Overview

The M30100, M30101 and M30102 groups consist of single-chip microcomputers that use high-performance silicon gate CMOS processes and have a on-chip M16C/60 series CPU core. The microcomputers are housed in 32-pin plastic mold QFP, 42-pin plastic mold SDIP, or 48-pin plastic mold QFP packages. These single-chip microcomputers have both high function instructions and high instruction efficiency and feature a one-megabyte address space and the capability to execute instructions at high speed.

The M30100, M30101 and M30102 groups consist of several products that have different on-chip memory types, sizes, and packages.

Features

• Basic machine language instructions	Compatible with the M16C/60 series
• Memory size	ROM/RAM (See the memory expansion diagram.)
• Shortest instruction execution time	62.5 ns (when f(X_{IN})=16MHz)
• Power supply voltage	4.2 V to 5.5V (when f(X_{IN})=16MHz) 2.7 V to 5.5V (when f(X_{IN})=5MHz)
• Interrupts	12 internal causes, 7 external causes, 4 software causes (including key input interrupts)
• 8-bit timers	4 with 8-bit prescaler (PWM output of Timer Y, Z: selectable)
• 16-bit timer	1 (time measurement timer)
• Serial I/O	UART or clock synchronization type x 2
• A-D converter	10-bit X 12 channels (can be expanded to 14 channels)
• D-A converter	1
• Watchdog timer	1
• Programmable I/O ports	34
• LED drive ports	8
• Clock generation circuits	3 internal circuits • Main clock generation circuit An internal feedback resistor and an externally attached ceramic resonator/quartz crystal oscillator/RC oscillator. • Sub clock generation circuit An internal feedback resistor and an externally attached ceramic resonator/quartz crystal oscillator • Ring oscillator

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

Applications

Home appliances, office devices, audio, automobile, other

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

Pin Configuration

Figures 1.1.1 to 1.1.3 show pin configurations (top view).

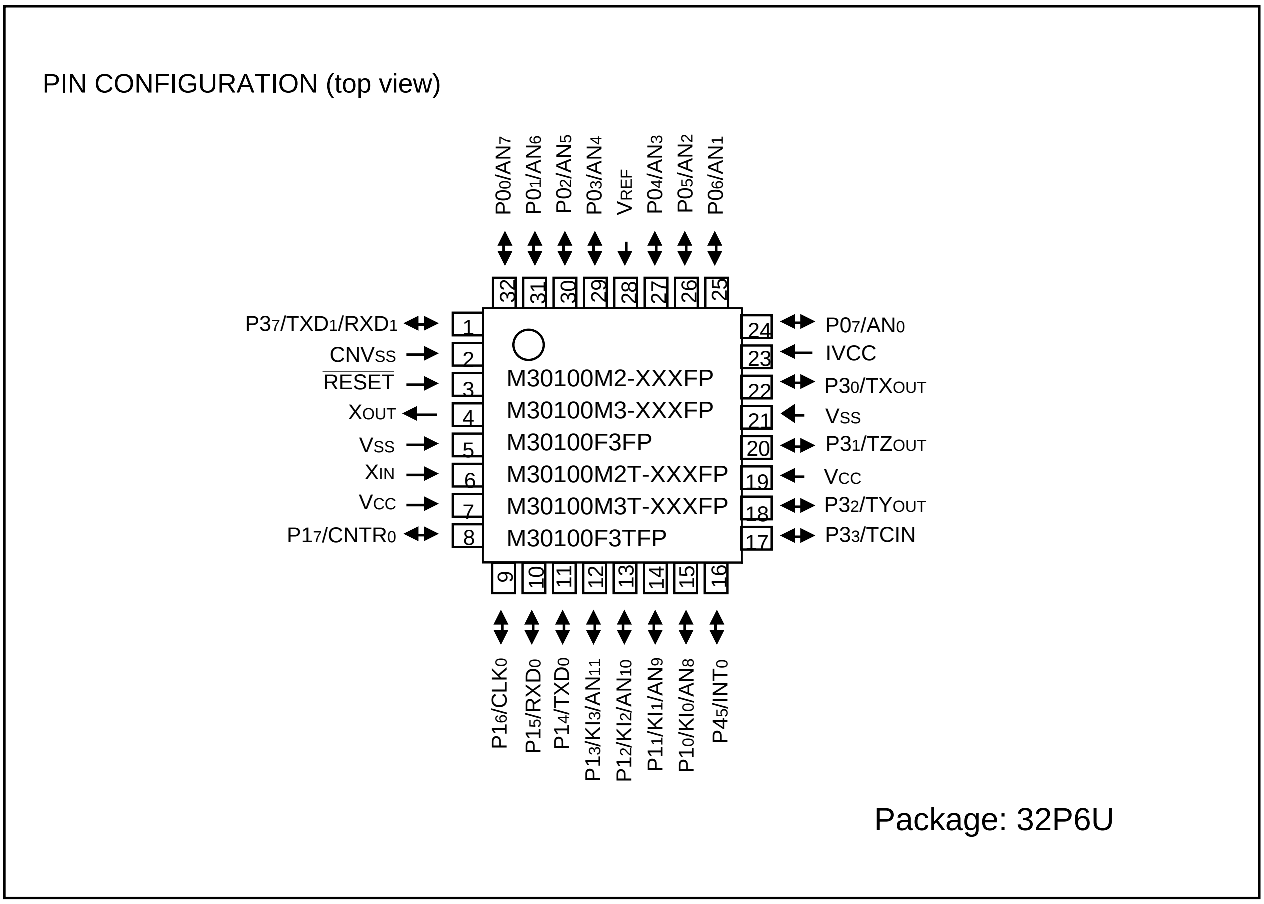
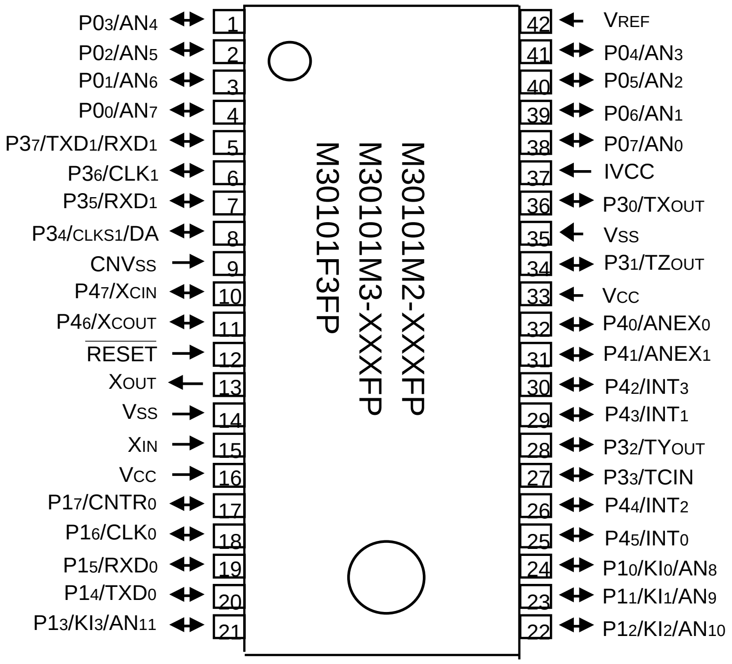


Figure 1.1.1. Pin configuration diagram (top view) of the M30100 group (QFP products)

PIN CONFIGURATION (top view)



Package: 42P2R

Figure 1.1.2. Pin configuration diagram (top view) of the M30101 group (shrink SSOP products)

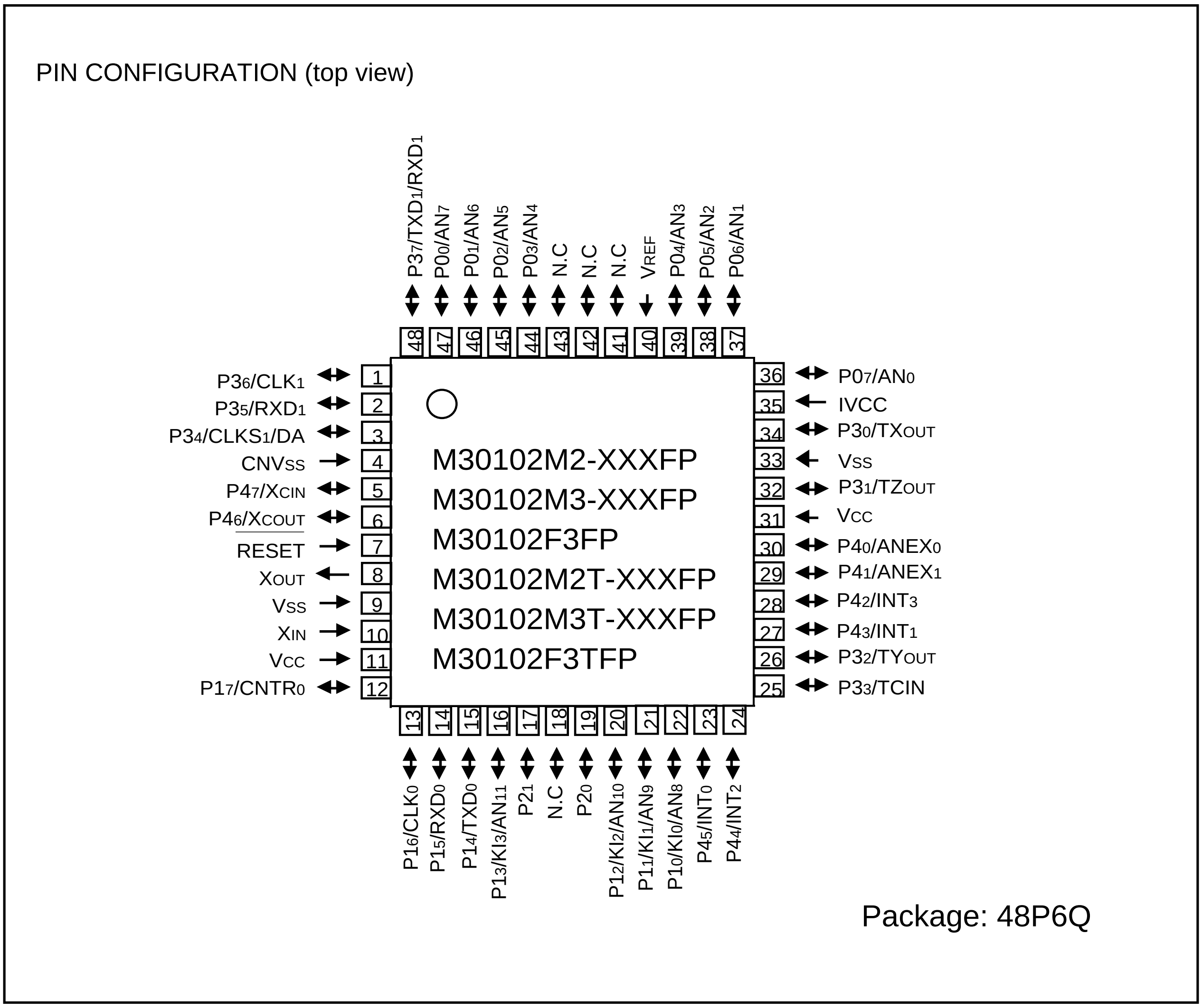
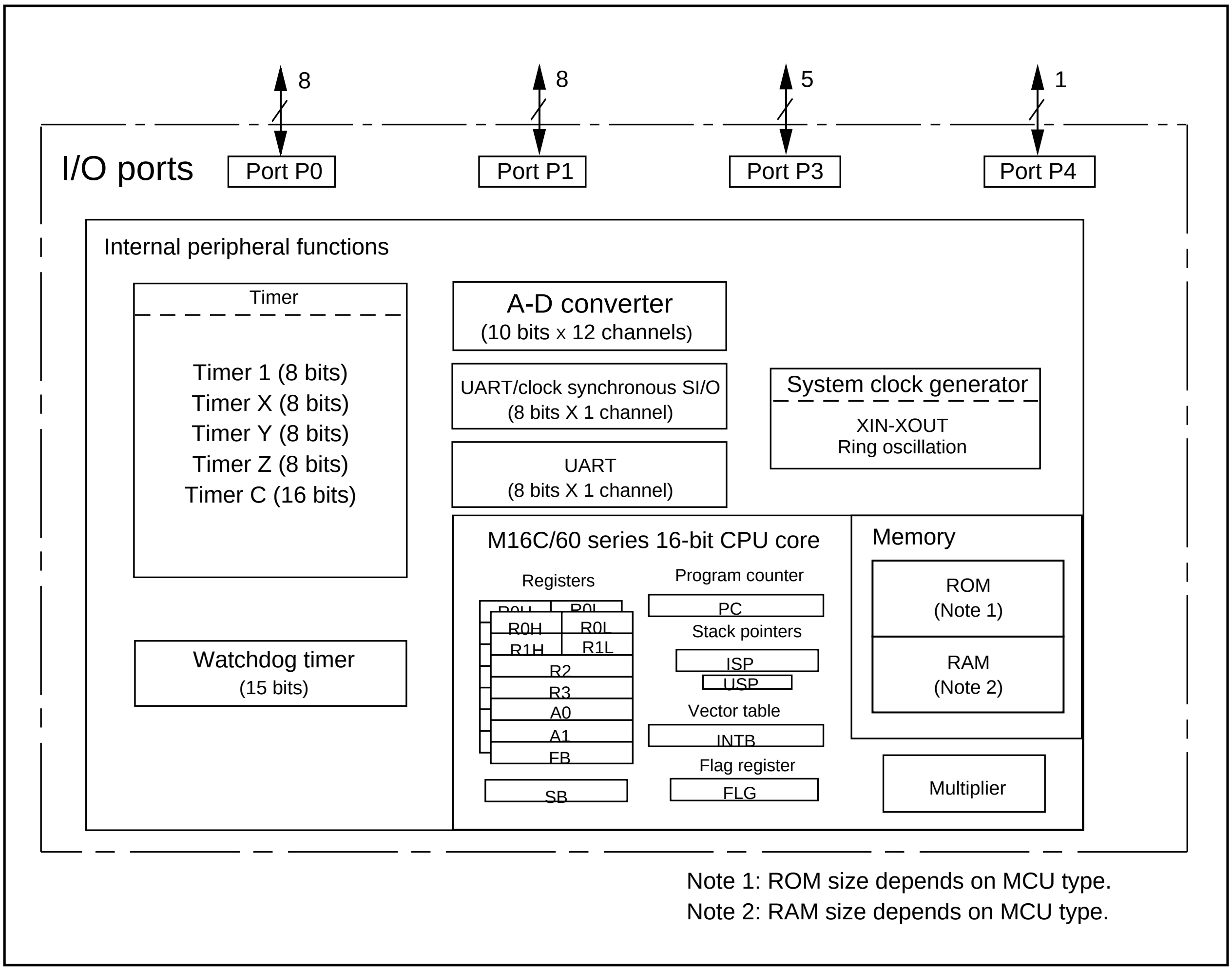


Figure 1.1.3. Pin configuration diagram (top view) of the M30102 group (QFP products)

Block Diagram

Figure 1.1.4 is a block diagram of the M30100 group.

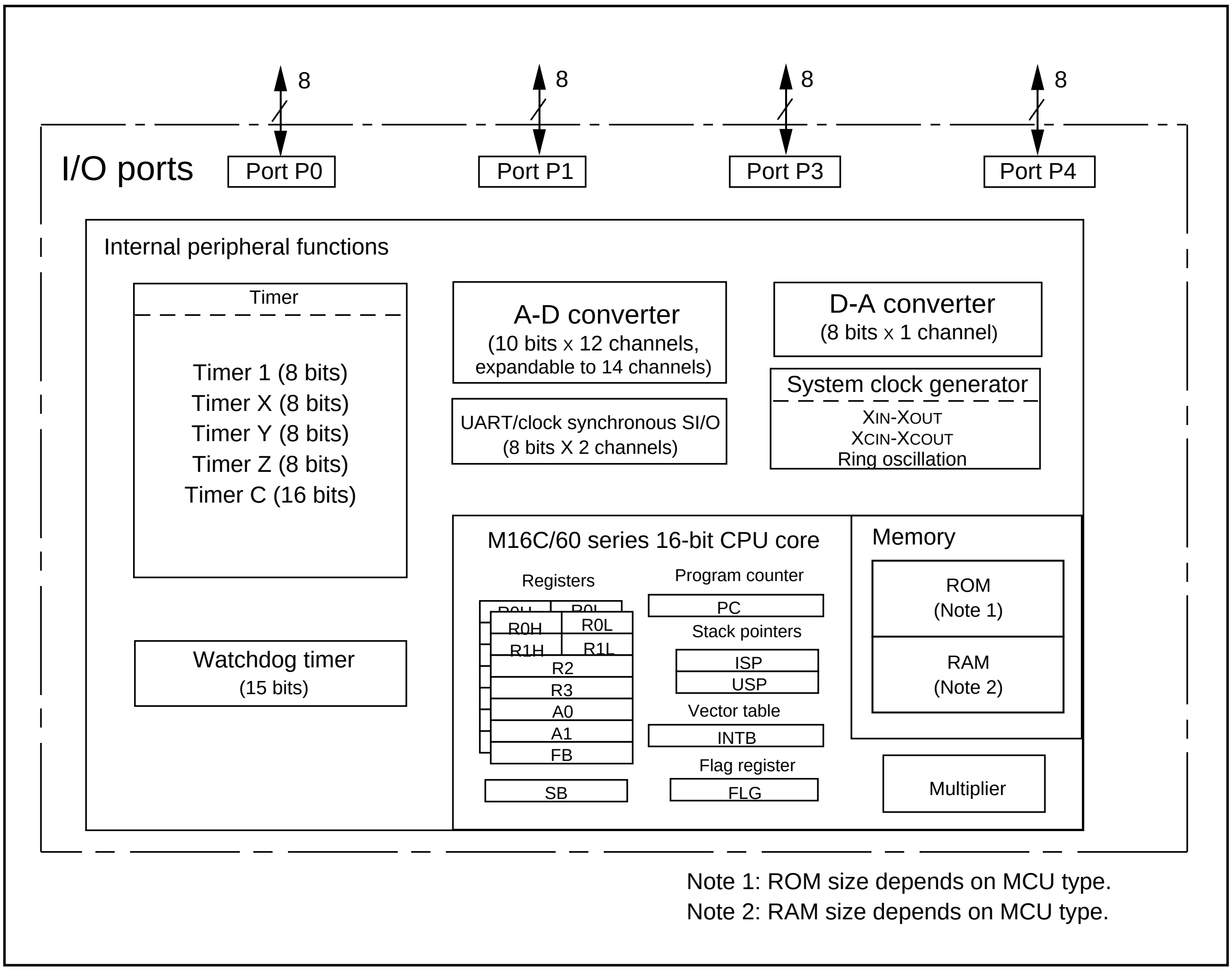


Note 1: ROM size depends on MCU type.
Note 2: RAM size depends on MCU type.

Figure 1.1.4. Block diagram for the M30100 group

Block Diagram

Figure 1.1.5 is a block diagram of the M30101 group.

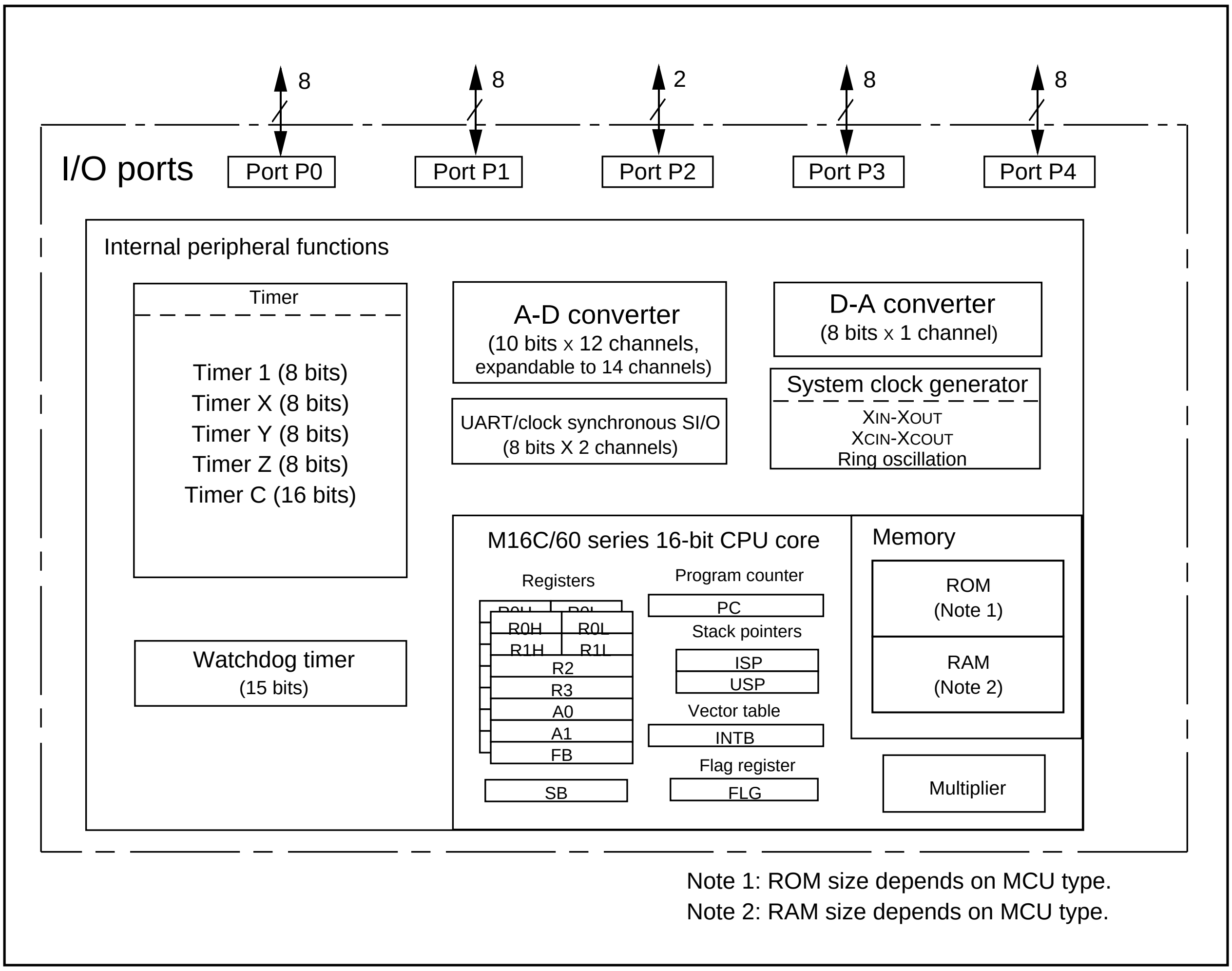


Note 1: ROM size depends on MCU type.
Note 2: RAM size depends on MCU type.

Figure 1.1.5. Block diagram for the M30101 group

Block Diagram

Figure 1.1.6 is a block diagram of the M30102 group.



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

Figure 1.1.6. Block diagram for the M30102 group

Performance Overview

Table 1.1.1 gives an overview of the M16C/10 group performance specification.

Table 1.1.1. M16C/10 group performance overview

Item		Performance		
		M30100	M30101	M30102
Number of basic instructions		91 instructions		
Shortest instruction execution time		62.5 ns (when f(X_{IN})=16MHz)		
Memory size	ROM	See the memory expansion diagram.		
	RAM	See the memory expansion diagram.		
I/O port		P0,P1,P3,P4: 22 lines	P0,P1,P3,P4: 32 lines	P0 to P4: 34 lines
Multifunction timer	T1	8 bits x 1		
	TX, TY, TZ	8 bits x 3		
	TC	16 bits x 1		
Serial I/O (UART or clock synchronous)		x 2 (one is exclusively for UART)	x 2	x 2
A-D converter (maximum resolution: 10 bits)		x 12 channels	x 12 channels (Expandable up to 14 channels)	
D-A converter		——	8 bits x 1	
Watchdog timer		15 bits x 1 (with prescaler)		
Interrupts		12 internal causes, 7 external causes (4 for M30100), 4 software causes		
Clock generating circuits		2 internal circuits	3 internal circuits	
Power supply voltage		4.2 V to 5.5V (when f(X_{IN})=16MHz) 2.7 V to 5.5V (when f(X_{IN})=5MHz)		
Power consumption		TBD		
I/O characteristics	I/O withstand voltage	5V		
	Output current	5mA (10mA:LED drive port)		
Device configuration		CMOS silicon gate		
Package		32-pin LQFP	42-pin SSOP	48-pin LQFP

Mitsubishi plans to release the following products in the M16C/10 group:

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

(2) Memory size

(3) Package

32P6U:	Plastic molded LQFP (mask ROM version and flash memory version)
42P2R:	Plastic molded SSOP (mask ROM version and flash memory version)

As of Oct., 2001

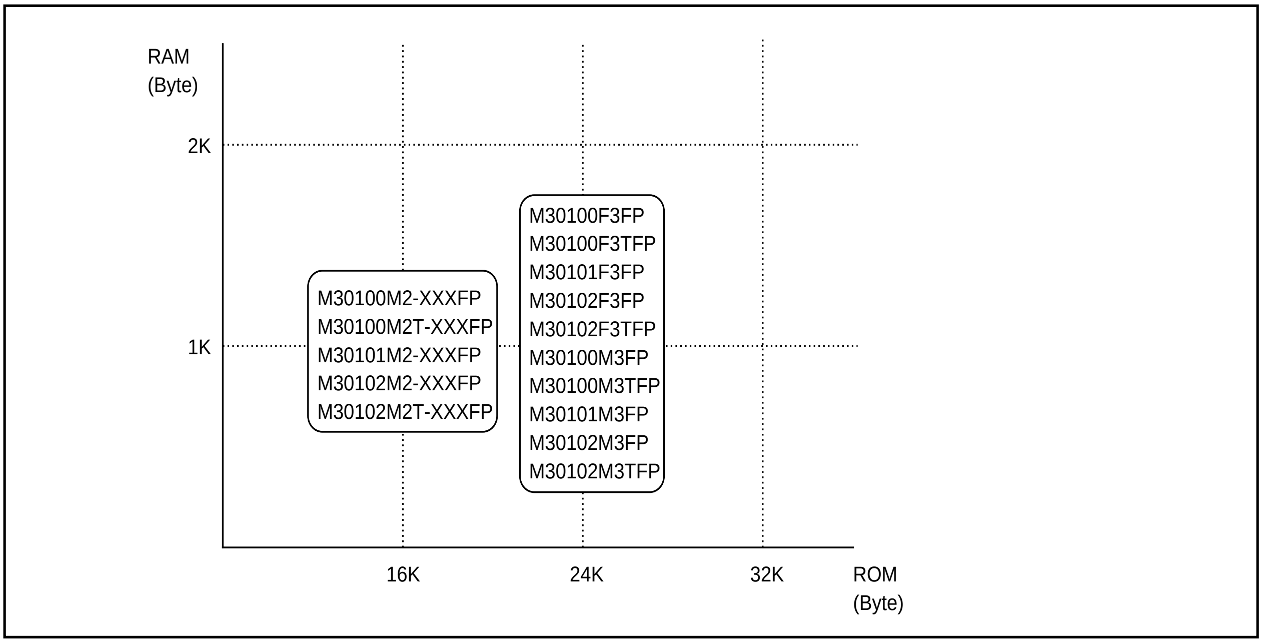


Figure 1.1.7. Memory expansion

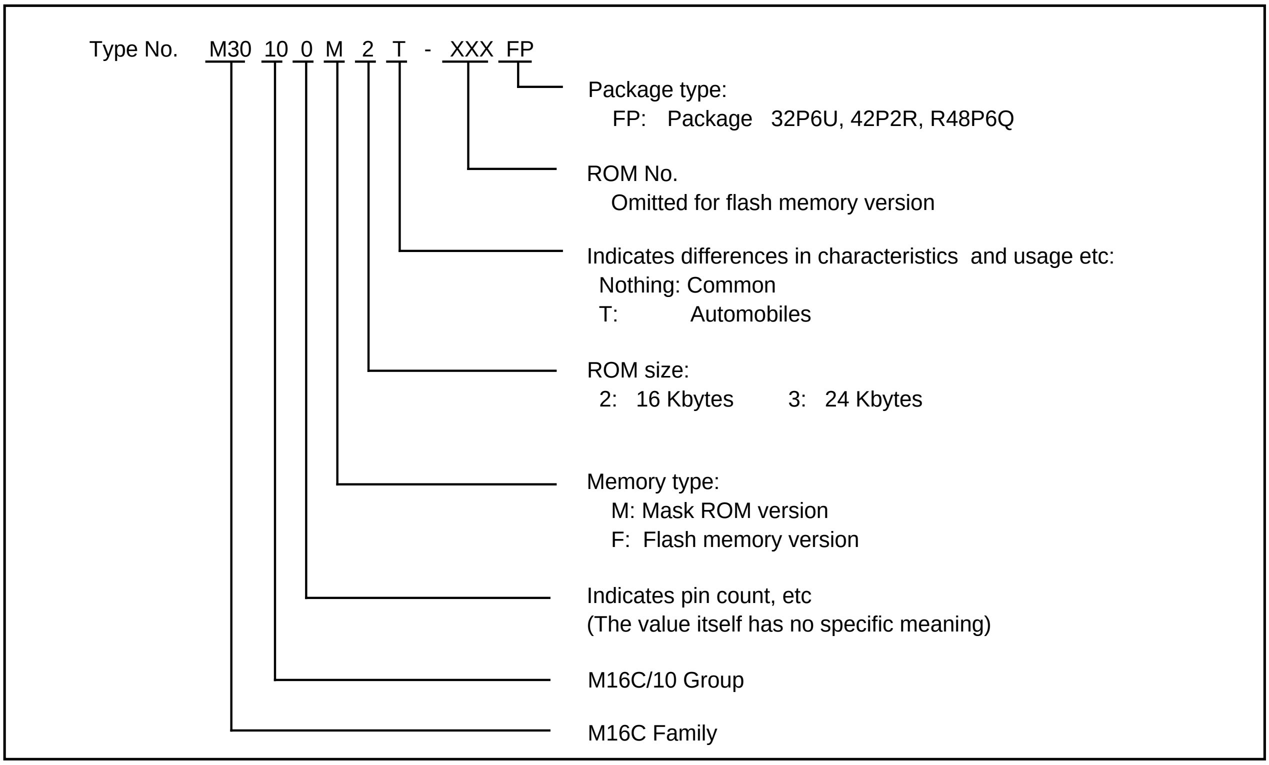

Type No. M30 10 0 M 2 T - XXX FP
Package type:
FP: Package 32P6U, 42P2R, R48P6Q
ROM No.
Omitted for flash memory version
Indicates differences in characteristics and usage etc:
Nothing: Common
T: Automobiles
ROM size:
2: 16 Kbytes 3: 24 Kbytes
Memory type:
M: Mask ROM version
F: Flash memory version
Indicates pin count, etc
(The value itself has no specific meaning)
M16C/10 Group
M16C Family

Figure 1.1.8. 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.
IVCC			Connect a capacitor (0.1 µF) between this pin and V_{SS}.
CNV_{SS}	CNV_{SS}	Input	Connect it to the V_{SS} pin.
$\overline{RESET}$	Reset input	Input	An “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 oscillation circuit. Connect a ceramic resonator or crystal between the X_{IN} and X_{OUT} pins. To use an externally derived clock, input it to the X_{IN} pin and leave the X_{OUT} pin open.
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. These pins are shared with analog input pins.
$P1_0$ to $P1_7$	I/O port P1	Input/output	This is an 8-bit I/O port equivalent to P0. $P1_0$ to $P1_3$ are shared with analog inputs and key input interrupts. $P1_4$ to $P1_6$ are shared with serial I/O pins. $P1_7$ is shared with timer input. Can be used as an LED drive port.
$P2_0$ to $P2_1$	I/O port P2	Input/output	This is a 2-bit I/O port equivalent to P0.
$P3_0$ to $P3_7$	I/O port P3	Input/output	This is a 8-bit I/O port equivalent to P0. $P3_0$ to $P3_3$ are shared with timer input/output. $P3_4$ to $P3_7$ are shared with serial I/O. $P3_4$ is shared with analog outputs.
$P4_0$ to $P4_7$	I/O port P4	Input/output	This is a 8-bit I/O port equivalent to P0. $P4_0$ to 4_1 are shared with analog inputs. $P4_2$ to $P4_5$ are shared with interrupt inputs. $P4_6$ to $P4_7$ are shared with the I/O pin of the clock oscillation circuit for the clock.

Operation of Functional Blocks

The M30100/M30101/M30102 contain the following devices on a single chip: ROM and RAM, which function as memory for storing instructions and data; a central processing unit (CPU) that executes operations; and peripheral devices, such as timers, serial I/O, an A-D converter, an D-A converter, and I/O ports.

The individual devices are described below.

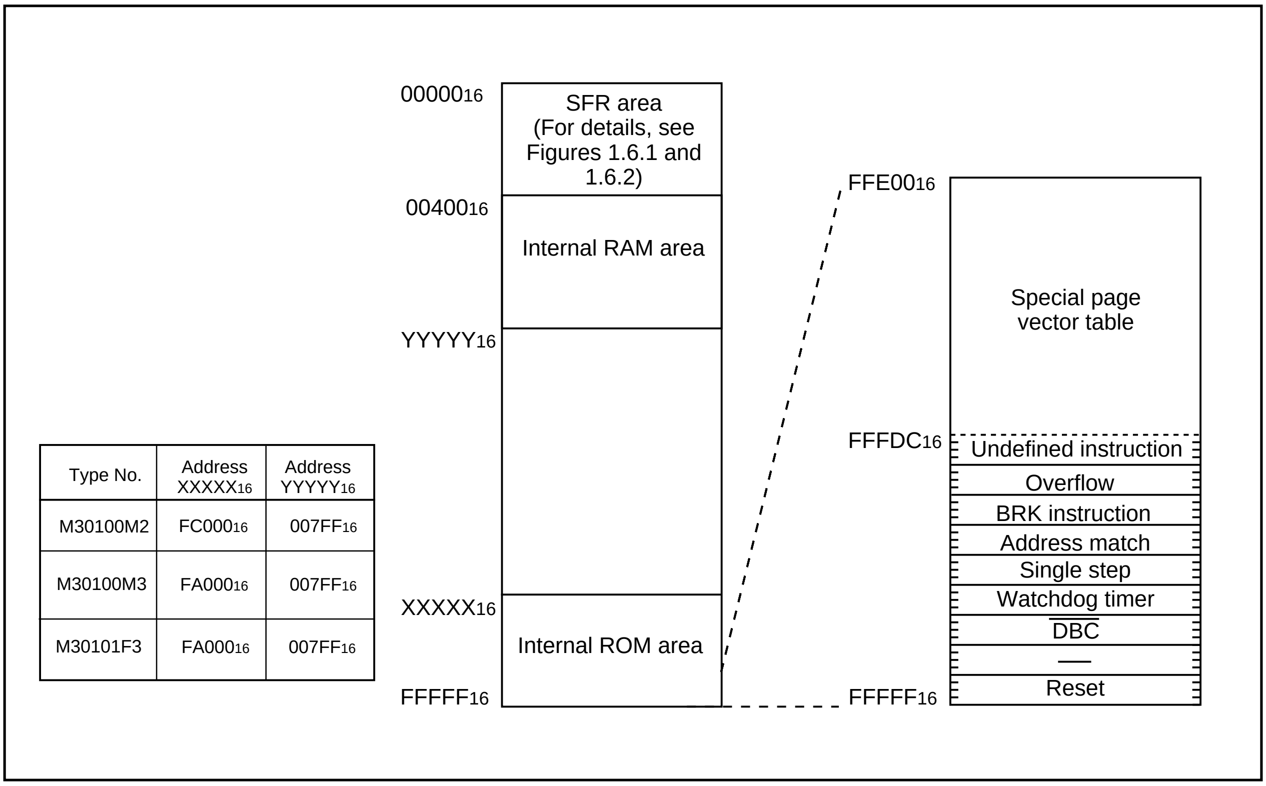
Memory

Figure 1.3.1 is a memory map. The address space extends the 1M bytes from address 00000_{16} to $FFFFF_{16}$. From $FFFFF_{16}$ down is ROM. For example, in the M30100M2-XXXFP, there is 16K bytes of internal ROM from $FC000_{16}$ 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 M30101M2-XXXFP, there is 1K byte of internal RAM from 00400_{16} to $007FF_{16}$. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.

The SFR area is mapped to 00000_{16} to $000FF_{16}$. This area accommodates the control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers, etc. Any part of the SFR area that is not occupied is reserved and cannot be used for other purposes.

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

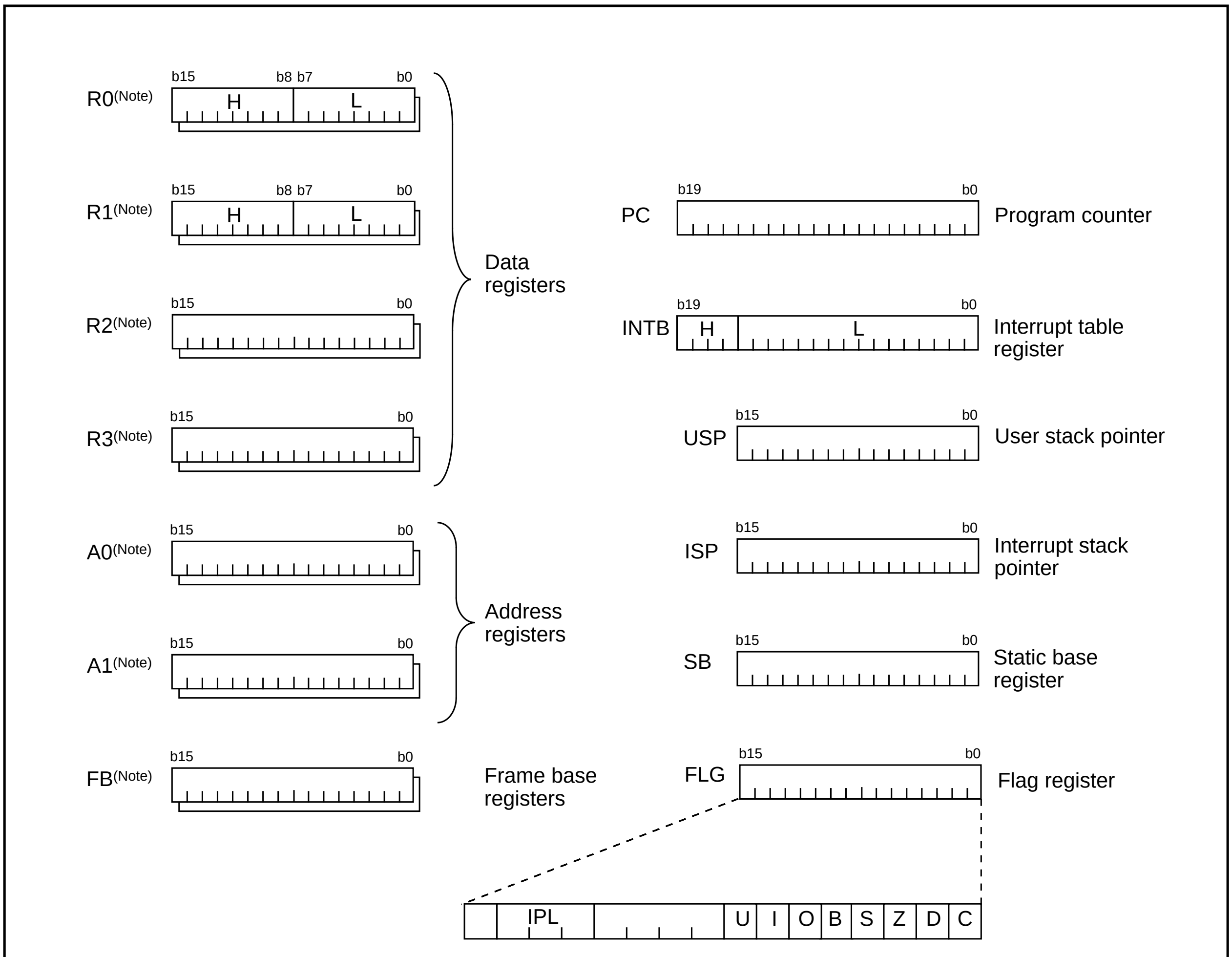


Type No.	Address $XXXXX_{16}$	Address $YYYYY_{16}$
M30100M2	$FC000_{16}$	$007FF_{16}$
M30100M3	$FA000_{16}$	$007FF_{16}$
M30101F3	$FA000_{16}$	$007FF_{16}$

Figure 1.3.1. Memory map

Central Processing Unit (CPU)

The CPU has a total of 13 registers shown in Figure 1.4.1. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these have two register banks.



Note: These registers consist of two register banks.

Figure 1.4.1. Central processing unit register

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

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

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

(2) Address registers (A0 and A1)

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

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

(3) Frame base register (FB)

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

(4) Program counter (PC)

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

(5) Interrupt table register (INTB)

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

(6) Stack pointer (USP/ISP)

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

Your desired type of stack pointer (USP or ISP) can be selected by a stack pointer select flag (U flag). This flag is located at the position of bit 7 in the flag register (FLG).

(7) Static base register (SB)

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

(8) Flag register (FLG)

Flag register (FLG) is configured with 11 bits, each bit is used as a flag. Figure 1.4.2 shows the flag register (FLG). The following explains the function of each flag:

- **Bit 0: Carry flag (C flag)**

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

- **Bit 1: Debug flag (D flag)**

  This flag enables a single-step interrupt.

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

- **Bit 2: Zero flag (Z flag)**

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

- **Bit 3: Sign flag (S flag)**

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

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

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

- **Bit 5: Overflow flag (O flag)**

  This flag is set to “1” when an arithmetic operation resulted in overflow.

- **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.

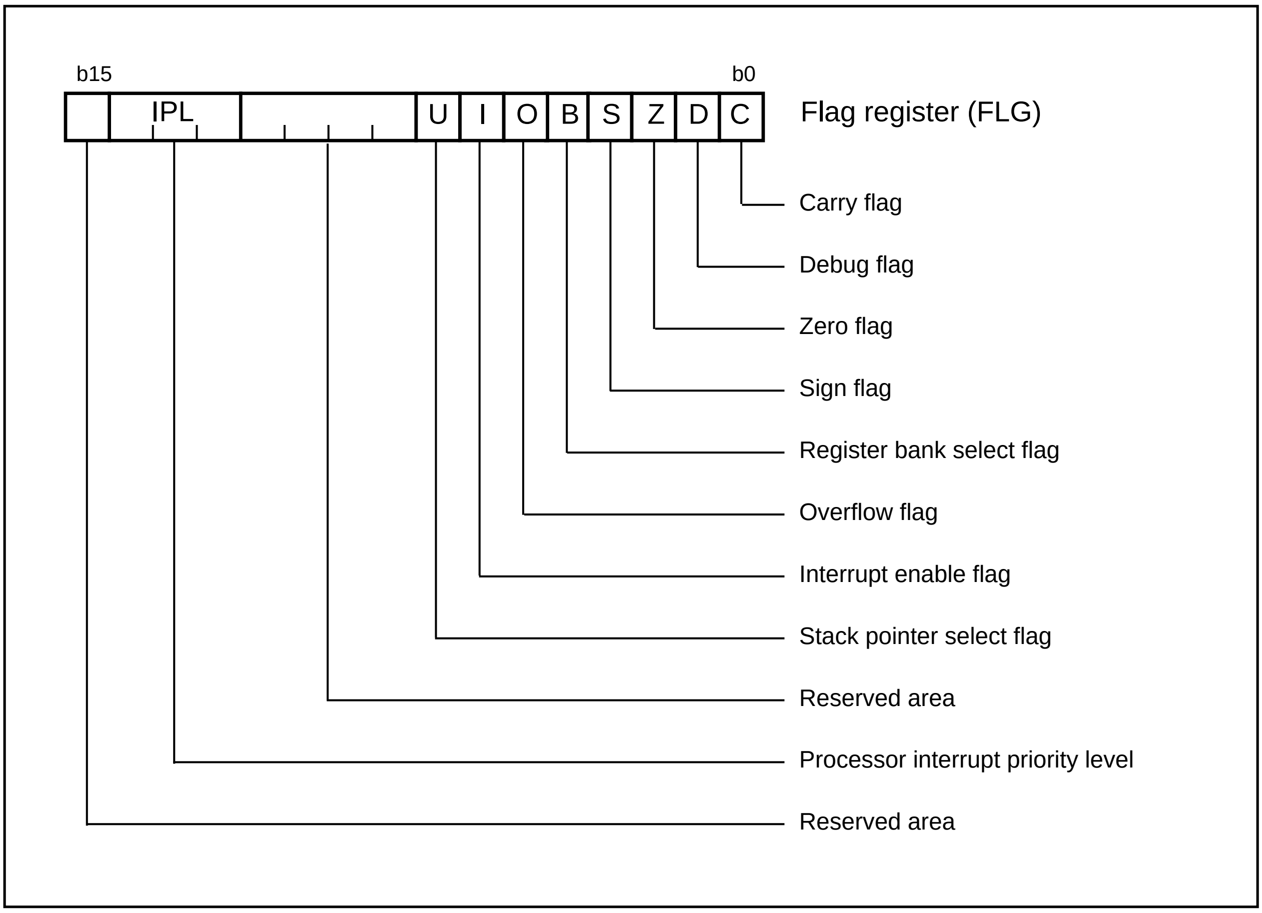


Figure 1.4.2. Flag register (FLG)

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, the reset status is cancelled and program execution resumes from the address in the reset vector table. Since the value of RAM is indeterminate when power is applied, the initial values must be set. Also, if a reset signal is input during write to RAM, the access to the RAM will be interrupted. Consequently, the value of the RAM being written may change to an unintended value due to the interruption.

Figures 1.5.1 and 1.5.2 show the example reset circuit. Figure 1.5.3 shows the reset sequence.

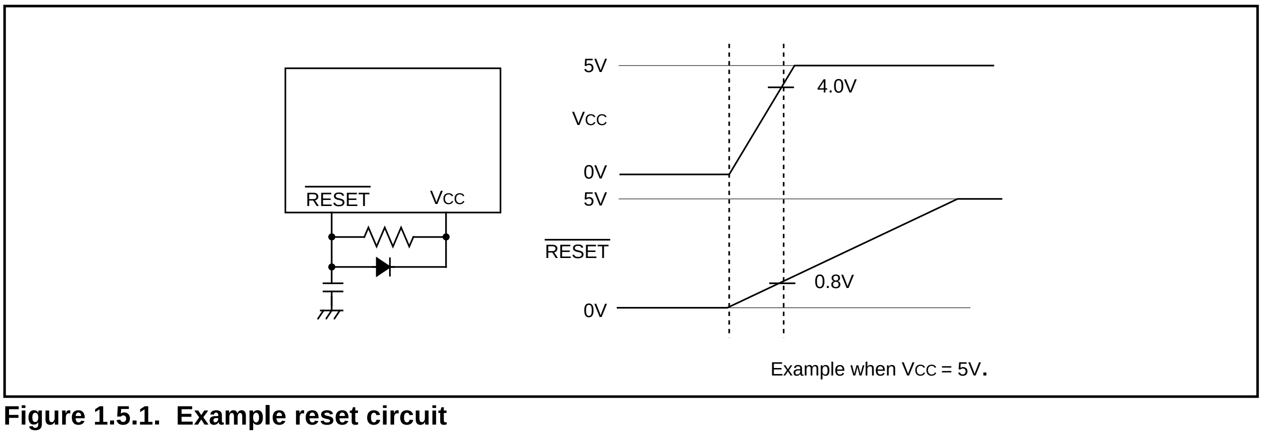


Figure 1.5.1. Example reset circuit

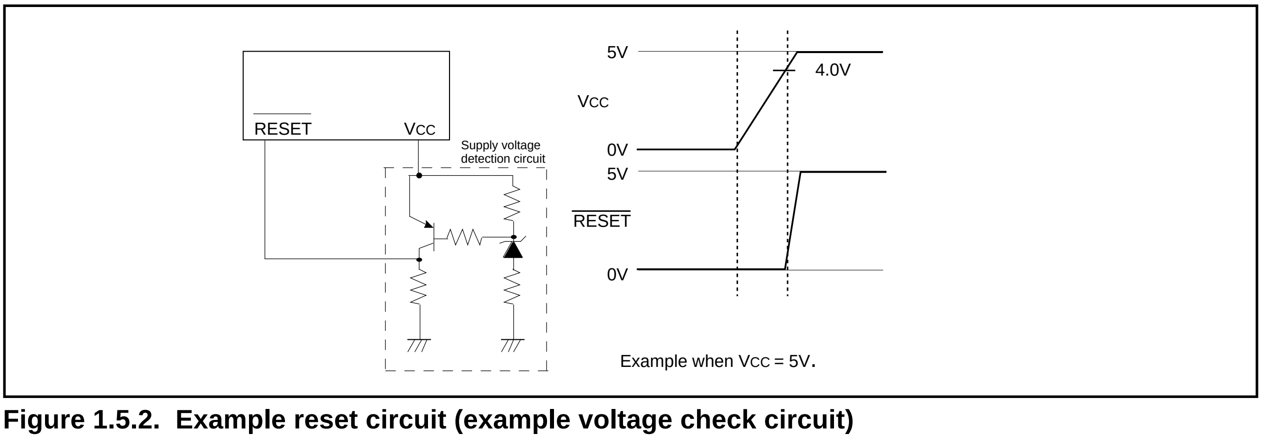


Figure 1.5.2. Example reset circuit (example voltage check circuit)

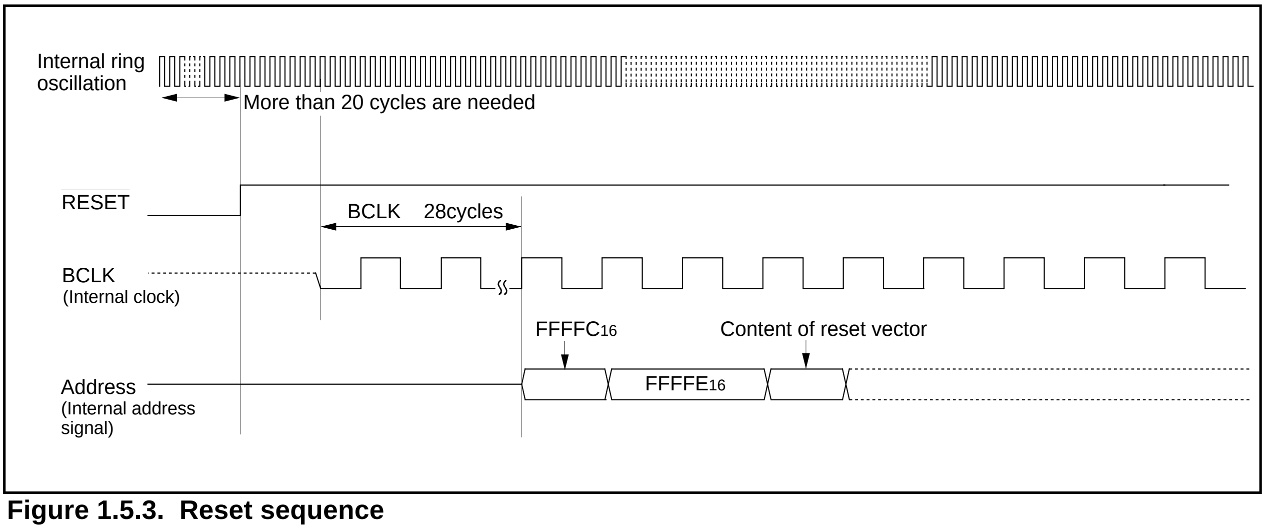


Figure 1.5.3. Reset sequence

(1)	Processor mode register 0	(0004$_{16}$)···	x x x x 0 x 0 0
(2)	Processor mode register 1	(0005$_{16}$)···	0 0 x x x 0 x 0
(3)	System clock control register 0	(0006$_{16}$)···	0 1 0 0 1 0 0 0
(4)	System clock control register 1	(0007$_{16}$)···	0 0 1 0 0 0 0 0
(5)	Address match interrupt enable register	(0009$_{16}$)···	x x x x x x 0 0
(6)	Protect register	(000A$_{16}$)···	x x x x x 0 0 0
(7)	Oscillation stop detection register	(000C$_{16}$)···	0 0 0 0 0 1 0 0
(8)	Watchdog timer control register	(000F$_{16}$)···	0 0 0 ? ? ? ? ?
(9)	Address match interrupt register 0	(0010$_{16}$)···	00$_{16}$
		(0011$_{16}$)···	00$_{16}$
		(0012$_{16}$)···	x x x x 0 0 0 0
(10)	Address match interrupt register 1	(0014$_{16}$)···	00$_{16}$
		(0015$_{16}$)···	00$_{16}$
		(0016$_{16}$)···	x x x x 0 0 0 0
(11)	INT0 input filter select register	(001E$_{16}$)···	x x x x x 0 0 0
(12)	Key input interrupt control register	(004D$_{16}$)···	x x x x ? 0 0 0
(13)	A-D conversion interrupt control register	(004E$_{16}$)···	x x x x ? 0 0 0
(14)	UART0 transmit interrupt control register	(0051$_{16}$)···	x x x x ? 0 0 0
(15)	UART0 receive interrupt control register	(0052$_{16}$)···	x x x x ? 0 0 0
(16)	UART1 transmit interrupt control register	(0053$_{16}$)···	x x x x ? 0 0 0
(17)	UART1 receive interrupt control register	(0054$_{16}$)···	x x x x ? 0 0 0
(18)	Timer 1 interrupt control register	(0055$_{16}$)···	x x x x ? 0 0 0
(19)	Timer X interrupt control register	(0056$_{16}$)···	x x x x ? 0 0 0
(20)	Timer Y interrupt control register	(0057$_{16}$)···	x x x x ? 0 0 0
(21)	Timer Z interrupt control register	(0058$_{16}$)···	x x x x ? 0 0 0
(22)	TCIN0 interrupt control register	(005A$_{16}$)···	x x x x ? 0 0 0
(23)	Timer C interrupt control register	(005B$_{16}$)···	x x x x ? 0 0 0
(24)	INT3 interrupt control register	(005C$_{16}$)···	x x x x ? 0 0 0
(25)	INT0 interrupt control register	(005D$_{16}$)···	x x 0 0 ? 0 0 0
(26)	INT1 interrupt control register	(005E$_{16}$)···	x x 0 0 ? 0 0 0
(27)	INT2 interrupt control register	(005F$_{16}$)···	x x 0 0 ? 0 0 0
(28)	Timer Y, Z mode register	(0080$_{16}$)···	0 0 0 0 0 0 x 0
(29)	Prescaler Y	(0081$_{16}$)···	FF$_{16}$
(30)	Timer Y secondary	(0082$_{16}$)···	FF$_{16}$
(31)	Timer Y primary	(0083$_{16}$)···	FF$_{16}$
(32)	Timer Y, Z waveform output control register	(0084$_{16}$)···	00$_{16}$
(33)	Prescaler Z	(0085$_{16}$)···	FF$_{16}$
(34)	Timer Z secondary	(0086$_{16}$)···	FF$_{16}$
(35)	Timer Z primary	(0087$_{16}$)···	FF$_{16}$

(36)	Prescaler 1	(0088$_{16}$)···	FF$_{16}$
(37)	Timer 1	(0089$_{16}$)···	01$_{16}$
(38)	Timer Y,Z output control register	(008A$_{16}$)···	x x x x x 0 0 0
(39)	Timer X mode register	(008B$_{16}$)···	x x x 0 0 0 0 0
(40)	Prescaler X	(008C$_{16}$)···	FF$_{16}$
(41)	Timer X	(008D$_{16}$)···	FF$_{16}$
(42)	Timer count source set register	(008E$_{16}$)···	x x 0 0 0 0 0 0
(43)	Clock prescaler reset flag	(008F$_{16}$)···	0 x x x x x x x
(44)	External input enable register	(0096$_{16}$)··	0 0 0 0 0 0 1 0
(45)	Key input enable register	(0098$_{16}$)··	x 0 0 0 0 0 0 0
(46)	Timer C control register 0	(009A$_{16}$)···	0 x x 0 0 0 0 0
(47)	Timer C control register 1	(009B$_{16}$)···	x x x x x x 1 1
(48)	UART0 transmit/receive mode register	(00A0$_{16}$)··	00$_{16}$
(49)	UART0 transmit/receive control register 0	(00A4$_{16}$)··	0 0 0 0 1 0 0 0
(50)	UART0 transmit/receive control register 1	(00A5$_{16}$)··	0 0 0 0 0 0 1 0
(51)	UART1 transmit/receive mode register	(00A8$_{16}$)···	00$_{16}$
(52)	UART1 transmit/receive control register 0	(00AC$_{16}$)··	0 0 0 0 1 0 0 0
(53)	UART1 transmit/receive control register 1	(00AD$_{16}$)··	0 0 0 0 0 0 1 0
(54)	UART transmit/receive control register 2	(00B0$_{16}$)··	00$_{16}$
(55)	A-D control register 2	(00D4$_{16}$)··	x x x x 0 0 0 0
(56)	A-D control register 0	(00D6$_{16}$)··	0 0 0 0 0 ? ? ?
(57)	A-D control register 1	(00D7$_{16}$)··	00$_{16}$
(58)	Port P0 direction register	(00E2$_{16}$)···	00$_{16}$
(59)	Port P1 direction register	(00E3$_{16}$)··	00$_{16}$
(60)	Port P2 direction register	(00E6$_{16}$)···	00$_{16}$
(61)	Port P3 direction register	(00E7$_{16}$)···	00$_{16}$
(62)	Port P4 direction register	(00EA$_{16}$)··	00$_{16}$
(63)	Pull-up control register 0	(00FC$_{16}$)···	0 0 x 0 0 0 0 0
(64)	Pull-up control register 1	(00FD$_{16}$)···	x x x x x x 0 0
(65)	Port P1 drive capacity control register	(00FE$_{16}$)··	00$_{16}$
(66)	Data registers (R0/R1/R2/R3)		0000$_{16}$
(67)	Address registers (A0/A1)		0000$_{16}$
(68)	Frame base register (FB)		0000$_{16}$
(69)	Interrupt table register (INTB)		00000$_{16}$
(70)	User stack pointer (USP)		0000$_{16}$
(71)	Interrupt stack pointer (ISP)		0000$_{16}$
(72)	Static base register (SB)		0000$_{16}$
(73)	Flag register (FLG)		0000$_{16}$

x : Nothing is mapped to this bit
? : Undefined

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

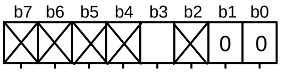
Figure 1.5.4. Device's internal status after a reset is cleared

Software Reset

Writing “1” to bit 3 of the processor mode register 0 (address 0004_{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 1.5.5 shows the processor mode register 0 and 1.

Processor mode register 0 (Note)

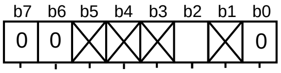


Symbol	Address	When reset
PM0	0004_{16}	$XXXX0X00_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.			—	—
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 1)



Symbol	Address	When reset
PM1	0005_{16}	$00XXX0X0_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
Nothing is assigned. In an attempt to write to this bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
PM12	WDT interrupt/reset switching bit	0 : Watchdog timer interrupt 1 : Reset (Note 2)	○	○
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 1: Set bit 1 of the protect register (address $000A_{16}$) to “1” when writing new values to this register.
Note 2: After setting this bit to "1", can not change to "0" by software.

Figure 1.5.5. Processor mode register 0 and 1.

0000$_{16}$	
0001$_{16}$	
0002$_{16}$	
0003$_{16}$	
0004$_{16}$	Processor mode register 0 (PM0)
0005$_{16}$	Processor mode register 1(PM1)
0006$_{16}$	System clock control register 0 (CM0)
0007$_{16}$	System clock control register 1 (CM1)
0008$_{16}$	
0009$_{16}$	Address match interrupt enable register (AIER)
000A$_{16}$	Protect register (PRCR)
000B$_{16}$	
000C$_{16}$	Oscillation stop detection register (CM2)
000D$_{16}$	
000E$_{16}$	Watchdog timer start register (WDTS)
000F$_{16}$	Watchdog timer control register (WDC)
0010$_{16}$	Address match interrupt register 0 (RMAD0)
0011$_{16}$	
0012$_{16}$	
0013$_{16}$	
0014$_{16}$	Address match interrupt register 1 (RMAD1)
0015$_{16}$	
0016$_{16}$	
0017$_{16}$	
0018$_{16}$	
0019$_{16}$	
001A$_{16}$	
001B$_{16}$	
001C$_{16}$	
001D$_{16}$	
001E$_{16}$	INT0 input filter select register (INT0F)
001F$_{16}$	
0020$_{16}$	
0021$_{16}$	
0022$_{16}$	
0023$_{16}$	
0024$_{16}$	
0025$_{16}$	
0026$_{16}$	
0027$_{16}$	
0028$_{16}$	
0029$_{16}$	
002A$_{16}$	
002B$_{16}$	
002C$_{16}$	
002D$_{16}$	
002E$_{16}$	
002F$_{16}$	
0030$_{16}$	
0031$_{16}$	
0032$_{16}$	
0033$_{16}$	
0034$_{16}$	
0035$_{16}$	
0036$_{16}$	
0037$_{16}$	
0038$_{16}$	
0039$_{16}$	
003A$_{16}$	
003B$_{16}$	
003C$_{16}$	
003D$_{16}$	
003E$_{16}$	
003F$_{16}$	

0040$_{16}$	
0041$_{16}$	
0042$_{16}$	
0043$_{16}$	
0044$_{16}$	
0045$_{16}$	
0046$_{16}$	
0047$_{16}$	
0048$_{16}$	
0049$_{16}$	
004A$_{16}$	
004B$_{16}$	
004C$_{16}$	
004D$_{16}$	Key input interrupt control register (KUPIC)
004E$_{16}$	A-D conversion interrupt control register (ADIC)
004F$_{16}$	
0050$_{16}$	
0051$_{16}$	UART0 transmit interrupt control register (S0TIC)
0052$_{16}$	UART0 receive interrupt control register (S0RIC)
0053$_{16}$	UART1 transmit interrupt control register (S1TIC)
0054$_{16}$	UART1 receive interrupt control register (S1RIC)
0055$_{16}$	Timer 1 interrupt control register (T1IC)
0056$_{16}$	Timer X interrupt control register (TXIC)
0057$_{16}$	Timer Y interrupt control register (TYIC)
0058$_{16}$	Timer Z interrupt control register (TZIC)
0059$_{16}$	CNTR0 interrupt control register (CNTR0IC)
005A$_{16}$	TCIN interrupt control register (TCINIC)
005B$_{16}$	Timer C interrupt control register (TCIC)
005C$_{16}$	INT3 interrupt control register (INT3IC)
005D$_{16}$	INT0 interrupt control register (INT0IC)
005E$_{16}$	INT1 interrupt control register (INT1IC)
005F$_{16}$	INT2 interrupt control register (INT2IC)

Figure 1.6.1. Location of peripheral unit control registers (1)

0080_{16}	Timer Y, Z mode register (TYZMR)	$00C0_{16}$	A-D register (AD)
0081_{16}	Prescaler Y (PREY)	$00C1_{16}$	
0082_{16}	Timer Y secondary (TYSC)	$00C2_{16}$	
0083_{16}	Timer Y primary (TYPR)	$00C3_{16}$	
0084_{16}	Timer Y, Z waveform output control register (PUM)	$00C4_{16}$	
0085_{16}	Prescaler Z (PREZ)	$00C5_{16}$	
0086_{16}	Timer Z secondary (TZSC)	$00C6_{16}$	
0087_{16}	Timer Z primary (TZPR)	$00C7_{16}$	
0088_{16}	Prescaler 1 (PRE1)	$00C8_{16}$	
0089_{16}	Timer 1 (T1)	$00C9_{16}$	
$008A_{16}$	Timer Y, Z output control register (TYZOC)	$00CA_{16}$	
$008B_{16}$	Timer X mode register (TXMR)	$00CB_{16}$	
$008C_{16}$	Prescaler X (PREX)	$00CC_{16}$	
$008D_{16}$	Timer X (TX)	$00CD_{16}$	
$008E_{16}$	Timer count source set register (TCSS)	$00CE_{16}$	
$008F_{16}$	Clock prescaler reset flag (CPSRF)	$00CF_{16}$	
0090_{16}	Timer C counter (TC)	$00D0_{16}$	
0091_{16}		$00D1_{16}$	
0092_{16}		$00D2_{16}$	
0093_{16}		$00D3_{16}$	
0094_{16}		$00D4_{16}$	A-D control register 2 (ADCON2)
0095_{16}		$00D5_{16}$	
0096_{16}	External input enable register (INTEN)	$00D6_{16}$	A-D control register 0 (ADCON0)
0097_{16}		$00D7_{16}$	A-D control register 1 (ADCON1)
0098_{16}	Key input enable register (KIEN)	$00D8_{16}$	D-A register (DA)
0099_{16}		$00D9_{16}$	
$009A_{16}$	Timer C control register 0 (TCC0)	$00DA_{16}$	
$009B_{16}$	Timer C control register 1 (TCC1)	$00DB_{16}$	
$009C_{16}$	Time measurement register (TM)	$00DC_{16}$	D-A control register (DACON)
$009D_{16}$		$00DD_{16}$	
$009E_{16}$		$00DE_{16}$	
$009F_{16}$		$00DF_{16}$	
$00A0_{16}$	UART0 transmit/receive mode register (U0MR)	$00E0_{16}$	Port P0 (P0)
$00A1_{16}$	UART0 bit rate generator (U0BRG)	$00E1_{16}$	Port P1 (P1)
$00A2_{16}$	UART0 transmit buffer register (U0TB)	$00E2_{16}$	Port P0 direction register (PD0)
$00A3_{16}$		$00E3_{16}$	Port P1 direction register (PD1)
$00A4_{16}$	UART0 transmit/receive control register 0 (U0C0)	$00E4_{16}$	Port P2 (P2)
$00A5_{16}$	UART0 transmit/receive control register 1 (U0C1)	$00E5_{16}$	Port P3 (P3)
$00A6_{16}$	UART0 receive buffer register (U0RB)	$00E6_{16}$	Port P2 direction register (PD2)
$00A7_{16}$		$00E7_{16}$	Port P3 direction register (PD3)
$00A8_{16}$	UART1 transmit/receive mode register (U1MR)	$00E8_{16}$	Port P4 (P4)
$00A9_{16}$	UART1 bit rate generator (U1BRG)	$00E9_{16}$	
$00AA_{16}$	UART1 transmit buffer register (U1TB)	$00EA_{16}$	Port P4 direction register (PD4)
$00AB_{16}$		$00EB_{16}$	
$00AC_{16}$	UART1 transmit/receive control register 0 (U1C0)	$00EC_{16}$	
$00AD_{16}$	UART1 transmit/receive control register 1 (U1C1)	$00ED_{16}$	
$00AE_{16}$	UART1 receive buffer register (U1RB)	$00EE_{16}$	
$00AF_{16}$		$00EF_{16}$	
$00B0_{16}$	UART transmit/receive control register 2 (UCON)	$00F0_{16}$	
$00B1_{16}$		$00F1_{16}$	
$00B2_{16}$		$00F2_{16}$	
$00B3_{16}$		$00F3_{16}$	
$00B4_{16}$		$00F4_{16}$	
$00B5_{16}$		$00F5_{16}$	
$00B6_{16}$		$00F6_{16}$	
$00B7_{16}$		$00F7_{16}$	
$00B8_{16}$		$00F8_{16}$	
$00B9_{16}$		$00F9_{16}$	
$00BA_{16}$		$00FA_{16}$	
$00BB_{16}$		$00FB_{16}$	
$00BC_{16}$		$00FC_{16}$	Pull-up control register 0 (PUR0)
$00BD_{16}$		$00FD_{16}$	Pull-up control register 1 (PUR1)
$00BE_{16}$		$00FE_{16}$	Port P1 drive capacity control register (DRR)
$00BF_{16}$		$00FF_{16}$	

Figure 1.6.2. Location of peripheral unit control registers (2)

Bus Control

During access, the memory areas (ROM, RAM, FLASH, etc.) and the SFR area have different bus cycles. As shown in Table 1.7.1, memory areas can be accessed in one cycle of the CPU operation clock BCLK. The SFR area can be accessed in two cycles of BCLK.

Table 1.7.1. Bus cycles for access areas

Area	Bus cycle
SFR	2 BCLK cycles
Internal ROM/RAM	1 BCLK cycles

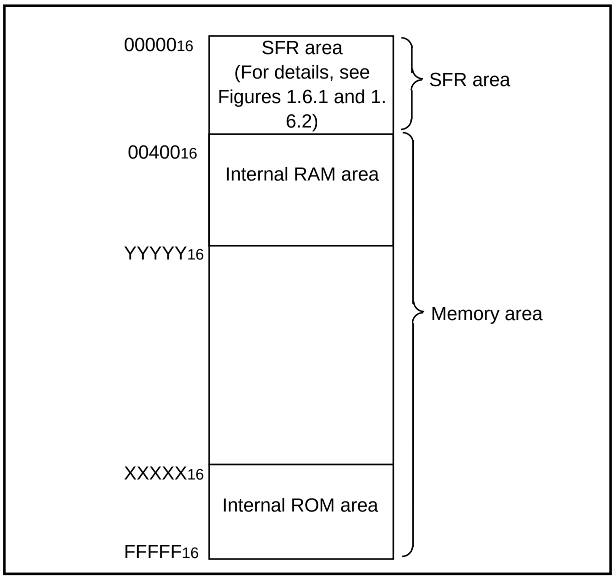
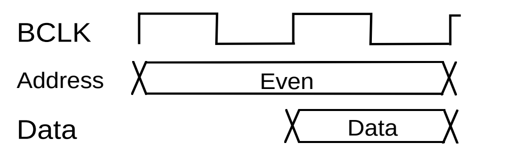
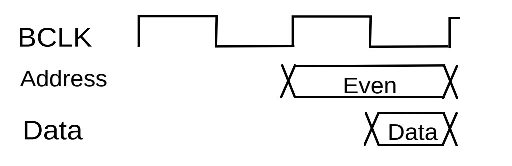
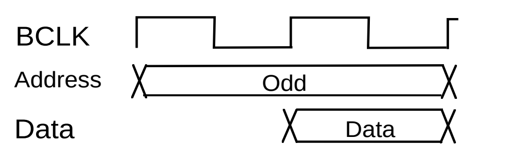
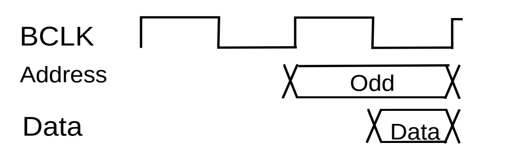
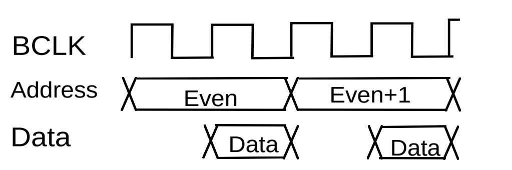
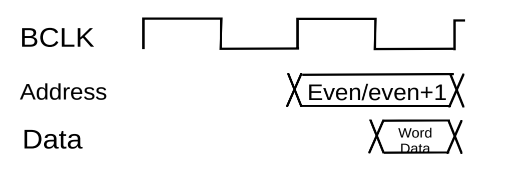
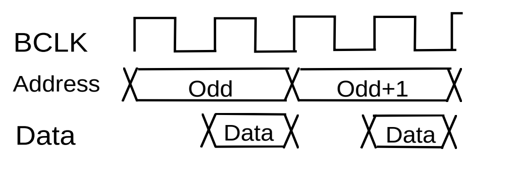
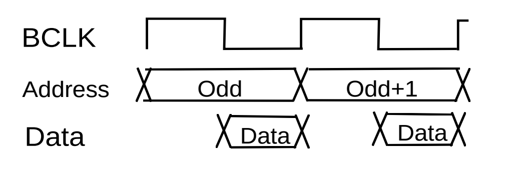


Figure 1.7.1. SFR area and memory areas

The memory areas and the SFR area also have different bus widths. The memory areas have a 16-bit bus width, while the SFR area has an 8-bit bus width. Consequently, different operations are used when the areas are accessed in word (16 bits) units. Table 1.7.2 shows the bus cycles that are necessary to access the SFR area and the memory areas.

Table 1.7.2. Cycles for access areas

Area	SFR area	Memory area
Even address byte access		
Add address byte access		
Even address word access		
Add address word access		

Clock Generating Circuit

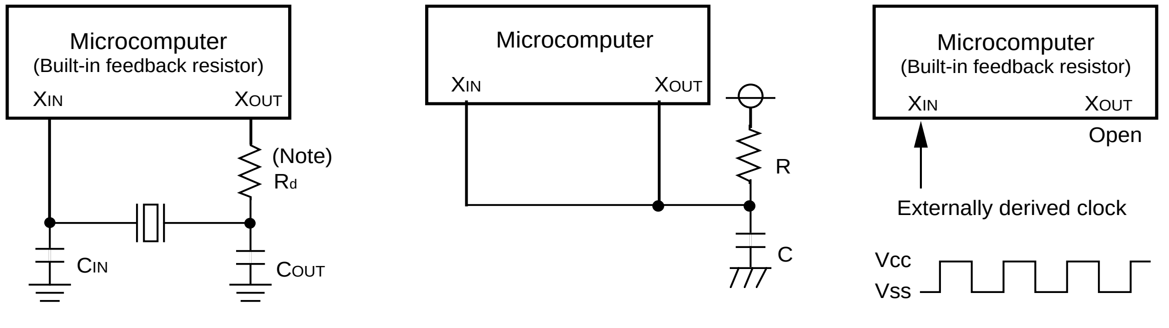
The clock generating circuit contains three oscillator circuits that supply the operating clock sources to the CPU and internal peripheral units.

Table 1.8.1. Main clock and sub-clock generating circuits

	Main clock generating circuit	Sub clock generating circuit	Ring generating circuit
Use of clock	• CPU's operating clock source • Internal peripheral units' operating clock source	• CPU's operating clock source • Timer 1/X/Y/Z/C's count clock source	• CPU's operating clock source • Timer Y/C's count clock source
Usable oscillator	Ceramic, crystal or RC oscillator	Crystal oscillator	–
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT	None (has internal pins)
Oscillation stop/restart function	Available	Available	Available
Oscillator status immediately after reset	Oscillating	Stopped	Oscillating
Other	Externally derived clock can be input		–

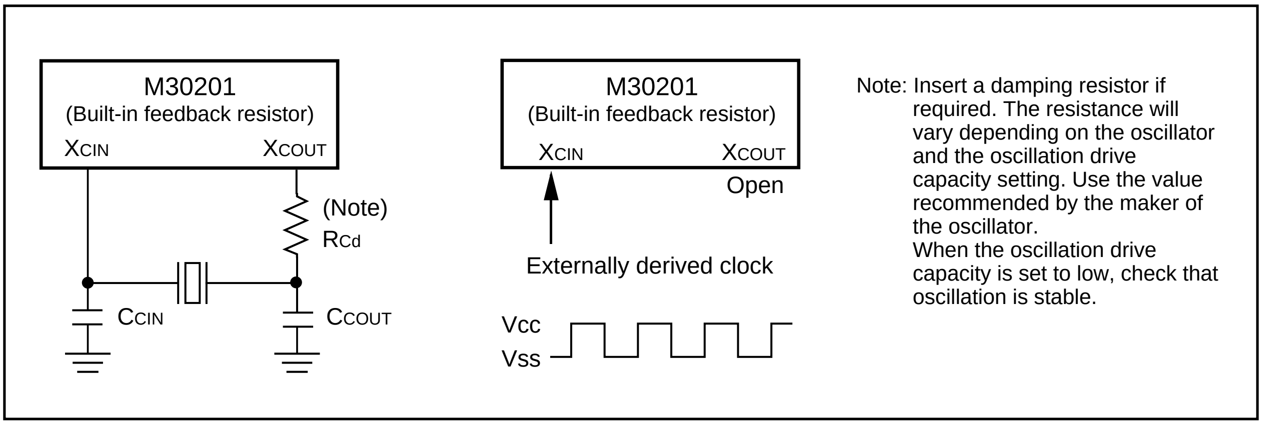
Example of oscillator circuit

Figure 1.8.1 shows some examples of the main clock circuit, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Figure 1.8.2 shows some examples of sub-clock circuits, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Circuit constants in Figures 1.8.1 and 1.8.2 vary with each oscillator used. Use the values recommended by the manufacturer of your oscillator.



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

Figure 1.8.1. Examples of main clock



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

Figure 1.8.2. Examples of sub-clock

A ring oscillator is built into the microcomputer. This ring oscillator can be used as the main clock in place of XIN by setting bit 2 of the oscillation stop detection register. The ring oscillator can also be used in a standby unit that only checks port values. In this case, lower power consumption can be realized because the oscillating frequency of the ring oscillator is much lower compared to that of XIN.

Clock Control

Figure 1.8.3 shows the block diagram of the clock generating circuit.

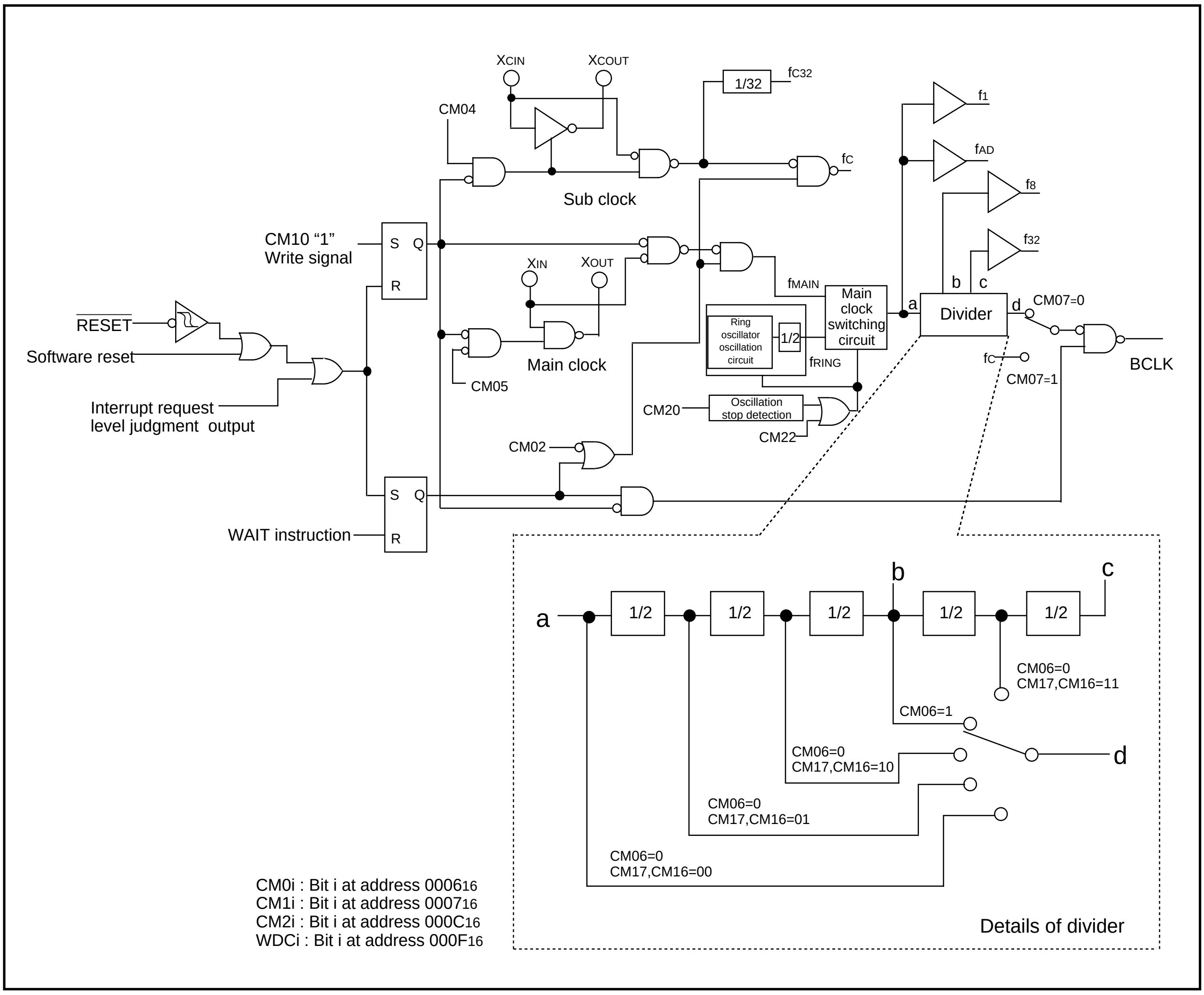


Figure 1.8.3. Clock generating circuit

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

(1) Main clock

The main clock is generated by the main clock oscillation circuit. Immediately after reset, only oscillation starts. The clock can be stopped using the main clock stop bit (bit 5 at address 0006_{16}). Stopping the clock reduces the power dissipation.

After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the XOUT pin can be reduced using the XIN-XOUT drive capacity select bit (bit 5 at address 0007_{16}). Reducing the drive capacity of the XOUT pin reduces the power dissipation. This bit 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 XCOUT pin can be reduced using the XCIN-XCOUT drive capacity select bit (bit 3 at address 0006_{16}). Reducing the drive capacity of the XCOUT pin reduces the power dissipation. This bit changes to “1” when shifting to stop mode and at a reset.

(3) BCLK

The BCLK is the clock that drives the CPU, and is fc, or the clock is derived from the main clock or by dividing it by 2, 4, 8, or 16. After reset, the BCLK is derived by dividing the clock supplied by the ring oscillator circuit 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 mode 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

a. f_1, f_8, f_{32}

The clock for the peripheral devices is derived from the main clock or by dividing it by 8 or 32. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at 0006_{16}) to “1” and then executing a WAIT instruction.

b. f_{AD}

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

(5) f_{C32}

This clock is derived by dividing the sub-clock by 32. It is used for the timer 1, timer X, timer Y and timer Z counts.

(6) f_C

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

(7) f_{RING}

This clock is supplied by the ring oscillator circuit. In the ring oscillator mode, the clock divided by the division ratio selected with the main clock division select bit 0 and bit 1(bit 6 at address 0006_{16}, and bit 6 and bit 7 at address 0007_{16}) is supplied as BCLK. Immediately after reset, 8 divisions of this clock is supplied as BCLK. The ring oscillator oscillation can be set to BCLK when oscillation stop is detected or with the main clock switching bit (bit 2 at address $000C_{16}$).

System clock control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

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

Bit symbol	Bit name	Function	R	W
Reserved bit		Always set to “0”	○	○
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)	○	○
CM03	XCIN-XCOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	○	○
CM04	Port XC select bit	0 : I/O port 1 : XCIN-XCOUT generation	○	○
CM05	Main clock (XIN-XOUT) stop bit (Note 3,4,5)	0 : On 1 : Off	○	○
CM06	Main clock division select bit 0 (Note 7)	0 : CM16 and CM17 valid 1 : Division by 8 mode	○	○
CM07	System clock select bit (Note 6)	0 : XIN, XOUT 1 : XCIN, XCOUT	○	○

Note 1: Set bit 0 of the protect register (address $000A_{16}$) to “1” before writing to this register.
Note 2: Changes to “1” when shifting to stop mode.
Note 3: This bit is used to stop the main clock when placing the device in a low-power mode. If you want to operate with 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 ON, so XIN turns pulled up to XOUT (“H”) via the feedback resistor.
Note 6: Set port Xc select bit (CM04) to “1” before setting this bit to “1”. Can not write to both bits at the same time.
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: fc_{32} is not included. Do not set to “1” when using low-speed or low power dissipation mode.

System clock control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
					0	0	

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

Bit symbol	Bit name	Function	R	W
CM10	All clock stop control bit (Note 4)	0 : Clock on 1 : All clocks off (stop mode)	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○
CM13	XIN oscillation select bit	0 : Ceramic oscillation or crystal oscillation 1 : RC oscillation	○	○
CM14	Ring oscillation stop bit	0 : Oscillation enabled 1 : Oscillation stopped (Note 5)	○	○
CM15	XIN-XOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	○	○
CM16	Main clock division select bit 1 (Note 3)	b7 b6 0 0 : No division mode 0 1 : Division by 2 mode 1 0 : Division by 4 mode 1 1 : Division by 16 mode	○	○
CM17				

Note 1: Set bit 0 of the protect register (address $000A_{16}$) to “1” before writing to this register.
Note 2: This bit changes to “1” when shifting from high-speed/middle-speed mode to stop mode or at 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 0006_{16}) 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 ineffective.
Note 5: This bit can be set to “1” only when both the main clock switch bit (CM22) and clock monitor bit (CM23) are set to “0” . Moreover, this bit is automatically set to “0” if the main clock switch bit (CM22) is set to “1”.

Figure 1.8.4. System clock control registers 0 and 1

Oscillation stop detection register (Note 5)

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

Symbol	Address	When reset
CM2	$000C_{16}$	04_{16}

Bit symbol	Bit name	Function	R	W
CM20	Oscillation stop detection bit	0: The function is invalid. 1: The function is valid. (Note 1)	○	○
CM21	Oscillation stop detection interrupt enable bit	0: Disabled 1: Enabled (Note 2)	○	○
CM22	Main clock switch bit	0: Select XIN clock. 1: Select ring oscillator. (Note 3)	○	○
CM23	Clock monitor bit (Note 4)	0: XIN oscillating normally 1: XIN stopping abnormally	○	×
Reserved bit		Always set to "0"	○	○

Note 1: Set to “0” before stopping the oscillation of the main clock (XIN-XOUT).
(stop mode, low power dissipation mode, ring oscillation mode)
An oscillation stop is detected if the oscillation of the main clock (XIN-XOUT) is stopped when the following two conditions are satisfied: (1) the oscillation stop detection function is valid and (2) CM21=1.
Note 2: Valid when CM20=1.
Note 3: CM22 bit switches to “1“ automatically if an oscillation stop is detected when both CM20 bit and CM 21 bit are “1”. CM22 bit cannot be cleared when CM23=1.
Note 4: This bit is valid when both CM20 bit and CM 21 bit are “1”. Use this bit for the purpose of confirming XIN operation for oscillation stop detection interrupt execution.
Note 5: In case of writing to this register, set bit 0 of the protect register($000A_{16}$) to "1".

Clock prescaler reset flag

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

Symbol	Address	When reset
CPSRF	$008F_{16}$	$0XXXXXXX_2$

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 1.8.5. Oscillation stop detection register and clock prescaler reset flag

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 X operate provided that the event counter mode is set to an external pulse, and UART0 and UART1 function provided an external clock is selected. Table 1.8.2 shows the status of the ports in stop mode.

Stop mode is cancelled by a hardware reset or an interrupt. If an interrupt is to be used to cancel stop mode, that interrupt must first have been enabled, and the priority level of the interrupt which is not used to cancel must have been changed to 0 before shifting to stop mode. If returning by an interrupt, that interrupt routine is executed. If only a hardware reset is used to cancel stop mode, change the priority level of all interrupt to 0, then shift to stop mode.

When shifting from high-speed/medium-speed mode to stop mode or 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 1.8.2. Port status during stop mode

Pin	States
Port	Retains status before stop mode

Wait Mode

When a WAIT instruction is executed, BCLK stops and the microcomputer enters the wait mode. In this mode, oscillation continues but BCLK and watchdog timer stop. Writing “1” to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. Table 1.8.3 shows the status of the ports in wait mode.

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

Table 1.8.3. Port status during wait mode

Pin	States
Port	Retains status before wait mode

Status Transition of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. Table 1.8.4 shows the operating modes corresponding to the settings of system clock control registers 0 and 1.

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

(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. 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 dissipation mode, 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.

(8) Ring oscillator mode

This mode sets the ring oscillator as BCLK. The same as when X_{IN} is the main clock, the modes are no division, 2-division, 4-division, 8-division, and 16-division.

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 1.8.4. Operating modes dictated by settings of system clock control registers 0 and 1

CM22	CM17	CM16	CM07	CM06	CM05	CM04	Operating mode of BCLK
0	0	1	0	0	0	Invalid	Division by 2 mode
0	0	0	0	0	0	Invalid	Division by 4 mode
0	Invalid	Invalid	0	1	0	Invalid	Division by 8 mode
0	1	1	0	0	0	Invalid	Division by 16 mode
0	0	0	0	0	0	Invalid	No-division mode
0	Invalid	Invalid	1	Invalid	0	1	Low-speed mode
0	Invalid	Invalid	1	Invalid	1	1	Low power dissipation mode
1	0	1	0	0	Invalid	Invalid	Ring oscillator mode(divided by 2)
1	0	0	0	0	Invalid	Invalid	Ring oscillator mode(divided by 4)
1	Invalid	Invalid	0	1	Invalid	Invalid	Ring oscillator mode(divided by 8)
1	1	1	0	0	Invalid	Invalid	Ring oscillator mode(divided by 16)
1	0	0	0	0	Invalid	Invalid	Ring oscillator mode(no division)

Power Control

This section gives an overview of power control.

Modes

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.

- Ring oscillator mode

This mode sets the ring oscillator as BCLK. The ring oscillator can be set to no division, 2-divisions, 4-division, 8-division, or 16-division mode according to the settings for CM06, CM16, and CM17. Increasing the division ratio lowers power consumption. When the microcomputer is operating with the ring oscillator, the XIN clock driver can be stopped by setting the main clock stop bit to “1.” This can lower the power dissipation even more.

(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 1.9.1 and 1.9.2 show the transition between each of the three modes, (1), (2), and (3).

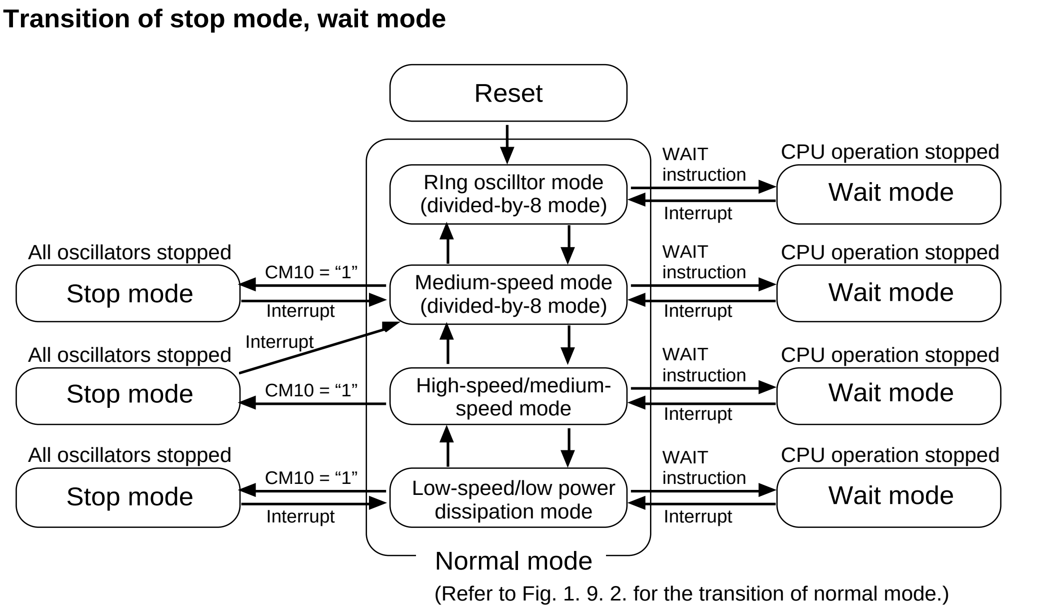
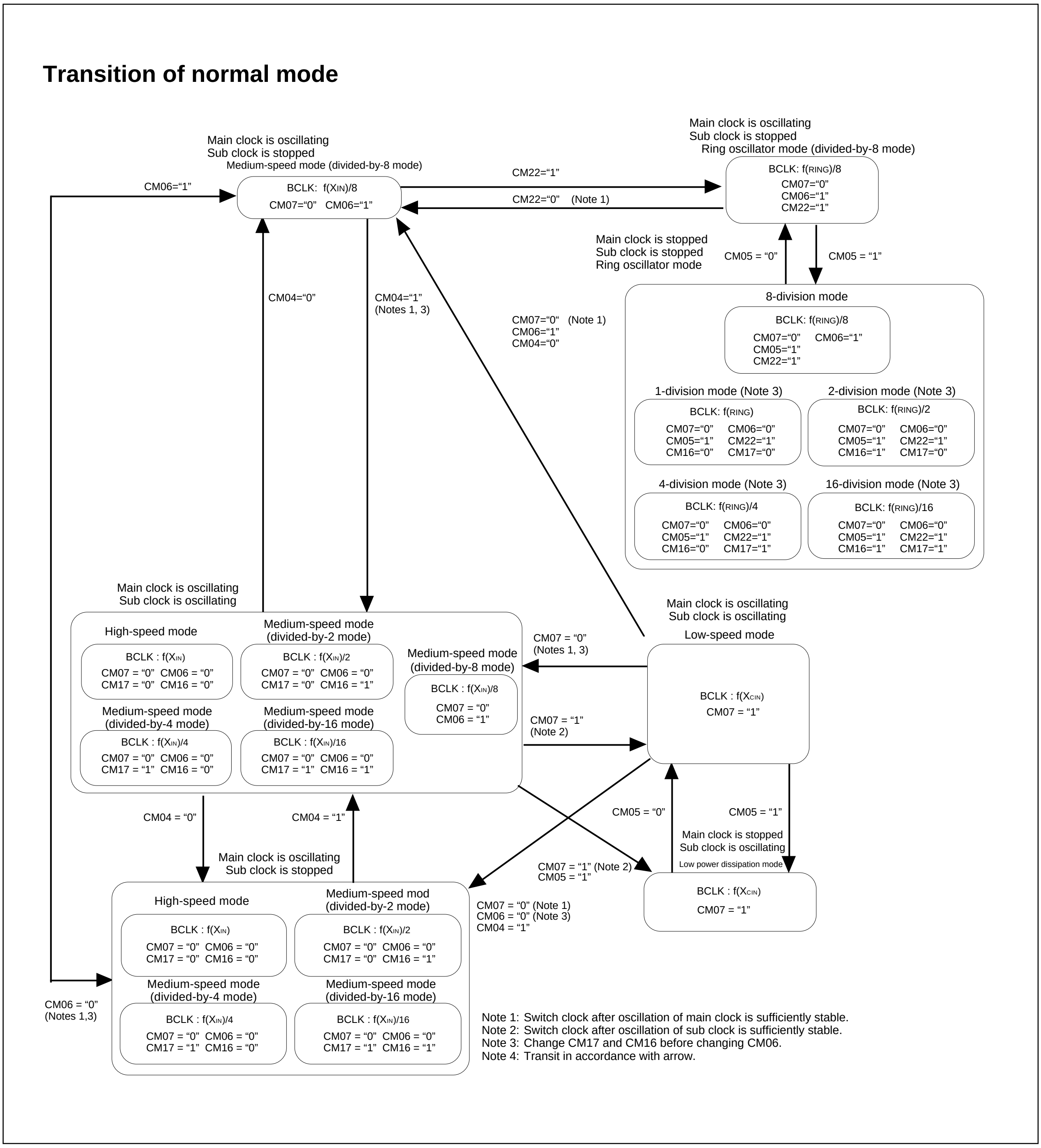

Transition of stop mode, wait mode
Reset
RIng oscilltor mode (divided-by-8 mode)
Medium-speed mode (divided-by-8 mode)
High-speed/medium-speed mode
Low-speed/low power dissipation mode
Normal mode
(Refer to Fig. 1. 9. 2. for the transition of normal mode.)
All oscillators stopped
Stop mode
CM10 = "1"
Interrupt
WAIT instruction
CPU operation stopped
Wait mode

Figure 1.9.1. Clock transition (1)



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 CM17 and CM16 before changing CM06.
Note 4: Transit in accordance with arrow.

Figure 1.9.2. Clock transition (2)

Oscillation Stop Detection Function

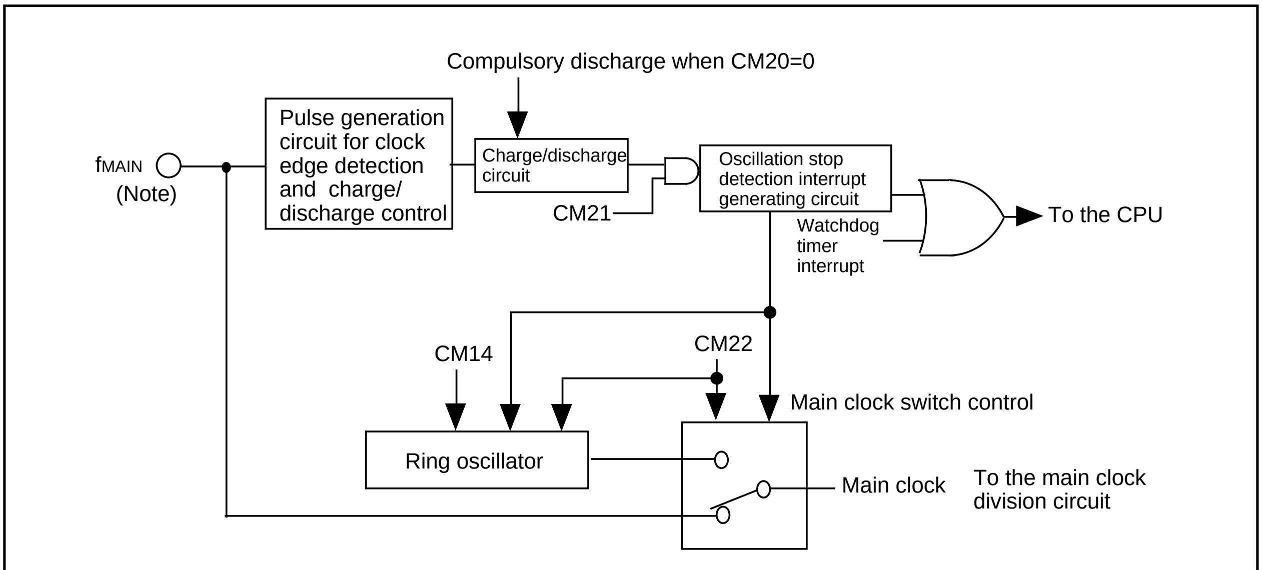
The oscillation stop detection function detects abnormal stopping of the main clock by causes such as opening and shorting of the X_{IN} oscillation circuit. When oscillation stop is detected, an oscillation stop detection interrupt is issued. When an oscillation stop detection interrupt is issued, the ring oscillator in the microcomputer operates automatically and is used as the main clock in place of the X_{IN} clock. This allows interrupt processing.

The oscillation stop detection function can be enabled/disabled with bit 0 and bit 1 of the oscillation stop detection register. When this bit is set to "11_2," the function is enabled. After the reset is released, the oscillation stop detection function becomes disabled because the bit value is "00."

Table 1.10.1 gives an specification overview of the oscillation stop detection function, Figure 1.10.2 is a configuration diagram of the oscillation stop detection circuit and Figure 1.10.3 shows the configuration of the oscillation stop detection register.

Table 1.10.1. Specification overview of the oscillation stop detection function

Item	Specification
Oscillation stop detectable clock and frequency bandwidth	$X_{IN} \geq 2$ MHz
Enabling condition for oscillation stop detection function	When the oscillation stop detection bit (bit 0 of address $000C_{16}$) and the oscillation stop detection interrupt enable bit (bit 1 of address $000C_{16}$) are set to "1"
Operation at oscillation stop detection	• Oscillation stop detection interrupt occurs
Notes on STOP mode, low power dissipation mode, and ring oscillator mode	Before stopping the main clock (X_{IN}-X_{OUT}), set the oscillation stop detection enable bit to "0" to disable the oscillation stop detection function. Enable main clock (X_{IN}-X_{OUT}) oscillation and after the oscillation stabilizes, set the bit to "1" again.
Notes on WAIT mode	If the peripheral function clock is stopped in WAIT mode with the WAIT mode peripheral function clock stop bit (bit 2 of the address 0006_{16}), oscillation stop will be detected. Do not stop the peripheral function clock in WAIT mode.


Compulsory discharge when CM20=0
fMAIN
(Note)
Pulse generation circuit for clock edge detection and charge/ discharge control
Charge/discharge circuit
CM21
Oscillation stop detection interrupt generating circuit
Watchdog timer interrupt
To the CPU
CM14
CM22
Main clock switch control
Ring oscillator
Main clock
To the main clock division circuit

#:When XIN is supplied, this repeats charge and discharge with pulses by XIN edge detection.
When XIN is not supplied, this continues charging. When the charge exceeds a certain level, it regards the oscillation as stopped.

Note: As for the fMAIN, refer to Figure 1.8.3 clock generating circuit.

Figure 1.10.1. Oscillation stop detection circuit

Oscillation stop detection register (Note 5)

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

Symbol	Address	When reset
CM2	000C$_{16}$	04$_{16}$

Bit symbol	Bit name	Function	R	W
CM20	Oscillation stop detection bit	0: The function is invalid. 1: The function is valid. (Note 1)	○	○
CM21	Oscillation stop detection interrupt enable bit	0: Disabled 1: Enabled (Note 2)	○	○
CM22	Main clock switch bit	0: Select XIN clock. 1: Select ring oscillator. (Note 3)	○	○
CM23	Clock monitor bit (Note 4)	0: XIN oscillating normally 1: XIN stopping abnormally	○	×
Reserved bit		Always set to "0"	○	○

Note 1: Set to “0” before stopping the oscillation of the main clock (XIN-XOUT).
(stop mode, low power dissipation mode, ring oscillation mode)
An oscillation stop is detected if the oscillation of the main clock (XIN-XOUT) is stopped when the following two conditions are satisfied: (1) the oscillation stop detection function is valid and (2) CM21=1.
Note 2: Valid when CM20=1.
Note 3: CM22 bit switches to “1“ automatically if an oscillation stop is detected when both CM20 bit and CM 21 bit are “1”. CM22 bit cannot be cleared when CM23=1.
Note 4: This bit is valid when both CM20 bit and CM 21 bit are “1”. Use this bit for the purpose of confirming XIN operation for oscillation stop detection interrupt execution.
Note 5: In case of writing to this register, set bit 0 of the protect register(000A16) to "1".

Figure 1.10.2. Oscillation stop detection register

Oscillation stop detection bit (CM20)

You can start the oscillation stop detection by setting this bit to "1" and CM21=1 (oscillation stop detection interrupt enabled). The detection is not executed when this bit is set to "0" or in reset status. Be sure to set this bit to "0" before setting for the stop-mode. Set this bit again to "1" after release from stop-mode. Set this bit to "0" also before setting the main clock stop bit (bit 5 at 0006_{16}) to "1".
Do not set this bit to "1" if the frequency of X_{IN} is lower than 2 MHz.
An oscillation stop is detected if CM02="1" (peripheral function clock has been set for stop in wait mode) and the mode is shifted to wait.

Oscillation stop detection interrupt enable bit (CM21)

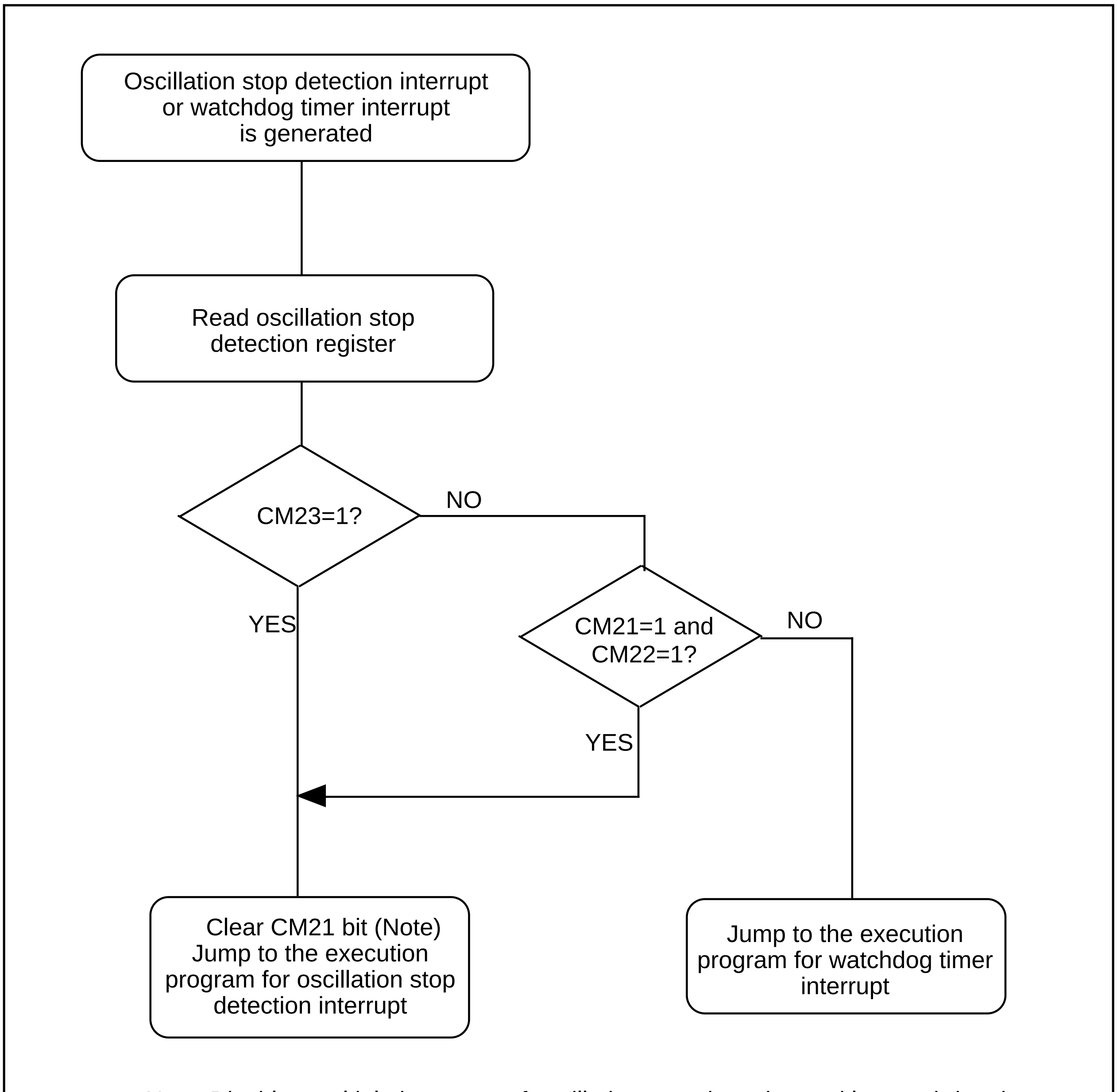
When CM20=1 and CM21=1, an oscillation stop detection interrupt is generated if an abnormal stop of X_{IN} is detected. The ring oscillator starts operation instead of the X_{IN} clock which stopped abnormally. The operation goes further with the main clock supplied from the ring oscillator. For the oscillation stop detection interrupt, judgment on the interrupt condition is necessary, because this interrupt shares the vector table with watchdog timer interrupt. Figure 1.10.3 shows flow of the judgment with oscillation stop detection interrupt processing program.

Main clock switch bit (CM22)

When setting this bit to "1", the ring oscillator is selected as main clock. At this time, the ring oscillator starts simultaneously if it has been stopped (CM14=1). This bit is cleared only when CM23 is "0" (when X_{IN} is oscillating).
If an oscillation stop is detected while both CM20 and CM21 are “1”, this bit automatically switches to “1”. When this bit is set to “1”, the ring oscillation stop bit (bit 4 of address 0007_{16}) is automatically set to “0”.

Clock monitor bit (CM23)

You can see the operation status of the X_{IN} clock. When this bit is "0", X_{IN} is operating correctly. You can check the oscillation status of X_{IN} when an oscillation stop detection interrupt is generated or after reset. When oscillation stop detection is invalid (CM20=“0”), the clock monitor bit is “0”.


Oscillation stop detection interrupt or watchdog timer interrupt is generated
Read oscillation stop detection register
CM23=1?
NO
YES
CM21=1 and CM22=1?
NO
YES
Clear CM21 bit (Note) Jump to the execution program for oscillation stop detection interrupt
Jump to the execution program for watchdog timer interrupt

Note: Disables multiple interrupts of oscillation stop detection and let watchdog timer take priority.

Figure 1.10.3. Flow of the judgment

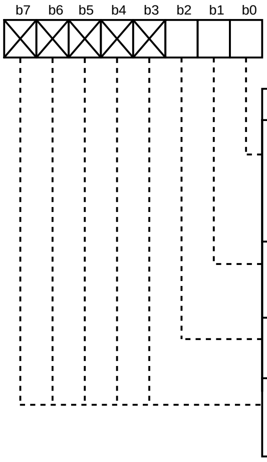
Protection

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 1.11.1 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 P0 direction register (address $00E2_{16}$) can only be changed when the respective bit in the protect register is set to “1”. Therefore, important outputs can be allocated to port P0.

If, after "1" (write-enabled) has been written to bit "enables writing to port P0 direction register" (bit 2 at address $000A_{16}$), a value is written to any address, the bit automatically reverts to "0" (write-inhibited).

The system clock control registers 0 and 1 and oscillation stop detection register write-enable bit (bit 0 at $000A_{16}$) and processor mode register 0 and 1 write-enable bit (bit 1 at $000A_{16}$) do not automatically return to “0” after a value has been written to an address. The program must therefore be written to return these bits to “0”.

Protect register



Symbol	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}) and oscillation stop detection register (address $000C_{16}$)	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 P0 direction register (address $00E2_{16}$) (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.			—	—

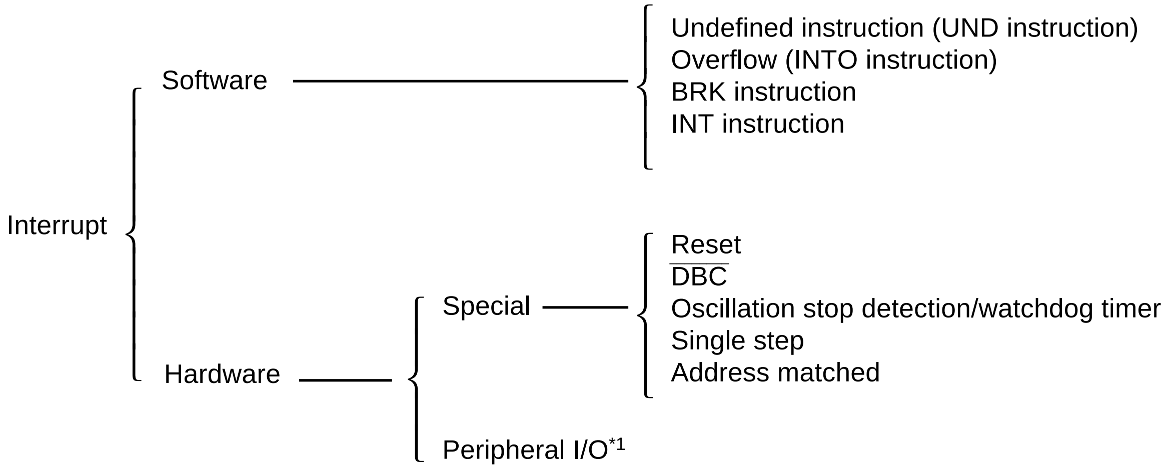
Note: Writing a value to an address after “1” is written to this bit returns the bit to “0” . Other bits do not automatically return to “0” and they must therefore be reset by the program.

Figure 1.11.1. Protect register

Overview of Interrupt

Type of Interrupts

Figure 1.12.1 lists the types of interrupts.



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

Figure 1.12.1. Classification of interrupts

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

Software Interrupts

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

• Undefined instruction interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

• Overflow interrupt

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

• BRK interrupt

A BRK interrupt occurs when executing the BRK instruction.

• INT interrupt

An INT interrupt occurs when 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 $\overline{\text{RESET}}$ pin.
- **UART1 receive interrupt**
  UART1 receive interrupt occurs when UART1 is received. This interrupt can be enabled with bit 2 of the INT0 input filter select register (address $001E_{16}$).
  This interrupt is exclusively for the debugger, do not use it in other circumstances.
- **$\overline{\text{DBC}}$ interrupt**
  This interrupt is exclusively for the debugger, do not use it in other circumstances.
- **Oscillation stop detection/watchdog timer interrupt**
  Generated by the oscillation stop detection or 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

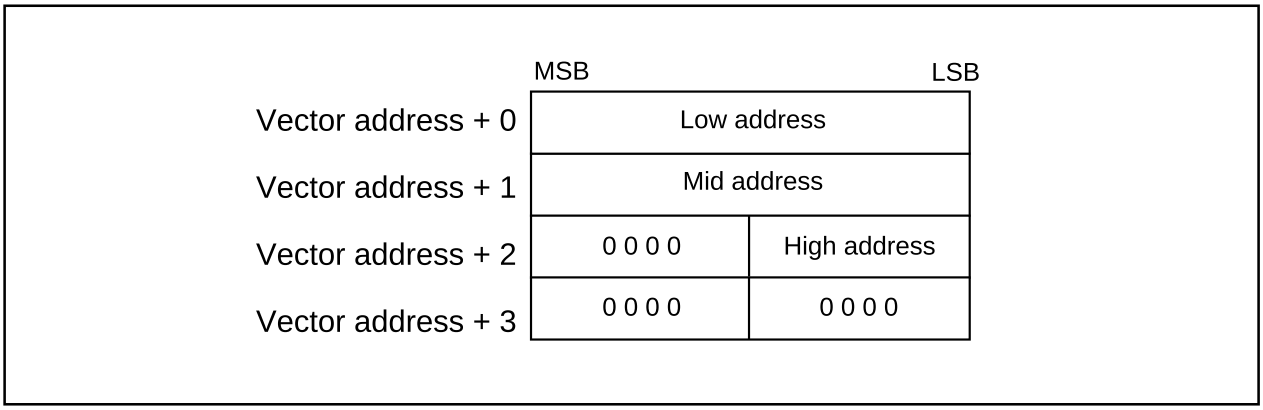
A peripheral I/O interrupt is generated by one of built-in peripheral functions. The interrupt vector table is the same as the one for software interrupt numbers 0 through 31 the INT instruction uses. Peripheral I/O interrupts are maskable interrupts.

- **Key-input interrupt**
  A key-input interrupt occurs if an “L” is input to the $\overline{\text{KI}}$ pin. Polarity can be switched.
- **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 X interrupt**
  This is an interrupts that timer X generates.
- **Timer Y interrupt**
  This is an interrupt that timer Y generates.
- **Timer Z interrupt**
  This is an interrupt that timer Z generates.
- **Timer C interrupt**
  This is an interrupt that timer C generates.
- **CNTR0 and TCIN interrupt**
  These interrupt occur if a falling edge is input to the CNTR0 pin and TCIN pin, respectively.
- **$\overline{\text{INT0}}$ to $\overline{\text{INT3}}$ interrupt**
  An $\overline{\text{INT}}$ interrupt occurs if any one of a rising edge, a falling edge or a both-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 1.12.2 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.



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

Figure 1.12.2. Format for specifying interrupt vector addresses

• Fixed vector tables

The fixed vector table is a table in which addresses are fixed. The vector tables are located in an area extending from FFFDC$_{16}$ to FFFFF$_{16}$. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table 1.12.1 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

Table 1.12.1. Interrupt and fixed vector address

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFDC$_{16}$ to FFFDF$_{16}$	Interrupt on UND instruction
Overflow	FFFE0$_{16}$ to FFFE3$_{16}$	Interrupt on INTO instruction
BRK instruction	FFFE4$_{16}$ to FFFE7$_{16}$	If the vector is filled with FF$_{16}$, program execution starts from the address shown by the vector in the variable vector table
Address match	FFFE8$_{16}$ to FFFEB$_{16}$	There is an address-matching interrupt enable bit
Single step (Note)	FFFEC$_{16}$ to FFFEF$_{16}$	Do not use
Oscillation stop detection/ watchdog timer	FFFF0$_{16}$ to FFFF3$_{16}$	
$\overline{\text{DBC}}$ (Note)	FFFF4$_{16}$ to FFFF7$_{16}$	Do not use
UART1 receive (Note)	FFFF8$_{16}$ to FFFFB$_{16}$	Do not use
Reset	FFFFC$_{16}$ to FFFFF$_{16}$	

Note: Interrupts used for debugging purposes only.

• Variable vector tables

The addresses in the variable vector table can be modified, according to the user’s settings. Indicate the first address using the interrupt table register (INTB). The 256-byte area subsequent to the address the INTB indicates becomes the area for the variable vector tables. One vector table comprises four bytes. Set the first address of the interrupt routine in each vector table. Table 1.12.2 shows the interrupts assigned to the variable vector tables and addresses of vector tables.

Table 1.12.2. 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 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 1	
Software interrupt number 22	+88 to +91 (Note)	Timer X	
Software interrupt number 23	+92 to +95 (Note)	Timer Y	
Software interrupt number 24	+96 to +99 (Note)	Timer Z	
Software interrupt number 25	+100 to +103 (Note)	CNTR0	
Software interrupt number 26	+104 to +107 (Note)	TCIN	
Software interrupt number 27	+108 to +111 (Note)	Timer C	
Software interrupt number 28	+112 to +115 (Note)	$\overline{\text{INT3}}$	
Software interrupt number 29	+116 to +119 (Note)	$\overline{\text{INT0}}$	
Software interrupt number 30	+120 to +123 (Note)	$\overline{\text{INT1}}$	
Software interrupt number 31	+124 to +127 (Note)	$\overline{\text{INT2}}$	
Software interrupt number 32 to Software interrupt number 63	+128 to +131 (Note) to +252 to +255 (Note)	Software interrupt	Cannot be masked by I flag

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

Interrupt Control

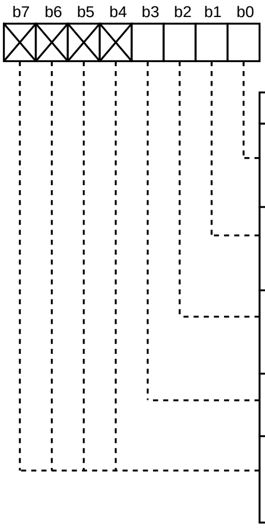
Descriptions are given here regarding how to enable or disable maskable interrupts and how to set the priority to be accepted. What is described here does not apply to non-maskable interrupts.

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 1.12.3 shows the interrupt control registers.

Interrupt control register

Symbol	Address	When reset
KUPIC	$004D_{16}$	$XXXXX000_2$
ADIC	$004E_{16}$	$XXXXX000_2$
SiTIC(i=0, 1)	0051_{16}, 0053_{16}	$XXXXX000_2$
SiRIC(i=0, 1)	0052_{16}, 0054_{16}	$XXXXX000_2$
T1IC	0055_{16}	$XXXXX000_2$
TXIC	0056_{16}	$XXXXX000_2$
TYIC	0057_{16}	$XXXXX000_2$
TZIC	0058_{16}	$XXXXX000_2$
CNTR0IC	0059_{16}	$XXXXX000_2$
TCINIC	$005A_{16}$	$XXXXX000_2$
TCIC	$005B_{16}$	$XXXXX000_2$
INT3IC	$005C_{16}$	$XXXXX000_2$

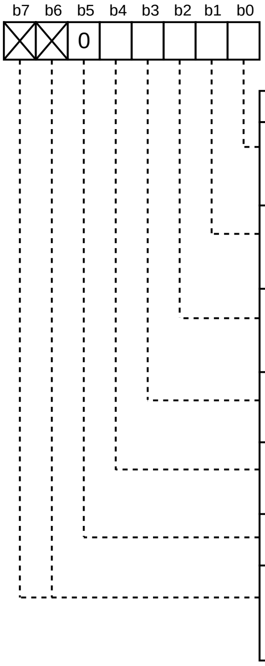


Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note 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.

Symbol	Address	When reset
INTiIC(i=0, 1, 2)	$005D_{16}$, $005E_{16}$	$XX00X000_2$
	$005F_{16}$	$XX00X000_2$



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 1.12.3. Interrupt control register

Interrupt Enable Flag (I Flag)

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

Interrupt Request Bit

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

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

Set the interrupt priority level using the interrupt priority level select bit, which is one of the component bits of the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to “0” disables the interrupt.
Table 1.12.3 shows the settings of interrupt priority levels and Table 1.12.4 shows the interrupt levels enabled, according to the consist of the IPL.

The following are conditions under which an interrupt is accepted:

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

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

Table 1.12.3. Settings of interrupt priority levels

Interrupt priority level select bit	Interrupt priority level	Priority order
b2 b1 b0 0 0 0	Level 0 (interrupt disabled)	—
0 0 1	Level 1	Low
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 1.12.4. Interrupt levels enabled according to the contents of the IPL

IPL	Enabled interrupt priority levels
IPL_2 IPL_1 IPL_0 0 0 0	Interrupt levels 1 and above are enabled
0 0 1	Interrupt levels 2 and above are enabled
0 1 0	Interrupt levels 3 and above are enabled
0 1 1	Interrupt levels 4 and above are enabled
1 0 0	Interrupt levels 5 and above are enabled
1 0 1	Interrupt levels 6 and above are enabled
1 1 0	Interrupt levels 7 and above are enabled
1 1 1	All maskable interrupts are disabled

Rewrite The Interrupt Control Register

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

Example 1:

```
INT_SWITCH1:
    FCLR      I                ; Disable interrupts.
    AND.B     #00h, 0055h      ; Clear T1IC int. priority level and int. request bit.
    NOP                        ;
    NOP
    FSET      I                ; Enable interrupts.
```

Example 2:

```
INT_SWITCH2:
    FCLR      I                ; Disable interrupts.
    AND.B     #00h, 0055h      ; Clear T1IC 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 T1IC int. priority level and int. request bit.
    POPC      FLG              ; Enable interrupts.
```

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

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

Interrupt Sequence

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

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

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

(1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 00000_{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.

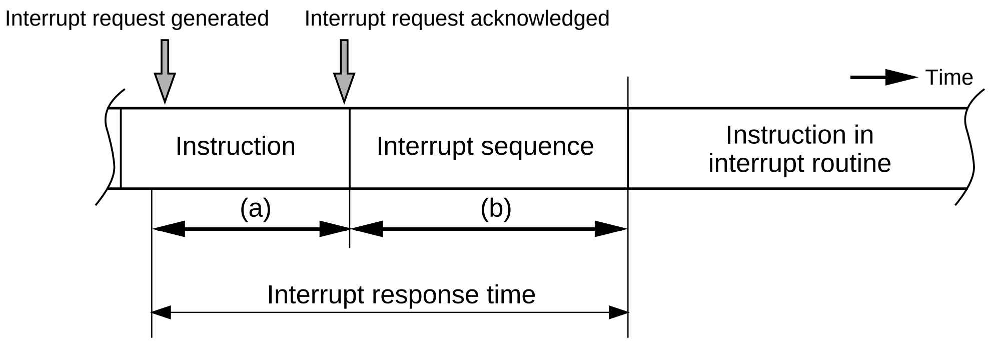
(6) Sets the interrupt priority level of the accepted instruction in the IPL.

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 1.12.4 shows the interrupt response time.


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

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

Figure 1.12.4. Interrupt response time

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

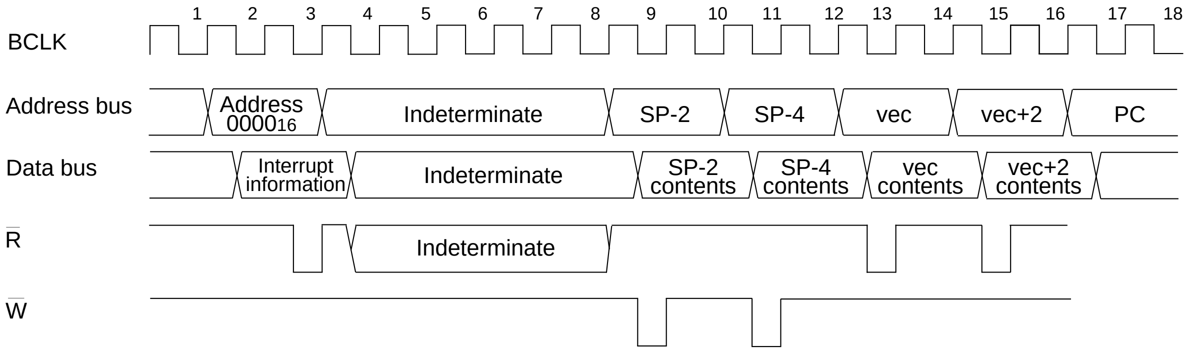
Time (b) is as shown in Table 1.12.5.

Table 1.12.5. Time required for executing the interrupt sequence

Interrupt vector address	Stack pointer (SP) value	Without wait
Even	Even	18 cycles (Note 1)
Even	Odd	19 cycles (Note 1)
Odd (Note 2)	Even	19 cycles (Note 1)
Odd (Note 2)	Odd	20 cycles (Note 1)

Note 1: Add 2 cycles in the case of a $\overline{\text{DBC}}$ interrupt; add 1 cycle in the case either of an address match interrupt or of a single-step interrupt.

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



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

Figure 1.12.5. Time required for executing the interrupt sequence

Variation of IPL when Interrupt Request is Accepted

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

If an interrupt request, that does not have an interrupt priority level, is accepted, one of the values shown in Table 1.12.6 is set in the IPL.

Table 1.12.6. Relationship between interrupts without interrupt priority levels and IPL

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

Saving Registers

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

First, the processor saves the 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 1.12.6 shows the state of the stack as it was before the acceptance of the interrupt request, and the state the stack after the acceptance of the interrupt request.

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

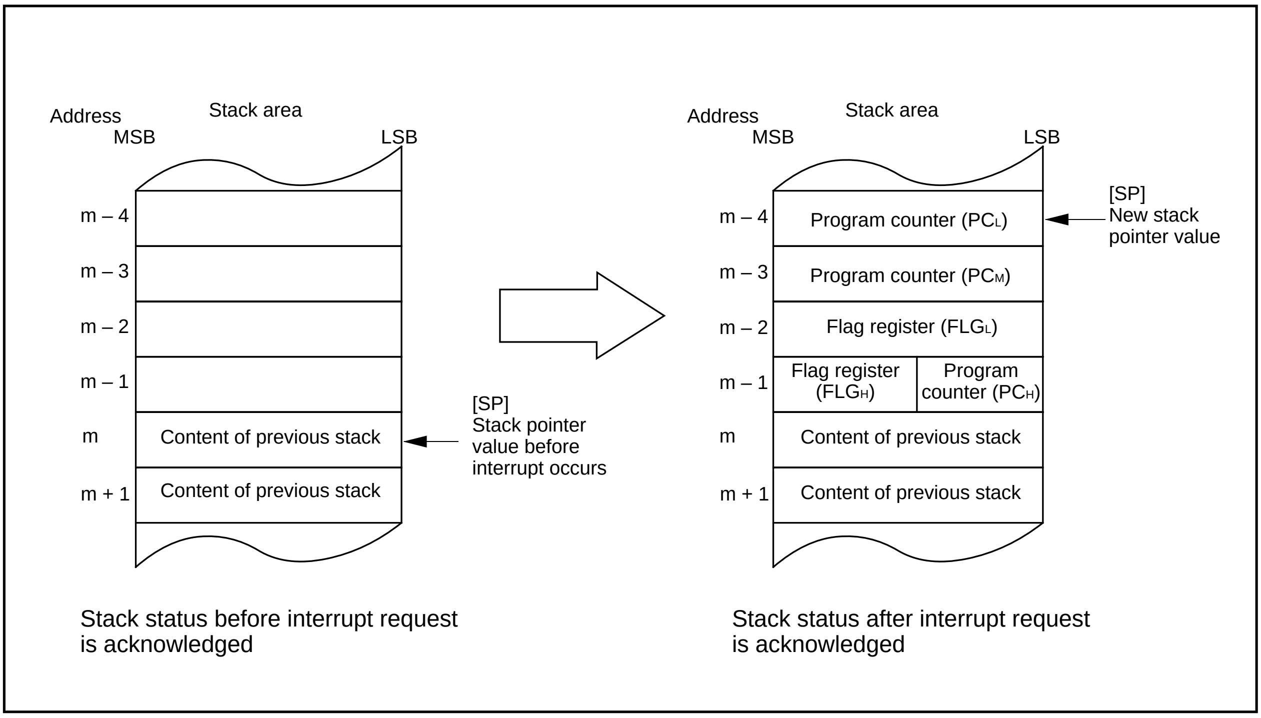
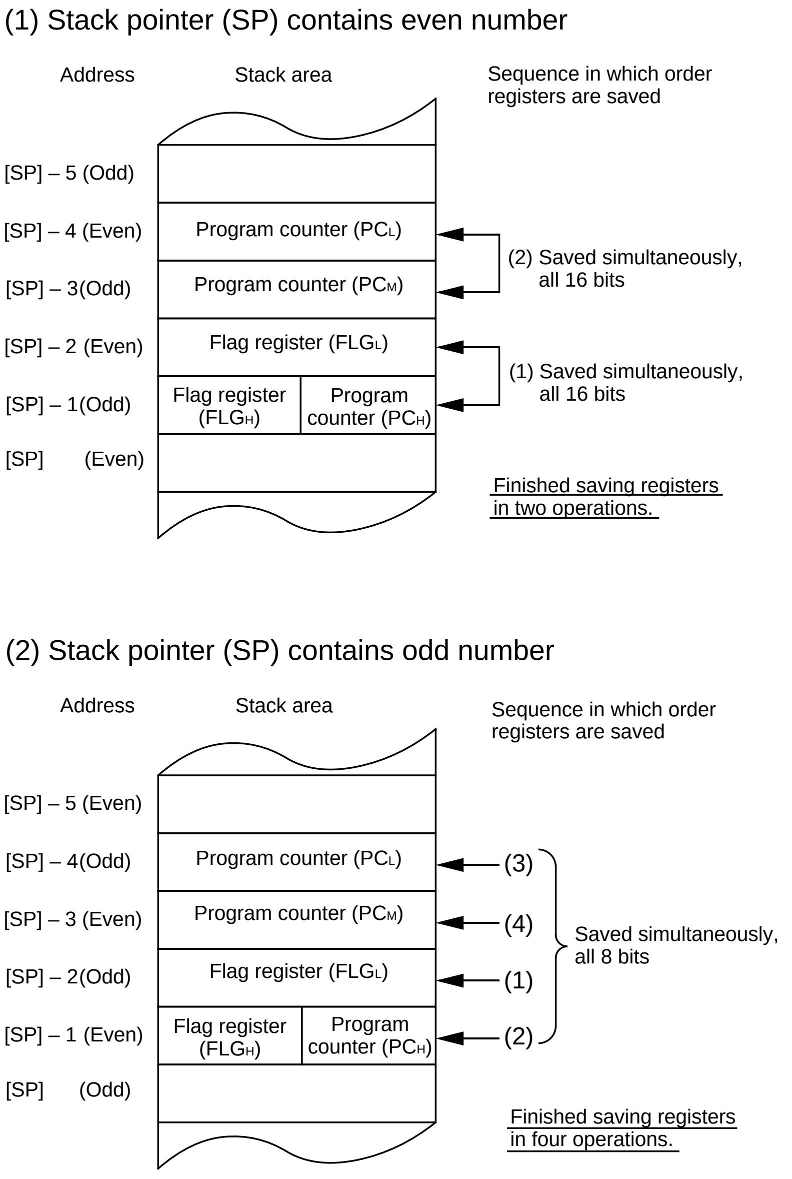


Figure 1.12.6. State of stack before and after acceptance of interrupt request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the content of the stack pointer (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 1.12.7 shows the operation of the saving registers.

Note: This is the stack pointer indicated by the U flag.



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

Figure 1.12.7. Operation of saving registers

Returning from an Interrupt Routine

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

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

Interrupt Priority

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

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

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

Figure 1.12.8 shows the priorities of hardware interrupts.

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

Interrupt Priority Level Judge Circuit

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

Figure 1.12.9 shows the interrupt resolution circuit.

Reset > UART1 receive > $\overline{\mathrm{DBC}}$ > Oscillation stop detection/watchdog timer > Peripheral I/O > Single step > Address match

Figure 1.12.8. Hardware interrupts priorities

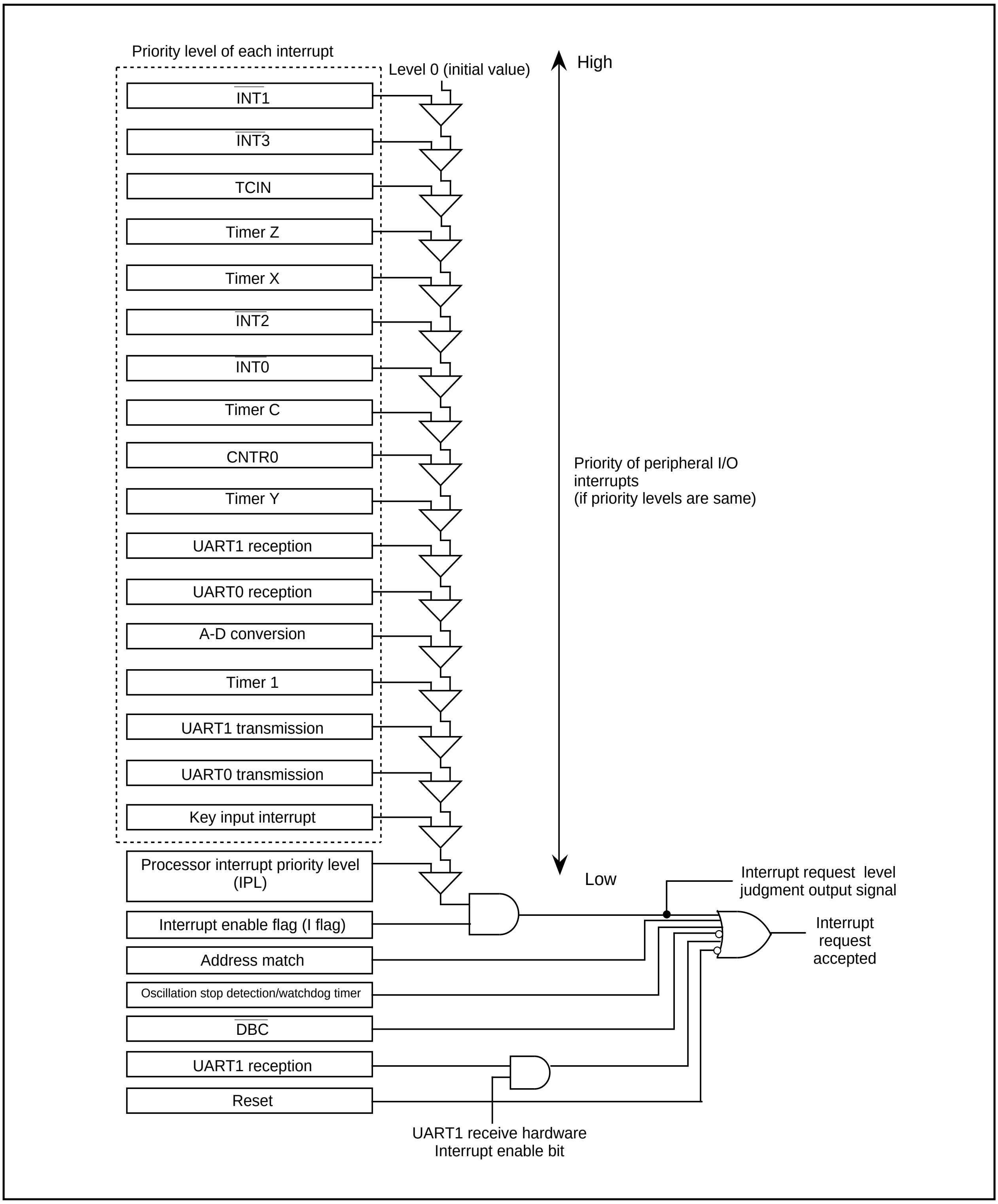

Priority level of each interrupt
Level 0 (initial value)
High
$\overline{\mathrm{INT1}}$
$\overline{\mathrm{INT3}}$
TCIN
Timer Z
Timer X
$\overline{\mathrm{INT2}}$
$\overline{\mathrm{INT0}}$
Timer C
CNTR0
Timer Y
UART1 reception
UART0 reception
A-D conversion
Timer 1
UART1 transmission
UART0 transmission
Key input interrupt
Priority of peripheral I/O interrupts (if priority levels are same)
Low
Processor interrupt priority level (IPL)
Interrupt enable flag (I flag)
Address match
Oscillation stop detection/watchdog timer
$\overline{\mathrm{DBC}}$
UART1 reception
Reset
Interrupt request level judgment output signal
Interrupt request accepted
UART1 receive hardware Interrupt enable bit

Figure 1.12.9. Interrupt resolution circuit

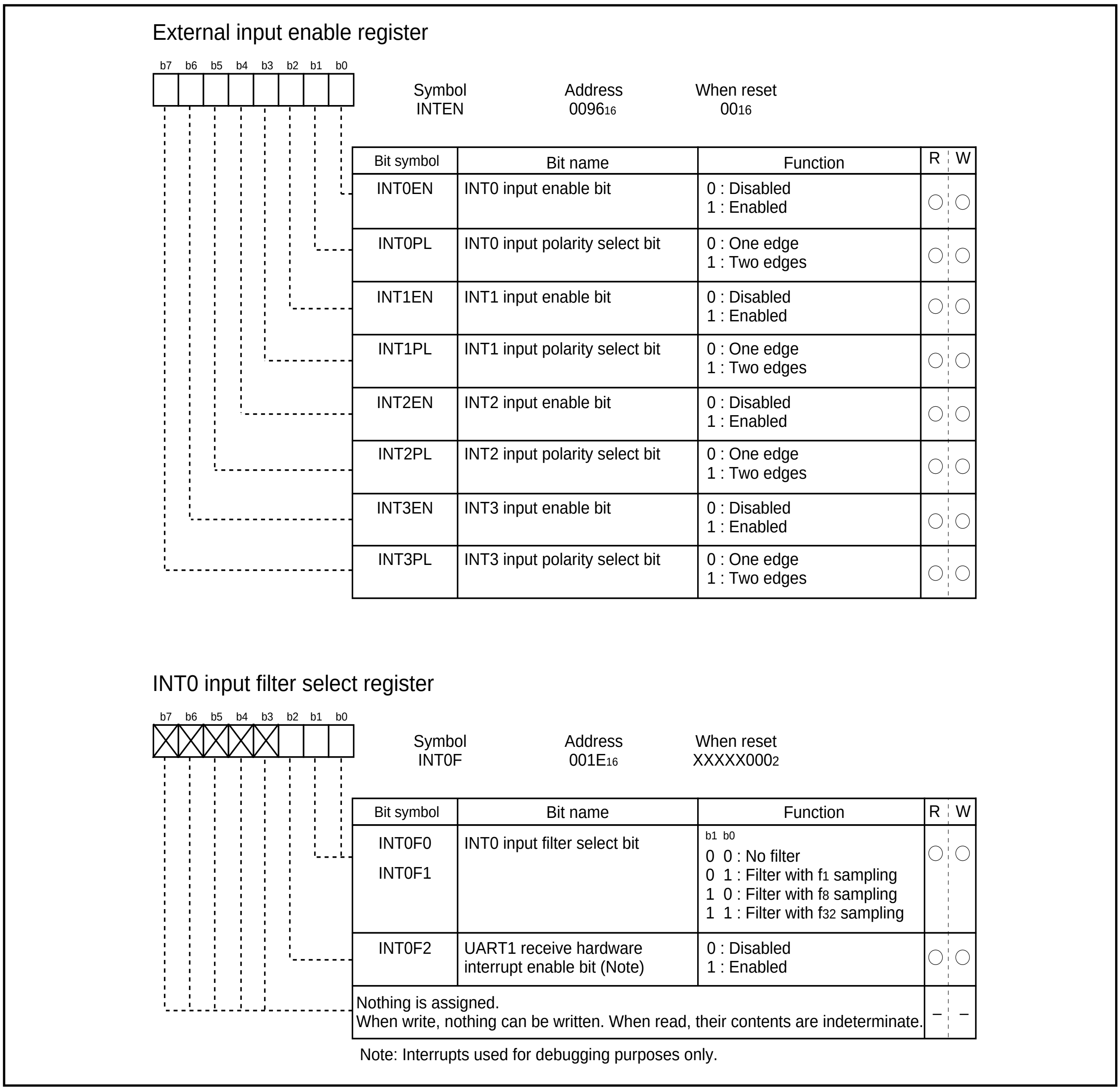
$\overline{\text{INT}}$ Interrupt

$\overline{\text{INT}}$0 to $\overline{\text{INT}}$3 are triggered by the edges of external inputs. The edge polarity of $\overline{\text{INT}}$0 to $\overline{\text{INT}}$2 is selected using the polarity select bit (bit 4 of addresses $005D_{16}$, $005E_{16}$ and $005F_{16}$). Input to INT0 is available via filter with three different sampling frequencies.

As to external interrupt input, an interrupt can be generated both at the rising edge and at the falling edge by setting the $\overline{\text{INT}}$i (i=0 to 3) input polarity select bit of the external input enable register (0096_{16}) to “1”. To select both edges, set the polarity switching bit of the corresponding interrupt control register to “0” (falling edge). To select one edge, set the polarity switching bit of the corresponding interrupt control register to either “1” (raising edge) or “0” (falling edge). Please note that when one edge is selected using $\overline{\text{INT3}}$, the polarity will be a falling edge.

After setting the external input enable register, clear the interrupt request bit, and then enable the corresponding input interrupt. Moreover, you should write to the external input enable bit only under conditions where the corresponding input interrupt does not occur.

Figure 1.12.10 shows the external input related registers.



External input enable register

Symbol	Address	When reset
INTEN	0096_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
INT0EN	INT0 input enable bit	0 : Disabled 1 : Enabled	○	○
INT0PL	INT0 input polarity select bit	0 : One edge 1 : Two edges	○	○
INT1EN	INT1 input enable bit	0 : Disabled 1 : Enabled	○	○
INT1PL	INT1 input polarity select bit	0 : One edge 1 : Two edges	○	○
INT2EN	INT2 input enable bit	0 : Disabled 1 : Enabled	○	○
INT2PL	INT2 input polarity select bit	0 : One edge 1 : Two edges	○	○
INT3EN	INT3 input enable bit	0 : Disabled 1 : Enabled	○	○
INT3PL	INT3 input polarity select bit	0 : One edge 1 : Two edges	○	○

INT0 input filter select register

Symbol	Address	When reset
INT0F	$001E_{16}$	$XXXXX000_2$

Bit symbol	Bit name	Function	R	W
INT0F0 INT0F1	INT0 input filter select bit	b1 b0 0 0 : No filter 0 1 : Filter with f_1 sampling 1 0 : Filter with f_8 sampling 1 1 : Filter with f_{32} sampling	○	○
INT0F2	UART1 receive hardware interrupt enable bit (Note)	0 : Disabled 1 : Enabled	○	○
Nothing is assigned. When write, nothing can be written. When read, their contents are indeterminate.			–	–

Note: Interrupts used for debugging purposes only.

Figure 1.12.10. External input related registers

INT0 Input Filter

The INT0 input has a digital filter which can be sampled by one of three sampling clocks. You select the sampling clock using the INT0 Input Filter Select bits, bits 1 and 0.

INT0 interrupt request occurs when the sampled input level matches three times.

Figure 1.12.11 shows the INT0 input filter.

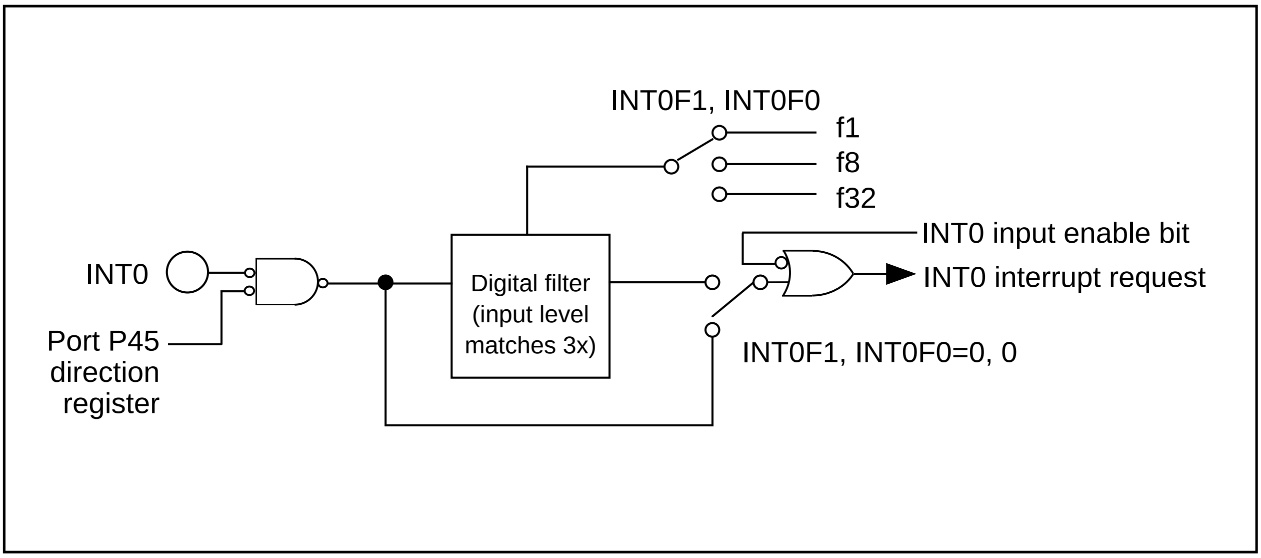


Figure 1.12.11. INT0 input filter

UART1 Receive Hardware Input

A hardware interrupt can be generated during UART1 receive by enabling the UART1 Receive Interrupt Enable bit. The interrupt vector is stored in addresses FFFF8$_{16}$ to FFFFB$_{16}$.

Note: The UART1 Receive Hardware Interrupt is a debugger exclusive interrupt.

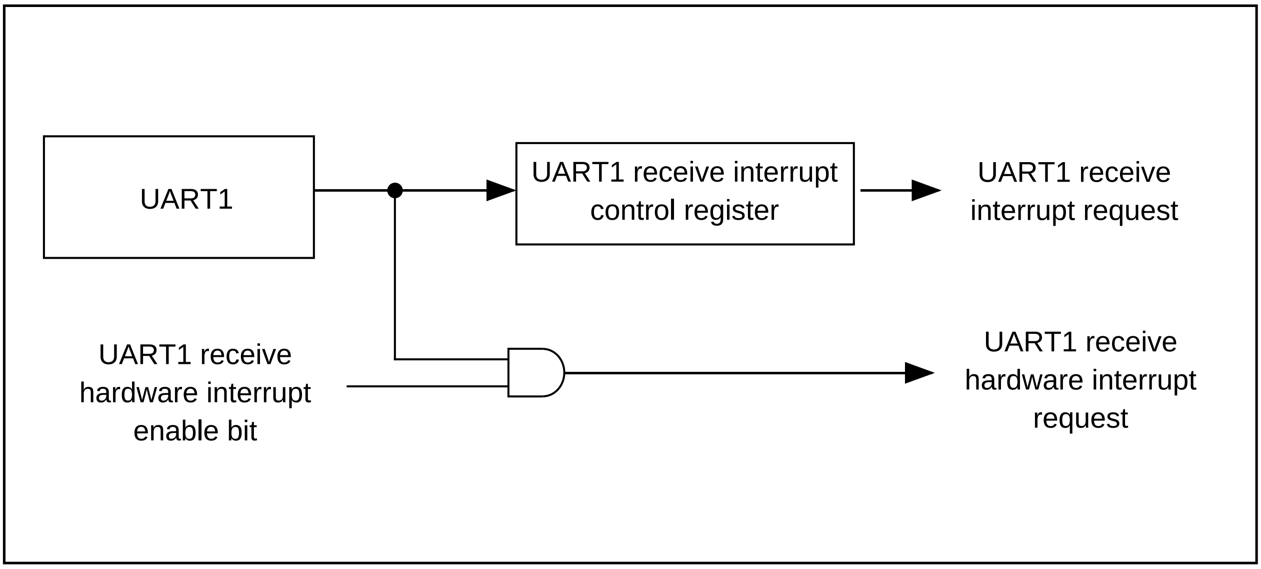

UART1
UART1 receive interrupt control register
UART1 receive interrupt request
UART1 receive hardware interrupt enable bit
UART1 receive hardware interrupt request

Figure 1.12.12. UART1 receive hardware interrupt

Key Input Interrupt

When the direction register of any of $P1_0$ to $P1_3$ is set for input and the KIi (i=0 to 3) input enable bit of this port is set for enabled, if a falling or rising 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 1.12.13 shows the block diagram of the key input interrupts. When the appropriate signal (“L” for a pin that has falling edge selected and “H” for a pin that has rising edge selected) is input to a pin for the input inhibit process has not been executed, inputs to the other pins are not detected as interrupts.

You should overwrite the KIi (i=0 to 3) input polarity select bit or the KIi (i =0 to 3) input enable bit only under conditions where the key input interrupt does not occur. After overwriting the KIi (i=0 to 3) input polarity select bit or the KIi (i=0 to 3) input enable bit, clear the interrupt request bit, and then enable the key input interrupt.

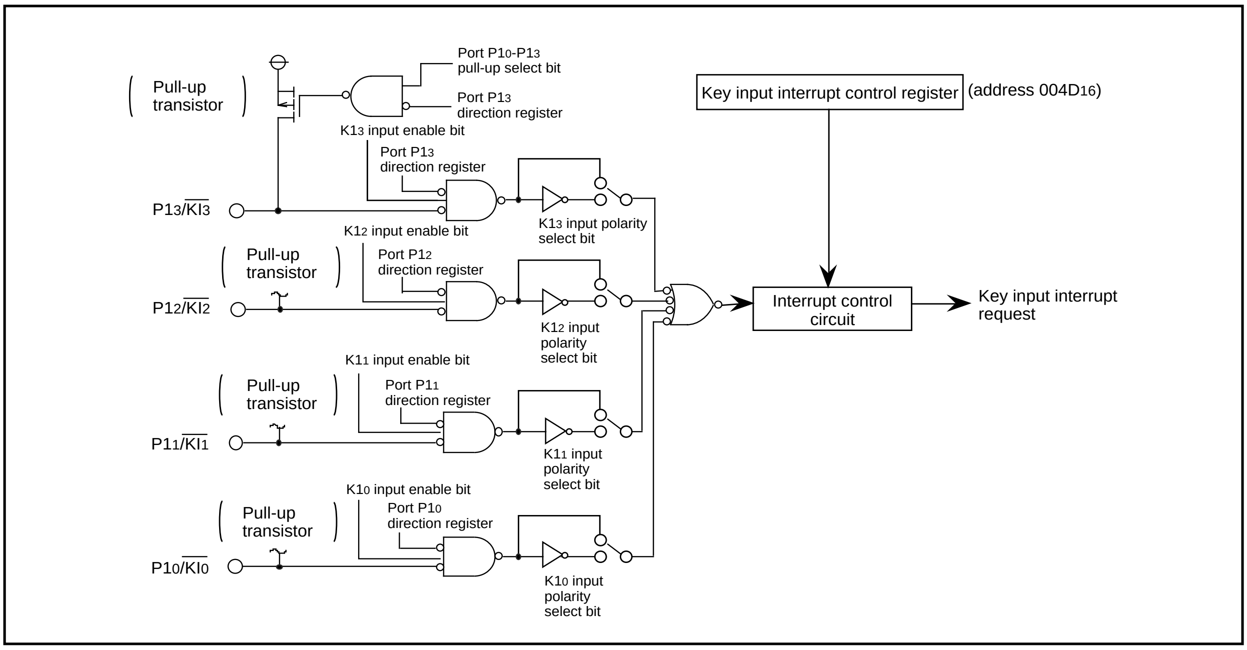

Port P10-P13 pull-up select bit
Pull-up transistor
Port P13 direction register
K13 input enable bit
Key input interrupt control register
(address 004D16)
Port P13 direction register
P13/KI3
K13 input polarity select bit
K12 input enable bit
Pull-up transistor
Port P12 direction register
P12/KI2
K12 input polarity select bit
Interrupt control circuit
Key input interrupt request
K11 input enable bit
Pull-up transistor
Port P11 direction register
P11/KI1
K11 input polarity select bit
K10 input enable bit
Pull-up transistor
Port P10 direction register
P10/KI0
K10 input polarity select bit

Figure 1.12.13. Block diagram of key input interrupt

Key input enable register

b7 b6 b5 b4 b3 b2 b1 b0

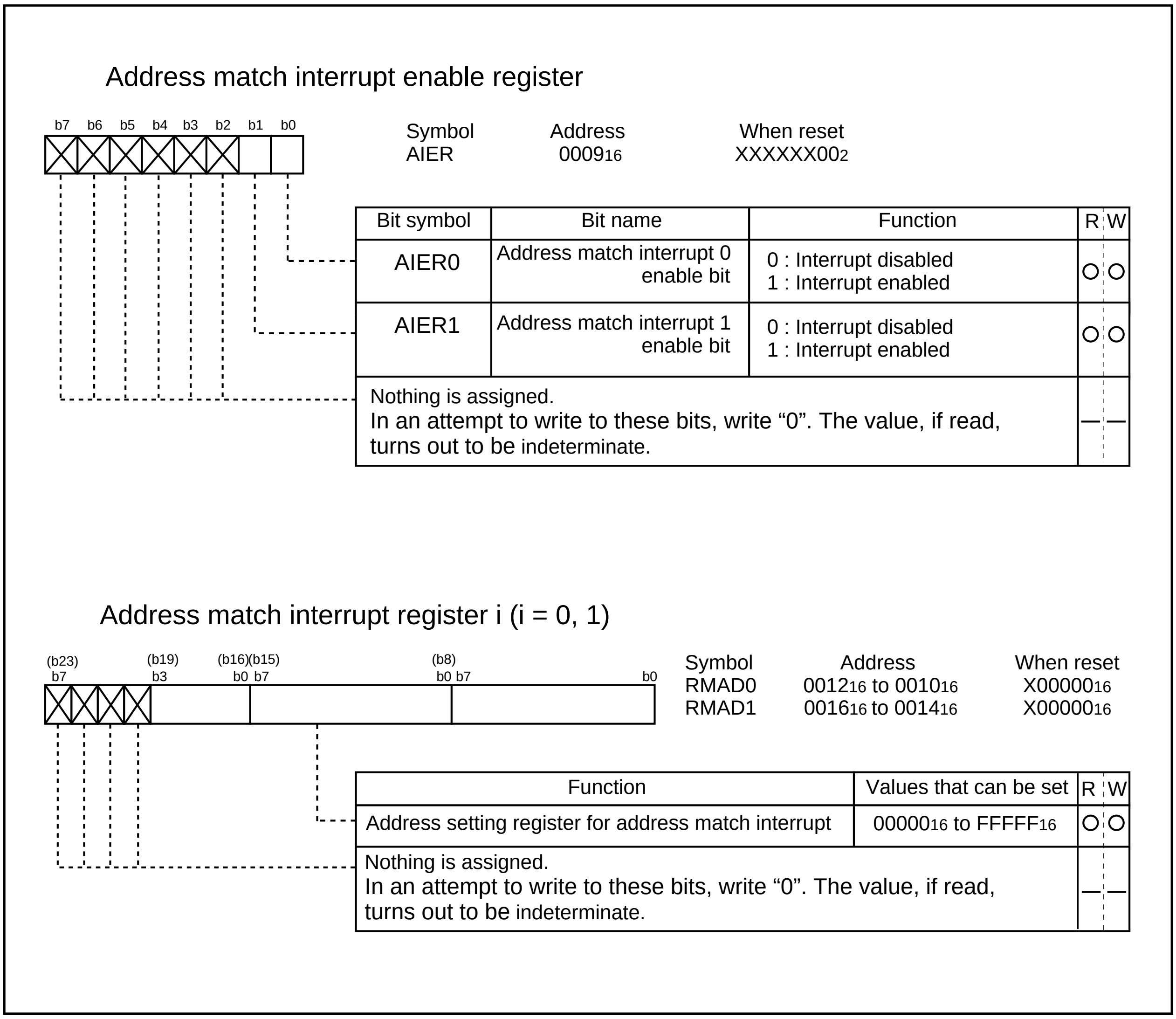
Symbol	Address	When reset
KIEN	0098$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
KI0EN	KI0 input enable bit	0 : Disabled 1 : Enabled	○	○
KI0PL	KI0 input polarity select bit	0 : Falling edge 1 : Rising edges	○	○
KI1EN	KI1 input enable bit	0 : Disabled 1 : Enabled	○	○
KI1PL	KI1 input polarity select bit	0 : Falling edge 1 : Rising edges	○	○
KI2EN	KI2 input enable bit	0 : Disabled 1 : Enabled	○	○
KI2PL	KI2 input polarity select bit	0 : Falling edge 1 : Rising edges	○	○
KI3EN	KI3 input enable bit	0 : Disabled 1 : Enabled	○	○
KI3PL	KI3 input polarity select bit	0 : Falling edge 1 : Rising edges	○	○

Figure 1.12.14. Key input enable register

Address Match Interrupt

An address match interrupt is generated immediately before the instruction at the address indicated by the address match interrupt register is executed. Two address match interrupts can be set, each of which can be enabled and disabled by an address match interrupt enable bit. Address match interrupts are not affected by the interrupt enable flag (I flag) and processor interrupt priority level (IPL). The value of the program counter (PC) for an address match interrupt varies depending on the instruction being executed. Figure 1.12.15 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 1.12.15. Address match interrupt-related registers

Precautions for Interrupts

(1) Reading address 0000$_{16}$

- When maskable interrupt is occurred, CPU reads 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 0000$_{16}$ will then be set to “0”.

  Even if the address 0000$_{16}$ is read out by software, “0” is set to the enabled highest priority interrupt source request bit. Therefore, interrupt can be canceled and unexpected interrupt can occur.

  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 000$_{16}$. 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 $\overline{\text{INT0}}$ to $\overline{\text{INT3}}$ regardless of the CPU operation clock.
- When changing a polarity of pins $\overline{\text{INT0}}$ to $\overline{\text{INT3}}$, the interrupt request bit may become "1". Clear the interrupt request bit after changing the polarity. Figure 1.12.16 shows the switching condition of $\overline{\text{INT}}$ interrupt request.

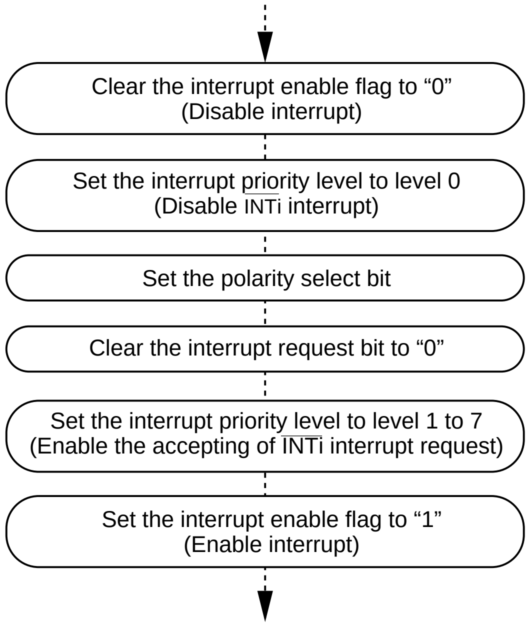

Clear the interrupt enable flag to “0”
(Disable interrupt)
Set the interrupt priority level to level 0
(Disable $\overline{\text{INTi}}$ interrupt)
Set the polarity select bit
Clear the interrupt request bit to “0”
Set the interrupt priority level to level 1 to 7
(Enable the accepting of $\overline{\text{INTi}}$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

Figure 1.12.16. Switching condition of $\overline{\text{INT}}$ interrupt request

(4) Changing interrupt control register

See "Rewrite The 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 or reset is generated when an underflow occurs in the watchdog timer. A watchdog timer interrupt or reset is selected by bit 2 of the processor mode register 1. When X_{IN} is selected for the BCLK, bit 7 of the watchdog timer control register (address $000F_{16}$) selects the prescaler division ratio (by 16 or by 128). When X_{CIN} is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address $000F_{16}$).

When X_{IN} is selected in BCLK

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

When X_{CIN} is selected in BCLK

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

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

The watchdog timer is initialized by writing to the watchdog timer start register (address $000E_{16}$) and when a watchdog timer interrupt request is generated. The prescaler is initialized only when the microcomputer is reset. After a reset is cancelled, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address $000E_{16}$).
Figure 1.13.1 shows the block diagram of the watchdog timer. Figure 1.13.2 shows the watchdog timer-related registers.

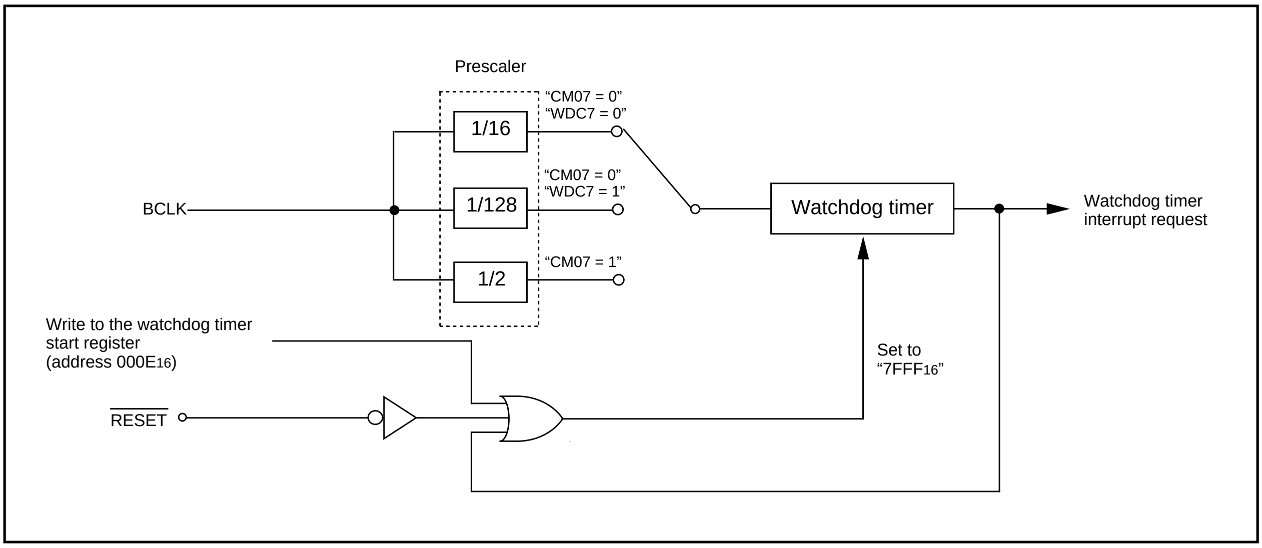
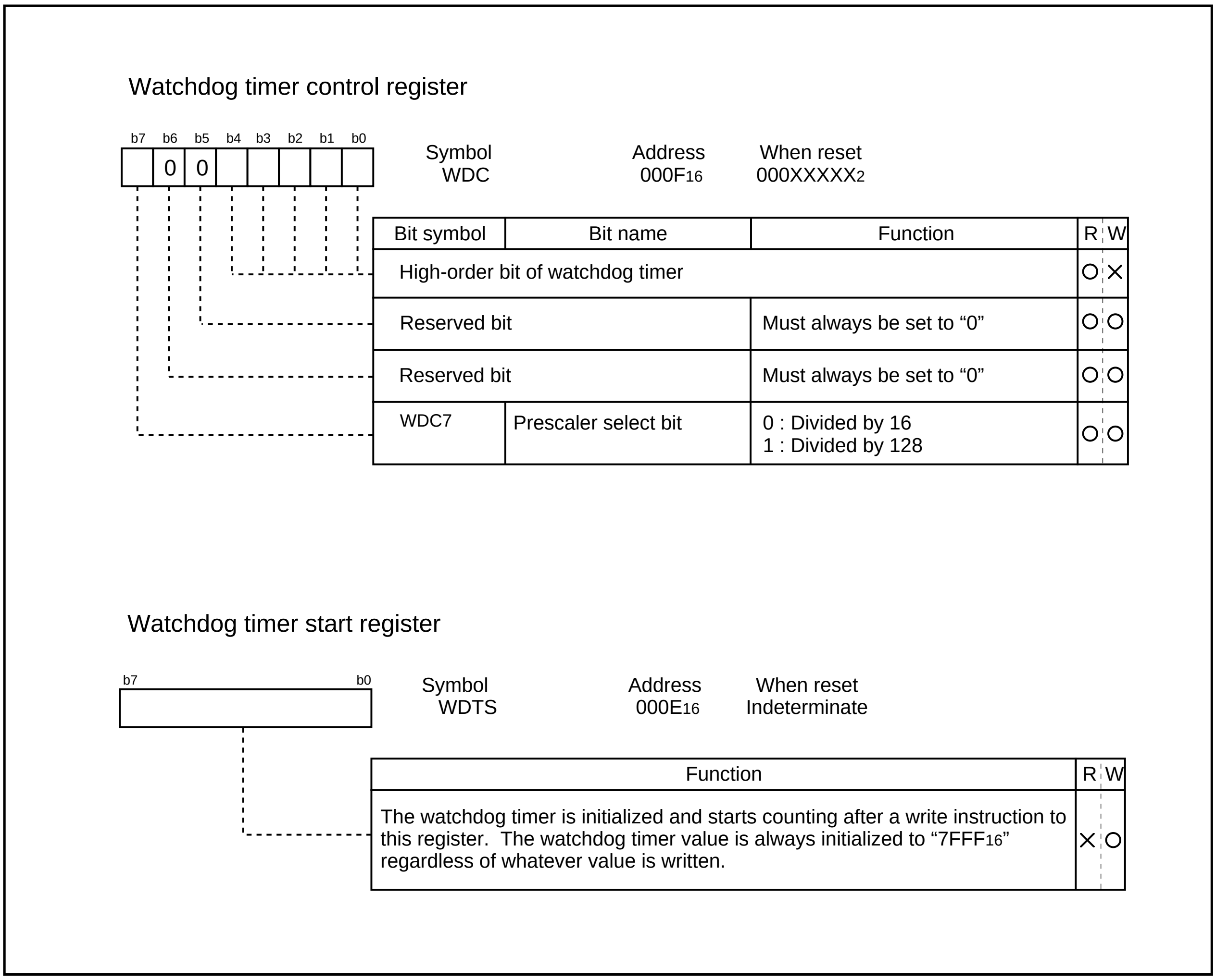


Figure 1.13.1. Block diagram of watchdog timer



Watchdog timer control register

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

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 1.13.2. Watchdog timer control and start registers

Timer

The microcomputer has a total of four timers: Timer 1, timer X, timer Y, and timer Z.

The divide-by ratios of all timers and prescalers are determined by 1 / (n + 1) where n = content of the timer reload register or prescaler reload register.

The timers are down-counters, so that the timer underflows at the next count pulse after it reached the minimum count of 0, and is reloaded with the content of the timer reload register. Also, when the timer underflows, the interrupt request bit corresponding to each timer is set to 1.

Timer 1

Timer 1 is an 8-bit timer, which always counts prescaler-1 output. When timer 1 underflows after reaching the minimum count, the timer 1 interrupt request bit is set.

Prescaler 1 is an 8-bit prescaler, which counts the signal selected with the timer 1 count source select bit.

Prescaler 1 and timer 1 respectively have a prescaler-1 reload register and a timer-1 reload register to hold their reload values. The value of the prescaler-1 reload register is transferred to prescaler 1 when it underflows. Similarly, the value of the timer-1 reload register is transferred to timer 1 when it underflows.

Any value written to prescaler 1 (PRE1) is also written to the prescaler-1 reload register at the same time.

Any value written to timer 1 (T1) is also written to the timer-1 reload register at the same time.

When prescaler 1 (PRE1) or timer 1 (T1) is accessed for read, their count value is read out.

Timer 1 always operates in timer mode.

Prescaler 1 counts the selected count source and each time the count clock is applied, the prescaler has its content decremented by one. After reaching “00_{16},” prescaler 1 underflows at the next count clock and is reloaded with the value transferred from the prescaler-1 reload register and starts counting over again.

The divide-by ratio of prescaler 1 is 1 / (n + 1) where n = set value of prescaler 1.

Timer 1 has its content decremented by one each time an underflow signal is received from prescaler 1. After reaching “00_{16},” timer 1 underflows at the next count clock and is reloaded with the value transferred from the timer-1 reload register and starts counting over again.

The divide-by ratio of timer 1 is 1 / (m + 1) where m = set value of Timer 1. Therefore, assuming n = set value of prescaler 1 and m = set value of timer 1, the divide-by ratio of timer 1 is 1 / ((n + 1) x (m + 1)).

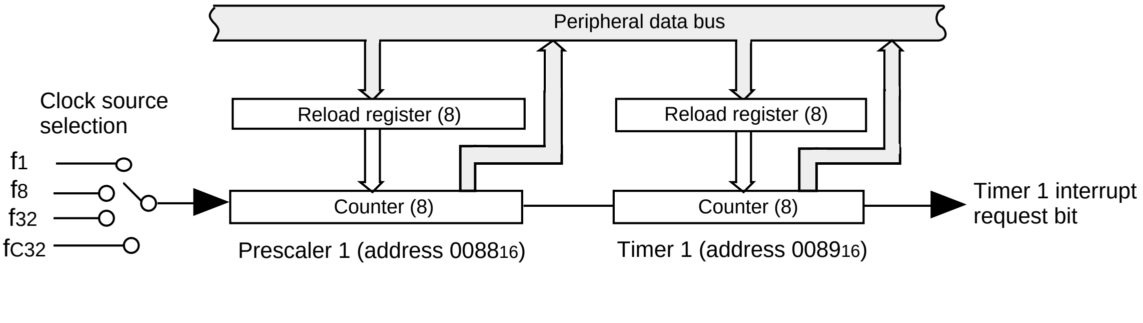


Figure 1.14.1. Block diagram of timer 1

Timer X

Timer X is an 8-bit timer, which always counts prescaler-X output. When timer X underflows after reaching the minimum count, the timer X interrupt request bit is set.

Prescaler X is an 8-bit prescaler, which counts the signal selected with the timer X count source select bit.

Prescaler X and timer X respectively have a prescaler-X reload register and a timer-X reload register to hold their reload values. The value of the prescaler-X reload register is transferred to prescaler X when it underflows. Similarly, the value of the timer-X reload register is transferred to timer X when it underflows.

Any value written to prescaler X (PREX) is also written to the prescaler-X reload register at the same time.

Any value written to timer X (TX) is also written to the timer-X reload register at the same time.

When prescaler X (PREX) or timer X (TX) is accessed for read, their count value is read out.

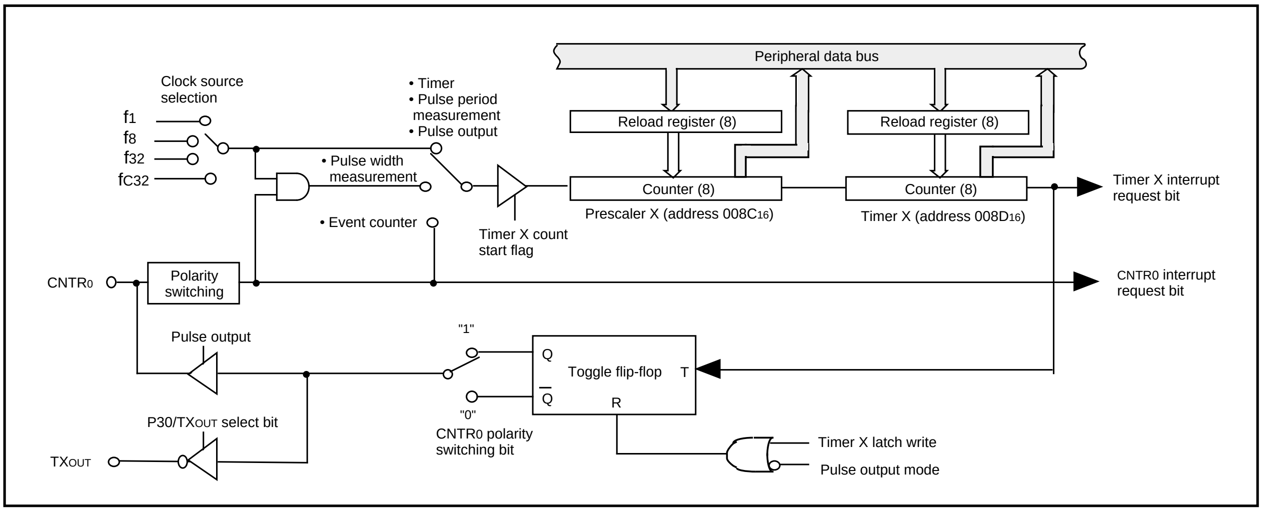


Figure 1.14.2. Block diagram of timer X

Timer X mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol TXMR

Address $008B_{16}$

When reset $XX000000_2$

Bit symbol	Bit name	Function	R	W
TXMOD0	Operation mode select bit 0, 1	b1 b0 0 0 : Timer mode or pulse period measurement mode 0 1 : Pulse output mode (Note 1) 1 0 : Event counter mode 1 1 : Pulse width measurement mode	○	○
TXMOD1			○	○
R0EDG	CNTR0 polarity switching bit	0 : Interrupt at rising edge • Count on rising edge (when event counter mode) • "H" width measurement (when pulse width measurement mode) 1: Interrupt at falling edge • Count on falling edge (When event counter mode) • "L" width measurement (When pulse width measurement mode)	○	○
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	○	○
TXOCNT	P3$_0$/TX$_{OUT}$ select bit	0 : Port P3$_0$ 1 : TX$_{OUT}$ output (Note 2)	○	○
TXMOD2	Operation mode select bit 2(Note 3)	0 : Timer mode 1 : Pulse period measurement mode	○	○
Nothing is assigned. When write, set "0". When read, their contents are "0".			—	—

Note 1: Output is set regardless of the setting of the direction register of port P1$_7$.
Note 2: Output is set regardless of the setting of the direction register of port P3$_0$.
Note 3: Except in pulse period measurement mode, the operation mode select bit 2 must be set to "0".

Figure 1.14.3. Timer X mode register

Timer X has five operation modes that can be selected by setting the Timer X Mode Register.

(1) Timer mode

Prescaler X counts the selected count source and each time the count clock is applied, the prescaler has its content decremented by one. After reaching "00$_{16}$," prescaler X underflows at the next count clock and is reloaded with the value transferred from the prescaler-X reload register and starts counting over again. The divide-by ratio of prescaler X is 1 / (n + 1) where n = set value of prescaler X.

Timer X has its content decremented by one each time an underflow signal is received from prescaler X. After reaching "00$_{16}$," timer X underflows at the next count clock and is reloaded with the value transferred from the timer-X reload register and starts counting over again. The divide-by ratio of timer X is 1 / (m + 1) where m = set value of Timer X. Therefore, assuming n = set value of prescaler X and m = set value of timer X, the divide-by ratio of timer X is 1 / ((n + 1) x (m + 1)).

(2) Pulse output mode

In pulse output mode, the microcomputer outputs from the CNTR0 pin a waveform whose polarity is inverted each time timer X underflows. In the pulse output mode, said output pin is always directed for output regardless of how the direction register for the port P1$_7$ is set.

The output level on the CNTR0 pin can be selected with the CNTR0 pin polarity select bit. When the CNTR0 pin polarity select bit = 0, the first data output from the CNTR0 pin is high. When the CNTR0 pin polarity select bit = 1, the first data output from the CNTR0 pin is low.

Also, by setting the P30/TXOUT output enable bit to "1", it is possible to output from the TXOUT pin an inverted waveform of the pulse that is output from the CNTR0 pin.

When setting the P3$_0$/TXOUT output enable bit to "1", said output pin is always directed for output regardless of how the direction register for the port P3$_0$ is set.

(3) Event counter mode

In event counter mode, timer X operates in the same way as in timer mode, except that the signal fed to the P1$_7$/CNTR0 pin is the count source. The active edge on CNTR0 pin input can be selected to be the rising or the falling edge with the CNTR0 polarity select bit.

(4) Pulse width measure mode

This mode is used to measure the pulse width of the signal fed to the P1$_7$/CNTR0 pin. In pulse width measure mode, timer X is controlled to run or stop by the input signal level on the CNTR0 pin. The value of the timer X is not reloaded until the timer underflows. When the timer X underflows, a timer X interrupt is generated.

When the CNTR0 polarity select bit = 0, timer X counts the signal selected with the timer X count source select bit while the CNTR0 pin input is high, and stops counting when the CNTR0 pin input is low. Conversely, when the CNTR0 polarity select bit = 1, timer X counts the signal selected with the timer X count source select bit while the CNTR0 pin input is low, and stops counting when the CNTR0 pin input is high.

(5) Pulse period measure mode

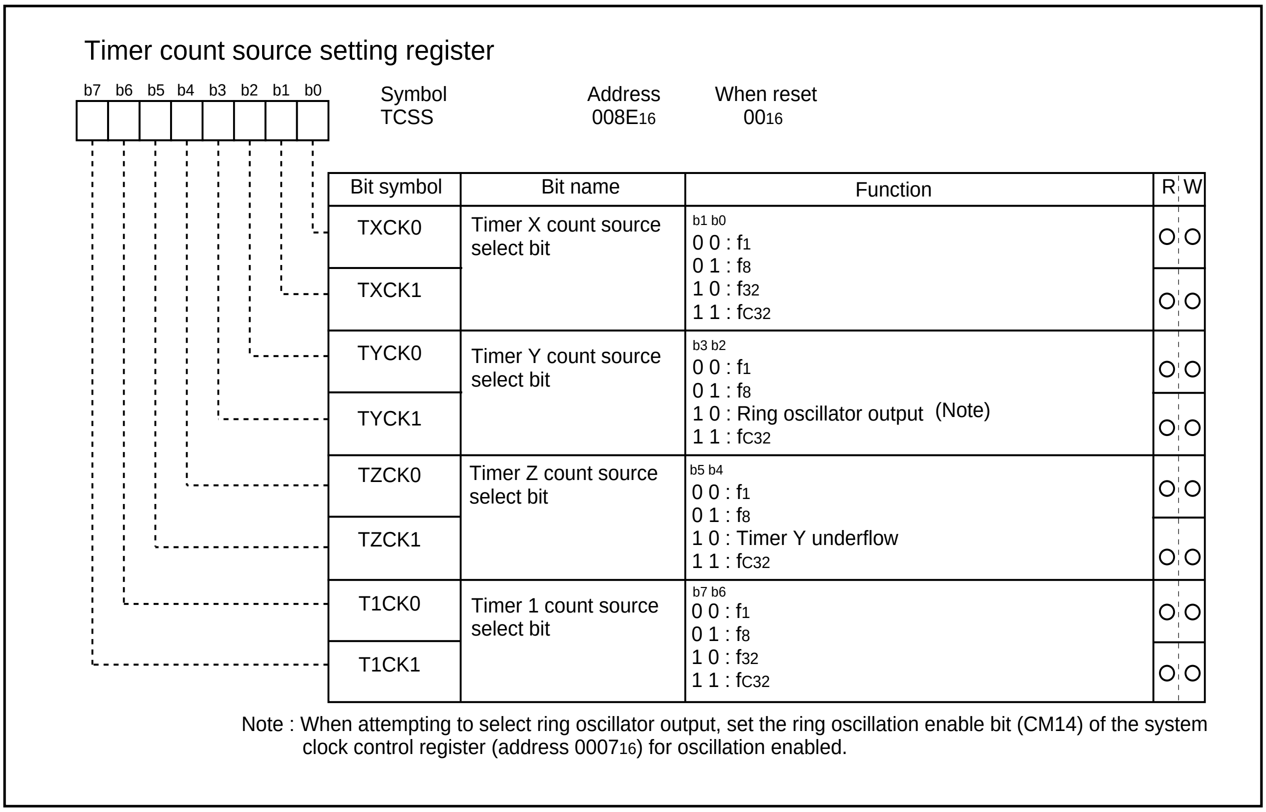
This mode is used to measure the pulse period of the signal fed to the $P1_7$/CNTR0 pin.
When the CNTR0 polarity select bit = 0, timer X counts the signal selected with the timer X count source select bit for a period from one rising edge to the next rising edge of the CNTR0 pin input signal. Upon occurrence of a rising edge, the value of timer X used for read purpose is retained and timer X is reloaded with the value as it newly starts counting. The value of timer X used for read purpose is no longer retained by reading timer X. When the CNTR0 polarity select bit = 1, timer X counts the signal selected with the timer X count source select bit for a period from one falling edge to the next falling edge of the CNTR0 pin input signal. Upon occurrence of a falling edge, the value of timer X used for read purpose is retained and timer X is reloaded with the value as it newly starts counting. The value of timer X used for read purpose is no longer retained by reading timer X.
The value of timer X must be read out before the pulse period measure mode is set. Furthermore, the timer X must be read within the next period of CNTR0. If the value is read beyond the period, the value of the next period of the period which was read last time is read out.

In any operation mode, timer X can be made to stop counting by setting the timer X count start flag to 0. When timer X underflows, the timer X interrupt request bit is set to 1.

■ Precaution

• CNTR0 polarity selection

The set value of the CNTR0 polarity select bit also affects the interrupt polarity (active-high or low). When the CNTR0 polarity select bit = 0, the CNTR0 interrupt request bit is set to 1 by the rising edge of the CNTR0 pin input. When the CNTR0 polarity select bit = 1, the CNTR0 interrupt request bit is set to 1 by the falling edge of the CNTR0 pin input.



Timer count source setting register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol TCSS　Address $008E_{16}$　When reset 00_{16}

Bit symbol	Bit name	Function	R	W
TXCK0	Timer X count source select bit	b1 b0 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	○	○
TXCK1			○	○
TYCK0	Timer Y count source select bit	b3 b2 0 0 : f_1 0 1 : f_8 1 0 : Ring oscillator output (Note) 1 1 : fc_{32}	○	○
TYCK1			○	○
TZCK0	Timer Z count source select bit	b5 b4 0 0 : f_1 0 1 : f_8 1 0 : Timer Y underflow 1 1 : fc_{32}	○	○
TZCK1			○	○
T1CK0	Timer 1 count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	○	○
T1CK1			○	○

Note : When attempting to select ring oscillator output, set the ring oscillation enable bit (CM14) of the system clock control register (address 0007_{16}) for oscillation enabled.

Figure 1.14.4. Timer count source setting register

Timer Y

Timer Y is an 8-bit timer, which always counts prescaler-Y output. When timer Y underflows after reaching the minimum count, the timer Y interrupt request bit is set.
Prescaler Y is an 8-bit prescaler, which counts the signal selected with the timer Y count source select bit. Prescaler Y has a prescaler-Y reload register to hold its reload values. The timer Y has a timer Y primary reload register and a secondary reload register to hold its reload values. The value of the prescaler-Y reload register is transferred to prescaler Y when it underflows.
The content of the timer Y primary reload register or timer Y secondary reload register is transferred to timer Y when the timer underflows. Which value, timer Y primary reload register or secondary reload register, is transferred to timer Y depends on the timer’s operation mode.
When writing to prescaler Y (PREY) and timer Y primary (TYPR) or timer Y secondary (TYSC), it is possible to choose whether to write to only the respective reload registeres or both prescaler Y and timer Y and their reload registers by setting the timer Y write control bit. Because setting this control bit is subject to limitations depending on operation modes used, be sure to observe precautions on each operation mode.

Reading prescaler Y (PREY) means reading out the count value of prescaler Y. Similarly, reading timer Y primary (TYPR) means reading out the count value of timer Y. The count value of timer Y can always be read out by reading timer Y primary (TYPR), regardless of whether the timer is counting the timer Y primary reload register or timer Y secondary reload register. Reading timer Y secondary (TYSC) results in an indeterminate value being read out.

Timer Y has two operation modes that can be selected by setting the Timer Y Mode Register and Timer Z Mode Register.

(1) Timer mode

Prescaler Y counts the selected count source and each time the count clock is applied, the prescaler has its content decremented by one. After reaching “00_{16},” prescaler Y underflows at the next count clock and is reloaded with the value transferred from the prescaler-Y reload register and starts counting over again. The divide-by ratio of prescaler Y is 1 / (n + 1) where n = set value of prescaler Y.
Timer Y has its content decremented by one each time an underflow signal is received from prescaler Y. After reaching “00_{16},” timer Y underflows at the next count clock and is reloaded with the value transferred from the timer-Y primary reload register and starts counting over again. (In timer mode, the timer always counts the content of the timer Y primary reload register. The timer Y secondary is unused in this mode.)
The divide-by ratio of timer Y is 1 / (m + 1) where m = set value of Timer Y. Therefore, assuming n = set value of prescaler Y and m = set value of timer Y, the divide-by ratio of timer Y is 1 / ((n + 1) x (m + 1)).
In timer mode, it is possible to choose whether to write to only the reload registers of prescaler Y and timer Y primary or both prescaler Y and timer Y and their reload registers by setting the timer Y write control bit.
In timer mode, an interrupt is generated by an underflow of timer Y.

(2) Programmable waveform generation mode

In programmable waveform generation mode, the microcomputer, while counting the set values of timer Y primary and timer Y secondary alternately, outputs from the TYOUT pin a waveform whose polarity is inverted each time timer Y underflows.

When operating in this mode, always be sure to set the timer Y write control bit to 1, thereby choosing to write to only the reload registers. In this mode, said pin is always directed for output regardless of how the direction register for the port $P3_2$ that is shared with the output pin is set.

The polarity of the output waveform is set by using the timer Y output level latch. If the timer Y output level latch is set to 0, the microcomputer outputs a high for the period equal to the set value of timer Y primary and a low for the period equal to the set value of timer Y secondary alternately. When the timer starts the output is high at the beginning and when it stops the output returns low. If the timer Y output level latch is set to 1, the microcomputer outputs a low for the period equal to the set value of timer Y primary and a high for the period equal to the set value of timer Y secondary alternately. When the timer starts the output is low at the beginning and when it stops the output returns high. If the Timer Y, Z Output Control Register's timer Y programmable waveform generation output switching bit is set to 1, the Port $P3_2$ Register value is output synchronously when timer Y secondary underflows.

In this mode also, the waveform output primary period and secondary period can each be extended 0.5 cycles of the count source by setting the timer Y primary waveform extension and timer Y secondary waveform extension control bits to 1. This helps to output waveforms with higher resolution.

When using the waveform extension control bits, the frequency and duty cycle of the output waveform become as shown below.

Waveform frequency: FYOUT=(2xTMYCL)/((2xTYPR+1)+2x(TYSC+1)+(EXPYPR+EXPYSC))
Duty: DYOUT=(2x(TYPR+1)+(EXPYPR)/((2x(TYPR+1)+EXPYPR)+(2x(TYSC+1)+EXPYSC))
TMYCL: Timer Y count source (frequency)
TYPR: Timer Y primary (8-bit)
TYSC: Timer Y secondary (8-bit)
EXPYPR: Timer Y primary waveform extension control bit (1 bit)
EXPYSC: Timer Y secondary waveform extension control bit (1 bit)

In programmable waveform generation mode, if the value of the timer Y primary, timer Y secondary, or primary or secondary waveform extension control bit is altered, control is exercised in such a way that the output waveform changes from the beginning of the waveform period (i.e., the timer Y primary waveform period).

To alter the count value, set the timer Y secondary, primary waveform extension control bit, secondary waveform extension control bit, and finally the timer Y primary. The values thus set are reflected collectively at the beginning of the next waveform period after writing to the timer Y primary. (Even when writing while the timer is idle, the timer Y primary is always the last register to be written.)

In programmable waveform generation mode, an interrupt is generated by only an underflow of the secondary period.

■ Precaution

• Change of set count values

When altering the set count values in programmable waveform generation mode, note that the set value of the timer Y secondary and those of the primary and secondary waveform extension control bits are made effective by setting the timer Y primary. Therefore, even when the set value of the timer Y primary does not need to be altered, be sure to write the already set value to the timer Y primary again.

• Use of the waveform extend function

The waveform extend function based on timer Y waveform extension control bits is useful only when the value “00_{16}” is set in prescaler Y. If any other value is set in prescaler Y, always be sure to set the timer Y primary waveform extension and timer Y secondary waveform extension control bits to 0. Note also that if timer Y underflow is selected for the count source, the waveform extend function cannot be used either.

• Timer Y write mode

When operating in programmable waveform generation mode, always be sure to set the timer Y write control bit to 1, thereby choosing to write to the reload register.

In any operation mode, timer Y can be made to stop counting by setting the timer Y count start flag to 0. When timer Y underflows, the timer Y interrupt request bit is set to 1.

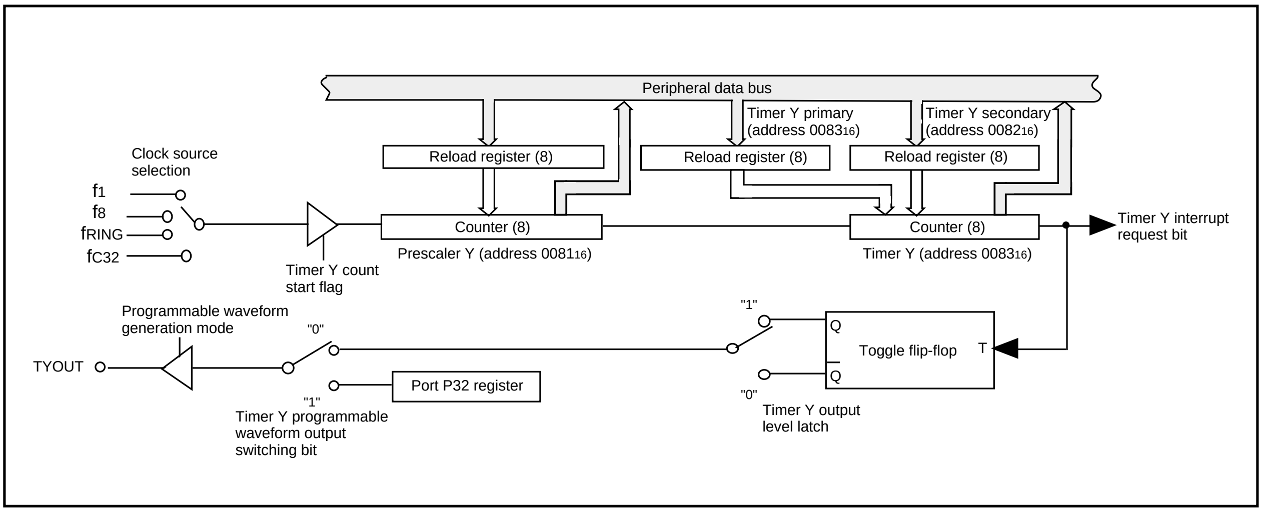


Figure 1.14.5. Block diagram of timer Y

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
						☒	

Symbol TYZMR   Address 0080_{16} When reset $000000X0_2$

Bit symbol	Bit name	Function	R	W
TYMOD0	Timer Y operation mode bit	0 : Timer mode 1 : Programmable waveform generation mode (Note 1)	○	○
Nothing is assigned. When write, set "0". When read, the content is "0".			—	—
TYWC	Timer Y write control bit	0 : Write to timer and reload register simultaneously 1 : Write to reload register	○	○
TYS	Timer Y count start flag	0 : Stops counting 1 : Starts counting	○	○
TZMOD0	Timer Z operation mode bit (Note 2)	b5 b4 0 0 : Timer mode 0 1 : Programmable waveform generation mode 1 0 : Programmable one-shot generation mode 1 1 : Programmable wait one-shot generation mode	○	○
TZMOD1			○	○
TZWC	Timer Z write control bit	0 : Write to timer and reload register simultaneously 1 : Write to reload register	○	○
TZS	Timer Z count start flag	0 : Stops counting 1 : Starts counting	○	○

Note 1: In programmable waveform generation mode, port $P3_2$ is set for output regardless of the value of the direction register.
Note 2: When timer Z operation mode bit is set for "01", "10" or "11", port $P3_1$ is set for output regardless of the value of the direction register.

Timer Y, Z waveform output control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol PUM   Address 0084_{16} When reset 00_{16}

Bit symbol	Bit name	Function	R	W
TYPUM 0	Timer Y primary waveform extension control bit	0 : No waveform extension 1 : Waveform extension	○	○
TYPUM 1	Timer Y secondary waveform extension control bit	0 : No waveform extension 1 : Waveform extension	○	○
TZPUM0	Timer Z primary waveform extension control bit	0 : No waveform extension 1 : Waveform extension	○	○
TZPUM1	Timer Z secondary waveform extension control bit	0 : No waveform extension 1 : Waveform extension	○	○
TYOPL	Timer Y output level latch	0 : Outputs an "L" 1 : Outputs an "H"	○	○
TZOP L	Timer Z output level latch	0 : Outputs an "L" 1 : Outputs an "H"	○	○
INOSTG	INT0 pin one-shot trigger control bit	0 : INT0 pin one-shot trigger invalid 1 : INT0 pin one-shot trigger valid	○	○
INOSEG	INT0 pin one-shot trigger polarity select bit	0 : Edge trigger at falling edge 1 : Edge trigger at rising edge	○	○

Timer Y, Z output control register

b7	b6	b5	b4	b3	b2	b1	b0
☒	☒	☒	☒	☒			

Symbol TYZOC   Address $008A_{16}$ When reset $XXXXX000_2$

Bit symbol	Bit name	Function	R	W
TZOS	Timer Z one-shot start bit	0 : Stops one-shot 1 : Starts one-shot	○	○
TYOCNT	Timer Y programmable waveform generation output switching bit (Note)	0 : Programmable waveform generation output 1 : $P3_2$ port output	○	○
TZOCNT	Timer Z programmable waveform generation output switching bit (Note)	0 : Programmable waveform generation output 1 : $P3_1$ port output	○	○
Nothing is assigned. When write, set "0". When read, their contents are "0".			—	—

Note: This bit is valid only when operating in the programmable waveform generation mode.

Figure 1.14.6. Timer Y, Z mode register and timer Y, Z waveform output control registers

Timer Z

Timer Z is an 8-bit timer, which always counts prescaler-Z output. When timer Z underflows after reaching the minimum count, the timer Z interrupt request bit is set.

Prescaler Z is an 8-bit prescaler, which counts the signal selected with the timer Z count source select bit. Prescaler Z has a prescaler-Z reload register to hold its reload values. The timer Z has a timer Z primary reload register and a secondary reload register to hold its reload values. The value of the prescaler-Z reload register is transferred to prescaler Z when it underflows.

The content of the timer Z primary reload register or timer Z secondary reload register is transferred to timer Z when the timer underflows. Which value, timer Z primary reload register or secondary reload register, is transferred to timer Z depends on the timer's operation mode.

When writing to prescaler Z (PREZ) and timer Z primary (TZPR) or timer Z secondary (TZSC), it is possible to choose whether to write to only the respective reload registeres or both prescaler Z and timer Z and their reload registeres by setting the timer Z write control bit. Because setting this control bit is subject to limitations depending on operation modes used, be sure to observe precautions on each operation mode.

Reading prescaler Z (PREZ) means reading out the count value of prescaler Z. Similarly, reading timer Z primary (TZPR) means reading out the count value of timer Z. The count value of timer Z can always be read out by reading timer Z primary (TZPR), regardless of whether the timer is counting the timer Z primary reload register or timer Z secondary reload register. Reading timer Z secondary (TZSC) results in an indeterminate value being read out.

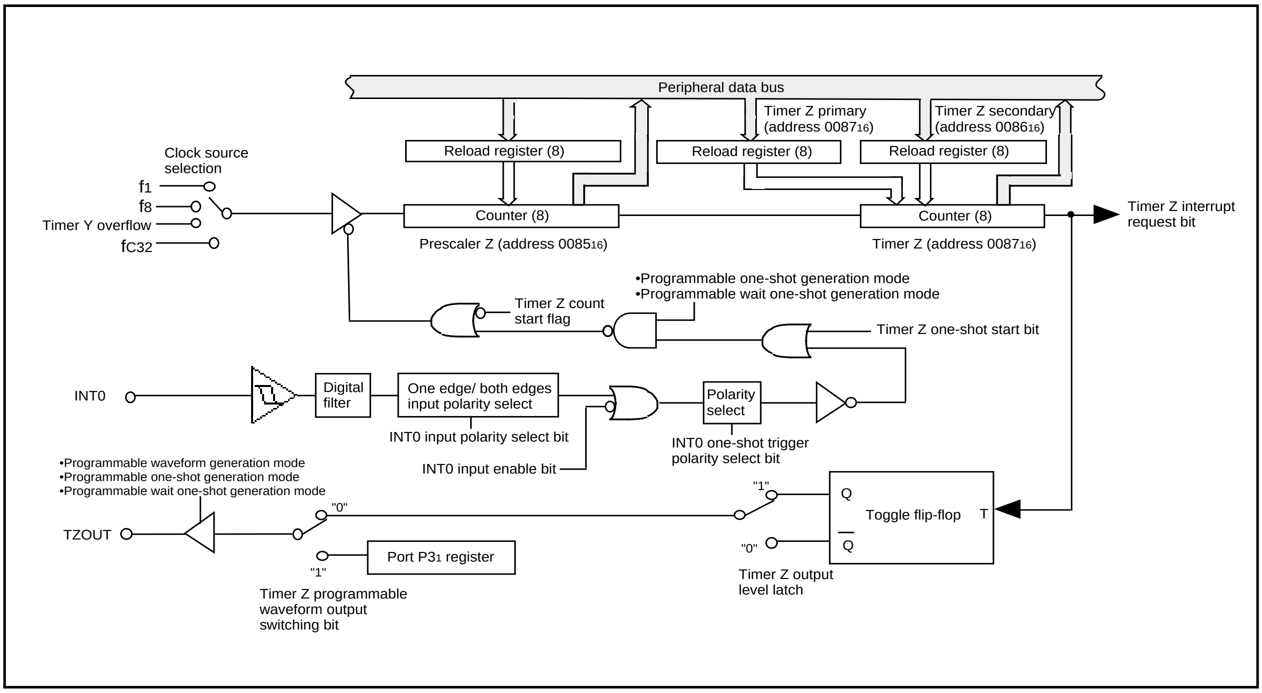

Peripheral data bus
Timer Z primary (address 0087₁₆)
Timer Z secondary (address 0086₁₆)
Clock source selection
Reload register (8)
Reload register (8)
Reload register (8)
f1
f8
Timer Y overflow
fC32
Counter (8)
Counter (8)
Prescaler Z (address 0085₁₆)
Timer Z (address 0087₁₆)
Timer Z interrupt request bit
•Programmable one-shot generation mode
•Programmable wait one-shot generation mode
Timer Z count start flag
Timer Z one-shot start bit
INT0
Digital filter
One edge/ both edges input polarity select
Polarity select
INT0 input polarity select bit
INT0 input enable bit
INT0 one-shot trigger polarity select bit
•Programmable waveform generation mode
•Programmable one-shot generation mode
•Programmable wait one-shot generation mode
"0"
"1"
TZOUT
Port P3₁ register
"1"
"0"
Q
Toggle flip-flop
T
Q
Timer Z programmable waveform output switching bit
Timer Z output level latch

Figure 1.14.7. Block diagram of timer Z

Timer Z has four operation modes that can be selected by setting the Timer Y Mode Register and Timer Z Mode Register.

Timer Y, Z output control register

b7	b6	b5	b4	b3	b2	b1	b0
☒	☒	☒	☒	☒			

Symbol TYZOC — Address 008A$_{16}$ — When reset XXXXX000$_2$

Bit symbol	Bit name	Function	R	W
TZOS	Timer Z one-shot start bit	0 : Stops one-shot 1 : Starts one-shot	○	○
TYOCNT	Timer Y programmable waveform generation output switching bit(Note)	0 : Programmable waveform generation output 1 : P3$_2$ port output	○	○
TZOCNT	Timer Z programmable waveform generation output switching bit (Note)	0 : Programmable waveform generation output 1 : P3$_1$ port output	○	○
Nothing is assigned. When write, set "0". When read, their contents are "0".			—	—

Note: This bit is valid only when operating in the programmable waveform generation mode.

Figure 1.14.8. Timer Y, Z output control register

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
						☒	

Symbol TYZMR — Address 0080$_{16}$ — When reset 000000X0$_2$

Bit symbol	Bit name	Function	R	W
TYMOD0	Timer Y operation mode bit	0 : Timer mode 1 : Programmable waveform generation mode (Note 1)	○	○
Nothing is assigned. When write, set "0". When read, the content is "0".			—	—
TYWC	Timer Y write control bit	0 : Write to timer and reload register simultaneously 1 : Write to reload register	○	○
TYS	Timer Y count start flag	0 : Stops counting 1 : Starts counting	○	○
TZMOD0	Timer Z operation mode bit (Note 2)	b5 b4 0 0 : Timer mode 0 1 : Programmable waveform generation mode 1 0 : Programmable one-shot generation mode 1 1 : Programmable wait one-shot generation mode	○	○
TZMOD1			○	○
TZWC	Timer Z write control bit	0 : Write to timer and reload register simultaneously 1 : Write to reload register	○	○
TZS	Timer Z count start flag	0 : Stops counting 1 : Starts counting	○	○

Note 1: In programmable waveform generation mode, port P3$_2$ is set for output regardless of the value of the direction register.
Note 2: When timer Z operation mode bit is set for "01", "10" or "11", port P3$_1$ is set for output regardless of the value of the direction register.

Timer Y, Z waveform output control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol PUM — Address 0084$_{16}$ — When reset 00$_{16}$

Bit symbol	Bit name	Function	R	W
TYPUM 0	Timer Y primary waveform extension control bit	0 : No waveform extension 1 : Waveform extension	○	○
TYPUM 1	Timer Y secondary waveform extension control bit	0 : No waveform extension 1 : Waveform extension	○	○
TZPUM0	Timer Z primary waveform extension control bit	0 : No waveform extension 1 : Waveform extension	○	○
TZPUM1	Timer Z secondary waveform extension control bit	0 : No waveform extension 1 : Waveform extension	○	○
TYOPL	Timer Y output level latch	0 : Outputs an "L" 1 : Outputs an "H"	○	○
TZOP L	Timer Z output level latch	0 : Outputs an "L" 1 : Outputs an "H"	○	○
INOSTG	INT0 pin one-shot trigger control bit	0 : INT0 pin one-shot trigger invalid 1 : INT0 pin one-shot trigger valid	○	○
INOSEG	INT0 pin one-shot trigger polarity select bit	0 : Edge trigger at falling edge 1 : Edge trigger at rising edge	○	○

Figure 1.14.9. Timer Y, Z mode register and timer Y, Z waveform output control register

(1) Timer mode

Prescaler Z counts the selected count source and each time the count clock is applied, the prescaler has its content decremented by one. After reaching "00_{16}," prescaler Z underflows at the next count clock and is reloaded with the value transferred from the prescaler-Z reload register and starts counting over again. The divide-by ratio of prescaler Z is 1 / (n + 1) where n = set value of prescaler Z.

Timer Z has its content decremented by one each time an underflow signal is received from prescaler Z. After reaching "00_{16}," timer Z underflows at the next count clock and is reloaded with the value transferred from the timer-Z primary reload register and starts counting over again. (In timer mode, the timer always counts the content of the timer Z primary reload register. The timer Z secondary is unused in this mode.)

The divide-by ratio of timer Z is 1 / (m + 1) where m = set value of Timer Z. Therefore, assuming n = set value of prescaler Z and m = set value of timer Z, the divide-by ratio of timer Z is 1 / ((n + 1) x (m + 1)).

In time mode, it is possible to choose whether to write to only the reload registeres of prescaler Z and timer Z primary or both prescaler Z and timer Z and their reload registeres by setting the timer Z write control bit.

In timer mode, an interrupt is generated by an underflow of timer Z.

(2) Programmable waveform generation mode

In programmable waveform generation mode, the microcomputer, while counting the set values of timer Z primary and timer Z secondary alternately, outputs from the TZ_{OUT} pin a waveform whose polarity is inverted each time timer Z underflows.

When operating in this mode, always be sure to set the timer Z write control bit to 1, thereby choosing to write to only the reload registers. In this mode, said pin is always directed for output regardless of how the direction register for the port $P3_1$ that is shared with the output pin is set.

The polarity of the output waveform is set by using the timer Z output level latch. If the timer Z output level latch is set to 0, the microcomputer outputs a high for the period equal to the set value of timer Z primary and a low for the period equal to the set value of timer Z secondary alternately. When the timer starts the output is high at the beginning and when it stops the output returns low. If the timer Z output level latch is set to 1, the microcomputer outputs a low for the period equal to the set value of timer Z primary and a high for the period equal to the set value of timer Z secondary alternately. When the timer starts the output is low at the beginning and when it stops the output returns high. If the Timer Y, Z Output Control Register's timer Z programmable waveform generation output switching bit is set to 1, the Port $P3_1$ Register value is output synchronously when timer Z secondary underflows.

In this mode also, the waveform output primary period and secondary period can each be extended 0.5 cycles of the count source by setting the timer Z primary waveform extension and timer Z secondary waveform extension control bits to 1. This helps to output waveforms with higher resolution.

When using the waveform extension control bits, the frequency and duty cycle of the output waveform become as shown below.

Waveform frequency: FZOUT=(2xTMZCL)/((2xTZPR+1)+2x(TZSC+1)+(EXPZPR+EXPZSC))
Duty: DZOUT=(2x(TZPR+1)+(EXPZPR)/((2x(TZPR+1)+EXPZPR)+(2x(TZSC+1)+EXPZSC))
TMZCL: Timer Z count source (frequency)
TZPR: Timer Z primary (8-bit)
TZSC: Timer Z secondary (8-bit)
EXPZPR: Timer Z primary waveform extension control bit (1 bit)
EXPZSC: Timer Z secondary waveform extension control bit (1 bit)

In programmable waveform generation mode, if the value of the timer Z primary, timer Z secondary, or primary or secondary waveform extension control bit is altered, control is exercised in such a way that the output waveform changes from the beginning of the waveform period (i.e., the timer Z primary waveform period).
To alter the count value, set the timer Z secondary, primary waveform extension control bit, secondary waveform extension control bit, and finally the timer Z primary. The values thus set are reflected collectively at the beginning of the next waveform period after writing to the timer Z primary. (Even when writing while the timer is idle, the timer Z primary is always the last register to be written.)
In programmable waveform generation mode, an interrupt is generated by only an underflow of the secondary period.

■ Precaution

• Change of set count values

When altering the set count values in programmable waveform generation mode, note that the set value of the timer Z secondary and those of the primary and secondary waveform extension control bits are made effective by setting the timer Z primary. Therefore, even when the set value of the timer Z primary does not need to be altered, be sure to write the already set value to the timer Z primary again.

• Use of the waveform extend function

The waveform extend function based on timer Z waveform extension control bits is useful only when the value “00_{16}” is set in prescaler Z. If any other value is set in prescaler Z, always be sure to set the timer Z primary waveform extension and timer Z secondary waveform extension control bits to 0. Note also that if timer Y underflow is selected for the count source, the waveform extend function cannot be used either.

• Timer Z write mode

When operating in programmable waveform generation mode, always be sure to set the timer Z write control bit to 1, thereby choosing to write to the reload register.

(3) Programmable one-shot generation mode

In programmable one-shot generation mode, upon software command or external trigger input, the microcomputer outputs the one-shot pulse based on the timer Z primary set value from the TZ_{OUT} pin. When operating in this mode, always be sure to set the timer Z write control bit to 1, thereby choosing to write to only the reload registers. In this mode, said pin is always directed for output regardless of how the direction register for the port $P3_1$ that is shared with the output pin is set. The timer Z secondary is unused in this mode.
The polarity of the output waveform is set by using the timer Z output level latch. If the timer Z output level latch is set to 0, the microcomputer outputs a high pulse for the period equal to the set value of timer Z primary. If the timer Z output level latch is set to 1, the microcomputer outputs a low pulse for the period equal to the set value of timer Z primary.
In programmable one-shot generation mode, timer Z is readied to receive a software command or a trigger from the external INT0 pin by writing 1 to the timer Z count start flag after setting the count value. When a trigger from the external INT0 pin is used, set the INT0 input enable bit (INT0EN) to "1 ." (Timer Z still remains idle at the time the timer Z count start flag is set to 1.)

Timer Z starts counting and the TZOUT pin output is inverted simultaneously with it by writing 1 to the timer Z one-shot start bit or writing 1 to the INT0 pin one-shot trigger control bit to enable the INT0 pin trigger and applying a valid trigger to the INT0 pin. When timer Z underflows, the TZOUT pin output is inverted again and timer Z stops. The one-shot start bit is also set to 1 in hardware when an INT0 pin trigger is received. When the INT0 input polarity bit (INT0PL) is set to one edge "0", the active trigger edge on the INT0 pin can be chosen to be the falling or the rising edge by using the INT0 pin one-shot trigger polarity select bit. When both rising and falling edges are set as an active trigger edge, set INT0PL to "1".

One-shot pulse output can be forcibly turned off by writing 0 to the timer Z one-shot start bit during a one-shot pulse output period.

In programmable one-shot pulse generation mode, the set values are reflected collectively beginning with the next one-shot pulse after writing to the timer Z primary. (Even when writing while the timer is idle, the timer Z primary is always the last register to be written.)

■ Precaution

• Change of set count values

When altering the set count values in programmable one-shot pulse generation mode, note that the set value of the primary waveform extension control bit is made effective by setting the timer Z primary. Therefore, even when the set value of the timer Z primary does not need to be altered, be sure to write the already set value to the timer Z primary again.

• Use of the waveform extend function

The waveform extend function based on timer Z waveform extension control bits is useful only when the value “00_{16}” is set in prescaler Z. If any other value is set in prescaler Z, always be sure to set the timer Z primary waveform extension control bit to 0. Note also that if timer Y underflow is selected for the count source, the waveform extend function cannot be used either.

• Timer Z write mode

When operating in programmable one-shot pulse generation mode, always be sure to set the timer Z write control bit to 1, thereby choosing to write to the reload register.

(4) Programmable wait one-shot generation mode

In programmable wait one-shot generation mode, upon software command or external trigger input, the microcomputer outputs the one-shot pulse based on the timer Z secondary set value from the TZOUT pin after waiting for a time equal to the timer Z primary set value.

When operating in this mode, always be sure to set the timer Z write control bit to 1, thereby choosing to write to only the reload registeres. In this mode, said pin is always directed for output regardless of how the direction register for the port $P3_1$ that is shared with the output pin is set.

The polarity of the output waveform is set by using the timer Z output level latch. If the timer Z output level latch is set to 0, the microcomputer outputs a high pulse for the period equal to the set value of timer Z secondary after waiting for a time equal to the timer Z primary set value. If the timer Z output level latch is set to 1, the microcomputer outputs a low pulse for the period equal to the set value of timer Z secondary after waiting for a time equal to the timer Z primary set value.

In this mode also, the wait period and one-shot pulse period can be extended 0.5 cycles of the count source by setting the timer Z primary waveform extension and timer Z secondary waveform extension control bits to 1. This helps to output waveforms with higher resolution.

In programmable wait one-shot generation mode, timer Z is readied to receive a software command or a trigger from the external INT0 pin by writing 1 to the timer Z count start flag after setting the count

value. When a trigger from the external INT0 pin is used, set the INT0 input enable bit (INT0EN) to "1 ". (Timer Z still remains idle at the time the timer Z count start flag is set to 1.)
Timer Z starts counting by writing 1 to the timer Z one-shot start bit or writing 1 to the INT0 pin one-shot trigger control bit to enable the INT0 pin trigger and applying a valid trigger to the INT0 pin. While timer Z is counting the timer Z primary, the TZOUT pin output retains its initial value intact. When the timer underflows, it is reloaded with the timer Z secondary and the TZOUT pin output is inverted simultaneously with it. The next time timer Z underflows, the TZOUT pin output is inverted again and the timer stops. The one-shot start bit is also set to 1 in hardware when an INT0 pin trigger is received. When the INT0 input polarity bit (INT0PL) is set to one edge "0", the active trigger edge on the INT0 pin can be chosen to be the falling or the rising edge by using the INT0 pin one-shot trigger polarity select bit. When both rising and falling edges are set as an active trigger edge, set INT0PL to "1".
One-shot pulse output can be forcibly turned off by writing 0 to the timer Z one-shot start bit during a wait period or a one-shot pulse output period.
To alter the set values in programmable wait one-shot pulse generation mode, set the timer Z secondary and then the timer Z primary. The set values are reflected collectively beginning with the next wait period after writing to the timer Z primary. (Even when writing while the timer is idle, the timer Z primary is always the last register to be written.)

■ Precaution

• Change of set count values

When altering the set count values in programmable wait one-shot generation mode, note that the set value of the timer Z secondary, the primary waveform extension and secondary waveform extension control bits are made effective by setting the timer Z primary. Therefore, even when the set value of the timer Z primary does not need to be altered, be sure to write the already set value to the timer Z primary again.

• Use of the waveform extend function

The waveform extend function based on timer Z waveform extension control bits is useful only when the value “00_{16}” is set in prescaler Z. If any other value is set in prescaler Z, always be sure to set the timer Z primary waveform extension and timer Z secondary waveform extension control bits to 0. Note also that if timer Y underflow is selected for the count source, the waveform extend function cannot be used either.

• Timer Z write mode

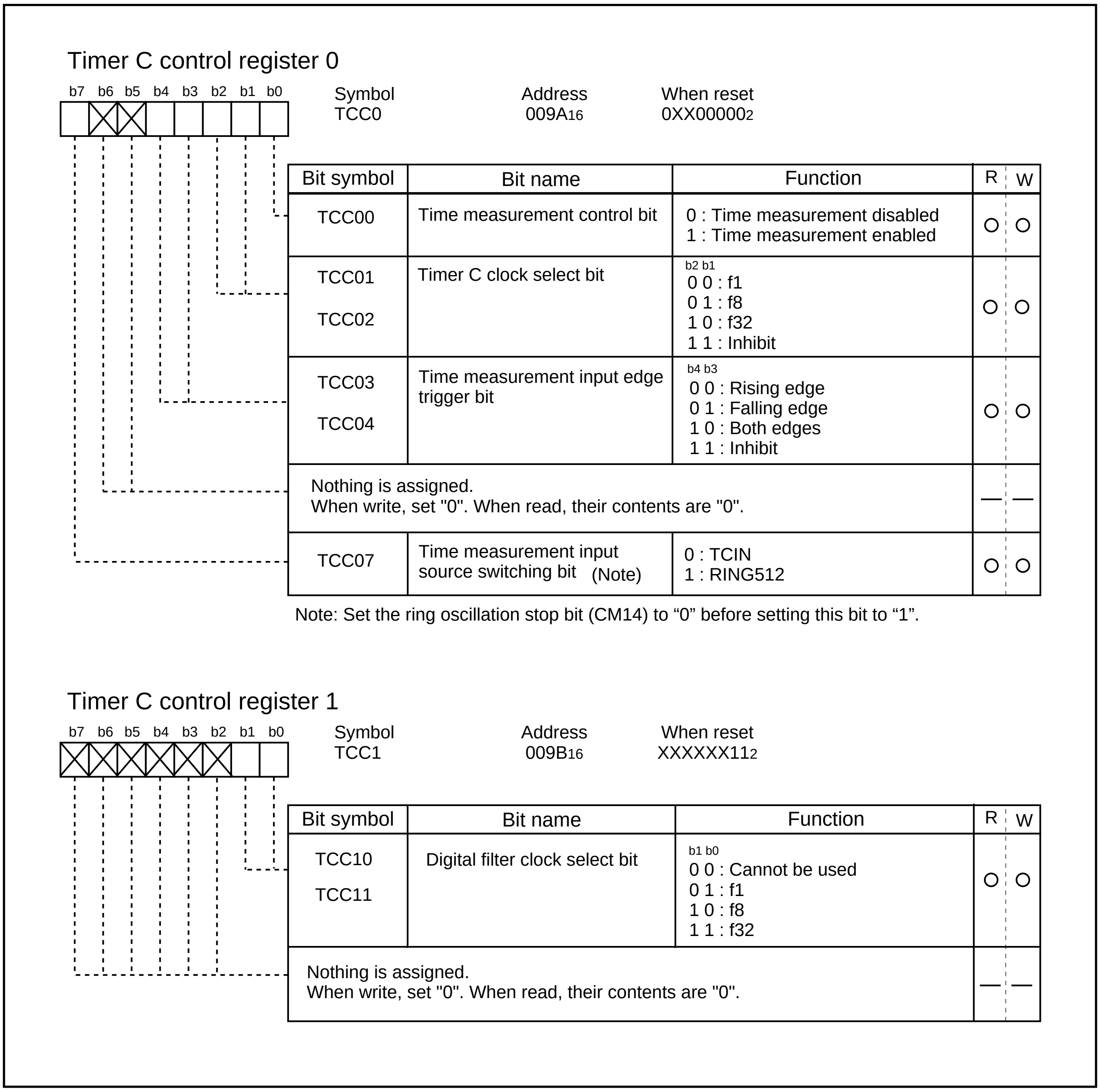
When operating in programmable wait one-shot pulse generation mode, always be sure to set the timer Z write control bit to 1, thereby choosing to write to the reload register.

In any operation mode, timer Z can be made to stop counting by setting the timer Z count start flag to 0. When timer Z underflows, the timer Z interrupt request bit is set to 1.

Timer C

The CPU uses edge detection of the pin input to latch the counter value of the 16-bit free-run timer and generate an interrupt.

- The count source for the 16-bit free-run timer can be selected from three settings: XIN/1, XIN/8, and XIN/32.
- The input edge detection setting can be selected from three settings: rising edge, falling edge, and both edges.
- An interrupt is generated when an overflow occurs and when the specified edge is detected.
- The input source for time measurement input trigger can be selected from TCIN and 512 ring oscillation divisions.



Timer C control register 0

Symbol	Address	When reset
TCC0	009A$_{16}$	0XX00000$_2$

Bit symbol	Bit name	Function	R	W
TCC00	Time measurement control bit	0 : Time measurement disabled 1 : Time measurement enabled	○	○
TCC01 TCC02	Timer C clock select bit	b2 b1 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : Inhibit	○	○
TCC03 TCC04	Time measurement input edge trigger bit	b4 b3 0 0 : Rising edge 0 1 : Falling edge 1 0 : Both edges 1 1 : Inhibit	○	○
Nothing is assigned. When write, set "0". When read, their contents are "0".			—	—
TCC07	Time measurement input source switching bit (Note)	0 : TCIN 1 : RING512	○	○

Note: Set the ring oscillation stop bit (CM14) to “0” before setting this bit to “1”.

Timer C control register 1

Symbol	Address	When reset
TCC1	009B$_{16}$	XXXXXX11$_2$

Bit symbol	Bit name	Function	R	W
TCC10 TCC11	Digital filter clock select bit	b1 b0 0 0 : Cannot be used 0 1 : f1 1 0 : f8 1 1 : f32	○	○
Nothing is assigned. When write, set "0". When read, their contents are "0".			—	—

Figure 1.14.10. Timer C control register

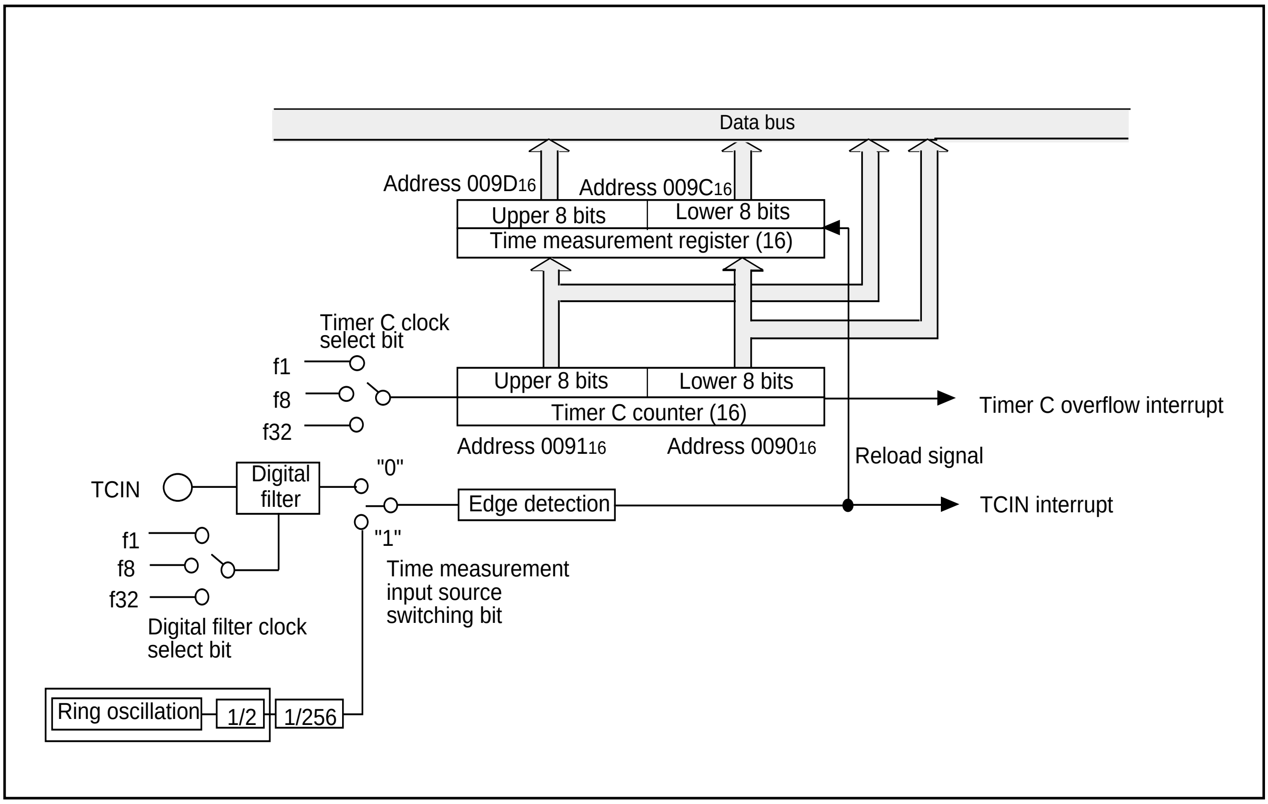


Figure 1.14.11. Block diagram of timer C control register

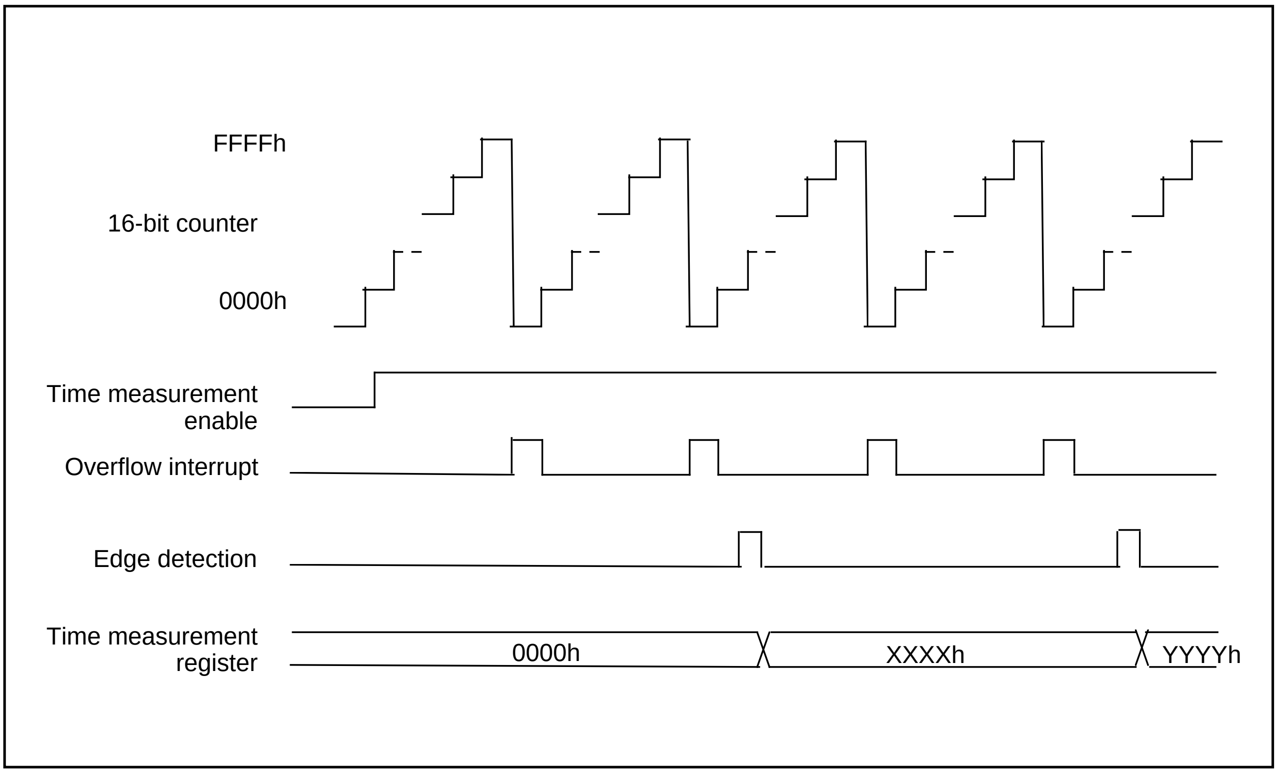


Figure 1.14.12. Timing chart of time measurement

Table 1.14.1. Specifications of Timer C

Item	Specification
Count source	f_1, f_8, f_{32}
Mesurement source	• TCIN or 512 divisions of ring oscillation
Count operation	• Up count • Transfer counter value to time measurement register at effective edge of measurement pulse • Do not reset counter value even if effective edge is detected
Count start condition	• Enables time measurement control bit
Counter stop and counter value reset	• Disables time measurement control bit
Interrupt request generation timing	• When effective edge of measurement pulse is input • When the time underflows
Effective edge detection enable pulse width	• f_1/2 and more
Digital filter sampling frequency	f_1, f_8, f_{32} (When latching the same value three times, the change of input becomes effective)

Note: Disables time measurement before changing the count source clock or the edge trigger select bit.

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 1.15.1 shows the block diagram of UARTi (i=0,1). Figure 1.15.2 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 00A0$_{16}$ and 00A8$_{16}$) determine whether UART0 is used as a clock synchronous serial I/O or as a UART. Although a few functions are different, UART0 and UART1 have almost the same functions.

Figures 1.15.3 through 1.15.5 show the registers related to UARTi.

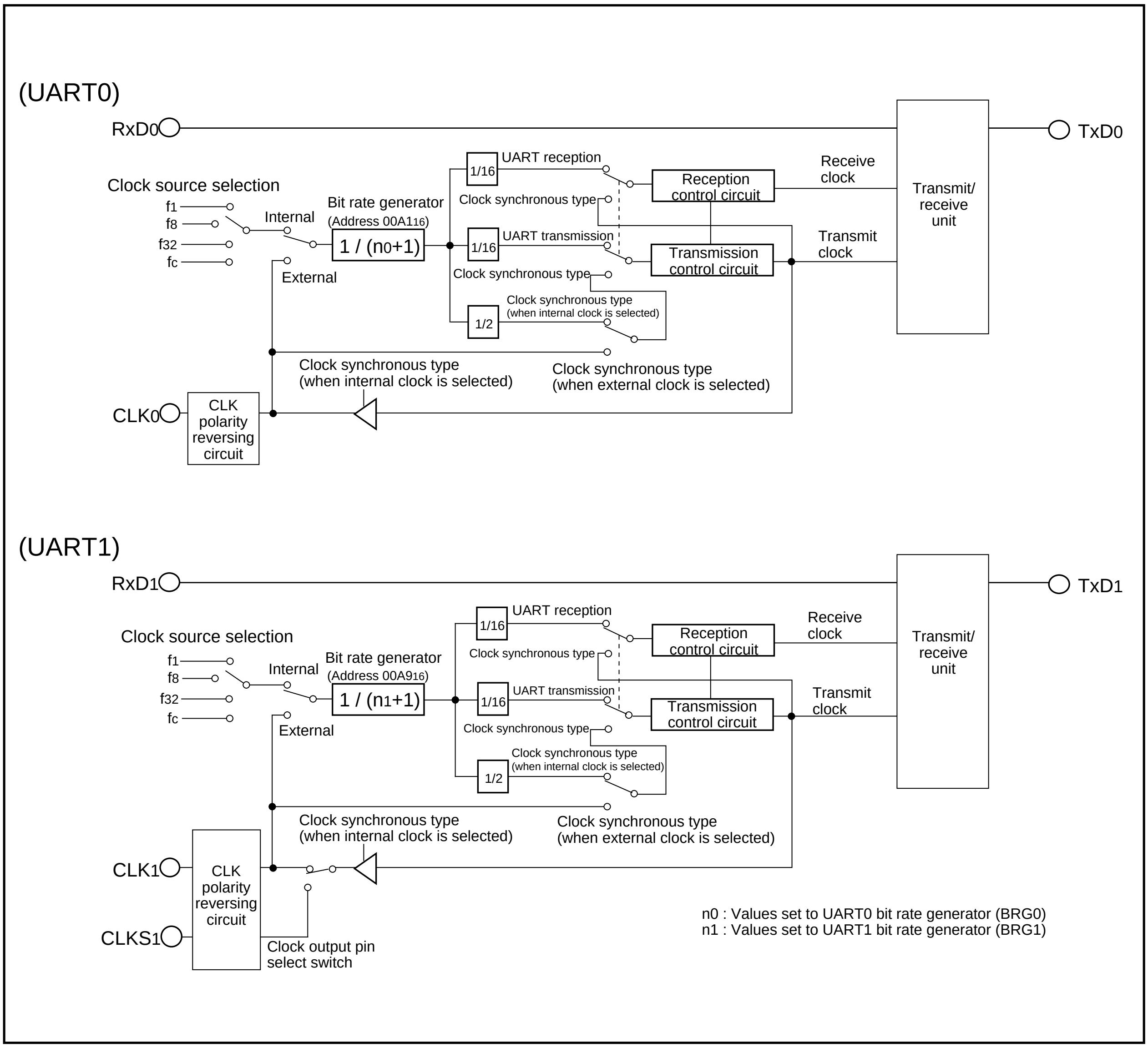


Figure 1.15.1. Block diagram of UARTi (i = 0, 1)

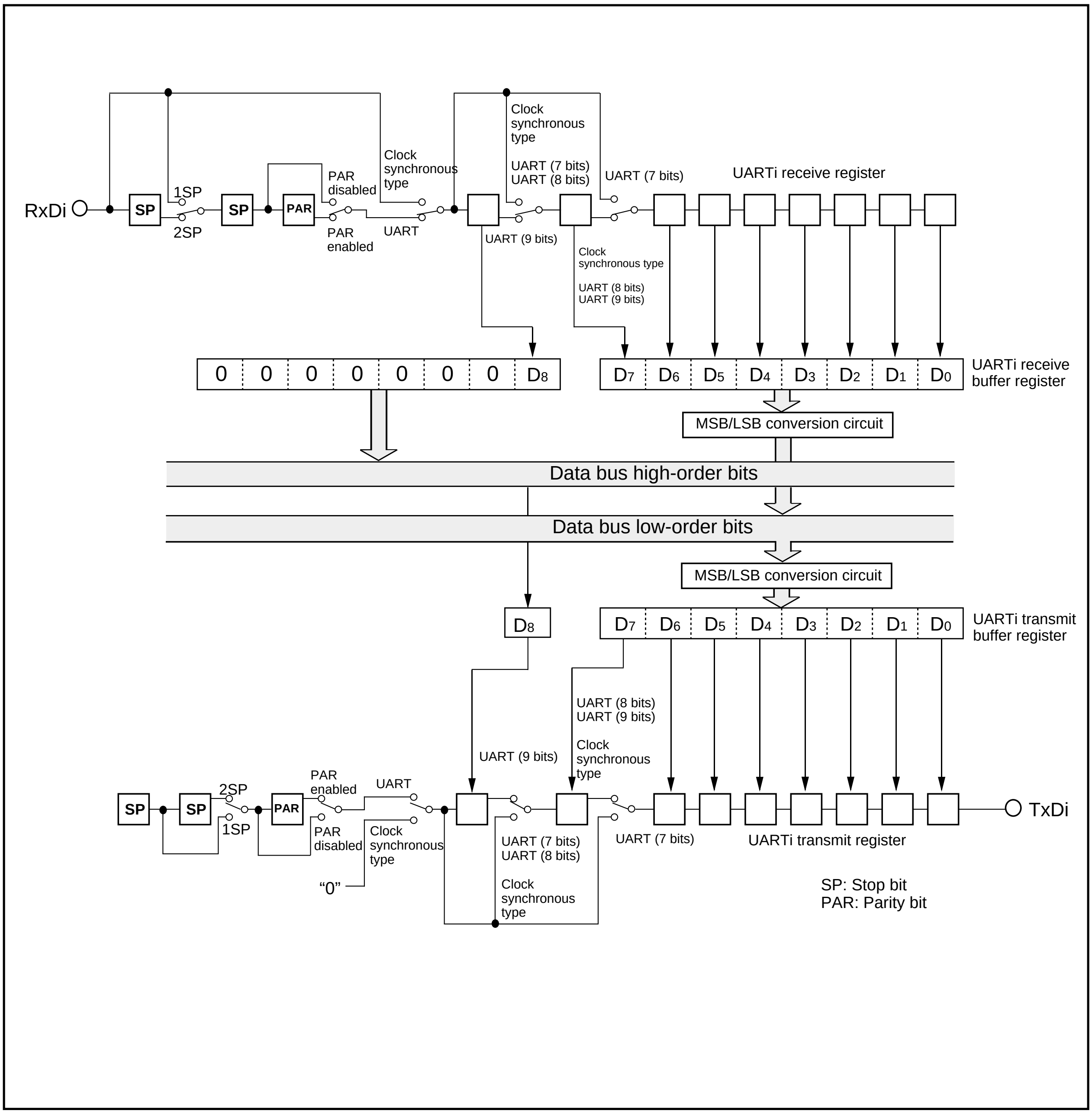
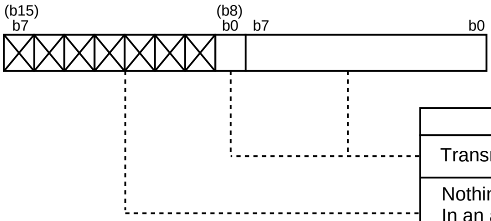


Figure 1.15.2. Block diagram of transmit/receive unit

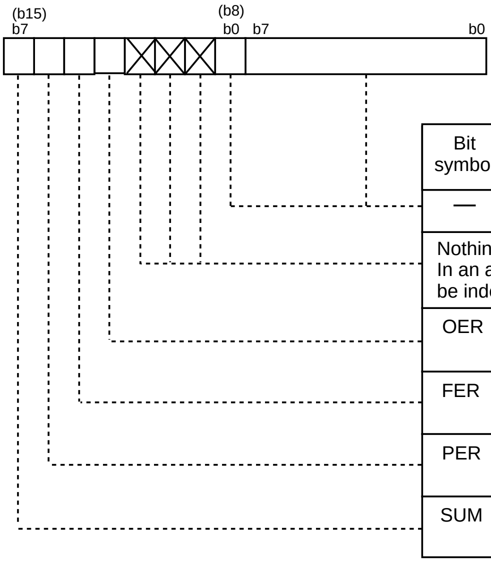
UARTi transmit buffer register



Symbol	Address	When reset
U0TB	$00A3_{16}$, $00A2_{16}$	Indeterminate
U1TB	$00AB_{16}$, $00AA_{16}$	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.	–	–

UARTi receive buffer register

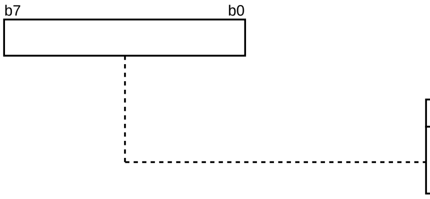


Symbol	Address	When reset
U0RB	$00A7_{16}$, $00A6_{16}$	Indeterminate
U1RB	$00AF_{16}$, $00AE_{16}$	Indeterminate

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 serial I/O mode select bits (bit 2 to 0 at addresses $00A0_{16}$ and $00A8_{16}$) are set to “000_2” or receive enable bit 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 $00A6_{16}$, and $00AE_{16}$) is read out or when this register is read out with a word-sized instruction.

UARTi bit rate generator



Symbol	Address	When reset
U0BRG	$00A1_{16}$	Indeterminate
U1BRG	$00A9_{16}$	Indeterminate

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

Figure 1.15.3. Serial I/O-related registers (1)

UARTi transmit/receive mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiMR(i=0,1)	$00A0_{16}$, $00A8_{16}$	00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit	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	0 : Internal clock 1 : External clock (Note)	0 : Internal clock 1 : External clock	○	○
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep select bit	Must always be “0”	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

Note: Set the corresponding port direction register to “0”.

UARTi transmit/receive control register 0

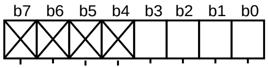
b7 b6 b5 b4 b3 b2 b1 b0
□ □ □ 1 □ 0 □ □

Symbol	Address	When reset
UiC0(i=0,1)	$00A4_{16}$, $00AC_{16}$	08_{16}

Bit symbol	Bit name	Function (Note) (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : fc is selected	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : fc is selected	○	○
CLK1				○	○
Set this bit to "0".				○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
Set this bit to "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”	○	○

Figure 1.15.4. Serial I/O-related registers (2)

UARTi transmit/receive control register 1

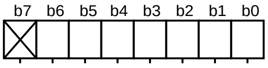


Symbol	Address	When reset
UiC1(i=0,1)	00A5$_{16}$,00AD$_{16}$	02$_{16}$

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	○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	○	×
RE	Receive enable bit (Note)	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	○	○
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	○	×
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".				—	—

Note : As for the UART1, set the RXD1 input port select bit before setting this bit to reception enabled. When the receive enable bit is set for enabled in M30100, select P3$_7$ with RXD1 input port select bit.

UART transmit/receive control register 2



Symbol	Address	When reset
UCON	00B0$_{16}$	00$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U1IRS	UART1 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	Invalid	○	○
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	Invalid	○	○
CLKMD0	CLK/CLKS select bit 0	Valid when bit 5 = “1” 0 : Clock output to CLK1 1 : Clock output to CLKS1	Invalid	○	○
CLKMD1	CLK/CLKS select bit 1 (Note 1)	0 : Normal mode (CLK output is CLK0 only) 1 : Transfer clock output from multiple pins function selected	Fixed to “0”	○	○
RXD1EN	RXD1 input port select bit	0 : P3$_7$ 1 : P3$_5$	0 : P3$_7$ 1 : P3$_5$	○	○
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".				—	—

Note 1: When using multiple pins to output the transfer clock, the following requirements must be met:
- UART1 internal/external clock select bit (bit 3 at address 00A0$_{16}$) = “0”.

Figure 1.15.5. Serial I/O-related registers (3)

(1) Clock synchronous serial I/O mode

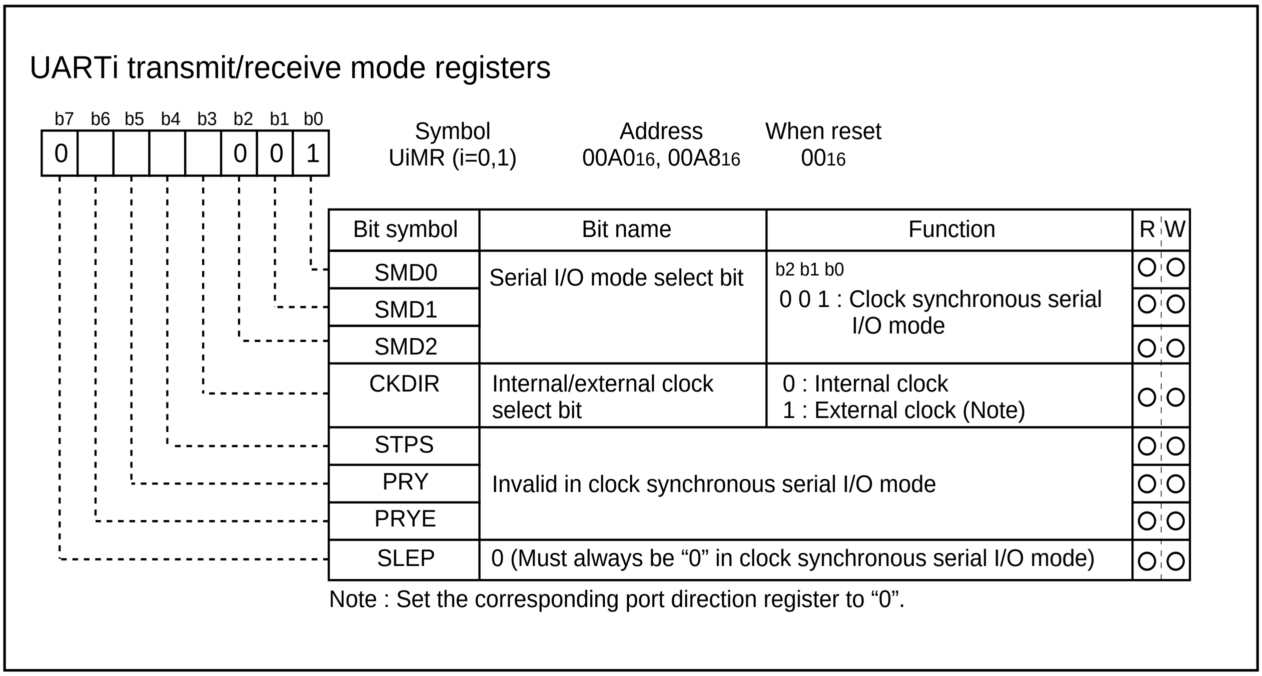
The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. (See Table 1.15.1.) Figure 1.15.6 shows the UARTi transmit/receive mode register.

Table 1.15.1. Specifications of clock synchronous serial I/O mode

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• When internal clock is selected (bit 3 at address 00A0$_{16}$,00A8$_{16}$ = “0”) : fi/ 2(n+1) (Note 1) fi = f$_1$, f$_8$, f$_{32}$, fc • When external clock is selected (bit 3 at address 00A0$_{16}$,00A8$_{16}$ = “1”) : Input from CLKi pin
Transmission start condition	• To start transmission, the following requirements must be met: – Transmit enable bit (bit 0 at address 00A5$_{16}$,00AD$_{16}$) = “1” – Transmit buffer empty flag (bit 1 at addresses 00A5$_{16}$,00AD$_{16}$) = “0” • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at address 00A4$_{16}$,00AC$_{16}$) = “0”: CLKi input level = “H” – CLKi polarity select bit (bit 6 at address 00A4$_{16}$,00AC$_{16}$) = “1”: CLKi input level = “L”
Reception start conditio	• To start reception, the following requirements must be met: – Receive enable bit (bit 2 at address 00A5$_{16}$,00AD$_{16}$) = “1” – Transmit enable bit (bit 0 at address 00A5$_{16}$,00AD$_{16}$) = “1” – Transmit buffer empty flag (bit 1 at address 00A5$_{16}$,00AD$_{16}$) = “0” • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at address 00A4$_{16}$,00AC$_{16}$) = “0”: CLKi input level = “H” – CLKi polarity select bit (bit 6 at address 00A4$_{16}$,00AC$_{16}$) = “1”: CLKi input level = “L”
Interrupt request generation timing	• When transmitting – Transmit interrupt cause select bit (bit 0 and bit 1 at address 00B0$_{16}$) = “0”: Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed – Transmit interrupt cause select bit (bit 0 and bit 1 at address 00B0$_{16}$) = “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 2) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out
Select function	• CLK polarity selection Whether transmit data is output/input at the rising edge or falling edge of the transfer clock can be selected • LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected • Continuous receive mode selection Reception is enabled simultaneously by a read from the receive buffer register • Transfer clock output from multiple pins selection UART1 transfer clock can be chosen by software to be output from one of the two pins set • RxD$_1$ input pin selection UART1 RxD$_1$ can be chosen by software to be input to 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 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

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

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note)	○	○
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 : Set the corresponding port direction register to “0”.

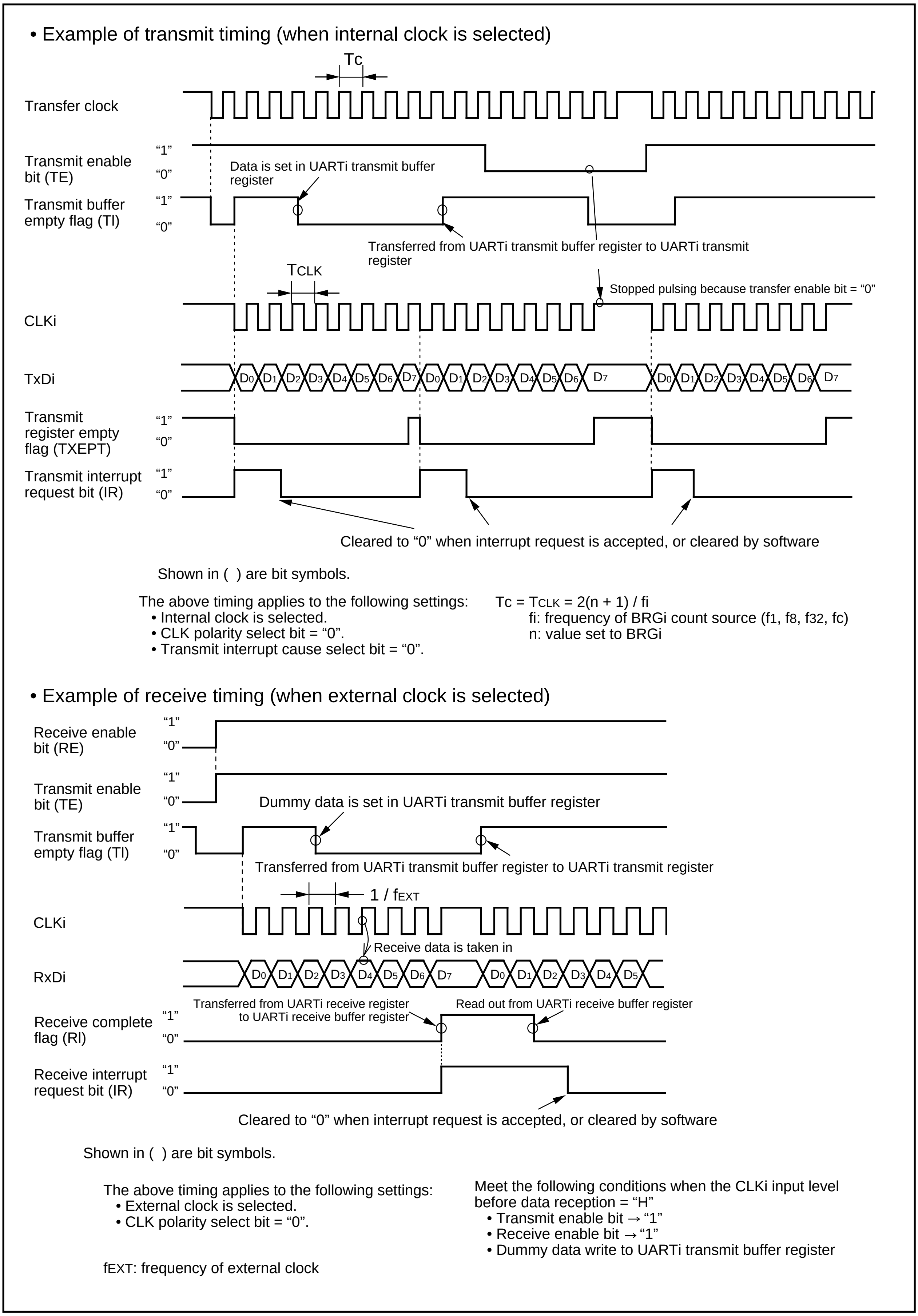
Figure 1.15.6. UARTi transmit/receive mode register in clock synchronous serial I/O mode

Table 1.15.2 lists the functions of the input/output pins during clock synchronous serial I/O mode. This table shows the pin functions when the transfer clock output from multiple pins is not selected. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an “H”. (If the N-channel open-drain is selected, this pin is in floating state.)

Table 1.15.2. Input/output pin functions in clock synchronous serial I/O mode

Pin name	Function	Method of selection
TxDi (P1$_4$, P3$_7$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P1$_5$, P3$_5$, P3$_7$)	Serial data input	Port P1$_5$, P3$_5$ and P3$_7$ direction register (bit 5 at address 00E1$_{16}$, bit 5 and 7 at address 00E7$_{16}$)= “0” (Can be used as an I/O port when performing transmission only)
CLKi (P1$_6$, P3$_6$)	Transfer clock output	Internal/external clock select bit (bit 3 at address 00A0$_{16}$ and 00A8$_{16}$) = “0”
	Transfer clock input	Internal/external clock select bit (bit 3 at address 00A0$_{16}$ and 00A8$_{16}$) = “1” Port P1$_6$ and P3$_6$ direction register (bit 6 at address 00E3$_{16}$ and 00E7$_{16}$) = “0”

(When transfer clock output from multiple pins is not selected)


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

- 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".

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

- 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".

fEXT: frequency of external clock

Meet the following conditions when the CLKi input level before data reception = "H"
- Transmit enable bit → "1"
- Receive enable bit → "1"
- Dummy data write to UARTi transmit buffer register

Figure 1.15.7. Typical transmit/receive timings in clock synchronous serial I/O mode

(a) Polarity select function

As shown in Figure 1.15.8, the CLK polarity select bit (bit 6 at addresses 00A4$_{16}$ and 00AC$_{16}$) allows selection of the polarity of the transfer clock.

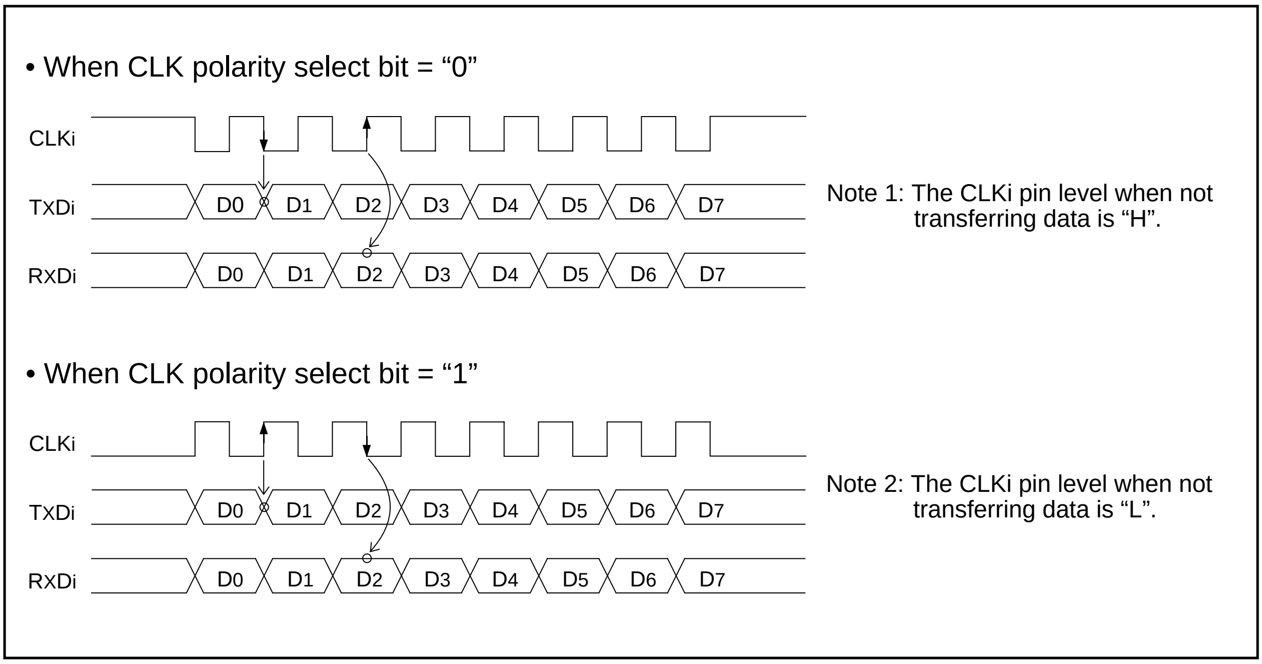

• When CLK polarity select bit = “0”
CLKi
TXDi
RXDi
D0
D1
D2
D3
D4
D5
D6
D7
Note 1: The CLKi pin level when not transferring data is “H”.
• When CLK polarity select bit = “1”
CLKi
TXDi
RXDi
Note 2: The CLKi pin level when not transferring data is “L”.

Figure 1.15.8. Polarity of transfer clock

(b) LSB first/MSB first select function

As shown in Figure 1.15.9, when the transfer format select bit (bit 7 at addresses 00A4$_{16}$ and 00AC$_{16}$) = “0”, the transfer format is “LSB first”; when the bit = “1”, the transfer format is “MSB first”.

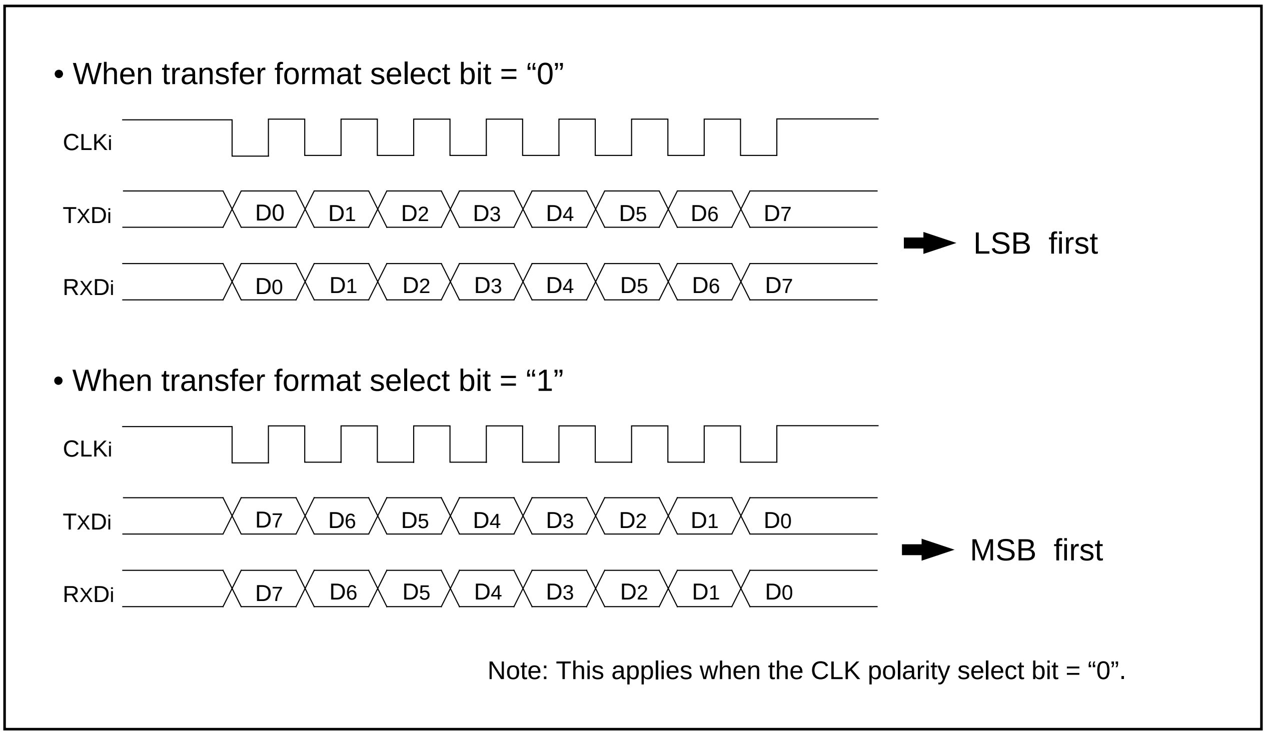
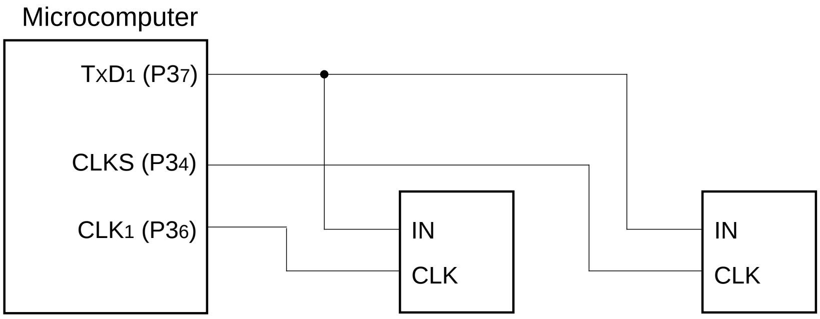

• When transfer format select bit = “0”
CLKi
TXDi
RXDi
D0
D1
D2
D3
D4
D5
D6
D7
LSB first
• When transfer format select bit = “1”
CLKi
TXDi
RXDi
D7
D6
D5
D4
D3
D2
D1
D0
MSB first
Note: This applies when the CLK polarity select bit = “0”.

Figure 1.15.9. Transfer format

(c) Transfer clock output from multiple pins function (UART1)

This function allows the setting two transfer clock output pins and choosing one of the two to output a clock by using the CLK and CLKS select bit (bits 4 and 5 at address $00B0_{16}$). (See Figure 1.15.10.) The multiple pins function is valid only when the internal clock is selected for UART1.



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

Figure 1.15.10. The transfer clock output from the multiple pins function usage

(d) Continuous receive mode

If the continuous receive mode enable bit (bits 2 and 3 at address $00B0_{16}$) 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.

(e) RxD1 input pin selection function (UART1)

This function allows the setting two RxD1 input pins and choosing one of the two to input serial data by using the RxD1 input pin select bit (bits 6 at address $00B0_{16}$).

(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 1.15.3.) Figure 1.15.11 shows the UARTi transmit/receive mode register.

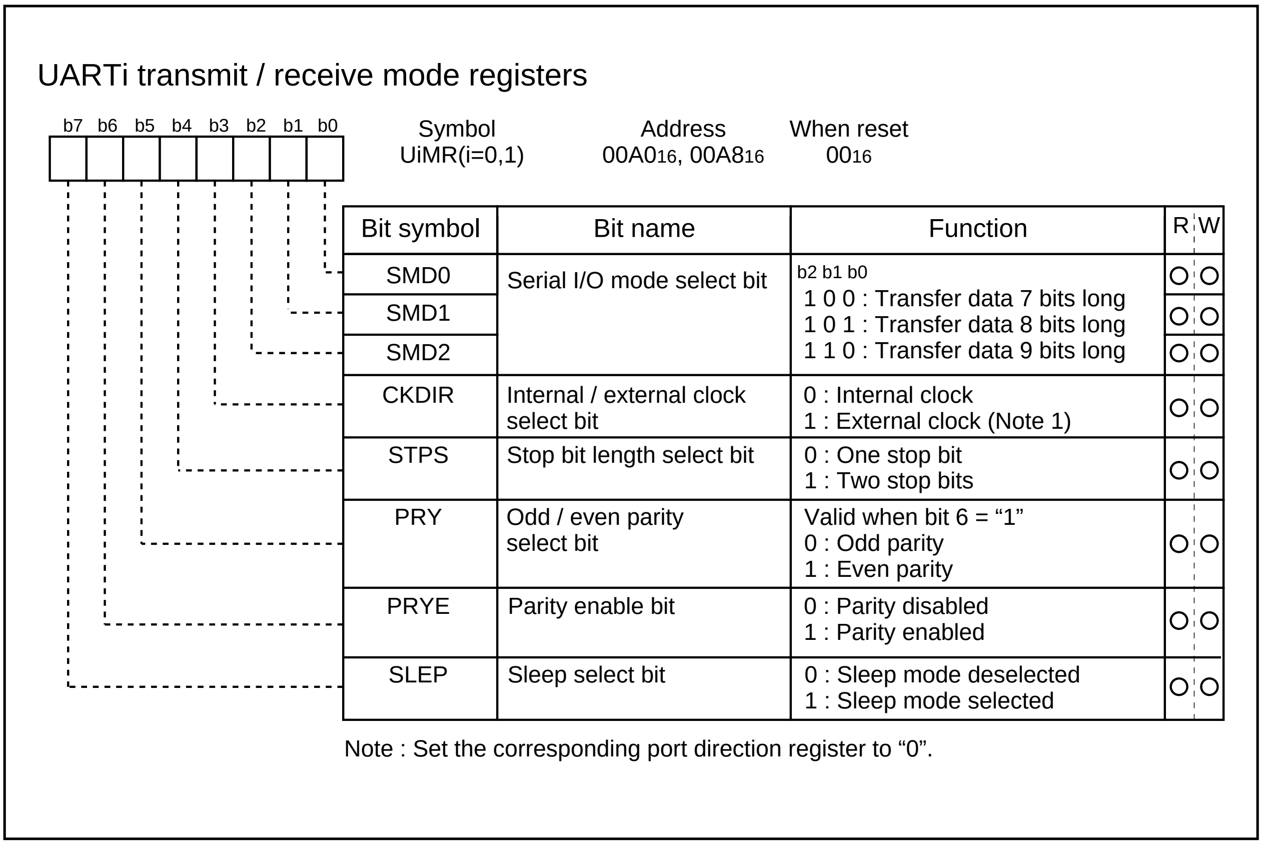
Table 1.15.3. Specifications of UART Mode

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 $00A0_{16}$, $00A8_{16}$ = “0”) : fi/16(n+1) (Note 1) fi = f_1, f_8, f_{32}, f_C • When external clock is selected (bit 3 at addresses $00A0_{16}$=“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 $00A5_{16}$, $00AD_{16}$) = “1” - Transmit buffer empty flag (bit 1 at addresses $00A5_{16}$, $00AD_{16}$) = “0”
Reception start condition	• To start reception, the following requirements must be met: - Receive enable bit (bit 2 at addresses $00A5_{16}$, $00AD_{16}$) = “1” - Start bit detection
Interrupt request generation timing	• When transmitting - Transmit interrupt cause select bits (bits 0,1 at address $00B0_{16}$) = “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 $00B0_{16}$) = “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 • RxD_1 input pin selection UART1 RxD_1 can be chosen by software to be input to 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: f_{EXT} is input from the CLKi pin.

Note 3: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit does not change.



UARTi transmit / receive mode registers

Symbol	Address	When reset
UiMR(i=0,1)	$00A0_{16}$, $00A8_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal / external clock select bit	0 : Internal clock 1 : External clock (Note 1)	○	○
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 : Set the corresponding port direction register to “0”.

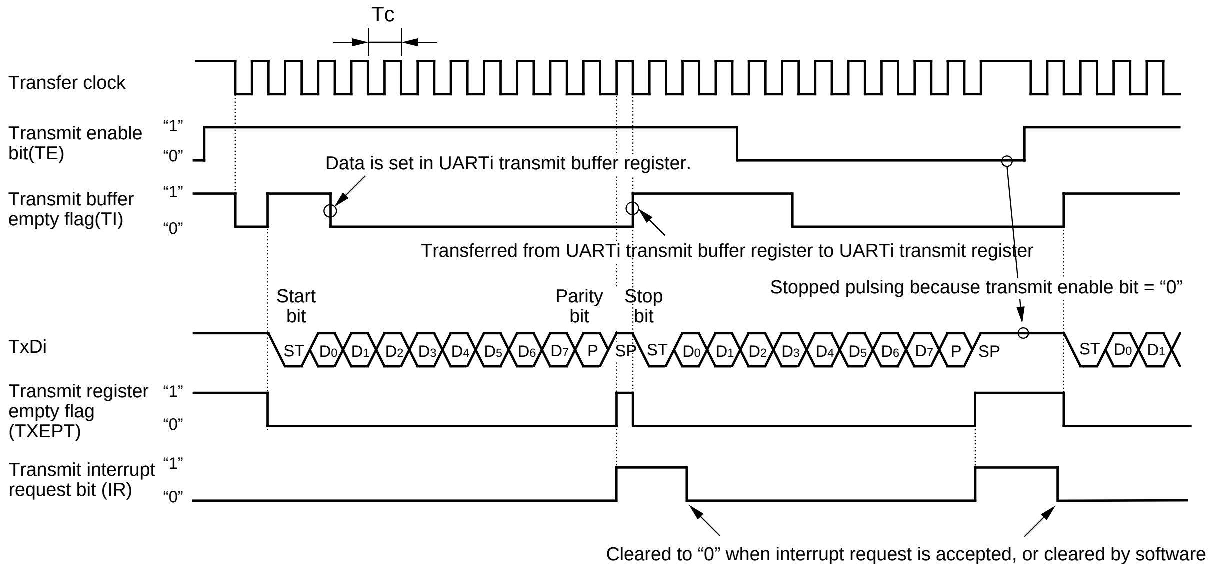
Figure 1.15.11. UARTi transmit/receive mode register in UART mode

Table 1.15.4 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 1.15.4. Input/output pin functions in UART mode

Pin name	Function	Method of selection
TxDi ($P1_4$, $P3_7$)	Serial data output	
RxDi ($P1_5$, $P3_5$, $P3_7$)	Serial data input	Port $P1_5$, $P3_5$ and $P3_7$ direction register (bit 5 at address $00E3_{16}$, bits 5 and 7 at address $00E7_{16}$)= “0” (Can be used as an I/O port when performing transmission only)
CLKi ($P1_6$, $P3_6$)	Programmable I/O port	Internal/external clock select bit (bit 3 at addresses $00A0_{16}$ and $00A8_{16}$) = “0”
	Transfer clock input	Internal/external clock select bit (bit 3 at addresses $00A0_{16}$ and $00A8_{16}$) = “1” Port $P1_6$ and $P3_6$ direction register (bit 6 at addresses $00E3_{16}$ $00E7_{16}$)= “0”

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

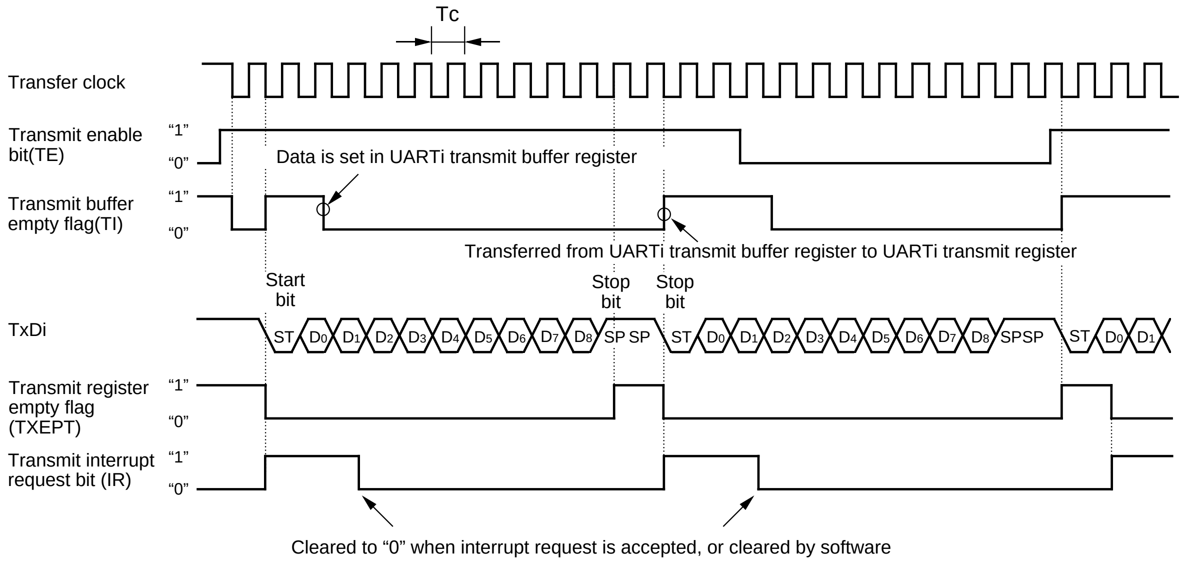

Tc
Transfer clock
Transmit enable bit(TE)
"1"
"0"
Data is set in UARTi transmit buffer register.
Transmit buffer empty flag(TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
Stopped pulsing because transmit enable bit = "0"
Start bit
Parity bit
Stop bit
TxDi
ST D0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

Shown in ( ) are bit symbols.

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

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

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


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

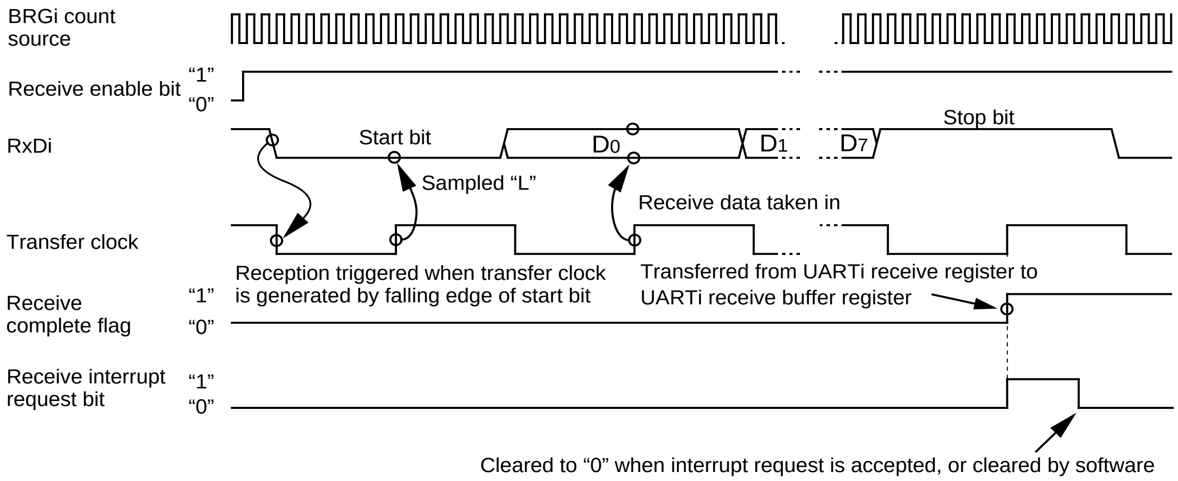
Shown in ( ) are bit symbols.

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

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

Figure 1.15.12. Typical transmit timings in UART mode

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


BRGi count source
Receive enable bit
"1"
"0"
RxDi
Start bit
D0
D1
D7
Stop bit
Sampled "L"
Receive data taken in
Transfer clock
Reception triggered when transfer clock is generated by falling edge of start bit
Transferred from UARTi receive register to UARTi receive buffer register
Receive complete flag
"1"
"0"
Receive interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

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

Figure 1.15.13. 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 00A016, 00A816) 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".

(b) RxD$_1$ input pin selection function (UART1)

This function allows the setting two RxD$_1$ input pins and choosing one of the two to input serial data by using the RxD$_1$ input pin select bit (bits 6 at address 00B0$_{16}$).

A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins $P0_0$ to $P0_7$, $P1_0$ to $P1_3$, $P4_0$ and $P4_1$ 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 $00D7_{16}$) can be used to isolate the resistance ladder of the A-D converter from the reference voltage input pin (VREF) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from VREF, reducing the power dissipation. When using the A-D converter, start A-D conversion only after connecting to VREF.

The result of A-D conversion is stored in the A-D registers. 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 1.16.1 shows the performance of the A-D converter. Figure 1.16.1 shows the block diagram of the A-D converter, and Figures 1.16.2 and 1.16.3 show the A-D converter-related registers.

Table 1.16.1. Performance of A-D converter

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to VCC

Operating clock ϕAD (Note 2)	VCC = 5V fAD, divide-by-2 of fAD, divide-by-4 of fAD, fAD=f(XIN)
	VCC = 3V divide-by-2 of fAD, divide-by-4 of fAD, fAD=f(XIN)

Resolution	8-bit or 10-bit (selectable)
Absolute precision	VCC = 5V • Without sample and hold function ±3LSB • With sample and hold function (8-bit resolution) ±2LSB • With sample and hold function (10-bit resolution) AN_0 to AN_{11} input : ±3LSB $ANEX_0$ and $ANEX_1$ input (including mode in which external operation amp is connected) : ±7LSB VCC = 3V • Without sample and hold function (8-bit resolution) ±2LSB
Operating modes	One-shot mode and repeat mode (Note 3)
Analog input pins	12 pins (AN_0 to AN_{11}) + 2 pins ($ANEX_0$ to $ANEX_1$)
A-D conversion start condition	• Software trigger A-D conversion starts when the A-D conversion start flag changes to “1”
Conversion speed per pin	• Without sample and hold function 8-bit resolution: 49 ϕAD cycles, 10-bit resolution: 59 ϕAD cycles • With sample and hold function 8-bit resolution: 28 ϕAD cycles, 10-bit resolution: 33 ϕAD cycles

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

Note 2: Divide fAD if (XIN) exceeds 10MHz, and make ϕAD equal to or lower than 10MHz. Also if Vcc is less than 4.2V, divide fAD and make ϕAD equal to or lower than fAD/2.
Without sample and hold function, set the ϕAD frequency to 250kHz min.
With the sample and hold function, set the ϕAD frequency to 1MHz min.

Note 3: In repeat mode, only 8-bit mode can be used.

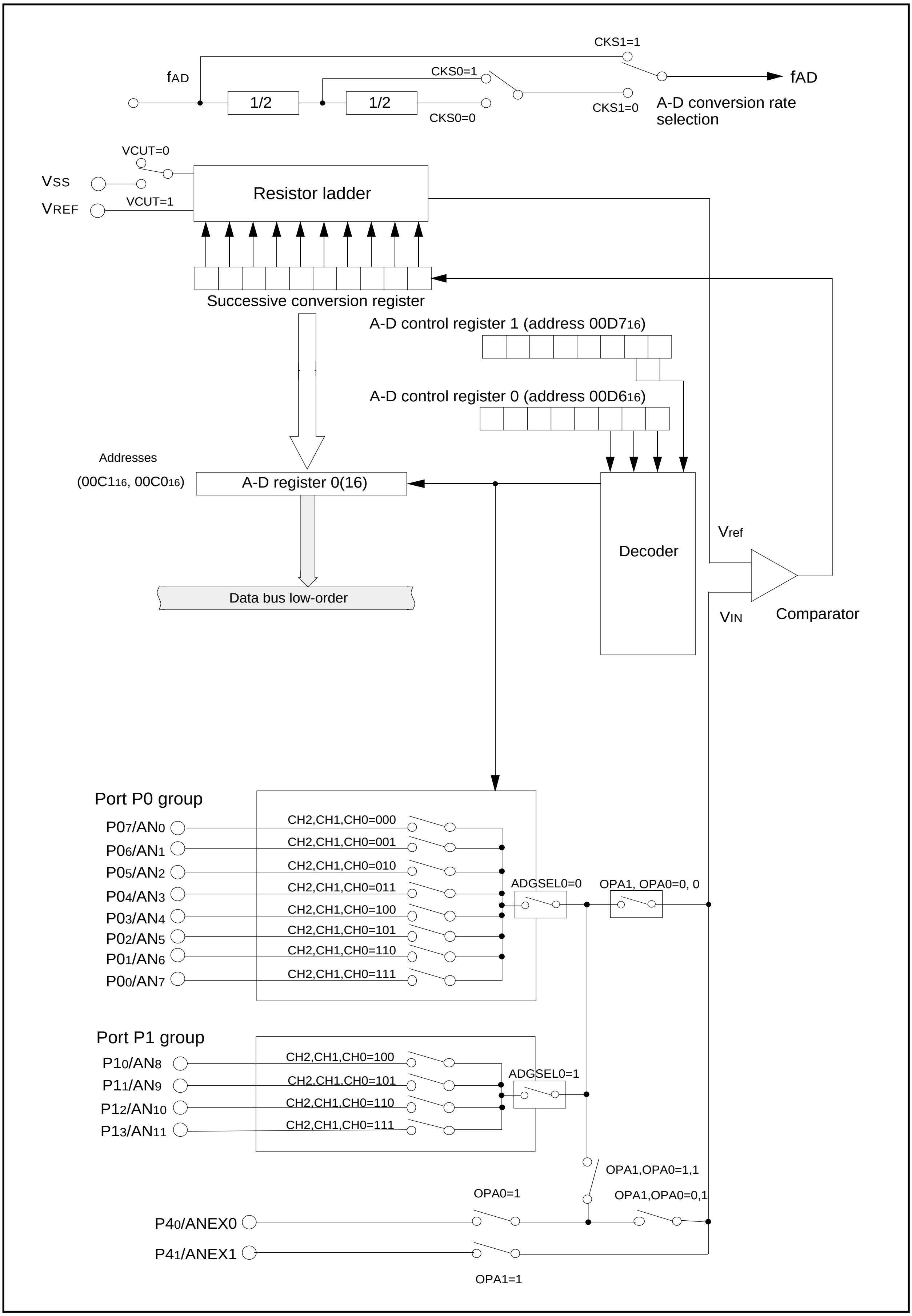


Figure 1.16.1. Block diagram of A-D converter

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Symbol	Address	When reset
ADCON0	00D6$_{16}$	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$_4$, AN$_8$ is selected 1 0 1 : AN$_5$, AN$_9$ is selected 1 1 0 : AN$_6$, AN$_{10}$ is selected 1 1 1 : AN$_7$, AN$_{11}$ is selected (Note 2, 3)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	0 : One-shot mode 1 : Repeat mode (Note 2)	○	○
ADGSEL0	A-D input group select bit	0 : Port P0 group is selected 1 : Port P1 group is selected	○	○
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$_4$ to AN$_7$ and AN$_8$ to AN$_{11}$ are selected by the A-D input group select bit.

A-D control register 1 (Note 1)

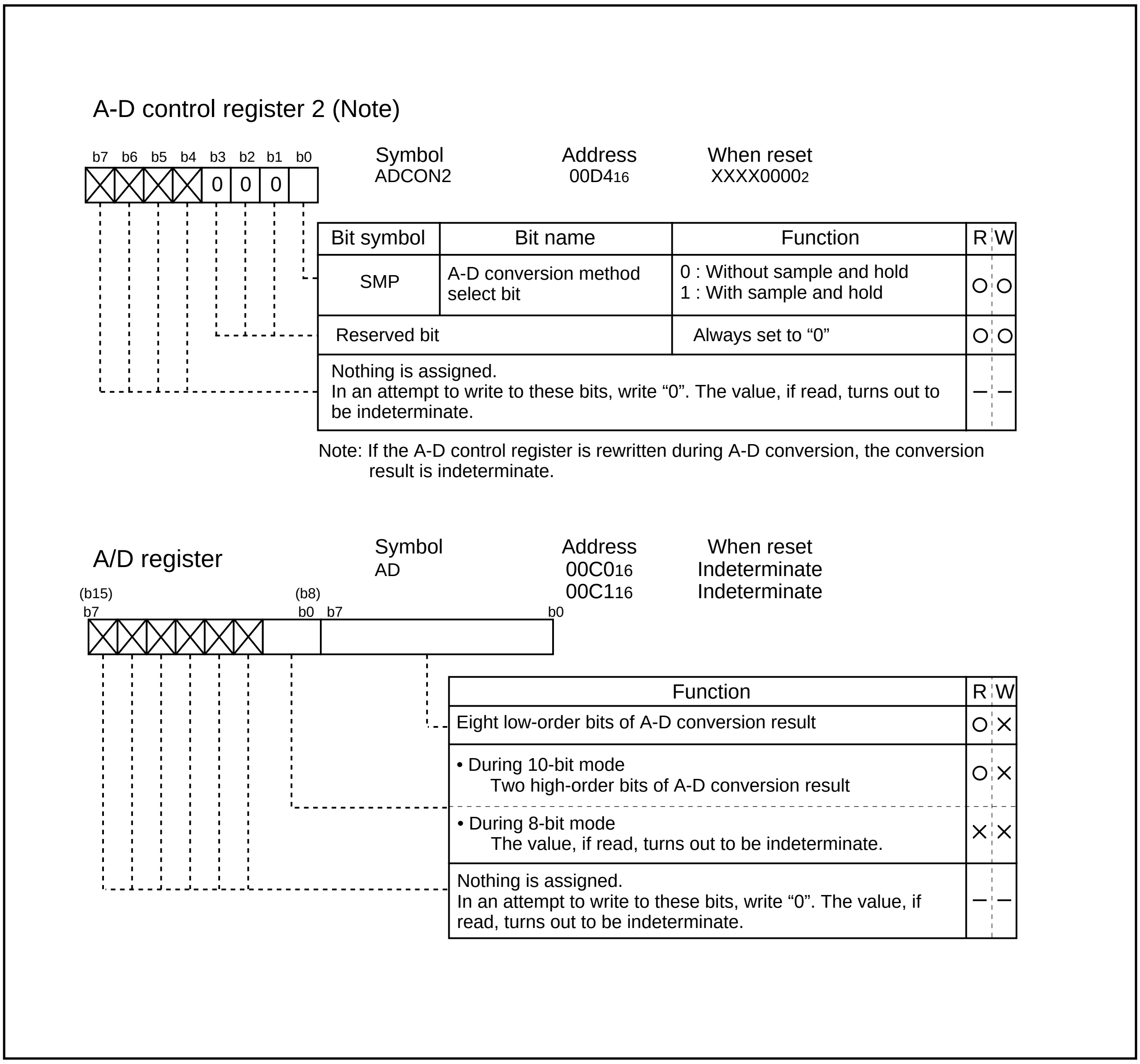
b7	b6	b5	b4	b3	b2	b1	b0
					0	0	0

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

Bit symbol	Bit name	Function	R	W
Set this bit to “0”. When read, their values are “0”.			○	○
BITS	8/10-bit mode select bit (Note 2)	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1 (Note 3)	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	○	○
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : External op-amp connection mode	○	○
OPA1			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: In repeat mode, only 8-bit mode can be used.
Note 3: When f(X$_{IN}$) is over 10 MHz, the ϕ_{AD} frequency must be under 10 MHz by dividing.

Figure 1.16.2. A-D converter-related registers (1)



A-D control register 2 (Note)

Symbol	Address	When reset
ADCON2	00D4$_{16}$	XXXX0000$_2$

Bit symbol	Bit name	Function	R	W
SMP	A-D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	○	○
Reserved bit		Always set to “0”	○	○
Nothing is assigned. 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

Symbol	Address	When reset
AD	00C0$_{16}$	Indeterminate
	00C1$_{16}$	Indeterminate

Function	R	W
Eight low-order bits of A-D conversion result	○	×
• During 10-bit mode Two high-order bits of A-D conversion result	○	×
• During 8-bit mode 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.	–	–

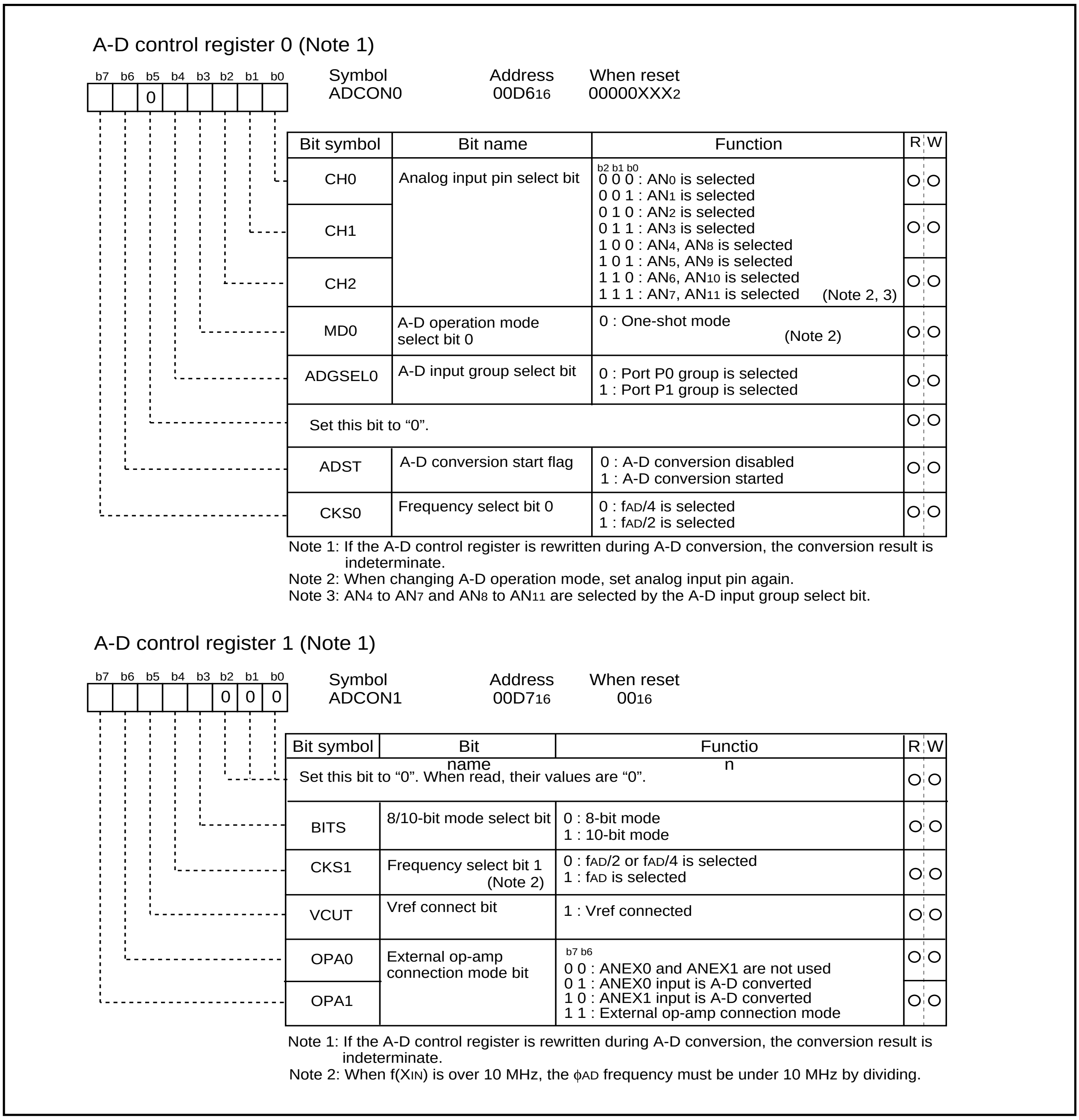
Figure 1.16.3. A-D converter-related registers (2)

(1) One-shot mode

In one-shot mode, the pin selected using the analog input pin select bit is used for one-shot A-D conversion. (See Table 1.16.2.) Figure 1.16.4 shows the A-D control register in one-shot mode.

Table 1.16.2. One-shot mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for one A-D conversion
Start condition	Writing “1” to A-D conversion start flag
Stop condition	• 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_{11}, as selected
Reading of result of A-D converter	Read A-D register



A-D control register 0 (Note 1)

Symbol	Address	When reset
ADCON0	$00D6_{16}$	$00000XXX_2$

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, AN_8 is selected 1 0 1 : AN_5, AN_9 is selected 1 1 0 : AN_6, AN_{10} is selected 1 1 1 : AN_7, AN_{11} is selected (Note 2, 3)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	0 : One-shot mode (Note 2)	○	○
ADGSEL0	A-D input group select bit	0 : Port P0 group is selected 1 : Port P1 group is selected	○	○
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_4 to AN_7 and AN_8 to AN_{11} are selected by the A-D input group select bit.

A-D control register 1 (Note 1)

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

Bit symbol	Bit name	Functio n	R	W
Set this bit to “0”. When read, their values are “0”.			○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1 (Note 2)	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : External op-amp connection mode	○	○
OPA1			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When $f(X_{IN})$ is over 10 MHz, the ϕ_{AD} frequency must be under 10 MHz by dividing.

Figure 1.16.4. A-D conversion register in one-shot mode

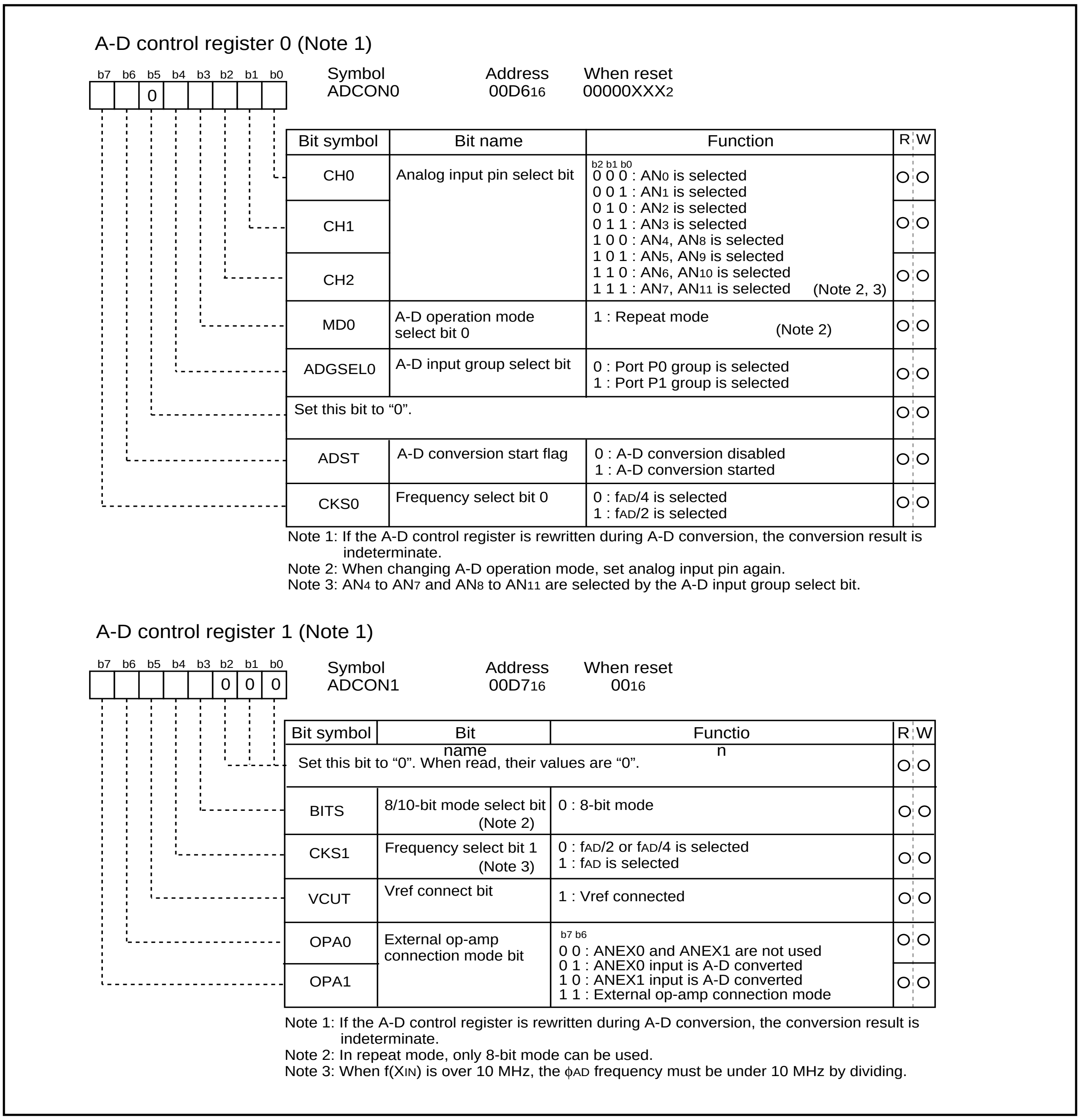
(2) Repeat mode

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A-D conversion. (See Table 1.16.3.) Figure 1.16.5 shows the A-D control register in repeat mode.

Table 1.16.3. 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$_{11}$, as selected (Note)
Reading of result of A-D converter	Read A-D register (at any time)

Note : AN$_4$ to AN$_7$ can be used in the same way as for AN$_8$ to AN$_{11}$.



A-D control register 0 (Note 1)

Symbol ADCON0 Address 00D6$_{16}$ 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$_4$, AN$_8$ is selected 1 0 1 : AN$_5$, AN$_9$ is selected 1 1 0 : AN$_6$, AN$_{10}$ is selected 1 1 1 : AN$_7$, AN$_{11}$ is selected (Note 2, 3)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	1 : Repeat mode (Note 2)	○	○
ADGSEL0	A-D input group select bit	0 : Port P0 group is selected 1 : Port P1 group is selected	○	○
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$_4$ to AN$_7$ and AN$_8$ to AN$_{11}$ are selected by the A-D input group select bit.

A-D control register 1 (Note 1)

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

Bit symbol	Bit name	Functio n	R	W
Set this bit to “0”. When read, their values are “0”.			○	○
BITS	8/10-bit mode select bit (Note 2)	0 : 8-bit mode	○	○
CKS1	Frequency select bit 1 (Note 3)	0 : f$_{AD}$/2 or f$_{AD}$/4 is selected 1 : f$_{AD}$ is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : External op-amp connection mode	○	○
OPA1			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: In repeat mode, only 8-bit mode can be used.
Note 3: When f(X$_{IN}$) is over 10 MHz, the φ$_{AD}$ frequency must be under 10 MHz by dividing.

Figure 1.16.5. A-D conversion register in repeat mode

- **Sample and hold**

Sample and hold is selected by setting bit 0 of the A-D control register 2 (address 00D4$_{16}$) to “1”. When sample and hold is selected, the rate of conversion of each pin increases. As a result, a 28 f$_{AD}$ cycle is achieved with 8-bit resolution and 33 f$_{AD}$ 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.

- **Extended analog input pins**

In one-shot mode and repeat mode, the input via the extended analog input pins ANEX$_0$ and ANEX$_1$ can also be converted from analog to digital.
When bit 6 of the A-D control register 1 (address 00D7$_{16}$) is “1” and bit 7 is “0”, input via ANEX$_0$ is converted from analog to digital.
When bit 6 of the A-D control register 1 (address 00D7$_{16}$) is “0” and bit 7 is “1”, input via ANEX$_1$ is converted from analog to digital.

- **External operation amp connection mode**

In this mode, multiple external analog inputs via the extended analog input pins, ANEX$_0$ and ANEX$_1$, can be amplified together by just one operation amp and used as the input for A-D conversion.
When bit 6 of the A-D control register 1 (address 00D7$_{16}$) is “1” and bit 7 is “1”, input via AN$_0$ to AN$_{11}$ is output from ANEX$_0$. The input from ANEX$_1$ is converted from analog to digital and the result stored in the A-D register. The speed of A-D conversion depends on the response of the external operation amp. Do not connect the ANEX$_0$ and ANEX$_1$ pins directly. Figure 1.16.6 is an example of how to connect the pins in external operation amp mode.

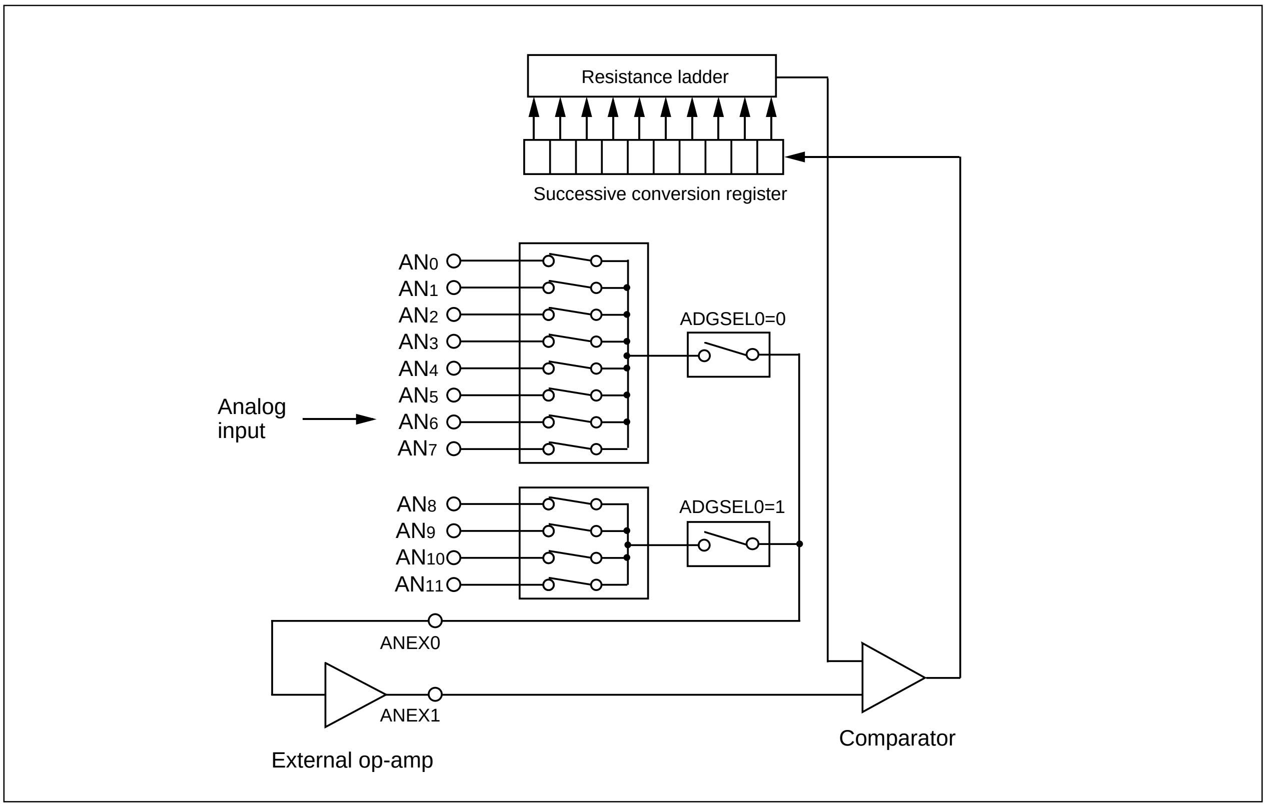


Figure 1.16.6. Example of external op-amp connection mode

D-A Converter

This is an 8-bit, R-2R type D-A converter. The microcomputer contains one independent D-A converter of this type. D-A conversion is performed when a value is written to the corresponding D-A register. Bit 0 (D-A output enable bit) of the D-A control register decide if the result of conversion is to be output. Do not set the target port to output mode if D-A conversion is to be performed. When D-A output is set for enabled, the corresponding port is inhibited to be pulled up.

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

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

V_{REF}: reference voltage

Table 1.17.1 lists the performance of the D-A converter. Figure 1.17.1 shows the block diagram of the D-A converter, Figure 1.17.2 shows the D-A control register and Figure 1.17.3 shows D-A converter equivalent circuit.

Table 1.17.1. Performance of D-A converter

Item	Performance
Conversion method	R-2R method
Resolution	8 bits
Analog output pin	1 channel

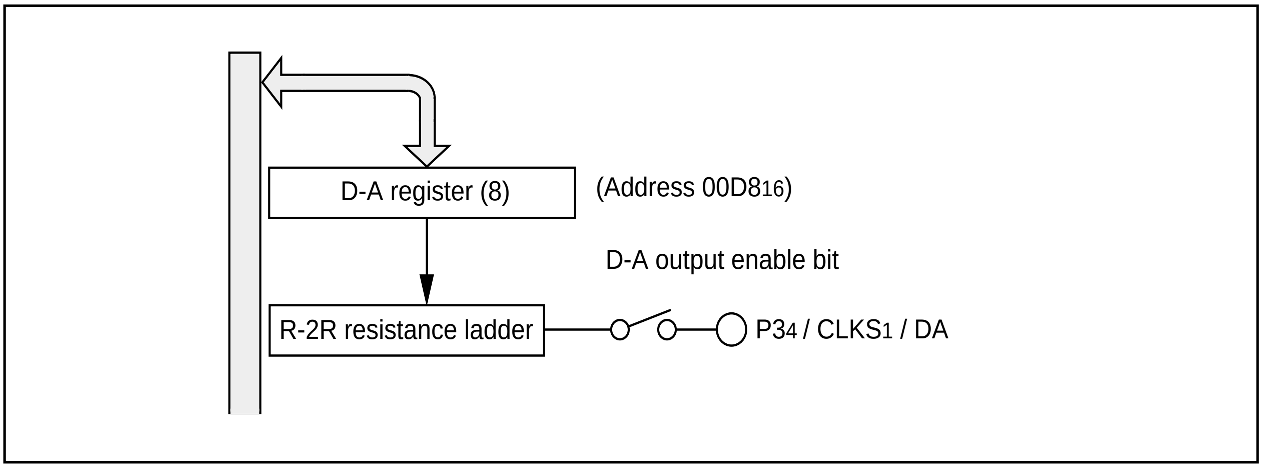
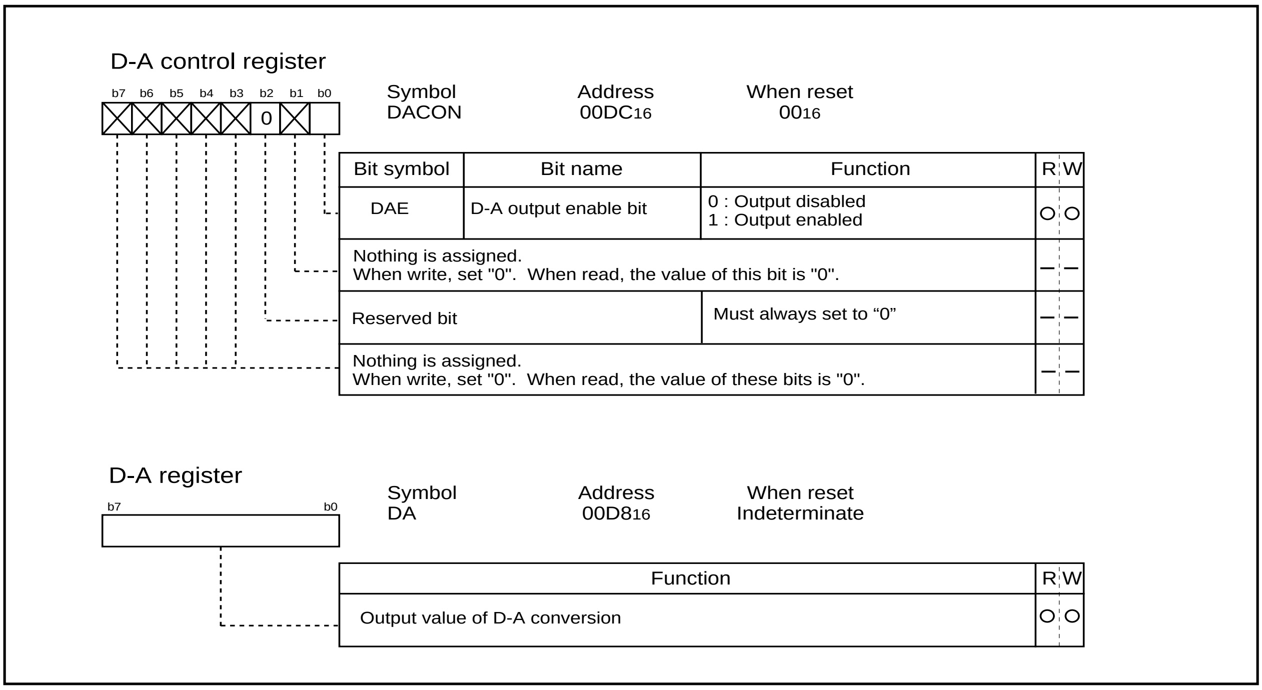


Figure 1.17.1. Block diagram of D-A converter



D-A control register

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

Symbol	Address	When reset
DACON	00DC16	0016

Bit symbol	Bit name	Function	R	W
DAE	D-A output enable bit	0 : Output disabled 1 : Output enabled	○	○
Nothing is assigned. When write, set "0". When read, the value of this bit is "0".			–	–
Reserved bit		Must always set to “0”	–	–
Nothing is assigned. When write, set "0". When read, the value of these bits is "0".			–	–

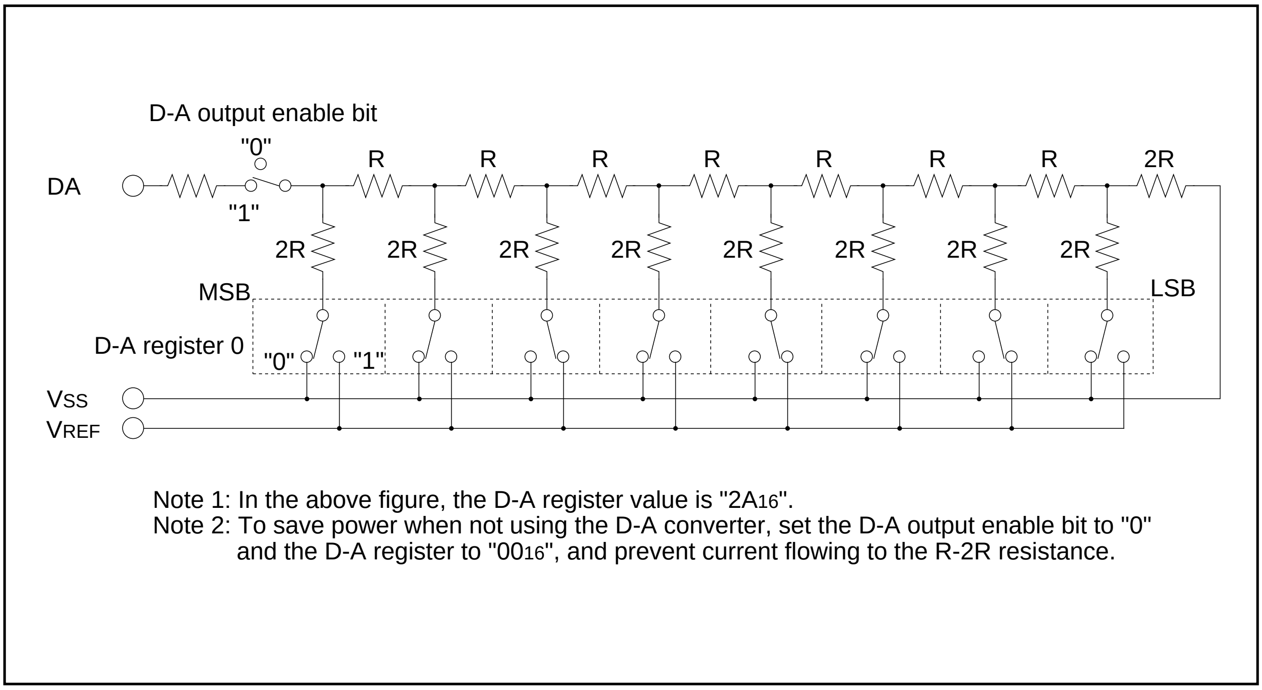
D-A register

b7	b0

Symbol	Address	When reset
DA	00D816	Indeterminate

Function	R	W
Output value of D-A conversion	○	○

Figure 1.17.2. D-A control register



Note 1: In the above figure, the D-A register value is "2A16".
Note 2: To save power when not using the D-A converter, set the D-A output enable bit to "0" and the D-A register to "0016", and prevent current flowing to the R-2R resistance.

Figure 1.17.3. D-A converter equivalent circuit

Programmable I/O Ports

There are 34 programmable I/O ports: P0 to P4 (when M30102). 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. The port P1 can be used as LED drive port if the drive capacity is set to “HIGH”.

Figures 1.18.1 to 1.18.3 show the programmable I/O ports.

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

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

(1) Direction registers

Figure 1.18.4 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 1.18.5 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 1.18.6 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 1.18.6 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.

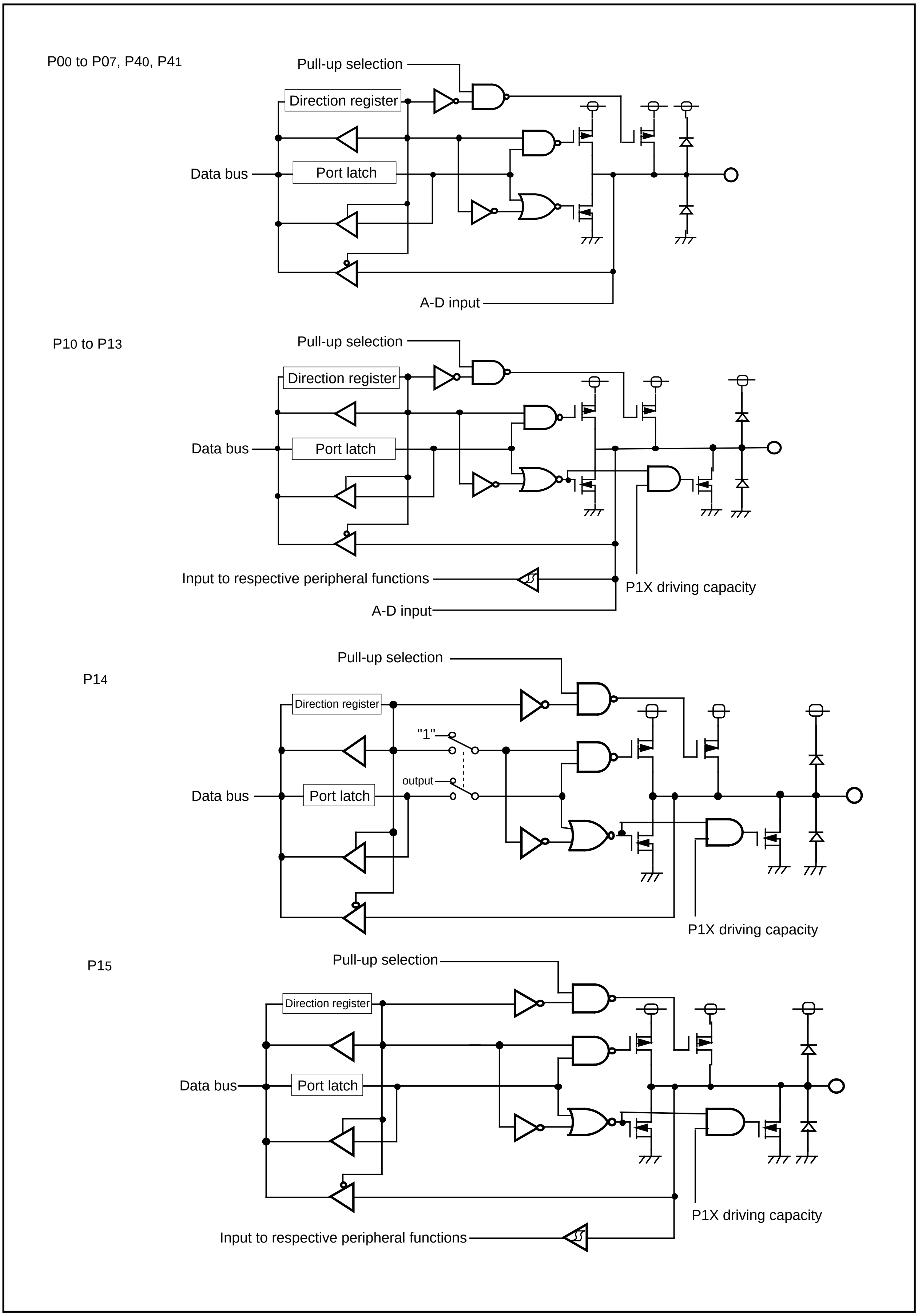


Figure 1.18.1. Programmable I/O ports (1)

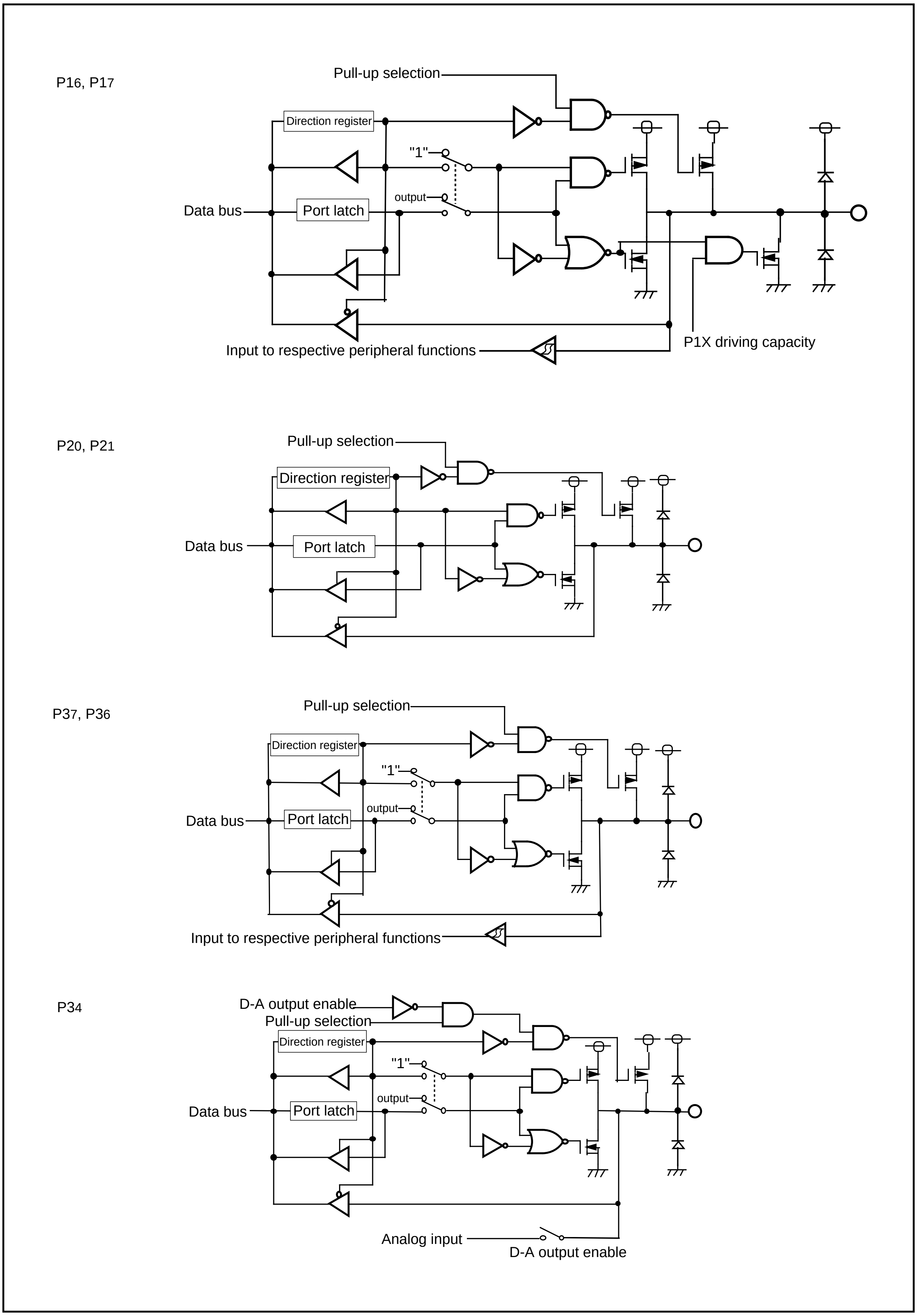
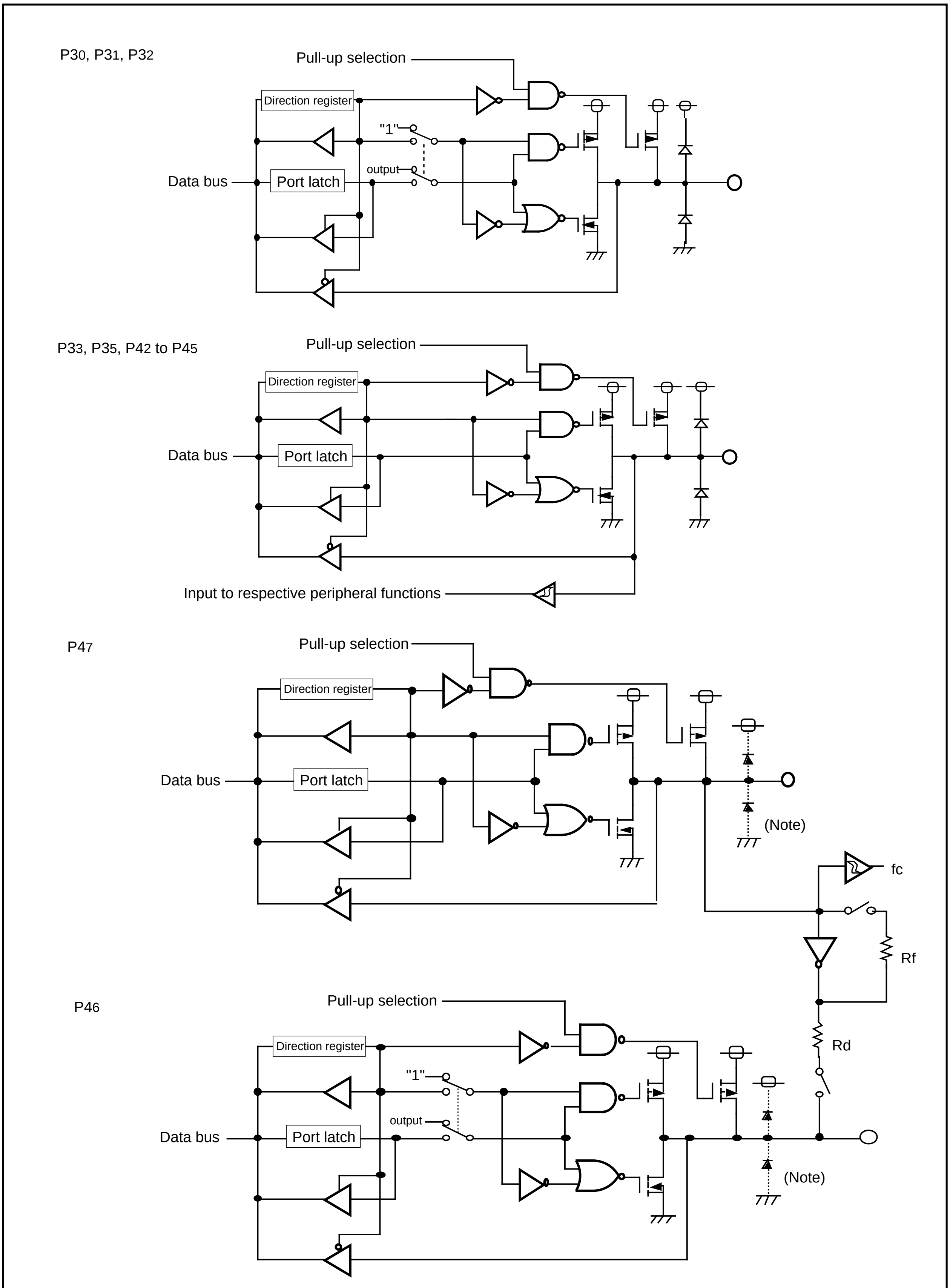
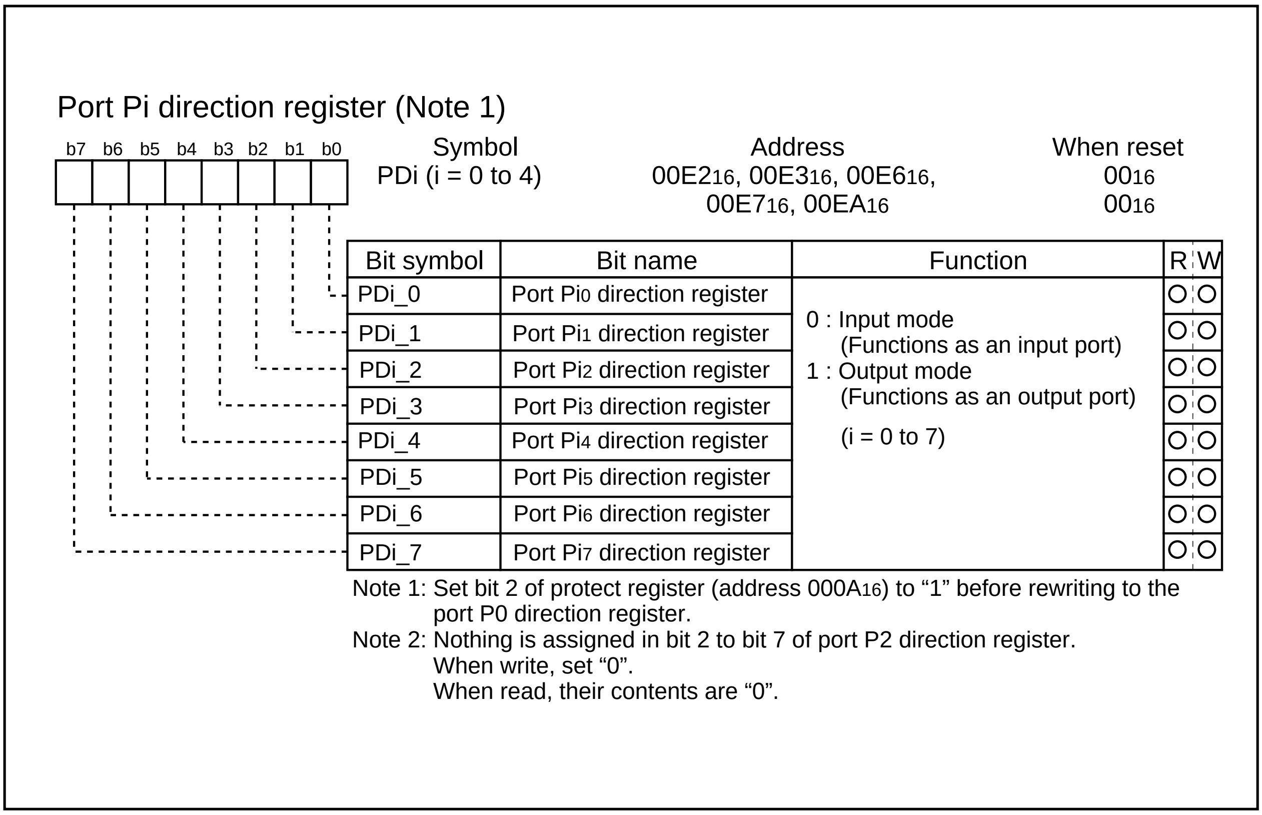


Figure 1.18.2. Programmable I/O ports (2)



Note : ········◄········symbolizes a parasitic diode.
Do not apply a voltage higher than Vcc to each port.

Figure 1.18.3. Programmable I/O ports (3)



Port Pi direction register (Note 1)

Symbol	Address	When reset
PDi (i = 0 to 4)	$00E2_{16}$, $00E3_{16}$, $00E6_{16}$,	00_{16}
	$00E7_{16}$, $00EA_{16}$	00_{16}

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

Note 1: Set bit 2 of protect register (address $000A_{16}$) to “1” before rewriting to the port P0 direction register.

Note 2: Nothing is assigned in bit 2 to bit 7 of port P2 direction register.
When write, set “0”.
When read, their contents are “0”.

Figure 1.18.4. Direction register

Port Pi register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
Pi (i = 0 to 4)	00E0$_{16}$, 00E1$_{16}$, 00E4$_{16}$, 00E5$_{16}$, 00E8$_{16}$	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)	○	○
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 bit 2 to bit 7 of port P2 register.
When write, set "0".
When read, their contents are "0".

Figure 1.18.5. Port register

Pull-up control register 0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
PUR0	00FC$_{16}$	00X00000$_2$

Bit symbol	Bit name	Function	R	W
PU00	$P0_0$ to $P0_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU01	$P0_4$ to $P0_7$ pull-up		○	○
PU02	$P1_0$ to $P1_3$ pull-up		○	○
PU03	$P1_4$ to $P1_7$ pull-up		○	○
PU04	$P2_0$, $P2_1$ pull-up		○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
PU06	$P3_0$ to $P3_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU07	$P3_4$ to $P3_7$ pull-up		○	○

Pull-up control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
PUR1	00FD$_{16}$	XXXXXX00$_2$

Bit symbol	Bit name	Function	R	W
PU10	$P4_0$ to $P4_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
			○	○
PU11	$P4_4$ to $P4_7$ pull-up		○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—

Port P1 drive capacity control register

b7 b6 b5 b4 b3 b2 b1 b0

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

Bit symbol	Bit name	Function	R	W
DRR0	Port $P1_0$ drive capacuty	Set P1 N-channel output transistor drive capacity 0 : LOW 1 : HIGH	○	○
DRR1	Port $P1_1$ drive capacuty		○	○
DRR2	Port $P1_2$ drive capacuty		○	○
DRR3	Port $P1_3$ drive capacuty		○	○
DRR4	Port $P1_4$ drive capacuty		○	○
DRR5	Port $P1_5$ drive capacuty		○	○
DRR6	Port $P1_6$ drive capacuty		○	○
DRR7	Port $P1_7$ drive capacuty		○	○

Figure 1.18.6. Pull-up control register

Example connection of unused pins

Table 1.18.1. Example connection of unused pins

Pin name	Connection
Ports P0 to P4	After setting for input mode, connect every pin to V_{SS} (pull-down); or after setting for output mode, leave these pins open.
X_{OUT} (Note)	Open
V_{REF}	Connect to V_{SS}

Note: With external clock input to X_{IN} pin.

Precautionary Notes in Using the Device

Serial I/O

(1) When reading data from the UARTi receive buffer in the clock asynchronous serial I/O mode, data should be read high-byte first then low-byte using a byte-size instruction. If data is read as low-byte then high-byte or with a word-size instruction, the framing error and parity error flags are cleared.

A code example is shown below.

```
MOV.B     00A3H. R0H;     Read the high-byte of UART0 receive buffer register
MOV.B     00A2H. R0L;     Read the low-byte of UART0 receive buffer register
```

A-D Converter

(1) Only write to each bit (except bit 6) of the AD Control Register 0, or each bit of the AD Control Register 1, or bit 0 of the AD Control Register 2 when AD conversion is stopped (before a trigger occurs).
(2) When the Vref Connection Bit is changed from “0” to “1”, wait 1 µs or longer before starting AD conversion.
(3) When changing AD operation mode, select an analog pin again.
(4) One Shot Mode
Read the AD register only after confirming AD conversion is completed, which can be determined by using the AD conversion interrupt.
(5) Repeat Mode
Use the undivided main clock as the internal CPU clock when using this mode. The main clock can be divided by an internal divider circuit but make sure that you use main clock when using this mode.

Stop and Wait Mode

(1) You must put at least four NOPs after a stop (All-Clock Stop Bit to "1") or a wait instruction. When switching to a stop or wait mode, 4 instructions are prefetched after the stop or wait instruction. And so, ensure that at least four NOPs follow the stop or wait instruction.

Interrupts

(1) Reading Address 0 by Firmware

- Please do not read address 0 by firmware. In the CPU's interrupt processing sequence, when a maskable interrupt occurs, the interrupt information (interrupt no. and interrupt request level) are read from address 0. This read in turn, clears the interrupt request bit to "0" even pending with higher request level. Reading address 0 by firmware may cause interrupt cancellation or unexpected interrupts so please do not read address 0 by firmware.

(2) Stack Pointer

- Set the value of the stack pointer before accepting interrupts. Immediately after a reset, the value of the stack pointer is 0000_{16}. Accepting an interrupt before setting a value of the stack pointer may produce unpredictable results (runaway program, etc.) Make sure that you set the value of the stack pointer before accepting interrupts.

(3) External interrupts

- Clear the interrupt request bit to "0" when the INT0 - INT3 polarity is changed. The reason being is that an interrupt request may be generated when the polarity is changed.

(4) Rewriting the Interrupt Control Register

- When rewriting the Interrupt Control Register, do it at a point where it does not generate an interrupt request for that register. If there is a possibility that an interrupt may occur, disable the interrupt before rewriting. Examples are shown below.

Example 1:

```
INT_SWITCH1:
    FCLR     I              ; Disable interrupts.
    AND.B    #00h, 0055h    ; Clear T1IC int. priority level and int. request bit.
    NOP                     ;
    NOP
    FSET     I              ; Enable interrupts.
```

Example 2:

```
INT_SWITCH2:
    FCLR     I              ; Disable interrupts.
    AND.B    #00h, 0055h    ; Clear T1IC 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 T1IC int. priority level and int. request bit.
    POPC     FLG            ; Enable interrupts.
```

Note: The reason why two NOP instructions or dummy read were inserted before the FSET I for ex. 1 & 2 is to prevent interrupt enable flag from being set, due to the effects of instruction queue, before the rewritten value of the interrupt control register takes effect.

- When an instruction to rewrite the interrupt control register is executed while the interrupt is disabled, depending on the instruction used for rewriting, there are times the interrupt request bit is not set even if an interrupt request for that register has been generated. If this creates a problem, please use any of the instructions below to rewrite the register.
  Instructions : AND, OR, BCLR, BSET

Noise

(1) Bypass Capacitor between VCC and VSS Pins

- Insert a bypass capacitor (at least 0.1 µF) between VCC and VSS pins as noise and latch-up countermeasures. In addition, make sure that connecting lines are the shortest and widest possible.

(2) Port Control Registers Data Read Error

- During severe noise testing, mainly power supply system noise, and introduction of external noise, the data of port related registers may changed. As a firmware countermeasure, it is recommended to periodically re-set the port registers, port direction registers and pull-up control registers. However, you should fully examine before introducing the re-set routine as conflicts may be created between this re-set routine and interrupt routines (i. e. ports are switched during interrupts).

Electrical characteristics

Table 1.19.1 Absolute maximum ratings

Symbol	Parameter		Condition	Rated value	Unit
Vcc	Supply voltage			- 0.3 to 6.5	V
V_I	Input voltage	$\overline{\text{RESET}}$, V_{REF}, X_{IN} $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_1$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, CNVss		- 0.3 to Vcc + 0.3	V
V_O	Output voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_1$,P30 to $P3_7$, $P4_0$ to $P4_7$,X_{OUT}		- 0.3 to Vcc + 0.3	V
		IV_{CC}		- 0.3 to 3.6V	V
P_d	Power dissipation		Ta = 25 °C	300	mW
T_{opr}	Operating ambient temperature			- 20 to 85 (Note 1)	°C
T_{stg}	Storage temperature			- 40 to 150 (Note 2)	°C

Note 1: Extended operating temperature version: -40 to 85 °C. When flash memory version is program/erase mode: 25±5 °C.
Specify a product of -40 to 85°C to use it.

Note 2: Extended operating temperature version: -65 to 85 °C.

Note 3: For M30100 (32-pin version), $P2_0$, $P2_1$, $P3_4$ to $P3_6$, and $P4_0$ to $P4_7$ are not accessed to external pins.

Note 4: For M30101 (42-pin version), $P2_0$ and $P2_1$ are not accessed to external pins.

Table 1.19.2 Recommended operating conditions (Note 1)

Symbol	Parameter			Standard			Unit
				Min	Typ.	Max.	
Vcc	Supply voltage			2.7	5.0	5.5	V
Vss	Supply voltage				0		V
V_{IH}	HIGH input voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_1$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, X_{IN}, RESET, CNV_{SS},		0.8Vcc		Vcc	V
V_{IL}	LOW input voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_1$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, X_{IN}, RESET, CNV_{SS},		0		0.2Vcc	V
$I_{OH (peak)}$	HIGH peak output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_1$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$,				- 10.0	mA
$I_{OH (avg)}$	HIGH average output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_1$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$,				- 5.0	mA
$I_{OL (peak)}$	LOW peak output current	$P0_0$ to $P0_7$, $P2_0$ to $P2_1$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$,				10.0	mA
$I_{OL (peak)}$		$P1_0$ to $P1_7$	HIGHPOWER			30.0	mA
			LOWPOWER			10.0	
$I_{OL (avg)}$	LOW average output current	$P0_0$ to $P0_7$, $P2_0$ to $P2_1$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$,				5.0	mA
$I_{OL (avg)}$		$P1_0$ to $P1_7$	HIGHPOWER			15.0	mA
			LOWPOWER			5.0	
f (X_{IN})	Main clock input oscillation frequency (Note 4)		V_{CC}=4.2V to 5.5V	0		16	MHz
			V_{CC}=2.7V to 4.2V	0		7.33 x V_{CC} - 14.791	MHz
f (X_{CIN})	Subclock oscillation frequency				32.768	50	kHz

Note 1: Unless otherwise noted: V_{CC} = 2.7V to 5.5V, Ta = – 20 to 85°C

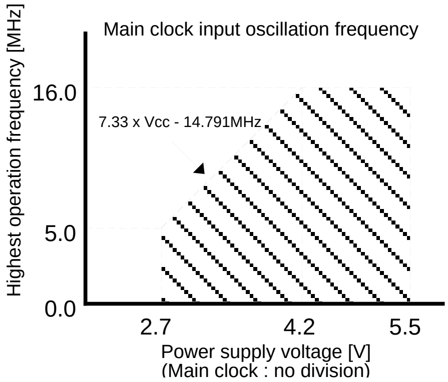
Note 2: The average output current is an average value measured over 100ms.

Note 3: Keep output current as follows:

The sum of port $P0_0$ to $P0_3$, $P1_3$ to $P1_7$, $P2_0$, $P3_4$ to $P3_7$, $P4_6$ to $P4_7$ I_{OL} (peak) is under 60 mA. The sum of port $P0_0$ to $P0_3$, $P1_3$ to $P1_7$, $P2_0$, $P3_4$ to $P3_7$, $P4_6$ to $P4_7$ I_{OH} (peak) is under 60 mA. The sum of port $P0_4$ to $P0_7$, $P1_0$ to $P1_2$, $P2_1$, $P3_0$ to $P3_3$, $P4_0$ to $P4_5$ I_{OL} (peak) is under 60 mA. The sum of port $P0_4$ to $P0_7$, $P1_0$ to $P1_2$, $P2_1$, $P3_0$ to $P3_3$, $P4_0$ to $P4_5$ I_{OH} (peak) is under 60 mA.

Note 4: Relationship between main clock oscillation frequency and supply voltage is shown as below.

Note 5: For M30100 (32-pin version), $P2_0$, $P2_1$, $P3_4$ to $P3_6$, and $P4_0$ to $P4_7$ are not accessed to external pins. For M30101 (42-pin version), $P2_0$ and $P2_1$ are not accessed to external pins.



$V_{CC} = 5V$

Table 1.19.3 Electrical characteristics (Note)

Symbol	Parameter			Measuring condition			Standard			Unit
							Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_1$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$		I_{OH} = - 5 mA			3.0			V
				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				3.3		V
			LOWPOWER	No load				3.3		
V_{OL}	LOW output voltage	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_1$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$		I_{OL} = 5 mA					2.0	V
				I_{OL} = 200 µA					0.45	V
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_{COUT}	HIGHPOWER	No load				0		V
			LOWPOWER	No load				0		
V_{T+} -V_{T-}	Hysteresis	$CNTR_0$,TC_{IN}, $\overline{INT_0}$ to $\overline{INT_3}$,CLK_0,CLK_1 RxD_0, RxD_1,KI_0 to KI_3					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 $P1_7$,$P2_0$ to $P2_1$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$, X_{IN} RESET, CNVss		V_I = 5V					5.0	µA
I_{IL}	LOW input current	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_1$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$, X_{IN} RESET, CNVss		V_I = 0V					-5.0	µA
R_{PULLUP}	Pull-up resistor	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_1$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$		V_I = 0V			30.0	50.0	167.0	kΩ
R_{fXIN}	Feedback resistor	X_{IN}						1.0		MΩ
R_{fXCIN}	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	Mask ROM, Flash memory	f(X_{IN})=16MHz Square wave, no division		28.0	38.0	mA
					Mask ROM	f(X_{CIN})=32kHz Square wave		90.0		µA
					Flash memory	f(X_{CIN})=32kHz Square wave		TBD		µA
					Mask ROM, Flash memory	f(X_{CIN})=32kHz When a WAIT instruction is executed		TBD		µA
						Ta=25°C when clock is stopped			TBD	µA
						Ta=85°C when clock is stopped			TBD	

Note: Unless otherwise noted: V_{CC} = 5V, V_{SS} = 0V at Ta = 25°C, f(X_{IN}) = 16MHz

V_{CC} = 5V

Table 1.19.4 A-D conversion characteristics (Note 1)

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	AN_0 to AN_{11} input			±3	LSB
				$ANEX_0$, $ANEX_1$ input, external op-amp connected mode			±7	LSB
		Sample & hold function available(8bit)	V_{REF} = V_{CC} = 5V				±2	LSB
R_{LADDER}	Ladder resistance		V_{REF} = V_{CC}		10		40	kΩ
t_{CONV}	Conversion time(10bit)		$f(X_{IN})$=10MHz, $ø_{AD}$=f_{AD}=10MHz		3.3			µs
t_{CONV}	Conversion time(8bit)		$f(X_{IN})$=10MHz, $ø_{AD}$=f_{AD}=10MHz		2.8			µs
t_{SAMP}	Sampling time		$f(X_{IN})$=10MHz, $ø_{AD}$=f_{AD}=10MHz		0.3			µs
V_{REF}	Reference voltage		$f(X_{IN})$=10MHz, $ø_{AD}$=f_{AD}=10MHz		2		V_{CC}	V
V_{IA}	Analog input voltage		$f(X_{IN})$=10MHz, $ø_{AD}$=f_{AD}=10MHz		0		V_{REF}	V

Note 1: Unless otherwise noted: V_{CC} = V_{REF} =5V, V_{SS} = 0V at Ta = 25°C, $f(X_{IN})$ = 16MHz

Note 2: Divide the f_{AD} if $f(X_{IN})$ exceeds 10MHz, and make AD operation clock frequency ($ø_{AD}$) equal to or lower than 10MHz.

Table 1.19.5 D-A conversion characteristics (Note 1)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
–	Resolution				8	Bits
–	Absolute accuracy				1.0	%
t_{su}	Setup time				3	µs
R_O	Output resistance		4	10	20	kΩ
I_{VREF}	Reference power supply input current	(Note 2)			1.5	mA

Note 1: Unless otherwise noted: V_{CC} = V_{REF} =5V, V_{SS} = 0V at Ta = 25°C, $f(X_{IN})$ = 16MHz

Note 2: The A-D converter's ladder resistance is not included.

When D-A register contents are not "00_{16}", the current I_{VREF} always flows even though V_{REF} may have been set to be unconnected by the A-D control register.

VCC = 5V

Timing requirements (Unless otherwise noted: VCC = 5V, VSS = 0V at Ta = 25^{O}C)

Table 1.19.6 CNTR0 input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CNTR0)}$	CNTR0 input cycle time	100		ns
$t_{wH(CNTR0)}$	CNTR0 input HIGH pulse width	40		ns
$t_{wL(CNTR0)}$	CNTR0 input LOW pulse width	40		ns

Table 1.19.7 TCIN input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TCIN)}$	TCIN input cycle time	400		ns
$t_{wH(TCIN)}$	TCIN input HIGH pulse width	200		ns
$t_{wL(TCIN)}$	TCIN input LOW pulse width	200		ns

Table 1.19.8 XIN input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(XIN)}$	XIN input cycle time	62.5		ns
$t_{wH(XIN)}$	XIN input HIGH pulse width	30		ns
$t_{wL(XIN)}$	XIN input LOW pulse width	30		ns

Table 1.19.9 Serial I/O

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

Table 1.19.10 External interrupt $\overline{\text{INTi}}$ input

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

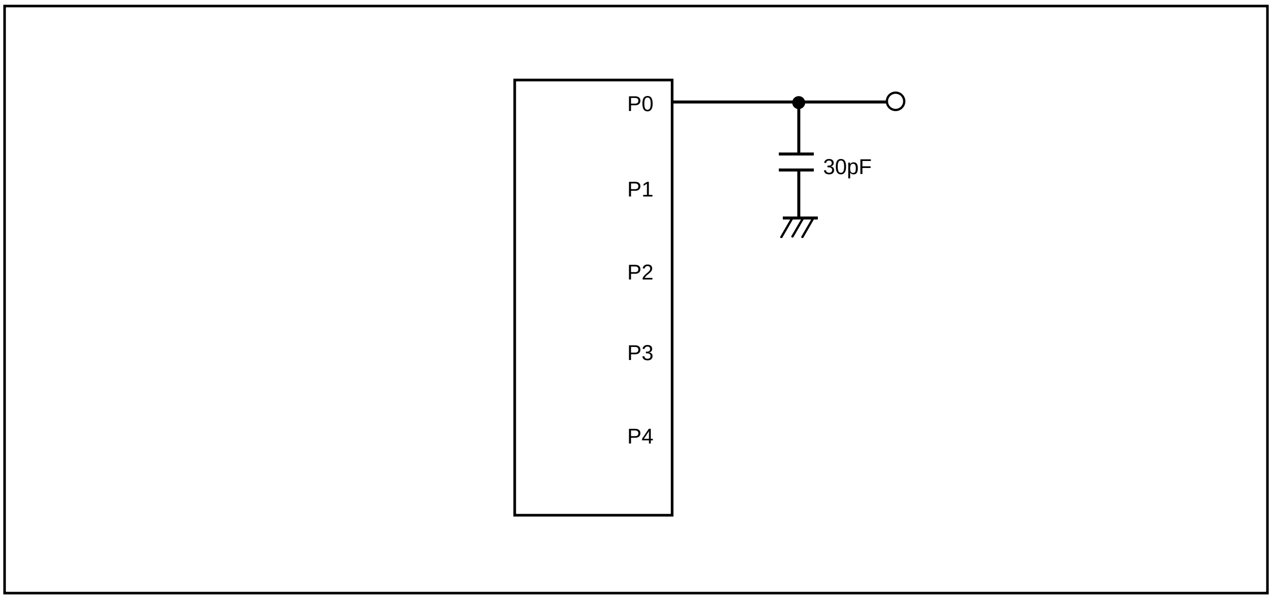


Figure 1.19.1 Port P0 to P4 measurement circuit

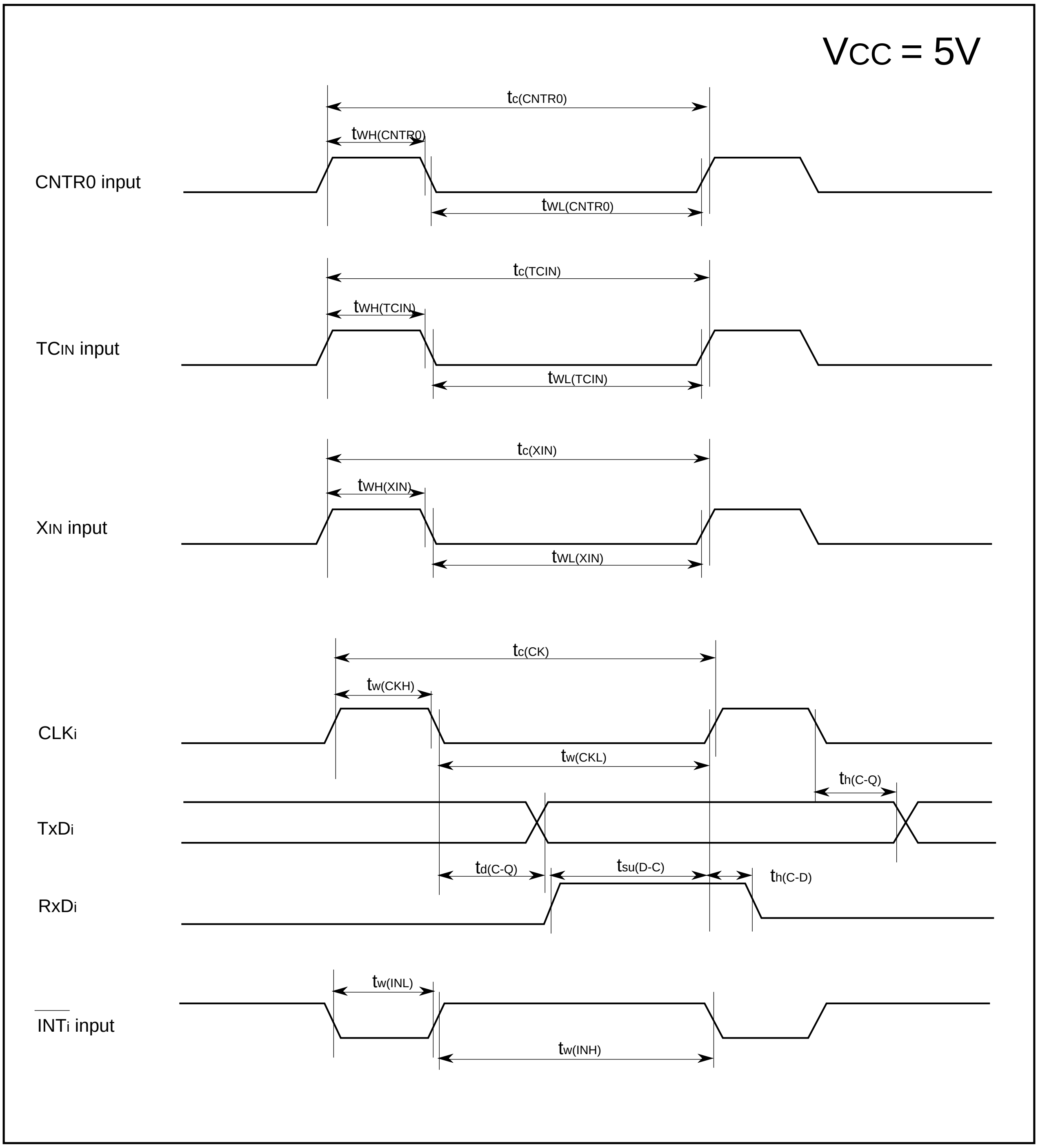


Figure 1.19.2 Vcc=5V timing diagram

V_{CC} = 3V

Table 1.19.11 Electrical characteristics (Note)

Symbol	Parameter			Measuring condition			Standard			Unit
							Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_1$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$		I_{OH} = - 1 mA			2.5			V
V_{OH}	HIGH output voltage	X_{OUT}	HIGHPOWER	I_{OH} = - 0.1 mA			2.5			V
			LOWPOWER	I_{OH} = - 50 µA			2.5			
V_{OH}	HIGH output voltage	X_{COUT}	HIGHPOWER	No load				Vcc		V
			LOWPOWER	No load				Vcc		
V_{OL}	LOW output voltage	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_1$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$		I_{OL} = 1 mA					0.5	V
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_{COUT}	HIGHPOWER	No load				0		V
			LOWPOWER	No load				0		
V_{T+} -V_{T-}	Hysteresis	$CNTR_0$,TC_{IN}, $\overline{INT_0}$ to $\overline{INT_3}$,CLK_0,CLK_1 RxD_0, RxD_1,KI_0 to KI_3					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 $P1_7$,$P2_0$ to $P2_1$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$, X_{IN} RESET, CNVss		V_I = 3V					4.0	µA
I_{IL}	LOW input current	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_1$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$, X_{IN} RESET, CNVss		V_I = 0V					-4.0	µA
R_{PULLUP}	Pull-up resistor	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_1$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$		V_I = 0V			66.0	120.0	500.0	kΩ
R_{fXIN}	Feedback resistor	X_{IN}						3.0		MΩ
R_{fXCIN}	Feedback resistor	X_{CIN}						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	Mask ROM, Flash memory	f(X_{IN})=5MHz Square wave, no division		TBD	TBD	mA
					Mask ROM	f(X_{CIN})=32kHz Square wave		TBD		µA
					Flash memory	f(X_{CIN})=32kHz Square wave		TBD		µA
					Mask ROM, Flash memory	f(X_{CIN})=32kHz When a WAIT instruction is executed		TBD		µA
						Ta=25°C when clock is stopped			1.0	µA
						Ta=85°C when clock is stopped			20.0	

Note: Unless otherwise noted: V_{CC} = 3V, V_{SS} = 0V at Ta = 25°C, f(X_{IN}) = 5MHz

VCC = 3V

Table 1.19.12 A-D conversion characteristics (Note)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
–	Resolution		VREF =VCC			10	Bits
–	Absolute accuracy	Sample & hold function not available (8-bit)	VREF =VCC = 3V, øAD=fAD/2			±2	LSB
RLADDER	Ladder resistance		VREF =VCC	10		40	kΩ
tCONV	Conversion time(8-bit)			14.0			µs
VREF	Reference voltage			2.7		VCC	V
VIA	Analog input voltage			0		VREF	V

Note: Unless otherwise noted: VCC = VREF =3V, VSS = 0V at Ta = 25^{o}C, f(XIN) = 7MHz

Table 1.19.13 D-A conversion characteristics (Note 1)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
−	Resolution				8	Bits
−	Absolute accuracy				1.0	%
tsu	Setup time				3	µs
RO	Output resistance		4	10	20	kΩ
IVREF	Reference power supply input current	(Note 2)			1.5	mA

Note 1: Unless otherwise noted: VCC = AVCC =VREF = 3V, VSS = AVSS =0V at Ta = 25^{o}C, f(XIN) = 7MHz

Note 2: The A-D converter's ladder resistance is not included.
Also, the current IVREF always flows even though VREF may have been set to be unconnected by the A-D control register.

VCC = 3V

Timing requirements (Unless otherwise noted: VCC = 3V, VSS = 0V at Ta = 25^{o}C)

Table 1.19.14 CNTR0 input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CNTR0)}$	CNTR0 input cycle time	300		ns
$t_{wH(CNTR0)}$	CNTR0 input HIGH pulse width	120		ns
$t_{wL(CNTR0)}$	CNTR0 input LOW pulse width	120		ns

Table 1.19.15 TCIN input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TCIN)}$	TCIN input cycle time	1200		ns
$t_{wH(TCIN)}$	TCIN input HIGH pulse width	600		ns
$t_{wL(TCIN)}$	TCIN input LOW pulse width	600		ns

Table 1.19.16 XIN input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(XIN)}$	XIN input cycle time	143		ns
$t_{wH(XIN)}$	XIN input HIGH pulse width	70		ns
$t_{wL(XIN)}$	XIN input LOW pulse width	70		ns

This applies when using one D-A converter, with the D-A register for the unused D-A converter set to "00_{16}".

Table 1.19.17 Serial I/O

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

Table 1.19.18 External interrupt $\overline{\text{INTi}}$ input

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

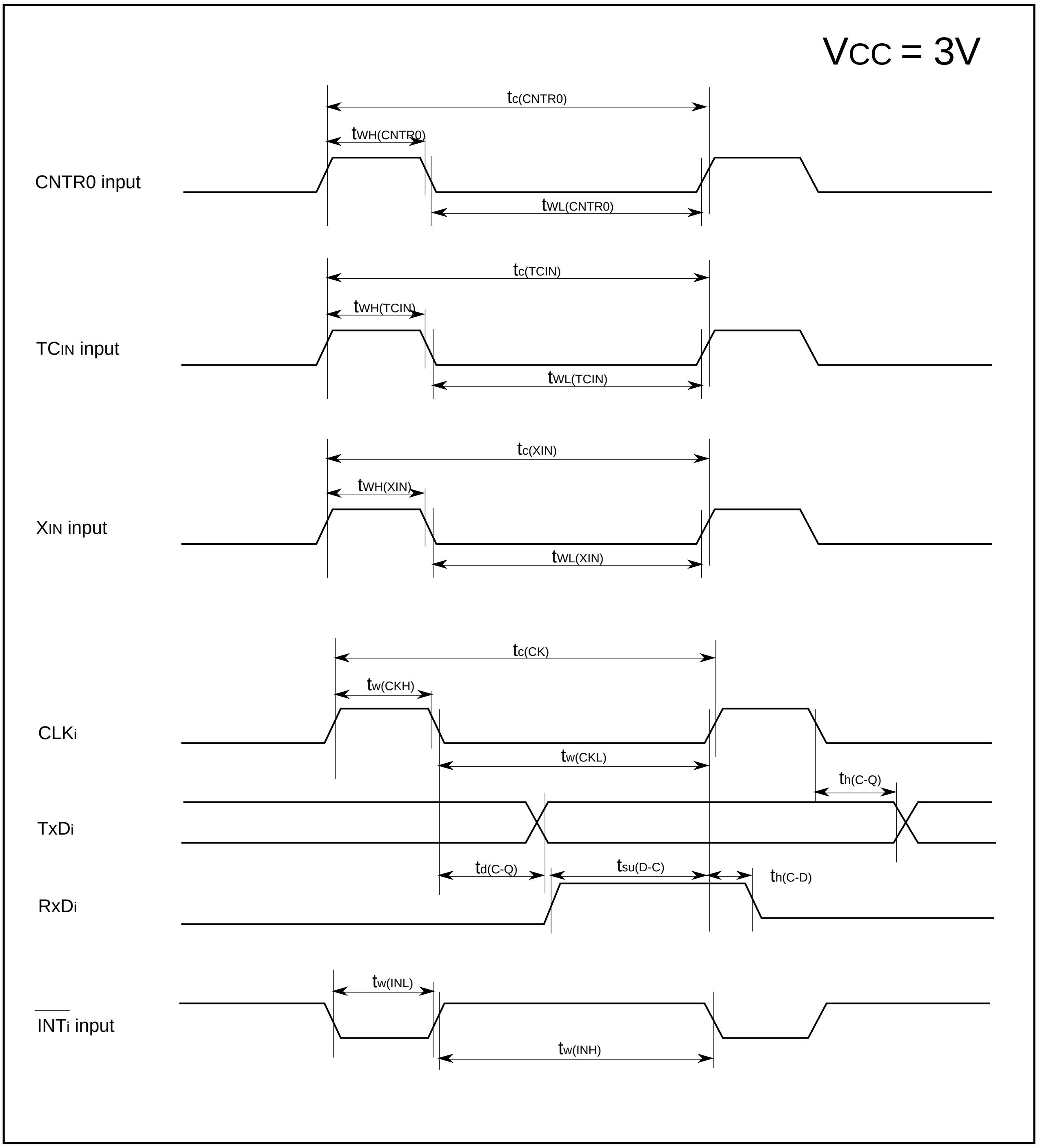


Figure 1.19.3 Vcc=3V timing diagram

REVISION HISTORY	M30100/M30101/M30102 GROUP DATA SHEET

Rev.	Date	Description	
		Page	Summary
B	04/20/01	All pages	Figure and Table numbers are revised.
		1	Features are partly revised.
		1	Page numbers of Table of Contents are partly revised.
		2 - 4	Figure 1.1.1 to Figure 1.1.3 are partly revised.
		5 - 7	Figure 1.1.4 to Figure 1.1.6 are partly revised.
		8	Table 1.1.1 is partly revised.
		9	Figure 1.1.7 is partly revised.
		9	Figure 1.1.8 is partly revised.
		10	Pin description is partly revised.
		11	Explanation of “Memory” is partly revised.
		15	Explanation of “Reset” is partly revised.
		15	Figure 1.5.2 (example reset circuit for voltage check circuit)is added.
		15	Figure 1.5.3 is partly revised.
		16	Figure 1.5.4 is partly revised.
		17	Explanation of “Software Reset” is partly revised.
		17	Processor mode register 0 in Figure 1.5.5 is partly revised. Note 2 is deleted.
		17	Processor mode register 1 is added to Figure 1.5.5.
		18 - 19	Figure 1.6.1 and Figure 1.6.2 are partly revised.
		21	Table 1.8.1 is partly revised.
		21	Figure 1.8.1 is partly revised. External RC oscillator is added.
		24	Figure 1.8.4 is partly revised.
		25	Figure 1.8.5 is added.
		26	Explanation of “Stop Mode” is partly revised.
		27	Table 1.8.4 is partly revised.
		29	Figure 1.9.1 is partly revised.
		31	Explanation of “Oscillation Stop Detection Function” is partly revised.
		31	Table 1.10.1 is partly revised.
		32	Figure 1.10.1 is partly revised.
		32	Figure 1.10.2 is partly revised. Note 2 and Note 3 is partly revised. Note 5 is deleted.
		33	Explanation of “Oscillation stop detection interrupt enable bit (CM21) partly revised.
		33	Operation select bit (CM27) is deleted.
		35	Figure 1.11.1 is partly revised.
		38	“UART1 receive interrupt” in (1) Special interrupts is partly revised.
		38	“Timer C interrupt” in (2) Peripheral I/O interrupt is partly revised.
		40	Table 1.12.2 is partly revised.
		42	Figure 1.12.3 is partly revised.
		46	Table 1.12.5 is partly revised.
		50	Figure 1.12.9 is partly revised.
		51	Explanation of “$\overline{\text{INT}}$ interrupt” is partly revised.
		51	External interrupt enable register in Figure 1.12.10 is partly revised.
		53	Explanation of “Key interrupt” is partly revised.
		53	Figure 1.12.13 is partly revised.
		53	Figure 1.12.14 is partly revised.
		56	Explanation of “Watchdog Timer” is partly revised.
		58	Explanation of “Timer 1” is partly revised.
		59	Figure 1.14.2 is partly revised.
		59	Figure 1.14.3 is partly revised. Note 1 and Note 2 are added.
		60	Explanation of “(2) Pulse output mode” is partly revised.
		60	Explanation of “(4) Pulse width measure mode” is partly revised.

REVISION HISTORY	M30100/M30101/M30102 GROUP DATA SHEET

Rev.	Date	Description	
		Page	Summary
B	04/20/01	61 61 61 62 63 64 65 65 66 66 67 68 72 72 72 73 73 74 75 78 78 78 79 79 81 84 85 86 89 89 90 91 92 94 94 95 95 96 96 97 98 100 101 102 103 104 105	Explanation of “(5) Pulse period measure mode” is partly revised. Explanation of precaution is partly revised. Figure 1.14.4 is partly revised. Explanation of Timer Y is partly revised. Explanation of “(2) Programmable waveform generation mode” is partly revised. Figure 1.14.5 is partly revised. Figure 1.14.6 is partly revised. Note 1 and Note 2 are added to Timer Y, Z mode register in Figure 1.14.6. Explanation of Timer Z is partly revised. Figure 1.14.7 is partly revised. Figure 1.14.9 is partly revised. Explanation of “(2) Programmable waveform generation mode” is partly revised. Explanation of “Timer C” is partly revised. Figure 1.14.10 is partly revised. Note 1 is added to Timer C control register 0 in Figure 1.14.10. Figure 1.14.11 is partly revised. Figure 1.14.12 is partly revised. Table 1.14.1 and its Note are partly revised. Figure 1.15.1 is partly revised. Note is added to UARTi transmit/receive mode register in Figure 1.15.4 is partly revised. UARTi transmit/receive control register 0 in Figure 1.15.4 is partly revised. Note 1 and Note 2 of UARTi transmit/receive control register 0 in Figure 1.15.4 are deleted . Note 1 is added to UARTi transmit/receive control register 1 in Figure 1.15.5. UARTi transmit/receive control register 2 is added to Figure 1.15.5. Table 1.15.2 is partly revised. Figure 1.15.10 is partly revised. Table 1.15.3 is partly revised. Table 1.15.4 is partly revised. Explanation of “A-D Converter” is partly revised. Table 1.16.1 is partly revised. Figure 1.16.1 is partly revised. A-D control register 0 in Figure 1.16.2 is partly revised. Figure 1.16.3 is partly revised. “Input pin” and Note in Table 1.16.3 are revised. Figure 1.16.5 is partly revised. Explanation of “Extended analog input pins” is partly revised. Explanation of “External operation amp connection mode” is partly revised. Explanation of “D-A Converter” is partly revised. Figure 1.17.1 is partly revised. Figure 1.17.3 is partly revised. Explanation of “Programmable I/O Ports” is partly revised. Figure 1.18.2 is revised. Figure 1.18.3 is revised. Figure 1.18.4 is revised. Figure 1.18.5 is revised. Figure 1.18.6 is revised. Table 1.18.1 is revised.
B1	05/15/01	15 19 31	Figure 1.5.3 is partly revised. Figure 1.6.2 is partly revised. Table 1.10.1 is partly revised.

Rev.	Date	Description	
		Page	Summary
B1	05/15/01	32	Figure 1.10.1 is partly revised.
		51	Explanation of “$\overline{\text{INT}}$ interrupt” is partly revised.
		89	Note 3 is added to Table 1.16.1.
		91	Figure 1.16.2 is partly revised.
		92	Figure 1.16. 3 is partly revised.
		93	Table 1.16.2 is partly revised.
		94	Table 1.16.3 is partly revised.
		94	Figure 1.16.5 is partly revised.
C1	11/20/01	01	Features are partly revised.
		01	Page numbers of Table of Contents are partly revised.
		02 - 05	Figure 1.1.1 to 1.1.4 are partly revised.
		08	Table 1.1.1 is partly revised
		09	Explanation of (3) package if partly revised.
		09	Figure 1.1.7 is partly revised.
		10	Pin description is partly revised.
		11	Figure 1.3.1 is partly revised.
		15	Explanation of reset is partly revised.
		15	Figure 1.5.3 is partly revised.
		16	Figure 1.5.4 is partly revised.
		19	Figure 1.6.2 is partly revised.
		21	Figure 1.8.1 is partly revised.
		22	Explanation is partly revised.
		22	Figure 1.8.3 is partly revised.
		23	Explanation of (1)main clock, (3)BCLK and (7)f_{RING} are partly revised.
		24	Note (2) and (5) of register CM1 in Figure 1.8.4 are partly revised.
		25	Register CM2 in Figure 1.8.5 is partly revised.
		26	Explanation of stop mode is partly revised.
		27	Explanation of “status transition of BCLK”, (3)division by 8 mode, (5)no-division mode and (8)ring oscillation mode are partly revised.
		27	Note is added.
		29	Explanation of power control is partly revised.
		31	Figure 1.9.2 is partly revised.
		32	Explanation of oscillation stop detection function is partly revised.
		32	Table 1.10.1 is partly revised.
		33	Figure 1.10.1 and 1.10.2 are partly revised.
		34	Explanation of CM20 to CM22 are partly revised.
		36	Explanation of protection is partly revised.
		39	Explanation of “UART1 receive interrupt” of (1)special interrupts is partly revised.
		39	Explanation of “$\overline{\text{INT0}}$ to $\overline{\text{INT3}}$ interrupt” of (2)peripheral I/O interrupts is partly revised.
		41	Table 1.12.2 is partly revised.
		45	Program examples are partly revised.
		52	Explanation of $\overline{\text{INT}}$ interrupt is partly revised.
		52	Figure 1.12.10 is partly revised.
		53	Figure 1.12.11 is partly revised.
		54	Explanation of key input interrupt is partly revised.
		54	Figure 1.12.13 and 1.12.14 is partly revised.
		56	Explanation of (1)reading address 0000_{16} is partly revised.
		59 - 72	“Latch” used for reload register related are changed to “reload register.”
		59 - 72	Expression of counter content is changed from “00h” to “00_{16}.”

REVISION HISTORY	M30100/M30101/M30102 GROUP DATA SHEET

Rev.	Date	Description	
		Page	Summary
		60	Figure 1.14.3 is partly revised.
		61	Explanation of (3) event counter mode is partly revised.
		61	Explanation of (4) pulse width measure mode is partly revised.
		62	Explanation of (5) pulse period measure mode is partly revised.
		62	Figure 1.14.4 is partly revised.
		64	Explanation of (2) programmable waveform generation mode is partly revised.
		65	Explanation of “use of the waveform extend function” is added.
		65	Last paragraph in precaution is partly revised.
		66	Figure 1.14.6 is partly revised.
		67	Figure 1.14.7 is partly revised.
		68	Figure 1.14.8 and 1.14.9 are partly revised.
		69	Explanation of (2) programmable waveform generation mode is partly revised.
		70	Explanation of (3) programmable one-shot generation mode is partly revised.
		71	Explanation of (4) programmable wait one-shot generation mode is partly revised.
		72	Explanation of “use of the waveform extend function” is added.
		72	Explanation of “change of set count values” is partly revised.
		72	Last paragraph in precaution is partly revised.
		73	Figure 1.14.10 is partly revised.
		76	Figure 1.15.1 is partly revised.
		77	Figure 1.15.2 is partly revised.
		78	Figure 1.15.3 is partly revised.
		79	Figure 1.15.4 is partly revised.
		80	Figure 1.15.5 is partly revised.
		81	Table 1.15.1 is partly revised.
		83	Figure 1.15.7 is partly revised.
		85	(e) RxD_1 input pin selection function (UART) is added.
		86	Table 1.15.3 is partly revised.
		89	Figure 1.15.13 is partly revised.
		89	(b) RxD_1 input pin selection function (UART) is added.
		90	Table 1.16.1, Note 2 is partly revised.
		91	Figure 1.16.1 is partly revised.
		92	Figure 1.16.2 is partly revised.
		94	Figure 1.16.4 is partly revised
		95	Table 1.16.3 is partly revised.
		95	Figure 1.16.5 is partly revised
		96	Explanation of extended analog input pins is partly revised.
		96	Figure 1.16.6 is partly revised.
		99	Explanation of programmable I/O ports is partly revised.
		100 - 105	Figure 1.18.1 to 1.18.6 are partly revised.
		106	Table 1.18.1 is partly revised.
		107 - 108	Explanation of usage of precaution is added.
		109 - 119	Section of electric characteristics is added.

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
M30100/M30101/M30102 Group Specification REV.C1

November First Edition 2001

Editioned by
Committee of editing of Mitsubishi Semiconductor

Published by
Mitsubishi Electric Corp., Kitaitami Works