To all our customers

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

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

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

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

Description

The M30201 group of single-chip microcomputers are built using the high-performance silicon gate CMOS process using a M16C/60 Series CPU core. M30201 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 M30201 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 1.4. ROM expansion.)
- Shortest instruction execution time ..... 100ns (f(XIN)=10MHz)
- Supply voltage ..... 4.0 to 5.5V (f(XIN)=10MHz) :mask ROM version
  2.7 to 5.5V (f(XIN)=3.5MHz):mask ROM version
  4.0 to 5.5V (f(XIN)=10MHz) :flash memory version
- Interrupts ..... 13 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

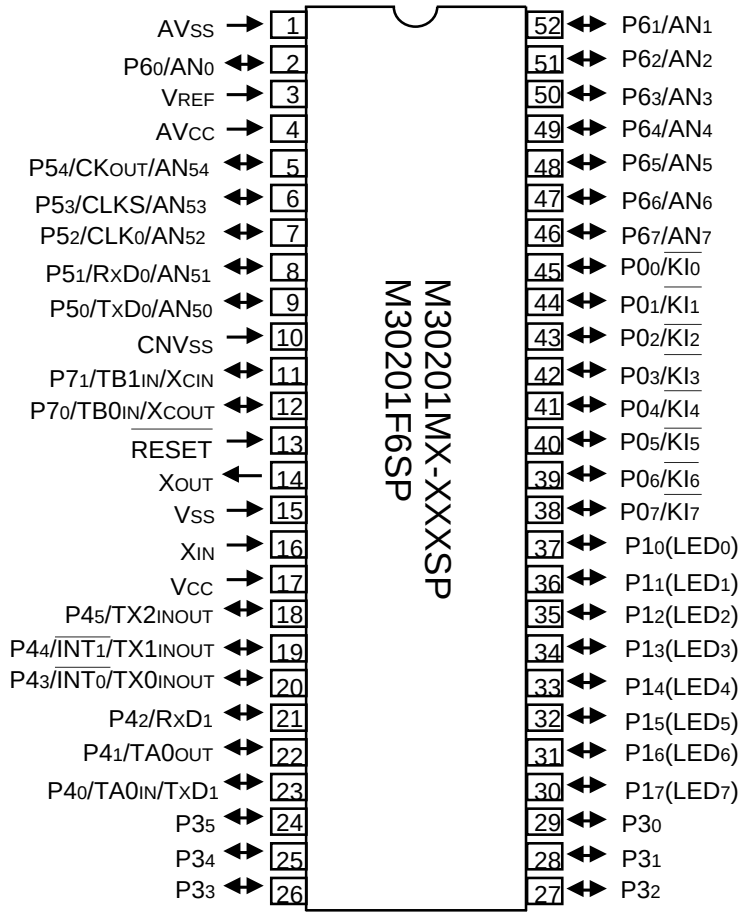
------Table of Contents------

Central Processing Unit (CPU) 12
Reset 15
Clock Generating Circuit 19
Protection 26
Interrupts 27
Watchdog Timer 45
Timer 47
Serial I/O 74
A-D Converter 88
Programmable I/O Ports 98
Electric Characteristics 110
Flash Memory version 124

Pin Configuration

Figures 1.1 to 1.2 show the pin configurations (top view).

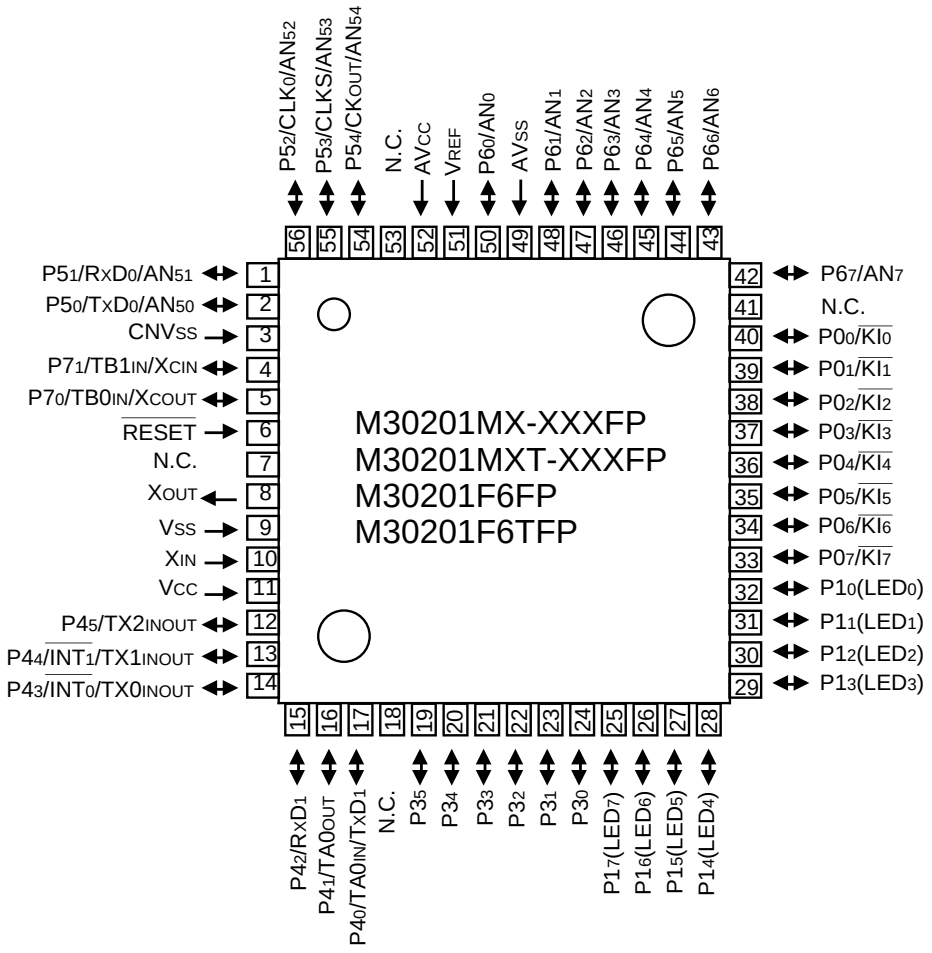
PIN CONFIGURATION (top view)



Package: 52P4B

Figure 1.1. Pin configuration for the M30201 group (shrink DIP product) (top view)

PIN CONFIGURATION (top view)

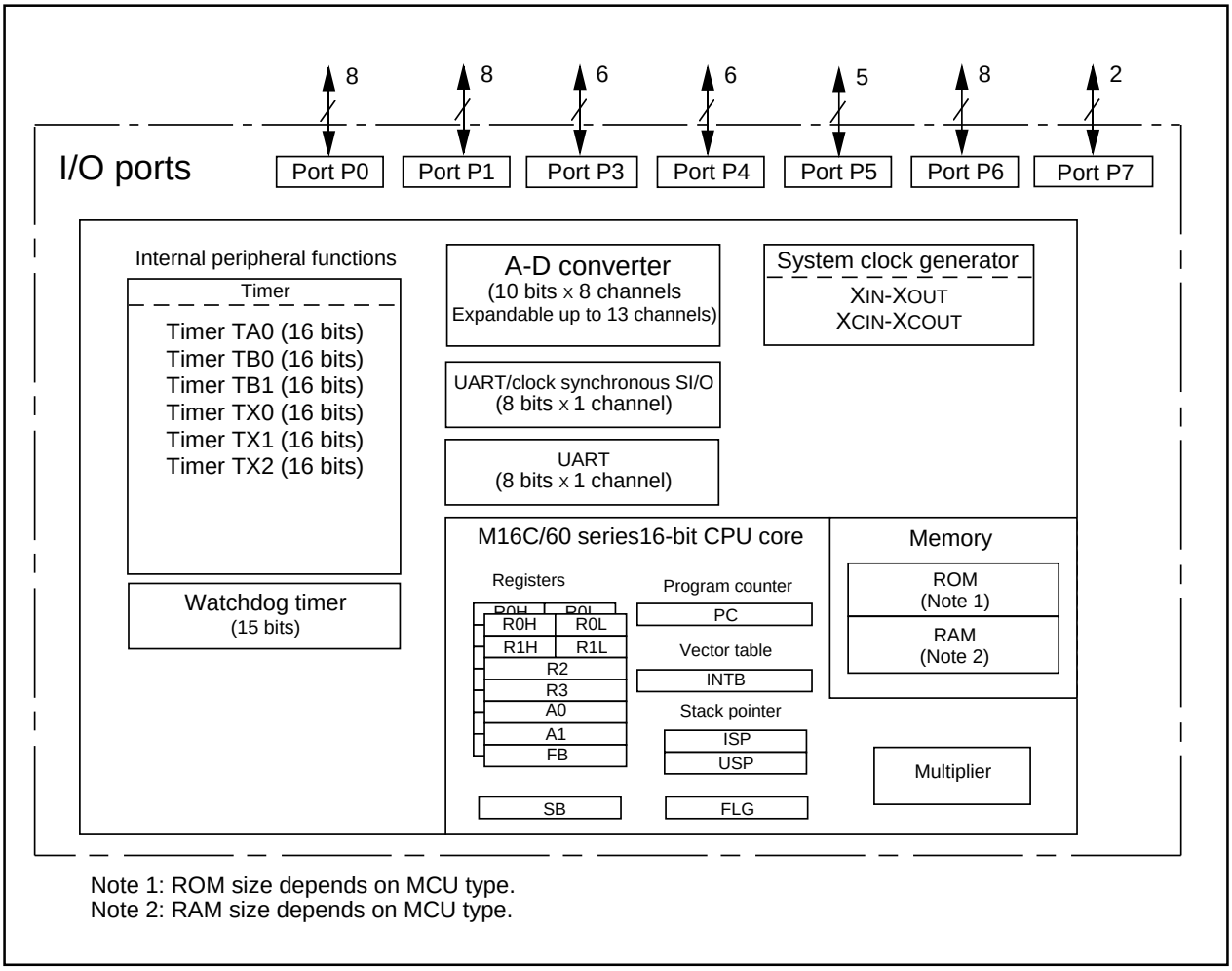


Package: 56P6S-A

Figure 1.2. Pin configuration for the M30201 group (QFP product) (top view)

Block Diagram

Figure 1.3 is a block diagram of the M30201 group.



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

Figure 1.3. Block diagram for the M30201 group

Performance Outline

Table 1.1 is performance outline of M30201 group.

Table 1.1. Performance outline of M30201 group

Item		Performance
Number of basic instructions		91 instructions
Shortest instruction execution time		100ns (f(XIN)=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		13 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(XIN)=10MHz) :mask ROM version 2.7 to 5.5V (f(XIN)=3.5MHz) :mask ROM version 4.0 to 5.5V (f(XIN)=10MHz) :flash memory version
Power consumption		11mW (f(XIN)=3.5MHz , Vcc=3V) :mask ROM version 95mW (f(XIN)=10MHz, Vcc=5V) :flash memory version
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

Description

Mitsubishi plans to release the following products in the M30201 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)

Apr. 2001

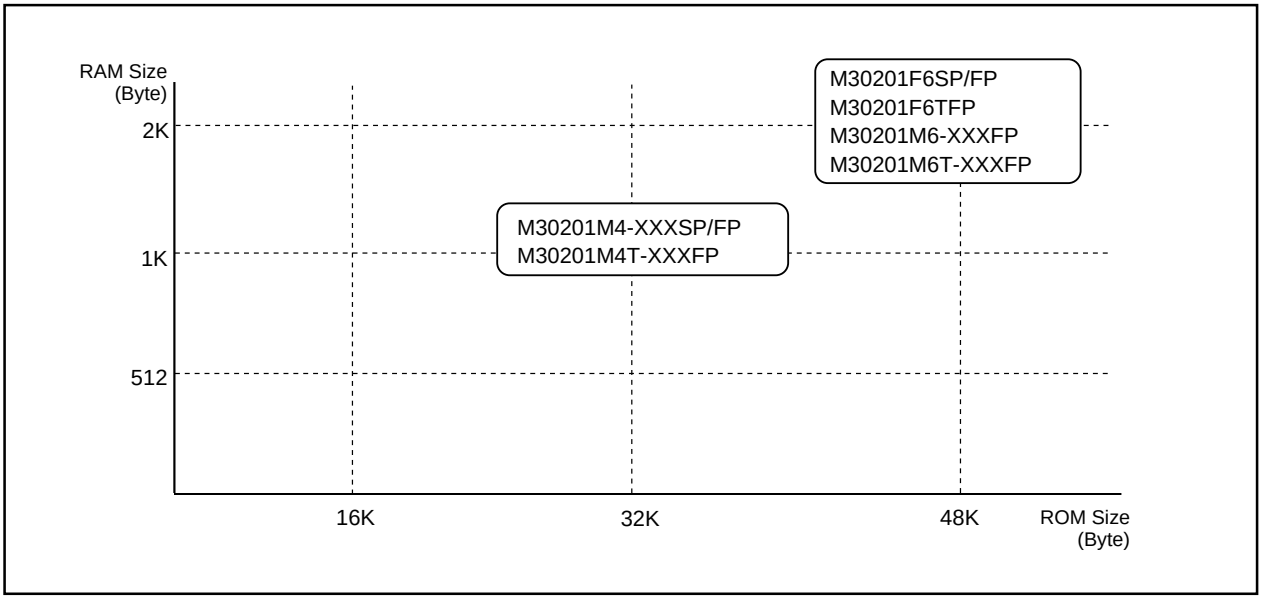


Figure 1.4. ROM expansion

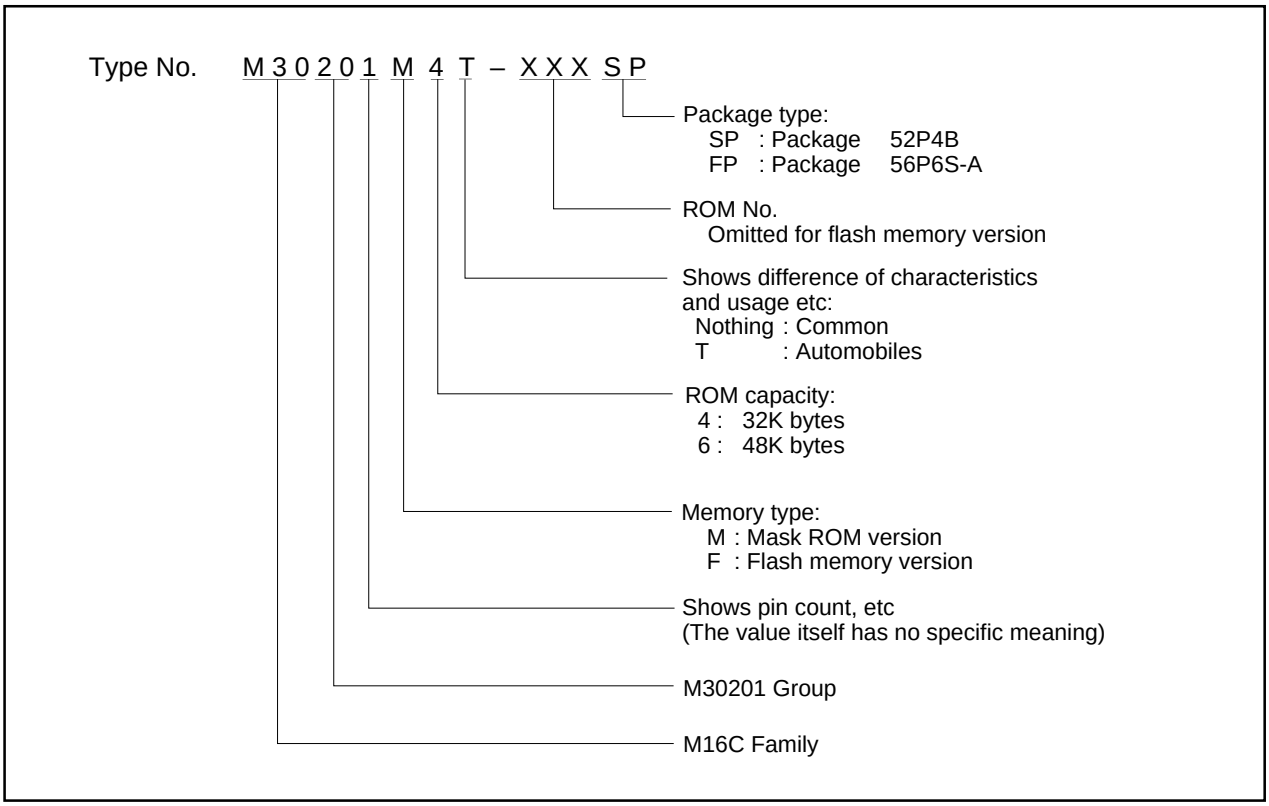


Figure 1.5. Type No., memory size, and package

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 1.6 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-XXXSP, 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-XXXSP, 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.

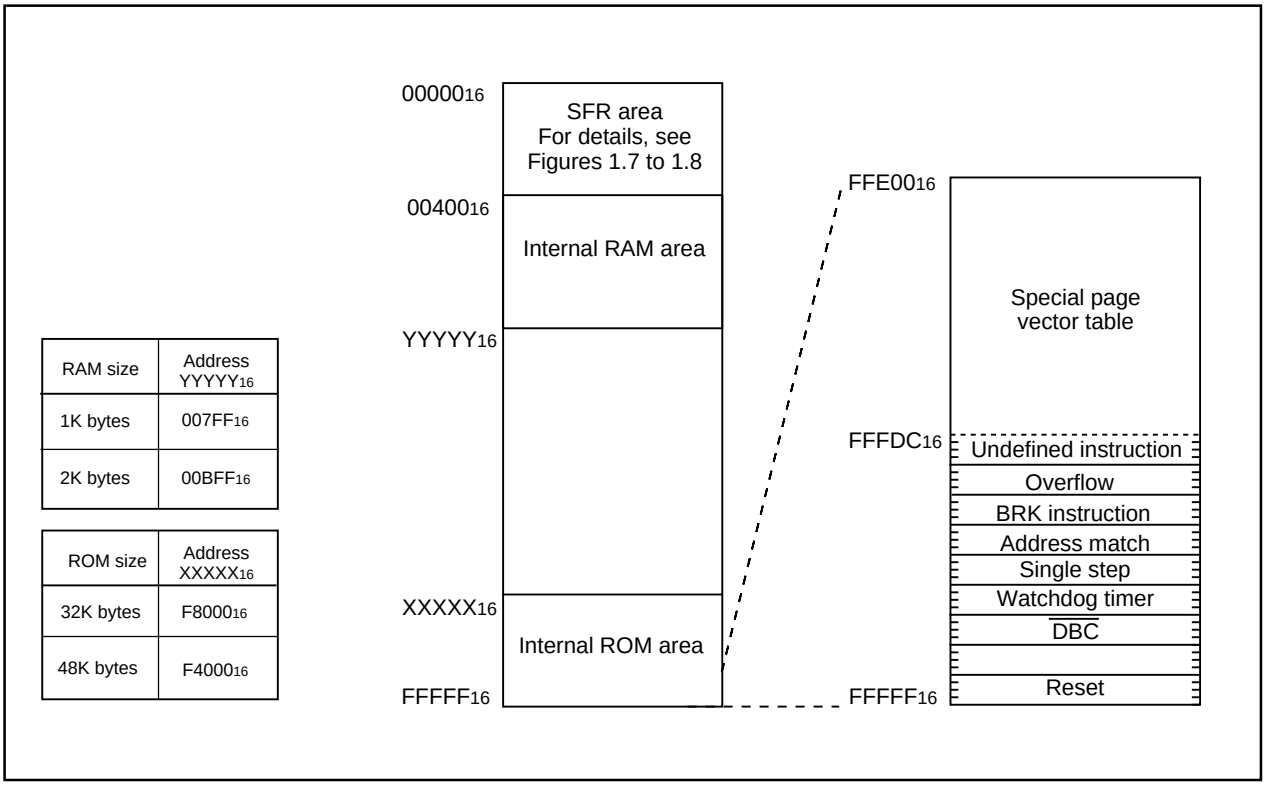


Figure 1.6. Memory map

0000₁₆	
0001₁₆	
0002₁₆	
0003₁₆	
0004₁₆	Processor mode register 0 (PM0)
0005₁₆	Processor mode register 1(PM1)
0006₁₆	System clock control register 0 (CM0)
0007₁₆	System clock control register 1 (CM1)
0008₁₆	
0009₁₆	Address match interrupt enable register (AIER)
000A₁₆	Protect register (PRCR)
000B₁₆	
000C₁₆	
000D₁₆	
000E₁₆	Watchdog timer start register (WDTS)
000F₁₆	Watchdog timer control register (WDC)
0010₁₆	Address match interrupt register 0 (RMAD0)
0011₁₆	
0012₁₆	
0013₁₆	
0014₁₆	Address match interrupt register 1 (RMAD1)
0015₁₆	
0016₁₆	
0017₁₆	
0018₁₆	
0019₁₆	
001A₁₆	
001B₁₆	
001C₁₆	
001D₁₆	
001E₁₆	
001F₁₆	
0020₁₆	
0021₁₆	
0022₁₆	
0023₁₆	
0024₁₆	
0025₁₆	
0026₁₆	
0027₁₆	
0028₁₆	
0029₁₆	
002A₁₆	
002B₁₆	
002C₁₆	
002D₁₆	
002E₁₆	
002F₁₆	
0030₁₆	
0031₁₆	
0032₁₆	
0033₁₆	
0034₁₆	
0035₁₆	
0036₁₆	
0037₁₆	
0038₁₆	
0039₁₆	
003A₁₆	
003B₁₆	
003C₁₆	
003D₁₆	
003E₁₆	
003F₁₆	

0040₁₆	
0041₁₆	
0042₁₆	
0043₁₆	
0044₁₆	
0045₁₆	
0046₁₆	
0047₁₆	
0048₁₆	
0049₁₆	
004A₁₆	
004B₁₆	
004C₁₆	
004D₁₆	Key input interrupt control register (KUPIC)
004E₁₆	A-D conversion interrupt control register (ADIC)
004F₁₆	
0050₁₆	
0051₁₆	UART0 transmit interrupt control register (S0TIC)
0052₁₆	UART0 receive interrupt control register (S0RIC)
0053₁₆	UART1 transmit interrupt control register (S1TIC)
0054₁₆	UART1 receive interrupt control register (S1RIC)
0055₁₆	Timer A0 interrupt control register (TA0IC)
0056₁₆	Timer X0 interrupt control register (TX0IC)
0057₁₆	Timer X1 interrupt control register (TX1IC)
0058₁₆	Timer X2 interrupt control register (TX2IC)
0059₁₆	
005A₁₆	Timer B0 interrupt control register (TB0IC)
005B₁₆	Timer B1 interrupt control register (TB1IC)
005C₁₆	
005D₁₆	INT0 interrupt control register (INT0IC)
005E₁₆	INT1 interrupt control register (INT1IC)
005F₁₆	

Note: Locations in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

Figure 1.7. Location of peripheral unit control registers (1)

0380₁₆	Count start flag (TABSR)
0381₁₆	Clock prescaler reset flag (CPSRF)
0382₁₆	One-shot start flag (ONSF)
0383₁₆	Trigger select register (TRGSR)
0384₁₆	Up-down flag (UDF)
0385₁₆	
0386₁₆	Timer A0 (TA0)
0387₁₆	
0388₁₆	Timer X0 (TX0)
0389₁₆	
038A₁₆	Timer X1 (TX1)
038B₁₆	
038C₁₆	Timer X2 (TX2)
038D₁₆	
038E₁₆	Clock divided counter (CDC)
038F₁₆	
0390₁₆	Timer B0 (TB0)
0391₁₆	
0392₁₆	Timer B1 (TB1)
0393₁₆	
0394₁₆	
0395₁₆	
0396₁₆	Timer A0 mode register (TA0MR)
0397₁₆	Timer X0 mode register (TX0MR)
0398₁₆	Timer X1 mode register (TX1MR)
0399₁₆	Timer X2 mode register (TX2MR)
039A₁₆	
039B₁₆	Timer B0 mode register (TB0MR)
039C₁₆	Timer B1 mode register (TB1MR)
039D₁₆	
039E₁₆	
039F₁₆	
03A0₁₆	UART0 transmit/receive mode register (U0MR)
03A1₁₆	UART0 bit rate generator (U0BRG)
03A2₁₆	UART0 transmit buffer register (U0TB)
03A3₁₆	
03A4₁₆	UART0 transmit/receive control register 0 (U0C0)
03A5₁₆	UART0 transmit/receive control register 1 (U0C1)
03A6₁₆	UART0 receive buffer register (U0RB)
03A7₁₆	
03A8₁₆	UART1 transmit/receive mode register (U1MR)
03A9₁₆	UART1 bit rate generator (U1BRG)
03AA₁₆	UART1 transmit buffer register (U1TB)
03AB₁₆	
03AC₁₆	UART1 transmit/receive control register 0 (U1C0)
03AD₁₆	UART1 transmit/receive control register 1 (U1C1)
03AE₁₆	UART1 receive buffer register (U1RB)
03AF₁₆	
03B0₁₆	UART transmit/receive control register 2 (UCON)
03B1₁₆	
03B2₁₆	
03B3₁₆	
03B4₁₆	Flash memory control register 0 (FCON0) (Note1)
03B5₁₆	Flash memory control register 1 (FCON1) (Note1)
03B6₁₆	Flash command register (FCMD) (Note)
03B7₁₆	
03B8₁₆	
03B9₁₆	
03BA₁₆	
03BB₁₆	
03BC₁₆	
03BD₁₆	
03BE₁₆	
03BF₁₆	

03C0₁₆	A-D register 0 (AD0)
03C1₁₆	
03C2₁₆	A-D register 1 (AD1)
03C3₁₆	
03C4₁₆	A-D register 2 (AD2)
03C5₁₆	
03C6₁₆	A-D register 3 (AD3)
03C7₁₆	
03C8₁₆	A-D register 4 (AD4)
03C9₁₆	
03CA₁₆	A-D register 5 (AD5)
03CB₁₆	
03CC₁₆	A-D register 6 (AD6)
03CD₁₆	
03CE₁₆	A-D register 7 (AD7)
03CF₁₆	
03D0₁₆	
03D1₁₆	
03D2₁₆	
03D3₁₆	
03D4₁₆	A-D control register 2 (ADCON2)
03D5₁₆	
03D6₁₆	A-D control register 0 (ADCON0)
03D7₁₆	A-D control register 1 (ADCON1)
03D8₁₆	
03D9₁₆	
03DA₁₆	
03DB₁₆	
03DC₁₆	
03DD₁₆	
03DE₁₆	
03DF₁₆	
03E0₁₆	Port P0 (P0)
03E1₁₆	Port P1 (P1)
03E2₁₆	Port P0 direction register (PD0)
03E3₁₆	Port P1 direction register (PD1)
03E4₁₆	Port P2 (P2) (Reserved)
03E5₁₆	Port P3 (P3)
03E6₁₆	Port P2 direction register (PD2) (Reserved)
03E7₁₆	Port P3 direction register (PD3)
03E8₁₆	Port P4 (P4)
03E9₁₆	Port P5 (P5)
03EA₁₆	Port P4 direction register (PD4)
03EB₁₆	Port P5 direction register (PD5)
03EC₁₆	Port P6 (P6)
03ED₁₆	Port P7 (P7)
03EE₁₆	Port P6 direction register (PD6)
03EF₁₆	Port P7 direction register (PD7)
03F0₁₆	
03F1₁₆	
03F2₁₆	
03F3₁₆	
03F4₁₆	
03F5₁₆	
03F6₁₆	
03F7₁₆	
03F8₁₆	
03F9₁₆	
03FA₁₆	
03FB₁₆	
03FC₁₆	Pull-up control register 0 (PUR0)
03FD₁₆	Pull-up control register 1 (PUR1)
03FE₁₆	Port P1 drive control register (DRR)
03FF₁₆	

Note 1: This register is only exist in flash memory version.
Note 2: Locations in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

Figure 1.8. Location of peripheral unit control registers (2)

Central Processing Unit (CPU)

The CPU has a total of 13 registers shown in Figure 1.9. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these have two register banks.

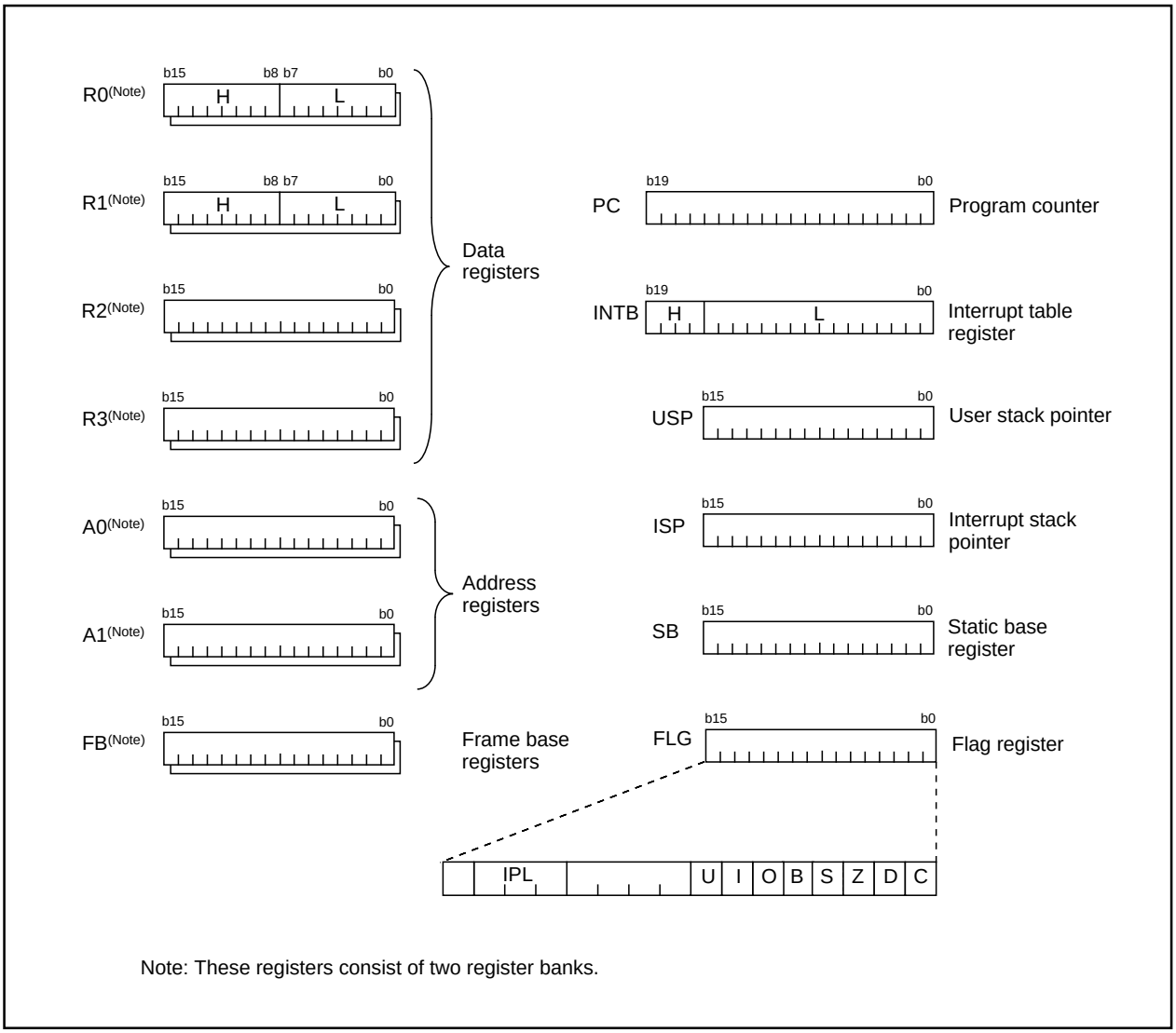


Figure 1.9. Central processing unit register

(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 1.10 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.

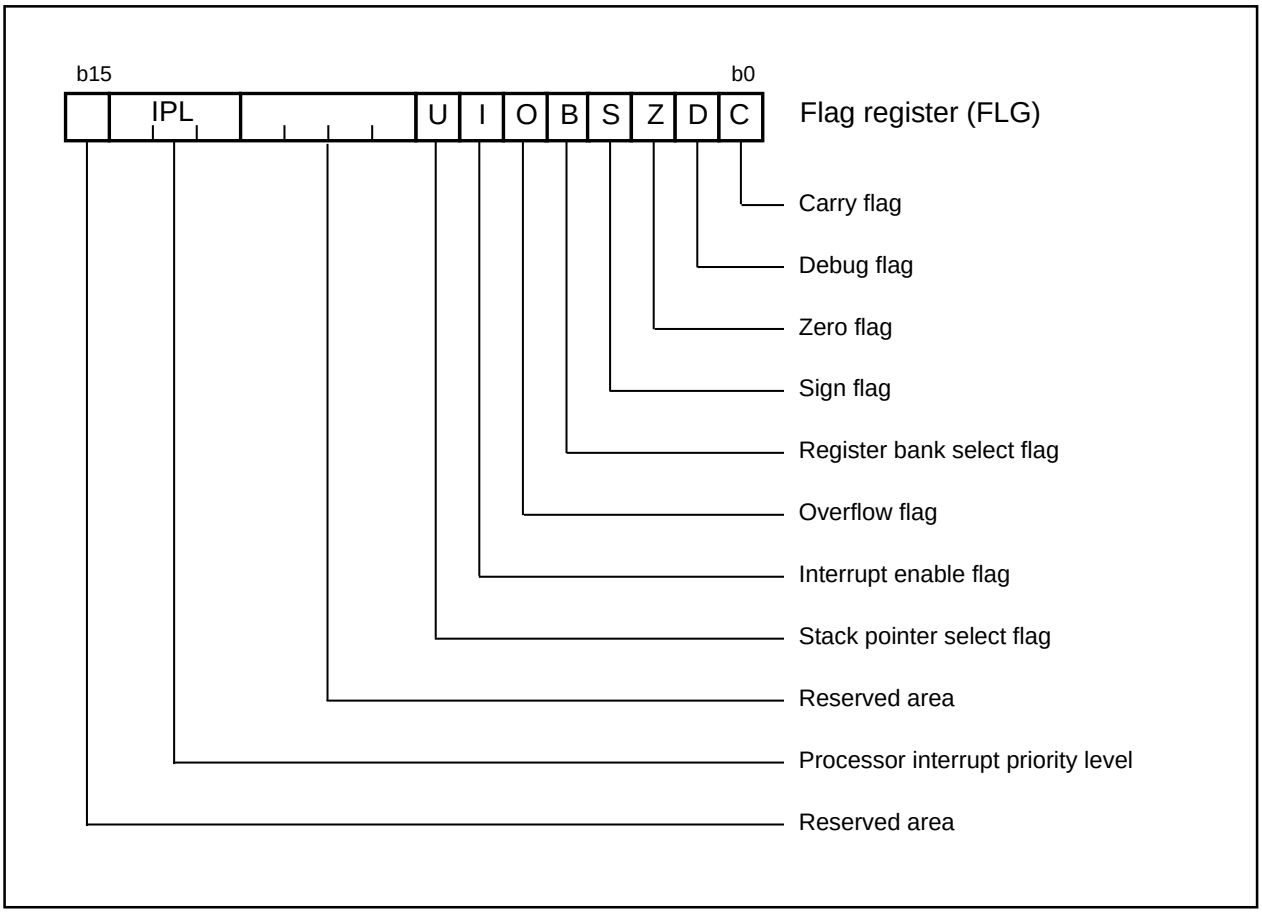


Figure 1.10. Flag register (FLG)

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 1.11 shows the example reset circuit. Figure 1.12 shows the reset sequence.

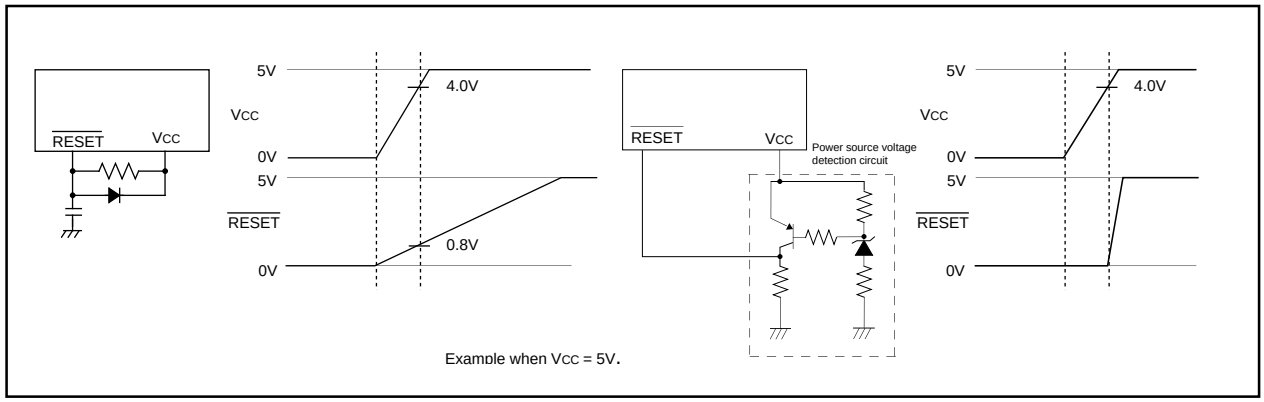


Figure 1.11. Example reset circuit

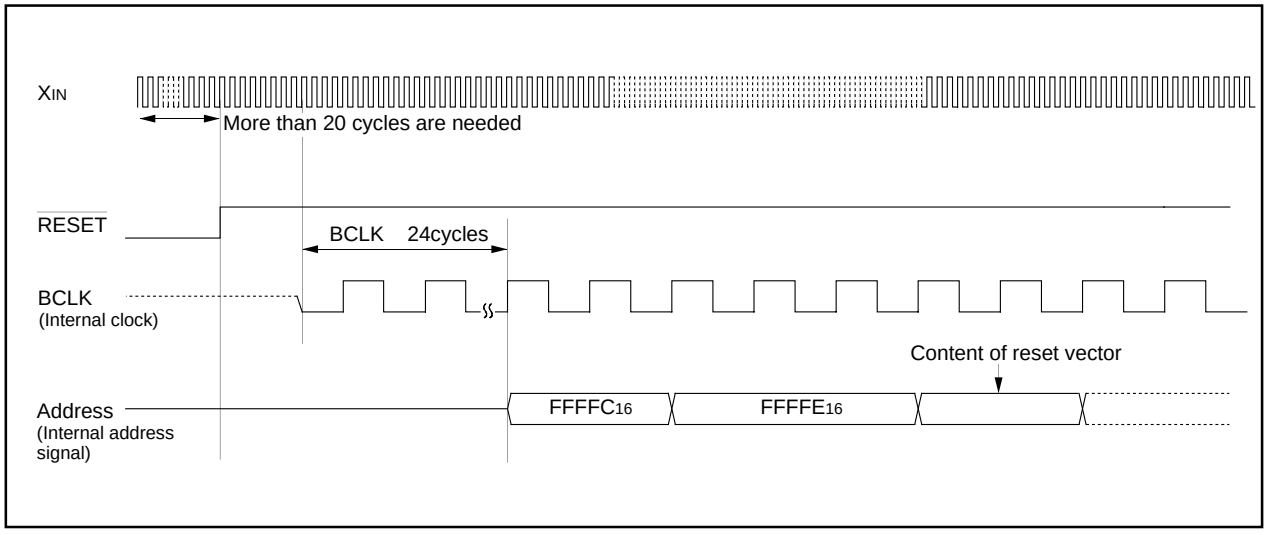
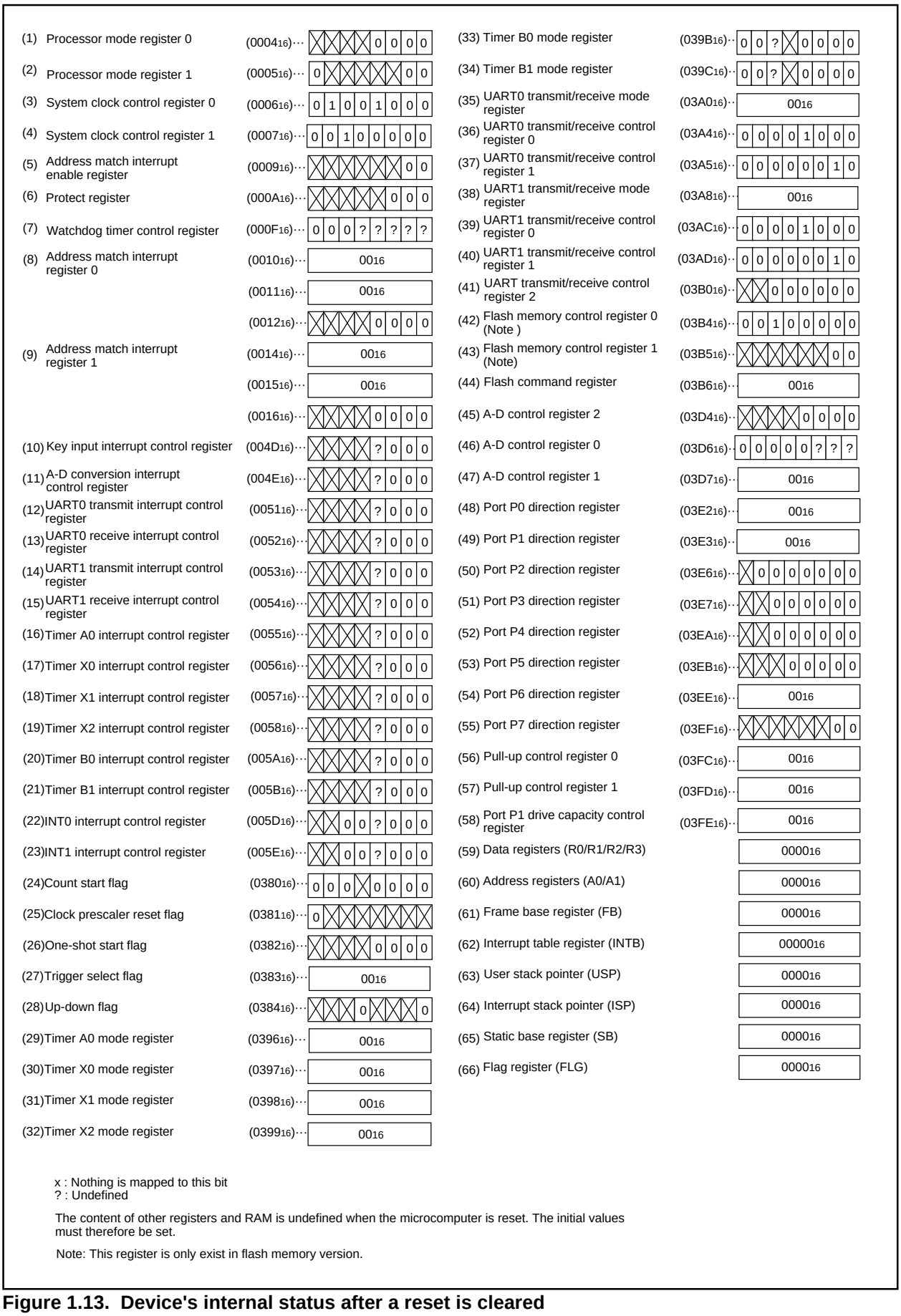


Figure 1.12. Reset sequence



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.

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

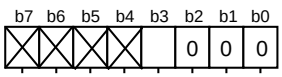
Figure 1.13. Device's internal status after a reset is cleared

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 1.14 shows the processor mode register 0 and 1.

Processor mode register 0 (Note)

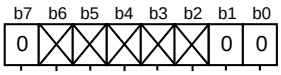


Symbol	Address	When reset
PM0	0004_{16}	$XXXX0000_2$

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. In an attempt to write to these bits, write “0”. The value, if read, turns out to be 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)



Symbol	Address	When reset
PM1	0005_{16}	$0XXXXXX0_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
Reserved bit		Must always be set to “0”	○	○

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

Figure 1.14. Processor mode register 0 and 1.

Clock Generating Circuit

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

Table 1.2. 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	XIN, XOUT	XCIN, XCOUT
Oscillation stop/restart function	Available	Available
Oscillator status immediately after reset	Oscillating	Stopped
Other	Externally derived clock can be input	

Example of oscillator circuit

Figure 1.15 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 1.16 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.

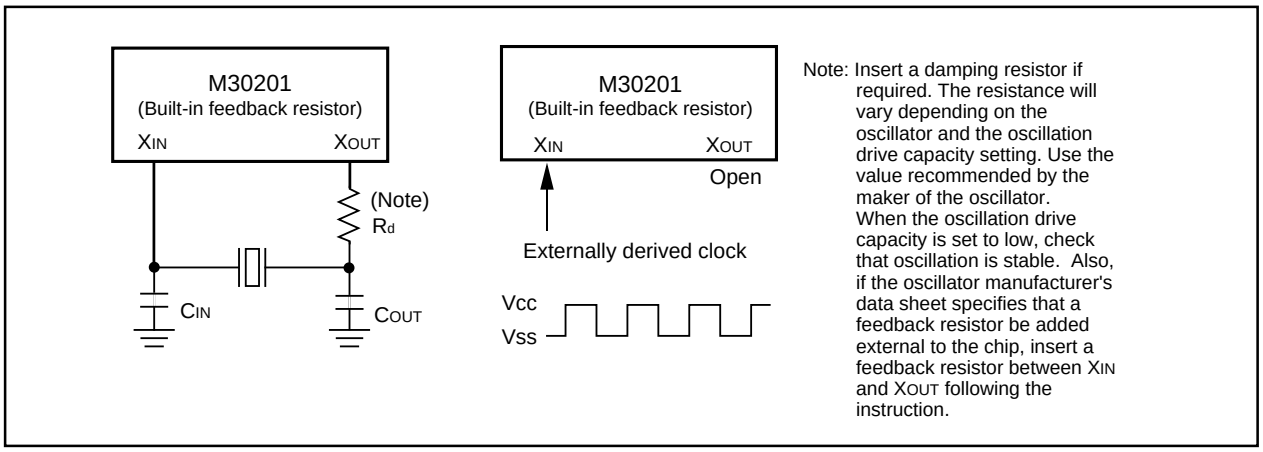


Figure 1.15. Examples of main clock

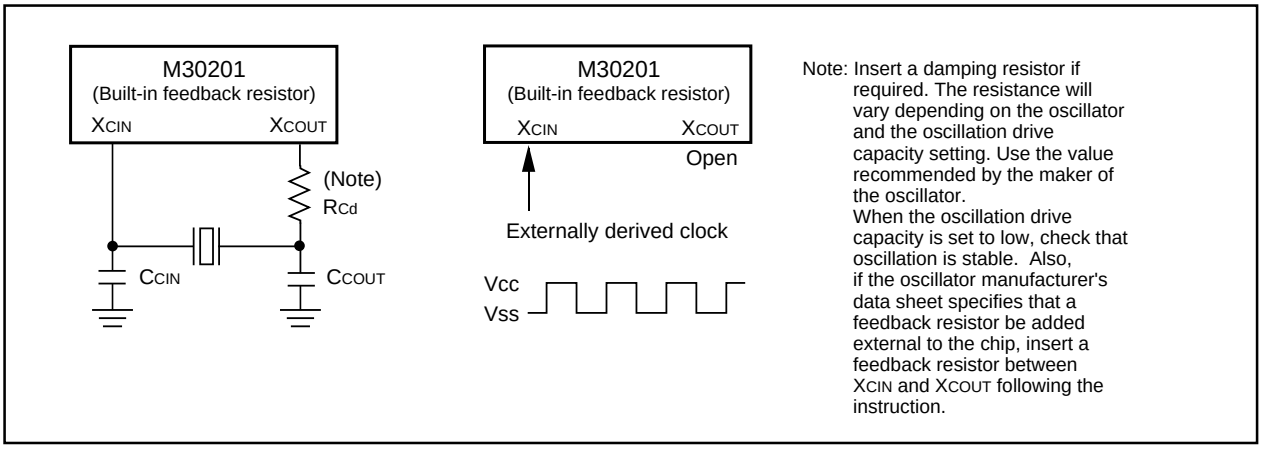


Figure 1.16. Examples of sub-clock

Clock Control

Figure 1.17 shows the block diagram of the clock generating circuit.

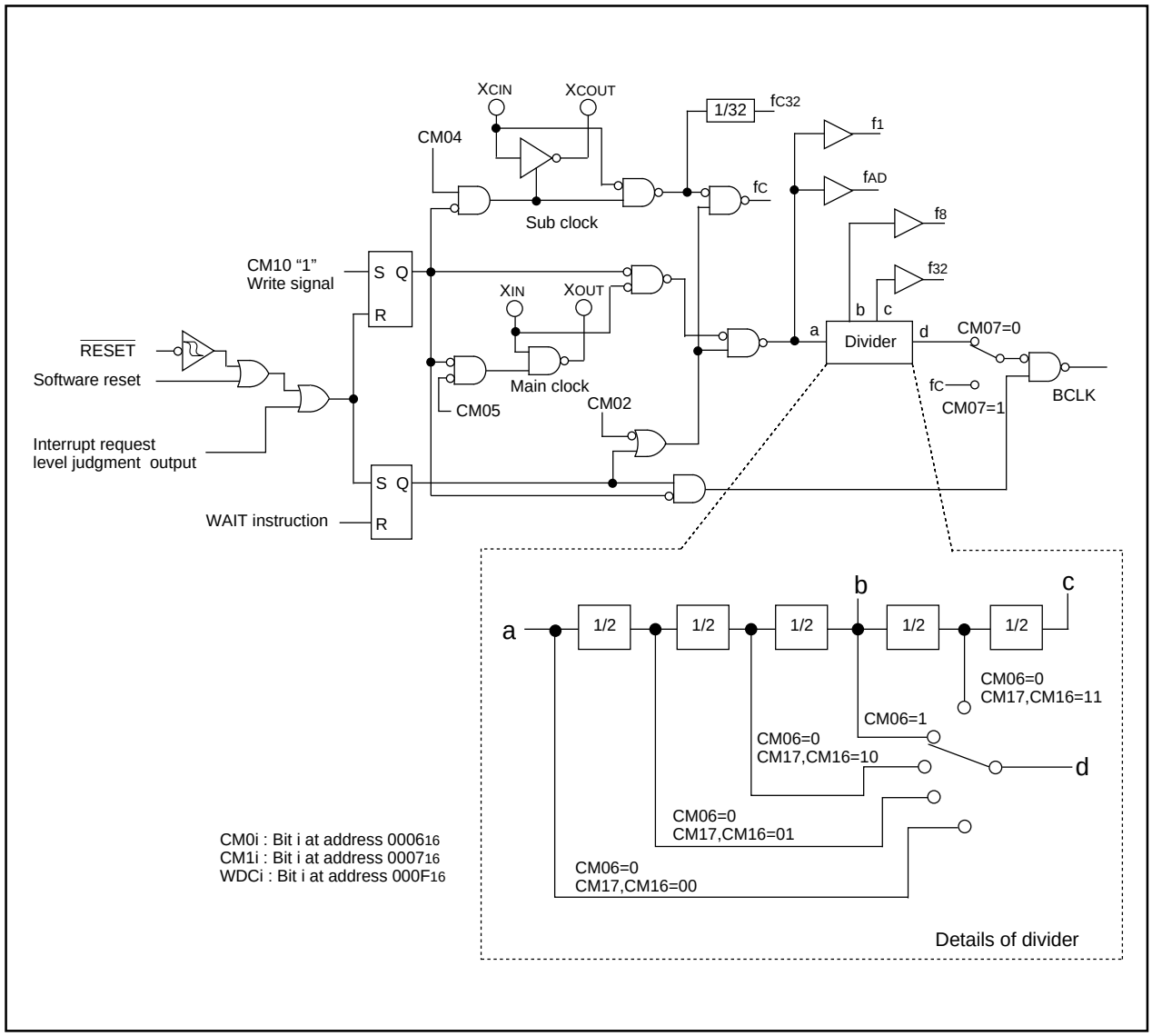


Figure 1.17. Clock generating circuit

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

(2) Sub-clock

The sub-clock is generated by the sub-clock oscillation circuit. No sub-clock is generated after a reset. After oscillation is started using the port Xc select bit (bit 4 at address 0006_{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 sub-clock oscillation circuit can be reduced using the XCIN-XCOUT drive capacity select bit (bit 3 at address 0006_{16}). Reducing the drive capacity of the sub-clock oscillation circuit reduces the power dissipation. This bit changes to "1" when shifting to stop mode and at a reset.

(3) BCLK

The BCLK is the clock that drives the CPU, and is fc or the clock is derived by dividing the main clock by 1, 2, 4, 8, or 16. The BCLK is derived by dividing the main clock by 8 after a reset.

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

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

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.

(5) f_{C32}

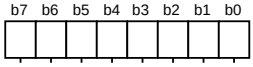
This clock is derived by dividing the sub-clock by 32. It is used for the timer A, timer B and timer X counts.

(6) f_C

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

Figure 1.18 shows the system clock control registers 0 and 1.

System clock control register 0 (Note 1)

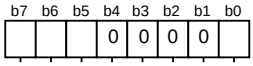


Symbol: CM0 — Address: 000616 — When reset: 4816

Bit symbol	Bit name	Function	R	W
CM00	Clock output function select bit	b1 b0 0 0 : I/O port P54 0 1 : fC output 1 0 : f8 output 1 1 : Clock divide counter output	O	O
CM01			O	O
CM02	WAIT peripheral function clock stop bit	0 : Do not stop peripheral function clock in wait mode 1 : Stop peripheral function clock in wait mode (Note 8)	O	O
CM03	XCIN-XCOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM04	Port XC select bit	0 : I/O port 1 : XCIN-XCOUT generation	O	O
CM05	Main clock (XIN-XOUT) stop bit (Note 3,4,5)	0 : On 1 : Off	O	O
CM06	Main clock division select bit 0 (Note 7)	0 : CM16 and CM17 valid 1 : Division by 8 mode	O	O
CM07	System clock select bit (Note 6)	0 : XIN, XOUT 1 : XCIN, XCOUT	O	O

Note 1: Set bit 0 of the protect register (address 000A16) to “1” before writing to this register.
Note 2: Changes to “1” when shifting to stop mode and at a reset.
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 XIN 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 inputting external clock, only clock oscillation buffer is stopped and clock input is acceptable.
Note 5: If this bit is set to “1”, XOUT turns “H”. The built-in feedback resistor remains being connected, so XIN turns pulled up to XOUT (“H”) via the feedback resistor.
Note 6: Set port Xc select bit (CM04) to “1” and stabilize the sub-clock oscillating before setting to this bit from “0” to “1”. Do not write to both bits at the same time. And also, set the main clock stop bit (CM05) to “0” and stabilize the main clock oscillating before setting this bit from “1” to “0”.
Note 7: This bit changes to “1” when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.
Note 8: fC32 is not included. Do not set to “1” when using low-speed or low power dissipation mode.

System clock control register 1 (Note 1)



Symbol: CM1 — Address: 000716 — When reset: 2016

Bit symbol	Bit name	Function	R	W
CM10	All clock stop control bit (Note 4)	0 : Clock on 1 : All clocks off (stop mode)	O	O
Reserved bit		Always set to “0”	O	O
Reserved bit		Always set to “0”	O	O
Reserved bit		Always set to “0”	O	O
Reserved bit		Always set to “0”	O	O
CM15	XIN-XOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
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	O	O
CM17				

Note 1: Set bit 0 of the protect register (address 000A16) to “1” before writing to this register.
Note 2: This bit changes to “1” when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.
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: If this bit is set to “1”, XOUT turns “H”, and the built-in feedback resistor is cut off. XCIN and XCOUT turn high-impedance state.

Figure 1.18. 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 1.19 shows a block diagram of clock output.

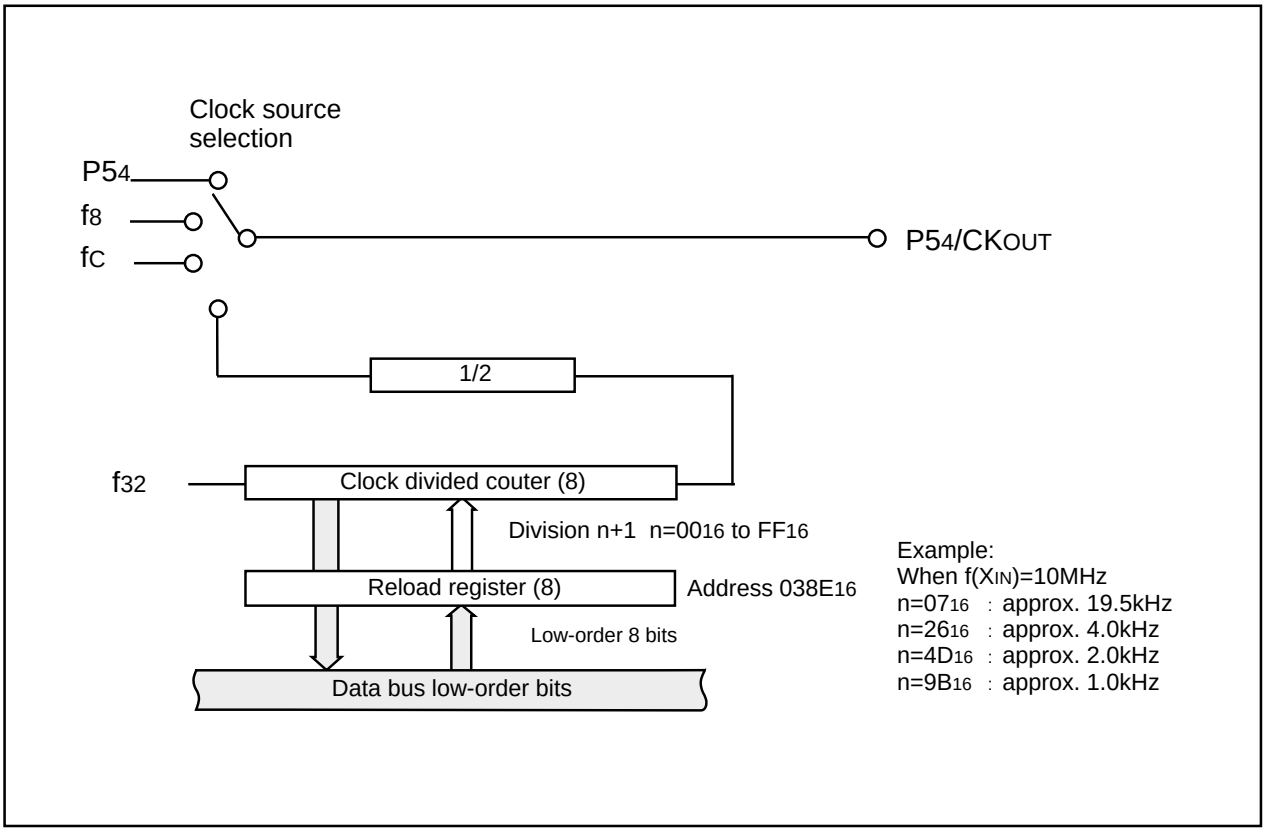


Figure 1.19. Block diagram of clock output

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 1.3 shows the status of the ports in stop mode.

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

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

Table 1.3. Port status during stop mode

Pin		States
Port		Retains status before stop mode
CLK_{OUT}	When fC 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. However, peripheral function clock f_{C32} does not stop so that the peripherals using f_{C32} do not contribute to the power saving. When the MCU running in low-speed or low power dissipation mode, do not enter WAIT mode with this bit set to “1”. Table 1.4 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 from the interrupt routine using as BCLK, the clock that had been selected when the WAIT instruction was executed.

Table 1.4. Port status during wait mode

Pin		States
Port		Retains status before wait mode
CLK_{OUT}	When fC 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 1.5 shows the operating modes corresponding to the settings of system clock control registers 0 and 1.

When reset, the device starts in division by 8 mode. The main clock division select bit 0(bit 6 at address 0006_{16}) changes to "1" when shifting from high-speed/medium-speed to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained. The following shows the operational modes of BCLK.

(1) Division by 2 mode

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

(2) Division by 4 mode

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

(3) Division by 8 mode

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

(4) Division by 16 mode

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

(5) No-division mode

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

(6) Low-speed mode

f_C 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

f_C is the BCLK and the main clock is stopped.

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

Table 1.5. 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

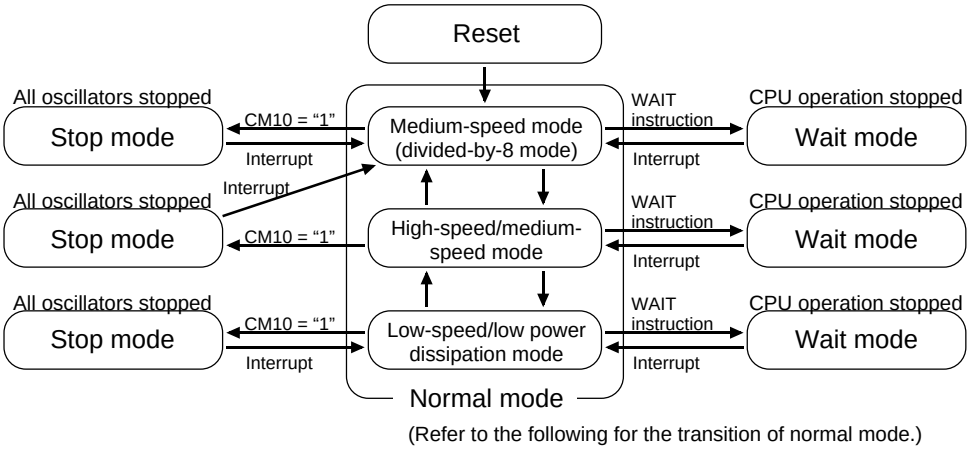
CPU operation is halted in this mode. The oscillator continues to run.

(3) Stop mode

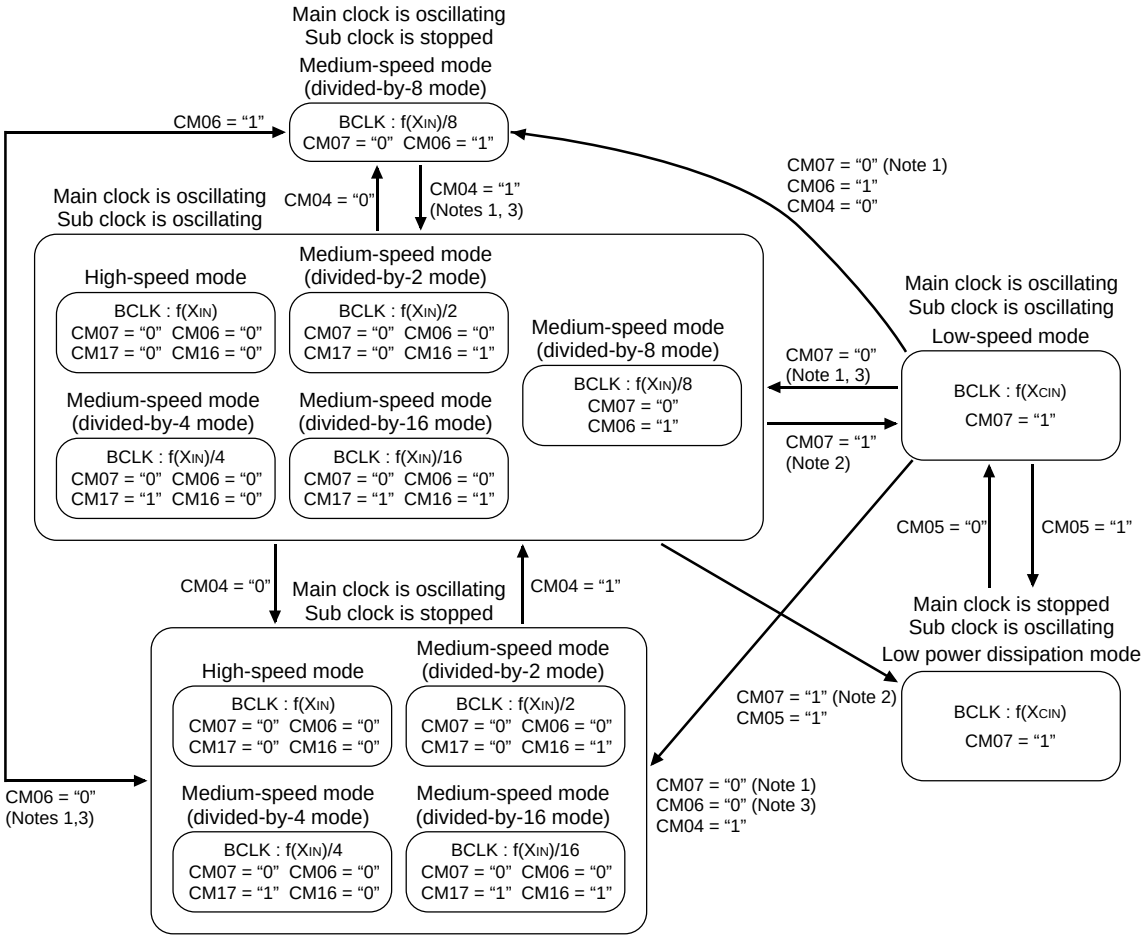
All oscillators stop in this mode. The CPU and internal peripheral functions all stop. Of all 3 power saving modes, power savings are greatest in this mode.

Figure 1.20 shows the transition between each of the three modes, (1), (2), and (3).

Transition of stop mode, wait mode



Transition of normal mode



Note 1: Switch clock after oscillation of main clock is sufficiently stable.
Note 2: Switch clock after oscillation of sub clock is sufficiently stable.
Note 3: Change CM06 after changing CM17 and CM16.
Note 4: Transit in accordance with arrow.

Figure 1.20. 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 1.21 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

b7	b6	b5	b4	b3	b2	b1	b0
✕	✕	✕	✕	✕			

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

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. In an attempt to write to these bits, write “0”. The value, if read, turns out to be 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 1.21. Protect register

Overview of Interrupt

Type of Interrupts

Figure 1.22 lists the types of interrupts.

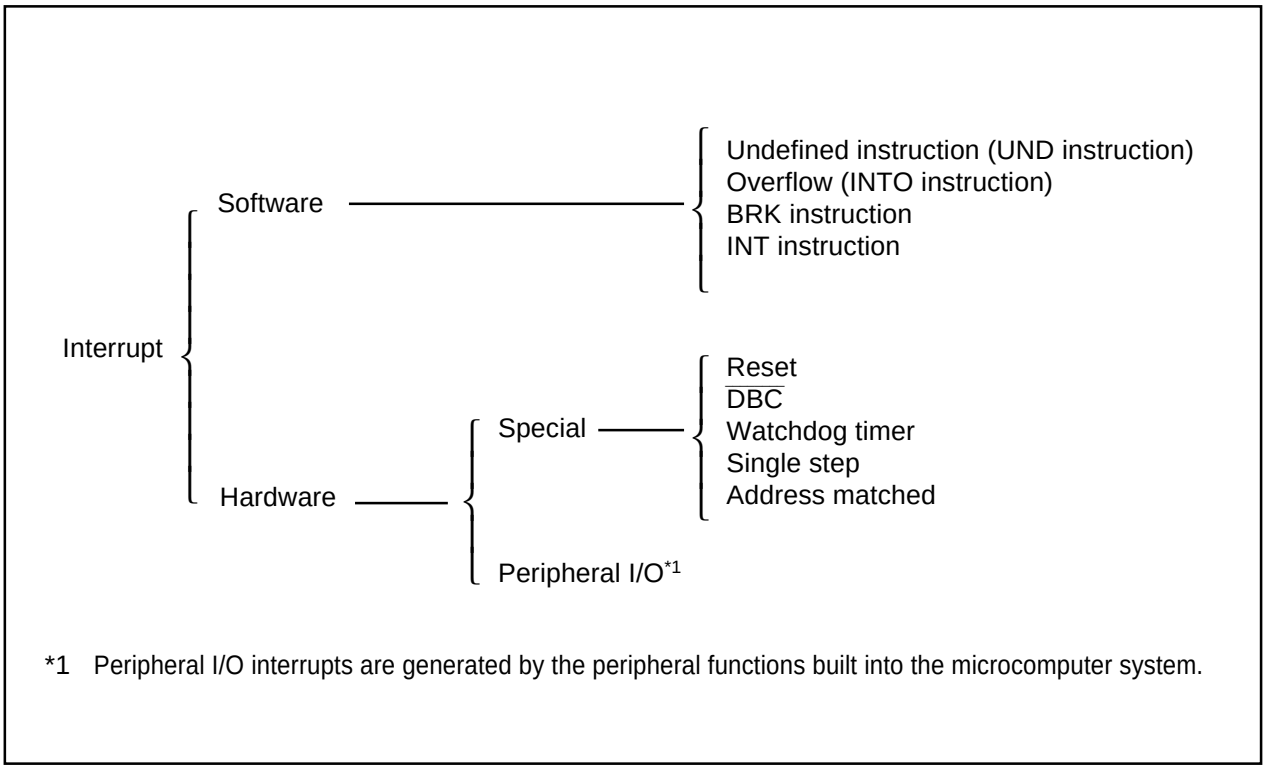


Figure 1.22. Classification of interrupts

- 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 $\overline{\text{KI}}$ 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.
- **$\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ interrupt**
  An $\overline{\text{INT}}$ interrupt occurs if either a rising edge or a falling edge is input to the $\overline{\text{INT}}$ pin.

Interrupts and Interrupt Vector Tables

If an interrupt request is accepted, a program branches to the interrupt routine set in the interrupt vector table. Set the first address of the interrupt routine in each vector table. Figure 1.23 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.

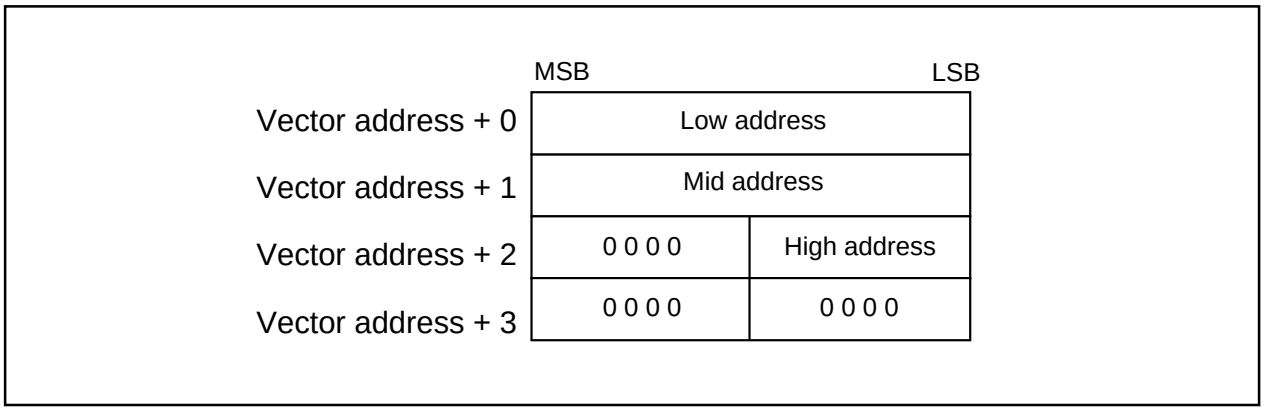


Figure 1.23. Format for specifying interrupt vector addresses

• 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 FFFDC16 to FFFFF16. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table 1.6 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

Table 1.6. Interrupt and fixed vector address

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFDC16 to FFFDF16	Interrupt on UND instruction
Overflow	FFFE016 to FFFE316	Interrupt on INTO instruction
BRK instruction	FFFE416 to FFFE716	If the vector is filled with FF16, program execution starts from the address shown by the vector in the variable vector table
Address match	FFFE816 to FFFEB16	There is an address-matching interrupt enable bit
Single step (Note)	FFFEC16 to FFFEF16	Do not use
Watchdog timer	FFFF016 to FFFF316	
$\overline{\text{DBC}}$ (Note)	FFFF416 to FFFF716	Do not use
-	FFFF816 to FFFFB16	-
Reset	FFFFC16 to FFFFF16	

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

Table 1.7. 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 1.24 shows the interrupt control registers.

Interrupt control register (Note 2)

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)	$005A_{16}$, $005B_{16}$	$XXXXX000_2$

b7 b6 b5 b4 b3 b2 b1 b0

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 1)
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			–	–

Note 1: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).
Note 2: To rewrite the interrupt control register, do so at a point that dose not generate the interrupt request for that register. For details, see the precautions for interrupts.

b7 b6 b5 b4 b3 b2 b1 b0
0

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 1)
POL	Polarity select bit	0 : Selects falling edge 1 : Selects rising edge	○	○
Reserved bit		Always set to “0”	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			–	–

Note 1: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).
Note 2: To rewrite the interrupt control register, do so at a point that dose not generate the interrupt request for that register. For details, see the precautions for interrupts.

Figure 1.24. 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 1.8 shows the settings of interrupt priority levels and Table 1.9 shows the interrupt levels enabled, according to the contents 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 1.8. Settings of interrupt priority levels

Interrupt priority level select bit (b2 b1 b0)	Interrupt priority level	Priority order
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 1.9. Interrupt levels enabled according to the contents of the IPL

IPL (IPL_2 IPL_1 IPL_0)	Enabled interrupt priority levels
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:

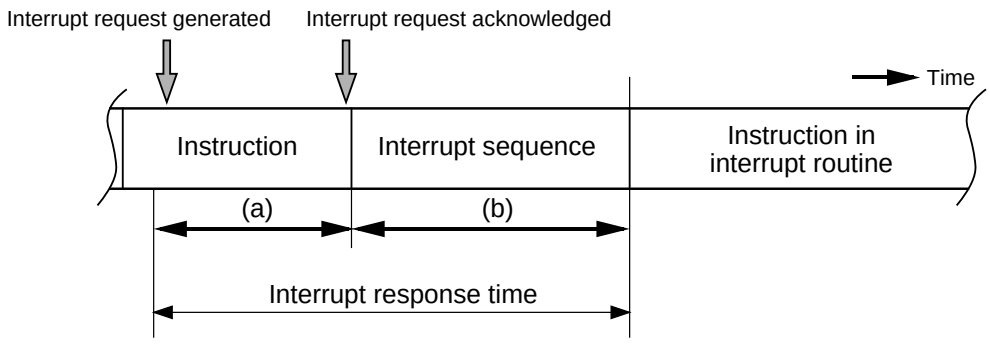
(1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 0000_{16}. After this, the corresponding interrupt request bit becomes "0".
(2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the 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 1.25 shows the 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 1.25. 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 1.10.

Table 1.10. 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 $\overline{DBC}$ 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.

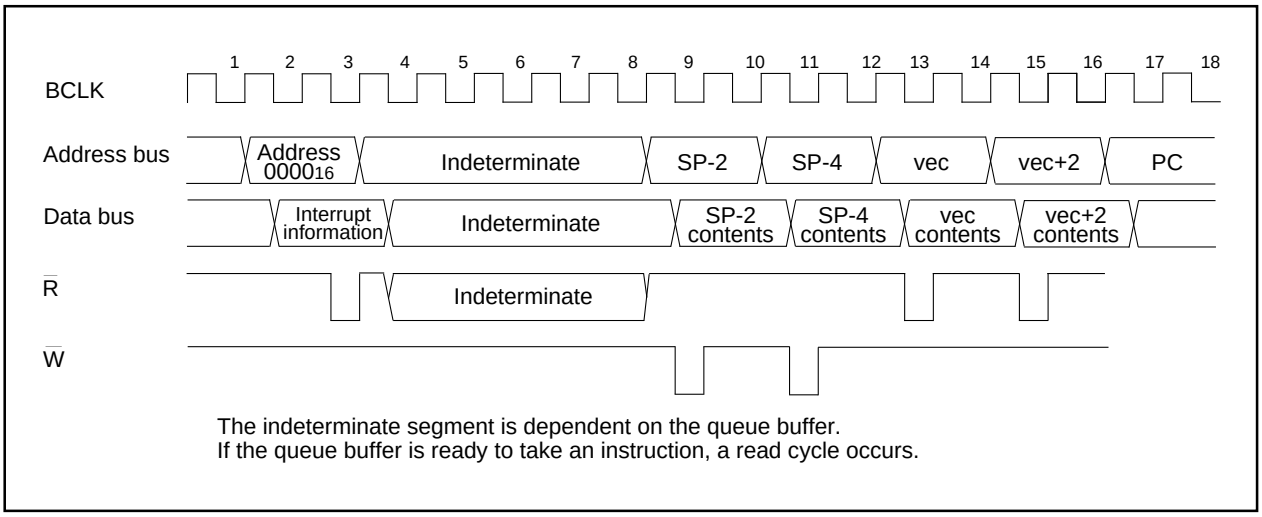


Figure 1.26. Time required for executing the interrupt sequence

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 1.11 is set in the IPL.

Table 1.11. 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 1.27 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).

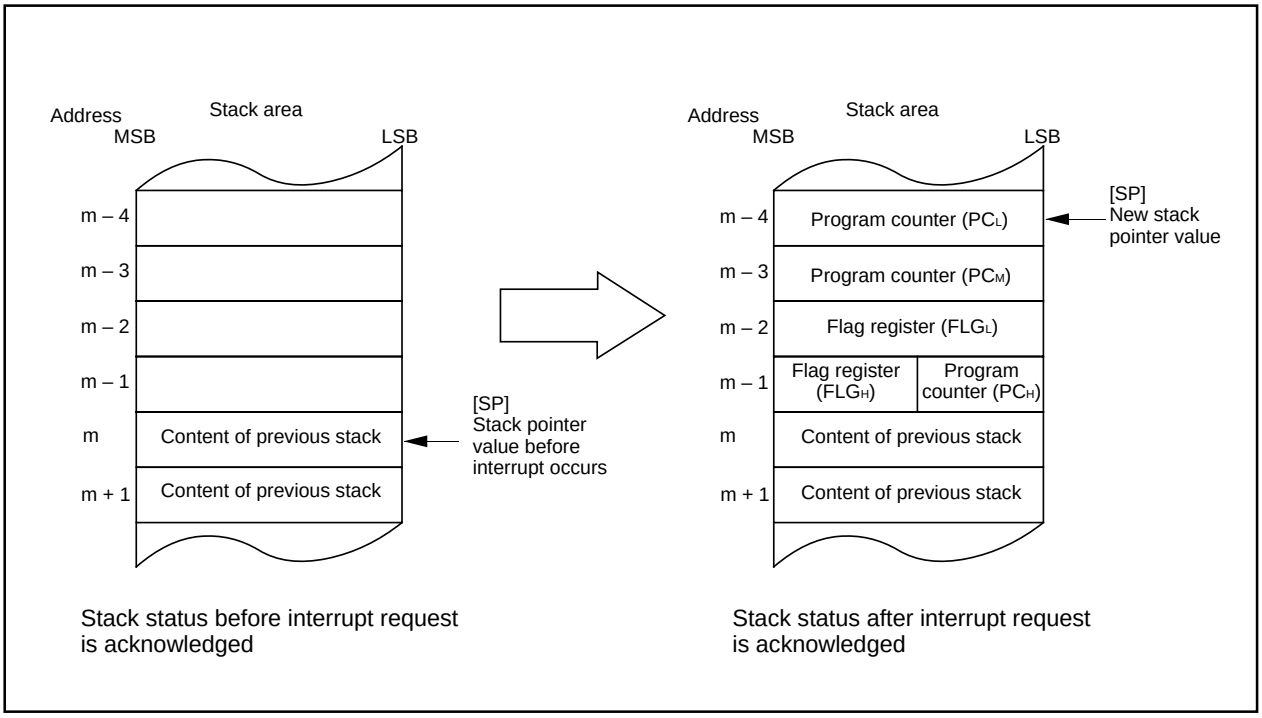


Figure 1.27. State of stack before and after acceptance of interrupt request

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 1.28 shows the operation of the saving registers.

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

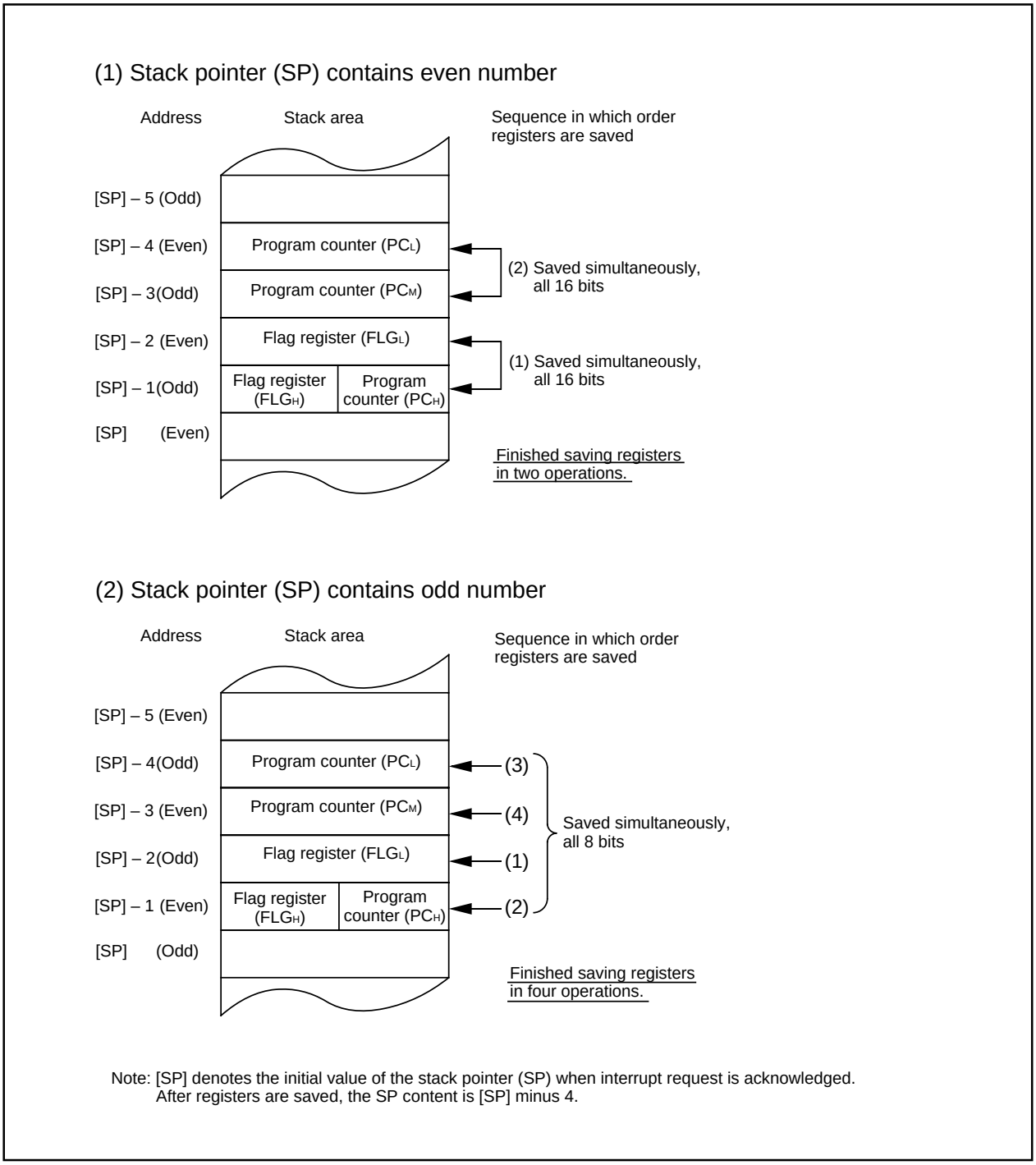


Figure 1.28. Operation of saving registers

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 1.29 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 1.30 shows the interrupt resolution circuit.

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

Figure 1.29. Hardware interrupts priorities

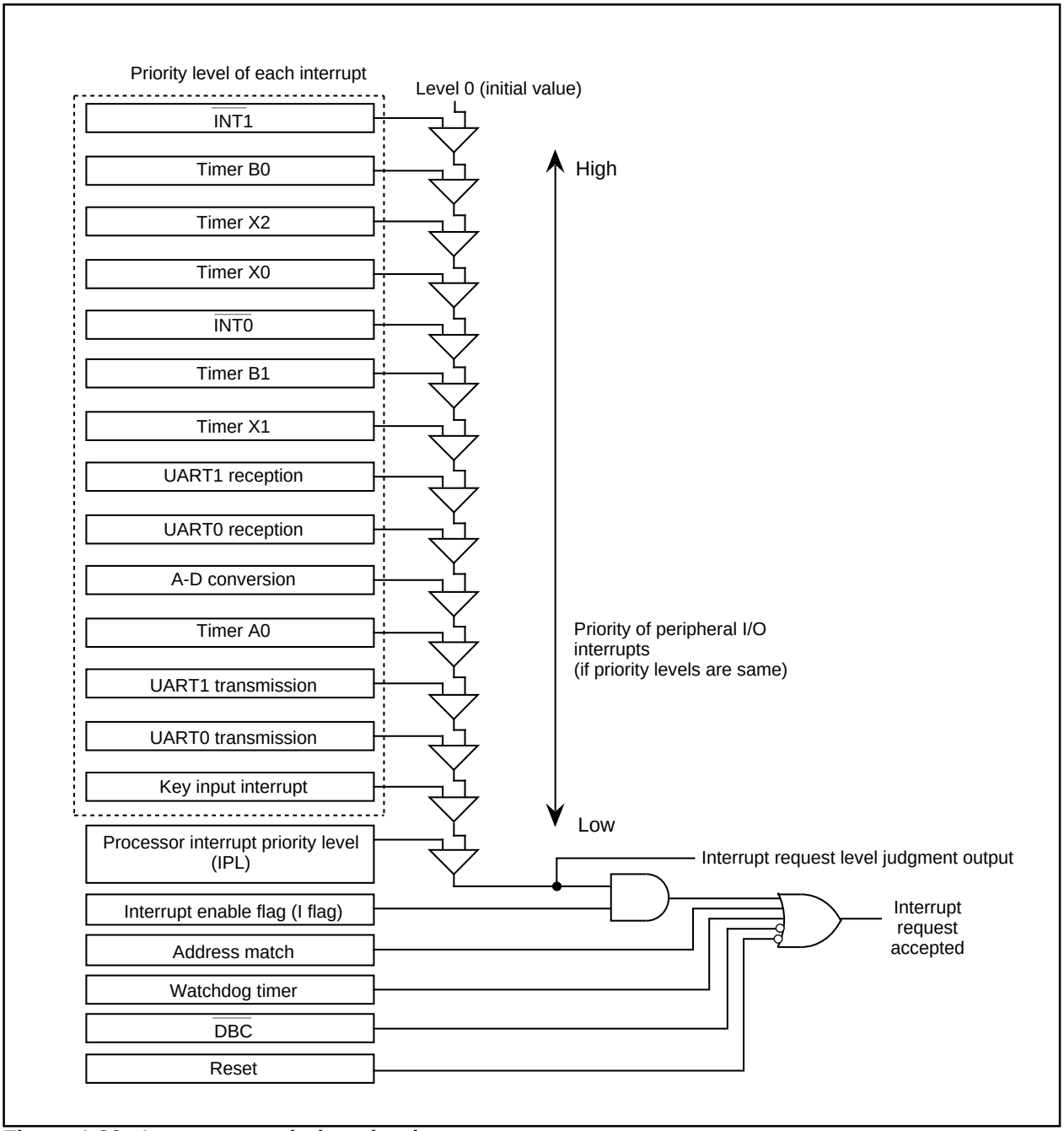


Figure 1.30. Interrupt resolution circuit

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 1.31 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.

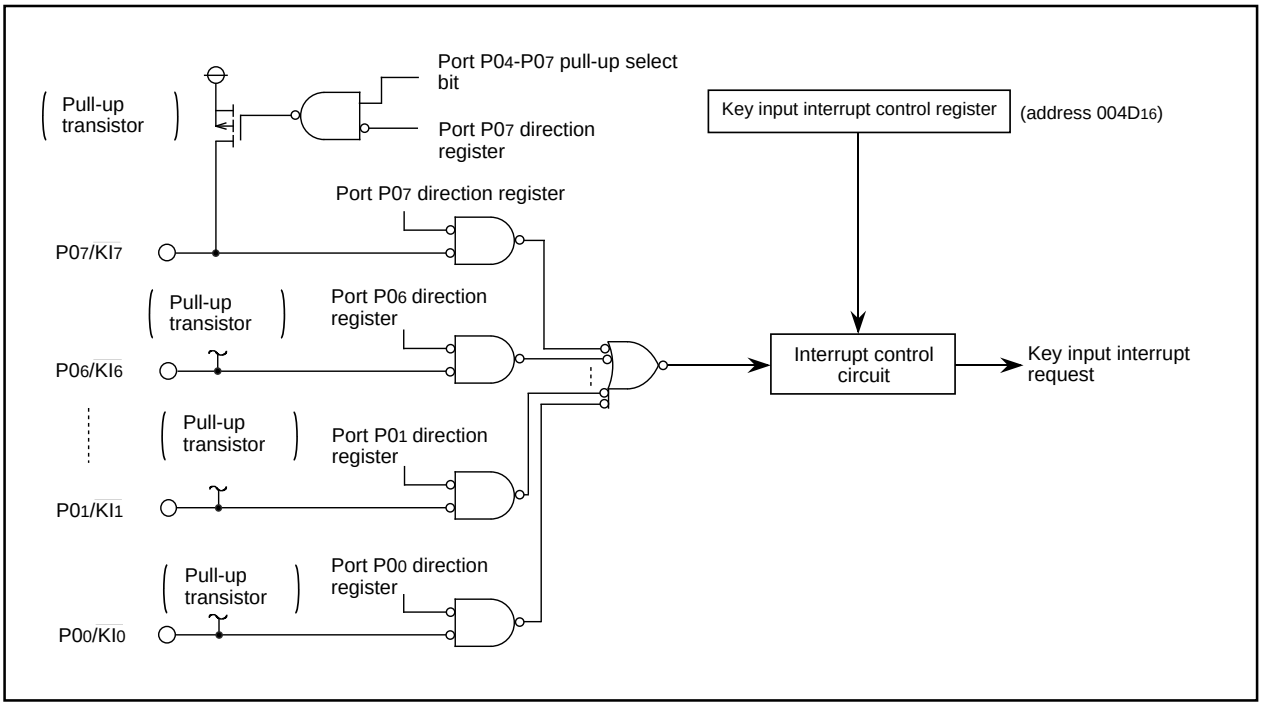


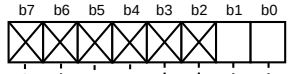
Figure 1.31. Block diagram of key input interrupt

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). For an address match interrupt, the value of the program counter (PC) that is saved to the stack area varies depending on the instruction being executed.

Figure 1.32 shows the address match interrupt-related registers.

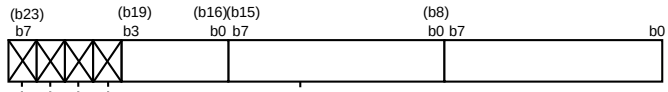
Address match interrupt enable register



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. In an attempt to write to these bits, write “0”. The value, if read, turns out to be 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. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.		—	—

Figure 1.32. 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 0000$_{16}$ 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 0000$_{16}$ 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 1.33 shows the switching condition of $\overline{\text{INT}}$ interrupt request.

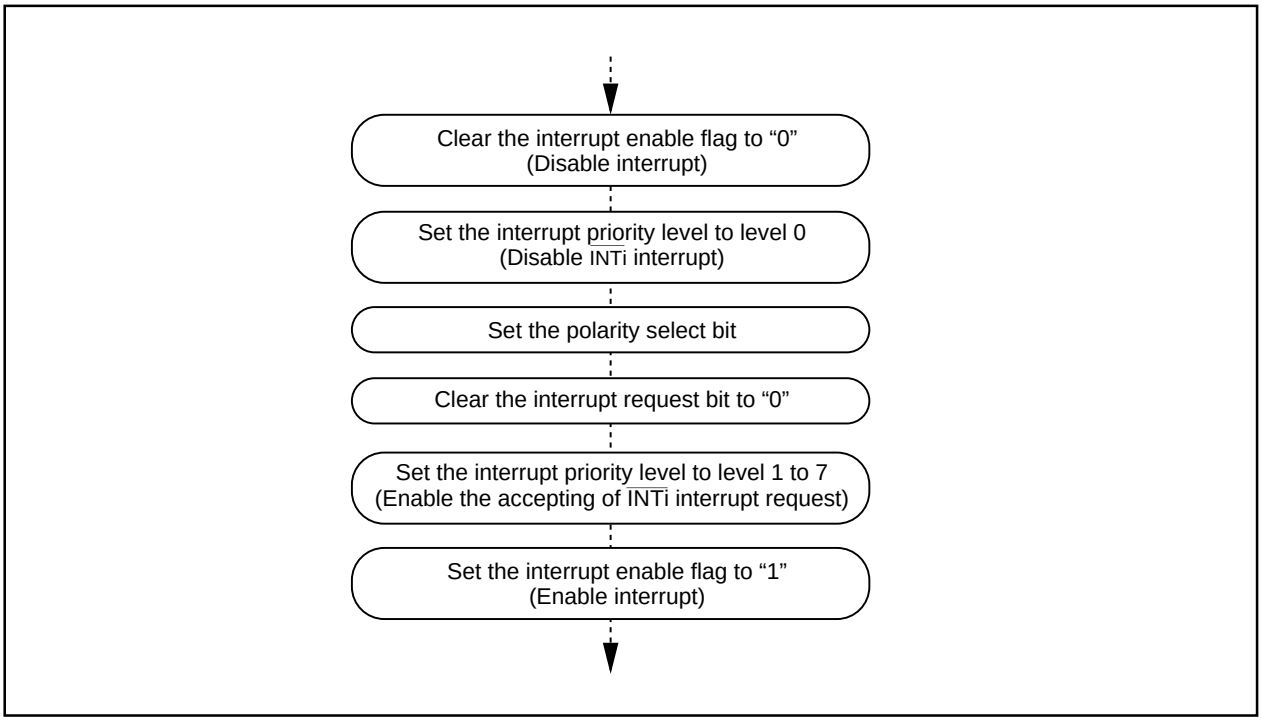


Figure 1.33. Switching condition of $\overline{\text{INT}}$ interrupt request

(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 XIN is selected for the BCLK, bit 7 of the watchdog timer control register (address 000F16) selects the prescaler division ratio (by 16 or by 128). When XCIN is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address 000F16).

When XIN is selected in BCLK

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

When XCIN 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 000E16) and when a watchdog timer interrupt request is generated. The prescaler is initialized only when the microcomputer is reset. After a reset is cancelled, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address 000E16). In stop mode and wait mode the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes are released. Figure 1.34 shows the block diagram of the watchdog timer. Figure 1.35 shows the watchdog timer-related registers.

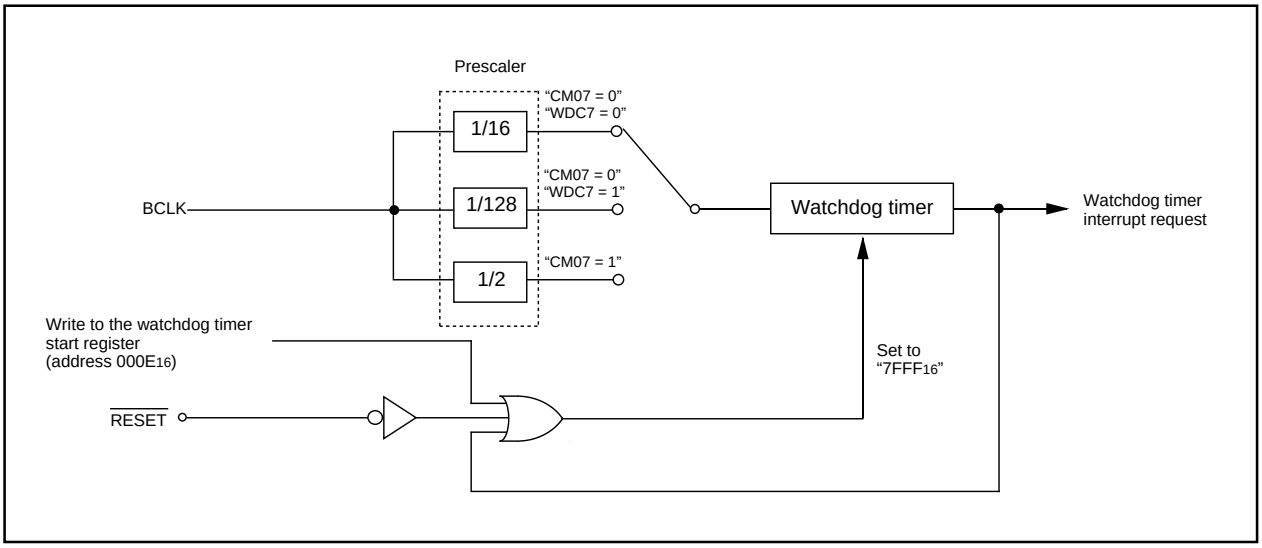


Figure 1.34. Block diagram of watchdog timer

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 1.35. 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. Figure 1.36 show the block diagram of timers.

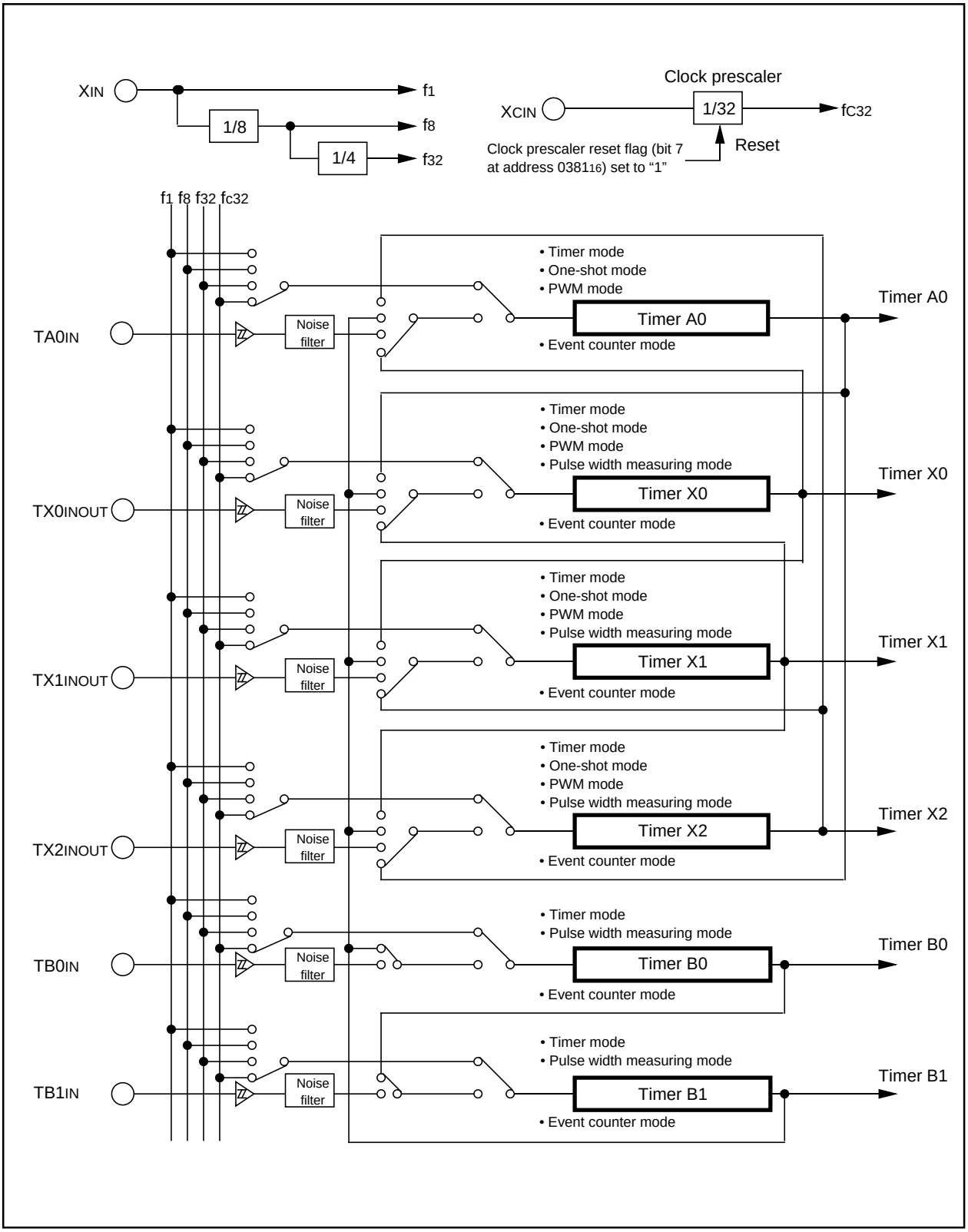


Figure 1.36. Timer block diagram

Timer A

Figure 1.37 shows the block diagram of timer A. Figures 1.38 to 1.40 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 "0000$_{16}$".
- Pulse width modulation (PWM) mode: The timer outputs pulses of a given width.

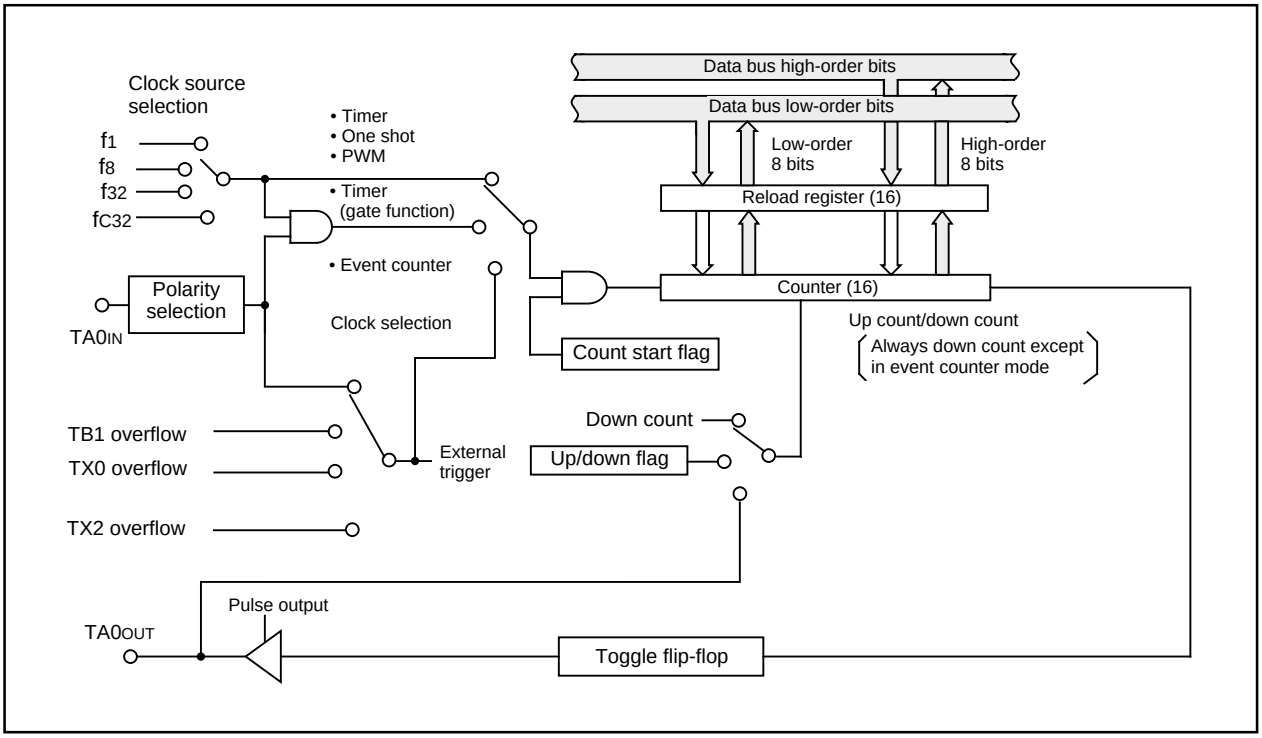


Figure 1.37. Block diagram of timer A

Timer A0 mode register

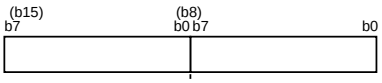
Symbol	Address	When reset
TA0MR	0396$_{16}$	00$_{16}$

b7	b6	b5	b4	b3	b2	b1	b0

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 1.38. Timer A-related registers (1)

Timer A0 register (Note 1)

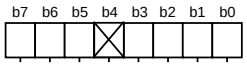


Symbol	Address	When reset
TA0	0387_{16},0386_{16}	Indeterminate

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 $FFFF_{16}$ (Note 2)	×	○
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000_{16} to $FFFE_{16}$ (Note 2)	×	○
• 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}(Note 2) (Both high-order and low-order addresses)	×	○

Note 1: Read and write data in 16-bit units.
Note 2: Use MOV instruction to write to this register.

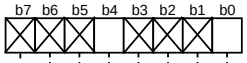
Count start flag



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. In an attempt to write to this bit, write “0”. The value, if read, turns out to be 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 (Note)



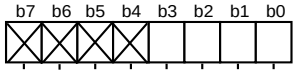
Symbol	Address	When reset
UDF	0384_{16}	$XXX0XXX0_2$

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. In an attempt to write to these bits, write “0”. The value, if read, turns out to be 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. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—

Note : Use MOV instruction to write to this register.

Figure 1.39. Timer A-related registers (2)

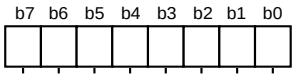
One-shot start flag



Symbol: ONSF | Address: 0382_{16} | When reset: $XXXX0000_2$

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. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—

Trigger select register

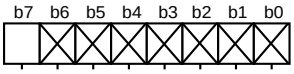


Symbol: TRGSR | Address: 0383_{16} | When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
TA0TGL	Timer A0 event/trigger select bit	b1 b0 0 0 : Input on TA0IN 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 TX0INOUT 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 TX1INOUT 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 TX2INOUT 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



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

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

Figure 1.40. Timer A-related registers (3)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.12.) Figure 1.41 shows the timer A0 mode register in timer mode.

Table 1.12. Specifications of timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, fc_{32}
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

Timer A0 mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol: TA0MR
Address: 0396_{16}
When reset: 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 "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 : fc_{32}	○	○
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 1.41. 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 1.13 lists timer specifications when counting a single-phase external signal. Figure 1.42 shows the timer A0 mode register in event counter mode. Table 1.14 lists timer specifications when counting a two-phase external signal. Figure 1.43 shows the timer A0 mode register in event counter mode.

Table 1.13. 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)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

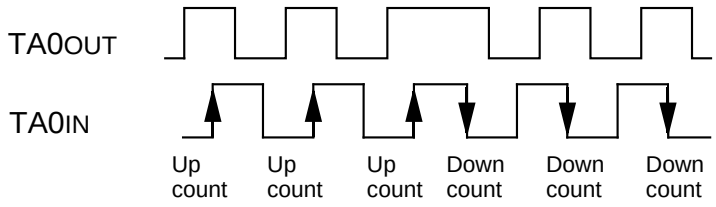
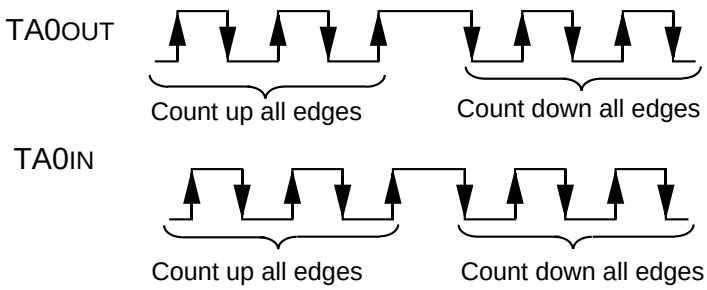
Symbol: TA0MR — Address: 0396_{16} — When reset: 00_{16}

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 "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 1.42. Timer A0 mode register in event counter mode

Table 1.14. 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"  • 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. 

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

Timer A0 mode register
(When using two-phase pulse signal processing)

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

Symbol: TA0MR
Address: 0396_{16}
When reset: 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 1.43. Timer A0 mode register in event counter mode

(3) One-shot timer mode

In this mode, the timer operates only once. (See Table 1.15.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 1.44 shows the timer A0 mode register in one-shot timer mode.

Table 1.15. 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)

Timer A0 mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

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 1.44. 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 1.16.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 1.45 shows the timer A0 mode register in pulse width modulation mode. Figure 1.46 shows the example of how a 16-bit pulse width modulator operates. Figure 1.47 shows the example of how an 8-bit pulse width modulator operates.

Table 1.16. 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.

Timer A0 mode register

b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

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 1.45. Timer A0 mode register in pulse width modulation mode

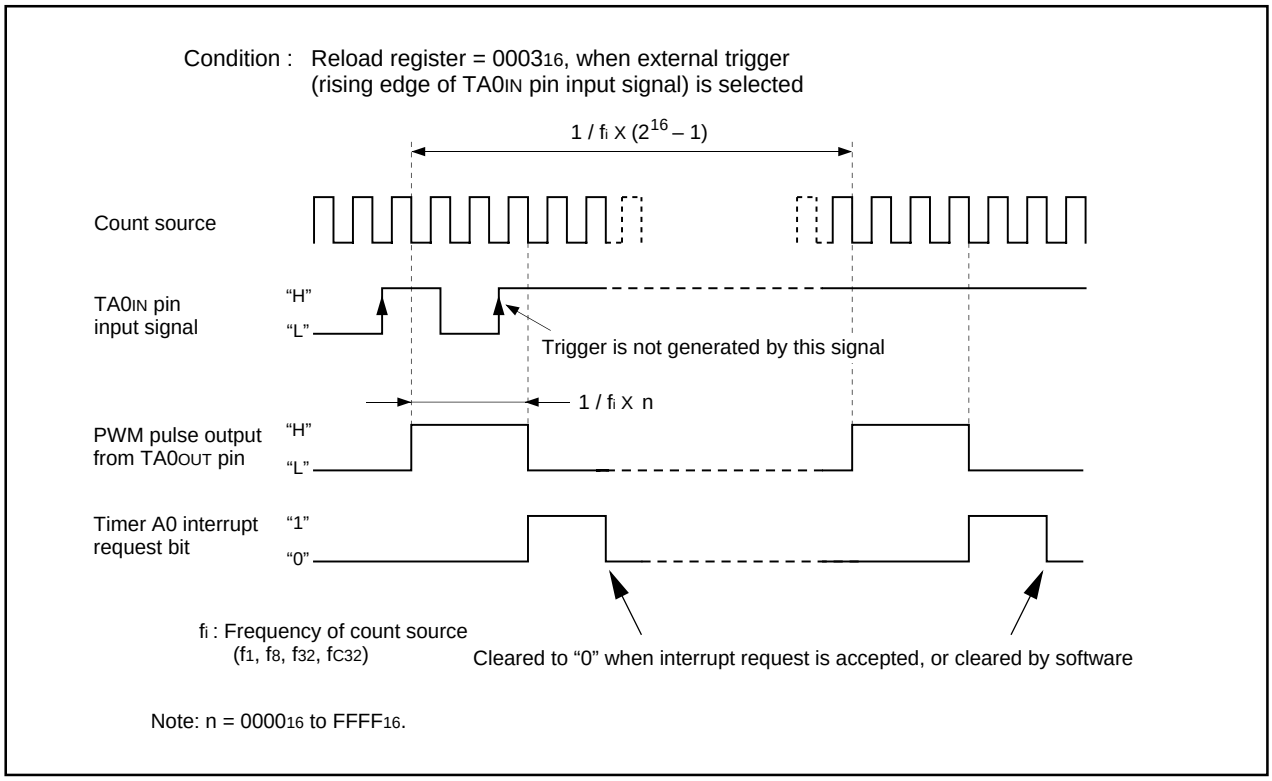


Figure 1.46. Example of how a 16-bit pulse width modulator operates

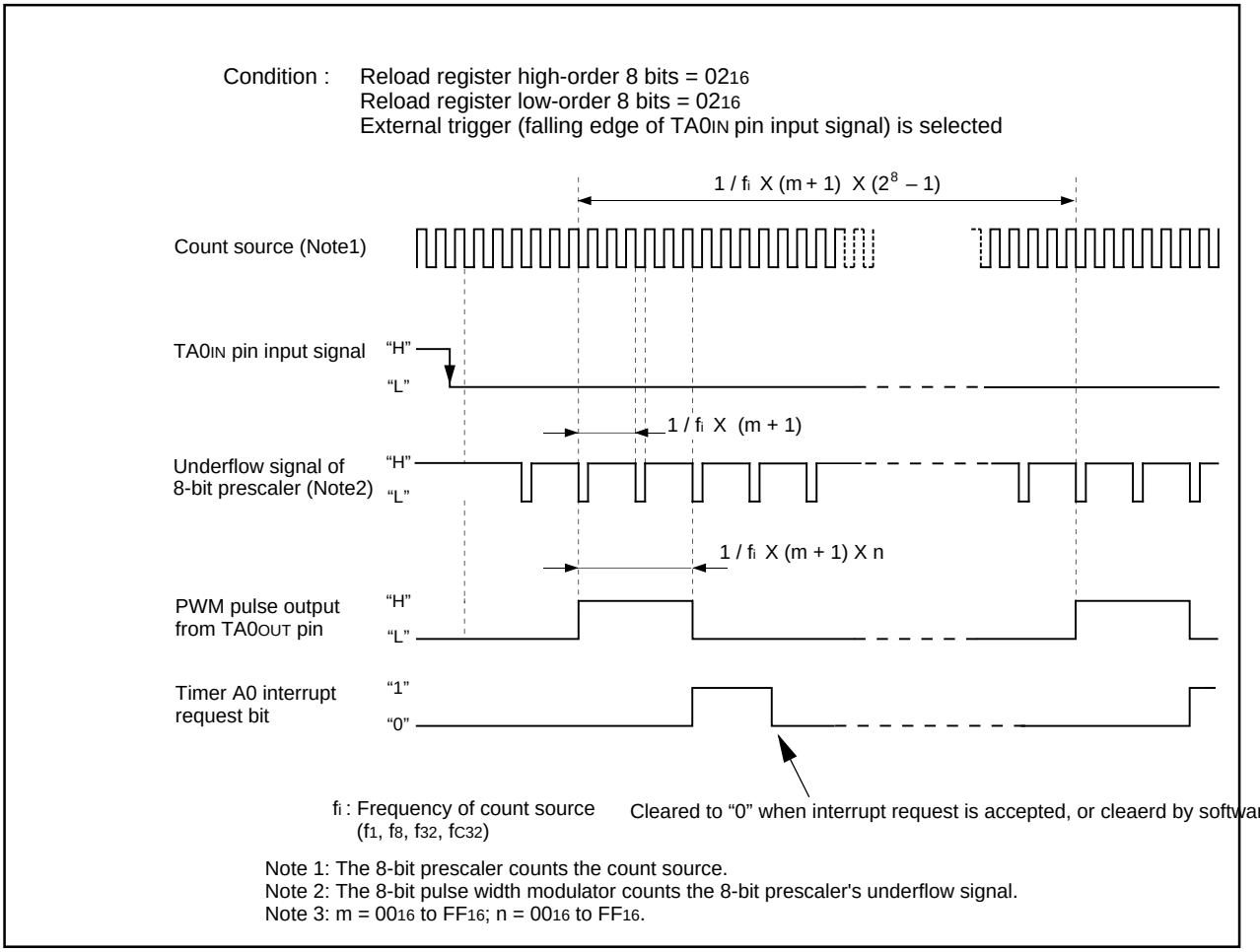


Figure 1.47. Example of how an 8-bit pulse width modulator operates

Timer B

Figure 1.48 shows the block diagram of timer B. Figures 1.49 and 1.50 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.

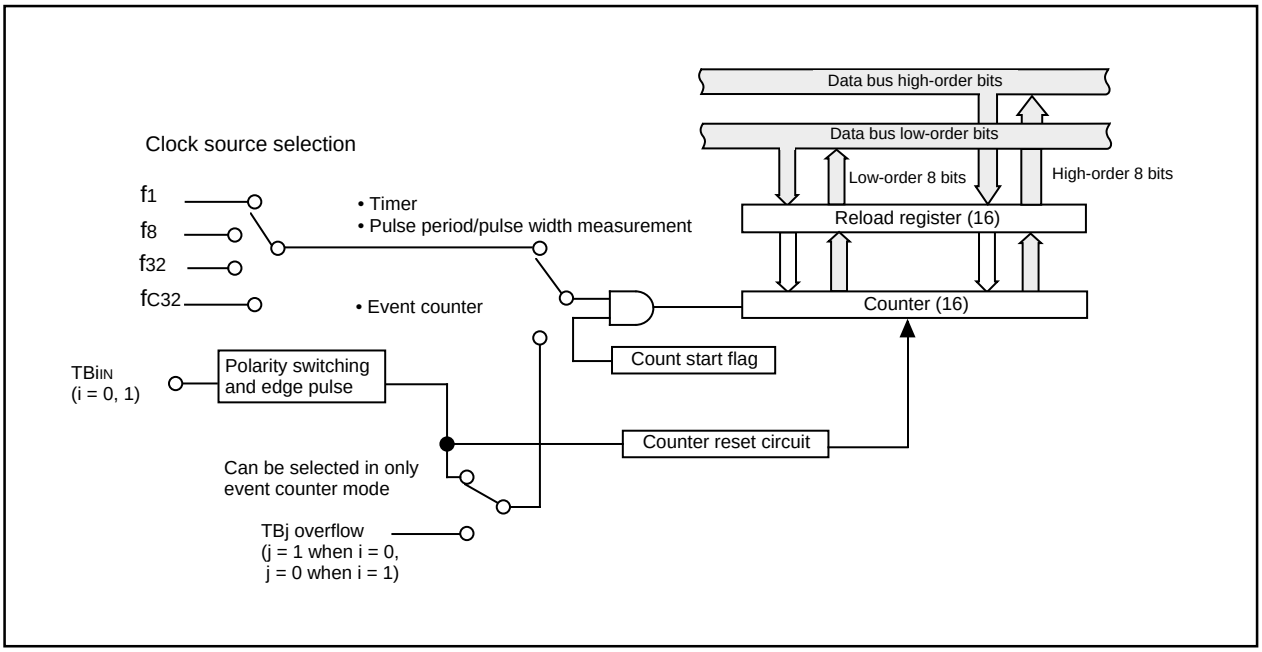


Figure 1.48. Block diagram of timer B

Timer Bi mode register

b7 b6 b5 b4 b3 b2 b1 b0

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

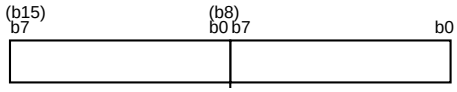
Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/pulse width measurement mode 1 1 : Inhibited	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○ (Note 1) × (Note 2)	○ ×
MR3			○	×
TCK0	Count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

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

Figure 1.49. Timer B-related registers (1)

Timer Bi register (Note)

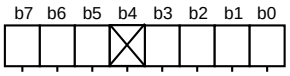
Symbol	Address	When reset
TB0	0391_{16}, 0390_{16}	Indeterminate
TB1	0393_{16}, 0392_{16}	Indeterminate



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.

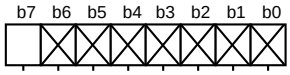
Count start flag



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. In an attempt to write to this bit, write “0”. The value, if read, turns out to be 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



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

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

Figure 1.50. Timer B-related registers (2)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.17.) Figure 1.51 shows the timer Bi mode register in timer mode.

Table 1.17. 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

b7	b6	b5	b4	b3	b2	b1	b0
			X			0	0

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

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. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.			—	—
MR3	Invalid in timer mode. In an attempt to write to this bit, write “0”. The value, if read in timer mode, turns out to be 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 1.51. 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 1.18.) Figure 1.52 shows the timer Bi mode register in event counter mode.

Table 1.18. Timer specifications in event counter mode

Item	Specification
Count source	• External signals input to TBiIN 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
TBiIN 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)

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
			☒			0	1

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

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	○	○
MR1		0 1 : Counts external signal's rising edges 1 0 : Counts external signal's falling and rising edges 1 1 : Inhibited	○	○
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.			—	—
MR3	Invalid in event counter mode. In an attempt to write to this bit, write "0". The value, if read in event counter mode, turns out to be indeterminate.		○	×
TCK0	Invalid in event counter mode. Can be "0" or "1".		○	○
TCK1	Event clock select	0 : Input from TBiIN pin (Note 2) 1 : TBj overflow (j = 1 when i = 0, j = 0 when i = 1)	○	○

Note 1: Valid only when input from the TBiIN 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 1.52. 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 1.19.) Figure 1.53 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure 1.54 shows the operation timing when measuring a pulse period. Figure 1.55 shows the operation timing when measuring a pulse width.

Table 1.19. 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

b7	b6	b5	b4	b3	b2	b1	b0
			⊠			1	0

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. In an attempt to write to this bit, write "0". The value, if read, turns out to be 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 1.53. Timer Bi mode register in pulse period/pulse width measurement mode

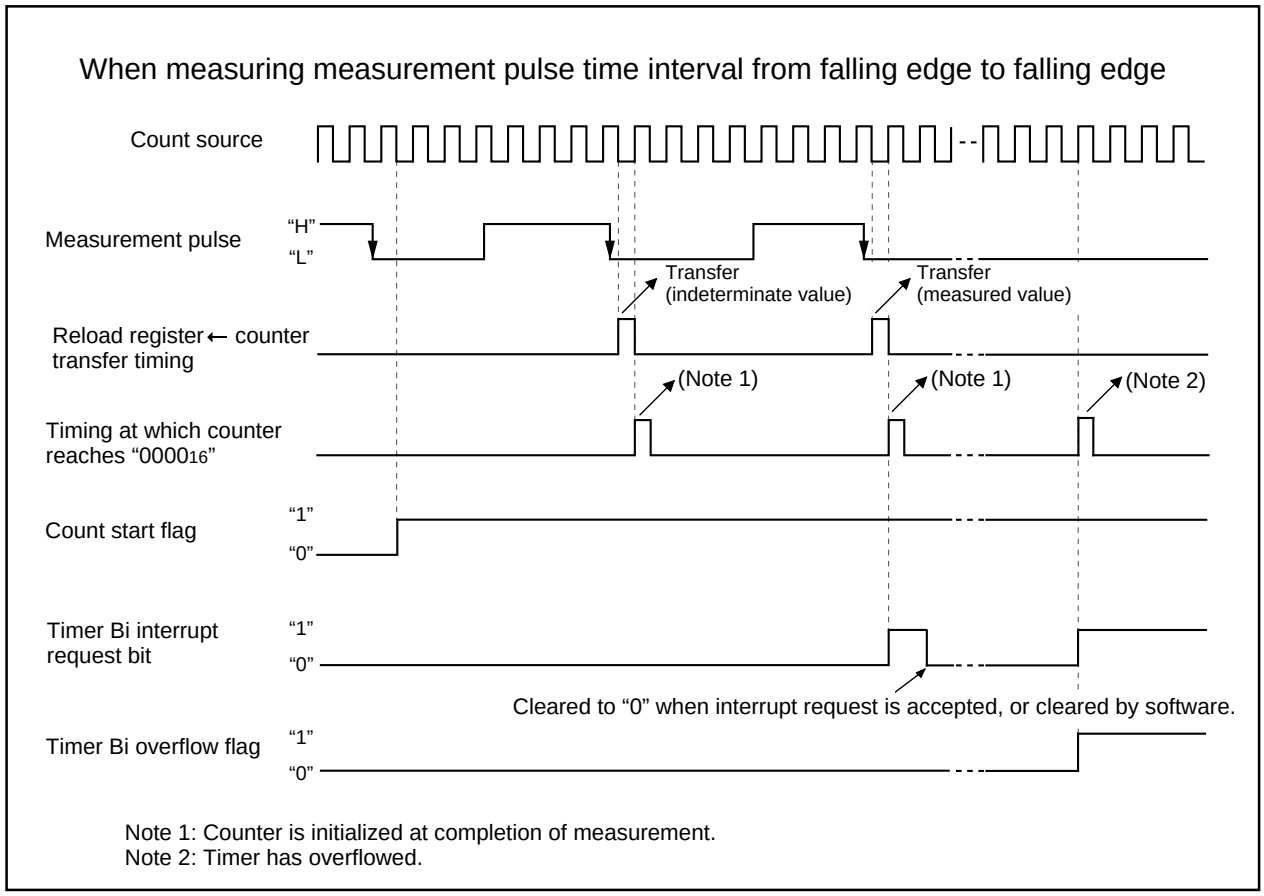


Figure 1.54. Operation timing when measuring a pulse period

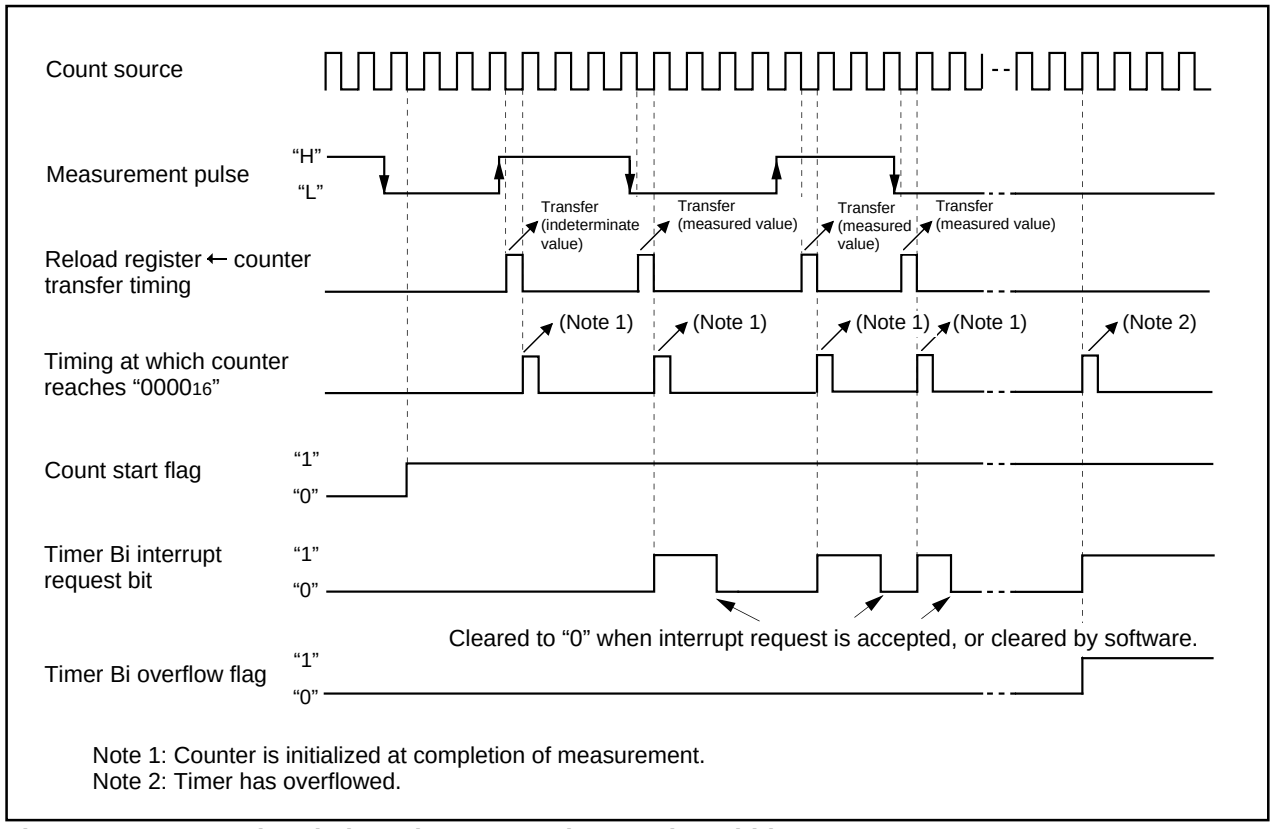


Figure 1.55. Operation timing when measuring a pulse width

Timer X

Figure 1.56 shows the block diagram of timer X. Figures 1.57 to 1.59 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.

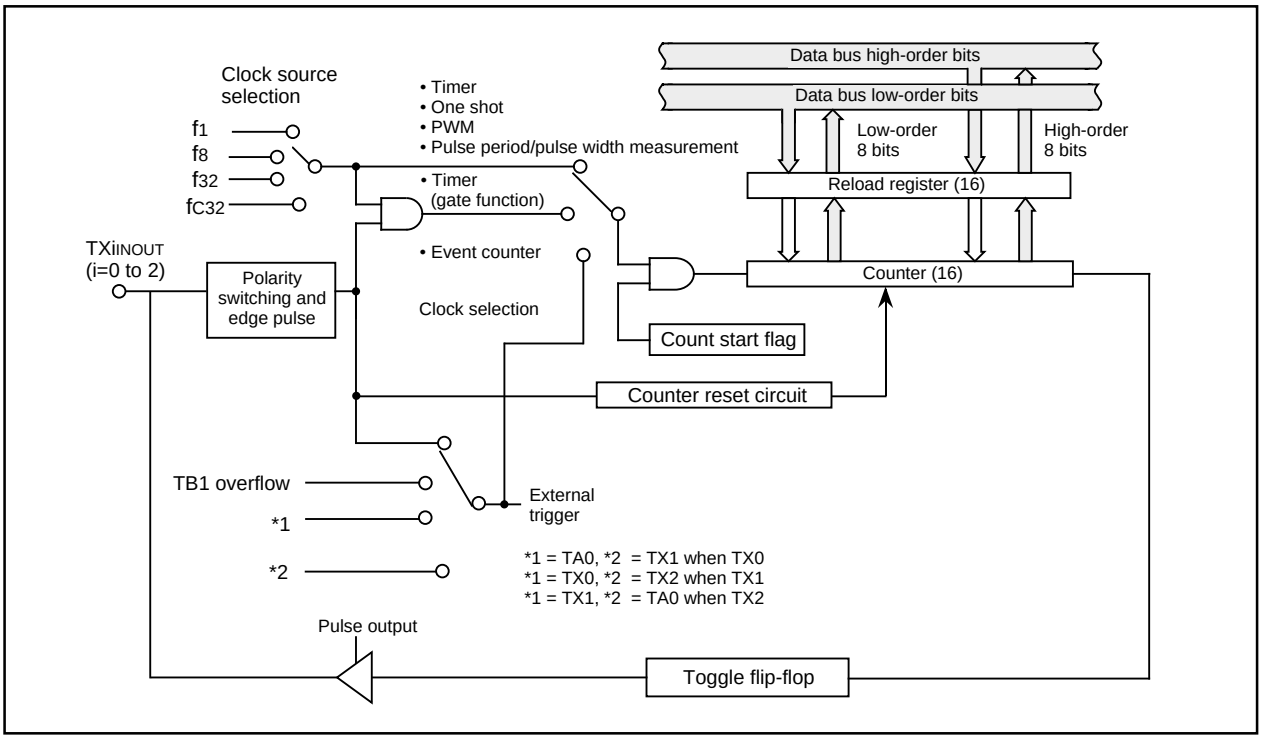


Figure 1.56. Block diagram of timer X

Timer Xi mode register

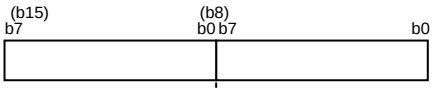
b7 b6 b5 b4 b3 b2 b1 b0

Symbol: TXiMR(i = 0 to 2) | Address: 039716 to 039916 | When reset: 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 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 1.57. Timer X-related registers (1)

Timer Xi register (Note 1)

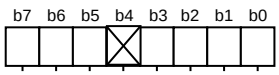


Symbol	Address	When reset
TX0	0389_{16},$038B_{16}$	Indeterminate
TX1	$038B_{16}$,$038A_{16}$	Indeterminate
TX2	$038D_{16}$,$038C_{16}$	Indeterminate

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 $FFFF_{16}$ (Note 2)	×	○
• 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 $FFFE_{16}$ (Note 2)	×	○
• 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} (Note 2) (High-order addresses) 00_{16} to FF_{16} (Note 2) (Low-order addresses)	×	○

Note 1: Read and write data in 16-bit units.
Note 2: Use MOV instruction to write to this register.

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. In an attempt to write to this bit, write “0”. The value, if read, turns out to be 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 1.58. Timer X-related registers (2)

One-shot start flag

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. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—

Trigger select register

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

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

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

Figure 1.59. Timer X-related registers (3)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.20.) Figure 1.60 shows the timer Xi mode register in timer mode.

Table 1.20. 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

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol: TXiMR(i = 0 to 2) Address: 039716 to 039916 When reset: 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 1.60. 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 1.21.) Figure 1.61 shows the timer Xi mode register in event counter mode.

Table 1.21. 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.

Timer Xi mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

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 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 "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 1.61. Timer Xi mode register in event counter mode

(3) One-shot timer mode

In this mode, the timer operates only once. (See Table 1.22.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 1.62 shows the timer Xi mode register in one-shot timer mode.

Table 1.22. 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}
TXiINOUT 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

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

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 (TXiINOOUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TXiINOOUT pin is a pulse output pin)	○	○
MR1	External trigger select bit (Note 2)	0 : Falling edge of TXiINOOUT pin's input signal (Note 3) 1 : Rising edge of TXiINOOUT 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 : f_{C32}	○	○
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 TXiINOUT 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 TXiINOUT pin is selected by the event/trigger select bit (addresses 0383_{16}).

Figure 1.62. 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 1.23.) Figure 1.63 shows the timer Xi mode register in pulse period/pulse width measurement mode. Figure 1.64 shows the operation timing when measuring a pulse period. Figure 1.65 shows the operation timing when measuring a pulse width.

Table 1.23. 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.)
TXiINOUT 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.

Timer Xi mode register

b7	b6	b5	b4	b3	b2	b1	b0
		1				1	0

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			○	○
MR2	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 1.63. Timer Xi mode register in pulse period/pulse width measurement mode

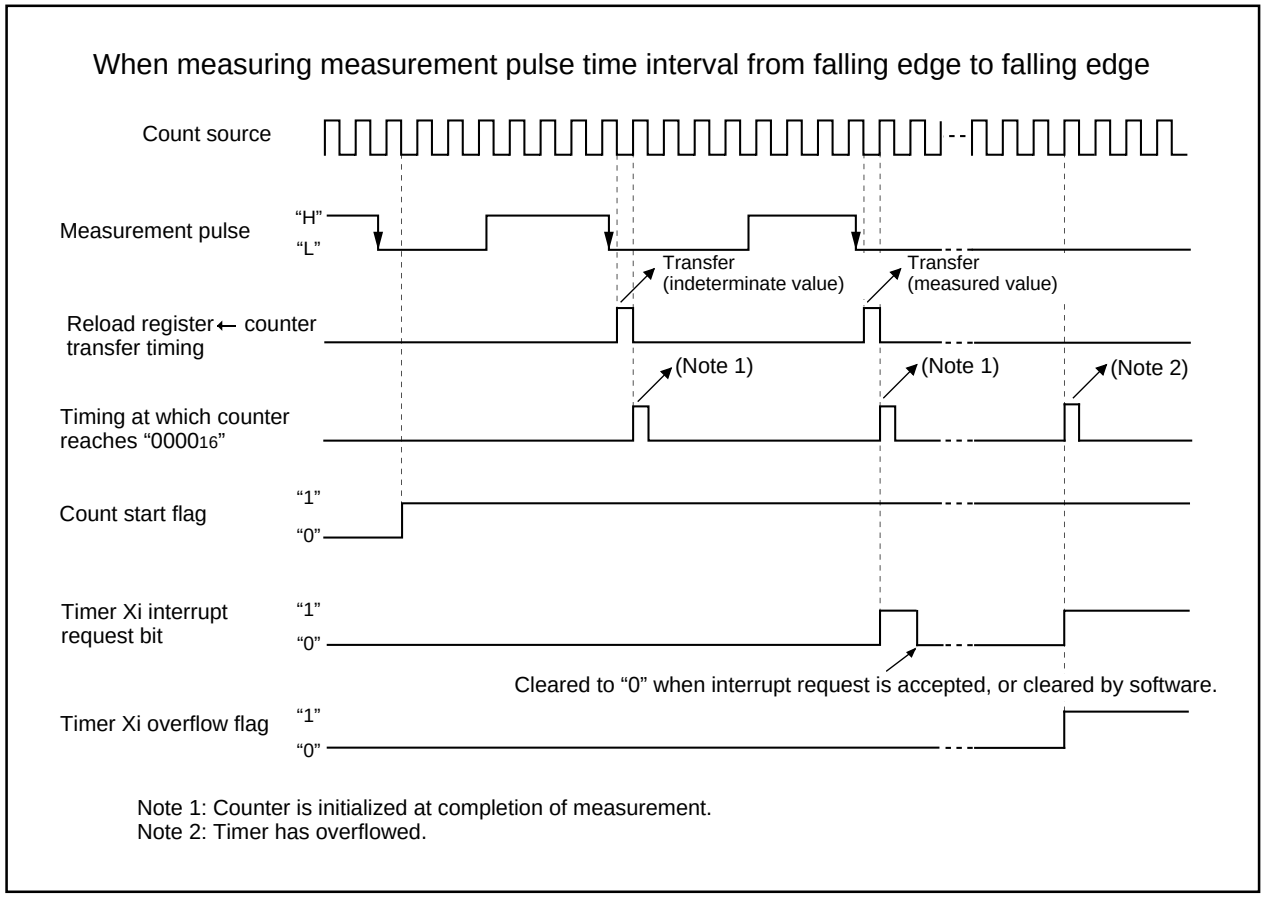


Figure 1.64. Operation timing when measuring a pulse period

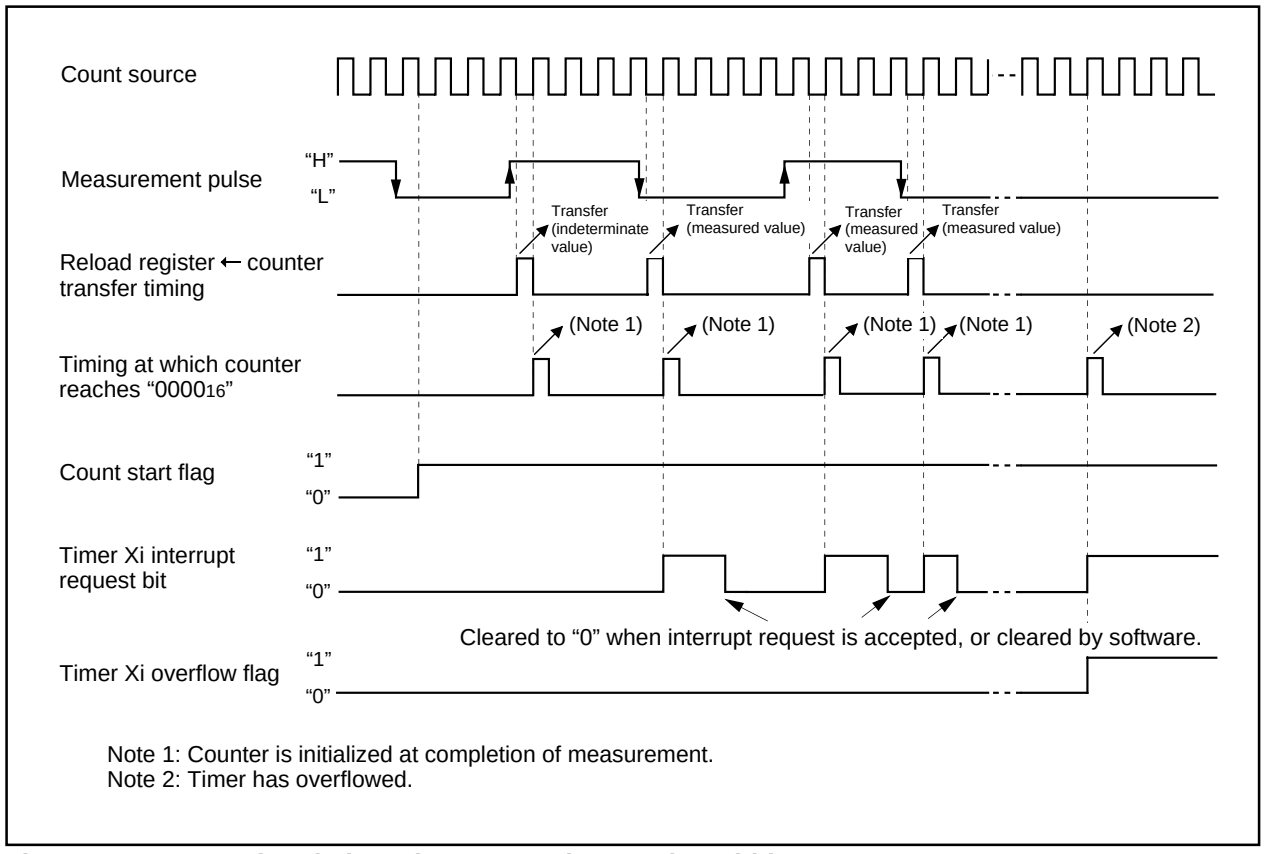


Figure 1.65. Operation timing when measuring a pulse width

(5) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table 1.24.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 1.66 shows the timer Xi mode register in pulse width modulation mode. Figure 1.67 shows the example of how a 16-bit pulse width modulator operates. Figure 1.68 shows the example of how an 8-bit pulse width modulator operates.

Table 1.24. Timer specifications in pulse width modulation mode

Item		Specification
Count source		f_1, f_8, f_{32}, f_{C32}
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 $n \times (m+1)$/ fi n:values set to timer Xi register's high-order address • Cycle time $(2^8-1) \times (m+1)$ / 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

b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

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 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 1.66. Timer Xi mode register in pulse width modulation mode

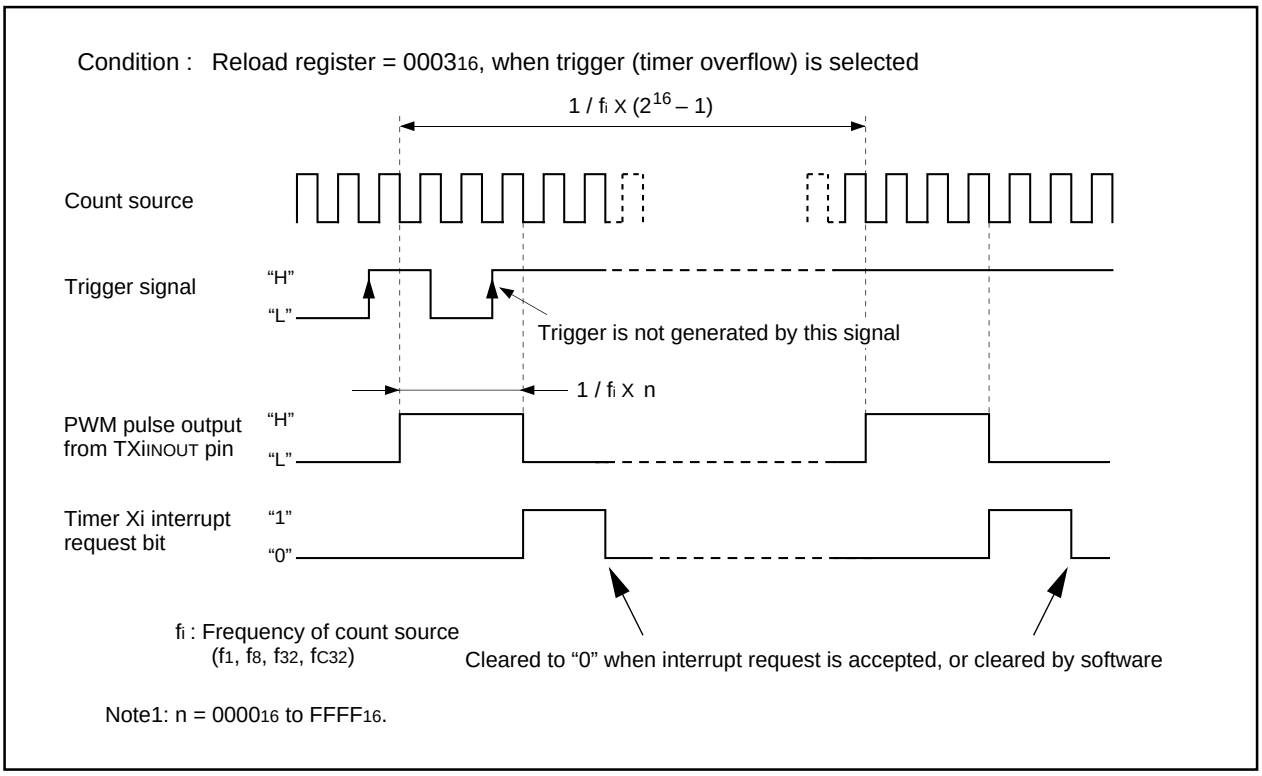


Figure 1.67. Example of how a 16-bit pulse width modulator operates

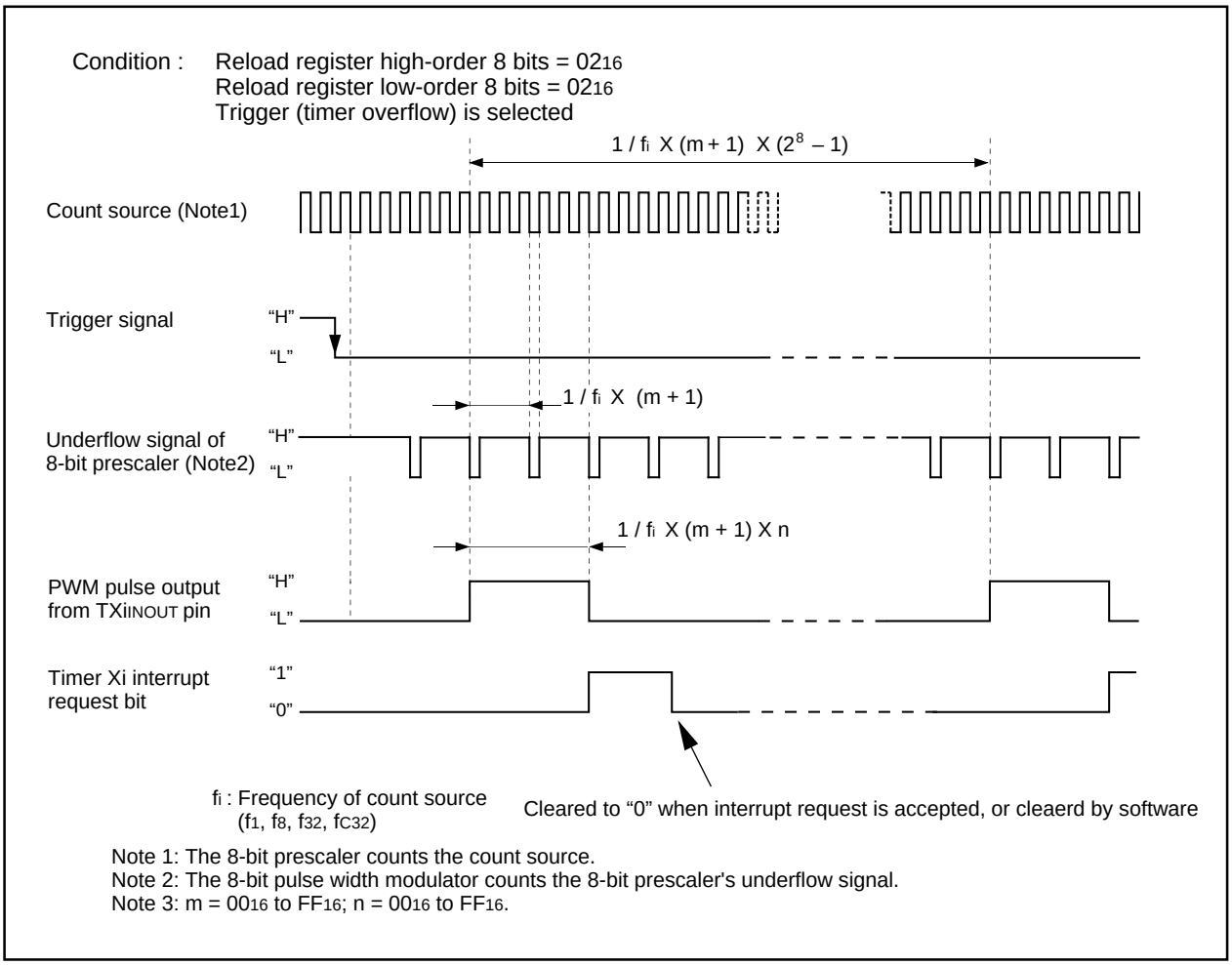


Figure 1.68. Example of how an 8-bit pulse width modulator operates

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 1.69 shows the block diagram of UART0 and UART1. Figure 1.70 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 1.71 through 1.73 show the registers related to UARTi.

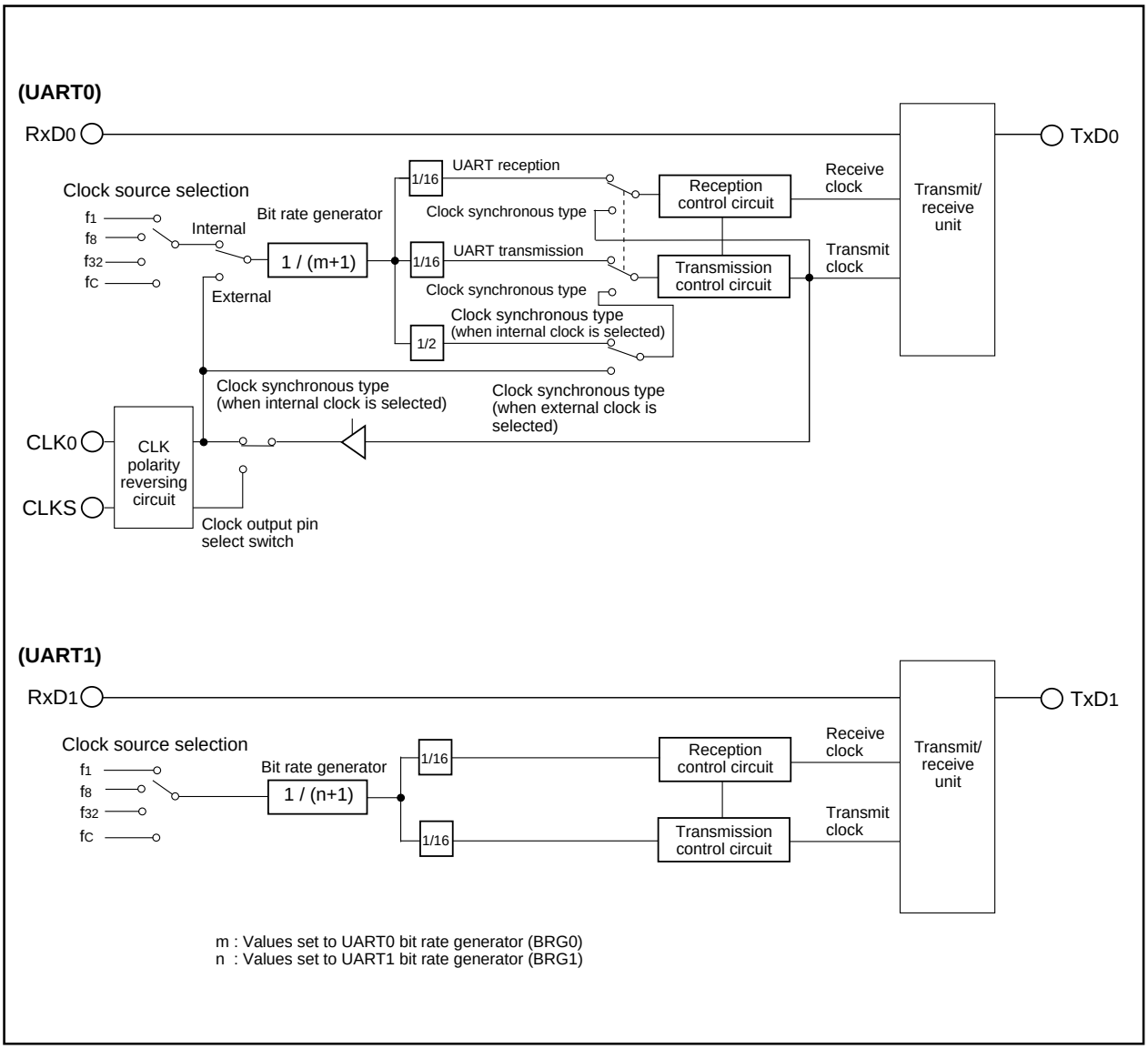
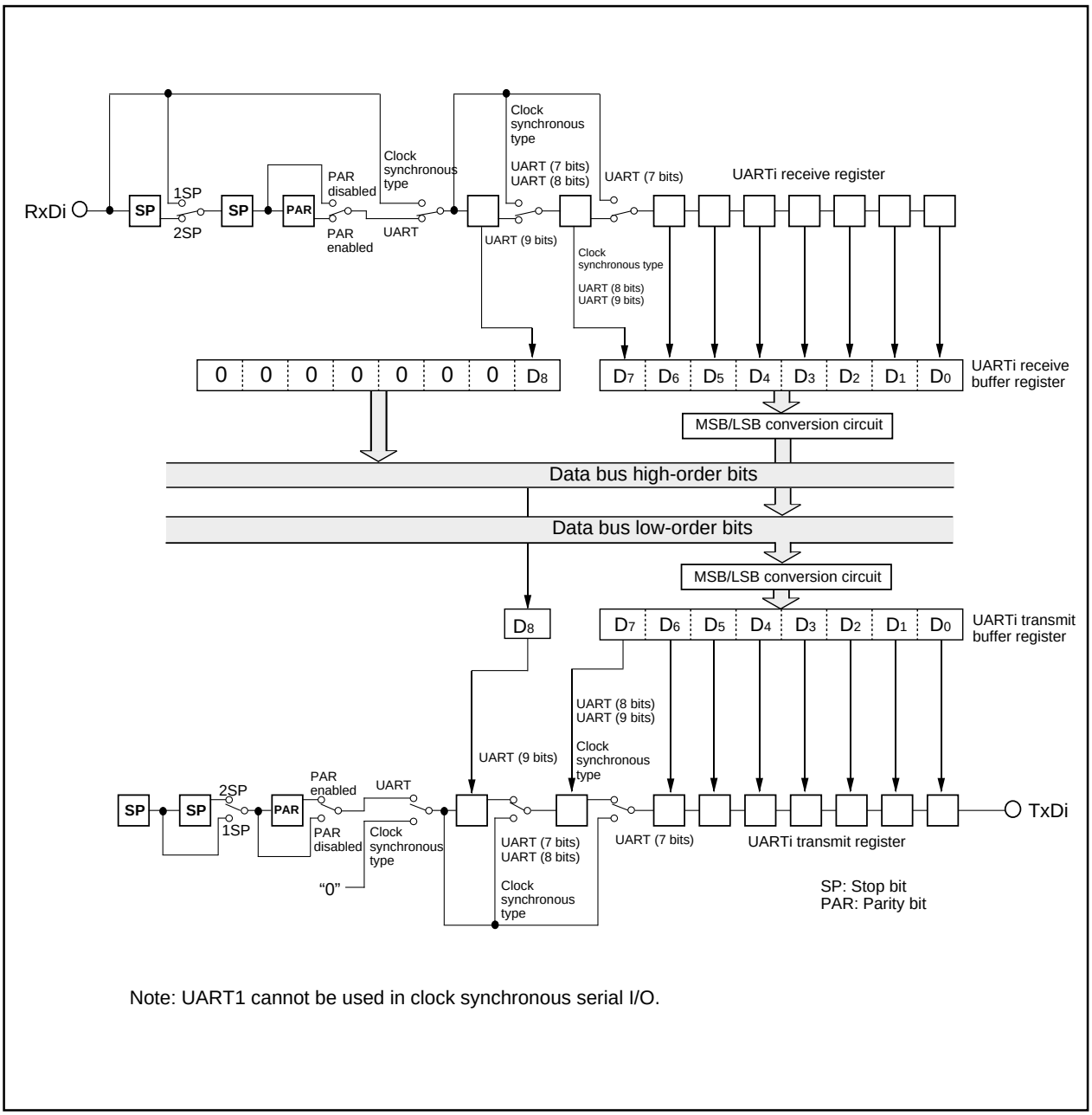


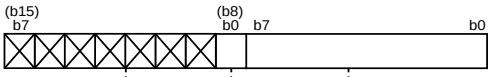
Figure 1.69. Block diagram of UARTi (i = 0, 1)



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

Figure 1.70. Block diagram of transmit/receive unit

UARTi transmit buffer register (Note)

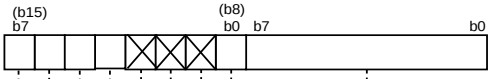


Symbol	Address	When reset
U0TB	$03A3_{16}$, $03A2_{16}$	Indeterminate
U1TB	$03AB_{16}$, $03AA_{16}$	Indeterminate

	R	W
Transmit data	×	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.	—	—

Note : Use MOV instruction to write to this register.

UARTi receive buffer register

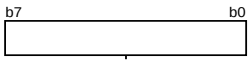


Symbol	Address	When reset
U0RB	$03A7_{16}$, $03A6_{16}$	Indeterminate
U1RB	$03AF_{16}$, $03AE_{16}$	Indeterminate

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. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.				—	—
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 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 $03AE_{16}$) is read out.

UARTi bit rate generator (Note 1, 2)



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}	×	○

Note 1: Write a value to this register while transmit/receive halts.
Note 2: Use MOV instruction to write to this register.

Figure 1.71. Serial I/O-related registers (1)

UARTi transmit/receive mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: UiMR(i=0,1) | Address: $03A0_{16}$, $03A8_{16}$ | When reset: 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 (Note 3) 1 : External clock (Note 4)	0 : Internal clock (Note 3) 1 : External clock (Note 4)	○	○
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".
Note 3: Set the corresponding port direction register to "1" (output mode).
Note 4: Set the corresponding port direction register to "0" (input mode).

UARTi transmit/receive control register 0

b7 b6 b5 b4 b3 b2 b1 b0 (1 at b4, 0 at b2)

Symbol: UiC0(i=0,1) | Address: $03A4_{16}$, $03AC_{16}$ | When reset: 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 "1".				○	○
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 1.72. Serial I/O-related registers (2)

UARTi transmit/receive control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: UiC1(i=0,1) | Address: $03A5_{16}$,$03AD_{16}$ | When reset: 02_{16}

Bit symbol	Bit name	Function (Note 1) (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	O	O
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	O	X
RE	Receive enable bit (Note 2)	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	O	O
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	O	X
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.				—	—

Note 1: UART1 cannot be used in clock synchronous serial I/O.
Note 2: If you are using clock asynchronous serial I/O mode, you can enable 'receive enable bit' when RxD port input is "H". If RxD port input is "L" and you have enabled 'receive enable bit' , then receive operation starts immediately.

UART transmit/receive control register 2

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: UCON | Address: $03B0_{16}$ | When reset: $XX000000_2$

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)	O	O
U1IRS	UART1 transmit interrupt cause select bit	Set this bit to "0".	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	O	O
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	Must always be "0"	O	O
Set this bit to "0".				O	O
CLKMD0	CLK/CLKS select bit 0	Valid when bit 5 = "1" 0 : Clock output to CLK1 1 : Clock output to CLKS1	Must always be "0"	O	O
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"	O	O
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be 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 1.73. 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 Table 1.25.) Figure 1.65 shows the UART0 transmit/receive mode register.

Table 1.25. 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 does not change.

UART0 transmit/receive mode registers

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

Symbol: U0MR
Address: $03A0_{16}$
When reset: 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 (Note 1) 1 : External clock (Note 2)	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
SLEP	0 (Must always be “0” in clock synchronous serial I/O mode)		○	○

Note 1: Set the corresponding port direction register to “1” (output mode).
Note 2: Set the corresponding port direction register to “0” (input mode).

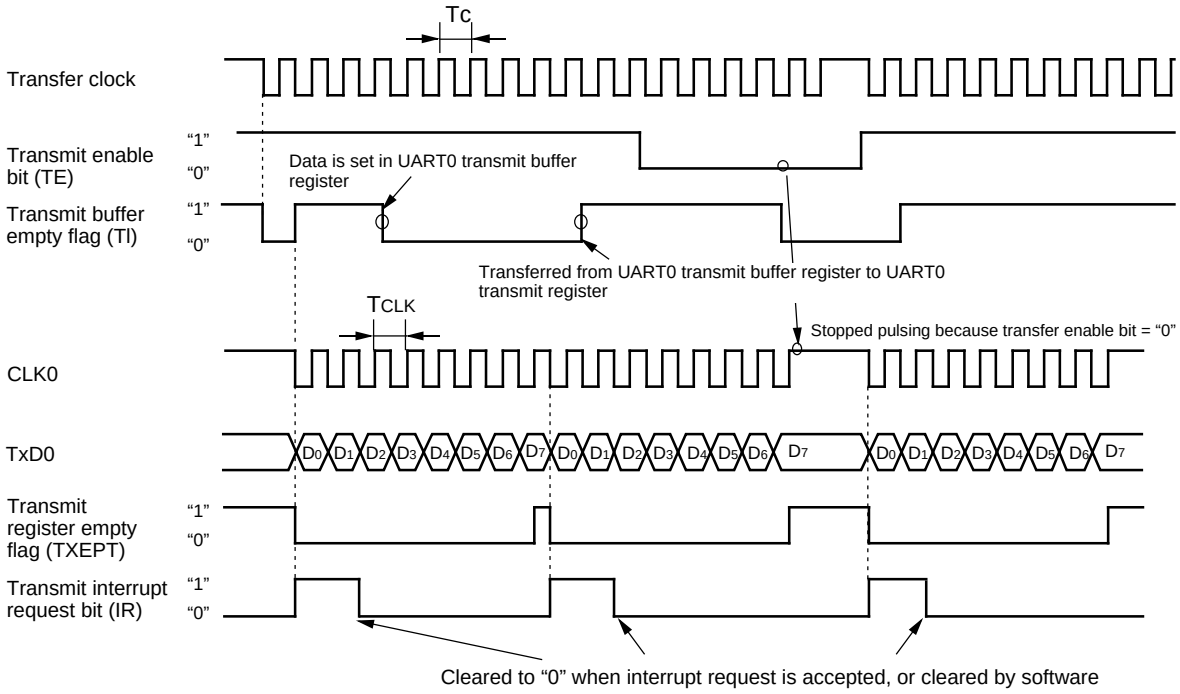
Figure 1.74. UART0 transmit/receive mode register in clock synchronous serial I/O mode

Table 1.26 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 1.26. 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)

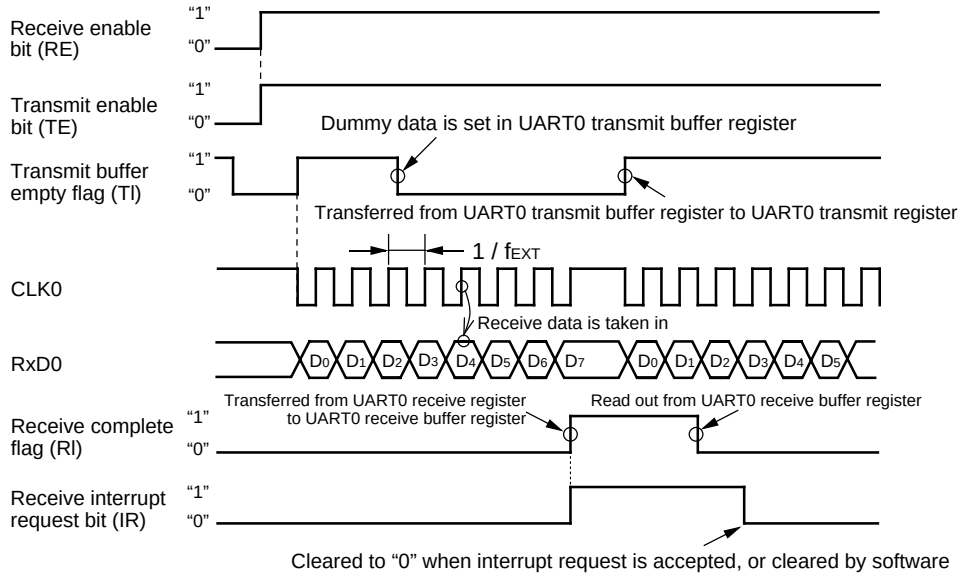


Shown in ( ) are bit symbols.

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

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

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



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

(a) Polarity select function

As shown in Figure 1.76, the CLK polarity select bit (bit 6 at addresses 03A4$_{16}$) allows selection of the polarity of the transfer clock.

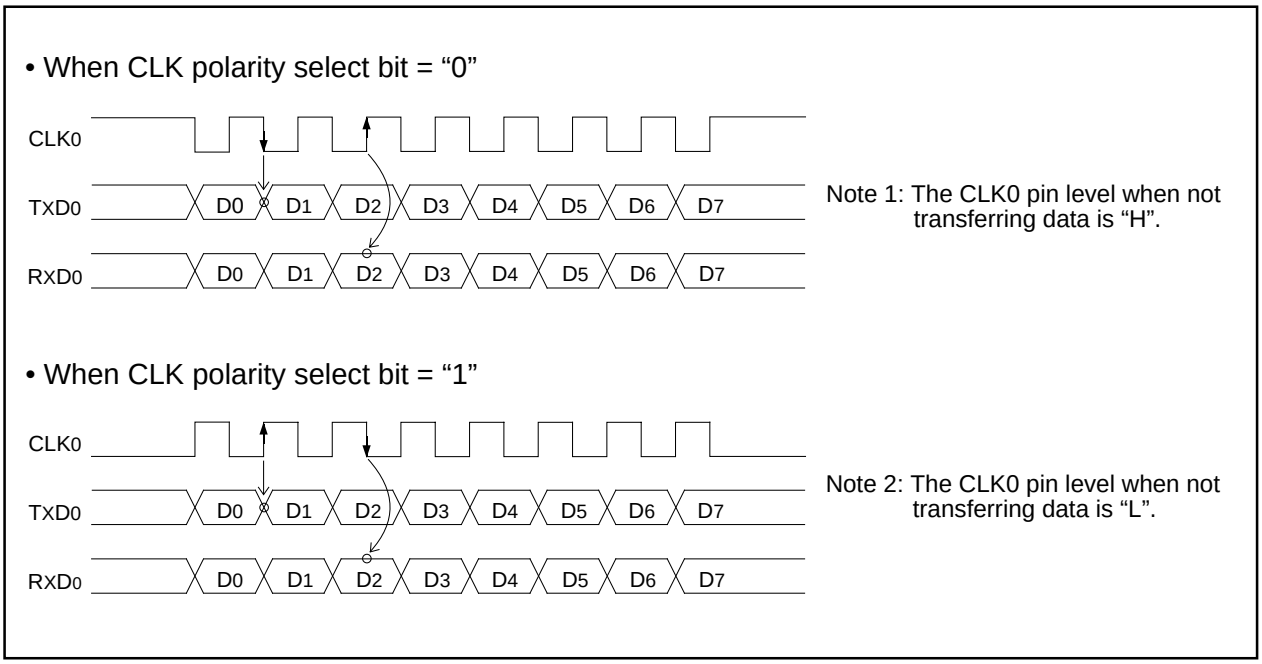


Figure 1.76. Polarity of transfer clock

(b) LSB first/MSB first select function

As shown in Figure 1.77, 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”.

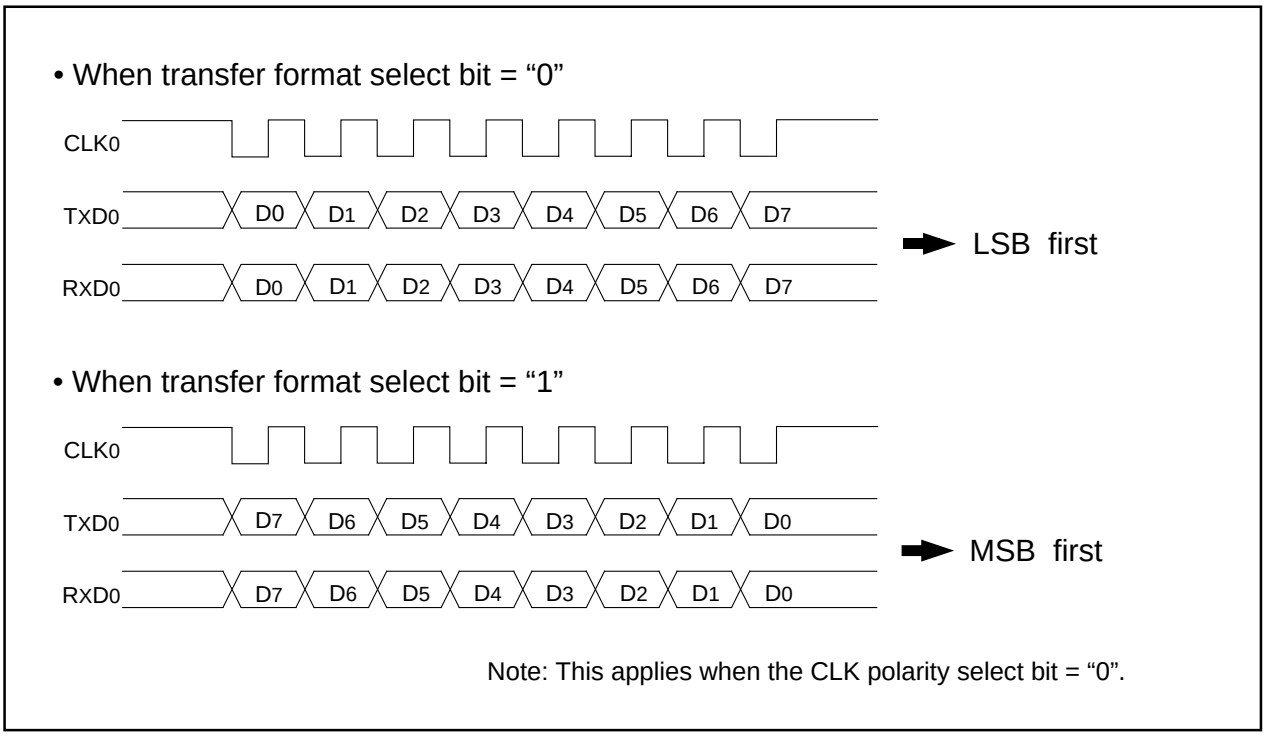
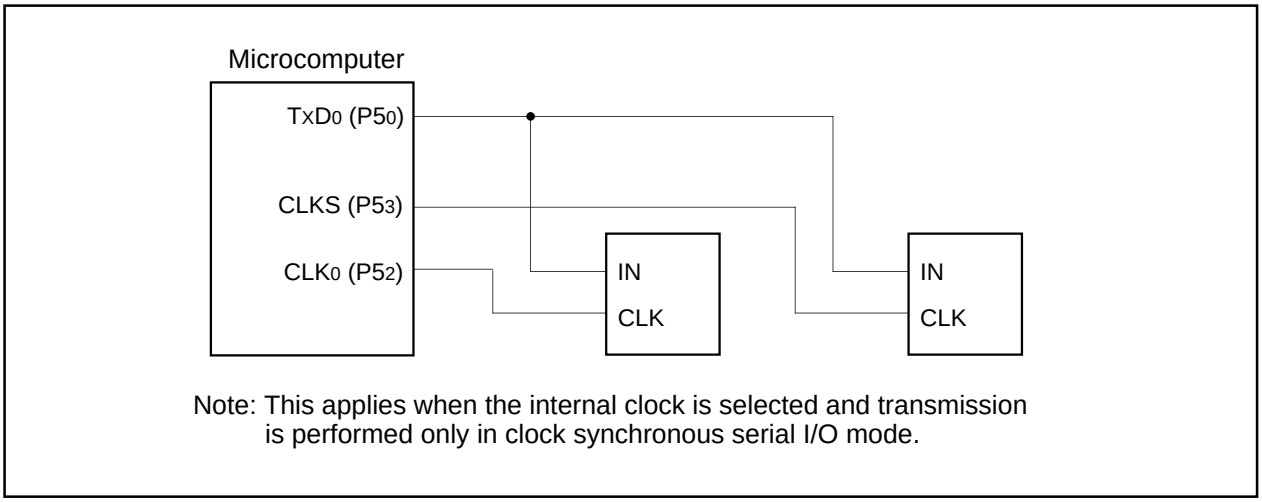


Figure 1.77. Transfer format

(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 1.78.) 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 1.78. 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 Table 1.27.) Figure 1.79 shows the UARTi transmit/receive mode register.

Table 1.27. 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 does not change.

UARTi transmit / receive mode registers

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: UiMR(i=0,1) — Address: $03A0_{16}$, $03A8_{16}$ — When reset: 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 1)	0 : Internal clock (Note 2) 1 : External clock (Note 3)	○	○
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 1: UART1 can use only internal clock. Must set this bit to "1".
Note 2: Set the corresponding port direction register to "1" (output mode).
Note 3: Set the corresponding port direction register to "0" (input mode).

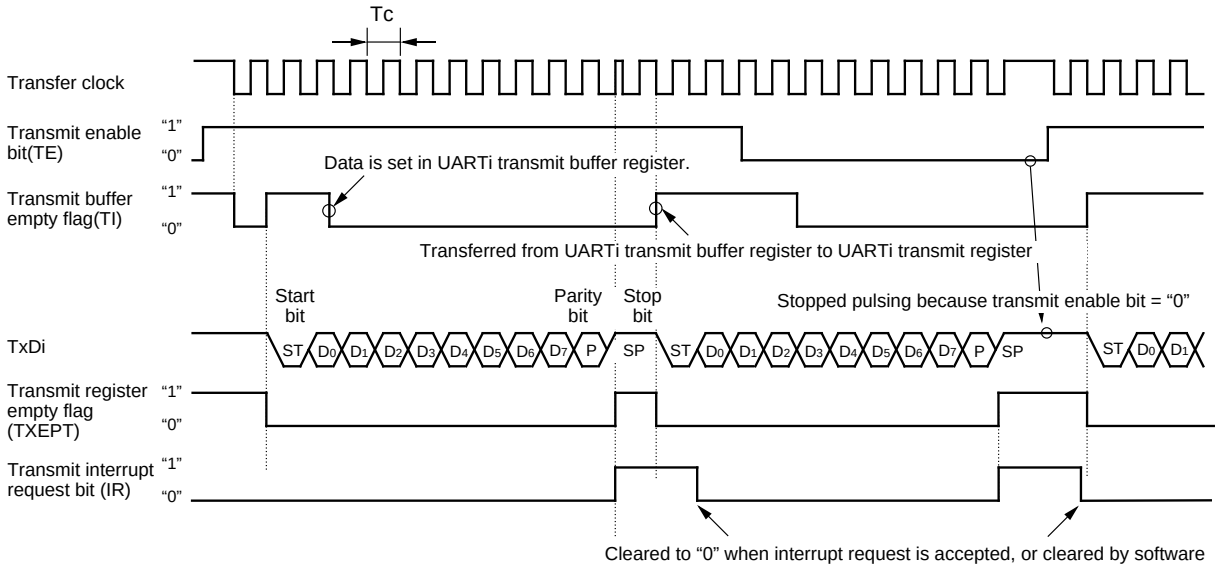
Figure 1.79. UARTi transmit/receive mode register in UART mode

Table 1.28 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 1.28. 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)

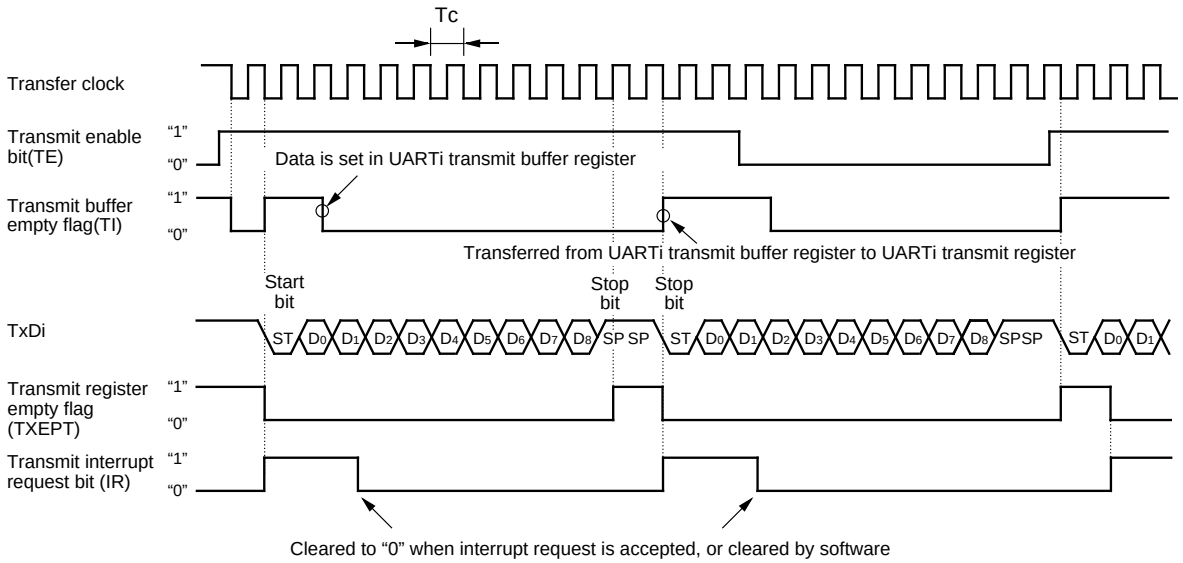


Shown in ( ) are bit symbols.

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

$Tc = 16 (n + 1) / fi$ or $16 (n + 1) / f_{EXT}$
fi : 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)



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".

$Tc = 16 (n + 1) / fi$ or $16 (n + 1) / f_{EXT}$
fi : 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 1.80. Typical transmit timings in UART mode

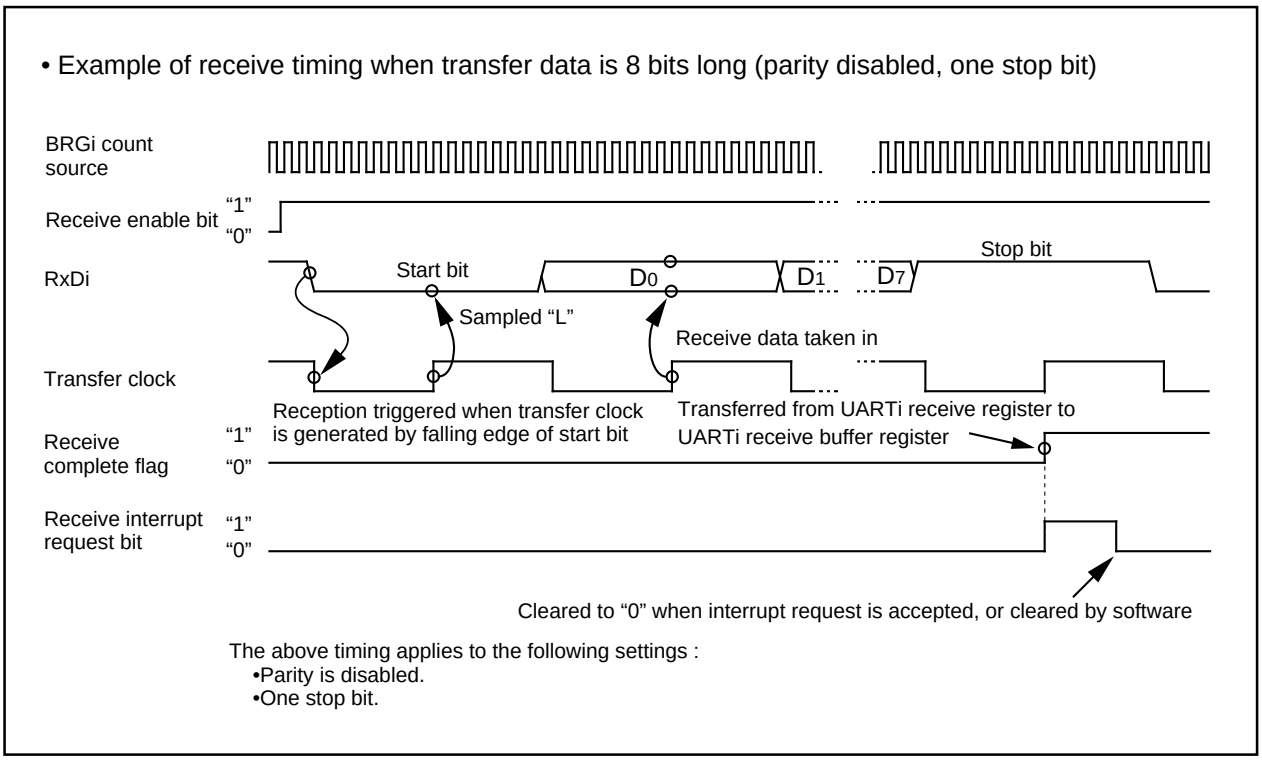


Figure 1.81. Typical receive timing in UART mode

(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 (V_{REF}) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from V_{REF}, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of $03D7_{16}$ to connect V_{REF}.

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 1.29 shows the performance of the A-D converter. Figure 1.82 shows the block diagram of the A-D converter, and Figures 1.83 and 1.84 show the A-D converter-related registers.

Table 1.29. Performance of A-D converter

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AV_{CC} (V_{CC})
Operating clock ϕ_{AD} (Note 2)	V_{CC} = 5V f_{AD}, divide-by-2 of f_{AD}, divide-by-4 of f_{AD}, $f_{AD}=f(X_{IN})$ V_{CC} = 3V divide-by-2 of f_{AD}, divide-by-4 of f_{AD}, $f_{AD}=f(X_{IN})$
Resolution	8-bit or 10-bit (selectable)
Absolute precision	V_{CC} = 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 V_{CC} = 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 ϕ_{AD} cycles, 10-bit resolution: 59 ϕ_{AD} cycles • With sample and hold function 8-bit resolution: 28 ϕ_{AD} cycles, 10-bit resolution: 33 ϕ_{AD} cycles

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

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

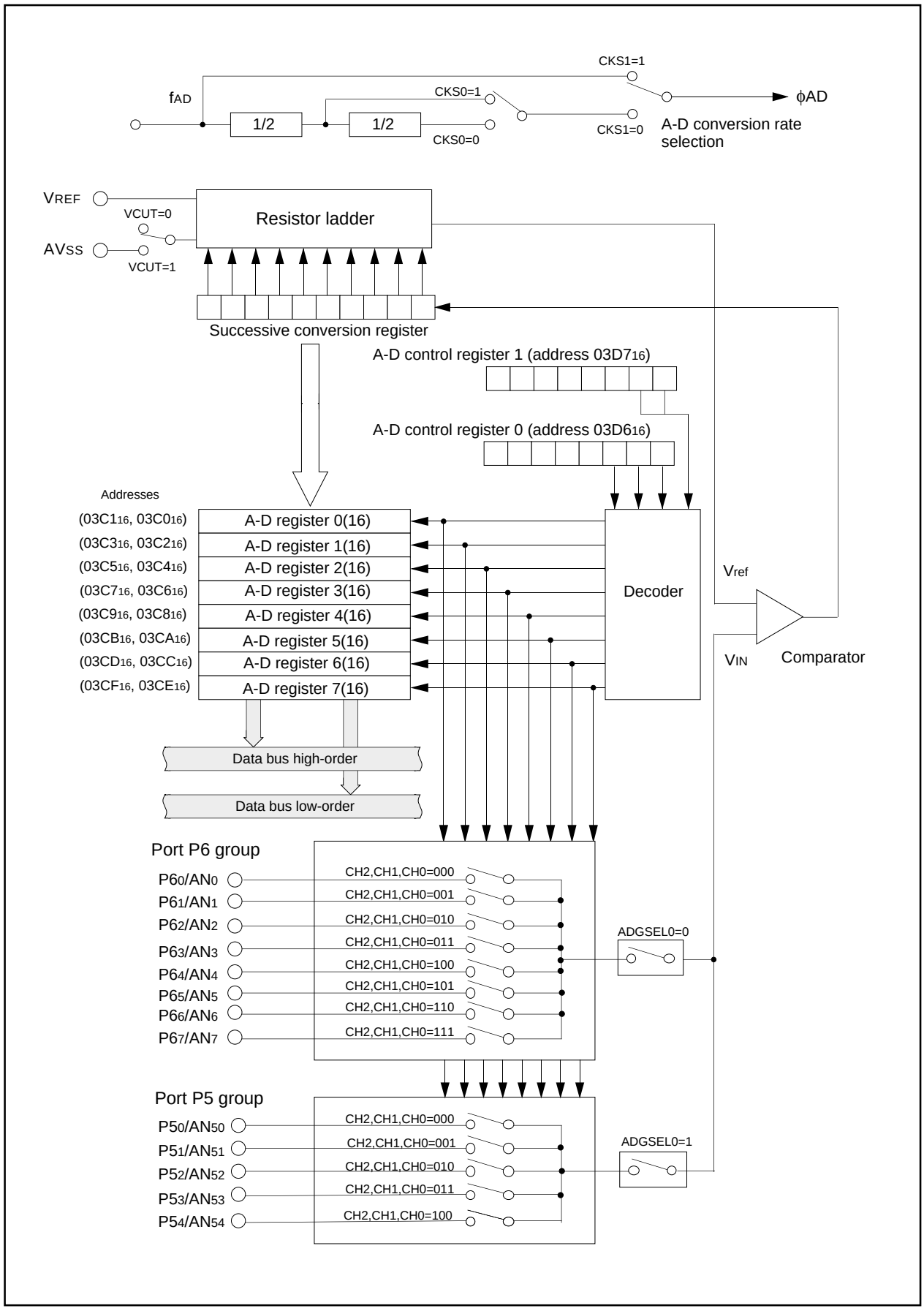


Figure 1.82. Block diagram of A-D converter

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, 3)	○	○
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 (Note 2)	○	○
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: When changing A-D operation mode, set analog input pin again.
Note 3: 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						

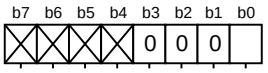
Symbol	Address	When reset
ADCON1	$03D7_{16}$	00_{16}

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 port P5 group is selected, the contents of A-D registers 5 to 7 are indeterminate. If port P5 group is selected, do not select 8 pins sweep mode.

Figure 1.83. A-D converter-related registers (1)

A-D control register 2 (Note)



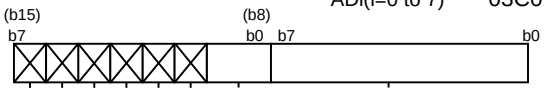
Symbol: ADCON2 | Address: $03D4_{16}$ | When reset: $XXXX0000_2$

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. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			–	–

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

A-D register i

Symbol: ADi(i=0 to 7) | Address: $03C0_{16}$ to $03CF_{16}$ | When reset: 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 The value, if read, turns out to be indeterminate.	×	×
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.	–	–

Figure 1.84. 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 1.30.) Figure 1.85 shows the A-D control register in one-shot mode.

Table 1.30. 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, 3)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode (Note 2)	○	○
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: When changing A-D operation mode, set analog input pin again.
Note 3: 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 1.85. 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 1.31.) Figure 1.86 shows the A-D control register in repeat mode.

Table 1.31. 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 (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 1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset
		0	0	1				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, 3)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 1 : Repeat mode (Note 2)	○	○
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: When changing A-D operation mode, set analog input pin again.
Note 3: 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	Symbol	Address	When reset
	0	1			0			ADCON1	$03D7_{16}$	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 1.86. 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 1.32.) Figure 1.87 shows the A-D control register in single sweep mode.

Table 1.32. 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: ADCON0 — Address: $03D6_{16}$ — When reset: $00000XXX_2$

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

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)	O	O
SCAN1			O	O
MD2	A-D operation mode select bit 1	Set this bit to "0" in this mode.	O	O
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	O	O
VCUT	Vref connect bit	1 : Vref connected	O	O
Set this bit to "0".			O	O
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	O	O

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. If port P5 group is selected, do not select 8 pins sweep mode.

Figure 1.87. 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 1.33.) Figure 1.88 shows the A-D control register in repeat sweep mode 0.

Table 1.33. 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	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 0	O	O
MD1			O	O
Set this bit to "0".			O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	O	O

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)	O	O
SCAN1			O	O
MD2	A-D operation mode select bit 1	Set this bit to "0" in this mode.	O	O
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	O	O
VCUT	Vref connect bit	1 : Vref connected	O	O
Set this bit to "0".			O	O
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	O	O

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. If port P5 group is selected, do not select 8 pins sweep mode.

Figure 1.88. 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 1.34.) Figure 1.89 shows the A-D control register in repeat sweep mode 1.

Table 1.34. 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)	○	○
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.

Figure 1.89. 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 1.90 to 1.92 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 1.93 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 1.94 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 1.95 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 1.95 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.

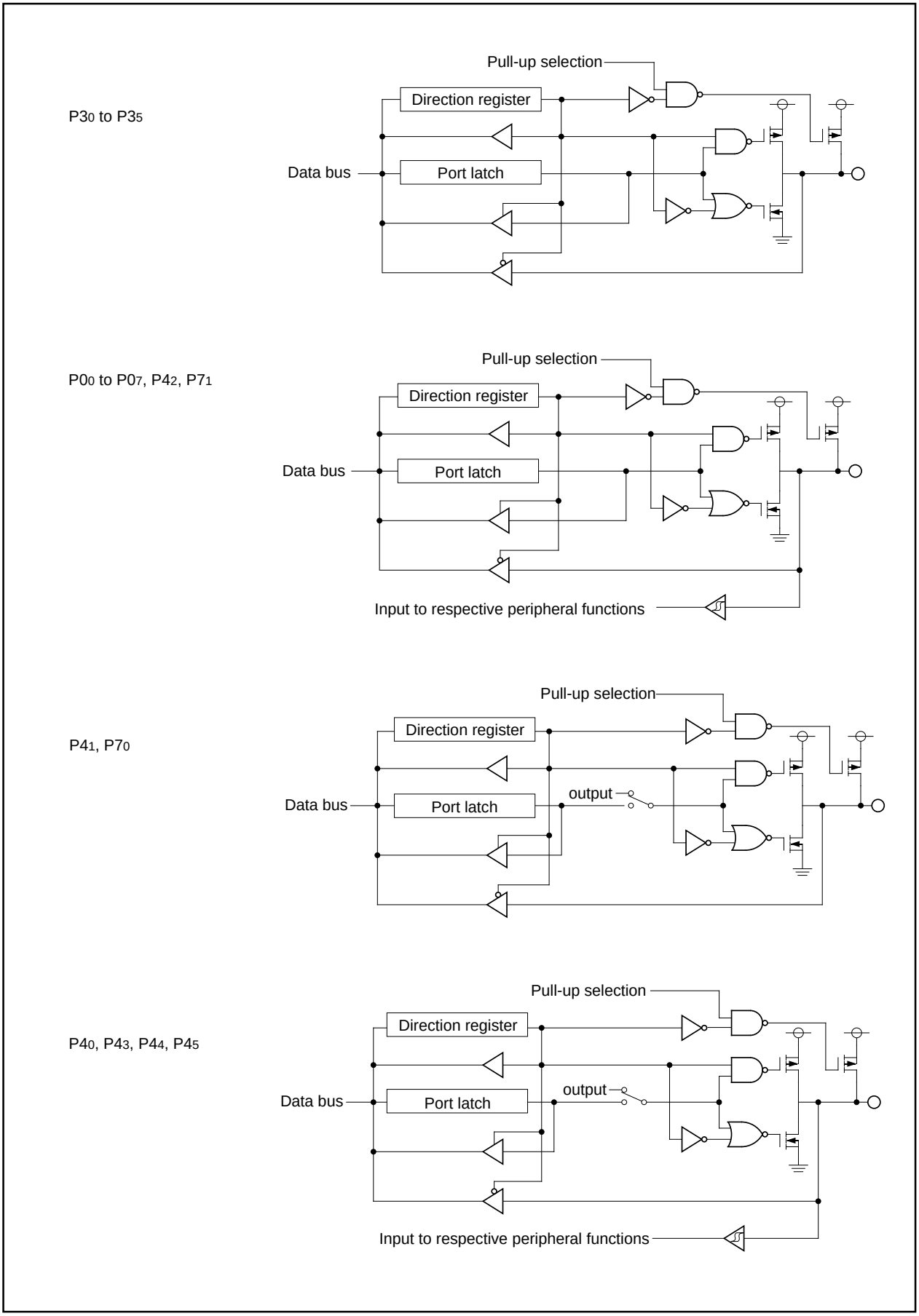


Figure 1.90. Programmable I/O ports (1)

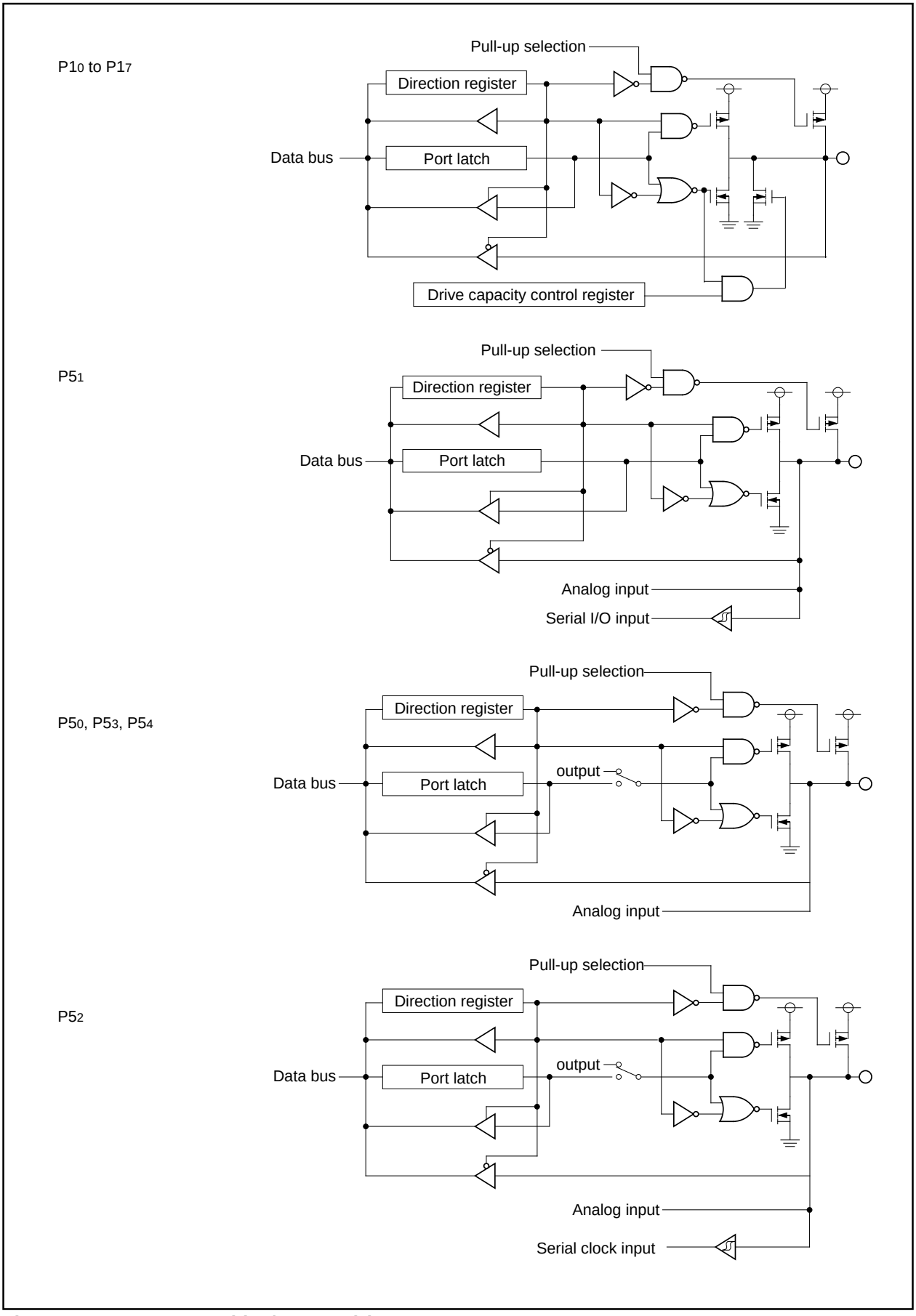


Figure 1.91. Programmable I/O ports (2)

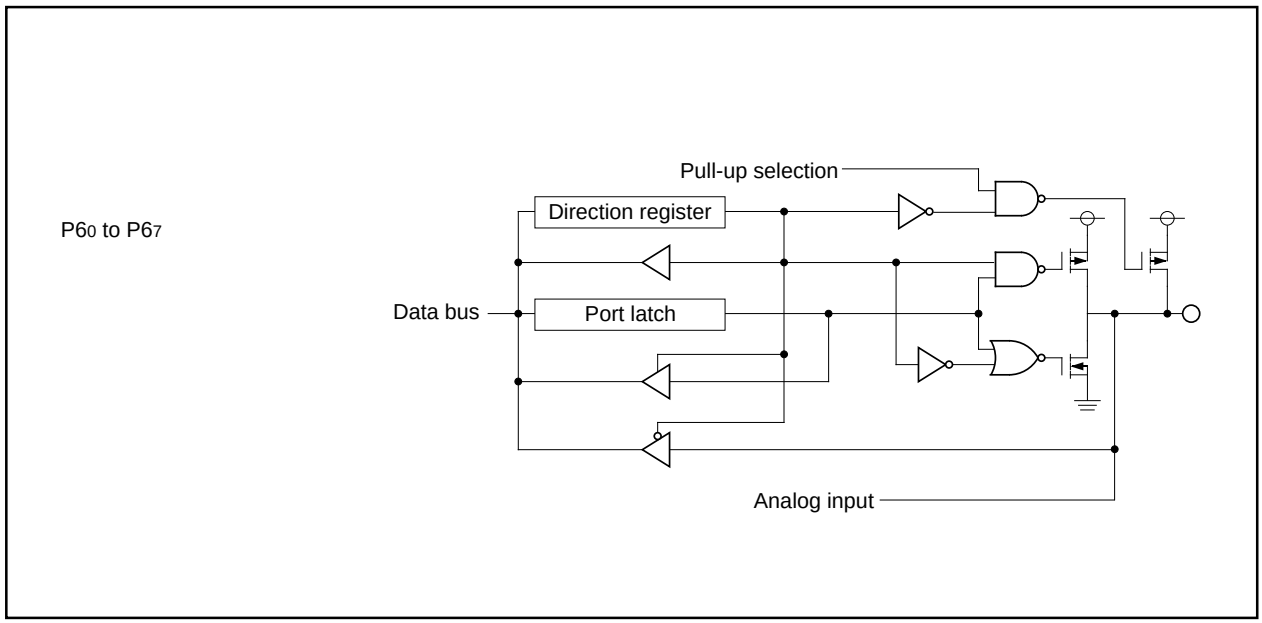


Figure 1.92. Programmable I/O ports (3)

Port Pi direction register (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PDi (i = 0 to 7)	03E2$_{16}$, 03E3$_{16}$, 03E7$_{16}$, 03EA$_{16}$, 03EB$_{16}$, 03EE$_{16}$, 03EF$_{16}$	00$_{16}$ 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 except 2)	○	○
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 1.93. Direction register

Port Pi register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
Pi (i = 0 to 7)	$03E0_{16}$, $03E1_{16}$, $03E5_{16}$, $03E8_{16}$, $03E9_{16}$, $03EC_{16}$, $03ED_{16}$	Indeterminate 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 except 2)	○	○
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 1.94. Port register

Pull-up control register 0

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUR0	03FC16	0016

Bit symbol	Bit name	Function	R	W
PU00	P00 to P03 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU01	P04 to P07 pull-up		○	○
PU02	P10 to P13 pull-up		○	○
PU03	P14 to P17 pull-up		○	○
—	—		○	○
—	—		○	○
PU06	P30 to P33 pull-up		○	○
PU07	P34 to P35 pull-up		○	○

Pull-up control register 1

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUR1	03FD16	0016

Bit symbol	Bit name	Function	R	W
PU10	P40 to P43 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU11	P44 to P47 pull-up		○	○
PU12	P50 to P53 pull-up		○	○
PU13	P54 pull-up		○	○
PU14	P60 to P63 pull-up		○	○
PU15	P64 to P67 pull-up		○	○
PU16	P70 to P71 pull-up		○	○
—	—		○	○

Port P1 drive capacity control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
DRR	03FE16	0016

Bit symbol	Bit name	Function	R	W
DRR0	Port P10 drive capacuty	Set P1 N-channel output transistor drive capacity 0 : LOW 1 : HIGH	○	○
DRR1	Port P11 drive capacuty		○	○
DRR2	Port P12 drive capacuty		○	○
DRR3	Port P13 drive capacuty		○	○
DRR4	Port P14 drive capacuty		○	○
DRR5	Port P15 drive capacuty		○	○
DRR6	Port P16 drive capacuty		○	○
DRR7	Port P17 drive capacuty		○	○

Figure 1.95. Pull-up control register

Example connection of unused pins

Table 1.36. 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 "FFFF16". 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 "FFFF16". 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 "FFFF16" by underflow or "000016" 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”.

(3) When the MCU running in low-speed or low power dissipation mode, do not enter WAIT mode with WAIT peripheral function clock stop bit set 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 $\overline{INT0}$ and $\overline{INT1}$, 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 1.36. Absolute maximum ratings

Symbol	Parameter	Condition	Rated value	Unit
Vcc	Supply voltage		- 0.3 to 6.5 (Note 1)	V
AVcc	Analog supply voltage		- 0.3 to 6.5 (Note 1)	V
V_I	Input voltage $\overline{RESET}$, 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 2)	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 3)	mW
T_{opr}	Operating ambient temperature		- 20 to 85 (Note 4)	°C
T_{stg}	Storage temperature		- 40 to 150 (Note 5)	°C

Note 1: Flash memory version: –0.3 to 7 (V) .

Note 2: When writing to flash MCU, CNVss is –0.3 to 13 (V) .

Note 3: Flat package (56P6S-A) is 300 mW.

Note 4: Extended operating temperature version: -40 to 85 °C. When flash memory version is program/erase mode: 25±5 °C.

Note 5: Extended operating temperature version: -65 to 150 °C.

Table 1.37. Recommended operating conditions (Note 1)

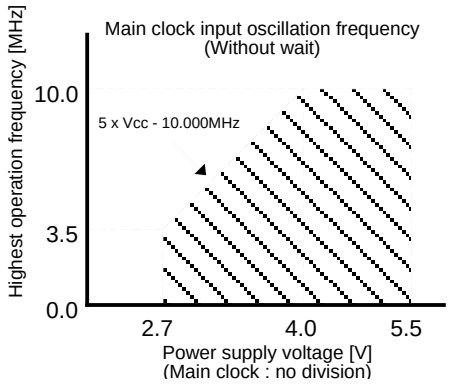
Symbol	Parameter			Standard Min	Standard Typ.	Standard Max.	Unit
V_{CC}	Supply voltage	Mask ROM version		2.7	5.0	5.5	V
		Flash memory version		4.0	5.0	5.5	
AVcc	Analog supply voltage				Vcc		V
Vss	Supply voltage				0		V
AVss	Analog supply voltage				0		V
V_{IH}	HIGH input voltage	$P0_0$ to $P0_7$, $P1_0$ 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, CNV_{SS}		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, CNV_{SS}		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	Mask ROM version	V_{CC}=4.0V to 5.5V	0		10	MHz
			V_{CC}=2.7V to 4.0V	0		5 x V_{CC} - 10.000	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: V_{CC} = 2.7V to 5.5V, Vss = 0V, Ta = – 20 to 85°C (Extended operating temperature version:– 40 to 85°C). Flash version: V_{CC} = 4.0V to 5.5V, Vss = 0V, Ta = – 20 to 85°C (Extended operating temperature version:– 40 to 85°C.)

Note 2: The average output current is an average value measured over 100ms.

Note 3: 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.

Note 4: Relationship between main clock oscillation frequency and supply voltage.



Electrical characteristics (Vcc = 5V)

V_{CC} = 5V

Table 1.38. Electrical characteristics (Note1)

Symbol	Parameter		Measuring condition		Standard Min.	Standard Typ.	Standard Max.	Unit
V_{OH}	HIGH output voltage	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ 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,$\overline{KI_0}$ to $\overline{KI_7}$ RxD_0, RxD_1			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 $P6_7$ $P7_0$,$P7_1$, $\overline{RESET}$, 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 $P6_7$, $P7_0$,$P7_1$, $\overline{RESET}$, CNVss	V_I = 0V				-5.0	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		30.0	50.0	167.0	kΩ
R_{XIN}	Feedback resistor	X_{IN}				1.0		MΩ
R_{XCIN}	Feedback resistor	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 When a WAIT instruction is executed (Note 2)		4.0		µA
				Ta=25℃ when clock is stopped			1.0	µA
				Ta=85℃ when clock is stopped			20.0	

Note 1: Unless otherwise noted: V_{CC} = 5V, V_{SS} = 0V at Ta = -20 to 85°C, f(X_{IN}) = 10MHz

(Extended operating temprature version; -40 to 85°C)

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

VCC = 5V

Table 1.39. A-D conversion characteristics (Note)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
–	Resolution		V_{REF} =V_{CC}			10	Bits
–	Absolute accuracy	Sample & hold function not available	V_{REF} =V_{CC} = 5V			±3	LSB
		Sample & hold function available(10bit)	V_{REF} =V_{CC}= 5V			±3	LSB
		Sample & hold function available(8bit)	V_{REF} = V_{CC} = 5V			±2	LSB
R_{LADDER}	Ladder resistance		V_{REF} =V_{CC}	10		40	kohm
t_{CONV}	Conversion time(10bit)			3.3			µs
t_{CONV}	Conversion time(8bit)			2.8			µs
t_{SAMP}	Sampling time			0.3			µs
V_{REF}	Reference voltage			2		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

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

Electrical characteristics (Vcc = 5V)

Vcc = 5V

Timing requirements (referenced to Vcc = 5V, Vss = 0V at Ta = -20 to 85°C(*) unless otherwise specified)

*** Extended operating temprature version; -40 to 85°C**

Table 1.40. 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 1.41. Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0IN input cycle time	100		ns
$t_{w(TAH)}$	TA0IN input HIGH pulse width	40		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	40		ns

Table 1.42. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0IN input cycle time	400		ns
$t_{w(TAH)}$	TA0IN input HIGH pulse width	200		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	200		ns

Table 1.43. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0IN input cycle time	200		ns
$t_{w(TAH)}$	TA0IN input HIGH pulse width	100		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	100		ns

Table 1.44. Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TA0IN input HIGH pulse width	100		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	100		ns

Table 1.45. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TA0OUT input cycle time	2000		ns
$t_{w(UPH)}$	TA0OUT input HIGH pulse width	1000		ns
$t_{w(UPL)}$	TA0OUT input LOW pulse width	1000		ns
$t_{su(UP-TIN)}$	TA0OUT input setup time	400		ns
$t_{h(TIN-UP)}$	TA0OUT input hold time	400		ns

Electrical characteristics (Vcc = 5V)

VCC = 5V

Timing requirements (referenced to Vcc = 5V, Vss = 0V at Ta = -20 to 85°C(*) unless otherwise specified)

*** Extended operating temprature version; -40 to 85°C**

Table 1.46. 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 1.47. 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 1.48. 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 1.49. 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 1.50. 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 1.51. 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

Electrical characteristics (Vcc = 5V)

Vcc = 5V

Timing requirements (referenced to Vcc = 5V, Vss = 0V at Ta = -20 to 85°C(*) unless otherwise specified)

*** Extended operating temprature version; -40 to 85°C**

Table 1.52. 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 1.53. 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

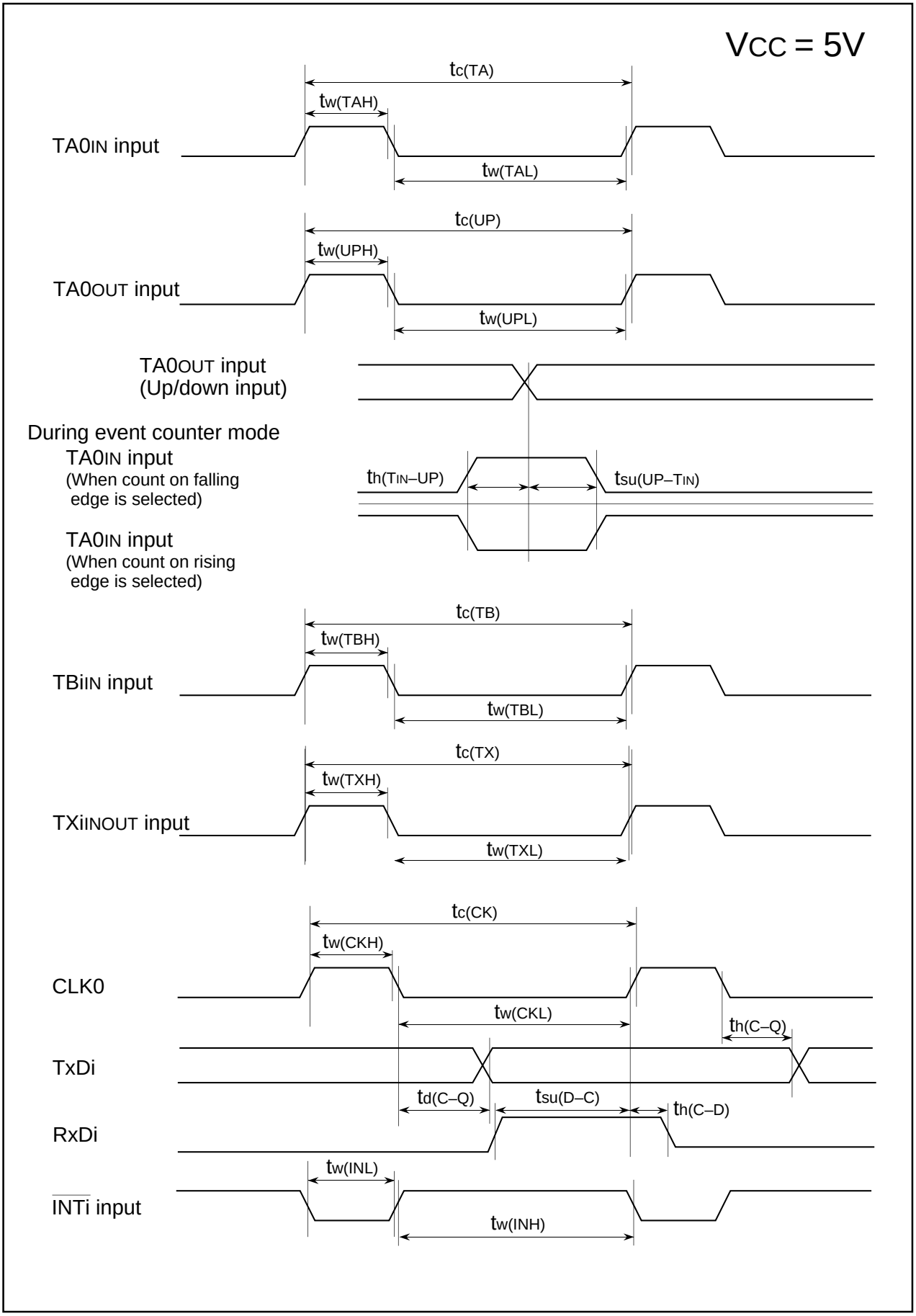
Table 1.54. Serial I/O

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 1.55. External interrupt $\overline{INTi}$ inputs

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

Electrical characteristics (Vcc = 5V)



Vcc = 3V

Table 1.56. Electrical characteristics (Note 1)

Symbol	Parameter		Measuring condition		Standard Min.	Standard Typ.	Standard Max.	Unit
V_{OH}	HIGH output voltage	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ 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	$TA0_{IN}$,$TX0_{INOUT}$,$TX1_{INOUT}$,$TX2_{INOUT}$ $TB0_{IN}$,$TB1_{IN}$ $\overline{INT0}$,$\overline{INT1}$,CLK_0,$\overline{KI0}$ to $\overline{KI7}$ RxD_0, RxD_1			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 $P6_7$, $P7_0$,$P7_1$, $\overline{RESET}$, 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 $P6_7$, $P7_0$,$P7_1$, $\overline{RESET}$, 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})=3.5MHz Square wave, no division		3.5	7.0	mA
				f(X_{CIN})=32kHz Square wave		40.0		µA
				f(X_{CIN})=32kHz When a WAIT instruction is executed Oscillation capacity HIGH (Note 2)		2.8		µA
				f(X_{CIN})=32kHz When a WAIT instruction is executed Oscillation capacity LOW (Note 2)		0.9		µA
				Ta=25℃ when clock is stopped			1.0	µA
				Ta=85℃ when clock is stopped			20.0	

Note 1: Unless otherwise noted: V_{CC} = 3V, V_{SS} = 0V at Ta = -20 to 85°C, f(X_{IN}) = 3.5MHz
(Extended operating temprature version; -40 to 85°C)

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

Vcc = 3V

Table 1.57. A-D conversion characteristics (Note)

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	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

Note : Unless otherwise noted: V_{CC} =AV_{CC} = V_{REF} =3V, V_{SS} =AV_{SS} = 0V at Ta = 25^{o}C, f(X_{IN}) = 3.5MHz.

Vcc = 3V

Timing requirements (referenced to Vcc = 3V, Vss = 0V at Ta = -20 to 85°C(*) unless otherwise specified)

*** Extended operating temprature version; -40 to 85°C**

Table 1.58. External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	286		ns
$t_{w(H)}$	External clock input HIGH pulse width	120		ns
$t_{w(L)}$	External clock input LOW pulse width	120		ns
t_r	External clock rise time		18	ns
t_f	External clock fall time		18	ns

Table 1.59. Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0IN input cycle time	300		ns
$t_{w(TAH)}$	TA0IN input HIGH pulse width	120		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	120		ns

Table 1.60. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0IN input cycle time	1200		ns
$t_{w(TAH)}$	TA0IN input HIGH pulse width	600		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	600		ns

Table 1.61. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0IN input cycle time	600		ns
$t_{w(TAH)}$	TA0IN input HIGH pulse width	300		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	300		ns

Table 1.62. Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TA0IN input HIGH pulse width	300		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	300		ns

Table 1.63. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TA0OUT input cycle time	6000		ns
$t_{w(UPH)}$	TA0OUT input HIGH pulse width	3000		ns
$t_{w(UPL)}$	TA0OUT input LOW pulse width	3000		ns
$t_{su(UP-TIN)}$	TA0OUT input setup time	1200		ns
$t_{h(TIN-UP)}$	TA0OUT input hold time	1200		ns

VCC = 3V

Timing requirements (referenced to Vcc = 3V, Vss = 0V at Ta = -20 to 85°C(*) unless otherwise specified)

*** Extended operating temprature version; -40 to 85°C**

Table 1.64. 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)	300		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on one edge)	120		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on one edge)	120		ns
$t_{c(TB)}$	TBiIN input cycle time (counted on both edges)	600		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on both edges)	320		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	320		ns

Table 1.65. Timer B input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	1200		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	600		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	600		ns

Table 1.66. Timer B input (pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	1200		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	600		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	600		ns

Table 1.67. Timer X input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	300		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	120		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	120		ns

Table 1.68. Timer X input (gate input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	1200		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	600		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	600		ns

Table 1.69. Timer X input (external trigger input in one-shot timer 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

Electrical characteristics (Vcc = 3V)

VCC = 3V

Timing requirements (referenced to Vcc = 3V, Vss = 0V at Ta = -20 to 85°C(*) unless otherwise specified)

*** Extended operating temprature version; -40 to 85°C**

Table 1.70. Timer X input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	1200		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	600		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	600		ns

Table 1.71. Timer X input (pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	1200		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	600		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	600		ns

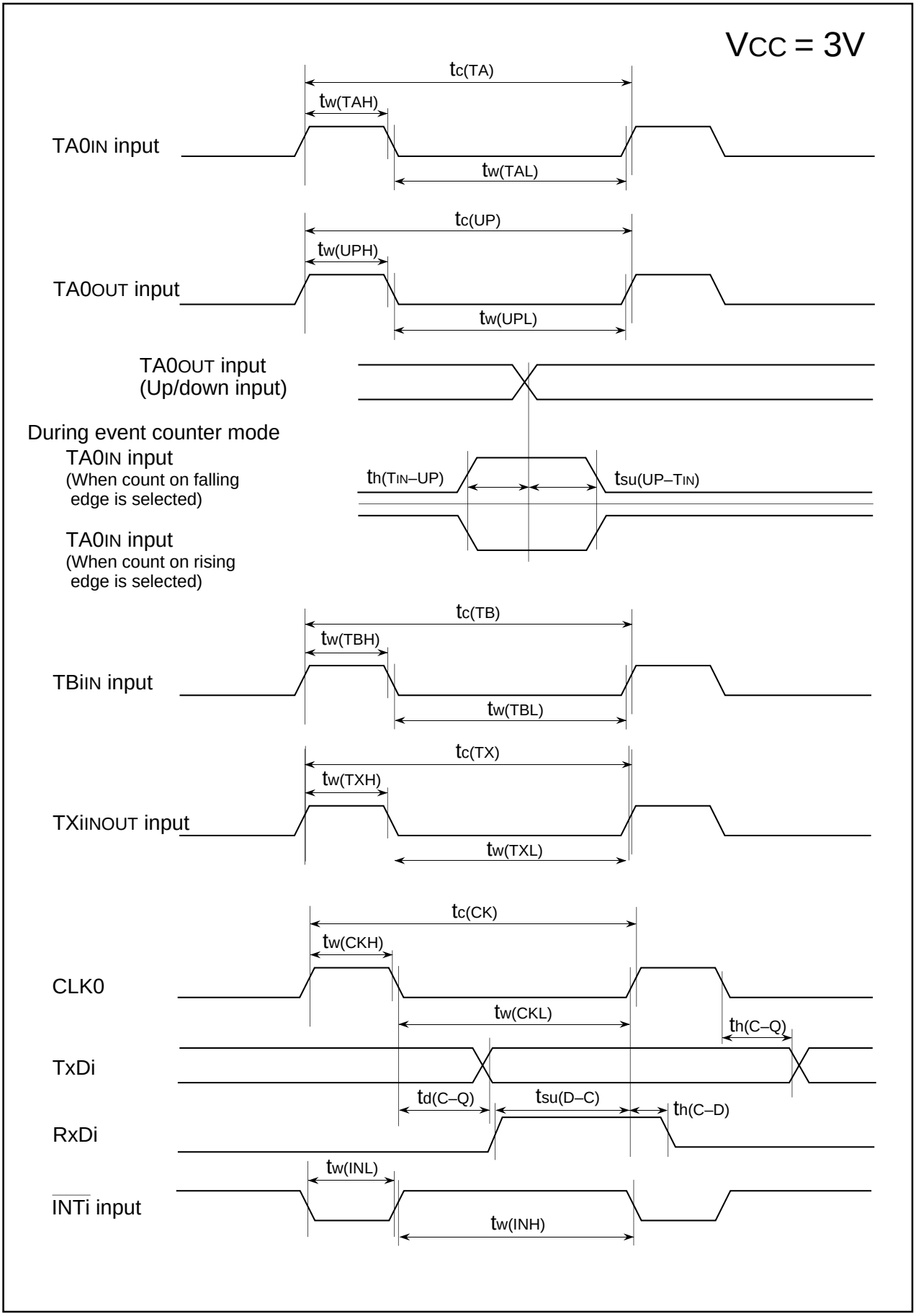
Table 1.72. Serial I/O

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 1.73. External interrupt $\overline{INTi}$ inputs

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

Electrical characteristics (Vcc = 3V)



Outline Performance

Table 1.74 shows the outline performance of the M30201 (flash memory version).

Table 1.74. Outline Performance of the M30201 (flash memory version)

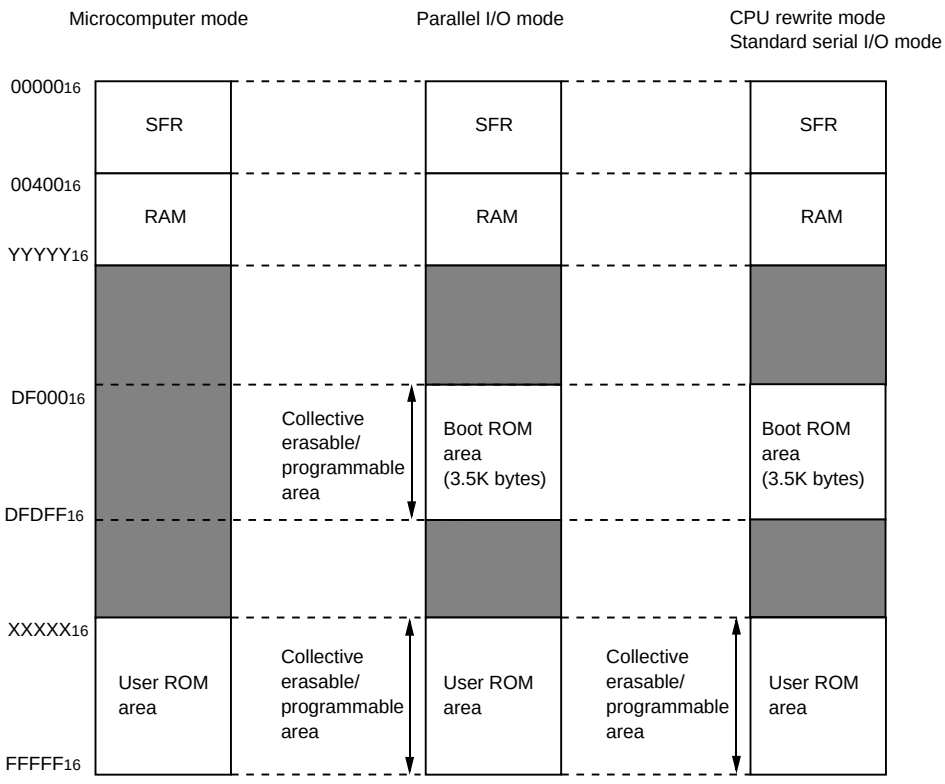
Item		Performance
Power supply voltage		4.0V to 5.5 V ($f(X_{IN})$=10MHz)
Program/erase voltage		V_{PP}=12V ± 5% ($f(X_{IN})$=10MHz, Ta=25±5°C)
		V_{CC}=5V ± 10% ($f(X_{IN})$=10MHz, Ta=25±5°C)
Flash memory operation mode		Three modes (parallel I/O, standard serial I/O, CPU rewrite)
Erase block division	User ROM area	See Figure 1.96
	Boot ROM area	One division (3.5 Kbytes) (Note)
Program method		In units of byte
Erase method		Collective erase
Program/erase control method		Program/erase control by software command
Number of commands		6 commands
Program/erase count		100 times
ROM code protect		Parallel I/O mode is supported.

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

Flash Memory

The M30201 (flash memory version) contains the NOR type of flash memory that requires a high-voltage V_{PP} power supply for program/erase operations, in addition to the V_{CC} power supply for device operation. For this flash memory, three flash memory modes are available in which to read, program, and erase: parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and a CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU). Each mode is detailed in the pages to follow.

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



Note 1: In CPU rewrite and standard serial I/O modes, the user ROM is the only erasable/programmable area.
Note 2: In parallel I/O mode, the area to be erased/programmed can be selected by the address A17 input. The user ROM area is selected when this address input is high and the boot ROM area is selected when this address input is low.

Type No.	XXXXX₁₆	YYYYY₁₆
M30201F6	F4000₁₆	00BFF₁₆

Figure 1.96. Block diagram of flash memory version

CPU Rewrite Mode

In CPU rewrite mode, the on-chip flash memory can be operated on (read, program, or erase) under control of the Central Processing Unit (CPU). In CPU rewrite mode, the flash memory can be operated on by reading or writing to the flash memory control register and flash command register. Figure 1.97, Figure 1.98 show the flash memory control register, and flash command register respectively.

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

In CPU rewrite mode, only the user ROM area shown in Figure 1.96 can be rewritten; the boot ROM area cannot be rewritten. Make sure the program and block commands are issued for only the user ROM area.

The control program for CPU rewrite mode can be stored in either user ROM or boot ROM area. In the CPU rewrite mode, because the flash memory cannot be read from the CPU, the rewrite control program must be transferred to internal RAM before it can be executed.

Flash memory control register 0

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

Symbol: FCON0 — Address: 03B4$_{16}$ — When reset: 00100000$_2$

Bit symbol	Bit name	Function	R	W
FCON00	CPU rewrite mode select bit	0: CPU rewrite mode is invalid 1: CPU rewrite mode is valid	O	O
Reserved bit		This bit can not write. The value, if read, turns out to be indeterminate.	—	—
FCON02	CPU rewrite mode monitor flag	0: CPU rewrite mode is invalid 1: CPU rewrite mode is valid	O	—
Reserved bit		Must always be set to "0".	O	O
Reserved bit		Must always be set to "1".	O	O
Nothing is assigned. In an attempt to write this bit, write "0". The value, if read, turns out to be "0".			—	—
Reserved bit		Must always be set to "0".	O	O

Flash memory control register 1

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

Symbol: FCON1 — Address: 03B5$_{16}$ — When reset: XXXXXX00$_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to "0".	O	O
Nothing is assigned. In an attempt to write these bits, write "0". The value, if read, turns out to be indeterminate.			—	—

Figure 1.97. Flash memory control register

Flash command register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: FCMD — Address: 03B6$_{16}$ — When reset: 00$_{16}$

Function	R	W
Writing of software command <Software command name> <Command code> •Read command "00$_{16}$" •Program command "40$_{16}$" •Program verify command "C0$_{16}$" •Erase command "20$_{16}$" +"20$_{16}$" •Erase verify command "A0$_{16}$" •Reset command "FF$_{16}$" +"FF$_6$"	X	O

Figure 1.98. Flash command register

Microcomputer Mode and Boot Mode

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

See Figure 1.96 for details about the boot ROM area.

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

When the microcomputer is reset by pulling the P52 pin high (VCC), the CNVSS pin high(VPPH), the CPU starts operating using the control program in the boot ROM area. This mode is called the “boot” mode. The control program in the boot ROM area can also be used to rewrite the user ROM area.

CPU rewrite mode operation procedure

The internal flash memory can be operated on to program, read, verify, or erase it while being placed on-board by writing commands from the CPU to the flash memory control register (addresses $03B4_{16}$, $03B5_{16}$) and flash command register (address $03B6_{16}$). Note that when in CPU rewrite mode, the boot ROM area cannot be accessed for program, read, verify, or erase operations. Before this can be accomplished, a CPU write control program must be written into the boot ROM area in parallel input/output mode. The following shows a CPU rewrite mode operation procedure.

<Start procedure (Note 1)>

(1) Apply VPPH to the CNVSS/VPP pin and VCC to the port P52 pin for reset release. Or the user can jump from the user ROM area to the boot ROM area using the JMP instruction and execute the CPU write control program. In this case, set the CPU write mode select bit of the flash memory control register to “1” before applying VPPH to the CNVSS/VPP pin.

(2) After transferring the CPU write control program from the boot ROM area to the internal RAM, jump to this control program in RAM. (The operations described below are controlled by this program.)

(3) Set the CPU rewrite mode select bit to “1”.

(4) Read the CPU rewrite mode monitor flag to see that the CPU rewrite mode is enabled.

(5) Execute operation on the flash memory by writing software commands to the flash command register.

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

<Clearing procedure>

(1) Apply VSS to the CNVSS/VPP pin.

(2) Set the CPU rewrite mode select bit to “0”.

Precautions on CPU Rewrite Mode

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

(1) Operation speed

During erase/program mode, set BCLK to 5 MHz or less by changing the divide ratio.

(2) Instructions inhibited against use

The instructions listed below cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory:
UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

(3) Interrupts inhibited against use

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

Software Commands

Table 1.75 lists the software commands available with the M30201 (flash memory version).

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

The content of each software command is explained below.

Table 1.75. List of Software Commands (CPU Rewrite Mode)

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read	Write	$03B6_{16}$	00_{16}			
Program	Write	$03B6_{16}$	40_{16}	Write	Program address	Program data
Program verify	Write	$03B6_{16}$	$C0_{16}$	Read	Verify address	Verify data
Erase	Write	$03B6_{16}$	20_{16}	Write	$03B6_{16}$	20_{16}
Erase verify	Write	$03B6_{16}$	$A0_{16}$	Read	Verify address	Verify data
Reset	Write	$03B6_{16}$	FF_{16}	Write	$03B6_{16}$	FF_{16}

Read Command (00_{16})

The read mode is entered by writing the command code "00_{16}" to the flash command register in the first bus cycle. When an address to be read is input in one of the bus cycles that follow, the content of the specified address is read out at the data bus (D_0–D_7), 8 bits at a time.

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

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

Program Command (40_{16})

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

During program operation, the watchdog timer remains idle, with the value "$7FFF_{16}$" set in it.

Note 1: The write operation is not completed immediately by writing a program command once. The user must always execute a program-verify command after each program command executed. And if verification fails, the user need to execute the program command repeatedly until the verification passes. See Figure 1.99 for an example of a programming flowchart.

Program-verify command (C0$_{16}$)

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

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

Erase command (20$_{16}$ + 20$_{16}$)

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

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

Note 1: The erase operation is not completed immediately by writing an erase command once. The user must always execute an erase-verify command after each erase command executed. And if verification fails, the user need to execute the erase command repeatedly until the verification passes. See Figure 1.99 for an example of an erase flowchart.

Erase-verify command (A0$_{16}$)

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

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

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

Reset command (FF16 + FF16)

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

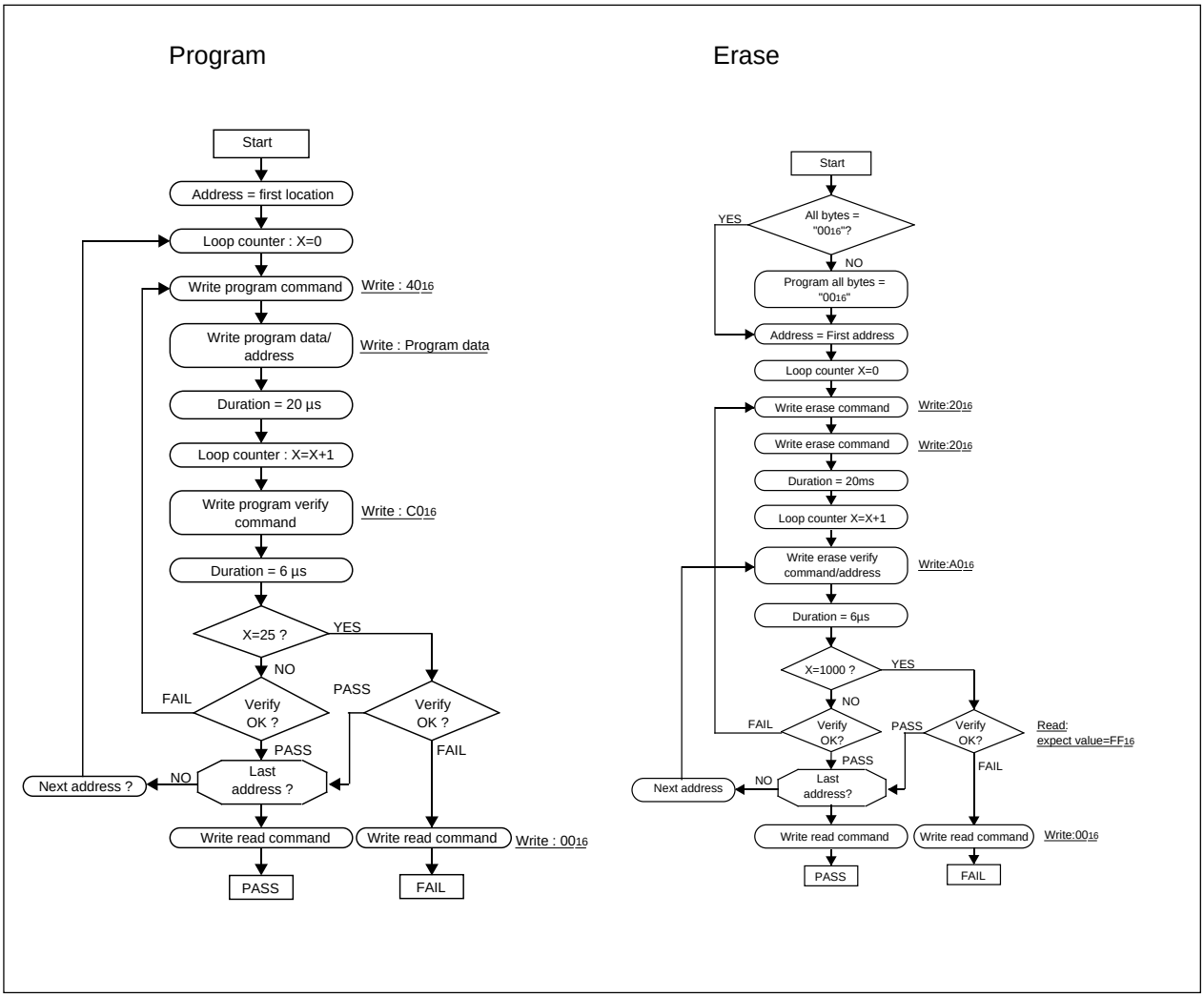


Figure 1.99. Program and erase execution flowchart in the CPU rewrite mode

Description of Pin Function (Flash Memory Parallel I/O Mode)

Pin name	Signal name	I/O	Function
V_{CC},V_{SS}	Power supply input		Apply 5 V ± 10 % to the Vcc pin and 0 V to the Vss pin.
CNV_{SS}	CNV_{SS}	I	Apply 12 V ± 5 % to the CNV_{SS} pin.
$\overline{RESET}$	Reset input	I	Connect this pin to V_{SS}.
X_{IN}	Clock input	I	Connect a ceramic or crystal resonator between the X_{IN} and X_{OUT} pins. When entering an externally derived clock, enter it from X_{IN} and leave X_{OUT} open.
X_{OUT}	Clock output	O	
AV_{CC}, AV_{SS}	Analog power supply input		Connect AV_{SS} to Vss and AVcc to Vcc, respectively.
V_{REF}	Reference voltage input	I	Connect this pin to V_{SS}.
$P0_0$ to $P0_7$	Data I/O D_0 to D_7	I/O	These are data D_0–D_7 input/output pins.
$P1_0$ to $P1_7$	Address input A_8 to A_{15}	I	These are address A_8–A_{15} input pins.
$P3_0$ to $P3_3$	Address input A_4 to A_7	I	These are address A_4–A_7 input pins.
$P3_4$ to $P3_5$	Input port P3	I	Enter low signals to these pins.
$P4_0$	$\overline{WE}$ input	I	This is a $\overline{WE}$ input pin.
$P4_1$	$\overline{OE}$ input	I	This is a $\overline{OE}$ input pin.
$P4_3$	$\overline{CE}$ input	I	This is a $\overline{CE}$ input pin.
$P4_2$, $P4_4$, $P4_5$	Input port P4	I	Enter high signals or low signals to these pins.
$P5_0$	Address input A_{17}	I	This is address A_{17} input pin.
$P5_1$	V_{RFY} input	I	Apply V_{IH} (5 V) to this pin when V_{PP} = V_{PPH} (12 V), or V_{IL} (0 V) when V_{PP} = V_{PPL} (5 V).
$P5_2$	Input port P5	I	Enter low signal to this pin.
$P5_3$, $P5_4$	Input port P5	I	Enter high signals or low signals to these pins.
$P6_0$ to $P6_3$	Address input A_0 to A_3	I	These are address A_0–A_3 input pins.
$P6_4$ to $P6_7$	Input port P6	I	Enter high signals or low signals to these pins.
$P7_0$ to $P7_1$	Input port P7	I	Enter high signals or low signals to these pins.

Parallel I/O Mode

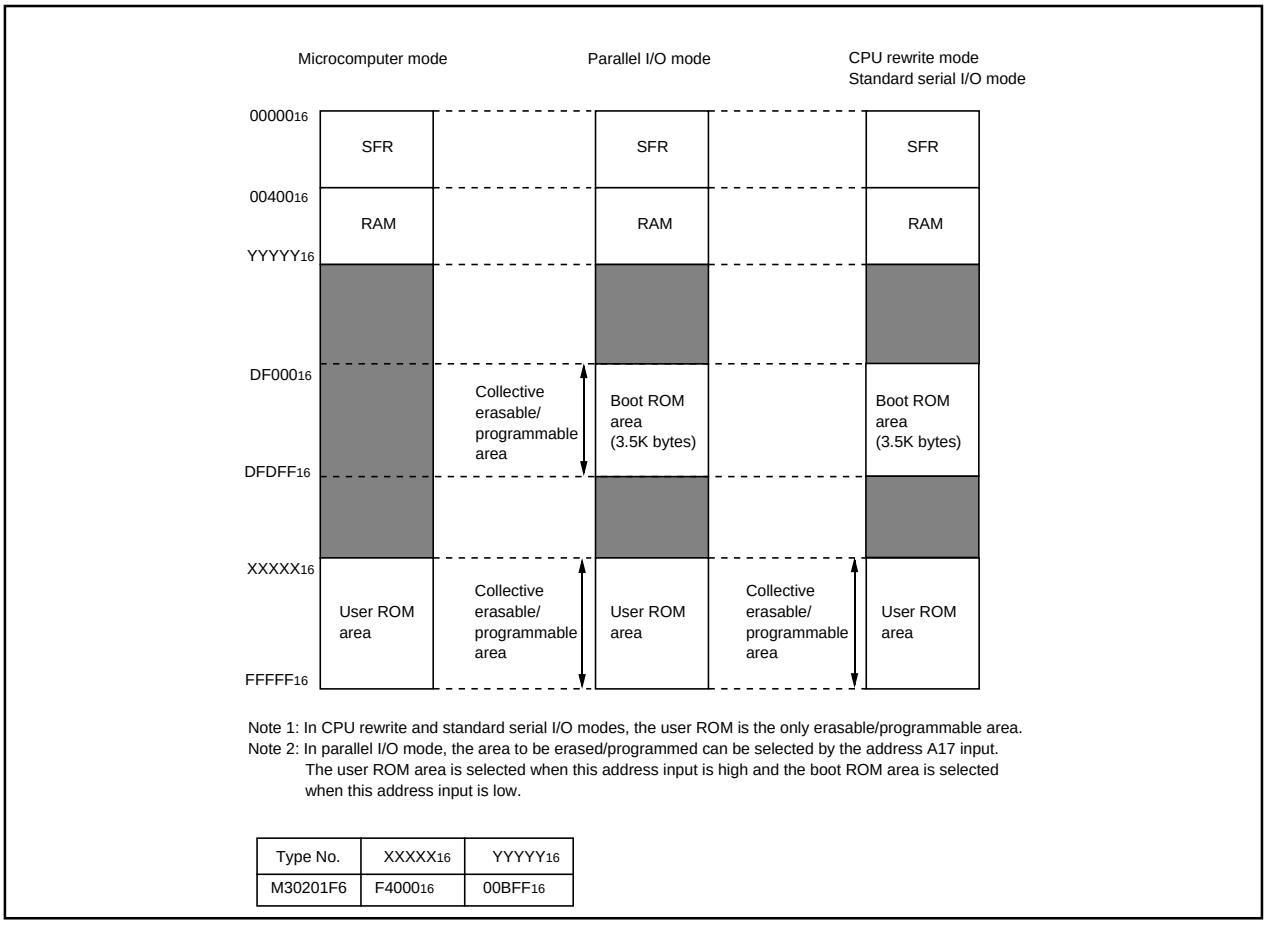
The parallel I/O mode is entered by making connections shown in Figures 1.101 and 1.102 and then turning the V_{PPH} power supply on. In this mode, the M30201 (flash memory version) operates in a manner similar to the NOR flash memory M5M28F101 from Mitsubishi. Note, however, that there are some differences with regard to the functions not available with the microcomputer (function of read device identification code) and matters related to memory capacity.

Table 1.76 shows pin relationship between the M30201 and M5M28F101 in parallel I/O mode.

Table 1.76. Pin relationship in parallel I/O mode

	M30201(flash memory version)	M5M28F101
V_{CC}	V_{CC}	V_{CC}
V_{SS}	V_{SS}	V_{SS}
Address input	$P6_0$ to $P6_3$, $P3_0$ to $P3_3$, $P1_0$ to $P1_7$, $P5_0$	A_0 to A_{15}, A_{17}
Data I/O	$P0_0$ to $P0_7$	D_0 to D_7
$\overline{OE}$ input	$P4_1$	$\overline{OE}$
$\overline{CE}$ input	$P4_3$	$\overline{CE}$
$\overline{WE}$ input	$P4_0$	$\overline{WE}$
V_{RFY} input (Note)	$P5_1$	———

Note: The V_{RFY} input only selects read-only or read/write mode, and does not have any pin associated with it on the M5M28F101.



Note 1: In CPU rewrite and standard serial I/O modes, the user ROM is the only erasable/programmable area.
Note 2: In parallel I/O mode, the area to be erased/programmed can be selected by the address A17 input. The user ROM area is selected when this address input is high and the boot ROM area is selected when this address input is low.

Type No.	XXXXX₁₆	YYYYY₁₆
M30201F6	F4000₁₆	00BFF₁₆

Figure 1.100. Block diagram of flash memory version

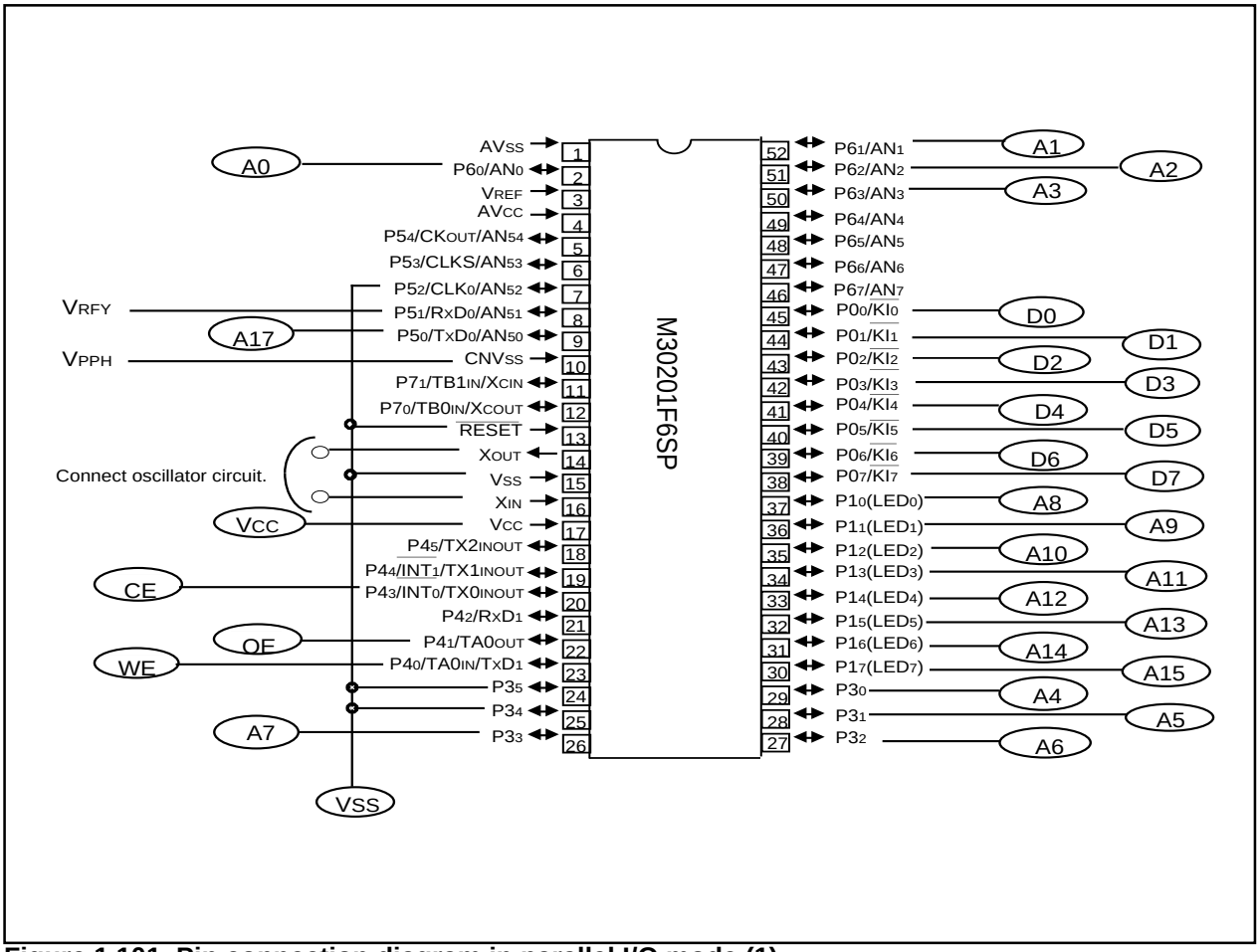


Figure 1.101. Pin connection diagram in parallel I/O mode (1)

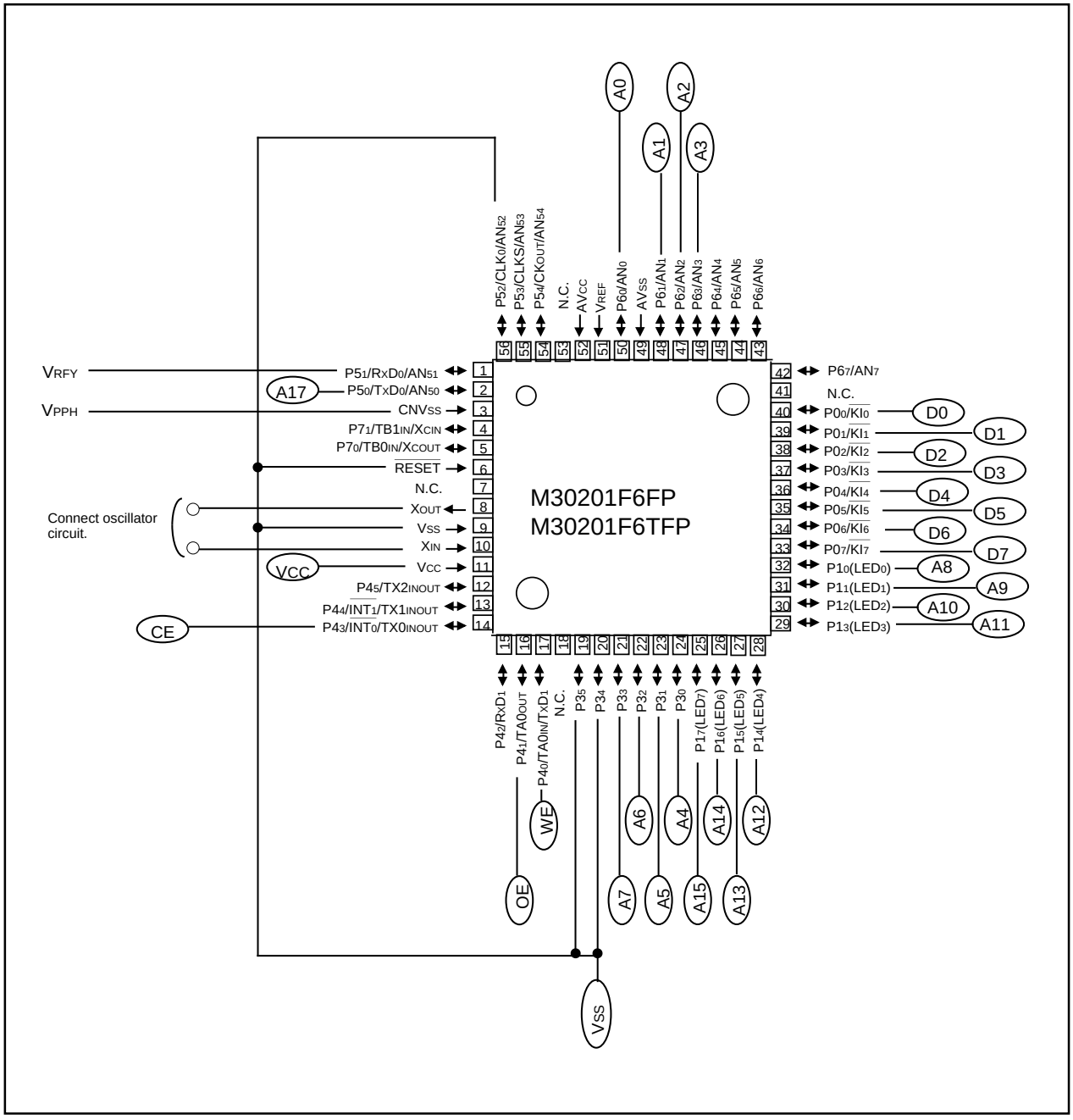


Figure 1.102. Pin connection diagram in parallel I/O mode (2)

User ROM and Boot ROM Areas

In parallel I/O mode, the user ROM and boot ROM areas shown in Figure 1.100 can be rewritten.
In the boot ROM area, an erase block operation is applied to only one 3.5 K byte block. The boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the Mitsubishi factory. Therefore, using the device in standard serial input/output mode, the user does not need to write to the boot ROM area.

Functional Outline (Parallel I/O Mode)

In parallel I/O mode, bus operation modes—Read, Output Disable, Standby, and Write—are selected by the status of the $\overline{CE}$, $\overline{OE}$, $\overline{WE}$, V_{RFY}, and CNV_{SS} input pins.
The contents of erase, program, and other operations are selected by writing a software command. The data in memory can only be read out by a read after software command input.
Program and erase operations are controlled using software commands.

Table 1.77. Relationship between control signals and bus operation modes

Mode \ Pin name		$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	V_{RFY}	V_{PP}	D_0 to D_7
Read only	Read	V_{IL}	V_{IL}	V_{IH}	V_{IL}	V_{PPH}	Data output
	Output disabled	V_{IL}	V_{IH}	V_{IH}	V_{IL}	V_{PPH}	Hi-Z
	Stand by	V_{IH}	X	X	V_{IL}	V_{PPH}	Hi-Z
Read/ Write	Read	V_{IL}	V_{IL}	V_{IH}	V_{IH}	V_{PPH}	Data output
	Output disabled	V_{IL}	V_{IH}	V_{IH}	V_{IH}	V_{PPH}	Hi-Z
	Stand by	V_{IH}	X	X	V_{IH}	V_{PPH}	Hi-Z
	Write	V_{IL}	V_{IH}	V_{IL}	V_{IH}	V_{PPH}	Data input

Note: X can be V_{IL} or V_{IH}.

The following explains about bus operation modes, software commands, and status register.

Bus Operation Modes

Read-only mode is entered by applying V_{PPH} to the CNV_{SS} pin and a low voltage to the V_{RFY} pin. Read-only mode has three states: Read, Output Disable, and Standby which are selected by setting the $\overline{CE}$, $\overline{OE}$, and $\overline{WE}$ pins high or low.

Read-write mode is entered by applying V_{PPH} to the CNV_{SS} pin and a high voltage to the V_{RFY} pin. Read-write mode has four states: Read, Output Disable, Standby, and Write which are selected by setting the $\overline{CE}$, $\overline{OE}$, and $\overline{WE}$ pins high or low.

Read

The Read mode is entered by pulling the $\overline{WE}$ pin high when the $\overline{CE}$ and $\overline{OE}$ pins are low. In Read mode, the data corresponding to each software command entered is output from the data I/O pins D_0–D_7.

Output Disable

The Output Disable mode is entered by pulling the $\overline{CE}$ pin low and the $\overline{WE}$ and $\overline{OE}$ pins high. Also, the data I/O pins are placed in the high-impedance state.

Standby

The Standby mode is entered by driving the $\overline{CE}$ pin high. Also, the data I/O pins are placed in the high-impedance state.

Write

The Write mode is entered by applying V_{PPH} to the CNV_{SS} pin and a high voltage to the V_{RFY} pin and then pulling the $\overline{WE}$ pin low when the $\overline{CE}$ pin is low and $\overline{OE}$ pin is high. In this mode, the device accepts the software commands or write data entered from the data I/O pins. A program, erase, or some other operation is initiated depending on the content of the software command entered here. The input data such as address is latched at the falling edge of $\overline{WE}$ pin. The input data such as software command is latched at the rising edge of $\overline{WE}$ pin.

Software Commands

Table 1.78 lists the software commands available with the M30201 (flash memory version). By entering a software command from the data I/O pins (D_0–D_7) in Write mode, specify the content of the operation, such as erase or program operation, to be performed.

The following explains the content of each software command.

Table 1.78. Software command list (parallel I/O mode)

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read	Write	x	00_{16}			
Program	Write	x	40_{16}	Write	Program address	Program data
Program verify	Write	x	$C0_{16}$	Read	x	Verify data
Erase	Write	x	20_{16}	Write	x	20_{16}
Erase verify	Write	Verify address	$A0_{16}$	Read	x	Verify data
Reset	Write	x	FF_{16}	Write	x	FF_{16}

Read Command (00_{16})

The read mode is entered by writing the command code "00_{16}" in the first bus cycle. When an address to be read is input in one of the bus cycles that follow, the content of the specified address is read out at the data I/O pins (D_0–D_7).

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

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

Program Command (40_{16})

The program mode is entered by writing the command code "40_{16}" in the first bus cycle. When an address and data to be program is write in the second bus cycle, the flash memory control circuit executes the program operation. The program operation requires approximately 20 μs. Wait for 20 μs or more before the user go to the next processing.

Note 1: The write operation is not completed immediately by writing a program command once. The user must always execute a program-verify command after each program command executed. And if verification fails, the user need to execute the program command repeatedly until the verification passes. See Figure 1.103 for an example of a programming flowchart.

Program-verify command ($C0_{16}$)

The program-verify mode is entered by writing the command code "$C0_{16}$" in the first bus cycle and the verify data is output from the data I/O pins (D_0–D_7) in the second bus cycle.

Erase command (20_{16} + 20_{16})

The flash memory control circuit executes an erase operation by writing command code "20_{16}" in the first bus cycle and the same command code again in the second bus cycle. The erase operation requires approximately 20 ms. Wait for 20 ms or more before the user go to the next processing.
Before this erase command can be performed, all memory locations to be erased must have had data "00_{16}" written to by using the program and program-verify commands.

Note 1: The erase operation is not completed immediately by writing an erase command once. The user must always execute an erase-verify command after each erase command executed. And if verification fails, the user need to execute the erase command repeatedly until the verification passes. See Figure 1.103 for an example of an erase flowchart.

Erase-verify command ($A0_{16}$)

The erase-verify mode is entered by writing the command code "$A0_{16}$" in the first bus cycle and the verify data is output from the data I/O pins (D_0–D_7) in the second bus cycle.

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

Reset command (FF16 + FF16)

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

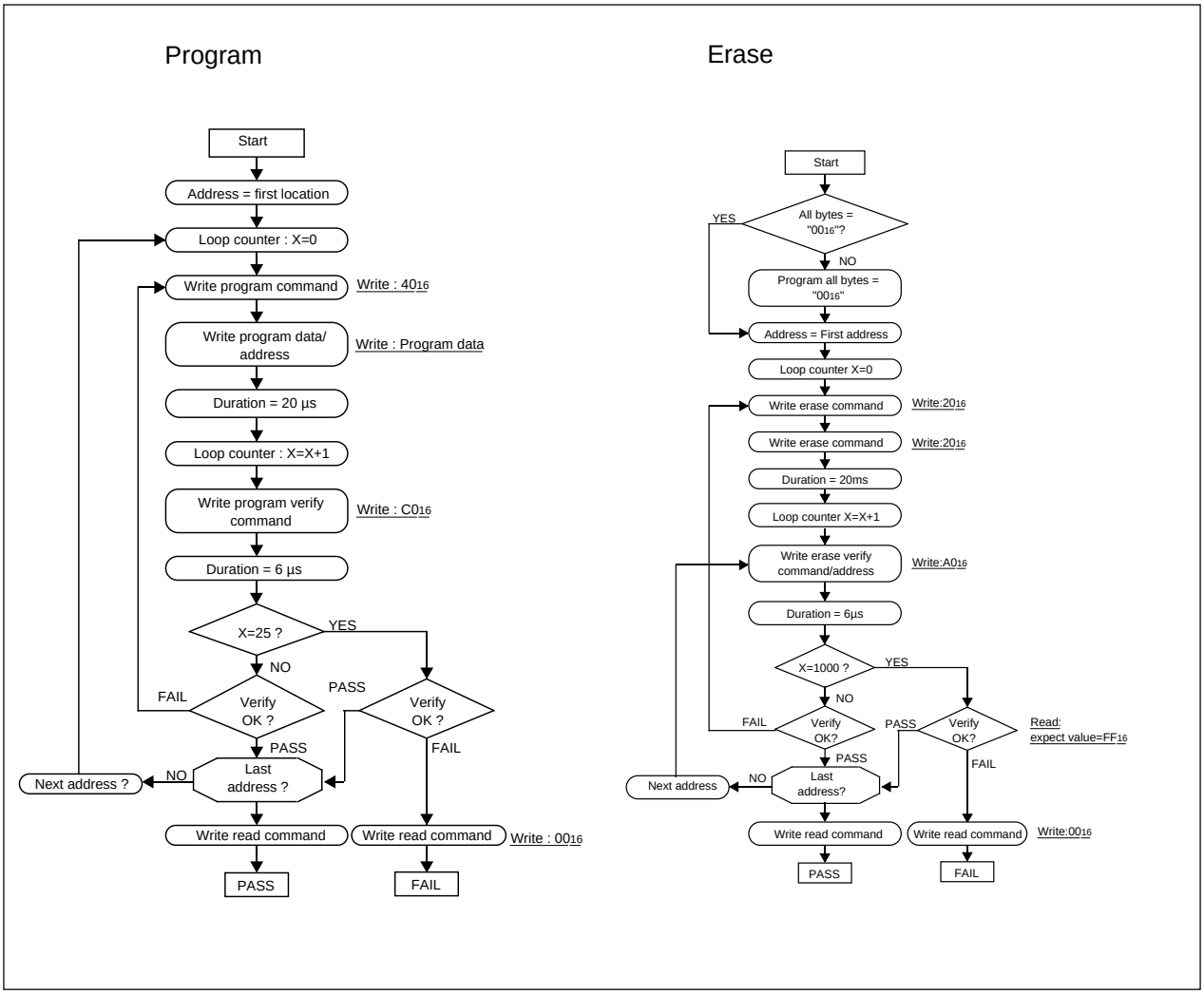


Figure 1.103. Program and erase execution flowchart in the CPU rewrite mode

Protect function

In parallel I/O mode, the internal flash memory has the “protect function” available. This function protects the flash memory contents from being read or rewritten easily.

Depending on the content at the protect control address (FFFFF16) in parallel I/O mode, this function inhibits the flash memory contents against read or modification. The protect control address (FFFFF16) is shown in Figure 1.104. (This address exists in the user ROM area.)

The protect function is enabled by setting one of the two protect set bits to “0”, so that the internal flash memory contents are inhibited against read or modification. The protect function is disabled by setting both of the two protect reset bits to “00”, so that the internal flash memory contents can be read or modified. Once the protect function is set, the user cannot change settings of the protect clear bits while in parallel I/O mode. Settings of the protect reset bits can only be changed in CPU rewrite mode.

Protect control address

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

Symbol: ROMCP
Address: FFFFF16
When shipping: FF16

Bit symbol	Bit name	Function
Reserved bit		Always set to "1".
ROMCR	Protect reset bit	b5 b4 00: Protect removed 01: Protect set bit effective 10: Protect set bit effective 11: Protect set bit effective
ROMCP	Protect set bit	b7 b6 00: Protect enabled 01: Protect enabled 10: Protect enabled 11: Protect disabled

Note 1: When protect is turned on, the flash memory version is protected against readout or modification in parallel I/O mode.
Note 2: The protect reset bits can be used to turn off protect . However, since these bits cannot be changed in parallel I/O mode, they need to be rewritten in CPU rewrite mode.

Figure 1.104. Protect control address

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

Pin	Name	I/O	Description
V_{CC},V_{SS}	Power input		Apply 5V ± 10 % to Vcc pin and 0 V to Vss pin.
CNV_{SS}	CNV_{SS}	I	Mode entry pin. Apply 12V ± 5 % to this pin.
$\overline{RESET}$	Reset input	I	Reset input pin. While reset is "L" level, a 20 cycle or longer clock must be input to XIN pin.
X_{IN}	Clock input	I	Connect a ceramic resonator or crystal oscillator between X_{IN} and X_{OUT} pins. To input an externally generated clock, input it to X_{IN} pin and open X_{OUT} pin.
X_{OUT}	Clock output	O	
AV_{CC}, AV_{SS}	Analog power supply input		Connect AV_{SS} to Vss and AVcc to Vcc, respectively.
V_{REF}	Reference voltage input	I	Enter the reference voltage for AD from this pin.
$P0_0$ to $P0_7$	Input port P0	I	Input "H" or "L" level signal or open.
$P1_0$ to $P1_7$	Input port P1	I	Input "H" or "L" level signal or open.
$P3_0$ to $P3_5$	Input port P3	I	Input "H" or "L" level signal or open.
$P4_0$ to $P4_5$	Input port P4	I	Input "H" or "L" level signal or open.
$P5_4$	Input port P5	I	Input "H" or "L" level signal or open.
$P5_0$	TxD output	O	Serial data output pin.
$P5_1$	RxD input	I	Serial data input pin.
$P5_2$	SCLK input	I	Mode entry pin. Supply "H" level when powering on MCU. When startup is completed this pin serves the serial input clock.
$P5_3$	BUSY	I -> O	This pin sets the type of serial flash programming mode. •An "H" level input (mode 1) sets the mode to clock synchronous. •An "L" level input (mode 2) sets the mode to clock asynchronous. This pin changes to "output" after entry into standard serial I/O mode.
$P6_0$ to $P6_7$	Input port P6	I	Input "H" or "L" level signal or open.
$P7_0$ to $P7_1$	Input port P7	I	Input "H" or "L" level signal or open.

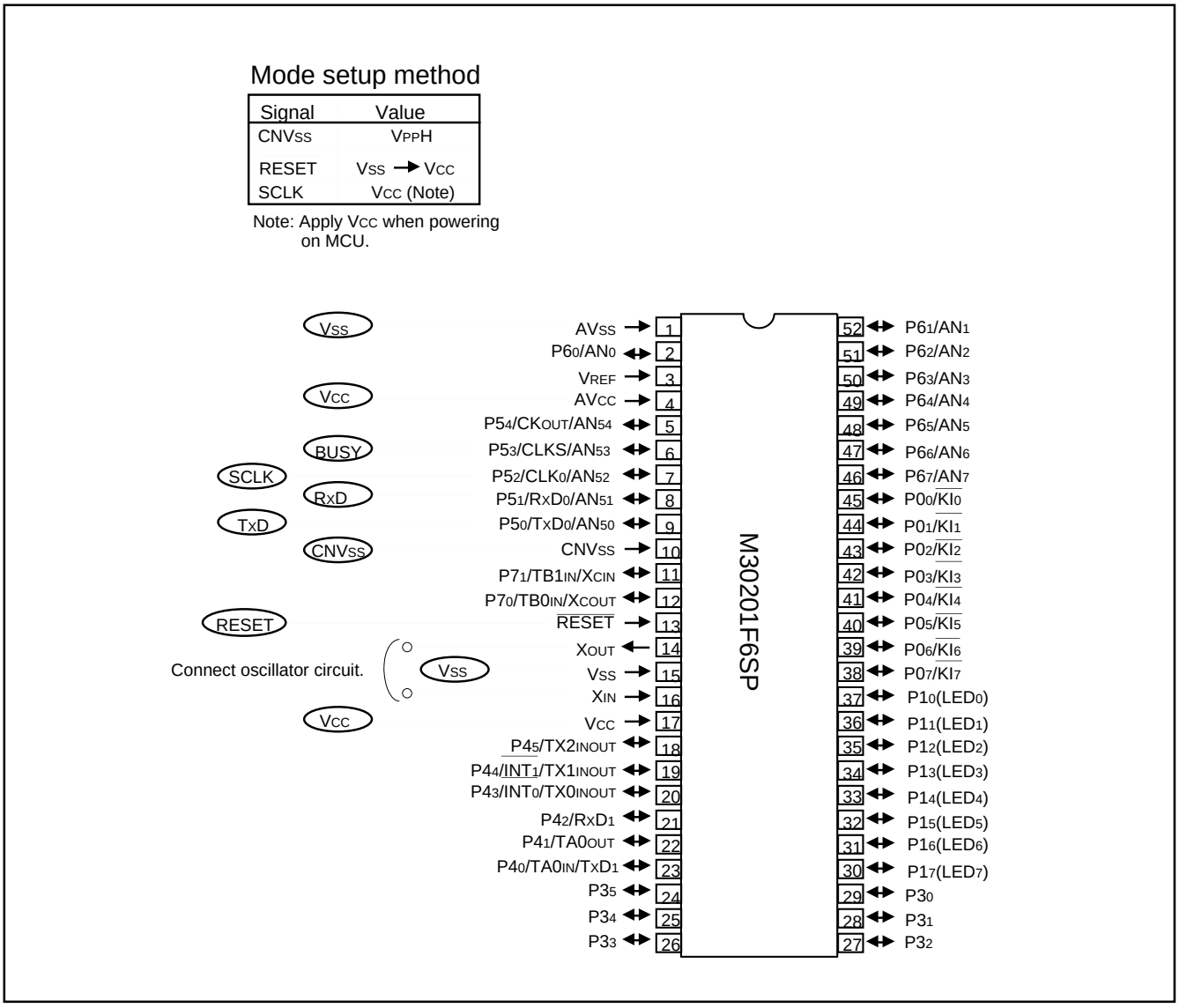


Figure 1.105. Pin connections for standard serial I/O mode (1)

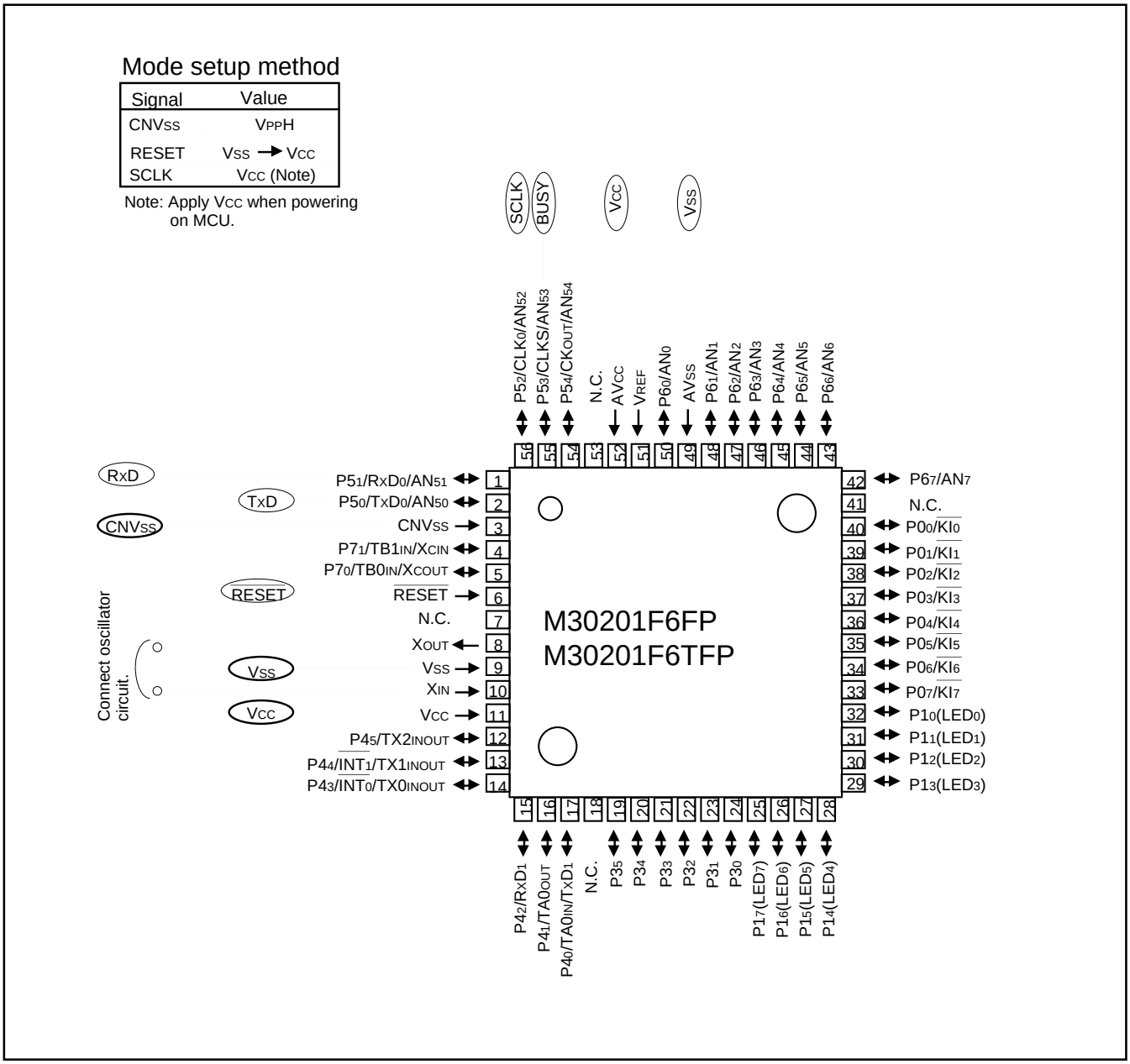


Figure 1.106. Pin connections for serial I/O mode (2)

Standard serial I/O mode

The standard serial I/O mode inputs and outputs the software commands, addresses and data needed to operate (read, program, erase, etc.) the internal flash memory. This I/O is serial. There are actually two standard serial I/O modes: mode 1, which is clock synchronized, and mode 2, which is asynchronized. Both modes require a purpose-specific peripheral unit.

The standard serial I/O mode is different from the parallel I/O mode in that the CPU controls flash memory rewrite (uses the CPU's rewrite mode), rewrite data input and so forth. It is started when the reset is released, which is done when the $P5_2$ (SCLK) pin is "H" level, the CNVss pin "VppH" level. (In the ordinary command mode, set CNVss pin to "L" level.)
This control program is written in the boot ROM area when the product is shipped from Mitsubishi. Accordingly, make note of the fact that the standard serial I/O mode cannot be used if the boot ROM area is rewritten in the parallel I/O mode. Figures 1.105 and 1.106 show the pin connections for the standard serial I/O mode. Serial data I/O uses UART0 and transfers the data serially in 8-bit units. Standard serial I/O switches between mode 1 (clock synchronized) and mode 2 (clock asynchronized) according to the level of $P5_3$ (BUSY) pin when the reset is released.

To use standard serial I/O mode 1 (clock synchronized), set the $P5_3$ (BUSY) pin to "H" level and release the reset. The operation uses the four UART0 pins CLK_0, RxD_0, TxD_0 and $P5_3$ (BUSY). The CLK_0 pin is the transfer clock input pin through which an external transfer clock is input. The TxD_0 pin is for CMOS output. The $P5_3$ (BUSY) pin outputs an "L" level when ready for reception and an "H" level when reception starts.

To use standard serial I/O mode 2 (clock asynchronized), set the $P5_3$ (BUSY) pin to "L" level and release the reset. The operation uses the two UART0 pins RxD_0 and TxD_0.

In the standard serial I/O mode, only the user ROM area indicated in Figure 1.96 can be rewritten. The boot ROM cannot.
In the standard serial I/O mode, a 7-byte ID code is used. When there is data in the flash memory, commands sent from the peripheral unit are not accepted unless the ID code matches.

Overview of standard serial I/O mode 1 (clock synchronized)

In standard serial I/O mode 1, software commands, addresses and data are input and output between the MCU and peripheral units (serial programer, etc.) using clock-synchronized serial I/O (UART0) and $P5_3$ ($\overline{BUSY}$). Standard serial I/O mode 1 is engaged by releasing the reset with the $P5_3$ ($\overline{BUSY}$) pin "H" level.

In reception, software commands, addresses and program data are synchronized with the rise of the transfer clock that is input to the CLK_0 pin, and are then input to the MCU via the RxD_0 pin. In transmission, the read data and status are synchronized with the fall of the transfer clock, and output from the TxD_0 pin.

The TxD_0 pin is for CMOS output. Transfer is in 8-bit units with LSB first.

When busy, such as during transmission, reception, erasing or program execution, the $P5_3$ ($\overline{BUSY}$) pin is "H" level. Accordingly, always start the next transfer after the $P5_3$ ($\overline{BUSY}$) pin is "L" level.

Also, data and status registers in memory can be read after inputting software commands. Status, such as the operating state of the flash memory or whether a program or erase operation ended successfully or not, can be checked by reading the status register. Here following are explained software commands, status registers, etc.

Software Commands

Table 1.79 lists software commands. In the standard serial I/O mode 1, erase operations, programs and reading are controlled by transferring software commands via the RxD_0 pin. Software commands are explained here below.

Table 1.79. Software commands (Standard serial I/O mode 1)

	Control command		2nd byte	3rd byte	4th byte	5th byte	6th byte		When ID is not verificate
1	Page read	FF_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41_{16}	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Erase all unlocked blocks	$A7_{16}$	$D0_{16}$						Not acceptable
4	Read status register	70_{16}	SRD output	SRD1 output					Acceptable
5	Clear status register	50_{16}							Not acceptable
6	Read lockbit status	71_{16}	Address (middle)	Address (high)	Lock bit data output				Not acceptable
7	ID check function	$F5_{16}$	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
8	Download function	FA_{16}	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
9	Version data output function	FB_{16}	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
10	Boot area output function	FC_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable

Note 1: Shading indicates transfer from flash memory microcomputer to peripheral unit. All other data is transferred from the peripheral unit to the flash memory microcomputer.

Note 2: SRD refers to status register data. SRD1 refers to status register 1 data.

Note 3: All commands can be accepted when the flash memory is totally blank.

Page Read Command

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

(1) Transfer the "FF_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first in sync with the rise of the clock.

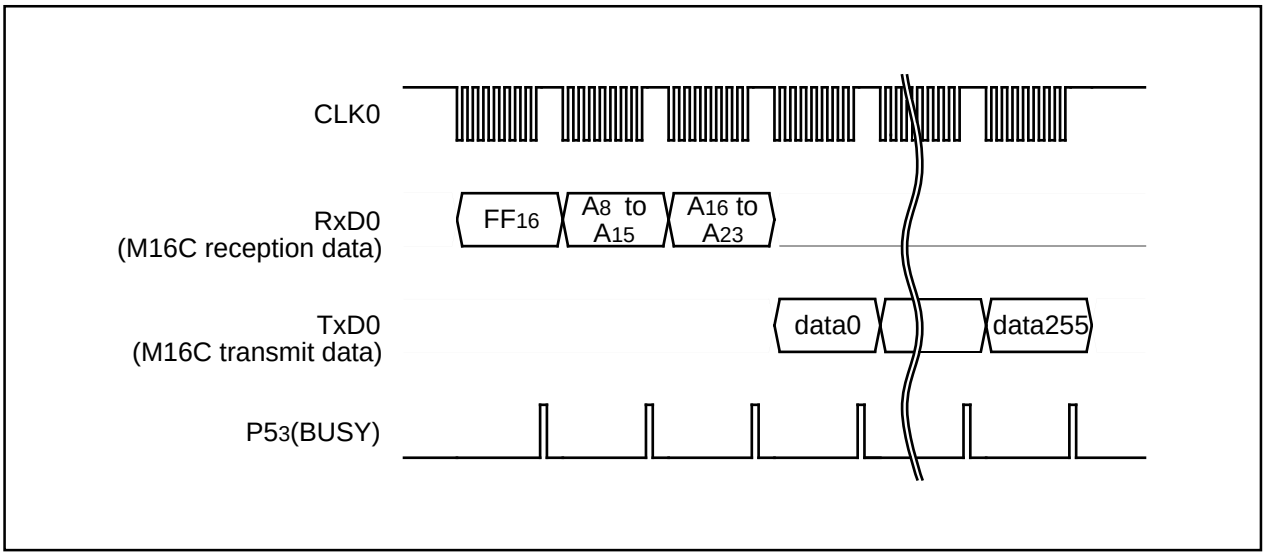


Figure 1.107. Timing for page read

Read Status Register Command

This command reads status information. When the "70_{16}" command code is sent with the 1st byte, the contents of the status register (SRD) specified with the 2nd byte and the contents of status register 1 (SRD1) specified with the 3rd byte are read.

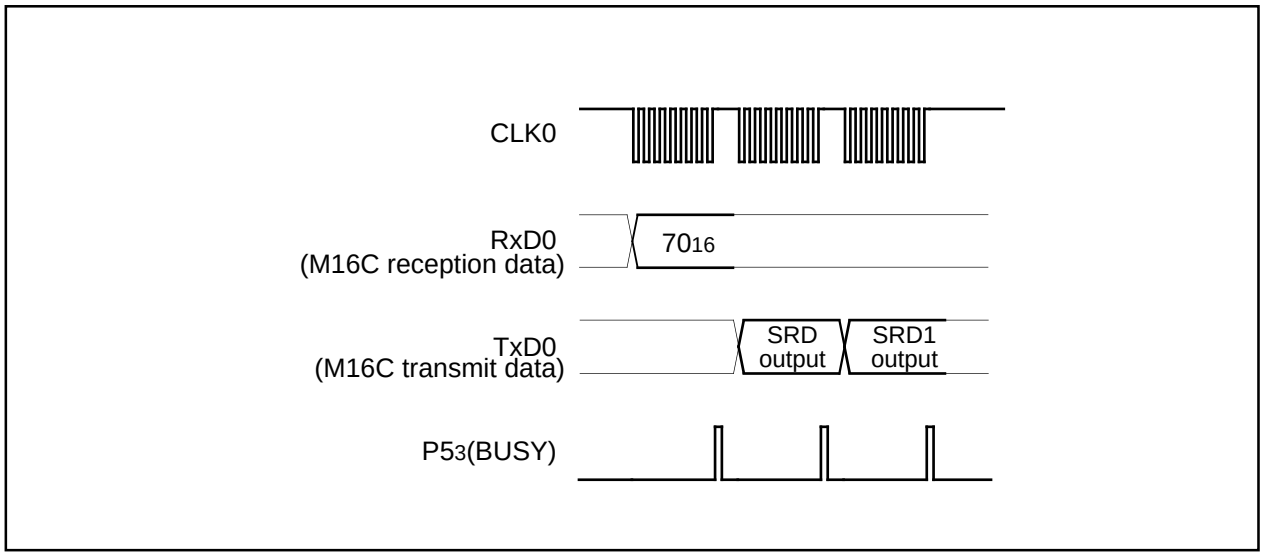


Figure 1.108. Timing for reading the status register

Clear Status Register Command

This command clears the bits (SR3–SR4) which are set when the status register operation ends in error. When the "50_{16}" command code is sent with the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the $P5_3$ (BUSY) signal changes from the "H" to the "L" level.

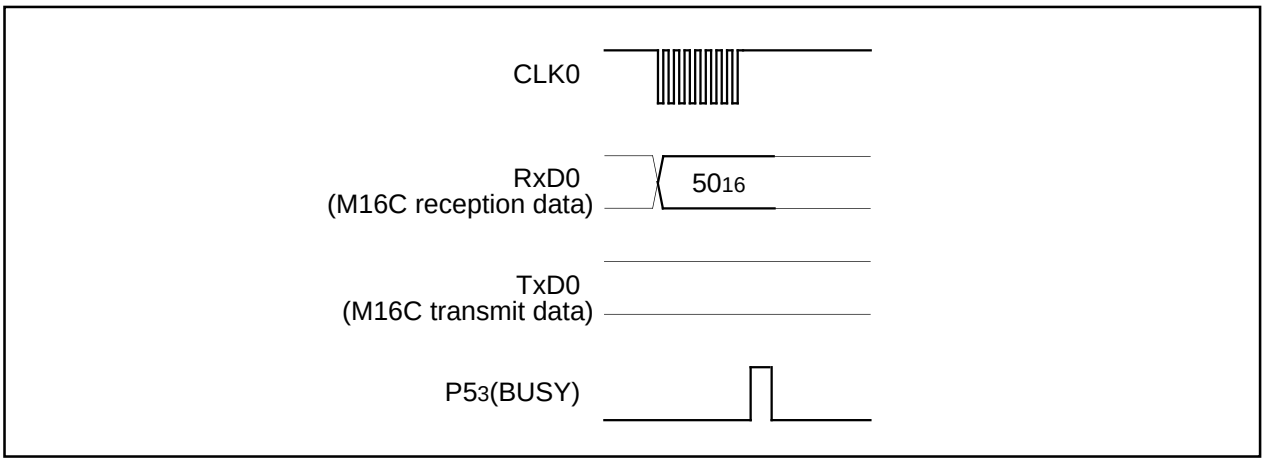


Figure 1.109. Timing for clearing the status register

Page Program Command

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

(1) Transfer the "41_{16}" command code with the 1st byte.
(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.
(3) From the 4th byte onward, as write data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the $P5_3$ (BUSY) signal changes from the "H" to the "L" level. The result of the page program can be known by reading the status register. For more information, see the section on the status register.

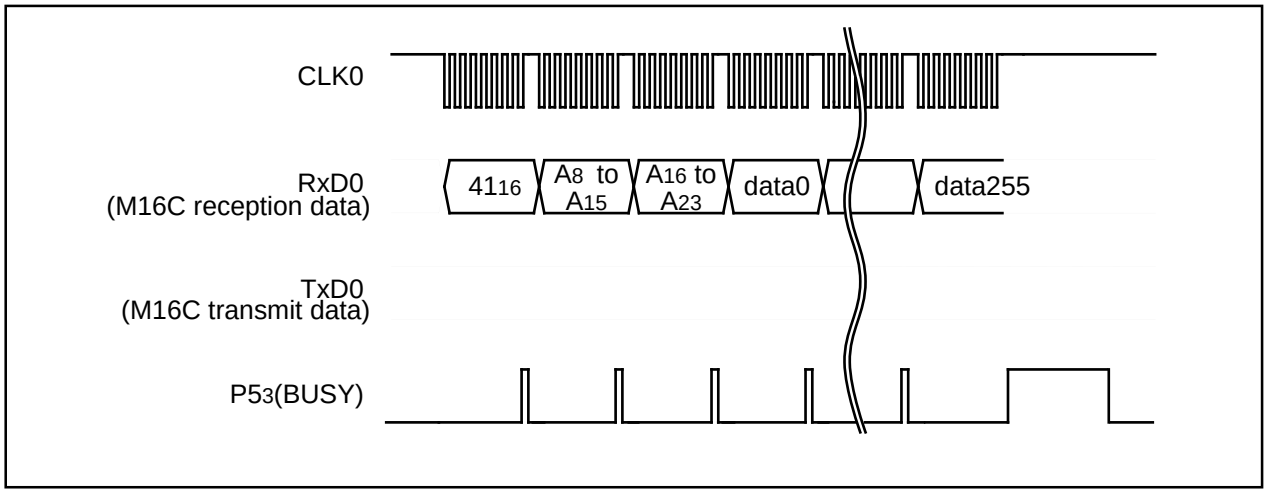


Figure 1.110. Timing for the page program

Erase All Unlocked Blocks Command

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

(1) Transfer the "A7$_{16}$" command code with the 1st byte.

(2) Transfer the verify command code "D0$_{16}$" with the 2nd byte. With the verify command code, the erase operation will start and continue for all blocks in the flash memory.

When block erasing ends, the P5$_3$ (BUSY) signal changes from the "H" to the "L" level. The result of the erase operation can be known by reading the status register.

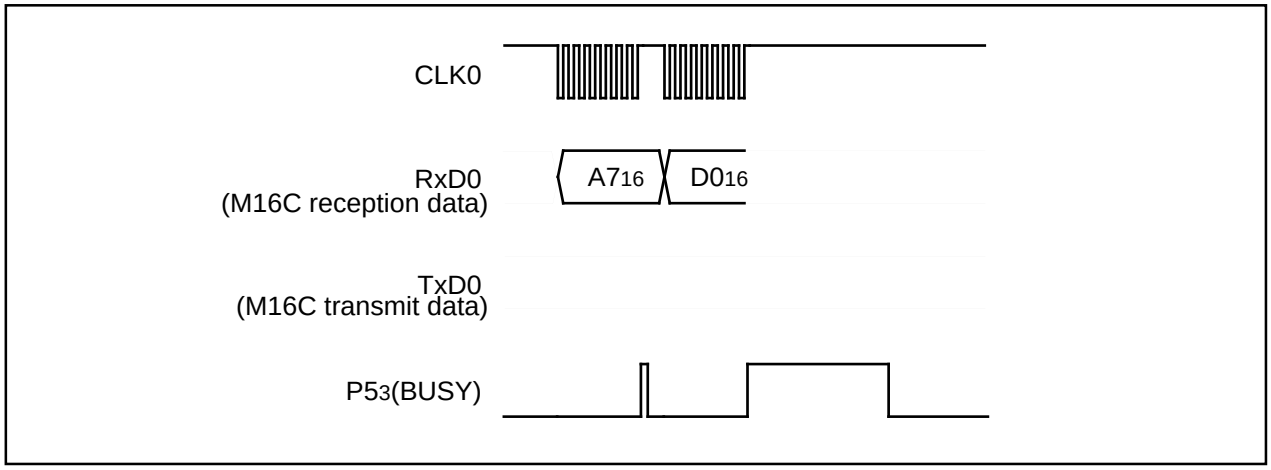


Figure 1.111. Timing for erasing all unlocked blocks

Read Lock Bit Status Command

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

(1) Transfer the "71$_{16}$" command code with the 1st byte.

(2) Transfer addresses A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ with the 2nd and 3rd bytes respectively.

(3) The lock bit data of the specified block is output with the 4th byte. Write the highest address of the specified block for addresses A$_8$ to A$_{23}$.

The M30201 (flash memory version) does not have the lock bit, so the read value is always "1" (block unlock).

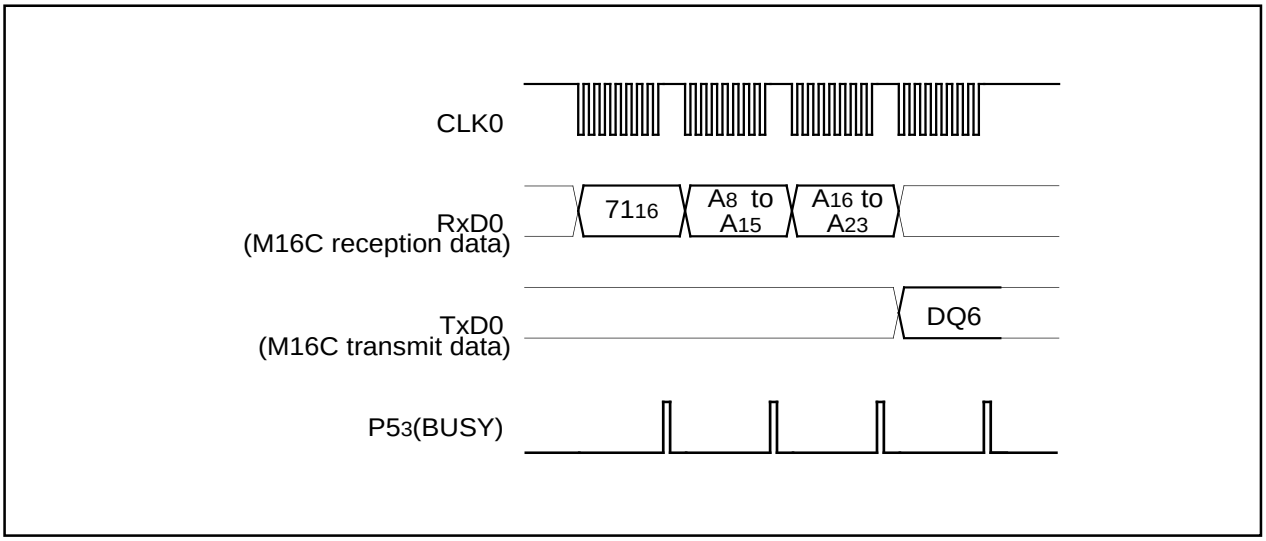


Figure 1.112. Timing for reading lock bit status

Download Command

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

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

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

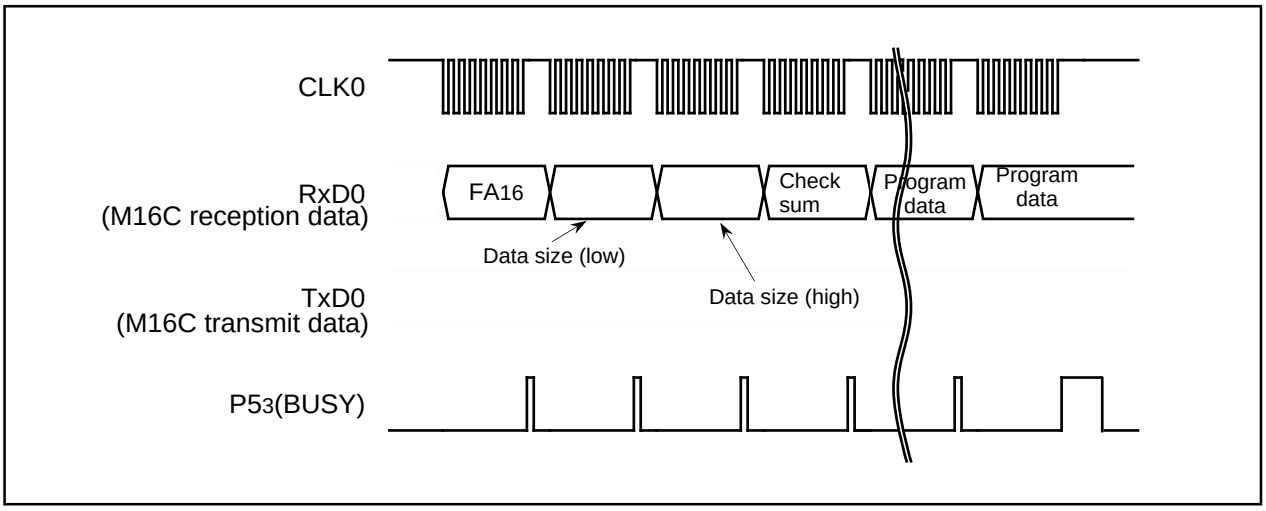


Figure 1.113. Timing for download

Version Information Output Command

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

(1) Transfer the "FB16" command code with the 1st byte.

(2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

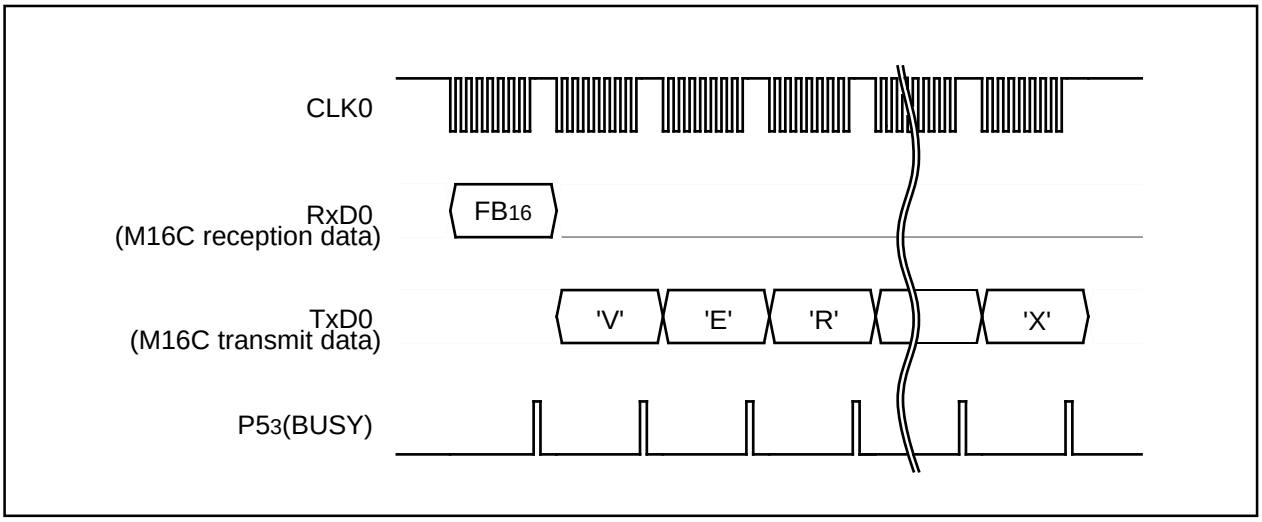


Figure 1.114. Timing for version information output

Boot ROM Area Output Command

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

(1) Transfer the "FC16" command code with the 1st byte.

(2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first, in sync with the fall of the clock.

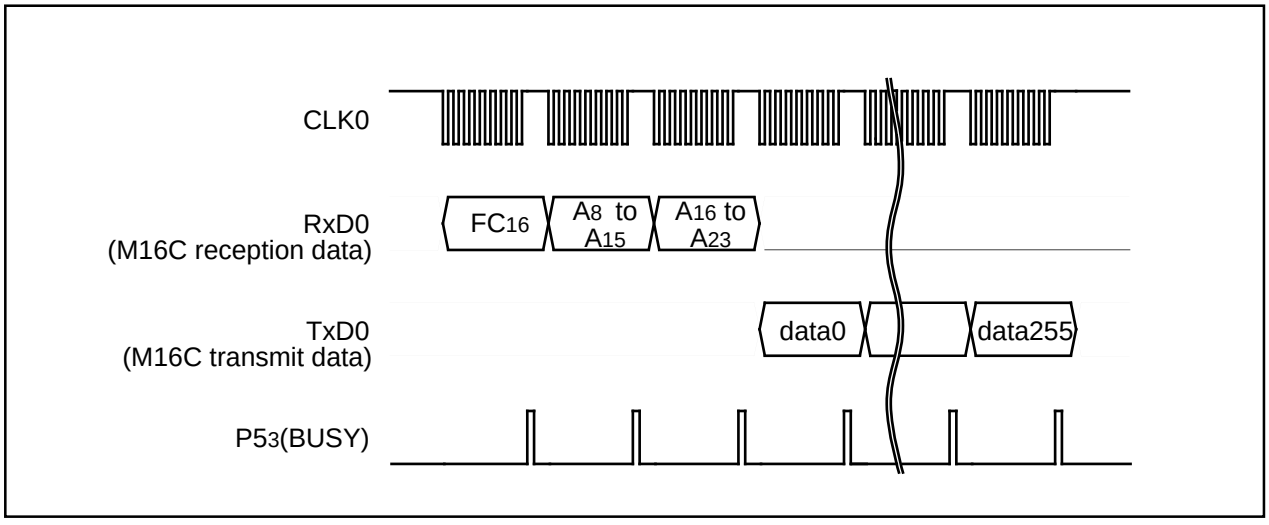


Figure 1.115. Timing for boot ROM area output

ID Check

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

(1) Transfer the "F5$_{16}$" command code with the 1st byte.

(2) Transfer addresses A$_0$ to A$_7$, A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ of the 1st byte of the ID code with the 2nd, 3rd and 4th bytes respectively.

(3) Transfer the number of data sets of the ID code with the 5th byte.

(4) The ID code is sent with the 6th byte onward, starting with the 1st byte of the code.

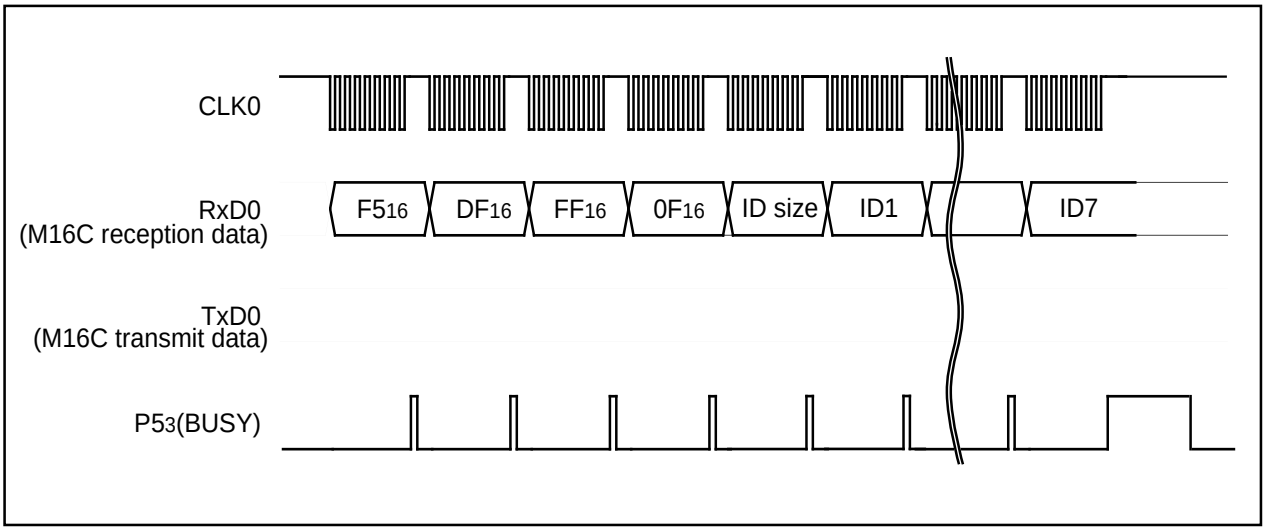


Figure 1.116. Timing for the ID check

ID Code

When the flash memory is not blank, the ID code sent from the peripheral units and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the peripheral units is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses 0FFFDF$_{16}$, 0FFFE3$_{16}$, 0FFFEB$_{16}$, 0FFFEF$_{16}$, 0FFFF3$_{16}$, 0FFFF7$_{16}$ and 0FFFFB$_{16}$. Write a program into the flash memory, which already has the ID code set for these addresses.

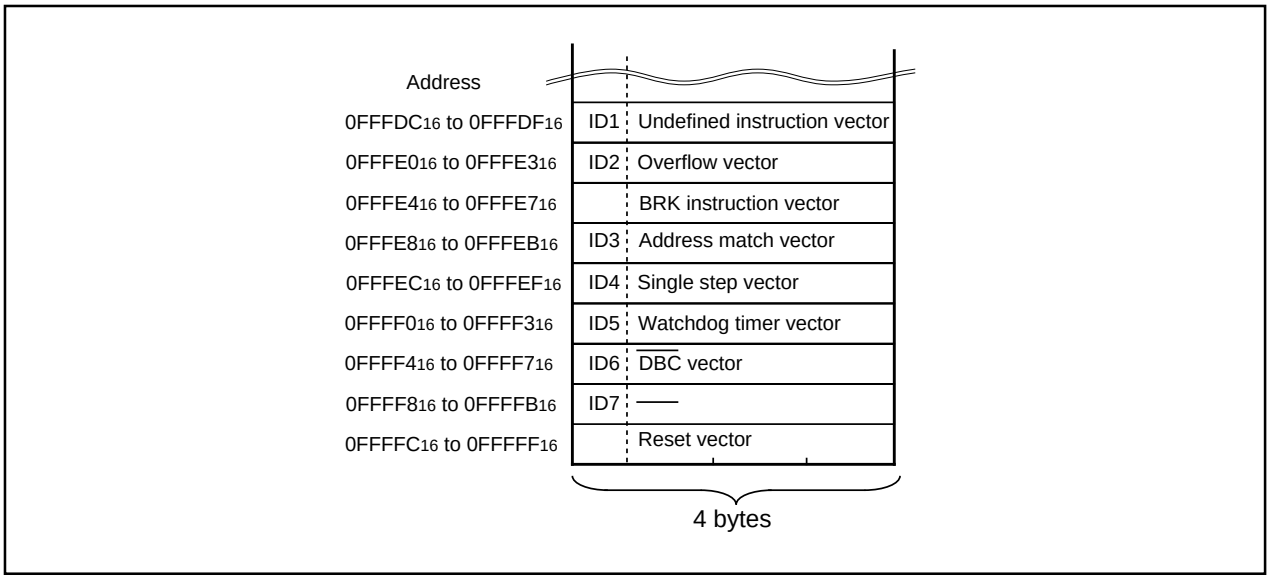


Figure 1.117. ID code storage addresses

Status Register (SRD)

The status register indicates operating status of the flash memory and status such as whether an erase operation or a program ended successfully or in error. It can be read by writing the read status register command (70_{16}). Also, the status register is cleared by writing the clear status register command (50_{16}). Table 1.80 gives the definition of each status register bit. After clearing the reset, the status register outputs "80_{16}".

Table 1.80. Status register (SRD)

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

Status bit (SR7)

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

Erase Status (SR5)

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

Program Status (SR4)

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

Status Register 1 (SRD1)

Status register 1 indicates the status of serial communications, results from ID checks and results from check sum comparisons. It can be read after the SRD by writing the read status register command (70_{16}). Also, status register 1 is cleared by writing the clear status register command (50_{16}).

Table 1.81 gives the definition of each status register 1 bit. "00_{16}" is output when power is turned ON and the flag status is maintained even after the reset.

Table 1.81. Status register 1 (SRD1)

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

Boot Update Completed Bit (SR15)

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

Check Sum Consistency Bit (SR12)

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

ID Check Completed Bits (SR11 and SR10)

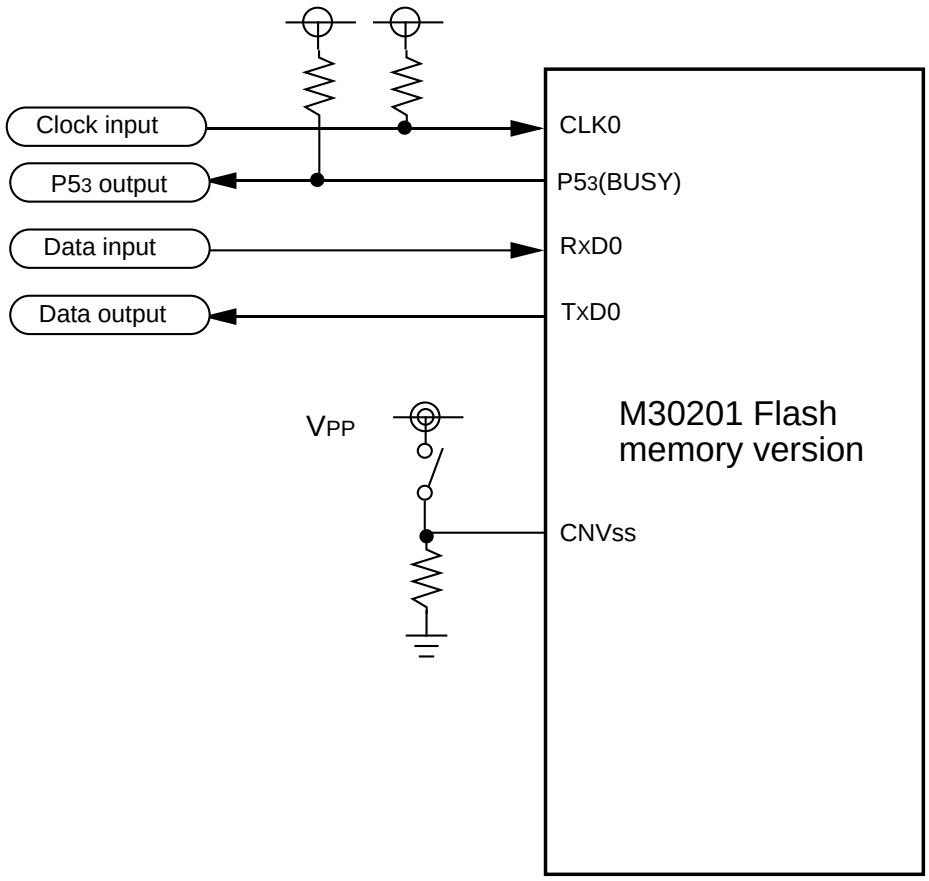
These flags indicate the result of ID checks. Some commands cannot be accepted without an ID check.

Data Reception Time Out (SR9)

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

Example Circuit Application for The Standard Serial I/O Mode 1

The below figure shows a circuit application for the standard serial I/O mode 1. Control pins will vary according to programmer, therefore see the peripheral unit manual for more information.



(1) Control pins and external circuitry will vary according to peripheral unit. For more information, see the peripheral unit manual.
(2) In this example, the microprocessor mode and standard serial I/O mode are switched via a switch.

Figure 1.118. Example circuit application for the standard serial I/O mode 1

Overview of standard serial I/O mode 2 (clock asynchronized)

In standard serial I/O mode 2, software commands, addresses and data are input and output between the MCU and peripheral units (serial programer, etc.) using 2-wire clock-asynchronized serial I/O (UART0). Standard serial I/O mode 2 is engaged by releasing the reset with the $P5_3$ (BUSY) pin "L" level.

The TxD_0 pin is for CMOS output. Data transfer is in 8-bit units with LSB first, 1 stop bit and parity OFF.

After the reset is released, connections can be established at 9,600 bps when initial communications (Figure 1.119) are made with a peripheral unit. However, this requires a main clock with a minimum 2 MHz input oscillation frequency. Baud rate can also be changed from 9,600 bps to 19,200, 38,400 or 57,600 bps by executing software commands. However, communication errors may occur because of the oscillation frequency of the main clock. If errors occur, change the main clock's oscillation frequency and the baud rate.

After executing commands from a peripheral unit that requires time to erase and write data, as with erase and program commands, allow a sufficient time interval or execute the read status command and check how processing ended, before executing the next command.

Data and status registers in memory can be read after transmitting software commands. Status, such as the operating state of the flash memory or whether a program or erase operation ended successfully or not, can be checked by reading the status register. Here following are explained initial communications with peripheral units, how frequency is identified and software commands.

Initial communications with peripheral units

After the reset is released, the bit rate generator is adjusted to 9,600 bps to match the oscillation frequency of the main clock, by sending the code as prescribed by the protocol for initial communications with peripheral units (Figure 1.119).

(1) Transmit "$B0_{16}$" from a peripheral unit. If the oscillation frequency input by the main clock is 10 MHz, the MCU with internal flash memory outputs the "$B0_{16}$" check code. If the oscillation frequency is anything other than 10 MHz, the MCU does not output anything.

(2) Transmit "00_{16}" from a peripheral unit 16 times. (The MCU with internal flash memory sets the bit rate generator so that "00_{16}" can be successfully received.)

(3) The MCU with internal flash memory outputs the "$B0_{16}$" check code and initial communications end successfully *1. Initial communications must be transmitted at a speed of 9,600 bps and a transfer interval of a minimum 15 ms. Also, the baud rate at the end of initial communications is 9,600 bps.

*1. If the peripheral unit cannot receive "$B0_{16}$" successfully, change the oscillation frequency of the main clock.

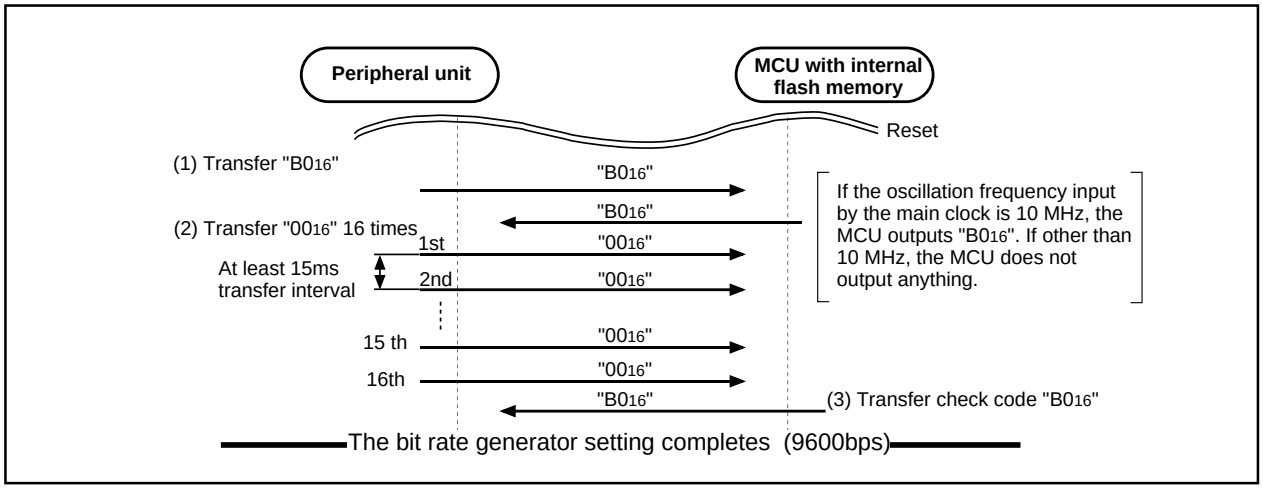


Figure 1.119. Peripheral unit and initial communication

How frequency is identified

When "00$_{16}$" data is received 16 times from a peripheral unit at a baud rate of 9,600 bps, the value of the bit rate generator is set to match the operating frequency (2 - 10 MHz). The highest speed is taken from the first 8 transmissions and the lowest from the last 8. These values are then used to calculate the bit rate generator value for a baud rate of 9,600 bps.

Baud rate cannot be attained with some operating frequencies. Table 1.82 gives the operation frequency and the baud rate that can be attained for.

Table 1.82 Operation frequency and the baud rate

Operation frequency (MHz)	Baud rate 9,600bps	Baud rate 19,200bps	Baud rate 38,400bps	Baud rate 57,600bps
10MHz	√	√	–	√
8MHz	√	√	–	√
7.3728MHz	√	√	√	√
6MHz	√	√	√	–
5MHz	√	√	–	–
4.5MHz	√	√	–	√
4.194304MHz	√	√	√	–
4MHz	√	√	–	–
3.58MHz	√	√	√	√
3MHz	√	√	√	–
2MHz	√	–	–	–

√ : Communications possible
– : Communications not possible

Software Commands

Table 1.83 lists software commands. In the standard serial I/O mode 2, erase operations, programs and reading are controlled by transferring software commands via the RxD_0 pin. Standard serial I/O mode 2 adds four transmission speed commands - 9,600, 19,200, 38,400 and 57,600 bps - to the software commands of standard serial I/O mode 1. Software commands are explained here below.

Table 1.83. Software commands (Standard serial I/O mode 2)

	Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte		When ID is not verified
1	Page read	FF_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41_{16}	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Erase all unlocked blocks	$A7_{16}$	$D0_{16}$						Not acceptable
4	Read status register	70_{16}	SRD output	SRD1 output					Acceptable
5	Clear status register	50_{16}							Not acceptable
6	Read lock bit status	71_{16}	Address (middle)	Address (high)	Lock bit data output				Not acceptable
7	Code processing function	$F5_{16}$	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
8	Download function	FA_{16}	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
9	Version data output function	FB_{16}	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
10	Boot ROM area output function	FC_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
11	Baud rate 9600	$B0_{16}$	$B0_{16}$						Acceptable
12	Baud rate 19200	$B1_{16}$	$B1_{16}$						Acceptable
13	Baud rate 38400	$B2_{16}$	$B2_{16}$						Acceptable
14	Baud rate 57600	$B3_{16}$	$B3_{16}$						Acceptable

Note 1: Shading indicates transfer from flash memory microcomputer to peripheral unit. All other data is transferred from the peripheral unit to the flash memory microcomputer.

Note 2: SRD refers to status register data. SRD1 refers to status register 1 data.

Note 3: All commands can be accepted when the flash memory is totally blank.

Page Read Command

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

(1) Transfer the "FF16" command code with the 1st byte.

(2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first in sync with the fall of the clock.

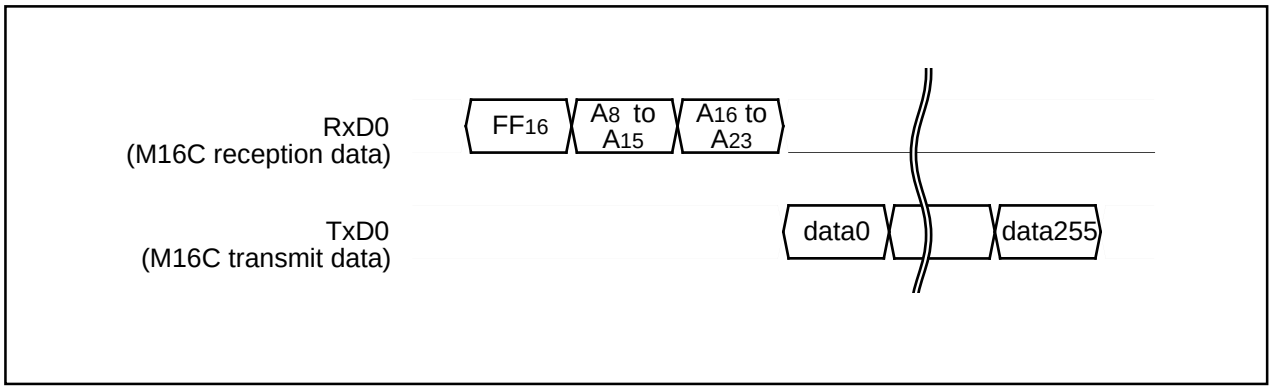


Figure 1.120. Timing for page read

Read Status Register Command

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

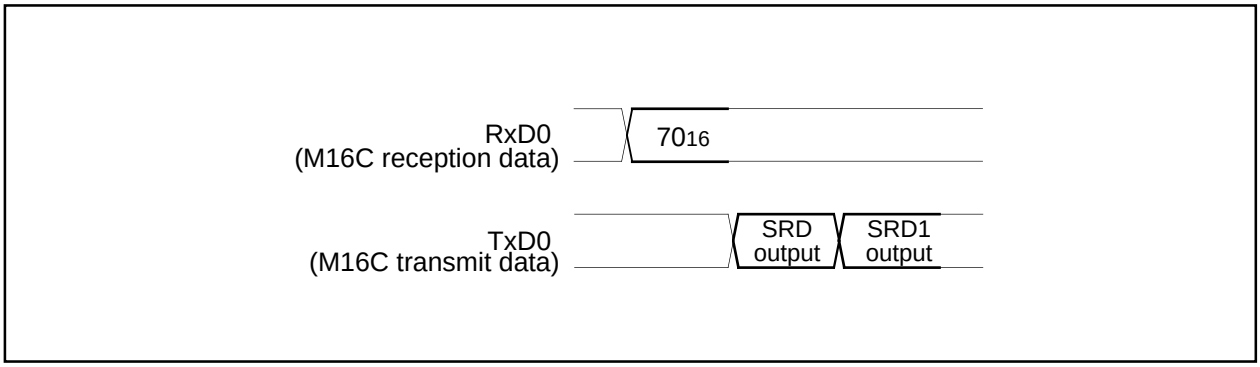


Figure 1.121. Timing for reading the status register

Clear Status Register Command

This command clears the bits (SR3–SR4) which are set when the status register operation ends in error. When the "50_{16}" command code is sent with the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the RTS_1 (BUSY) signal changes from the "H" to the "L" level.

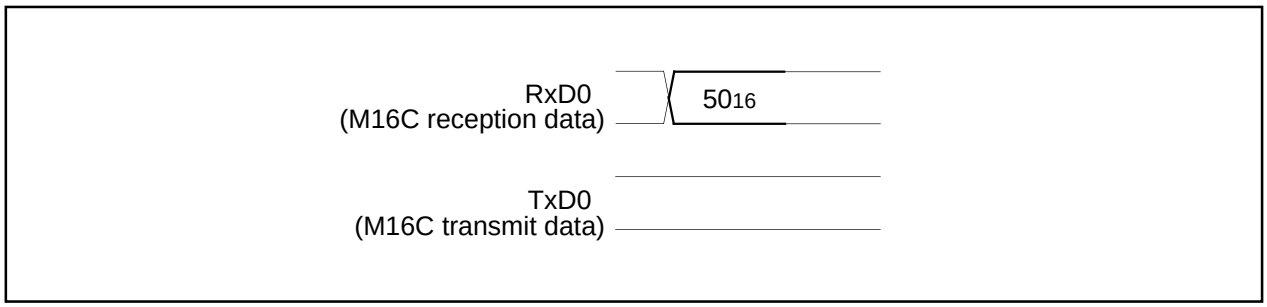


Figure 1.122. Timing for clearing the status register

Page Program Command

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

(1) Transfer the "41_{16}" command code with the 1st byte.
(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.
(3) From the 4th byte onward, as write data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the RTS_1 (BUSY) signal changes from the "H" to the "L" level. The result of the page program can be known by reading the status register. For more information, see the section on the status register.

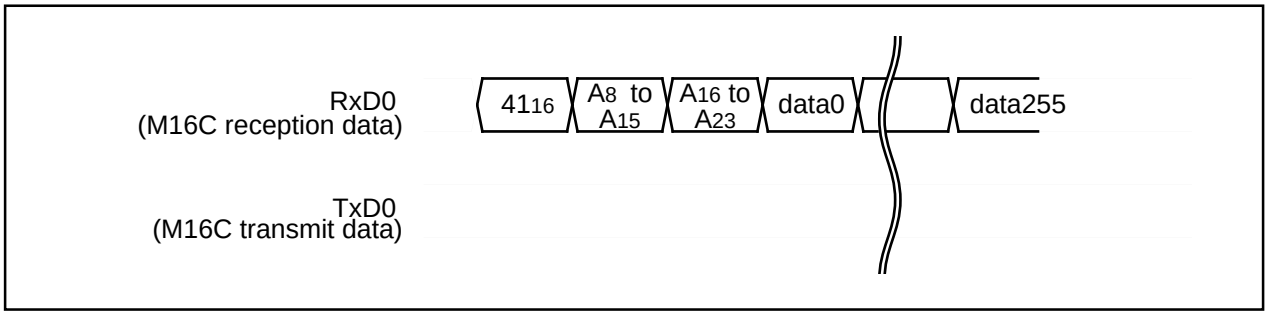


Figure 1.123. Timing for the page program

Erase All Unlocked Blocks Command

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

(1) Transfer the "A7$_{16}$" command code with the 1st byte.

(2) Transfer the verify command code "D0$_{16}$" with the 2nd byte. With the verify command code, the erase operation will start and continue for all blocks in the flash memory.

When block erasing ends, the RTS$_1$ (BUSY) signal changes from the "H" to the "L" level. The result of the erase operation can be known by reading the status register.

Figure 1.124. Timing for erasing all unlocked blocks

Read Lock Bit Status Command

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

(1) Transfer the "71_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) The lock bit data of the specified block is output with the 4th byte. Write the highest address of the specified block for addresses A_8 to A_{23}.

The M30201 (flash memory version) does not have the lock bit, so the read value is always "1" (block unlock).

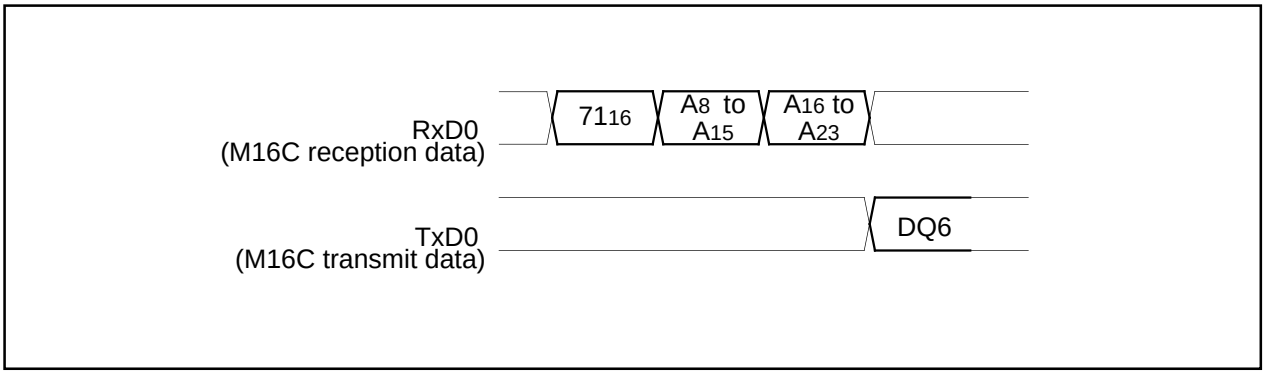


Figure 1.125. Timing for reading lock bit status

Download Command

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

(1) Transfer the "FA_{16}" command code with the 1st byte.

(2) Transfer the program size with the 2nd and 3rd bytes.

(3) Transfer the check sum with the 4th byte. The check sum is added to all data sent with the 5th byte onward.

(4) The program to execute is sent with the 5th byte onward.

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

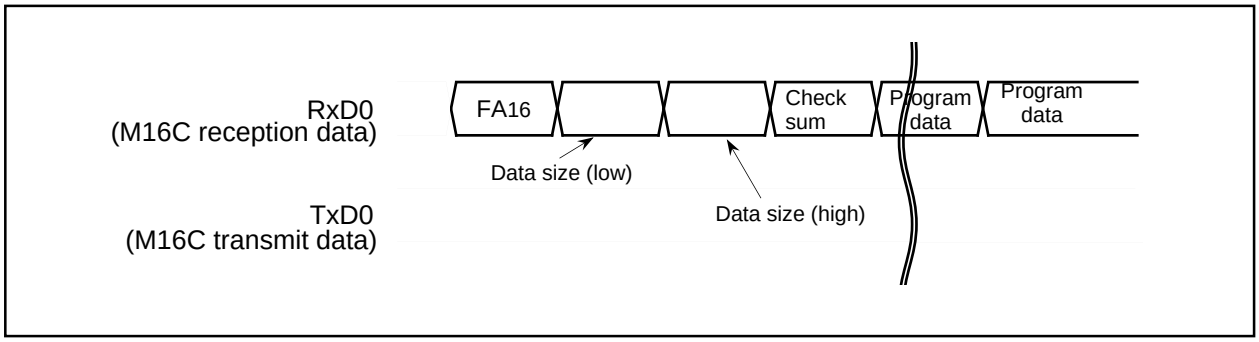


Figure 1.126. Timing for download

Version Information Output Command

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

(1) Transfer the "FB16" command code with the 1st byte.

(2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

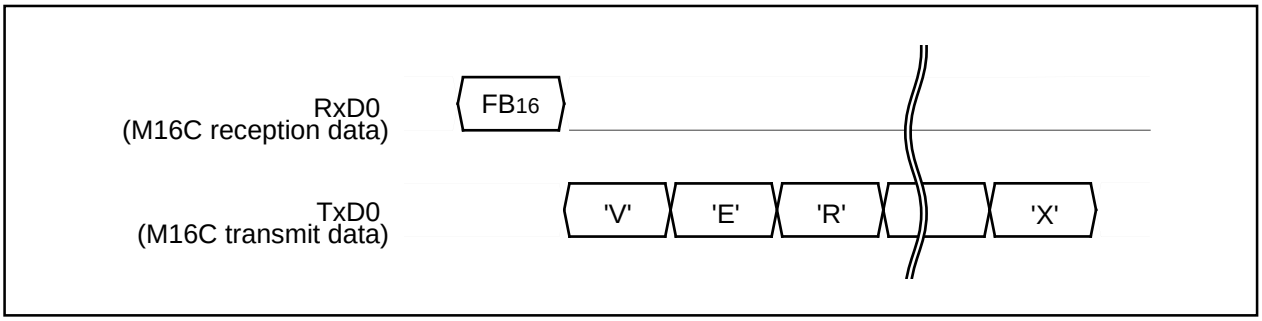


Figure 1.127. Timing for version information output

Boot ROM Area Output Command

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

(1) Transfer the "FC16" command code with the 1st byte.

(2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first.

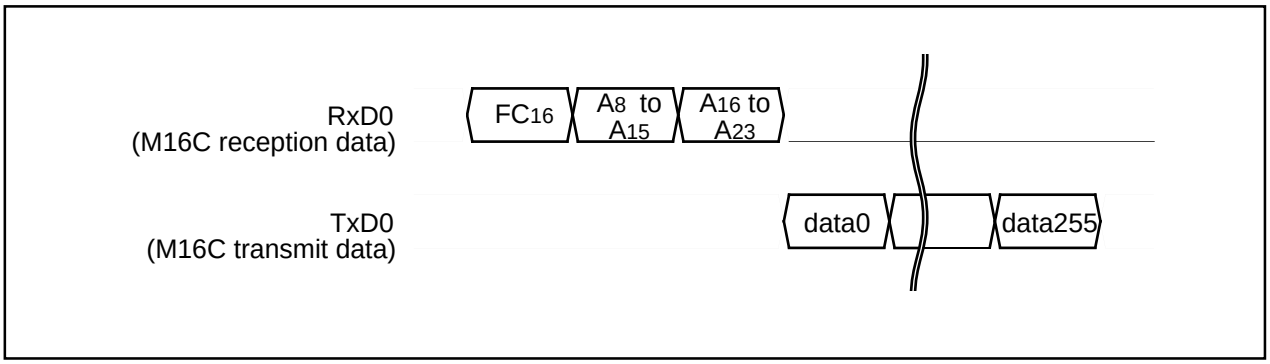


Figure 1.128. Timing for boot ROM area output

ID Check

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

(1) Transfer the "F5$_{16}$" command code with the 1st byte.

(2) Transfer addresses A$_0$ to A$_7$, A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ of the 1st byte of the ID code with the 2nd, 3rd and 4th bytes respectively.

(3) Transfer the number of data sets of the ID code with the 5th byte.

(4) The ID code is sent with the 6th byte onward, starting with the 1st byte of the code.

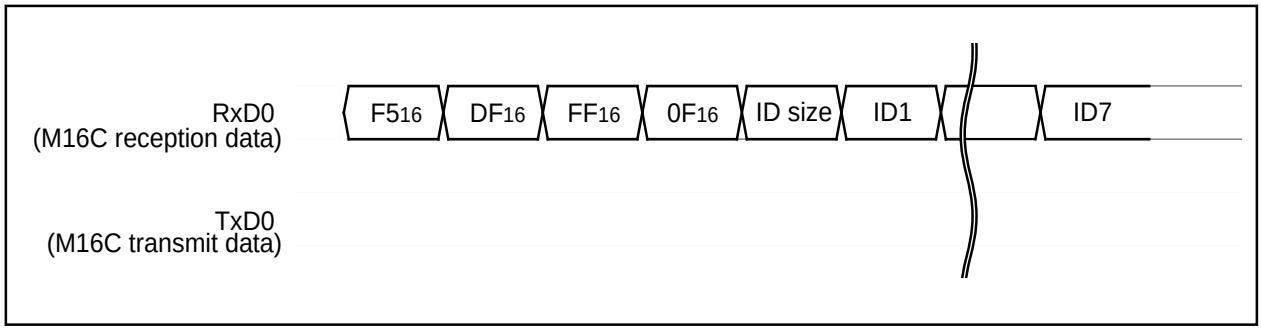


Figure 1.129. Timing for the ID check

ID Code

When the flash memory is not blank, the ID code sent from the peripheral units and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the peripheral units is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses 0FFFDF$_{16}$, 0FFFE3$_{16}$, 0FFFEB$_{16}$, 0FFFEF$_{16}$, 0FFFF3$_{16}$, 0FFFF7$_{16}$ and 0FFFFB$_{16}$. Write a program into the flash memory, which already has the ID code set for these addresses.

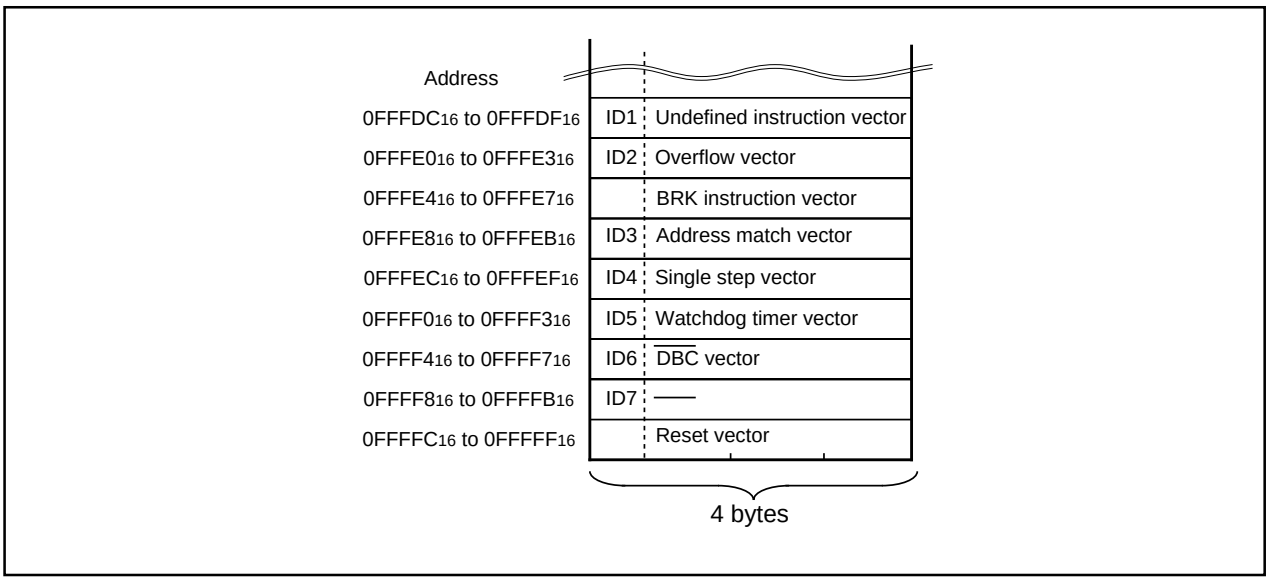


Figure 1.130. ID code storage addresses

Baud Rate 9600

This command changes baud rate to 9,600 bps. Execute it as follows.

(1) Transfer the "B0$_{16}$" command code with the 1st byte.

(2) After the "B0$_{16}$" check code is output with the 2nd byte, change the baud rate to 9,600 bps.

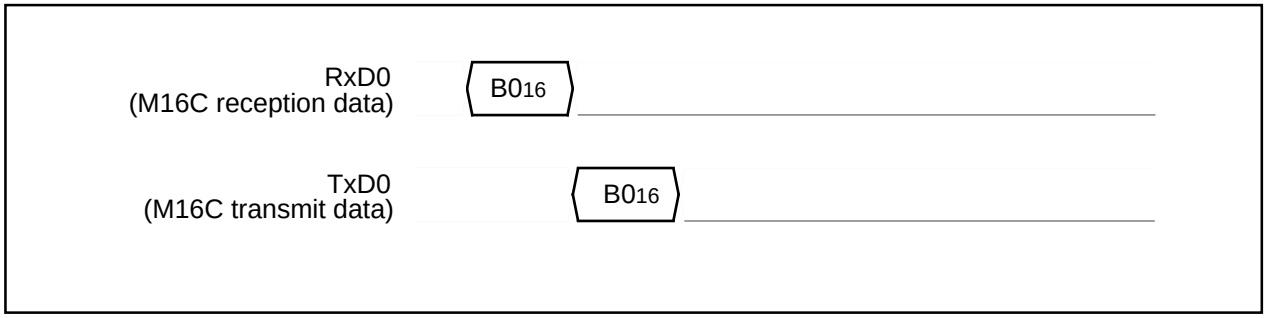


Figure 1.131. Timing of baud rate 9600

Baud Rate 19200

This command changes baud rate to 19,200 bps. Execute it as follows.

(1) Transfer the "B1$_{16}$" command code with the 1st byte.

(2) After the "B1$_{16}$" check code is output with the 2nd byte, change the baud rate to 19,200 bps.

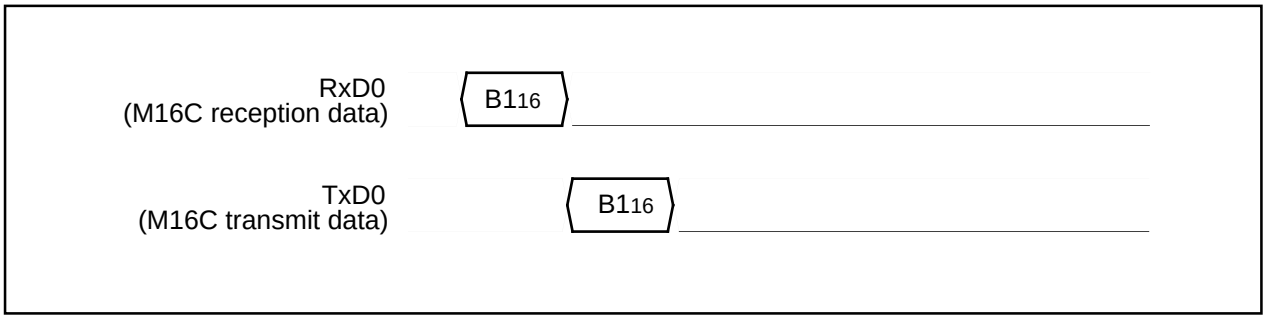


Figure 1.132. Timing of baud rate 19200

Baud Rate 38400

This command changes baud rate to 38,400 bps. Execute it as follows.

(1) Transfer the "B2$_{16}$" command code with the 1st byte.

(2) After the "B2$_{16}$" check code is output with the 2nd byte, change the baud rate to 38,400 bps.

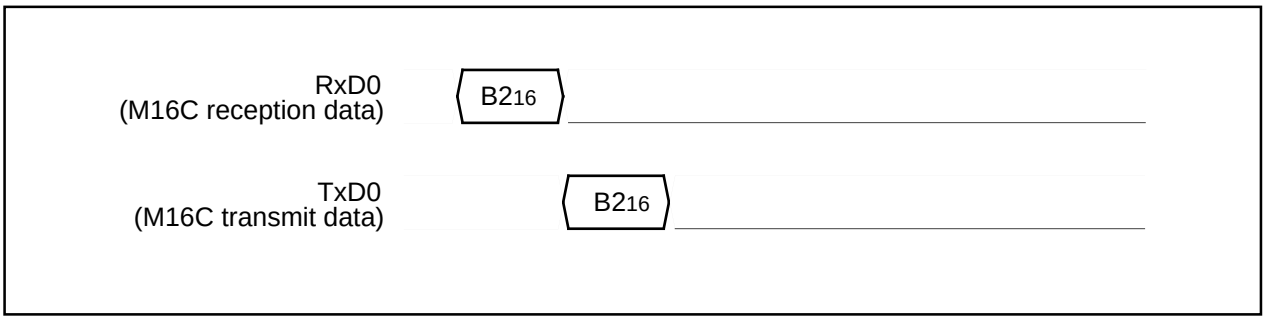


Figure 1.133. Timing of baud rate 38400

Baud Rate 57600

This command changes baud rate to 57,600 bps. Execute it as follows.

(1) Transfer the "B3$_{16}$" command code with the 1st byte.

(2) After the "B3$_{16}$" check code is output with the 2nd byte, change the baud rate to 57,600 bps.

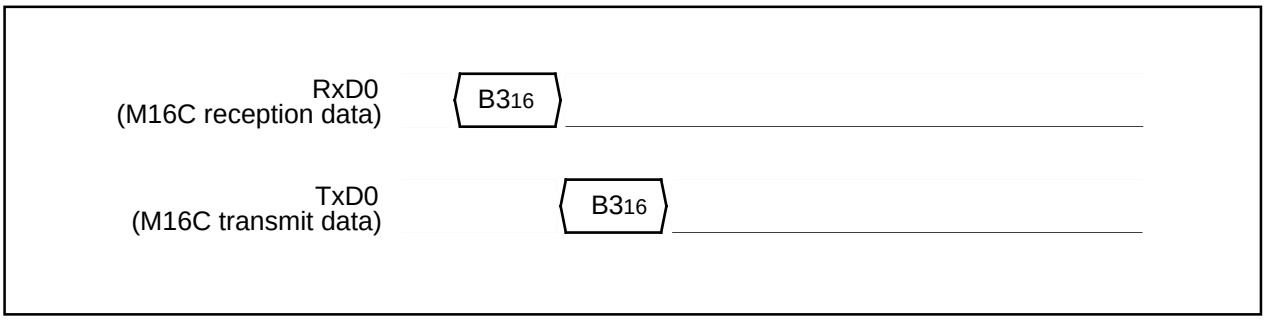
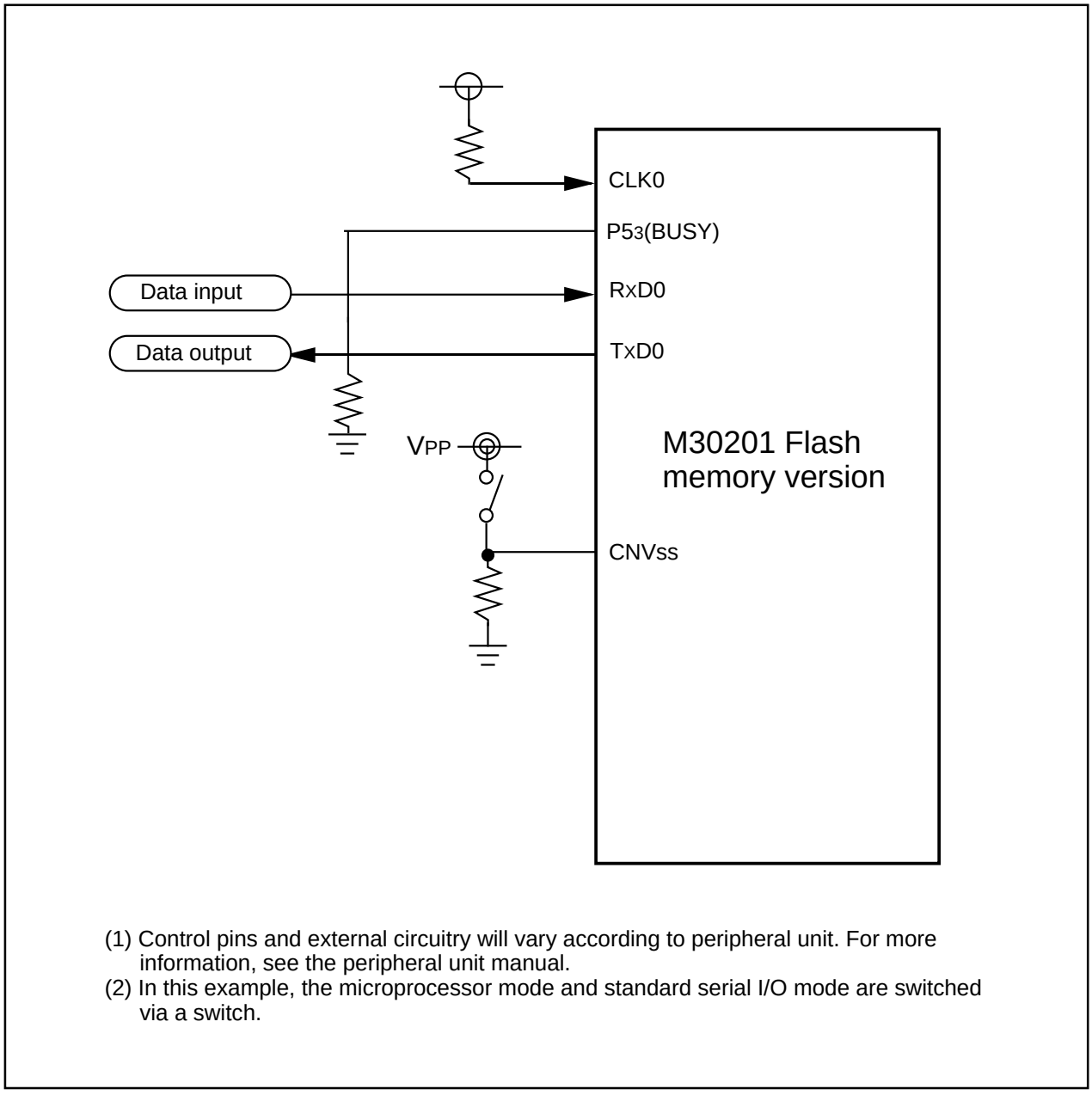


Figure 1.134. Timing of baud rate 57600

Example Circuit Application for The Standard Serial I/O Mode 2

The below figure shows a circuit application for the standard serial I/O mode 2.



(1) Control pins and external circuitry will vary according to peripheral unit. For more information, see the peripheral unit manual.
(2) In this example, the microprocessor mode and standard serial I/O mode are switched via a switch.

Figure 1.135. Example circuit application for the standard serial I/O mode 2

Revision History

Version	Contents for change	Revision date
REV.E	Pages 1, 5 internal interrupt 9 ->13 Pages 1, 5 2.7 to 5.5V (f(XIN)=7MHz with software one-wait):mask ROM version ->2.7 to 5.5V (f(XIN)=3.5MHz):mask ROM version Page 5 Power consumption 18mA (f(XIN)=7MHz with software one-wait, VCC=3V) ->11mA (f(XIN)=3.5MHz , VCC=3V) Page 6 M30201M2-XXXSP/FP, M30201M2T-XXXSP/FP ->Delete M30201M4T-XXXSP, M30201F6T-XXXSP ->Delete M30201M6-XXXFP, M30201M6T-XXXFP ->Addition Pages 9, 10 Figures 1.7 and 1.8 are partly revised. Page 14 Figure 1.11 is partly revised. Page 16 Figure 1.14 is partly revised (Bit 7 of the processor mode register 1). Wait bit ->Reserved bit Page 17 ~~Software wait~~ Page 20 Figure 1.18 is partly revised (Note 8 is partly revised). Page 21 Figure 1.19 is partly revised (n=07$_{16}$: approx. 16.5kHz -> 19.5kHz). Page 33 Figure 1.24 is partly revised (Note 2 is added). Page 49 Figure 1.39 is partly revised. Page 77 Figure 1.72 is partly revised (UARTi transmit/receive mode register). Page 78 Figure 1.73 is partly revised. Page 80 Figure 1.74 is partly revised. Page 85 Figure 1.79 is partly revised. Pages 90 to 96 Figures 1.83 to 1.89 are partly revised. Pages 110 to 113, 118 to 122 Tables 1.36 to 1.39 and 1.56 to 1.71 are partly revised. Page 124 Table 1.74 is partly revised (Boot ROM area 4 K bytes -> 3.5 K bytes) . Page 142 to 168 Standard serial I/O mode 2 is added.	01.4.12

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

MITSUBISHI SEMICONDUCTORS
M30201 Group DATA SHEET REV.E

April First Edition 1998
July Second Edition 1998
February Third Edition 1999
May Fourth Edition 1999
April Fifth Edition 2001
Editioned by
Committee of editing of Mitsubishi Semiconductor DATA SHEET

Published by
Mitsubishi Electric Corp., Kitaitami Works

This book, or parts thereof, may not be reproduced in any form without permission of Mitsubishi Electric Corporation.
©2001 MITSUBISHI ELECTRIC CORPORATION