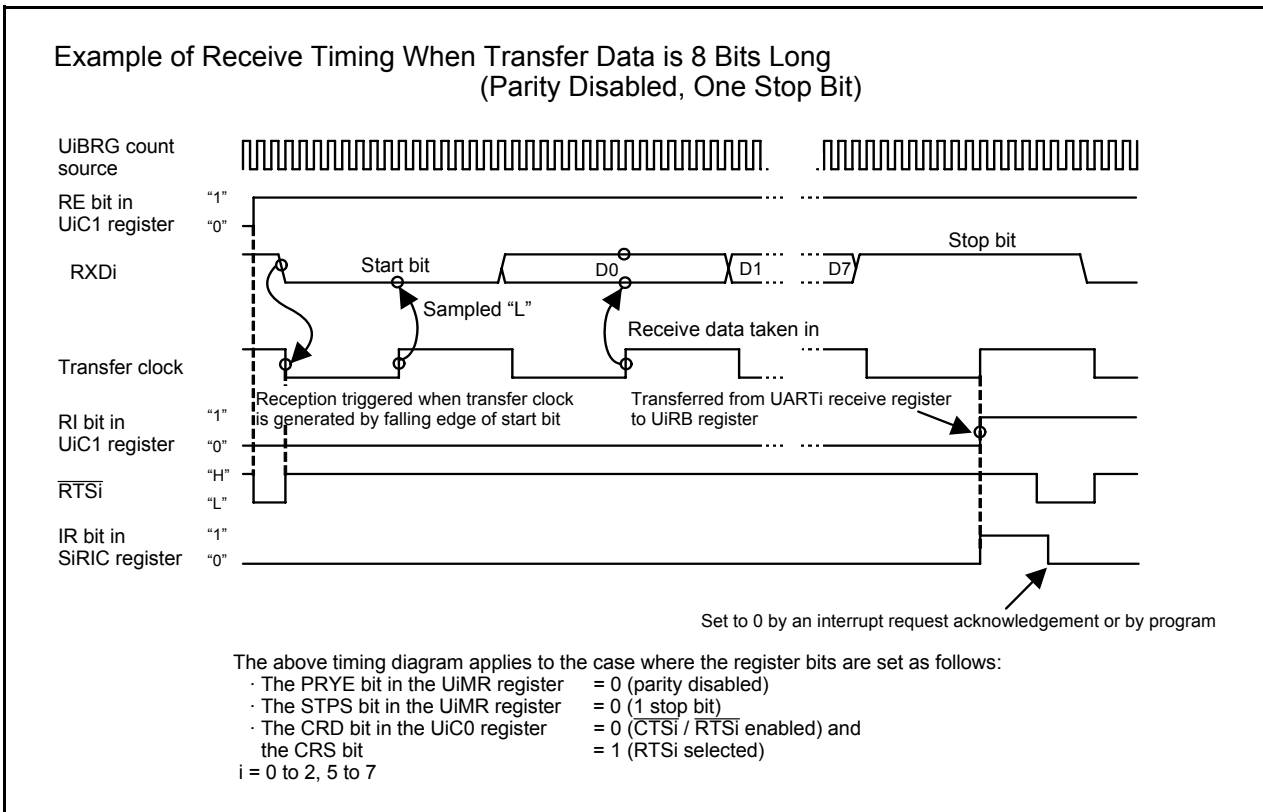


Figure 17.18 Transmit Timing in UART Mode



**Figure 17.19 Receive Timing in UART Mod**

**17.1.2.1 Bit Rate**

In UART mode, the frequency set by the UiBRG register (i = 0 to 2, 5 to 7) divided by 16 become bit rates. Table 17.9 lists an Example of Bit Rates and Settings.

**Table 17.9 Example of Bit Rates and Settings**

Bit Rate (bps)	Count Source of UiBRG	Peripheral Function Clock: 16 MHz		Peripheral Function Clock: 24 MHz	
		Set Value of UiBRG: n	Bit Rate (bps)	Set value of UiBRG: n	Bit Rate (bps)
1200	f8SIO	103 (67h)	1202	155 (9Bh)	1202
2400	f8SIO	51 (33h)	2404	77 (4Dh)	2404
4800	f8SIO	25 (19h)	4808	38 (26h)	4808
9600	f1SIO	103 (67h)	9615	155 (9Bh)	9615
14400	f1SIO	68 (44h)	14493	103 (67h)	14423
19200	f1SIO	51 (33h)	19231	77 (4Dh)	19231
28800	f1SIO	34 (22h)	28571	51 (33h)	28846
31250	f1SIO	31 (1Fh)	31250	47 (2Fh)	31250
38400	f1SIO	25 (19h)	38462	38 (26h)	38462
51200	f1SIO	19 (13h)	50000	28 (1Ch)	51724

### 17.1.2.2 Counter Measure for Communication Error

If a communication error occurs while transmitting or receiving in UART mode, follow the procedures below.

- Resetting the UiRB register (i = 0 to 2, 5 to 7)
  - (1) Set the RE bit in the UiC1 register to 0 (reception disabled)
  - (2) Set the RE bit in the UiC1 register to 1 (reception enabled)
- Resetting the UiTB register
  - (1) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled)
  - (2) Reset bits SMD2 to SMD0 in the UiMR register to 001b, 101b, and 110b.
  - (3) 1 is written to the RE bit in the UiC1 register (transmission enabled), regardless of the TE bit in the UiC1 register

### 17.1.2.3 LSB First / MSB First Select Function

As shown in Figure 17.20, use the UFORM bit in the UiC0 register to select the transfer format. This function is valid when transfer data is 8 bits long.

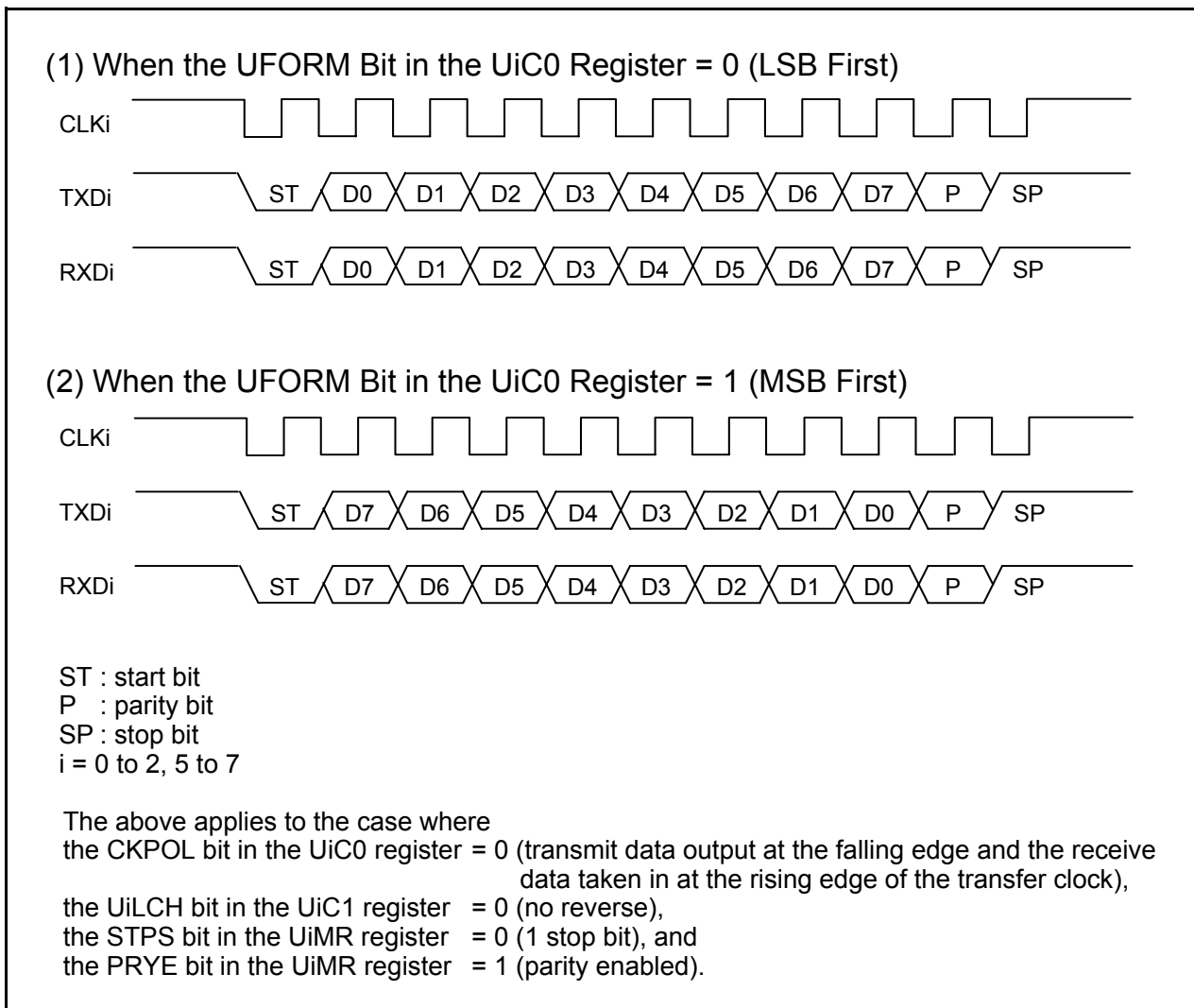


Figure 17.20 Transfer Format

### 17.1.2.4 Serial Data Logic Switching Function

The data written to the UiTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the UiRB register. Figure 17.21 shows Serial Data Logic Switching.

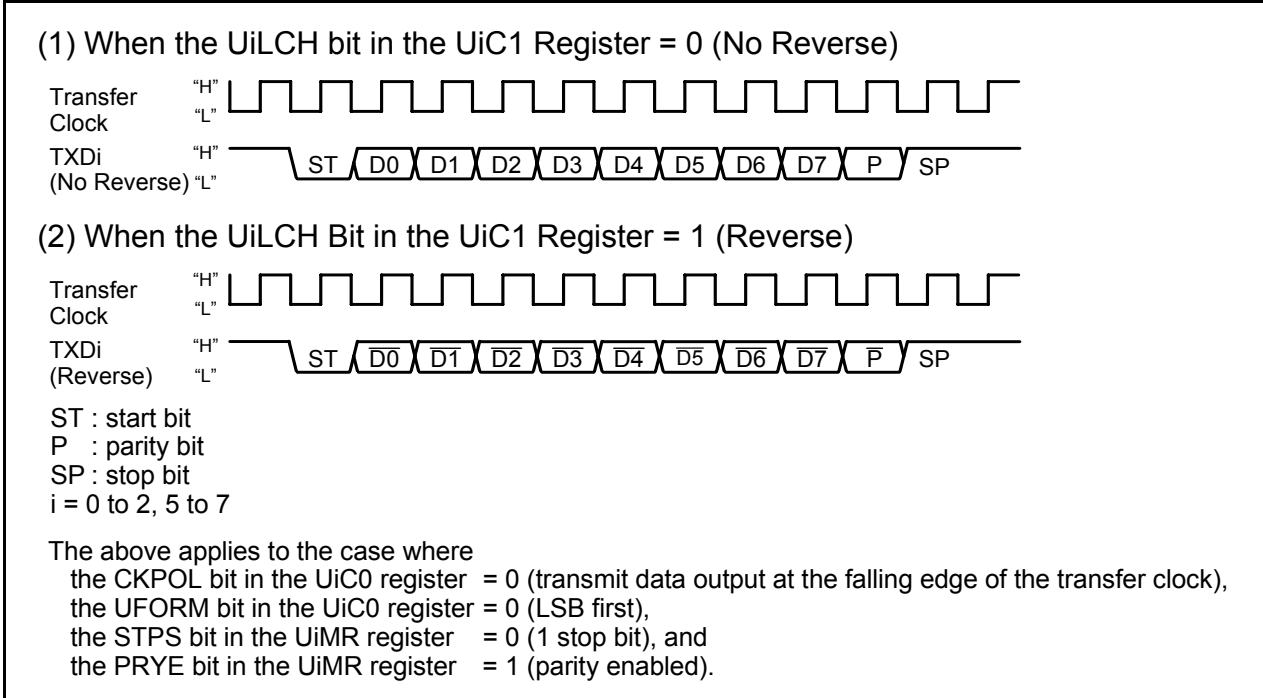


Figure 17.21 Serial Data Logic Switching

### 17.1.2.5 TXD and RXD I/O Polarity Reverse Function

This function reverses the polarities of the TXDi pin output and RXDi pin input. The logic levels of all input / output data (including bits for start, stop, and parity) are reversed. Figure 17.22 shows the TXD and RXD I/O Polarity Reverse.

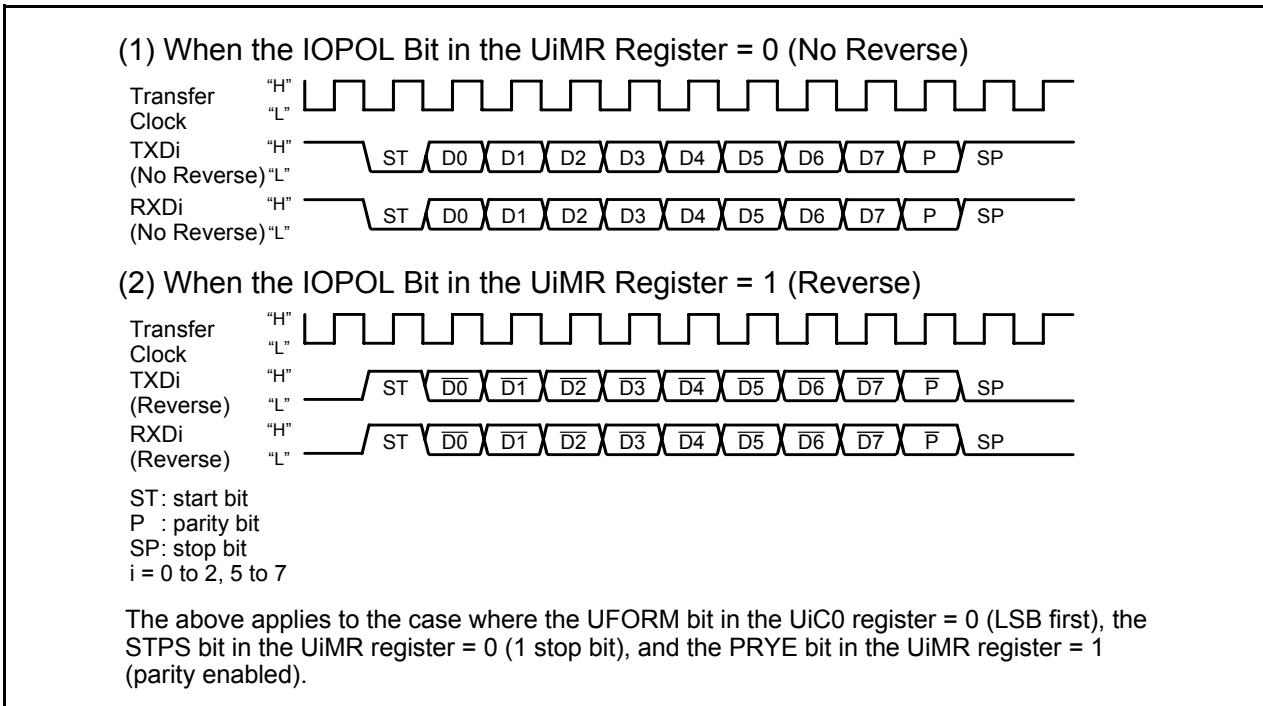


Figure 17.22 TXD and RXD I/O Polarity Reverse

### 17.1.2.6 $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ Function

The  $\overline{\text{CTS}}$  function is used to start transmit operation when “L” is applied to the  $\overline{\text{CTS}}_i$  /  $\overline{\text{RTS}}_i$  ( $i = 0$  to 2, 5 to 7) pin. Transmit operation begins when the  $\overline{\text{CTS}}_i$  /  $\overline{\text{RTS}}_i$  pin is held “L”. If the “L” signal is switched to “H” during a transmit operation, the operation stops after the ongoing transmit / receive operation is completed.

When the  $\overline{\text{RTS}}$  function is used, the  $\overline{\text{CTS}}_i$  /  $\overline{\text{RTS}}_i$  pin outputs “L” when the microcomputer is ready to receive. The output level becomes “H” on the first falling edge of the  $\text{CLK}_i$  pin.

- CRD bit in the UIC0 register = 1 (disable  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  function)  $\overline{\text{CTS}}_i$  /  $\overline{\text{RTS}}_i$  pin is programmable I/O function
- The CRD bit = 0, the CRS bit = 0 ( $\overline{\text{CTS}}$  function is selected)  $\overline{\text{CTS}}_i$  /  $\overline{\text{RTS}}_i$  pin is  $\overline{\text{CTS}}$  function
- The CRD bit = 0, the CRS bit = 1 ( $\overline{\text{RTS}}$  function is selected)  $\overline{\text{CTS}}_i$  /  $\overline{\text{RTS}}_i$  pin is  $\overline{\text{RTS}}$  function

### 17.1.2.7 $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ 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 inputs  $\overline{\text{CTS}}_0$  from the P6\_4 pin. To use this function, set the register bits as shown below.

- The CRD bit in the UOC0 register = 0 (enable  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  of UART0)
- The CRS bit in the UOC0 register = 1 (output  $\overline{\text{RTS}}$  of UART0)
- The CRD bit in the U1C0 register = 0 (enable  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  of UART1)
- The CRS bit in the U1C0 register = 0 (input  $\overline{\text{CTS}}$  of UART1)
- The RCSP bit in the UCON register = 1 (inputs  $\overline{\text{CTS}}_0$  from the P6\_4 pin)
- The CLKMD1 bit in the UCON register = 0 (CLKS1 not used)

Note that when using the  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  separate function,  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  of UART1 function cannot be used.

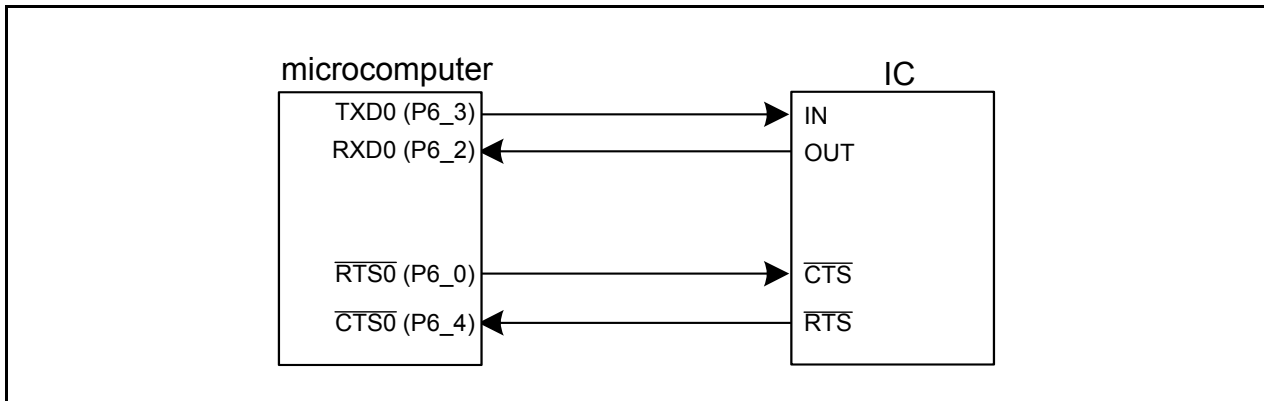


Figure 17.23  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  Separate Function

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

I<sup>2</sup>C mode is provided for use as a simplified I<sup>2</sup>C interface compatible mode. Table 17.10 lists the specifications of I<sup>2</sup>C mode. Tables 17.11 and 17.12 list the registers used in I<sup>2</sup>C mode and the register values set. Table 17.13 lists the I<sup>2</sup>C Mode Functions. Figure 17.24 shows the block diagram for I<sup>2</sup>C mode. Figure 17.25 shows Transfer to UiRB Register and Interrupt Timing.

As shown in Table 17.13, the microcomputer is placed in I<sup>2</sup>C mode by setting bits SMD2 to SMD0 to 010b and the IICM bit to 1. Because SDA<sub>i</sub> transmit output has a delay circuit attached, SDA<sub>i</sub> output does not change state until SCL<sub>i</sub> goes low and remains stably low.

**Table 17.10 I<sup>2</sup>C Mode Specifications**

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer Clock	<ul style="list-style-type: none"> <li>• During master CKDIR bit in the UiMR register = 0 (internal clock): <math>f_j / (2(n+1))</math> <math>f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}</math> <math>n =</math> setting value of the UiBRG register 00h to FFh</li> <li>• During slave CKDIR bit = 1 (external clock): input from the SCL<sub>i</sub> pin</li> </ul>
Transmission Start Condition	Before transmission starts, satisfy the following requirements <sup>(1)</sup> <ul style="list-style-type: none"> <li>• The TE bit in the UiC1 register = 1 (transmission enabled)</li> <li>• The TI bit in the UiC1 register = 0 (data present in UiTB register)</li> </ul>
Reception Start Condition	Before reception starts, satisfy the following requirements <sup>(1)</sup> <ul style="list-style-type: none"> <li>• The RE bit in the UiC1 register = 1 (reception enabled)</li> <li>• The TE bit in the UiC1 register = 1 (transmission enabled)</li> <li>• The TI bit in the UiC1 register = 0 (data present in the UiTB register)</li> </ul>
Interrupt Request Generation Timing	When start or stop condition is detected, acknowledge undetected, or acknowledge detected
Error Detection	Overrun error <sup>(2)</sup> This error occurs if the serial interface started receiving the next data before reading the UiRB register and received the 8th bit of the next data
Select Function	<ul style="list-style-type: none"> <li>• Arbitration lost Timing at which the ABT bit in the UiRB register is updated can be selected</li> <li>• SDA<sub>i</sub> digital delay No digital delay or a delay of 2 to 8 UiBRG count source clock cycles selectable</li> <li>• Clock phase setting With or without clock delay selectable</li> </ul>

$i = 0$  to  $2, 5$  to  $7$

NOTES:

1. When an external clock is selected, the conditions must be met while the external clock is in high state.
2. If an overrun error occurs, the received data of the UiRB register will be indeterminate. The IR bit in the SiRIC register does not change.

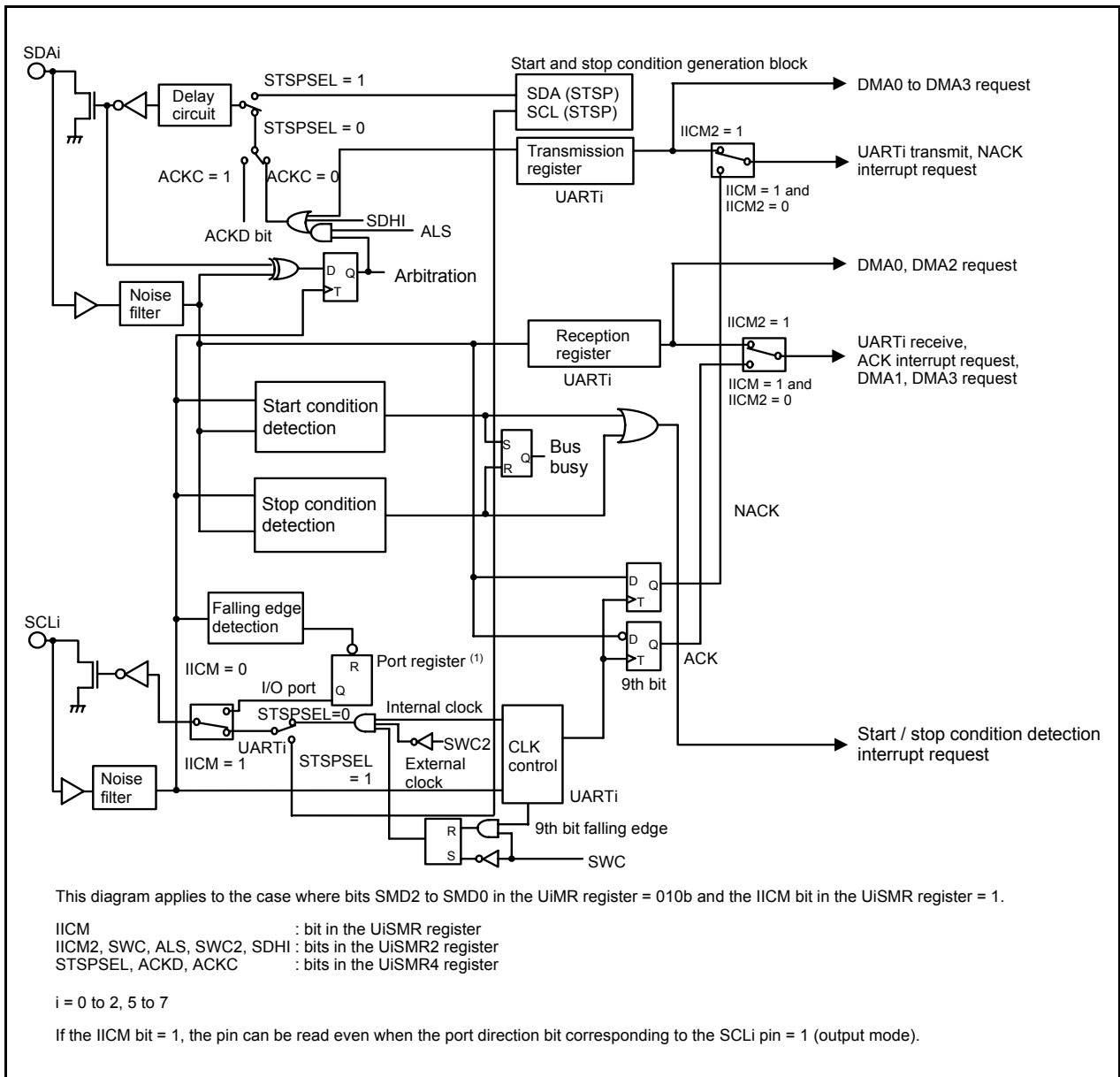


Figure 17.24 I2C Mode Block Diagram



**Table 17.11 Registers to Be Used and Settings in I<sup>2</sup>C Mode (1)**

Register	Bit	Function	
		Master	Slave
UiTB	0 to 7	Set transmission data	Set transmission data
UiRB (3)	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
UiBRG	0 to 7	Set a bit rate	Invalid
UiMR (3)	SMD2 to SMD0	Set to 010b	Set to 010b
	CKDIR	Set to 0	Set to 1
	IOPOL	Set to 0	Set to 0
UiC0	CLK1, CLK0	Select the count source for the UiBRG register	Invalid
	CRS	Invalid because CRD = 1	Invalid because CRD = 1
	TXEPT	Transmit register empty flag	Transmit register empty flag
	CRD (4)	Set to 1	Set to 1
	NCH	Set to 1 (2)	Set to 1 (2)
	CKPOL	Set to 0	Set to 0
	UFORM	Set to 1	Set to 1
UiC1	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
	UiIRS (1)	Invalid	Invalid
	UiRRM (1), UiLCH, UiERE	Set to 0	Set to 0
UiSMR	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
UiSMR2	IICM2	See <b>Table 17.13 I<sup>2</sup>C Mode Functions</b>	See <b>Table 17.13 I<sup>2</sup>C Mode Functions</b>
	CSC	Set this bit to 1 to enable clock synchronization	Set to 0
	SWC	Set this bit to 1 to have SCLi output fixed to "L" at the falling edge of the 9th bit of clock	Set this bit to 1 to have SCLi output fixed to "L" at the falling edge of the 9th bit of clock
	ALS	Set this bit to 1 to have SDAi output stopped when arbitration lost is detected	Set to 0
	STAC	Set to 0	Set this bit to 1 to initialize UAR <sub>Ti</sub> at start condition detection
	SWC2	Set this bit to 1 to have SCLi output forcibly pulled low	Set this bit to 1 to have SCLi output forcibly pulled low
	SDHI	Set this bit to 1 to disable SDAi output	Set this bit to 1 to disable SDAi output
	7	Set to 0	Set to 0

i = 0 to 2, 5 to 7

NOTES:

1. Set the bit 4 and bit 5 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.
2. The TXD2 pin is N channel open-drain output. No NCH bit in the U2C0 register is assigned. When write, set to 0.
3. Set the bits not listed above to 0 when writing to the registers in I<sup>2</sup>C mode.
4. When using UAR<sub>T1</sub> in I<sup>2</sup>C mode and enabling the  $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  separate function of UAR<sub>T0</sub>, set the CRD bit in the U1C0 register to 0 ( $\overline{\text{CTS}}$  /  $\overline{\text{RTS}}$  enabled) and the CRS bit to 0 ( $\overline{\text{CTS}}$  input).

**Table 17.12 Registers to Be Used and Settings in I<sup>2</sup>C Mode (2)**

Register	Bit	Function	
		Master	Slave
UiSMR3	0, 2, 4, and NODC	Set to 0	Set to 0
	CKPH	See Table 17.13 “I <sup>2</sup> C Mode Functions”	See Table 17.13 “I <sup>2</sup> C Mode Functions”
	DL2 to DL0	Set the amount of SDA <sub>i</sub> digital delay	Set the amount of SDA <sub>i</sub> digital delay
UiSMR4	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 SCL <sub>i</sub> output stopped when stop condition is detected	Set to 0
	SWC9	Set to 0	Set this bit to 1 to set the SCL <sub>i</sub> to “L” hold at the falling edge of the 9th bit of clock
IFSR2A	IFSR26, ISFR27	Set to 1	Set to 1
UCON	U0IRS, U1IRS	Invalid	Invalid
	2 to 7	Set to 0	Set to 0

i = 0 to 2, 5 to 7

**Table 17.13 I<sup>2</sup>C Mode Functions**

Function	Clock Synchronous Serial I/O Mode (SMD2 to SMD0 = 001b, IICM = 0)	I <sup>2</sup> C Mode (SMD2 to SMD0 = 010b, 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 5, 6, 7, 10, 27, and 28 (1, 5, 7)	-	Start condition detection or stop condition detection (See Table 17.14 "STSPSEL Bit Functions")			
Factor of Interrupt Number 15, 17, 19, 21, 23, and 24 (1, 6)	UARTi transmission Transmission started or completed (selected by UiIRS)	No acknowledgment detection (NACK) Rising edge of SCLi 9th bit	UARTi transmission Rising edge of SCLi 9th bit	UARTi transmission Falling edge of SCLi next to the 9th bit	
Factor of Interrupt Number 16, 18, 20, 22, 25, and 26 (1, 6)	UARTi reception When 8th bit received CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Acknowledgment detection (ACK) Rising edge of SCLi 9th bit	UARTi reception Falling edge of SCLi 9th bit		
Timing for Transferring Data from the UART Reception Shift Register to the UiRB Register	CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Rising edge of SCLi 9th bit	Falling edge of SCLi 9th bit	Falling and rising edges of SCLi 9th bit	
UARTi Transmission Output Delay	Not delayed	Delayed			
Functions of TXDi / SDAi	TXDi output	SDAi input / output			
Functions of RXDi / SCLi	RXDi input	SCLi input / output			
Functions of CLKi	CLKi input or output port selected	- (Cannot be used in I <sup>2</sup> C mode)			
Noise Filter Width	15ns	200ns			
Read RXDi and SCLi 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 TXDi and SDAi Outputs	CKPOL = 0 ("H") CKPOL = 1 ("L")	The value set in the port register before setting I <sup>2</sup> C mode (2)			
Initial and End Values of SCLi	-	"H"	"L"	"H"	"L"
DMA1 Factor (6)	UARTi reception	Acknowledgment detection (ACK)		UARTi reception Falling edge of SCLi 9th bit	
Store Received Data	1st to 8th bits of the received data are stored into bits 0 to 7 in the UiRB register	1st to 8th bits of the received data are stored into bits 7 to 0 in the UiRB register	1st to 7th bits of the received data are stored into bits 6 to 0 in the UiRB register. 8th bit is stored into bit 8 in the UiRB register		
			1st to 8th bits are stored into bits 7 to 0 in the UiRB register (3)		
Read Received Data	The UiRB register status is read			Bits 6 to 0 in the UiRB register are read as bits 7 to 1. Bit 8 in the UiRB register is read as bit 0 (4)	

i = 0 to 2, 5 to 7

NOTES:

1. If the source or factor 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 24.7 "Interrupt")  
 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.  
 Bits SMD2 to SMD0 in the UiMR register, the IICM bit in the UiSMR register, the IICM2 bit in the UiSMR register, and the CKPH bit in the UiSMR3 register
2. Set the initial value of SDAi output while bits SMD2 to SMD0 in the UiMR register = 000b (serial interface disabled).
3. Second data transfer to the UiRB register (rising edge of SCLi 9th bit)
4. First data transfer to the UiRB register (falling edge of SCLi 9th bit)
5. See Figure 17.27 "STSPSEL Bit Functions".
6. See Figure 17.25 "Transfer to UiRB Register and Interrupt Timing".
7. When using UART0, be sure to set the IFSR26 bit in the IFSR2A register to 1 (factor of interrupt: UART0 bus collision).  
 When using UART1, be sure to set the IFSR27 bit in the IFSR2A register to 1 (factor of interrupt: UART1 bus collision).

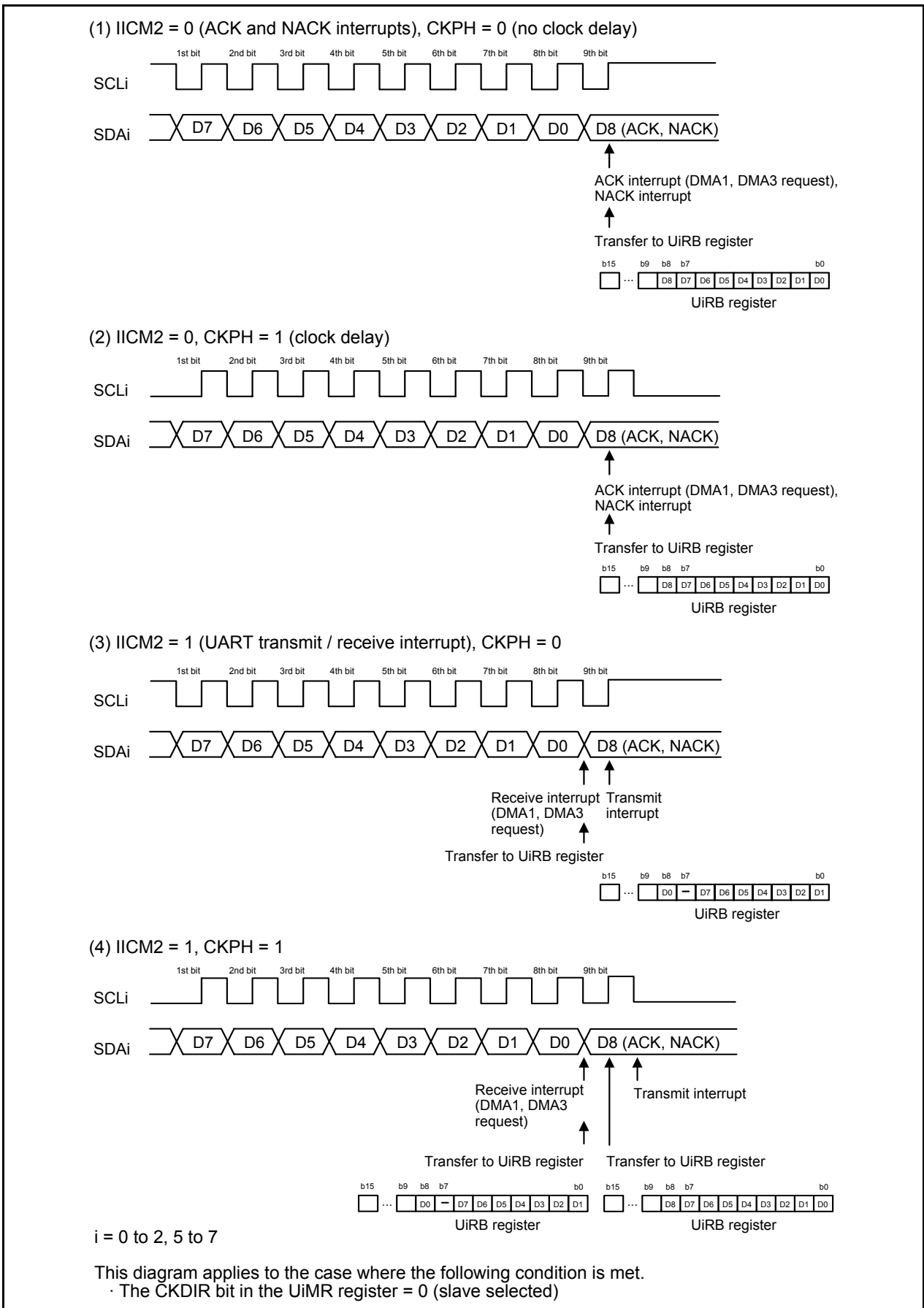


Figure 17.25 Transfer to UiRB Register and Interrupt Timing

### 17.1.3.1 Detection of Start and Stop Condition

Whether a start or a stop condition has been detected is determined.

A start condition detect interrupt request is generated when the SDAi pin changes state from high to low while the SCLi pin is in the high state. A stop condition detect interrupt request is generated when the SDAi pin changes state from low to high while the SCLi pin is in the high state.

Because the start and stop condition detect interrupts share the interrupt control register and vector, check the BBS bit in the UiSMR register to determine which interrupt source is requesting the interrupt.

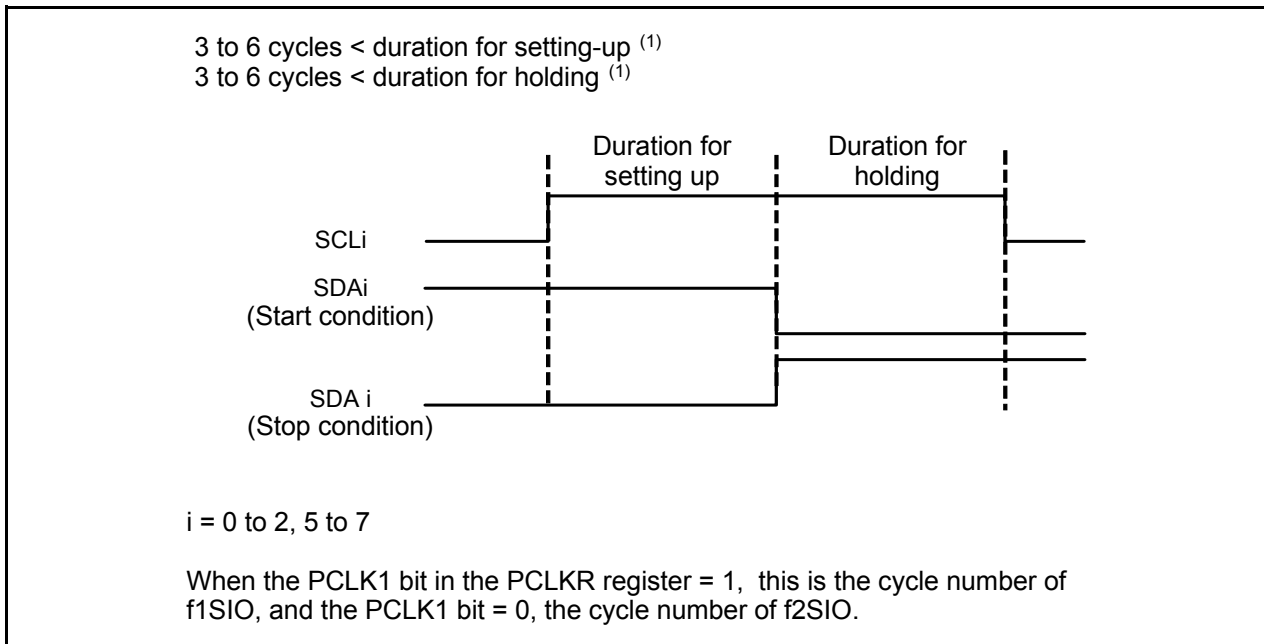


Figure 17.26 Detection of Start and Stop Condition

### 17.1.3.2 Output of Start and Stop Condition

A start condition is generated by setting the STAREQ bit in the UiSMR4 register ( $i = 0$  to 2, 5 to 7) to 1 (start).

A restart condition is generated by setting the RSTAREQ bit in the UiSMR4 register to 1 (start).

A stop condition is generated by setting the STPREQ bit in the UiSMR4 register to 1 (start).

The output procedure is described below.

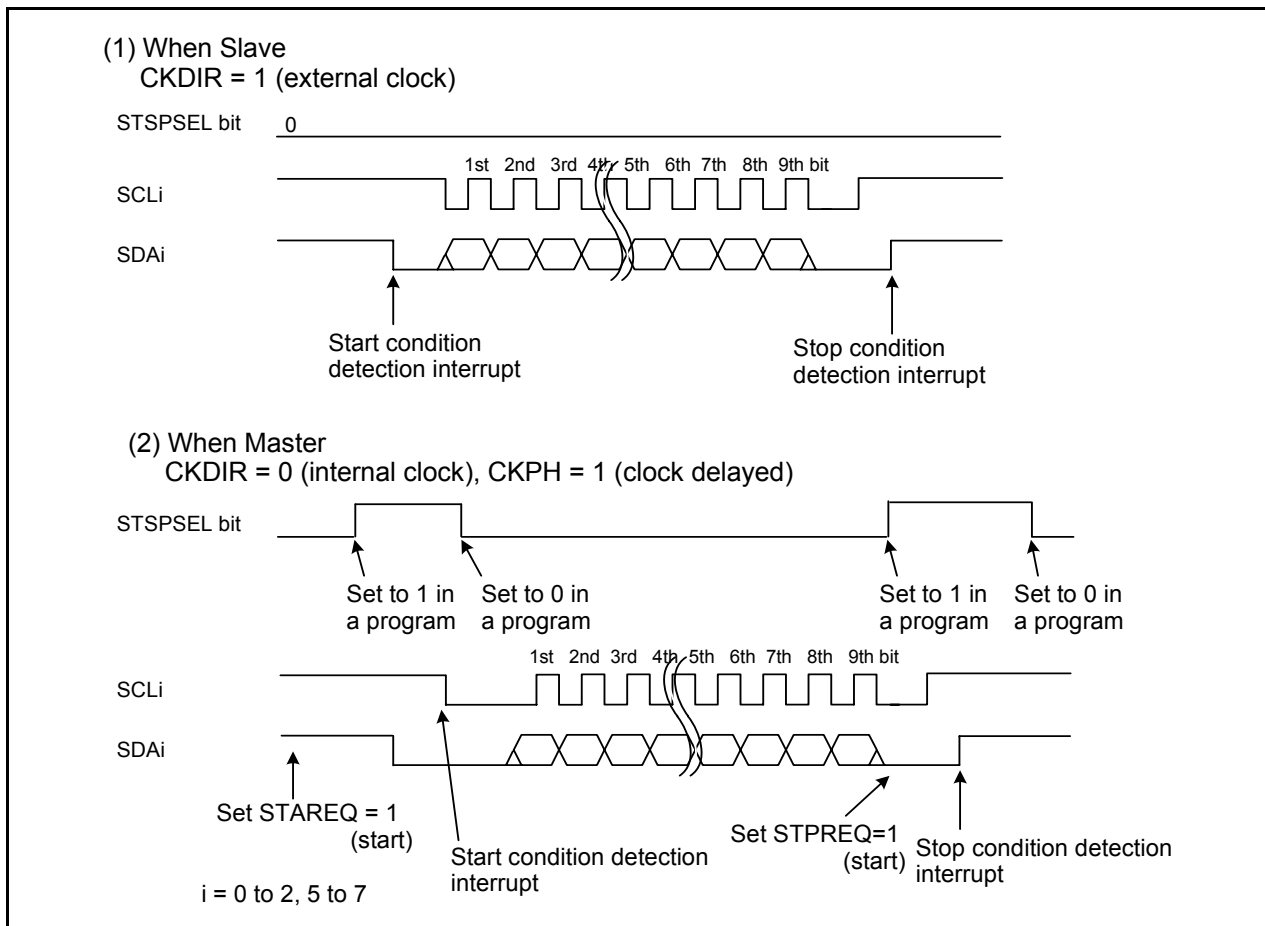
(1) Set the STAREQ bit, RSTAREQ bit or STPREQ bit to 1 (start).

(2) Set the STSPSEL bit in the UiSMR4 register to 1 (output).

The function of the STSPSEL bit is shown in Tables 17.14 and 17.27.

**Table 17.14 STSPSEL Bit Functions**

Function	STSPSEL = 0	STSPSEL = 1
Output of Pins SCLi and SDAi	Output of transfer clock and data 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 bits STAREQ, RSTAREQ, and STPREQ
Start / Stop Condition Interrupt Request Generation Timing	Detect start / stop condition	Complete generating start / stop condition



**Figure 17.27 STSPSEL Bit Functions**

**17.1.3.3 Arbitration**

Unmatching of the transmit data and SDAi pin input data is checked synchronously with the rising edge of SCLi. Use the ABC bit in the UiSMR register to select the timing at which the ABT bit in the UiRB register is updated. If the ABC bit = 0 (update per bit), 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 ever detected, 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 per byte, clear the ABT bit to 0 (undetected) after detecting acknowledge in the first byte, before transferring the next byte. Setting the ALS bit in the UiSMR2 register to 1 (SDA output stop enabled) factors arbitration-lost to occur, in which case the SDAi pin is placed in the high-impedance state at the same time the ABT bit is set to 1 (unmatching detected).

#### 17.1.3.4 Transfer Clock

The transfer clock is used to transmit and receive data as is shown in Figure 17.25 Transfer to UiRB Register and Interrupt Timing.

The CSC bit in the UiSMR2 register is used to synchronize the internally generated clock (internal SCLi) and an external clock supplied to the SCLi pin. In cases when the CSC bit is set to 1 (clock synchronization enabled), if a falling edge on the SCLi pin is detected while the internal SCLi is high, the internal SCLi goes low, at which time the value of the UiBRG register is reloaded with and starts counting in the low-level interval. If the internal SCLi changes state from low to high while the SCLi pin is low, counting stops, and when the SCLi pin goes high, counting restarts.

In this way, the UARTi transfer clock is equivalent to AND of the internal SCLi and the clock signal applied to the SCLi pin. The transfer clock works between a half cycle before the falling edge of the internal SCLi 1st bit and the rising edge of the 9th bit. To use this function, select an internal clock for the transfer clock.

The SWC bit in the UiSMR2 register determines whether the SCLi pin is fixed to be or freed from low-level output at the falling edge of the 9th clock pulse.

If the SCLHI bit in the UiSMR4 register is set to 1 (enabled), SCLi output is turned off (placed in the high-impedance state) when a stop condition is detected.

Setting the SWC2 bit in the UiSMR2 register = 1 (0 output) makes it possible to forcibly output a low-level signal from the SCLi 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 SCLi pin, instead of outputting a low-level signal.

If the SWC9 bit in the UiSMR4 register is set to 1 (SCL hold low enabled) when the CKPH bit in the UiSMR3 register = 1, the SCLi pin is fixed to low-level output at the falling edge of the clock pulse next to the 9th. Setting the SWC9 bit = 0 (SCL hold low disabled) frees the SCLi pin from low-level output.

#### 17.1.3.5 SDA Output

The data written to bits 7 to 0 (D7 to D0) in the UiTB register is output in descending order from D7.

The 9th bit (D8) is ACK or NACK.

Set the initial value of SDAi transmit output when IICM = 1 (I<sup>2</sup>C mode) and bits SMD2 to SMD0 in the UiMR register = 000b (serial interface disabled).

Bits DL2 to DL0 in the UiSMR3 register allow to add no delays or a delay of 2 to 8 UiBRG count source clock cycles to SDAi output.

Setting the SDHI bit in the UiSMR2 register = 1 (SDA output disabled) forcibly places the SDAi pin in the high-impedance state. Do not write to the SDHI bit at the rising edge of the UARTi transfer clock. This is because the ABT bit may inadvertently be set to 1 (detected).

#### 17.1.3.6 SDA Input

When the IICM2 bit = 0, the 1st to 8th bits (D7 to D0) of received data are stored in bits 7 to 0 in the UiRB register. The 9th bit (D8) is ACK or NACK.

When the IICM2 bit = 1, the 1st to 7th bits (D7 to D1) of received data are stored in bits 6 to 0 in the UiRB register and the 8th bit (D0) is stored in bit 8 in the UiRB register. Even when the IICM2 bit = 1, providing the CKPH bit = 1, the same data as when the IICM2 bit = 0 can be read. To read the data, read the UiRB register after the rising edge of 9th bit of the corresponding clock pulse.

### 17.1.3.7 ACK and NACK

If the STSPSEL bit in the UiSMR4 register is set to 0 (start and stop conditions not generated) and the ACKC bit in the UiSMR4 register is set to 1 (ACK data output), the value of the ACKD bit in the UiSMR4 register is output from the SDAi pin.

If the IICM2 bit = 0, the NACK interrupt request is generated if the SDAi pin remains high at the rising edge of the 9th bit of transmit clock pulse. The ACK interrupt request is generated if the SDAi pin is low at the rising edge of the 9th bit of transmit clock pulse.

If ACKi is selected to generate a DMA1 or DMA3 request source, a DMA transfer can be activated by detection of an acknowledge.

### 17.1.3.8 Initialization of Transmission / Reception

If a start condition is detected while the STAC bit = 1 (UARTi initialization enabled), the serial interface operates as described below.

- The transmit shift register is initialized, and the content of the UiTB register is transferred to the transmit shift register. In this way, the serial interface starts sending data synchronously with the next clock pulse applied. However, the UARTi 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 interface starts receiving data synchronously with the next clock pulse applied.
- The SWC bit is set to 1 (SCL wait output enabled). Consequently, the SCLi pin is pulled low at the falling edge of the 9th clock pulse.

Note that when UARTi transmission / reception is started using this function, the TI bit does not change state. Select the external clock as the transfer clock to start UARTi transmission / reception with this setting.



### 17.1.4 Special Mode 2

In special mode 2, serial communication between one or multiple masters and multiple slaves is available. Transfer clock polarity and phase are selectable. Table 17.15 lists the Special Mode 2 Specifications. Table 17.16 lists the Registers to Be Used and Settings in Special Mode 2. Figure 17.28 shows Special Mode 2 Communication Control Example (UART2).

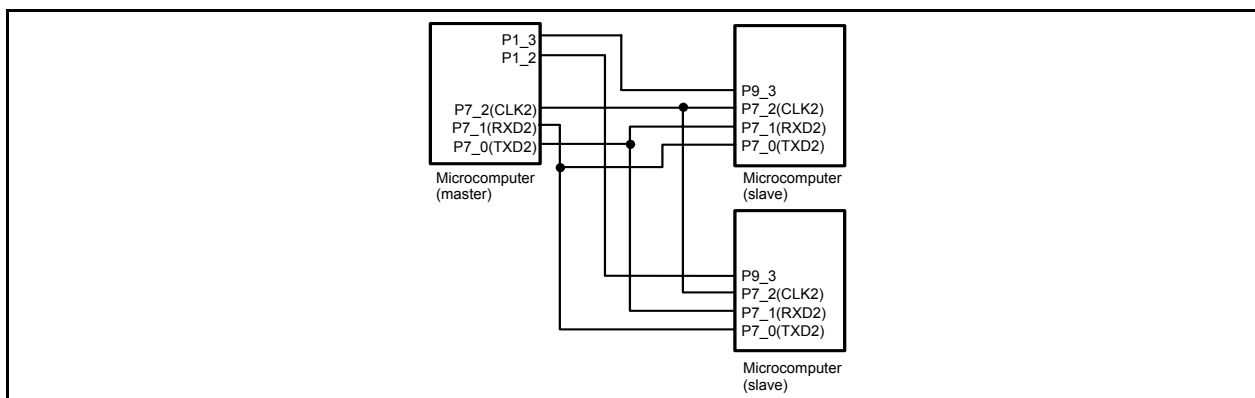
**Table 17.15 Special Mode 2 Specifications**

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer Clock	<ul style="list-style-type: none"> <li>Master mode The CKDIR bit in the UiMR register = 0 (internal clock): <math>f_j / (2(n + 1))</math> <math>f_j = f1SIO, f2SIO, f8SIO, f32SIO</math> n: setting value of UiBRG register 00h to FFh</li> <li>Slave mode</li> <li>The CKDIR bit = 1 (external clock selected): input from the CLKi pin</li> </ul>
Transmit / Receive Control	Controlled by input / output ports
Transmission Start Condition	Before transmission starts, satisfy the following requirements (1) <ul style="list-style-type: none"> <li>The TE bit in the UiC1 register = 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register = 0 (data present in UiTB register)</li> </ul>
Reception Start Condition	Before reception starts, satisfy the following requirements (1) <ul style="list-style-type: none"> <li>The RE bit in the UiC1 register = 1 (reception enabled)</li> <li>The TE bit = 1 (transmission enabled)</li> <li>The TI bit = 0 (data present in the UiTB register)</li> </ul>
Interrupt Request Generation Timing	While transmitting, one of the following conditions can be selected <ul style="list-style-type: none"> <li>The UiIRS bit in the UiC1 register = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission)</li> <li>The UiIRS bit = 1 (transfer completed): when the serial interface completed sending data from the UARTi transmit register</li> </ul> While receiving <ul style="list-style-type: none"> <li>When transferring data from the UARTi receive register to the UiRB register (at completion of reception)</li> </ul>
Error Detection	Overrun error (2) This error occurs if the serial interface starts receiving the next data before reading the UiRB register and receives the 7th bit of the next data
Select Function	Clock phase setting Selectable from four combinations of transfer clock polarities and phases

i = 0 to 2, 5 to 7

**NOTES:**

- When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 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 high state; if the 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 low state.
- If an overrun error occurs, the received data of the UiRB register will be indeterminate. The IR bit in the SiRIC register does not change.



**Figure 17.28 Special Mode 2 Communication Control Example (UART2)**

**Table 17.16 Registers to Be Used and Settings in Special Mode 2**

Register	Bit	Function
UiTB (3)	0 to 7	Set transmission data
UiRB (3)	0 to 7	Reception data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a bit rate
UiMR (3)	SMD2 to SMD0	Set to 001b
	CKDIR	Set to 0 in master mode or 1 in slave mode
	IOPOL	Set to 0
UiC0	CLK0, CLK1	Select the count source for the UiBRG register
	CRS	Invalid because CRD = 1
	TXEPT	Transmit register empty flag
	CRD	Set to 1
	NCH	Select TXDi pin output format (2)
	CKPOL	Clock phases can be set in combination with the CKPH bit in the UiSMR3 register
	UFORM	Set to 0
UiC1	TE	Set to 1 to enable transmission / reception
	TI	Transmit buffer empty flag
	RE	Set to 1 to enable reception
	RI	Reception complete flag
	UiIRS (1)	Select UART2 transmit interrupt source
	UiRRM (1), UiLCH, UiERE	Set to 0
UiSMR	0 to 7	Set to 0
UiSMR2	0 to 7	Set to 0
UiSMR3	CKPH	Clock phases can be set in combination with the CKPOL bit in the UiC0 register
	NODC	Set to 0
	0, 2, 4 to 7	Set to 0
UiSMR4	0 to 7	Set to 0
UON	U0IRS, U1IRS	Select UART0 and UART1 transmit interrupt source
	U0RRM, U1RRM	Set to 0
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1, RCSP, 7	Set to 0

i = 0 to 2, 5 to 7

NOTES:

1. Set bits 4 and 5 in registers U0C0 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UON register.
2. The TXD2 pin is N channel open-drain output. No NCH bit in the U2C0 register is assigned. When write, set to 0.
3. Set the bits not listed above to 0 when writing to the registers in special mode 2.

### 17.1.4.1 Clock Phase Setting Function

One of four combinations of transfer clock phases and polarities can be selected using the CKPH bit in the UiSMR3 register and the CKPOL bit in the UiC0 register.

Make sure the transfer clock polarity and phase are the same for the master and slaves to be communicated.

Figure 17.29 shows the Transmission and Reception Timing in Master Mode (Internal Clock).

Figure 17.30 shows the Transmission and Reception Timing (CKPH = 0) in Slave Mode (External Clock) while Figure 17.31 shows the Transmission and Reception Timing (CKPH = 1) in Slave Mode (External Clock).

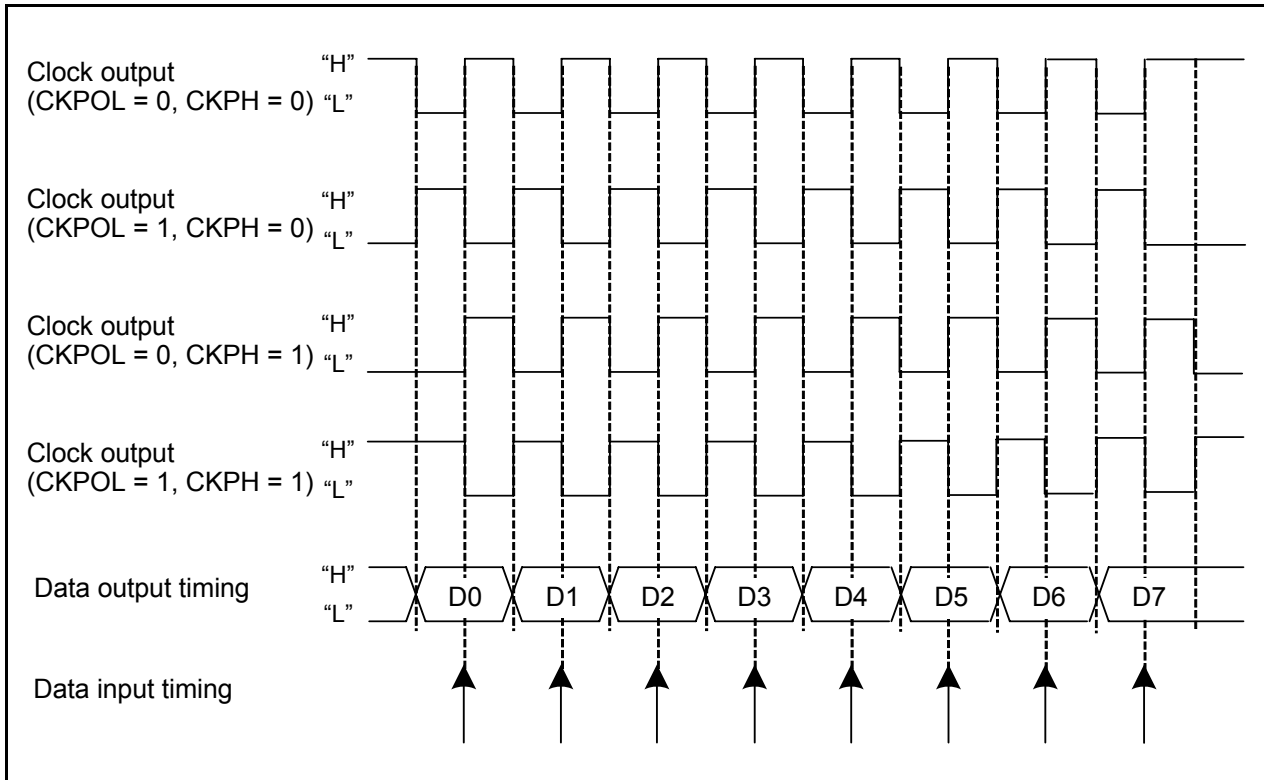


Figure 17.29 Transmission and Reception Timing in Master Mode (Internal Clock)

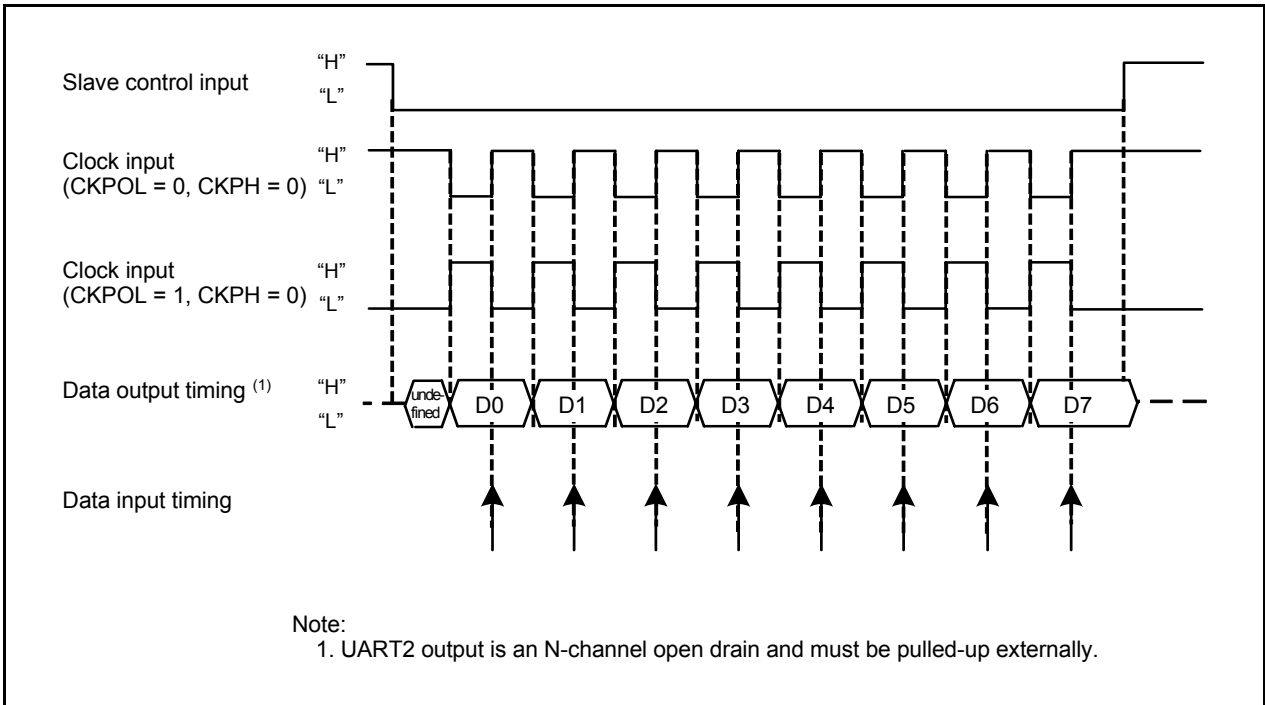


Figure 17.30 Transmission and Reception Timing (CKPH = 0) in Slave Mode (External Clock)

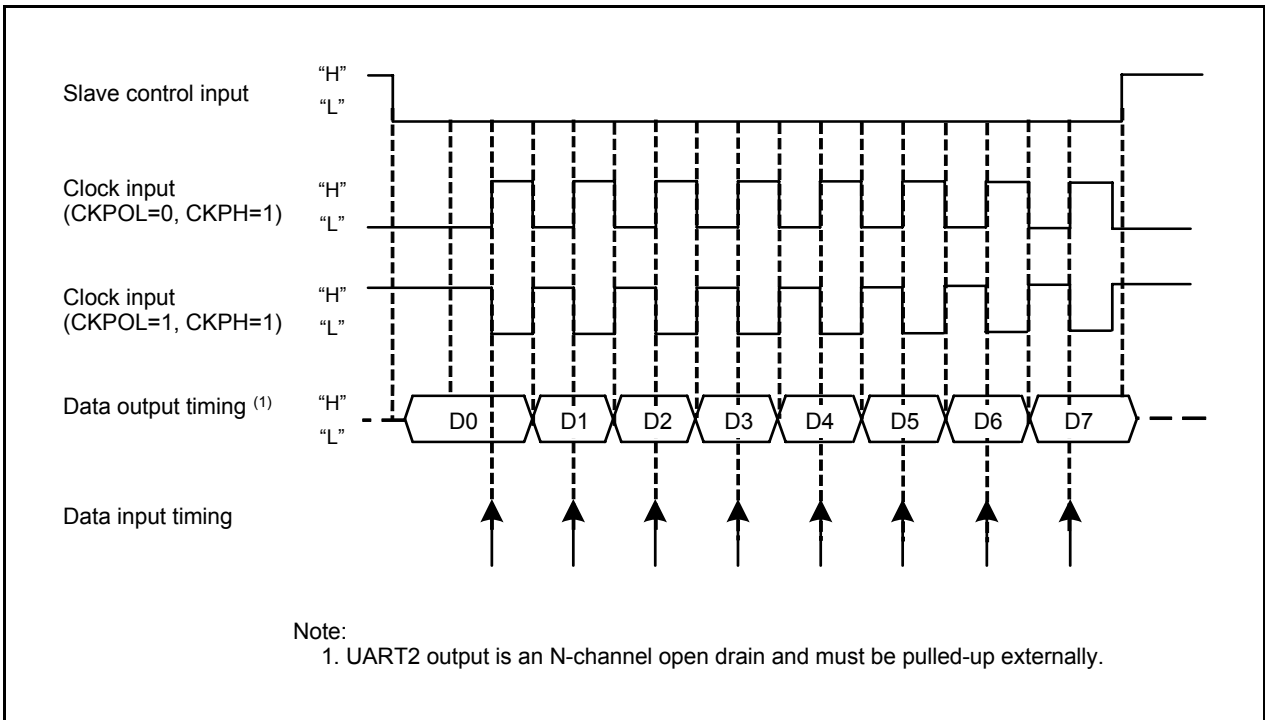


Figure 17.31 Transmission and Reception Timing (CKPH = 1) in Slave Mode (External Clock)

### 17.1.5 Special Mode 3 (IE mode)

In this mode, one bit of IEBus is approximated with one byte of UART mode waveform.

Table 17.17 lists the Registers to Be Used and Settings in IE Mode. Figure 17.32 shows the Bus Collision Detect Function-Related Bits.

If the TXDi pin (i = 0 to 2, 5 to 7) output level and RXDi pin input level do not match, a UARTi bus collision detect interrupt request is generated.

Use bits IFSR26 and IFSR27 in the IFSR2A register to enable the UART0 / UART1 bus collision detect function.

**Table 17.17 Registers to Be Used and Settings in IE Mode**

Register	Bit	Function
UiTB	0 to 8	Set transmission data
UiRB <sup>(3)</sup>	0 to 8	Reception data can be read
	OER, FER, PER, SUM	Error flag
UiBRG	0 to 7	Set a bit rate
UiMR	SMD2 to SMD0	Set to 110b
	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 and RXD input / output polarity
UiC0	CLK1, CLK0	Select the count source for the UiBRG register
	CRS	Invalid because CRD = 1
	TXEPT	Transmit register empty flag
	CRD	Set to 1
	NCH	Select TXDi pin output format <sup>(2)</sup>
	CKPOL	Set to 0
	UFORM	Set to 0
UIC1	TE	Set to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set to 1 to enable reception
	RI	Reception complete flag
	UiIRS <sup>(1)</sup>	Select the source of UARTi transmit interrupt
	UiRRM <sup>(1)</sup> , UiLCH, UiERE	Set to 0
UiSMR	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
UiSMR2	0 to 7	Set to 0
UiSMR3	0 to 7	Set to 0
UiSMR4	0 to 7	Set to 0
IFSR2A	IFSR26, IFSR27	Set to 1
UCON	U0IRS, U1IRS	Select the source of UART0 / UART1 transmit interrupt
	U0RRM, U1RRM	Set to 0
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1, RCSP, 7	Set to 0

i = 0 to 2, 5 to 7

NOTES:

- Set bits 4 and 5 in registers U0C0 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.
- The TXD2 pin is N channel open-drain output. No NCH bit in the U2C0 register is assigned. When write, set to 0.
- Set the bits not listed above to 0 when writing to the registers in IE mode.

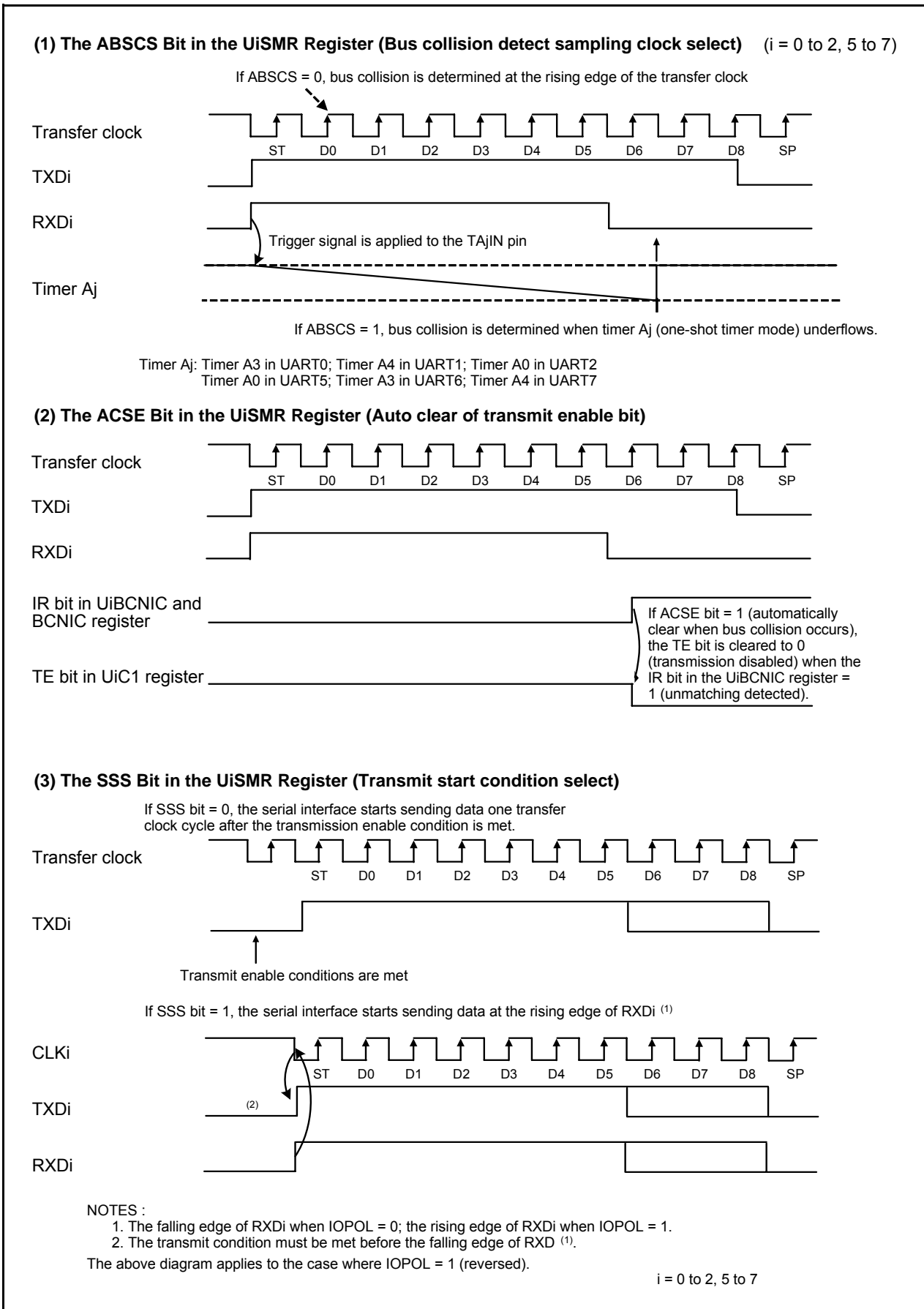


Figure 17.32 Bus Collision Detect Function-Related Bits

### 17.1.6 Special Mode 4 (SIM Mode) (UART2)

SIM interface devices can communicate in UART mode. Both direct and inverse formats are available. The TXD2 pin outputs a low-level signal when a parity error is detected.

Table 17.18 lists the SIM Mode Specifications. Table 17.19 lists the Registers to Be Used and Settings in SIM Mode.

**Table 17.18 SIM Mode Specifications**

Item	Specification
Transfer Data Format	<ul style="list-style-type: none"> <li>• Direct format</li> <li>• Inverse format</li> </ul>
Transfer Clock	<ul style="list-style-type: none"> <li>• The CKDIR bit in the U2MR register = 0 (internal clock): <math>f_i / (16(n + 1))</math>  <math>f_i = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}</math>  <math>n =</math> setting value of the U2BRG register 00h to FFh</li> <li>• The CKDIR bit = 1 (external clock): <math>f_{EXT} / (16(n + 1))</math>  <math>f_{EXT} =</math> input from the CLK2 pin  <math>n =</math> setting value of the U2BRG register 00h to FFh</li> </ul>
Transmission Start Condition	Before transmission starts, satisfy the following requirements <ul style="list-style-type: none"> <li>• The TE bit in the U2C1 register = 1 (transmission enabled)</li> <li>• The TI bit in the U2C1 register = 0 (data present in the U2TB register)</li> </ul>
Reception Start Condition	Before reception starts, satisfy the following requirements <ul style="list-style-type: none"> <li>• The RE bit in the U2C1 register = 1 (reception enabled)</li> <li>• Start bit detection</li> </ul>
Interrupt Request Generation Timing <sup>(2)</sup>	<ul style="list-style-type: none"> <li>• While transmitting                      When the serial interface completed sending data from the UART2 transmit register (the U2IRS bit = 1)</li> <li>• While receiving                      When transferring data from the UART2 receive register to the U2RB register (at completion of reception)</li> </ul>
Error Detection	<ul style="list-style-type: none"> <li>• Overrun error <sup>(1)</sup>                      This error occurs if the serial interface started receiving the next data before reading the U2RB register and received the bit one before the last stop bit of the next data</li> <li>• Framing error <sup>(3)</sup>                      This error occurs when the number of stop bits set is not detected</li> <li>• Parity error <sup>(3)</sup>                      During reception, if a parity error is detected, parity error signal is output from the TXD2 pin.                      During transmission, a parity error is detected by the level of input to the RXD2 pin when a transmission interrupt occurs</li> <li>• Error sum flag                      This flag is set to 1 when one of the overrun, framing, and parity errors occurs</li> </ul>

NOTES:

1. If an overrun error occurs, the received data of the U2RB register will be indeterminate. The IR bit in the S2RIC register does not change.
2. A transmit interrupt request is generated by setting the U2IRS bit to 1 (transmission completed) and the U2ERE bit to 1 (error signal output) in the U2C1 register after reset is canceled. Therefore, when using SIM mode, set the IR bit to 0 (interrupt not requested) after setting the bits.
3. The timing at which the framing error flag and the parity error flag are set is detected when data is transferred from the UART2 receive register to the U2RB register.

**Table 17.19 Registers to Be Used and Settings in SIM Mode**

Register	Bit	Function
U2TB (1)	0 to 7	Set transmission data
U2RB (1)	0 to 7	Reception data can be read
	OER,FER,PER,SUM	Error flag
U2BRG	0 to 7	Set a bit rate
U2MR	SMD2 to SMD0	Set to 101b
	CKDIR	Select the internal clock or external clock
	STPS	Set to 0
	PRY	Set to 1 in direct format or 0 in inverse format
	PRYE	Set to 1
	IOPOL	Set to 0
U2C0	CLK0,CLK1	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 to 0 in direct format or 1 in inverse format
U2C1	TE	Set to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set to 1 to enable reception
	RI	Reception complete flag
	U2IRS	Set to 1
	U2RRM	Set to 0
	U2LCH	Set to 0 in direct format or 1 in inverse format
	U2ERE	Set to 1
U2SMR (1)	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:

1. Set the bits not listed above to 0 when writing to the registers in SIM mode.



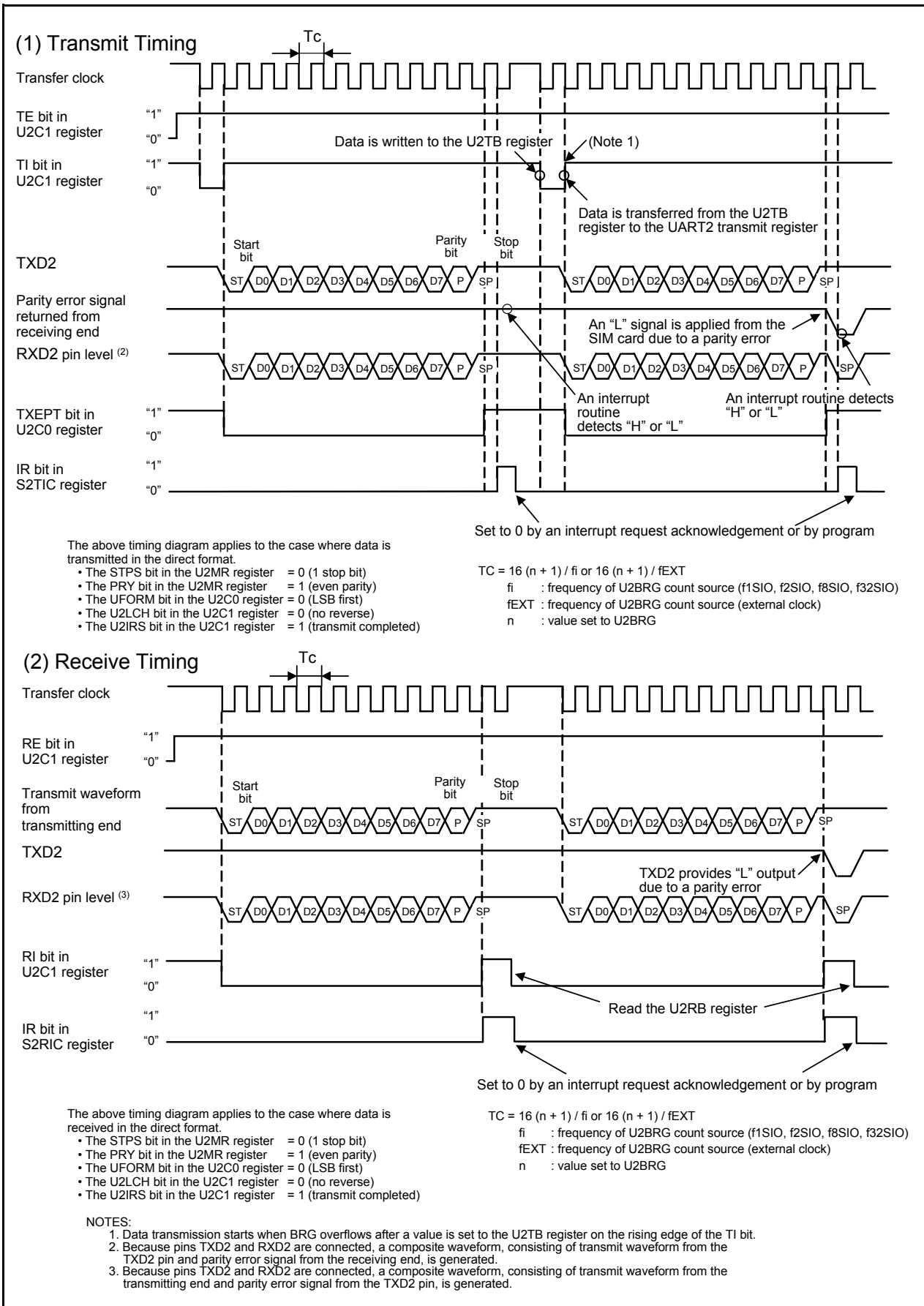


Figure 17.33 Transmit and Receive Timing in SIM Mode

Figure 17.34 shows an Example of SIM Interface Connection. Connect TXD2 and RXD2, and then place a pull-up resistance.

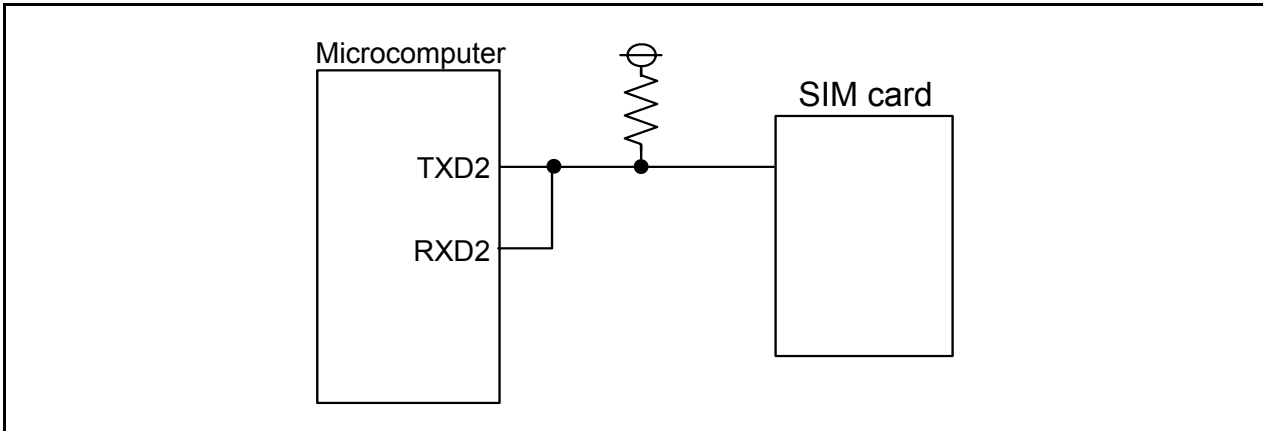


Figure 17.34 Example of SIM Interface Connection

### 17.1.6.1 Parity Error Signal Output

The parity error signal is enabled by setting the U2ERE bit in the U2C1 register to 1 (error signal output).

The parity error signal is output when a parity error is detected while receiving data. A low-level signal is output from the TXD2 pin in the timing shown in Figure 17.35. If the U2RB register is read while outputting a parity error signal, the PER bit is cleared to 0 (no parity error) and at the same time the TXD2 output is returned high.

When transmitting, a transmission complete interrupt request is generated at the falling edge of the transfer clock pulse that immediately follows the stop bit. Therefore, whether a parity error signal has been returned can be determined by reading the port that shares the RXD2 pin in a transmission complete interrupt routine.

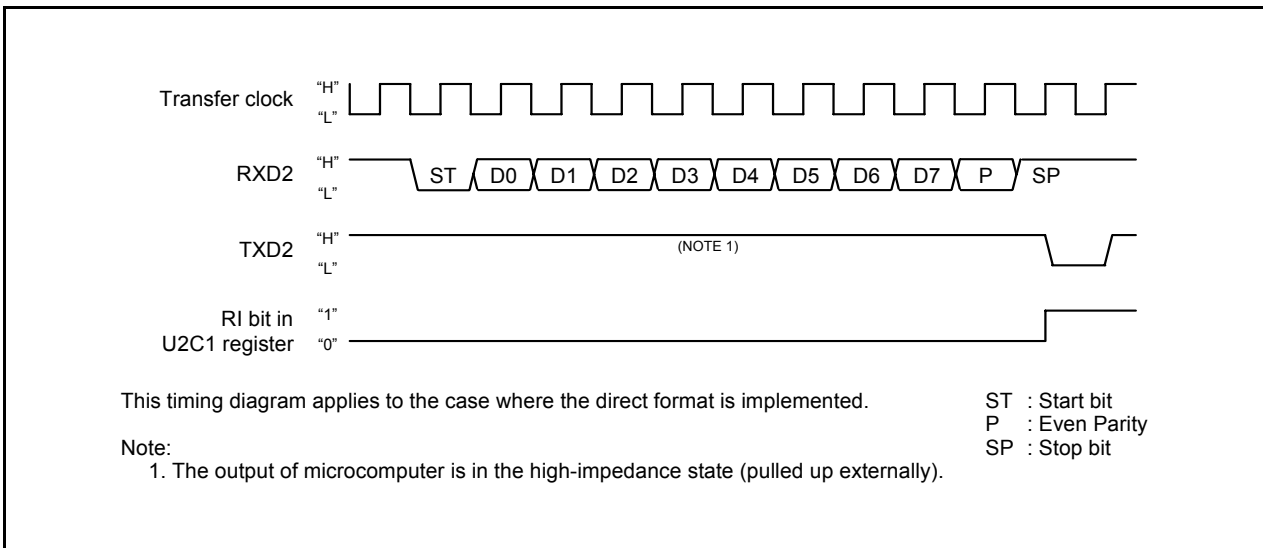


Figure 17.35 Parity Error Signal Output Timing

### 17.1.6.2 Format

Two formats are available: direct format and inverse format.

In direct format, set the PRYE bit in the U2MR register to 1 (parity enabled), the PRY bit to 1 (even parity), the UFORM bit in the U2C0 register to 0 (LSB first) and the U2LCH bit in the U2C1 register to 0 (not inverted). When data are transmitted, data set in the U2TB register are transmitted with the even-numbered parity, starting from D0. When data are received, received data are stored in the U2RB register, starting from D0. The even-numbered parity determines whether a parity error occurs. In inverse format, set the PRYE bit to 1, the PRY bit to 0 (odd parity), the UFORM bit to 1 (MSB first), and the U2LCH bit to 1 (inverted). When data are transmitted, values set in the U2TB register are logically inversed and are transmitted with the odd-numbered parity, starting from D7. When data are received, received data are logically inversed to be stored in the U2RB register, starting from D7. The odd-numbered parity determines whether a parity error occurs.

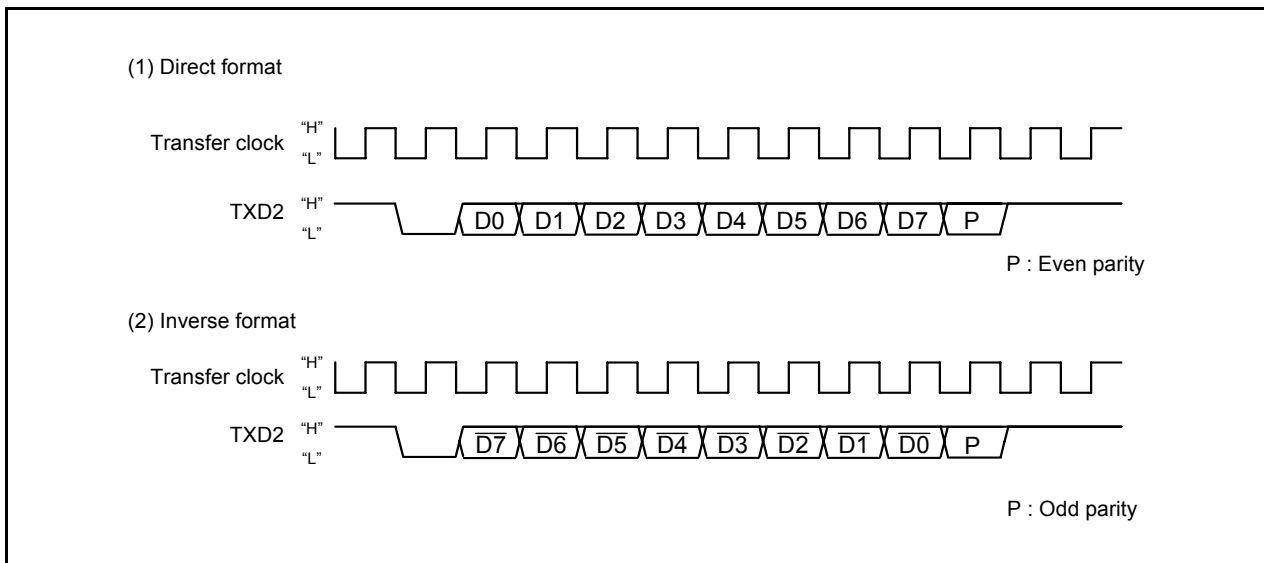


Figure 17.36 SIM Interface Format

### 17.2 SI/O3 and SI/O4

SI/O3 and SI/O4 are exclusive clock-synchronous serial I/Os.

Figure 17.37 shows the SI/O3 and SI/O4 Block Diagram, and Figure 17.38 shows the Registers S3C, S4C, S3BRG, S4BRG, S3TRR, and S4TRR.

Table 17.20 shows the SI/O3 and SI/O4 Specifications.

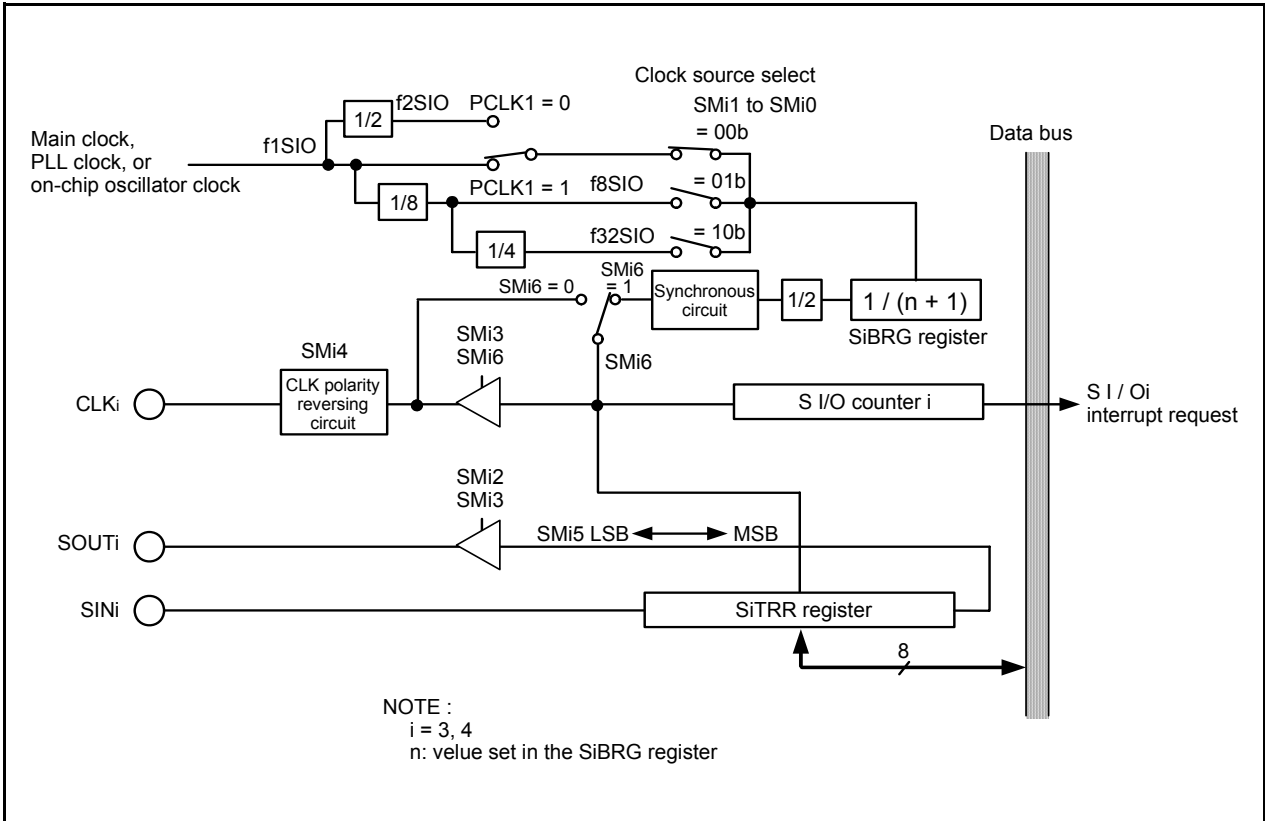


Figure 17.37 SI/O3 and SI/O4 Block Diagram

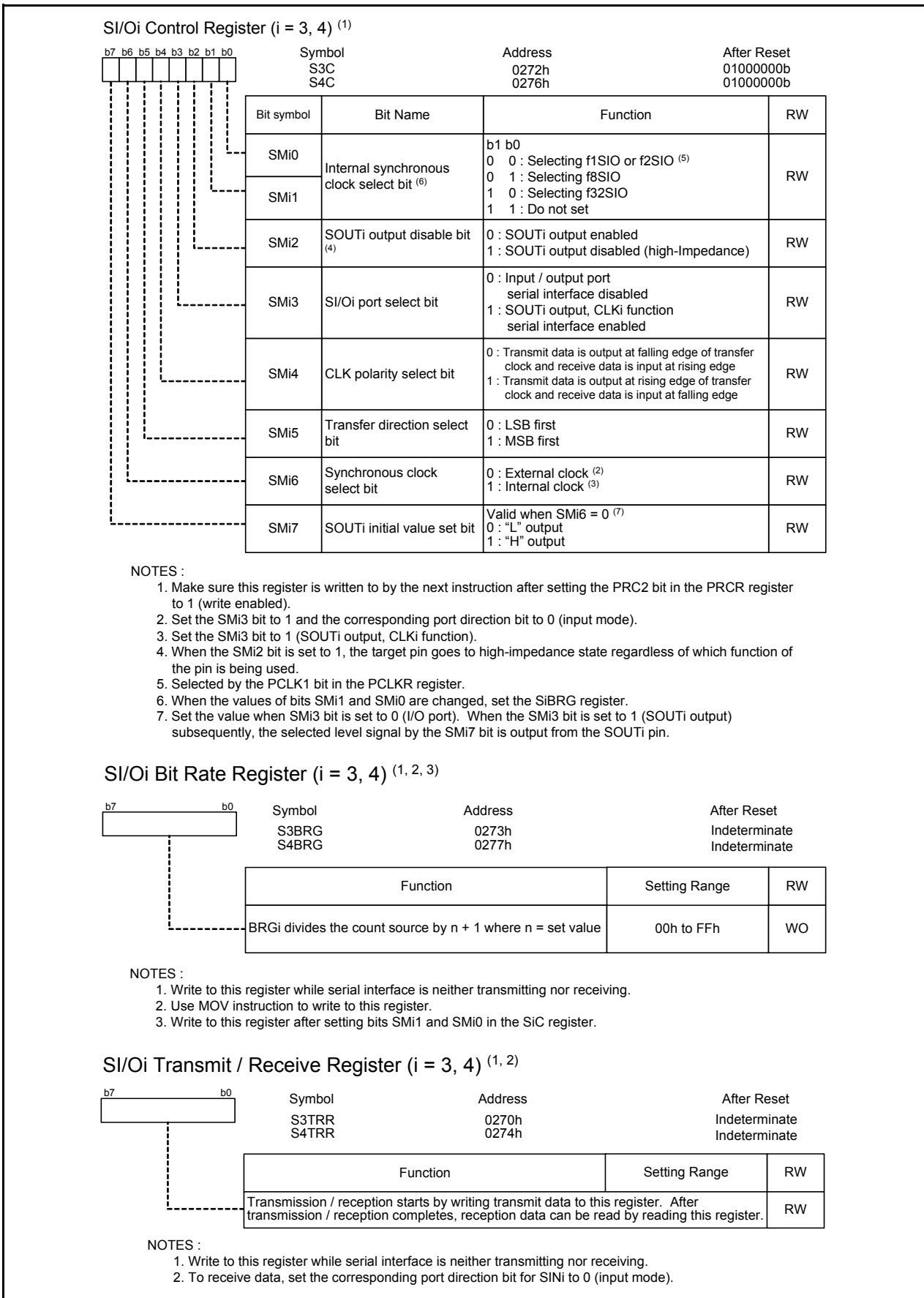
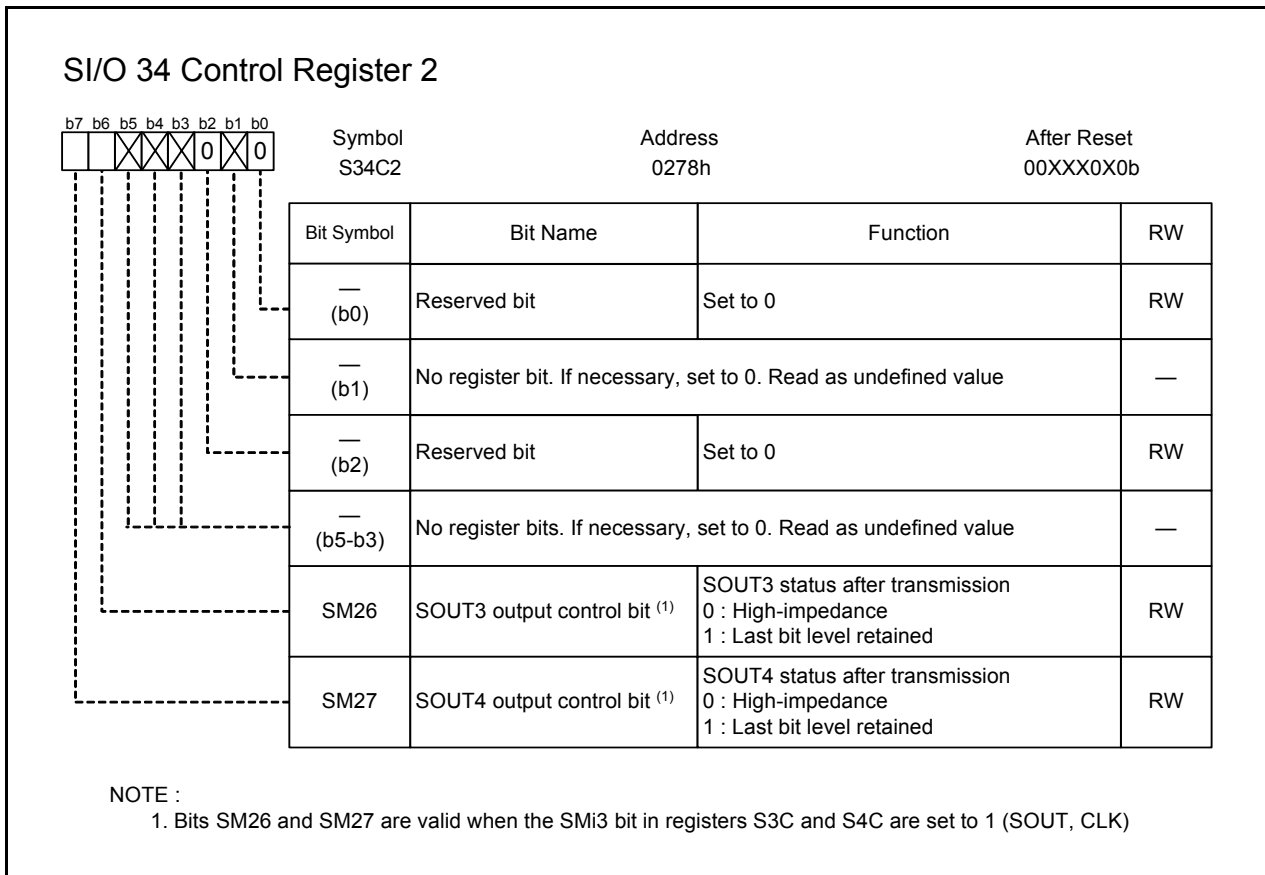


Figure 17.38 Registers S3C, S4C, S3BRG, S4BRG, S3TRR, and S4TRR



**Figure 17.39 S34C2 Register**

Table 17.20 SI/O3 and SI/O4 Specifications

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer Clock	<ul style="list-style-type: none"> <li>The SMi6 bit in the SiC (i = 3, 4) register = 1 (internal clock): <math>f_j / (2(n + 1))</math>  <math>f_j = f_{1SIO}, f_{8SIO}, f_{32SIO}</math>  <math>n =</math> setting value of the SiBRG register 00h to FFh</li> <li>The SMi6 bit = 0 (external clock): input from the CLKi pin <sup>(1)</sup></li> </ul>
Transmission / Reception Start Condition	Before transmission / reception starts, satisfy the following requirements Write transmit data to the SiTRR register <sup>(2, 3)</sup>
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>When the SMi4 bit in the SiC register = 0 The rising edge of the last transfer clock pulse <sup>(4)</sup></li> <li>When the SMi4 bit = 1 The falling edge of the last transfer clock pulse <sup>(4)</sup></li> </ul>
CLKi Pin Function	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	<ul style="list-style-type: none"> <li>LSB first or MSB first selection Whether to start sending / receiving data beginning with bit 0 or beginning with bit 7 can be selected</li> <li>CLK polarity selection Whether transmit data is output / input at the rising edge or falling edge of transfer clock can be selected.</li> <li>Function for setting an SOUTi initial value set function When the SMi6 bit in the SiC register = 0 (external clock), the SOUTi pin output level while not transmitting can be selected.</li> <li>SOUTi state selection after transmission Whether to set to high-impedance or retain the last bit level can be selected.</li> </ul>

## NOTES:

- To set the SMi6 bit in the SiC register to 0 (external clock), follow the procedure described below.
  - If the SMi4 bit in the SiC register = 0, write transmit data to the SiTRR register while input on the CLKi pin is high. The same applies when rewriting the SMi7 bit in the SiC register.
  - 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.
- 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.
- When the SMi6 bit = 1 (internal clock) and bits SM26 (SOUT3) and SM27 (SOUT4) in the S34C2 register = 0 (high-impedance after transmission), SOUTi retains the last data for a 1/2 transfer clock period after completion of transfer and, thereafter, goes to high-impedance state. However, if transmit data is written to the SiTRR register during this period, SOUTi immediately goes to high-impedance state, with the data hold time thereby reduced.
- When the 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.

### 17.2.1 SI/Oi Operation Timing

Figure 17.40 shows the SI/Oi Operation Timing

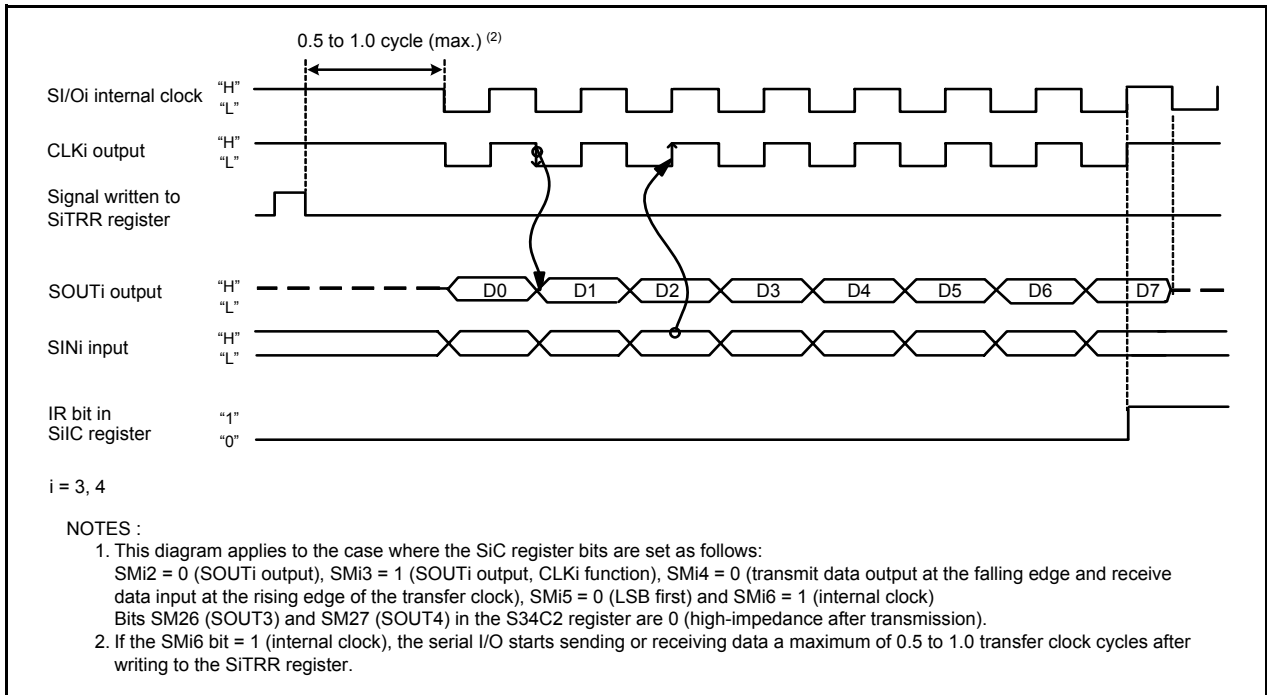


Figure 17.40 SI/Oi Operation Timing

### 17.2.2 CLK Polarity Selection

The SMi4 bit in the SiC register allows selection of the polarity of the transfer clock. Figure 17.41 shows the Polarity of Transfer Clock

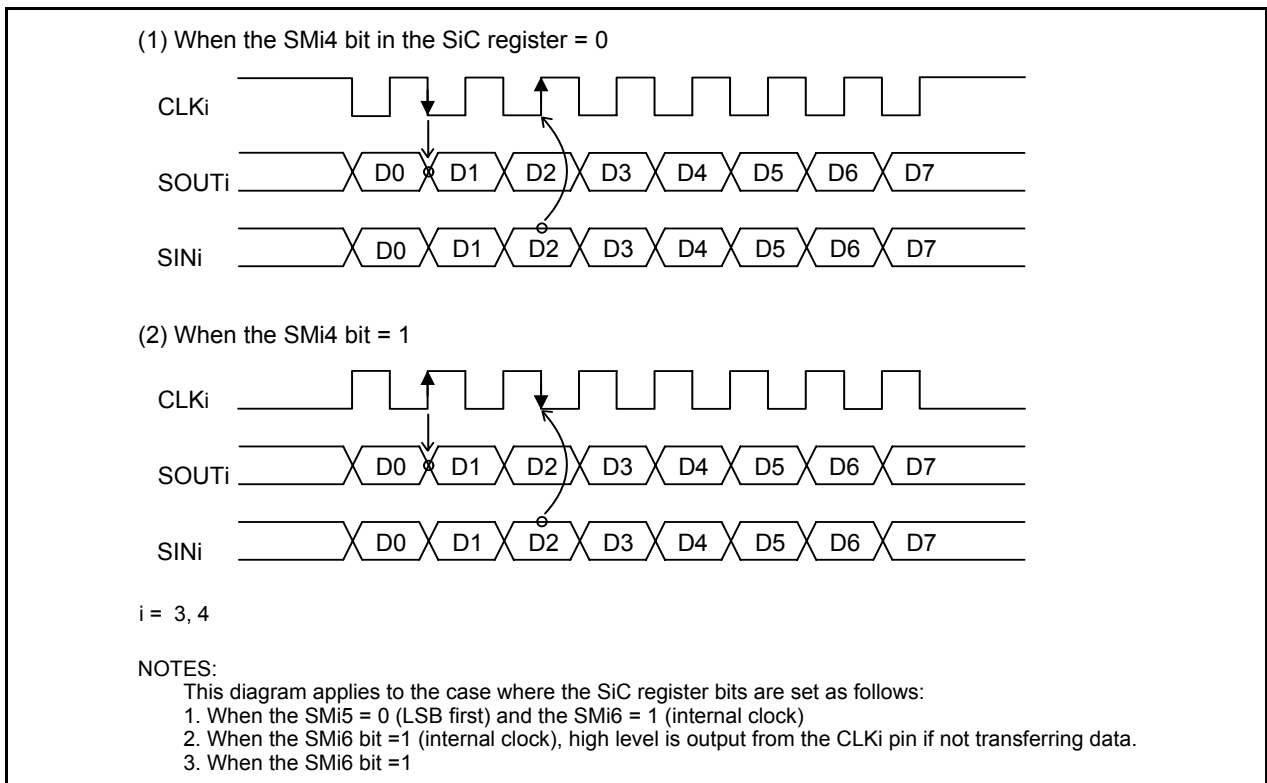


Figure 17.41 Polarity of Transfer Clock



### 17.2.3 Functions for Setting an SOUTi Initial Value

If the SMI6 bit in the SiC register = 0 (external clock), the SOUTi pin output can be fixed high or low when not transferring. However, the last bit value of the former data is retained between data and data when transmitting data consecutively. Figure 17.42 shows the timing chart for setting an SOUTi initial value and how to set it.

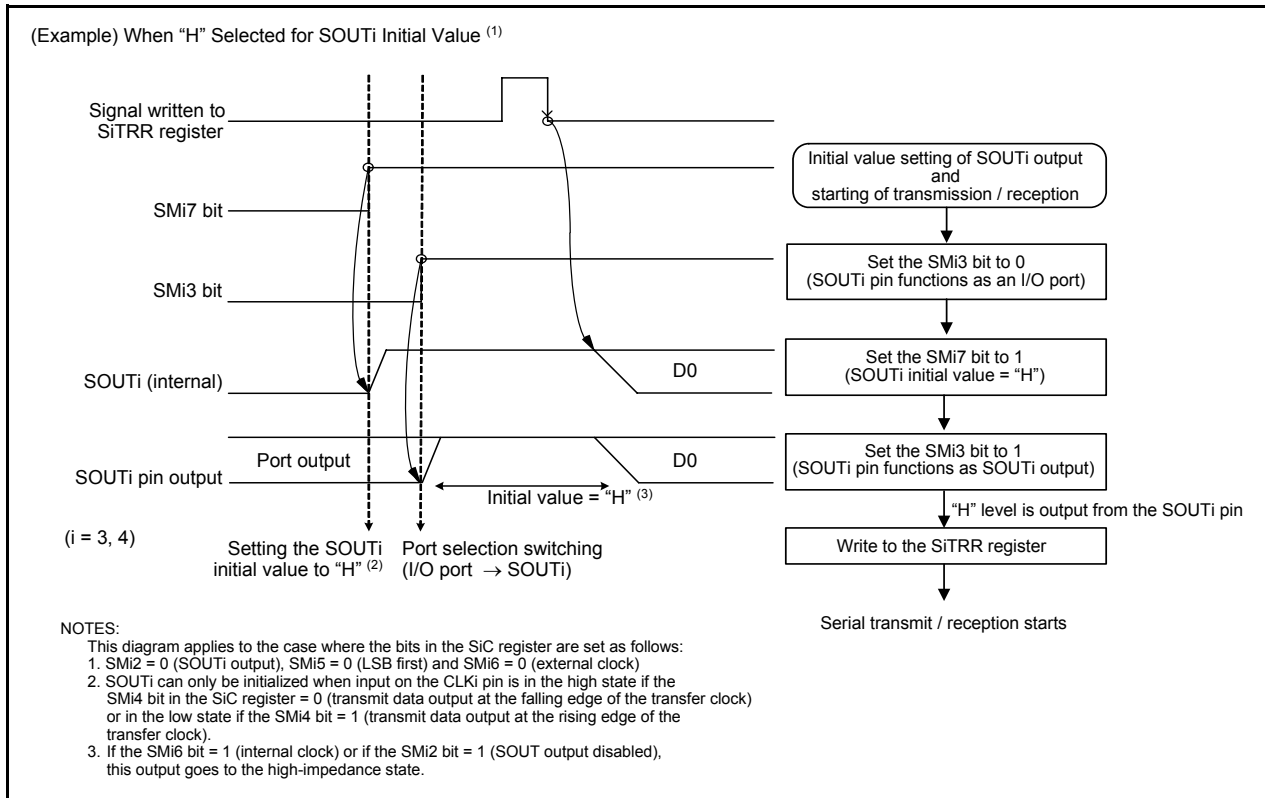


Figure 17.42 SOUTi Initial Value Setting

### 17.2.4 Functions for Selecting SOUTi State after Transmission

If bits SM26 and SM27 in the S34C2 register = 1 (last bit level retained), output from the SOUTi pin retains the last bit level after transmission. Figure 17.43 shows the Level of SOUT3 Pin after Transmission.

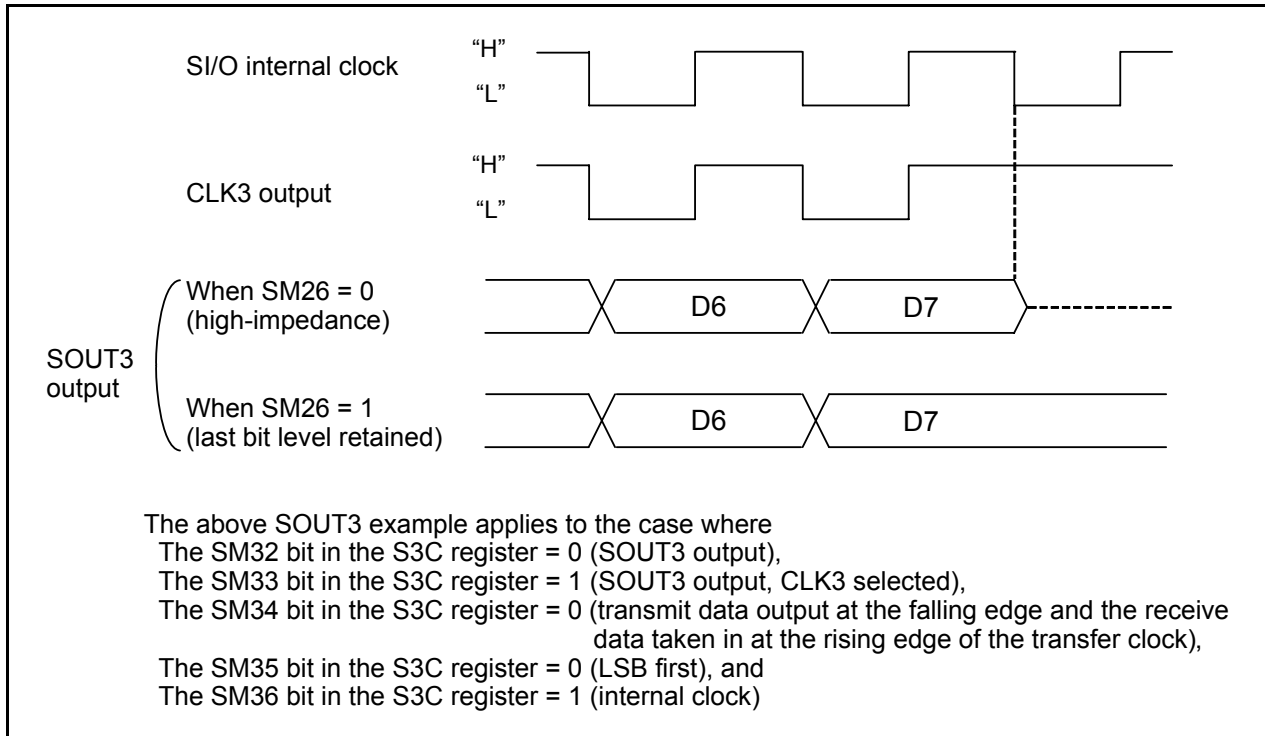


Figure 17.43 Level of SOUT3 Pin after Transmission

## 18. A/D Converter

The microcomputer contains one A/D converter circuit based on 10-bit successive approximation method. The analog inputs share the pins with P10\_0 to P10\_7, P9\_5, P9\_6, P0\_0 to P0\_7, and P2\_0 to P2\_7. Similarly,  $\overline{\text{ADTRG}}$  input shares the pin with P9\_7. Therefore, when using these inputs, make sure the corresponding port direction bits are set to 0 (input mode).

When not using the A/D converter, set the ADSTBY bit to 0 (A/D operation stop: standby), so that no current will flow for the A/D converter, helping to reduce the power consumption of the chip.

The A/D conversion result is stored in the ADi register for pins ANi, AN0\_i, and AN2\_i (i = 0 to 7).

Table 18.1 shows the A/D Converter Specifications. Figure 18.1 shows the A/D Converter Block Diagram, and Figures 18.2 and 18.3 show the A/D converter-related registers.

**Table 18.1 A/D Converter Specifications**

Item	Performance
A/D Conversion Method	Successive approximation
Analog Input Voltage (1)	0V to AVCC (VCC1)
Operating clock $\phi_{\text{AD}}$ (1)	fAD, divide-by-2 of fAD, divide-by-3 of fAD, divide-by-4 of fAD, divide-by-6 of fAD, divide-by-12 of fAD
Resolution	10-bit
Integral Nonlinearity Error	When AVCC = VREF = 5V AN0 to AN7 input, AN0_0 to AN0_7 input or AN2_0 to AN2_7 input: $\pm 3\text{LSB}$ ANEX0 or ANEX1 input: $\pm 3\text{LSB}$ When AVCC = VREF = 3.0V AN0 to AN7 input, AN0_0 to AN0_7 input or AN2_0 to AN2_7 input: $\pm 3\text{LSB}$ ANEX0 or ANEX1 input: $\pm 3\text{LSB}$
Operating Modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, repeat sweep mode 1
Analog Input Pins	8 pins (AN0 to AN7) + 2 pins (ANEX0 and ANEX1) + 8 pins (AN0_0 to AN0_7) + 8 pins (AN2_0 to AN2_7)
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>Software trigger                              The ADST bit in the ADCON0 register is set to 1 (A/D conversion start)</li> <li>External trigger (retriggerable)                              Input on the <math>\overline{\text{ADTRG}}</math> pin changes state from high to low after the ADST bit is set to 1 (A/D conversion start)</li> </ul>
Conversion Speed per Pin	43 $\phi_{\text{AD}}$ cycles minimum

**NOTES:**

- Set  $\phi_{\text{AD}}$  frequency as follows:  
 When VCC1 = 4.0 to 5.5 V,  $2\text{ MHz} \leq \phi_{\text{AD}} \leq 25\text{ MHz}$   
 When VCC1 = 3.2 to 4.0 V,  $2\text{ MHz} \leq \phi_{\text{AD}} \leq 16\text{ MHz}$   
 When VCC1 = 3.0 to 3.2 V,  $2\text{ MHz} \leq \phi_{\text{AD}} \leq 10\text{ MHz}$

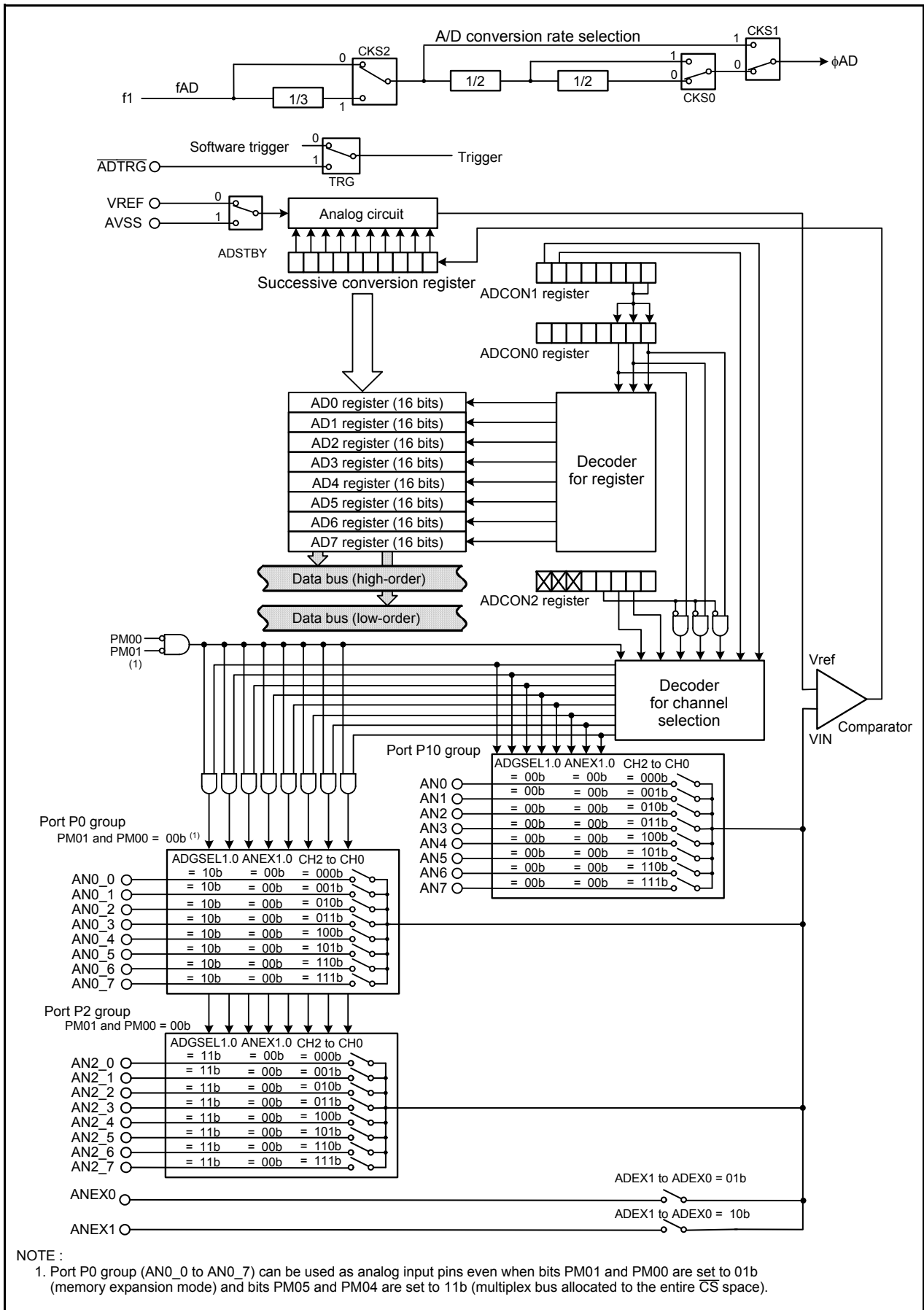


Figure 18.1 A/D Converter Block Diagram

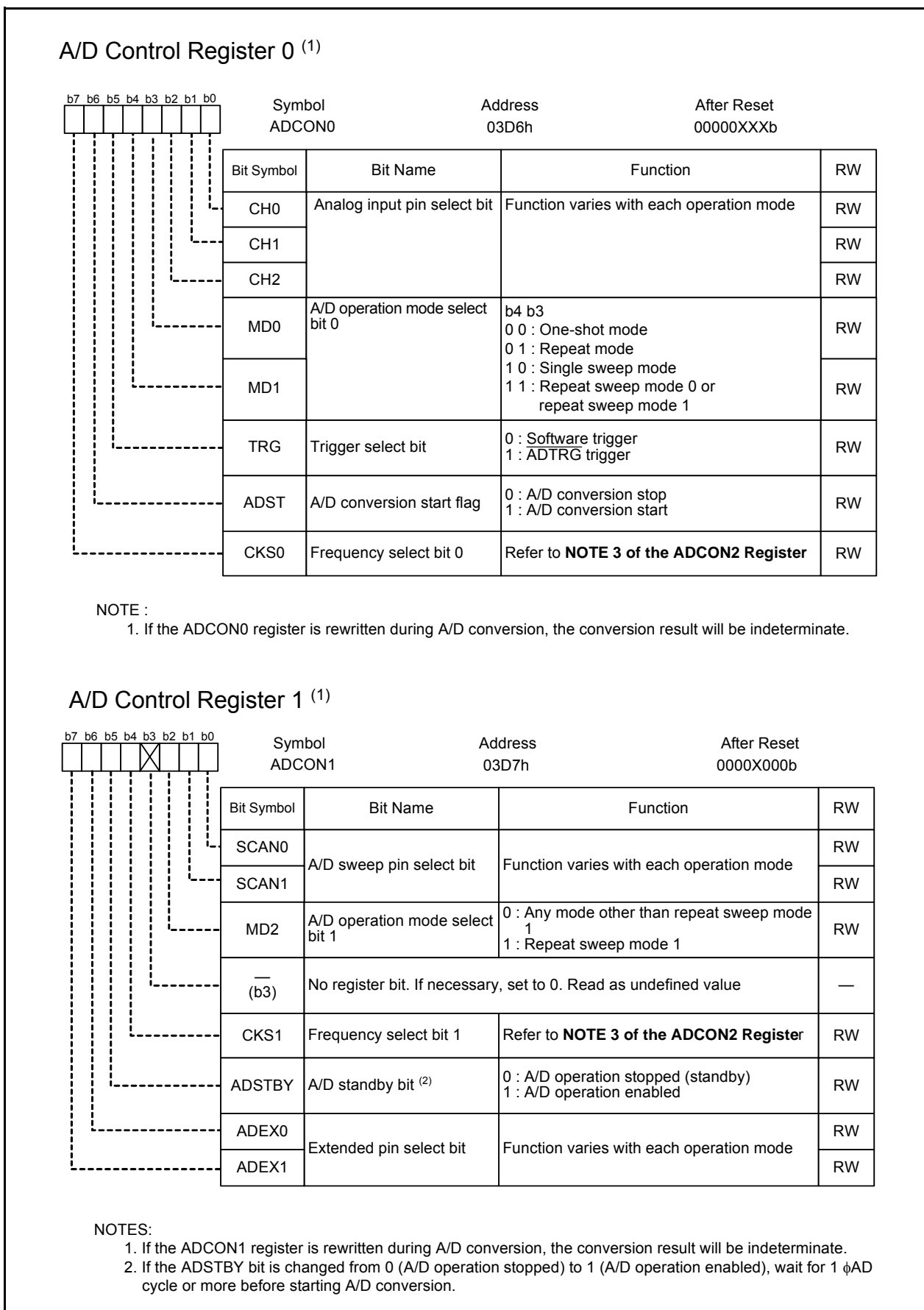


Figure 18.2 Registers ADCON0 and ADCON1

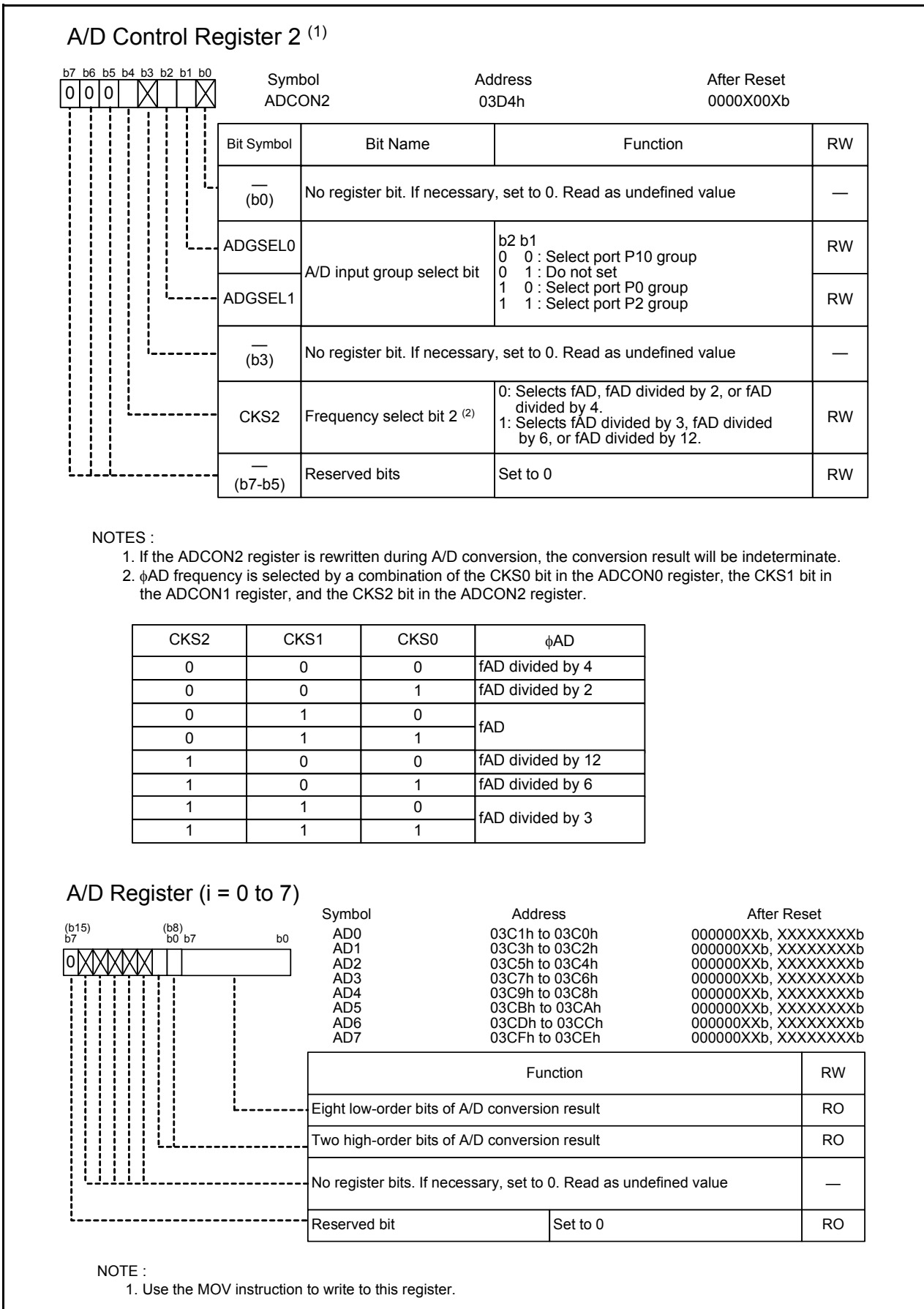


Figure 18.3 Registers ADCON2 and AD0 to AD7

## 18.1 Mode Description

### 18.1.1 One-Shot Mode

In one-shot mode, analog voltage applied to a selected pin is converted to a digital code once. Table 18.2 shows the One-Shot Mode Specifications. Figure 18.4 shows the Registers ADCON0 and ADCON1 (One-shot Mode).

**Table 18.2 One-Shot Mode Specifications**

Item	Specification
Function	Bits CH2 to CH0 in the ADCON0 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register, or bits ADEX1 and ADEX0 in the ADCON1 register select a pin. Analog voltage applied to the pin is converted to a digital code once.
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>• When the TRG bit in the ADCON0 register is 0 (software trigger) the ADST bit in the ADCON0 register is set to 1 (A/D conversion starts)</li> <li>• When the TRG bit is 1 (<math>\overline{\text{ADTRG}}</math> trigger) input on the <math>\overline{\text{ADTRG}}</math> pin changes state from high to low after the ADST bit is set to 1 (A/D conversion start)</li> </ul>
A/D Conversion Stop Condition	<ul style="list-style-type: none"> <li>• Completion of A/D conversion (The ADST bit is cleared to 0 (A/D conversion stop))</li> <li>• Set the ADST bit to 0</li> </ul>
Interrupt Request Generation Timing	Completion of A/D conversion
Analog Input Pin	Select one pin from AN0 to AN7, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0, and ANEX1
Reading of Result of A/D Converter	Read one of the registers AD0 to AD7 that corresponds to the selected pin

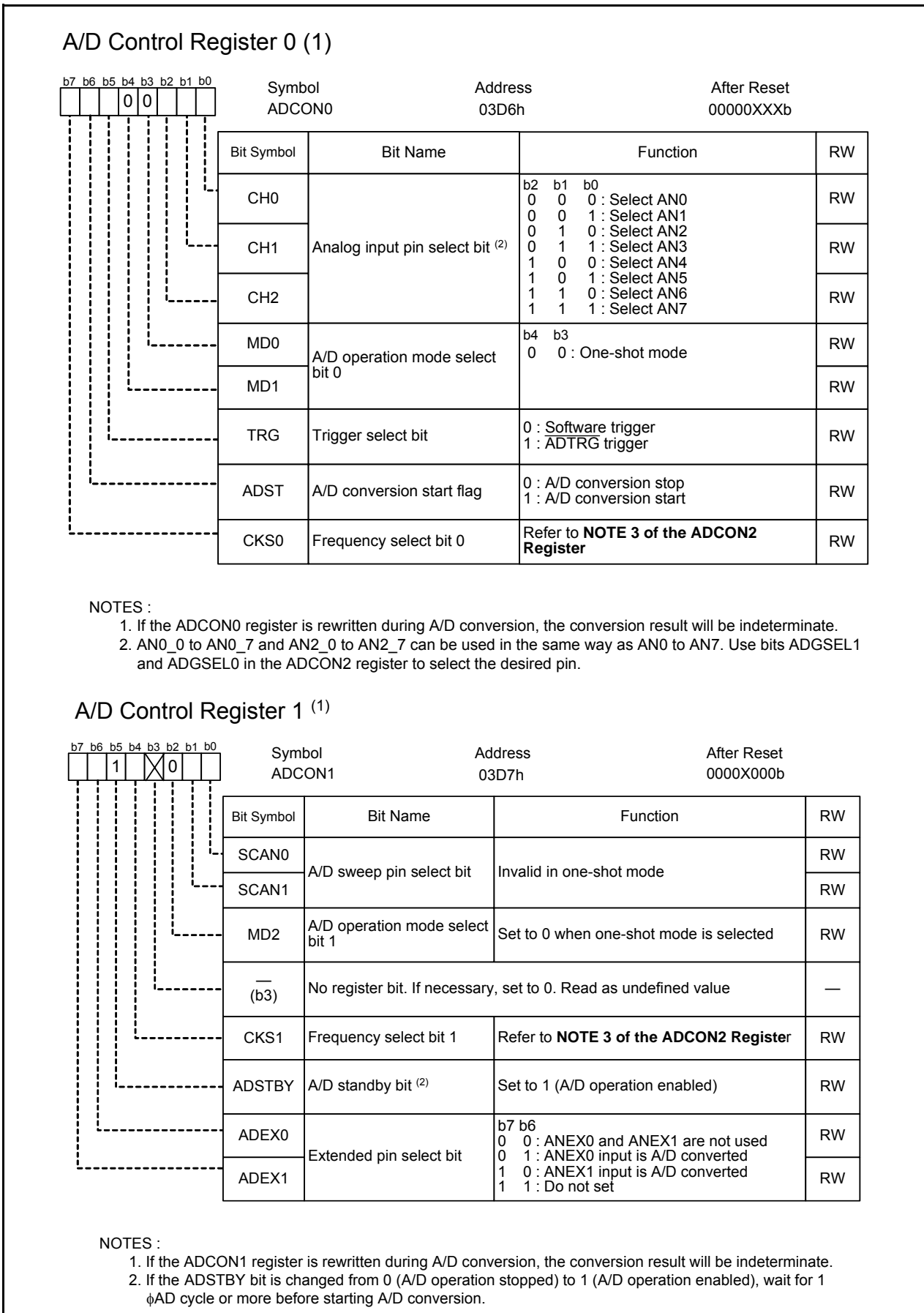


Figure 18.4 Registers ADCON0 and ADCON1 (One-shot Mode)



### 18.1.2 Repeat Mode

In repeat mode, analog voltage applied to a selected pin is repeatedly converted to a digital code. Table 18.3 shows the Repeat Mode Specifications. Figure 18.5 shows the Registers ADCON0 and ADCON1 (Repeat Mode).

**Table 18.3 Repeat Mode Specifications**

Item	Specification
Function	Bits CH2 to CH0 in the ADCON0 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register, or bits ADEX1 and ADEX0 in the ADCON1 register select a pin. Analog voltage applied to this pin is repeatedly converted to a digital code.
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>• When the TRG bit in the ADCON0 register is 0 (software trigger) the ADST bit in the ADCON0 register is set to 1 (A/D conversion start)</li> <li>• When the TRG bit is 1 (<math>\overline{\text{ADTRG}}</math> trigger) input on the <math>\overline{\text{ADTRG}}</math> pin changes state from high to low after the ADST bit is set to 1 (A/D conversion start)</li> </ul>
A/D Conversion Stop Condition	Set the ADST bit to 0 (A/D conversion stop)
Interrupt Request Generation timing	No interrupt requests generated
Analog Input Pin	Select one pin from AN0 to AN7, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0, and ANEX1
Reading of Result of A/D Converter	Read one of the registers AD0 to AD7 that corresponds to the selected pin

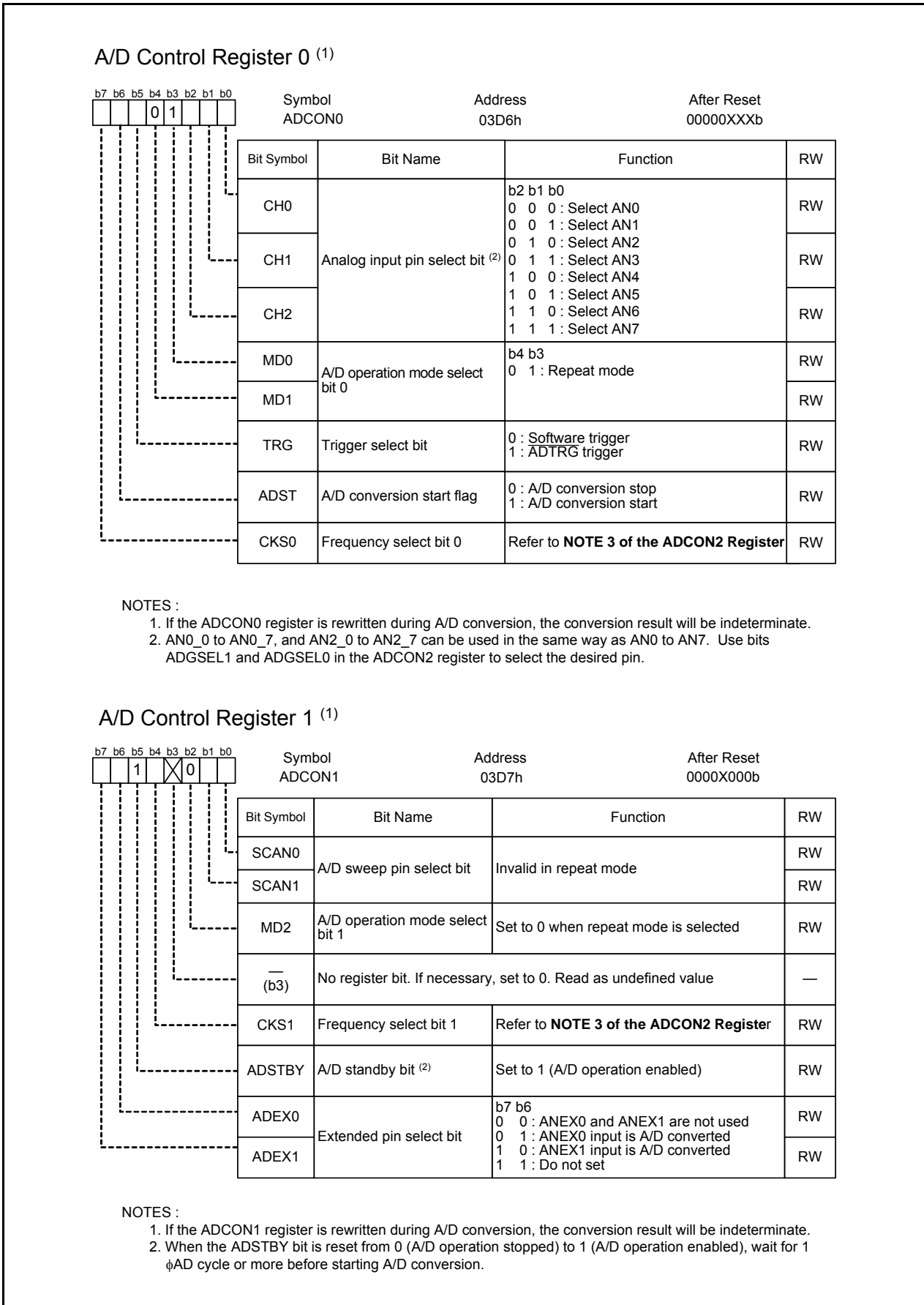


Figure 18.5 Registers ADCON0 and ADCON1 (Repeat Mode)

### 18.1.3 Single Sweep Mode

In single sweep mode, analog voltage that is applied to selected pins is converted one-by-one to a digital code. Table 18.4 shows the Single Sweep Mode Specifications. Figure 18.6 shows Registers ADCON0 and ADCON1 (Single Sweep Mode).

**Table 18.4 Single Sweep Mode Specifications**

Item	Specification
Function	Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the pins is converted one-by-one to a digital code.
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>• When the TRG bit in the ADCON0 register is 0 (software trigger) the ADST bit in the ADCON0 register is set to 1 (A/D conversion start)</li> <li>• When the TRG bit is 1 (<math>\overline{\text{ADTRG}}</math> trigger) input on the <math>\overline{\text{ADTRG}}</math> pin changes state from high to low after the ADST bit is set to 1 (A/D conversion start)</li> </ul>
A/D Conversion Stop Condition	<ul style="list-style-type: none"> <li>• Completion of A/D conversion (The ADST bit is cleared to 0 (A/D conversion stop))</li> <li>• Set the ADST bit to 0</li> </ul>
Interrupt Request Generation timing	Completion of A/D conversion
Analog Input Pin	Select from AN0 and AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), and AN0 to AN7 (8 pin) <sup>(1)</sup>
Reading of Result of A/D Converter	Read one of the registers AD0 to AD7 that corresponds to the selected pin

NOTE:

1. AN0\_0 to AN0\_7 and AN2\_0 to AN2\_7 can be used in the same way as AN0 to AN7.

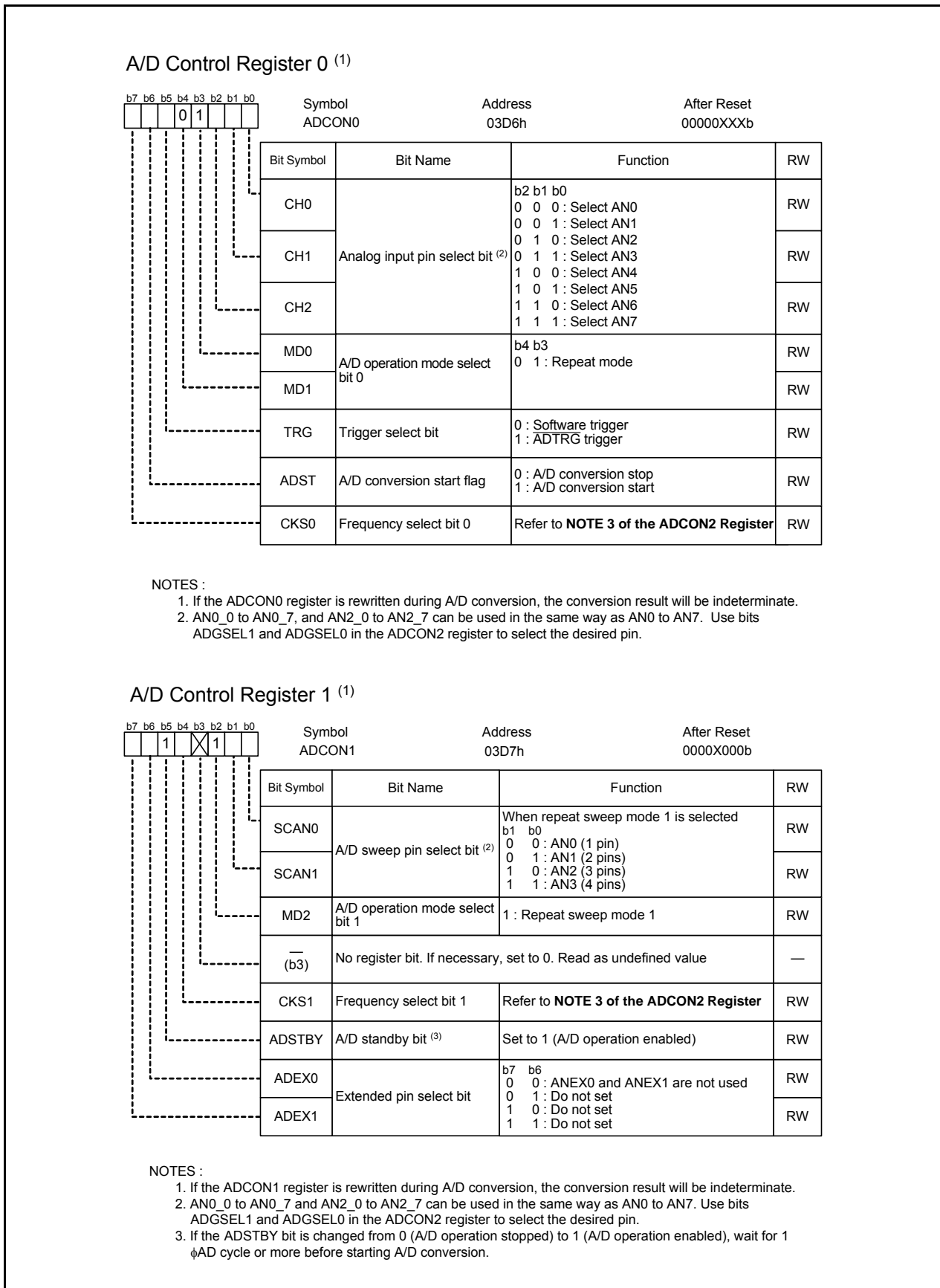


Figure 18.6 Registers ADCON0 and ADCON1 (Single Sweep Mode)

### 18.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, analog voltage applied to selected pins is repeatedly converted to a digital code.

Table 18.5 shows the Repeat Sweep Mode 0 Specifications. Figure 18.7 shows Registers ADCON0 and ADCON1 (Repeat Sweep Mode 0).

**Table 18.5 Repeat Sweep Mode 0 Specifications**

Item	Specification
Function	Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the pins is repeatedly converted to a digital code.
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>• When the TRG bit in the ADCON0 register is 0 (software trigger) the ADST bit in the ADCON0 register is set to 1 (A/D conversion start)</li> <li>• When the TRG bit is 1 (<math>\overline{\text{ADTRG}}</math> trigger) input on the <math>\overline{\text{ADTRG}}</math> pin changes state from high to low after the ADST bit is set to 1 (A/D conversion start)</li> </ul>
A/D Conversion Stop Condition	Set the ADST bit to 0 (A/D conversion stop)
Interrupt Request Generation timing	No interrupt requests generated
Analog Input Pin	Select from AN0 and AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), and AN0 to AN7 (8 pin) <sup>(1)</sup>
Reading of Result of A/D Converter	Read one of the registers AD0 to AD7 that corresponds to the selected pin

NOTE:

1. AN0\_0 to AN0\_7 and AN2\_0 to AN2\_7 can be used in the same way as AN0 to AN7.

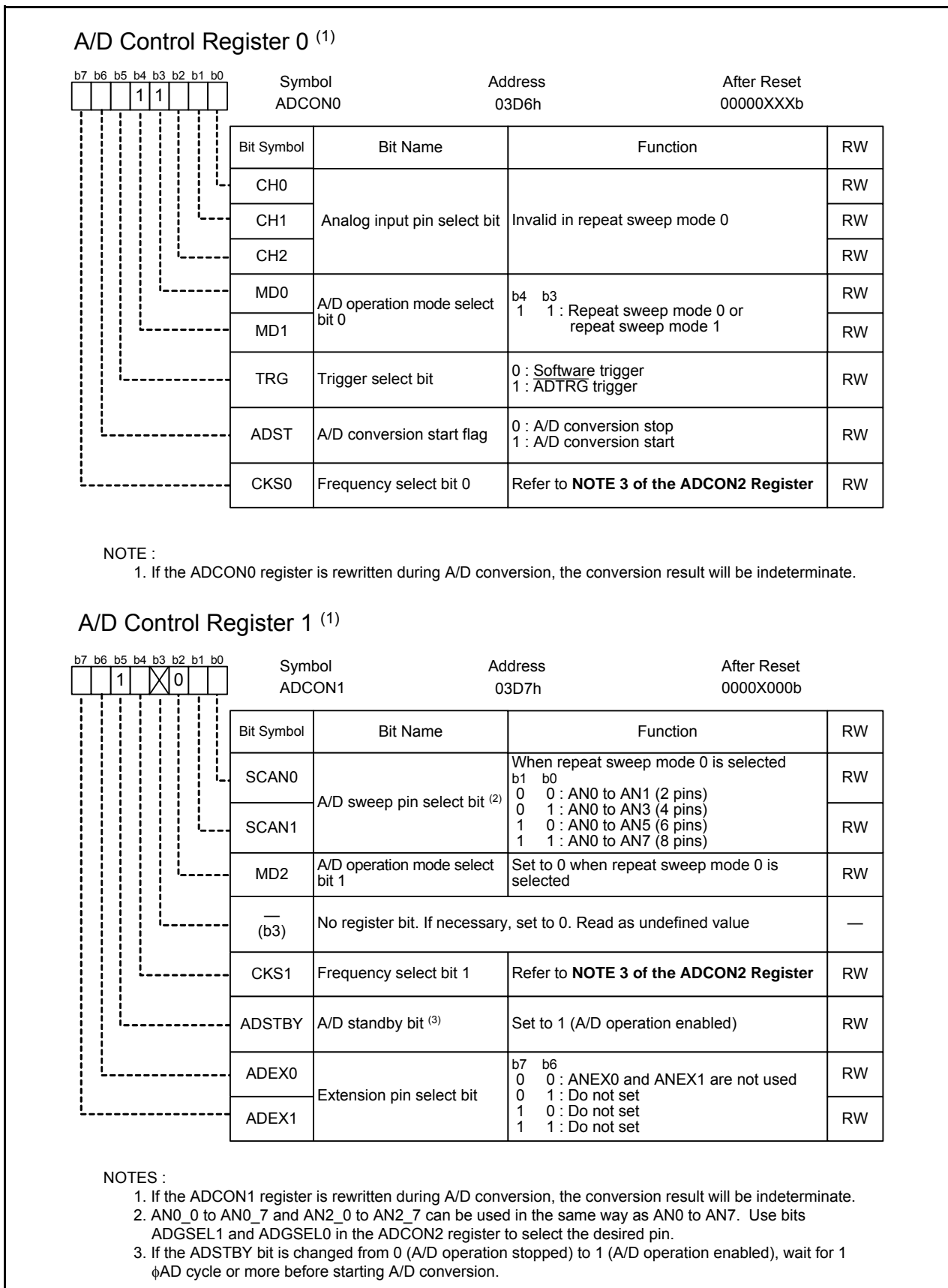


Figure 18.7 Registers ADCON0 and ADCON1 (Repeat Sweep Mode 0)

### 18.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltage selectively applied to all pins is repeatedly converted to a digital code. Table 18.6 shows the Repeat Sweep Mode 1 Specifications. Figure 18.8 shows Registers ADCON0 and ADCON1 (Repeat Sweep Mode 1).

**Table 18.6 Repeat Sweep Mode 1 Specifications**

Item	Specification
Function	The input voltages on all pins selected by bits ADGSEL1 and ADGSEL0 in the ADCON2 register are A/D converted repeatedly, with priority given to pins selected by bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0. Example: If AN0 selected, input voltages are A/D converted in order of AN0→AN1→AN0→AN2→AN0→AN3, and so on.
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>When the TRG bit in the ADCON0 register is 0 (software trigger), the ADST bit in the ADCON0 register is set to 1 (A/D conversion start)</li> <li>When the TRG bit is 1 (ADTRG trigger), input on the <math>\overline{\text{ADTRG}}</math> pin changes state from high to low after the ADST bit is set to 1 (A/D conversion start)</li> </ul>
A/D Conversion Stop Condition	Set the ADST bit to 0 (A/D conversion stop)
Interrupt Request Generation timing	No interrupt requests generated
Analog Input Pins to be Given Priority When A/D Converted	Select from AN0 (1 pin), AN0 and AN1 (2 pins), AN0 to AN2 (3 pins), and AN0 to AN3 (4 pins) <sup>(1)</sup>
Reading of Result of A/D Converter	Read one of the registers AD0 to AD7 that corresponds to the selected pin

NOTES:

- AN0\_0 to AN0\_7 and AN2\_0 to AN2\_7 can be used in the same way as AN0 to AN7.

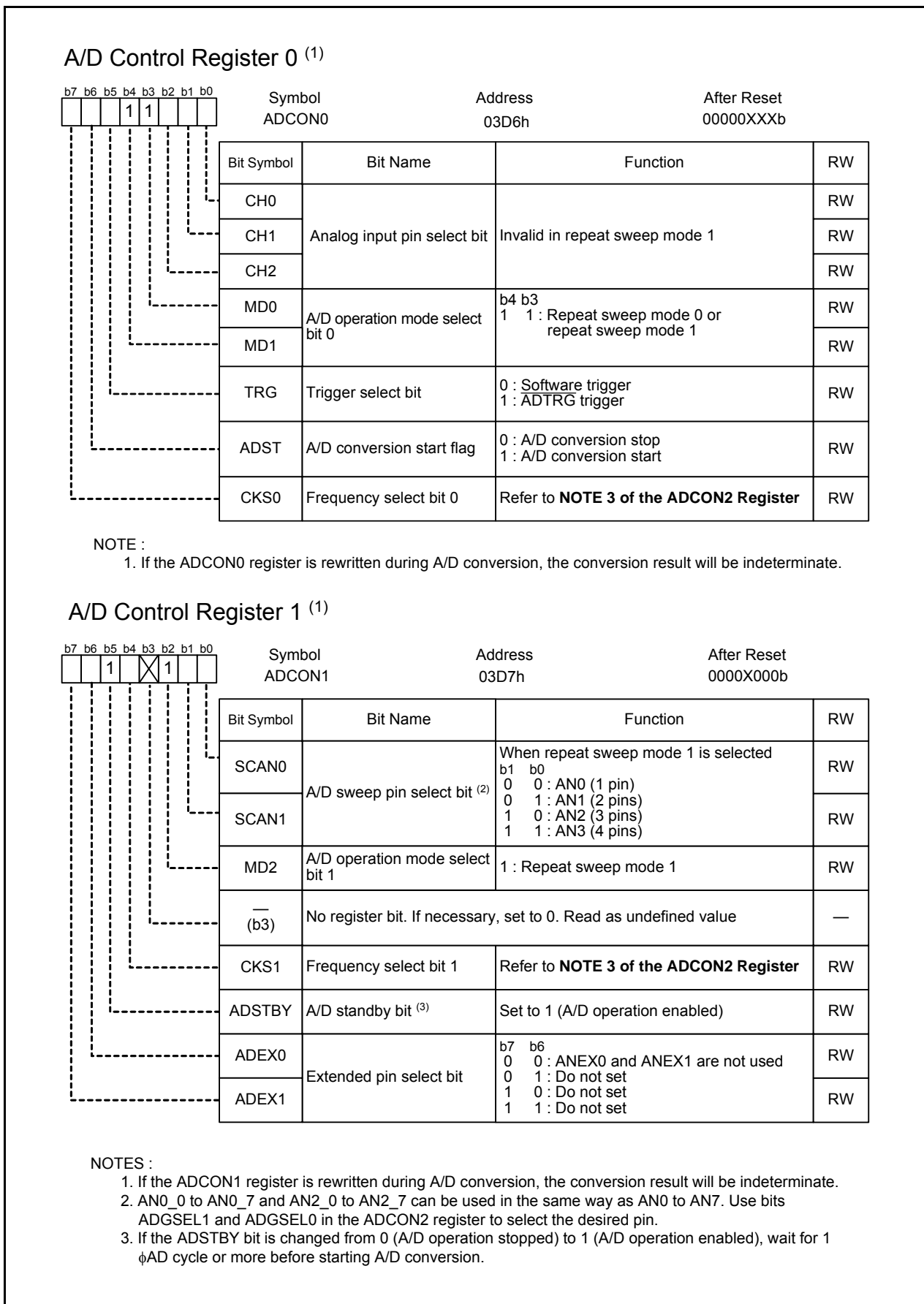


Figure 18.8 Registers ADCON0 and ADCON1 (Repeat Sweep Mode 1)



## 18.2 Conversion Rate

The conversion rate is defined as follows.

Start dummy time depends on which  $\phi_{AD}$  is selected. Table 18.7 shows Start Dummy Time. When the ADST bit in the ADCON0 register is set to 1 (A/D conversion start), A/D conversion starts after start dummy time elapses. 0 (A/D conversion stop) is read if the ADST bit is read before A/D conversion starts. For multiple pins or A/D conversion repeat mode, for each pin, between-execution dummy time is inserted between A/D conversion execution time and the next A/D conversion execution time.

The ADST bit is set to 0 during the end dummy time, and the last A/D conversion result is set to the ADI register in one-shot mode and single sweep mode.

While in one-shot mode:

Start dummy time + A/D conversion execution time + end dummy time

When two pins are selected while in single sweep mode:

Start dummy time + (A/D conversion execution time + between-execution dummy time + A/D conversion execution time) + end dummy time

Start dummy time: See **Table 18.7 “Start Dummy Time”**

A/D conversion execution time: 40  $\phi_{AD}$  cycles per pin

Between-execution dummy time: 1  $\phi_{AD}$  cycle

End dummy time: 2 to 3 cycles of  $f_{AD}$

**Table 18.7 Start Dummy Time**

$\phi_{AD}$ Selection	Start Dummy Time
$f_{AD}$	1 to 2 cycles of $f_{AD}$
$f_{AD}$ divided by 2	2 to 3 cycles of $f_{AD}$
$f_{AD}$ divided by 3	3 to 4 cycles of $f_{AD}$
$f_{AD}$ divided by 4	3 to 4 cycles of $f_{AD}$
$f_{AD}$ divided by 6	4 to 5 cycles of $f_{AD}$
$f_{AD}$ divided by 12	7 to 8 cycles of $f_{AD}$

## 18.3 Extended Analog Input Pins

In one-shot and repeat modes, pins ANEX0 and ANEX1 can be used as analog input pins. Use bits ADEX1 and ADEX0 in the ADCON1 register to select whether or not to use ANEX0 and ANEX1.

The A/D conversion results of ANEX0 and ANEX1 inputs are stored in registers AD0 and AD1, respectively.

## 18.4 Current Consumption Reducing Function

When not using the A/D converter, power consumption can be reduced by setting the ADSTBY bit in the ADCON1 register to 0 (A/D operation stopped: standby) to shut off any analog circuit current flow.

To use the A/D converter, set the ADSTBY bit to 1 (A/D operation enabled) after operating longer than one cycle of a timer count source, and then set the ADST bit in the ADCON0 register to 1 (A/D conversion start). Do not set bits ADST and ADSTBY to 1 at the same time.

Also, do not set the ADSTBY bit to 0 (A/D operation stopped: standby) during A/D conversion.

### 18.5 Output Impedance of Sensor under A/D Conversion

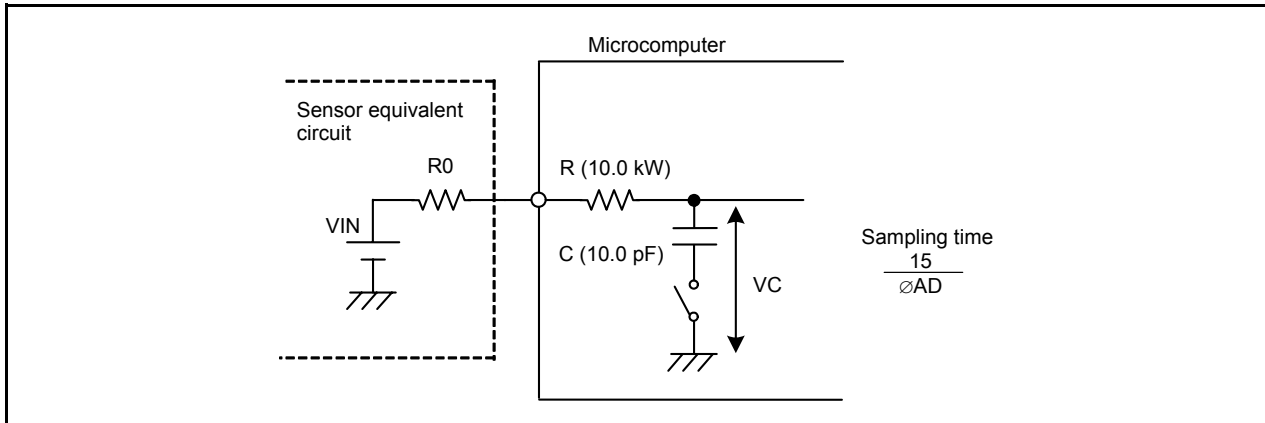


Figure 18.9 Analog Input Pin and External Sensor Equivalent Circuit

## 19. D/A Converter

### 19.1 Summary

The D/A converter consists of two independent 8-bit R-2R type D/A converters.

D/A conversion is performed by writing to the DA<sub>i</sub> register (i = 0 to 1). To output the result of conversion, set the DA<sub>i</sub>E bit in the DACON register to 1 (output enabled). Before using D/A conversion, clear the corresponding port direction bit to 0 (input mode). When the DA<sub>i</sub>E bit is set to 1 (input enabled), pull-up of a corresponding port is disabled.

Output analog voltage (V) is determined by a set value (n: decimal) in the DA<sub>i</sub> register.

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

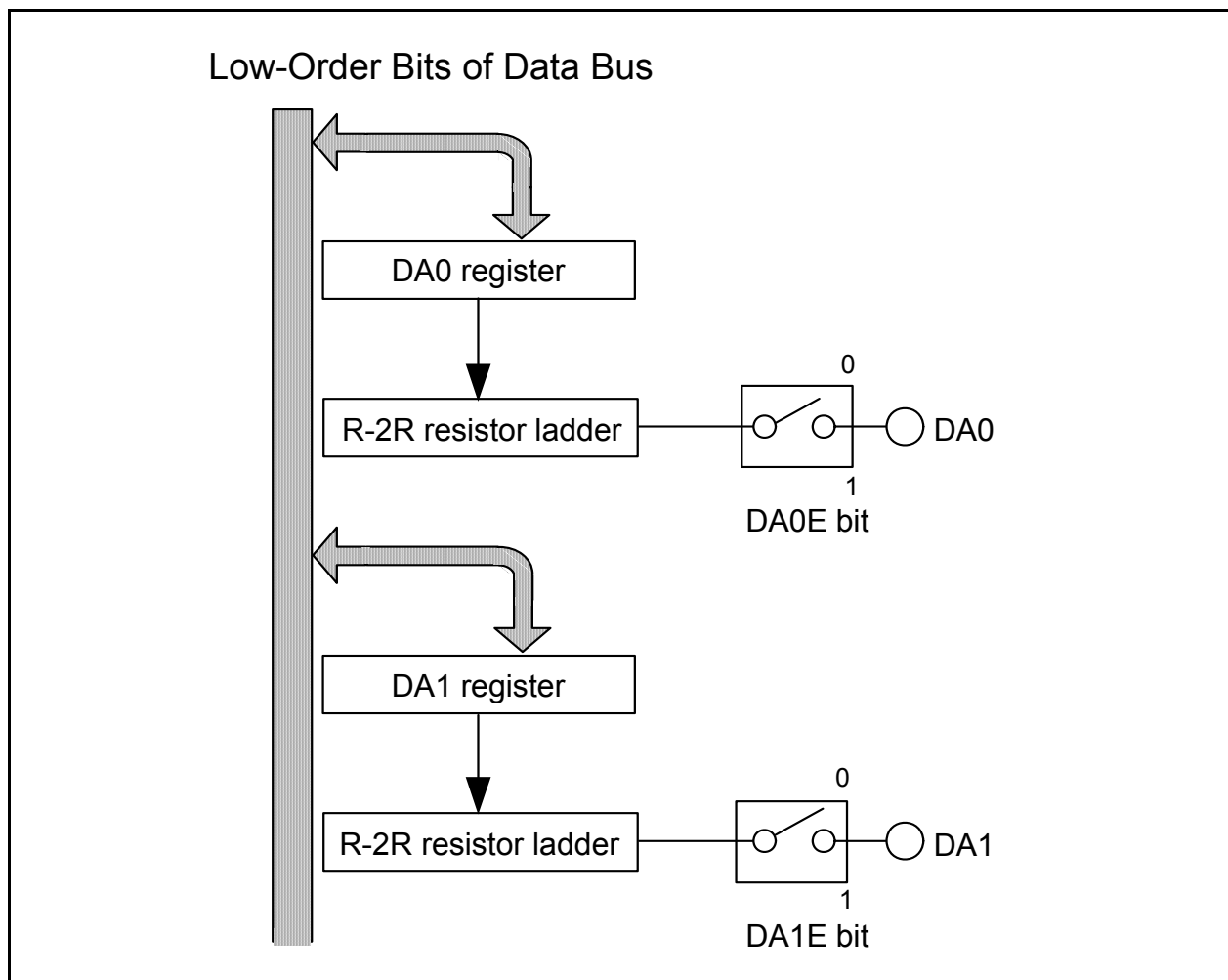
V<sub>REF</sub>: reference voltage

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

Figure 19.2 shows Registers DACON, DA<sub>0</sub>, and DA<sub>1</sub>. Figure 19.3 shows the D/A Converter Equivalent Circuit.

**Table 19.1 D/A Converter Performance**

Item	Performance
D/A Conversion Method	R-2R
Resolution	8 bits
Analog Output Pin	2 channels (DA <sub>0</sub> and DA <sub>1</sub> )



**Figure 19.1 D/A Converter Block Diagram**

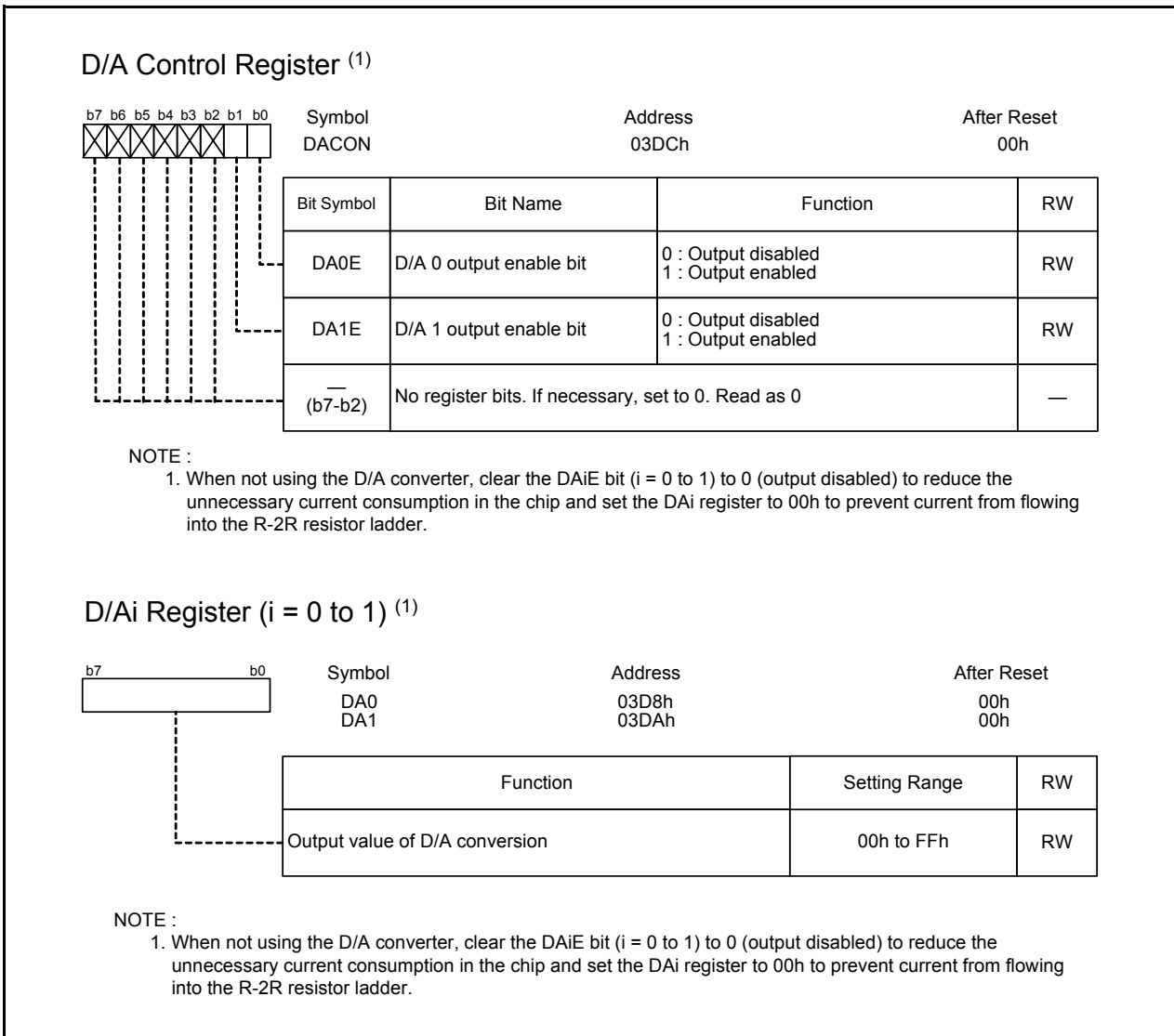


Figure 19.2 Registers DACON, DA0, and DA1

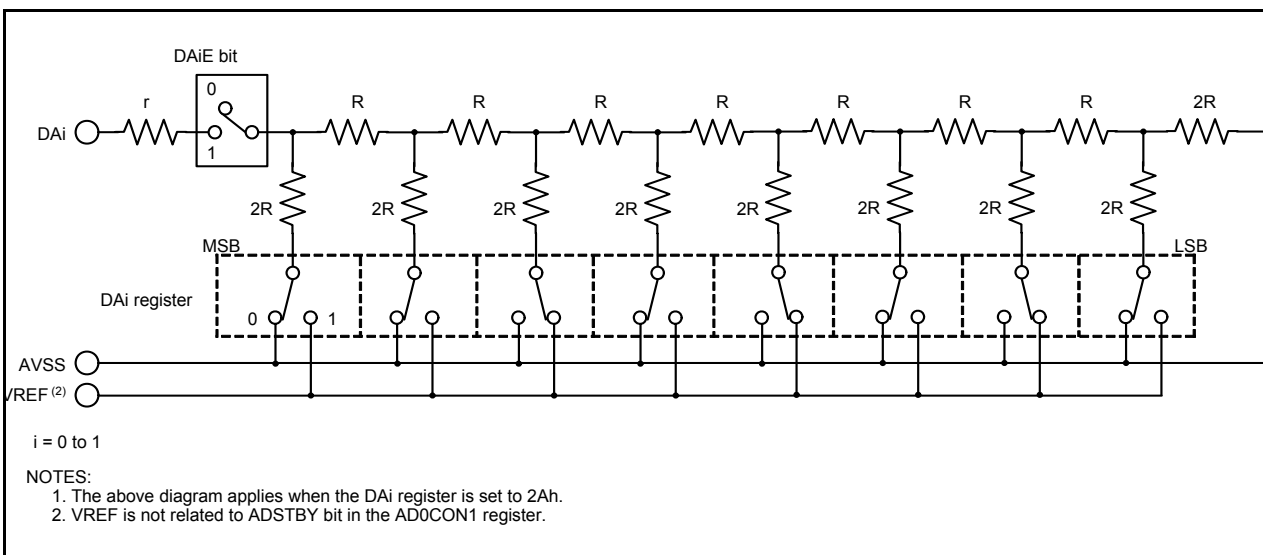


Figure 19.3 D/A Converter Equivalent Circuit

## 20. CRC Operation

The Cyclic Redundancy Check (CRC) operation detects an error in data blocks. The microcomputer uses a generator polynomial of CRC\_CCITT ( $X^{16} + X^{12} + X^5 + 1$ ) to generate CRC code.

The CRC code consists of 16 bits which are generated for each data block in given length, separated in 8 bit units. After the initial value is set in the CRCD register, the CRC code is set in that register each time one byte of data is written to the CRCIN register. CRC code generation for one-byte data is finished in two cycles.

Figure 20.1 shows the CRC Circuit Block Diagram. Figure 20.2 shows Registers CRCD and CRCIN.

Figure 20.3 shows an example using the CRC Operation.

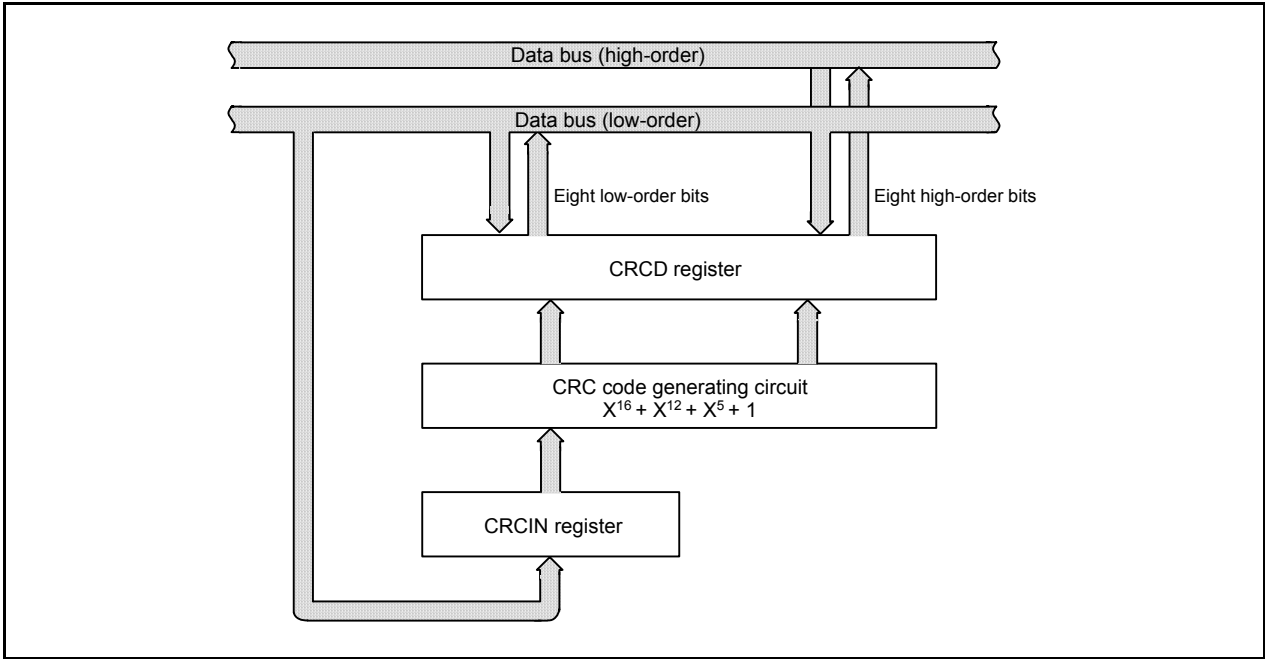


Figure 20.1 CRC Circuit Block Diagram

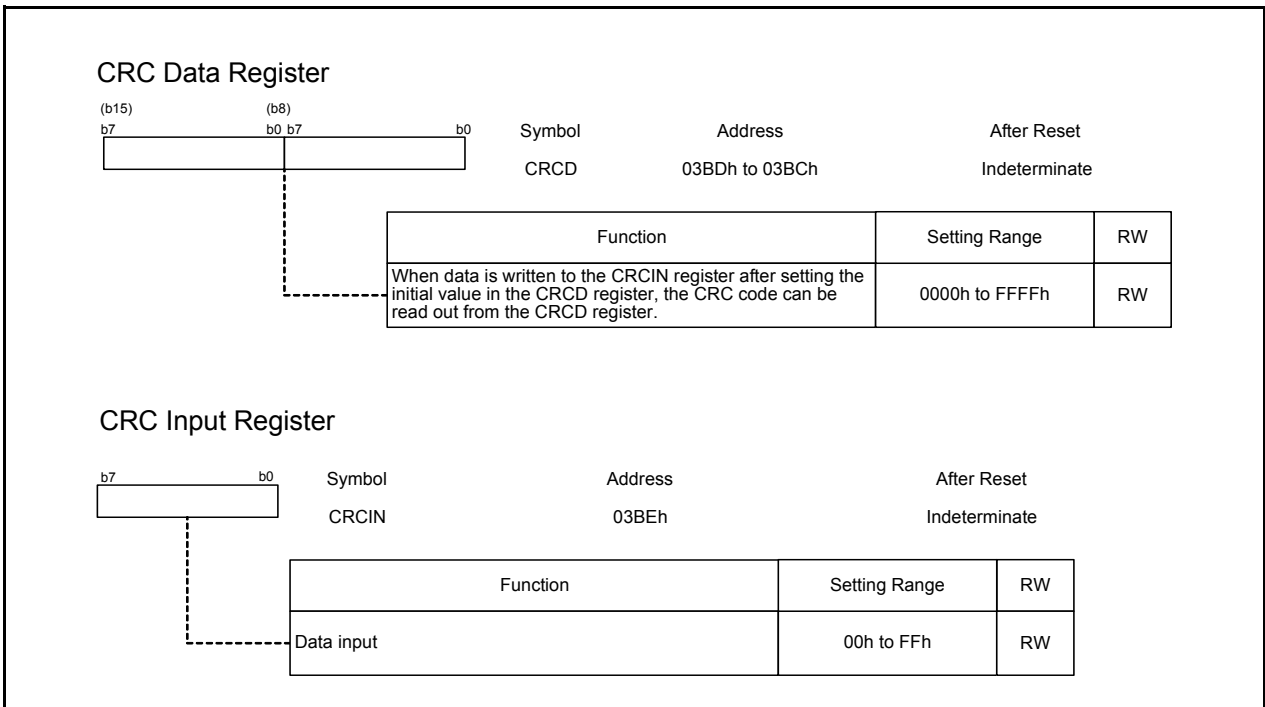


Figure 20.2 Registers CRCD and CRCIN

**Setup procedure and CRC operation when generating CRC code "80C4h"**

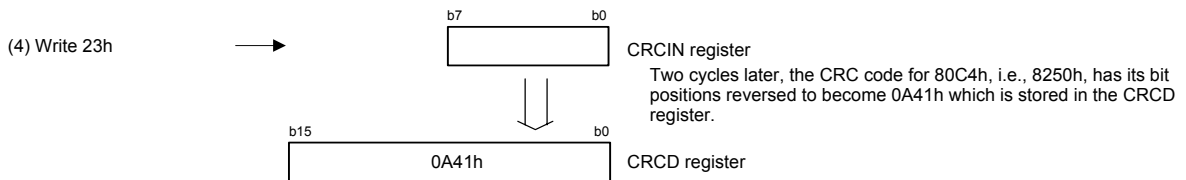
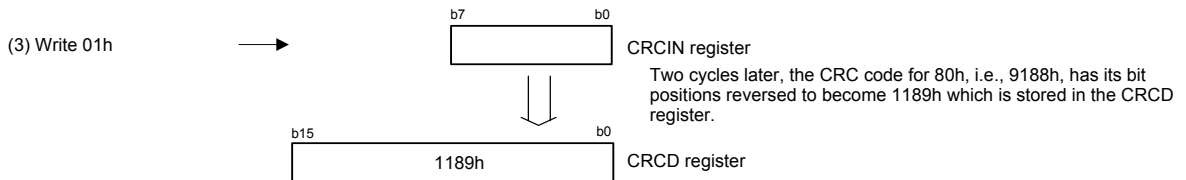
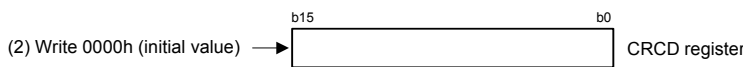
- CRC operation performed by the M16C

CRC code : remainder of a division in which the value written to the CRCIN register with its bit positions reversed is divided by the generator polynomial  
 Generator polynomial:  $X^{16} + X^{12} + X^5 + 1$  (1 0001 0000 0010 0001b)

- Setting procedure

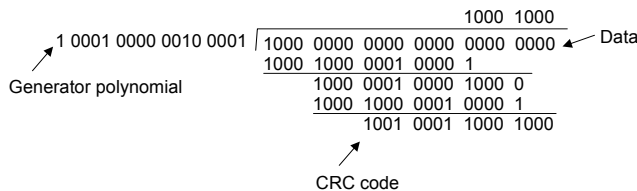
(1) Reverse the bit positions of the value "80C4h" by program in 1-byte units.

80h → 01h, C4h → 23h



- Details of CRC operation

As shown in (3) above, bit position of 01h (00000001b) written to the CRCIN register is reversed and becomes 10000000b. Add 1000 0000 0000 0000 0000 0000b, as 10000000b plus 16 digits, to 0000 0000 0000 0000 0000 0000b, as 0000 0000 0000 0000b plus 8 digits as the default value of the CRCD register to perform the modulo-2 division.



Modulo-2 operation is operation that complies with the law given below.

0 + 0 = 0  
 0 + 1 = 1  
 1 + 0 = 1  
 1 + 1 = 0  
 - 1 = 1

0001 0001 1000 1001b (1189h), the remainder 1001 0001 1000 1000b (9188h) with inversed bit position, can be read from the CRCD register.

When going on to (4) above, 23h (00100011b) written in the CRCIN register is reversed and becomes 11000100b. Add 1100 0100 0000 0000 0000 0000b, as 11000100b plus 16 digits, to 1001 0001 1000 1000 0000 0000b, as 1001 0001 1000 1000b plus 8 digits as a remainder of (3) left in the CRCD register to perform the modulo-2 division. 0000 1010 0100 0001b (0A41h), the remainder with reversed bit position, can be read from the CRCD register.

Figure 20.3 CRC Operation

## 21. Programmable I/O Ports

88 programmable input / output ports (I/O ports) are available. The direction registers determine individual port status, input or output. The pull-up control registers determine whether the ports, divided into groups of four ports, are pulled up or not. P8\_5 is an input port and no pull-up is allowed. Port P8\_5 shares the pin with  $\overline{\text{NMI}}$ , so that the  $\overline{\text{NMI}}$  input level can be read from the P8\_5 bit in the P8 register.

Figures 21.1 to 21.5 show the I/O ports. Figure 21.6 shows the I/O Pins.

Each pin functions as an I/O port, a peripheral function input / output, or a bus control pin.

To set peripheral functions, refer to the description for individual functions. If any pin is used as a peripheral function input or D/A converter output pin, set the direction bit of the corresponding pin to 0 (input mode). Any pin used as an output pin for peripheral functions other than the D/A converter is directed for output no matter how the corresponding direction bit is set.

To use as bus control pins, refer to **8.2 “Bus Control”**.

P0 to P5 are capable of VCC2-level input / output; P6 to P10 are capable of VCC1-level input / output.

### 21.1 Port Pi Direction Register (PDi Register, i = 0 to 10)

Figure 21.7 shows the Pi Direction Registers.

This register selects whether the I/O port is to be used for input or output. Each bit in the PDi register corresponds to one port.

During memory extension or microprocessor mode, the PDi registers for the pins functioning as bus control pins (A0 to A19, D0 to D15, CS0 to CS3,  $\overline{\text{RD}}$ , WRL / WR, WRH / BHE, ALE, RDY, HOLD, HLDA, and BCLK) cannot be modified.

### 21.2 Port Pi Register (Pi Register, i = 0 to 10)

Figure 21.8 shows the Pi Registers.

Data input / output to and from external devices are accomplished by reading and writing to the Pi register.

Each bit of 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. Each bit in the Pi register correspond to each port.

During memory extension or microprocessor mode, the Pi registers for the pins functioning as bus control pins (A0 to A19, D0 to D15, CS0 to CS3,  $\overline{\text{RD}}$ , WRL / WR, WRH / BHE, ALE, RDY, HOLD, HLDA, and BCLK) cannot be modified.

### 21.3 Pull-up Control Register 0 to Pull-up Control Register 2 (Registers PUR0 to PUR2)

Figures 21.9 and 21.10 show the Registers PUR0 to PUR2.

Bits in registers PUR0 to PUR2 can be used to select whether or not to pull the corresponding port high in 4 pin 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.

However, the pull-up control register has no effect on P0 to P3, P4\_0 to P4\_3, and P5 during memory extension or microprocessor mode. Although the register contents can be modified, no pull-up resistors are connected.

### 21.4 Port Control Register (PCR Register)

Figure 21.11 shows the PCR Register.

When the P1 register is read after setting the PCR0 bit in the PCR register to 1, the corresponding port latch can be read no matter how the PD1 register is set.

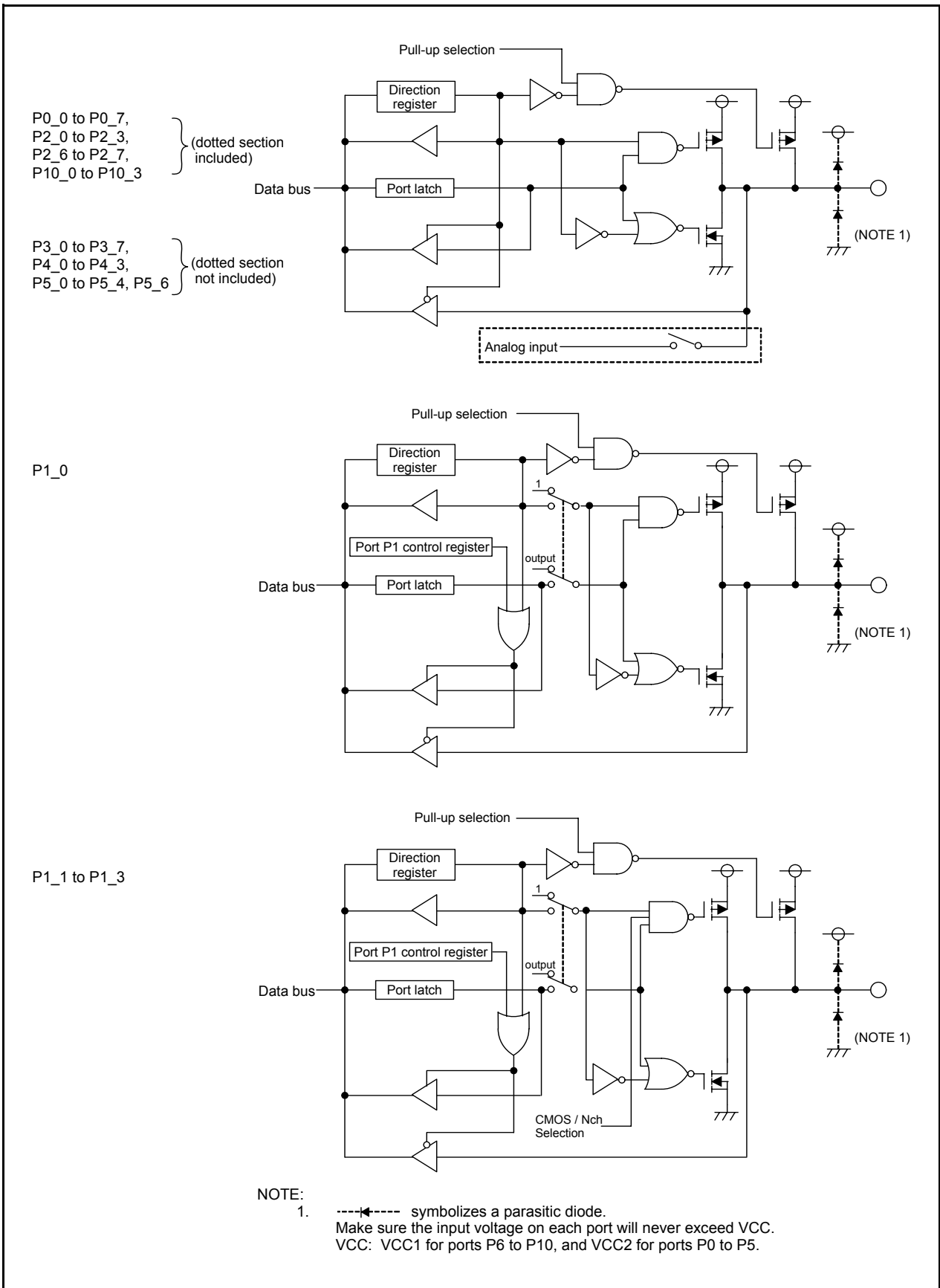


Figure 21.1 I/O Ports (1)



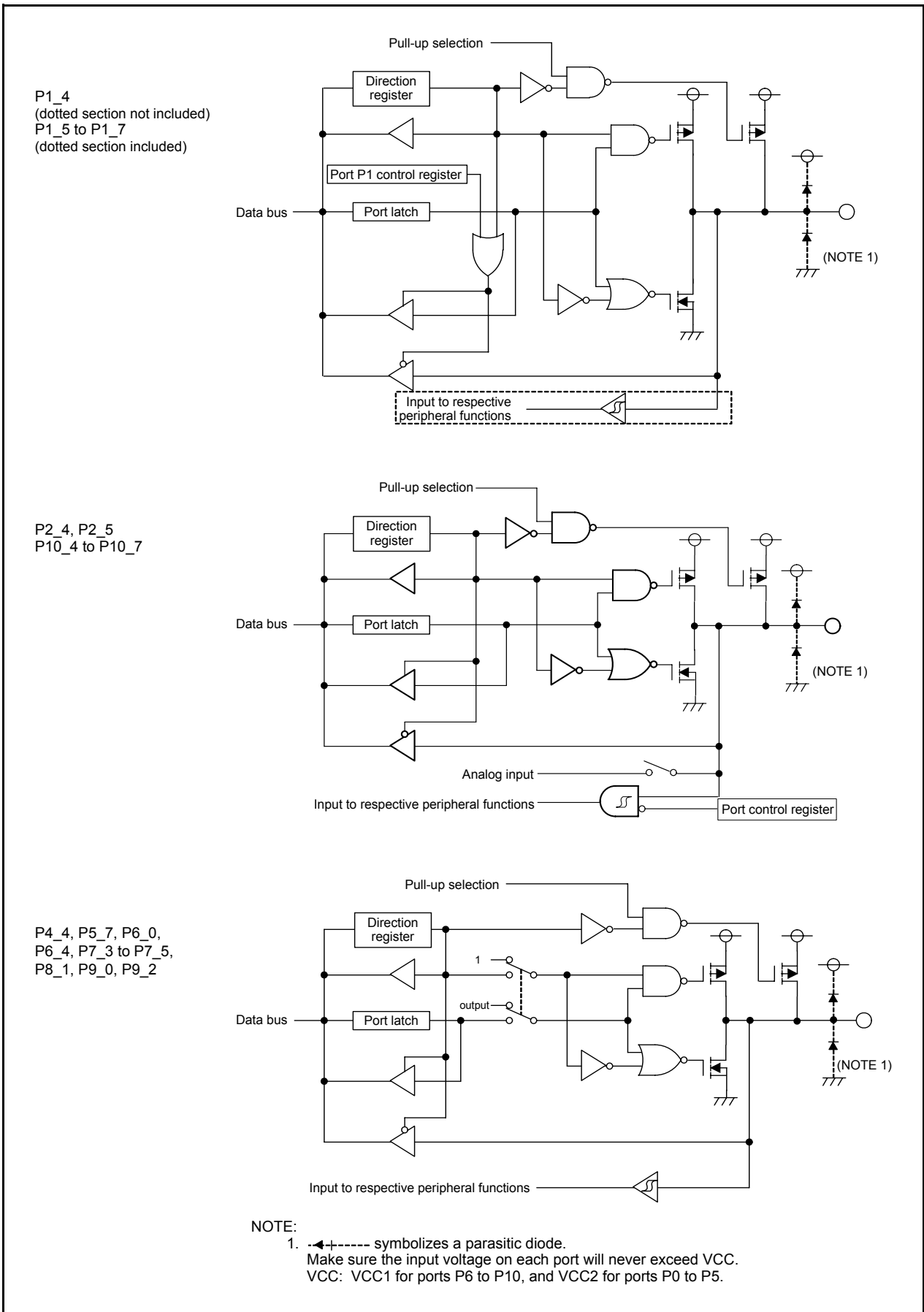


Figure 21.2 I/O Ports (2)

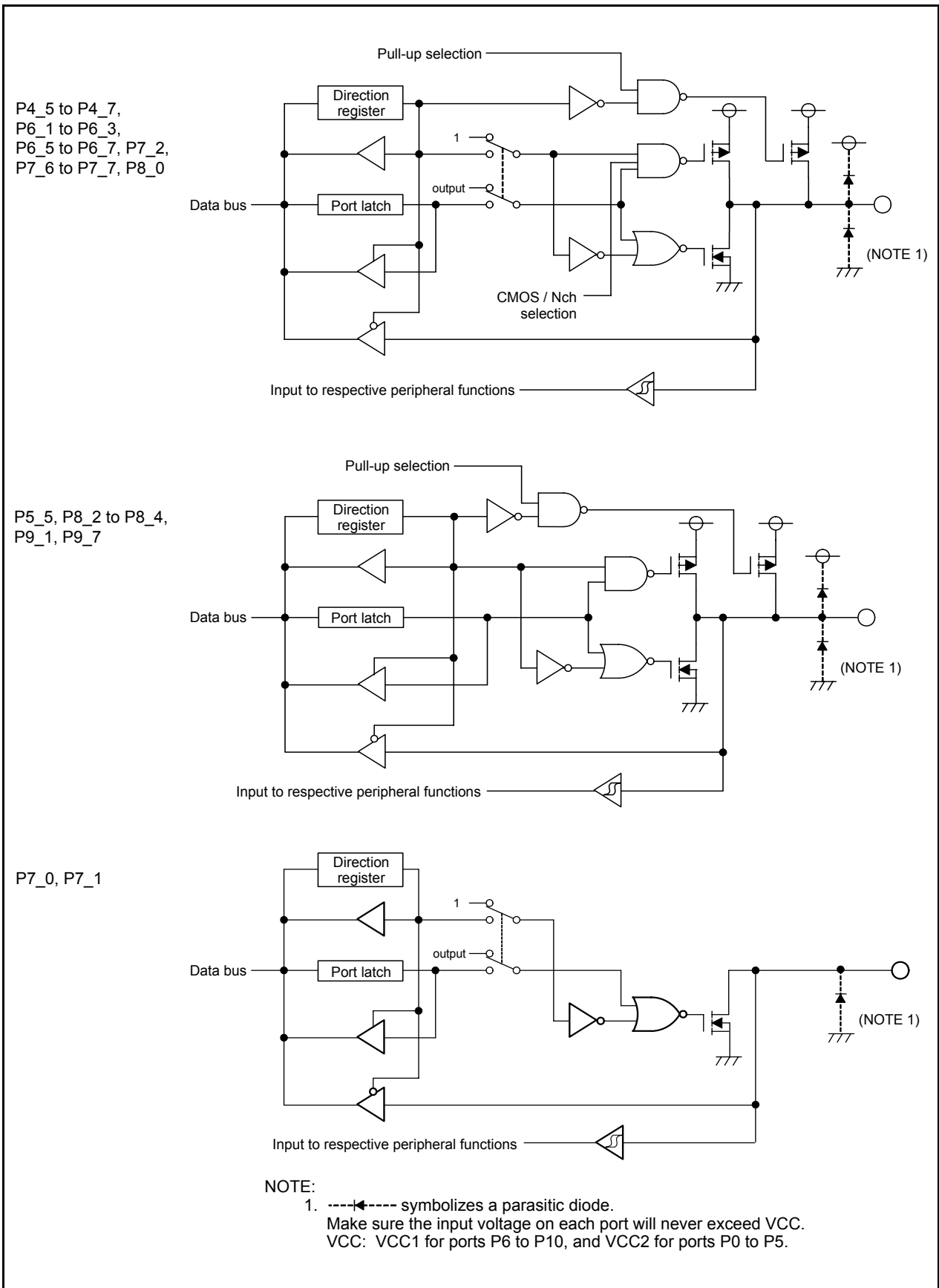


Figure 21.3 I/O Ports (3)

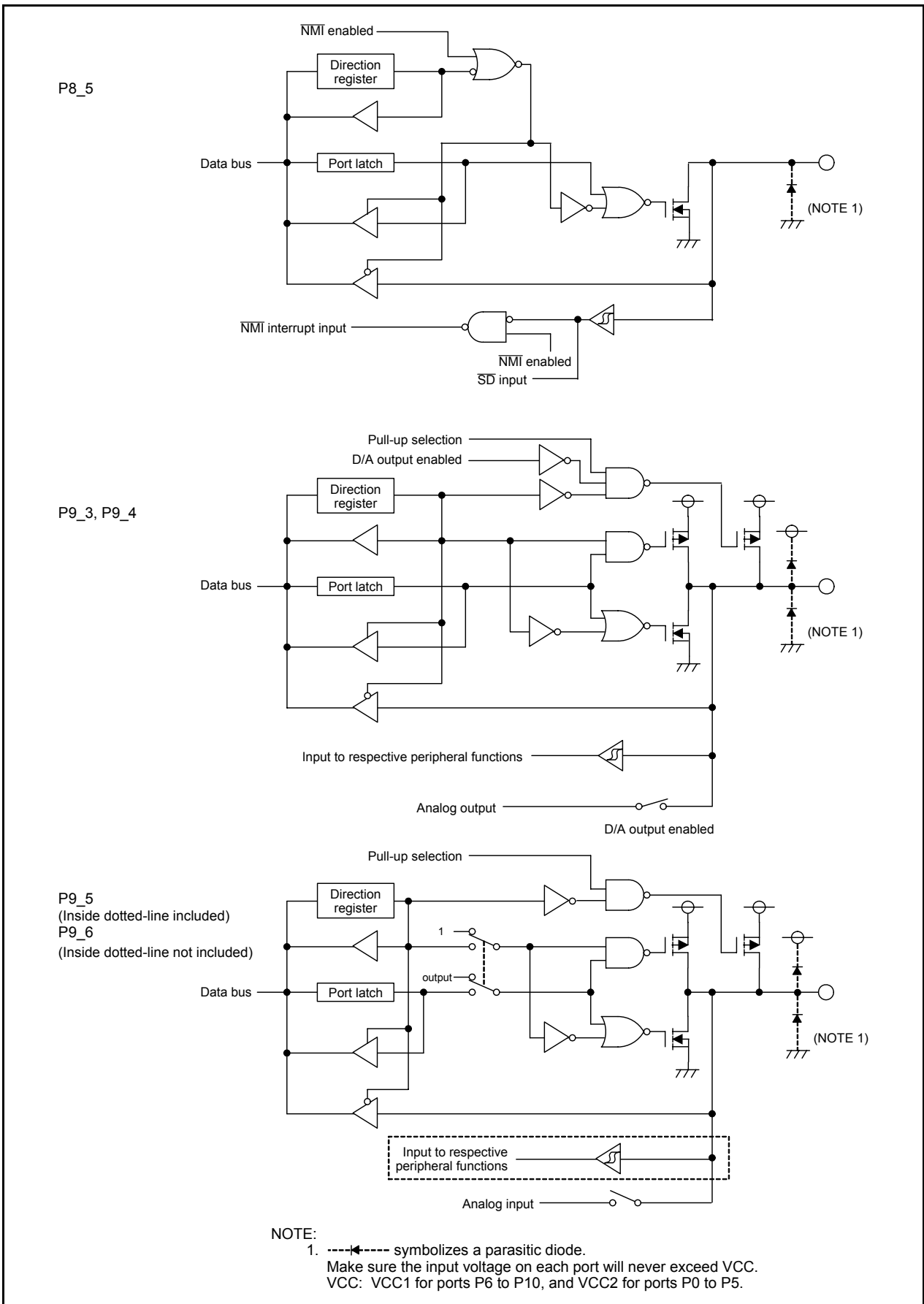


Figure 21.4 I/O Ports (4)

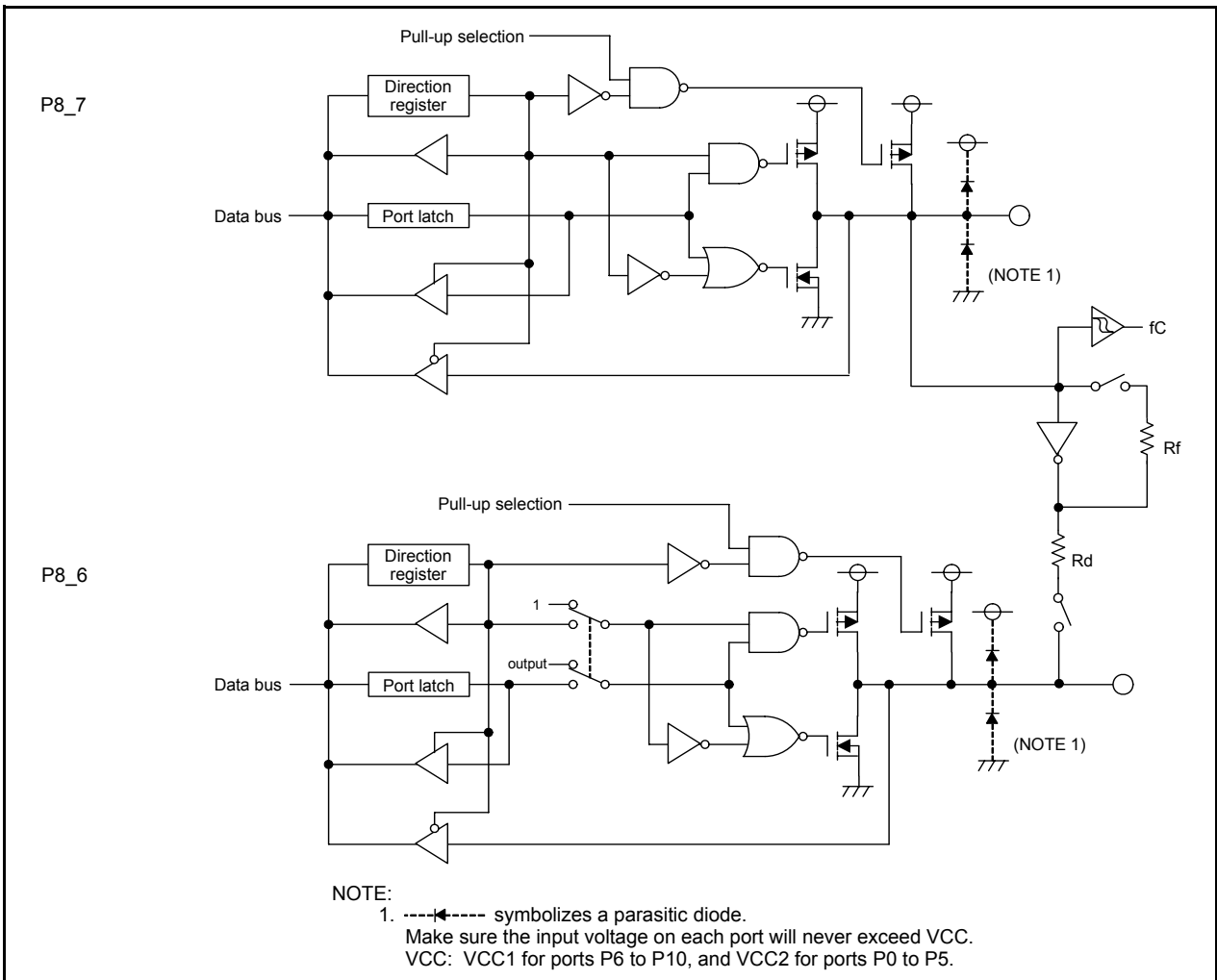


Figure 21.5 I/O Ports (5)

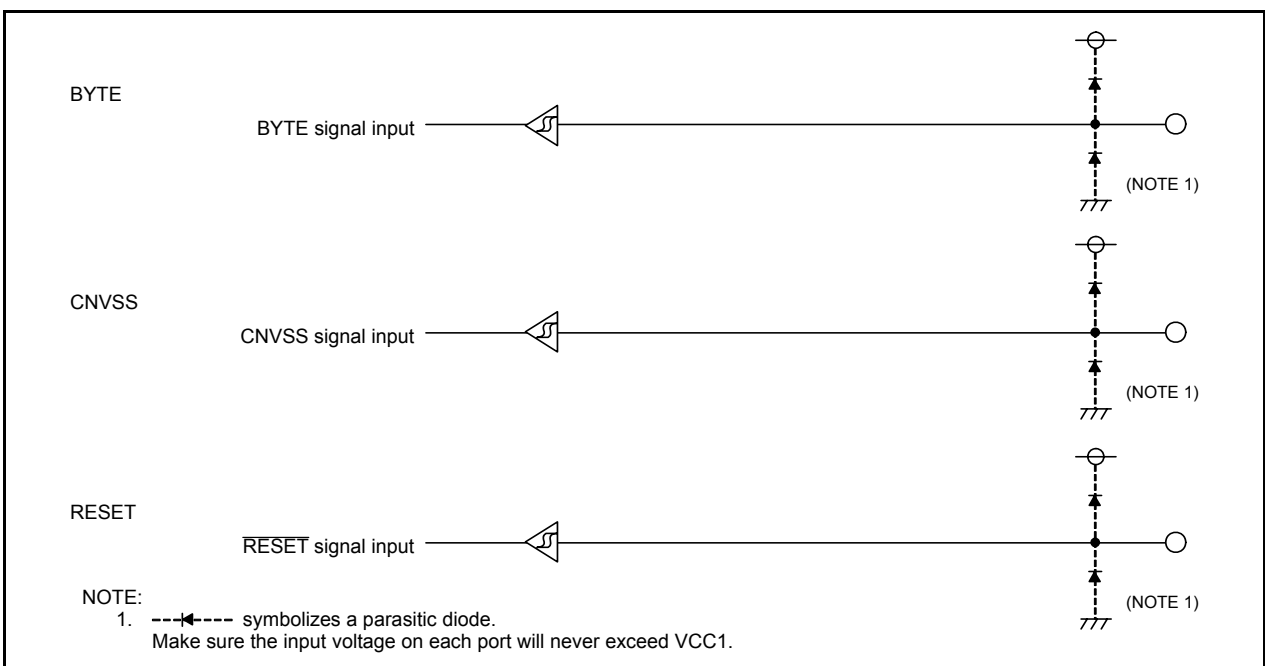


Figure 21.6 I/O Pins

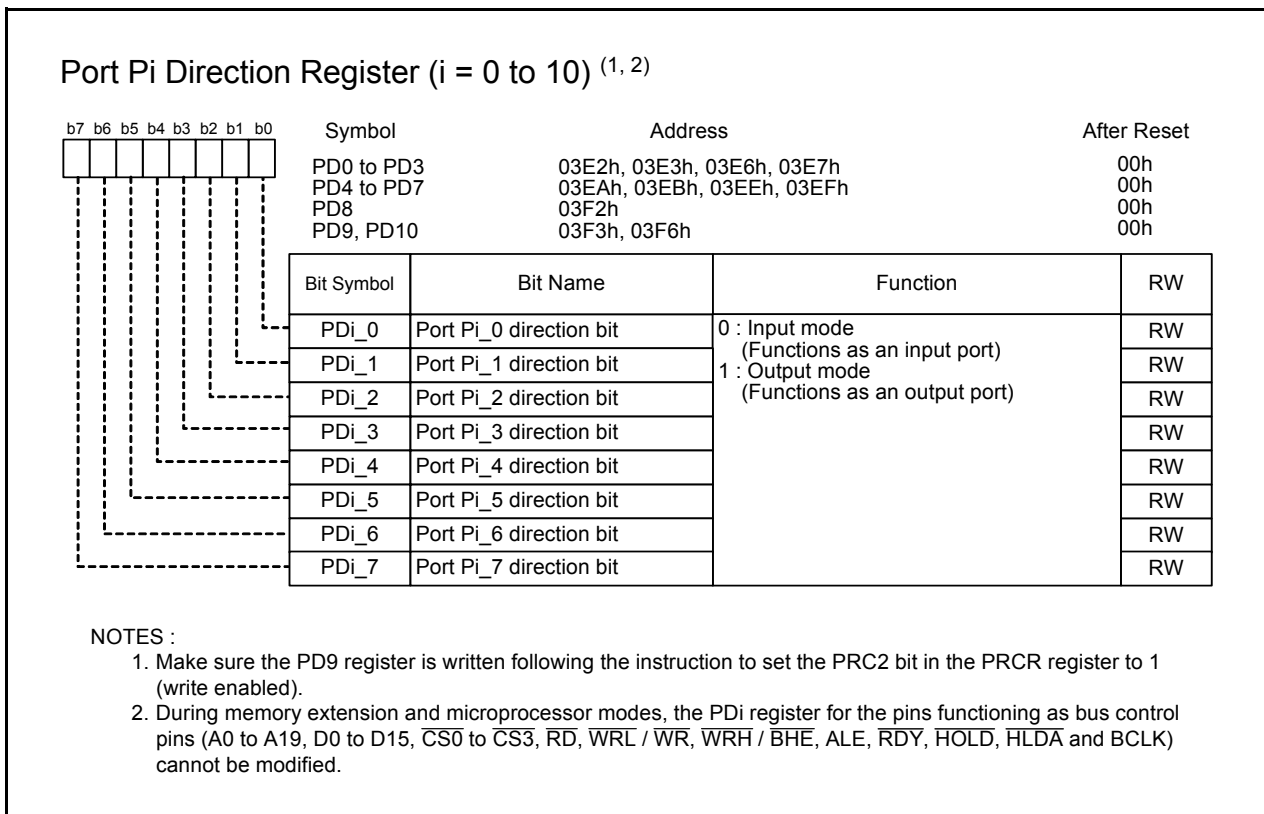
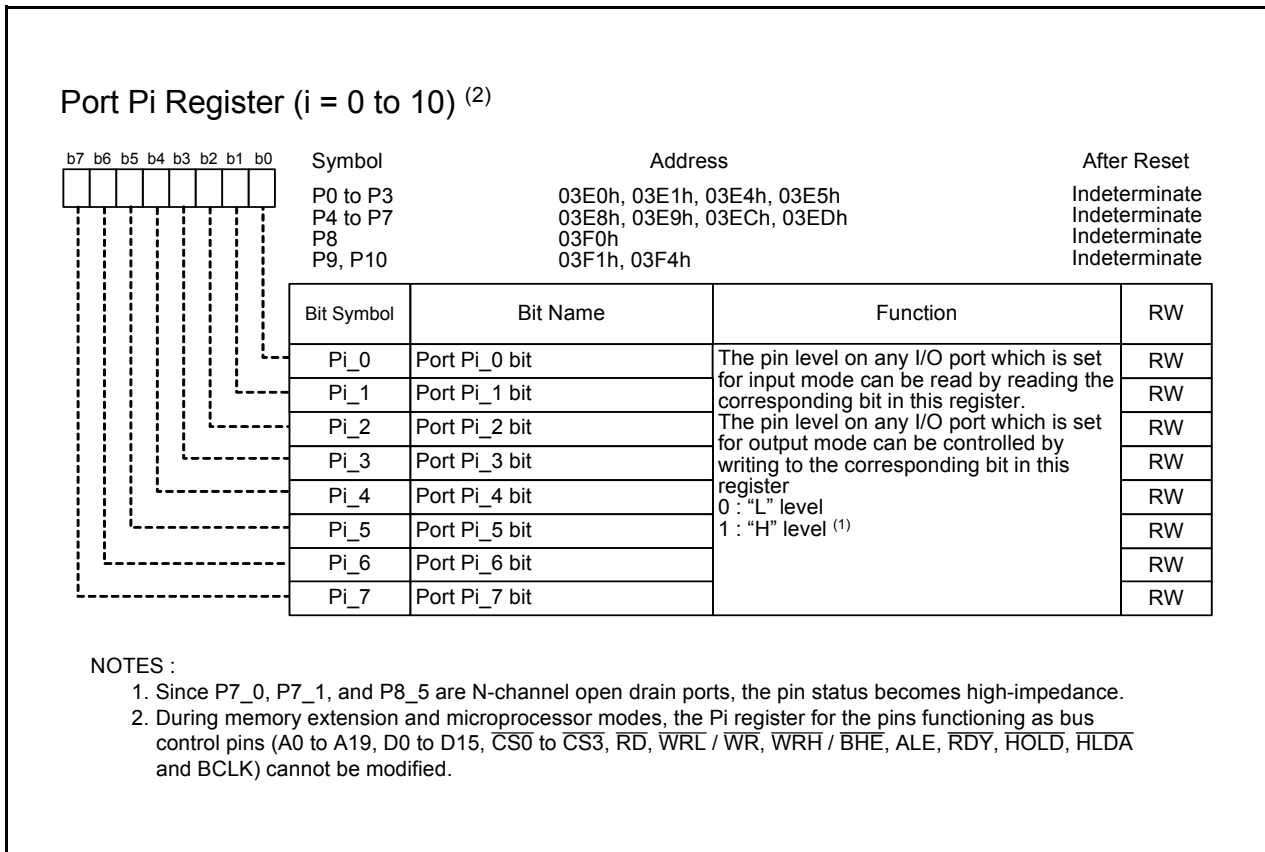
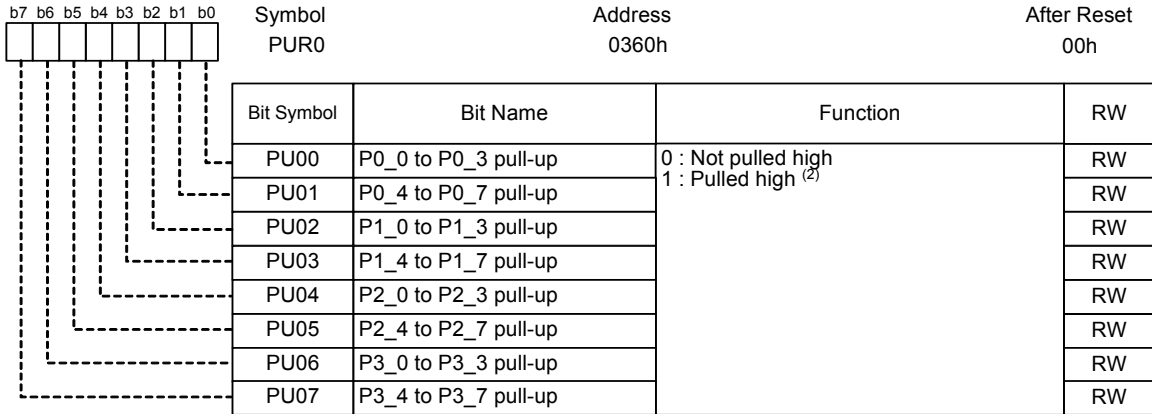


Figure 21.7 Registers PD0 to PD10



**Figure 21.8 Registers P0 to P10**

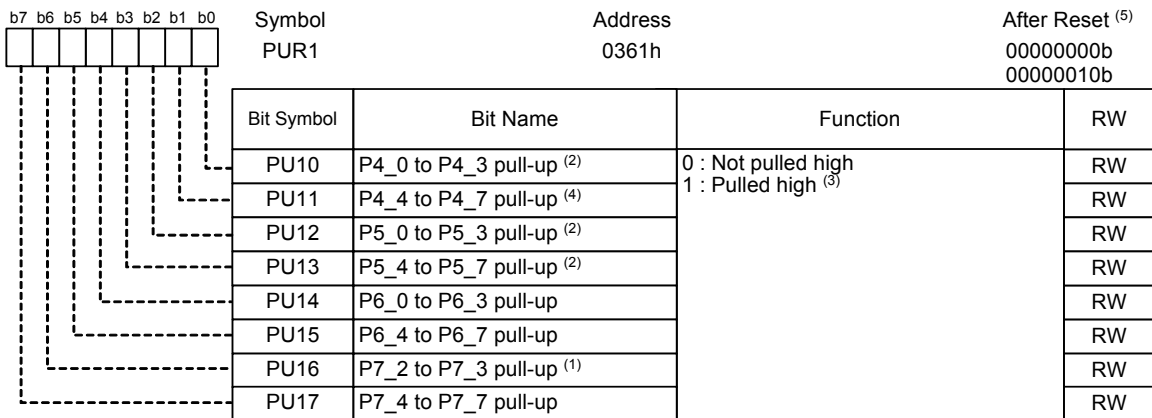
**Pull-up Control Register 0 <sup>(1)</sup>**



**NOTES :**

1. During memory extension or microprocessor mode, the corresponding register contents can be modified, but the pins are not pulled high.
2. The pin for which this bit is 1 (pulled high) and the direction bit is 0 (input mode) is pulled high.

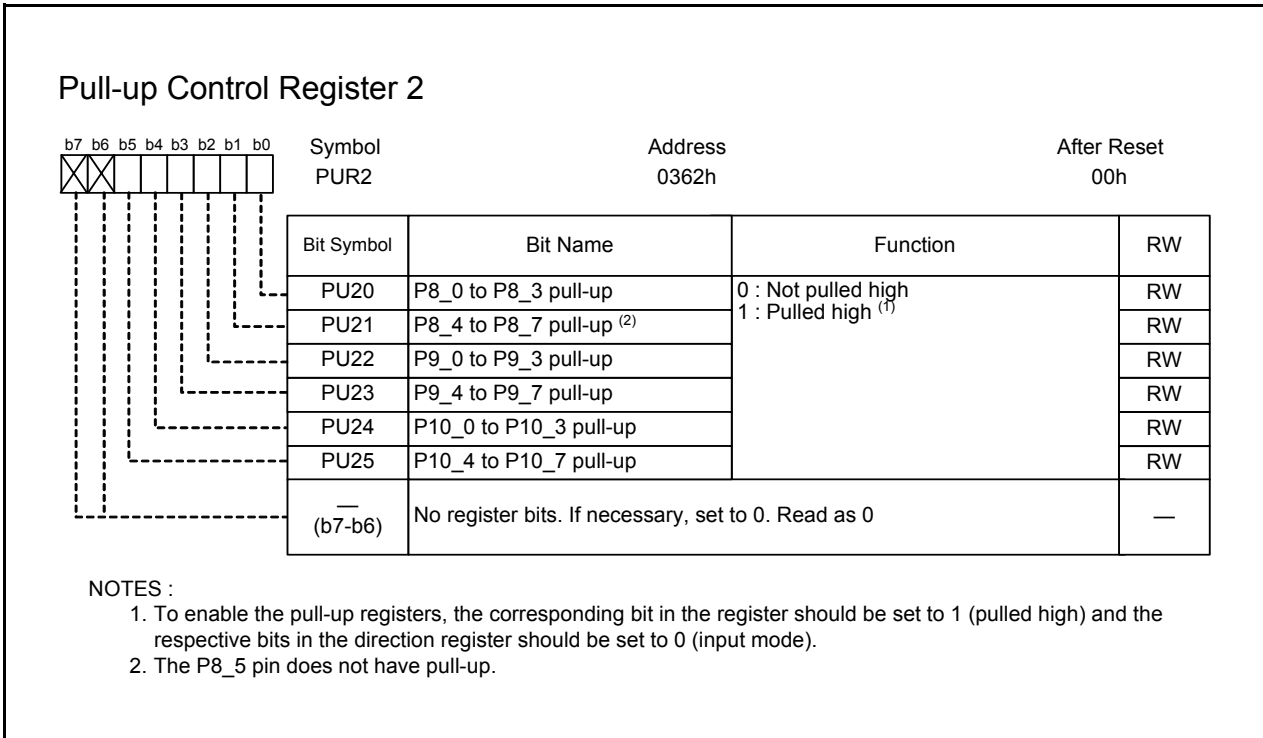
**Pull-up Control Register 1**



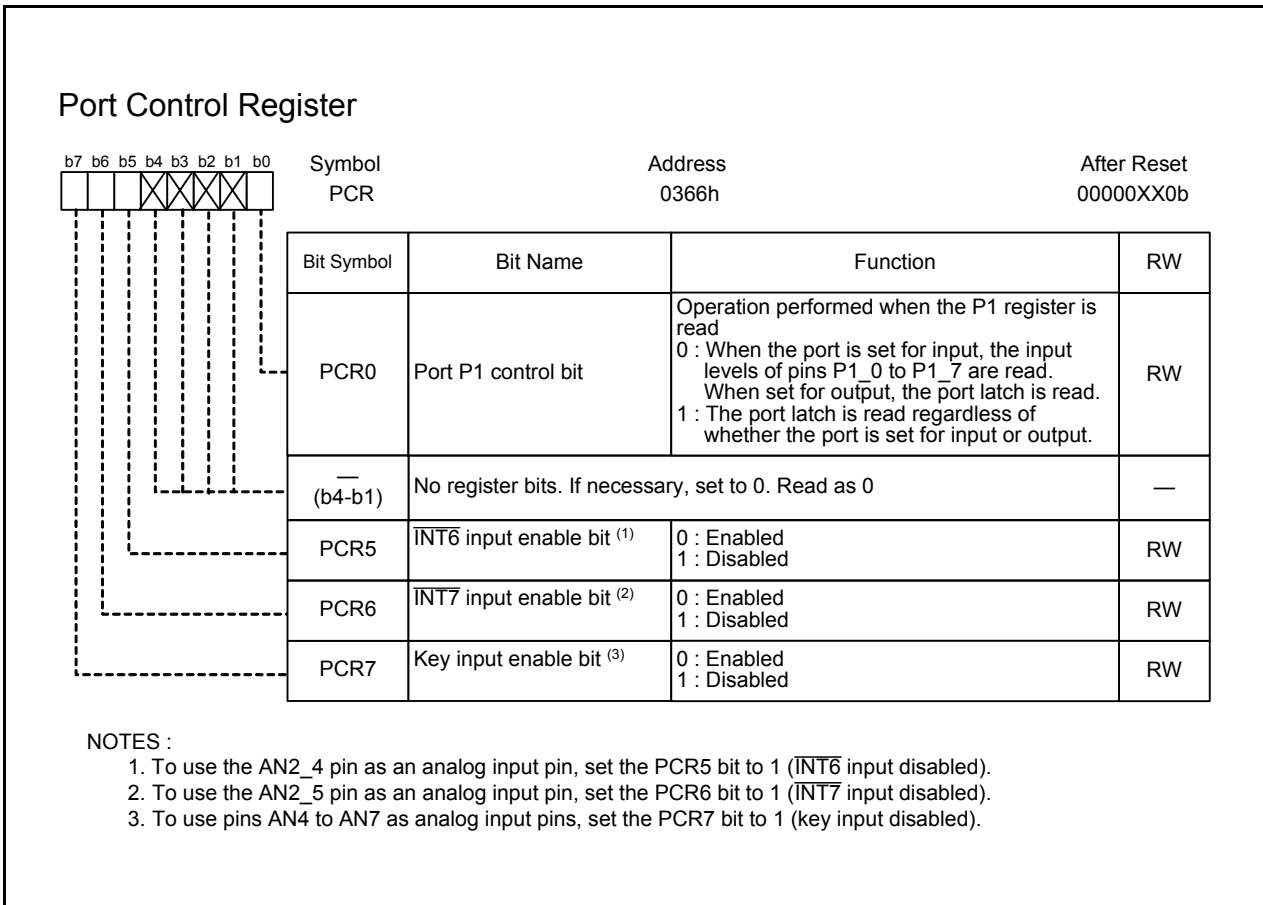
**NOTES :**

1. Pins P7\_0 and P7\_1 do not have pull-ups.
2. During memory extension and microprocessor modes, the pins are not pulled high although the contents of these bits can be modified.
3. To enable the pull-up registers, the corresponding bit in the register should be set to 1 (pulled high) and the respective bits in the direction register should be set to 0 (input mode).
4. If bits PM01 to PM00 in the PM0 register are set to 01b (memory expansion mode) or 11b (microprocessor mode) in a program during single-chip mode, the PU11 bit becomes 1.
5. The values after hardware reset 1 or brown-out reset is as follows:
  - 00000000b when input on CNVSS pin is "L"
  - 00000010b when input on CNVSS pin is "H"
 The values after software reset, watchdog timer reset, and oscillation stop detection reset are as follows:
  - 00000000b when bits PM01 to PM00 are 00b (single-chip mode)
  - 00000010b when bits PM01 to PM00 are 01b (memory expansion mode) or 11b (microprocessor mode)

**Figure 21.9 Registers PUR0 and PUR1**



**Figure 21.10 PUR2 Register**



**Figure 21.11 PCR Register**



**Table 21.1 Unassigned Pin Handling in Single-chip Mode**

Pin Name	Connection (2)
Ports P0 to P5	One of the followings: Set for input mode and connect a pin to VSS via resistor (pull-down) Set for input mode and connect a pin to VCC2 via resistor (pull-up) Set for output mode and leave the pins open (1)
Ports P6 to P10	One of the followings: Set for input mode and connect a pin to VSS via resistor (pull-down) Set for input mode and connect a pin to VCC1 via resistor (pull-up) Set for output mode and leave the pins open (1, 3)
XOUT (4)	Open
XIN	Connect to VCC1 (pull-up) via resistor
AVCC, VREF	Connect to VCC1
AVSS, BYTE	Connect to VSS

**NOTES:**

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.
2. Furthermore, by considering a possibility that the contents of the direction registers could be changed by noise or noise-induced loss of control, it is recommended that the contents of the direction registers be regularly reset in software to improve reliability of the program.
3. Make sure the unused pins are processed with the shortest possible wiring from the microcomputer pins (within 2 cm).
4. When the ports P7\_0, P7\_1, and P8\_5 are set for output mode, make sure a low-level signal is output from the pins.
5. The ports P7\_0, P7\_1, and P8\_5 are N-channel open-drain outputs.
6. This applies when external clock is input to the XIN pin or when VCC1 is connected to via a resistor.

**Table 21.2 Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode**

Pin Name	Connection <sup>(2)</sup>
Ports P0 to P5	One of the followings: Set for input mode and connect a pin to VSS via resistor (pull-down) Set for input mode and connect a pin to VCC2 via resistor (pull-up) Set for output mode and leave the pins open <sup>(1, 3)</sup>
Ports P6 to P10	One of the followings: Set for input mode and connect a pin to VSS via resistor (pull-down) Set for input mode and connect a pin to VCC1 via resistor (pull-up) Set for output mode and leave the pins open <sup>(1, 4)</sup>
BHE, ALE, HLDA, XOUT <sup>(5)</sup> , BCLK <sup>(6)</sup>	Open
HOLD, RDY	Connect to VCC2 (pull-up) via resistor
XIN	Connect to VCC1 (pull-up) via resistor
AVCC, VREF	Connect to VCC1
AVSS	Connect to VSS

## NOTES:

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.
2. Furthermore, by considering a possibility that the contents of the direction registers could be changed by noise or noise-induced loss of control, it is recommended that the contents of the direction registers be regularly reset in software to improve reliability of the program.
3. Make sure the unused pins are processed with the shortest possible wiring from the microcomputer pins (within 2 cm).
4. If the CNVSS pin has the VSS level applied to it, these pins are set for input ports until the processor mode is switched over in a program after reset. For this reason, the voltage levels on these pins become indeterminate, causing the power supply current to increase while they remain set for input ports.
5. When the ports P7\_0, P7\_1, and P8\_5 are set for output mode, make sure a low-level signal is output from the pins.
6. The ports P7\_0, P7\_1, and P8\_5 are N-channel open-drain outputs.
7. This applies when external clock is input to the XIN pin or when VCC1 is connected to via a resistor.
8. If the PM07 bit in the PM0 register is set to 1 (BCLK not output), connect this pin to VCC2 via a resistor (pulled high).

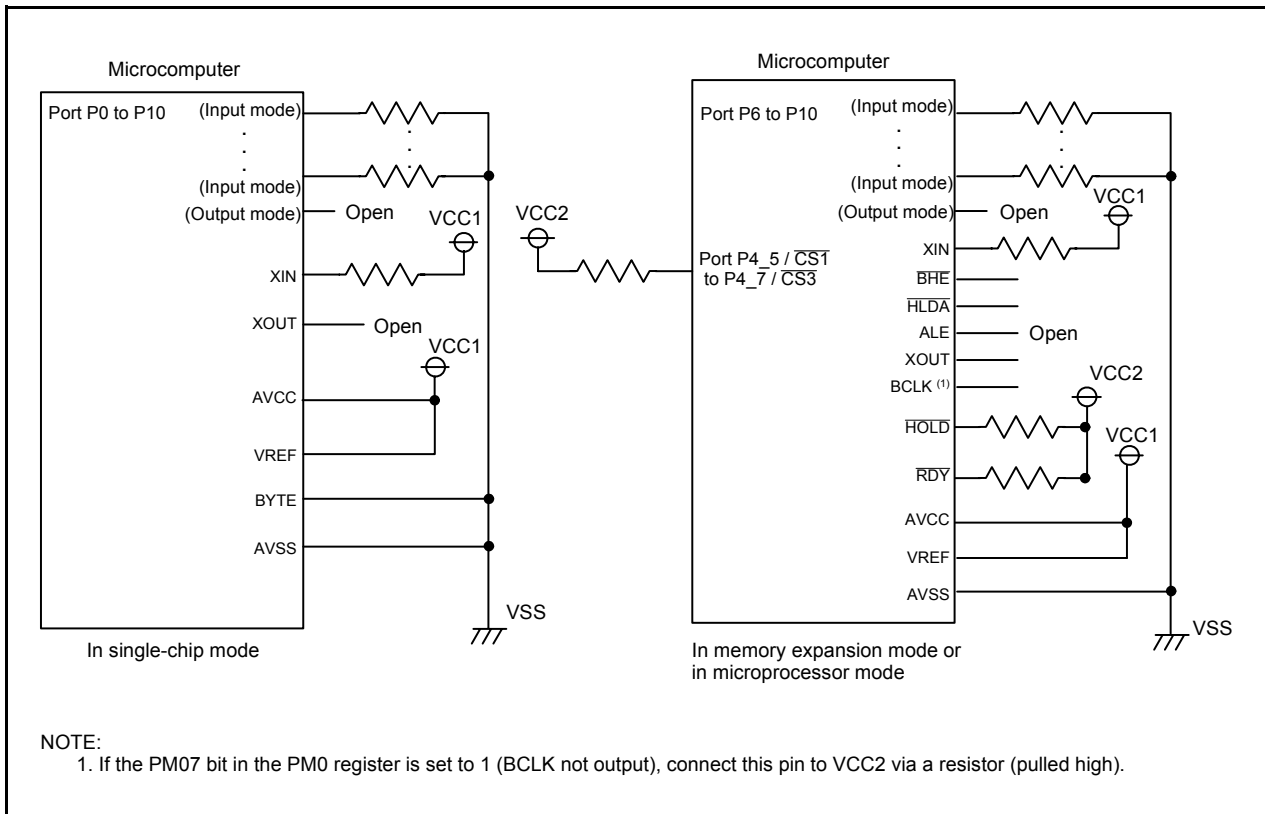


Figure 21.12 Unassigned Pin Handling

## 22. Flash Memory Version

The flash memory can perform in three rewrite modes: CPU rewrite mode, standard serial I/O mode, and parallel I/O mode. Table 22.1 lists specifications of the flash memory version. See Tables 1.1 and 1.2 Specifications Overview for the items not listed in Table 22.1.

**Table 22.1 Flash Memory Version Specifications**

Item	Specification
Flash Memory Rewrite Mode	3 modes (CPU rewrite, standard serial I/O, parallel I/O)
Erase Block	Program ROM 1
	Program ROM 2
	Data Flash
Program Method	In units of 2 words
Erase Method	Block erase
Program and Erase Control Method	Program and erase controlled by software command
Protect Method	The lock bit protects each block
Number of Commands	8 commands
Program and Erase Endurance	100 times (2, 3)
Data Retention	10 years
ROM Code Protection	Parallel I/O and standard serial I/O modes are supported

NOTE:

1. Definition of program and erase endurance

The program and erase endurance refers to the number of per-block erasures.

For example, assume a case where a 4 Kbyte block is programmed in 1,024 operations, writing two words at a time, and erased thereafter. In this case, the block is reckoned as having been programmed and erased once.

If the program and erase endurance is 100 times, each block can be erased up to 100 times.

**Table 22.2 Flash Memory Rewrite Modes Overview**

Flash Memory Rewrite Mode	CPU rewrite Mode (1)	Standard Serial I/O Mode	Parallel I/O Mode
Function	Program ROM 1, program ROM 2, and data flash are rewritten when the CPU executes software commands. EW0 mode: Rewritable in areas other than flash memory (2) EW1 mode: Rewritable in the flash memory	Program ROM 1, program ROM 2, and data flash are rewritten using a dedicated serial programmer. Standard serial I/O mode 1: clock synchronous serial I/O Standard serial I/O mode 2: clock asynchronous serial I/O	Program ROM 1, program ROM 2 and data flash are rewritten using a dedicated parallel programmer.
Areas Which Can Be Rewritten	Program ROM 1, program ROM 2, and data flash	Program ROM 1, program ROM 2, and data flash	Program ROM 1 and program ROM 2
Operating Mode	Single-chip mode Memory expansion mode (EW0 mode)	Boot mode	Parallel I/O mode
ROM Programmer	None	Serial programmer	Parallel programmer

NOTES:

1. The PM13 bit remains set to 1 while the FMR01 bit in the FMR0 register = 1 (CPU rewrite mode enabled). The PM13 bit is reverted to its original value by clearing the FMR01 bit to 0 (CPU rewrite mode disabled). However, if the PM13 bit is changed during CPU rewrite mode, its changed value is not reflected until after the FMR01 bit is cleared to 0.
2. In CPU rewrite mode, bits PM10 and PM13 in the PM1 register are set to 1. The rewrite control program can only be executed in the internal RAM or in an external area that is enabled for use when the PM13 bit = 1. When the PM13 bit = 0 and the flash memory is used in 4-Mbyte mode, the extended accessible area (40000h to BFFFFh) cannot be used.

## 22.1 Memory Map

The flash memory contains program ROM 1, program ROM 2, and data flash. Figure 22.1 shows a Flash Memory Block Diagram.

Program ROM 1 is divided into several blocks, each of which can be protected (locked) from program or erase. Program ROM 1 and program ROM 2 can be rewritten in CPU rewrite, standard serial I/O, and parallel I/O modes.

Program ROM 2 can be used when the PRG2C0 bit in the PRG2C register is set to 0 (program ROM 2 enabled). The user boot code area is in program ROM 2. Data flash can be used when the PM10 bit in the PM1 register is set to 1 (0E000h to 0FFFFh: data flash). Data flash is divided into block A and block B.

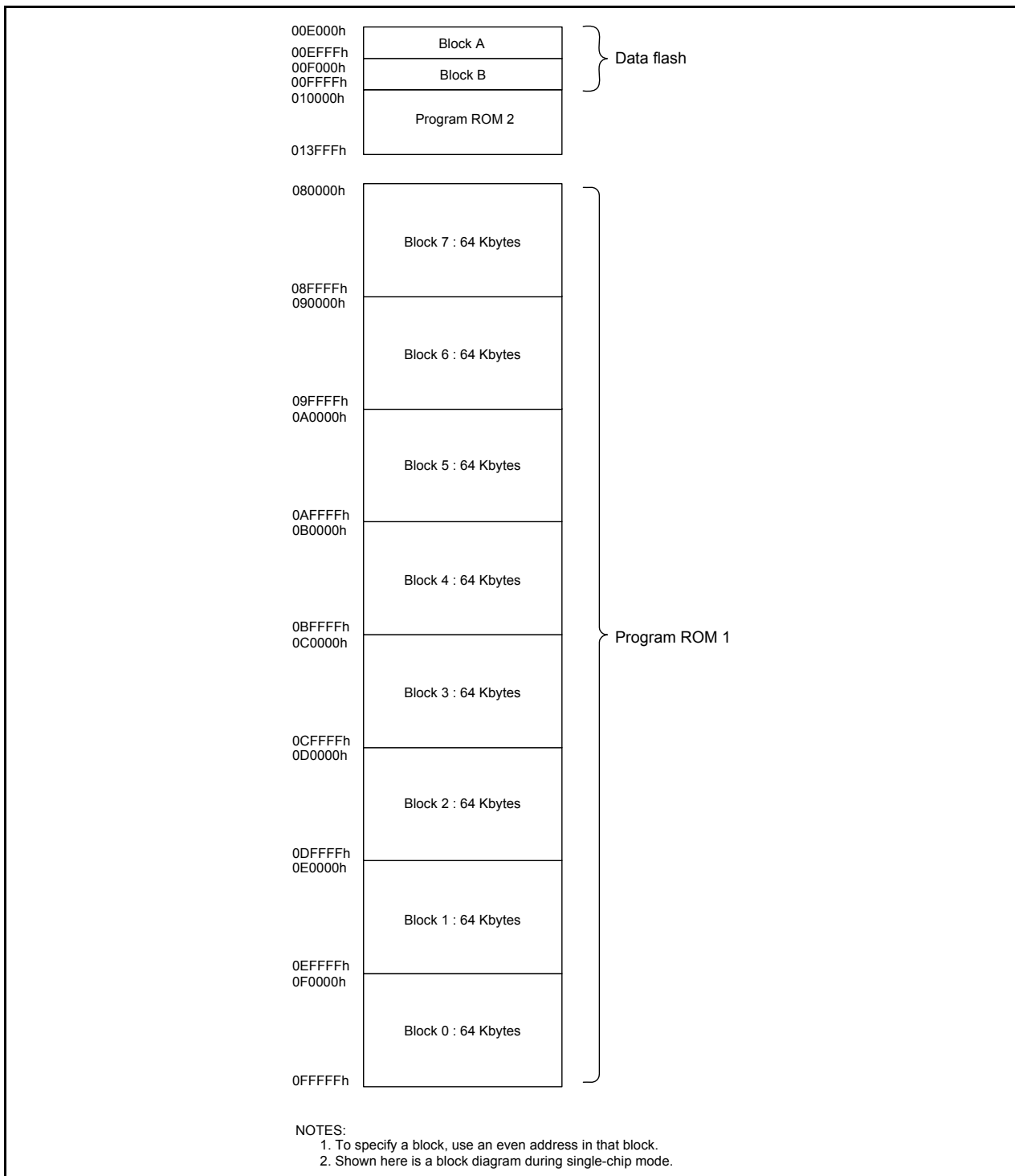


Figure 22.1 Flash Memory Block Diagram

### 22.1.1 Boot Mode

The microcomputer enters boot mode when a hardware reset occurs while an “L” signal is applied to the P5\_5 pin and an “H” signal is applied to pins CNVSS and P5\_0. In boot mode, user boot mode or standard serial I/O mode is selected in accordance with the data in the user boot code area. Refer to **22.4 “Standard Serial I/O Mode”** for details.

### 22.1.2 User Boot Function

User boot mode can be selected by the status of a port when the MCU starts in boot mode. Table 22.3 shows the user boot function specifications.

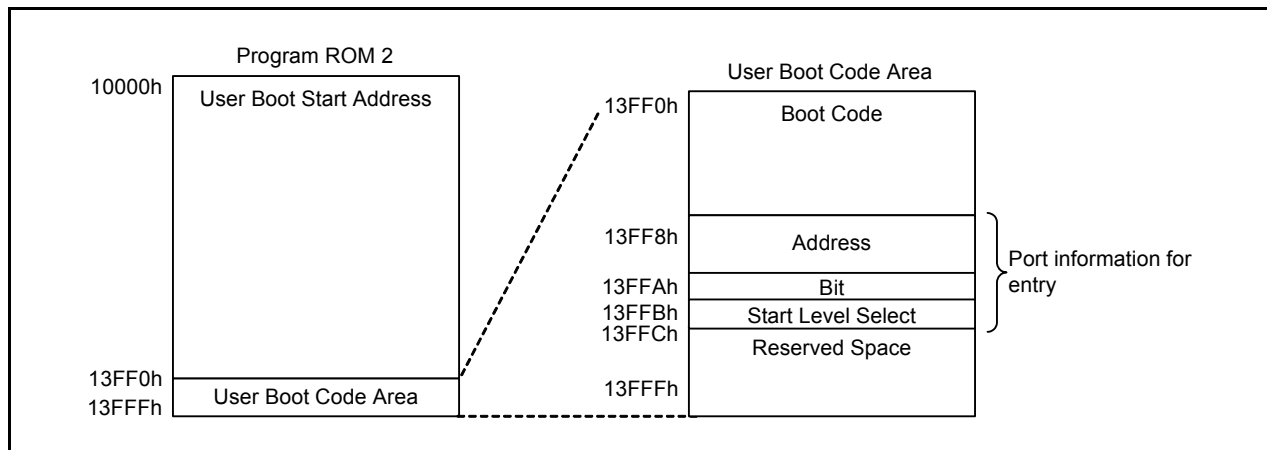
**Table 22.3 User Boot Function Specifications**

Item	Specification
Entry Pin	None or select a port from P0_0 to P10_7
User Boot Start Level	Select “H” or “L”
User Boot Start Address	Address 10000h (the start address of program ROM 2)

Set “UserBoot” in ASCII code to the addresses 13FF0h to 13FF7h in the user boot code area and select a port for entry from addresses 13FF8h to 13FF9h and the start level with the address 13FFBh. After starting boot mode, user boot mode or standard serial I/O mode is selected in accordance with the level of the selected port.

In addition, if addresses 13FF0h to 13FF7h are set to “UserBoot” in ASCII code and address 13FF8h to 13FFBh are set to “00h”, user boot mode is selected.

In user boot mode, the program of address 10000h (the start address of program ROM2) is executed. Figure 22.2 shows user boot code area, Table 22.4 shows the start mode, Tables 22.5 and 22.6 the values to be set to the user boot code area.



**Figure 22.2 User Boot Code Area**

**Table 22.4 Start Mode (When the Port P<sub>j</sub> is Selected for Entry)**

Boot Code (13FF0h to 13FF7h)	Port information for entry			Port P <sub>i</sub> input level	Start Mode
	Address (13FF8h to 13FF9h)	Bit (13FFAh)	Start level Select (13FFBh)		
"UserBoot" in ASCII code	0000h	00h	00h	–	User boot mode
	Pi register address	00h to 07h (value of j)	00h	H	Standard serial I/O mode
				L	User boot mode
	Pi register address	00h to 07h (value of j)	01h	H	User boot mode
L				Standard serial I/O mode	
Other Than "UserBoot" in ASCII code	–	–	–	–	Standard serial I/O mode

i=0 to 10, j=0 to 7

## NOTES:

1. Do not use another combination of values apart from Table 22.4.
2. Refer to **Table 22.5 "UserBoot" in ASCII code**
3. Refer to **Table 22.6 "Addresses of Selectable Ports for Entry"**

**Table 22.5 "UserBoot" in ASCII code**

Address	13FF0h	13FF1h	13FF2h	13FF3h	13FF4h	13FF5h	13FF6h	13FF7h
ASCII code	55h (U)	73h (s)	65h (e)	72h (r)	42h (B)	6Fh (o)	6Fh (o)	74h (t)
	Upper- case	Lower-case			Upper- case	Lower-case		

**Table 22.6 Addresses of Selectable Ports for Entry**

Port	Address	Port	Address
P0	03E0h	P6	03ECh
P1	03E1h	P7	03EDh
P2	03E4h	P8	03F0h
P3	03E5h	P9	03F1h
P4	03E8h	P10	03F4h
P5	03E9h		

## 22.2 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 I/O mode to prevent the flash memory from reading or rewriting.

### 22.2.1 ROM Code Protect Function

The ROM code protect function inhibits the flash memory from being read or rewritten during parallel input/output mode. Figure 22.3 shows the OFS1 Address. The OFS1 address is located in block 0 in program ROM 1.

The ROM code protect function is enabled when the ROMCP1 bit is set to 0.

When exiting ROM code protect, erase block 0 including the OFS1 address by the CPU rewrite mode or the standard serial I/O mode.

### 22.2.2 ID Code Check Function

Use the ID code check function in standard serial I/O mode. The ID code sent from the serial programmer is compared with the ID code written in the flash memory for a match. If the ID codes do not match, commands sent from the serial programmer are not accepted. However, if the four bytes of the reset vector are "FFFFFFFFh", ID codes are not compared, allowing all commands to be accepted.

The ID codes are 7-byte data stored consecutively, starting with the first byte, into addresses 0FFFDf, 0FFFE3h, 0FFFEb, 0FFFEFh, 0FFFF3h, 0FFFF7h, and 0FFFFBh. The flash memory must have a program with the ID codes set in these addresses.

Table 22.7 shows address for ID code stored.

The reserved character sequence of the ASCII codes "ALeRASE" is used for forced erase function. The reserved character sequence of the ASCII codes "Protect" is used for standard serial I/O mode disabled function. Table 22.7 lists reserved character sequence.

When the ID codes stored in the ID code addresses in the user ROM area are set to the ASCII codes: "ALeRASE" as the combination table listed in Table 22.7, forced erase function becomes active. When the forced erase function or standard serial I/O mode disabled function is not used, use another combination of the ASCII codes.

**Table 22.7 Reserved Character Sequence (Reserved Word)**

ID Code Address		Reserved word combination of ID Code (ASCII)	
		ALeRASE	Protect
FFFDf	ID1	41h (A)	50h (upper-case P)
FFFE3h	ID2	4Ch (L)	72h (lower-case r)
FFFEb	ID3	65h (e)	6Fh (lower-case o)
FFFEFh	ID4	52h (R)	74h (lower-case t)
FFFF3h	ID5	41h (A)	65h (lower-case e)
FFFF7h	ID6	53h (S)	63h (lower-case c)
FFFFBh	ID7	45h (E)	74h (lower-case t)

Reserve word for forced erase function: A set of reserved characters that match all the ID code addresses in sequence as the combination table listed in Table 22.7.



### 22.2.3 Forced Erase Function

This function is available only in standard serial I/O mode.

When the reserved characters, "ALeRASE" in ASCII code, are sent from the serial programmer as ID codes, the content of the user ROM area will be erased at once. However, if the ID codes stored in the ID code addresses in the user ROM area are set to other than a reserved word "ALeRASE" (other than the combination table listed in Table 22.7) when the ROMCP bit in the ROMCP address is set to other than 11b (ROM code protect enabled), forced erase function is ignored and ID code check is executed. Table 22.8 lists conditions and functions for forced erase function.

When both the ID codes sent from the serial programmer and the ID codes stored in the ID code addresses correspond to the reserved word "ALeRASE", the user ROM area will be erased. However, when the serial programmer sends other than "ALeRASE", even if the ID codes stored in the ID code addresses are "ALeRASE", there is no ID match and any command is ignored. The user ROM area remains protected accordingly.

**Table 22.8 Forced Erase Function**

Condition			Function
ID code from serial programmer	Code in ID code stored address	ROMCP1 bit in the OFS1 address	
ALeRASE	ALeRASE	–	User ROM area all erase (forced erase function)
	Other than ALeRASE (1)	1 (ROM code protect disabled)	
		0 (ROM code protect enabled)	ID code check
Other than ALeRASE	ALeRASE	–	ID code check (no ID match)
	Other than ALeRASE (1)	–	ID code check

NOTE:

1. For the combination of the stored addresses is "Protect", refer to **22.2.4 "Standard Serial I/O Mode Disable Function"**.

### 22.2.4 Standard Serial I/O Mode Disable Function

This function is available in standard serial I/O mode. When the ID codes in the ID code stored addresses are set to "Protect" in ASCII code, the MCU does not communicate with a serial programmer. Therefore, the flash memory cannot be read, written or erased by a serial programmer. User boot mode can be selected, when the ID codes set to "Protect".

When the ID codes are set to "Protect" and the ROMCP1 bit in the address OFS1 is set to 0 (ROM code protect enabled), ROM code protection cannot be disabled by a serial programmer. Therefore, the flash memory cannot be read, written or erased by a serial or parallel programmer.

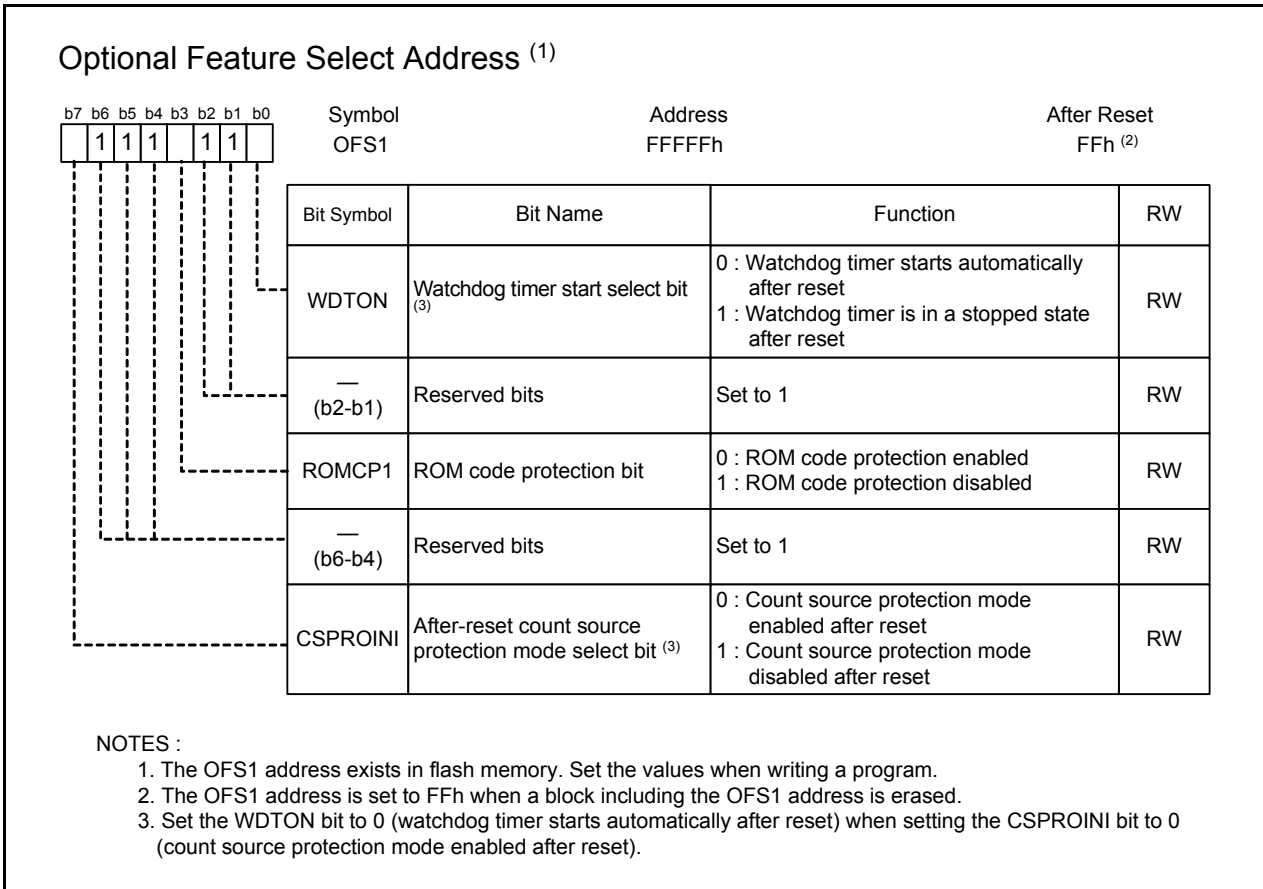


Figure 22.3 OFS1 Address

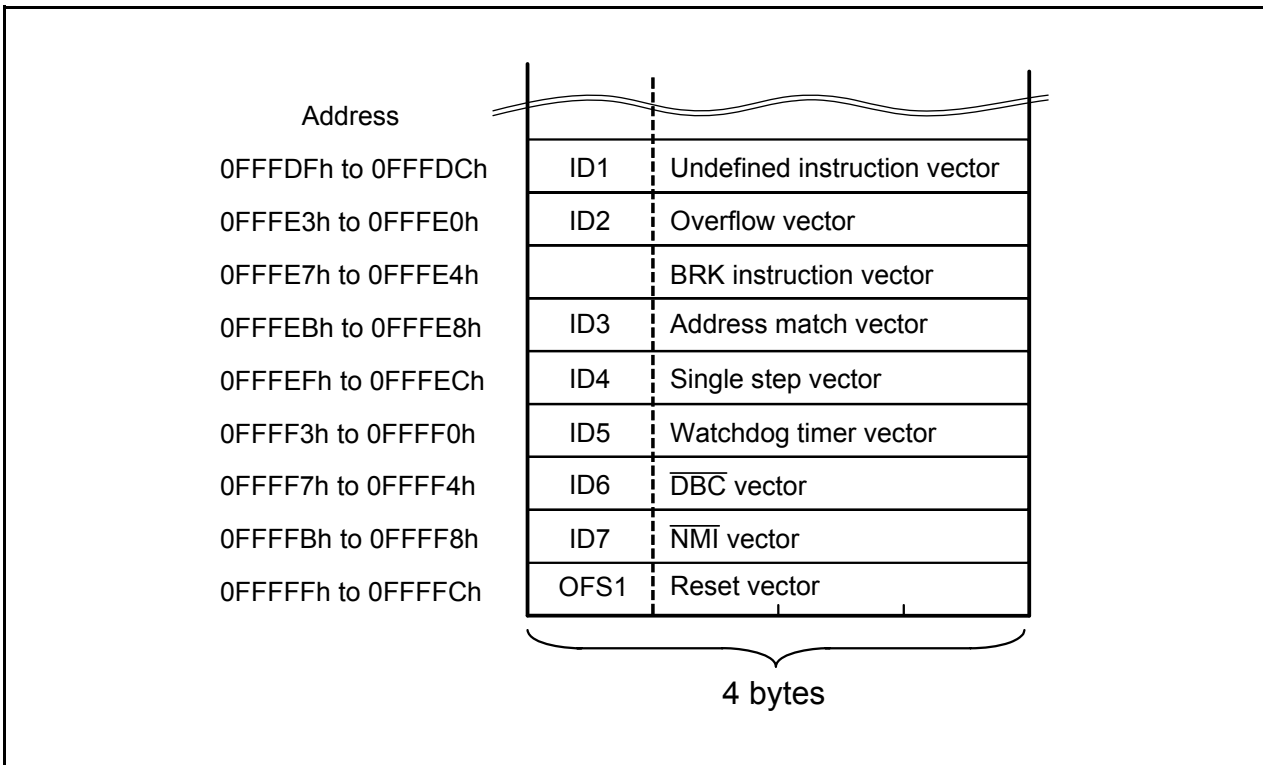


Figure 22.4 Address for ID Code Stored

## 22.3 CPU Rewrite Mode

In CPU rewrite mode, the flash memory can be rewritten when the CPU executes software commands. Program ROM 1, program ROM 2, and data flash can be rewritten with the microcomputer mounted on a board without using a ROM programmer.

The program and block erase commands are executed only in each block area of program ROM 1, program ROM 2, and data flash.

Erase-write 0 (EW0) mode and erase-write 1 (EW1) mode are provided as CPU rewrite mode. Table 22.9 lists differences between erase-write 0 (EW0) and erase-write 1 (EW1) modes.

**Table 22.9 EW0 Mode and EW1 Mode**

Item	EW0 Mode	EW1 Mode
Operating Mode	<ul style="list-style-type: none"> <li>• Single-chip mode</li> <li>• Memory expansion mode</li> </ul>	Single-chip mode
Rewrite Control Program Allocatable Area	<ul style="list-style-type: none"> <li>• Program ROM 1</li> <li>• Program ROM 2</li> </ul>	<ul style="list-style-type: none"> <li>• Program ROM 1</li> <li>• Program ROM 2</li> </ul>
Rewrite Control Program Executable Area	The rewrite control program must be transferred to any area other than the flash memory (e.g., RAM) before being executed <sup>(2)</sup>	The rewrite control program can be executed in program ROM 1, program ROM 2, and data flash.
Rewritable Area	<ul style="list-style-type: none"> <li>• Program ROM 1</li> <li>• Program ROM 2</li> <li>• Data flash</li> </ul>	Program ROM 1, program ROM 2, and data flash, excluding blocks with the rewrite control program
Software Command Restriction	None	<ul style="list-style-type: none"> <li>• Program and block erase commands cannot be executed in a block having the rewrite control program.</li> <li>• Read status register command cannot be used.</li> </ul>
Mode after Program or Erase	Read status register mode	Read array mode
CPU State during Auto Write and Auto Erase	Operating	Maintains hold state (I/O ports maintains the state before the command execution) <sup>(1)</sup>
Flash Memory Status Detection	<ul style="list-style-type: none"> <li>• Read bits FMR00, FMR06, and FMR07 in the FMR0 register by program</li> <li>• Execute the read status register command to read bits SR7, SR5, and SR4 in the status register.</li> </ul>	Read bits FMR00, FMR06, and FMR07 in the FMR0 register by program

**NOTES:**

1. Do not generate an interrupt (except  $\overline{NMI}$  interrupt) or start a DMA transfer.
2. When in CPU rewrite mode, bits PM10 and PM13 in the PM1 register are set to 1. The rewrite control program can only be executed in the internal RAM or in an external area that is enabled for use when the PM13 bit = 1. When the PM13 bit = 0 and the flash memory is used in 4-Mbyte mode, the extended accessible area (40000h to BFFFFh) cannot be used.

### 22.3.1 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 accept commands. EW0 mode is selected by setting the FMR60 bit in the FMR6 register to 0. Figure 22.7 shows setting and resetting of EW0 mode.

The software commands control programming and erasing. The FMR0 register or the status register indicates whether a program or erase operation is completed as expected or not.

### 22.3.2 EW1 Mode

EW1 mode is selected by setting the FMR60 bit to 1 after setting the FMR01 bit to 1. Figure 22.8 shows setting and resetting of EW1 mode.

The FMR0 register indicates whether or not a program or erase operation has been completed as expected. The status register cannot be read in EW1 mode.

When a program / erase operation is initiated, the CPU halts all program execution until the operation is completed.

### 22.3.3 Flash Memory Control Register (Registers FMR0, FMR1, FMR2 and FMR6)

Figures 22.5 to 22.8 show the registers FMR0, FMR1, FMR2 and FMR6, respectively.

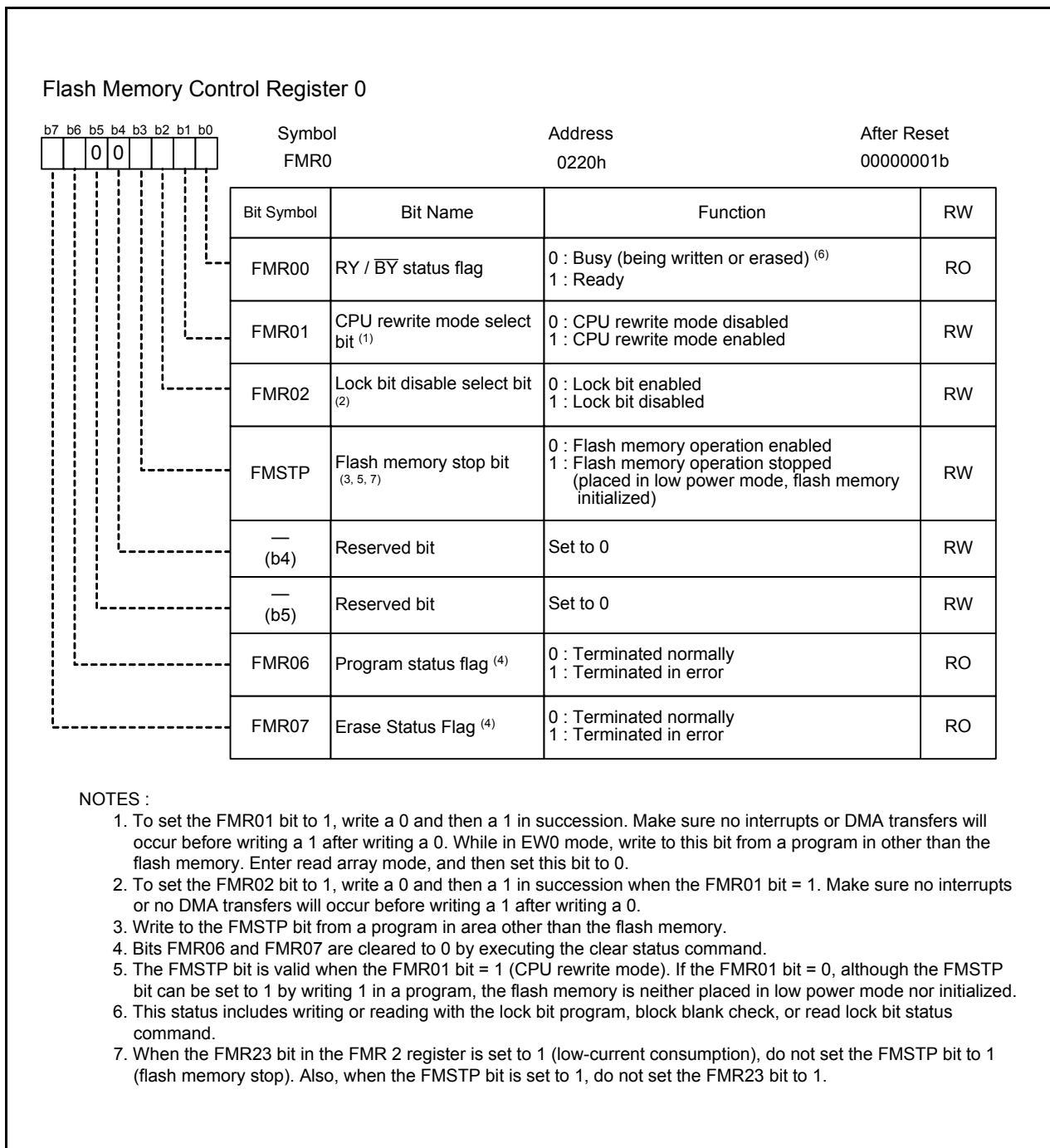
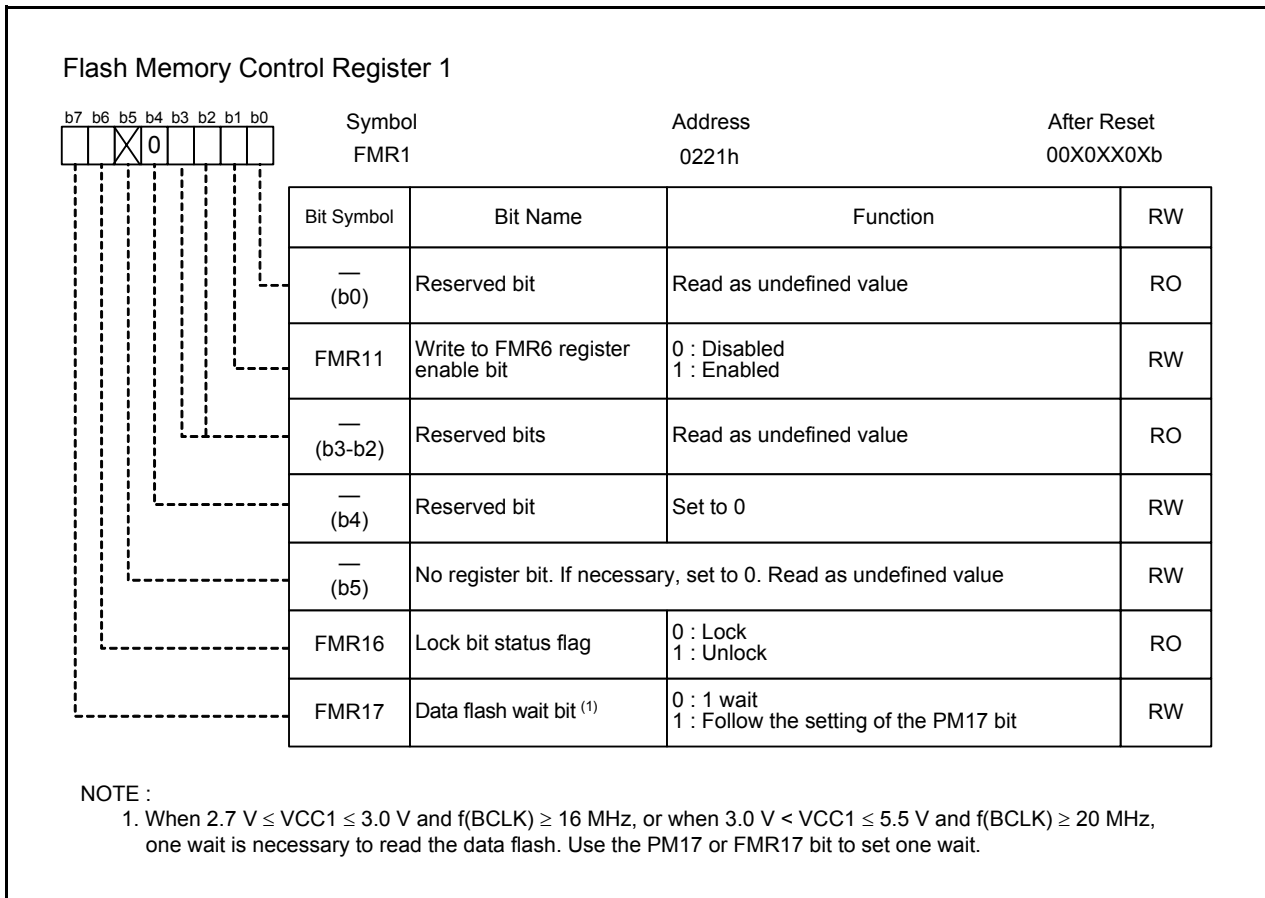
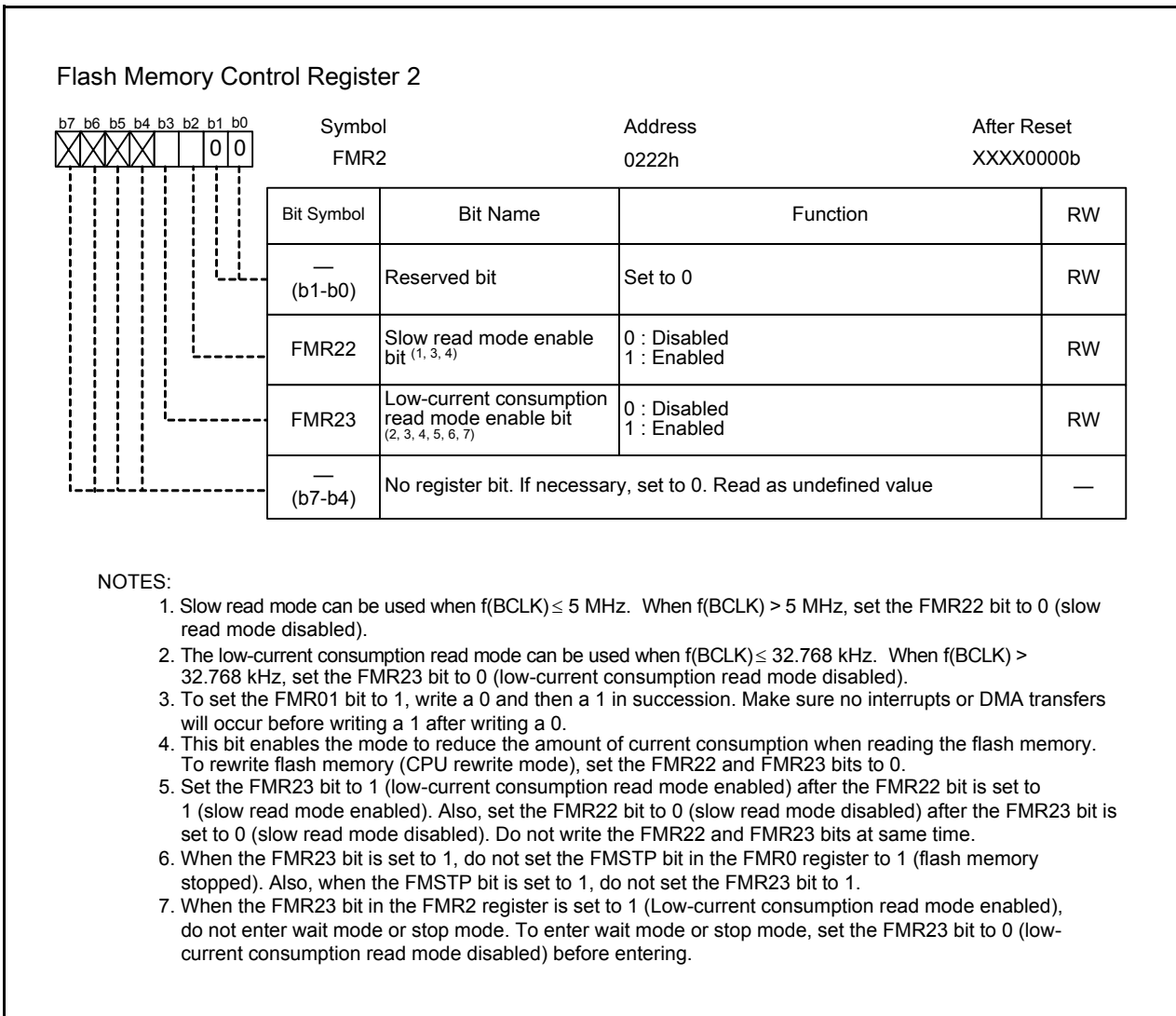


Figure 22.5 FMR0 Register



**Figure 22.6 FMR1 Register**



**Figure 22.7 FMR2 Register**

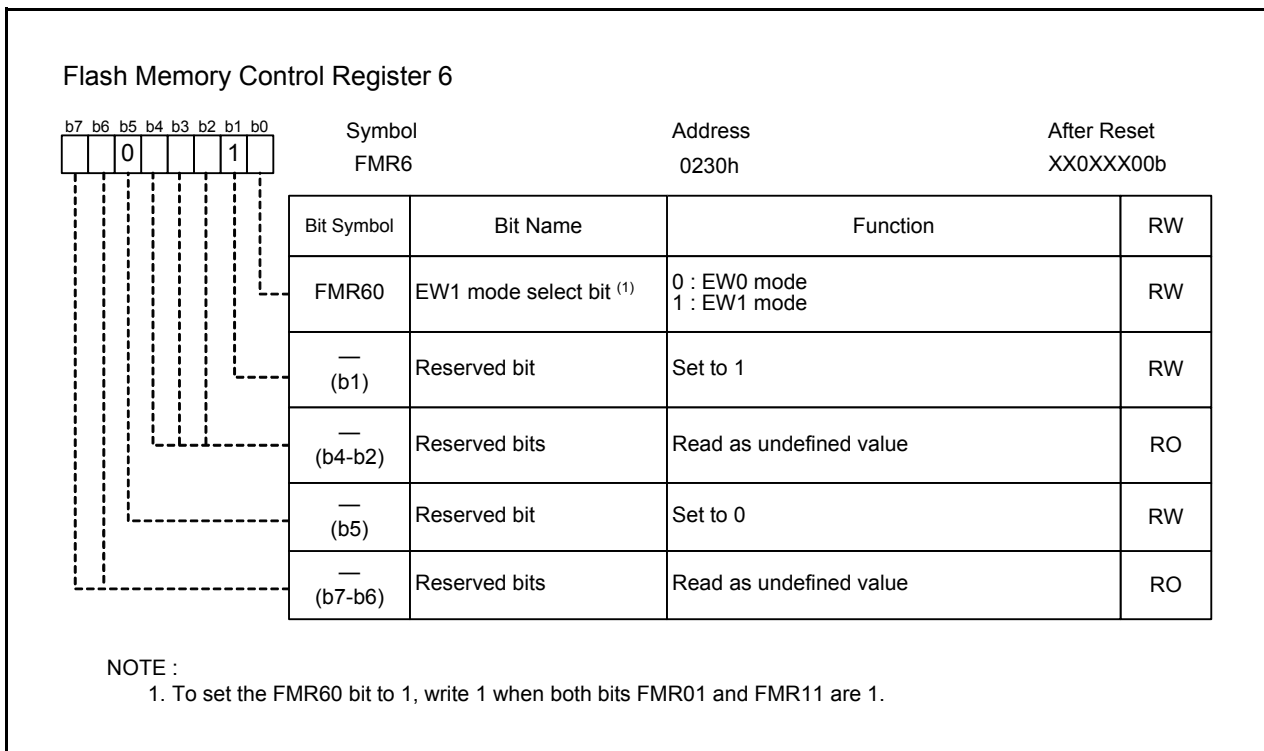


Figure 22.8 FMR6 Register



### 22.3.3.1 FMR00 Bit

This bit indicates the flash memory operating state. It is set to 0 while the program, block erase, lock bit program, read lock bit status command, or block blank check command is being executed; otherwise, it is set to 1.

### 22.3.3.2 FMR01 Bit

The microcomputer can accept commands when the FMR01 bit is set to 1 (CPU rewrite mode).

### 22.3.3.3 FMR02 Bit

The lock bit is disabled by setting the FMR02 bit to 1 (lock bit disabled). (Refer to **22.3.6 “Data Protect Function”**.) The lock bit is enabled by setting the FMR02 bit to 0 (lock bit enabled).

The FMR02 bit does not change the lock bit status but disables the lock bit function. If an erase command is executed when the FMR02 bit is set to 1, the lock bit status changes 0 (locked) to 1 (unlocked) after command execution is completed.

### 22.3.3.4 FMSTP Bit

The FMSTP 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 (flash memory stops). Set the FMSTP bit by program in an area other than the flash memory.

Set the FMSTP bit to 1 if one of the followings occurs:

- A flash memory access error occurs while erasing or programming in EW0 mode (the FMR00 bit does not switch back to 1 (ready)).
- Low-power consumption mode or on-chip oscillator low-power consumption mode is entered

Use the following steps to stop the flash memory.

- (1) Set the FMSTP bit to 1
- (2) Wait tps (the wait time to stabilize the flash memory circuit)

Use the following steps to restart.

- (1) Set the FMSTP bit to 0
- (2) Wait tps (the wait time to stabilize the flash memory circuit)

Figure 22.13 shows a Flow Chart Illustrating How to Start and Stop the Flash Memory Processing Before and After Low-Power Consumption Mode or On-Chip Oscillator Low-Power Consumption Mode. Follow the procedure on this flow chart.

When entering stop or wait mode, the flash memory is automatically turned off. When exiting stop or wait mode, the flash memory is turned back on. The FMR0 register does not need to be set.

### 22.3.3.5 FMR06 Bit

This is a read-only bit indicating an auto program operation state. The FMR06 bit is set to 1 when a program error occurs; otherwise, it is set to 0. Refer to **22.3.8 “Full Status Check”**.

### 22.3.3.6 FMR07 Bit

This is a read-only bit indicating the auto erase operation status. The FMR07 bit is set to 1 when an erase error occurs; otherwise, it is set to 0. The FMR07 bit is also used for blank check. For details, refer to **22.3.8 “Full Status Check”**.

### 22.3.3.7 FMR11 Bit

The FMR11 bit enables programming to the FMR6 register.

### 22.3.3.8 FMR16 Bit

This is a read-only bit indicating the execution result of the read lock bit status command. When the block, where the read lock bit status command is executed, is locked, the FMR16 bit is set to 0. When the block, where the read lock bit status command is executed, is unlocked, the FMR16 bit is set to 1.

### 22.3.3.9 FMR17 Bit

This is a bit to select wait state for data flash.

### 22.3.3.10 FMR22 Bit

This bit enables the mode to reduce the amount of current consumption when reading the flash memory. When rewriting the flash memory (CPU rewrite mode), set the FMR22 bit to 0 (slow read mode disabled).

Also, when  $f(\text{BCLK}) > 5 \text{ MHz}$ , set the FMR22 bit to 0 (slow read mode disabled).

Figure 22.9 shows setting and resetting of the slow read mode.

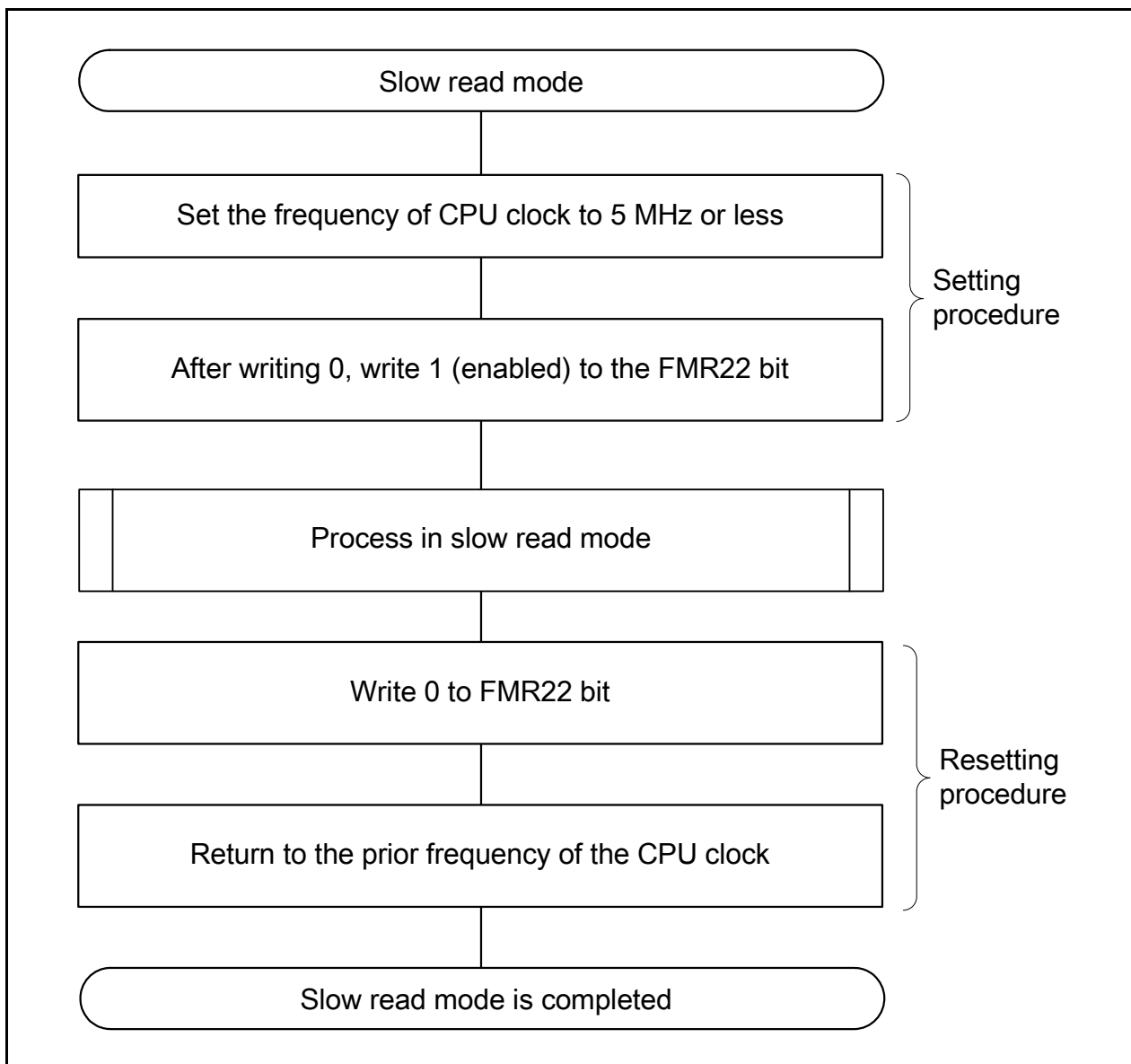


Figure 22.9 Setting and Resetting of Slow Read Mode

### 22.3.3.11 FMR23 Bit

This bit enables the mode to reduce the amount of current consumption when reading the flash memory. When rewriting the flash memory (CPU rewrite mode), set the FMR23 bit to 0 (low-current consumption read mode disabled).

This bit is effective when the FMR22 bit is enabled. When  $f(\text{BCLK}) > 32.768 \text{ kHz}$ , set the FMR23 bit to 0 (low-current consumption read mode disabled).

Figure 22.10 shows setting and resetting of the low-current consumption read mode.

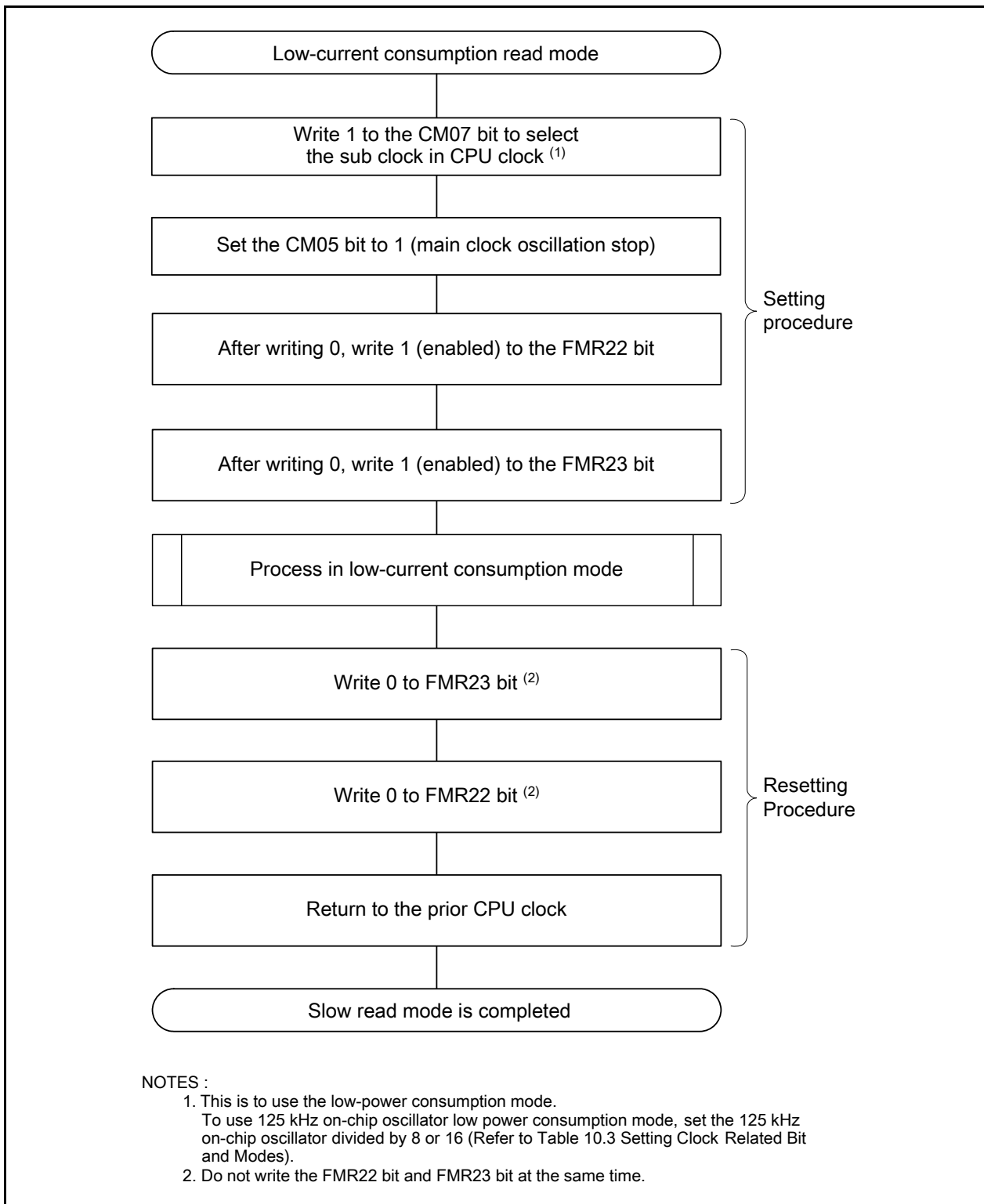


Figure 22.10 Setting and Resetting of Low-current Consumption Read Mode

### 22.3.3.12 FMR60 Bit

This bit is used to select EW1 mode when the FMR01 bit is set to 1 (CPU rewrite mode enabled).

Figure 22.11 shows Setting and Resetting of EW0 Mode. Figure 22.12 shows Setting and Resetting of EW1 Mode.

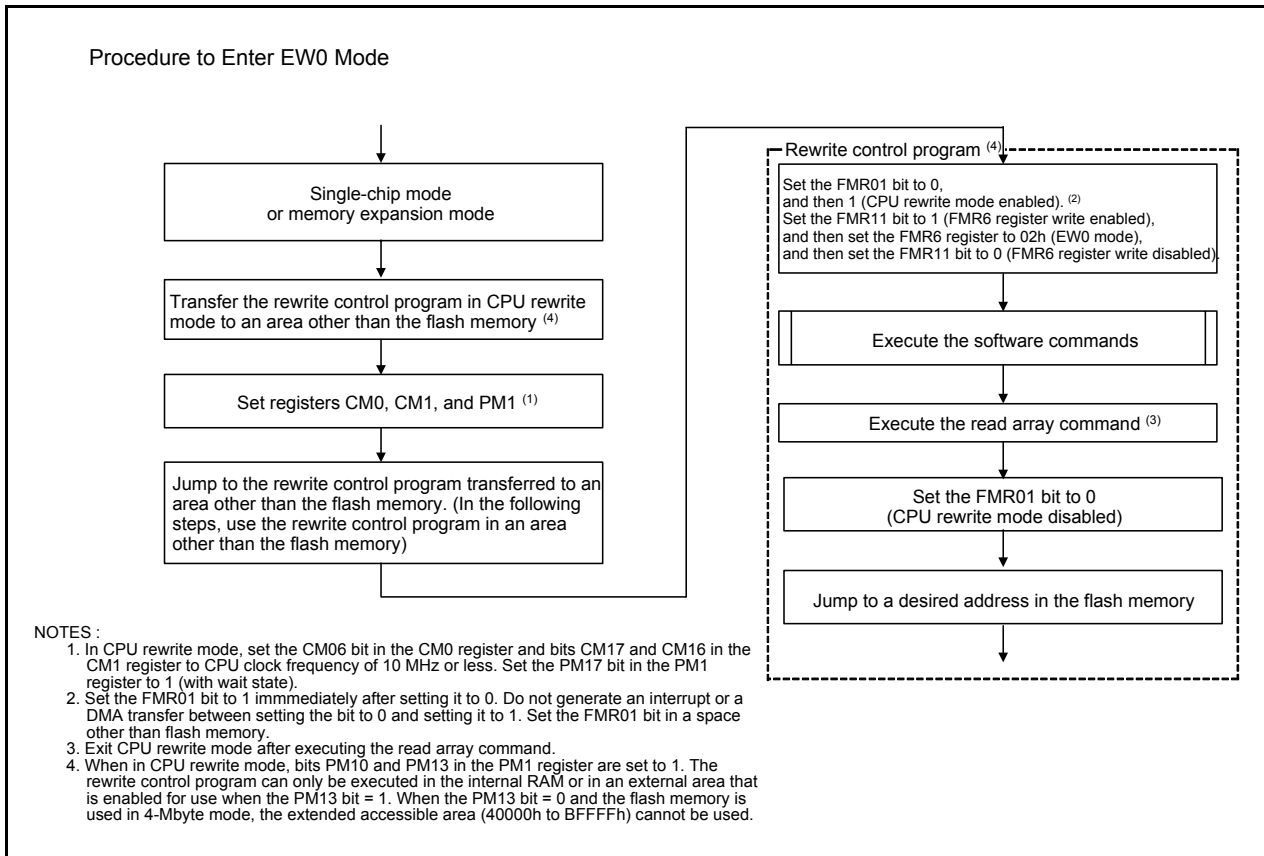


Figure 22.11 Setting and Resetting of EW0 Mode

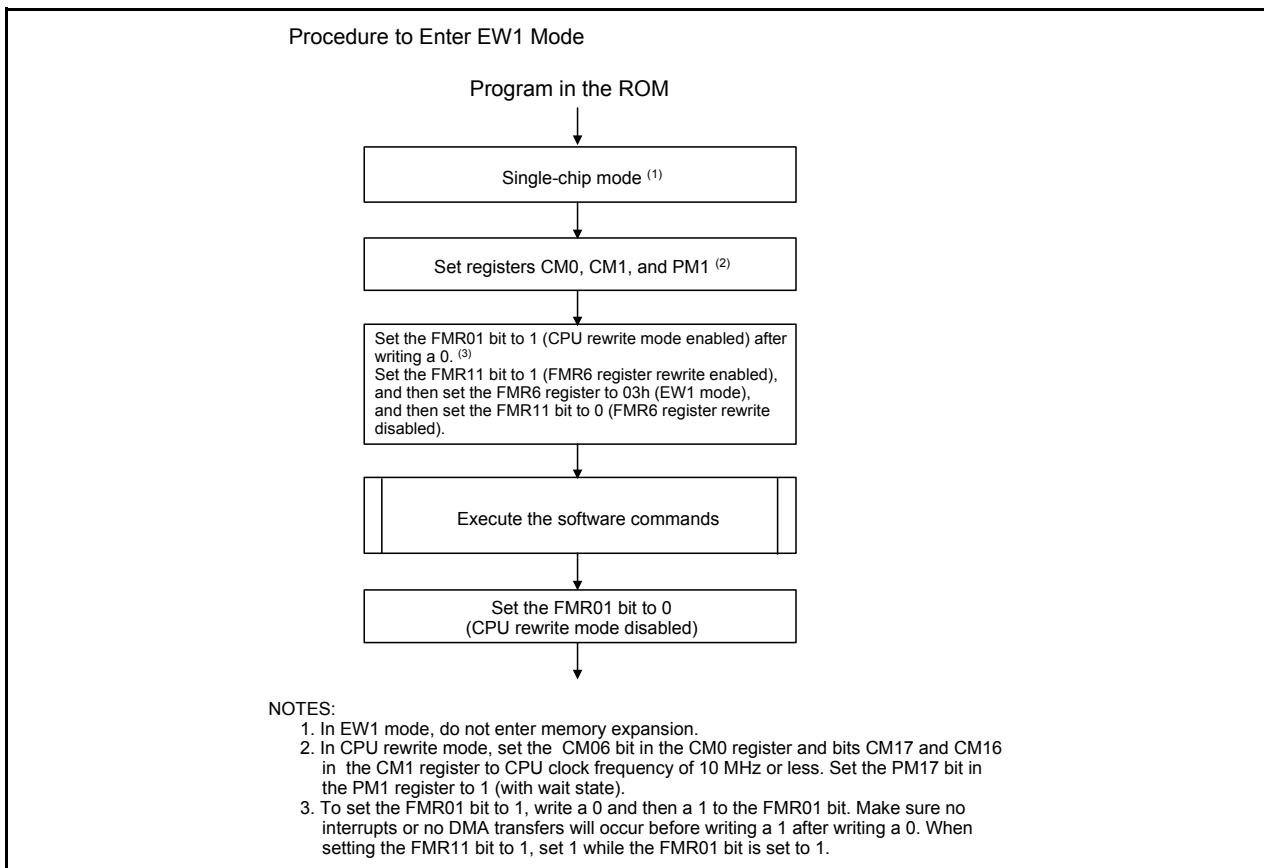


Figure 22.12 Setting and Resetting of EW1 Mode

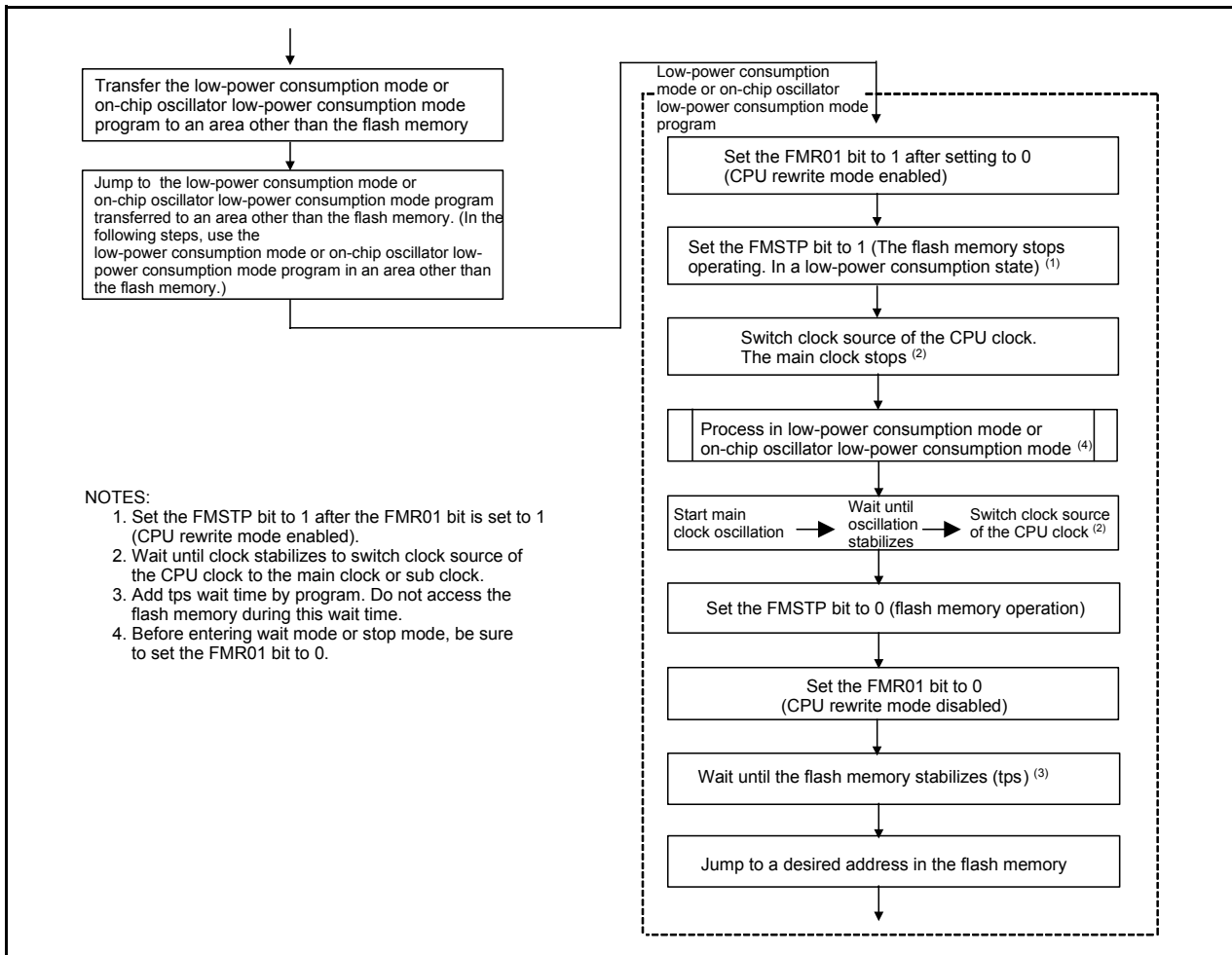


Figure 22.13 Processing Before and After Low-Power Consumption Mode or On-Chip Oscillator Low-Power Consumption Mode

## 22.3.4 Precautions on CPU Rewrite Mode

### 22.3.4.1 Operating Speed

Set the CM06 bit in the CM0 register and bits CM17 and CM16 in the CM1 register to a CPU clock frequency of 10 MHz or less before entering CPU rewrite mode (EW0 or EW1 mode). Also, set the PM17 bit in the PM1 register to 1 (wait state).

### 22.3.4.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because the CPU tries to read data in the flash memory: the UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction.

### 22.3.4.3 Interrupts (EW0 mode)

- To use interrupts with vectors in a relocatable vector table, relocate the vectors to the RAM area.
- The  $\overline{\text{NMI}}$  and watchdog timer interrupts are available since registers FMR0 and FMR1 are forcibly reset when either interrupt occurs. Allocate the jump addresses for each interrupt routine to the fixed vector table. Flash memory rewrite operation stops when the  $\overline{\text{NMI}}$  or watchdog timer interrupt occurs. Execute the rewrite program again after exiting the interrupt routine.
- The address match interrupt is not available since the CPU tries to read data in the flash memory.

### 22.3.4.4 Interrupts (EW1 mode)

- Do not acknowledge any interrupts with vectors in a relocatable vector table or address match interrupt during the auto program or auto erase period.
- Do not use the watchdog timer interrupt.
- The  $\overline{\text{NMI}}$  interrupt is available since registers FMR0 and FMR1 are forcibly reset when the interrupt occurs. Allocate the jump address for the interrupt routine to the fixed vector table. Flash memory rewrite operation stops when the  $\overline{\text{NMI}}$  interrupt occurs. Execute the rewrite program again after exiting the interrupt routine.

### 22.3.4.5 How to Access

To set the FMR01 or FMR02 bit to 1, write a 1 after first setting the bit to 0. Make sure that no interrupts or no DMA transfers will occur before writing a 1 after writing a 0.

### 22.3.4.6 Rewrite (EW0 mode)

If the supply voltage drops while rewriting the block where the rewrite control program is stored, the rewrite control program is not correctly rewritten. This may cause the flash memory not to be rewritten. If this error occurs, use standard serial I/O mode or parallel I/O mode for rewriting.

### 22.3.4.7 Rewrite (EW1 mode)

Do not rewrite any block in which the rewrite control program is stored.

### 22.3.4.8 DMA Transfer

In EW1 mode, do not generate a DMA transfer while the FMR00 bit in the FMR0 register is set to 0 (auto programming or auto erasing).

### 22.3.4.9 Writing Command and Data

Write commands and data to even addresses.

### 22.3.4.10 Wait Mode

When entering wait mode, set the FMR01 bit to 0 (CPU rewrite mode disabled) before executing the WAIT instruction.

### 22.3.4.11 Stop Mode

To enter stop mode, set the FMR01 bit to 0 (CPU rewrite mode disabled), and then disable DMA transfer before setting the CM10 bit to 1 (stop mode).

### 22.3.4.12 Low-Power Consumption Mode and On-Chip Oscillator Low-power Consumption Mode

When the CM05 bit is set to 1 (main clock stopped), do not execute the following commands:

- Program
- Block erase
- Lock bit program
- Read lock bit status
- Block blank check



### 22.3.5 Software Commands

Software commands are described below. Read and write the command code and data in 16-bit units, from and to even addresses in the program ROM 1, program ROM 2, and data flash. When the command code is written, the 8 high-order bits (D15 to D8) are ignored.

**Table 22.10 Software Commands**

Command	First Bus Cycle			Second Bus Cycle			Third Bus Cycle		
	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)
Read Array	Write	x	xxFFh						
Read Status Register	Write	x	xx70h	Read	x	SRD			
Clear Status Register	Write	x	xx50h						
Program	Write	WA0	xx41h	Write	WA0	WD0	Write	WA1	WD1
Block Erase	Write	x	xx20h	Write	BA	xxD0h			
Lock Bit Program	Write	BA	xx77h	Write	BA	xxD0h			
Read Lock Bit Status	Write	x	xx71h	Write	BA	xxD0h			
Block Blank Check	Write	x	xx25h	Write	BA	xxD0h			

- SRD : Data in the status register (D7 to D0)
- WA0 : Address which low-order words are written (The address specified in the first bus cycle is the same even address as the address specified in the second bus cycle.)
- WA1 : Address which high-order words are written
- WD0 : Write data low-order word (16 bits)
- WD1 : Write data high-order word (16 bits)
- BA : Highest-order block address (even address)
- x : Given even address in the program ROM 1, program ROM 2, and data flash
- xx : Eight high-order bits of command code (ignored)

#### 22.3.5.1 Read Array Command

The read array command reads the flash memory. By writing the command code xxFFh in the first bus cycle, read array mode is entered. Content of a specified address can be read in 16-bit units by entering an address to be read after the next bus cycle. The microcomputer remains in read array mode until another command is written. Therefore, contents from multiple addresses can be read consecutively.

#### 22.3.5.2 Read Status Register Command

The read status register command reads the status register. By writing the command code xx70h in the first bus cycle, the status register can be read in the second bus cycle (refer to **22.3.7 “Status Register”**). Read an even address in the program ROM 1, program ROM 2, and data flash. Do not execute this command in EW1 mode.

### 22.3.5.3 Clear Status Register Command

The clear status register command clears the status register. By writing xx50h in the first bus cycle, bits FMR07 and FMR06 in the FMR0 register are set to 00b, and bits SR5 and SR4 in the status register are set to 00b.

### 22.3.5.4 Program Command

The program command writes 2-word (4 bytes) data to the flash memory. By writing xx41h in the first bus cycle and data to the write address in the second and third bus cycles, an auto program operation (data program and verify) will start. The address value specified in the first bus cycle must be the same even address as the write address specified in the second bus cycle.

The FMR00 bit in the FMR0 register indicates whether an auto program operation has been completed. The FMR00 bit is set to 0 (busy) during auto program and to 1 (ready) while in an auto program operation.

After the completion of an auto program operation, the FMR06 bit in the FMR0 register indicates whether or not the auto program operation has been completed as expected. (Refer to 22.3.8 “Full Status Check”.)

An address that is already written cannot be altered or rewritten. Figure 22.14 shows a Flow Chart of the Program Command Programming.

The lock bit protects each block from being programmed inadvertently. (Refer to 22.3.6 “Data Protect Function”.)

In EW1 mode, do not execute this command on the block to which the rewrite control program is allocated.

In EW0 mode, the microcomputer enters read status register mode as soon as an auto program operation starts. The status register can be read. The SR7 bit in the status register is set to 0 at the same time an auto program operation starts. It is set to 1 when the auto program operation is completed. The microcomputer remains in read status register mode until the read array command is written. After completion of an auto program operation, the status register indicates whether or not the auto program operation has been completed as expected.

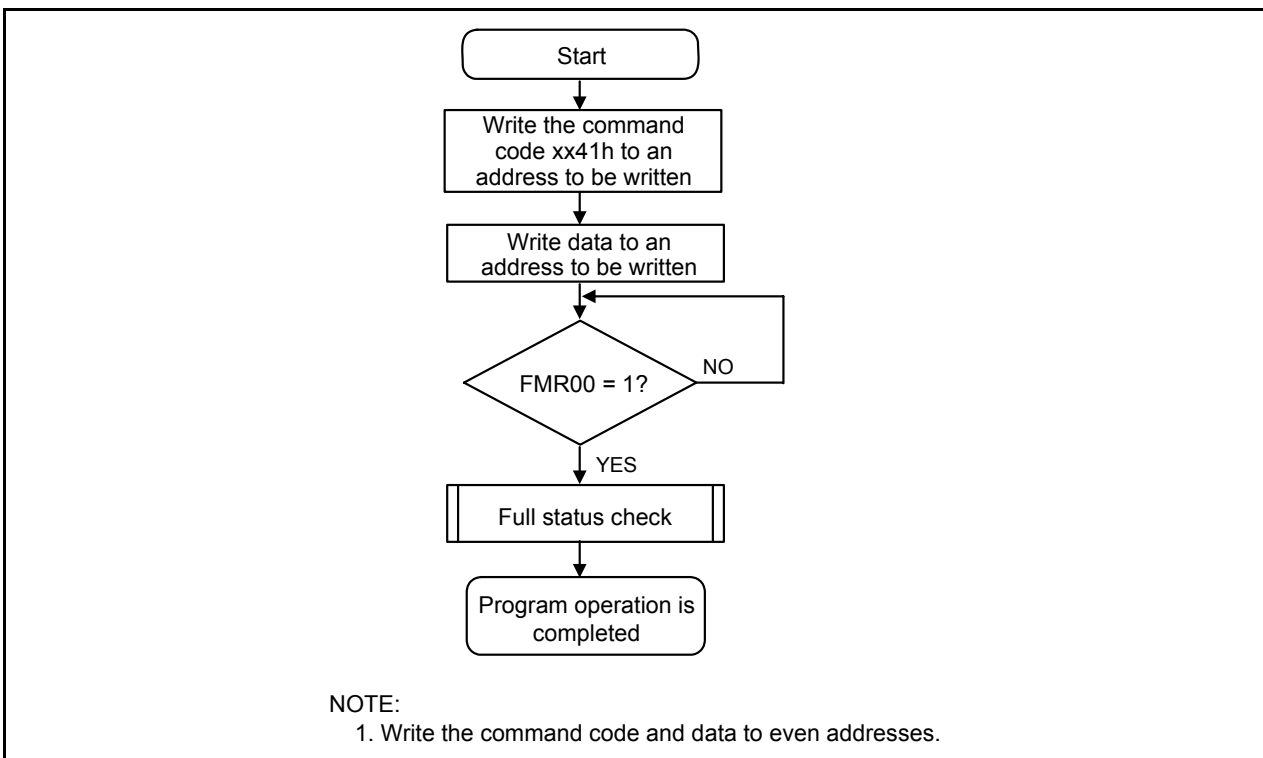


Figure 22.14 Program Command

### 22.3.5.5 Block Erase Command

By writing xx20h in the first bus cycle and xxD0h in the second bus cycle to the highest-order even address of a block, an auto erase operation (erase and verify) will start in the specified block.

The FMR00 bit in the FMR0 register indicates whether an auto erase operation has been completed. The FMR00 bit is set to 0 (busy) during auto erase and to 1 (ready) when the auto erase operation is completed.

After the completion of an auto erase operation, the FMR07 bit in the FMR0 register indicates whether or not the auto erase operation has been completed as expected. (Refer to 22.3.8 “Full Status Check”.)

Figure 22.15 shows a Block Erase Command.

The lock bit protects each block from being erased inadvertently. (Refer to 22.3.6 “Data Protect 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 an auto erase operation starts. The status register can be read. The SR7 bit in the status register is set to 0 at the same time an auto erase operation starts. It is set to 1 when an auto erase operation is completed. The microcomputer remains in read status register mode until the read array command or read lock bit status command is written. If an erase error occurs, execute the clear status register command and then block erase command at least 3 times until an erase error is not generated.

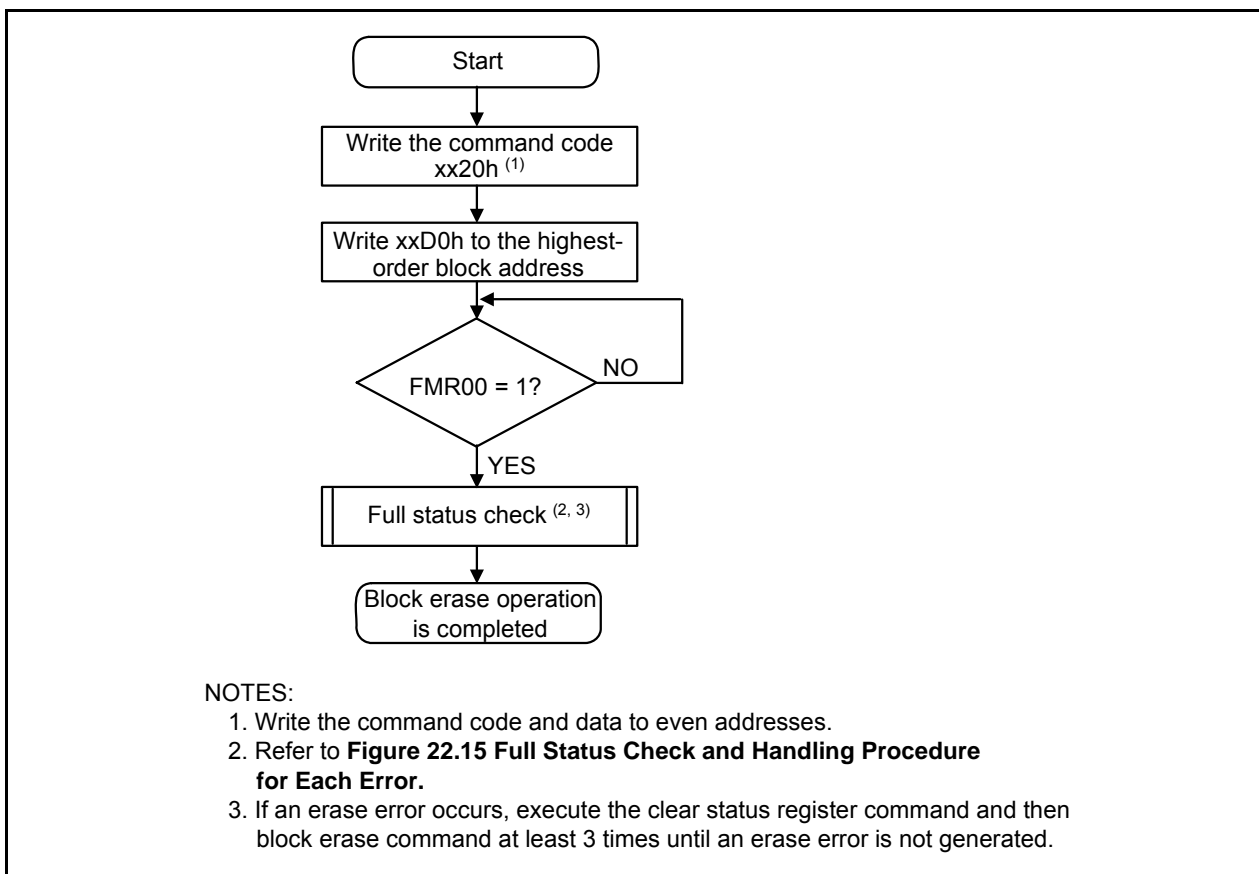


Figure 22.15 Block Erase Command

### 22.3.5.6 Lock Bit Program Command

The lock bit program command sets the lock bit for a specified block to 0 (locked).

By writing xx77h in the first bus cycle and xxD0h in the second bus cycle to the highest-order even address of a block, the lock bit for the specified block is set to 0. The address value specified in the first bus cycle must be the same highest-order address of a block specified in the second bus cycle.

Figure 22.16 shows a Flow Chart of the Lock Bit Program Command Programming. Execute read lock bit status command to read lock bit state (lock bit data).

The FMR00 bit in the FMR0 register indicates whether a lock bit program operation is completed.

Refer to **22.3.6 “Data Protect Function”** for details on lock bit functions and how to set it to 1 (unlocked).

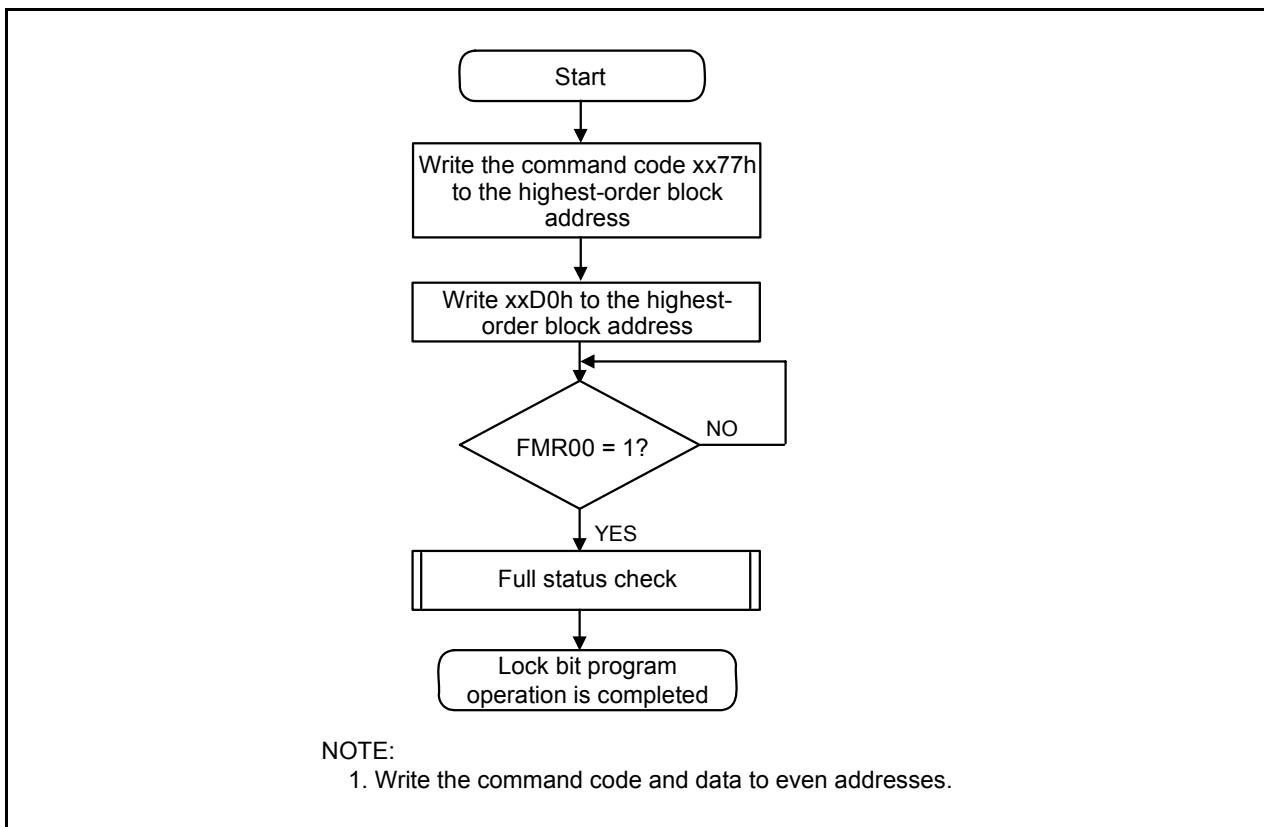


Figure 22.16 Lock Bit Program Command

### 22.3.5.7 Read Lock Bit Status Command

The read lock bit status command reads the lock bit state of a specified block.

By writing xx71h in the first bus cycle and xxD0h in the second bus cycle to the highest-order even address of a block, the FMR16 bit in the FMR1 register stores information on the lock bit status of a specified block. Read the FMR16 bit after the FMR00 bit in the FMR0 register is set to 1 (ready).

Figure 22.17 shows a Flow Chart of the Read Lock Bit Status Command Programming.

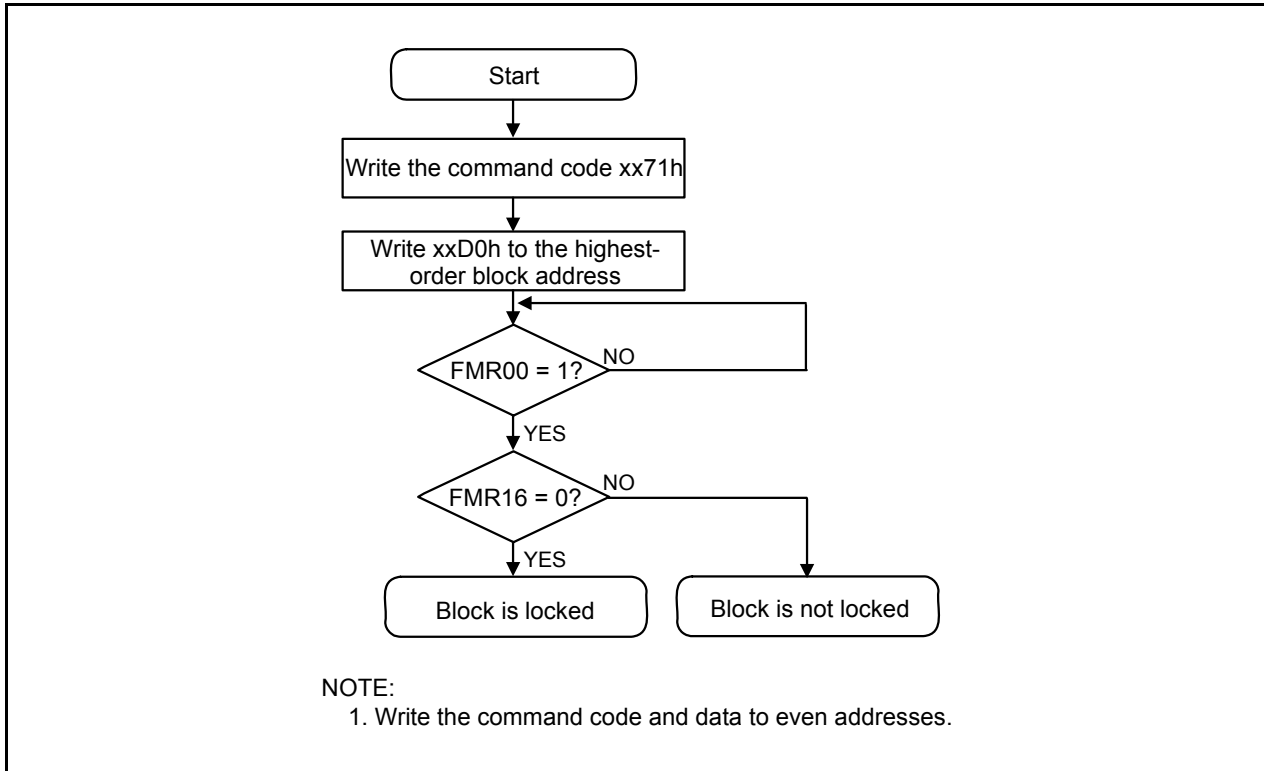


Figure 22.17 Read Lock Bit Status Command

### 22.3.5.8 Block Blank Check

The block blank check command checks whether or not a specified block is blank (state after erase). By writing xx25h in the first bus cycle and xxD0h in the second bus cycle to the highest-order even address of a block, the check result is stored in the FMR07 bit in the FMR0 register. Read the FMR07 bit after the FMR00 bit in the FMR0 register is set to 1 (ready).

The block blank check command is valid for unlocked blocks. If the block blank check command is executed to a block whose lock bit is 0 (locked), the FMR07 bit (SR5) is set to 1 (not blank) regardless of the FMR02 bit state.

Figure 22.18 shows a Flow Chart of the Block Blank Check Command Programming.

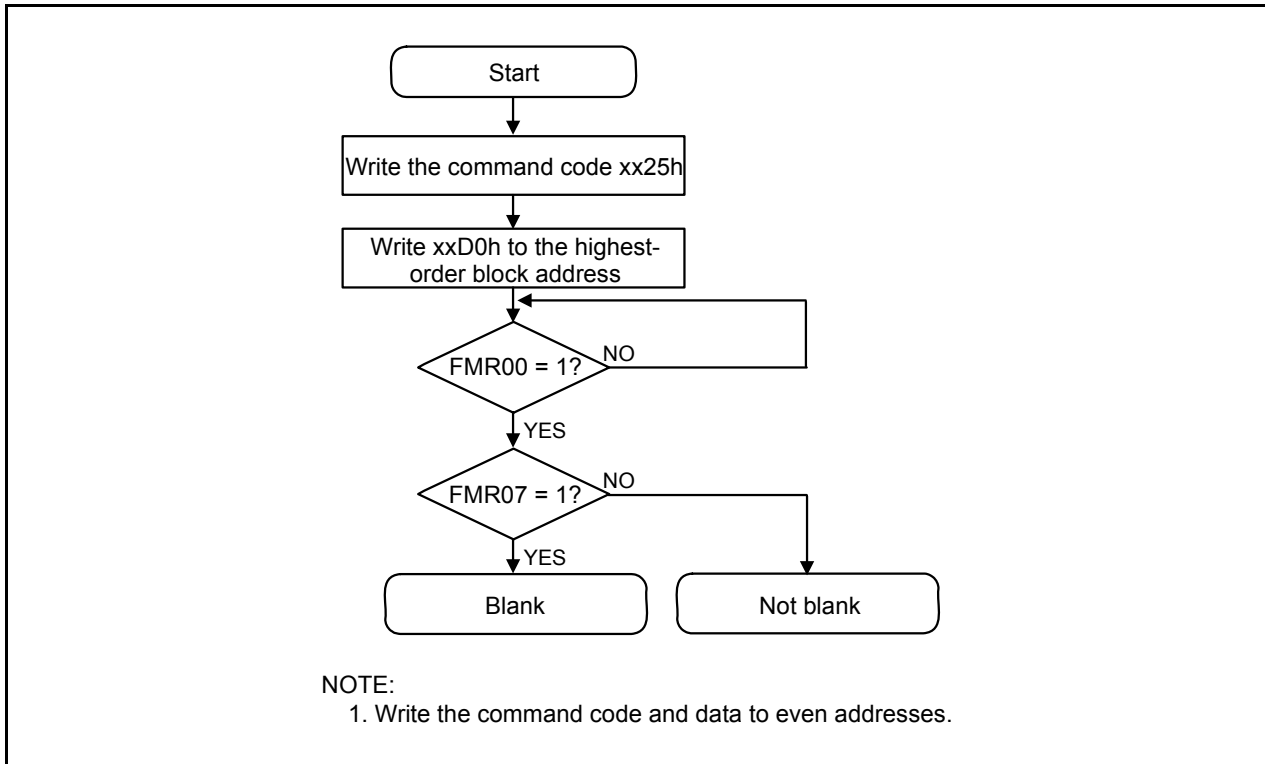


Figure 22.18 Block Blank Check Command

### 22.3.6 Data Protect Function

Each block in the flash memory has a nonvolatile lock bit. The lock bit is enabled by setting the FMR02 bit to 0 (lock bit enabled). The lock bit allows each block to be individually protected (locked) against program and erase. This prevents data from being inadvertently written to or erased from the flash memory.

A block changes its status according to the lock bit status:

- When the lock bit status is set to 0, the block is locked (block is protected against program and erase).
- When the lock bit status is set to 1, the block is not locked (block can be programmed or erased).

The lock bit status is set to 0 (locked) by executing the lock bit program command and to 1 (unlocked) by erasing the block. No commands can set the lock bit status to 1.

The lock bit status can be read by the read lock bit status command.

When the FMR02 bit is set to 1, the lock bit function is disabled, and all blocks are unlocked. However, individual lock bit status remains unchanged. The lock bit function is enabled by setting the FMR02 bit to 0. Lock bit status is retained.

If the block erase command is executed while the FMR02 bit is set to 1, the target block or all blocks are erased regardless of lock bit status. The lock bit status of each block is set to 1 after an erase operation is completed.

Refer to 22.3.5 “**Software Commands**” for details on each command.

### 22.3.7 Status Register

The status register indicates the flash memory operation state and whether or not an erase or program operation is completed as expected. Bits FMR00, FMR06, and FMR07 in the FMR0 register indicate status register states.

Table 22.11 shows the Status Register.

In EW0 mode, the status register can be read when the followings occur.

- Any even address in the program ROM 1, program ROM 2, or data flash is read after writing the read status register command.
- Any even address in the program ROM 1, program ROM 2, or data flash is read from when the program, block erase, lock bit program, or block blank check command is executed until when the read array command is executed.

#### 22.3.7.1 Sequence Status (Bits SR7 and FMR00)

The sequence status indicates the flash memory operation state. It is set to 0 while the program, block erase, lock bit program, block blank check, or read lock bit status command is being executed; otherwise, it is set to 1.

#### 22.3.7.2 Erase Status (Bits SR5 and FMR07)

Refer to 22.3.8 “**Full Status Check**”.

#### 22.3.7.3 Program Status (Bits SR4 and FMR06)

Refer to 22.3.8 “**Full Status Check**”.

**Table 22.11 Status Register**

Bits in Status Register	Bit in FMR0 Register	Status name	Definition		Value after Reset
			0	1	
SR0 (D0)	-	Reserved	-	-	-
SR1 (D1)	-	Reserved	-	-	-
SR2 (D2)	-	Reserved	-	-	-
SR3 (D3)	-	Reserved	-	-	-
SR4 (D4)	FMR06	Program status	Terminated normally	Terminated in error	0
SR5 (D5)	FMR07	Erase status	Terminated normally	Terminated in error	0
SR6 (D6)	-	Reserved	-	-	-
SR7 (D7)	FMR00	Sequencer status	Busy	Ready	1

D0 to D7 are the data buses read when the read status register command is executed.

Bits FMR07 (SR5) and FMR06 (SR4) are set to 0 when the clear status register command is executed.

When the FMR07 (SR5) or FMR06 bit (SR4) is set to 1, the program, block erase, lock bit program, block blank check, and read lock bit status commands are not accepted.



### 22.3.8 Full Status Check

If an error occurs when a program or erase operation is completed, bits FMR06 and FMR07 in the FMR0 register are set to 1, indicating a specific error. Therefore, execution results can be confirmed by checking these status (full status check).

Table 22.12 lists Errors and FMR0 Register State. Figure 22.19 shows a Full Status Check and Handling Procedure for Each Error.

**Table 22.12 Errors and FMR0 Register State**

FMR00 Register (Status Register) State		Error	Error Occurrence Conditions
FMR07 bit (SR5 bit)	FMR06 bit (SR4 bit)		
1	1	Command Sequence error	<ul style="list-style-type: none"> <li>• Command is written incorrectly</li> <li>• A value other than xxD0h or xxFFh is written in the second bus cycle of the lock bit program or block erase command (1)</li> </ul>
1	0	Erase error	<ul style="list-style-type: none"> <li>• The block erase command is executed on a locked block (2)</li> <li>• The block erase command is executed on an unlocked block, but auto erase operation is not completed as expected</li> <li>• The block blank check command is executed, and the check result is not blank</li> <li>• The block blank check command is executed on a locked block</li> </ul>
0	1	Program error	<ul style="list-style-type: none"> <li>• The program command is executed on a locked block (2)</li> <li>• The program command is executed on an unlocked block, but program operation is not completed as expected</li> <li>• The lock bit program command is executed, but the lock bit is not written as expected (2)</li> </ul>

**NOTES:**

1. The flash memory enters read array mode by writing command code xxFFh in the second bus cycle of the commands. The command code written in the first bus cycle becomes invalid.
2. When the FMR02 bit is set to 1 (lock bit disabled), no error occurs even under the conditions above.

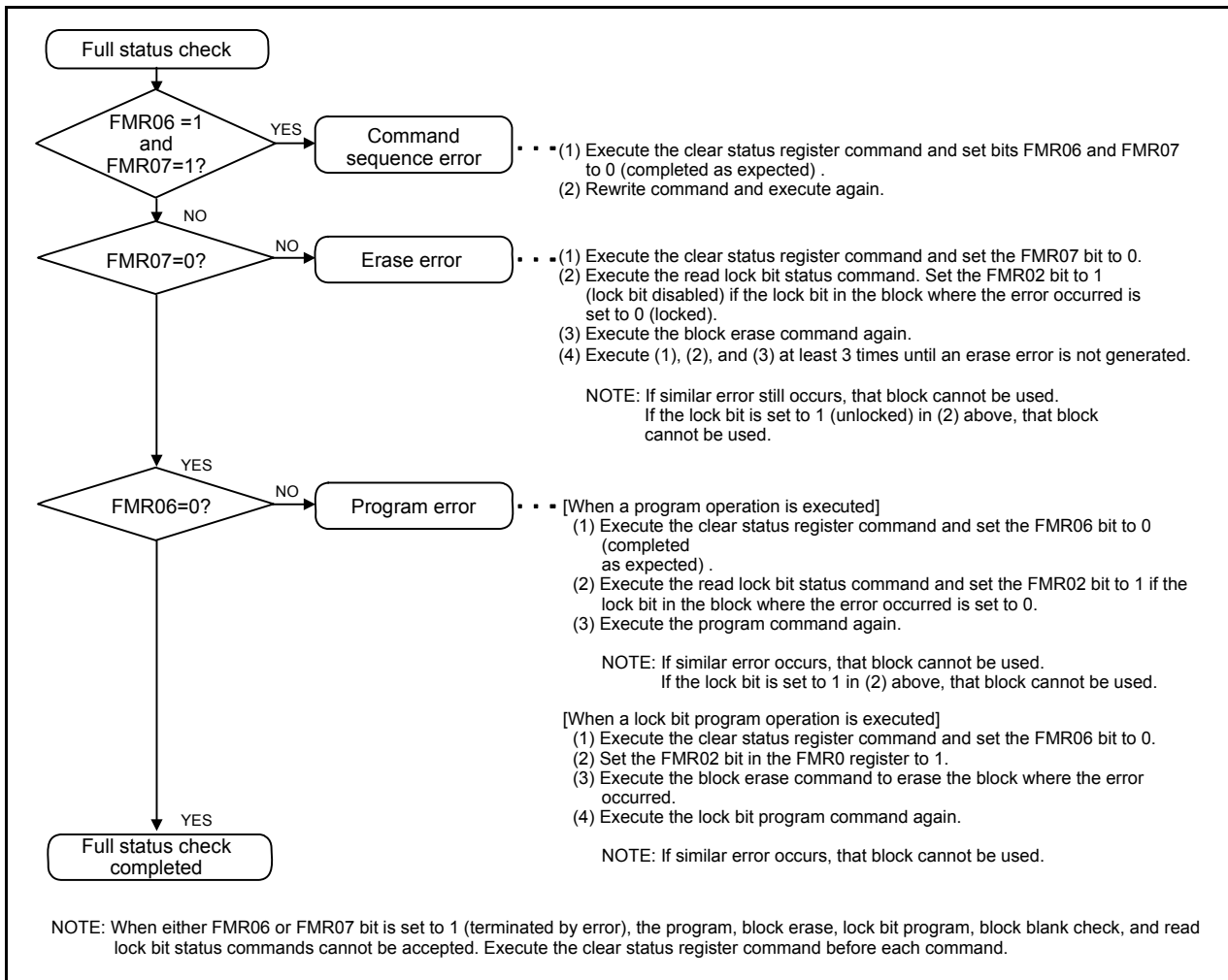


Figure 22.19 Full Status Check and Handling Procedure for Each Error

## 22.4 Standard Serial I/O Mode

In standard serial I/O mode, the serial programmer supporting the M16C/64 Group can be used to rewrite the program ROM 1, program ROM 2, and data flash in the microcomputer mounted on a board.

For more information about the serial programmer, contact your serial programmer manufacturer. Refer to the user's manual included with your serial programmer for instructions.

Table 22.13 lists Pin Functions (Flash Memory Standard Serial I/O Mode). Figures 22.20 and 22.21 show Pin Connections in Serial I/O Mode.

### 22.4.1 ID Code Check Function

The ID code check function determines whether the ID codes sent from the serial programmer match those written in the flash memory. (Refer to **22.2 “Functions to Prevent Flash Memory from Rewriting”**.)

Table 22.13 Pin Functions (Flash Memory Standard Serial I/O Mode)

Pin	Name	I/O	Power Supply	Description
VCC1, VCC2, VSS	Power input		-	Apply the flash program and erase voltage to the VCC1 pin, and VCC2 to the VCC2 pin. The VCC apply condition is that VCC2 = VCC1. Apply 0 V to the VSS pin.
CNVSS	CNVSS	I	VCC1	Connect to VCC1 pin.
RESET	Reset input	I	VCC1	Reset input pin. While the RESET pin is "L" level, input a 20-cycle or longer clock to the XIN pin.
XIN	Clock input	I	VCC1	Connect a ceramic resonator or crystal oscillator between pins XIN and XOUT. To input an externally generated clock, input it to the XIN pin and open the XOUT pin.
XOUT	Clock output	O	VCC1	
BYTE	BYTE input	I	VCC1	Connect this pin to VCC1 or VSS.
AVCC, AVSS	Analog power supply input			Connect AVSS to VSS and AVCC to VCC1, respectively.
VREF	Reference voltage input	I		Reference voltage input pin for A/D converter. Connect to VCC1.
P0_0 to P0_7	Input port P0	I	VCC2	Input "H" or "L" level signal or open.
P1_0 to P1_7	Input port P1	I	VCC2	Input "H" or "L" level signal or open.
P2_0 to P2_7	Input port P2	I	VCC2	Input "H" or "L" level signal or open.
P3_0 to P3_7	Input port P3	I	VCC2	Input "H" or "L" level signal or open.
P4_0 to P4_7	Input port P4	I	VCC2	Input "H" or "L" level signal or open.
P5_1 to P5_4, P5_6, P5_7	Input port P5	I	VCC2	Input "H" or "L" level signal or open.
P5_0	CE input	I	VCC2	Input "H" level signal.
P5_5	EPM input	I	VCC2	Input "L" level signal.
P6_0 to P6_3	Input port P6	I	VCC1	Input "H" or "L" level signal or open.
P6_4 / RTS1	BUSY output	O	VCC1	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: monitor signal output pin to check the boot program operation
P6_5/CLK1	SCLK input	I	VCC1	Standard serial I/O mode 1: serial clock input pin Standard serial I/O mode 2: Input "L".
P6_6 / RXD1	RXD input	I	VCC1	Serial data input pin.
P6_7 / TXD1	TXD input	O	VCC1	Serial data output pin. (1)
P7_0 to P7_7	Input port P7	I	VCC1	Input "H" or "L" level signal or open.
P8_0 to P8_3, P8_6, P8_7	Input port P8	I	VCC1	Input "H" or "L" level signal or open.
P8_4	P8_4 input	I	VCC1	Input "L" level signal. (2)
P8_5 / NMI	NMI input	I	VCC1	Input "H" or "L" level signal or open.
P9_0 to P9_7	Input port P9	I	VCC1	Input "H" or "L" level signal or open.
P10_0 to P10_7	Input port P10	I	VCC1	Input "H" or "L" level signal or open.

## NOTES:

- When using the standard serial I/O mode, the internal pull-up is enabled for the TXD1 (P6\_7) pin while the RESET pin is "L".
- When using the standard serial I/O mode, pins P0\_0 to P0\_7 and P1\_0 to P1\_7 may become indeterminate while the P8\_4 pin is "H" and the RESET pin is "L". If this causes a program, apply "L" to the P8\_4 pin.

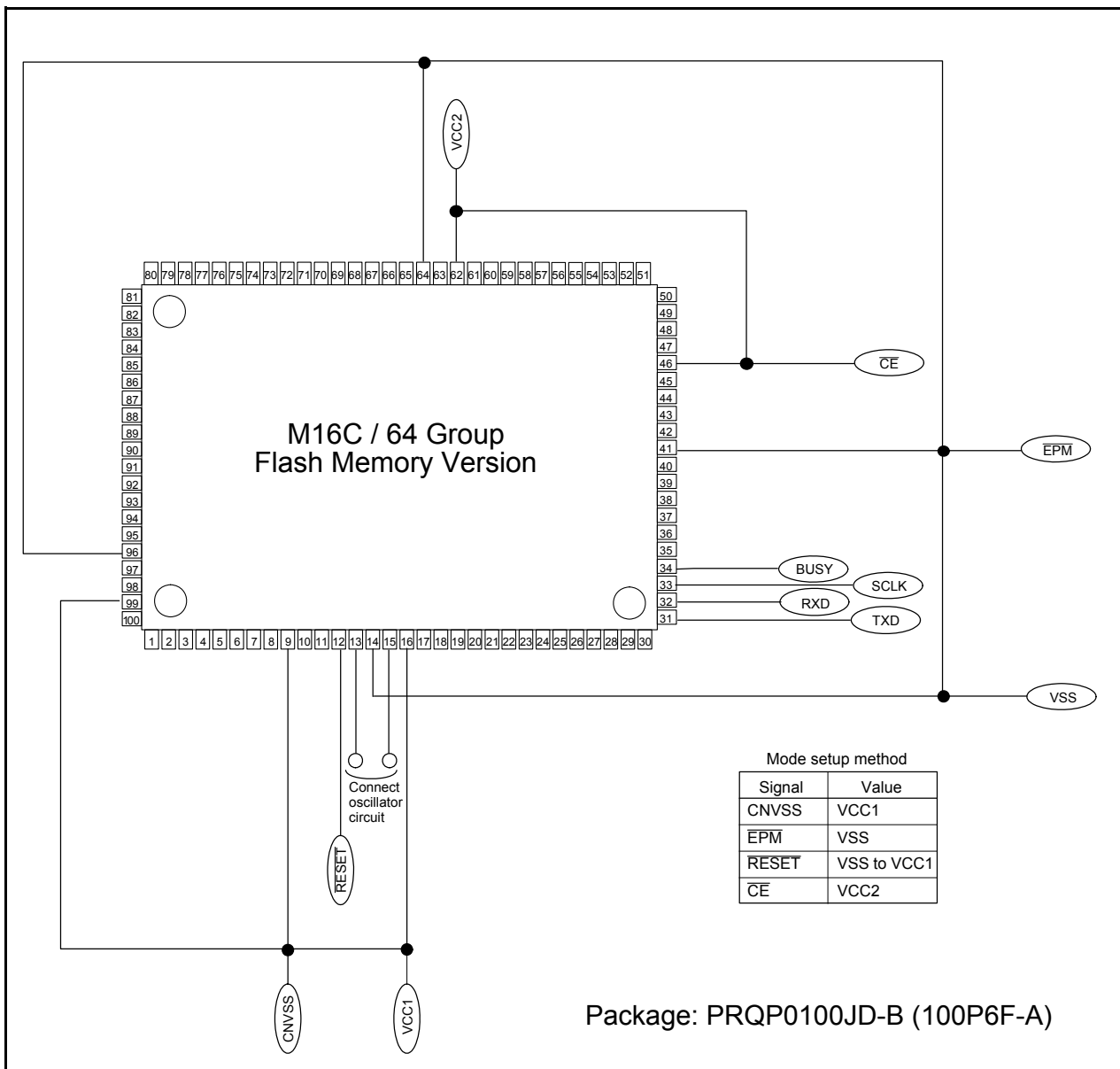


Figure 22.20 Pin Connections for Standard Serial I/O Mode (1)

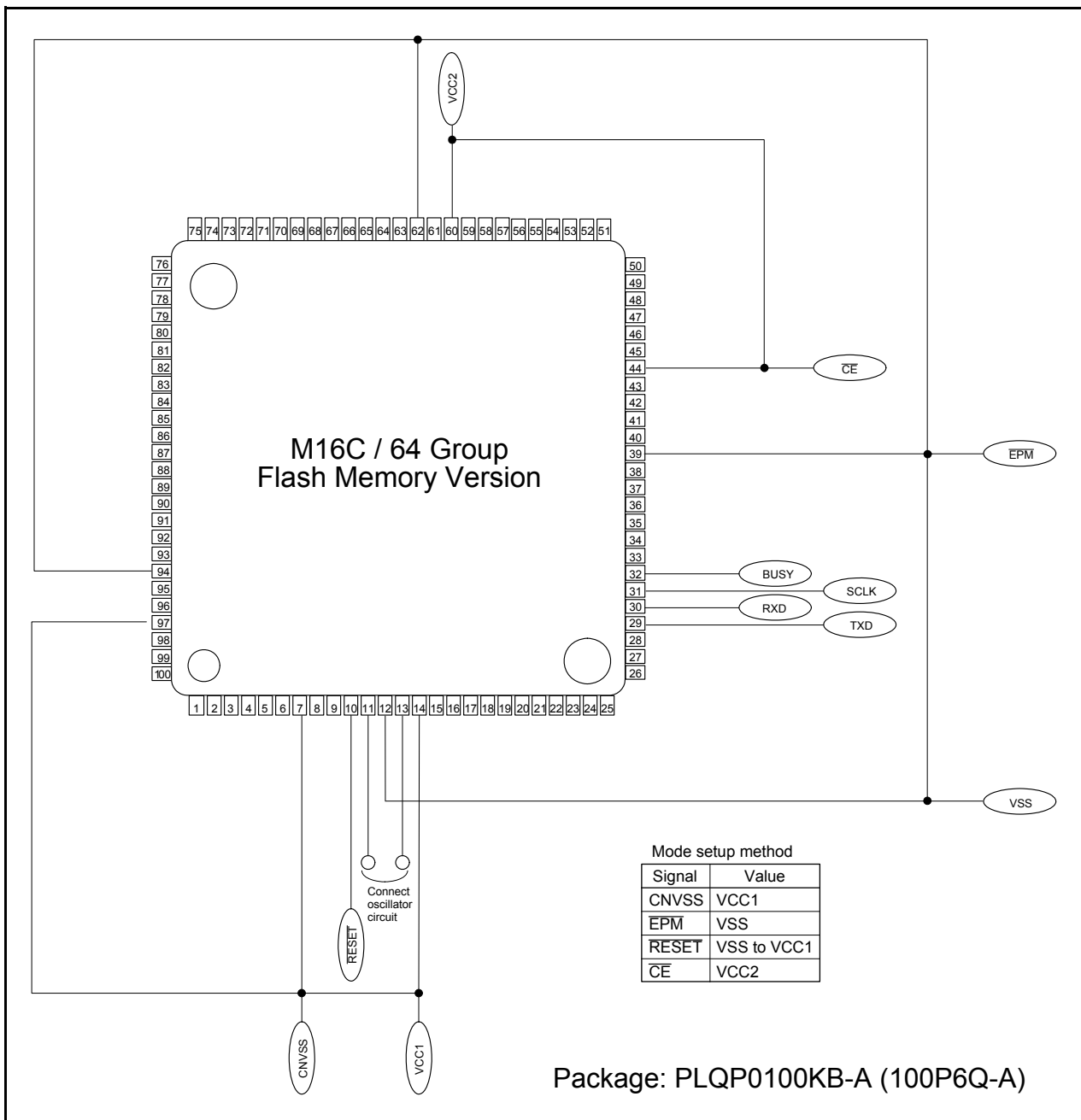


Figure 22.21 Pin Connections for Standard Serial I/O Mode (2)

### 22.4.2 Example of Circuit Application in the Standard Serial I/O Mode

Figures 22.22 and 22.23 show examples of Circuit Application in Standard Serial I/O Mode 1 and Mode 2, respectively. Refer to the user's manual of your serial programmer to handle pins controlled by the serial programmer.

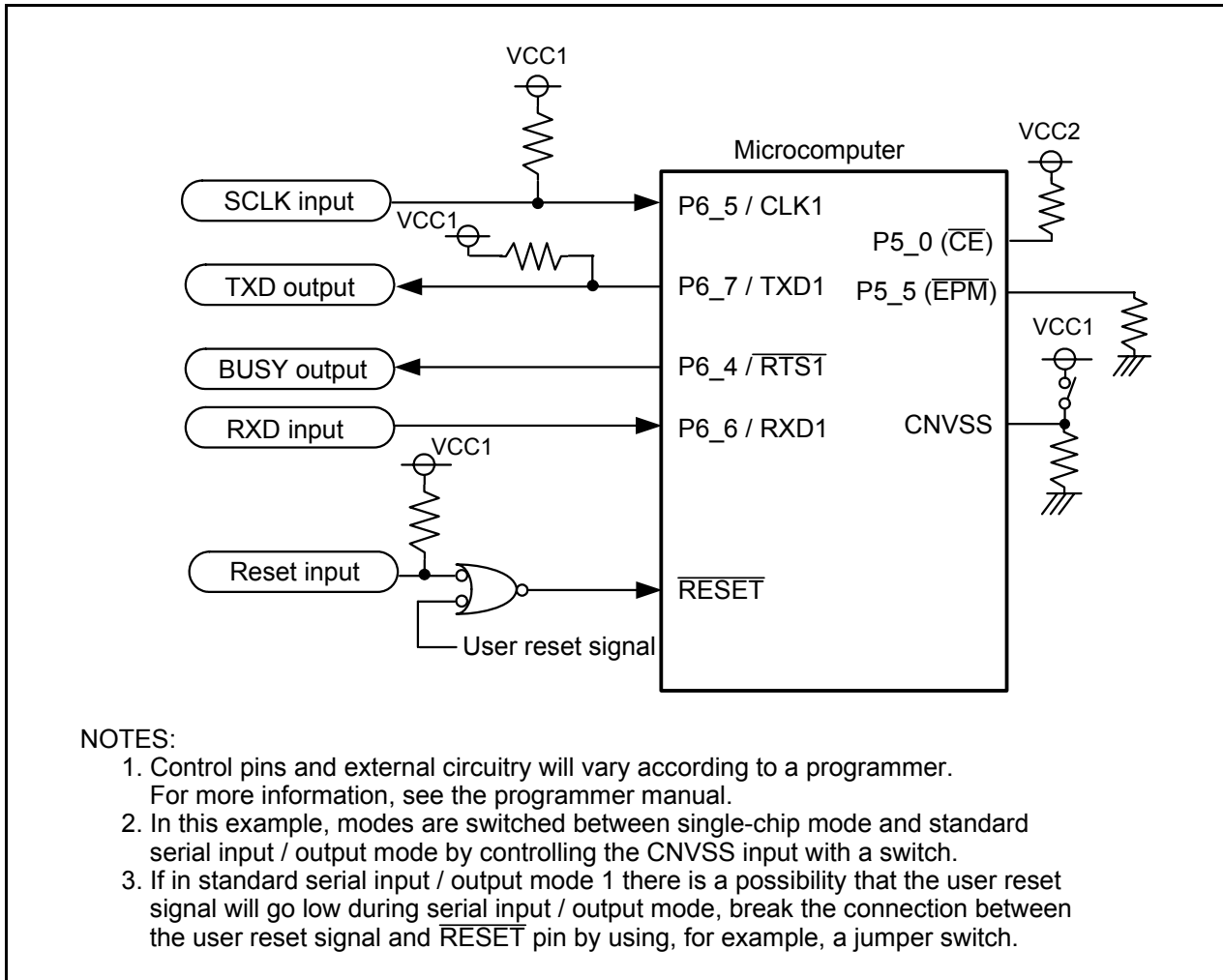


Figure 22.22 Circuit Application in Standard Serial I/O Mode 1

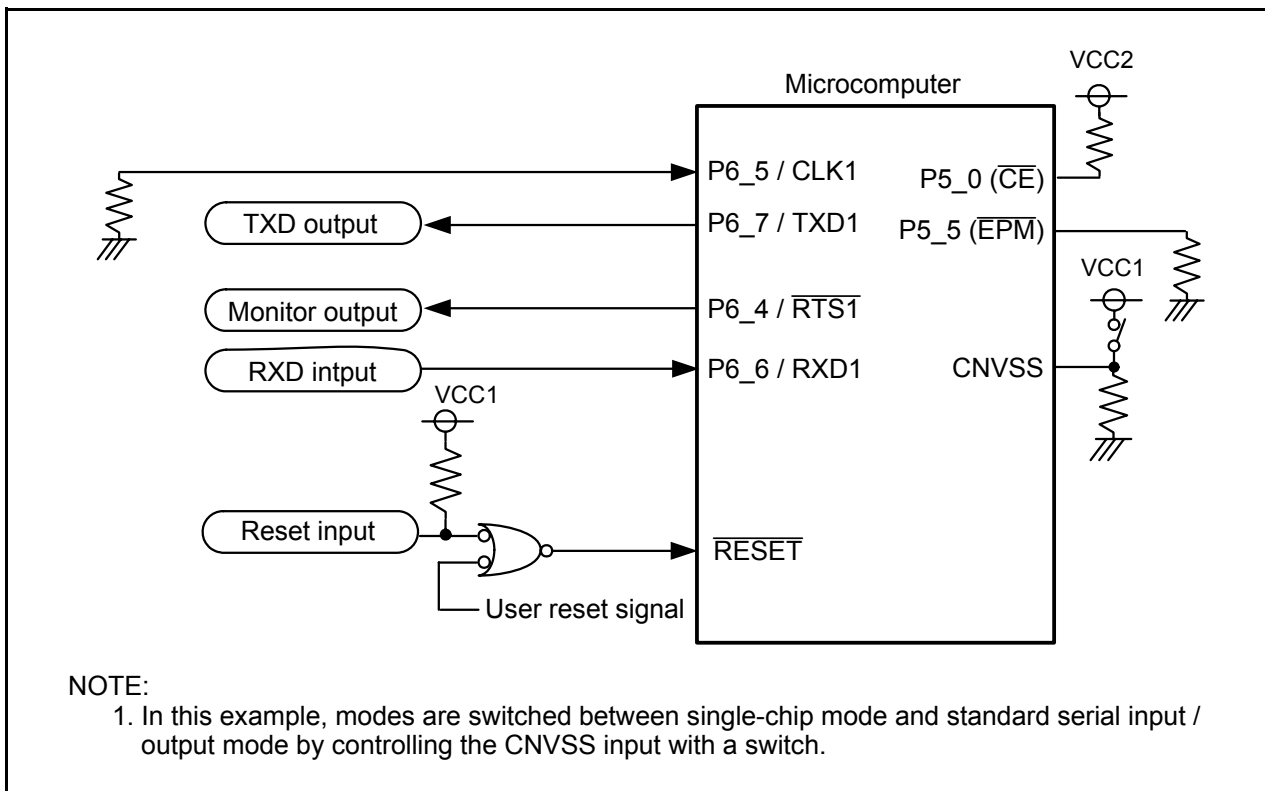


Figure 22.23 Circuit Application in Standard Serial I/O Mode 2



## 22.5 Parallel I/O Mode

In parallel I/O mode, the program ROM 1 and program ROM 2 can be rewritten by a parallel programmer supporting the M16C/64 Group. Contact your parallel programmer manufacturer for more information on the parallel programmer. Refer to the user's manual included with your parallel programmer for instructions.

### 22.5.1 ROM Code Protect Function

The ROM code protect function prevents the flash memory from being read and rewritten. (Refer to 22.2 “Functions to Prevent Flash Memory from Rewriting”.)

## 23. Electrical Characteristics

### 23.1 Electrical Characteristics

**Table 23.1 Absolute Maximum Ratings**

Symbol	Parameter		Condition	Rated Value	Unit
VCC1, VCC2	Supply Voltage		VCC1=VCC2 =AVCC	-0.3 to 6.5	V
AVCC	Analog Supply Voltage		VCC1=AVCC	-0.3 to 6.5	V
VI	Input Voltage	RESET, CNVSS, BYTE, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P14_0, P14_1, XIN		-0.3 to VCC1+0.3	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P7_0, P7_1, P8_5		-0.3 to VCC2+0.3	V
		P7_0, P7_1, P8_5		-0.3 to 6.5	V
VO	Output Voltage	P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, XOUT		-0.3 to VCC1+0.3	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P7_0, P7_1, P8_5		-0.3 to VCC2+0.3	V
		P7_0, P7_1, P8_5		-0.3 to 6.5	V
Pd	Power Dissipation		-40°C < Topr ≤ 85°C	300	mW
Topr	Operating Ambient Temperature	When the Microcomputer is Operating		-20 to 85 / -40 to 85	°C
		Flash Program Erase		0 to 60	
Tstg	Storage Temperature			-65 to 150	°C

**Table 23.2 Recommended Operating Conditions (1)**

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
VCC1, VCC2	Supply Voltage (VCC1 = VCC2)		2.7	5.0	5.5	V
AVCC	Analog Supply Voltage			VCC1		V
VSS	Supply Voltage			0		V
AVSS	Analog Supply Voltage			0		V
VIH	HIGH Input Voltage	P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7	0.8VCC2		VCC2	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (during single-chip mode)	0.8VCC2		VCC2	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (data input during memory expansion and microprocessor mode)	0.5VCC2		VCC2	V
		P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	0.8VCC1		VCC1	V
		P7_0, P7_1, P8_5	0.8VCC1		6.5	V
VIL	LOW Input Voltage	P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7	0		0.2VCC2	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (during single-chip mode)	0		0.2VCC2	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (data input during memory expansion and microprocessor mode)	0		0.16VCC2	V
		P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	0		0.2VCC	V
IOH(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, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7			-10.0	mA
IOH(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, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7			-5.0	mA
IOL(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, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7			10.0	mA
IOL(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, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7			5.0	mA
f(XIN)	Main Clock Input Oscillation Frequency	VCC1=2.7V to 5.5V	0		20	MHz
f(XCIN)	Sub-Clock Oscillation Frequency			32.768	50	kHz
f(OCO)	125kHz On-chip Oscillation Frequency			125		kHz
f(PLL)	PLL Clock Oscillation Frequency	VCC1=2.7V to 5.5V	10		25	MHz
f(BCLK)	CPU Operation Clock		0		25	MHz
tSU(PLL)	PLL Frequency Synthesizer Stabilization Wait Time	VCC1=5.5V			20	ms

NOTES:

1. Referenced to VCC1 = VCC2 = 2.7 to 5.5V at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified.
2. The Average Output Current is the mean value within 100ms.
3. The total IOL(peak) for ports P0, P1, P2, P8\_6, P8\_7, P9 and P10 must be 80mA max. The total IOL(peak) for ports P3, P4, P5, P6, P7 and P8\_0 to P8\_5 must be 80mA max. The total IOH(peak) for ports P0, P1, and P2 must be -40mA max. The total IOH(peak) for ports P3, P4 and P5 must be -40mA max. The total IOH(peak) for ports P6, P7\_2 to P7\_7 and P8\_0 to P8\_4 must be -40mA max.

**Table 23.3 A/D Conversion Characteristics (1)**

Symbol	Parameter		Measuring Condition	Standard			Unit
				Min.	Typ.	Max.	
-	Resolution		VREF=VCC1			10	Bits
INL	Integral Non-Linearity Error	10bit	VREF= AN0 to AN7 input, VCC1= AN0_0 to AN0_7 input, 5.0V AN2_0 to AN2_7 input, ANEX0, ANEX1 input			±3	LSB
			VREF= AN0 to AN7 input, VCC1= AN0_0 to AN0_7 input, 3.3V AN2_0 to AN2_7 input, ANEX0, ANEX1 input			±3	LSB
			VREF= AN0 to AN7 input, VCC1= AN0_0 to AN0_7 input, 3.0V AN2_0 to AN2_7 input, ANEX0, ANEX1 input			±3	LSB
-	Absolute Accuracy	10bit	VREF= AN0 to AN7 input, VCC1= AN0_0 to AN0_7 input, 5.0V AN2_0 to AN2_7 input, ANEX0, ANEX1 input			±3	LSB
			VREF= AN0 to AN7 input, VCC1= 3.3V AN0_0 to AN0_7 input, AN2_0 to AN2_7 input, ANEX0, ANEX1 input			±3	LSB
			VREF= AN0 to AN7 input, VCC1= 3.0V AN0_0 to AN0_7 input, AN2_0 to AN2_7 input, ANEX0, ANEX1 input			±3	LSB
-	Tolerance Level Impedance				3	kΩ	
DNL	Differential Non-Linearity Error					±1	LSB
-	Offset Error					±3	LSB
-	Gain Error					±3	LSB
RLADDER	Ladder Resistance		VREF=VCC1	10		40	kΩ
tCONV	10-bit Conversion Time		VREF=VCC1=5V, φAD=25MHz	1.60			μs
tSAMP	Sampling Time			0.60			μs
VREF	Reference Voltage				VCC1		V
VIA	Analog Input Voltage			0		VREF	V

NOTES:

1. Referenced to VCC1=AVCC=VREF=3.0 to 5.5V, VSS=AVSS=0V at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified.
2. Set φAD frequency as follows:  
 When VCC1 = 4.0 to 5.5 V, 2 MHz ≤ φAD ≤ 25 MHz  
 When VCC1 = 3.2 to 4.0 V, 2 MHz ≤ φAD ≤ 16 MHz  
 When VCC1 = 3.0 to 3.2 V, 2 MHz ≤ φAD ≤ 10 MHz
3. Use when VREF=VCC1.

**Table 23.4 D/A Conversion Characteristics (1)**

Symbol	Parameter	Measuring Condition	Standard			Unit
			Min.	Typ.	Max.	
-	Resolution				8	Bits
-	Absolute Accuracy				2.5	LSB
tSU	Setup Time				3	μs
RO	Output Resistance			6		kΩ
IVREF	Reference Power Supply Input Current	(NOTE 2)			1.5	mA

NOTES:

1. Referenced to VCC1=VREF=3.3 to 5.5V, VSS=AVSS=0V at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified.
2. This applies when using one D/A converter, with the D/A register for the unused D/A converter set to "00h". The resistor ladder of the A/D converter is not included. Also, when D/A register contents are not "00h", the IVREF will flow even if Vref id disconnected by the A/D control register.

**Table 23.5 Flash Memory Version Electrical Characteristics (1)**

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
-	Program and Erase Endurance (2)	Other than data flash	100			cycle
		Data flash	100			cycle
-	2 Word Program Time (VCC1=3.3V at Topr=25°C)	Other than data flash		150		μs
		Data flash		300		μs
-	Lock Bit Program Time (VCC1=3.3V at Topr=25°C)	Other than data flash		70		μs
		Data flash		140		μs
-	Block Erase Time (VCC1=3.3V at Topr=25°C)	4-Kbyte block		0.20		s
		16-Kbyte block		0.20		s
		64-Kbyte block		0.20		s
tPS	Flash Memory Circuit Stabilization Wait Time				50	μs
-	Data Hold Time (3)		10			year

NOTES:

1. Referenced to VCC1=2.7 to 5.5V at Topr = 0 to 60 °C unless otherwise specified.
2. Definition of program and erase endurance  
 The program and erase endurance refers to the number of per-block erasures.  
 If the program and erase endurance is n (n=100), each block can be erased n times.  
 For example, if a 4 Kbyte block is erased after writing two word data 1,024 times, each to a different address, this counts as one program and erase endurance. Data cannot be written to the same address more than once without erasing the block. (Rewrite prohibited)
3. Topr = -40 to 85 °C / -20 to 85 °C

**Table 23.6 Flash Memory Version Program / Erase Voltage and Read Operation Voltage Characteristics (at Topr = 0 to 60 °C)**

Flash Program, Erase Voltage	Flash Read Operation Voltage
VCC2 = 2.7 to 5.5 V	VCC1=2.7 to 5.5 V

**Table 23.7 Low Voltage Detection Circuit Electrical Characteristics**

Symbol	Parameter	Measuring Condition	Standard			Unit
			Min.	Typ.	Max.	
Vdet2	Low Voltage Detection Voltage (1)	VCC1=0.8V to 5.5V	3.3	3.8	4.4	V
Vdet0	Reset Level Detection Voltage (1)			1.9		V
Vdet2 -Vdet0	Electric potential difference of Low Voltage Detection and Reset Level Detection		0.3			V
Vdet0s	Low Voltage Reset Retention Voltage				0.8	V
Vdet0r	Low Voltage Reset Release Voltage (2)			2.0		V

NOTES:

1. Vdet2 > Vdet0.
2. Vdet0r > Vdet0 is not guaranteed.
3. The voltage detection circuit is designed to use when VCC1 is set to 5V.

**Table 23.8 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	VCC1=2.7V to 5.5V			5	ms
td(R-S)	STOP Release Time				150	μs
td(W-S)	Low Power Dissipation Mode Wait Mode Release Time				150	μs
td(S-R)	Brown-out Detection Reset (Hardware Reset 2) Release Wait Time	VCC1=Vdet3r to 5.5V		6 (1)	20	ms
td(E-A)	Low Voltage Detection Circuit Operation Start Time	VCC1=2.7V to 5.5V			20	μs

NOTE:

1. When VCC1 = 5V.

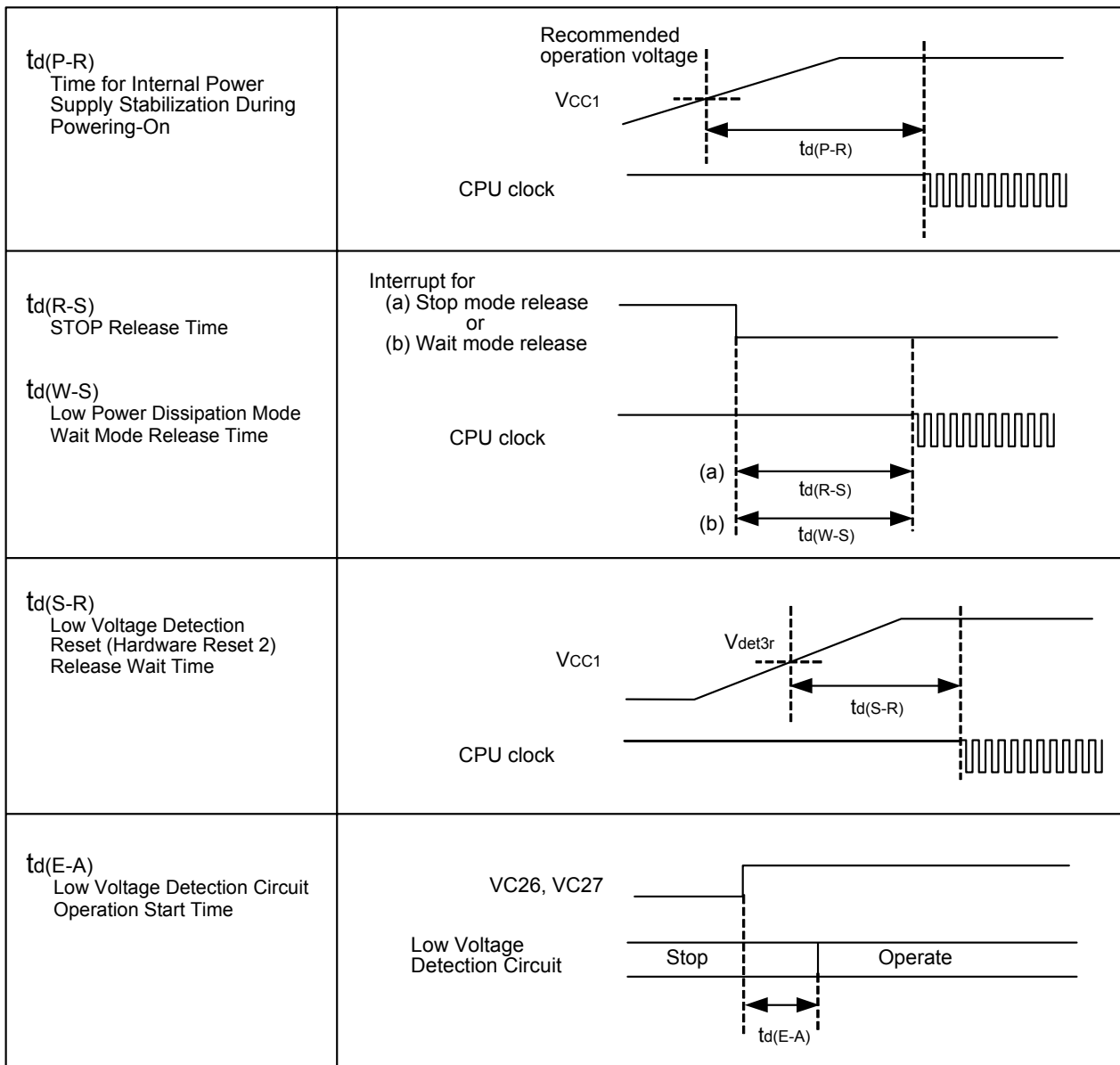


Figure 23.1 Power Supply Circuit Timing Diagram

$$VCC1=VCC2=5V$$

**Table 23.9 Electrical Characteristics (1) (1)**

Symbol	Parameter		Measuring Condition	Standard			Unit
				Min.	Typ.	Max.	
VOH	HIGH Output Voltage	P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7	IOH=-5mA	VCC1-2.0		VCC1	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7	IOH=-5mA	VCC2-2.0		VCC2	
VOH	HIGH Output Voltage	P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7	OH=-200μA	VCC1-0.3		VCC1	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7	IOH=-200μA	VCC2-0.3		VCC2	
VOH	HIGH Output Voltage	XOUT	HIGHPOWER	IOH=-1mA	VCC1-2.0		V
			LOWPOWER	IOH=-0.5mA	VCC1-2.0		
	HIGH Output Voltage	XCOUT	HIGHPOWER	With no load applied		2.9	V
			LOWPOWER	With no load applied		2.2	
VOL	LOW Output Voltage	P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7	IOL=5mA			2.0	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7	IOL=5mA			2.0	
VOL	LOW Output Voltage	P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7	IOL=200μA			0.45	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7	IOL=200μA			0.45	
VOL	LOW Output Voltage	XOUT	HIGHPOWER	IOL=1mA		2.0	V
			LOWPOWER	IOL=0.5mA		2.0	
	LOW Output Voltage	XCOUT	HIGHPOWER	With no load applied		0	V
			LOWPOWER	With no load applied		0	
VT+-VT-	Hysteresis	HOLD, RDY, TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT7, NMI, ADTRG, CTS0 to CTS2, CTS5 to CTS7, SCL0 to SCL2, SCL5 to SCL7, SDA0 to SDA2, SDA5 to SDA7, CLK0 to CLK7, TA0OUT to TA4OUT, K10 to K13, RXD0 to RXD2, RXD5 to RXD7, SIN3, SIN4		0.2		1.0	V
VT+-VT-	Hysteresis	RESET		0.2		2.5	V
IIH	HIGH Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	VI=5V			5.0	μA
IIL	LOW Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	VI=0V			-5.0	μA
RPULLUP	Pull-Up Resistance	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7	VI=0V	30	50	170	kΩ
RFXIN	Feedback Resistance	XIN			1.5		MΩ
RFXCIN	Feedback Resistance	XCIN			15		MΩ
VRAM	RAM Retention Voltage		At stop mode	2.0			V

NOTES:

1. Referenced to VCC1=VCC2=4.2 to 5.5V, VSS = 0V at Topr = -20 to 85°C / -40 to 85°C, f(BCLK)=25MHz unless otherwise specified.



**Table 23.10 Electrical Characteristics (2) (1)**

Symbol	Parameter		Measuring Condition		Standard			Unit
					Min.	Typ.	Max.	
ICC	Power Supply Current (VCC1=VCC2=4.0V to 5.5V)	In single-chip mode, the output pins are open and other pins are VSS	Flash Memory	f(BCLK)=25MHz, No division, PLL operation		20		mA
				No division, 125 kHz On-chip oscillation		450		μA
			Flash Memory Program	f(BCLK)=10MHz, VCC1=5.0V		20		mA
			Flash Memory Erase	f(BCLK)=10MHz, VCC1=5.0V		30		mA
			Flash Memory	f(BCLK)=32kHz Low power dissipation mode, RAM (3)		45		μA
				f(BCLK)=32kHz Low power dissipation mode, Flash Memory (3) FMR22=FMR23=1		160		μA
				125 kHz On-chip oscillation, Wait mode		12		μA
				f(BCLK)=32kHz Wait mode (2), Oscillation capability High		11.5		μA
				f(BCLK)=32kHz Wait mode (2), Oscillation capability Low		6.2		μA
					Stop mode Topr =25°C		3.0	
I <sub>det2</sub>	Low Voltage Detection Dissipation Current (4)				3.0		μA	
I <sub>det0</sub>	Reset Area Detection Dissipation Current (4)				6.0		μA	

NOTES:

1. Referenced to VCC1=VCC2=4.2 to 5.5V, VSS = 0V at Topr = -20 to 85°C / -40 to 85°C, f(BCLK)=25MHz unless otherwise specified.
2. With one timer operated using fC32.
3. This indicates the memory in which the program to be executed exists.
4. I<sub>det</sub> is dissipation current when the following bit is set to "1" (detection circuit enabled).  
 I<sub>det2</sub>: VC27 bit in the VCR2 register  
 I<sub>det0</sub>: VC25 bit in the VCR2 register

VCC1=VCC2=5V

**Timing Requirements**

(VCC1 = VCC2 = 5V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified)

**Table 23.11 External Clock Input (XIN input) (1)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc	External Clock Input Cycle Time	50		ns
tw(H)	External Clock Input HIGH Pulse Width	25		ns
tw(L)	External Clock Input LOW Pulse Width	25		ns
tr	External Clock Rise Time		15	ns
tf	External Clock Fall Time		15	ns

NOTE:

1. The condition is VCC1=VCC2=3.0 to 5.0V.

**Table 23.12 Memory Expansion Mode and Microprocessor Mode**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tac1(RD-DB)	Data Input Access Time (for setting with no wait)		(NOTE 1)	ns
tac2(RD-DB)	Data Input Access Time (for setting with wait)		(NOTE 2)	ns
tac3(RD-DB)	Data Input Access Time (when accessing multiplex bus area)		(NOTE 3)	ns
tsu(DB-RD)	Data Input Setup Time	40		ns
tsu(RDY-BCLK)	RDY Input Setup Time	30		ns
tsu(HOLD-BCLK)	HOLD Input Setup Time	40		ns
th(RD-DB)	Data Input Hold Time	0		ns
th(BCLK-RDY)	RDY Input Hold Time	0		ns
th(BCLK-HOLD)	HOLD Input Hold Time	0		ns

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 45[\text{ns}]$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 45[\text{ns}] \quad n \text{ is "2" for 1-wait setting, "3" for 2-wait setting and "4" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 45[\text{ns}] \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

$$VCC1=VCC2=5V$$

**Timing Requirements**

(VCC1 = VCC2 = 5V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified)

**Table 23.13 Timer A Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN Input Cycle Time	100		ns
tw(TAH)	TAiIN Input HIGH Pulse Width	40		ns
tw(TAL)	TAiIN Input LOW Pulse Width	40		ns

**Table 23.14 Timer A Input (Gating Input in Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN Input Cycle Time	400		ns
tw(TAH)	TAiIN Input HIGH Pulse Width	200		ns
tw(TAL)	TAiIN Input LOW Pulse Width	200		ns

**Table 23.15 Timer A Input (External Trigger Input in One-shot Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN Input Cycle Time	200		ns
tw(TAH)	TAiIN Input HIGH Pulse Width	100		ns
tw(TAL)	TAiIN Input LOW Pulse Width	100		ns

**Table 23.16 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(TAH)	TAiIN Input HIGH Pulse Width	100		ns
tw(TAL)	TAiIN Input LOW Pulse Width	100		ns

**Table 23.17 Timer A Input (Counter Increment/Decrement Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(UP)	TAiOUT Input Cycle Time	2000		ns
tw(UPH)	TAiOUT Input HIGH Pulse Width	1000		ns
tw(UPL)	TAiOUT Input LOW Pulse Width	1000		ns
tsu(UP-TIN)	TAiOUT Input Setup Time	400		ns
th(TIN-UP)	TAiOUT Input Hold Time	400		ns

**Table 23.18 Timer A Input (Two-phase Pulse Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN Input Cycle Time	800		ns
tsu(TAIN-TAOUT)	TAiOUT Input Setup Time	200		ns
tsu(TAOUT-TAIN)	TAiIN Input Setup Time	200		ns

$$VCC1=VCC2=5V$$

**Timing Requirements**

(VCC1 = VCC2 = 5V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified)

**Table 23.19 Timer B Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIn Input Cycle Time (counted on one edge)	100		ns
tw(TBH)	TBiIn Input HIGH Pulse Width (counted on one edge)	40		ns
tw(TBL)	TBiIn Input LOW Pulse Width (counted on one edge)	40		ns
tc(TB)	TBiIn Input Cycle Time (counted on both edges)	200		ns
tw(TBH)	TBiIn Input HIGH Pulse Width (counted on both edges)	80		ns
tw(TBL)	TBiIn Input LOW Pulse Width (counted on both edges)	80		ns

**Table 23.20 Timer B Input (Pulse Period Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIn Input Cycle Time	400		ns
tw(TBH)	TBiIn Input HIGH Pulse Width	200		ns
tw(TBL)	TBiIn Input LOW Pulse Width	200		ns

**Table 23.21 Timer B Input (Pulse Width Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIn Input Cycle Time	400		ns
tw(TBH)	TBiIn Input HIGH Pulse Width	200		ns
tw(TBL)	TBiIn Input LOW Pulse Width	200		ns

**Table 23.22 A/D Trigger Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(AD)	ADTRG Input Cycle Time	1000		ns
tw(ADL)	ADTRG input LOW Pulse Width	125		ns

**Table 23.23 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CK)	CLKi Input Cycle Time	200		ns
tw(CKH)	CLKi Input HIGH Pulse Width	100		ns
tw(CKL)	CLKi Input LOW Pulse Width	100		ns
td(C-Q)	TXDi Output Delay Time		80	ns
th(C-Q)	TXDi Hold Time	0		ns
tsu(D-C)	RXDi Input Setup Time	70		ns
th(C-D)	RXDi Input Hold Time	90		ns

**Table 23.24 External Interrupt INTi Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(INH)	INTi Input HIGH Pulse Width	250		ns
tw(INL)	INTi Input LOW Pulse Width	250		ns

VCC1=VCC2=5V

Switching Characteristics

(VCC1 = VCC2 = 5V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified)

Table 23.25 Memory Expansion and Microprocessor Modes (for setting with no wait)

Symbol	Parameter	Measuring Condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address Output Delay Time	See Figure 23.2		25	ns
th(BCLK-AD)	Address Output Hold Time (in relation to BCLK)		4		ns
th(RD-AD)	Address Output Hold Time (in relation to RD)		0		ns
th(WR-AD)	Address Output Hold Time (in relation to WR)		(NOTE 2)		ns
td(BCLK-CS)	Chip Select Output Delay Time			25	ns
th(BCLK-CS)	Chip Select Output Hold Time (in relation to BCLK)		4		ns
td(BCLK-ALE)	ALE Signal Output Delay Time			15	ns
th(BCLK-ALE)	ALE Signal Output Hold Time		-4		ns
td(BCLK-RD)	RD Signal Output Delay Time			25	ns
th(BCLK-RD)	RD Signal Output Hold Time		0		ns
td(BCLK-WR)	WR Signal Output Delay Time			25	ns
th(BCLK-WR)	WR Signal Output Hold Time		0		ns
td(BCLK-DB)	Data Output Delay Time (in relation to BCLK)			40	ns
th(BCLK-DB)	Data Output Hold Time (in relation to BCLK) (3)		4		ns
td(DB-WR)	Data Output Delay Time (in relation to WR)		(NOTE 1)		ns
th(WR-DB)	Data Output Hold Time (in relation to WR) (3)		(NOTE 2)		ns
td(BCLK-HLDA)	HLDA Output Delay Time			40	ns

NOTES:

- Calculated according to the BCLK frequency as follows:  

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40[\text{ns}] \quad f(\text{BCLK}) \text{ is } 12.5\text{MHz or less.}$$
- Calculated according to the BCLK frequency as follows:  

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10[\text{ns}]$$
- This standard value shows the timing when the output is off, and does not show hold time of data bus.  
 Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.  
 Hold time of data bus is expressed in  

$$t = -CR \times \ln(1 - \text{VOL} / \text{VCC2})$$
  
 by a circuit of the right figure.  
 For example, when VOL = 0.2VCC2, C = 30pF, R = 1kΩ, hold time of output "L" level is  

$$t = -30\text{pF} \times 1\text{k}\Omega \times \ln(1 - 0.2\text{VCC2} / \text{VCC2}) = 6.7\text{ns.}$$

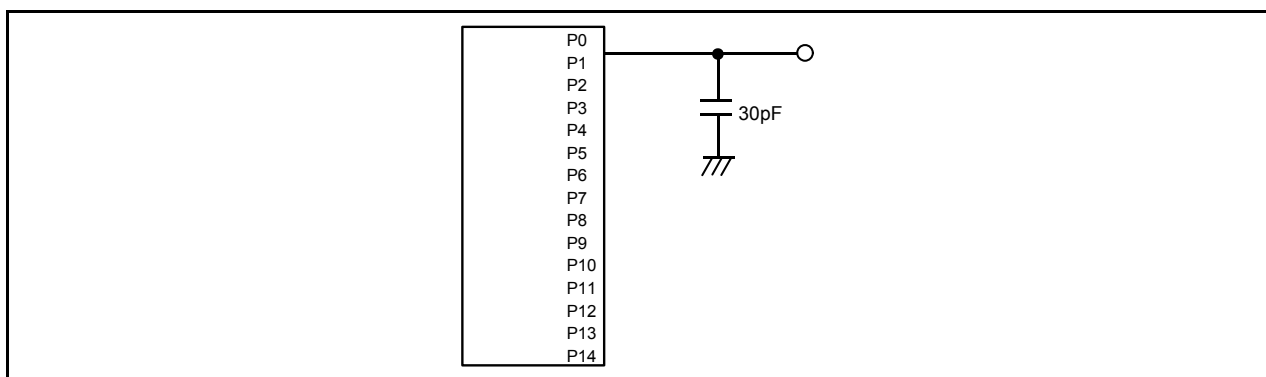
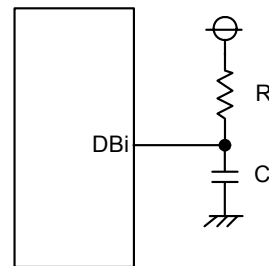


Figure 23.2 Ports P0 to P14 Measurement Circuit

VCC1=VCC2=5V

**Switching Characteristics**

(VCC1 = VCC2 = 5V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified)

**Table 23.26 Memory Expansion and Microprocessor Modes (for 1- to 3-wait setting and external area access)**

Symbol	Parameter	Measuring Condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address Output Delay Time	See Figure 23.2		25	ns
th(BCLK-AD)	Address Output Hold Time (in relation to BCLK)		4		ns
th(RD-AD)	Address Output Hold Time (in relation to RD)		0		ns
th(WR-AD)	Address Output Hold Time (in relation to WR)		(NOTE 2)		ns
td(BCLK-CS)	Chip Select Output Delay Time			25	ns
th(BCLK-CS)	Chip Select Output Hold Time (in relation to BCLK)		4		ns
td(BCLK-ALE)	ALE Signal Output Delay Time			15	ns
th(BCLK-ALE)	ALE Signal Output Hold Time		-4		ns
td(BCLK-RD)	RD Signal Output Delay Time			25	ns
th(BCLK-RD)	RD Signal Output Hold Time		0		ns
td(BCLK-WR)	WR Signal Output Delay Time			25	ns
th(BCLK-WR)	WR Signal Output Hold Time		0		ns
td(BCLK-DB)	Data Output Delay Time (in relation to BCLK)			40	ns
th(BCLK-DB)	Data Output Hold Time (in relation to BCLK) <sup>(3)</sup>		4		ns
td(DB-WR)	Data Output Delay Time (in relation to WR)		(NOTE 1)		ns
th(WR-DB)	Data Output Hold Time (in relation to WR) <sup>(3)</sup>		(NOTE 2)		ns
td(BCLK-HLDA)	HLDA Output Delay Time			40	ns

NOTES:

- Calculated according to the BCLK frequency as follows:  

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 40[\text{ns}]$$
 n is "1" for 1-wait setting, "2" for 2-wait setting and "3" for 3-wait setting. (BCLK) is 12.5MHz or less.

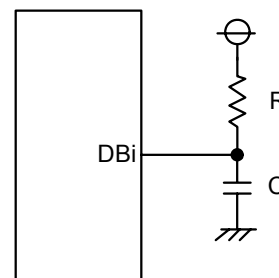
- Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10[\text{ns}]$$

- This standard value shows the timing when the output is off, and does not show hold time of data bus. Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value. Hold time of data bus is expressed in  

$$t = -CR \times \ln(1 - \text{VOL} / \text{VCC2})$$
 by a circuit of the right figure. For example, when VOL = 0.2VCC2, C = 30pF, R = 1kΩ, hold time of output "L" level is  

$$t = -30\text{pF} \times 1\text{k}\Omega \times \ln(1 - 0.2\text{VCC2} / \text{VCC2}) = 6.7\text{ns}.$$



VCC1=VCC2=5V

**Switching Characteristics**

(VCC1 = VCC2 = 5V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified)

**Table 23.27 Memory Expansion and Microprocessor Modes (for 2- to 3-wait setting, external area access and multiplex bus selection)**

Symbol	Parameter	Measuring Condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address Output Delay Time	See Figure 23.2		25	ns
th(BCLK-AD)	Address Output Hold Time (in relation to BCLK)		4		ns
th(RD-AD)	Address Output Hold Time (in relation to RD)		(NOTE 1)		ns
th(WR-AD)	Address Output Hold Time (in relation to WR)		(NOTE 1)		ns
td(BCLK-CS)	Chip Select Output Delay Time			25	ns
th(BCLK-CS)	Chip Select Output Hold Time (in relation to BCLK)		4		ns
th(RD-CS)	Chip Select Output Hold Time (in relation to RD)		(NOTE 1)		ns
th(WR-CS)	Chip Select Output Hold Time (in relation to WR)		(NOTE 1)		ns
td(BCLK-RD)	RD Signal Output Delay Time			25	ns
th(BCLK-RD)	RD Signal Output Hold Time		0		ns
td(BCLK-WR)	WR Signal Output Delay Time			25	ns
th(BCLK-WR)	WR Signal Output Hold Time		0		ns
td(BCLK-DB)	Data Output Delay Time (in relation to BCLK)			40	ns
th(BCLK-DB)	Data Output Hold Time (in relation to BCLK)		4		ns
td(DB-WR)	Data Output Delay Time (in relation to WR)		(NOTE 2)		ns
th(WR-DB)	Data Output Hold Time (in relation to WR)		(NOTE 1)		ns
td(BCLK-HLDA)	HLDA Output Delay Time			40	ns
td(BCLK-ALE)	ALE Signal Output Delay Time (in relation to BCLK)			15	ns
th(BCLK-ALE)	ALE Signal Output Hold Time (in relation to BCLK)		-4		ns
td(AD-ALE)	ALE Signal Output Delay Time (in relation to Address)		(NOTE 3)		ns
th(AD-ALE)	ALE Signal Output Hold Time (in relation to Address)	(NOTE 4)		ns	
td(AD-RD)	RD Signal Output Delay From the End of Address	0		ns	
td(AD-WR)	WR Signal Output Delay From the End of Address	0		ns	
tdz(RD-AD)	Address Output Floating Start Time		8	ns	

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10[\text{ns}]$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 40[\text{ns}] \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 25[\text{ns}]$$

4. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 15[\text{ns}]$$

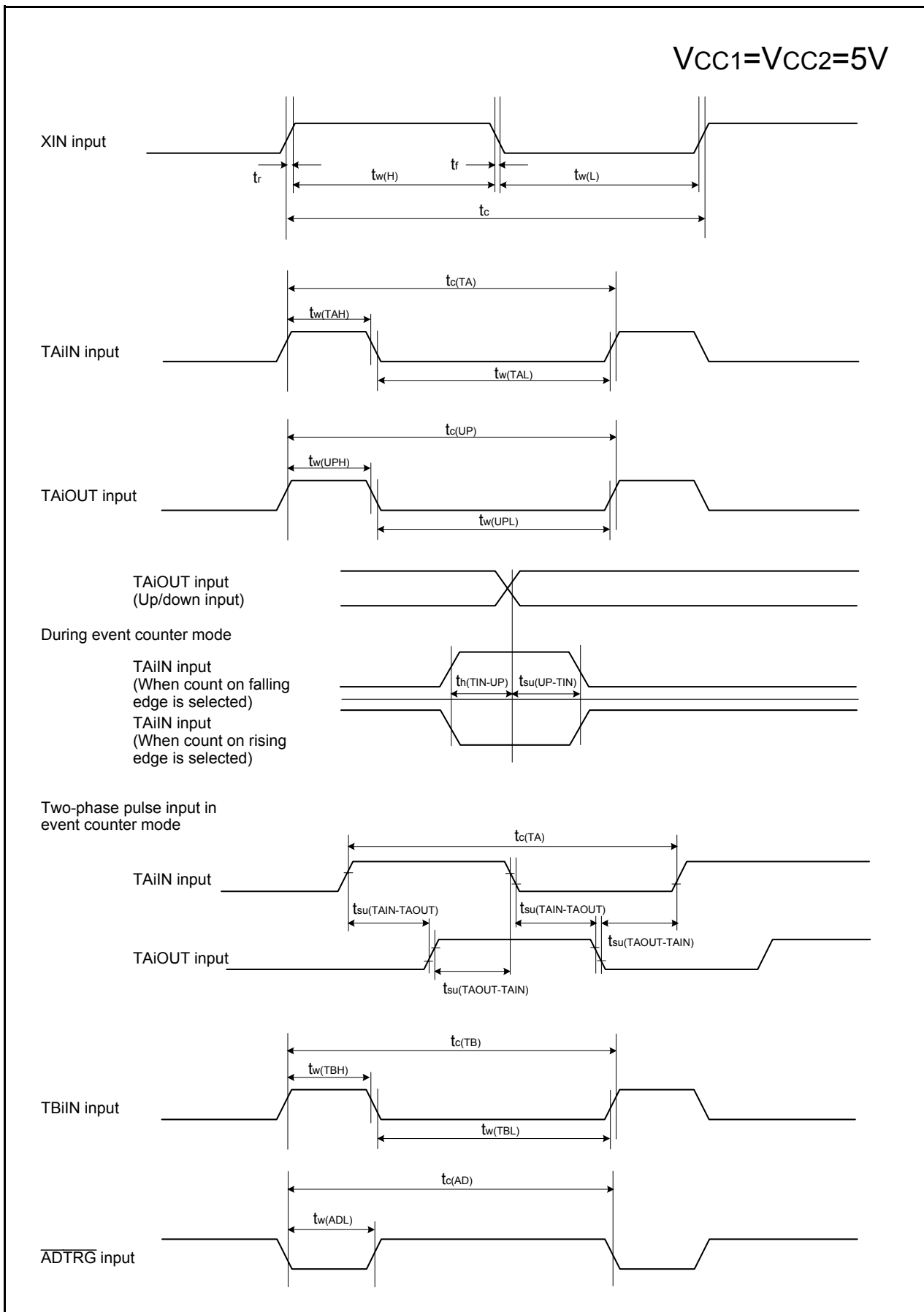


Figure 23.3 Timing Diagram (1)



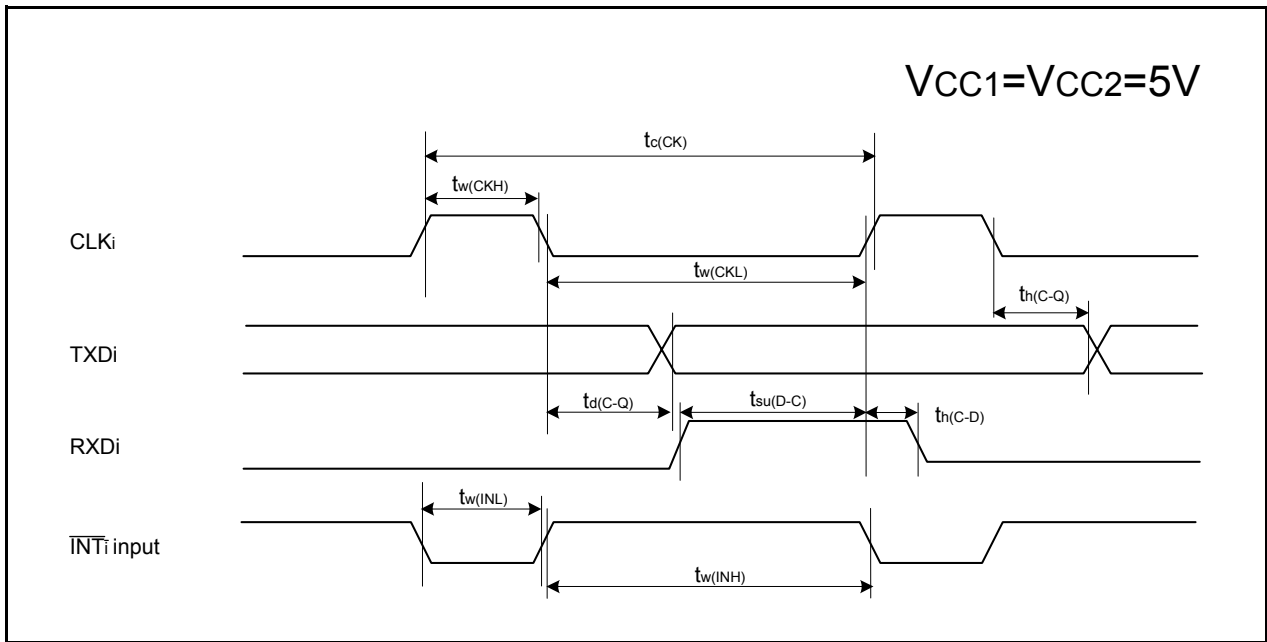


Figure 23.4 Timing Diagram (2)

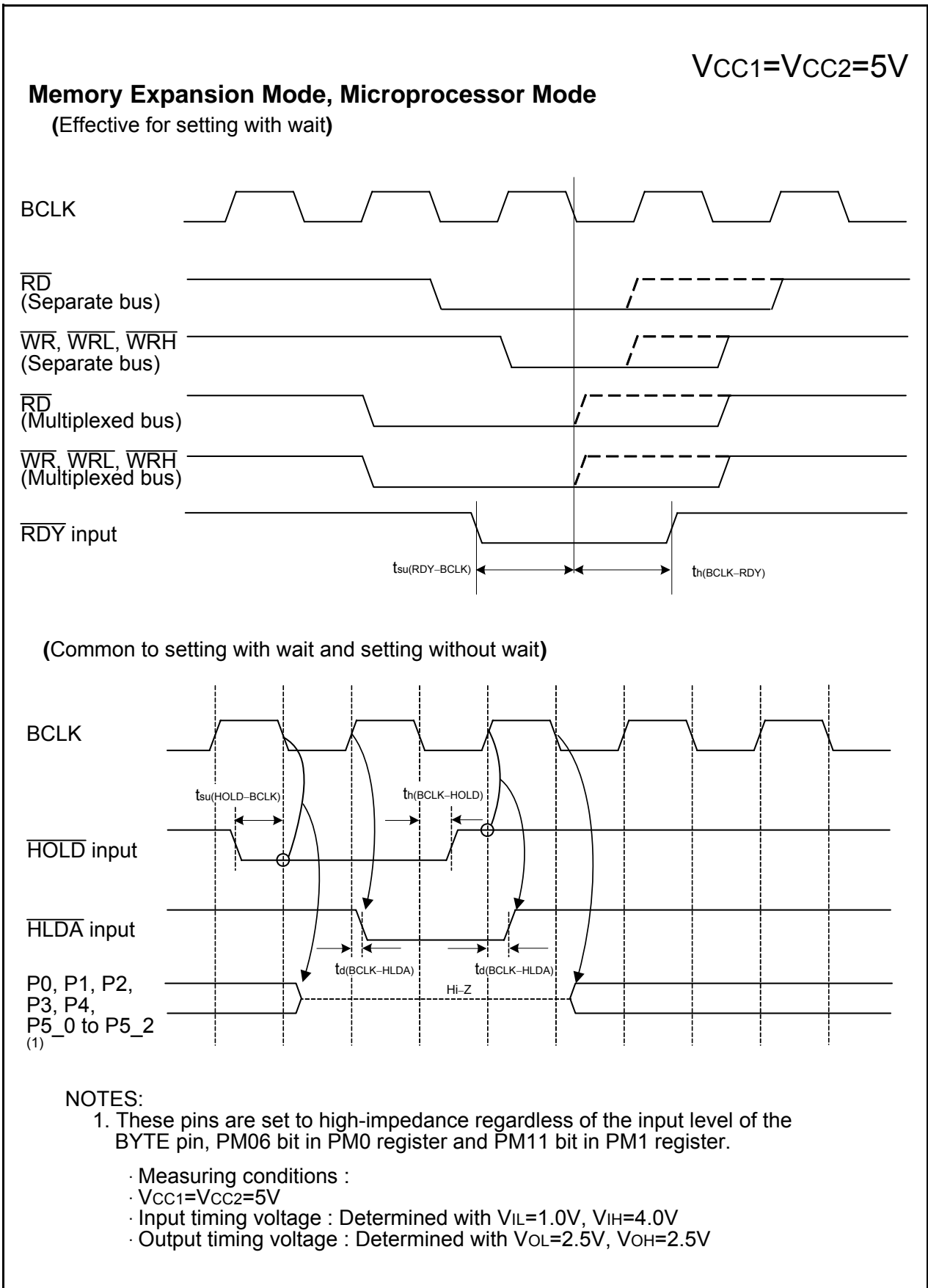


Figure 23.5 Timing Diagram (3)

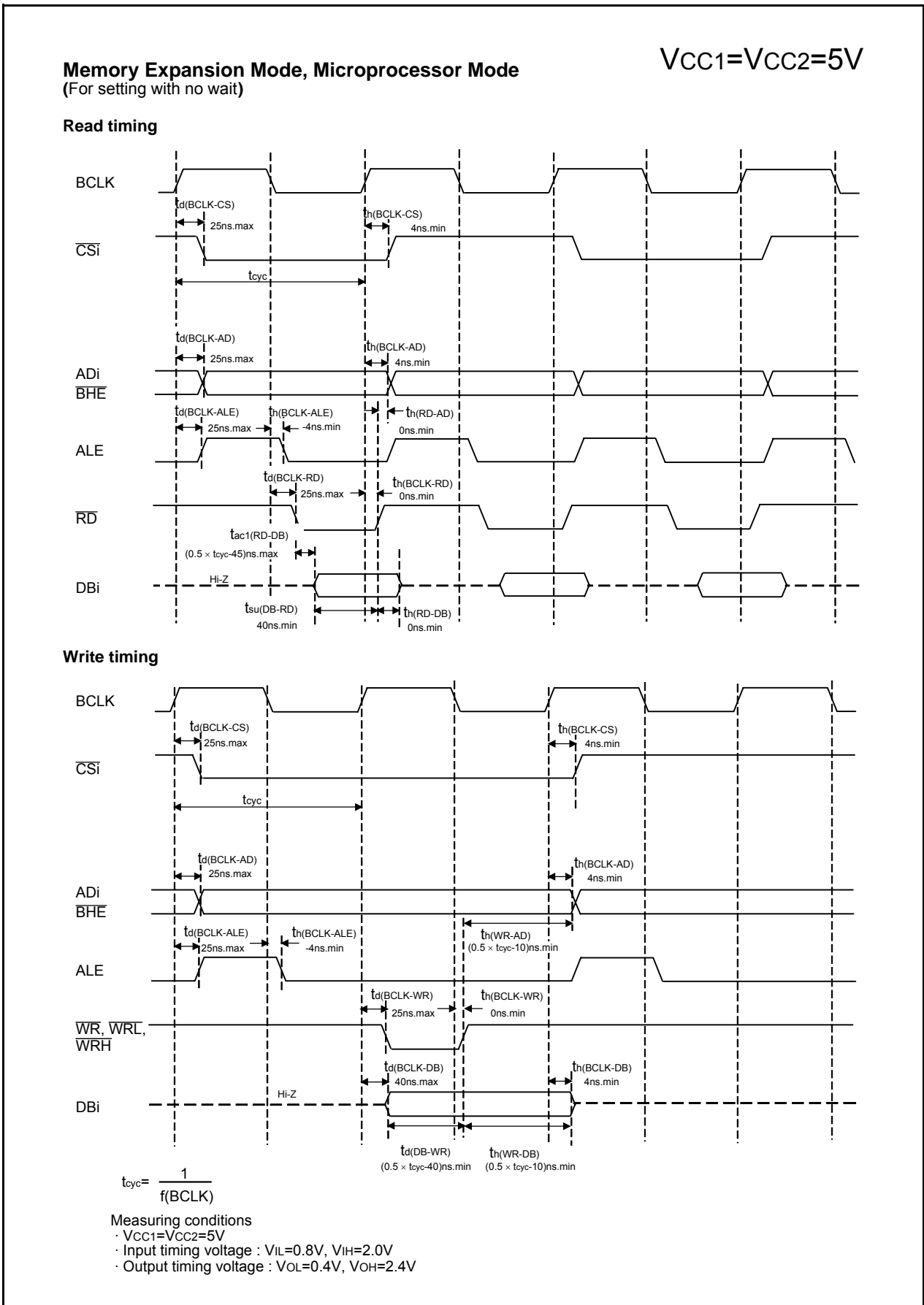


Figure 23.6 Timing Diagram (4)

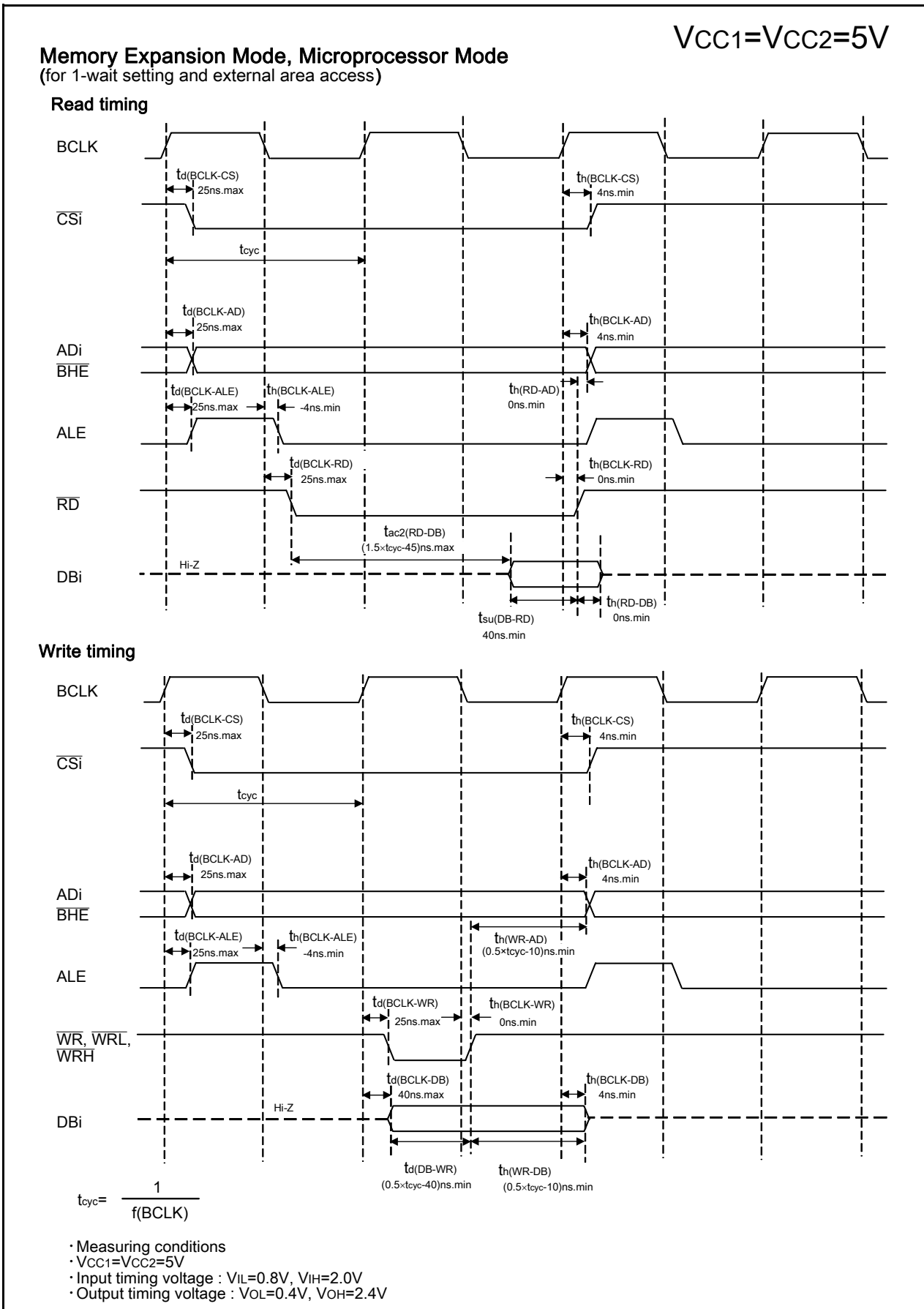


Figure 23.7 Timing Diagram (5)

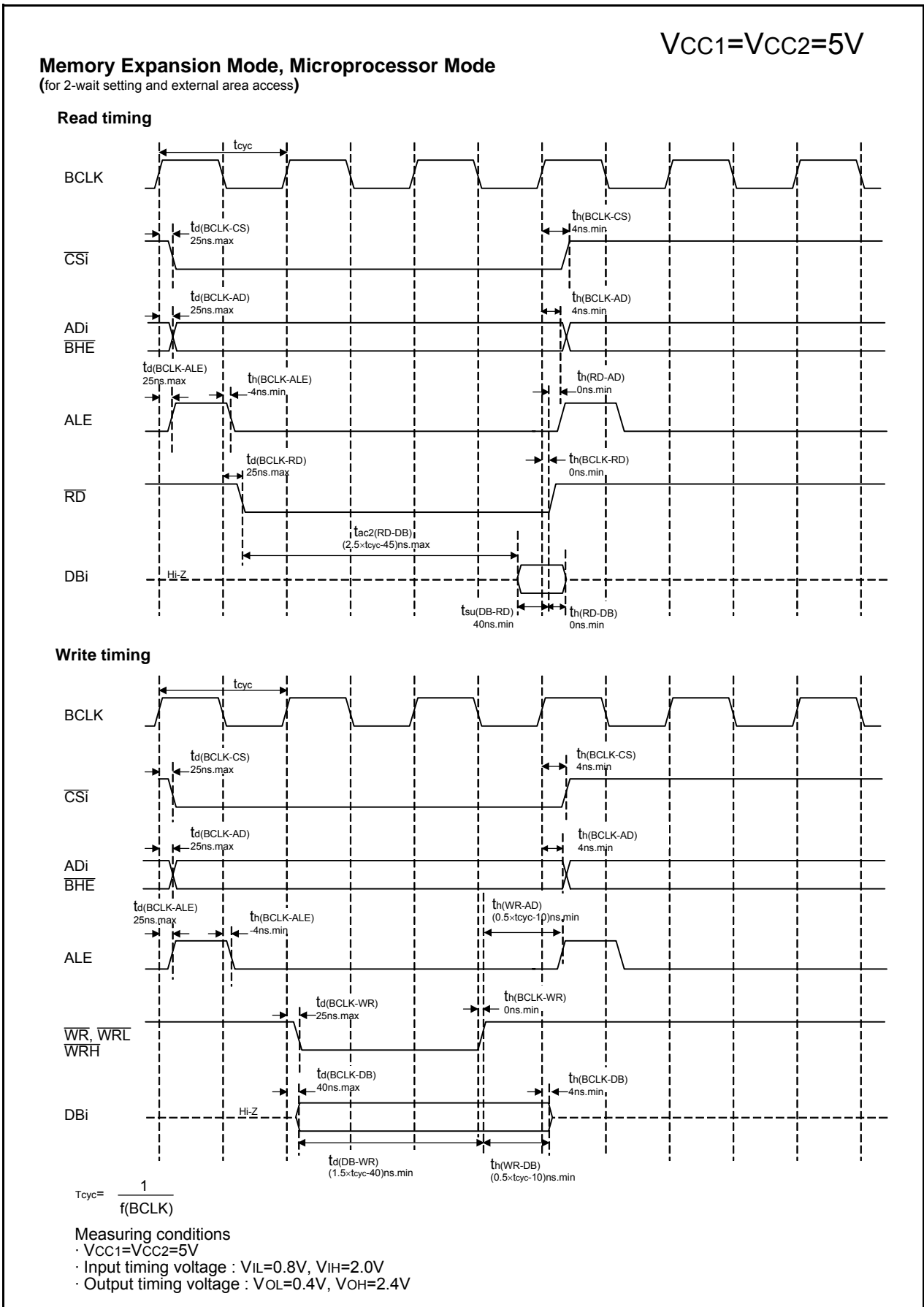


Figure 23.8 Timing Diagram (6)

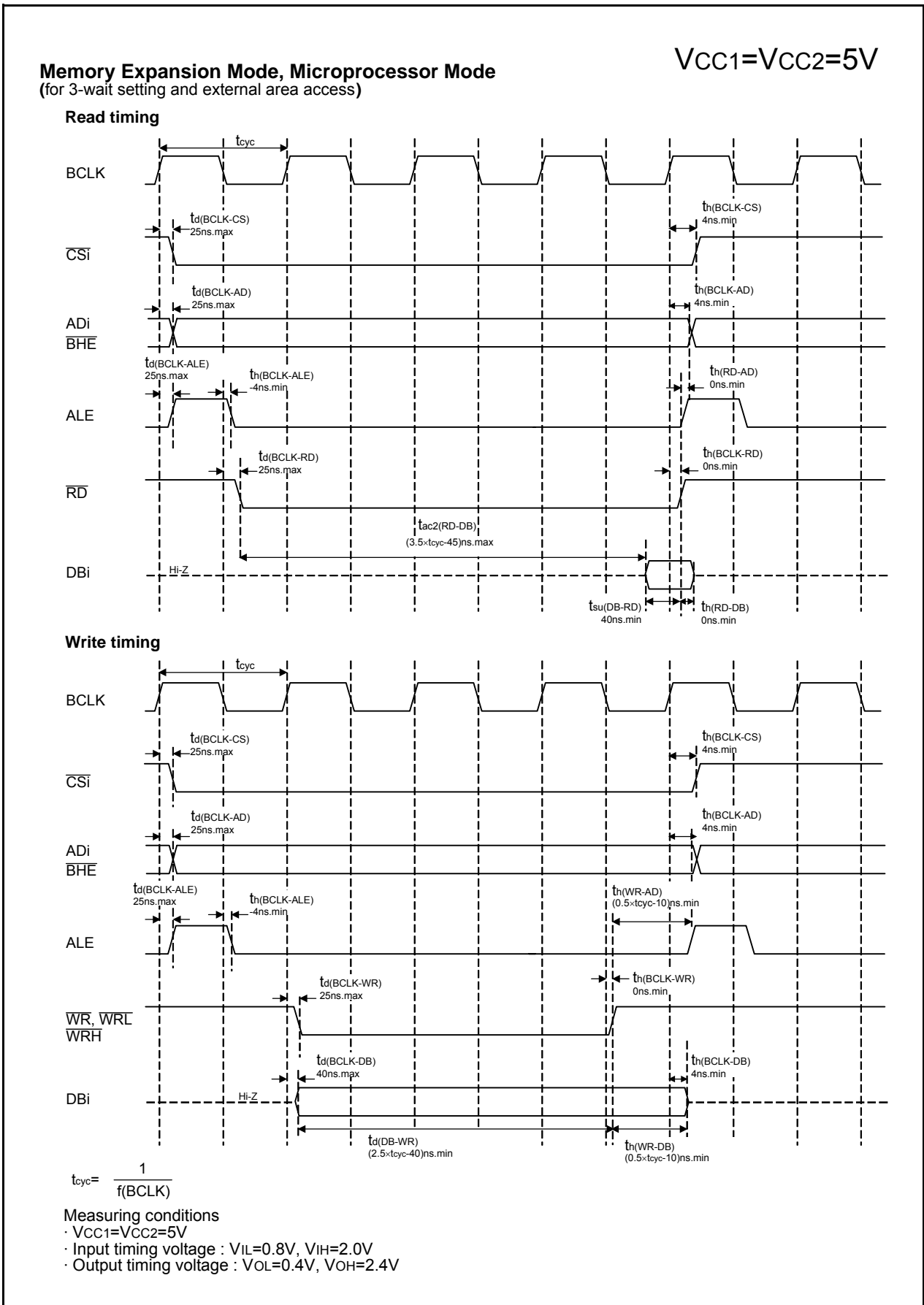


Figure 23.9 Timing Diagram (7)

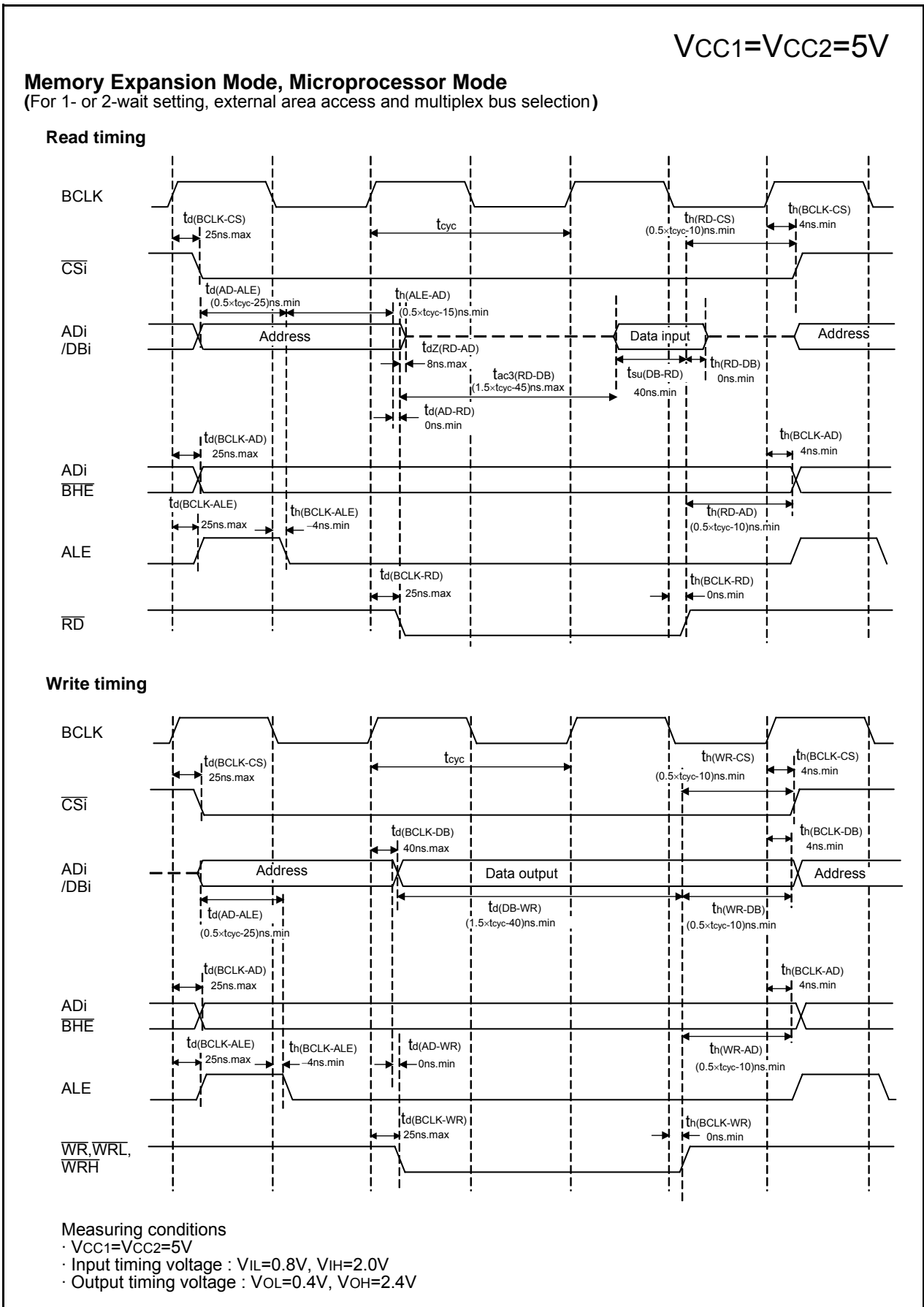


Figure 23.10 Timing Diagram (8)

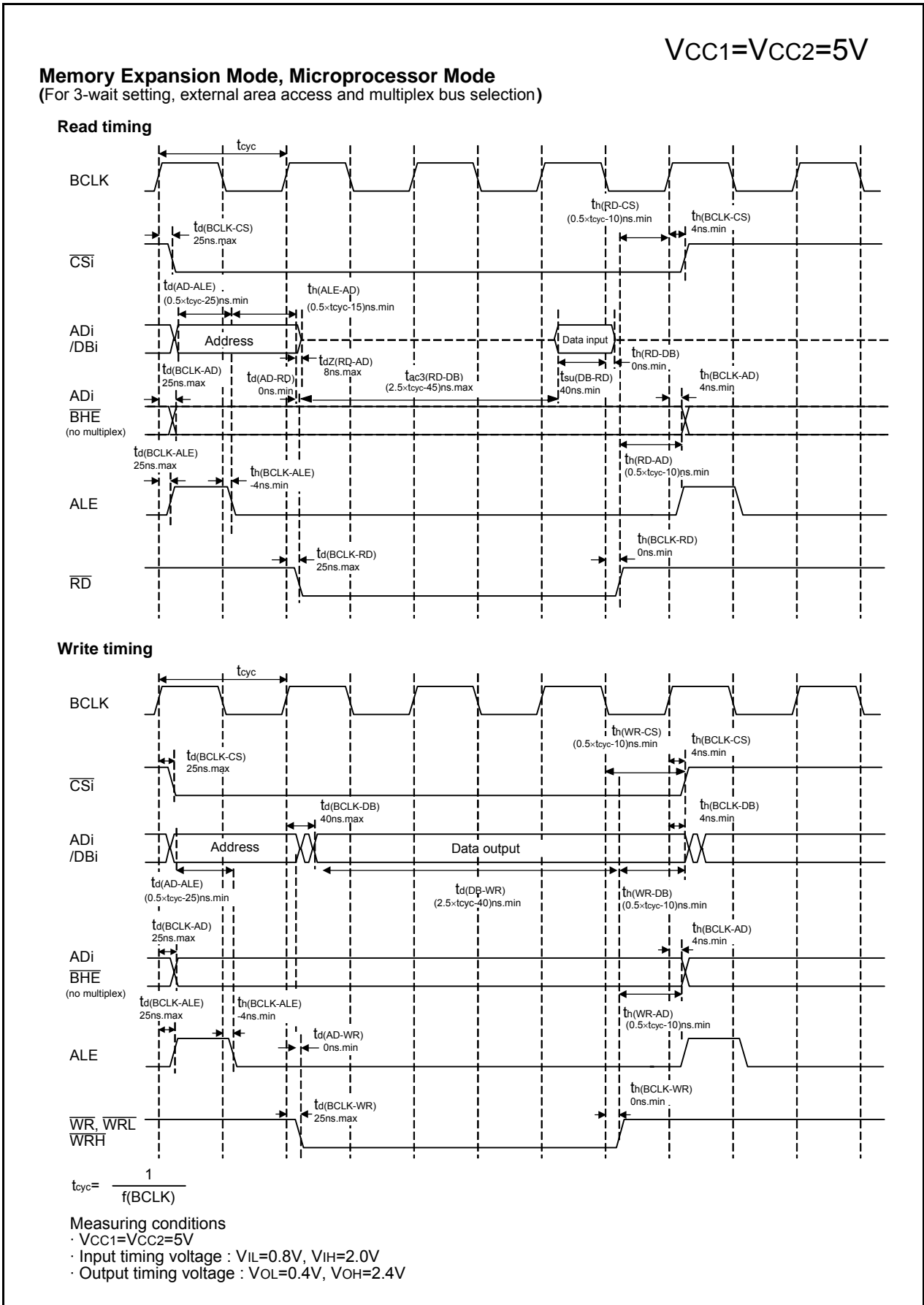


Figure 23.11 Timing Diagram (9)



$$VCC1=VCC2=3V$$

**Table 23.28 Electrical Characteristics (1) (1)**

Symbol	Parameter		Measuring Condition	Standard			Unit	
				Min.	Typ.	Max.		
VOH	HIGH Output Voltage	P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7	IOH=-1mA	VCC1-0.5		VCC1	V	
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7	IOH=-1mA (2)	VCC2-0.5		VCC2		
VOH	HIGH Output Voltage	XOUT	HIGHPOWER	IOH=-0.1mA	VCC1-0.5		VCC1	V
			LOWPOWER	IOH=-50μA	VCC1-0.5		VCC1	
	HIGH Output Voltage	XCOUT	HIGHPOWER	With no load applied		2.9		V
			LOWPOWER	With no load applied		2.2		
VOL	LOW Output Voltage	P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7	IOI=1mA			0.5	V	
			IOI=1mA (2)			0.5		
VOL	LOW Output Voltage	XOUT	HIGHPOWER	IOI=0.1mA		0.5	V	
			LOWPOWER	IOI=50μA		0.5		
	LOW Output Voltage	XCOUT	HIGHPOWER	With no load applied		0	V	
			LOWPOWER	With no load applied		0		
VT+-VT-	Hysteresis	HOLD, RDY, TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT7, NMI, ADTRG, CTS0 to CTS2, CTS5 to CTS7, SCL0 to SCL2, SCL5 to SCL7, SDA0 to SDA2, SDA5 to SDA7, CLK0 to CLK7, TA0OUT to TA4OUT, KI0 to KI3, RXD0 to RXD2, RXD5 to RXD7, SIN3, SIN4		0.2		0.8	V	
VT+-VT-	Hysteresis	RESET		0.2	(0.7)	1.8	V	
IIH	HIGH Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	VI=3V			4.0	μA	
IIL	LOW Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	VI=0V			-4.0	μA	
RPUL-LUP	Pull-Up Resistance	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7	VI=0V	50	100	500	kΩ	
RfXIN	Feedback Resistance	XIN			3.0		MΩ	
RfXCIN	Feedback Resistance	XCIN			25		MΩ	
VRAM	RAM Retention Voltage		At stop mode	2.0			V	

NOTES:

1. Referenced to VCC1 = VCC2 = 2.7 to 3.3V, VSS = 0V at Topr = -20 to 85°C / -40 to 85°C, f(XIN)=25MHz unless otherwise specified.
2. VCC1 for the port P6 to P10 and VCC2 for the port P0 to P5.

**Table 23.29 Electrical Characteristics (2) (1)**

Symbol	Parameter		Measuring Condition		Standard			Unit		
					Min.	Typ.	Max.			
ICC	Power Supply Current (VCC1=VCC2=2.7V to 3.6V)	In single-chip mode, the output pins are open and other pins are VSS	Flash Memory	f(BCLK)=25MHz, No division		20		mA		
				No division, 125 kHz On-chip oscilla- tion		450		μA		
			Flash Memory Program	f(BCLK)=10MHz, VCC1=3.0V		20		mA		
			Flash Memory Erase	f(BCLK)=10MHz, VCC1=3.0V		30		mA		
			Flash Memory	f(BCLK)=32kHz Low power dissipation mode, RAM (3)		40		μA		
					f(BCLK)=32kHz Low power dissipation mode, Flash Memory (3) FMR22=FMR23=1		160		μA	
						125 kHz On-chip oscillation, Wait mode		9		μA
						f(BCLK)=32kHz Wait mode (2), Oscillation capability High		9.5		μA
						f(BCLK)=32kHz Wait mode (2), Oscillation capability Low		5.7		μA
			Stop mode T <sub>opr</sub> =25°C		3		μA			
Idet2	Low Voltage Detection Dissipation Current (4)				3		μA			
Idet0	Reset Area Detection Dissipation Current (4)				6		μA			

NOTES:

1. Referenced to VCC1=VCC2=2.7 to 3.3V, VSS = 0V at T<sub>opr</sub> = -20 to 85°C / -40 to 85°C, f(BCLK)=25MHz unless otherwise specified.
2. With one timer operated using fC32.
3. This indicates the memory in which the program to be executed exists.
4. I<sub>det</sub> is dissipation current when the following bit is set to "1" (detection circuit enabled).  
 I<sub>det2</sub>: VC27 bit in the VCR2 register  
 I<sub>det0</sub>: VC25 bit in the VCR2 register

VCC1=VCC2=3V

**Timing Requirements**

(VCC1 = VCC2 = 3V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified)

**Table 23.30 External Clock Input (XIN input)<sup>(1)</sup>**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc	External Clock Input Cycle Time	50		ns
tw(H)	External Clock Input HIGH Pulse Width	20		ns
tw(L)	External Clock Input LOW Pulse Width	20		ns
tr	External Clock Rise Time		9	ns
tf	External Clock Fall Time		9	ns

NOTE:

1. The condition is VCC1=VCC2=2.7 to 3.0V.

**Table 23.31 Memory Expansion Mode and Microprocessor Mode**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tac1(RD-DB)	Data Input Access Time (for setting with no wait)		(NOTE 1)	ns
tac2(RD-DB)	Data Input Access Time (for setting with wait)		(NOTE 2)	ns
tac3(RD-DB)	Data Input Access Time (when accessing multiplex bus area)		(NOTE 3)	ns
tsu(DB-RD)	Data Input Setup Time	50		ns
tsu(RDY-BCLK)	$\overline{\text{RDY}}$ Input Setup Time	40		ns
tsu(HOLD-BCLK)	$\overline{\text{HOLD}}$ Input Setup Time	50		ns
th(RD-DB)	Data Input Hold Time	0		ns
th(BCLK-RDY)	$\overline{\text{RDY}}$ Input Hold Time	0		ns
th(BCLK-HOLD)	$\overline{\text{HOLD}}$ Input Hold Time	0		ns

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 60[\text{ns}]$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 60[\text{ns}] \quad n \text{ is "2" for 1-wait setting, "3" for 2-wait setting and "4" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 60[\text{ns}] \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

$$VCC1=VCC2=3V$$

**Timing Requirements**

(VCC1 = VCC2 = 3V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified)

**Table 23.32 Timer A Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN Input Cycle Time	150		ns
tw(TAH)	TAiIN Input HIGH Pulse Width	60		ns
tw(TAL)	TAiIN Input LOW Pulse Width	60		ns

**Table 23.33 Timer A Input (Gating Input in Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN Input Cycle Time	600		ns
tw(TAH)	TAiIN Input HIGH Pulse Width	300		ns
tw(TAL)	TAiIN Input LOW Pulse Width	300		ns

**Table 23.34 Timer A Input (External Trigger Input in One-shot Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN Input Cycle Time	300		ns
tw(TAH)	TAiIN Input HIGH Pulse Width	150		ns
tw(TAL)	TAiIN Input LOW Pulse Width	150		ns

**Table 23.35 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(TAH)	TAiIN Input HIGH Pulse Width	150		ns
tw(TAL)	TAiIN Input LOW Pulse Width	150		ns

**Table 23.36 Timer A Input (Counter Increment/Decrement Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(UP)	TAiOUT Input Cycle Time	3000		ns
tw(UPH)	TAiOUT Input HIGH Pulse Width	1500		ns
tw(UPL)	TAiOUT Input LOW Pulse Width	1500		ns
tsu(UP-TIN)	TAiOUT Input Setup Time	600		ns
th(TIN-UP)	TAiOUT Input Hold Time	600		ns

**Table 23.37 Timer A Input (Two-phase Pulse Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN Input Cycle Time	2		μs
tsu(TAIN-TAOUT)	TAiOUT Input Setup Time	500		ns
tsu(TAOUT-TAIN)	TAiIN Input Setup Time	500		ns

$$VCC1=VCC2=3V$$

**Timing Requirements**

(VCC1 = VCC2 = 3V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified)

**Table 23.38 Timer B Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIN Input Cycle Time (counted on one edge)	150		ns
tw(TBH)	TBiIN Input HIGH Pulse Width (counted on one edge)	60		ns
tw(TBL)	TBiIN Input LOW Pulse Width (counted on one edge)	60		ns
tc(TB)	TBiIN Input Cycle Time (counted on both edges)	300		ns
tw(TBH)	TBiIN Input HIGH Pulse Width (counted on both edges)	120		ns
tw(TBL)	TBiIN Input LOW Pulse Width (counted on both edges)	120		ns

**Table 23.39 Timer B Input (Pulse Period Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIN Input Cycle Time	600		ns
tw(TBH)	TBiIN Input HIGH Pulse Width	300		ns
tw(TBL)	TBiIN Input LOW Pulse Width	300		ns

**Table 23.40 Timer B Input (Pulse Width Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIN Input Cycle Time	600		ns
tw(TBH)	TBiIN Input HIGH Pulse Width	300		ns
tw(TBL)	TBiIN Input LOW Pulse Width	300		ns

**Table 23.41 A/D Trigger Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(AD)	ADTRG Input Cycle Time	1500		ns
tw(ADL)	ADTRG Input LOW Pulse Width	200		ns

**Table 23.42 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CK)	CLKi Input Cycle Time	300		ns
tw(CKH)	CLKi Input HIGH Pulse Width	150		ns
tw(CKL)	CLKi Input LOW Pulse Width	150		ns
td(C-Q)	TXDi Output Delay Time		160	ns
th(C-Q)	TXDi Hold Time	0		ns
tsu(D-C)	RXDi Input Setup Time	100		ns
th(C-D)	RXDi Input Hold Time	90		ns

**Table 23.43 External Interrupt  $\overline{INTi}$  Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(INH)	$\overline{INTi}$ Input HIGH Pulse Width	380		ns
tw(INL)	$\overline{INTi}$ Input LOW Pulse Width	380		ns

VCC1=VCC2=3V

Switching Characteristics

(VCC1 = VCC2 = 3V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified)

Table 23.44 Memory Expansion and Microprocessor Modes (for setting with no wait)

Symbol	Parameter	Measuring Condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address Output Delay Time	See Figure 23.12		30	ns
th(BCLK-AD)	Address Output Hold Time (in relation to BCLK)		4		ns
th(RD-AD)	Address Output Hold Time (in relation to RD)		0		ns
th(WR-AD)	Address Output Hold Time (in relation to WR)		(NOTE 2)		ns
td(BCLK-CS)	Chip Select Output Delay Time			30	ns
th(BCLK-CS)	Chip Select Output Hold Time (in relation to BCLK)		4		ns
td(BCLK-ALE)	ALE Signal Output Delay Time			25	ns
th(BCLK-ALE)	ALE Signal Output Hold Time		-4		ns
td(BCLK-RD)	RD Signal Output Delay Time			30	ns
th(BCLK-RD)	RD Signal Output Hold Time		0		ns
td(BCLK-WR)	WR Signal Output Delay Time			30	ns
th(BCLK-WR)	WR Signal Output Hold Time		0		ns
td(BCLK-DB)	Data Output Delay Time (in relation to BCLK)			40	ns
th(BCLK-DB)	Data Output Hold Time (in relation to BCLK) (3)		4		ns
td(DB-WR)	Data Output Delay Time (in relation to WR)		(NOTE 1)		ns
th(WR-DB)	Data Output Hold Time (in relation to WR) (3)		(NOTE 2)		ns
td(BCLK-HLDA)	HLDA Output Delay Time			40	ns

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40[\text{ns}] \quad f(\text{BCLK}) \text{ is } 12.5\text{MHz or less.}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10[\text{ns}]$$

This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

$$t = -CR \times \ln(1 - \text{VOL} / \text{VCC2})$$

by a circuit of the right figure.

For example, when VOL = 0.2VCC2, C = 30pF, R = 1kΩ,

hold time of output "L" level is

$$t = -30\text{pF} \times 1\text{k}\Omega \times \ln(1 - 0.2\text{VCC2} / \text{VCC2}) = 6.7\text{ns.}$$

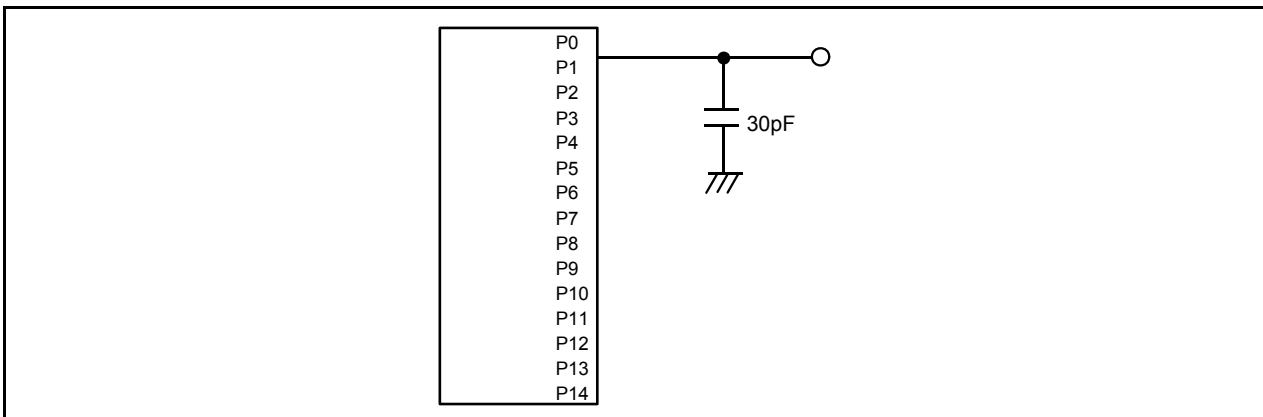
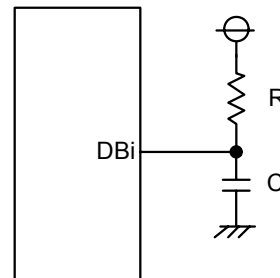


Figure 23.12 Ports P0 to P14 Measurement Circuit

VCC1=VCC2=3V

**Switching Characteristics**

(VCC1 = VCC2 = 5V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified)

**Table 23.45 Memory Expansion and Microprocessor Modes (for 1- to 3-wait setting and external area access)**

Symbol	Parameter	Measuring Condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address Output Delay Time	See Figure 23.12		30	ns
th(BCLK-AD)	Address Output Hold Time (in relation to BCLK)		4		ns
th(RD-AD)	Address Output Hold Time (in relation to RD)		0		ns
th(WR-AD)	Address Output Hold Time (in relation to WR)		(NOTE 2)		ns
td(BCLK-CS)	Chip Select Output Delay Time			30	ns
th(BCLK-CS)	Chip Select Output Hold Time (in relation to BCLK)		4		ns
td(BCLK-ALE)	ALE Signal Output Delay Time			25	ns
th(BCLK-ALE)	ALE Signal Output Hold Time		-4		ns
td(BCLK-RD)	RD Signal Output Delay Time			30	ns
th(BCLK-RD)	RD Signal Output Hold Time		0		ns
td(BCLK-WR)	WR Signal Output Delay Time			30	ns
th(BCLK-WR)	WR Signal Output Hold Time		0		ns
td(BCLK-DB)	Data Output Delay Time (in relation to BCLK)			40	ns
th(BCLK-DB)	Data Output Hold Time (in relation to BCLK) (3)		4		ns
td(DB-WR)	Data Output Delay Time (in relation to WR)		(NOTE 1)		ns
th(WR-DB)	Data Output Hold Time (in relation to WR)(3)		(NOTE 2)		ns
td(BCLK-HLDA)	HLDA Output Delay Time			40	ns

NOTES:

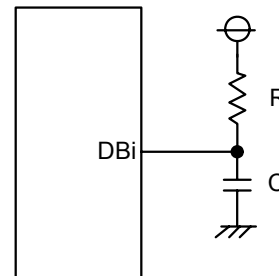
1. Calculated according to the BCLK frequency as follows:  

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 40[\text{ns}]$$
 n is "1" for 1-wait setting, "2" for 2-wait setting and "3" for 3-wait setting. (BCLK) is 12.5MHz or less.

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10[\text{ns}]$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus. Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value. Hold time of data bus is expressed in  $t = -CR \times \ln(1 - VOL / VCC2)$  by a circuit of the right figure. For example, when VOL = 0.2VCC2, C = 30pF, R = 1kΩ, hold time of output "L" level is  $t = -30\text{pF} \times 1\text{k}\Omega \times \ln(1 - 0.2VCC2 / VCC2) = 6.7\text{ns}$ .



VCC1=VCC2=3V

**Switching Characteristics**

(VCC1 = VCC2 = 5V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C unless otherwise specified)

**Table 23.46 Memory Expansion and Microprocessor Modes (for 2- to 3-wait setting, external area access and multiplex bus selection)**

Symbol	Parameter	Measuring Condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address Output Delay Time	See Figure 23.12		50	ns
th(BCLK-AD)	Address Output Hold Time (in relation to BCLK)		4		ns
th(RD-AD)	Address Output Hold Time (in relation to RD)		(NOTE 1)		ns
th(WR-AD)	Address Output Hold Time (in relation to WR)		(NOTE 1)		ns
td(BCLK-CS)	Chip Select Output Delay Time			50	ns
th(BCLK-CS)	Chip Select Output Hold Time (in relation to BCLK)		4		ns
th(RD-CS)	Chip Select Output Hold Time (in relation to RD)		(NOTE 1)		ns
th(WR-CS)	Chip Select Output Hold Time (in relation to WR)		(NOTE 1)		ns
td(BCLK-RD)	RD Signal Output Delay Time			40	ns
th(BCLK-RD)	RD Signal Output Hold Time		0		ns
td(BCLK-WR)	WR Signal Output Delay Time			40	ns
th(BCLK-WR)	WR Signal Output Hold Time		0		ns
td(BCLK-DB)	Data Output Delay Time (in relation to BCLK)			50	ns
th(BCLK-DB)	Data Output Hold Time (in relation to BCLK)		4		ns
td(DB-WR)	Data Output Delay Time (in relation to WR)		(NOTE 2)		ns
th(WR-DB)	Data Output Hold Time (in relation to WR)		(NOTE 1)		ns
td(BCLK-HLDA)	HLDA Output Delay Time			40	ns
td(BCLK-ALE)	ALE Signal Output Delay Time (in relation to BCLK)			25	ns
th(BCLK-ALE)	ALE Signal Output Hold Time (in relation to BCLK)		-4		ns
td(AD-ALE)	ALE Signal Output Delay Time (in relation to Address)		(NOTE 3)		ns
th(AD-ALE)	ALE Signal Output Hold Time (in relation to Address)	(NOTE 4)		ns	
td(AD-RD)	RD Signal Output Delay From the End of Address	0		ns	
td(AD-WR)	WR Signal Output Delay From the End of Address	0		ns	
tdz(RD-AD)	Address Output Floating Start Time		8	ns	

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10[\text{ns}]$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 50[\text{ns}] \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40[\text{ns}]$$

4. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 15[\text{ns}]$$



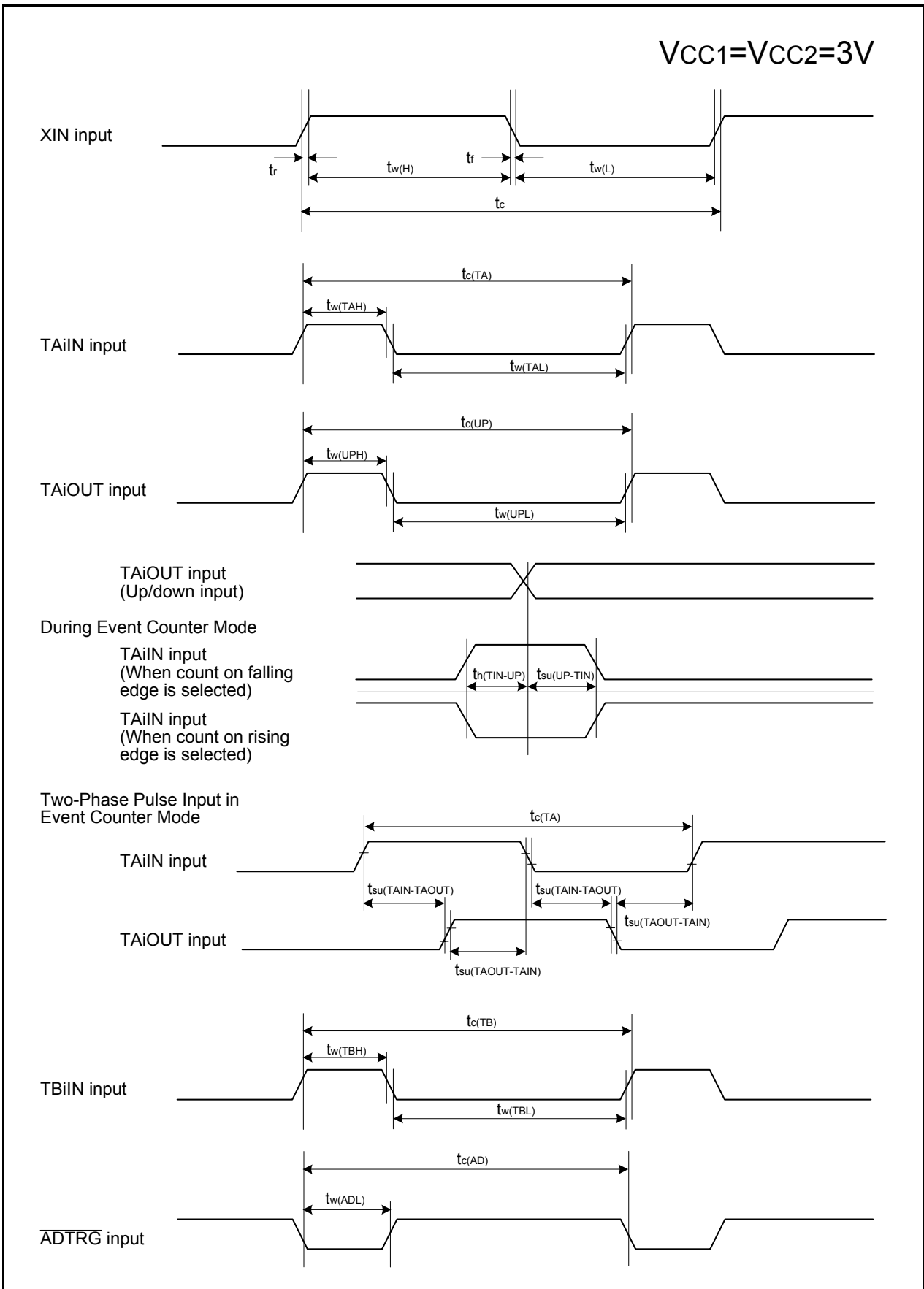


Figure 23.13 Timing Diagram (1)

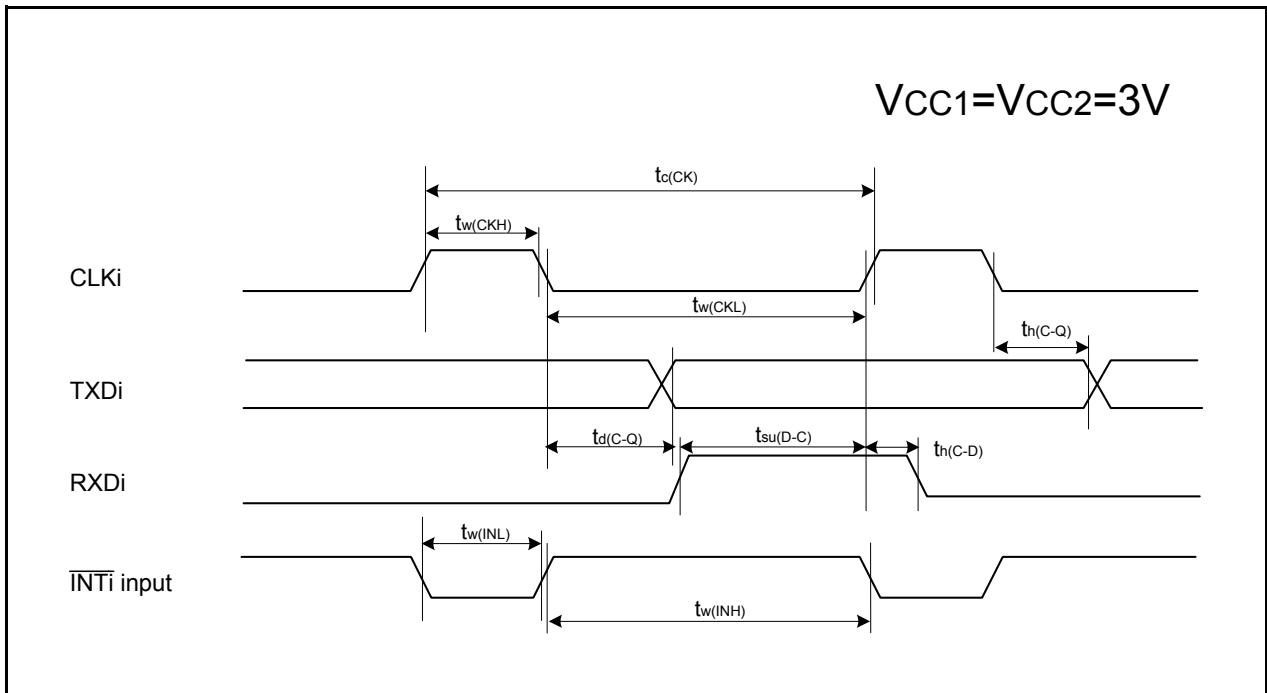


Figure 23.14 Timing Diagram (2)

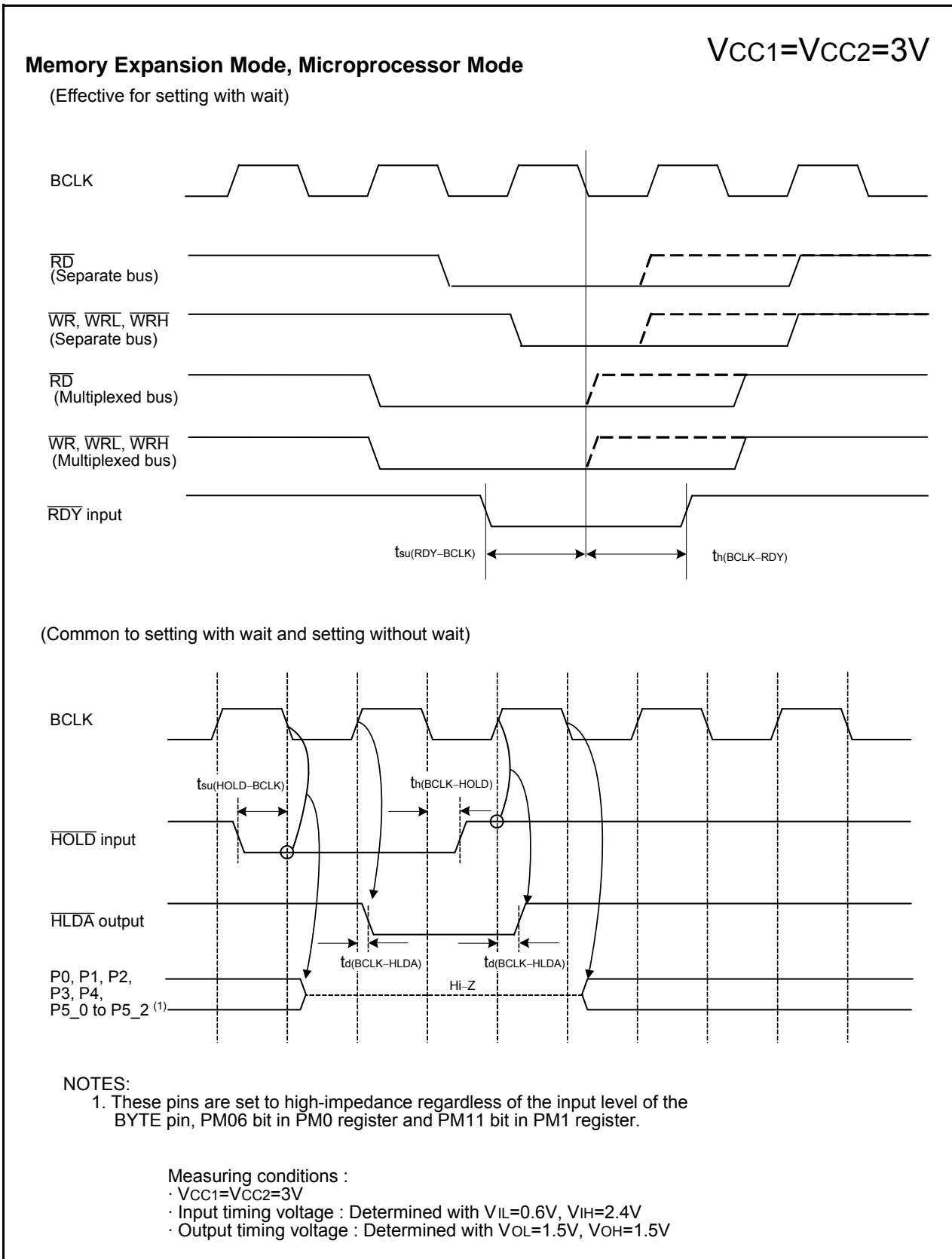


Figure 23.15 Timing Diagram (3)

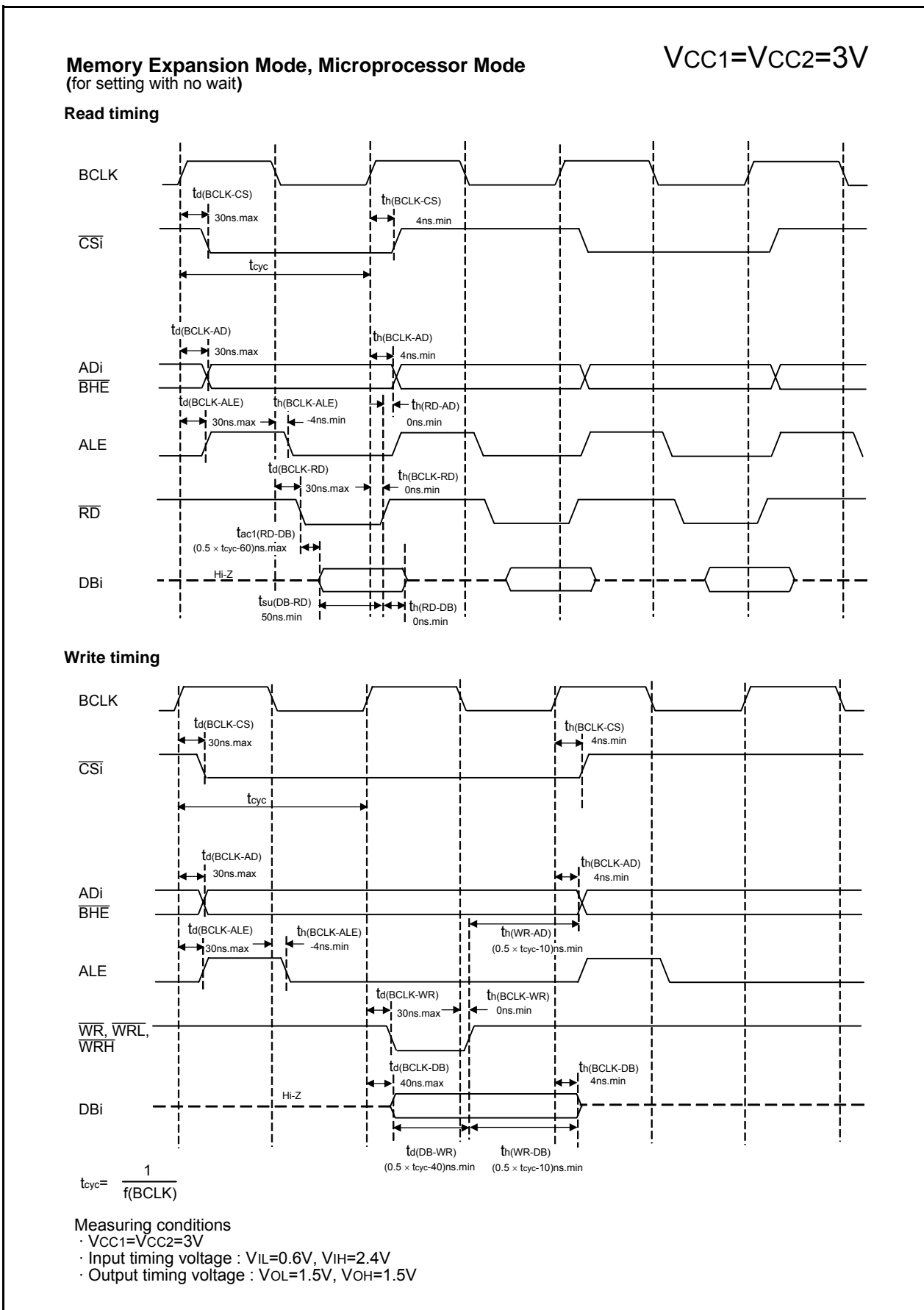


Figure 23.16 Timing Diagram (4)

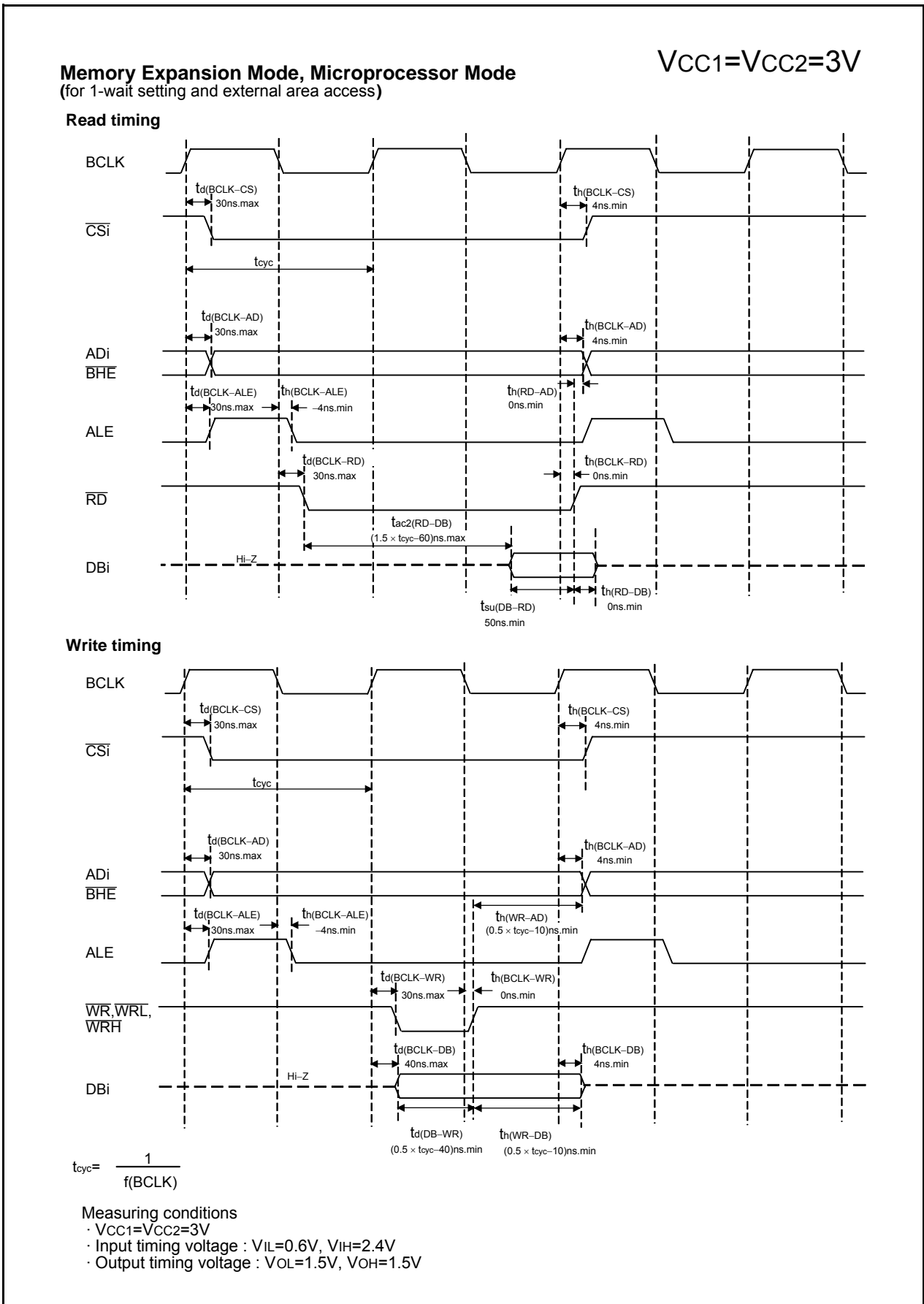


Figure 23.17 Timing Diagram (5)

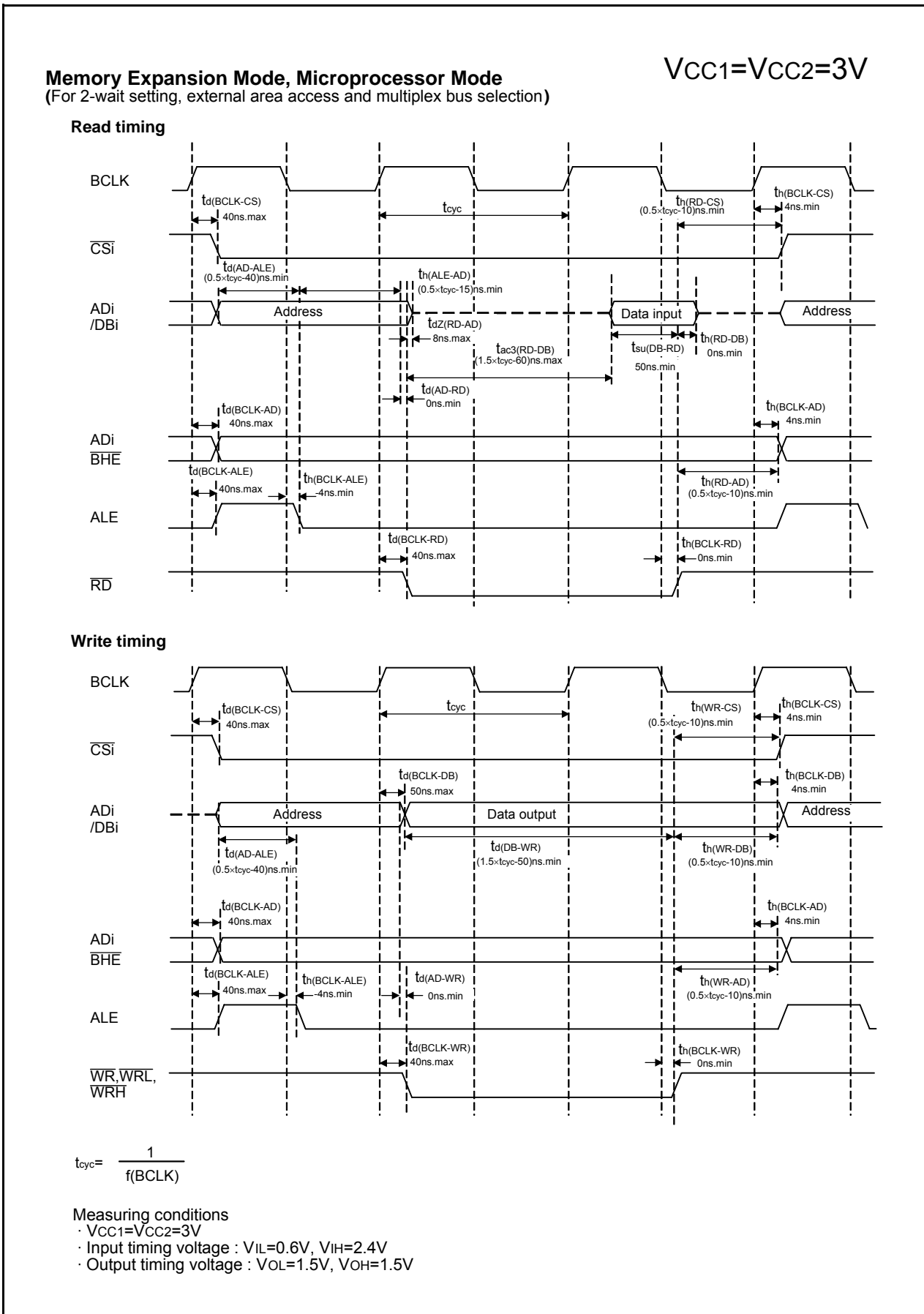


Figure 23.18 Timing Diagram (6)

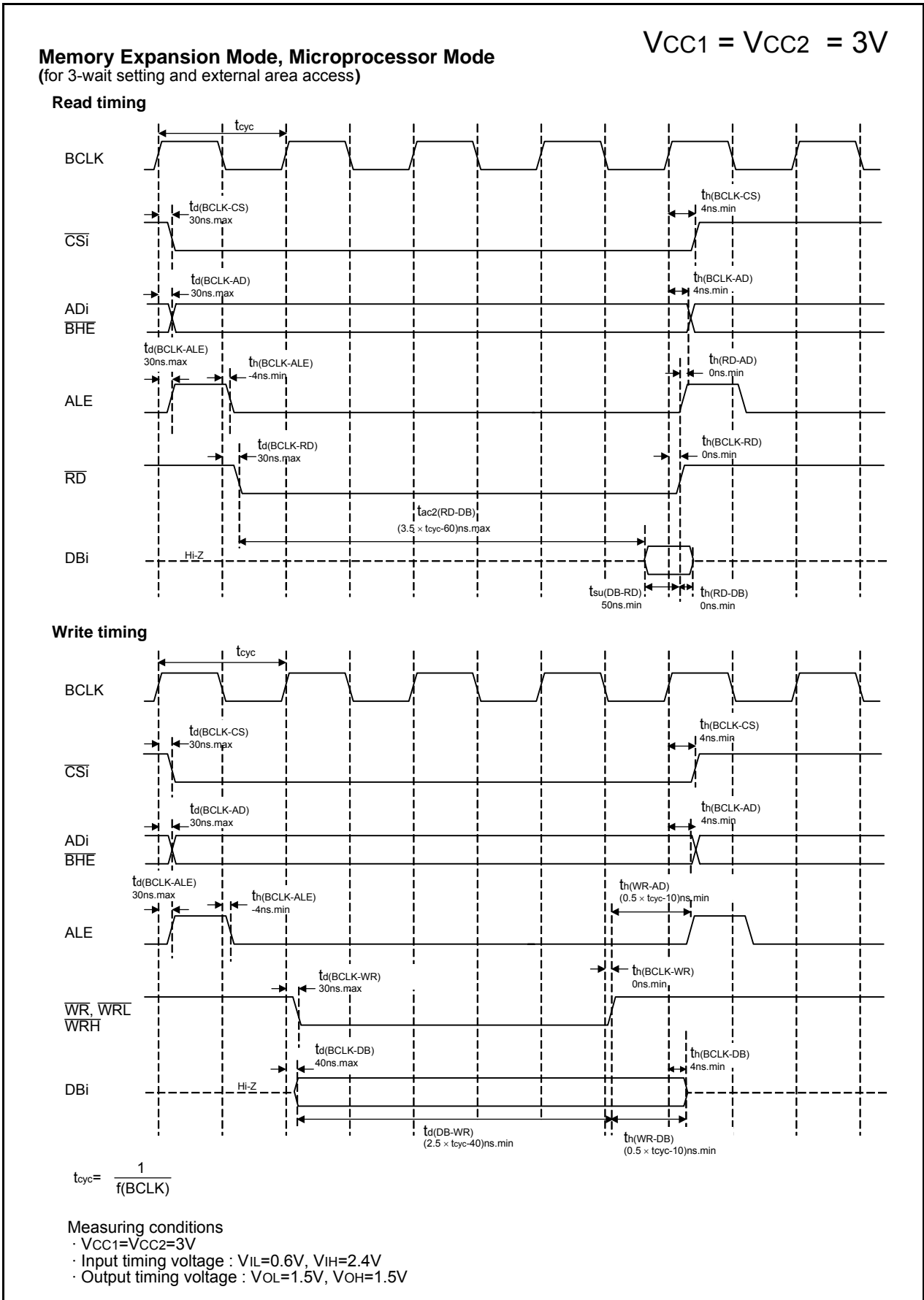


Figure 23.19 Timing Diagram (7)

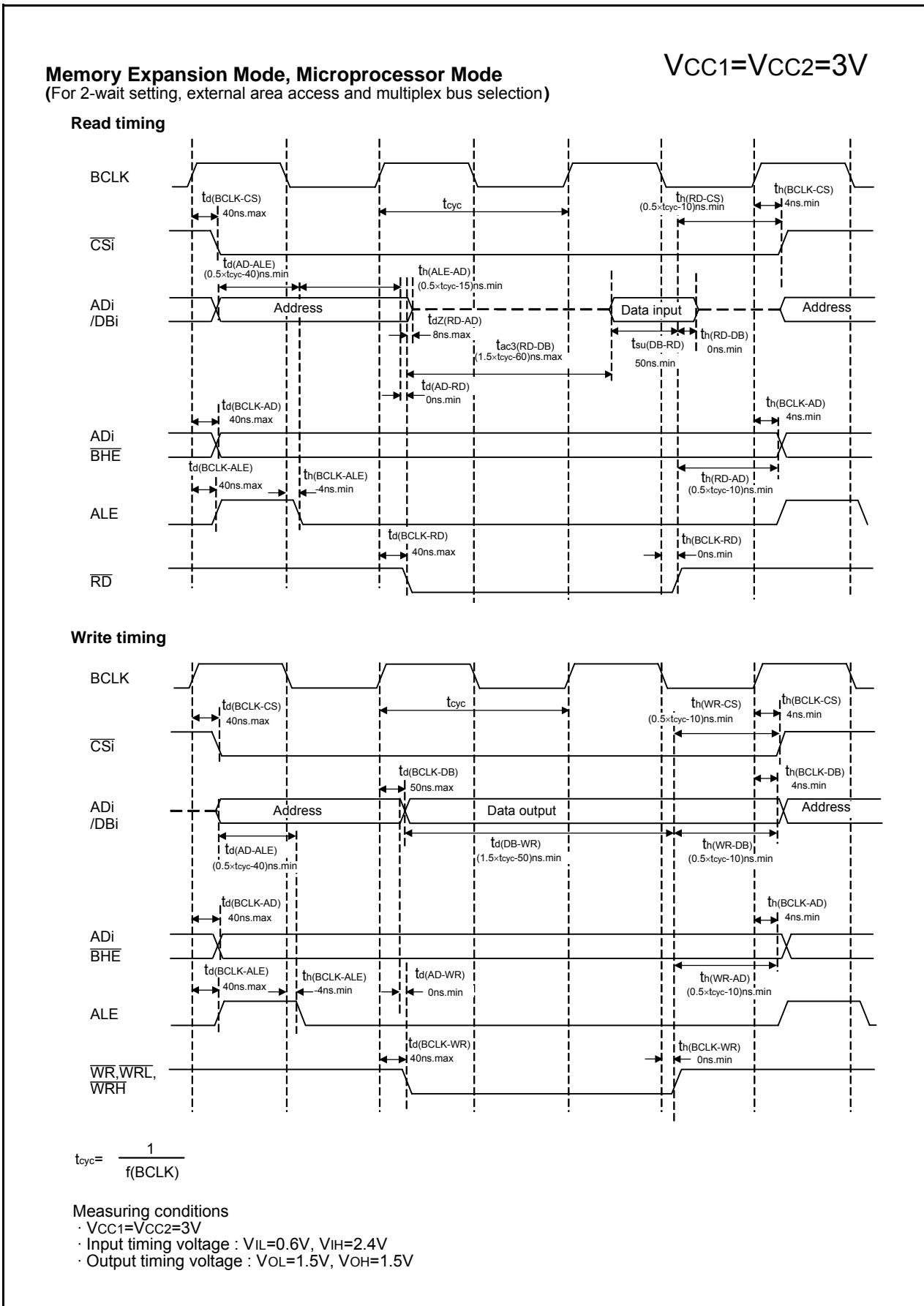


Figure 23.20 Timing Diagram (8)



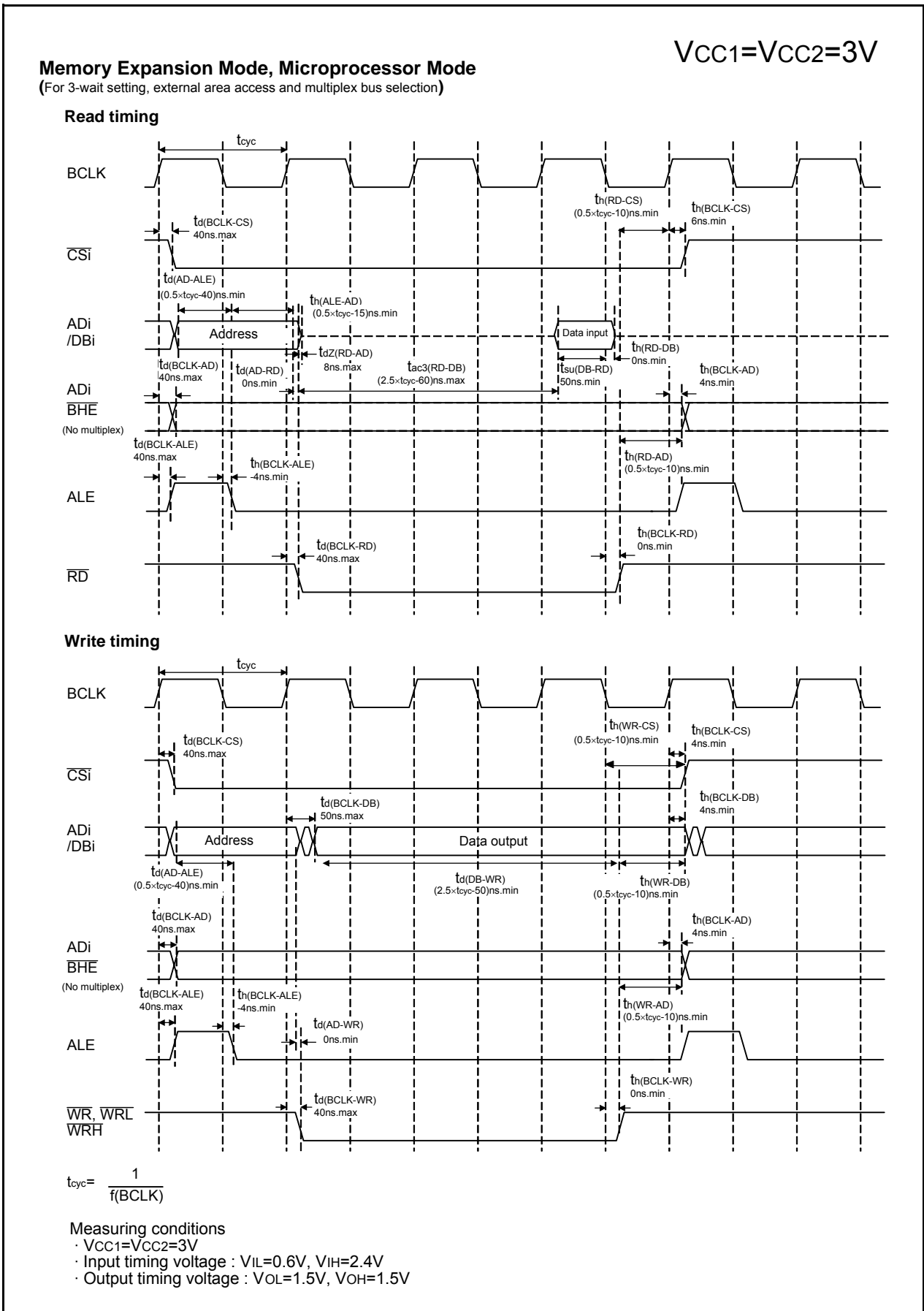


Figure 23.21 Timing Diagram (9)

## 24. Precautions

### 24.1 SFR

#### 24.1.1 Register Settings

Table 24.1 lists Registers with Write-Only Bits. Set these registers with immediate values. When establishing a next value by altering the existing value, write the existing value to the RAM as well as to the register. Transfer the next value to the register after making changes in the RAM.

**Table 24.1 Registers with Write-Only Bits**

Register	Symbol	Address
Watchdog timer reset register	WDTR	037Dh
Watchdog timer start register	WDTS	037Eh
Timer A1-1 register	TA11	0303h to 0302h
Timer A2-1 register	TA21	0305h to 0304h
Timer A4-1 register	TA41	0307h to 0306h
Dead time timer	DTT	030Ch
Timer B2 interrupt generation frequency set counter	ICTB2	030Dh
SI/O3 bit rate register	S3BRG	0273h
SI/O4 bit rate register	S4BRG	0277h
UART0 bit rate register	U0BRG	0249h
UART1 bit rate register	U1BRG	0259h
UART2 bit rate register	U2BRG	0269h
UART5 bit rate register	U5BRG	0289h
UART6 bit rate register	U6BRG	0299h
UART7 bit rate register	U7BRG	02A9h
UART0 transmit buffer register	U0TB	024Bh to 024Ah
UART1 transmit buffer register	U1TB	025Bh to 025Ah
UART2 transmit buffer register	U2TB	026Bh to 026Ah
UART5 transmit buffer register	U5TB	028Bh to 028Ah
UART6 transmit buffer register	U6TB	029Bh to 029Ah
UART7 transmit buffer register	U7TB	02ABh to 02AAh
Timer A0 register	TA0	0327h to 0326h
Timer A1 register	TA1	0329h to 0328h
Timer A2 register	TA2	032Bh to 032Ah
Timer A3 register	TA3	032Dh to 032Ch
Timer A4 register	TA4	032Fh to 032Eh

## 24.2 Reset

### 24.2.1 VCC1

When supplying power to the microcomputer, the power supply voltage applied to the VCC1 pin must meet the conditions of SVCC.

Symbol	Parameter	Standard			Unit
		Min.	Typ.	Max.	
SVCC	Power supply rising gradient (VCC1) (Voltage range 0 to 2)	0.05			V / ms

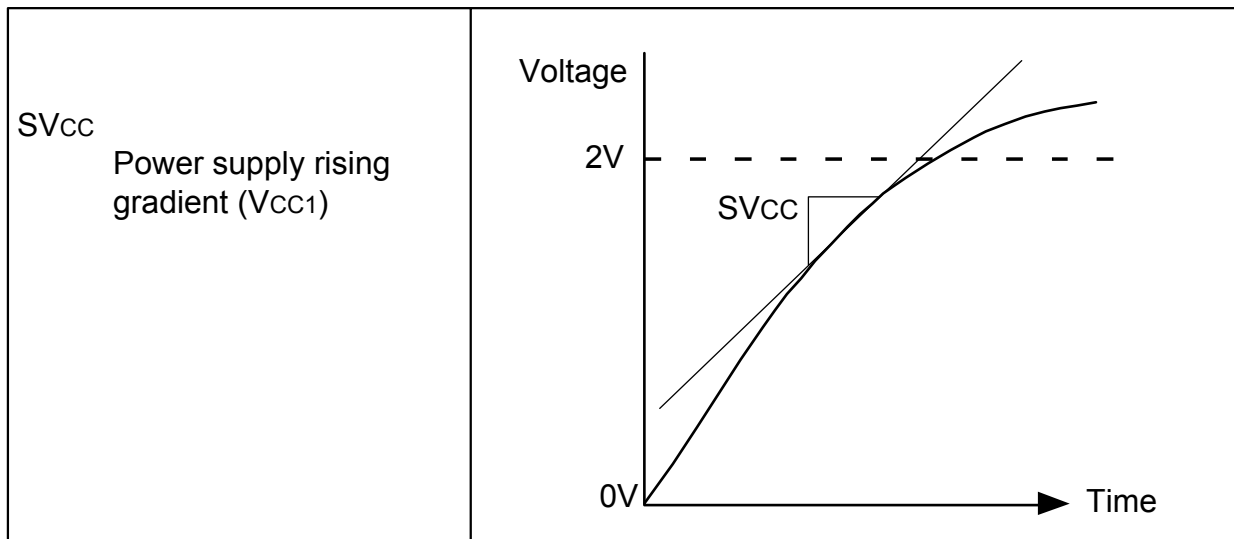


Figure 24.1 Timing of SVCC

### 24.2.2 CNVSS

To start to operate in single-chip mode after reset, connect to VSS via resistor. The internal pull-up of the CNVSS pin is on immediately after hardware reset 1 or 2 is released in single-chip mode. Therefore, the CNVSS pin level becomes high for two cycles of fOCO-S maximum.

### 24.3 Bus

- When hardware reset 1 or brown-out reset is performed with “H” input on the CNVSS pin, contents of internal ROM cannot be read.

### 24.4 PLL Frequency Synthesizer

To use the PLL frequency synthesizer, stabilize supply voltage so that the standard of the power supply ripple is met.

Symbol	Parameter	Standard			Unit
		Min.	Typ.	Max.	
f (ripple)	Power supply ripple allowable frequency (VCC1)			10	kHz
VP-P (ripple)	Power supply ripple allowable amplitude voltage	(VCC1 = 5V)		0.5	V
		(VCC1 = 3V)		0.3	V
VCC ( $\Delta V / \Delta T$ )	Power supply ripple rising / falling gradient	(VCC1 = 5V)		0.3	V / ms
		(VCC1 = 3V)		0.3	V / ms

<p>f (ripple) Power supply ripple allowable frequency (VCC1)</p> <p>Vp-p (ripple) Power supply ripple allowable amplitude voltage</p>	
-----------------------------------------------------------------------------------------------------------------------------------------------	--

Figure 24.2 Voltage Fluctuation Timing

## 24.5 Power Control

- When exiting stop mode by hardware reset 1, set the  $\overline{\text{RESET}}$  pin to “L” until a main clock oscillation is stabilized.
- Set the MR0 bit in the TAIMR register ( $i = 0$  to 4) to 0 (pulse is not output) to use the timer A to exit stop mode.
- After the WAIT instruction, insert at least four NOP instructions. When entering wait mode, the instruction queue reads ahead the instructions following WAIT, and depending on timing, some of these may execute before the microcomputer enters wait mode.

Program example when entering wait mode is shown below.

```

Program Example:      FSET      I      ;
                     WAIT      ;Enter wait mode
                     NOP      ;More than four NOP instructions
                     NOP
                     NOP
                     NOP

```

- When entering stop mode, insert a JMP.B instruction immediately after executing an instruction which sets the CM10 bit in the CM1 register to 1, and then insert at least four NOP instructions. When entering stop mode, the instruction queue reads ahead the instructions following the instruction which sets the CM10 bit to 1 (all clock stop), and some of these may execute before the microcomputer enters stop mode or before the interrupt routine for returning from stop mode.

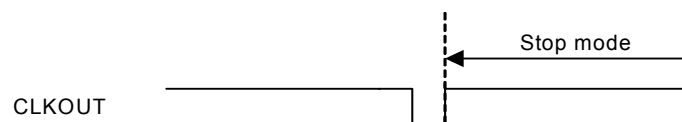
Program example when entering stop mode

```

Program Example:      FSET      I
                     BSET      0, CM1 ; Enter stop mode
                     JMP.B     L2    ; Insert a JMP.B instruction
L2:
                     NOP      ; More than four NOP instructions
                     NOP
                     NOP
                     NOP

```

- The CLKOUT pin outputs high in stop mode. Therefore, when the CLKOUT pin changes state from high to low and is immediately driven in stop mode, the low level width becomes short.



- Wait until the main clock oscillation stabilizes, before switching the clock source for the CPU clock to the main clock. Similarly, wait until the sub clock oscillates stably before switching the clock source for the CPU clock to the sub clock.
- Do not stop the externally-generated clock when the externally-generated clock is input to the XIN pin and the main clock is used as the clock source for the CPU clock.

- Suggestions to reduce power consumption  
Refer to the following descriptions when designing a system or programming.

**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 output ports. A pass current flows to input ports in high-impedance state. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

**A/D converter**

When A/D conversion is not performed, set the ADSTBY bit in the ADCON1 register to 0 (A/D operation stop). When A/D conversion is performed, start the A/D conversion at least 1  $\phi_{AD}$  cycle or longer after setting the ADSTBY bit to 1 (A/D operation enabled).

**D/A converter**

When not performing D/A conversion, set the DAiE bit ( $i = 0, 1$ ) in the DACON register to 0 (output inhibited) and the DAi register to 00h.

**Stopping peripheral functions**

Use the CM02 bit in the CM0 register to stop the unnecessary peripheral functions during wait mode.

**Switching the oscillation-driving capacity**

Set the driving capacity to "L" when oscillation is stable.

## 24.6 Protect

Set the PRC2 bit to 1 (write enabled) and then write to given SFR address, and the PRC2 bit will be cleared to 0 (write protected). Change the registers protected by the PRC2 bit 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.



## 24.7 Interrupt

### 24.7.1 Reading address 00000h

Do not read the address 00000h 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 00000h during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to 0. If the address 00000h is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to 0. This factors a problem that the interrupt is canceled, or an unexpected interrupt request is generated.

### 24.7.2 SP Setting

Set any value in the SP (USP, ISP) before accepting an interrupt. The SP (USP, ISP) is cleared to 0000h 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 the  $\overline{\text{NMI}}$  interrupt, set a value in the ISP at the beginning of the program. Only for the first instruction after reset, all interrupts including the  $\overline{\text{NMI}}$  interrupt are disabled.

### 24.7.3 $\overline{\text{NMI}}$ Interrupt

- The  $\overline{\text{NMI}}$  interrupt cannot be disabled. If this interrupt is not used, set the PM24 bit in the PM2 register to 0 (port P8\_5 function).
- Stop mode cannot be entered into while input on the  $\overline{\text{NMI}}$  pin is low because the CM10 bit in the CM1 register is fixed to 0.
- Do not enter wait mode while input on the  $\overline{\text{NMI}}$  pin is low because the CPU clock remains active even though the CPU stops, and therefore, the current consumption in the chip does not drop. In this case, normal condition is restored by a subsequent interrupt generated.
- Set the low and high level durations of the input signal to the  $\overline{\text{NMI}}$  pin to 2 CPU clock cycles + 300 ns or more.

### 24.7.4 Changing an 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). To use an interrupt, change the interrupt generate factor, and then 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 source, 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 24.3 shows the Procedure for Changing the Interrupt Generate Factor.

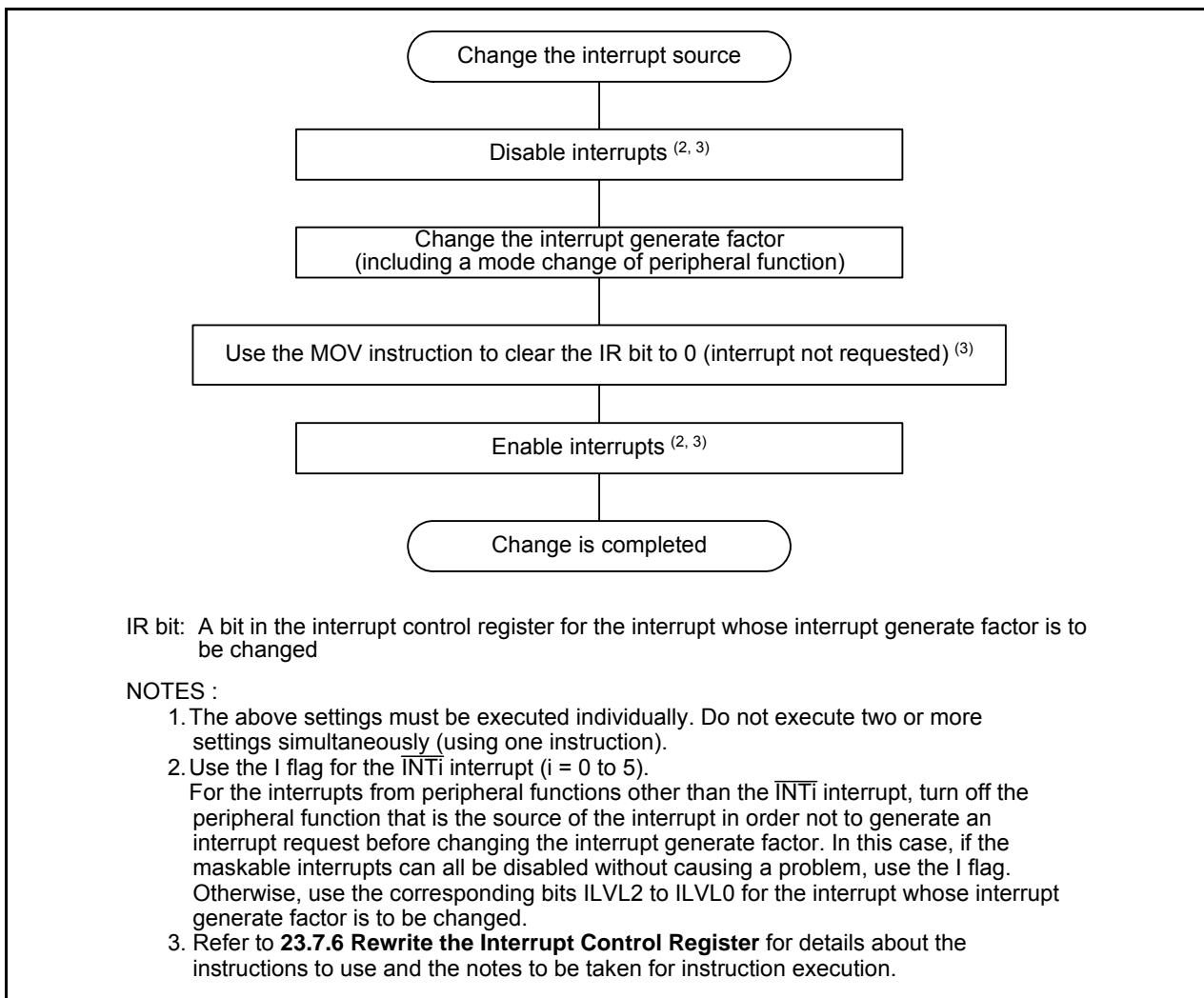


Figure 24.3 Procedure for Changing the Interrupt Generate Factor

### 24.7.5 $\overline{\text{INT}}$ Interrupt

- Either an “L” level of at least  $t_w$  (INL) width or an “H” level of at least  $t_w$  (INH) width is necessary for the signal input to pins  $\overline{\text{INT}}_0$  through  $\overline{\text{INT}}_7$  regardless of the CPU operation clock.
- If the POL bit in registers INT0IC to INT7IC, bits IFSR7 to IFSR0 in the IFSR register, or bits IFSR31 and IFSR30 in the IFSR3A 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.

## 24.7.6 Rewriting the Interrupt Control Register

- (a) 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.
- (b) 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  
When interrupts corresponding to the register occur, the IR bit may not become 1 (interrupt requested) and the interrupts may be ignored. If this causes any troubles, use any of the following instructions to change registers.  
Instruction: AND, OR, BCLR, or 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.
- (c) When using the I flag to disable an interrupt, set the I flag while referring to the sample program fragments shown below. (Refer to (b) for details about rewriting 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 (interrupt enabled) before the interrupt control register is rewritten, 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 00h.
NOP
NOP
FSET      I           ; Enable interrupts.
```

The number of the NOP instructions is as follows.

PM20 = 1 (1 wait) : 2, PM20 = 0 (2 waits) : 3, when using the HOLD function : 4.

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 00h.
MOV.W    MEM, R0     ; Dummy read.
FSET      I           ; Enable interrupts.
```

Example 3: Using the POPC instruction to change the I flag

```
INT_SWITCH3:
PUSHC    FLG
FCLR      I           ; Disable interrupts.
AND.B     #00h, 0055h ; Set the TA0IC register to 00h.
POPC     FLG         ; Enable interrupts.
```

### 24.7.7 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt occurs.

## 24.8 DMAC

### 24.8.1 Write to the DMAE Bit in the DMiCON Register (i = 0 to 3)

When both of the conditions below are met, follow the steps below.

#### Conditions

- The DMAE bit is set to 1 (DMAi is in active state) again while it remains 1.
- A DMA request may occur simultaneously when the DMAE bit is being written.

#### Steps

- (1) Write a 1 to the DMAE bit and DMAS bit in the DMiCON register simultaneously (1).
- (2) Make sure that the DMAi is in initial state <sup>(2)</sup> in a program.  
If the DMAi is not in initial state, repeat the above steps.

#### NOTE:

1. The DMAS bit remains unchanged even if a 1 is written. However, if a 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.
2. 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.
3. Read the TCRi register to verify whether the DMAi is in 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 initial state. (In the case a DMA request occurs after writing to the DMAE bit, the read value is a value written to the TCRi register minus one.) If the read value is a value in the middle of transfer, the DMAi is not in initial state.

## 24.9 Timers

### 24.9.1 Timer A

#### 24.9.1.1 Timer A (Timer Mode)

The timer is stopped after reset. Set the mode, count source, counter value, and others using registers TAI<sub>i</sub>MR, TAI<sub>i</sub>, TACS0 to TACS2, and TAPOFS before setting the TAI<sub>i</sub>S bit in the TABSR register to 1 (count starts) (i = 0 to 4).

Always make sure registers TAI<sub>i</sub>MR, TACS0 to TACS2, and TAPOFS are modified while the TAI<sub>i</sub>S bit is 0 (count stops) regardless of whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAI<sub>i</sub> register. However, if the counter is read at the same time it is reloaded, the value FFFFh is read. Also, if the counter is read before it starts counting after a value is set in the TAI<sub>i</sub> register while not counting, the set value is read.

If a low-level signal is applied to the  $\overline{SD}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on the  $\overline{SD}$  pin enabled), pins TA1OUT, TA2OUT, and TA4OUT go to high-impedance state.

### 24.9.1.2 Timer A (Event Counter Mode)

The timer is stopped after reset. Set the mode, count source, counter value, and others using the TAI<sub>i</sub>MR register, the TAI<sub>i</sub> register, the UDF register, bits TAZIE, TA0TGL, and TA0TGH in the ONSF register and the TRGSR register, and TAPOS register before setting the TAI<sub>i</sub>S bit in the TABSR register to 1 (count starts) (i = 0 to 4).

Always make sure the TAI<sub>i</sub>MR register, the UDF register, bits TAZIE, TA0TGL, and TA0TGH in the ONSF register, the TRGSR register, and TAPOS register are modified while the TAI<sub>i</sub>S bit is 0 (count stops) regardless of whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAI<sub>i</sub> register. However, while reloading, FFFFh can be read in underflow, and 0000h in overflow. When the counter is read before it starts counting after a value is set in the TAI<sub>i</sub> register while not counting, the set value is read.

If a low-level signal is applied to the  $\overline{SD}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on  $\overline{SD}$  pin enabled), pins TA1OUT, TA2OUT, and TA4OUT go to high-impedance state.

### 24.9.1.3 Timer A (One-Shot Timer Mode)

The timer is stopped after reset. Set the mode, count source, counter value, and others using the TAI<sub>i</sub>MR register, the TAI<sub>i</sub> register, bits TA0TGL and TA0TGH in the ONSF register, the TRGSR register, registers TACS<sub>0</sub> to TACS<sub>2</sub> and the TAPOFS register before setting the TAI<sub>i</sub>S bit in the TABSR register to 1 (count starts) (i = 0 to 4).

Always make sure the TAI<sub>i</sub>MR register, bits TA0TGL and TA0TGH in the ONSF register, the TRGSR register, registers TACS<sub>0</sub> to TACS<sub>2</sub>, and the TAPOFS register are modified while the TAI<sub>i</sub>S bit is 0 (count stops) regardless of whether after reset or not.

When setting the TAI<sub>i</sub>S bit to 0 (count stops), the followings occur:

- A counter stops counting and a content of reload register is reloaded.
- The TAI<sub>i</sub>OUT pin outputs “L” when the POFS<sub>i</sub> bit in the TAPOFS register is 0; outputs “H” when 1.
- After one cycle of the CPU clock, the IR bit in the TAI<sub>i</sub>C register is set to 1 (interrupt requested).

Output in one-shot timer mode synchronizes with a count source internally generated. When an external trigger is selected, one-and-half-cycle delay of a count source as maximum occurs between a trigger input to the TAI<sub>i</sub>IN pin and output in one-shot timer mode.

The IR bit is set to 1 when timer operating mode is set with any of the following procedures:

- Select one-shot timer mode after reset.
- Change an operating mode from timer mode to one-shot timer mode.
- Change an operating mode from event counter mode to one-shot timer mode.

To use the Timer A<sub>i</sub> interrupt (the IR bit), set the IR bit to 0 after the changes listed above are made.

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 generating the previous trigger and operating longer than one cycle of a timer count source.

If a low-level signal is applied to the  $\overline{SD}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on the  $\overline{SD}$  pin enabled), pins TA1OUT, TA2OUT, and TA4OUT go to high-impedance state.



#### 24.9.1.4 Timer A (Pulse Width Modulation Mode)

The timer is stopped after reset. Set the mode, count source, counter value, and others using the TAI<sub>i</sub>MR register, the TAI<sub>i</sub> register, bits TA0TGL and TA0TGH in the ONSF register, the TRGSR register, registers TACS0 to TACS2, and the TAPOF register before setting the TAI<sub>i</sub>S bit in the TABSR register to 1 (count starts) (i = 0 to 4).

Always make sure the TAI<sub>i</sub>MR register, bits TA0TGL and TA0TGH in the ONSF register, the TRGSR register, registers TACS0 to TACS2, and the TAPOFS register are modified while the TAI<sub>i</sub>S bit is 0 (count stops) regardless of whether after reset or not.

The IR bit is set to 1 when setting a timer operating mode with any of the following procedures:

- Select PWM mode after reset.
- Change an operating mode from timer mode to PWM mode.
- Change an operating mode from event counter mode to PWM mode.

To use the timer A<sub>i</sub> interrupt (IR bit), set the IR bit to 0 by program after the changes listed above are made.

When setting the TAI<sub>i</sub>S register to 0 (count stops) during PWM pulse output, the following action occurs.

When the POFS<sub>i</sub> bit in the TAPOFS register is 0:

- Stop counting.
- When the TAI<sub>i</sub>OUT pin is output "H," output level is set to "L" and the IR bit is set to 1.
- When the TAI<sub>i</sub>OUT pin is output "L," both output level and the IR bit remains unchanged.

When the POFS<sub>i</sub> bit in the TAPOFS register is 1:

- Stop counting.
- When the TAI<sub>i</sub>OUT pin is output "L," output level is set to "H" and the IR bit is set to 1.
- When the TAI<sub>i</sub>OUT pin is output "H," both output level and the IR bit remains unchanged.

If a low-level signal is applied to the  $\overline{SD}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on the  $\overline{SD}$  pin enabled), pins TA1OUT, TA2OUT, and TA4OUT go to high-impedance state.

## 24.9.2 Timer B

### 24.9.2.1 Timer B (Timer Mode)

The timer is stopped after reset. Set the mode, count source, counter value, and others using registers TBiMR, TBi, and TBCS0 to TBCS3 before setting the TBiS bit in the TABSR or the TBSR register to 1 (count starts) (i = 0 to 5).

Always make sure the TBiMR register and registers TBCS0 to TBCS3 are modified while the TBiS bit is 0 (count stops) regardless of whether after reset or not.

A value of a counter while counting, can be read in the TBi register at any time. FFFFh is read while reloading. If the counter is read before it starts counting after a value is set in the TBi register while not counting, the set value is read.

### 24.9.2.2 Timer B (Event Counter Mode)

The timer is stopped after reset. Set the mode, count source, counter value, and others using the TBiMR register and TBi register before setting the TBiS bit in the TABSR or the TBSR register to 1 (count starts) (i = 0 to 5).

Always make sure the TBiMR register is modified while the TBiS bit is 0 (count stops) regardless of whether after reset or not.

While counting is in progress, the counter value can be read out 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 FFFFh. 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 value set in the register.

### 24.9.2.3 Timer B (Pulse Period / Pulse Width Measurement Mode)

The timer is stopped after reset. Set the mode, count source, etc. using the TBiMR register before setting the TBiS bit in the TABSR or the TBSR register to 1 (count starts) ( $i = 0$  to 5).

Always make sure the TBiMR register and registers TBCS0 to TBCS3 are modified while the TBiS bit is 0 (count stops) regardless of 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 bits TMOD0, TMOD1, MR0, MR1, TCK0, and TCK1 and a 0 to bit 4.

The IR bit in the TBiIC register goes to 1 (interrupt requested) when an effective edge of a measurement pulse is input or Timer Bi is overflowed ( $i = 0$  to 5). The factor of interrupt request can be determined by use of the MR3 bit in the TBiMR register within the interrupt routine.

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.

Use the IR bit in the TBiIC register to detect only overflows. Use the MR3 bit only to determine the interrupt factor.

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.

A value of the counter is indeterminate after reset. If a count is started in this state, the MR3 bit may be set to 1 and Timer Bi interrupt request may be generated after a count start before an effective edge input. When a value is set to the TBi register while the TBiS bit is 0 (count stops), the same value is written to the counter.

For pulse width measurement, pulse widths are successively measured. Use program to check whether the measurement result is "H" level width or "L" level width.

## 24.10 Serial Interface

### 24.10.1 Clock Synchronous Serial I/O

#### 24.10.1.1 Transmission / Reception

When the  $\overline{\text{RTS}}$  function is used with an external clock,  $\overline{\text{RTSi}}$  pin ( $i = 0$  to 2, 5 to 7) outputs “L,” which informs the transmitting side that the MCU is ready for a receive operation. The  $\overline{\text{RTSi}}$  pin outputs “H” when a receive operation starts. Therefore, a transmit timing and receive timing can be synchronized by connecting the  $\overline{\text{RTSi}}$  pin to the  $\overline{\text{CTSi}}$  pin of the transmitting side. The  $\overline{\text{RTS}}$  function is disabled when an internal clock is selected.

If a low-level signal is applied to the  $\overline{\text{SD}}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on  $\overline{\text{SD}}$  pin enabled), the following pins go to high-impedance state:

P7\_2/CLK2/TA1OUT/V, P7\_3/ $\overline{\text{CTS2}}$ / $\overline{\text{RTS2}}$ /TA1IN/ $\overline{\text{V}}$ , P7\_4/TA2OUT/W, P7\_5/TA2IN/ $\overline{\text{W}}$ , P8\_0/TA4OUT/RXD5/SCL5/U, P8\_1/TA4IN/ $\overline{\text{CTS5}}$ / $\overline{\text{RTS5}}$ / $\overline{\text{U}}$

### 24.10.1.2 Transmission

If an external clock is selected, the following conditions must be met while the external clock is held “H” when the CKPOL bit in the UiC0 register (i = 0 to 2, 5 to 7) is set to 0 (transmit data output at the falling edge and receive data input at the rising edge of the serial clock), or while the external clock is held “L” when the CKPOL bit is set to 1 (transmit data output at the rising edge and receive data input at the falling edge of the serial clock).

- The TE bit in the UiC1 register = 1 (transmission enabled)
- The TI bit in the UiC1 register = 0 (data present in the UiTB register)
- If CTS function is selected, input on the CTSi pin = “L”

### 24.10.1.3 Reception

In clock synchronous serial I/O mode, the shift clock is generated by activating a transmitter. Set the UAR*T**i*-associated registers for a transmit operation even if the MCU is used for receive operation only. Dummy data is output from the TXD*i* pin (i = 0 to 2, 5 to 7) while receiving.

When an internal clock is selected, the shift clock is generated by setting the TE bit in the UiC1 register to 1 (transmission enabled) and placing dummy data in the UiTB register. When an external clock is selected, set the TE bit to 1 (transmission enabled), place dummy data in the UiTB register, and input an external clock to the CLK*i* pin to generate the shift clock.

If data is received consecutively, an overrun error occurs when the RI bit in the UiC1 register is set to 1 (data present in the UiRB register) and the next receive data is received in the UAR*T**i* receive register. And then, the OER bit in the UiRB register is set to 1 (overrun error occurred). At this time, the UiRB register is undefined. If an overrun error occurs, the IR bit in the SiRIC register remains unchanged.

To receive data consecutively, set dummy data in the low-order byte in the UiTB register per each receive operation.

If an external clock is selected, the following conditions must be met while the external clock is held “H” when the CKPOL bit is set to 0 (transmit data output at the falling edge and receive data input at the rising edge of the serial clock), or while the external clock is held “L” when the CKPOL bit is set to 1 (transmit data output at the rising edge and receive data input at the falling edge of the serial clock).

- The RE bit in the UiC1 register = 1 (reception enabled)
- The TE bit in the UiC1 register = 1 (transmission enabled)
- The TI bit in the UiC1 register = 0 (data present in the UiTB register)

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

### 24.10.2.1 Transmission / Reception

When the  $\overline{RTS}$  function is used with an external clock,  $\overline{RTSi}$  pin ( $i = 0$  to  $2, 5$  to  $7$ ) outputs “L,” which informs the transmitting side that the MCU is ready for a receive operation. The  $\overline{RTSi}$  pin outputs “H” when a receive operation starts. Therefore, a transmit timing and receive timing can be synchronized by connecting the  $\overline{RTSi}$  pin to the  $\overline{CTS}$  pin of the transmitting side. The  $\overline{RTS}$  function is disabled when an internal clock is selected.

If a low-level signal is applied to the  $\overline{SD}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on  $\overline{SD}$  pin enabled), the following pins go to high-impedance state:

P7\_2/CLK2/TA1OUT/V, P7\_3/ $\overline{CTS}$ 2/ $\overline{RTS}$ 2/TA1IN/ $\overline{V}$ , P7\_4/TA2OUT/W, P7\_5/TA2IN/ $\overline{W}$ , P8\_0/TA4OUT/RXD5/SCL5/U, P8\_1/TA4IN/ $\overline{CTS}$ 5/ $\overline{RTS}$ 5/ $\overline{U}$

### 24.10.2.2 Transmission

If an external clock is selected, the following conditions must be met while the external clock is held “H” when the CKPOL bit in the UiC0 register ( $i = 0$  to  $2, 5$  to  $7$ ) is set to 0 (transmit data output at the falling edge and receive data input at the rising edge of the serial clock), or while the external clock is held “L” when the CKPOL bit is set to 1 (transmit data output at the rising edge and receive data input at the falling edge of the serial clock).

- The TE bit in the UiC1 register = 1 (transmission enabled)
- The TI bit in the UiC1 register = 0 (data present in the UiTB register)
- If  $\overline{CTS}$  function is selected, input on the  $\overline{CTS}$  pin = “L”

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

When generating start, stop, and restart conditions, set the STSPSEL bit in the UiSMR4 register ( $i = 0$  to  $2, 5$  to  $7$ ) to 0 and wait for more than half cycle of the transfer clock before setting each condition generation bit (STAREQ, RSTAREQ, and STPREQ) from 0 to 1.

### 24.10.4 Special Mode 4 (SIM Mode)

A transmit interrupt request is generated by setting bits U2IRS and U2ERE in the U2C1 register to 1 (transmission completed) and 1 (error signal output), respectively. Therefore, when using SIM mode, make sure to clear the IR bit to 0 (interrupt not requested) after setting these bits.

### 24.10.5 S<sub>I/O3</sub>, S<sub>I/O4</sub>

The SOUT<sub>i</sub> default value which is set to the SOUT<sub>i</sub> pin by the SMi7 bit approximately 10 ns may be output when changing the SMi3 bit from 0 (I/O port) to 1 (SOUT<sub>i</sub> output and CLK function) while the SMi2 bit in the SiC to 0 (SOUT<sub>i</sub> output) and the SMi6 bit is set to 1 (internal clock). And then the SOUT<sub>i</sub> pin is held high-impedance.

If the output level from the SOUT<sub>i</sub> pin is a problem when changing the SMi3 bit from 0 to 1, set the default value of the SOUT<sub>i</sub> pin by the SMi7 bit.

i = 3, 4



### 24.11 A/D Converter

Set registers ADCON0 (except bit 6), ADCON1, and ADCON2 when A/D conversion is stopped (before a trigger occurs). After A/D conversion is stopped, set the ADSTBY bit from 1 to 0.

When the ADSTBY bit in the ADCON1 register is changed from 0 (A/D operation stopped) to 1 (A/D operation enabled), wait for 1  $\phi_{AD}$  cycle or longer to start A/D conversion.

To prevent noise-induced device malfunction or latchup, as well as to minimize conversion errors, insert capacitors between pins AVCC, VREF, analog input (ANi (i = 0 to 7), AN0\_i, AN2\_i), and AVSS. Similarly, insert a capacitor between pins VCC1 and VSS. Figure 24.4 shows an example connection of individual pin.

Make sure the port direction bits corresponding to the pins that are used as analog inputs are set to 0 (input mode).

When the TRG bit in the ADCON0 register = 1 (external trigger), make sure the port direction bit for the ADTRG pin is set to 0 (input mode).

When using key input interrupts, do not use any of the four pins AN4 to AN7 as analog inputs. (A key input interrupt request is generated when the A/D input voltage goes low.)

When changing an A/D operating mode, set bits CH2 to CH0 in the ADCON0 register and bits SCAN1 to SCAN0 in the ADCON1 register again to select analog input pins.

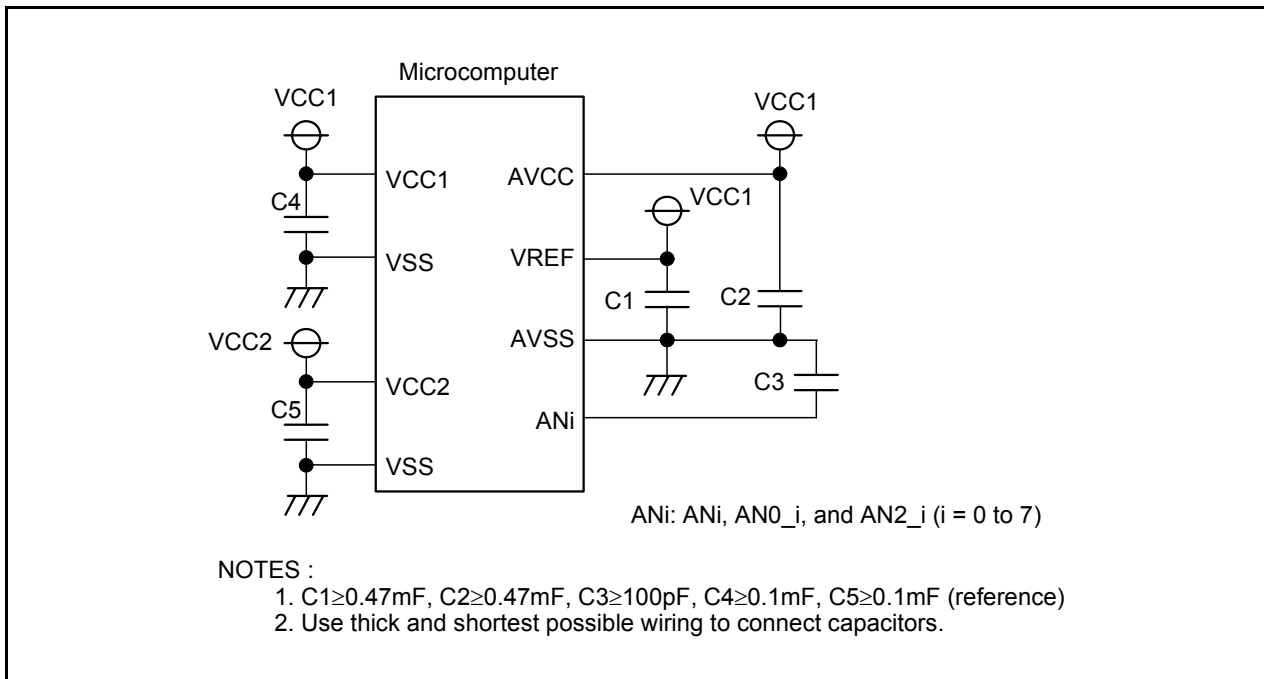


Figure 24.4 Use of Capacitors to Reduce Noise

When A/D conversion is forcibly terminated by setting the ADST bit in the AD0CON0 register to 0 (A/D conversion stops) by program during A/D conversion, the A/D conversion result is undefined. The ADi register not performing A/D conversion may also be undefined. If the ADST bit is set to 0 by program during A/D conversion, do not use values obtained from any ADi registers.

The applied intermediate potential may cause more increase in power consumption to AN4 to AN7 than to other analog input pins (AN0 to AN3, AN0\_0 to AN0\_7, and AN2\_0 to AN2\_7) since AN4 to AN7 are used with KI0 to KI3.

When A/D conversion is stopped in one-shot mode or single sweep mode, the ADST bit in the ADCON0 register becomes 0 (A/D conversion stop). Also, when a trigger by  $\overline{\text{ADTRG}}$  is selected, the ADST bit becomes 0. Therefore, set the ADST bit to 1 (A/D conversion start) by a program if there is a possibility that a trigger is input subsequently.

Connect the VREF pin to VCC1 pin. Because the VREF pin is connected to VCC1 pin inside, current flows if potential difference occurs between the pins.

## 24.12 Programmable I/O Ports

If a low-level signal is applied to the  $\overline{SD}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on the  $\overline{SD}$  pin enabled), pins P7\_2 to P7\_5 and P8\_0 and P8\_1 go to high-impedance state.

Setting the SM32 bit in the S3C register to 1 causes the P9\_2 pin to go to high-impedance state. Similarly, setting the SM42 bit in the S4C register to 1 causes the P9\_6 pin to go to high-impedance state.

## 24.13 Flash Memory Version

### 24.13.1 Functions to Inhibit Rewriting Flash Memory

ID codes are stored in addresses 0FFFDfH, 0FFFE3h, 0FFFEbH, 0FFFEfH, 0FFFF3h, 0FFFF7h, and 0FFFFbH. If wrong data are written to these addresses, the flash memory cannot be read or written in standard serial I/O mode.

The ROMCP register is mapped in address 0FFFFfH. 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.

### 24.13.2 Stop Mode

When the microcomputer enters stop mode, execute the instruction which sets the CM10 bit in the CM1 register to 1 (stop mode) after setting the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) and disabling the DMA transfer.

### 24.13.3 Wait Mode

When shifting to wait mode, set the FMR01 bit to 0 (CPU rewrite mode disabled) before executing the WAIT instruction.

### 24.13.4 Low Power Consumption Mode, On-Chip Oscillator Low Power Consumption Mode

If the CM05 bit in the CM0 register is set to 1 (main clock stop), do not execute the following commands.

- Program
- Block erase
- Lock bit program

### 24.13.5 Writing Command and Data

Write the command code and data at even addresses.

### 24.13.6 Program Command

Write xx41h 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.

### 24.13.7 Lock Bit Program Command

Write xx77h in the first bus cycle and write xxD0h in the second bus cycle to the highest-order even address of a block, and the lock bit for the specified block is cleared to 0. Make sure then address value specified in the first bus cycle is the same highest-order block address that is specified in the second bus cycle.

### 24.13.8 Operation Speed

Before entering CPU rewrite mode (EW0 or EW1 mode), set the CM11 bit in the CM1 register to 0 (main clock), select 10 MHz or less for CPU clock using the CM06 bit in the CM0 register and bits CM17 and CM16 in the CM1 register. Also, set the PM17 bit in the PM1 register to 1 (with wait state).

### 24.13.9 Instructions Inhibited against Use

The following instructions cannot be used in EW0 mode because the flash memory internal data is referred: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction.

### 24.13.10 Interrupts

#### EW0 Mode

- Any interrupt which has a vector in the relocatable vector table can be used providing that its vector is transferred into the RAM area.
- The  $\overline{\text{NMI}}$  and watchdog timer (oscillation stop, re-oscillation detect, and low voltage detect) interrupts can be used because registers FMR0 and FMR1 are initialized when one of those interrupts occurs. The jump addresses for those interrupt routines should be set in the fixed vector table.
- Because the rewrite operation stops when an  $\overline{\text{NMI}}$  or watchdog timer (oscillation stop, re-oscillation detect, and low voltage detect) interrupt occurs, the rewrite program must be executed again after exiting the interrupt routine.
- The address match interrupt cannot be used because the flash memory internal data is referred.

#### EW1 Mode

- Make sure that any interrupt which has a vector in the relocatable vector table or address match interrupt will not be accepted during the auto program or auto erase period.
- Avoid using watchdog timer (oscillation stop, re-oscillation detect, and low voltage detect) interrupts.
- The  $\overline{\text{NMI}}$  interrupt can be used because registers FMR0 and FMR1 are initialized when this interrupt occurs. The jump address for the interrupt routine should be set in the fixed vector table.
- Because the rewrite operation is halted when an  $\overline{\text{NMI}}$  interrupt occurs, execute the rewrite program again after exiting the interrupt routine.

### 24.13.11 How to Access

To set the FMR01, FMR02, or FMR11 bit to 1, write 0 and then 1 in succession. Ensure that no interrupts or DMA transfers will occur before writing 1 after writing 0. Perform it when the  $\overline{\text{NMI}}$  pin is "H" level if the PM24 bit is 1 ( $\overline{\text{NMI}}$ ).

### 24.13.12 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, the rewrite control program may not be correctly rewritten and, consequently, the flash memory becomes unable to be rewritten thereafter. In this case, use standard serial I/O or parallel I/O mode.

#### EW1 Mode

- Avoid rewriting any block in which the rewrite control program is stored.

### 24.13.13 DMA transfer

In EW1 mode, make sure that no DMA transfers will occur while the FMR00 bit in the FMR0 register = 0 (during the auto program or auto erase period).

#### 24.13.14 Programming / Erasing Endurance and Execution Time

As the number of programming / erasing times increases, so does the execution time for software commands (program, block erase, and lock bit program commands).

The software commands are stopped by hardware reset 1, brown-out reset,  $\overline{\text{NMI}}$  interrupt, and watch-dog timer (oscillation stop, re-oscillation detect, and low voltage detect) interrupt. If a software command is stopped by such reset or interrupt, erase the block in process before reexecuting the stopped command.

## 24.14 Noise

Connect a bypass capacitor (approximately 0.1  $\mu\text{F}$ ) across pins VCC1 and VSS, and pins VCC2 and VSS using the shortest and thicker possible wiring. Figure 24.5 shows the Bypass Capacitor Connection.

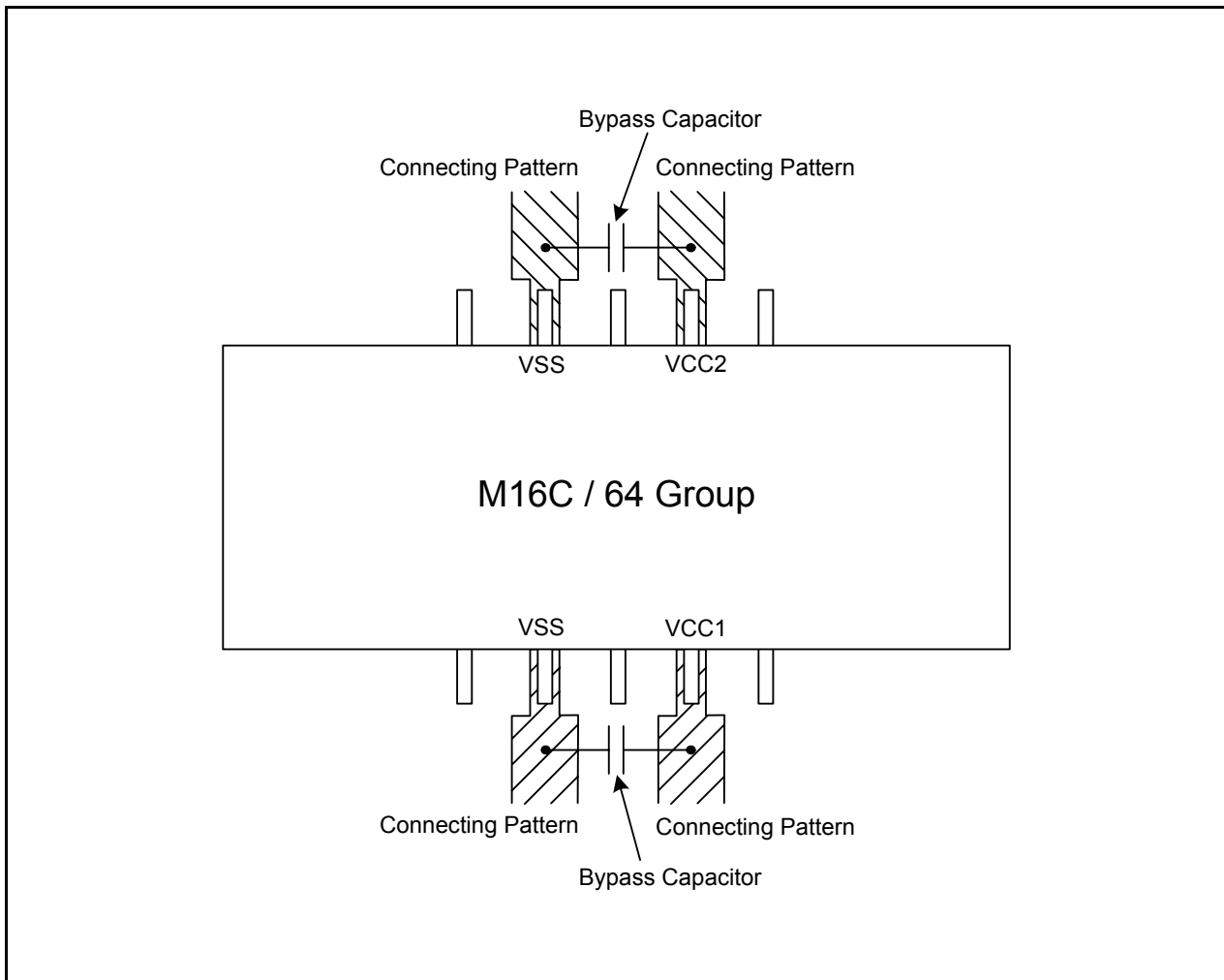
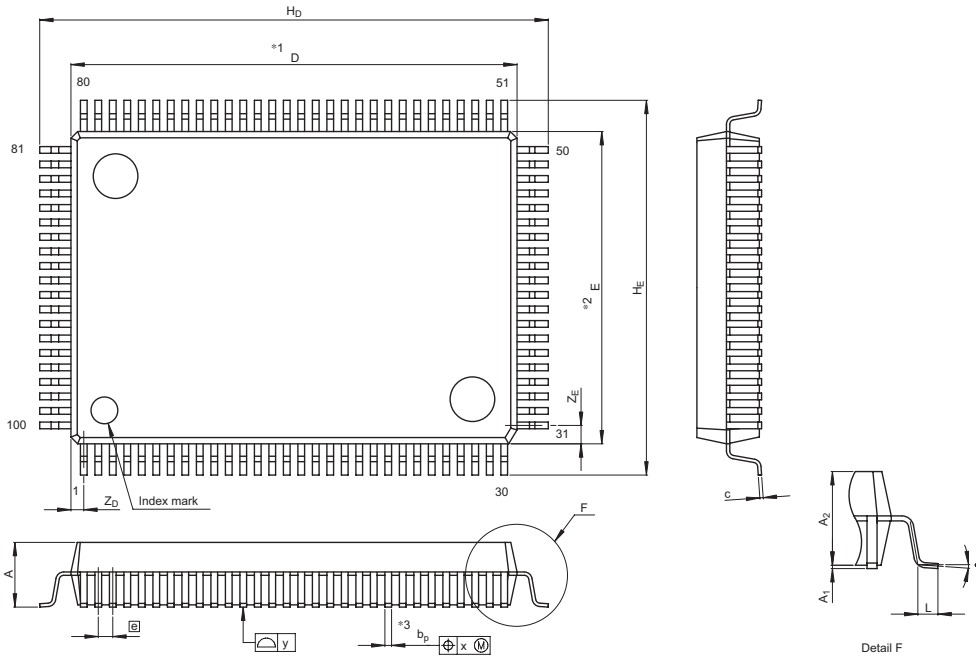


Figure 24.5 Bypass Capacitor Connection

# Appendix 1. Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-QFP100-14x20-0.65	PRQP0100JD-B	100P6F-A	1.8g

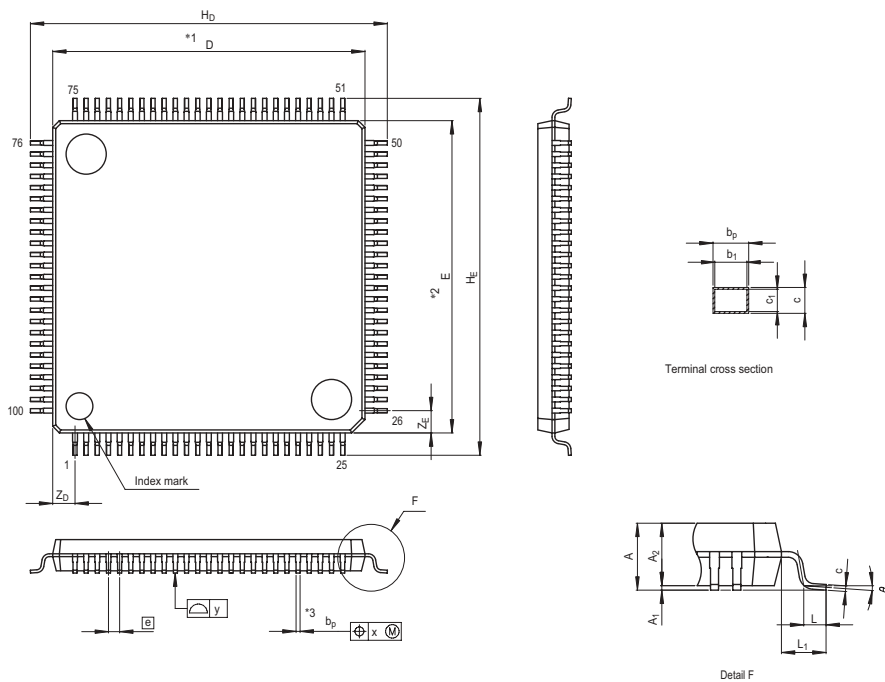
Under development



NOTE)  
 1. DIMENSIONS \*1\* AND \*2\* DO NOT INCLUDE MOLD FLASH.  
 2. DIMENSION \*3\* DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	19.8	20.0	20.2
E	13.8	14.0	14.2
$A_2$	—	2.8	—
$H_D$	22.5	22.8	23.1
$H_E$	16.5	16.8	17.1
A	—	—	3.05
$A_1$	0	0.1	0.2
$b_p$	0.25	0.3	0.4
c	0.13	0.15	0.2
$\theta$	0°	—	10°
$\ominus$	—	0.65	—
x	—	—	0.13
y	—	—	0.10
$Z_D$	—	0.575	—
$Z_E$	—	0.825	—
L	0.4	0.6	0.8

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP100-14x14-0.50	PLQP0100KB-A	100P6Q-A / FP-100U / FP-100UV	0.6g



NOTE)  
 1. DIMENSIONS \*1\* AND \*2\* DO NOT INCLUDE MOLD FLASH.  
 2. DIMENSION \*3\* DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
$A_2$	—	1.4	—
$H_D$	15.8	16.0	16.2
$H_E$	15.8	16.0	16.2
A	—	—	1.7
$A_1$	0.05	0.1	0.15
$b_p$	0.15	0.20	0.25
$d_1$	—	0.18	—
c	0.09	0.145	0.20
$c_1$	—	0.125	—
$\theta$	0°	—	8°
$\ominus$	—	0.5	—
x	—	—	0.08
y	—	—	0.08
$Z_D$	—	1.0	—
$Z_E$	—	1.0	—
L	0.35	0.5	0.65
$L_1$	—	1.0	—



**REGISTER INDEX**

<b>A</b>	
AD0 to AD7 .....	233
ADCON0, ADCON1 .....	
.....	232, 235, 237, 239, 241, 243
ADCON2 .....	233
ADIC .....	105, 106
AIER .....	117
AIER2 .....	117

<b>B</b>	
BCNIC .....	105, 106

<b>C</b>	
CM0 .....	76
CM1 .....	77
CM2 .....	78
CPSRF .....	141, 156
CRCD .....	248
CRCIN .....	248
CSE .....	62
CSPR .....	120
CSR .....	55

<b>D</b>	
D4INT .....	39
DA0, DA1 .....	247
DACON .....	247
DAR0 to DAR3 .....	128
DBR .....	67
DM0CON to DM3CON .....	127
DM0IC .....	106
DM0IC to DM3IC .....	105
DM0SL to DM3SL .....	125
DTT .....	169

<b>F</b>	
FMR0 .....	272
FMR1 .....	273
FMR6 .....	275

<b>I</b>	
ICTB2 .....	169
IDB0, IDB1 .....	169
IFSR .....	113
IFSR2A, IFSR3A .....	114
INT0IC to INT7IC .....	106
INVC0 .....	167
INVC1 .....	168

<b>K</b>	
KUPIC .....	105

<b>O</b>	
OFS1 .....	120, 269
ONSF .....	140

<b>P</b>	
P0 to P10 .....	257
PCLKR .....	79
PCR .....	114, 259
PD0 to PD10 .....	256
PLC0 .....	80
PM0 .....	48
PM1 .....	49
PM2 .....	79
PRCR .....	98
PRG2C .....	50
PUR0, PUR1 .....	258
PUR2 .....	259

<b>R</b>	
RMAD0 to RMAD3 .....	117
RSTFR .....	46

<b>S</b>	
S0RIC to S2RIC, S5RIC to S7RIC .....	105
S0TIC to S2TIC, S5TIC to S7TIC .....	105
S34C2 .....	225
S3BRG, S4BRG .....	224
S3C, S4C .....	224
S3IC, S4IC .....	106
S3TRR, S4TRR .....	224
SAR0 to SAR3 .....	128

**T**

TA0 to TA4 .....	138
TA0IC to TA4IC .....	105
TA0MR to TA4MR ..	138, 143, 145, 150, 152
TA1, TA2 .....	170
TA11 .....	170
TA1MR, TA2MR .....	172
TA21 .....	170
TA2MR to TA4MR .....	147
TA4 .....	170
TA41 .....	170
TA4MR .....	172
TABSR .....	139, 156, 171
TACS0, TACS1 .....	141
TACS2 .....	142
TAPOFS .....	142
TB0 to TB5 .....	155
TB0IC to TB5IC .....	105
TB0MR to TB5MR .....	155, 159, 161, 163
TB2 .....	171
TB2MR .....	172
TB2SC .....	170
TBCS0 to TBCS3 .....	157
TBSR .....	156
TCR0 to TCR3 .....	128
TRGSR .....	140, 171

**U**

U0BCNIC, U1BCNIC, U5BCNIC to U7BCNIC .....	105
U0BRG to U2BRG .....	180
U0C0 to U2C0 .....	181
U0C1 to U2C1 .....	182
U0MR to U2MR .....	180
U0RB to U2RB .....	179
U0SMR to U2SMR .....	183
U0SMR2 to U2SMR2 .....	184
U0SMR3 to U2SMR3 .....	184
U0SMR4 to U2SMR4 .....	185
U0TB to U2TB .....	179
U1BCNIC .....	106
U5BCNIC to U7BCNIC .....	106
U5BRG to U7BRG .....	180
U5C0 to U7C0 .....	181
U5C1 to U7C1 .....	182
U5MR to U7MR .....	180
U5RB to U7RB .....	179
U5SMR to U7SMR .....	183
U5SMR2 to U7SMR2 .....	184
U5SMR3 to U7SMR3 .....	184
U5SMR4 to U7SMR4 .....	185
U5TB to U7TB .....	179
UCON .....	183
UDF .....	139

**V**

VCR1 .....	38
VCR2 .....	38
VW0C .....	40

**W**

WDTS .....	119
------------	-----

<b>Revision History</b>	<b>M16C/64 Group Hardware Manual</b>
-------------------------	--------------------------------------

Rev.	Date	Description	
		Page	Summary
0.51	Jun 06, 2007	-	First Edition issued.
0.61	Jun 22, 2007	3	Table 1.2 Specifications (2) is partly revised.
		4	Table 1.3 Product List is partly revised.
0.62	Jul 04, 2007	17	3. Memory (including the figure) is partly revised.
		32	Figure 5.1 Example Reset Circuit is partly revised.
		51	7.3 Internal Memory is partly revised.
		75	Figure 10.1 System Clock Generation Circuit is partly revised.
		163	Figure 15.24 TBiMR Register in Pulse Period and Pulse Width Measurement Mode is partly revised.
		186	Table 17.1 Clock Synchronous Serial I/O Mode Specifications is partly revised.
		220	Figure 17.33 Transmit and Receive Timing in SIM Mode is partly revised.
		224	Figure 17.38 Registers S3C, S4C, S3BRG, S4BRG, S3TRR, and S4TRR is partly revised.
		256	Table 19.2 One-Shot Mode Specifications is partly revised.
		260	Table 19.4 Single Sweep Mode Specifications is partly revised.
		267	Figure 19.9 Analog Input Pin and External Sensor Equivalent Circuit is
		264	22.1 Memory Map is partly revised.
		287	23.1.1 Boot Mode is partly revised.
		265	22.1.2 User Boot Function is added.
		290	23.2.4 Standard Serial I/O Mode Disable Function is added.
		267	22.2.2 ID Code Check Function is added.
		268	22.2.3 Forced Erase Function is added.
319	Figure 23.20 Pin Connections for Standard Serial I/O Mode (1) is partly revised.		
320	Figure 23.21 Pin Connections for Standard Serial I/O Mode (2) is partly revised.		
321	Figure 23.22 Circuit Application in Standard Serial I/O Mode 1 is partly revised.		
322	Figure 23.23 Circuit Application in Standard Serial I/O Mode 2 is partly revised.		
343	24.5 Power Control is partly revised.		
362	24.11 A/D Converter is partly revised.		
0.63	Sep 21, 2007	3	Table 1.2 Specifications (2) is partly revised.
		5	Figure 1.2 Marking Diagram of Flash Memory Version (Top View) is partly revised.
		29	Table 4.12 SFR Information (12) is partly revised.
		79	Figure 10.5 PM2 Register is partly revised.
		92	Table 10.7 Pin Status in Stop Mode is partly revised.
		114	Figure 12.12 Registers IFSR2A, IFSR3A, and PCR is partly revised.
		234	Table 18.2 One-Shot Mode Specifications is partly revised.
		238	Table 18.4 Single Sweep Mode Specifications is partly revised.
259	Figure 21.11 PCR Register is partly revised.		
263	Table 22.1 Flash Memory Version Specifications is partly revised.		

<b>Revision History</b>	<b>M16C/64 Group Hardware Manual</b>
-------------------------	--------------------------------------

Rev.	Date	Description	
		Page	Summary
		263	Table 22.2 Flash Memory Rewrite Modes Overview is partly revised.
		268	22.1.3 Standard Serial I/O Mode Disable Function is moved to 22.2.4.
		268	Table 22.8 Forced Erase Function is partly revised
		272	Figure 22.5 FMR0 Register is partly revised.
		274	Figure 22.7 FMR2 Register is partly revised.
		303	Figure 23.3 A/D Conversion Characteristics is partly revised.
		304	Table 23.5 Flash Memory Version Electrical Characteristics is partly revised.
0.64	Oct 12, 2007	3	Table 1.2 Specifications (2) is partly revised.
		11	Table 1.6 Pin Functions (1) is partly revised.
		13	Table 1.8 Pin Functions (3) is partly revised.
		32	Figure 5.1 Example Reset Circuit is partly revised.
		92	10.4.3 Stop Mode is partly revised.
		230	Table 18.1 A/D Converter Specifications is partly revised.
		231	Figure 18.1 A/D Converter Block Diagram is partly revised.
		233	Figure 18.3 Registers ADCON2 and AD0 to AD7 is partly revised.
		234	Table 18.2 One-Shot Mode Specifications is partly revised.
		235	Figure 18.4 Registers ADCON0 and ADCON1 (One-shot Mode) is partly revised.
		236	Table 18.3 Repeat Mode Specifications is partly revised.
		237	Figure 18.5 Registers ADCON0 and ADCON1 (Repeat Mode) is partly revised.
		238	Table 18.4 Single Sweep Mode Specifications is partly revised.
		239	Figure 18.6 Registers ADCON0 and ADCON1 (Single Sweep Mode) is partly revised.
		240	Table 18.5 Repeat Sweep Mode 0 Specifications is partly revised.
		241	Figure 18.7 Registers ADCON0 and ADCON1 (Repeat Sweep Mode 0) is partly revised.
		242	Table 18.6 Repeat Sweep Mode 1 Specifications is partly revised.
		243	Figure 18.8 Registers ADCON0 and ADCON1 (Repeat Sweep Mode 1) is partly revised.
		260	Table 21.1 Unassigned Pin Handling in Single-chip Mode is partly revised.
		261	Table 21.2 Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode is partly revised.
		262	Figure 21.12 Unassigned Pin Handling is partly revised.
		295	Table 22.13 Pin Functions (Flash Memory Standard Serial I/O Mode) is partly revised.
		301	Table 23.1 Absolute Maximum Ratings is partly revised.
		302	Table 23.2 Recommended Operating Conditions is partly revised.
		303	Table 23.3 A/D Conversion Characteristics is partly revised.
		307	Table 23.9 Electrical Characteristics (1) is partly revised.
		342	24.2.1 VCC1 is added.
		342	24.2.2 CNVSS is added.
		364	Figure 24.4 Use of Capacitors to Reduce Noise is partly revised.
		365	24.11 A/D Converter is partly revised.

---

M16C/64 Group Hardware Manual

Publication Date: Feb 28, 2007      Rev.0.20  
                         Oct 12, 2007      Rev.0.64

Published by:      Sales Strategic Planning Div.  
                         Renesas Technology Corp.  
                         2-6-2, Otemachi, Chiyoda-ku, Tokyo 100-0004

---

© 2007. Renesas Technology Corp., All rights reserved. Printed in Japan.

# M16C/64 GROUP Hardware Manual



Renesas Technology Corp.  
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan