

To all our customers

---

## **Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.**

---

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp.  
Customer Support Dept.  
April 1, 2003

## Description

The M30218 group of single-chip microcomputers are built using the high-performance silicon gate CMOS process using a M16C/60 Series CPU core and are packaged in a 100-pin plastic molded QFP. These single-chip microcomputers operate using sophisticated instructions featuring a high level of instruction efficiency. With 1M bytes of address space, they are capable of executing instructions at high speed. They also feature a built-in multiplier and DMAC, making them ideal for controlling musical instruments, household appliances and other high-speed processing applications.

The M30218 group includes a wide range of products with different internal memory types and sizes and various package types.

## Features

- Basic machine instructions ..... Compatible with the M16C/60 series
- Memory capacity ..... ROM / RAM (See figure memory expansion)
- Shortest instruction execution time . 100ns (f(X<sub>IN</sub>)=10MHz)
- Supply voltage ..... 4.0V to 5.5V (f(X<sub>IN</sub>)=10MHz)  
2.7V to 5.5V (f(X<sub>IN</sub>)=3.5MHz)(Note)
- Interrupts ..... 19 internal and 6 external interrupt sources, 4 software
- Multifunction 16-bit timer ..... Timer A X 5, Timer B X 3
- FLD controller ..... total 56 pins  
(high-breakdown-voltage P-channel open-drain output : 52pins)
- Serial I/O ..... 2 channels for UART or clock synchronous,  
1 channels for clock synchronous  
(max.256 bytes automatic transfer function)
- DMAC ..... 2 channels (triggers: 15 sources)
- A-D converter ..... 10 bits X 8 channels
- D-A converter ..... 8 bits X 2 channels
- CRC calculation circuit ..... 1 circuit
- Watchdog timer ..... 1 pin
- Programmable I/O ..... 48 pins
- High-breakdown-voltage output ..... 52 pins
- Clock generating circuit ..... 2 built-in clock generation circuit  
(built-in feedback resistor, and external ceramic or quartz oscillator)

Note: Only mask ROM version.

## Applications

Household appliances, office equipment, Audio etc.

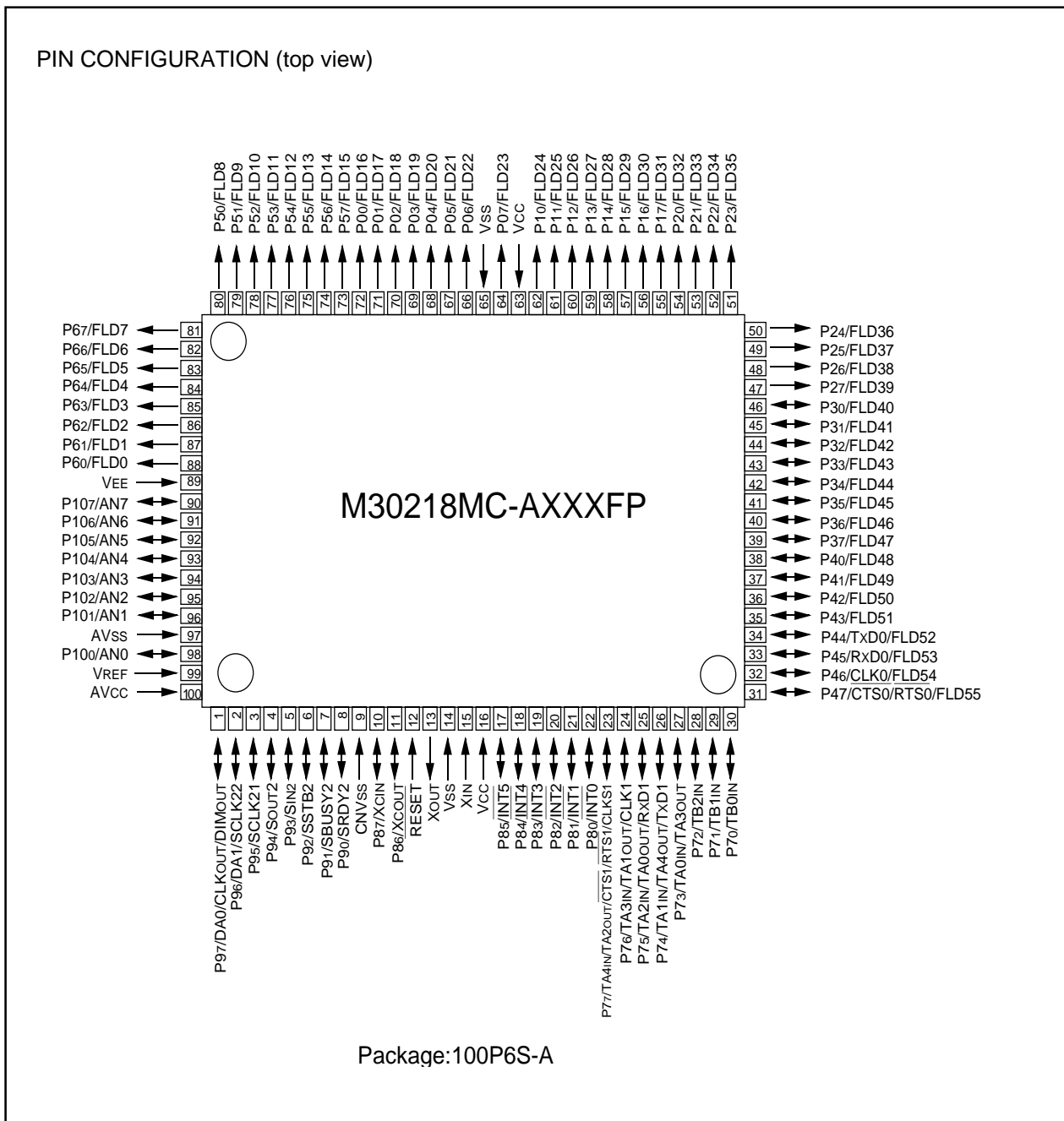
## -----Table of Contents-----

Central Processing Unit (CPU) .....	10	Timer .....	70
Reset .....	14	Serial I/O .....	87
Clock Generating Circuit .....	18	A-D Converter .....	114
Protection .....	26	D-A Converter .....	124
Interrupts .....	27	CRC Calculation Circuit .....	126
Watchdog Timer .....	45	Programmable I/O Ports .....	128
DMAC .....	47	Flash memory version .....	152
FLD controller .....	53		

Description

Pin Configuration

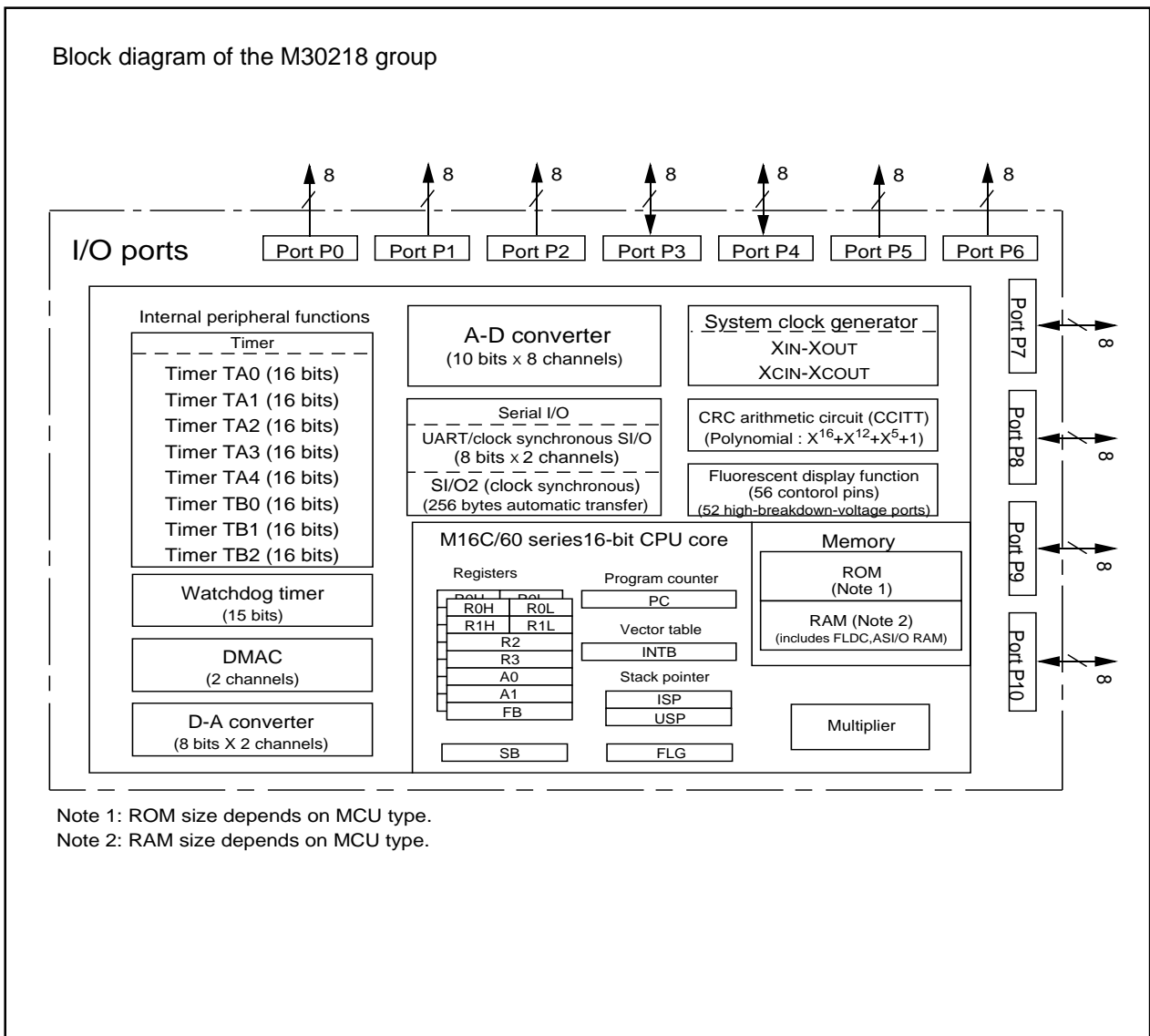
Figures AA-1 show the pin configurations (top view).



FigureAA-1. Pin configuration (top view)

**Block Diagram**

Figure AA-2 is a block diagram of the M30218 group.



**FigureAA-2. Block diagram of M30218 group**

## Description

## Performance Outline

Table AA-1 is a performance outline of M30218 group.

**Table AA-1. Performance outline of M30218 group**

Item		Performance	
Number of basic instructions		91 instructions	
Shortest instruction execution time		100ns(f(XIN)=10MHz)	
Memory capacity	ROM	See figure memory expansion	
	RAM	See figure memory expansion	
I/O port	P3, P4, P7 to P10	8 bits x 6	
Output port	P0 to P2, P5, P6	8 bit x 5	
Multifunction timer	TA0, TA1, TA2, TA3, TA4	16 bits x 5	
	TB0, TB1, TB2	16 bits x 3	
Serial I/O	UART0, UART1	(UART or clock synchronous) x 2	
	SI/O2	(Clock synchronous) x 1 (with automatic transfer function)	
Fluorescent display		56 pins	
A-D converter		10 bits x 8 channels	
D-A converter		8 bits x 2	
DMAC		2 channels (triggers :15 sources)	
CRC calculation circuit		1 circuit (polynomial: $X^{16} + X^{12} + X^5 + 1$ )	
Watchdog timer		15 bits x 1 (with prescaler)	
Interrupt		19 internal and 6 external sources, 4 software sources, 7 levels	
Clock generating circuit		2 built-in clock generation circuits (built-in feedback resistor, and external ceramic or quartz oscillator)	
Supply voltage		4.0 to 5.5V (f(XIN)=10MHz) 2.7 to 5.5V (f(XIN)=3.5MHz) (Note)	
Power consumption		18 mW (VCC=3V, f(XIN)=5MHz)	
I/O characteristics	I/O withstand voltage	VCC-48V (output ports : P0 to P2, P5, P6, I/O ports : P3, P40 to P43)	
		0 to VCC (I/O ports :P44 to P47, P7 to P10)	
	Output current	H	- 18mA (P0 to P3, P40 to P43, P5, P6) :high-breakdown-voltage, P-channel open-drain
		L	- 5mA (P44 to P47, P7 to P10) 5mA (P44 to P47, P7 to P10)
Operating ambient temperature		-20 to 85°C	
Device configuration		CMOS silicon gate	
Package		100-pin plastic mold QFP	

**Note: Only mask ROM version.**

Mitsubishi plans to release the following products in the M30218 group:

- (1) Support for mask ROM version and flash memory version
- (2) Memory capacity
- (3) Package
  - 100P6S : Plastic molded QFP (mask ROM version and flash memory version)

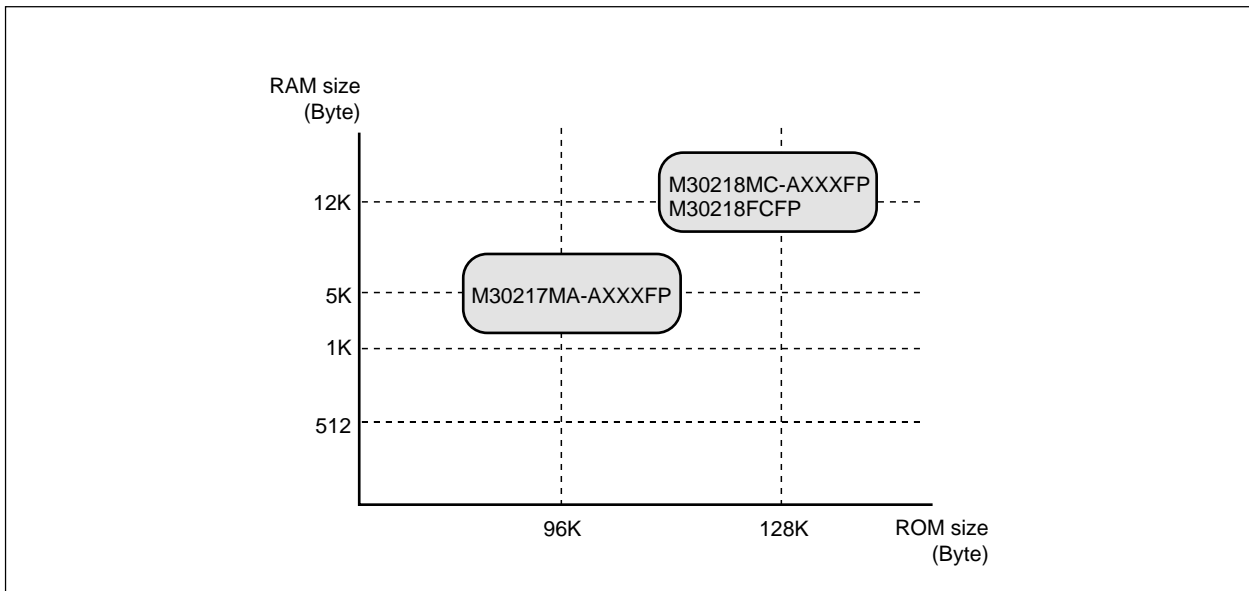


Figure AA-3. ROM expansion

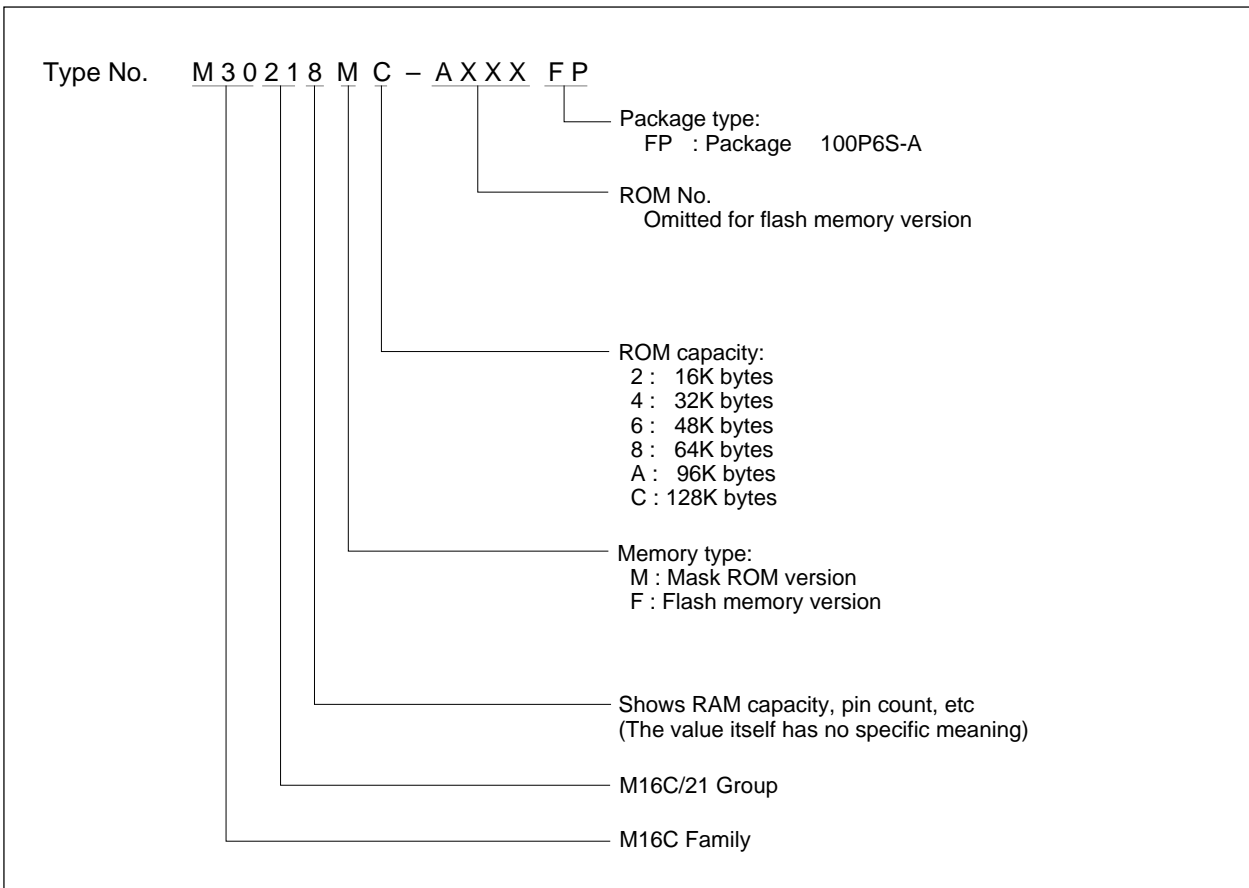


Figure AA-4. Type No., memory size, and package

**Pin Description**

Pin name	Signal name	I/O type	Function
Vcc, Vss	Power supply input		Supply 2.7V(Note1) to 5.5 V to the Vcc pin. Supply 0 V to the Vss pin. Connect a bypass capacitor across the Vcc pin and Vss pin.
CNVss	CNVss	Input	Connect it to the Vss pin.
RESET	Reset input	Input	A "L" on this input resets the microcomputer.
XIN XOUT	Clock input Clock output	Input Output	These pins are provided for the main clock generating circuit. Connect a ceramic resonator or crystal between the XIN and the XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open.
AVcc	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to Vcc.
AVss	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to Vss.
VREF	Reference voltage input	Input	This pin is a reference voltage input for the A-D converter.
VEE	pull-down power source		Apply voltage supplied to pull-down resistors of ports P0 to P1,P5,P6.
P00/FLD16 to P07/FLD23	Output port P0	Output	This is an 8-bit CMOS output port and high-breakdown-voltage P-channel open-drain output structure. A pull-down resistor is built in between port P0 and VEE pin. At reset, this port is set to VEE level. P0 function as FLD controller output pins as selected by software.
P10/FLD24 to P17/FLD31	Output port P1	Output	This is an 8-bit output port equivalent to P0. Pins in this port also function as FLD controller output pins as selected by software.
P20/FLD32 to P27/FLD39	Output port P2	Output	This is an 8-bit output port equivalent to P0. A pull-down resistor is not built in between P2 and VEE pin. Pins in this port also function as FLD controller output pins as selected by software.
P30/FLD40 to P37/FLD47	I/O port P3	Input/output	This is an 8-bit I/O port. A pull-down resistor is not built in between P3 and VEE pin. It has an input/output port direction register that allows the user to set each pin for input or output. This is low-voltage input level, and high-breakdown-voltage P-channel open-drain output structure. Pins in this port also function as FLD controller output pins as selected by software.
P40/FLD48 to P47/FLD56	I/O port P4	Input/output	This is an 8-bit I/O port equivalent to P3. This is low-voltage input level. P40 to P43 is high-breakdown-voltage P-channel open-drain output structure, P44 to P47 is CMOS output. A pull-down resistor is not built in between P4(P40 to P43) and VEE pin. Pins in this port also function as FLD controller output pins as selected by software. P44 to P47 also function as UART0 I/O pins as selected by software. When set for input, the user can specify in units of four bits by software whether or not they are tied to a pull-up resistor.
P50/FLD8 to P57/FLD15	Output port P5	Output	This is an 8-bit output port equivalent to P0. Pins in this port also function as FLD controller output pins as selected by software.
P60/FLD0 to P67/FLD7	Output port P6	Output	This is an 8-bit output port equivalent to P0. Pins in this port also function as FLD controller output pins as selected by software.

## Pin Description

## Pin Description

Pin name	Signal name	I/O type	Function
P7 <sub>0</sub> to P7 <sub>7</sub>	I/O port P7	Input/output	This is an 8-bit I/O port equivalent to P3. This is CMOS input/output. When set for input, the user can specify in units of four bits by software whether or not they are tied to a pull-up resistor. P7 <sub>0</sub> to P7 <sub>2</sub> function as TimerB0 to B2 input pins as selected by software. P7 <sub>3</sub> function as TimerA0 I/O pin as selected by software. P7 <sub>4</sub> to P7 <sub>7</sub> function as TimerA1 to A4 I/O pins, and UART1 I/O pins as selected by software.
P8 <sub>0</sub> to P8 <sub>7</sub>	I/O port P8	Input/output	This is an 8-bit I/O port equivalent to P7. When set for input, the user can specify in units of four bits by software whether or not they are tied to a pull-up resistor. P8 <sub>0</sub> to P8 <sub>5</sub> function as external interrupt input pins as selected by software. P8 <sub>6</sub> , P8 <sub>7</sub> function as sub-clock input pin as selected by software. In this case, connect a quartz oscillator between P8 <sub>6</sub> (XOUT pin) and P8 <sub>7</sub> (XCIN pin)
P9 <sub>0</sub> to P9 <sub>7</sub>	I/O port P9	Input/output	This is an 8-bit I/O port equivalent to P7. When set for input, the user can specify in units of four bits by software whether or not they are tied to a pull-up resistor. P9 <sub>7</sub> function as D-A converter output pins, clock output pins (same frequency of X <sub>IN</sub> /8, X <sub>IN</sub> /32 or XCIN) and DIM signal output pin of FLD controller as selected by software. P9 <sub>6</sub> function as D-A converter output pins and clock I/O pin of serial I/O with automatic transfer as selected by software. P9 <sub>0</sub> to P9 <sub>5</sub> function as I/O pin of serial I/O with automatic transfer as selected by software.
P10 <sub>0</sub> to P10 <sub>7</sub>	I/O port P10	Input/output	This is an 8-bit I/O port equivalent to P7. When set for input, the user can specify in units of four bits by software whether or not they are tied to a pull-up resistor. Pins in this port also function as A-D converter input pins as selected by software.

Note 1: Supply 4.0V to 5.5V to the Vcc pin in flash memory version.



## Memory

## Operation of Functional Blocks

The M30218 group accommodates certain units in a single chip. These units include ROM and RAM to store instructions and data and the central processing unit (CPU) to execute arithmetic/logic operations. Also included are peripheral units such as timers, FLD controller, serial I/O, D-A converter, DMAC, CRC calculation circuit, A-D converter, and I/O ports.

The following explains each unit.

## Memory

Figure BA-1 is a memory map of the M30218 group. The address space extends the 1M bytes from address  $00000_{16}$  to  $FFFFFF_{16}$ . From  $FFFFFF_{16}$  down is ROM. For example, in the M30218MC-AXXXFP, there is 128K bytes of internal ROM from  $E0000_{16}$  to  $FFFFFF_{16}$ . The vector table for fixed interrupts such as the reset are mapped to  $FFFDC_{16}$  to  $FFFFFF_{16}$ . The starting address of the interrupt routine is stored here. The address of the vector table for timer interrupts, etc., can be set as desired using the internal register (INTB). See the section on interrupts for details.

From  $00400_{16}$  up is RAM. For example, in the M30218MC-AXXXFP, there is 12K bytes of internal RAM from  $00400_{16}$  to  $033FF_{16}$ . In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated. (From  $00400_{16}$  to  $004FF_{16}$  is RAM for SIO2. From  $00500_{16}$  to  $005DF_{16}$  is RAM for FLD.)

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

The special page vector table is mapped to  $FFE00_{16}$  to  $FFFDB_{16}$ . If the starting addresses of subroutines or the destination addresses of jumps are stored here, subroutine call instructions and jump instructions can be used as 2-byte instructions, reducing the number of program steps.

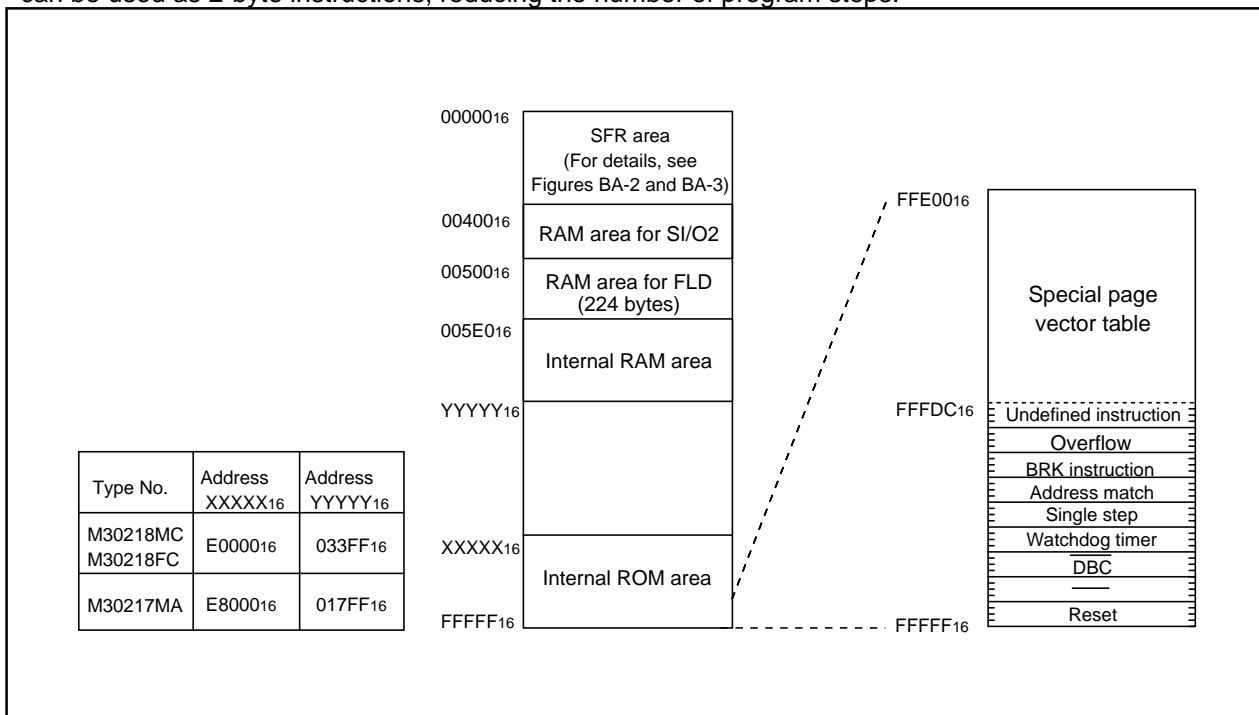


Figure BA-1. Memory map

## Memory

0000 <sub>16</sub>		0040 <sub>16</sub>	
0001 <sub>16</sub>		0041 <sub>16</sub>	
0002 <sub>16</sub>		0042 <sub>16</sub>	
0003 <sub>16</sub>		0043 <sub>16</sub>	
0004 <sub>16</sub>	Processor mode register 0 (PM0)	0044 <sub>16</sub>	
0005 <sub>16</sub>	Processor mode register 1 (PM1)	0045 <sub>16</sub>	
0006 <sub>16</sub>	System clock control register 0 (CM0)	0046 <sub>16</sub>	
0007 <sub>16</sub>	System clock control register 1 (CM1)	0047 <sub>16</sub>	INT3 interrupt control register (INT3IC)
0008 <sub>16</sub>		0048 <sub>16</sub>	INT4 interrupt control register (INT4IC)
0009 <sub>16</sub>	Address match interrupt enable register (AIER)	0049 <sub>16</sub>	INT5 interrupt control register (INT5IC)
000A <sub>16</sub>	Protect register (PRCR)	004A <sub>16</sub>	
000B <sub>16</sub>		004B <sub>16</sub>	DMA0 interrupt control register (DM0IC)
000C <sub>16</sub>		004C <sub>16</sub>	DMA1 interrupt control register (DM1IC)
000D <sub>16</sub>		004D <sub>16</sub>	
000E <sub>16</sub>	Watchdog timer start register (WDTS)	004E <sub>16</sub>	A-D conversion interrupt control register (ADIC)
000F <sub>16</sub>	Watchdog timer control register (WDC)	004F <sub>16</sub>	SI/O automatic transfer interrupt control register (ASIOIC)
0010 <sub>16</sub>		0050 <sub>16</sub>	FLD interrupt control register (FLDIC)
0011 <sub>16</sub>	Address match interrupt register 0 (RMAD0)	0051 <sub>16</sub>	UART0 transmit interrupt control register (S0TIC)
0012 <sub>16</sub>		0052 <sub>16</sub>	UART0 receive interrupt control register (S0RIC)
0013 <sub>16</sub>		0053 <sub>16</sub>	UART1 transmit interrupt control register (S1TIC)
0014 <sub>16</sub>		0054 <sub>16</sub>	UART1 receive interrupt control register (S1RIC)
0015 <sub>16</sub>	Address match interrupt register 1 (RMAD1)	0055 <sub>16</sub>	Timer A0 interrupt control register (TA0IC)
0016 <sub>16</sub>		0056 <sub>16</sub>	Timer A1 interrupt control register (TA1IC)
0017 <sub>16</sub>		0057 <sub>16</sub>	Timer A2 interrupt control register (TA2IC)
0018 <sub>16</sub>		0058 <sub>16</sub>	Timer A3 interrupt control register (TA3IC)
0019 <sub>16</sub>		0059 <sub>16</sub>	Timer A4 interrupt control register (TA4IC)
001A <sub>16</sub>		005A <sub>16</sub>	Timer B0 interrupt control register (TB0IC)
001B <sub>16</sub>		005B <sub>16</sub>	Timer B1 interrupt control register (TB1IC)
001C <sub>16</sub>		005C <sub>16</sub>	Timer B2 interrupt control register (TB2IC)
001D <sub>16</sub>		005D <sub>16</sub>	INT0 interrupt control register (INT0IC)
001E <sub>16</sub>		005E <sub>16</sub>	INT1 interrupt control register (INT1IC)
001F <sub>16</sub>		005F <sub>16</sub>	INT2 interrupt control register (INT2IC)
0020 <sub>16</sub>		0340 <sub>16</sub>	Serial I/O2 automatic transfer data pointer (SIO2DP)
0021 <sub>16</sub>	DMA0 source pointer (SAR0)	0341 <sub>16</sub>	
0022 <sub>16</sub>		0342 <sub>16</sub>	Serial I/O2 control register 1 (SIO2CON1)
0023 <sub>16</sub>		0343 <sub>16</sub>	
0024 <sub>16</sub>	DMA0 destination pointer (DAR0)	0344 <sub>16</sub>	Serial I/O2 control register 2 (SIO2CON2)
0025 <sub>16</sub>		0345 <sub>16</sub>	
0026 <sub>16</sub>		0346 <sub>16</sub>	Serial I/O2 register / transfer counter (SIO2)
0027 <sub>16</sub>		0347 <sub>16</sub>	
0028 <sub>16</sub>	DMA0 transfer counter (TCR0)	0348 <sub>16</sub>	Serial I/O2 control register 3 (SIO2CON3)
0029 <sub>16</sub>		0349 <sub>16</sub>	
002A <sub>16</sub>		034A <sub>16</sub>	
002B <sub>16</sub>		034B <sub>16</sub>	
002C <sub>16</sub>	DMA0 control register (DM0CON)	034C <sub>16</sub>	
002D <sub>16</sub>		034D <sub>16</sub>	
002E <sub>16</sub>		034E <sub>16</sub>	
002F <sub>16</sub>		034F <sub>16</sub>	
0030 <sub>16</sub>	DMA1 source pointer (SAR1)	0350 <sub>16</sub>	FLD mode register (FLDM)
0031 <sub>16</sub>		0351 <sub>16</sub>	FLD output control register (FLDCON)
0032 <sub>16</sub>		0352 <sub>16</sub>	Tdisp time set register (TDISP)
0033 <sub>16</sub>		0353 <sub>16</sub>	
0034 <sub>16</sub>	DMA1 destination pointer (DAR1)	0354 <sub>16</sub>	Toff1 time set register (TOFF1)
0035 <sub>16</sub>		0355 <sub>16</sub>	
0036 <sub>16</sub>		0356 <sub>16</sub>	Toff2 time set register (TOFF2)
0037 <sub>16</sub>		0357 <sub>16</sub>	
0038 <sub>16</sub>	DMA1 transfer counter (TCR1)	0358 <sub>16</sub>	FLD data pointer (FLDDP)
0039 <sub>16</sub>		0359 <sub>16</sub>	P2 FLD/port switch register (P2FPR)
003A <sub>16</sub>		035A <sub>16</sub>	P3 FLD/port switch register (P3FPR)
003B <sub>16</sub>		035B <sub>16</sub>	P4 FLD/port switch register (P4FPR)
003C <sub>16</sub>	DMA1 control register (DM1CON)	035C <sub>16</sub>	P5 digit output set register (P5DOR)
003D <sub>16</sub>		035D <sub>16</sub>	P6 digit output set register (P6DOR)
003E <sub>16</sub>		035E <sub>16</sub>	
003F <sub>16</sub>		035F <sub>16</sub>	

Figure BA-2. Location of peripheral unit control registers (1)

## Memory

0380 <sub>16</sub>	Count start flag (TABSR)	03C0 <sub>16</sub>	A-D register 0 (AD0)
0381 <sub>16</sub>	Clock prescaler reset flag (CPSRF)	03C1 <sub>16</sub>	
0382 <sub>16</sub>	One-shot start flag (ONSF)	03C2 <sub>16</sub>	A-D register 1 (AD1)
0383 <sub>16</sub>	Trigger select register (TRGSR)	03C3 <sub>16</sub>	
0384 <sub>16</sub>	Up-down flag (UDF)	03C4 <sub>16</sub>	A-D register 2 (AD2)
0385 <sub>16</sub>		03C5 <sub>16</sub>	
0386 <sub>16</sub>	Timer A0 (TA0)	03C6 <sub>16</sub>	A-D register 3 (AD3)
0387 <sub>16</sub>		03C7 <sub>16</sub>	
0388 <sub>16</sub>	Timer A1 (TA1)	03C8 <sub>16</sub>	A-D register 4 (AD4)
0389 <sub>16</sub>		03C9 <sub>16</sub>	
038A <sub>16</sub>	Timer A2 (TA2)	03CA <sub>16</sub>	A-D register 5 (AD5)
038B <sub>16</sub>		03CB <sub>16</sub>	
038C <sub>16</sub>	Timer A3 (TA3)	03CC <sub>16</sub>	A-D register 6 (AD6)
038D <sub>16</sub>		03CD <sub>16</sub>	
038E <sub>16</sub>	Timer A4 (TA4)	03CE <sub>16</sub>	A-D register 7 (AD7)
038F <sub>16</sub>		03CF <sub>16</sub>	
0390 <sub>16</sub>	Timer B0 (TB0)	03D0 <sub>16</sub>	
0391 <sub>16</sub>		03D1 <sub>16</sub>	
0392 <sub>16</sub>	Timer B1 (TB1)	03D2 <sub>16</sub>	
0393 <sub>16</sub>		03D3 <sub>16</sub>	
0394 <sub>16</sub>	Timer B2 (TB2)	03D4 <sub>16</sub>	A-D control register 2 (ADCON2)
0395 <sub>16</sub>		03D5 <sub>16</sub>	
0396 <sub>16</sub>	Timer A0 mode register (TA0MR)	03D6 <sub>16</sub>	A-D control register 0 (ADCON0)
0397 <sub>16</sub>	Timer A1 mode register (TA1MR)	03D7 <sub>16</sub>	A-D control register 1 (ADCON1)
0398 <sub>16</sub>	Timer A2 mode register (TA2MR)	03D8 <sub>16</sub>	D-A register 0 (DA0)
0399 <sub>16</sub>	Timer A3 mode register (TA3MR)	03D9 <sub>16</sub>	
039A <sub>16</sub>	Timer A4 mode register (TA4MR)	03DA <sub>16</sub>	D-A register 1 (DA1)
039B <sub>16</sub>	Timer B0 mode register (TB0MR)	03DB <sub>16</sub>	
039C <sub>16</sub>	Timer B1 mode register (TB1MR)	03DC <sub>16</sub>	D-A control register (DACON)
039D <sub>16</sub>	Timer B2 mode register (TB2MR)	03DD <sub>16</sub>	
039E <sub>16</sub>		03DE <sub>16</sub>	
039F <sub>16</sub>		03DF <sub>16</sub>	
03A0 <sub>16</sub>	UART0 transmit/receive mode register (U0MR)	03E0 <sub>16</sub>	Port P0 (P0)
03A1 <sub>16</sub>	UART0 bit rate generator (U0BRG)	03E1 <sub>16</sub>	Port P1 (P1)
03A2 <sub>16</sub>		03E2 <sub>16</sub>	
03A3 <sub>16</sub>	UART0 transmit buffer register (U0TB)	03E3 <sub>16</sub>	
03A4 <sub>16</sub>	UART0 transmit/receive control register 0 (U0C0)	03E4 <sub>16</sub>	Port P2 (P2)
03A5 <sub>16</sub>	UART0 transmit/receive control register 1 (U0C1)	03E5 <sub>16</sub>	Port P3 (P3)
03A6 <sub>16</sub>		03E6 <sub>16</sub>	
03A7 <sub>16</sub>	UART0 receive buffer register (U0RB)	03E7 <sub>16</sub>	Port P3 direction register (PD3)
03A8 <sub>16</sub>	UART1 transmit/receive mode register (U1MR)	03E8 <sub>16</sub>	Port P4 (P4)
03A9 <sub>16</sub>	UART1 bit rate generator (U1BRG)	03E9 <sub>16</sub>	Port P5 (P5)
03AA <sub>16</sub>		03EA <sub>16</sub>	Port P4 direction register (PD4)
03AB <sub>16</sub>	UART1 transmit buffer register (U1TB)	03EB <sub>16</sub>	
03AC <sub>16</sub>	UART1 transmit/receive control register 0 (U1C0)	03EC <sub>16</sub>	Port P6 (P6)
03AD <sub>16</sub>	UART1 transmit/receive control register 1 (U1C1)	03ED <sub>16</sub>	Port P7 (P7)
03AE <sub>16</sub>		03EE <sub>16</sub>	
03AF <sub>16</sub>	UART1 receive buffer register (U1RB)	03EF <sub>16</sub>	Port P7 direction register (PD7)
03B0 <sub>16</sub>	UART transmit/receive control register 2 (UCON)	03F0 <sub>16</sub>	Port P8 (P8)
03B1 <sub>16</sub>		03F1 <sub>16</sub>	Port P9 (P9)
03B2 <sub>16</sub>		03F2 <sub>16</sub>	Port P8 direction register (PD8)
03B3 <sub>16</sub>		03F3 <sub>16</sub>	Port P9 direction register (PD9)
03B4 <sub>16</sub>	Flash memory control register 0 (FCON0) (Note)	03F4 <sub>16</sub>	Port P10 (P10)
03B5 <sub>16</sub>	Flash memory control register 1 (FCON1) (Note)	03F5 <sub>16</sub>	
03B6 <sub>16</sub>	Flash command register (FCMD) (Note)	03F6 <sub>16</sub>	Port P10 direction register (PD10)
03B7 <sub>16</sub>		03F7 <sub>16</sub>	
03B8 <sub>16</sub>	DMA0 request cause select register (DM0SL)	03F8 <sub>16</sub>	
03B9 <sub>16</sub>		03F9 <sub>16</sub>	
03BA <sub>16</sub>	DMA1 request cause select register (DM1SL)	03FA <sub>16</sub>	
03BB <sub>16</sub>		03FB <sub>16</sub>	
03BC <sub>16</sub>		03FC <sub>16</sub>	
03BD <sub>16</sub>	CRC data register (CRCD)	03FD <sub>16</sub>	Pull-up control register 0 (PUR0)
03BE <sub>16</sub>	CRC input register (CRCIN)	03FE <sub>16</sub>	Pull-up control register 1 (PUR1)
03BF <sub>16</sub>		03FF <sub>16</sub>	

Note: This register is only exist in flash memory version.

Figure BA-3. Location of peripheral unit control registers (2)

## Central Processing Unit (CPU)

The CPU has a total of 13 registers shown in Figure CA-1. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these have two register banks.

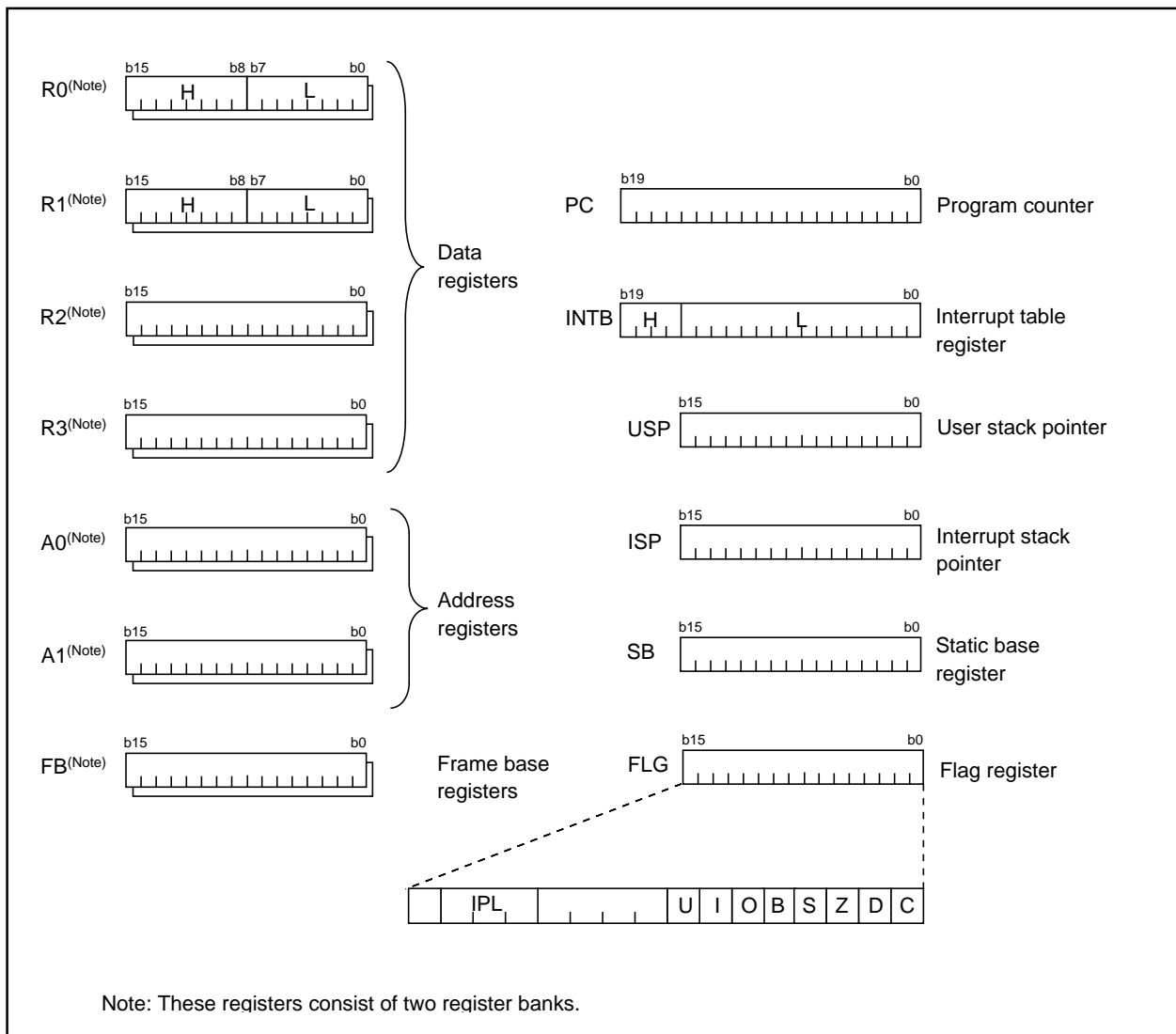


Figure CA-1. Central processing unit register

### (1) Data registers (R0, R0H, R0L, R1, R1H, R1L, R2, and R3)

Data registers (R0, R1, R2, and R3) are configured with 16 bits, and are used primarily for transfer and arithmetic/logic operations.

Registers R0 and R1 each can be used as separate 8-bit data registers, high-order bits as (R0H, R1H), and low-order bits as (R0L, R1L). In some instructions, registers R2 and R0, as well as R3 and R1 can use as 32-bit data registers (R2R0, R3R1).

### (2) Address registers (A0 and A1)

Address registers (A0 and A1) are configured with 16 bits, and have functions equivalent to those of data registers. These registers can also be used for address register indirect addressing and address register relative addressing.

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

**(3) Frame base register (FB)**

Frame base register (FB) is configured with 16 bits, and is used for FB relative addressing.

**(4) Program counter (PC)**

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

**(5) Interrupt table register (INTB)**

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

**(6) Stack pointer (USP/ISP)**

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

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

This flag is located at the position of bit 7 in the flag register (FLG).

**(7) Static base register (SB)**

Static base register (SB) is configured with 16 bits, and is used for SB relative addressing.

**(8) Flag register (FLG)**

Flag register (FLG) is configured with 11 bits, each bit is used as a flag. Figure CA-2 shows the flag register (FLG). The following explains the function of each flag:

**• Bit 0: Carry flag (C flag)**

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

**• Bit 1: Debug flag (D flag)**

This flag enables a single-step interrupt.

When this flag is "1", a single-step interrupt is generated after instruction execution. This flag is cleared to "0" when the interrupt is acknowledged.

**• Bit 2: Zero flag (Z flag)**

This flag is set to "1" when an arithmetic operation resulted in 0; otherwise, cleared to "0".

**• Bit 3: Sign flag (S flag)**

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

**• Bit 4: Register bank select flag (B flag)**

This flag chooses a register bank. Register bank 0 is selected when this flag is "0"; register bank 1 is selected when this flag is "1".

**• Bit 5: Overflow flag (O flag)**

This flag is set to "1" when an arithmetic operation resulted in overflow; otherwise, cleared to "0".

**• Bit 6: Interrupt enable flag (I flag)**

This flag enables a maskable interrupt.

An interrupt is disabled when this flag is "0", and is enabled when this flag is "1". This flag is cleared to "0" when the interrupt is acknowledged.

- **Bit 7: Stack pointer select flag (U flag)**

Interrupt stack pointer (ISP) is selected when this flag is “0” ; user stack pointer (USP) is selected when this flag is “1”.

This flag is cleared to “0” when a hardware interrupt is acknowledged or an INT instruction of software interrupt Nos. 0 to 31 is executed.

- **Bits 8 to 11: Reserved area**

- **Bits 12 to 14: Processor interrupt priority level (IPL)**

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

If a requested interrupt has priority greater than the processor interrupt priority level (IPL), the interrupt is enabled.

- **Bit 15: Reserved area**

The C, Z, S, and O flags are changed when instructions are executed. See the software manual for details.

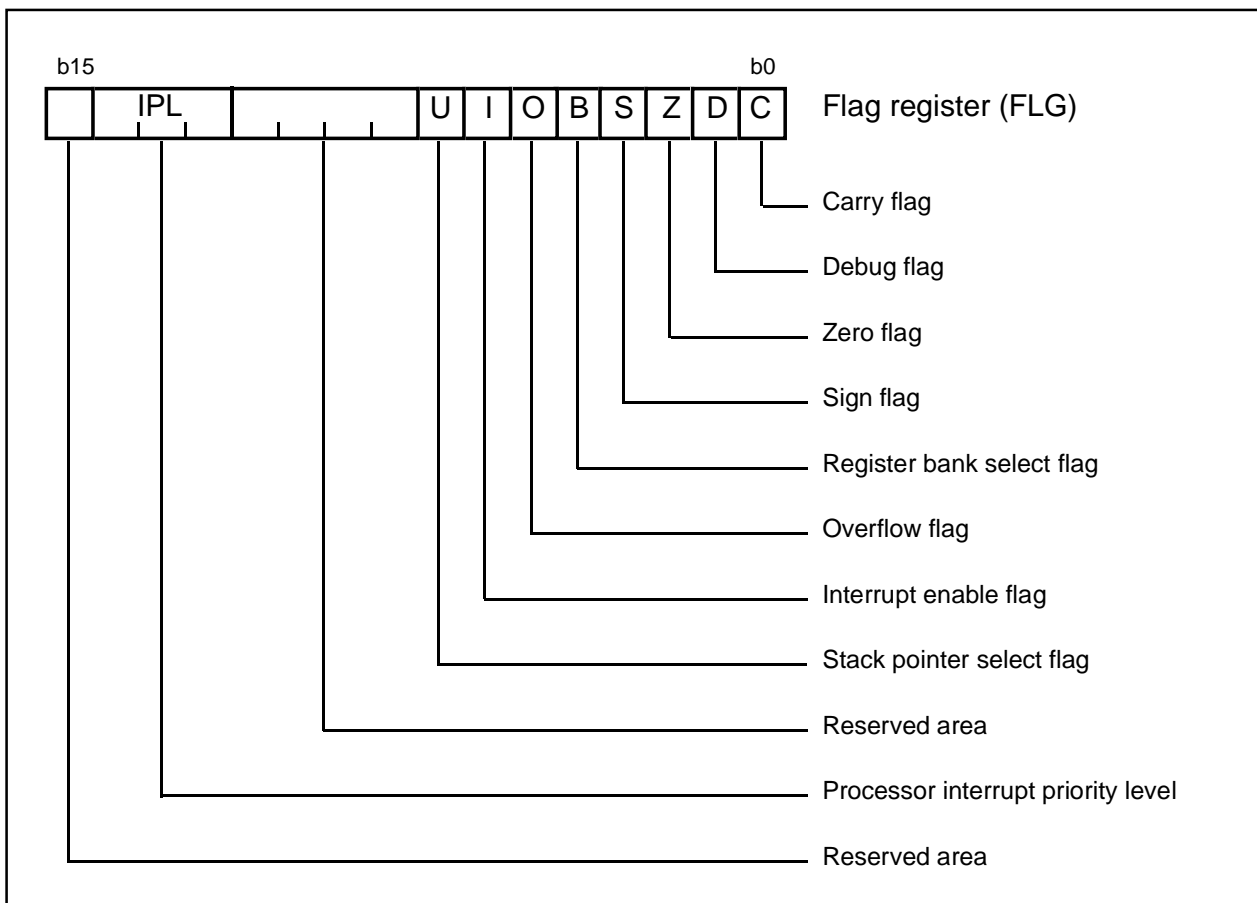


Figure CA-2. Flag register (FLG)

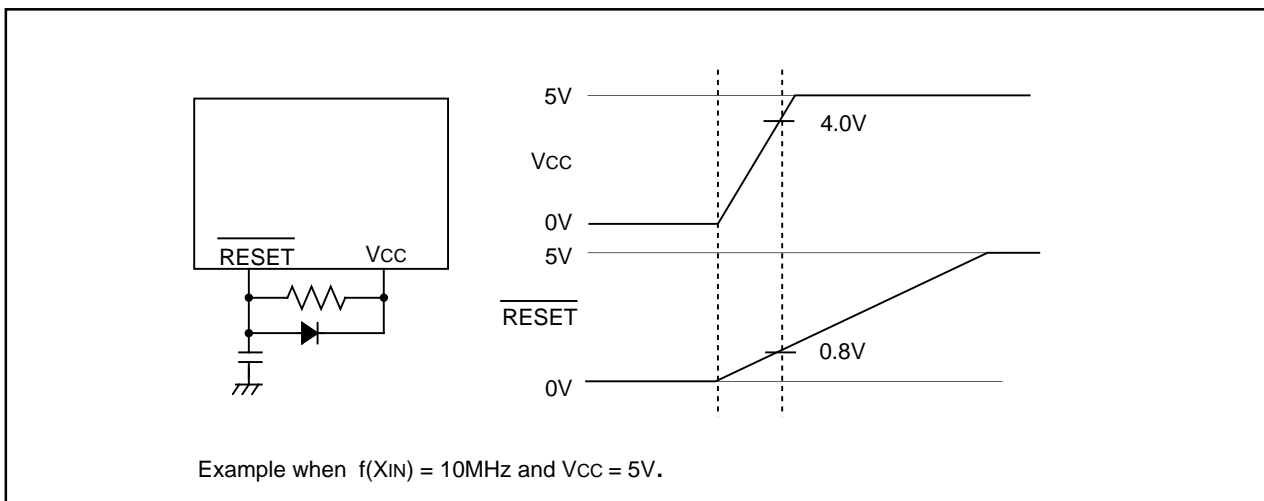
# Reset

## Reset

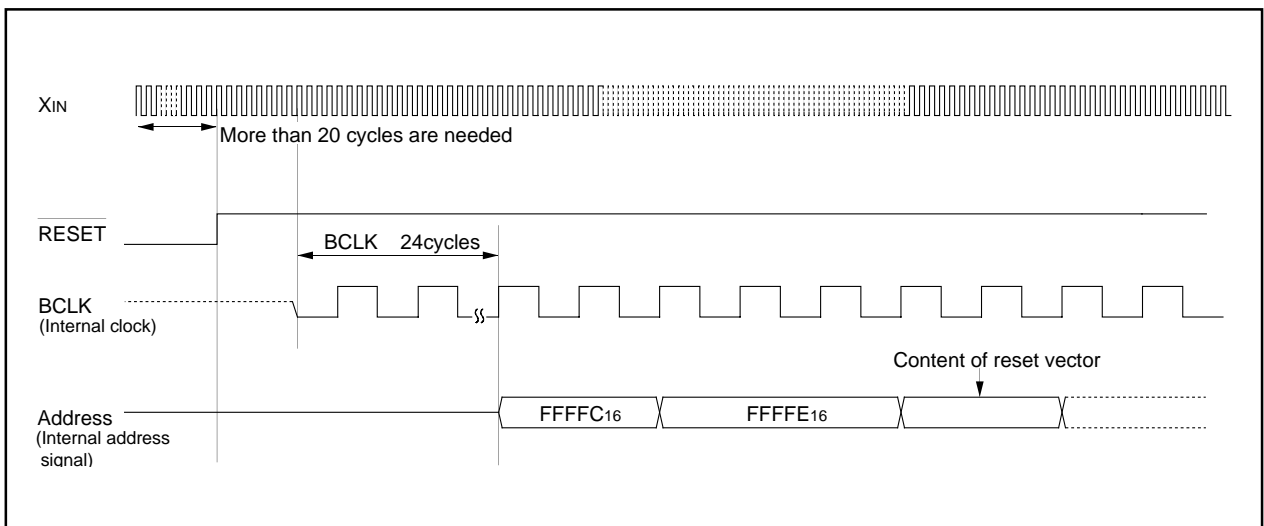
There are two kinds of resets; hardware and software. In both cases, operation is the same after the reset. (See "Software Reset" for details of software resets.) This section explains on hardware resets.

When the supply voltage is in the range where operation is guaranteed, a reset is effected by holding the reset pin level "L" (0.2V<sub>CC</sub> max.) for at least 20 cycles. When the reset pin level is then returned to the "H" level while main clock is stable, the reset status is cancelled and program execution resumes from the address in the reset vector table.

Figure DA-1 shows the example reset circuit. Figure DA-2 shows the reset sequence.



**Figure DA-1. Example reset circuit**



**Figure DA-2. Reset sequence**

## Reset

(1) Processor mode register 0	(0004 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & 0 & 0 & 0 & 0 \\ \hline \end{array}$	(24) Timer A0 interrupt control register	(0055 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$
(2) Processor mode register 1	(0005 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline 0 & \times & \times & \times & \times & \times & 0 & 0 \\ \hline \end{array}$	(25) Timer A1 interrupt control register	(0056 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$
(3) System clock control register 0	(0006 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline \end{array}$	(26) Timer A2 interrupt control register	(0057 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$
(4) System clock control register 1	(0007 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$	(27) Timer A3 interrupt control register	(0058 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$
(5) Address match interrupt enable register	(0009 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & \times & \times & 0 & 0 \\ \hline \end{array}$	(28) Timer A4 interrupt control register	(0059 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$
(6) Protect register	(000A <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & \times & 0 & 0 & 0 \\ \hline \end{array}$	(29) Timer B0 interrupt control register	(005A <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$
(7) Watchdog timer control register	(000F <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline 0 & 0 & 0 & ? & ? & ? & ? & ? \\ \hline \end{array}$	(30) Timer B1 interrupt control register	(005B <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$
(8) Address match interrupt register 0	(0010 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$	(31) Timer B2 interrupt control register	(005C <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$
	(0011 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$	(32) INT0 interrupt control register	(005D <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & 0 & 0 & ? & 0 & 0 & 0 \\ \hline \end{array}$
	(0012 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & 0 & 0 & 0 & 0 \\ \hline \end{array}$	(33) INT1 interrupt control register	(005E <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & 0 & 0 & ? & 0 & 0 & 0 \\ \hline \end{array}$
(9) Address match interrupt register 1	(0014 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$	(34) INT2 interrupt control register	(005F <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & 0 & 0 & ? & 0 & 0 & 0 \\ \hline \end{array}$
	(0015 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$	(35) Serial I/O 2 control register 1	(0342 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$
	(0016 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & 0 & 0 & 0 & 0 \\ \hline \end{array}$	(36) Serial I/O 2 control register 2	(0344 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$
(10) DMA0 control register	(002C <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline 0 & 0 & 0 & 0 & 0 & ? & 0 & 0 \\ \hline \end{array}$	(37) Serial I/O 2 control register 3	(0348 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$
(11) DMA1 control register	(003C <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline 0 & 0 & 0 & 0 & 0 & ? & 0 & 0 \\ \hline \end{array}$	(38) FLDC mode register	(0350 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$
(12) INT3 interrupt control register	(0044 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & 0 & 0 & ? & 0 & 0 & 0 \\ \hline \end{array}$	(39) FLD output control register	(0351 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$
(13) INT4 interrupt control register	(0048 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & 0 & 0 & ? & 0 & 0 & 0 \\ \hline \end{array}$	(40) Tdisp time set register	(0352 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$
(14) INT5 interrupt control register	(0049 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & 0 & 0 & ? & 0 & 0 & 0 \\ \hline \end{array}$	(41) Toff1 time set register	(0354 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{FF}_{16} \\ \hline \end{array}$
(15) DMA0 interrupt control register	(004B <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$	(42) Toff2 time set register	(0356 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{FF}_{16} \\ \hline \end{array}$
(16) DMA1 interrupt control register	(004C <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$	(43) P2 FLD/port switch register	(0359 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$
(17) A-D conversion interrupt control register	(004E <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$	(44) P3 FLD/port switch register	(035A <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$
(18) SI/O automatic transfer interrupt control register	(004F <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$	(45) P4 FLD/port switch register	(035B <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$
(19) FLD interrupt control register	(0050 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$	(46) P5 digit output set register	(035C <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$
(20) UART0 transmit interrupt control register	(0051 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$	(47) P6 digit output set register	(035D <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \text{00}_{16} \\ \hline \end{array}$
(21) UART0 receive interrupt control register	(0052 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$			
(22) UART1 transmit interrupt control register	(0053 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$			
(23) UART1 receive interrupt control register	(0054 <sub>16</sub> )...	$\begin{array}{ c c c c c c c c } \hline \times & \times & \times & \times & ? & 0 & 0 & 0 \\ \hline \end{array}$			

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

x : Nothing is mapped to this bit  
? : Undefined

Figure DA-3. Device's internal status after a reset is cleared



(48) Count start flag	(0380 <sub>16</sub> )...	00 <sub>16</sub>	(77) Port P3 direction register	(03E7 <sub>16</sub> )...	00 <sub>16</sub>
(49) Clock prescaler reset flag	(0381 <sub>16</sub> )...	0XXXXXX	(78) Port P4 direction register	(03EA <sub>16</sub> )...	00 <sub>16</sub>
(50) One-shot start flag	(0382 <sub>16</sub> )...	00XX0000	(79) Port P7 direction register	(03EF <sub>16</sub> )...	00 <sub>16</sub>
(51) Trigger select flag	(0383 <sub>16</sub> )...	00 <sub>16</sub>	(80) Port P8 direction register	(03F2 <sub>16</sub> )...	00 <sub>16</sub>
(52) Up-down flag	(0384 <sub>16</sub> )...	00 <sub>16</sub>	(81) Port P9 direction register	(03F3 <sub>16</sub> )...	00 <sub>16</sub>
(53) Timer A0 mode register	(0396 <sub>16</sub> )...	00 <sub>16</sub>	(82) Port P10 direction register	(03F6 <sub>16</sub> )...	00 <sub>16</sub>
(54) Timer A1 mode register	(0397 <sub>16</sub> )...	00 <sub>16</sub>	(83) Pull-up control register 0	(03FD <sub>16</sub> )...	00 <sub>16</sub>
(55) Timer A2 mode register	(0398 <sub>16</sub> )...	00 <sub>16</sub>	(84) Pull-up control register 1	(03FE <sub>16</sub> )...	00 <sub>16</sub>
(56) Timer A3 mode register	(0399 <sub>16</sub> )...	00 <sub>16</sub>	(85) Data registers (R0/R1/R2/R3)		0000 <sub>16</sub>
(57) Timer A4 mode register	(039A <sub>16</sub> )...	00 <sub>16</sub>	(86) Address registers (A0/A1)		0000 <sub>16</sub>
(58) Timer B0 mode register	(039B <sub>16</sub> )...	00?XX000	(87) Frame base register (FB)		0000 <sub>16</sub>
(59) Timer B1 mode register	(039C <sub>16</sub> )...	00?XX000	(88) Interrupt table register (INTB)		00000 <sub>16</sub>
(60) Timer B2 mode register	(039D <sub>16</sub> )...	00?XX000	(89) User stack pointer (USP)		0000 <sub>16</sub>
(61) UART0 transmit/receive mode register	(03A0 <sub>16</sub> )...	00 <sub>16</sub>	(90) Interrupt stack pointer (ISP)		0000 <sub>16</sub>
(62) UART0 transmit/receive control register 0	(03A4 <sub>16</sub> )...	000001000	(91) Static base register (SB)		0000 <sub>16</sub>
(63) UART0 transmit/receive control register 1	(03A5 <sub>16</sub> )...	000000010	(92) Flag register (FLG)		0000 <sub>16</sub>
(64) UART1 transmit/receive mode register	(03A8 <sub>16</sub> )...	00 <sub>16</sub>			
(65) UART1 transmit/receive control register 0	(03AC <sub>16</sub> )...	000001000			
(66) UART1 transmit/receive control register 1	(03AD <sub>16</sub> )...	000000010			
(67) UART transmit/receive control register 2	(03B0 <sub>16</sub> )...	XXXXXX0			
(68) Flash memory control register 0 (Note)	(03B4 <sub>16</sub> )...	00100000			
(69) Flash memory control register 1 (Note)	(03B5 <sub>16</sub> )...	XXXXXXXX0			
(70) Flash command register (Note)	(03B6 <sub>16</sub> )...	00 <sub>16</sub>			
(71) DMA0 cause select register	(03B8 <sub>16</sub> )...	00 <sub>16</sub>			
(72) DMA1 cause select register	(03BA <sub>16</sub> )...	00 <sub>16</sub>			
(73) A-D control register 2	(03D4 <sub>16</sub> )...	XXXXXXXX0			
(74) A-D control register 0	(03D6 <sub>16</sub> )...	00000???			
(75) A-D control register 1	(03D7 <sub>16</sub> )...	00 <sub>16</sub>			
(76) D-A control register	(03DC <sub>16</sub> )...	00 <sub>16</sub>			

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

x : Nothing is mapped to this bit  
? : Undefined

Note: This register is only exist in flash memory version.

Figure DA-4. Device's internal status after a reset is cleared

## Software Reset

Writing "1" to bit 3 of the processor mode register 0 (address 0004<sub>16</sub>) applies a (software reset) reset to the microcomputer. A software reset has almost the same effect as a hardware reset. The contents of internal RAM are preserved.

Figure DA-5 shows the processor mode register 0 and 1.

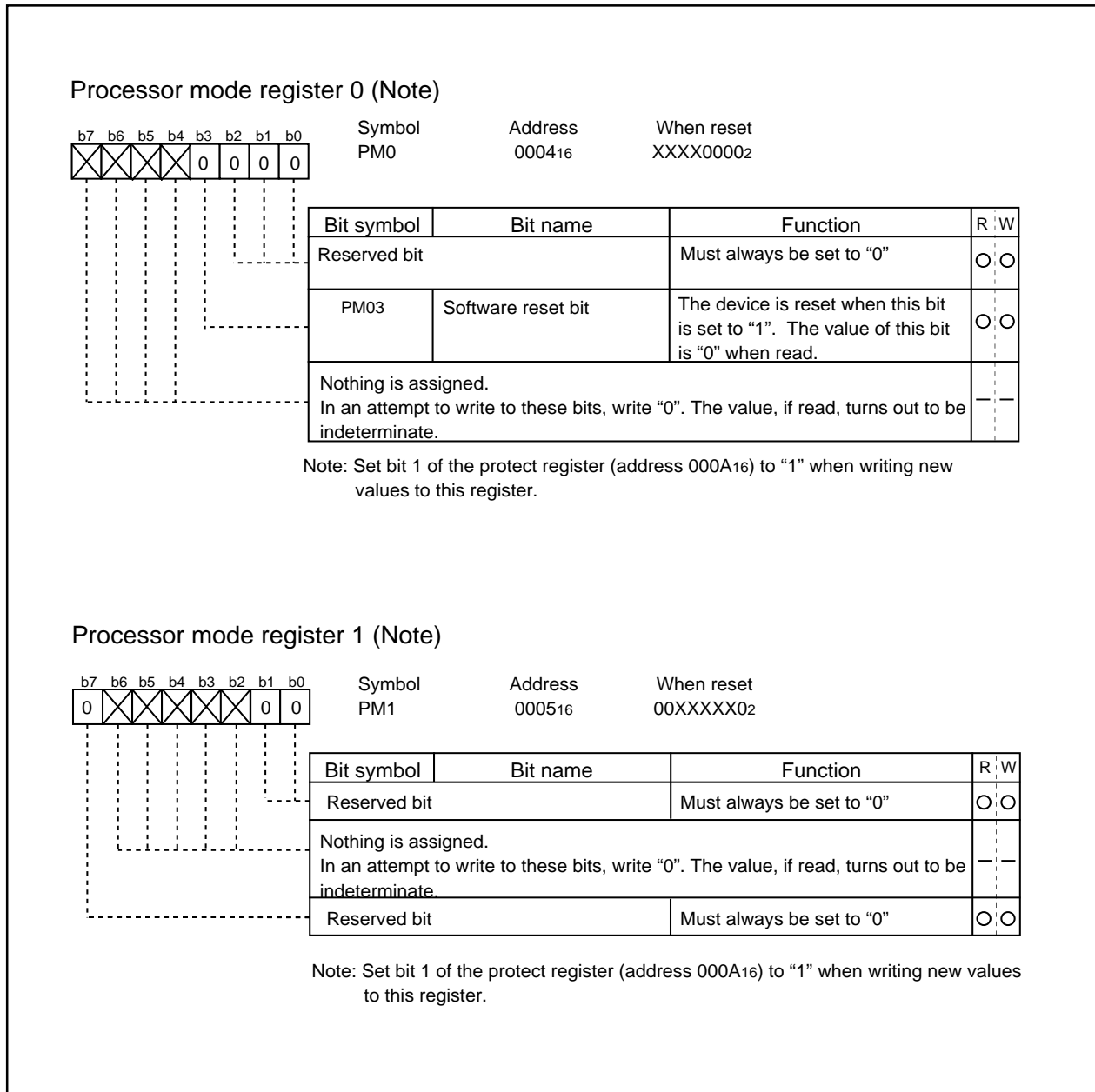


Figure DA-5. Processor mode register 0 and 1.

## Clock Generating Circuit

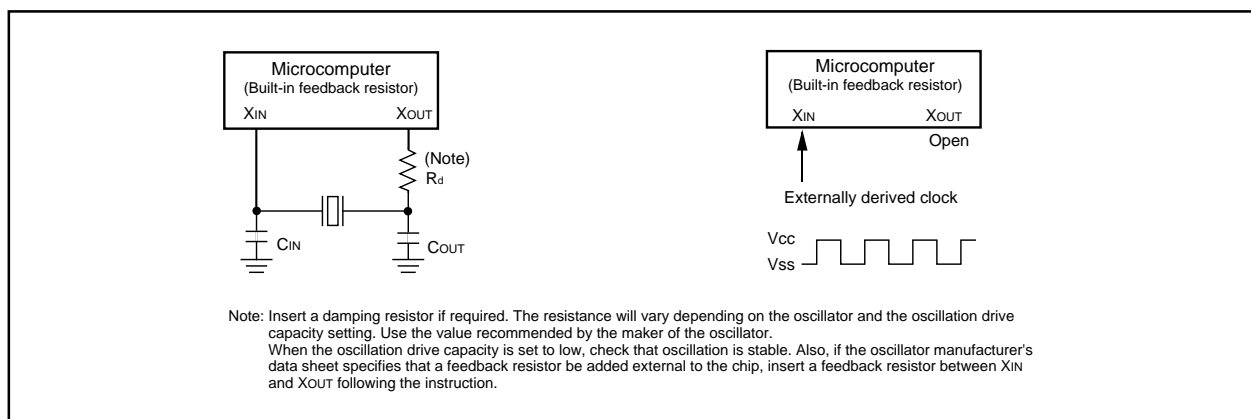
The clock generating circuit contains two oscillator circuits that supply the operating clock sources to the CPU and internal peripheral units.

**Table WA-1. Main clock and sub clock generating circuits**

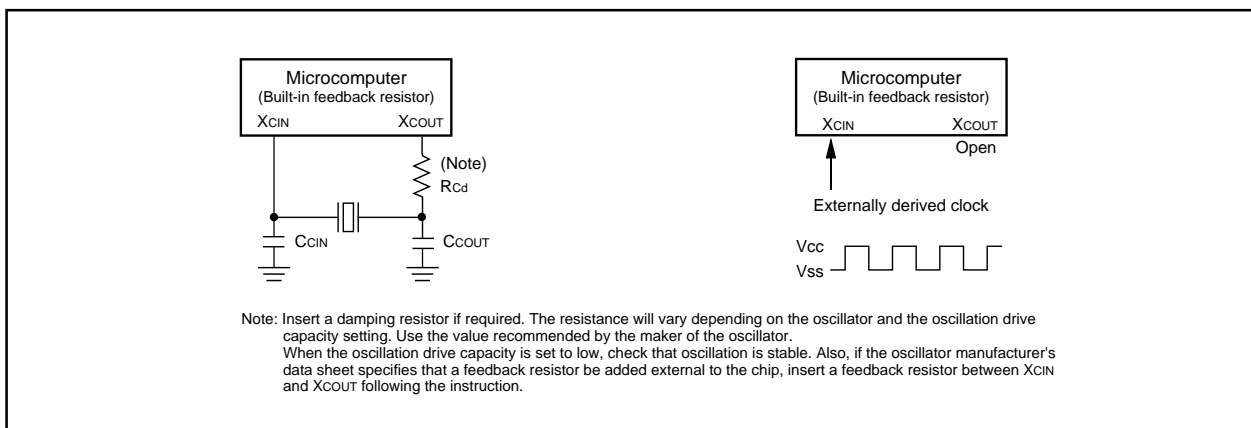
	Main clock generating circuit	Sub clock generating circuit
Use of clock	<ul style="list-style-type: none"> <li>• CPU's operating clock source</li> <li>• Internal peripheral units' operating clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU's operating clock source</li> <li>• Timer A/B's count clock source</li> </ul>
Usable oscillator	Ceramic or crystal oscillator	Crystal oscillator
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT
Oscillation stop/restart function	Available	Available
Oscillator status immediately after reset	Oscillating	Stopped
Other	Externally derived clock can be input	

## Example of oscillator circuit

Figure WA-1 shows some examples of the main clock circuit, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Figure WA-2 shows some examples of sub clock circuits, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Circuit constants in Figures WA-1 and WA-2 vary with each oscillator used. Use the values recommended by the manufacturer of your oscillator.



**Figure WA-1. Examples of main clock**



**Figure WA-2. Examples of sub clock**

**Clock Control**

Figure WA-3 shows the block diagram of the clock generating circuit.

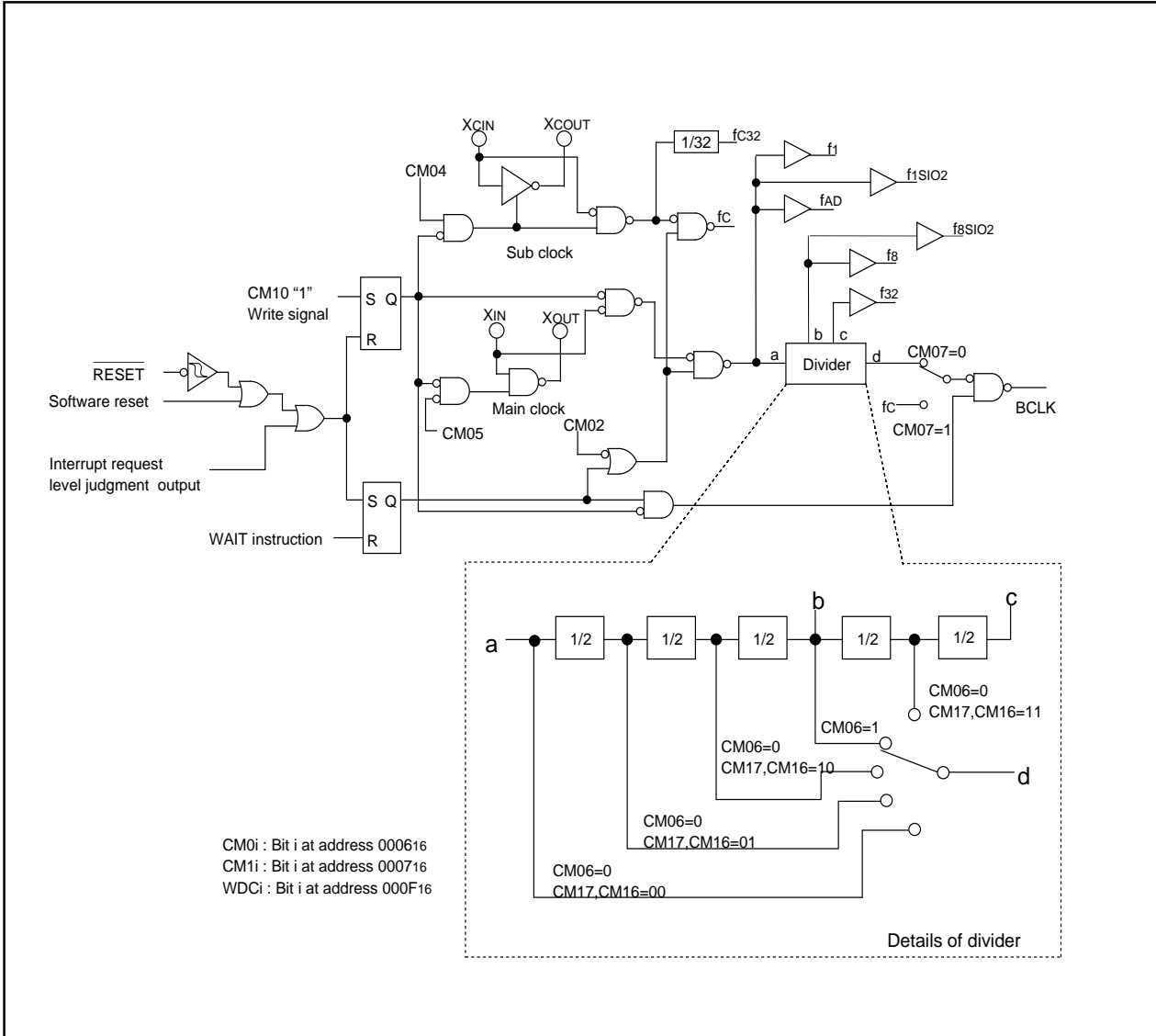


Figure WA-3. Clock generating circuit

The following paragraphs describes the clocks generated by the clock generating circuit.

### (1) Main clock

The main clock is generated by the main clock oscillation circuit. After a reset, the clock is divided by 8 to the BCLK. The clock can be stopped using the main clock stop bit (bit 5 at address 0006<sub>16</sub>). Stopping the clock, after switching the operating clock source of CPU to the sub-clock, reduces the power dissipation. After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the main clock oscillation circuit can be reduced using the XIN-XOUT drive capacity select bit (bit 5 at address 0007<sub>16</sub>). Reducing the drive capacity of the main clock oscillation circuit reduces the power dissipation. This bit changes to "1" when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

### (2) Sub-clock

The sub-clock is generated by the sub-clock oscillation circuit. No sub-clock is generated after a reset. After oscillation is started using the port Xc select bit (bit 4 at address 0006<sub>16</sub>), the sub-clock can be selected as the BCLK by using the system clock select bit (bit 7 at address 0006<sub>16</sub>). However, be sure that the sub-clock oscillation has fully stabilized before switching.

After the oscillation of the sub-clock oscillation circuit has stabilized, the drive capacity of the sub-clock oscillation circuit can be reduced using the XCIN-XCOUT drive capacity select bit (bit 3 at address 0006<sub>16</sub>). Reducing the drive capacity of the sub-clock oscillation circuit reduces the power dissipation. This bit changes to "1" when shifting to stop mode and at a reset.

### (3) BCLK

The BCLK is the clock that drives the CPU, and is  $f_c$  or the clock is derived by dividing the main clock by 1, 2, 4, 8, or 16. The BCLK is derived by dividing the main clock by 8 after a reset. The BCLK signal can be output from BCLK pin by the BCLK output disable bit (bit 7 at address 0004<sub>16</sub>) in the memory expansion and the microprocessor modes.

The main clock division select bit 0 (bit 6 at address 0006<sub>16</sub>) changes to "1" when shifting from high-speed/medium-speed to stop mode and at reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

### (4) Peripheral function clock( $f_1$ , $f_8$ , $f_{32}$ , $f_{AD}$ , $f_{SIO2}$ , $f_{SIO2}$ )

The clock for the peripheral devices is derived from the main clock or by dividing it by 1, 8, or 32. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at 0006<sub>16</sub>) to "1" and then executing a WAIT instruction.

### (5) $f_{c32}$

This clock is derived by dividing the sub-clock by 32. It is used for the timer A and timer B counts.

### (6) $f_c$

This clock has the same frequency as the sub-clock. It is used for the BCLK and for the watchdog timer.

Figure WA-4 shows the system clock control registers 0 and 1.

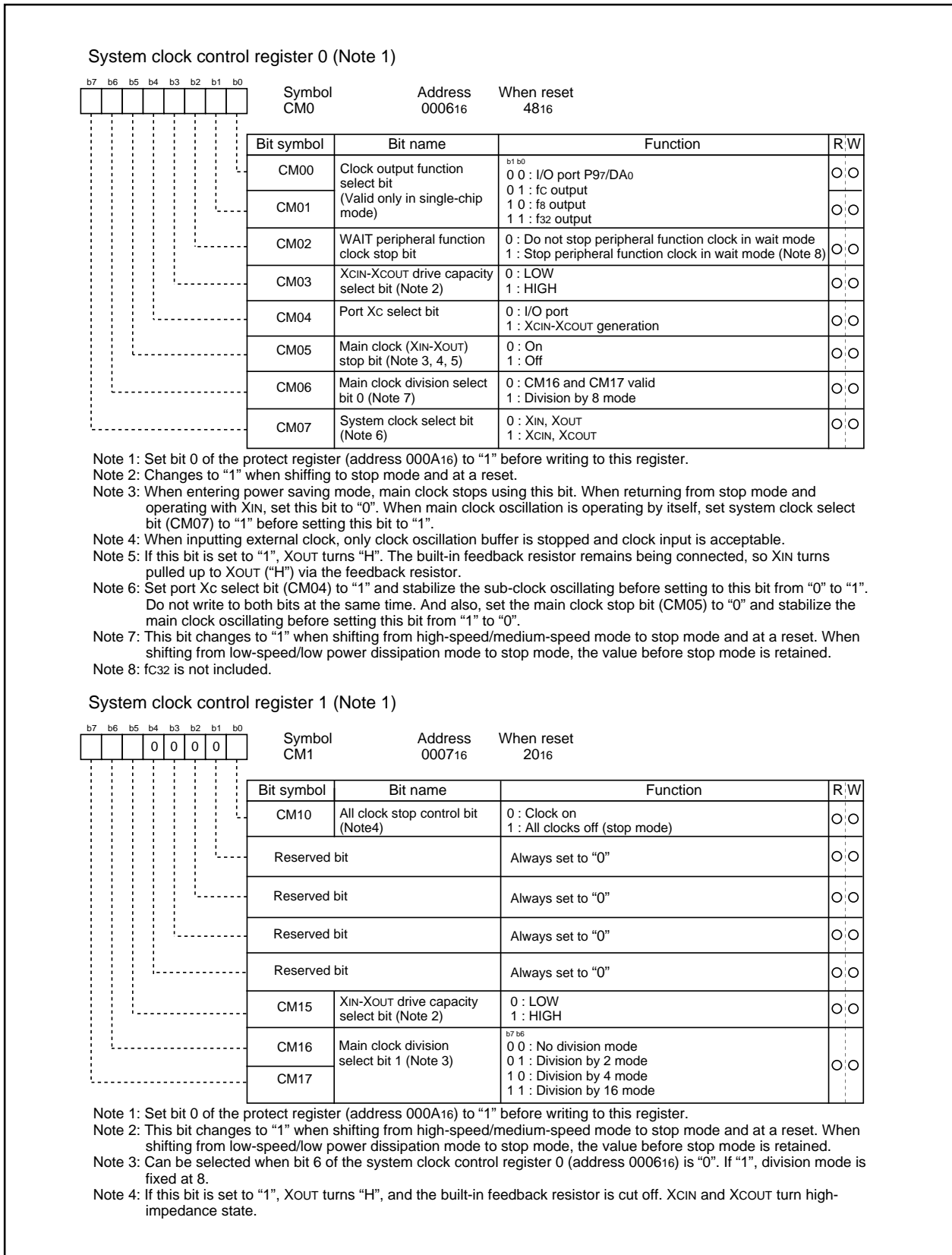


Figure WA-4. Clock control registers 0 and 1

## Clock output

**Clock Output**

The clock output function select bit (bit 0,1 at address 0006<sub>16</sub>) allows you to choose the clock from f<sub>8</sub>, f<sub>32</sub>, or f<sub>c</sub> to be output from the P97/DA<sub>0</sub>/CLKOUT/DIMOUT pin. When the WAIT peripheral function clock stop bit (bit 2 at address 0006<sub>16</sub>) is set to "1", the output of f<sub>8</sub> and f<sub>32</sub> stop by executing of WAIT instruction.

**Stop Mode**

Writing "1" to the all-clock stop control bit (bit 0 at address 0007<sub>16</sub>) stops all oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that VCC remains above 2V.

Because the oscillation of BCLK, f<sub>1</sub> to f<sub>32</sub>, f<sub>c</sub>, f<sub>c32</sub>, and f<sub>AD</sub> stops in stop mode, peripheral functions such as the fluorescent display function, serial I/O 2, A-D converter and watchdog timer do not function. However, timer A and timer B operate provided that the event counter mode is set to an external pulse, and UART0 and UART2 functions provided an external clock is selected. Table WA-2 shows the status of the ports in stop mode.

Stop mode is cancelled by a hardware reset or an interrupt. If an interrupt is to be used to cancel stop mode, that interrupt must first have been enabled. If returning by an interrupt, that interrupt routine is executed.

When shifting from high-speed/medium-speed mode to stop mode and at a reset, the main clock division select bit 0 (bit 6 at address 0006<sub>16</sub>) is set to "1". When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

**Table WA-2. Port status during stop mode**

Pin		States
Port		Retains status before stop mode
CLKOUT	When f <sub>c</sub> selected	"H"
	When f <sub>8</sub> , f <sub>32</sub> selected	Retains status before stop mode

**Wait Mode**

When a WAIT instruction is executed, BCLK stops and the microcomputer enters the wait mode. In this mode, oscillation continues but BCLK and watchdog timer stop. Writing "1" to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. Table WA-3 shows the status of the ports in wait mode.

Wait mode is cancelled by a hardware reset or an interrupt. If an interrupt is used to cancel wait mode, the microcomputer restarts from the interrupt routine using as BCLK, the clock that had been selected when the WAIT instruction was executed.

**Table WA-3. Port status during wait mode**

Pin		States
Port		Retains status before wait mode
CLKOUT	When f <sub>c</sub> selected	Does not stop
	When f <sub>8</sub> , f <sub>32</sub> selected	Does not stop when the WAIT peripheral function clock stop bit is "0". (Note) When the WAIT peripheral function clock stop bit is "1", the status immediately prior to entering wait mode is maintained.

Note: Attention that reducing the power dissipation is impossible.

## Status Transition of BCLK

### Status Transition Of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. Table WA-4 shows the operating modes corresponding to the settings of system clock control registers 0 and 1.

When reset, the device starts in division by 8 mode. The main clock division select bit 0(bit 6 at address 000616) changes to "1" when shifting from high-speed/medium-speed to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained. The following shows the operational modes of BCLK.

#### (1) Division by 2 mode

The main clock is divided by 2 to obtain the BCLK.

#### (2) Division by 4 mode

The main clock is divided by 4 to obtain the BCLK.

#### (3) Division by 8 mode

The main clock is divided by 8 to obtain the BCLK. When reset, the device starts operating from this mode. Before the user can go from this mode to no division mode, division by 2 mode, or division by 4 mode, the main clock must be oscillating stably. When going to low-speed or lower power consumption mode, make sure the sub-clock is oscillating stably.

#### (4) Division by 16 mode

The main clock is divided by 16 to obtain the BCLK.

#### (5) No-division mode

The main clock is divided by 1 to obtain the BCLK.

#### (6) Low-speed mode

fc is used as the BCLK. Note that oscillation of both the main and sub-clocks must have stabilized before transferring from this mode to another or vice versa. At least 2 to 3 seconds are required after the sub-clock starts. Therefore, the program must be written to wait until this clock has stabilized immediately after powering up and after stop mode is cancelled.

#### (7) Low power dissipation mode

fc is the BCLK and the main clock is stopped.

Note : Before the count source for BCLK can be changed from XIN to XCIN or vice versa, the clock to which the count source is going to be switched must be oscillating stably. Allow a wait time in software for the oscillation to stabilize before switching over the clock.

**Table WA-4. Operating modes dictated by settings of system clock control registers 0 and 1**

CM17	CM16	CM07	CM06	CM05	CM04	Operating mode of BCLK
0	1	0	0	0	Invalid	Division by 2 mode
1	0	0	0	0	Invalid	Division by 4 mode
Invalid	Invalid	0	1	0	Invalid	Division by 8 mode
1	1	0	0	0	Invalid	Division by 16 mode
0	0	0	0	0	Invalid	No-division mode
Invalid	Invalid	1	Invalid	0	1	Low-speed mode
Invalid	Invalid	1	Invalid	1	1	Low power dissipation mode



## Power control

The following is a description of the three available power control modes:

### Modes

Power control is available in three modes.

#### (a) Normal operation mode

- **High-speed mode**

Divide-by-1 frequency of the main clock becomes the BCLK. The CPU operates with the internal clock selected. Each peripheral function operates according to its assigned clock.

- **Medium-speed mode**

Divide-by-2, divide-by-4, divide-by-8, or divide-by-16 frequency of the main clock becomes the BCLK. The CPU operates according to the internal clock selected. Each peripheral function operates according to its assigned clock.

- **Low-speed mode**

fc becomes the BCLK. The CPU operates according to the fc clock. The fc clock is supplied by the secondary clock. Each peripheral function operates according to its assigned clock.

- **Low power consumption mode**

The main clock operating in low-speed mode is stopped. The CPU operates according to the fc clock. The fc clock is supplied by the secondary clock. The only peripheral functions that operate are those with the sub-clock selected as the count source.

#### (b) Wait mode

The CPU operation is stopped. The oscillators do not stop.

#### (c) Stop mode

All oscillators stop. The CPU and all built-in peripheral functions stop. This mode, among the three modes listed here, is the most effective in decreasing power consumption.

Figure WA-5 is the state transition diagram of the above modes.

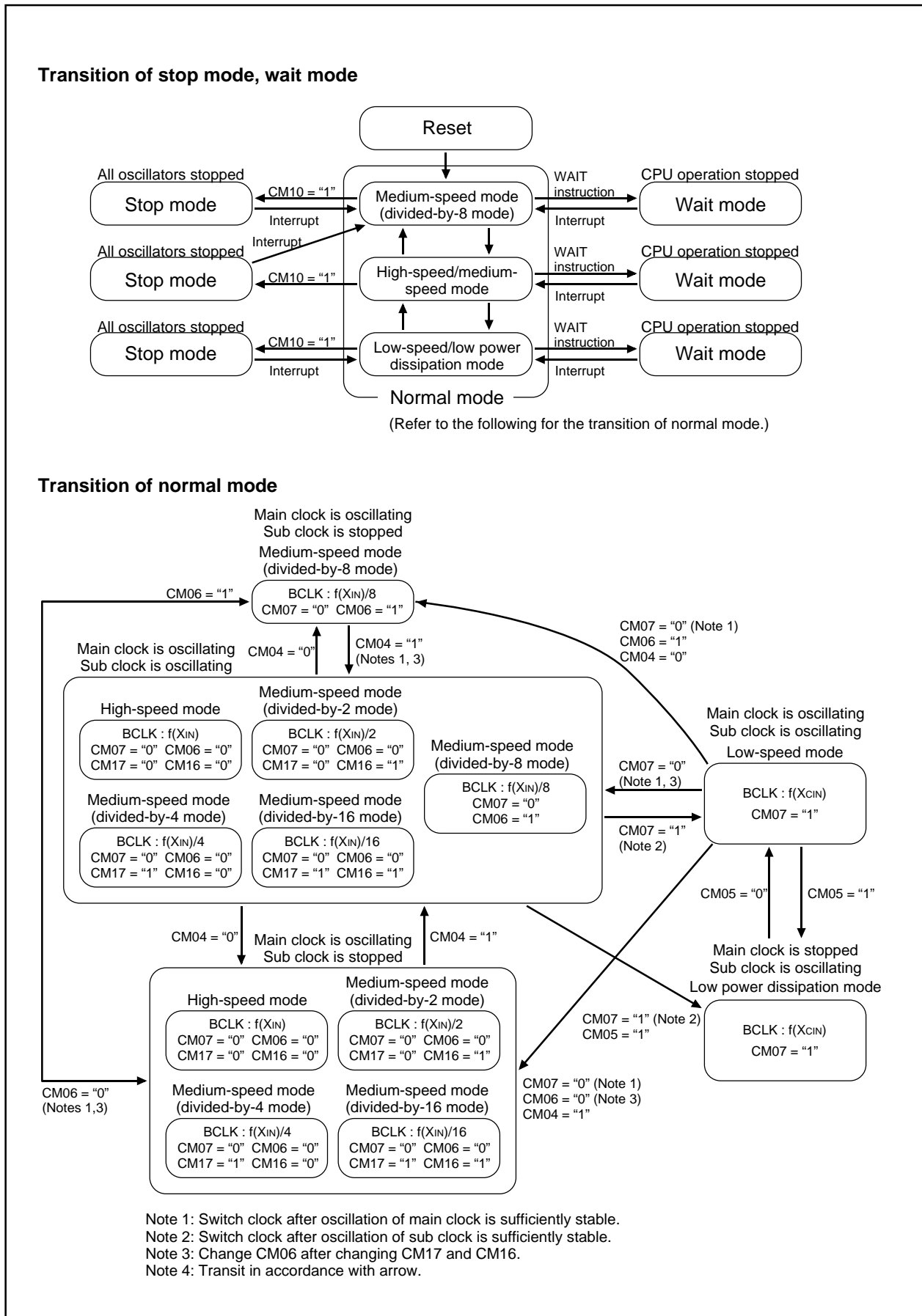


Figure WA-5. State transition diagram of Power control mode

## Protection

## Protection

The protection function is provided so that the values in important registers cannot be changed in the event that the program runs out of control. Figure WA-6 shows the protect register. The values in the processor mode register 0 (address 0004<sub>16</sub>), processor mode register 1 (address 0005<sub>16</sub>), system clock control register 0 (address 0006<sub>16</sub>), and system clock control register 1 (address 0007<sub>16</sub>) can only be changed when the respective bit in the protect register is set to "1".

The system clock control registers 0 and 1 write-enable bit (bit 0 at address 000A<sub>16</sub>) and processor mode register 0 and 1 write-enable bit (bit 1 at address 000A<sub>16</sub>) do not automatically return to "0" after a value has been written to an address. The program must therefore be written to return these bits to "0".

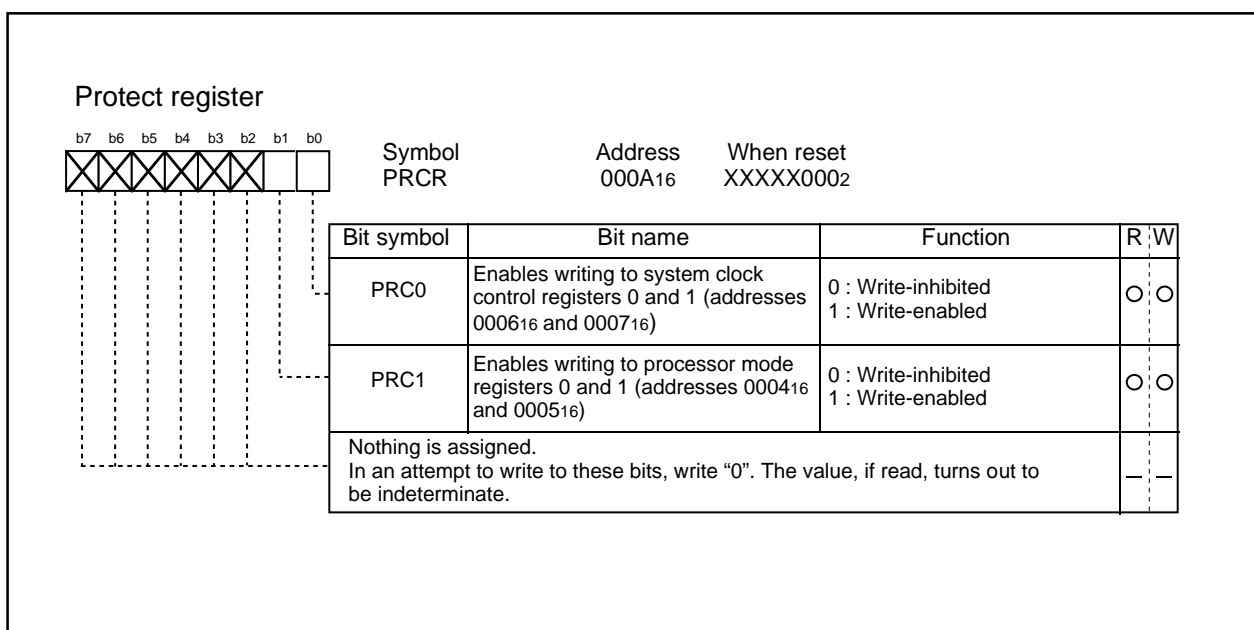


Figure WA-6. Protect register

## Overview of Interrupt

### Type of Interrupts

Figure DD-1 lists the types of interrupts.

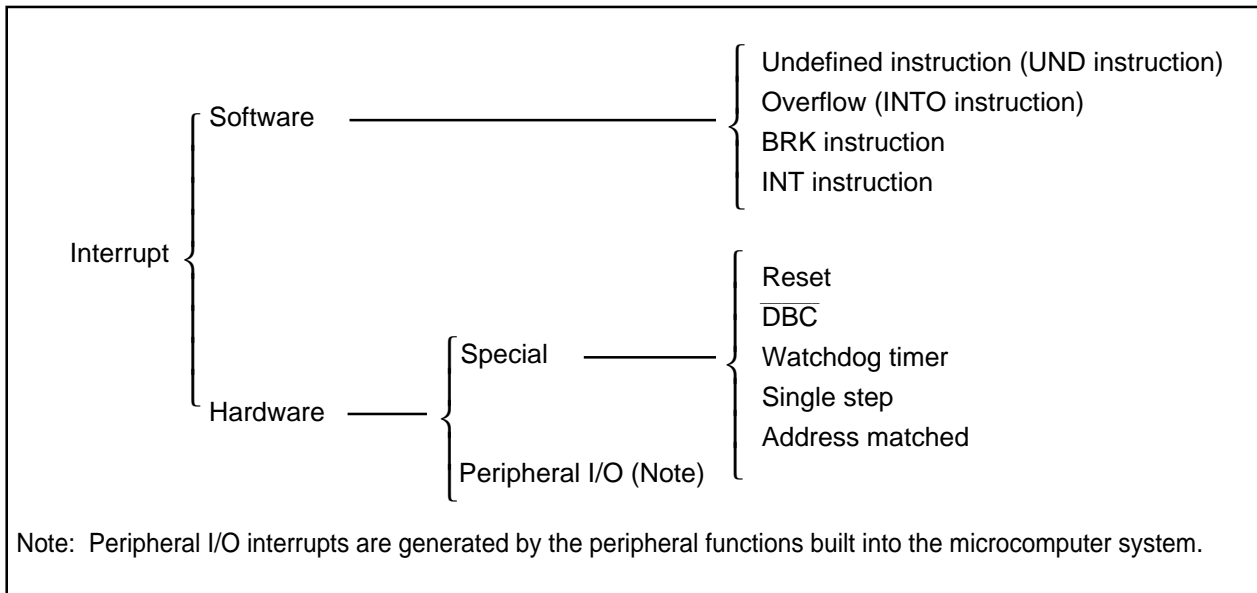


Figure DD-1. Classification of interrupts

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

## Software Interrupts

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

- **Undefined instruction interrupt**

An undefined instruction interrupt occurs when executing the UND instruction.

- **Overflow interrupt**

An overflow interrupt occurs when executing the INTO instruction with the overflow flag (O flag) set to "1". The following are instructions whose O flag changes by arithmetic:

ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB

- **BRK interrupt**

A BRK interrupt occurs when executing the BRK instruction.

- **INT interrupt**

An INT interrupt occurs when specifying one of software interrupt numbers 0 through 63 and executing the INT instruction. Software interrupt numbers 0 through 31 are assigned to peripheral I/O interrupts, so executing the INT instruction allows executing the same interrupt routine that a peripheral I/O interrupt does.

The stack pointer (SP) used for the INT interrupt is dependent on which software interrupt number is involved.

So far as software interrupt numbers 0 through 31 are concerned, the microcomputer saves the stack pointer assignment flag (U flag) when it accepts an interrupt request. If change the U flag to "0" and select the interrupt stack pointer (ISP), and then execute an interrupt sequence. When returning from the interrupt routine, the U flag is returned to the state it was before the acceptance of interrupt request. So far as software numbers 32 through 63 are concerned, the stack pointer does not make a shift.

## Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral I/O interrupts.

### (1) Special interrupts

Special interrupts are non-maskable interrupts.

- **Reset**

Reset occurs if an “L” is input to the  $\overline{\text{RESET}}$  pin.

- **$\overline{\text{DBC}}$  interrupt**

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

- **Watchdog timer interrupt**

Generated by the watchdog timer.

- **Single-step interrupt**

This interrupt is exclusively for the debugger, do not use it in other circumstances. With the debug flag (D flag) set to “1”, a single-step interrupt occurs after one instruction is executed.

- **Address match interrupt**

An address match interrupt occurs immediately before the instruction held in the address indicated by the address match interrupt register is executed with the address match interrupt enable bit set to “1”. If an address other than the first address of the instruction in the address match interrupt register is set, no address match interrupt occurs.

### (2) Peripheral I/O interrupts

A peripheral I/O interrupt is generated by one of built-in peripheral functions. Built-in peripheral functions are dependent on classes of products, so the interrupt factors too are dependent on classes of products. The interrupt vector table is the same as the one for software interrupt numbers 0 through 31 the INT instruction uses. Peripheral I/O interrupts are maskable interrupts.

- **DMA0 interrupt, DMA1 interrupt**

These are interrupts that DMA generates.

- **A-D conversion interrupt**

This is an interrupt that the A-D converter generates.

- **UART0 and UART1 transmission interrupt**

These are interrupts that the serial I/O transmission generates.

- **UART0 and UART1 reception interrupt**

These are interrupts that the serial I/O reception generates.

- **SI/O automatic transfer interrupt**

This is an interrupt that the SI/O automatic transfer generates.

- **Timer A0 interrupt through timer A4 interrupt**

These are interrupts that timer A generates

- **Timer B0 interrupt through timer B2 interrupt**

These are interrupts that timer B generates.

- **$\overline{\text{INT0}}$  interrupt through  $\overline{\text{INT5}}$  interrupt**

An  $\overline{\text{INT}}$  interrupt occurs if either a rising edge or a falling edge is input to the  $\overline{\text{INT}}$  pin.

## Interrupts and Interrupt Vector Tables

If an interrupt request is accepted, a program branches to the interrupt routine set in the interrupt vector table. Set the first address of the interrupt routine in each vector table. Figure DD-2 shows the format for specifying the address.

Two types of interrupt vector tables are available — fixed vector table in which addresses are fixed and variable vector table in which addresses can be varied by the setting.

	MSB	LSB
Vector address + 0	Low address	
Vector address + 1	Mid address	
Vector address + 2	0 0 0 0	High address
Vector address + 3	0 0 0 0	0 0 0 0

Figure DD-2. Format for specifying interrupt vector addresses

### • Fixed vector tables

The fixed vector table is a table in which addresses are fixed. The vector tables are located in an area extending from FFFDC<sub>16</sub> to FFFFF<sub>16</sub>. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table DD-1 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

Table DD-1. Interrupts assigned to the fixed vector tables and addresses of vector tables

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFD <sub>16</sub> to FFFD <sub>16</sub> F	Interrupt on UND instruction
Overflow	FFFE0 <sub>16</sub> to FFFE3 <sub>16</sub>	Interrupt on INTO instruction
BRK instruction	FFFE4 <sub>16</sub> to FFFE7 <sub>16</sub>	If the vector contains FF <sub>16</sub> , program execution starts from the address shown by the vector in the variable vector table
Address match	FFFE8 <sub>16</sub> to FFFEB <sub>16</sub>	There is an address-matching interrupt enable bit
Single step (Note)	FFFE <sub>16</sub> C to FFFEF <sub>16</sub>	Do not use
Watchdog timer	FFFF0 <sub>16</sub> to FFFF3 <sub>16</sub>	
DBC (Note)	FFFF4 <sub>16</sub> to FFFF7 <sub>16</sub>	Do not use
-	FFFF8 <sub>16</sub> to FFFFB <sub>16</sub>	-
Reset	FFFF <sub>16</sub> C to FFFFF <sub>16</sub>	

Note: Interrupts used for debugging purposes only.

## Interrupt

- Variable vector tables

The addresses in the variable vector table can be modified, according to the user's settings. Indicate the first address using the interrupt table register (INTB). The 256-byte area subsequent to the address the INTB indicates becomes the area for the variable vector tables. One vector table comprises four bytes. Set the first address of the interrupt routine in each vector table. Table DD-2 shows the interrupts assigned to the variable vector tables and addresses of vector tables.

**Table DD-2. Interrupts assigned to the variable vector tables and addresses of vector tables**

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note)	BRK instruction	Cannot be masked I flag
—		—	
Software interrupt number 7	+28 to +31 (Note)	$\overline{\text{INT3}}$	
Software interrupt number 8	+32 to +35 (Note)	$\overline{\text{INT4}}$	
Software interrupt number 9	+36 to +39 (Note)	$\overline{\text{INT5}}$	
—		—	
Software interrupt number 11	+44 to +47 (Note)	DMA0	
Software interrupt number 12	+48 to +51 (Note)	DMA1	
—		—	
Software interrupt number 14	+56 to +59 (Note)	A-D	
Software interrupt number 15	+60 to +63 (Note)	SI/O automatic transfer	
Software interrupt number 16	+64 to +67 (Note)	FLD	
Software interrupt number 17	+68 to +71 (Note)	UART0 transmit	
Software interrupt number 18	+72 to +75 (Note)	UART0 receive	
Software interrupt number 19	+76 to +79 (Note)	UART1 transmit	
Software interrupt number 20	+80 to +83 (Note)	UART1 receive	
Software interrupt number 21	+84 to +87 (Note)	Timer A0	
Software interrupt number 22	+88 to +91 (Note)	Timer A1	
Software interrupt number 23	+92 to +95 (Note)	Timer A2	
Software interrupt number 24	+96 to +99 (Note)	Timer A3	
Software interrupt number 25	+100 to +103 (Note)	Timer A4	
Software interrupt number 26	+104 to +107 (Note)	Timer B0	
Software interrupt number 27	+108 to +111 (Note)	Timer B1	
Software interrupt number 28	+112 to +115 (Note)	Timer B2	
Software interrupt number 29	+116 to +119 (Note)	$\overline{\text{INT0}}$	
Software interrupt number 30	+120 to +123 (Note)	$\overline{\text{INT1}}$	
Software interrupt number 31	+124 to +127 (Note)	$\overline{\text{INT2}}$	
Software interrupt number 32 to Software interrupt number 63	+128 to +131 (Note) to +252 to +255 (Note)	Software interrupt	Cannot be masked I flag

Note : Address relative to address in interrupt table register (INTB).



## Interrupt Control

Descriptions are given here regarding how to enable or disable maskable interrupts and how to set the priority to be accepted. What is described here does not apply to non-maskable interrupts.

Enable or disable a maskable interrupt using the interrupt enable flag (I flag), interrupt priority level selection bit, or processor interrupt priority level (IPL). Whether an interrupt request is present or absent is indicated by the interrupt request bit. The interrupt request bit and the interrupt priority level selection bit are located in the interrupt control register of each interrupt. Also, the interrupt enable flag (I flag) and the IPL are located in the flag register (FLG).

Figure DD-3 shows the memory map of the interrupt control registers.

Interrupt

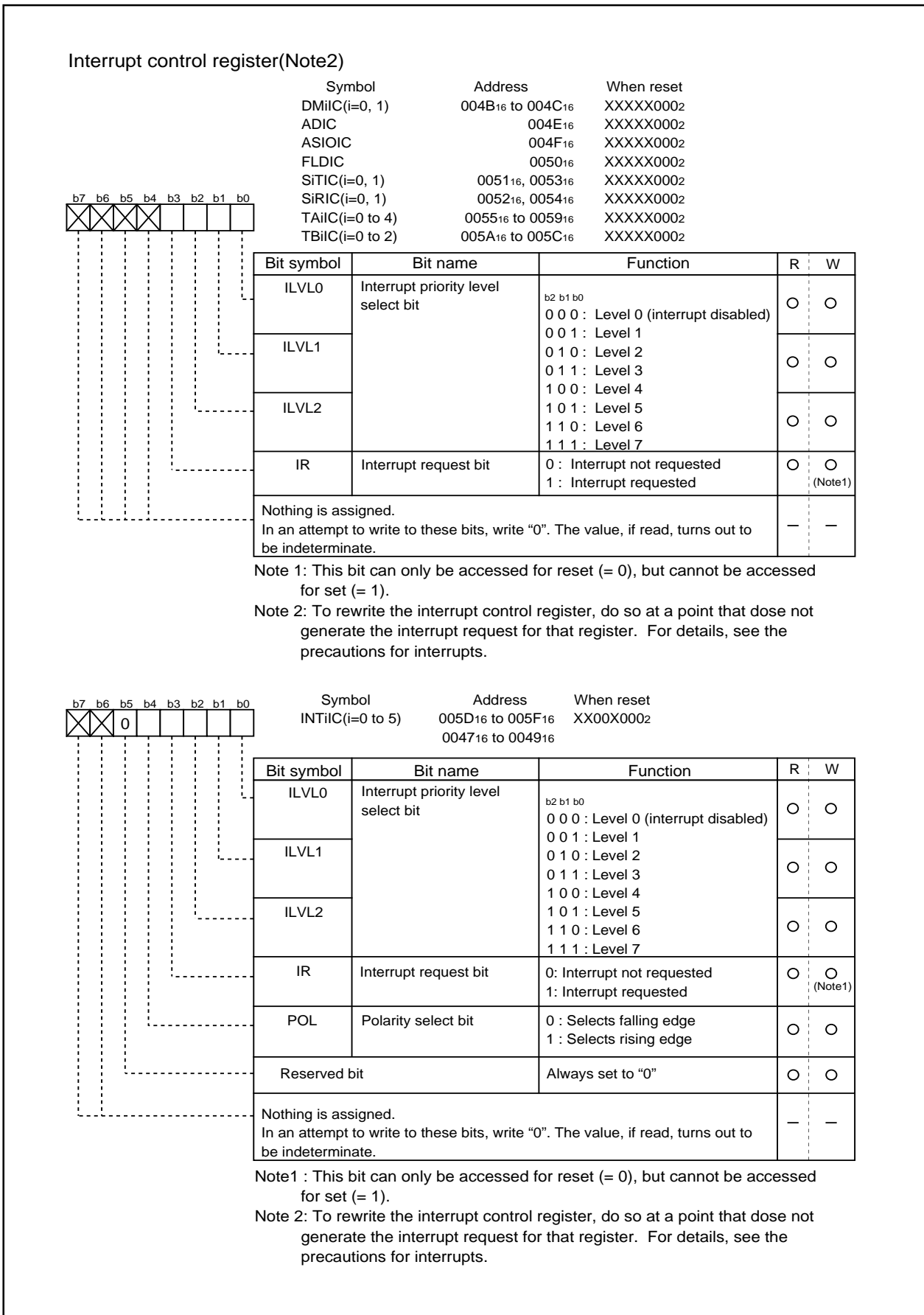


Figure DD-3. Interrupt control registers

### Interrupt Enable Flag (I flag)

The interrupt enable flag (I flag) controls the enabling and disabling of maskable interrupts. Setting this flag to "1" enables all maskable interrupts; setting it to "0" disables all maskable interrupts. This flag is set to "0" after reset.

### Interrupt Request Bit

The interrupt request bit is set to "1" by hardware when an interrupt is requested. After the interrupt is accepted and jumps to the corresponding interrupt vector, the request bit is set to "0" by hardware. The interrupt request bit can also be set to "0" by software. (Do not set this bit to "1").

### Interrupt Priority Level Select Bit and Processor Interrupt Priority Level (IPL)

Set the interrupt priority level using the interrupt priority level select bit, which is one of the component bits of the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to "0" disables the interrupt.

Table DD-3 shows the settings of interrupt priority levels and Table DD-4 shows the interrupt levels enabled, according to the consist of the IPL.

The following are conditions under which an interrupt is accepted:

- interrupt enable flag (I flag) = 1
- interrupt request bit = 1
- interrupt priority level > IPL

The interrupt enable flag (I flag), the interrupt request bit, the interrupt priority select bit, and the IPL are independent, and they are not affected by one another.

**Table DD-3. Settings of interrupt priority levels**

Interrupt priority level select bit	Interrupt priority level	Priority order
b2 b1 b0 0 0 0	Level 0 (interrupt disabled)	_____
0 0 1	Level 1	Low ↓ High
0 1 0	Level 2	
0 1 1	Level 3	
1 0 0	Level 4	
1 0 1	Level 5	
1 1 0	Level 6	
1 1 1	Level 7	

**Table DD-4. Interrupt levels enabled according to the contents of the IPL**

IPL	Enabled interrupt priority levels
IPL <sub>2</sub> IPL <sub>1</sub> IPL <sub>0</sub> 0 0 0	Interrupt levels 1 and above are enabled
0 0 1	Interrupt levels 2 and above are enabled
0 1 0	Interrupt levels 3 and above are enabled
0 1 1	Interrupt levels 4 and above are enabled
1 0 0	Interrupt levels 5 and above are enabled
1 0 1	Interrupt levels 6 and above are enabled
1 1 0	Interrupt levels 7 and above are enabled
1 1 1	All maskable interrupts are disabled

**Rewrite the interrupt control register**

To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occur, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

**Example 1:**

```
INT_SWITCH1:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  NOP
  NOP
  FSET  I           ; Enable interrupts.
```

**Example 2:**

```
INT_SWITCH2:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  MOV.W MEM, R0    ; Dummy read.
  FSET  I           ; Enable interrupts.
```

**Example 3:**

```
INT_SWITCH3:
  PUSHC FLG       ;
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  POPC  FLG       ; Enable interrupts.
```

The reason why two NOP instructions or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

## Interrupt Sequence

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

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

In the interrupt sequence, the processor carries out the following in sequence given:

- (1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 00000<sub>16</sub>.
- (2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.
- (3) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer select flag (U flag) to "0" (the U flag, however does not change if the INT instruction, in software interrupt numbers 32 through 63, is executed)
- (4) Saves the content of the temporary register (Note) within the CPU in the stack area.
- (5) Saves the content of the program counter (PC) in the stack area.
- (6) Sets the interrupt priority level of the accepted instruction in the IPL.

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

Note: This register cannot be utilized by the user.

## Interrupt Response Time

'Interrupt response time' is the period between the instant an interrupt occurs and the instant the first instruction within the interrupt routine has been executed. This time comprises the period from the occurrence of an interrupt to the completion of the instruction under execution at that moment (a) and the time required for executing the interrupt sequence (b). Figure DD-4 shows the interrupt response time.

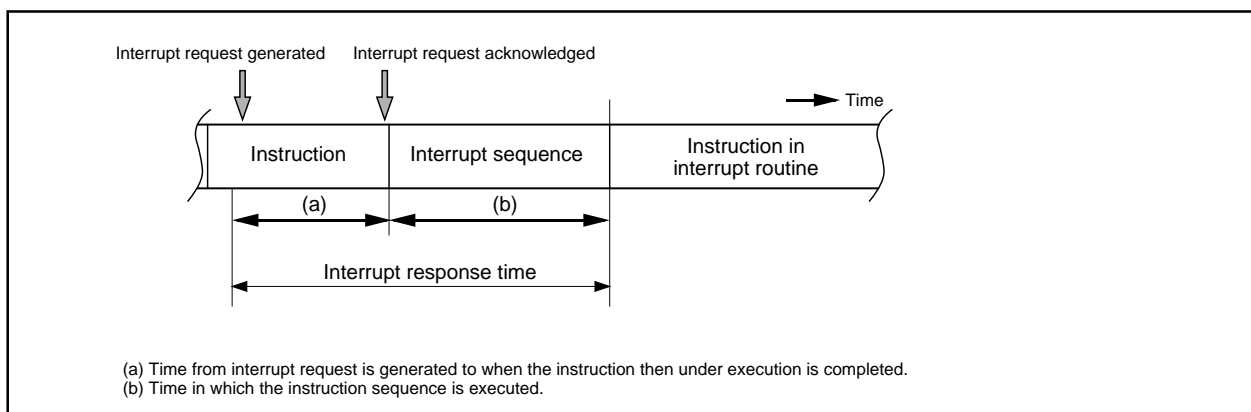


Figure DD-4. Interrupt response time

## Interrupt

Time (a) is dependent on the instruction under execution. Thirty cycles is the maximum required for the DIVX instruction.

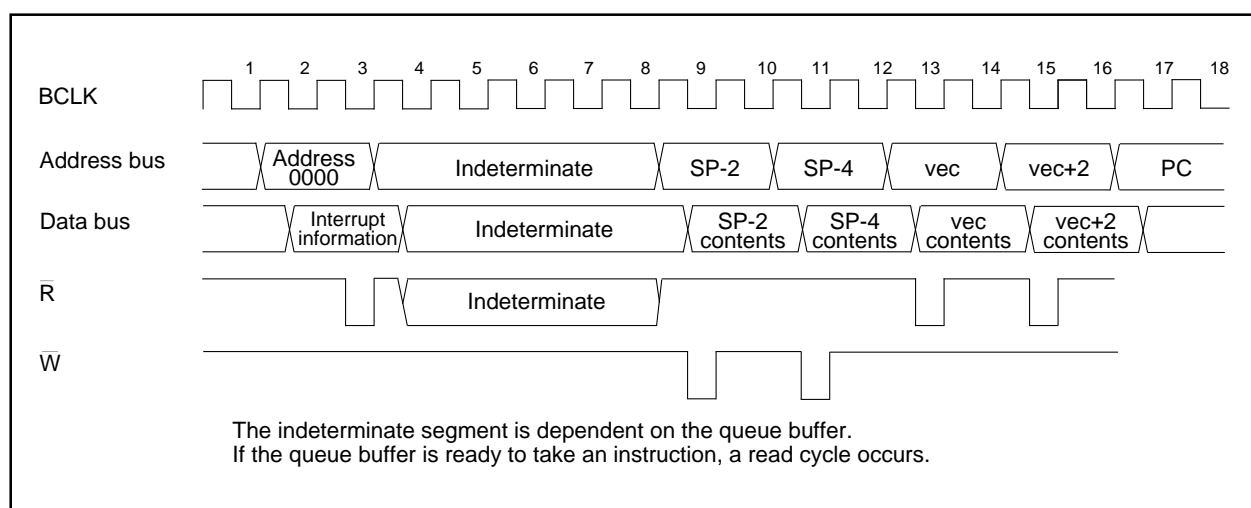
Time (b) is as shown in Table DD-5.

**Table DD-5. Time required for executing the interrupt sequence**

Interrupt vector address	Stack pointer (SP) value	16-Bit bust	8-Bit bus
Even	Even	18 cycles (Note 1)	20 cycles (Note 1)
Even	Odd	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Even	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Odd	20 cycles (Note 1)	20 cycles (Note 1)

Note 1: Add 2 cycles in the case of a DBC interrupt; add 1 cycle in the case either of an address coincidence interrupt or of a single-step interrupt.

Note 2: Locate an interrupt vector address in an even address, if possible.



**Figure DD-5. Time required for executing the interrupt sequence**

### Variation of IPL when Interrupt Request is Accepted

If an interrupt request is accepted, the interrupt priority level of the accepted interrupt is set in the IPL.

If an interrupt request, that does not have an interrupt priority level, is accepted, one of the values shown in Table DD-6 is set in the IPL.

**Table DD-6. Relationship between interrupts without interrupt priority levels and IPL**

Interrupt sources without priority levels	Value set in the IPL
Watchdog timer	7
Reset	0
Other	Not changed

### Saving Registers

In the interrupt sequence, only the contents of the flag register (FLG) and that of the program counter (PC) are saved in the stack area.

First, the processor saves the four higher-order bits of the program counter, and 4 upper-order bits and 8 lower-order bits of the FLG register, 16 bits in total, in the stack area, then saves 16 lower-order bits of the program counter. Figure DD-6 shows the state of the stack as it was before the acceptance of the interrupt request, and the state the stack after the acceptance of the interrupt request.

Save other necessary registers at the beginning of the interrupt routine using software. Using the PUSHM instruction alone can save all the registers except the stack pointer (SP).

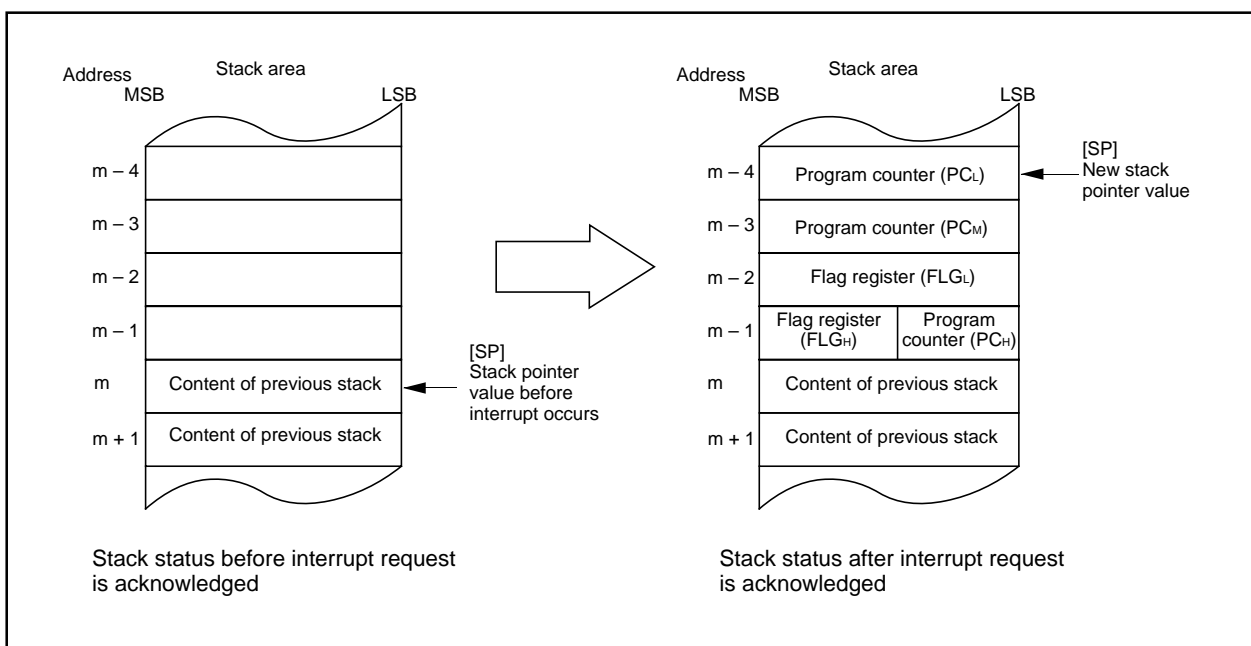


Figure DD-6. State of stack before and after acceptance of interrupt request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the content of the stack pointer, at the time of acceptance of an interrupt request, is even or odd- If the content of the stack pointer (Note) is even, the content of the flag register (FLG) and the content of the program counter (PC) are saved, 16 bits at a time. If odd, their contents are saved in two steps, 8 bits at a time. Figure DD-7 shows the operation of the saving registers.

Note: Stack pointer indicated by U flag.

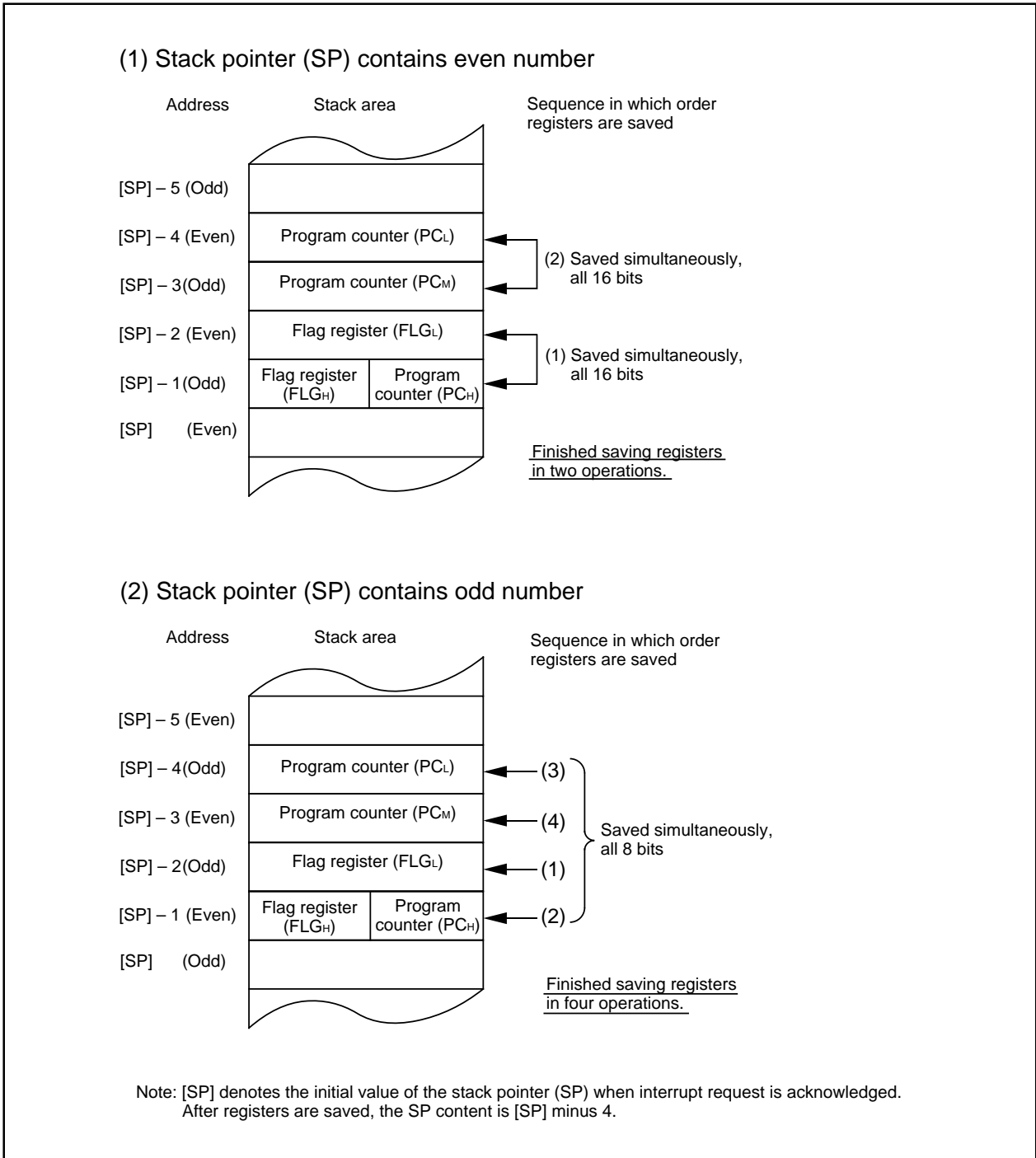


Figure DD-7. Operation of saving registers



## Returning from an Interrupt Routine

Executing the REIT instruction at the end of an interrupt routine returns the contents of the flag register (FLG) as it was immediately before the start of interrupt sequence and the contents of the program counter (PC), both of which have been saved in the stack area. Then control returns to the program that was being executed before the acceptance of the interrupt request, so that the suspended process resumes. Return the other registers saved by software within the interrupt routine using the POPM or similar instruction before executing the REIT instruction.

## Interrupt Priority

If there are two or more interrupt requests occurring at a point in time within a single sampling (checking whether interrupt requests are made), the interrupt assigned a higher priority is accepted.

Assign an arbitrary priority to maskable interrupts (peripheral I/O interrupts) using the interrupt priority level select bit. If the same interrupt priority level is assigned, however, the interrupt assigned a higher hardware priority is accepted.

Priorities of the special interrupts, such as Reset (dealt with as an interrupt assigned the highest priority), watchdog timer interrupt, etc. are regulated by hardware.

Figure DD-8 shows the priorities of hardware interrupts.

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

Reset >  $\overline{DBC}$  > Watchdog timer > Peripheral I/O > Single step > Address match

Figure DD-8. Hardware interrupts priorities

## Interrupt resolution circuit

When two or more interrupts are generated simultaneously, this circuit selects the interrupt with the highest priority level. Figure DD-9 shows the circuit that judges the interrupt priority level.

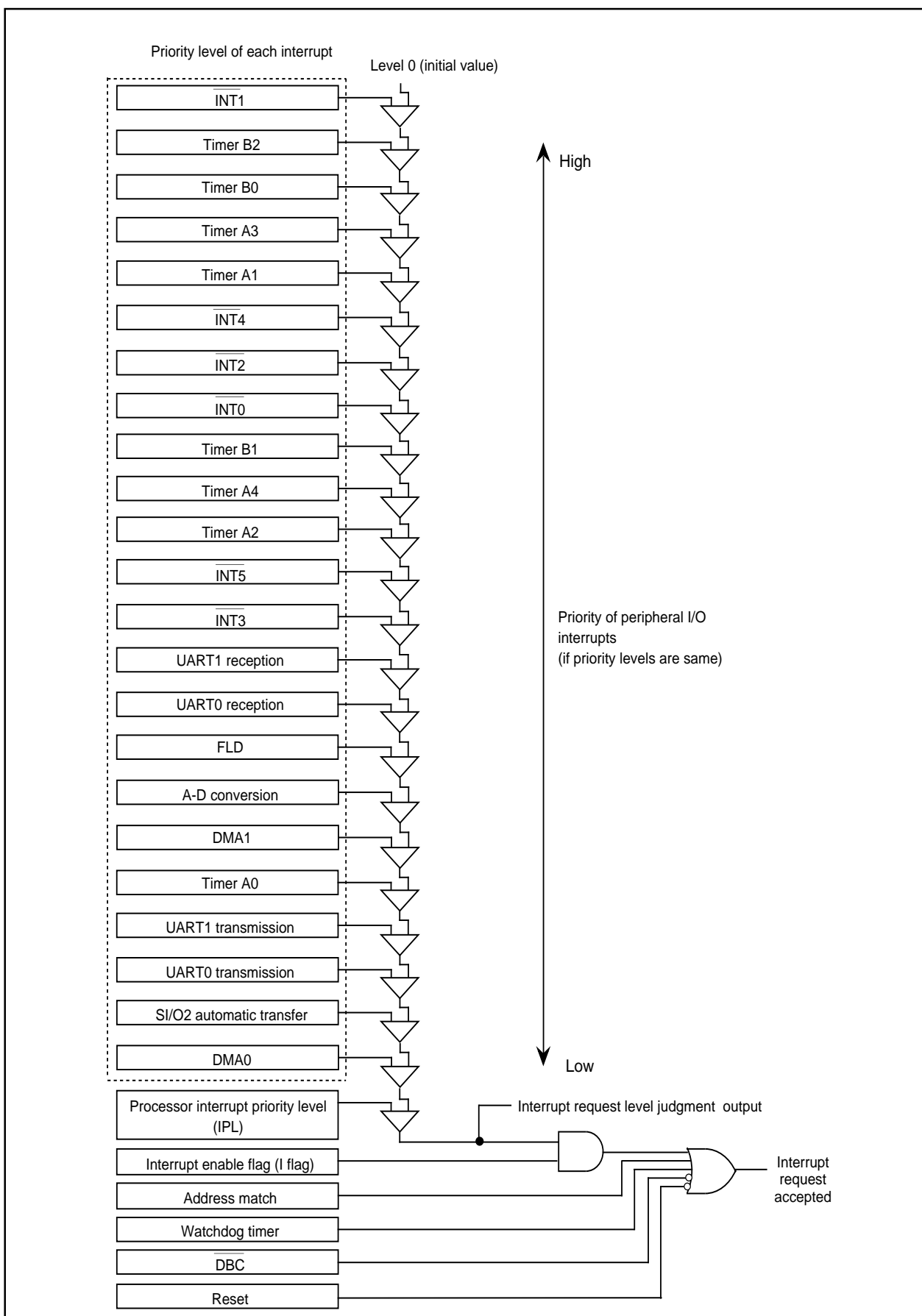


Figure DD-9. Maskable interrupts priorities

## Address Match Interrupt

## Address Match Interrupt

An address match interrupt is generated when the address match interrupt address register contents match the program counter value. Two address match interrupts can be set, each of which can be enabled and disabled by an address match interrupt enable bit. Address match interrupts are not affected by the interrupt enable flag (I flag) and processor interrupt priority level (IPL). The value of the program counter (PC) for an address match interrupt varies depending on the instruction being executed.

Figure DD-12 shows the address match interrupt-related registers.

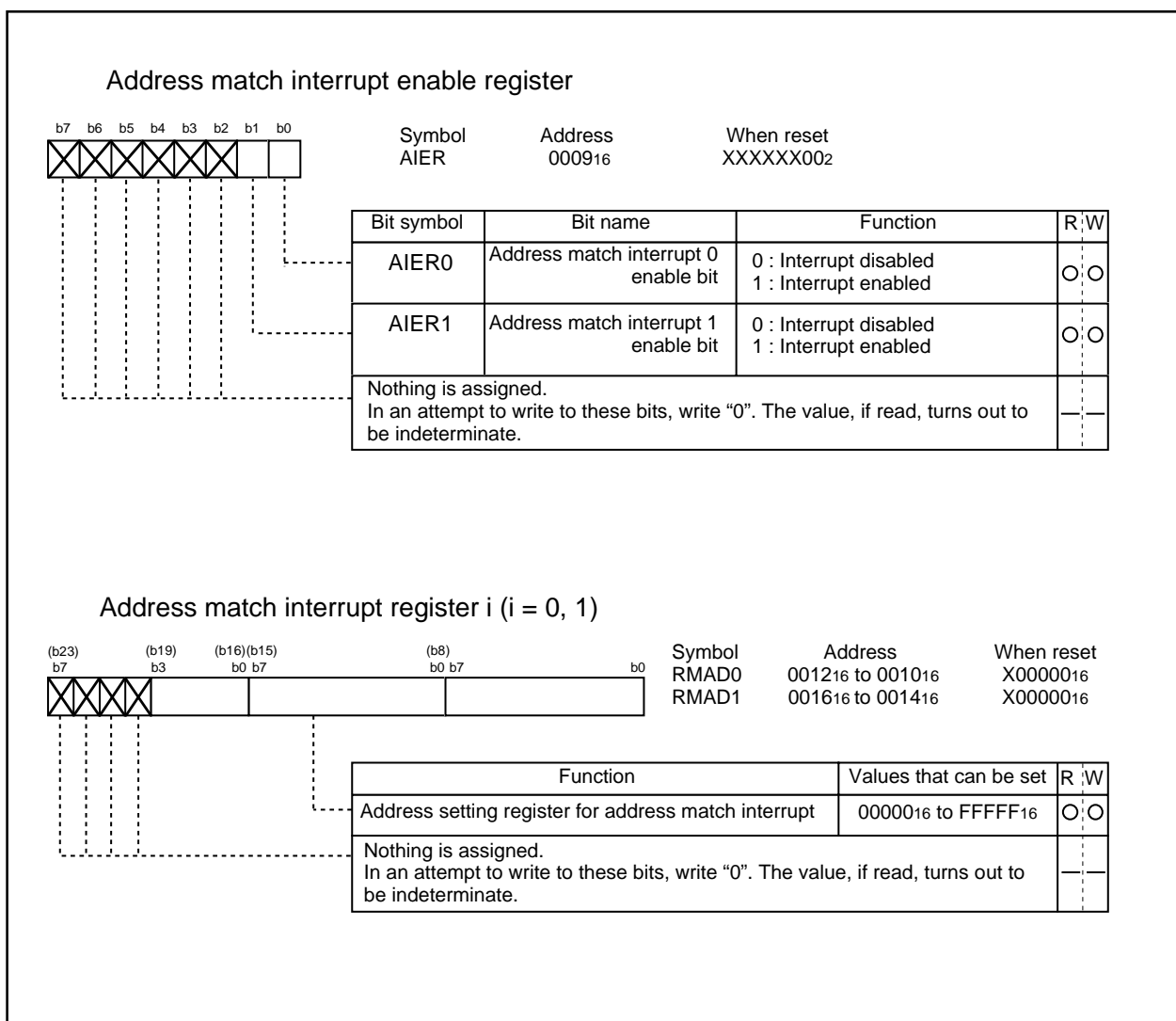


Figure DD-12. Address match interrupt-related registers

## Precautions for Interrupts

## Precautions for Interrupts

**(1) Reading address 00000<sub>16</sub>**

- When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.

The interrupt request bit of the certain interrupt written in address 00000<sub>16</sub> will then be set to "0".

Reading address 00000<sub>16</sub> by software sets enabled highest priority interrupt source request bit to "0".

Though the interrupt is generated, the interrupt routine may not be executed.

Do not read address 00000<sub>16</sub> by software.

**(2) Setting the stack pointer**

- The value of the stack pointer immediately after reset is initialized to 0000<sub>16</sub>. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt.

**(3) External interrupt**

- Either an "L" level or an "H" level of at least 250 ns width is necessary for the signal input to pins  $\overline{\text{INT}}_0$  through  $\overline{\text{INT}}_5$  regardless of the CPU operation clock.
- When the polarity of the  $\overline{\text{INT}}_0$  through  $\overline{\text{INT}}_5$  pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Figure DD-13 shows the procedure for changing the  $\overline{\text{INT}}$  interrupt generate factor.

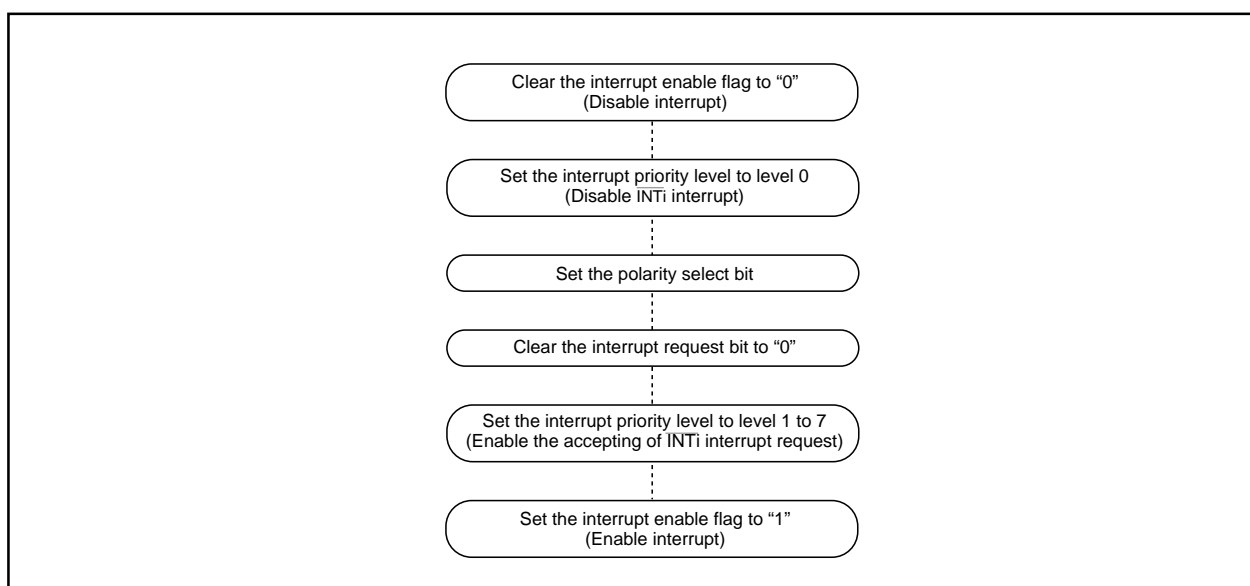


Figure DD-13. Switching condition of  $\overline{\text{INT}}$  interrupt request

**(5) Rewrite the interrupt control register**

- To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occur, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

**Example 1:**

```

INT_SWITCH1:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  NOP
  NOP
  FSET  I           ; Enable interrupts.

```

**Example 2:**

```

INT_SWITCH2:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  MOV.W MEM, R0    ; Dummy read.
  FSET  I           ; Enable interrupts.

```

**Example 3:**

```

INT_SWITCH3:
  PUSHC FLG
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  POPC  FLG        ; Enable interrupts.

```

The reason why two NOP instructions or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

- When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

## Watchdog Timer

## Watchdog Timer

The watchdog timer has the function of detecting when the program is out of control. The watchdog timer is a 15-bit counter which down-counts the clock derived by dividing the BCLK using the prescaler. A watchdog timer interrupt is generated when an underflow occurs in the watchdog timer. When XIN is selected for the BCLK, bit 7 of the watchdog timer control register (address 000F16) selects the prescaler division ratio (by 16 or by 128). When XCIN is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address 000F16). Thus the watchdog timer's period can be calculated as given below. The watchdog timer's period is, however, subject to an error due to the prescaler.

## With XIN chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{prescaler dividing ratio (16 or 128)} \times \text{watchdog timer count (32768)}}{\text{BCLK}}$$

## With XCIN chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{prescaler dividing ratio (2)} \times \text{watchdog timer count (32768)}}{\text{BCLK}}$$

For example, suppose that BCLK runs at 10 MHz and that 16 has been chosen for the dividing ratio of the prescaler, then the watchdog timer's period becomes approximately 52.4 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address 000E16) and when a watchdog timer interrupt request is generated. The prescaler is initialized only when the microcomputer is reset. After a reset is cancelled, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address 000E16).

Figure FA-1 shows the block diagram of the watchdog timer. Figure FA-2 shows the watchdog timer-related registers.

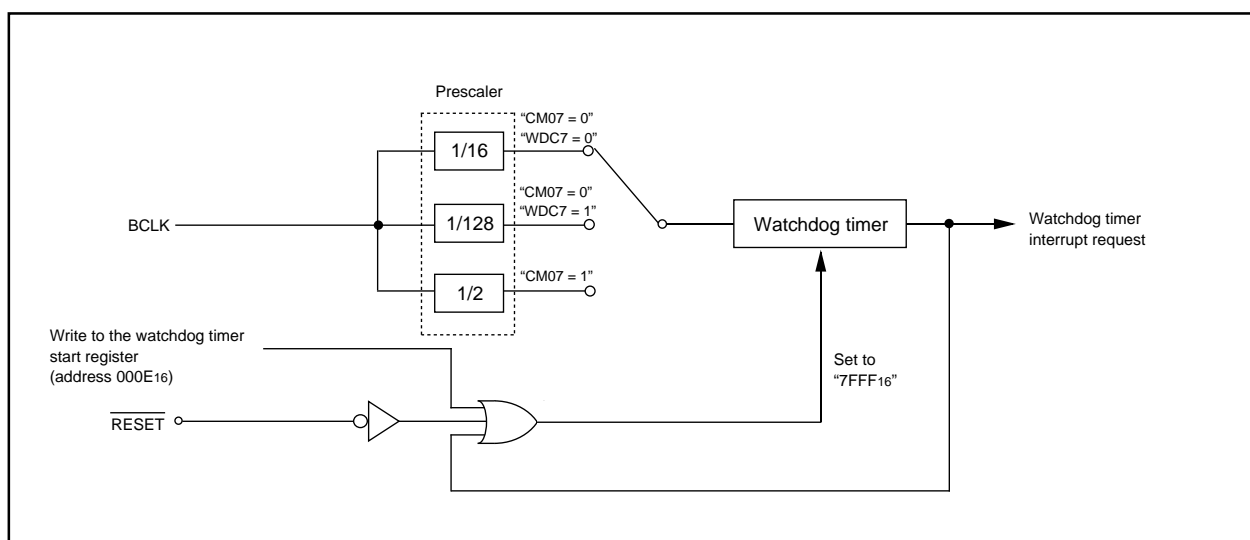


Figure FA-1. Block diagram of watchdog timer

Watchdog Timer

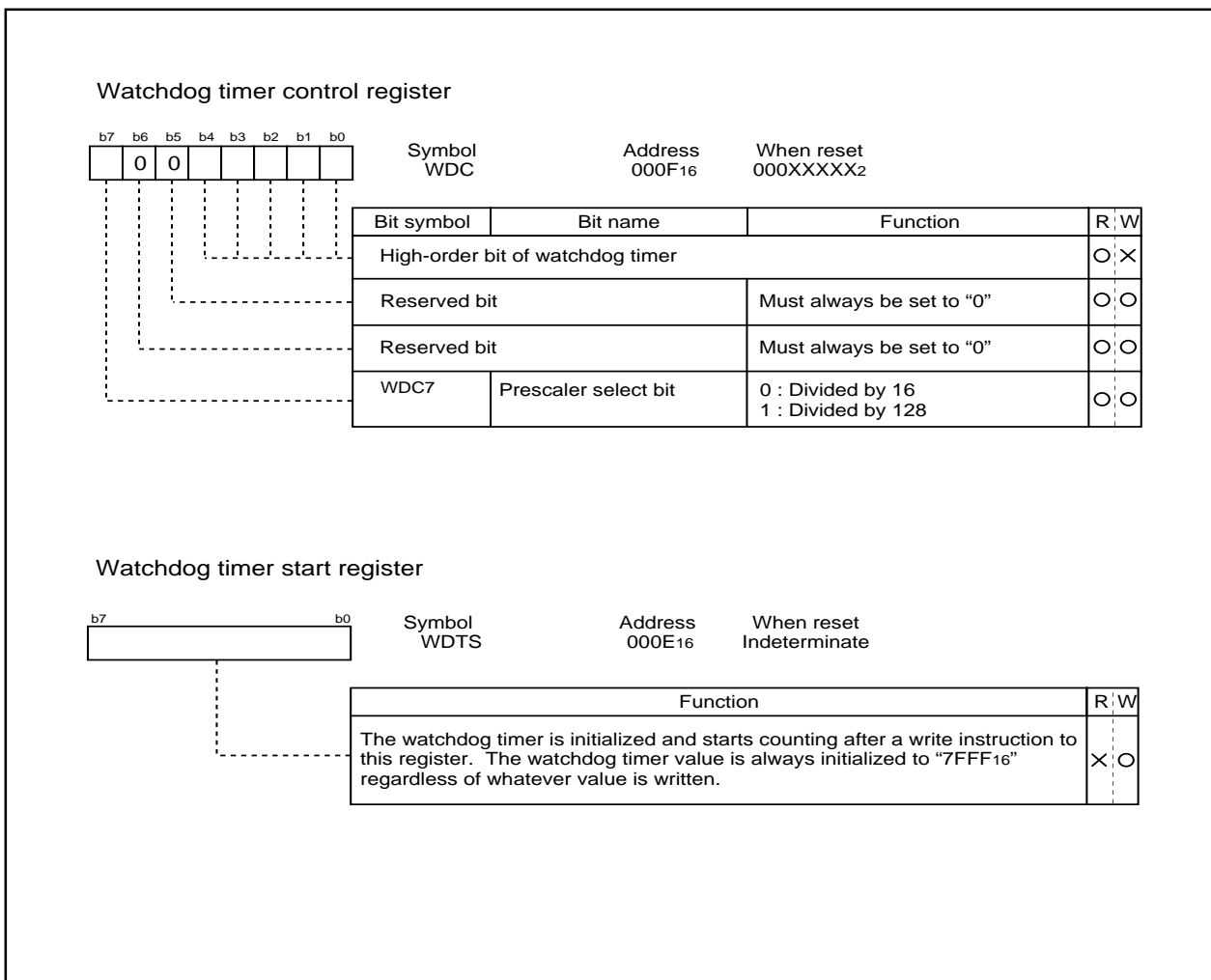


Figure FA-2. Watchdog timer control and start registers

## DMAC

This microcomputer has two DMAC (direct memory access controller) channels that allow data to be sent to memory without using the CPU. DMAC shares the same data bus with the CPU. The DMAC is given a higher right of using the bus than the CPU, which leads to working the cycle stealing method. On this account, the operation from the occurrence of DMA transfer request signal to the completion of 1-word (16-bit) or 1-byte (8-bit) data transfer can be performed at high speed. Figure EC-1 shows the block diagram of the DMAC. Table EC-1 shows the DMAC specifications. Figure EC-2 to Figure EC-3 show the registers used by the DMAC.

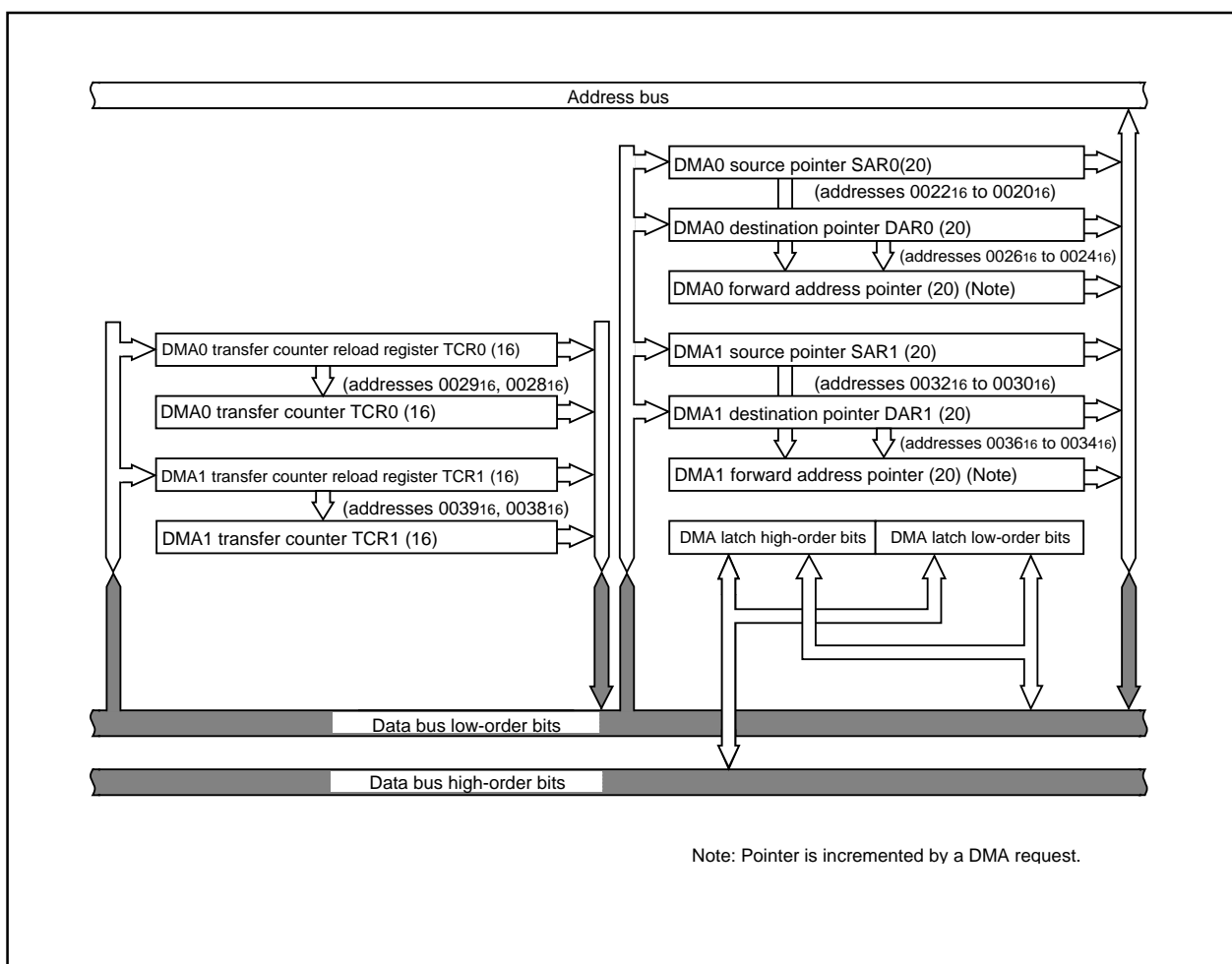


Figure EC-1. Block diagram of DMAC

Either a write signal to the software DMA request bit or an interrupt request signal is used as a DMA transfer request signal. But the DMA transfer is affected neither by the interrupt enable flag (I flag) nor by the interrupt priority level. The DMA transfer doesn't affect any interrupts either.

If the DMAC is active (the DMA enable bit is set to 1), data transfer starts every time a DMA transfer request signal occurs. If the cycle of the occurrences of DMA transfer request signals is higher than the DMA transfer cycle, there can be instances in which the number of transfer requests doesn't agree with the number of transfers. For details, see the description of the DMA request bit.



Table EC-1. DMAC specifications

Item	Specification
No. of channels	2 (cycle steal method)
Transfer memory space	<ul style="list-style-type: none"> <li>• From any address in the 1M bytes space to a fixed address</li> <li>• From a fixed address to any address in the 1M bytes space</li> <li>• From a fixed address to a fixed address</li> </ul> (Note that DMA-related registers [0020 <sub>16</sub> to 003F <sub>16</sub> ] cannot be accessed)
Maximum No. of bytes transferred	128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers)
DMA request factors (Note)	Falling edge of $\overline{INT0}$ or $\overline{INT1}$ ( $\overline{INT0}$ can be selected by DMA0, $\overline{INT1}$ by DMA1) Timer A0 to timer A4 interrupt requests Timer B0 to timer B2 interrupt requests UART0 transmission and reception interrupt requests UART1 transmission and reception interrupt requests A-D conversion interrupt requests Software triggers
Channel priority	DMA0 takes precedence if DMA0 and DMA1 requests are generated simultaneously
Transfer unit	8 bits or 16 bits
Transfer address direction	forward or fixed (forward direction cannot be specified for both source and destination simultaneously)
Transfer mode	<ul style="list-style-type: none"> <li>• Single transfer mode After the transfer counter underflows, the DMA enable bit turns to "0", and the DMAC turns inactive</li> <li>• Repeat transfer mode After the transfer counter underflows, the value of the transfer counter reload register is reloaded to the transfer counter. The DMAC remains active unless a "0" is written to the DMA enable bit.</li> </ul>
DMA interrupt request generation timing	When an underflow occurs in the transfer counter
Active	When the DMA enable bit is set to "1", the DMAC is active. When the DMAC is active, data transfer starts every time a DMA transfer request signal occurs.
Inactive	<ul style="list-style-type: none"> <li>• When the DMA enable bit is set to "0", the DMAC is inactive.</li> <li>• After the transfer counter underflows in single transfer mode</li> </ul>
Forward address pointer and load timing for transfer counter	At the time of starting data transfer immediately after turning the DMAC active, the value of one of source pointer and destination pointer - the one specified for the forward direction - is reloaded to the forward direction address pointer, and the value of the transfer counter reload register is reloaded to the transfer counter.
Writing to register	Registers specified for forward direction transfer are always write enabled. Registers specified for fixed address transfer are write-enabled when the DMA enable bit is "0".
Reading the register	Can be read at any time. However, when the DMA enable bit is "1", reading the register set up as the forward register is the same as reading the value of the forward address pointer.

Note: DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the interrupt enable flag (I flag) nor by the interrupt priority level.

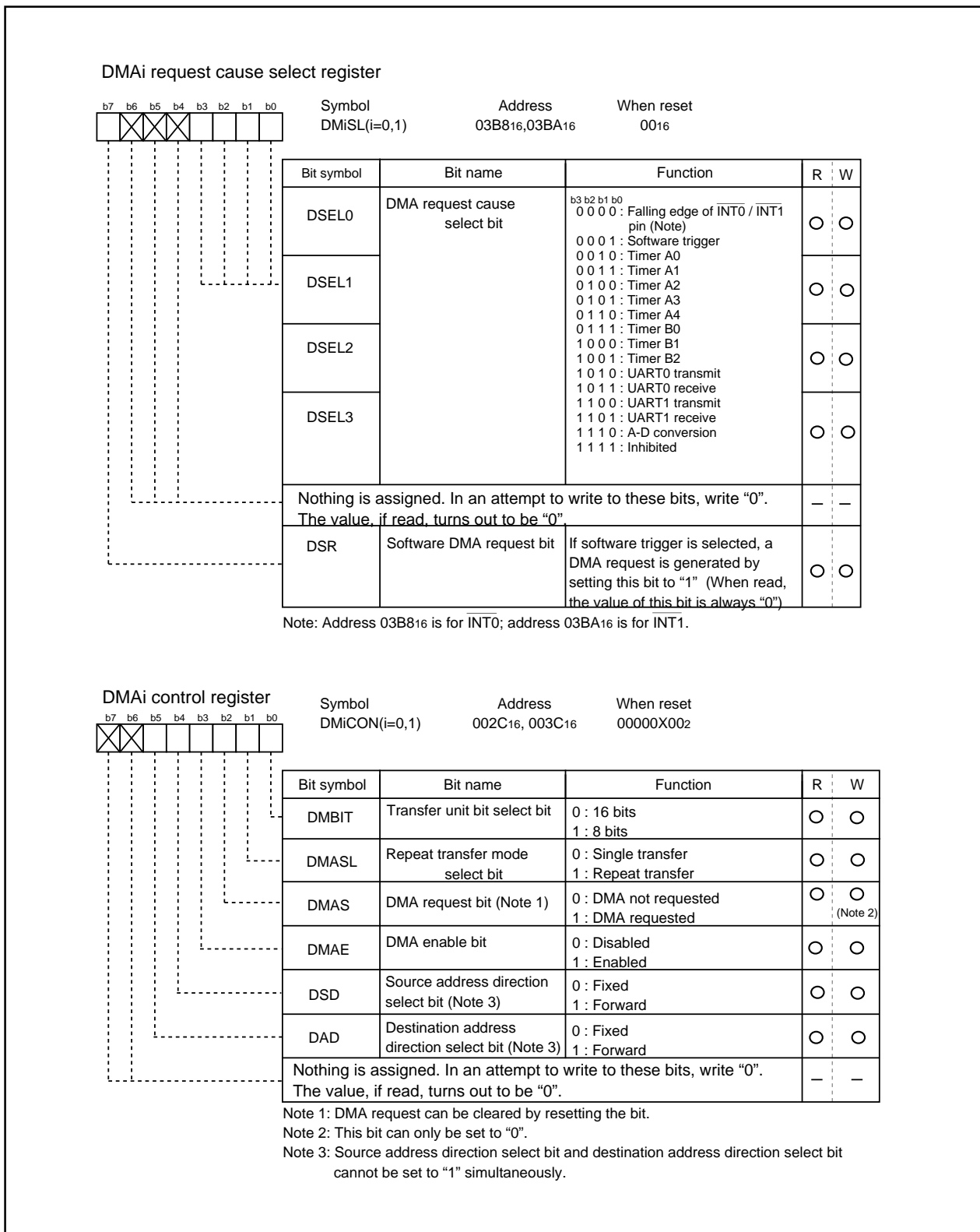


Figure EC-2. DMAC-related registers (1)

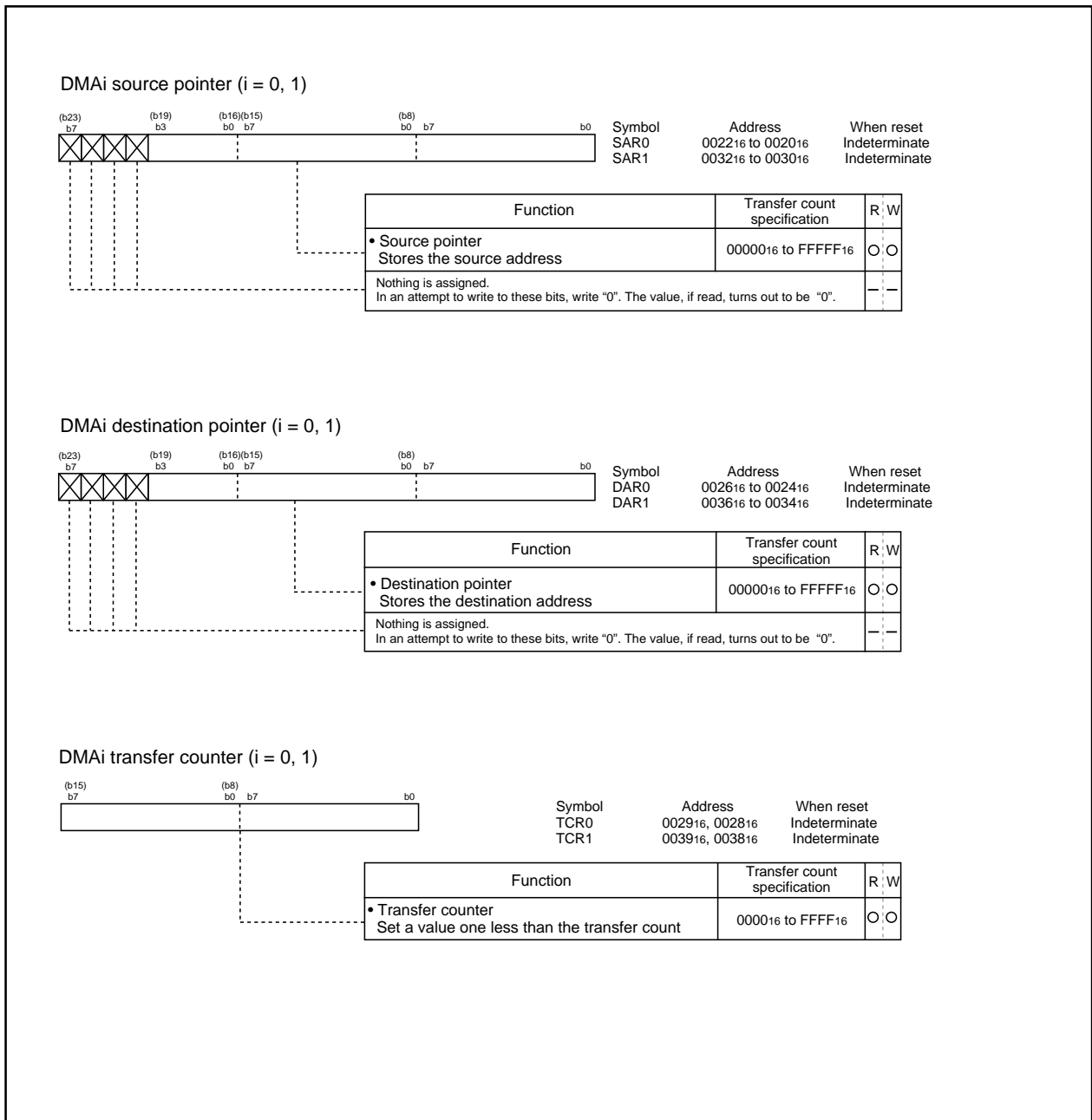


Figure EC-3. DMAC-related registers (2)

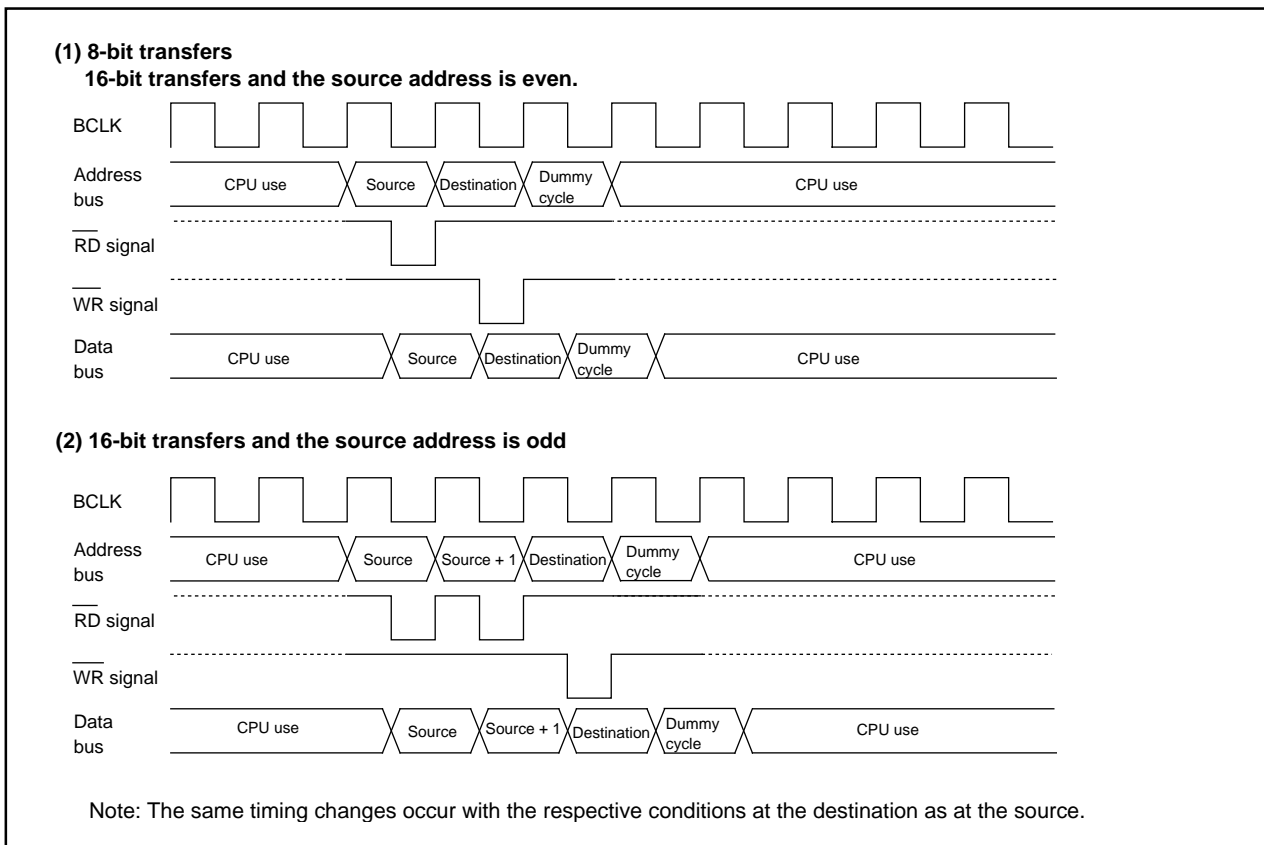
**(1) Transfer cycle**

The transfer cycle consists of the bus cycle in which data is read from memory or from the SFR area (source read) and the bus cycle in which the data is written to memory or to the SFR area (destination write). The number of read and write bus cycles depends on the source and destination addresses.

(a) Effect of source and destination addresses

When 16-bit data is transferred on a 16-bit data bus, and the source and destination both start at odd addresses, there are one more source read cycle and destination write cycle than when the source and destination both start at even addresses.

Figure EC-4 shows the example of the transfer cycles (a state of internal bus) for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating the transfer cycle, remember to apply the respective conditions to both the destination write cycle and the source read cycle.



**Figure EC-4. Example of transfer cycles for a source read (the state of internal bus)**

**(2) DMAC Transfer**

Any combination of even or odd transfer read and write addresses is possible. Table EC-2 shows the number of DMAC transfer cycles.

The number of DMAC transfer cycles can be calculated as follows:

$$\text{No. of transfer cycles per transfer unit} = \text{No. of read cycles} \times j + \text{No. of write cycles} \times k$$

**Table EC-2. No. of DMAC transfer cycles**

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

**Coefficient j, k**

Internal memory	
Internal ROM/RAM	SFR area
1	2

## FLD Controller

The M30218 group has fluorescent display (FLD) drive and control circuits.

Table KA-0 shows the FLD controller specifications.

**Table KA-0. FLD controller specifications**

Item		Specification
FLD controller port	High-breakdown-voltage output port	• 52 pins ( 20 pins can switch general purpose port)
	CMOS port	• 4 pins ( 4 pins can switch general purpose port) (A driver must be installed externally)
Display pixel number		<ul style="list-style-type: none"> <li>• Used FLD output 28 segment X 28 digit (segment number + digit number ≤ 56)</li> <li>• Used digit output 40 segment X 16 digit (segment number ≤ 40, digit number ≤ 16)</li> <li>• Connected to M35501 56 segment X (connect number of M35501) digit (segment number ≤ 56, digit number ≤ number of M35501 X 16)</li> <li>• Used P44 to P47 expansion 52 segment X 16 digit (segment number ≤ 52, digit number ≤ 16)</li> </ul>
Period		<ul style="list-style-type: none"> <li>• 3.2 μs to 819.2 μs (count source X<sub>IN</sub>/32,10MHz)</li> <li>• 12.8 μs to 3276.8 μs (count source X<sub>IN</sub>/128,10MHz)</li> </ul>
Dimmer time		<ul style="list-style-type: none"> <li>• 3.2 μs to 819.2 μs (count source X<sub>IN</sub>/32,10MHz)</li> <li>• 12.8 μs to 3276.8 μs (count source X<sub>IN</sub>/128,10MHz)</li> </ul>
Interrupt		<ul style="list-style-type: none"> <li>• Digit interrupt</li> <li>• FLD blanking interrupt</li> </ul>
Key-scan		<ul style="list-style-type: none"> <li>• Key-scan used digit</li> <li>• Key-scan used segment</li> </ul>
Expand function		<ul style="list-style-type: none"> <li>• Digit pulse output function This function automatically outputs digit pulse.</li> <li>• M35501 connect function The number of digits can be increased easily by using the output of DIMOUT(P97) as CLK for the M35501.</li> <li>• Toff section generate / not generate function This function does not generate Toff1 section when the connected outputs are the same.</li> <li>• Gradation display function This function allows each segment to be set for dark or bright display.</li> <li>• P44 to P47 expansion function This function provides 16 lines of digit outputs from four ports by attaching a 4 — 16 decoder.</li> </ul>

FLD controller

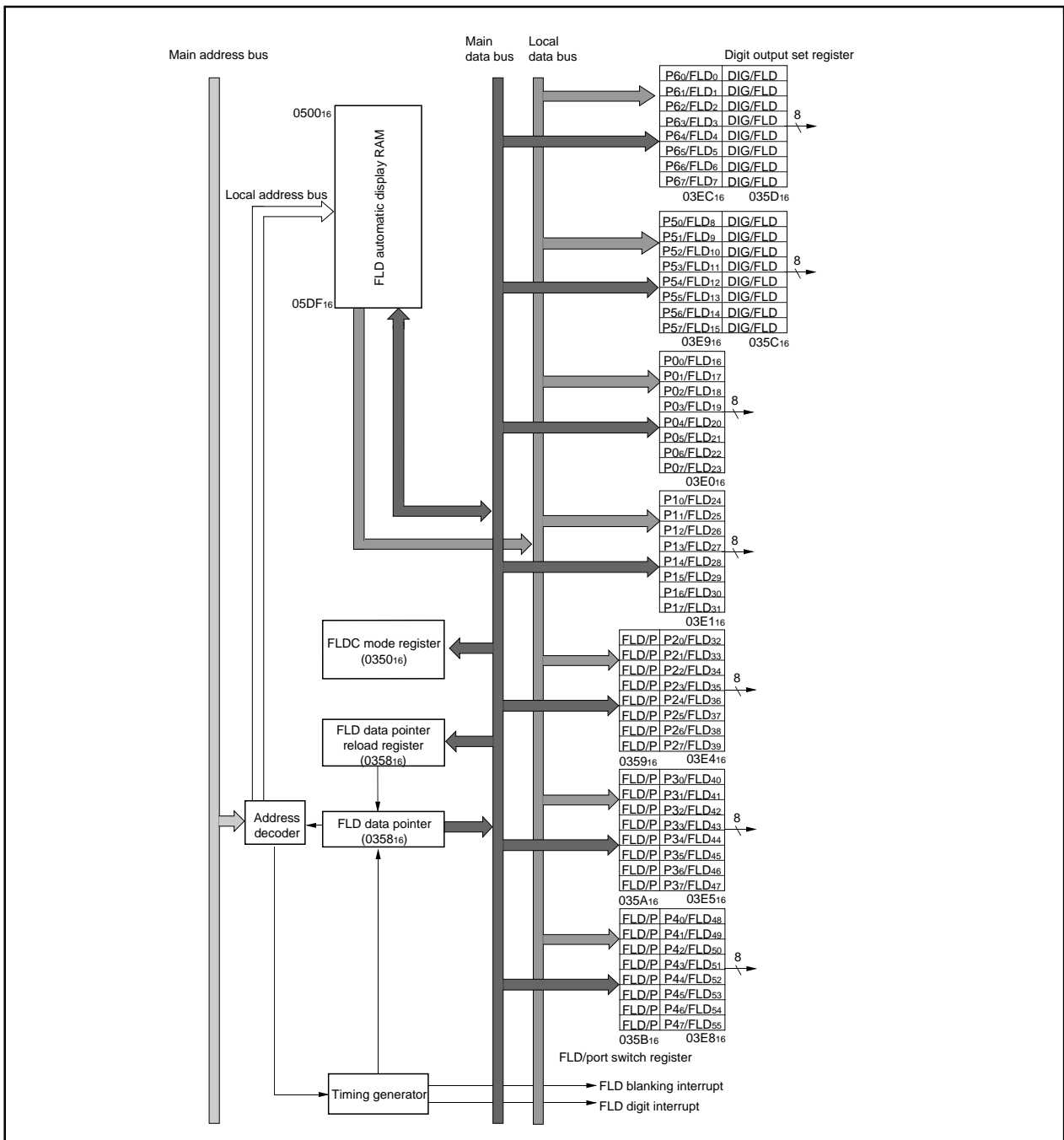


Figure KA-1. Block Diagram for FLD Control Circuit

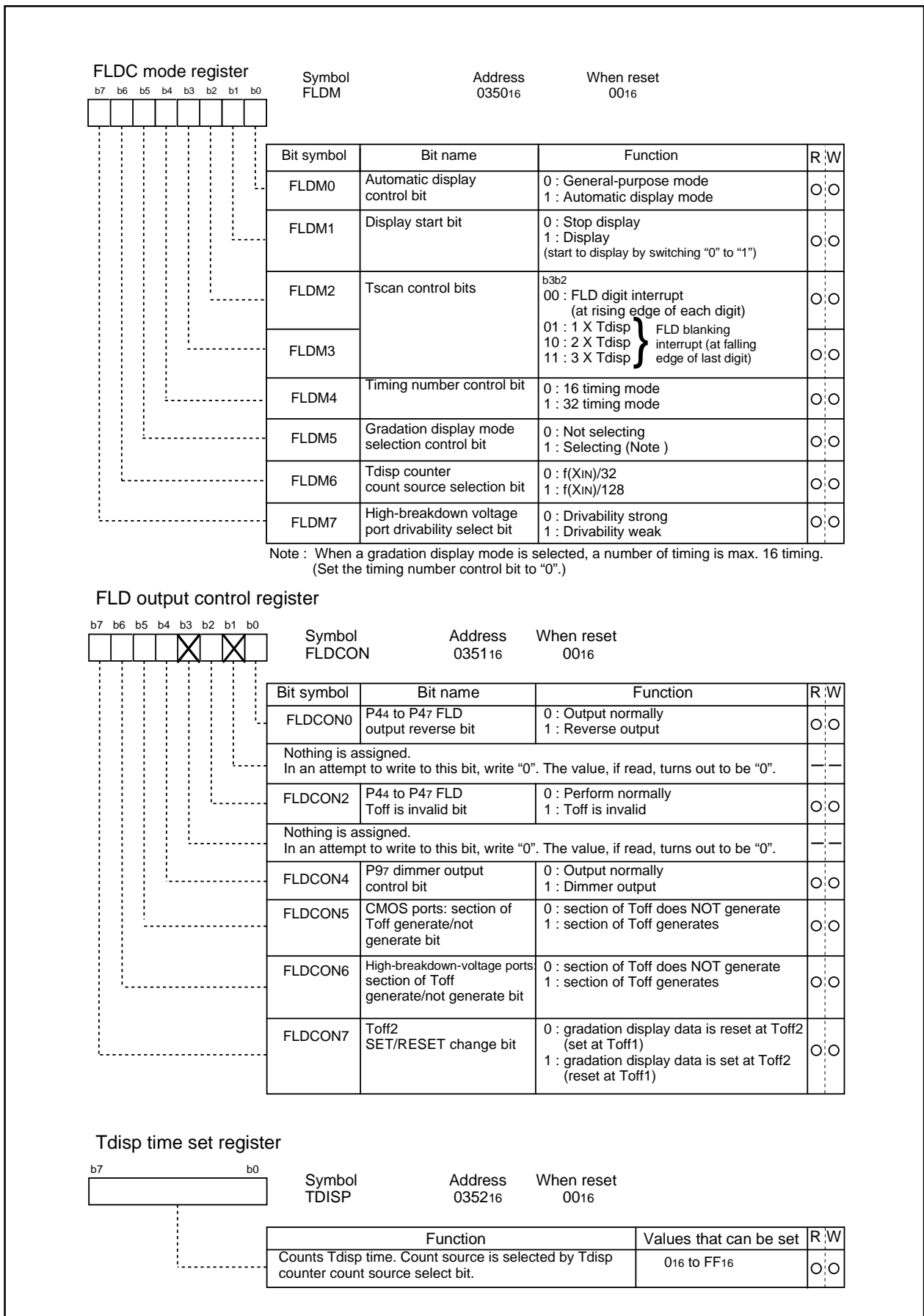


Figure KA-2. FLDC-related Register(1)



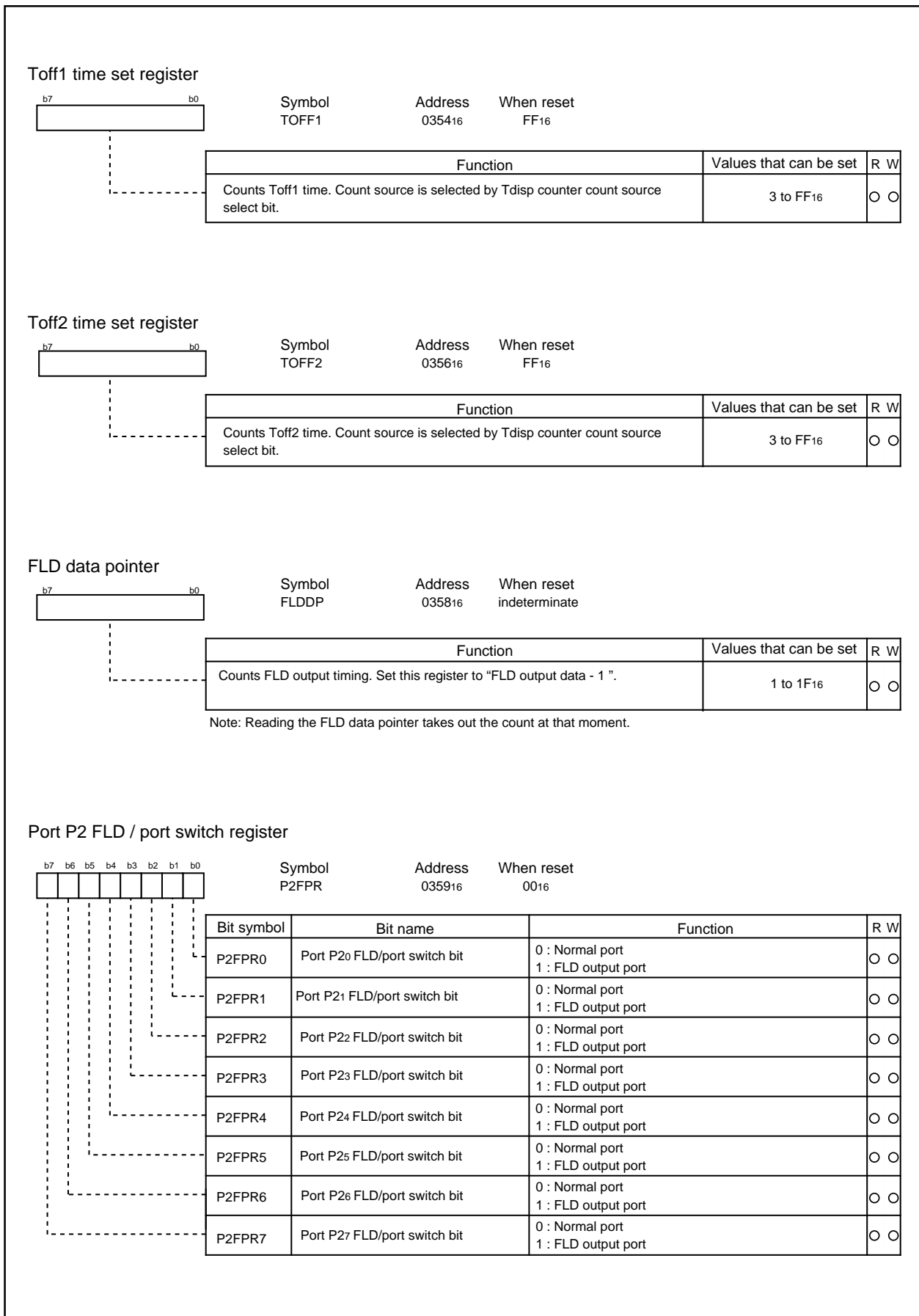


Figure KA-2A. FLDC-related Register(2)

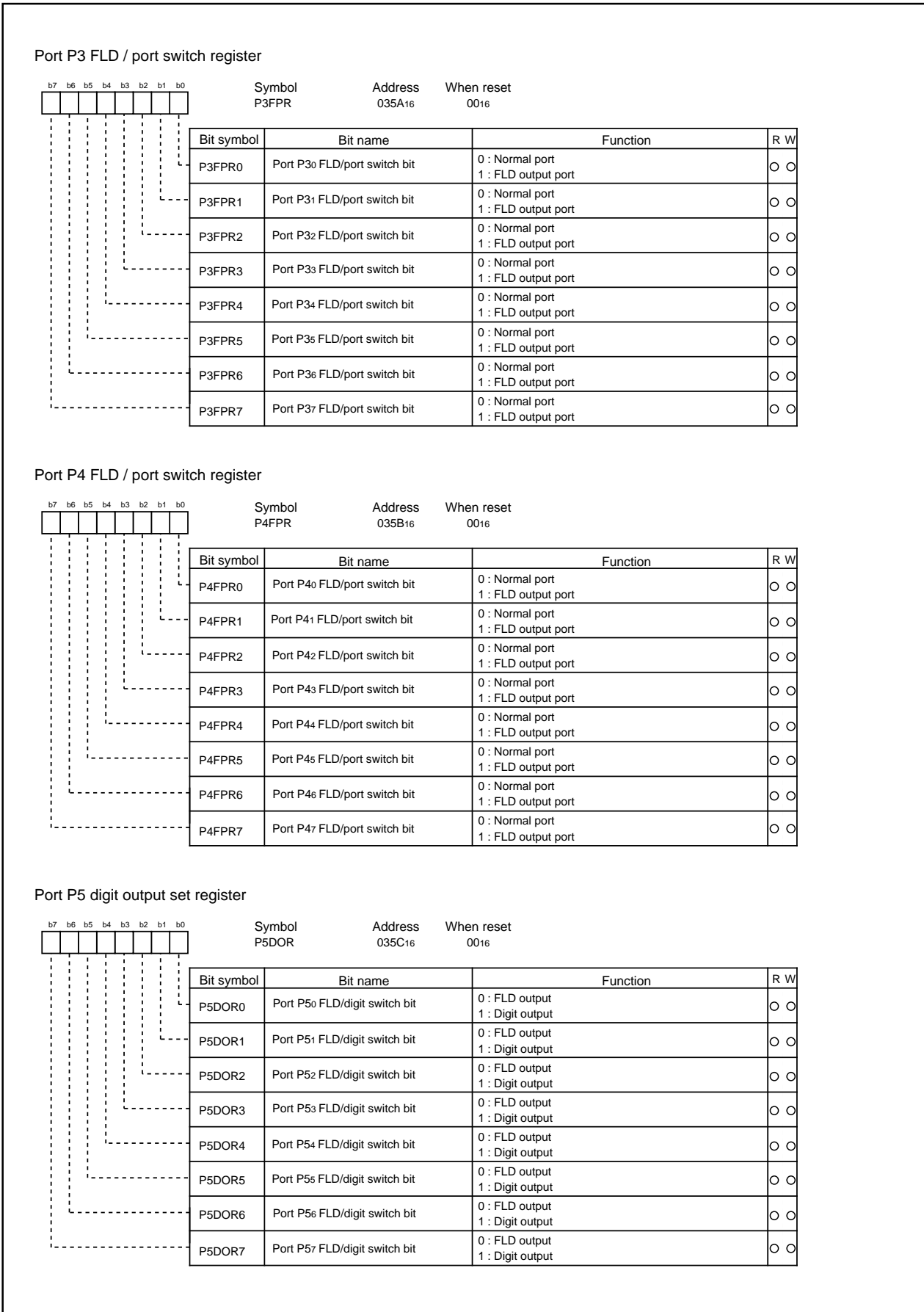


Figure KA-2B. FLDC-related Register(3)

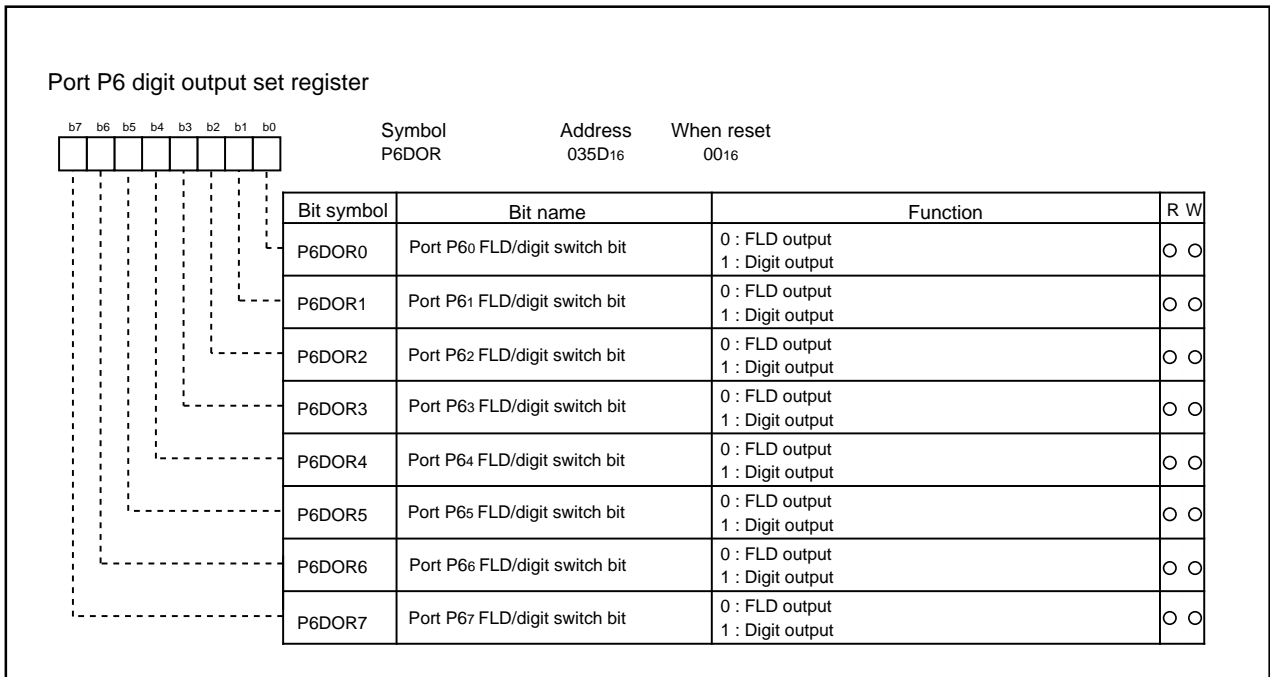


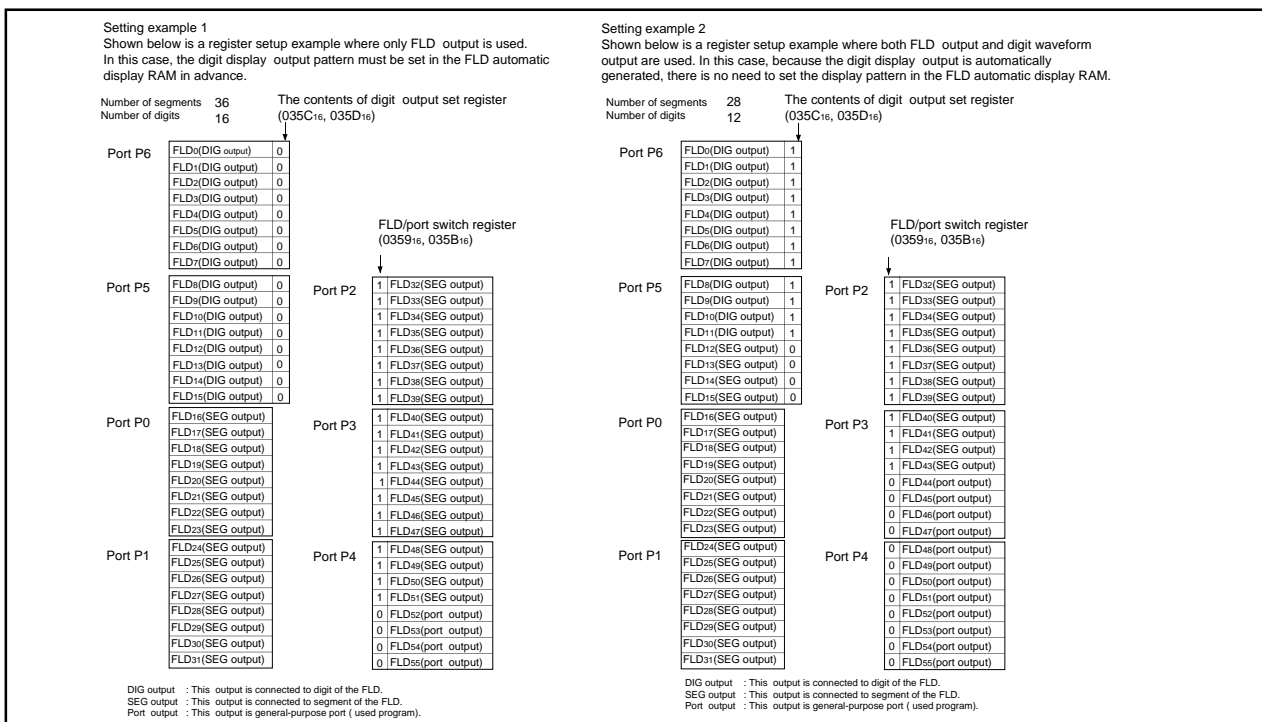
Figure KA-2C. FLDC-related Register(4)

### FLD automatic display pins

P0 to P6 are the pins capable of automatic display output for the FLD. The FLD start operating by setting the automatic display control bit (bit 0 at address 035016) to "1". There is the FLD output function that outputs RAM contents from the port every timing or the digit output function that drives the port high with digit timing. The FLD can be displayed using the FLD output for the segments and the digit or FLD output for the digits. When using the FLD output for the digits, be sure to write digit display patterns to the RAM in advance. The remaining segment and digit lines can be used as general-purpose ports. Settings of each port are shown below.

**Table KA-1. Pins in FLD Automatic Display Mode**

Port Name	Automatic Display Pins	Setting Method
P5, P6	FLD <sub>0</sub> to FLD <sub>15</sub>	The individual bits of the digit output set register (address 035C16, 035D16) can set each pin either FLD port ("0") or digit port ("1"). When the pins are set for the digit port, the digit pulse output function is enabled, so the digit pulses can always be output regardless the value of FLD automatic display RAM.
P0, P1	FLD <sub>16</sub> to FLD <sub>31</sub>	FLD exclusive use port (automatic display control bit (bit 0 of address 035016)="1")
P2, P3, P44 to P43	FLD <sub>32</sub> to FLD <sub>51</sub>	The individual bits of the FLD/port switch register (addresses 035916 to 035B16) can set each pin to either FLD port ("1") or general-purpose port ("0").
P44 to P47	FLD <sub>52</sub> to FLD <sub>55</sub>	The individual bits of the FLD/port switch register (address 035B16) can set each pin to either FLD port ("1") or general-purpose port ("0"). The digit pulse output function turns to available, and the digit pulse can output by setting of the FLD output set register (address 035116). The port output format is the CMOS output. When using the port as a display pin, a driver must be installed externally.



**Figure KA-3. Segment/Digit Setting Example**

## FLD automatic display RAM

The FLD automatic display RAM uses the 224 bytes of addresses 0500<sub>16</sub> to 05DF<sub>16</sub>. For FLD, the 3 modes of 16-timing ordinary mode, 16-timing•gradation display mode and 32-timing mode are available depending on the number of timings and the use/not use of gradation display.

The automatic display RAM in each mode is as follows:

### (1) 16-timing•Ordinary Mode

This mode is used when the display timing is 16 or less. The 112 bytes of addresses 0570<sub>16</sub> to 05DF<sub>16</sub> are used as a FLD display data store area. Because addresses 0500<sub>16</sub> to 056F<sub>16</sub> are not used as the automatic display RAM, they can be the ordinary RAM.

### (2) 16-timing•Gradation Display Mode

This mode is used when the display timing is 16 or less, in which mode each segment can be set for dark or bright display. The 224 bytes of addresses 0500<sub>16</sub> to 05DF<sub>16</sub> are used. The 112 bytes of addresses 0570<sub>16</sub> to 05DF<sub>16</sub> are used as an FLD display data store area, while the 112 bytes of addresses 0500<sub>16</sub> to 056F<sub>16</sub> are used as a gradation display control data store area.

### (3) 32-timing Mode

This mode is used when the display timing is 16 or greater. This mode can be used for up to 32-timing. The 224 bytes of addresses 0500<sub>16</sub> to 05DF<sub>16</sub> are used as an FLD display data store area.

The FLD data pointer (address 0358<sub>16</sub>) is a register to count display timings. This pointer has a reload register and when the terminal count is reached, it starts counting over again after being reloaded with the initial count. Make sure the timing count – 1 is set to the FLD data pointer. When writing data to this address, the data is written to the FLD data pointer reload register; when reading data from this address, the value in the FLD data pointer is read.

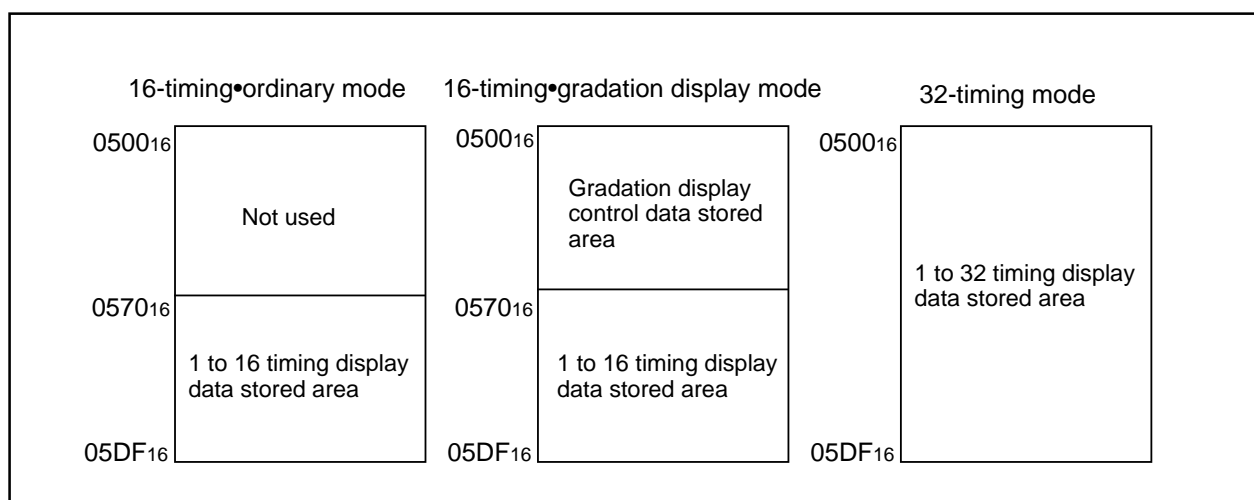


Figure KA-4. FLD Automatic Display RAM Assignment

Data setup

**(1) 16-timing•Ordinary Mode**

The area of addresses 0570<sub>16</sub> to 05DF<sub>16</sub> are used as a FLD automatic display RAM.

When data is stored in the FLD automatic display RAM, the last data of FLD port P4 is stored at address 0570<sub>16</sub>, the last data of FLD port P3 is stored at address 0580<sub>16</sub>, the last data of FLD port P2 is stored at address 0590<sub>16</sub>, the last data of FLD port P1 is stored at address 05A0<sub>16</sub>, the last data of FLD port P0 is stored at address 05B0<sub>16</sub>, the last data of FLD port P5 is stored at address 05C0<sub>16</sub>, and the last data of FLD port P6 is stored at address 05D0<sub>16</sub>, to assign in sequence from the last data respectively.

The first data of the FLD port P4, P3, P2, P1, P0, P5, and P6 is stored at an address which adds the value of (the timing number – 1) to the corresponding address 0570<sub>16</sub>, 0580<sub>16</sub>, 0590<sub>16</sub>, 05A0<sub>16</sub>, 05B0<sub>16</sub>, 05C0<sub>16</sub> and 05DF<sub>16</sub>.

Set the FLD data pointer reload register to the value given by the number of digits – 1.

**(2) 16-timing•Gradation Display Mode**

Display data setting is performed in the same way as that of the 16-timing•ordinary mode. Gradation display control data is arranged at an address resulting from subtracting 0070<sub>16</sub> from the display data store address of each timing and pin. Bright display is performed by setting “0”, and dark display is performed by setting “1” .

**(3) 32-timing Mode**

The area of addresses 0500<sub>16</sub> to 05DF<sub>16</sub> are used as a FLD automatic display RAM.

When data is stored in the FLD automatic display RAM, the last data of FLD port P4 is stored at address 0500<sub>16</sub>, the last data of FLD port P3 is stored at address 0520<sub>16</sub>, the last data of FLD port P2 is stored at address 0540<sub>16</sub>, the last data of FLD port P1 is stored at address 0560<sub>16</sub>, the last data of FLD port P0 is stored at address 0580<sub>16</sub>, the last data of FLD port P5 is stored at address 05A0<sub>16</sub>, and the last data of FLD port P6 is stored at address 05C0<sub>16</sub>, to assign in sequence from the last data respectively.

The first data of the FLD port P4, P3, P2, P0, P1, P5, and P6 is stored at an address which adds the value of (the timing number – 1) to the corresponding address 0500<sub>16</sub>, 0520<sub>16</sub>, 0540<sub>16</sub>, 0560<sub>16</sub>, 0580<sub>16</sub>, 05A0<sub>16</sub> and 05C0<sub>16</sub>.

Set the FLD data pointer reload register to the value given by the number of digits - 1.

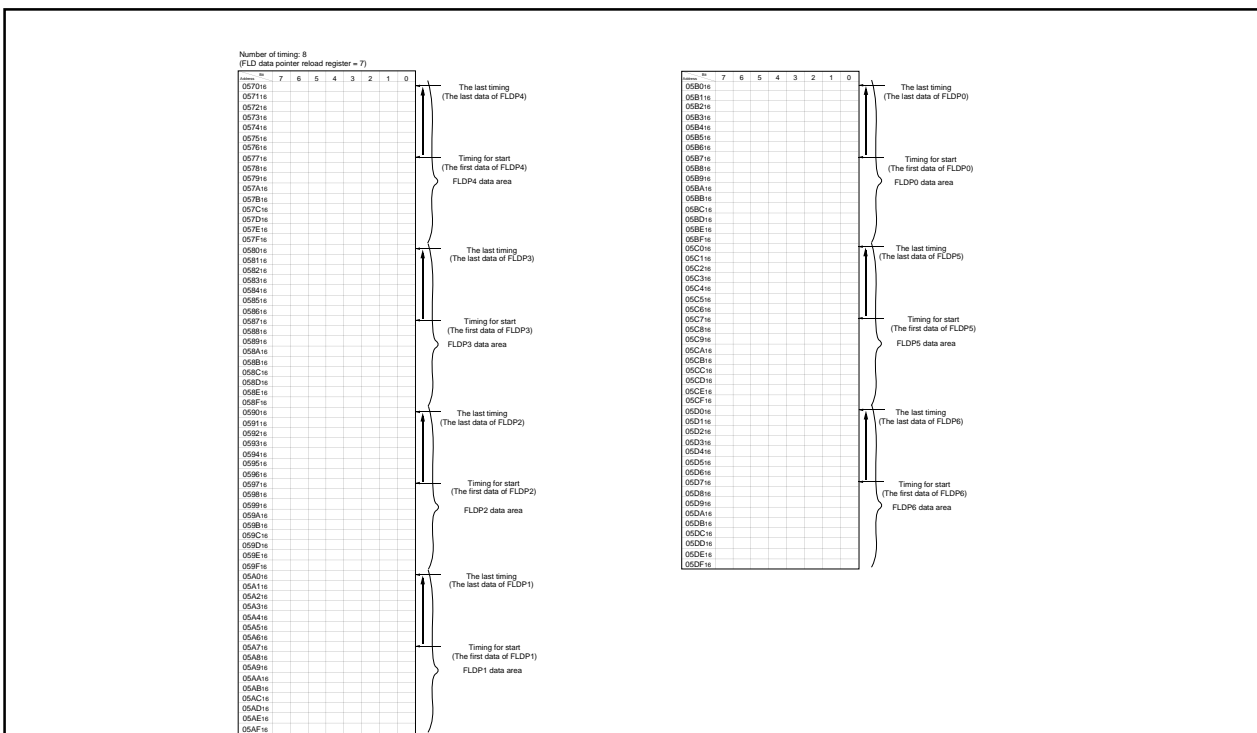


Figure KA-5. Example of Using the FLD Automatic Display RAM in 16-timing•Ordinary Mode

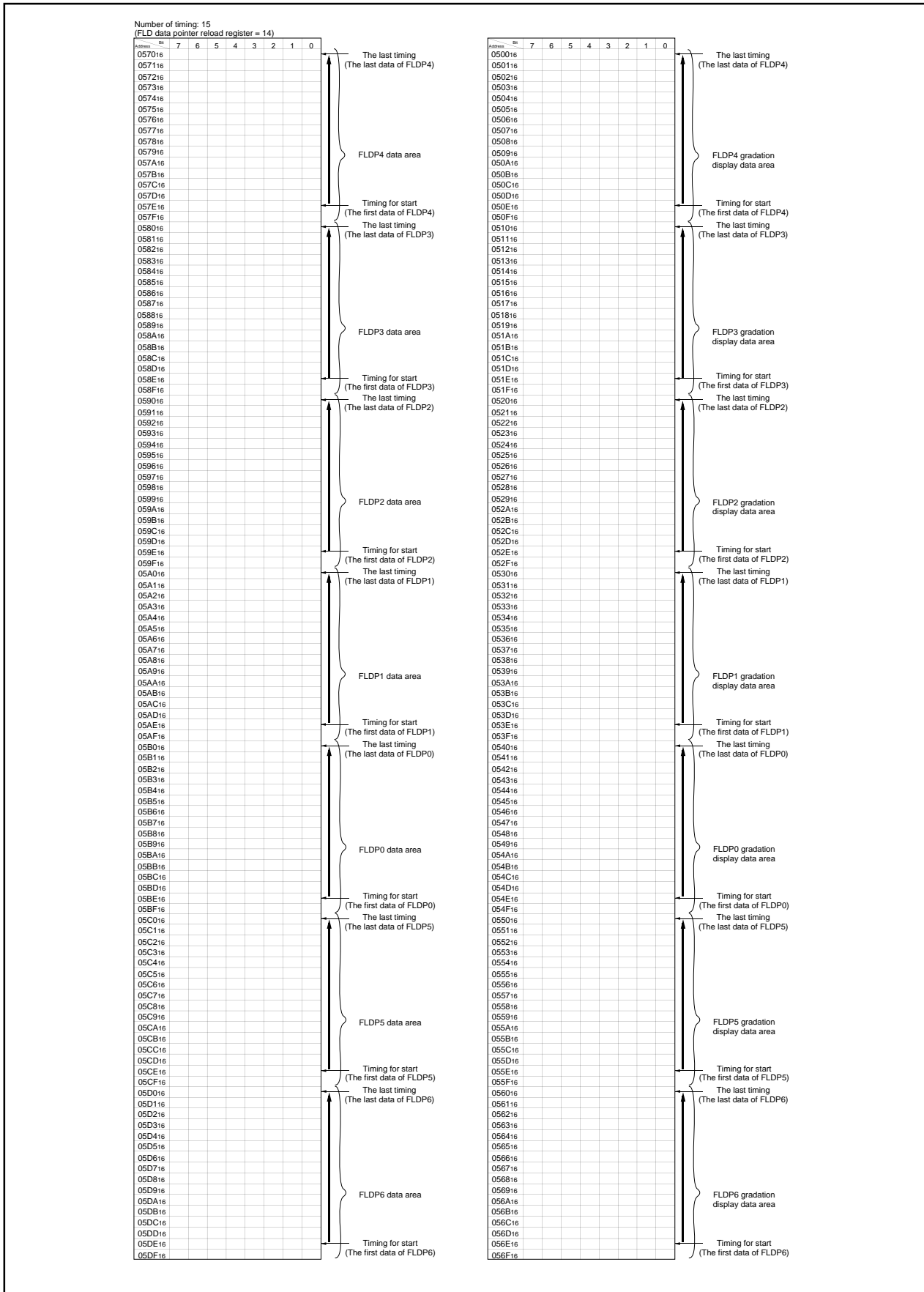


Figure KA-6. Example of Using the FLD Automatic Display RAM in 16-timing Gradation Display Mode

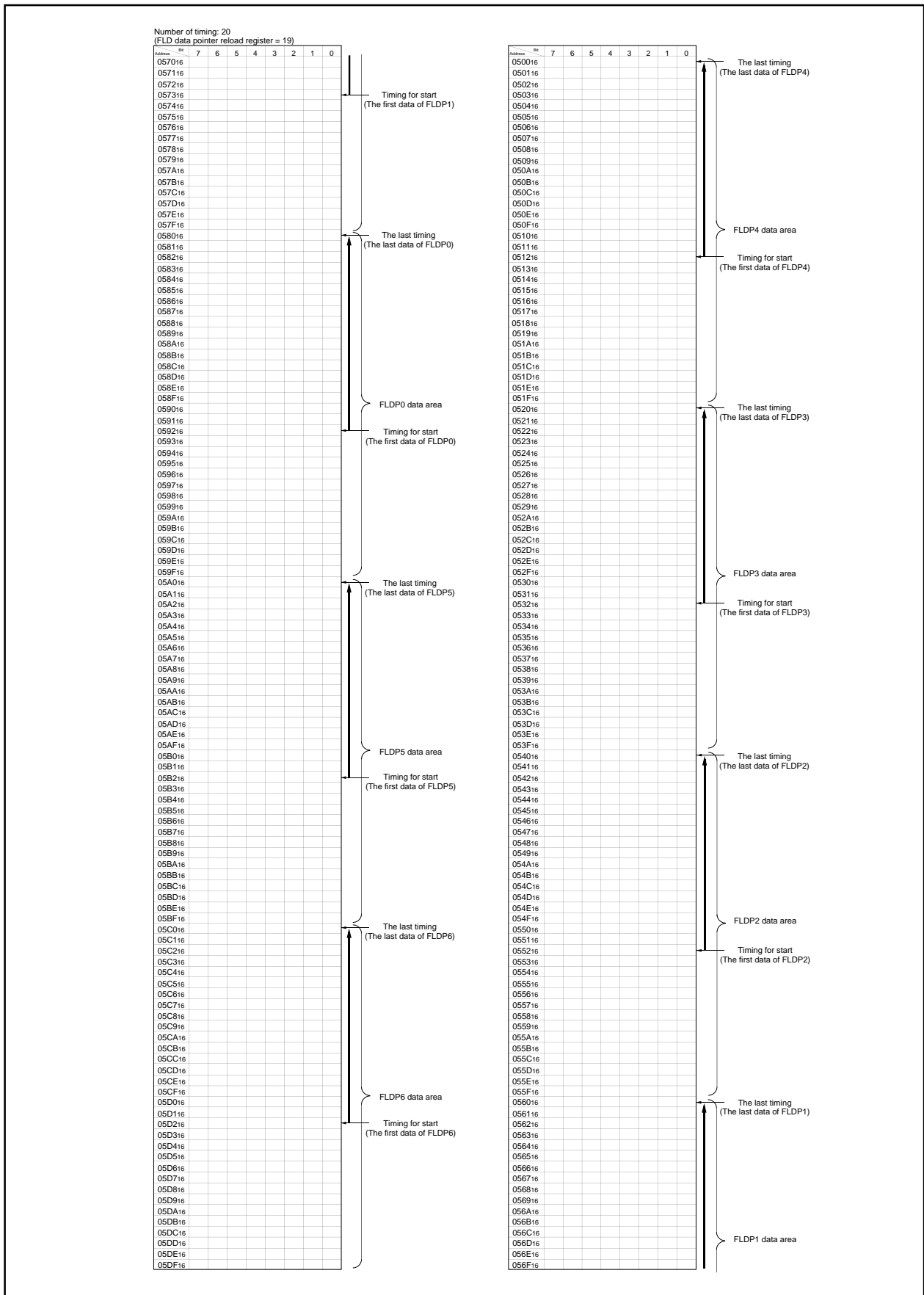


Figure KA-7. Example of Using the FLD Automatic Display RAM in 32-timing Mode



## Timing setting

Each timing is set by the FLDC mode register, Tdisp time set register, Toff1 time set register, and Toff2 time set register.

### •Tdisp time setting

The Tdisp time represents the length of display timing. In non-gradation display mode, it consists of a FLD display output period and a Toff1 time. In gradation display mode, it consists of the display output period and Toff1 time plus a low signal output period for dark display. Set the Tdisp time by the Tdisp counter count source select bit of the FLDC mode register and the Tdisp time set register. Supposing that the value of the Tdisp time set register is n, the Tdisp time is represented as  $T_{disp} = (n+1) \times t$  (t: count source). When the Tdisp counter count source select bit of the FLDC mode register is "0" and the value of the Tdisp time set register is 200 (C8<sub>16</sub>), the Tdisp time is:  $T_{disp} = (200+1) \times 3.2$  (at  $X_{IN} = 10$  MHz) = 643  $\mu$ s. When reading the Tdisp time set register, the value in the counter is read out.

### •Toff1 time setting

The Toff1 time represents a non-output (low signal output) time to prevent blurring of FLD, and to dim the display. Use the Toff1 time set register to set this Toff1 time. Make sure the value set to Toff1 is smaller than Tdisp and Toff2. Supposing that the value of the Toff1 time set register is n1, the Toff1 time is represented as  $T_{off1} = n1 \times t$ . When the Tdisp counter count source select bit of the FLDC mode register is "0" and the value of the Toff1 time set register is 30 (1E<sub>16</sub>),  $T_{off1} = 30 \times 3.2$  (at  $X_{IN} = 10$  MHz) = 96  $\mu$ s.

### •Toff2 time setting

The Toff2 time is provided for dark display. For bright display, the FLD display output remains effective until the counter that is counting Tdisp reaches the terminal count. For dark display, however, "L" (or "off") signal is output when the counter that is counting Toff2 reaches the terminal count. This Toff2 time setting is valid only for FLD ports which are in the gradation display mode and whose gradation display control RAM value is "1".

Set the Toff2 time by the Toff2 time set register. Make sure the value set to Toff2 is smaller than Tdisp but larger than Toff1. Supposing that the value of the Toff2 time set register is n2, the Toff2 time is represented as  $T_{off2} = n2 \times t$ . When the Tdisp counter count source select bit of the FLDC mode register is "0" and the value of the Toff2 time set register is 180 (B4<sub>16</sub>),  $T_{off2} = 180 \times 3.2$  (at  $X_{IN} = 10$  MHz) = 576  $\mu$ s.

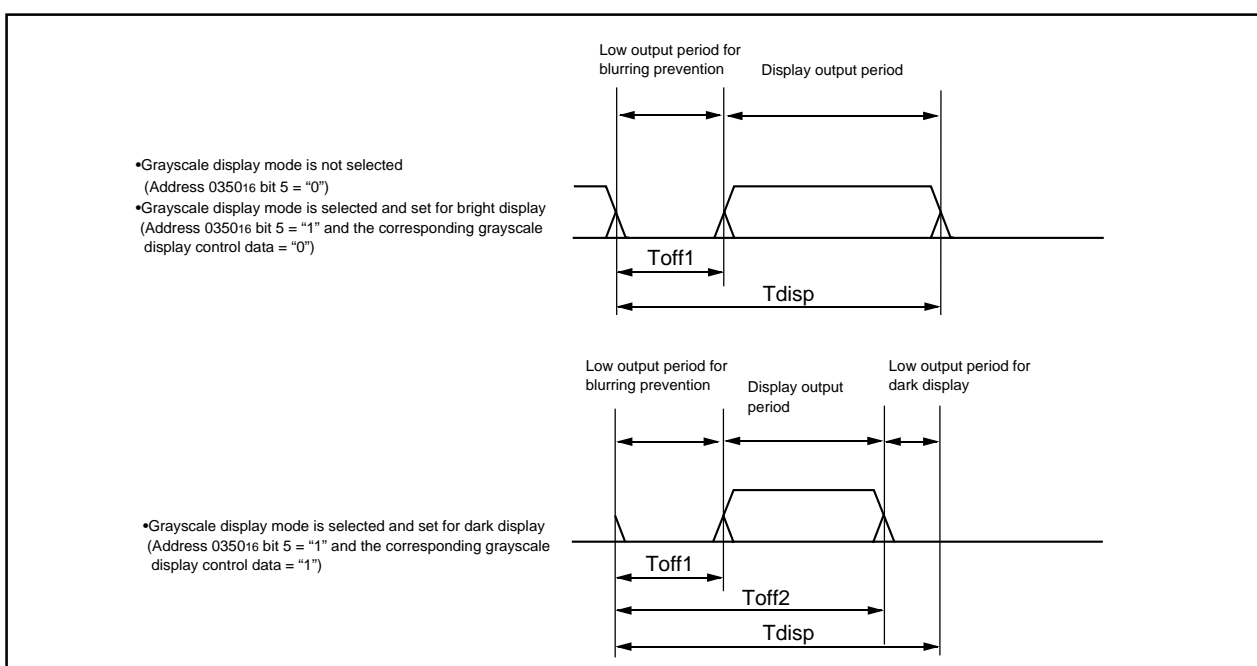


Figure KA-11. FLDC Timing

## FLD automatic display start

Automatic display starts by setting both the automatic display control bit (bit 0 of address 0350<sub>16</sub>) and the display start bit (bit 1 of address 0350<sub>16</sub>) to "1". The RAM content at a location apart from the start address of the automatic display RAM for each port by (FLD data pointer (address 0358<sub>16</sub>) – 1) is output to each port. The FLD data pointer (address 0358<sub>16</sub>) counts down in the T<sub>disp</sub> interval. When the count "FF<sub>16</sub>" is reached, the pointer is reloaded and starts counting over again. Before setting the display start bit (bit 1 of address 0350<sub>16</sub>) to "1", be sure to set the FLD/port switch register, FLD/DIG switch register, FLDC mode register, T<sub>disp</sub> time set register, T<sub>off1</sub> time set register, T<sub>off2</sub> time set register, and FLD data pointer. During FLD automatic display, bit 1 of the FLDC mode register (address 0350<sub>16</sub>) always keeps "1", and FLD automatic display can be interrupted by writing "0" to bit 1.

## Key-scan and interrupt

Either a FLD digit interrupt or FLD blanking interrupt can be selected using the Tscan control bits (bits 2, 3 of address 0350<sub>16</sub>).

The FLD digit interrupt is generated when the T<sub>off1</sub> time in each timing expires (at rising edge of digit output). Key scanning that makes use of FLD digits can be achieved using each FLD digit interrupt. To use FLD digit interrupts for key scanning, follow the procedure described below.

- (1) Read the port value each time the interrupt occurs.
- (2) The key is fixed on the last digit interrupt.

The digit positions output can be determined by reading the FLD data pointer (address 0358<sub>16</sub>).

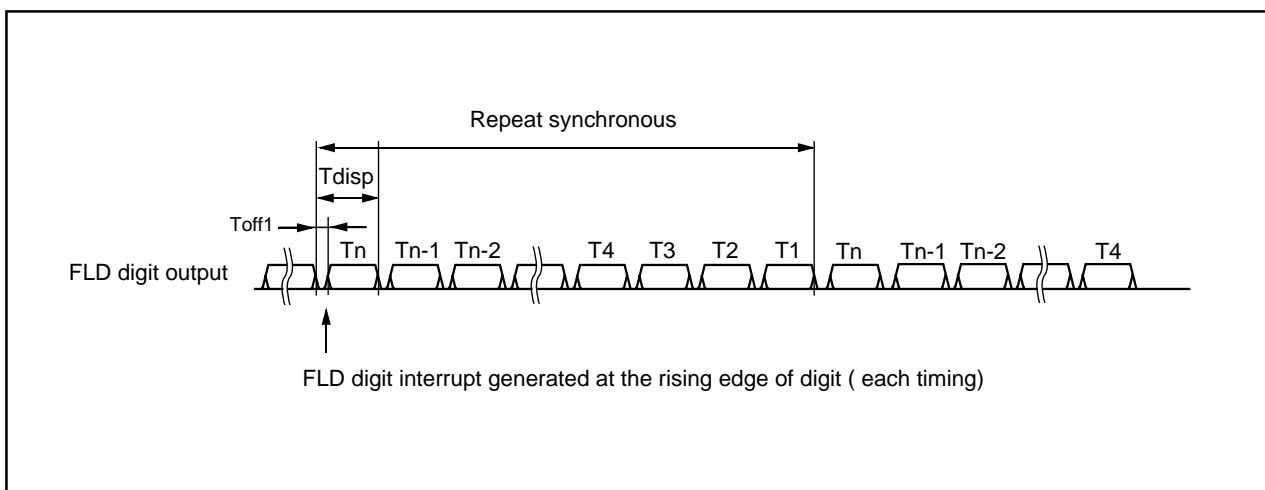


Figure KA-12A. Timing using digit interrupt

## FLD controller

The FLD blanking interrupt is generated when the FLD data pointer (address 0358<sub>16</sub>) reaches "FF<sub>16</sub>". The FLD automatic display output is turned off for a duration of 1 x  $T_{disp}$ , 2 x  $T_{disp}$ , or 3 x  $T_{disp}$  depending on post-interrupt settings. During this time, key scanning that makes use of FLD segments can be achieved.

When a key-scan is performed with the segment during key-scan blanking period  $T_{scan}$ , take the following sequence:

1. Write "0" to bit 0 of the FLDC mode register (address 0350<sub>16</sub>).
2. Set the port corresponding to the segment for key-scan to the output port.
3. Perform the key-scan.
4. After the key-scan is performed, write "1" to bit 0 of FLDC mode register (address 0350<sub>16</sub>).

•**Note:**

When performing a key-scan according to the above steps 1 to 4, take the following points into consideration.

1. Do not set "0" in bit 1 of the FLDC mode register (address 0350<sub>16</sub>).
2. Do not set "1" in the ports corresponding to digits.

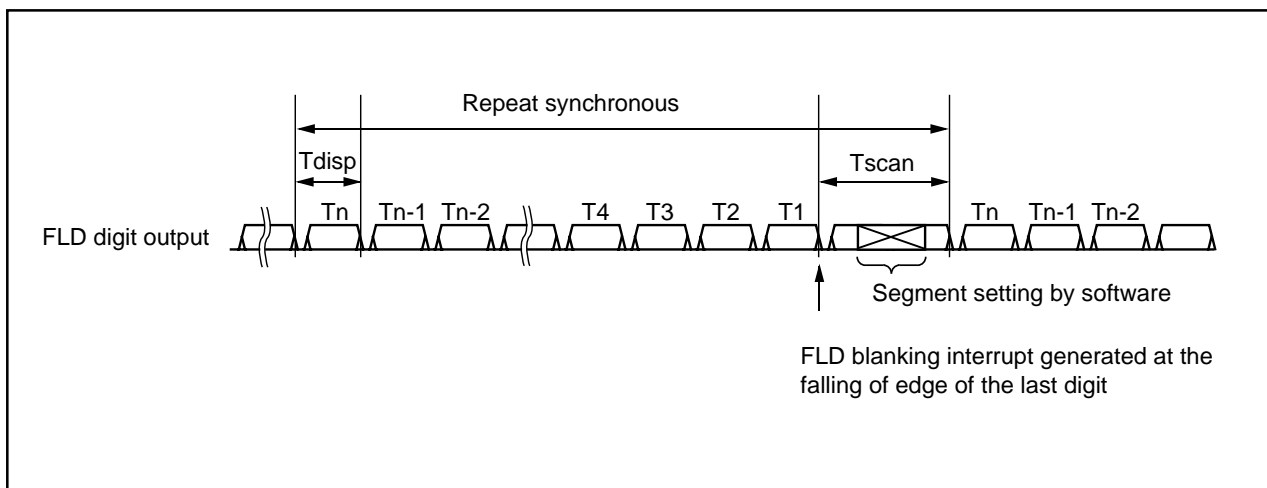


Figure KA-12B. Timing using FLD blanking interrupt

### P44 to P47 Expansion Function

P44 to P47 are CMOS output-type ports. FLD digit outputs can be increased as many as 16 lines by connecting a 4-bit to 16-bit decoder to these ports. P44 to P47 have the function to allow for connection to a 4-bit to 16-bit decoder.

#### (1) P44 to P47 Toff invalid Function

This function disables the Toff1 time and Toff2 time and outputs display data for the duration of Tdisp. (See Figure KA-13.) This can be accomplished by setting the P44 to P47 Toff disable bit (address 035016 bit 2) to "1".

Unlike the Toff section generate/not generate function, this function disables all display data.

#### (2) Dimmer signal output Function

This function allows a dimmer signal creation signal to be output from DIMOUT (P97). The dimmer function can be materialized by controlling the decoder with this signal. (See Figure KA-13.) This function can be set by writing P97 dimmer output control bit (bit 4 of address 035116) to "1".

#### (3) P44 to P47 FLD Output Reverse Bit

P44 to P47 are provided with a function to reverse the polarity of the FLD output. This function is useful in adjusting the polarity when using an externally installed driver.

The output polarity can be reversed by setting bit 0 of the FLD output control register (address 035116) to "1".

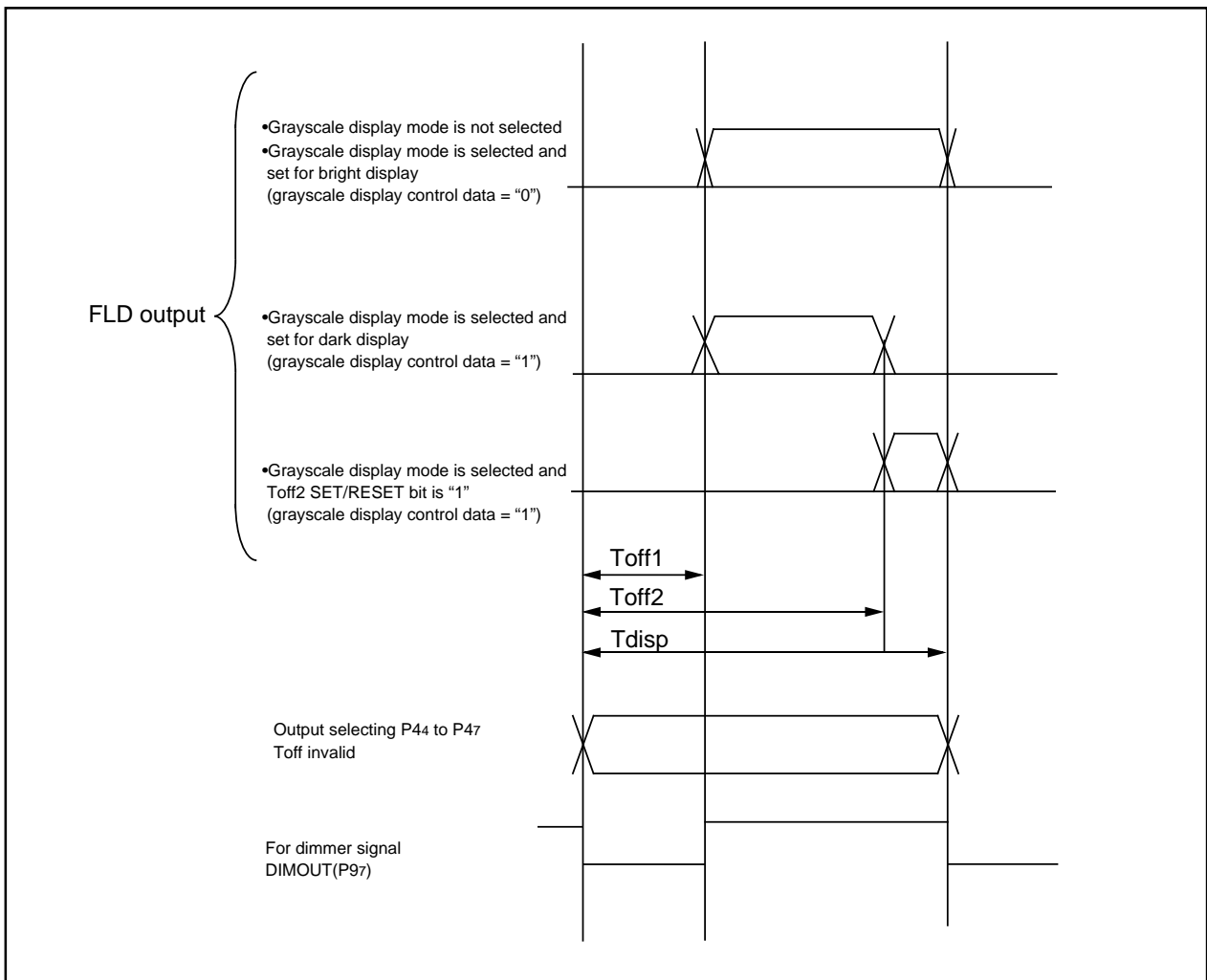


Figure KA-13. P4 to P47 FLD Output pulses

## Toff section generate/not generate Function

The function is for reduction of useless noises which generated as every switching of ports, because of the combined capacity of among FLD ports. In case the continuous data output to each FLD ports, the Toff1 section of the continuous parts is not generated. (See Figure KA-15)

If it needs Toff1 section on FLD pulses, set "CMOS ports: section of Toff generate / not generate bit" to "1" and set "high-breakdown-voltage ports: section of Toff generate / not generate bit" to "1". High-breakdown-voltage ports (P5, P6, P3, P2, P1, P0, P40 to P43, total 52 pins) generate Toff1 section, by setting "high-breakdown-voltage ports: section of Toff generate / not generate bit" to "1".

The CMOS ports ( P44 to P47, total 4 pins ) generate Toff1 section, by setting "high-breakdown-voltage ports: section of Toff generate / not generate bit" to "1".

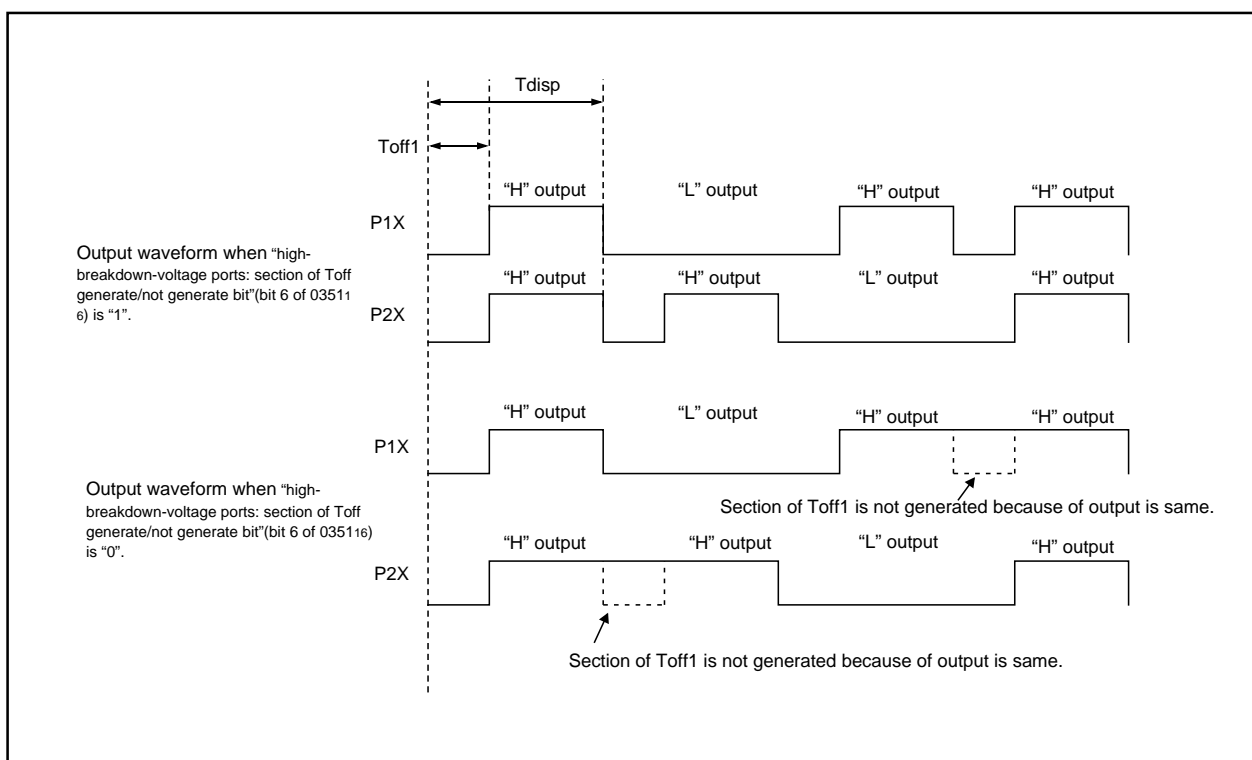


Fig. KA-15. Toff Section Generated/not generated Function

## Toff2 SET/RESET change bit

In gradation display mode, the values set by the Toff2 time set register (TOFF2) are effective. When the FLD output control register (bit 7 of address 035116) in the initial state = "0", RAM data is output to the FLD output ports (SET) at the time that is set by TOFF1 and is turned to "0" (RESET) at the time that is set by TOFF2. When bit 7 = "1", RAM data is output (SET) at the time that is set by TOFF2 and is turned to "0" (RESET) when the Tdisp time expires.

### Digit pulses output Function

P50 to P57 and P60 to P67 allow digit pulses to be output using the FLD/digit switch register. Set the digit output set register by writing as many consecutive 1s as the timing count from P60. The contents of FLD automatic display RAM for the ports that have been selected for digit output are disabled, and the pulse shown in Figure KA-16 is output automatically. In gradation display mode use,  $T_{off2}$  time becomes effective for the port which selected digit output. Because the contents of FLD automatic display RAM are disabled, the segment data can be changed easily even when segment data and digit data coexist at the same address in the FLD automatic display RAM.

This function is effective in 16-timing normal mode and 16-timing gradation display mode. If a value is set exceeding the timing count (FLD data pointer reload register's set value + 1) for any port, the output of such port is "L".

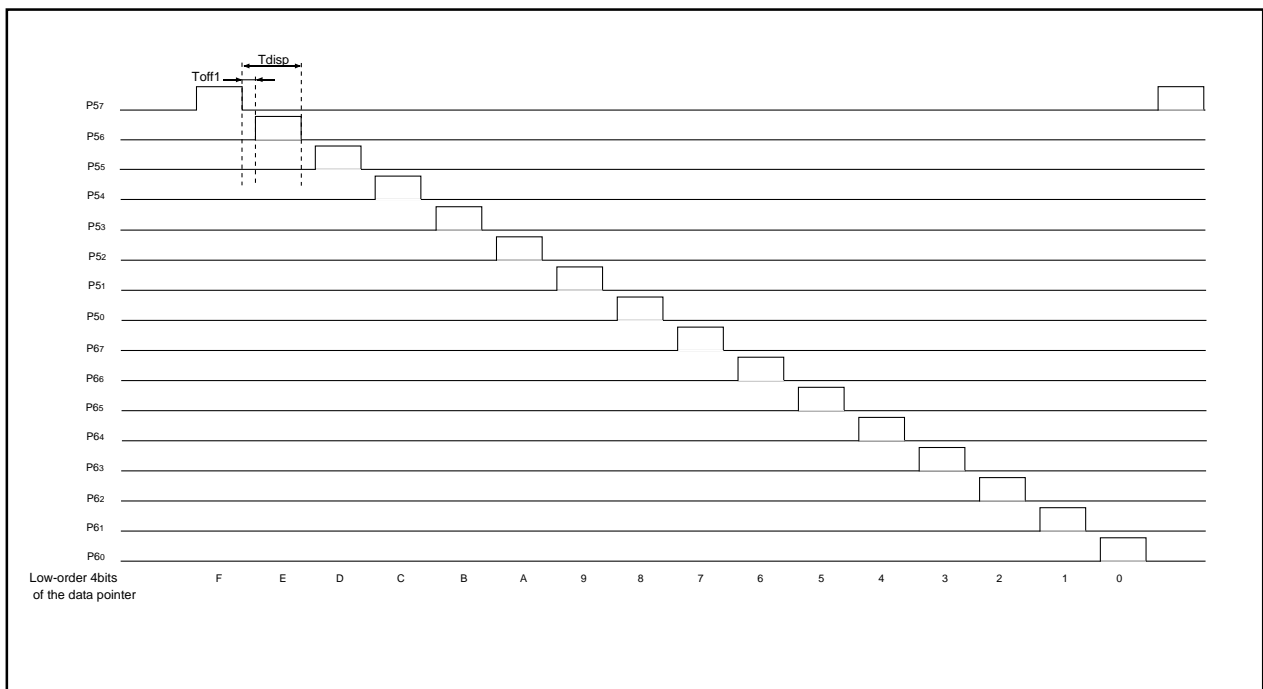


Fig. KA-16. Digit Pulses Output Function

Timer

Timer

There are eight 16-bit timers. These timers can be classified by function into timers A (five) and timers B (three). All these timers function independently. Figures FB-1 show the block diagram of timers.

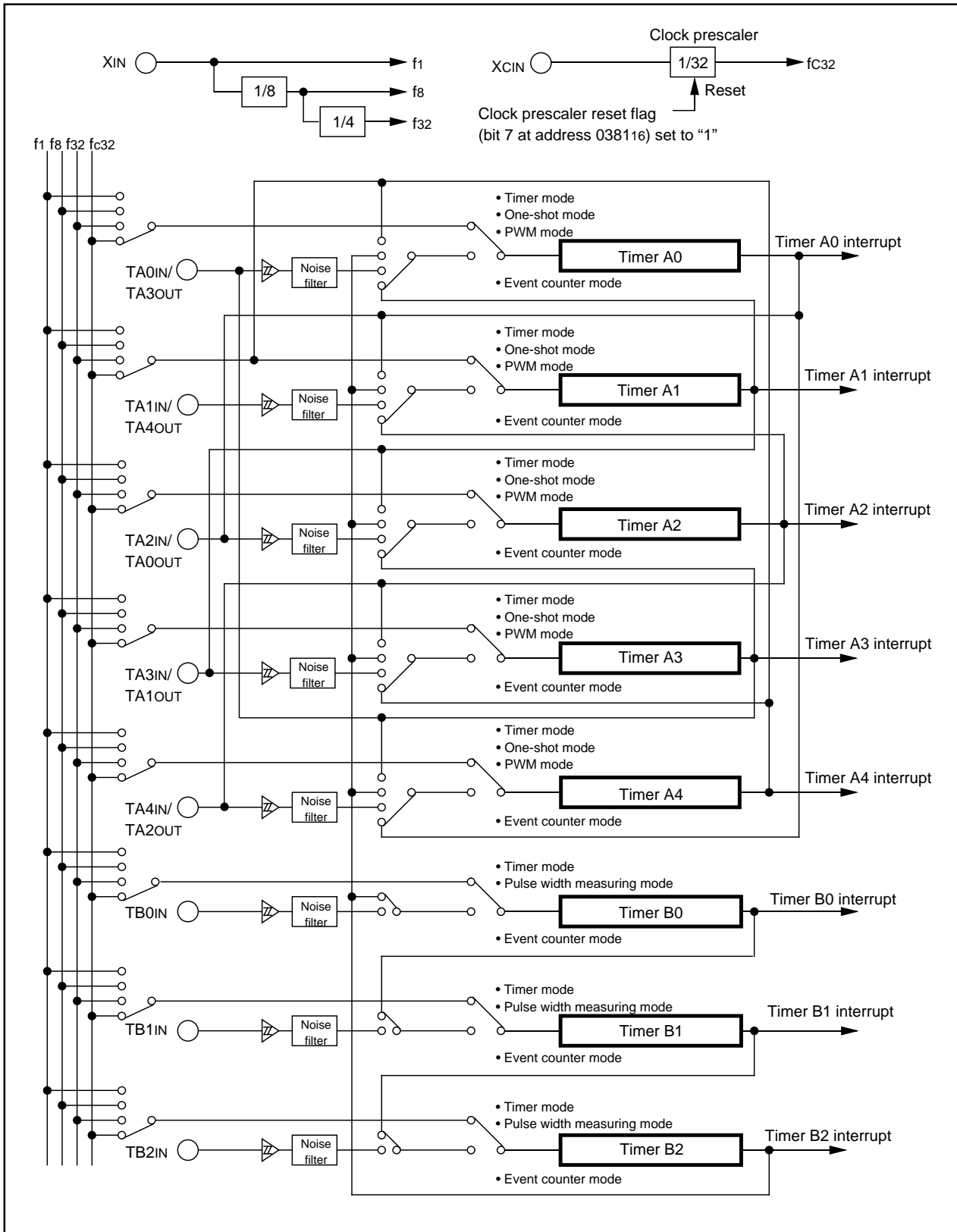


Figure FB-1. Timer block diagram

### Timer A

Figure FB-2 shows the block diagram of timer A. Figures FB-3 to FB-5 show the timer A-related registers. Except in event counter mode, timers A0 through A4 all have the same function. Use the timer Ai mode register (i = 0 to 4) bits 0 and 1 to choose the desired mode.

Timer A has the four operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer's over flow.
- One-shot timer mode: The timer stops counting when the count reaches "0000<sub>16</sub>".
- Pulse width modulation (PWM) mode: The timer outputs pulses of a given width.

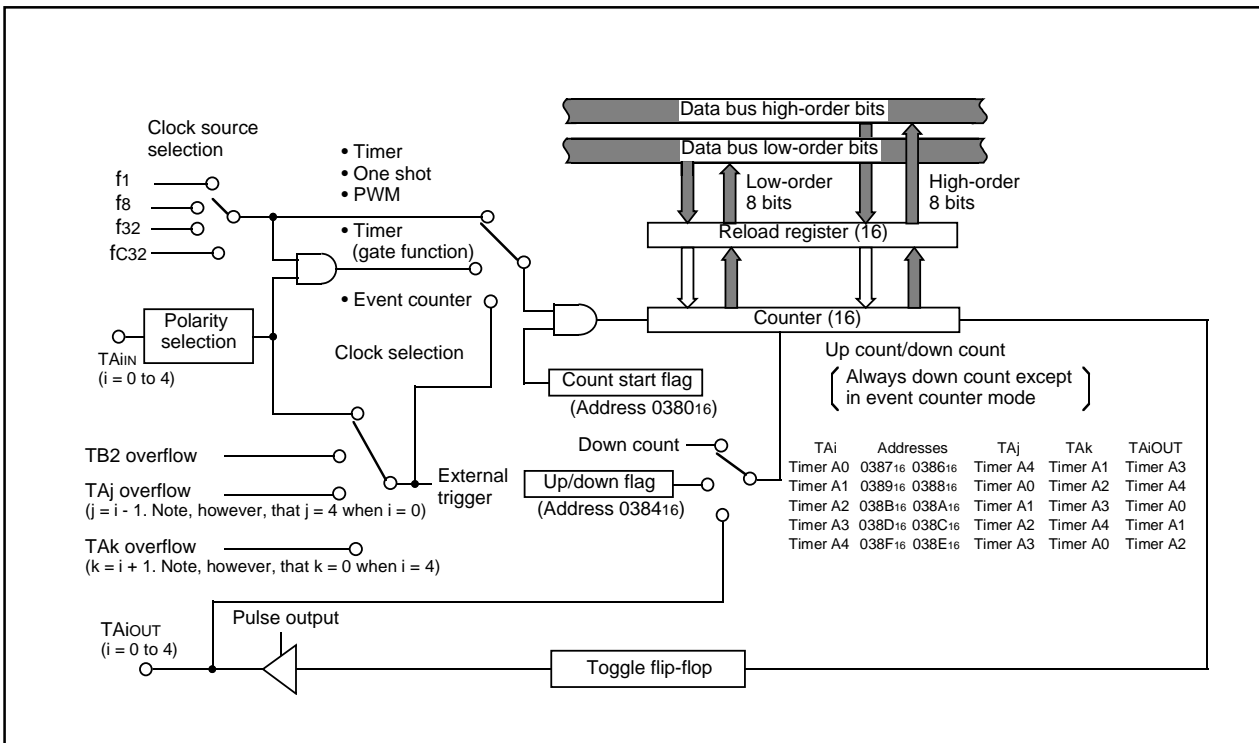


Figure FB-2. Block diagram of timer A

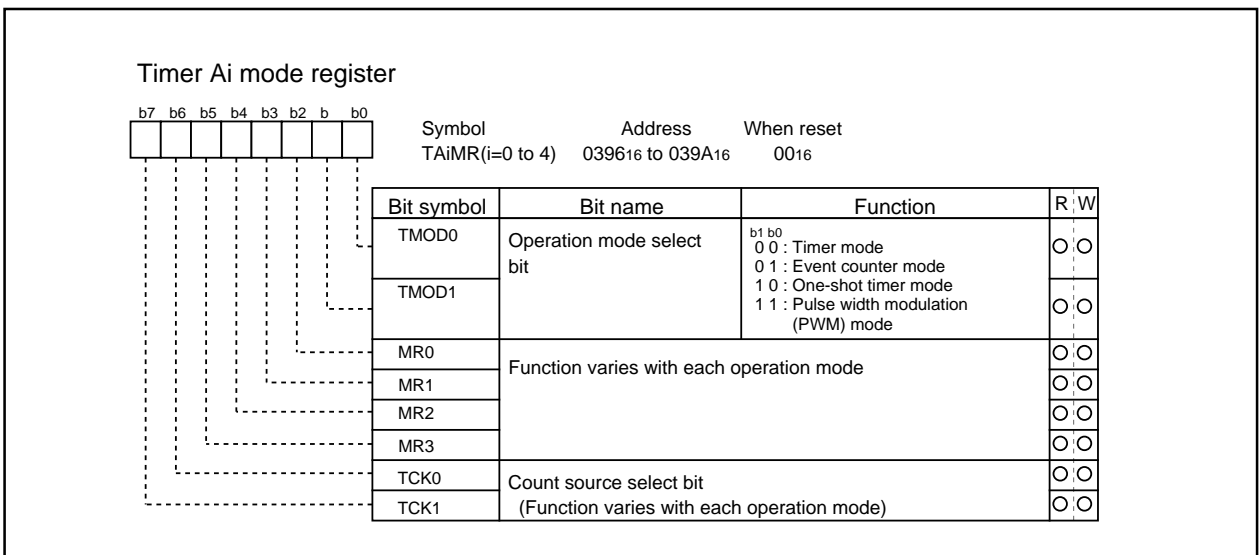


Figure FB-3. Timer A-related registers (1)



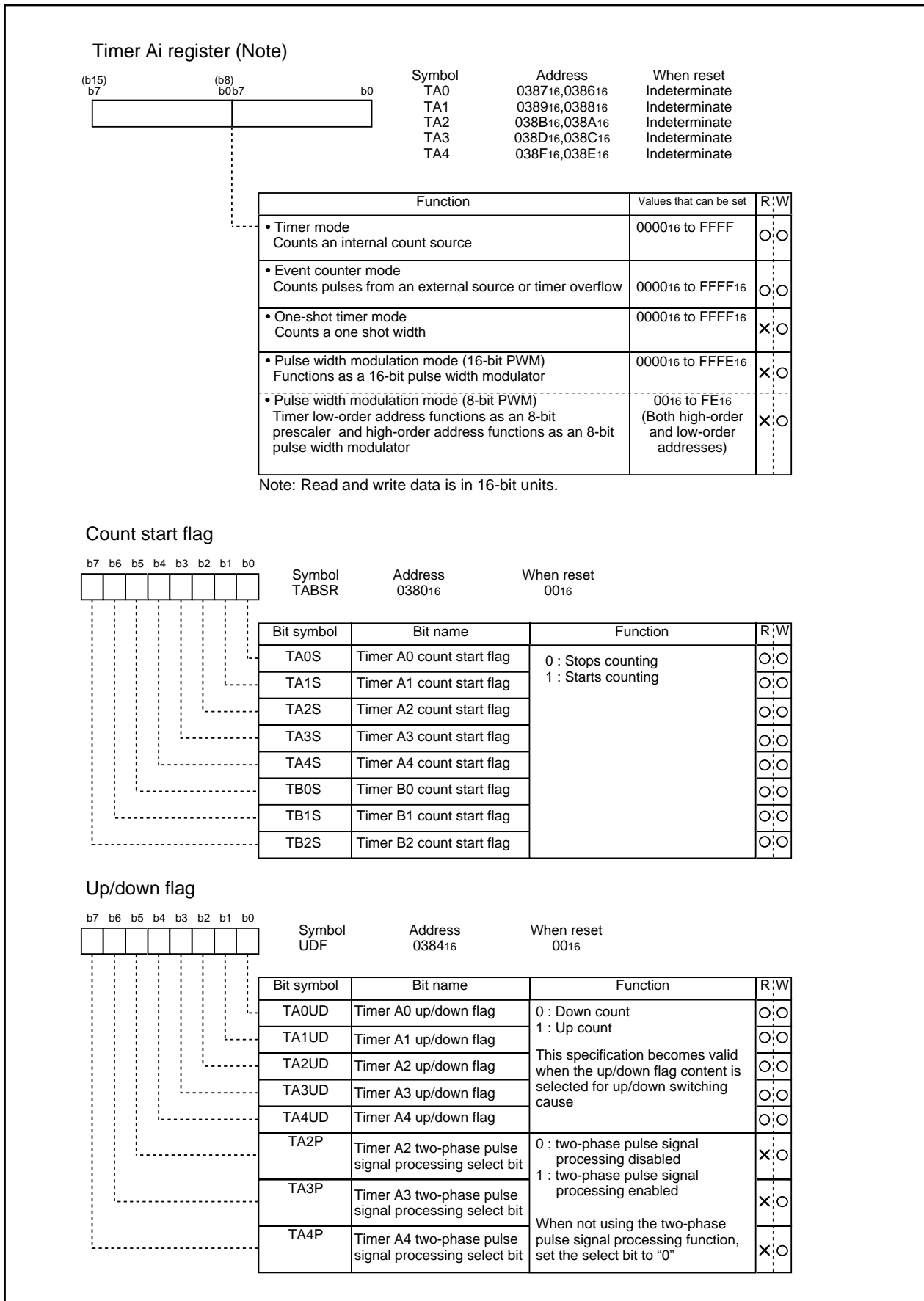


Figure FB-4. Timer A-related registers (2)

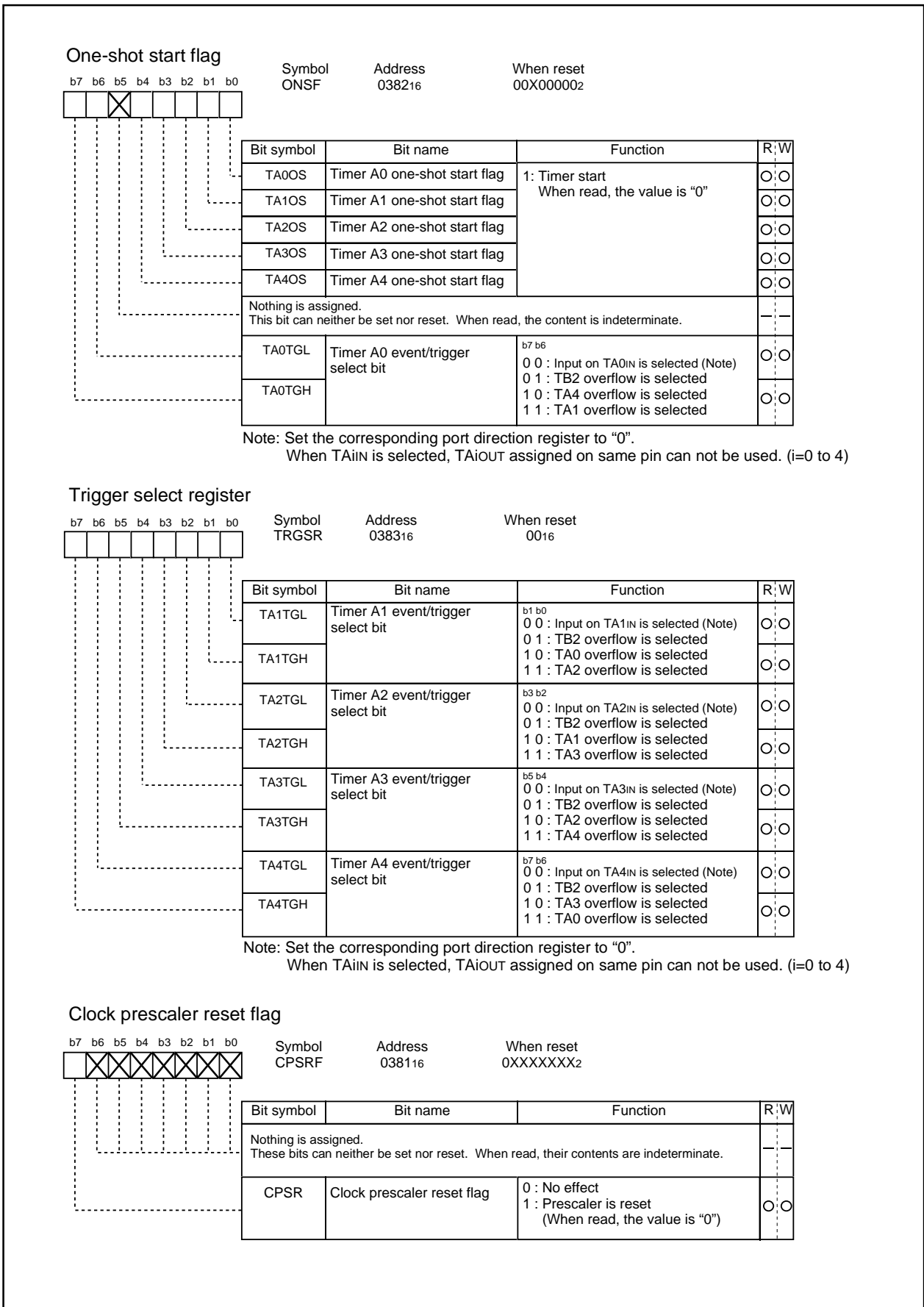


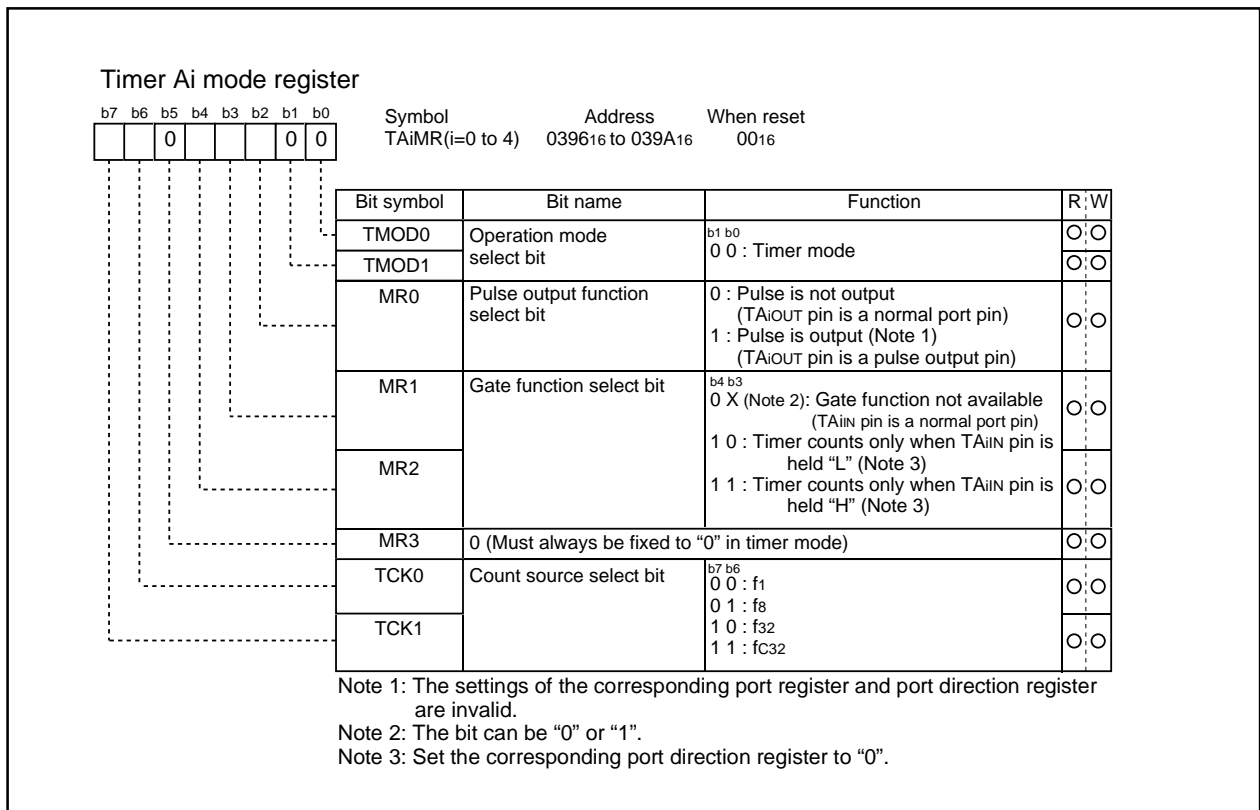
Figure FB-5. Timer A-related registers (3)

**(1) Timer mode**

In this mode, the timer counts an internally generated count source. (See Table FB-1.) Figure FB-6 shows the timer Ai mode register in timer mode.

**Table FB-1. Specifications of timer mode**

Item	Specification
Count source	f1, f8, f32, fC32
Count operation	<ul style="list-style-type: none"> <li>Down count</li> <li>When the timer underflows, it reloads the reload register contents before continuing counting</li> </ul>
Divide ratio	1/(n+1)    n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	When the timer underflows
TAiIN pin function	Programmable I/O port or gate input
TAiOUT pin function	Programmable I/O port or pulse output
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter</li> <li>When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>Gate function Counting can be started and stopped by the TAiIN pin's input signal</li> <li>Pulse output function Each time the timer underflows, the TAiOUT pin's polarity is reversed</li> </ul>



**Figure FB-6. Timer Ai mode register in timer mode**

**(2) Event counter mode**

In this mode, the timer counts an external signal or an internal timer's overflow. Timers A0 and A1 can count a single-phase external signal. Timers A2, A3, and A4 can count a single-phase and a two-phase external signal. Table FB-2 lists timer specifications when counting a single-phase external signal. Figure FB-7 shows the timer Ai mode register in event counter mode.

Table FB-3 lists timer specifications when counting a two-phase external signal. Figure FB-8 shows the timer Ai mode register in event counter mode.

Table FB-2. Timer 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 TAIiN pin (effective edge can be selected by software)</li> <li>TB2 overflow, TAj overflow</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>Up count or down count can be selected by external signal or software</li> <li>When the timer overflows or underflows, the reload register's content is reloaded and the timer starts over again.(Note)</li> </ul>
Divide ratio	$1 / (FFFF_{16} - n + 1)$ for up count $1 / (n + 1)$ for down count                      n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer overflows or underflows
TAiIN pin function	Programmable I/O port or count source input
TAiOUT pin function	Programmable I/O port, pulse output, or up/down count select input
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter</li> <li>When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)</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 Each time the timer overflows or underflows, the TAIOUT pin's polarity is reversed</li> </ul>

Note: This does not apply when the free-run function is selected.

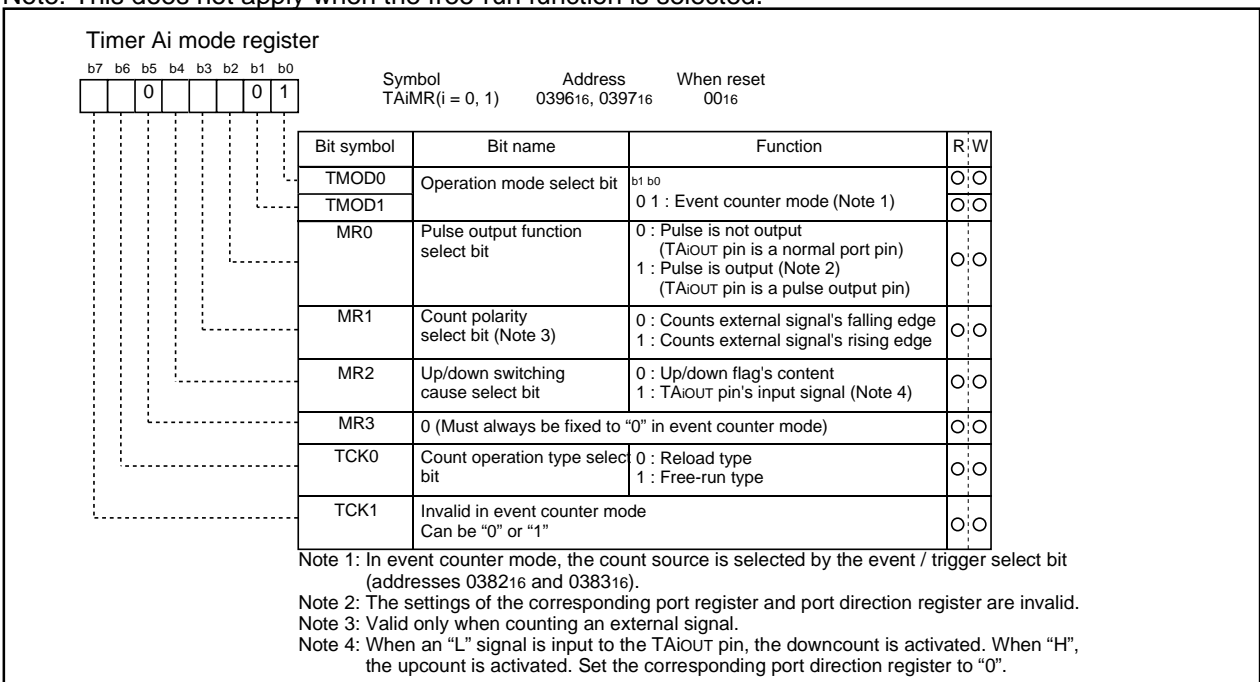


Figure FB-7. Timer Ai mode register in event counter mode



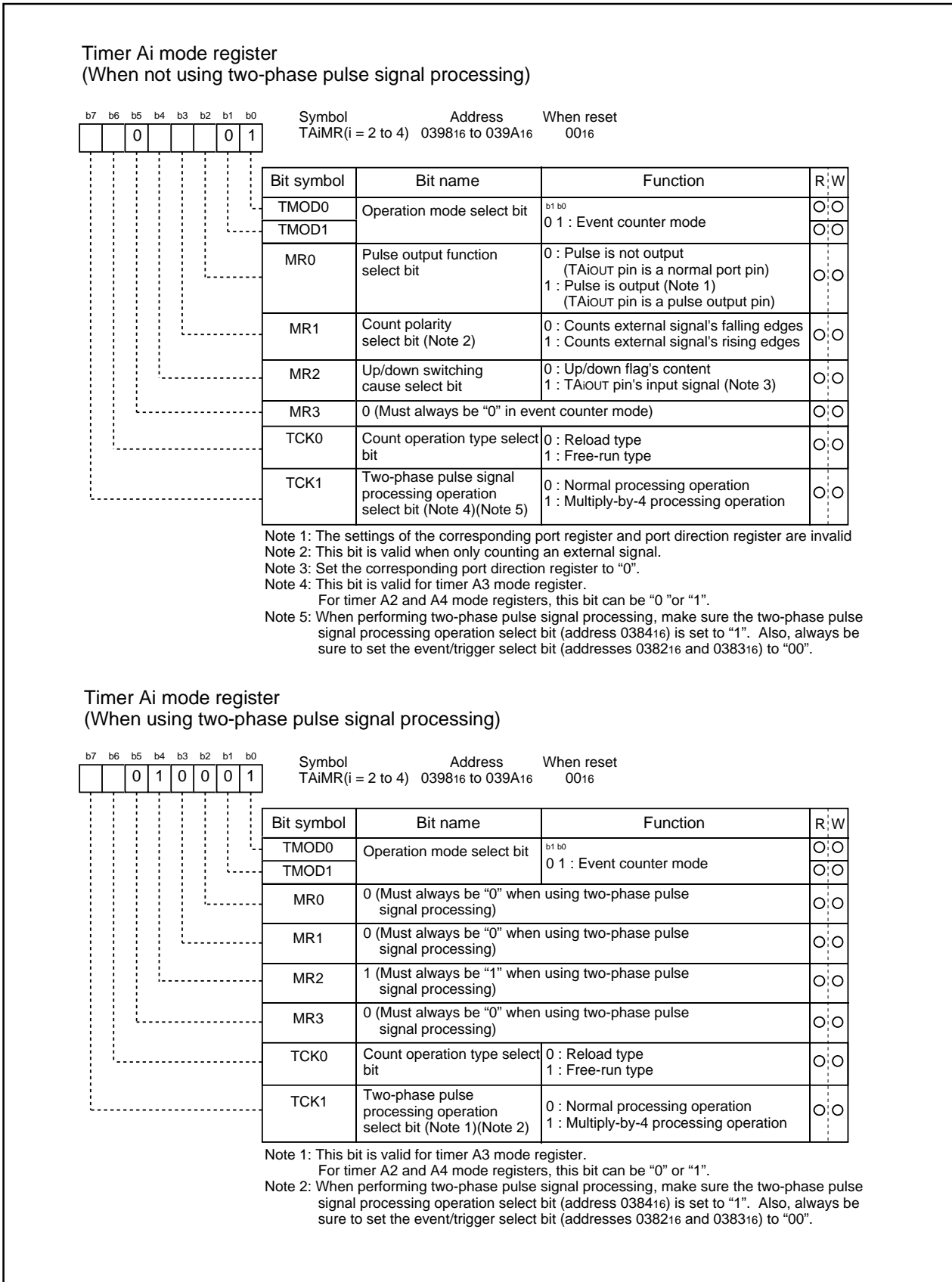


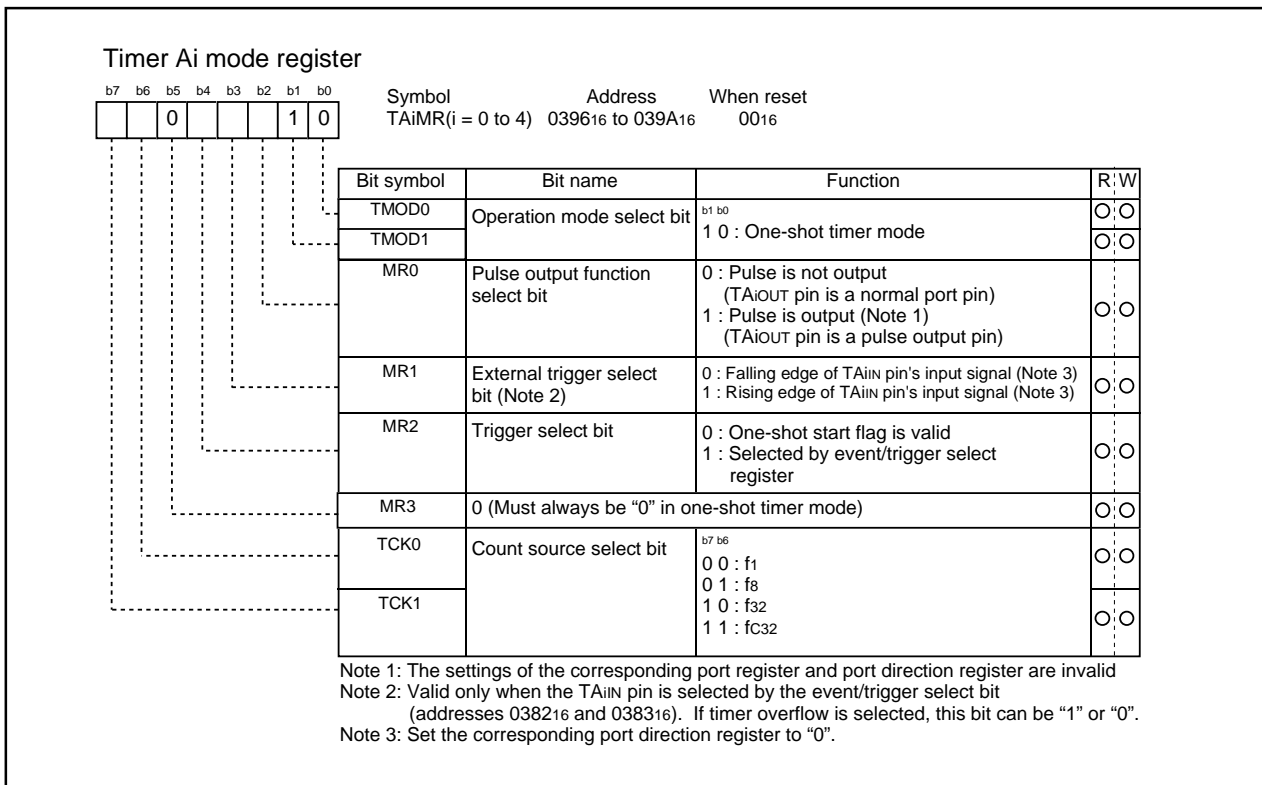
Figure FB-8. Timer Ai mode register in event counter m

### (3) One-shot timer mode

In this mode, the timer operates only once. (See Table FB-4.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure FB-9 shows the timer Ai mode register in one-shot timer mode.

**Table FB-4. Timer specifications in one-shot timer mode**

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>The timer counts down</li> <li>When the count reaches 0000<sub>16</sub>, the timer stops counting after reloading a new count</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
Count start condition	<ul style="list-style-type: none"> <li>An external trigger is input</li> <li>The timer overflows</li> <li>The one-shot start flag is set (= 1)</li> </ul>
Count stop condition	<ul style="list-style-type: none"> <li>A new count is reloaded after the count has reached 0000<sub>16</sub></li> <li>The count start flag is reset (= 0)</li> </ul>
Interrupt request generation timing	The count reaches 0000 <sub>16</sub>
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Programmable I/O port or pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter</li> <li>When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)</li> </ul>



**Figure FB-9. Timer Ai mode register in one-shot timer mode**

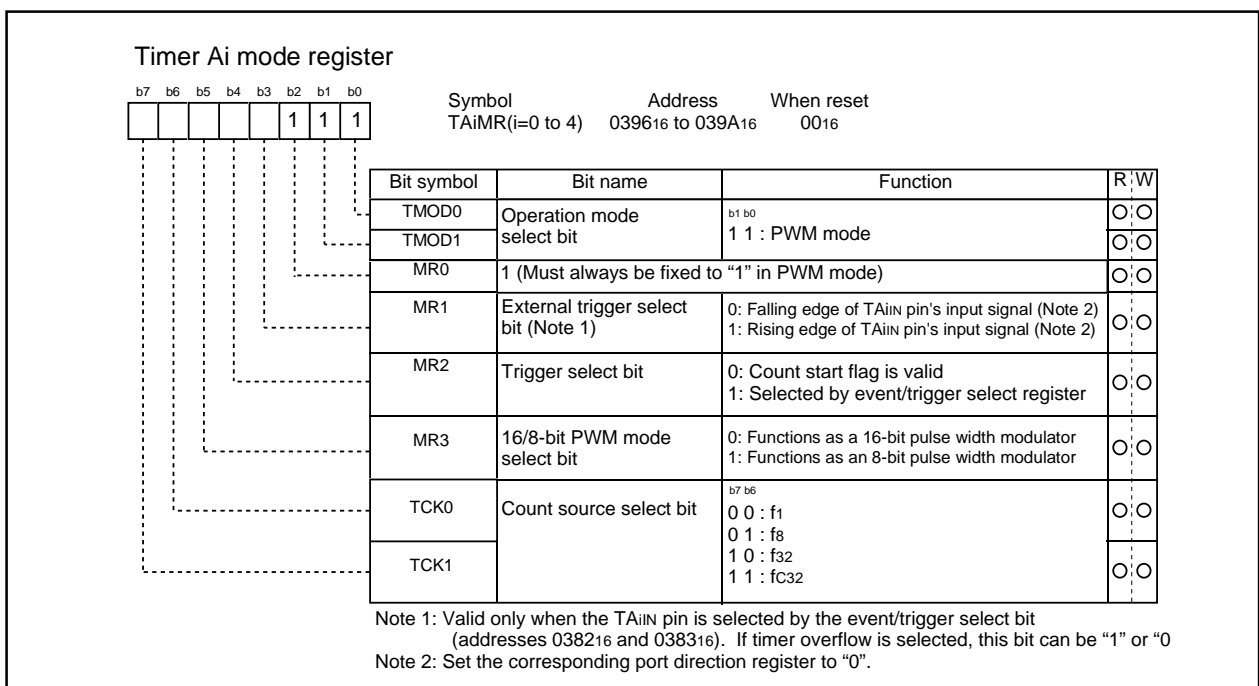


### (4) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table FB-5.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure FB-10 shows the timer Ai mode register in pulse width modulation mode. Figure FB-11 shows the example of how a 16-bit pulse width modulator operates. Figure FB-12 shows the example of how an 8-bit pulse width modulator operates.

**Table FB-5. Timer specifications in pulse width modulation mode**

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>•The timer counts down (operating as an 8-bit or a 16-bit pulse width modulator)</li> <li>•The timer reloads a new count at a rising edge of PWM pulse and continues counting</li> <li>•The timer is not affected by a trigger that occurs when counting</li> </ul>
16-bit PWM	<ul style="list-style-type: none"> <li>•High level width <math>n / f_i</math> n : Set value</li> <li>•Cycle time <math>(2^{16}-1) / f_i</math> fixed</li> </ul>
8-bit PWM	<ul style="list-style-type: none"> <li>•High level width <math>n \times (m+1) / f_i</math> n : values set to timer Ai register's high-order address</li> <li>•Cycle time <math>(2^8-1) \times (m+1) / f_i</math> m : values set to timer Ai register's low-order address</li> </ul>
Count start condition	<ul style="list-style-type: none"> <li>•External trigger is input</li> <li>•The timer overflows</li> <li>•The count start flag is set (= 1)</li> </ul>
Count stop condition	•The count start flag is reset (= 0)
Interrupt request generation timing	PWM pulse goes "L"
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	<ul style="list-style-type: none"> <li>•When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter</li> <li>•When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)</li> </ul>



**Figure FB-10. Timer Ai mode register in pulse width modulation mode**



Timer A

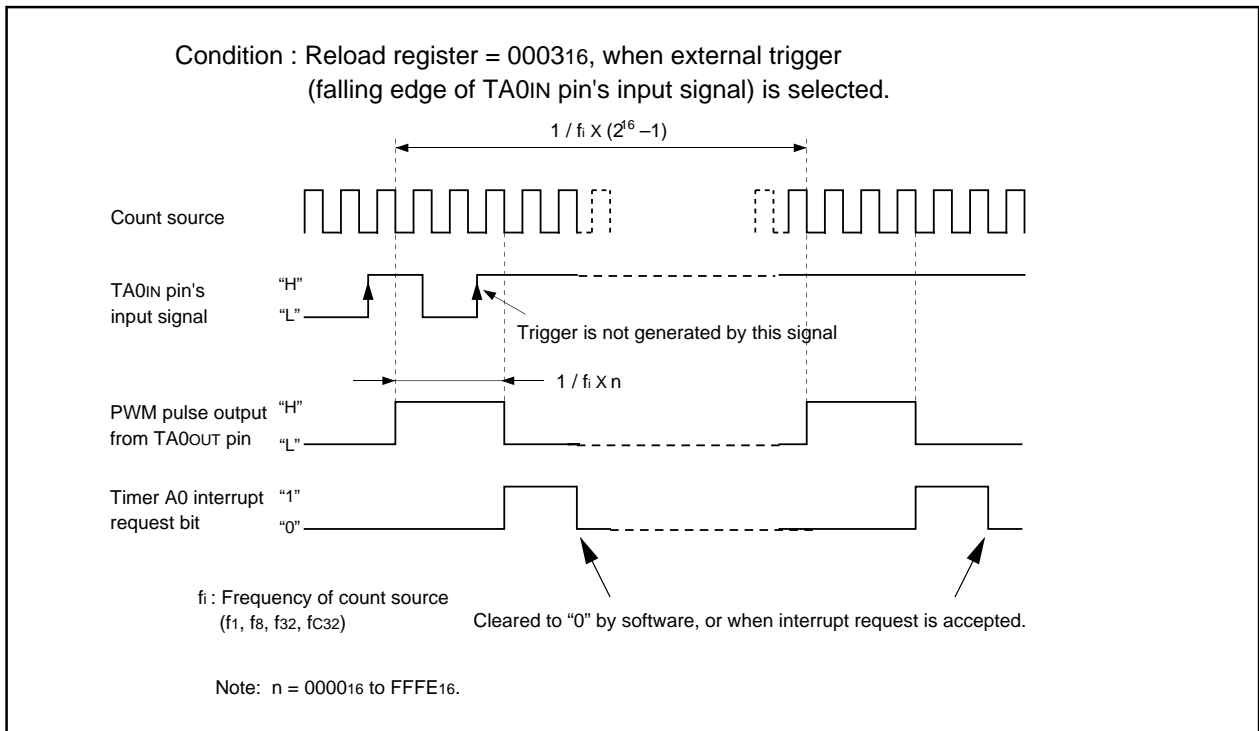


Figure FB-11. Example of how a 16-bit pulse width modulator operates

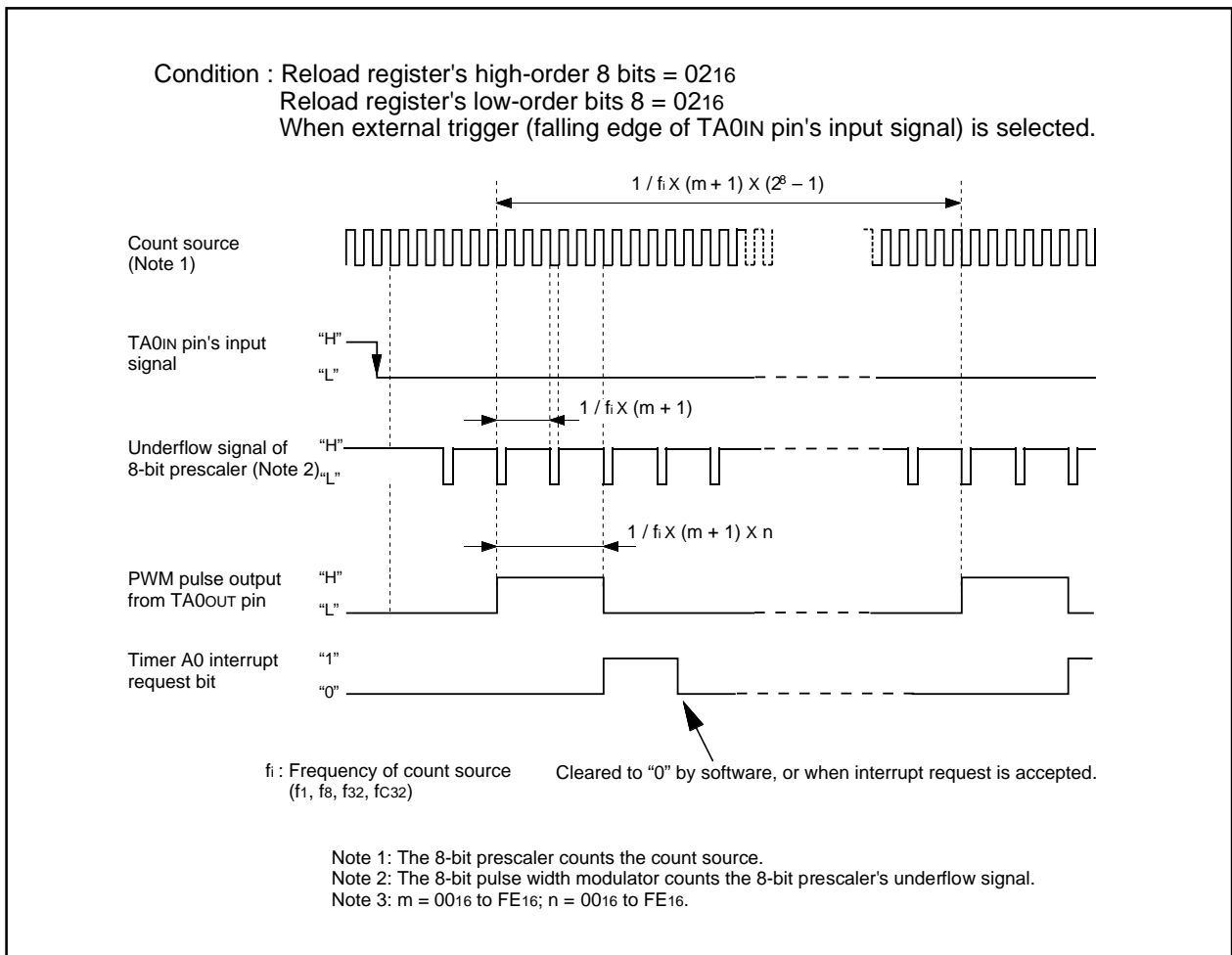


Figure FB-12. Example of how an 8-bit pulse width modulator operates

### Timer B

Figure TA-1 shows the block diagram of timer B. Figures TA-2 and TA-3 show the timer B-related registers. Use the timer Bi mode register (i = 0 to 2) bits 0 and 1 to choose the desired mode.

Timer B has three operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer overflow.
- Pulse period/pulse width measuring mode: The timer measures an external signal's pulse period or pulse width.

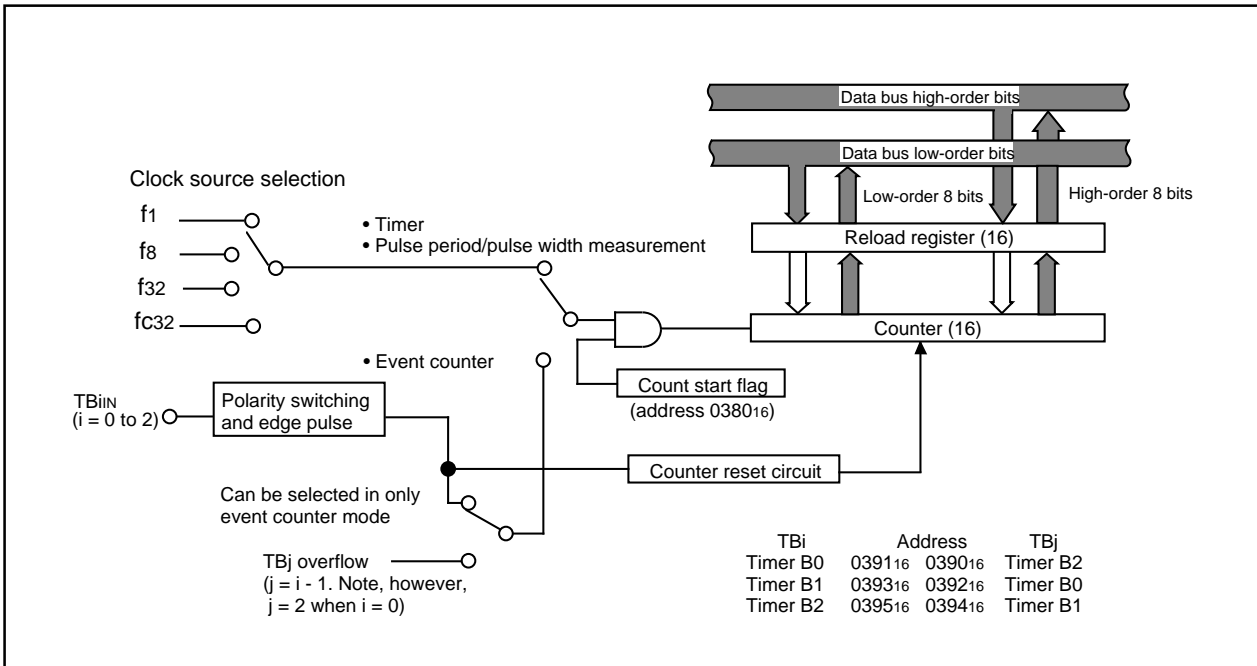


Figure TA-1. Block diagram of timer B

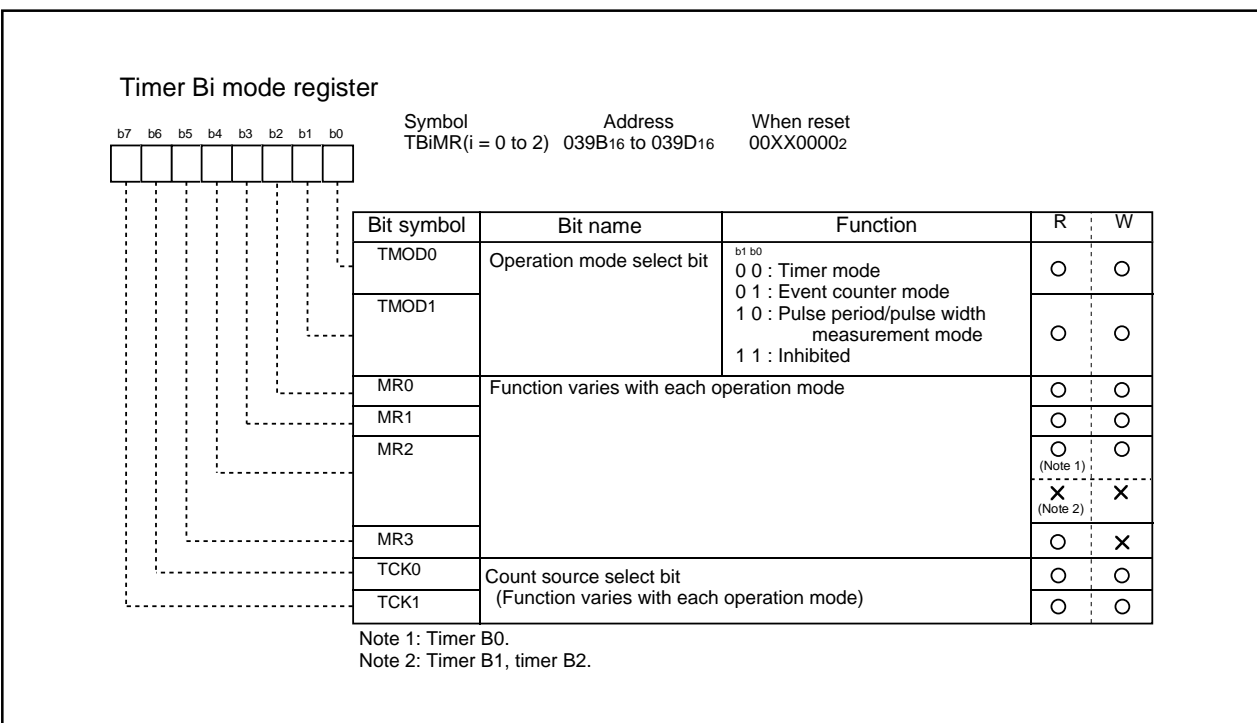


Figure TA-2. Timer B-related registers (1)

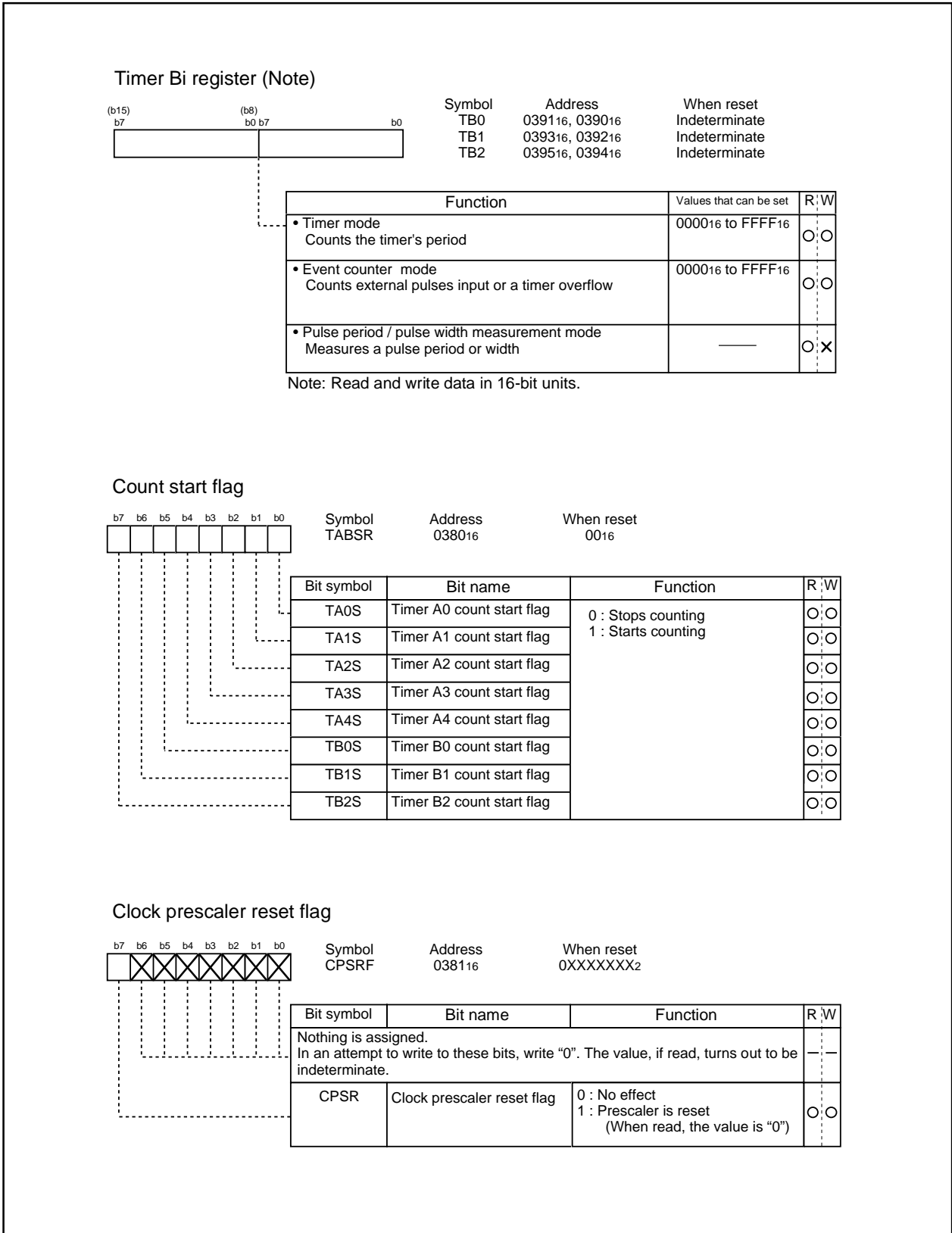


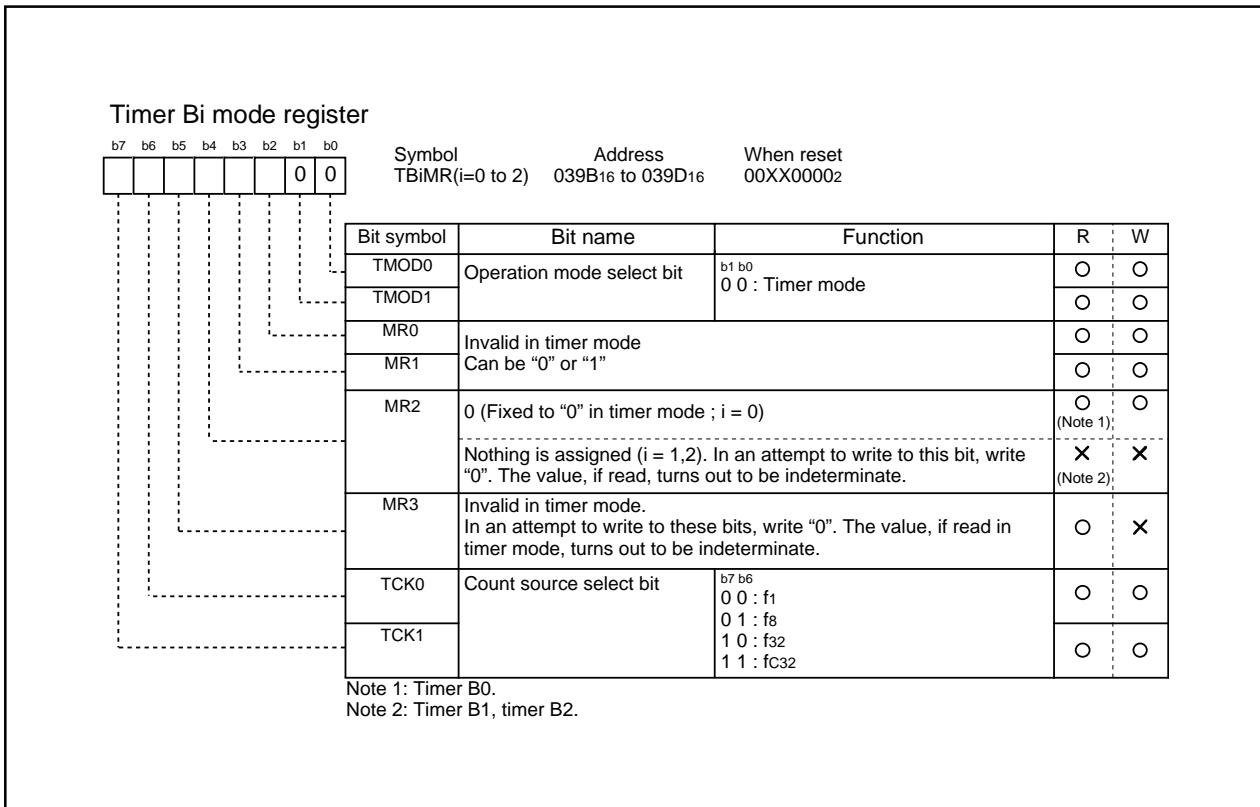
Figure TA-3. Timer B-related registers (2)

**(1) Timer mode**

In this mode, the timer counts an internally generated count source. (See Table TA-1.) Figure TA-4 shows the timer Bi mode register in timer mode.

**Table TA-1. Timer specifications in timer mode**

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>Counts down</li> <li>When the timer underflows, the reload register's content is reloaded and the timer starts over again.</li> </ul>
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Programmable I/O port
Read from timer	Count value is read out by reading timer Bi register
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter</li> <li>When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)</li> </ul>



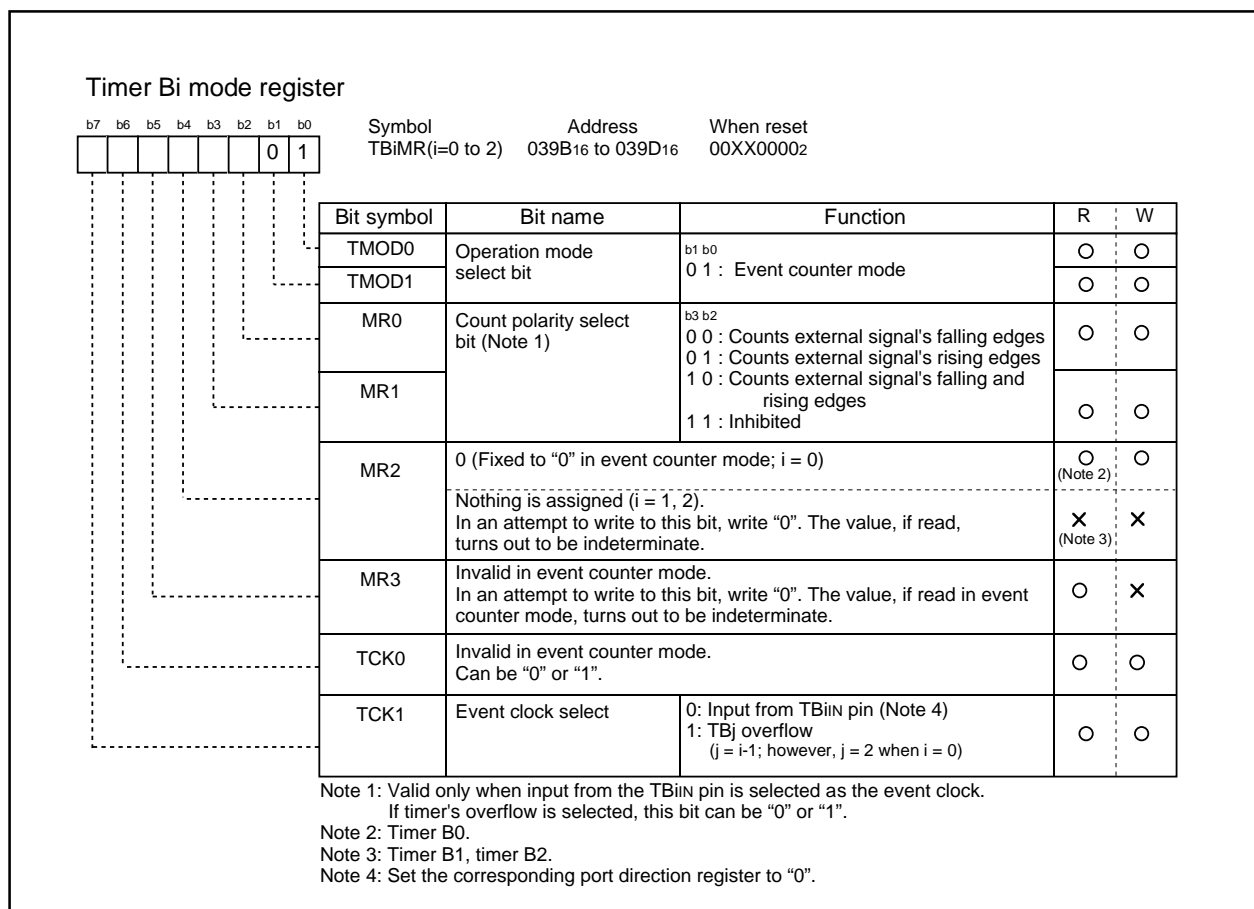
**Figure TA-4. Timer Bi mode register in timer mode**

## (2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table TA-2.) Figure TA-5 shows the timer Bi mode register in event counter mode.

**Table TA-2. Timer specifications in event counter mode**

Item	Specification
Count source	<ul style="list-style-type: none"> <li>External signals input to TBIIN pin</li> <li>Effective edge of count source can be a rising edge, a falling edge, or falling and rising edges as selected by software</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>Counts down</li> <li>When the timer underflows, it reloads the reload register contents before continuing counting</li> </ul>
Divide ratio	1/(n+1)      n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBIIN pin function	Count source input
Read from timer	Count value can be read out by reading timer Bi register
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter</li> <li>When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)</li> </ul>



**Figure TA-5. Timer Bi mode register in event counter mode**

### (3) Pulse period/pulse width measurement mode

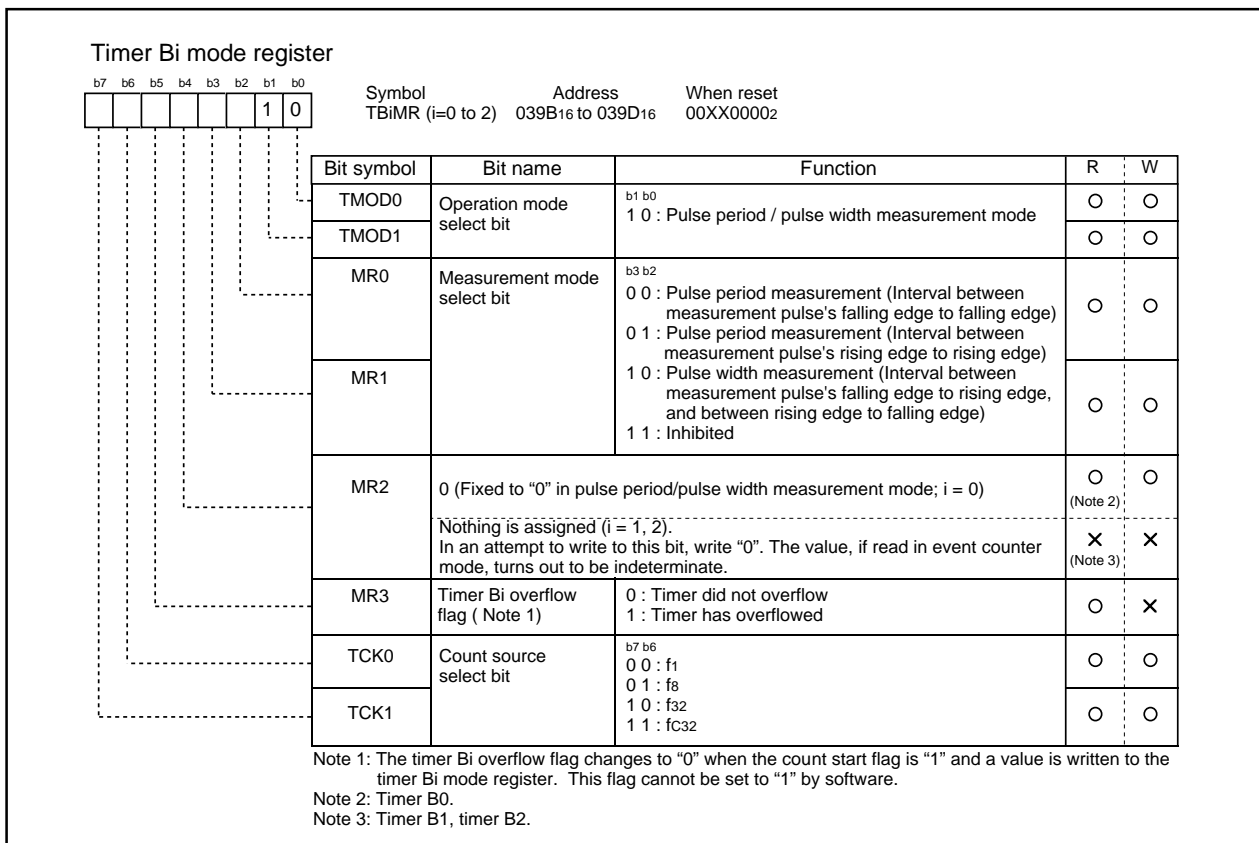
In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table TA-3.) Figure TA-6 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure TA-7 shows the operation timing when measuring a pulse period. Figure TA-8 shows the operation timing when measuring a pulse width.

**Table TA-3. Timer specifications in pulse period/pulse width measurement mode**

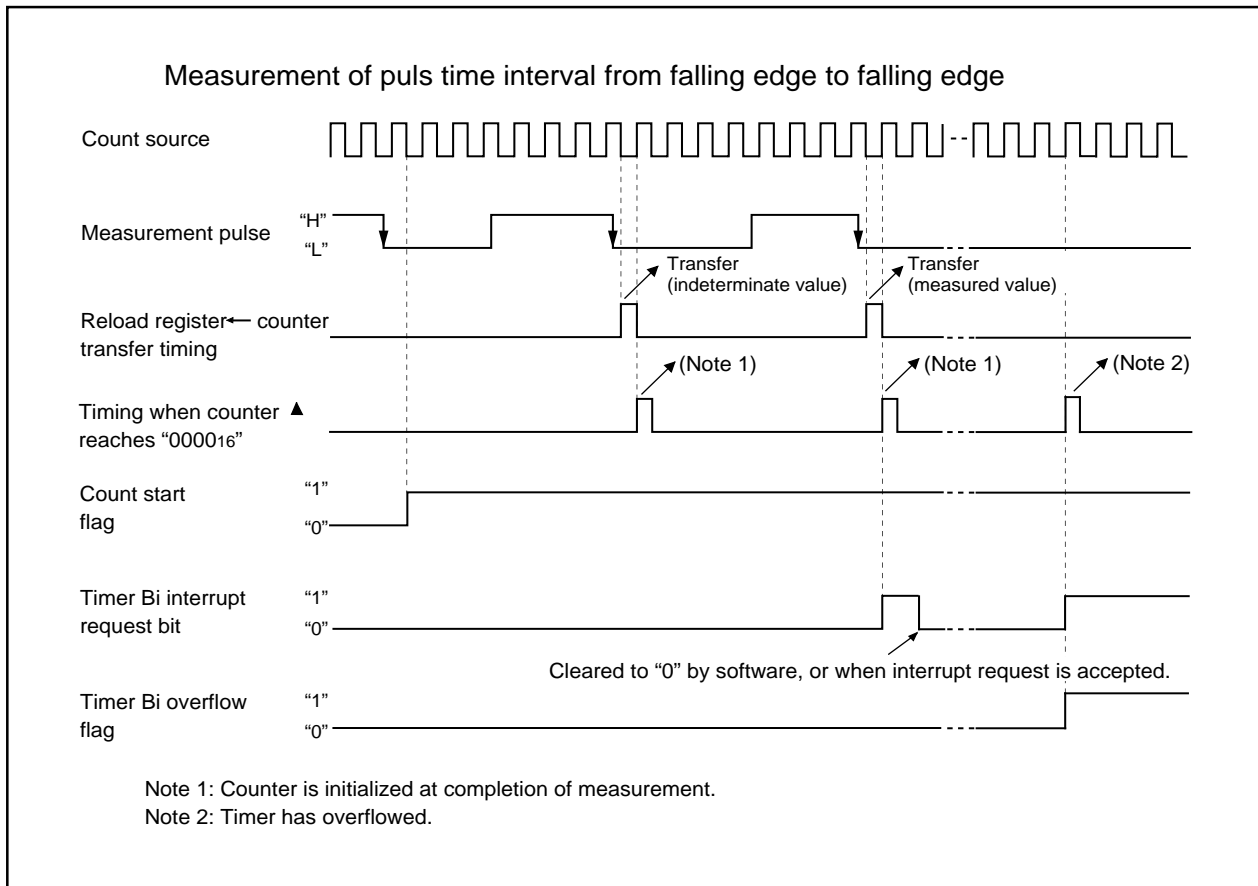
Item	Specification
Count source	f1, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>•Up count</li> <li>•Counter value "0000<sub>16</sub>" is transferred to reload register at measurement pulse's effective edge and the timer continues counting</li> </ul>
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	<ul style="list-style-type: none"> <li>•When measurement pulse's effective edge is input (Note 1)</li> <li>•When an overflow occurs. (Simultaneously, the timer Bi overflow flag changes to "1". The timer Bi overflow flag changes to "0" when the count start flag is "1" and a value is written to the timer Bi mode register.)</li> </ul>
TBiIN pin function	Measurement pulse input
Read from timer	When timer Bi register is read, it indicates the reload register's content (measurement result) (Note 2)
Write to timer	Cannot be written to

Note 1: An interrupt request is not generated when the first effective edge is input after the timer has started counting.

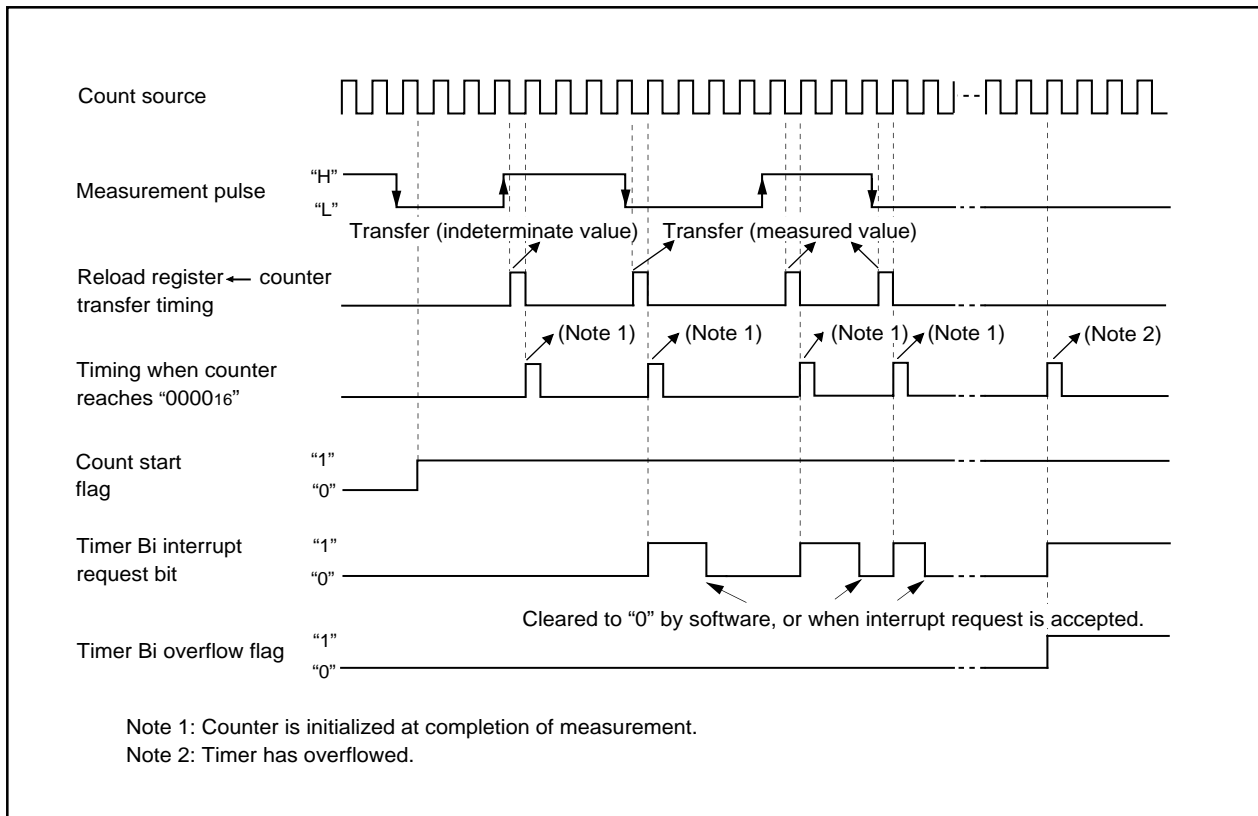
Note 2: The value read out from the timer Bi register is indeterminate until the second effective edge is input after the timer.



**Figure TA-6. Timer Bi mode register in pulse period/pulse width measurement mode**



**Figure TA-7. Operation timing when measuring a pulse period**



**Figure TA-8. Operation timing when measuring a pulse width**

## Serial I/O

Serial I/O is configured as two channels: UART0 and UART1.

UART0 and UART1 each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure GA-1 shows the block diagram of UART0 and UART1. Figures GA-2 shows the block diagram of the transmit/receive unit.

UART<sub>i</sub> (i=0, 1) has two operation modes: a clock synchronous serial I/O mode and a clock asynchronous serial I/O mode (UART mode). The contents of the serial I/O mode select bits (bits 0 to 2 at addresses 03A0<sub>16</sub> and 03A8<sub>16</sub>) determine whether UART<sub>i</sub> is used as a clock synchronous serial I/O or as a UART.

Although a few function are different, UART0 and UART1 have almost same functions.

Figures GA-3 through GA-5 show the registers related to UART<sub>i</sub>.

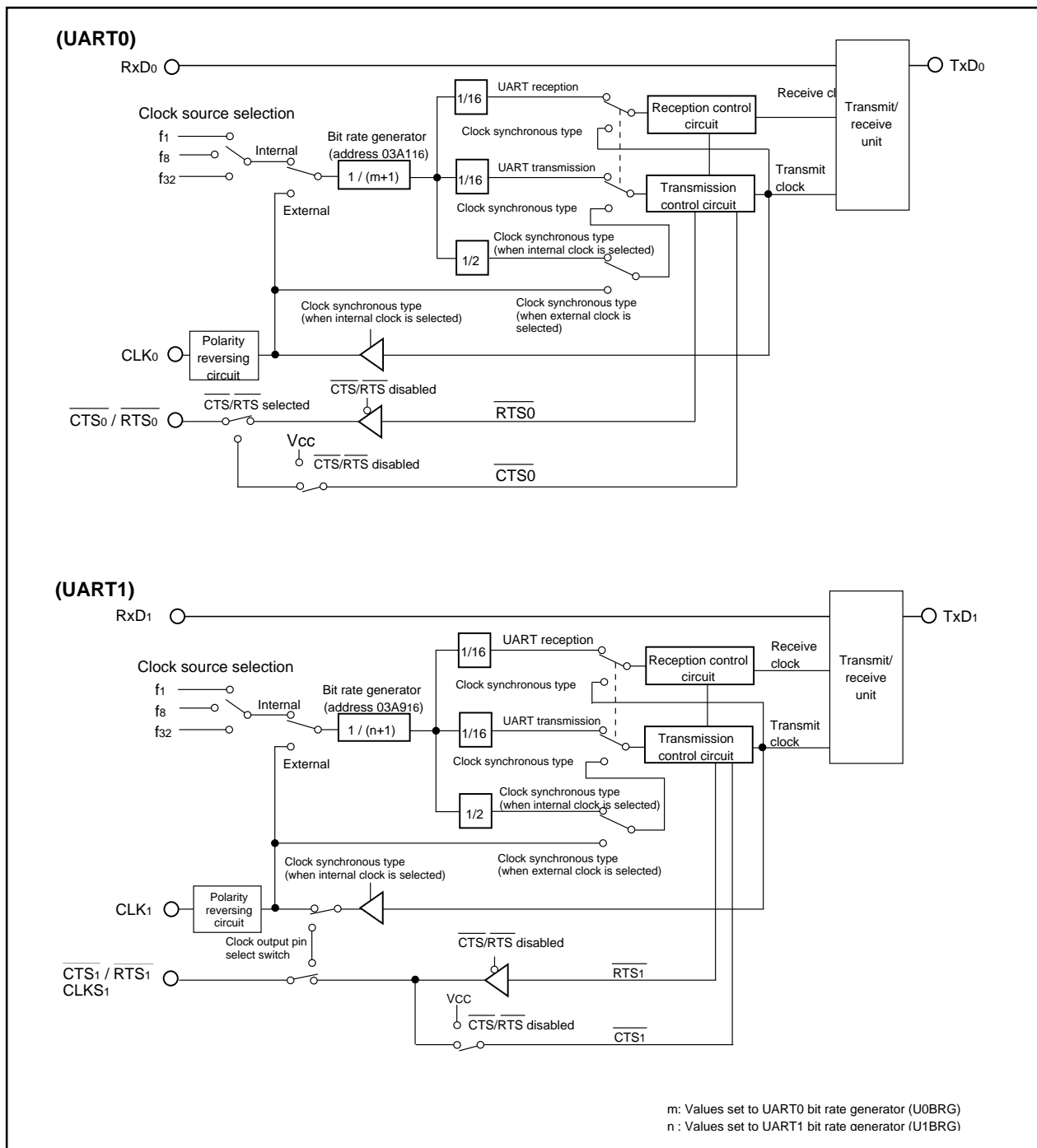


Figure GA-1. Block diagram of UART<sub>i</sub> (i = 0, 1)



Serial I/O

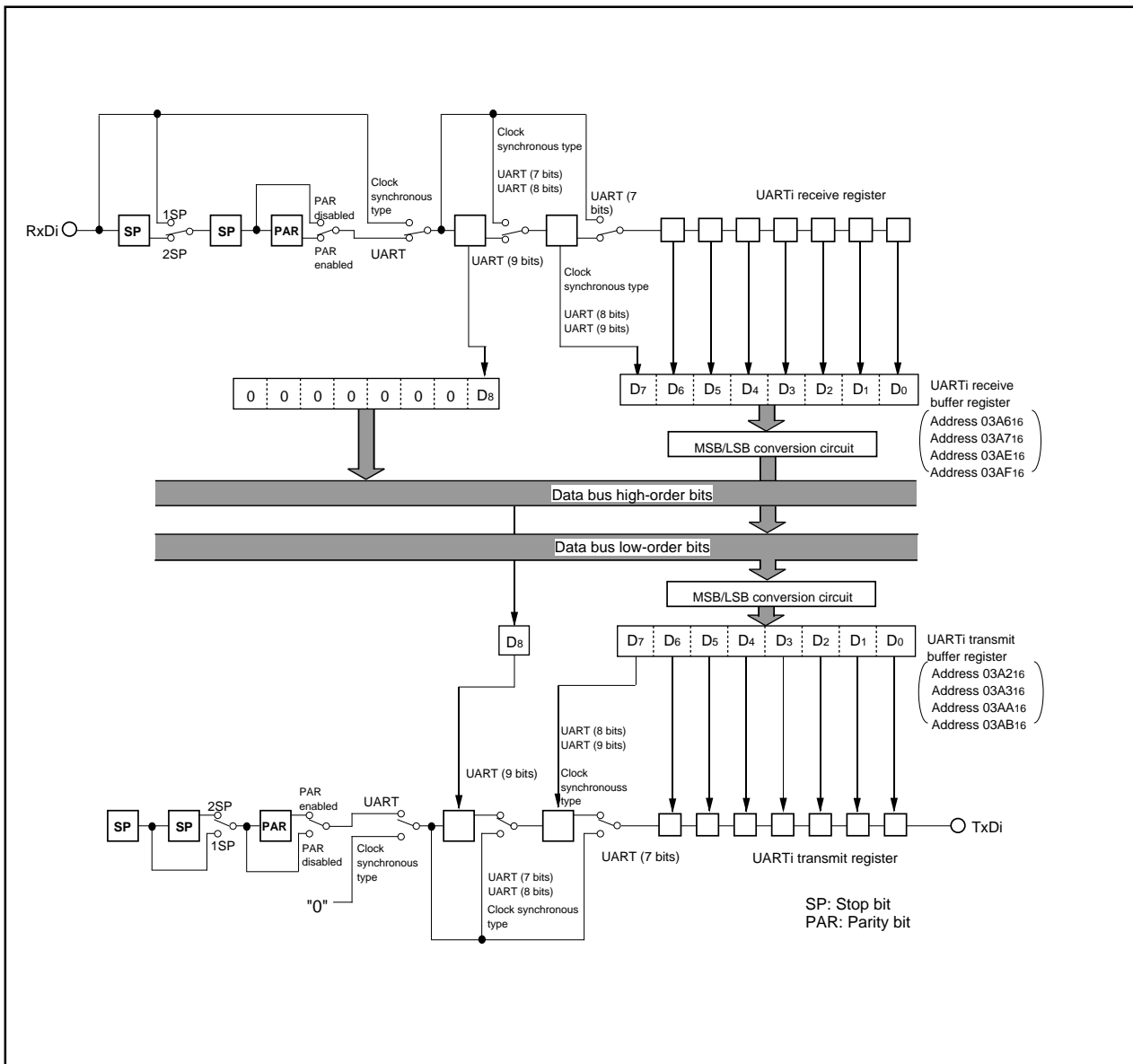


Figure GA-2. Block diagram of transmit/receive unit

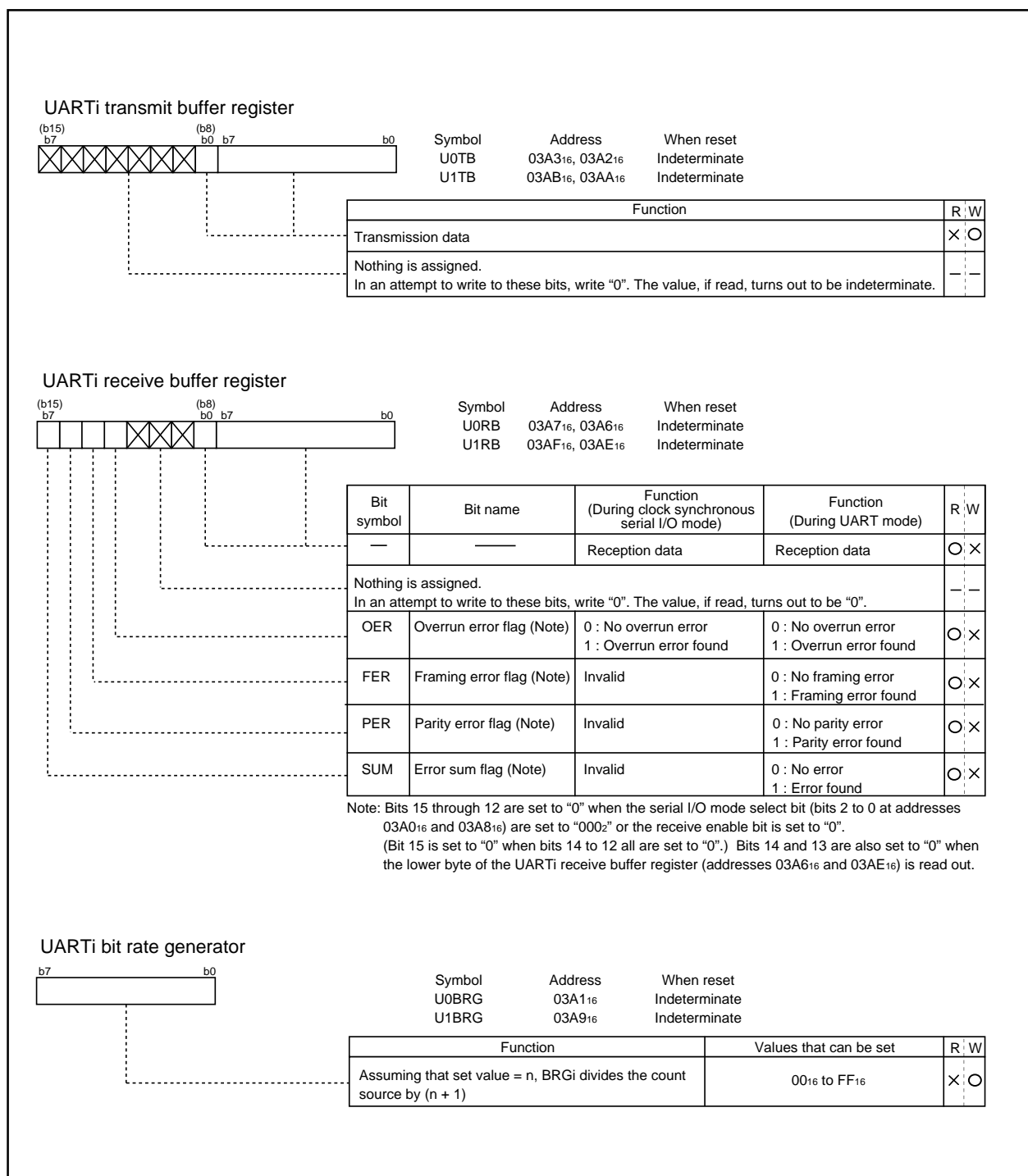


Figure GA-3. Serial I/O-related registers (1)

## Serial I/O

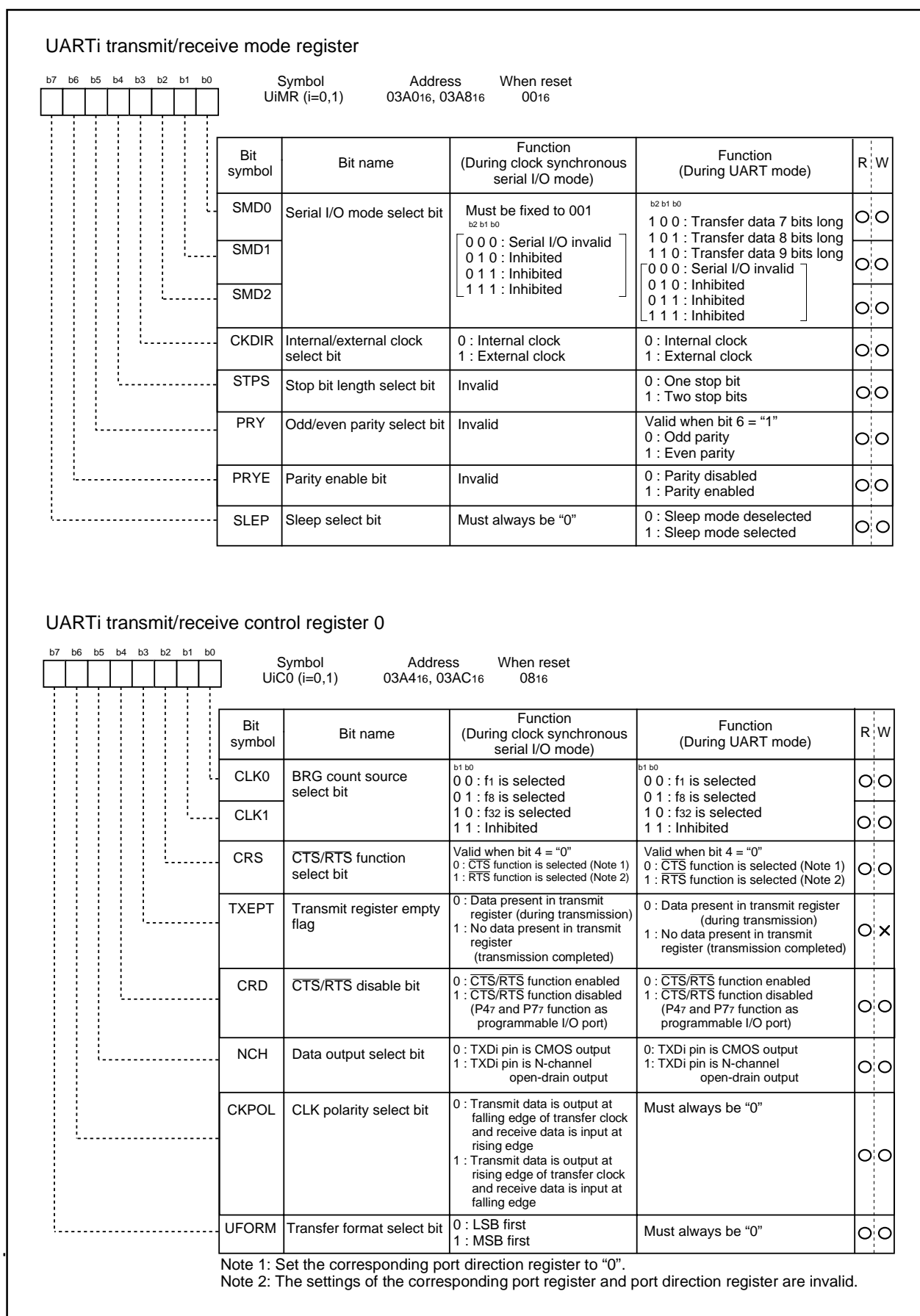


Figure GA-4. Serial I/O-related registers (2)

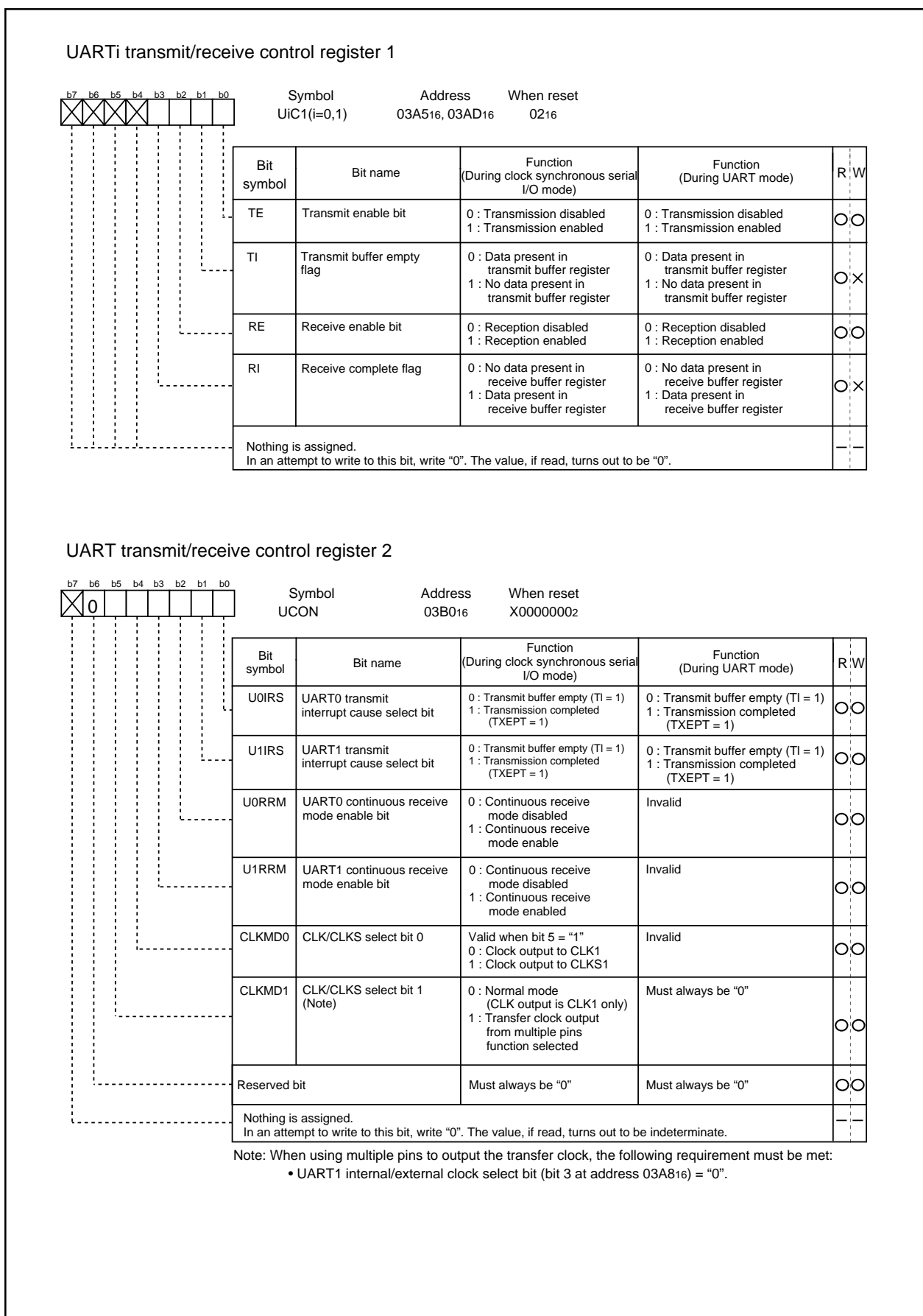


Figure GA-5. Serial I/O-related registers (3)

## Clock synchronous serial I/O mode

**(1) Clock synchronous serial I/O mode**

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Table GA-1 lists the specifications of the clock synchronous serial I/O mode. Figure GA-6 shows the UARTi transmit/receive mode register.

**Table GA-1. Specifications of clock synchronous serial I/O mode**

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• When internal clock is selected (bit 3 at address 03A0 <sub>16</sub> , 03A8 <sub>16</sub> = "0") : $f_i / 2(n+1)$ (Note 1) $f_i = f_1, f_8, f_{32}$ • When external clock is selected (bit 3 at address 03A0 <sub>16</sub> , 03A8 <sub>16</sub> = "1") : Input from CLKi pin (Note 2)
Transmission/reception control	• CTS function/ RTS function/ CTS,RTS function chosen to be invalid
Transmission start condition	• To start transmission, the following requirements must be met: – Transmit enable bit (bit 0 at address 03A5 <sub>16</sub> , 03AD <sub>16</sub> ) = "1" – Transmit buffer empty flag (bit 1 at addresses 03A5 <sub>16</sub> , 03AD <sub>16</sub> ) = "0" – When CTS function is selected, CTS input level = "L" • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "1": CLKi input level = "L"
Reception start condition	• To start reception, the following requirements must be met: – Receive enable bit (bit 2 at address 03A5 <sub>16</sub> , 03AD <sub>16</sub> ) = "1" – Transmit enable bit (bit 0 at address 03A5 <sub>16</sub> , 03AD <sub>16</sub> ) = "1" – Transmit buffer empty flag (bit 1 at address 03A5 <sub>16</sub> , 03AD <sub>16</sub> ) = "0" • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "1": CLKi input level = "L"
Interrupt request generation timing	• When transmitting – Transmit interrupt cause select bit (bits 0,1 at address 03B0 <sub>16</sub> ) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed – Transmit interrupt cause select bit (bits 0,1 at address 03B0 <sub>16</sub> ) = "1": Interrupts requested when data transmission from UARTi transfer register is completed • When receiving – Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed
Error detection	• Overrun error (Note 3) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out
Select function	• CLK polarity selection Whether transmit data is output/input at the rising edge or falling edge of the transfer clock can be selected • LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected • Continuous receive mode selection Reception is enabled simultaneously by a read from the receive buffer register • Transfer clock output from multiple pins selection UART1 transfer clock can be set 2 pins, and can be selected to output from which pin.

Note 1: "n" denotes the value 00<sub>16</sub> to FF<sub>16</sub> that is set to the UART bit rate generator.

Note 2: Maximum 5 Mbps.

Note 3: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit is not set to "1".

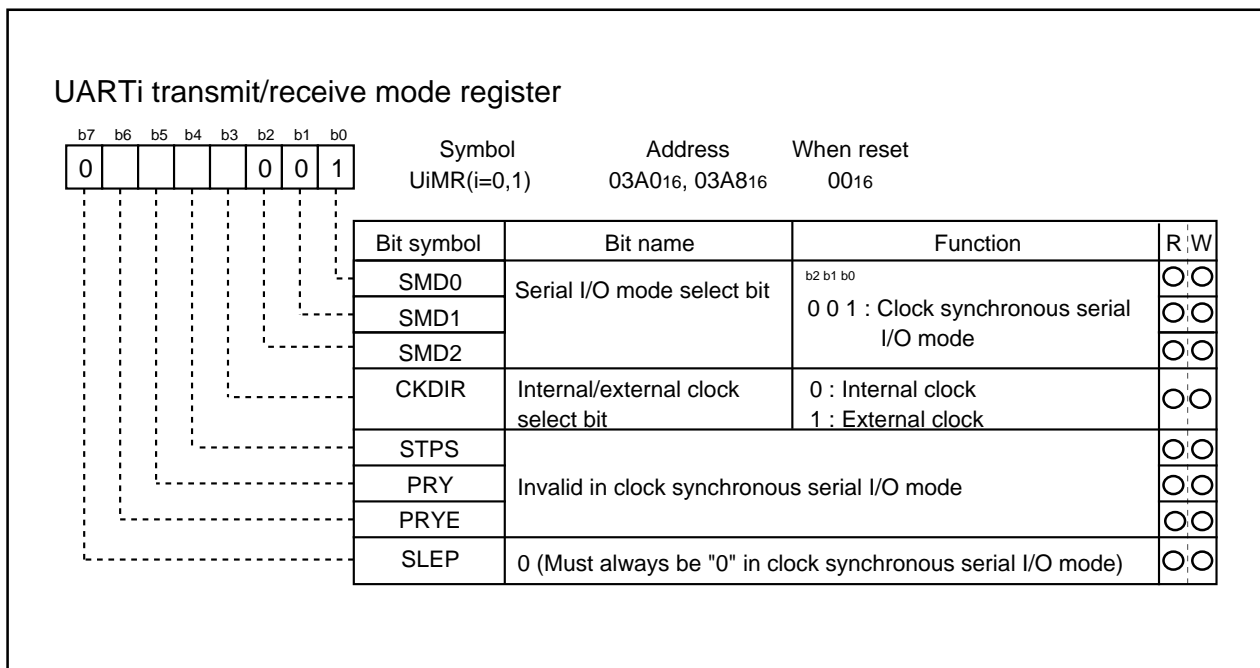


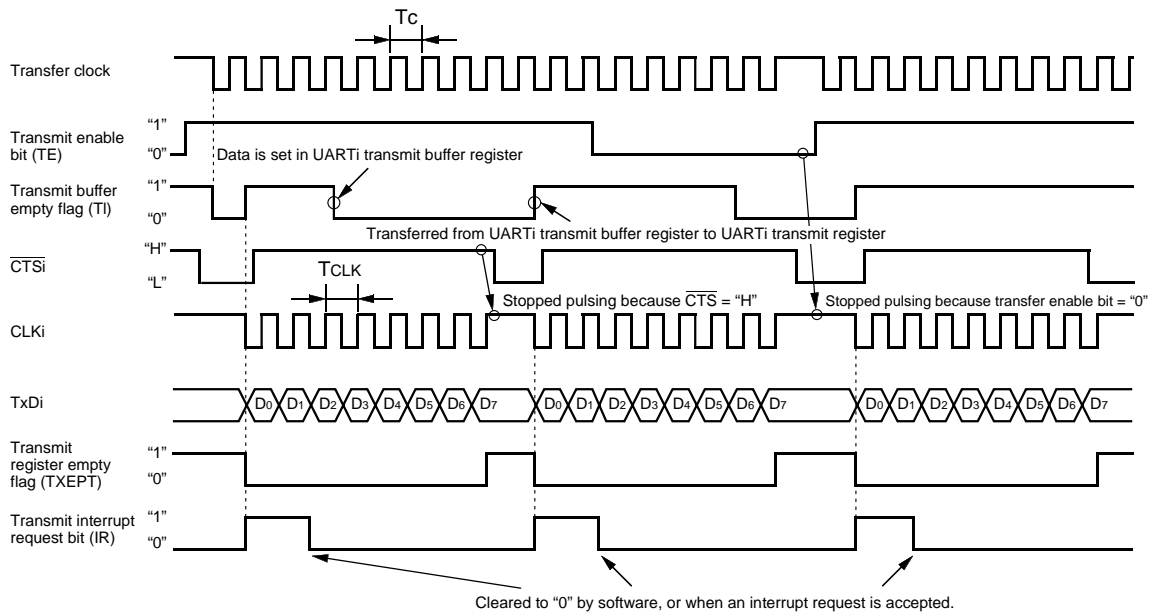
Figure GA-6. UARTi transmit/receive mode register in clock synchronous serial I/O mode (i=0,1)

Table GA-2 lists the functions of the input/output pins during clock synchronous serial I/O mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

Table GA-2. Input/output pin functions in clock synchronous serial I/O mode (i=0,1)

Pin name	Function	Method of selection
TxDi (P44, P74)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P45, P75)	Serial data input	Port P45, P75 direction register (bits 5 at address 03EA <sub>16</sub> and 03EF <sub>16</sub> ) = "0" (Can be used as an input port when performing transmission only)
CLKi (P46, P76)	Transfer clock output	Internal/external clock select bit (bit 3 at address 03A0 <sub>16</sub> , 03A8 <sub>16</sub> ) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0 <sub>16</sub> , 03A8 <sub>16</sub> ) = "1" Port P46, P76 direction register (bits 6 at address 03EA <sub>16</sub> and 03EF <sub>16</sub> ) = "0"
CTS <sub>i</sub> /RTS <sub>i</sub> (P47, P77)	CTS input	CTS/RTS disable bit (bit 4 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "0" CTS/RTS function select bit (bit 2 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "0" Port P47, P77 direction register (bits 7 address 03EA <sub>16</sub> and 03EF <sub>16</sub> ) = "0"
	RTS output	CTS/RTS disable bit (bit 4 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "0" CTS/RTS function select bit (bit 2 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "1"
	Programmable I/O port	CTS/RTS disable bit (bit 4 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "1"

• Example of transmit timing (when internal clock is selected)



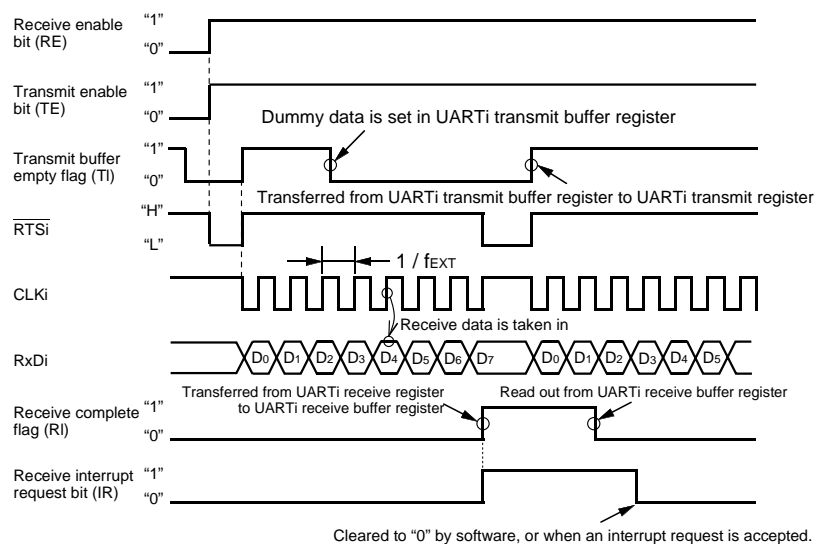
Shown in ( ) are bit symbols.  
The above timing applies to the following settings:

- Internal clock is selected.
- CTS function is selected.
- CLK polarity select bit = "0".
- Transmit interrupt cause select bit = "0".

$$T_c = T_{CLK} = 2(n + 1) / f_i$$

$f_i$ : frequency of BRGi's count source ( $f_1, f_8, f_{32}$ )  
 $n$ : value set to BRGi

• Example of receive timing (when external clock is selected)



$f_{EXT}$ : frequency of external clock

Shown in ( ) are bit symbols.  
The above timing applies to the following settings:

- External clock is selected.
- RTS function is selected.
- CLK polarity select bit = "0".

Meet the following conditions when the CLK input before data reception = "H"

- Transmit enable bit → "1"
- Receive enable bit → "1"
- Dummy data write to UARTi transmit buffer register

Figure GA-7. Typical transmit/receive timings in clock synchronous serial I/O mode

**(a) Polarity select function**

As shown in Figure GA-8, the CLK polarity select bit (bit 6 at addresses 03A4<sub>16</sub>, 03AC<sub>16</sub>) allows selection of the polarity of the transfer clock.

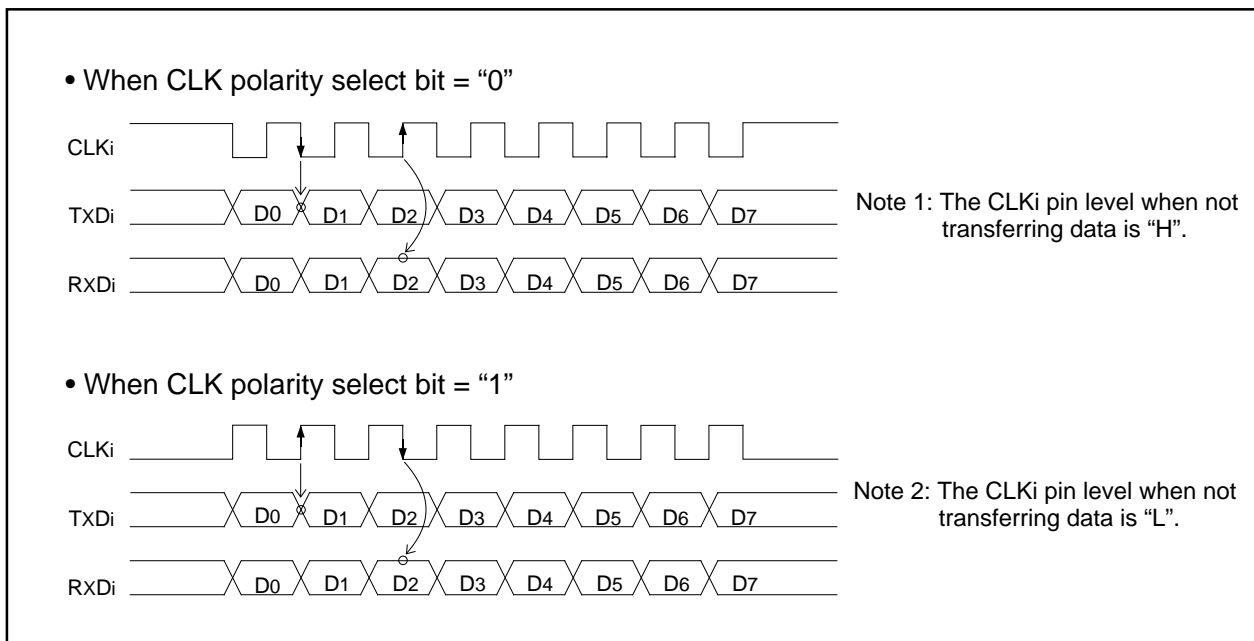


Figure GA-8. Polarity of transfer clock

**(b) LSB first/MSB first select function**

As shown in Figure GA-9, when the transfer format select bit (bit 7 at addresses 03A4<sub>16</sub>, 03AC<sub>16</sub>) = "0", the transfer format is "LSB first"; when the bit = "1", the transfer format is "MSB first".

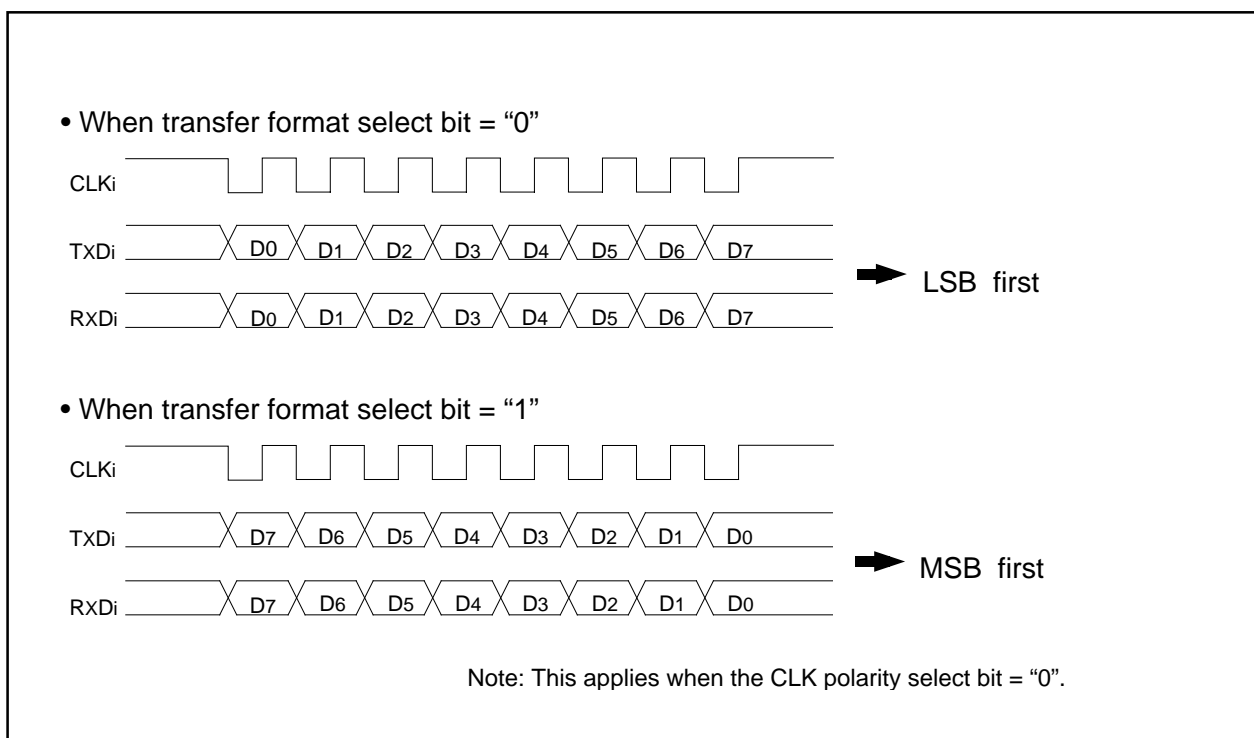


Figure GA-9. Transfer format



**(c) Transfer clock output from multiple pins function**

This function allows the setting two transfer clock output pins and choosing one of the two to output a clock by using the CLK and CLKS select bit (bits 4 and 5 at address 03B016). (See Figure GA-10.)

The multiple pins function is valid only when the internal clock is selected for UART1. Note that when this function is selected,  $\overline{\text{CTS}}/\overline{\text{RTS}}$  function of UART1 cannot be used.

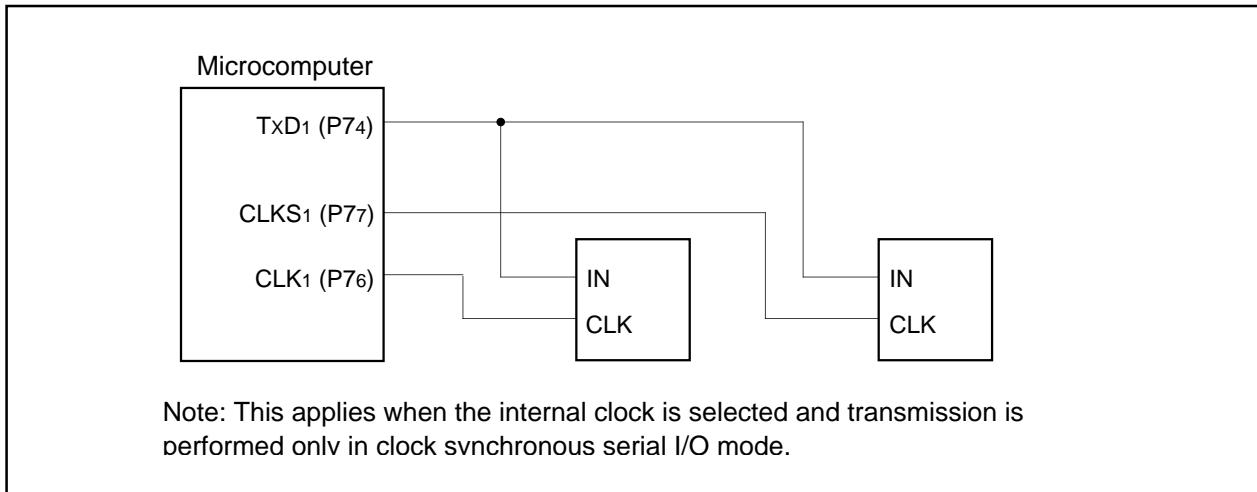


Figure GA-10. The transfer clock output from the multiple pins function usage

**(d) Continuous receive mode**

If the continuous receive mode enable bit (bits 2 and 3 at address 03B016) is set to "1", the unit is placed in continuous receive mode. In this mode, when the receive buffer register is read out, the unit simultaneously goes to a receive enable state without having to set dummy data to the transmit buffer register back again.

## Clock asynchronous serial I/O (UART) mode

**(2) Clock asynchronous serial I/O (UART) mode**

The UART allows transmitting and receiving data after setting the desired transfer rate and transfer data format. Tables GA-3 lists the specifications of the UART mode. Figure GA-11 shows the UARTi transmit/receive mode register.

**Table GA-3. Specifications of clock synchronous serial I/O mode**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>•Character bit (transfer data): 7 bits, 8 bits or 9 bits as selected</li> <li>•Start bit: 1 bit</li> <li>•Parity bit: Odd, even or nothing as selected</li> <li>•Stop bit: 1 bit or 2 bits as selected</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>•When internal clock is selected (bit 3 at addresses 03A0<sub>16</sub>, 03A8<sub>16</sub> = "0") : f<sub>i</sub>/16(n+1) (Note 1) f<sub>i</sub> = f<sub>1</sub>, f<sub>8</sub>, f<sub>32</sub></li> <li>•When external clock is selected (bit 3 at addresses 03A0<sub>16</sub>, 03A8<sub>16</sub> = "1") : f<sub>EXT</sub>/16(n+1) (Note 1) (Note 2)</li> </ul>
Transmission/reception control	<ul style="list-style-type: none"> <li>•CTS function/RTS function/CTS, RTS function chosen to be invalid</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>•To start transmission, the following requirements must be met: <ul style="list-style-type: none"> <li>- Transmit enable bit (bit 0 at addresses 03A5<sub>16</sub>, 03AD<sub>16</sub>) = "1"</li> <li>- Transmit buffer empty flag (bit 1 at addresses 03A5<sub>16</sub>, 03AD<sub>16</sub>) = "0"</li> <li>- When CTS function is selected, CTS input level = "L"</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>•To start reception, the following requirements must be met: <ul style="list-style-type: none"> <li>- Receive enable bit (bit 2 at addresses 03A5<sub>16</sub>, 03AD<sub>16</sub>) = "1"</li> <li>- Start bit detection</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>•When transmitting <ul style="list-style-type: none"> <li>- Transmit interrupt cause select bits (bits 0,1 at address 03B0<sub>16</sub>) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed</li> <li>- Transmit interrupt cause select bits (bits 0, 1 at address 03B0<sub>16</sub>) = "1": Interrupts requested when data transmission from UARTi transfer register is completed</li> </ul> </li> <li>•When receiving <ul style="list-style-type: none"> <li>- Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>•Overrun error (Note 3) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out</li> <li>•Framing error This error occurs when the number of stop bits set is not detected</li> <li>•Parity error This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set</li> <li>•Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered</li> </ul>
select function	<ul style="list-style-type: none"> <li>•Sleep mode selection This mode is used to transfer data to and from one of multiple slave microcomputers</li> </ul>

Note 1: 'n' denotes the value 00<sub>16</sub> to FF<sub>16</sub> that is set to the UARTi bit rate generator.

Note 2: f<sub>EXT</sub> is input from the CLKi pin.

Note 3: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit is not set to "1".

## Clock asynchronous serial I/O (UART) mode

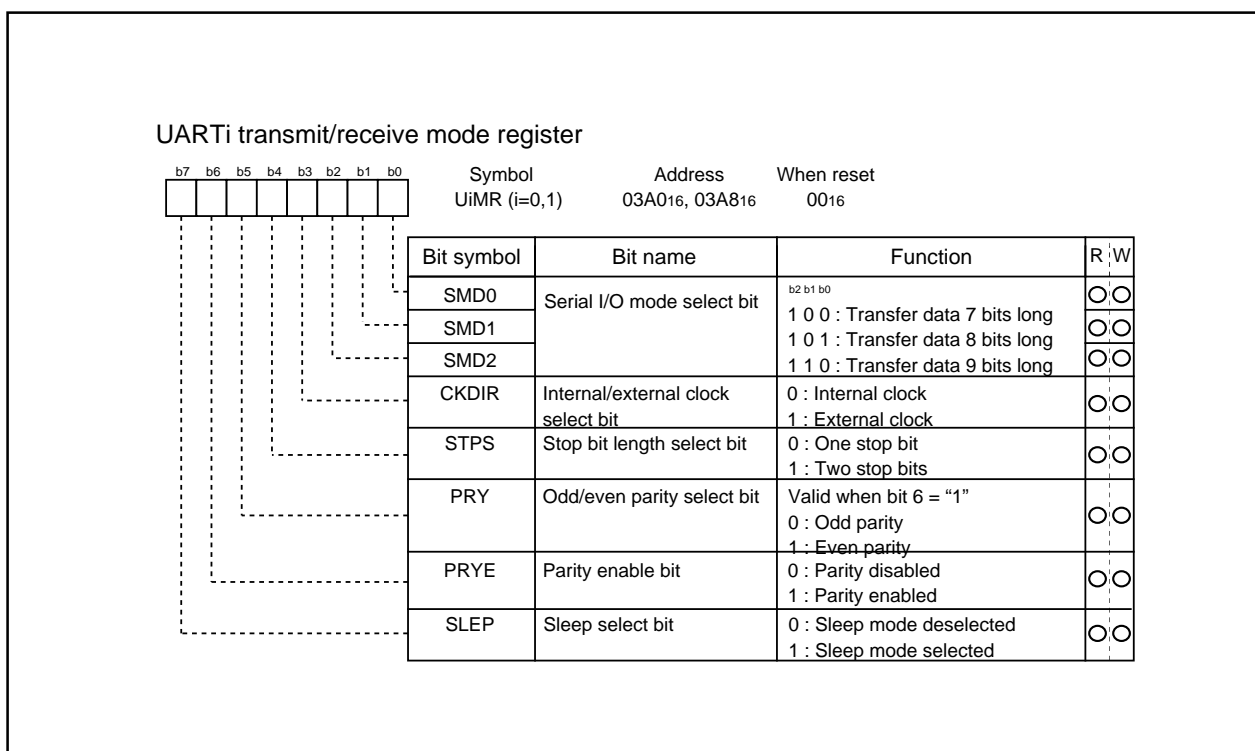
Figure GA-11. UART<sub>i</sub> transmit/receive mode register in UART mode

Table GA-4 lists the functions of the input/output pins during UART mode. Note that for a period from when the UART<sub>i</sub> operation mode is selected to when transfer starts, the TxDi pin outputs a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

Table GA-4. Input/output pin functions in UART mode (i=0,1)

Pin name	Function	Method of selection
TxD <sub>i</sub> (P44, P74)	Serial data output	(Outputs dummy data when performing reception only)
RxD <sub>i</sub> (P45, P75)	Serial data input	Port P45, P75 direction register (bits 5 at address 03EA <sub>16</sub> and 03EF <sub>16</sub> ) = "0" (Can be used as an input port when performing transmission only)
CLK <sub>i</sub> (P46, P76)	Programmable I/O port	Internal/external clock select bit (bit 3 at address 03A0 <sub>16</sub> , 03A8 <sub>16</sub> ) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0 <sub>16</sub> , 03A8 <sub>16</sub> ) = "1"
CTS <sub>i</sub> /RTS <sub>i</sub> (P47, P77)	CTS input	CTS/RTS disable bit (bit 4 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "0" CTS/RTS function select bit (bit 2 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "0" Port P47, P77 direction register (bits 7 at address 03EA <sub>16</sub> and 03EF <sub>16</sub> ) = "0"
	RTS output	CTS/RTS disable bit (bit 4 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "0" CTS/RTS function select bit (bit 2 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "1"
	Programmable I/O port	CTS/RTS disable bit (bit 4 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> ) = "1"

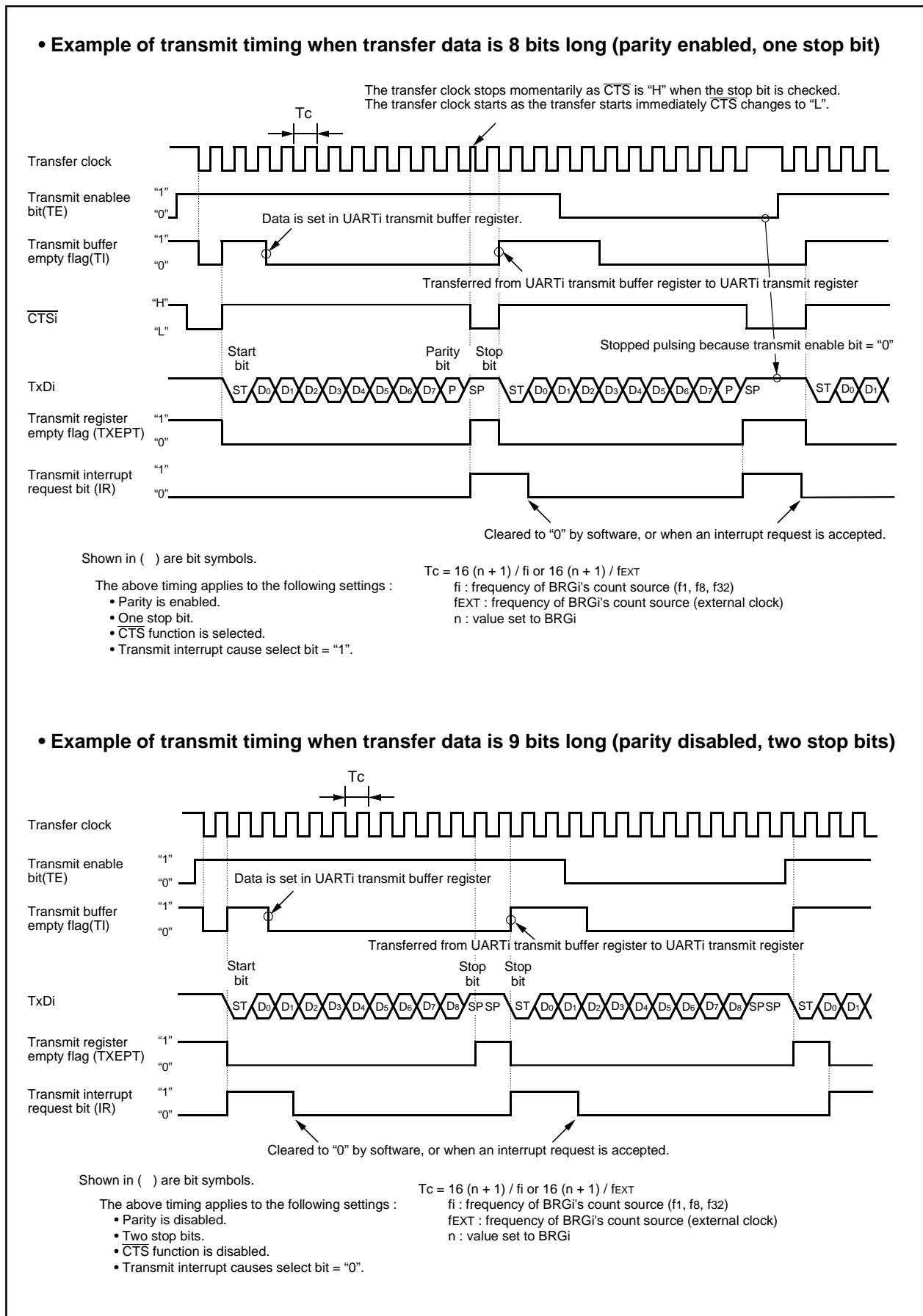


Figure GA-12. Typical transmit timings in UART mode

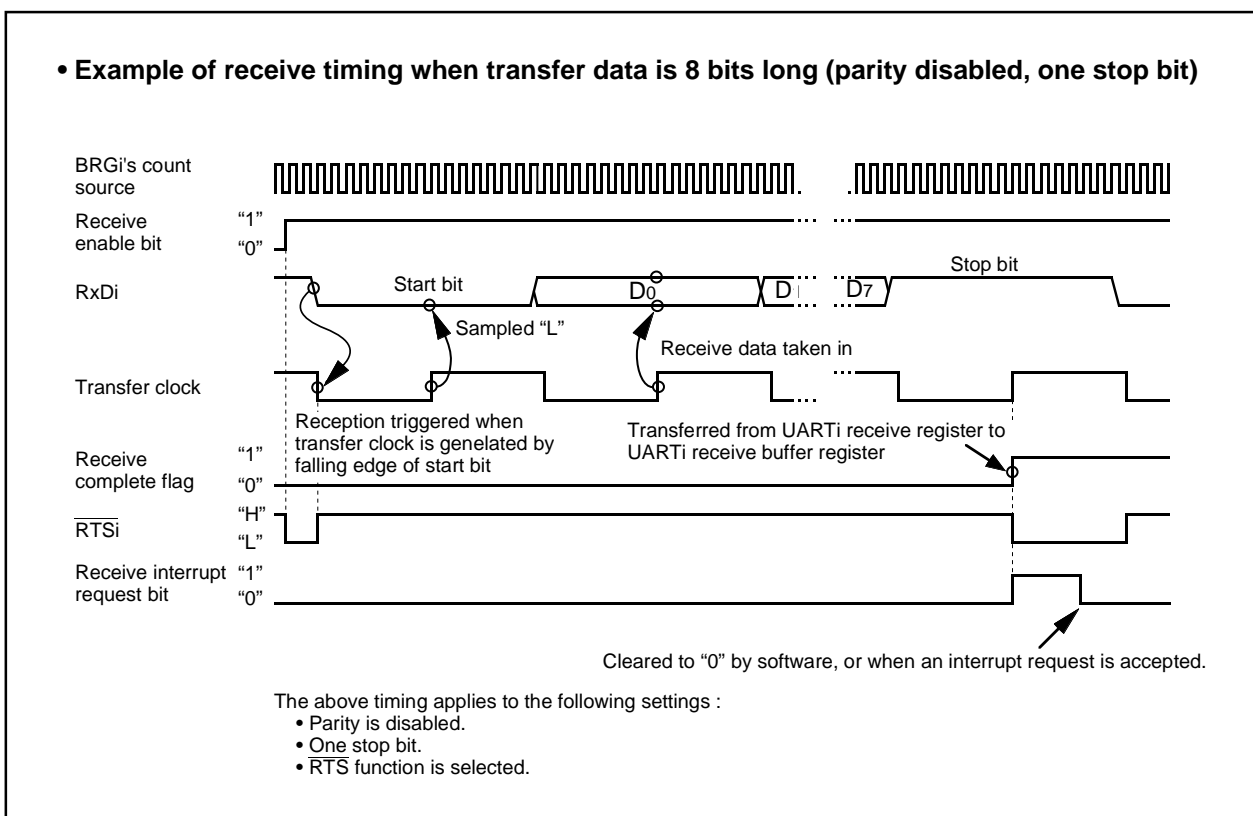


Figure GA-13. Typical receive timing in UART mode

(a) Sleep mode (UART0, UART1)

This mode is used to transfer data between specific microcomputers among multiple microcomputers connected using UARTi. The sleep mode is selected when the sleep select bit (bit 7 at addresses 03A016, 03A816) is set to "1" during reception. In this mode, the unit performs receive operation when the MSB of the received data = "1" and does not perform receive operation when the MSB = "0".

## Serial I/O2

Serial I/O2 is used as the clock synchronous serial I/O and has an ordinary mode and an automatic transfer mode. In the automatic transfer mode, serial transfer is performed through the serial I/O automatic transfer RAM which has up to 256 bytes (addresses 0040016 to 004FF16).

The SRDY2, SBUSY2 and SSTB2 pins each have a handshake I/O signal function and can select either “H” active or “L” active for active logic.

**Table GA-1. Specifications of clock synchronous serial I/O2**

Item	Specification
Serial mode	<ul style="list-style-type: none"> <li>• 8-bit serial I/O mode (non-automatic transfer)</li> <li>• Automatic transfer serial I/O mode</li> </ul>
Transfer data format	<ul style="list-style-type: none"> <li>• Transfer data length: 8 bits</li> <li>• Full duplex mode / transmit-only mode selected by bit 5 at address 034216</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>• When internal clock is selected (bit 2 at address 034216 = “0”) : selected by bits 5 to 7 at address 034816</li> <li>• When external clock is selected (bit 2 at address 034216 = “1”) : Input from SCLK21 pin, SCLK22 pin(Note 2)</li> </ul>
Transfer rate	<ul style="list-style-type: none"> <li>• When internal clock is selected : <math>f(XIN)/4</math>, <math>f(XIN)/8</math>, <math>f(XIN)/16</math>, <math>f(XIN)/32</math>, <math>f(XIN)/64</math>, <math>f(XIN)/128</math>, <math>f(XIN)/256</math></li> <li>• When external clock is selected : input cycle 0.95 <math>\mu</math>s or less</li> </ul>
Transmission/reception control	<ul style="list-style-type: none"> <li>• SSTB2 output / SBUSY2 input or output / SRDY2 input or output chosen</li> </ul>
Transmission / reception start condition	<ul style="list-style-type: none"> <li>• To start transmission / reception, the following requirements must be met: <ul style="list-style-type: none"> <li>– Serial I/O initialization bit (bit 4 at address 034216) = “1”</li> <li>– When <math>\overline{SBUSY2}</math> input, or <math>\overline{SRDY2}</math> input is selected : selected input level = “H”</li> <li>– When <math>\overline{SBUSY2}</math> input, or <math>\overline{SRDY2}</math> input is selected : selected input level = “L”</li> </ul> </li> <li>• Furthermore, if external clock is selected, the following requirements must also be met: <ul style="list-style-type: none"> <li>– Input level of SCLK21 or SCLK22 = “H”</li> </ul> </li> </ul>
Transmission and reception stop condition	<ul style="list-style-type: none"> <li>• To stop transmission and reception, set serial I/O initialization bit (bit 4 at address 034216) to “0” regardless internal clock and external clock.</li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• 8-bit serial I/O mode : Interrupts requested when 8-bit data transfer is completed</li> <li>• Automatic transfer serial I/O mode : Interrupts requested when last receive data transfer to Automatic transfer RAM</li> </ul>
Select function	<ul style="list-style-type: none"> <li>• SOUT2 P-channel output disable function CMOS output or N-channel open-drain output can be selected</li> <li>• LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected</li> <li>• Serial I/O2 clock pin select bit Serial clock input/output can be selected; SCLK21 or SCLK22</li> <li>• SBUSY output, SSTB2 output select function (only automatic transfer serial mode) SBUSY output, SSTB2 output can be selected; 1-byte data transfer unit or all data transfer unit</li> <li>• SOUT2 pin control bit Either output active or high-impedance can be selected as a SOUT2 pin state at serial non-transfer .</li> </ul>

Note 1: It is necessary to set the serial I/O clock pin select bit ( bit 7 at address 034216)

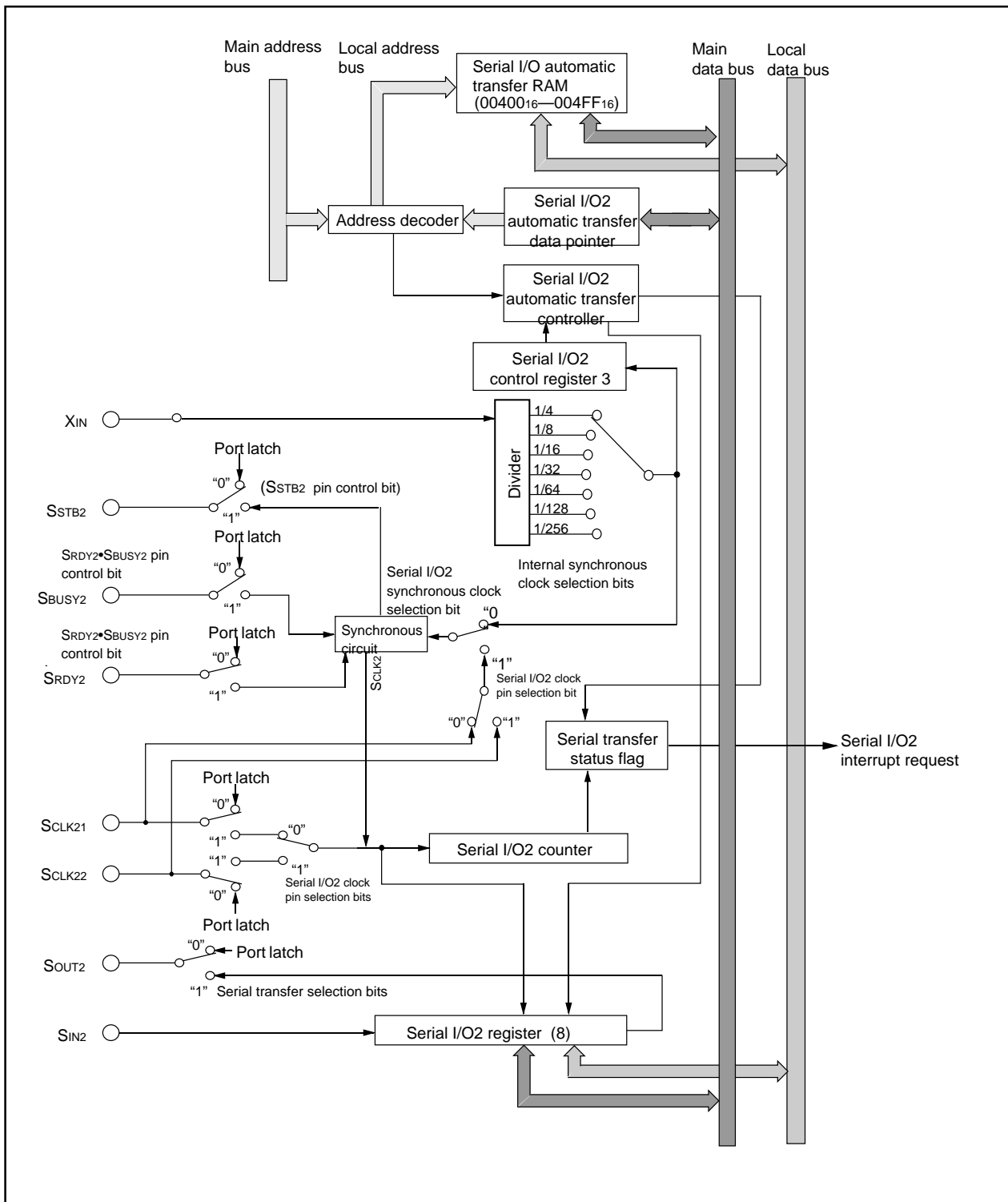


Figure GA-1. Block Diagram of Serial I/O2

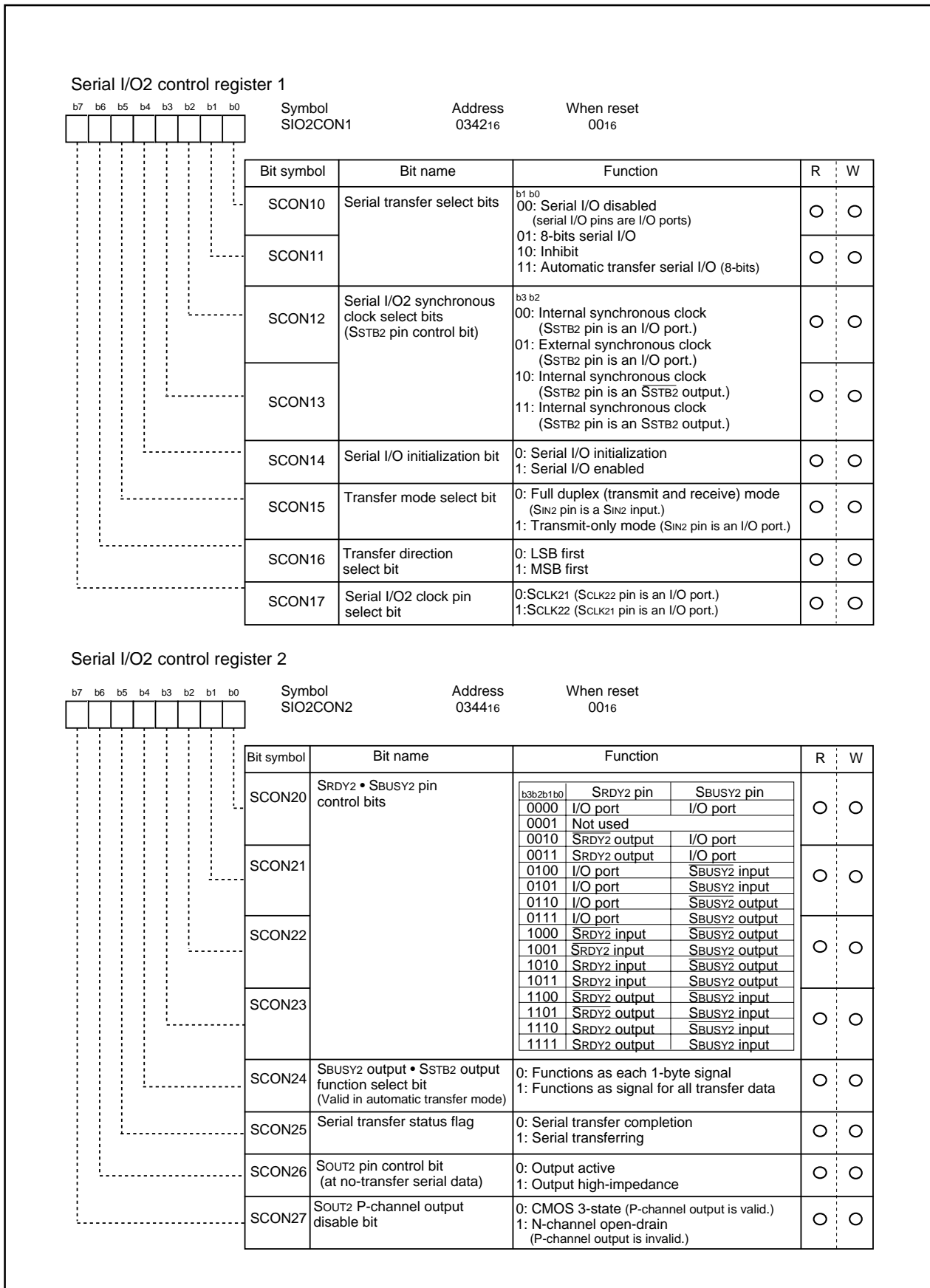


Figure GA-2. Serial I/O2 Control Registers 1, 2



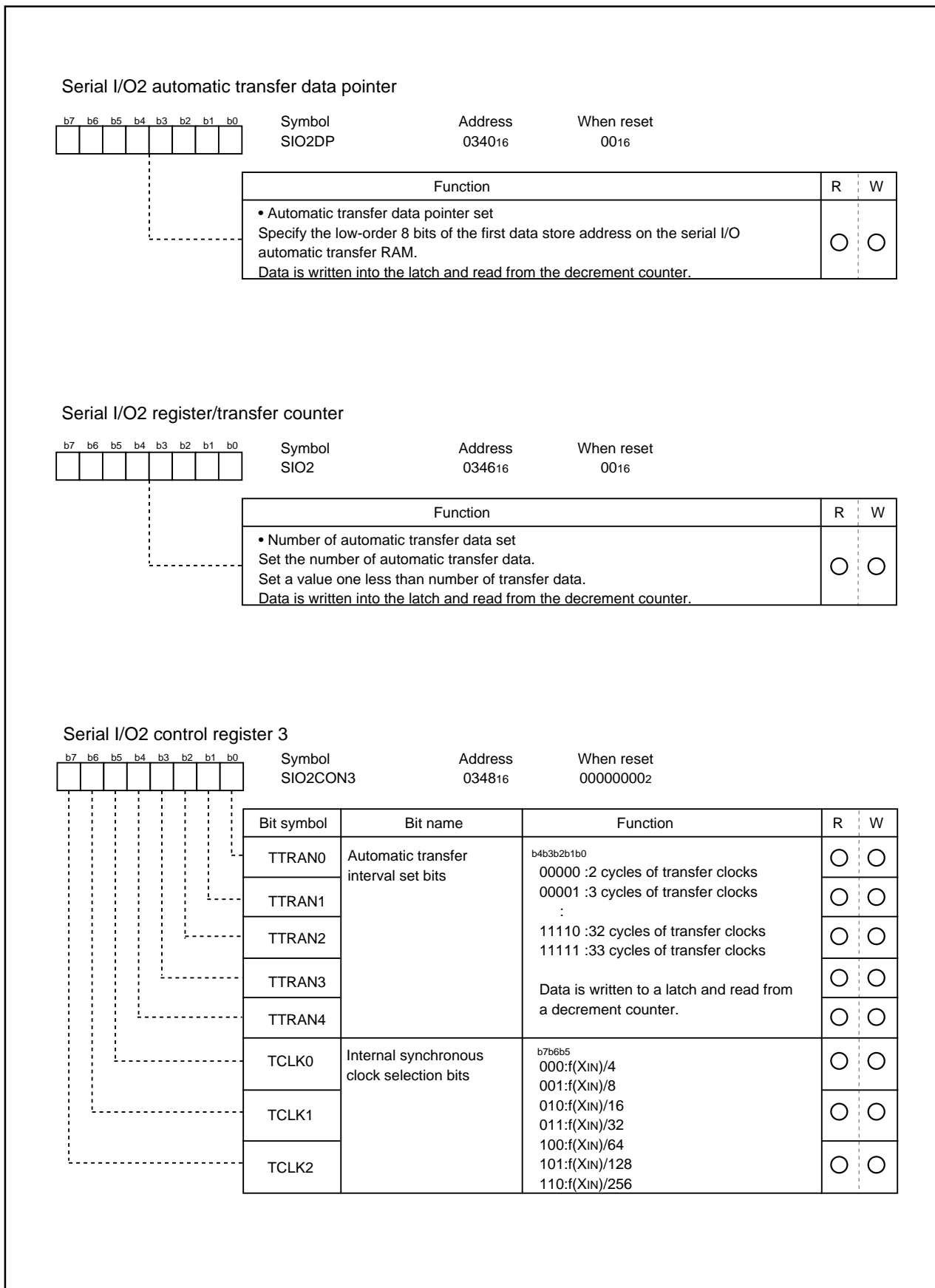


Figure GA-3. Serial I/O2 automatic transfer data pointer

Table GA-2 lists the functions of the serial I/O2 input/output pins

**Table GA-2. Functions of the serial I/O2 input/output pins**

Pin name	Function	Method of selection
SOUT2 (P94)	Serial data output	Port P94 direction register (bit 4 at address 03F316)= "1" SOUT2 P-channel output disable bit (bit 7 at address 034416)= "0", "1" SOUT2 pin control bit (bit 6 at address 034416)= "0", "1" (Outputs dummy data when performing reception only)
SIN2 (P93)	Serial data input	Port P93 direction register (bit 4 at address 03F316)= "0" Transfer mode select bit (bit 5 at address 034216)= "0" (Input/output port when transfer mode select bit (bit 5 at address 034216)= "1")
SCLK21 (P95)	Transfer clock output	Serial I/O2 synchronous clock select bits (bits 2, 3 at address 034216) = "00", "01" Serial I/O2 clock pin select bit (bit 7 at address 034216) = "0"
	Transfer clock input	Serial I/O2 synchronous clock select bits (bits 2, 3 at address 034216) = "01", "11" Serial I/O2 clock pin select bit (bit 7 at address 034216) = "0" Port P95 direction register (bit 5 at address 03F316)= "0"
SCLK22 (P96)	Transfer clock output	Serial I/O2 synchronous clock select bits (bits 2, 3 at address 034216) = "00", "01" Serial I/O2 clock pin select bit (bit 7 at address 034216) = "1"
	Transfer clock input	Serial I/O2 synchronous clock select bits (bits 2, 3 at address 034216) = "01", "11" Serial I/O2 clock pin select bit (bit 7 at address 034216) = "1" Port P96 direction register (bit 6 at address 03F316)= "0"
SRDY2 (P90)	SRDY input / output	Set by SRDY2 • SBUSY2 pin control bits (bits 0 to 3 at address 034416)
SBUSY2 (P91)	SBUSY input / output	Set by SRDY2 • SBUSY2 pin control bits (bits 0 to 3 at address 034416) SBUSY2 output • SSTB2 output function select bit (bit 4 at address 034416)= "0", "1"
SSTB2 (P92)	SSTB input / output	Serial I/O2 synchronous clock select bits (bits 2, 3 at address 034216) = "10", "11" SBUSY2 output • SSTB2 output function select bit (bit 4 at address 034416)= "0", "1"

## SOUT2 Output

Either output active or high-impedance can be selected as a SOUT2 pin state at serial non-transfer by the SOUT2 pin control bit (bit 6 of address 034416).

However, when the external synchronous clock is selected, perform the following setup to put the SOUT2 pin into a high-impedance state.

When the SCLK2i (i = 1, 2) input is "H" after completion of transfer, set the SOUT2 pin control bit to "1". When the SCLK2i (i = 1, 2) input goes to "L" after the start of the next serial transfer, the SOUT2 pin control bit is automatically reset to "0" and put into an output active state.

## Serial I/O2 Mode

There are two types of serial I/O2 modes: 8-bit serial I/O mode where automatic transfer RAM is not used, and an automatic transfer serial I/O mode.

### (1) 8-bit Serial I/O Mode

Address 0346<sub>16</sub> is assigned to the serial I/O2 register. When the internal synchronous clock is selected, a serial transfer of the 8-bit serial I/O is started by a write signal to the serial I/O2 register (address 0346<sub>16</sub>).

The serial transfer status flag (bit 5 of address 0344<sub>16</sub>) is set to "1" by writing into the serial I/O2 register and reset to "0" after completion of 8-bit transfer. At the same time, a serial I/O2 interrupt request occurs. If the transfer is completed, the receive data is read out from serial I/O2 register. When the external synchronous clock is selected, the contents of the serial I/O2 register are continuously shifted while transfer clocks are input to SCLK21 or SCLK22. Therefore, the clock needs to be controlled externally.

### (2) Automatic Transfer Serial I/O Mode

Address 0346<sub>16</sub> is assigned to the transfer counter (1-byte units). The serial I/O2 automatic transfer controller controls the write and read operations of the serial I/O2 register. The serial I/O automatic transfer RAM is mapped to addresses 00400<sub>16</sub> to 004FF<sub>16</sub>. Before starting transfer, make sure the 8 low-order bits of the address that contains the beginning data to be serially transferred is set to the automatic transfer data pointer (address 0340<sub>16</sub>).

When the internal synchronous clock is selected, the transfer interval is inserted between one data and another in the following cases:

1. When using no handshake signal
2. When using the SRDY2 output, SBUSY2 output, and SSTB2 output of the handshake signal independently
3. When using a combination of SRDY2 output and SSTB2 output or a combination of SBUSY2 output and SSTB2 output of the handshake signal

The transfer interval can be set in the range of 2 to 23 cycles using the automatic transfer interval set bit (bits 0–4 of address 0348<sub>16</sub>).

Also, when using SBUSY2 output as a signal for each occurrence of the all transfer data, a transfer interval is inserted before the system starts sending or receiving the first data and after the system finished sending or receiving the last data, not just between one data and another.

Furthermore, when using SSTB2 output, the transfer interval between each 1-byte data is extended by 2 cycles from the set value no matter how the SBUSY2 output. SSTB2 output function select bit (bit 4 of address 0344<sub>16</sub>) is set.

When using SBUSY2 output and SSTB2 output in combination as a signal for each occurrence of the all transfer data, the transfer interval after the system finished sending or receiving the last data is extended by 2 cycles from the set value.

When an external synchronous clock is selected, the automatic transfer interval is disabled.

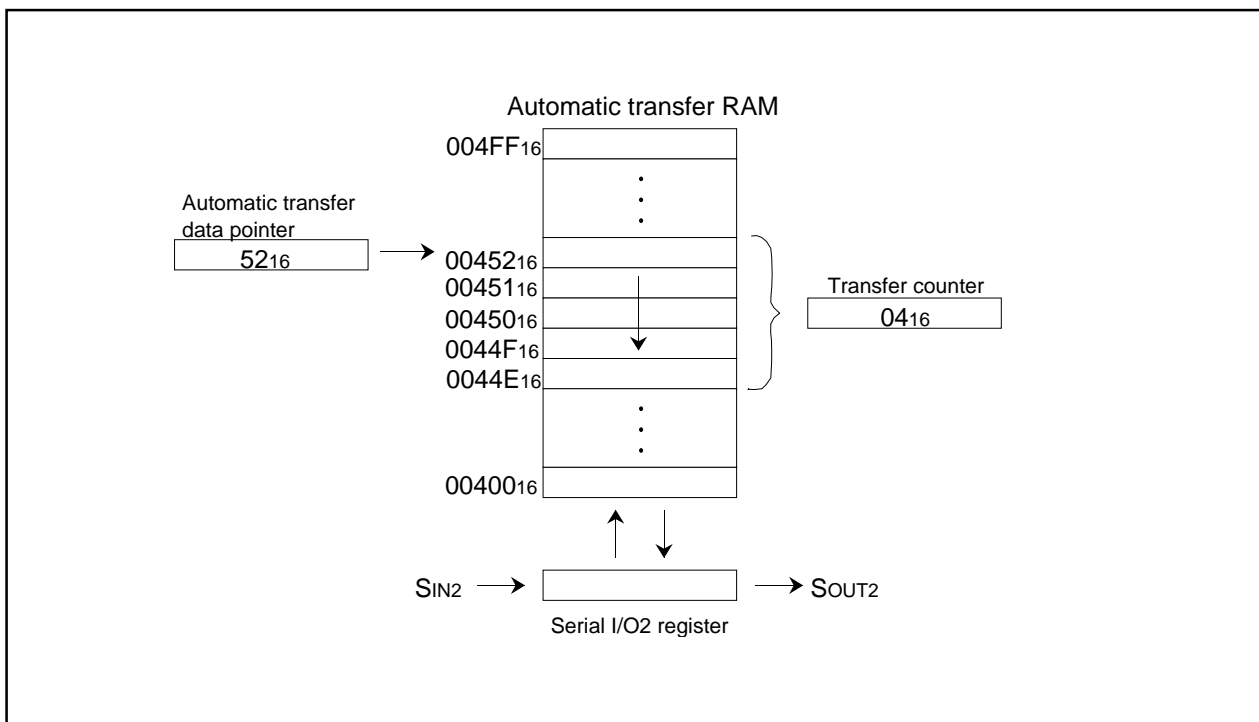
When the internal synchronous clock is selected, automatic serial transfer starts by writing 1 less than the number of transfer bytes to the transfer counter (address 0346<sub>16</sub>). When an external sync clock is selected, automatic serial transfer starts by writing 1 less than the number of transfer bytes to the transfer counter and the transfer clock is input. In this case, allow for at least 5 cycles of internal system clock before the transfer clock is input after writing to the transfer counter.

Also, for data to data transfer intervals, allow at least 5 cycles of internal system clock reckoning from a rise of clock at the last bit of one-byte data.

Regardless of whether the internal or external synchronous clock is selected, the automatic transfer data pointer and the transfer counter are decreased after each 1-byte data is received and then written into the automatic transfer RAM. The serial transfer status flag (bit5 of address 0344<sub>16</sub>) is set to "1" by writing data into the transfer counter. The serial transfer status flag is reset to "0" after the last data is written into the automatic transfer RAM. At the same time, a serial I/O2 interrupt request occurs.

The values written in the automatic transfer data pointer (address 0340<sub>16</sub>) and the automatic transfer interval set bits (bit 0 to bit 4 of address 0348<sub>16</sub>) are held in the latch.

When data is written into the transfer counter, the values latched in the automatic transfer data pointer (address 0340<sub>16</sub>) and the automatic transfer interval set bits (bit 0 to bit 4) are transferred to the decrement counter.



**Figure GA-5. Automatic Transfer Serial I/O Operation**

## Handshake Signal

There are five types of handshake signal : SSTB2 output, SBUSY2 input/output, and SRDY2 input/output.

### (1) SSTB2 output signal

The SSTB2 output is a signal to inform an end of transmission/reception to the serial transfer destination. The SSTB2 output signal can be used only when the internal synchronous clock is selected. In the initial status [ serial I/O initialization bit (bit 4 of address 034216) = "0" ], the SSTB2 output goes to "L" (bits 2, 3 of address 034216=11), or the  $\overline{\text{SSTB2}}$  output goes to "H" (bits 2, 3 of address 034216=10). At the end of transmit/receive operation, after the all data of the serial I/O2 register (address 034616) is output from SOUT2, SSTB2 output is "H" (or  $\overline{\text{SSTB2}}$  output is "L") in the period of 1 cycle of the transfer clock. Furthermore, after 1 cycle, the serial transfer status flag (bit 5 of address 034416) is reset to "0".

In the automatic transfer serial I/O mode, whether the SSTB2 output is to be output at an end of each 1-byte data or after completion of transfer of all data can be selected by the SBUSY2 output • SSTB2 output function select bit (bit 4 of address 034416).

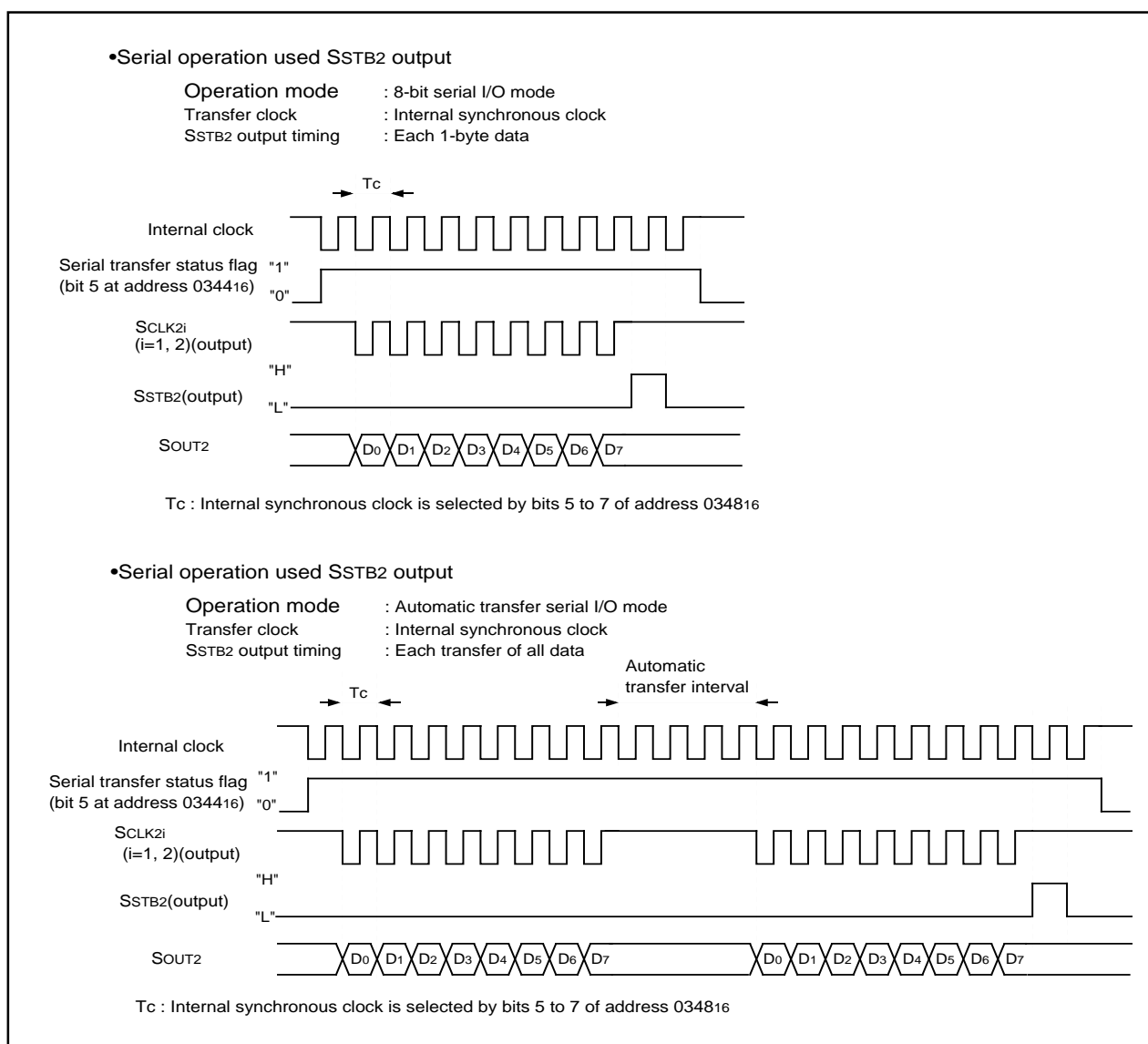
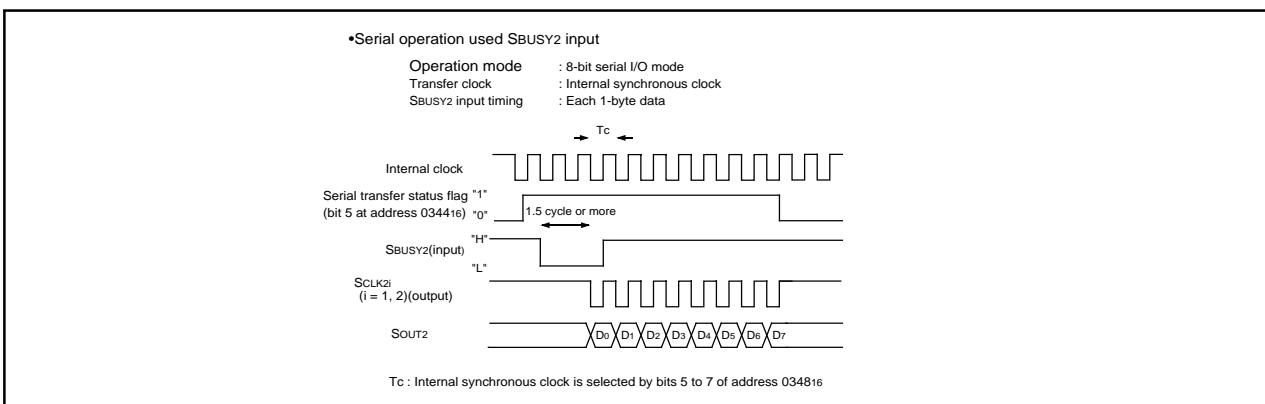


Figure GA-6. SSTB2 Output Operation

**(2) S<sub>BUSY2</sub> input signal**

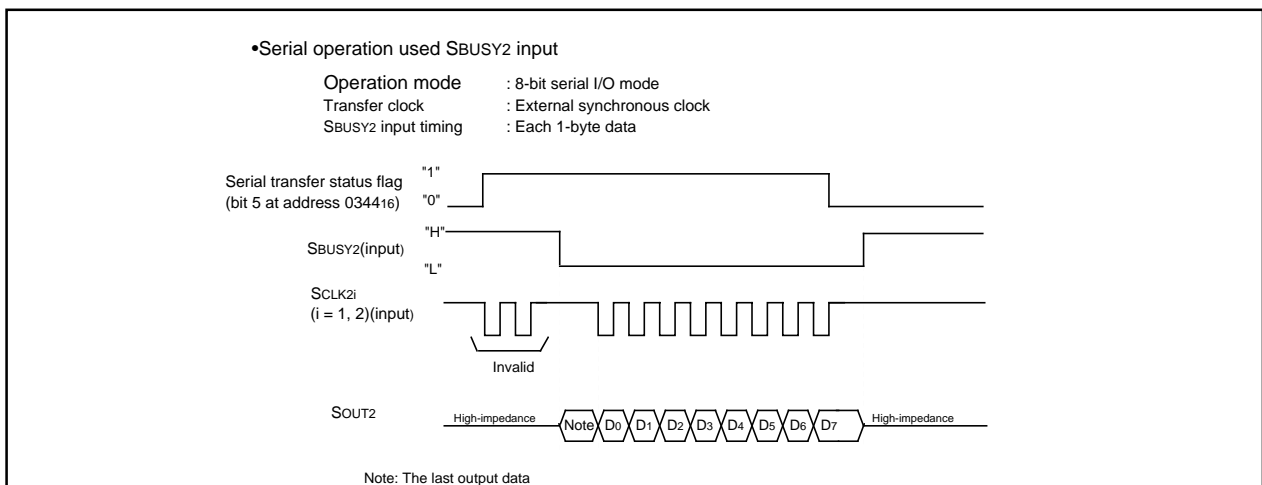
The S<sub>BUSY2</sub> input is a signal requested to stop of transmission/reception from the serial transfer destination.

When the internal synchronous clock is selected, input a "H" level signal into the S<sub>BUSY2</sub> input (or a "L" level signal into the  $\overline{\text{S}}\text{BUSY2}$  input) in the initial status [serial I/O initialization bit (bit 4 of address 034216) = "0"]. When a "L" level signal into the S<sub>BUSY2</sub> ( or "H" on  $\overline{\text{S}}\text{BUSY2}$  ) input for 1.5 cycles or more of transfer clock, transfer clocks are output from SCLK2i (i = 1, 2), and transmit/receive operation is started. When S<sub>BUSY2</sub> input is driven "H" ( or  $\overline{\text{S}}\text{BUSY2}$  input is driven "L" ) during transmit/receive operation, the transfer clock being output from SCLK2i (i = 1, 2) remains active until after the system finishes sending or receiving the designated number of bits, without stopping the transmit/receive operation immediately. The handshake unit of the 8-bit serial I/O is 8 bits, and that of the automatic transfer serial I/O is 8 bits.

**Figure GA-7. S<sub>BUSY2</sub> Input Operation (1)**

When the external synchronous clock is selected, input a "H" level signal into the S<sub>BUSY2</sub> input (or a "L" level signal into the  $\overline{\text{S}}\text{BUSY2}$  input) in the initial status[serial I/O initialization bit (bit 4 of address 034216) = "0"]. At this time, the transfer clock become invalid. The transfer clock become valid while a "L" level signal is input into the S<sub>BUSY2</sub> input (or a "H" level signal into the  $\overline{\text{S}}\text{BUSY2}$  input) and transmit/receive operation work.

When changing the input values into the S<sub>BUSY2</sub> ( or  $\overline{\text{S}}\text{BUSY2}$  ) input at these operations, change them when the transfer clock input is in a "H" state. When the high-impedance of the SOUT2 output is selected by the SOUT2 pin control bit (bit 6 of address 034416), the SOUT2 becomes high-impedance, while a "H" level signal is input into the S<sub>BUSY2</sub> input (or a "L" level signal into the  $\overline{\text{S}}\text{BUSY2}$  input.)

**Figure GA-8. S<sub>BUSY2</sub> Input Operation (2)**

### (3) SBUSY2 output signal

The SBUSY2 output is a signal which requests to stop of transmission/reception to the serial transfer destination. In the automatic transfer serial I/O mode, regardless of the internal or external synchronous clock, whether the SBUSY2 output is to be output at transfer of each 1-byte data or during transfer of all data can be selected by the SBUSY2 output • SSTB2 output function select bit (bit 4 of address 034416). In the initial status[ serial I/O initialization bit (bit 4 of address 034216) = "0" ], the status in which the SBUSY2 outputs "H" (or the  $\overline{\text{SBUSY2}}$  outputs "L").

When the internal synchronous clock is selected, in the 8-bit serial I/O mode and the automatic transfer serial I/O mode (SBUSY2 output function: each 1-byte signal is selected), the SBUSY2 output goes to "L" (or the  $\overline{\text{SBUSY2}}$  output goes to "H") before 0.5 cycle of the timing at which the transfer clock goes to "L". In the automatic transfer serial I/O mode (the SBUSY2 output function: all transfer data is selected), the SBUSY2 output goes to "L" (or the  $\overline{\text{SBUSY2}}$  output goes to "H") when the first transmit data is written into the serial I/O2 register (address 034616).

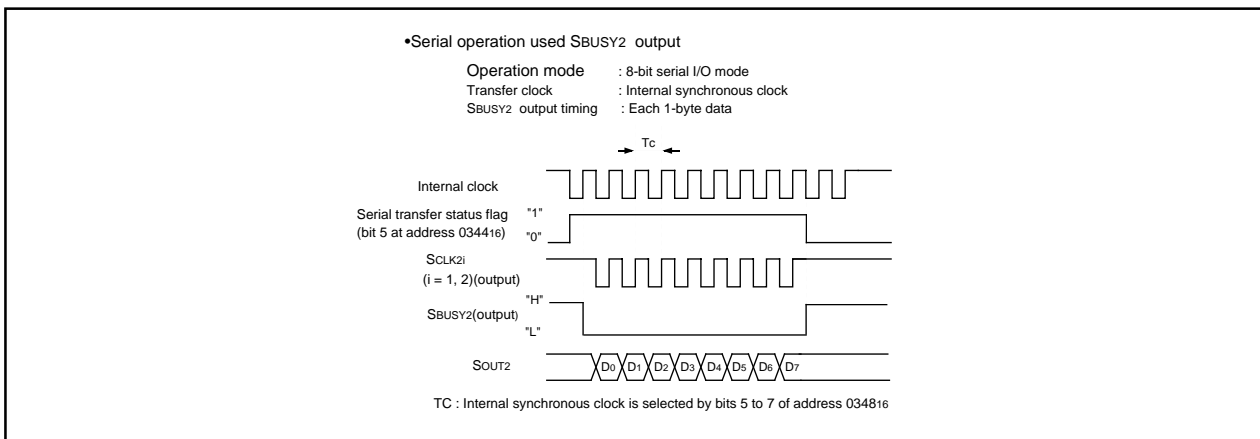


Figure GA-9. SBUSY2 Output Operation (1)

When the external synchronous clock is selected, the SBUSY2 output goes to "L" (or the  $\overline{\text{SBUSY2}}$  output goes to "H") when transmit data is written into the serial I/O2 register(address 034616), regardless of the serial I/O transfer mode.

At termination of transmit/receive operation, in the 8-bit serial I/O mode, the SBUSY2 output goes to "H" (or the  $\overline{\text{SBUSY2}}$  output returns to "L"), when the serial transfer status flag is set to "0", regardless of whether the internal or external synchronous clock is selected. Furthermore, in the automatic transfer serial I/O mode (SBUSY2 output function: each 1-byte signal is selected), the SBUSY2 output goes to "H" (or the  $\overline{\text{SBUSY2}}$  output goes to "L") each time 1-byte of receive data is written into the automatic transfer RAM.

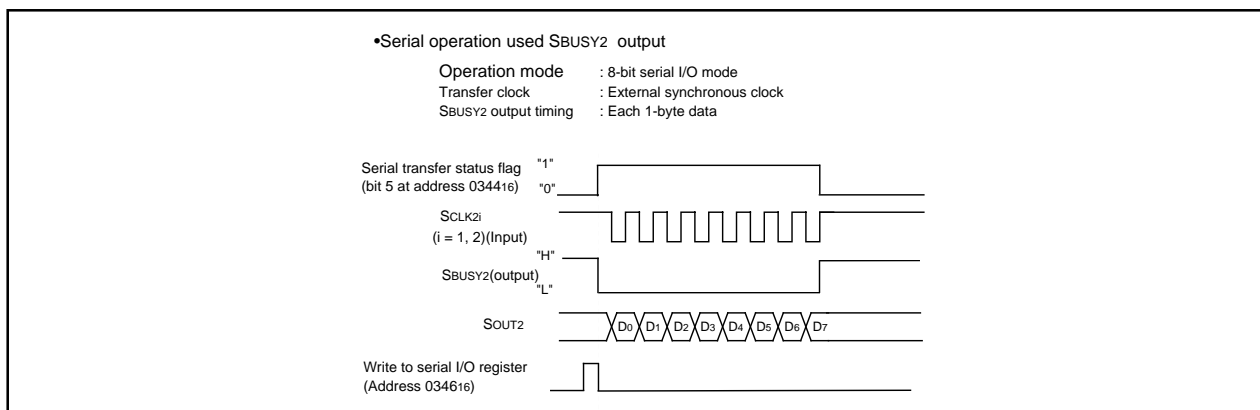


Figure GA-10. SBUSY2 Output Operation (2)

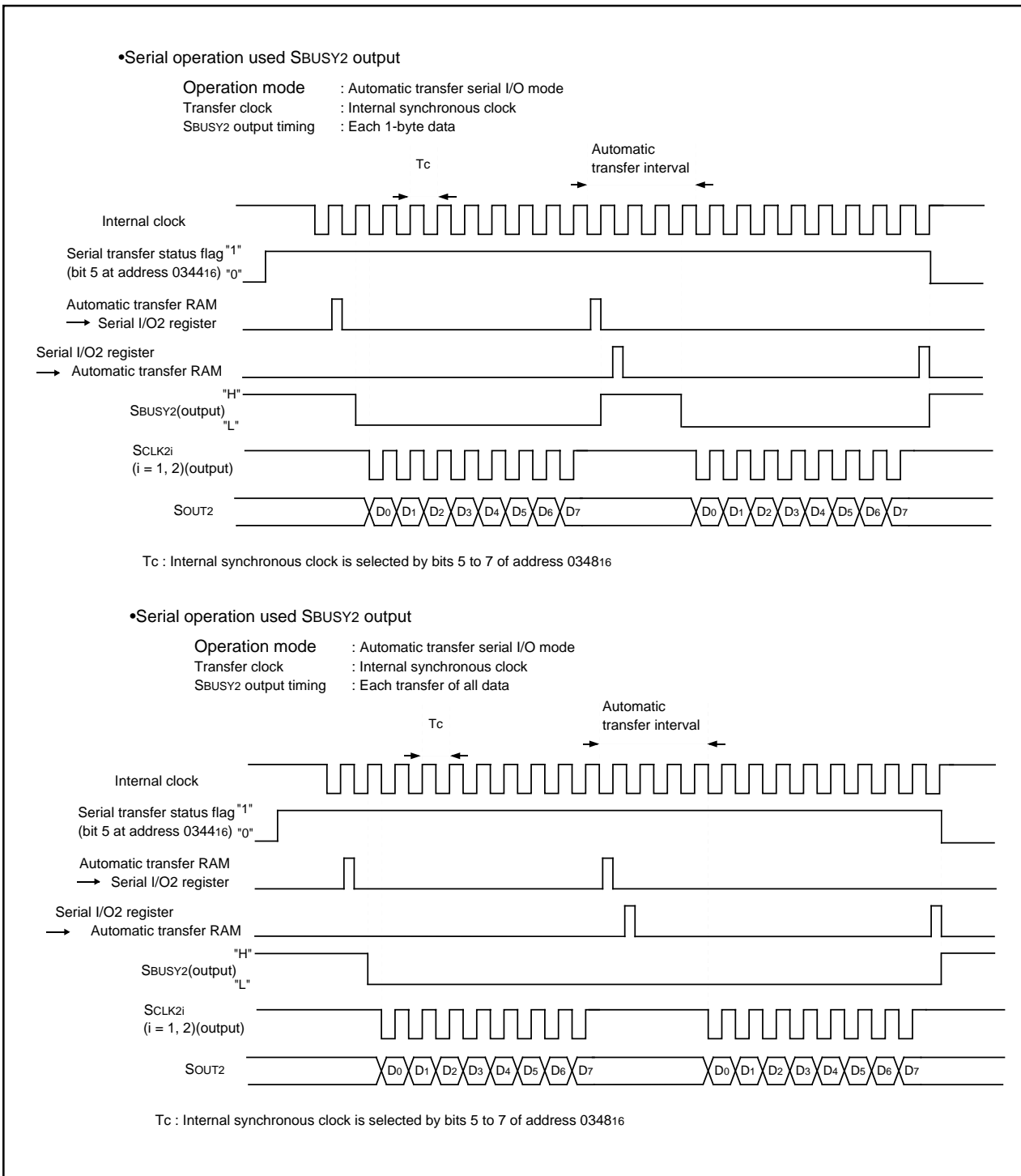


Figure GA-11. S<sub>BUSY2</sub> Output Operation (3)



#### (4) SRDY2 output signal

The SRDY2 output is a transmit/receive enable signal which informs the serial transfer destination that transmit/receive is ready. In the initial status[serial I/O initialization bit (bit 4 of address 034216) = "0" ], the SRDY2 output goes to "L" (or the  $\overline{\text{SRDY2}}$  output goes to "H"). When the transmitted data is written to the serial I/O2 register (address 034616), the SRDY2 output goes to "H" (or the  $\overline{\text{SRDY2}}$  output goes to "L"). When a transmit/receive operation is started and the transfer clock goes to "L", the SRDY2 output goes to "L" (or the  $\overline{\text{SRDY2}}$  output goes to "H").

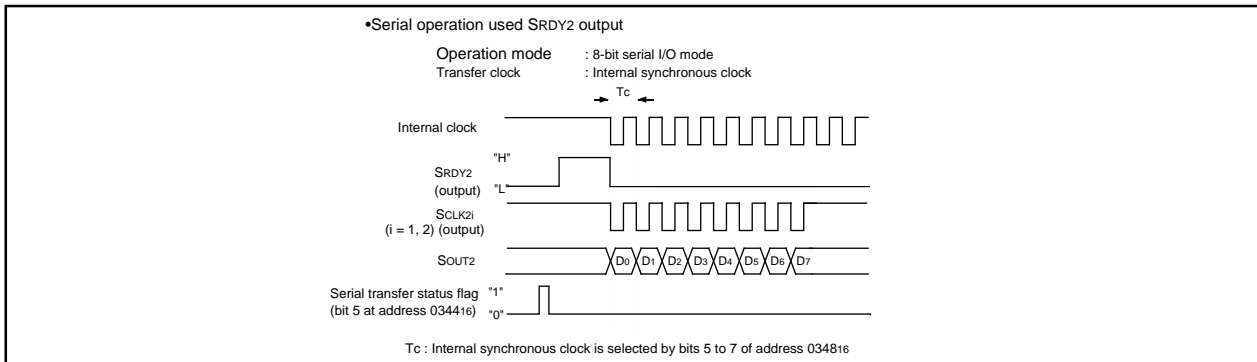


Figure GA-12. SRDY2 Output Operation

#### (5) SRDY2 input signal

The SRDY2 input is a signal for receiving a transmit/receive ready completion signal from the serial transfer destination. The SRDY2 input signal becomes valid only when the SRDY2 input and the SBUSY2 output are used.

When the internal synchronous clock is selected, input a "L" level signal into the SRDY2 input (or a "H" level signal into the  $\overline{\text{SRDY2}}$  input) in the initial status[serial I/O initialization bit (bit 4 of address 034216) = "0" ]. When a "H" level signal is input into the SRDY2 input (or a "L" level signal is input into the  $\overline{\text{SRDY2}}$  input) for a period of 1.5 cycles or more of transfer clock, transfer clocks are output from the SCLK2i (i = 1, 2) output and a transmit/receive operation is started. When SRDY2 input is driven "L" (or  $\overline{\text{SRDY2}}$  input is driven "H") during transmit/receive operation, the transfer clock being output from SCLK2i (i = 1, 2) remains active until after the system finishes sending or receiving the designated number of bits, without stopping the transmit/receive operation immediately.

The handshake unit of the 8-bit serial I/O is 8 bits, and that of the automatic transfer serial I/O is 8 bits. When the external synchronous clock is selected, the SRDY2 input becomes one of the triggers to output the SBUSY2 signal. To start a transmit/receive operation (SBUSY2 output: "L", (or  $\overline{\text{SBUSY2}}$  output: "H")), input a "H" level signal into the SRDY2 input (or a "L" level signal into the  $\overline{\text{SRDY2}}$  input,) and also write transmit data into the serial I/O2 register (address 034616).

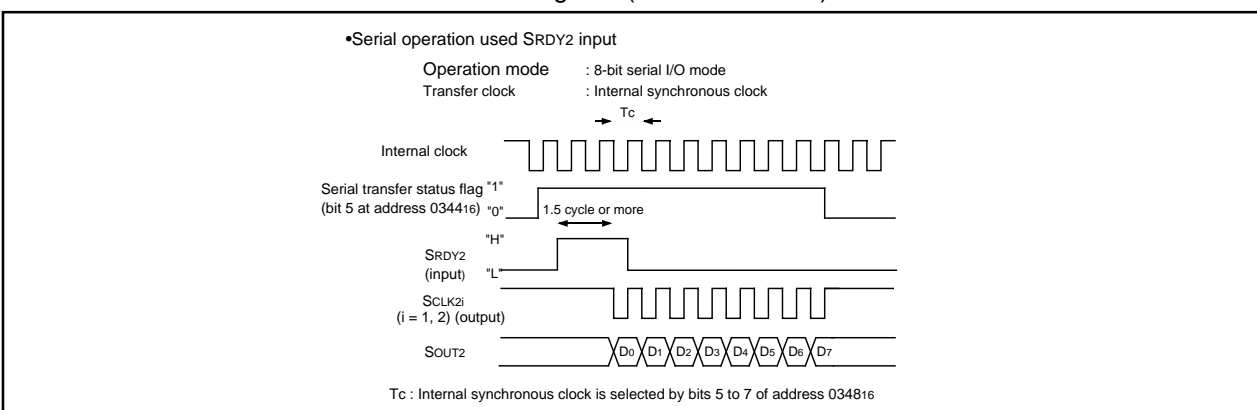
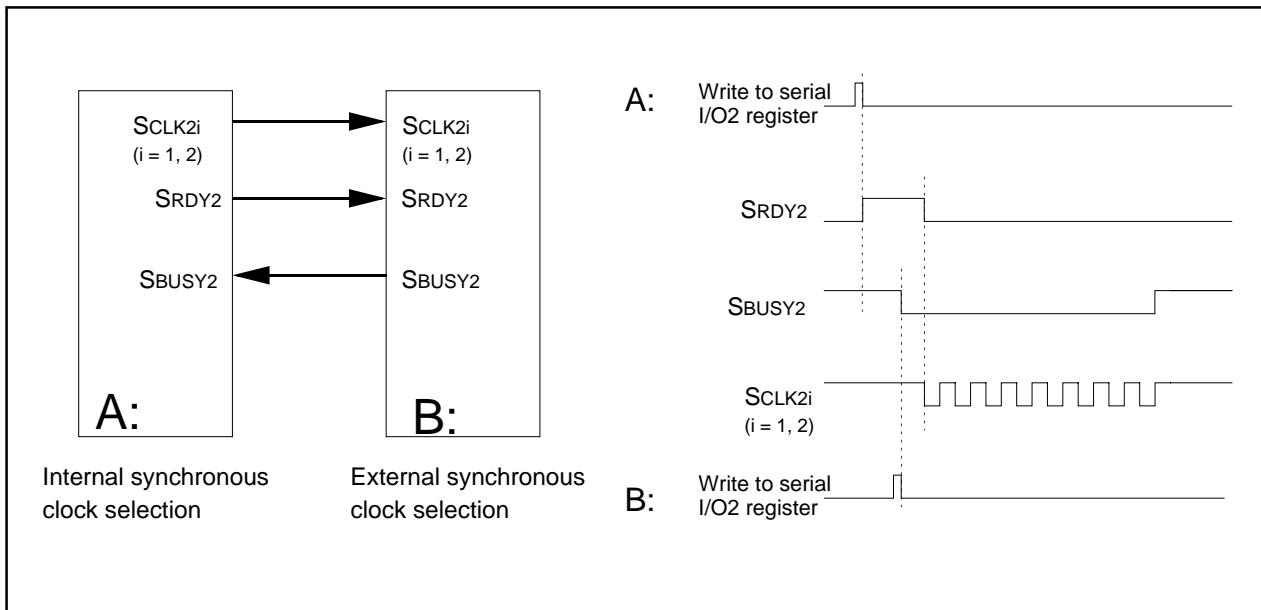
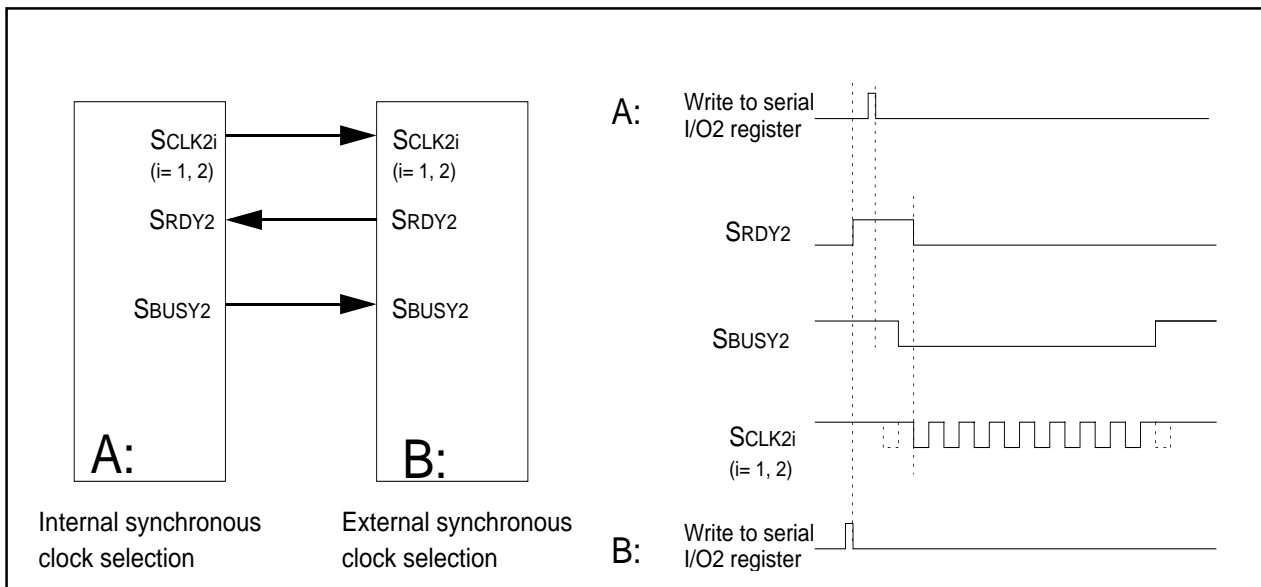


Figure GA-13. SRDY2 Input Operation



**Figure GA-14. Handshake Operation at Serial I/O2 Mutual Connecting (1)**



**Figure GA-15. Handshake Operation at Serial I/O2 Mutual Connecting (2)**

## A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins P10<sub>0</sub> to P10<sub>7</sub> also function as the analog signal input pins. The direction registers of these pins for A-D conversion must therefore be set to input. The Vref connect bit (bit 5 at address 03D7<sub>16</sub>) can be used to isolate the resistance ladder of the A-D converter from the reference voltage input pin (VREF) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from VREF, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of 03D7<sub>16</sub> to connect VREF.

The result of A-D conversion is stored in the A-D registers of the selected pins. When set to 10-bit precision, the low 8 bits are stored in the even addresses and the high 2 bits in the odd addresses. When set to 8-bit precision, the low 8 bits are stored in the even addresses.

Table JA-1 shows the performance of the A-D converter. Figure JA-1 shows the block diagram of the A-D converter, and Figures JA-2 and JA-3 show the A-D converter-related registers.

**Table JA-1. Performance of A-D converter**

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AVCC (VCC)
Operating clock $\phi_{AD}$ (Note 2)	VCC = 5V $f_{AD}$ /divide-by-2 of $f_{AD}$ /divide-by-4 of $f_{AD}$ , $f_{AD}=f(XIN)$ VCC = 3V divide-by-2 of $f_{AD}$ /divide-by-4 of $f_{AD}$ , $f_{AD}=f(XIN)$
Resolution	8-bit or 10-bit (selectable)
Absolute precision	VCC = 5V <ul style="list-style-type: none"> <li>• Without sample and hold function <math>\pm 3LSB</math></li> <li>• With sample and hold function (8-bit resolution) <math>\pm 2LSB</math></li> <li>• Without sample and hold function (10-bit resolution) <math>\pm 3LSB</math></li> </ul> VCC = 3V <ul style="list-style-type: none"> <li>• Without sample and hold function (8-bit resolution)(Note 3) <math>\pm 2LSB</math></li> </ul>
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	8pins (AN <sub>0</sub> to AN <sub>7</sub> )
A-D conversion start condition	•Software trigger A-D conversion starts when the A-D conversion start flag changes to "1"
Conversion speed per pin	•Without sample and hold function 8-bit resolution: 49 $\phi_{AD}$ cycles, 10-bit resolution: 59 $\phi_{AD}$ cycles • With sample and hold function 8-bit resolution: 28 $\phi_{AD}$ cycles, 10-bit resolution: 33 $\phi_{AD}$ cycles

Note 1: Does not depend on use of sample and hold function.

Note 2: Without sample and hold function, set the  $\phi_{AD}$  frequency to 250kHz min.

With the sample and hold function, set the  $\phi_{AD}$  frequency to 1MHz min.

Note 3: Only mask ROM version.

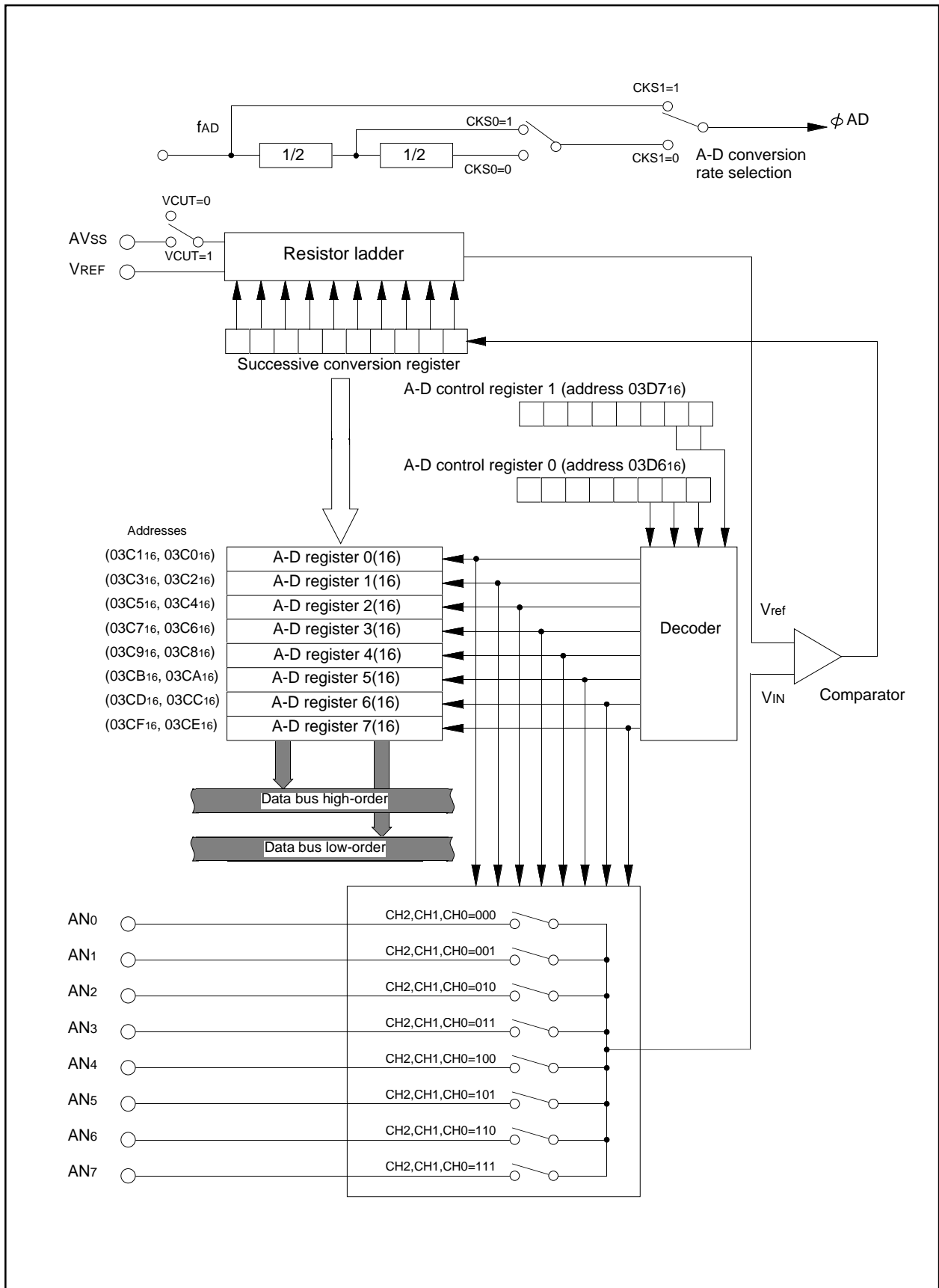


Figure JA-1. Block diagram of A-D converter

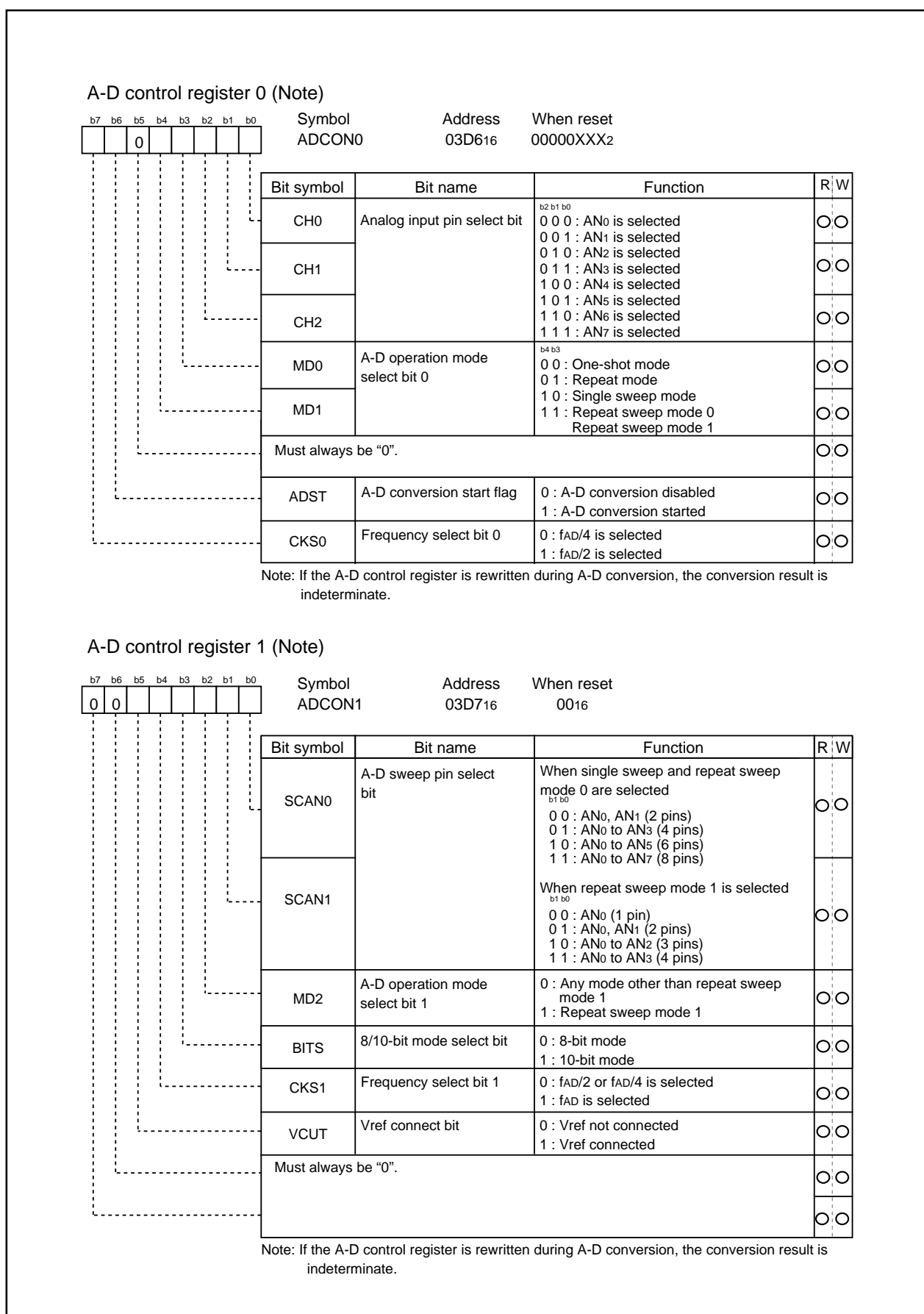


Figure JA-2. A-D converter-related registers (1)

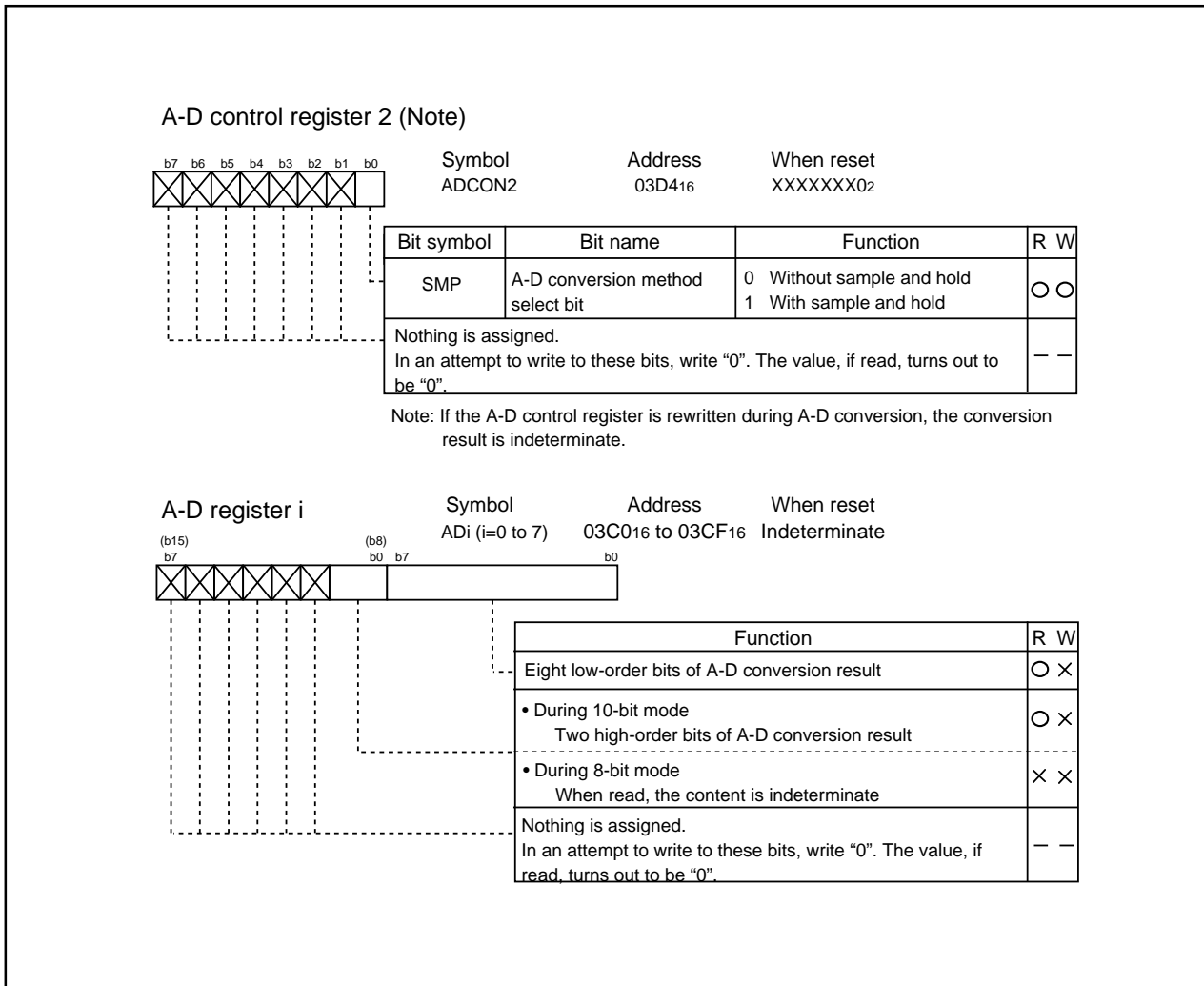


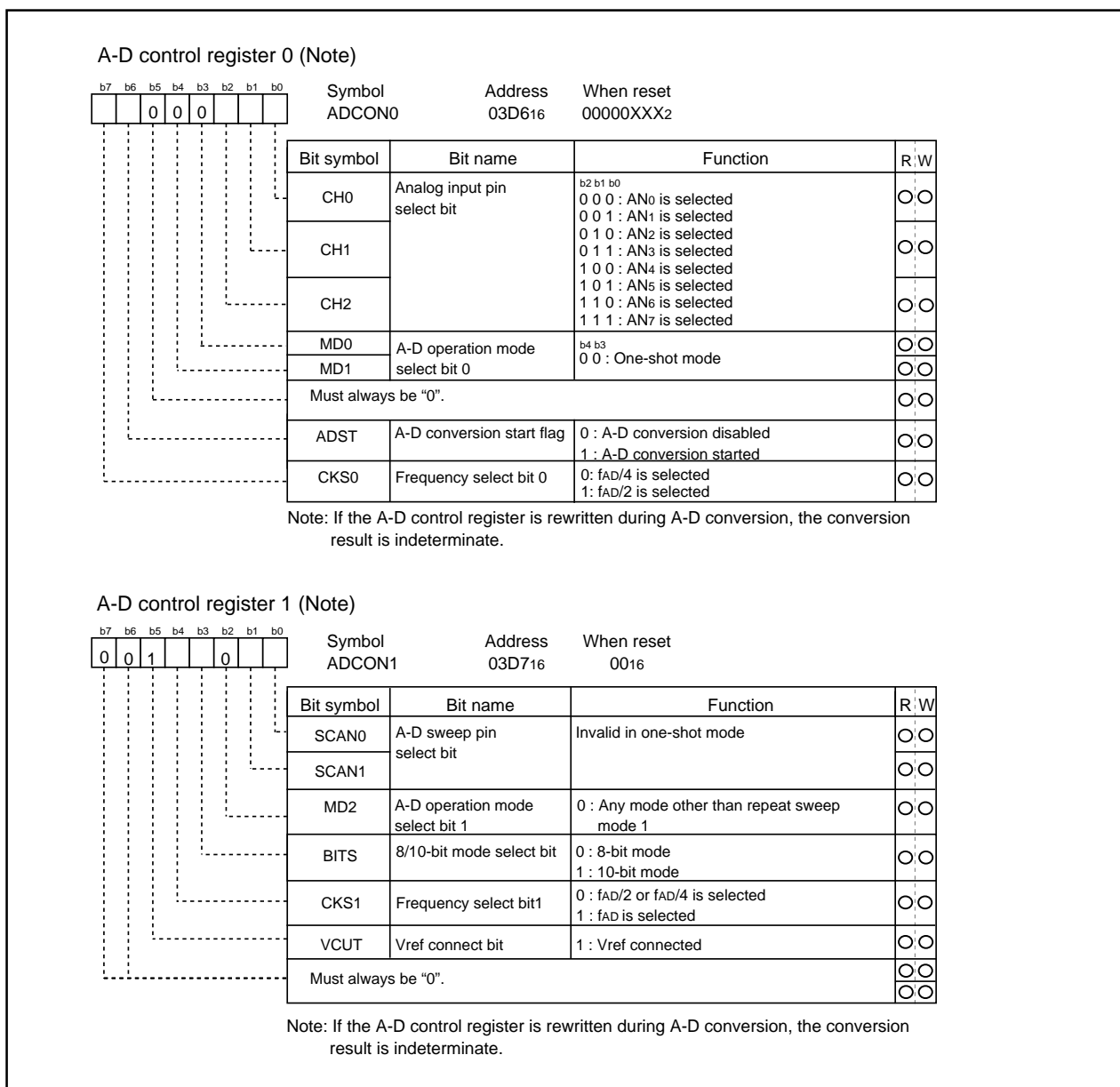
Figure JA-3. A-D converter-related registers (2)

### (1) One-shot mode

In one-shot mode, the pin selected using the analog input pin select bit is used for one-shot A-D conversion. Table JA-2 shows the specifications of one-shot mode. Figure JA-4 shows the A-D control register in one-shot mode.

**Table JA-2. One-shot mode specifications**

Item	Specification
Function	The pin selected by the analog input pin select bit is used for one A-D conversion
Start condition	Writing "1" to A-D conversion start flag
Stop condition	<ul style="list-style-type: none"> <li>End of A-D conversion (A-D conversion start flag changes to "0")</li> <li>Writing "0" to A-D conversion start flag</li> </ul>
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN0 to AN7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin



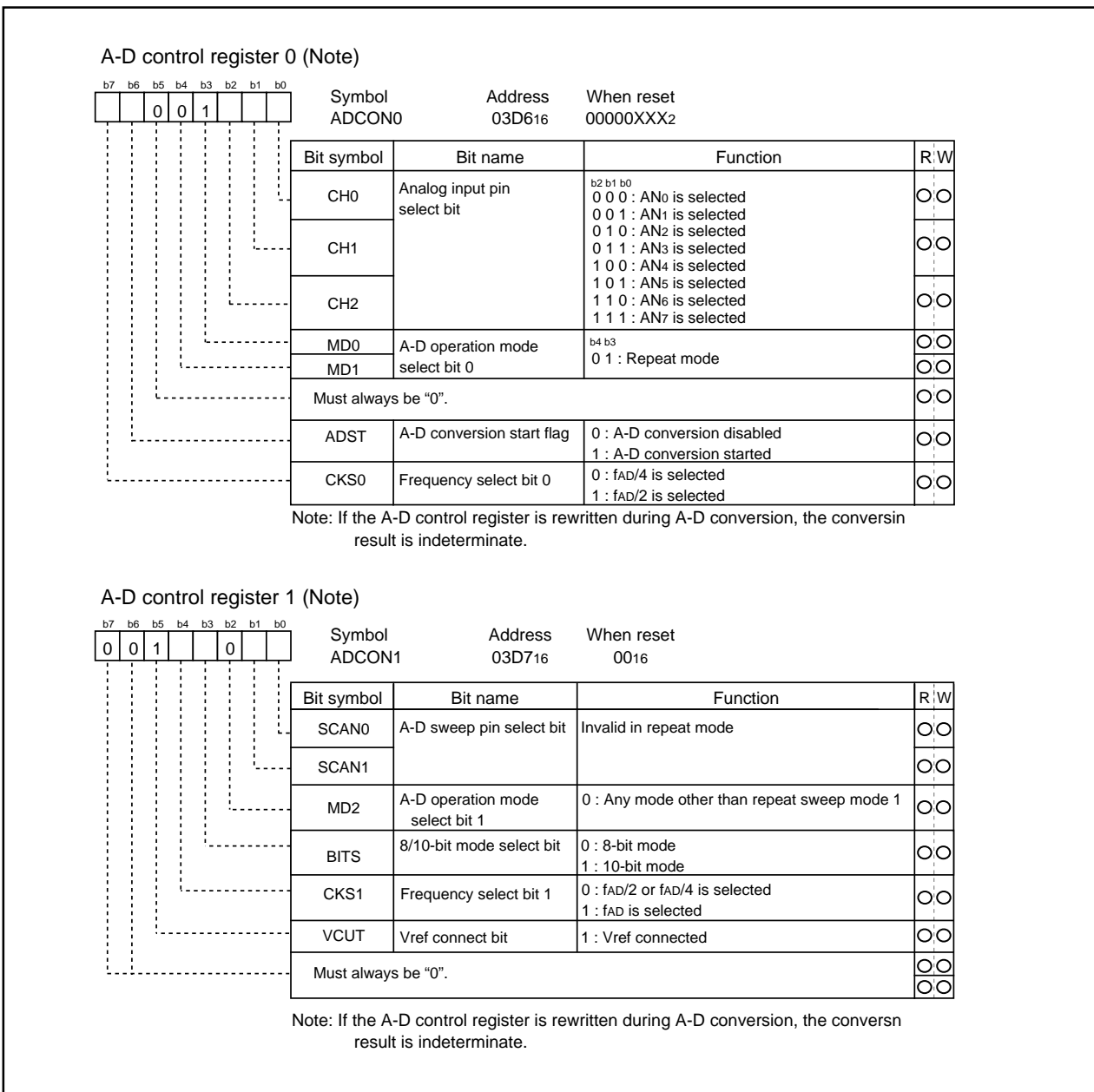
**Figure JA-4. A-D conversion register in one-shot mode**

**(2) Repeat mode**

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A-D conversion. Table JA-3 shows the specifications of repeat mode. Figure JA-5 shows the A-D control register in repeat mode.

**Table JA-3. Repeat mode specifications**

Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated A-D conversion
Star condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	One of AN0 to AN7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin



**Figure JA-5. A-D conversion register in repeat mode**

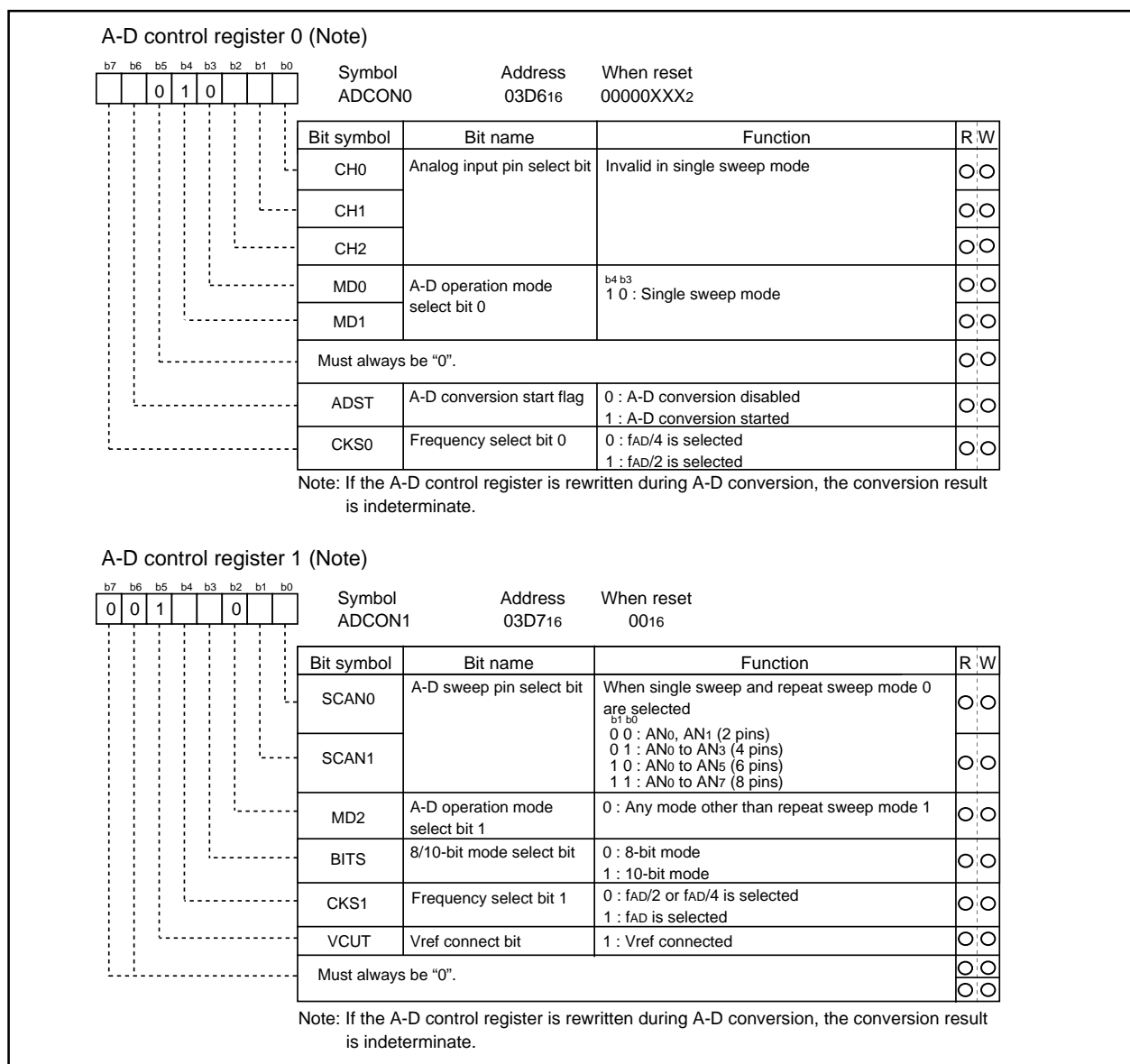


### (3) Single sweep mode

In single sweep mode, the pins selected using the A-D sweep pin select bit are used for one-by-one A-D conversion. Table JA-4 shows the specifications of single sweep mode. Figure JA-6 shows the A-D control register in single sweep mode.

**Table JA-4. Single sweep mode specifications**

Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for one-by-one A-D conversion
Start condition	Writing "1" to A-D converter start flag
Stop condition	<ul style="list-style-type: none"> <li>•End of A-D conversion (A-D conversion start flag changes to "0", except when external trigger is selected)</li> <li>•Writing "0" to A-D conversion start flag</li> </ul>
Interrupt request generation timing	End of A-D conversion
Input pin	AN0 and AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (8 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin



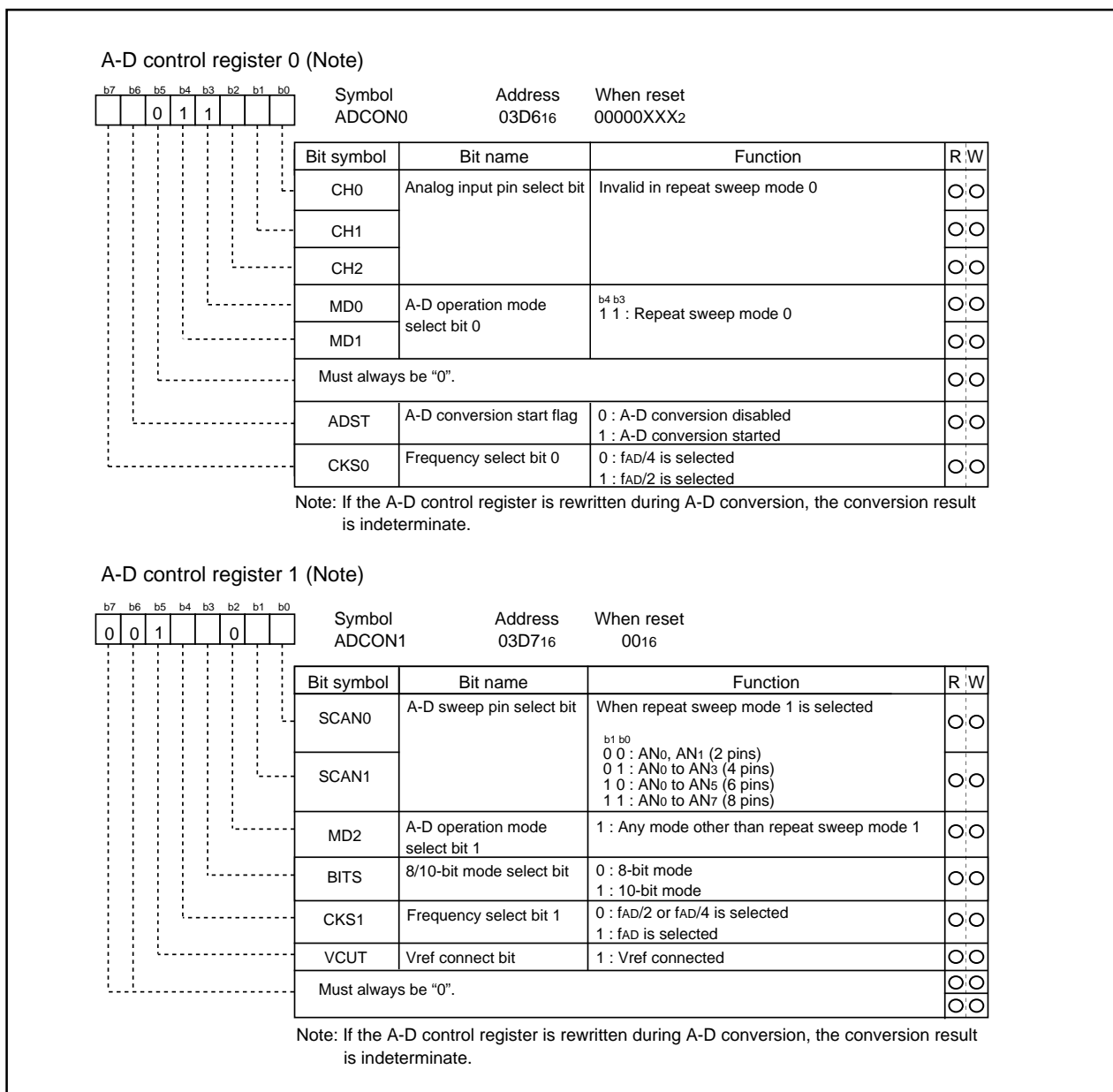
**Figure JA-6. A-D conversion register in single sweep mode**

### (4) Repeat sweep mode 0

In repeat sweep mode 0, the pins selected using the A-D sweep pin select bit are used for repeat sweep A-D conversion. Table JA-5 shows the specifications of repeat sweep mode 0. Figure JA-7 shows the A-D control register in repeat sweep mode 0.

**Table JA-5. Repeat sweep mode 0 specifications**

Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for repeat sweep A-D conversion
Start condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN <sub>0</sub> and AN <sub>1</sub> (2 pins), AN <sub>0</sub> to AN <sub>3</sub> (4 pins), AN <sub>0</sub> to AN <sub>5</sub> (6 pins), or AN <sub>0</sub> to AN <sub>7</sub> (8 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)



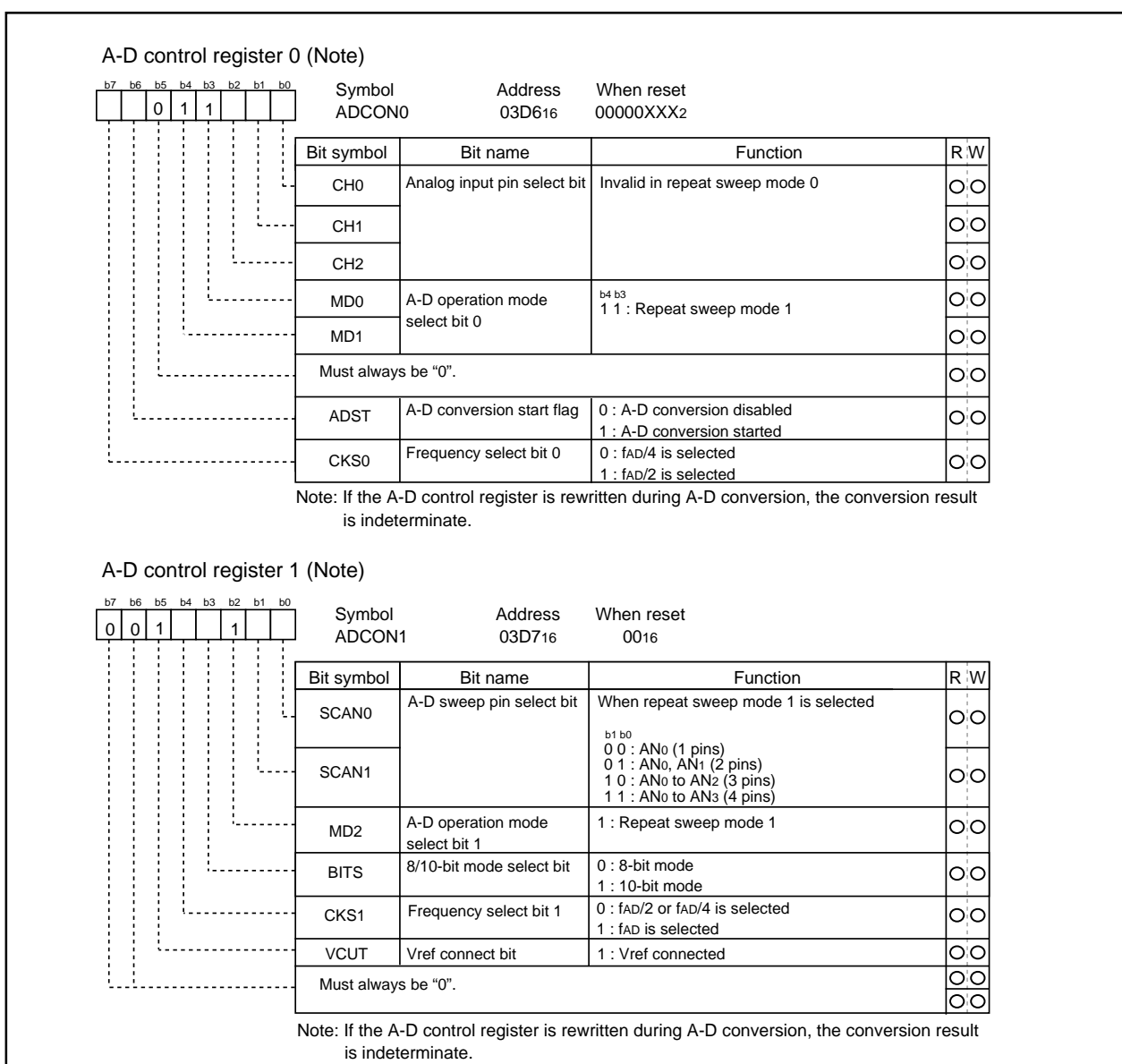
**Figure JA-7. A-D conversion register in repeat sweep mode 0**

### (5) Repeat sweep mode 1

In repeat sweep mode 1, all pins are used for A-D conversion with emphasis on the pin or pins selected using the A-D sweep pin select bit. Table JA-6 shows the specifications of repeat sweep mode 1. Figure JA-8 shows the A-D control register in repeat sweep mode 1.

**Table JA-6. Repeat sweep mode 1 specifications**

Item	Specification
Function	All pins perform repeat sweep A-D conversion, with emphasis on the pin or pins selected by the A-D sweep pin select bit Example : AN <sub>0</sub> selected -> AN <sub>0</sub> -> AN <sub>1</sub> -> AN <sub>0</sub> -> AN <sub>2</sub> -> AN <sub>0</sub> -> AN <sub>3</sub> , etc
Start condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	Emphasis on the pin AN <sub>0</sub> (1 pin), AN <sub>0</sub> and AN <sub>1</sub> (2 pins), AN <sub>0</sub> to AN <sub>2</sub> (3 pins), AN <sub>0</sub> to AN <sub>3</sub> (4 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)



**Figure JA-8. A-D conversion register in repeat sweep mode 1**

**(a) Sample and hold**

Sample and hold is selected by setting bit 0 of the A-D control register 2 (address 03D4<sub>16</sub>) to "1". When sample and hold is selected, the rate of conversion of each pin increases. As a result, 28  $\phi$  AD cycles are achieved with 8-bit resolution and 33  $\phi$  AD cycles with 10-bit resolution. Sample and hold can be selected in all modes. However, in all modes, be sure to specify before starting A-D conversion whether sample and hold is to be used.

## D-A converter

**D-A Converter**

This is an 8-bit, R-2R type D-A converter. The microcomputer contains two independent D-A converters of this type. D-A conversion is performed when a value is written to the corresponding D-A register. Bits 0 and 1 (D-A output enable bits) of the D-A control register decide if the result of conversion is to be output. Do not set the target port to output mode if D-A conversion is to be performed.

Output analog voltage (V) is determined by a set value (n : decimal) in the D-A register.

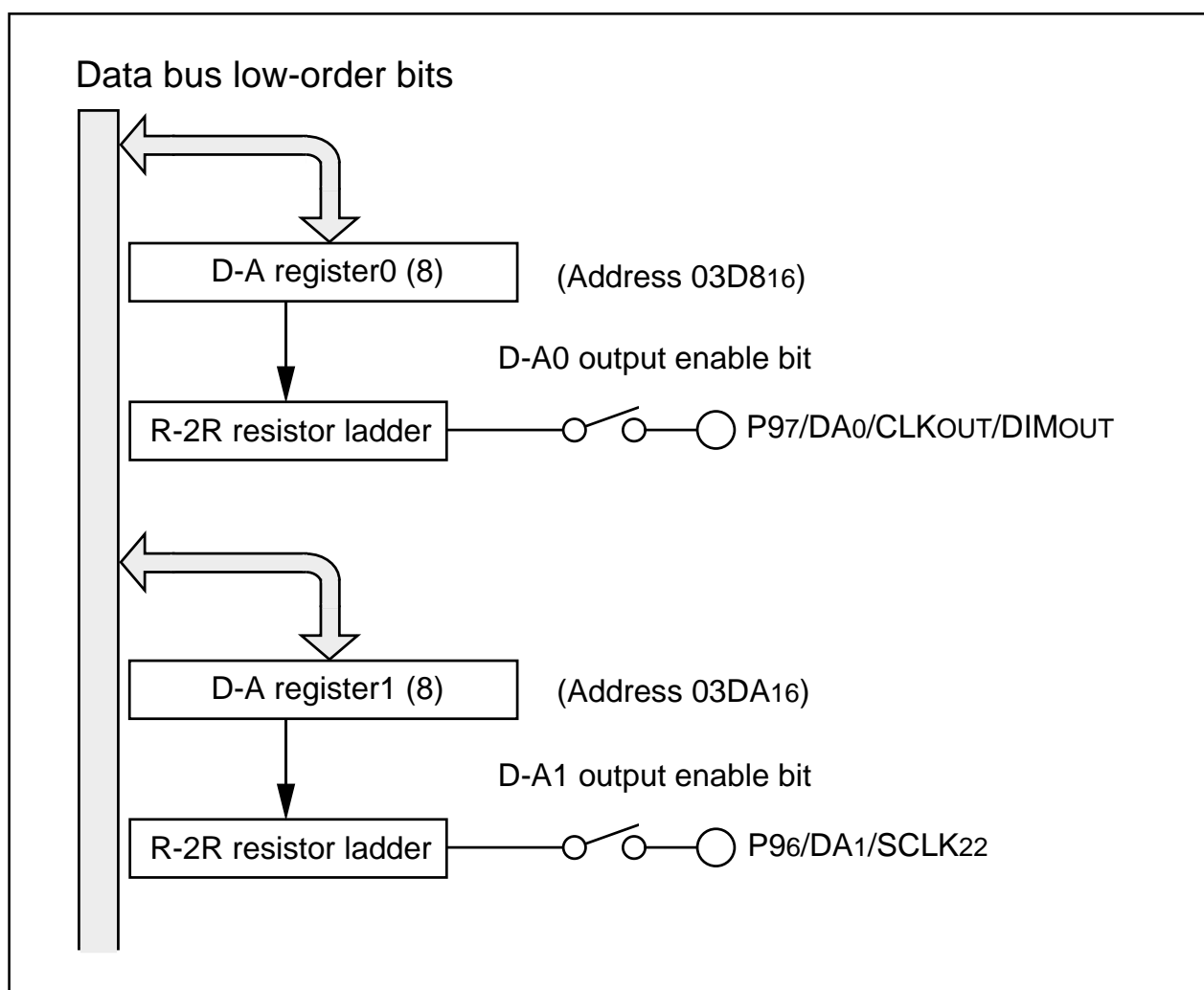
$$V = V_{REF} \times n / 256 \quad (n = 0 \text{ to } 255)$$

$V_{REF}$  : reference voltage

Table JB-1 lists the performance of the D-A converter. Figure JB-1 shows the block diagram of the D-A converter. Figure JB-2 shows the D-A control register. Figure JB-3 shows the D-A converter equivalent circuit.

**Table JB-1. Performance of D-A converter**

Item	Performance
Conversion method	R-2R method
Resolution	8 bits
Analog output pin	2 channels



**Figure JB-1. Block diagram of D-A converter**

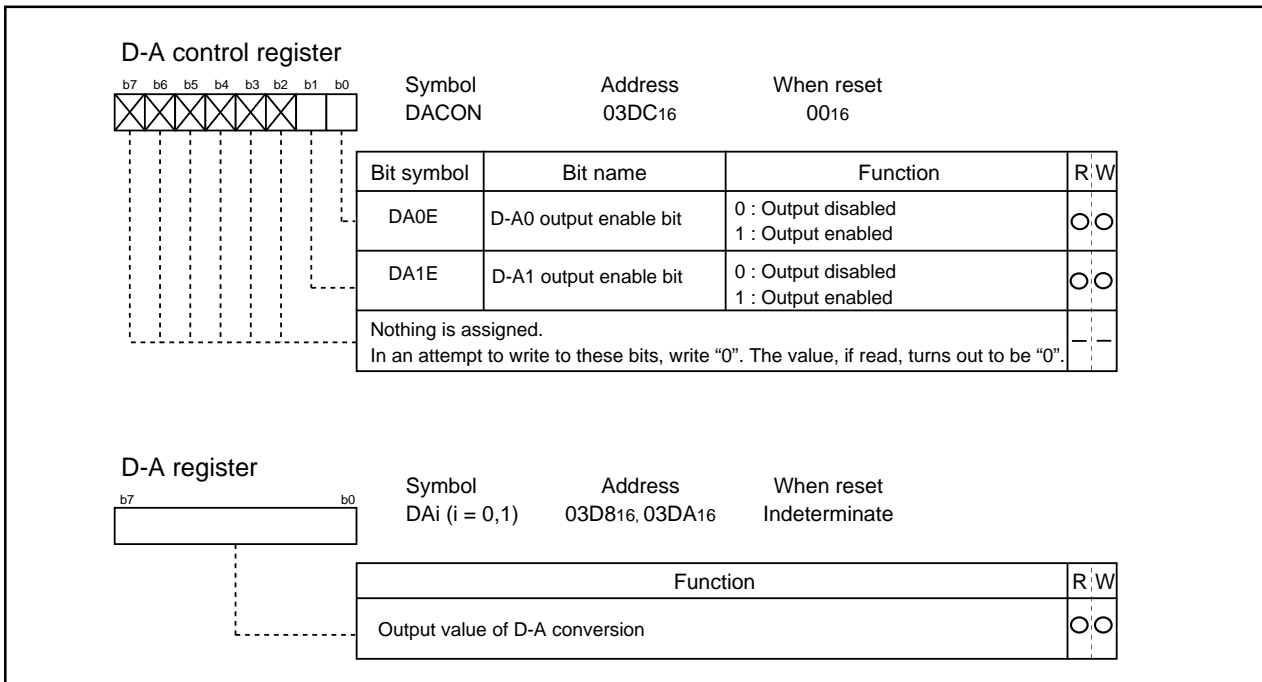


Figure JB-2. D-A control register

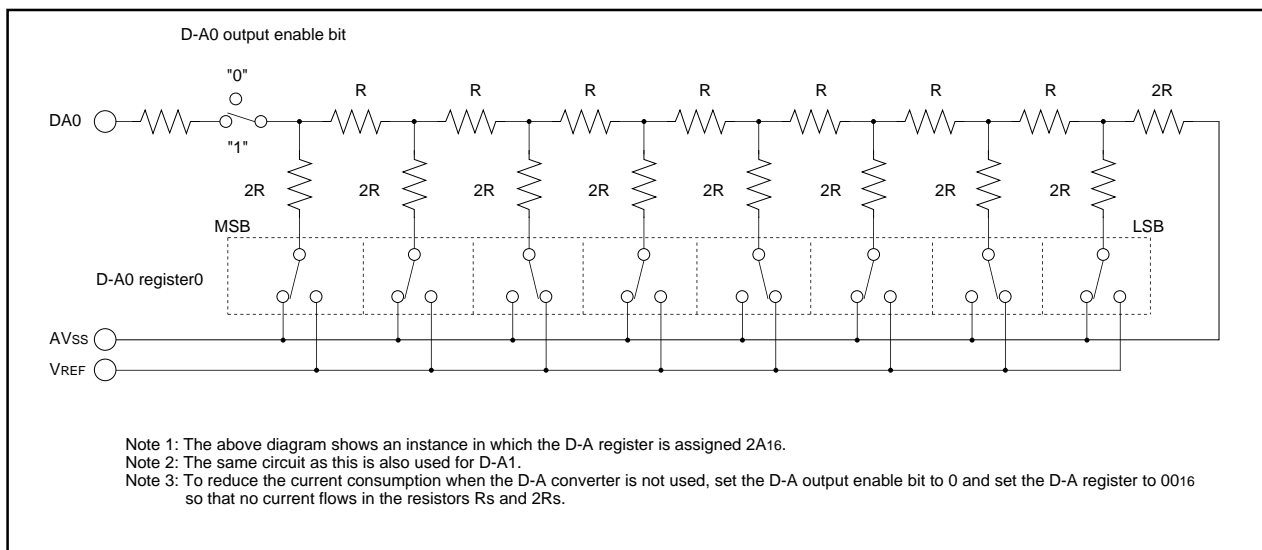


Figure JB-3. D-A converter equivalent circuit

CRC Calculation Circuit

**CRC Calculation Circuit**

The Cyclic Redundancy Check (CRC) calculation circuit 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 is a 16-bit code generated for a block of a given data length in multiples of 8 bits. The CRC code is set in a CRC data register each time one byte of data is transferred to a CRC input register after writing an initial value into the CRC data register. Generation of CRC code for one byte of data is completed in two machine cycles.

Figure UC-1 shows the block diagram of the CRC circuit. Figure UC-2 shows the CRC-related registers. Figure UC-3 shows the calculation example using the CRC calculation circuit

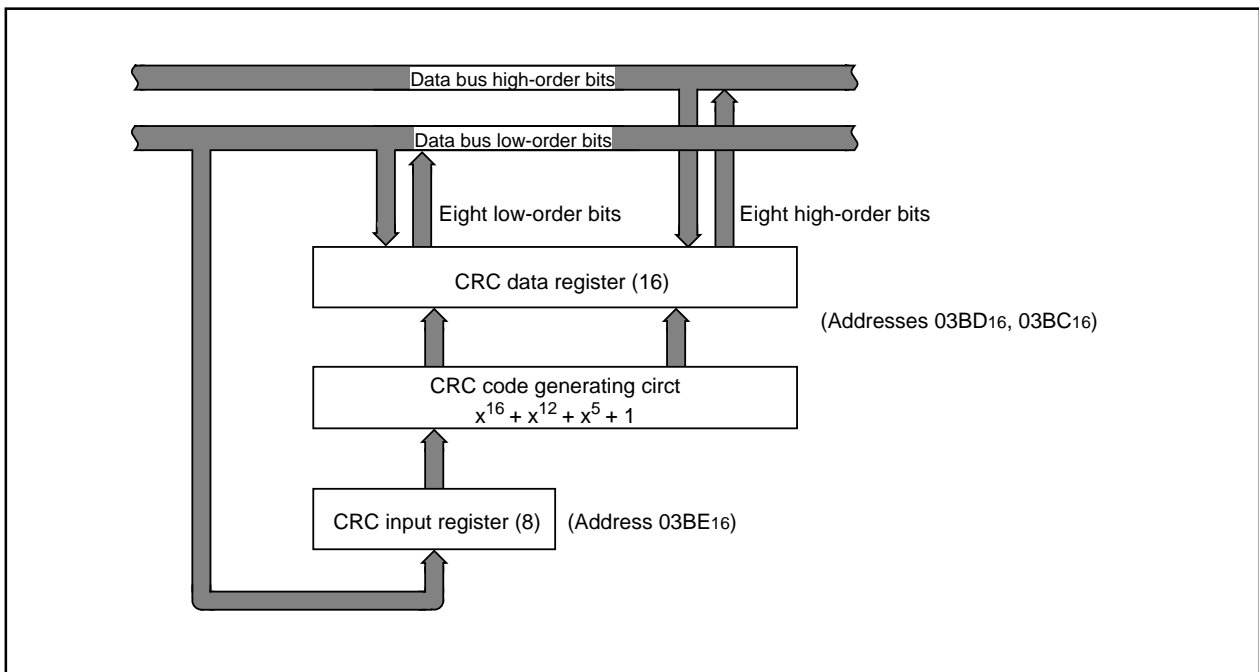


Figure UC-1. Block diagram of CRC circuit

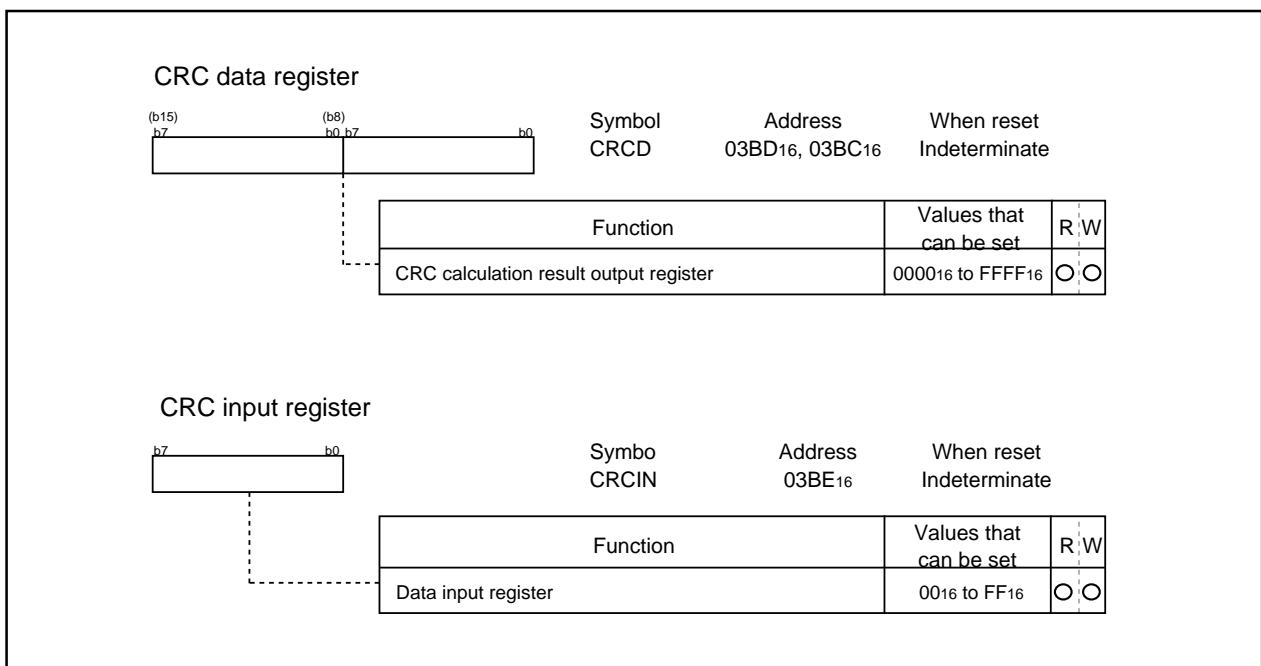


Figure UC-2. CRC-related registers





## Programmable I/O Ports

There are 48 programmable I/O ports: P3, P4 and P7 to P10. Each port can be set independently for input or output using the direction register. A pull-up resistance for each block of 4 ports can be set.

P3 and P40 to P43 are high-breakdown-voltage, P-channel open drain outputs, and have no built-in pull-down resistance.

Figures UA-1, UA-2 show the programmable I/O ports.

Each pin functions as a programmable I/O port and as the I/O for the built-in peripheral devices.

To use the pins as the inputs for the built-in peripheral devices, set the direction register of each pin to input mode. When the pins are used as the outputs for the built-in peripheral devices (other than the D-A converter), they function as outputs regardless of the contents of the direction registers. When pins are to be used as the outputs for the D-A converter, do not set the direction registers to output mode. See the descriptions of the respective functions for how to set up the built-in peripheral devices.

### (1) Direction registers

Figure UA-3 shows the direction registers.

These registers are used to choose the direction of the programmable I/O ports. Each bit in these registers corresponds one for one to each I/O pin.

### (2) Port registers

Figure UA-4 shows the port registers.

These registers are used to write and read data for input and output to and from an external device. A port register consists of a port latch to hold output data and a circuit to read the status of a pin. Each bit in port registers corresponds one for one to each I/O pin.

### (3) Pull-up control registers

Figure UA-5 shows the pull-up control registers.

The pull-up control register can be set to apply a pull-up resistance to each block of 4 ports. When ports are set to have a pull-up resistance, the pull-up resistance is connected only when the direction register is set for input.

Note: P3, P40 to P43 have no built-in pull-up resistance, because of these pin's are high-breakdown-voltage, P-channel open drain outputs.

## Exclusive High-breakdown-voltage Output Ports

There are 40 exclusive output Ports: P0 to P2, P5 and P6.

All ports have structure of high-breakdown-voltage P-channel open drain output. Exclusive output ports except P2 have built-in pull-down resistance.

Figure UA-1 shows the configuration of the exclusive high-breakdown-voltage output ports.

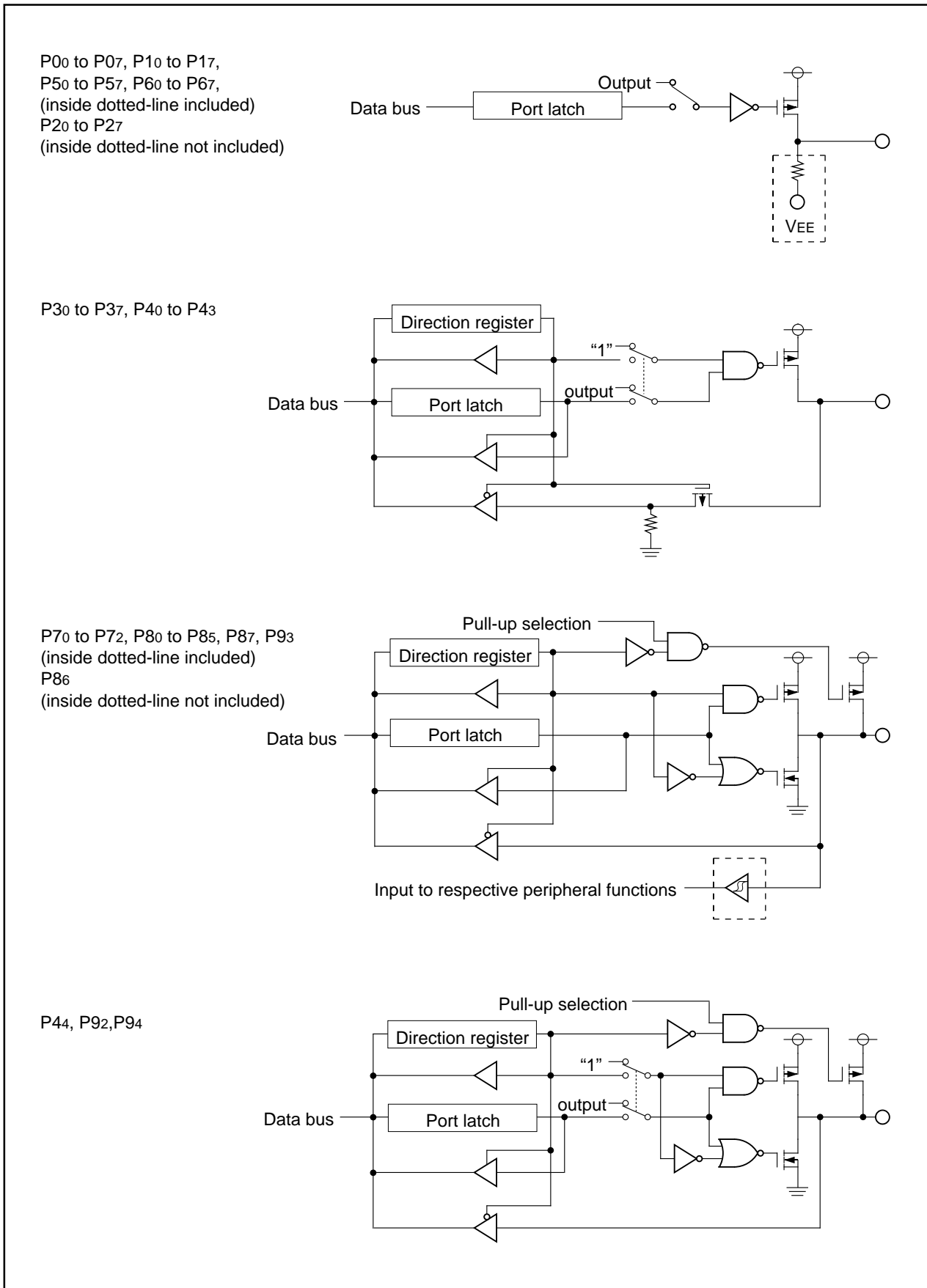


Figure UA-1. Programmable I/O ports (1)

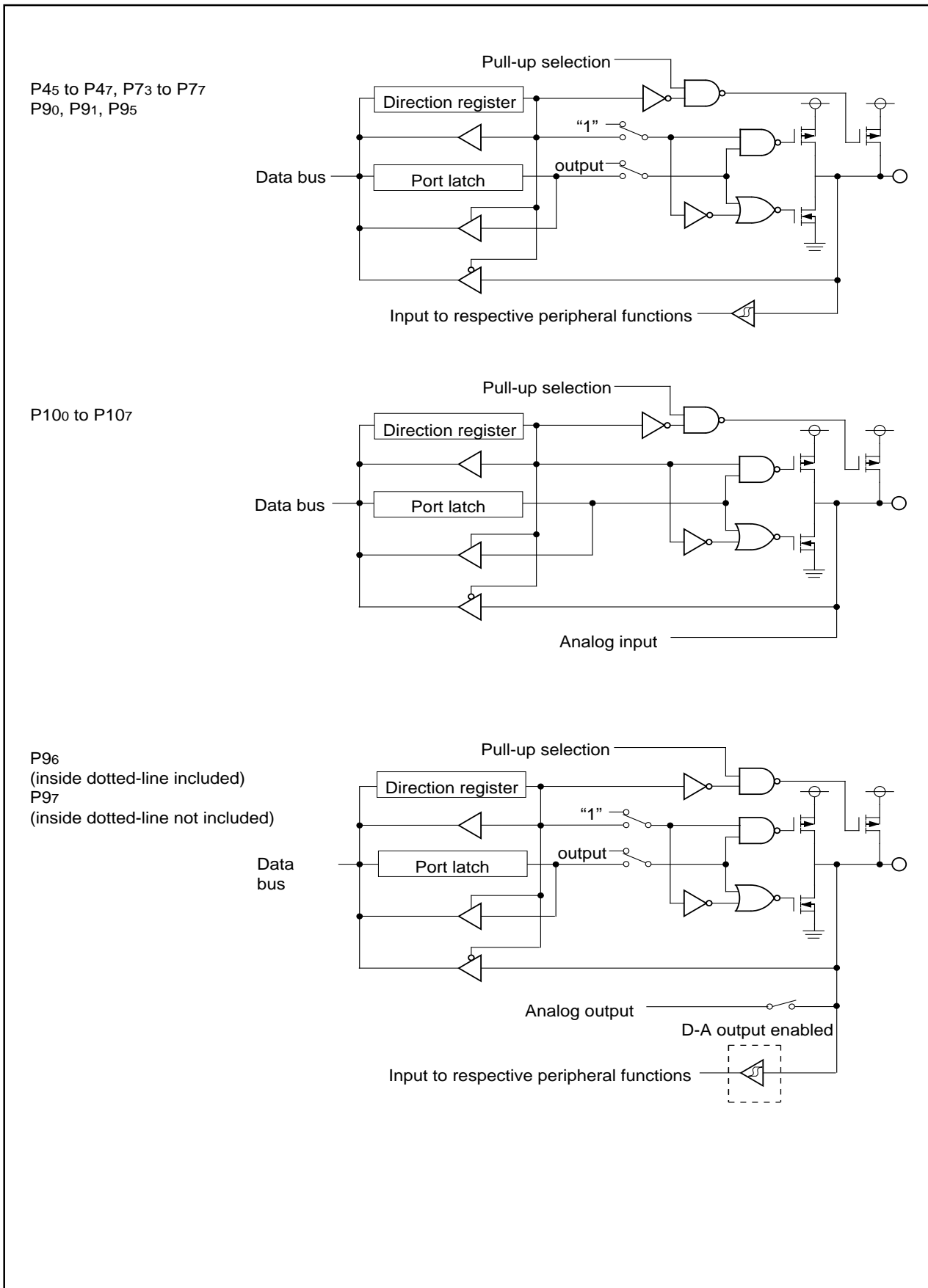


Figure UA-2. Programmable I/O ports (2)

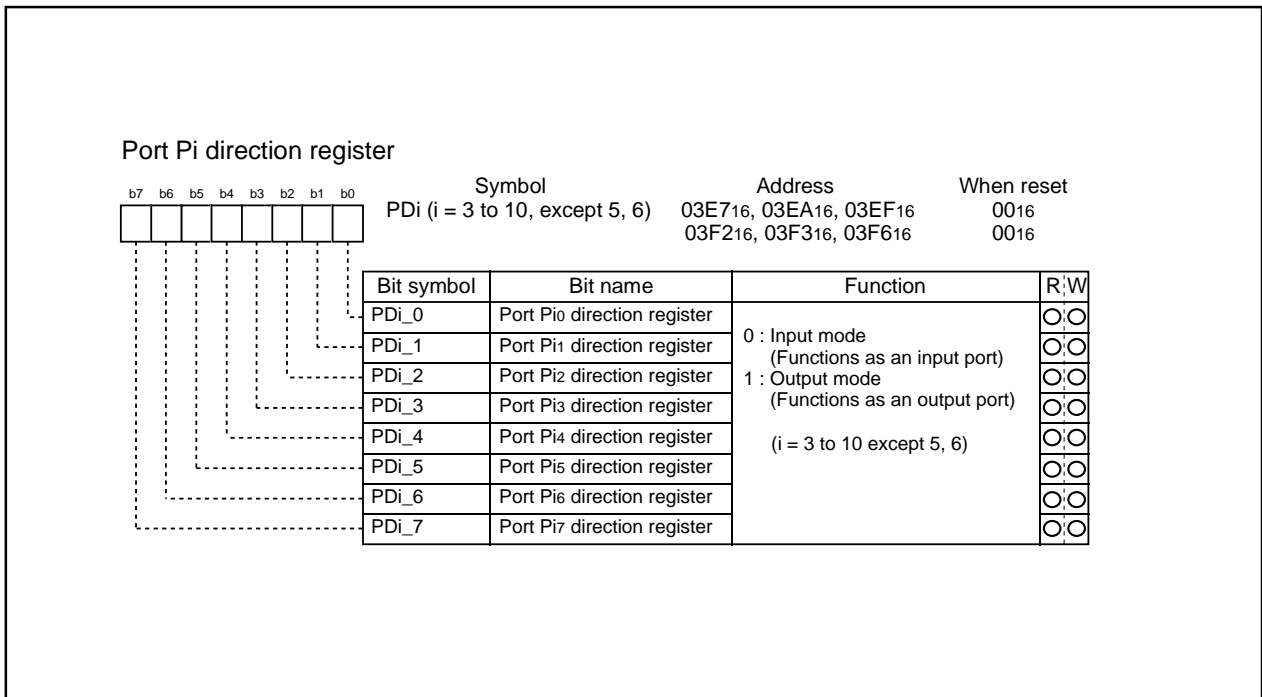


Figure UA-3. Direction register

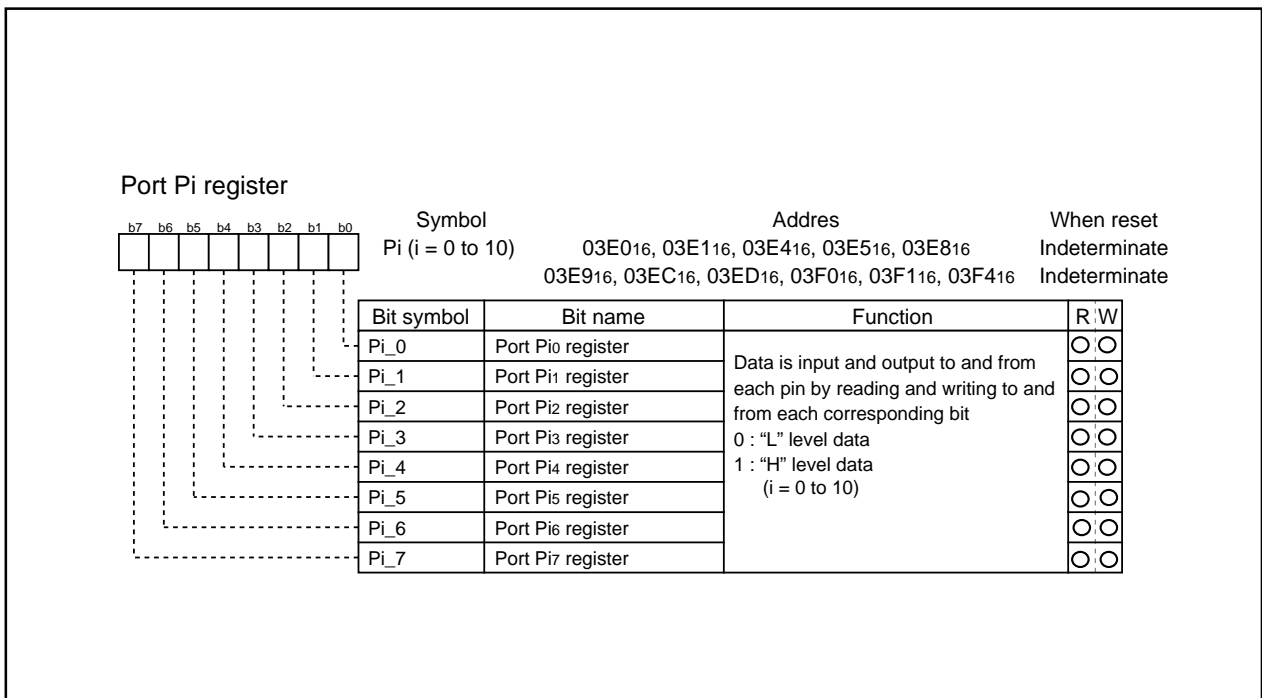


Figure UA-4. Port register

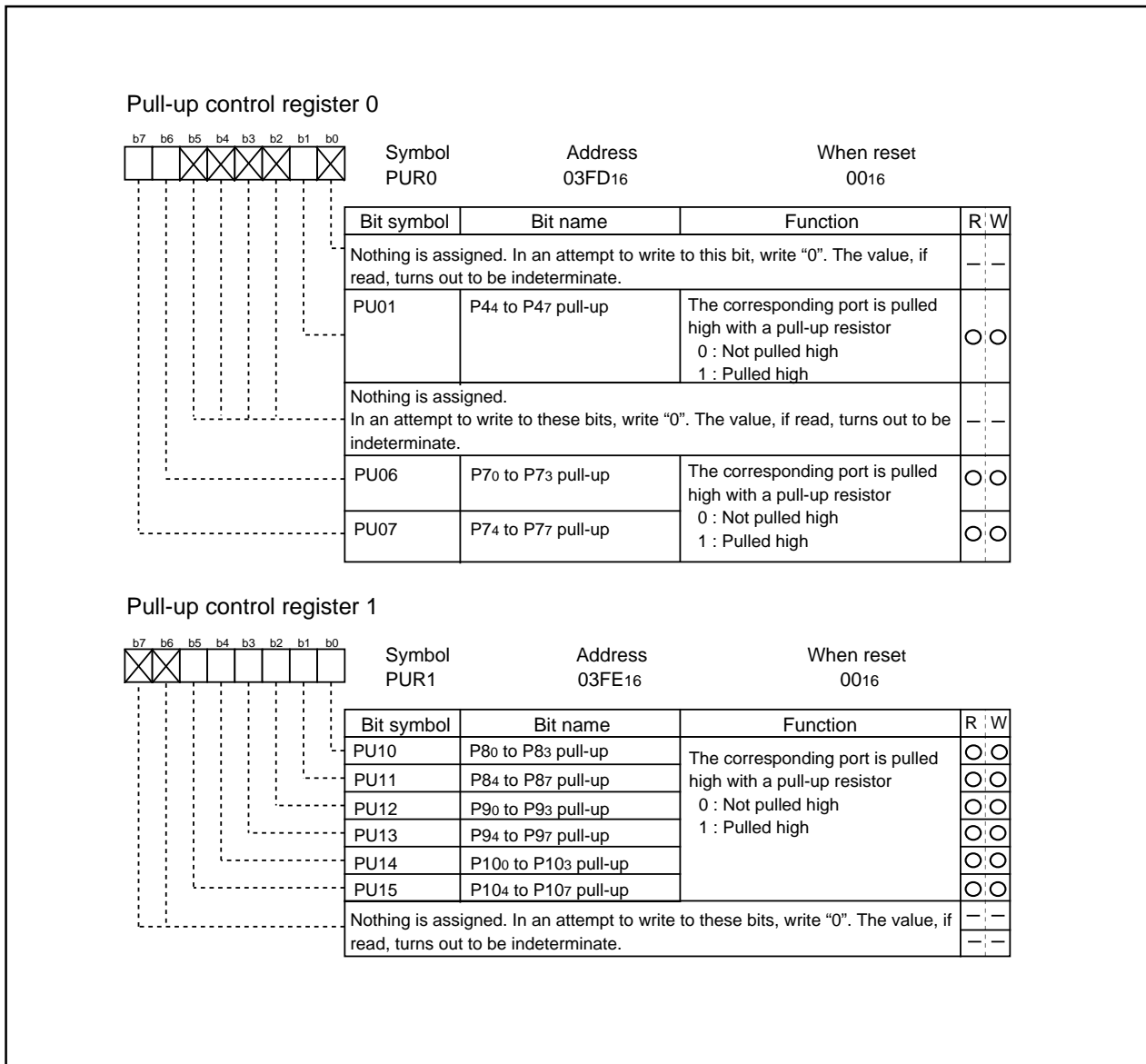
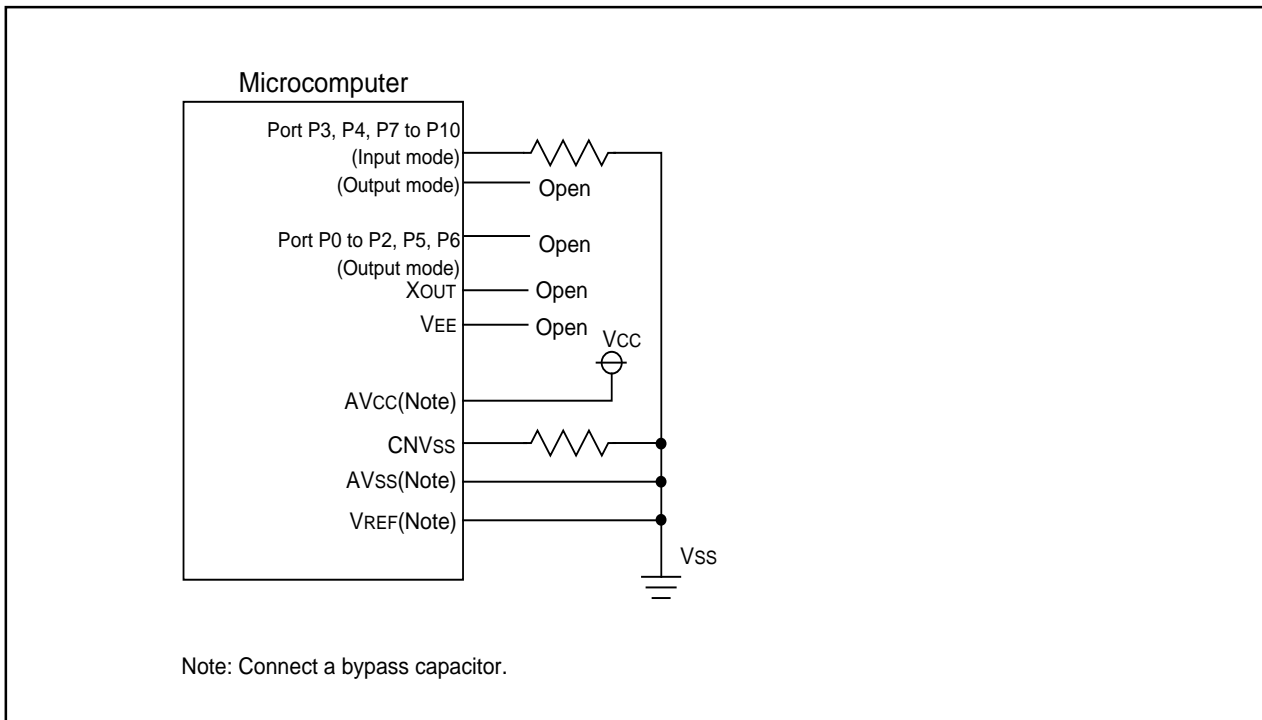


Figure UA-5. Pull-up control register

**Table UA-1. Example connection of unused pins**

Pin name	Connection
Ports P3, P4, P7 to P10	Specify output mode, and leave these pins open; or specify input mode, and connect to Vss via resistor (pull-down)
Ports P0 to P2, P5, P6	Leave these pins open
XOUT (Note 1), VEE	Open
AVcc	Connect to Vcc (Note 2)
AVss, VREF	Connect to Vss (Note 2)
CNVss	Connect to Vss via resistor

Note 1: With external clock input to XIN pin.  
 Note 2: Connect a bypass capacitor.



**Figure UA-6. Example connection of unused pins**

**Power Dissipation Calculating Method**

(Fixed number depending on microcomputer's standard)

- $V_{OH}$  output fall voltage of high-breakdown port  
2 V (max.); | Current value | = at 18 mA
- Resistor value = 68 k $\Omega$  (min.)
- Power dissipation of internal circuit (CPU, ROM, RAM etc.) = 5 V X 38 mA = 190 mW

(Fixed number depending on use condition)

- Apply voltage to VEE pin:  $V_{cc} - 50$  V
- Timing number a; digit number b; segment number c
- Ratio of Toff time corresponding Tdisp time: 1/16
- Turn ON segment number during repeat cycle: d
- All segment number during repeat cycle: e (= a X c)
- Total number of built-in resistor: for digit; f, for segment; g
- Digit pin current value h (mA)
- Segment pin current value i (mA)

(1) Digit pin power dissipation

$$\{h \times b \times (1 - T_{off} / T_{disp}) \times \text{voltage}\} / a$$

(2) Segment pin power dissipation

$$\{i \times d \times (1 - T_{off} / T_{disp}) \times \text{voltage}\} / a$$

(3) Pull-down resistor power dissipation (digit)

$$\{\text{power dissipation per 1 digit} \times (b \times f / b) \times (1 - T_{off} / T_{disp})\} / a$$

(4) Pull-down resistor power dissipation (segment)

$$\{\text{power dissipation per 1 segment} \times (d \times g / c) \times (1 - T_{off} / T_{disp})\} / a$$

(5) Internal circuit power dissipation (CPU, ROM, RAM etc.) = 190 mW

$$(1) + (2) + (3) + (4) + (5) = \underline{\underline{X \text{ mW}}}$$

**Power Dissipation Calculating example 1**

Fixed number depending on microcomputer's standard

- $V_{OH}$  output fall voltage of high-breakdown port  
2 V (max.); | Current value | = at 18 mA
- Resistor value 68 k $\Omega$  (min.)
- Power dissipation of internal circuit (CPU, ROM, RAM etc.) = 5 V X 38 mA = 190 mW

Fixed number depending on use condition

- Apply voltage to VEE pin:  $V_{cc} - 50$  V
- Timing number 17; digit number 16; segment number 20
- Ratio of Toff time corresponding Tdisp time: 1/16
- Turn ON segment number during repeat cycle: 31
- All segment number during repeat cycle: 340 (= 17 X 20)
- Total number of built-in resistor: for digit; 16, for segment; 20
- Digit pin current value: 18 (mA)
- Segment pin current value: 3 (mA)

(1) Digit pin power dissipation

$$\{18 \times 16 \times (1 - 1/16) \times 2\} / 17 = 31.77 \text{ mW}$$

(2) Segment pin power dissipation

$$\{3 \times 31 \times (1 - 1/16) \times 2\} / 17 = 10.26 \text{ mW}$$

(3) Pull-down resistor power dissipation (digit)

$$(50 - 2)^2 / 68 \times (16 \times 16/16) \times (1 - 1/16) / 17 = 29.90 \text{ mW}$$

(4) Pull-down resistor power dissipation (segment)

$$(50 - 2)^2 / 68 \times (31 \times 20/20) \times (1 - 1/16) / 17 = 57.93 \text{ mW}$$

(5) Internal circuit power dissipation (CPU, ROM, RAM etc.) = 190.00 mW

$$(1) + (2) + (3) + (4) + (5) = \underline{\underline{319.86 \text{ mW}}}$$

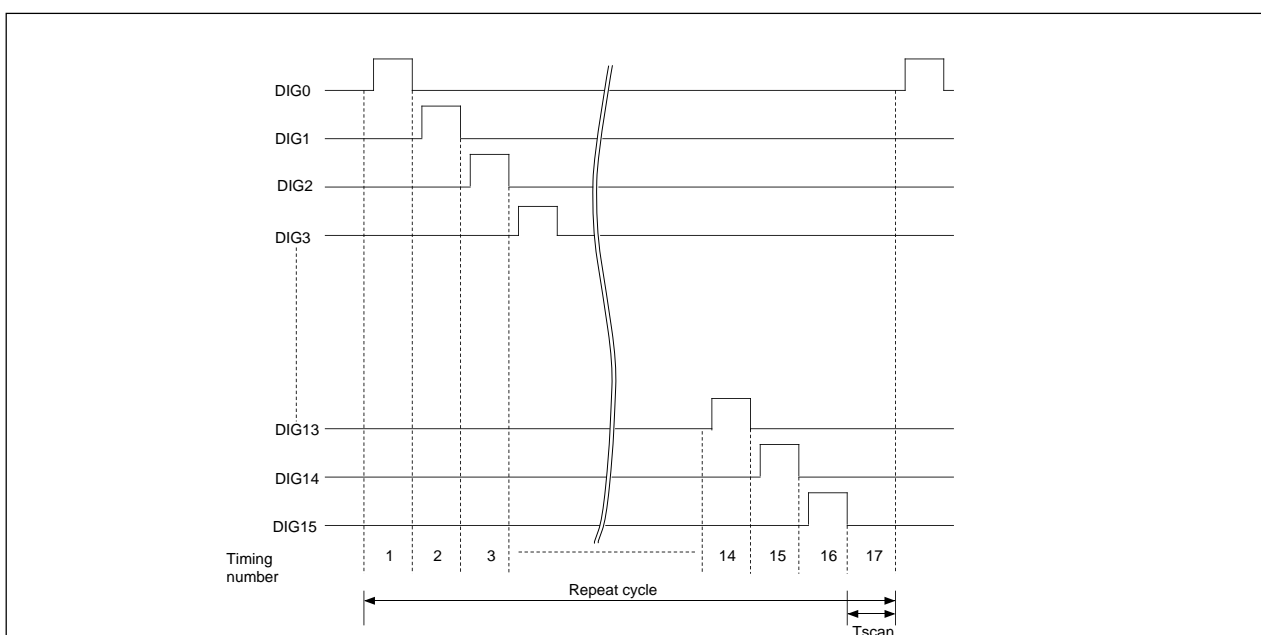


Figure S-1. Digit timing waveform (1)



**Power Dissipation Calculating example 2 (when 2 or more digit is turned ON at same time)**

Fixed number depending on microcomputer's standard

- $V_{OH}$  output fall voltage of high-breakdown port  
2 V (max.); | Current value | = at 18 mA
- Resistor value 68 k $\Omega$  (min.)
- Power dissipation of internal circuit (CPU, ROM, RAM etc.) = 5 V X 38 mA = 190 mW

Fixed number depending on use condition

- Apply voltage to VEE pin:  $V_{cc} - 50$  V
- Timing number 11; digit number 12; segment number 24
- Ratio of Toff time corresponding Tdisp time: 1/16
- Turn ON segment number during repeat cycle: 114
- All segment number during repeat cycle: 264 (= 11 X 24)
- Total number of built-in resistor: for digit; 10, for segment; 22
- Digit pin current value: 18 (mA)
- Segment pin current value: 3 (mA)

(1) Digit pin power dissipation

$$\{18 \times 12 \times (1 - 1/16) \times 2\} / 11 = 36.82 \text{ mW}$$

(2) Segment pin power dissipation

$$\{3 \times 114 \times (1 - 1/16) \times 2\} / 11 = 58.30 \text{ mW}$$

(3) Pull-down resistor power dissipation (digit)

$$(50 - 2)^2 / 68 \times (12 \times 10 / 12) \times (1 - 1/16) / 11 = 28.88 \text{ mW}$$

(4) Pull-down resistor power dissipation (segment)

$$(50 - 2)^2 / 68 \times (114 \times 22 / 24) \times (1 - 1/16) / 11 = 301.77 \text{ mW}$$

(5) Internal circuit power dissipation (CPU, ROM, RAM etc.) = 190.00 mW

$$(1) + (2) + (3) + (4) + (5) = \underline{\underline{615.77 \text{ mW}}} \text{ (There is a limit of use temperature)}$$

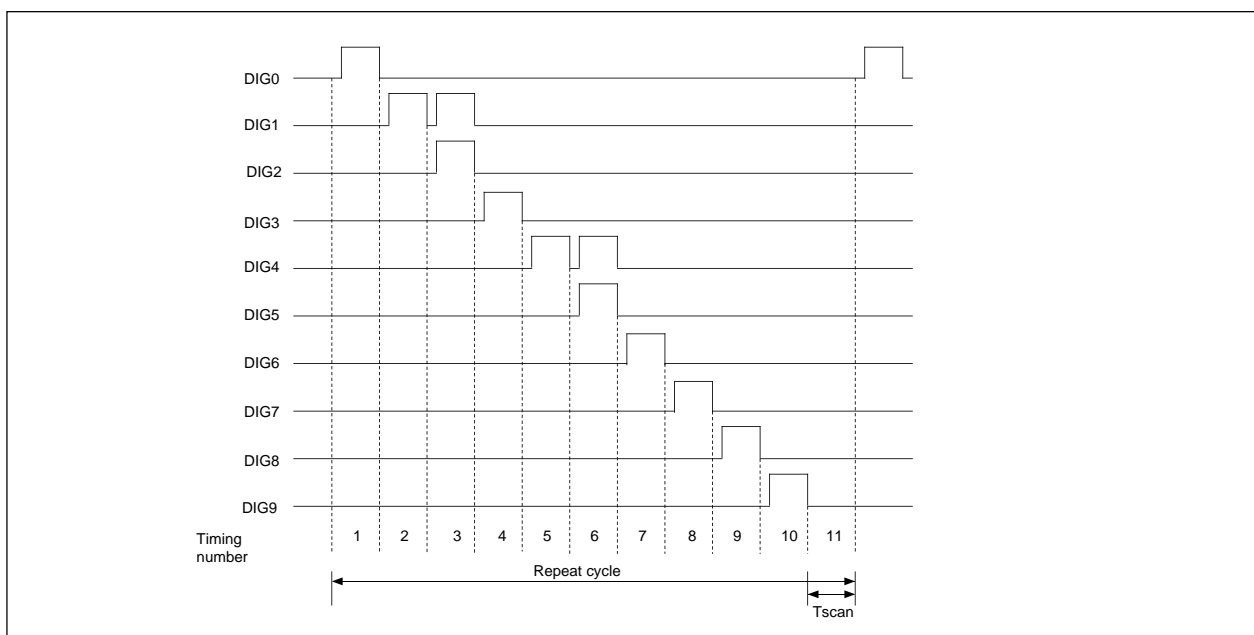


Figure S-2. Digit timing waveform (2)

**Power Dissipation Calculating example 3****(when 2 or more digit is turned ON at same time, and used Toff invalid function)**

Fixed number depending on microcomputer's standard

- V<sub>OH</sub> output fall voltage of high-breakdown port 2 V (max.); | Current value | = at 18 mA
- Resistor value 68 kΩ (min.)
- Power dissipation of internal circuit (CPU, ROM, RAM etc.) = 5 V X 38 mA = 190 mW

Fixed number depending on use condition

- Apply voltage to V<sub>EE</sub> pin: V<sub>cc</sub> – 50 V
- Timing number 11; digit number 12; segment number 24
- Ratio of Toff time corresponding T<sub>disp</sub> time: 1/16
- Turn ON segment number during repeat cycle: 114 ( for Toff invalid waveform;50)
- All segment number during repeat cycle: 264 (= 11 X 24)
- Total number of built-in resistor: for digit; 10, for segment; 22
- Digit pin current value: 18 (mA)
- Segment pin current value: 3 (mA)

(1) Digit pin power dissipation

$$[(18 \times 10 \times (1 - 1/16) \times 2) + \{18 \times 2 \times 2\}] / 11 = 37.23 \text{ mW}$$

(2) Segment pin power dissipation

$$[3 \times 64 \times (1 - 1/16) \times 2] + \{3 \times 50 \times 2\} / 11 = 60.00 \text{ mW}$$

(3) Pull-down resistor power dissipation (digit)

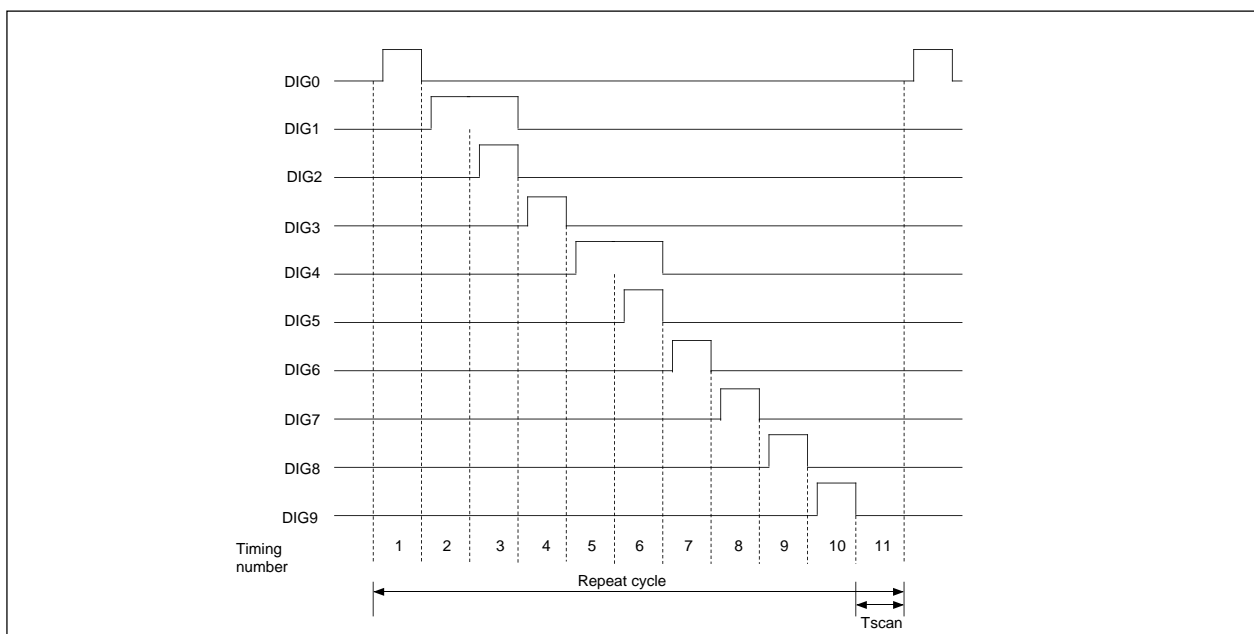
$$[(50 - 2)^2 / 68 \times (10 \times 10 / 12) \times (1 - 1 / 16)] + \{(50 - 2)^2 / 68 \times (2 \times 10 / 12)\} / 11 = 29.20 \text{ mW}$$

(4) Pull-down resistor power dissipation (segment)

$$[(50 - 2)^2 / 68 \times (64 \times 22 / 24) \times (1 - 1 / 16)] + \{(50 - 2)^2 / 68 \times (50 \times 22 / 24)\} / 11 = 310.59 \text{ mW}$$

(5) Internal circuit power dissipation (CPU, ROM, RAM etc.) = 190.00 mW

$$(1) + (2) + (3) + (4) + (5) = \underline{627.02 \text{ mW}} \text{ (There is a limit of use temperature)}$$

**Figure S-3. Digit timing waveform (3)**

**Table Z-1. Absolute maximum ratings**

Symbol	Parameter	Condition	Standard	Unit
V <sub>cc</sub>	Supply voltage		- 0.3 to 6.5	V
AV <sub>cc</sub>	Analog supply voltage		- 0.3 to 6.5	V
V <sub>EE</sub>	Pull-down supply voltage		V <sub>cc</sub> - 50 to V <sub>cc</sub> +0.3V	V
V <sub>I</sub>	Input voltage RESET, CNV <sub>ss</sub> , P44 to P47, P70 to P77, P80 to P87, P90 to P97, P100 to P107, V <sub>REF</sub> , X <sub>IN</sub>		- 0.3 to V <sub>cc</sub> +0.3 (Note)	V
V <sub>I</sub>	Input voltage P30 to P37, P40 to P43		V <sub>cc</sub> - 50 to V <sub>cc</sub> +0.3	V
V <sub>O</sub>	Output voltage P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P43, P50 to P57, P60 to P67		V <sub>cc</sub> - 50 to V <sub>cc</sub> +0.3	V
V <sub>O</sub>	Output voltage P44 to P47, P70 to P77, P80 to P87, P90 to P97, P100 to P107, X <sub>OUT</sub>		-0.3 to V <sub>cc</sub> +0.3	V
P <sub>d</sub>	Power dissipation	T <sub>a</sub> =-20 to 60 °C	750	mW
		T <sub>a</sub> =60 to 85 °C	750-12 X (T <sub>a</sub> -60)	mW
T <sub>opr</sub>	Operating ambient temperature		-20 to 85	°C
T <sub>stg</sub>	Storage temperature		-40 to 150	°C

Note 1: When writing to flash ,only CNV<sub>ss</sub> is -0.3 to 13 (V) .

**Table Z-2. Recommended operating conditions (referenced to V<sub>cc</sub> = 2.7V to 5.5V at T<sub>a</sub> = - 20 to 85°C unless otherwise specified) (Note)**

Symbol	Parameter	Standard			Unit
		Min	Typ.	Max.	
V <sub>cc</sub>	Supply voltage	2.7(Note1)	5.0	5.5	V
AV <sub>cc</sub>	Analog supply voltage		V <sub>cc</sub>		V
V <sub>ss</sub>	Supply voltage		0		V
AV <sub>ss</sub>	Analog supply voltage		0		V
V <sub>EE</sub>	Pull-down supply voltage	V <sub>cc</sub> -48		V <sub>cc</sub>	V
V <sub>IH</sub>	HIGH input voltage P70 to P77, P80 to P87, P90 to P97, P100 to P107, X <sub>IN</sub> , RESET, CNV <sub>ss</sub>	0.8V <sub>cc</sub>		V <sub>cc</sub>	V
V <sub>IH</sub>	HIGH input voltage P44 to P47	0.50V <sub>cc</sub>		V <sub>cc</sub>	V
V <sub>IH</sub>	HIGH input voltage P30 to P37, P40 to P43	0.52V <sub>cc</sub>		V <sub>cc</sub>	V
V <sub>IL</sub>	LOW input voltage P70 to P77, P80 to P87, P90 to P97, P100 to P107, X <sub>IN</sub> , RESET, CNV <sub>ss</sub>	0		0.2V <sub>cc</sub>	V
V <sub>IL</sub>	LOW input voltage P30 to P37, P40 to P43	0		0.16V <sub>cc</sub>	V
V <sub>IL</sub>	LOW input voltage P44 to P47	0		0.16V <sub>cc</sub>	V

Note: V<sub>cc</sub> = 4.0V to 5.5V in flash memory version.

**Table Z-3. Recommended operating conditions (referenced to V<sub>CC</sub> = 2.7V to 5.5V at Ta = - 20 to 85°C unless otherwise specified) (Note 6)**

Symbol	Parameter	Standard			Unit
		Min	Typ.	Max	
I <sub>OH</sub> (peak)	HIGH total peak output current (Note 1) P0 <sub>0</sub> to P0 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub>			-240	mA
I <sub>OH</sub> (peak)	HIGH total peak output current (Note 1) P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>3</sub>			-240	mA
I <sub>OH</sub> (peak)	HIGH total peak output current (Note 1) P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>5</sub>			-80	mA
I <sub>OH</sub> (peak)	HIGH total peak output current (Note 1) P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub>			-80	mA
I <sub>OL</sub> (peak)	LOW total peak output current (Note 1) P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>5</sub>			80	mA
I <sub>OL</sub> (peak)	LOW total peak output current (Note 1) P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub>			80	mA
I <sub>OH</sub> (avg)	HIGH total average output current (Note 1) P0 <sub>0</sub> to P0 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub>			-120	mA
I <sub>OH</sub> (avg)	HIGH total average output current (Note 1) P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>3</sub>			-120	mA
I <sub>OH</sub> (avg)	HIGH total average output current (Note 1) P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>5</sub>			-40	mA
I <sub>OH</sub> (avg)	HIGH total average output current (Note 1) P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub>			-40	mA
I <sub>OL</sub> (avg)	LOW total average output current (Note 1) P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>5</sub>			40	mA
I <sub>OL</sub> (avg)	LOW total average output current (Note 1) P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub>			40	mA
I <sub>OH</sub> (peak)	HIGH peak output current (Note 2) P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>3</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub>			-40	mA
I <sub>OH</sub> (peak)	HIGH peak output current (Note 2) P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub>			-10	mA
I <sub>OL</sub> (peak)	LOW peak output current (Note 2) P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub>			10	mA
I <sub>OH</sub> (avg)	HIGH average output current (Note 3) P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>3</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub>			-18	mA
I <sub>OH</sub> (avg)	HIGH average output current (Note 3) P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub>			-5	mA
I <sub>OL</sub> (avg)	LOW average output current (Note 3) P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub>			5	mA
f (X <sub>IN</sub> )	Main clock input oscillation frequency (Note 4, 7)	V <sub>CC</sub> =4.0V to 5.5V	0	10	MHz
		V <sub>CC</sub> =2.7V to 4.0V	0	5 X V <sub>CC</sub> -10	MHz
f (X <sub>CIN</sub> )	Sub clock oscillation frequency (Note 4, 5)		32.768	50	kHz

Note 1: The total output current is the sum of all the currents through the applicable ports. The total average value measured over 100ms. The total peak current is the peak of all the currents.

Note 2: The peak output current is the peak current flowing in each port.

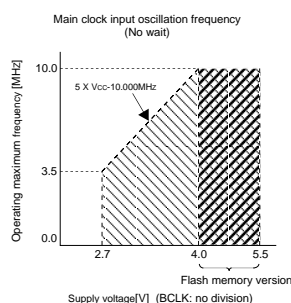
Note 3: The average output current in an average value measured over 100ms.

Note 4: When the oscillating frequency has a duty cycle of 50 %.

Note 5: When using the microcomputer in low-speed mode, set the sub-clock input oscillation frequency on condition that f(X<sub>CIN</sub>) < f(X<sub>IN</sub>) / 3.

Note 6: V<sub>CC</sub>=4.0V to 5.5V in flash memory version.

Note 7: Relationship between main clock oscillation frequency and supply voltage.



V<sub>CC</sub>=5V

**Table Z-4. Electrical characteristics (referenced to V<sub>CC</sub> = 5V, V<sub>SS</sub> = 0V at Ta = 25°C,  
f(X<sub>IN</sub>) = 10MHz unless otherwise specified)**

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
V <sub>OH</sub>	HIGH output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>3</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub>	I <sub>OH</sub> = - 18mA	3.5			V
			I <sub>OH</sub> = - 5mA	4.5			
V <sub>OH</sub>	HIGH output voltage	P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub>	I <sub>OH</sub> = - 5mA	3.0			V
V <sub>OH</sub>	HIGH output voltage	X <sub>OUT</sub>	HIGH POWER	I <sub>OH</sub> = - 1mA	3.0		V
			LOW POWER	I <sub>OH</sub> = - 0.5mA	3.0		
V <sub>OL</sub>	LOW output voltage	P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub>	I <sub>OL</sub> =5mA			2.0	V
V <sub>OL</sub>	LOW output voltage	X <sub>OUT</sub>	HIGH POWER	I <sub>OL</sub> =1mA		2.0	V
			LOW POWER	I <sub>OL</sub> =0.5mA		2.0	
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	TA0 <sub>IN</sub> to TA4 <sub>IN</sub> , TB0 <sub>IN</sub> to TB2 <sub>IN</sub> , INT <sub>0</sub> to INT <sub>5</sub> , CTS <sub>0</sub> , CTS <sub>1</sub> , CLK <sub>0</sub> , CLK <sub>1</sub> , SRDY2 <sub>IN</sub> , SBSY2 <sub>IN</sub> , SIN <sub>2</sub> , SCLK2 <sub>1</sub> , SCLK2 <sub>2</sub> , RxD0, RxD1		0.2		0.8	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	RESET		0.2		1.8	V
I <sub>IH</sub>	HIGH input current	P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub>	V <sub>I</sub> =5V			5.0	μA
I <sub>IH</sub>		P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>3</sub> (Note 1)	V <sub>I</sub> =5V			5.0	μA
I <sub>IL</sub>	LOW input current	P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub>	V <sub>I</sub> =0V			- 5.0	μA
I <sub>IL</sub>		P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>3</sub> (Note1)	V <sub>I</sub> =0V			- 5.0	μA
R <sub>PULLUP</sub>	Pull-up resistance	P4 <sub>4</sub> to P4 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub>	V <sub>I</sub> =0V	30.0	50.0	167.0	kΩ
R <sub>PULLD</sub>	Pull-down resistance	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub>	V <sub>EE</sub> =V <sub>CC</sub> - 48V, V <sub>OL</sub> =V <sub>CC</sub> Output transistors "off"	68	80	120	kΩ
I <sub>LEAK</sub>	Output leak current	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>4</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub>	V <sub>EE</sub> =V <sub>CC</sub> - 48V, V <sub>OL</sub> =V <sub>CC</sub> - 48V Output transistors "off"			- 10	μA
R <sub>I<sub>XIN</sub></sub>	Feedback resistance X <sub>IN</sub>				1.0		MΩ
R <sub>I<sub>XCIN</sub></sub>	Feedback resistance X <sub>CIN</sub>				6.0		MΩ
V <sub>RAM</sub>	RAM retention voltage		When clock is stopped	2.0			V
I <sub>CC</sub>	Power supply current (Note 3)	The output pins are open and other pins are V <sub>SS</sub>	f(X <sub>IN</sub> )=10MHz Square wave, no division		19.0	38.0	mA
			f(X <sub>IN</sub> )=10MHz Square wave, 8 division		4.2		
			f(X <sub>CIN</sub> )=32kHz Square wave (Note2)		90.0		μA
			f(X <sub>CIN</sub> )=32kHz When a WAIT instruction is executed (Note2)		4.0		μA
			Ta=25 °C when clock is stopped			1.0	μA
			Ta=85 °C when clock is stopped			20.0	

Note 1: Except when reading ports P3, P40 to P43.

Note 2: Fixed X<sub>CIN</sub>-X<sub>COUT</sub> drive capacity select bit to "HIGH" and X<sub>IN</sub> pin to "H" level.

Note 3: This contains an electric current to flow into AV<sub>CC</sub> pin.

$V_{CC}=5V$ **Table Z-5. A-D conversion characteristics (referenced to  $V_{CC} = AV_{CC} = V_{REF} = 5V$ ,  $V_{SS} = AV_{SS} = 0V$   
at  $T_a = 25^\circ C$ ,  $f(X_{IN}) = 10MHz$  unless otherwise specified)**

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
-	Resolution		$V_{REF} = V_{CC}$			10	Bits
-	Absolute accuracy	Sample & hold function not available	$V_{REF} = V_{CC} = 5V$			$\pm 3$	LSB
		Sample & hold function available(10bit)	$V_{REF} = V_{CC} = 5V$ AN0 to AN7 input			$\pm 3$	LSB
		Sample & hold function available(8bit)	$V_{REF} = V_{CC} = 5V$			$\pm 2$	LSB
$R_{LADDER}$	Ladder resistance		$V_{REF} = V_{CC}$	10		40	$k\Omega$
$t_{CONV}$	Conversion time (10bit)			3.3			$\mu s$
$t_{CONV}$	Conversion time (8bit)			2.8			$\mu s$
$t_{SAMP}$	Sampling time			0.3			$\mu s$
$V_{REF}$	Reference voltage			2		$V_{CC}$	V
$V_{IA}$	Analog input voltage			0		$V_{REF}$	V

**Table Z-6. D-A conversion characteristics (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = AV_{SS} = 0V$ ,  $V_{REF} = 5V$   
at  $T_a = 25^\circ C$ ,  $f(X_{IN}) = 10MHz$  unless otherwise specified)**

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
-	Resolution					8	Bits
-	Absolute accuracy					1.0	%
$t_{su}$	Setup time					3	$\mu s$
$R_o$	Output resistance			4	10	20	$k\Omega$
$I_{VREF}$	Reference power supply input current		(Note)			1.5	mA

Note: This applies when using one D-A converter, with the D-A register for the unused D-A converter set to "0016".

The A-D converter's ladder resistance is not included.

Also, when the  $V_{ref}$  is unconnected at the A-D control register,  $I_{VREF}$  is sent.

$V_{CC}=5V$

Timing requirements (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_a = 25^\circ C$  unless otherwise specified)

Table Z-7. External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_c$	External clock input cycle time	100		ns
$t_{w(H)}$	External clock input HIGH pulse width	40		ns
$t_{w(L)}$	External clock input LOW pulse width	40		ns
$t_r$	External clock rise time		15	ns
$t_f$	External clock fall time		15	ns

Switching characteristics (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_a = 25^\circ C$  unless otherwise specified)

Table Z-8. High-breakdown voltage p-channel open-drain output port

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
$t_{r(Pch-strg)}$	P-channel high-breakdown voltage output rising time (Note 1)	$C_L=100pF$ $V_{EE}=V_{CC} - 43V$		55		ns
$t_{r(Pch-weak)}$	P-channel high-breakdown voltage output rising time (Note 2)	$C_L=100pF$ $V_{EE}=V_{CC} - 43V$		1.8		$\mu s$

Note 1: When bit 7 of the FLDC mode register (address 0350<sub>16</sub>) is at "0".  
Note 2: When bit 7 of the FLDC mode register (address 0350<sub>16</sub>) is at "1".

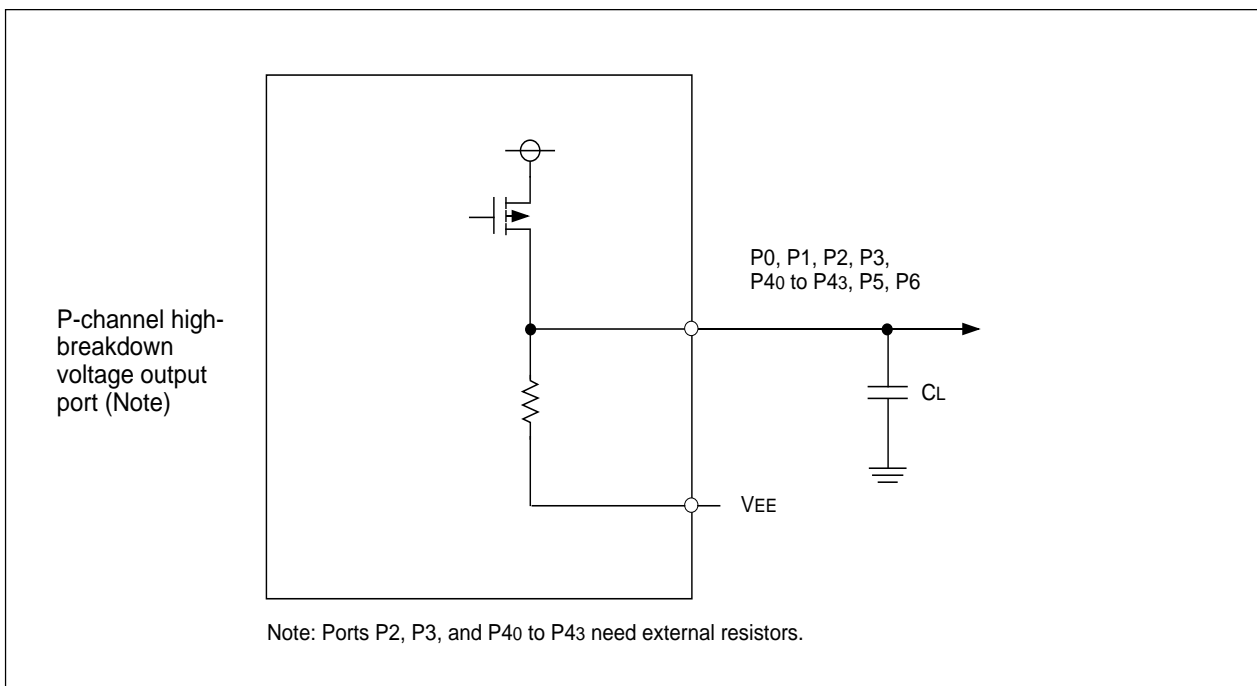


Figure Z-2. Circuit for measuring output switching characteristics

Timing ( $V_{CC}=5V$ ) $V_{CC}=5V$ Timing requirements (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_a = 25^\circ C$  unless otherwise specified)

Table Z-9. Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIn input cycle time	100		ns
$t_w(TAH)$	TAiIn input HIGH pulse width	40		ns
$t_w(TAL)$	TAiIn input LOW pulse width	40		ns

Table Z-10. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIn input cycle time	400		ns
$t_w(TAH)$	TAiIn input HIGH pulse width	200		ns
$t_w(TAL)$	TAiIn input LOW pulse width	200		ns

Table Z-11. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIn input cycle time	200		ns
$t_w(TAH)$	TAiIn input HIGH pulse width	100		ns
$t_w(TAL)$	TAiIn input LOW pulse width	100		ns

Table Z-12. Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_w(TAH)$	TAiIn input HIGH pulse width	100		ns
$t_w(TAL)$	TAiIn input LOW pulse width	100		ns

Table Z-13. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	2000		ns
$t_w(UPH)$	TAiOUT input HIGH pulse width	1000		ns
$t_w(UPL)$	TAiOUT input LOW pulse width	1000		ns
$t_{su}(UP-TiN)$	TAiOUT input setup time	400		ns
$t_h(TiN-UP)$	TAiOUT input hold time	400		ns



$V_{CC}=5V$ Timing requirements (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_a = 25^\circ C$  unless otherwise specified)

Table Z-14. Timer B input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIn input cycle time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBiIn input HIGH pulse width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIn input LOW pulse width (counted on one edge)	40		ns
$t_{c(TB)}$	TBiIn input cycle time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBiIn input HIGH pulse width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIn input LOW pulse width (counted on both edges)	80		ns

Table Z-15. Timer B input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIn input cycle time	400		ns
$t_{w(TBH)}$	TBiIn input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIn input LOW pulse width	200		ns

Table Z-16. Timer B input (pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIn input cycle time	400		ns
$t_{w(TBH)}$	TBiIn input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIn input LOW pulse width	200		ns

Table Z-17. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	30		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

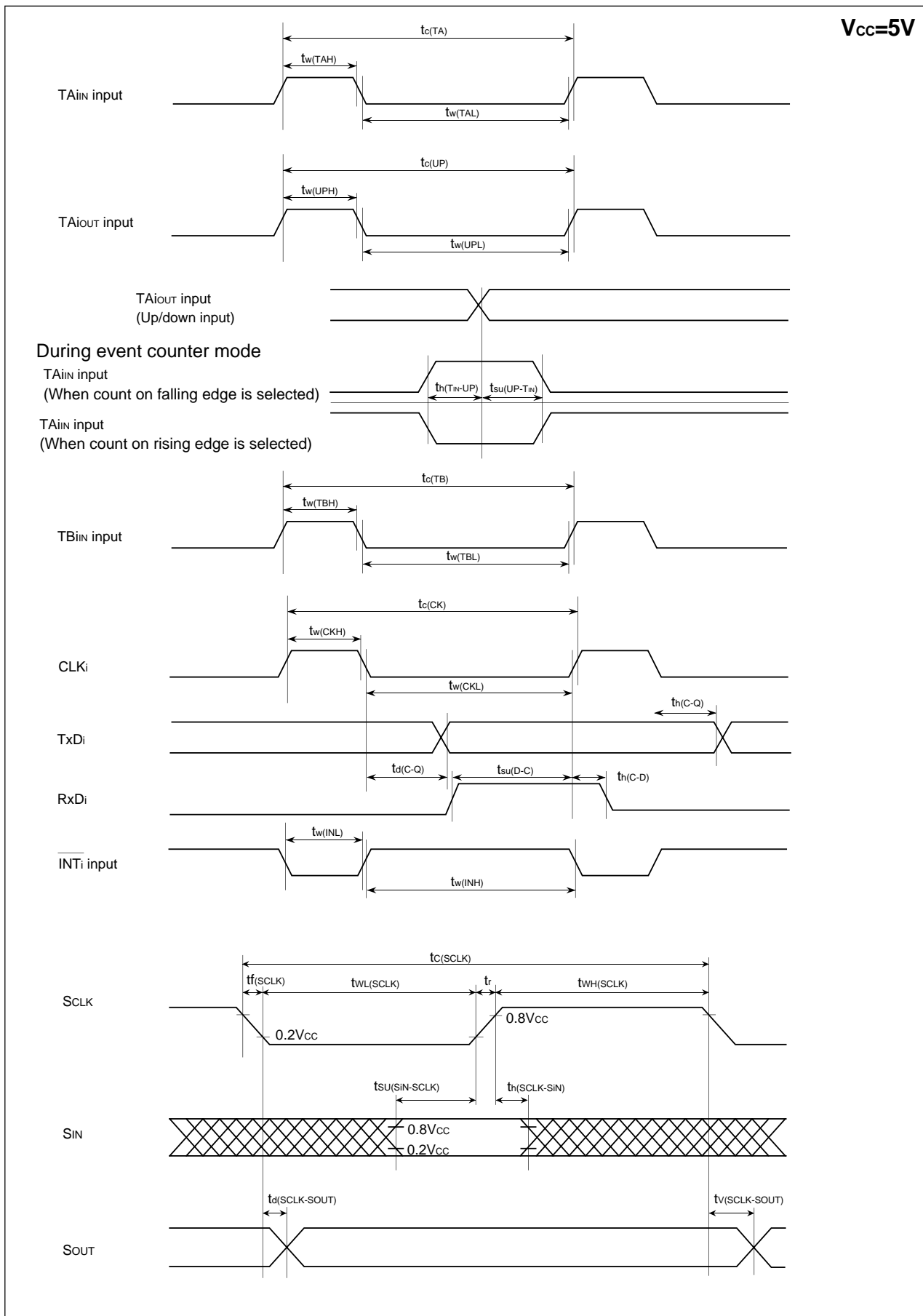
Table Z-18. External interrupt  $\overline{INT}_i$  inputs

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INT}_i$ input HIGH pulse width	250		ns
$t_{w(INL)}$	$\overline{INT}_i$ input LOW pulse width	250		ns

Table Z-19. Automatic transfer serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(SCLK)}$	Serial I/O clock input cycle time	0.95		$\mu s$
$t_{wH(SCLK)}$	Serial I/O clock input HIGH pulse width	400		ns
$t_{wL(SCLK)}$	Serial I/O clock input LOW pulse width	400		ns
$t_{su(SCLK-SIN)}$	Serial I/O input setup time	200		ns
$t_{h(SCLK-SIN)}$	Serial I/O input hold time	200		ns

Timing ( $V_{CC}=5V$ )



V<sub>CC</sub>=3V

**Table Z-20. Electrical characteristics (referenced to V<sub>CC</sub> = 3V, V<sub>SS</sub> = 0V at Ta = 25°C,  
f(X<sub>IN</sub>) =5MHz unless otherwise specified)**

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
V <sub>OH</sub>	HIGH output voltage	P00 to P07,P10 to P17,P20 to P27, P30 to P37,P40 to P43,P50 to P57, P60 to P67	I <sub>OH</sub> = - 18mA	1.5			V
			I <sub>OH</sub> = - 5mA	2.5			
V <sub>OH</sub>	HIGH output voltage	P44 to P47,P70 to P77,P80 to P87, P90 to P97,P100 to P107	I <sub>OH</sub> = - 1mA	2.5			V
V <sub>OH</sub>	HIGH output voltage	X <sub>OUT</sub>	HIGH POWER	I <sub>OH</sub> = - 0.1mA	2.5		V
			LOW POWER	I <sub>OH</sub> = - 50μA	2.5		
V <sub>OL</sub>	LOW output voltage	P44 to P47,P70 to P77,P80 to P87, P90 to P97,P100 to P107	I <sub>OL</sub> =1mA			0.5	V
V <sub>OL</sub>	LOW output voltage	X <sub>OUT</sub>	HIGH POWER	I <sub>OL</sub> =0.1mA		0.5	V
			LOW POWER	I <sub>OL</sub> =50μA		0.5	
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	TA0IN to TA4IN,TB0IN to TB2IN, INT0 to INT5, CTS0, CTS1, CLK0,CLK1,SRDY2IN,SBSY2IN, SIN2,SCLK21,SCLK22 RTS0,RTS1		0.2		0.8	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	RESET		0.2		1.8	V
I <sub>IH</sub>	HIGH input current	P44 to P47,P70 to P77,P80 to P87, P90 to P97,P100 to P107, X <sub>IN</sub> , RESET, CNV <sub>SS</sub>	V <sub>i</sub> =3V			4.0	μA
I <sub>IH</sub>		P30 to P37,P40 to P43(Note 1)	V <sub>i</sub> =3V			4.0	μA
I <sub>IL</sub>	LOW input current	P44 to P47,P70 to P77,P80 to P87, P90 to P97,P100 to P107, X <sub>IN</sub> , RESET, CNV <sub>SS</sub>	V <sub>i</sub> =0V			- 4.0	μA
I <sub>IL</sub>		P30 to P37,P40 to P43(Note 1)	V <sub>i</sub> =0V			- 4.0	μA
R <sub>PULLUP</sub>	Pull-up resistance	P44 to P47,P70 to P77, P80 to P87,P90 to P97, P100 to P107	V <sub>i</sub> =0V	66.0	120.0	500.0	kΩ
R <sub>PULLD</sub>	Pull-down resistance	P00 to P07,P10 to P17, P50 to P57,P60 to P67	V <sub>EE</sub> =V <sub>CC</sub> - 48V,V <sub>OL</sub> =V <sub>CC</sub> Output transistors "off"	68	80	120	kΩ
I <sub>LEAK</sub>	Output leak current	P00 to P07,P10 to P17, P20 to P27,P30 to P37, P40 to P44,P50 to P57, P60 to P67	V <sub>EE</sub> =V <sub>CC</sub> - 48V,V <sub>OL</sub> =V <sub>CC</sub> - 48V Output transistors "off"			- 10	μA
R <sub>XIN</sub>	Feedback resistance X <sub>IN</sub>				3.0		MΩ
R <sub>XIN</sub>	Feedback resistance X <sub>CIN</sub>				10.0		MΩ
V <sub>RAM</sub>	RAM retention voltage		When clock is stopped	2.0			V
I <sub>CC</sub>	Power supply current (Note 3)	The output pins are open and other pins are V <sub>SS</sub>	f(X <sub>IN</sub> )=5MHz Square wave, no division		6.0	15.0	mA
			f(X <sub>IN</sub> )=5MHz Square wave, 8 division		1.6		
			f(X <sub>CIN</sub> )=32kHz Square wave		50.0		μA
			f(X <sub>CIN</sub> )=32kHz When a WAIT instruction is executed. Oscillation capacity High (Note2)		2.8		
			f(X <sub>CIN</sub> )=32kHz When a WAIT instruction is executed. Oscillation capacity Low (Note2)		0.9		μA
			Ta=25 °C when clock is stopped			1.0	
			Ta=85 °C when clock is stopped			20.0	

Note 1: Except when reading ports P3, P40 to P43.

Note 2: With one timer operated using fc32.

Note 3: This contains an electric current to flow into AV<sub>CC</sub> pin.

**$V_{CC}=3V$** 

**Table Z-21. A-D conversion characteristics (referenced to  $V_{CC} = AV_{CC} = V_{REF} = 3V$ ,  $V_{SS} = AV_{SS} = 0V$   
at  $T_a = 25^\circ C$ ,  $f(X_{IN}) = 5MHz$  unless otherwise specified)**

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max	
-	Resolution	$V_{REF} = V_{CC}$			10	Bits
-	Absolute accuracy	Sample & hold function not available (8 bit) $V_{REF} = V_{CC} = 3V$ , $\phi_{AD} = f(X_{IN})/2$			$\pm 2$	LSB
$R_{LADDER}$	Ladder resistance	$V_{REF} = V_{CC}$	10		40	$k\Omega$
$t_{CONV}$	Conversion time (8bit)		14.0			$\mu s$
$V_{REF}$	Reference voltage		2.7		$V_{CC}$	V
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V

**Table Z-22. D-A conversion characteristics (referenced to  $V_{CC} = 3V$ ,  $V_{SS} = AV_{SS} = 0V$ ,  $V_{REF} = 3V$   
at  $T_a = 25^\circ C$ ,  $f(X_{IN}) = 5MHz$  unless otherwise specified)**

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max	
-	Resolution				8	Bits
-	Absolute accuracy				1.0	%
$t_{SU}$	Setup time				3	$\mu s$
$R_o$	Output resistance		4	10	20	$k\Omega$
$I_{VREF}$	Reference power supply input current	(Note)			1.0	mA

Note: This applies when using one D-A converter, with the D-A register for the unused D-A converter set to "0016".

The A-D converter's ladder resistance is not included.

Also, when the  $V_{ref}$  is unconnected at the A-D control register,  $I_{VREF}$  is sent.

**$V_{CC}=3V$** Timing requirements (referenced to  $V_{CC} = 3V$ ,  $V_{SS} = 0V$  at  $T_a = 25^\circ C$  unless otherwise specified)

Table Z-23. External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_c$	External clock input cycle time	200		ns
$t_{w(H)}$	External clock input HIGH pulse width	85		ns
$t_{w(L)}$	External clock input LOW pulse width	85		ns
$t_r$	External clock rise time		18	ns
$t_f$	External clock fall time		18	ns

Timing ( $V_{CC}=3V$ , only mask ROM version) **$V_{CC}=3V$** Timing requirements (referenced to  $V_{CC} = 3V$ ,  $V_{SS} = 0V$  at  $T_a = 25^\circ C$  unless otherwise specified)**Table Z-24. Timer A input (counter input in event counter mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	150		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	60		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	60		ns

**Table Z-25. Timer A input (gating input in timer mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	600		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	300		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	300		ns

**Table Z-26. Timer A input (external trigger input in one-shot timer mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	300		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

**Table Z-27. Timer A input (external trigger input in pulse width modulation mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

**Table Z-28. Timer A input (up/down input in event counter mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	3000		ns
$t_{w(UPH)}$	TAiOUT input HIGH pulse width	1500		ns
$t_{w(UPL)}$	TAiOUT input LOW pulse width	1500		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	600		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	600		ns

$V_{CC}=3V$ Timing requirements (referenced to  $V_{CC} = 3V$ ,  $V_{SS} = 0V$  at  $T_a = 25^\circ C$  unless otherwise specified)**Table Z-29. Timer B input (counter input in event counter mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (counted on one edge)	150		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on one edge)	60		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on one edge)	60		ns
$t_{c(TB)}$	TBiIN input cycle time (counted on both edges)	300		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on both edges)	160		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	160		ns

**Table Z-30. Timer B input (pulse period measurement mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	600		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	300		ns

**Table Z-31. Timer B input (pulse width measurement mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	600		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	300		ns

**Table Z-32. Serial I/O**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	150		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	150		ns
$t_d(C-Q)$	TxDi output delay time		160	ns
$t_h(C-Q)$	TxDi hold time	0		ns
$t_{su}(D-C)$	RxDi input setup time	50		ns
$t_h(C-D)$	RxDi input hold time	90		ns

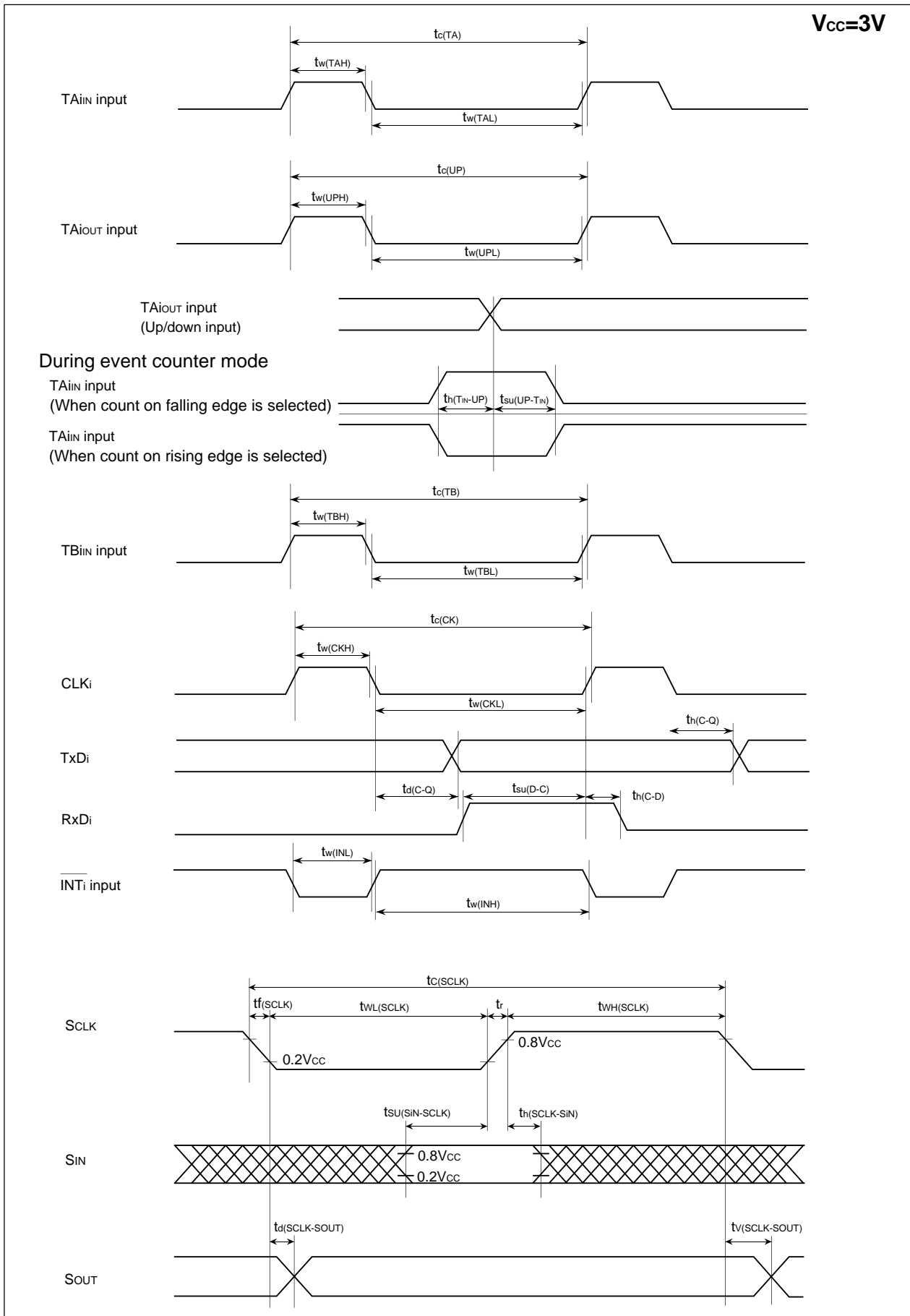
**Table Z-33. External interrupt  $\overline{INT}_i$  inputs**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INT}_i$ input HIGH pulse width	380		ns
$t_{w(INL)}$	$\overline{INT}_i$ input LOW pulse width	380		ns

**Table Z-34. Automatic transfer serial I/O**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(SCLK)}$	Serial I/O clock input cycle time	2.0		$\mu s$
$t_{wH(SCLK)}$	Serial I/O clock input HIGH pulse width	1000		ns
$t_{wL(SCLK)}$	Serial I/O clock input LOW pulse width	1000		ns
$t_{su}(SCLK-SIN)$	Serial I/O input setup time	400		ns
$t_h(SCLK-SIN)$	Serial I/O input hold time	400		ns

Timing ( $V_{CC}=3V$ , only mask ROM version)





## Description

**Outline Performance**

Table AA-1 shows the outline performance of the M30218 group (flash memory version).

**Table AA-1. Outline Performance of the M30218 group (flash memory version)**

Item		Performance
Power supply voltage		4.0V to 5.5 V (f(XIN)=10MHz)
Program/erase voltage		VPP=12V $\pm$ 5% (f(XIN)=10MHz)
		VCC=5V $\pm$ 10% (f(XIN)=10MHz)
Flash memory operation mode		Three modes (parallel I/O, standard serial I/O, CPU rewrite)
Erase block division	User ROM area	See Figure 1.AA.3.
	Boot ROM area	One division (3.5 K bytes) (Note)
Program method		In units of byte
Erase method		Collective erase / block erase
Program/erase control method		Program/erase control by software command
Number of commands		6 commands
Program/erase count		100 times
ROM code protect		Standard serial I/O mode is supported.

Note: The boot ROM area contains a standard serial I/O mode control program which is stored in it when shipped from the factory. This area can be erased and programmed in only parallel I/O mode.

Description

Flash Memory

The M30218 group (flash memory version) contains the NOR type of flash memory that requires a high-voltage VPP power supply for program/erase operations, in addition to the VCC power supply for device operation. For this flash memory, three flash memory modes are available in which to read, program, and erase: parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and a CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU). Each mode is detailed in the pages to follow.

In addition to the ordinary user ROM area to store a microcomputer operation control program, the flash memory has a boot ROM area that is used to store a program to control rewriting in CPU rewrite and standard serial I/O modes. This boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. However, the user can write a rewrite control program in this area that suits the user's application system. This boot ROM area can be rewritten in only parallel I/O mode.

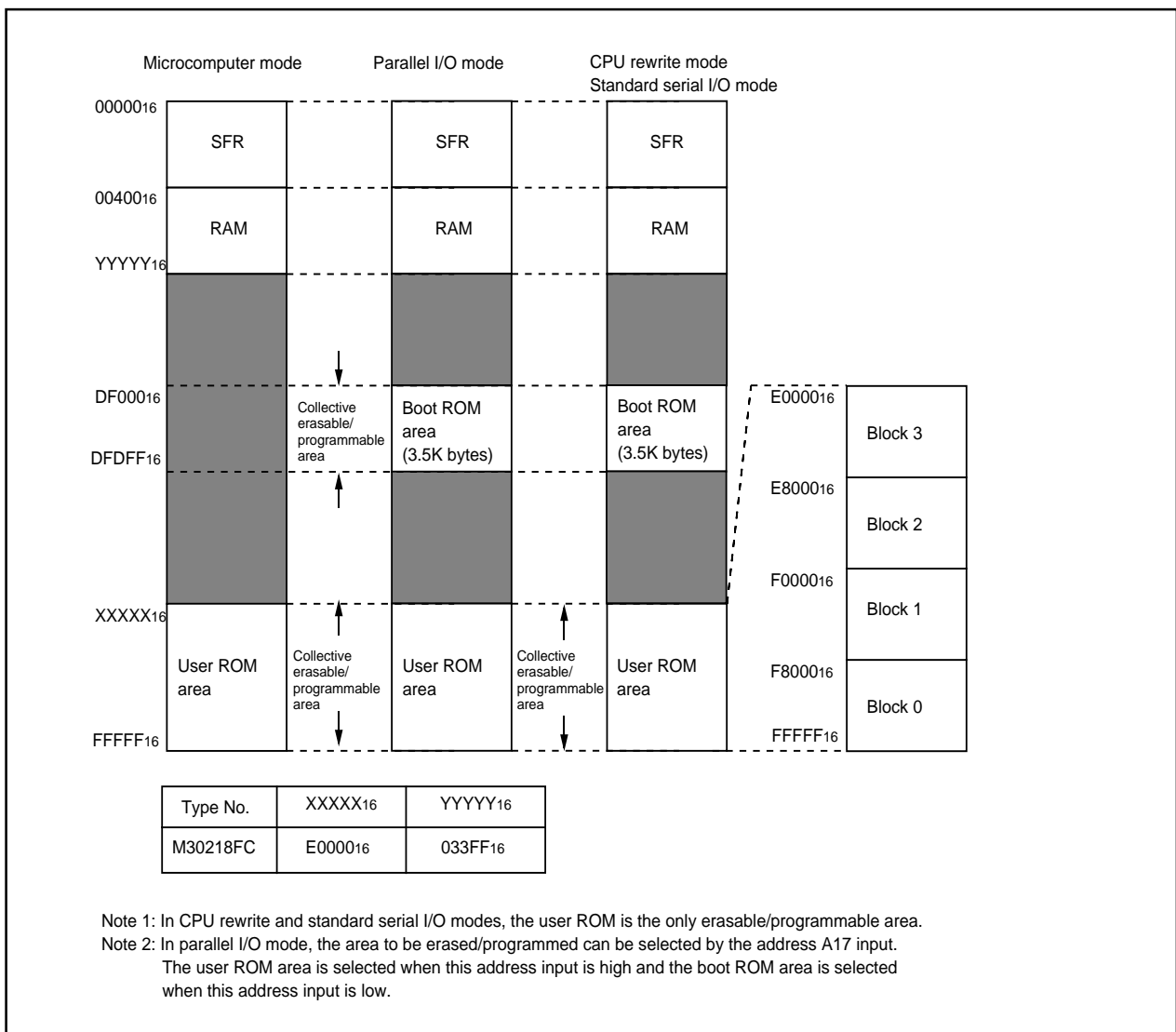


Figure AA-3. Block diagram of flash memory version

### CPU Rewrite Mode

In CPU rewrite mode, the on-chip flash memory can be operated on (read, program, or erase) under control of the Central Processing Unit (CPU). In CPU rewrite mode, the flash memory can be operated on by reading or writing to the flash memory control register and flash command register. Figure BB-1, Figure BB-2 show the flash memory control register, and flash command register respectively.

Also, in CPU rewrite mode, the CNVss pin is used as the VPP power supply pin. Apply the power supply voltage, VPPH, from an external source to this pin.

In CPU rewrite mode, only the user ROM area shown in Figure AA-3 can be rewritten; the boot ROM area cannot be rewritten. Make sure the program and block commands are issued for only the user ROM area. The control program for CPU rewrite mode can be stored in either user ROM or boot ROM area. In the CPU rewrite mode, because the flash memory cannot be read from the CPU, the rewrite control program must be transferred to internal RAM before it can be executed.

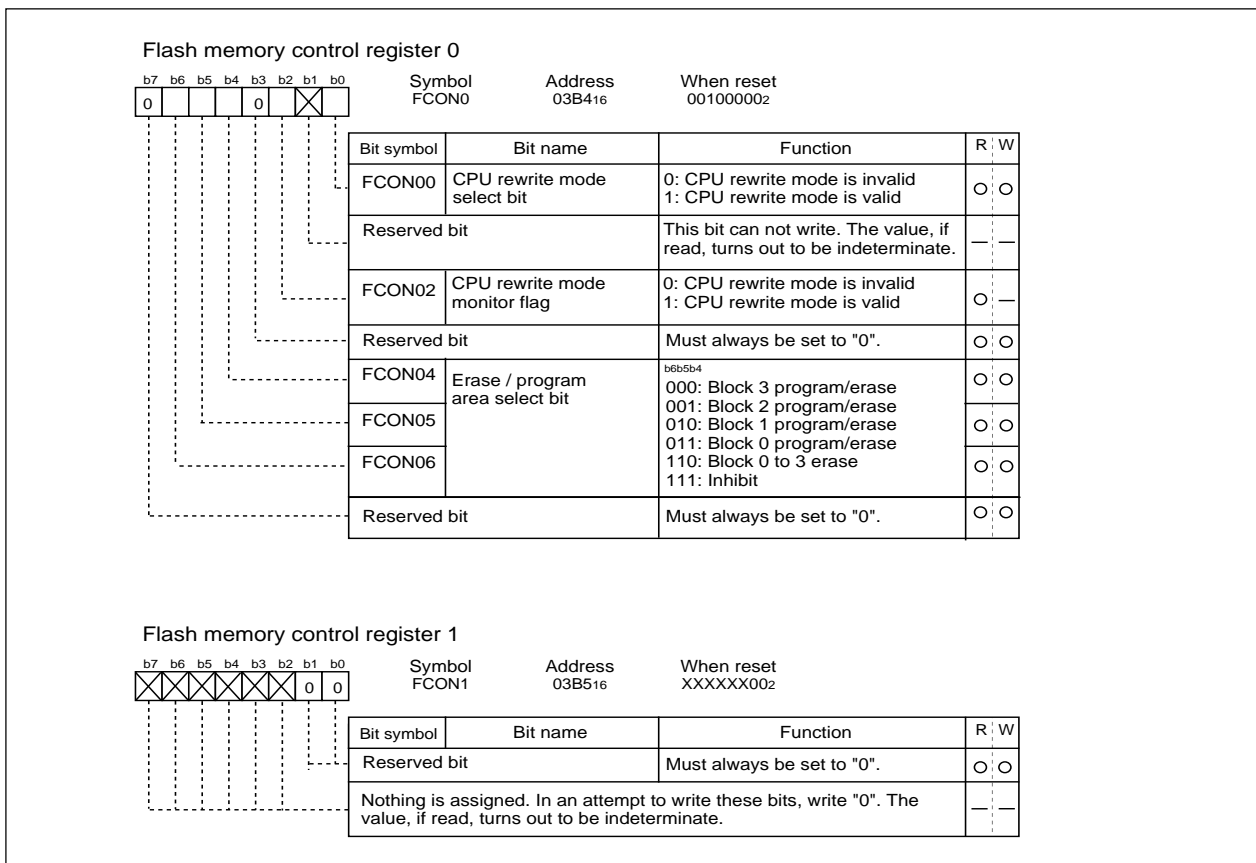


Figure BB-1. Flash memory control register

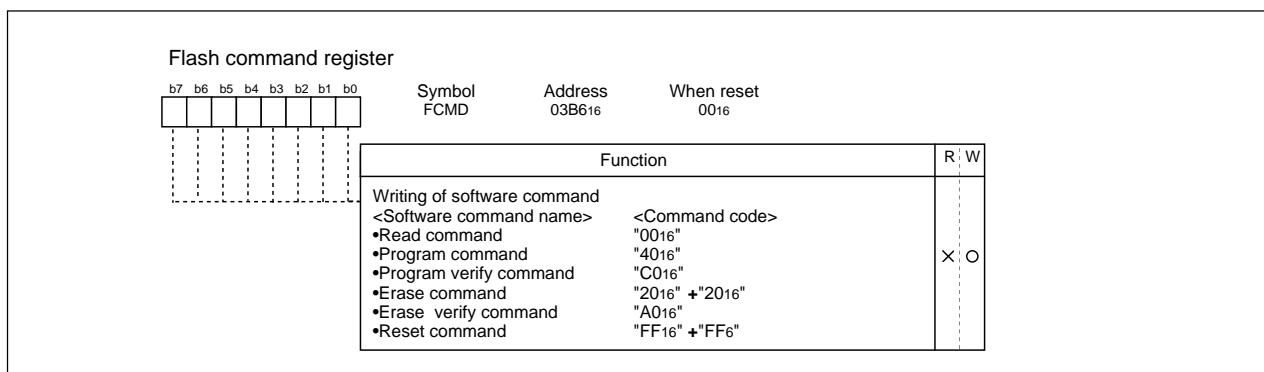


Figure BB-2. Flash command register

## Microcomputer Mode and Boot Mode

The control program for CPU rewrite mode must be written into the user ROM or boot ROM area in parallel I/O mode beforehand. (If the control program is written into the boot ROM area, the standard serial I/O mode becomes unusable.)

See Figure AA-3 for details about the boot ROM area.

Normal microcomputer mode is entered when the microcomputer is reset with pulling CNVSS pin low (Vss). In this case, the CPU starts operating using the control program in the user ROM area.

When the microcomputer is reset by pulling the P52 pin high (VCC), the CNVSS pin high (VPPH), the CPU starts operating using the control program in the boot ROM area. This mode is called the "boot" mode.

The control program in the boot ROM area can also be used to rewrite the user ROM area.

## CPU rewrite mode operation procedure

The internal flash memory can be operated on to program, read, verify, or erase it while being placed on-board by writing commands from the CPU to the flash memory control register (addresses 03B416, 03B516) and flash command register (address 03B616). Note that when in CPU rewrite mode, the boot ROM area cannot be accessed for program, read, verify, or erase operations. Before this can be accomplished, a CPU write control program must be written into the boot ROM area in parallel input/output mode. The following shows a CPU rewrite mode operation procedure.

### <Start procedure (Note 1)>

- (1) Apply VPPH to the CNVSS/VPP pin and VCC to the port P46 pin for reset release. Or the user can jump from the user ROM area to the boot ROM area using the JMP instruction and execute the CPU write control program. In this case, set the CPU write mode select bit of the flash memory control register to "1" before applying VPPH to the CNVSS/VPP pin.
- (2) After transferring the CPU write control program from the boot ROM area to the internal RAM, jump to this control program in RAM. (The operations described below are controlled by this program.)
- (3) Set the CPU rewrite mode select bit to "1".
- (4) Read the CPU rewrite mode monitor flag to see that the CPU rewrite mode is enabled.
- (5) Execute operation on the flash memory by writing software commands to the flash command register.

Note 1: In addition to the above, various other operations need to be performed, such as for entering the data to be written to flash memory from an external source (e.g., serial I/O), initializing the ports, and writing to the watchdog timer.

### <Clearing procedure>

- (1) Apply VSS to the CNVSS/VPP pin.
- (2) Set the CPU rewrite mode select bit to "0".

**Precautions on CPU Rewrite Mode**

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

**(1) Operation speed**

During erase/program mode, set BCLK to one of the following frequencies by changing the divide ratio:

5 MHz or less when wait bit (bit 7 at address 0005<sub>16</sub>) = 0 (without internal access wait state)

10 MHz or less when wait bit (bit 7 at address 0005<sub>16</sub>) = 1 (with internal access wait state)(Note 1)

**(2) Instructions inhibited against use**

The instructions listed below cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory:

UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

**(3) Interrupts inhibited against use**

No interrupts can be used that look up the fixed vector table in the flash memory area. Maskable interrupts may be used by setting the interrupt vector table in a location outside the flash memory area.

Note 1: Internal access wait state can be set in CPU rewrite mode. In this time, the following function is only used.

- CPU, ROM, RAM, timer, UART, SI/O2(non-automatic transfer), port

In case of setting internal access wait state, refer to the following explain (software wait).

**Software wait**

A software wait can be inserted by setting the wait bit (bit 7) of the processor mode register 1 (address 0005<sub>16</sub>) (Note 2).

A software wait is inserted in the internal ROM/RAM area by setting the wait bit of the processor mode register 1. When set to "0", each bus cycle is executed in one BCLK cycle. When set to "1", each bus cycle is executed in two BCLK cycles. After the microcomputer has been reset, this bit defaults to "0".

The SFR area is always accessed in two BCLK cycles regardless of the setting of this control bit.

Table DA-1 shows the software wait and bus cycles. Figure DA-6 shows example bus timing when using software waits.

Note 2: Before attempting to change the contents of the processor mode register 1, set bit 1 of the protect register (address 000A<sub>16</sub>) to "1".

**Table DA-1. Software waits and bus cycles**

Area	Wait bit	Bus cycle
SFR	Invalid	2 BCLK cycles
Internal ROM/RAM	0	1 BCLK cycle
	1	2 BCLK cycles

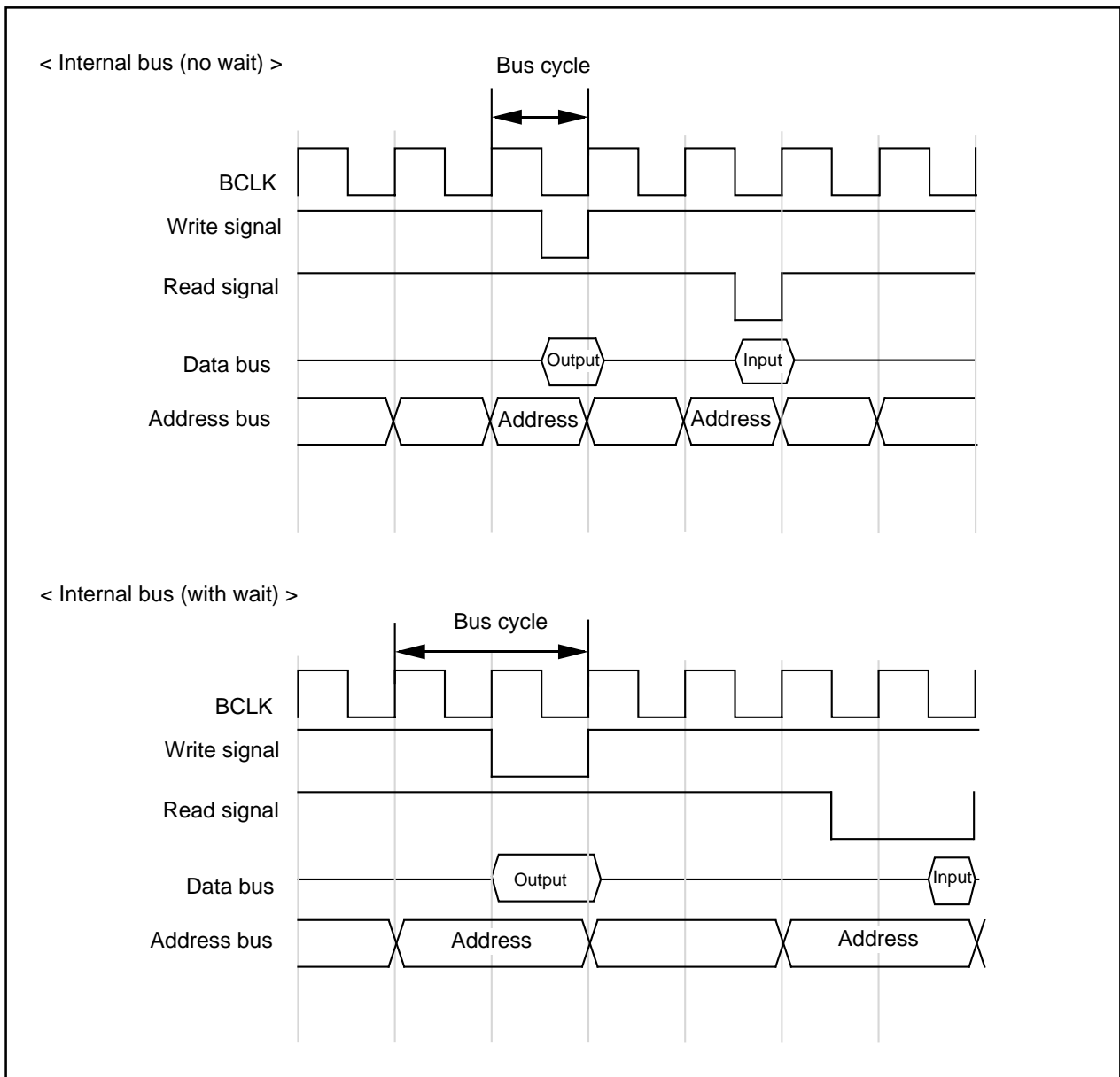


Figure DA-6. Typical bus timings using software wait

## Software Commands

Table BB-1 lists the software commands available with the M30218 group (flash memory version).

When CPU rewrite mode is enabled, write software commands to the flash command register to specify the operation to erase or program.

The content of each software command is explained below.

**Table BB-1. List of Software Commands (CPU Rewrite Mode)**

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D <sub>0</sub> to D <sub>7</sub> )	Mode	Address	Data (D <sub>0</sub> to D <sub>7</sub> )
Read	Write	03B6 <sub>16</sub>	00 <sub>16</sub>			
Program	Write	03B6 <sub>16</sub>	40 <sub>16</sub>	Write	Program address	Program data
Program verify	Write	03B6 <sub>16</sub>	C0 <sub>16</sub>	Read	Verify address	Verify data
Erase	Write	03B6 <sub>16</sub>	20 <sub>16</sub>	Write	03B6 <sub>16</sub>	20 <sub>16</sub>
Erase verify	Write	03B6 <sub>16</sub>	A0 <sub>16</sub>	Read	Verify address	Verify data
Reset	Write	03B6 <sub>16</sub>	FF <sub>16</sub>	Write	03B6 <sub>16</sub>	FF <sub>16</sub>

### Read Command (00<sub>16</sub>)

The read mode is entered by writing the command code “00<sub>16</sub>” to the flash command register in the first bus cycle. When an address to be read is input in one of the bus cycles that follow, the content of the specified address is read out at the data bus (D<sub>0</sub>–D<sub>7</sub>), 8 bits at a time.

The read mode is retained intact until another command is written.

After reset and after the reset command is executed, the read mode is set.

### Program Command (40<sub>16</sub>)

The program mode is entered by writing the command code “40<sub>16</sub>” to the flash command register in the first bus cycle. When the user execute an instruction to write byte data to the desired address (e.g., STE instruction) in the second bus cycle, the flash memory control circuit executes the program operation. The program operation requires approximately 20 μs. Wait for 20 μs or more before the user go to the next processing.

During program operation, the watchdog timer remains idle, with the value “7FFF<sub>16</sub>” set in it.

Note 1: The write operation is not completed immediately by writing a program command once. The user must always execute a program-verify command after each program command executed. And if verification fails, the user need to execute the program command repeatedly until the verification passes. See Figure BB.3 for an example of a programming flowchart.

**Program-verify command (C0<sub>16</sub>)**

The program-verify mode is entered by writing the command code "C0<sub>16</sub>" to the flash command register in the first bus cycle. When the user execute an instruction (e.g., LDE instruction) to read byte data from the address to be verified (the previously programmed address) in the second bus cycle, the content that has actually been written to the address is read out from the memory.

The CPU compares this read data with the data that it previously wrote to the address using the program command. If the compared data do not match, the user need to execute the program and program-verify operations one more time.

**Erase command (20<sub>16</sub> + 20<sub>16</sub>)**

The flash memory control circuit executes an erase operation by writing command code "20<sub>16</sub>" to the flash command register in the first bus cycle and the same command code to the flash command register again in the second bus cycle. The erase operation requires approximately 20 ms. Wait for 20 ms or more before the user go to the next processing.

Before this erase command can be performed, all memory locations to be erased must have had data "00<sub>16</sub>" written to by using the program and program-verify commands. During erase operation, the watchdog timer remains idle, with the value "7FFF<sub>16</sub> set in it.

Note 1: The erase operation is not completed immediately by writing an erase command once. The user must always execute an erase-verify command after each erase command executed. And if verification fails, the user need to execute the erase command repeatedly until the verification passes. See Figure BB-3 for an example of an erase flowchart.

**Erase-verify command (A0<sub>16</sub>)**

The erase-verify mode is entered by writing the command code "A0<sub>16</sub>" to the flash command register in the first bus cycle. When the user execute an instruction to read byte data from the address to be verified (e.g., LDE instruction) in the second bus cycle, the content of the address is read out.

The CPU must sequentially erase-verify memory contents one address at a time, over the entire area erased. If any address is encountered whose content is not "FF<sub>16</sub>" (not erased), the CPU must stop erase-verify at that point and execute erase and erase-verify operations one more time.

Note 1: If any unerased memory location is encountered during erase-verify operation, be sure to execute erase and erase-verify operations one more time. In this case, however, the user does not need to write data "00<sub>16</sub>" to memory before erasing.



**Reset command (FF16 + FF16)**

The reset command is used to stop the program command or the erase command in the middle of operation. After writing command code "4016" or "2016" twice to the flash command register, write command code "FF16" to the flash command register in the first bus cycle and the same command code to the flash command register again in the second bus cycle. The program command or erase command is disabled, with the flash memory placed in read mode.

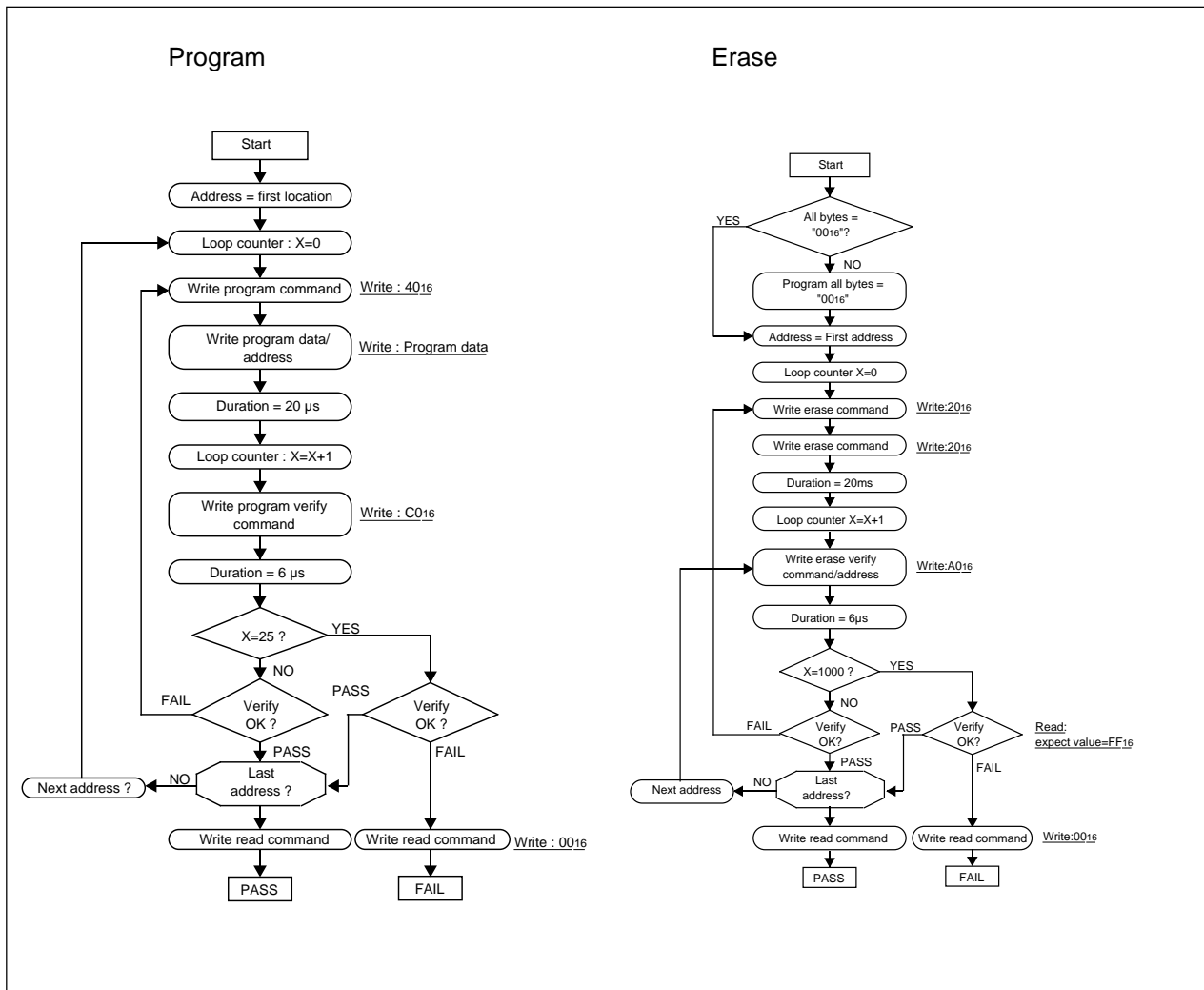
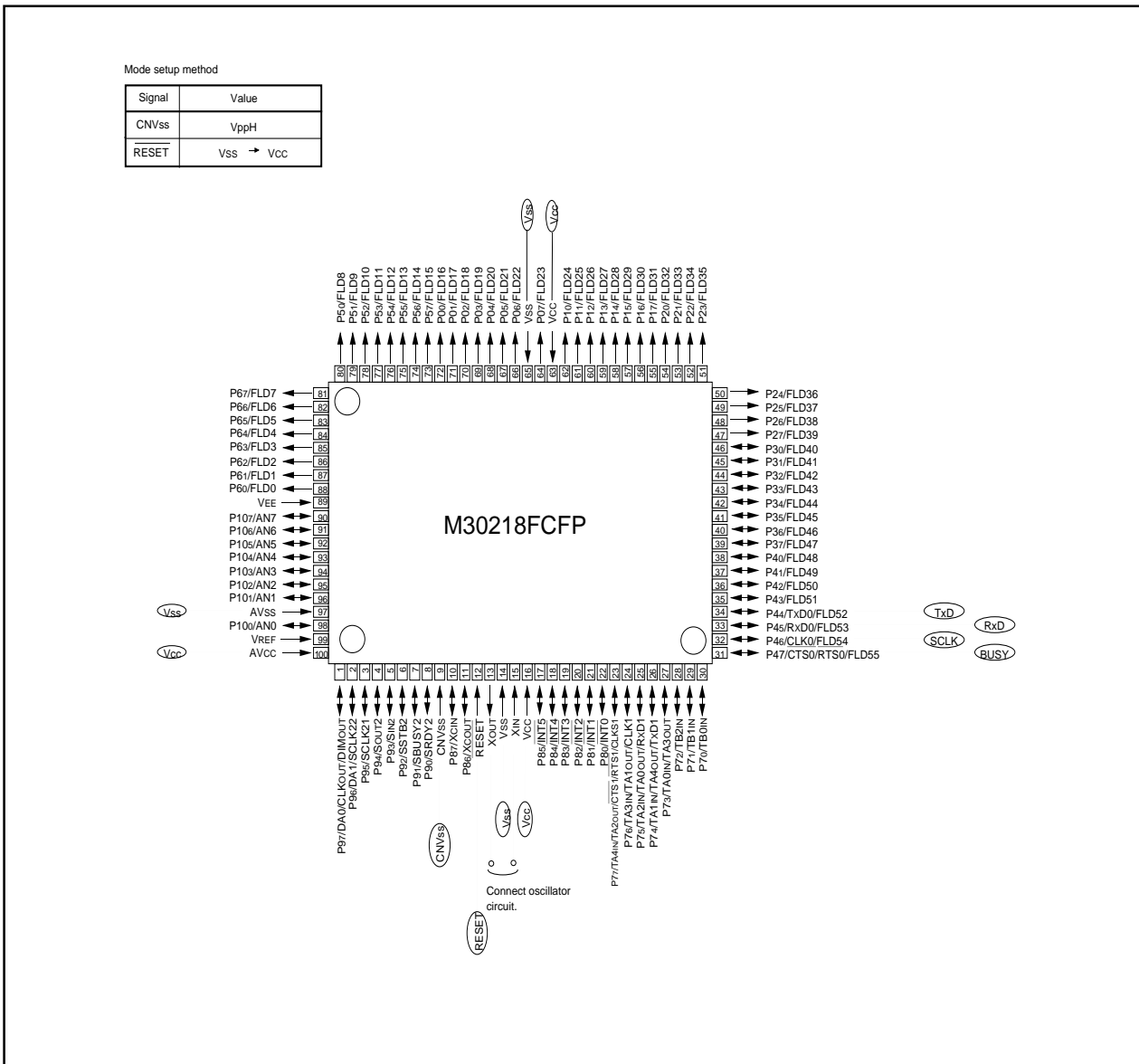


Figure BB-3. Program and erase execution flowchart in the CPU rewrite mode

**Pin functions (Flash memory standard serial I/O mode)**

Pin	Name	I/O	Description
Vcc, Vss	Power input		Apply 5V $\pm$ 10 % to Vcc pin and 0 V to Vss pin.
CNVss	CNVss	I	Apply 12V $\pm$ 5 % to this pin.
RESET	Reset input	I	Reset input pin. While reset is "L" level, a 20 cycle or longer clock must be input to XIN pin.
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock output	O	
AVcc, AVss	Analog power supply input		Connect AVss to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for AD from this pin.
P00 to P07	Output port P0	O	Output exclusive use pin.
P10 to P17	Output port P1	O	Output exclusive use pin.
P20 to P27	Output port P2	O	Output exclusive use pin.
P30 to P37	Input port P3	I	Input "H" or "L" level signal or open.
P40 to P43	Input port P4	I	Input "H" or "L" level signal or open.
P44	TxD output	O	Serial data output pin.
P45	RxD input	I	Serial data input pin.
P46	SCLK input	I	Serial clock input pin.
P47	BUSY output	O	BUSY signal output pin.
P50 to P57	Output port P5	O	Output exclusive use pin.
P60 to P67	Output port P6	O	Output exclusive use pin.
P70 to P77	Input port P7	I	Input "H" or "L" level signal or open.
P80 to P87	Input port P8	I	Input "H" or "L" level signal or open.
P90 to P97	Input port P9	I	Input "H" or "L" level signal or open.
P100 to P107	Input port P10	I	Input "H" or "L" level signal or open.



## Standard Serial I/O Mode

The standard serial I/O mode serially inputs and outputs the software commands, addresses and data necessary for operating (read, program, erase, etc.) the internal flash memory. It uses a purpose-specific serial programmer.

The standard serial I/O mode differs from the parallel I/O mode in that the CPU controls operations like rewriting (uses the CPU rewrite mode) in the flash memory or serial input for rewriting data. The standard serial I/O mode is started by clearing the reset with  $V_{PPH}$  at the CNVss pin. (For the normal microprocessor mode, set CNVss to "L".)

This control program is written in the boot ROM area when shipped from Mitsubishi Electric. Therefore, if the boot ROM area is rewritten in the parallel I/O mode, the standard serial I/O mode cannot be used. Figure DD-1 shows the pin connections for the standard serial I/O mode. Serial data I/O uses three UART0 pins: CLK0, RxDo, TxDo, and RTS0 (BUSY).

The CLK0 pin is the transfer clock input pin and it transfers the external transfer clock. The TxDo pin outputs the CMOS signal. The RTS0 (BUSY) pin outputs an "L" level when reception setup ends and an "H" level when the reception operation starts. Transmission and reception data is transferred serially in 8-byte blocks.

In the standard serial I/O mode, only the user ROM area shown in Figure AA-3 can be rewritten, the boot ROM area cannot.

The standard serial I/O mode has a 7-byte ID code. When the flash memory is not blank and the ID code does not match the content of the flash memory, the command sent from the programmer is not accepted.

## Function Overview (Standard Serial I/O Mode)

In the standard serial I/O mode, software commands, addresses and data are input and output between the flash memory and an external device (serial programmer, etc.) using a clock synchronized serial I/O (UART0). In reception, the software commands, addresses and program data are synchronized with the rise of the transfer clock input to the CLK0 pin and input into the flash memory via the RxDo pin.

In transmission, the read data and status are synchronized with the fall of the transfer clock and output to the outside from the TxDo pin.

The TxDo pin is CMOS output. Transmission is in 8-bit blocks and LSB first.

When busy, either during transmission or reception, or while executing an erase operation or program, the RTS0 (BUSY) pin is "H" level. Accordingly, do not start the next transmission until the RTS0 (BUSY) pin is "L" level.

Also, data in memory and the status register can be read after inputting a software command. It is possible to check flash memory operating status or whether a program or erase operation ended successfully or in error by reading the status register.

Software commands and the status register are explained here following.

## Software Commands

Table DD-1 lists software commands. In the standard serial I/O mode, erase operations, programs and reading are controlled by transferring software commands via the RxD pin. Software commands are explained here below.

**Table DD-1. Software commands (Standard serial I/O mode)**

	Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte		When ID is not verificate
1	Page read	FF <sub>16</sub>	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41 <sub>16</sub>	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Bclock ease	20 <sub>16</sub>	Address (middle)	Address (high)	D0 <sub>16</sub>				Not acceptable
4	Erase all unlocked blocks	A7 <sub>16</sub>	D0 <sub>16</sub>						Not acceptable
5	Read status register	70 <sub>16</sub>	SRD output	SRD1 output					Acceptable
6	Clear status register	50 <sub>16</sub>							Not acceptable
7	Read lockbit status	71 <sub>16</sub>	Address (middle)	Address (high)	Lock bit data output				Not acceptable
8	ID check function	F5 <sub>16</sub>	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
9	Download function	FA <sub>16</sub>	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
10	Version data output function	FB <sub>16</sub>	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
11	Boot area output function	FC <sub>16</sub>	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable

Note1: Shading indicates transfer from flash memory microcomputer to serial programmer. All other data is transferred from the serial programmer to the flash memory microcomputer.

Note2: SRD refers to status register data. SRD1 refers to status register 1 data.

Note3: All commands can be accepted when the flash memory is totally blank.

**Page Read Command**

This command reads the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page read command as explained here following.

- (1) Send the "FF16" command code in the 1st byte of the transmission.
- (2) Send addresses A8 to A15 and A16 to A23 in the 2nd and 3rd bytes of the transmission respectively.
- (3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first in sync with the rise of the clock.

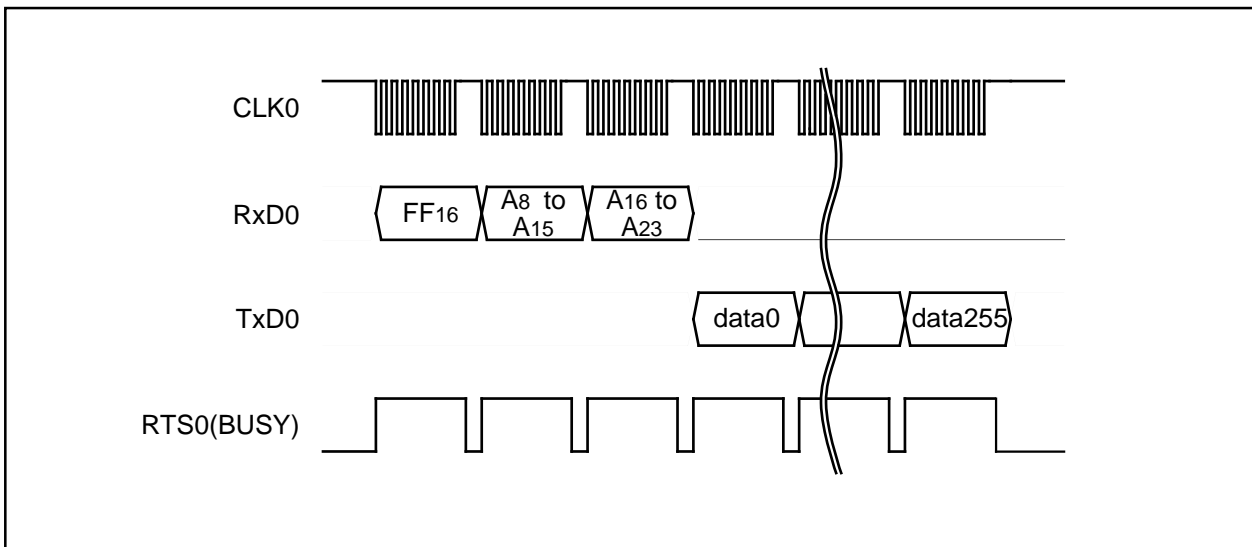


Figure DD-2. Timing for page read

**Read Status Register Command**

This command reads status information. When the "7016" command code is sent in the 1st byte of the transmission, the contents of the status register (SRD) specified in the 2nd byte of the transmission and the contents of status register 1 (SRD1) specified in the 3rd byte of the transmission are read.

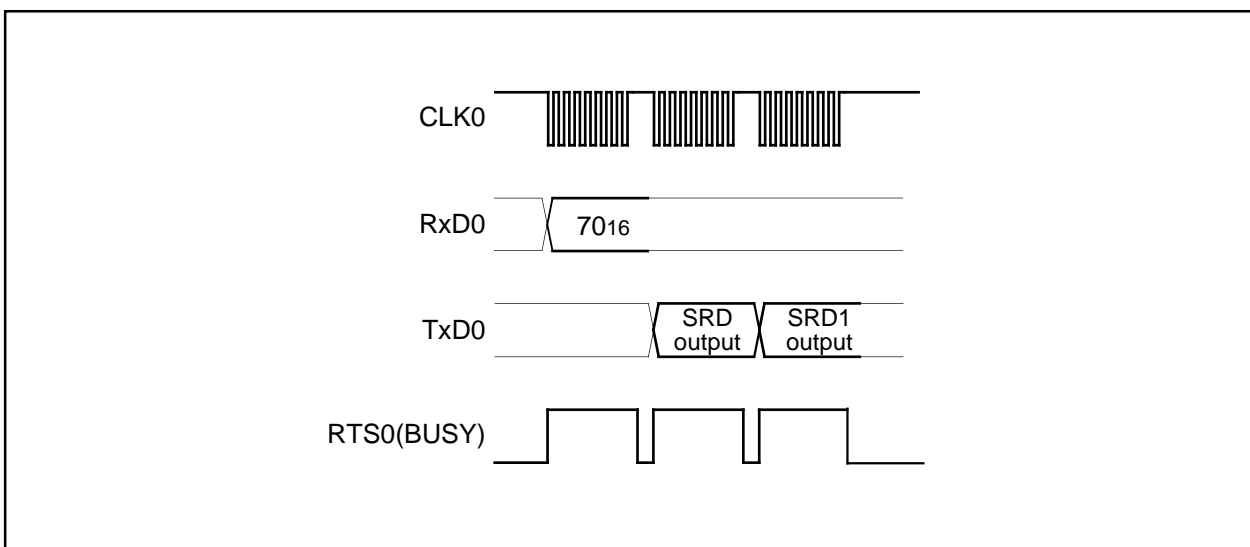


Figure DD-3. Timing for reading the status register

**Clear Status Register Command**

This command clears the bits (SR3–SR4) which are set when the status register operation ends in error. When the “5016” command code is sent in the 1st byte of the transmission, the aforementioned bits are cleared. When the clear status register operation ends, the RTS0 (BUSY) signal changes from the “H” to the “L” level.

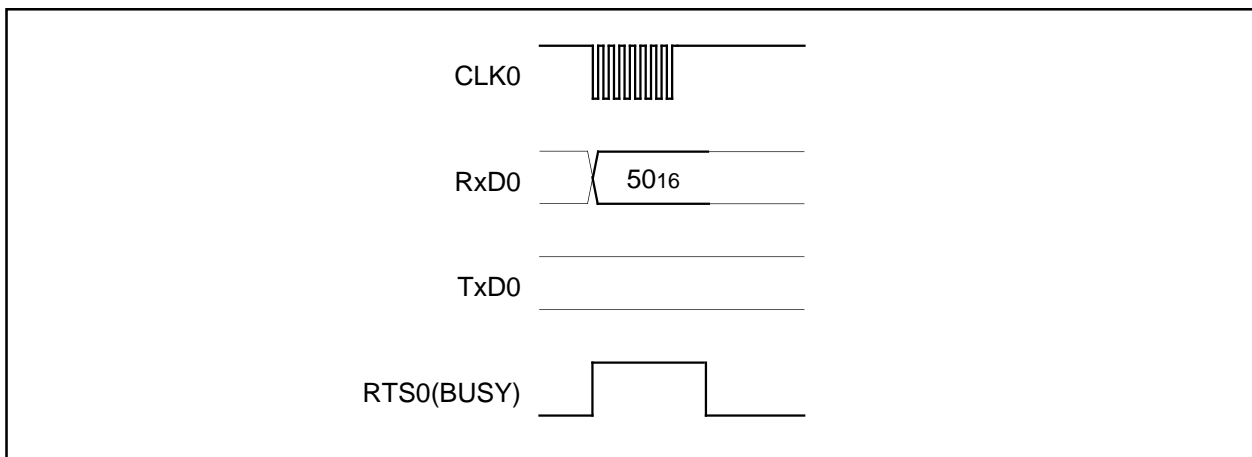


Figure DD-4. Timing for clearing the status register

**Page Program Command**

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

- (1) Send the “4116” command code in the 1st byte of the transmission.
- (2) Send addresses A8 to A15 and A16 to A23 in the 2nd and 3rd bytes of the transmission respectively.
- (3) From the 4th byte onward, as write data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the RTS0 (BUSY) signal changes from the “H” to the “L” level. The result of the page program can be known by reading the status register. For more information, see the section on the status register.

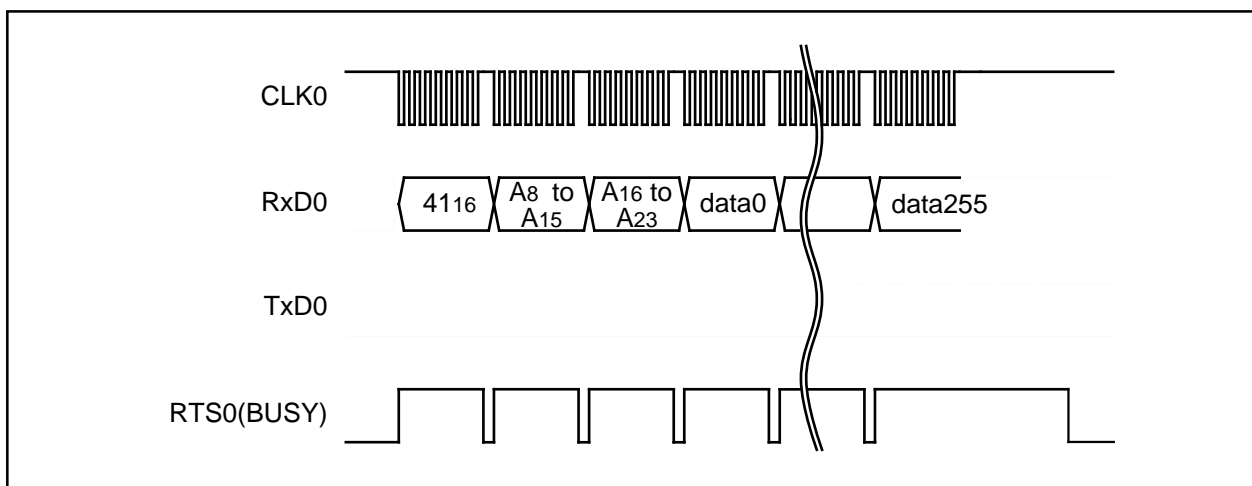


Figure DD-5. Timing for the page program

**Block Erase Command**

This command erases the data in the specified block. Execute the block erase command as explained here following.

- (1) Send the "20<sub>16</sub>" command code in the 1st byte of the transmission.
- (2) Send addresses A<sub>8</sub> to A<sub>15</sub> and A<sub>16</sub> to A<sub>23</sub> in the 2nd and 3rd bytes of the transmission respectively.
- (3) Send the verify command code "D0<sub>16</sub>" in the 4th byte of the transmission. With the verify command code, the erase operation will start for the specified block in the flash memory. Write the highest address of the specified block for addresses A<sub>16</sub> to A<sub>23</sub>.

When block erasing ends, the RTS0 (BUSY) signal changes from the "H" to the "L" level. After block erase ends, the result of the block erase operation can be known by reading the status register. For more information, see the section on the status register.

Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.

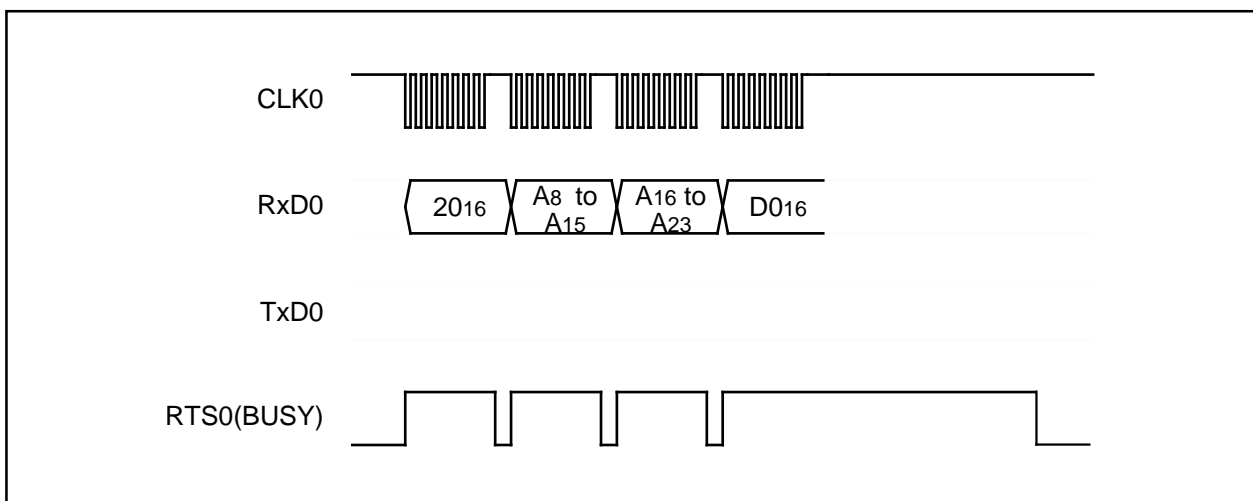


Figure DD-6. Timing for block erasing



**Erase All Unlocked Blocks Command**

This command erases the content of all blocks. Execute the erase all unlocked blocks command as explained here following.

- (1) Send the "A7<sub>16</sub>" command code in the 1st byte of the transmission.
- (2) Send the verify command code "D0<sub>16</sub>" in the 2nd byte of the transmission. With the verify command code, the erase operation will start and continue for all blocks in the flash memory.

When block erasing ends, the RTS<sub>0</sub> (BUSY) signal changes from the "H" to the "L" level. The result of the erase operation can be known by reading the status register.

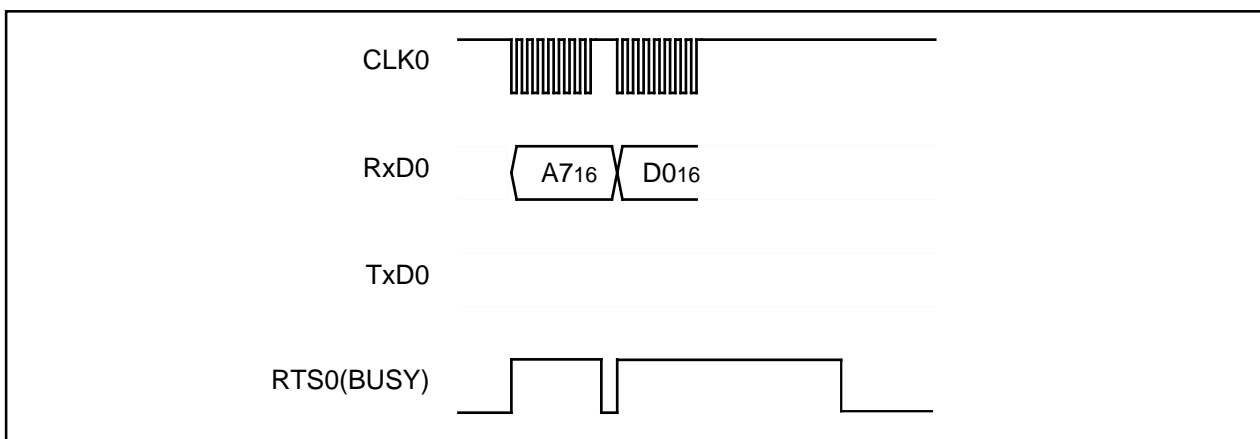


Figure DD-7. Timing for erasing all unlocked blocks

**Read Lock Bit Status Command**

This command reads the lock bit status of the specified block. Execute the read lock bit status command as explained here following.

- (1) Send the "71<sub>16</sub>" command code in the 1st byte of the transmission.
- (2) Send addresses A<sub>8</sub> to A<sub>15</sub> and A<sub>16</sub> to A<sub>23</sub> in the 2nd and 3rd bytes of the transmission respectively.
- (3) The lock bit data of the specified block is output in the 4th byte of the transmission. Write the highest address of the specified block for addresses A<sub>8</sub> to A<sub>23</sub>.

The M30218 group (flash memory version) does not have the lock bit, so the read value is always "1" (block unlock).

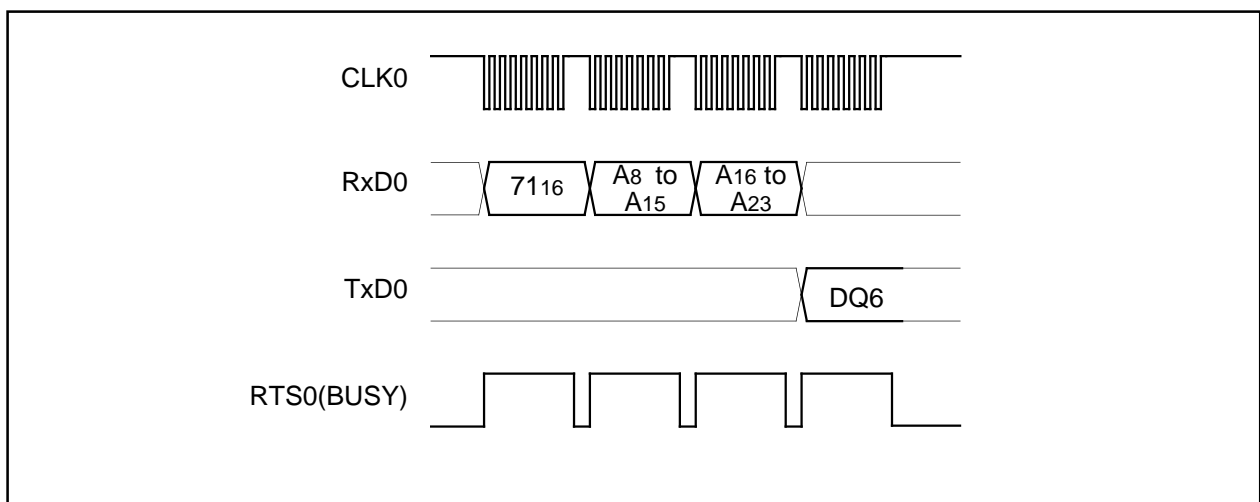


Figure DD-8. Timing for reading lock bit status

**Download Command**

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

- (1) Send the "FA16" command code in the 1st byte of the transmission.
- (2) Send the program size in the 2nd and 3rd bytes of the transmission.
- (3) Send the check sum in the 4th byte of the transmission. The check sum is added to all data sent in the 5th byte onward.
- (4) The program to execute is sent in the 5th byte onward.

When all data has been transmitted, if the check sum matches, the downloaded program is executed. The size of the program will vary according to the internal RAM.

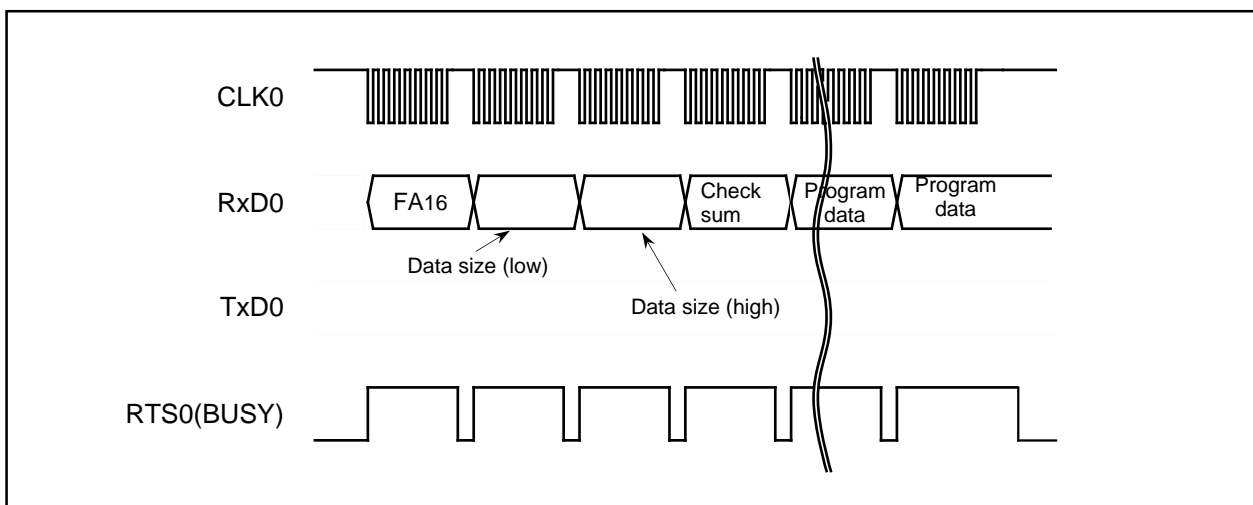


Figure DD-9. Timing for download

### Version Information Output Command

This command outputs the version information of the control program stored in the boot area. Execute the version information output command as explained here following.

- (1) Send the "FB16" command code in the 1st byte of the transmission.
- (2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

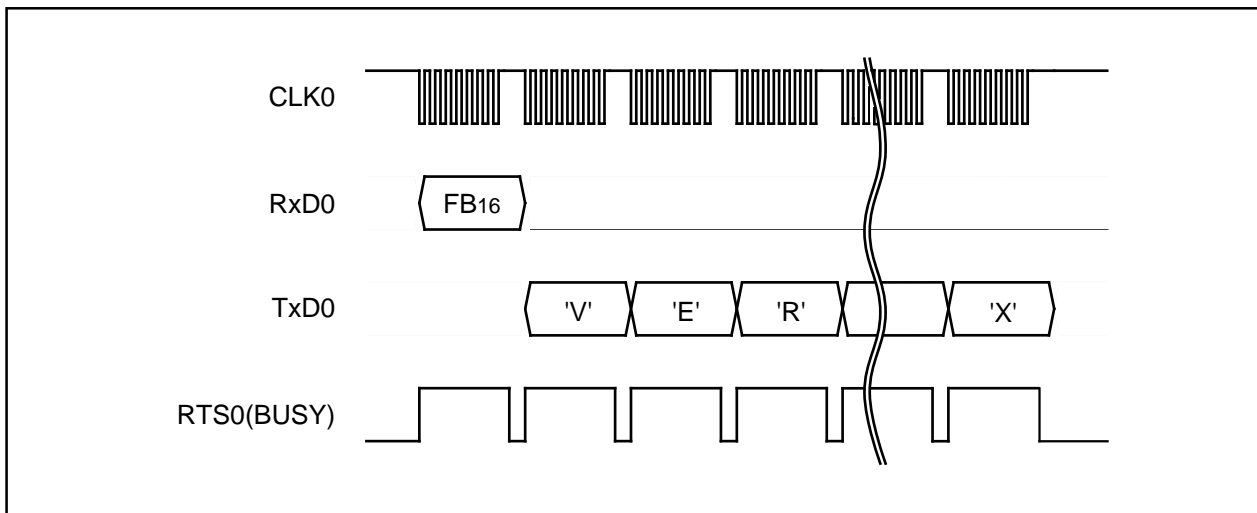


Figure DD-10. Timing for version information output

### Boot Area Output Command

This command outputs the control program stored in the boot area in one page blocks (256 bytes). Execute the boot area output command as explained here following.

- (1) Send the "FC16" command code in the 1st byte of the transmission.
- (2) Send addresses A8 to A15 and A16 to A23 in the 2nd and 3rd bytes of the transmission respectively.
- (3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first, in sync with the rise of the clock.

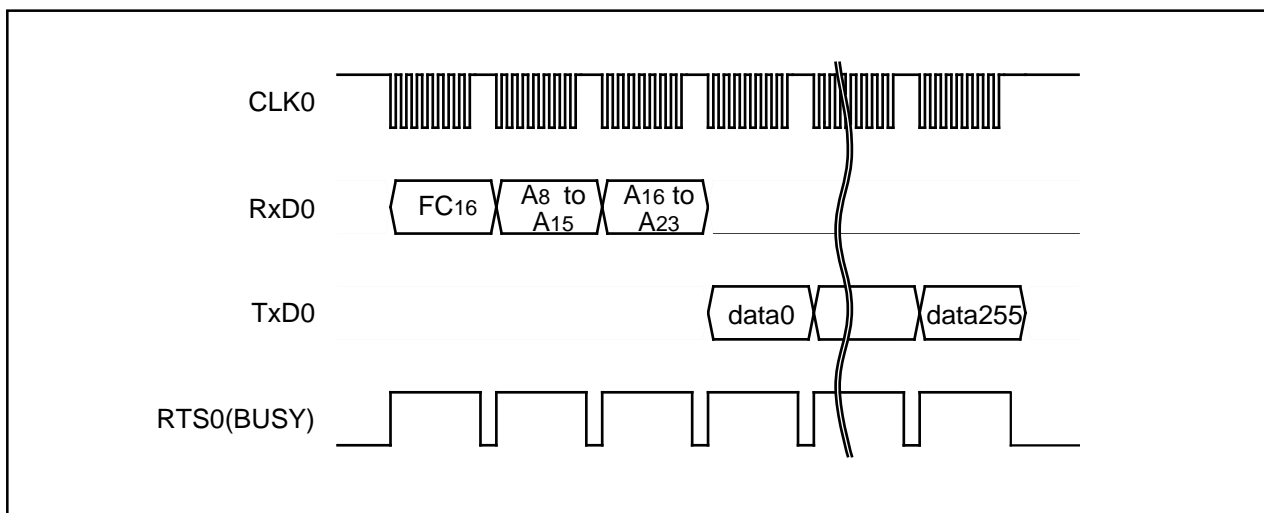


Figure DD-11. Timing for boot area output

### ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

- (1) Send the "F5<sub>16</sub>" command code in the 1st byte of the transmission.
- (2) Send addresses A<sub>0</sub> to A<sub>7</sub>, A<sub>8</sub> to A<sub>15</sub> and A<sub>16</sub> to A<sub>23</sub> of the 1st byte of the ID code in the 2nd, 3rd and 4th bytes of the transmission respectively.
- (3) Send the number of data sets of the ID code in the 5th byte.
- (4) The ID code is sent in the 6th byte onward, starting with the 1st byte of the code.

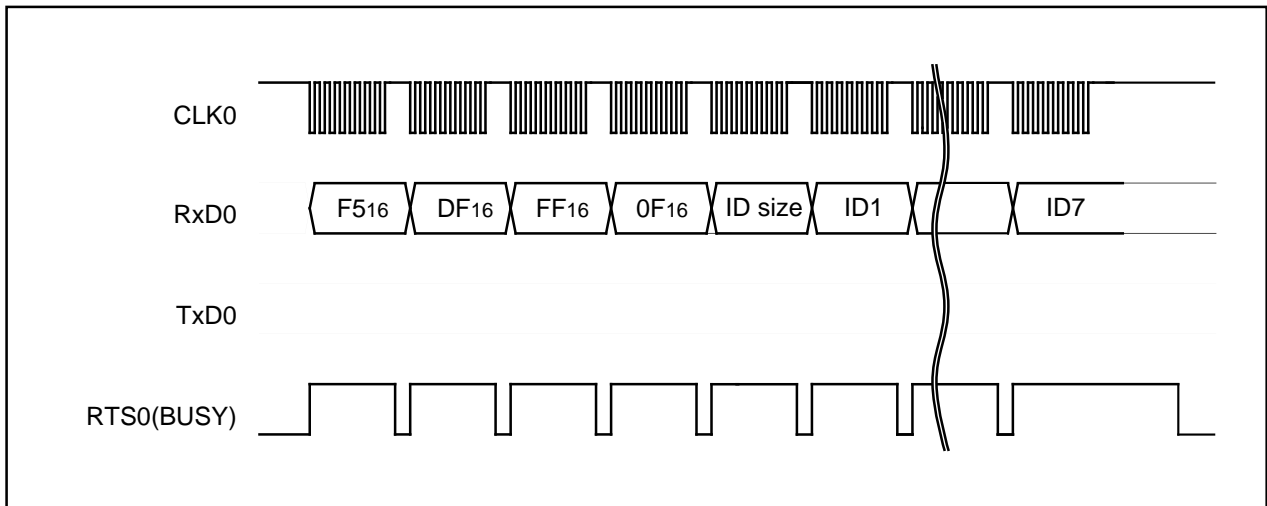


Figure DD-12. Timing for the ID check

### ID Code

When the flash memory is not blank, the ID code sent from the serial programmer and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the serial programmer is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses 0FFFDF<sub>16</sub>, 0FFFE3<sub>16</sub>, 0FFFE7<sub>16</sub>, 0FFFEF<sub>16</sub>, 0FFFF3<sub>16</sub>, and 0FFFF7<sub>16</sub>. Write a program into the flash memory, which already has the ID code set for these addresses.

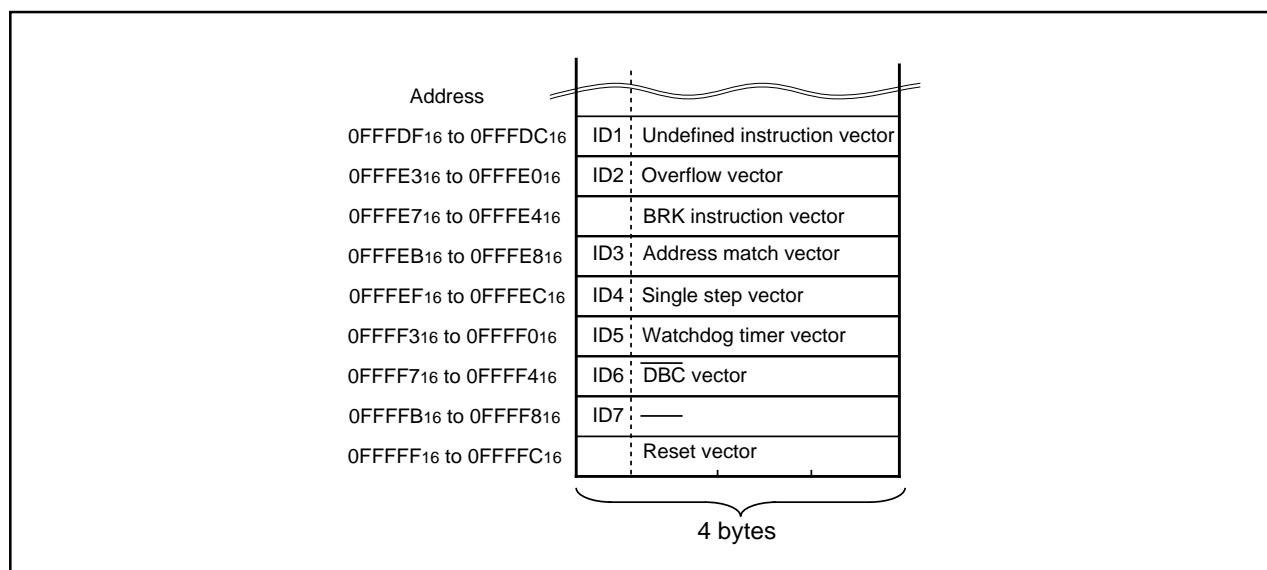


Figure DD-13. ID code storage addresses

## Status Register (SRD)

The status register indicates operating status of the flash memory and status such as whether an erase operation or a program ended successfully or in error. It can be read by writing the read status register command (70<sub>16</sub>). Also, the status register is cleared by writing the clear status register command (50<sub>16</sub>). Table DD-2 gives the definition of each status register bit. After clearing the reset, the status register outputs "80<sub>16</sub>".

**Table DD-2. Status register (SRD)**

SRD0 bits	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Status bit	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase bit	Terminated in error	Terminated normally
SR4 (bit4)	Program bit	Terminated in error	Terminated normally
SR3 (bit3)	Reserved	-	-
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

### Status Bit (SR7)

The status bit indicates the operating status of the flash memory. When power is turned on, "1" (ready) is set for it. The bit is set to "0" (busy) during an auto write or auto erase operation, but it is set back to "1" when the operation ends.

### Erase Bit (SR5)

The erase bit reports the operating status of the auto erase operation. If an erase error occurs, it is set to "1". When the erase status is cleared, it is set to "0".

### Program Bit (SR4)

The program bit reports the operating status of the auto write operation. If a write error occurs, it is set to "1". When the program status is cleared, it is set to "0".

**Status Register 1 (SRD1)**

Status register 1 indicates the status of serial communications, results from ID checks and results from check sum comparisons. It can be read after the SRD by writing the read status register command (7016). Also, status register 1 is cleared by writing the clear status register command (5016).

Table DD-3 gives the definition of each status register 1 bit. "0016" is output when power is turned ON and the flag status is maintained even after the reset.

**Table DD-3. Status register 1 (SRD1)**

SRD1 bits	Status name	Definition	
		"1"	"0"
SR15 (bit7)	Boot update completed bit	Update completed	Not update
SR14 (bit6)	Reserved	-	-
SR13 (bit5)	Reserved	-	-
SR12 (bit4)	Checksum match bit	Match	Mismatch
SR11 (bit3) SR10 (bit2)	ID check completed bits	00 01 10 11	Not verified Verification mismatch Reserved Verified
SR9 (bit1)	Data receive time out	Time out	Normal operation
SR8 (bit0)	Reserved	-	-

**Boot Update Completed Bit (SR15)**

This flag indicates whether the control program was downloaded to the RAM or not, using the download function.

**Check Sum Consistency Bit (SR12)**

This flag indicates whether the check sum matches or not when a program, is downloaded for execution using the download function.

**ID Check Completed Bits (SR11 and SR10)**

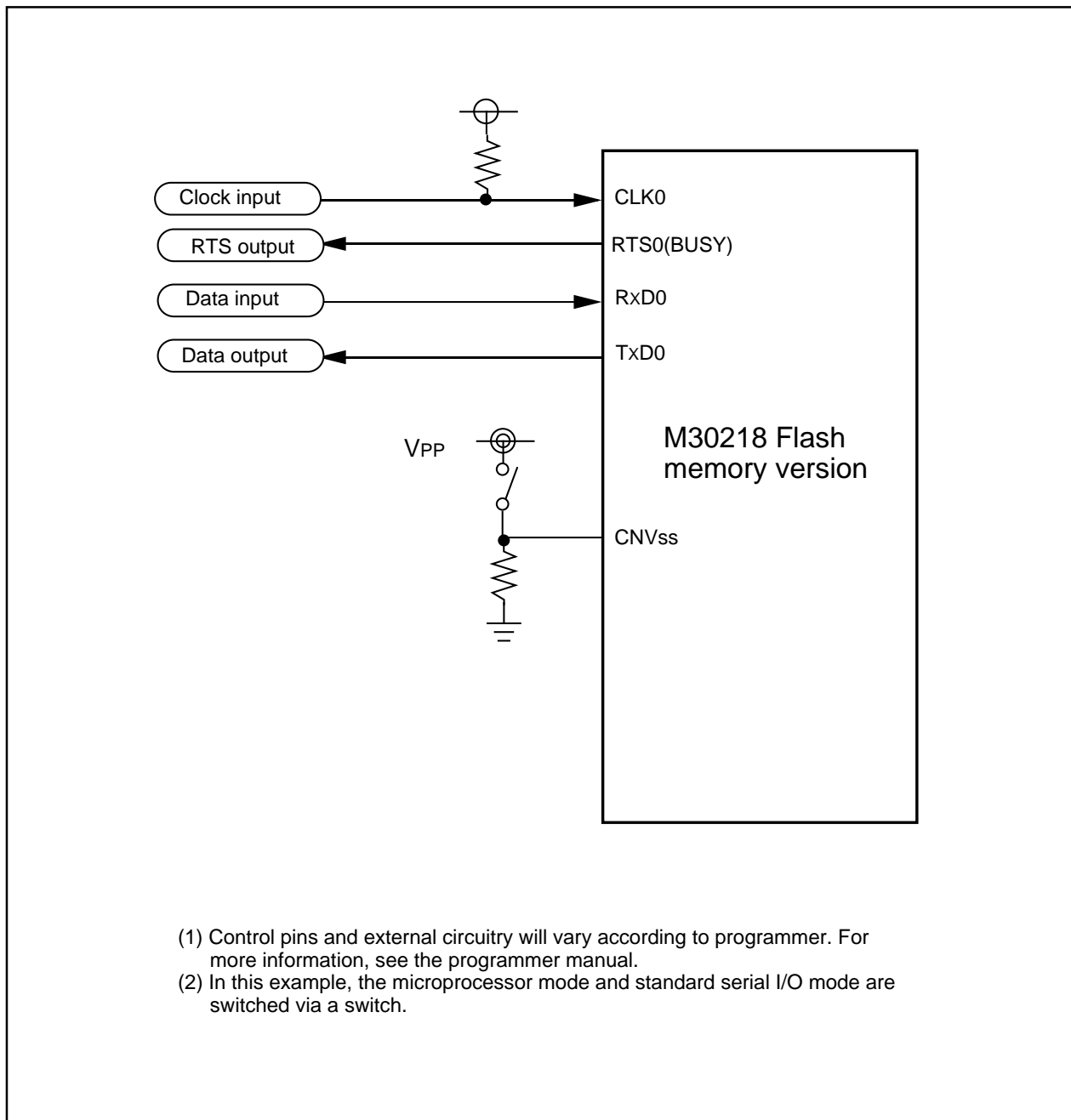
These flags indicate the result of ID checks. Some commands cannot be accepted without an ID check.

**Data Reception Time Out (SR9)**

This flag indicates when a time out error is generated during data reception. If this flag is attached during data reception, the received data is discarded and the microcomputer returns to the command wait state.

**Example Circuit Application for The Standard Serial I/O Mode**

The below figure shows a circuit application for the standard serial I/O mode. Control pins will vary according to programmer, therefore see the programmer manual for more information.

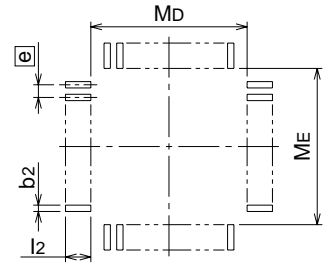
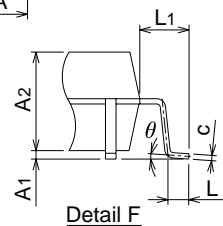
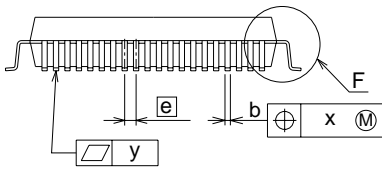
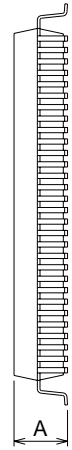
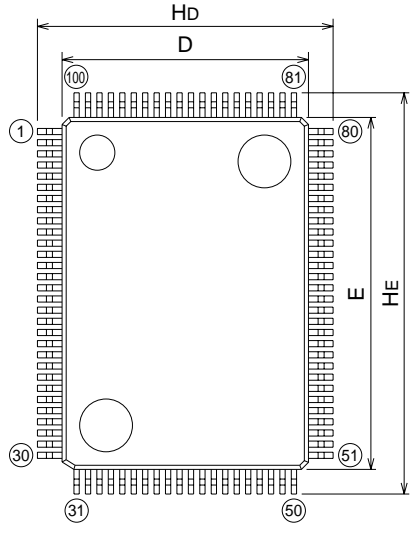


**Figure DD-14. Example circuit application for the standard serial I/O mode**

**100P6S-A** (MMP)

**Plastic 100pin 14X20mm body QFP**

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP100-P-1420-0.65	—	1.58	Alloy 42



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	3.05
A1	0	0.1	0.2
A2	—	2.8	—
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	—	0.65	—
Hd	16.5	16.8	17.1
HE	22.5	22.8	23.1
L	0.4	0.6	0.8
L1	—	1.4	—
x	—	—	0.13
y	—	—	0.1
$\theta$	0°	—	10°
b2	—	0.35	—
l2	1.3	—	—
MD	—	14.6	—
ME	—	20.6	—



## Revision History

Version	Contents for change	Revision date
REV.A1	<p>Page 2 Figure AA-1            M30218-<del>XXX</del>FP ---&gt; M30218-<del>XXXX</del>FP</p> <p>Page 10 Figure BA-3            03B8<sub>16</sub> DMA0 cause select register ---&gt; DMA0 <u>request</u> cause select register            03BA<sub>16</sub> DMA1 cause select register ---&gt; DMA1 <u>request</u> cause select register</p> <p>Page 55 Figure KA-2 FLDC mode register            bit3, bit2 (at rising edge of each <u>edge</u>) ---&gt; (at rising edge of each <u>digit</u>)            11: ---&gt; 10:</p> <p>Page 90 Figure GA-4 UARTi transmit/receive control register 0            bit4 (P47 and P74 function as) ---&gt; (P47 and P77 function as)</p> <p>Page 128 Exclusive High-breakdown]voltage Output Ports Line 2            All ports have structure of high-breakdown-voltage P-channel open drain output and pull-down resistance. ---&gt; All ports have structure of high-breakdown-voltage P-channel open drain output. Exclusive output ports except P2 have built-in pull-down resistance.</p> <p>Page 134            Add to Note 3.</p>	99.12.21
REV.B	<p>Page 150 Figure Z-34 Automatic transfer serial I/O            Decided electrical standard values (at Vcc = 3V)</p> <p>Page 153 Figure AA-3            User ROM area block number            E0000<sub>16</sub> to E7FFF<sub>16</sub> Block 0 ---&gt; Block 3            E8000<sub>16</sub> to EFFFF<sub>16</sub> Block 1 ---&gt; Block 2            F0000<sub>16</sub> to F7FFF<sub>16</sub> Block 2 ---&gt; Block 1            F8000<sub>16</sub> to FFFFF<sub>16</sub> Block 3 ---&gt; Block 0</p> <p>Page 154 Figure BB-1 Flash memory control register 0            bit6 bit5 bit4            0 0 0 : <u>Block 0</u> program/erase ---&gt; <u>Block 3</u> program/erase            0 0 1 : <u>Block 1</u> program/erase ---&gt; <u>Block 2</u> program/erase            0 1 0 : <u>Block 2</u> program/erase ---&gt; <u>Block 1</u> program/erase            0 1 1 : <u>Block 3</u> program/erase ---&gt; <u>Block 0</u> program/erase</p>	00.11.10
	<p>Page 2, 5, 6, 7, 128, 133, 134, 140, 142 and 146            Delete about mask option specification of pull-down resistor</p>	01.1.24
Revision history	M30218 Data sheet	

### Keep safety first in your circuit designs!

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

### Notes regarding these materials

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

MITSUBISHI SEMICONDUCTORS  
M30218 Group Data Sheet REV.B

---

Feb. First Edition 2001

Edited by  
Committee of editing of Mitsubishi Semiconductor

Published by  
Mitsubishi Electric Corp., Kitaitami Works

---

This book, or parts thereof, may not be reproduced in any form without  
permission of Mitsubishi Electric Corporation.

©2001 MITSUBISHI ELECTRIC CORPORATION