Description

The M16C/20 group of single-chip microcomputers are built using the high-performance silicon gate CMOS process using a M16C/60 Series CPU core. M16C/20 group is packaged in a 52-pin plastic molded SDIP, or 56-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.

The M16C/20 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 4. ROM expansion.)
• Shortest instruction execution time	100ns (f(X_{IN})=10MHz)
• Supply voltage	4.0 to 5.5V (f(X_{IN})=10MHz) :mask ROM version 2.7 to 5.5V (f(X_{IN})=7MHz with software one-wait):mask ROM version 4.0 to 5.5V (f(X_{IN})=10MHz) :flash memory version
• Interrupts	9 internal and 3 external interrupt sources, 4 software (including key input interrupt)
• Multifunction 16-bit timer	Timer A x 1, timer B x 2, timer X x 3
• Clock output	
• Serial I/O	1 channel for UART or clock synchronous, 1 for UART
• A-D converter	10 bits X 8 channels (Expandable up to 13 channels)
• Watchdog timer	1 line
• Programmable I/O	43 lines
• LED drive ports	8 ports
• Clock generating circuit	2 built-in clock generation circuits (built-in feedback resistor, and external ceramic or quartz oscillator)

Applications

Home appliances, Audio, office equipment, Automobiles

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

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

Pin Configuration

Figures 1 to 2 show the pin configurations (top view).

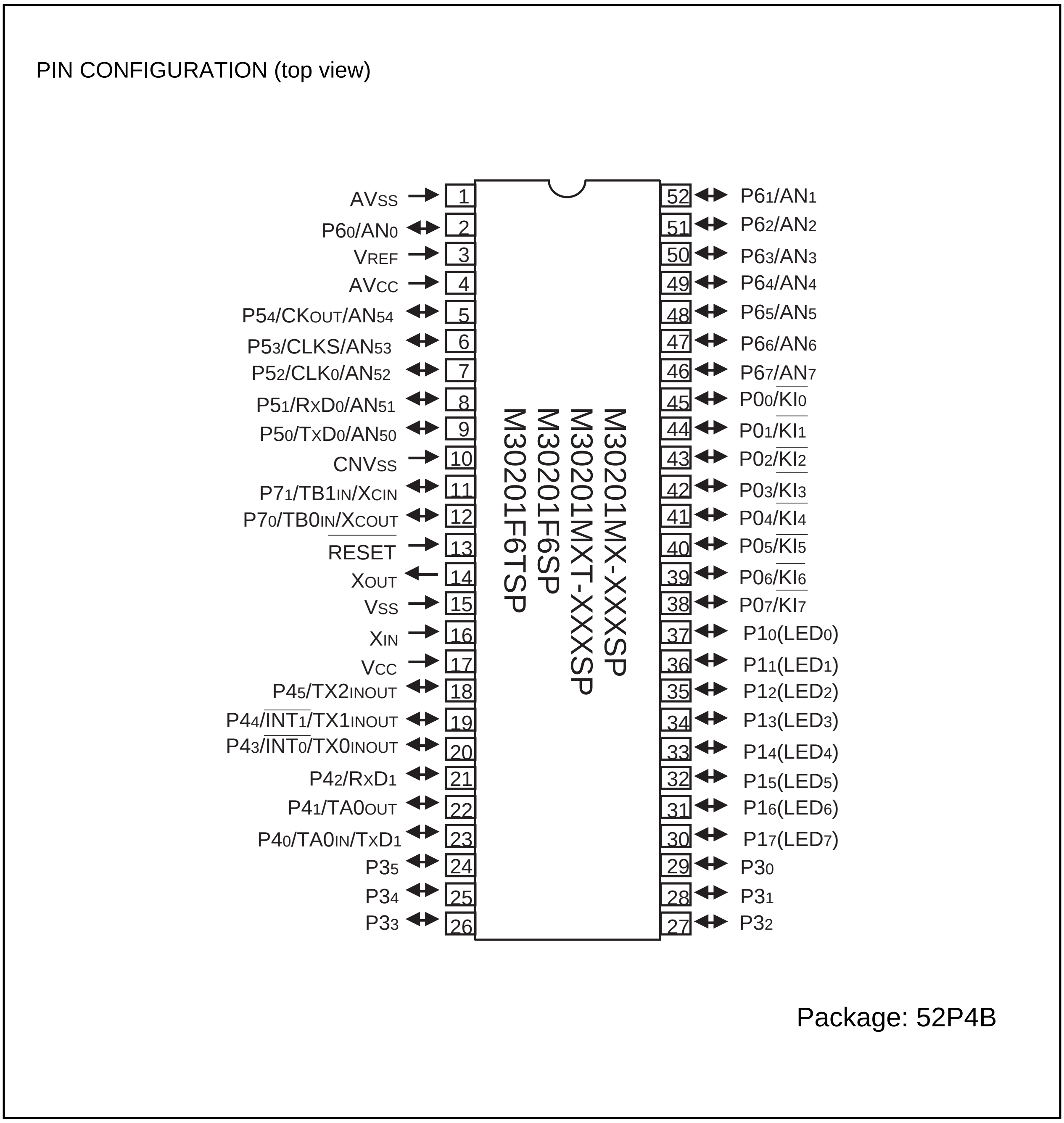
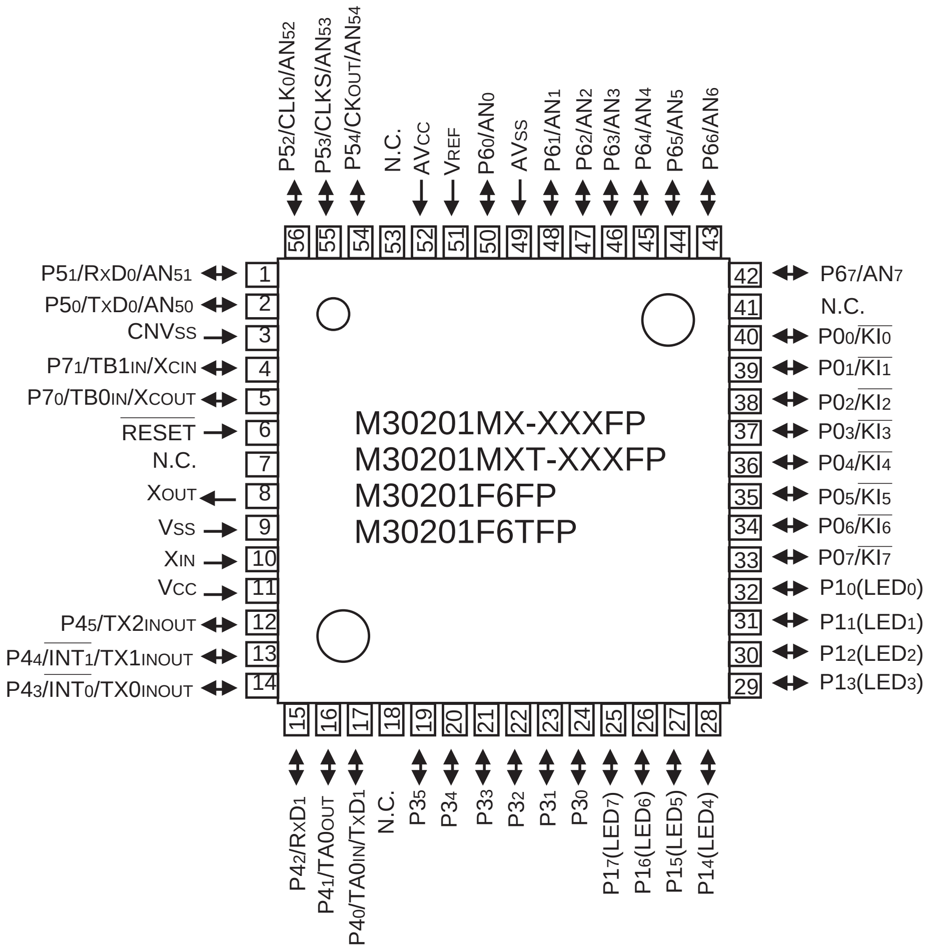


Figure 1. Pin configuration for the M16C/20 group (shrink DIP product) (top view)

PIN CONFIGURATION (top view)

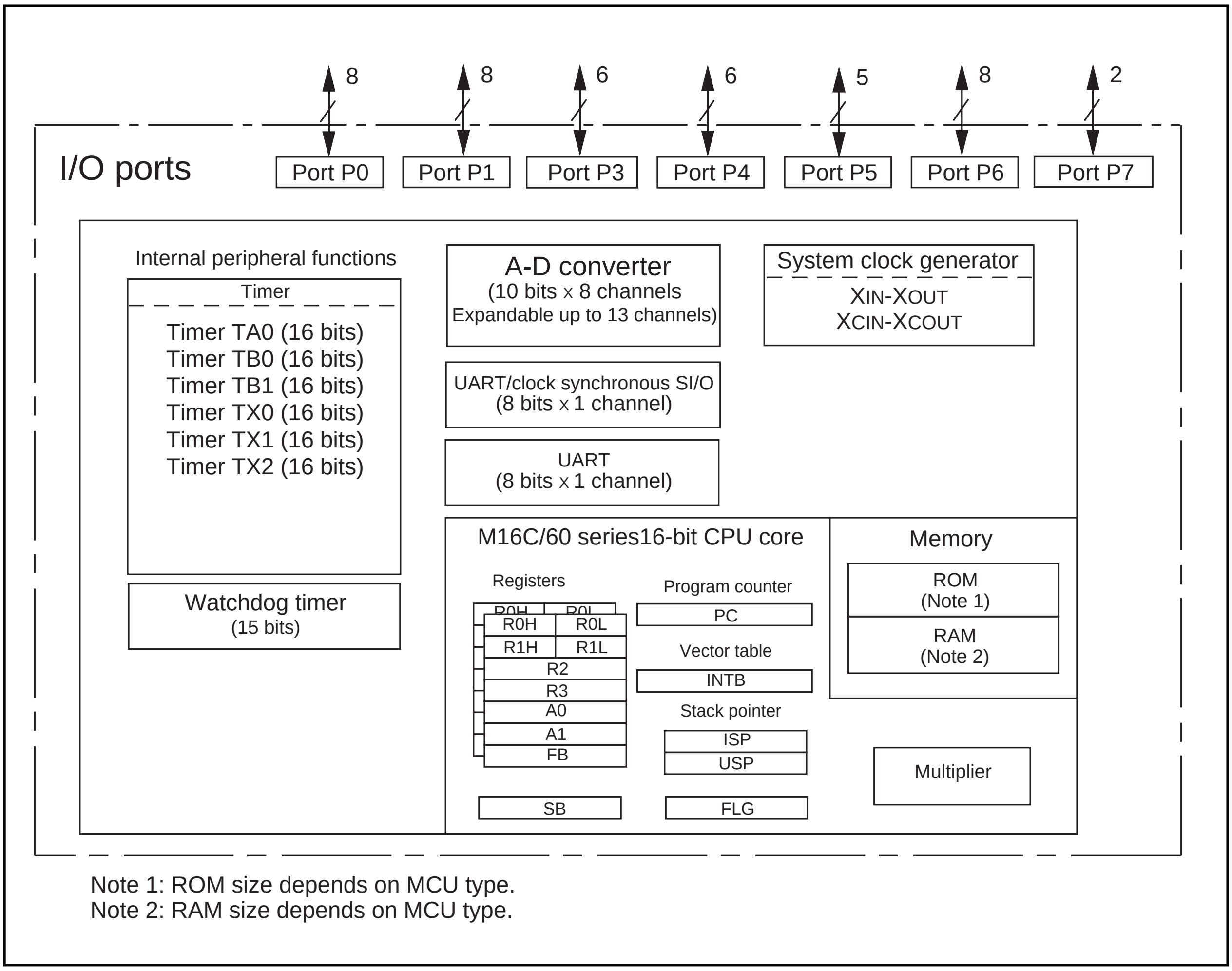


Package: 56P6S-A

Figure 2. Pin configuration for the M16C/20 group (QFP product) (top view)

Block Diagram

Figure 3 is a block diagram of the M16C/20 group.



Note 1: ROM size depends on MCU type.
Note 2: RAM size depends on MCU type.

Figure 3. Block diagram for the M16C/20 group

Performance Outline

Table 1 is performance outline of M16C/20 group.

Table 1. Performance outline of M16C/20 group

Item		Performance
Number of basic instructions		91 instructions
Shortest instruction execution time		100ns (f(X_{IN})=10MHz
Memory capacity	ROM	(See figure 4. ROM expansion.)
	RAM	(See figure 4. ROM expansion.)
I/O port	P0 to P7	43 lines
Multifunction timer	TA0	16 bits x 1
	TB0, TB1	16 bits x 2
	TX0, TX1, TX2	16 bits x 3
Serial I/O	UART0	(UART or clock synchronous) x 1
	UART1	UART x 1
A-D converter		10 bits x 8 channels (Expandable up to 13 channels)
Watchdog timer		15 bits x 1 (with prescaler)
Interrupt		9 internal and 3 external sources, 4 software sources
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(X_{IN})=10MHz) :mask ROM version 2.7 to 5.5V (f(X_{IN})=7MHz with software one-wait) :mask ROM version 4.0 to 5.5V (f(X_{IN})=10MHz) :flash memory version
Power consumption		18mW (f(X_{IN}) = 7MHz with software one-wait, Vcc=3V)
I/O characteristics	I/O withstand voltage	5V
	Output current	5mA (15mA:LED drive port)
Device configuration		CMOS silicon gate
Package		52-pin plastic mold SDIP 56-pin plastic mold QFP

Mitsubishi plans to release the following products in the M16C/20 group:

(1) Support for mask ROM version and flash memory version

(2) ROM capacity

(3) Package

52P4B	: Plastic molded SDIP (mask ROM version and flash memory version)
56P6S-A	: Plastic molded QFP (mask ROM version and flash memory version)

July 1998

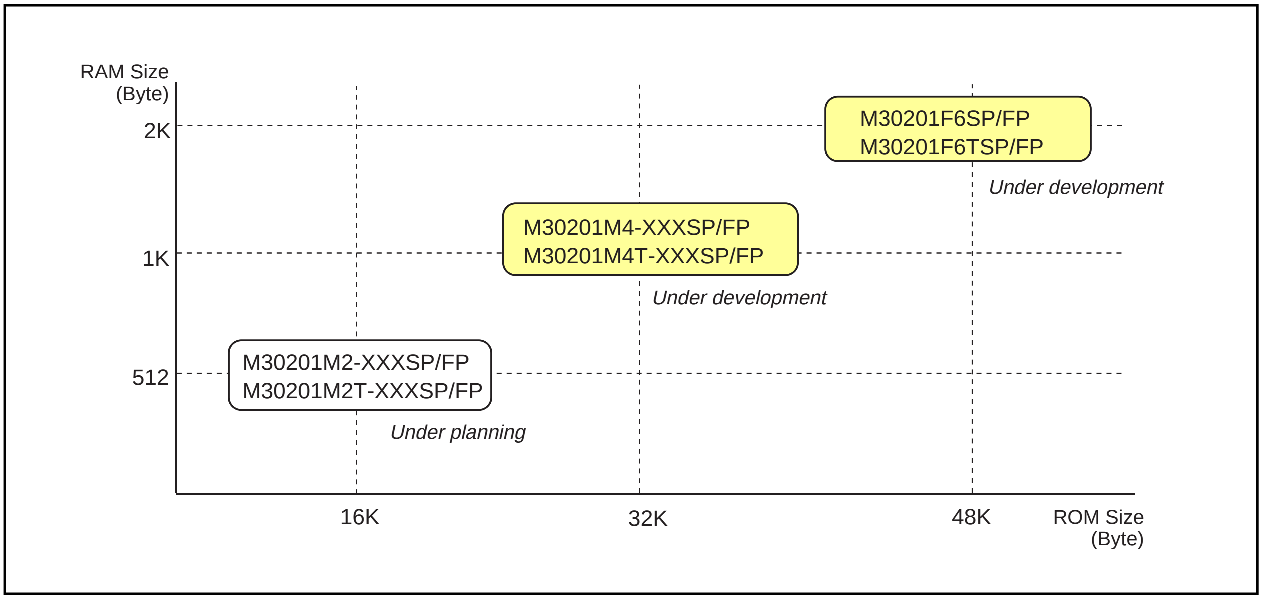


Figure 4. ROM expansion

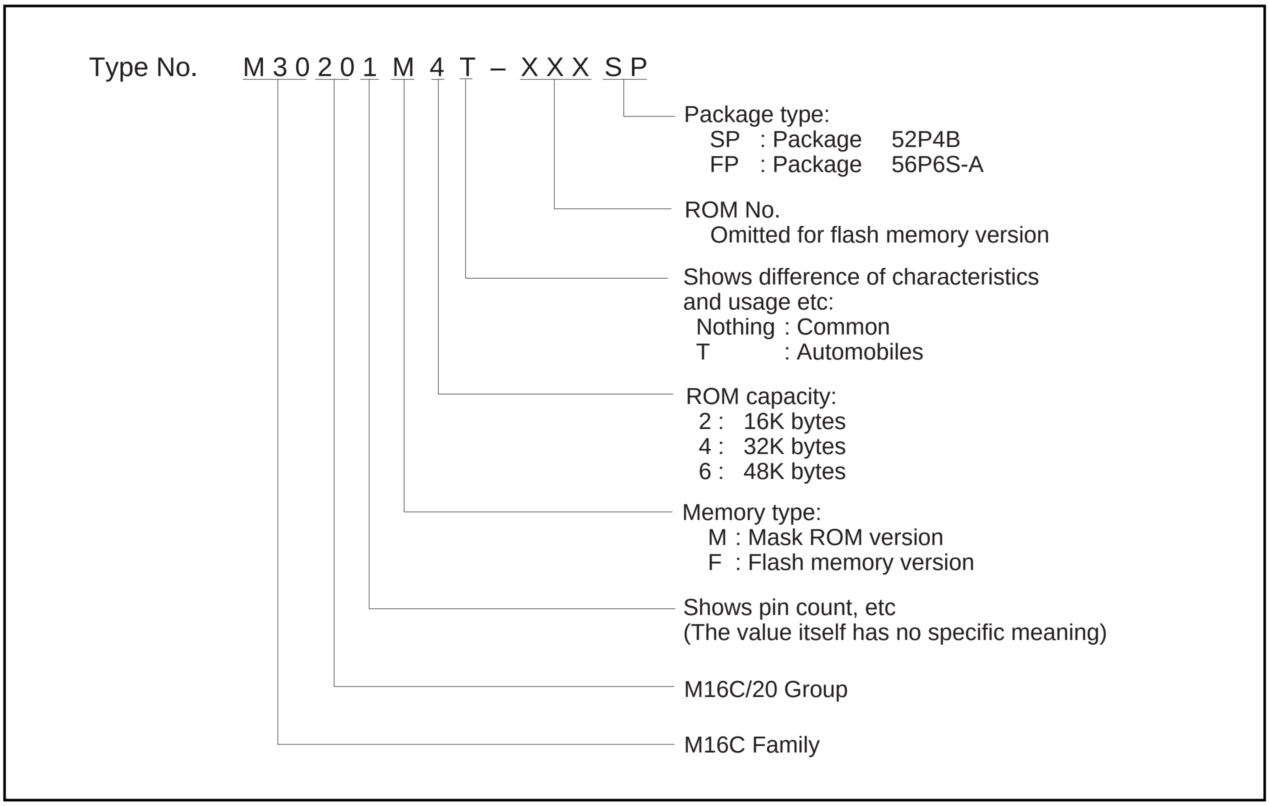


Figure 5. Type No., memory size, and package

Pin Description

Pin name	Signal name	I/O type	Function
V_{CC}, V_{SS}	Power supply input		Supply 2.7 to 5.5 V to the V_{CC} pin. Supply 0 V to the V_{SS} pin.
CNV_{SS}	CNV_{SS}	Input	Connect it to the V_{SS} pin.
$\overline{RESET}$	Reset input	Input	A “L” on this input resets the microcomputer.
X_{IN} X_{OUT}	Clock input Clock output	Input Output	These pins are provided for the main clock generating circuit. Connect a ceramic resonator or crystal between the X_{IN} and the X_{OUT} pins. To use an externally derived clock, input it to the X_{IN} pin and leave the X_{OUT} pin open.
AV_{CC}	Analog power supply input		This pin is a power supply input for the A-D converter. Connect it to V_{CC}.
AV_{SS}	Analog power supply input		This pin is a power supply input for the A-D converter. Connect it to V_{SS}.
V_{REF}	Reference voltage input	Input	This pin is a reference voltage input for the A-D converter.
$P0_0$ to $P0_7$	I/O port P0	Input/output	This is an 8-bit CMOS I/O port. It has an input/output port direction register that allows the user to set each pin for input or output individually. When set for input, the user can specify in units of four bits via software whether or not they are tied to a pull-up resistor.
$P1_0$ to $P1_7$	I/O port P1	Input/output	This is an 8-bit I/O port equivalent to P0.
$P3_0$ to $P3_5$	I/O port P3	Input/output	This is a 6-bit I/O port equivalent to P0.
$P4_0$ to $P4_5$	I/O port P4	Input/output	This is a 6-bit I/O port equivalent to P0. The $P4_0$ pin is shared with timer A0 input and serial I/O output TxD1. The $P4_1$ pin is shared with timer A0 output. The $P4_2$ pin is shared with serial I/O input RxD1. The $P4_3$ pin is shared with external interrupt INT0 and timer X0 input/output $TX0_{INOUT}$. The $P4_4$ pin is shared with external interrupt INT1 and timer X1 input/output $TX1_{INOUT}$. The $P4_5$ pin is shared with timer X2 input/output $TX2_{INOUT}$.
$P5_0$ to $P5_4$	I/O port P5	Input/output	This is a 5-bit I/O port equivalent to P0. The $P5_0$, $P5_1$, $P5_2$, and $P5_3$ pins are shared with serial I/O pins TxD_0, RxD_0, CLK_0, and CLKS. The $P5_4$ pin is shared with clock output CLK_{OUT}. Also, these pins are shared with analog input pins AN_{50} through AN_{54}.
$P6_0$ to $P6_7$	I/O port P6	Input/output	This is an 8-bit I/O port equivalent to P0. These pins are shared with analog input pins AN_0 through AN_7.
$P7_0$ to $P7_1$	I/O port P7	Input/output	This is a 2-bit I/O port equivalent to P0 . These pins are used for input/output to and from the oscillator circuit for the clock. Connect a crystal oscillator between the X_{CIN} and the X_{COUT} pins.

Operation of Functional Blocks

The M30201 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, serial I/O, A-D converter, and I/O ports.

The following explains each unit.

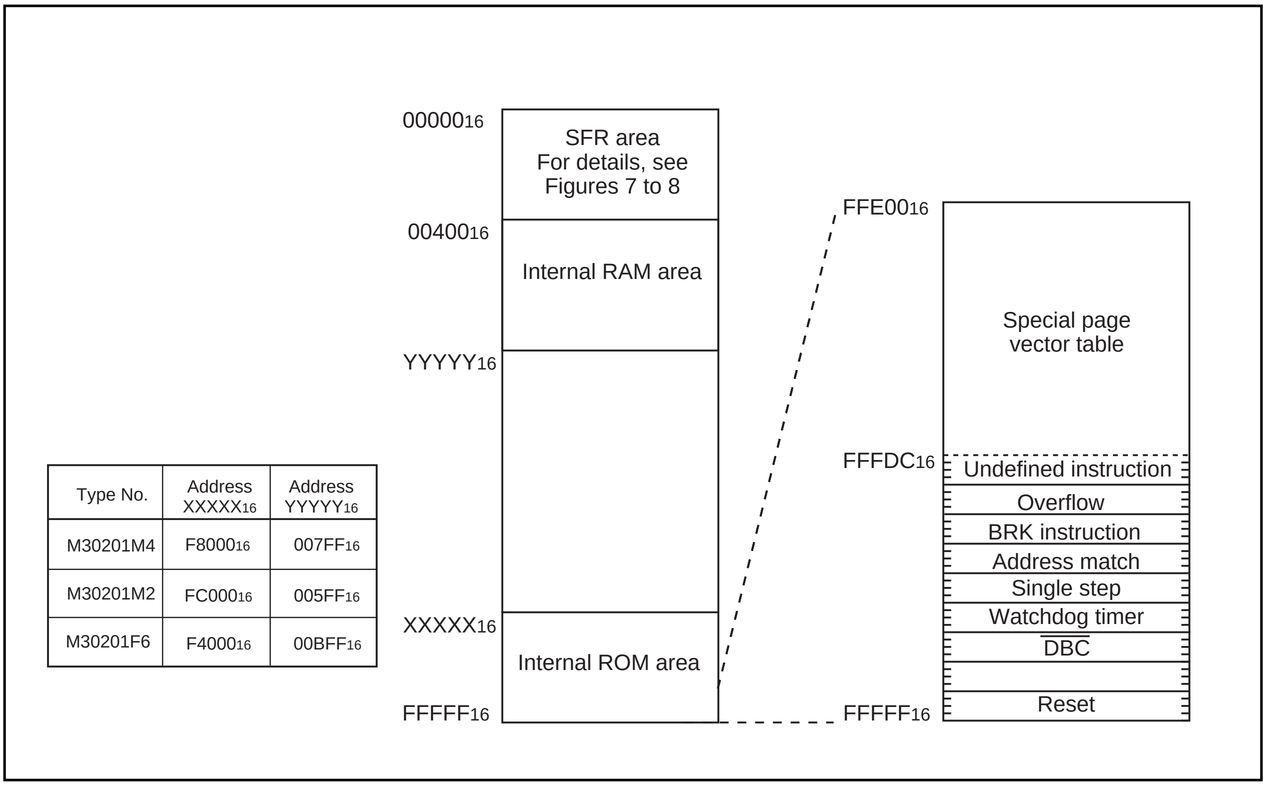
Memory

Figure 6 is a memory map of the M30201. The address space extends the 1M bytes from address 00000_{16} to $FFFFF_{16}$. From $FFFFF_{16}$ down is ROM. For example, in the M30201M4-XXXFP, there is 32K bytes of internal ROM from $F8000_{16}$ to $FFFFF_{16}$. The vector table for fixed interrupts such as the reset are mapped to $FFFDC_{16}$ to $FFFFF_{16}$. The starting address of the interrupt routine is stored here. The address of the vector table for timer interrupts, etc., can be set as desired using the internal register (INTB). See the section on interrupts for details.

From 00400_{16} up is RAM. For example, in the M30201M4-XXXFP, there is 1K byte of internal RAM from 00400_{16} to $007FF_{16}$. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.

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.



Type No.	Address $XXXXX_{16}$	Address $YYYYY_{16}$
M30201M4	$F8000_{16}$	$007FF_{16}$
M30201M2	$FC000_{16}$	$005FF_{16}$
M30201F6	$F4000_{16}$	$00BFF_{16}$

Figure 6. Memory map

0000$_{16}$	
0001$_{16}$	
0002$_{16}$	
0003$_{16}$	
0004$_{16}$	Processor mode register 0 (PM0)
0005$_{16}$	Processor mode register 1(PM1)
0006$_{16}$	System clock control register 0 (CM0)
0007$_{16}$	System clock control register 1 (CM1)
0008$_{16}$	
0009$_{16}$	Address match interrupt enable register (AIER)
000A$_{16}$	Protect register (PRCR)
000B$_{16}$	
000C$_{16}$	
000D$_{16}$	
000E$_{16}$	Watchdog timer start register (WDTS)
000F$_{16}$	Watchdog timer control register (WDC)
0010$_{16}$	Address match interrupt register 0 (RMAD0)
0011$_{16}$	
0012$_{16}$	
0013$_{16}$	
0014$_{16}$	Address match interrupt register 1 (RMAD1)
0015$_{16}$	
0016$_{16}$	
0017$_{16}$	
0018$_{16}$	
0019$_{16}$	
001A$_{16}$	
001B$_{16}$	
001C$_{16}$	
001D$_{16}$	
001E$_{16}$	
001F$_{16}$	
0020$_{16}$	
0021$_{16}$	
0022$_{16}$	
0023$_{16}$	
0024$_{16}$	
0025$_{16}$	
0026$_{16}$	
0027$_{16}$	
0028$_{16}$	
0029$_{16}$	
002A$_{16}$	
002B$_{16}$	
002C$_{16}$	
002D$_{16}$	
002E$_{16}$	
002F$_{16}$	
0030$_{16}$	
0031$_{16}$	
0032$_{16}$	
0033$_{16}$	
0034$_{16}$	
0035$_{16}$	
0036$_{16}$	
0037$_{16}$	
0038$_{16}$	
0039$_{16}$	
003A$_{16}$	
003B$_{16}$	
003C$_{16}$	
003D$_{16}$	
003E$_{16}$	
003F$_{16}$	

0040$_{16}$	
0041$_{16}$	
0042$_{16}$	
0043$_{16}$	
0044$_{16}$	
0045$_{16}$	
0046$_{16}$	
0047$_{16}$	
0048$_{16}$	
0049$_{16}$	
004A$_{16}$	
004B$_{16}$	
004C$_{16}$	
004D$_{16}$	Key input interrupt control register (KUPIC)
004E$_{16}$	A-D conversion interrupt control register (ADIC)
004F$_{16}$	
0050$_{16}$	
0051$_{16}$	UART0 transmit interrupt control register (S0TIC)
0052$_{16}$	UART0 receive interrupt control register (S0RIC)
0053$_{16}$	UART1 transmit interrupt control register (S1TIC)
0054$_{16}$	UART1 receive interrupt control register (S1RIC)
0055$_{16}$	Timer A0 interrupt control register (TA0IC)
0056$_{16}$	Timer X0 interrupt control register (TX0IC)
0057$_{16}$	Timer X1 interrupt control register (TX1IC)
0058$_{16}$	Timer X2 interrupt control register (TX2IC)
0059$_{16}$	
005A$_{16}$	Timer B0 interrupt control register (TB0IC)
005B$_{16}$	Timer B1 interrupt control register (TB1IC)
005C$_{16}$	
005D$_{16}$	INT0 interrupt control register (INT0IC)
005E$_{16}$	INT1 interrupt control register (INT1IC)
005F$_{16}$	

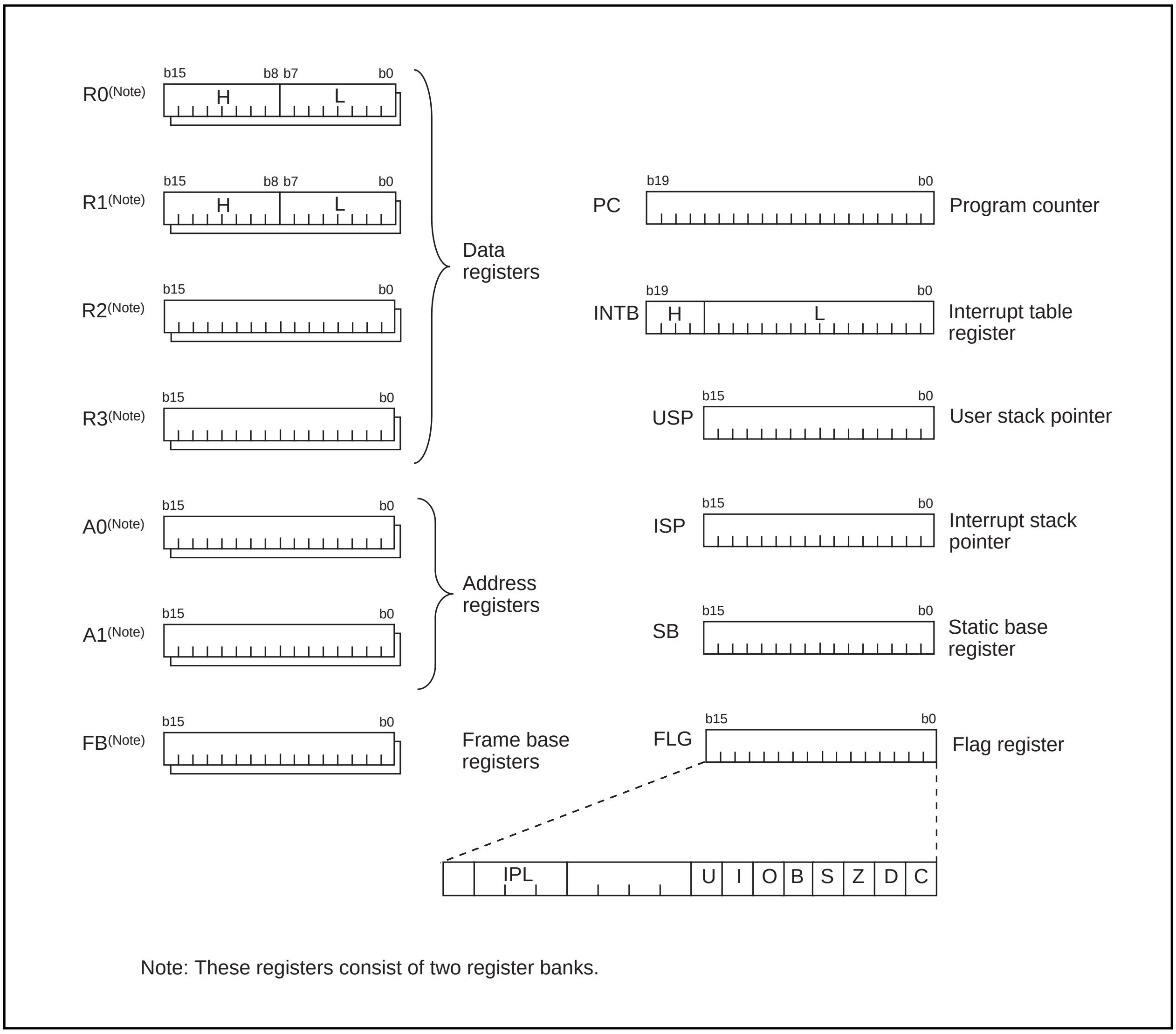
Figure 7. Location of peripheral unit control registers (1)

0380_{16}	Count start flag (TABSR)	$03C0_{16}$	A-D register 0 (AD0)
0381_{16}	Clock prescaler reset flag (CPSRF)	$03C1_{16}$	
0382_{16}	One-shot start flag (ONSF)	$03C2_{16}$	A-D register 1 (AD1)
0383_{16}	Trigger select register (TRGSR)	$03C3_{16}$	
0384_{16}	Up-down flag (UDF)	$03C4_{16}$	A-D register 2 (AD2)
0385_{16}		$03C5_{16}$	
0386_{16}	Timer A0 (TA0)	$03C6_{16}$	A-D register 3 (AD3)
0387_{16}		$03C7_{16}$	
0388_{16}	Timer X0 (TX0)	$03C8_{16}$	A-D register 4 (AD4)
0389_{16}		$03C9_{16}$	
$038A_{16}$	Timer X1 (TX1)	$03CA_{16}$	A-D register 5 (AD5)
$038B_{16}$		$03CB_{16}$	
$038C_{16}$	Timer X2 (TX2)	$03CC_{16}$	A-D register 6 (AD6)
$038D_{16}$		$03CD_{16}$	
$038E_{16}$	Clock divided counter (CDC)	$03CE_{16}$	A-D register 7 (AD7)
$038F_{16}$		$03CF_{16}$	
0390_{16}	Timer B0 (TB0)	$03D0_{16}$	
0391_{16}		$03D1_{16}$	
0392_{16}	Timer B1 (TB1)	$03D2_{16}$	
0393_{16}		$03D3_{16}$	
0394_{16}		$03D4_{16}$	A-D control register 2 (ADCON2)
0395_{16}		$03D5_{16}$	
0396_{16}	Timer A0 mode register (TA0MR)	$03D6_{16}$	A-D control register 0 (ADCON0)
0397_{16}	Timer X0 mode register (TX0MR)	$03D7_{16}$	A-D control register 1 (ADCON1)
0398_{16}	Timer X1 mode register (TX1MR)	$03D8_{16}$	
0399_{16}	Timer X2 mode register (TX2MR)	$03D9_{16}$	
$039A_{16}$		$03DA_{16}$	
$039B_{16}$	Timer B0 mode register (TB0MR)	$03DB_{16}$	
$039C_{16}$	Timer B1 mode register (TB1MR)	$03DC_{16}$	
$039D_{16}$		$03DD_{16}$	
$039E_{16}$		$03DE_{16}$	
$039F_{16}$		$03DF_{16}$	
$03A0_{16}$	UART0 transmit/receive mode register (U0MR)	$03E0_{16}$	Port P0 (P0)
$03A1_{16}$	UART0 bit rate generator (U0BRG)	$03E1_{16}$	Port P1 (P1)
$03A2_{16}$	UART0 transmit buffer register (U0TB)	$03E2_{16}$	Port P0 direction register (PD0)
$03A3_{16}$		$03E3_{16}$	Port P1 direction register (PD1)
$03A4_{16}$	UART0 transmit/receive control register 0 (U0C0)	$03E4_{16}$	Port P2 (P2) (Reserved)
$03A5_{16}$	UART0 transmit/receive control register 1 (U0C1)	$03E5_{16}$	Port P3 (P3)
$03A6_{16}$	UART0 receive buffer register (U0RB)	$03E6_{16}$	Port P2 direction register (PD2) (Reserved)
$03A7_{16}$		$03E7_{16}$	Port P3 direction register (PD3)
$03A8_{16}$	UART1 transmit/receive mode register (U1MR)	$03E8_{16}$	Port P4 (P4)
$03A9_{16}$	UART1 bit rate generator (U1BRG)	$03E9_{16}$	Port P5 (P5)
$03AA_{16}$	UART1 transmit buffer register (U1TB)	$03EA_{16}$	Port P4 direction register (PD4)
$03AB_{16}$		$03EB_{16}$	Port P5 direction register (PD5)
$03AC_{16}$	UART1 transmit/receive control register 0 (U1C0)	$03EC_{16}$	Port P6 (P6)
$03AD_{16}$	UART1 transmit/receive control register 1 (U1C1)	$03ED_{16}$	Port P7 (P7)
$03AE_{16}$	UART1 receive buffer register (U1RB)	$03EE_{16}$	Port P6 direction register (PD6)
$03AF_{16}$		$03EF_{16}$	Port P7 direction register (PD7)
$03B0_{16}$	UART transmit/receive control register 2 (UCON)	$03F0_{16}$	
$03B1_{16}$		$03F1_{16}$	
$03B2_{16}$		$03F2_{16}$	
$03B3_{16}$		$03F3_{16}$	
$03B4_{16}$		$03F4_{16}$	
$03B5_{16}$		$03F5_{16}$	
$03B6_{16}$		$03F6_{16}$	
$03B7_{16}$		$03F7_{16}$	
$03B8_{16}$		$03F8_{16}$	
$03B9_{16}$		$03F9_{16}$	
$03BA_{16}$		$03FA_{16}$	
$03BB_{16}$		$03FB_{16}$	
$03BC_{16}$		$03FC_{16}$	Pull-up control register 0 (PUR0)
$03BD_{16}$		$03FD_{16}$	Pull-up control register 1 (PUR1)
$03BE_{16}$		$03FE_{16}$	Port P1 drive control register (DRR)
$03BF_{16}$		$03FF_{16}$	

Figure 8. Location of peripheral unit control registers (2)

Central Processing Unit (CPU)

The CPU has a total of 13 registers shown in Figure 9. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these have two register banks.



Note: These registers consist of two register banks.

Figure 9. 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 10 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.

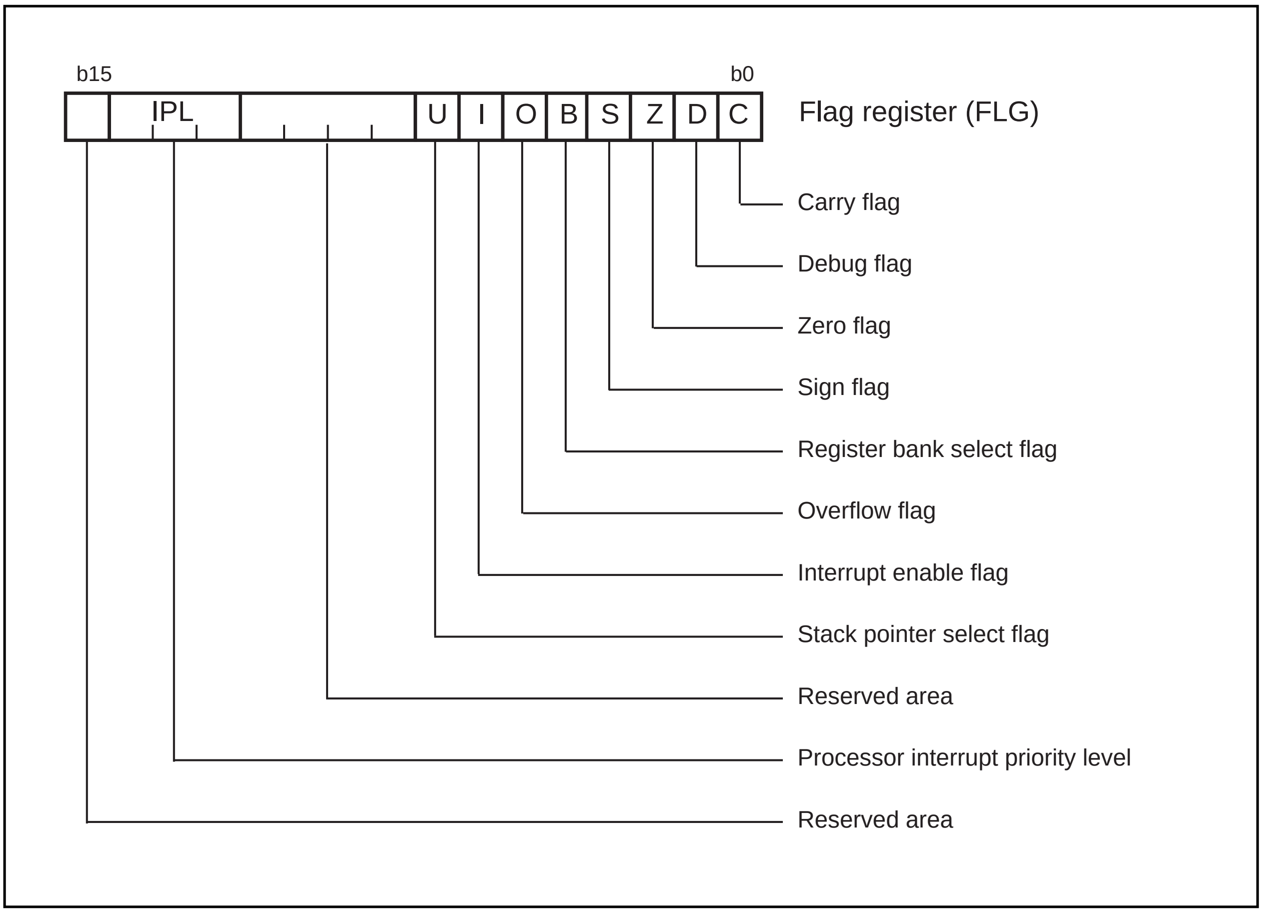


Figure 10. Flag register (FLG)

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” (0.2VCC 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 11 shows the example reset circuit. Figure 12 shows the reset sequence.

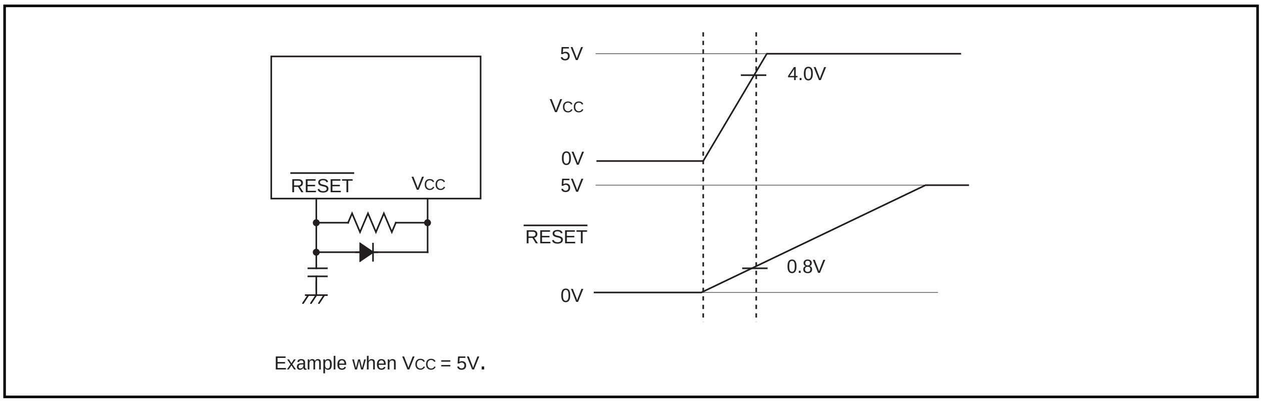


Figure 11. Example reset circuit

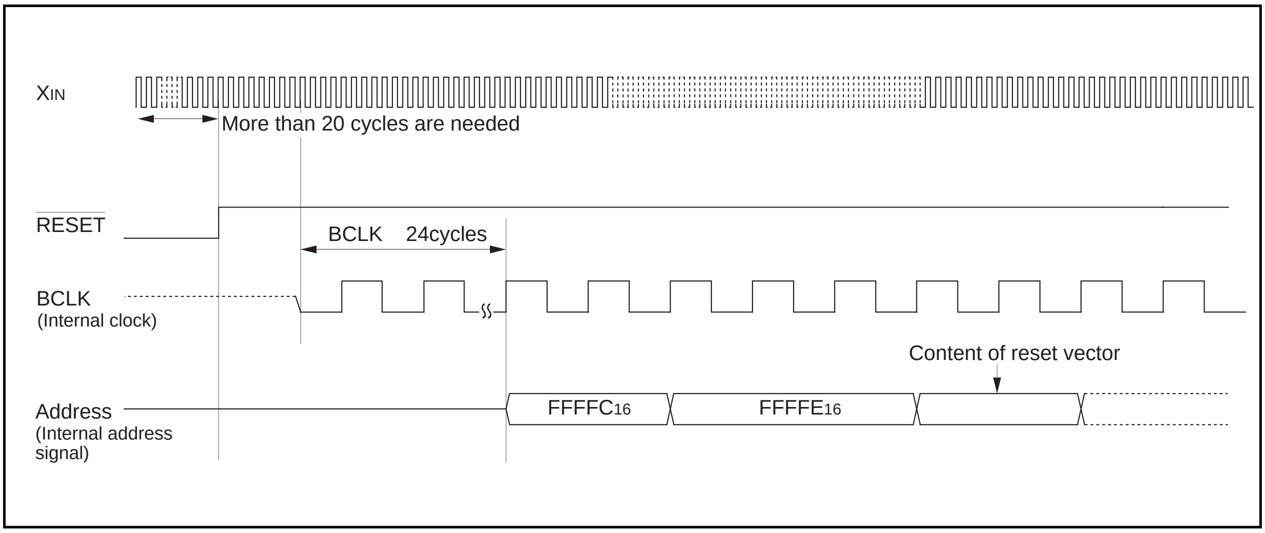


Figure 12. Reset sequence

(1)	Processor mode register 0	(0004$_{16}$)···	x x x x 0 0 0 0
(2)	Processor mode register 1	(0005$_{16}$)···	0 x x x x x 0 0
(3)	System clock control register 0	(0006$_{16}$)···	0 1 0 0 1 0 0 0
(4)	System clock control register 1	(0007$_{16}$)···	0 0 1 0 0 0 0 0
(5)	Address match interrupt enable register	(0009$_{16}$)···	x x x x x x 0 0
(6)	Protect register	(000A$_{16}$)···	x x x x x 0 0 0
(7)	Watchdog timer control register	(000F$_{16}$)···	0 0 0 ? ? ? ? ?
(8)	Address match interrupt register 0	(0010$_{16}$)···	00$_{16}$
		(0011$_{16}$)···	00$_{16}$
		(0012$_{16}$)···	x x x x 0 0 0 0
(9)	Address match interrupt register 1	(0014$_{16}$)···	00$_{16}$
		(0015$_{16}$)···	00$_{16}$
		(0016$_{16}$)···	x x x x 0 0 0 0
(10)	Key input interrupt control register	(004D$_{16}$)···	x x x x ? 0 0 0
(11)	A-D conversion interrupt control register	(004E$_{16}$)···	x x x x ? 0 0 0
(12)	UART0 transmit interrupt control register	(0051$_{16}$)···	x x x x ? 0 0 0
(13)	UART0 receive interrupt control register	(0052$_{16}$)···	x x x x ? 0 0 0
(14)	UART1 transmit interrupt control register	(0053$_{16}$)···	x x x x ? 0 0 0
(15)	UART1 receive interrupt control register	(0054$_{16}$)···	x x x x ? 0 0 0
(16)	Timer A0 interrupt control register	(0055$_{16}$)···	x x x x ? 0 0 0
(17)	Timer X0 interrupt control register	(0056$_{16}$)···	x x x x ? 0 0 0
(18)	Timer X1 interrupt control register	(0057$_{16}$)···	x x x x ? 0 0 0
(19)	Timer X2 interrupt control register	(0058$_{16}$)···	x x x x ? 0 0 0
(20)	Timer B0 interrupt control register	(005A$_{16}$)···	x x x x ? 0 0 0
(21)	Timer B1 interrupt control register	(005B$_{16}$)···	x x x x ? 0 0 0
(22)	INT0 interrupt control register	(005D$_{16}$)···	x x 0 0 ? 0 0 0
(23)	INT1 interrupt control register	(005E$_{16}$)···	x x 0 0 ? 0 0 0
(24)	Count start flag	(0380$_{16}$)···	0 0 0 x 0 0 0 0
(25)	Clock prescaler reset flag	(0381$_{16}$)···	0 x x x x x x x
(26)	One-shot start flag	(0382$_{16}$)···	x x x x 0 0 0 0
(27)	Trigger select flag	(0383$_{16}$)···	00$_{16}$
(28)	Up-down flag	(0384$_{16}$)···	x x x 0 x x x 0
(29)	Timer A0 mode register	(0396$_{16}$)···	00$_{16}$
(30)	Timer X0 mode register	(0397$_{16}$)···	00$_{16}$
(31)	Timer X1 mode register	(0398$_{16}$)···	00$_{16}$
(32)	Timer X2 mode register	(0399$_{16}$)···	00$_{16}$
(33)	Timer B0 mode register	(039B$_{16}$)··	0 0 ? x 0 0 0 0
(34)	Timer B1 mode register	(039C$_{16}$)··	0 0 ? x 0 0 0 0
(35)	UART0 transmit/receive mode register	(03A0$_{16}$)··	00$_{16}$
(36)	UART0 transmit/receive control register 0	(03A4$_{16}$)··	0 0 0 0 1 0 0 0
(37)	UART0 transmit/receive control register 1	(03A5$_{16}$)··	0 0 0 0 0 0 1 0
(38)	UART1 transmit/receive mode register	(03A8$_{16}$)··	00$_{16}$
(39)	UART1 transmit/receive control register 0	(03AC$_{16}$)··	0 0 0 0 1 0 0 0
(40)	UART1 transmit/receive control register 1	(03AD$_{16}$)··	0 0 0 0 0 0 1 0
(41)	UART transmit/receive control register 2	(03B0$_{16}$)··	x 0 0 0 0 0 0 0
(42)	A-D control register 2	(03D4$_{16}$)··	x x x x 0 0 0 0
(43)	A-D control register 0	(03D6$_{16}$)··	0 0 0 0 0 ? ? ?
(44)	A-D control register 1	(03D7$_{16}$)··	00$_{16}$
(45)	Port P0 direction register	(03E2$_{16}$)···	00$_{16}$
(46)	Port P1 direction register	(03E3$_{16}$)··	00$_{16}$
(47)	Port P2 direction register	(03E6$_{16}$)···	x 0 0 0 0 0 0 0
(48)	Port P3 direction register	(03E7$_{16}$)···	x x 0 0 0 0 0 0
(49)	Port P4 direction register	(03EA$_{16}$)···	x x 0 0 0 0 0 0
(50)	Port P5 direction register	(03EB$_{16}$)···	x x x 0 0 0 0 0
(51)	Port P6 direction register	(03EE$_{16}$)··	00$_{16}$
(52)	Port P7 direction register	(03EF$_{16}$)···	x x x x x x 0 0
(53)	Pull-up control register 0	(03FC$_{16}$)··	00$_{16}$
(54)	Pull-up control register 1	(03FD$_{16}$)··	00$_{16}$
(55)	Port P1 drive capacity control register	(03FE$_{16}$)··	00$_{16}$
(56)	Data registers (R0/R1/R2/R3)		0000$_{16}$
(57)	Address registers (A0/A1)		0000$_{16}$
(58)	Frame base register (FB)		0000$_{16}$
(59)	Interrupt table register (INTB)		00000$_{16}$
(60)	User stack pointer (USP)		0000$_{16}$
(61)	Interrupt stack pointer (ISP)		0000$_{16}$
(62)	Static base register (SB)		0000$_{16}$
(63)	Flag register (FLG)		0000$_{16}$

x : Nothing is mapped to this bit
? : Undefined

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

Figure 13. Device's internal status after a reset is cleared

Software Reset

Writing “1” to bit 3 of the processor mode register 0 (address 0004_{16}) applies a (software) reset to the microcomputer. A software reset has almost the same effect as a hardware reset. The contents of internal RAM are preserved.

Figure 14 shows the processor mode register 0 and 1.

Processor mode register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
☒	☒	☒	☒	0	0	0	0

Symbol	Address	When reset
PM0	0004_{16}	$XXXX0000_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
PM03	Software reset bit	The device is reset when this bit is set to “1”. The value of this bit is “0” when read.	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

Note: Set bit 1 of the protect register (address $000A_{16}$) to “1” when writing new values to this register.

Processor mode register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
0	☒	☒	☒	☒	☒	0	0

Symbol	Address	When reset
PM1	0005_{16}	$00XXXXX0_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
PM17	Wait bit	0 : No wait state 1 : Wait state inserted	○	○

Note: Set bit 1 of the protect register (address $000A_{16}$) to “1” when writing new values to this register.

Figure 14. Processor mode register 0 and 1.

Software wait

The wait bit (bit 7) of the processor mode register 1 (address 0005_{16})(note) allows you to insert software wait states for the internal ROM/RAM areas. If this bit is 0, the bus cycle is executed in one BCLK (internal clock) period; if the bit is 1, the bus cycle is executed in two BCLK periods. This bit is cleared to 0 after a reset.

The SFR area is unaffected by this control bit; it is always accessed in two BCLK periods.

Table 2 shows the relationship between software wait states and bus cycles.

Note: Before attempting to change the contents of the processor mode register 1, set bit 1 of the protect register (address $000A_{16}$) to “1”.

Table 2. 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

Clock Generating Circuit

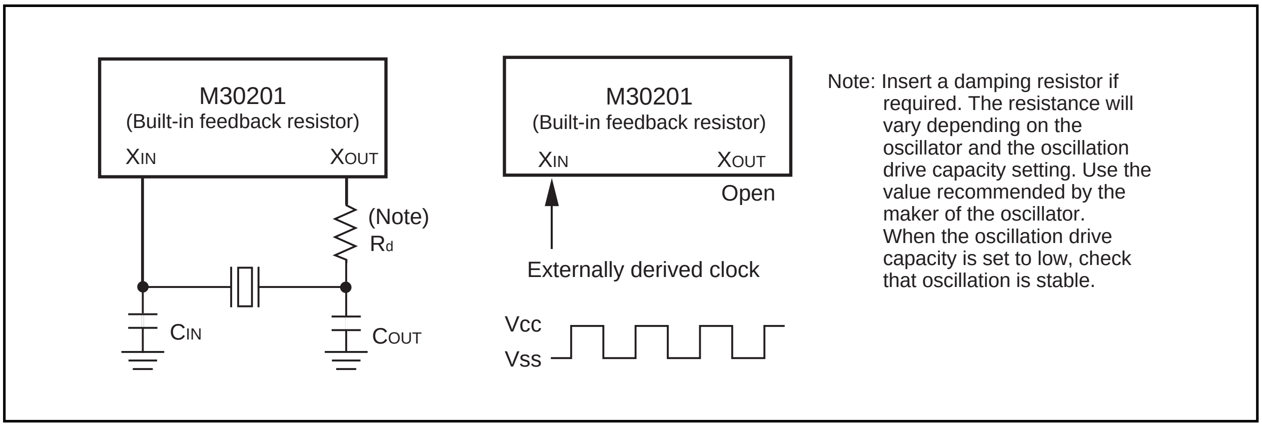
The clock generating circuit contains two oscillator circuits that supply the operating clock sources to the CPU and internal peripheral units.

Table 3. Main clock and sub clock generating circuits

	Main clock generating circuit	Sub clock generating circuit
Use of clock	• CPU's operating clock source • Internal peripheral units' operating clock source	• CPU's operating clock source • Timer A/B/X's count clock source
Usable oscillator	Ceramic or crystal oscillator	Crystal oscillator
Pins to connect oscillator	X_{IN}, X_{OUT}	X_{CIN}, X_{COUT}
Oscillation stop/restart function	Available	Available
Oscillator status immediately after reset	Oscillating	Stopped
Other	Externally derived clock can be input	

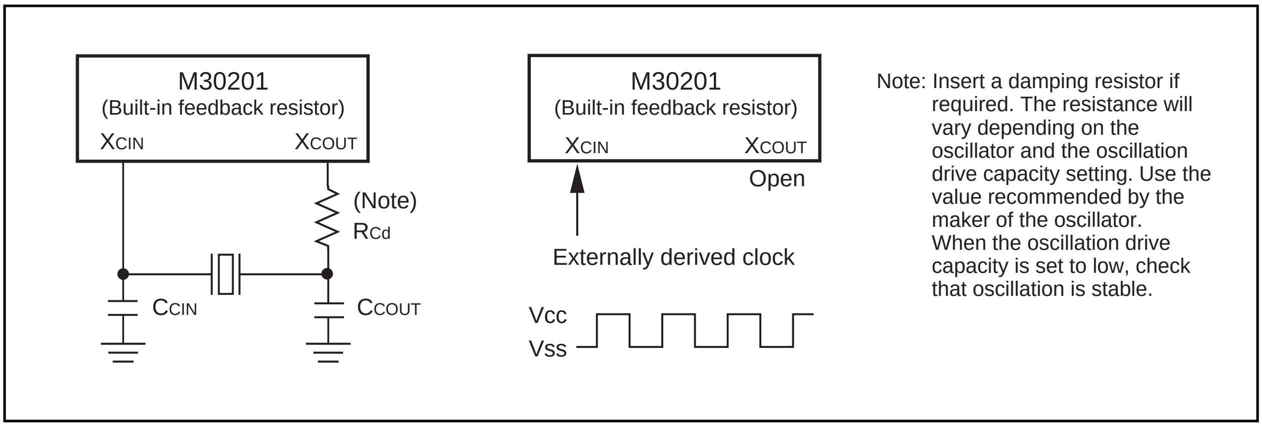
Example of oscillator circuit

Figure 15 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 16 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 15 and 16 vary with each oscillator used. Use the values recommended by the manufacturer of your oscillator.



Note: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator. When the oscillation drive capacity is set to low, check that oscillation is stable.

Figure 15. Examples of main clock



Note: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator. When the oscillation drive capacity is set to low, check that oscillation is stable.

Figure 16. Examples of sub clock

Clock Control

Figure 17 shows the block diagram of the clock generating circuit.

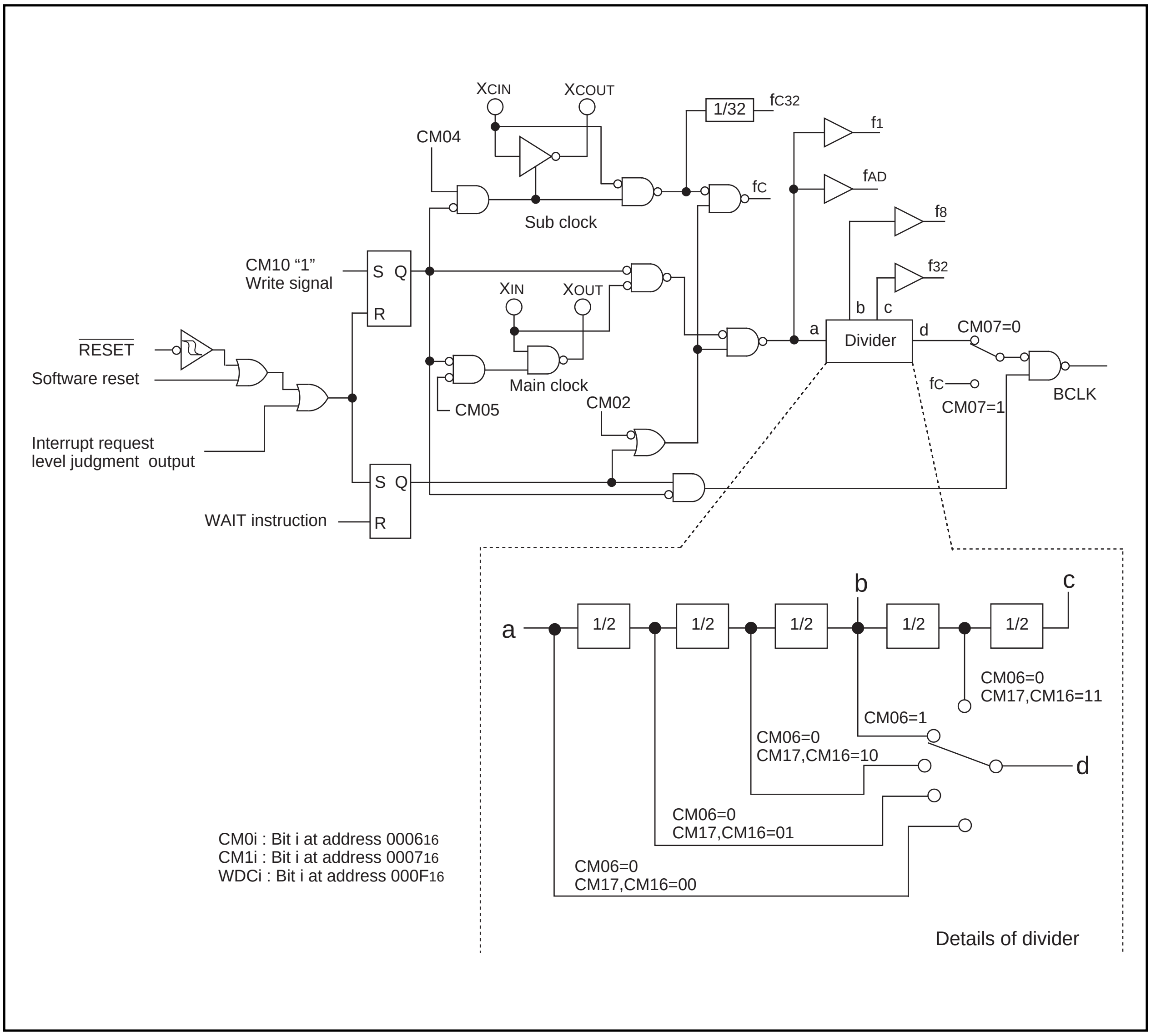


Figure 17. Clock generating circuit

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 BCLK. The clock can be stopped using the main clock stop bit (bit 5 at address 0006_{16}). Stopping the clock reduces the power dissipation.

After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the XOUT pin can be reduced using the XIN-XOUT drive capacity select bit (bit 5 at address 0007_{16}). Reducing the drive capacity of the XOUT pin reduces the power dissipation. This bit defaults to “1” when shifting to stop mode and after a reset.

(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_{16}), the sub clock can be selected as BCLK by using the system clock select bit (bit 7 at address 0006_{16}). 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 XCOUT pin can be reduced using the XCIN-XCOUT drive capacity select bit (bit 3 at address 0006_{16}). Reducing the drive capacity of the XCOUT pin reduces the power dissipation. This bit changes to “1” when shifting to stop mode and at a reset.

(3) BCLK

BCLK is the clock that drives the CPU, and is either the main clock or fc is derived by dividing the main clock by 2, 4, 8, or 16. BCLK is derived by dividing the main clock by 8 after a reset.

When shifting to stop mode, the main clock division select bit (bit 6 at 0006_{16}) is set to “1”.

(4) Peripheral function clock

- **f_1, f_8, f_{32}**

  The clock for the peripheral devices is derived from the main clock or by dividing it by 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_{16}) to “1” and then executing a WAIT instruction.

- **f_{AD}**

  This clock has the same frequency as the main clock and is used for A-D conversion.

(5) f_{C32}

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

(6) f_C

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

Figure 18 shows the system clock control registers 0 and 1.

System clock control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
CM0	0006_{16}	48_{16}

Bit symbol	Bit name	Function	R	W
CM00	Clock output function select bit	b1 b0 0 0 : I/O port $P5_4$ 0 1 : f_C output 1 0 : f_8 output 1 1 : Clock divide counter output	○	○
CM01			○	○
CM02	WAIT peripheral function clock stop bit	0 : Do not stop f_1, f_8, f_{32} in wait mode 1 : Stop f_1, f_8, f_{32} in wait mode	○	○
CM03	X_{CIN}-X_{COUT} drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	○	○
CM04	Port X_C select bit	0 : I/O port 1 : X_{CIN}-X_{COUT} generation	○	○
CM05	Main clock (X_{IN}-X_{OUT}) stop bit (Note 3,4,5)	0 : On 1 : Off	○	○
CM06	Main clock division select bit 0 (Note 2)	0 : CM16 and CM17 valid 1 : Division by 8 mode	○	○
CM07	System clock select bit (Note 6)	0 : X_{IN}, X_{OUT} 1 : X_{CIN}, X_{COUT}	○	○

Note 1: Set bit 0 of the protect register (address $000A_{16}$) to “1” before writing to this register.
Note 2: Changes to “1” when shifting to stop mode.
Note 3: This bit is used to stop the main clock when placing the device in a low-power mode. If you want to operate with X_{IN} after exiting from the stop mode, set this bit to “0”. When operating with a self-excited oscillator, set the system clock select bit (CM07) to “1” before setting this bit to “1”.
Note 4: When operating with an external clock, the device is placed in a mode in which only the clock oscillation buffer is stopped and clock input is accepted.
Note 5: When this bit is “1”, X_{OUT} becomes “H” level. As internal feedback resister is still ON, X_{IN} is pulled-up to X_{OUT} (“H” level) through the feedback resister.
Note 6: Before setting this bit to “1”, be sure to set the port Xc select bit (CM04) to “1”. You cannot set both bits simultaneously.

System clock control register 1 (Note 1)

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

Symbol	Address	When reset
CM1	0007_{16}	20_{16}

Bit symbol	Bit name	Function	R	W
CM10	All clock stop control bit (Note 4)	0 : Clock on 1 : All clocks off (stop mode)	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○
CM15	X_{IN}-X_{OUT} drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	○	○
CM16	Main clock division select bit 1 (Note 3)	b7 b6 0 0 : No division mode 0 1 : Division by 2 mode 1 0 : Division by 4 mode 1 1 : Division by 16 mode	○	○
CM17				

Note 1: Set bit 0 of the protect register (address $000A_{16}$) to “1” before writing to this register.
Note 2: Changes to “1” when shifting to stop mode.
Note 3: Can be selected when bit 6 of the system clock control register 0 (address 000616) is “0”. If “1”, division mode is fixed at 8.
Note 4: When this bit is “1”, X_{OUT} becomes “H” level and internal feedback resister is invalid.

Figure 18. Clock control registers 0 and 1

Clock Output

The clock output function select bit allows you to choose the clock from f_8, fc, or a divide-by-n clock that is output from the P5$_4$/CK$_{OUT}$ pin. The clock divide counter is an 8-bit counter whose count source is f_{32}, and its divide ratio can be set in the range of 00$_{16}$ to FF$_{16}$. Figure 19 shows a block diagram of clock output.

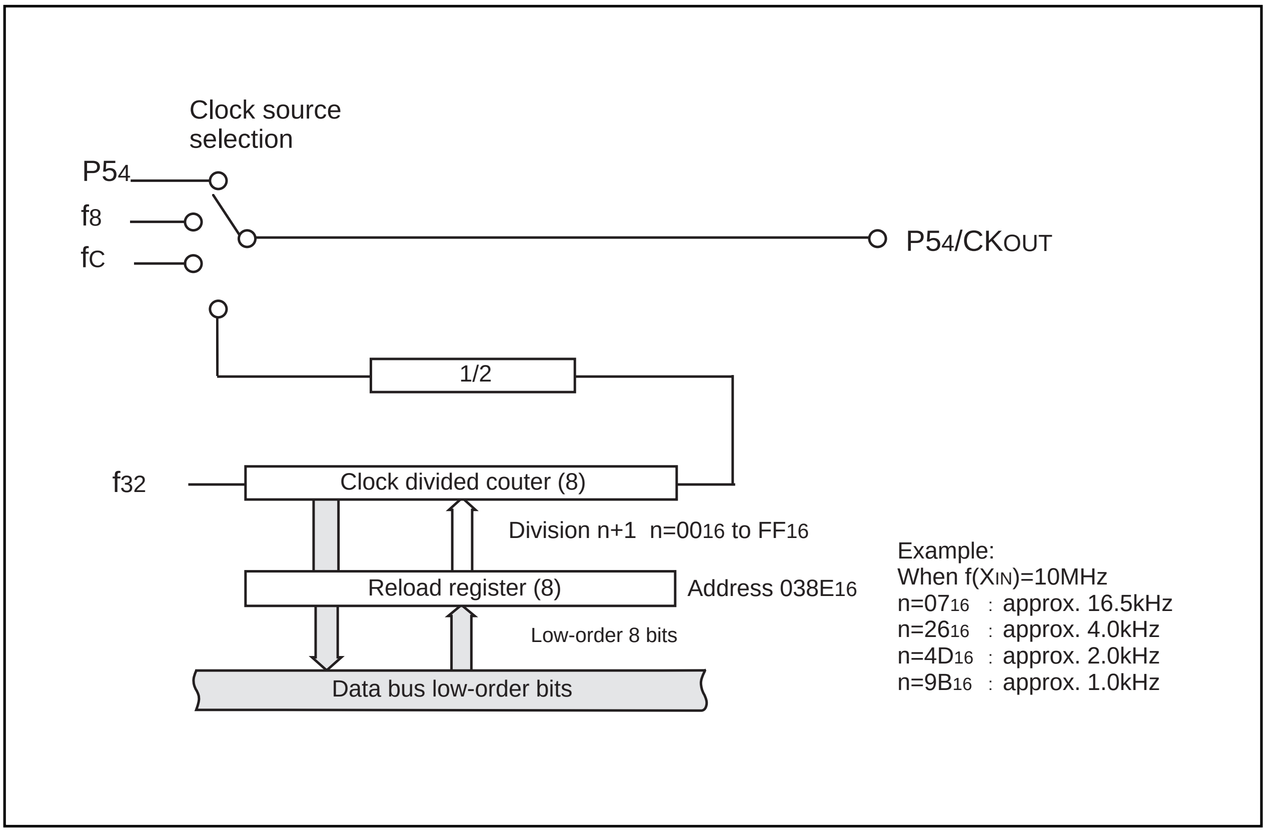


Figure 19. Block diagram of clock output

Stop Mode

Writing “1” to the all-clock stop control bit (bit 0 at address 0007_{16}) stops all oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that V_{CC} remains above 2V.

Because the oscillation of BCLK, f_1 to f_{32}, fc, fc_{32}, and f_{AD} stops in stop mode, peripheral functions such as the A-D converter and watchdog timer do not function. However, timer A, timer B and timer X operate provided that the event counter mode is set to an external pulse, and UART0 functions provided an external clock is selected. Table 4 shows the status of the ports in stop mode.

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

When shifting to stop mode, the main clock division select bit 0 (bit 6 at 0006_{16}) is set to “1”.

Table 4. Port status during stop mode

Pin		States
Port		Retains status before stop mode
CLK_{OUT}	When f_C selected	“H”
	When f_8, clock devided counter output 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 5 shows the status of the ports in wait mode.

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

Table 5. Port status during wait mode

Pin		States
Port		Retains status before wait mode
CLK_{OUT}	When f_C selected	Does not stop
	When f_8, clock devided counter output selected	Does not stop when the WAIT peripheral function clock stop bit is “0”. When the WAIT peripheral function clock stop bit is “1”,the status immediately prior to entering wait mode is maintained.

Status Transition of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. Table 6 shows the operating modes corresponding to the settings of system clock control registers 0 and 1.

After a reset, operation defaults to division by 8 mode. When shifting to stop mode, the main clock division select bit 0 (bit 6 at address 0006_{16}) is set to “1”. 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. Note that oscillation of the main clock must have stabilized before transferring from this mode to another mode.

(4) Division by 16 mode

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

(5) No-division mode

The main clock is used as BCLK.

(6) Low-speed mode

fc is used as 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.

Table 6. 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 Saving

There are three power save modes.

(1) Normal operating mode

- **High-speed mode**
  In this mode, one main clock cycle forms BCLK. The CPU operates on the BCLK. The peripheral functions operate on the clocks specified for each respective function.
- **Medium-speed mode**
  In this mode, the main clock is divided into 2, 4, 8, or 16 to form BCLK. The CPU operates on the BCLK. The peripheral functions operated on the clocks specified for each respective function.
- **Low-speed mode**
  In this mode, fc forms BCLK. The CPU operates on the fc clock. fc is the clock supplied by the subclock. The peripheral functions operate on the clocks specified for each respective function.
- **Low power-dissipation mode**
  This mode is selected when the main clock is stopped from low-speed mode. The CPU operates on the fc clock. fc is the clock supplied by the subclock. Only the peripheral functions for which the subclock was selected as the count source continue to run.

(2) Wait mode

CPU operation is halted in this mode. The oscillator continues to run.

(3) Stop mode

All oscillators stop in this mode. The CPU and internal peripheral functions all stop. Of all 3 power saving modes, power savings are greatest in this mode.

Figure 20 shows the transition between each of the three modes, (1), (2), and (3).

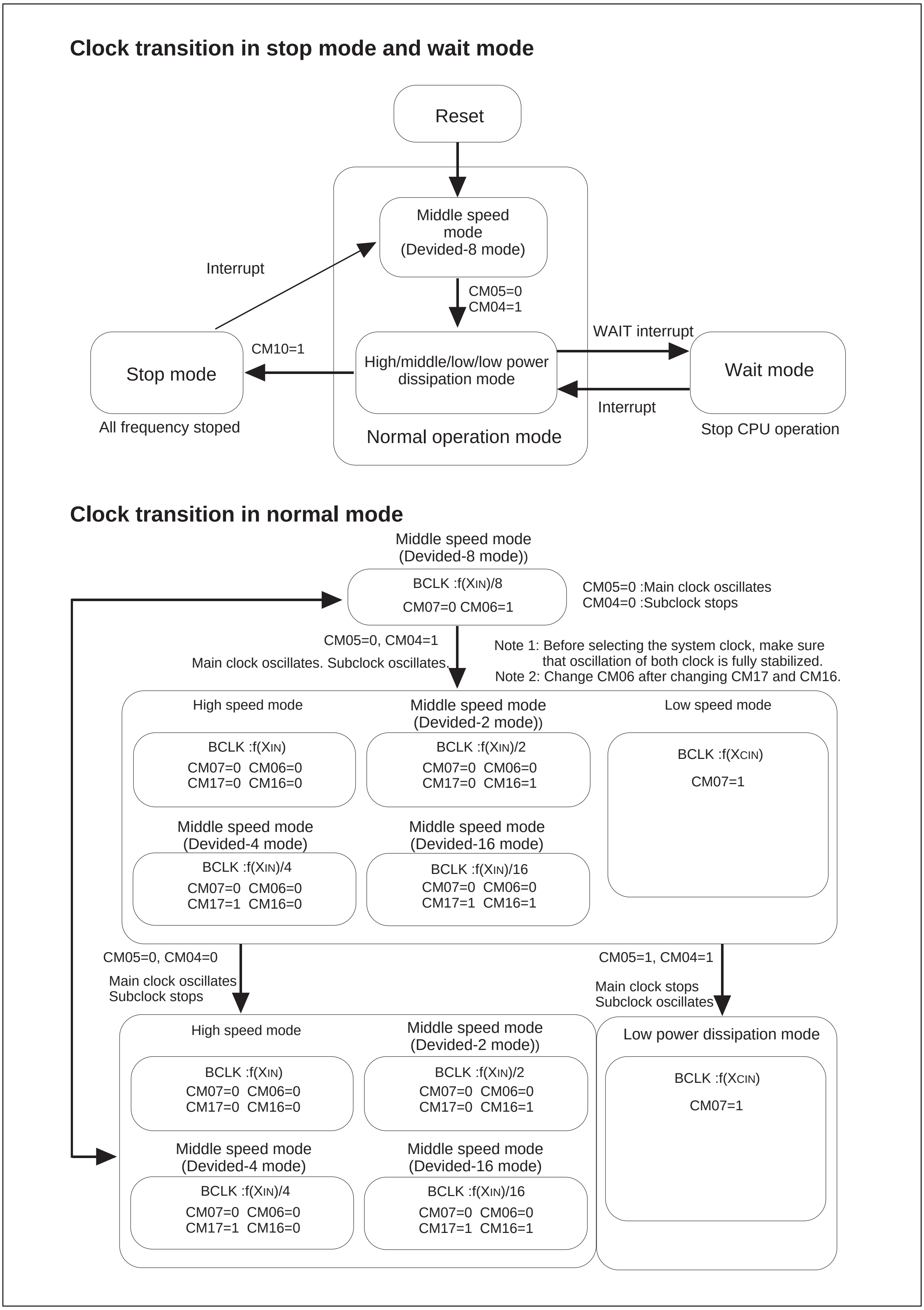
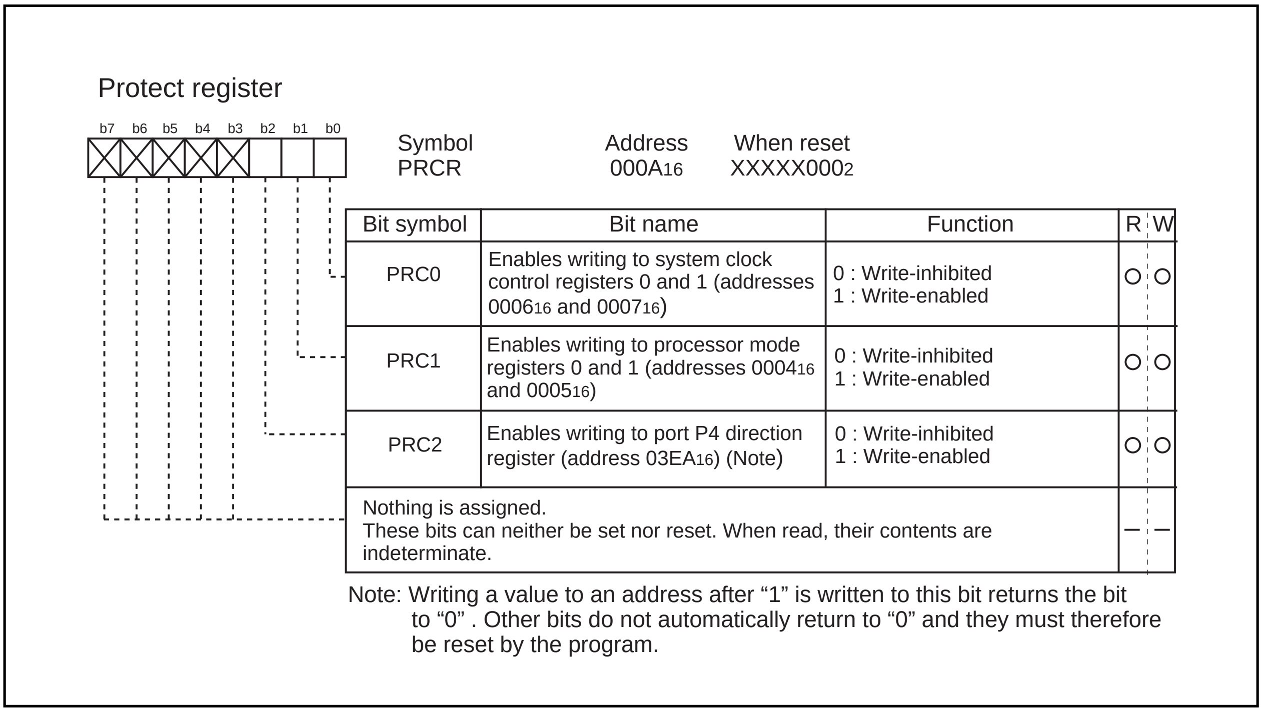


Figure 20. Clock transition

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 21 shows the protect register. The values in the processor mode register 0 (address 0004_{16}), processor mode register 1 (address 0005_{16}), system clock control register 0 (address 0006_{16}), system clock control register 1 (address 0007_{16}) and port P4 direction register (address $03EA_{16}$) can only be changed when the respective bit in the protect register is set to “1”. Therefore, important outputs can be allocated to port P4.

If, after “1” (write-enabled) has been written to the port P4 direction register write-enable bit (bit 2 at address $000A_{16}$), a value is written to any address, the bit automatically reverts to “0” (write-inhibited). However, the system clock control registers 0 and 1 write-enable bit (bit 0 at $000A_{16}$) and processor mode register 0 and 1 write-enable bit (bit 1 at $000A_{16}$) 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”.



Protect register

Symbol PRCR   Address $000A_{16}$ When reset $XXXXX000_2$

Bit symbol	Bit name	Function	R	W
PRC0	Enables writing to system clock control registers 0 and 1 (addresses 0006_{16} and 0007_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC1	Enables writing to processor mode registers 0 and 1 (addresses 0004_{16} and 0005_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC2	Enables writing to port P4 direction register (address $03EA_{16}$) (Note)	0 : Write-inhibited 1 : Write-enabled	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—

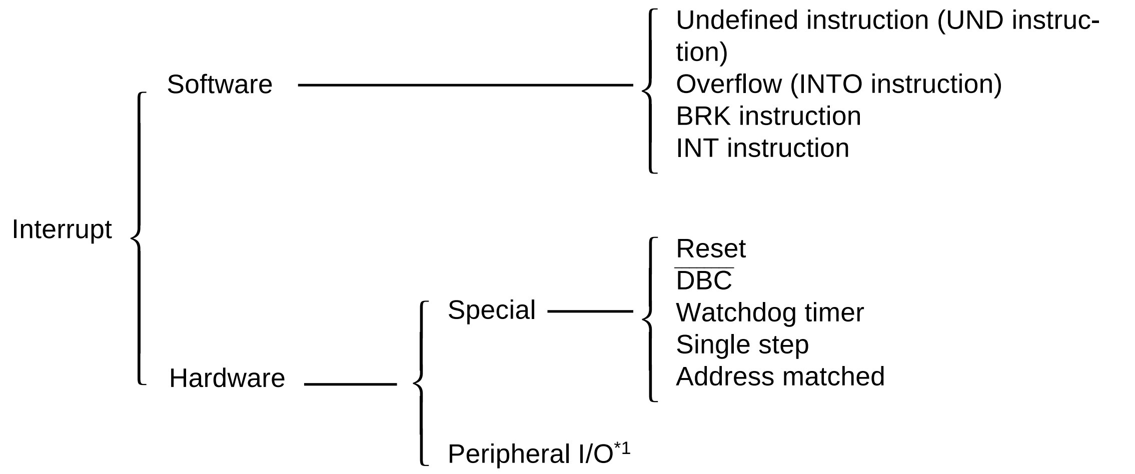
Note: Writing a value to an address after “1” is written to this bit returns the bit to “0” . Other bits do not automatically return to “0” and they must therefore be reset by the program.

Figure 21. Protect register

Overview of Interrupt

Type of Interrupts

Figure 22 lists the types of interrupts.



*1 Peripheral I/O interrupts are generated by the peripheral functions built into the microcomputer system.

Figure 22. 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 assigning 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 RESET pin.
- **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. 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.

- **Key-input interrupt**
  A key-input interrupt occurs if an “L” is input to the $\overline{\mathrm{KI}}$ pin.
- **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.
- **Timer A0 interrupt**
  This is an interrupts that timer A0 generates.
- **Timer B0 and timer B2 interrupt**
  These are interrupts that timer B generates.
- **Timer X0 to timer X2 interrupt**
  These are interrupts that timer X generates.
- **$\overline{\mathbf{INT0}}$ and $\overline{\mathbf{INT1}}$ interrupt**
  An $\overline{\mathrm{INT}}$ interrupt occurs if either a rising edge or a falling edge is input to the $\overline{\mathrm{INT}}$ 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 23 shows format for specifying interrupt vector addresses.

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.

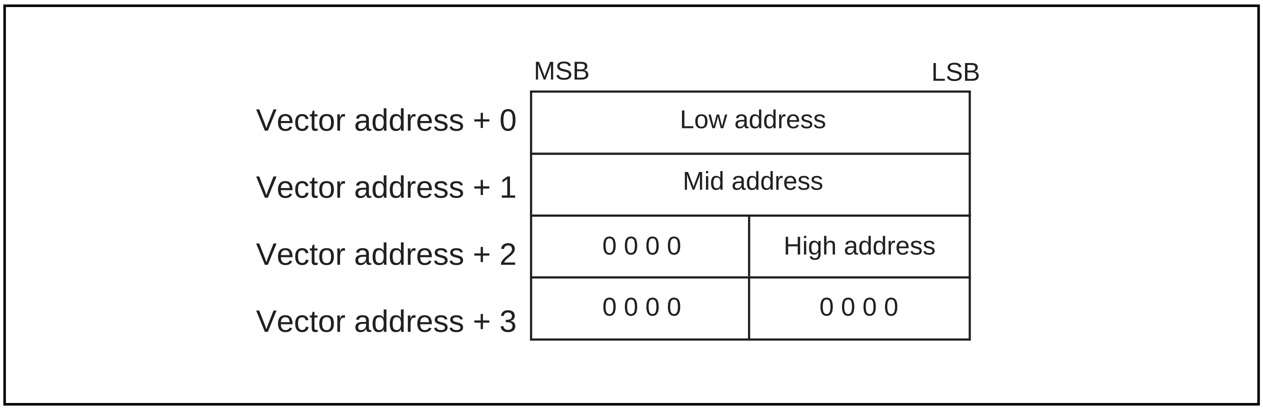


Figure 23. Format for specifying interrupt vector addresses

• 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$_{16}$ to FFFFF$_{16}$. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table 7 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

Table 7. Interrupt and fixed vector address

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFDC$_{16}$ to FFFDF$_{16}$	Interrupt on UND instruction
Overflow	FFFE0$_{16}$ to FFFE3$_{16}$	Interrupt on INTO instruction
BRK instruction	FFFE4$_{16}$ to FFFE7$_{16}$	If the vector is filled with FF$_{16}$, program execution starts from the address shown by the vector in the variable vector table
Address match	FFFE8$_{16}$ to FFFEB$_{16}$	There is an address-matching interrupt enable bit
Single step (Note)	FFFEC$_{16}$ to FFFEF$_{16}$	Do not use
Watchdog timer	FFFF0$_{16}$ to FFFF3$_{16}$	
$\overline{\text{DBC}}$ (Note)	FFFF4$_{16}$ to FFFF7$_{16}$	Do not use
-	FFFF8$_{16}$ to FFFFB$_{16}$	-
Reset	FFFFC$_{16}$ to FFFFF$_{16}$	

Note: Interrupts used for debugging purposes only.

• 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 8 shows the interrupts assigned to the variable vector tables and addresses of vector tables.

Table 8. Interrupt causes (variable interrupt vector addresses)

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 by I flag
——		——	
Software interrupt number 11	+44 to +47 (Note)	——	
Software interrupt number 12	+48 to +51 (Note)	——	
Software interrupt number 13	+52 to +55 (Note)	Key input interrupt	
Software interrupt number 14	+56 to +59 (Note)	A-D	
——		——	
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 X0	
Software interrupt number 23	+92 to +95 (Note)	Timer X1	
Software interrupt number 24	+96 to +99 (Note)	Timer X2	
Software interrupt number 25	+100 to +103 (Note)	——	
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)	——	
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)	——	
Software interrupt number 32 to Software interrupt number 63	+128 to +131 (Note) to +252 to +255 (Note)	Software interrupt	Cannot be masked by I flag

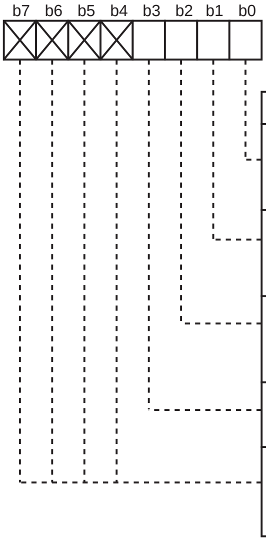
Note : Address relative to address in interrupt table register (INTB).

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 select bit, and 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 24 shows the interrupt control registers.

Interrupt control register

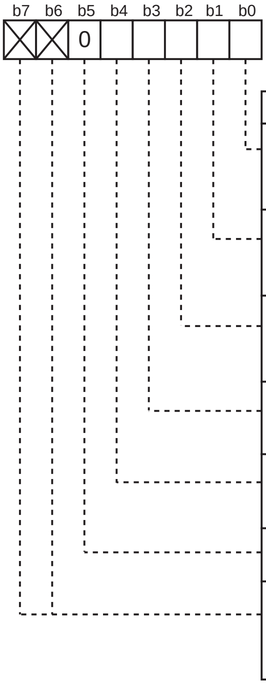
Symbol	Address	When reset
KUPIC	$004D_{16}$	$XXXXX000_2$
ADIC	$004E_{16}$	$XXXXX000_2$
SiTIC(i=0, 1)	0051_{16}, 0053_{16}	$XXXXX000_2$
SiRIC(i=0, 1)	0052_{16}, 0054_{16}	$XXXXX000_2$
TAiIC(i=0)	0055_{16}	$XXXXX000_2$
TXiIC(i=0 to 2)	0056_{16} to 0058_{16}	$XXXXX000_2$
TBiIC(i=0, 1)	$005A_{16}$, $005B_{16}$	$XXXXX000_2$



Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 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	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note)
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

Note: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).

Symbol	Address	When reset
INTiIC(i=0, 1)	$005D_{16}$, $005E_{16}$	$XX00X000_2$



Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 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	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0: Interrupt not requested 1: Interrupt requested	○	○ (Note)
POL	Polarity select bit	0 : Selects falling edge 1 : Selects rising edge	○	○
Reserved bit		Always set to “0”	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

Note: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).

Figure 24. Interrupt control register

Interrupt Enable 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 9 shows the settings of interrupt priority levels and Table 10 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 9. 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
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	High

Table 10. Interrupt levels enabled according to the contents of the IPL

IPL	Enabled interrupt priority levels
IPL_2 IPL_1 IPL_0 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

Changing the Interrupt Control Register

< Program examples >

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                     ; Four NOP instructions are required when using HOLD function.
    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            ; Push Flag register onto stack
    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.

If changing the interrupt control register using an instruction other than the instructions listed hear, and if an interrupt occurs associated with this register during execution of the instruction, there can be instances in which the interrupt request bit is not set. To avoid this problem, use one of the instructions given below to change the register.
Following instructions: AND, OR, BCLR or 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_{16}. After this, the corresponding interrupt request bit becomes "0".

(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.

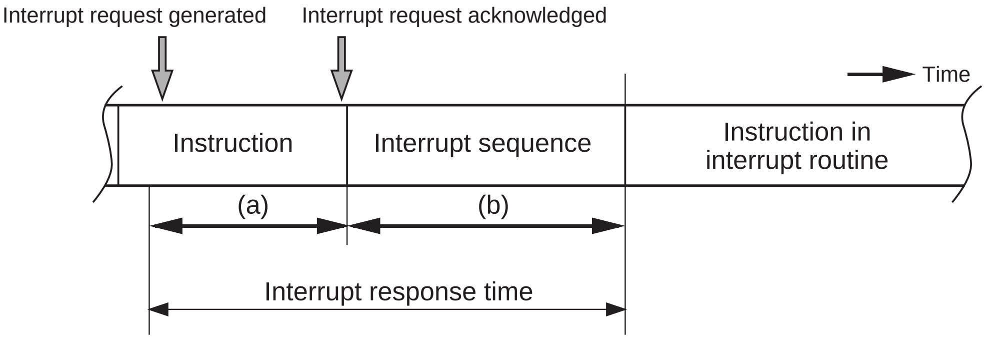
(6) Sets the interrupt priority level of the accepted instruction in the IPL.

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 25 shows the interrupt response time.


Interrupt request generated
Interrupt request acknowledged
Time
Instruction
Interrupt sequence
Instruction in interrupt routine
(a)
(b)
Interrupt response time

(a) Time from interrupt request is generated to when the instruction then under execution is completed.
(b) Time in which the instruction sequence is executed.

Figure 25. Interrupt response time

Time (a) is dependent on the instruction under execution. Thirty cycles is the maximum required for the DIVX instruction (without wait).

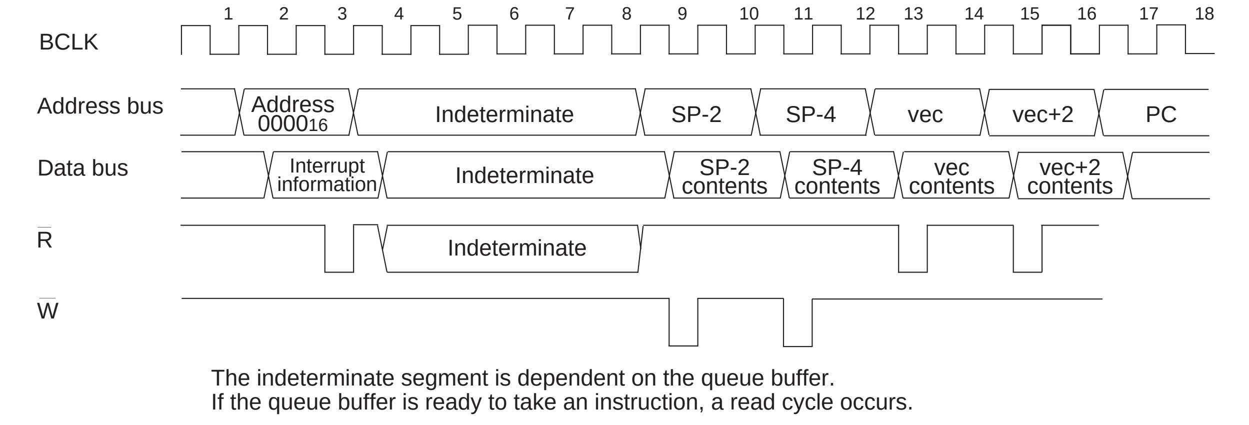
Time (b) is as shown in Table 11.

Table 11. Time required for executing the interrupt sequence

Interrupt vector address	Stack pointer (SP) value	16-bit bus, without wait	8-bit bus, without wait
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 $\overline{\text{DBC}}$ interrupt; add 1 cycle in the case either of an address match interrupt or of a single-step interrupt.

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



The indeterminate segment is dependent on the queue buffer.
If the queue buffer is ready to take an instruction, a read cycle occurs.

Figure 26. 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 12 is set in the IPL.

Table 12. 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 4 high-order bits of the program counter, and 4 high-order bits and 8 low-order bits of the FLG register, 16 bits in total, in the stack area, then saves 16 low-order bits of the program counter. Figure 27 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).

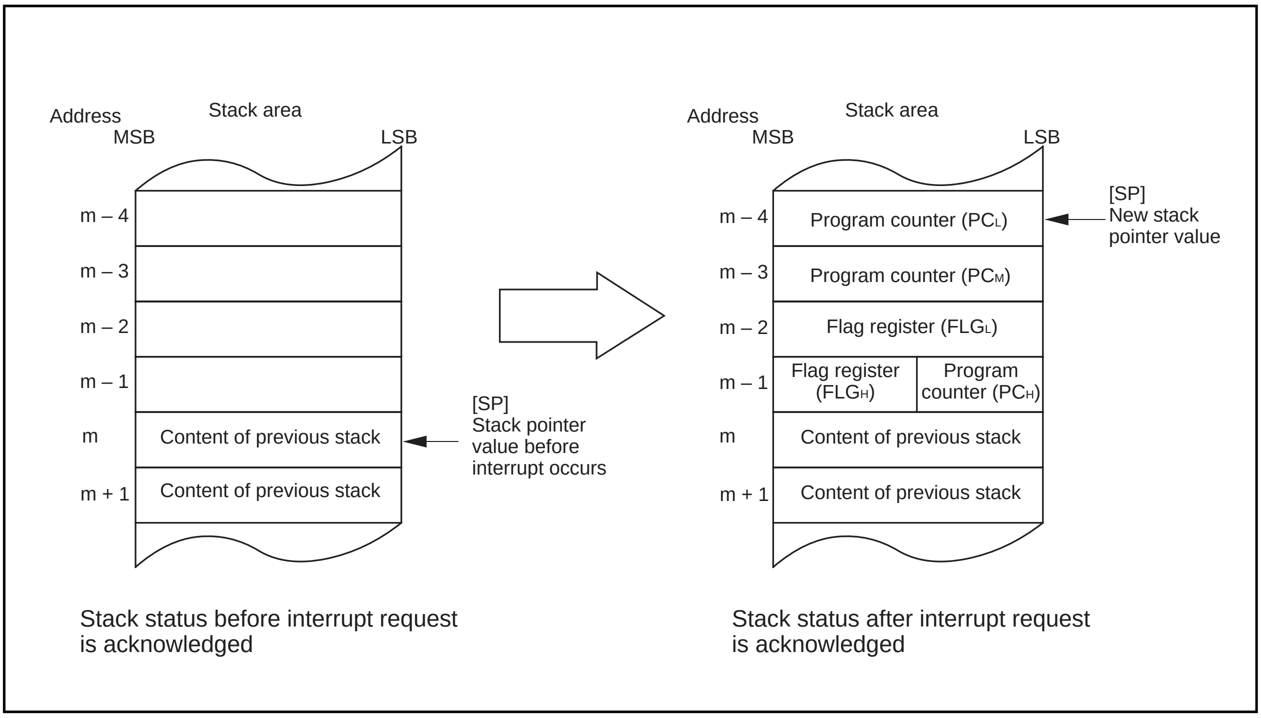
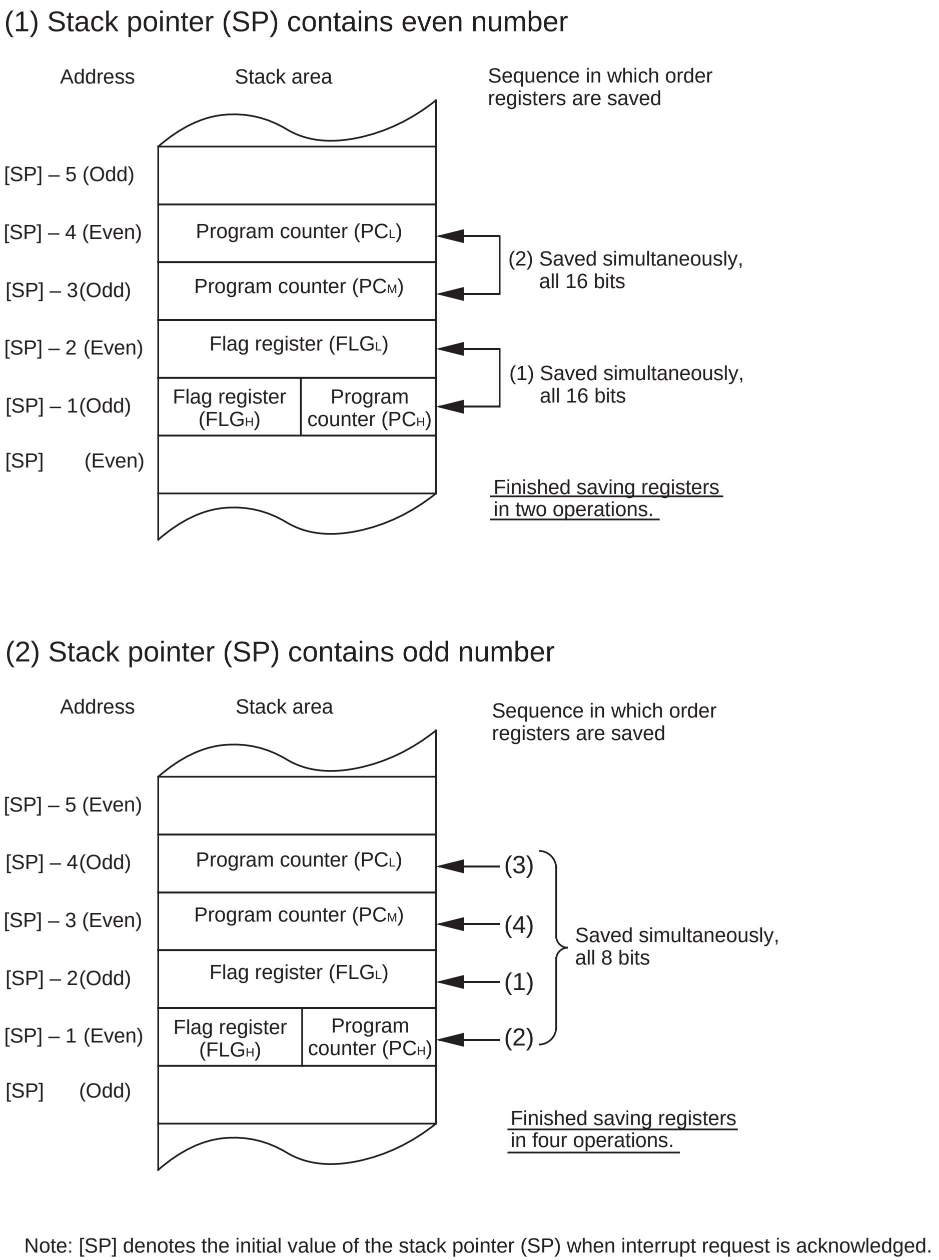


Figure 27. State of stack before and after acceptance of interrupt request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the content of the stack pointer (Note), 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 28 shows the operation of the saving registers.

Note: Stack pointer indicated by U flag.



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

Figure 28. 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 29 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.

Interrupt Priority Level Judge Circuit

This circuit selects the interrupt with the highest priority level when two or more interrupts are generated simultaneously.

Figure 30 shows the interrupt resolution circuit.

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

Figure 29. Hardware interrupts priorities

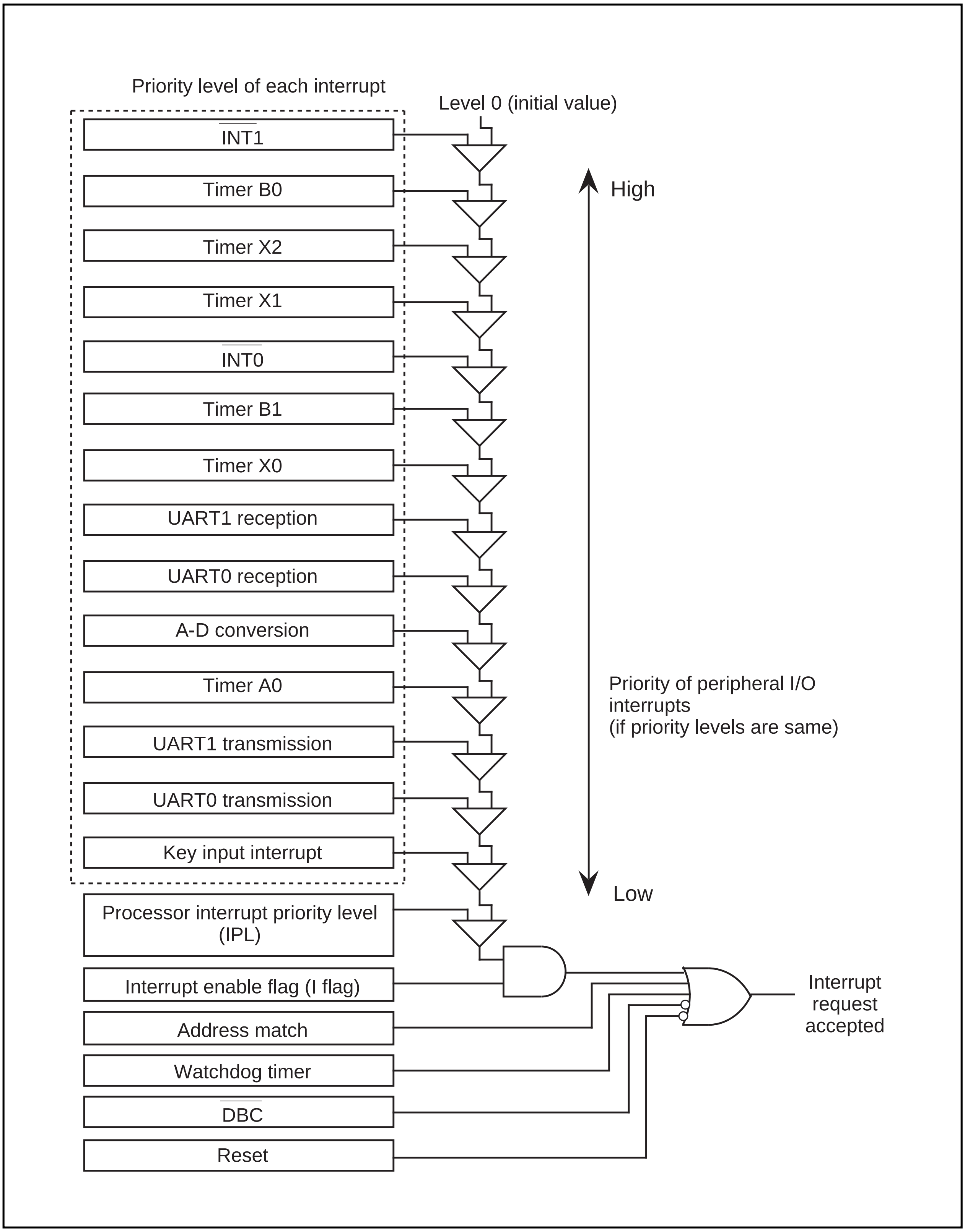


Figure 30. Interrupt resolution circuit

Key Input Interrupt

If the direction register of any of P0$_0$ to P0$_7$ is set for input and a falling edge is input to that port, a key input interrupt is generated. A key input interrupt can also be used as a key-on wakeup function for cancelling the wait mode or stop mode. Figure 31 shows the block diagram of the key input interrupt. Note that if an “L” level is input to any pin that has not been disabled for input, inputs to the other pins are not detected as an interrupt.

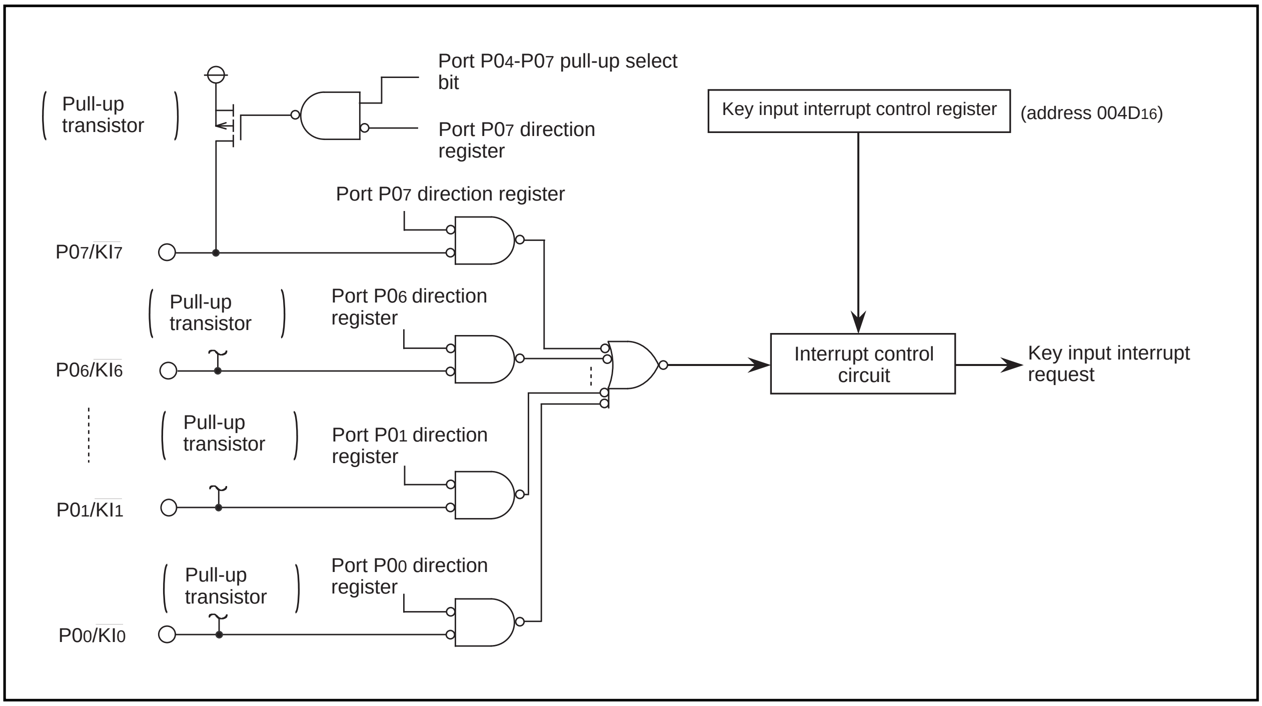
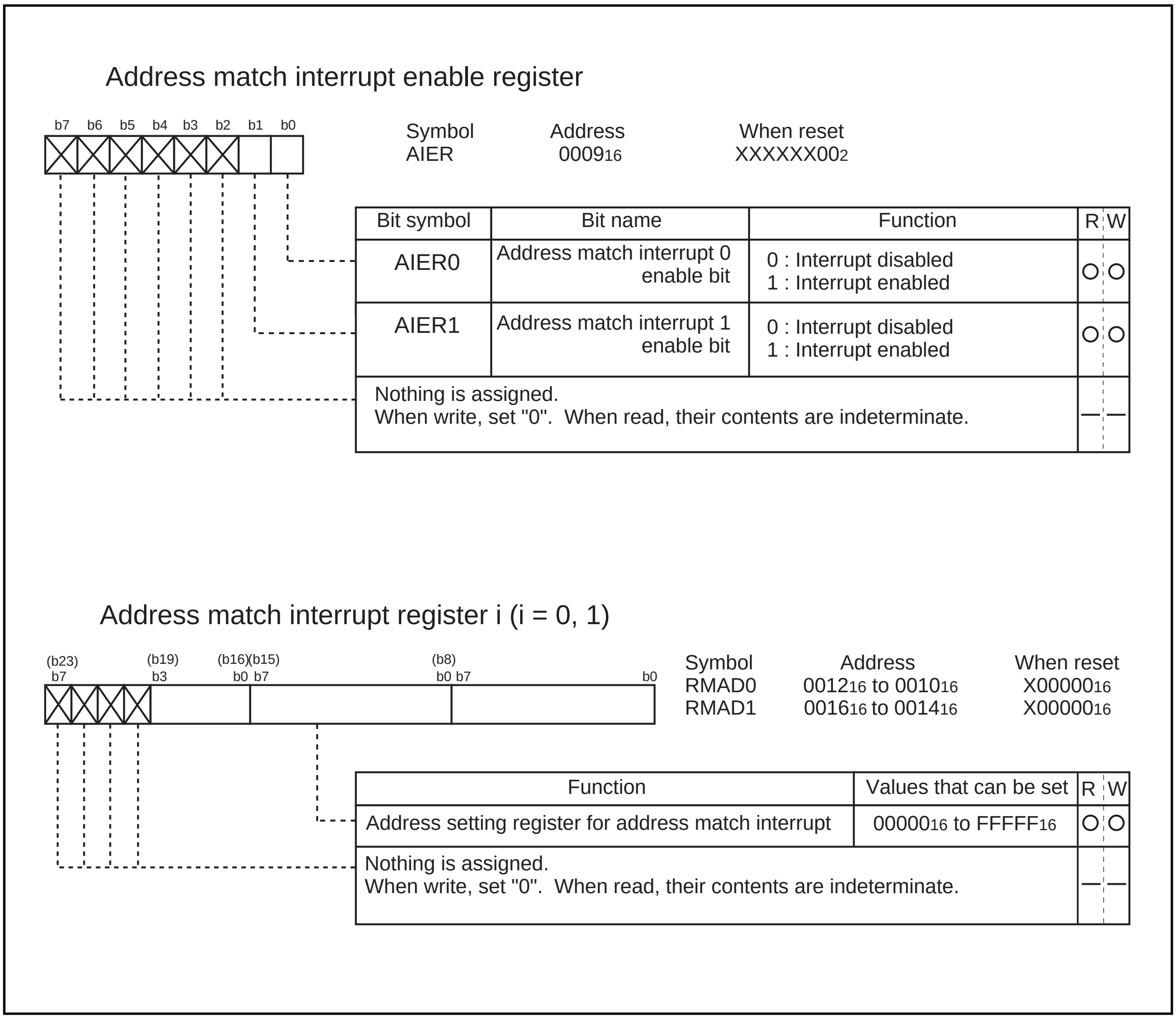

Pull-up transistor
Port P04-P07 pull-up select bit
Port P07 direction register
Key input interrupt control register
(address 004D16)
Port P07 direction register
P07/KI7
Pull-up transistor
Port P06 direction register
P06/KI6
Interrupt control circuit
Key input interrupt request
Pull-up transistor
Port P01 direction register
P01/KI1
Pull-up transistor
Port P00 direction register
P00/KI0

Figure 31. Block diagram of key input 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).

Figure 32 shows the address match interrupt-related registers.



Address match interrupt enable register

Symbol	Address	When reset
AIER	0009$_{16}$	XXXXXX00$_{2}$

Bit symbol	Bit name	Function	R	W
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

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

Symbol	Address	When reset
RMAD0	0012$_{16}$ to 0010$_{16}$	X00000$_{16}$
RMAD1	0016$_{16}$ to 0014$_{16}$	X00000$_{16}$

Function	Values that can be set	R	W
Address setting register for address match interrupt	00000$_{16}$ to FFFFF$_{16}$	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.		—	—

Figure 32. Address match interrupt-related registers

Precautions for Interrupts

(1) Reading address 00000_{16}

- 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_{16} will then be set to “0”.

  Reading address 00000_{16} 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_{16} by software.

(2) Setting the stack pointer

- The value of the stack pointer immediately after reset is initialized to 0000_{16}. 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. Concerning the first instruction immediately after reset, generating any interrupts is prohibited.

(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 $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ regardless of the CPU operation clock.
- When changing a polarity of pins $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$, the interrupt request bit may become "1". Clear the interrupt request bit after changing the polarity. Figure 33 shows the switching condition of $\overline{\text{INT}}$ interrupt request.

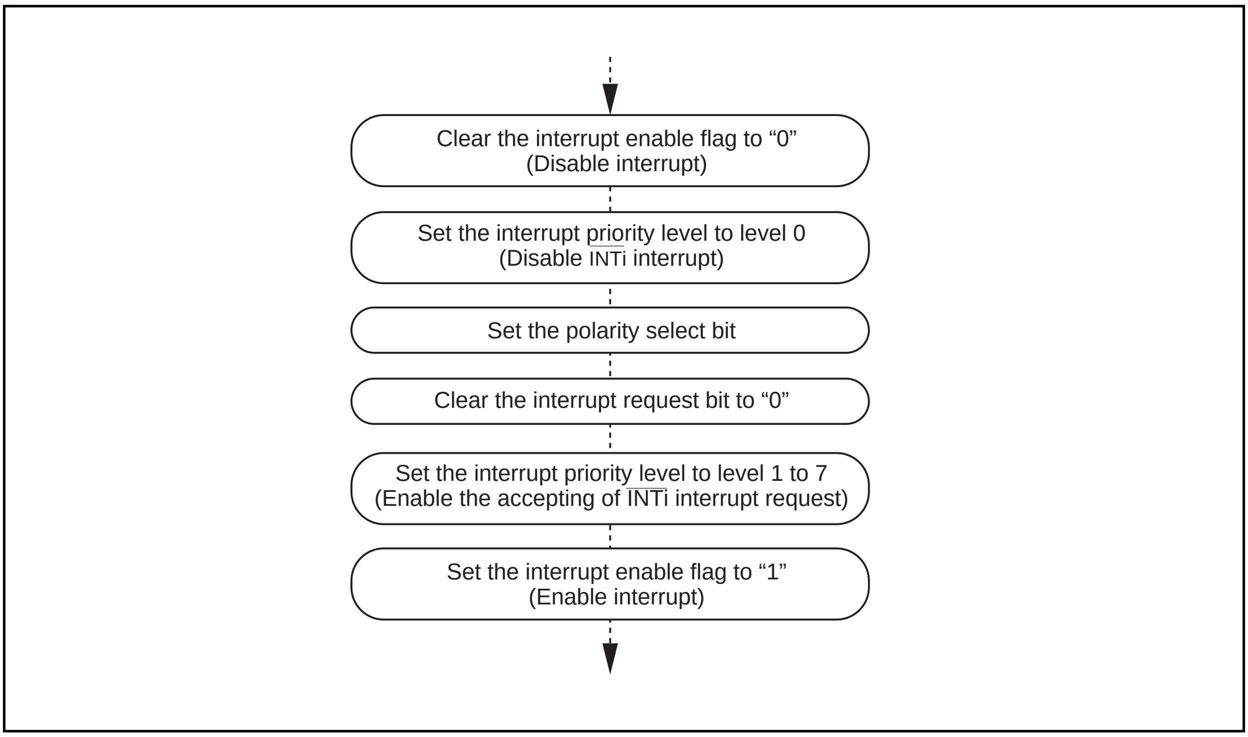

Clear the interrupt enable flag to “0”
(Disable interrupt)
Set the interrupt priority level to level 0
(Disable $\overline{\text{INT}}$i interrupt)
Set the polarity select bit
Clear the interrupt request bit to “0”
Set the interrupt priority level to level 1 to 7
(Enable the accepting of $\overline{\text{INT}}$i interrupt request)
Set the interrupt enable flag to “1”
(Enable interrupt)

Figure 33. Switching condition of $\overline{\text{INT}}$ interrupt request

(4) Changing interrupt control register

See "Changing Interrupt Control Register".

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 X_{IN} is selected for the BCLK, bit 7 of the watchdog timer control register (address $000F_{16}$) selects the prescaler division ratio (by 16 or by 128). When X_{CIN} is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address $000F_{16}$).

When X_{IN} is selected in BCLK

$$\text{Watchdog timer cycle} = \frac{\text{Prescaler division ratio (16 or 128) x watchdog timer count (32768)}}{\text{BCLK}}$$

When X_{CIN} is selected in BCLK

$$\text{Watchdog timer cycle} = \frac{\text{Prescaler division ratio (2) x watchdog timer count (32768)}}{\text{BCLK}}$$

For example, when BCLK is 10MHz and the prescaler division ratio is set to 16, the watchdog timer cycle is approximately 52.4 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address $000E_{16}$) 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 $000E_{16}$).

Figure 34 shows the block diagram of the watchdog timer. Figure 35 shows the watchdog timer-related registers.

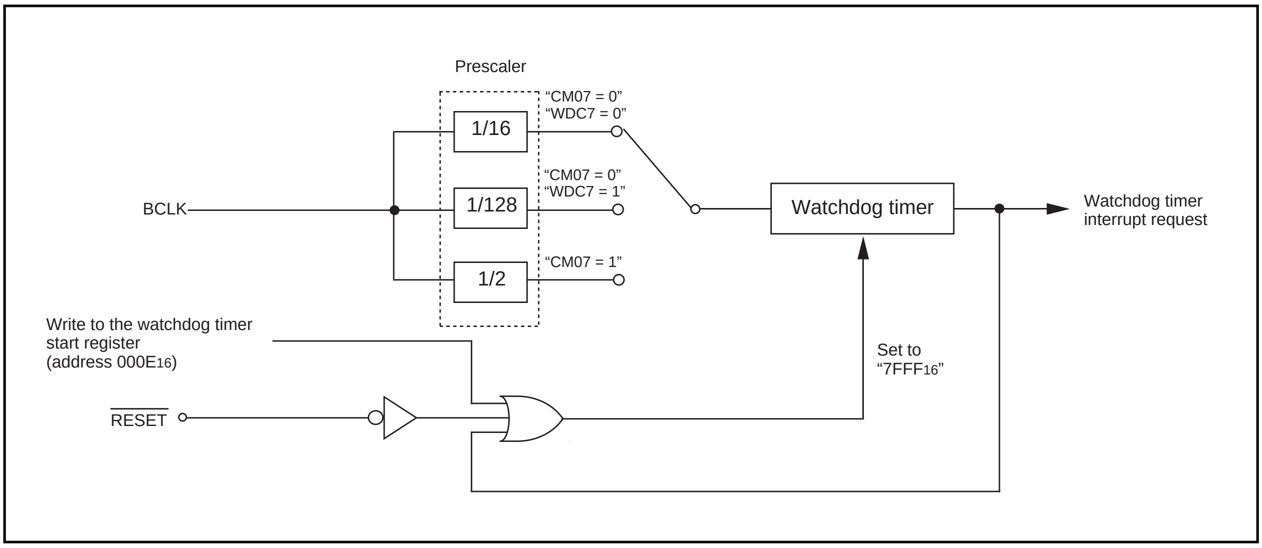


Figure 34. Block diagram of watchdog timer

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol	Address	When reset
WDC	000F$_{16}$	000XXXXX$_2$

Bit symbol	Bit name	Function	R	W
High-order bit of watchdog timer			○	×
Reserved bit		Must always be set to “0”	○	○
Reserved bit		Must always be set to “0”	○	○
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	○	○

Watchdog timer start register

b7 … b0

Symbol	Address	When reset
WDTS	000E$_{16}$	Indeterminate

Function	R	W
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to “7FFF$_{16}$” regardless of whatever value is written.	×	○

Figure 35. Watchdog timer control and start registers

Timer

There are six 16-bit timers. These timers can be classified by function into timer A (one), timers B (two) and timers X (three). All these timers function independently. Figures 36 show the block diagram of timers.

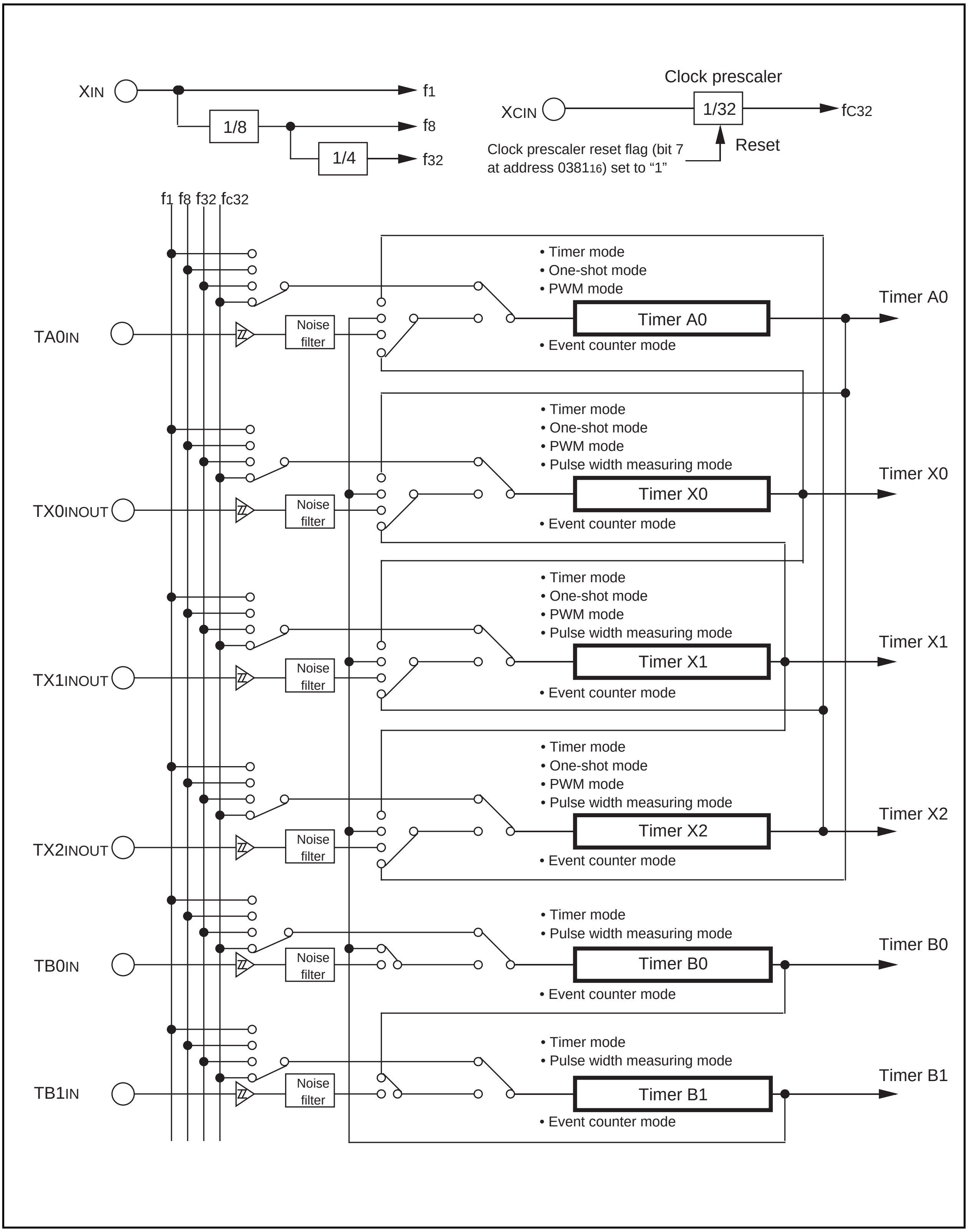


Figure 36. Timer block diagram

Timer A

Figure 37 shows the block diagram of timer A. Figures 38 to 40 show the timer A-related registers.

Use the timer A0 mode register 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 over flow.
- One-shot timer mode: The timer stops counting when the count reaches “0000₁₆”.
- Pulse width modulation (PWM) mode: The timer outputs pulses of a given width.

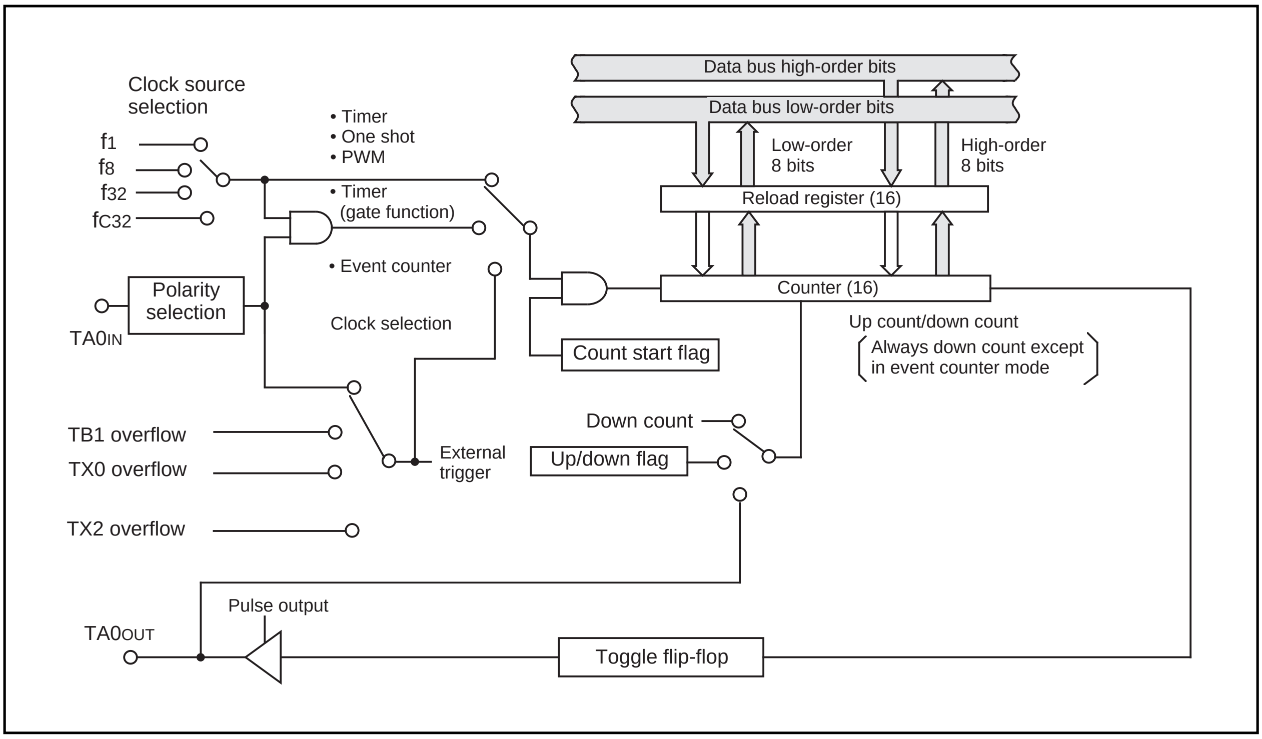
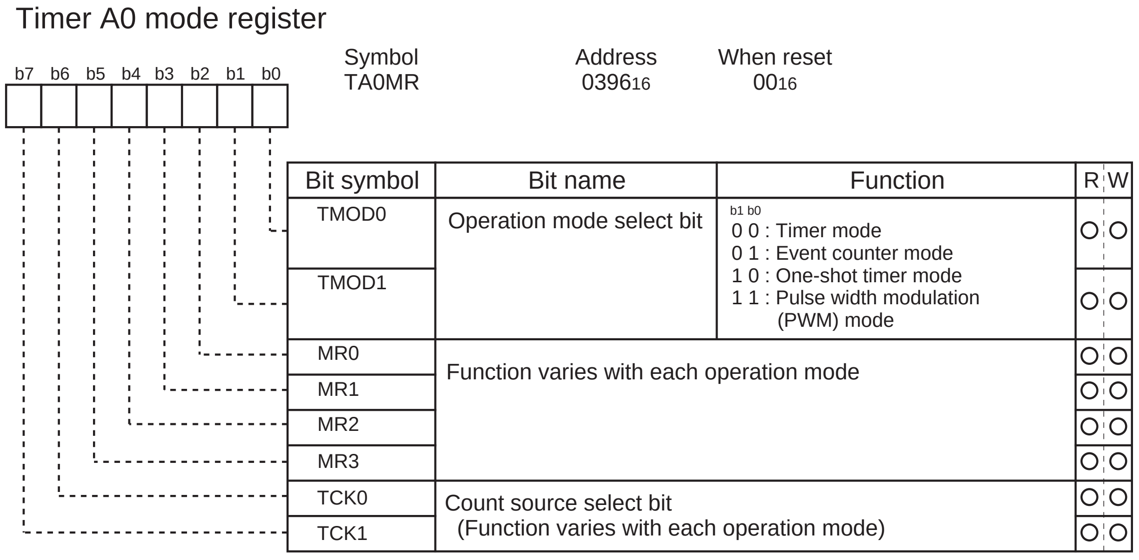


Figure 37. Block diagram of timer A



Timer A0 mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TA0MR	0396₁₆	00₁₆

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse width modulation (PWM) mode	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○	○
MR3			○	○
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Figure 38. Timer A-related registers (1)

Timer A0 register (Note)

(b15) b7 (b8) b0 b7 b0

Symbol	Address	When reset
TA0	0387_{16},0386_{16}	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	0000_{16} to $FFFF_{16}$	○	○
• Event counter mode Counts pulses from an external source or timer overflow	0000_{16} to $FFFF_{16}$	○	○
• One-shot timer mode Counts a one shot width	0000_{16} to $FFFF_{16}$	×	○
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000_{16} to $FFFE_{16}$	×	○
• Pulse width modulation mode (8-bit PWM) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	00_{16} to FF_{16} (High-order addresses) 00_{16} to FE_{16} (Low-order addresses)	×	○

Note: Read and write data in 16-bit units.

Count start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TABSR	0380_{16}	$000X0000_2$

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TX0S	Timer X0 count start flag		○	○
TX1S	Timer X1 count start flag		○	○
TX2S	Timer X2 count start flag		○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
TB0S	Timer B0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB1S	Timer B1 count start flag		○	○
CDCS	Clock devided count start flag		○	○

Up/down flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UDF	0384_{16}	$XXX0XXX0_2$

Bit symbol	Bit name	Function	R	W
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count This specification becomes valid when the up/down flag content is selected for up/down switching cause	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
TA0P	Timer A0 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled When not using the two-phase pulse signal processing function, set the select bit to “0”	×	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

Figure 39. Timer A-related registers (2)

One-shot start flag

b7	b6	b5	b4	b3	b2	b1	b0
☒	☒	☒	☒				

Symbol	Address	When reset
ONSF	0382$_{16}$	XXXX0000$_2$

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is “0”	○	○
TX0OS	Timer X0 one-shot start flag		○	○
TX1OS	Timer X1 one-shot start flag		○	○
TX2OS	Timer X2 one-shot start flag		○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.			—	—

Trigger select register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
TRGSR	0383$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TA0TGL	Timer A0 event/trigger select bit	b1 b0 0 0 : Input on TA0$_{IN}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX2 overflow is selected 1 1 : TX0 overflow is selected	○	○
TA0TGH			○	○
TX0TGL	Timer X0 event/trigger select bit	b3 b2 0 0 : Input on TX0$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TX1 overflow is selected	○	○
TX0TGH			○	○
TX1TGL	Timer X1 event/trigger select bit	b5 b4 0 0 : Input on TX1$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX0 overflow is selected 1 1 : TX2 overflow is selected	○	○
TX1TGH			○	○
TX2TGL	Timer X2 event/trigger select bit	b7 b6 0 0 : Input on TX2$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX1 overflow is selected 1 1 : TA0 overflow is selected	○	○
TX2TGH			○	○

Note: Set the corresponding port direction register to “0”(input mode).

Clock prescaler reset flag

b7	b6	b5	b4	b3	b2	b1	b0
	☒	☒	☒	☒	☒	☒	☒

Symbol	Address	When reset
CPSRF	0381$_{16}$	0XXXXXXX$_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

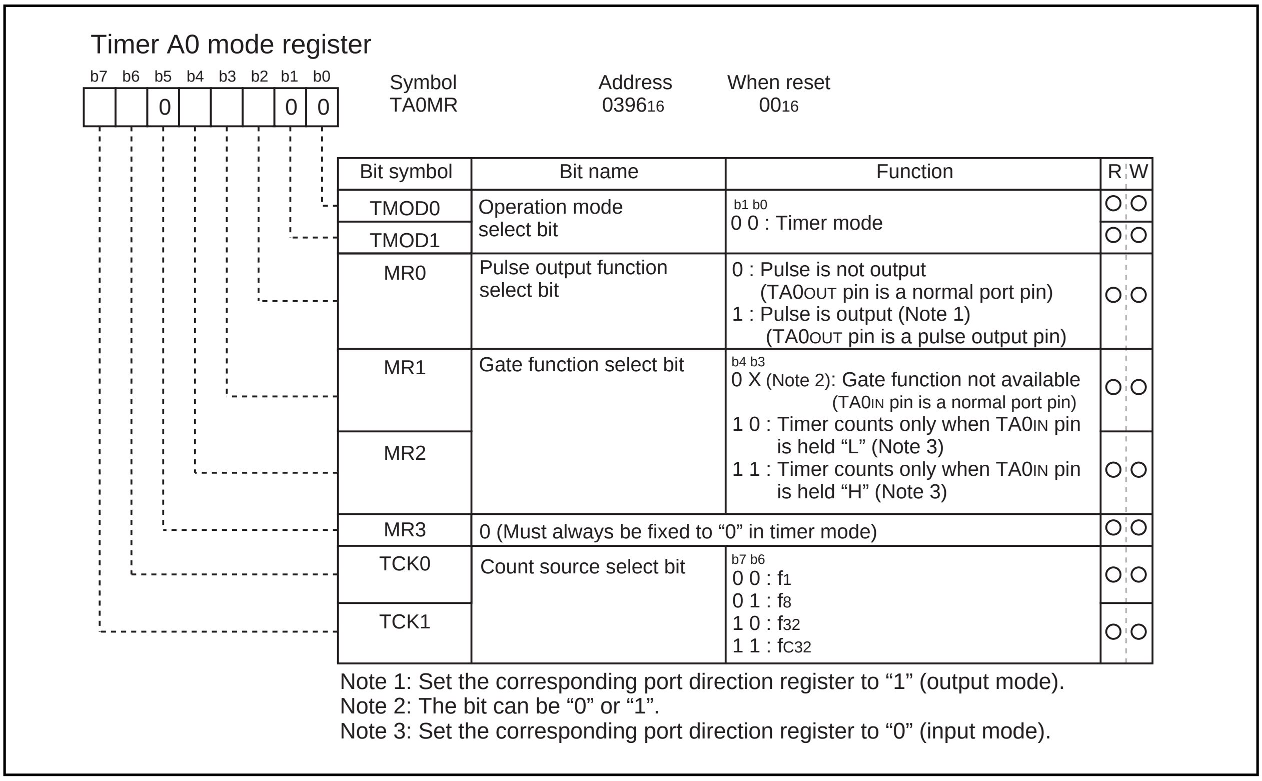
Figure 40. Timer A-related registers (3)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 13.) Figure 41 shows the timer A0 mode register in timer mode.

Table 13. Specifications of timer mode

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
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
TA0IN pin function	Programmable I/O port or gate input
TA0OUT pin function	Programmable I/O port or pulse output
Read from timer	Count value can be read out by reading timer A0 register
Write to timer	• When counting stopped When a value is written to timer A0 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A0 register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Gate function Counting can be started and stopped by the TA0IN pin's input signal • Pulse output function Each time the timer underflows, the TA0OUT pin's polarity is reversed



Symbol	Address	When reset
TA0MR	0396$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TA0OUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TA0OUT pin is a pulse output pin)	○	○
MR1	Gate function select bit	b4 b3 0 X (Note 2): Gate function not available (TA0IN pin is a normal port pin) 1 0 : Timer counts only when TA0IN pin is held “L” (Note 3) 1 1 : Timer counts only when TA0IN pin is held “H” (Note 3)	○	○
MR2			○	○
MR3	0 (Must always be fixed to “0” in timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : fc32	○	○
TCK1			○	○

Note 1: Set the corresponding port direction register to “1” (output mode).
Note 2: The bit can be “0” or “1”.
Note 3: Set the corresponding port direction register to “0” (input mode).

Figure 41. Timer A0 mode register in timer mode

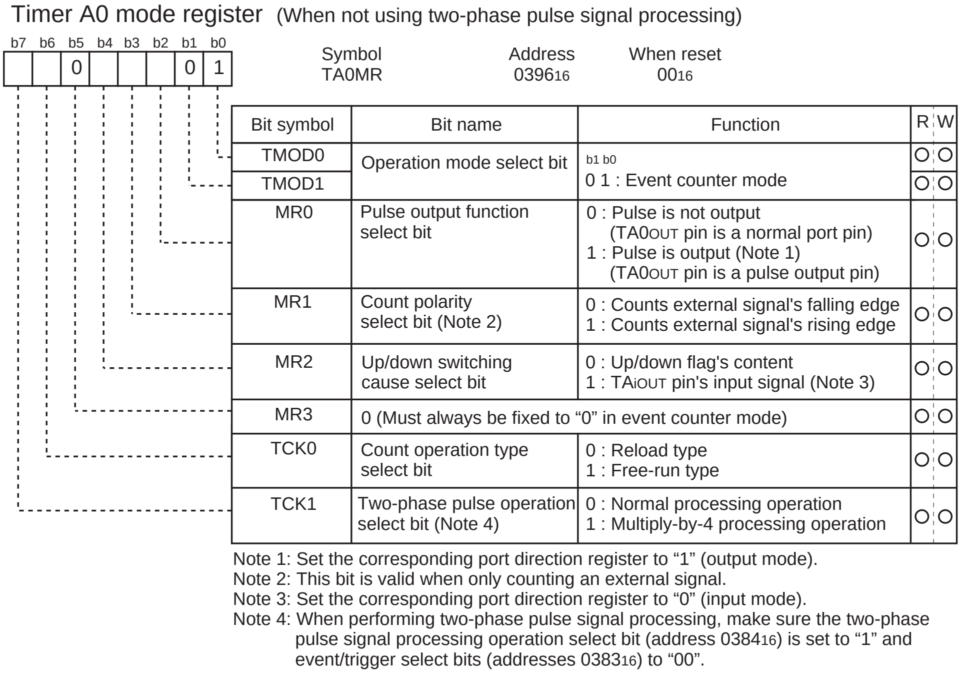
(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. Timer A0 can count a single-phase and a two-phase external signal. Table 14 lists timer specifications when counting a single-phase external signal. Figure 42 shows the timer A0 mode register in event counter mode.
Table 15 lists timer specifications when counting a two-phase external signal. Figure 43 shows the timer A0 mode register in event counter mode.

Table 14. Timer specifications in event counter mode (when not processing two-phase pulse signal)

Item	Specification
Count source	• External signals input to TA0IN pin (effective edge can be selected by software) • TB1 overflow, TX0 overflow, TX2 overflow
Count operation	• Up count or down count can be selected by external signal or software • When the timer overflows or underflows, it reloads the reload register con tents before continuing counting (Note)
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
TA0IN pin function	Programmable I/O port or count source input
TA0OUT 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 A0 register
Write to timer	• When counting stopped When a value is written to timer A0 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A0 register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it • Pulse output function Each time the timer overflows or underflows, the TA0OUT pin's polarity is reversed

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



Timer A0 mode register (When not using two-phase pulse signal processing)

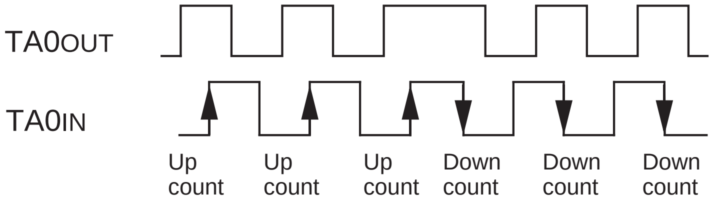
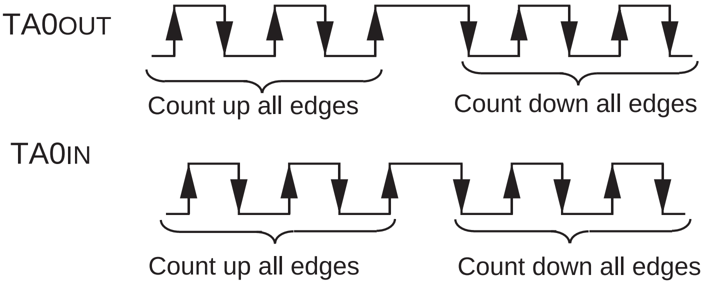
Symbol	Address	When reset
TA0MR	0396$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TA0OUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TA0OUT pin is a pulse output pin)	○	○
MR1	Count polarity select bit (Note 2)	0 : Counts external signal's falling edge 1 : Counts external signal's rising edge	○	○
MR2	Up/down switching cause select bit	0 : Up/down flag's content 1 : TAiOUT pin's input signal (Note 3)	○	○
MR3	0 (Must always be fixed to "0" in event counter mode)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Two-phase pulse operation select bit (Note 4)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	○	○

Note 1: Set the corresponding port direction register to "1" (output mode).
Note 2: This bit is valid when only counting an external signal.
Note 3: Set the corresponding port direction register to "0" (input mode).
Note 4: When performing two-phase pulse signal processing, make sure the two-phase pulse signal processing operation select bit (address 0384$_{16}$) is set to "1" and event/trigger select bits (addresses 0383$_{16}$) to "00".

Figure 42. Timer A0 mode register in event counter mode

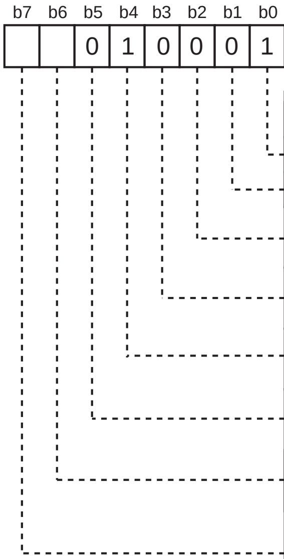
Table 15. Timer specifications in event counter mode (when processing two-phase pulse signal)

Item	Specification
Count source	• Two-phase pulse signals input to TA0IN or TA0OUT pin
Count operation	• Up count or down count can be selected by two-phase pulse signal • When the timer overflows or underflows, the reload register content is reloaded and the timer starts over again (Note)
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	Timer overflows or underflows
TA0IN pin function	Two-phase pulse input
TA0OUT pin function	Two-phase pulse input
Read from timer	Count value can be read out by reading timer A0 register
Write to timer	• When counting stopped When a value is written to timer A0 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A0 register, it is written to only reload register. (Transferred to counter at next reload time.)
Select function	• Normal processing operation The timer counts up rising edges or counts down falling edges on the TA0IN pin when input signal on the TA0OUT pin is "H"  TA0OUT TA0IN Up count Up count Up count Down count Down count Down count • Multiply-by-4 processing operation If the phase relationship is such that the TA0IN pin goes "H" when the input signal on the TA0OUT pin is "H", the timer counts up rising and falling edges on the TA0OUT and TA0IN pins. If the phase relationship is such that the TA0IN pin goes "L" when the input signal on the TA0OUT pin is "H", the timer counts down rising and falling edges on the TA0OUT and TA0IN pins.  TA0OUT Count up all edges Count down all edges TA0IN Count up all edges Count down all edges

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

Timer A0 mode register

(When using two-phase pulse signal processing)



Symbol	Address	When reset
TA0MR	0396_{16}	00_{16}

	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	0 (Must always be “0” when using two-phase pulse signal processing)		○	○
MR1	0 (Must always be “0” when using two-phase pulse signal processing)		○	○
MR2	1 (Must always be “1” when using two-phase pulse signal processing)		○	○
MR3	0 (Must always be “0” when using two-phase pulse signal processing)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Two-phase pulse processing operation select bit (Note)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	○	○

Note: When performing two-phase pulse signal processing, make sure the two-phase pulse signal processing operation select bit (address 0384_{16}) is set to “1”. Also, always be sure to set the event/trigger select bit (addresses 0383_{16}) to “00”.

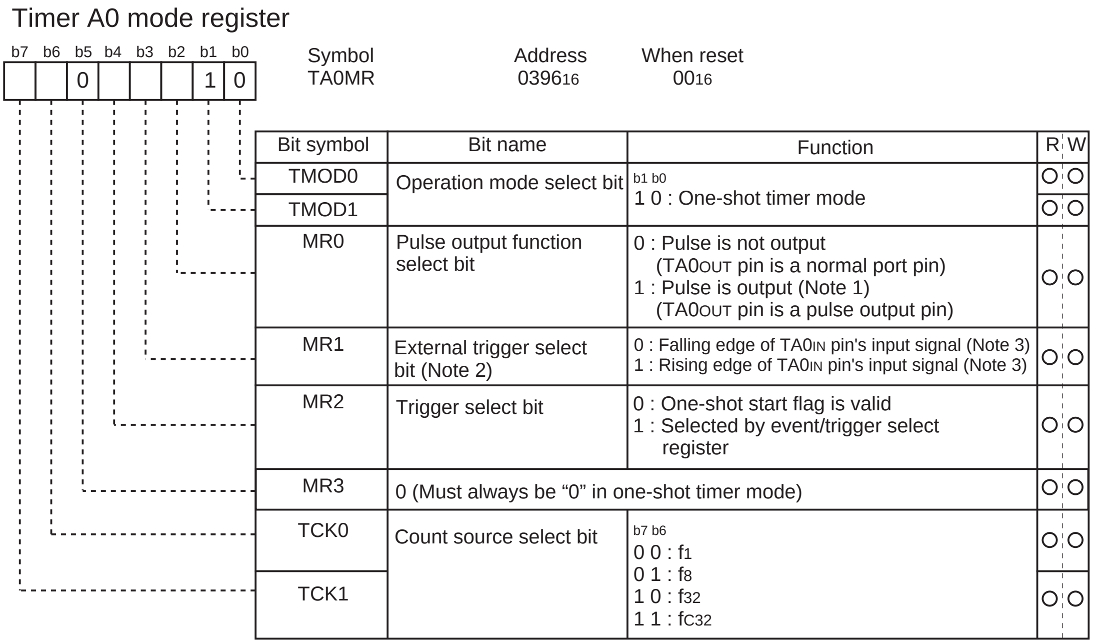
Figure 43. Timer A0 mode register in event counter mode

(3) One-shot timer mode

In this mode, the timer operates only once. (See Table 16.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 44 shows the timer A0 mode register in one-shot timer mode.

Table 16. Timer specifications in one-shot timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• The timer counts down • When the count reaches 0000$_{16}$, the timer stops counting after reloading a new count • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : Set value
Count start condition	• An external trigger is input • The timer overflows • The one-shot start flag is set (= 1)
Count stop condition	• A new count is reloaded after the count has reached 0000$_{16}$ • The count start flag is reset (= 0)
Interrupt request generation timing	The count reaches 0000$_{16}$
TA0IN pin function	Programmable I/O port or trigger input
TA0OUT pin function	Programmable I/O port or pulse output
Read from timer	When timer A0 register is read, it indicates an indeterminate value
Write to timer	• When counting stopped When a value is written to timer A0 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A0 register, it is written to only reload register (Transferred to counter at next reload time)



Symbol TA0MR — Address 0396$_{16}$ — When reset 00$_{16}$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TA0OUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TA0OUT pin is a pulse output pin)	○	○
MR1	External trigger select bit (Note 2)	0 : Falling edge of TA0IN pin's input signal (Note 3) 1 : Rising edge of TA0IN pin's input signal (Note 3)	○	○
MR2	Trigger select bit	0 : One-shot start flag is valid 1 : Selected by event/trigger select register	○	○
MR3	0 (Must always be “0” in one-shot timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Set the corresponding port direction register to “1” (output mode).
Note 2: Valid only when the TA0IN pin is selected by the event/trigger select bit (addresses 0383$_{16}$). If timer overflow is selected, this bit can be “1” or “0”.
Note 3: Set the corresponding port direction register to “0” (input mode).

Figure 44. Timer A0 mode register in one-shot timer mode

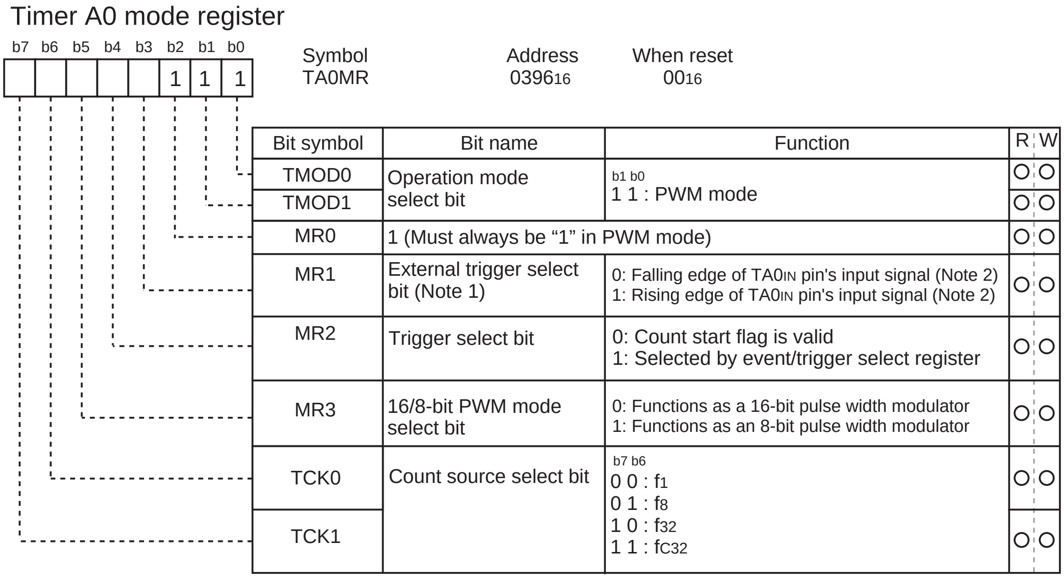
(4) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table 17.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 45 shows the timer A0 mode register in pulse width modulation mode. Figure 46 shows the example of how a 16-bit pulse width modulator operates. Figure 47 shows the example of how an 8-bit pulse width modulator operates.

Table 17. Timer specifications in pulse width modulation mode

Item		Specification
Count source		f$_{1}$, f$_{8}$, f$_{32}$, fc$_{32}$
Count operation		• The timer counts down (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new count at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs when counting
16-bit PWM		• High level width n / fi n : Set value • Cycle time $(2^{16}-1)$ / fi fixed
8-bit PWM		• High level width n ×(m+1) / fi n : values set to timer A0 register's high-order address • Cycle time $(2^{8}-1)$ ×(m+1) / fi m : values set to timer A0 register's low-order address
Count start condition		• External trigger is input • The timer overflows • The count start flag is set (= 1)
Count stop condition		• The count start flag is reset (= 0)
Interrupt request generation timing	8 bits PWM	• Set value of "H" level width is except FF$_{16}$, 00$_{16}$: PWM pulse goes "L" • Set value of "H" level width is FF$_{16}$, 00$_{16}$: Timing that count value goes to 01$_{16}$
	16 bits PWM	• Set value of "H" level width is except FFFF$_{16}$, 0000$_{16}$: PWM pulse goes "L" • Set value of "H" level width is FFFF$_{16}$, 0000$_{16}$: Timing that count value goes to 0001$_{16}$
TA0$_{IN}$ pin function		Programmable I/O port or trigger input
TA0$_{OUT}$ pin function		Pulse output
Read from timer		When timer A0 register is read, it indicates an indeterminate value
Write to timer		• When counting stopped :When a value is written to timer A0 register, it is written to both reload register and counter • When counting in progress : When a value is written to timer A0 register, it is written to only reload register (Transferred to counter at next reload time)

Note: When set value of "H" level width is 00$_{16}$ or 0000$_{16}$, pulse outputs "L" level and inversion value, FF$_{16}$ or FFFF$_{16}$ is set to timer.



Symbol	Address	When reset
TA0MR	0396$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 1 : PWM mode	○	○
TMOD1			○	○
MR0	1 (Must always be "1" in PWM mode)		○	○
MR1	External trigger select bit (Note 1)	0: Falling edge of TA0$_{IN}$ pin's input signal (Note 2) 1: Rising edge of TA0$_{IN}$ pin's input signal (Note 2)	○	○
MR2	Trigger select bit	0: Count start flag is valid 1: Selected by event/trigger select register	○	○
MR3	16/8-bit PWM mode select bit	0: Functions as a 16-bit pulse width modulator 1: Functions as an 8-bit pulse width modulator	○	○
TCK0	Count source select bit	b7 b6 0 0 : f$_{1}$ 0 1 : f$_{8}$ 1 0 : f$_{32}$ 1 1 : fc$_{32}$	○	○
TCK1			○	○

Note 1: Valid only when the TA0$_{IN}$ pin is selected by the event/trigger select bit (addresses 0383$_{16}$). If timer overflow is selected, this bit can be "1" or "0".
Note 2: Set the corresponding port direction register to "0" (input mode).
Note 3: Set the corresponding port direction register to "1" (output mode) when the pulse is output.

Figure 45. Timer A0 mode register in pulse width modulation mode

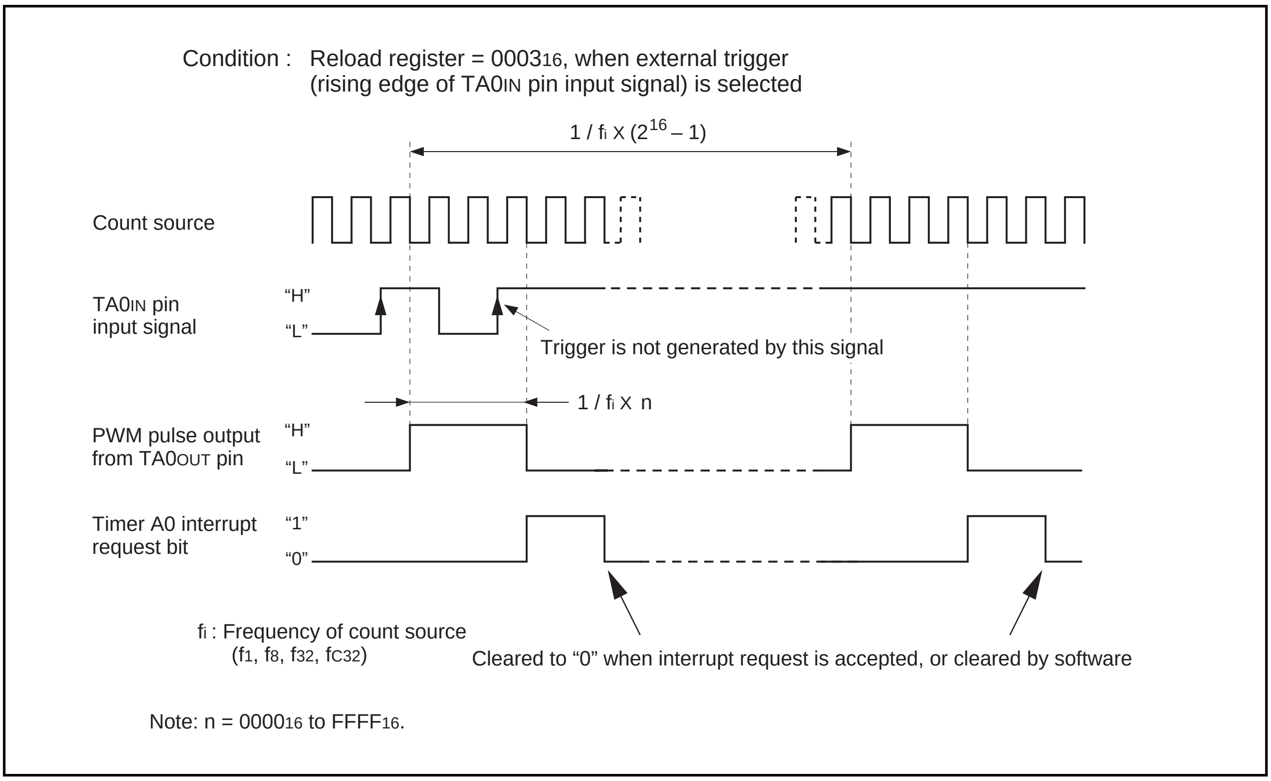

Condition : Reload register = 0003₁₆, when external trigger (rising edge of TA0IN pin input signal) is selected
1 / fi X (2^16 – 1)
Count source
TA0IN pin input signal
"H"
"L"
Trigger is not generated by this signal
1 / fi X n
PWM pulse output from TA0OUT pin
"H"
"L"
Timer A0 interrupt request bit
"1"
"0"
fi : Frequency of count source (f1, f8, f32, fC32)
Cleared to "0" when interrupt request is accepted, or cleared by software
Note: n = 0000₁₆ to FFFF₁₆.

Figure 46. Example of how a 16-bit pulse width modulator operates

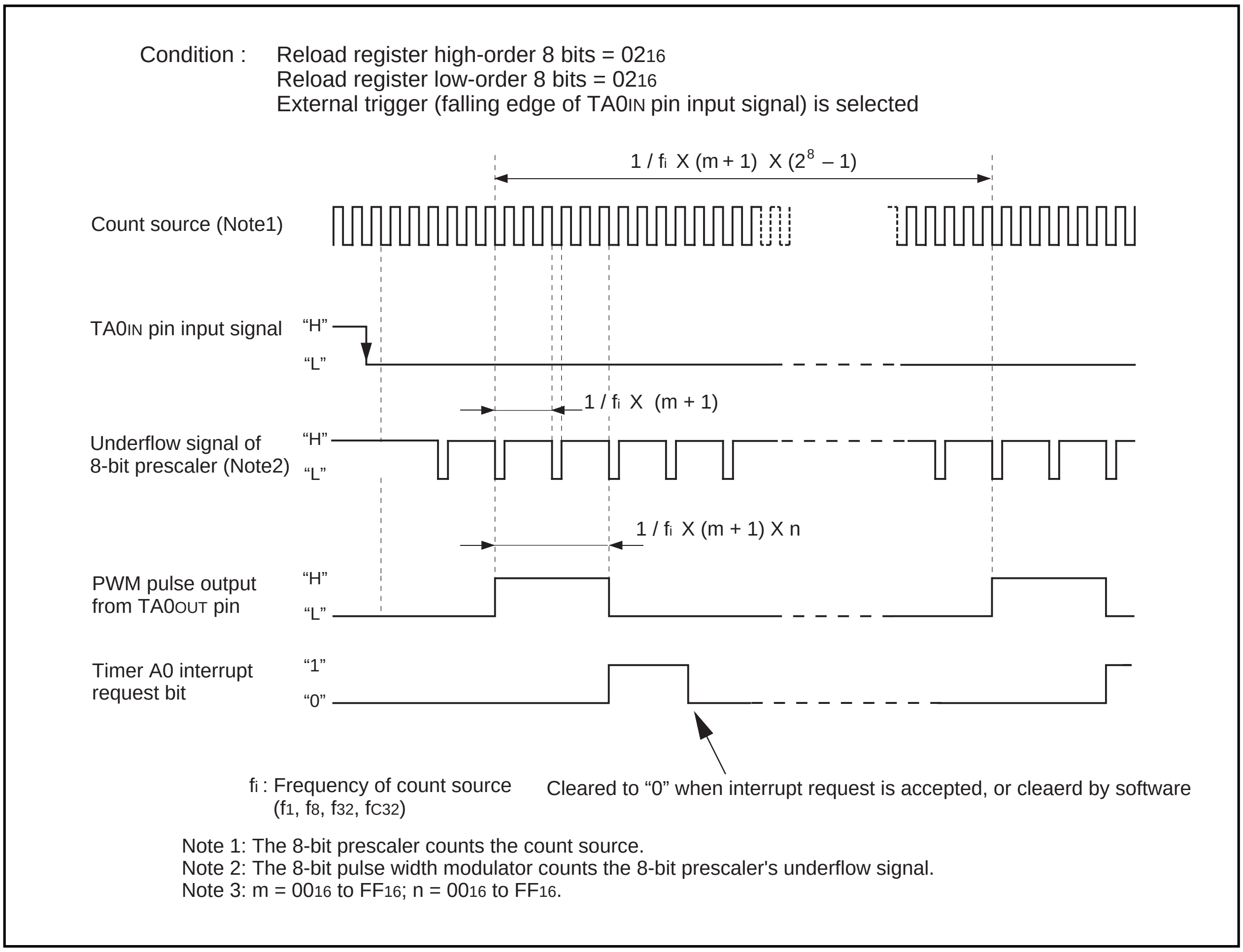

Condition : Reload register high-order 8 bits = 02₁₆
Reload register low-order 8 bits = 02₁₆
External trigger (falling edge of TA0IN pin input signal) is selected
1 / fi X (m + 1) X (2^8 – 1)
Count source (Note1)
TA0IN pin input signal
"H"
"L"
1 / fi X (m + 1)
Underflow signal of 8-bit prescaler (Note2)
"H"
"L"
1 / fi X (m + 1) X n
PWM pulse output from TA0OUT pin
"H"
"L"
Timer A0 interrupt request bit
"1"
"0"
fi : Frequency of count source (f1, f8, f32, fC32)
Cleared to "0" when interrupt request is accepted, or cleaerd by software
Note 1: The 8-bit prescaler counts the count source.
Note 2: The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
Note 3: m = 00₁₆ to FF₁₆; n = 00₁₆ to FF₁₆.

Figure 47. Example of how an 8-bit pulse width modulator operates

Timer B

Figure 48 shows the block diagram of timer B. Figures 49 and 50 show the timer B-related registers. Use the timer Bi mode register (i = 0, 1) 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.

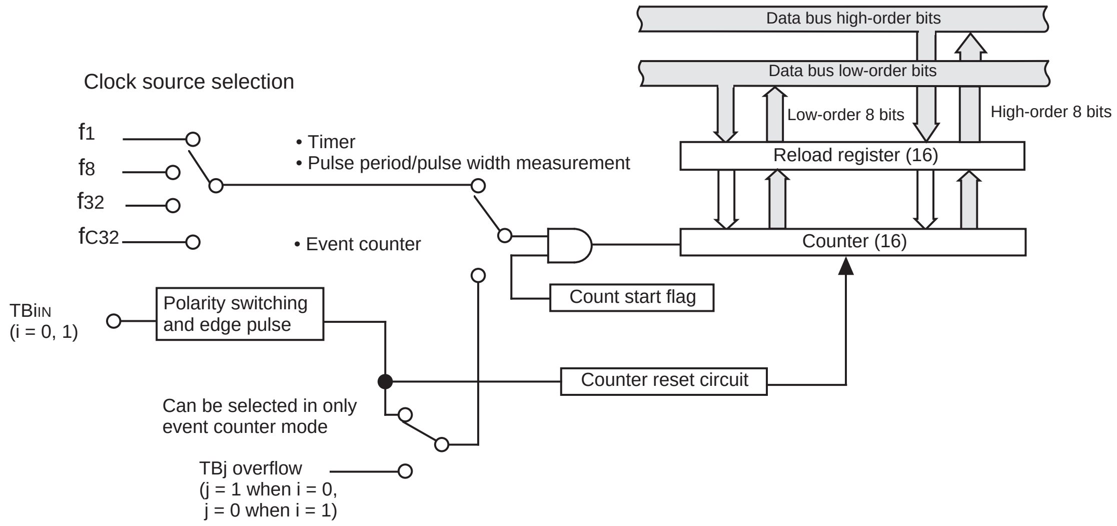


Figure 48. Block diagram of timer B

Timer Bi mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TBiMR(i = 0, 1)	039B16, 039C16	00XX00002

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/pulse width measurement mode 1 1 : Inhibited	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○ (Note 1) × (Note 2)	○ ×
MR3			○	×
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

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

Figure 49. Timer B-related registers (1)

Timer Bi register (Note)

Symbol	Address	When reset
TB0	0391_{16}, 0390_{16}	Indeterminate
TB1	0393_{16}, 0392_{16}	Indeterminate

(b15) b7 (b8) b0 b7 b0

Function	Values that can be set	R	W
• Timer mode Counts the timer's period	0000_{16} to $FFFF_{16}$	○	○
• Event counter mode Counts external pulses input or a timer overflow	0000_{16} to $FFFF_{16}$	○	○
• Pulse period / pulse width measurement mode Measures a pulse period or width	——	○	×

Note1: Read and write data in 16-bit units.

Count start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TABSR	0380_{16}	$000X0000_2$

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TX0S	Timer X0 count start flag		○	○
TX1S	Timer X1 count start flag		○	○
TX2S	Timer X2 count start flag		○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
TB0S	Timer B0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB1S	Timer B1 count start flag		○	○
CDCS	Clock devided count start flag		○	○

Clock prescaler reset flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
CPSRF	0381_{16}	$0XXXXXXX_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

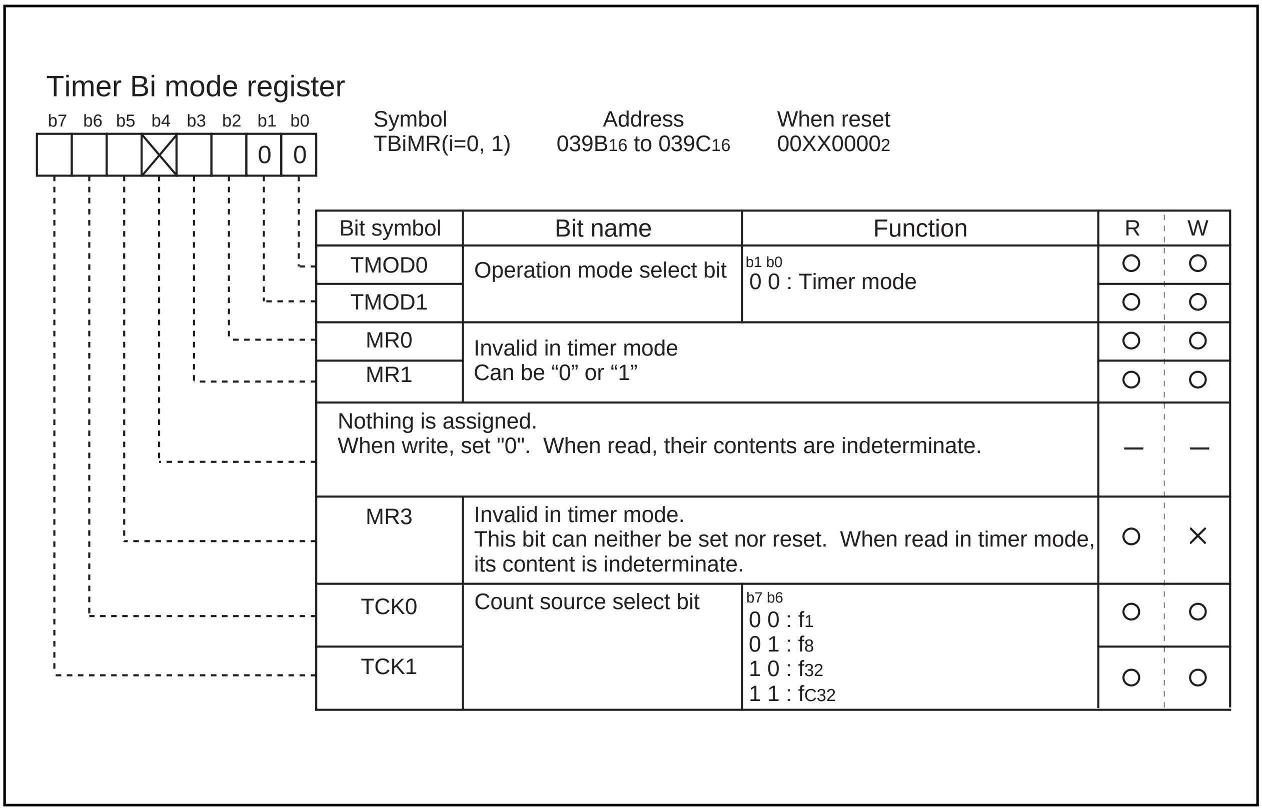
Figure 50. Timer B-related registers (2)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 18.) Figure 51 shows the timer Bi mode register in timer mode.

Table 18. Timer specifications in timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Counts down • When the timer underflows, it reloads the reload register contents before continuing counting
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
TBi_{IN} pin function	Programmable I/O port
Read from timer	Count value is read out by reading timer Bi register
Write to timer	• When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • 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)



Timer Bi mode register

Symbol	Address	When reset
TBiMR(i=0, 1)	$039B_{16}$ to $039C_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Invalid in timer mode Can be “0” or “1”		○	○
MR1			○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
MR3	Invalid in timer mode. This bit can neither be set nor reset. When read in timer mode, its content is indeterminate.		○	×
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

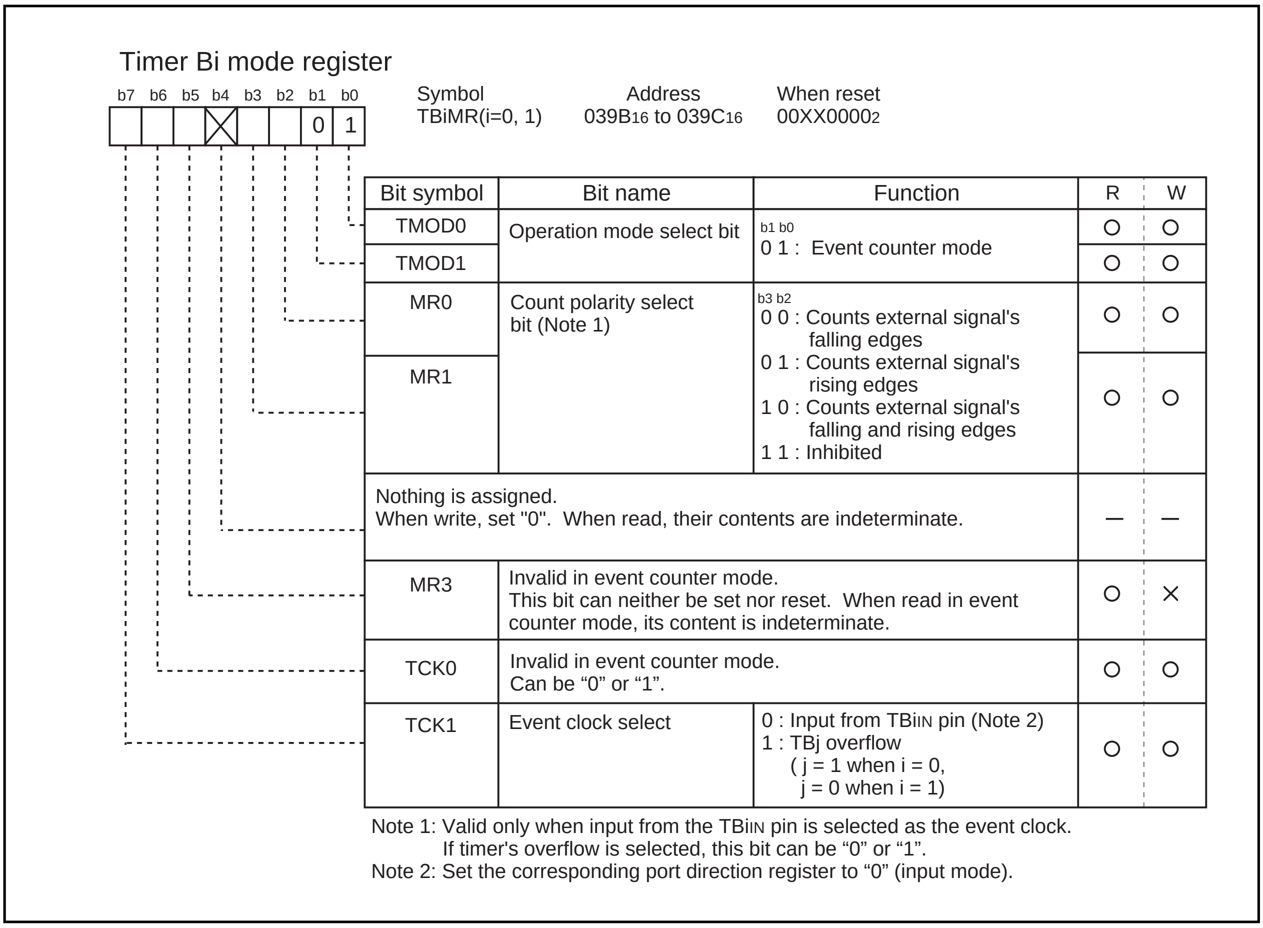
Figure 51. Timer Bi mode register in timer mode

(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table 19.) Figure 52 shows the timer Bi mode register in event counter mode.

Table 19. Timer specifications in event counter mode

Item	Specification
Count source	• External signals input to TBi$_{IN}$ pin • Effective edge of count source can be a rising edge, a falling edge, or falling and rising edges as selected by software
Count operation	• Counts down • When the timer underflows, it reloads the reload register contents before continuing counting
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
TBi$_{IN}$ pin function	Count source input
Read from timer	Count value can be read out by reading timer Bi register
Write to timer	• When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • 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)



Symbol	Address	When reset
TBiMR(i=0, 1)	039B$_{16}$ to 039C$_{16}$	00XX0000$_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	Count polarity select bit (Note 1)	b3 b2 0 0 : Counts external signal's falling edges 0 1 : Counts external signal's rising edges 1 0 : Counts external signal's falling and rising edges 1 1 : Inhibited	○	○
MR1			○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
MR3	Invalid in event counter mode. This bit can neither be set nor reset. When read in event counter mode, its content is indeterminate.		○	×
TCK0	Invalid in event counter mode. Can be “0” or “1”.		○	○
TCK1	Event clock select	0 : Input from TBi$_{IN}$ pin (Note 2) 1 : TBj overflow (j = 1 when i = 0, j = 0 when i = 1)	○	○

Note 1: Valid only when input from the TBi$_{IN}$ pin is selected as the event clock. If timer's overflow is selected, this bit can be “0” or “1”.
Note 2: Set the corresponding port direction register to “0” (input mode).

Figure 52. Timer Bi mode register in event counter mode

(3) Pulse period/pulse width measurement mode

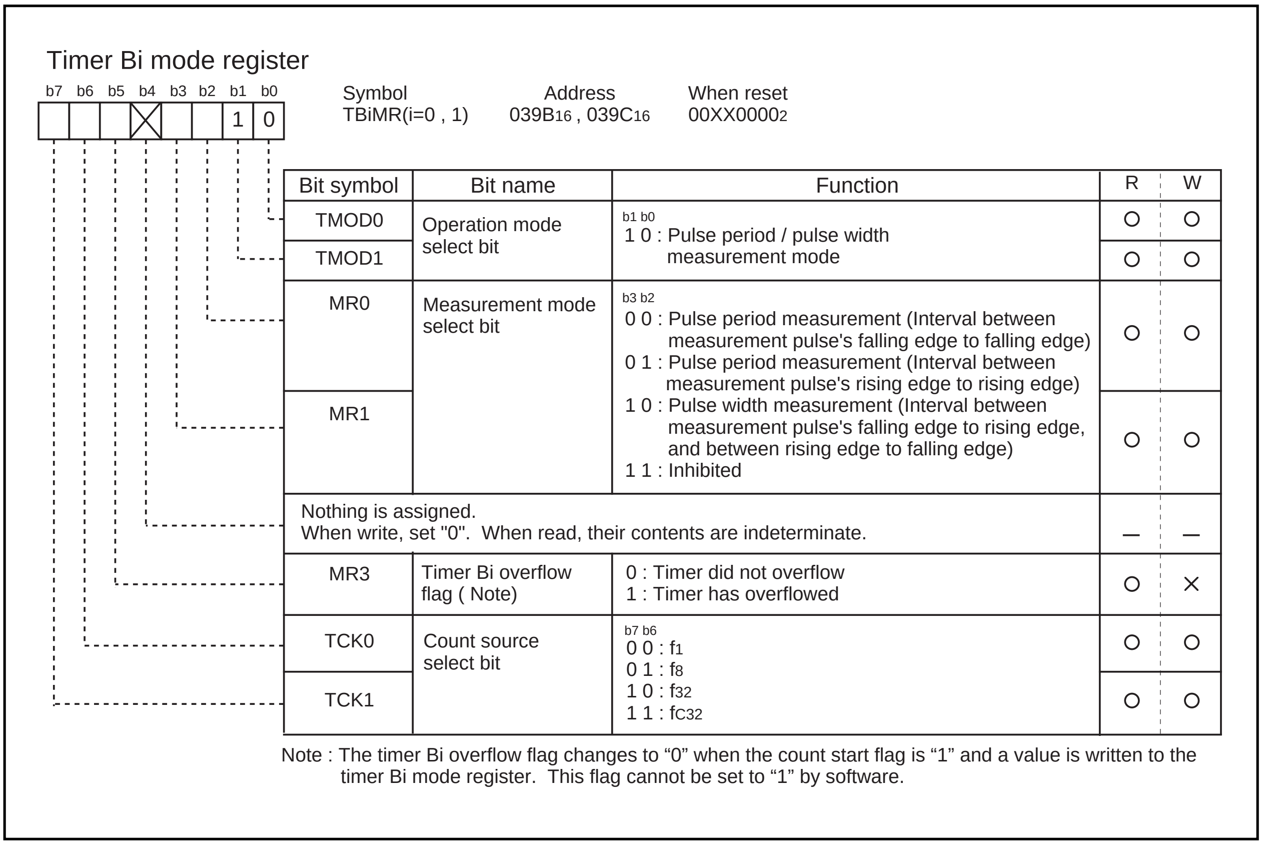
In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table 20.) Figure 53 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure 54 shows the operation timing when measuring a pulse period. Figure 55 shows the operation timing when measuring a pulse width.

Table 20. Timer specifications in pulse period/pulse width measurement mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Up count • Counter value “0000_{16}” is transferred to reload register at measurement pulse's effective edge and the timer continues counting
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	• When measurement pulse's effective edge is input (Note 1) • 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.)
TBi_{IN} 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.



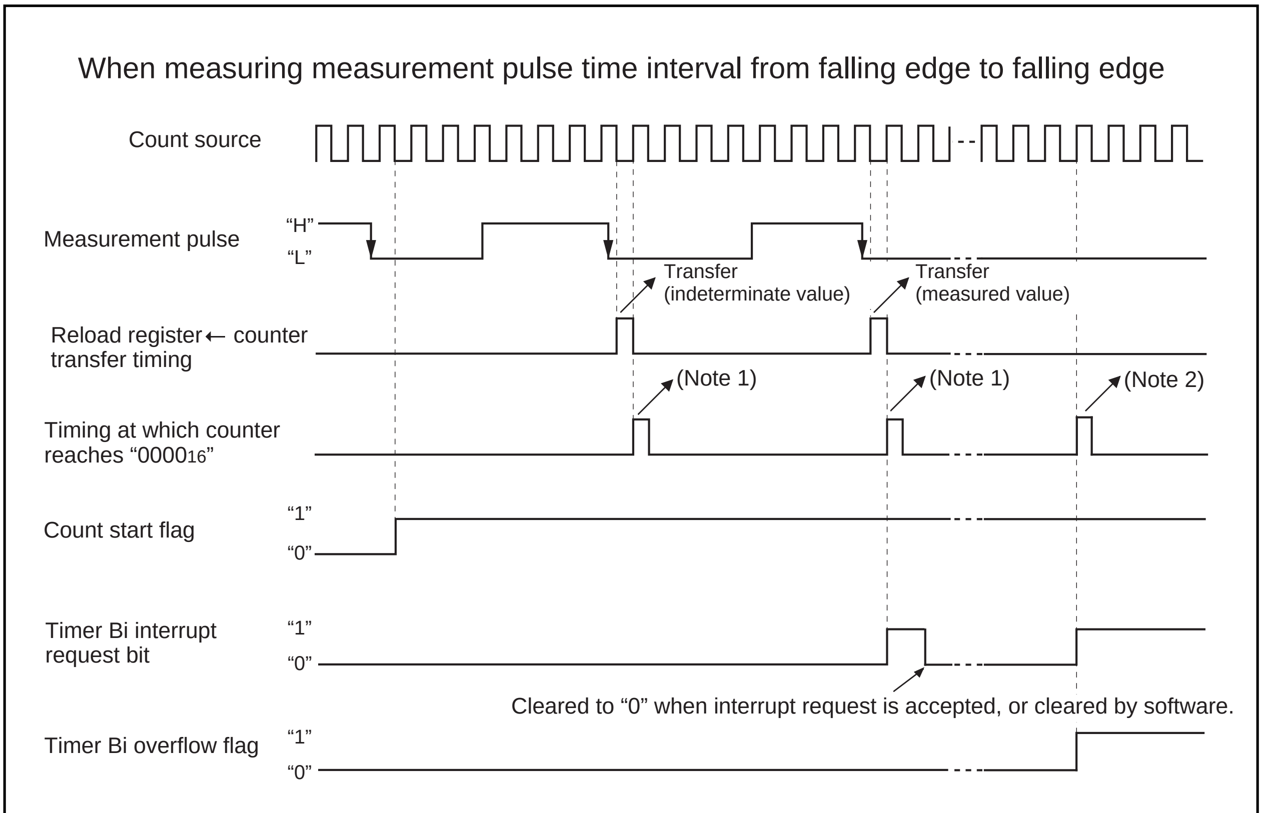
Timer Bi mode register

Symbol: TBiMR(i=0 , 1) | Address: 039B$_{16}$, 039C$_{16}$ | When reset: 00XX0000$_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : Pulse period / pulse width measurement mode	○	○
TMOD1			○	○
MR0	Measurement mode select bit	b3 b2 0 0 : Pulse period measurement (Interval between measurement pulse's falling edge to falling edge) 0 1 : Pulse period measurement (Interval between measurement pulse's rising edge to rising edge) 1 0 : Pulse width measurement (Interval between measurement pulse's falling edge to rising edge, and between rising edge to falling edge) 1 1 : Inhibited	○	○
MR1			○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
MR3	Timer Bi overflow flag (Note)	0 : Timer did not overflow 1 : Timer has overflowed	○	×
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

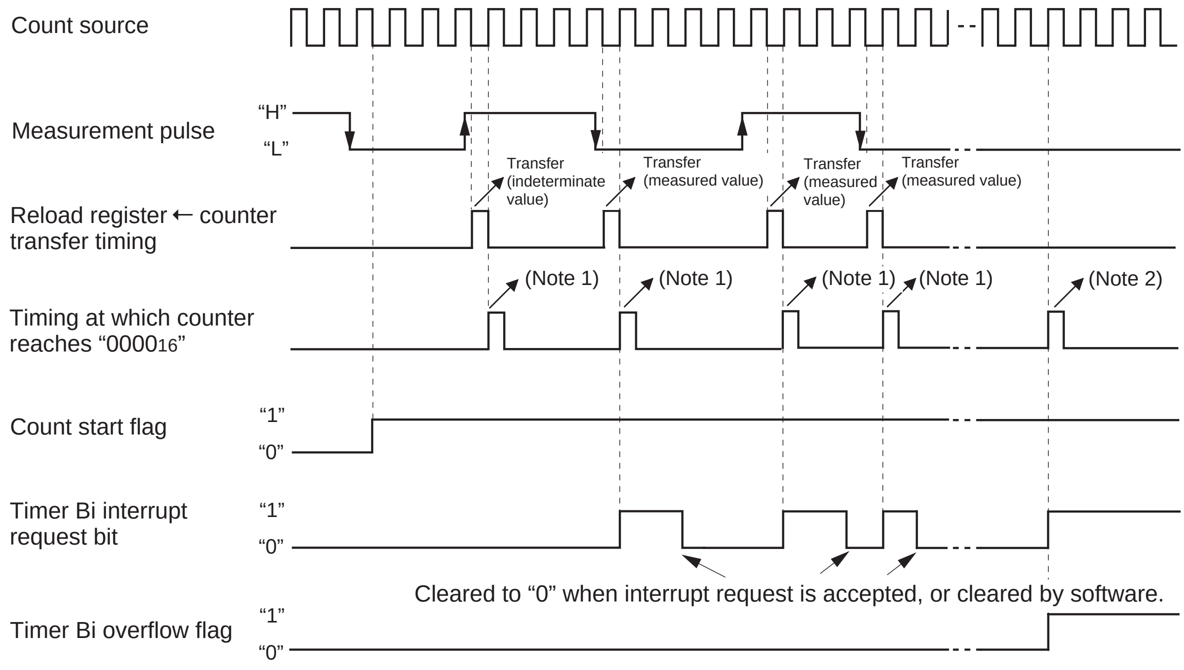
Note : The timer Bi overflow flag changes to “0” when the count start flag is “1” and a value is written to the timer Bi mode register. This flag cannot be set to “1” by software.

Figure 53. Timer Bi mode register in pulse period/pulse width measurement mode



Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 54. Operation timing when measuring a pulse period



Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 55. Operation timing when measuring a pulse width

Timer X

Figure 56 shows the block diagram of timer X. Figures 57 to 59 show the timer X-related registers.
Use the timer Xi mode register bits 0 and 1 to choose the desired mode.
Timer X has the five 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.
- One-shot timer mode : The timer stops counting when the count reaches “0000$_{16}$”.
- Pulse period/pulse width measuring mode : The timer measures an external signal's pulse period or pulse width.
- Pulse width modulation (PWM) mode : The timer outputs pulses of a given width.

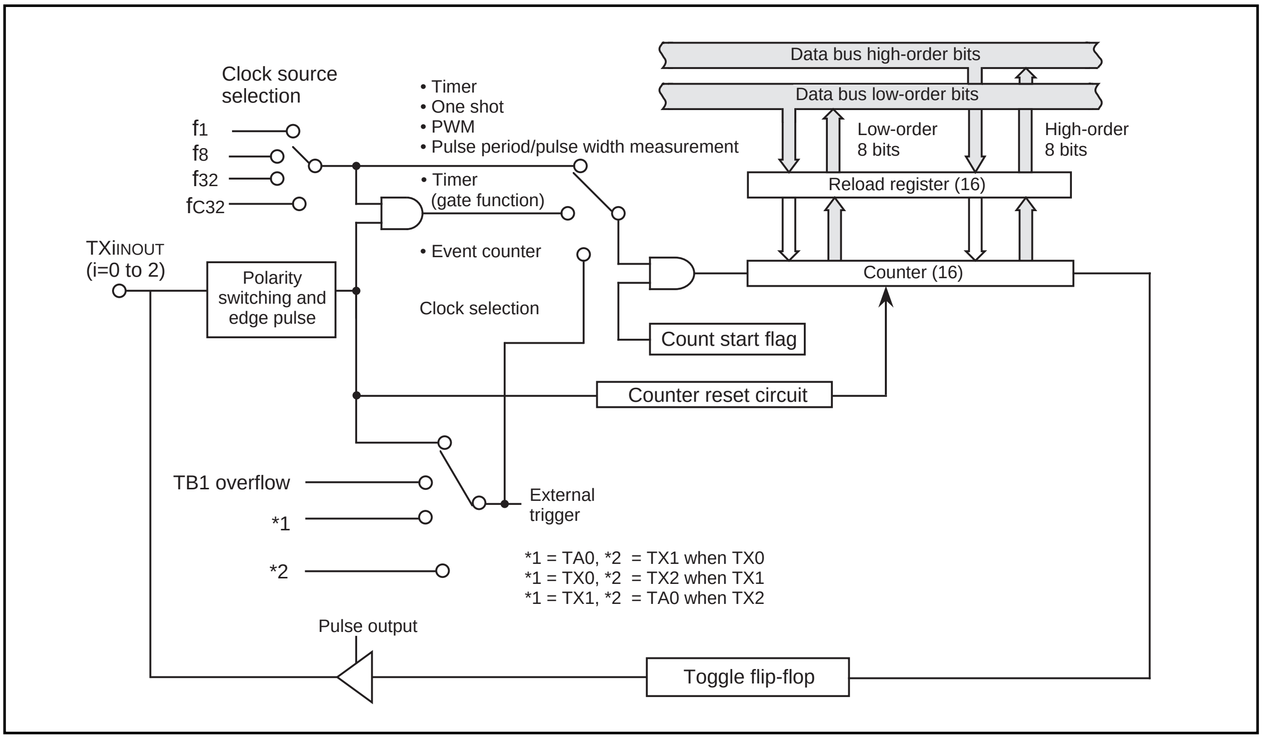


Figure 56. Block diagram of timer X

Timer Xi mode register

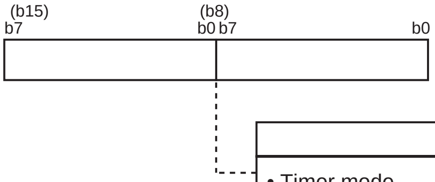
b7 b6 b5 b4 b3 b2 b1 b0

Symbol TXiMR(i = 0 to 2) Address 0397$_{16}$ to 0399$_{16}$ When reset 00$_{16}$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode or pulse period/ pulse width measurement mode 1 1 : Pulse width modulation (PWM) mode	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○	○
MR3			○	○
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Figure 57. Timer X-related registers (1)

Timer Xi register (Note)

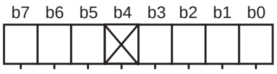


Symbol	Address	When reset
TX0	0389_{16},0388_{16}	Indeterminate
TX1	$038B_{16}$,$038A_{16}$	Indeterminate
TX2	$038D_{16}$,$038C_{16}$	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	0000_{16} to $FFFF_{16}$	○	○
• Event counter mode Counts pulses from an external source or timer overflow	0000_{16} to $FFFF_{16}$	○	○
• One-shot timer mode Counts a one shot width	0000_{16} to $FFFF_{16}$	×	○
• Pulse period / pulse width measurement mode Measures a pulse period or width	———	○	×
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000_{16} to $FFFE_{16}$	×	○
• Pulse width modulation mode (8-bit PWM) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	00_{16} to FF_{16} (High-order addresses) 00_{16} to FF_{16} (Low-order addresses)	×	○

Note: Read and write data in 16-bit units.

Count start flag



Symbol	Address	When reset
TABSR	0380_{16}	$000X0000_2$

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TX0S	Timer X0 count start flag		○	○
TX1S	Timer X1 count start flag		○	○
TX2S	Timer X2 count start flag		○	○
Nothing is assigned. When write, set "0" When read, their contents are indeterminate.			—	—
TB0S	Timer B0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB1S	Timer B1 count start flag		○	○
CDCS	Clock devided count start flag		○	○

Figure 58. Timer X-related registers (2)

One-shot start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
ONSF	0382$_{16}$	XXXX0000$_2$

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is “0”	○	○
TX0OS	Timer X0 one-shot start flag		○	○
TX1OS	Timer X1 one-shot start flag		○	○
TX2OS	Timer X2 one-shot start flag		○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.			—	—

Trigger select register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TRGSR	0383$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TA0TGL	Timer A0 event/trigger select bit	b1 b0 0 0 : Input on TA0$_{IN}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX2 overflow is selected 1 1 : TX0 overflow is selected	○	○
TA0TGH			○	○
TX0TGL	Timer X0 event/trigger select bit	b3 b2 0 0 : Input on TX0$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TX1 overflow is selected	○	○
TX0TGH			○	○
TX1TGL	Timer X1 event/trigger select bit	b5 b4 0 0 : Input on TX1$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX0 overflow is selected 1 1 : TX2 overflow is selected	○	○
TX1TGH			○	○
TX2TGL	Timer X2 event/trigger select bit	b7 b6 0 0 : Input on TX2$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX1 overflow is selected 1 1 : TA0 overflow is selected	○	○
TX2TGH			○	○

Note: Set the corresponding port direction register to “0”(input mode).

Clock prescaler reset flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
CPSRF	0381$_{16}$	0XXXXXXX$_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

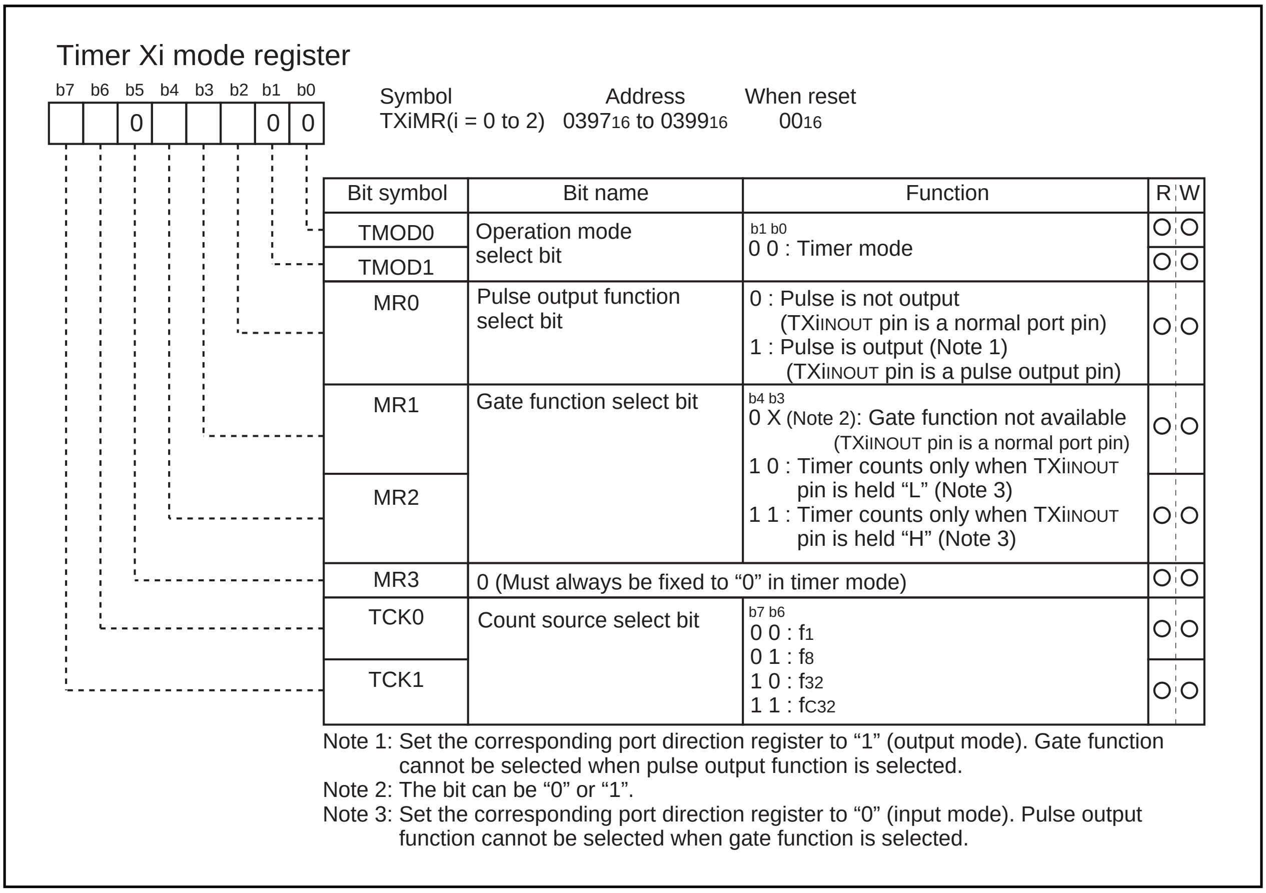
Figure 59. Timer X-related registers (3)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 21.) Figure 60 shows the timer Xi mode register in timer mode.

Table 21. Specifications of timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
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
TXiINOUT pin function	Programmable I/O port, gate input or pulse output
Read from timer	Count value can be read out by reading timer Xi register
Write to timer	• When counting stopped When a value is written to timer Xi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Xi register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Gate function Counting can be started and stopped by the TXiINOUT pin's input signal • Pulse output function Each time the timer underflows, the TXiINOUT pin's polarity is reversed



Timer Xi mode register

Symbol	Address	When reset
TXiMR(i = 0 to 2)	0397₁₆ to 0399₁₆	00₁₆

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TXiINOUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TXiINOUT pin is a pulse output pin)	○	○
MR1	Gate function select bit	b4 b3 0 X (Note 2): Gate function not available (TXiINOUT pin is a normal port pin) 1 0 : Timer counts only when TXiINOUT pin is held "L" (Note 3) 1 1 : Timer counts only when TXiINOUT pin is held "H" (Note 3)	○	○
MR2			○	○
MR3	0 (Must always be fixed to "0" in timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Set the corresponding port direction register to "1" (output mode). Gate function cannot be selected when pulse output function is selected.
Note 2: The bit can be "0" or "1".
Note 3: Set the corresponding port direction register to "0" (input mode). Pulse output function cannot be selected when gate function is selected.

Figure 60. Timer Xi mode register in timer mode

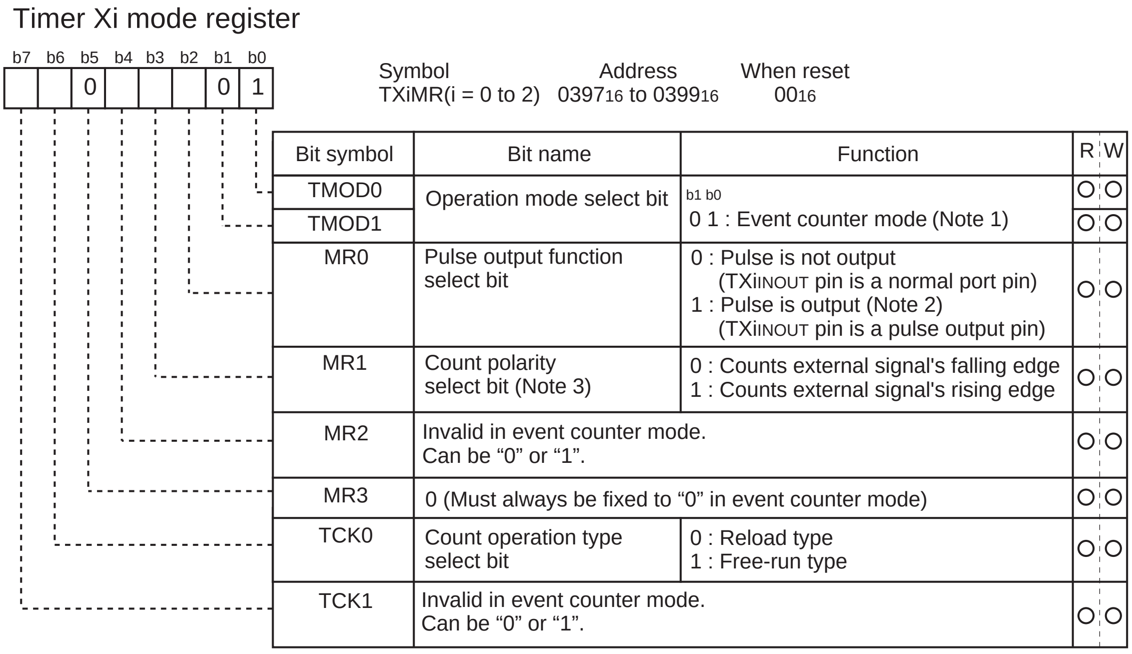
(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table 22.) Figure 61 shows the timer Xi mode register in event counter mode.

Table 22. Timer specifications in event counter mode (when not processing two-phase pulse signal)

Item	Specification
Count source	• External signals input to TXiINOUT pin (effective edge can be selected by software) • TB1 overflow, TA0 overflow, TXi overflow
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting (Note)
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
TXiINOUT pin function	Programmable I/O port, count source input or pulse output
Read from timer	Count value can be read out by reading timer Xi register
Write to timer	• When counting stopped When a value is written to timer Xi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Xi register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Free-run count function Even when the timer underflows, the reload register content is not reloaded to it • Pulse output function Each time the timer underflows, the TXiINOUT pin's polarity is reversed

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



Symbol TXiMR(i = 0 to 2) Address 0397_{16} to 0399_{16} When reset 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode (Note 1)	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TXiINOUT pin is a normal port pin) 1 : Pulse is output (Note 2) (TXiINOUT pin is a pulse output pin)	○	○
MR1	Count polarity select bit (Note 3)	0 : Counts external signal's falling edge 1 : Counts external signal's rising edge	○	○
MR2	Invalid in event counter mode. Can be "0" or "1".		○	○
MR3	0 (Must always be fixed to "0" in event counter mode)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Invalid in event counter mode. Can be "0" or "1".		○	○

Note 1: Count source is selected by event/trigger select bit(address 0383_{16}) in event counter mode.
Note 2: Set the corresponding port direction register to "1" (output mode). TXiINOUT pin input is not selected as count source when pulse output function is selected.
Note 3: This bit is valid when only counting an external signal.

Figure 61. Timer Xi mode register in event counter mode

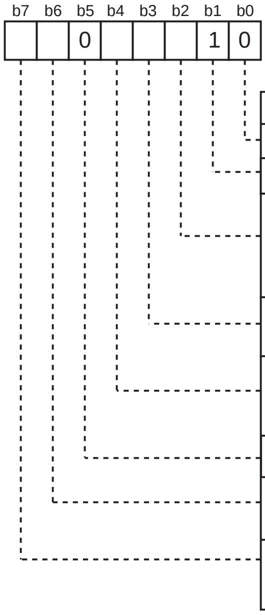
(3) One-shot timer mode

In this mode, the timer operates only once. (See Table 23.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 62 shows the timer Xi mode register in one-shot timer mode.

Table 23. Timer specifications in one-shot timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, fc_{32}
Count operation	• The timer counts down • When the count reaches 0000_{16}, the timer stops counting after reloading a new count • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : Set value
Count start condition	• An external trigger is input • The timer overflows • The one-shot start flag is set (= 1)
Count stop condition	• A new count is reloaded after the count has reached 0000_{16} • The count start flag is reset (= 0)
Interrupt request generation timing	The count reaches 0000_{16}
TX$_{\mathrm{iINOUT}}$ pin function	Programmable I/O port, trigger input or pulse output
Read from timer	When timer Xi register is read, it indicates an indeterminate value
Write to timer	• When counting stopped When a value is written to timer Xi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Xi register, it is written to only reload register (Transferred to counter at next reload time)

Timer Xi mode register



Symbol TXiMR(i = 0 to 2)
Address 0397_{16} to 0399_{16}
When reset 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode or pulse period / pulse width measurement mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TX$_{\mathrm{iINOOUT}}$ pin is a normal port pin) 1 : Pulse is output (Note 1) (TX$_{\mathrm{iINOOUT}}$ pin is a pulse output pin)	○	○
MR1	External trigger select bit (Note 2)	0 : Falling edge of TX$_{\mathrm{iINOOUT}}$ pin's input signal (Note 3) 1 : Rising edge of TX$_{\mathrm{iINOOUT}}$ pin's input signal (Note 3)	○	○
MR2	Trigger select bit	0 : One-shot start flag is valid 1 : Selected by event/trigger select register (Note 4)	○	○
MR3	0 (Must always be “0” in one-shot timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	○	○
TCK1			○	○

Note 1: Set the corresponding port direction register to “1” (output mode). External trigger cannot be selected as count start condition when pulse output function is selected.
Note 2: Valid only when the TX$_{\mathrm{iINOUT}}$ pin is selected by the event/trigger select bit (addresses 0383_{16}). If timer overflow is selected, this bit can be “1” or “0”.
Note 3: Set the corresponding port direction register to “0” (input mode).
Note 4: Pulse output function cannot be selected when TX$_{\mathrm{iINOUT}}$ pin is selected by the event/trigger select bit (addresses 0383_{16}).

Figure 62. Timer Xi mode register in one-shot timer mode

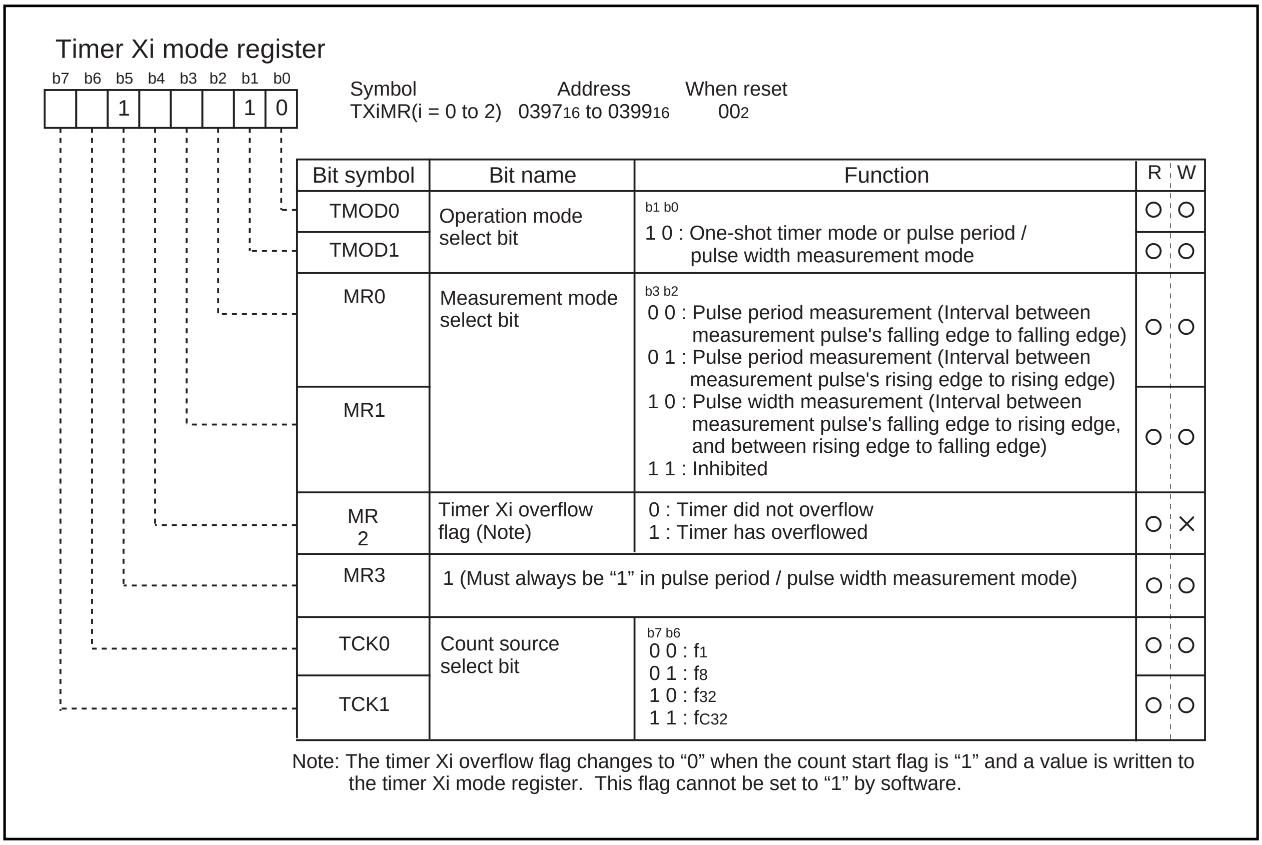
(4) Pulse period/pulse width measurement mode

In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table 24.) Figure 63 shows the timer Xi mode register in pulse period/pulse width measurement mode. Figure 64 shows the operation timing when measuring a pulse period. Figure 65 shows the operation timing when measuring a pulse width.

Table 24. Timer specifications in pulse period/pulse width measurement mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Up count • Counter value “0000_{16}” is transferred to reload register at measurement pulse's effective edge and the timer continues counting
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	• When measurement pulse's effective edge is input (Note 1) • When an overflow occurs. (Simultaneously, the timer Xi overflow flag changes to “1”. The timer Xi overflow flag changes to “0” when the count start flag is “1” and a value is written to the timer Xi mode register.)
TXi_{INOUT} pin function	Measurement pulse input
Read from timer	When timer Xi 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 Xi register is indeterminate until the second effective edge is input after the timer.



Symbol	Address	When reset
TXiMR(i = 0 to 2)	0397_{16} to 0399_{16}	00_2

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode or pulse period / pulse width measurement mode	○	○
TMOD1			○	○
MR0	Measurement mode select bit	b3 b2 0 0 : Pulse period measurement (Interval between measurement pulse's falling edge to falling edge) 0 1 : Pulse period measurement (Interval between measurement pulse's rising edge to rising edge) 1 0 : Pulse width measurement (Interval between measurement pulse's falling edge to rising edge, and between rising edge to falling edge) 1 1 : Inhibited	○	○
MR1			○	○
MR 2	Timer Xi overflow flag (Note)	0 : Timer did not overflow 1 : Timer has overflowed	○	×
MR3	1 (Must always be “1” in pulse period / pulse width measurement mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note: The timer Xi overflow flag changes to “0” when the count start flag is “1” and a value is written to the timer Xi mode register. This flag cannot be set to “1” by software.

Figure 63. Timer Xi mode register in pulse period/pulse width measurement mode

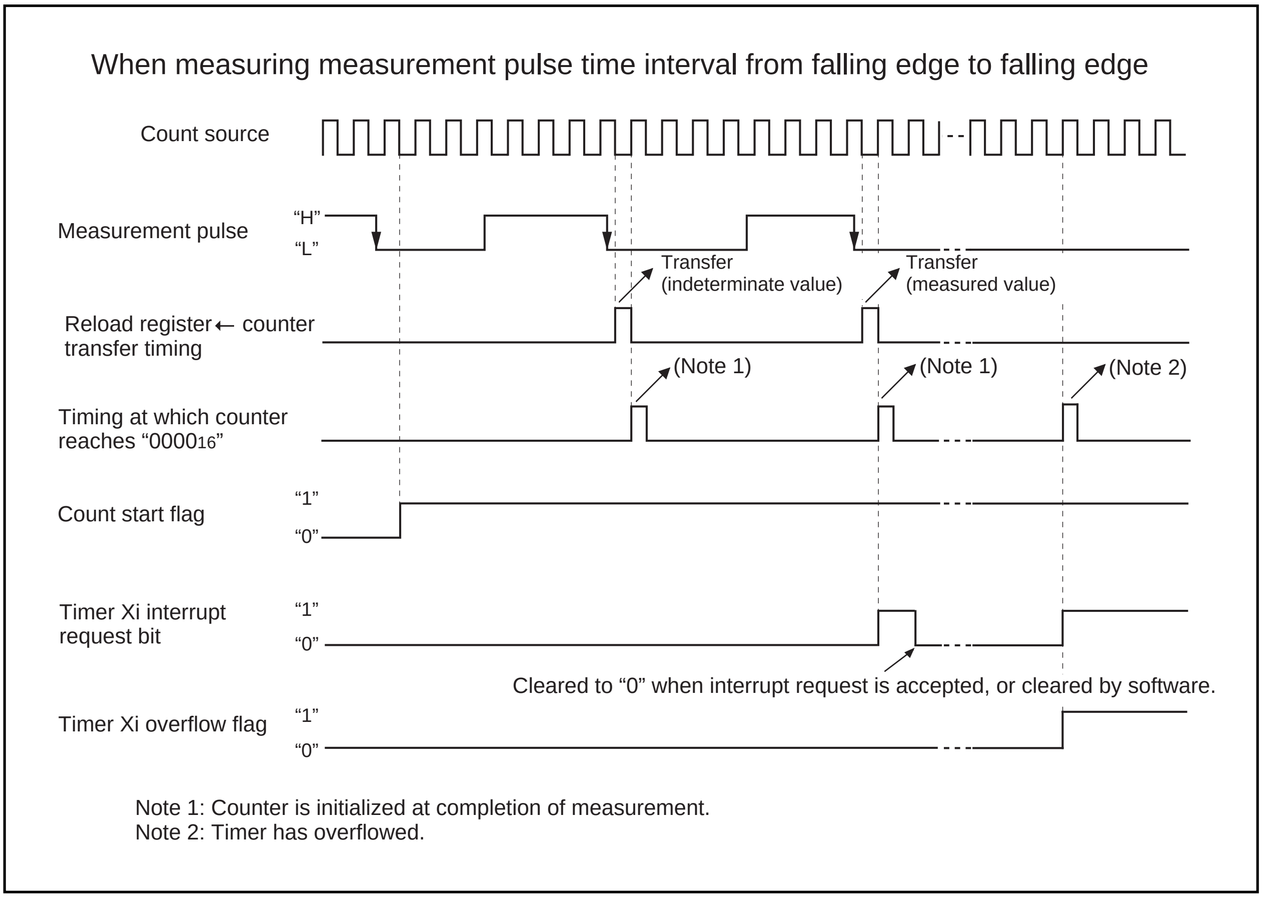

When measuring measurement pulse time interval from falling edge to falling edge
Count source
Measurement pulse
"H"
"L"
Transfer (indeterminate value)
Transfer (measured value)
Reload register ← counter transfer timing
(Note 1)
(Note 1)
(Note 2)
Timing at which counter reaches "0000₁₆"
Count start flag
"1"
"0"
Timer Xi interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software.
Timer Xi overflow flag
"1"
"0"
Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 64. Operation timing when measuring a pulse period

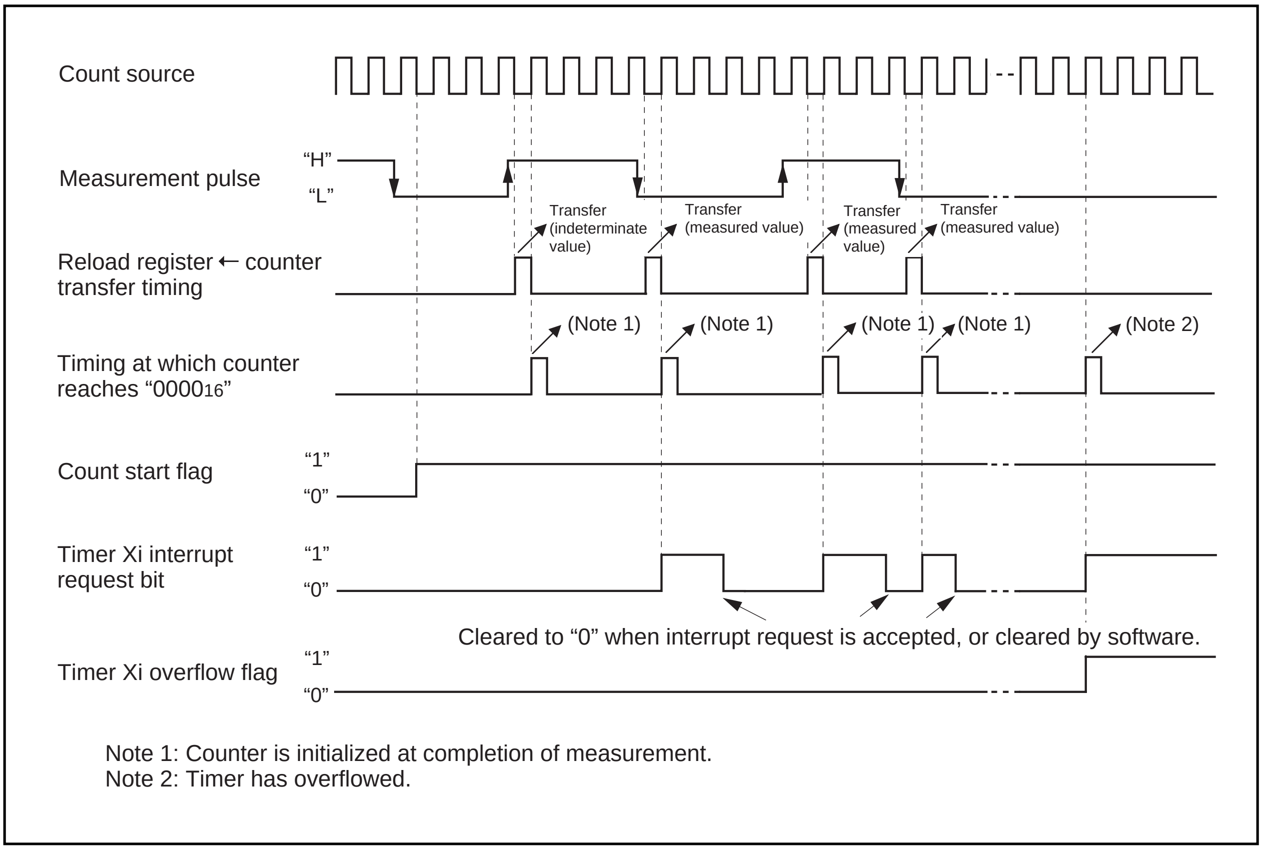

Count source
Measurement pulse
"H"
"L"
Transfer (indeterminate value)
Transfer (measured value)
Transfer (measured value)
Transfer (measured value)
Reload register ← counter transfer timing
(Note 1)
(Note 1)
(Note 1)
(Note 1)
(Note 2)
Timing at which counter reaches "0000₁₆"
Count start flag
"1"
"0"
Timer Xi interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software.
Timer Xi overflow flag
"1"
"0"
Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 65. Operation timing when measuring a pulse width

(5) Pulse width modulation (PWM) mode

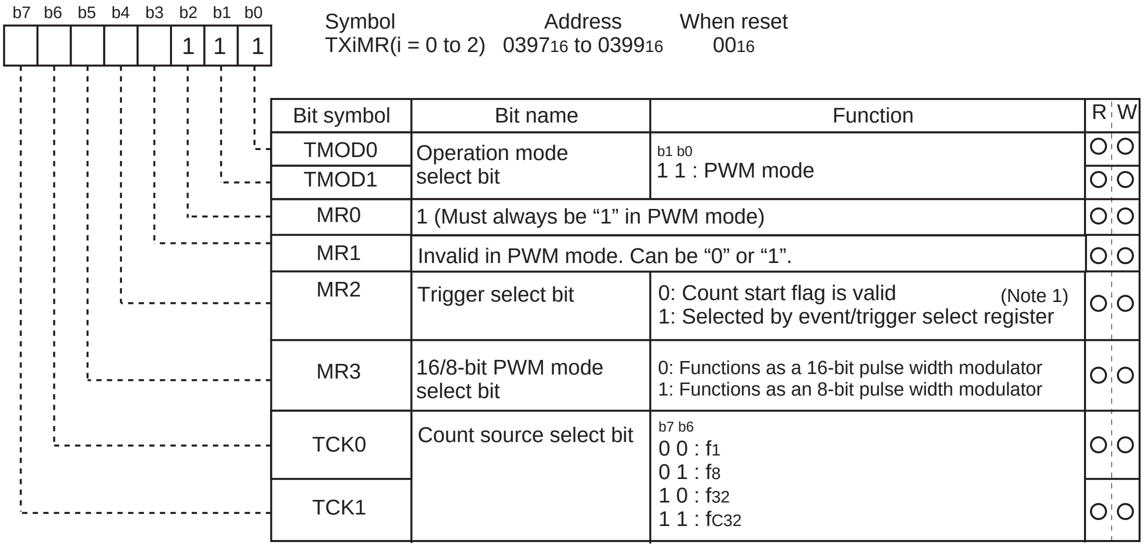
In this mode, the timer outputs pulses of a given width in succession. (See Table 25.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 66 shows the timer Xi mode register in pulse width modulation mode. Figure 67 shows the example of how a 16-bit pulse width modulator operates. Figure 68 shows the example of how an 8-bit pulse width modulator operates.

Table 25. Timer specifications in pulse width modulation mode

Item		Specification
Count source		f1, f8, f32, fc32
Count operation		• Down counts (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new count at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs when counting
16-bit PWM		• "H" level width n / fi n : Set value • Cycle time $(2^{16}-1)$ / fi fixed
8-bit PWM		• "H" level width n×(m+1)/ fi n:values set to timer Xi register's high-order address • Cycle time (2^8-1)×(m+1) / fi m : values set to timer Xi register's low-order address
Count start condition		• The timer overflows • The count start flag is set (= 1)
Count stop condition		• The count start flag is reset (= 0)
Interrupt request generation timing	8 bits PWM	• Set value of "H" level width is except FF$_{16}$, 00$_{16}$: PWM pulse goes "L" • Set value of "H" level width is FF$_{16}$, 00$_{16}$: Timing that count value goes to 01$_{16}$
	16 bits PWM	• Set value of "H" level width is except FFFF$_{16}$, 0000$_{16}$: PWM pulse goes "L" • Set value of "H" level width is FFFF$_{16}$, 0000$_{16}$: Timing that count value goes to 0001$_{16}$
TXiINOUT pin function		Pulse output
Read from timer		When timer Xi register is read, it indicates an indeterminate value
Write to timer		• When counting stopped When a value is written to timer Xi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Xi register, it is written to only reload register (Transferred to counter at next reload time)

Note: When set value of "H" level width is 00$_{16}$ or 0000$_{16}$, pulse outputs "L" level and inversion value, FF$_{16}$ or FFFF$_{16}$ is set to timer.

Timer Xi mode register



Symbol	Address	When reset
TXiMR(i = 0 to 2)	0397$_{16}$ to 0399$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 1 : PWM mode	○	○
TMOD1			○	○
MR0	1 (Must always be "1" in PWM mode)		○	○
MR1	Invalid in PWM mode. Can be "0" or "1".		○	○
MR2	Trigger select bit	0: Count start flag is valid (Note 1) 1: Selected by event/trigger select register	○	○
MR3	16/8-bit PWM mode select bit	0: Functions as a 16-bit pulse width modulator 1: Functions as an 8-bit pulse width modulator	○	○
TCK0	Count source select bit	b7 b6 0 0 : f$_1$ 0 1 : f$_8$ 1 0 : f$_{32}$ 1 1 : f$_{C32}$	○	○
TCK1			○	○

Note 1: TXiINOUT pin inout cannot be selected by the event/trigger select bit(addresses 0383$_{16}$).
Note 2: Set the corresponding port direction register to "1" (output mode).

Figure 66. Timer Xi mode register in pulse width modulation mode

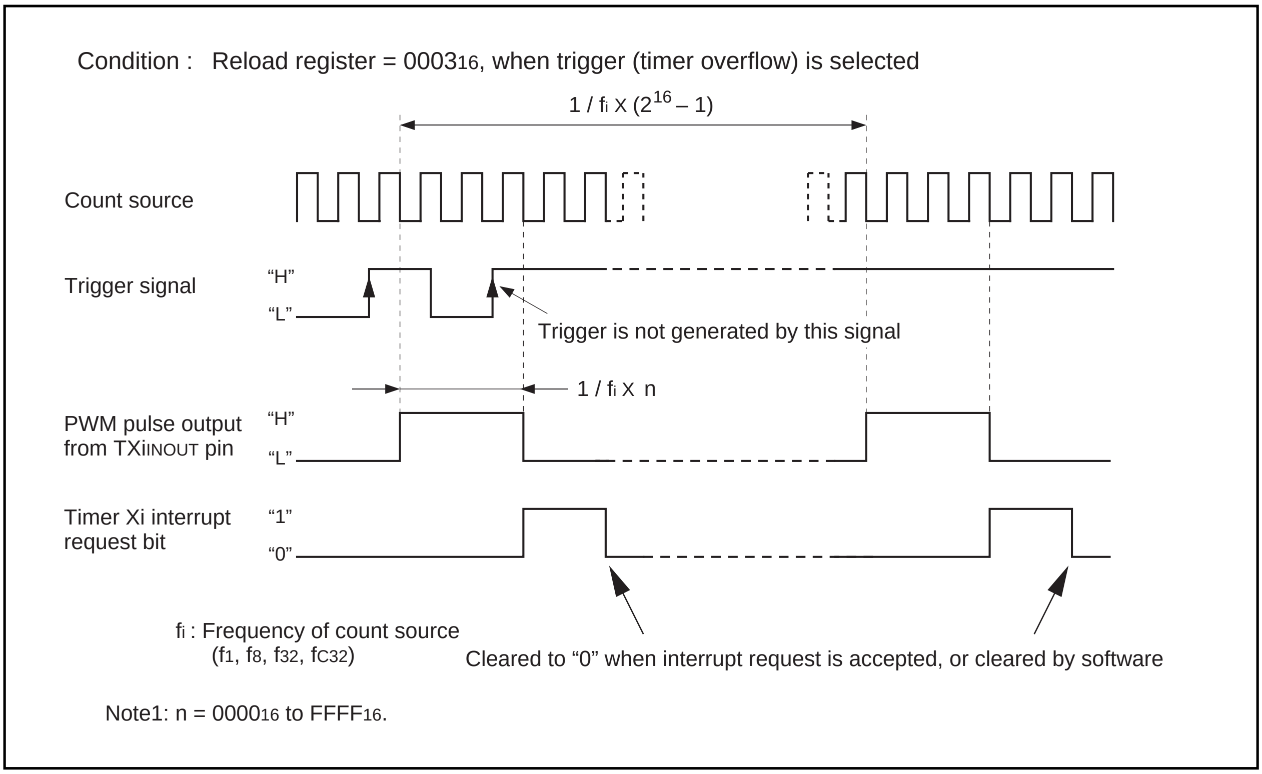

Condition : Reload register = 0003$_{16}$, when trigger (timer overflow) is selected
1 / fi X (2^{16} – 1)
Count source
Trigger signal
"H"
"L"
Trigger is not generated by this signal
1 / fi X n
PWM pulse output from TXiINOUT pin
"H"
"L"
Timer Xi interrupt request bit
"1"
"0"
fi : Frequency of count source (f1, f8, f32, fC32)
Cleared to "0" when interrupt request is accepted, or cleared by software
Note1: n = 0000$_{16}$ to FFFF$_{16}$.

Figure 67. Example of how a 16-bit pulse width modulator operates

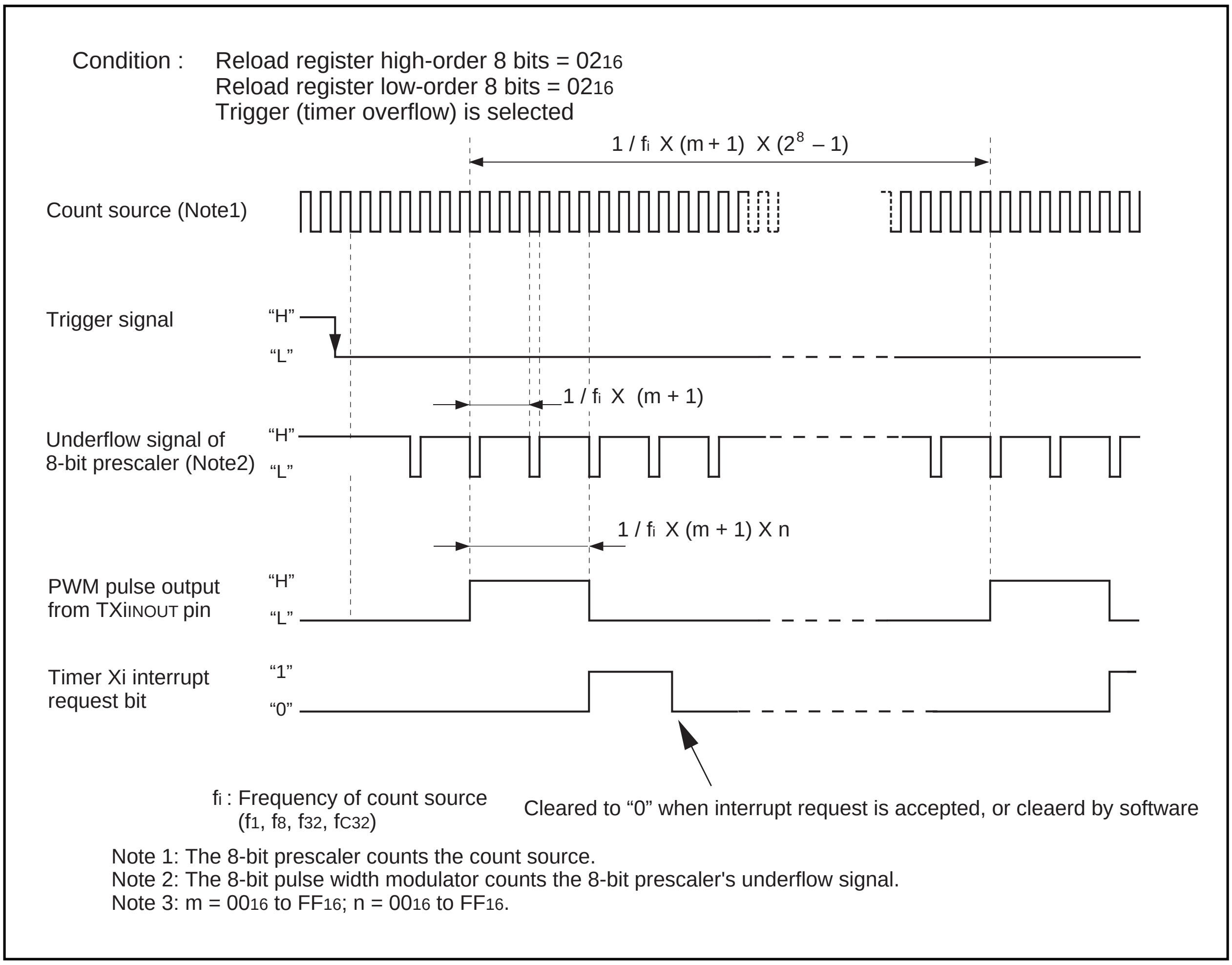

Condition : Reload register high-order 8 bits = 02$_{16}$
Reload register low-order 8 bits = 02$_{16}$
Trigger (timer overflow) is selected
1 / fi X (m + 1) X (2^{8} – 1)
Count source (Note1)
Trigger signal
"H"
"L"
1 / fi X (m + 1)
Underflow signal of 8-bit prescaler (Note2)
"H"
"L"
1 / fi X (m + 1) X n
PWM pulse output from TXiINOUT pin
"H"
"L"
Timer Xi interrupt request bit
"1"
"0"
fi : Frequency of count source (f1, f8, f32, fC32)
Cleared to "0" when interrupt request is accepted, or cleaerd by software
Note 1: The 8-bit prescaler counts the count source.
Note 2: The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
Note 3: m = 00$_{16}$ to FF$_{16}$; n = 00$_{16}$ to FF$_{16}$.

Figure 68. Example of how an 8-bit pulse width modulator operates

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 69 shows the block diagram of UART0 and UART1. Figures 70 shows the block diagram of the transmit/receive unit.

UART0 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$_{16}$ and 03A8$_{16}$) determine whether UART0 is used as a clock synchronous serial I/O or as a UART.

UART1 is used as a UART only.

Figures 71 through 73 show the registers related to UARTi.

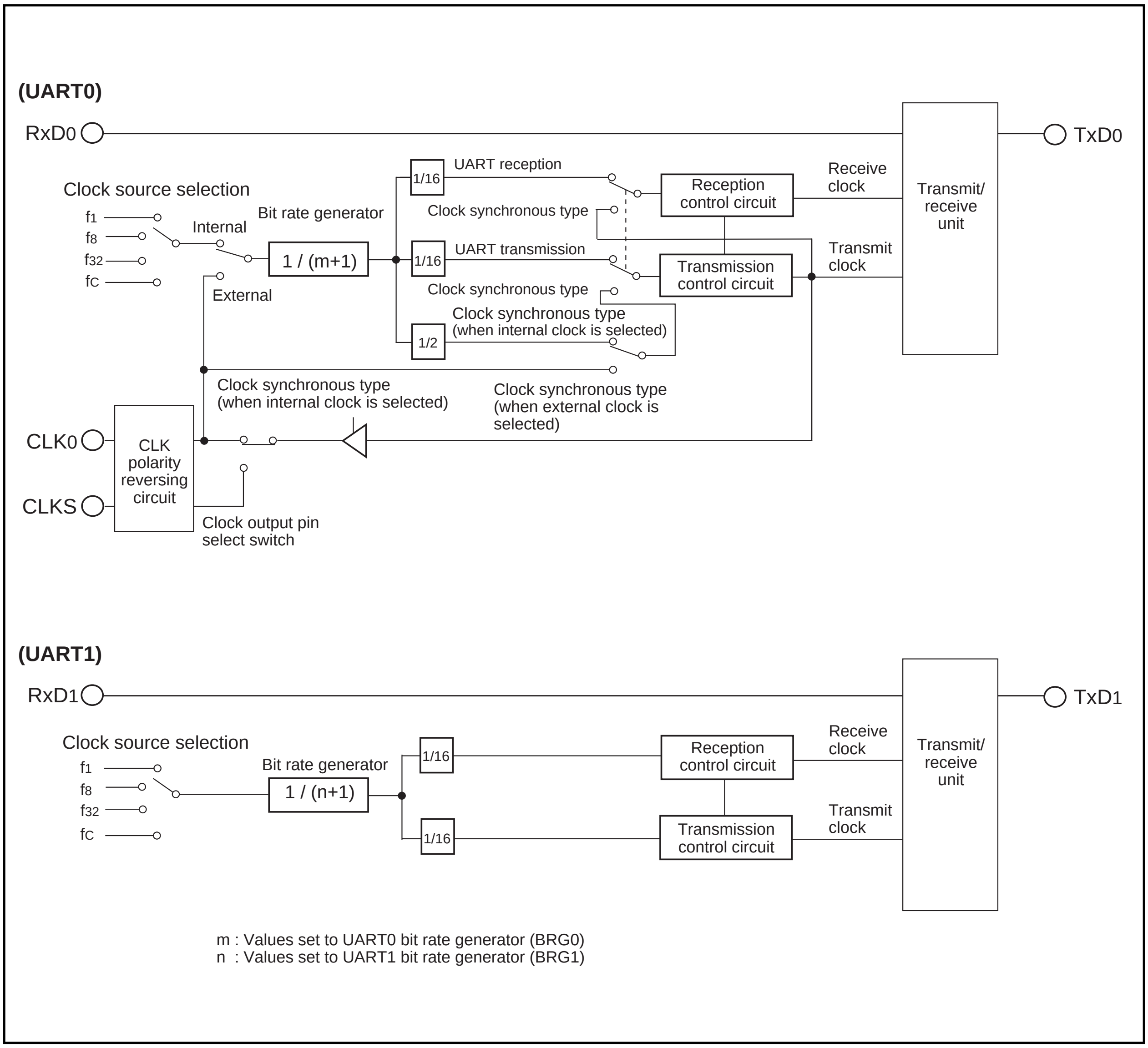
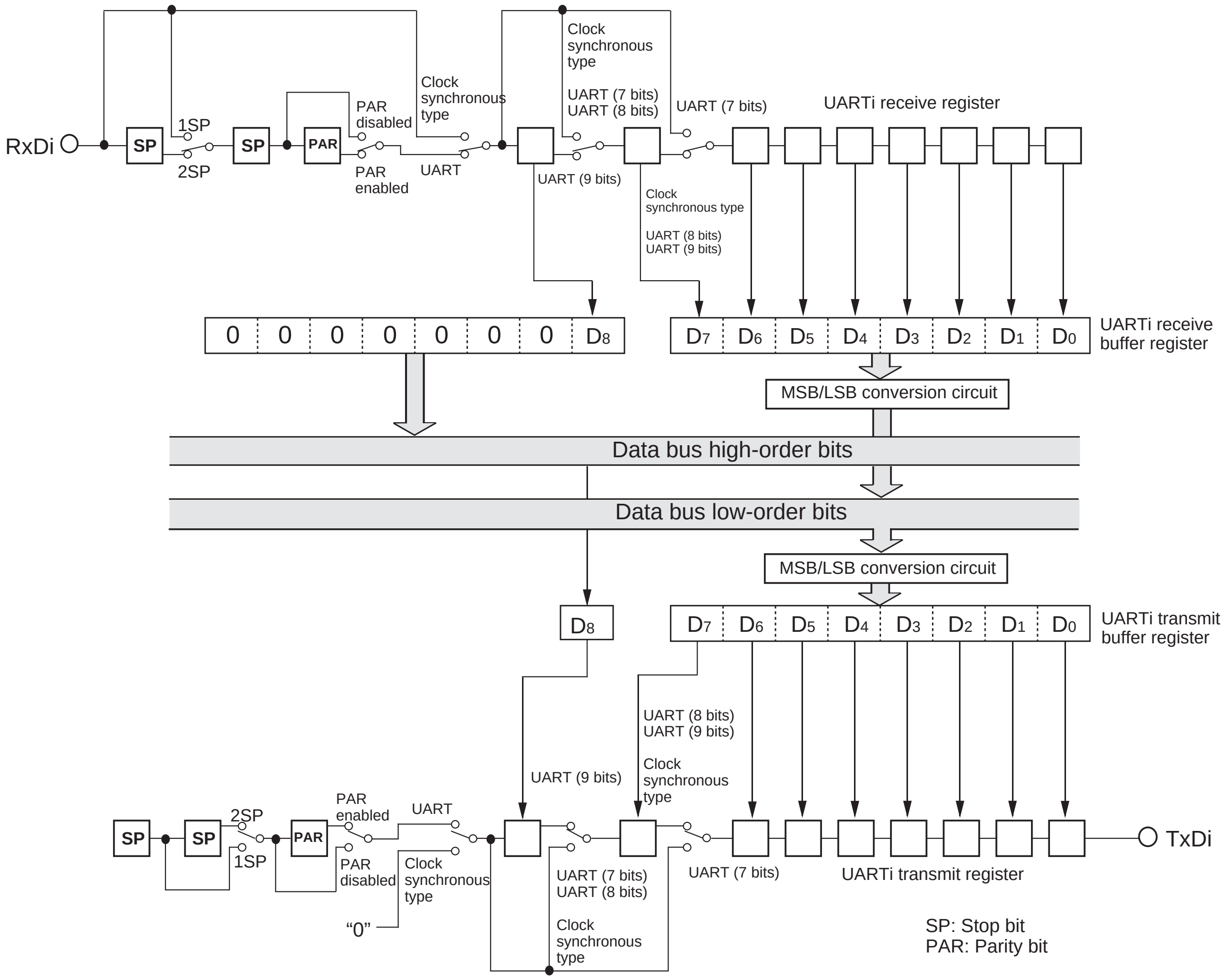


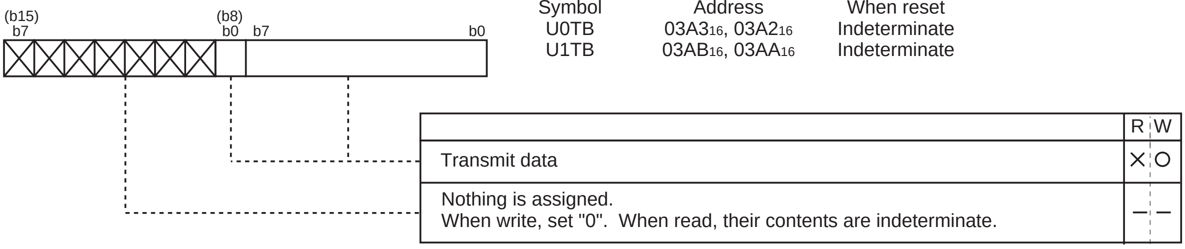
Figure 69. Block diagram of UARTi (i = 0, 1)



Note: UART1 cannot be used in clock synchronous serial I/O.

Figure 70. Block diagram of transmit/receive unit

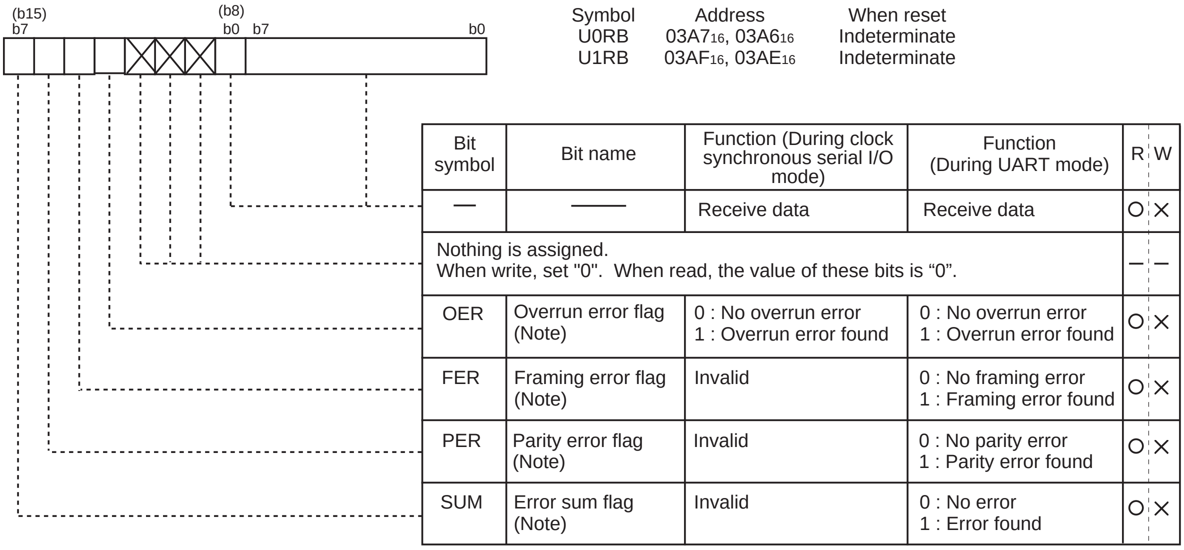
UARTi transmit buffer register



Symbol	Address	When reset
U0TB	$03A3_{16}$, $03A2_{16}$	Indeterminate
U1TB	$03AB_{16}$, $03AA_{16}$	Indeterminate

	R	W
Transmit data	×	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.	–	–

UARTi receive buffer register

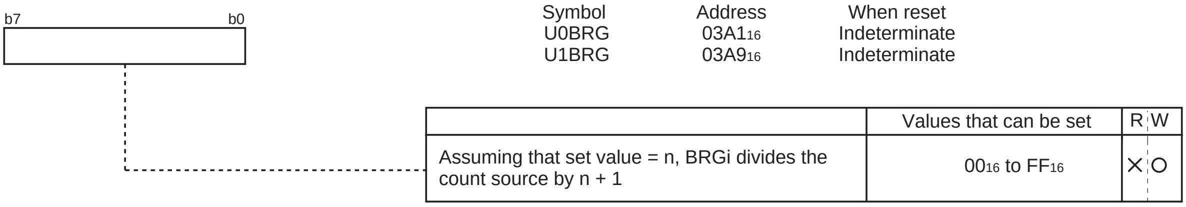


Symbol	Address	When reset
U0RB	$03A7_{16}$, $03A6_{16}$	Indeterminate
U1RB	$03AF_{16}$, $03AE_{16}$	Indeterminate

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
—	——	Receive data	Receive data	○	×
Nothing is assigned. When write, set "0". When read, the value of these bits is “0”.				–	–
OER	Overrun error flag (Note)	0 : No overrun error 1 : Overrun error found	0 : No overrun error 1 : Overrun error found	○	×
FER	Framing error flag (Note)	Invalid	0 : No framing error 1 : Framing error found	○	×
PER	Parity error flag (Note)	Invalid	0 : No parity error 1 : Parity error found	○	×
SUM	Error sum flag (Note)	Invalid	0 : No error 1 : Error found	○	×

Note: Bits 15 through 12 are set to “0” when the serial I/O mode select bit (bits 2 to 0 at addresses $03A0_{16}$ and $03A8_{16}$) are set to “000_2” or the receive enable bit is set to “0”. (Bit 15 is set to “0” when bits 14 to 12 all are set to “0”.) Bits 14 and 13 are also set to “0” when the lower byte of the UARTi receive buffer register (addresses $03A6_{16}$, and 0 $3AE_{16}$) is read out.

UARTi bit rate generator



Symbol	Address	When reset
U0BRG	$03A1_{16}$	Indeterminate
U1BRG	$03A9_{16}$	Indeterminate

	Values that can be set	R	W
Assuming that set value = n, BRGi divides the count source by n + 1	00_{16} to FF_{16}	×	○

Figure 71. Serial I/O-related registers (1)

UARTi transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
UiMR(i=0,1)	03A0$_{16}$, 03A8$_{16}$	00$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit (Note 1)	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	○	○
SMD1				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit (Note 2)	0 : Internal clock 1 : External clock	0 : Internal clock 1 : External clock	○	○
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep select bit	Must always be “0”	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

Note 1: UART1 cannot be used in clock synchronous serial I/O.
Note 2: UART1 can use only internal clock. Must set this bit to “1”.

UARTi transmit/receive control register 0

b7	b6	b5	b4	b3	b2	b1	b0
			0		0		

Symbol	Address	When reset
UiC0(i=0,1)	03A4$_{16}$, 03AC$_{16}$	08$_{16}$

Bit symbol	Bit name	Function (Note) (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : fc is selected	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : fc is selected	○	○
CLK1				○	○
Set this bit to “0”.				○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
Set this bit to “0”.				○	○
NCH	Data output select bit	0 : TXDi pin is CMOS output 1 : TXDi pin is N-channel open-drain output	0: TXDi pin is CMOS output 1: TXDi pin is N-channel open-drain output	○	○
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Must always be “0”	○	○
UFORM	Transfer format select bit	0 : LSB first 1 : MSB first	Must always be “0”	○	○

Note: UART1 cannot be used in clock synchronous serial I/O.

Figure 72. Serial I/O-related registers (2)

UARTi transmit/receive control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiC1(i=0,1)	03A5$_{16}$,03AD$_{16}$	02$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	○	×
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	○	○
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	○	×
Nothing is assigned. When write, set "0". When read, the value of these bits is "0".				–	–

Note: UART1 cannot be used in clock synchronous serial I/O.

UART transmit/receive control register 2

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UCON	03B0$_{16}$	XX0000000$_2$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U1IRS	UART1 transmit interrupt cause select bit	Set this bit to "0".	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	Invalid	○	○
Set this bit to "0".				○	○
CLKMD0	CLK/CLKS select bit 0	Valid when bit 5 = "1" 0 : Clock output to CLK1 1 : Clock output to CLKS1	Invalid	○	○
CLKMD1	CLK/CLKS select bit 1 (Note 2)	0 : Normal mode (CLK output is CLK0 only) 1 : Transfer clock output from multiple pins function selected	Must always be "0"	○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.				–	–

Note 1: UART1 cannot be used in clock synchronous serial I/O.
Note 2: When using multiple pins to output the transfer clock, the following requirements must be met:
- UART0 internal/external clock select bit (bit 3 at address 03A0$_{16}$) = "0".

Figure 73. Serial I/O-related registers (3)

(1) Clock synchronous serial I/O mode

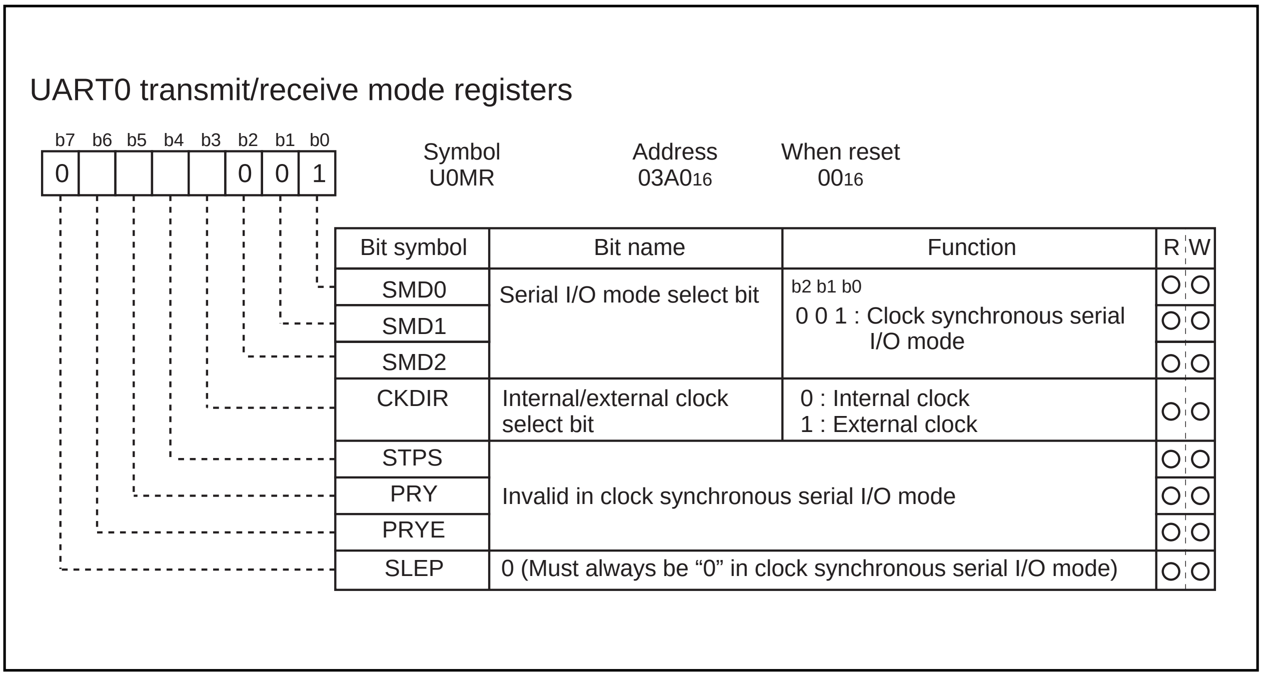
The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. (See Tables 26.) Figure 65 shows the UART0 transmit/receive mode register.

Table 26. 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$_{16}$ = “0”) : fi/ 2(n+1) (Note 1) fi = f$_1$, f$_8$, f$_{32}$, fc • When external clock is selected (bit 3 at address 03A0$_{16}$ = “1”) : Input from CLK0 pin
Transmission start condition	• To start transmission, the following requirements must be met: – Transmit enable bit (bit 0 at address 03A5$_{16}$) = “1” – Transmit buffer empty flag (bit 1 at addresses 03A5$_{16}$) = “0” • Furthermore, if external clock is selected, the following requirements must also be met: – CLK0 polarity select bit (bit 6 at address 03A4$_{16}$) = “0”: CLK0 input level = “H” – CLK0 polarity select bit (bit 6 at address 03A4$_{16}$) = “1”: CLK0 input level = “L”
Reception start conditio	• To start reception, the following requirements must be met: – Receive enable bit (bit 2 at address 03A5$_{16}$) = “1” – Transmit enable bit (bit 0 at address 03A5$_{16}$) = “1” – Transmit buffer empty flag (bit 1 at address 03A5$_{16}$) = “0” • Furthermore, if external clock is selected, the following requirements must also be met: – CLK0 polarity select bit (bit 6 at address 03A4$_{16}$) = “0”: CLK0 input level = “H” – CLK0 polarity select bit (bit 6 at address 03A4$_{16}$) = “1”: CLK0 input level = “L”
Interrupt request generation timing	• When transmitting – Transmit interrupt cause select bit (bit 0 at address 03B0$_{16}$) = “0”: Interrupts requested when data transfer from UART0 transfer buffer register to UART0 transmit register is completed – Transmit interrupt cause select bit (bit 0 at address 03B0$_{16}$) = “1”: Interrupts requested when data transmission from UART0 transfer register is completed • When receiving – Interrupts requested when data transfer from UART0 receive register to UART0 receive buffer register is completed
Error detection	• Overrun error (Note 2) This error occurs when the next data is ready before contents of UART0 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 UART0 transfer clock can be chosen by software to be output from one of the two pins set

Note 1: “n” denotes the value 00$_{16}$ to FF$_{16}$ that is set to the UART bit rate generator.

Note 2: If an overrun error occurs, the UART0 receive buffer will have the next data written in. Note also that the UART0 receive interrupt request bit is not set to “1”.



UART0 transmit/receive mode registers

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

Symbol	Address	When reset
U0MR	03A0$_{16}$	00$_{16}$

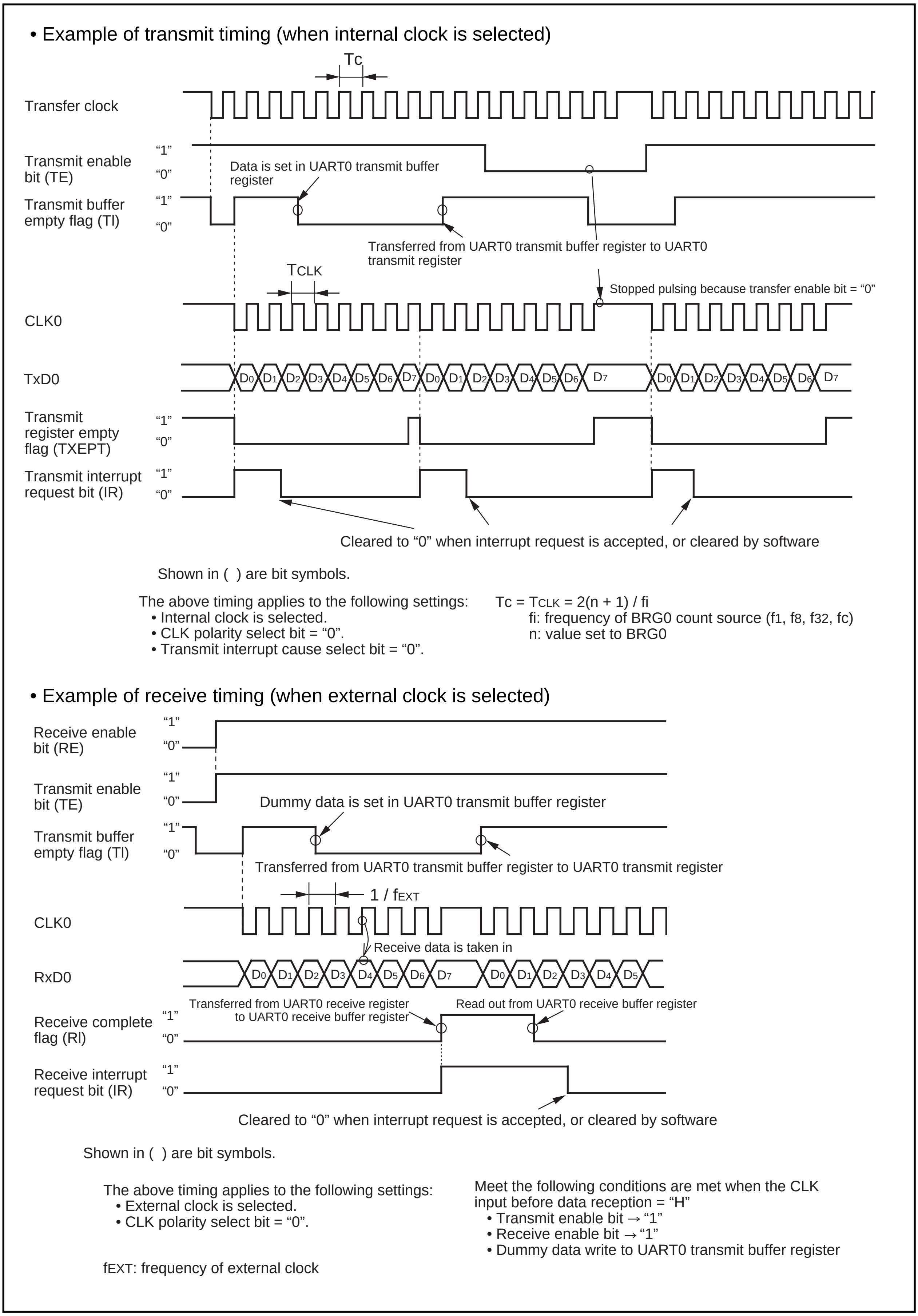
Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
SLEP	0 (Must always be “0” in clock synchronous serial I/O mode)		○	○

Figure 74. UART0 transmit/receive mode register in clock synchronous serial I/O mode

Table 27 lists the functions of the input/output pins during clock synchronous serial I/O mode. Note that for a period from when the UART0 operation mode is selected to when transfer starts, the TxD0 pin outputs a “H”. (If the N-channel open-drain is selected, this pin is in floating state.)

Table 27. Input/output pin functions in clock synchronous serial I/O mode

Pin name	Function	Method of selection
TxD0 (P5$_0$)	Serial data output	Port P5$_0$ direction register (bit 0 at address 03EB$_{16}$)= “1” (Outputs dummy data when performing reception only)
RxD0 (P5$_1$)	Serial data input	Port P5$_1$ direction register (bit 1 at address 03EB$_{16}$)= “0” (Can be used as an input port when performing transmission only)
CLK0 (P5$_2$)	Transfer clock output	Internal/external clock select bit (bit 3 at address 03A0$_{16}$) = “0”
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0$_{16}$) = “1” Port P5$_2$ direction register (bit 2 at address 03EB$_{16}$) = “0”


• Example of transmit timing (when internal clock is selected)
Tc
Transfer clock
Transmit enable bit (TE)
"1"
"0"
Data is set in UART0 transmit buffer register
Transmit buffer empty flag (TI)
"1"
"0"
Transferred from UART0 transmit buffer register to UART0 transmit register
TCLK
Stopped pulsing because transfer enable bit = "0"
CLK0
TxD0
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software
• Example of receive timing (when external clock is selected)
Receive enable bit (RE)
"1"
"0"
Transmit enable bit (TE)
"1"
"0"
Dummy data is set in UART0 transmit buffer register
Transmit buffer empty flag (TI)
"1"
"0"
Transferred from UART0 transmit buffer register to UART0 transmit register
1 / fEXT
CLK0
Receive data is taken in
RxD0
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5
Transferred from UART0 receive register to UART0 receive buffer register
Read out from UART0 receive buffer register
Receive complete flag (RI)
"1"
"0"
Receive interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- Internal clock is selected.
- CLK polarity select bit = "0".
- Transmit interrupt cause select bit = "0".

$T_C = T_{CLK} = 2(n + 1) / fi$
fi: frequency of BRG0 count source (f_1, f_8, f_{32}, fc)
n: value set to BRG0

Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- External clock is selected.
- CLK polarity select bit = "0".

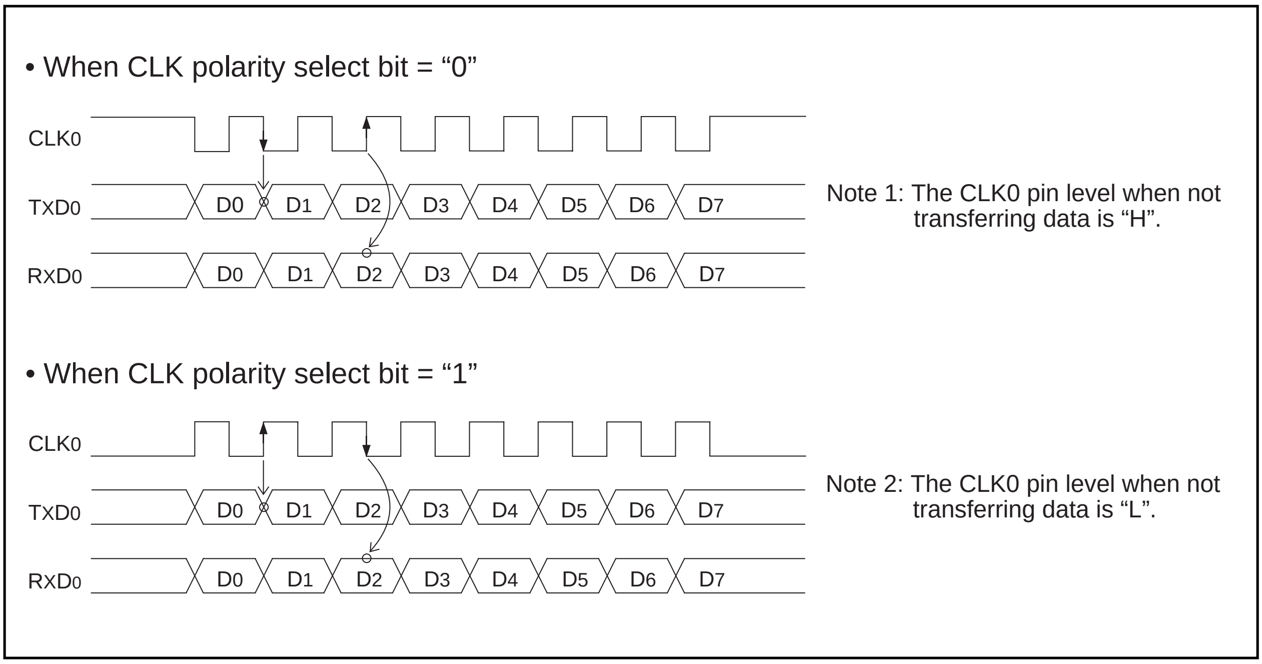
f_{EXT}: frequency of external clock

Meet the following conditions are met when the CLK input before data reception = "H"
- Transmit enable bit → "1"
- Receive enable bit → "1"
- Dummy data write to UART0 transmit buffer register

Figure 75. Typical transmit/receive timings in clock synchronous serial I/O mode

(a) Polarity select function

As shown in Figure 76, the CLK polarity select bit (bit 6 at addresses 03A4$_{16}$) allows selection of the polarity of the transfer clock.



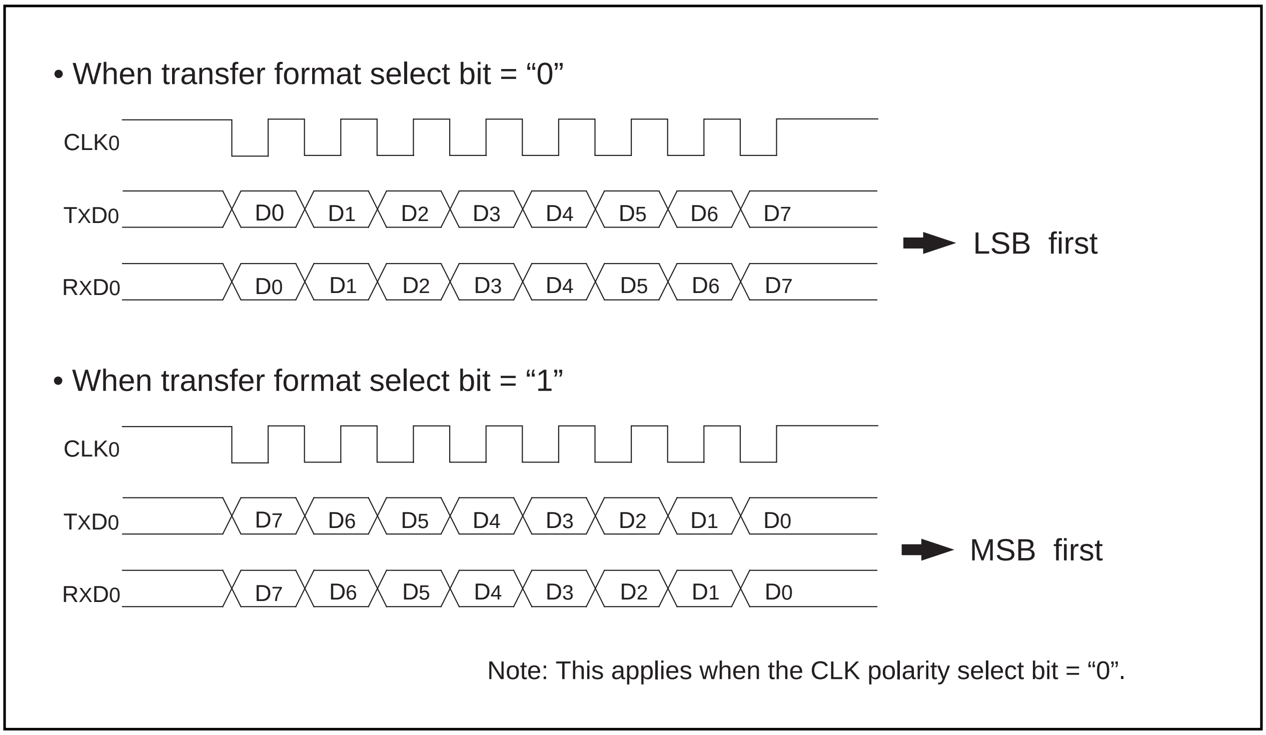
Note 1: The CLK0 pin level when not transferring data is “H”.

Note 2: The CLK0 pin level when not transferring data is “L”.

Figure 76. Polarity of transfer clock

(b) LSB first/MSB first select function

As shown in Figure 77, when the transfer format select bit (bit 7 at addresses 03A4$_{16}$) = “0”, the transfer format is “LSB first”; when the bit = “1”, the transfer format is “MSB first”.

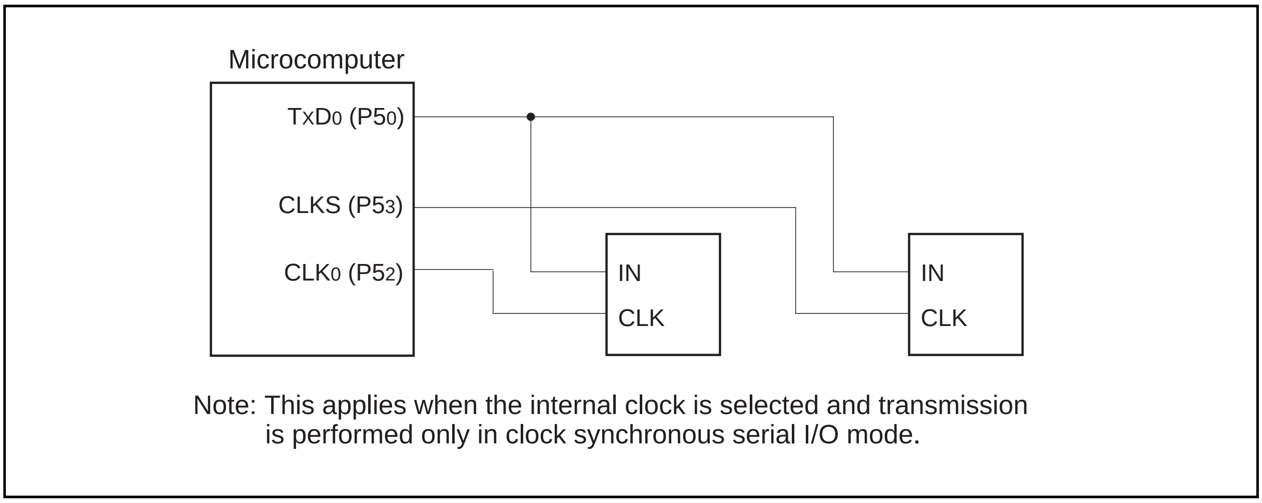


Note: This applies when the CLK polarity select bit = “0”.

Figure 77. 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 $03B0_{16}$). (See Figure 78.) The multiple pins function is valid only when the internal clock is selected for UART0.



Note: This applies when the internal clock is selected and transmission is performed only in clock synchronous serial I/O mode.

Figure 78. 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 $03B0_{16}$) 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.

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

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. (See Tables 28.) Figure 79 shows the UARTi transmit/receive mode register.

Table 28. Specifications of UART Mode

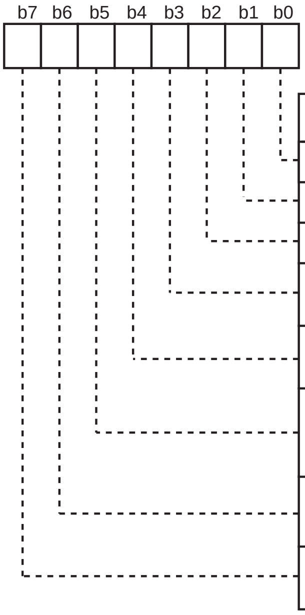
Item	Specification
Transfer data format	• Character bit (transfer data): 7 bits, 8 bits, or 9 bits as selected • Start bit: 1 bit • Parity bit: Odd, even, or nothing as selected • Stop bit: 1 bit or 2 bits as selected
Transfer clock	• When internal clock is selected (bit 3 at addresses $03A0_{16}$, $03A8_{16}$ = “0”) : $f_i/16(n+1)$ (Note 1) $f_i = f_1, f_8, f_{32}, f_C$ • When external clock is selected (bit 3 at addresses $03A0_{16}$=“1”) : $f_{EXT}/16(n+1)$ (Note 1) (Note 2)
Transmission start condition	• To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 at addresses $03A5_{16}$, $03AD_{16}$) = “1” - Transmit buffer empty flag (bit 1 at addresses $03A5_{16}$, $03AD_{16}$) = “0”
Reception start condition	• To start reception, the following requirements must be met: - Receive enable bit (bit 2 at addresses $03A5_{16}$, $03AD_{16}$) = “1” - Start bit detection
Interrupt request generation timing	• When transmitting - Transmit interrupt cause select bits (bits 0,1 at address $03B0_{16}$) = “0”: Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed - Transmit interrupt cause select bits (bits 0, 1 at address $03B0_{16}$) = “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 • Framing error This error occurs when the number of stop bits set is not detected • 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 • Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered
Select function	• Sleep mode selection This mode is used to transfer data to and from one of multiple slave microcomputers

Note 1: ‘n’ denotes the value 00_{16} to FF_{16} that is set to the UART bit rate generator.

Note 2: f_{EXT} is input from the CLK0 pin. Since UART1 does not have this pin, cannot select external clock.

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”.

UARTi transmit / receive mode registers



Symbol	Address	When reset
UiMR(i=0,1)	03A0$_{16}$, 03A8$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal / external clock select bit (Note)	0 : Internal clock 1 : External clock	○	○
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd / even parity select bit	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep select bit	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

Note: UART1 can use only internal clock. Must set this bit to “1”.

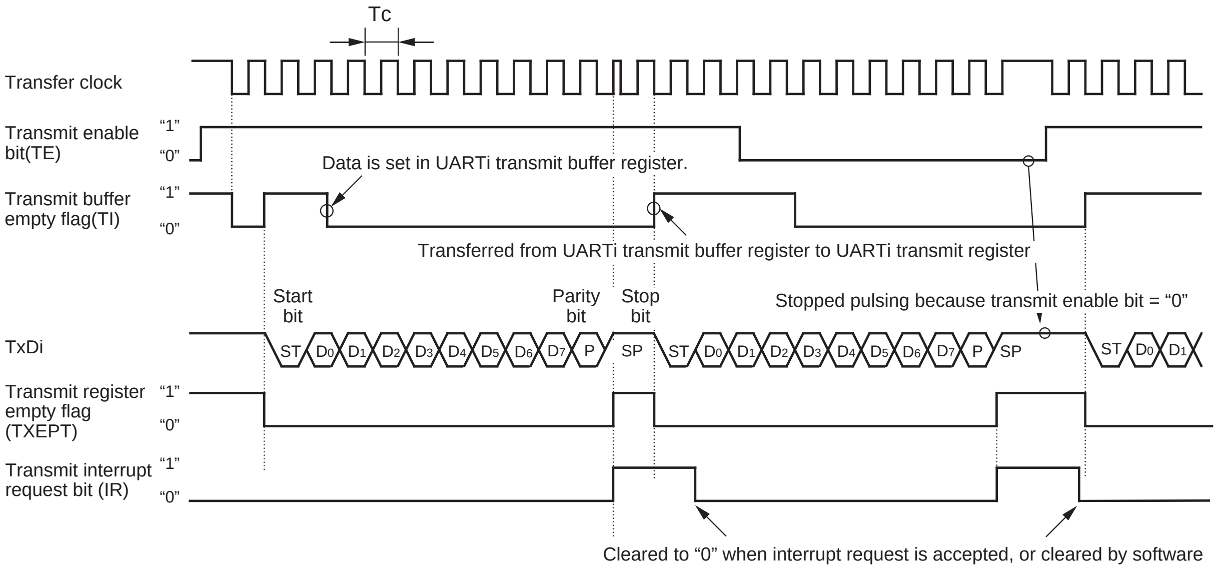
Figure 79. UARTi transmit/receive mode register in UART mode

Table 29 lists the functions of the input/output pins during UART 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 29. Input/output pin functions in UART mode

Pin name	Function	Method of selection
TxDi (P5$_0$, P4$_0$)	Serial data output	Port P5$_1$ and P4$_2$ direction register (bit 0 at address 03EB$_{16}$, bit 0 at address 03EA$_{16}$)= “1” (Can be used as an input port when performing reception only)
RxDi (P5$_1$, P4$_2$)	Serial data input	Port P5$_1$ and P4$_2$ direction register (bit 1 at address 03EB$_{16}$, bit 2 at address 03EA$_{16}$)= “0” (Can be used as an input port when performing transmission only)
CLK0 (P5$_2$)	Programmable I/O port	Internal/external clock select bit (bit 3 at address 03A0$_{16}$) = “0”
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0$_{16}$) = “1”

- Example of transmit timing when transfer data is 8 bits long (parity enabled, one stop bit)

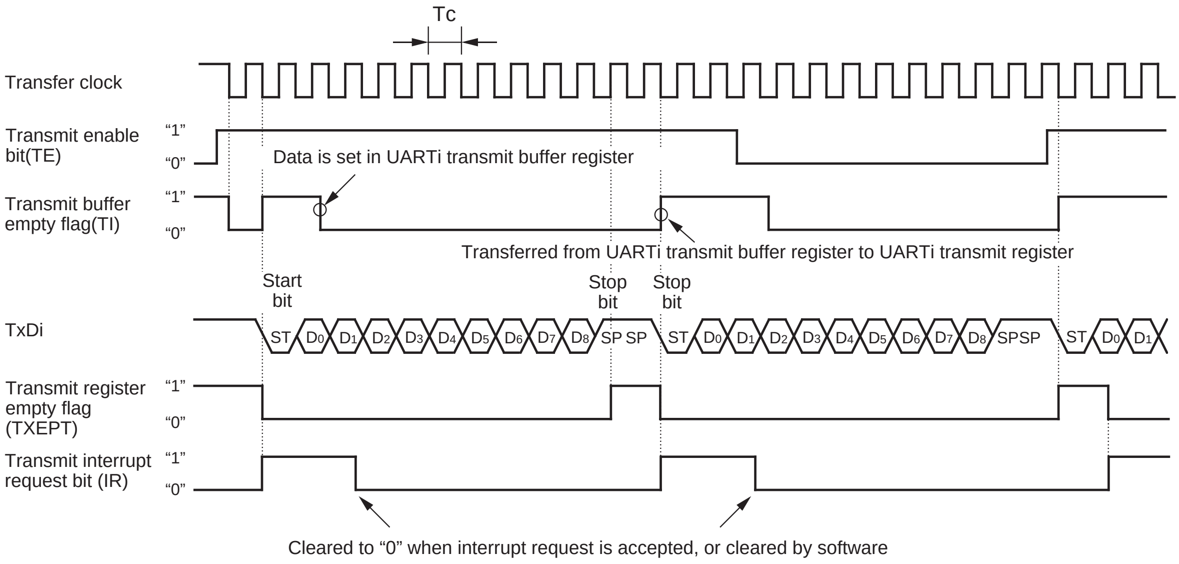

Tc
Transfer clock
Transmit enable bit(TE)
"1"
"0"
Data is set in UARTi transmit buffer register.
Transmit buffer empty flag(TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
Start bit
Parity bit
Stop bit
Stopped pulsing because transmit enable bit = "0"
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "1".

Tc = 16 (n + 1) / f_i or 16 (n + 1) / f_{EXT}
f_i : frequency of BRGi count source (f_1, f_8, f_{32}, f_C)
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi

- Example of transmit timing when transfer data is 9 bits long (parity disabled, two stop bits)


Tc
Transfer clock
Transmit enable bit(TE)
"1"
"0"
Data is set in UARTi transmit buffer register
Transmit buffer empty flag(TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
Start bit
Stop bit
Stop bit
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

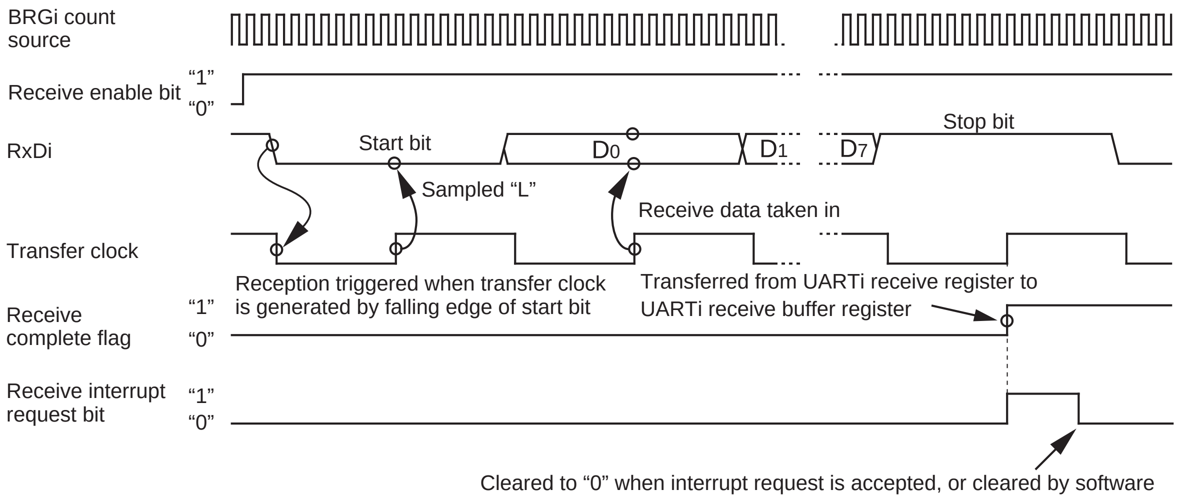
Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is disabled.
- Two stop bits.
- Transmit interrupt cause select bit = "0".

Tc = 16 (n + 1) / f_i or 16 (n + 1) / f_{EXT}
f_i : frequency of BRGi count source (f_1, f_8, f_{32})
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi

Figure 80. Typical transmit timings in UART mode

• Example of receive timing when transfer data is 8 bits long (parity disabled, one stop bit)


BRGi count source
Receive enable bit
"1"
"0"
RxDi
Start bit
D_0
D_1
D_7
Stop bit
Sampled "L"
Receive data taken in
Transfer clock
Reception triggered when transfer clock is generated by falling edge of start bit
Transferred from UARTi receive register to UARTi receive buffer register
Receive complete flag
"1"
"0"
Receive interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

The above timing applies to the following settings :
- Parity is disabled.
- One stop bit.

Figure 81. Typical receive timing in UART mode

(a) Sleep mode

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 $03A0_{16}$, $03A8_{16}$) 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".

A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins $P6_0$ to $P6_7$, and $P5_0$ to $P5_4$ 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 $03D7_{16}$) 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 $03D7_{16}$ 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 30 shows the performance of the A-D converter. Figure 82 shows the block diagram of the A-D converter, and Figures 83 and 84 show the A-D converter-related registers.

Table 30. 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 ϕAD (Note 2)	VCC = 5V fAD, divide-by-2 of fAD, divide-by-4 of fAD, fAD=f(XIN)
	VCC = 3V divide-by-2 of fAD, divide-by-4 of fAD, fAD=f(XIN)

Resolution	8-bit or 10-bit (selectable)
Absolute precision	VCC = 5V • Without sample and hold function ±3LSB • With sample and hold function (8-bit resolution) ±2LSB • With sample and hold function (10-bit resolution) ±3LSB VCC = 3V • Without sample and hold function (8-bit resolution) ±2LSB
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	8 pins (AN_0 to AN_7) + 5 pins (AN_{50} to AN_{54})
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 ϕAD cycles, 10-bit resolution: 59 ϕAD cycles • With sample and hold function 8-bit resolution: 28 ϕAD cycles, 10-bit resolution: 33 ϕAD cycles

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

Note 2: Without sample and hold function, set the ϕAD frequency to 250kHz min.
With the sample and hold function, set the ϕAD frequency to 1MHz min.

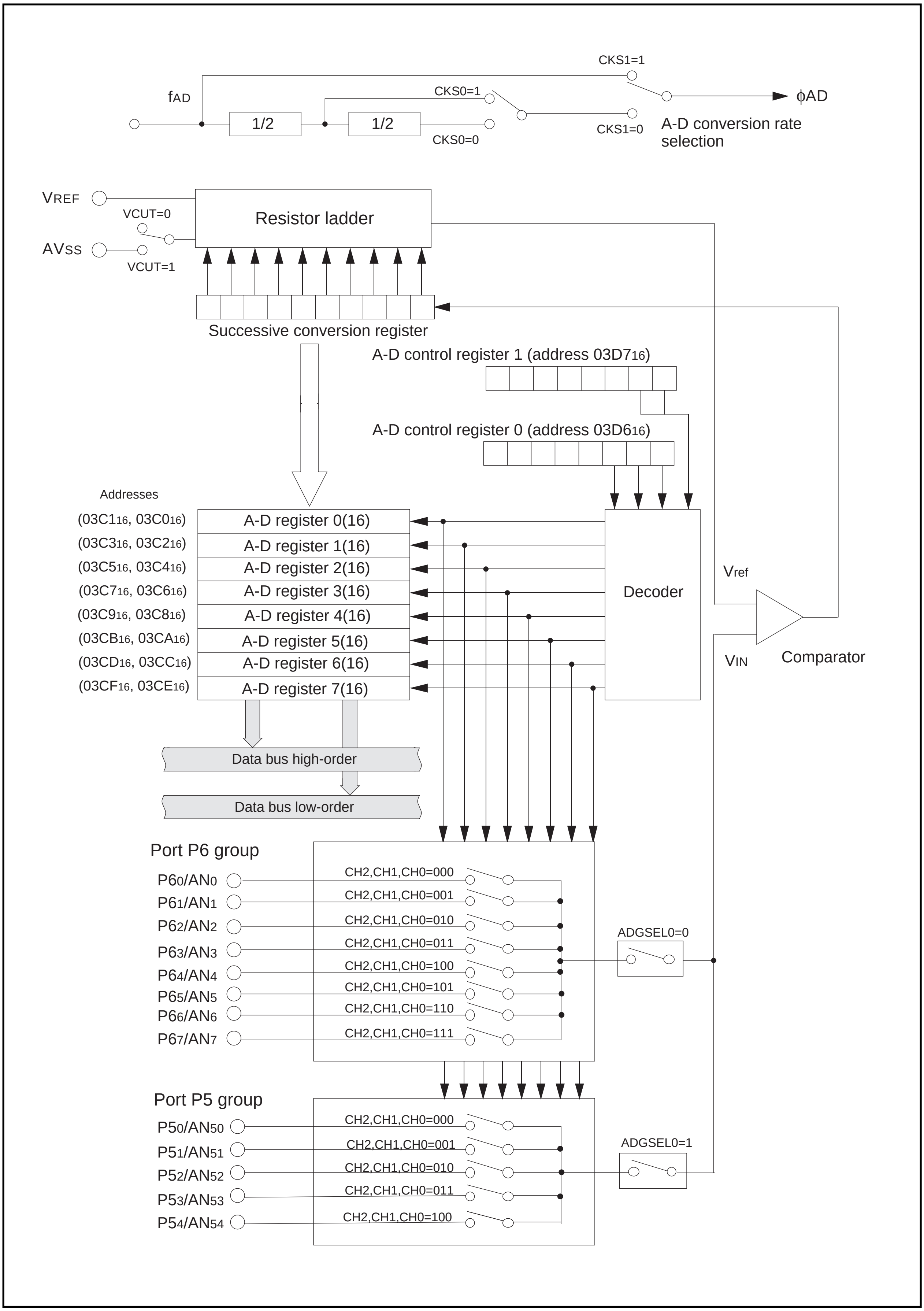


Figure 82. Block diagram of A-D converter

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Symbol	Address	When reset
ADCON0	03D6$_{16}$	00000XXX$_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN$_0$ is selected 0 0 1 : AN$_1$ is selected 0 1 0 : AN$_2$ is selected 0 1 1 : AN$_3$ is selected 1 0 0 : AN$_4$ is selected 1 0 1 : AN$_5$ is selected 1 1 0 : AN$_6$ is selected 1 1 1 : AN$_7$ is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 Repeat sweep mode 1	○	○
MD1			○	○
Set this bit to “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.

A-D control register 1 (Note 1)

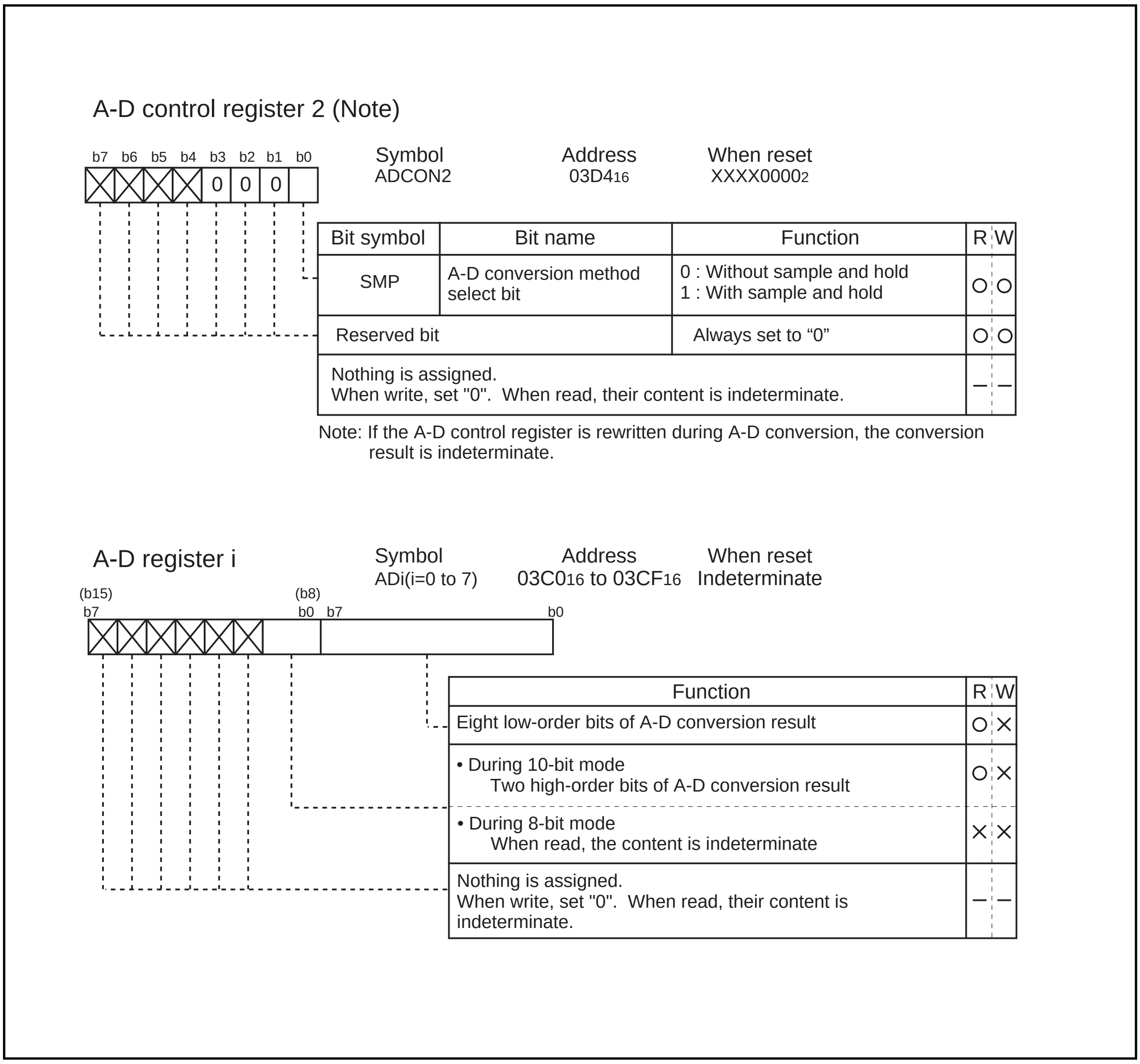
b7	b6	b5	b4	b3	b2	b1	b0
	0						

Symbol	Address	When reset
ADCON1	03D7$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN$_0$, AN$_1$ (2 pins) 0 1 : AN$_0$ to AN$_3$ (4 pins) 1 0 : AN$_0$ to AN$_5$ (6 pins) 1 1 : AN$_0$ to AN$_7$ (8 pins)	○	○
SCAN1		When repeat sweep mode 1 is selected b1 b0 0 0 : AN$_0$ (1 pin) 0 1 : AN$_0$, AN$_1$ (2 pins) 1 0 : AN$_0$ to AN$_2$ (3 pins) 1 1 : AN$_0$ to AN$_3$ (4 pins) (Note 2, 3)	○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1 1 : Repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : f$_{AD}$/2 or f$_{AD}$/4 is selected 1 : f$_{AD}$ is selected	○	○
VCUT	Vref connect bit	0 : Vref not connected 1 : Vref connected	○	○
Set this bit to “0”.			○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.
Note 3: If the repeat sweep mode is selected for the port P5 group, the contents of A-D registers 5 to 7 are indeterminate.

Figure 83. A-D converter-related registers (1)



A-D control register 2 (Note)

Symbol	Address	When reset
ADCON2	$03D4_{16}$	$XXXX0000_2$

Bit symbol	Bit name	Function	R	W
SMP	A-D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	○	○
Reserved bit		Always set to “0”	○	○
Nothing is assigned. When write, set "0". When read, their content is indeterminate.			–	–

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D register i

Symbol	Address	When reset
ADi(i=0 to 7)	$03C0_{16}$ to $03CF_{16}$	Indeterminate

Function	R	W
Eight low-order bits of A-D conversion result	○	×
• During 10-bit mode Two high-order bits of A-D conversion result	○	×
• During 8-bit mode When read, the content is indeterminate	×	×
Nothing is assigned. When write, set "0". When read, their content is indeterminate.	–	–

Figure 84. A-D converter-related registers (2)

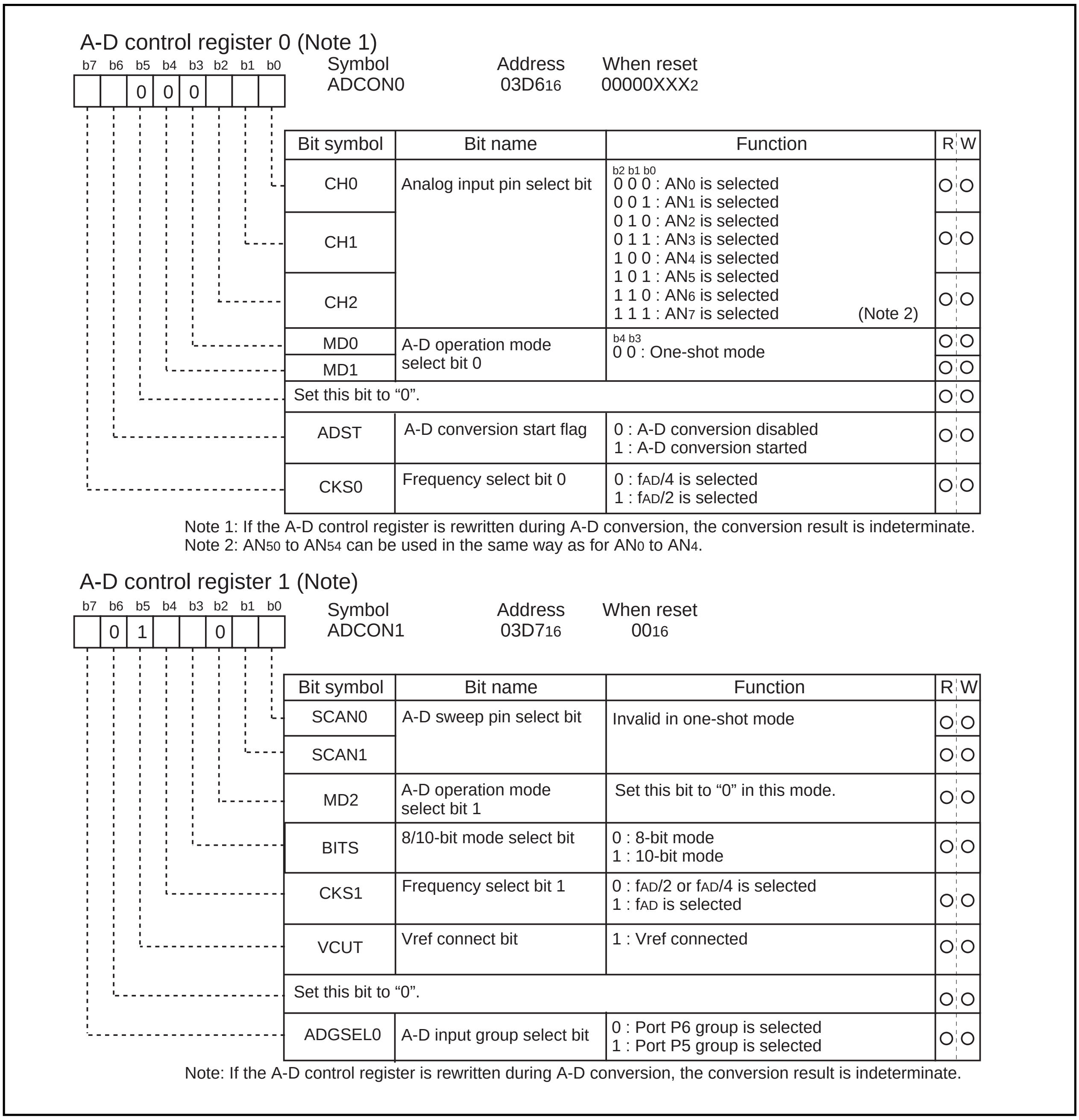
(1) One-shot mode

In one-shot mode, the pin selected using the analog input pin select bit is used for one-shot A-D conversion. (See Table 31.) Figure 85 shows the A-D control register in one-shot mode.

Table 31. 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	• End of A-D conversion (A-D conversion start flag changes to “0”) • Writing “0” to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN$_0$ to AN$_7$, as selected (Note)
Reading of result of A-D converter	Read A-D register corresponding to selected pin

Note : AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.



A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0			

Symbol ADCON0 Address 03D6$_{16}$ When reset 00000XXX$_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN$_0$ is selected 0 0 1 : AN$_1$ is selected 0 1 0 : AN$_2$ is selected 0 1 1 : AN$_3$ is selected 1 0 0 : AN$_4$ is selected 1 0 1 : AN$_5$ is selected 1 1 0 : AN$_6$ is selected 1 1 1 : AN$_7$ is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode	○	○
MD1			○	○
Set this bit to “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
	0	1			0		

Symbol ADCON1 Address 03D7$_{16}$ When reset 00$_{16}$

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	Invalid in one-shot mode	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set this bit to “0” in this mode.	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : f$_{AD}$/2 or f$_{AD}$/4 is selected 1 : f$_{AD}$ is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Set this bit to “0”.			○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Figure 85. A-D conversion register in one-shot mode

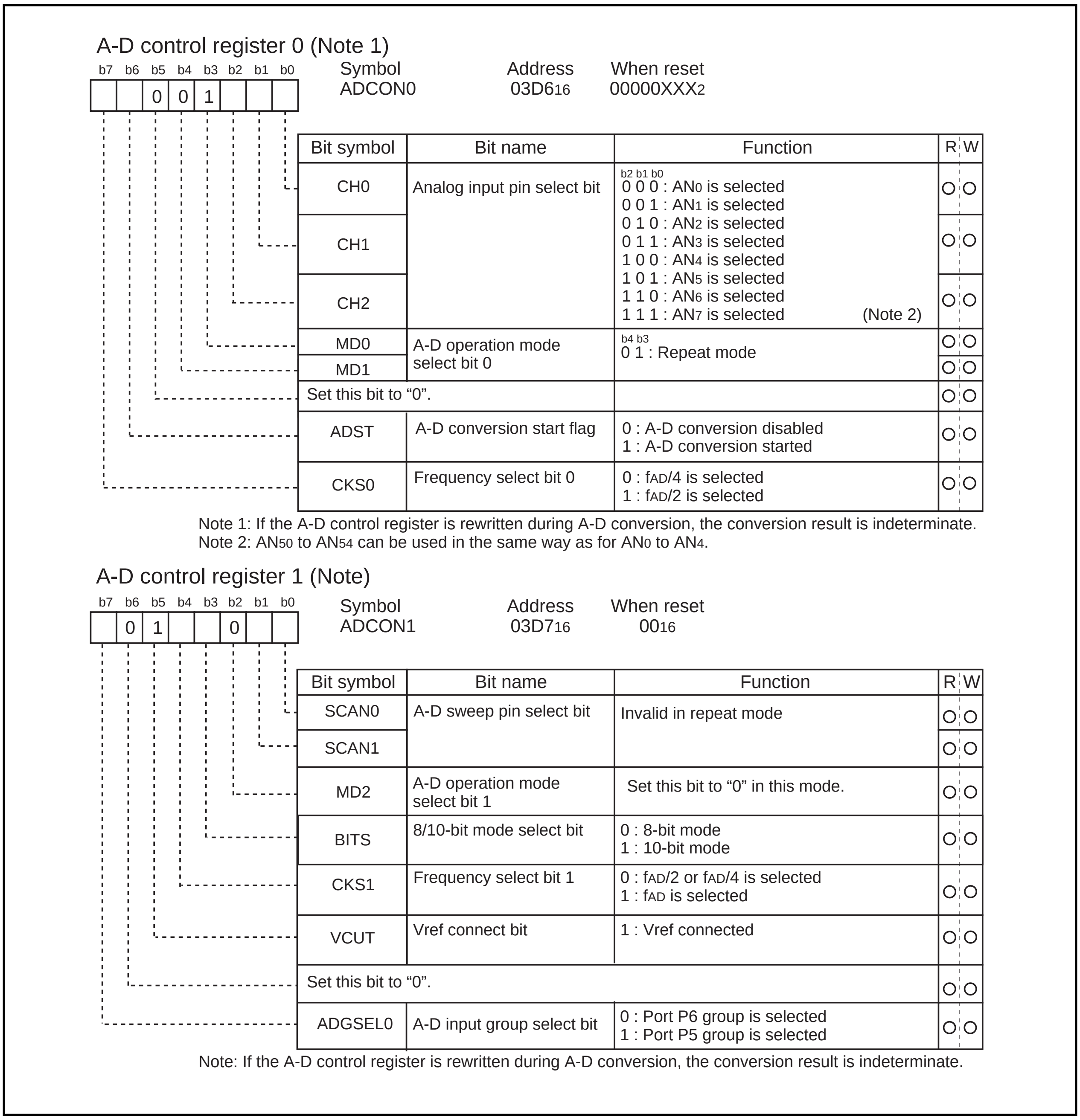
(2) Repeat mode

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A-D conversion. (See Table 32.) Figure 86 shows the A-D control register in repeat mode.

Table 32. Repeat mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated 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	One of AN$_0$ to AN$_7$, as selected (Note)
Reading of result of A-D converter	Read A-D register corresponding to selected pin

Note : AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.



A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	1			

Symbol ADCON0 Address 03D6$_{16}$ When reset 00000XXX$_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN$_0$ is selected 0 0 1 : AN$_1$ is selected 0 1 0 : AN$_2$ is selected 0 1 1 : AN$_3$ is selected 1 0 0 : AN$_4$ is selected 1 0 1 : AN$_5$ is selected 1 1 0 : AN$_6$ is selected 1 1 1 : AN$_7$ is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 1 : Repeat mode	○	○
MD1			○	○
Set this bit to “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
	0	1			0		

Symbol ADCON1 Address 03D7$_{16}$ When reset 00$_{16}$

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	Invalid in repeat mode	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set this bit to “0” in this mode.	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : f$_{AD}$/2 or f$_{AD}$/4 is selected 1 : f$_{AD}$ is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Set this bit to “0”.			○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Figure 86. A-D conversion register in repeat mode

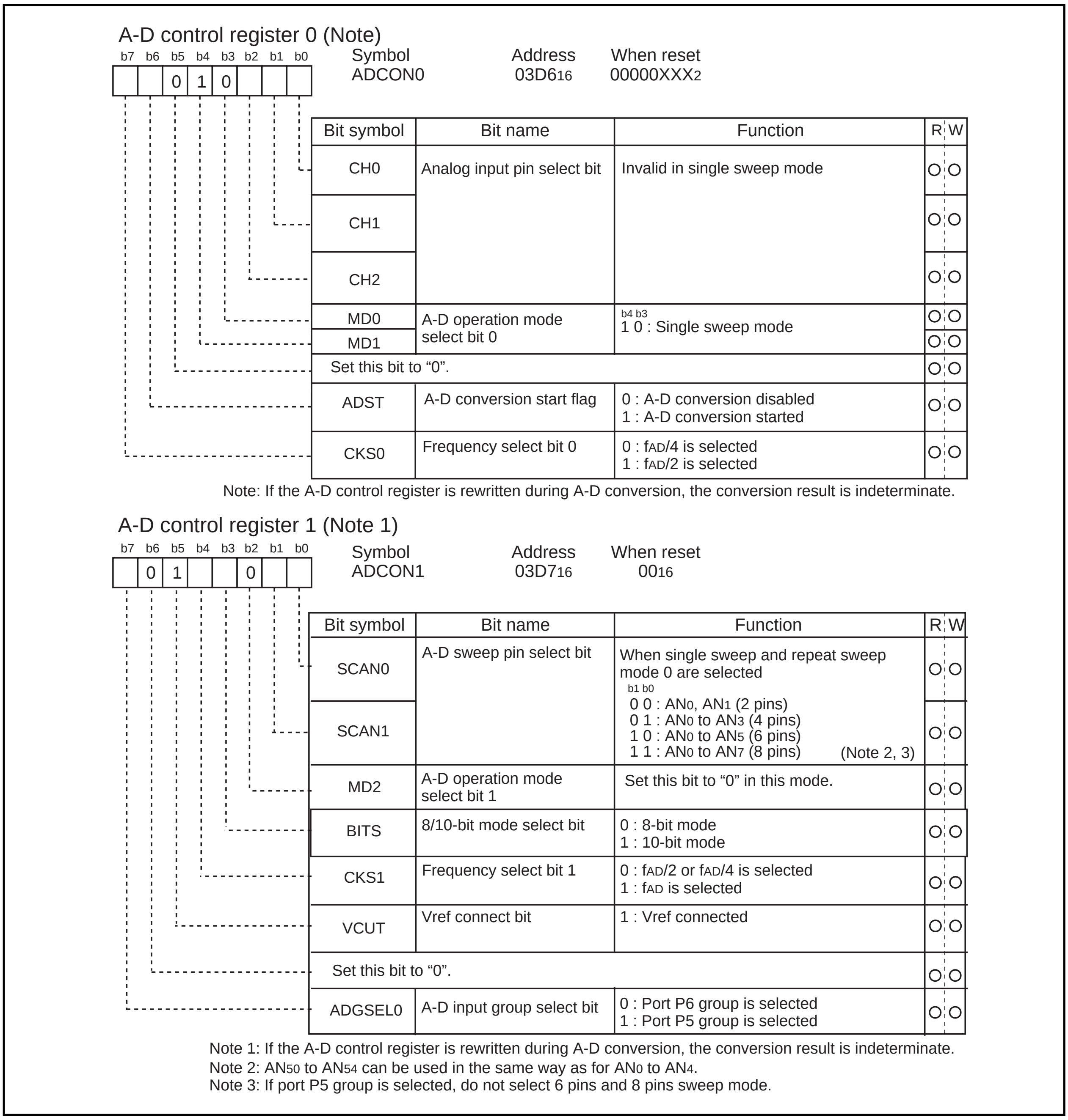
(3) Single sweep mode

In single sweep mode, the pins selected using the A-D sweep pin select bit are used for one-by-one A-D conversion. (See Table 33.) Figure 87 shows the A-D control register in single sweep mode.

Table 33. 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	• End of A-D conversion (A-D conversion start flag changes to “0”.) • Writing “0” to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	AN$_0$ and AN$_1$ (2 pins), AN$_0$ to AN$_3$ (4 pins), AN$_0$ to AN$_5$ (6 pins), or AN$_0$ to AN$_7$ (8 pins)(Note)
Reading of result of A-D converter	Read A-D register corresponding to selected pin

Note : AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.



A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	0			

Symbol	Address	When reset
ADCON0	03D6$_{16}$	00000XXX$_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in single sweep mode	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 0 : Single sweep mode	○	○
MD1			○	○
Set this bit to “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
	0	1			0		

Symbol	Address	When reset
ADCON1	03D7$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN$_0$, AN$_1$ (2 pins) 0 1 : AN$_0$ to AN$_3$ (4 pins) 1 0 : AN$_0$ to AN$_5$ (6 pins) 1 1 : AN$_0$ to AN$_7$ (8 pins) (Note 2, 3)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set this bit to “0” in this mode.	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : f$_{AD}$/2 or f$_{AD}$/4 is selected 1 : f$_{AD}$ is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Set this bit to “0”.			○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.
Note 3: If port P5 group is selected, do not select 6 pins and 8 pins sweep mode.

Figure 87. A-D conversion register in single sweep mode

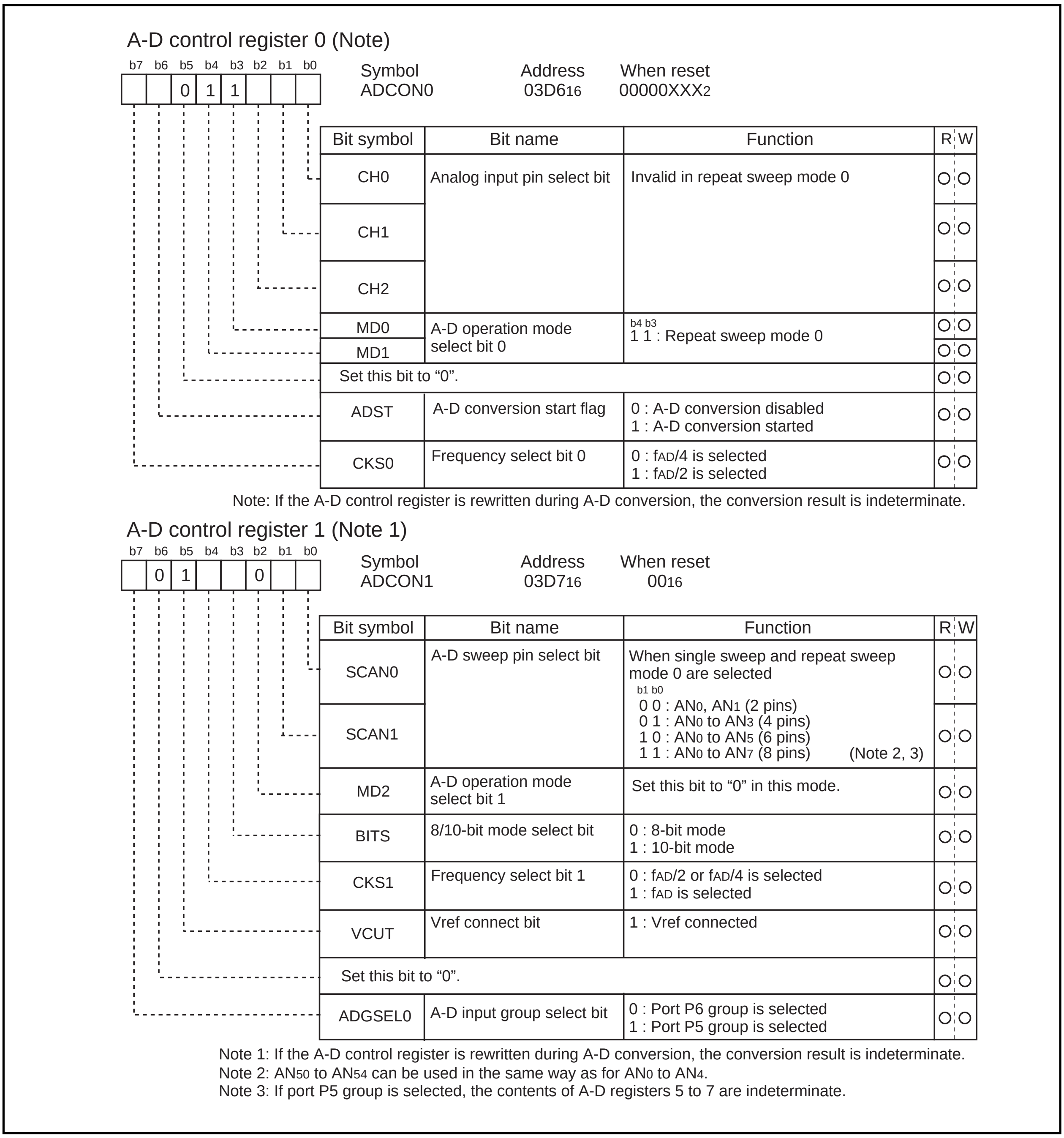
(4) Repeat sweep mode 0

In repeat sweep mode 0, the pins selected using the A-D sweep pin select bit are used for repeat sweep A-D conversion. (See Table 34.) Figure 88 shows the A-D control register in repeat sweep mode 0.

Table 34. 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$_0$ and AN$_1$ (2 pins), AN$_0$ to AN$_3$ (4 pins), AN$_0$ to AN$_5$ (6 pins), or AN$_0$ to AN$_7$ (8 pins)(Note)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)

Note : AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.



A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	1			

Symbol ADCON0 Address 03D6$_{16}$ When reset 00000XXX$_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in repeat sweep mode 0	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 0	○	○
MD1			○	○
Set this bit to “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
	0	1			0		

Symbol ADCON1 Address 03D7$_{16}$ When reset 00$_{16}$

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN$_0$, AN$_1$ (2 pins) 0 1 : AN$_0$ to AN$_3$ (4 pins) 1 0 : AN$_0$ to AN$_5$ (6 pins) 1 1 : AN$_0$ to AN$_7$ (8 pins) (Note 2, 3)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set this bit to “0” in this mode.	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : f$_{AD}$/2 or f$_{AD}$/4 is selected 1 : f$_{AD}$ is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Set this bit to “0”.			○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.
Note 3: If port P5 group is selected, the contents of A-D registers 5 to 7 are indeterminate.

Figure 88. A-D conversion register in repeat sweep mode 0

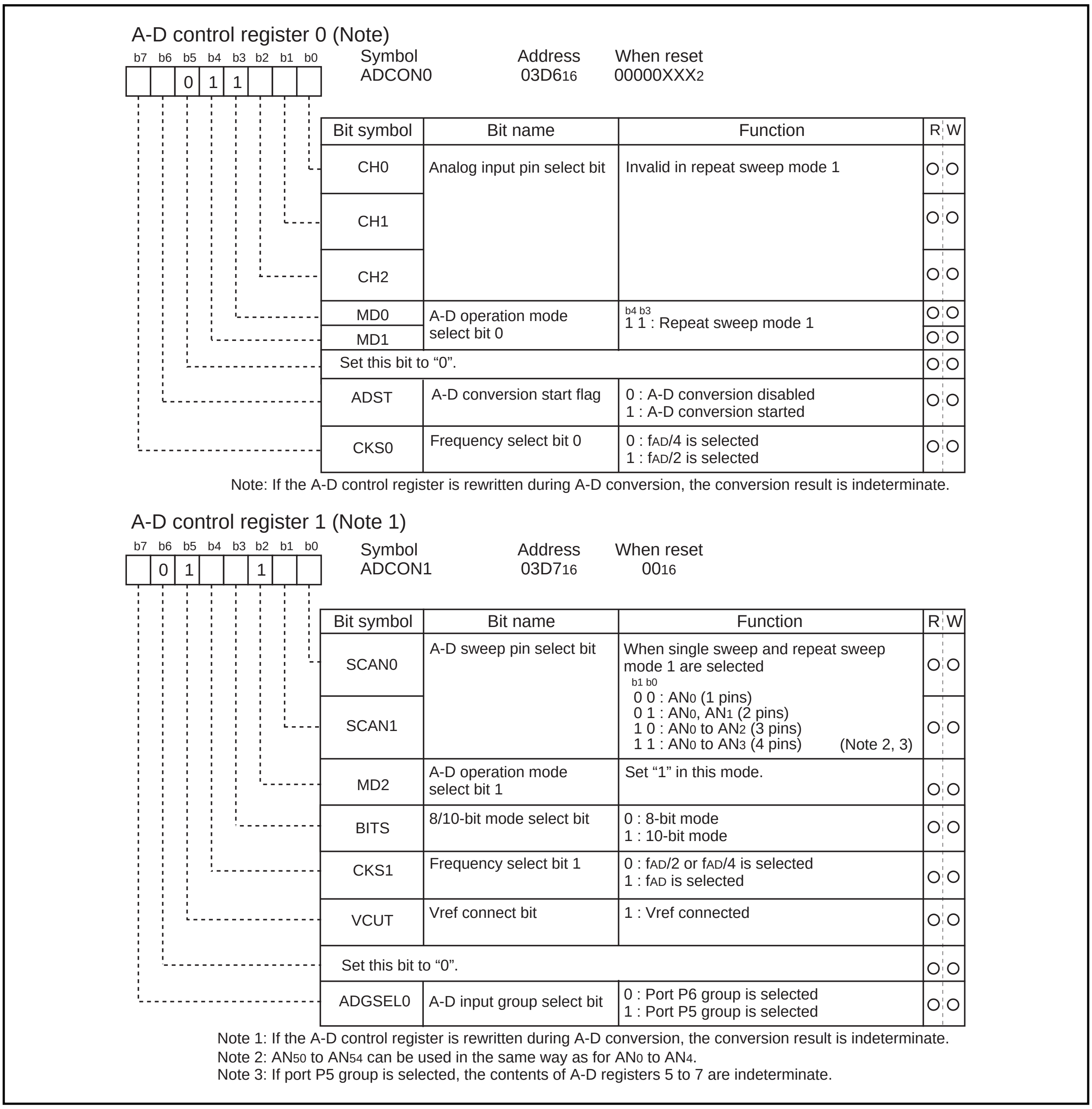
(5) Repeat sweep mode 1

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. (See Table 35.) Figure 89 shows the A-D control register in repeat sweep mode 1.

Table 35. 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 : AN0 selected AN0 → AN1 → AN0 → AN2 → AN0 → AN3, 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	AN0 (1 pin), AN0 and AN1 (2 pins), AN0 to AN2 (3 pins), AN0 to AN3 (4 pins) (Note)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)

Note : AN50 to AN54 can be used in the same way as for AN0 to AN4.



A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	1			

Symbol ADCON0 Address 03D6₁₆ When reset 00000XXX₂

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in repeat sweep mode 1	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 1	○	○
MD1			○	○
	Set this bit to “0”.		○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : fAD/4 is selected 1 : fAD/2 is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
	0	1			1		

Symbol ADCON1 Address 03D7₁₆ When reset 00₁₆

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 1 are selected b1 b0 0 0 : AN0 (1 pins) 0 1 : AN0, AN1 (2 pins) 1 0 : AN0 to AN2 (3 pins) 1 1 : AN0 to AN3 (4 pins) (Note 2, 3)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set “1” in this mode.	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : fAD/2 or fAD/4 is selected 1 : fAD is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
	Set this bit to “0”.		○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN50 to AN54 can be used in the same way as for AN0 to AN4.
Note 3: If port P5 group is selected, the contents of A-D registers 5 to 7 are indeterminate.

Figure 89. A-D conversion register in repeat sweep mode 1

• Sample and hold

Sample and hold is selected by setting bit 0 of the A-D control register 2 (address $03D4_{16}$) to “1”. When sample and hold is selected, the rate of conversion of each pin increases. As a result, a 28 ϕ_{AD} cycle is achieved with 8-bit resolution and 33 ϕ_{AD} 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.

Programmable I/O Ports

There are 43 programmable I/O ports: P0 to P7. 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. The port P1 allows the drive capacity of its N-channel output transistor to be set as necessary.
Figures 90 to 92 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, they function as outputs regardless of the contents of the direction registers. See the descriptions of the respective functions for how to set up the built-in peripheral devices.

(1) Direction registers

Figure 93 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 94 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 95 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.

(4) Port P1 drive capacity control register

Figure 95 shows a structure of the port P1 drive capacity control register.
This register is used to control the drive capacity of the port P1's N-channel output transistor. Each bit in this register corresponds one for one to the port pins.

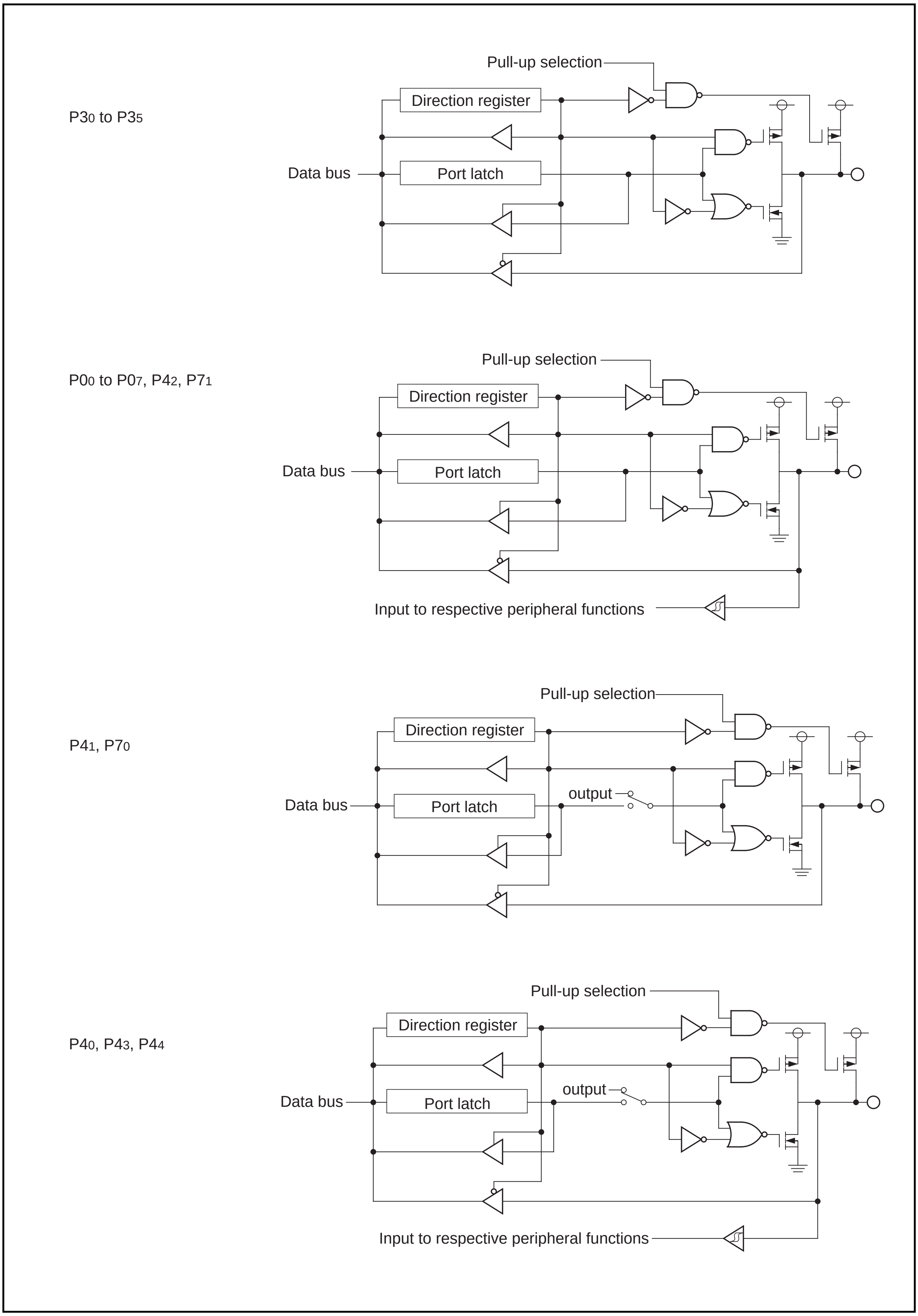


Figure 90. Programmable I/O ports (1)

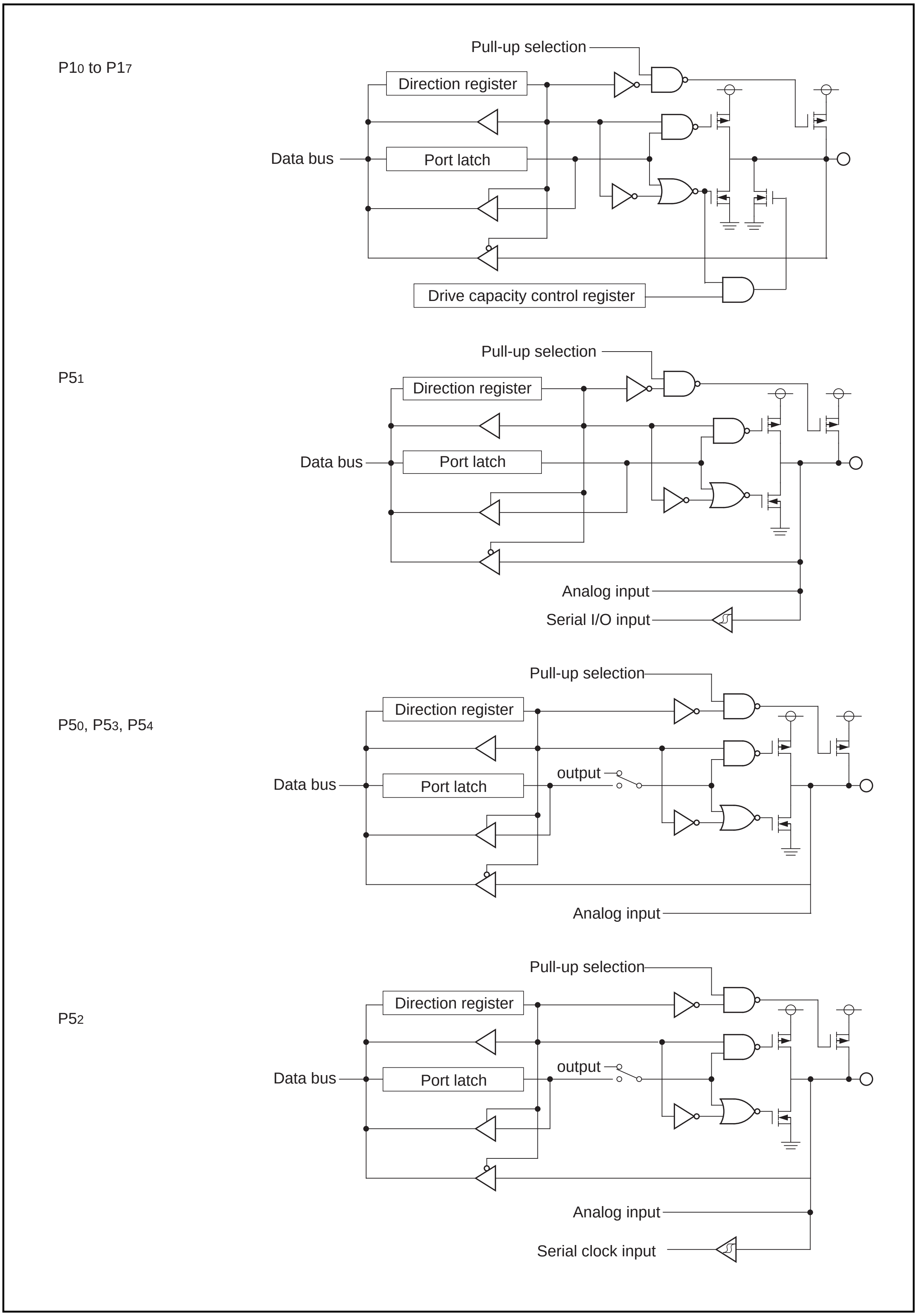


Figure 91. Programmable I/O ports (2)

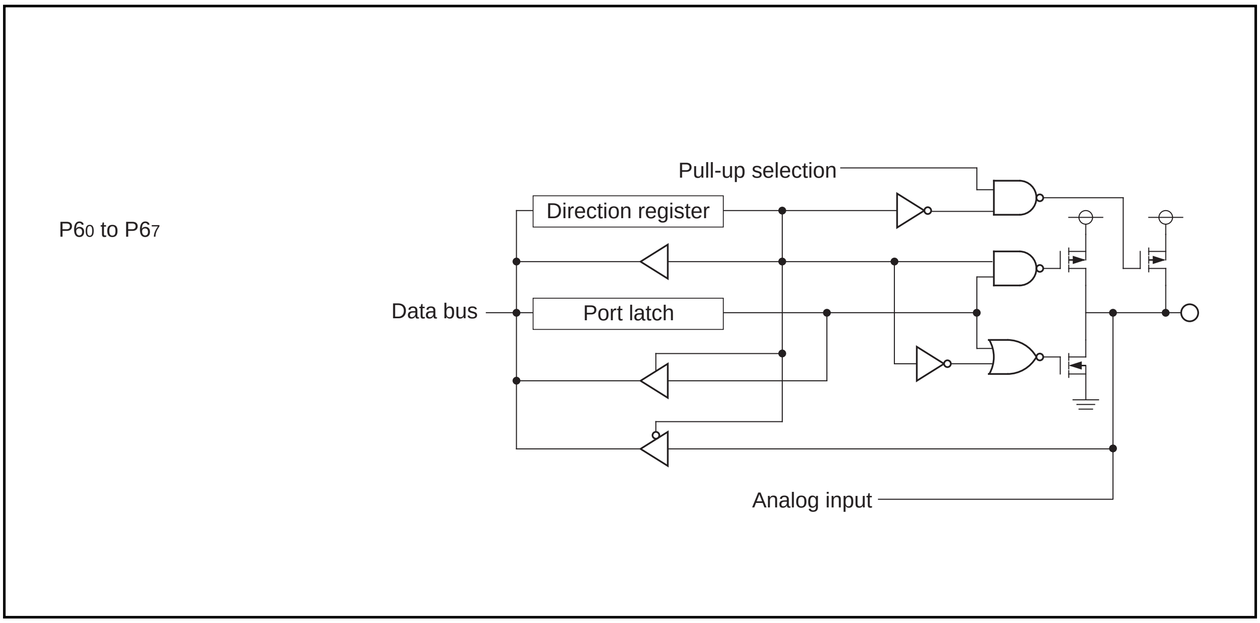
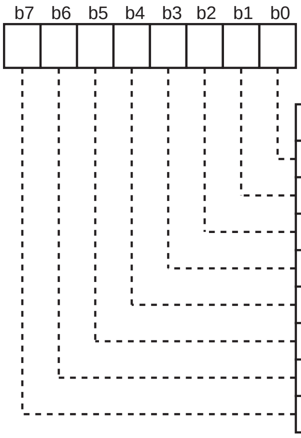


Figure 92. Programmable I/O ports (3)

Port Pi direction register (Note 1)



Symbol	Address	When reset
PDi (i = 0 to 7)	$03E2_{16}$, $03E3_{16}$, $03E6_{16}$, $03E7_{16}$	00_{16}
	$03EA_{16}$, $03EB_{16}$, $03EE_{16}$, $03EF_{16}$	00_{16}

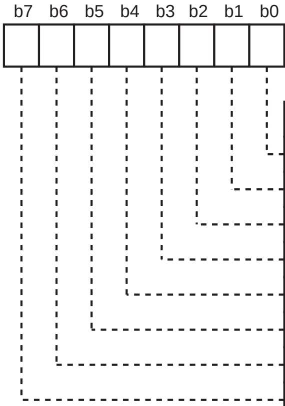
Bit symbol	Bit name	Function	R	W
PDi_0	Port Pi_0 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 0 to 7)	○	○
PDi_1	Port Pi_1 direction register		○	○
PDi_2	Port Pi_2 direction register		○	○
PDi_3	Port Pi_3 direction register		○	○
PDi_4	Port Pi_4 direction register		○	○
PDi_5	Port Pi_5 direction register		○	○
PDi_6	Port Pi_6 direction register		○	○
PDi_7	Port Pi_7 direction register		○	○

Note 1: Set bit 2 of protect register (address $000A_{16}$) to “1” before rewriting to the port P4 direction register.

Note 2: Nothing is assigned in direction register of $P3_6$, $P3_7$, $P4_6$, $P4_7$, $P5_5$ to $p5_7$, $P7_2$ to $P7_7$. These bits can either be set nor reset. When read, its contents are indeterminate.

Figure 93. Direction register

Port Pi register



Symbol	Address	When reset
Pi (i = 0 to 7)	03E0$_{16}$, 03E1$_{16}$, 03E4$_{16}$, 03E5$_{16}$	Indeterminate
	03E8$_{16}$, 03E9$_{16}$, 03EC$_{16}$, 03ED$_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
Pi_0	Port Pi$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : “L” level data 1 : “H” level data (i = 0 to 7)	○	○
Pi_1	Port Pi$_1$ register		○	○
Pi_2	Port Pi$_2$ register		○	○
Pi_3	Port Pi$_3$ register		○	○
Pi_4	Port Pi$_4$ register		○	○
Pi_5	Port Pi$_5$ register		○	○
Pi_6	Port Pi$_6$ register		○	○
Pi_7	Port Pi$_7$ register		○	○

Note: Nothing is assigned in direction register of P3$_6$, P3$_7$, P4$_6$, P4$_7$, P5$_5$ to p5$_7$, P7$_2$ to P7$_7$. This bit can either be set nor reset. When read, its content is indeterminate.

Figure 94. Port register

Pull-up control register 0

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUR0	03FC$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
PU00	P0$_0$ to P0$_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU01	P0$_4$ to P0$_7$ pull-up		○	○
PU02	P1$_0$ to P1$_3$ pull-up		○	○
PU03	P1$_4$ to P1$_7$ pull-up		○	○
—	—		○	○
—	—		○	○
PU06	P3$_0$ to P3$_3$ pull-up		○	○
PU07	P3$_4$ to P3$_5$ pull-up		○	○

Pull-up control register 1

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUR1	03FD$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
PU10	P4$_0$ to P4$_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU11	P4$_4$ to P4$_7$ pull-up		○	○
PU12	P5$_0$ to P5$_3$ pull-up		○	○
PU13	P5$_4$ pull-up		○	○
PU14	P6$_0$ to P6$_3$ pull-up		○	○
PU15	P6$_4$ to P6$_7$ pull-up		○	○
PU16	P7$_0$ to P7$_1$ pull-up		○	○
—	—		○	○

Port P1 drive capacity control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
DRR	03FE$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
DRR0	Port P1$_0$ drive capacuty	Set P1 N-channel output transistor drive capacity 0 : LOW 1 : HIGH	○	○
DRR1	Port P1$_1$ drive capacuty		○	○
DRR2	Port P1$_2$ drive capacuty		○	○
DRR3	Port P1$_3$ drive capacuty		○	○
DRR4	Port P1$_4$ drive capacuty		○	○
DRR5	Port P1$_5$ drive capacuty		○	○
DRR6	Port P1$_6$ drive capacuty		○	○
DRR7	Port P1$_7$ drive capacuty		○	○

Figure 95. Pull-up control register

Example connection of unused pins

Table 36. Example connection of unused pins

Pin name	Connection
Ports P0, P1, P3 to P7	After setting for input mode, connect every pin to VSS (pull-down); or after setting for output mode, leave these pins open.
XOUT (Note)	Open
AVCC	Connect to VCC
AVSS, VREF	Connect to VSS

Note: With external clock input to XIN pin.

Usage Precaution

Timer A (timer mode)

(1) Reading the timer A0 register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer A0 register with the reload timing gets "FFFF$_{16}$". Reading the timer A0 register after setting a value in the timer A0 register with a count halted but before the counter starts counting gets a proper value.

Timer A (event counter mode)

(1) Reading the timer A0 register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer A0 register with the reload timing gets "FFFF$_{16}$" by underflow or "0000$_{16}$" by overflow. Reading the timer A0 register after setting a value in the timer A0 register with a count halted but before the counter starts counting gets a proper value.

(2) When stop counting in free run type, set timer again.

Timer A (one-shot timer mode)

(1) Setting the count start flag to "0" while a count is in progress causes as follows:
- The counter stops counting and a content of reload register is reloaded.
- The TA0$_{OUT}$ pin outputs "L" level.
- The interrupt request generated and the timer A0 interrupt request bit goes to "1".

(2) The timer A0 interrupt request bit goes to "1" if the timer's operation mode is set using any of the following procedures:
- Selecting one-shot timer mode after reset.
- Changing operation mode from timer mode to one-shot timer mode.
- Changing operation mode from event counter mode to one-shot timer mode.

Therefore, to use timer A0 interrupt (interrupt request bit), set timer A0 interrupt request bit to "0" after the above listed changes have been made.

Timer A (pulse width modulation mode)

(1) The timer A0 interrupt request bit becomes "1" if setting operation mode of the timer in compliance with any of the following procedures:
- Selecting PWM mode after reset.
- Changing operation mode from timer mode to PWM mode.
- Changing operation mode from event counter mode to PWM mode.

Therefore, to use timer A0 interrupt (interrupt request bit), set timer A0 interrupt request bit to "0" after the above listed changes have been made.

(2) Setting the count start flag to "0" while PWM pulses are being output causes the counter to stop counting. If the TA0$_{OUT}$ pin is outputting an "H" level in this instance, the output level goes to "L", and the timer A0 interrupt request bit goes to "1". If the TA0$_{OUT}$ pin is outputting an "L" level in this instance, the level does not change, and the timer A0 interrupt request bit does not becomes "1".

Timer B (timer mode, event counter mode)

(1) Reading the timer Bi register while a count is in progress allows reading , with arbitrary timing, the value of the counter. Reading the timer Bi register with the reload timing gets “FFFF$_{16}$”. Reading the timer Bi register after setting a value in the timer Bi register with a count halted but before the counter starts counting gets a proper value.

Timer B (pulse period/pulse width measurement mode)

(1) If changing the measurement mode select bit is set after a count is started, the timer Bi interrupt request bit goes to “1”.

(2) When the first effective edge is input after a count is started, an indeterminate value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.

Timer X (timer mode)

(1) Reading the timer Xi register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Xi register with the reload timing gets “FFFF$_{16}$”. Reading the timer A0 register after setting a value in the timer Xi register with a count halted but before the counter starts counting gets a proper value.

Timer X (event counter mode)

(1) Reading the timer Xi register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Xi register with the reload timing gets “FFFF$_{16}$” by underflow or “0000$_{16}$” by overflow. Reading the timer Xi register after setting a value in the timer Xi register with a count halted but before the counter starts counting gets a proper value.

(2) When stop counting in free run type, set timer again.

Timer X (one-shot timer mode)

(1) Setting the count start flag to “0” while a count is in progress causes as follows:

- The counter stops counting and a content of reload register is reloaded.
- The TXiINOUT pin outputs “L” level.
- The interrupt request generated and the timer Xi interrupt request bit goes to “1”.

(2) The timer Xi interrupt request bit goes to “1” if the timer's operation mode is set using any of the following procedures:

- Selecting one-shot timer mode after reset.
- Changing operation mode from timer mode to one-shot timer mode.
- Changing operation mode from event counter mode to one-shot timer mode.

Therefore, to use timer Xi interrupt (interrupt request bit), set timer Xi interrupt request bit to “0” after the above listed changes have been made.

Timer X (pulse width modulation mode)

(1) The timer Xi interrupt request bit becomes “1” if setting operation mode of the timer in compliance with any of the following procedures:

- Selecting PWM mode after reset.
- Changing operation mode from timer mode to PWM mode.
- Changing operation mode from event counter mode to PWM mode.

Therefore, to use timer Xi interrupt (interrupt request bit), set timer Xi interrupt request bit to “0” after the above listed changes have been made.

(2) Setting the count start flag to “0” while PWM pulses are being output causes the counter to stop counting. If the TXiINOUT pin is outputting an “H” level in this instance, the output level goes to “L”, and the timer Xi interrupt request bit goes to “1”. If the TXiINOUT pin is outputting an “L” level in this instance, the level does not change, and the timer Xi interrupt request bit does not becomes “1”.

Timer X (pulse period/pulse width measurement mode)

(1) If changing the measurement mode select bit is set after a count is started, the timer Xi interrupt request bit goes to “1”.

(2) When the first effective edge is input after a count is started, an indeterminate value is transferred to the reload register. At this time, timer Xi interrupt request is not generated.

A-D Converter

(1) Write to each bit (except bit 6) of A-D control register 0, to each bit of A-D control register 1, and to bit 0 of A-D control register 2 when A-D conversion is stopped (before a trigger occurs).
In particular, when the Vref connection bit is changed from “0” to “1”, start A-D conversion after an elapse of 1 µs or longer.

(2) When changing A-D operation mode, select analog input pin again.

(3) Using one-shot mode or single sweep mode
Read the correspondence A-D register after confirming A-D conversion is finished. (It is known by A-D conversion interrupt request bit.)

(4) Using repeat mode, repeat sweep mode 0 or repeat sweep mode 1
Use the undivided main clock as the internal CPU clock.

Stop Mode and Wait Mode

(1) When returning from stop mode by hardware reset, $\overline{\text{RESET}}$ pin must be set to “L” level until main clock oscillation is stabilized.

(2) When shifting to WAIT mode or STOP mode, the program stops after reading 8 bytes from the WAIT instruction and the instruction that sets all clock stop bits to “1” in the instruction queue. Therefore, insert a minimum of 8 NOPs after the WAIT instruction and the instruction that sets all clock stop bits to “1”.

Interrupts

(1) Reading address 00000_{16}

- 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_{16} will then be set to “0”.

  Reading address 00000_{16} 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_{16} by software.

(2) Setting the stack pointer

- The value of the stack pointer immediately after reset is initialized to 0000_{16}. 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.

  Concerning the first instruction immediately after reset, generating any interrupt is prohibited.

(3) External interrupt

- When changing a polarity of pins $\overline{\mathrm{INT0}}$ and $\overline{\mathrm{INT1}}$, the interrupt request bit may become "1". Clear the interrupt request bit after changing the polarity.

(4) Changing interrupt control register

See "Changing Interrupt Control Register".

Electrical characteristics

Table 37. Absolute maximum ratings

Symbol	Parameter		Condition	Rated value	Unit
Vcc	Supply voltage			- 0.3 to 7	V
AVcc	Analog supply voltage			- 0.3 to 7	V
V_I	Input voltage	$\overline{\text{RESET}}$, CNVss, $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P3_0$ to $P3_5$, $P4_0$ to $P4_5$, $P5_0$ to $P5_4$, $P6_0$ to $P6_7$, $P7_0$, $P7_1$, V_{REF}, X_{IN}		- 0.3 to Vcc + 0.3 (Note 1)	V
V_O	Output voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P3_0$ to $P3_5$, $P4_0$ to $P4_5$, $P5_0$ to $P5_4$, $P6_0$ to $P6_7$, $P7_0$, $P7_1$, V_{REF}, X_{IN}		- 0.3 to Vcc + 0.3	V
P_d	Power dissipation		Ta = 25 °C	1000 (Note 2)	mW
T_{opr}	Operating ambient temperature			- 20 to 85 (Note 3)	°C
T_{stg}	Storage temperature			- 40 to 150 (Note 4)	°C

Note 1: When writing to frash MCU, CNVss is –0.3 to 13 (V) .

Note 2: Flat package (56P6S-A) is 300 mW.

Note 3: Extended operating temperature version: -40 to 85 °C.

Note 4: Extended operating temperature version: -65 to 150 °C.

VCC = 5V

Table 38. Recommended operating conditions (Note 1)

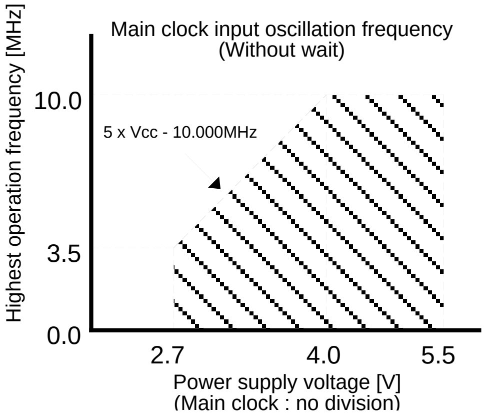
Symbol	Parameter				Standard			Unit
					Min	Typ.	Max.	
Vcc	Supply voltage (Note 2)	Mask ROM version			2.7	5.0	5.5	V
		Flash memory version			4.0	5.0	5.5	
AVcc	Analog supply voltage					Vcc		V
Vss	Supply voltage					0		V
AVss	Analog supply voltage					0		V
V_{IH}	HIGH input voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P3_0$ to $P3_5$, $P4_0$ to $P4_5$, $P5_0$ to $P5_4$, $P6_0$ to $P6_7$, $P7_0$, $P7_1$, X_{IN}, RESET, CNVss,			0.8Vcc		Vcc	V
V_{IL}	LOW input voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P3_0$ to $P3_5$, $P4_0$ to $P4_5$, $P5_0$ to $P5_4$, $P6_0$ to $P6_7$, $P7_0$, $P7_1$, X_{IN}, RESET, CNVss			0		0.2Vcc	V
$I_{OH (peak)}$	HIGH peak output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P3_0$ to $P3_5$, $P4_0$ to $P4_5$, $P5_0$ to $P5_4$, $P6_0$ to $P6_7$, $P7_0$, $P7_1$					- 10.0	mA
$I_{OL (peak)}$	LOW peak output current	$P0_0$ to $P0_7$, $P3_0$ to $P3_5$, $P4_0$ to $P4_5$, $P5_0$ to $P5_4$, $P6_0$ to $P6_7$, $P7_0$, $P7_1$					10.0	mA
$I_{OL (peak)}$	LOW peak output current	$P1_0$ to $P1_7$	HIGHPOWER				30.0	mA
			LOWPOWER				10.0	
$I_{OH (avg)}$	HIGH average output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P3_0$ to $P3_5$, $P4_0$ to $P4_5$, $P5_0$ to $P5_4$, $P6_0$ to $P6_7$, $P7_0$, $P7_1$					- 5.0	mA
$I_{OL (avg)}$	LOW average output current	$P0_0$ to $P0_7$, $P3_0$ to $P3_5$, $P4_0$ to $P4_5$, $P5_0$ to $P5_4$, $P6_0$ to $P6_7$, $P7_0$, $P7_1$					5.0	mA
$I_{OL (avg)}$	LOW average output current	$P1_0$ to $P1_7$	HIGHPOWER				15.0	mA
			LOWPOWER				5.0	
$f(X_{IN})$	Main clock input oscillation frequency	Without wait	Mask ROM version	V_{CC}=4.0V to 5.5V	0		10	MHz
				V_{CC}=2.7V to 4.0V	0		5 x VCC - 10.000	MHz
			Flash memory version	V_{CC}=4.0V to 5.5V	0		10	MHz
		With wait	Mask ROM version	V_{CC}=4.0V to 5.5V	0		10	MHz
				V_{CC}=2.7V to 4.0V	0		2.31 x VCC +0.760	MHz
			Flash memory version	V_{CC}=4.0V to 5.5V	0		10	MHz
$f(X_{CIN})$	Subclock oscillation frequency					32.768	50	kHz

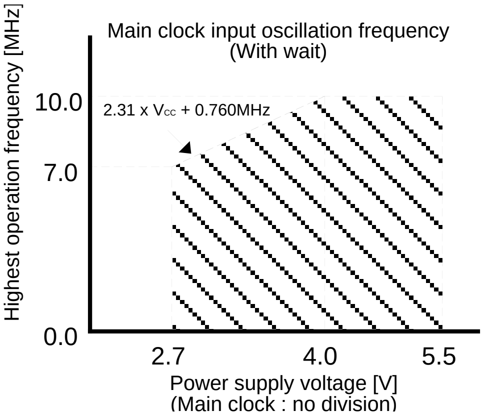
Note 1: Unless otherwise noted: VCC = 2.7V to 5.5V, Vss = 0V, Ta = – 20 to 85°C (Extended operating temperature version:– 40 to 85°C). Flash version: VCC = 4.0V to 5.5V, Vss = 0V, Ta = – 20 to 85°C (Extended operating temperature version:– 40 to 85°C.)

Note 2: Flash version: VCC = 4.0V to 5.5V

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

Note 4: Keep output current as follows:
The sum of port P3 and P4 I_{OL} (peak) is under 40 mA. The sum of port P1 I_{OL} (peak) is under 60 mA. The sum of port P1, P3 and P4 I_{OH} (peak) is under 40 mA. The sum of port P0, P5, P6 and P7 I_{OL} (peak) is under 80 mA. The sum of port P0, P5, P6 and P7 I_{OH} (peak) is under 80 mA.





$V_{CC} = 5V$

Table 39. Electrical characteristics (Note)

Symbol	Parameter			Measuring condition		Standard			Unit
						Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$,$P7_0$,$P7_1$		I_{OH} = - 5 mA		3.0			V
V_{OH}	HIGH output voltage	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$,$P7_0$,$P7_1$		I_{OH} = - 200 µA		4.7			V
V_{OH}	HIGH output voltage	X_{OUT}	HIGHPOWER	I_{OH} = - 1 mA		3.0			V
			LOWPOWER	I_{OH} = - 0.5 mA		3.0			
V_{OH}	HIGH output voltage	X_{COUT}	HIGHPOWER	No load			3.0		V
			LOWPOWER	No load			1.6		
V_{OL}	LOW output voltage	$P0_0$ to $P0_7$,$P3_0$ to $P3_5$,$P4_0$ to $P4_5$ $P5_0$ to $P5_4$,$P6_0$ to $P6_7$,$P7_0$,$P7_1$		I_{OL} = 5 mA				2.0	V
V_{OL}	LOW output voltage	$P0_0$ to $P0_7$,$P3_0$ to $P3_5$,$P4_0$ to $P4_5$ $P5_0$ to $P5_4$,$P6_0$ to $P6_7$,$P7_0$,$P7_1$		I_{OL} = 200 µA				0.45	V
V_{OL}	LOW output voltage	$P1_0$ to $P1_7$	HIGHPOWER	I_{OL} = 15mA				2.0	V
			LOWPOWER	I_{OL} = 5 mA				2.0	
V_{OL}	LOW output voltage	$P1_0$ to $P1_7$	HIGHPOWER	I_{OL} = 200 µA				0.3	V
			LOWPOWER	I_{OL} = 200 µA				0.45	
V_{OL}	LOW output voltage	X_{OUT}	HIGHPOWER	I_{OH} = 1 mA				2.0	V
			LOWPOWER	I_{OH} = 0.5 mA				2.0	
V_{OL}	LOW output voltage	X_{OUT}	HIGHPOWER	No load			0		V
			LOWPOWER	No load			0		
V_{T+} -V_{T-}	Hysteresis	$TA0_{IN}$,$TX0_{INOUT}$,$TX1_{INOUT}$ $TX2_{INOUT}$,$TB0_{IN}$,$TB1_{IN}$ $\overline{INT_0}$,$\overline{INT_1}$, CLK_0				0.2		0.8	V
V_{T+} -V_{T-}	Hysteresis	$\overline{RESET}$				0.2		1.8	V
I_{IH}	HIGH input current	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$ $P7_0$,$P7_1$,X_{IN}, RESET, CNVss		V_I = 5V				5.0	µA
I_{IL}	LOW input current	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$, $P7_0$,$P7_1$,X_{IN}, RESET, CNVss		V_I = 0V				-5.0	µA
R_{PULLUP}	Pull-up resister	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$,$P7_0$,$P7_1$		V_I = 0V		30.0	50.0	167.0	kΩ
R_{XIN}	Feedback resister	X_{IN}					1.0		MΩ
R_{XCIN}	Feedback resister	X_{CIN}					6.0		MΩ
V_{RAM}	RAM retention voltage			When clock is stopped		2.0			V
Icc	Power supply current			I/O pin has no load	f(X_{IN})=10MHz Square wave, no division		19.0	38.0	mA
					f(X_{CIN})=32kHz Square wave		90.0		µA
					f(X_{CIN})=32kHz With wait		4.0		µA
					Ta=25°C when clock is stopped			1.0	µA
					Ta=85°C when clock is stopped			20.0	

Note: Unless otherwise noted: V_{CC} = 5V, V_{SS} = 0V at Ta = 25°C, f(X_{IN}) = 10MHz)

VCC = 5V

Table 40. A-D conversion characteristics

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
–	Resolution		VREF =VCC			10	Bits
–	Absolute accuracy	Sample & hold function not available	VREF =VCC = 5V			±3	LSB
		Sample & hold function available(10bit)	VREF =VCC= 5V			±3	LSB
		Sample & hold function available(8bit)	VREF = VCC = 5V			±2	LSB
RLADDER	Ladder resistance		VREF =VCC	10		40	kohm
tCONV	Conversion time(10bit)			3.3			µs
tCONV	Conversion time(8bit)			2.8			µs
tSAMP	Sampling time			0.3			µs
VREF	Reference voltage			2		VCC	V
VIA	Analog input voltage			0		VREF	V

V_{CC} = 5V

Timing requirements (referenced to V_{CC} = 5V, V_{SS} = 0V at Ta = 25°C unless otherwise specified)

Table 41. 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

Table 42. Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	$TA0_{IN}$ input cycle time	100		ns
$t_{w(TAH)}$	$TA0_{IN}$ input HIGH pulse width	40		ns
$t_{w(TAL)}$	$TA0_{IN}$ input LOW pulse width	40		ns

Table 43. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	$TA0_{IN}$ input cycle time	400		ns
$t_{w(TAH)}$	$TA0_{IN}$ input HIGH pulse width	200		ns
$t_{w(TAL)}$	$TA0_{IN}$ input LOW pulse width	200		ns

Table 44. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	$TA0_{IN}$ input cycle time	200		ns
$t_{w(TAH)}$	$TA0_{IN}$ input HIGH pulse width	100		ns
$t_{w(TAL)}$	$TA0_{IN}$ input LOW pulse width	100		ns

Table 45. Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	$TA0_{IN}$ input HIGH pulse width	100		ns
$t_{w(TAL)}$	$TA0_{IN}$ input LOW pulse width	100		ns

Table 46. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	$TA0_{OUT}$ input cycle time	2000		ns
$t_{w(UPH)}$	$TA0_{OUT}$ input HIGH pulse width	1000		ns
$t_{w(UPL)}$	$TA0_{OUT}$ input LOW pulse width	1000		ns
$t_{su(UP-T_{IN})}$	$TA0_{OUT}$ input setup time	400		ns
$t_{h(T_{IN}-UP)}$	$TA0_{OUT}$ input hold time	400		ns

VCC = 5V

Timing requirements (referenced to VCC = 5V, VSS = 0V at Ta = 25°C unless otherwise specified)

Table 47. 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 48. 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 49. 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 50. Timer X input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	100		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	40		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	40		ns

Table 51. Timer X input (gate input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	400		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	200		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	200		ns

Table 52. Timer X input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	200		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	100		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	100		ns

VCC = 5V

Timing requirements (referenced to VCC = 5V, VSS = 0V at Ta = 25°C unless otherwise specified)

Table 53. Timer X input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	400		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	200		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	200		ns

Table 54. Timer X input (pulse width measurement mode)

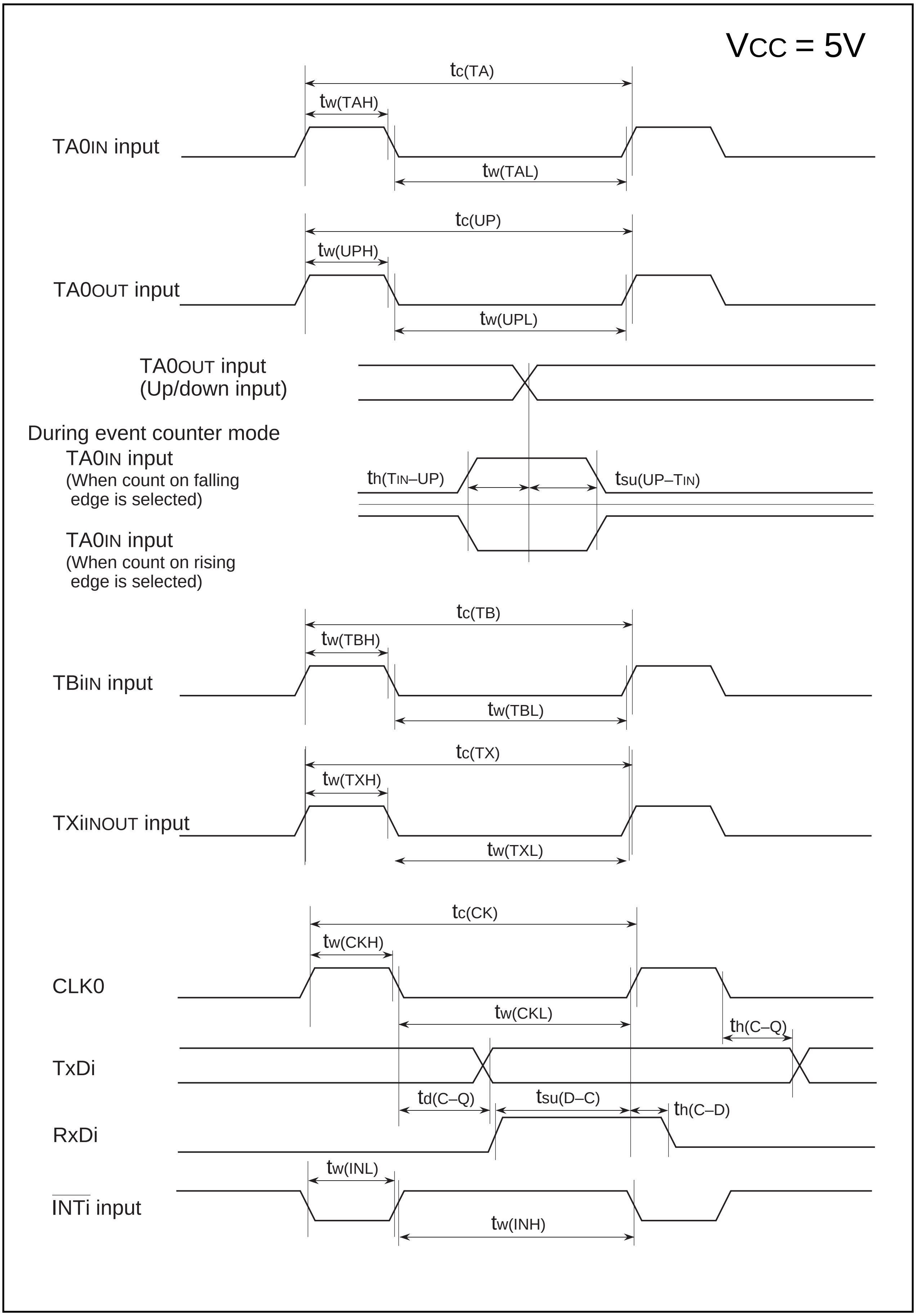
Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	400		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	200		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	200		ns

Table 55. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLK0 input cycle time	200		ns
$t_{w(CKH)}$	CLK0 input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLK0 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 56. External interrupt $\overline{\text{INTi}}$ inputs

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


VCC = 5V
tc(TA)
tw(TAH)
TA0IN input
tw(TAL)
tc(UP)
tw(UPH)
TA0OUT input
tw(UPL)
TA0OUT input
(Up/down input)
During event counter mode
TA0IN input
(When count on falling
edge is selected)
th(TIN–UP)
tsu(UP–TIN)
TA0IN input
(When count on rising
edge is selected)
tc(TB)
tw(TBH)
TBiIN input
tw(TBL)
tc(TX)
tw(TXH)
TXiINOUT input
tw(TXL)
tc(CK)
tw(CKH)
CLK0
tw(CKL)
th(C–Q)
TxDi
td(C–Q)
tsu(D–C)
th(C–D)
RxDi
tw(INL)
INTi input
tw(INH)

$V_{CC} = 3V$

Table 57. Electrical characteristics (Note 1)

Symbol	Parameter			Measuring condition		Standard			Unit
						Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$,$P7_0$,$P7_1$		I_{OH} = - 1mA		2.5			V
V_{OH}	HIGH output voltage	X_{OUT}	HIGHPOWER	I_{OH} = - 1 mA		2.5			V
			LOWPOWER	I_{OH} = - 50 µA		2.5			
V_{OH}	HIGH output voltage	X_{COUT}	HIGHPOWER	No load			3.0		V
			LOWPOWER	No load			1.6		
V_{OL}	LOW output voltage	$P0_0$ to $P0_7$,$P3_0$ to $P3_5$,$P4_0$ to $P4_5$ $P5_0$ to $P5_4$,$P6_0$ to $P6_7$,$P7_0$,$P7_1$		I_{OL} = 1 mA				0.5	V
V_{OL}	LOW output voltage	$P1_0$ to $P1_7$	HIGHPOWER	I_{OL} = 3 mA				0.5	V
			LOWPOWER	I_{OL} = 1 mA				0.5	
V_{OL}	LOW output voltage	X_{OUT}	HIGHPOWER	I_{OH} = 0.1 mA				0.5	V
			LOWPOWER	I_{OH} = 50 µA				0.5	
V_{OL}	LOW output voltage	X_{OUT}	HIGHPOWER	No load			0		V
			LOWPOWER	No load			0		
V_{T+} -V_{T-}	Hysteresis	$TA0_{IN}$,$TX0_{INOUT}$,$TX1_{INOUT}$, $TX2_{INOUT}$,$TB0_{IN}$,$TB1_{IN}$ $\overline{INT_0}$,$\overline{INT_1}$, CLK_0				0.2		0.8	V
V_{T+} -V_{T-}	Hysteresis	$\overline{RESET}$				0.2		1.8	V
I_{IH}	HIGH input current	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$, $P7_0$,$P7_1$,X_{IN}, RESET, CNVss		V_I = 3V				4.0	µA
I_{IL}	LOW input current	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$, $P7_0$,$P7_1$,X_{IN}, RESET, CNVss		V_I = 0V				-4.0	µA
R_{PULLUP}	Pull-up resistor	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$,$P7_0$,$P7_1$		V_I = 0V		66.0	120.0	500.0	kΩ
R_{XIN}	Feedback resistor	X_{IN}					3.0		MΩ
R_{XIN}	Feedback resistor	X_{IN}					10.0		MΩ
V_{RAM}	RAM retention voltage			When clock is stopped		2.0			V
Icc	Power supply current			I/O pin has no load	$f(X_{IN})$=7MHz Square wave, no division		6.0	15.0	mA
					$f(X_{CIN})$=32kHz Square wave		40.0		µA
					$f(X_{CIN})$=32kHz With wait. Oscillation capacity HIGH (Note 2)		2.8		µA
					$f(X_{CIN})$=32kHz With wait. Oscillation capacity LOW (Note 2)		0.9		µA
					Ta=25°C when clock is stopped			1.0	µA
					Ta=85°C when clock is stopped			20.0	

Note 1: Unless otherwise noted: V_{CC} = 3V, V_{SS} = 0V at Ta = 25°C, $f(X_{IN})$ = 7MHz, with wait)

Note 2: With one timer operated using f_{C32}.

$V_{CC} = 3V$

Table 58. A-D conversion characteristics

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
–	Resolution		$V_{REF} = V_{CC}$			10	Bits
–	Absolute accuracy	Sample & hold function not available (8bit)	$V_{REF} = V_{CC} = 3V$, Ø$_{AD}$ = f$_{AD}$/2			±2	LSB
R_{LADDER}	Ladder resistance		$V_{REF} = V_{CC}$	10		40	kohm
t_{CONV}	Conversion time(8bit)			14.0			µs
V_{REF}	Reference voltage			2.7		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

VCC = 3V

Timing requirements (referenced to VCC = 3V, VSS = 0V at Ta = 25°C unless otherwise specified)

Table 59. External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	143		ns
$t_{w(H)}$	External clock input HIGH pulse width	60		ns
$t_{w(L)}$	External clock input LOW pulse width	60		ns
t_r	External clock rise time		18	ns
t_f	External clock fall time		18	ns

Table 60. Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0$_{IN}$ input cycle time	150		ns
$t_{w(TAH)}$	TA0$_{IN}$ input HIGH pulse width	60		ns
$t_{w(TAL)}$	TA0$_{IN}$ input LOW pulse width	60		ns

Table 61. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0$_{IN}$ input cycle time	600		ns
$t_{w(TAH)}$	TA0$_{IN}$ input HIGH pulse width	300		ns
$t_{w(TAL)}$	TA0$_{IN}$ input LOW pulse width	300		ns

Table 62. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0$_{IN}$ input cycle time	300		ns
$t_{w(TAH)}$	TA0$_{IN}$ input HIGH pulse width	150		ns
$t_{w(TAL)}$	TA0$_{IN}$ input LOW pulse width	150		ns

Table 63. Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TA0$_{IN}$ input HIGH pulse width	150		ns
$t_{w(TAL)}$	TA0$_{IN}$ input LOW pulse width	150		ns

Table 64. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TA0$_{OUT}$ input cycle time	3000		ns
$t_{w(UPH)}$	TA0$_{OUT}$ input HIGH pulse width	1500		ns
$t_{w(UPL)}$	TA0$_{OUT}$ input LOW pulse width	1500		ns
$t_{su(UP-TIN)}$	TA0$_{OUT}$ input setup time	600		ns
$t_{h(TIN-UP)}$	TA0$_{OUT}$ input hold time	600		ns

VCC = 3V

Timing requirements (referenced to VCC = 3V, VSS = 0V at Ta = 25°C unless otherwise specified)

Table 65. Timer B input (counter input in event counter mode)

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

Table 66. Timer B input (pulse period measurement mode)

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

Table 67. Timer B input (pulse width measurement mode)

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

Table 68. Timer X input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXi_{INOUT} input cycle time	150		ns
$t_{w(TXH)}$	TXi_{INOUT} input HIGH pulse width	60		ns
$t_{w(TXL)}$	TXi_{INOUT} input LOW pulse width	60		ns

Table 69. Timer X input (gate input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXi_{INOUT} input cycle time	600		ns
$t_{w(TXH)}$	TXi_{INOUT} input HIGH pulse width	300		ns
$t_{w(TXL)}$	TXi_{INOUT} input LOW pulse width	300		ns

Table 70. Timer X input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXi_{INOUT} input cycle time	300		ns
$t_{w(TXH)}$	TXi_{INOUT} input HIGH pulse width	150		ns
$t_{w(TXL)}$	TXi_{INOUT} input LOW pulse width	150		ns

VCC = 3V

Timing requirements (referenced to VCC = 3V, VSS = 0V at Ta = 25°C unless otherwise specified)

Table 71. Timer X input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	600		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	300		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	300		ns

Table 72. Timer X input (pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	600		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	300		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	300		ns

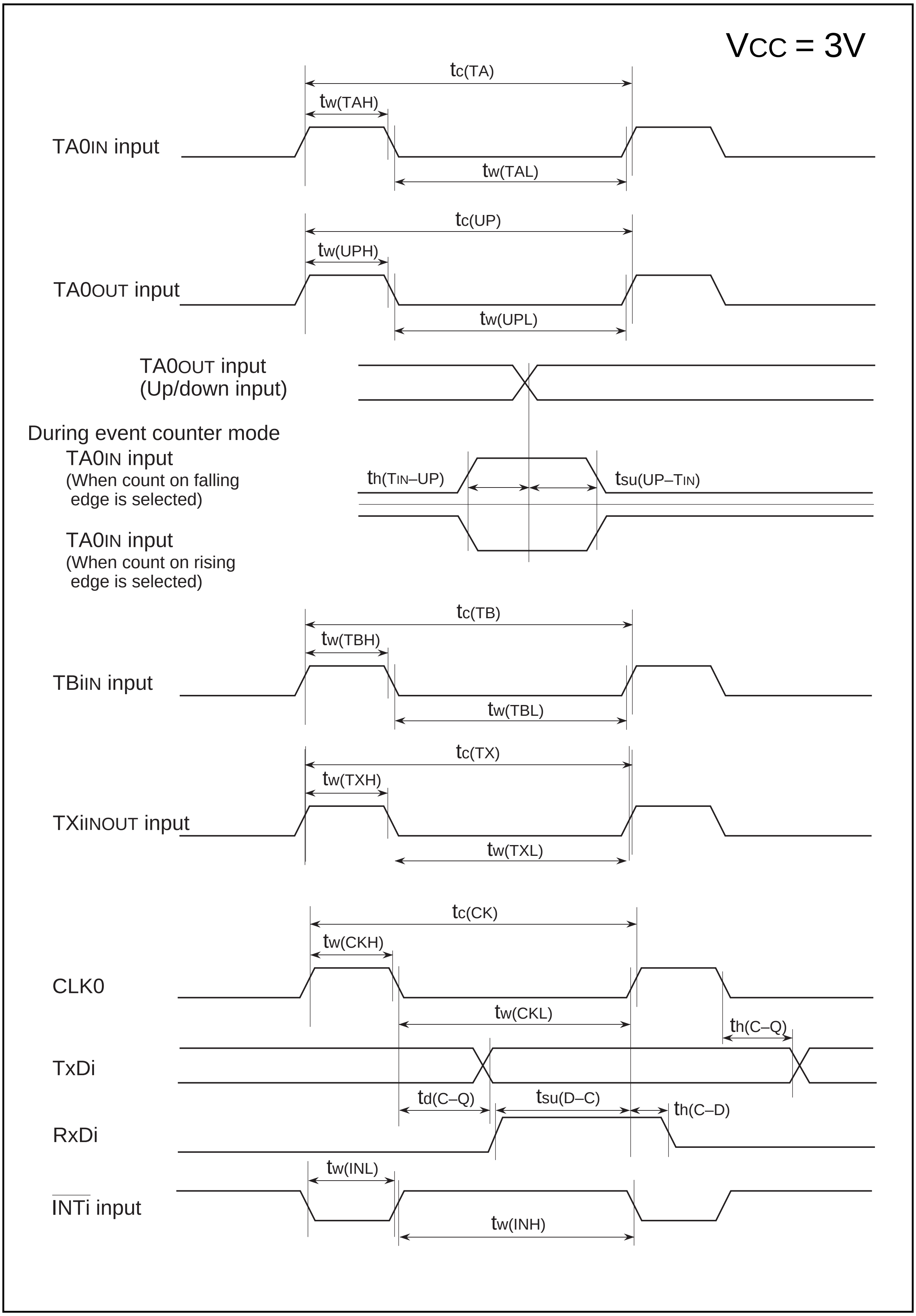
Table 73. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLK0 input cycle time	300		ns
$t_{w(CKH)}$	CLK0 input HIGH pulse width	150		ns
$t_{w(CKL)}$	CLK0 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 74. External interrupt $\overline{\text{INTi}}$ inputs

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

Electrical characteristics (Vcc = 3V)



MITSUBISHI SEMICONDUCTORS
M16C/20 Group DATA SHEET REV.B

April First Edition 1998
July Second Edition 1998

Editioned by
Committee of editing of Mitsubishi Semiconductor DATA SHEET

Published by
Mitsubishi Electric Corp., Kitaitami Works