To all our customers

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

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

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

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

Description

The M16C/30L group of single-chip microcomputers are built using the high-performance silicon gate CMOS process using a M16C/60 Series CPU core and are packaged in a 100-pin plastic molded QFP. These single-chip microcomputers operate using sophisticated instructions featuring a high level of instruction efficiency. With 1M bytes of address space, low voltage (2.2V to 3.6V), they are capable of executing instructions at high speed. They also feature a built-in multiplier and DMAC, making them ideal for controlling office, communications, industrial equipment, and other high-speed processing applications.
The M16C/30L group includes a wide range of products with different internal memory sizes and various package types.

Features

- Memory capacity ..... ROM (See Figure 1.1.4. ROM Expansion)
  RAM 2K to 3K bytes
- Shortest instruction execution time ..... 62.5ns (f(X_{IN})=16MHz, V_{CC}=3.0V to 3.6V)
  142.9ns (f(X_{IN})=7MHz, V_{CC}=2.4V to 3.6V without software wait)
- Supply voltage ..... 3.0V to 3.6V (f(X_{IN})=16MHz, without software wait)
  2.4V to 3.6V (f(X_{IN})=7MHz, without software wait)
  2.2V to 3.6V (f(X_{IN})=7MHz, with software one-wait)
- Low power consumption ..... 34.0mW (V_{CC} = 3V, f(X_{IN})=10MHz, without software wait)
  66.0mW (V_{CC} = 3.3V, f(X_{IN})=16MHz, without software wait)
- Interrupts ..... 16 internal and 5 external interrupt sources, 4 software interrupt sources; 7 levels (including key input interrupt)
- Multifunction 16-bit timer ..... 3 output timers + 2 input timers
- Serial I/O ..... 3 channels (3 for UART or clock synchronous)
- DMAC ..... 1 channels (trigger: 14 sources)
- A-D converter ..... 10 bits X 8 channels (Expandable up to 10 channels)
- Watchdog timer ..... 1 line
- Programmable I/O port ..... 87 lines
- Input port ..... 1 line (P8$_5$ shared with $\overline{NMI}$ pin)
- Memory expansion ..... Available (to a maximum of 1M bytes)
- Chip select output ..... 4 lines
- Clock generating circuit ..... 2 built-in clock generation circuits
  (built-in feedback resistor, and external ceramic or quartz oscillator)

Applications

Audio, cameras, office equipment, communications equipment, portable equipment

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

Pin Configuration

Figures 1.1.1 and 1.1.2 show the pin configurations (top view).

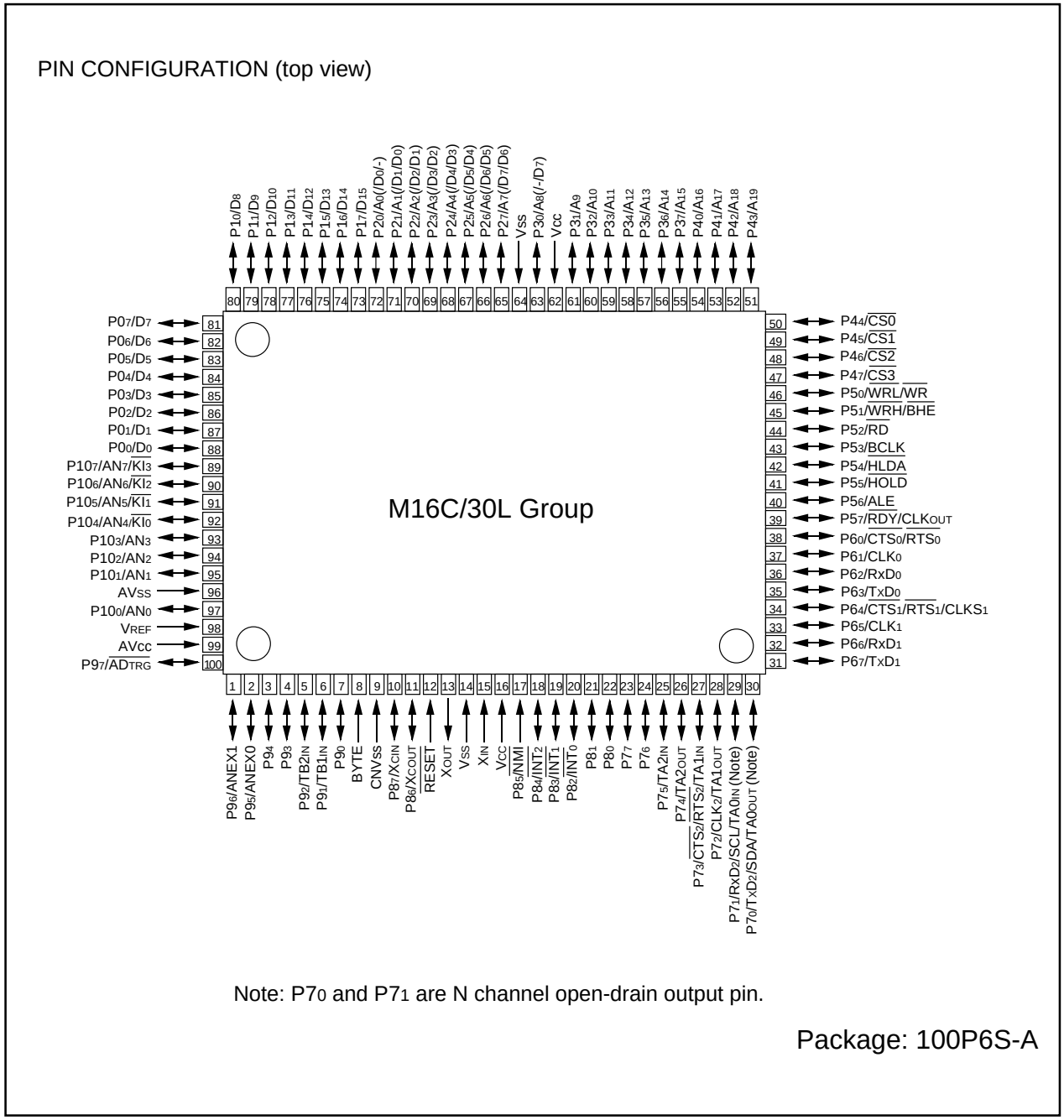
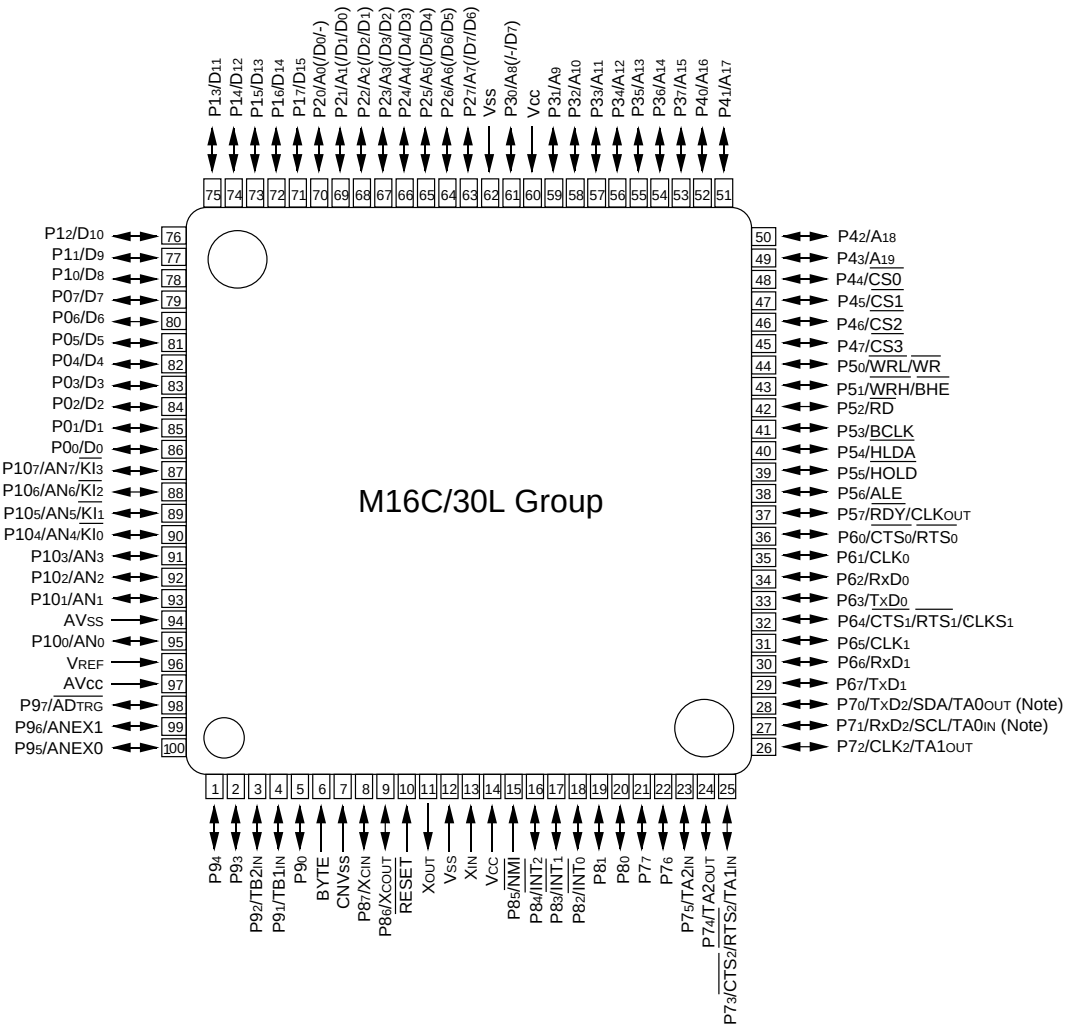


Figure 1.1.1. Pin configuration (top view)

PIN CONFIGURATION (top view)



Note: P70 and P71 are N channel open-drain output pin.

Package: 100P6Q-A

Figure 1.1.2. Pin configuration (top view)

Block Diagram

Figure 1.1.3 is a block diagram of the M16C/30L group.

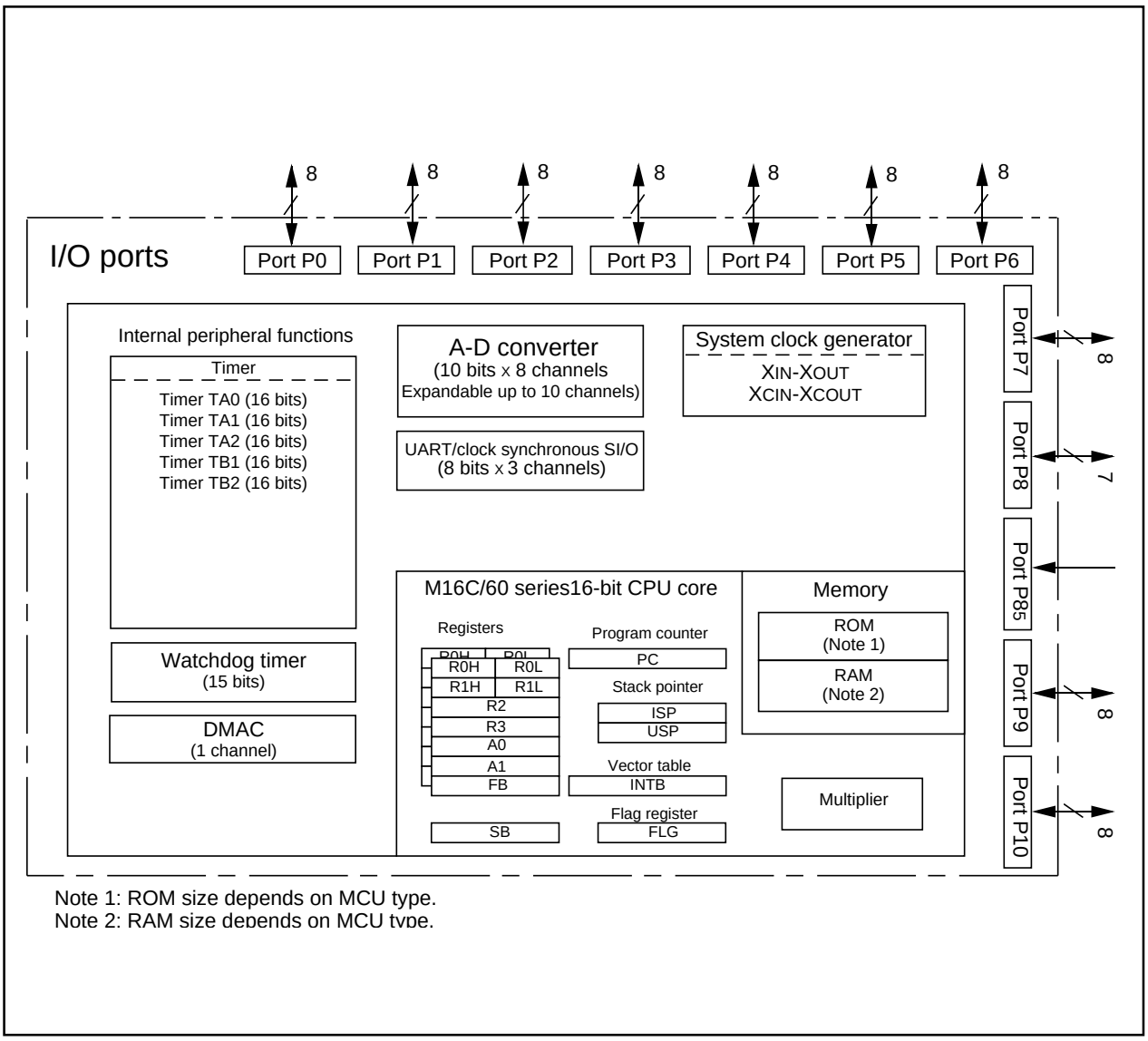


Figure 1.1.3. Block diagram of M16C/30L group

Performance Outline

Table 1.1.1 is a performance outline of M16C/30L group.

Table 1.1.1. Performance outline of M16C/30L group

Item		Performance
Number of basic instructions		91 instructions
Shortest instruction execution time		62.5ns(f(XIN)=16MHz, VCC=3.0V to 3.6V) 142.9ns(f(XIN)=7MHz, VCC=2.4V to 3.6V, without software wait)
Memory capacity	ROM	(See the figure 1.1.4. ROM Expansion)
	RAM	2K to 3K bytes
I/O port	P0 to P10 (except P85)	8 bits x 10, 7 bits x 1
Input port	P85	1 bit x 1
Multifunction timer	TA0, TA1, TA2	16 bits x 3
	TB1, TB2	16 bits x 2
Serial I/O	UART0, UART1, UART2	(UART or clock synchronous) x 3
A-D converter		10 bits x (8+2) channels
DMAC		1 channels (trigger: 14 sources)
Watchdog timer		15 bits x 1 (with prescaler)
Interrupt		16 internal and 5 external sources, 4 software sources, 7 levels
Clock generating circuit		2 built-in clock generation circuits (built-in feedback resistor, and external ceramic or quartz oscillator)
Supply voltage		3.0V to 3.6V (f(XIN)=16MHz, without software wait) 2.4V to 3.6V (f(XIN)=7MHz, without software wait) 2.2V to 3.6V (f(XIN)=7MHz, with software one-wait)
Power consumption		34.0mW (VCC=3V, f(XIN)=10MHz, without software wait) 66.0mW (VCC=3.3V, f(XIN)=16MHz, without software wait)
I/O characteristics	I/O withstand voltage	3.3V
	Output current	1mA
Memory expansion		Available (to a maximum of 1M bytes)
Device configuration		CMOS high performance silicon gate
Package		100-pin plastic mold QFP

Mitsubishi plans to release the following products in the M16C/30L group:

(1) Support for mask ROM version

(2) ROM capacity

(3) Package

100P6S-A : Plastic molded QFP

100P6Q-A : Plastic molded QFP

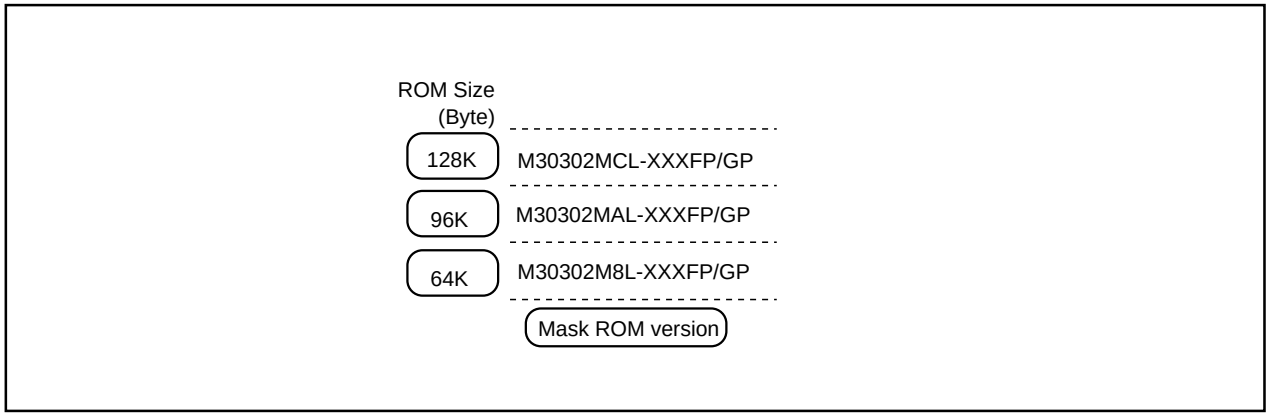


Figure 1.1.4. ROM expansion

The M16C/30L group products currently supported are listed in Table 1.1.2.

Table 1.1.2. M16C/30L group

June, 2002

Type No.	ROM capacity	RAM capacity	Package type	Remarks
M30302M8L-XXXFP **	64K byte	2K byte	100P6S-A	Mask ROM version
M30302M8L-XXXGP **			100P6Q-A	
M30302MAL-XXXFP **	96K byte	3K byte	100P6S-A	
M30302MAL-XXXGP **			100P6Q-A	
M30302MCL-XXXFP *	128K byte		100P6S-A	
M30302MCL-XXXGP *			100P6Q-A	

**: Under development

* : New product

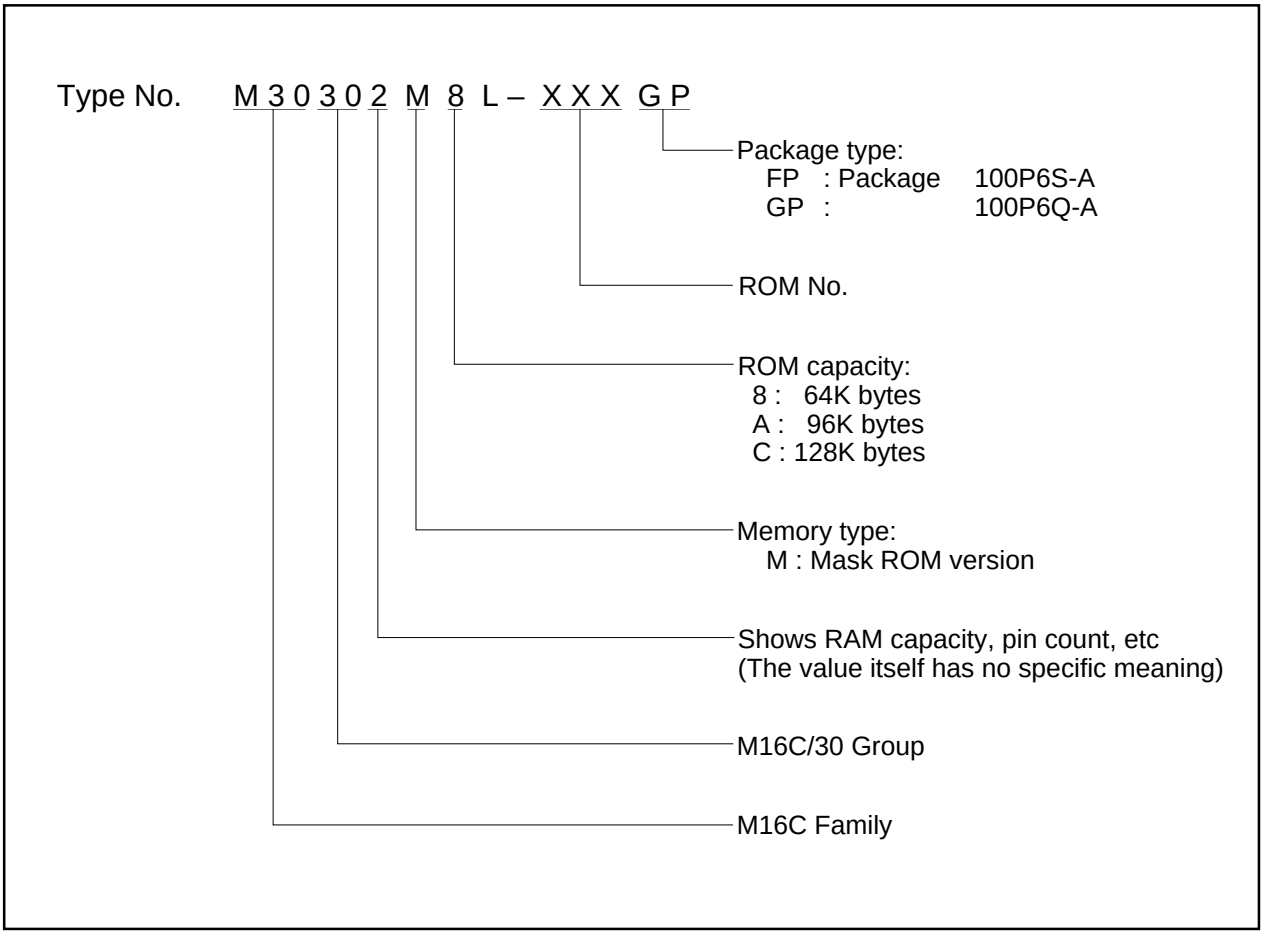


Figure 1.1.5. Type No., memory size, and package

Pin Description

Pin name	Signal name	I/O type	Function
V_{CC}, V_{SS}	Power supply input		Supply 2.2V to 3.6 V to the V_{CC} pin. Supply 0 V to the V_{SS} pin.
CNV_{SS}	CNV_{SS}	Input	This pin switches between processor modes. Connect this pin to the V_{SS} pin when after a reset you want to start operation in single-chip mode (memory expansion mode) or the V_{CC} pin when starting operation in microprocessor mode.
$\overline{RESET}$	Reset input	Input	A "L" on this input resets the microcomputer.
X_{IN} X_{OUT}	Clock input Clock output	Input Output	These pins are provided for the main clock generating circuit.Connect a ceramic resonator or crystal between the X_{IN} and the X_{OUT} pins. To use an externally derived clock, input it to the X_{IN} pin and leave the X_{OUT} pin open.
BYTE	External data bus width select input	Input	This pin selects the width of an external data bus. A 16-bit width is selected when this input is "L"; an 8-bit width is selected when this input is "H". This input must be fixed to either "H" or "L". Connect this pin to the V_{SS} pin when not using external data bus.
AV_{CC}	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to V_{CC}.
AV_{SS}	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to V_{SS}.
V_{REF}	Reference voltage input	Input	This pin is a reference voltage input for the A-D converter.
$P0_0$ to $P0_7$	I/O port P0	Input/output	This is an 8-bit CMOS I/O port. It has an input/output port direction register that allows the user to set each pin for input or output individually. When used for input in single-chip mode, the port can be set to have or not have a pull-up resistor in units of four bits by software. In memory expansion and microprocessor modes, selection of the internal pull-resistor is not available.
D_0 to D_7		Input/output	When set as a separate bus, these pins input and output data (D_0–D_7).
$P1_0$ to $P1_7$	I/O port P1	Input/output	This is an 8-bit I/O port equivalent to P0.
D_8 to D_{15}		Input/output	When set as a separate bus, these pins input and output data (D_8–D_{15}).
$P2_0$ to $P2_7$	I/O port P2	Input/output	This is an 8-bit I/O port equivalent to P0.
A_0 to A_7		Output	These pins output 8 low-order address bits (A_0–A_7).
A_0/D_0 to A_7/D_7		Input/output	If the external bus is set as an 8-bit wide multiplexed bus, these pins input and output data (D_0–D_7) and output 8 low-order address bits (A_0–A_7) separated in time by multiplexing.
A_0 A_1/D_0 to A_7/D_6		Output Input/output	If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D_0–D_6) and output address (A_1–A_7) separated in time by multiplexing. They also output address (A_0).
$P3_0$ to $P3_7$	I/O port P3	Input/output	This is an 8-bit I/O port equivalent to P0.
A_8 to A_{15}		Output	These pins output 8 middle-order address bits (A_8–A_{15}).
A_8/D_7, A_9 to A_{15}		Input/output Output	If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D_7) and output address (A_8) separated in time by multiplexing. They also output address (A_9–A_{15}).
$P4_0$ to $P4_7$	I/O port P4	Input/output	This is an 8-bit I/O port equivalent to P0.
A_{16} to A_{19}, $\overline{CS_0}$ to $\overline{CS_3}$		Output Output	These pins output A_{16}–A_{19} and $\overline{CS_0}$–$\overline{CS_3}$ signals. A_{16}–A_{19} are 4 high-order address bits. $\overline{CS_0}$–$\overline{CS_3}$ are chip select signals used to specify an access space.

Pin Description

Pin name	Signal name	I/O type	Function
P5$_0$ to P5$_7$	I/O port P5	Input/output	This is an 8-bit I/O port equivalent to P0. In single-chip mode, P5$_7$ in this port outputs a divide-by-8 or divide-by-32 clock of X_{IN} or a clock of the same frequency as X_{CIN} as selected by software.
$\overline{WRL}$ / $\overline{WR}$, $\overline{WRH}$ / BHE, RD, BCLK, HLDA, HOLD, ALE, $\overline{RDY}$		Output Output Output Output Output Input Output Input	Output $\overline{WRL}$, $\overline{WRH}$ ($\overline{WR}$ and $\overline{BHE}$), $\overline{RD}$, BCLK, $\overline{HLDA}$, and ALE signals. $\overline{WRL}$ and $\overline{WRH}$, and $\overline{BHE}$ and $\overline{WR}$ can be switched using software control. ■ $\overline{WRL}$, $\overline{WRH}$, and $\overline{RD}$ selected With a 16-bit external data bus, data is written to even addresses when the $\overline{WRL}$ signal is “L” and to the odd addresses when the $\overline{WRH}$ signal is “L”. Data is read when $\overline{RD}$ is “L”. ■ $\overline{WR}$, $\overline{BHE}$, and $\overline{RD}$ selected Data is written when $\overline{WR}$ is “L”. Data is read when $\overline{RD}$ is “L”. Odd addresses are accessed when $\overline{BHE}$ is “L”. Use this mode when using an 8-bit external data bus. While the input level at the $\overline{HOLD}$ pin is “L”, the microcomputer is placed in the hold state. While in the hold state, $\overline{HLDA}$ outputs a “L” level. ALE is used to latch the address. While the input level of the $\overline{RDY}$ pin is “L”, the microcomputer is in the ready state.
P6$_0$ to P6$_7$	I/O port P6	Input/output	This is an 8-bit I/O port equivalent to P0. When used for input in single-chip, memory expansion, and microprocessor modes, the port can be set to have or not have a pull-up resistor in units of four bits by software. Pins in this port also function as UART0 and UART1 I/O pins as selected by software.
P7$_0$ to P7$_7$	I/O port P7	Input/output	This is an 8-bit I/O port equivalent to P6 (P7$_0$ and P7$_1$ are N channel open-drain output). Pins in this port also function as timer A0–A2, or UART2 I/O pins as selected by software.
P8$_0$ to P8$_4$, P8$_6$, P8$_7$, P8$_5$	I/O port P8 I/O port P8$_5$	Input/output Input/output Input/output Input	P8$_0$ to P8$_4$, P8$_6$, and P8$_7$ are I/O ports with the same functions as P6. Using software, they can be made to function as the I/O pins for the input pins for external interrupts. P8$_6$ and P8$_7$ can be set using software to function as the I/O pins for a sub clock generation circuit. In this case, connect a quartz oscillator between P8$_6$ (X_{COUT} pin) and P8$_7$ (X_{CIN} pin). P8$_5$ is an input-only port that also functions for $\overline{NMI}$. The $\overline{NMI}$ interrupt is generated when the input at this pin changes from “H” to “L”. The $\overline{NMI}$ function cannot be cancelled using software. The pull-up cannot be set for this pin.
P9$_0$ to P9$_7$	I/O port P9	Input/output	This is an 8-bit I/O port equivalent to P6. Pins in this port also function as, timer B1, B2 input pins, A-D converter extended input pins, or A-D trigger input pins as selected by software.
P10$_0$ to P10$_7$	I/O port P10	Input/output	This is an 8-bit I/O port equivalent to P6. Pins in this port also function as A-D converter input pins as selected by software. Furthermore, P10$_4$ –P10$_7$ also function as input pins for the key input interrupt function.

Operation of Functional Blocks

The M16C/30L group accommodates certain units in a single chip. These units include ROM and RAM to store instructions and data and the central processing unit (CPU) to execute arithmetic/logic operations. Also included are peripheral units such as timers, serial I/O, DMAC, A-D converter, and I/O ports.
The following explains each unit.

Memory

Figure 1.3.1 is a memory map of the M16C/30L group. The address space extends the 1M bytes from address 00000_{16} to $FFFFF_{16}$. From $FFFFF_{16}$ down is ROM. For example, in the M30302MCL-XXXGP, there is 128K bytes of internal ROM from $E0000_{16}$ to $FFFFF_{16}$. The vector table for fixed interrupts such as the reset and $\overline{NMI}$ 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 M30302MCL-XXXGP, 3K bytes of internal RAM is mapped to the space from 00400_{16} to $00FFF_{16}$. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.
The SFR area is mapped to 00000_{16} to $003FF_{16}$. This area accommodates the control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers, etc. Figures 1.6.1 to 1.6.3 are location of peripheral unit control registers. 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.
In memory expansion mode and microprocessor mode, a part of the spaces are reserved and cannot be used. For example, in the M30302MCL-XXXGP, the following spaces cannot be used.

- The space between 01800_{16} and $03FFF_{16}$ (Memory expansion and microprocessor modes)
- The space between $D0000_{16}$ and $DFFFF_{16}$ (Memory expansion mode)

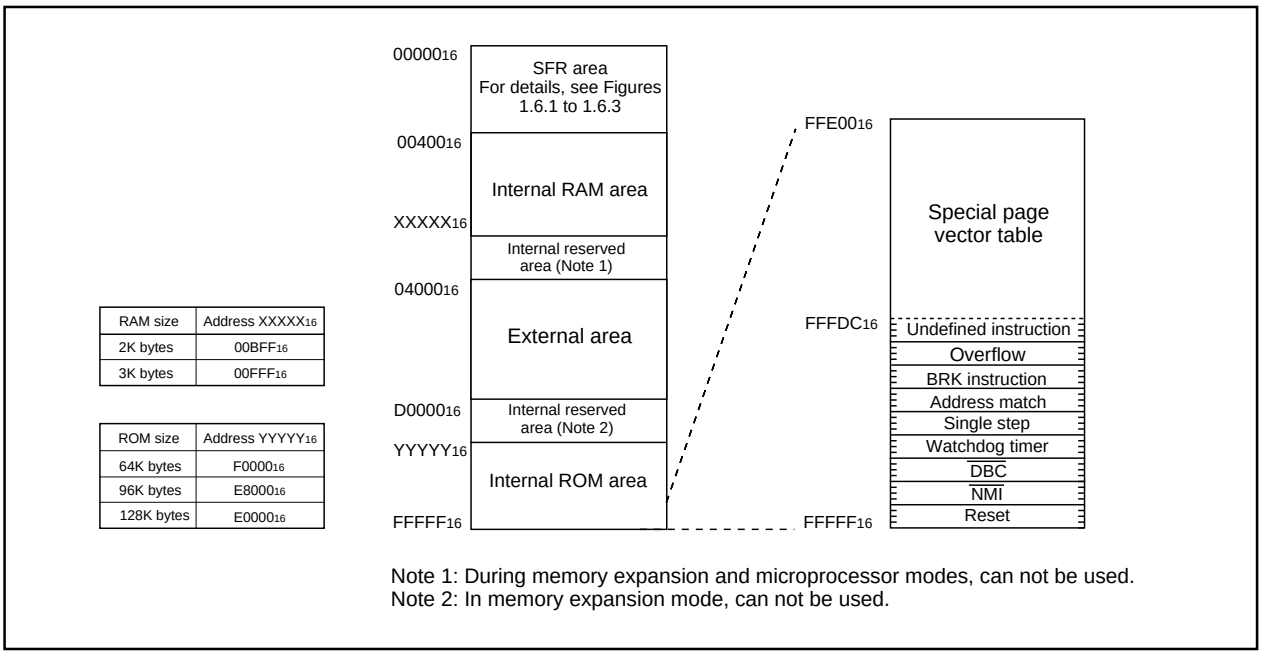
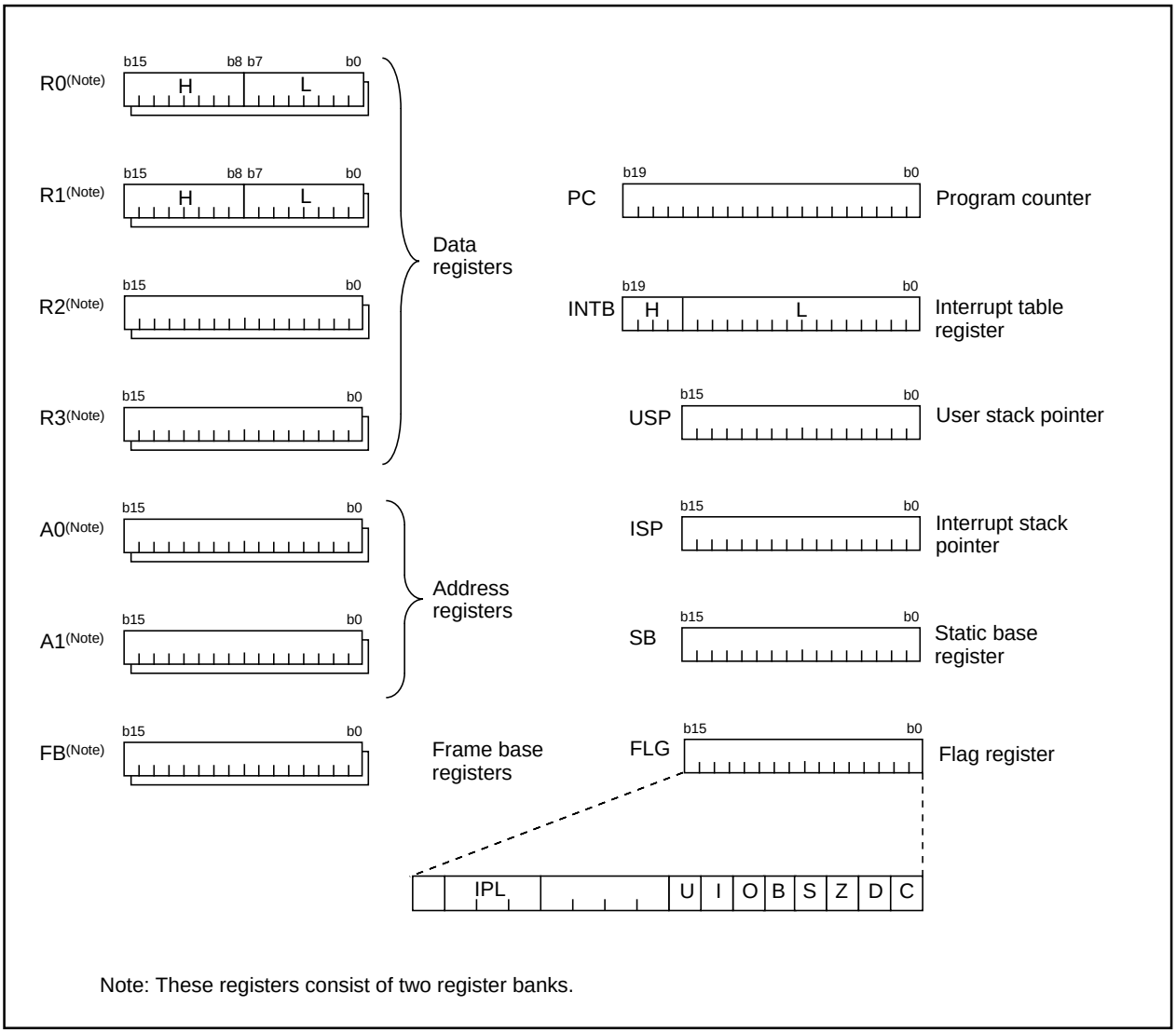


Figure 1.3.1. Memory map

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, R1, R1H, 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.

Register R0 can be used as separate 8-bit data registers, R0H and R0L. Register R1 can be used as separate 8-bit data registers, R1H and R1L. In some instructions, registers R2 and R0, as well as R3 and R1 can use as 32-bit data registers (R2R0 or 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 or 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; otherwise, cleared to “0”.
- **Bit 6: Interrupt enable flag (I flag)**
  This flag enables a maskable interrupt.
  An interrupt is disabled when this flag is “0”, and is enabled when this flag is “1”. This flag is cleared to “0” when the interrupt is acknowledged.

- **Bit 7: Stack pointer select flag (U flag)**
  Interrupt stack pointer (ISP) is selected when this flag is “0” ; user stack pointer (USP) is selected when this flag is “1”.
  This flag is cleared to “0” when a hardware interrupt is acknowledged or an INT instruction of software interrupt Nos. 0 to 31 is executed.
- **Bits 8 to 11: Reserved area**
- **Bits 12 to 14: Processor interrupt priority level (IPL)**
  Processor interrupt priority level (IPL) is configured with three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.
  If a requested interrupt has priority greater than the processor interrupt priority level (IPL), the interrupt is enabled.
- **Bit 15: Reserved area**

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

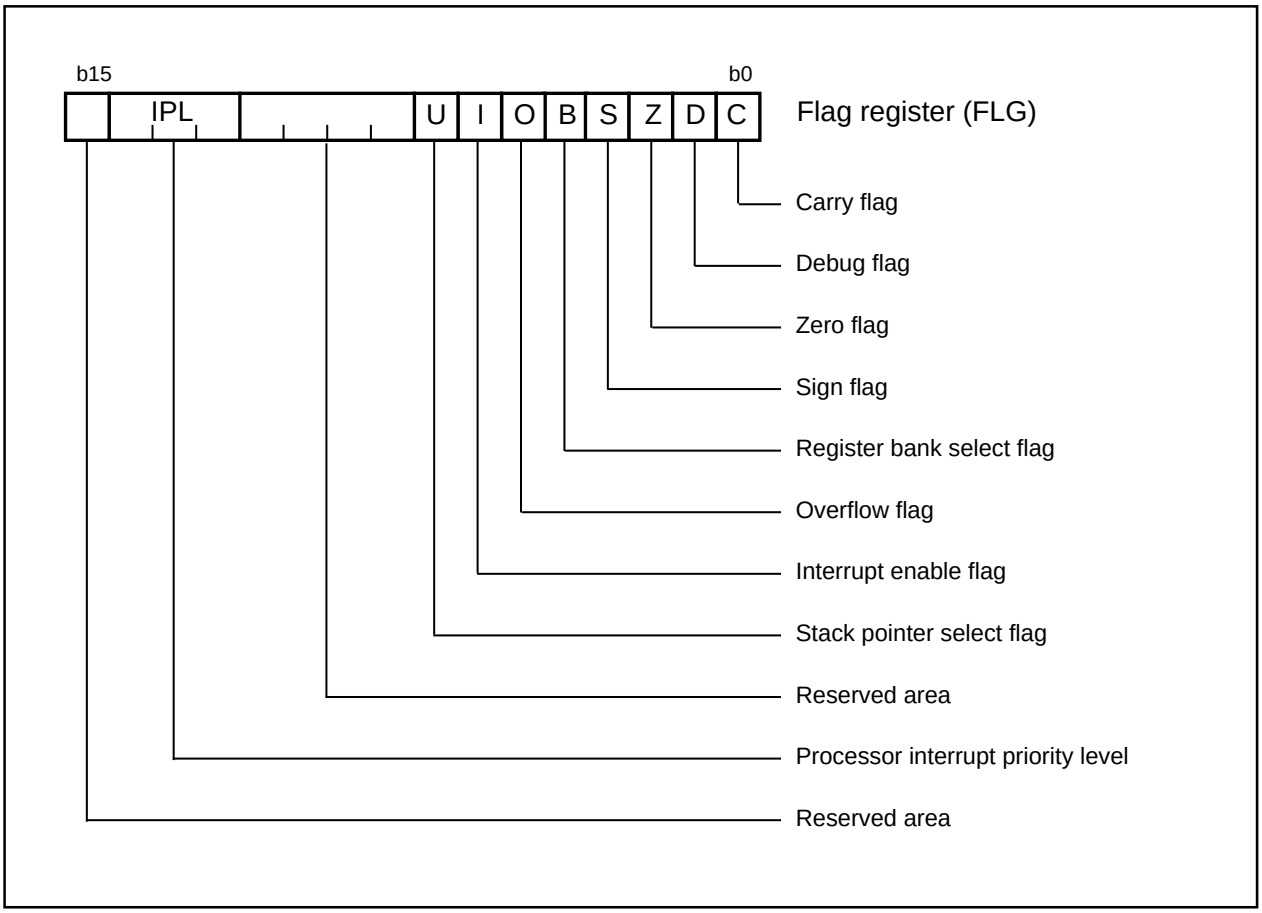


Figure 1.4.2. Flag register (FLG)

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 while main clock is stable, the reset status is cancelled and program execution resumes from the address in the reset vector table.

The RAM is undefined at power on. The initial value must therefore be set. When a reset signal is applied while the CPU is writing a value to the RAM, the value may be set as unknown due to the termination of the CPU access.

Figure 1.5.1 shows the example reset circuit. Figure 1.5.2 shows the reset sequence.

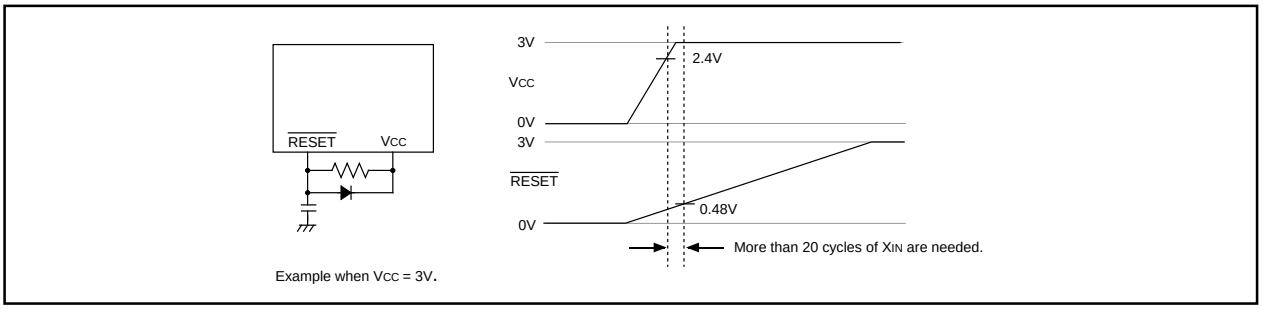


Figure 1.5.1. Example reset circuit

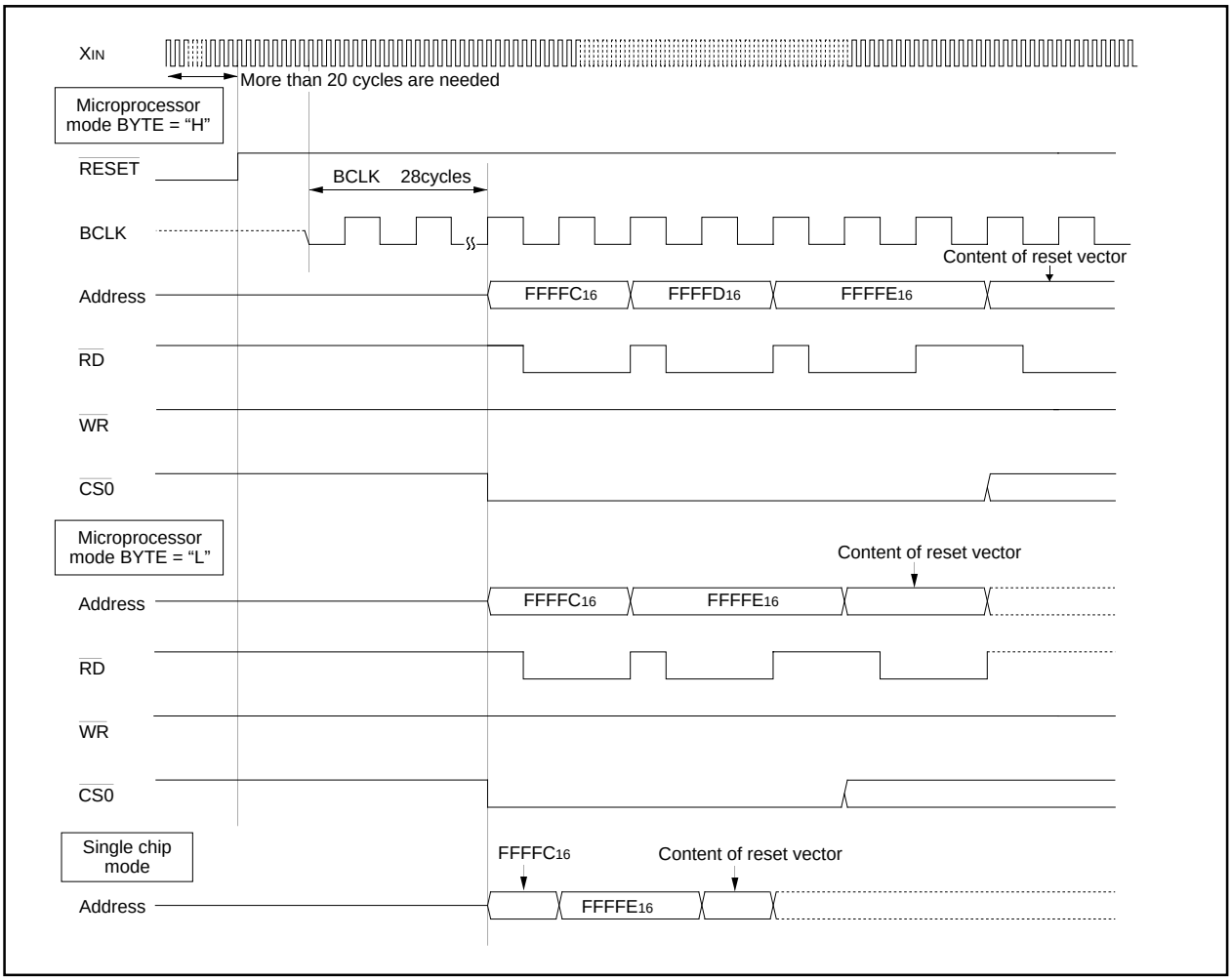


Figure 1.5.2. Reset sequence

Table 1.5.1 shows the statuses of the other pins while the $\overline{\text{RESET}}$ pin level is “L”. Figures 1.5.3 and 1.5.4 show the internal status of the microcomputer immediately after the reset is cancelled.

Table 1.5.1. Pin status when $\overline{\text{RESET}}$ pin level is “L”

Pin name	Status		
	CNVSS = VSS	CNVSS = VCC	
		BYTE = VSS	BYTE = VCC
P0	Input port (floating)	Data input (floating)	Data input (floating)
P1	Input port (floating)	Data input (floating)	Input port (floating)
P2, P3, P40 to P43	Input port (floating)	Address output (undefined)	Address output (undefined)
P44	Input port (floating)	$\overline{\text{CS0}}$ output (“H” level is output)	$\overline{\text{CS0}}$ output (“H” level is output)
P45 to P47	Input port (floating)	Input port (floating) (pull-up resistor is on)	Input port (floating) (pull-up resistor is on)
P50	Input port (floating)	$\overline{\text{WR}}$ output (“H” level is output)	$\overline{\text{WR}}$ output (“H” level is output)
P51	Input port (floating)	$\overline{\text{BHE}}$ output (undefined)	$\overline{\text{BHE}}$ output (undefined)
P52	Input port (floating)	$\overline{\text{RD}}$ output (“H” level is output)	$\overline{\text{RD}}$ output (“H” level is output)
P53	Input port (floating)	BCLK output	BCLK output
P54	Input port (floating)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the $\overline{\text{HOLD}}$ pin)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the $\overline{\text{HOLD}}$ pin)
P55	Input port (floating)	$\overline{\text{HOLD}}$ input (floating)	$\overline{\text{HOLD}}$ input (floating)
P56	Input port (floating)	ALE output (“L” level is output)	ALE output (“L” level is output)
P57	Input port (floating)	$\overline{\text{RDY}}$ input (floating)	$\overline{\text{RDY}}$ input (floating)
P6, P7, P80 to P84, P86, P87, P9, P10	Input port (floating)	Input port (floating)	Input port (floating)

(1) Processor mode register 0 (Note)	(0004$_{16}$)··· 00$_{16}$	(20) UART1 transmit interrupt control register	(0053$_{16}$)··· X X X X ? 0 0 0
(2) Processor mode register 1	(0005$_{16}$)··· 0 0 0 0 0 0 X 0	(21) UART1 receive interrupt control register	(0054$_{16}$)··· X X X X ? 0 0 0
(3) System clock control register 0	(0006$_{16}$)··· 0 1 0 0 1 0 0 0	(22) Timer A0 interrupt control register	(0055$_{16}$)··· X X X X ? 0 0 0
(4) System clock control register 1	(0007$_{16}$)··· 0 0 1 0 0 0 0 0	(23) Timer A1 interrupt control register	(0056$_{16}$)··· X X X X ? 0 0 0
(5) Chip select control register	(0008$_{16}$)··· 0 0 0 0 0 0 0 1	(24) Timer A2 interrupt control register	(0057$_{16}$)··· X X X X ? 0 0 0
(6) Address match interrupt enable register	(0009$_{16}$)··· X X X X X X 0 0	(25) Timer B1 interrupt control register	(005B$_{16}$)··· X X X X ? 0 0 0
(7) Protect register	(000A$_{16}$)··· X X X X X 0 0 0	(26) Timer B2 interrupt control register	(005C$_{16}$)··· X X X X ? 0 0 0
(8) Watchdog timer control register	(000F$_{16}$)··· 0 0 0 ? ? ? ? ?	(27) INT0 interrupt control register	(005D$_{16}$)··· X X 0 0 ? 0 0 0
(9) Address match interrupt register 0	(0010$_{16}$)··· 00$_{16}$	(28) INT1 interrupt control register	(005E$_{16}$)··· X X 0 0 ? 0 0 0
	(0011$_{16}$)··· 00$_{16}$	(29) INT2 interrupt control register	(005F$_{16}$)··· X X 0 0 ? 0 0 0
	(0012$_{16}$)··· X X X X 0 0 0 0	(30) Interrupt cause select register	(035F$_{16}$)··· 00$_{16}$
(10) Address match interrupt register 1	(0014$_{16}$)··· 00$_{16}$	(31) UART2 special mode register 3	(0375$_{16}$)··· 00$_{16}$
	(0015$_{16}$)··· 00$_{16}$	(32) UART2 special mode register 2	(0376$_{16}$)··· 00$_{16}$
	(0016$_{16}$)··· X X X X 0 0 0 0	(33) UART2 special mode register	(0377$_{16}$)··· 80$_{16}$
(11) DMA0 control register	(002C$_{16}$)··· 0 0 0 0 0 ? 0 0	(34) UART2 transmit/receive mode register	(0378$_{16}$)··· 00$_{16}$
(12) Bus collision detection interrupt control register	(004A$_{16}$)··· X X X X ? 0 0 0	(35) UART2 transmit/receive control register 0	(037C$_{16}$)··· 0 0 0 0 1 0 0 0
(13) DMA0 interrupt control register	(004B$_{16}$)··· X X X X ? 0 0 0	(36) UART2 transmit/receive control register 1	(037D$_{16}$)··· 0 0 0 0 0 0 1 0
(14) Key input interrupt control register	(004D$_{16}$)··· X X X X ? 0 0 0	(37) Count start flag	(0380$_{16}$)··· 00$_{16}$
(15) A-D conversion interrupt control register	(004E$_{16}$)··· X X X X ? 0 0 0	(38) Clock prescaler reset flag	(0381$_{16}$)··· 0 X X X X X X X
(16) UART2 transmit interrupt control register	(004F$_{16}$)··· X X X X ? 0 0 0	(39) One-shot start flag	(0382$_{16}$)··· 0 0 X 0 0 0 0 0
(17) UART2 receive interrupt control register	(0050$_{16}$)··· X X X X ? 0 0 0	(40) Trigger select flag	(0383$_{16}$)··· 00$_{16}$
(18) UART0 transmit interrupt control register	(0051$_{16}$)··· X X X X ? 0 0 0	(41) Up-down flag	(0384$_{16}$)··· 00$_{16}$
(19) UART0 receive interrupt control register	(0052$_{16}$)··· X X X X ? 0 0 0		

x : Nothing is mapped to this bit
? : Undefined

The content of other registers are undefined when the microcomputer is reset. The initial values must therefore be set. The RAM is undefined at power on. The initial values must therefore be set. When a reset signal is applied while the CPU is writing a value to the RAM, the value may be set as unknown due to the termination of the CPU access.

Note: When the V_{CC} level is applied to the CNV_{SS} pin, it is 03$_{16}$ at a reset.

Figure 1.5.3. Device's internal status after a reset is cleared

(42) Timer A0 mode register	(0396_{16})···	00_{16}	(62) Port P4 direction register	($03EA_{16}$)···	00_{16}
(43) Timer A1 mode register	(0397_{16})···	00_{16}	(63) Port P5 direction register	($03EB_{16}$)···	00_{16}
(44) Timer A2 mode register	(0398_{16})···	00_{16}	(64) Port P6 direction register	($03EE_{16}$)···	00_{16}
(45) Timer B1 mode register	($039C_{16}$)···	00?x0000	(65) Port P7 direction register	($03EF_{16}$)···	00_{16}
(46) Timer B2 mode register	($039D_{16}$)···	00?x0000	(66) Port P8 direction register	($03F2_{16}$)···	00x00000
(47) UART0 transmit/receive mode register	($03A0_{16}$)···	00_{16}	(67) Port P9 direction register	($03F3_{16}$)···	00_{16}
(48) UART0 transmit/receive control register 0	($03A4_{16}$)···	00001000	(68) Port P10 direction register	($03F6_{16}$)···	00_{16}
(49) UART0 transmit/receive control register 1	($03A5_{16}$)···	00000010	(69) Pull-up control register 0	($03FC_{16}$)···	00_{16}
(50) UART1 transmit/receive mode register	($03A8_{16}$)···	00_{16}	(70) Pull-up control register 1(Note)	($03FD_{16}$)···	00_{16}
(51) UART1 transmit/receive control register 0	($03AC_{16}$)···	00001000	(71) Pull-up control register 2	($03FE_{16}$)···	00_{16}
(52) UART1 transmit/receive control register 1	($03AD_{16}$)···	00000010	(72) Port control register	($03FF_{16}$)···	00_{16}
(53) UART transmit/receive control register 2	($03B0_{16}$)···	x0000000	(73) Data registers (R0/R1/R2/R3)		0000_{16}
(54) DMA0 cause select register	($03B8_{16}$)···	00_{16}	(74) Address registers (A0/A1)		0000_{16}
(55) A-D control register 2	($03D4_{16}$)···	0000???0	(75) Frame base register (FB)		0000_{16}
(56) A-D control register 0	($03D6_{16}$)···	00000???	(76) Interrupt table register (INTB)		00000_{16}
(57) A-D control register 1	($03D7_{16}$)···	00_{16}	(77) User stack pointer (USP)		0000_{16}
(58) Port P0 direction register	($03E2_{16}$)···	00_{16}	(78) Interrupt stack pointer (ISP)		0000_{16}
(59) Port P1 direction register	($03E3_{16}$)···	00_{16}	(79) Static base register (SB)		0000_{16}
(60) Port P2 direction register	($03E6_{16}$)···	00_{16}	(80) Flag register (FLG)		0000_{16}
(61) Port P3 direction register	($03E7_{16}$)···	00_{16}			

x : Nothing is mapped to this bit
? : Undefined

The content of other registers are undefined when the microcomputer is reset. The initial values must therefore be set. The RAM is undefined at power on. The initial values must therefore be set. When a reset signal is applied while the CPU is writing a value to the RAM, the value may be set as unknown due to the termination of the CPU access.

Note: When the V_{CC} level is applied to the CNV_{SS} pin, it is 02_{16} at a reset.

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

0000₁₆	
0001₁₆	
0002₁₆	
0003₁₆	
0004₁₆	Processor mode register 0 (PM0)
0005₁₆	Processor mode register 1(PM1)
0006₁₆	System clock control register 0 (CM0)
0007₁₆	System clock control register 1 (CM1)
0008₁₆	Chip select control register (CSR)
0009₁₆	Address match interrupt enable register (AIER)
000A₁₆	Protect register (PRCR)
000B₁₆	
000C₁₆	
000D₁₆	
000E₁₆	Watchdog timer start register (WDTS)
000F₁₆	Watchdog timer control register (WDC)
0010₁₆	Address match interrupt register 0 (RMAD0)
0011₁₆	
0012₁₆	
0013₁₆	
0014₁₆	Address match interrupt register 1 (RMAD1)
0015₁₆	
0016₁₆	
0017₁₆	
0018₁₆	
0019₁₆	
001A₁₆	
001B₁₆	
001C₁₆	
001D₁₆	
001E₁₆	
001F₁₆	
0020₁₆	DMA0 source pointer (SAR0)
0021₁₆	
0022₁₆	
0023₁₆	
0024₁₆	DMA0 destination pointer (DAR0)
0025₁₆	
0026₁₆	
0027₁₆	
0028₁₆	DMA0 transfer counter (TCR0)
0029₁₆	
002A₁₆	
002B₁₆	
002C₁₆	DMA0 control register (DM0CON)
002D₁₆	
002E₁₆	
002F₁₆	
0030₁₆	
0031₁₆	
0032₁₆	
0033₁₆	
0034₁₆	
0035₁₆	
0036₁₆	
0037₁₆	
0038₁₆	
0039₁₆	
003A₁₆	
003B₁₆	
003C₁₆	
003D₁₆	
003E₁₆	
003F₁₆	

0040₁₆	
0041₁₆	
0042₁₆	
0043₁₆	
0044₁₆	
0045₁₆	
0046₁₆	
0047₁₆	
0048₁₆	
0049₁₆	
004A₁₆	Bus collision detection interrupt control register (BCNIC)
004B₁₆	DMA0 interrupt control register (DM0IC)
004C₁₆	
004D₁₆	Key input interrupt control register (KUPIC)
004E₁₆	A-D conversion interrupt control register (ADIC)
004F₁₆	UART2 transmit interrupt control register (S2TIC)
0050₁₆	UART2 receive interrupt control register (S2RIC)
0051₁₆	UART0 transmit interrupt control register (S0TIC)
0052₁₆	UART0 receive interrupt control register (S0RIC)
0053₁₆	UART1 transmit interrupt control register (S1TIC)
0054₁₆	UART1 receive interrupt control register (S1RIC)
0055₁₆	Timer A0 interrupt control register (TA0IC)
0056₁₆	Timer A1 interrupt control register (TA1IC)
0057₁₆	Timer A2 interrupt control register (TA2IC)
0058₁₆	
0059₁₆	
005A₁₆	
005B₁₆	Timer B1 interrupt control register (TB1IC)
005C₁₆	Timer B2 interrupt control register (TB2IC)
005D₁₆	INT0 interrupt control register (INT0IC)
005E₁₆	INT1 interrupt control register (INT1IC)
005F₁₆	INT2 interrupt control register (INT2IC)
0060₁₆	
0061₁₆	
0062₁₆	
0063₁₆	
0064₁₆	
0065₁₆	
≈	≈
032A₁₆	
032B₁₆	
032C₁₆	
032D₁₆	
032E₁₆	
032F₁₆	
0330₁₆	
0331₁₆	
0332₁₆	
0333₁₆	
0334₁₆	
0335₁₆	
0336₁₆	
0337₁₆	
0338₁₆	
0339₁₆	
033A₁₆	
033B₁₆	
033C₁₆	
033D₁₆	
033E₁₆	
033F₁₆	

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

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

0340₁₆	
0341₁₆	
0342₁₆	
0343₁₆	
0344₁₆	
0345₁₆	
0346₁₆	
0347₁₆	
0348₁₆	
0349₁₆	
034A₁₆	
034B₁₆	
034C₁₆	
034D₁₆	
034E₁₆	
034F₁₆	
0350₁₆	
0351₁₆	
0352₁₆	
0353₁₆	
0354₁₆	
0355₁₆	
0356₁₆	
0357₁₆	
0358₁₆	
0359₁₆	
035A₁₆	
035B₁₆	
035C₁₆	
035D₁₆	
035E₁₆	
035F₁₆	Interrupt cause select register (IFSR)
0360₁₆	
0361₁₆	
0362₁₆	
0363₁₆	
0364₁₆	
0365₁₆	
0366₁₆	
0367₁₆	
0368₁₆	
0369₁₆	
036A₁₆	
036B₁₆	
036C₁₆	
036D₁₆	
036E₁₆	
036F₁₆	
0370₁₆	
0371₁₆	
0372₁₆	
0373₁₆	
0374₁₆	
0375₁₆	UART2 special mode register 3(U2SMR3)
0376₁₆	UART2 special mode register 2(U2SMR2)
0377₁₆	UART2 special mode register (U2SMR)
0378₁₆	UART2 transmit/receive mode register (U2MR)
0379₁₆	UART2 bit rate generator (U2BRG)
037A₁₆ 037B₁₆	UART2 transmit buffer register (U2TB)
037C₁₆	UART2 transmit/receive control register 0 (U2C0)
037D₁₆	UART2 transmit/receive control register 1 (U2C1)
037E₁₆ 037F₁₆	UART2 receive buffer register (U2RB)

0380₁₆	Count start flag (TABSR)
0381₁₆	Clock prescaler reset flag (CPSRF)
0382₁₆	One-shot start flag (ONSF)
0383₁₆	Trigger select register (TRGSR)
0384₁₆	Up-down flag (UDF)
0385₁₆	
0386₁₆ 0387₁₆	Timer A0 register (TA0)
0388₁₆ 0389₁₆	Timer A1 register (TA1)
038A₁₆ 038B₁₆	Timer A2 register (TA2)
038C₁₆	
038D₁₆	
038E₁₆	
038F₁₆	
0390₁₆	
0391₁₆	
0392₁₆ 0393₁₆	Timer B1 register (TB1)
0394₁₆ 0395₁₆	Timer B2 register (TB2)
0396₁₆	Timer A0 mode register (TA0MR)
0397₁₆	Timer A1 mode register (TA1MR)
0398₁₆	Timer A2 mode register (TA2MR)
0399₁₆	
039A₁₆	
039B₁₆	
039C₁₆	Timer B1 mode register (TB1MR)
039D₁₆	Timer B2 mode register (TB2MR)
039E₁₆	
039F₁₆	
03A0₁₆	UART0 transmit/receive mode register (U0MR)
03A1₁₆	UART0 bit rate generator (U0BRG)
03A2₁₆ 03A3₁₆	UART0 transmit buffer register (U0TB)
03A4₁₆	UART0 transmit/receive control register 0 (U0C0)
03A5₁₆	UART0 transmit/receive control register 1 (U0C1)
03A6₁₆ 03A7₁₆	UART0 receive buffer register (U0RB)
03A8₁₆	UART1 transmit/receive mode register (U1MR)
03A9₁₆	UART1 bit rate generator (U1BRG)
03AA₁₆ 03AB₁₆	UART1 transmit buffer register (U1TB)
03AC₁₆	UART1 transmit/receive control register 0 (U1C0)
03AD₁₆	UART1 transmit/receive control register 1 (U1C1)
03AE₁₆ 03AF₁₆	UART1 receive buffer register (U1RB)
03B0₁₆	UART transmit/receive control register 2 (UCON)
03B1₁₆	
03B2₁₆	
03B3₁₆	
03B4₁₆	
03B5₁₆	
03B6₁₆	
03B7₁₆	
03B8₁₆	DMA0 request cause select register (DM0SL)
03B9₁₆	
03BA₁₆	
03BB₁₆	
03BC₁₆	
03BD₁₆	
03BE₁₆	
03BF₁₆	

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

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

03C0₁₆	A-D register 0 (AD0)
03C1₁₆	
03C2₁₆	A-D register 1 (AD1)
03C3₁₆	
03C4₁₆	A-D register 2 (AD2)
03C5₁₆	
03C6₁₆	A-D register 3 (AD3)
03C7₁₆	
03C8₁₆	A-D register 4 (AD4)
03C9₁₆	
03CA₁₆	A-D register 5 (AD5)
03CB₁₆	
03CC₁₆	A-D register 6 (AD6)
03CD₁₆	
03CE₁₆	A-D register 7 (AD7)
03CF₁₆	
03D0₁₆	
03D1₁₆	
03D2₁₆	
03D3₁₆	
03D4₁₆	A-D control register 2 (ADCON2)
03D5₁₆	
03D6₁₆	A-D control register 0 (ADCON0)
03D7₁₆	A-D control register 1 (ADCON1)
03D8₁₆	
03D9₁₆	
03DA₁₆	
03DB₁₆	
03DC₁₆	
03DD₁₆	
03DE₁₆	
03DF₁₆	
03E0₁₆	Port P0 register (P0)
03E1₁₆	Port P1 register (P1)
03E2₁₆	Port P0 direction register (PD0)
03E3₁₆	Port P1 direction register (PD1)
03E4₁₆	Port P2 register (P2)
03E5₁₆	Port P3 register (P3)
03E6₁₆	Port P2 direction register (PD2)
03E7₁₆	Port P3 direction register (PD3)
03E8₁₆	Port P4 register (P4)
03E9₁₆	Port P5 register (P5)
03EA₁₆	Port P4 direction register (PD4)
03EB₁₆	Port P5 direction register (PD5)
03EC₁₆	Port P6 register (P6)
03ED₁₆	Port P7 register (P7)
03EE₁₆	Port P6 direction register (PD6)
03EF₁₆	Port P7 direction register (PD7)
03F0₁₆	Port P8 register (P8)
03F1₁₆	Port P9 register (P9)
03F2₁₆	Port P8 direction register (PD8)
03F3₁₆	Port P9 direction register (PD9)
03F4₁₆	Port P10 register (P10)
03F5₁₆	
03F6₁₆	Port P10 direction register (PD10)
03F7₁₆	
03F8₁₆	
03F9₁₆	
03FA₁₆	
03FB₁₆	
03FC₁₆	Pull-up control register 0 (PUR0)
03FD₁₆	Pull-up control register 1 (PUR1)
03FE₁₆	Pull-up control register 2 (PUR2)
03FF₁₆	Port control register (PCR)

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

Figure 1.6.3. Location of peripheral unit control registers (3)

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 the same effect as a hardware reset. The contents of internal RAM are preserved.

Processor Mode

(1) Types of Processor Mode

One of three processor modes can be selected: single-chip mode, memory expansion mode, and microprocessor mode. The functions of some pins, the memory map, and the access space differ according to the selected processor mode.

- **Single-chip mode**

  In single-chip mode, only internal memory space (SFR, internal RAM, and internal ROM) can be accessed. However, after the reset has been released and the operation of shifting from the microprocessor mode has started ("H" applied to the CNVSS pin), the internal ROM area cannot be accessed even if the CPU shifts to the single-chip mode.

  Ports P0 to P10 can be used as programmable I/O ports or as I/O ports for the internal peripheral functions.

- **Memory expansion mode**

  In memory expansion mode, external memory can be accessed in addition to the internal memory space (SFR, internal RAM, and internal ROM). However, after the reset has been released and the operation of shifting from the microprocessor mode has started ("H" applied to the CNVSS pin), the internal ROM area cannot be accessed even if the CPU shifts to the memory expansion mode.

  In this mode, some of the pins function as the address bus, the data bus, and as control signals. The number of pins assigned to these functions depends on the bus and register settings. (See "Bus Settings" for details.)

- **Microprocessor mode**

  In microprocessor mode, the SFR, internal RAM, and external memory space can be accessed. The internal ROM area cannot be accessed.

  In this mode, some of the pins function as the address bus, the data bus, and as control signals. The number of pins assigned to these functions depends on the bus width and register settings. (See "Bus Settings" for details.)

(2) Setting Processor Modes

The processor mode is set using the CNVSS pin and the processor mode bits (bits 1 and 0 at address 0004_{16}). Do not set the processor mode bits to "10_2".

Regardless of the level of the CNVSS pin, changing the processor mode bits selects the mode. Therefore, never change the processor mode bits when changing the contents of other bits. Do not change the processor mode bits simultaneously with other bits when changing the processor mode bits "01_2" or "11_2". Change the processor mode bits after changing the other bits. Also do not attempt to shift to or from the microprocessor mode within the program stored in the internal ROM area.

- **Applying Vss to CNVss pin**

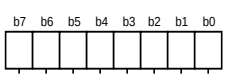
  The microcomputer begins operation in single-chip mode after being reset. Memory expansion mode is selected by writing "01_2" to the processor mode bits.

- **Applying Vcc to CNVss pin**

  The microcomputer starts to operate in microprocessor mode after being reset.

Figure 1.7.1 shows the processor mode register 0 and 1.
Figure 1.7.2 shows the memory maps in each processor modes.

Processor mode register 0 (Note 1)

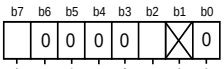


Symbol: PM0 | Address: 0004₁₆ | When reset: 00₁₆ (Note 2)

Bit symbol	Bit name	Function	R	W
PM00	Processor mode bit	b1 b0 0 0: Single-chip mode 0 1: Memory expansion mode 1 0: Must not be set 1 1: Microprocessor mode	○	○
PM01			○	○
PM02	R/W mode select bit	0 : $\overline{RD}$,$\overline{BHE}$,$\overline{WR}$ 1 : RD,WRH,WRL	○	○
PM03	Software reset bit	The device is reset when this bit is set to "1". The value of this bit is "0" when read.	○	○
PM04	Multiplexed bus space select bit	b5 b4 0 0 : Multiplexed bus is not used 0 1 : Allocated to CS2 space 1 0 : Allocated to CS1 space 1 1 : Allocated to entire space (Note4)	○	○
PM05			○	○
PM06	Port $P4_0$ to $P4_3$ function select bit (Note 3)	0 : Address output 1 : Port function (Address is not output)	○	○
PM07	BCLK output disable bit	0 : BCLK is output 1 : BCLK is not output (Pin is left floating)	○	○

Note 1: Set bit 1 of the protect register (address 000A₁₆) to "1" when writing new values to this register.
Note 2: If the V_{CC} voltage is applied to the CNV_{SS}, the value of this register when reset is 03₁₆. (PM00 and PM01 both are set to "1".)
Note 3: Valid in microprocessor and memory expansion modes.
Note 4: If the entire space is of multiplexed bus in memory expansion mode, choose an 8-bit width.The processor operates using the separate bus after reset is revoked, so the entire space multiplexed bus cannot be chosen in microprocessor mode.
$P3_1$ to $P3_7$ become a port if the entire space multiplexed bus is chosen, so only 256 bytes can be used in each chip select.

Processor mode register 1 (Note 1)



Symbol: PM1 | Address: 0005₁₆ | When reset: 000000X0₂

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.			–	–
PM12	Watchdog timer function select bit	0 : Interrupt 1 : Reset (Note 2)	○	○ (Note 2)
Reserved bit		Must always be set to "0"	○	○
Reserved bit		Must always be set to "0"	○	○
Reserved bit		Must always be set to "0"	○	○
Reserved bit		Must always be set to "0"	○	○
PM17	Wait bit	0 : No wait state 1 : Wait state inserted	○	○

Note 1: Set bit 1 of the protect register (address 000A₁₆) to "1" when writing new values to this register.
Note 2: This bit can only be set to "1".

Figure 1.7.1. Processor mode register 0 and 1

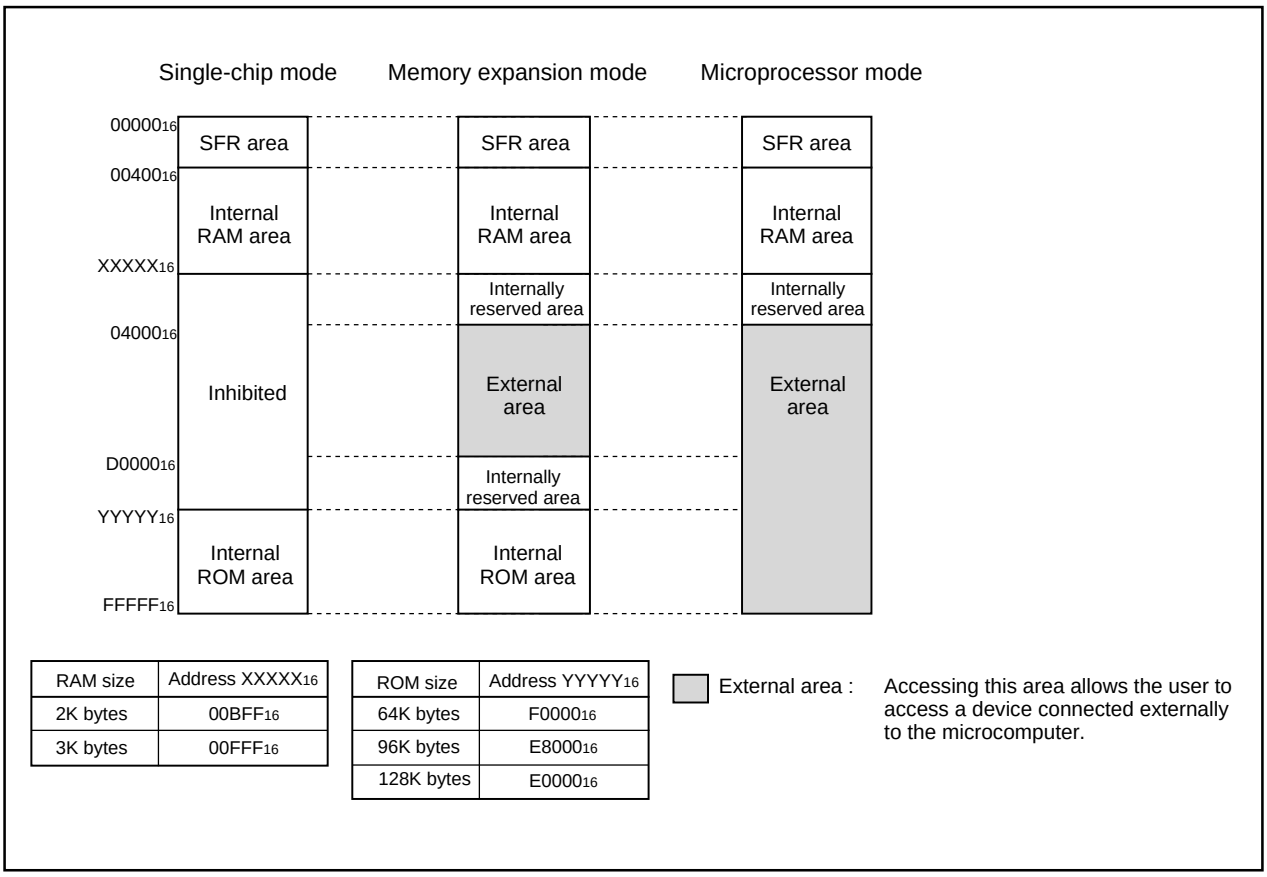


Figure 1.7.2. Memory maps in each processor modes

Bus Settings

The BYTE pin and bits 4 to 6 of the processor mode register 0 (address 0004_{16}) are used to change the bus settings. Table 1.8.1 shows the factors used to change the bus settings.

Table 1.8.1. Factors for switching bus settings

Bus setting	Switching factor
Switching external address bus width	Bit 6 of processor mode register 0
Switching external data bus width	BYTE pin
Switching between separate and multiplex bus	Bits 4 and 5 of processor mode register 0

(1) Selecting external address bus width

The address bus width for external output in the 1M bytes of address space can be set to 16 bits (64K bytes address space) or 20 bits (1M bytes address space). When bit 6 of the processor mode register 0 is set to “1”, the external address bus width is set to 16 bits, and P2 and P3 become part of the address bus. $P4_0$ to $P4_3$ can be used as programmable I/O ports. When bit 6 of processor mode register 0 is set to “0”, the external address bus width is set to 20 bits, and P2, P3, and $P4_0$ to $P4_3$ become part of the address bus.

(2) Selecting external data bus width

The external data bus width can be set to 8 or 16 bits. (Note, however, that only the separate bus can be set.) When the BYTE pin is “L”, the bus width is set to 16 bits; when “H”, it is set to 8 bits. (The internal bus width is permanently set to 16 bits.) While operating, fix the BYTE pin either to “H” or to “L”.

(3) Selecting separate/multiplex bus

The bus format can be set to multiplex or separate bus using bits 4 and 5 of the processor mode register 0.

- **Separate bus**

  In this mode, the data and address are input and output separately. The data bus can be set using the BYTE pin to be 8 or 16 bits. When the BYTE pin is “H”, the data bus is set to 8 bits and P0 functions as the data bus and P1 as a programmable I/O port. When the BYTE pin is “L”, the data bus is set to 16 bits and P0 and P1 are both used for the data bus.

  When the separate bus is used for access, a software wait can be selected.

- **Multiplex bus**

  In this mode, data and address I/O are time multiplexed. With the BYTE pin = “H”, the 8 bits from D_0 to D_7 are multiplexed with A_0 to A_7.

  With the BYTE pin = “L”, the 8 bits from D_0 to D_7 are multiplexed with A_1 to A_8. D_8 to D_{15} are not multiplexed. In this case, the external devices connected to the multiplexed bus are mapped to the microcomputer’s even addresses (every 2nd address). To access these external devices, access the even addresses as bytes.

  The ALE signal latches the address. It is output from $P5_6$.

  Before using the multiplex bus for access, be sure to insert a software wait.

  If the entire space is of multiplexed bus in memory expansion mode, choose an 8-bit width.

  The processor operates using the separate bus after reset is revoked, so the entire space multiplexed bus cannot be chosen in microprocessor mode.

  $P3_1$ to $P3_7$ become a port if the entire space multiplexed bus is chosen, so only 256 bytes can be used in each chip select.

Table 1.8.2. Pin functions for each processor mode

Processor mode	Single-chip mode	Memory expansion mode/microprocessor modes				Memory expansion mode
Multiplexed bus space select bit		"01", "10" Either CS1 or CS2 is for multiplexed bus and others are for separate bus		"00" (separate bus)		"11" (Note 1) multiplexed bus for the entire space
Data bus width BYTE pin level		8 bits "H"	16 bits "L"	8 bits "H"	16 bits "L"	8 bit "H"
$P0_0$ to $P0_7$	I/O port	Data bus	Data bus	Data bus	Data bus	I/O port
$P1_0$ to $P1_7$	I/O port	I/O port	Data bus	I/O port	Data bus	I/O port
$P2_0$	I/O port	Address bus /data bus(Note 2)	Address bus	Address bus	Address bus	Address bus /data bus
$P2_1$ to $P2_7$	I/O port	Address bus /data bus(Note 2)	Address bus /data bus(Note 2)	Address bus	Address bus	Address bus /data bus
$P3_0$	I/O port	Address bus	Address bus /data bus(Note 2)	Address bus	Address bus	A_8/D_7
$P3_1$ to $P3_7$	I/O port	Address bus	Address bus	Address bus	Address bus	I/O port
$P4_0$ to $P4_3$ Port $P4_0$ to $P4_3$ function select bit = 1	I/O port	I/O port	I/O port	I/O port	I/O port	I/O port
$P4_0$ to $P4_3$ Port $P4_0$ to $P4_3$ function select bit = 0	I/O port	Address bus	Address bus	Address bus	Address bus	I/O port
$P4_4$ to $P4_7$	I/O port	$\overline{CS}$ (chip select) or programmable I/O port (For details, refer to "Bus control")				
$P5_0$ to $P5_3$	I/O port	Outputs $\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$, and BCLK or $\overline{RD}$, $\overline{BHE}$, $\overline{WR}$, and BCLK (For details, refer to "Bus control")				
$P5_4$	I/O port	$\overline{HLDA}$	$\overline{HLDA}$	$\overline{HLDA}$	$\overline{HLDA}$	$\overline{HLDA}$
$P5_5$	I/O port	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$
$P5_6$	I/O port	ALE	ALE	ALE	ALE	ALE
$P5_7$	I/O port	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$

Note 1: If the entire space is of multiplexed bus in memory expansion mode, choose an 8-bit width.
The processor operates using the separate bus after reset is revoked, so the entire space multiplexed bus cannot be chosen in microprocessor mode.
$P3_1$ to $P3_7$ become a port if the entire space multiplexed bus is chosen, so only 256 bytes can be used in each chip select.
Note 2: Address bus when in separate bus mode.

Bus Control

The following explains the signals required for accessing external devices and software waits. The signals required for accessing the external devices are valid when the processor mode is set to memory expansion mode and microprocessor mode. The software waits are valid in all processor modes.

(1) Address bus/data bus

The address bus consists of the 20 pins A_0 to A_{19} for accessing the 1M bytes of address space.

The data bus consists of the pins for data I/O. When the BYTE pin is “H”, the 8 ports D_0 to D_7 function as the data bus. When BYTE is “L”, the 16 ports D_0 to D_{15} function as the data bus.

When a change is made from single-chip mode to memory expansion mode, the value of the address bus is undefined until external memory is accessed.

(2) Chip select signal

The chip select signal is output using the same pins as $P4_4$ to $P4_7$. Bits 0 to 3 of the chip select control register (address 0008_{16}) set each pin to function as a port or to output the chip select signal. The chip select control register is valid in memory expansion mode and microprocessor mode. In single-chip mode, $P4_4$ to $P4_7$ function as programmable I/O ports regardless of the value in the chip select control register.

In microprocessor mode, only $\overline{CS0}$ outputs the chip select signal after the reset state has been cancelled. $\overline{CS1}$ to $\overline{CS3}$ function as input ports. Figure 1.9.1 shows the chip select control register.

The chip select signal can be used to split the external area into as many as four blocks. Tables 1.9.1 shows the external memory areas specified using the chip select signal.

Table 1.9.1. External areas specified by the chip select signals

Processor mode	Chip select signal			
	$\overline{CS0}$	$\overline{CS1}$	$\overline{CS2}$	$\overline{CS3}$
Memory expansion mode	30000_{16} to $CFFFF_{16}$ (640K bytes)	28000_{16} to $2FFFF_{16}$ (32K bytes)	08000_{16} to $27FFF_{16}$ (128K bytes)	04000_{16} to $07FFF_{16}$ (16K bytes)
Microprocessor mode	30000_{16} to $FFFFF_{16}$ (832K bytes)			

Chip select control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: CSR
Address: 0008$_{16}$
When reset: 01$_{16}$

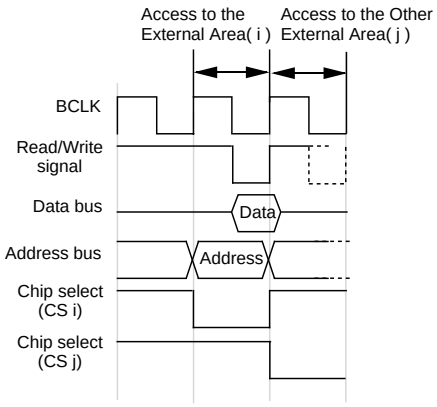
Bit symbol	Bit name	Function	R	W
CS0	$\overline{CS0}$ output enable bit	0 : Chip select output disabled (Normal port pin) 1 : Chip select output enabled	○	○
CS1	$\overline{CS1}$ output enable bit		○	○
CS2	$\overline{CS2}$ output enable bit		○	○
CS3	$\overline{CS3}$ output enable bit		○	○
CS0W	$\overline{CS0}$ wait bit	0 : Wait state inserted 1 : No wait state	○	○
CS1W	$\overline{CS1}$ wait bit		○	○
CS2W	$\overline{CS2}$ wait bit		○	○
CS3W	$\overline{CS3}$ wait bit		○	○

Figure 1.9.1. Chip select control register

The timing of the chip select signal changing to “L”(active) is synchronized with the address bus. But the timing of the chip select signal changing to “H” depends on the area which will be accessed in the next cycle. Figure 1.9.2 shows the output example of the address bus and chip select signal.

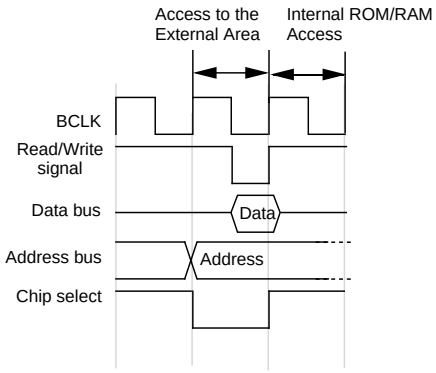
Example 1) After access the external area, both the address signal and the chip select signal change concurrently in the next cycle.

In this example, after access to the external area(i), an access to the area indicated by the other chip select signal(j) will occur in the next cycle. In this case, both the address bus and the chip select signal change between the two cycles.



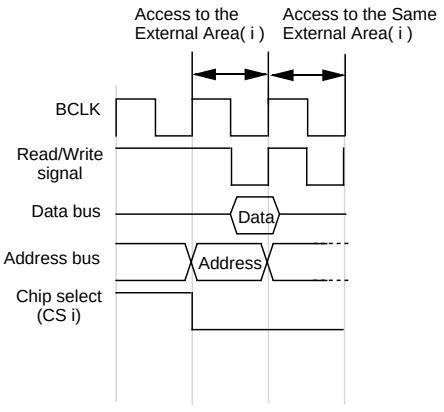
Example 2) After access the external area, only the chip select signal changes in the next cycle (the address bus does not change).

In this example, an access to the internal ROM or the internal RAM in the next cycle will occur, after access to the external area. In this case, the chip select signal changes between the two cycles, but the address does not change.



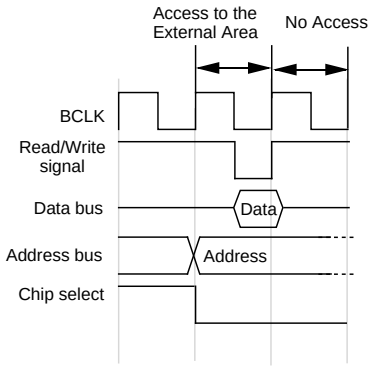
Example 3) After access the external area, only the address bus changes in the next cycle (the chip select signal does not change).

In this example, after access to the external area(i), an access to the area indicated by the same chip select signal(i) will occur in the next cycle. In this case, the address bus changes between the two cycles, but the chip select signal does not change.



Example 4) After access the external area, either the address signal and the chip select signal do not change in the next cycle.

In this example, any access to any area does not occur in the next cycle (either instruction prefetch does not occur). In this case,either the address bus and chip select signal do not change between the two cycles.



Note : These examples show the address bus and chip select signal within the successive two cycles.
According to the combination of these examples, the chip select can be elongated to over 2cycles.

Figure 1.9.2. Output Examples about Address Bus and Chip Select Signal (Separated Bus without Wait)

(3) Read/write signals

With a 16-bit data bus (BYTE pin ="L"), bit 2 of the processor mode register 0 (address 0004_{16}) select the combinations of $\overline{RD}$, $\overline{BHE}$, and $\overline{WR}$ signals or $\overline{RD}$, $\overline{WRL}$, and $\overline{WRH}$ signals. With an 8-bit data bus (BYTE pin = "H"), use the combination of $\overline{RD}$, $\overline{WR}$, and $\overline{BHE}$ signals. (Set bit 2 of the processor mode register 0 (address 0004_{16}) to "0".) Tables 1.9.3 and 1.9.4 show the operation of these signals.
After a reset has been cancelled, the combination of $\overline{RD}$, $\overline{WR}$, and $\overline{BHE}$ signals is automatically selected. When switching to the $\overline{RD}$, $\overline{WRL}$, and $\overline{WRH}$ combination, do not write to external memory until bit 2 of the processor mode register 0 (address 0004_{16}) has been set (Note).

Note: Before attempting to change the contents of the processor mode register 0, set bit 1 of the protect register (address $000A_{16}$) to "1".

Table 1.9.3. Operation of $\overline{RD}$, $\overline{WRL}$, and $\overline{WRH}$ signals

Data bus width	$\overline{RD}$	$\overline{WRL}$	$\overline{WRH}$	Status of external data bus
16-bit (BYTE = "L")	L	H	H	Read data
	H	L	H	Write 1 byte of data to even address
	H	H	L	Write 1 byte of data to odd address
	H	L	L	Write data to both even and odd addresses

Table 1.9.4. Operation of $\overline{RD}$, $\overline{WR}$, and $\overline{BHE}$ signals

Data bus width	$\overline{RD}$	$\overline{WR}$	$\overline{BHE}$	A0	Status of external data bus
16-bit (BYTE = "L")	H	L	L	H	Write 1 byte of data to odd address
	L	H	L	H	Read 1 byte of data from odd address
	H	L	H	L	Write 1 byte of data to even address
	L	H	H	L	Read 1 byte of data from even address
	H	L	L	L	Write data to both even and odd addresses
	L	H	L	L	Read data from both even and odd addresses
8-bit (BYTE = "H")	H	L	Not used	H / L	Write 1 byte of data
	L	H	Not used	H / L	Read 1 byte of data

(4) ALE signal

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

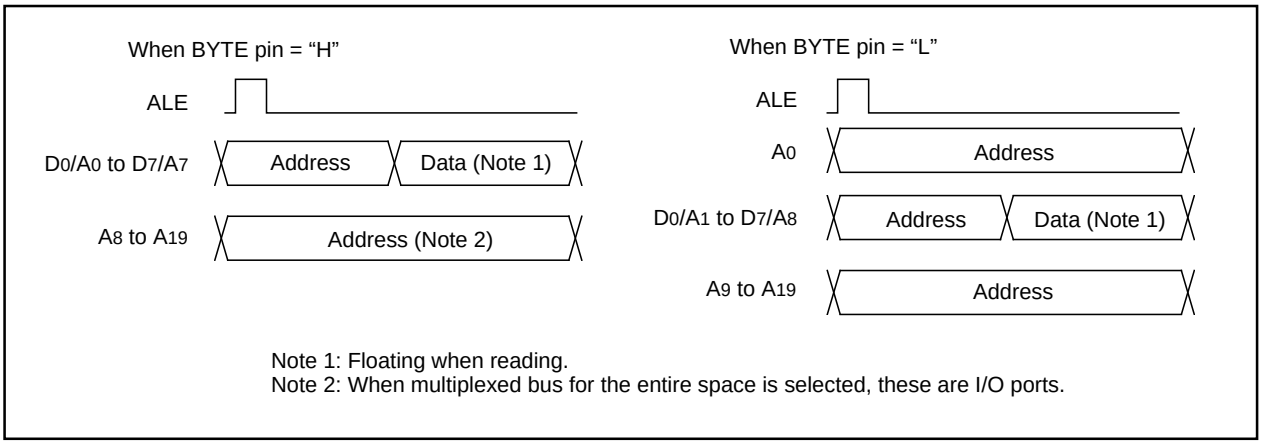


Figure 1.9.3. ALE signal and address/data bus

(5) The $\overline{RDY}$ signal

$\overline{RDY}$ is a signal that facilitates access to an external device that requires long access time. As shown in Figure 1.9.4, if an “L” is being input to the $\overline{RDY}$ at the BCLK falling edge, the bus turns to the wait state. If an “H” is being input to the $\overline{RDY}$ pin at the BCLK falling edge, the bus cancels the wait state. Table 1.9.5 shows the state of the microcomputer with the bus in the wait state, and Figure 1.9.4 shows an example in which the $\overline{RD}$ signal is prolonged by the $\overline{RDY}$ signal.

The $\overline{RDY}$ signal is valid when accessing the external area during the bus cycle in which bits 4 to 7 of the chip select control register (address 0008_{16}) are set to “0”. The $\overline{RDY}$ signal is invalid when setting “1” to all bits 4 to 7 of the chip select control register (address 0008_{16}), but the $\overline{RDY}$ pin should be treated as properly as in non-using.

Table 1.9.5. Microcomputer status in wait state (Note)

Item	Status
Oscillation	On
R/$\overline{W}$ signal, address bus, data bus, $\overline{CS}$ ALE signal, $\overline{HLDA}$, programmable I/O ports	Maintain status when $\overline{RDY}$ signal received
Internal peripheral circuits	On

Note: The $\overline{RDY}$ signal cannot be received immediately prior to a software wait.

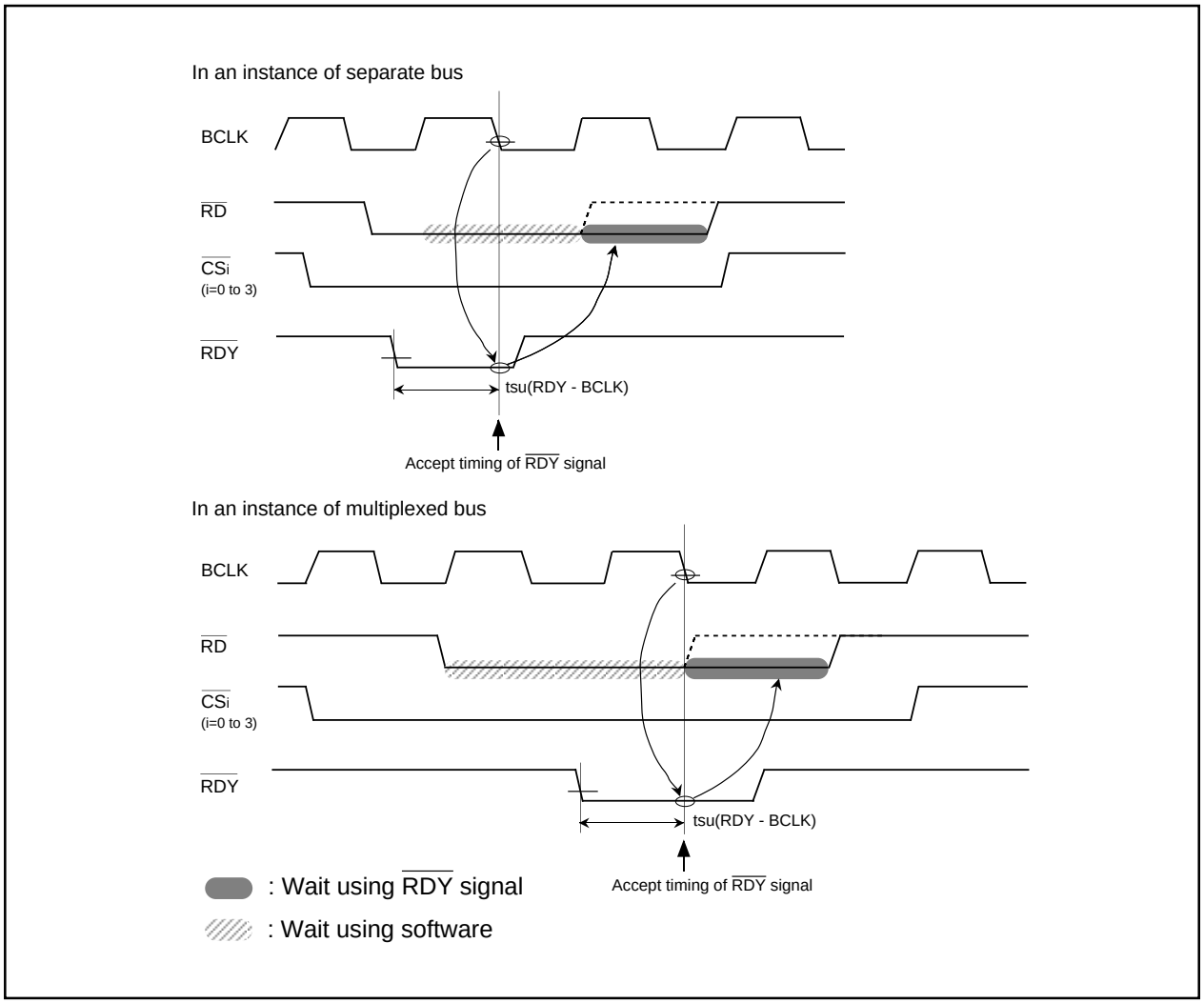


Figure 1.9.4. Example of $\overline{RD}$ signal extended by $\overline{RDY}$ signal

(6) Hold signal

The hold signal is used to transfer the bus privileges from the CPU to the external circuits. Inputting “L” to the $\overline{\text{HOLD}}$ pin places the microcomputer in the hold state at the end of the current bus access. This status is maintained and “L” is output from the $\overline{\text{HLDA}}$ pin as long as “L” is input to the $\overline{\text{HOLD}}$ pin. Table 1.9.6 shows the microcomputer status in the hold state.

Bus-using priorities are given to $\overline{\text{HOLD}}$, DMAC, and CPU in order of decreasing precedence.

$\overline{\text{HOLD}}$ > DMAC > CPU

Figure 1.9.5. Bus-using priorities

Table 1.9.6. Microcomputer status in hold state

Item		Status
Oscillation		ON
R/$\overline{\text{W}}$ signal, address bus, data bus, $\overline{\text{CS}}$, $\overline{\text{BHE}}$		Floating
Programmable I/O ports	P0, P1, P2, P3, P4, P5	Floating
	P6, P7, P8, P9, P10	Maintains status when hold signal is received
$\overline{\text{HLDA}}$		Output “L”
Internal peripheral circuits		ON (but watchdog timer stops)
ALE signal		Undefined

(7) External bus status when the internal area is accessed

Table 1.9.7 shows the external bus status when the internal area is accessed.

Table 1.9.7. External bus status when the internal area is accessed

Item		SFR accessed	Internal ROM/RAM accessed
Address bus		Address output	Maintain status before accessed address of external area
Data bus	When read	Floating	Floating
	When write	Output data	Undefined
$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$		$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$ output	Output “H”
$\overline{\text{BHE}}$		$\overline{\text{BHE}}$ output	Maintain status before accessed status of external area
$\overline{\text{CS}}$		Output “H”	Output “H”
ALE		Output “L”	Output “L”

(8) BCLK output

The user can choose the BCLK output by use of bit 7 of processor mode register 0 (0004_{16}) (Note). When set to “1”, the output floating.

Note: Before attempting to change the contents of the processor mode register 0, set bit 1 of the protect register (address $000A_{16}$) to “1”.

(9) Software wait

A software wait can be inserted by setting the wait bit (bit 7) of the processor mode register 1 (address 0005_{16}) (Note) and bits 4 to 7 of the chip select control register (address 0008_{16}).

A software wait is inserted in the internal ROM/RAM area and in the external memory area by setting the wait bit of the processor mode register 1. When set to “0”, each bus cycle is executed in one BCLK cycle. When set to “1”, each bus cycle is executed in two or three BCLK cycles. After the microcomputer has been reset, this bit defaults to “0”. When set to “1”, a wait is applied to all memory areas (two or three BCLK cycles), regardless of the contents of bits 4 to 7 of the chip select control register. Set this bit after referring to the recommended operating conditions (main clock input oscillation frequency) of the electric characteristics. However, when the user is using the $\overline{RDY}$ signal, the relevant bit in the chip select control register’s bits 4 to 7 must be set to “0”.

When the wait bit of the processor mode register 1 is “0”, software waits can be set independently for each of the 4 areas selected using the chip select signal. Bits 4 to 7 of the chip select control register correspond to chip selects $\overline{CS0}$ to $\overline{CS3}$. When one of these bits is set to “1”, the bus cycle is executed in one BCLK cycle. When set to “0”, the bus cycle is executed in two or three BCLK cycles. These bits default to “0” after the microcomputer has been reset.

The SFR area is always accessed in two BCLK cycles regardless of the setting of these control bits. Also, insert a software wait if using the multiplex bus to access the external memory area.

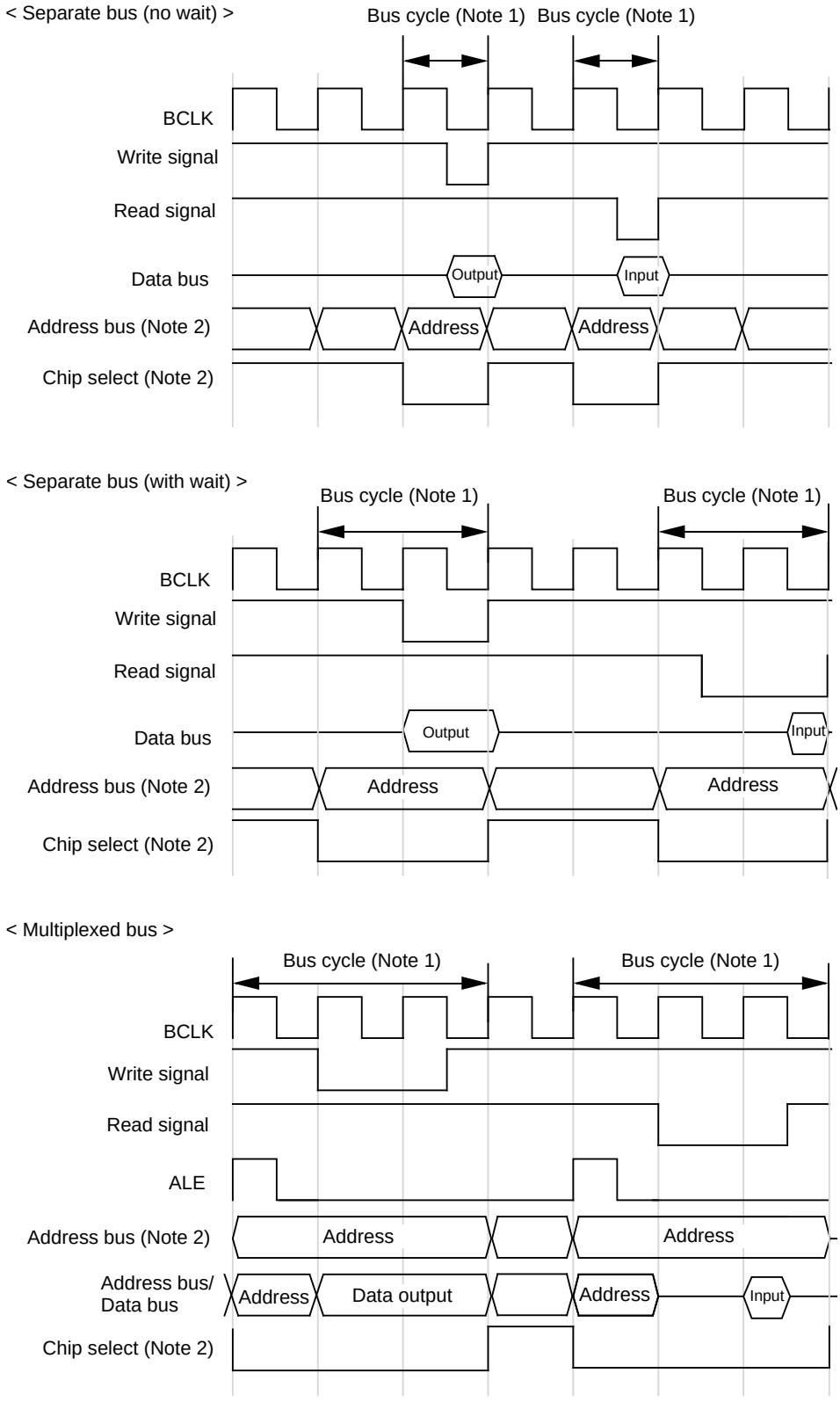
Table 1.9.8 shows the software wait and bus cycles. Figure 1.9.6 shows example of bus timing when using software waits.

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

Table 1.9.8. Software waits and bus cycles

Area	Bus status	Wait bit	Bits 4 to 7 of chip select control register	Bus cycle
SFR	———	Invalid	Invalid	2 BCLK cycles
Internal ROM/RAM	———	0	Invalid	1 BCLK cycle
	———	1	Invalid	2 BCLK cycles
External memory area	Separate bus	0	1	1 BCLK cycle
	Separate bus	0	0	2 BCLK cycles
	Separate bus	1	0 (Note)	2 BCLK cycles
	Multiplex bus	0	0	3 BCLK cycles
	Multiplex bus	1	0 (Note)	3 BCLK cycles

Note: When using the $\overline{RDY}$ signal, always set to “0”.



Note 1: These example timing charts indicate bus cycle length.
After this bus cycle sometimes come read and write cycles in succession.
Note 2: The address bus and chip select may be extended depending on the CPU status such as that of the instruction queue buffer.

Figure 1.9.6. Typical bus timings using software wait

Clock Generating Circuit

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

Table 1.10.1. Main clock and sub-clock generating circuits

	Main clock generating circuit	Sub-clock generating circuit
Use of clock	• CPU's operating clock source • Internal peripheral units' operating clock source	• CPU's operating clock source • Timer A/B's count clock source
Usable oscillator	Ceramic or crystal oscillator	Crystal oscillator
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT
Oscillation stop/restart function	Available	Available
Oscillator status immediately after reset	Oscillating	Stopped
Other	Externally derived clock can be input	

Example of oscillator circuit

Figure 1.10.1 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 1.10.2 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 1.10.1 and 1.10.2 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 1.10.1. 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. Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip, insert a feedback resistor between XCIN and XCOUT following the instruction.

Figure 1.10.2. Examples of sub-clock

Clock Control

Figure 1.10.3 shows the block diagram of the clock generating circuit.

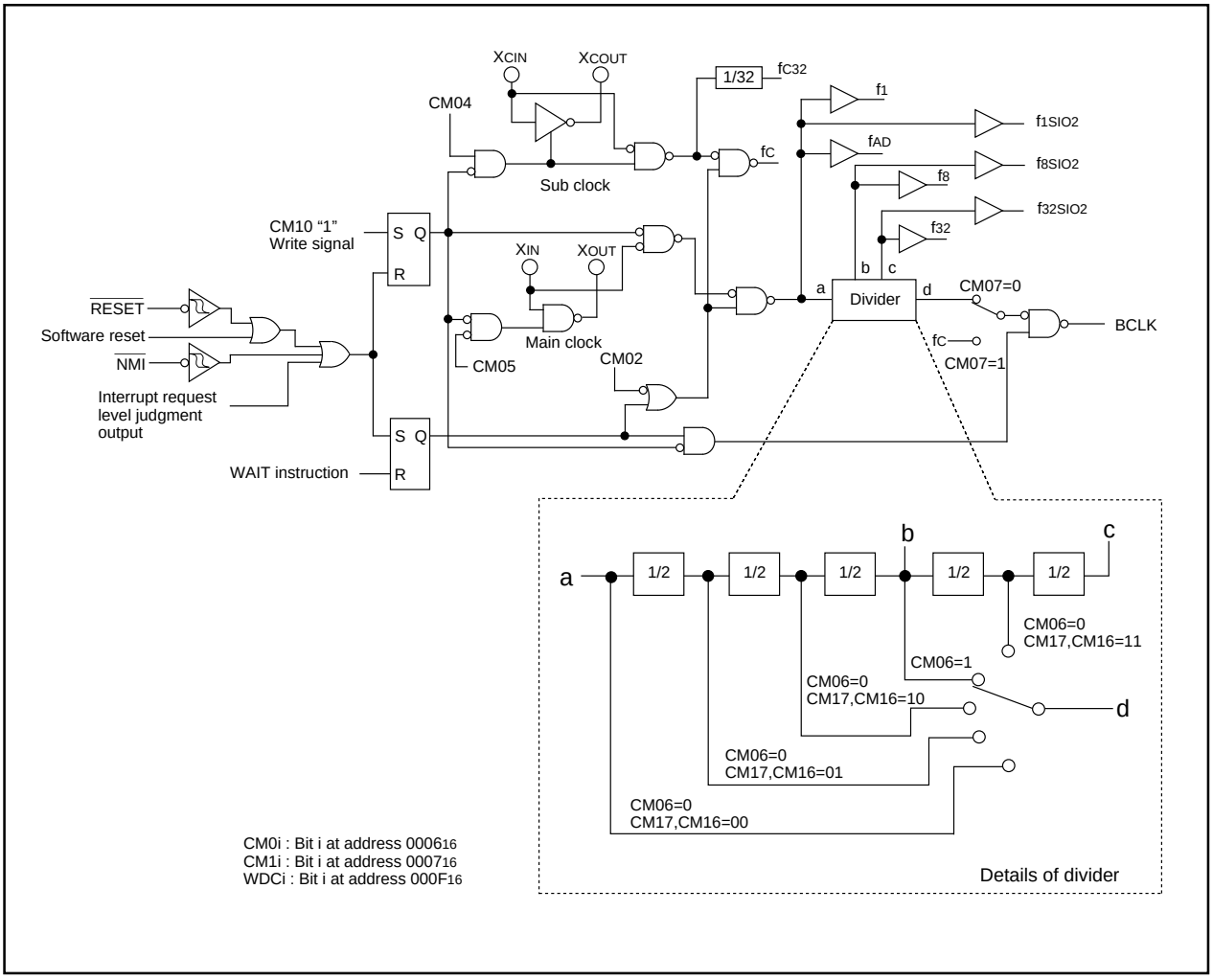


Figure 1.10.3. Clock generating circuit

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

(1) Main clock

The main clock is generated by the main clock oscillation circuit. After a reset, the clock is divided by 8 to the BCLK. The clock can be stopped using the main clock stop bit (bit 5 at address 0006_{16}). Stopping the clock, after switching the operating clock source of CPU to the sub-clock, reduces the power dissipation. After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the main clock oscillation circuit can be reduced using the XIN-XOUT drive capacity select bit (bit 5 at address 0007_{16}). Reducing the drive capacity of the main clock oscillation circuit reduces the power dissipation. This bit changes to “1” when shifting from high-speed/medium-speed mode to stop mode, shifting to low power dissipation mode and at a reset. When shifting from high-speed/medium-speed mode to low-speed mode, the value before high-speed/medium-speed 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 the BCLK by using the system clock select bit (bit 7 at address 0006_{16}). However, be sure that the sub-clock oscillation has fully stabilized before switching.

After the oscillation of the sub-clock oscillation circuit has stabilized, the drive capacity of the sub-clock oscillation circuit can be reduced using the XCIN-XCOUT drive capacity select bit (bit 3 at address 0006_{16}). Reducing the drive capacity of the sub-clock oscillation circuit reduces the power dissipation. This bit changes to “1” when the port XC select bit (bit 4 at address 0006_{16}) is set to “0” , shifting to stop mode and at a reset.

When the XCIN/XCOUT is used, set ports $P8_6$ and $P8_7$ as the input ports without pull-up.

(3) BCLK

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

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

(4) Peripheral function clock(f_1, f_8, f_{32}, f_{1SIO2}, f_{8SIO2},f_{32SIO2},f_{AD})

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

(5) f_{C32}

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

(6) fC

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

Figure 1.10.4 shows the system clock control registers 0 and 1.

System clock control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

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

Bit symbol	Bit name	Function	R	W
CM00	Clock output function select bit (Valid only in single-chip mode)	b1 b0 0 0 : I/O port P57 0 1 : fC output 1 0 : f8 output 1 1 : f32 output	O	O
CM01			O	O
CM02	WAIT peripheral function clock stop bit	0 : Do not stop peripheral function clock in wait mode 1 : Stop peripheral function clock in wait mode (Note 8)	O	O
CM03	XCIN-XCOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM04	Port XC select bit (Note 10)	0 : I/O port 1 : XCIN-XCOUT generation (Note 9)	O	O
CM05	Main clock (XIN-XOUT) stop bit (Note 3, 4, 5)	0 : On 1 : Off	O	O
CM06	Main clock division select bit 0 (Note 7)	0 : CM16 and CM17 valid 1 : Division by 8 mode	O	O
CM07	System clock select bit (Note 6)	0 : XIN, XOUT 1 : XCIN, XCOUT	O	O

Note 1: Set bit 0 of the protect register (address 000A16) to “1” before writing to this register.
Note 2: Changes to “1” when the port XC select bit (CM04) is set to “0”, shiffing to stop mode and at a reset.
Note 3: When entering low power dissipation mode, main clock stops by using this bit. To stop the main clock, when the sub clock oscillation is stable, set system clock select bit (CM07) to “1” before setting this bit to “1”. The main clock division select bit 0 (CM06) and the XIN-XOUT drive capacity select bit (CM15) change to “1” when this bit is set to “1”.
Note 4: When inputting external clock, only clock oscillation buffer is stopped and clock input is acceptable.
Note 5: If this bit is set to “1”, XOUT turns “H”. The built-in feedback resistor remains being connected, so XIN turns pulled up to XOUT (“H”) via the feedback resistor.
Note 6: Set port XC select bit (CM04) to “1” and stabilize the sub-clock oscillating before setting this bit from “0” to “1”. Do not write to both bits at the same time. And also, set the main clock stop bit (CM05) to “0” and stabilize the main clock oscillating before setting this bit from “1” to “0”.
Note 7: This bit changes to “1” when shifting from high-speed/medium-speed mode to stop mode, shifting to low power dissipation mode and at a reset. When shifting from high-speed/medium-speed mode to low-speed mode, the value before high-speed/medium-speed mode is retained.
Note 8: fC32 is not included. Do not set to “1” when using low-speed or low power dissipation mode.
Note 9: When the XCIN/XCOUT is used, set ports P86 and P87 as the input ports without pull-up.
Note10: The XCIN-XCOUT drive capacity select bit changes to “1” when this bit is set to “0”.

System clock control register 1 (Note 1)

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

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

Bit symbol	Bit name	Function	R	W
CM10	All clock stop control bit (Note4)	0 : Clock on 1 : All clocks off (stop mode)	O	O
Reserved bit		Must always be set to “0”	O	O
Reserved bit		Must always be set to “0”	O	O
Reserved bit		Must always be set to “0”	O	O
Reserved bit		Must always be set to “0”	O	O
CM15	XIN-XOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM16	Main clock division select bit 1 (Note 3)	b7 b6 0 0 : No division mode 0 1 : Division by 2 mode 1 0 : Division by 4 mode 1 1 : Division by 16 mode	O	O
CM17				

Note 1: Set bit 0 of the protect register (address 000A16) to “1” before writing to this register.
Note 2: This bit changes to “1” when shifting from high-speed/medium-speed mode to stop mode, shifting to low power dissipation mode and at a reset. When shifting from high-speed/medium-speed mode to low-speed mode, the value before high-speed/medium-speed mode is retained.
Note 3: Can be selected when bit 6 of the system clock control register 0 (address 000616) is “0”. If “1”, division mode is fixed at 8.
Note 4: If this bit is set to “1”, XOUT turns “H”, and the built-in feedback resistor is cut off. XCIN and XCOUT turn high-impedance state.

Figure 1.10.4. Clock control registers 0 and 1

Clock Output

In single-chip mode, the clock output function select bits (bits 0 and 1 at address 0006_{16}) enable f_8, f_{32}, or fc to be output from the P57/CLKOUT pin. When the WAIT peripheral function clock stop bit (bit 2 at address 0006_{16}) is set to “1”, the output of f_8 and f_{32} stops when a WAIT instruction is executed.

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 , BCLK, f_1 to f_{32}, f_{1SIO2} to f_{32SIO2}, f_C, f_{C32}, and f_{AD} stops in stop mode, peripheral functions such as the A-D converter and watchdog timer do not function. However, timer A and timer B operate provided that the event counter mode is set to an external pulse, and UARTi(i = 0 to 2) functions provided an external clock is selected. Table 1.10.2 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. If returning by an interrupt, that interrupt routine is executed. If only a hardware reset or an $\overline{NMI}$ interrupt is used to cancel stop mode, change the priority level of all interrupt to 0, then shift to stop mode.

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, shifting to low power dissipation mode and at reset. When shifting from high-speed/medium-speed mode to low-speed mode, the value before high-speed/medium-speed mode is retained.

Table 1.10.2. Port status during stop mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, $\overline{CS0}$ to $\overline{CS3}$, $\overline{BHE}$		Retains status before stop mode	
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		“H”	
$\overline{HLDA}$, BCLK		“H”	
ALE		“H”	
Port		Retains status before stop mode	Retains status before stop mode
CLKOUT	When fc selected	Valid only in single-chip mode	“H”
	When f_8, f_{32} selected	Valid only in single-chip mode	Retains status before stop mode

Wait Mode

When a WAIT instruction is executed, the BCLK stops and the microcomputer enters the wait mode. In this mode, oscillation continues but the BCLK and watchdog timer stop. Writing “1” to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. However, peripheral function clock f_{C32} does not stop so that the peripherals using f_{C32} do not contribute to the power saving. When the MCU running in low-speed or low power dissipation mode, do not enter WAIT mode with this bit set to “1”. Table 1.10.3 shows the status of the ports in wait mode.

Wait mode is cancelled by a hardware reset or an interrupt. If an interrupt is used to cancel wait mode, 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. If returning by an interrupt, the clock in which the WAIT instruction executed is set to BCLK by the microcomputer, and the action is resumed from the interrupt routine. If only a hardware reset or an $\overline{NMI}$ interrupt is used to cancel wait mode, change the priority level of all interrupt to 0,then shift to wait mode.

Table 1.10.3. Port status during wait mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, $\overline{CS0}$ to $\overline{CS3}$, $\overline{BHE}$		Retains status before wait mode	
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		“H”	
$\overline{HLDA}$,BCLK		“H”	
ALE		“H”	
Port		Retains status before wait mode	Retains status before wait mode
CLK_{OUT}	When f_C selected	Valid only in single-chip mode	Does not stop
	When f_8, f_{32} selected	Valid only in single-chip mode	Does not stop when the WAIT peripheral function clock stop bit is “0”. When the WAIT peripheral function clock stop bit is “1”, the status immediately prior to entering wait mode is maintained.

Status Transition of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. Table 1.10.4 shows the operating modes corresponding to the settings of system clock control registers 0 and 1.

When reset, the device starts in division by 8 mode. The main clock division select bit 0 (bit 6 at address 0006_{16}) and the XIN-XOUT drive capacity select bit (bit 5 at address 0007_{16}) change to "1" when shifting from high-speed/medium-speed mode to stop mode, shifting to low power dissipation mode and at a reset. When shifting from high-speed/medium-speed mode to low-speed mode, the value before high-speed/medium-speed mode is retained. The following shows the operational modes of BCLK.

(1) Division by 2 mode

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

(2) Division by 4 mode

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

(3) Division by 8 mode

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

(4) Division by 16 mode

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

(5) No-division mode

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

(6) Low-speed mode

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

(7) Low power dissipation mode

fC is the BCLK and the main clock is stopped.

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

Table 1.10.4. Operating modes dictated by settings of system clock control registers 0 and 1

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

CM1i : Bit i of the address 0007_{16}

CM0i : Bit i of the address 0006_{16}

Power control

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

Modes

Power control is available in three modes.

(a) Normal operation mode

- **High-speed mode**
  Divide-by-1 frequency of the main clock becomes the BCLK. The CPU operates with the BCLK. Each peripheral function operates according to its assigned clock.

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

- **Low-speed mode**
  fC becomes the BCLK. The CPU operates according to the fC clock. The fC clock is supplied by the subclock. Each peripheral function operates according to its assigned clock.

- **Low power dissipation mode**
  The main clock operating in low-speed mode is stopped. The CPU operates according to the fC clock. The fC clock is supplied by the subclock. The only peripheral functions that operate are those with the subclock selected as the count source.

(b) Wait mode

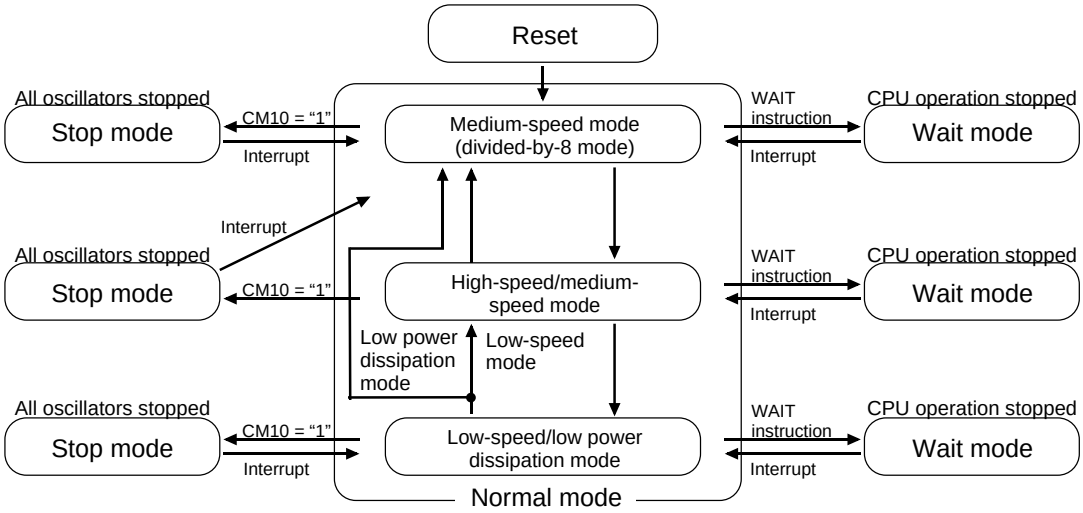
The CPU operation is stopped. The oscillators do not stop.

(c) Stop mode

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

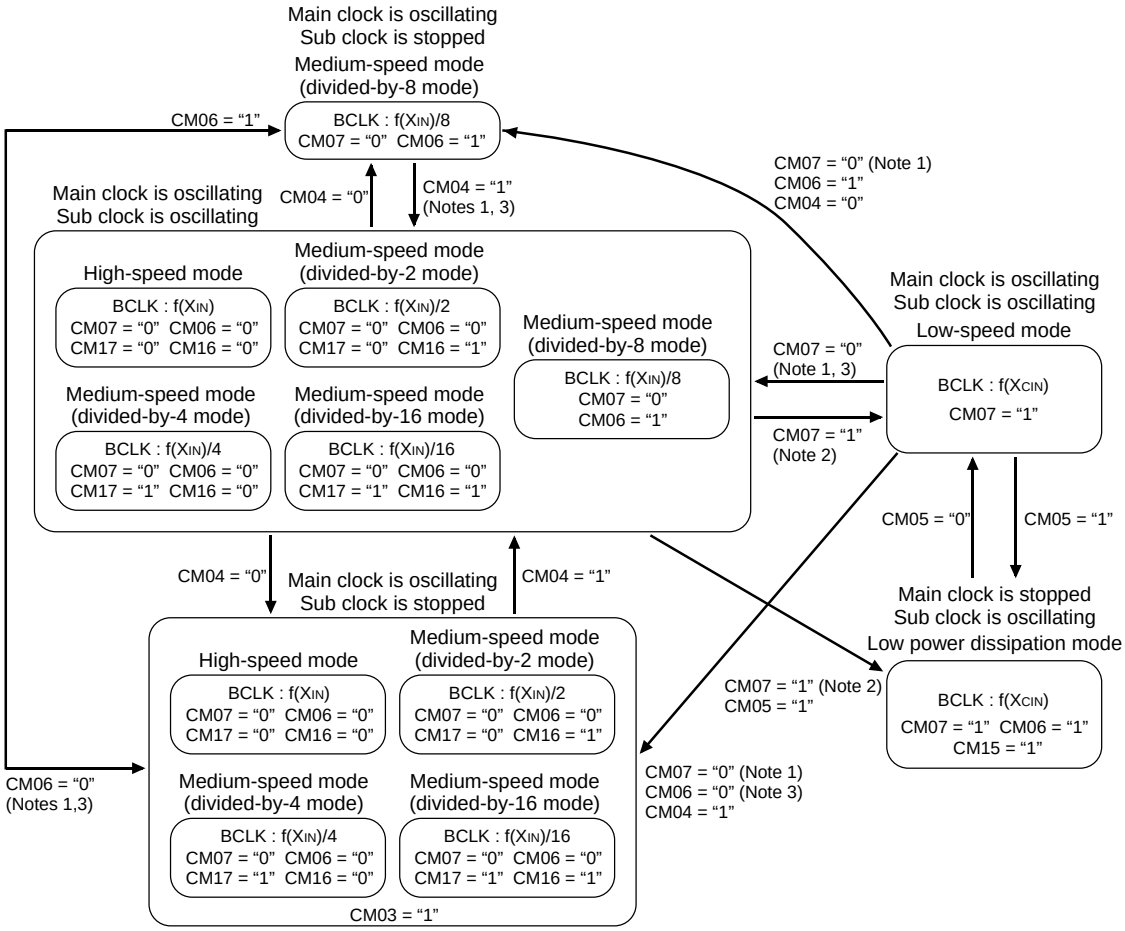
Figure 1.10.5 is the state transition diagram of the above modes.

Transition of stop mode, wait mode



(Refer to the following for the transition of normal mode.)

Transition of normal mode



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

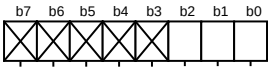
Figure 1.10.5. State transition diagram of Power control mode

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 1.10.6 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}), port P9 direction register (address $03F3_{16}$) can only be changed when the respective bit in the protect register is set to “1”. Therefore, important outputs can be allocated to port P9.

If, after “1” (write-enabled) has been written to the port P9 direction register write-enable bit (bit 2 at address $000A_{16}$), a value is written to any address, the bit automatically reverts to “0” (write-inhibited). However, the system clock control registers 0 and 1 write-enable bit (bit 0 at $000A_{16}$) and processor mode register 0 and 1 write-enable bit (bit 1 at $000A_{16}$) do not automatically return to “0” after a value has been written to an address. The program must therefore be written to return these bits to “0”.

Protect register



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

Bit symbol	Bit name	Function	R	W
PRC0	Enables writing to system clock control registers 0 and 1 (addresses 0006_{16} and 0007_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC1	Enables writing to processor mode registers 0 and 1 (addresses 0004_{16} and 0005_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC2	Enables writing to port P9 direction register (address $03F3_{16}$) (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 1.10.6. Protect register

Overview of Interrupt

Type of Interrupts

Figure 1.11.1 lists the types of interrupts.

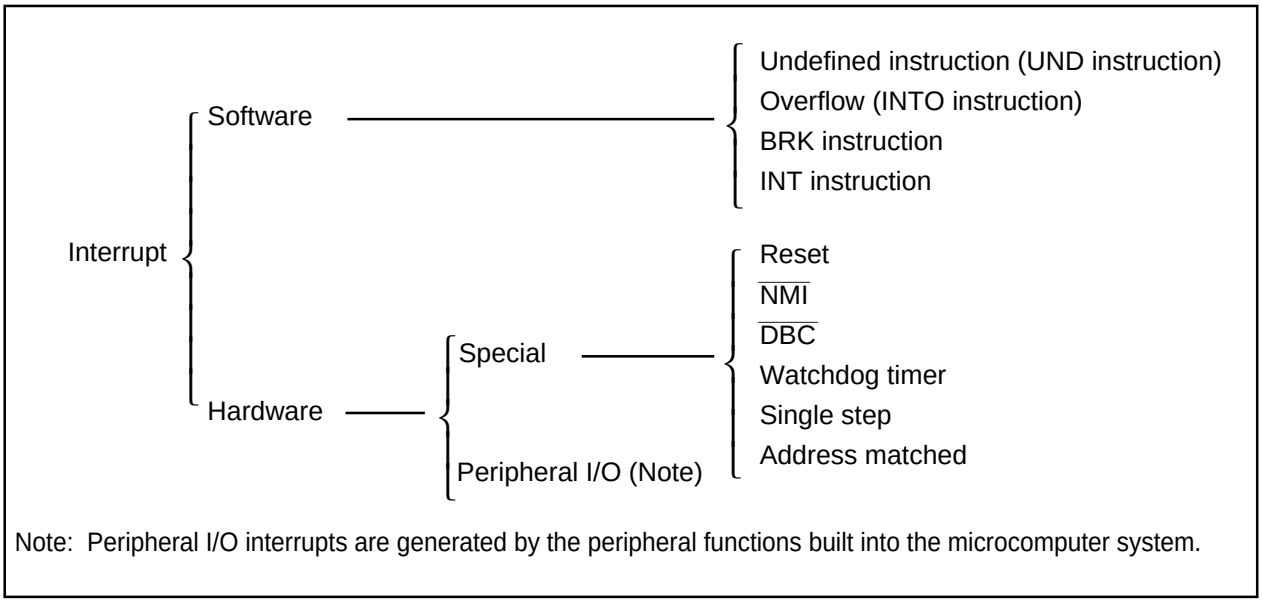


Figure 1.11.1. Classification of interrupts

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

Software Interrupts

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

- **Undefined instruction interrupt**
  An undefined instruction interrupt occurs when executing the UND instruction.
- **Overflow interrupt**
  An overflow interrupt occurs when executing the INTO instruction with the overflow flag (O flag) set to "1". The following are instructions whose O flag changes by arithmetic:
  ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB
- **BRK interrupt**
  A BRK interrupt occurs when executing the BRK instruction.
- **INT interrupt**
  An INT interrupt occurs when specifying one of software interrupt numbers 0 through 63 and executing the INT instruction. Software interrupt numbers 0 through 31 are assigned to peripheral I/O interrupts, so executing the INT instruction allows executing the same interrupt routine that a peripheral I/O interrupt does.
  The stack pointer (SP) used for the INT interrupt is dependent on which software interrupt number is involved.
  So far as software interrupt numbers 0 through 31 are concerned, the microcomputer saves the stack pointer assignment flag (U flag) when it accepts an interrupt request. If change the U flag to "0" and select the interrupt stack pointer (ISP), and then execute an interrupt sequence. When returning from the interrupt routine, the U flag is returned to the state it was before the acceptance of interrupt request. So far as software numbers 32 through 63 are concerned, the stack pointer does not make a shift.

Hardware Interrupts

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

(1) Special interrupts

Special interrupts are non-maskable interrupts.

- **Reset**

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

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

  An $\overline{\text{NMI}}$ interrupt occurs if an “L” is input to the $\overline{\text{NMI}}$ pin.

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

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

- **Watchdog timer interrupt**

  Generated by the watchdog timer. Write to the watchdog timer start register after the watchdog timer interrupt occurs (initialize watchdog timer).

- **Single-step interrupt**

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

- **Address match interrupt**

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

(2) Peripheral I/O interrupts

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

- **Bus collision detection interrupt**

  This is an interrupt that the serial I/O bus collision detection generates.

- **DMA0 interrupt**

  This is an interrupts that DMA generates.

- **Key-input interrupt**

  A key-input interrupt occurs if an “L” is input to the $\overline{\text{KI}}$ pin.

- **A-D conversion interrupt**

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

- **UART0, UART1, UART2/NACK transmission interrupt**

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

- **UART0, UART1, UART2/ACK reception interrupt**

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

- **Timer A0 interrupt through timer A2 interrupt**

  These are interrupts that timer A generates

- **Timer B1, timer B2 interrupt**

  These are interrupts that timer B generates.

- **$\overline{\text{INT0}}$ interrupt through $\overline{\text{INT2}}$ interrupt**

  An $\overline{\text{INT}}$ interrupt occurs if either a rising edge or 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 1.11.2 shows the format for specifying the address.

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

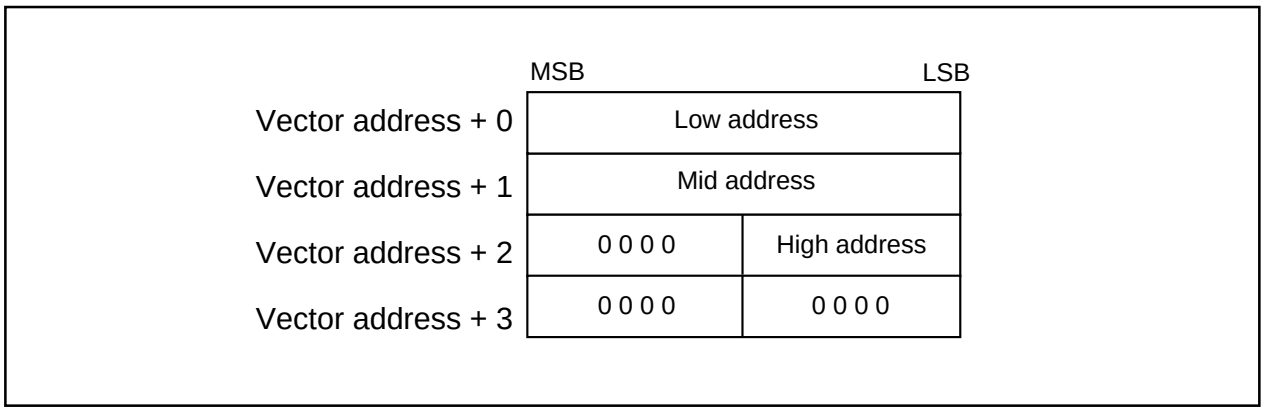


Figure 1.11.2. Format for specifying interrupt vector addresses

• 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 FFFDC16 to FFFFF16. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table 1.11.1 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

Table 1.11.1. Interrupts assigned to the fixed vector tables and addresses of vector tables

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

Note: Interrupts used for debugging purposes only.

• **Variable vector tables**

The addresses in the variable vector table can be modified, according to the user's settings. Indicate the first address using the interrupt table register (INTB). The 256-byte area subsequent to the address the INTB indicates becomes the area for the variable vector tables. One vector table comprises four bytes. Set the first address of the interrupt routine in each vector table. Table 1.11.2 shows the interrupts assigned to the variable vector tables and addresses of vector tables.

Table 1.11.2. Interrupts assigned to the variable vector tables and addresses of vector tables

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note 1)	BRK instruction	Cannot be masked I flag
Software interrupt number 10	+40 to +43 (Note 1)	Bus collision detection	
Software interrupt number 11	+44 to +47 (Note 1)	DMA0	
Software interrupt number 13	+52 to +55 (Note 1)	Key input interrupt	
Software interrupt number 14	+56 to +59 (Note 1)	A-D	
Software interrupt number 15	+60 to +63 (Note 1)	UART2 transmit/NACK (Note 2)	
Software interrupt number 16	+64 to +67 (Note 1)	UART2 receive/ACK (Note 2)	
Software interrupt number 17	+68 to +71 (Note 1)	UART0 transmit	
Software interrupt number 18	+72 to +75 (Note 1)	UART0 receive	
Software interrupt number 19	+76 to +79 (Note 1)	UART1 transmit	
Software interrupt number 20	+80 to +83 (Note 1)	UART1 receive	
Software interrupt number 21	+84 to +87 (Note 1)	Timer A0	
Software interrupt number 22	+88 to +91 (Note 1)	Timer A1	
Software interrupt number 23	+92 to +95 (Note 1)	Timer A2	
Software interrupt number 27	+108 to +111 (Note 1)	Timer B1	
Software interrupt number 28	+112 to +115 (Note 1)	Timer B2	
Software interrupt number 29	+116 to +119 (Note 1)	$\overline{\text{INT0}}$	
Software interrupt number 30	+120 to +123 (Note 1)	$\overline{\text{INT1}}$	
Software interrupt number 31	+124 to +127 (Note 1)	$\overline{\text{INT2}}$	
Software interrupt number 32 to Software interrupt number 63	+128 to +131 (Note 1) to +252 to +255 (Note 1)	Software interrupt	Cannot be masked I flag

Note 1: Address relative to address in interrupt table register (INTB).
Note 2: When IIC mode is selected, NACK and ACK interrupts are selected.

Interrupt Control

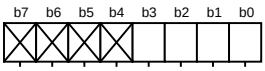
Descriptions are given here regarding how to enable or disable maskable interrupts and how to set the priority to be accepted. What is described here does not apply to non-maskable interrupts.

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

Figure 1.11.3 shows the memory map of the interrupt control registers.

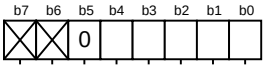
Interrupt control register (Note 2)

Symbol	Address	When reset
BCNIC	$004A_{16}$	$XXXXX000_2$
DM0IC	$004B_{16}$	$XXXXX000_2$
KUPIC	$004D_{16}$	$XXXXX000_2$
ADIC	$004E_{16}$	$XXXXX000_2$
SiTIC(i=0 to 2)	0051_{16}, 0053_{16}, $004F_{16}$	$XXXXX000_2$
SiRIC(i=0 to 2)	0052_{16}, 0054_{16}, 0050_{16}	$XXXXX000_2$
TAiIC(i=0 to 2)	0055_{16} to 0057_{16}	$XXXXX000_2$
TBiIC(i=1, 2)	$005B_{16}$, $005C_{16}$	$XXXXX000_2$



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 does not generate the interrupt request for that register. For details, see the precautions for interrupts.



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

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

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 does not generate the interrupt request for that register. For details, see the precautions for interrupts.

Figure 1.11.3. Interrupt control registers

Interrupt Enable Flag (I flag)

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

Interrupt Request Bit

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

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

Set the interrupt priority level using the interrupt priority level select bit, which is one of the component bits of the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to “0” disables the interrupt.
Table 1.11.3 shows the settings of interrupt priority levels and Table 1.11.4 shows the interrupt levels enabled, according to the contents of the IPL.

The following are conditions under which an interrupt is accepted:
· interrupt enable flag (I flag) = “1”
· interrupt request bit = “1”
· interrupt priority level > IPL

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

Table 1.11.3. Settings of interrupt priority levels

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

Table 1.11.4. Interrupt levels enabled according to the contents of the IPL

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

Rewrite the interrupt control register

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

Example 1:

```
INT_SWITCH1:
    FCLR    I             ; Disable interrupts.
    AND.B   #00h, 0055h   ; Clear TA0IC int. priority level and int. request bit.
    NOP                   ; Four NOP instructions are required when using HOLD function.
    NOP
    FSET    I             ; Enable interrupts.
```

Example 2:

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

Example 3:

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

The reason why two NOP instructions (four when using the HOLD function) 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 changing an interrupt control register in a sate of interrupts being disabled, please read the following precautions on instructions used before changing the register.

Changing a non-interrupt request bit

If an interrupt request for an interrupt control register is generated during an instruction to rewrite the register is being executed, there is a case that the interrupt request bit is not set and consequently the interrupt is ignored. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

Changing the interrupt request bit

When attempting to clear the interrupt request bit of an interrupt control register, the interrupt request bit is not cleared sometimes. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : MOV

Interrupt Sequence

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

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

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

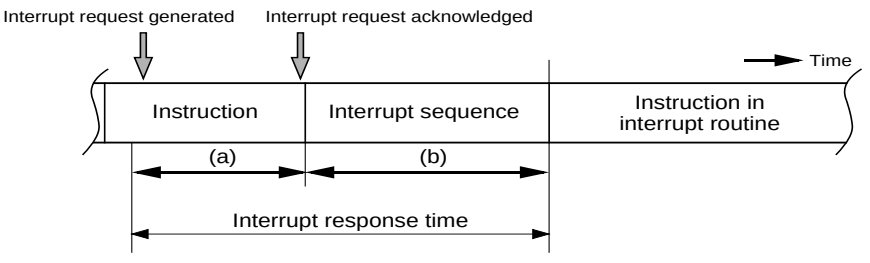
(1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 0000_{16}. After this, the corresponding interrupt request bit becomes “0”.
(2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.
(3) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer select flag (U flag) to “0” (the U flag, however does not change if the INT instruction, in software interrupt numbers 32 through 63, is executed)
(4) Saves the content of the temporary register (Note) within the CPU in the stack area.
(5) Saves the content of the program counter (PC) in the stack area.
(6) Sets the interrupt priority level of the accepted instruction in the IPL.

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

Note: This register cannot be utilized by the user.

Interrupt Response Time

'Interrupt response time' is the period between the instant an interrupt occurs and the instant the first instruction within the interrupt routine has been executed. This time comprises the period from the occurrence of an interrupt to the completion of the instruction under execution at that moment (a) and the time required for executing the interrupt sequence (b). Figure 1.11.4 shows the interrupt response time.



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

Figure 1.11.4. 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 1.11.5.

Table 1.11.5. Time required for executing the interrupt sequence

Interrupt vector address	Stack pointer (SP) value	16-Bit bus, without wait	8-Bit bus, without wait
Even	Even	18 cycles (Note 1)	20 cycles (Note 1)
Even	Odd	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Even	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Odd	20 cycles (Note 1)	20 cycles (Note 1)

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

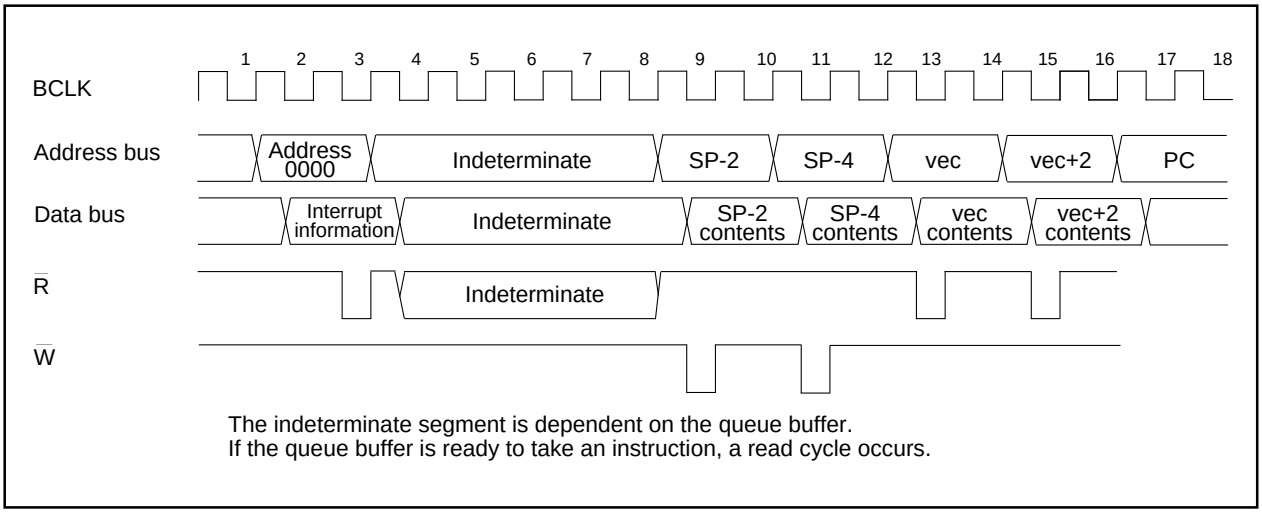


Figure 1.11.5. Time required for executing the interrupt sequence

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 1.11.6 is set in the IPL.

Table 1.11.6. Relationship between interrupts without interrupt priority levels and IPL

Interrupt sources without priority levels	Value set in the IPL
Watchdog timer, $\overline{\text{NMI}}$	7
Reset	0
Other	Not changed

Saving Registers

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

First, the processor saves the four higher-order bits of the program counter, and 4 upper-order bits and 8 lower-order bits of the FLG register, 16 bits in total, in the stack area, then saves 16 lower-order bits of the program counter. Figure 1.11.6 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).

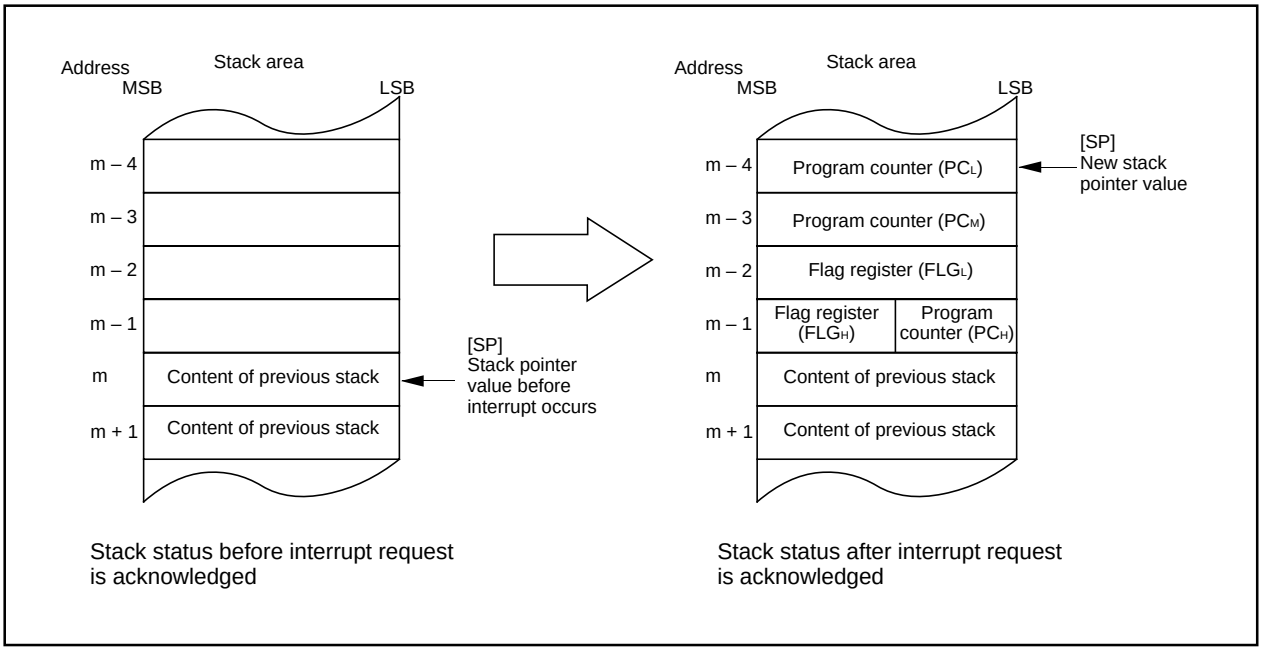


Figure 1.11.6. State of stack before and after acceptance of interrupt request

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 1.11.7 shows the operation of the saving registers.

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

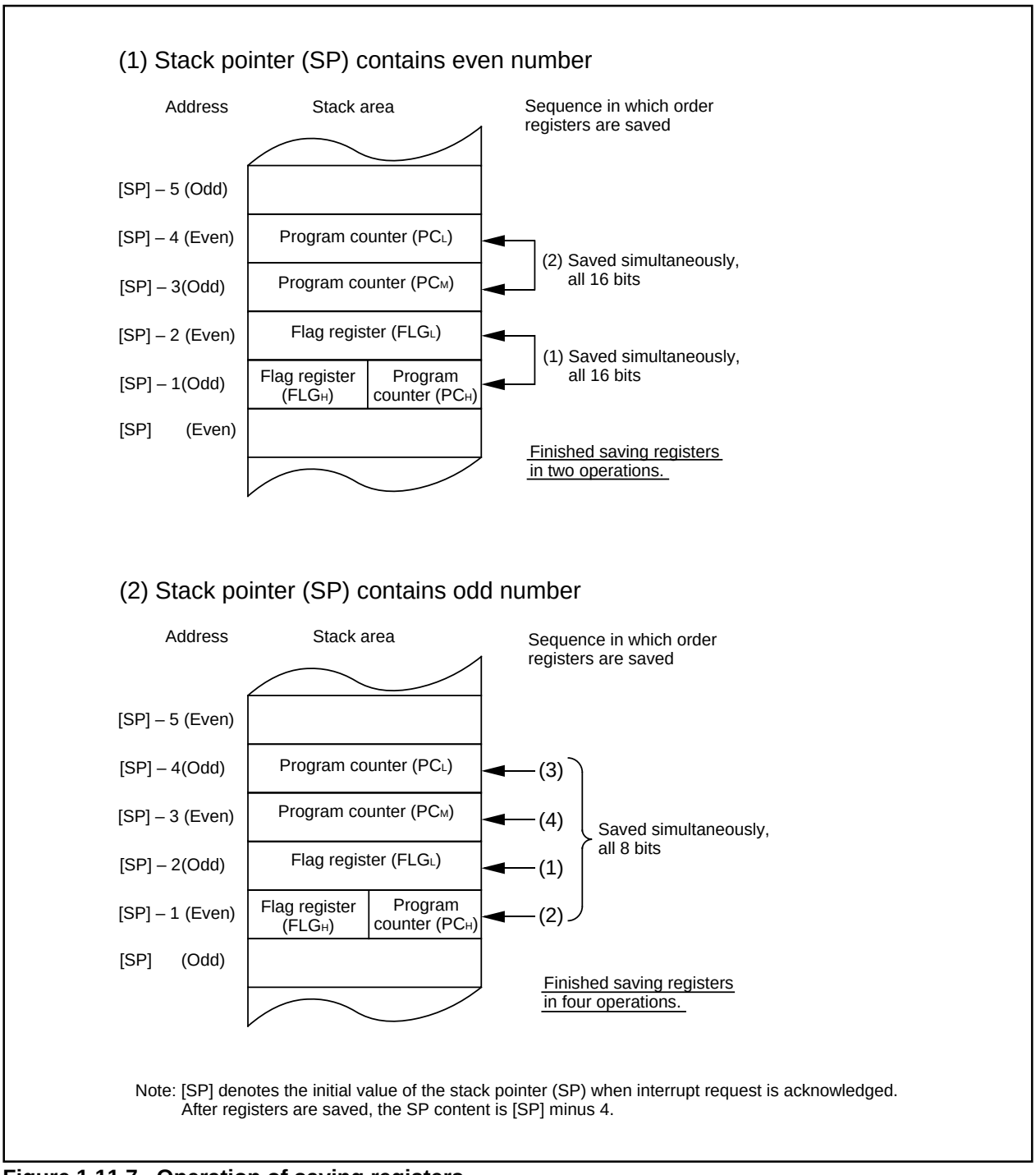


Figure 1.11.7. Operation of saving registers

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 1.11.8 shows the priorities of hardware interrupts.

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

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

Figure 1.11.8. Hardware interrupts priorities

Interrupt resolution circuit

When two or more interrupts are generated simultaneously, this circuit selects the interrupt with the highest priority level. Figure 1.11.9 shows the circuit that judges the interrupt priority level.

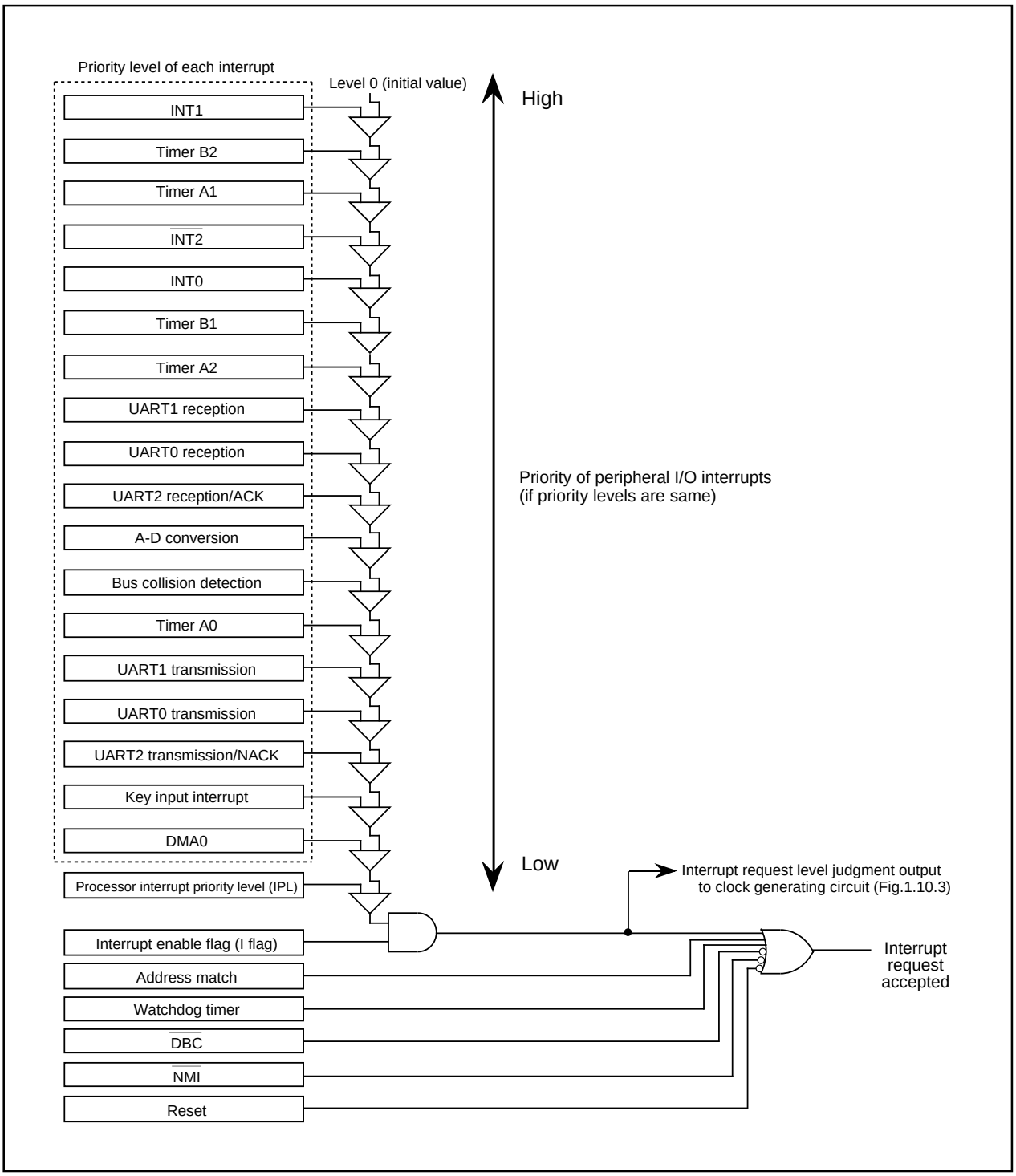


Figure 1.11.9. Maskable interrupts priorities (peripheral I/O interrupts)

INT Interrupt

INT0 to INT2 are triggered by the edges of external inputs. The edge polarity is selected using the polarity select bit.

As for external interrupt input, an interrupt can be generated both at the rising edge and at the falling edge by setting “1” in the INTi interrupt polarity switching bit of the interrupt request cause select register (035F16). To select both edges, set the polarity switching bit of the corresponding interrupt control register to ‘falling edge’ (“0”).

Figure 1.11.10 shows the Interrupt request cause select register.

Interrupt request cause select register

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

Symbol: IFSR
Address: 035F16
When reset: 0016

Bit symbol	Bit name	Function	R	W
IFSR0	INT0 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR1	INT1 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR2	INT2 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
Reserved bit		Must always be set to “0”	○	○
Reserved bit		Must always be set to “0”	○	○
Reserved bit		Must always be set to “0”	○	○
Reserved bit		Must always be set to “0”	○	○
Reserved bit		Must always be set to “0”	○	○

Figure 1.11.10. Interrupt request cause select register

NMI Interrupt

An NMI interrupt is generated when the input to the P85/NMI pin changes from “H” to “L”. The NMI interrupt is a non-maskable external interrupt. The pin level can be checked in the port P85 register (bit 5 at address 03F016).

This pin cannot be used as a normal port input.

Key Input Interrupt

If the direction register of any of P104 to P107 is set for input and a falling edge is input to that port, a key input interrupt is generated. A key input interrupt can also be used as a key-on wakeup function for cancelling the wait mode or stop mode. However, if you intend to use the key input interrupt, do not use P104 to P107 as A-D input ports. Figure 1.11.11 shows the block diagram of the key input interrupt. Note that if an “L” level is input to any pin that has not been disabled for input, inputs to the other pins are not detected as an interrupt.

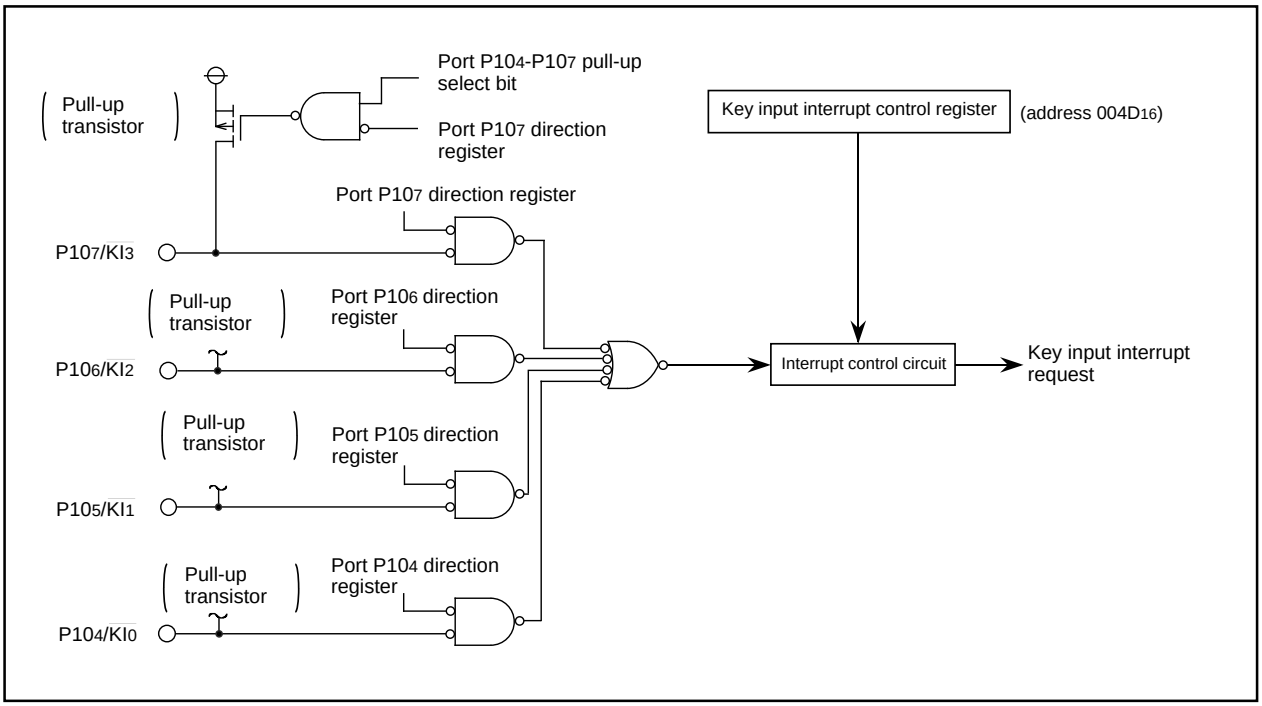


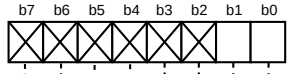
Figure 1.11.11. Block diagram of key input interrupt

Address Match Interrupt

An address match interrupt is generated when the address match interrupt address register contents match the program counter value. Two address match interrupts can be set, each of which can be enabled and disabled by an address match interrupt enable bit. Address match interrupts are not affected by the interrupt enable flag (I flag) and processor interrupt priority level (IPL). For an address match interrupt, the value of the program counter (PC) that is saved to the stack area varies depending on the instruction being executed. Note that when using the external data bus in width of 8 bits, the address match interrupt cannot be used for external area.

Figure 1.11.12 shows the address match interrupt-related registers.

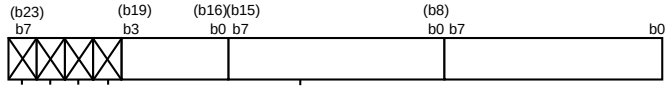
Address match interrupt enable register



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 indeterminated.			—	—

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 indeterminated.		—	—

Figure 1.11.12. Address match interrupt-related registers

Precautions for Interrupts

(1) Reading address 00000$_{16}$

- 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 00000$_{16}$ will then be set to "0".
  Even if the address 00000$_{16}$ 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 00000$_{16}$ by software.

(2) Setting the stack pointer

- The value of the stack pointer immediately after reset is initialized to 0000$_{16}$. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt. When using the $\overline{\text{NMI}}$ interrupt, initialize the stack pointer at the beginning of a program. Concerning the first instruction immediately after reset, generating any interrupts including the $\overline{\text{NMI}}$ interrupt is prohibited.

(3) The $\overline{\text{NMI}}$ interrupt

- The $\overline{\text{NMI}}$ interrupt can not be disabled. Be sure to connect $\overline{\text{NMI}}$ pin to Vcc via a pull-up resistor if unused. Be sure to work on it.
- The $\overline{\text{NMI}}$ pin also serves as P8$_5$, which is exclusively input. Reading the contents of the P8 register allows reading the pin value. Use the reading of this pin only for establishing the pin level at the time when the $\overline{\text{NMI}}$ interrupt is input.
- Do not attempt to go into stop mode with the input to the $\overline{\text{NMI}}$ pin being in the "L" state. With the input to the $\overline{\text{NMI}}$ being in the "L" state, the CM10 is fixed to "0", so attempting to go into stop mode is turned down.
- Do not attempt to go into wait mode with the input to the $\overline{\text{NMI}}$ pin being in the "L" state. With the input to the $\overline{\text{NMI}}$ pin being in the "L" state, the CPU stops but the oscillation does not stop, so no power is saved. In this instance, the CPU is returned to the normal state by a later interrupt.
- Signals input to the $\overline{\text{NMI}}$ pin require "L" level and "H" level of 2 clock +300ns or more, from the operation clock of the CPU.

(4) External interrupt

- Either an "L" level or an "H" level of at least 250 ns width is necessary for the signal input to pins $\overline{\text{INT}_0}$ through $\overline{\text{INT}_2}$ regardless of the CPU operation clock.
- When the polarity of the $\overline{\text{INT}_0}$ to $\overline{\text{INT}_2}$ pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Figure 1.11.13 shows the procedure for changing the $\overline{\text{INT}}$ interrupt generate factor.

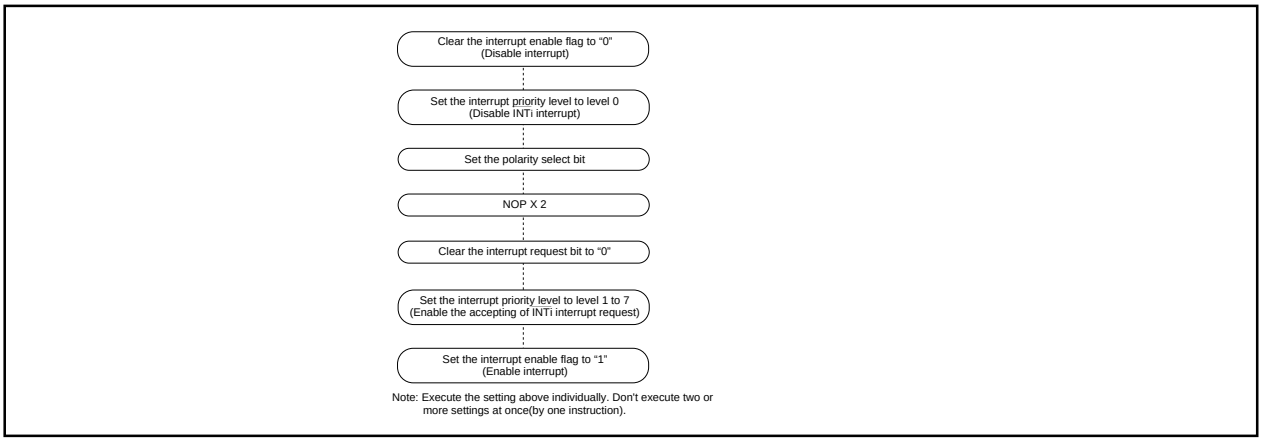


Figure 1.11.13. Switching condition of $\overline{\text{INT}}$ interrupt request

(5) Watchdog timer interrupt

• Write to the watchdog timer start register after the watchdog timer interrupt occurs (initialize watchdog timer).

(6) Rewrite the interrupt control register

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

Example 1:

```
INT_SWITCH1:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    NOP                     ; Four NOP instructions are required when using HOLD function.
    NOP
    FSET    I               ; Enable interrupts.
```

Example 2:

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

Example 3:

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

The reason why two NOP instructions (four when using the HOLD function) 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 changing an interrupt control register in a sate of interrupts being disabled, please read the following precautions on instructions used before changing the register.

Changing a non-interrupt request bit

If an interrupt request for an interrupt control register is generated during an instruction to rewrite the register is being executed, there is a case that the interrupt request bit is not set and consequently the interrupt is ignored. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

Changing the interrupt request bit

When attempting to clear the interrupt request bit of an interrupt control register, the interrupt request bit is not cleared sometimes. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : MOV

Watchdog Timer

The watchdog timer has the function of detecting when the program is out of control. Therefore, we recommend using the watchdog timer to improve reliability of a system. The watchdog timer is a 15-bit counter which down-counts the clock derived by dividing the BCLK using the prescaler. Whether a watchdog timer interrupt is generated or reset is selected when an underflow occurs in the watchdog timer. When the watchdog timer interrupt is selected, write to the watchdog timer start register after the watchdog timer interrupt occurs (initialize watchdog timer). Watchdog timer interrupt is selected when bit 2 (PM12) of the processor mode register 1 (address 0005_{16}) is "0" and reset is selected when PM12 is "1". No value other than "1" can be written in PM12. Once when reset is selected (PM12="1"), watchdog timer interrupt cannot be selected by software.

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 $000F_{16}$). Thus the watchdog timer's period can be calculated as given below. The watchdog timer's period is, however, subject to an error due to the prescaler.

With X_{IN} chosen for BCLK

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

With X_{CIN} chosen for BCLK

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

For example, suppose that BCLK runs at 16 MHz and that 16 has been chosen for the dividing ratio of the prescaler, then the watchdog timer's period becomes approximately 32.8 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 $000E_{16}$). In stop mode, wait mode and hold state, the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes or state are released.

Also PM12 is initialized only when reset. The watchdog timer interrupt is selected after reset is cancelled.

Figure 1.12.1 shows the block diagram of the watchdog timer. Figure 1.12.2 shows the watchdog timer-related registers.

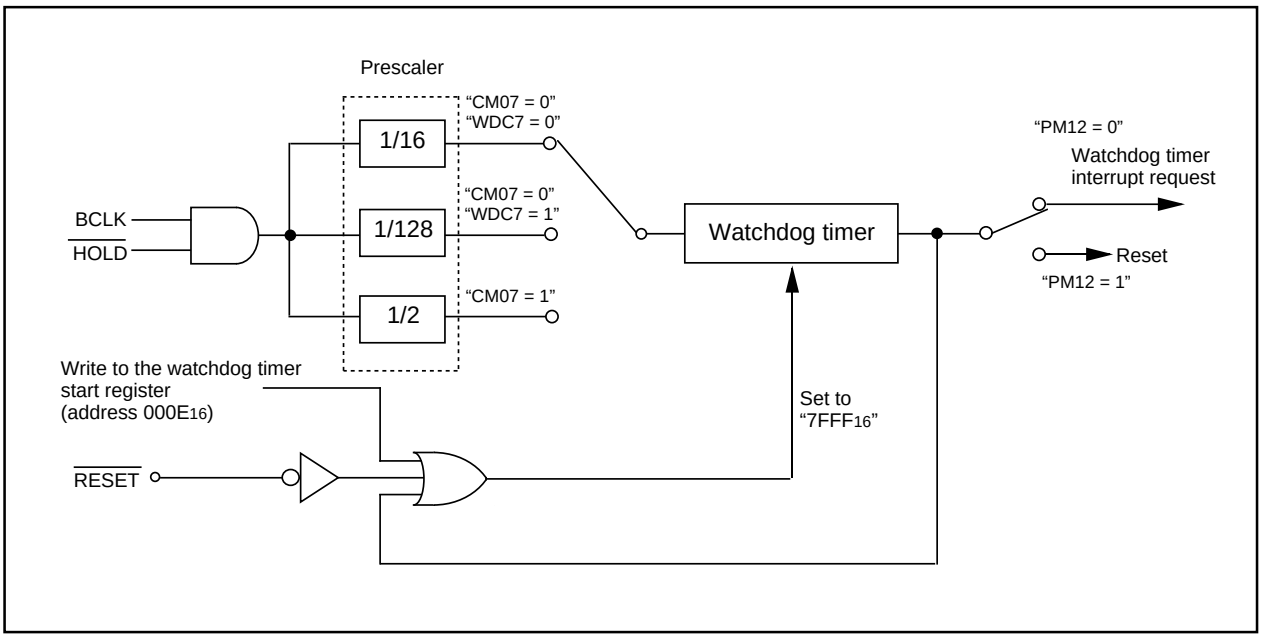


Figure 1.12.1. Block diagram of watchdog timer

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol: WDC | Address: 000F16 | When reset: 000XXXXX2

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

Watchdog timer start register

b7	b0

Symbol: WDTS | Address: 000E16 | When reset: 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 "7FFF16" regardless of whatever value is written.	×	○

Figure 1.12.2. Watchdog timer control and start registers

DMAC

This microcomputer has one DMAC (direct memory access controller) channel that allow data to be sent to memory without using the CPU. DMAC shares the same data bus with the CPU. The DMAC is given a higher right of using the bus than the CPU, which leads to working the cycle stealing method. On this account, the operation from the occurrence of DMA transfer request signal to the completion of 1-word (16-bit) or 1-byte (8-bit) data transfer can be performed at high speed. Figure 1.13.1 shows the block diagram of the DMAC. Table 1.13.1 shows the DMAC specifications. Figures 1.13.2 to 1.13.4 show the registers used by the DMAC.

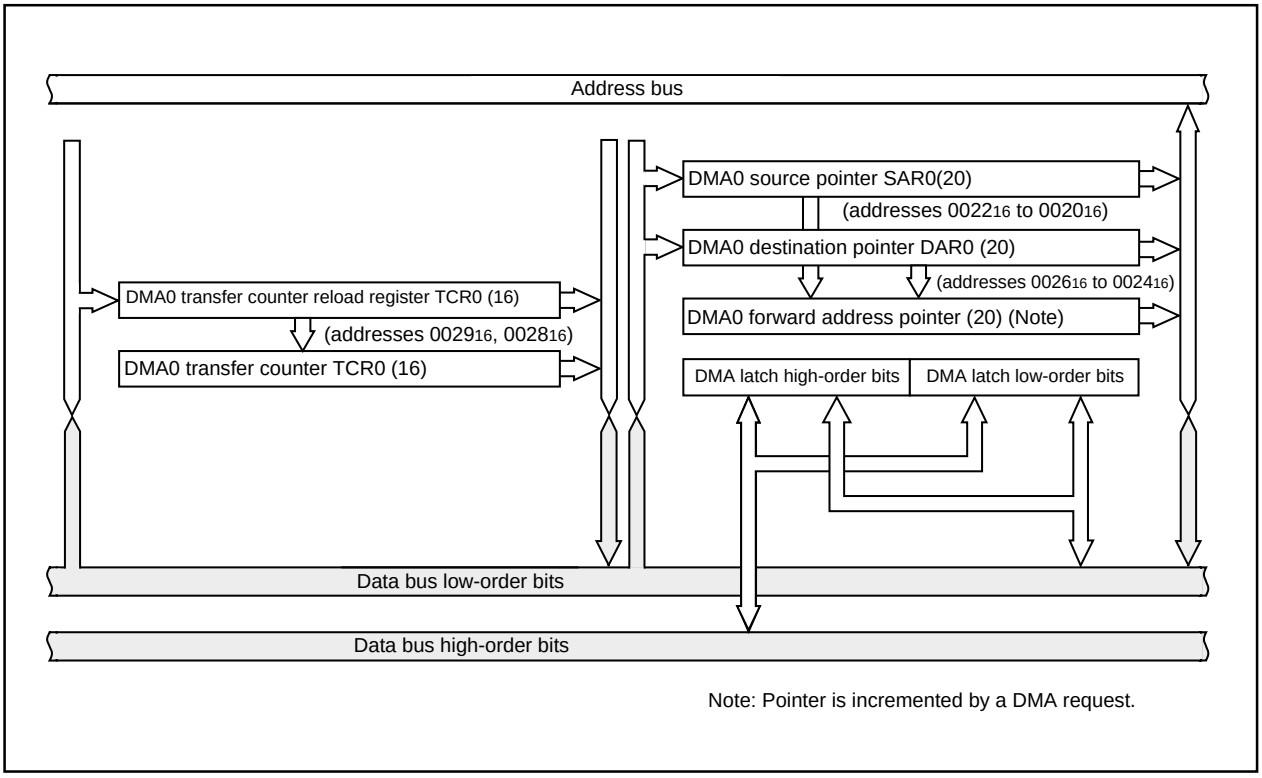


Figure 1.13.1. Block diagram of DMAC

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

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

Table 1.13.1. DMAC specifications

Item	Specification
No. of channels	2 (cycle steal method)
Transfer memory space	• From any address in the 1M bytes space to a fixed address • From a fixed address to any address in the 1M bytes space • From a fixed address to a fixed address (Note that DMA-related registers [0020$_{16}$ to 003F$_{16}$] cannot be accessed)
Maximum No. of bytes transferred	128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers)
DMA request factors (Note)	Falling edge of $\overline{\text{INT0}}$ or both edge Timer A0 to timer A2 interrupt requests Timer B1 and timer B2 interrupt requests UART0 transfer and reception interrupt requests UART1 transfer and reception interrupt requests UART2 transfer and reception interrupt requests A-D conversion interrupt requests Software triggers
Transfer unit	8 bits or 16 bits
Transfer address direction	forward/fixed (forward direction cannot be specified for both source and destination simultaneously)
Transfer mode	• Single transfer mode After the transfer counter underflows, the DMA enable bit turns to "0", and the DMAC turns inactive • Repeat transfer mode After the transfer counter underflows, the value of the transfer counter reload register is reloaded to the transfer counter. The DMAC remains active unless a "0" is written to the DMA enable bit.
DMA interrupt request generation timing	When an underflow occurs in the transfer counter
Active	When the DMA enable bit is set to "1". When the DMAC is active, data transfer starts every time a DMA transfer request signal occurs.
Inactive	• When the DMA enable bit is set to "0".
Reload timing for forward address pointer and transfer counter	At the time of starting data transfer immediately after turning the DMAC active, the value of one of source pointer and destination pointer - the one specified for the forward direction - is reloaded to the forward direction address pointer, and the value of the transfer counter reload register is reloaded to the transfer counter.
Writing to register	Registers specified for forward direction transfer are always write enabled. Registers specified for fixed address transfer are write-enabled when the DMA enable bit is "0".
Reading the register	Can be read at any time. However, when the DMA enable bit is "1", reading the register set up as the forward register is the same as reading the value of the forward address pointer.

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

DMA0 request cause select register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: DM0SL
Address: $03B8_{16}$
When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
DSEL0	DMA request cause select bit	b6 b5 b4 b3 b2 b1 b0 (DMS) 0 0 0 0 0 0 0 : Falling edge of $\overline{INT0}$ pin 0 0 0 0 0 0 1 : Software trigger 0 0 0 0 0 1 0 : Timer A0 0 0 0 0 0 1 1 : Timer A1 0 0 0 0 1 0 0 : Timer A2 0 0 0 0 1 0 1 : Must not be set 0 0 0 0 1 1 0 : Must not be set 0 0 0 0 1 1 1 : Must not be set 0 0 0 1 0 0 0 : Timer B1 0 0 0 1 0 0 1 : Timer B2 0 0 0 1 0 1 0 : UART0 transmit 0 0 0 1 0 1 1 : UART0 receive 0 0 0 1 1 0 0 : UART2 transmit 0 0 0 1 1 0 1 : UART2 receive 0 0 0 1 1 1 0 : A-D conversion 0 0 0 1 1 1 1 : UART1 transmit 1 0 0 0 1 1 0 : Two edges of $\overline{INT0}$ pin Must not be set except the above.	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
DSEL4			○	○
DSEL5			○	○
DMS			○	○
DSR	Software DMA request bit	If software trigger is selected, a DMA request is generated by setting this bit to “1” (When read, the value of this bit is always “0”)	○	○

Figure 1.13.2. DMAC register (1)

DMA0 control register

b7	b6	b5	b4	b3	b2	b1	b0
⊠	⊠						

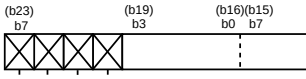
Symbol	Address	When reset
DM0CON	$002C_{16}$	$00000X00_2$

Bit symbol	Bit name	Function	R	W
DMBIT	Transfer unit bit select bit	0 : 16 bits 1 : 8 bits	○	○
DMASL	Repeat transfer mode select bit	0 : Single transfer 1 : Repeat transfer	○	○
DMAS	DMA request bit (Note 1)	0 : DMA not requested 1 : DMA requested	○	○ (Note 2)
DMAE	DMA enable bit	0 : Disabled 1 : Enabled	○	○
DSD	Source address direction select bit (Note 3)	0 : Fixed 1 : Forward	○	○
DAD	Destination address direction select bit (Note 3)	0 : Fixed 1 : Forward	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–

Note 1: DMA request can be cleared by resetting the bit.
Note 2: This bit can only be set to “0”.
Note 3: Source address direction select bit and destination address direction select bit cannot be set to “1” simultaneously.

Figure 1.13.3. DMAC register (2)

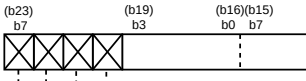
DMA0 source pointer



Symbol	Address	When reset
SAR0	0022$_{16}$ to 0020$_{16}$	Indeterminate

Function	Transfer address specification	R	W
• Source pointer Stores the source address	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 “0”.		—	—

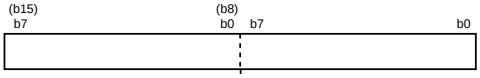
DMA0 destination pointer



Symbol	Address	When reset
DAR0	0026$_{16}$ to 0024$_{16}$	Indeterminate

Function	Transfer address specification	R	W
• Destination pointer Stores the destination address	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 “0”.		—	—

DMA0 transfer counter



Symbol	Address	When reset
TCR0	0029$_{16}$, 0028$_{16}$	Indeterminate

Function	Transfer count specification	R	W
• Transfer counter Set a value one less than the transfer count	0000$_{16}$ to FFFF$_{16}$	○	○

Figure 1.13.4. DMAC register (3)

(1) Transfer cycle

The transfer cycle consists of the bus cycle in which data is read from memory or from the SFR area (source read) and the bus cycle in which the data is written to memory or to the SFR area (destination write). The number of read and write bus cycles depends on the source and destination addresses. In memory expansion mode and microprocessor mode, the number of read and write bus cycles also depends on the level of the BYTE pin. Also, the bus cycle itself is longer when software waits are inserted.

(a) Effect of source and destination addresses

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

(b) Effect of BYTE pin level

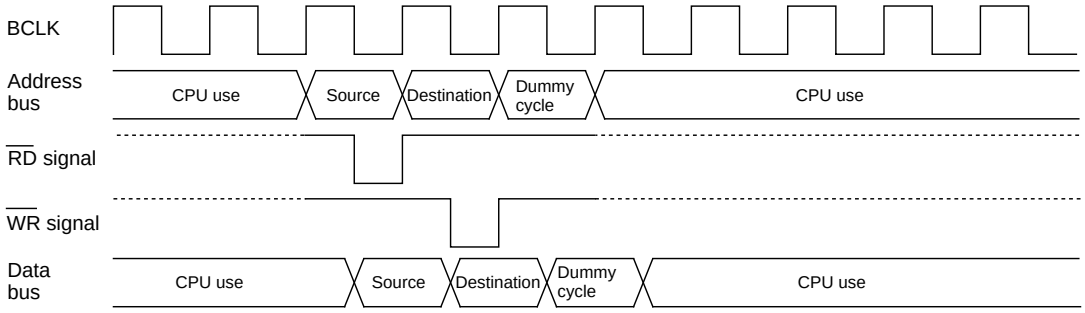
When transferring 16-bit data over an 8-bit data bus (BYTE pin = “H”) in memory expansion mode and microprocessor mode, the 16 bits of data are sent in two 8-bit blocks. Therefore, two bus cycles are required for reading the data and two are required for writing the data. Also, in contrast to when the CPU accesses internal memory, when the DMAC accesses internal memory (internal ROM, internal RAM, and SFR), these areas are accessed using the data size selected by the BYTE pin.

(c) Effect of software wait

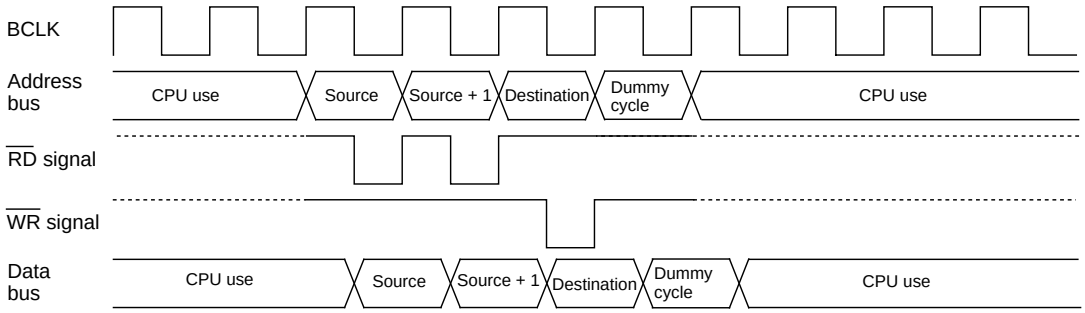
When the SFR area or a memory area with a software wait is accessed, the number of cycles is increased for the wait by 1 bus cycle. The length of the cycle is determined by BCLK.

Figure 1.13.5 shows the example of the transfer cycles for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating the transfer cycle, remember to apply the respective conditions to both the destination write cycle and the source read cycle. For example (2) in Figure 1.13.5, if data is being transferred in 16-bit units on an 8-bit bus, two bus cycles are required for both the source read cycle and the destination write cycle.

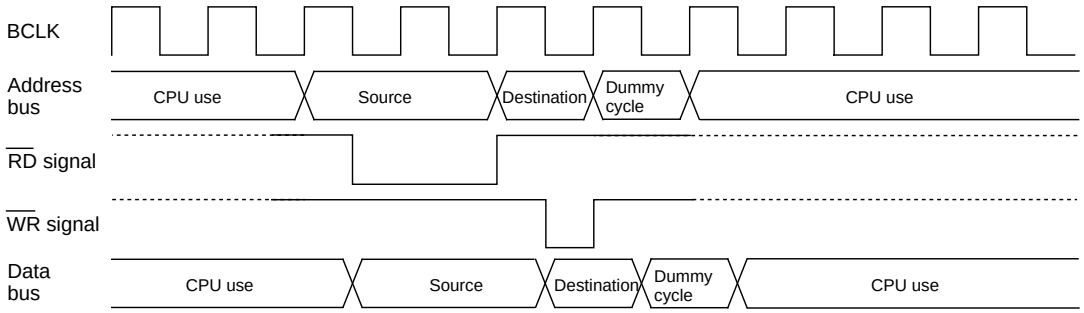
(1) 8-bit transfers
16-bit transfers and the source address is even.



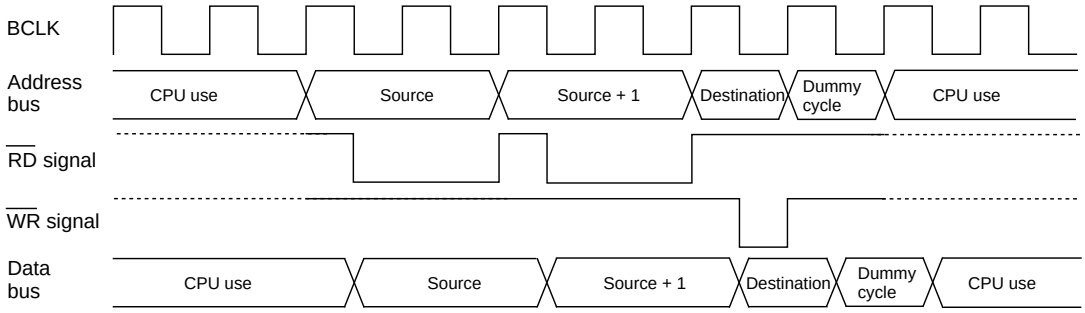
(2) 16-bit transfers and the source address is odd
Transferring 16-bit data on an 8-bit data bus (In this case, there are also two destination cycles).



(3) One wait is inserted into the source read under the conditions in (1)



(4) One wait is inserted into the source read under the conditions in (2)
(When 16-bit data is transferred on an 8-bit data bus, there are two destination cycles).



Note: The same timing changes occur with the respective conditions at the destination as at the source.

Figure 1.13.5. Example of the transfer cycles for a source read

(2) DMAC transfer cycles

Any combination of even or odd transfer read and write addresses is possible. Table 1.13.2 shows the number of DMAC transfer cycles.

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

No. of transfer cycles per transfer unit = No. of read cycles x j + No. of write cycles x k

Table 1.13.2. No. of DMAC transfer cycles

Transfer unit	Bus width	Access address	Single-chip mode		Memory expansion mode Microprocessor mode	
			No. of read cycles	No. of write cycles	No. of read cycles	No. of write cycles
8-bit transfers (DMBIT= “1”)	16-bit (BYTE= “L”)	Even	1	1	1	1
		Odd	1	1	1	1
	8-bit (BYTE = “H”)	Even	—	—	1	1
		Odd	—	—	1	1
16-bit transfers (DMBIT= “0”)	16-bit (BYTE = “L”)	Even	1	1	1	1
		Odd	2	2	2	2
	8-bit (BYTE = “H”)	Even	—	—	2	2
		Odd	—	—	2	2

Coefficient j, k

Internal memory			External memory		
Internal ROM/RAM No wait	Internal ROM/RAM With wait	SFR area	Separate bus No wait	Separate bus With wait	Multiplex bus
1	2	2	1	2	3

DMA enable bit

Setting the DMA enable bit to “1” makes the DMAC active. The DMAC carries out the following operations at the time data transfer starts immediately after DMAC is turned active.

(1) Reloads the value of one of the source pointer and the destination pointer - the one specified for the forward direction - to the forward direction address pointer.

(2) Reloads the value of the transfer counter reload register to the transfer counter.

Thus overwriting “1” to the DMA enable bit with the DMAC being active carries out the operations given above, so the DMAC operates again from the initial state at the instant “1” is overwritten to the DMA enable bit.

DMA request bit

The DMAC can generate a DMA transfer request signal triggered by a factor chosen in advance out of DMA request factors.

DMA request factors include the following.

* Factors effected by using the interrupt request signals from the built-in peripheral functions and software DMA factors (internal factors) effected by a program.
* External factors effected by utilizing the input from external interrupt signals.

For the selection of DMA request factors, see the descriptions of the DMA0 factor selection register.

The DMA request bit turns to “1” if the DMA transfer request signal occurs regardless of the DMAC's state (regardless of whether the DMA enable bit is set to “1” or “0”). It turns to “0” immediately before data transfer starts.

In addition, it can be set to “0” by use of a program, but cannot be set to “1”.

There can be instances in which a change in DMA request factor selection bit causes the DMA request bit to turn to “1”. So be sure to set the DMA request bit to “0” after the DMA request factor selection bit is changed.

The DMA request bit turns to “1” if a DMA transfer request signal occurs, and turns to “0” immediately before data transfer starts. If the DMAC is active, data transfer starts immediately, so the value of the DMA request bit, if read by use of a program, turns out to be “0” in most cases. To examine whether the DMAC is active, read the DMA enable bit.

Here follows the timing of changes in the DMA request bit.

(1) Internal factors

Except the DMA request factors triggered by software, the timing for the DMA request bit to turn to “1” due to an internal factor is the same as the timing for the interrupt request bit of the interrupt control register to turn to “1” due to several factors.

Turning the DMA request bit to “0” due to an internal factor is timed to be effected immediately before the transfer starts.

(2) External factors

An external factor is a factor caused to occur by the leading edge of input from the $\overline{\text{INT0}}$ pin.

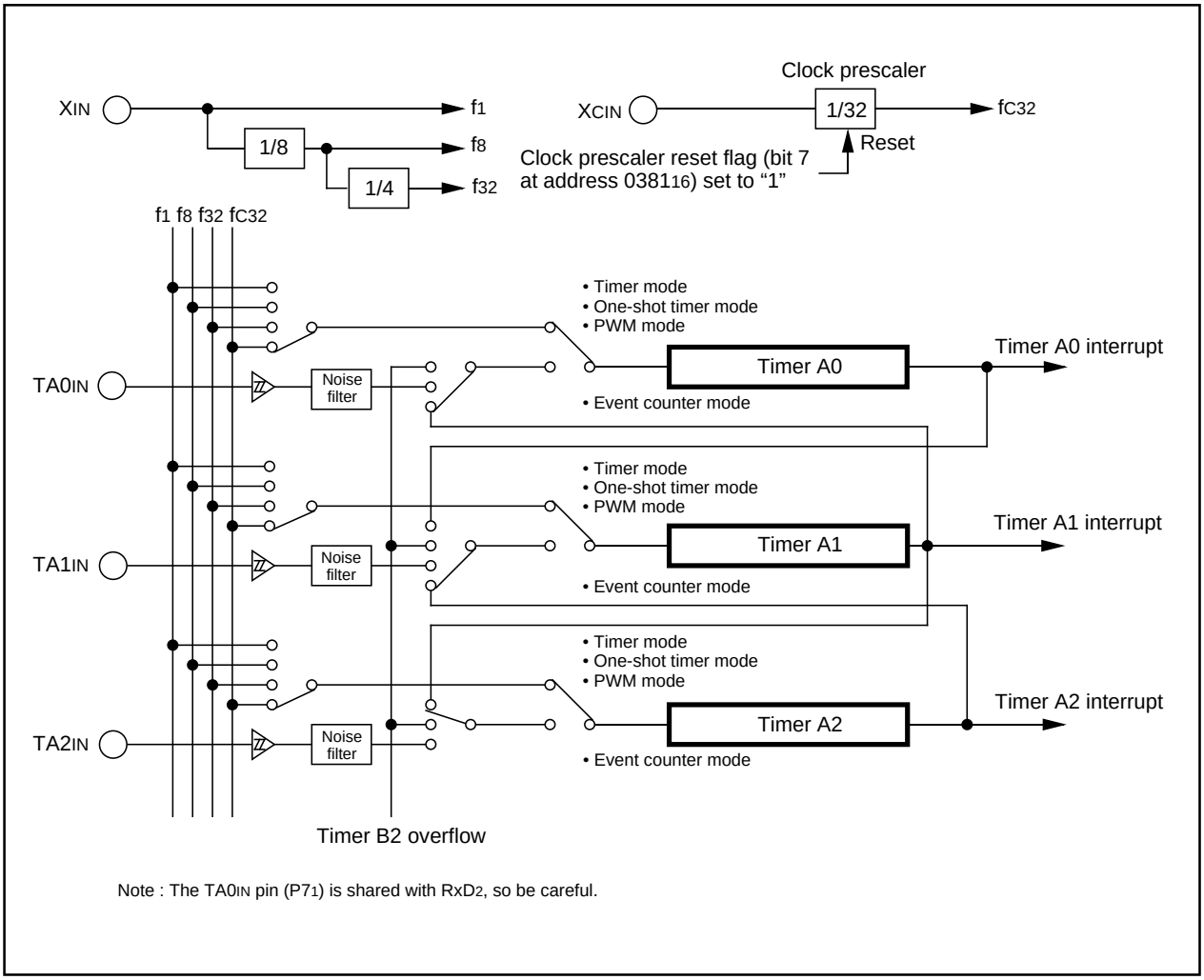
Selecting the $\overline{\text{INT0}}$ pins as external factors using the DMA request factor selection bit causes input from these pins to become the DMA transfer request signals.

The timing for the DMA request bit to turn to “1” when an external factor is selected synchronizes with the signal's edge applicable to the function specified by the DMA request factor selection bit (synchronizes with the trailing edge of the input signal to $\overline{\text{INT0}}$ pin, for example).

With an external factor selected, the DMA request bit is timed to turn to “0” immediately before data transfer starts similarly to the state in which an internal factor is selected.

Timer

There are five 16-bit timers. These timers can be classified by function into timers A (three) and timers B (two). All these timers function independently. Figures 1.14.1 and 1.14.2 show the block diagram of timers.



Note : The TA0IN pin (P71) is shared with RxD2, so be careful.

Figure 1.14.1. Timer A block diagram

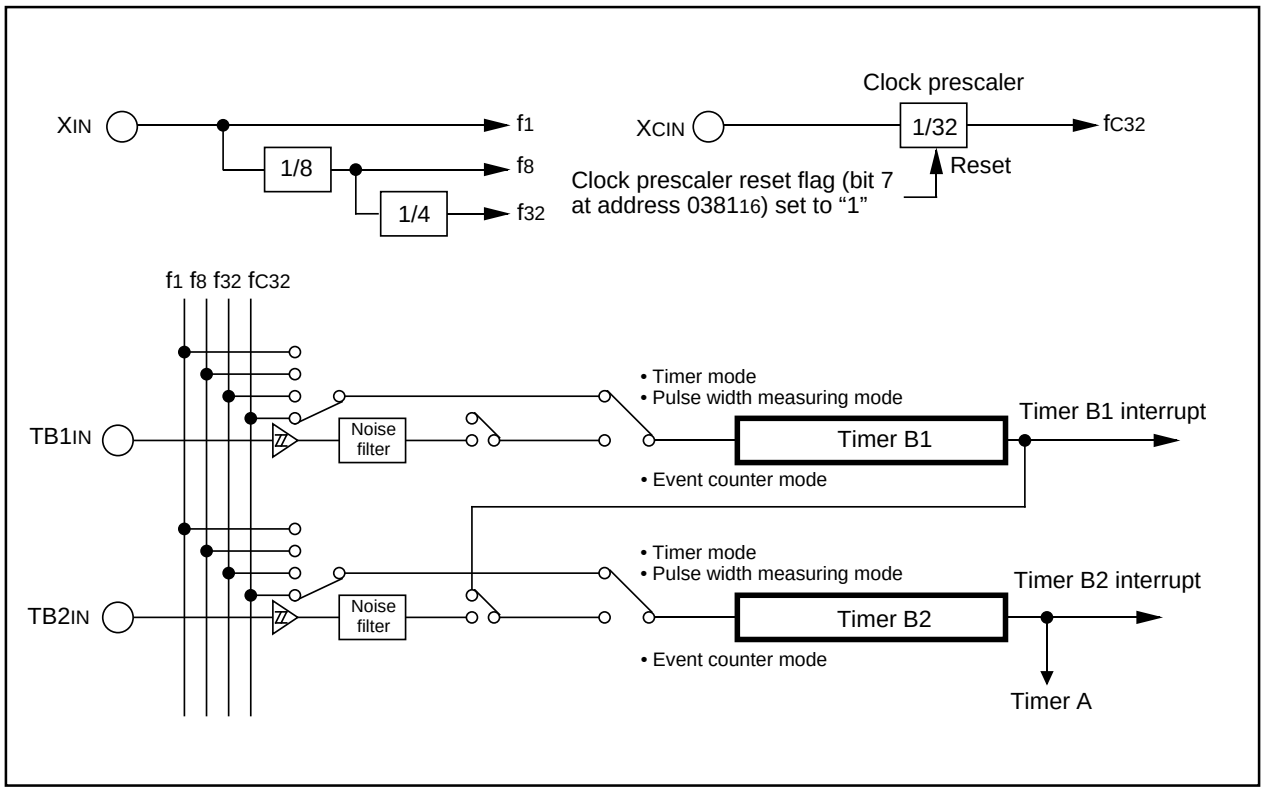


Figure 1.14.2. Timer B block diagram

Timer A

Figure 1.14.3 shows the block diagram of timer A. Figures 1.14.4 to 1.14.6 show the timer A-related registers.

Except in event counter mode, timers A0 through A2 all have the same function. Use the timer Ai mode register (i = 0 to 2) bits 0 and 1 to choose the desired mode.

Timer A has the four operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer overflow.
- One-shot timer mode: The timer stops counting when the count reaches "0000$_{16}$".
- Pulse width modulation (PWM) mode: The timer outputs pulses of a given width.

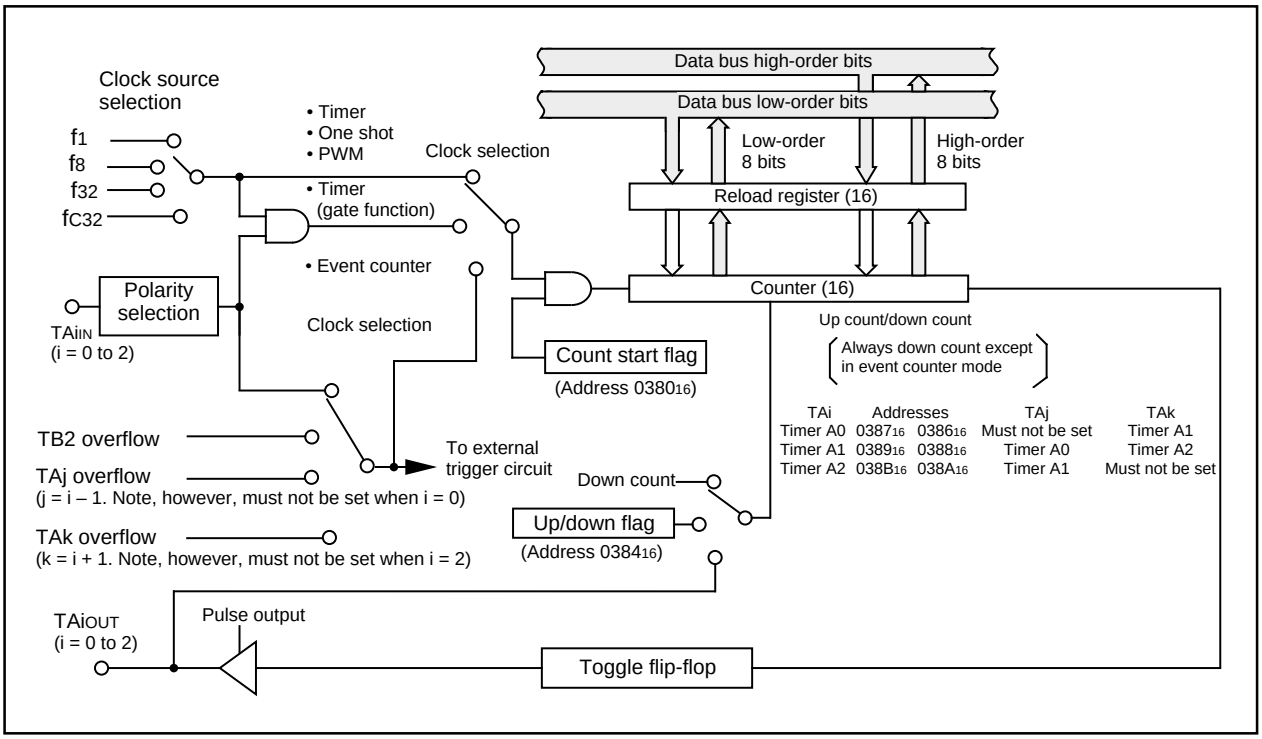


Figure 1.14.3. Block diagram of timer A

Timer Ai mode register

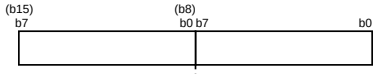
Symbol	Address	When reset
TAiMR(i=0 to 2)	039616 to 039816	0016

b7	b6	b5	b4	b3	b2	b1	b0

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

Figure 1.14.4. Timer A-related registers (1)

Timer Ai register (Note 1)



Symbol	Address	When reset
TA0	0387$_{16}$,0386$_{16}$	Indeterminate
TA1	0389$_{16}$,0388$_{16}$	Indeterminate
TA2	038B$_{16}$,038A$_{16}$	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	0000$_{16}$ to FFFF$_{16}$	○	○
• Event counter mode Counts pulses from an external source or timer overflow	0000$_{16}$ to FFFF$_{16}$	○	○
• One-shot timer mode Counts a one shot width	0000$_{16}$ to FFFF$_{16}$ (Note 2,4)	×	○
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000$_{16}$ to FFFE$_{16}$ (Note 3,4)	×	○
• Pulse width modulation mode (8-bit PWM) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	00$_{16}$ to FE$_{16}$ (High-order address) 00$_{16}$ to FF$_{16}$ (Low-order address) (Note 3,4)	×	○

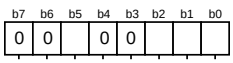
Note 1: Read and write data in 16-bit units.
Note 2: When the timer Ai register is set to "0000$_{16}$", the counter does not operate and the timer Ai interrupt request is not generated. When the pulse is set to output, the pulse does not output from the TAiOUT pin.
Note 3: When the timer Ai register is set to "0000$_{16}$", the pulse width modulator does not operate and the output level of the TAiOUT pin remains "L" level, therefore the timer Ai interrupt request is not generated. This also occurs in the 8-bit pulse width modulator mode when the significant 8 high-order bits in the timer Ai register are set to "00$_{16}$".
Note 4: Use MOV instruction to write to this register.

Count start flag

Symbol	Address	When reset
TABSR	0380$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TA1S	Timer A1 count start flag		○	○
TA2S	Timer A2 count start flag		○	○
Reserved bit		Must always be set to "0"	○	○
TB1S	Timer B1 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB2S	Timer B2 count start flag		○	○

Up/down flag (Note 1)



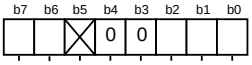
Symbol	Address	When reset
UDF	0384$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count This specification becomes valid when the up/down flag content is selected for up/down switching cause	○	○
TA1UD	Timer A1 up/down flag		○	○
TA2UD	Timer A2 up/down flag		○	○
Reserved bit		Must always be set to "0"	○	○
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled (Note 2) When not using the two-phase pulse signal processing function, set the select bit to "0"	×	○
Reserved bit		Must always be set to "0"	×	○

Note 1: Use MOV instruction to write to this register.
Note 2: Set the TAiIN and TAiOUT pins correspondent port direction registers to "0".

Figure 1.14.5. Timer A-related registers (2)

One-shot start flag

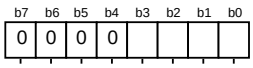


Symbol	Address	When reset
ONSF	0382_{16}	$00X00000_2$

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is “0”	○	○
TA1OS	Timer A1 one-shot start flag		○	○
TA2OS	Timer A2 one-shot start flag		○	○
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.			—	—
TA0TGL	Timer A0 event/trigger select bit	b7 b6 0 0 : Input on $TA0_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : Must not be set 1 1 : TA1 overflow is selected	○	○
TA0TGH			○	○

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

Trigger select register

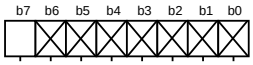


Symbol	Address	When reset
TRGSR	0383_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on $TA1_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TA2 overflow is selected	○	○
TA1TGH			○	○
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on $TA2_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA1 overflow is selected 1 1 : Must not be set	○	○
TA2TGH			○	○
Reserved bit		Must always be set to “0”	○	○

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

Clock prescaler reset flag



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

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

Figure 1.14.6. Timer A-related registers (3)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.14.1.) Figure 1.14.7 shows the timer Ai mode register in timer mode.

Table 1.14.1. Specifications of timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	When the timer underflows
TAiIN pin function	Programmable I/O port or gate input
TAiOUT pin function	Programmable I/O port or pulse output
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Gate function Counting can be started and stopped by the TAiIN pin's input signal • Pulse output function Each time the timer underflows, the TAiOUT pin's polarity is reversed

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol: TAiMR(i=0 to 2) — Address: 039_{16} to 039_{16} — When reset: 00_{16}

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

Note 1: The settings of the corresponding port register and port direction register are invalid.
Note 2: The bit can be "0" or "1".
Note 3: Set the corresponding port direction register to "0".

Figure 1.14.7. Timer Ai mode register in timer mode

(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. Timers A0 and A1 can count a single-phase external signal. Timer A2 can count a single-phase and a two-phase external signal. Table 1.14.2 lists timer specifications when counting a single-phase external signal. Figure 1.14.8 shows the timer Ai mode register in event counter mode.

Table 1.14.3 lists timer specifications when counting a two-phase external signal. Figure 1.14.9 shows the timer Ai mode register in event counter mode.

Table 1.14.2. Timer specifications in event counter mode (when not processing two-phase pulse signal)

Item	Specification
Count source	• External signals input to TAiIN pin (effective edge can be selected by software) • TB2 overflow, TAj overflow
Count operation	• Up count or down count can be selected by external signal or software • When the timer overflows or underflows, it reloads the reload register con tents before continuing counting (Note)
Divide ratio	1/ (FFFF16 - n + 1) for up count 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer overflows or underflows
TAiIN pin function	Programmable I/O port or count source input
TAiOUT pin function	Programmable I/O port, pulse output, or up/down count select input
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it • Pulse output function Each time the timer overflows or underflows, the TAiOUT pin's polarity is reversed

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

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

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

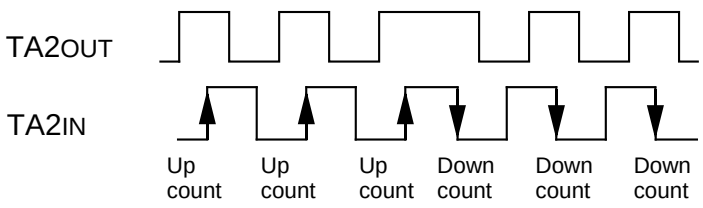
Symbol TAiMR(i = 0 to 2) Address 039616 to 039816 When reset 0016

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

Note 1: In event counter mode, the count source is selected by the event / trigger select bit (addresses 038216 and 038316).
Note 2: The settings of the corresponding port register and port direction register are invalid.
Note 3: Valid only when counting an external signal.
Note 4: When an "L" signal is input to the TAiOUT pin, the downcount is activated. When "H", the upcount is activated. Set the corresponding port direction register to "0".

Figure 1.14.8. Timer Ai mode register in event counter mode

Table 1.14.3. Timer specifications in event counter mode (when processing two-phase pulse signal with timer A2)

Item	Specification
Count source	• Two-phase pulse signals input to TA2IN or TA2OUT pin
Count operation	• Up count or down count can be selected by two-phase pulse signal • When the timer overflows or underflows, the reload register content is reloaded and the timer starts over again (Note)
Divide ratio	1/ (FFFF16 - n + 1) for up count 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	Timer overflows or underflows
TA2IN pin function	Two-phase pulse input (Set the TA2IN pin correspondent port direction register to “0”.)
TA2OUT pin function	Two-phase pulse input (Set the TA2OUT pin correspondent port direction register to “0”.)
Read from timer	Count value can be read out by reading timer A2 register
Write to timer	• When counting stopped When a value is written to timer A2 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A2 register, it is written to only reload register. (Transferred to counter at next reload time.)
Select function	• Normal processing operation The timer counts up rising edges or counts down falling edges on the TA2IN pin when input signal on the TA2OUT pin is “H”. 

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

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

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

Symbol: TA2MR
Address: 039816
When reset: 0016

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

Note : When performing two-phase pulse signal processing, make sure the two-phase pulse signal processing operation select bit (address 038416) is set to “1”. Also, always be sure to set the event/trigger select bit (address 038316) to “00”.

Figure 1.14.9. Timer Ai mode register in event counter mode

(3) One-shot timer mode

In this mode, the timer operates only once. (See Table 1.14.4.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 1.14.10 shows the timer Ai mode register in one-shot timer mode.

Table 1.14.4. Timer specifications in one-shot timer mode

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

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

Symbol: TAiMR(i = 0 to 2) | Address: 0396_{16} to 0398_{16} | When reset: 00_{16}

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

Note 1: The settings of the corresponding port register and port direction register are invalid.
Note 2: Valid only when the TAi_{IN} pin is selected by the event/trigger select bit (addresses 0382_{16} and 0383_{16}). If timer overflow is selected, this bit can be "1" or "0".
Note 3: Set the corresponding port direction register to "0".

Figure 1.14.10. Timer Ai mode register in one-shot timer mode

(4) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table 1.14.5.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 1.14.11 shows the timer Ai mode register in pulse width modulation mode. Figure 1.14.12 shows the example of how a 16-bit pulse width modulator operates. Figure 1.14.13 shows the example of how an 8-bit pulse width modulator operates.

Table 1.14.5. Timer specifications in pulse width modulation mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• The timer counts down (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new count at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs when counting
16-bit PWM	• High level width n / fi n : Set value • Cycle time $(2^{16}-1)$ / fi fixed
8-bit PWM	• High level width n×(m+1) / fi n : values set to timer Ai register's high-order address • Cycle time $(2^{8}-1)$× (m+1) / fi m : values set to timer Ai register's low-order address
Count start condition	• External trigger is input • The timer overflows • The count start flag is set (= 1)
Count stop condition	• The count start flag is reset (= 0)
Interrupt request generation timing	PWM pulse goes "L"
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

Symbol	Address	When reset
TAiMR(i=0 to 2)	039_{16} to 039_{816}	00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 1 : PWM mode	○	○
TMOD1			○	○
MR0	1 (Must always be "1" in PWM mode)		○	○
MR1	External trigger select bit (Note 1)	0: Falling edge of TAiIN pin's input signal (Note 2) 1: Rising edge of TAiIN pin's input signal (Note 2)	○	○
MR2	Trigger select bit	0: Count start flag is valid 1: Selected by event/trigger select bits	○	○
MR3	16/8-bit PWM mode select bit	0: Functions as a 16-bit pulse width modulator 1: Functions as an 8-bit pulse width modulator	○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Valid only when the TAiIN pin is selected by the event/trigger select bit (addresses 038_{216} and 038_{316}). If timer overflow is selected, this bit can be "1" or "0".
Note 2: Set the corresponding port direction register to "0".

Figure 1.14.11. Timer Ai mode register in pulse width modulation mode

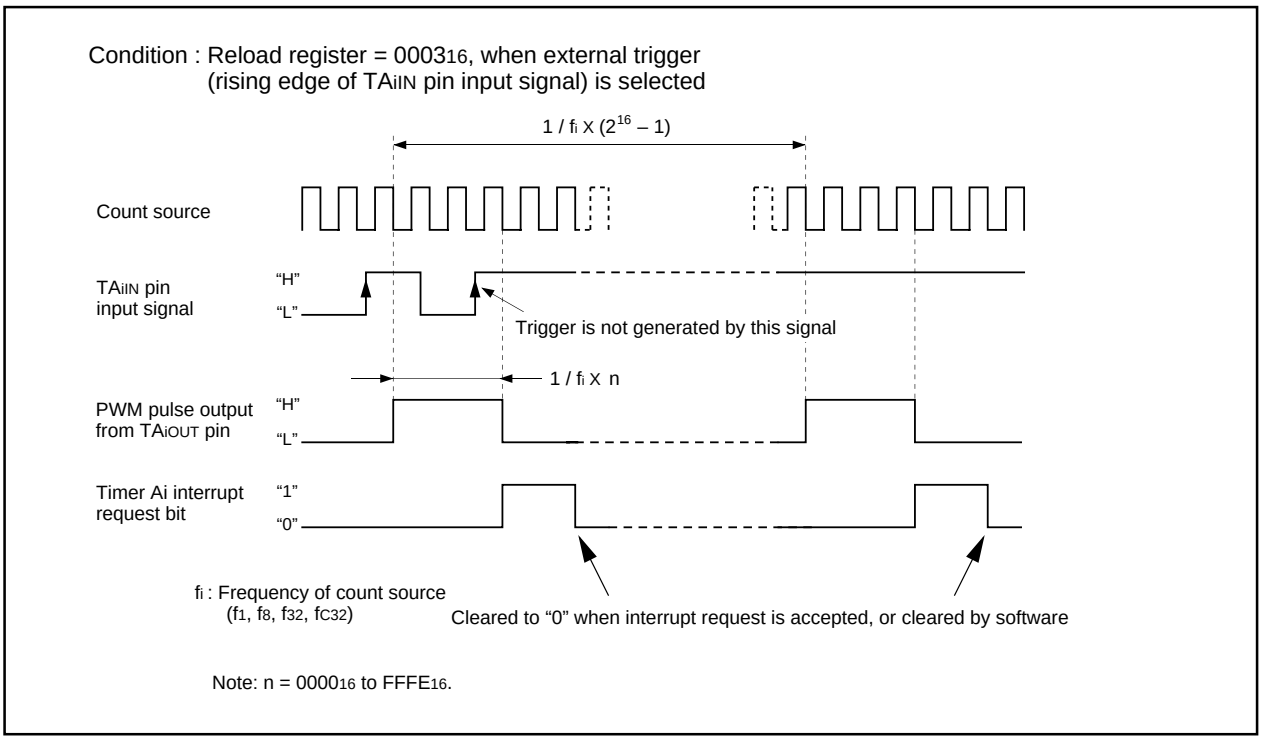


Figure 1.14.12. Example of how a 16-bit pulse width modulator operates

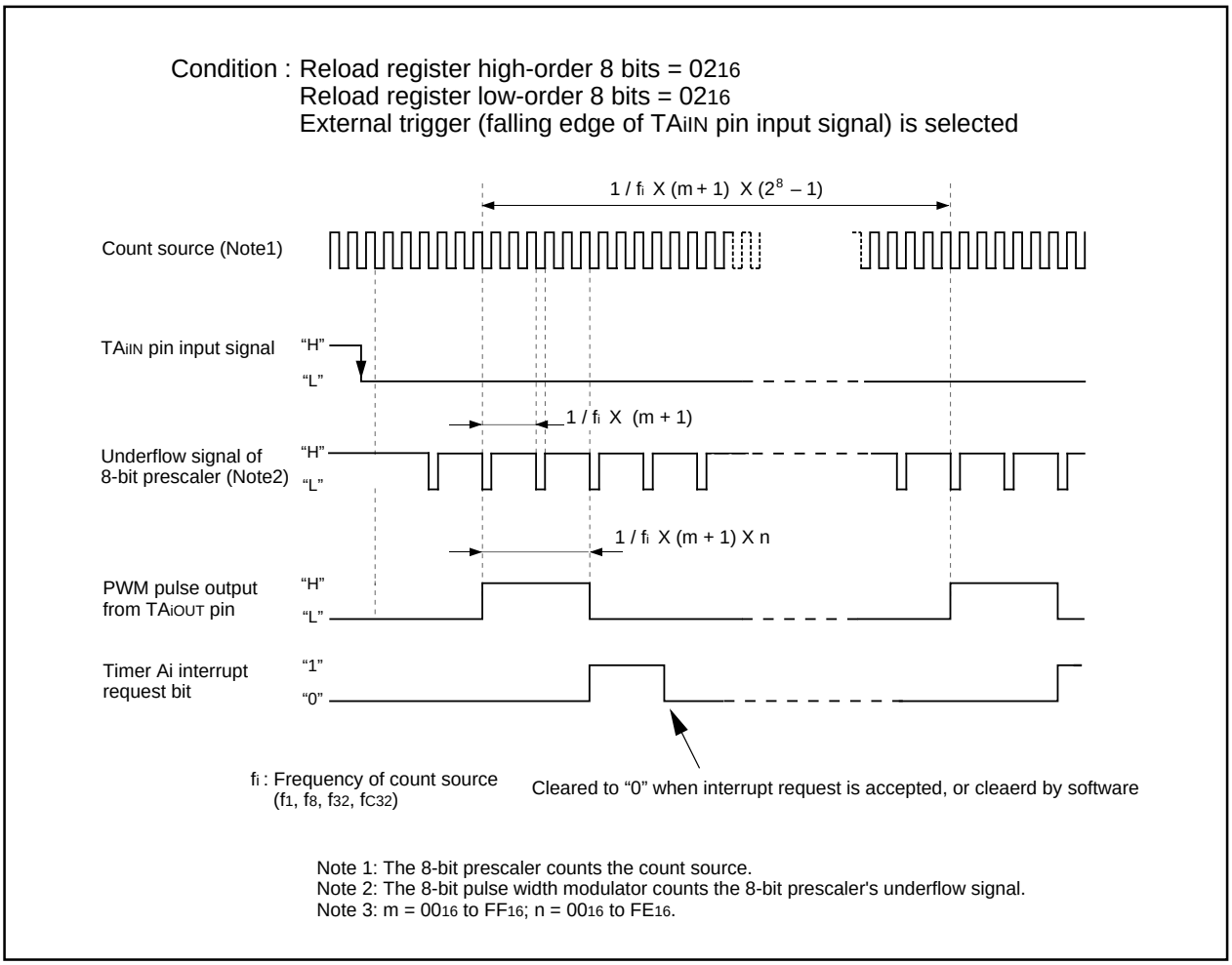


Figure 1.14.13. Example of how an 8-bit pulse width modulator operates

Timer B

Figure 1.14.14 shows the block diagram of timer B. Figures 1.14.15 and 1.14.16 show the timer B-related registers.

Use the timer Bi mode register (i = 1, 2) bits 0 and 1 to choose the desired mode.

Timer B has three operation modes listed as follows:

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

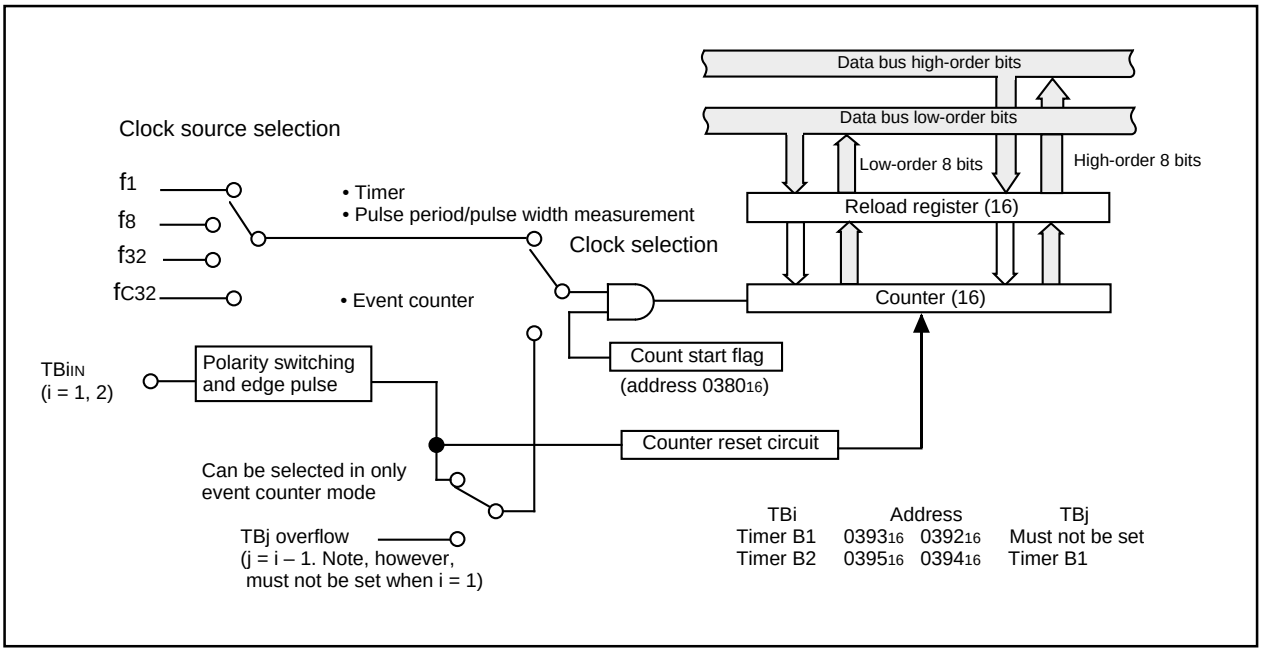


Figure 1.14.14. Block diagram of timer B

Timer Bi mode register

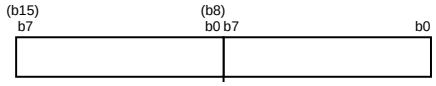
b7	b6	b5	b4	b3	b2	b1	b0
			X				

Symbol: TBiMR(i = 1, 2) — Address: 039C16, 039D16 — When reset: 00XX00002

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/pulse width measurement mode 1 1 : Must not be set.	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
MR3	Function varies with each operation mode		○	×
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Figure 1.14.15. Timer B-related registers (1)

Timer Bi register (Note)

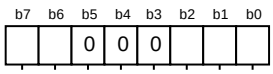


Symbol	Address	When reset
TB1	0393_{16}, 0392_{16}	Indeterminate
TB2	0395_{16}, 0394_{16}	Indeterminate

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

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

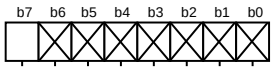
Count start flag



Symbol	Address	When reset
TABSR	0380_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TA1S	Timer A1 count start flag		○	○
TA2S	Timer A2 count start flag		○	○
Reverved bit		Must always be set to “0”	○	○
TB1S	Timer B1 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB2S	Timer B2 count start flag		○	○

Clock prescaler reset flag



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

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

Figure 1.14.16. Timer B-related registers (2)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.14.6.) Figure 1.14.17 shows the timer Bi mode register in timer mode.

Table 1.14.6. Timer specifications in timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Counts down • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Programmable I/O port
Read from timer	Count value is read out by reading timer Bi register
Write to timer	• When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
			X			0	0

Symbol: TBiMR(i=1, 2) Address: $039C_{16}$, $039D_{16}$ When reset: $00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Invalid in timer mode Can be “0” or “1”		○	○
MR1			○	○
Nothing is assiigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.			—	—
MR3	Invalid in timer mode. In an attempt to write to this bit, write “0”. The value, if read in timer mode, turns out to be indeterminate.		○	×
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Figure 1.14.17. Timer Bi mode register in timer mode

(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table 1.14.7.) Figure 1.14.18 shows the timer Bi mode register in event counter mode.

Table 1.14.7. Timer specifications in event counter mode

Item	Specification
Count source	• External signals input to TBiIN pin • Effective edge of count source can be a rising edge, a falling edge, or falling and rising edges as selected by software
Count operation	• Counts down • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Count source input
Read from timer	Count value can be read out by reading timer Bi register
Write to timer	• When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
			X			0	1

Symbol: TBiMR(i=1, 2) — Address: 039C16, 039D16 — When reset: 00XX00002

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	Count polarity select bit (Note 1)	b3 b2 0 0 : Counts external signal's falling edges 0 1 : Counts external signal's rising edges 1 0 : Counts external signal's falling and rising edges 1 1 : Must not be set.	○	○
MR1			○	○
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.			—	—
MR3	Invalid in event counter mode. In an attempt to write to this bit, write "0". The value, if read in event counter mode, turns out to be indeterminate.		○	×
TCK0	Invalid in event counter mode. Can be "0" or "1".		○	○
TCK1	Event clock select	0 : Input from TBiIN pin (Note 2) 1 : TBj overflow (j = i – 1; however, must not be set, when i = 1)	○	○

Note 1: Valid only when input from the TBiIN pin is selected as the event clock. If timer's overflow is selected, this bit can be "0" or "1".
Note 2: Set the corresponding port direction register to "0".

Figure 1.14.18. Timer Bi mode register in event counter mode

(3) Pulse period/pulse width measurement mode

In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table 1.14.8.) Figure 1.14.19 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure 1.14.20 shows the operation timing when measuring a pulse period. Figure 1.14.21 shows the operation timing when measuring a pulse width.

Table 1.14.8. Timer specifications in pulse period/pulse width measurement mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Up count • Counter value "0000_{16}" is transferred to reload register at measurement pulse's effective edge and the timer continues counting
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	• When measurement pulse's effective edge is input (Note 1) • When an overflow occurs. (Simultaneously, the timer Bi overflow flag changes to "1". Assume that the count start flag condition is "1" and then the timer Bi overflow flag becomes "1". If the timer Bi mode register has a write-access after next count cycle of the timer from the above condition, the timer Bi overflow flag becomes "0".)
TBi_{IN} pin function	Measurement pulse input
Read from timer	When timer Bi register is read, it indicates the reload register's content (measurement result) (Note 2)
Write to timer	Cannot be written to

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

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

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
			⊠			1	0

Symbol: TBiMR(i=1, 2) — Address: $039C_{16}$, $039D_{16}$ — When reset: $00XX0000_2$

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

Note: It is indeterminate when reset. Assume that the count start flag condition is "1" and then the timer Bi overflow flag becomes "1". If the timer Bi mode register has a write access after next count cycle of the timer from the above condition, the timer Bi overflow flag becomes "0". This flag cannot be set to "1" by software.

Figure 1.14.19. Timer Bi mode register in pulse period/pulse width measurement mode

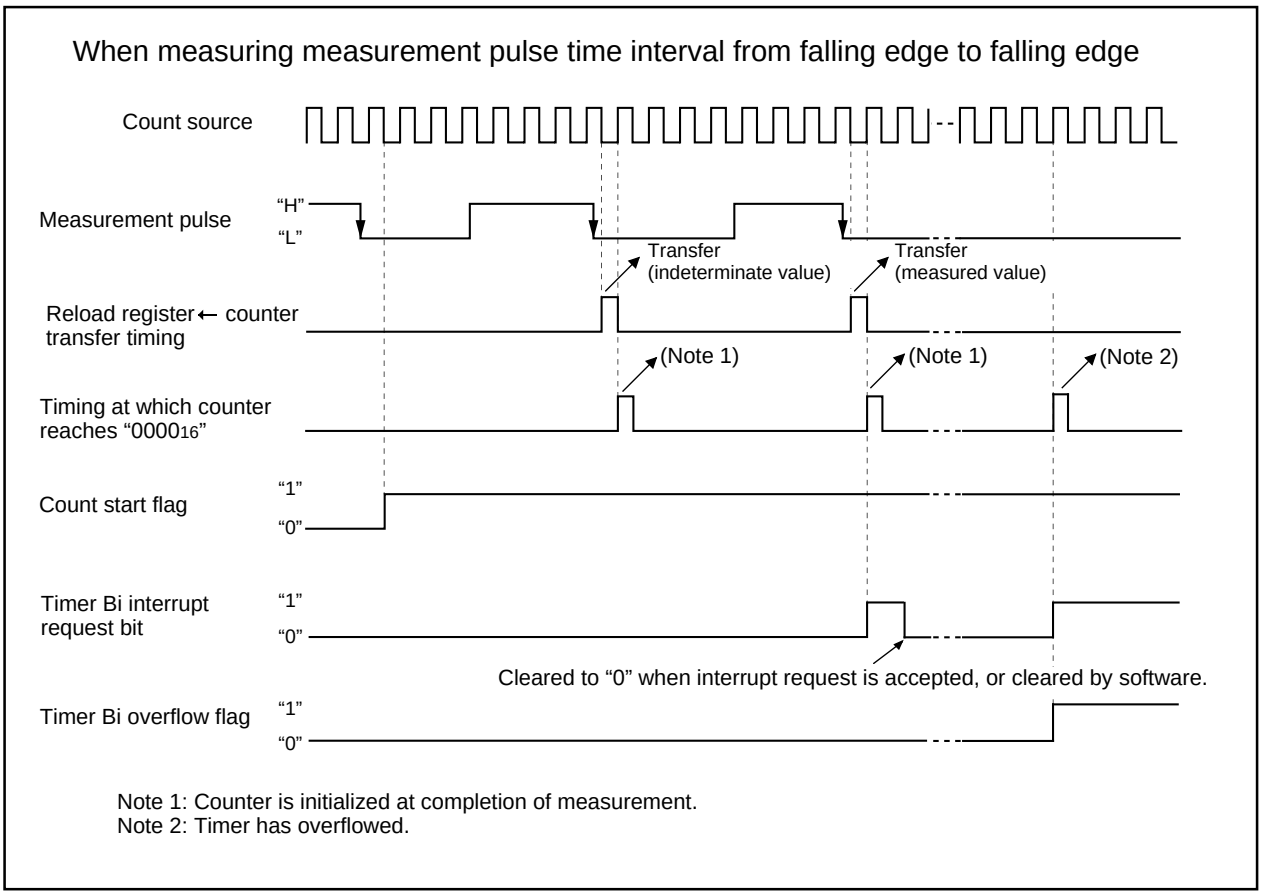


Figure 1.14.20. Operation timing when measuring a pulse period

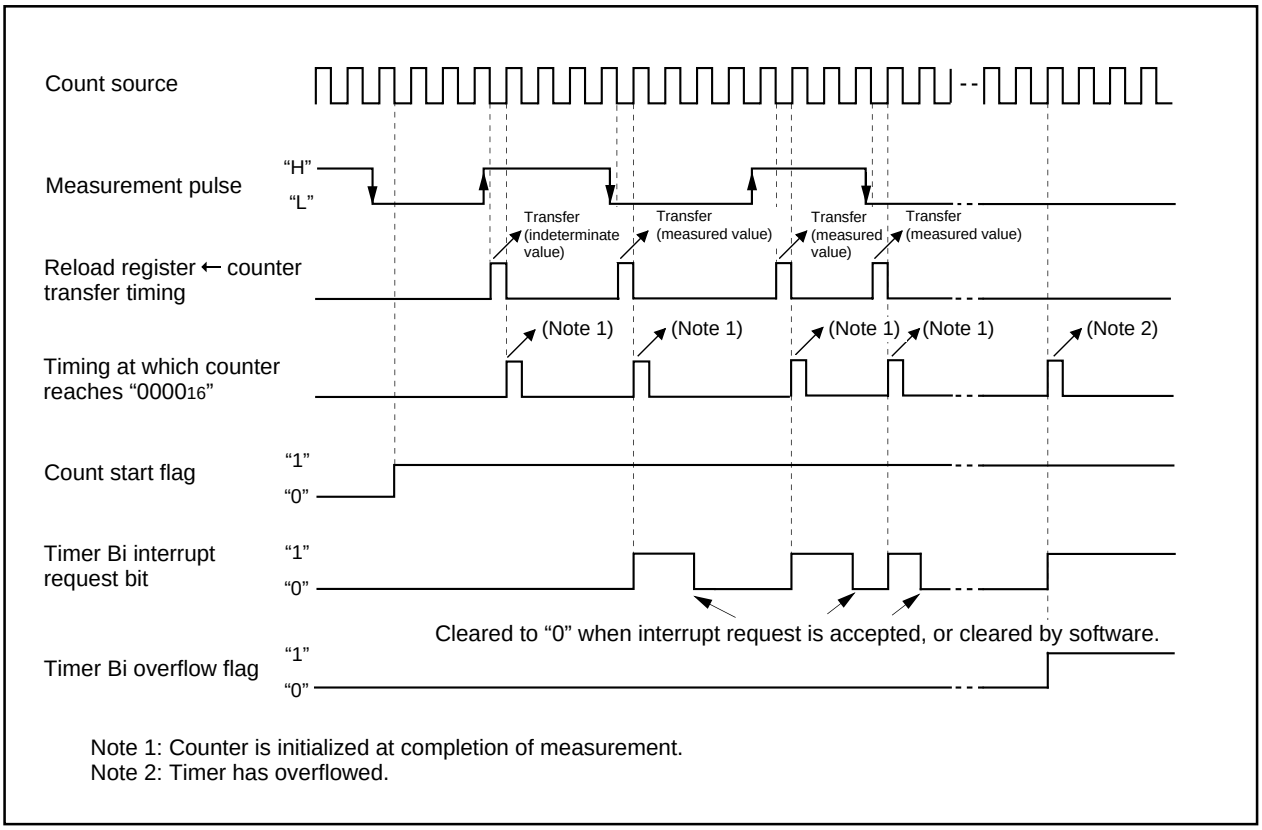


Figure 1.14.21. Operation timing when measuring a pulse width

Serial I/O

Serial I/O is configured as three channels: UART0, UART1, UART2.

UART0, UART1 and UART2 each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 1.16.1 shows the block diagram of UART0, UART1 and UART2. Figures 1.16.2 and 1.16.3 show the block diagram of the transmit/receive unit.

UARTi (i = 0 to 2) has two operation modes: a clock synchronous serial I/O mode and a clock asynchronous serial I/O mode (UART mode). The contents of the serial I/O mode select bits (bits 0 to 2 at addresses $03A0_{16}$, $03A8_{16}$ and 0378_{16}) determine whether UARTi is used as a clock synchronous serial I/O or as a UART. Although a few functions are different, UART0, UART1 and UART2 have almost the same functions. UART2, in particular, is used for the SIM interface with some extra settings added in clock-asynchronous serial I/O mode (Note). It also has the bus collision detection function that generates an interrupt request if the TxD pin and the RxD pin are different in level.

Table 1.16.1 shows the comparison of functions of UART0 through UART2, and Figures 1.16.4 to 1.16.9 show the registers related to UARTi.

Note: SIM : Subscriber Identity Module

Table 1.16.1. Comparison of functions of UART0 through UART2

Function	UART0	UART1	UART2
CLK polarity selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)
LSB first / MSB first selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 2)
Continuous receive mode selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)
Transfer clock output from multiple pins selection	Impossible	Possible (Note 1)	Impossible
Serial data logic switch	Impossible	Impossible	Possible (Note 4)
Sleep mode selection	Possible (Note 3)	Possible (Note 3)	Impossible
TxD, RxD I/O polarity switch	Impossible	Impossible	Possible
TxD, RxD port output format	CMOS output	CMOS output	N-channel open-drain output
Parity error signal output	Impossible	Impossible	Possible (Note 4)
Bus collision detection	Impossible	Impossible	Possible

Note 1: Only when clock synchronous serial I/O mode.
Note 2: Only when clock synchronous serial I/O mode and 8-bit UART mode.
Note 3: Only when UART mode.
Note 4: Using for SIM interface.

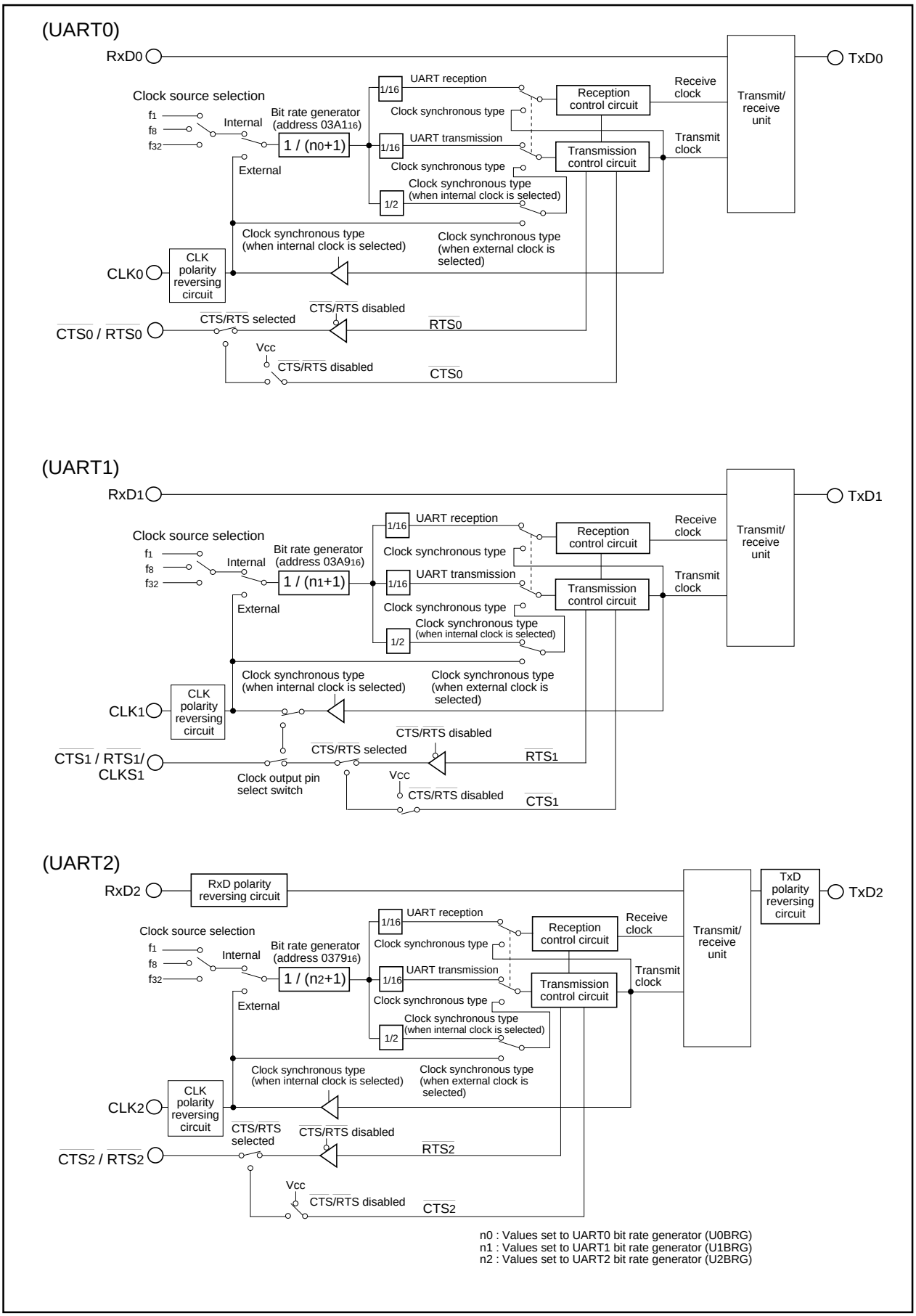


Figure 1.16.1. Block diagram of UARTi (i = 0 to 2)

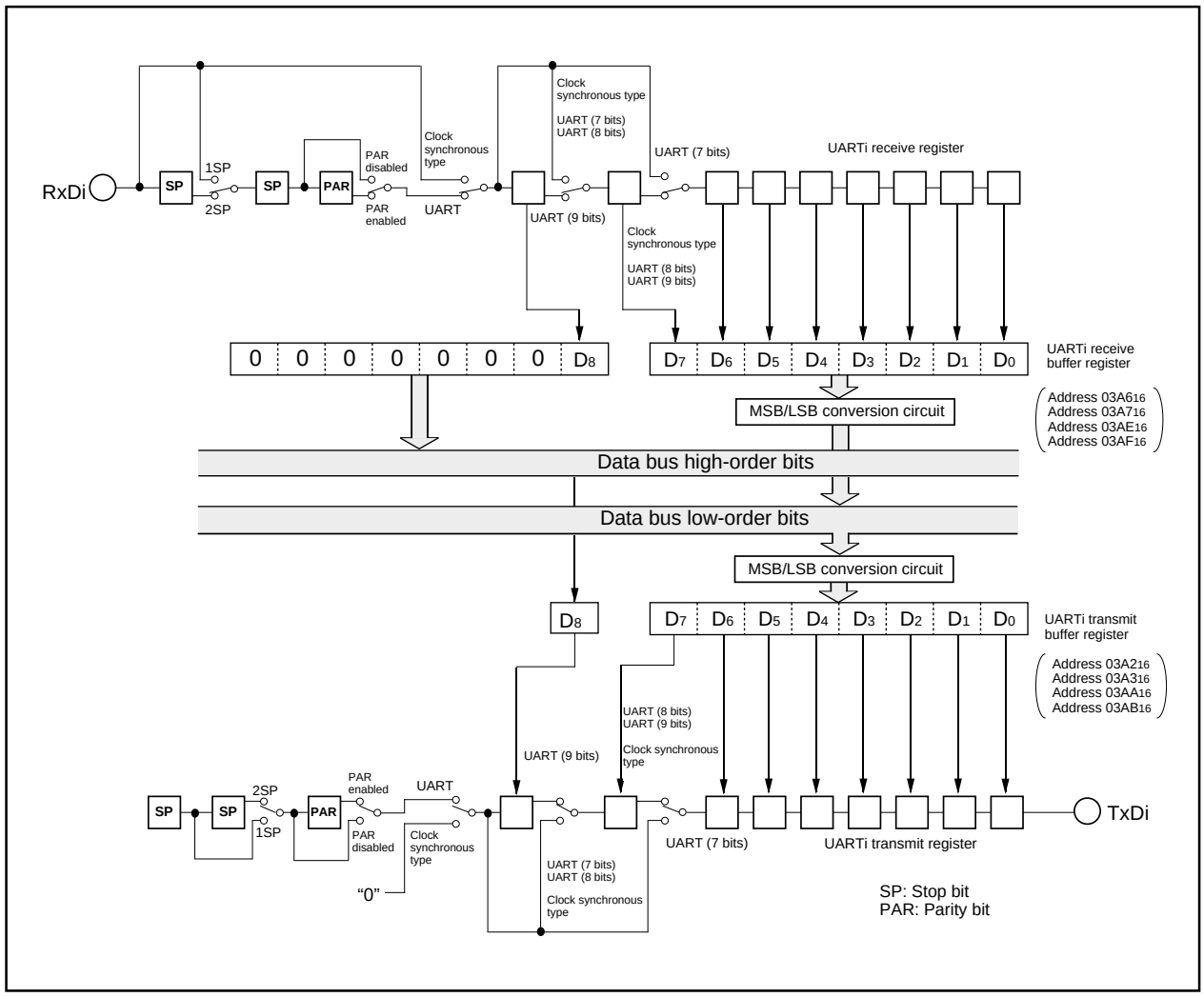


Figure 1.16.2. Block diagram of UARTi (i = 0, 1) transmit/receive unit

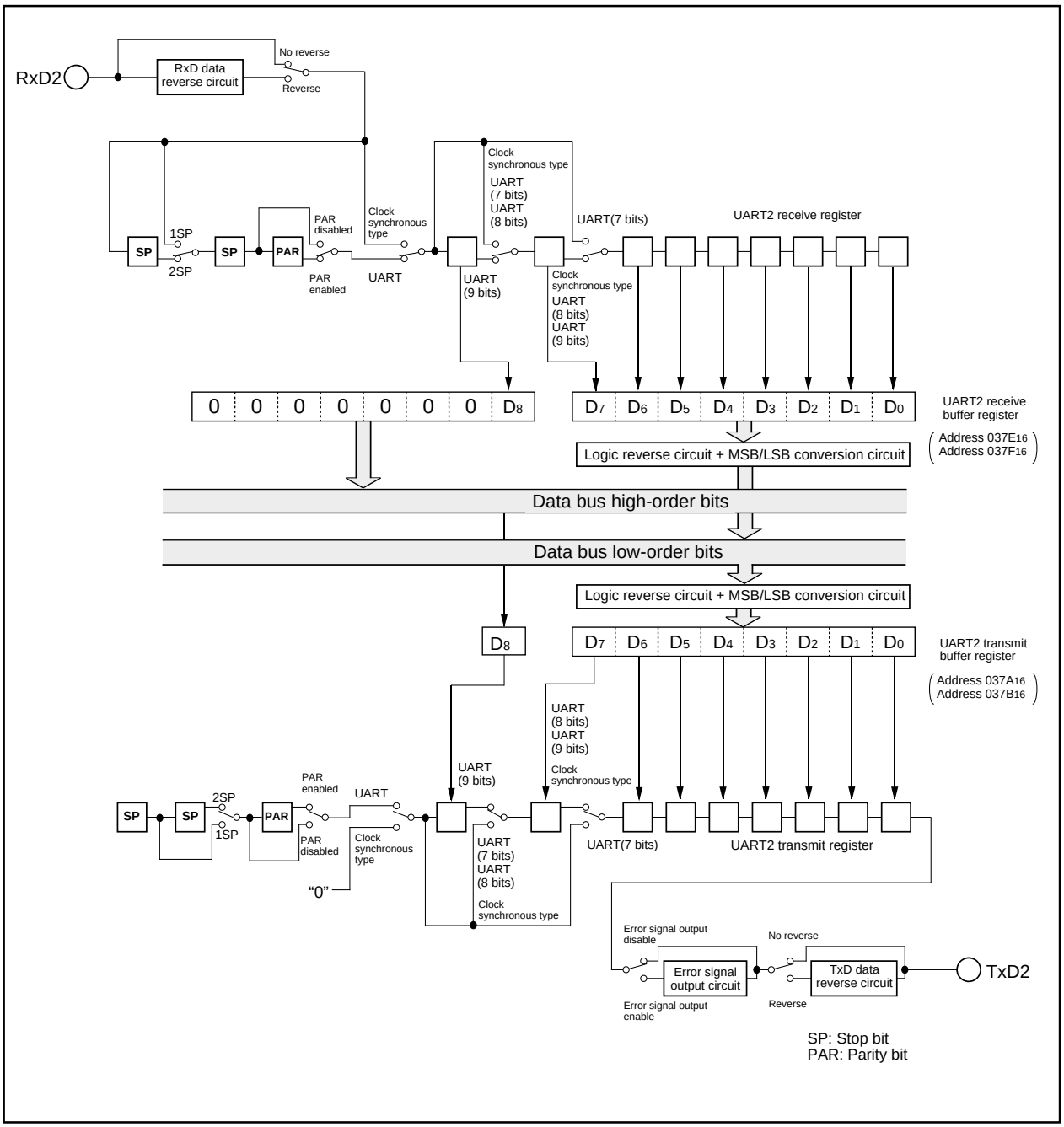
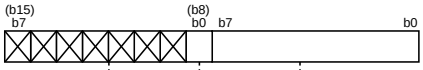


Figure 1.16.3. Block diagram of UART2 transmit/receive unit

UARTi transmit buffer register (Note)

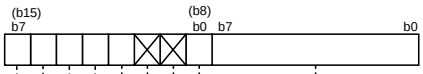


Symbol	Address	When reset
U0TB	$03A3_{16}$, $03A2_{16}$	Indeterminate
U1TB	$03AB_{16}$, $03AA_{16}$	Indeterminate
U2TB	$037B_{16}$, $037A_{16}$	Indeterminate

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

Note: Use MOV instruction to write to this register.

UARTi receive buffer register

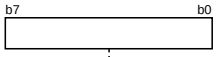


Symbol	Address	When reset
U0RB	$03A7_{16}$, $03A6_{16}$	Indeterminate
U1RB	$03AF_{16}$, $03AE_{16}$	Indeterminate
U2RB	$037F_{16}$, $037E_{16}$	Indeterminate

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 "0".				—	—
ABT	Arbitration lost detecting flag (Note 2)	0 : Not detected 1 : Detected	Invalid	○	○
OER	Overrun error flag (Note 1)	0 : No overrun error 1 : Overrun error found	0 : No overrun error 1 : Overrun error found	○	×
FER	Framing error flag (Note 1)	Invalid	0 : No framing error 1 : Framing error found	○	×
PER	Parity error flag (Note 1)	Invalid	0 : No parity error 1 : Parity error found	○	×
SUM	Error sum flag (Note 1)	Invalid	0 : No error 1 : Error found	○	×

Note 1: Bits 15 through 12 are set to "0" when the serial I/O mode select bit (bits 2 to 0 at addresses $03A0_{16}$, $03A8_{16}$ and 0378_{16}) are set to "000_2" or the receive enable bit is set to "0".
(Bit 15 is set to "0" when bits 14 to 12 all are set to "0".) Bits 14 and 13 are also set to "0" when the lower byte of the UARTi receive buffer register (addresses $03A6_{16}$, $03AE_{16}$ and $037E_{16}$) is read out.
Note 2: Arbitration lost detecting flag is allocated to U2RB and noting but "0" may be written. Nothing is assigned in bit 11 of U0RB and U1RB. When write, set "0". The value, if read, turns out to be "0".

UARTi bit rate generator (Note 1, 2)



Symbol	Address	When reset
U0BRG	$03A1_{16}$	Indeterminate
U1BRG	$03A9_{16}$	Indeterminate
U2BRG	0379_{16}	Indeterminate

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

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

Figure 1.16.4. Serial I/O-related registers (1)

UARTi transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0

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

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit	Must always be "001" b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Must not be set. 0 1 1 : Must not be set. 1 1 1 : Must not be set.	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 : Must not be set. 0 1 1 : Must not be set. 1 1 1 : Must not be set.	○	○
SMD1				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note)	0 : Internal clock 1 : External clock (Note)	○	○
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".

UART2 transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0

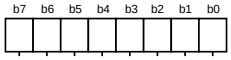
Symbol: U2MR — Address: 0378$_{16}$ — When reset: 00$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit	Must always be "001" b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : (Note 1) 0 1 1 : Must not be set. 1 1 1 : Must not be set.	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 : Must not be set. 0 1 1 : Must not be set. 1 1 1 : Must not be set.	○	○
SMD1				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note 2)	Must always be "0"	○	○
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	○	○
IOPOL	TxD, RxD I/O polarity reverse bit	0 : No reverse 1 : Reverse Usually set to "0"	0 : No reverse 1 : Reverse Usually set to "0"	○	○

Note 1: Bit 2 to bit 0 are set to "010$_2$" when I^{2}C mode is used.
Note 2: Set the corresponding port direction register to "0".

Figure 1.16.5. Serial I/O-related registers (2)

UARTi transmit/receive control register 0

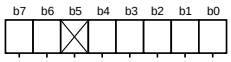


Symbol: UiC0(i=0,1) — Address: 03A4$_{16}$, 03AC$_{16}$ — When reset: 08$_{16}$

Bit symbol	Bit name	Function (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 : Must not be set.	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Must not be set.	○	○
CLK1				○	○
CRS	$\overline{CTS}/\overline{RTS}$ function select bit	Valid when bit 4 = "0" 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 2)	Valid when bit 4 = "0" 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 2)	○	○
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)	○	×
CRD	$\overline{CTS}/\overline{RTS}$ disable bit	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : CTS/RTS function disabled (P6$_0$ and P6$_4$ function as programmable I/O port)	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : CTS/RTS function disabled (P6$_0$ and P6$_4$ function as programmable I/O port)	○	○
NCH	Data output select bit	0 : TxDi pin is CMOS output 1 : TxDi pin is N-channel open-drain output	0: TxDi pin is CMOS output 1: TxDi pin is N-channel open-drain output	○	○
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Must always be "0"	○	○
UFORM	Transfer format select bit	0 : LSB first 1 : MSB first	Must always be "0"	○	○

Note 1: Set the corresponding port direction register to "0".
Note 2: The settings of the corresponding port register and port direction register are invalid.

UART2 transmit/receive control register 0



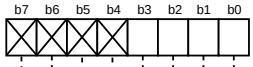
Symbol: U2C0 — Address: 037C$_{16}$ — When reset: 08$_{16}$

Bit symbol	Bit name	Function (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 : Must not be set.	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Must not be set.	○	○
CLK1				○	○
CRS	$\overline{CTS}/\overline{RTS}$ function select bit	Valid when bit 4 = "0" 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 2)	Valid when bit 4 = "0" 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 2)	○	○
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)	○	×
CRD	$\overline{CTS}/\overline{RTS}$ disable bit	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : CTS/RTS function disabled (P7$_3$ functions programmable I/O port)	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : CTS/RTS function disabled (P7$_3$ functions programmable I/O port)	○	○
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be "0".				–	–
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 (Note 3)	0 : LSB first 1 : MSB first	0 : LSB first 1 : MSB first	○	○

Note 1: Set the corresponding port direction register to "0".
Note 2: The settings of the corresponding port register and port direction register are invalid.
Note 3: Only clock synchronous serial I/O mode and 8-bit UART mode are valid.

Figure 1.16.6. Serial I/O-related registers (3)

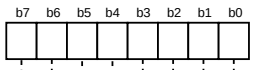
UARTi transmit/receive control register 1



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

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	○	×
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	○	○
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	○	×
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.				–	–

UART2 transmit/receive control register 1



Symbol: U2C1 | Address: 037D$_{16}$ | When reset: 02$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	○	×
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	○	○
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	○	×
U2IRS	UART2 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	○	○
U2RRM	UART2 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Must always be "0"	○	○
U2LCH	Data logic select bit	0 : No reverse 1 : Reverse	0 : No reverse 1 : Reverse	○	○
U2ERE	Error signal output enable bit	Must always be "0"	0 : Output disabled 1 : Output enabled	○	○

Figure 1.16.7. Serial I/O-related registers (4)

UART transmit/receive control register 2

b7	b6	b5	b4	b3	b2	b1	b0
X	0						

Symbol: UCON — Address: $03B0_{16}$ — When reset: $X0000000_2$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U1IRS	UART1 transmit interrupt cause select bit	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	Must always be “0”	○	○
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Must always be “0”	○	○
CLKMD0	CLK/CLKS select bit 0	Valid when bit 5 = “1” 0 : Clock output to CLK1 1 : Clock output to CLKS1	Invalid	○	○
CLKMD1	CLK/CLKS select bit 1 (Note)	0 : Normal mode (CLK output is CLK1 only) 1 : Transfer clock output from multiple pins function selected	Must always be “0”	○	○
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.				—	—

Note: 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 $03A8_{16}$) = “0”.

UART2 special mode register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: U2SMR — Address: 0377_{16} — When reset: 80_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
IICM	I^2C mode select bit	0 : Normal mode 1 : I^2C mode	Must always be “0”	○	○
ABC	Arbitration lost detecting flag control bit	0 : Update per bit 1 : Update per byte	Must always be “0”	○	○
BBS	Bus busy flag	0 : STOP condition detected 1 : START condition detected	Must always be “0”	○	○ (Note1)
LSYN	SCLL sync output enable bit	0 : Disabled 1 : Enabled	Must always be “0”	○	○
ABSCS	Bus collision detect sampling clock select bit	Must always be “0”	0 : Rising edge of transfer clock 1 : Underflow signal of timer A0	○	○
ACSE	Auto clear function select bit of transmit enable bit	Must always be “0”	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	○	○
SSS	Transmit start condition select bit	Must always be “0”	0 : Ordinary 1 : Falling edge of RxD_2	○	○
SDDS	SDA digital delay select bit (Note 2)	0 : Must always be “0” when not using I^2C mode 1 : Digital delay output is selected	Must always be “0”	○	○

Note 1: Nothing but “0” may be written.
Note 2: When not in I^2C mode, do not set this bit by writing a “1”. During normal mode, fix it to “0”. When this bit = “0”, UART2 special mode register 3 (U2SMR3 at address 0375_{16}) bits 7 to 5 (DL2 to DL0 = SDA digital delay setup bits) are initialized to “000”. Also, when SDDS = “0”, the U2SMR3 register cannot be read or written to.

Figure 1.16.8. Serial I/O-related registers (5)

UART2 special mode register 2 (I^2C bus exclusive use register)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: U2SMR2 | Address: 037616 | When reset: 0016

Bit symbol	Bit name	Function (I^2C bus exclusive use)	R	W
IICM2	I^2C mode select bit 2	Refer to Table 1.16.11	○	○
CSC	Clock-synchronous bit	0 : Disabled 1 : Enabled	○	○
SWC	SCL wait output bit	0 : Disabled 1 : Enabled	○	○
ALS	SDA output stop bit	0 : Disabled 1 : Enabled	○	○
STAC	UART2 initialization bit	0 : Disabled 1 : Enabled	○	○
SWC2	SCL wait output bit 2	0: UART2 clock 1: 0 output	○	○
SDHI	SDA output disable bit	0: Enabled 1: Disabled (high impedance)	○	○
SHTC	Start/stop condition control bit	1: Set this bit to “1” in I^2C mode (refer to Table 1.16.12)	○	○

UART2 special mode register 3 (I^2C bus exclusive use register)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: U2SMR3 | Address: 037516 | When reset: 0016

Bit symbol	Bit name	Function (I^2C bus exclusive use register)	R	W
		Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate. However, when SDDS = “1”, the value “0” is read out (Note 1)	—	—
DL0	SDA digital delay setup bit (Note 1, Note 2, Note 3)	b7 b6 b5 0 0 0 : Must not be set when using I^2C mode 0 0 1 : 1 to 2 cycle(s) of 1/f(XIN) 0 1 0 : 2 to 3 cycles of 1/f(XIN) 0 1 1 : 3 to 4 cycles of 1/f(XIN) 1 0 0 : 4 to 5 cycles of 1/f(XIN) 1 0 1 : 5 to 6 cycles of 1/f(XIN) 1 1 0 : 6 to 7 cycles of 1/f(XIN) 1 1 1 : 7 to 8 cycles of 1/f(XIN) Digital delay is selected	○	○
DL1			○	○
DL2			○	○

Note 1: This bit can be read or written to when UART2 special mode register (U2SMR at address 037716) bit 7 (SDDS: SDA digital delay select bit) = “1”. When the initial value of UART2 special mode register 3 (U2SMR3) is read after setting SDDS = “1”, the value is “0016”. When writing to UART2 special mode register 3 (U2SMR3) after setting SDDS = “1”, be sure to write 0's to bits 0–4. When SDDS = “0”, this register cannot be written to; when read, the value is indeterminate.

Note 2: These bits are initialized to “000” when SDDS = “0”. After a reset, these bits are set to “000”. However, because these bits can be read only when SDDS = “1”, the value read from these bits when SDDS = “0” is indeterminate.

Note 3: The amount of delay varies with the load on SCL and SDA pins. Also, when using an external clock, the amount of delay increases by about 200 ns, so be sure to take this into account when using the device.

Figure 1.16.9. Serial I/O-related registers (6)

(1) Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Tables 1.16.2 and 1.16.3 list the specifications of the clock synchronous serial I/O mode. Figure 1.16.10 shows the UARTi transmit/receive mode register.

Table 1.16.2. Specifications of clock synchronous serial I/O mode (1)

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• When internal clock is selected (bit 3 at addresses 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$ = "0") : fi/ 2(n+1) (Note 1) fi = f_1, f_8, f_{32} • When external clock is selected (bit 3 at addresses 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$ = "1") : Input from CLKi pin
Transmission/reception control	• $\overline{CTS}$ function, $\overline{RTS}$ function, $\overline{CTS}$ and $\overline{RTS}$ function invalid: selectable
Transmission start condition	• To start transmission, the following requirements must be met: – Transmit enable bit (bit 0 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = "1" – Transmit buffer empty flag (bit 1 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = "0" – When $\overline{CTS}$ function selected, $\overline{CTS}$ input level = "L" • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = "1": CLKi input level = "L"
Reception start condition	• To start reception, the following requirements must be met: – Receive enable bit (bit 2 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = "1" – Transmit enable bit (bit 0 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = "1" – Transmit buffer empty flag (bit 1 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = "0" • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = "1": CLKi input level = "L"
Interrupt request generation timing	• When transmitting – Transmit interrupt cause select bit (bits 0, 1 at address 03B0$_{16}$, bit 4 at address 037D$_{16}$) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed – Transmit interrupt cause select bit (bits 0, 1 at address 03B0$_{16}$, bit 4 at address 037D$_{16}$) = "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

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.

Table 1.16.3. Specifications of clock synchronous serial I/O mode (2)

Item	Specification
Select function	• CLK polarity selection Whether transmit data is output/input timing 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) UART1 transfer clock can be chosen by software to be output from one of the two pins set • Switching serial data logic (UART2) Whether to reverse data in writing to the transmission buffer register or reading the reception buffer register can be selected. • TxD, RxD I/O polarity reverse (UART2) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.

UARTi transmit/receive mode registers

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

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

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 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”.

UART2 transmit/receive mode register

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

Symbol: U2MR — Address: 0378$_{16}$ — When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note 2)	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
IOPOL	TxD, RxD I/O polarity reverse bit (Note 1)	0 : No reverse 1 : Reverse	○	○

Note 1: Usually set to “0”.
Note 2: Set the corresponding port direction register to “0”.

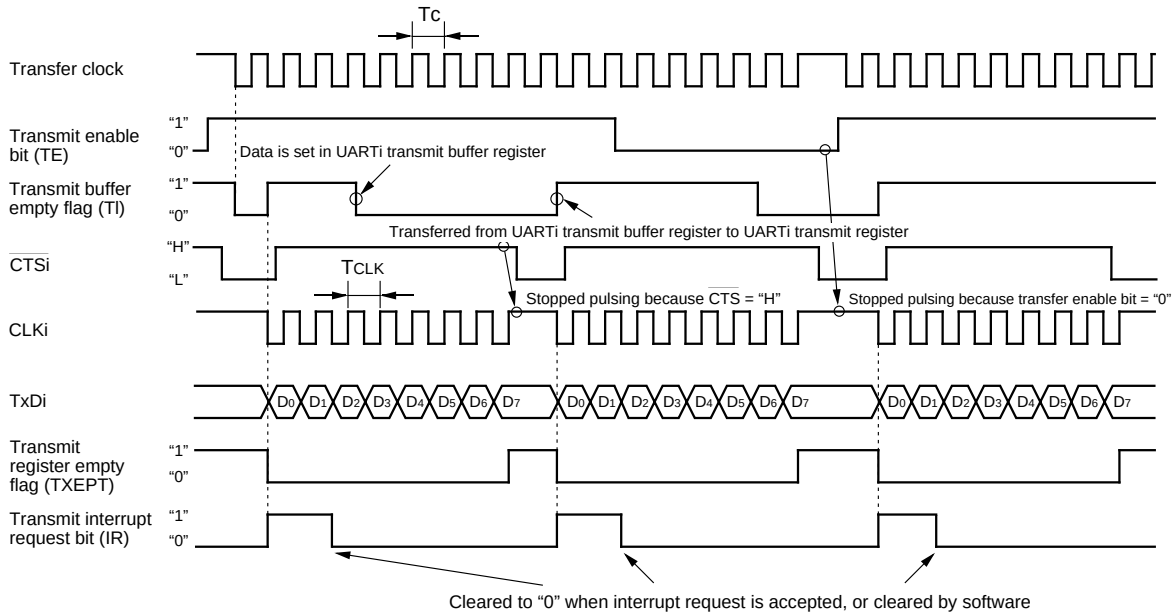
Figure 1.16.10. UARTi transmit/receive mode register in clock synchronous serial I/O mode

Table 1.16.4 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 function 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 1.16.4. Input/output pin functions in clock synchronous serial I/O mode
(when transfer clock output from multiple pins is not selected)

Pin name	Function	Method of selection
TxDi ($P6_3$, $P6_7$, $P7_0$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi ($P6_2$, $P6_6$, $P7_1$)	Serial data input	Port $P6_2$, $P6_6$ and $P7_1$ direction register (bits 2 and 6 at address $03EE_{16}$, bit 1 at address $03EF_{16}$)= “0” (Can be used as an input port when performing transmission only)
CLKi ($P6_1$, $P6_5$, $P7_2$)	Transfer clock output	Internal/external clock select bit (bit 3 at address $03A0_{16}$, $03A8_{16}$, 0378_{16}) = “0”
	Transfer clock input	Internal/external clock select bit (bit 3 at address $03A0_{16}$, $03A8_{16}$, 0378_{16}) = “1” Port $P6_1$, $P6_5$ and $P7_2$ direction register (bits 1 and 5 at address $03EE_{16}$, bit 2 at address $03EF_{16}$) = “0”
$\overline{\text{CTS}}$i/$\overline{\text{RTS}}$i ($P6_0$, $P6_4$, $P7_3$)	$\overline{\text{CTS}}$ input	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) =“0” $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “0” Port $P6_0$, $P6_4$ and $P7_3$ direction register (bits 0 and 4 at address $03EE_{16}$, bit 3 at address $03EF_{16}$) = “0”
	$\overline{\text{RTS}}$ output	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “0” $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “1”
	Programmable I/O port	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “1”

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

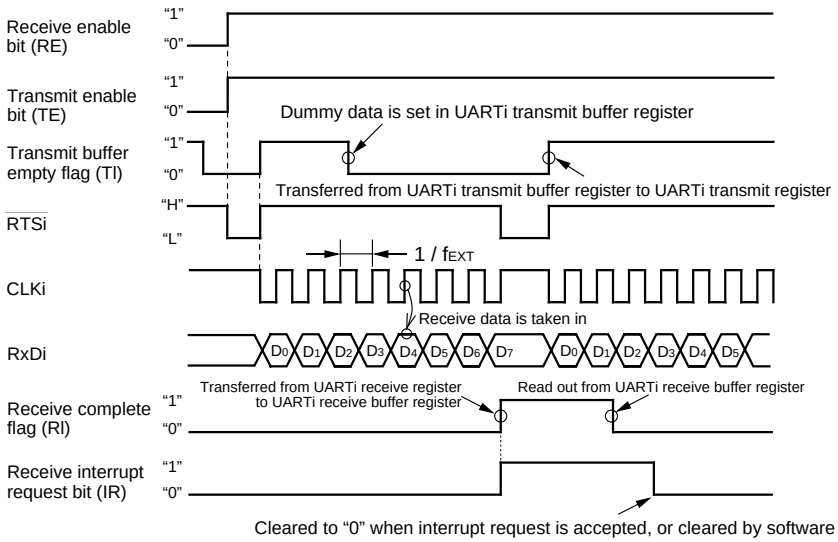


Shown in ( ) are bit symbols.

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

Tc = TCLK = 2(n + 1) / fi
fi: frequency of BRGi count source (f1, f8, f32)
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.
- RTS function is selected.
- CLK polarity select bit = "0".

fEXT: frequency of external clock

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

Figure 1.16.11. Typical transmit/receive timings in clock synchronous serial I/O mode

(a) Polarity select function

As shown in Figure 1.16.12, the CLK polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) allows selection of the polarity of the transfer clock.

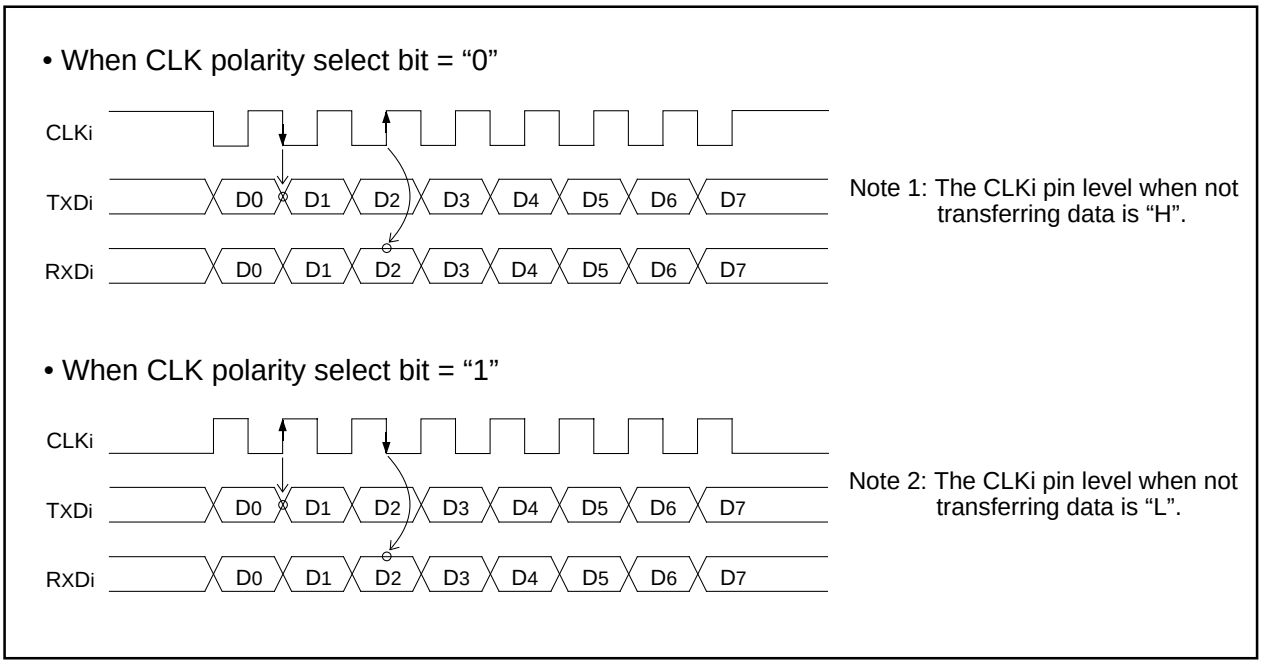


Figure 1.16.12. Polarity of transfer clock

(b) LSB first/MSB first select function

As shown in Figure 1.16.13, when the transfer format select bit (bit 7 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “0”, the transfer format is “LSB first”; when the bit = “1”, the transfer format is “MSB first”.

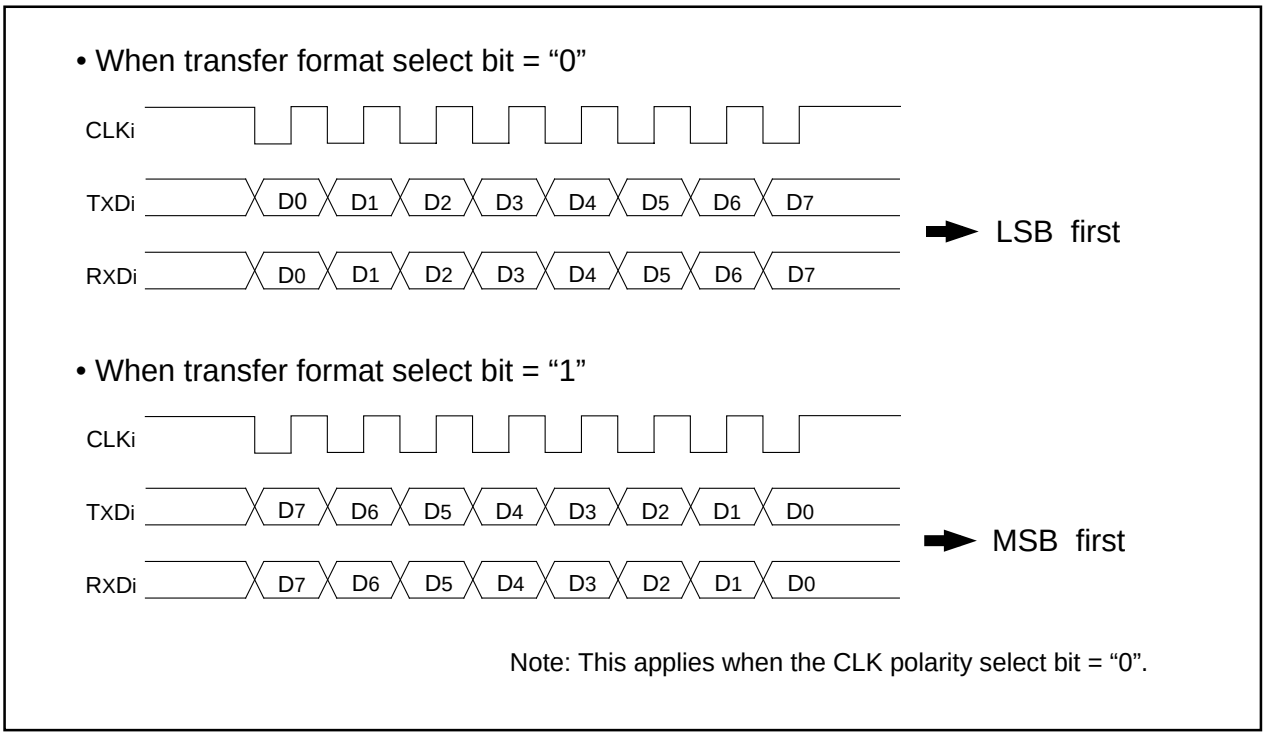


Figure 1.16.13. Transfer format

(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 03B0$_{16}$). (See Figure 1.16.14.) The multiple pins function is valid only when the internal clock is selected for UART1.

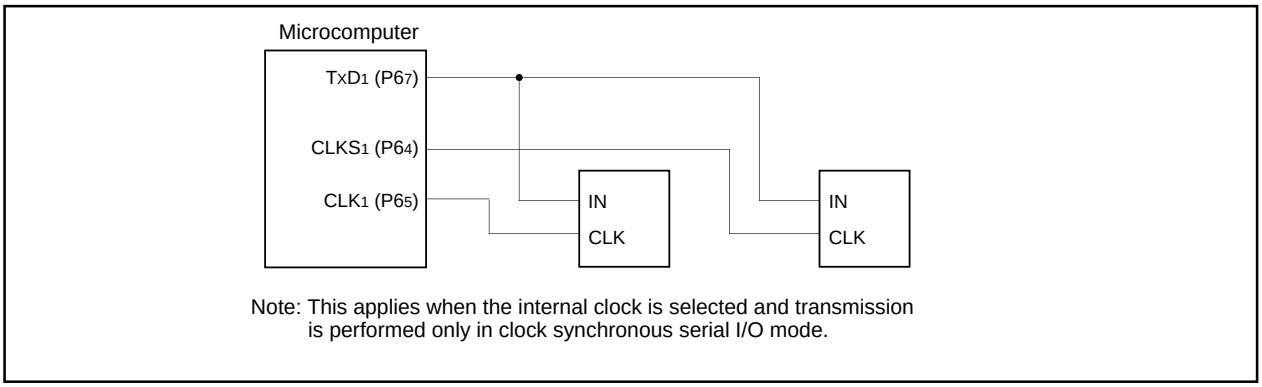


Figure 1.16.14. The transfer clock output from the multiple pins function usage

(d) Continuous receive mode

If the continuous receive mode enable bit (bits 2 and 3 at address 03B0$_{16}$, bit 5 at address 037D$_{16}$) 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) Serial data logic switch function (UART2)

When the data logic select bit (bit6 at address 037D$_{16}$) = "1", and writing to transmit buffer register or reading from receive buffer register, data is reversed. Figure 1.16.15 shows the example of serial data logic switch timing.

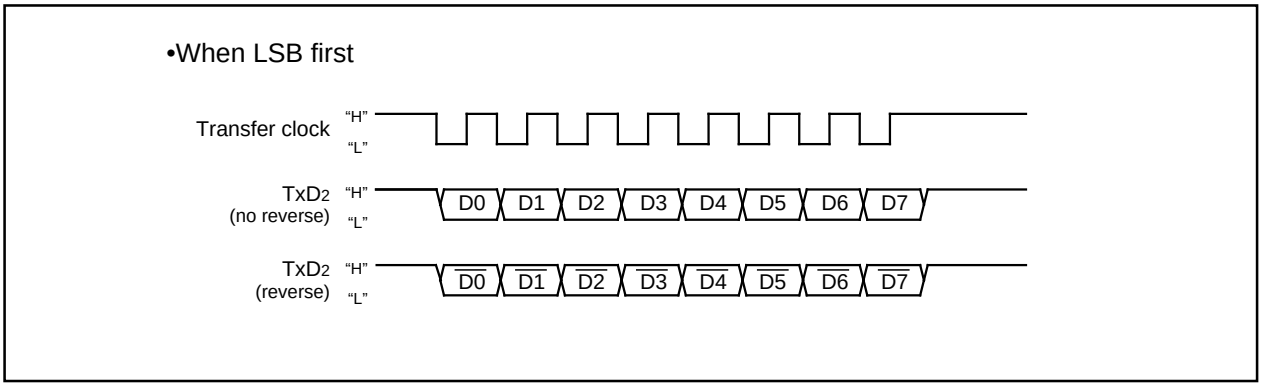


Figure 1.16.15. Serial data logic switch timing

(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. Tables 1.16.5 and 1.16.6 list the specifications of the UART mode. Figure 1.16.16 shows the UARTi transmit/receive mode register.

Table 1.16.5. Specifications of UART Mode (1)

Item	Specification
Transfer data format	• Character bit (transfer data): 7 bits, 8 bits, or 9 bits as selected • Start bit: 1 bit • Parity bit: Odd, even, or nothing as selected • Stop bit: 1 bit or 2 bits as selected
Transfer clock	• When internal clock is selected (bit 3 at addresses $03A0_{16}$, $03A8_{16}$, 0378_{16} = "0") : fi/16(n+1) (Note 1) fi = f_1, f_8, f_{32} • When external clock is selected (bit 3 at addresses $03A0_{16}$, $03A8_{16}$ ="1") : f_{EXT}/16(n+1) (Note 1) (Note 2) (Do not set external clock for UART2)
Transmission/reception control	• $\overline{CTS}$ function, $\overline{RTS}$ function, $\overline{CTS}$ and $\overline{RTS}$ function invalid: selectable
Transmission start condition	• To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 at addresses $03A5_{16}$, $03AD_{16}$, $037D_{16}$) = "1" - Transmit buffer empty flag (bit 1 at addresses $03A5_{16}$, $03AD_{16}$, $037D_{16}$) = "0" - When $\overline{CTS}$ function selected, $\overline{CTS}$ input level = "L"
Reception start condition	• To start reception, the following requirements must be met: - Receive enable bit (bit 2 at addresses $03A5_{16}$, $03AD_{16}$, $037D_{16}$) = "1" - Start bit detection
Interrupt request generation timing	• When transmitting - Transmit interrupt cause select bits (bits 0,1 at address $03B0_{16}$, bit4 at address $037D_{16}$) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed - Transmit interrupt cause select bits (bits 0, 1 at address $03B0_{16}$, bit4 at address $037D_{16}$) = "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

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

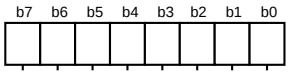
Note 2: f_{EXT} is input from the CLKi pin.

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.

Table 1.16.6. Specifications of UART Mode (2)

Item	Specification
Select function	• Sleep mode selection (UART0, UART1) This mode is used to transfer data to and from one of multiple slave microcomputers • Serial data logic switch (UART2) This function is reversing logic value of transferring data. Start bit, parity bit and stop bit are not reversed. • TxD, RxD I/O polarity switch (UART2) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.

UARTi transmit / receive mode registers

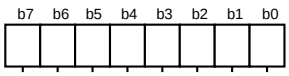


Symbol: UiMR(i=0,1) | Address: 03A0$_{16}$, 03A8$_{16}$ | When reset: 00$_{16}$

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

UART2 transmit / receive mode register



Symbol: U2MR | Address: 0378$_{16}$ | When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal / external clock select bit	Must always be “0”	○	○
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	○	○
IOPOL	TxD, RxD I/O polarity reverse bit (Note)	0 : No reverse 1 : Reverse	○	○

Note: Usually set to “0”.

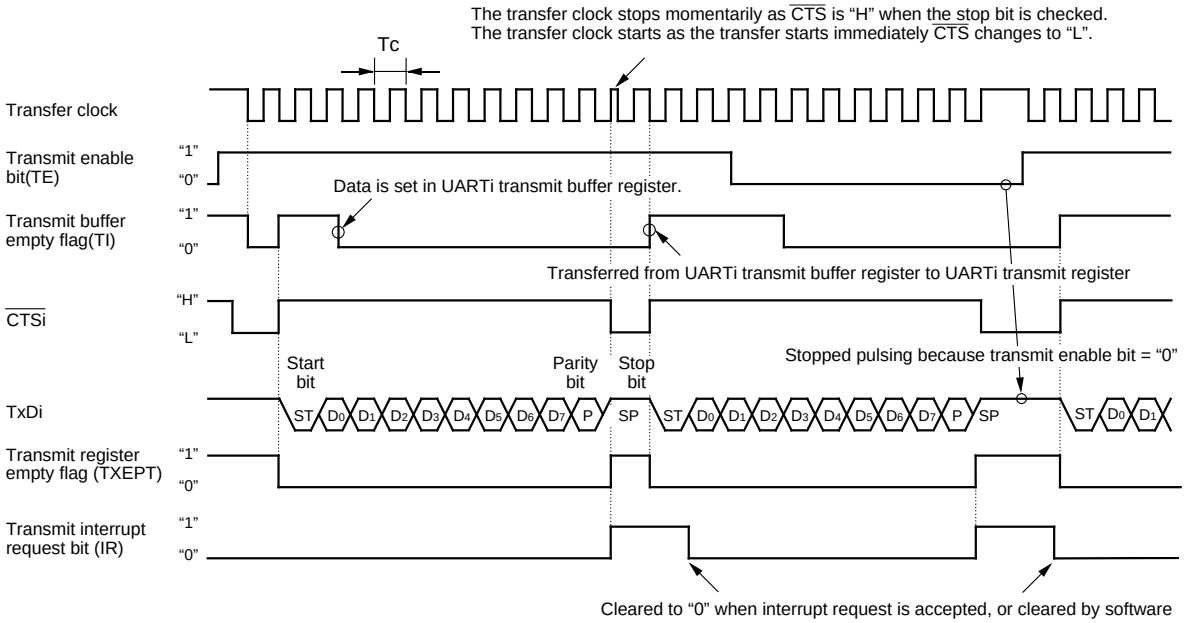
Figure 1.16.16. UARTi transmit/receive mode register in UART mode

Table 1.16.7 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 an “H”. (If the N-channel open-drain is selected, this pin is in floating state.)

Table 1.16.7. Input/output pin functions in UART mode

Pin name	Function	Method of selection
TxDi (P6$_3$, P6$_7$, P7$_0$)	Serial data output	
RxDi (P6$_2$, P6$_6$, P7$_1$)	Serial data input	Port P6$_2$, P6$_6$ and P7$_1$ direction register (bits 2 and 6 at address 03EE$_{16}$, bit 1 at address 03EF$_{16}$)= “0” (Can be used as an input port when performing transmission only)
CLKi (P6$_1$, P6$_5$, P7$_2$)	Programmable I/O port	Internal/external clock select bit (bit 3 at address 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$) = “0”
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0$_{16}$, 03A8$_{16}$) = “1” Port P6$_1$, P6$_5$ direction register (bits 1 and 5 at address 03EE$_{16}$) = “0” (Do not set external clock for UART2)
$\overline{\text{CTS}}$i/$\overline{\text{RTS}}$i (P6$_0$, P6$_4$, P7$_3$)	$\overline{\text{CTS}}$ input	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) =“0” $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit (bit 2 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “0” Port P6$_0$, P6$_4$ and P7$_3$ direction register (bits 0 and 4 at address 03EE$_{16}$, bit 3 at address 03EF$_{16}$) = “0”
	$\overline{\text{RTS}}$ output	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “0” $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit (bit 2 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “1”
	Programmable I/O port	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “1”

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

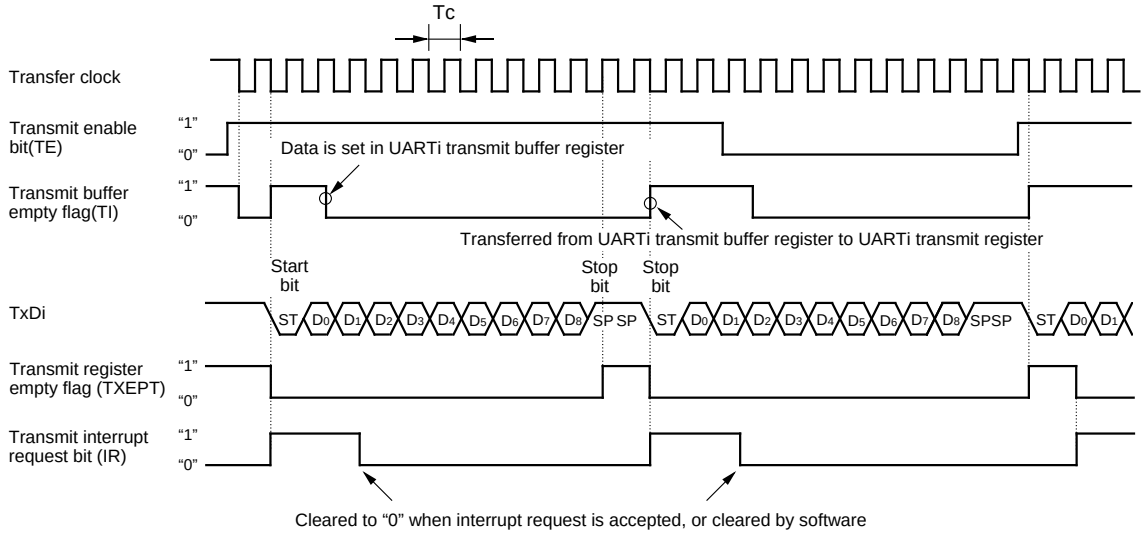


Shown in ( ) are bit symbols.

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

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1, f8, f32)
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)



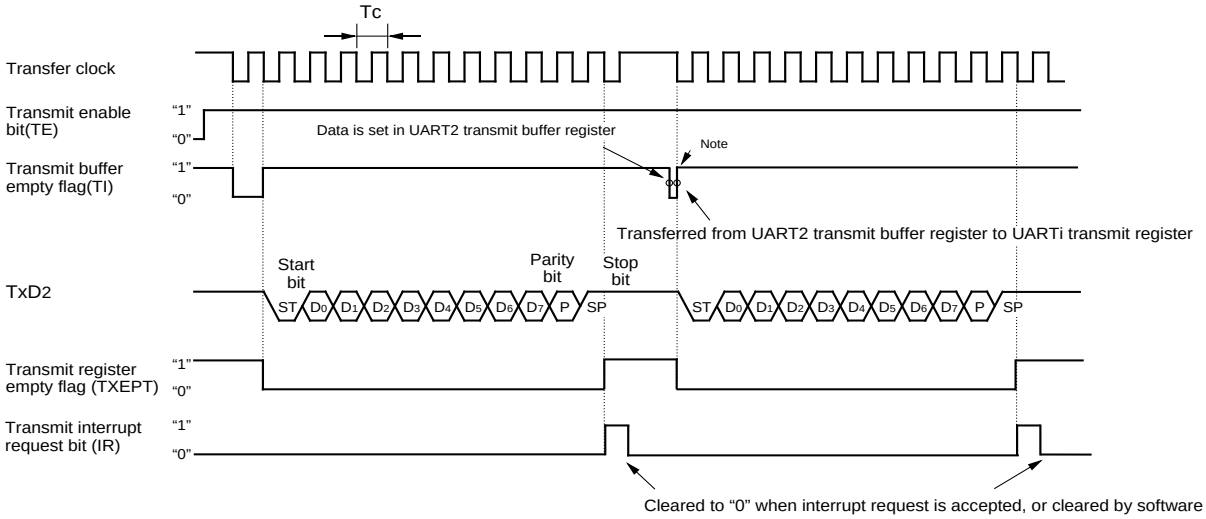
Shown in ( ) are bit symbols.

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

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1, f8, f32)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

Figure 1.16.17. Typical transmit timings in UART mode(UART0,UART1)

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



Shown in ( ) are bit symbols.

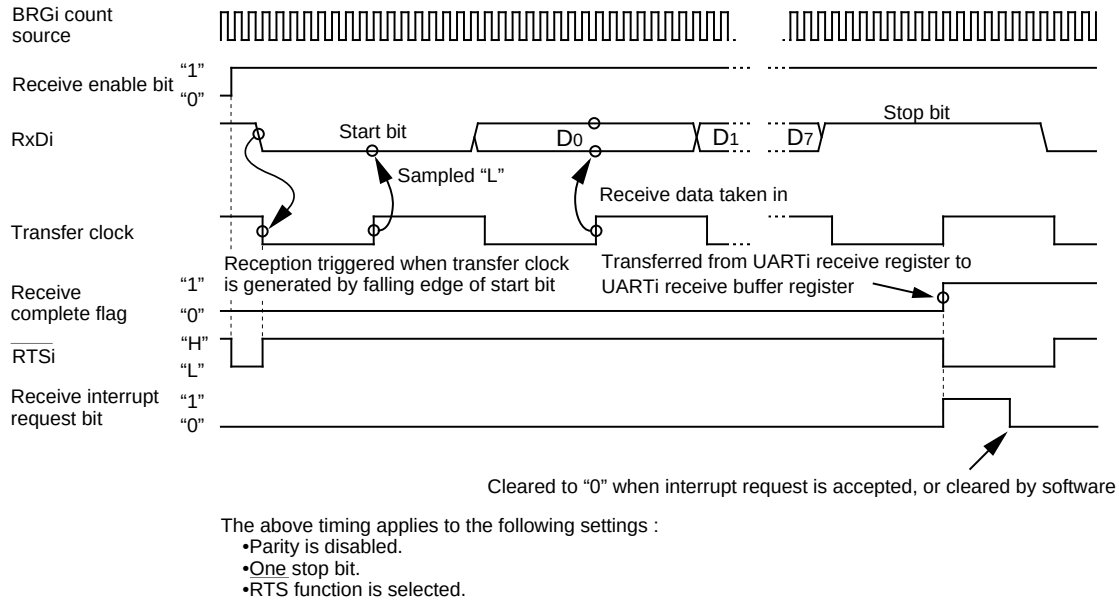
The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "1".

Tc = 16 (n + 1) / fi
fi : frequency of BRG2 count source (f_1, f_8, f_{32})
n : value set to BRG2

Note: The transmit is started with overflow timing of BRG after having written in a value at the transmit buffer in the above timing.

Figure 1.16.18. Typical transmit timings in UART mode(UART2)

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



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

Figure 1.16.19. Typical receive timing in UART mode

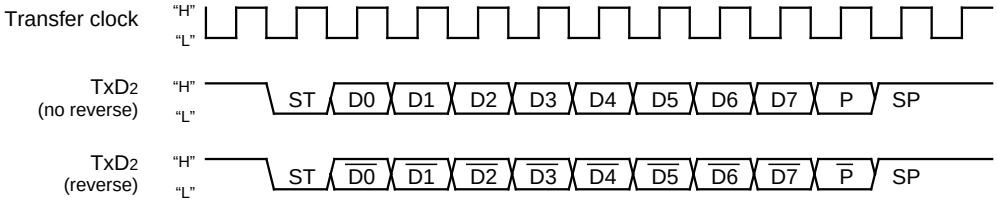
(a) Sleep mode (UART0, UART1)

This mode is used to transfer data between specific microcomputers among multiple microcomputers connected using UARTi. The sleep mode is selected when the sleep select bit (bit 7 at addresses $03A0_{16}$, $03A8_{16}$) is set to "1" during reception. In this mode, the unit performs receive operation when the MSB of the received data = "1" and does not perform receive operation when the MSB = "0".

(b) Function for switching serial data logic (UART2)

When the data logic select bit (bit 6 of address $037D_{16}$) is assigned "1", data is inverted in writing to the transmission buffer register or reading the reception buffer register. Figure 1.16.20 shows the example of timing for switching serial data logic.

• When LSB first, parity enabled, one stop bit



ST : Start bit
P : Even parity
SP : Stop bit

Figure 1.16.20. Timing for switching serial data logic

(c) TxD, RxD I/O polarity reverse function (UART2)

This function is to reverse TXD pin output and RXD pin input. The level of any data to be input or output (including the start bit, stop bit(s), and parity bit) is reversed. Set this function to “0” (not to reverse) for usual use.

(d) Bus collision detection function (UART2)

This function is to sample the output level of the TXD pin and the input level of the RXD pin at the rising edge of the transfer clock; if their values are different, then an interrupt request occurs. Figure 1.16.21 shows the example of detection timing of a bus collision (in UART mode).

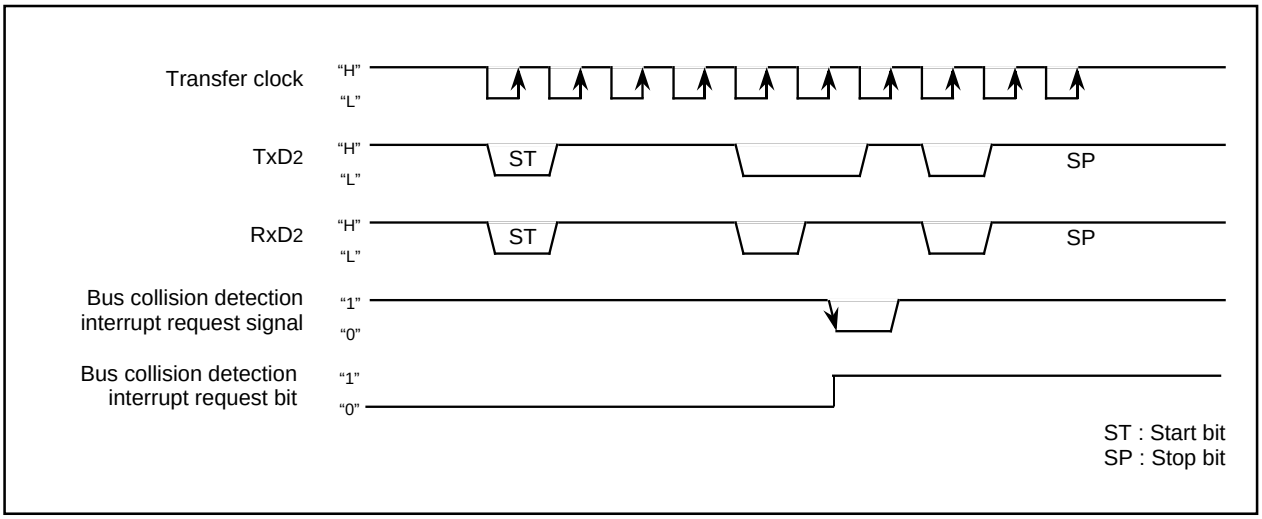


Figure 1.16.21. Detection timing of a bus collision (in UART mode)

(3) Clock-asynchronous serial I/O mode (used for the SIM interface)

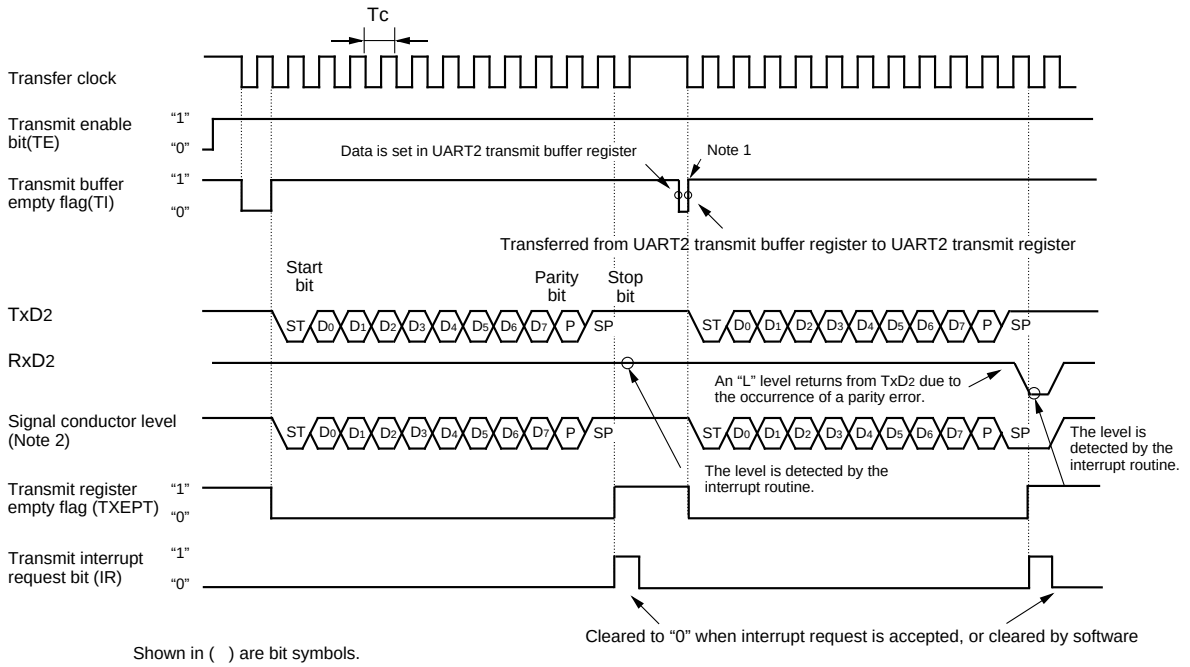
The SIM interface is used for connecting the microcomputer with a memory card or the like; adding some extra settings in UART2 clock-asynchronous serial I/O mode allows the user to effect this function. Table 1.16.8 shows the specifications of clock-asynchronous serial I/O mode (used for the SIM interface).

Table 1.16.8. Specifications of clock-asynchronous serial I/O mode (used for the SIM interface)

Item	Specification
Transfer data format	• Transfer data 8-bit UART mode (bit 2 through bit 0 of address 0378_{16} = "101_2") • One stop bit (bit 4 of address 0378_{16} = "0") • With the direct format chosen Set parity to "even" (bit 5 and bit 6 of address 0378_{16} = "1" and "1" respectively) Set data logic to "direct" (bit 6 of address $037D_{16}$ = "0"). Set transfer format to LSB (bit 7 of address $037C_{16}$ = "0"). • With the inverse format chosen Set parity to "odd" (bit 5 and bit 6 of address 0378_{16} = "0" and "1" respectively) Set data logic to "inverse" (bit 6 of address $037D_{16}$ = "1") Set transfer format to MSB (bit 7 of address $037C_{16}$ = "1")
Transfer clock	• With the internal clock chosen (bit 3 of address 0378_{16} = "0") : fi / 16 (n + 1) (Note 1) : fi=f_1, f_8, f_{32} (Do not set external clock)
Transmission / reception control	• Disable the $\overline{CTS}$ and $\overline{RTS}$ function (bit 4 of address $037C_{16}$ = "1")
Other settings	• The sleep mode select function is not available for UART2 • Set transmission interrupt factor to "transmission completed" (bit 4 of address $037D_{16}$ = "1")
Transmission start condition	• To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 of address $037D_{16}$) = "1" - Transmit buffer empty flag (bit 1 of address $037D_{16}$) = "0"
Reception start condition	• To start reception, the following requirements must be met: - Reception enable bit (bit 2 of address $037D_{16}$) = "1" - Detection of a start bit
Interrupt request generation timing	• When transmitting When data transmission from the UART2 transmit register is completed (bit 4 of address $037D_{16}$ = "1") • When receiving When data transfer from the UART2 receive register to the UART2 receive buffer register is completed
Error detection	• Overrun error (see the specifications of clock-asynchronous serial I/O) (Note 2) • Framing error (see the specifications of clock-asynchronous serial I/O) • Parity error (see the specifications of clock-asynchronous serial I/O) - On the reception side, an "L" level is output from the TxD_2 pin by use of the parity error signal output function (bit 7 of address $037D_{16}$ = "1") when a parity error is detected - On the transmission side, a parity error is detected by the level of input to the RxD_2 pin when a transmission interrupt occurs • The error sum flag (see the specifications of clock-asynchronous serial I/O)

Note 1: 'n' denotes the value 00_{16} to FF_{16} that is set to the UART2 bit rate generator.

Note 2: If an overrun error occurs, the UART2 receive buffer will have the next data written in. Note also that the UART2 receive interrupt request bit does not change.

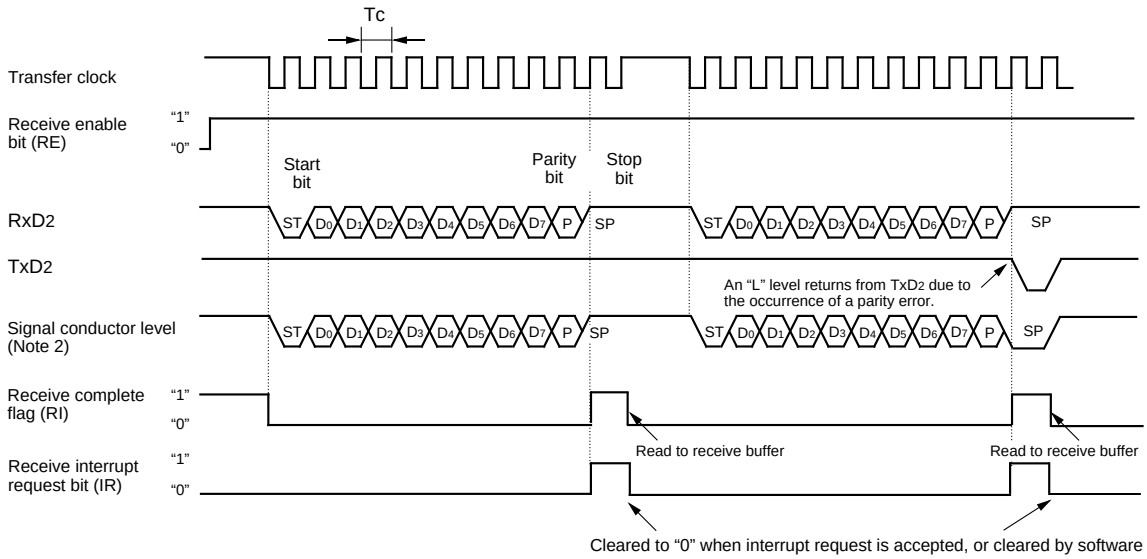


Shown in ( ) are bit symbols.

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
fi : frequency of BRG2 count source (f1, f8, f32)
n : value set to BRG2

Note 1: The transmit is started with overflow timing of BRG after having written in a value at the transmit buffer in the above timing.



Shown in ( ) are bit symbols.

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

Tc = 16 (n + 1) / fi
fi : frequency of BRG2 count source (f1, f8, f32)
n : value set to BRG2

Note 2: Equal in waveform because TxD2 and RxD2 are connected.

Figure 1.16.22. Typical transmit/receive timing in UART mode (used for the SIM interface)

(a) Function for outputting a parity error signal

During reception, with the error signal output enable bit (bit 7 of address 037D16) assigned “1”, you can output an “L” level from the TXD2 pin when a parity error is detected. And during transmission, comparing with the case in which the error signal output enable bit (bit 7 of address 037D16) is assigned “0”, the transmission completion interrupt occurs in the half cycle later of the transfer clock. Therefore parity error signals can be detected by a transmission completion interrupt program. Figure 1.16.23 shows the output timing of the parity error signal.

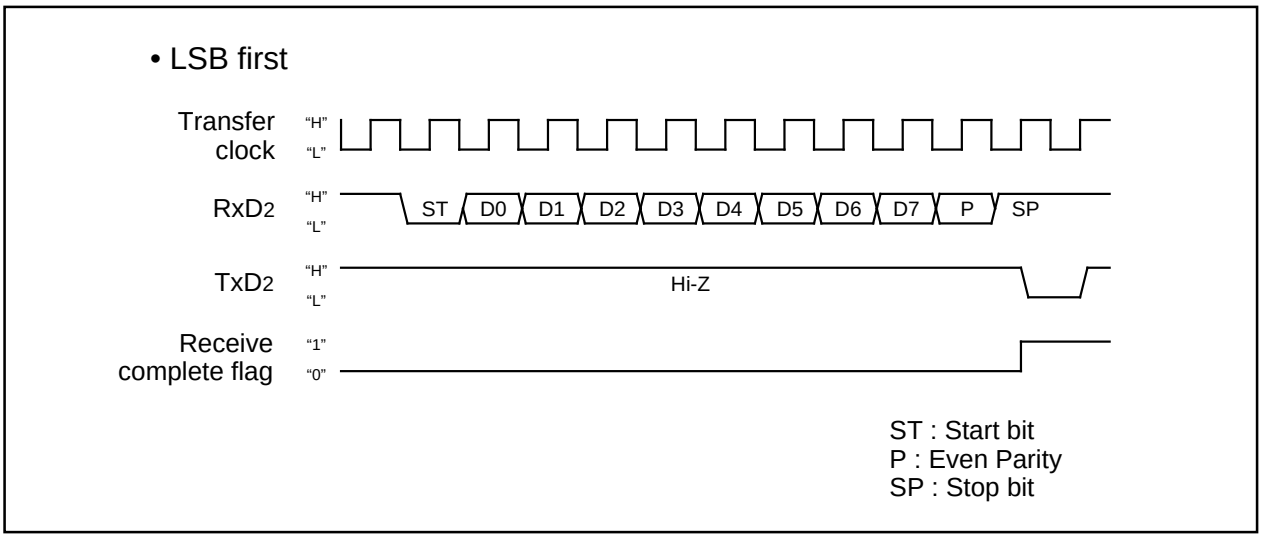


Figure 1.16.23. Output timing of the parity error signal

(b) Direct format/inverse format

Connecting the SIM card allows you to switch between direct format and inverse format. If you choose the direct format, D0 data is output from TxD2. If you choose the inverse format, D7 data is inverted and output from TxD2.

Figure 1.16.24 shows the SIM interface format.

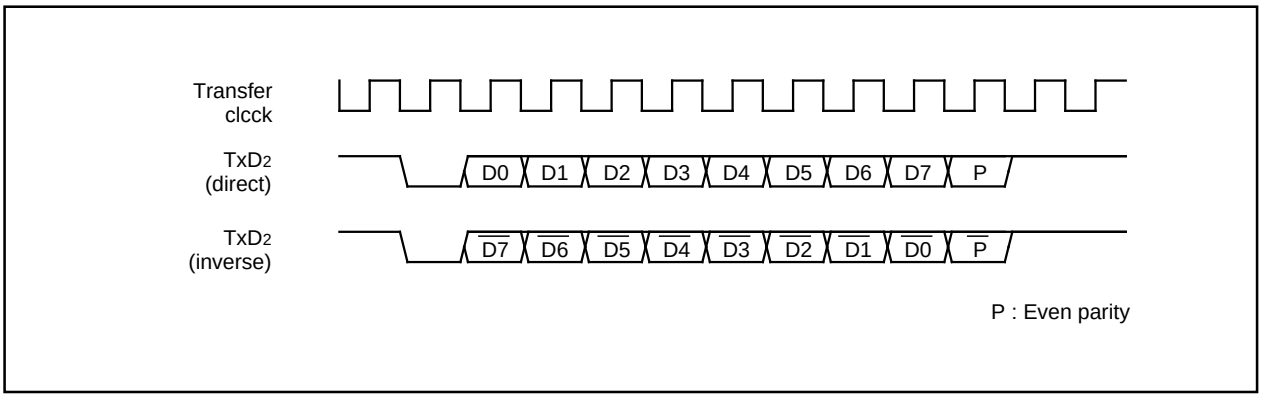


Figure 1.16.24. SIM interface format

Figure 1.16.25 shows the example of connecting the SIM interface. Connect TxD2 and RxD2 and apply pull-up.

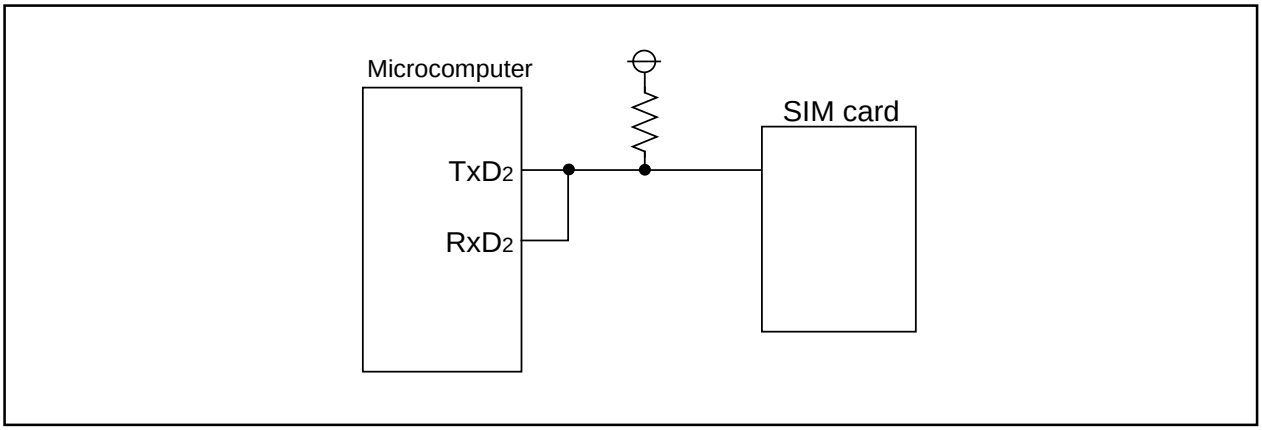


Figure 1.16.25. Connecting the SIM interface

UART2 Special Mode Register

The UART2 special mode register (address 037716) is used to control UART2 in various ways.

Figure 1.16.26 shows the UART2 special mode register.

Bit 0 of the UART2 special mode register (037716) is used as the I2C mode select bit.

Setting "1" in the I2C mode select bit (bit 0) goes the circuit to achieve the I2C bus (simplified I2C bus) interface effective.

Table 1.16.9 shows the relation between the I2C mode select bit and respective control workings.

Since this function uses clock-synchronous serial I/O mode, set this bit to "0" in UART mode.

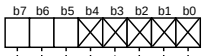
UART2 special mode register

Symbol U2SMR | Address 037716 | When reset 8016

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
IICM	I2C mode select bit	0 : Normal mode 1 : I2C mode	Must always be "0"	○	○
ABC	Arbitration lost detecting flag control bit	0 : Update per bit 1 : Update per byte	Must always be "0"	○	○
BBS	Bus busy flag	0 : STOP condition detected 1 : START condition detected	Must always be "0"	○	○ (Note1)
LSYN	SCL L sync output enable bit	0 : Disabled 1 : Enabled	Must always be "0"	○	○
ABSCS	Bus collision detect sampling clock select bit	Must always be "0"	0 : Rising edge of transfer clock 1 : Underflow signal of timer A0	○	○
ACSE	Auto clear function select bit of transmit enable bit	Must always be "0"	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	○	○
SSS	Transmit start condition select bit	Must always be "0"	0 : Ordinary 1 : Falling edge of RxD2	○	○
SDDS	SDA digital delay select bit (Note 2)	0 : Must always be "0" when not using I2C mode 1 : Digital delay output is selected	Must always be "0"	○	○

Note 1: Nothing but "0" may be written.
Note 2: When not in I2C mode, do not set this bit by writing a "1". During normal mode, fix it to "0". When this bit = "0", UART2 special mode register 3 (U2SMR3 at address 037516) bits 7 to 5 (DL2 to DL0 = SDA digital delay setup bits) are initialized to "000". Also, when SDDS = "0", the U2SMR3 register cannot be read or written to.

UART2 special mode register 3 (I2C bus exclusive use register)



Symbol U2SMR3 | Address 037516 | When reset 0016

Bit symbol	Bit name	Function (I2C bus exclusive use register)	R	W
		Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate. However, when SDDS = "1", the value "0" is read out (Note 1)	—	—
DL0	SDA digital delay setup bit (Note 1, Note 2, Note 3)	b7 b6 b5 0 0 0 : Must not be set when using I2C mode 0 0 1 : 1 to 2 cycle(s) of 1/f(XIN) 0 1 0 : 2 to 3 cycles of 1/f(XIN) 0 1 1 : 3 to 4 cycles of 1/f(XIN) 1 0 0 : 4 to 5 cycles of 1/f(XIN) 1 0 1 : 5 to 6 cycles of 1/f(XIN) 1 1 0 : 6 to 7 cycles of 1/f(XIN) 1 1 1 : 7 to 8 cycles of 1/f(XIN) Digital delay is selected	○	○
DL1			○	○
DL2			○	○

Note 1: This bit can be read or written to when UART2 special mode register (U2SMR at address 037716) bit 7 (SDDS: SDA digital delay select bit) = "1". When the initial value of UART2 special mode register 3 (U2SMR3) is read after setting SDDS = "1", the value is "0016". When writing to UART2 special mode register 3 (U2SMR3) after setting SDDS = "1", be sure to write 0's to bits 0–4. When SDDS = "0", this register cannot be written to; when read, the value is indeterminate.
Note 2: These bits are initialized to "000" when SDDS = "0". After a reset, these bits are set to "000". However, because these bits can be read only when SDDS = "1", the value read from these bits when SDDS = "0" is indeterminate.
Note 3: The amount of delay varies with the load on SCL and SDA pins. Also, when using an external clock, the amount of delay increases by about 200 ns, so be sure to take this into account when using the device.

Figure 1.16.26. UART2 special mode register

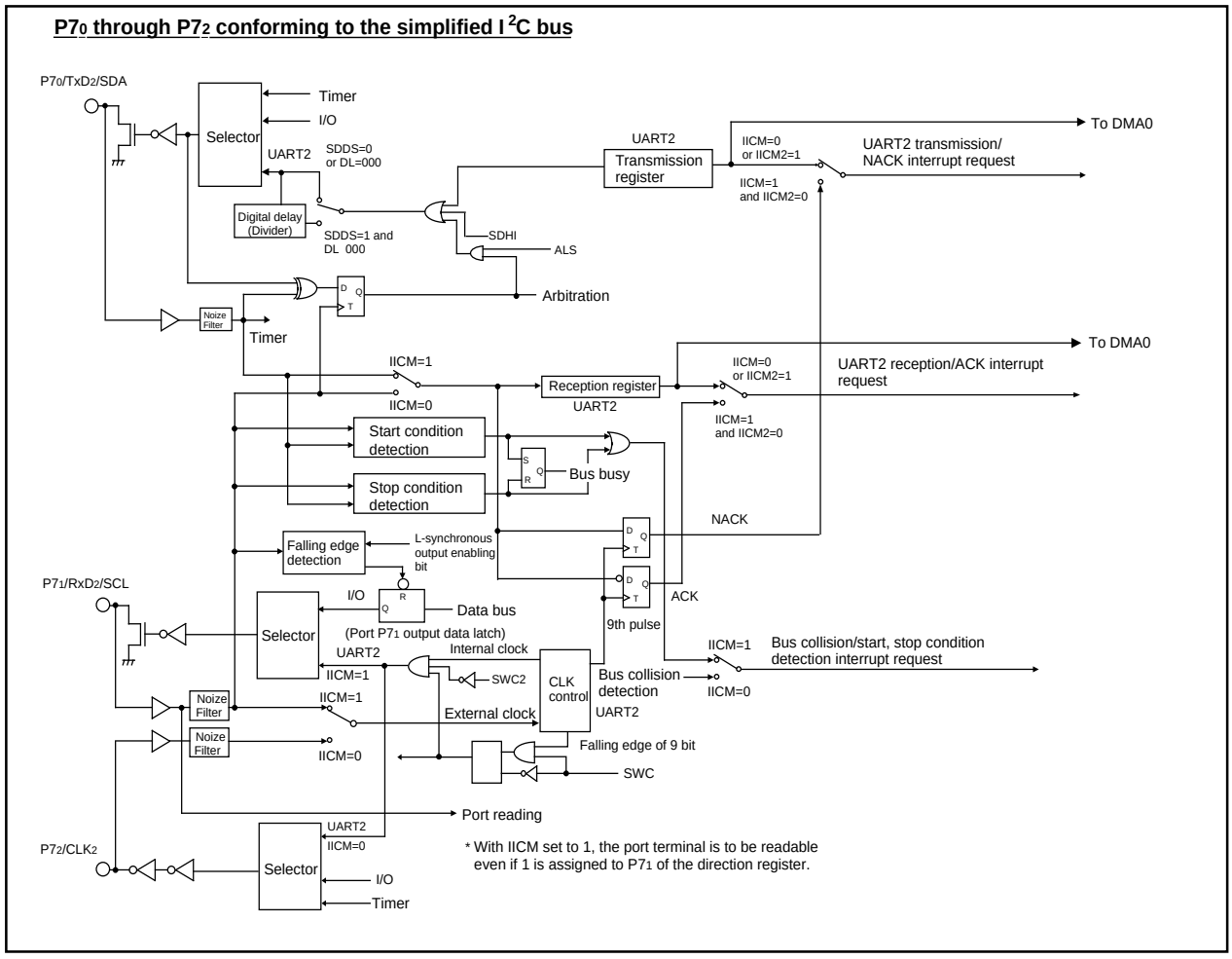


Figure 1.16.27. Functional block diagram for I²C mode

Table 1.16.9. Features in I²C mode

	Function	Normal mode	I²C mode (Note 1)
1	Factor of interrupt number 10 (Note 2)	Bus collision detection	Start condition detection or stop condition detection
2	Factor of interrupt number 15 (Note 2)	UART2 transmission	No acknowledgment detection (NACK)
3	Factor of interrupt number 16 (Note 2)	UART2 reception	Acknowledgment detection (ACK)
4	UART2 transmission output delay	Not delayed	Delayed (digital delay)
5	P70 at the time when UART2 is in use	TxD2 (output)	SDA (input/output) (Note 3)
6	P71 at the time when UART2 is in use	RxD2 (input)	SCL (input/output)
7	P72 at the time when UART2 is in use	CLK2	P72
8	Noise filter width	15ns	200ns
9	Reading P71	Reading the terminal when 0 is assigned to the direction register	Reading the terminal regardless of the value of the direction register
10	Initial value of UART2 output	H level (when 0 is assigned to the CLK polarity select bit)	The value set in latch P70 when the port is selected

Note 1: Make the settings given below when I²C mode is in use.
Set 0 1 0 in bits 2, 1, 0 of the UART2 transmission/reception mode register.
Disable the RTS/CTS function. Choose the MSB First function.
Note 2: Follow the steps given below to switch from a factor to another.
1. Disable the interrupt of the corresponding number.
2. Switch from a factor to another.
3. Reset the interrupt request flag of the corresponding number.
4. Set an interrupt level of the corresponding number.
Note 3: Set an initial value of SDA transmission output when serial I/O is invalid.

Figure 1.16.27 shows the functional block diagram for I^2C mode. Setting "1" in the I^2C mode select bit (IICM) causes ports $P7_0$, $P7_1$, and $P7_2$ to work as data transmission-reception terminal SDA, clock input-output terminal SCL, and port $P7_2$ respectively. A delay circuit is added to the SDA transmission output, so the SDA output changes after SCL fully goes to "L". The amount of delay can be selected in the range of 2 cycles to 8 cycles of f1 using UART2 special mode register 3 (at address 0375_{16}). Delay circuit select conditions are shown in Table 1.16.10.

Table 1.16.10. Delay circuit select conditions

	Register value			Contents
	IICM	SDDS	DL	
Digital delay is selected	1	1	001 to 111	Digital delay is added
No delay	0	0	(000)	When IICM = "0", no delay circuit is selected. When IICM = "0", however, always make sure SDDS = "0".

An attempt to read Port $P7_1$ (SCL) results in getting the terminal's level regardless of the content of the port direction register. The initial value of SDA transmission output goes to the value set in port $P7_0$. The interrupt factors of the bus collision detection interrupt, UART2 transmission interrupt, and of UART2 reception interrupt turn to the start/stop condition detection interrupt, acknowledgment non-detection interrupt, and acknowledgment detection interrupt respectively.

The start condition detection interrupt refers to the interrupt that occurs when the falling edge of the SDA terminal ($P7_0$) is detected with the SCL terminal ($P7_1$) staying "H". The stop condition detection interrupt refers to the interrupt that occurs when the rising edge of the SDA terminal ($P7_0$) is detected with the SCL terminal ($P7_1$) staying "H". The bus busy flag (bit 2 of the UART2 special mode register) is set to "1" by the start condition detection, and set to "0" by the stop condition detection.

The acknowledgment non-detection interrupt refers to the interrupt that occurs when the SDA terminal level is detected still staying "H" at the rising edge of the 9th transmission clock. The acknowledgment detection interrupt refers to the interrupt that occurs when SDA terminal's level is detected already went to "L" at the 9th transmission clock.

Bit 1 of the UART2 special mode register (0377_{16}) is used as the arbitration lost detecting flag control bit. Arbitration means the act of detecting the nonconformity between transmission data and SDA terminal data at the timing of the SCL rising edge. This detecting flag is located at bit 11 of the UART2 reception buffer register ($037F_{16}$, $037E_{16}$), and "1" is set in this flag when nonconformity is detected. Use the arbitration lost detecting flag control bit to choose which way to use to update the flag, bit by bit or byte by byte. When setting this bit to "1" and updated the flag byte by byte if nonconformity is detected, the arbitration lost detecting flag is set to "1" at the falling edge of the 9th transmission clock.

If update the flag byte by byte, must judge and clear ("0") the arbitration lost detecting flag after completing the first byte acknowledge detect and before starting the next one byte transmission.

Bit 3 of the UART2 special mode register is used as SCL- and L-synchronous output enable bit. Setting this bit to "1" goes the $P7_1$ data register to "0" in synchronization with the SCL terminal level going to "L".

Some other functions added are explained here. Figure 1.16.28 shows their workings.

Bit 4 of the UART2 special mode register is used as the bus collision detect sampling clock select bit. The bus collision detect interrupt occurs when the RxD2 level and TxD2 level do not match, but the nonconformity is detected in synchronization with the rising edge of the transfer clock signal if the bit is set to “0”. If this bit is set to “1”, the nonconformity is detected at the timing of the overflow of timer A0 rather than at the rising edge of the transfer clock.

Bit 5 of the UART2 special mode register is used as the auto clear function select bit of transmit enable bit. Setting this bit to “1” automatically resets the transmit enable bit to “0” when “1” is set in the bus collision detect interrupt request bit (nonconformity).

Bit 6 of the UART2 special mode register is used as the transmit start condition select bit. Setting this bit to “1” starts the TxD transmission in synchronization with the falling edge of the RxD terminal.

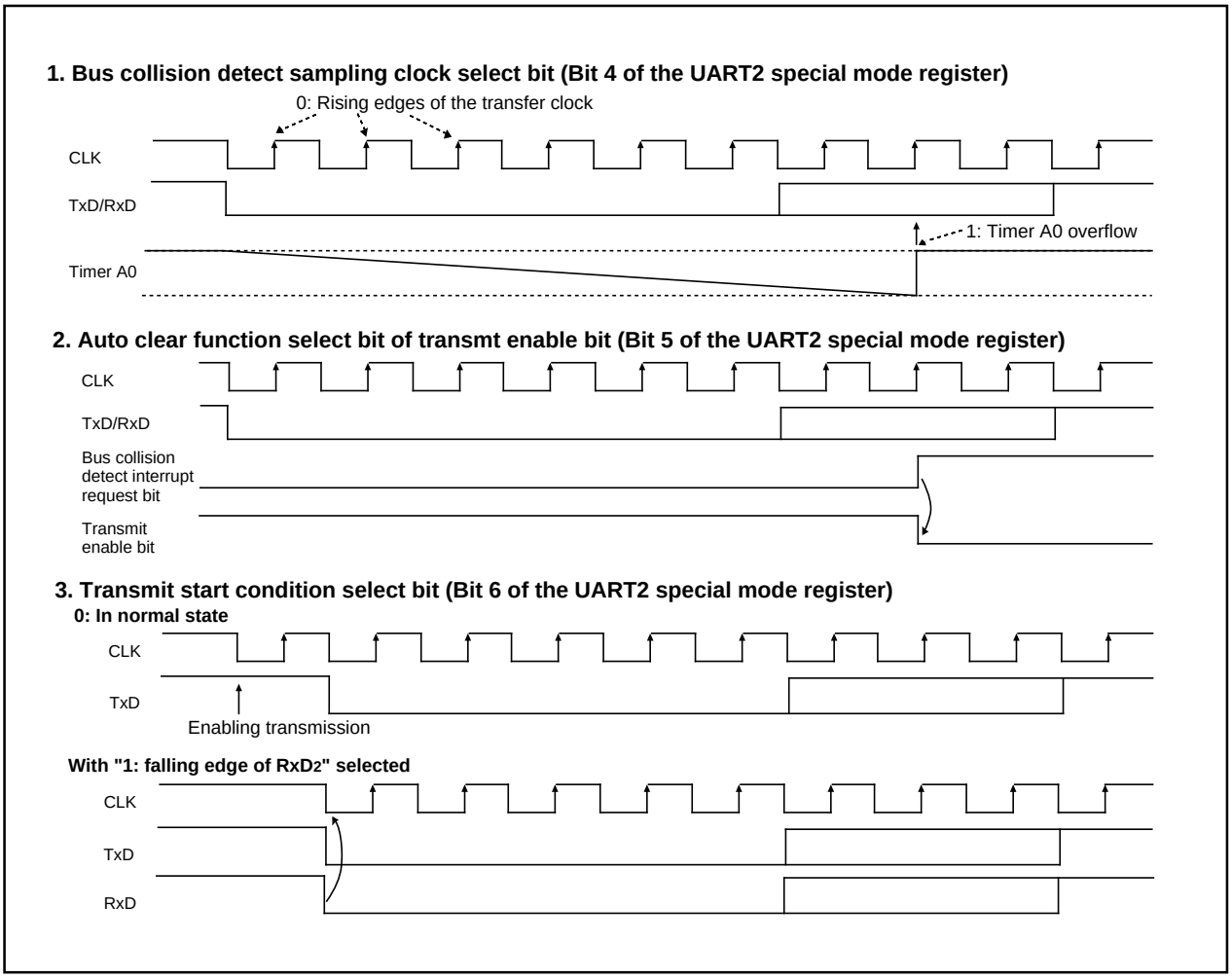
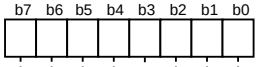


Figure 1.16.28. Some other functions added

UART2 Special Mode Register 2

UART2 special mode register 2 (address 037616) is used to further control UART2 in I2C mode. Figure 1.16.29 shows the UART2 special mode register 2.

UART2 special mode register 2 (I2C bus exclusive use register)



Symbol: U2SMR2 | Address: 037616 | When reset: 0016

Bit symbol	Bit name	Function	R	W
IICM2	I2C mode select bit 2	Refer to Table 1.16.11	○	○
CSC	Clock-synchronous bit	0 : Disabled 1 : Enabled	○	○
SWC	SCL wait output bit	0 : Disabled 1 : Enabled	○	○
ALS	SDA output stop bit	0 : Disabled 1 : Enabled	○	○
STAC	UART2 initialization bit	0 : Disabled 1 : Enabled	○	○
SWC2	SCL wait output bit 2	0: UART2 clock 1: 0 output	○	○
SDHI	SDA output disable bit	0: Enabled 1: Disabled (high impedance)	○	○
SHTC	Start/stop condition control bit	1: Set this bit to “1” in I2C mode (refer to Table 1.16.12)	○	○

Figure 1.16.29. UART2 special mode register 2

Bit 0 of the UART2 special mode register 2 (address 0376_{16}) is used as the I^2C mode select bit 2. Table 1.16.11 shows the types of control to be changed by I^2C mode select bit 2 when the I^2C mode select bit is set to "1". Table 1.16.12 shows the timing characteristics of detecting the start condition and the stop condition. Set the start/stop condition control bit (bit 7 of UART2 special mode register 2) to "1" in I^2C mode.

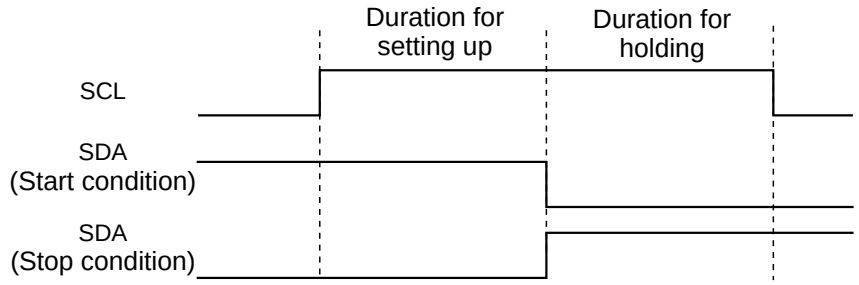
Table 1.16.11. Functions changed by I^2C mode select bit 2

	Function	IICM2 = 0	IICM2 = 1
1	Factor of interrupt number 15	No acknowledgment detection (NACK)	UART2 transmission (the rising edge of the final bit of the clock)
2	Factor of interrupt number 16	Acknowledgment detection (ACK)	UART2 reception (the falling edge of the final bit of the clock)
3	Timing for transferring data from the UART2 reception shift register to the reception buffer.	The rising edge of the final bit of the reception clock	The falling edge of the final bit of the reception clock
4	Timing for generating a UART2 reception/ACK interrupt request	The rising edge of the final bit of the reception clock	The falling edge of the final bit of the reception clock

Table 1.16.12. Timing characteristics of detecting the start condition and the stop condition (Note 1)

3 to 6 cycles < duration for setting-up (Note 2)
3 to 6 cycles < duration for holding (Note 2)

Note 1 : When the start/stop condition control bit SHTC is "1" .
Note 2 : "Cycles" is in terms of the input oscillation frequency f(X_{IN}) of the main clock.



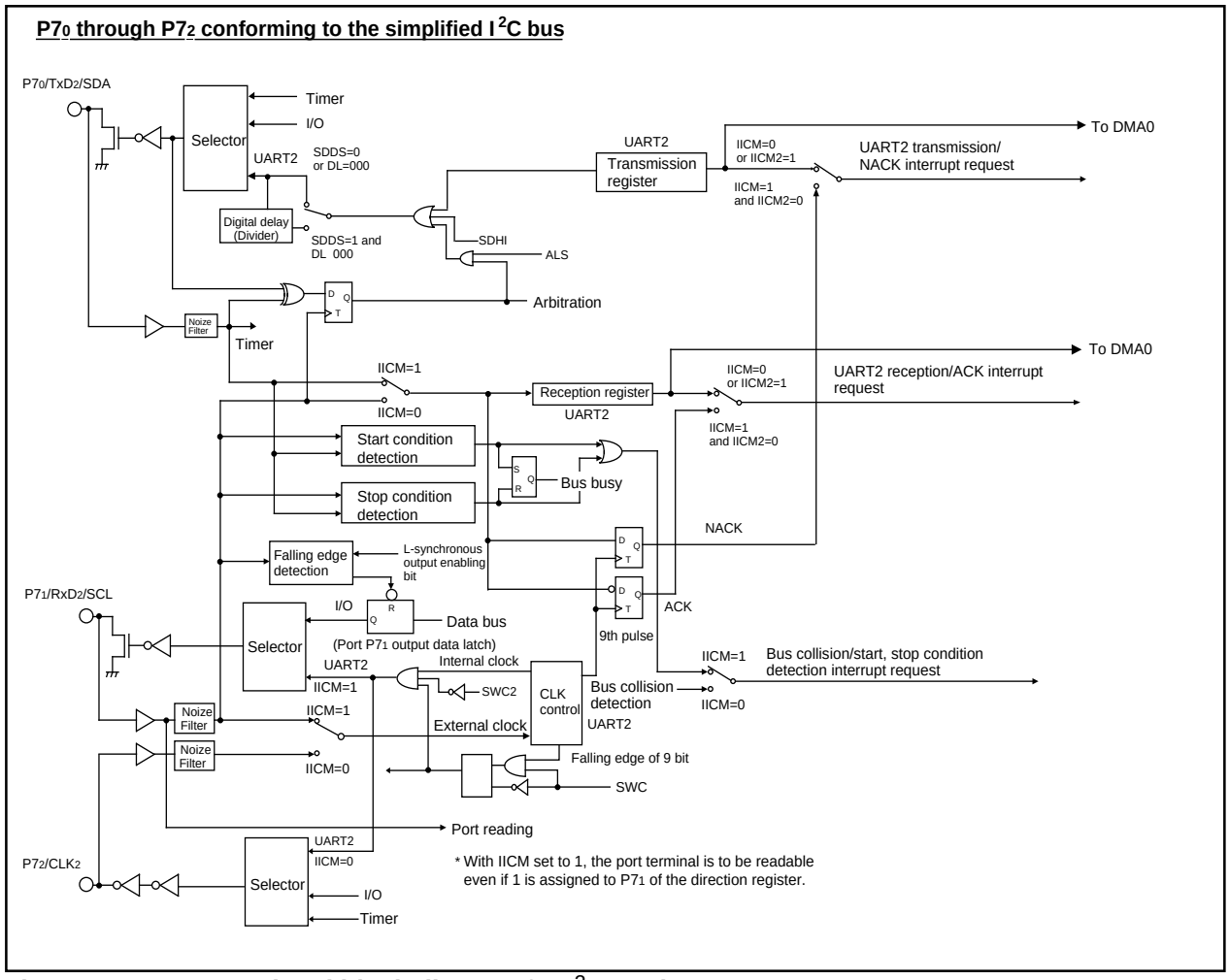


Figure 1.16.30. Functional block diagram for I²C mode

Functions available in I²C mode are shown in Figure 1.16.30 — a functional block diagram.

Bit 3 of the UART2 special mode register 2 (address 037616) is used as the SDA output stop bit. Setting this bit to “1” causes an arbitration loss to occur, and the SDA pin turns to high-impedance state at the instant when the arbitration lost detecting flag is set to “1”.

Bit 1 of the UART2 special mode register 2 (address 037616) is used as the clock synchronization bit. With this bit set to “1” at the time when the internal SCL is set to “H”, the internal SCL turns to “L” if the falling edge is found in the SCL pin; and the baud rate generator reloads the set value, and start counting within the “L” interval. When the internal SCL changes from “L” to “H” with the SCL pin set to “L”, stops counting the baud rate generator, and starts counting it again when the SCL pin turns to “H”. Due to this function, the UART2 transmission-reception clock becomes the logical product of the signal flowing through the internal SCL and that flowing through the SCL pin. This function operates over the period from the moment earlier by a half cycle than falling edge of the UART2 first clock to the rising edge of the ninth bit. To use this function, choose the internal clock for the transfer clock.

Bit 2 of the UART2 special mode register 2 (037616) is used as the SCL wait output bit. Setting this bit to “1” causes the SCL pin to be fixed to “L” at the falling edge of the ninth bit of the clock. Setting this bit to “0” frees the output fixed to “L”.

UART2 Special Mode Register 2

Bit 4 of the UART2 special mode register 2 (address 0376_{16}) is used as the UART2 initialization bit. Setting this bit to “1”, and when the start condition is detected, the microcomputer operates as follows.

(1) The transmission shift register is initialized, and the content of the transmission register is transferred to the transmission shift register. This starts transmission by dealing with the clock entered next as the first bit. The UART2 output value, however, doesn’t change until the first bit data is output after the entrance of the clock, and remains unchanged from the value at the moment when the microcomputer detected the start condition.

(2) The reception shift register is initialized, and the microcomputer starts reception by dealing with the clock entered next as the first bit.

(3) The SCL wait output bit turns to “1”. This turns the SCL pin to “L” at the falling edge of the ninth bit of the clock.

Starting to transmit/receive signals to/from UART2 using this function doesn’t change the value of the transmission buffer empty flag. To use this function, choose the external clock for the transfer clock.

Bit 5 of the UART2 special mode register 2 (0376_{16}) is used as the SCL wait output bit 2. Setting this bit to “1” with the serial I/O specified allows the user to forcibly output an “1” from the SCL pin even if UART2 is in operation. Setting this bit to “0” frees the “L” output from the SCL pin, and the UART2 clock is input/output.

Bit 6 of the UART2 special mode register 2 (0376_{16}) is used as the SDA output disable bit. Setting this bit to “1” forces the SDA pin to turn to the high-impedance state. Refrain from changing the value of this bit at the rising edge of the UART2 transfer clock. There can be instances in which arbitration lost detecting flag is turned on.

A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins P10$_0$ to P10$_7$, P9$_5$, and P9$_6$ also function as the analog signal input pins. The direction registers of these pins for A-D conversion must therefore be set to input. The Vref connect bit (bit 5 at address 03D7$_{16}$) can be used to isolate the resistance ladder of the A-D converter from the reference voltage input pin (V_{REF}) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from V_{REF}, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of 03D7$_{16}$ to connect V_{REF}. The result of A-D conversion is stored in the A-D registers of the selected pins. When set to 10-bit precision, the low 8 bits are stored in the even addresses and the high 2 bits in the odd addresses. When set to 8-bit precision, the low 8 bits are stored in the even addresses.

Table 1.17.1 shows the performance of the A-D converter. Figure 1.17.1 shows the block diagram of the A-D converter, and Figures 1.17.2 and 1.17.3 show the A-D converter-related registers.

Table 1.17.1. Performance of A-D converter

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AV$_{CC}$ (V_{CC})
Operating clock ϕ_{AD} (Note 2)	V_{CC} = 3.3V f_{AD}/divide-by-2 of f_{AD}/divide-by-4 of f_{AD}, f_{AD}=f(X_{IN})
Resolution	8-bit or 10-bit (selectable)
Absolute precision	V_{CC} = 3.3V • Without sample and hold function ±5LSB • With sample and hold function (8-bit resolution) ±2LSB • With sample and hold function (10-bit resolution) AN$_0$ to AN$_7$ input : ±5LSB ANEX0 and ANEX1 input (including mode in which external operation amp is connected) : ±7LSB
Operating modes	One-shot mode
Analog input pins	8pins (AN$_0$ to AN$_7$) + 2pins (ANEX0 and ANEX1)
A-D conversion start condition	• Software trigger A-D conversion starts when the A-D conversion start flag changes to “1” • External trigger (can be retriggered) A-D conversion starts when the A-D conversion start flag is “1” and the ADTRG/P9$_7$ input changes from “H” to “L”
Conversion speed per pin	• Without sample and hold function 8-bit resolution: 49 ϕ_{AD} cycles, 10-bit resolution: 59 ϕ_{AD} cycles • With sample and hold function 8-bit resolution: 28 ϕ_{AD} cycles, 10-bit resolution: 33 ϕ_{AD} cycles

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

Note 2: Divide the f_{AD} if f(X_{IN}) exceeds 10MHz, and make ϕ_{AD} frequency equal to or less than 10MHz. And divide the f_{AD} if V_{CC} is less than 3.0V, and make ϕ_{AD} frequency equal to or lower than f_{AD}/2.
Without sample and hold function, set the ϕ_{AD} frequency to 250kHz min.
With the sample and hold function, set the ϕ_{AD} frequency to 1MHz min.

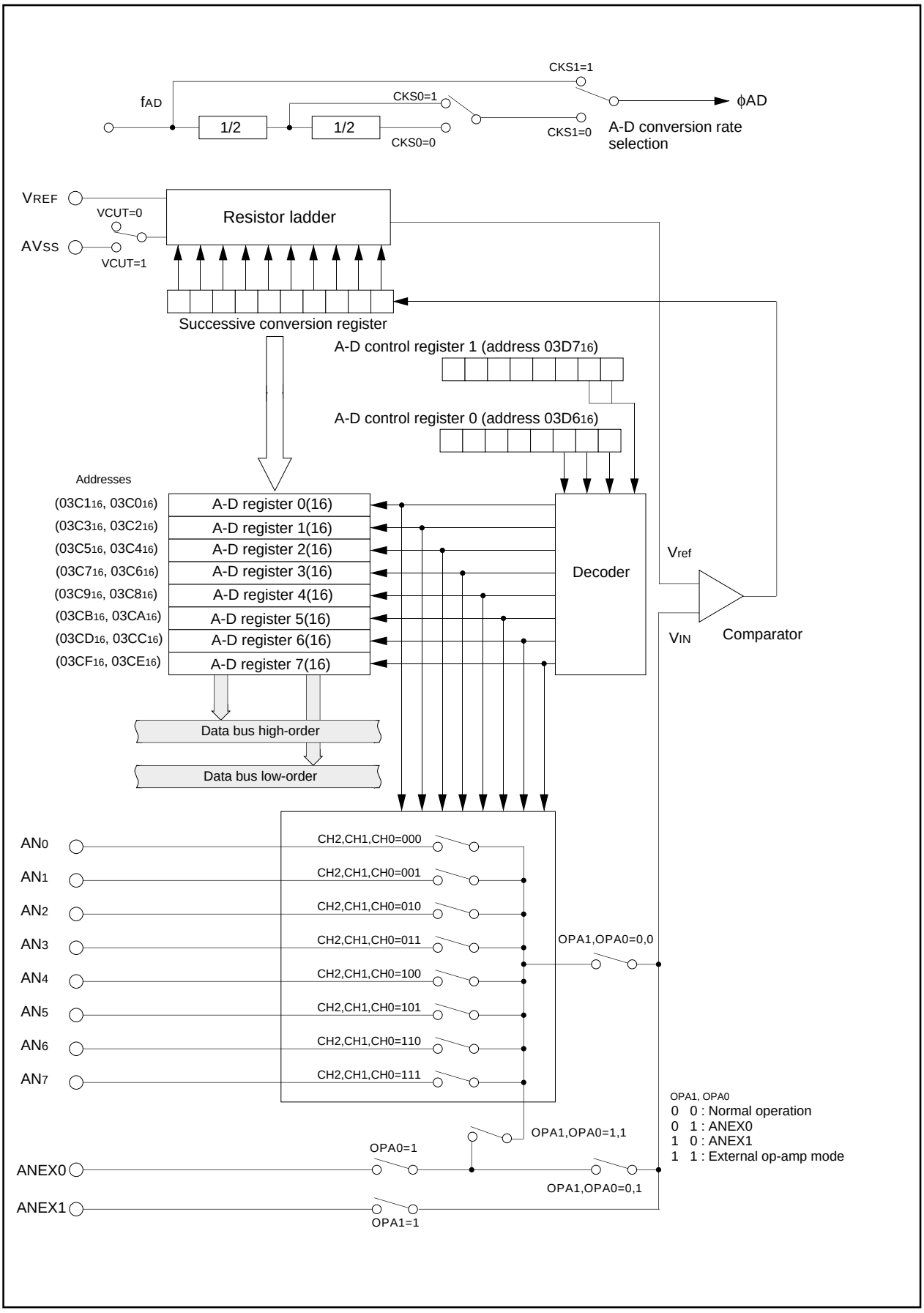


Figure 1.17.1. Block diagram of A-D converter

A-D control register 0 (Note)

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

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN$_0$ is selected 0 0 1 : AN$_1$ is selected 0 1 0 : AN$_2$ is selected 0 1 1 : AN$_3$ is selected 1 0 0 : AN$_4$ is selected 1 0 1 : AN$_5$ is selected 1 1 0 : AN$_6$ is selected 1 1 1 : AN$_7$ is selected	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode Must not be set except “00”	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	○	○

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

A-D control register 1 (Note)

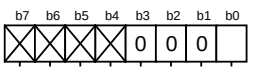
Symbol	Address	When reset
ADCON1	03D7$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : f$_{AD}$/2 or f$_{AD}$/4 is selected 1 : f$_{AD}$ is selected	○	○
VCUT	Vref connect bit	0 : Vref not connected 1 : Vref connected	○	○
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: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Figure 1.17.2. A-D converter-related registers (1)

A-D control register 2 (Note)



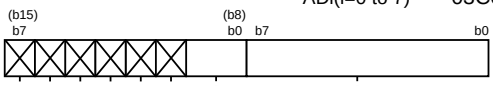
Symbol: ADCON2 | Address: $03D4_{16}$ | When reset: $0000XXX0_2$

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		Must always be set to “0”	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–

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

A-D register i

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



Function	R	W
Eight low-order bits of A-D conversion result	○	×
• During 10-bit mode Two high-order bits of A-D conversion result	○	×
• During 8-bit mode When read, the content is indeterminate	×	×
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.	–	–

Figure 1.17.3. A-D converter-related registers (2)

(1) One-shot mode

In one-shot mode, the pin selected using the analog input pin select bit is used for one-shot A-D conversion. Table 1.17.2 shows the specifications of one-shot mode. Figure 1.17.4 shows the A-D control register in one-shot mode.

Table 1.17.2. 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”, except when external trigger is selected) • Writing “0” to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN_0 to AN_7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
			0	0			

Symbol ADCON0 — Address $03D6_{16}$ — When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : AD_{TRG} trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0: $f_{AD}/4$ is selected 1: $f_{AD}/2$ is selected	○	○

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

A-D control register 1 (Note)

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

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

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit1	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: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Figure 1.17.4. A-D conversion register in one-shot mode

(a) Sample and hold

Sample and hold is selected by setting bit 0 of the A-D control register 2 (address 03D4$_{16}$) to “1”. When sample and hold is selected, the rate of conversion of each pin increases. As a result, a 28 ϕ_{AD} cycle is achieved with 8-bit resolution and 33 ϕ_{AD} with 10-bit resolution. Sample and hold can be selected in all modes. However, in all modes, be sure to specify before starting A-D conversion whether sample and hold is to be used.

(b) Extended analog input pins

In one-shot mode and repeat mode, the input via the extended analog input pins ANEX0 and ANEX1 can also be converted from analog to digital.

When bit 6 of the A-D control register 1 (address 03D7$_{16}$) is “1” and bit 7 is “0”, input via ANEX0 is converted from analog to digital. The result of conversion is stored in A-D register 0.

When bit 6 of the A-D control register 1 (address 03D7$_{16}$) is “0” and bit 7 is “1”, input via ANEX1 is converted from analog to digital. The result of conversion is stored in A-D register 1.

(c) External operation amp connection mode

In this mode, multiple external analog inputs via the extended analog input pins, ANEX0 and ANEX1, 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 03D7$_{16}$) is “1” and bit 7 is “1”, input via AN$_0$ to AN$_7$ is output from ANEX0. The input from ANEX1 is converted from analog to digital and the result stored in the corresponding A-D register. The speed of A-D conversion depends on the response of the external operation amp. Do not connect the ANEX0 and ANEX1 pins directly. Figure 1.17.5 is an example of how to connect the pins in external operation amp mode.

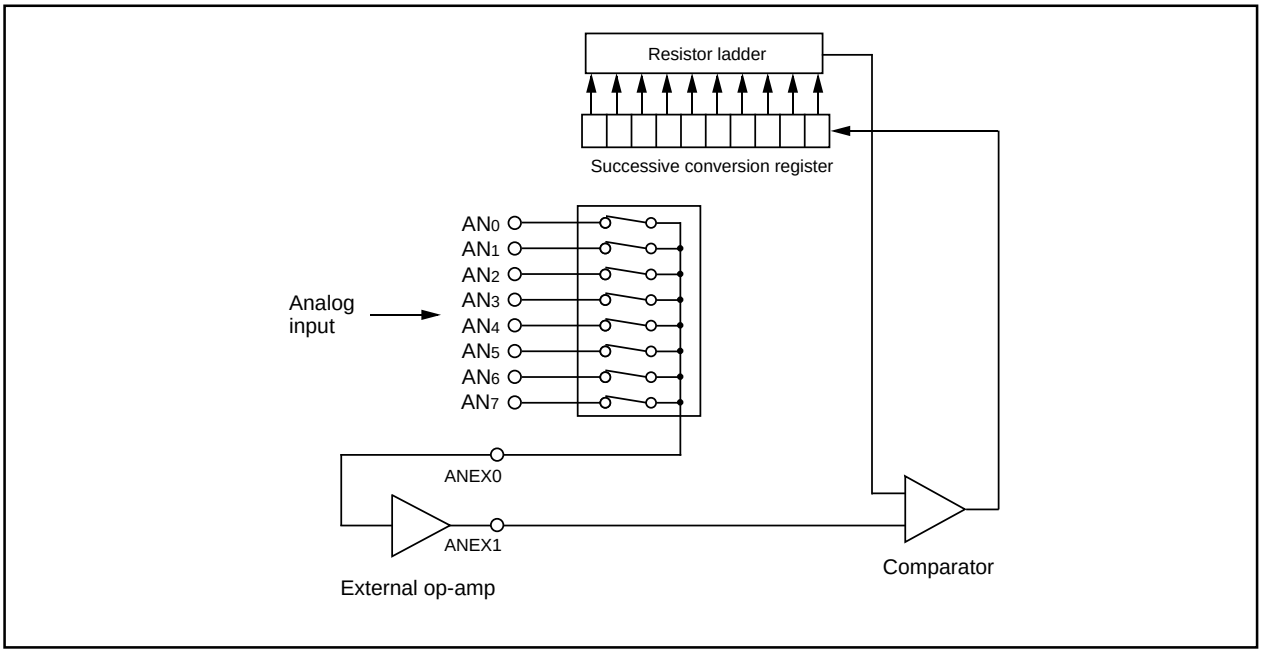


Figure 1.17.5. Example of external op-amp connection mode

Programmable I/O Ports

There are 87 programmable I/O ports: P0 to P10 (excluding $P8_5$). 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. $P8_5$ is an input-only port and has no built-in pull-up resistance.

Figures 1.20.1 to 1.20.4 show the programmable I/O ports. Figure 1.20.5 shows the I/O pins.

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 1.20.6 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.

In memory expansion and microprocessor mode, the contents of corresponding direction register of pins A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}/\overline{WR}$, $\overline{WRH}/\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$ and BCLK cannot be modified.

Note: There is no direction register bit for $P8_5$.

(2) Port registers

Figure 1.20.7 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.

In memory expansion and microprocessor mode, the contents of corresponding port register of pins A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}/\overline{WR}$, $\overline{WRH}/\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$ and BCLK cannot be modified.

(3) Pull-up control registers

Figure 1.20.8 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.

However, in memory expansion mode and microprocessor mode, the pull-up control register of P0 to P3, $P4_0$ to $P4_3$, and P5 is invalid. The contents of register can be changed, but the pull-up resistance is not connected.

(4) Port control register

Figure 1.20.9 shows the port control register.

The bit 0 of port control register is used to read port P1 as follows:

0 : When port P1 is input port, port input level is read.
When port P1 is output port , the contents of port P1 register is read.

1 : The contents of port P1 register is read always.

This register is valid in the following:

- External bus width is 8 bits in microprocessor mode or memory expansion mode.
- Port P1 can be used as a port in multiplexed bus for the entire space.

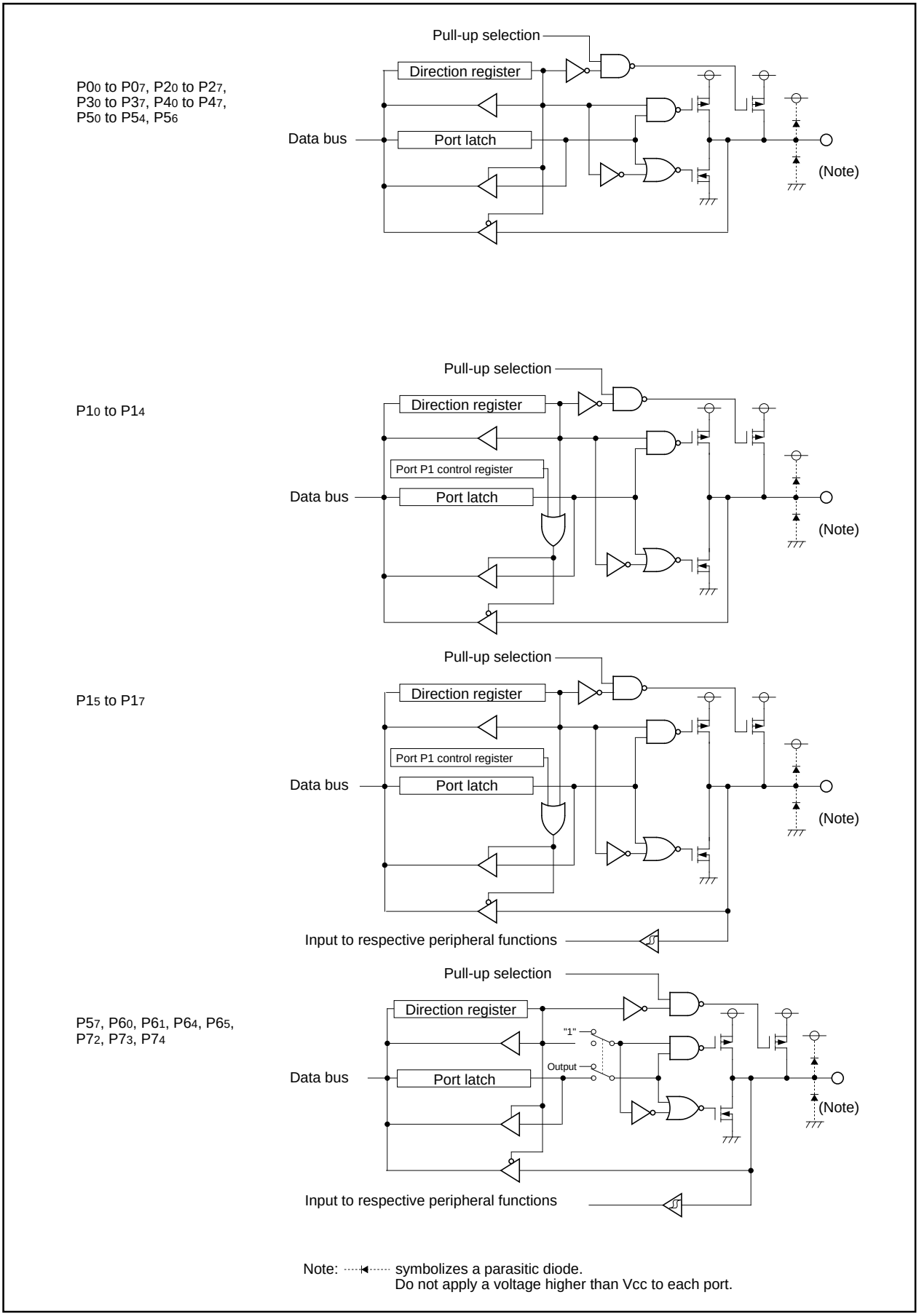


Figure 1.20.1. Programmable I/O ports (1)

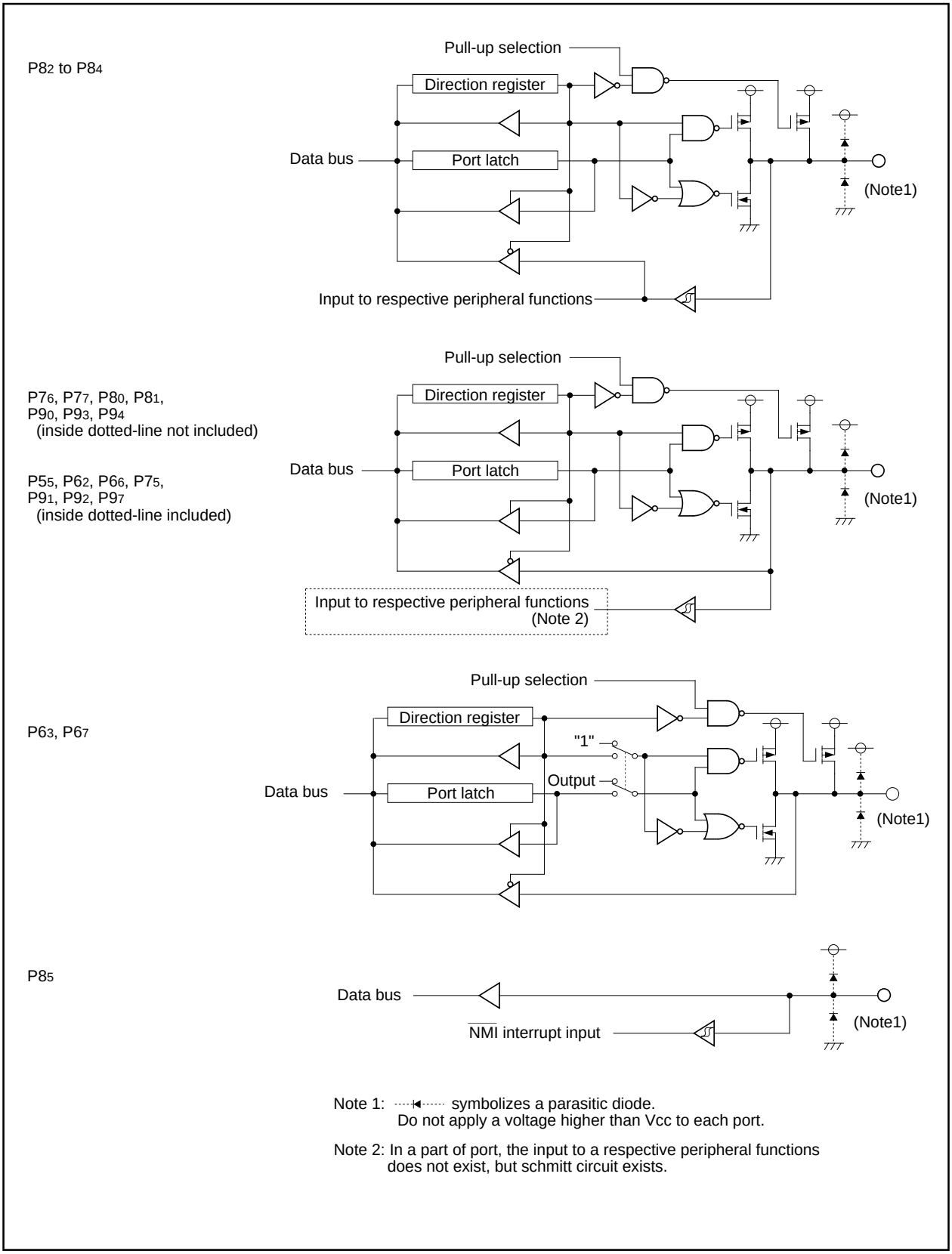


Figure 1.20.2. Programmable I/O ports (2)

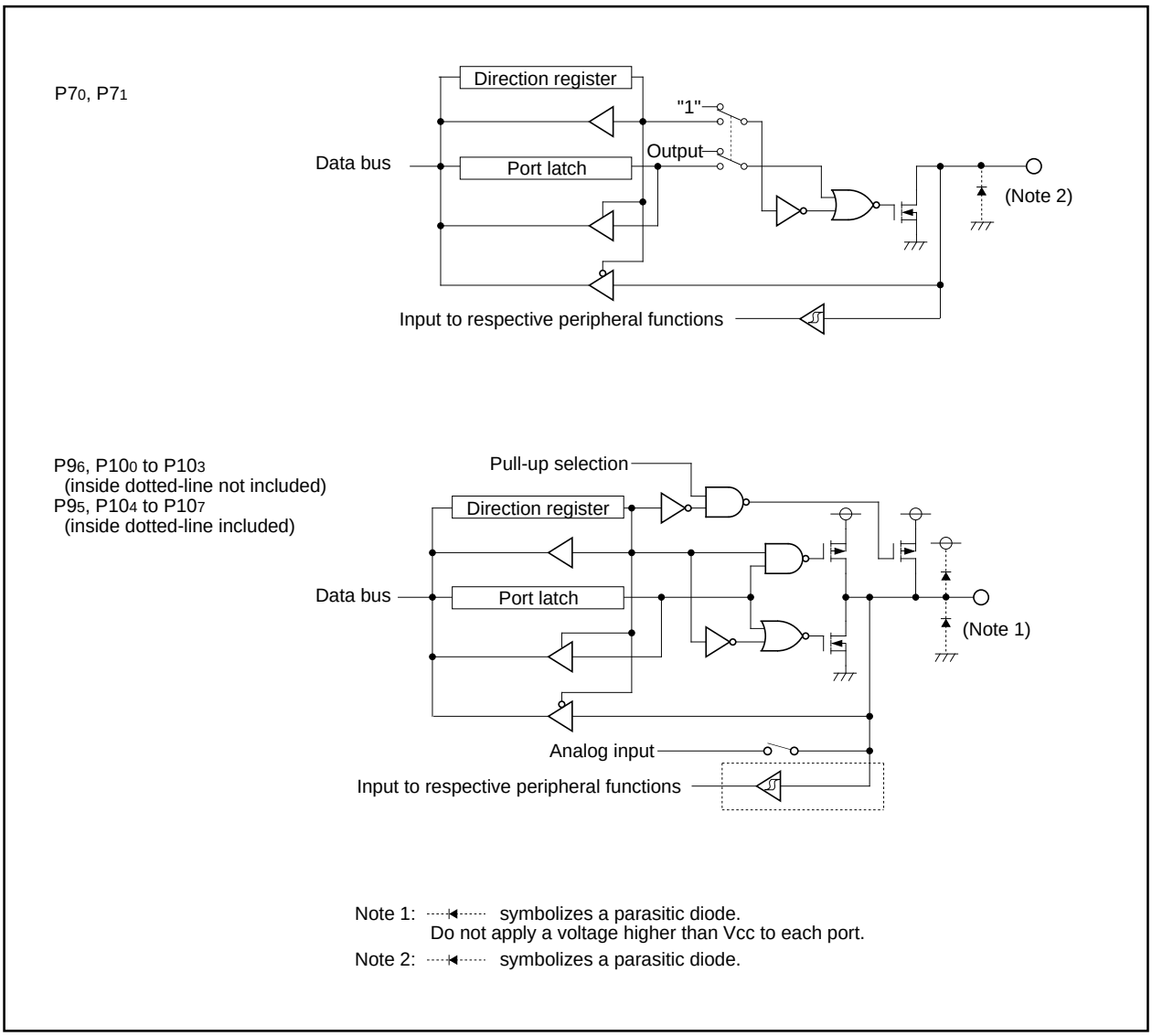


Figure 1.20.3. Programmable I/O ports (3)

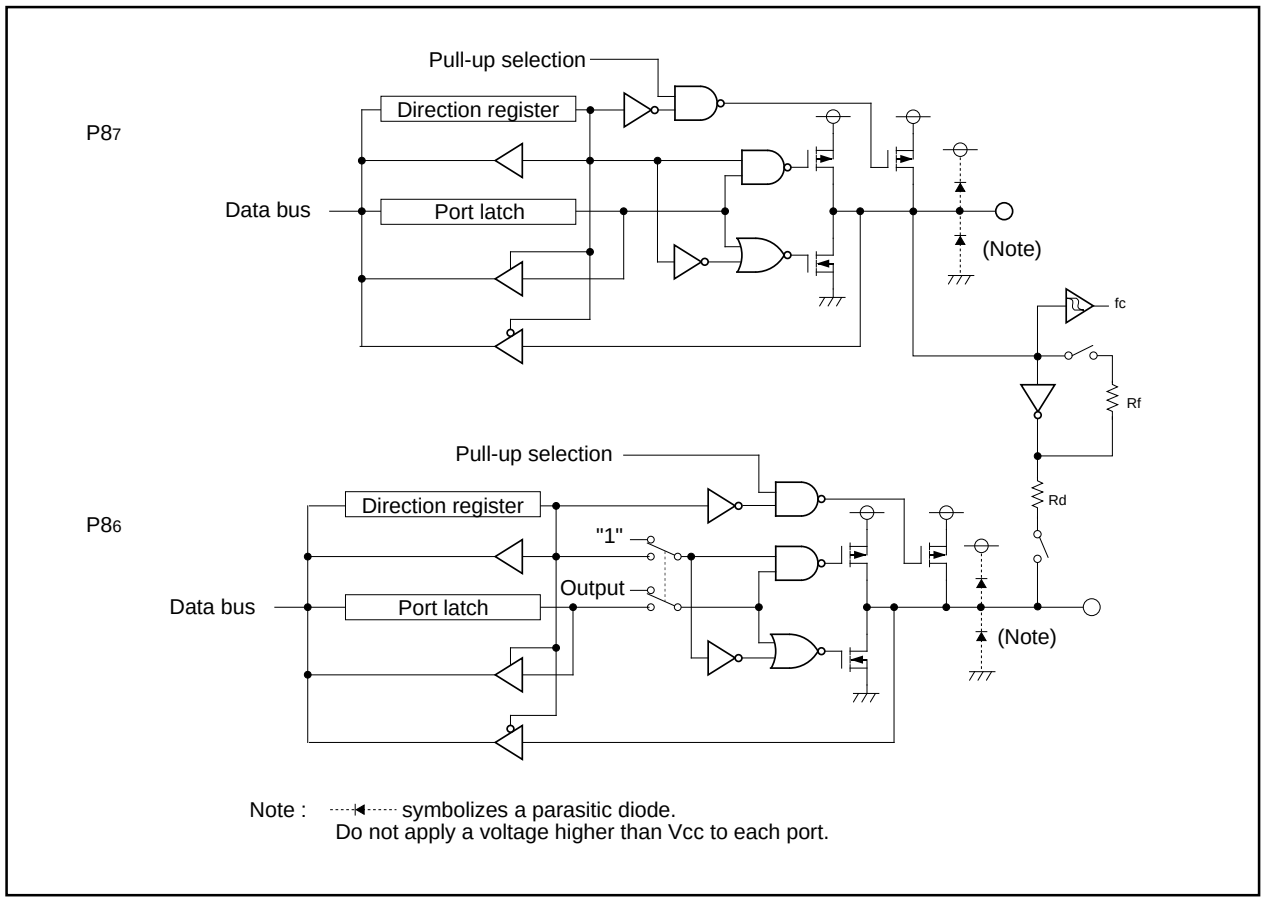


Figure 1.20.4. Programmable I/O ports (4)

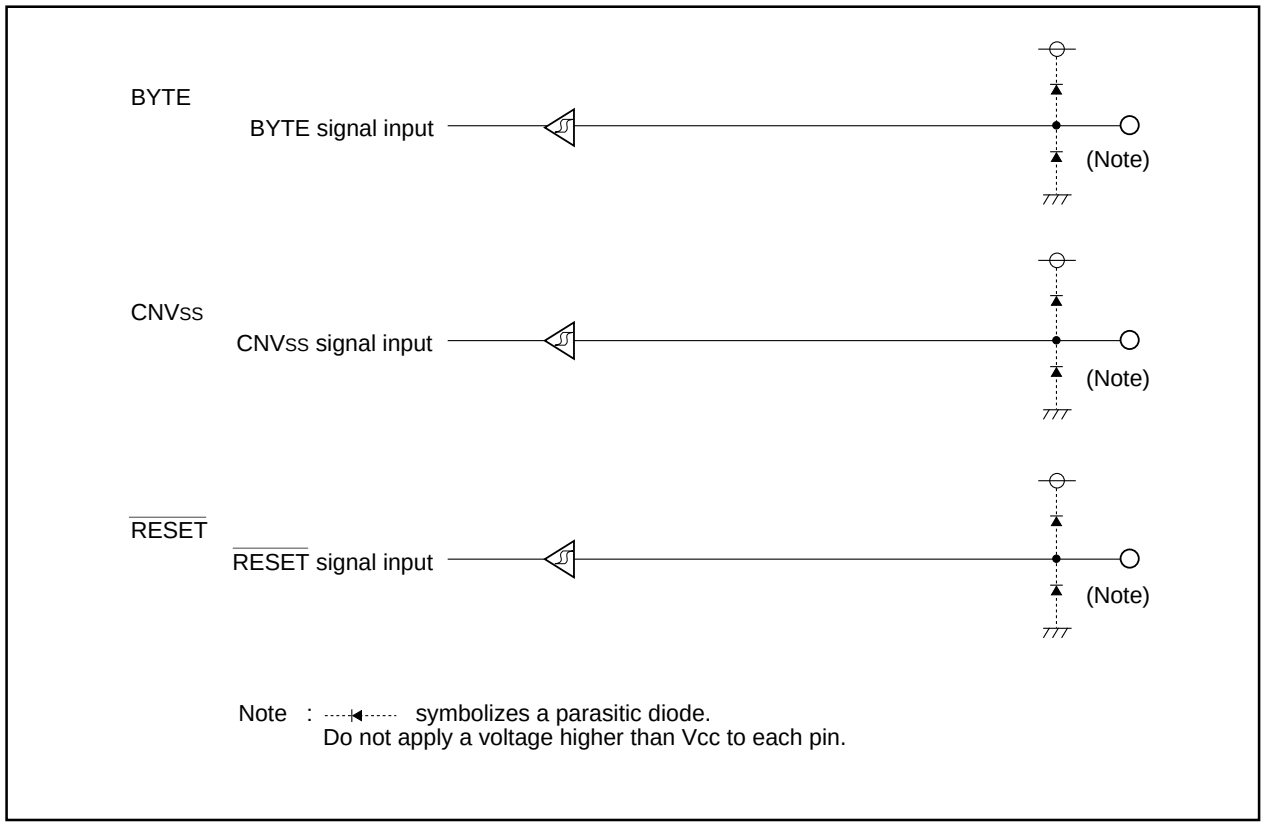
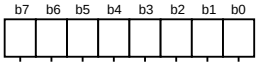


Figure 1.20.5. I/O pins

Port Pi direction register (Note 1, 2)

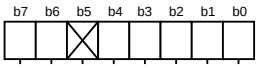


Symbol	Address	When reset
PDi (i = 0 to 10, except 8)	03E2$_{16}$, 03E3$_{16}$, 03E6$_{16}$, 03E7$_{16}$, 03EA$_{16}$ 03EB$_{16}$, 03EE$_{16}$, 03EF$_{16}$, 03F3$_{16}$, 03F6$_{16}$	00$_{16}$ 00$_{16}$

Bit symbol	Bit name	Function	R	W
PDi_0	Port Pi$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 0 to 10 except 8)	○	○
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 P9 direction register.
Note 2: In memory expansion and microprocessor mode, the contents of corresponding port Pi direction register of pins $\overline{A_0}$ to $\overline{A_{19}}$, $\overline{D_0}$ to $\overline{D_{15}}$, $\overline{CS_0}$ to $\overline{CS_3}$, $\overline{RD}$, $\overline{WRL}$/$\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, ALE, RDY, HOLD, HLDA and BCLK cannot be modified.

Port P8 direction register

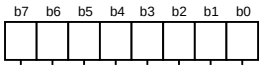


Symbol	Address	When reset
PD8	03F2$_{16}$	00X00000$_2$

Bit symbol	Bit name	Function	R	W
PD8_0	Port P8$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PD8_1	Port P8$_1$ direction register		○	○
PD8_2	Port P8$_2$ direction register		○	○
PD8_3	Port P8$_3$ direction register		○	○
PD8_4	Port P8$_4$ direction register		○	○
	Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.		—	—
PD8_6	Port P8$_6$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PD8_7	Port P8$_7$ direction register		○	○

Figure 1.20.6. Direction register

Port Pi register (Note 2)

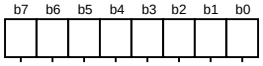


Symbol	Address	When reset
Pi (i = 0 to 10, except 8)	03E0$_{16}$, 03E1$_{16}$, 03E4$_{16}$, 03E5$_{16}$, 03E8$_{16}$ 03E9$_{16}$, 03EC$_{16}$, 03ED$_{16}$, 03F1$_{16}$, 03F4$_{16}$	Indeterminate Indeterminate

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 (Note 1) (i = 0 to 10 except 8)	○	○
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 1: Since P7$_0$ and P7$_1$ are N-channel open drain ports, the data is high-impedance.
Note 2: In memory expansion and microprocessor mode, the contents of corresponding port Pi register of pins A$_0$ to A$_{19}$, D$_0$ to D$_{15}$, $\overline{CS_0}$ to $\overline{CS_3}$, $\overline{RD}$, $\overline{WRL}$/$\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$ and BCLK cannot be modified.

Port P8 register

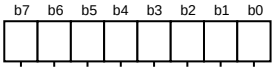


Symbol	Address	When reset
P8	03F0$_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
P8_0	Port P8$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit (except for P8$_5$) 0 : “L” level data 1 : “H” level data	○	○
P8_1	Port P8$_1$ register		○	○
P8_2	Port P8$_2$ register		○	○
P8_3	Port P8$_3$ register		○	○
P8_4	Port P8$_4$ register		○	○
P8_5	Port P8$_5$ register		○	×
P8_6	Port P8$_6$ register		○	○
P8_7	Port P8$_7$ register		○	○

Figure 1.20.7. Port register

Pull-up control register 0 (Note)

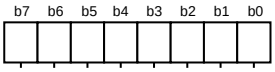


Symbol: PUR0 | Address: 03FC16 | When reset: 0016

Bit symbol	Bit name	Function	R	W
PU00	P00 to P03 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU01	P04 to P07 pull-up		○	○
PU02	P10 to P13 pull-up		○	○
PU03	P14 to P17 pull-up		○	○
PU04	P20 to P23 pull-up		○	○
PU05	P24 to P27 pull-up		○	○
PU06	P30 to P33 pull-up		○	○
PU07	P34 to P37 pull-up		○	○

Note : In memory expansion and microprocessor mode, the content of this register can be changed, but the pull-up resistance is not connected.

Pull-up control register 1

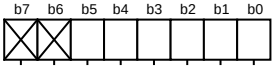


Symbol: PUR1 | Address: 03FD16 | When reset: 0016 (Note 2)

Bit symbol	Bit name	Function	R	W
PU10	P40 to P43 pull-up (Note 3)	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU11	P44 to P47 pull-up		○	○
PU12	P50 to P53 pull-up (Note 3)		○	○
PU13	P54 to P57 pull-up (Note 3)		○	○
PU14	P60 to P63 pull-up		○	○
PU15	P64 to P67 pull-up		○	○
PU16	P72 to P73 pull-up (Note 1)		○	○
PU17	P74 to P77 pull-up		○	○

Note 1: Since P70 and P71 are N-channel open drain ports, pull-up is not available for them.
Note 2: When the VCC level is being impressed to the CNVSS terminal, this register becomes to 0216 when reset (PU11 becomes to “1”).
Note 3: In memory expansion and microprocessor mode, the content of these bits can be changed, but the pull-up resistance is not connected.

Pull-up control register 2

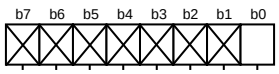


Symbol: PUR2 | Address: 03FE16 | When reset: 0016

Bit symbol	Bit name	Function	R	W
PU20	P80 to P83 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU21	P84 to P87 pull-up (Except P85)		○	○
PU22	P90 to P93 pull-up		○	○
PU23	P94 to P97 pull-up		○	○
PU24	P100 to P103 pull-up		○	○
PU25	P104 to P107 pull-up		○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			—	—
			—	—

Figure 1.20.8. Pull-up control register

Port control register



Symbpl: PCR
Address: 03FF16
When reset: 0016

Bit symbol	Bit name	Function	R	W
PCR0	Port P1 control register	0 : When input port, read port input level. When output port, read the contents of port P1 register. 1 : Read the contents of port P1 register though input/output port.	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			—	—

Figure 1.20.9. Port control register

Table 1.20.1. Example connection of unused pins in single-chip mode

Pin name	Connection
Ports P0 to P10 (excluding P8$_5$)	After setting for input mode, connect every pin to V_{SS} via a resistor (pull-down); or after setting for output mode, leave these pins open.
X_{OUT} (Note)	Open
$\overline{NMI}$	Connect via resistor to V_{CC} (pull-up)
AV_{CC}	Connect to V_{CC}
AV_{SS}, V_{REF}, BYTE	Connect to V_{SS}

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

Table 1.20.2. Example connection of unused pins in memory expansion mode and microprocessor mode

Pin name	Connection
Ports P6 to P10 (excluding P8$_5$)	After setting for input mode, connect every pin to V_{SS} via a resistor (pull-down); or after setting for output mode, leave these pins open.
P4$_5$ / $\overline{CS1}$ to P4$_7$ / $\overline{CS3}$	Set ports to input mode, set output enable bits of $\overline{CS1}$ through $\overline{CS3}$ to 0, and connect to Vcc via resistors (pull-up).
$\overline{BHE}$, ALE, $\overline{HLDA}$, X_{OUT} (Note 1), BCLK (Note 2)	Open
$\overline{HOLD}$, $\overline{RDY}$, $\overline{NMI}$	Connect via resistor to V_{CC} (pull-up)
AV_{CC}	Connect to V_{CC}
AV_{SS}, V_{REF}	Connect to V_{SS}

Note 1: With external clock input to X_{IN} pin.
Note 2: When the BCLK output disable bit (bit 7 at address 0004$_{16}$) is set to “1”, connect to V_{CC} via a resistor (pull-up).

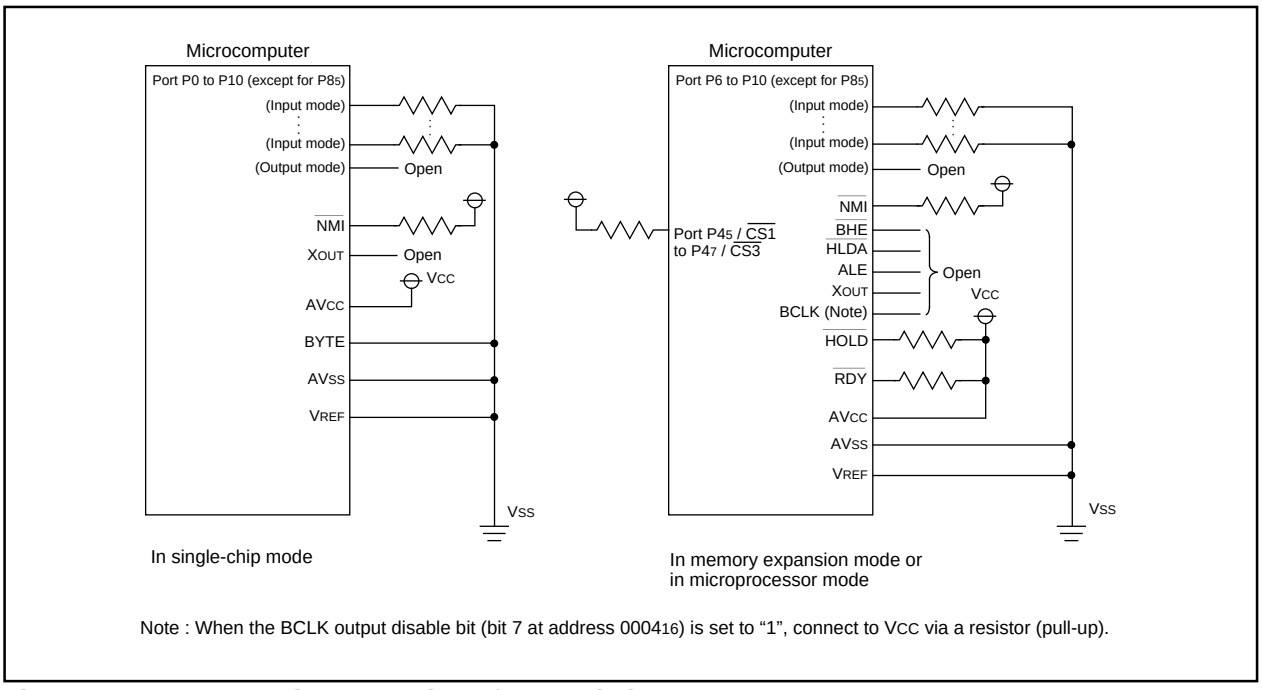


Figure 1.20.10. Example connection of unused pins

Electrical characteristics

Table 1.26.1. Absolute maximum ratings

Symbol	Parameter		Condition	Rated value	Unit
V_{CC}	Supply voltage		V_{CC}=AV$_{CC}$	- 0.3 to 4.2	V
AVcc	Analog supply voltage		V_{CC}=AV$_{CC}$	- 0.3 to 4.2	V
V_I	Input voltage	$\overline{RESET}$, CNV$_{SS}$, BYTE, P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$,P4$_0$ to P4$_7$, P5$_0$ to P5$_7$, P6$_0$ to P6$_7$, P7$_2$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_7$, P10$_0$ to P10$_7$, V_{REF}, X_{IN}		- 0.3 to Vcc + 0.3	V
		P7$_0$, P7$_1$		- 0.3 to 4.2	V
V_O	Output voltage	P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P4$_0$ to P4$_7$, P5$_0$ to P5$_7$, P6$_0$ to P6$_7$, P7$_2$ to P7$_7$, P8$_0$ to P8$_4$, P8$_6$, P8$_7$, P9$_0$ to P9$_7$, P10$_0$ to P10$_7$, X_{OUT}		- 0.3 to Vcc + 0.3	V
		P7$_0$, P7$_1$		- 0.3 to 4.2	V
P_d	Power dissipation		Topr=25 °C	300	mW
T_{opr}	Operating ambient temperature			- 20 to 85 / -40 to 85 (Note)	°C
T_{stg}	Storage temperature			- 65 to 150	°C

Note: Specify a product of -40°C to 85°C to use it.

Table 1.26.2. Recommended operating conditions (referenced to V_{CC} = 2.2V to 3.6V at Topr = – 20°C to 85°C / – 40°C to 85°C(Note 3) unless otherwise specified)

Symbol	Parameter			Standard			Unit
				Min.	Typ.	Max.	
Vcc	Supply voltage			2.2	3.3	3.6	V
AVcc	Analog supply voltage				Vcc		V
Vss	Supply voltage				0		V
AVss	Analog supply voltage				0		V
V_{IH}	HIGH input voltage	$P3_1$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE		0.8Vcc		Vcc	V
		$P7_0$, $P7_1$		0.8Vcc		4.2	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (during single-chip mode)		0.8Vcc		Vcc	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (data input function during memory expansion and microprocessor modes)		0.5Vcc		Vcc	V
V_{IL}	LOW input voltage	$P3_1$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE		0		0.2Vcc	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (during single-chip mode)		0		0.2Vcc	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (data input function during memory expansion and microprocessor modes)		0		0.16Vcc	V
$I_{OH (peak)}$	HIGH peak output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$				- 10.0	mA
$I_{OH (avg)}$	HIGH average output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$				- 5.0	mA
$I_{OL (peak)}$	LOW peak output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$				10.0	mA
$I_{OL (avg)}$	LOW average output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$				5.0	mA
f (X_{IN})	Main clock input oscillation frequency (Note 4)	No wait	Vcc=3.0V to 3.6V	0		16	MHz
			Vcc=2.4V to 3.0V	0		15 X Vcc - 29	MHz
			Vcc=2.2V to 2.4V	0		17.5 X Vcc - 35	MHz
		With wait	Vcc=3.0V to 3.6V	0		16	MHz
			Vcc=2.4V to 3.0V	0		11.25 X Vcc - 17.75	MHz
			Vcc=2.2V to 2.4V	0		11.25 X Vcc - 17.75	MHz
f (X_{CIN})	Subclock oscillation frequency				32.768	50	kHz

Note 1: The mean output current is the mean value within 100ms.

Note 2: The total I_{OL} (peak) for ports P0, P1, P2, $P8_6$, $P8_7$, P9, and P10 must be 80mA max. The total I_{OH} (peak) for ports P0, P1, P2, $P8_6$, $P8_7$, P9, and P10 must be 80mA max. The total I_{OL} (peak) for ports P3, P4, P5, P6, P7, and $P8_0$ to $P8_4$ must be 80mA max. The total I_{OH} (peak) for ports P3, P4, P5, P6, $P7_2$ to $P7_7$, and $P8_0$ to $P8_4$ must be 80mA max.

Note 3: Specify a product of -40°C to 85°C to use it.

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

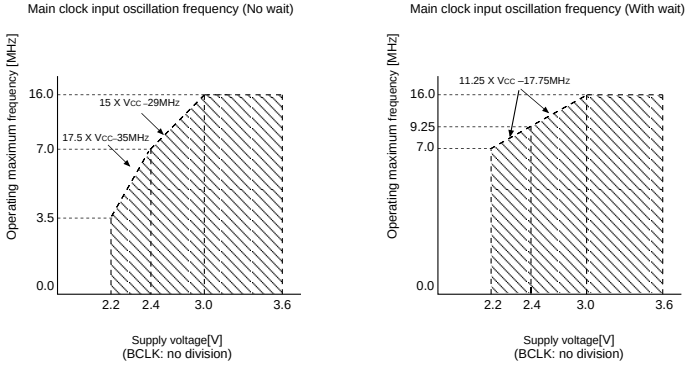


Table 1.26.3. Electrical characteristics (referenced to V_{CC} = 3.0V to 3.6V, V_{SS} = 0V at Topr = – 20°C to 85°C / – 40°C to 85°C (Note 1), f(X_{IN}) = 16MHz unless otherwise specified)

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 $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$		I_{OH} = -1mA V_{CC} = 3.3V	2.8			V
V_{OH}	HIGH output voltage X_{OUT}	HIGHPOWER	I_{OH} = -0.1mA, V_{CC} = 3.3V	2.8			V
		LOWPOWER	I_{OH} = -50µA, V_{CC} = 3.3V	2.8			
	HIGH output voltage X_{COUT}	HIGHPOWER	With no load applied, V_{CC} = 3.3V		2.8		V
		LOWPOWER	With no load applied, V_{CC} = 3.3V		1.6		
V_{OL}	LOW output voltage $P0_0$ to $P0_7$,$P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$		I_{OL} = 1mA V_{CC} = 3.3V			0.5	V
V_{OL}	LOW output voltage X_{OUT}	HIGHPOWER	I_{OL} = 0.1mA, V_{CC} = 3.3V			0.5	V
		LOWPOWER	I_{OL} = 50µA, V_{CC} = 3.3V			0.5	
	LOW output voltage X_{COUT}	HIGHPOWER	With no load applied, V_{CC} = 3.3V		0		V
		LOWPOWER	With no load applied, V_{CC} = 3.3V		0		
V_{T+}-V_{T-}	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, TA0$_{IN}$ to TA2$_{IN}$, TB1$_{IN}$, TB2$_{IN}$, $\overline{INT0}$ to $\overline{INT2}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS0}$ to $\overline{CTS2}$, SCL, SDA CLK0 to CLK2,TA2$_{OUT}$, $\overline{KI0}$ to $\overline{KI3}$, RxD0 to RxD2		V_{CC} = 3.3V	0.2		0.8	V
V_{T+}-V_{T-}	Hysteresis $\overline{RESET}$		V_{CC} = 3.3V	0.2		1.8	V
I_{IH}	HIGH input current $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$,$P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNVss, BYTE		V_I = 3V V_{CC} = 3.3V			4.0	µA
I_{IL}	LOW input current $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNVss, BYTE		V_I = 0V V_{CC} = 3.3V			-4.0	µA
R_{PULLUP}	Pull-up resistance $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$ to $P9_7$,$P10_0$ to $P10_7$		V_I = 0V V_{CC} = 3.3V	20.0	100.0	500.0	kΩ
R_{fXIN}	Feedback resistance X_{IN}				3.0		MΩ
R_{fXCIN}	Feedback resistance X_{CIN}				10.0		MΩ
V_{RAM}	RAM retention voltage		When clock is stopped	2.0			V
Icc	Power supply current	In single-chip mode, the output pins are open and other pins are V_{SS}	f(X_{IN}) = 16MHz, Square wave, no division		12.5	25.0	mA
			f(X_{CIN}) = 32kHz, V_{CC} = 3.3V Square wave		40.0		µA
			f(X_{CIN}) = 32kHz, V_{CC} = 3.3V When a WAITinstruction is executed. Oscillation capacity High (Note 2)		5.8		µA
			f(X_{CIN}) = 32kHz, V_{CC} = 3.3V When a WAIT instruction is executed. Oscillation capacity Low (Note 2)		2.7		µA
			Topr = 25°C, V_{CC} = 3.3V when clock is stopped		0.1	2.0	µA
			Topr = 85°C, V_{CC} = 3.3V when clock is stopped		0.4	100	

Note 1: Specify a product of -40°C to 85°C to use it.
Note 2: With one timer operated using f_{C32}.

Table 1.26.4. A-D conversion characteristics (referenced to V_{CC} = AV_{CC} = V_{REF} = 2.4V to 3.6V, V_{SS} = AV_{SS} = 0V, at Topr = – 20°C to 85°C / – 40°C to 85°C (Note 4), f(X_{IN}) =16MHz unless otherwise specified)

Symbol	Paramete		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} = 3.3V			±2	±5	LSB
		Sample & hold function available(10bit)	V_{REF}=V_{CC} = 3.3V	AN_0 to AN_7 input		±2	±5	LSB
				ANEX0, ANEX1 input			±7	LSB
		Sample & hold function available(8bit)	V_{REF} = V_{CC} = 3.3V				±2	LSB
R_{LADDER}	Ladder resistance		V_{REF} = V_{CC}		10		40	kΩ
t_{CONV}	Conversion time(10bit)				3.3			µs
t_{CONV}	Conversion time(8bit)				2.8			µs
t_{SAMP}	Sampling time				0.3			µs
V_{REF}	Reference voltage				2.4		V_{CC}	V
V_{IA}	Analog input voltage				0		V_{REF}	V

Note 1: Do f(X_{IN}) in range of main clock input oscillation frequency prescribed with recommended operating conditions of table 1.26.2. Divide the f_{AD} if f(X_{IN}) exceeds 10MHz, and make AD operation clock frequency (ØAD) equal to or lower than 10MHz. And divide the f_{AD} if V_{CC} is less than 3.0V, and make AD operation clock frequency (ØAD) equal to or lower than f_{AD}/2.

Note 2: A case without sample & hold function turn AD operation clock frequency (ØAD) into 250 kHz or more in addition to a limit of Note 1.
A case with sample & hold function turn AD operation clock frequency (ØAD) into 1MHz or more in addition to a limit of Note 1.

Note 3: Connect AV_{CC} pin to V_{CC} pin and apply the same electric potential.

Note 4: Specify a product of -40°C to 85°C to use it.

Timing requirements

(referenced to V_{CC} = 3.3V, V_{SS} = 0V, at Topr = – 20°C to 85°C / – 40°C to 85°C (*) unless otherwise specified)

* : Specify a product of -40°C to 85°C to use it.

Table 1.26.8. External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	62.5		ns
$t_{w(H)}$	External clock input HIGH pulse width	25		ns
$t_{w(L)}$	External clock input LOW pulse width	25		ns
t_r	External clock rise time		15	ns
t_f	External clock fall time		15	ns

Table 1.26.9. Memory expansion and microprocessor modes

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{ac1(RD-DB)}$	Data input access time (no wait)		(Note)	ns
$t_{ac2(RD-DB)}$	Data input access time (with wait)		(Note)	ns
$t_{ac3(RD-DB)}$	Data input access time (when accessing multiplex bus area)		(Note)	ns
$t_{su(DB-RD)}$	Data input setup time	50		ns
$t_{su(RDY-BCLK)}$	$\overline{RDY}$ input setup time	50		ns
$t_{su(HOLD-BCLK)}$	$\overline{HOLD}$ input setup time	100		ns
$t_{h(RD-DB)}$	Data input hold time	0		ns
$t_{h(BCLK-RDY)}$	$\overline{RDY}$ input hold time	0		ns
$t_{h(BCLK-HOLD)}$	$\overline{HOLD}$ input hold time	0		ns
$t_{d(BCLK-HLDA)}$	$\overline{HLDA}$ output delay time		40	ns

Note: Calculated according to the BCLK frequency as follows:

$$t_{ac1(RD-DB)} = \frac{10^9}{f(BCLK) \times 2} - 90 \quad [ns]$$

$$t_{ac2(RD-DB)} = \frac{3 \times 10^9}{f(BCLK) \times 2} - 90 \quad [ns]$$

$$t_{ac3(RD-DB)} = \frac{3 \times 10^9}{f(BCLK) \times 2} - 90 \quad [ns]$$

Timing requirements

(referenced to V_{CC} = 3.3V, V_{SS} = 0V, at Topr = – 20°C to 85°C / – 40°C to 85°C (*) unless otherwise specified)

* : Specify a product of –40°C to 85°C to use it.

Table 1.26.10. Timer A input (counter input in event counter mode)

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

Table 1.26.11. Timer A input (gating input in timer mode)

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

Table 1.26.12. Timer A input (external trigger input in one-shot timer mode)

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

Table 1.26.13. Timer A input (external trigger input in pulse width modulation mode)

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

Table 1.26.14. Timer A input (up/down input in event counter mode)

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

Timing requirements

(referenced to V_{CC} = 3.3V, V_{SS} = 0V, at Topr = – 20°C to 85°C / – 40°C to 85°C (*) unless otherwise specified)

* : Specify a product of –40°C to 85°C to use it.

Table 1.26.15. Timer B input (counter input in event counter mode)

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

Table 1.26.16. Timer B input (pulse period measurement mode)

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

Table 1.26.17. Timer B input (pulse width measurement mode)

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

Table 1.26.18. A-D trigger input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input LOW pulse width	125		ns

Table 1.26.19. 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		100	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 1.26.20. External interrupt $\overline{INTi}$ inputs

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

Switching characteristics (referenced to V_{CC} = 3.3V, V_{SS} = 0V at Topr = – 20°C to 85°C / – 40°C to 85°C (Note 3), CM15 = "1" unless otherwise specified)

Table 1.26.21. Memory expansion and microprocessor modes (with no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.26.1		50	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		4		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		0		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		0		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			50	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			40	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		–4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			40	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			40	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (BCLK standard)			50	ns
$t_{h(BCLK-DB)}$	Data output hold time (BCLK standard)		4		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note1)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)(Note2)		0		ns

Note 1: Calculated according to the BCLK frequency as follows:

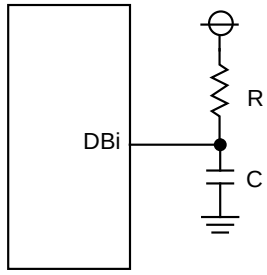
$$td(DB - WR) = \frac{10^9}{f(BCLK) \times 2} - 50 \quad [ns]$$

Note 2: This is standard value shows the timing when the output is off, and doesn't show hold time of data bus.
Hold time of data bus is different by capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in

$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.
For example, when V_{OL} = 0.2V_{CC}, C = 30pF, R = 1kΩ, hold time of output "L" level is

$$t = -30pF \times 1k\Omega \times \ln(1 - 0.2V_{CC} / V_{CC})$$
$$= 6.7ns.$$



Note 3: Specify a product of -40°C to 85°C to use it.

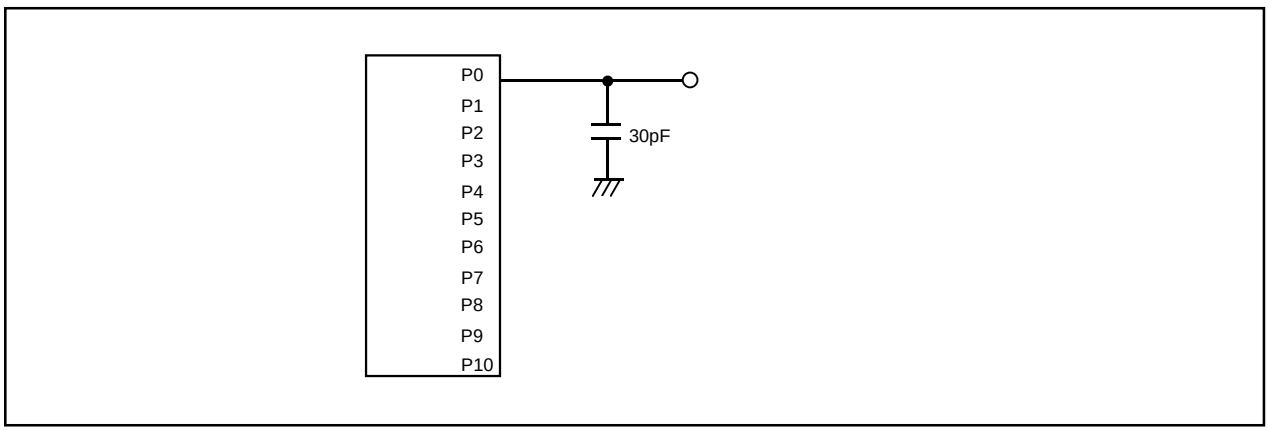


Figure 1.26.1. Port P0 to P10 measurement circuit

Switching characteristics (referenced to V_{CC} = 3.3V, V_{SS} = 0V at Topr = – 20°C to 85°C / – 40°C to 85°C (Note 3), CM15 = "1" unless otherwise specified)

Table 1.26.22. Memory expansion and microprocessor modes (when accessing external memory area with wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.26.1		50	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		4		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		0		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		0		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			50	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			40	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		– 4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			40	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			40	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (BCLK standard)			50	ns
$t_{h(BCLK-DB)}$	Data output hold time (BCLK standard)		4		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note1)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)(Note2)		0		ns

Note 1: Calculated according to the BCLK frequency as follows:

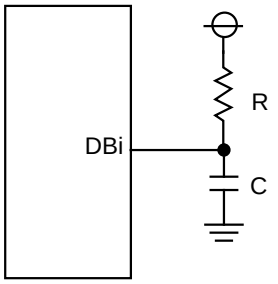
$$td(DB - WR) = \frac{10^9}{f(BCLK)} - 50 \quad [ns]$$

Note 2: This is standard value shows the timing when the output is off, and doesn't show hold time of data bus.
Hold time of data bus is different by capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in

$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.
For example, when $V_{OL} = 0.2V_{CC}$, C = 30pF, R = 1kΩ, hold time of output "L" level is

$$t = -30pF \times 1k\Omega \times \ln(1 - 0.2V_{CC} / V_{CC})$$
$$= 6.7ns.$$



Note 3: Specify a product of -40°C to 85°C to use it.

Switching characteristics (referenced to V_{CC} = 3.3V, V_{SS} = 0V at Topr = – 20°C to 85°C / – 40°C to 85°C (Note 2), CM15 = "1" unless otherwise specified)

Table 1.26.23. Memory expansion and microprocessor modes (when accessing external memory area with wait, and select multiplexed bus)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.26.1		50	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		4		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		(Note1)		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note1)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			50	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		4		ns
$t_{h(RD-CS)}$	Chip select output hold time (RD standard)		(Note1)		ns
$t_{h(WR-CS)}$	Chip select output hold time (WR standard)		(Note1)		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			40	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			40	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (BCLK standard)			50	ns
$t_{h(BCLK-DB)}$	Data output hold time (BCLK standard)		4		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note1)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note1)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time (BCLK standard)			40	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time (BCLK standard)		– 4		ns
$t_{d(AD-ALE)}$	ALE signal output delay time (Address standard)		(Note1)		ns
$t_{h(ALE-AD)}$	ALE signal output hold time (Adderss standard)		30		ns
$t_{d(AD-RD)}$	Post-address RD signal output delay time		0		ns
$t_{d(AD-WR)}$	Post-address WR signal output delay time		0		ns
$t_{dZ(RD-AD)}$	Address output floating start time			8	ns

Note 1: Calculated according to the BCLK frequency as follows:

$$th(RD - AD) = \frac{10^9}{f(BCLK) \times 2} + 0 \quad [ns]$$

$$th(WR - AD) = \frac{10^9}{f(BCLK) \times 2} + 0 \quad [ns]$$

$$th(RD - CS) = \frac{10^9}{f(BCLK) \times 2} + 0 \quad [ns]$$

$$th(WR - CS) = \frac{10^9}{f(BCLK) \times 2} + 0 \quad [ns]$$

$$td(DB - WR) = \frac{10^9 \times 3}{f(BCLK) \times 2} - 50 \quad [ns]$$

$$th(WR - DB) = \frac{10^9}{f(BCLK) \times 2} + 0 \quad [ns]$$

$$td(AD - ALE) = \frac{10^9}{f(BCLK) \times 2} - 40 \quad [ns]$$

Note 2: Specify a product of -40°C to 85°C to use it.

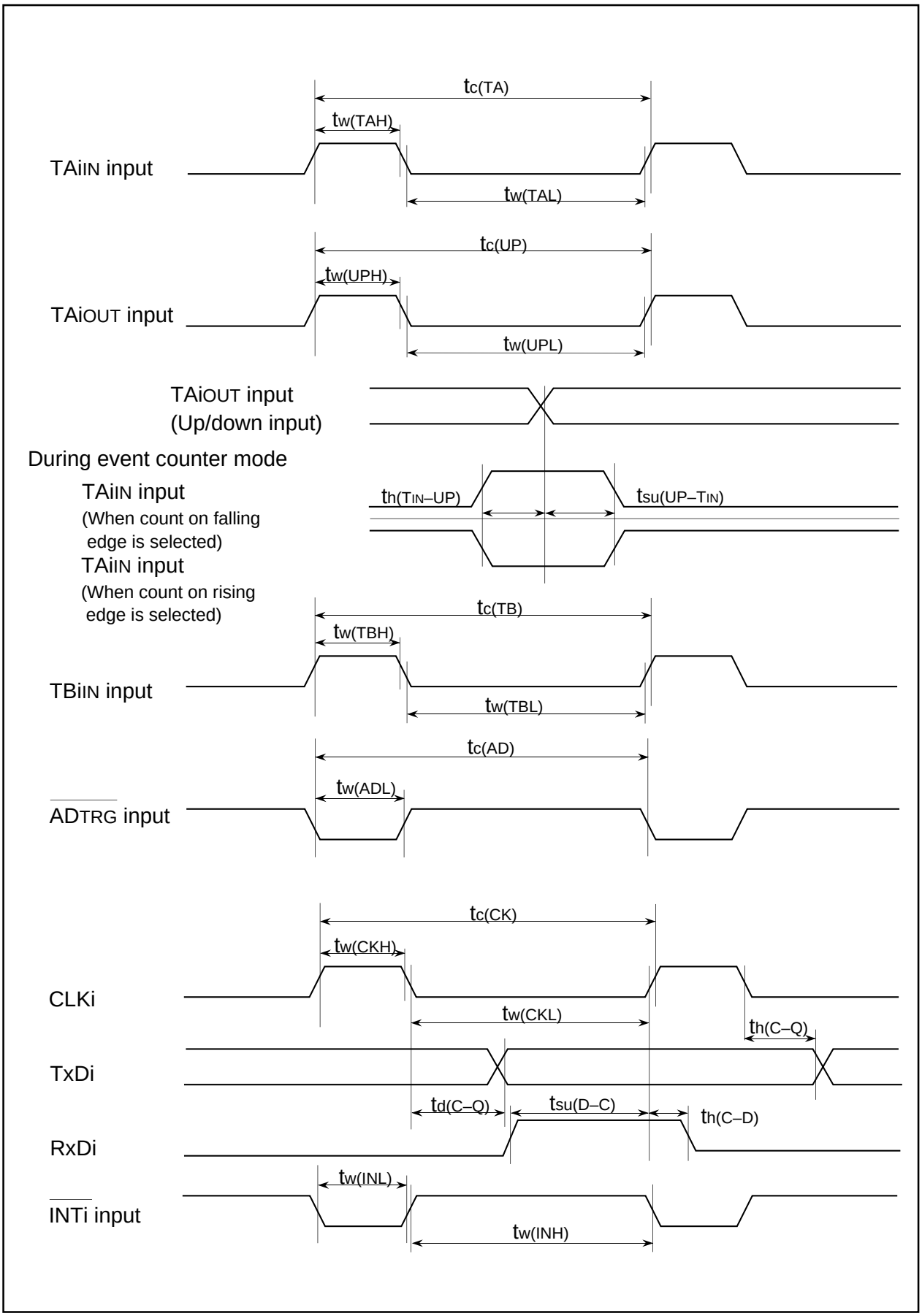
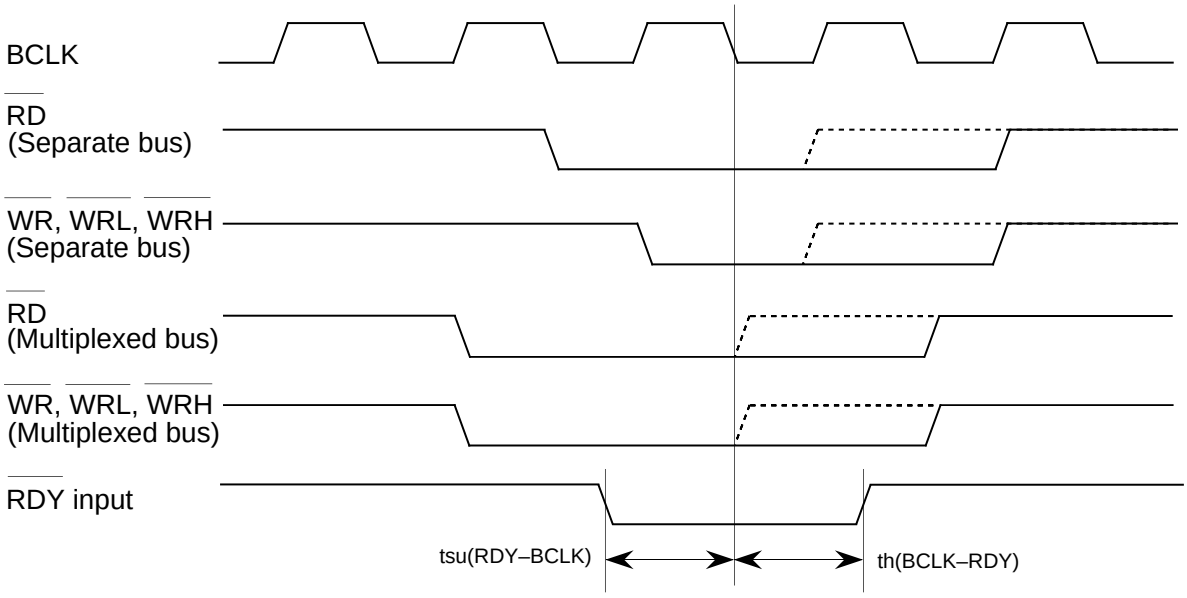


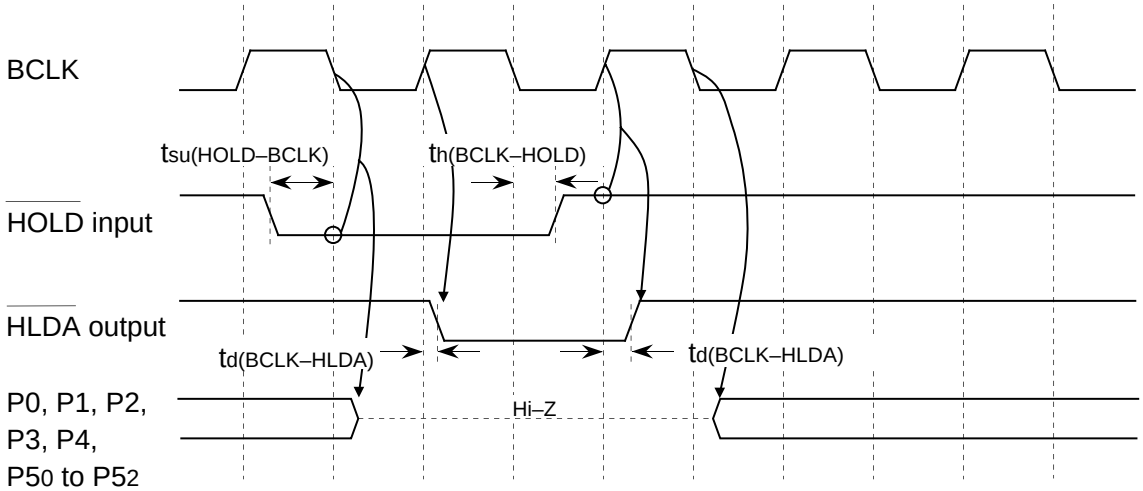
Figure 1.26.2. Timing diagram (1)

Memory Expansion Mode and Microprocessor Mode

(Valid only with wait)



(Valid with or without wait)



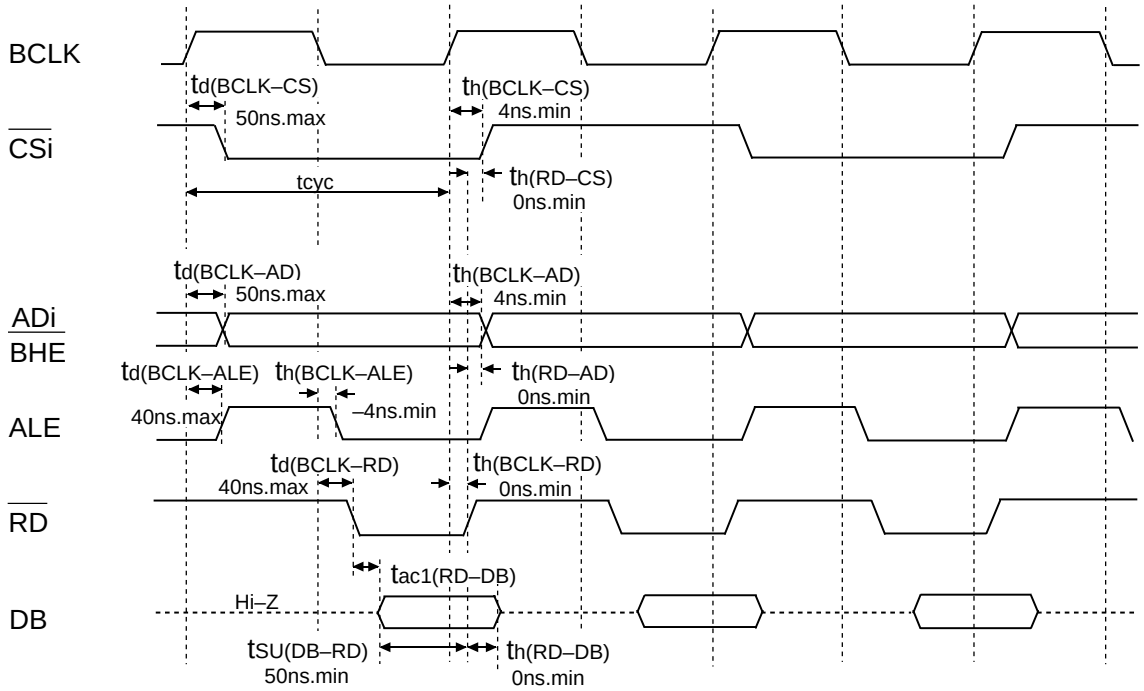
Note: The above pins are set to high-impedance regardless of the input level of the BYTE pin and bit (PM06) of processor mode register 0 selects the function of ports P40 to P43.

Measuring conditions :
- V_{CC}=3.3V
- Input timing voltage : Determined with V_{IL}=0.66V, V_{IH}=2.64V
- Output timing voltage : Determined with V_{OL}=1.65V, V_{OH}=1.65V

Figure 1.26.3. Timing diagram (2)

Memory Expansion Mode and Microprocessor Mode
(With no wait)

Read timing



Write timing

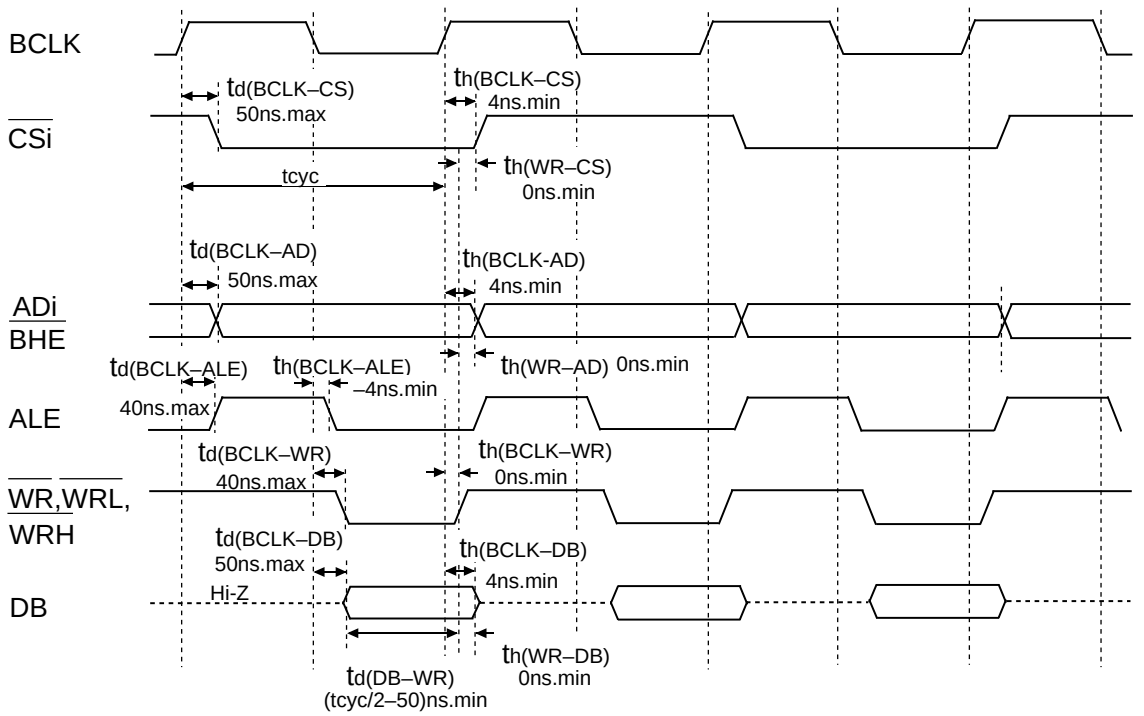
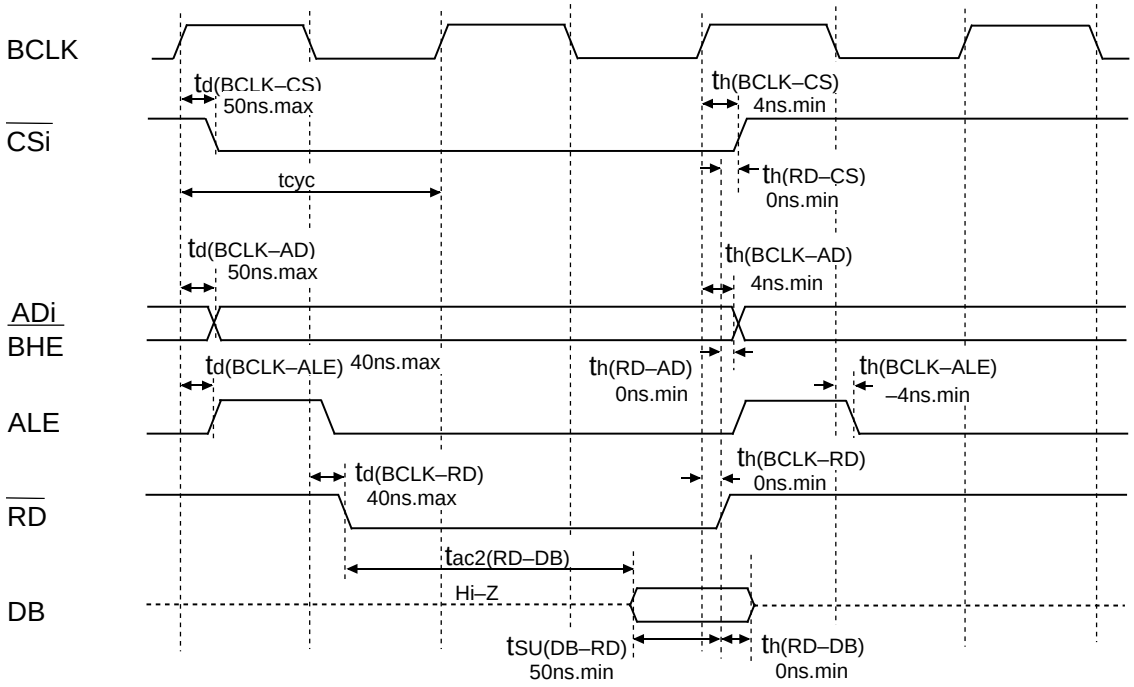


Figure 1.26.4. Timing diagram (3)

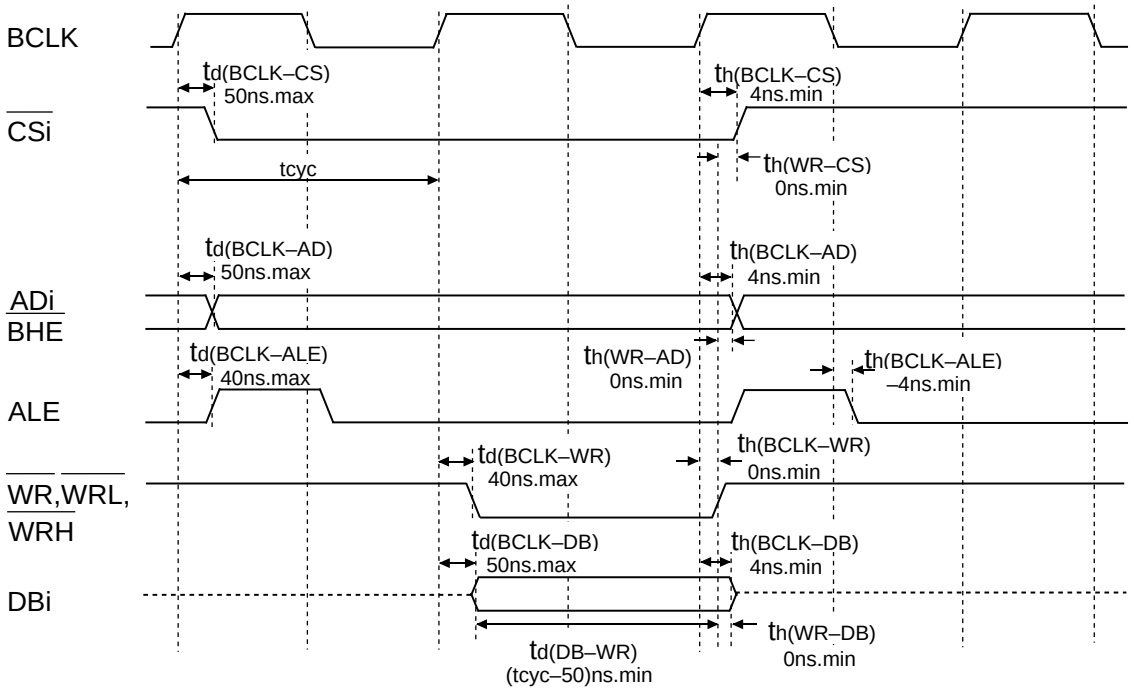
Memory Expansion Mode and Microprocessor Mode

(When accessing external memory area with wait)

Read timing



Write timing



Measuring conditions :

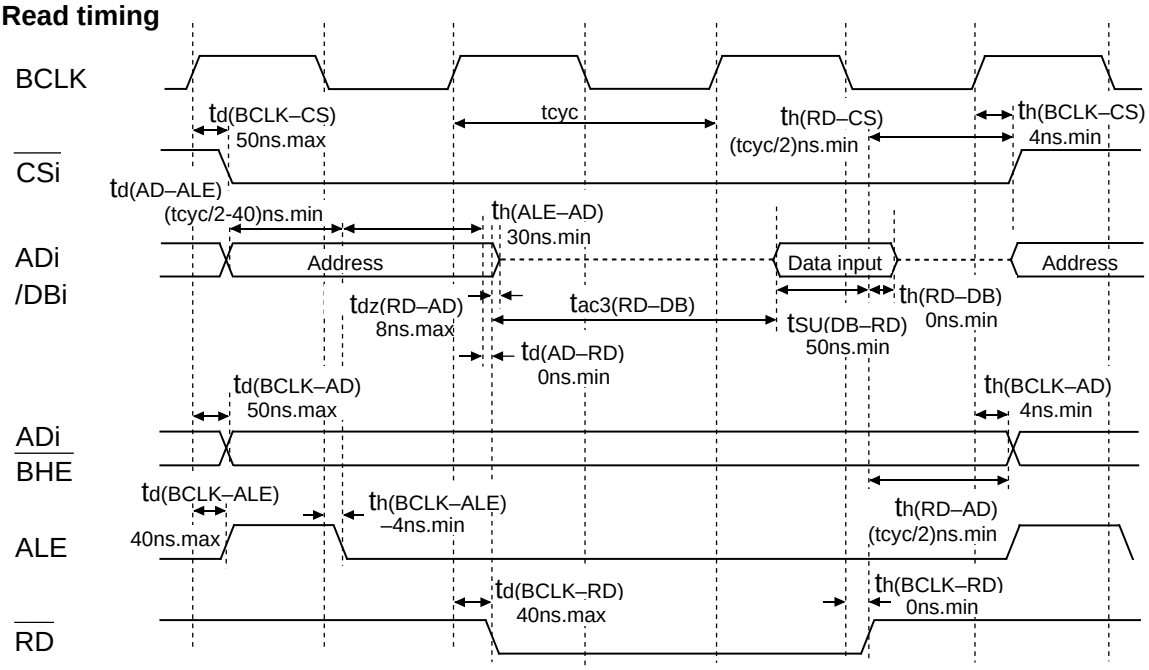
- V_{CC}=3.3V
- Input timing voltage : Determined with: V_{IL}=0.52V, V_{IH}=1.65V
- Output timing voltage : Determined with: V_{OL}=1.65V, V_{OH}=1.65V

Figure 1.26.5. Timing diagram (4)

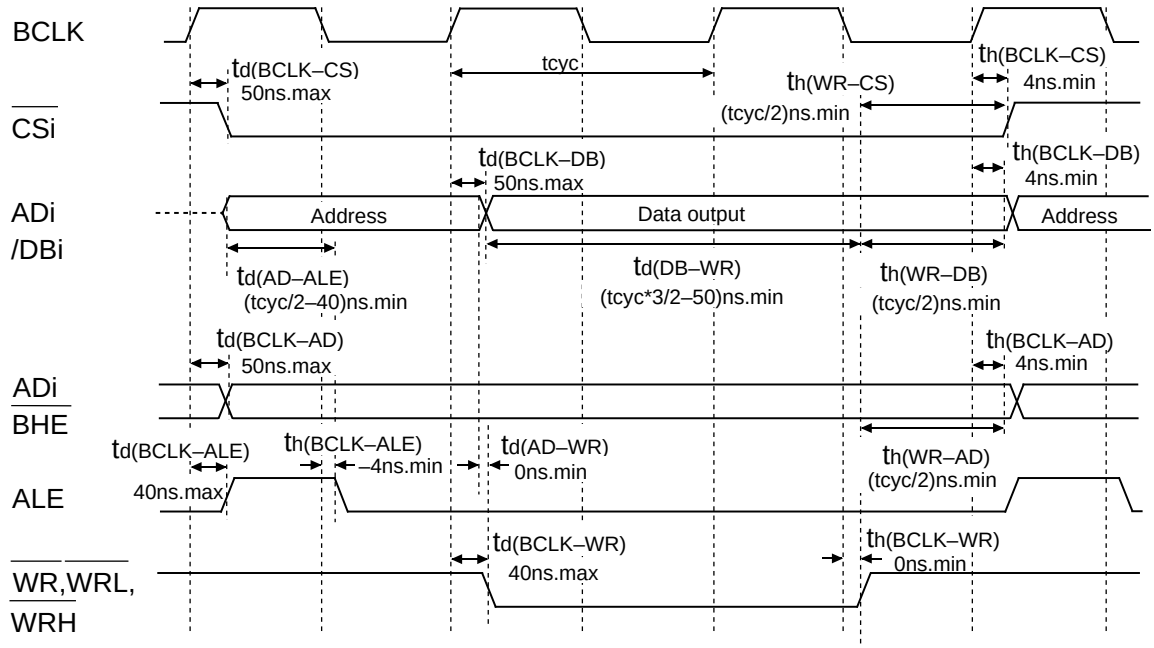
Memory Expansion Mode and Microprocessor Mode

(When accessing external memory area with wait, and select multiplexed bus)

Read timing



Write timing



Measuring conditions :

- V_{CC}=3.3V
- Input timing voltage : Determined with V_{IL}=0.52V, V_{IH}=1.65V
- Output timing voltage : Determined with V_{OL}=1.65V, V_{OH}=1.65V

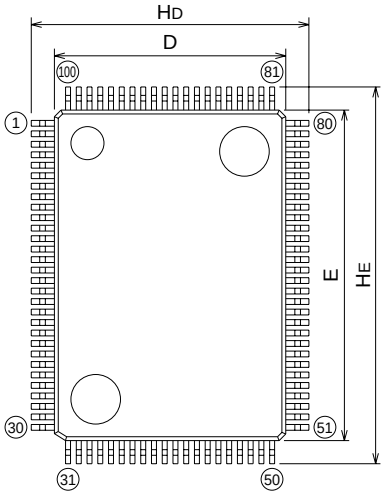
Figure 1.26.6. Timing diagram (5)

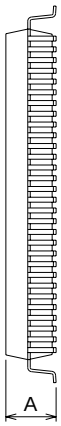
Package Outline

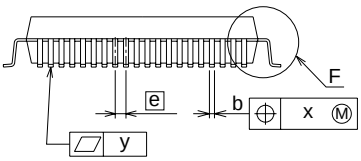
100P6S-A (MMP)

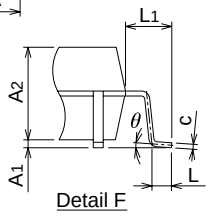
Plastic 100pin 14×20mm body QFP

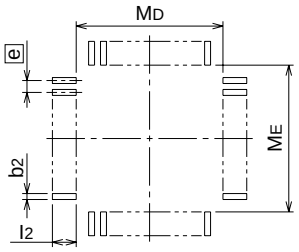
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP100-P-1420-0.65	–	1.58	Alloy 42











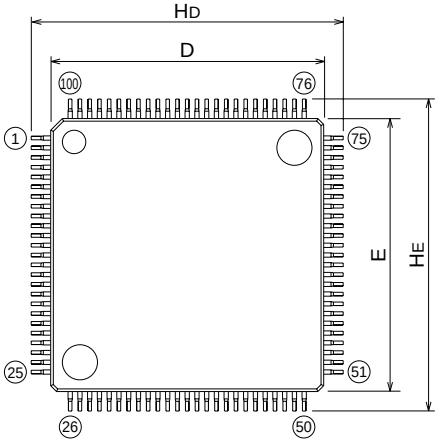
Recommended Mount Pad

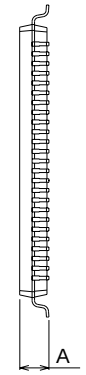
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	3.05
A1	0	0.1	0.2
A2	–	2.8	–
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	–	0.65	–
HD	16.5	16.8	17.1
HE	22.5	22.8	23.1
L	0.4	0.6	0.8
L1	–	1.4	–
x	–	–	0.13
y	–	–	0.1
θ	0°	–	10°
b2	–	0.35	–
l2	1.3	–	–
MD	–	14.6	–
ME	–	20.6	–

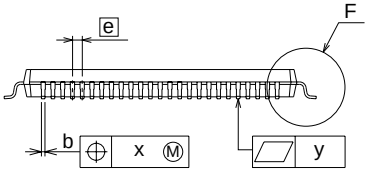
100P6Q-A (MMP)

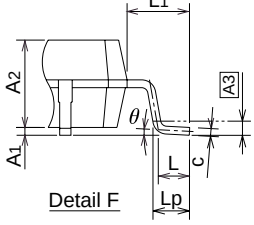
Plastic 100pin 14×14mm body LQFP

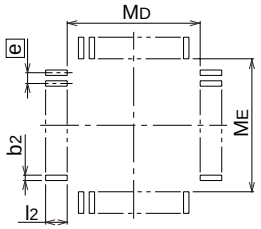
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP100-P-1414-0.50	–	0.63	Cu Alloy











Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	1.7
A1	0	0.1	0.2
A2	–	1.4	–
b	0.13	0.18	0.28
c	0.105	0.125	0.175
D	13.9	14.0	14.1
E	13.9	14.0	14.1
e	–	0.5	–
HD	15.8	16.0	16.2
HE	15.8	16.0	16.2
L	0.3	0.5	0.7
L1	–	1.0	–
Lp	0.45	0.6	0.75
A3	–	0.25	–
x	–	–	0.08
y	–	–	0.1
θ	0°	–	10°
b2	–	0.225	–
l2	0.9	–	–
MD	–	14.4	–
ME	–	14.4	–

SFR difference between M16C/62N and M16C/30L

Address	Symbol	M16C/62N	M16C/30L
0005	PM1	**Processor mode register 1** bit0:Reserved bit bit1:Nothing is assigned bit2:Nothing is assigned **PM13:Internal reserved area expansion bit** **PM14:Memory area expantion bit** **PM15:** bit6:Reserved bit PM17:Wait bit	**Processor mode register 1** bit0:Reserved bit bit1:Nothing is assigned bit2:Nothing is assigned **bit3:Reserved bit** **bit4:Reserved bit** **bit5:Reserved bit** bit6:Reserved bit PM17:Wait bit
000A	PRCR	**Protect register** PRC0:Enables writing to system clock control registers 0 and 1 PRC1:Enables writing to processor mode registers 0 and 1 **PRC2:Enables writing to port P9 direction register and SI/O control register** bit3-bit7:Nothing is assigned	**Protect register** PRC0:Enables writing to system clock control registers 0 and 1 PRC1:Enables writing to processor mode registers 0 and 1 **PRC2:Enables writing to port P9 direction register** bit3-bit7:Nothing is assigned
000B	DBR	**Data bank register**	**Reserved register**
0030	SAR1	**DMA1 source pointer**	**Reserved register**
0031	SAR1	**DMA1 source pointer**	**Reserved register**
0032	SAR1	**DMA1 source pointer**	**Reserved register**
0034	DAR1	**DMA1 destination pointer**	**Reserved register**
0035	DAR1	**DMA1 destination pointer**	**Reserved register**
0036	DAR1	**DMA1 destination pointer**	**Reserved register**
0038	TCR1	**DMA1 transfer counter**	**Reserved register**
0039	TCR1	**DMA1 transfer counter**	**Reserved register**
003C	DM1CON	**DMA1 control register**	**Reserved register**
0044	INT3IC	**INT3 interrupt control register**	**Reserved register**
0045	TB5IC	**Timer B5 interrupt control register**	**Reserved register**
0046	TB4IC	**Timer B4 interrupt control register**	**Reserved register**
0047	TB3IC	**Timer B3 interrupt control register**	**Reserved register**
0048	S4IC/INT5IC	**SI/O4,INT5 interrupt control register**	**Reserved register**
0049	S3IC/INT4IC	**SI/O3,INT4 interrupt control register**	**Reserved register**
004C	DM1IC	**DMA1 interrupt control register**	**Reserved register**
0058	TA3IC	**Timer A3 interrupt control register**	**Reserved register**
0059	TA4IC	**Timer A4 interrupt control register**	**Reserved register**
005A	TB0IC	**Timer B0 interrupt control register**	**Reserved register**
0340	TBSR	**Timer B3, 4, 5 count start flag**	**Reserved register**
0342	TA11	**Timer A1-1 register**	**Reserved register**
0343	TA11	**Timer A1-1 register**	**Reserved register**
0344	TA21	**Timer A2-1 register**	**Reserved register**
0345	TA21	**Timer A2-1 register**	**Reserved register**
0346	TA41	**Timer A4-1 register**	**Reserved register**
0347	TA41	**Timer A4-1 register**	**Reserved register**
0348	INVC0	**Three-phase PWM control register 0**	**Reserved register**
0349	INVC1	**Three-phase PWM control register 1**	**Reserved register**
034A	IDB0	**Three-phase output buffer register 0**	**Reserved register**
034B	IDB1	**Three-phase output buffer register 1**	**Reserved register**
034C	DTT	**Dead time timer**	**Reserved register**
034D	ICTB2	**Timer B2 interrupt occurrence frequency set counter**	**Reserved register**
0350	TB3	**Timer B3 register**	**Reserved register**
0351	TB3	**Timer B3 register**	**Reserved register**
0352	TB4	**Timer B4 register**	**Reserved register**
0353	TB4	**Timer B4 register**	**Reserved register**
0354	TB5	**Timer B5 register**	**Reserved register**
0355	TB5	**Timer B5 register**	**Reserved register**
035B	TB3MR	**Timer B3 mode register**	**Reserved register**
035C	TB4MR	**Timer B4 mode register**	**Reserved register**
035D	TB5MR	**Timer B5 mode register**	**Reserved register**

Don't access a Reserved register.

SFR difference between M16C/62N and M16C/30L

Address	Symbol	M16C/62N	M16C/30L
035F	IFSR	**Interrupt cause select register** IFSR0:INT0 interrupt polarity switching bit IFSR1:INT1 interrupt polarity switching bit IFSR2:INT2 interrupt polarity switching bit **IFSR3:INT3 interrupt polarity switching bit** **IFSR4:INT4 interrupt polarity switching bit** **IFSR5:INT5 interrupt polarity switching bit** **IFSR6:Interrupt request cause select bit** **IFSR7:Interrupt request cause select bit**	**Interrupt cause select register** IFSR0:INT0 interrupt polarity switching bit IFSR1:INT1 interrupt polarity switching bit IFSR2:INT2 interrupt polarity switching bit **bit3:Reserved bit** **bit4:Reserved bit** **bit5:Reserved bit** **bit6:Reserved bit** **bit7:Reserved bit**
0360	S3TRR	**SI/O3 transmit/receive register**	**Reserved register**
0362	S3C	**SI/O3 control register**	**Reserved register**
0363	S3BRG	**SI/O3 bit rate generator**	**Reserved register**
0364	S4TRR	**SI/O4 transmit/receive register**	**Reserved register**
0366	S4C	**SI/O4 control register**	**Reserved register**
0367	S4BRG	**SI/O4 bit rate generator**	**Reserved register**
0380	TABSR	**Count start flag** TA0S:Timer A0 count start flag TA1S:Timer A1 count start flag TA2S:Timer A2 count start flag **TA3S:Timer A3 count start flag** **TA4S:Timer A4 count start flag** **TB0S:Timer B0 count start flag** TB1S:Timer B1 count start flag TB2S:Timer B2 count start flag	**Count start flag** TA0S:Timer A0 count start flag TA1S:Timer A1 count start flag TA2S:Timer A2 count start flag **bit3:Reserved bit** **bit4:Reserved bit** **bit5:Reserved bit** TB1S:Timer B1 count start flag TB2S:Timer B2 count start flag
0382	ONSF	**One-shot start flag** TA0OS:Timer A0 one-shot start flag TA1OS:Timer A1 one-shot start flag TA2OS:Timer A2 one-shot start flag **TA3OS:Timer A3 one-shot start flag** **TA4OS:Timer A4 one-shot start flag** bit5:Nothing is assigned **TA0TGL:Timer A0 event/trigger select bit** **TA0TGH: "00","01","10"and"11" can be chosen.**	**One-shot start flag** TA0OS:Timer A0 one-shot start flag TA1OS:Timer A1 one-shot start flag TA2OS:Timer A2 one-shot start flag **bit3:Reserved bit** **bit3:Reserved bit** bit5:Nothing is assigned **TA0TGL:Timer A0 event/trigger select bit** **TA0TGH: "00 ","01"and"11" can be chosen. "10" can't be chosen.**
0383	TRGSR	**Trigger select register** TA1TGL:Timer A1 event/trigger select bit TA1TGH: "00","01","10"and"11" can be chosen. **TA2TGL:Timer A2 event/trigger select bit** **TA2TGH: "00","01","10"and"11" can be chosen.** **TA3TGL:Timer A3 event/trigger select bit** **TA3TGH: "00","01","10"and"11" can be chosen.** **TA4TGL:Timer A4 event/trigger select bit** **TA4TGH: "00","01","10"and"11" can be chosen.**	**Trigger select register** TA1TGL:Timer A1 event/trigger select bit TA1TGH: "00","01","10"and"11" can be chosen. **TA2TGL:Timer A2 event/trigger select bit** **TA2TGH: "00","01"and"10" can be chosen. "11" can't be chosen.** **bit4:Reserved bit** **bit5:Reserved bit** **bit6:Reserved bit** **bit7:Reserved bit**
0384	UDF	**Up-down flag** TA0UD:Timer A0 up/down flag TA1UD:Timer A1 up/down flag TA2UD:Timer A2 up/down flag **TA3UD:Timer A3 up/down flag** **TA4UD:Timer A4 up/down flag** TA2P:Timer A2 two-phase pulse signal processing select bit **TA3P:Timer A3 two-phase pulse signal processing select bit** **TA4P:Timer A4 two-phase pulse signal processing select bit**	**Up-down flag** TA0UD:Timer A0 up/down flag TA1UD:Timer A1 up/down flag TA2UD:Timer A2 up/down flag **bit3:Reserved bit** **bit4:Reserved bit** TA2P:Timer A2 two-phase pulse signal processing select bit **bit6:Reserved bit** **bit7:Reserved bit**
038C	TA3	**Timer A3**	**Reserved register**
038D	TA3	**Timer A3**	**Reserved register**
038E	TA4	**Timer A4**	**Reserved register**
038F	TA4	**Timer A4**	**Reserved register**
0390	TB0	**Timer B0**	**Reserved register**
0391	TB0	**Timer B0**	**Reserved register**
0399	TA3MR	**Timer A3 mode register**	**Reserved register**
039A	TA4MR	**Timer A4 mode register**	**Reserved register**
039B	TB0MR	**Timer B0 mode register**	**Reserved register**

Don't access a Reserved register.

SFR difference between M16C/62N and M16C/30L

Address	Symbol	M16C/62N	M16C/30L
039C	TB1MR	Timer B1 mode register Event counter mode TMOD0:Operation mode select bit TMOD1: MR0:Count polarity select bit MR1: MR2:Nothing is assigned MR3:Invalid **TCK1:Event clock select "0" and "1" can be chosen.**	Timer B1 mode register Event counter mode TMOD0:Operation mode select bit TMOD1: MR0:Count polarity select bit MR1: MR2:Nothing is assigned MR3:Invalid **TCK1:Event clock select "0 " can be chosen. "1" can't be chosen.**
03B4	FIDR	**Flash identification register**	**Reserved register**
03B7	FMR0	**Flash memory control register 0**	**Reserved register**
03B8	DM0SL	**DMA0 request cause select register** **DMA0 request factors** Falling edge of INT0 pin Software trigger Timer A0 Timer A1 Timer A2 **Timer A3** **Timer A4** two edges of INT0 pin **Timer B0** Timer B1 Timer B2 **Timer B3** **Timer B4** **Timer B5** UART0 transmit UART0 receive UART2 transmit UART2 receive UART1 transmit A-D conversion	**DMA0 request cause select register** **DMA0 request factors** Falling edge of INT0 pin Software trigger Timer A0 Timer A1 Timer A2 two edges of INT0 pin Timer B1 Timer B2 UART0 transmit UART0 receive UART2 transmit UART2 receive UART1 transmit A-D conversion
03BA	DM1SL	**DMA1 request cause select register**	**Reserved register**
03BC	CRCD	**CRC data register**	**Reserved register**
03BD	CRCIN	**CRC data register**	**Reserved register**
03BE	CRCIN	**CRC input register**	**Reserved register**
03D4	ADCON2	**A-D control register 2** SMP:A-D conversion method select bit **ADGSL0:Analog input group select bit** **ADGSL1:** bit3:Reserved bit bit4-bit7:Nothing is assigned	**A-D control register 2** SMP:A-D conversion method select bit **bit1:Reserved bit** **bit2:Reserved bit** bit3:Reserved bit bit4-bit7:Nothing is assigned
03D6	ADCON0	**A-D control register 0** CH0:Analog input pin select bit CH1: CH2: **MD0:A-D operation mode select bit 0** **MD1: "00","01","10"and"11" can be chosen** TRG:Trigger select bit ADST:A-D conversion start flag CKS0:Frequency select bit 0	**A-D control register 0** CH0:Analog input pin select bit CH1: CH2: **MD0:A-D operation mode select bit 0** **MD1: "00 " can be chosen. "01","10"and"11" can't be chosen.** TRG:Trigger select bit ADST:A-D conversion start flag CKS0:Frequency select bit 0
03D7	ADCON1	**A-D control register 1** **SCAN0:A-D sweep pin select bit** **SCAN1:** **MD2:A-D operation mode select bit 1** BITS:8/10-bit mode select bit CKS1:Frequency select bit 1 VCUT:Vref connect bit OPA0:External op-amp connection mode bit OPA1:	**A-D control register 1** **bit0:Reserved bit** **bit1:Reserved bit** **bit2:Reserved bit** BITS:8/10-bit mode select bit CKS1:Frequency select bit 1 VCUT:Vref connect bit OPA0:External op-amp connection mode bit OPA1:
03D8	DA0	**D-A register 0**	**Reserved register**
03DA	DA1	**D-A register 1**	**Reserved register**
03DC	DACON	**D-A control register**	**Reserved register**

Don't access a Reserved register.

Renesas Technology Corp.
Nippon Bldg.,6-2,Otemachi 2-chome,Chiyoda-ku,Tokyo,100-0004 Japan

Printed in Japan (ROD) II
New publication, effective June. 2002.
Specifications subject to change without notice.

RENESAS
Renesas Technology Corp.