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

Overview

The M16C/62P 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 and 128-pin plastic molded QFP. These single-chip microcomputers operate using sophisticated instructions featuring a high level of instruction efficiency. With 1M bytes of address space, they are capable of executing instructions at high speed. In addition, this microcomputer contains a multiplier and DMAC which combined with fast instruction processing capability, makes it suitable for control of various OA, communication, and industrial equipment which requires high-speed arithmetic/logic operations.

Applications

Audio, cameras, office/communications/portable/industrial equipment, etc

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

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

Performance Outline

Table 1.1.1 lists performance outline of M16C/62P group.

Table 1.1.1. Performance outline of M16C/62P group

Item		Performance	
Number of basic instructions		91 instructions	
Shortest instruction execution time		41.7 ns (f(BCLK)= 24MHz, V_{CC1}= 3.0V to 5.5V) 100 ns (f(BCLK)= 10MHz, V_{CC1}= 2.7V to 5.5V)	
Memory capacity	ROM	(See the product list)	
	RAM	(See the product list)	
I/O port	100-pin version P0 to P10 (except P8$_5$)	8 bits x 10, 7 bits x 1	P0 to P5: V_{CC2} ports P6 to P10: V_{CC1} ports
	128-pin version P0 to P14 (except P8$_5$)	8 bits x 13, 7 bits x 1, 2 bits x 1	P0 to P5, P12, P13: V_{CC2} ports P6 to P10, P11, P14: V_{CC1} ports
Input port	P8$_5$	1 bit x 1 ($\overline{NMI}$ pin level judgment): V_{CC1} ports	
Multifunction timer			
	Output	16 bits x 5 channels (TA0, TA1, TA2, TA3, TA40)	
	Input	16 bits x 6 channels (TB0, TB1, TB2, TB3, TB4, TB5)	
Serial I/O		3 channels (UART0, UART1, UART2) UART, clock synchronous, I^{2}C bus[1] (option[3]), or IE bus[2] (option[3]) 2 channels (SI/O3, SI/O4) Clock synchronous	
A-D converter		10 bits x (8 x 3 + 2) channels	
D-A converter		8 bits x 2	
DMAC		2 channels (trigger: 25 sources)	
CRC calculation circuit		CRC-CCITT	
Watchdog timer		15 bits x 1 (with prescaler)	
Interrupt		25 internal and 8 external sources, 4 software sources, 7 levels	
Clock generation circuit		4 circuits • Main clock, • Sub-clock (These circuits contain a built-in feedback resistor and external ceramic/quartz oscillator) • Ring oscillator (for main-clock oscillation stop detect function) • PLL frequency synthesizer	
Voltage detection circuit		Present (option[3])	
Power supply voltage		V_{CC1}=3.0V to 5.5V, V_{CC2}=3.0V to V_{CC1}(f(BCLK)=24MHz) V_{CC1}=V_{CC2}=2.7V to 5.5V (f(BCLK)=10MHz)	
Flash memory	Program/erase voltage	3.3V ± 0.3V or 5.0V ± 0.5V	
	Number of program/erase	100 times	
Power consumption		14mA (V_{CC1}=V_{CC2}=5V, f(BCLK)=24MHz) 8mA (V_{CC1}=V_{CC2}=3V, f(BCLK)=10MHz) 1.8µA (V_{CC1}=V_{CC2}=3V, f(X_{CIN})=32kHz, when wait mode)	
I/O characteristics	I/O withstand voltage	5.0V	
	Output current	5mA	
Memory expansion		Available (to 4M bytes)	
Operating ambient temperature		-20 to 85°C -40 to 85°C (option[3])	
Device configuration		CMOS high performance silicon gate	
Package		100-pin and 128-pin plastic mold QFP	

Notes:
1. I^{2}C Bus is a registered trademark of PHILIPS.
2. IE Bus is a registered trademark of NEC.
3. If you desire this option, please so specify.

Block Diagram

Figure 1.1.1 is a block diagram of the M16C/62P group.

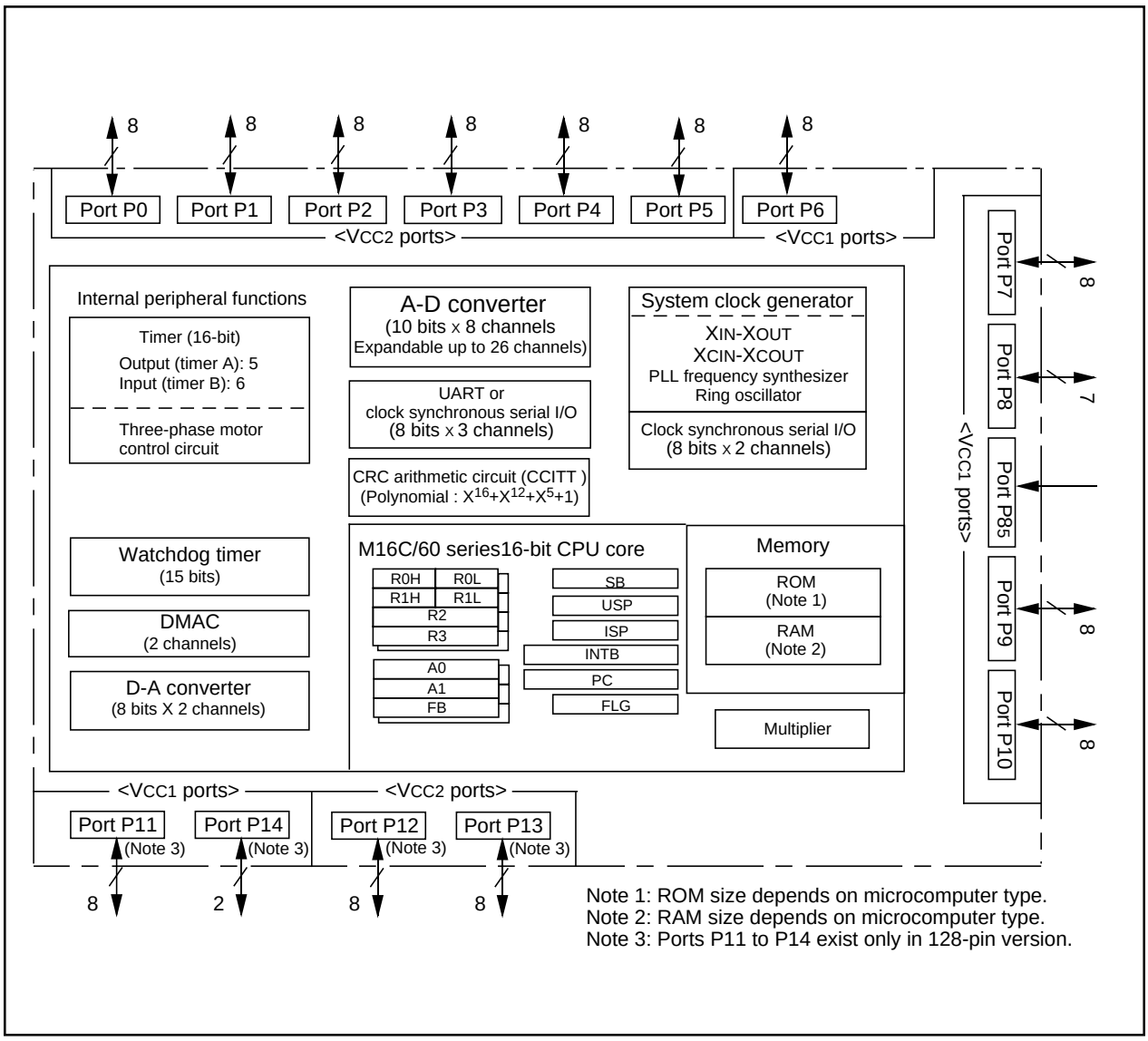


Figure 1.1.1. Block Diagram

Product List

Tables 1.1.2 and 1.1.3 list the M16C/62P group products and Figure 1.1.2 shows the type numbers, memory sizes and packages.

Table 1.1.2. Product List (1)

As of January 2003

Type No.	ROM capacity	RAM capacity	Package type	Remarks
M30622M6P-XXXFP **	48K bytes	4K bytes	100P6S-A	MASK ROM version
M30622M6P-XXXGP **			100P6Q-A	
M30622M8P-XXXFP **	64K bytes	4K bytes	100P6S-A	
M30622M8P-XXXGP **			100P6Q-A	
M30622MAP-XXXFP **	96K bytes	5K bytes	100P6S-A	
M30622MAP-XXXGP **			100P6Q-A	
M30620MCP-XXXFP **	128K bytes	10K bytes	100P6S-A	
M30620MCP-XXXGP **			100P6Q-A	
M30622MEP-XXXFP **	192K bytes	12K bytes	100P6S-A	
M30622MEP-XXXGP **			100P6Q-A	
M30623MEP-XXXGP **			128P6Q-A	
M30622MGP-XXXFP **	256K bytes	12K bytes	100P6S-A	
M30622MGP-XXXGP **			100P6Q-A	
M30623MGP-XXXGP **			128P6Q-A	
M30624MGP-XXXFP **		20K bytes	100P6S-A	
M30624MGP-XXXGP **			100P6Q-A	
M30625MGP-XXXGP **			128P6Q-A	
M30622MWP-XXXFP **	320K bytes	16K bytes	100P6S-A	
M30622MWP-XXXGP **			100P6Q-A	
M30623MWP-XXXGP **			128P6Q-A	
M30624MWP-XXXFP **		24K bytes	100P6S-A	
M30624MWP-XXXGP **			100P6Q-A	
M30625MWP-XXXGP **			128P6Q-A	
M30626MWP-XXXFP **		31K bytes	100P6S-A	
M30626MWP-XXXGP **			100P6Q-A	
M30627MWP-XXXGP **			128P6Q-A	
M30622MHP-XXXFP **	384K bytes	16K bytes	100P6S-A	
M30622MHP-XXXGP **			100P6Q-A	
M30623MHP-XXXGP **			128P6Q-A	
M30624MHP-XXXFP **		24K bytes	100P6S-A	
M30624MHP-XXXGP **			100P6Q-A	
M30625MHP-XXXGP **			128P6Q-A	
M30626MHP-XXXFP **		31K bytes	100P6S-A	
M30626MHP-XXXGP **			100P6Q-A	
M30627MHP-XXXGP **			128P6Q-A	

* : Under planning
** : Under development

Table 1.1.3. Product List (2)

As of January 2003

Type No.	ROM capacity	RAM capacity	Package type	Remarks
M30622F8PFP **	64K bytes	4K bytes	100P6S-A	Flash memory version
M30622F8PGP **			100P6Q-A	
M30620FCPFP **	128K bytes	10K bytes	100P6S-A	
M30620FCPGP **			100P6Q-A	
M30624FGPFP **	256K bytes	20K bytes	100P6S-A	
M30624FGPGP **			100P6Q-A	
M30625FGPGP **			128P6Q-A	
M30626FHPFP **	384K bytes	31K bytes	100P6S-A	
M30626FHPGP **			100P6Q-A	
M30627FHPGP **			128P6Q-A	
M30626FJPFP *	512K bytes	31K bytes	100P6S-A	
M30626FJPGP *			100P6Q-A	
M30627FJPGP *			128P6Q-A	
M30620SPFP **	—	10K bytes	100P6S-A	External ROM version
M30620SPGP **			100P6Q-A	
M30622SPFP **		4K bytes	100P6S-A	
M30622SPGP **			100P6Q-A	

* : Under planning
** : Under development

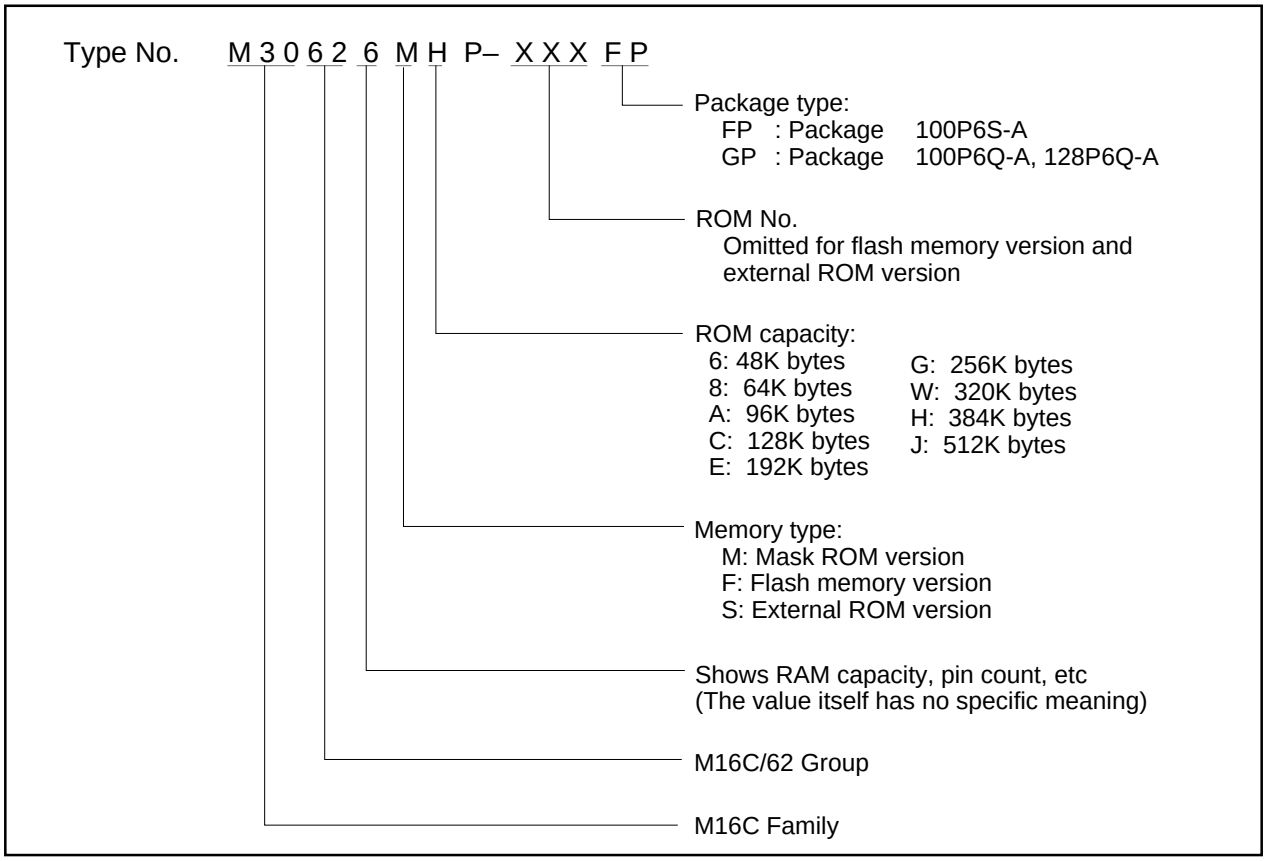
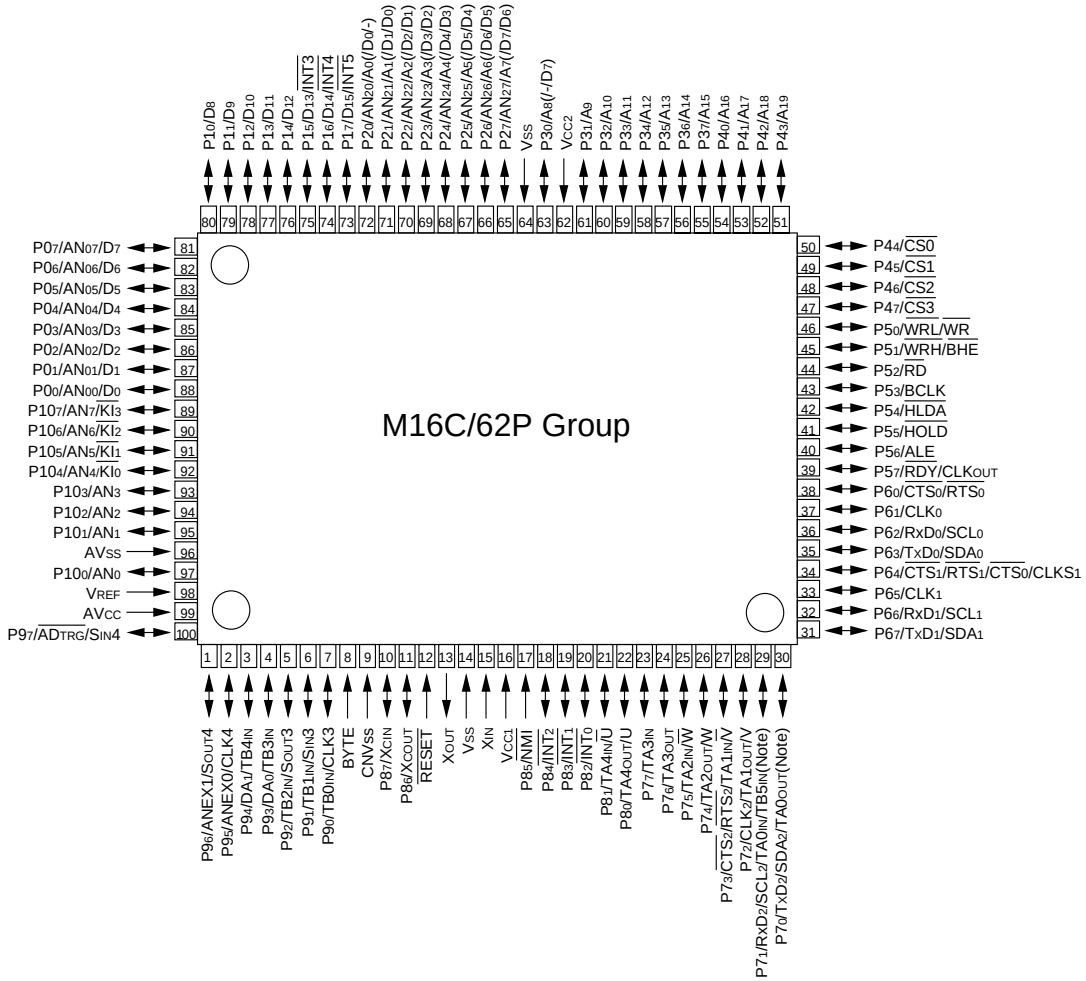


Figure 1.1.2. Type No., Memory Size, and Package

Pin Configuration

Figures 1.1.3 to 1.1.5 show the pin configurations (top view).

PIN CONFIGURATION (top view)

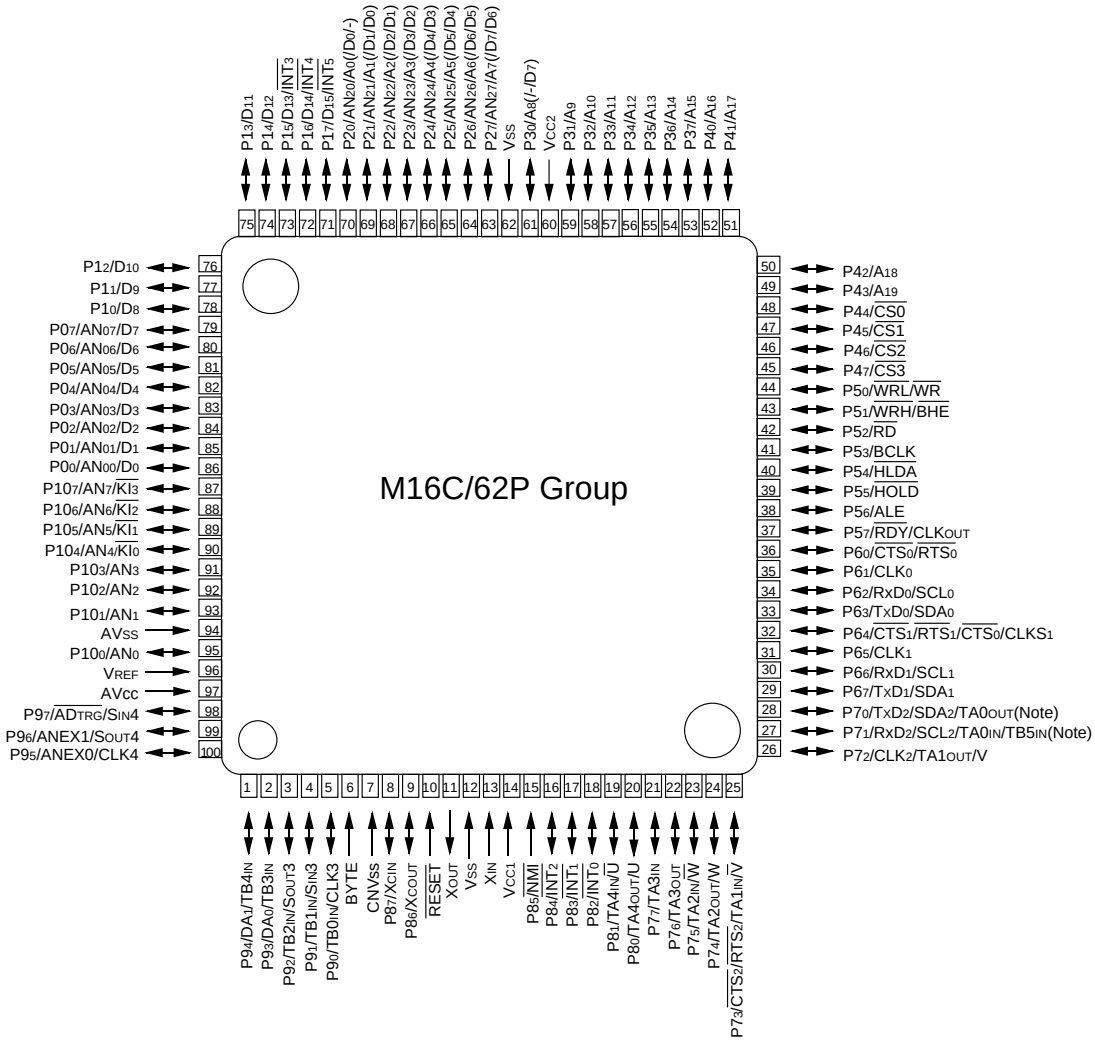


Package: 100P6S-A

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

Figure 1.1.3. Pin Configuration (Top View)

PIN CONFIGURATION (top view)

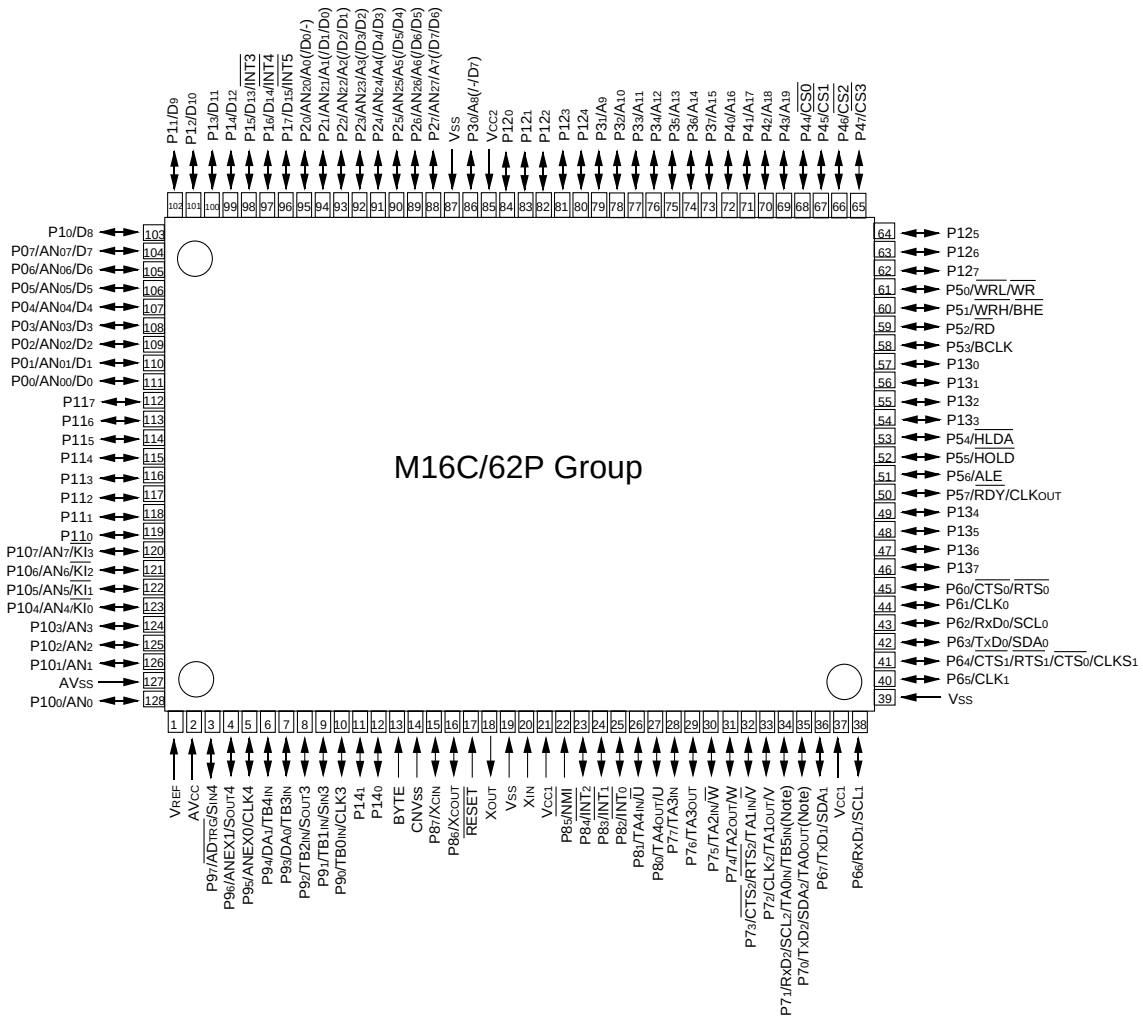


Package: 100P6Q-A

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

Figure 1.1.4. Pin Configuration (Top View)

PIN CONFIGURATION (top view)



Package: 128P6Q-A

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

Figure 1.1.5. Pin Configuration (Top View)

Table 1.1.4 Pin Description (100-pin and 128-pin Packages) (Continued)

Pin name	Signal name	I/O type	Power supply	Function
V_{CC1}, V_{CC2}, V_{SS}	Power supply input		—	Apply 2.7V to 5.5 V to the V_{CC1} and V_{CC2} pins and 0 V to the V_{SS} pin. The Vcc apply condition is that $V_{CC2} \leq V_{CC1}$ (Note)
CNV$_{SS}$	CNV$_{SS}$	Input	V_{CC1}	This pin switches between processor modes. Connect this pin to V_{SS} pin when after a reset you want to start operation in single-chip mode (memory expansion mode) or the V_{CC1} pin when starting operation in microprocessor mode.
$\overline{\text{RESET}}$	Reset input	Input	V_{CC1}	"L" on this input resets the microcomputer.
X_{IN} X_{OUT}	Clock input Clock output	Input Output	V_{CC1}	These pins are provided for the main clock generating circuit input/output. 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 operating in single-chip mode.
AV$_{CC}$	Analog power supply input			This pin is a power supply input for the A-D converter. Connect this pin to V_{CC1}.
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	V_{CC2}	This is an 8-bit CMOS I/O port. This port has an input/output select direction register, allowing each pin in that port to be directed for input or output individually. If any port is set for input, selection can be made for it in a program whether or not to have a pull-up resistor in 4 bit units. This selection is unavailable in memory extension and microprocessor modes. This port can function as input pins for the A-D converter when so selected in a program.
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	V_{CC2}	This is an 8-bit I/O port equivalent to P0. $P1_5$ to $P1_7$ also function as $\overline{\text{INT}}$ interrupt input pins as selected by a program.
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	V_{CC2}	This is an 8-bit I/O port equivalent to P0. This port can function as input pins for the A-D converter when so selected in a program.
A_0 to A_7		Output		These pins output 8 low-order address bits (A_0 to 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 to D_7) and output 8 low-order address bits (A_0 to 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 to D_6) and output address (A_1 to A_7) separated in time by multiplexing. They also output address (A_0).
$P3_0$ to $P3_7$	I/O port P3	Input/output	V_{CC2}	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 to 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 to A_{15}).
$P4_0$ to $P4_7$	I/O port P4	Input/output	V_{CC2}	This is an 8-bit I/O port equivalent to P0.
$\overline{A_{16}}$ to $\overline{A_{19}}$, $\overline{CS_0}$ to $\overline{CS_3}$		Output Output		These pins output A_{16} to A_{19} and $\overline{CS_0}$ to $\overline{CS_3}$ signals. A_{16} to A_{19} are 4 high- order address bits. $\overline{CS_0}$ to $\overline{CS_3}$ are chip select signals used to specify an access space.

Note: In this manual, hereafter, V_{CC} refers to V_{CC1} unless otherwise noted.

Table 1.1.5 Pin Description (100-pin and 128-pin Packages) (Continued)

Pin name	Signal name	I/O type	Power supply	Function
P5$_0$ to P5$_7$	I/O port P5	Input/output	V_{CC2}	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 program.
$\overline{WRL}$ / $\overline{WR}$, $\overline{WRH}$ / $\overline{BHE}$, $\overline{RD}$, BCLK, $\overline{HLDA}$, HOLD, $\overline{ALE}$, $\overline{RDY}$		Output Output Output Output Output Input Output Input		Output $\overline{WRL}$/$\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, $\overline{RD}$, BCLK, $\overline{HLDA}$, and ALE signals. $\overline{WRL}$/$\overline{WR}$ and $\overline{WRH}$/$\overline{BHE}$ are switchable in a program. Note that $\overline{WRL}$ and $\overline{WRH}$ are always used as a pair, so as $\overline{WR}$ and $\overline{BHE}$. ■ $\overline{WRL}$, $\overline{WRH}$, and $\overline{RD}$ selected If the external data bus is 16 bits wide, data are written to even addresses when the $\overline{WRL}$ signal is low, and written to odd addresses when the $\overline{WRH}$ signal is low. Data are read out when the $\overline{RD}$ signal is low. ■ $\overline{WR}$, $\overline{BHE}$, and $\overline{RD}$ selected Data are written when the $\overline{WR}$ signal is low, or read out when the $\overline{RD}$ signal is low. Odd addresses are accessed when the $\overline{BHE}$ signal is low. Use this mode when the external data bus is 8 bits wide. The microcomputer goes to a hold state when input to the $\overline{HOLD}$ pin is held low. While in the hold state, $\overline{HLDA}$ outputs a low level. ALE is used to latch the address. While the input level of the $\overline{RDY}$ pin is low, the bus of the microcomputer goes to a wait state.
P6$_0$ to P6$_7$	I/O port P6	Input/output	V_{CC1}	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as UART0 and UART1 I/O pins as selected by program.
P7$_0$ to P7$_7$	I/O port P7	Input/output	V_{CC1}	This is an 8-bit I/O port equivalent to P0 (P7$_0$ and P7$_1$ are N channel open-drain output). This port can function as input/output pins for timers A0 to A3 when so selected in a program. Furthermore, P7$_0$ to P7$_5$, P7$_1$, and P7$_2$ to P7$_5$ can also function as input/output pins for UART2, an input pin for timer B5, and output pins for the three-phase motor control timer, respectively.
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	V_{CC1}	P8$_0$ to P8$_4$, P8$_6$, and P8$_7$ are I/O ports with the same functions as P0. When so selected in a program, P8$_0$ to P8$_1$ and P8$_2$ to P8$_4$ can function as input/output pins for timer A4 or output pins for the three-phase motor control timer and $\overline{INT}$ interrupt input pins, respectively. P8$_6$ and P8$_7$, when so selected in a program, both can function as input/output pins for the subclock oscillator circuit. In that case, connect a crystal resonator between P8$_6$ (X_{COUT} pin) and P8$_7$ (X_{CIN} pin). P8$_5$ is an input-only port shared with $\overline{NMI}$. An $\overline{NMI}$ interrupt is generated when input on this pin changes state from high to low. The $\overline{NMI}$ function cannot be disabled in a program. A pull-up cannot be set for this pin.
P9$_0$ to P9$_7$	I/O port P9	Input/output	V_{CC1}	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as SI/O3 and SI/O4 I/O pins, Timer B0 to B4 input pins, D-A converter output pins, A-D converter input pins, or A-D trigger input pins as selected by program.
P10$_0$ to P10$_7$	I/O port P10	Input/output	V_{CC1}	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as A-D converter input pins as selected by program. Furthermore, P10$_4$ to P10$_7$ also function as input pins for the key input interrupt function.

Table 1.1.6 Pin Description (3) (128-pin Package) (Continued)

Pin name	Signal name	I/O type	Power supply circuit block	Function
P11$_0$ to P11$_7$	I/O port P11	Input/output	V_{CC1}	This is an 8-bit I/O port equivalent to P0.
P12$_0$ to P12$_7$	I/O port P12	Input/output	V_{CC2}	This is an 8-bit I/O port equivalent to P0.
P13$_0$ to P13$_7$	I/O port P13	Input/output	V_{CC2}	This is an 8-bit I/O port equivalent to P0.
P14$_0$, P14$_1$	I/O port P14	Input/output	V_{CC1}	This is an 2-bit I/O port equivalent to P0.

Memory

Figure 1.2.1 is a memory map of the M16C/62P group. The address space extends the 1M bytes from address 00000_{16} to $FFFFF_{16}$.

The internal ROM is allocated in a lower address direction beginning with address $FFFFF_{16}$. For example, a 64-Kbyte internal ROM is allocated to the addresses from $F0000_{16}$ to $FFFFF_{16}$.

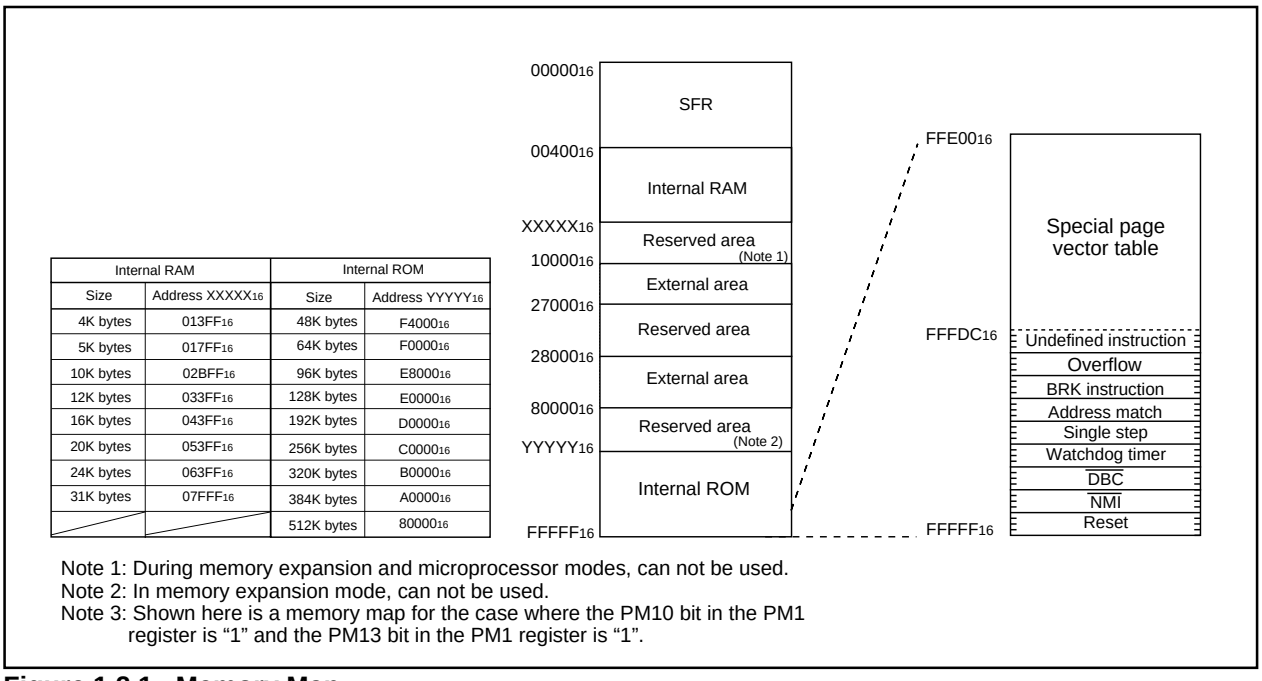
The fixed interrupt vector table is allocated to the addresses from $FFFDC_{16}$ to $FFFFF_{16}$. Therefore, store the start address of each interrupt routine here.

The internal RAM is allocated in an upper address direction beginning with address 00400_{16}. For example, a 10-Kbytes internal RAM is allocated to the addresses from 00400_{16} to $02BFF_{16}$. In addition to storing data, the internal RAM also stores the stack used when calling subroutines and when interrupts are generated.

The SRF is allocated to the addresses from 00000_{16} to $003FF_{16}$. Peripheral function control registers are located here. Of the SFR, any area which has no functions allocated is reserved for future use and cannot be used by users.

The special page vector table is allocated to the addresses from $FFE00_{16}$ to $FFFDB_{16}$. This vector is used by the JMPS or JSRS instruction. For details, refer to the “M16C/60 and M16C/20 Series Software Manual.”

In memory expansion and microprocessor modes, some areas are reserved for future use and cannot be used by users.



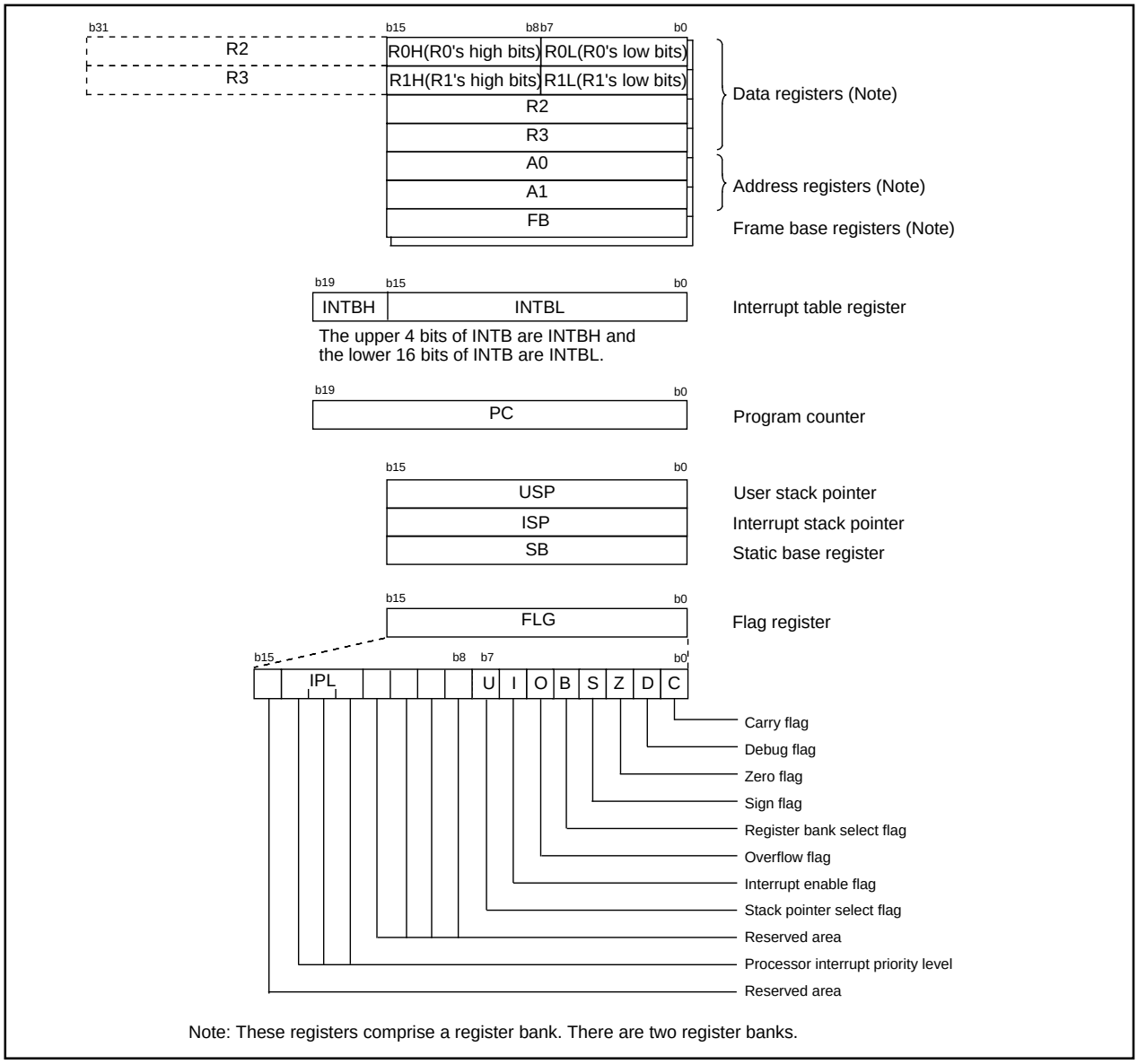
Internal RAM		Internal ROM	
Size	Address XXXXX$_{16}$	Size	Address YYYYY$_{16}$
4K bytes	013FF$_{16}$	48K bytes	F4000$_{16}$
5K bytes	017FF$_{16}$	64K bytes	F0000$_{16}$
10K bytes	02BFF$_{16}$	96K bytes	E8000$_{16}$
12K bytes	033FF$_{16}$	128K bytes	E0000$_{16}$
16K bytes	043FF$_{16}$	192K bytes	D0000$_{16}$
20K bytes	053FF$_{16}$	256K bytes	C0000$_{16}$
24K bytes	063FF$_{16}$	320K bytes	B0000$_{16}$
31K bytes	07FFF$_{16}$	384K bytes	A0000$_{16}$
		512K bytes	80000$_{16}$

Note 1: During memory expansion and microprocessor modes, can not be used.
Note 2: In memory expansion mode, can not be used.
Note 3: Shown here is a memory map for the case where the PM10 bit in the PM1 register is “1” and the PM13 bit in the PM1 register is “1”.

Figure 1.2.1. Memory Map

Central Processing Unit (CPU)

Figure 1.3.1 shows the CPU registers. The CPU has 13 registers. Of these, R0, R1, R2, R3, A0, A1 and FB comprise a register bank. There are two register banks.



Note: These registers comprise a register bank. There are two register banks.

Figure 1.3.1. Central Processing Unit Register

(1) Data Registers (R0, R1, R2 and R3)

The R0 register consists of 16 bits, and is used mainly for transfers and arithmetic/logic operations. R1 to R3 are the same as R0.

The R0 register can be separated between high (R0H) and low (R0L) for use as two 8-bit data registers. R1H and R1L are the same as R0H and R0L. Conversely, R2 and R0 can be combined for use as a 32-bit data register (R2R0). R3R1 is the same as R2R0.

(2) Address Registers (A0 and A1)

The register A0 consists of 16 bits, and is used for address register indirect addressing and address register relative addressing. They also are used for transfers and logic/logic operations. A1 is the same as A0.

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

(3) Frame Base Register (FB)

FB is configured with 16 bits, and is used for FB relative addressing.

(4) Interrupt Table Register (INTB)

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

(5) Program Counter (PC)

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

(6) User Stack Pointer (USP) and Interrupt Stack Pointer (ISP)

Stack pointer (SP) comes in two types: USP and ISP, each configured with 16 bits.
Your desired type of stack pointer (USP or ISP) can be selected by the U flag of FLG.

(7) Static Base Register (SB)

SB is configured with 16 bits, and is used for SB relative addressing.

(8) Flag Register (FLG)

FLG consists of 11 bits, indicating the CPU status.

- **Carry Flag (C Flag)**
  This flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic/logic unit.
- **Debug Flag (D Flag)**
  The D flag is used exclusively for debugging purpose. During normal use, it must be set to “0”.
- **Zero Flag (Z Flag)**
  This flag is set to “1” when an arithmetic operation resulted in 0; otherwise, it is “0”.
- **Sign Flag (S Flag)**
  This flag is set to “1” when an arithmetic operation resulted in a negative value; otherwise, it is “0”.
- **Register Bank Select Flag (B Flag)**
  Register bank 0 is selected when this flag is “0” ; register bank 1 is selected when this flag is “1”.
- **Overflow Flag (O Flag)**
  This flag is set to “1” when the operation resulted in an overflow; otherwise, it is “0”.
- **Interrupt Enable Flag (I Flag)**
  This flag enables a maskable interrupt.
  Maskable interrupts are disabled when the I flag is “0”, and are enabled when the I flag is “1”. The I flag is cleared to “0” when the interrupt request is accepted.
- **Stack Pointer Select Flag (U Flag)**
  ISP is selected when the U flag is “0”; USP is selected when the U flag is “1”.
  The U flag is cleared to “0” when a hardware interrupt request is accepted or an INT instruction for software interrupt Nos. 0 to 31 is executed.
- **Processor Interrupt Priority Level (IPL)**
  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 IPL, the interrupt is enabled.
- **Reserved Area**
  When write to this bit, write "0". When read, its content is indeterminate.

Address	Register	Symbol	After reset
0000₁₆			
0001₁₆			
0002₁₆			
0003₁₆			
0004₁₆	Processor mode register 0 (Note 2)	PM0	00000000₂(CNVss pin is "L") 00000011₂(CNVss pin is "H")
0005₁₆	Processor mode register 1	PM1	00001000₂
0006₁₆	System clock control register 0	CM0	01001000₂
0007₁₆	System clock control register 1	CM1	00100000₂
0008₁₆	Chip select control register	CSR	00000001₂
0009₁₆	Address match interrupt enable register	AIER	XXXXXX00₂
000A₁₆	Protect register	PRCR	XX000000₂
000B₁₆	Data bank register	DBR	00₁₆
000C₁₆	Oscillation stop detection register (Note 3)	CM2	0000X000₂
000D₁₆			
000E₁₆	Watchdog timer start register	WDTS	??₁₆
000F₁₆	Watchdog timer control register	WDC	00??????₂(Note 4)
0010₁₆	Address match interrupt register 0	RMAD0	00₁₆
0011₁₆			00₁₆
0012₁₆			X0₁₆
0013₁₆			
0014₁₆	Address match interrupt register 1	RMAD1	00₁₆
0015₁₆			00₁₆
0016₁₆			X0₁₆
0017₁₆			
0018₁₆			
0019₁₆	Power supply detection register 1 (Note 5)	VCR1	00001000₂
001A₁₆	Power supply detection register 2 (Note 5)	VCR2	00₁₆
001B₁₆	Chip select expansion control register	CSE	00₁₆
001C₁₆	PLL control register 0	PLC0	0001X010₂
001D₁₆			
001E₁₆	Processor mode register 2	PM2	XXX00000₂
001F₁₆	Power supply down detection interrupt register	D4INT	00₁₆
0020₁₆	DMA0 source pointer	SAR0	??₁₆
0021₁₆			??₁₆
0022₁₆			X?₁₆
0023₁₆			
0024₁₆	DMA0 destination pointer	DAR0	??₁₆
0025₁₆			??₁₆
0026₁₆			X?₁₆
0027₁₆			
0028₁₆	DMA0 transfer counter	TCR0	??₁₆
0029₁₆			??₁₆
002A₁₆			
002B₁₆			
002C₁₆	DMA0 control register	DM0CON	00000?00₂
002D₁₆			
002E₁₆			
002F₁₆			
0030₁₆	DMA1 source pointer	SAR1	??₁₆
0031₁₆			??₁₆
0032₁₆			X?₁₆
0033₁₆			
0034₁₆	DMA1 destination pointer	DAR1	??₁₆
0035₁₆			??₁₆
0036₁₆			X?₁₆
0037₁₆			
0038₁₆	DMA1 transfer counter	TCR1	??₁₆
0039₁₆			??₁₆
003A₁₆			
003B₁₆			
003C₁₆	DMA1 control register	DM1CON	00000?00₂
003D₁₆			
003E₁₆			
003F₁₆			

Note 1: The blank areas are reserved and cannot be used by users.
Note 2: The PM00 and PM01 bits do not change at software reset, watchdog timer reset and oscillation stop detection reset.
Note 3: The CM20, CM21, and CM27 bits do not change at oscillation stop detection reset.
Note 4: The WDC5 bit is "0" (cold start) immediately after power-on. It can only be set to "1" in a program. It is set to "0" when the input voltag
at the V_{CC1} pin drops to Vdet2 or less while the VC25 bit in the VCR2 register is set to "1" (RAM retention limit detection circuit enable
Note 5: This register does not change at software reset, watchdog timer reset and oscillation stop detection reset.
X : Nothing is mapped to this bit
? : Undefined

Address	Register	Symbol	After reset
0040₁₆			
0041₁₆			
0042₁₆			
0043₁₆			
0044₁₆	INT3 interrupt control register	INT3IC	XX00?000₂
0045₁₆	Timer B5 interrupt control register	TB5IC	XXXX?000₂
0046₁₆	Timer B4 interrupt control register, UART1 BUS collision detection interrupt control register	TB4IC, U1BCNIC	XXXX?000₂
0047₁₆	Timer B3 interrupt control register, UART0 BUS collision detection interrupt control register	TB3IC, U0BCNIC	XXXX?000₂
0048₁₆	SI/O4 interrupt control register (S4IC), INT5 interrupt control register	S4IC, INT5IC	XX00?000₂
0049₁₆	SI/O3 interrupt control register, INT4 interrupt control register	S3IC, INT4IC	XX00?000₂
004A₁₆	UART2 Bus collision detection interrupt control register	BCNIC	XXXX?000₂
004B₁₆	DMA0 interrupt control register	DM0IC	XXXX?000₂
004C₁₆	DMA1 interrupt control register	DM1IC	XXXX?000₂
004D₁₆	Key input interrupt control register	KUPIC	XXXX?000₂
004E₁₆	A-D conversion interrupt control register	ADIC	XXXX?000₂
004F₁₆	UART2 transmit interrupt control register	S2TIC	XXXX?000₂
0050₁₆	UART2 receive interrupt control register	S2RIC	XXXX?000₂
0051₁₆	UART0 transmit interrupt control register	S0TIC	XXXX?000₂
0052₁₆	UART0 receive interrupt control register	S0RIC	XXXX?000₂
0053₁₆	UART1 transmit interrupt control register	S1TIC	XXXX?000₂
0054₁₆	UART1 receive interrupt control register	S1RIC	XXXX?000₂
0055₁₆	Timer A0 interrupt control register	TA0IC	XXXX?000₂
0056₁₆	Timer A1 interrupt control register	TA1IC	XXXX?000₂
0057₁₆	Timer A2 interrupt control register	TA2IC	XXXX?000₂
0058₁₆	Timer A3 interrupt control register	TA3IC	XXXX?000₂
0059₁₆	Timer A4 interrupt control register	TA4IC	XXXX?000₂
005A₁₆	Timer B0 interrupt control register	TB0IC	XXXX?000₂
005B₁₆	Timer B1 interrupt control register	TB1IC	XXXX?000₂
005C₁₆	Timer B2 interrupt control register	TB2IC	XXXX?000₂
005D₁₆	INT0 interrupt control register	INT0IC	XX00?000₂
005E₁₆	INT1 interrupt control register	INT1IC	XX00?000₂
005F₁₆	INT2 interrupt control register	INT2IC	XX00?000₂
0060₁₆			
0061₁₆			
0062₁₆			
0063₁₆			
0064₁₆			
0065₁₆			
0066₁₆			
0067₁₆			
0068₁₆			
0069₁₆			
006A₁₆			
006B₁₆			
006C₁₆			
006D₁₆			
006E₁₆			
006F₁₆			
0070₁₆			
0071₁₆			
0072₁₆			
0073₁₆			
0074₁₆			
0075₁₆			
0076₁₆			
0077₁₆			
0078₁₆			
0079₁₆			
007A₁₆			
007B₁₆			
007C₁₆			
007D₁₆			
007E₁₆			
007F₁₆			

Note :The blank areas are reserved and cannot be used by users.

X : Nothing is mapped to this bit
? : Undefined

Address	Register	Symbol	After reset
0080₁₆			
0081₁₆			
0082₁₆			
0083₁₆			
0084₁₆			
0085₁₆			
0086₁₆			
≈			≈
01B0₁₆			
01B1₁₆			
01B2₁₆			
01B3₁₆			
01B4₁₆	Flash identification register (Note 2)	FIDR	XXXXXX00₂
01B5₁₆	Flash memory control register 1 (Note 2)	FMR1	0?00??0?₂
01B6₁₆			
01B7₁₆	Flash memory control register 0 (Note 2)	FMR0	??0000012
01B8₁₆ 01B9₁₆ 01BA₁₆	Address match interrupt register 2	RMAD2	00₁₆ 00₁₆ X0₁₆
01BB₁₆	Address match interrupt enable register 2	AIER2	XXXXXX00₂
01BC₁₆ 01BD₁₆ 01BE₁₆	Address match interrupt register 3	RMAD3	00₁₆ 00₁₆ X0₁₆
01BF₁₆			
≈			≈
0250₁₆			
0251₁₆			
0252₁₆			
0253₁₆			
0254₁₆			
0255₁₆			
0256₁₆			
0257₁₆			
0258₁₆			
0259₁₆			
025A₁₆			
025B₁₆			
025C₁₆			
025D₁₆			
025E₁₆	Peripheral clock select register	PCLKR	00000011₂
025F₁₆			
≈			≈
0330₁₆			
0331₁₆			
0332₁₆			
0333₁₆			
0334₁₆			
0335₁₆			
0336₁₆			
0337₁₆			
0338₁₆			
0339₁₆			
033A₁₆			
033B₁₆			
033C₁₆			
033D₁₆			
033E₁₆			
033F₁₆			

Note 1: The blank areas are reserved and cannot be used by users.
Note 2: This register is included in the flash memory version.

X : Nothing is mapped to this bit
? : Undefined

Address	Register	Symbol	After reset
034016	Timer B3, 4, 5 count start flag	TBSR	000XXXXX2
034116			
034216	Timer A1-1 register	TA11	??16
034316			??16
034416	Timer A2-1 register	TA21	??16
034516			??16
034616	Timer A4-1 register	TA41	??16
034716			??16
034816	Three-phase PWM control register 0	INVC0	0016
034916	Three-phase PWM control register 1	INVC1	0016
034A16	Three-phase output buffer register 0	IDB0	0016
034B16	Three-phase output buffer register 1	IDB1	0016
034C16	Dead time timer	DTT	??16
034D16	Timer B2 interrupt occurrence frequency set counter	ICTB2	??16
034E16			
034F16			
035016	Timer B3 register	TB3	??16
035116			??16
035216	Timer B4 register	TB4	??16
035316			??16
035416	Timer B5 register	TB5	??16
035516			??16
035616			
035716			
035816			
035916			
035A16			
035B16	Timer B3 mode register	TB3MR	00??00002
035C16	Timer B4 mode register	TB4MR	00?X00002
035D16	Timer B5 mode register	TB5MR	00?X00002
035E16	Interrupt cause select register 2	IFSR2A	00XXXXXX2
035F16	Interrupt cause select register	IFSR	0016
036016	SI/O3 transmit/receive register	S3TRR	??16
036116			
036216	SI/O3 control register	S3C	010000002
036316	SI/O3 bit rate generator	S3BRG	??16
036416	SI/O4 transmit/receive register	S4TRR	??16
036516			
036616	SI/O4 control register	S4C	010000002
036716	SI/O4 bit rate generator	S4BRG	??16
036816			
036916			
036A16			
036B16			
036C16	UART0 special mode register 4	U0SMR4	0016
036D16	UART0 special mode register 3	U0SMR3	000X0X0X2
036E16	UART0 special mode register 2	U0SMR2	X0000000 2
036F16	UART0 special mode register	U0SMR	X0000000 2
037016	UART1 special mode register 4	U1SMR4	0016
037116	UART1 special mode register 3	U1SMR3	000X0X0X2
037216	UART1 special mode register 2	U1SMR2	X0000000 2
037316	UART1 special mode register	U1SMR	X0000000 2
037416	UART2 special mode register 4	U2SMR4	0016
037516	UART2 special mode register 3	U2SMR3	000X0X0X2
037616	UART2 special mode register 2	U2SMR2	X0000000 2
037716	UART2 special mode register	U2SMR	X0000000 2
037816	UART2 transmit/receive mode register	U2MR	0016
037916	UART2 bit rate generator	U2BRG	??16
037A16	UART2 transmit buffer register	U2TB	????????2
037B16			XXXXXXX?2
037C16	UART2 transmit/receive control register 0	U2C0	000010002
037D16	UART2 transmit/receive control register 1	U2C1	000000102
037E16	UART2 receive buffer register	U2RB	????????2
037F16			?????XX?2

Note : The blank areas are reserved and cannot be used by users.
X : Nothing is mapped to this bit
? : Undefined

Address	Register	Symbol	After reset
038016	Count start flag	TABSR	0016
038116	Clock prescaler reset flag	CPSRF	0XXXXXXX2
038216	One-shot start flag	ONSF	0016
038316	Trigger select register	TRGSR	0016
038416	Up-down flag	UDF	0016
038516			
038616	Timer A0 register	TA0	??16
038716			??16
038816	Timer A1 register	TA1	??16
038916			??16
038A16	Timer A2 register	TA2	??16
038B16			??16
038C16	Timer A3 register	TA3	??16
038D16			??16
038E16	Timer A4 register	TA4	??16
038F16			??16
039016	Timer B0 register	TB0	??16
039116			??16
039216	Timer B1 register	TB1	??16
039316			??16
039416	Timer B2 register	TB2	??16
039516			??16
039616	Timer A0 mode register	TA0MR	0016
039716	Timer A1 mode register	TA1MR	0016
039816	Timer A2 mode register	TA2MR	0016
039916	Timer A3 mode register	TA3MR	0016
039A16	Timer A4 mode register	TA4MR	0016
039B16	Timer B0 mode register	TB0MR	00??00002
039C16	Timer B1 mode register	TB1MR	00?X00002
039D16	Timer B2 mode register	TB2MR	00?X00002
039E16	Timer B2 special mode register	TB2SC	XXXXXX002
039F16			
03A016	UART0 transmit/receive mode register	U0MR	0016
03A116	UART0 bit rate generator	U0BRG	??16
03A216	UART0 transmit buffer register	U0TB	????????2
03A316			XXXXXXX?2
03A416	UART0 transmit/receive control register 0	U0C0	000010002
03A516	UART0 transmit/receive control register 1	U0C1	000000102
03A616	UART0 receive buffer register	U0RB	????????2
03A716			?????XX?2
03A816	UART1 transmit/receive mode register	U1MR	0016
03A916	UART1 bit rate generator	U1BRG	??16
03AA16	UART1 transmit buffer register	U1TB	????????2
03AB16			XXXXXXX?2
03AC16	UART1 transmit/receive control register 0	U1C0	000010002
03AD16	UART1 transmit/receive control register 1	U1C1	000000102
03AE16	UART1 receive buffer register	U1RB	????????2
03AF16			?????XX?2
03B016	UART transmit/receive control register 2	UCON	X00000002
03B116			
03B216			
03B316			
03B416			
03B516			
03B616			
03B716			
03B816	DMA0 request cause select register	DM0SL	0016
03B916			
03BA16	DMA1 request cause select register	DM1SL	0016
03BB16			
03BC16	CRC data register	CRCD	??16
03BD16			??16
03BE16	CRC input register	CRCIN	??16
03BF16			

Note : The blank areas are reserved and cannot be used by users.
X : Nothing is mapped to this bit
? : Undefined

Address	Register	Symbol	After reset
03C0₁₆ 03C1₁₆	A-D register 0	AD0	????????₂ XXXXXX??₂
03C2₁₆ 03C3₁₆	A-D register 1	AD1	????????₂ XXXXXX??₂
03C4₁₆ 03C5₁₆	A-D register 2	AD2	????????₂ XXXXXX??₂
03C6₁₆ 03C7₁₆	A-D register 3	AD3	????????₂ XXXXXX??₂
03C8₁₆ 03C9₁₆	A-D register 4	AD4	????????₂ XXXXXX??₂
03CA₁₆ 03CB₁₆	A-D register 5	AD5	????????₂ XXXXXX??₂
03CC₁₆ 03CD₁₆	A-D register 6	AD6	????????₂ XXXXXX??₂
03CE₁₆ 03CF₁₆	A-D register 7	AD7	????????₂ XXXXXX??₂
03D0₁₆			
03D1₁₆			
03D2₁₆			
03D3₁₆			
03D4₁₆	A-D control register 2	ADCON2	00₁₆
03D5₁₆			
03D6₁₆	A-D control register 0	ADCON0	00000???₂
03D7₁₆	A-D control register 1	ADCON1	00₁₆
03D8₁₆	D-A register 0	DA0	??₁₆
03D9₁₆			
03DA₁₆	D-A register 1	DA1	??₁₆
03DB₁₆			
03DC₁₆	D-A control register	DACON	00₁₆
03DD₁₆			
03DE₁₆	Port P14 control register	PC14	XX00XXXX₂
03DF₁₆	Pull-up control register 3	PUR3	00₁₆
03E0₁₆	Port P0 register	P0	??₁₆
03E1₁₆	Port P1 register	P1	??₁₆
03E2₁₆	Port P0 direction register	PD0	00₁₆
03E3₁₆	Port P1 direction register	PD1	00₁₆
03E4₁₆	Port P2 register	P2	??₁₆
03E5₁₆	Port P3 register	P3	??₁₆
03E6₁₆	Port P2 direction register	PD2	00₁₆
03E7₁₆	Port P3 direction register	PD3	00₁₆
03E8₁₆	Port P4 register	P4	??₁₆
03E9₁₆	Port P5 register	P5	??₁₆
03EA₁₆	Port P4 direction register	PD4	00₁₆
03EB₁₆	Port P5 direction register	PD5	00₁₆
03EC₁₆	Port P6 register	P6	??₁₆
03ED₁₆	Port P7 register	P7	??₁₆
03EE₁₆	Port P6 direction register	PD6	00₁₆
03EF₁₆	Port P7 direction register	PD7	00₁₆
03F0₁₆	Port P8 register	P8	??₁₆
03F1₁₆	Port P9 register	P9	??₁₆
03F2₁₆	Port P8 direction register	PD8	00X000002
03F3₁₆	Port P9 direction register	PD9	00₁₆
03F4₁₆	Port P10 register	P10	??₁₆
03F5₁₆	Port P11 register	P11	??₁₆
03F6₁₆	Port P10 direction register	PD10	00₁₆
03F7₁₆	Port P11 direction register	PD11	00₁₆
03F8₁₆	Port P12 register	P12	??₁₆
03F9₁₆	Port P13 register	P13	??₁₆
03FA₁₆	Port P12 direction register	PD12	00₁₆
03FB₁₆	Port P13 direction register	PD13	00₁₆
03FC₁₆	Pull-up control register 0	PUR0	00₁₆
03FD₁₆	Pull-up control register 1	PUR1	00000000₂ (Note 2) 00000010₂
03FE₁₆	Pull-up control register 2	PUR2	00₁₆
03FF₁₆	Port control register	PCR	00₁₆

Note 1: The blank areas are reserved and cannot be used by users.
Note 2: At hardware reset 1 or hardware reset 2, the register is as follows:
- "00000000₂" where "L" is inputted to the CNVSS pin
- "00000010₂" where "H" is inputted to the CNVSS pin

At software reset, watchdog timer reset and oscillation stop detection reset, the register is as follows:
- "00000000₂" where the PM01 to PM00 bits in the PM0 register are "00₂" (single-chip mode)
- "00000010₂" where the PM01 to PM00 bits in the PM0 register are "01₂" (memory expansion mode) or "11₂" (microprocessor mode)

X : Nothing is mapped to this bit
? : Undefined

Reset

There are four types of resets: a hardware reset, a software reset, an watchdog timer reset, and an oscillation stop detection reset.

Hardware Reset

There are two types of hardware resets: a hardware reset 1 and a hardware reset 2.

Hardware Reset 1

A reset is applied using the $\overline{\text{RESET}}$ pin. When an “L” signal is applied to the $\overline{\text{RESET}}$ pin while the power supply voltage is within the recommended operating condition, the pins are initialized (see Table 1.5.1). The oscillation circuit is initialized and the main clock starts oscillating. When the input level at the $\overline{\text{RESET}}$ pin is released from “L” to “H”, the CPU and SFR are initialized, and the program is executed starting from the address indicated by the reset vector. The internal RAM is not initialized. If the $\overline{\text{RESET}}$ pin is pulled “L” while writing to the internal RAM, the internal RAM becomes indeterminate.

Figure 1.5.1 shows the example reset circuit. Figure 1.5.2 shows the reset sequence. Table 1.5.1 shows the statuses of the other pins while the $\overline{\text{RESET}}$ pin is “L”. Figure 1.5.3 shows the CPU register status after reset. Refer to “SFR” for SFR status after reset.

1. When the power supply is stable
(1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin.
(2) Supply a clock for 20 cycles or more to the X_{IN} pin.
(3) Apply an “H” signal to the $\overline{\text{RESET}}$ pin.

2. Power on
(1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin.
(2) Let the power supply voltage increase until it meets the recommended operating condition.
(3) Wait td(P-R) or more until the internal power supply stabilizes.
(4) Supply a clock for 20 cycles or more to the X_{IN} pin.
(5) Apply an “H” signal to the $\overline{\text{RESET}}$ pin.

Hardware Reset 2

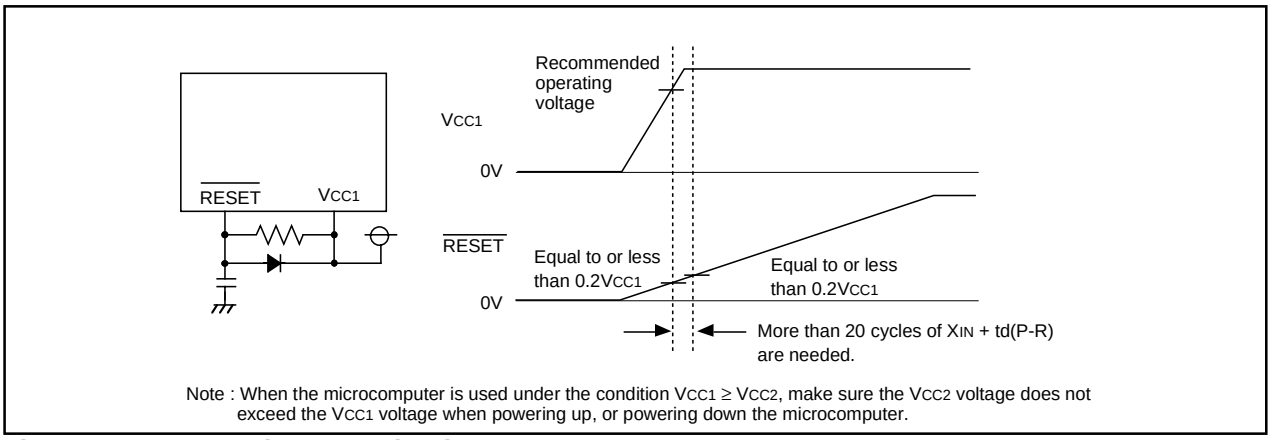
This reset is generated by the microcomputer’s internal voltage detection circuit. The voltage detection circuit monitors the voltage supplied to the V_{CC1} pin.

If the VC26 bit in the VCR2 register is set to “1” (reset level detection circuit enabled), the microcomputer is reset when the voltage at the V_{CC1} input pin drops below Vdet3.

Similarly, if the VC25 bit in the VCR2 register is set to “1” (RAM retention limit detection circuit enabled), the microcomputer is reset when the voltage at the VCC1 input pin drops below Vdet2.

Conversely, when the input voltage at the V_{CC1} pin rises to Vdet3 or more, the pins and the CPU and SFR are initialized, and the program is executed starting from the address indicated by the reset vector. It takes about td(S-R) before the program starts running after Vdet3 is detected. The initialized pins and registers and the status thereof are the same as in hardware reset 1.

Set the CM10 bit in the CM1 register to “1” (stop mode) after setting the VC25 bit to “1” (RAM retention limit detection circuit enabled), and the microcomputer will be reset when the voltage at the VCC1 input pin drops below Vdet2 and comes out of reset when the voltage at the V_{CC1} input pin rises above Vdet3. During stop mode, the value set in the VC26 bit has no effect. Therefore, no reset is generated even when the input voltage at the V_{CC1} pin drops to Vdet3 or less.



Note : When the microcomputer is used under the condition $V_{CC1} \geq V_{CC2}$, make sure the V_{CC2} voltage does not exceed the V_{CC1} voltage when powering up, or powering down the microcomputer.

Figure 1.5.1. Example Reset Circuit

Software Reset

When the PM03 bit in the PM0 register is set to “1” (microcomputer reset), the microcomputer has its pins, CPU, and SFR initialized. Then the program is executed starting from the address indicated by the reset vector.

Select the main clock for the CPU clock source, and set the PM03 bit to “1” with main clock oscillation satisfactorily stable.

At software reset, some SFR’s are not initialized. Refer to “SFR”. Also, since the PM01 to PM00 bits in the PM0 register are not initialized, the processor mode remains unchanged.

Watchdog Timer Reset

Where the PM12 bit in the PM1 register is “1” (reset when watchdog timer underflows), the microcomputer initializes its pins, CPU and SFR if the watchdog timer underflows. Then the program is executed starting from the address indicated by the reset vector.

At watchdog timer reset, some SFR’s are not initialized. Refer to “SFR”. Also, since the PM01 to PM00 bits in the PM0 register are not initialized, the processor mode remains unchanged.

Oscillation Stop Detection Reset

Where the CM27 bit in the CM2 register is “0” (reset at oscillation stop detection), the microcomputer initializes its pins, CPU and SFR, coming to a halt if it detects main clock oscillation circuit stop. Refer to the section “oscillation stop, re-oscillation detection function”.

At oscillation stop detection reset, some SFR’s are not initialized. Refer to the section “SFR”. Also, since the PM01 to PM00 bits in the PM0 register are not initialized, the processor mode remains unchanged.

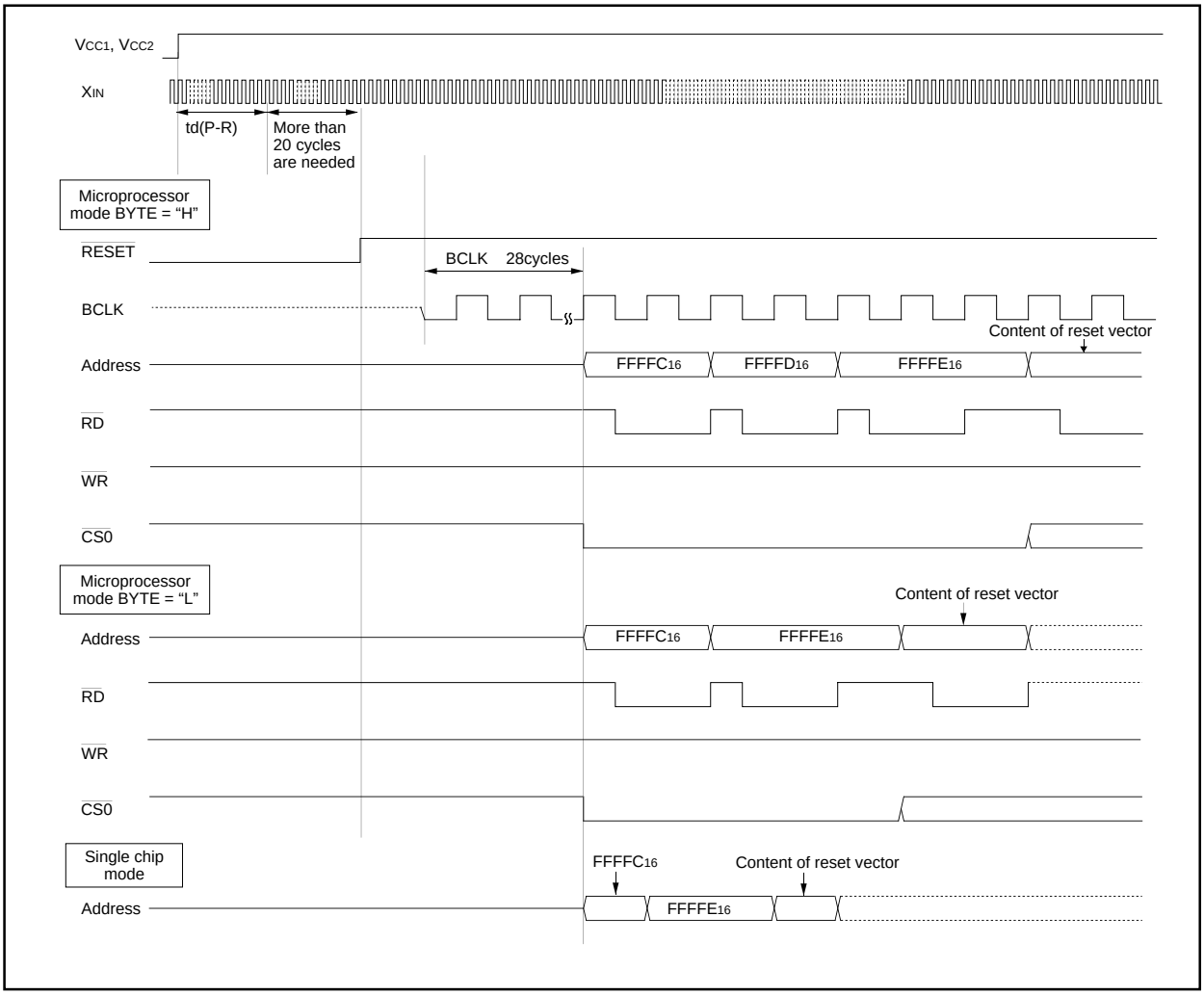


Figure 1.5.2. Reset Sequence

Table 1.5.1. Pin Status When $\overline{\text{RESET}}$ Pin Level is "L"

Pin name	Status		
	CNVSS = VSS	CNVSS = VCC1	
		BYTE = VSS	BYTE = VCC
P0	Input port	Data input	Data input
P1	Input port	Data input	Input port
P2, P3, P40 to P43	Input port	Address output (undefined)	Address output (undefined)
P44	Input port	$\overline{\text{CS0}}$ output ("H" is output)	$\overline{\text{CS0}}$ output ("H" is output)
P45 to P47	Input port	Input port (Pulled high)	Input port (Pulled high)
P50	Input port	$\overline{\text{WR}}$ output ("H" is output)	$\overline{\text{WR}}$ output ("H" is output)
P51	Input port	$\overline{\text{BHE}}$ output (undefined)	$\overline{\text{BHE}}$ output (undefined)
P52	Input port	$\overline{\text{RD}}$ output ("H" is output)	$\overline{\text{RD}}$ output ("H" is output)
P53	Input port	BCLK output	BCLK output
P54	Input port	$\overline{\text{HLDA}}$ 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	$\overline{\text{HOLD}}$ input	$\overline{\text{HOLD}}$ input
P56	Input port	ALE output ("L" is output)	ALE output ("L" is output)
P57	Input port	$\overline{\text{RDY}}$ input	$\overline{\text{RDY}}$ input
P6, P7, P80 to P84, P86, P87, P9, P10	Input port	Input port	Input port
P11, P12, P13, P140, P141 (Note)	Input port	Input port	Input port

Note : P11, P12, P13, P140, P141 pins exist in 128-pin version.

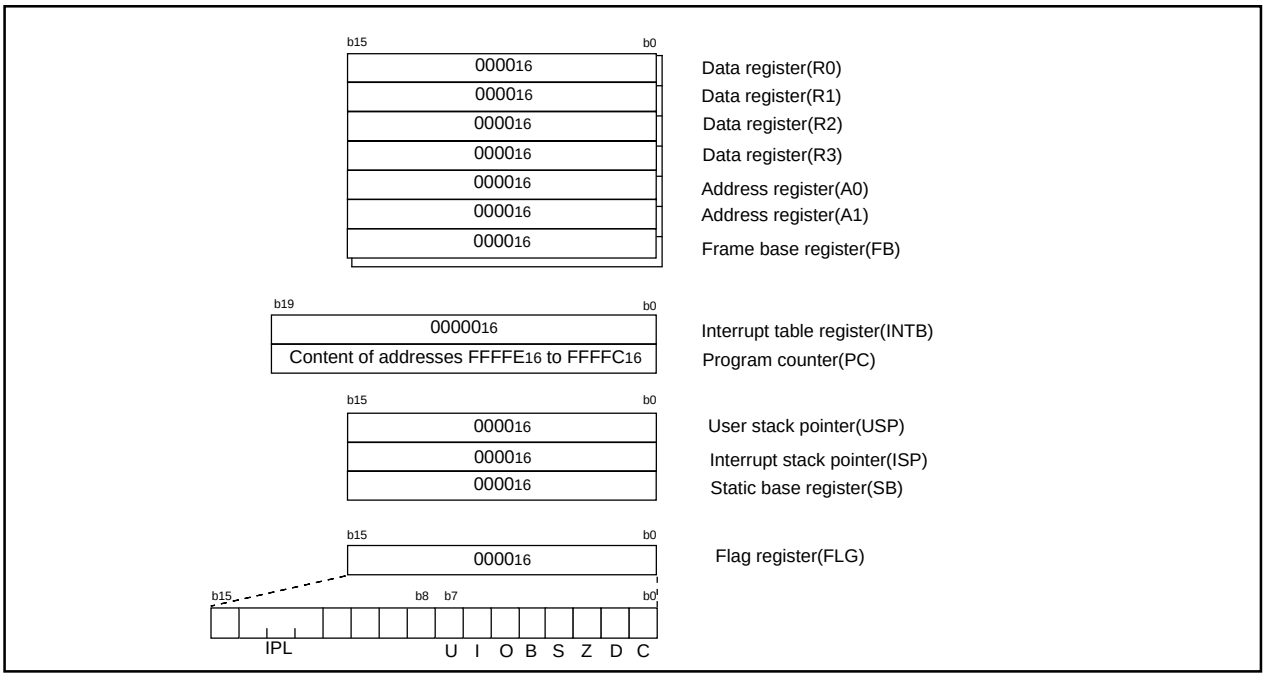


Figure 1.5.3. CPU Register Status After Rreset

Voltage Detection Circuit

The voltage detection circuit has circuits to monitor the input voltage at the V_{CC1} pin, each checking the input voltage with respect to Vdet2, Vdet3, and Vdet4, respectively. Use the VC25 to VC27 bits in the VCR2 register to select whether or not to enable these circuits.

Enable the RAM retention limit detection circuit when using hardware reset 2 in stop mode, or when using the WDC5 bit in the WDC register. The WDC5 bit indicates that the RAM is retained.

Use the reset level detection circuit for hardware reset 2.

The power supply down detection circuit can be set to detect whether the input voltage is equal to or greater than Vdet4 or less than Vdet4 by using the VC13 bit in the VCR1 register. Furthermore, a power supply down detection interrupt can be used.

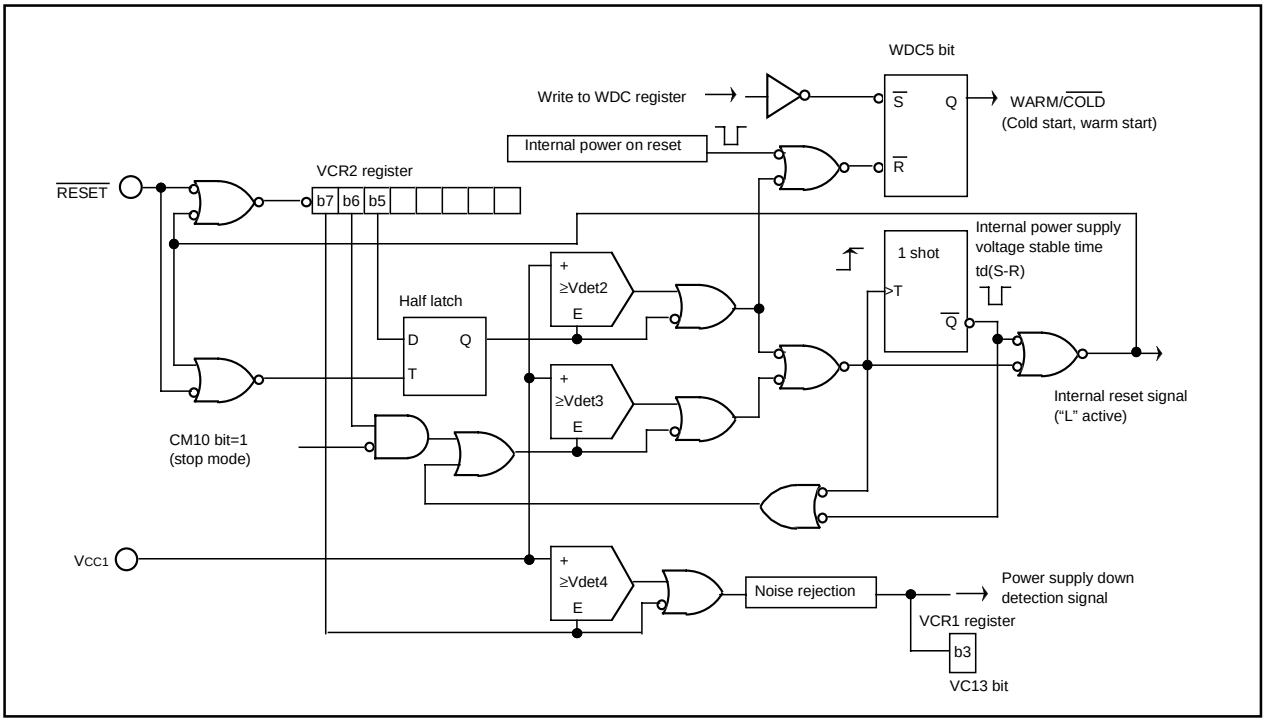


Figure 1.5.4. Reset Circuit Block

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0
	0						

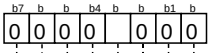
Symbol: WDC
Address: 000F16
After reset: 00XXXXXX2(Note2)

Bit symbol	Bit name	Function	RW
$\overline{(b4\text{-}b0)}$	High-order bit of watchdog timer		RO
WDC5	Cold start / warm start discrimination flag (Note 1)	0 : Cold start 1 : Warm start	RW
$\overline{(b6)}$	Reserved bit	Set to "0"	RW
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	RW

Note 1: The WDC5 bit is always "1" (= warm start) no matter how it is set by writing a "0" or "1".
Note 2: The WDC5 bit is "0" (= cold start) immediately after power-on. It can only be set to "1" in a program. It is set to "0" when the input voltage at the V_{CC1} pin drops to V_{det2} or less while the VCR2 register's VC25 bit = 1 (RAM retention limit detection circuit enabled).

Figure 1.5.5. WDC Register

Power supply detection register 1

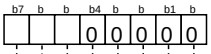


Symbol VCR1 — Address 0019₁₆ — After reset (Note 2) 0000 1000₂

Bit symbol	Bit name	Function	RW
— (b2-b0)	Reserved bit	Must set to “0”	RW
VC13	Power supply down monitor flag (Note 1)	0: V_{CC1} < Vdet4 1: V_{CC1} Vdet4	RO
— (b7-b4)	Reserved bit	Must set to “0”	RW

Note 1: The VC13 bit is useful when the VC27 bit of VCR2 register is set to “1” (power supply down detection circuit enable). The VC13 bit is always “1” (V_{CC1} 4 V) when the VC27 bit in the VCR2 register is set to “0” (power supply down detection circuit disable).
Note 2: This register does not change at software reset, watchdog timer reset and oscillation stop detection reset.

Power supply detection register 2 (Note 1)

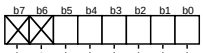


Symbol VCR2 — Address 001A₁₆ — After reset (Note 6) 00₁₆

Bit symbol	Bit name	Function	RW
— (b4-b0)	Reserved bit	Must set to “0”	RW
VC25	RAM retention limit detection monitor bit (Notes 3, 4, 7)	0: Disable RAM retention limit detection circuit 1: Enable RAM retention limit detection circuit	RW
VC26	Reset level monitor bit (Notes 2, 3, 7)	0: Disable reset level detection circuit 1: Enable reset level detection circuit	RW
VC27	Power supply down monitor bit (Note 5)	0: Disable power supply down detection circuit 1: Enable power supply down detection circuit	RW

Note 1: Write to this register after setting the PRC3 bit in the PRCR register to “1” (write enable).
Note 2: To use hardware reset 2, set the VC26 bit to “1” (reset level detection circuit enable).
Note 3: To use hardware reset 2 in stop mode, set the VC25 bit to “1” (RAM retention limit detection circuit enable). VC26 bit is disabled in stop mode. (The microcomputer is not reset even if the voltage input to V_{CC1} pin becomes lower than Vdet3.)
Note 4: To use the WDC5 bit in the WDC register, set the VC25 bit to “1” (RAM retention limit detection circuit enable).
Note 5: Where the VC13 bit in the VCR1 register and D42 bit in the D4INT register are used or the D40 bit is set to “1” (power supply down detection interrupt enable), set the VC27 bit to “1” (power supply down detection circuit enable).
Note 6: This register does not change at software reset, watchdog timer reset and oscillation stop detection reset.
Note 7: The detection circuit does not start operation until td(E-A) elapses after the VC25 bit, VC26 bit, or VC27 bit is set to “1”.

Power supply down detection interrupt register (Note 1)



Symbol D4INT — Address 001F₁₆ — After reset 00₁₆

Bit symbol	Bit name	Function	RW
D40	Power supply down detection interrupt enable bit (Note 5)	0 : Disable 1 : Enable	RW
D41	STOP mode deactivation control bit (Note 4)	0: Disable (do not use the power supply down detection interrupt to get out of stop mode) 1: Enable (use the power supply down detection interrupt to get out of stop mode)	RW
D42	Power supply change detection flag (Note 2)	0: Not detected 1: Vdet4 passing detection	RW (Note 3)
D43	WDT overflow detect flag	0: Not detected 1: Detected	RW (Note 3)
DF0	Sampling clock select bit	b5b4 00 : CPU clock divided by 8 01 : CPU clock divided by 16 10 : CPU clock divided by 32 11 : CPU clock divided by 64	RW
DF1			RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

Note 1: Write to this register after setting the PRC3 bit in the PRCR register to “1” (write enable).
Note 2: Useful when the VC27 bit in the VCR2 register is set to “1” (power supply down detection circuit enabled). If the VC27 bit is set to “0” (power supply down detection circuit disable), the D42 bit is set to “0” (Not detect).
Note 3: This bit is set to “0” by writing a “0” in a program. (Writing a “1” has no effect.)
Note 4: If the power supply down detection interrupt needs to be used to get out of stop mode again after once used for that purpose, reset the D41 bit by writing a “0” and then a “1”.
Note 5: The D40 bit is useful where the VC27 bit in the VCR2 register is set to “1”.

Figure 1.5.6. VCR1 Register, VCR2 Register, and D4INT Register

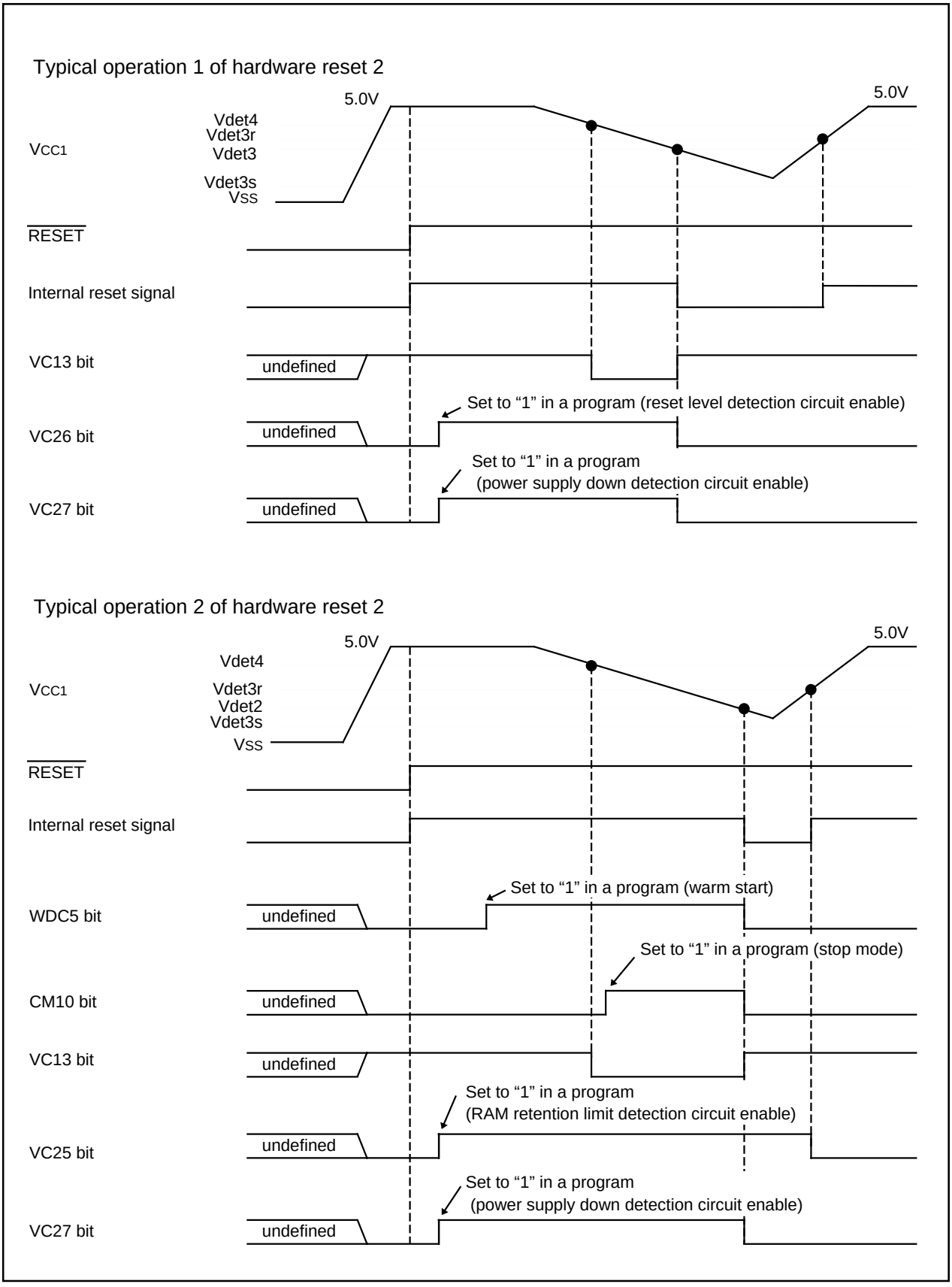


Figure 1.5.7. Typical Operation of Hardware Reset 2

Power Supply Down Detection Interrupt

A power supply down detection interrupt request is generated when the input voltage at the V_{CC1} pin rises to Vdet4 or more or drops below Vdet4 while the D40 bit in the D4INT register is set to "1" (power supply down detection interrupt enable). The power supply down detection interrupt shares the interrupt vector with the watchdog timer interrupt and oscillation stop, re-oscillation detection interrupt.

To use the power supply down detection interrupt to get out of stop mode, set the D41 bit in the D4INT register to "1" (enable).

The D42 bit in the D4INT register becomes "1" when passing through Vdet4 is detected after the voltage inputted to the V_{CC1} pin is up or down.

A power supply down detection interrupt is generated when the D42 bit changes state from "0" to "1". The D42 bit needs to be set to "0" in a program. However, where the D41 bit is "1" and the stop mode is selected, the power supply down detection interrupt request arises, and the microcomputer is reset from the stop mode with no regard for the status of D42 bit if it is detected that the voltage applied to the V_{CC1} pin has increased, passing through Vdet4.

Table 1.5.2 shows the power supply down detection interrupt request generation conditions.

It is possible to set the sampling clock detecting that the voltage applied to the V_{CC1} pin has passed through Vdet4 with the DF1 to DF0 bits of D4INT register. Table 1.5.3 shows sampling clock periods.

Table 1.5.2. Power Supply Down Detection Interrupt Request Generation Conditions

Bit, Vdet4 passing detection, operation mode condition							Power supply down detection Interrupt request
VC27 bit	D40 bit	Vdet4 passing detection	D42 bit	D41 bit	VC13 bit	Operation mode (Notes 1, 2)	
0	—	—	—	—	—	—	Not generated
1	0	—	—	—	—	—	
	1	Not detected	—	—	—	—	
		Detected	From 0 to 1	0	—	Normal, wait	Generated
						Stop	Not generated
				1	—	—	Generated
			From 1 to 1 (No change)	0	—	—	Not generated
				1	From 0 to 1 (Up)	Normal, wait	
						Stop	Generated
					From 1 to 0 (Down)	—	Not generated

Note 1: The status except the wait mode and stop mode is handled as the normal mode.(Refer to "Clock generating circuit")
Note 2: Refer to "Limitations on stop mode", "Limitations on wait mode".

Table 1.5.3. Sampling Clock Periods

CPU clock (MHz)	Sampling clock (µs)			
	divided by 8	divided by 16	divided by 32	divided by 64
16	1.5	3.0	6.0	12.0

Precautions

1. Limitations on Stop Mode

If the CM10 bit in the CM1 register is set to "1" (stop mode) when the VC13 bit in the VCR1 register is "1" ($V_{CC1} \geq$ Vdet4) while the VC27 bit in the VCR2 register is "1" (power supply down detection circuit enable) and the D40 bit in the D4INT register is "1" (power supply down detection interrupt enable) and D41 bit in the D4INT register is "1" (power supply down detection interrupt is used to get out of stop mode), a power supply down detection interrupt is immediately generated, causing the microcomputer to exit stop mode.

In systems where the microcomputer enters stop mode when the input voltage at the V_{CC1} pin drops below Vdet4 and exits stop mode when the input voltage rises to Vdet4 or more, make sure the CM10 bit is set to "1" when VC13 bit is "0" (V_{CC1} < Vdet4).

2. Limitations on WAIT Instruction

If the WAIT instruction is executed when the VC13 bit in the VCR1 register is “1” ($V_{CC1} \geq$ Vdet4) while the VC27 bit in the VCR2 register is “1” (power supply down detection circuit enable) and the D40 bit in the D4INT register is “1” (power supply down detection interrupt enable), a power supply down detection interrupt is immediately generated, causing the microcomputer to exit wait mode.

In systems where the microcomputer enters wait mode when the input voltage at the V_{CC1} pin drops below Vdet4 and exits wait mode when the input voltage rises to Vdet4 or more, make sure the WAIT instruction is executed when VC13 bit is “0” (V_{CC1} < Vdet4).

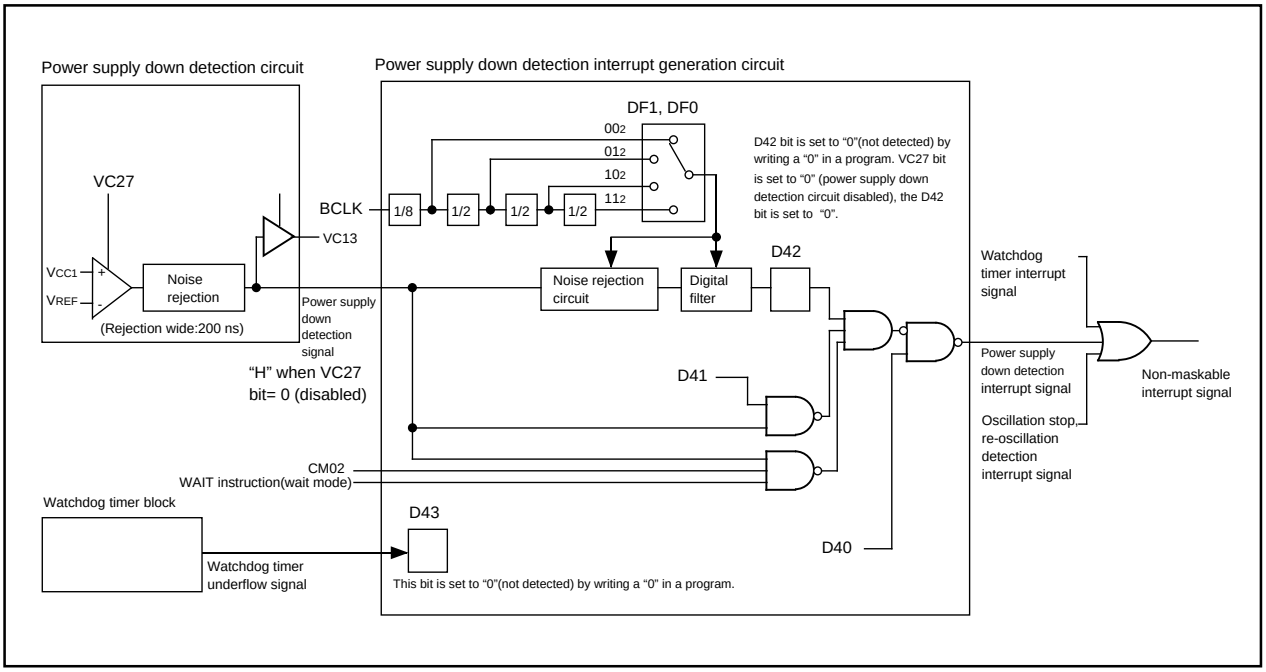
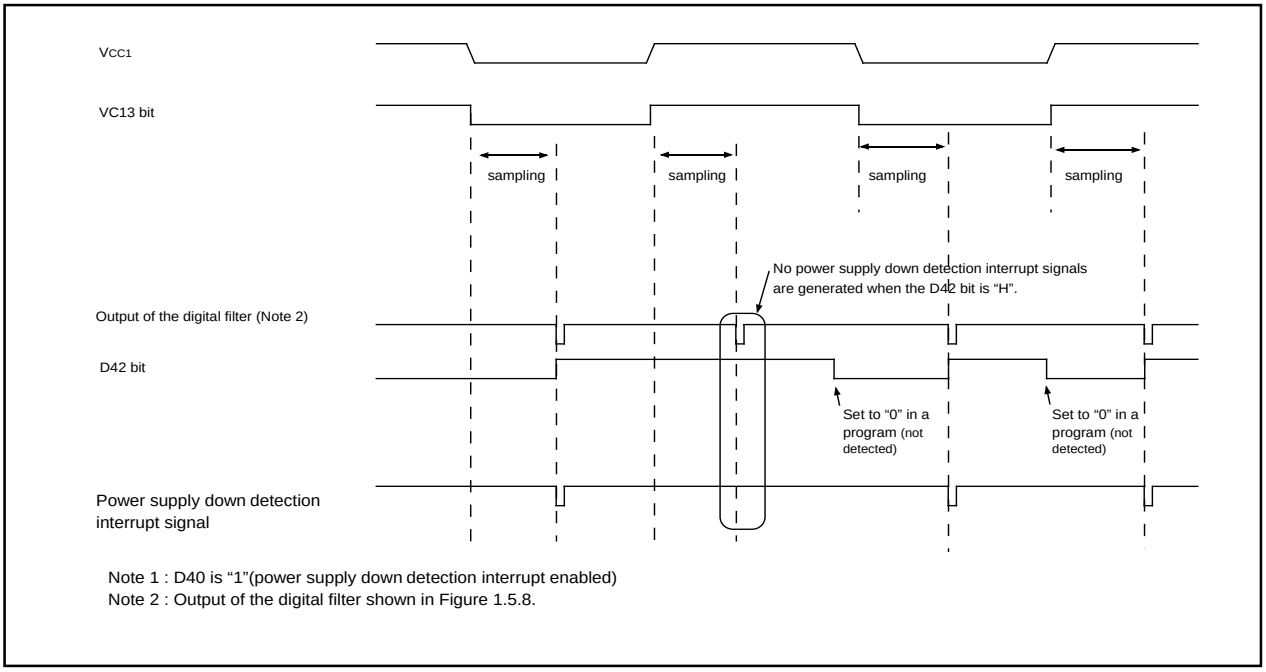


Figure 1.5.8. Power Supply Down Detection Interrupt Generation Block



Note 1 : D40 is “1”(power supply down detection interrupt enabled)
Note 2 : Output of the digital filter shown in Figure 1.5.8.

Figure 1.5.9. Power Supply Down Detection Interrupt Generation Circuit Operation Example

Processor Mode

(1) Types of Processor Mode

Three processor modes are available to choose from: single-chip mode, memory expansion mode, and microprocessor mode. Table 1.6.1 shows the features of these processor modes.

Table 1.6.1. Features of Processor Modes

Processor modes	Access space	Pins which are assigned I/O ports
Single-chip mode	SFR, internal RAM, internal ROM	All pins are I/O ports or peripheral function I/O pins
Memory expansion mode	SFR, internal RAM, internal ROM, external area (Note)	Some pins serve as bus control pins (Note)
Microprocessor mode	SFR, internal RAM, external area (Note)	Some pins serve as bus control pins (Note)

Note : Refer to “Bus”.

(2) Setting Processor Modes

Processor mode is set by using the CNVSS pin and the PM01 to PM00 bits in the PM0 register.
Table 1.6.2 shows the processor mode after hardware reset. Table 1.6.3 shows the PM01 to PM00 bit set values and processor modes.

Table 1.6.2. Processor Mode After Hardware Reset

CNVSS pin input level	Processor mode
VSS	Single-chip mode
VCC1 (Note 1, Note 2)	Microprocessor mode

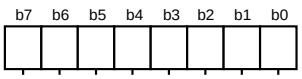
Note 1: If the microcomputer is reset in hardware by applying VCC1 to the CNVSS pin (hardware reset 1 or hardware reset 2), the internal ROM cannot be accessed regardless of PM10 to PM00 bits.
Note 2: The multiplexed bus cannot be assigned to the entire CS space.

Table 1.6.3. PM01 to PM00 Bits Set Values and Processor Modes

PM01 to PM00 bits	Processor modes
00_2	Single-chip mode
01_2	Memory expansion mode
10_2	Must not be set
11_2	Microprocessor mode

Rewriting the PM01 to PM00 bits places the microcomputer in the corresponding processor mode regardless of whether the input level on the CNVSS pin is “H” or “L”. Note, however, that the PM01 to PM00 bits cannot be rewritten to “01_2” (memory expansion mode) or “11_2” (microprocessor mode) at the same time the PM07 to PM02 bits are rewritten. Note also that these bits cannot be rewritten to enter microprocessor mode in the internal ROM, nor can they be rewritten to exit microprocessor mode in areas overlapping the internal ROM.
If the microcomputer is reset in hardware by applying VCC1 to the CNVSS pin (hardware reset 1 or hardware reset 2), the internal ROM cannot be accessed regardless of PM01 to PM00 bits.
Figures 1.6.1 and 1.6.2 show the registers associated with processor modes. Figure 1.6.3 show the memory map in single chip mode.

Processor mode register 0 (Note 1)



Symbol	Address	After reset (Note 4)
PM0	0004_{16}	00000000_2 (CNVSS pin = "L") 00000011_2 (CNVSS pin = "H")

Bit symbol	Bit name	Function	RW
PM00	Processor mode bit (Note 4)	b1 b0 0 0: Single-chip mode 0 1: Memory expansion mode 1 0: Must not be set 1 1: Microprocessor mode	RW
PM01			RW
PM02	R/W mode select bit (Note 2)	0 : $\overline{RD}$,$\overline{BHE}$,$\overline{WR}$ 1 : $\overline{RD}$,WRH,WRL	RW
PM03	Software reset bit	Setting this bit to "1" resets the microcomputer. When read, its content is "0".	RW
PM04	Multiplexed bus space select bit (Note 2)	b5 b4 0 0 : Multiplexed bus is unused (Separate bus in the entire $\overline{CS}$ space) 0 1 : Allocated to $\overline{CS2}$ space 1 0 : Allocated to $\overline{CS1}$ space 1 1 : Allocated to the entire $\overline{CS}$ space (Note 3)	RW
PM05			RW
PM06	Port $P4_0$ to $P4_3$ function select bit (Note 2)	0 : Address output 1 : Port function (Address is not output)	RW
PM07	BCLK output disable bit (Note 2)	0 : BCLK is output 1 : BCLK is not output (Pin is left high-impedance)	RW

Note 1: Write to this register after setting the PRC1 bit in the PRCR register to "1" (write enable).
Note 2: Effective when the PM01 to PM00 bits are set to "01_2" (memory expansion mode) or "11_2" (microprocessor mode).
Note 3: To set the PM01 to PM00 bits are "01_2" and the PM05 to PM04 bits are "11_2" (multiplexed bus assigned to the entire $\overline{CS}$ space), apply an "H" signal to the BYTE pin (external data bus is 8 bits wide). While the CNVSS pin is held "H" (= V_{CC1}), do not rewrite the PM05 to PM04 bits to "11_2" after reset.
If the PM05 to PM04 bits are set to "11_2" during memory expansion mode, $P3_1$ to $P3_7$ and $P4_0$ to $P4_3$ become I/O ports, in which case the accessible area for each $\overline{CS}$ is 256 bytes.
Note 4: The PM01 to PM00 bits do not change at software reset, watchdog timer reset and oscillation stop detection reset.

Figure 1.6.1. PM0 Register

Processor mode register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
	0						

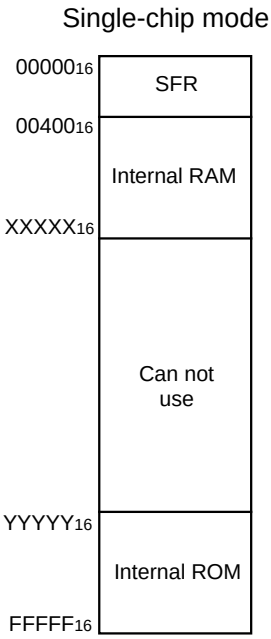
Symbol	Address	After reset
PM1	0005_{16}	$0X001000_2$

Bit symbol	Bit name	Function	RW
PM10	CS2 area switch bit (data block enable bit) (Note 2)	0: 08000_{16} to $26FFF_{16}$ (block A disable) 1: 10000_{16} to $26FFF_{16}$ (block A enable)	RW
PM11	Port $P3_7$ to $P3_4$ function select bit (Note 3)	0 : Address output 1 : Port function	RW
PM12	Watchdog timer function select bit	0 : Watchdog timer interrupt 1 : Watchdog timer reset (Note 4)	RW
PM13	Internal reserved area expansion bit	See Note 6	RW
PM14	Memory area expansion bit (Note 3)	b5 b4 0 0 : 1 Mbyte mode (Do not expand) 0 1 : Must not be set 1 0 : Must not be set 1 1 : 4 Mbyte mode	RW
PM15			RW
— (b6)	Reserved bit	Should be set to “0”.	RW
PM17	Wait bit (Note 5)	0 : No wait state 1 : With wait state (1 wait)	RW

Note 1: Write to this register after setting the PRC1 bit in the PRCR register to “1” (write enable).
Note 2: For the mask ROM version, this bit must be set to “0” . For the flash memory version, the PM10 bit also controls block A by enabling or disabling it. However, the PM10 bit is automatically set to “1” when the FMR01 bit in the FMR0 register is “1” (CPU rewrite mode).
Note 3: Effective when the PM01 to PM00 bits are set to 01_2 (memory expansion mode) or 11_2 (microprocessor mode).
Note 4: PM12 bit is set to “1” by writing a “1” in a program. (Writing a “0” has no effect.)
Note 5: When PM17 bit is set to “1” (with wait state), one wait state is inserted when accessing the internal RAM, internal ROM, or an external area. If the CSiW bit (i = 0 to 3) in the CSR register is “0” (with wait state), the CSi area is always accessed with one or more wait states regardless of whether the PM17 bit is set or not. Where the RDY signal is used or multiplex bus is used, set the CSiW bit to “0” (with wait state).
Note 6: The access area is changed by the PM13 bit as listed in the table below.

Access area		PM13=0	PM13=1
Internal	RAM	Up to addresses 00400_{16} to $03FFF_{16}$ (15 Kbytes)	The entire area is usable
	ROM	Up to addresses $D0000_{16}$ to $FFFFF_{16}$ (192 Kbytes)	The entire area is usable
External		Addresses 04000_{16} to $07FFF_{16}$ are usable Addresses 80000_{16} to $CFFFF_{16}$ are usable	Addresses 04000_{16} to $07FFF_{16}$ are reserved Addresses 80000_{16} to $CFFFF_{16}$ are reserved

Figure 1.6.2. PM1 Register



PM13=0

Internal RAM		Internal ROM	
Capacity	Address XXXXX$_{16}$	Capacity	Address YYYYY16
4K bytes	013FF$_{16}$	48K bytes	F4000$_{16}$
5K bytes	017FF$_{16}$	64K bytes	F0000$_{16}$
10K bytes	02BFF$_{16}$	96K bytes	E8000$_{16}$
12K bytes	033FF$_{16}$	128K bytes	E0000$_{16}$
16K bytes	03FFF$_{16}$(Note 2)	192K bytes	D0000$_{16}$
20K bytes	03FFF$_{16}$(Note 2)	256K bytes	D0000$_{16}$(Note 2)
24K bytes	03FFF$_{16}$(Note 2)	320K bytes	D0000$_{16}$(Note 2)
31K bytes	03FFF$_{16}$(Note 2)	384K bytes	D0000$_{16}$(Note 2)
		512K bytes	D0000$_{16}$(Note 2)

PM13=1

Internal RAM		Internal ROM	
Capacity	Address XXXXX$_{16}$	Capacity	Address YYYYY$_{16}$
4K bytes	013FF$_{16}$	48K bytes	F4000$_{16}$
5K bytes	017FF$_{16}$	64K bytes	F0000$_{16}$
10K bytes	02BFF$_{16}$	96K bytes	E8000$_{16}$
12K bytes	033FF$_{16}$	128K bytes	E0000$_{16}$
16K bytes	043FF$_{16}$	192K bytes	D0000$_{16}$
20K bytes	053FF$_{16}$	256K bytes	C0000$_{16}$
24K bytes	063FF$_{16}$	320K bytes	B0000$_{16}$
31K bytes	07FFF$_{16}$	384K bytes	A0000$_{16}$
		512K bytes	80000$_{16}$

Note 1: For the mask ROM version, set the PM10 bit to “0” (08000$_{16}$ to 26FFF$_{16}$ for $\overline{CS2}$ area).
Note 2: If PM13 bit is set to “0”, 15 Kbytes of the internal RAM and 192 Kbytes of the internal ROM can be used.

Figure 1.6.3. Memory Map in Single Chip Mode

Bus

During memory expansion or microprocessor mode, some pins serve as the bus control pins to perform data input/output to and from external devices. These bus control pins include 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.

Bus Mode

The bus mode, either multiplexed or separate, can be selected using the PM05 to PM04 bits.

Separate Bus

In this bus mode, data and address are separate.

Multiplexed Bus

In this bus mode, data and address are multiplexed. If the data bus is 8 bits wide, D_0 to D_7 and A_0 to A_7 are multiplexed. If the data bus is 16 bits wide, D_0 to D_7 and A_1 to A_8 are multiplexed, with D_8 to D_{15} not multiplexed. In this case, external devices connecting to the multiplexed bus are mapped to the even addresses of the microcomputer.

Bus Control

The following describes the signals needed for accessing external devices and the functionality of software wait.

(1) Address Bus

The address bus consists of 20 lines, A_0 to A_{19}. The address bus width can be chosen to be 12, 16 or 20 bits by using the PM06 bit in the PM0 register and the PM11 bit in the PM1 register. Table 1.7.1 shows the PM06 and PM11 bit set values and address bus widths.

Table 1.7.1. PM06 and PM11 Bits Set Value and Address Bus Width

Set value(Note)	Pin function	Address bus wide
PM11=1 PM06=1	$P3_4$ to $P3_7$ $P4_0$ to $P4_3$	12 bits
PM11=0 PM06=1	A_{12} to A_{15} $P4_0$ to $P4_3$	16 bits
PM11=0 PM06=0	A_{12} to A_{15} A_{16} to A_{19}	20 bits

Note 1: No values other than those shown above can be set.

When processor mode is changed from single-chip mode to memory extension mode, the address bus is indeterminate until any external area is accessed.

(2) Data Bus

When input on the BYTE pin is high, 8 lines D_0 to D_7 comprise the data bus; when input on the BYTE pin is low, 16 lines D_0 to D_{15} comprise the data bus.

Do not change the input level on the BYTE pin while in operation.

(3) Chip Select Signal

The chip select (hereafter referred to as the $\overline{CSi}$) signals are output from the $\overline{CSi}$ (i = 0 to 3) pins. These pins can be chosen to function as I/O ports or as $\overline{CS}$ by using the CSi bit in the CSR register. Figure 1.7.1 shows the CSR register.

During 1 Mbyte mode, the external area can be separated into up to 4 by the $\overline{CSi}$ signal which is output from the $\overline{CSi}$ pin. During 4 Mbyte mode, $\overline{CSi}$ signal or bank number is output from the $\overline{CSi}$ pin. Refer to "Memory space expansion function". Figure 1.7.2 shows the example of address bus and $\overline{CSi}$ signal output in 1 Mbyte mode.

Chip select control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: CSR — Address: 0008_{16} — After reset: 00000001_2

Bit symbol	Bit name	Function	RW
CS0	$\overline{CS0}$ output enable bit	0 : Chip select output disabled (functions as I/O port) 1 : Chip select output enabled	RW
CS1	$\overline{CS1}$ output enable bit		RW
CS2	$\overline{CS2}$ output enable bit		RW
CS3	$\overline{CS3}$ output enable bit		RW
CS0W	$\overline{CS0}$ wait bit	0 : With wait state 1 : Without wait state (Note 1, Note 2, Note 3)	RW
CS1W	$\overline{CS1}$ wait bit		RW
CS2W	$\overline{CS2}$ wait bit		RW
CS3W	$\overline{CS3}$ wait bit		RW

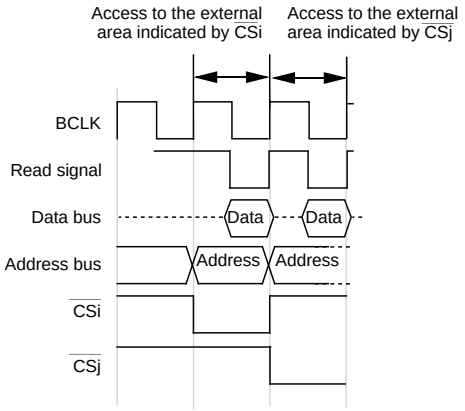
Note 1: Where the $\overline{RDY}$ signal is used in the area indicated by $\overline{CSi}$ (i = 0 to 3) or the multiplex bus is used, set the CSiW bit to "0" (Wait state).
Note 2: If the PM17 bit in the PM1 register is set to "1" (with wait state), the external area indicated by $\overline{CS0}$ to $\overline{CS3}$ is always accessed with one wait state even when the CSiW bit is "1" (without wait state).
Note 3: When the CSiW bit = "0" (with wait state), the number of wait states (interms of clock cycles) can be selected using the CSEi1W to CSEi0W bits in the CSE register.

Figure 1.7.1. CSR Register

Example 1

To access the external area indicated by $\overline{CSj}$ in the next cycle after accessing the external area indicated by $\overline{CSi}$

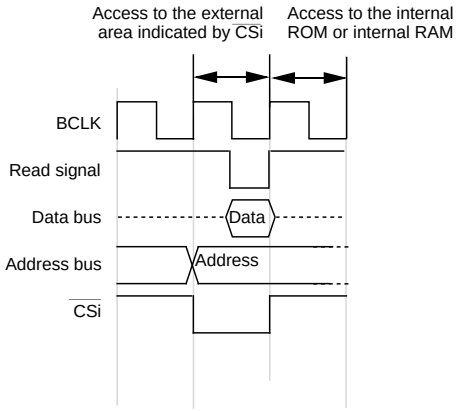
The address bus and the chip select signal both change state between these two cycles.



Example 2

To access the internal ROM or internal RAM in the next cycle after accessing the external area indicated by $\overline{CSi}$

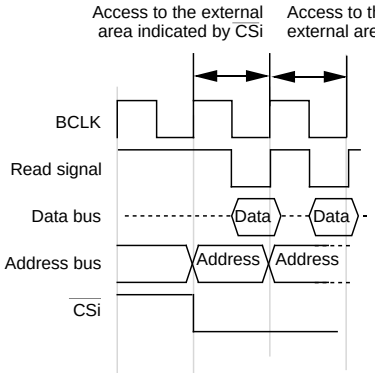
The chip select signal changes state but the address bus does not change state



Example 3

To access the external area indicated by $\overline{CSi}$ in the next cycle after accessing the external area indicated by the same $\overline{CSi}$

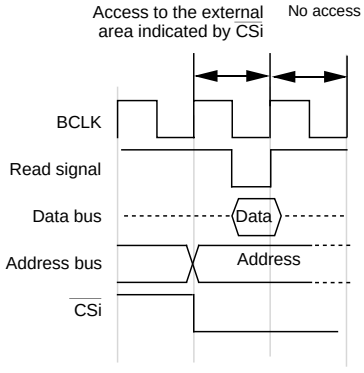
The address bus changes state but the chip select signal does not change state



Example 4

Not to access any area (nor instruction prefetch generated) in the next cycle after accessing the external area indicated by CSi

Neither the address bus nor the chip select signal changes state between these two cycles



Note : These examples show the address bus and chip select signal when accessing areas in two successive cycles. The chip select bus cycle may be extended more than two cycles depending on a combination of these examples.

Shown above is the case where separate bus is selected and the area is accessed for read without wait states. i = 0 to 3, j = 0 to 3 (not including i, however)

Figure 1.7.2. Example of Address Bus and $\overline{CSi}$ Signal Output in 1 Mbyte Mode

(4) Read and Write Signals

When the data bus is 16 bits wide, the read and write signals can be chosen to be a combination of $\overline{RD}$, $\overline{BHE}$ and $\overline{WR}$ or a combination of $\overline{RD}$, $\overline{WRL}$ and $\overline{WRH}$ by using the PM02 bit in the PM0 register. When the data bus is 8 bits wide, use a combination of $\overline{RD}$, $\overline{WR}$ and $\overline{BHE}$.

Table 1.7.2 shows the operation of $\overline{RD}$, $\overline{WRL}$, and $\overline{WRH}$ signals. Table 1.7.3 shows the operation of operation of $\overline{RD}$, $\overline{WR}$, and $\overline{BHE}$ signals.

Table 1.7.2. 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 pin input = "L")	L	H	H	Read data
	H	L	H	Write 1 byte of data to an even address
	H	H	L	Write 1 byte of data to an odd address
	H	L	L	Write data to both even and odd addresses

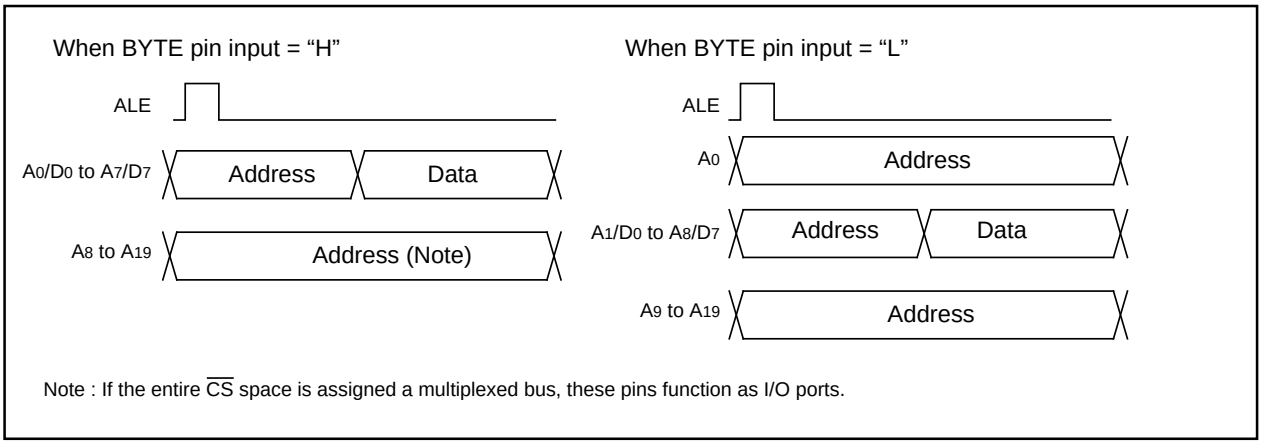
Table 1.7.3. Operation of $\overline{RD}$, $\overline{WR}$ and $\overline{BHE}$ Signals

Data bus width	$\overline{RD}$	$\overline{WR}$	$\overline{BHE}$	A0	Status of external data bus
16-bit (BYTE pin input = "L")	H	L	L	H	Write 1 byte of data to an odd address
	L	H	L	H	Read 1 byte of data from an odd address
	H	L	H	L	Write 1 byte of data to an even address
	L	H	H	L	Read 1 byte of data from an 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 pin input = "H")	H	L	— (Note)	H or L	Write 1 byte of data
	L	H	— (Note)	H or L	Read 1 byte of data

Note : Do not use.

(5) ALE Signal

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



Note : If the entire $\overline{CS}$ space is assigned a multiplexed bus, these pins function as I/O ports.

Figure 1.7.3. ALE Signal, Address Bus, Data Bus

(6) The $\overline{RDY}$ Signal

This signal is provided for accessing external devices which need to be accessed at low speed. If input on the $\overline{RDY}$ pin is asserted low at the last falling edge of BCLK of the bus cycle, one wait state is inserted in the bus cycle. While in a wait state, the following signals retain the state in which they were when the $\overline{RDY}$ signal was acknowledged.

A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{WR}$, $\overline{BHE}$, ALE, $\overline{HLDA}$

Then, when the input on the $\overline{RDY}$ pin is detected high at the falling edge of BCLK, the remaining bus cycle is executed. Figure 1.7.4 shows example in which the wait state was inserted into the read cycle by the $\overline{RDY}$ signal. To use the $\overline{RDY}$ signal, set the corresponding bit (CS3W to CS0W bits) in the CSR register to "0" (with wait state). When not using the $\overline{RDY}$ signal, process the $\overline{RDY}$ pin as an unused pin.

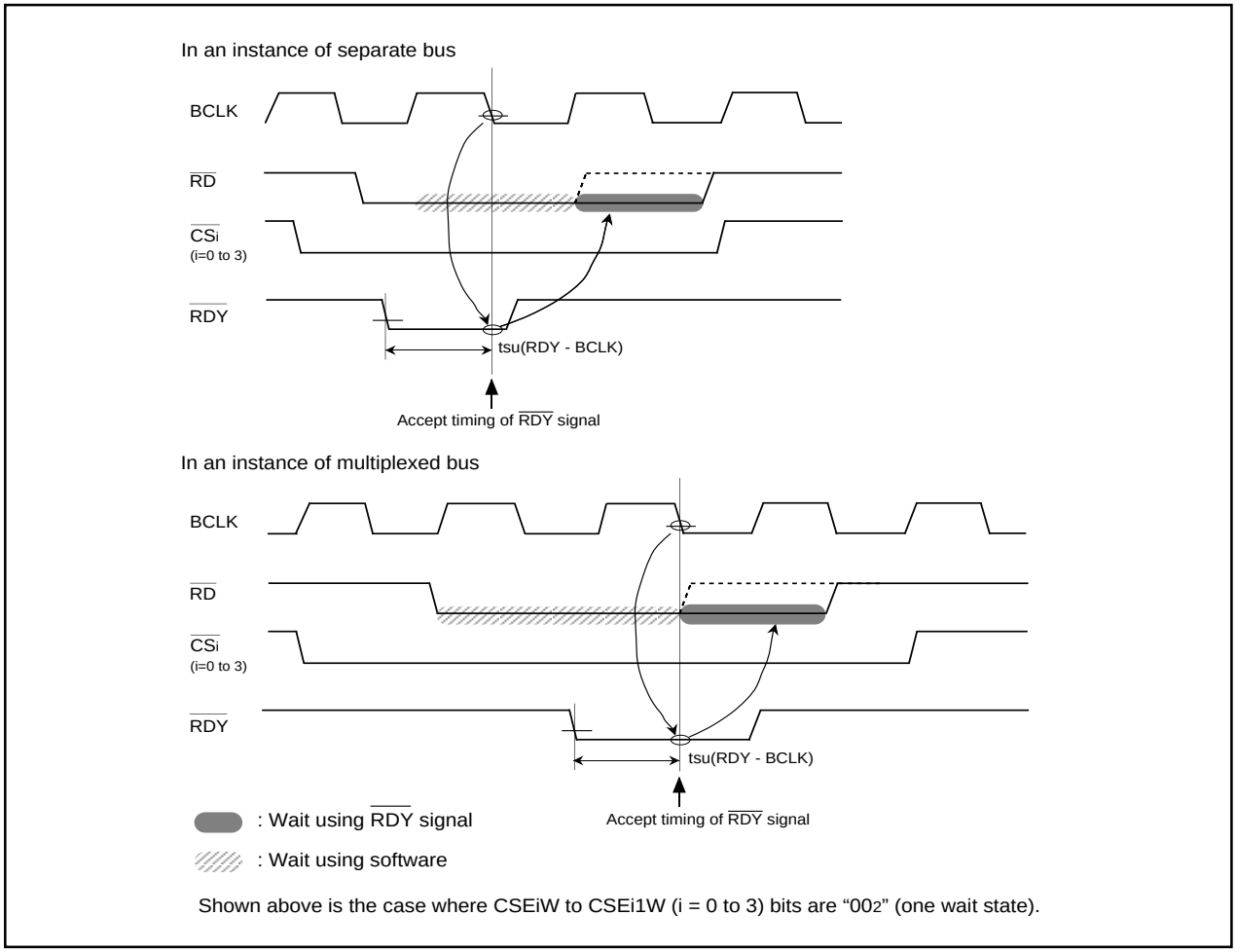


Figure 1.7.4. Example in which Wait State was Inserted into Read Cycle by $\overline{RDY}$ Signal

(7) Hold Signal

This signal is used to transfer control of the bus from the CPU or DMA to an external circuit. When input on the $\overline{HOLD}$ pin is asserted “L”, the microcomputer goes to a hold state after completing the bus access then in progress. While the $\overline{HOLD}$ pin is held “L”, the microcomputer remains in a hold state, outputting a low signal from the $\overline{HLDA}$ pin.

Table 1.7.4 shows the microcomputer status in the hold state.

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

$\overline{HOLD}$ > DMAC > CPU

Figure 1.7.5. Bus-using Priorities

Table 1.7.4. Microcomputer Status in Hold State

Item		Status
BCLK		Output
A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{WR}$, $\overline{BHE}$		High-impedance
I/O ports	P0, P1, P3, P4(Note 2)	High-impedance
	P6 to P14(Note 1)	Maintains status when hold signal is received
$\overline{HLDA}$		Output “L”
Internal peripheral circuits		ON (but watchdog timer stops)
ALE signal		Undefined

Note 1: P11 to P14 are included in the 128-pin version.

Note 2: When I/O port function is selected.

(8) BCLK Output

If the PM07 bit in the PM0 register is set to “0” (output enable), a clock with the same frequency as that of the CPU clock is output as BCLK from the BCLK pin. Refer to “CPU clock and pheripheral clock”.

Table 1.7.5. Pin Functions for Each Processor Mode

Processor mode		Memory expansion mode or microprocessor mode				Memory expansion mode
PM05–PM04 bits		00_2(separate bus)		01_2($\overline{CS2}$ is for multiplexed bus and others are for separate bus) 10_2($\overline{CS1}$ is for multiplexed bus and others are for separate bus)		11_2(multiplexed bus for the entire space) (Note 1)
Data bus width BYTE pin		8 bits "H"	16 bits "L"	8 bits "H"	16 bits "L"	8 bits "H"
$P0_0$ to $P0_7$		D_0 to D_7	D_0 to D_7	D_0 to D_7	D_0 to D_7	I/O ports
$P1_0$ to $P1_7$		I/O ports	D_8 to D_{15}	I/O ports	D_8 to D_{15}	I/O ports
$P2_0$		A_0	A_0	A_0/D_0(Note 2)	A_0	A_0/D_0
$P2_1$ to $P2_7$		A_1 to A_7	A_1 to A_7	A_1 to A_7/D_1 to D_7 (Note 2)	A_1 to A_7/D_0 to D_6 (Note 2)	A_1 to A_7/D_1 to D_7
$P3_0$		A_8	A_8	A_8	A_8/D_7(Note 2)	A_8
$P3_1$ to $P3_3$		A_9 to A_{11}				I/O ports
$P3_4$ to $P3_7$	PM11=0	A_{12} to A_{15}				I/O ports
	PM11=1	I/O ports				
$P4_0$ to $P4_3$	PM06=0	A_{16} to A_{19}				I/O ports
	PM06=1	I/O ports				
$P4_4$	CS0=0	I/O ports				
	CS0=1	$\overline{CS0}$				
$P4_5$	CS1=0	I/O ports				
	CS1=1	$\overline{CS1}$				
$P4_6$	CS2=0	I/O ports				
	CS2=1	$\overline{CS2}$				
$P4_7$	CS3=0	I/O ports				
	CS3=1	$\overline{CS3}$				
$P5_0$	PM02=0	$\overline{WR}$				
	PM02=1	—— (Note 3)	$\overline{WRL}$	—— (Note 3)	$\overline{WRL}$	—— (Note 3)
$P5_1$	PM02=0	$\overline{BHE}$				
	PM02=1	—— (Note 3)	$\overline{WRH}$	—— (Note 3)	$\overline{WRH}$	—— (Note 3)
$P5_2$		$\overline{RD}$				
$P5_3$		BCLK				
$P5_4$		$\overline{HLDA}$				
$P5_5$		$\overline{HOLD}$				
$P5_6$		ALE				
$P5_7$		$\overline{RDY}$				

I/O ports: Function as I/O ports or peripheral function I/O pins.

Note 1: To set the PM01 to PM00 bits are set to "01_2" and the PM05 to PM04 bits are set to "11_2" (multiplexed bus assigned to the entire $\overline{CS}$ space), apply "H" to the BYTE pin (external data bus 8 bits wide). While the CNV_{SS} pin is held "H" (= V_{CC1}), do not rewrite the PM05 to PM04 bits to "11_2" after reset. If the PM05 to PM04 bits are set to "11_2" during memory expansion mode, $P3_1$ to $P3_7$ and $P4_0$ to $P4_3$ become I/O ports, in which case the accessible area for each $\overline{CS}$ is 256 bytes.

Note 2: In separate bus mode, these pins serve as the address bus.

Note 3: If the data bus is 8 bits wide, make sure the PM02 bit is set to "0" ($\overline{RD}$, $\overline{BHE}$, $\overline{WR}$).

(9) External Bus Status When Internal Area Accessed

Table 1.7.6 shows the external bus status when the internal area is accessed.

Table 1.7.6. External Bus Status When Internal Area Accessed

Item		SFR accessed	Internal ROM, RAM accessed
A_0 to A_{19}		Address output	Maintain status before accessed address of external area or SFR
D_0 to D_{15}	When read	High-impedance	High-impedance
	When write	Output data	Undefined
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$ output	Output “H”
$\overline{BHE}$		$\overline{BHE}$ output	Maintain status before accessed status of external area or SFR
$\overline{CS0}$ to $\overline{CS3}$		Output “H”	Output “H”
ALE		Output “L”	Output “L”

(10) Software Wait

Software wait states can be inserted by using the PM17 bit in the PM1 register, the CS0W to CS3W bits in the CSR register, and the CSE register.

To use the $\overline{RDY}$ signal, set the corresponding CS3W to CS0W bit to “0”. Figure 1.7.6 shows the CSE register. Table 1.7.7 shows the software wait related bits and bus cycles. Figure 1.7.7 and 1.7.8 show the typical bus timings using software wait.

Chip select expansion control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: CSE
Address: $001B_{16}$
After reset: 00_{16}

Bit symbol	Bit name	Function	RW
CSE00W	$\overline{CS0}$ wait expansion bit (Note)	b1 b0 0 0: 1 wait 0 1: 2 waits 1 0: 3 waits 1 1: Must not be set	RW
CSE01W			RW
CSE10W	$\overline{CS1}$ wait expansion bit (Note)	b3 b2 0 0: 1 wait 0 1: 2 waits 1 0: 3 waits 1 1: Must not be set	RW
CSE11W			RW
CSE20W	$\overline{CS2}$ wait expansion bit (Note)	b5 b4 0 0: 1 wait 0 1: 2 waits 1 0: 3 waits 1 1: Must not be set	RW
CSE21W			RW
CSE30W	$\overline{CS3}$ wait expansion bit (Note)	b7 b6 0 0: 1 wait 0 1: 2 waits 1 0: 3 waits 1 1: Must not be set	RW
CSE31W			RW

Note: Set the CSiW bit (i = 0 to 3) in the CSR register to “0” (with wait state) before writing to the CSEi1W to CSEi0W bits. If the CSiW bit needs to be set to “1” (without wait state), set the CSEi1W to CSEi0W bits to “00_2” before setting it.

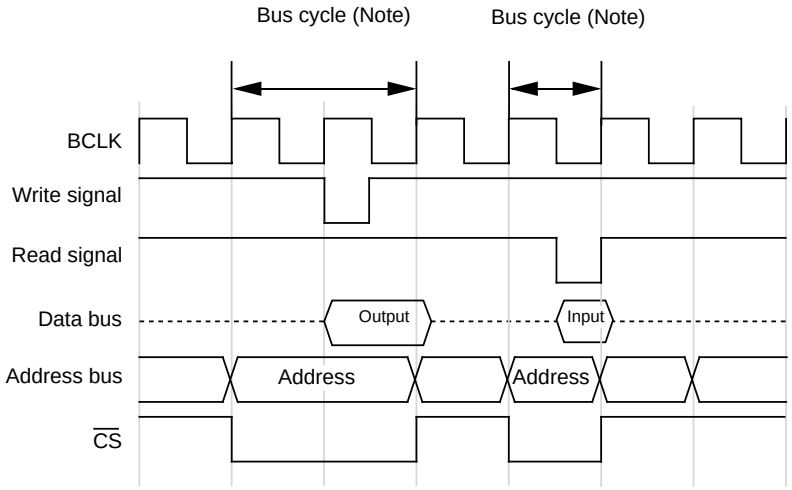
Figure 1.7.6. CSE Register

Table 1.7.7. Bit and Bus Cycle Related to Software Wait

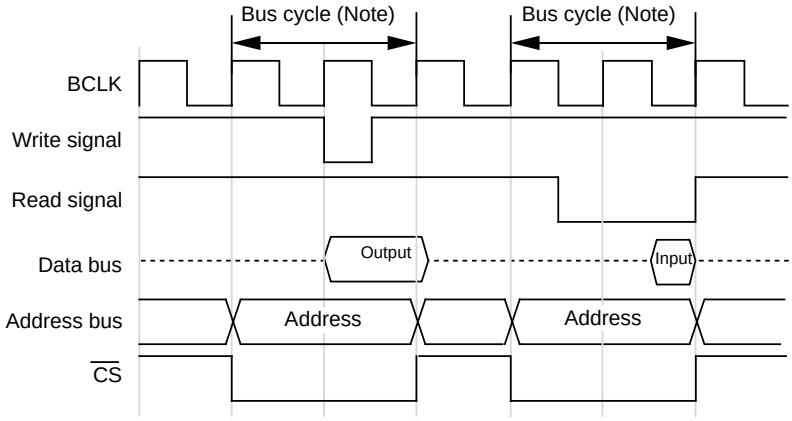
Area	Bus mode	PM1 register PM17 bit	CSR register CS3W bit (Note 1) CS2W bit (Note 1) CS1W bit (Note 1) CS0W bit (Note 1)	CSE register CSE31W to CSE30W bit CSE21W to CSE20W bit CSE11W to CSE10W bit CSE01W to CSE00W bit	Software wait	Bus cycle
Internal RAM, ROM	—	0	—	—	No wait	1 BCLK cycle (Note 3)
	—	1	—	—	1 wait	2 BCLK cycles
External area	Separate bus	0	1	00_2	No wait	1 BCLK cycle (read)
						2 BCLK cycles (write)
		—	0	00_2	1 wait	2 BCLK cycles (Note 3)
		—	0	01_2	2 waits	3 BCLK cycles
		—	0	10_2	3 waits	4 BCLK cycles
		1	1	00_2	1 wait	2 BCLK cycles
	Multiplexed bus (Note 2)	—	0	00_2	1 wait	3 BCLK cycles
		—	0	01_2	2 waits	3 BCLK cycles
		—	0	10_2	3 waits	4 BCLK cycles
		1	0	00_2	1 wait	3 BCLK cycles

Note 1: To use the $\overline{RDY}$ signal, set this bit to “0”.
Note 2: To access in multiplexed bus mode, set the corresponding bit of CS0W to CS3W to “0” (with wait state).
Note 3: After reset, the PM17 bit is set to “0” (without wait state), all of the CS0W to CS3W bits are set to “0” (with wait state), and the CSE register is set to “00_{16}” (one wait state for $\overline{CS0}$ to $\overline{CS3}$). Therefore, the internal RAM and internal ROM are accessed with no wait states, and all external areas are accessed with one wait state.

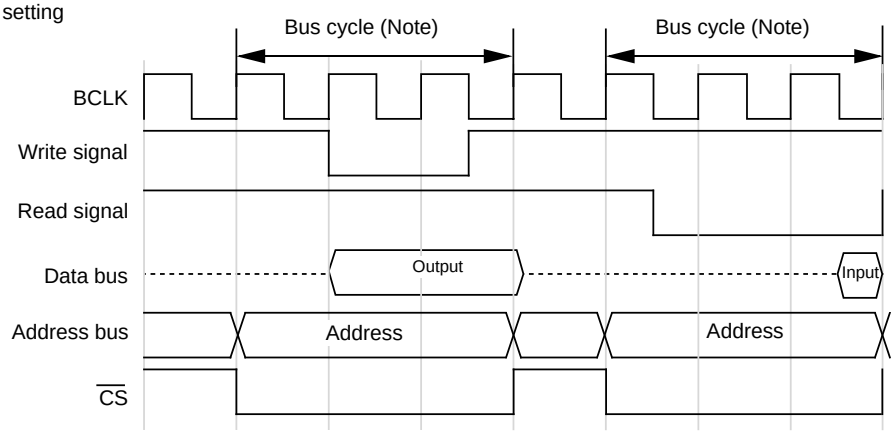
(1) Separate bus, No wait setting



(2) Separate bus, 1-wait setting



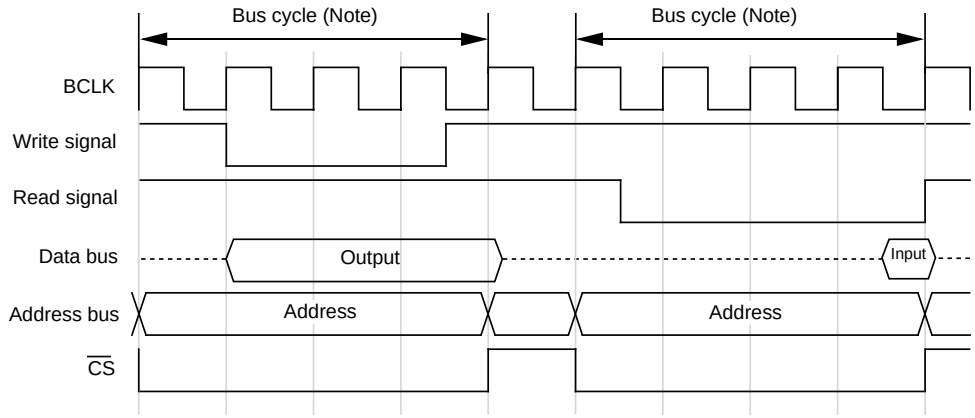
(3) Separate bus, 2-wait setting



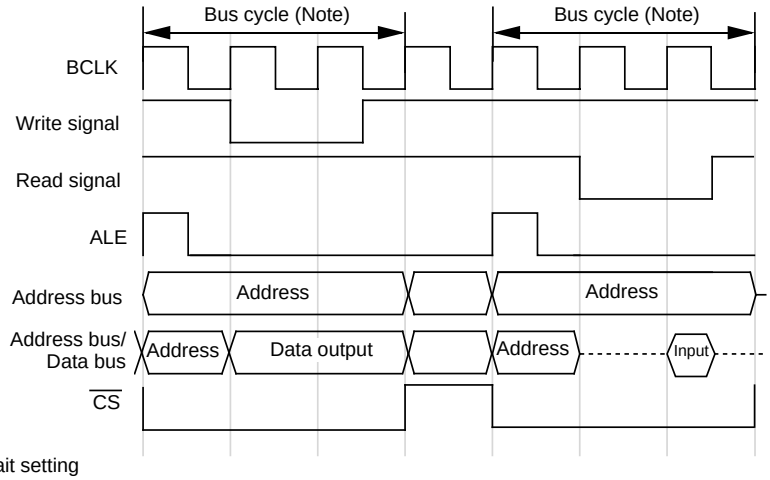
Note : These example timing charts indicate bus cycle length. After this bus cycle sometimes come read and write cycles in succession.

Figure 1.7.7. Typical Bus Timings Using Software Wait (1)

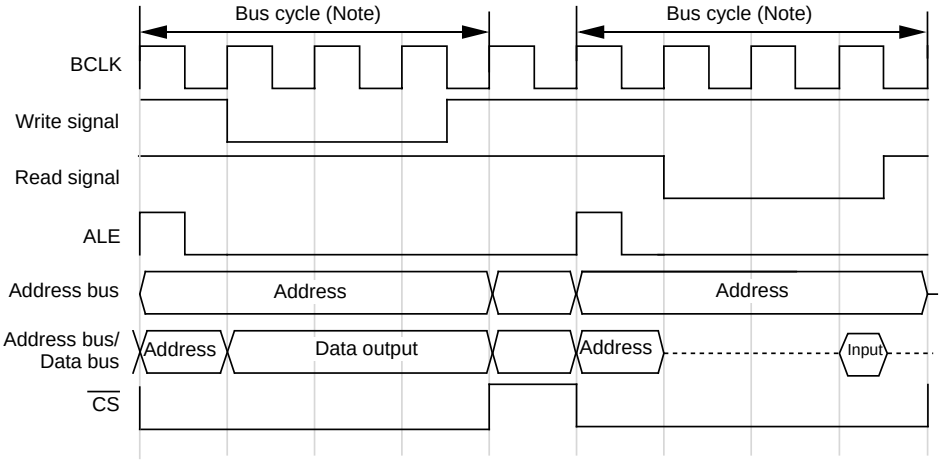
(1) Separate bus, 3-wait setting



(2)Multiplexed bus, 1- or 2-wait setting



(3)Multiplexed bus, 3-wait setting



Note : These example timing charts indicate bus cycle length. After this bus cycle sometimes come read and write cycles in succession.

Figure 1.7.8. Typical Bus Timings Using Software Wait (2)

Memory Space Expansion Function

The following describes a memory space extension function.

During memory expansion or microprocessor mode, the memory space expansion function allows the access space to be expanded using the appropriate register bits.

Table 1.8.1 shows the way of setting memory space expansion function, memory spaces.

Table 1.8.1. The Way of Setting Memory Space Expansion Function, Memory Space

Memory space expansion function	How to set (PM15 to PM14)	Memory space
1 Mbytes mode	00_2	1 Mbytes (no expansion)
4 Mbytes mode	11_2	4 Mbytes

(1) 1 Mbyte Mode

In this mode, the memory space is 1 Mbytes. In 1 Mbyte mode, the external area to be accessed is specified using the $\overline{CSi}$ (i = 0 to 3) signals (hereafter referred to as the $\overline{CSi}$ area). Figures 1.8.2 to 1.8.3 show the memory mapping and $\overline{CS}$ area in 1 Mbyte mode.

(2) 4 Mbyte Mode

In this mode, the memory space is 4 Mbytes. Figure 1.8.1 shows the DBR register. The BSR2 to BSR0 bits select a bank number which is to be accessed to read or write data. Setting the OFS bit to “1” (with offset) allows the accessed address to be offset by 40000_{16}.

In 4 Mbyte mode, the $\overline{CSi}$ (i=0 to 3) pin functions differently for each area to be accessed.

Addresses 04000_{16} to $3FFFF_{16}$, $C0000_{16}$ to $FFFFF_{16}$

- The $\overline{CSi}$ signal is output from the $\overline{CSi}$ pin (same operation as 1 Mbyte mode. However the last address of $\overline{CS1}$ area is $3FFFF_{16}$)

Addresses 40000_{16} to $BFFFF_{16}$

- The $\overline{CS0}$ pin outputs “L”
- The $\overline{CS1}$ to $\overline{CS3}$ pins output the value of the BSR2 to BSR0 bits (bank number)

Figures 1.8.4 to 1.8.5 show the memory mapping and $\overline{CS}$ area in 4 Mbyte mode. Note that banks 0 to 6 are data-only areas. Locate the program in bank 7 or the $\overline{CSi}$ area.

Data bank register (Note)

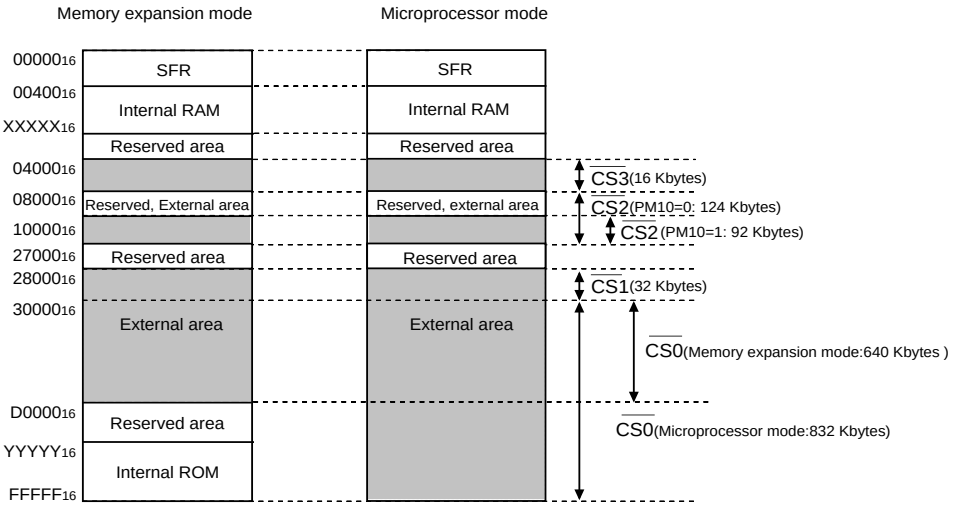
b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
DBR	$000B_{16}$	00_{16}

Bit symbol	Bit name	Description	RW
— (b1-b0)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—
OFS	Offset bit	0: Not offset 1: Offset	RW
BSR0	Bank selection bits	b5 b4 b3 0 0 0: Bank 0 0 1 0: Bank 2 1 0 0: Bank 4 1 1 0: Bank 6 b5 b4 b3 0 0 1: Bank 1 0 1 1: Bank 3 1 0 1: Bank 5 1 1 1: Bank 7	RW
BSR1			RW
BSR2			RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

Note : Effective when the PM01 to PM00 bits are set to “01_2” (memory expansion mode) or “11_2” (microprocessor mode).

Figure 1.8.1. DBR Register

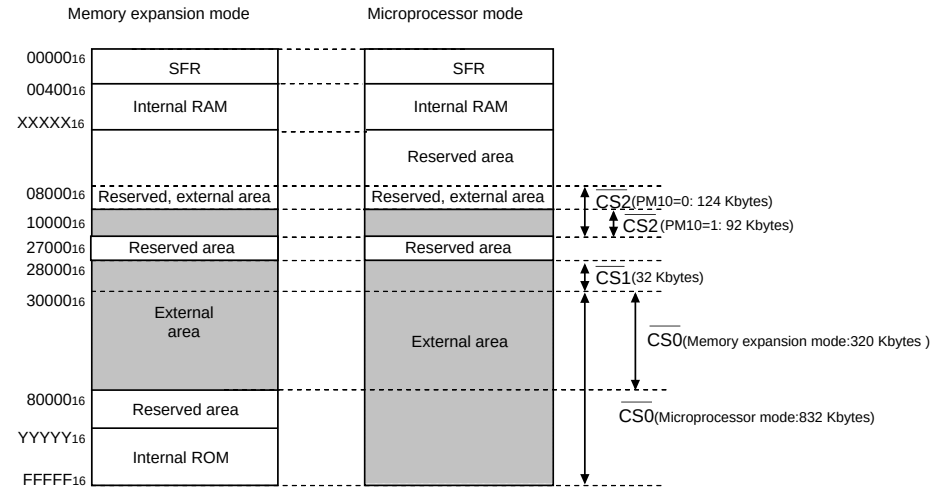


PM13=0

Internal RAM		Internal ROM		External area			
Capacity	Address XXXXX₁₆	Capacity	Address YYYYY₁₆	CS0	CS1	CS2	CS3
4 Kbytes	013FF₁₆	48 Kbytes	F4000₁₆	Memory expansion mode 30000₁₆–CFFFF₁₆ Microprocessor mode 30000₁₆–FFFFF₁₆	28000₁₆–2FFFF₁₆	When PM10=0 08000₁₆–26FFF₁₆ When PM10=1 10000₁₆–26FFF₁₆	04000₁₆–07FFF₁₆
5 Kbytes	017FF₁₆	64 Kbytes	F0000₁₆				
10 Kbytes	02BFF₁₆	96 Kbytes	E8000₁₆				
12 Kbytes	033FF₁₆	128 Kbytes	E0000₁₆				
16 Kbytes	03FFF₁₆(Note)	192 Kbytes	D0000₁₆				
20 Kbytes	03FFF₁₆(Note)	256 Kbytes	D0000₁₆(Note)				
24 Kbytes	03FFF₁₆(Note)	320 Kbytes	D0000₁₆(Note)				
31 Kbytes	03FFF₁₆(Note)	384 Kbytes	D0000₁₆(Note)				
		512 Kbytes	D0000₁₆(Note)				

Note : If PM13 bit is set to “0”, 15 Kbytes of the internal RAM and 192 Kbytes of the internal ROM can be used.

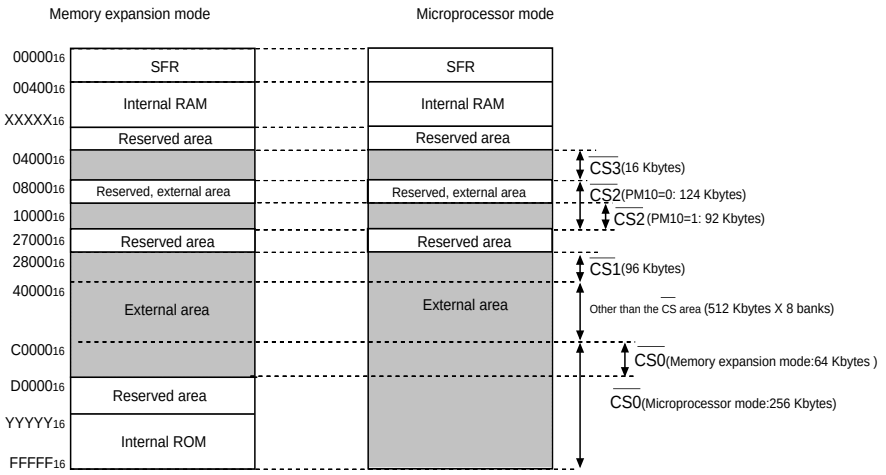
Figure 1.8.2. Memory Mapping and CS Area in 1 Mbyte Mode (PM13=0)



PM13=1

Internal RAM		Internal ROM		External area			
Capacity	Address XXXXX₁₆	Capacity	Address YYYYY₁₆	CS0	CS1	CS2	CS3
4 Kbytes	013FF₁₆	48 Kbytes	F4000₁₆	Memory expansion mode 30000₁₆–7FFFF₁₆ Microprocessor mode 30000₁₆–FFFFF₁₆	28000₁₆–2FFFF₁₆	When PM10=0 08000₁₆–26FFF₁₆ When PM10=1 10000₁₆–26FFF₁₆	No area
5 Kbytes	017FF₁₆	64 Kbytes	F0000₁₆				
10 Kbytes	02BFF₁₆	96 Kbytes	E8000₁₆				
12 Kbytes	033FF₁₆	128 Kbytes	E0000₁₆				
16 Kbytes	043FF₁₆	192 Kbytes	D0000₁₆				
20 Kbytes	053FF₁₆	256 Kbytes	C0000₁₆				
24 Kbytes	063FF₁₆	320 Kbytes	B0000₁₆				
31 Kbytes	07FFF₁₆	384 Kbytes	A0000₁₆				
		512 Kbytes	80000₁₆				

Figure 1.8.3. Memory Mapping and CS Area in 1 Mbyte Mode (PM13=1)

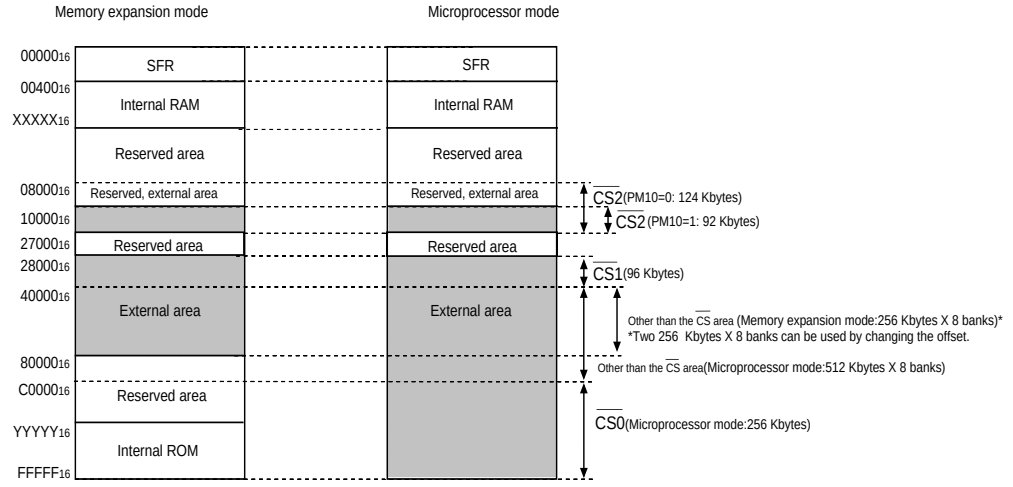


PM13=0

Internal RAM		Internal ROM		External area				
Capacity	Address XXXXX₁₆	Capacity	Address YYYYY₁₆	CS0	CS1	CS2	CS3	Other than the CS area (Note 1)
4 Kbytes	013FF₁₆	48 Kbytes	F4000₁₆	Memory expansion mode C0000₁₆–CFFFF₁₆ Microprocessor mode C0000₁₆–FFFFF₁₆	28000₁₆–3FFFF₁₆	When PM10=0 08000₁₆–26FFF₁₆ When PM10=1 10000₁₆–26FFF₁₆	04000₁₆–07FFF₁₆	40000₁₆–BFFFF₁₆
5 Kbytes	017FF₁₆	64 Kbytes	F0000₁₆					
10 Kbytes	02BFF₁₆	96 Kbytes	E8000₁₆					
12 Kbytes	033FF₁₆	128 Kbytes	E0000₁₆					
16 Kbytes	03FFF₁₆(Note2)	192 Kbytes	D0000₁₆					
20 Kbytes	03FFF₁₆(Note2)	256 Kbytes	D0000₁₆(Note2)					
24 Kbytes	03FFF₁₆(Note2)	320 Kbytes	D0000₁₆(Note2)					
31 Kbytes	03FFF₁₆(Note2)	384 Kbytes	D0000₁₆(Note2)					
		512 Kbytes	D0000₁₆(Note2)					

Note 1: The CS0 pin outputs a low signal, and the CS1–CS3 pins output a bank number.
Note 2: If PM13 bit is set to "0", 15 Kbytes of the internal RAM and 192 Kbytes of the internal ROM can be used.

Figure 1.8.4. Memory Mapping and CS Area in 4 Mbyte Mode (PM13=0)



PM13=1

Internal RAM		Internal ROM		External area				
Capacity	Address XXXXX₁₆	Capacity	Address YYYYY₁₆	CS0	CS1	CS2	CS3	Other than the CS area (Note)
4 Kbytes	013FF₁₆	48 Kbytes	F4000₁₆	Microprocessor mode C0000₁₆–FFFFF₁₆	28000₁₆–3FFFF₁₆	When PM10=0 08000₁₆–26FFF₁₆ When PM10=1 10000₁₆–26FFF₁₆	No area	Memory expansion mode 04000₁₆–7FFF₁₆ Microprocessor mode 04000₁₆–BFFFF₁₆
5 Kbytes	017FF₁₆	64 Kbytes	F0000₁₆					
10 Kbytes	02BFF₁₆	96 Kbytes	E8000₁₆					
12 Kbytes	033FF₁₆	128 Kbytes	E0000₁₆					
16 Kbytes	043FF₁₆	192 Kbytes	D0000₁₆					
20 Kbytes	053FF₁₆	256 Kbytes	C0000₁₆					
24 Kbytes	063FF₁₆	320 Kbytes	B0000₁₆					
31 Kbytes	07FFF₁₆	384 Kbytes	A0000₁₆					
		512 Kbytes	80000₁₆					

Note : The CS0 pin outputs a low signal, and the CS1–CS3 pins output a bank number.

Figure 1.8.5. Memory Mapping and CS Area in 4 Mbyte Mode (PM13=1)

Figure 1.8.6 shows the external memory connect example in 4 Mbyte mode.
In this example, the $\overline{CS}$ pin of 4-Mbyte ROM is connected to the $\overline{CS0}$ pin of microcomputer. The address input AD21, AD20 and AD19 pins are connected to the $\overline{CS3}$, $\overline{CS2}$ and $\overline{CS1}$ pins of microcomputer, respectively. The address input AD18 pin is connected to the A19 pin of microcomputer. Figures 1.8.7 to 1.8.9 show the relationship of addresses between the 4-Mbyte ROM and the microcomputer for the case of a connection example in Figure 1.8.6.
In microprocessor mode, or in memory expansion mode where the PM13 bit is “0”, banks are located every 512 Kbytes. Setting the OFS bit to “1” allows the accessed address to be offset by 40000_{16}, so that even the data overlapping a bank boundary can be accessed in succession.
In memory expansion mode where the PM13 bit is “1”, each 512-Kbyte bank can be accessed in 256 Kbyte units by switching them over with the OFS bit.
Because the SRAM can be accessed on condition that the chip select signals S2 = “H” and $\overline{S1}$ =“L”, $\overline{CS0}$ and $\overline{CS2}$ can be connected to $\overline{S2}$ and $\overline{S1}$, respectively. If the SRAM does not have the input pins to accept “H” active and “L” active chip select signals, $\overline{CS0}$ and $\overline{CS2}$ should be decoded external to the chip.

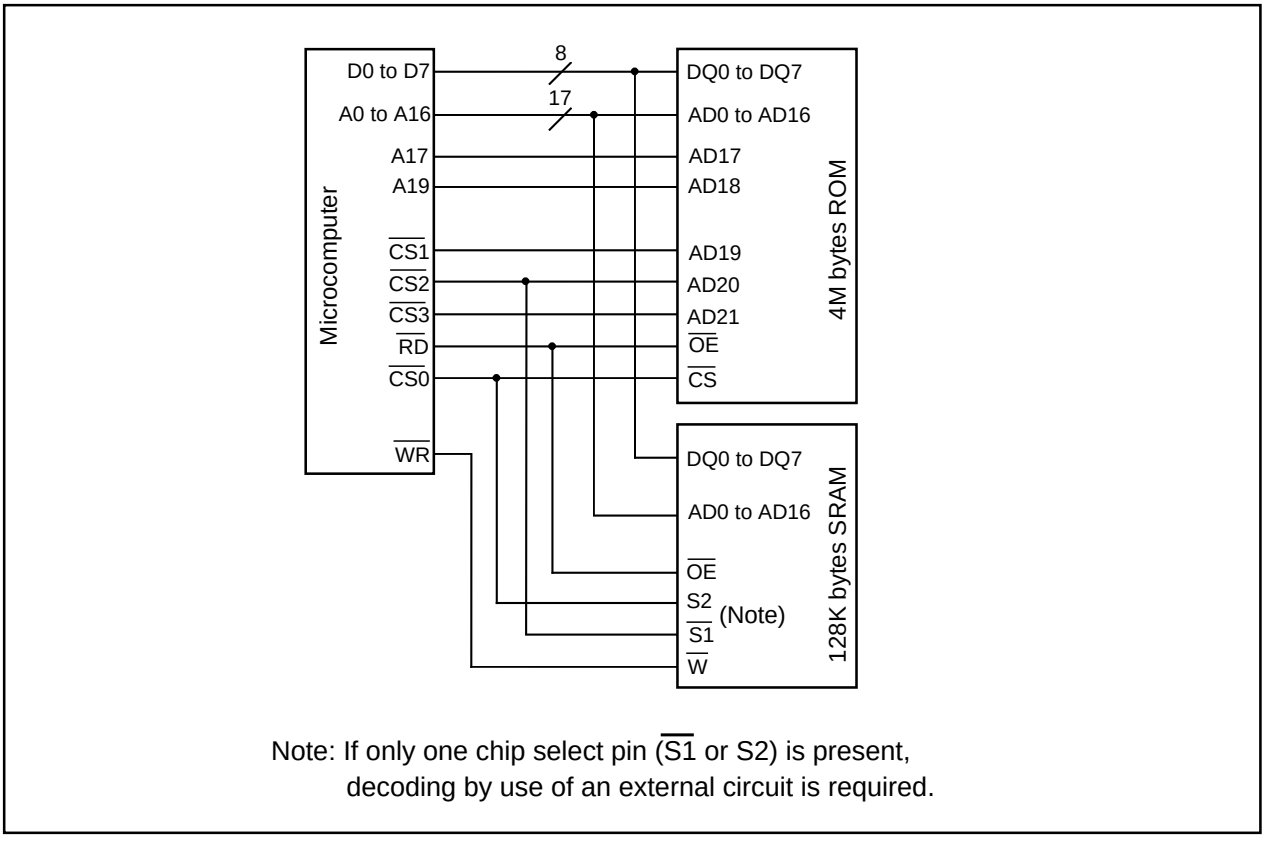
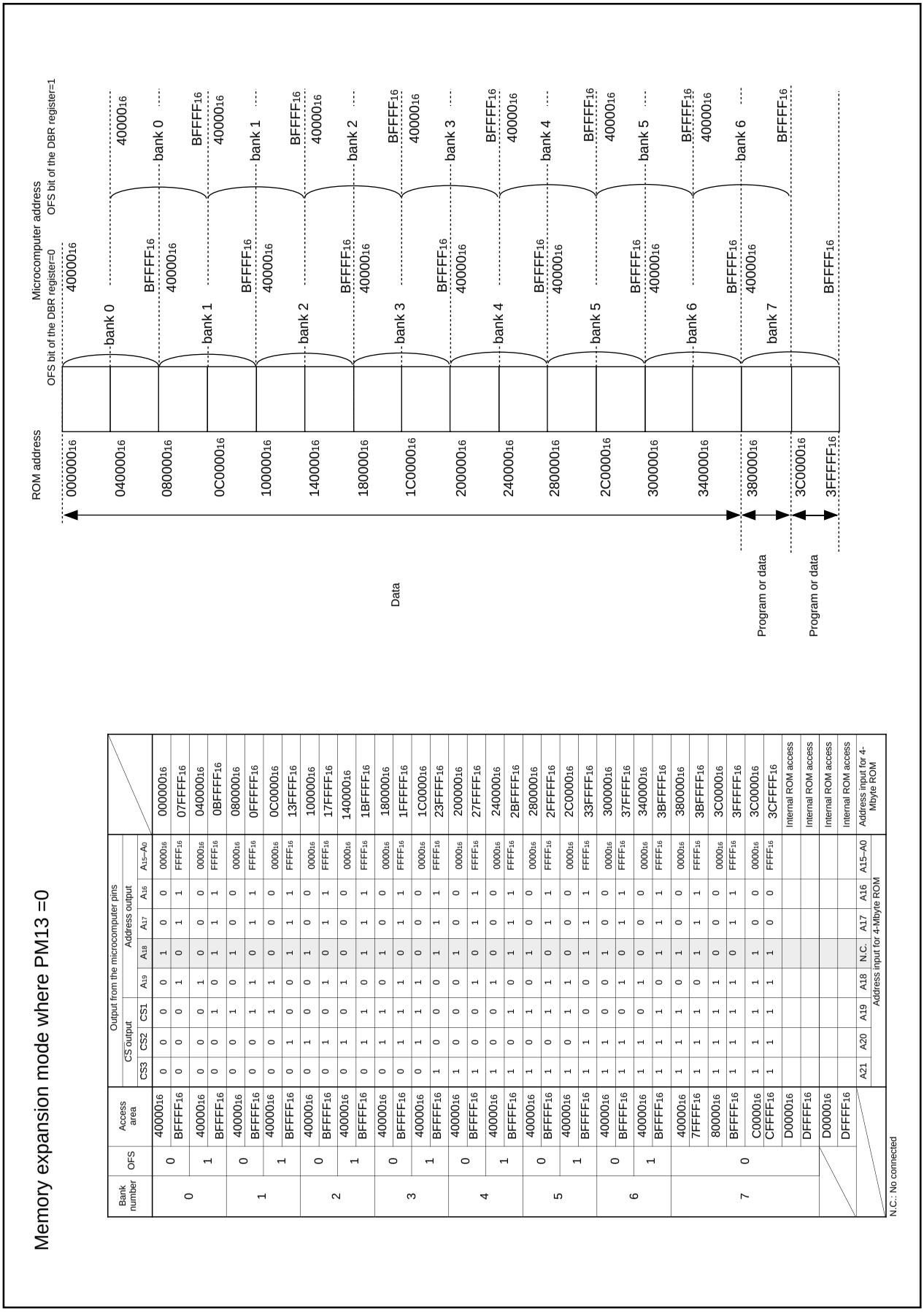


Figure 1.8.6. External Memory Connect Example in 4M Byte Mode

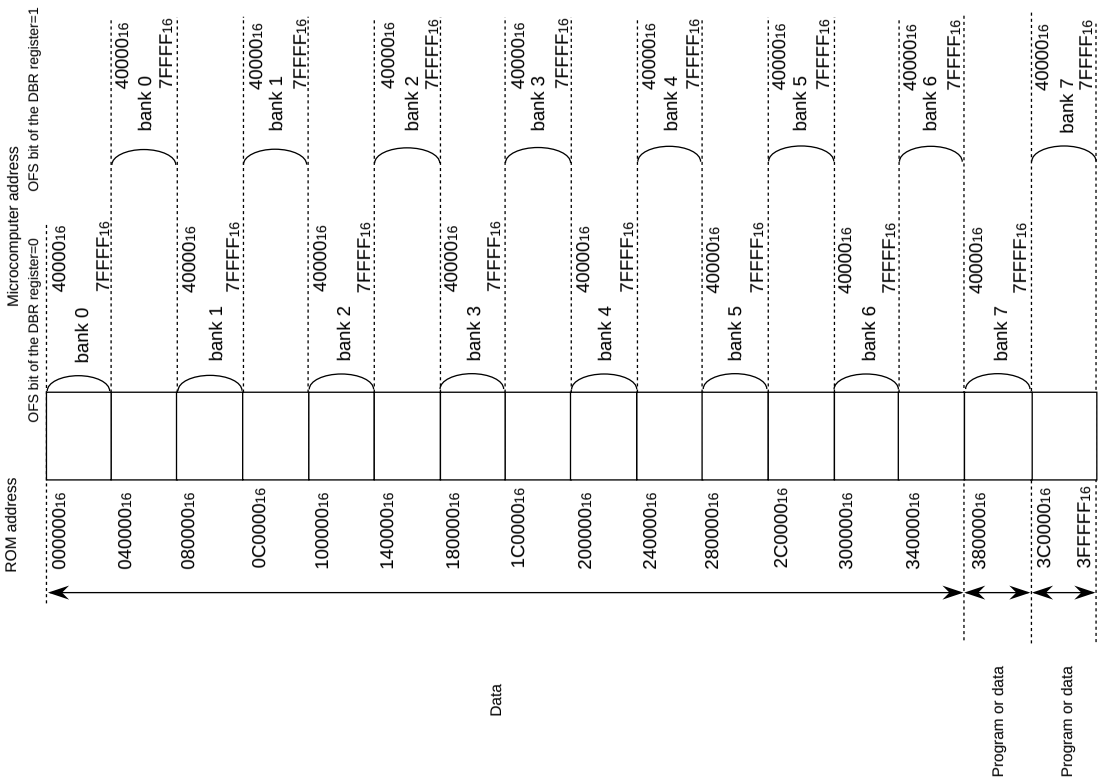


Bank number	OFS	Access area	CS output CS3	CS output CS2	CS output CS1	Output from the microcomputer pins: Address output A19	A18	A17	A16	A15–A0	
0	0	400000₁₆	0	0	0	0	1	0	0	0000₁₆	000000₁₆
		BFFFF₁₆	0	0	0	1	0	1	1	FFFF₁₆	07FFFF₁₆
	1	400000₁₆	0	0	0	1	0	0	0	0000₁₆	040000₁₆
		BFFFF₁₆	0	0	1	0	1	1	1	FFFF₁₆	0BFFFF₁₆
1	0	400000₁₆	0	0	1	0	0	0	0	0000₁₆	080000₁₆
		BFFFF₁₆	0	0	1	1	0	1	1	FFFF₁₆	0FFFFF₁₆
	1	400000₁₆	0	0	1	1	0	0	0	0000₁₆	0C0000₁₆
		BFFFF₁₆	0	1	0	0	1	1	1	FFFF₁₆	13FFFF₁₆
2	0	400000₁₆	0	1	0	0	1	0	0	0000₁₆	100000₁₆
		BFFFF₁₆	0	1	0	1	0	1	1	FFFF₁₆	17FFFF₁₆
	1	400000₁₆	0	1	0	1	0	0	0	0000₁₆	140000₁₆
		BFFFF₁₆	0	1	1	0	1	1	1	FFFF₁₆	1BFFFF₁₆
3	0	400000₁₆	0	1	1	0	1	0	0	0000₁₆	180000₁₆
		BFFFF₁₆	0	1	1	1	0	1	1	FFFF₁₆	1FFFFF₁₆
	1	400000₁₆	0	1	1	1	0	0	0	0000₁₆	1C0000₁₆
		BFFFF₁₆	1	0	0	0	1	1	1	FFFF₁₆	23FFFF₁₆
4	0	400000₁₆	1	0	0	0	1	0	0	0000₁₆	200000₁₆
		BFFFF₁₆	1	0	0	1	0	1	1	FFFF₁₆	27FFFF₁₆
	1	400000₁₆	1	0	0	1	0	0	0	0000₁₆	240000₁₆
		BFFFF₁₆	1	0	1	0	1	1	1	FFFF₁₆	2BFFFF₁₆
5	0	400000₁₆	1	0	1	0	1	0	0	0000₁₆	280000₁₆
		BFFFF₁₆	1	0	1	1	0	1	1	FFFF₁₆	2FFFFF₁₆
	1	400000₁₆	1	0	1	1	0	0	0	0000₁₆	2C0000₁₆
		BFFFF₁₆	1	1	0	0	1	1	1	FFFF₁₆	33FFFF₁₆
6	0	400000₁₆	1	1	0	0	1	0	0	0000₁₆	300000₁₆
		BFFFF₁₆	1	1	0	1	0	1	1	FFFF₁₆	37FFFF₁₆
	1	400000₁₆	1	1	0	1	0	0	0	0000₁₆	340000₁₆
		BFFFF₁₆	1	1	1	0	1	1	1	FFFF₁₆	3BFFFF₁₆
7	0	400000₁₆	1	1	1	0	1	0	0	0000₁₆	380000₁₆
		7FFFF₁₆	1	1	1	0	1	1	1	FFFF₁₆	3BFFFF₁₆
		800000₁₆	1	1	1	1	0	0	0	0000₁₆	3C0000₁₆
		BFFFF₁₆	1	1	1	1	0	1	1	FFFF₁₆	3FFFFF₁₆
		C0000₁₆	1	1	1	1	1	0	0	0000₁₆	3C0000₁₆
		CFFFF₁₆	1	1	1	1	1	0	0	FFFF₁₆	3CFFFF₁₆
		D0000₁₆									Internal ROM access
		DFFFF₁₆									Internal ROM access
		D0000₁₆									Internal ROM access
		DFFFF₁₆									Internal ROM access
			A21	A20	A19	A18	N.C.	A17	A16	A15–A0	Address input for 4-Mbyte ROM
			Address input for 4-Mbyte ROM								

N.C.: No connected

Figure 1.8.7. Relationship Between Addresses on 4-M Byte ROM and Those on Microcomputer (1)

Memory expansion mode where PM13 =1

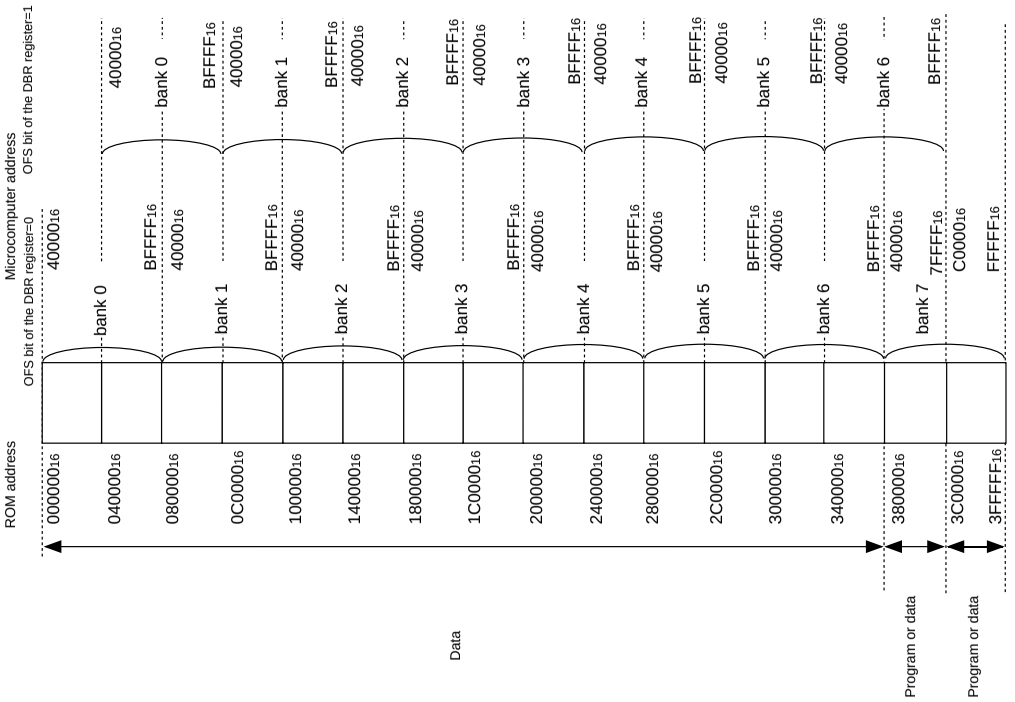


Bank number	OFS	Access area	Output from the microcomputer pins: CS output CS3	CS2	CS1	Address output A19	A18	A17	A16	A15–A0	
0	0	400000₁₆	0	0	0	0	1	0	0	0000₁₆	000000₁₆
		7FFFF₁₆	0	0	0	0	1	1	1	FFFF₁₆	03FFFF₁₆
	1	400000₁₆	0	0	0	1	0	0	0	0000₁₆	040000₁₆
		7FFFF₁₆	0	0	0	1	0	1	1	FFFF₁₆	07FFFF₁₆
1	0	400000₁₆	0	0	1	0	1	0	0	0000₁₆	080000₁₆
		7FFFF₁₆	0	0	1	0	1	1	1	FFFF₁₆	0BFFFF₁₆
	1	400000₁₆	0	0	1	1	0	0	0	0000₁₆	0C0000₁₆
		7FFFF₁₆	0	0	1	1	0	1	1	FFFF₁₆	0FFFFF₁₆
2	0	400000₁₆	0	1	0	0	1	0	0	0000₁₆	100000₁₆
		7FFFF₁₆	0	1	0	0	1	1	1	FFFF₁₆	13FFFF₁₆
	1	400000₁₆	0	1	0	1	0	0	0	0000₁₆	140000₁₆
		7FFFF₁₆	0	1	0	1	0	1	1	FFFF₁₆	17FFFF₁₆
3	0	400000₁₆	0	1	1	0	1	0	0	0000₁₆	180000₁₆
		7FFFF₁₆	0	1	1	0	1	1	1	FFFF₁₆	1BFFFF₁₆
	1	400000₁₆	0	1	1	1	0	0	0	0000₁₆	1C0000₁₆
		7FFFF₁₆	0	1	1	1	0	1	1	FFFF₁₆	1FFFFF₁₆
4	0	400000₁₆	1	0	0	0	1	0	0	0000₁₆	200000₁₆
		7FFFF₁₆	1	0	0	0	1	1	1	FFFF₁₆	23FFFF₁₆
	1	400000₁₆	1	0	0	1	0	0	0	0000₁₆	240000₁₆
		7FFFF₁₆	1	0	0	1	0	1	1	FFFF₁₆	27FFFF₁₆
5	0	400000₁₆	1	0	1	0	1	0	0	0000₁₆	280000₁₆
		7FFFF₁₆	1	0	1	0	1	1	1	FFFF₁₆	2BFFFF₁₆
	1	400000₁₆	1	0	1	1	0	0	0	0000₁₆	2C0000₁₆
		7FFFF₁₆	1	0	1	1	0	1	1	FFFF₁₆	2FFFFF₁₆
6	0	400000₁₆	1	1	0	0	1	0	0	0000₁₆	300000₁₆
		7FFFF₁₆	1	1	0	0	1	1	1	FFFF₁₆	33FFFF₁₆
	1	400000₁₆	1	1	0	1	0	0	0	0000₁₆	340000₁₆
		7FFFF₁₆	1	1	0	1	0	1	1	FFFF₁₆	37FFFF₁₆
7	0	400000₁₆	1	1	1	0	1	0	0	0000₁₆	380000₁₆
		7FFFF₁₆	1	1	1	0	1	1	1	FFFF₁₆	3BFFFF₁₆
		800000₁₆									Internal ROM access
		FFFFF₁₆									Internal ROM access
	1	400000₁₆	1	1	1	1	0	0	0	0000₁₆	3C0000₁₆
		7FFFF₁₆	1	1	1	1	0	1	1	FFFF₁₆	3FFFFF₁₆
		800000₁₆									Internal ROM access
		FFFFF₁₆									Internal ROM access
			A21	A20	A19	A18	N.C.	A17	A16	A15–A0	Address input for 4-Mbyte ROM
			Address input for 4-Mbyte ROM								

N.C.: No connected

Figure 1.8.8. Relationship Between Addresses on 4-M Byte ROM and Those on Microcomputer (2)

Microprocessor mode



Bank number	OFS	Access area	Output from the microcomputer pins: CS output CS3	CS2	CS1	Address output A19	A18	A17	A16	A15–A0	
0	0	400000₁₆	0	0	0	0	1	0	0	0000₁₆	000000₁₆
		BFFFF₁₆	0	0	0	1	0	1	1	FFFF₁₆	07FFFF₁₆
	1	400000₁₆	0	0	0	1	0	0	0	0000₁₆	040000₁₆
		BFFFF₁₆	0	0	1	0	1	1	1	FFFF₁₆	0BFFFF₁₆
1	0	400000₁₆	0	0	1	0	1	0	0	0000₁₆	080000₁₆
		BFFFF₁₆	0	0	1	1	0	1	1	FFFF₁₆	0FFFFF₁₆
	1	400000₁₆	0	0	1	1	0	0	0	0000₁₆	0C0000₁₆
		BFFFF₁₆	0	1	0	0	1	1	1	FFFF₁₆	13FFFF₁₆
2	0	400000₁₆	0	1	0	0	1	0	0	0000₁₆	100000₁₆
		BFFFF₁₆	0	1	0	1	0	1	1	FFFF₁₆	17FFFF₁₆
	1	400000₁₆	0	1	0	1	0	0	0	0000₁₆	140000₁₆
		BFFFF₁₆	0	1	1	0	1	1	1	FFFF₁₆	1BFFFF₁₆
3	0	400000₁₆	0	1	1	0	1	0	0	0000₁₆	180000₁₆
		BFFFF₁₆	0	1	1	1	0	1	1	FFFF₁₆	1FFFFF₁₆
	1	400000₁₆	0	1	1	1	0	0	0	0000₁₆	1C0000₁₆
		BFFFF₁₆	1	0	0	0	1	1	1	FFFF₁₆	23FFFF₁₆
4	0	400000₁₆	1	0	0	0	1	0	0	0000₁₆	200000₁₆
		BFFFF₁₆	1	0	0	1	0	1	1	FFFF₁₆	27FFFF₁₆
	1	400000₁₆	1	0	0	1	0	0	0	0000₁₆	240000₁₆
		BFFFF₁₆	1	0	1	0	1	1	1	FFFF₁₆	2BFFFF₁₆
5	0	400000₁₆	1	0	1	0	1	0	0	0000₁₆	280000₁₆
		BFFFF₁₆	1	0	1	1	0	1	1	FFFF₁₆	2FFFFF₁₆
	1	400000₁₆	1	0	1	1	0	0	0	0000₁₆	2C0000₁₆
		BFFFF₁₆	1	1	0	0	1	1	1	FFFF₁₆	33FFFF₁₆
6	0	400000₁₆	1	1	0	0	1	0	0	0000₁₆	300000₁₆
		BFFFF₁₆	1	1	0	1	0	1	1	FFFF₁₆	37FFFF₁₆
	1	400000₁₆	1	1	0	1	0	0	0	0000₁₆	340000₁₆
		BFFFF₁₆	1	1	1	0	1	1	1	FFFF₁₆	3BFFFF₁₆
7	0	400000₁₆	1	1	1	0	1	0	0	0000₁₆	380000₁₆
		7FFFF₁₆	1	1	1	0	1	1	1	FFFF₁₆	3BFFFF₁₆
		800000₁₆	1	1	1	1	0	0	0	0000₁₆	3C0000₁₆
		BFFFF₁₆	1	1	1	1	0	1	1	FFFF₁₆	3FFFFF₁₆
		C0000₁₆	1	1	1	1	1	0	0	0000₁₆	3C0000₁₆
		FFFFF₁₆	1	1	1	1	1	1	1	FFFF₁₆	3FFFFF₁₆
			A21	A20	A19	A18	N.C.	A17	A16	A15–A0	Address input for 4-Mbyte ROM
			Address input for 4-Mbyte ROM								

N.C.: No connected

Figure 1.8.9. Relationship Between Addresses on 4-M Byte ROM and Those on Microcomputer (3)

Clock Generation Circuit

The clock generation circuit contains four oscillator circuits as follows:

(1) Main clock oscillation circuit
(2) Sub clock oscillation circuit
(3) Ring oscillator (oscillation stop detect function)
(4) PLL frequency synthesizer

Table 1.9.1 lists the clock generation circuit specifications. Figure 1.9.1 shows the clock generation circuit. Figures 1.9.2 to 1.9.6 show the clock-related registers.

Table 1.9.1. Clock Generation Circuit Specifications

Item	Main clock oscillation circuit	Sub clock oscillation circuit	Ring oscillator	PLL frequency synthesizer
Use of clock	• CPU clock source • Peripheral function clock source	•CPU clock source • Timer A, B's clock source	• CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when the main clock stops oscillating	• CPU clock source • Peripheral function clock source
Clock frequency	0 to 16 MHz	32.768 kHz	About 1 MHz	10 to 24 MHz
Usable oscillator	• Ceramic oscillator • Crystal oscillator	• Crystal oscillator	_____	_____
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT	_____	_____
Oscillation stop, restart function	Presence	Presence	Presence	Presence
Oscillator status after reset	Oscillating	Stopped	Stopped	Stopped
Other	Externally derived clock can be input		_____	_____

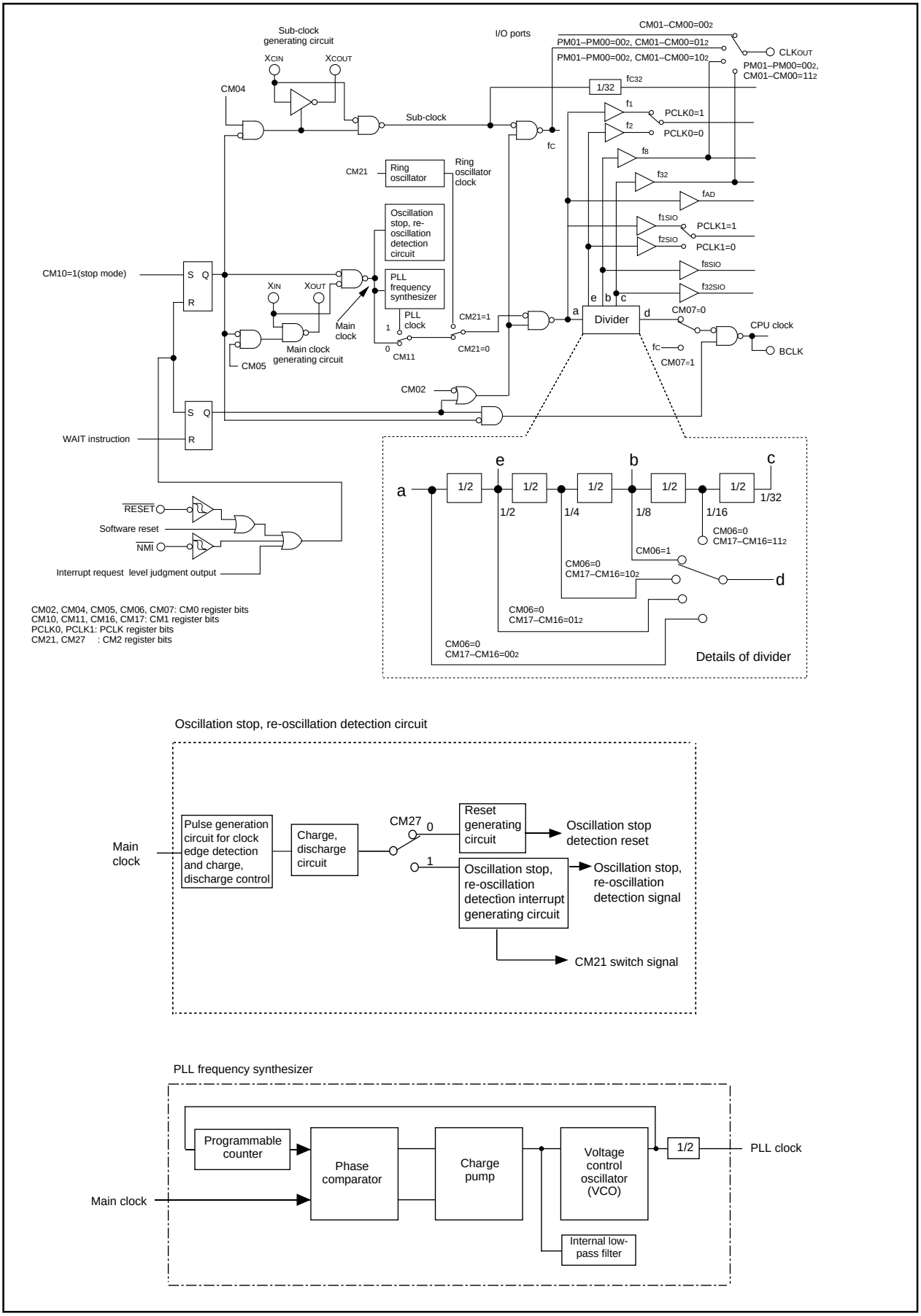
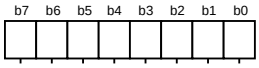


Figure 1.9.1. Clock Generation Circuit

System clock control register 0 (Note 1)



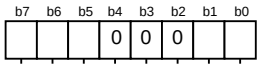
Symbol	Address	After reset
CM0	0006_{16}	01001000_2

Bit symbol	Bit name	Function	RW
CM00	Clock output function select bit (Valid only in single-chip mode)	b1 b0 0 0 : I/O port $P5_7$ 0 1 : f_C output 1 0 : f_8 output 1 1 : f_{32} output	RW
CM01			RW
CM02	WAIT peripheral function clock stop bit (Note 10)	0 : Do not stop peripheral function clock in wait mode 1 : Stop peripheral function clock in wait mode (Note 8)	RW
CM03	X_{CIN}-X_{COUT} drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	RW
CM04	Port X_C select bit (Note 2)	0 : I/O port $P8_6$, $P8_7$ 1 : X_{CIN}-X_{COUT} generation function(Note 9)	RW
CM05	Main clock stop bit (Notes 3, 10, 12, 13)	0 : On 1 : Off (Note 4, Note5)	RW
CM06	Main clock division select bit 0 (Notes 7, 13)	0 : CM16 and CM17 valid 1 : Division by 8 mode	RW
CM07	System clock select bit (Notes 6, 10, 11, 12)	0 : Main clock, PLL clock, or ring oscillator clock 1 : Sub-clock	RW

Note 1: Write to this register after setting the PRC0 bit of PRCR register to “1” (write enable).
Note 2: The CM03 bit is set to “1” (high) when the CM04 bit is set to “0” (I/O port) or the microcomputer goes to a stop mode.
Note 3: This bit is provided to stop the main clock when the low power dissipation mode or ring oscillator low power dissipation mode is selected. This bit cannot be used for detection as to whether the main clock stopped or not. To stop the main clock, the following setting is required:
(1) Set the CM07 bit to “1” (Sub-clock select) or the CM21 bit of CM2 register to “1” (Ring oscillator select) with the sub-clock stably oscillating.
(2) Set the CM20 bit of CM2 register to “0” (Oscillation stop, re-oscillation detection function disabled).
(3) Set the CM05 bit to “1” (Stop).
Note 4: During external clock input, only the clock oscillation buffer is turned off and clock input is accepted.
Note 5: When CM05 bit is set to “1, the X_{OUT} pin goes “H”. Furthermore, because the internal feedback resistor remains connected, the X_{IN} pin is pulled “H” to the same level as X_{OUT} via the feedback resistor.
Note 6: After setting the CM04 bit to “1” (X_{CIN}-X_{COUT} oscillator function), wait until the sub-clock oscillates stably before switching the CM07 bit from “0” to “1” (sub-clock).
Note 7: When entering stop mode from high or middle speed mode, ring oscillator mode or ring oscillator low power mode, the CM06 bit is set to “1” (divide-by-8 mode).
Note 8: The f_{C32} clock does not stop. During low speed or low power dissipation mode, do not set this bit to “1” (peripheral clock turned off when in wait mode).
Note 9: To use a sub-clock, set this bit to “1”. Also make sure ports $P8_6$ and $P8_7$ are directed for input, with no pull-ups.
Note 10: When the PM21 bit of PM2 register is set to “1” (clock modification disable), writing to the CM02, CM05, and CM07 bits has no effect.
Note 11: If the PM21 bit needs to be set to “1”, set the CM07 bit to “0”(main clock) before setting it.
Note 12: To use the main clock as the clock source for the CPU clock, follow the procedure below.
(1) Set the CM05 bit to “0” (oscillate).
(2) Wait until td(M-L) elapses or the main clock oscillation stabilizes, whichever is longer.
(3) Set the CM11, CM21 and CM07 bits all to “0”.
Note 13: If the CM05 bit is set to “1” (main clock turned off) in low speed mode, the CM06 bit is set to “1” (divide-by-8 mode) and the CM15 bit is set to “1” (drive capability high). Avoid changing the CM06 bit in low power dissipation mode.
During ring oscillator mode, the CM06 and CM15 bits do not change even if the CM05 bit is set to “1”. During ring oscillator low power dissipation mode, the divide-by-n value can be selected using the CM06 and CM17 to CM16 bits. To return to high or middle speed mode, however, set the CM06 bit to “1” and the CM15 bit to “1” before selecting the desired mode.

Figure 1.9.2. CM0 Register

System clock control register 1 (Note 1)



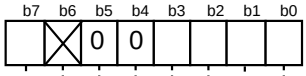
Symbol: CM1 — Address: 0007_{16} — After reset: 00100000_2

Bit symbol	Bit	Function	RW
CM10	All clock stop control bit (Notes 4, 6)	0 : Clock on 1 : All clocks off (stop mode)	RW
CM11	System clock select bit 1 (Notes 6, 7)	0 : Main clock 1 : PLL clock (Note 5)	RW
$\overline{\text{(b4-b2)}}$	Reserved bit	Must set to "0"	RW
CM15	XIN-XOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	RW
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	RW
CM17			RW

Note 1: Write to this register after setting the PRC0 bit of PRCR register to "1" (write enable).
Note 2: When entering stop mode from high or middle speed mode, or when the CM05 bit is set to "1" (main clock turned off) in low speed mode, the CM15 bit is set to "1" (drive capability high).
Note 3: Effective when the CM06 bit is "0" (CM16 and CM17 bits enable).
Note 4: If the CM10 bit is "1" (stop mode), XOUT goes "H" and the internal feedback resistor is disconnected. The XCIN and XCOUT pins are placed in the high-impedance state. When the CM11 bit is set to "1" (PLL clock), or the CM20 bit of CM2 register is set to "1" (oscillation stop, re-oscillation detection function enabled), do not set the CM10 bit to "1".
Note 5: After setting the PLC07 bit in PLC0 register to "1" (PLL operation), wait until Tsu (PLL) elapses before setting the CM11 bit to "1" (PLL clock).
Note 6: When the PM21 bit of PM2 register is set to "1" (clock modification disable), writing to the CM10, CM011 bits has no effect. When the PM22 bit of PM2 register is set to "1" (watchdog timer count source is ring oscillator clock), writing to the CM10 bit has no effect.
Note 7: Effective when CM07 bit is "0" and CM21 bit is "0" .

Figure 1.9.3. CM1 Register

Oscillation stop detection register (Note 1)



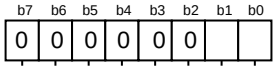
Symbol: CM2 | Address: 000C16 | After reset: 0X0000002(Note 11)

Bit symbol	Bit name	Function	RW
CM20	Oscillation stop, re-oscillation detection bit (Notes 7, 9, 10, 11)	0: Oscillation stop, re-oscillation detection function disabled 1: Oscillation stop, re-oscillation detection function enabled	RW
CM21	System clock select bit 2 (Notes 2, 3, 6, 8, 11)	0: Main clock or PLL clock (Ring oscillator turned off) 1: Ring oscillator clock (Ring oscillator oscillating)	RW
CM22	Oscillation stop, re-oscillation detection flag (Note 4)	0: Main clock stop, re-oscillation not detected 1: Main clock stop, re-oscillation detected	RW
CM23	XIN monitor flag (Note 5)	0: Main clock oscillating 1: Main clock turned off	RO
— (b5-b4)	Reserved bit	Must set to “0”	RW
— (b6)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
CM27	Operation select bit (when an oscillation stop, re-oscillation is detected) (Note 11)	0: Oscillation stop detection reset 1: Oscillation stop, re-oscillation detection interrupt	RW

Note 1: Write to this register after setting the PRC0 bit of PRCR register to “1” (write enable).
Note 2: When the CM20 bit is “1” (oscillation stop, re-oscillation detection function enabled), the CM27 bit is “1” (oscillation stop, re-oscillation detection interrupt), and the CPU clock source is the main clock, the CM21 bit is set to “1” (ring oscillator clock) if the main clock stop is detected.
Note 3: If the CM20 bit is “1” and the CM23 bit is “1” (main clock turned off), do not set the CM21 bit to “0”.
Note 4: This bit becomes “1” at main clock stop detection and main clock re-oscillation detection. When this bit changes from “0” to “1”, there arise oscillation stop, re-oscillation detection interrupt. Use this register to discriminate the causes for oscillation stop, re-oscillation detection interrupt and watchdog timer interrupt in the interrupt processing program. By writing “0” in the program, this bit becomes “0”. (Even when “1” is written in the program, no change is identified for the bit. Also, this bit is not set to “0” where there occur oscillation stop, re-oscillation detection interrupt.) When the CM22 bit is “1”, no oscillation stop, re-oscillation detection interrupt occur even if oscillation stop or re-oscillation is detected.
Note 5: Read the CM23 bit in an oscillation stop, re-oscillation detection interrupt handling routine to determine the main clock status.
Note 6: Effective when the CM07 bit of CM0 register is “0”.
Note 7: When the PM21 bit of PM2 register is “1” (clock modification disabled), writing to the CM20 bit has no effect.
Note 8: Where the CM20 bit is “1” (oscillation stop, re-oscillation detection function enabled), the CM27 bit is “1” (oscillation stop, re-oscillation detection interrupt), and the CM11 bit is “0” (the CPU clock source is PLL clock), the CM21 bit remains unchanged even when main clock stop is detected. If the CM22 bit is “0” under these conditions, oscillation stop, re-oscillation detection interrupt occur at main clock stop detection; it is, therefore, necessary to set the CM21 bit to “1” (ring oscillator clock) inside the interrupt routine.
Note 9: Set the CM20 bit to “0” (disable) before entering stop mode. After exiting stop mode, set the CM20 bit back to “1” (enable).
Note 10: Set the CM20 bit to “0” (disable) before setting the CM05 bit of CM0 register.
Note 11: The CM20, CM21 and CM27 bits do not change at oscillation stop detection reset.

Figure 1.9.4. CM2 Register

Peripheral clock select register (Note)

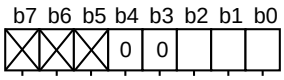


Symbol: PCLKR
Address: $025E_{16}$
When reset: 00000011_2

Bit symbol	Bit name	Function	RW
PCLK0	Timers A, B clock select bit (Clock source for the timers A, B, and the dead time timer)	0 : f_2 1 : f_1	RW
PCLK1	SI/O clock select bit (Clock source for UART0 to UART2, SI/O3, SI/O4)	0 : f_{2SIO} 1 : f_{1SIO}	RW
—— (b7-b2)	Reserved bit	Must set to “0”	RW

Note: Write to this register after setting the PRC0 bit of PRCR register to “1” (write enable).

Processor mode register 2 (Note 1)



Symbol: PM2
Address: $001E_{16}$
After reset: $XXX00000_2$

Bit symbol	Bit name	Function	RW
PM20	Specifying wait when accessing SFR at PLL operation (Note 2)	0 : 2 waits 1 : 1 wait	RW
PM21	System clock protective bit (Note 3, Note 4)	0 : Clock is protected by PRCR register 1 : Clock modification disabled	RW
PM22	WDT count source protective bit (Note 3, Note 5)	0 : CPU clock is used for the watchdog timer count source 1 : Ring oscillator clock is used for the watchdog timer count source	RW
— (b4-b3)	Reserved bit	Must set to “0”	RW
— (b7-b5)	Nothing is assigned. When write, set to “0”. When read, its content is interdeterminate.		—

Note 1: Write to this register after setting the PRC1 bit of PRCR register to “1” (write enable).
Note 2: This bit can only be rewritten while the PLC07 bit is “0” (PLL turned off). Also, to select a 16 MHz or higher PLL clock, set this bit to “0” (2 waits). Note that if the clock source for the CPU clock is to be changed from the PLL clock to another, the PLC07 bit must be set to “0” before setting the PM20 bit.
Note 3: Once this bit is set to “1”, it cannot be cleared to “0” in a program.
Note 4: Setting the PM21 bit to “1” results in the following conditions:
- The BCLK is not halted by executing the WAIT instruction.
- Writing to the following bits has no effect.
  - CM02 bit of CM0 register
  - CM05 bit of CM0 register (main clock is not halted)
  - CM07 bit of CM0 register (CPU clock source does not change)
  - CM10 bit of CM1 register (stop mode is not entered)
  - CM11 bit of CM1 register (CPU clock source does not change)
  - CM20 bit of CM2 register (oscillation stop, re-oscillation detection function settings do not change)
  - All bits of PLC0 register (PLL frequency synthesizer settings do not change)

Note 5: Setting the PM22 bit to “1” results in the following conditions:
- The ring oscillator starts oscillating, and the ring oscillator clock becomes the watchdog timer count source.
- The CM10 bit of CM1 register is disabled against write. (Writing a “1” has no effect, nor is stop mode entered.)
- The watchdog timer does not stop when in wait mode or hold state.

Figure 1.9.5. PCLKR Register and PM2 Register

PLL control register 0 (Note 1, Note 2)

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

Symbol	Address	After reset
PLC0	$001C_{16}$	$0001\ X010_2$

Bit symbol	Bit name	Function	RW
PLC00	PLL multiplying factor select bit (Note 3)	b2 b1 b0 0 0 0: Do not set 0 0 1: Multiply by 2 0 1 0: Multiply by 4 0 1 1: Multiply by 6 1 0 0: Multiply by 8 1 0 1: 1 1 0: 1 1 1: } Do not set	RW
PLC01			RW
PLC02			RW
—— (b3)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
—— (b4)	Reserved bit	Must set to “1”	RW
—— (b6-b5)	Reserved bit	Must set to “0”	RW
PLC07	Operation enable bit (Note 4)	0: PLL Off 1: PLL On	RW

Note 1: Write to this register after setting the PRC0 bit of PRCR register to “1” (write enable).
Note 2: When the PM21 bit of PM2 register is “1” (clock modification disable), writing to this register has no effect.
Note 3: These three bits can only be modified when the PLC07 bit = 0 (PLL turned off). The value once written to this bit cannot be modified.
Note 4: Before setting this bit to “1” , set the CM07 bit to “0” (main clock), set the CM17 to CM16 bits to “00_2” (main clock undivided mode), and set the CM06 bit to “0” (CM16 and CM17 bits enable).

Figure 1.9.6. PLC0 Register

The following describes the clocks generated by the clock generation circuit.

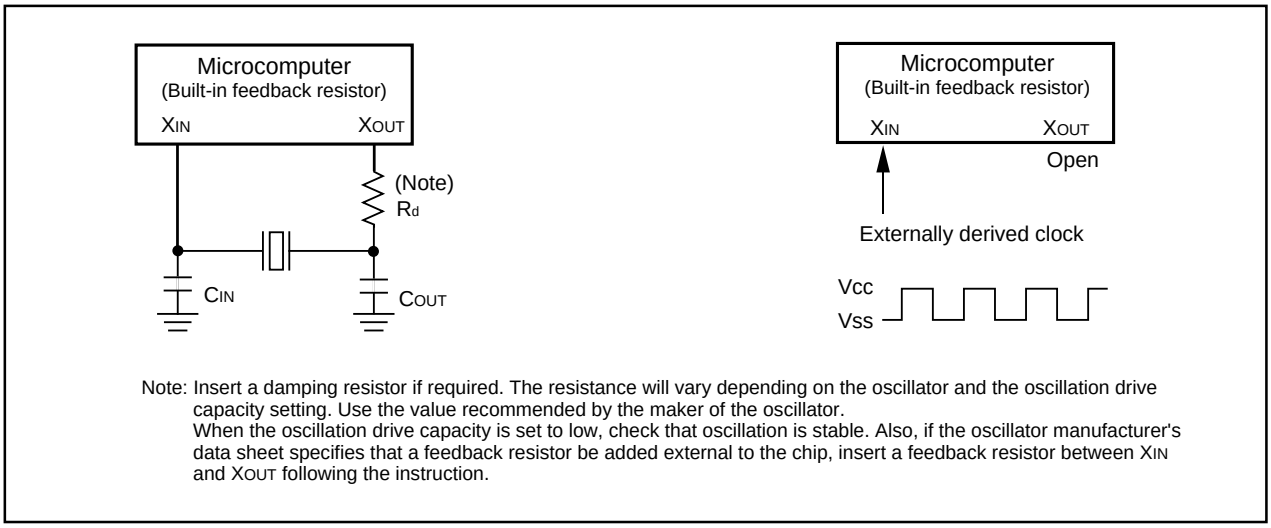
(1) Main Clock

This clock is used as the clock source for the CPU and peripheral function clocks. This clock is used as the clock source for the CPU and peripheral function clocks. The main clock oscillator circuit is configured by connecting a resonator between the XIN and XOUT pins. The main clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The main clock oscillator circuit may also be configured by feeding an externally generated clock to the XIN pin. Figure 1.9.7 shows the examples of main clock connection circuit.

After reset, the main clock divided by 8 is selected for the CPU clock.

The power consumption in the chip can be reduced by setting the CM05 bit of CM0 register to “1” (main clock oscillator circuit turned off) after switching the clock source for the CPU clock to a sub clock or ring oscillator clock. In this case, XOUT goes “H”. Furthermore, because the internal feedback resistor remains on, XIN is pulled “H” to XOUT via the feedback resistor. Note that if an externally generated clock is fed into the XIN pin, the main clock cannot be turned off by setting the CM05 bit to “1”. If necessary, use an external circuit to turn off the clock.

During stop mode, all clocks including the main clock are turned off. Refer to “power control”.



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 1.9.7. Examples of Main Clock Connection Circuit

(2) Sub Clock

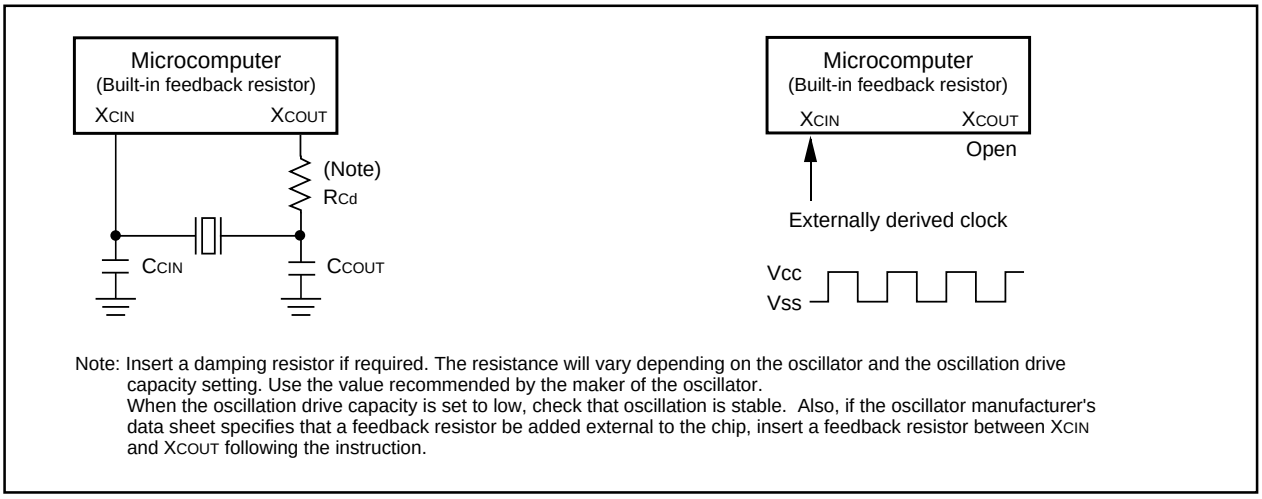
The sub clock is generated by the sub clock oscillation circuit. This clock is used as the clock source for the CPU clock, as well as the timer A and timer B count sources. In addition, an fc clock with the same frequency as that of the sub clock can be output from the CLKOUT pin.

The sub clock oscillator circuit is configured by connecting a crystal resonator between the XCIN and XCOUT pins. The sub clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The sub clock oscillator circuit may also be configured by feeding an externally generated clock to the XCIN pin. Figure 1.9.8 shows the examples of sub clock connection circuit.

After reset, the sub clock is turned off. At this time, the feedback resistor is disconnected from the oscillator circuit.

To use the sub clock for the CPU clock, set the CM07 bit of CM0 register to “1 ” (sub clock) after the sub clock becomes oscillating stably.

During stop mode, all clocks including the sub clock are turned off. Refer to “power control”.



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 1.9.8. Examples of Sub Clock Connection Circuit

(3) Ring Oscillator Clock

This clock, approximately 1 MHz, is supplied by a ring oscillator. This clock is used as the clock source for the CPU and peripheral function clocks. In addition, if the PM22 bit of PM2 register is “1” (ring oscillator clock for the watchdog timer count source), this clock is used as the count source for the watchdog timer. After reset, the ring oscillator clock is turned off. It is turned on by setting the CM21 bit of CM2 register to “1” (ring oscillator clock), and is used as the clock source for the CPU and peripheral function clocks, in place of the main clock. If the main clock stops oscillating when the CM20 bit of CM2 register is “1” (oscillation stop, re-oscillation detection function enabled) and the CM27 bit is “1” (oscillation stop, re-oscillation detection interrupt), the ring oscillator automatically starts operating, supplying the necessary clock for the microcomputer.

(4) PLL Clock

The PLL clock is generated from the main clock by a PLL frequency synthesizer. This clock is used as the clock source for the CPU and peripheral function clocks. After reset, the PLL clock is turned off. The PLL frequency synthesizer is activated by setting the PLC07 bit to “1” (PLL operation). When the PLL clock is used as the clock source for the CPU clock, wait t_{su}(PLL) for the PLL clock to be stable, and then set the CM11 bit in the CM1 register to “1”.

To enter wait or stop mode, set the CM11 bit to “0” (main clock for the CPU clock source) and then the PLC07 bit of PLC0 register to “0” (PLL off) before entering that mode. Figure 1.9.9 shows the procedure for using the PLL clock as the clock source for the CPU.

The PLL clock frequency is determined by the equation below.

PLL clock frequency=f(X_{IN}) X (multiplying factor set by the PLC02 to PLC00 bits PLC0 register

(However, 10 MHz ≤ PLL clock frequency ≤ 24 MHz)

The PLC02 to PLC00 bits can be set only once after reset. Table 1.9.2 shows the example for setting PLL clock frequencies.

Table 1.9.2. Example for Setting PLL Clock Frequencies

X_{IN} (MHz)	PLC02	PLC01	PLC00	Multiplying factor	PLL clock (MHz)(Note)
10	0	0	1	2	20
5	0	1	0	4	
3.33	0	1	1	6	
2.5	1	0	0	8	
12	0	0	1	2	24
6	0	1	0	4	
4	0	1	1	6	
3	1	0	0	8	

Note: 10MHz≤PLL clock frequency≤24MHz.

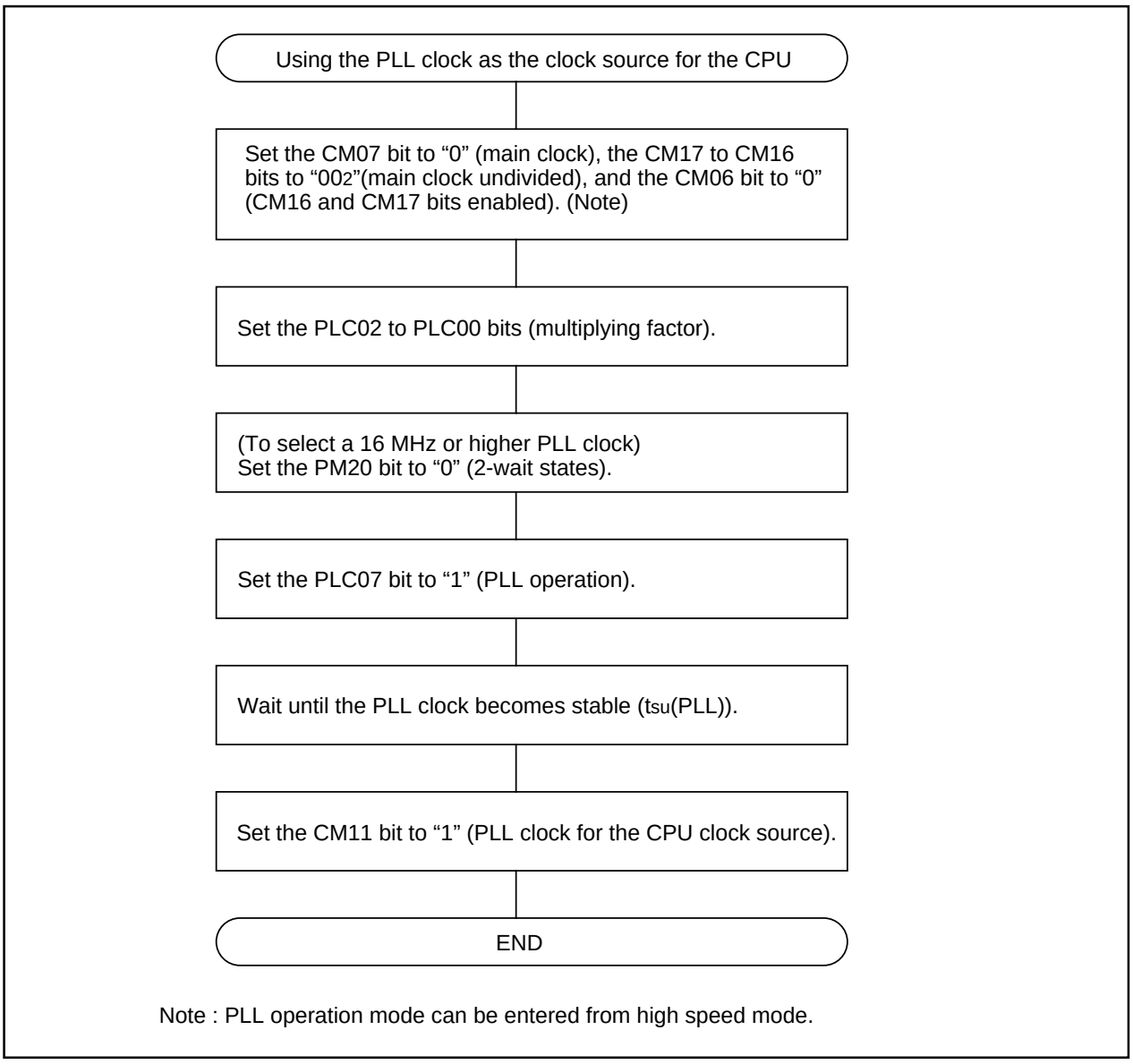


Figure 1.9.9. Procedure to Use PLL Clock as CPU Clock Source

CPU Clock and Peripheral Function Clock

Two type clocks: CPU clock to operate the CPU and peripheral function clocks to operate the peripheral functions.

(1) CPU Clock and BCLK

These are operating clocks for the CPU and watchdog timer.

The clock source for the CPU clock can be chosen to be the main clock, sub clock, ring oscillator clock or the PLL clock.

If the main clock or ring oscillator clock is selected as the clock source for the CPU clock, the selected clock source can be divided by 1 (undivided), 2, 4, 8 or 16 to produce the CPU clock. Use the CM06 bit in CM0 register and the CM17 to CM16 bits in CM1 register to select the divide-by-n value.

When the PLL clock is selected as the clock source for the CPU clock, the CM06 bit should be set to "0" and the CM17 to CM16 bits to "00_2" (undivided).

After reset, the main clock divided by 8 provides the CPU clock.

During memory expansion or microprocessor mode, a BCLK signal with the same frequency as the CPU clock can be output from the BCLK pin by setting the PM07 bit of PM0 register to "0" (output enabled).

Note that when entering stop mode from high or middle speed mode, ring oscillator mode or low power ring oscillator mode, or when the CM05 bit of CM0 register is set to "1" (main clock turned off) in low-speed mode, the CM06 bit of CM0 register is set to "1" (divide-by-8 mode).

(2) Peripheral Function Clock(f_1, f_2, f_8, f_{32}, f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}, f_{AD}, f_{C32})

These are operating clocks for the peripheral functions.

Of these, fi (i = 1, 2, 8, 32) and f_{iSIO} are derived from the main clock, PLL clock or ring oscillator clock by dividing them by i. The clock fi is used for timers A and B, and f_{iSIO} is used for serial I/O. The f_8 and f_{32} clocks can be output from the CLK_{OUT} pin.

The f_{AD} clock is produced from the main clock, PLL clock or ring oscillator clock, and is used for the A-D converter.

When the WAIT instruction is executed after setting the CM02 bit of CM0 register to "1" (peripheral function clock turned off during wait mode), or when the microcomputer is in low power dissipation mode, the fi, f_{iSIO} and f_{AD} clocks are turned off.

The f_{C32} clock is produced from the sub clock, and is used for timers A and B. This clock can be used when the sub clock is on.

Clock Output Function

During single-chip mode, the f_8, f_{32} or f_C clock can be output from the CLK_{OUT} pin. Use the CM01 to CM00 bits of CM0 register to select.

Power Control

There are three power control modes. For convenience' sake, all modes other than wait and stop modes are referred to as normal operation mode here.

(1) Normal Operation Mode

Normal operation mode is further classified into seven modes.

In normal operation mode, because the CPU clock and the peripheral function clocks both are on, the CPU and the peripheral functions are operating. Power control is exercised by controlling the CPU clock frequency. The higher the CPU clock frequency, the greater the processing capability. The lower the CPU clock frequency, the smaller the power consumption in the chip. If the unnecessary oscillator circuits are turned off, the power consumption is further reduced.

Before the clock sources for the CPU clock can be switched over, the new clock source to which switched must be oscillating stably. If the new clock source is the main clock, sub clock or PLL clock, allow a sufficient wait time in a program until it becomes oscillating stably.

Note that operation modes cannot be changed directly from low speed or low power dissipation mode to ring oscillator or ring oscillator low power dissipation mode. Nor can operation modes be changed directly from ring oscillator or ring oscillator low power dissipation mode to low speed or low power dissipation mode. Where the CPU clock source is changed from the ring oscillator to the main clock, change the operation mode to the medium speed mode (divided by 8 mode) after the clock was divided by 8 (the CM06 bit of CM0 register was set to "1") in the ring oscillator mode.

- **High-speed Mode**

  The main clock divided by 1 provides the CPU clock. If the sub clock is on, f_{C32} can be used as the count source for timers A and B.

- **PLL Operation Mode**

  The main clock multiplied by 2, 4, 6 or 8 provides the PLL clock, and this PLL clock serves as the CPU clock. If the sub clock is on, f_{C32} can be used as the count source for timers A and B. PLL operation mode can be entered from high speed mode. If PLL operation mode is to be changed to wait or stop mode, first go to high speed mode before changing.

- **Medium-speed Mode**

  The main clock divided by 2, 4, 8 or 16 provides the CPU clock. If the sub clock is on, f_{C32} can be used as the count source for timers A and B.

- **Low-speed Mode**

  The sub clock provides the CPU clock. The main clock is used as the clock source for the peripheral function clock when the CM21 bit is set to "0" (ring oscillator turned off), and the ring oscillator clock is used when the CM21 bit is set to "1" (ring oscillator oscillating).

  The f_{C32} clock can be used as the count source for timers A and B.

- **Low Power Dissipation Mode**

  In this mode, the main clock is turned off after being placed in low speed mode. The sub clock provides the CPU clock. The f_{C32} clock can be used as the count source for timers A and B. f_{C32} is the only peripheral function clock available when the CM21 bit is set to "0" (ring oscillator turned off). If the CM21 bit is set to "1" (ring oscillator oscillating), then f_{C32} and the ring oscillator clock can be used.

  Simultaneously when this mode is selected, the CM06 bit of CM0 register becomes "1" (divided by 8 mode). In the low power dissipation mode, do not change the CM06 bit. Consequently, the medium speed (divided by 8) mode is to be selected when the main clock is operated next.

- **Ring Oscillator Mode**

  The ring oscillator clock divided by 1 (undivided), 2, 4, 8 or 16 provides the CPU clock. The ring oscillator clock is also the clock source for the peripheral function clocks. If the sub clock is on, f_{C32} can be used as the count source for timers A and B.

- **Ring Oscillator Low Power Dissipation Mode**

  The main clock is turned off after being placed in ring oscillator mode. The CPU clock can be selected as in the ring oscillator mode. The ring oscillator clock is the clock source for the peripheral function clocks. If the sub clock is on, f_{C32} can be used as the count source for timers A and B. When the operation mode is returned to the high and medium speed modes, set the CM06 bit to “1” (divided by 8 mode).

Table 1.9.3. Setting Clock Related Bit and Modes

Modes		CM2 register	CM1 register		CM0 register			
		CM21	CM11	CM17, CM16	CM07	CM06	CM05	CM04
PLL operation mode		0	1	00_2	0	0	0	—
High-speed mode		0	0	00_2	0	0	0	—
Midium-speed mode	divided by 2	0	0	01_2	0	0	0	—
	divided by 4	0	0	10_2	0	0	0	—
	divided by 8	0	0	—	0	1	0	—
	divided by 16	0	0	11_2	0	0	0	—
Low-speed mode		—	—	—	1	—	0	1
Low power dissipation mode		—	—	—	1	1(Note 1)	1(Note 1)	1
Ring oscillator mode	divided by 1	1	—	00_2	0	0	0	—
	divided by 2	1	—	01_2	0	0	0	—
	divided by 4	1	—	10_2	0	0	0	—
	divided by 8	1	—	—	0	1	0	—
	divided by 16	1	—	11_2	0	0	0	—
Ring oscillator low power dissipation mode		1	—	(Note 2)	0	(Note 2)	1	—

Note 1: When the CM05 bit is set to “1” (main clock turned off) in low-speed mode, the mode goes to low power dissipation mode and CM06 bit is set to “1” (divided by 8 mode) simultaneously.
Note 2: The divide-by-n value can be selected the same way as in ring oscillator mode.

(2) Wait Mode

In wait mode, the CPU clock is turned off, so are the CPU (because operated by the CPU clock) and the watchdog timer. However, if the PM22 bit of PM2 register is “1” (ring oscillator clock for the watchdog timer count source), the watchdog timer remains active. Because the main clock, sub clock, ring oscillator clock and PLL clock all are on, the peripheral functions using these clocks keep operating.

- **Peripheral Function Clock Stop Function**

  If the CM02 bit is “1” (peripheral function clocks turned off during wait mode), the f_1, f_2, f_8, f_{32}, f_{1SIO}, f_{8SIO}, f_{32SIO} and f_{AD} clocks are turned off when in wait mode, with the power consumption reduced that much. However, f_{C32} remains on.

- **Entering Wait Mode**

  The microcomputer is placed into wait mode by executing the WAIT instruction.

  If the CM11 bit is “1” (PLL clock for the CPU clock source), set the CM11 bit to “0” (main clock for the CPU clock source) and then the PLC07 bit to “0” (PLL turned off) before entering wait mode.

- **Pin Status During Wait Mode**

  Table 1.9.4 lists pin status during wait mode

- **Exiting Wait Mode**

  The microcomputer is moved out of wait mode by a hardware reset, $\overline{\text{NMI}}$ interrupt or peripheral function interrupt.

  If the microcomputer is to be moved out of exit wait mode by a hardware reset or $\overline{\text{NMI}}$ interrupt, set the peripheral function interrupt priority ILVL2 to ILVL0 bits to “000_2” (interrupts disabled) before executing the WAIT instruction.

  The peripheral function interrupts are affected by the CM02 bit. If CM02 bit is “0” (peripheral function clocks not turned off during wait mode), all peripheral function interrupts can be used to exit wait

mode. If CM02 bit is “1” (peripheral function clocks turned off during wait mode), the peripheral functions using the peripheral function clocks stop operating, so that only the peripheral functions clocked by external signals can be used to exit wait mode.

Table 1.9.4. Pin Status During Wait Mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ 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”	
I/O ports		Retains status before wait mode	Retains status before wait mode
CLK_{OUT}	When f_C selected		Does not stop
	When f_8, f_{32} selected		Does not stop when the CM02 bit is “0”. When the CM02 bit is “1”, the status immediately prior to entering wait mode is maintained.

Table 1.9.5. Interrupts to Exit Wait Mode

Interrupt	CM02=0	CM02=1
$\overline{NMI}$ interrupt	Can be used	Can be used
Serial I/O interrupt	Can be used when operating with internal or external clock	Can be used when operating with external clock
key input interrupt	Can be used	Can be used
A-D conversion interrupt	Can be used in one-shot mode or single sweep mode	— (Do not use)
Timer A interrupt Timer B interrupt	Can be used in all modes	Can be used in event counter mode or when the count source is fC32
$\overline{INT}$ interrupt	Can be used	Can be used

Table 1.9.5 lists the interrupts to exit wait mode.

If the microcomputer is to be moved out of wait mode by a peripheral function interrupt, set up the following before executing the WAIT instruction.

1. In the ILVL2 to ILVL0 bits of interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit wait mode.
   Also, for all of the peripheral function interrupts not used to exit wait mode, set the ILVL2 to ILVL0 bits to “000_2” (interrupt disable).
2. Set the I flag to “1”.
3. Enable the peripheral function whose interrupt is to be used to exit wait mode.
   In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt routine is executed.

The CPU clock turned on when exiting wait mode by a peripheral function interrupt is the same CPU clock that was on when the WAIT instruction was executed.

(3) Stop Mode

In stop mode, all oscillator circuits are turned off, so are the CPU clock and the peripheral function clocks. Therefore, the CPU and the peripheral functions clocked by these clocks stop operating. The least amount of power is consumed in this mode. If the voltage applied to Vcc1 and Vcc2 pins is V_{RAM} or more, the internal RAM is retained. When applying 2.7 or less voltage to Vcc1 and Vcc2 pins, make sure $Vcc1 \geq Vcc2 \geq V_{RAM}$.

However, the peripheral functions clocked by external signals keep operating. The following interrupts can be used to exit stop mode.

- $\overline{\text{NMI}}$ interrupt
- Key interrupt
- $\overline{\text{INT}}$ interrupt
- Timer A, Timer B interrupt (when counting external pulses in event counter mode)
- Serial I/O interrupt (when external clock is seleted)

• Entering Stop Mode

The microcomputer is placed into stop mode by setting the CM10 bit of CM1 register to “1” (all clocks turned off). At the same time, the CM06 bit of CM0 register is set to “1” (divide-by-8 mode) and the CM15 bit of CM10 register is set to “1” (main clock oscillator circuit drive capability high).

Before entering stop mode, set the CM20 bit to “0” (oscillation stop, re-oscillation detection function disable).

Also, if the CM11 bit is “1” (PLL clock for the CPU clock source), set the CM11 bit to “0” (main clock for the CPU clock source) and the PLC07 bit to “0” (PLL turned off) before entering stop mode.

• Pin Status in Stop Mode

Table 1.9.6 lists pin status during stop mode

• Exiting Stop Mode

The microcomputer is moved out of stop mode by a hardware reset, $\overline{\text{NMI}}$ interrupt or peripheral function interrupt.

If the microcomputer is to be moved out of stop mode by a hardware reset or $\overline{\text{NMI}}$ interrupt, set the peripheral function interrupt priority ILVL2 to ILVL0 bits to “000_2” (interrupts disable) before setting the CM10 bit to “1”.

If the microcomputer is to be moved out of stop mode by a peripheral function interrupt, set up the following before setting the CM10 bit to “1”.

1. In the ILVL2 to ILVL0 bits of interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit stop mode.
   Also, for all of the peripheral function interrupts not used to exit stop mode, set the ILVL2 to ILVL0 bits to “000_2”.
2. Set the I flag to “1”.
3. Enable the peripheral function whose interrupt is to be used to exit stop mode.
   In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt service routine is executed.

Which CPU clock will be used after exiting stop mode by a peripheral function or $\overline{\text{NMI}}$ interrupt is determined by the CPU clock that was on when the microcomputer was placed into stop mode as follows:

If the CPU clock before entering stop mode was derived from the sub clock: sub clock

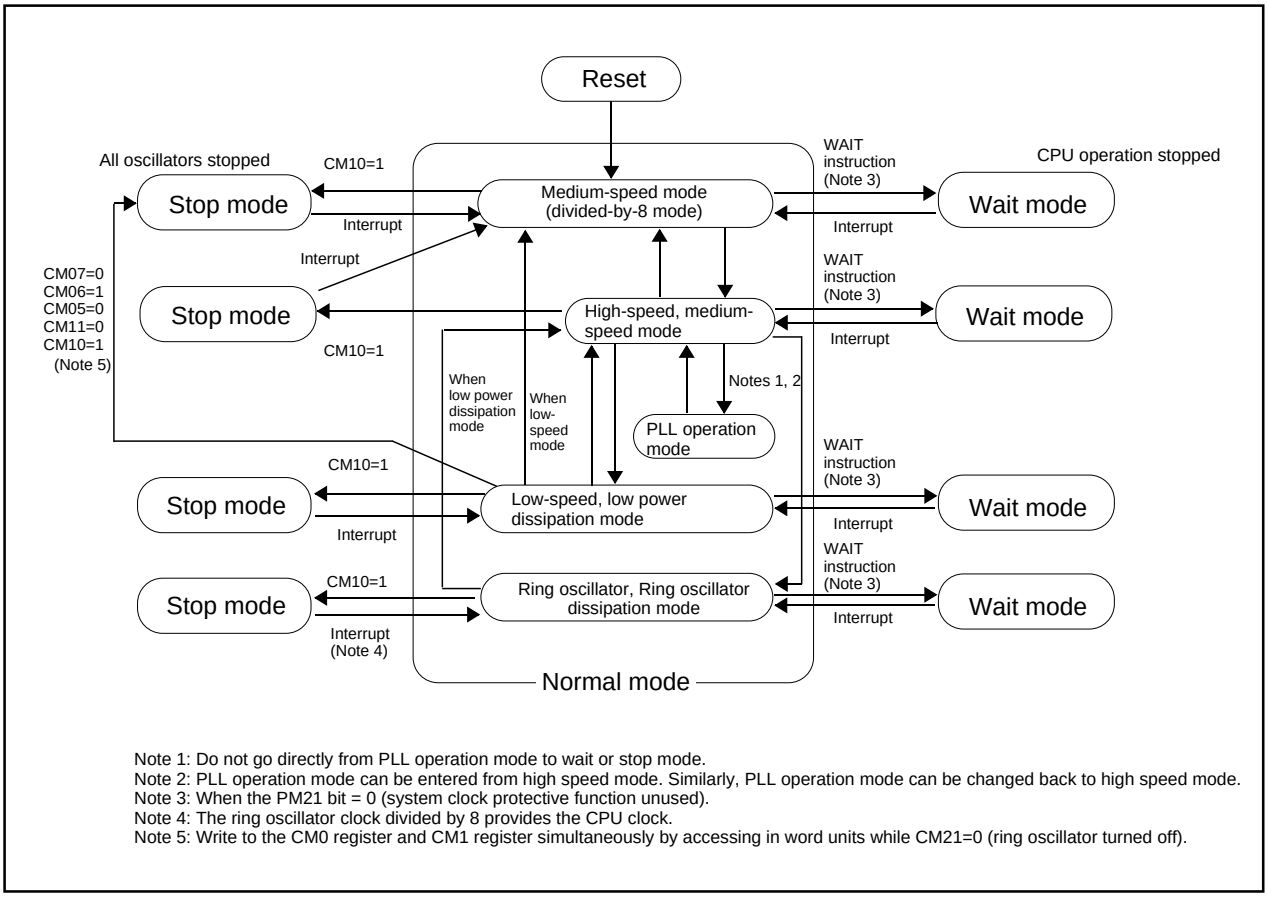
If the CPU clock before entering stop mode was derived from the main clock: main clock divide-by-8

If the CPU clock before entering stop mode was derived from the ring oscillator clock: ring oscillator clock divide-by-8

Table 1.9.6. Pin Status in Stop Mode

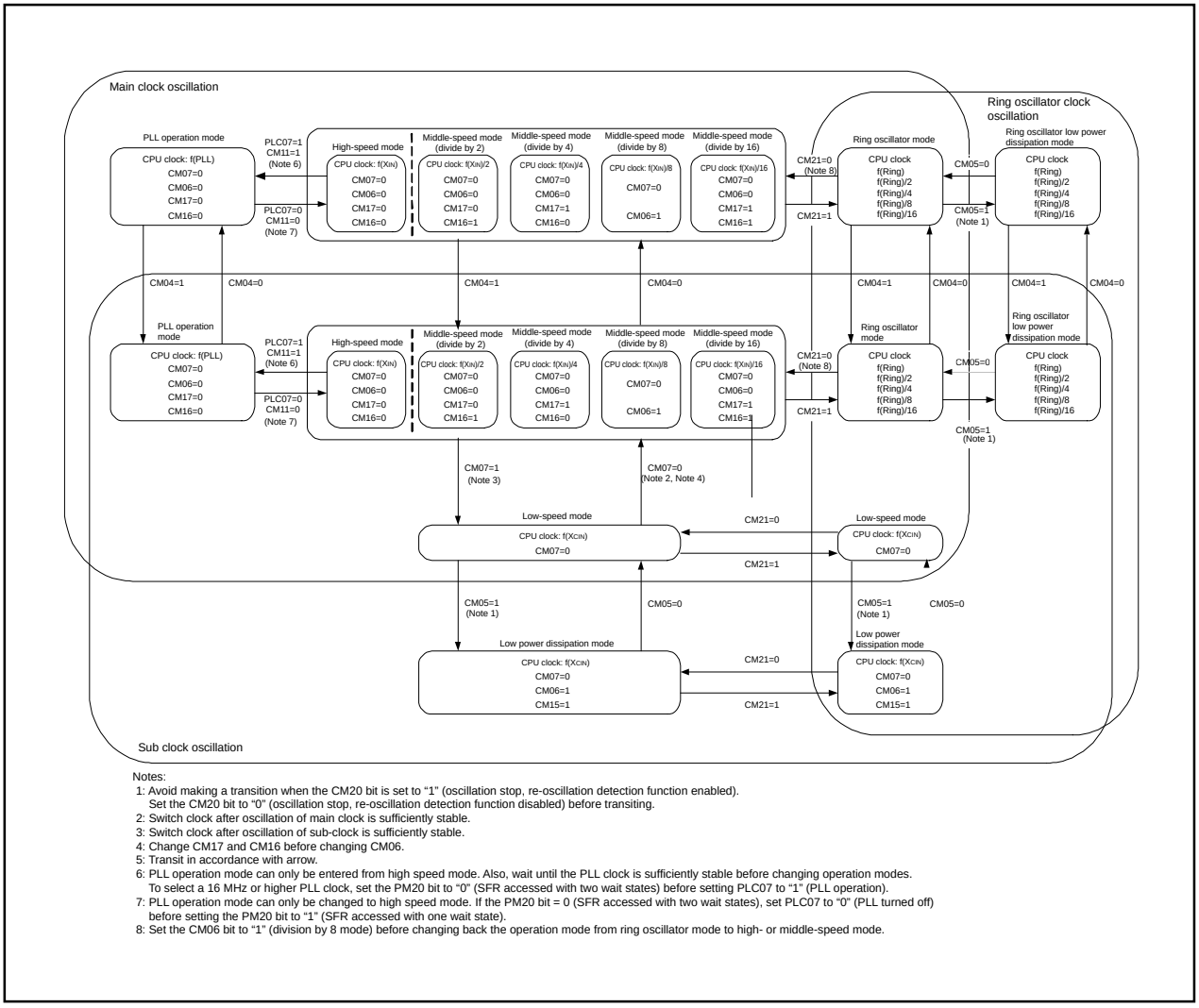
Pin		Memory expansion mode Microprocessor mode	Single-chip mode
A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ to $\overline{CS3}$, $\overline{BHE}$		Retains status before stop mode	
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		"H"	
$\overline{HLDA}$, BCLK		"H"	
ALE		"H"	
I/O ports		Retains status before stop mode	Retains status before stop mode
CLK_{OUT}	When fc selected		"H"
	When f_8, f_{32} selected		Retains status before stop mode

Figure 1.9.10 shows the state transition from normal operation mode to stop mode and wait mode. Figure 1.9.11 shows the state transition in normal operation mode.
Table 1.9.7 shows a state transition matrix describing allowed transition and setting. The vertical line shows current state and horizontal line shows state after transition.



Note 1: Do not go directly from PLL operation mode to wait or stop mode.
Note 2: PLL operation mode can be entered from high speed mode. Similarly, PLL operation mode can be changed back to high speed mode.
Note 3: When the PM21 bit = 0 (system clock protective function unused).
Note 4: The ring oscillator clock divided by 8 provides the CPU clock.
Note 5: Write to the CM0 register and CM1 register simultaneously by accessing in word units while CM21=0 (ring oscillator turned off).

Figure 1.9.10. State Transition to Stop Mode and Wait Mode



Notes:
1: Avoid making a transition when the CM20 bit is set to "1" (oscillation stop, re-oscillation detection function enabled).
Set the CM20 bit to "0" (oscillation stop, re-oscillation detection function disabled) before transiting.
2: Switch clock after oscillation of main clock is sufficiently stable.
3: Switch clock after oscillation of sub-clock is sufficiently stable.
4: Change CM17 and CM16 before changing CM06.
5: Transit in accordance with arrow.
6: PLL operation mode can only be entered from high speed mode. Also, wait until the PLL clock is sufficiently stable before changing operation modes.
To select a 16 MHz or higher PLL clock, set the PM20 bit to "0" (SFR accessed with two wait states) before setting PLC07 to "1" (PLL operation).
7: PLL operation mode can only be changed to high speed mode. If the PM20 bit = 0 (SFR accessed with two wait states), set PLC07 to "0" (PLL turned off) before setting the PM20 bit to "1" (SFR accessed with one wait state).
8: Set the CM06 bit to "1" (division by 8 mode) before changing back the operation mode from ring oscillator mode to high- or middle-speed mode.

Figure 1.9.11. State Transition in Normal Mode

Table 1.9.7. Allowed Transition and Setting

		State after transition							
		High-speed mode, middle-speed mode	Low-speed mode[1]	Low power dissipation mode[1]	PLL operation mode[1]	Ring oscillator mode	Ring oscillator low power dissipation mode	Stop mode	Wait mode
Current state	High-speed mode, middle-speed mode	See Table A	(9)[7]	--	(13)[3]	(15)	--	(16)[1]	(17)
	Low-speed mode[2]	(8)		(11)[1, 6]	--	--	--	(16)[1]	(17)
	Low power dissipation mode[2]	--	(10)		--	--	--	(16)[1]	(17)
	PLL operation mode[2]	(12)[3]	--	--		--	--	--	--
	Ring oscillator mode	(14)[4]	--	--	--	See Table A	(11)[1]	(16)[1]	(17)
	Ring oscillator low power dissipation mode	--	--	--	--	(10)	See Table A	(16)[1]	(17)
	Stop mode	(18)[5]	(18)	(18)	--	(18)[5]	(18)[5]		--
	Wait mode	(18)	(18)	(18)	--	(18)	(18)	--	

--: Cannot transit

Table 1. State Transition with Main Clock Division Ration in High- or Middle-speed Mode, Ring Oscillator Mode, and Ring Oscillator Low Power Dissipation Mode

		Sub clock oscillating					Sub clock turned off				
		No division	Divided by 2	Divided by 4	Divided by 8	Divided by 16	No division	Divided by 2	Divided by 4	Divided by 8	Divided by 16
Sub clock oscillating	No division		(4)	(5)	(7)	(6)	(1)	--	--	--	--
	Divided by 2	(3)		(5)	(7)	(6)	--	(1)	--	--	--
	Divided by 4	(3)	(4)		(7)	(6)	--	--	(1)	--	--
	Divided by 8	(3)	(4)	(5)		(6)	--	--	--	(1)	--
	Divided by 16	(3)	(4)	(5)	(7)		--	--	--	--	(1)
Sub clock turned off	No division	(2)	--	--	--	--		(4)	(5)	(7)	(6)
	Divided by 2	--	(2)	--	--	--	(3)		(5)	(7)	(6)
	Divided by 4	--	--	(2)	--	--	(3)	(4)		(7)	(6)
	Divided by 8	--	--	--	(2)	--	(3)	(4)	(5)		(6)
	Divided by 16	--	--	--	--	(2)	(3)	(4)	(5)	(7)	

--: Cannot transit

Notes:
1. Avoid making a transition when the CM21 bit is set to "1" (oscillation stop, re-oscillation detection function enabled). Set the CM21 bit to "0" (oscillation stop, re-oscillation detection function disabled) before transiting.
2. Ring oscillator clock oscillates and stops in low-speed mode and low power dissipation mode. In these mode, the ring oscillator can be used as peripheral function clock.
   Sub clock oscillates and stops in PLL operation mode. In this mode, sub clock can be used as peripheral function clock.
3. PLL operation mode can only be entered from and changed to high-speed mode.
4. Set the CM06 bit to "1" (division by 8 mode) before transiting from ring oscillator mode to high- or middle-speed mode.
5. When exiting stop mode, the CM06 bit is set to "1" (division by 8 mode).
6. If the CM05 bit is set to "1" (main clock stop), then the CM06 bit is set to "1" (division by 8 mode).
7. A transition can be made only when sub clock is oscillating.

Table B. Setting and Operation

	Setting	Operation
(1)	CM04 = 0	Sub clock turned off
(2)	CM04 = 1	Sub clock oscillating
(3)	CM06 = 0, CM17 = 0 , CM16 = 0	CPU clock no division mode
(4)	CM06 = 0, CM17 = 0 , CM16 = 1	CPU clock division by 2 mode
(5)	CM06 = 0, CM17 = 1 , CM16 = 0	CPU clock division by 4 mode
(6)	CM06 = 0, CM17 = 1 , CM16 = 1	CPU clock division by 16 mode
(7)	CM06 = 1	CPU clock division by 8 mode
(8)	CM07 = 0	Main clock, PLL clock, or ring oscillator clock selected
(9)	CM07 = 1	Sub clock selected
(10)	CM05 = 0	Main clock oscillating
(11)	CM05 = 1	Main clock turned off
(12)	PLC07 = 0, CM11 = 0	Main clock selected
(13)	PLC07 = 1, CM11 = 1	PLL clock selected
(14)	CM21 = 0	Main clock or PLL clock selected
(15)	CM21 = 1	Ring oscillator clock selected
(16)	CM10 = 1	Transition to stop mode
(17)	wait	Transition to wait mode
(18)	Hardware interrupt	Exit stop mode or wait mode

System Clock Protective Function

When the main clock is selected for the CPU clock source, this function disables the clock against modifications in order to prevent the CPU clock from becoming halted by run-away.

If the PM21 bit of PM2 register is set to “1” (clock modification disabled), the following bits are protected against writes:

- CM02, CM05, and CM07 bits in CM0 register
- CM10, CM11 bits in CM1 register
- CM20 bit in CM2 register
- All bits in PLC0 register

Before the system clock protective function can be used, the following register settings must be made while the CM05 bit of CM0 register is “0” (main clock oscillating) and CM07 bit is “0” (main clock selected for the CPU clock source):

(1) Set the PRC1 bit of PRCR register to “1” (enable writes to PM2 register).

(2) Set the PM21 bit of PM2 register to “1” (disable clock modification).

(3) Set the PRC1 bit of PRCR register to “0” (disable writes to PM2 register).

Do not execute the WAIT instruction when the PM21 bit is “1”.

Oscillation Stop and Re-oscillation Detect Function

The oscillation stop and re-oscillation detect function is such that main clock oscillation circuit stop and re-oscillation are detected. At oscillation stop, re-oscillation detection, reset or oscillation stop, re-oscillation detection interrupt are generated. Which is to be generated can be selected using the CM27 bit of CM2 register. Table 1.9.4 lists an specification overview of the oscillation stop and re-oscillation detect function.

Table 1.9.7. Specification Overview of Oscillation Stop and Re-oscillation Detect Function

Item	Specification
Oscillation stop detectable clock and frequency bandwidth	$f(X_{IN}) \geq 2$ MHz
Enabling condition for oscillation stop, re-oscillation detection function	Set CM20 bit to “1”(enable)
Operation at oscillation stop, re-oscillation detection	•Reset occurs (when CM27 bit =0) •Oscillation stop, re-oscillation detection interrupt occurs(when CM27 bit =1)

(1) Operation When CM27 bit = 0 (Oscillation Stop Detection Reset)

Where main clock stop is detected when the CM20 bit is “1” (oscillation stop, re-oscillation detection function enabled), the microcomputer is initialized, coming to a halt (oscillation stop reset; refer to “SFR”, “Reset”).

This status is reset with hardware reset 1 or hardware reset 2. Also, even when re-oscillation is detected, the microcomputer can be initialized and stopped; it is, however, necessary to avoid such usage. (During main clock stop, do not set the CM20 bit to “1” and the CM27 bit to “0”.)

(2) Operation When CM27 bit = 0 (Oscillation Stop and Re-oscillation Detect Interrupt)

Where the main clock corresponds to the CPU clock source and the CM20 bit is “1” (oscillation stop and re-oscillation detect function enabled), the system is placed in the following state if the main clock comes to a halt:

- Oscillation stop and re-oscillation detect interrupt request occurs.
- The ring oscillator starts oscillation, and the ring oscillator clock becomes the CPU clock and clock source for peripheral functions in place of the main clock.
- CM21 bit = 1 (ring oscillator clock for CPU clock source)
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)

Where the PLL clock corresponds to the CPU clock source and the CM20 bit is “1”, the system is placed in the following state if the main clock comes to a halt: Since the CM21 bit remains unchanged, set it to “1” (ring oscillator clock) inside the interrupt routine.

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)
- CM21 bit remains unchanged

Where the CM20 bit is “1”, the system is placed in the following state if the main clock re-oscillates from the stop condition:

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit = 1 (main clock re-oscillation detected)
- CM23 bit = 0 (main clock oscillation)
- CM21 bit remains unchanged

How to Use Oscillation Stop and Re-oscillation Detect Function

- The oscillation stop and re-oscillation detect interrupt shares the vector with the watchdog timer interrupt. If the re-oscillation detection and watchdog timer interrupts both are used, read the CM22 bit in an interrupt routine to determine which interrupt source is requesting the interrupt.
- Where the main clock re-oscillated after oscillation stop, return the main clock to the CPU clock and peripheral function clock source in the program. Figure 1.9.12 shows the procedure for switching the clock source from the ring oscillator to the main clock.
- Simultaneously with oscillation stop, re-oscillation detection interrupt occurrence, the CM22 bit becomes “1”. When the CM22 bit is set at “1”, oscillation stop, re-oscillation detection interrupt are disabled. By setting the CM22 bit to “0” in the program, oscillation stop, re-oscillation detection interrupt are enabled.
- If the main clock stops during low speed mode where the CM20 bit is “1”, an oscillation stop, re-oscillation detection interrupt request is generated. At the same time, the ring oscillator starts oscillating. In this case, although the CPU clock is derived from the sub clock as it was before the interrupt occurred, the peripheral function clocks now are derived from the ring oscillator clock.
- To enter wait mode while using the oscillation stop, re-oscillation detection function, set the CM02 bit to “0” (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop , re-oscillation detection function is provided in preparation for main clock stop due to external factors, set the CM20 bit to “0” (Oscillation stop , re-oscillation detection function disabled) where the main clock is stopped or oscillated in the program, that is where the stop mode is selected or the CM05 bit is altered.
- This function cannot be used if the main clock frequency is 2 MHz or less. In that case, set the CM20 bit to “0”.

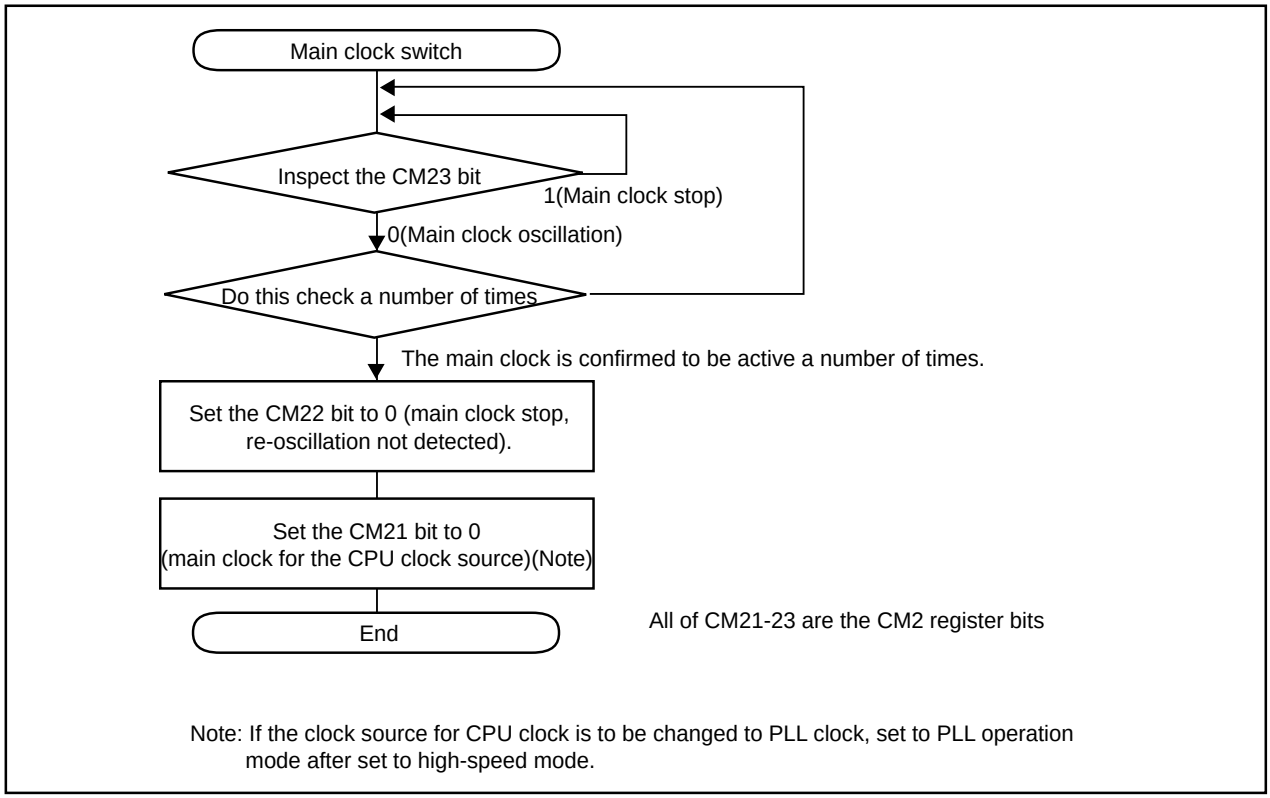


Figure 1.9.12. Procedure to Switch Clock Source From Ring Oscillator to Main Clock

Protection

In the event that a program runs out of control, this function protects the important registers so that they will not be rewritten easily. Figure 1.10.1 shows the PRCR register. The following lists the registers protected by the PRCR register.

- Registers protected by PRC0 bit: CM0, CM1, CM2, PLC0 and PCLKR registers
- Registers protected by PRC1 bit: PM0, PM1, PM2, TB2SC, INVC0 and INVC1 registers
- Registers protected by PRC2 bit: PD9, S3C and S4C registers
- Registers protected by PRC3 bit: VCR2 and D4INT registers

Set the PRC2 bit to “1” (write enabled) and then write to any address, and the PRC2 bit will be cleared to “0” (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to “1”. Make sure no interrupts or DMA transfers will occur between the instruction in which the PRC2 bit is set to “1” and the next instruction. The PRC0, PRC1 and PRC3 bits are not automatically cleared to “0” by writing to any address. They can only be cleared in a program.

Protect register

b7	b6	b5	b4	b3	b2	b1	b0
✕	✕	0	0				

Symbol: PRCR
Address: $000A_{16}$
After reset: $XX000000_2$

Bit symbol	Bit name	Function	RW
PRC0	Protect bit 0	Enable write to CM0, CM1, CM2, PLC0 and PCLKR registers 0 : Write protected 1 : Write enabled	RW
PRC1	Protect bit 1	Enable write to PM0, PM1, PM2, TB2SC, INVC0 and INVC1 registers 0 : Write protected 1 : Write enabled	RW
PRC2	Protect bit 2	Enable write to PD9, S3C and S4C registers 0 : Write protected 1 : Write enabled	RW
PRC3	Protect bit 3	Enable write to VCR2 and D4INT registers 0 : Write protected 1 : Write enabled	RW
—— (b5-b4)	Reserved bit	Must set to “0”	RW
—— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is interdeterminate.		—

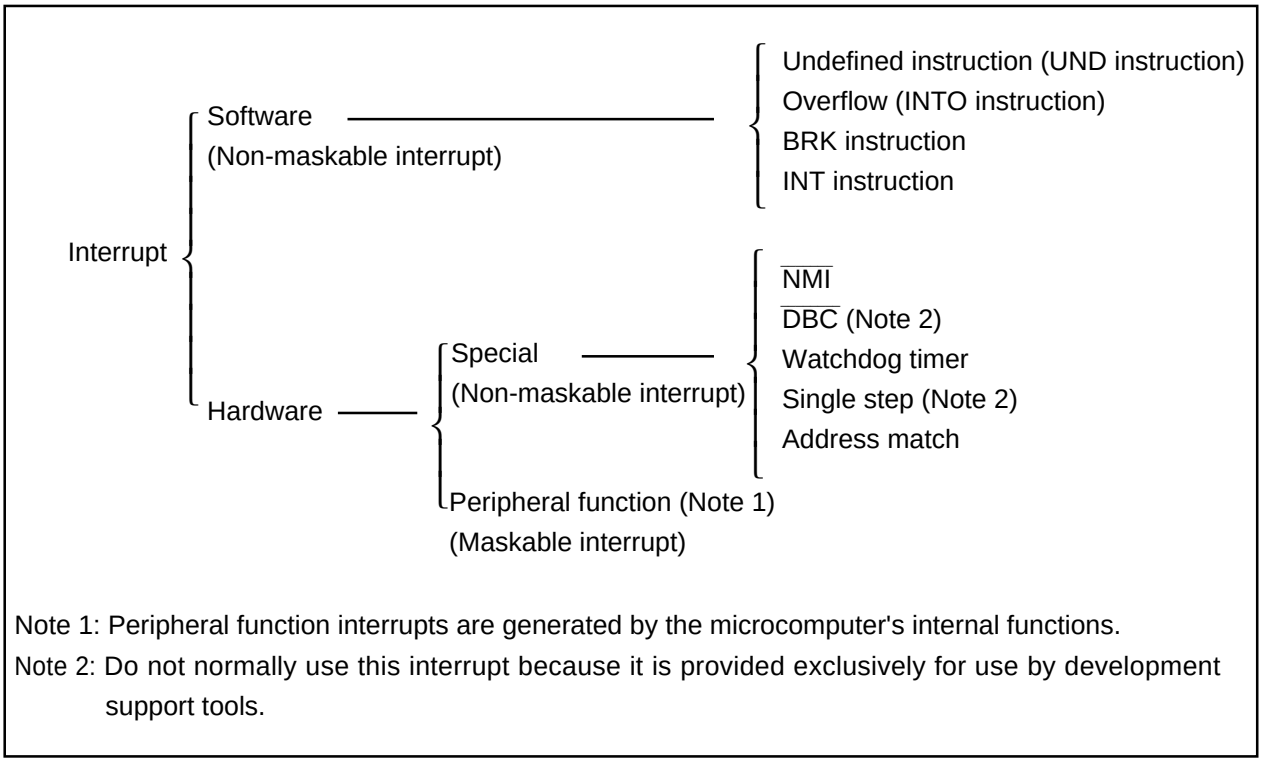
Note: The PRC2 bit is set to “0” by writing to any address after setting it to “1”. Other bits are not set to “0” by writing to any address, and must therefore be set in a program.

Figure 1.10.1. PRCR Register

Interrupts

Type of Interrupts

Figure 1.11.1 shows types of interrupts.



Note 1: Peripheral function interrupts are generated by the microcomputer's internal functions.
Note 2: Do not normally use this interrupt because it is provided exclusively for use by development support tools.

Figure 1.11.1. 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 I0nterrupt: 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 O flag set to “1” (the operation resulted in an overflow). 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 Instruction Interrupt**
  An INT instruction interrupt occurs when executing the INT instruction. Software interrupt Nos. 0 to 63 can be specified for the INT instruction. Because software interrupt Nos. 4 to 31 are assigned to peripheral function interrupts, the same interrupt routine as for peripheral function interrupts can be executed by executing the INT instruction.
  In software interrupt Nos. 0 to 31, the U flag is saved to the stack during instruction execution and is cleared to “0” (ISP selected) before executing an interrupt sequence. The U flag is restored from the stack when returning from the interrupt routine. In software interrupt Nos. 32 to 63, the U flag does not change state during instruction execution, and the SP then selected is used.

Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral function interrupts.

(1) Special Interrupts

Special interrupts are non-maskable interrupts.

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

An $\overline{\text{NMI}}$ interrupt is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low. For details about the $\overline{\text{NMI}}$ interrupt, refer to the section "NMI interrupt".

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

Do not normally use this interrupt because it is provided exclusively for use by development support tools.

- **Watchdog Timer Interrupt**

Generated by the watchdog timer. Once a watchdog timer interrupt is generated, be sure to initialize the watchdog timer. For details about the watchdog timer, refer to the section "watchdog timer".

- **Oscillation Stop and Re-oscillation Detection Interrupt**

Generated by the oscillation stop and re-oscillation detection function. For details about the oscillation stop detection function, refer to the section "clock generating circuit".

- **Power Supply Down Detection Interrupt**

Generated by the voltage detection circuit. For details about the voltage detection circuit, refer to the section "voltage detection circuit".

- **Single-step Interrupt**

Do not normally use this interrupt because it is provided exclusively for use by development support tools.

- **Address Match Interrupt**

An address match interrupt is generated immediately before executing the instruction at the address indicated by the RMAD0 to RMAD3 register that corresponds to one of the AIER register's AIER0 or AIER1 bit or the AIER2 register's AIER20 or AIER21 bit which is "1" (address match interrupt enabled). For details about the address match interrupt, refer to the section "address match interrupt".

(2) Peripheral Function Interrupts

Peripheral function interrupts are maskable interrupts and generated by the microcomputer's internal functions. The interrupt sources for peripheral function interrupts are listed in Table 1.11.2. For details about the peripheral functions, refer to the description of each peripheral function in this manual.

Interrupts and Interrupt Vector

One interrupt vector consists of 4 bytes. Set the start address of each interrupt routine in the respective interrupt vectors. When an interrupt request is accepted, the CPU branches to the address set in the corresponding interrupt vector. Figure 1.11.2 shows the interrupt vector.

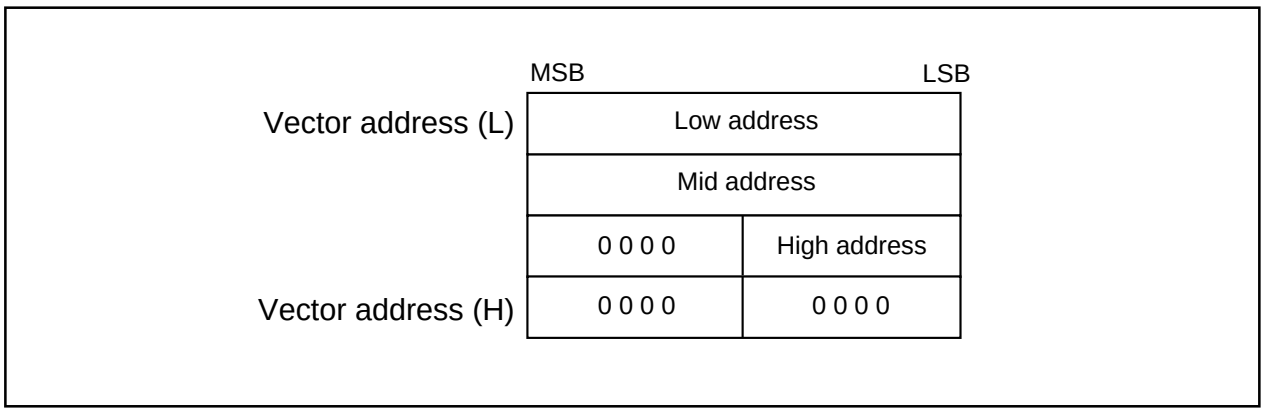


Figure 1.11.2. Interrupt Vector

• Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDC16 to FFFFF16. Table 1.11.1 lists the fixed vector tables. In the flash memory version of microcomputer, the vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to the section "flash memory rewrite disabling function".

Table 1.11.1. Fixed Vector Tables

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks	Reference
Undefined instruction	FFFDC16 to FFFDF16	Interrupt on UND instruction	M16C/60, M16C/20 serise software maual
Overflow	FFFE016 to FFFE316	Interrupt on INTO instruction	
BRK instruction	FFFE416 to FFFE716	If the contents of address FFFE716 is FF16, program execution starts from the address shown by the vector in the relocatable vector table.	
Address match	FFFE816 to FFFEB16		Address match interrupt
Single step (Note)	FFFEC16 to FFFEF16		
Watchdog timer Oscillation stop and re-oscillation detection Power supply down detection	FFFF016 to FFFF316		Watchdog timer Clock generating circuit Voltage detection circuit
$\overline{\text{DBC}}$ (Note)	FFFF416 to FFFF716		
$\overline{\text{NMI}}$	FFFF816 to FFFFB16		$\overline{\text{NMI}}$ interrupt
Reset	FFFFC16 to FFFFF16		Reset

Note: Do not normally use this interrupt because it is provided exclusively for use by development support tools.

• Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a reloacatable vector table area. Table 1.11.2 lists the relocatable vector tables. Setting an even address in the INTB register results in the interrupt sequence being executed faster than in the case of odd addresses.

Table 1.11.2. Relocatable Vector Tables

Interrupt source	Vector address (Note 1) Address (L) to address (H)	Software interrupt number	Reference
BRK instruction (Note 5)	+0 to +3 (0000_{16} to 0003_{16})	0	M16C/60, M16C/20 series software manual
——— (Reserved)		1 to 3	
$\overline{INT3}$	+16 to +19 (0010_{16} to 0013_{16})	4	$\overline{INT}$ interrupt
Timer B5	+20 to +23 (0014_{16} to 0017_{16})	5	Timer
Timer B4, UART1 bus collision detect (Note 4)	+24 to +27 (0018_{16} to $001B_{16}$)	6	Timer Serial I/O
Timer B3, UART0 bus collision detect (Note 4)	+28 to +31 ($001C_{16}$ to $001F_{16}$)	7	
SI/O4, $\overline{INT5}$ (Note 2)	+32 to +35 (0020_{16} to 0023_{16})	8	$\overline{INT}$ interrupt Serial I/O
SI/O3, $\overline{INT4}$ (Note 2)	+36 to +39 (0024_{16} to 0027_{16})	9	
UART 2 bus collision detection	+40 to +43 (0028_{16} to $002B_{16}$)	10	Serial I/O
DMA0	+44 to +47 ($002C_{16}$ to $002F_{16}$)	11	DMAC
DMA1	+48 to +51 (0030_{16} to 0033_{16})	12	
Key input interrupt	+52 to +55 (0034_{16} to 0037_{16})	13	Key input interrupt
A-D	+56 to +59 (0038_{16} to $003B_{16}$)	14	A-D convertor
UART2 transmit, NACK2 (Note 3)	+60 to +63 ($003C_{16}$ to $003F_{16}$)	15	Serial I/O
UART2 receive, ACK2 (Note 3)	+64 to +67 (0040_{16} to 0043_{16})	16	
UART0 transmit, NACK0 (Note 3)	+68 to +71 (0044_{16} to 0047_{16})	17	
UART0 receive, ACK0 (Note 3)	+72 to +75 (0048_{16} to $004B_{16}$)	18	
UART1 transmit, NACK1 (Note 3)	+76 to +79 ($004C_{16}$ to $004F_{16}$)	19	
UART1 receive, ACK1 (Note 3)	+80 to +83 (0050_{16} to 0053_{16})	20	
Timer A0	+84 to +87 (0054_{16} to 0057_{16})	21	Timer
Timer A1	+88 to +91 (0058_{16} to $005B_{16}$)	22	
Timer A2	+92 to +95 ($005C_{16}$ to $005F_{16}$)	23	
Timer A3	+96 to +99 (0060_{16} to 0063_{16})	24	
Timer A4	+100 to +103 (0064_{16} to 0067_{16})	25	
Timer B0	+104 to +107 (0068_{16} to $006B_{16}$)	26	
Timer B1	+108 to +111 ($006C_{16}$ to $006F_{16}$)	27	
Timer B2	+112 to +115 (0070_{16} to 0073_{16})	28	
$\overline{INT0}$	+116 to +119 (0074_{16} to 0077_{16})	29	$\overline{INT}$ interrupt
$\overline{INT1}$	+120 to +123 (0078_{16} to $007B_{16}$)	30	
$\overline{INT2}$	+124 to +127 ($007C_{16}$ to $007F_{16}$)	31	
Software interrupt (Note 5)	+128 to +131 (0080_{16} to 0083_{16}) to +252 to +255 ($00FC_{16}$ to $00FF_{16}$)	32 to 63	M16C/60, M16C/20 series software manual

Note 1: Address relative to address in INTB.
Note 2: Use the IFSR register's IFSR6 and IFSR7 bits to select.
Note 3: During I^2C mode, NACK and ACK interrupts comprise the interrupt source.
Note 4: Use the IFSR2A register's IFSR26 and IFSR27 bits to select.
Note 5: These interrupts cannot be disabled using the I flag.

Interrupt Control

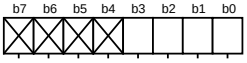
The following describes how to enable/disable the maskable interrupts, and how to set the priority in which order they are accepted. What is explained here does not apply to nonmaskable interrupts.

Use the FLG register's I flag, IPL, and each interrupt control register's ILVL2 to ILVL0 bits to enable/disable the maskable interrupts. Whether an interrupt is requested is indicated by the IR bit in each interrupt control register.

Figure 1.11.3 shows the interrupt control registers.

Interrupt control register (Note 2)

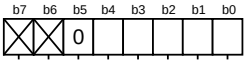
Symbol	Address	After reset
TB5IC	0045_{16}	$XXXXX000_2$
TB4IC/U1BCNIC (Note 3)	0046_{16}	$XXXXX000_2$
TB3IC/U0BCNIC (Note 3)	0047_{16}	$XXXXX000_2$
BCNIC	$004A_{16}$	$XXXXX000_2$
DM0IC, DM1IC	$004B_{16}$, $004C_{16}$	$XXXXX000_2$
KUPIC	$004D_{16}$	$XXXXX000_2$
ADIC	$004E_{16}$	$XXXXX000_2$
S0TIC to S2TIC	0051_{16}, 0053_{16}, $004F_{16}$	$XXXXX000_2$
S0RIC to S2RIC	0052_{16}, 0054_{16}, 0050_{16}	$XXXXX000_2$
TA0IC to TA4IC	0055_{16} to 0059_{16}	$XXXXX000_2$
TB0IC to TB2IC	$005A_{16}$ to $005C_{16}$	$XXXXX000_2$



Bit symbol	Bit name	Function	RW
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	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	RW (Note 1)
— (b7-b4)	No functions are assigned. When writing to these bits, write “0”. The values in these bits when read are indeterminate.		—

Note 1: This bit can only be reset by writing "0" (Do not write "1").
Note 2: To rewrite the interrupt control registers, do so at a point that does not generate the interrupt request for that register. For details, see the precautions for interrupts.
Note 3: Use the IFSR2A register to select.

Symbol	Address	After reset
INT3IC (Note 4)	0044_{16}	$XX00X000_2$
S4IC/INT5IC	0048_{16}	$XX00X000_2$
S3IC/INT4IC	0049_{16}	$XX00X000_2$
INT0IC to INT2IC	$005D_{16}$ to $005F_{16}$	$XX00X000_2$



Bit symbol	Bit name	Function	RW
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	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0: Interrupt not requested 1: Interrupt requested	RW (Note 1)
POL	Polarity select bit	0 : Selects falling edge (Notes 3, 5) 1 : Selects rising edge	RW
—	Reserved bit	Must always be set to “0”	RW
— (b7-b6)	No functions are assigned. When writing to these bits, write “0”. The values in these bits when read are indeterminate.		RW

Note 1: This bit can only be reset by writing "0" (Do not write "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.
Note 3: If the IFSR register’s IFSRi bit (i = 0 to 5) is "1" (both edges), set the INTiIC register’s POL bit to "0 "(falling edge).
Note 4: During memory expansion and microprocessor modes, set the INT3IC register’s ILVL2 to ILVL0 bits to ‘000_2’ (interrupt disabled).
Note 5: Set the S3IC or S4IC register’s POL bit to "0" (falling edge) when the IFSR register’s IFSR6 bit = 0 (SI/O3 selected) or IFSR7 bit = 0 (SI/O4 selected), respectively.

Figure 1.11.3. Interrupt Control Registers

I Flag

The I flag enables or disables the maskable interrupt. Setting the I flag to “1” (= enabled) enables the maskable interrupt. Setting the I flag to “0” (= disabled) disables all maskable interrupts.

IR Bit

The IR bit is set to “1” (= interrupt requested) when an interrupt request is generated. Then, when the interrupt request is accepted and the CPU branches to the corresponding interrupt vector, the IR bit is cleared to “0” (= interrupt not requested).
The IR bit can be cleared to “0” in a program. Note that do not write “1” to this bit.

ILVL2 to ILVL0 Bits and IPL

Interrupt priority levels can be set using the ILVL2 to ILVL0 bits.
Table 1.11.3 shows the settings of interrupt priority levels and Table 1.11.4 shows the interrupt priority levels enabled by the IPL.

The following are conditions under which an interrupt is accepted:

- I flag = “1”
- IR bit = “1”
- interrupt priority level > IPL

The I flag, IR bit, ILVL2 to ILVL0 bits and IPL are independent of each other. In no case do they affect one another.

Table 1.11.3. Settings of Interrupt Priority Levels

ILVL2 to ILVL0 bits	Interrupt priority level	Priority order
000_2	Level 0 (interrupt disabled)	———
001_2	Level 1	Low
010_2	Level 2	↓
011_2	Level 3	↓
100_2	Level 4	↓
101_2	Level 5	↓
110_2	Level 6	↓
111_2	Level 7	High

Table 1.11.4. Interrupt Priority Levels Enabled by IPL

IPL	Enabled interrupt priority levels
000_2	Interrupt levels 1 and above are enabled
001_2	Interrupt levels 2 and above are enabled
010_2	Interrupt levels 3 and above are enabled
011_2	Interrupt levels 4 and above are enabled
100_2	Interrupt levels 5 and above are enabled
101_2	Interrupt levels 6 and above are enabled
110_2	Interrupt levels 7 and above are enabled
111_2	All maskable interrupts are disabled

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.

The CPU behavior during the interrupt sequence is described below. Figure 1.11.4 shows time required for executing the interrupt sequence.

(1) The CPU gets interrupt information (interrupt number and interrupt request priority level) by reading the address 00000_{16}. Then it clears the IR bit for the corresponding interrupt to “0” (interrupt not requested).
(2) The FLG register immediately before entering the interrupt sequence is saved to the CPU’s internal temporary register(Note 1).
(3) The I, D and U flags in the FLG register become as follows:
The I flag is cleared to “0” (interrupts disabled).
The D flag is cleared to “0” (single-step interrupt disabled).
The U flag is cleared to “0” (ISP selected).
However, the U flag does not change state if an INT instruction for software interrupt Nos. 32 to 63 is executed.
(4) The CPU’s internal temporary register (Note 1) is saved to the stack.
(5) The PC is saved to the stack.
(6) The interrupt priority level of the accepted interrupt is set in the IPL.
(7) The start address of the relevant interrupt routine set in the interrupt vector is stored in the PC.

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

Note: This register cannot be used by user.

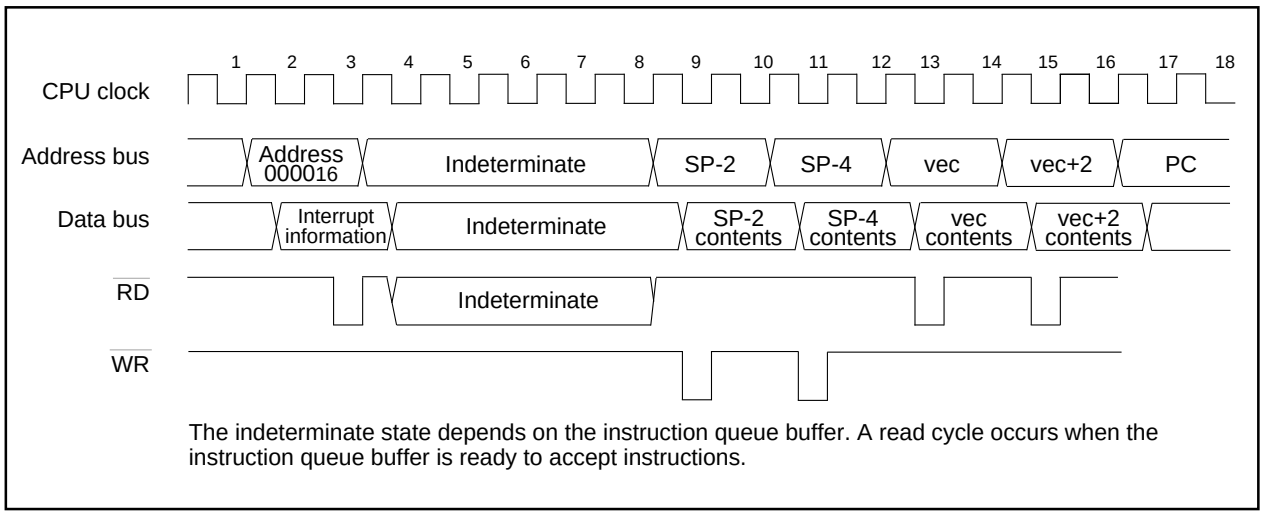
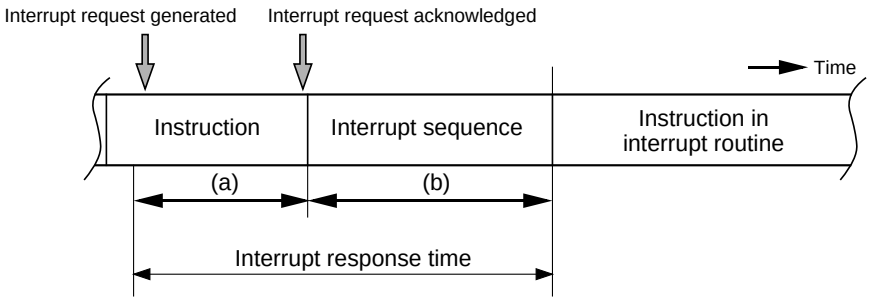


Figure 1.11.4. Time Required for Executing Interrupt Sequence

Interrupt Response Time

Figure 1.11.5 shows the interrupt response time. The interrupt response or interrupt acknowledge time denotes a time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of a time from when an interrupt request is generated till when the instruction then executing is completed ((a) in Figure 1.11.5) and a time during which the interrupt sequence is executed ((b) in Figure 1.11.5).



(a) A time from when an interrupt request is generated till when the instruction then executing is completed. The length of this time varies with the instruction being executed. The DIVX instruction requires the longest time, which is equal to 30 cycles (without wait state, the divisor being a register).

(b) A time during which the interrupt sequence is executed. For details, see the table below. Note, however, that the values in this table must be increased 2 cycles for the $\overline{\text{DBC}}$ interrupt and 1 cycle for the address match and single-step interrupts.

Interrupt vector address	SP value	16-Bit bus, without wait	8-Bit bus, without wait
Even	Even	18 cycles	20 cycles
Even	Odd	19 cycles	20 cycles
Odd	Even	19 cycles	20 cycles
Odd	Odd	20 cycles	20 cycles

Figure 1.11.5. Interrupt response time

Variation of IPL when Interrupt Request is Accepted

When a maskable interrupt request is accepted, the interrupt priority level of the accepted interrupt is set in the IPL.

When a software interrupt or special interrupt request is accepted, one of the interrupt priority levels listed in Table 1.11.5 is set in the IPL. Shown in Table 1.11.5 are the IPL values of software and special interrupts when they are accepted.

Table 1.11.5. IPL Level That is Set to IPL When A Software or Special Interrupt Is Accepted

Interrupt sources	Level that is set to IPL
Watchdog timer, $\overline{\text{NMI}}$	7
Software, address match, $\overline{\text{DBC}}$, single-step	Not changed

Saving Registers

In the interrupt sequence, the FLG register and PC are saved to the stack.

At this time, the 4 high-order bits of the PC and the 4 high-order (IPL) and 8 low-order bits of the FLG register, 16 bits in total, are saved to the stack first. Next, the 16 low-order bits of the PC are saved. Figure 1.11.6 shows the stack status before and after an interrupt request is accepted.

The other necessary registers must be saved in a program at the beginning of the interrupt routine. Use the PUSHM instruction, and all registers except SP can be saved with a single instruction.

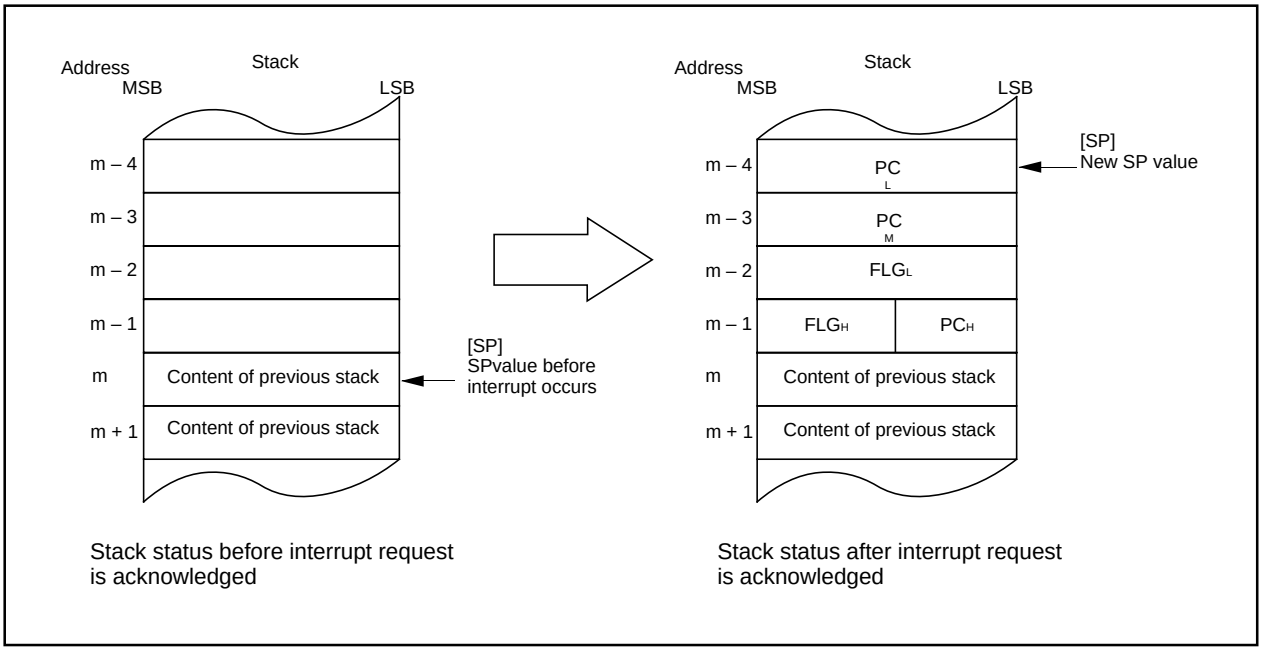


Figure 1.11.6. Stack StatusBefore and After Acceptance of Interrupt Request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP(Note), at the time of acceptance of an interrupt request, is even or odd. If the stack pointer (Note) is even, the FLG register and the PC are saved, 16 bits at a time. If odd, they 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 SP indicated by the U flag. Otherwise, it is the ISP.

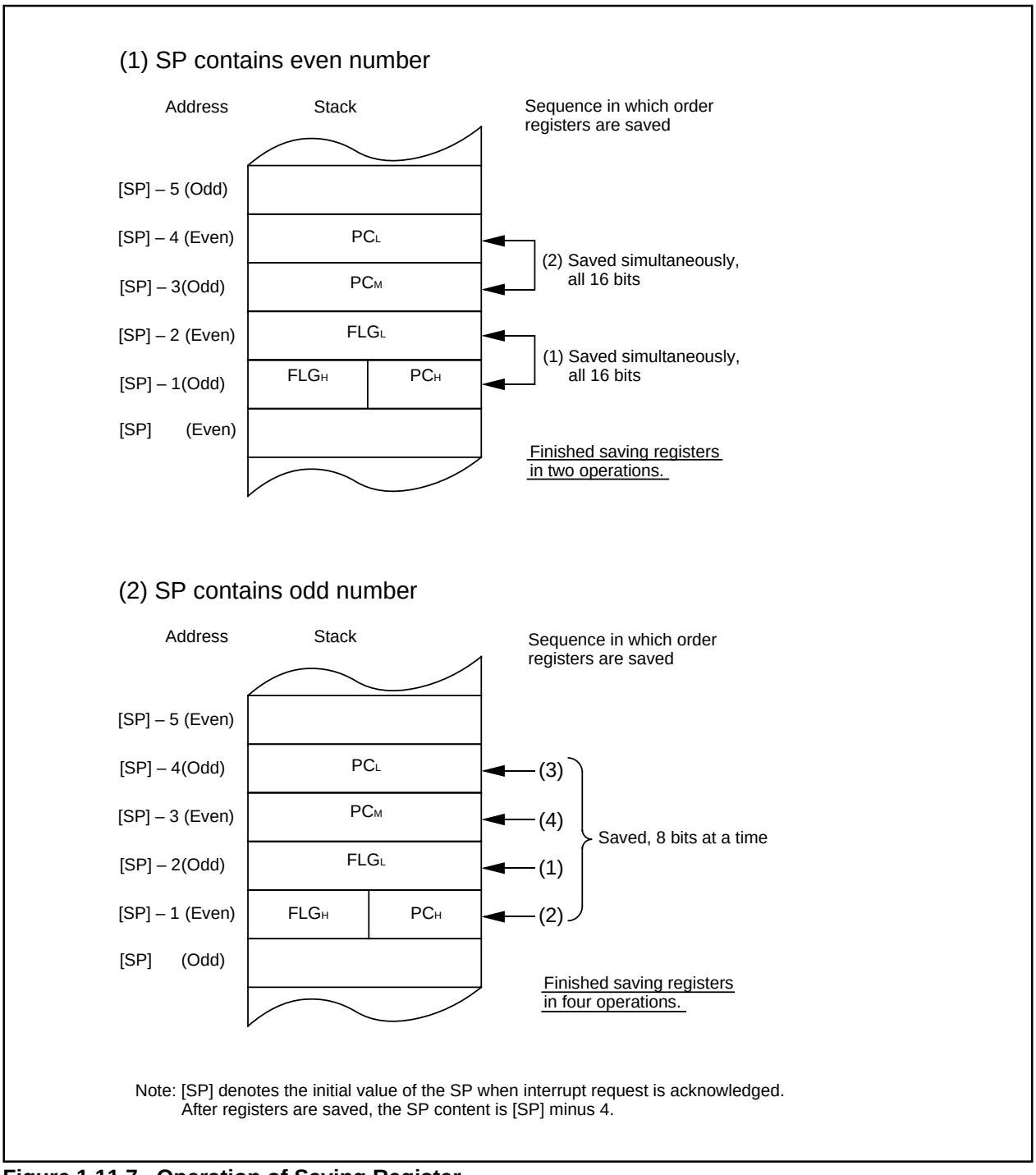


Figure 1.11.7. Operation of Saving Register

Returning from an Interrupt Routine

The FLG register and PC in the state in which they were immediately before entering the interrupt sequence are restored from the stack by executing the REIT instruction at the end of the interrupt routine. Thereafter the CPU returns to the program which was being executed before accepting the interrupt request.

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

Interrupt Priority

If two or more interrupt requests are generated while executing one instruction, the interrupt request that has the highest priority is accepted.

For maskable interrupts (peripheral functions), any desired priority level can be selected using the ILVL2 to ILVL0 bits. However, if two or more maskable interrupts have the same priority level, their interrupt priority is resolved by hardware, with the highest priority interrupt accepted.

The watchdog timer and other special interrupts have their priority levels set in 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 > $\overline{\text{NMI}}$ > $\overline{\text{DBC}}$ > WDT > Peripheral function > Single step > Address match

Figure 1.11.8. Hardware Interrupt Priority

Interrupt Priority Resolution Circuit

The interrupt priority resolution circuit is used to select the interrupt with the highest priority among those requested.

Figure 1.11.9 shows the circuit that judges the interrupt priority level.

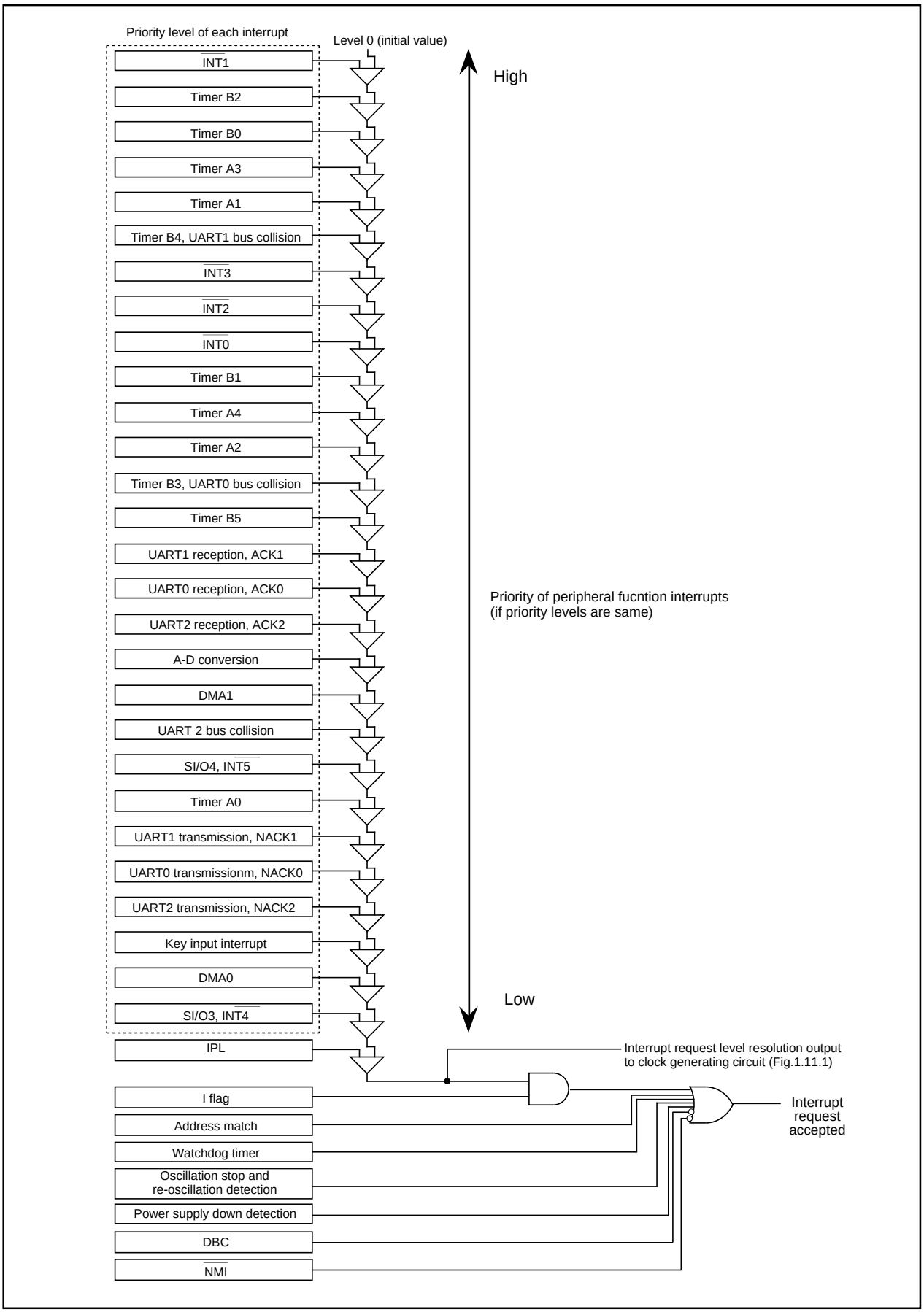


Figure 1.11.9. Interrupts Priority Select Circuit

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

$\overline{\text{INT}}$i interrupt (i=0 to 5) is triggered by the edges of external inputs. The edge polarity is selected using the IFSR register's IFSRi bit.

$\overline{\text{INT}}$4 and $\overline{\text{INT}}$5 share the interrupt vector and interrupt control register with SI/O3 and SI/O4, respectively. To use the $\overline{\text{INT}}$4 interrupt, set the IFSR register's IFSR6 bit to "1" (= $\overline{\text{INT}}$4). To use the $\overline{\text{INT}}$5 interrupt, set the IFSR register's IFSR7 bit to "1" (= $\overline{\text{INT}}$5).

After modifying the IFSR6 or IFSR7 bit, clear the corresponding IR bit to "0" (= interrupt not requested) before enabling the interrupt.

Figure 1.11.10 shows the IFSR and IFSR2A registers.

Interrupt request cause select register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: IFSR | Address: $035F_{16}$ | After reset: 00_{16}

Bit symbol	Bit name	Function	RW
IFSR0	INT0 interrupt polarity switching bit	0 : One edge 1 : Both edges (Note 1)	RW
IFSR1	INT1 interrupt polarity switching bit	0 : One edge 1 : Both edges (Note 1)	RW
IFSR2	INT2 interrupt polarity switching bit	0 : One edge 1 : Both edges (Note 1)	RW
IFSR3	INT3 interrupt polarity switching bit	0 : One edge 1 : Both edges (Note 1)	RW
IFSR4	INT4 interrupt polarity switching bit	0 : One edge 1 : Both edges (Note 1)	RW
IFSR5	INT5 interrupt polarity switching bit	0 : One edge 1 : Both edges (Note 1)	RW
IFSR6	Interrupt request cause select bit (Note 2)	0 : SI/O3 (Note 3) 1 : $\overline{\text{INT4}}$	RW
IFSR7	Interrupt request cause select bit (Note 2)	0 : SI/O4 1 : $\overline{\text{INT5}}$	RW

Note 1: When setting this bit to "1" (= both edges), make sure the INT0IC to INT5IC register's POL bit is set to "0" (= falling edge).
Note 2: During memory expansion and microprocessor modes, set this bit to "0" (= SI/O3, SI/O4)
Note 3: When setting this bit to "0" (= SI/O3, SI/O4), make sure the S3IC and S4IC registers' POL bit is set to "0" (= falling edge).

Interrupt request cause select register 2

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: IFSR2A | Address: $035E_{16}$ | After reset: $00XXXXXX_2$

Bit symbol	Bit name	Function	RW
— (b5-b0)	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.		—
IFSR26	Interrupt request cause select bit	0 : Timer B3 1 : UART0 bus collision detection	RW
IFSR27	Interrupt request cause select bit	0 : Timer B4 1 : UART1 bus collision detection	RW

Figure 1.11.10. IFSR Register and IFSR2A Register

$\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low. The $\overline{\text{NMI}}$ interrupt is a non-maskable interrupt.

The input level of this $\overline{\text{NMI}}$ interrupt input pin can be read by accessing the P8 register's P8_5 bit.

This pin cannot be used as an input port.

Key Input Interrupt

Of $P10_4$ to $P10_7$, a key input interrupt is generated when input on any of the $P10_4$ to $P10_7$ pins which has had the PD10 register's PD10_4 to PD10_7 bits set to "0" (= input) goes low. Key input interrupts can be used as a key-on wakeup function, the function which gets the microcomputer out of wait or stop mode. However, if you intend to use the key input interrupt, do not use $P10_4$ to $P10_7$ as analog input ports. Figure 1.11.11 shows the block diagram of the key input interrupt. Note, however, that while input on any pin which has had the PD10_4 to PD10_7 bits set to "0" (= input mode) is pulled low, inputs on all other pins of the port are not detected as interrupts.

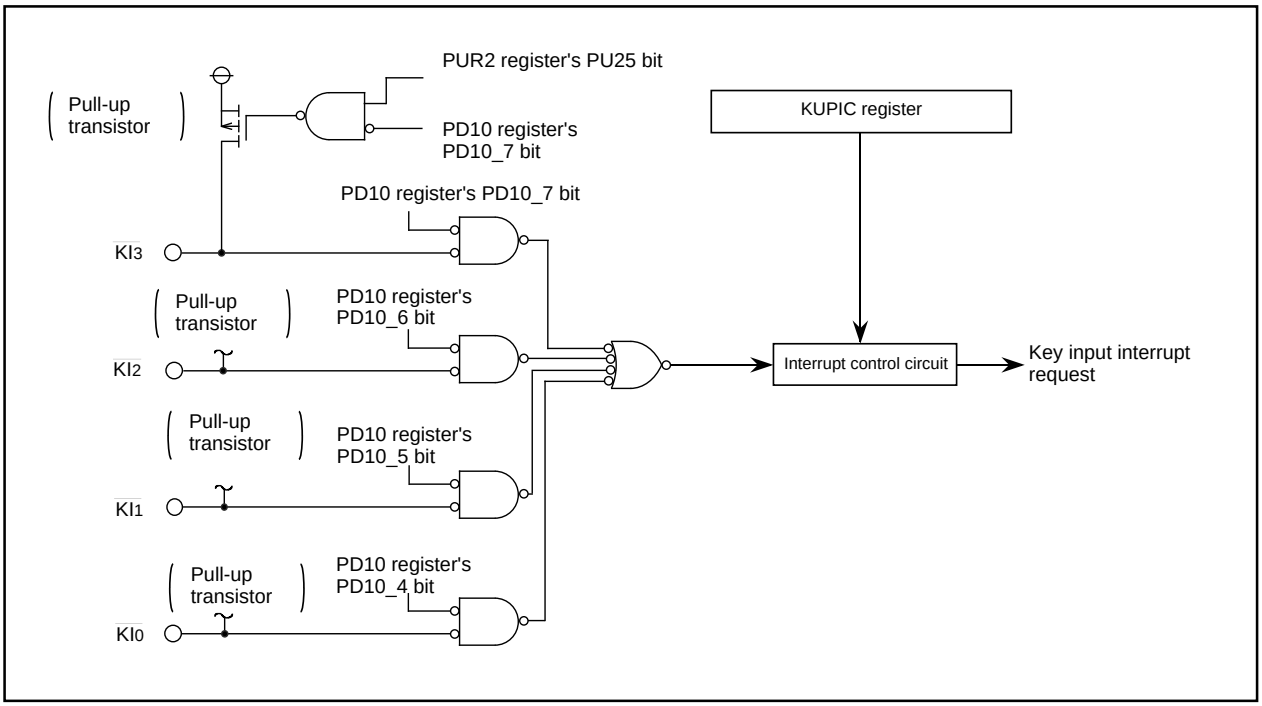


Figure 1.11.11. Key Input Interrupt

Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by the RMADi register (i=0 to 3). Set the start address of any instruction in the RMADi register. Use the AIER register's AIER0 and AIER1 bits and the AIER2 register's AIER20 and AIER21 bits to enable or disable the interrupt. Note that the address match interrupt is unaffected by the I flag and IPL. For address match interrupts, the value of the PC that is saved to the stack area varies depending on the instruction being executed. Figure 1.11.12 shows the instruction just before execution and address stored in the stack when there occurs interruption.

Note that when using the external data bus in width of 8 bits, the address match interrupt cannot be used for external area.

Figure 1.11.13 shows the AIER, AIER2, and RMAD0 to RMAD3 registers.

(1) Instructions in which the "return destination + 2" address is stored in the stack when address match interrupt occurs

- 16-bit operation code
- Instruction shown below among 8-bit operation code instructions

ADD.B:S	#IMM8,dest	SUB.B:S	#IMM8,dest	AND.B:S	#IMM8,dest
OR.B:S	#IMM8,dest	MOV.B:S	#IMM8,dest	STZ.B:S	#IMM8,dest
STNZ.B:S	#IMM8,dest	STZX.B:S	#IMM81,#IMM82,dest		
CMP.B:S	#IMM8,dest	PUSHM	src	POPM	dest
JMPS	#IMM8	JSRS	#IMM8		
MOV.B:S	#IMM,dest (However, dest = A0 or A1)				

(2) Instructions in which the "return destination + 1" address is stored in the stack when address match interrupt occurs

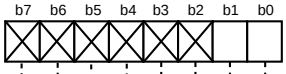
- Instructions other than the above

Figure 1.11.12. Instruction Just Before Execution and Address Stored in Stack When There Occurs Interrupts

Table 1.11.6. Relationship Between Address Match Interrupt Sources and Associated Registers

Address match interrupt sources	Address match interrupt enable bit	Address match interrupt register
Address match interrupt 0	AIER0	RMAD0
Address match interrupt 1	AIER1	RMAD1
Address match interrupt 2	AIER20	RMAD2
Address match interrupt 3	AIER21	RMAD3

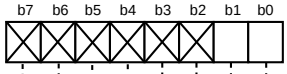
Address match interrupt enable register



Symbol	Address	After reset
AIER	0009_{16}	$XXXXXX00_2$

Bit symbol	Bit name	Function	RW
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
—— (b7-b2)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

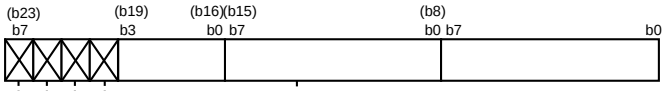
Address match interrupt enable register 2



Symbol	Address	After reset
AIER2	$01BB_{16}$	$XXXXXX00_2$

Bit symbol	Bit name	Function	RW
AIER20	Address match interrupt 2 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
AIER21	Address match interrupt 3 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
—— (b7-b2)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

Address match interrupt register i (i = 0 to 3)



Symbol	Address	After reset
RMAD0	0012_{16} to 0010_{16}	$X0000016_{16}$
RMAD1	0016_{16} to 0014_{16}	$X0000016_{16}$
RMAD2	$01BA_{16}$ to $01B8_{16}$	$X0000016_{16}$
RMAD3	$01BE_{16}$ to $01BC_{16}$	$X0000016_{16}$

Function	Setting range	RW
Address setting register for address match interrupt	00000_{16} to $FFFFF_{16}$	RW
Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

Figure 1.11.13. AIER Register, AIER2 Register and RMAD0 to RMAD3 Registers

Precautions for Interrupts

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

- Do not read the address 00000$_{16}$ in a program. When a maskable interrupt request is accepted, the CPU reads interrupt information (interrupt number and interrupt request priority level) from the address 00000$_{16}$ during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to “0”. If the address 00000$_{16}$ is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to “0”. This causes a problem that the interrupt is canceled, or an unexpected interrupt is generated.

(2) SP Setting

- Set any value in the SP before accepting an interrupt. The SP is cleared to ‘0000$_{16}$’ after reset. Therefore, if an interrupt is accepted before setting any value in the SP, the program may go out of control. Especially when using $\overline{\text{NMI}}$ interrupt, set a value in the SP at the beginning of the program. For the first and only the first instruction after reset, all interrupts including $\overline{\text{NMI}}$ interrupt are disabled.

(3) $\overline{\text{NMI}}$ Interrupt

- The $\overline{\text{NMI}}$ interrupt cannot be disabled. If this interrupt is unused, connect the $\overline{\text{NMI}}$ pin to V_{CC} via a resistor (pull-up).
- The input level of the $\overline{\text{NMI}}$ pin can be read by accessing the P8 register’s P8_5 bit. Note that the P8_5 bit can only be read when determining the pin level after an $\overline{\text{NMI}}$ interrupt is generated.
- Stop mode cannot be entered into while input on the $\overline{\text{NMI}}$ pin is low. This is because while input on the $\overline{\text{NMI}}$ pin is low the CM1 register’s CM10 bit is fixed to “0”.
- Do not go to wait mode while input on the $\overline{\text{NMI}}$ pin is low. This is because when input on the $\overline{\text{NMI}}$ pin goes low, the CPU stops but CPU clock remains active; therefore, the current consumption in the chip does not drop. In this case, normal condition is restored by an interrupt generated thereafter.
- The low and high level durations of the input signal to the $\overline{\text{NMI}}$ pin must each be 2 CPU clock cycles + 300 ns or more.

(4) $\overline{\text{INT}}$ Interrupt

- Either an “L” level or an “H” level of at least 250 ns width is necessary for the signal input to the $\overline{\text{INT}_0}$ through $\overline{\text{INT}_5}$ pins regardless of the CPU clock.
- When the polarity of the $\overline{\text{INT}_0}$ to $\overline{\text{INT}_5}$ pins is changed, the IR bit is sometimes set to “1” (=interrupt requested). After changing the polarity, set the IR bit to “0” (=interrupt not requested). Figure 1.11.13 shows the procedure for changing the $\overline{\text{INT}}$ interrupt generate factor.

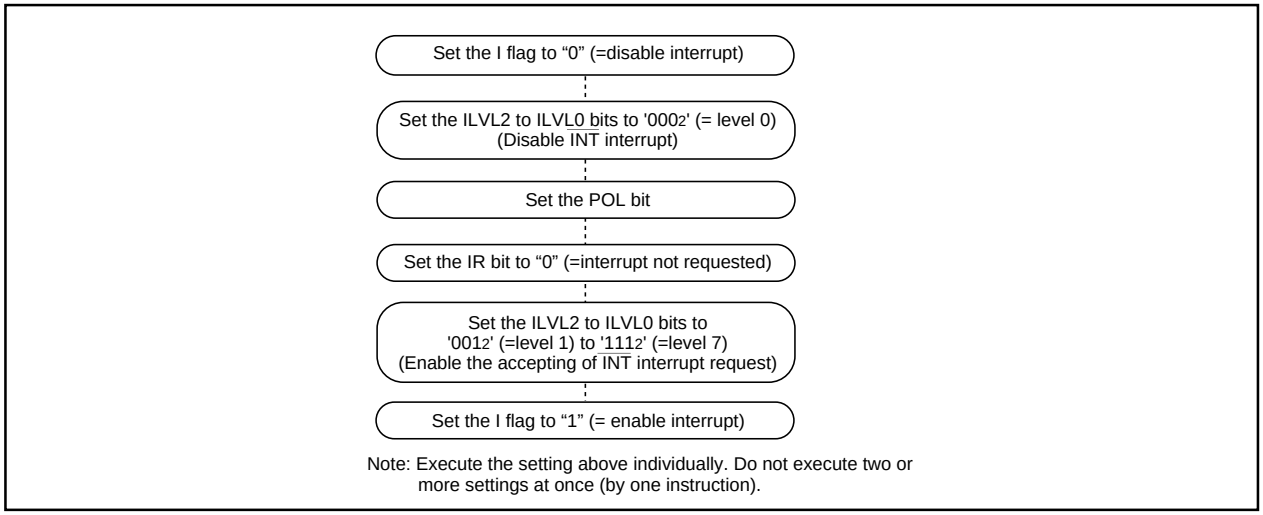


Figure 1.11.14. Switching Procedure for $\overline{\text{INT}}$ Interrupt Request

(5) Watchdog Timer Interrupt

• Initialize the watchdog timer after the watchdog timer interrupt occurs.

(6) Modifying Interrupt Control Register

• Each interrupt control register can only be modified while no interrupt requests corresponding to that register are generated. If interrupt requests managed by any interrupt control register are likely to occur, disable the interrupts before modifying the register. A sample program is shown below.

Example 1:

```
INT_SWITCH1:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Set the TA0IC register to "00₁₆".
    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     ; Set the TA0IC register to "00₁₆".
    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     ; Set the TA0IC register to "00₁₆".
    POPC    FLG             ; Enable interrupts.
```

Why the FSET I instruction is preceded by two NOP instructions (four when using HOLD function) in Example 1 and why the FSET I instruction is preceded by a dummy read in Example 2
This is to prevent the I flag from being set to "1" before writing to the interrupt control register for reasons of the instruction queue buffer.

To modify any interrupt control register after disabling interrupts, be careful with the instructions used.

Modifying other than the IR bit
If an interrupt request corresponding to that register is generated while executing the instruction, the IR bit may not be set to "1" (= interrupt requested), with the result that the interrupt request is ignored. If this presents a problem, use the following instructions to modify the register.
Instructions to use: AND, OR, BCLR, BSET

Modifying the IR bit
Even when the IR bit is cleared to "0" (= interrupt not requested), it may not actually be cleared to "0" depending on the instruction used. Therefore, use the MOV instruction to clear the IR bit.

Watchdog Timer

The watchdog timer is 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 contains a 15-bit counter which counts down the clock derived by dividing the CPU clock using the prescaler. Whether to generate a watchdog timer interrupt request or apply a watchdog timer reset as an operation to be performed when the watchdog timer underflows after reaching the terminal count can be selected using the PM12 bit of PM1 register. The PM12 bit can only be set to “1” (reset). Once this bit is set to “1”, it cannot be set to “0” (watchdog timer interrupt) in a program.
The pin, CPU and SFR initialized where the monitor timer underflows when the PM12 bit is “1” are the same as in software reset.
When the main clock is selected for CPU clock, the divide-by-N value for the prescaler can be chosen to be 16 or 128. If a sub-clock is selected for CPU clock, the divide-by-N value for the prescaler is always 2 no matter how the WDC7 bit is set. The period of watchdog timer can be calculated as given below. The period of watchdog timer is, however, subject to an error due to the prescaler.

With main clock chosen for CPU clock

$$\text{Watchdog timer period} = \frac{\text{Prescaler dividing (16 or 128) X Watchdog timer count (32768)}}{\text{CPU clock}}$$

With sub-clock chosen for CPU clock

$$\text{Watchdog timer period} = \frac{\text{Prescaler dividing (2) X Watchdog timer count (32768)}}{\text{CPU clock}}$$

For example, when CPU clock = 16 MHz and the divide-by-N value for the prescaler= 16, the watchdog timer period is approx. 32.8 ms.

The watchdog timer is initialized by writing to the WDTS register. The prescaler is initialized after reset. Note that the watchdog timer and the prescaler both are inactive after reset, so that the watchdog timer is activated to start counting by writing to the WDTS register.
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.
Figure 1.12.1 shows the block diagram of the watchdog timer. Figure 1.12.2 shows the watchdog timer-related registers.

• Count source protective mode
In this mode, a ring oscillator clock is used for the watchdog timer count source. The watchdog timer can be kept being clocked even when CPU clock stops as a result of run-away.
Before this mode can be used, the following register settings are required:
(1) Set the PRC1 bit of PRCR register to “1” (enable writes to PM1 and PM2 registers).
(2) Set the PM12 bit of PM1 register to “1” (reset when the watchdog timer underflows).
(3) Set the PM22 bit of PM2 register to “1” (ring oscillator clock used for the watchdog timer count source).
(4) Set the PRC1 bit of PRCR register to “0” (disable writes to PM1 and PM2 registers).
(5) Write to the WDTS register (watchdog timer starts counting).

Setting the PM22 bit to “1” results in the following conditions

• The ring oscillator starts oscillating, and the ring oscillator clock becomes the watchdog timer count source.

• The CM10 bit of CM1 register is disabled against write. (Writing a “1” has no effect, nor is stop mode entered.)

• The watchdog timer does not stop when in wait mode or hold state.

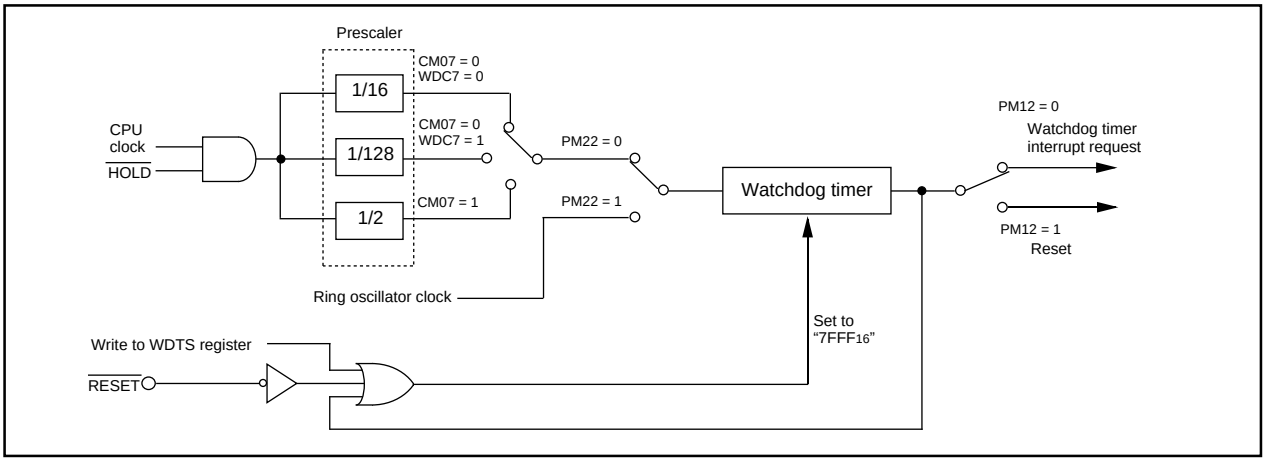


Figure 1.12.1. Watchdog Timer Block Diagram

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0
	0						

Symbol	Address	After reset
WDC	000F16	00XXXXXX2(Note2)

Bit symbol	Bit name	Function	RW
$\overline{\text{(b4-b0)}}$	High-order bit of watchdog timer		RO
WDC5	Cold start / warm start discrimination flag (Note 1)	0 : Cold start 1 : Warm start	RW
$\overline{\text{(b6)}}$	Reserved bit	Must set to “0”	RW
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	RW

Note 1: The WDC5 bit is always “1” (warm start) no matter how it is set by writing a “0” or “1”.
Note 2: The WDC5 bit is “0” (cold start) immediately after power-on. It can only be set to “1” in a program. It is set to “0” when the input voltage at the V_{CC1} pin drops to V_{det2} or less while the VC25 bit in the VCR2 register is set to “1” (RAM retention limit detection circuit enable).

Watchdog timer start register (Note)

b7	b0

Symbol	Address	After reset
WDTS	000E16	Indeterminate

Function	RW
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.	WO

Note : Write to the WDTS register after the watchdog timer interrupt occurs.

Figure 1.12.2. WDC Register and WDTS Register

DMAC

The DMAC (Direct Memory Access Controller) allows data to be transferred without the CPU intervention. Two DMAC channels are included. Each time a DMA request occurs, the DMAC transfers one (8 or 16-bit) data from the source address to the destination address. The DMAC uses the same data bus as used by the CPU. Because the DMAC has higher priority of bus control than the CPU and because it makes use of a cycle steal method, it can transfer one word (16 bits) or one byte (8 bits) of data within a very short time after a DMA request is generated. 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 DMAC-related registers.

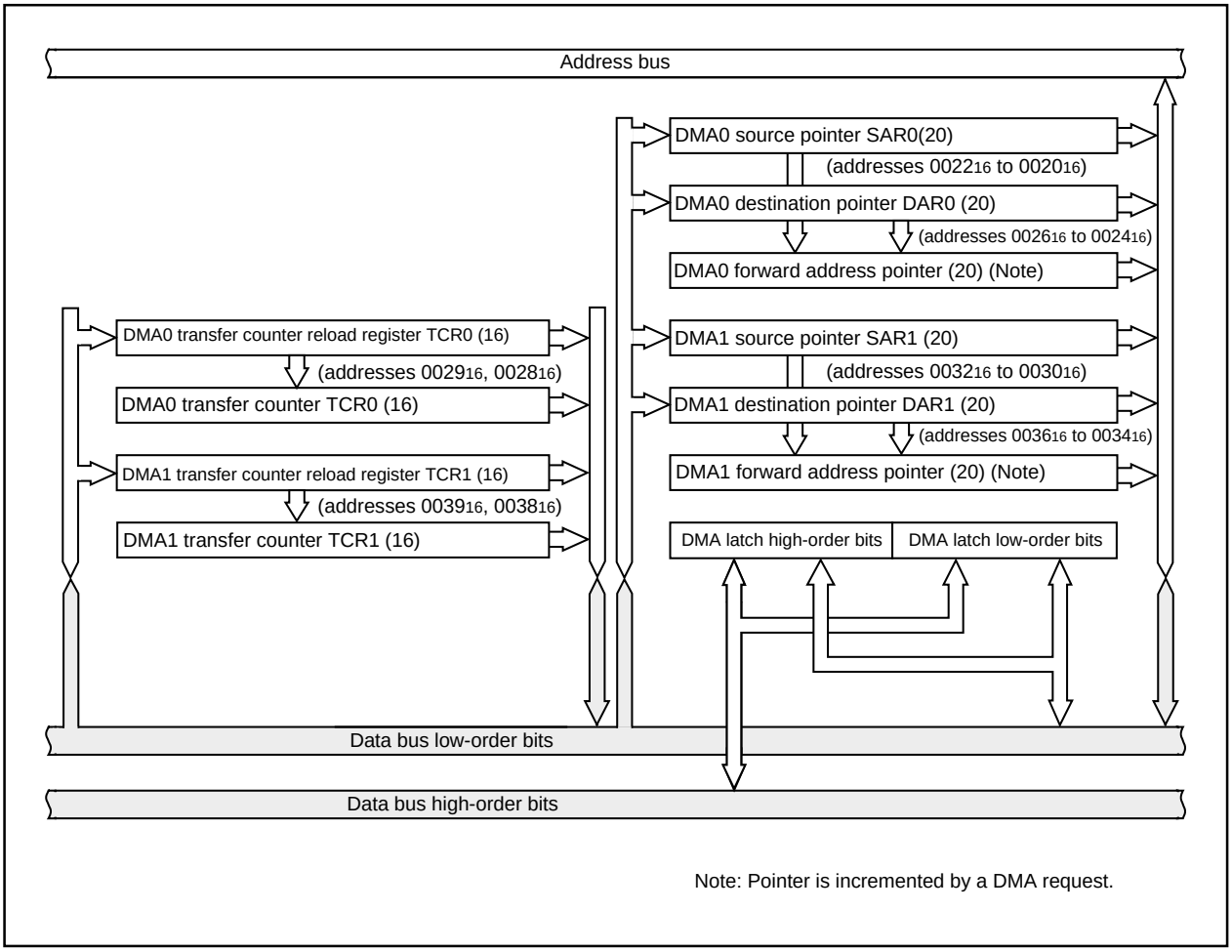


Figure 1.13.1. DMAC Block Diagram

A DMA request is generated by a write to the DMiSL register (i = 0–1)'s DSR bit, as well as by an interrupt request which is generated by any function specified by the DMiSL register's DMS and DSEL3–DSEL0 bits. However, unlike in the case of interrupt requests, DMA requests are not affected by the I flag and the interrupt control register, so that even when interrupt requests are disabled and no interrupt request can be accepted, DMA requests are always accepted. Furthermore, because the DMAC does not affect interrupts, the interrupt control register's IR bit does not change state due to a DMA transfer.

A data transfer is initiated each time a DMA request is generated when the DMiCON register's DMAE bit = "1" (DMA enabled). However, if the cycle in which a DMA request is generated is faster than the DMA transfer cycle, the number of transfer requests generated and the number of times data is transferred may not match. For details, refer to "DMA Requests".

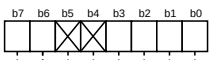
Table 1.13.1. DMAC Specifications

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
Maximum No. of bytes transferred		128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers)
DMA request factors (Note 1, Note 2)		Falling edge of $\overline{INT0}$ or $\overline{INT1}$ Both edge of $\overline{INT0}$ or $\overline{INT1}$ Timer A0 to timer A4 interrupt requests Timer B0 to timer B5 interrupt requests UART0 transfer, UART0 reception interrupt requests UART1 transfer, UART1 reception interrupt requests UART2 transfer, UART2 reception interrupt requests SI/O3, SI/O4 interrpt requests A-D conversion interrupt requests Software triggers
Channel priority		DMA0 > DMA1 (DMA0 takes precedence)
Transfer unit		8 bits or 16 bits
Transfer address direction		forward or fixed (The source and destination addresses cannot both be in the forward direction.)
Transfer mode	•Single transfer	Transfer is completed when the DMAi transfer counter (i = 0–1) underflows after reaching the terminal count.
	•Repeat transfer	When the DMAi transfer counter underflows, it is reloaded with the value of the DMAi transfer counter reload register and a DMA transfer is con tinued with it.
DMA interrupt request generation timing		When the DMAi transfer counter underflowed
DMA startup		Data transfer is initiated each time a DMA request is generated when the DMAiCON register's DMAE bit = "1" (enabled).
DMA shutdown	•Single transfer	• When the DMAE bit is set to "0" (disabled) • After the DMAi transfer counter underflows
	•Repeat transfer	When the DMAE bit is set to "0" (disabled)
Reload timing for forward address pointer and transfer counter		When a data transfer is started after setting the DMAE bit to "1" (en abled), the forward address pointer is reloaded with the value of the SARi or the DARi pointer whichever is specified to be in the forward direction and the DMAi transfer counter is reloaded with the value of the DMAi transfer counter reload register.

Notes:

1. DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the I flag nor by the interrupt control register.
2. The selectable causes of DMA requests differ with each channel.
3. Make sure that no DMAC-related registers (addresses 0020_{16}–$003F_{16}$) are accessed by the DMAC.

DMA0 request cause select register



Symbol: DM0SL | Address: 03B8$_{16}$ | After reset: 00$_{16}$

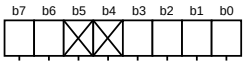
Bit symbol	Bit name	Function	RW
DSEL0	DMA request cause select bit	Refer to note	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
— (b5-b4)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—
DMS	DMA request cause expansion select bit	0: Basic cause of request 1: Extended cause of request	RW
DSR	Software DMA request bit	A DMA request is generated by setting this bit to “1” when the DMS bit is “0” (basic cause) and the DSEL3 to DSEL0 bits are “0001$_2$” (software trigger). The value of this bit when read is “0” .	RW

Note: The causes of DMA0 requests can be selected by a combination of DMS bit and DSEL3 to DSEL0 bits in the manner described below.

DSEL3 to DSEL0	DMS=0(basic cause of request)	DMS=1(extended cause of request)
0 0 0 0$_2$	Falling edge of INT0 pin	–
0 0 0 1$_2$	Software trigger	–
0 0 1 0$_2$	Timer A0	–
0 0 1 1$_2$	Timer A1	–
0 1 0 0$_2$	Timer A2	–
0 1 0 1$_2$	Timer A3	–
0 1 1 0$_2$	Timer A4	Two edges of INT0 pin
0 1 1 1$_2$	Timer B0	Timer B3
1 0 0 0$_2$	Timer B1	Timer B4
1 0 0 1$_2$	Timer B2	Timer B5
1 0 1 0$_2$	UART0 transmit	–
1 0 1 1$_2$	UART0 receive	–
1 1 0 0$_2$	UART2 transmit	–
1 1 0 1$_2$	UART2 receive	–
1 1 1 0$_2$	A-D conversion	–
1 1 1 1$_2$	UART1 transmit	–

Figure 1.13.2. DM0SL Register

DMA1 request cause select register



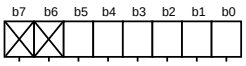
Symbol	Address	After reset
DM1SL	03BA16	0016

Bit symbol	Bit name	Function	RW
DSEL0	DMA request cause select bit	Refer to note	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
— (b5-b4)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—
DMS	DMA request cause expansion select bit	0: Basic cause of request 1: Extended cause of request	RW
DSR	Software DMA request bit	A DMA request is generated by setting this bit to “1” when the DMS bit is “0” (basic cause) and the DSEL3 to DSEL0 bits are “00012” (software trigger). The value of this bit when read is “0” .	RW

Note: The causes of DMA1 requests can be selected by a combination of DMS bit and DSEL3 to DSEL0 bits in the manner described below.

DSEL3 to DSEL0	DMS=0(basic cause of request)	DMS=1(extended cause of request)
0 0 0 02	Falling edge of INT1 pin	–
0 0 0 12	Software trigger	–
0 0 1 02	Timer A0	–
0 0 1 12	Timer A1	–
0 1 0 02	Timer A2	–
0 1 0 12	Timer A3	SI/O3
0 1 1 02	Timer A4	SI/O4
0 1 1 12	Timer B0	Two edges of INT1
1 0 0 02	Timer B1	–
1 0 0 12	Timer B2	–
1 0 1 02	UART0 transmit	–
1 0 1 12	UART0 receive/ACK0	–
1 1 0 02	UART2 transmit	–
1 1 0 12	UART2 receive/ACK2	–
1 1 1 02	A-D conversion	–
1 1 1 12	UART1 receive/ACK1	–

DMAi control register(i=0,1)



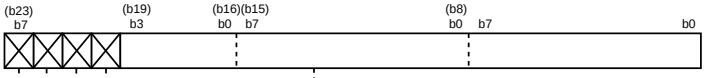
Symbol	Address	After reset
DM0CON	002C16	00000X002
DM1CON	003C16	00000X002

Bit symbol	Bit name	Function	RW
DMBIT	Transfer unit bit select bit	0 : 16 bits 1 : 8 bits	RW
DMASL	Repeat transfer mode select bit	0 : Single transfer 1 : Repeat transfer	RW
DMAS	DMA request bit	0 : DMA not requested 1 : DMA requested	RW (Note 1)
DMAE	DMA enable bit	0 : Disabled 1 : Enabled	RW
DSD	Source address direction select bit (Note 2)	0 : Fixed 1 : Forward	RW
DAD	Destination address direction select bit (Note 2)	0 : Fixed 1 : Forward	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

Note 1: The DMAS bit can be set to “0” by writing “0” in a program (This bit remains unchanged even if “1” is written).
Note 2: At least one of the DAD and DSD bits must be “0” (address direction fixed).

Figure 1.13.3. DM1SL Register, DM0CON Register, and DM1CON Registers

DMAi source pointer (i = 0, 1) (Note)

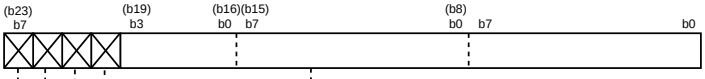


Symbol	Address	After reset
SAR0	0022_{16} to 0020_{16}	Indeterminate
SAR1	0032_{16} to 0030_{16}	Indeterminate

Function	Setting range	RW
Set the source address of transfer	00000_{16} to $FFFFF_{16}$	RW
Nothing is assigned. When write, set "0". When read, these contents are "0".		—

Note: If the DSD bit of DMiCON register is "0" (fixed), this register can only be written to when the DMAE bit of DMiCON register is "0" (DMA disabled).
If the DSD bit is "1" (forward direction), this register can be written to at any time.
If the DSD bit is "1" and the DMAE bit is "1" (DMA enabled), the DMAi forward address pointer can be read from this register. Otherwise, the value written to it can be read.

DMAi destination pointer (i = 0, 1)(Note)

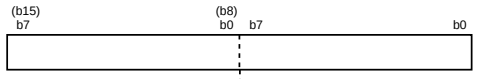


Symbol	Address	After reset
DAR0	0026_{16} to 0024_{16}	Indeterminate
DAR1	0036_{16} to 0034_{16}	Indeterminate

Function	Setting range	RW
Set the destination address of transfer	00000_{16} to $FFFFF_{16}$	RW
Nothing is assigned. When write, set "0". When read, these contents are "0".		—

Note: If the DAD bit of DMiCON register is "0" (fixed), this register can only be written to when the DMAE bit of DMiCON register is "0"(DMA disabled).
If the DAD bit is "1" (forward direction), this register can be written to at any time.
If the DAD bit is "1" and the DMAE bit is "1" (DMA enabled), the DMAi forward address pointer can be read from this register. Otherwise, the value written to it can be read.

DMAi transfer counter (i = 0, 1)



Symbol	Address	After reset
TCR0	0029_{16}, 0028_{16}	Indeterminate
TCR1	0039_{16}, 0038_{16}	Indeterminate

Function	Setting range	RW
Set the transfer count minus 1. The written value is stored in the DMAi transfer counter reload register, and when the DMAE bit of DMiCON register is set to "1" (DMA enabled) or the DMAi transfer counter underflows when the DMASL bit of DMiCON register is "1" (repeat transfer), the value of the DMAi transfer counter reload register is transferred to the DMAi transfer counter. When read, the DMAi transfer counter is read.	0000_{16} to $FFFF_{16}$	RW

Figure 1.13.4. SAR0, SAR1, DAR0, DAR1, TCR0, and TCR1 Registers

1. Transfer Cycles

The transfer cycle consists of a memory or SFR read (source read) bus cycle and a write (destination write) bus cycle. The number of read and write bus cycles is affected by the source and destination addresses of transfer. During memory extension and microprocessor modes, it is also affected by the BYTE pin level. Furthermore, the bus cycle itself is extended by a software wait or $\overline{\text{RDY}}$ signal.

(a) Effect of Source and Destination Addresses

If the transfer unit and data bus both are 16 bits and the source address of transfer begins with an odd address, the source read cycle consists of one more bus cycle than when the source address of transfer begins with an even address.

Similarly, if the transfer unit and data bus both are 16 bits and the destination address of transfer begins with an odd address, the destination write cycle consists of one more bus cycle than when the destination address of transfer begins with an even address.

(b) Effect of BYTE Pin Level

During memory extension and microprocessor modes, if 16 bits of data are to be transferred on an 8-bit data bus (input on the BYTE pin = high), the operation is accomplished by transferring 8 bits of data twice. Therefore, this operation requires two bus cycles to read data and two bus cycles to write data. Furthermore, if the DMAC is to access the internal area (internal ROM, internal RAM, or SFR), unlike in the case of the CPU, the DMAC does it through the data bus width selected by the BYTE pin.

(c) Effect of Software Wait

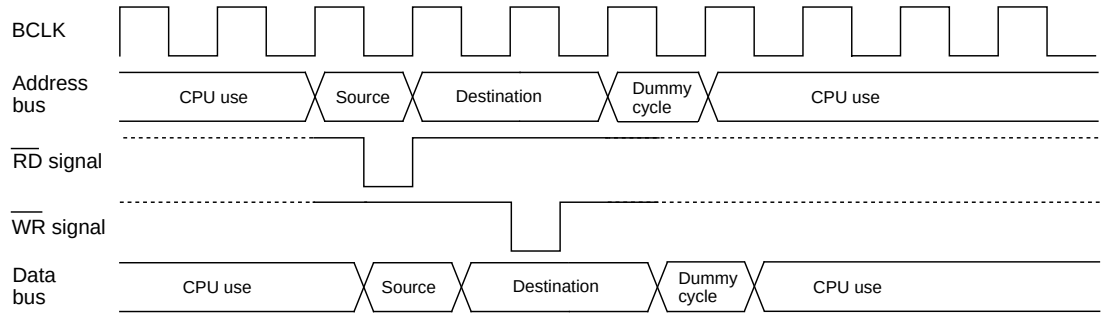
For memory or SFR accesses in which one or more software wait states are inserted, the number of bus cycles required for that access increases by an amount equal to software wait states.

(d) Effect of $\overline{\text{RDY}}$ Signal

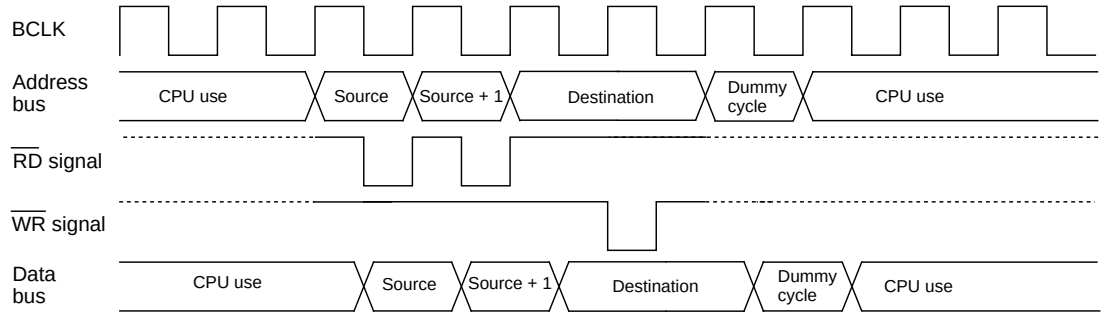
During memory extension and microprocessor modes, DMA transfers to and from an external area are affected by the $\overline{\text{RDY}}$ signal. Refer to "$\overline{\text{RDY}}$ signal".

Figure 1.13.5 shows the example of the 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 transfer cycles, take into consideration each condition for the source read and the destination write cycle, respectively. For example, when data is transferred in 16 bit units using an 8-bit bus ((2) in Figure 1.13.5), two source read bus cycles and two destination write bus cycles are required.

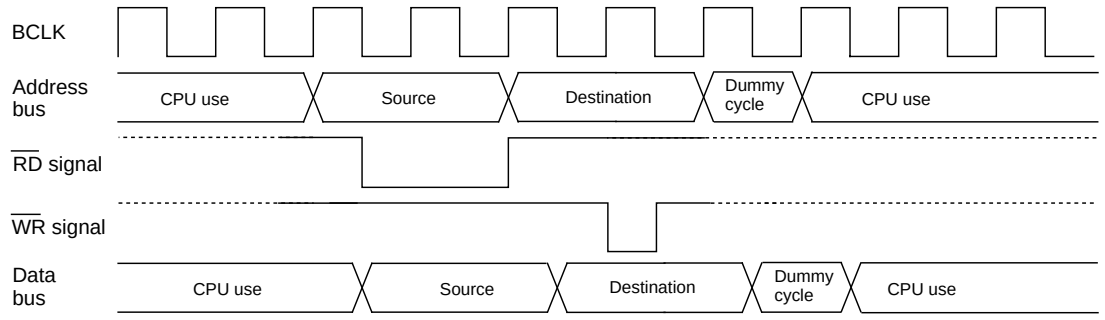
(1) When the transfer unit is 8 or 16 bits and the source of transfer is an even address



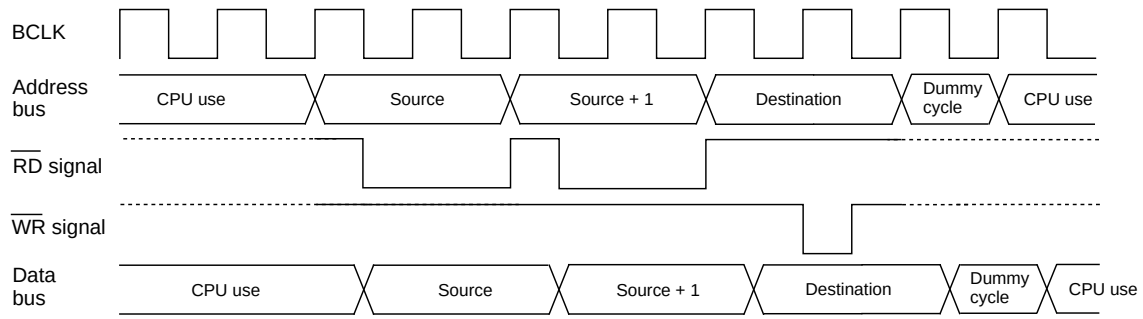
(2) When the transfer unit is 16 bits and the source address of transfer is an odd address, or when the transfer unit is 16 bits and an 8-bit bus is used



(3) When the source read cycle under condition (1) has one wait state inserted



(4) When the source read cycle under condition (2) has one wait state inserted



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

Figure 1.13.5. Transfer Cycles for Source Read

2. DMA Transfer Cycles

Any combination of even or odd transfer read and write addresses is possible. Table 1.13.2 shows the number of DMA transfer cycles. Table 1.13.3 shows the Coefficient j, k.

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. DMA 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

Table 1.13.3. Coefficient j, k

	Internal area				External area						
	Internal ROM, RAM		SFR		Separate bus				Multiplex bus		
	No wait	With wait	1-wait[2]	2-wait[2]	No wait	With wait[1]			With wait[1]		
						1 wait	2 waits	3 waits	1wait	2 waits	3 waits
j	1	2	2	3	1	2	3	4	3	3	4
k	1	2	2	3	2	2	3	4	3	3	4

Notes:
1. Depends on the set value of CSE register.
2. Depends on the set value of PM20 bit in PM2 register.

3. DMA Enable

When a data transfer starts after setting the DMAE bit in DMiCON register (i = 0, 1) to "1" (enabled), the DMAC operates as follows:

(1) Reload the forward address pointer with the SARi register value when the DSD bit in DMiCON register is "1" (forward) or the DARi register value when the DAD bit of DMiCON register is "1" (forward).

(2) Reload the DMAi transfer counter with the DMAi transfer counter reload register value.

If the DMAE bit is set to "1" again while it remains set, the DMAC performs the above operation. However, if a DMA request may occur simultaneously when the DMAE bit is being written, follow the steps below.

Step 1: Write "1" to the DMAE bit and DMAS bit in DMiCON register simultaneously.

Step 2: Make sure that the DMAi is in an initial state as described above (1) and (2) in a program.

If the DMAi is not in an initial state, the above steps should be repeated.

4. DMA Request

The DMAC can generate a DMA request as triggered by the cause of request that is selected with the DMS and DSEL3 to DSEL0 bits of DMiSL register (i = 0, 1) on either channel. Table 1.13.4 shows the timing at which the DMAS bit changes state.

Whenever a DMA request is generated, the DMAS bit is set to "1" (DMA requested) regardless of whether or not the DMAE bit is set. If the DMAE bit was set to "1" (enabled) when this occurred, the DMAS bit is set to "0" (DMA not requested) immediately before a data transfer starts. This bit cannot be set to "1" in a program (it can only be set to "0").

The DMAS bit may be set to "1" when the DMS or the DSEL3 to DSEL0 bits change state. Therefore, always be sure to set the DMAS bit to "0" after changing the DMS or the DSEL3 to DSEL0 bits.

Because if the DMAE bit is "1", a data transfer starts immediately after a DMA request is generated, the DMAS bit in almost all cases is "0" when read in a program. Read the DMAE bit to determine whether the DMAC is enabled.

Table 1.13.4. Timing at Which the DMAS Bit Changes State

DMA factor	DMAS bit of the DMiCON register	
	Timing at which the bit is set to "1"	Timing at which the bit is set to "0"
Software trigger	When the DSR bit of DMiCON register is set to "1"	• Immediately before a data transfer starts • When set by writing "0" in a program
Peripheral function	When the interrupt control register for the peripheral function that is selected by the DSEL3 to DSEL0 and DMS bits of DMiCON register has its IR bit set to "1"	

Channel Priority and DMA Transfer Timing

If both DMA0 and DMA1 are enabled and DMA transfer request signals from DMA0 and DMA1 are detected active in the same sampling period (one period from a falling edge to the next falling edge of BCLK), the DMAS bit on each channel is set to “1” (DMA requested) at the same time. In this case, the DMA requests are arbitrated according to the channel priority, DMA0 > DMA1. The following describes DMAC operation when DMA0 and DMA1 requests are detected active in the same sampling period. Figure 1.13.6 shows an example of DMA transfer effected by external factors.

In Figure 1.13.6, because DMA0 and DMA1 requests occurred at the same time, DMA0 which has higher channel priority is accepted first and a DMA transfer on it starts. When DMA0 finishes one transfer unit, it relinquishes control of the bus to the CPU, and when the CPU finishes one bus access, DMA1 starts a transfer next and after completion of one transfer unit, returns control of the bus to the CPU.

Note that because there is only one DMAS bit on each channel, the number of times DMA is requested cannot be counted. Therefore, even if multiple DMA requests occurred before gaining control of the bus as in the case of DMA1 in Figure 1.13.6, the DMAS bit is set to “0” when control of the bus is gained and after completion of one transfer unit, control of the bus is returned to the CPU.

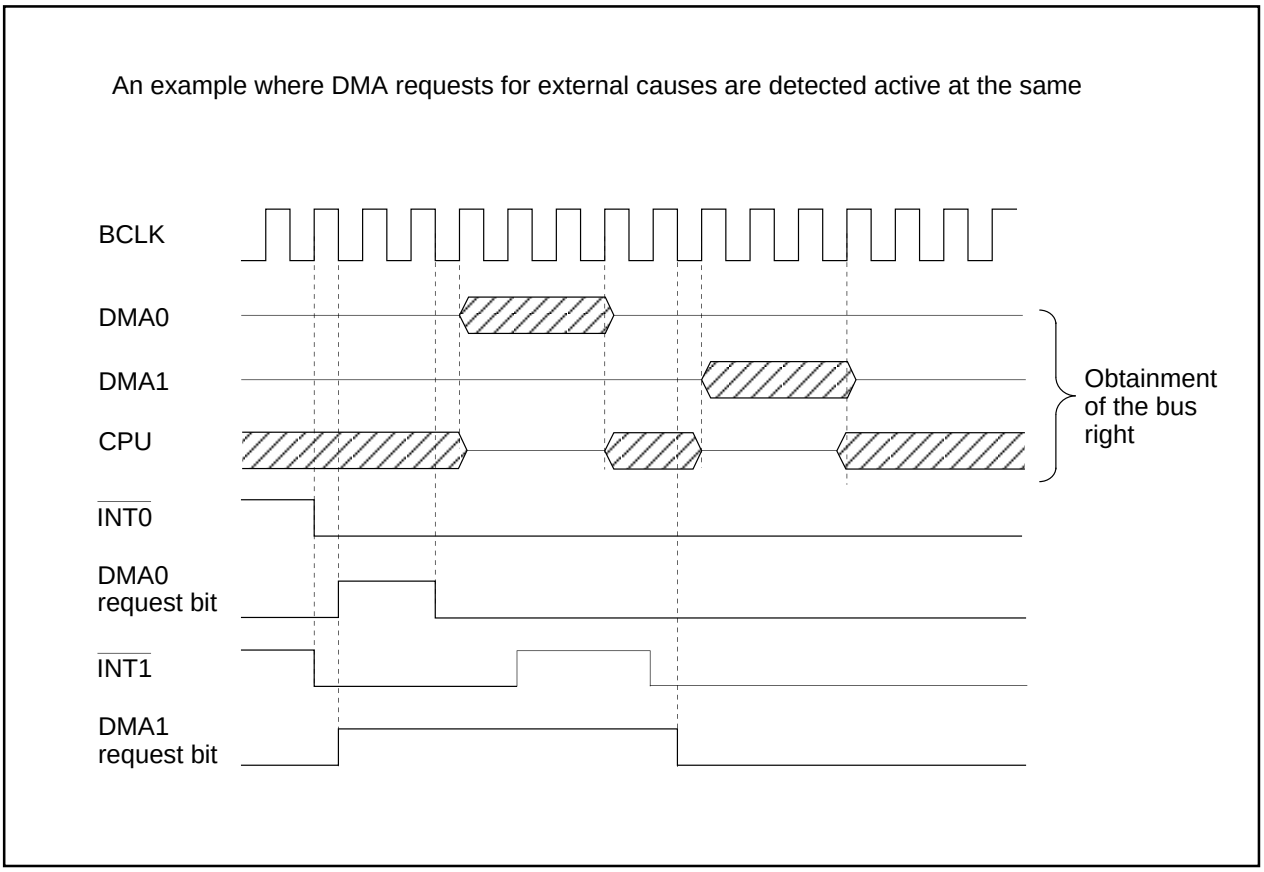
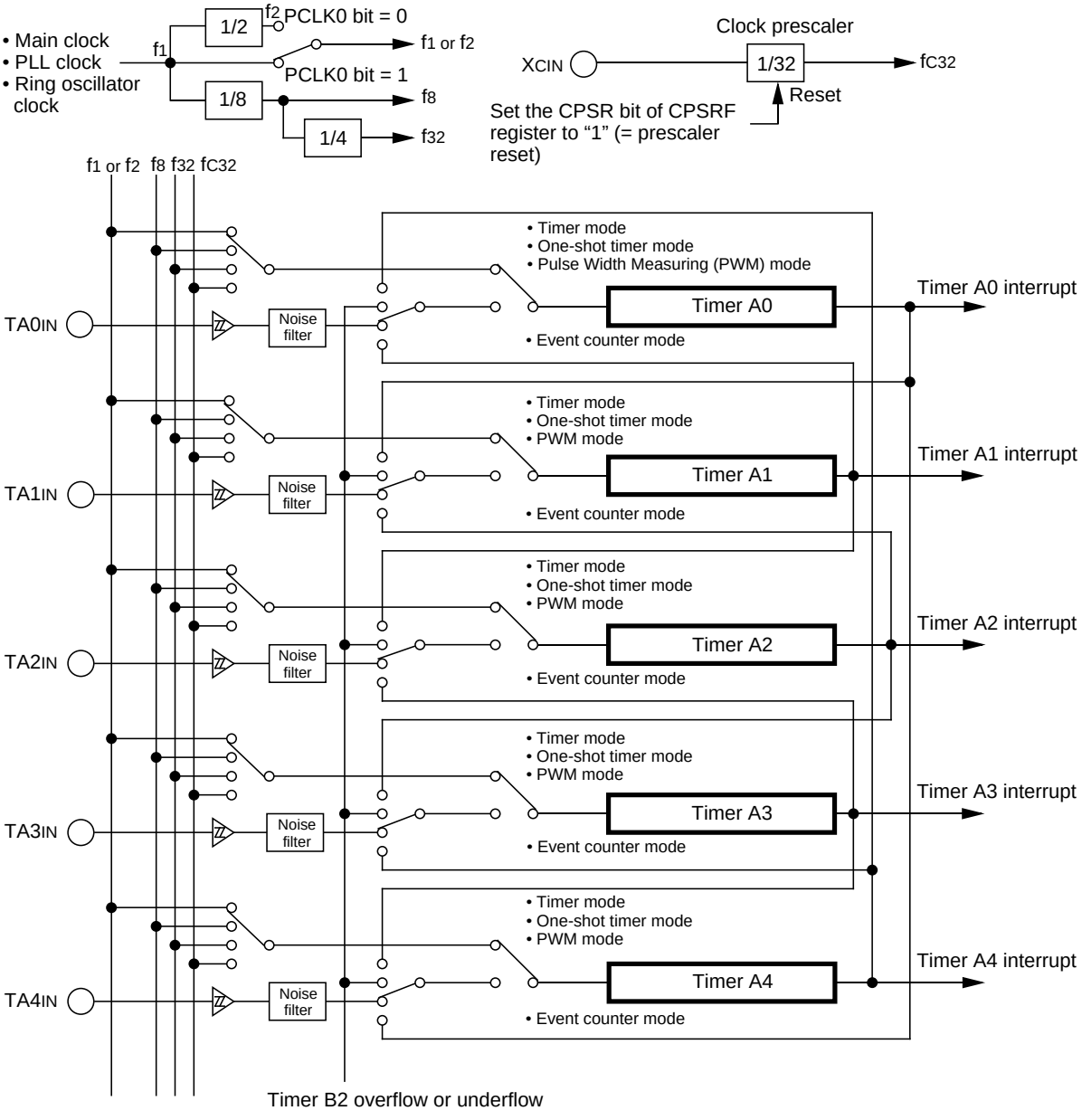


Figure 1.13.6. DMA Transfer by External Factors

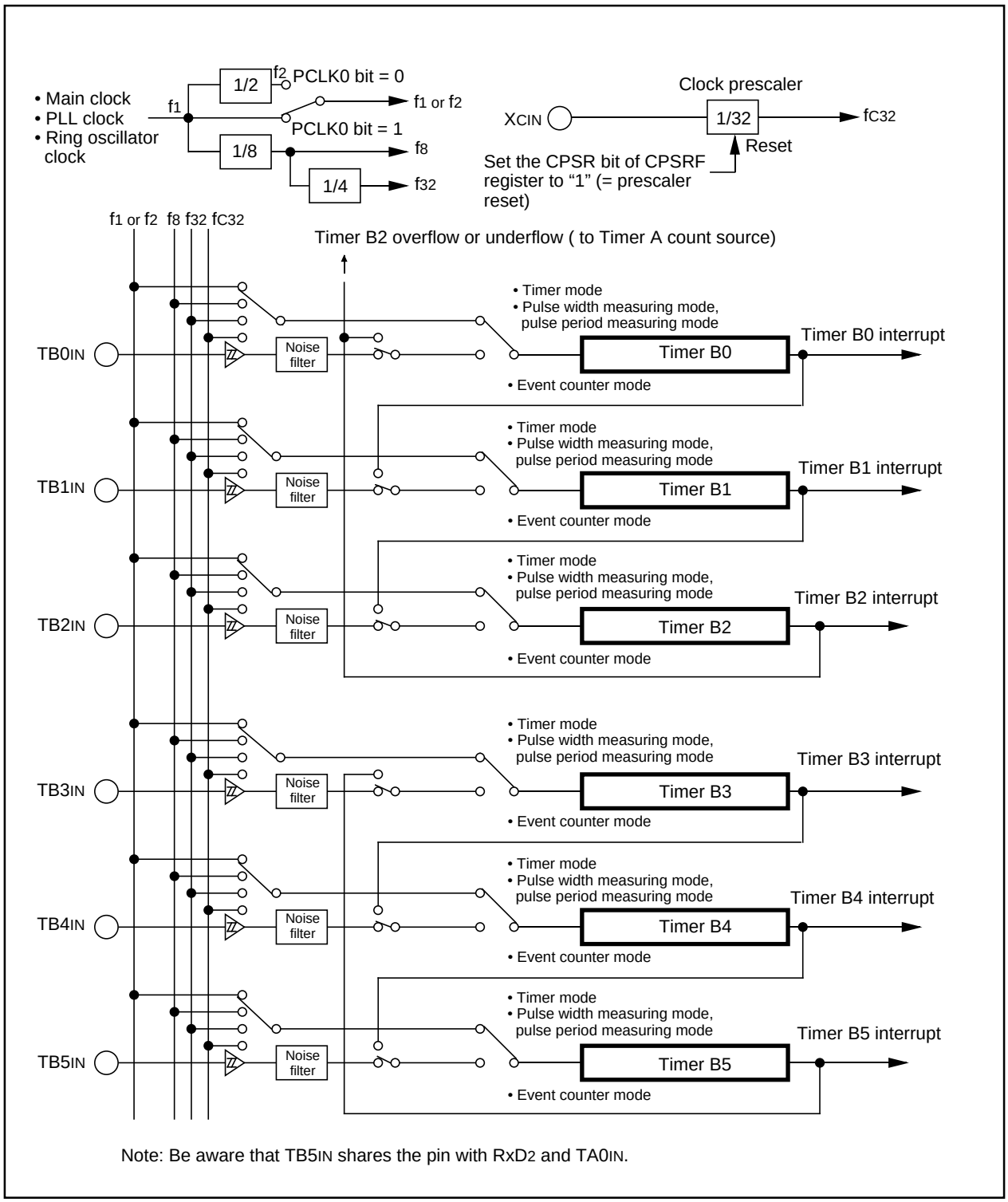
Timers

Eleven 16-bit timers, each capable of operating independently of the others, can be classified by function as either timer A (five) and timer B (six). The count source for each timer acts as a clock, to control such timer operations as counting, reloading, etc. Figures 1.14.1 and 1.14.2 show block diagrams of timer A and timer B configuration, respectively.



Note: Be aware that TA0IN shares the pin with RxD2 and TB5IN.

Figure 1.14.1. Timer A Configuration



Note: Be aware that TB5IN shares the pin with RxD2 and TA0IN.

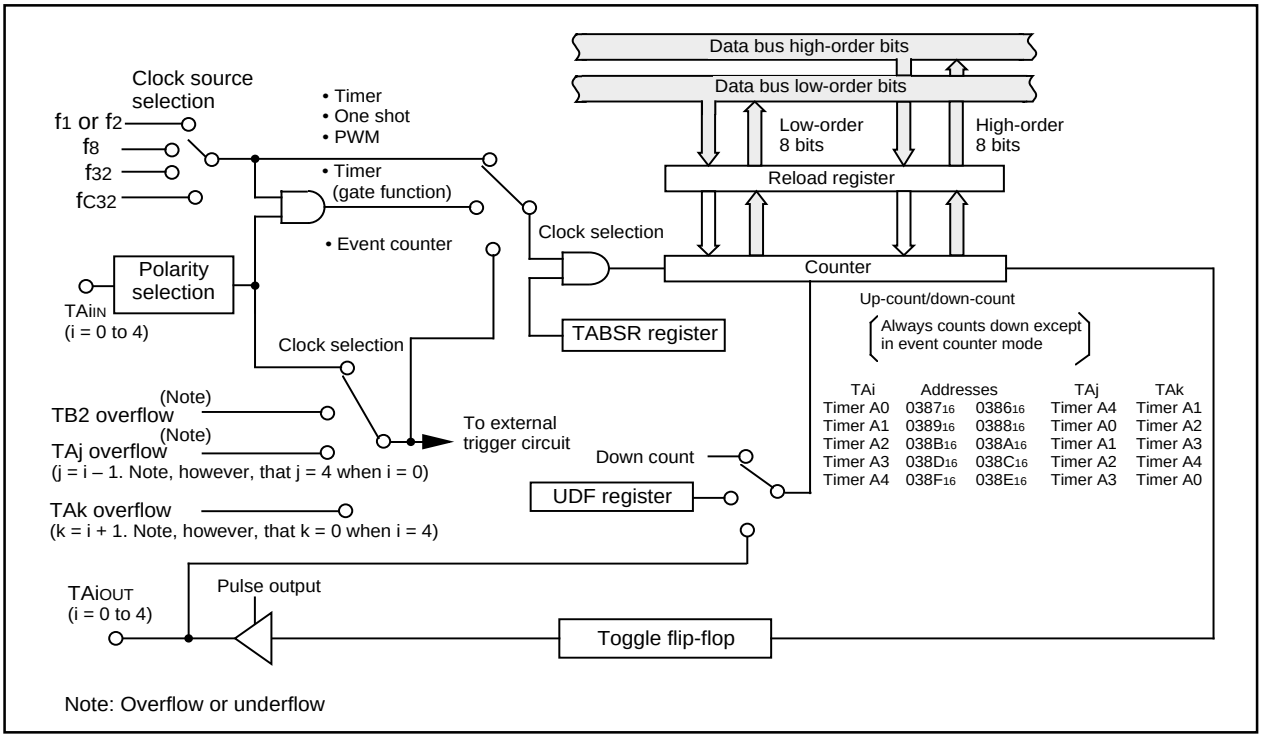
Figure 1.14.2. Timer B Configuration

Timer A

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

The timer A supports the following four modes. Except in event counter mode, timers A0 to A4 all have the same function. Use the TMOD1 to TMOD0 bits of TAiMR register (i = 0 to 4) to select the desired mode.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external device or overflows and underflows of other timers.
- One-shot timer mode: The timer outputs a pulse only once before it reaches the minimum count "0000_{16}."
- Pulse width modulation (PWM) mode: The timer outputs pulses in a given width successively.



TAi	Addresses		TAj	TAk
Timer A0	0387_{16}	0386_{16}	Timer A4	Timer A1
Timer A1	0389_{16}	0388_{16}	Timer A0	Timer A2
Timer A2	$038B_{16}$	$038A_{16}$	Timer A1	Timer A3
Timer A3	$038D_{16}$	$038C_{16}$	Timer A2	Timer A4
Timer A4	$038F_{16}$	$038E_{16}$	Timer A3	Timer A0

Figure 1.14.3. Timer A Block Diagram

Timer Ai mode register (i=0 to 4)

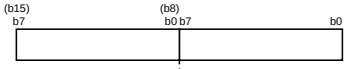
Symbol	Address	After reset
TA0MR to TA4MR	0396_{16} to $039A_{16}$	00_{16}

b7	b6	b5	b4	b3	b2	b1	b0

Bit symbol	Bit name	Function	RW
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	RW
TMOD1			RW
MR0	———	Function varies with each operation mode	RW
MR1			RW
MR2			RW
MR3			RW
TCK0	Count source select bit	Function varies with each operation mode	RW
TCK1			RW

Figure 1.14.4. TA0MR to TA4MR Registers

Timer Ai register (i= 0 to 4) (Note 1)



Symbol	Address	After reset
TA0	0387$_{16}$, 0386$_{16}$	Indeterminate
TA1	0389$_{16}$, 0388$_{16}$	Indeterminate
TA2	038B$_{16}$, 038A$_{16}$	Indeterminate
TA3	038D$_{16}$, 038C$_{16}$	Indeterminate
TA4	038F$_{16}$, 038E$_{16}$	Indeterminate

Mode	Function	Setting range	RW
Timer mode	Divide the count source by n + 1 where n = set value	0000$_{16}$ to FFFF$_{16}$	RW
Event counter mode	Divide the count source by FFFF$_{16}$ – n + 1 where n = set value when counting up or by n + 1 when counting down (Note 5)	0000$_{16}$ to FFFF$_{16}$	RW
One-shot timer mode	Divide the count source by n where n = set value and cause the timer to stop	0000$_{16}$ to FFFF$_{16}$ (Notes 2, 4)	WO
Pulse width modulation mode (16-bit PWM)	Modify the pulse width as follows: PWM period: (2^{16} – 1) / fj High level PWM pulse width: n / fj where n = set value, fj = count source frequency	0000$_{16}$ to FFFE$_{16}$ (Note 3, 4)	WO
Pulse width modulation mode (8-bit PWM)	Modify the pulse width as follows: PWM period: (2^8 – 1) x (m + 1)/ fj High level PWM pulse width: (m + 1)n / fj where n = high-order address set value, m = low-order address set value, fj = count source frequency	00$_{16}$ to FE$_{16}$ (High-order address) 00$_{16}$ to FF$_{16}$ (Low-order address) (Note 3, 4)	WO

Note 1: The register must be accessed in 16 bit units.
Note 2: If the TAi register is set to '0000$_{16}$,' the counter does not work and timer Ai interrupt requests are not generated either. Furthermore, if "pulse output" is selected, no pulses are output from the TAiOUT pin.
Note 3: If the TAi register is set to '0000$_{16}$,' the pulse width modulator does not work, the output level on the TAiOUT pin remains low, and timer Ai interrupt requests are not generated either. The same applies when the 8 high-order bits of the timer TAi register are set to '001 6' while operating as an 8-bit pulse width modulator.
Note 4: Use the MOV instruction to write to the TAi register.
Note 5: The timer counts pulses from an external device or overflows or underflows in other timers.

Count start flag

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

Bit symbol	Bit name	Function	RW
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	RW
TA1S	Timer A1 count start flag		RW
TA2S	Timer A2 count start flag		RW
TA3S	Timer A3 count start flag		RW
TA4S	Timer A4 count start flag		RW
TB0S	Timer B0 count start flag		RW
TB1S	Timer B1 count start flag		RW
TB2S	Timer B2 count start flag		RW

Up/down flag (Note 1)

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

Bit symbol	Bit name	Function	RW
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count Enabled by setting the TAiMR register's MR2 bit to "0" (= switching source in UDF register) during event counter mode.	RW
TA1UD	Timer A1 up/down flag		RW
TA2UD	Timer A2 up/down flag		RW
TA3UD	Timer A3 up/down flag		RW
TA4UD	Timer A4 up/down flag		RW
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled (Notes 2, 3)	WO
TA3P	Timer A3 two-phase pulse signal processing select bit		WO
TA4P	Timer A4 two-phase pulse signal processing select bit		WO

Note 1: Use MOV instruction to write to this register.
Note 2: Make sure the port direction bits for the TA2IN to TA4IN and TA2OUT to TA4OUT pins are set to "0" (input mode).
Note 3: When not using the two-phase pulse signal processing function, set the corresponding bit to "0".

Figure 1.14.5. TA0 to TA4 Registers, TABSR Register, and UDF Register

One-shot start flag

Symbol	Address	After reset
ONSF	0382_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TA0OS	Timer A0 one-shot start flag	The timer starts counting by setting this bit to "1" while the TMOD1 to TMOD0 bits of TAiMR register (i = 0 to 4) = '10_2' (= one-shot timer mode) and the MR2 bit of TAiMR register = "0" (=TAiOS bit enabled). When read, its content is "0".	RW
TA1OS	Timer A1 one-shot start flag		RW
TA2OS	Timer A2 one-shot start flag		RW
TA3OS	Timer A3 one-shot start flag		RW
TA4OS	Timer A4 one-shot start flag		RW
TA4OS	Z-phase input enable bit	0 : Z-phase input disabled 1 : Z-phase input enabled	RW
TA0TGL	Timer A0 event/trigger select bit	b7 b6 0 0 : Input on TA0IN is selected (Note 1) 0 1 : TB2 overflow is selected (Note 2) 1 0 : TA4 overflow is selected (Note 2) 1 1 : TA1 overflow is selected (Note 2)	RW
TA0TGH			RW

Note 1: Make sure the PD7_1 bit of PD7 register is set to "0" (= input mode).
Note 2: Overflow or underflow

Trigger select register

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

Bit symbol	Bit name	Function	RW
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on TA1IN is selected (Note 1) 0 1 : TB2 overflow is selected (Note 2) 1 0 : TA0 overflow is selected (Note 2) 1 1 : TA2 overflow is selected (Note 2)	RW
TA1TGH			RW
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on TA2IN is selected (Note 1) 0 1 : TB2 overflow is selected (Note 2) 1 0 : TA1 overflow is selected (Note 2) 1 1 : TA3 overflow is selected (Note 2)	RW
TA2TGH			RW
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on TA3IN is selected (Note 1) 0 1 : TB2 overflow is selected (Note 2) 1 0 : TA2 overflow is selected (Note 2) 1 1 : TA4 overflow is selected (Note 2)	RW
TA3TGH			RW
TA4TGL	Timer A4 event/trigger select bit	b7 b6 0 0 : Input on TA4IN is selected (Note 1) 0 1 : TB2 overflow is selected (Note 2) 1 0 : TA3 overflow is selected (Note 2) 1 1 : TA0 overflow is selected (Note 2)	RW
TA4TGH			RW

Note 1: Make sure the port direction bits for the TA1IN to TA4IN pins are set to "0" (= input mode).
Note 2: Overflow or underflow

Clock prescaler reset flag

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

Bit symbol	Bit name	Function	RW
— (b6-b0)	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.		—
CPSR	Clock prescaler reset flag	Setting this bit to "1" initializes the prescaler for the timekeeping clock. (When read, its content is "0".)	RW

Figure 1.14.6. ONSF Register, TRGSR Register, and CPSRF Register

1. Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 1.14.1). Figure 1.14.7 shows TAiMR register in timer mode.

Table 1.14.1. Specifications in Timer Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TAiMR register (i= 0 to 4) 0000_{16} to $FFFF_{16}$
Count start condition	Set TAiS bit of TABSR register to "1" (= start counting)
Count stop condition	Set TAiS bit to "0" (= stop counting)
Interrupt request generation timing	Timer underflow
TAiIN pin function	I/O port or gate input
TAiOUT pin function	I/O port or pulse output
Read from timer	Count value can be read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Gate function Counting can be started and stopped by an input signal to TAiIN pin • Pulse output function Whenever the timer underflows, the output polarity of TAiOUT pin is inverted. When not counting, the pin outputs a low.

Timer Ai mode register (i=0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol: TA0MR to TA4MR — Address: 0396_{16} to $039A_{16}$ — After reset: 00_{16}

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
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)	RW
MR1	Gate function select bit	b4 b3 0 0 : 0 1 : } Gate function not available (TAiIN pin functions as I/O port) 1 0 : Counts while input on the TAiIN pin is low (Note 2) 1 1 : Counts while input on the TAiIN pin is high (Note 2)	RW
MR2			RW
MR3	Must be set to "0" in timer mode		RW
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TCK1			RW

Note 1: TA0OUT pin is N-channel open drain output.
Note 2: The port direction bit for the TAiIN pin must be set to "0" (= input mode).

Figure 1.14.7. Timer Ai Mode Register in Timer Mode

2. Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Timers A2, A3 and A4 can count two-phase external signals. Table 1.14.2 lists specifications in event counter mode (when not processing two-phase pulse signal). Table 1.14.3 lists specifications in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4). Figure 1.14.8 shows TAiMR register in event counter mode (when not processing two-phase pulse signal). Figure 1.14.9 shows TA2MR to TA4MR registers in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4).

Table 1.14.2. Specifications in Event Counter Mode (when not processing two-phase pulse signal)

Item	Specification
Count source	• External signals input to TAiIN pin (i=0 to 4) (effective edge can be selected in program) • Timer B2 overflows or underflows, timer Aj (j=i-1, except j=4 if i=0) overflows or underflows, timer Ak (k=i+1, except k=0 if i=4) overflows or underflows
Count operation	• Up-count or down-count can be selected by external signal or program • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divided ratio	1/ ($FFFF_{16}$ - n + 1) for up-count 1/ (n + 1) for down-count n : set value of TAi register 0000_{16} to $FFFF_{16}$
Count start condition	Set TAiS bit of TABSR register to “1” (= start counting)
Count stop condition	Set TAiS bit to “0” (= stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	I/O port or count source input
TAiOUT pin function	I/O port, pulse output, or up/down-count select input
Read from timer	Count value can be read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
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 Whenever the timer underflows or underflows, the output polarity of TAiOUT pin is inverted . When not counting, the pin outputs a low.

Timer Ai mode register (i=0 to 4)
(When not using two-phase pulse signal processing)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

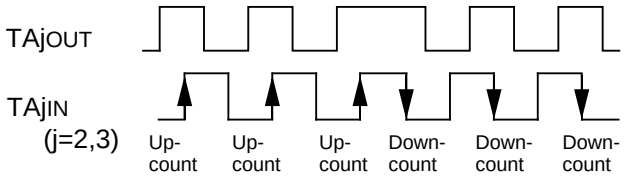
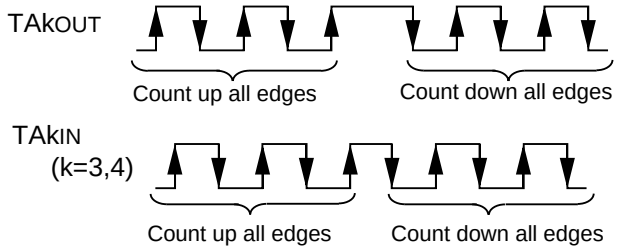
Symbol	Address	After reset
TA0MR to TA4MR	0396_{16} to $039A_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0	RW
TMOD1		0 1 : Event counter mode (Note 1)	RW
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin functions as I/O port) 1 : Pulse is output (Note 2) (TAiOUT pin functions as pulse output pin)	RW
MR1	Count polarity select bit (Note 3)	0 : Counts external signal's falling edge 1 : Counts external signal's rising edge	RW
MR2	Up/down switching cause select bit	0 : UDF register 1 : Input signal to TAiOUT pin (Note 4)	RW
MR3	Must be set to “0” in event counter mode		RW
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	RW
TCK1	Can be “0” or “1” when not using two-phase pulse signal processing		RW

Note 1: During event counter mode, the count source can be selected using the ONSF and TRGSR registers.
Note 2: TA0OUT pin is N-channel open drain output.
Note 3: Effective when the TAiGH and TAiGL bits of ONSF or TRGSR register are ‘00_2’ (TAiIN pin input).
Note 4: Count down when input on TAiOUT pin is low or count up when input on that pin is high. The port direction bit for TAiOUT pin must be set to “0” (= input mode).

Figure 1.14.8. TAiMR Register in Event Counter Mode (when not using two-phase pulse signal processing)

Table 1.14.3. Specifications in Event Counter Mode (when processing two-phase pulse signal with timers A2, A3 and A4)

Item	Specification
Count source	• Two-phase pulse signals input to TAiIN or TAiOUT pins (i = 2 to 4)
Count operation	• Up-count or down-count can be selected by two-phase pulse signal • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divide ratio	1/ (FFFF16 - n + 1) for up-count 1/ (n + 1) for down-count n : set value of TAi register 000016 to FFFF16
Count start condition	Set TAiS bit of TABSR register to “1” (= start counting)
Count stop condition	Set TAiS bit to “0” (= stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	Count value can be read by reading timer A2, A3 or A4 register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to reload register (Transferred to counter when reloaded next)
Select function (Note)	• Normal processing operation (timer A2 and timer A3) The timer counts up rising edges or counts down falling edges on TAjIN pin when input signals on TAjOUT pin is “H”.  • Multiply-by-4 processing operation (timer A3 and timer A4) If the phase relationship is such that TAkIN(k=3, 4) pin goes “H” when the input signal on TAkOUT pin is “H”, the timer counts up rising and falling edges on TAkOUT and TAkIN pins. If the phase relationship is such that TAkIN pin goes “L” when the input signal on TAkOUT pin is “H”, the timer counts down rising and falling edges on TAkOUT and TAkIN pins.  • Counter initialization by Z-phase input (timer A3) The timer count value is initialized to 0 by Z-phase input.

Notes:
1. Only timer A3 is selectable. Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.

Timer Ai mode register (i=2 to 4)
(When using two-phase pulse signal processing)

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

Symbol: TA2MR to TA4MR
Address: 039816 to 039A16
After reset: 0016

	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
MR0	To use two-phase pulse signal processing, set this bit to “0”.		RW
MR1	To use two-phase pulse signal processing, set this bit to “0”.		RW
MR2	To use two-phase pulse signal processing, set this bit to “1”.		RW
MR3	To use two-phase pulse signal processing, set this bit to “0”.		RW
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	RW
TCK1	Two-phase pulse signal processing operation select bit (Note 1)(Note 2)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	RW

Note 1: TCK1 bit is valid for timer A3 mode register. No matter how this bit is set, timers A2 and A4 always operate in normal processing mode and x4 processing mode, respectively.
Note 2: If two-phase pulse signal processing is desired, following register settings are required:
- Set the UDF register’s TAiP bit to “1” (two-phase pulse signal processing function enabled).
- Set the TRGSR register’s TAiGH and TAiGL bits to ‘002’ (TAiIN pin input).
- Set the port direction bits for TAiIN and TAiOUT to “0” (input mode).

Figure 1.14.9. TA2MR to TA4MR Registers in Event Counter Mode (when using two-phase pulse signal processing with timer A2, A3 or A4)

• Counter Initialization by Two-Phase Pulse Signal Processing

This function initializes the timer count value to “0” by Z-phase (counter initialization) input during two-phase pulse signal processing.

This function can only be used in timer A3 event counter mode during two-phase pulse signal processing, free-running type, x4 processing, with Z-phase entered from the $\overline{INT2}$ pin.

Counter initialization by Z-phase input is enabled by writing “0000_{16}” to the TA3 register and setting the TAZIE bit in ONSF register to “1” (= Z-phase input enabled).

Counter initialization is accomplished by detecting Z-phase input edge. The active edge can be chosen to be the rising or falling edge by using the POL bit of INT2IC register. The Z-phase pulse width applied to the $\overline{INT2}$ pin must be equal to or greater than one clock cycle of the timer A3 count source.

The counter is initialized at the next count timing after recognizing Z-phase input. Figure 1.14.10 shows the relationship between the two-phase pulse (A phase and B phase) and the Z phase.

If timer A3 overflow or underflow coincides with the counter initialization by Z-phase input, a timer A3 interrupt request is generated twice in succession. Do not use the timer A3 interrupt when using this function.

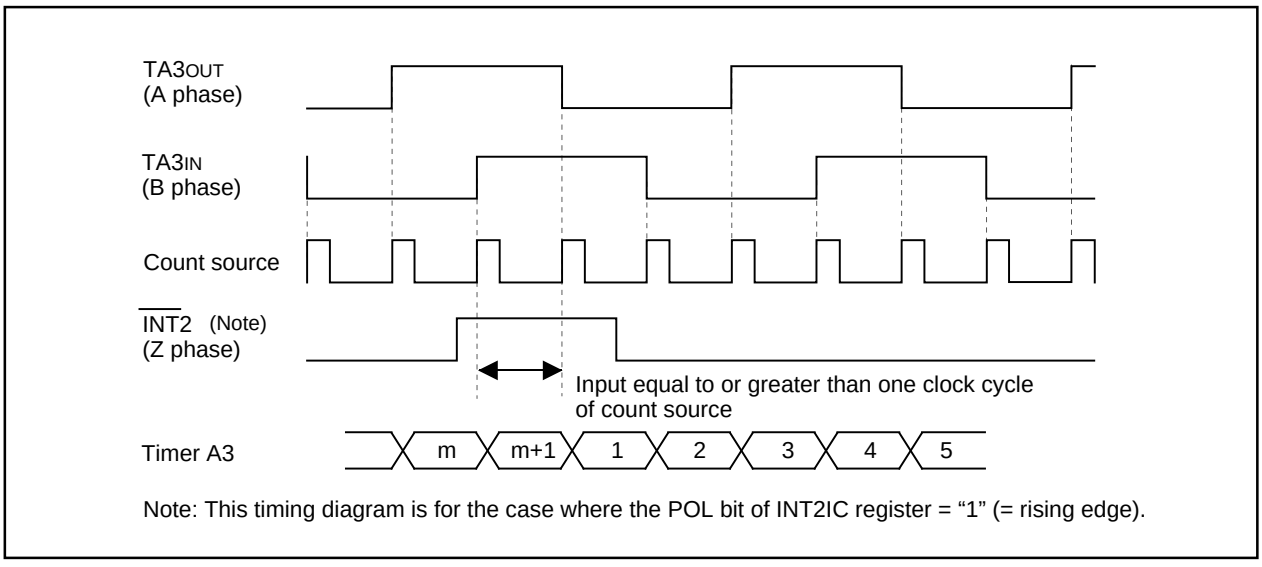


Figure 1.14.10. Two-phase Pulse (A phase and B phase) and the Z Phase

3. One-shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger. (See Table 1.14.4.) When the trigger occurs, the timer starts up and continues operating for a given period. Figure 1.14.12 shows the TAiMR register in one-shot timer mode.

Table 1.14.4. Specifications in One-shot Timer Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Down-count • When the counter reaches 0000_{16}, it stops counting after reloading a new value • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : set value of TAi register 0000_{16} to $FFFF_{16}$ However, the counter does not work if the divide-by-n value is set to 0000_{16}.
Count start condition	TAiS bit of TABSR register = "1" (start counting) and one of the following triggers occurs. • External trigger input from the TAi_{IN} pin • Timer B2 overflow or underflow, timer Aj (j=i-1, except j=4 if i=0) overflow or underflow, timer Ak (k=i+1, except k=0 if i=4) overflow or underflow • The TAiOS bit of ONSF register is set to "1" (= timer starts)
Count stop condition	• When the counter is reloaded after reaching "0000_{16}" • TAiS bit is set to "0" (= stop counting)
Interrupt request generation timing	When the counter reaches "0000_{16}"
TAi_{IN} pin function	I/O port or trigger input
TAi_{OUT} pin function	I/O port or pulse output
Read from timer	An indeterminate value is read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Pulse output function The timer outputs a low when not counting and a high when counting.

Timer Ai mode register (i=0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

Symbol	Address	After reset
TA0MR to TA4MR	396_{16} to $039A_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode	RW
TMOD1			RW
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin functions as I/O port) 1 : Pulse is output (Note 1) (TAiOUT pin functions as a pulse output pin)	RW
MR1	External trigger select bit (Note 2)	0 : Falling edge of input signal to TAiIN pin (Note 3) 1 : Rising edge of input signal to TAiIN pin (Note 3)	RW
MR2	Trigger select bit	0 : TAiOS bit is enabled 1 : Selected by TAiTGH to TAiTGL bits	RW
MR3	Must be set to “0” in one-shot timer mode		RW
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TCK1			RW

Note 1: TA0OUT pin is N-channel open drain output.
Note 2: Effective when the TAiGH and TAiGL bits of ONSF or TRGSR register are ‘00_2’ (TAiIN pin input).
Note 3: The port direction bit for the TAiIN pin must be set to “0” (= input mode).

Figure 1.14.12. TAiMR Register in One-shot Timer Mode

4. Pulse Width Modulation (PWM) Mode

In PWM mode, the timer outputs pulses of a given width in succession (see Table 1.14.5). The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator. Figure 1.14.13 shows TAiMR register in pulse width modulation mode. Figures 1.14.14 and 1.14.15 show examples of how a 16-bit pulse width modulator operates and how an 8-bit pulse width modulator operates.

Table 1.14.5. Specifications in PWM Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Down-count (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new value at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs during counting
16-bit PWM	• High level width n / fj n : set value of TAi register (i=o to 4) • Cycle time $(2^{16}-1)$ / fj fixed fj: count source frequency (f_1, f_2, f_8, f_{32}, f_{C32})
8-bit PWM	• High level width n x (m+1) / fj n : set value of TAiMR register high-order address • Cycle time (2^8-1) x (m+1) / fj m : set value of TAiMR register low-order address
Count start condition	• External trigger input from the TAi_{IN} pin • Timer B2 overflow or underflow, timer Aj (j=i-1, except j=4 if i=0) overflow or underflow, timer Ak (k=i+1, except k=0 if i=4) overflow or underflow • TAiS bit of TABSR register is set to “1” (= start counting)
Count stop condition	TAiS bit is set to “0” (= stop counting)
Interrupt request generation timing	PWM pulse goes “L”
TAi_{IN} pin function	I/O port or trigger input
TAi_{OUT} pin function	Pulse output
Read from timer	An indeterminate value is read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)

Timer Ai mode register (i= 0 to 4)

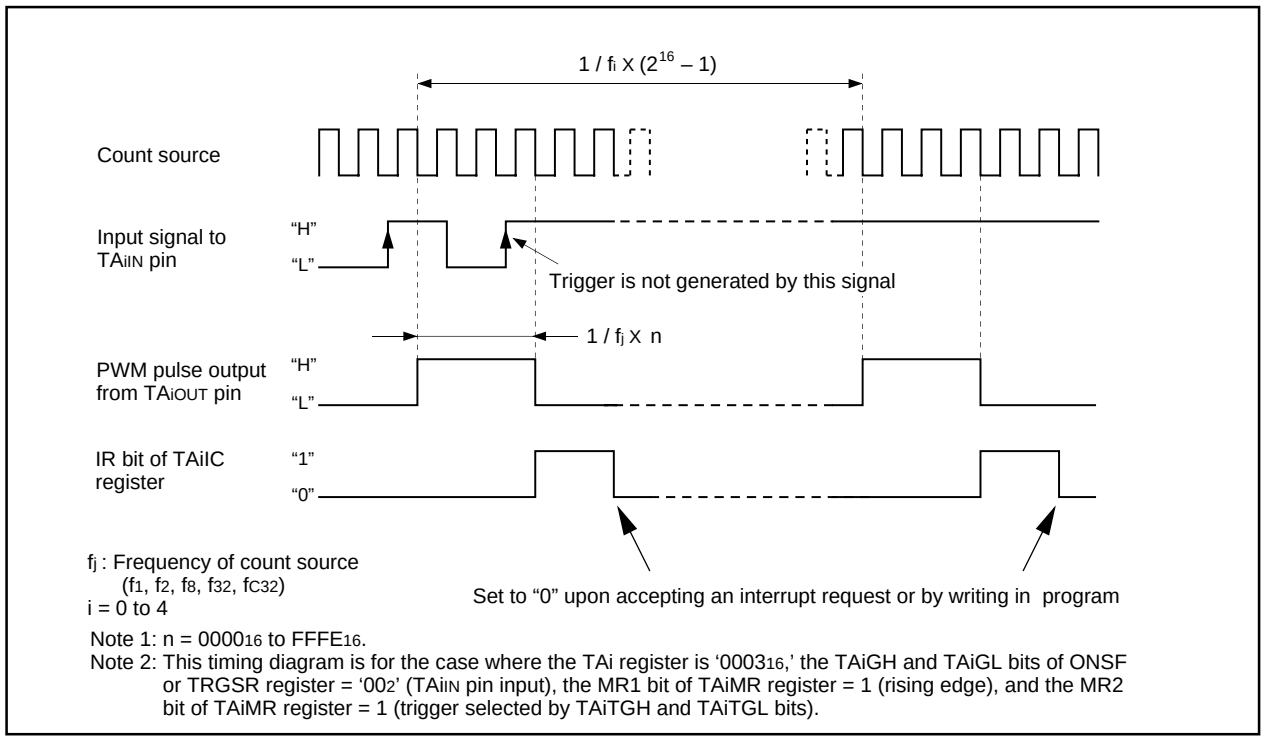
b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

Symbol	Address	After reset
TA0MR to TA4MR	0396_{16} to $039A_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 1 : PWM mode (Note 1)	RW
TMOD1			RW
MR0	Must be set to “1” in PWM mode		RW
MR1	External trigger select bit (Note 2)	0: Falling edge of input signal to TAiIN pin(Note 3) 1: Rising edge of input signal to TAiIN pin(Note 3)	RW
MR2	Trigger select bit	0 : TAiOS bit is enabled 1 : Selected by TAiTGH to TAiTGL bits	RW
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	RW
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	RW
TCK1			RW

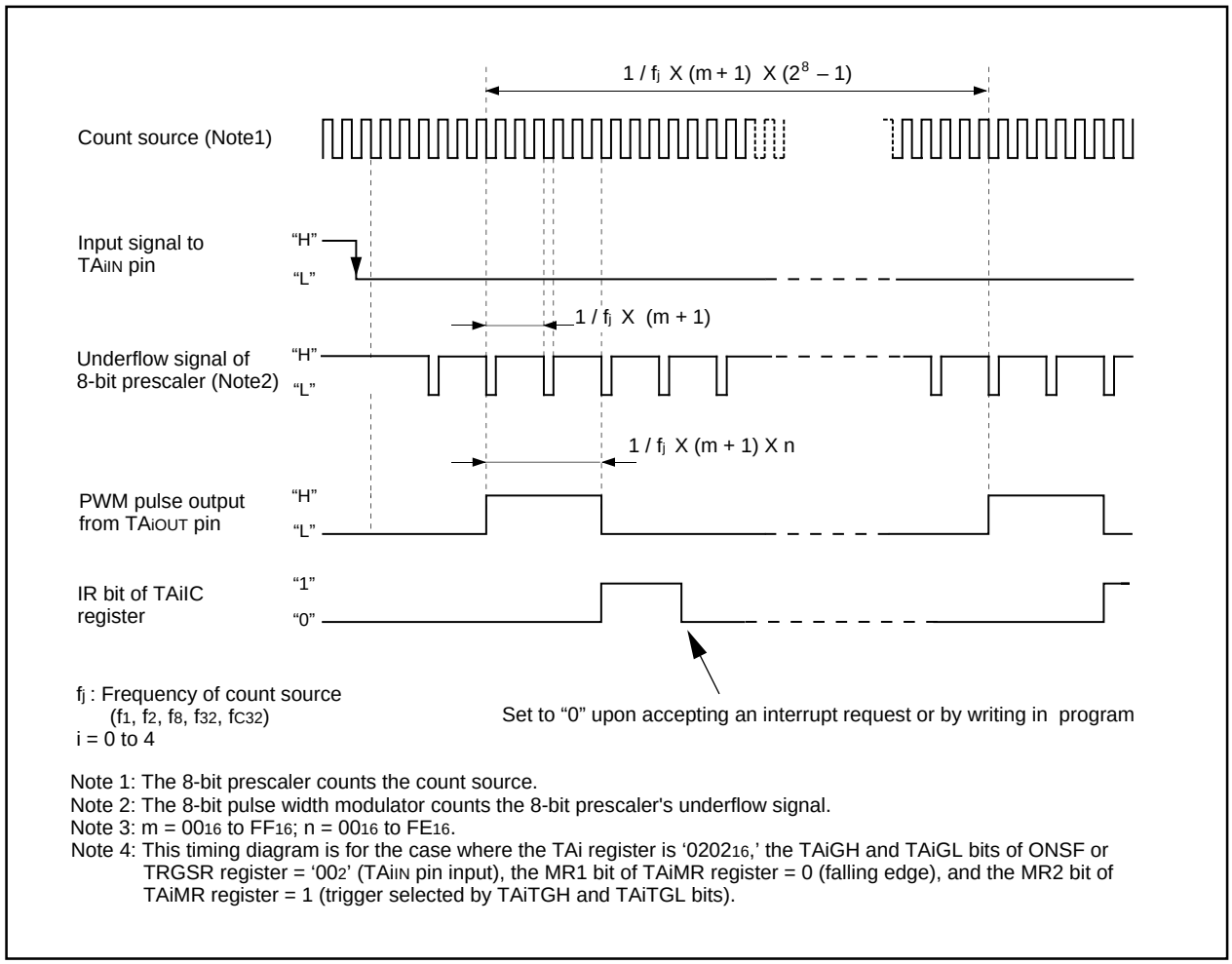
Note 1: TA0OUT pin is N-channel open drain output.
Note 2: Effective when the TAiGH and TAiGL bits of ONSF or TRGSR register are ‘00_2’ (TAiIN pin input).
Note 3: The port direction bit for the TAiIN pin must be set to “0” (= input mode).

Figure 1.14.13. TAiMR Register in PWM Mode



Note 1: n = 0000$_{16}$ to FFFE$_{16}$.
Note 2: This timing diagram is for the case where the TAi register is '0003$_{16}$,' the TAiGH and TAiGL bits of ONSF or TRGSR register = '00$_2$' (TAi$_{IN}$ pin input), the MR1 bit of TAiMR register = 1 (rising edge), and the MR2 bit of TAiMR register = 1 (trigger selected by TAiTGH and TAiTGL bits).

Figure 1.14.14. Example of 16-bit Pulse Width Modulator Operation



Note 1: The 8-bit prescaler counts the count source.
Note 2: The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
Note 3: m = 00$_{16}$ to FF$_{16}$; n = 00$_{16}$ to FE$_{16}$.
Note 4: This timing diagram is for the case where the TAi register is '0202$_{16}$,' the TAiGH and TAiGL bits of ONSF or TRGSR register = '00$_2$' (TAi$_{IN}$ pin input), the MR1 bit of TAiMR register = 0 (falling edge), and the MR2 bit of TAiMR register = 1 (trigger selected by TAiTGH and TAiTGL bits).

Figure 1.14.15. Example of 8-bit Pulse Width Modulator Operation

Timer B

Figure 1.15.1 shows a block diagram of the timer B. Figures 1.15.2 and 1.15.3 show registers related to the timer B.

Timer B supports the following three modes. Use the TMOD1 and TMOD0 bits of TBiMR register (i = 0 to 5) to select the desired mode.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external device or overflows or underflows of other timers.
- Pulse period/pulse width measuring mode: The timer measures an external signal's pulse period or pulse width.

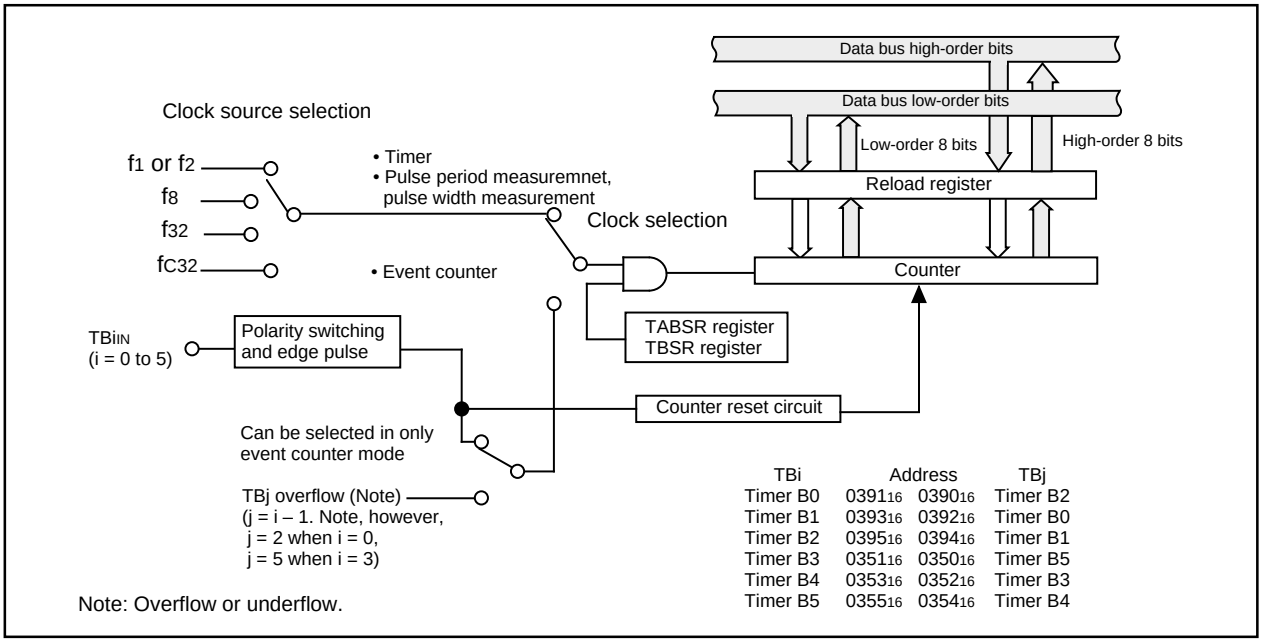


Figure 1.15.1. Timer B Block Diagram

Timer Bi mode register (i=0 to 5)

Symbol	Address	After reset
TB0MR to TB2MR	039B16 to 039D16	00XX00002
TB3MR to TB5MR	035B16 to 035D16	00XX00002

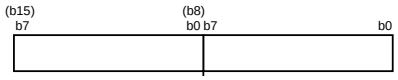
Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period measurement mode, pulse width measurement mode 1 1 : Must not be set	RW
TMOD1			RW
MR0	———	Function varies with each operation mode	RW
MR1			RW
MR2			RW (Note 1) ——— (Note 2)
MR3			RO
TCK0	Count source select bit	Function varies with each operation mode	RW
TCK1			RW

Note 1: Timer B0, timer B3.
Note 2: Timer B1, timer B2, timer B4, timer B5.

Figure 1.15.2. TB0MR to TB5MR Registers

Timer Bi register (i=0 to 5)(Note 1)

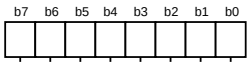
Symbol	Address	After reset
TB0	0391$_{16}$, 0390$_{16}$	Indeterminate
TB1	0393$_{16}$, 0392$_{16}$	Indeterminate
TB2	0395$_{16}$, 0394$_{16}$	Indeterminate
TB3	0351$_{16}$, 0350$_{16}$	Indeterminate
TB4	0353$_{16}$, 0352$_{16}$	Indeterminate
TB5	0355$_{16}$, 0354$_{16}$	Indeterminate



Mode	Function	Setting range	RW
Timer mode	Divide the count source by n + 1 where n = set value	0000$_{16}$ to FFFF$_{16}$	RW
Event counter mode	Divide the count source by n + 1 where n = set value (Note 2)	0000$_{16}$ to FFFF$_{16}$	RW
Pulse period modulation mode, Pulse width modulation mode	Measures a pulse period or width	——	RO

Note 1: The register must be accessed in 16 bit units.
Note 2: The timer counts pulses from an external device or overflows or underflows of other timers.

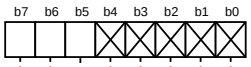
Count start flag



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

Bit symbol	Bit name	Function	RW
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	RW
TA1S	Timer A1 count start flag		RW
TA2S	Timer A2 count start flag		RW
TA3S	Timer A3 count start flag		RW
TA4S	Timer A4 count start flag		RW
TB0S	Timer B0 count start flag		RW
TB1S	Timer B1 count start flag		RW
TB2S	Timer B2 count start flag		RW

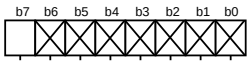
Timer B3, B4, B5 count start flag



Symbol	Address	After reset
TBSR	0340$_{16}$	000XXXXX$_2$

Bit symbol	Bit name	Function	RW
—— (b4-b0)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—
TB3S	Timer B3 count start flag	0 : Stops counting 1 : Starts counting	RW
TB4S	Timer B4 count start flag		RW
TB5S	Timer B5 count start flag		RW

Clock prescaler reset flag



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

Bit symbol	Bit name	Function	RW
—— (b6-b0)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—
CPSR	Clock prescaler reset flag	Setting this bit to “1” initializes the prescaler for the timekeeping clock. (When read, the value of this bit is “0”.)	RW

Figure 1.15.3. TB0 to TB5 Registers, TABSR Register, TBSR Register, CPSRF Register

1. Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 1.15.1). Figure 1.15.4 shows TBiMR register in timer mode.

Table 1.15.1. Specifications in Timer Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TBiMR register (i= 0 to 5) 0000_{16} to $FFFF_{16}$
Count start condition	Set TBiS bit(Note) to "1" (= start counting)
Count stop condition	Set TBiS bit to "0" (= stop counting)
Interrupt request generation timing	Timer underflow
TBi_{IN} pin function	I/O port
Read from timer	Count value can be read by reading TBi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)

Note : The TB0S to TB2S bits are assigned to the TABSR register bit 5 to bit 7, and the TB3S to TB5S bits are assigned to the TBSR register bit 5 to bit 7.

Timer Bi mode register (i= 0 to 5)

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol	Address	After reset
TB0MR to TB2MR	$039B_{16}$ to $039D_{16}$	$00XX0000_2$
TB3MR to TB5MR	$035B_{16}$ to $035D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0	Has no effect in timer mode Can be set to "0" or "1"		RW
MR1			RW
MR2	TB0MR, TB3MR registers Must be set to "0" in timer mode		RW
	TB1MR, TB2MR, TB4MR, TB5MR registers Nothing is assigned. When write, set to "0". When read, its content is indeterminate		—
MR3	When write in timer mode, set to "0". When read in timer mode, its content is indeterminate.		RO
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TCK1			RW

Figure 1.15.4. TBiMR Register in Timer Mode

2. Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers (see Table 1.15.2) . Figure 1.15.5 shows TBiMR register in event counter mode.

Table 1.15.2. Specifications in Event Counter Mode

Item	Specification
Count source	• External signals input to TBiIN pin (i=0 to 5) (effective edge can be selected in program) • Timer Bj overflow or underflow (j=i-1, except j=2 if i=0, j=5 if i=3)
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TBi register 0000_{16} to $FFFF_{16}$
Count start condition	Set TBiS bit[1] to “1” (= start counting)
Count stop condition	Set TBiS bit to “0” (= stop counting)
Interrupt request generation timing	Timer underflow
TBiIN pin function	Count source input
Read from timer	Count value can be read by reading TBi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)

Notes:
1. The TB0S to TB2S bits are assigned to the TABSR register bit 5 to bit 7, and the TB3S to TB5S bits are assigned to the TBSR register bit 5 to bit 7.

Timer Bi mode register (i=0 to 5)

b7	b6	b5	b4	b3	b2	b1	b0
						0	1

Symbol	Address	After reset
TB0MR to TB2MR	$039B_{16}$ to $039D_{16}$	$00XX0000_2$
TB3MR to TB5MR	$035B_{16}$ to $035D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
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	RW
MR1			RW
MR2	TB0MR, TB3MR registers Must be set to “0” in timer mode		RW
	TB1MR, TB2MR, TB4MR, TB5MR registers Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
MR3	When write in event counter mode, set to “0”. When read in event counter mode, its content is indeterminate.		RO
TCK0	Has no effect in event counter mode. Can be set to “0” or “1”.		RW
TCK1	Event clock select	0 : Input from TBiIN pin (Note 2) 1 : TBj overflow or underflow (j = i – 1, except j = 2 if i = 0, j = 5 if i = 3)	RW

Note 1: Effective when the TCK1 bit = “0” (input from TBiIN pin). If the TCK1 bit = “1” (TBj overflow or underflow), these bits can be set to “0” or “1”.
Note 2: The port direction bit for the TBiIN pin must be set to “0” (= input mode).

Figure 1.15.5. TBiMR Register in Event Counter Mode

3. Pulse Period and Pulse Width Measurement Mode

In pulse period and pulse width measurement mode, the timer measures pulse period or pulse width of an external signal (see Table 1.15.3). Figure 1.15.6 shows TBiMR register in pulse period and pulse width measurement mode. Figure 1.15.7 shows the operation timing when measuring a pulse period. Figure 1.15.8 shows the operation timing when measuring a pulse width.

Table 1.15.3. Specifications in Pulse Period and Pulse Width Measurement Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Up-count • Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to "0000_{16}" to continue counting.
Count start condition	Set TBiS (i=0 to 5) bit[3] to "1" (= start counting)
Count stop condition	Set TBiS bit to "0" (= stop counting)
Interrupt request generation timing	• When an effective edge of measurement pulse is input[1] • Timer overflow. When an overflow occurs, MR3 bit of TBiMR register is set to "1" (overflowed) simultaneously. MR3 bit is cleared to "0" (no overflow) by writing to TBiMR register at the next count timing or later after MR3 bit was set to "1". At this time, make sure TBiS bit is set to "1" (start counting).
TBi_{IN} pin function	Measurement pulse input
Read from timer	Contents of the reload register (measurement result) can be read by reading TBi register[2]
Write to timer	Value written to TBi register is written to neither reload register nor counter

Notes:
1. Interrupt request is not generated when the first effective edge is input after the timer started counting.
2. Value read from TBi register is indeterminate until the second valid edge is input after the timer starts counting.
3. The TB0S to TB2S bits are assigned to the TABSR register bit 5 to bit 7, and the TB3S to TB5S bits are assigned to the TBSR register bit 5 to bit 7.

Timer Bi mode register (i=0 to 5)

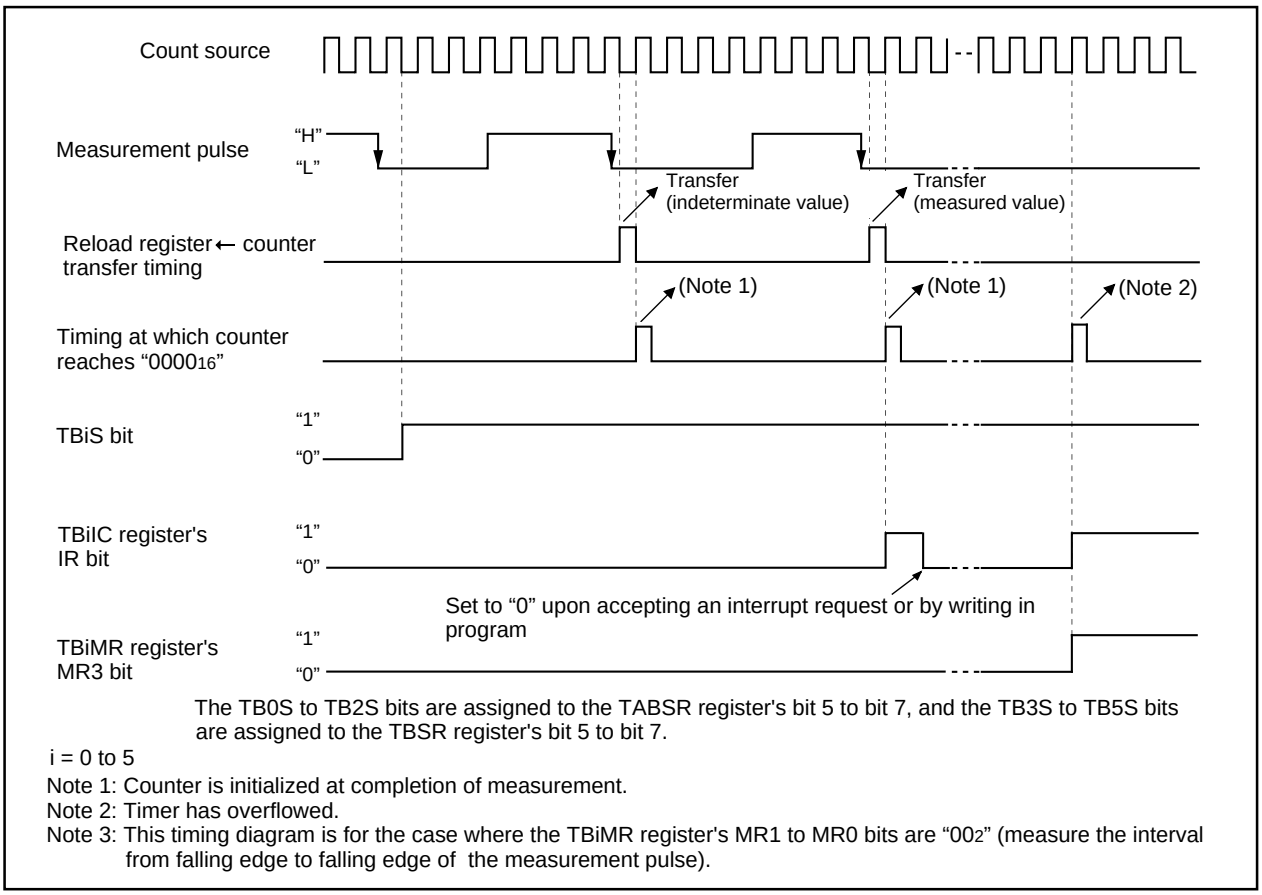
b7	b6	b5	b4	b3	b2	b1	b0
						1	0

Symbol	Address	After reset
TB0MR to TB2MR	$039B_{16}$ to $039D_{16}$	$00XX0000_2$
TB3MR to TB5MR	$035B_{16}$ to $035D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 0 : Pulse period / pulse width measurement mode	RW
TMOD1			RW
MR0	Measurement mode select bit	b3 b2 0 0 : Pulse period measurement (Measurement between a falling edge and the next falling edge of measured pulse) 0 1 : Pulse period measurement (Measurement between a rising edge and the next rising edge of measured pulse) 1 0 : Pulse width measurement (Measurement between a falling edge and the next rising edge of measured pulse and between a rising edge and the next falling edge) 1 1 : Must not be set.	RW
MR1			RW
MR2	TB0MR and TB3MR registers Must be set to "0" in pulse period and pulse width measurement mode		RW
	TB1MR, TB2MR, TB4MR, TB5MR registers Nothing is assigned. When write, set to "0". When read, its content turns out to be indeterminate.		—
MR3	Timer Bi overflow flag (Note)	0 : Timer did not overflow 1 : Timer has overflowed	RO
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TCK1			RW

Note: This flag is indeterminate after reset. When the TBiS bit = 1 (start counting), the MR3 bit is cleared to "0" (no overflow) by writing to the TBiMR register at the next count timing or later after the MR3 bit was set to "1" (overflowed). The MR3 bit cannot be set to "1" in a program. The TB0S to TB2S bits are assigned to the TABSR register's bit 5 to bit 7, and the TB3S to TB5S bits are assigned to the TBSR register's bit 5 to bit 7.

Figure 1.15.6. TBiMR Register in Pulse Period and Pulse Width Measurement Mode

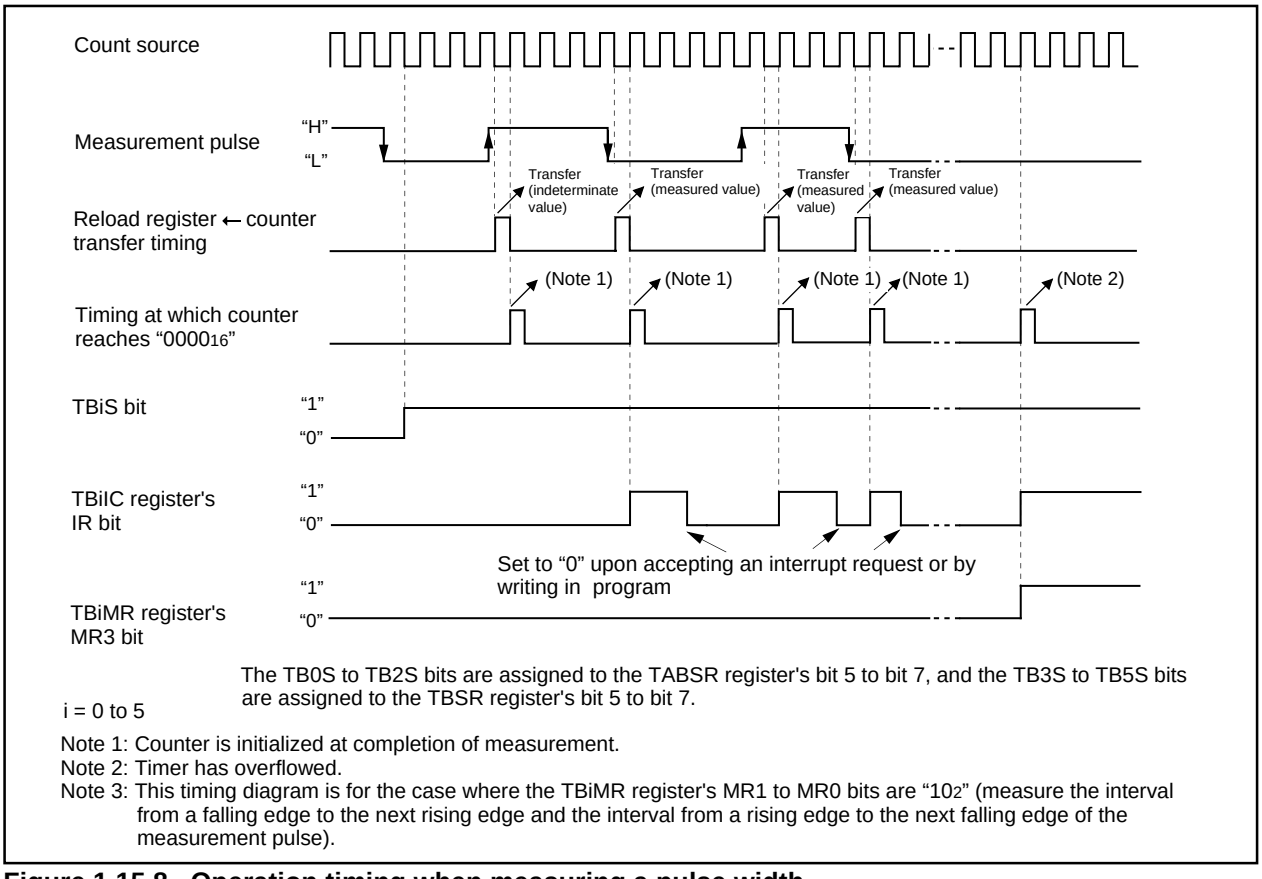


The TB0S to TB2S bits are assigned to the TABSR register's bit 5 to bit 7, and the TB3S to TB5S bits are assigned to the TBSR register's bit 5 to bit 7.

i = 0 to 5

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.
Note 3: This timing diagram is for the case where the TBiMR register's MR1 to MR0 bits are "00₂" (measure the interval from falling edge to falling edge of the measurement pulse).

Figure 1.15.7. Operation timing when measuring a pulse period



The TB0S to TB2S bits are assigned to the TABSR register's bit 5 to bit 7, and the TB3S to TB5S bits are assigned to the TBSR register's bit 5 to bit 7.

i = 0 to 5

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.
Note 3: This timing diagram is for the case where the TBiMR register's MR1 to MR0 bits are "10₂" (measure the interval from a falling edge to the next rising edge and the interval from a rising edge to the next falling edge of the measurement pulse).

Figure 1.15.8. Operation timing when measuring a pulse width

Three-phase Motor Control Timer Function

Timers A1, A2, A4 and B2 can be used to output three-phase motor drive waveforms. Table 1.16.1 lists the specifications of the three-phase motor control timer function. Figure 1.16.1 shows the block diagram for three-phase motor control timer function. Also, the related registers are shown on Figure 1.16.2 to Figure 1.16.7.

Table 1.16.1. Three-phase Motor Control Timer FunctionS Specifications

Item	Specification
Three-phase waveform output pin	Six pins (U, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$)
Forced cutoff input[1]	Input "L" to $\overline{NMI}$ pin
Used Timers	Timer A4, A1, A2 (used in the one-shot timer mode) Timer A4: U- and $\overline{U}$-phase waveform control Timer A1: V- and $\overline{V}$-phase waveform control Timer A2: W- and $\overline{W}$-phase waveform control Timer B2 (used in the timer mode) Carrier wave cycle control Dead timer timer (3 eight-bit timer and shared reload register) Dead time control
Output waveform	Triangular wave modulation, Sawtooth wave modification Enable to output "H" or "L" for one cycle Enable to set positive-phase level and negative-phase level respectively
Carrier wave cycle	Triangular wave modulation: count source x (m+1) x 2 Sawtooth wave modulation: count source x (m+1) m: Setting value of TB2 register, 0 to 65535 Count source: f_1, f_2, f_8, f_{32}, f_{C32}
Three-phase PWM output width	Triangular wave modulation: count source x n x 2 Sawtooth wave modulation: count source x n n: Setting value of TA4, TA1 and TA2 register (of TA4, TA41, TA1, TA11, TA2 and TA21 registers when setting the INV11 bit to "1"), 1 to 65535 Count source: f_1, f_2, f_8, f_{32}, f_{C32}
Dead time active disable function	Count source x p, or no dead time p: Setting value of DTT register, 1 to 255 Count source: f_1, f_2, f_1 divided by 2, f_2 divided by 2
Active level	Eable to select "H" or "L"
Positive and negative-phase concurrent	Positive and negative-phases concurrent active disable function Positive and negative-phases concurrent active detect func tion
Interrupt frequency	For Timer B2 interrupt, select a carrier wave cycle-to-cycle basis through 15 times carrier wave cycle-to-cycle basis

Notes:
1. Forced cutoff with $\overline{NMI}$ input is effective when the IVPCR1 bit of TB2SC register is set to "1" (three-phase output forcible cutoff by $\overline{NMI}$ input enabled). If an "L" signal is applied to the $\overline{NMI}$ pin when the IVPCR1 bit is "1", the related pins go to a high-impedance state regardless of which functions of those pins are being used.

Related pins
P7$_2$/CLK$_2$/TA1$_{OUT}$/V
P7$_3$/$\overline{CTS_2}$/$\overline{RTS_2}$/TA1$_{IN}$/$\overline{V}$
P7$_4$/TA2$_{OUT}$/W
P7$_5$/TA2$_{IN}$/$\overline{W}$
P8$_0$/TA4$_{OUT}$/U
P8$_1$/TA4$_{IN}$/$\overline{U}$

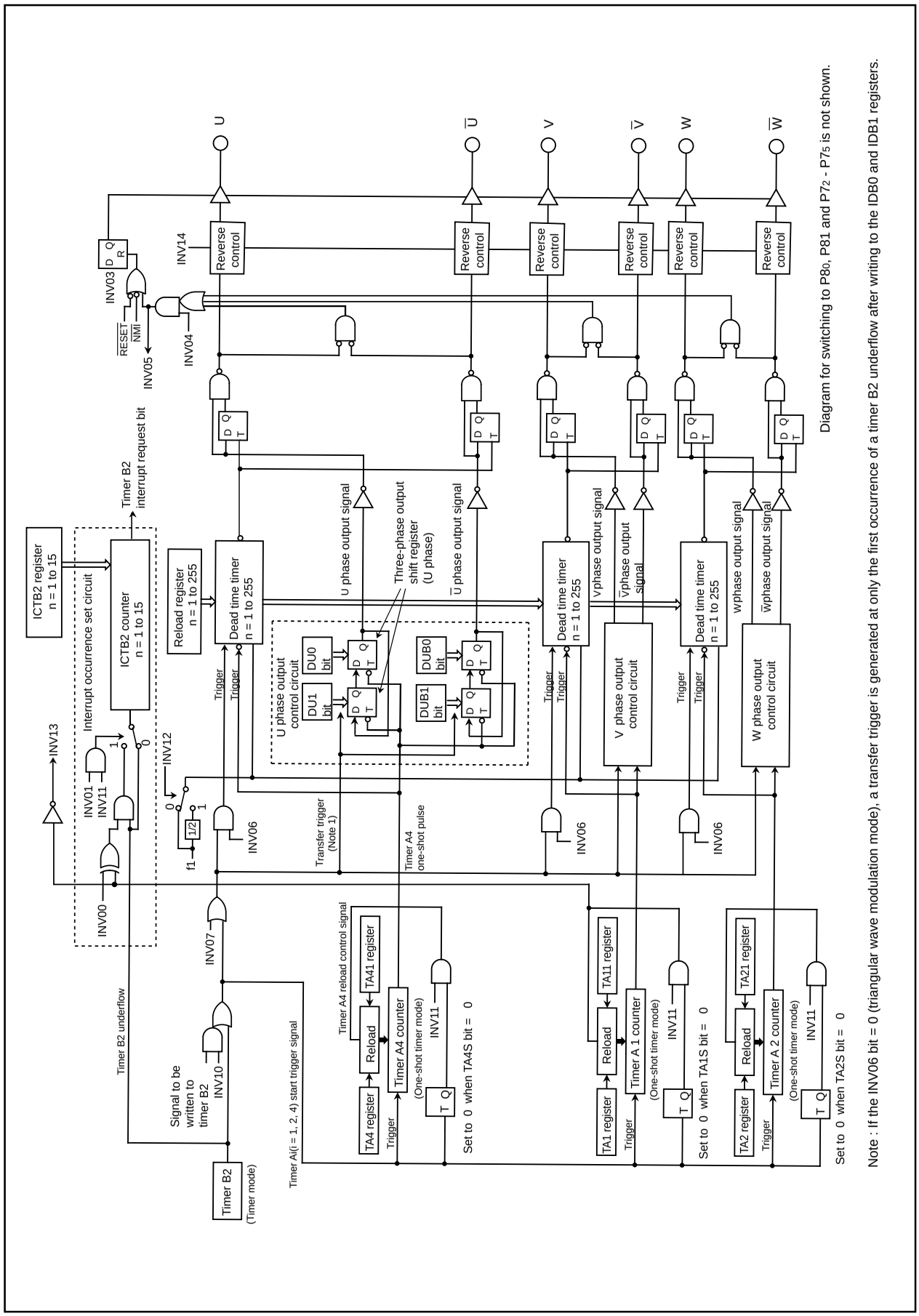
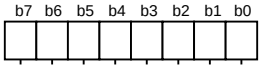


Figure 1.16.1. Three-phase Motor Control Timer Functions Block Diagram

Three-phase PWM control register 0 (Note 1)



Symbol: INVC0 | Address: 0348₁₆ | After reset: 00₁₆

Bit symbol	Bit name	Description	RW
INV00	Effective interrupt output polarity select bit	0: ICTB2 counter incremented by 1 at odd-numbered occurrences of a timer B2 underflow 1: ICTB2 counter incremented by 1 at even-numbered occurrences of a timer B2 underflow (Note 3)	RW
INV01	Effective interrupt output specification bit (Note 2)	0: ICTB2 counter incremented by 1 at a timer B2 underflow 1: Selected by INV00 bit (Note 3)	RW
INV02	Mode select bit (Note 4)	0: Three-phase motor control timer function unused (Note 5) 1: Three-phase motor control timer function	RW
INV03	Output control bit (Note 6)	0: Three-phase motor control timer output disabled (Note 5) 1: Three-phase motor control timer output enabled	RW
INV04	Positive and negative phases concurrent output disable function enable bit	0: Simultaneous active output enabled 1: Simultaneous active output disabled	RW
INV05	Positive and negative phases concurrent output detect flag	0: Not detected yet 1: Already detected (Note 7)	RW
INV06	Modulation mode select bit (Note 8)	0: Triangular wave modulation mode (Note 9) 1: Sawtooth wave modulation mode	RW
INV07	Software trigger select bit	Setting this bit to “1” generates a transfer trigger. If the INV06 bit is “1”, a trigger for the dead time timer is also generated. The value of this bit when read is “0”.	RW

Note 1: Write to this register after setting the PRC1 bit of PRCR register to “1” (write enable). Note also that this register can only be rewritten when timers A1, A2, A4 and B2 are idle.
Note 2: If this bit needs to be set to “1”, set any value in the ICTB2 register before writing to it.
Note 3: Effective when the INV11 bit is “1” (three-phase mode 1). If INV11 is “0” (three-phase mode 0), the ICTB2 counter is incremented by “1” each time the timer B2 underflows, regardless of whether the INV00 and INV01 bits are set.
Note 4: Setting the INV02 bit to “1” activates the dead time timer, U/V/W-phase output control circuits and ICTB2 counter.
Note 5: All of the U, $\overline{U}$, V, $\overline{V}$, W and $\overline{W}$ pins are placed in the high-impedance state by setting the INV02 bit to 1 (three-phase motor control timer function) and setting the INV03 bit to “0” (three-phase motor control timer output disable).
Note 6: The INV03 bit is set to “0” in the following cases:
- When reset
- When positive and negative go active simultaneously while INV04 bit is “1”
- When set to “0” in a program
- When input on the NMI pin changes state from “H” to “L” (The INV03 bit cannot be set to “1” when $\overline{\text{NMI}}$ input is “L”.)

Note 7: Can only be set by writing “0” in a program, and cannot be set to “1”.
Note 8: The effects of the INV06 bit are described in the table below.

Item	INV06=0	INV06=1
Mode	Triangular wave modulation mode	Sawtooth wave modulation mode
Timing at which transferred from IDB0 to IDB1 registers to three-phase output shift register	Transferred only once synchronously with the transfer trigger after writing to the IDB0 to IDB1 registers	Transferred every transfer trigger
Timing at which dead time timer trigger is generated when INV16 bit is “0”	Synchronous with the falling edge of timer A1, A2, or A4 one-shot pulse	Synchronous with the transfer trigger and the falling edge of timer A1, A2, or A4 one-shot pulse
INV13 bit	Effective when INV11 is “1” and INV06 is “0”	Has no effect

Transfer trigger: Timer B2 underflow, write to the INV07 bit or write to the TB2 register when INV10 is “1”
Note 9: If the INV06 bit is “1”, set the INV11 bit to “0” (three-phase mode 0) and set the PWCON bit to “0” (timer B2 reloaded by a timer B2 underflow).

Figure 1.16.2. INVC0 Register

Three-phase PWM control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
INVC1	0349_{16}	00_{16}

Bit symbol	Bit name	Description	RW
INV10	Timer A1, A2, A4 start trigger signal select bit	0: Timer B2 underflow 1: Timer B2 underflow and write to the TB2 register	RW
INV11	Timer A1-1, A2-1, A4-1 control bit (Note 2)	0: Three-phase mode 0 1: Three-phase mode 1 (Note 3)	RW
INV12	Dead time timer count source select bit	0 : f_1 or f_2 1 : f_1 divided by 2 or f_2 divided by 2	RW
INV13	Carrier wave detect flag (Note 4)	0: Timer A output at even-numbered occurrences (TAj1 register value counted) 1: Timer A output at odd-numbered occurrences (TAj1 register value counted)	RO
INV14	Output polarity control bit	0 : Output waveform “L” active 1 : Output waveform “H” active	RW
INV15	Dead time invalid bit	0: Dead time timer enabled 1: Dead time timer disabled	RW
INV16	Dead time timer trigger select bit	0: Falling edge of timer A4, A1 or A2 one-shot pulse 1: Rising edge of three-phase output shift register (U, V or W phase) output (Note 5)	RW
(b7)	Reserved bit	This bit should be set to “0”	RW

Note 1: Write to this register after setting the PRC1 bit of PRCR register to “1” (write enable). Note also that this register can only be rewritten when timers A1, A2, A4 and B2 are idle.

Note 2: The effects of the INV11 bit are described in the table below.

Item	INV11=0	INV11=1
Mode	Three-phase mode 0	Three-phase mode 1
TA11, TA21, TA41 registers	Not used	Used
INV00 bit, INV01 bit	Has no effect. ICTB2 counted every time timer B2 underflows regardless of whether the INV00 to INV01 bits are set.	Effect
INV13 bit	Has no effect	Effective when INV11 bit is “1” and INV06 bit is “0”

Note 3: If the INV06 bit is “1” (sawtooth wave modulation mode), set this bit to “0” (three-phase mode 0). Also, if the INV11 bit is “0”, set the PWCON bit to “0” (timer B2 reloaded by a timer B2 underflow).

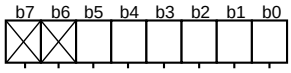
Note 4: The INV13 bit is effective only when the INV06 bit is “0” (triangular wave modulation mode) and the INV11 bit is “1” (three-phase mode 1).

Note 5: If all of the following conditions hold true, set the INV16 bit to “1” (dead time timer triggered by the rising edge of three-phase output shift register output)
- The INV15 bit is “0” (dead time timer enabled)
- When the INV03 bit is set to “1” (three-phase motor control timer output enabled), the Dij bit and DiBj bit (i:U, V, or W, j: 0 to 1) have always different values (the positive-phase and negative-phase always output different levels during the period other than dead time).

Conversely, if either one of the above conditions holds false, set the INV16 bit to “0” (dead time timer triggered by the falling edge of one-shot pulse).

Figure 1.16.3. INVC1 Register

Three-phase output buffer register i (i=0, 1) (Note)

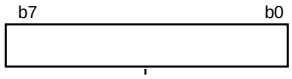


Symbol	Address	After reset
IDB0	034A16	0016
IDB1	034B16	0016

Bit Symbol	Bit name	Function	RW
DUi	U phase output buffer i	Write the output level 0: Active level 1: Inactive level When read, these bits show the three-phase output shift register value.	RW
DUBi	$\overline{U}$ phase output buffer i		RW
DVi	V phase output buffer i		RW
DVBi	$\overline{V}$ phase output buffer i		RW
DWi	W phase output buffer i		RW
DWBi	$\overline{W}$ phase output buffer i		RW
—— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		——

Note: The IDB0 and IDB1 register values are transferred to the three-phase shift register by a transfer trigger. The value written to the IDB0 register after a transfer trigger represents the output signal of each phase, and the next value written to the IDB1 register at the falling edge of the timer A1, A2 or A4 one-shot pulse represents the output signal of each phase.

Dead time timer (Note 1, Note 2)



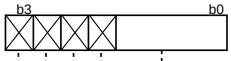
Symbol	Address	After reset
DTT	034C16	Indeterminate

Function	Setting range	RW
Assuming the set value = n, upon a start trigger the timer starts counting the count source selected by the INV12 bit and stops after counting it n times. The positive or negative phase whichever is going from an inactive to an active level changes at the same time the dead time timer stops.	1 to 255	WO

Note 1: Use MOV instruction to write to this register.
Note 2: Effective when the INV15 bit is “0” (dead time timer enable). If the INV15 bit is “1”, the dead time timer is disabled and has no effect.

Figure 1.16.4. IDB0 Register, IDB1Register, and DTT Register

Timer B2 interrupt occurrences frequency set counter

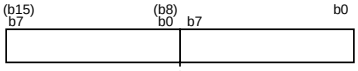


Symbol	Address	After reset
ICTB2	034D16	Indeterminate

Function	Setting range	RW
If the INV01 bit is "0" (ICTB2 counter counted every time timer B2 underflows), assuming the set value = n, a timer B2 interrupt is generated at every n'th occurrence of a timer B2 underflow. If the INV01 bit is "1" (ICTB2 counter count timing selected by the INV00 bit), assuming the set value = n, a timer B2 interrupt is generated at every n'th occurrence of a timer B2 underflow that meets the condition selected by the INV00 bit. (Note)	1 to 15	WO
Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—

Note : Use MOV instruction to write to this register.
If the INV01 bit = "1", make sure the TB2S bit also = "0" (timer B2 count stopped) when writing to this register.
If the INV01 bit = "0", although this register can be written even when the TB2S bit = "1" (timer B2 count start), do not write synchronously with a timer B2 underflow.

Timer Ai, Ai-1 register (i=1, 2, 4) (Note 1, Note 2, Note 3, Note 4, Note 5, Note 6)

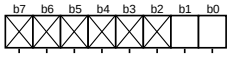


Symbol	Address	After reset
TA1	0389₁₆-0388₁₆	Indeterminate
TA2	038B₁₆-038A₁₆	Indeterminate
TA4	038F₁₆-038E₁₆	Indeterminate
TA11	0343₁₆-0342₁₆	Indeterminate
TA21	0345₁₆-0344₁₆	Indeterminate
TA41	0347₁₆-0346₁₆	Indeterminate

Function	Setting range	RW
Assuming the set value = n, upon a start trigger the timer starts counting the count source and stops after counting it n times. The positive and negative phases change at the same time timer A, A2 or A4 stops.	0000₁₆ to FFFF₁₆	WO

Note 1: The register must be accessed in 16 bit units.
Note 2: When the timer Ai register is set to "0000₁₆", the counter does not operate and a timer Ai interrupt does not occur.
Note 3: Use MOV instruction to write to these registers.
Note 4: If the INV15 bit is "0" (dead time timer enable), the positive or negative phase whichever is going from an inactive to an active level changes at the same time the dead time timer stops.
Note 5: If the INV11 bit is "0" (three-phase mode 0), the TAi register value is transferred to the reload register by a timer Ai (i = 1, 2 or 4) start trigger.
If the INV11 bit is "1" (three-phase mode 1), the TAi1 register value is transferred to the reload register by a timer Ai start trigger first and then the TAi register value is transferred to the reload register by the next timer Ai start trigger. Thereafter, the TAi1 register and TAi register values are transferred to the reload register alternately.
Note 6: Do not write to these registers synchronously with a timer B2 underflow.
Note 7: Write to the TAi1 register as follows:
(1) Write a value to the TAi1 register.
(2) Wait for one cycle of timer Ai count source.
(3) Write the same value to the TAi1 register again.

Timer B2 special mode register



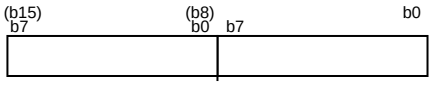
Symbol	Address	After reset
TB2SC	039E₁₆	XXXXXX00₂

Bit symbol	Bit name	Function	RW
PWCOM	Timer B2 reload timing switching bit	0 : Timer B2 underflow 1 : Timer A output at odd-numbered occurrences (Note 2)	RW
IVPCR1	Three phase output port NMI control bit 1 (Note 3)	0 : Three-phase output forcible cutoff by NMI input (high impedance) disabled 1 : Three-phase output forcible cutoff by NMI input (high impedance) enabled	RW
— (b7-b2)	Nothing is assigned. When write, set to "0". When read, its content is "0".		—

Note 1: Write to this register after setting the PRC1 bit of PRCR register to "1" (write enable).
Note 2: If the INV11 bit is "0" (three-phase mode 0) or the INV06 bit is "1" (triangular wave modulation mode), set this bit to "0" (timer B2 underflow).
Note 3: Related pins are U(P8₀), $\overline{U}$(P8₁), V(P7₂), $\overline{V}$(P7₃), W(P7₄) and $\overline{W}$(P7₅). If a low-level signal is applied to the $\overline{NMI}$ pin when the IVPCR1 bit = 1, the target pins go to a high-impedance state regardless of which functions of those pins are being used. After forced interrupt (cutoff), input "H" to the NMI pin and set IVPCR1 bit to "0": this forced cutoff will be reset.

Figure 1.16.5. ICTB2 Register, TA1, TA2, TA4, TA11, TA21 and TA41 Registers, and TB2SC Registers

Timer B2 register (Note)

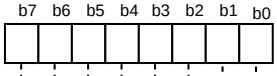


Symbol	Address	After reset
TB2	039516-039416	Indeterminate

Function	Setting range	RW
Divide the count source by n + 1 where n = set value. Timer A1, A2 and A4 are started at every occurrence of underflow.	000016 to FFFF16	RW

Note : The register must be accessed in 16 bit units.

Trigger select register

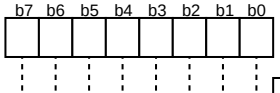


Symbol	Address	After reset
TRGSR	038316	0016

Bit symbol	Bit name	Function	RW
TA1TGL	Timer A1 event/trigger select bit	To use the V-phase output control circuit, set these bits to "012"(TB2 underflow).	RW
TA1TGH			RW
TA2TGL	Timer A2 event/trigger select bit	To use the W-phase output control circuit, set these bits to "012"(TB2 underflow).	RW
TA2TGH			RW
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on TA3IN is selected (Note 1) 0 1 : TB2 overflow is selected (Note 2) 1 0 : TA2 overflow is selected (Note 2) 1 1 : TA4 overflow is selected (Note 2)	RW
TA3TGH			RW
TA4TGL	Timer A4 event/trigger select bit	To use the U-phase output control circuit, set these bits to "012"(TB2 underflow).	RW
TA4TGH			RW

Note 1: Set the corresponding port direction bit to "0" (input mode).
Note 2: Overflow or underflow.

Count start flag



Symbol	Address	After reset
TABSR	038016	0016

Bit symbol	Bit name	Function	RW
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	RW
TA1S	Timer A1 count start flag		RW
TA2S	Timer A2 count start flag		RW
TA3S	Timer A3 count start flag		RW
TA4S	Timer A4 count start flag		RW
TB0S	Timer B0 count start flag		RW
TB1S	Timer B1 count start flag		RW
TB2S	Timer B2 count start flag		RW

Figure 1.16.6. TB2 Register, TRGSR Register, and TABSR Register

Timer Ai mode register

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

Symbol	Address	After reset
TA1MR	039716	0016
TA2MR	039816	0016
TA4MR	$039A16$	0016

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	Must set to "102" (one-shot timer mode) for the three-phase motor control timer function	RW
TMOD1			RW
MR0	Pulse output function select bit	Must set to "0" for the three-phase motor control timer function	—
MR1	External trigger select bit	Has no effect for the three-phase motor control timer function	RW
MR2	Trigger select bit	Must set to "1" (selected by event/trigger select register) for the three-phase motor control timer function	RW
MR3	Must set to "0" for the three-phase motor control timer function		RW
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	RW
TCK1			RW

Timer B2 mode register

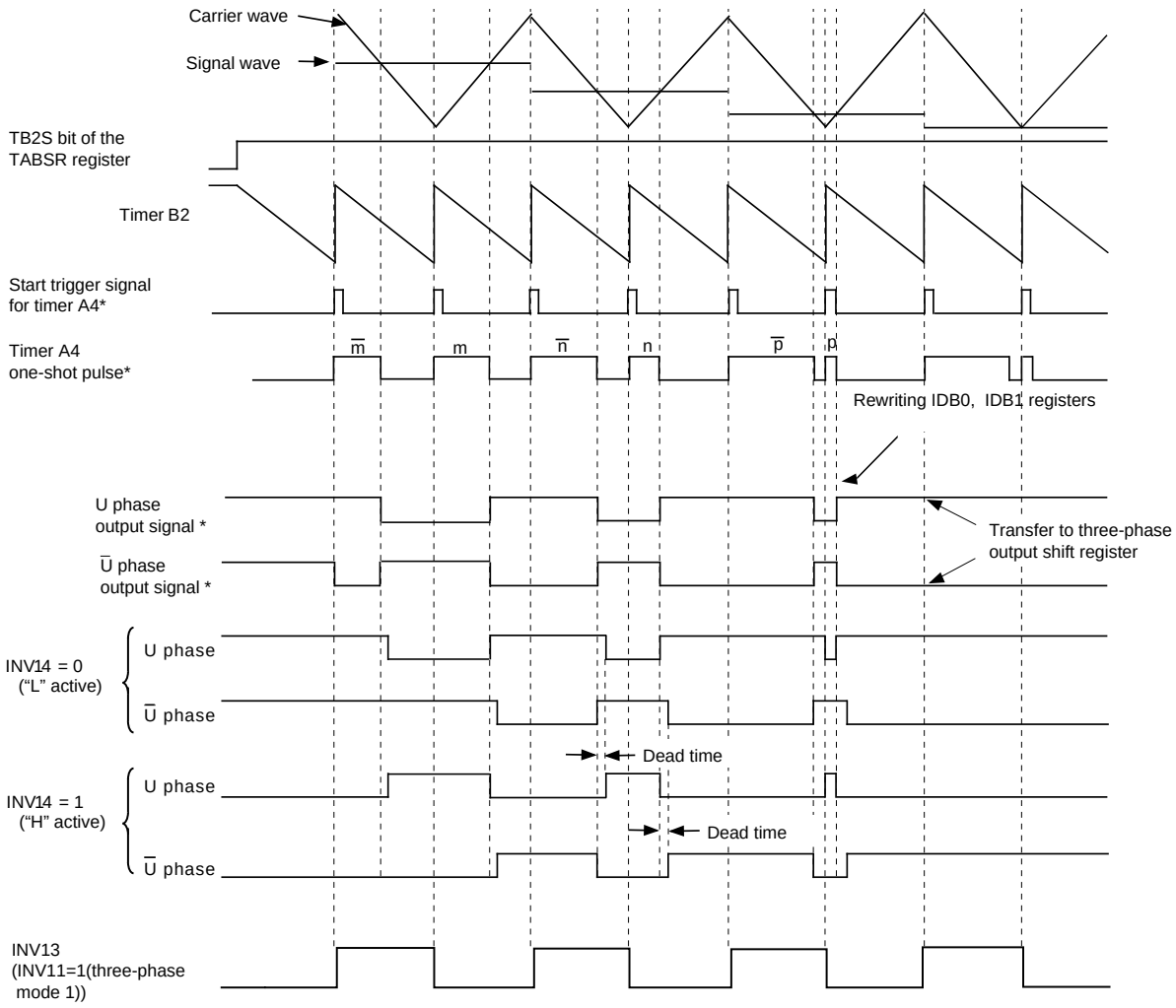
b7	b6	b5	b4	b3	b2	b1	b0
			0			0	0

Symbol	Address	After reset
TB2MR	$039D16$	$00XX0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	Set to "002" (timer mode) for the three-phase motor control timer function	RW
TMOD1			RW
MR0	Has no effect for the three-phase motor control timer function. When write, set to "0". When read, its content is indeterminate.		RW
MR1			RW
MR2	Must set to "0" for the three-phase motor control timer function		RW
MR3	When write in three-phase motor control timer function, write "0". When read, its content is indeterminate.		RO
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	RW
TCK1			RW

Figure 1.16.7. TA1MR, TA2MR, TA4MR, and TB2MR Registers

The three-phase motor control timer function is enabled by setting the INV02 bit of INVC0 register to “1”. When this function is on, timer B2 is used to control the carrier wave, and timers A4, A1 and A2 are used to control three-phase PWM outputs (U, $\overline{U}$, V, $\overline{V}$, W and $\overline{W}$). The dead time is controlled by a dedicated dead time timer. Figure 1.16.8 shows the example of triangular modulation waveform and Figure 1.16.9 shows the example of sawtooth modulation waveform.



* Internal signals. See the block diagram of the three-phase motor control timer function.

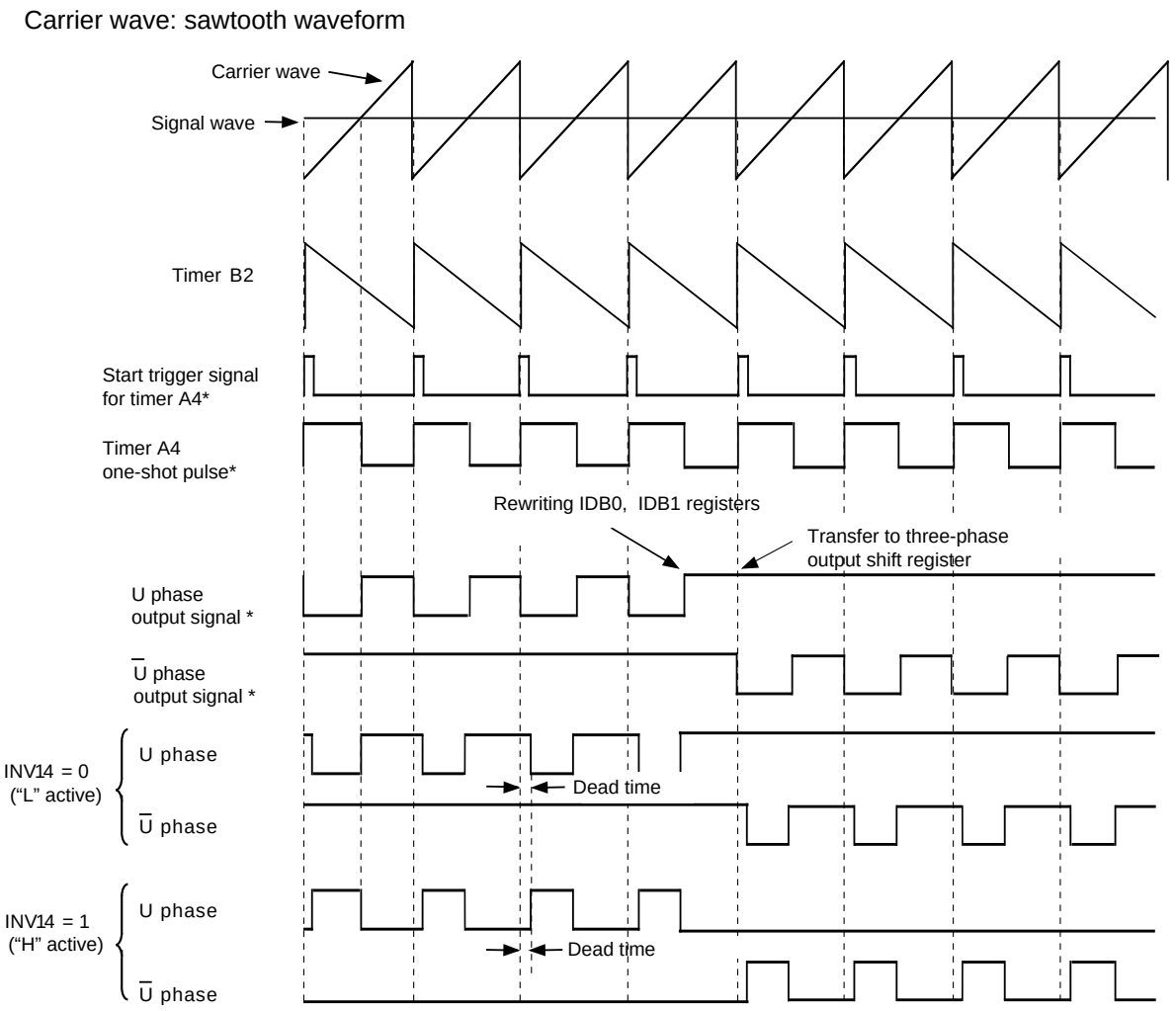
Shown here is a typical waveform for the case where INVC0 = 00XX11XX$_2$ (X = set as suitable for the system) and INVC1 = 010XXXX0$_2$.
An example for changing PWM outputs is shown below.

(1)When INV11=1(three-phase mode 1)
- INV01=0, ICTB2=2$_{16}$(timer B2 interrupt is generated at every 2’th occurrence of a timer B2 underflow), or INV01=1, INV00=1, ICTB2=1$_{16}$(timer B2 interrupt is generated at even-numbered occurrences of a timer B2 underflow).
- Initial timer value: TA41=m̄, TA4=m. The TA4 and TA41 registers are modified every time a timer B2 interrupt occurs. First time, TA41= n̄, TA4 = n. Second time, TA41 = p̄, TA4 = p.
- Initial values of IDB0 and IDB1 registers: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1.The register values are changed to DU0 = 1, DUB0 = 0, DU1= 1 and DUB1 = 0 the third time a timer B2 interrupt occurs.

(2)When INV11=0(three-phase mode 0)
- INV01=0, ICTB2=1$_{16}$(timer B2 interrupt is generated at every occurrence of a timer B2 underflow)
- Initial timer value: TA4 = m̄. The TA4 register is modified each time a timer B2 interrupt occurs. First time, TA4 = m. Second time, TA4 = n̄. Third time, TA4 = n. Fourth time, TA4 = p̄. Fifth time, TA4 = p.
- Initial values of IDB0 and IDB1 registers: DU0=1, DUB0=0, DU1=0, DUB1=1.The register values are changed to DU0 = 1, DUB0 = 0, DU1= 1 and DUB1 = 0 the sixth time a timer B2 interrupt occurs.

The value written to the TA4 register and TA41 register are inverted at odd-numbered timer A outputs.

Figure 1.16.8. Triangular Wave Modulation Operation



* Internal signals. See the block diagram of the three-phase motor control timer function.

Shown here is a typical waveform for the case where INVC0= 01XX110X2 (X = set as suitable for the system) and INVC1 = 010XXX002. An example for changing PWM outputs is shown below.
• ICTB2=n (timer B2 interrupt is generated at every n'th occurrence of a timer B2 underflow)
• Initial values of IDB0 and IDB1 registers: DU0=0, DUB0=1, DU1=1, DUB1=1. The register values are changed to DU0=1, DUB0=0, DU1=1, DUB1=1 a timer B2 interrupt occurs.

Figure 1.16.9. Sawtooth Wave Modulation Operation

Serial I/O

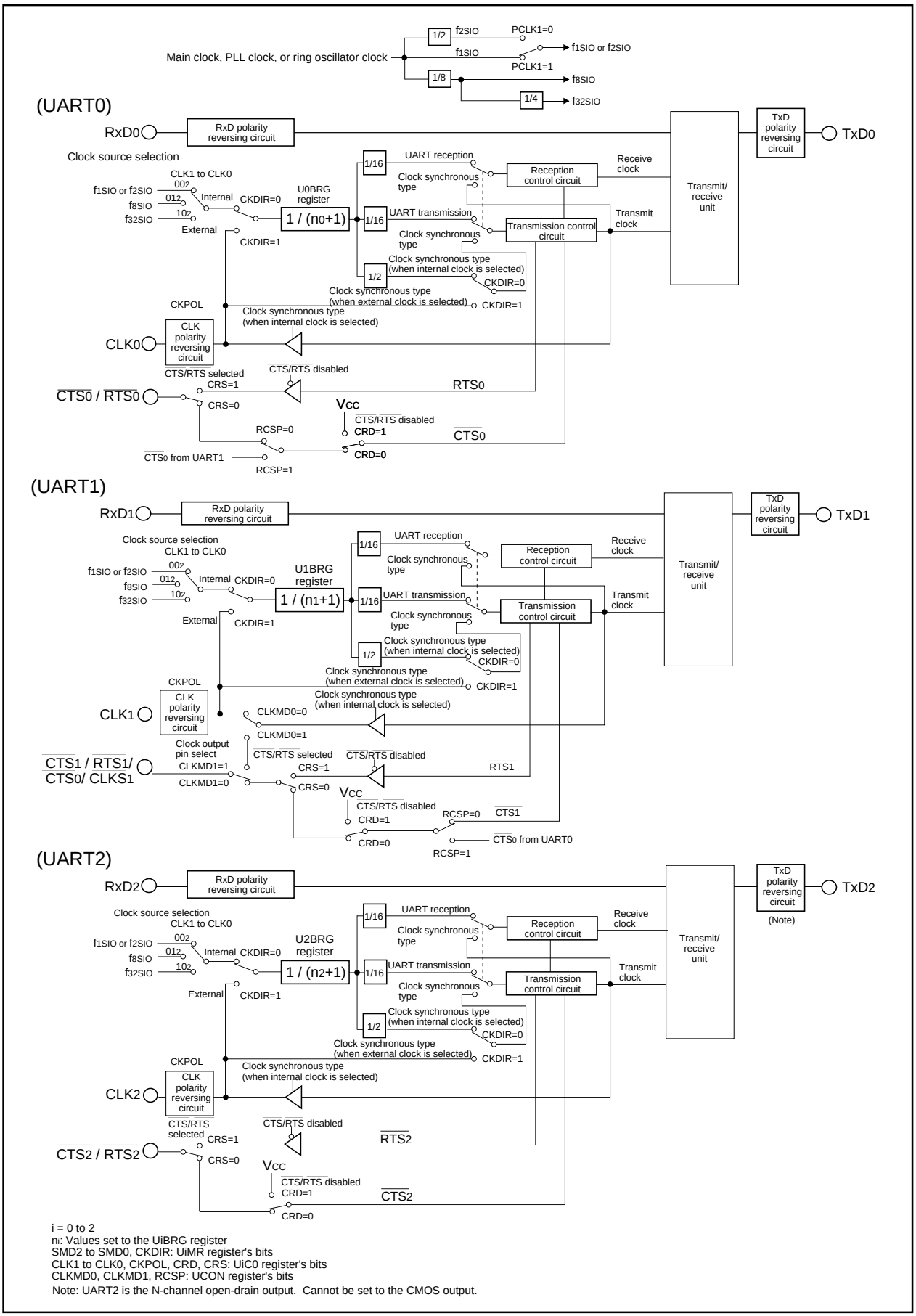
Serial I/O is configured with five channels: UART0 to UART2, SI/O3 and SI/O4.

UARTi (i=0 to 2)

UARTi each have an exclusive timer to generate a transfer clock, so they operate independently of each other.
Figure 1.17.1 shows the block diagram of UARTi. Figures 1.17.2 shows the block diagram of the UARTi transmit/receive.

UARTi has the following modes:
- Clock synchronous serial I/O mode
- Clock asynchronous serial I/O mode (UART mode).
- Special mode 1 (I^2C mode)
- Special mode 2
- Special mode 3 (Bus collision detection function, IE mode) : UART0, UART1
- Special mode 4 (SIM mode) : UART2

Figures 1.17.3 to 1.17.8 show the UARTi-related registers.
Refer to tables listing each mode for register setting.



i = 0 to 2
ni: Values set to the UiBRG register
SMD2 to SMD0, CKDIR: UiMR register's bits
CLK1 to CLK0, CKPOL, CRD, CRS: UiC0 register's bits
CLKMD0, CLKMD1, RCSP: UCON register's bits
Note: UART2 is the N-channel open-drain output. Cannot be set to the CMOS output.

Figure 1.17.1. UARTi Block Diagram

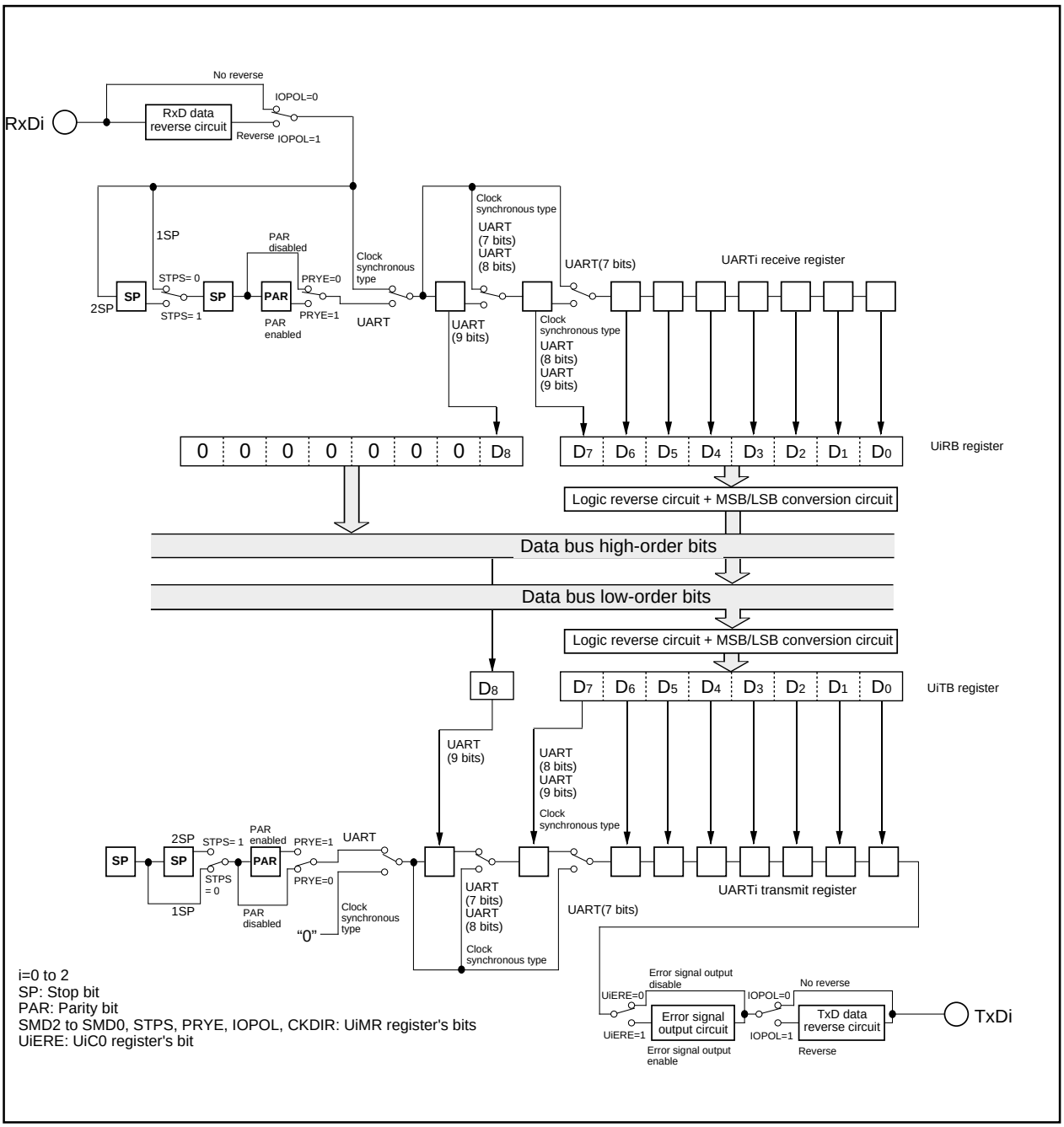
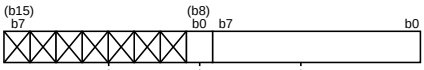


Figure 1.17.2. UARTi Transmit/Receive Unit

UARTi transmit buffer register (i=0 to 2)(Note)

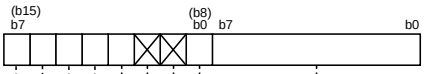


Symbol	Address	After reset
U0TB	$03A3_{16}$-$03A2_{16}$	Indeterminate
U1TB	$03AB_{16}$-$03AA_{16}$	Indeterminate
U2TB	$037B_{16}$-$037A_{16}$	Indeterminate

Function	RW
Transmit data	WO
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 (i=0 to 2)

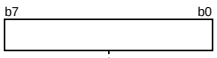


Symbol	Address	After reset
U0RB	$03A7_{16}$-$03A6_{16}$	Indeterminate
U1RB	$03AF_{16}$-$03AE_{16}$	Indeterminate
U2RB	$037F_{16}$-$037E_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
— (b7-b0)	———	Receive data (D_7 to D_0)	RO
— (b8)	———	Receive data (D_8)	RO
— (b10-b9)	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	RW
OER	Overrun error flag (Note 1)	0 : No overrun error 1 : Overrun error found	RO
FER	Framing error flag (Note 1)	0 : No framing error 1 : Framing error found	RO
PER	Parity error flag (Note 1)	0 : No parity error 1 : Parity error found	RO
SUM	Error sum flag (Note 1)	0 : No error 1 : Error found	RO

Note 1: When the UiMR register’s SMD2 to SMD0 bits = “000_2” (serial I/O disabled) or the UiC1 register’s RE bit = “0” (reception disabled), all of the SUM, PER, FER and OER bits are set to “0” (no error). The SUM bit is set to “0” (no error) when all of the PER, FER and OER bits = “0” (no error). Also, the PER and FER bits are set to “0” by reading the lower byte of the UiRB register.
Note 2: The ABT bit is set to “0” by writing “0” in a program. (Writing “1” has no effect.)

UARTi bit rate generator (i=0 to 2)(Notes 1, 2)



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

Function	Setting range	RW
Assuming that set value = n, UiBRG divides the count source by n + 1	00_{16} to FF_{16}	WO

Note 1: Write to this register while serial I/O is neither transmitting nor receiving.
Note 2: Use MOV instruction to write to this register.

Figure 1.17.3. U0TB to U2TB Register, U0RB to U2RB Register, and U0BRG to U2BRG Register

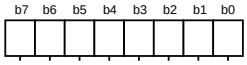
UARTi transmit/receive mode register (i=0 to 2)

Symbol	Address	After reset
U0MR to U2MR	$03A0_{16}$, $03A8_{16}$, 0378_{16}	00_{16}

Bit symbol	Bit name	Function	RW
SMD0	Serial I/O mode select bit (Note 2)	b2 b1 b0 0 0 0 : Serial I/O disabled 0 0 1 : Clock synchronous serial I/O mode 0 1 0 : I^2C mode (Note 3) 1 0 0 : UART mode transfer data 7 bits long 1 0 1 : UART mode transfer data 8 bits long 1 1 0 : UART mode transfer data 9 bits long Must not be set except above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note 1)	RW
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	RW
PRY	Odd/even parity select bit	Effective when PRYE = 1 0 : Odd parity 1 : Even parity	RW
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	RW
IOPOL	TxD, RxD I/O polarity reverse bit	0 : No reverse 1 : Reverse	RW

Note 1: Set the corresponding port direction bit for each CLKi pin to "0" (input mode).
Note 2: To receive data, set the corresponding port direction bit for each RxDi pin to "0" (input mode).
Note 3: Set the corresponding port direction bit for SCL and SDA pins to "0" (input mode).

UARTi transmit/receive control register 0 (i=0 to 2)



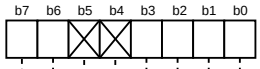
Symbol	Address	After reset
U0C0 to U2C0	$03A4_{16}$, $03AC_{16}$, $037C_{16}$	00001000_2

Bit symbol	Bit name	Function	RW
CLK0	BRG count source select bit	b1 b0 0 0 : $f1_{SIO}$ or $f2_{SIO}$ is selected 0 1 : $f8_{SIO}$ is selected 1 0 : $f32_{SIO}$ is selected 1 1 : Must not be set	RW
CLK1			RW
CRS	$\overline{CTS}/\overline{RTS}$ function select bit (Note 4)	Effective when CRD = 0 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected	RW
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	RO
CRD	$\overline{CTS}/\overline{RTS}$ disable bit	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : CTS/RTS function disabled ($P6_0$, $P6_4$ and $P7_3$ can be used as I/O ports)	RW
NCH	Data output select bit (Note 2)	0 : TxDi/SDAi and SCLi pins are CMOS output 1 : TxDi/SDAi and SCLi pins are N-channel open-drain output	RW
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	RW
UFORM	Transfer format select bit (Note 3)	0 : LSB first 1 : MSB first	RW

Note 1: Set the corresponding port direction bit for each $\overline{CTSi}$ pin to "0" (input mode).
Note 2: TxD_2/SDA_2 and SCL_2 are N-channel open-drain output. Cannot be set to the CMOS output. Set the NCH bit of the U2C0 register to "0".
Note 3: Effective for clock synchronous serial I/O mode and UART mode transfer data 8 bits long.
Note 4: CTS_1/RTS_1 can be used when the UCON register's CLKMD1 bit = "0" (only CLK_1 output) and the UCON register's RCSP bit = "0" (CTS_0/RTS_0 not separated).

Figure 1.17.4. U0MR to U2MR Register and U0C0 to U2C0 Register

UARTi transmit/receive control register 1 (i=0, 1)



Symbol: U0C1, U1C1 — Address: $03A5_{16}$,$03AD_{16}$ — After reset: 00000010_2

Bit symbol	Bit name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data present in UiTB register 1 : No data present in UiTB register	RO
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag	0 : No data present in UiRB register 1 : Data present in UiRB register	RO
— (b5-b4)	Nothing is assigned. When write, set “0”. When read, these contents are “0”.		—
UiLCH	Data logic select bit	0 : No reverse 1 : Reverse	RW
UiERE	Error signal output enable bit	0 : Output disabled 1 : Output enabled	RW

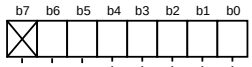
UART2 transmit/receive control register 1

Symbol: U2C1 — Address: $037D_{16}$ — After reset: 00000010_2

Bit symbol	Bit name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data present in U2TB register 1 : No data present in U2TB register	RO
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag	0 : No data present in U2RB register 1 : Data present in U2RB register	RO
U2IRS	UART2 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	RW
U2RRM	UART2 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	RW
U2LCH	Data logic select bit	0 : No reverse 1 : Reverse	RW
U2ERE	Error signal output enable bit	0 : Output disabled 1 : Output enabled	RW

Figure 1.17.5. U0C1 to U2C1 Registers

UART transmit/receive control register 2

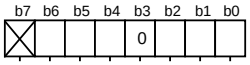


Symbol: UCON | Address: $03B0_{16}$ | After reset: $X0000000_2$

Bit symbol	Bit name	Function	RW
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	RW
U1IRS	UART1 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	RW
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	RW
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	RW
CLKMD0	UART1 CLK/CLKS select bit 0	Effective when CLKMD1 = "1" 0 : Clock output from CLK1 1 : Clock output from CLKS1	RW
CLKMD1	UART1 CLK/CLKS select bit 1 (Note)	0 : CLK output is only CLK1 1 : Transfer clock output from multiple pins function selected	RW
RCSP	Separate UART0 CTS/RTS bit	0 : $\overline{CTS}/\overline{RTS}$ shared pin 1 : $\overline{CTS}/\overline{RTS}$ separated (CTS_0 supplied from the $P6_4$ pin)	RW
— (b7)	Nothing is assigned. When write, set "0". When read, its content is indeterminate.		—

Note: When using multiple transfer clock output pins, make sure the following conditions are met:
U1MR register's CKDIR bit = "0" (internal clock)

UART2 special mode register (i=0 to 2)



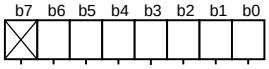
Symbol: U0SMR to U2SMR | Address: $036F_{16}$, 0373_{16}, 0377_{16} | After reset: $X0000000_2$

Bit symbol	Bit name	Function	RW
IICM	I^2C mode select bit	0 : Other than I^2C mode 1 : I^2C mode	RW
ABC	Arbitration lost detecting flag control bit	0 : Update per bit 1 : Update per byte	RW
BBS	Bus busy flag	0 : STOP condition detected 1 : START condition detected (busy)	RW (Note1)
— (b3)	Reserved bit	Set to "0"	RW
ABSCS	Bus collision detect sampling clock select bit	0 : Rising edge of transfer clock 1 : Underflow signal of timer Aj (Note 2)	RW
ACSE	Auto clear function select bit of transmit enable bit	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	RW
SSS	Transmit start condition select bit	0 : Irrelevant to RxDi 1 : Synchronous with RxDi (Note 3)	RW
— (b7)	Nothing is assigned. When write, set "0". When read, its content is indeterminate.		—

Note 1: The BBS bit is set to "0" by writing "0" in a program. (Writing "1" has no effect.).
Note 2: Underflow signal of timer A3 in UART0, underflow signal of timer A4 in UART1, underflow signal of timer A0 in UART2.
Note 3: When a transfer begins, the SSS bit is set to "0" (irrelevant to RxDi).

Figure 1.17.6. UCON Register and U0SMR to U2SMR Registers

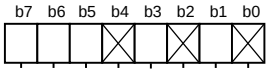
UARTi special mode register 2 (i=0 to 2)



Symbol	Address	After reset
U0SMR2 to U2SMR2	036E16, 037216, 037616	X00000002

Bit symbol	Bit name	Function	RW
IICM2	I^2C mode select bit 2	Refer to Table 1.20.4	RW
CSC	Clock-synchronous bit	0 : Disabled 1 : Enabled	RW
SWC	SCL wait output bit	0 : Disabled 1 : Enabled	RW
ALS	SDA output stop bit	0 : Disabled 1 : Enabled	RW
STAC	UARTi initialization bit	0 : Disabled 1 : Enabled	RW
SWC2	SCL wait output bit 2	0: Transfer clock 1: 0 output	RW
SDHI	SDA output disable bit	0: Enabled 1: Disabled (high impedance)	RW
— (b7)	Nothing is assigned. When write, set “0”. When read, its content is indeterminate.		—

UARTi special mode register 3 (i=0 to 2)



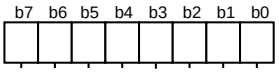
Symbol	Address	After reset
U0SMR3 to U2SMR3	036D16, 037116, 037516	000X0X0X2

Bit symbol	Bit name	Function	RW
— (b0)	Nothing is assigned. When write, set “0”. When read, its content is indeterminate.		—
CKPH	Clock phase set bit	0 : Without clock delay 1 : With clock delay	RW
— (b2)	Nothing is assigned. When write, set “0”. When read, its content is indeterminate.		—
NODC	Clock output select bit	0 : CLKi is CMOS output 1 : CLKi is N-channel open drain output	RW
— (b4)	Nothing is assigned. When write, set “0”. When read, its content is indeterminate.		—
DL0	SDAi digital delay setup bit (Note 1, Note 2)	b7 b6 b5 0 0 0 : Without delay 0 0 1 : 1 to 2 cycle(s) of UiBRG count source 0 1 0 : 2 to 3 cycles of UiBRG count source 0 1 1 : 3 to 4 cycles of UiBRG count source 1 0 0 : 4 to 5 cycles of UiBRG count source 1 0 1 : 5 to 6 cycles of UiBRG count source 1 1 0 : 6 to 7 cycles of UiBRG count source 1 1 1 : 7 to 8 cycles of UiBRG count source	RW
DL1			RW
DL2			RW

Note 1 : The DL2 to DL0 bits are used to generate a delay in SDAi output by digital means during I^2C mode. In other than I^2C mode, set these bits to “0002” (no delay).
Note 2 : The amount of delay varies with the load on SCLi and SDAi pins. Also, when using an external clock, the amount of delay increases by about 100 ns.

Figure 1.17.7. U0SMR2 to U2SMR2 Registers and U0SMR3 to U2SMR3 Registers

UARTi special mode register 4 (i=0 to 2)



Symbol	Address	After reset
U0SMR4 to U2SMR4	$036C_{16}$, 0370_{16}, 0374_{16}	00_{16}

Bit symbol	Bit name	Function	RW
STAREQ	Start condition generate bit (Note)	0 : Clear 1 : Start	RW
RSTAREQ	Restart condition generate bit (Note)	0 : Clear 1 : Start	RW
STPREQ	Stop condition generate bit (Note)	0 : Clear 1 : Start	RW
STSPSEL	SCL,SDA output select bit	0 : Start and stop conditions not output 1 : Start and stop conditions output	RW
ACKD	ACK data bit	0 : ACK 1 : NACK	RW
ACKC	ACK data output enable bit	0 : Serial I/O data output 1 : ACK data output	RW
SCLHI	SCL output stop enable bit	0 : Disabled 1 : Enabled	RW
SWC9	SCL wait bit 3	0 : SCL “L” hold disabled 1 : SCL “L” hold enabled	RW

Note: Set to “0” when each condition is generated.

Figure 1.17.8. U0SMR4 to U2SMR4 Registers

Clock Synchronous serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Table 1.18.1 lists the specifications of the clock synchronous serial I/O mode. Table 1.18.2 lists the registers used in clock synchronous serial I/O mode and the register values set.

Table 1.18.1. Clock Synchronous Serial I/O Mode Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• UiMR(i=0 to 2) register's CKDIR bit = "0" (internal clock) : fj/ 2(n+1) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of UiBRG register 00_{16} to FF_{16} • CKDIR bit = "1" (external clock) : Input from CLKi pin
Transmission, reception control	• Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS}/\overline{RTS}$ function disable
Transmission start condition	• Before transmission can start, the following requirements must be met (Note 1) – The TE bit of UiC1 register= 1 (transmission enabled) – The TI bit of UiC1 register = 0 (data present in UiTB register) – If $\overline{CTS}$ function is selected, input on the $\overline{CTS}$i pin = "L"
Reception start condition	• Before reception can start, the following requirements must be met (Note 1) – The RE bit of UiC1 register= 1 (reception enabled) – The TE bit of UiC1 register= 1 (transmission enabled) – The TI bit of UiC1 register= 0 (data present in the UiTB register)
Interrupt request generation timing	• For transmission, one of the following conditions can be selected – The UiIRS bit (Note 3) = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) – The UiIRS bit =1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register • For reception When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	• Overrun error (Note 2) This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 7th bit of the next data
Select function	• CLK polarity selection Transfer data input/output can be chosen to occur synchronously with the rising or the falling edge of the transfer clock • LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Continuous receive mode selection Reception is enabled immediately by reading the UiRB register • Switching serial data logic This function reverses the logic value of the transmit/receive data • Transfer clock output from multiple pins selection (UART1) The output pin can be selected in a program from two UART1 transfer clock pins that have been set • Separate $\overline{CTS}/\overline{RTS}$ pins (UART0) $\overline{CTS_0}$ and $\overline{RTS_0}$ are input/output from separate pins

Note 1: When an external clock is selected, the conditions must be met while if the UiC0 register's CKPOL bit = "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the UiC0 register's CKPOL bit = "1" (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
Note 2: If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit of SiRIC register does not change.
Note 3: The U0IRS and U1IRS bits respectively are the UCON register bits 0 and 1; the U2IRS bit is the U2C1 register bit 4.

Table 1. 18. 2. Registers to Be Used and Settings in Clock Synchronous Serial I/O Mode

Register	Bit	Function
UiTB(Note3)	0 to 7	Set transmission data
UiRB(Note3)	0 to 7	Reception data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a transfer rate
UiMR(Note3)	SMD2 to SMD0	Set to “001_2”
	CKDIR	Select the internal clock or external clock
	IOPOL	Set to “0”
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Select $\overline{CTS}$ or $\overline{RTS}$ to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the $\overline{CTS}$ or $\overline{RTS}$ function
	NCH	Select TxDi pin output mode (Note 2)
	CKPOL	Select the transfer clock polarity
	UFORM	Select the LSB first or MSB first
UiC1	TE	Set this bit to “1” to enable transmission/reception
	TI	Transmit buffer empty flag
	RE	Set this bit to “1” to enable reception
	RI	Reception complete flag
	U2IRS (Note 1)	Select the source of UART2 transmit interrupt
	U2RRM (Note 1)	Set this bit to “1” to use continuous receive mode
	UiLCH	Set this bit to “1” to use inverted data logic
	UiERE	Set to “0”
UiSMR	0 to 7	Set to “0”
UiSMR2	0 to 7	Set to “0”
UiSMR3	0 to 2	Set to “0”
	NODC	Select clock output mode
	4 to 7	Set to “0”
UiSMR4	0 to 7	Set to “0”
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set this bit to “1” to use continuous receive mode
	CLKMD0	Select the transfer clock output pin when CLKMD1 = 1
	CLKMD1	Set this bit to “1” to output UART1 transfer clock from two pins
	RCSP	Set this bit to “1” to accept as input the UART0 $\overline{CTS_0}$ signal from the $P6_4$ pin
	7	Set to “0”

Note 1: Set the U0C1 and U1C1 register bit 4 and bit 5 to “0”. The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.

Note 2: TxD2 pin is N channel open-drain output. Set the U2C0 register's NCH bit to “0”.

Note 3: Not all register bits are described above. Set those bits to “0” when writing to the registers in clock synchronous serial I/O mode.

i=0 to 2

Table 1.18.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. Table 1.18.3 shows pin functions for the case where the multiple transfer clock output pin select function is deselected. Table 1.18.4 lists the P6$_4$ pin functions during clock synchronous serial I/O mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an “H”. (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

Table 1.18.3. Pin Functions (When Not Select Multiple Transfer Clock Output Pin Function)

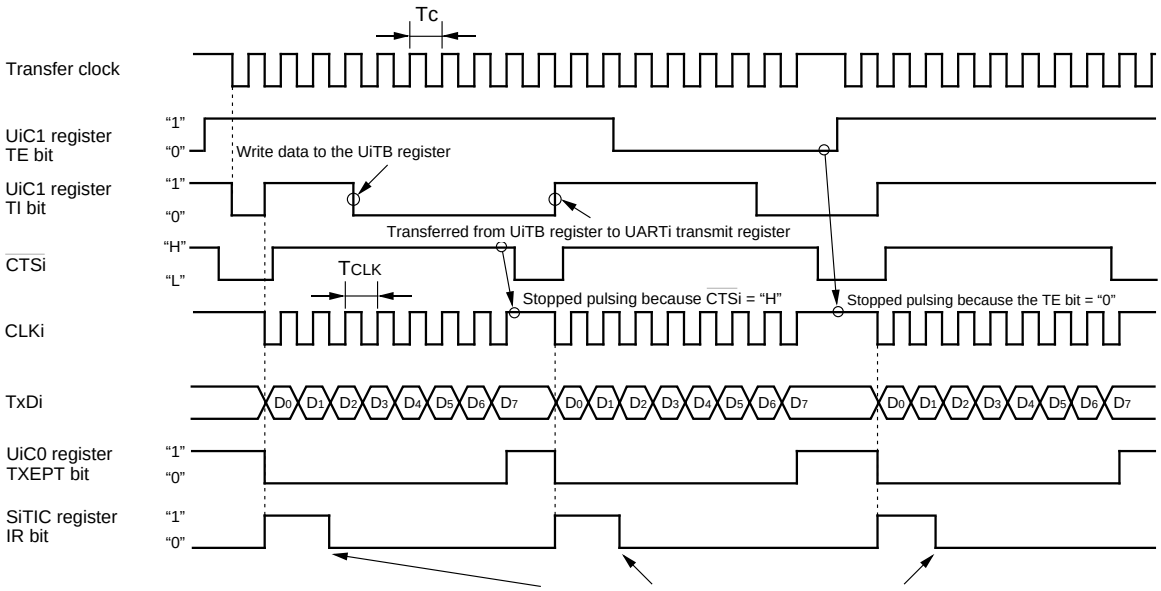
Pin name	Function	Method of selection
TxDi (i = 0 to 2) (P6$_3$, P6$_7$, P7$_0$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P6$_2$, P6$_6$, P7$_1$)	Serial data input	PD6 register’s PD6_2 bit=0, PD6_6 bit=0, PD7 register’s PD7_1 bit=0 (Can be used as an input port when performing transmission only)
CLKi (P6$_1$, P6$_5$, P7$_2$)	Transfer clock output	UiMR register’s CKDIR bit=0
	Transfer clock input	UiMR register’s CKDIR bit=1 PD6 register’s PD6_1 bit=0, PD6_5 bit=0, PD7 register’s PD7_2 bit=0
$\overline{\text{CTS}}$i/$\overline{\text{RTS}}$i (P6$_0$, P6$_4$, P7$_3$)	$\overline{\text{CTS}}$ input	UiC0 register’s CRD bit=0 UiC0 register’s CRS bit=0 PD6 register’s PD6_0 bit=0, PD6_4 bit=0, PD7 register’s PD7_3 bit=0
	$\overline{\text{RTS}}$ output	UiC0 register’s CRD bit=0 UiC0 register’s CRS bit=1
	I/O port	UiC0 register’s CRD bit=1

Table 1.18.4. P6$_4$ Pin Functions

Pin function	Bit set value					
	U1C0 register		UCON register			PD6 register
	CRD	CRS	RCSP	CLKMD1	CLKMD0	PD6_4
P6$_4$	1	—	0	0	—	Input: 0, Output: 1
$\overline{\text{CTS}}_1$	0	0	0	0	—	0
$\overline{\text{RTS}}_1$	0	1	0	0	—	—
$\overline{\text{CTS}}_0$(Note1)	0	0	1	0	—	0
CLKS$_1$	—	—	—	1(Note 2)	1	—

Note 1: In addition to this, set the U0C0 register’s CRD bit to “0” ($\overline{\text{CTS}}_0$/$\overline{\text{RTS}}_0$ enabled) and the U0 C0 register’s CRS bit to “1” ($\overline{\text{RTS}}_0$ selected).
Note 2: When the CLKMD1 bit = 1 and the CLKMD0 bit = 0, the following logic levels are output:
• High if the U1C0 register’s CLKPOL bit = 0
• Low if the U1C0 register’s CLKPOL bit = 1

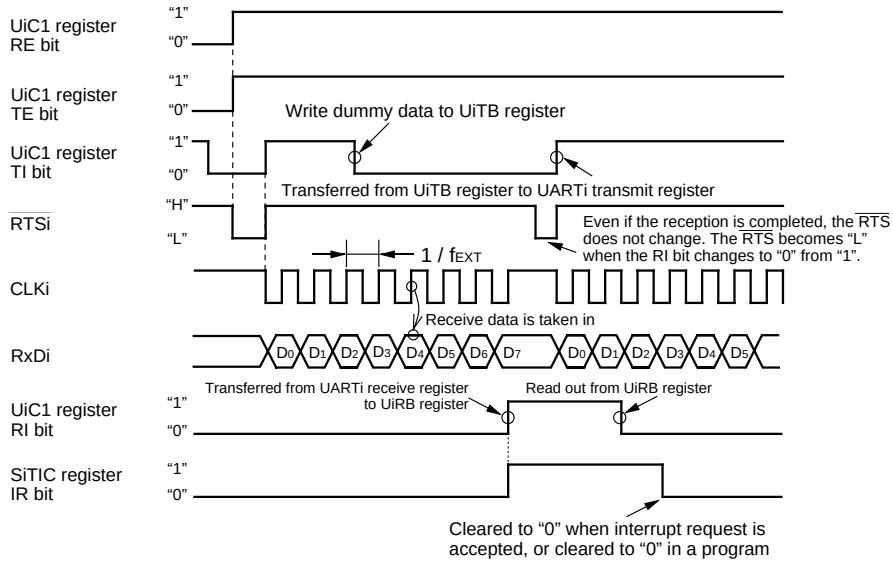
(1) Example of transmit timing (when internal clock is selected)



Tc = TCLK = 2(n + 1) / fj
fj: frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
n: value set to UiBRG register
i: 0 to 2

The above timing diagram applies to the case where the register bits are set as follows:
- UiMR register CKDIR bit = 0 (internal clock)
- UiC0 register CRD bit = 0 (CTS/RTS enabled), CRS bit = 0 (CTS selected)
- UiC0 register CKPOL bit = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)
- UiRS bit = 0 (an interrupt request occurs when the transmit buffer becomes empty): U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

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



The above timing diagram applies to the case where the register bits are set as follows:
- UiMR register CKDIR bit = 1 (external clock)
- UiC0 register CRD bit = 0 (CTS/RTS enabled), CRS bit = 1 (RTS selected)
- UiC0 register CKPOL bit = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)

Make sure the following conditions are met when input to the CLKi pin before receiving data is high:
- UiC0 register TE bit = 1 (transmit enabled)
- UiC0 register RE bit = 1 (Receive enabled)
- Write dummy data to the UiTB register

fEXT: frequency of external clock

Figure 1.18.1. Transmit and Receive Operation

(a) CLK Polarity Select Function

Use the UiC0 register (i = 0 to 2)'s CKPOL bit to select the transfer clock polarity. Figure 1.18.2 shows the polarity of the transfer clock.

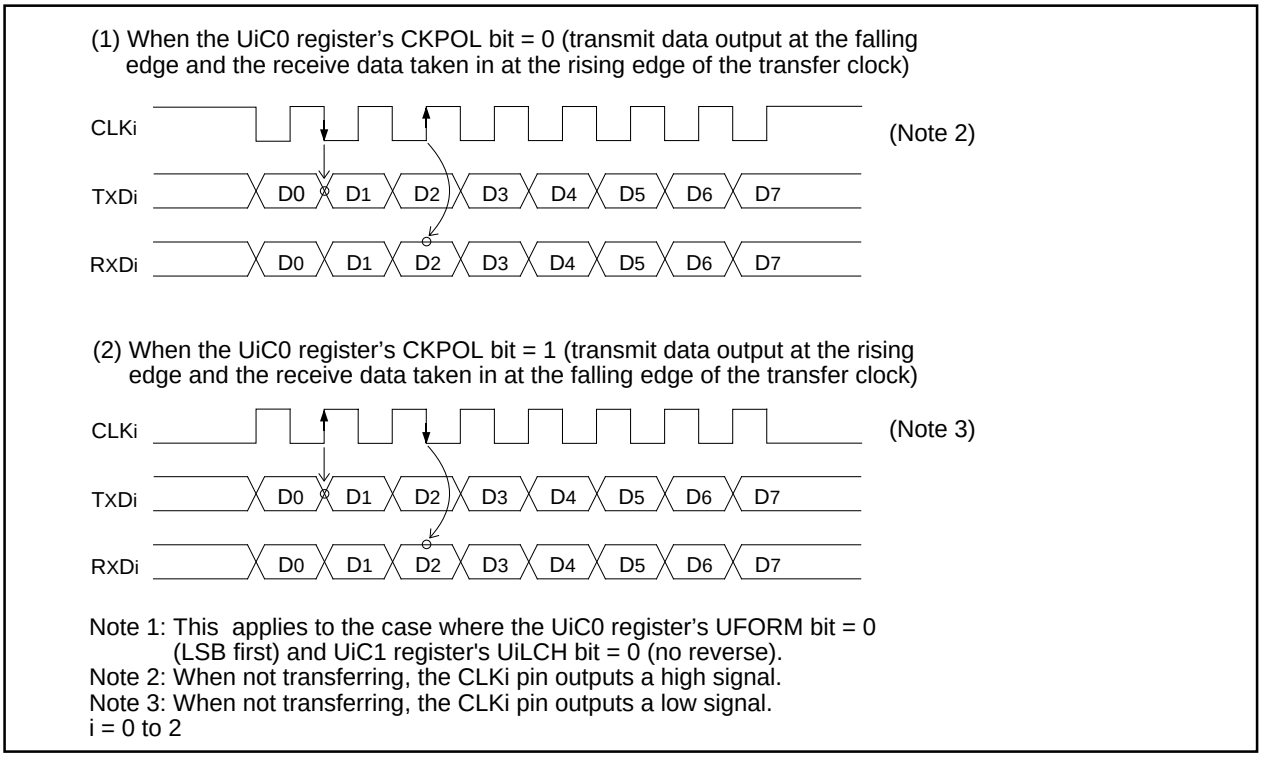


Figure 1.18.2. Transfer Clock Polarity

(b) LSB First/MSB First Select Function

Use the UiC0 register (i = 0 to 2)'s UFORM bit to select the transfer format. Figure 1.18.3 shows the transfer format.

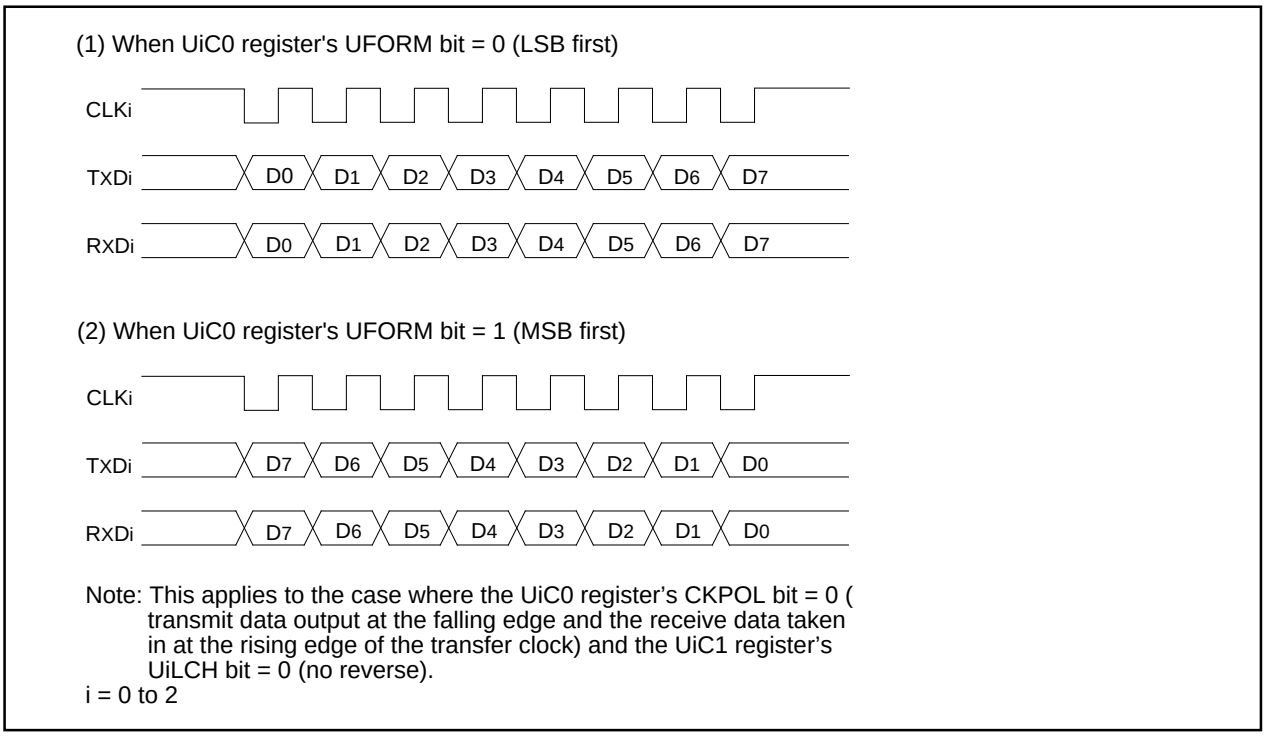


Figure 1.18.3. Transfer Format

(c) Continuous Receive Mode

When the UiRRM bit (i = 0 to 2) = 1 (continuous receive mode), the UiC1 register's TI bit is set to "1" (data present in the UiTB register) by reading the UiRB register. In this case, i.e., UiRRM bit = 1, do not write dummy data to the UiTB register in a program. The U0RRM and U1RRM bits are the UCON register bit 2 and bit 3, respectively, and the U2RRM bit is the U2C1 register bit 4.

(d) Serial Data Logic Switching Function

When the UiC1 register (i = 0 to 2)'s UiLCH bit = 1 (reverse), the data written to the UiTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the UiRB register. Figure 1.18.4 shows serial data logic.

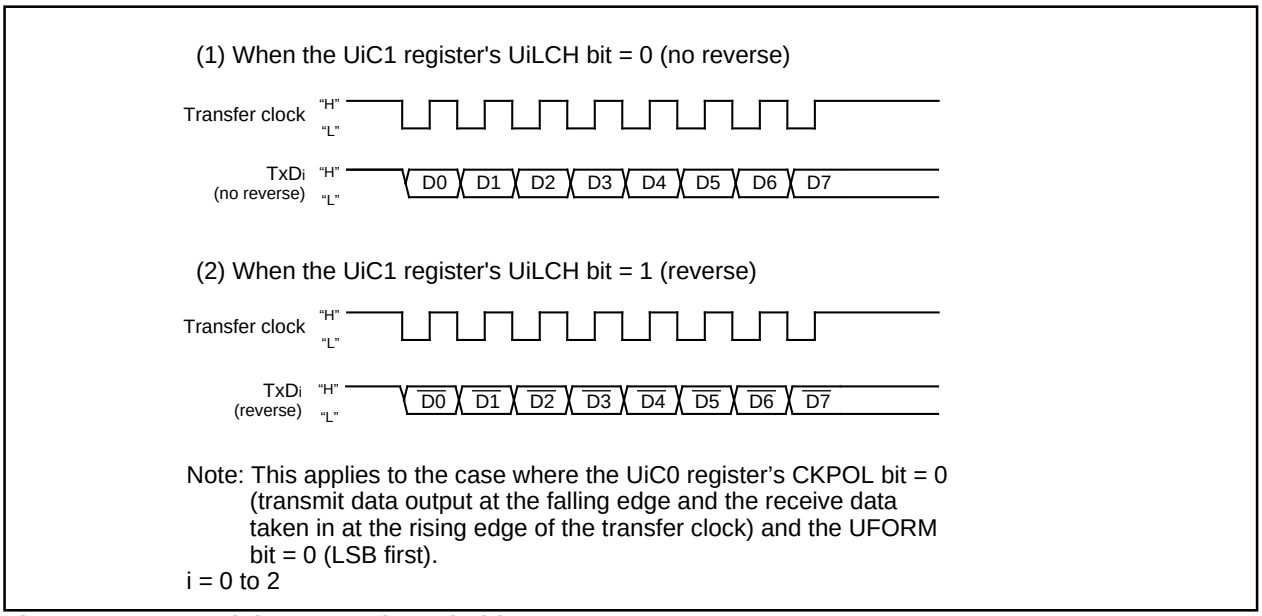


Figure 1.18.4. Serial Data Logic Switching

(e) Transfer Clock Output From Multiple Pins (UART1)

Use the UCON register's CLKMD1 to CLKMD0 bits to select one of the two transfer clock output pins. (See Figure 1.18.5.) This function can be used when the selected transfer clock for UART1 is an internal clock.

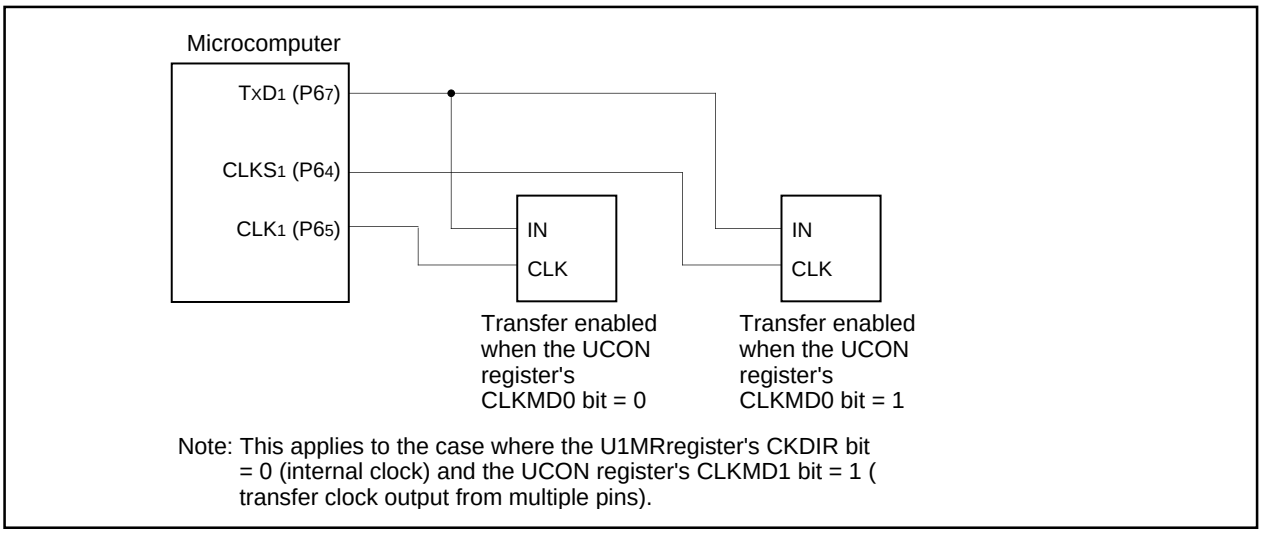


Figure 1.18.5. Transfer Clock Output From Multiple Pins

(f) $\overline{CTS}/\overline{RTS}$ Separate Function (UART0)

This function separates $\overline{CTS}_0/\overline{RTS}_0$, outputs $\overline{RTS}_0$ from the $P6_0$ pin, and accepts as input the $\overline{CTS}_0$ from the $P6_4$ pin. To use this function, set the register bits as shown below.

- U0C0 register's CRD bit = 0 (enables UART0 $\overline{CTS}/\overline{RTS}$)
- U0C0 register's CRS bit = 1 (outputs UART0 $\overline{RTS}$)
- U1C0 register's CRD bit = 0 (enables UART1 $\overline{CTS}/\overline{RTS}$)
- U1C0 register's CRS bit = 0 (inputs UART1 $\overline{CTS}$)
- UCON register's RCSP bit = 1 (inputs $\overline{CTS}_0$ from the $P6_4$ pin)
- UCON register's CLKMD1 bit = 0 ($CLKS_1$ not used)

Note that when using the $\overline{CTS}/\overline{RTS}$ separate function, UART1 $\overline{CTS}/\overline{RTS}$ separate function cannot be used.

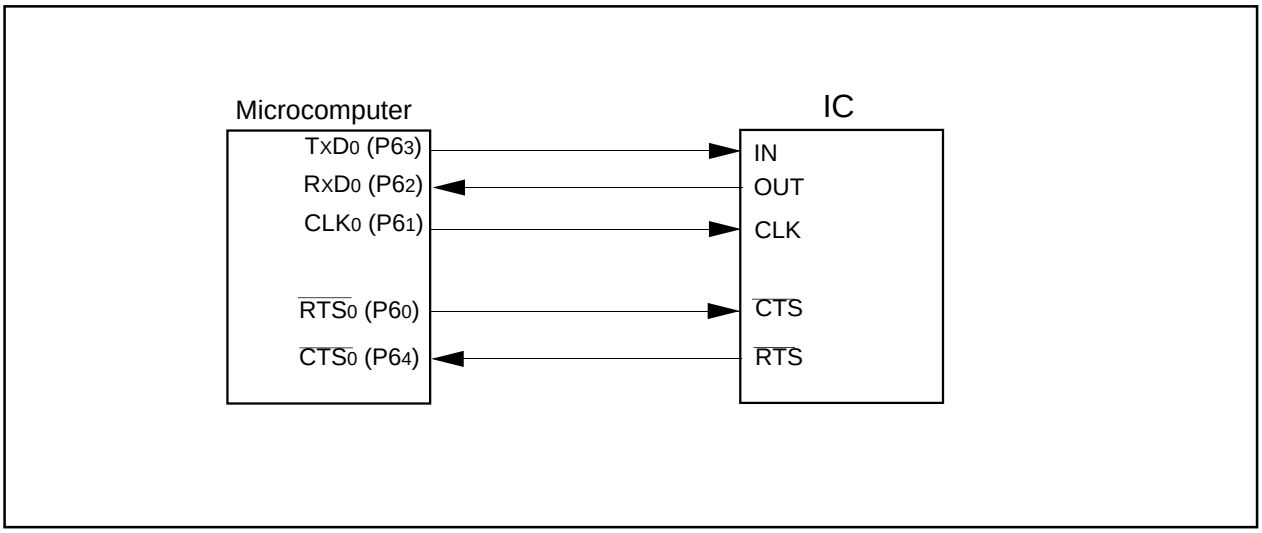


Figure 1.18.6. $\overline{CTS}/\overline{RTS}$ Separat Function

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 1.19.1 lists the specifications of the UART mode.

Table 1.19.1. UART Mode Specifications

Item	Specification
Transfer data format	• Character bit (transfer data): Selectable from 7, 8 or 9 bits • Start bit: 1 bit • Parity bit: Selectable from odd, even, or none • Stop bit: Selectable from 1 or 2 bits
Transfer clock	• UiMR(i=0 to 2) register's CKDIR bit = 0 (internal clock) : fj/ 16(n+1) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of UiBRG register 00_{16} to FF_{16} • CKDIR bit = "1" (external clock) : f_{EXT}/16(n+1) f_{EXT}: Input from CLKi pin. n :Setting value of UiBRG register 00_{16} to FF_{16}
Transmission, reception control	• Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS}/\overline{RTS}$ function disable
Transmission start condition	• Before transmission can start, the following requirements must be met – The TE bit of UiC1 register= 1 (transmission enabled) – The TI bit of UiC1 register = 0 (data present in UiTB register) – If $\overline{CTS}$ function is selected, input on the $\overline{CTS}$i pin = "L"
Reception start condition	• Before reception can start, the following requirements must be met – The RE bit of UiC1 register= 1 (reception enabled) – Start bit detection
Interrupt request generation timing	• For transmission, one of the following conditions can be selected – The UiIRS bit (Note 2) = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) – The UiIRS bit =1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register • For reception When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	• Overrun error (Note 1) This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the bit one before the last stop bit of the next data • Framing error This error occurs when the number of stop bits set is not detected • Parity error This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set • Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered
Select function	• LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Serial data logic switch This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed. • TxD, RxD I/O polarity switch This function reverses the polarities of hte TxD pin output and RxD pin input. The logic levels of all I/O data is reversed. • Separate $\overline{CTS}/\overline{RTS}$ pins (UART0) $\overline{CTS_0}$ and $\overline{RTS_0}$ are input/output from separate pins

Note 1: If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit of SiRIC register does not change.
Note 2: The U0IRS and U1IRS bits respectively are the UCON register bits 0 and 1; the U2IRS bit is the U2C1 register bit 4.

Table 1. 19. 2. Registers to Be Used and Settings in UART Mode

Register	Bit	Function
UiTB	0 to 8	Set transmission data (Note 1)
UiRB	0 to 8	Reception data can be read (Note 1)
	OER,FER,PER,SUM	Error flag
UiBRG	---	Set a transfer rate
UiMR	SMD2 to SMD0	Set these bits to '100$_2$' when transfer data is 7 bits long Set these bits to '101$_2$' when transfer data is 8 bits long Set these bits to '110$_2$' when transfer data is 9 bits long
	CKDIR	Select the internal clock or external clock
	STPS	Select the stop bit
	PRY, PRYE	Select whether parity is included and whether odd or even
	IOPOL	Select the TxD/RxD input/output polarity
UiC0	CLK0, CLK1	Select the count source for the UiBRG register
	CRS	Select $\overline{CTS}$ or $\overline{RTS}$ to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the $\overline{CTS}$ or $\overline{RTS}$ function
	NCH	Select TxDi pin output mode (Note 2)
	CKPOL	Set to "0"
	UFORM	LSB first or MSB first can be selected when transfer data is 8 bits long. Set this bit to "0" when transfer data is 7 or 9 bits long.
UiC1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS (Note 2)	Select the source of UART2 transmit interrupt
	U2RRM (Note 2)	Set to "0"
	UiLCH	Set this bit to "1" to use inverted data logic
	UiERE	Set to "0"
UiSMR	0 to 7	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set to "0"
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1	Set to "0"
	RCSP	Set this bit to "1" to accept as input the UART0 $\overline{CTS_0}$ signal from the P6$_4$ pin
	7	Set to "0"

Note 1: The bits used for transmit/receive data are as follows: Bit 0 to bit 6 when transfer data is 7 bits long; bit 0 to bit 7 when transfer data is 8 bits long; bit 0 to bit 8 when transfer data is 9 bits long.

Note 2: Set the U0C1 and U1C1 registers bit 4 to bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are included in the UCON register.

Note 3: TxD2 pin is N channel open-drain output. Set the U2C0 register's NCH bit to "0".

i=0 to 2

Table 1.19.3 lists the functions of the input/output pins during UART mode. Table 1.19.4 lists the P6$_4$ pin functions 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 output is selected, this pin is in a high-impedance state.)

Table 1.19.3. I/O Pin Functions

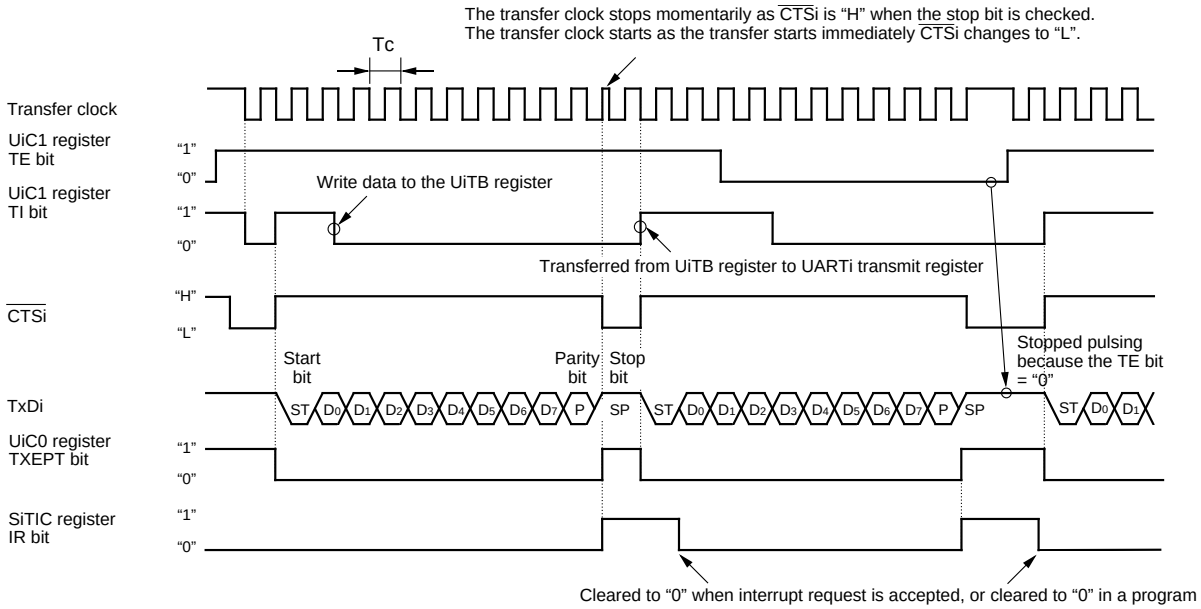
Pin name	Function	Method of selection
TxDi (i = 0 to 2) (P6$_3$, P6$_7$, P7$_0$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P6$_2$, P6$_6$, P7$_1$)	Serial data input	PD6 register’s PD6_2 bit=0, PD6_6 bit=0, PD7 register’s PD7_1 bit=0 (Can be used as an input port when performing transmission only)
CLKi (P6$_1$, P6$_5$, P7$_2$)	Input/output port	UiMR register’s CKDIR bit=0
	Transfer clock input	UiMR register’s CKDIR bit=1 PD6 register’s PD6_1 bit=0, PD6_5 bit=0, PD7 register’s PD7_2 bit=0
$\overline{\text{CTS}}$i/$\overline{\text{RTS}}$i (P6$_0$, P6$_4$, P7$_3$)	$\overline{\text{CTS}}$ input	UiC0 register’s CRD bit=0 UiC0 register’s CRS bit=0 PD6 register’s PD6_0 bit=0, PD6_4 bit=0, PD7 register’s PD7_3 bit=0
	$\overline{\text{RTS}}$ output	UiC0 register’s CRD bit=0 UiC0 register’s CRS bit=1
	Input/output port	UiC0 register’s CRD bit=1

Table 1.19.4. P6$_4$ Pin Functions

Pin function	Bit set value				
	U1C0 register		UCON register		PD6 register
	CRD	CRS	RCSP	CLKMD1	PD6_4
P6$_4$	1	—	0	0	Input: 0, Output: 1
$\overline{\text{CTS}}_1$	0	0	0	0	0
$\overline{\text{RTS}}_1$	0	1	0	0	—
$\overline{\text{CTS}}_0$ (Note)	0	0	1	0	0

Note: In addition to this, set the U0C0 register’s CRD bit to “0” ($\overline{\text{CTS}}_0$/$\overline{\text{RTS}}_0$ enabled) and the U0C0 register’s CRS bit to “1” ($\overline{\text{RTS}}_0$ selected).

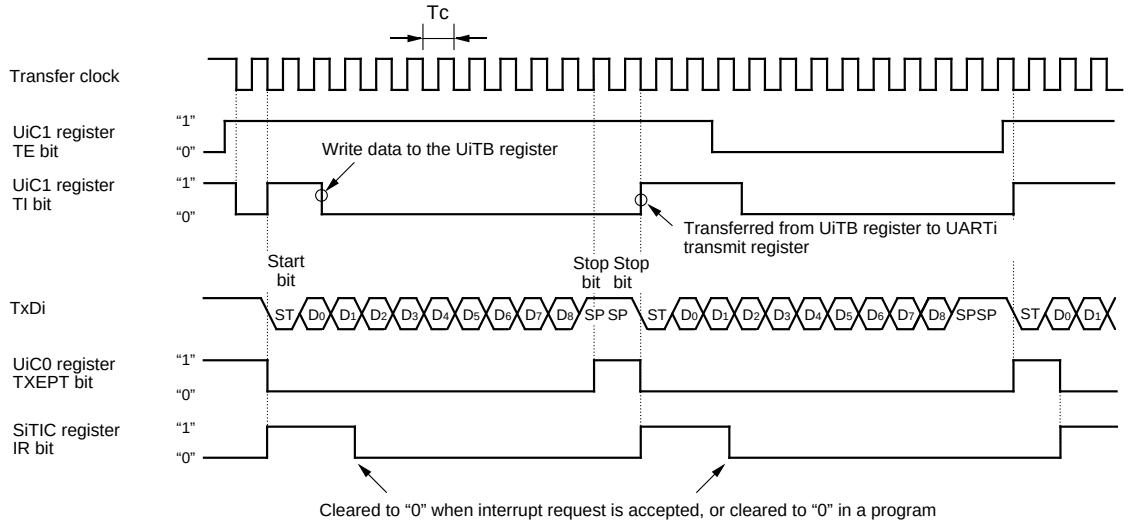
(1) Example of transmit timing when transfer data is 8 bits long (parity enabled, one stop bit)



The above timing diagram applies to the case where the register bits are set as follows:
- UiMR register PRYE bit = 1 (parity enabled)
- UiMR register STPS bit = 0 (1 stop bit)
- UiC0 register CRD bit = 0 (CTS/RTS enabled), CRS bit = 0 (CTS selected)
- UiRS bit = 1 (an interrupt request occurs when transmit completed):
  U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

$Tc = 16 (n + 1) / fj$ or $16 (n + 1) / f_{EXT}$
fj : frequency of UiBRG count source ($f1_{SIO}$, $f2_{SIO}$, $f8_{SIO}$, $f32_{SIO}$)
f_{EXT} : frequency of UiBRG count source (external clock)
n : value set to UiBRG
i: 0 to 2

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

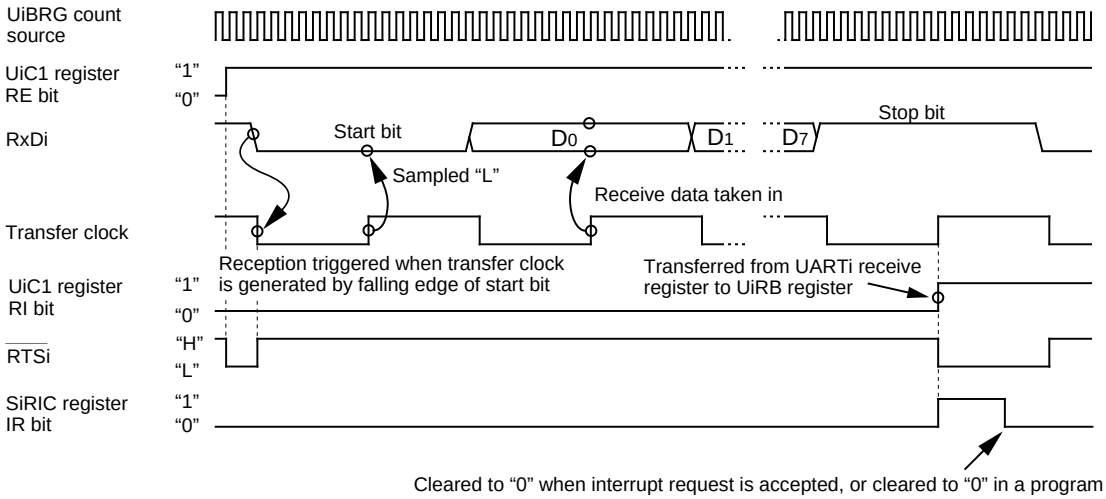


The above timing diagram applies to the case where the register bits are set as follows:
- UiMR register PRYE bit = 0 (parity disabled)
- UiMR register STPS bit = 1 (2 stop bits)
- UiC0 register CRD bit = 1 (CTS/RTS disabled)
- UiRS bit = 0 (an interrupt request occurs when transmit buffer becomes empty):
  U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

$Tc = 16 (n + 1) / fj$ or $16 (n + 1) / f_{EXT}$
fj : frequency of UiBRG count source ($f1_{SIO}$, $f2_{SIO}$, $f8_{SIO}$, $f32_{SIO}$)
f_{EXT} : frequency of UiBRG count source (external clock)
n : value set to UiBRG
i: 0 to 2

Figure 1.19.1. Transmit Operation

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



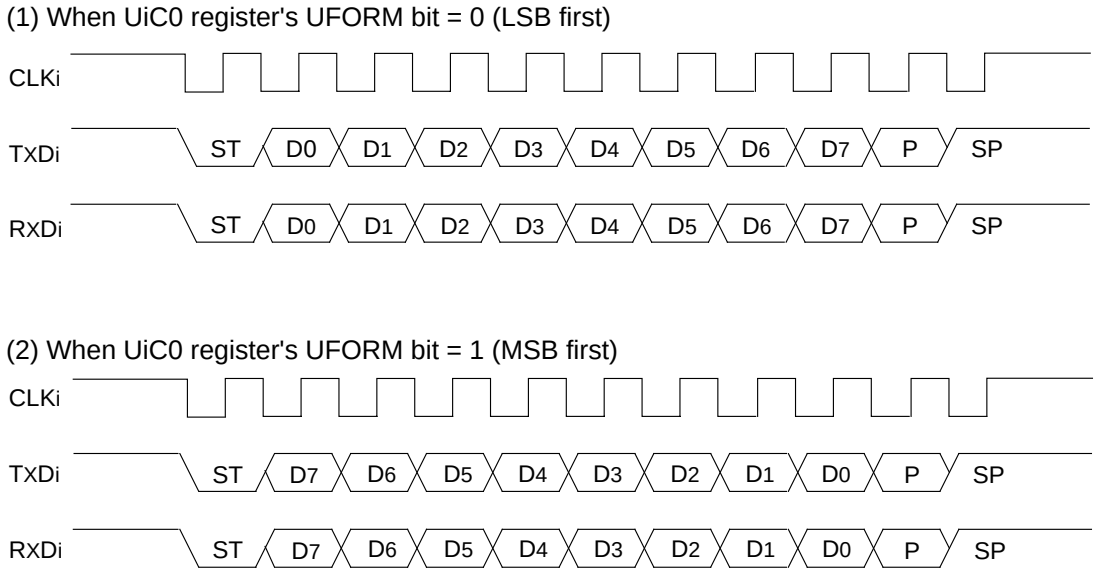
The above timing diagram applies to the case where the register bits are set as follows:
- UiMR register PRYE bit = 0 (parity disabled)
- UiMR register STPS bit = 0 (1 stop bit)
- UiC0 register CRD bit = 0 (CTSi/RTSi enabled), CRS bit = 1 (RTSi selected)

i = 0 to 2

Figure 1.19.2. Receive Operation

(a) LSB First/MSB First Select Function

As shown in Figure 1.19.3, use the UiC0 register's UFORM bit to select the transfer format. This function is valid when transfer data is 8 bits long.



Note: This applies to the case where the UiC0 register's CKPOL bit = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the UiC1 register's UiLCH bit = 0 (no reverse), UiMR register's STPS bit = 0 (1 stop bit) and UiMR register's PRYE bit = 1 (parity enabled).

ST : Start bit
P : Parity bit
SP : Stop bit
i = 0 to 2

Figure 1.19.3. Transfer Format

(b) Serial Data Logic Switching Function

The data written to the UiTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the UiRB register. Figure 1.19.4 shows serial data logic.

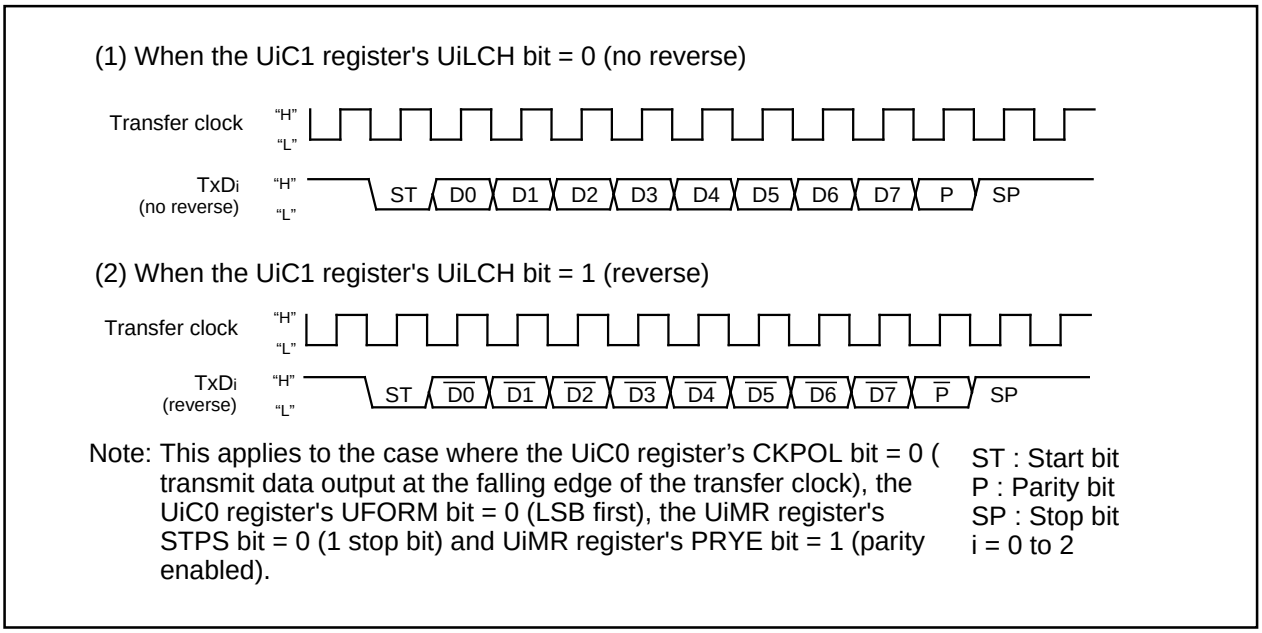


Figure 1.19.4. Serial Data Logic Switching

(c) TxD and RxD I/O Polarity Inverse Function

This function inverses the polarities of the TxDi pin output and RxDi pin input. The logic levels of all input/output data (including the start, stop and parity bits) are inversed. Figure 1.19.5 shows the TxD pin output and RxD pin input polarity inverse.

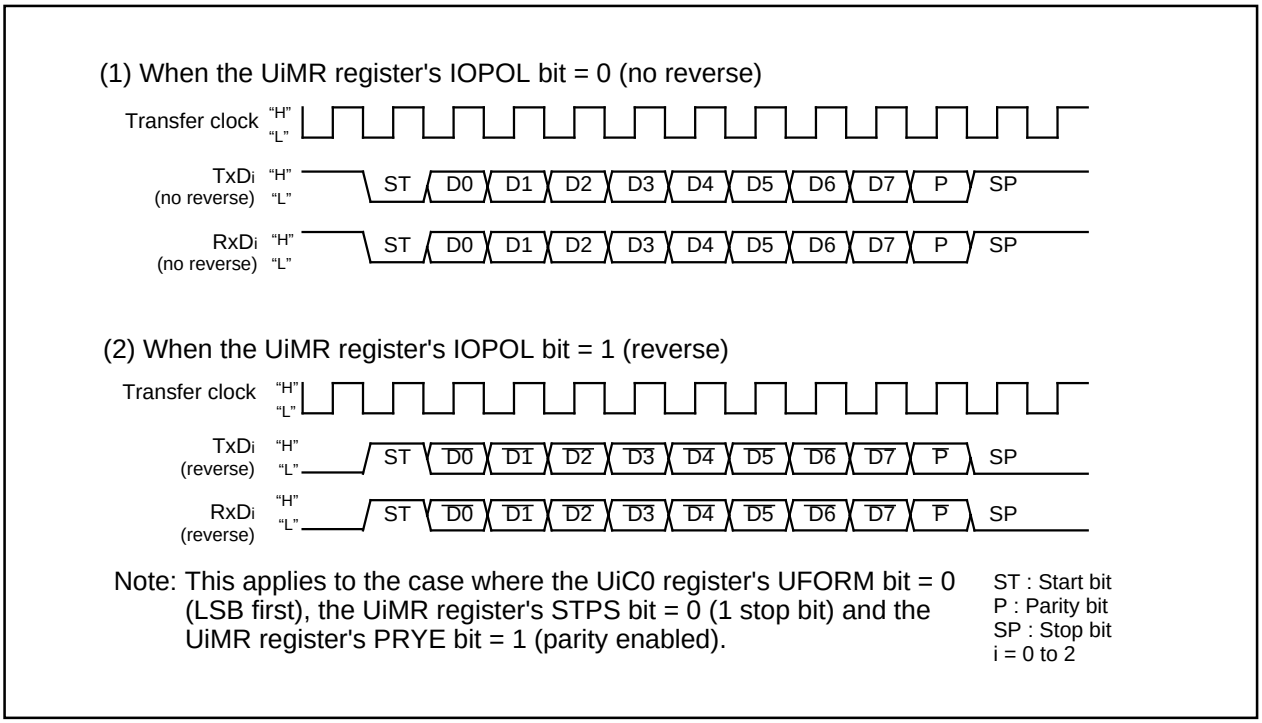


Figure 1.19.5. TxD and RxD I/O Polarity Inverse

(d) $\overline{CTS}/\overline{RTS}$ Separate Function (UART0)

This function separates $\overline{CTS}_0/\overline{RTS}_0$, outputs $\overline{RTS}_0$ from the $P6_0$ pin, and accepts as input the $\overline{CTS}_0$ from the $P6_4$ pin. To use this function, set the register bits as shown below.

- U0C0 register's CRD bit = 0 (enables UART0 $\overline{CTS}/\overline{RTS}$)
- U0C0 register's CRS bit = 1 (outputs UART0 $\overline{RTS}$)
- U1C0 register's CRD bit = 0 (enables UART1 $\overline{CTS}/\overline{RTS}$)
- U1C0 register's CRS bit = 0 (inputs UART1 $\overline{CTS}$)
- UCON register's RCSP bit = 1 (inputs $\overline{CTS}_0$ from the $P6_4$ pin)
- UCON register's CLKMD1 bit = 0 ($CLKS_1$ not used)

Note that when using the $\overline{CTS}/\overline{RTS}$ separate function, UART1 $\overline{CTS}/\overline{RTS}$ separate function cannot be used.

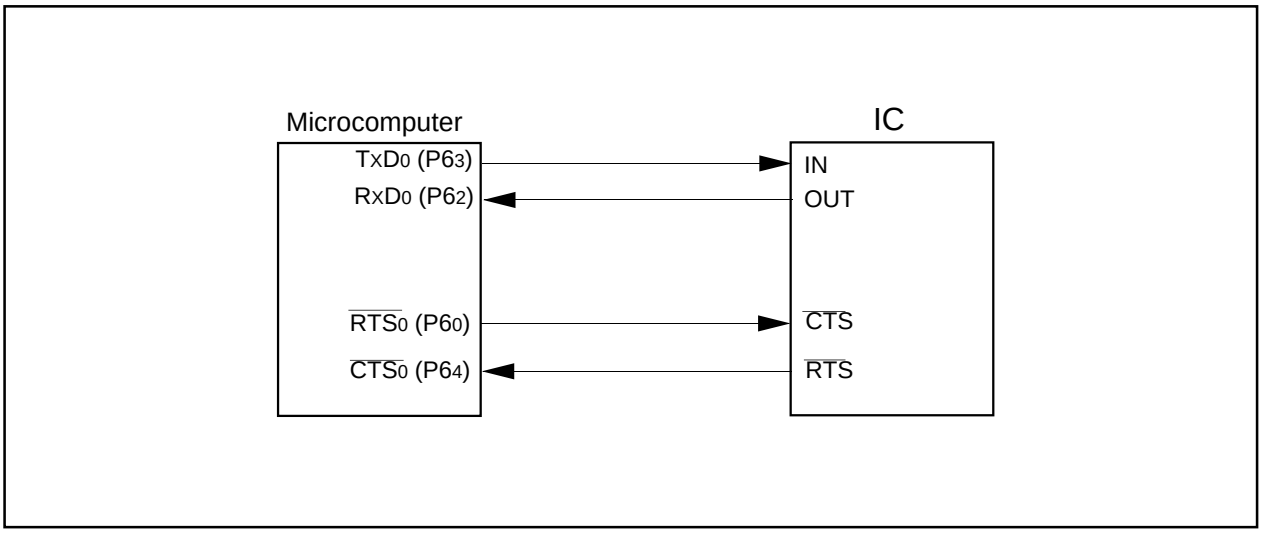


Figure 1.19.6. $\overline{CTS}/\overline{RTS}$ Separate Function

Special Mode 1 (I^2C mode)

I^2C mode is provided for use as a simplified I^2C interface compatible mode. Table 1.20.1 lists the specifications of the I^2C mode. Table 1.20.2 lists the registers used in the I^2C mode and the register values set. Figure 1.20.1 shows the block diagram for I^2C mode. Figure 1.20.2 shows SCLi timing.

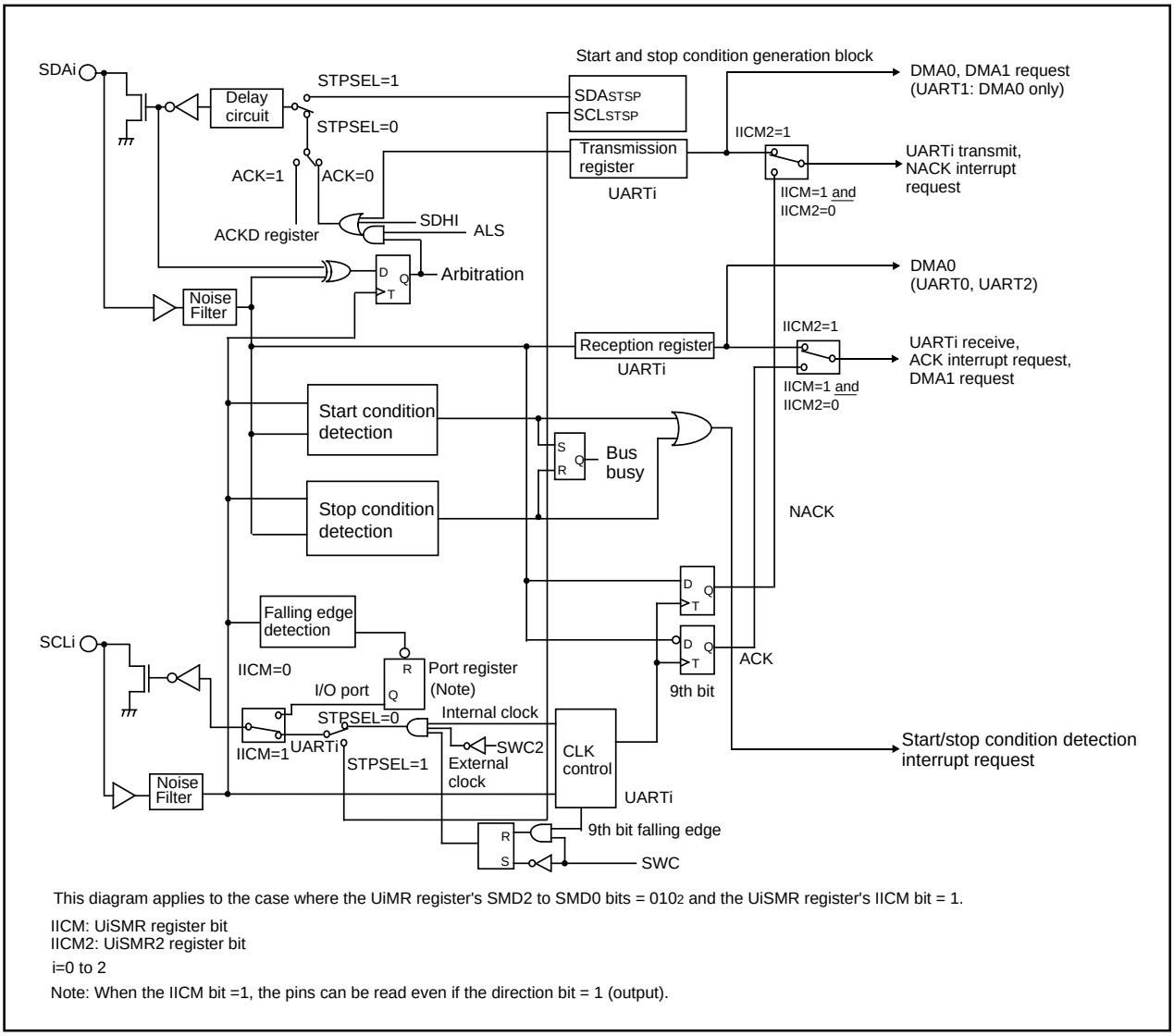
As shown in Table 1.20.3, the microcomputer is placed in I^2C mode by setting the SMD2 to SMD0 bits to '010_2' and the IICM bit to "1". Because SDAi transmit output has a delay circuit attached, SDAi output does not change state until SCLi goes low and remains stably low.

Table 1.20.1. I^2C Mode Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• During master UiMR(i=0 to 2) register's CKDIR bit = "0" (internal clock) : fj/ 2(n+1) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of UiBRG register 00_{16} to FF_{16} • During slave CKDIR bit = "1" (external clock) : Input from CLKi pin
Transmission start condition	• Before transmission can start, the following requirements must be met (Note 1) – The TE bit of UiC1 register= 1 (transmission enabled) – The TI bit of UiC1 register = 0 (data present in UiTB register)
Reception start condition	• Before reception can start, the following requirements must be met (Note 1) – The RE bit of UiC1 register= 1 (reception enabled) – The TE bit of UiC1 register= 1 (transmission enabled) – The TI bit of UiC1 register= 0 (data present in the UiTB register)
Interrupt request generation timing	When start or stop condition is detected, acknowledge undetected, and acknowledge detected
Error detection	• Overrun error (Note 2) This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 8th bit of the next data
Select function	• Arbitration lost Timing at which the UiRB register's ABT bit is updated can be selected • SDAi digital delay No digital delay or a delay of 2 to 8 UiBRG count source clock cycles selectable • Clock phase setting With or without clock delay selectable

Note 1: When an external clock is selected, the conditions must be met while the external clock is in the high state.

Note 2: If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit of SiRIC register does not change.



This diagram applies to the case where the UiMR register's SMD2 to SMD0 bits = 010$_2$ and the UiSMR register's IICM bit = 1.

IICM: UiSMR register bit
IICM2: UiSMR2 register bit

i=0 to 2

Note: When the IICM bit =1, the pins can be read even if the direction bit = 1 (output).

Figure 1.20.1. I^{2}C Mode Block Diagram

Table 1. 20. 2. Registers to Be Used and Settings in I²C Mode (1) (Continued)

Register	Bit	Function	
		Master	Slave
UiTB[3]	0 to 7	Set transmission data	Set transmission data
UiRB[3]	0 to 7	Reception data can be read	Reception data can be read
	8	ACK or NACK is set in this bit	ACK or NACK is set in this bit
	ABT	Arbitration lost detection flag	Invalid
	OER	Overrun error flag	Overrun error flag
UiBRG	---	Set a transfer rate	Invalid
UiMR[3]	SMD2 to SMD0	Set to '010$_2$'	Set to '010$_2$'
	CKDIR	Set to "0"	Set to "1"
	IOPOL	Set to "0"	Set to "0"
UiC0	CLK1, CLK0	Select the count source for the UiBRG register	Invalid
	CRS	Invalid because CRD = 1	Invalid because CRD = 1
	TXEPT	Transmit buffer empty flag	Transmit buffer empty flag
	CRD	Set to "1"	Set to "1"
	NCH	Set to "1"[2]	Set to "1"[2]
	CKPOL	Set to "0"	Set to "0"
	UFORM	Set to "1"	Set to "1"
UiC1	TE	Set this bit to "1" to enable transmission	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception	Set this bit to "1" to enable reception
	RI	Reception complete flag	Reception complete flag
	U2IRS[1]	Invalid	Invalid
	U2RRM[1], UiLCH, UiERE	Set to "0"	Set to "0"
UiSMR	IICM	Set to "1"	Set to "1"
	ABC	Select the timing at which arbitration-lost is detected	Invalid
	BBS	Bus busy flag	Bus busy flag
	3 to 7	Set to "0"	Set to "0"
UiSMR2	IICM2	Refer to Table 1.20.4.	Refer to Table 1.20.4.
	CSC	Set this bit to "1" to enable clock synchronization	Set to "0"
	SWC	Set this bit to "1" to have SCLi output fixed to "L" at the falling edge of the 9th bit of clock	Set this bit to "1" to have SCLi output fixed to "L" at the falling edge of the 9th bit of clock
	ALS	Set this bit to "1" to have SDAi output stopped when arbitration-lost is detected	Set to "0"
	STAC	Set to "0"	Set this bit to "1" to initialize UARTi at start condition detection
	SWC2	Set this bit to "1" to have SCLi output forcibly pulled low	Set this bit to "1" to have SCLi output forcibly pulled low
	SDHI	Set this bit to "1" to disable SDAi output	Set this bit to "1" to disable SDAi output
	7	Set to "0"	Set to "0"
UiSMR3	0, 2, 4 and NODC	Set to "0"	Set to "0"
	CKPH	Refer to Table 1.20.4	Refer to Table 1.20.4
	DL2 to DL0	Set the amount of SDAi digital delay	Set the amount of SDAi digital delay

i=0 to 2

Notes:

1. Set the U0C1 and U1C1 register bit 4 and bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.
2. TxD2 pin is N channel open-drain output. Set the NCH bit in the U2C0 register to "0".
3. Not all register bits are described above. Set those bits to "0" when writing to the registers in I²C mode.

Table 1. 20. 3. Registers to Be Used and Settings in I²C Mode (2) (Continued)

Register	Bit	Function	
		Master	Slave
UiSMR4	STAREQ	Set this bit to “1” to generate start condition	Set to “0”
	RSTAREQ	Set this bit to “1” to generate restart condition	Set to “0”
	STPREQ	Set this bit to “1” to generate stop condition	Set to “0”
	STSPSEL	Set this bit to “1” to output each condition	Set to “0”
	ACKD	Select ACK or NACK	Select ACK or NACK
	ACKC	Set this bit to “1” to output ACK data	Set this bit to “1” to output ACK data
	SCLHI	Set this bit to “1” to have SCLi output stopped when stop condition is detected	Set to “0”
	SWC9	Set to “0”	Set this bit to “1” to set the SCLi to “L” hold at the falling edge of the 9th bit of clock
IFSR2A	IFSR26, ISFR27	Set to “1”	Set to “1”
UCON	U0IRS, U1IRS	Invalid	Invalid
	2 to 7	Set to “0”	Set to “0”

i=0 to 2

Table 1.20.4. I^2C Mode Functions

Function	Clock synchronous serial I/O mode (SMD2 to SMD0 = 001_2, IICM = 0)	I^2C mode (SMD2 to SMD0 = 010_2, IICM = 1)			
		IICM2 = 0 (NACK/ACK interrupt)		IICM2 = 1 (UART transmit/ receive interrupt)	
		CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)	CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)
Factor of interrupt number 6, 7 and 10 (Note 1)(Refer to Fig 1.20.2)	———	Start condition detection or stop condition detection (Refer to Fig 1.20.4)			
Factor of interrupt number 15, 17 and 19 (Note 1)(Refer to Fig 1.20.2)	UARTi transmission Transmission started or completed (selected by UiIRS)	No acknowledgment detection (NACK) Rising edge of SCLi 9th bit		UARTi transmission Rising edge of SCLi 9th bit	UARTi transmission Falling edge of SCLi next to the 9th bit
Factor of interrupt number 16, 18 and 20 (Note 1)(Refer to Fig 1.20.2)	UARTi reception When 8th bit received CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Acknowledgment detection (ACK) Rising edge of SCLi 9th bit		UARTi transmission Falling edge of SCLi 9th bit	
Timing for transferring data from the UART reception shift register to the UiRB register	CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Rising edge of SCLi 9th bit		Falling edge of SCLi 9th bit	Falling and rising edges of SCLi 9th bit
UARTi transmission output delay	Not delayed	Delayed			
Functions of $P6_3$, $P6_7$ and $P7_0$ pins	TxDi output	SDAi input/output			
Functions of $P6_2$, $P6_6$ and $P7_1$ pins	RxDi input	SCLi input/output			
Functions of $P6_1$, $P6_5$ and $P7_2$ pins	CLKi input or output selected	——— (Cannot be used in I^2C mode)			
Noise filter width	15ns	200ns			
Read RxDi and SCLi pin levels	Possible when the corresponding port direction bit = 0	Always possible no matter how the corresponding port direction bit is set			
Initial value of TxDi and SDAi outputs	CKPOL = 0 (H) CKPOL = 1 (L)	The value set in the port register before setting I^2C mode (Note 2)			
Initial and end values of SCLi	———	H	L	H	L
DMA1 factor (Refer to Fig 1.20.2)	UARTi reception	Acknowledgment detection (ACK)		UARTi reception Falling edge of SCLi 9th bit	
Store received data	1st to 8th bits are stored in UiRB register bit 0 to bit 7	1st to 8th bits are stored in UiRB register bit 7 to bit 0		1st to 7th bits are stored in UiRB register bit 6 to bit 0, with 8th bit stored in UiRB register bit 8	1st to 8th bits are stored in UiRB register bit 7 to bit 0 (Note 3)
Read received data	UiRB register status is read directly as is				Read UiRB register Bit 6 to bit 0 as bit 7 to bit 1, and bit 8 as bit 0 (Note 4)

i = 0 to 2

Note 1: To change the interrupt sources from one to another, follow the procedure described below.
1. Disable the interrupt of the corresponding interrupt number to be changed.
2. Change interrupt sources from one to another.
3. Set the IR bit for the corresponding interrupt number to 0 (no interrupt request).
4. Set the IPL2 to IPL0 bits for the corresponding interrupt number.

Note 2: Set the initial value of SDAi output while the UiMR register's SMD2 to SMD0 bits = '000_2' (serial I/O disabled).
Note 3: Second data transfer to UiRB register (Rising edge of SCLi 9th bit)
Note 4. First data transfer to UiRB register (Falling edge of SCLi 9th bit)

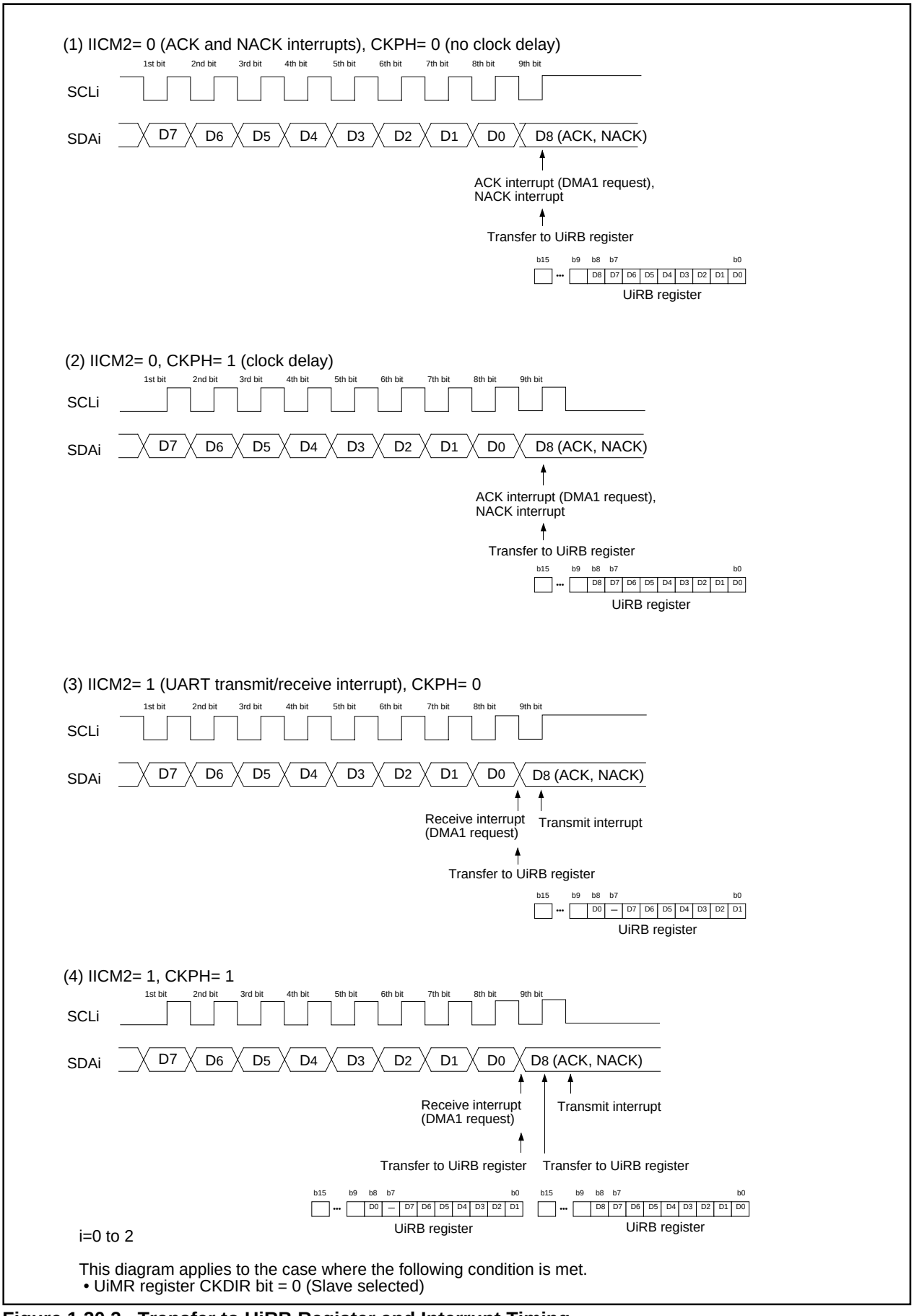


Figure 1.20.2. Transfer to UiRB Register and Interrupt Timing

• Detection of Start and Stop Condtion

Whether a start or a stop condition has been detected is determined.

A start condition-detected interrupt request is generated when the SDAi pin changes state from high to low while the SCLi pin is in the high state. A stop condition-detected interrupt request is generated when the SDAi pin changes state from low to high while the SCLi pin is in the high state.

Because the start and stop condition-detected interrupts share the interrupt control register and vector, check the UiSMR register's BBS bit to determine which interrupt source is requesting the interrupt.

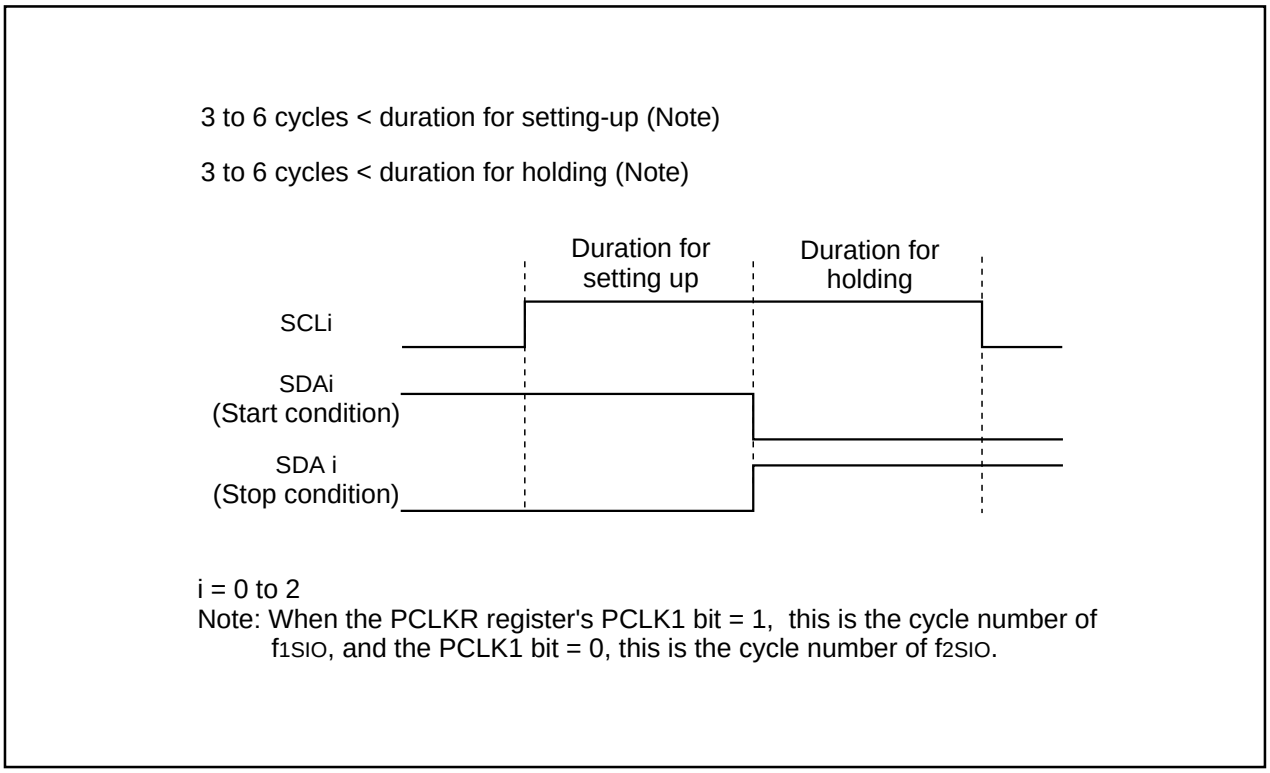


Figure 1.20.3. Detection of Start and Stop Condition

• Output of Start and Stop Condition

A start condition is generated by setting the UiSMR4 register (i = 0 to 2)'s STAREQ bit to "1" (start).

A restart condition is generated by setting the UiSMR4 register's RSTAREQ bit to "1" (start).

A stop condition is generated by setting the UiSMR4 register's STPREQ bit to "1" (start).

A start condition is output by setting the STAREQ bit to "1" and then the UiSMR4 register's STSPSEL bit to "1" (start). Similarly, a restart condition is output by setting the RSTAREQ bit to "1" and then the STSPSEL bit to "1", and a stop condition is output by setting the STPREQ bit to "1" and then the STSPSEL bit to "1".

Table 1.20.5 and Figure 1.20.4 show the functions of the STSPSEL.

If start, stop and restart conditions are to be output, make sure no interrupts will occur between the instruction that sets the STAREQ, STPREQ or RSTAREQ bit to "1" and the instruction that sets the STSPSEL bit to "1".

Also, if a start condition is to be output, make sure the STAREQ bit is set to "1" before setting the STSPSEL bit to "1".

Table 1.20.5. STSPSEL Bit Functions

Function	STSPSEL = 0	STSPSEL = 1
Output of SCLi and SDAi pins	Output of transfer clock and data Output of start/stop condition is accomplished by a program using ports (not automatically generated in hardware)	Output of a start/stop condition according to the STAREQ, RSTAREQ and STPREQ bit
Star/stop condition interrupt request generation timing	Start/stop condition detection	Finish generating start/stop condition

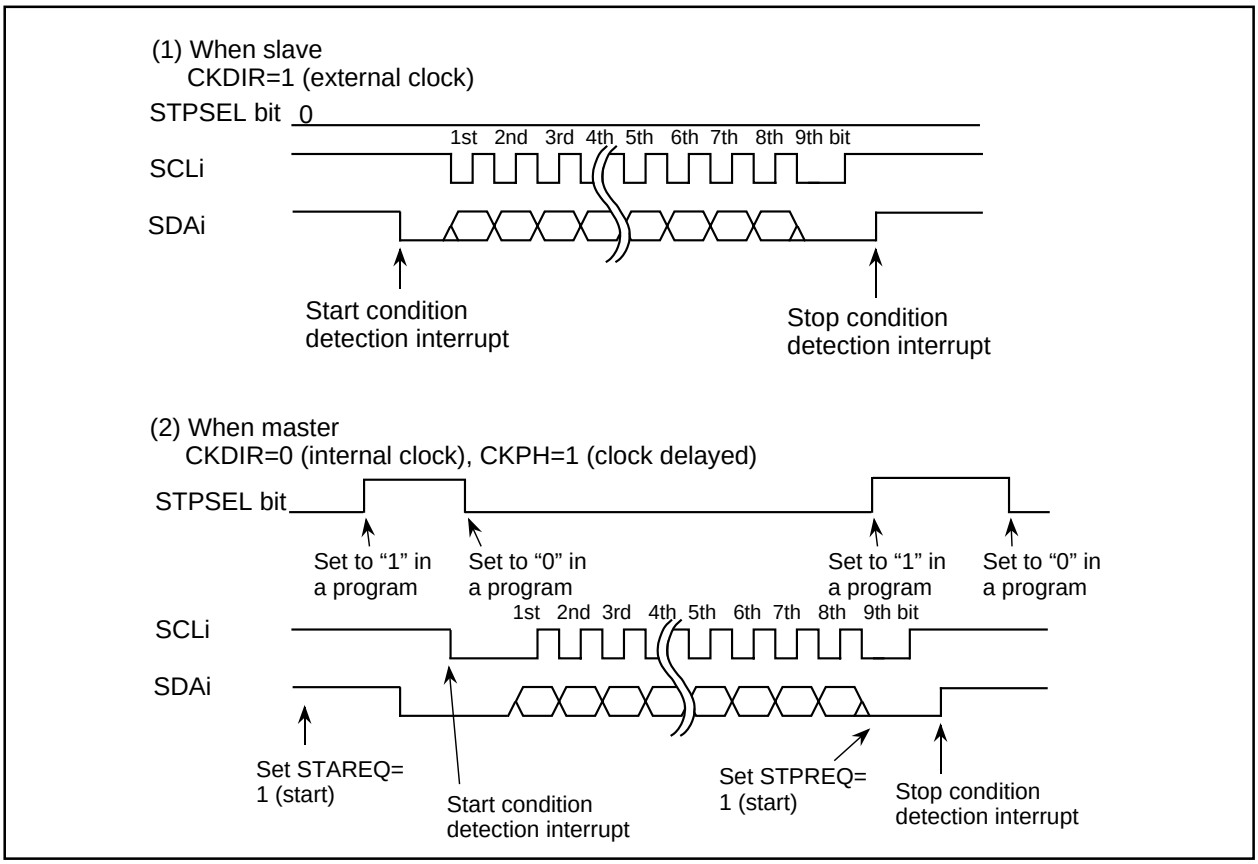


Figure 1.20.4. STSPSEL Bit Functions

- **Arbitration**

Unmatching of the transmit data and SDAi pin input data is checked synchronously with the rising edge of SCLi. Use the UiSMR register's ABC bit to select the timing at which the UiRB register's ABT bit is updated. If the ABC bit = 0 (updated bitwise), the ABT bit is set to "1" at the same time unmatching is detected during check, and is cleared to "0" when not detected. In cases when the ABC bit is set to "1", if unmatching is detected even once during check, the ABT bit is set to "1" (unmatching detected) at the falling edge of the clock pulse of 9th bit. If the ABT bit needs to be updated bytewise, clear the ABT bit to "0" (undetected) after detecting acknowledge in the first byte, before transferring the next byte.

Setting the UiSMR2 register's ALS bit to "1" (SDA output stop enabled) causes arbitration-lost to occur, in which case the SDAi pin is placed in the high-impedance state at the same time the ABT bit is set to "1" (unmatching detected).

• Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in Figure 1.20.4.

The UiSMR2 register's CSC bit is used to synchronize the internally generated clock (internal SCLi) and an external clock supplied to the SCLi pin. In cases when the CSC bit is set to "1" (clock synchronization enabled), if a falling edge on the SCLi pin is detected while the internal SCLi is high, the internal SCLi goes low, at which time the UiBRG register value is reloaded with and starts counting in the low-level interval. If the internal SCLi changes state from low to high while the SCLi pin is low, counting stops, and when the SCLi pin goes high, counting restarts.

In this way, the UARTi transfer clock is comprised of the logical product of the internal SCLi and SCLi pin signal. The transfer clock works from a half period before the falling edge of the internal SCLi 1st bit to the rising edge of the 9th bit. To use this function, select an internal clock for the transfer clock.

The UiSMR2 register's SWC bit allows to select whether the SCLi pin should be fixed to or freed from low-level output at the falling edge of the 9th clock pulse.

If the UiSMR4 register's SCLHI bit is set to "1" (enabled), SCLi output is turned off (placed in the high-impedance state) when a stop condition is detected.

Setting the UiSMR2 register's SWC2 bit = 1 (0 output) makes it possible to forcibly output a low-level signal from the SCLi pin even while sending or receiving data. Clearing the SWC2 bit to "0" (transfer clock) allows the transfer clock to be output from or supplied to the SCLi pin, instead of outputting a low-level signal.

If the UiSMR4 register's SWC9 bit is set to "1" (SCL hold low enabled) when the UiSMR3 register's CKPH bit = 1, the SCLi pin is fixed to low-level output at the falling edge of the clock pulse next to the ninth. Setting the SWC9 bit = 0 (SCL hold low disabled) frees the SCLi pin from low-level output.

• SDA Output

The data written to the UiTB register bit 7 to bit 0 (D_7 to D_0) is sequentially output beginning with D_7. The ninth bit (D_8) is ACK or NACK.

The initial value of SDAi transmit output can only be set when IICM = 1 (I^2C mode) and the UiMR register's SMD2 to SMD0 bits = '000_2' (serial I/O disabled).

The UiSMR3 register's DL2 to DL0 bits allow to add no delays or a delay of 2 to 8 UiBRG count source clock cycles to SDAi output.

Setting the UiSMR2 register's SDHI bit = 1 (SDA output disabled) forcibly places the SDAi pin in the high-impedance state. Do not write to the SDHI bit synchronously with the rising edge of the UARTi transfer clock. This is because the ABT bit may inadvertently be set to "1" (detected).

• SDA Input

When the IICM2 bit = 0, the 1st to 8th bits (D_7 to D_0) of received data are stored in the UiRB register bit 7 to bit 0. The 9th bit (D_8) is ACK or NACK.

When the IICM2 bit = 1, the 1st to 7th bits (D_7 to D_1) of received data are stored in the UiRB register bit 6 to bit 0 and the 8th bit (D_0) is stored in the UiRB register bit 8. Even when the IICM2 bit = 1, providing the CKPH bit = 1, the same data as when the IICM2 bit = 0 can be read out by reading the UiRB register after the rising edge of the corresponding clock pulse of 9th bit.

• ACK and NACK

If the STSPSEL bit in the UiSMR4 register is set to “0” (start and stop conditions not generated) and the ACKC bit in the UiSMR4 register is se to “1” (ACK data output), the value of the ACKD bit in the UiSMR4 register is output from the SDAi pin.

If the IICM2 bit = 0, a NACK interrupt request is generated if the SDAi pin remains high at the rising edge of the 9th bit of transmit clock pulse. An ACK interrupt request is generated if the SDAi pin is low at the rising edge of the 9th bit of transmit clock pulse.

If ACKi is selected for the cause of DMA1 request, a DMA transfer can be activated by detection of an acknowledge.

• Initialization of Transmission/Reception

If a start condition is detected while the STAC bit = 1 (UARTi initialization enabled), the serial I/O operates as described below.

- The transmit shift register is initialized, and the content of the UiTB register is transferred to the transmit shift register. In this way, the serial I/O starts sending data synchronously with the next clock pulse applied. However, the UARTi output value does not change state and remains the same as when a start condition was detected until the first bit of data is output synchronously with the input clock.
- The receive shift register is initialized, and the serial I/O starts receiving data synchronously with the next clock pulse applied.
- The SWC bit is set to “1” (SCL wait output enabled). Consequently, the SCLi pin is pulled low at the falling edge of the ninth clock pulse.

Note that when UARTi transmission/reception is started using this function, the TI does not change state. Note also that when using this function, the selected transfer clock should be an external clock.

Special Mode 2

Multiple slaves can be serially communicated from one master. Synchronous clock polarity and phase are selectable. Table 1.20.6 lists the specifications of Special Mode 2. Table 1.20.7 lists the registers used in Special Mode 2 and the register values set. Figure 1.20.5 shows communication control example for Special Mode 2.

Table 1.20.6. Special Mode 2 Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• Master mode UiMR(i=0 to 2) register's CKDIR bit = "0" (internal clock) : fj/ 2(n+1) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of UiBRG register 00_{16} to FF_{16} • Slave mode CKDIR bit = "1" (external clock selected) : Input from CLKi pin
Transmit/receive control	Controlled by input/output ports
Transmission start condition	• Before transmission can start, the following requirements must be met (Note 1) – The TE bit of UiC1 register= 1 (transmission enabled) – The TI bit of UiC1 register = 0 (data present in UiTB register)
Reception start condition	• Before reception can start, the following requirements must be met (Note 1) – The RE bit of UiC1 register= 1 (reception enabled) – The TE bit of UiC1 register= 1 (transmission enabled) – The TI bit of UiC1 register= 0 (data present in the UiTB register)
Interrupt request generation timing	• For transmission, one of the following conditions can be selected – The UiIRS bit of UiC1 register = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) – The UiIRS bit =1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register • For reception When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	• Overrun error (Note 2) This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 7th bit of the next data
Select function	• Clock phase setting Selectable from four combinations of transfer clock polarities and phases

Note 1: When an external clock is selected, the conditions must be met while if the UiC0 register's CKPOL bit = "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the UiC0 register's CKPOL bit = "1" (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.

Note 2: If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit of SiRIC register does not change.

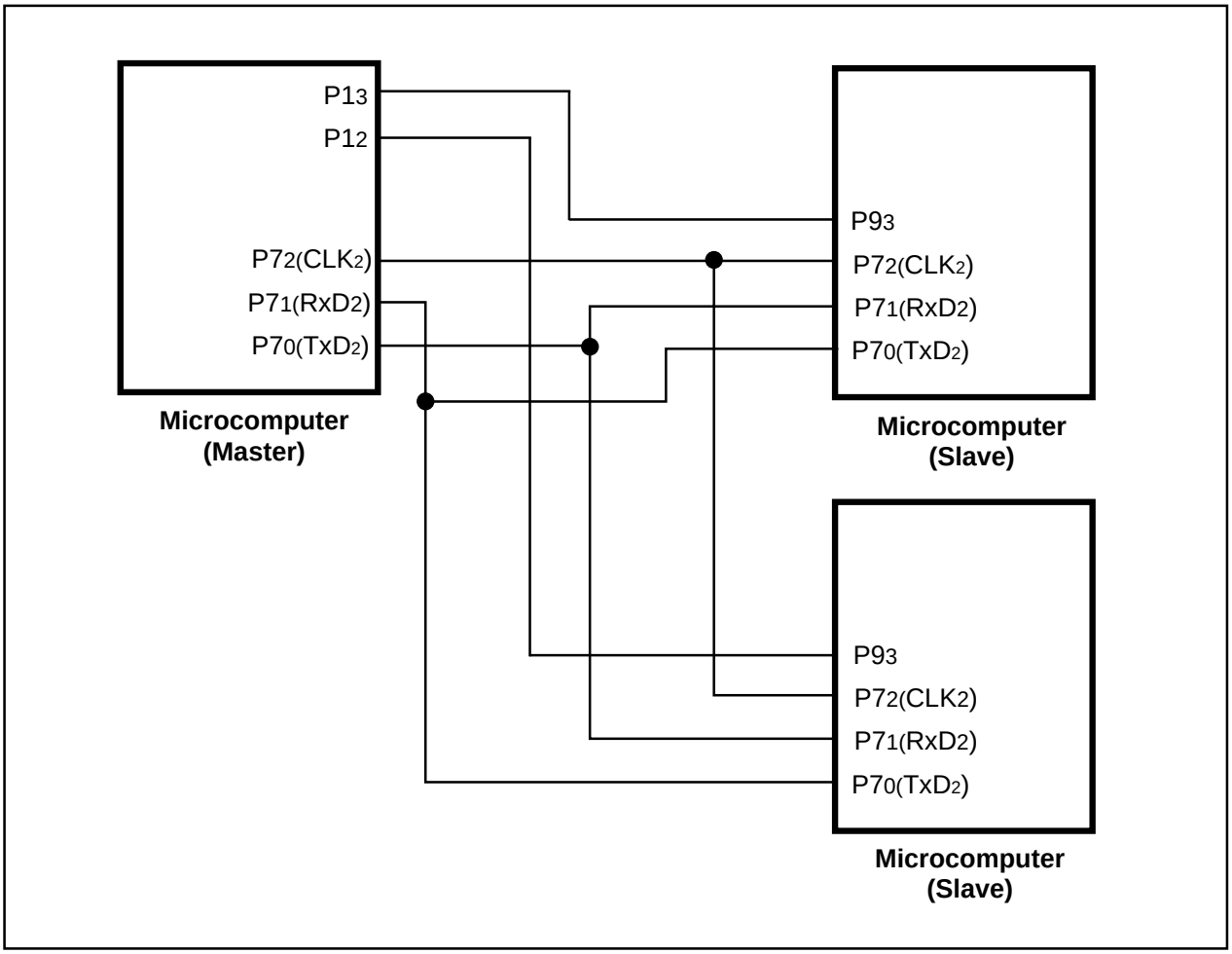


Figure 1.20.5. Serial Bus Communication Control Example (UART2)

Table 1. 20. 7. Registers to Be Used and Settings in Special Mode 2

Register	Bit	Function
UiTB(Note3)	0 to 7	Set transmission data
UiRB(Note3)	0 to 7	Reception data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a transfer rate
UiMR(Note3)	SMD2 to SMD0	Set to '001$_2$'
	CKDIR	Set this bit to "0" for master mode or "1" for slave mode
	IOPOL	Set to "0"
UiC0	CLK1, CLK0	Select the count source for the UiBRG register
	CRS	Invalid because CRD = 1
	TXEPT	Transmit register empty flag
	CRD	Set to "1"
	NCH	Select TxDi pin output format(Note 2)
	CKPOL	Clock phases can be set in combination with the UiSMR3 register's CKPH bit
	UFORM	Set to "0"
UiC1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS (Note 1)	Select UART2 transmit interrupt cause
	U2RRM(Note 1), U2LCH, UiERE	Set to "0"
UiSMR	0 to 7	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	CKPH	Clock phases can be set in combination with the UiC0 register's CKPOL bit
	NODC	Set to "0"
	0, 2, 4 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select UART0 and UART1 transmit interrupt cause
	U0RRM, U1RRM	Set to "0"
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1, RCSP, 7	Set to "0"

Note 1: Set the U0C0 and U1C1 register bit 4 and bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.

Note 2: TxD2 pin is N channel open-drain output. Set the U2C0 register's NCH bit to "0".

Note 3: Not all register bits are described above. Set those bits to "0" when writing to the registers in Special Mode 2.

i = 0 to 2

• **Clock Phase Setting Function**

One of four combinations of transfer clock phases and polarities can be selected using the UiSMR3 register's CKPH bit and the UiC0 register's CKPOL bit.

Make sure the transfer clock polarity and phase are the same for the master and salves to be communicated.

(a) Master (Internal Clock)

Figure 1.20.6 shows the transmission and reception timing in master (internal clock).

(b) Slave (External Clock)

Figure 1.20.7 shows the transmission and reception timing (CKPH=0) in slave (external clock) while Figure 1.20.8 shows the transmission and reception timing (CKPH=1) in slave (external clock).

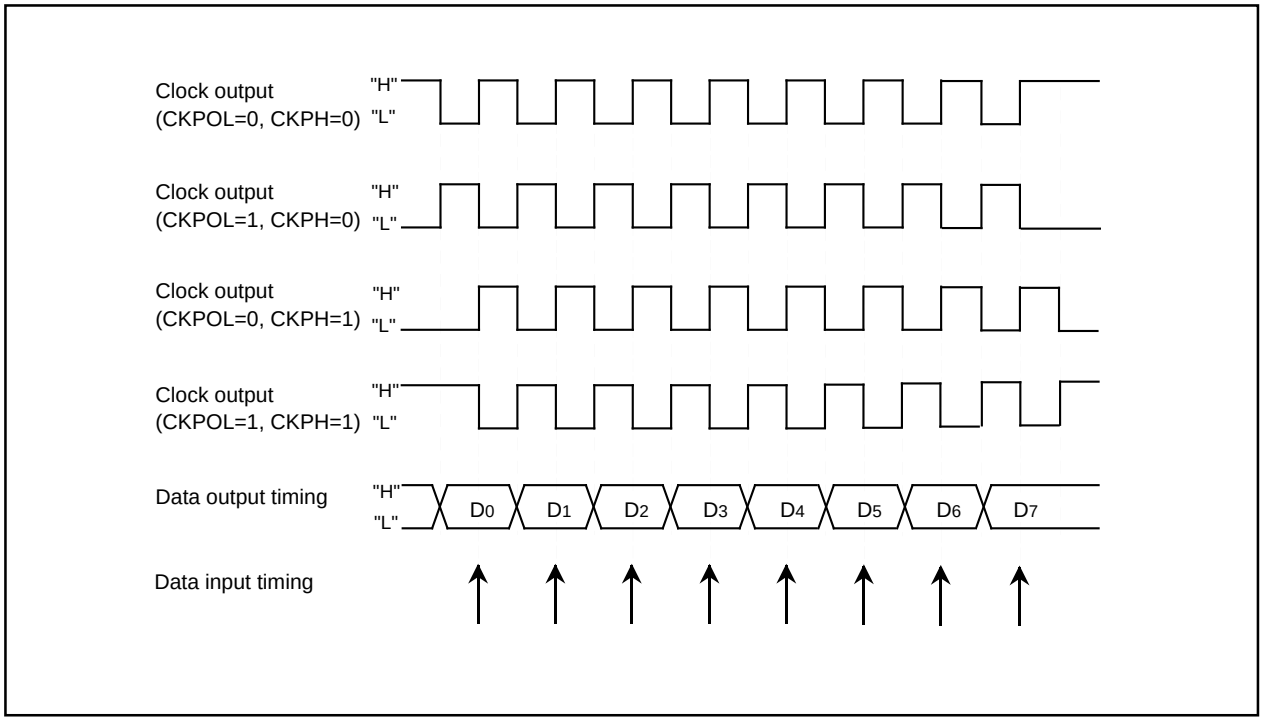


Figure 1.20.6. Transmission and Reception Timing in Master Mode (Internal Clock)

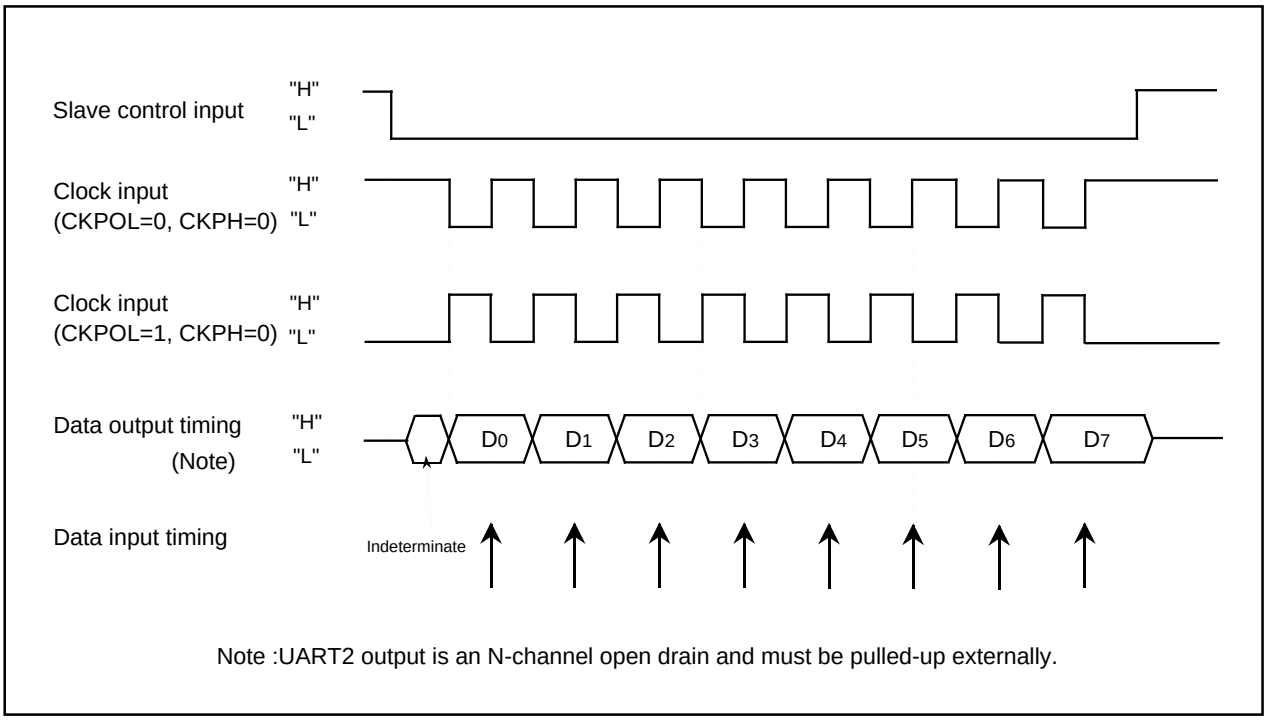


Figure 1.20.7. Transmission and Reception Timing (CKPH=0) in Slave Mode (External Clock)

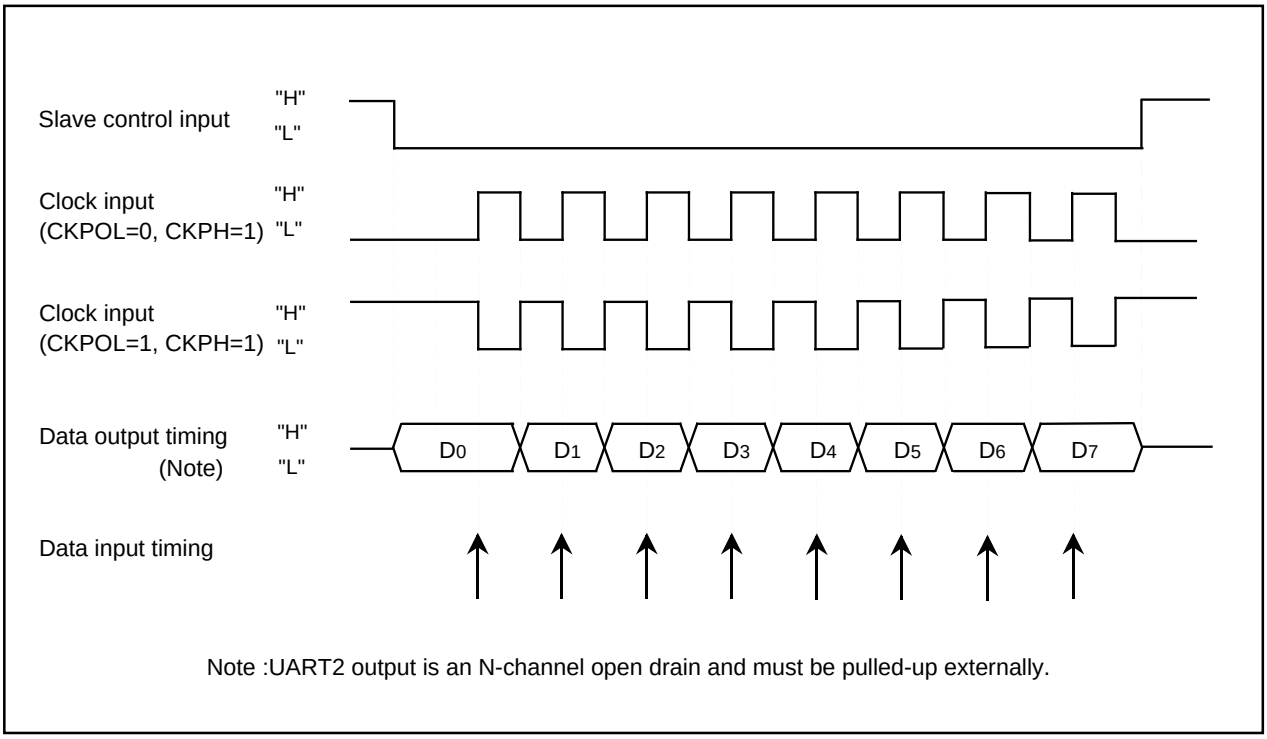


Figure 1.20.8. Transmission and Reception Timing (CKPH=1) in Slave Mode (External Clock)

Special Mode 3 (IE mode)

In this mode, one bit of IEBus is approximated with one byte of UART mode waveform.

Table 1.20.8 lists the registers used in IE mode and the register values set. Figure 1.20.9 shows the functions of bus collision detect function related bits.

If the TxDi pin (i = 0 to 2) output level and RxDi pin input level do not match, a UARTi bus collision detect interrupt request is generated.

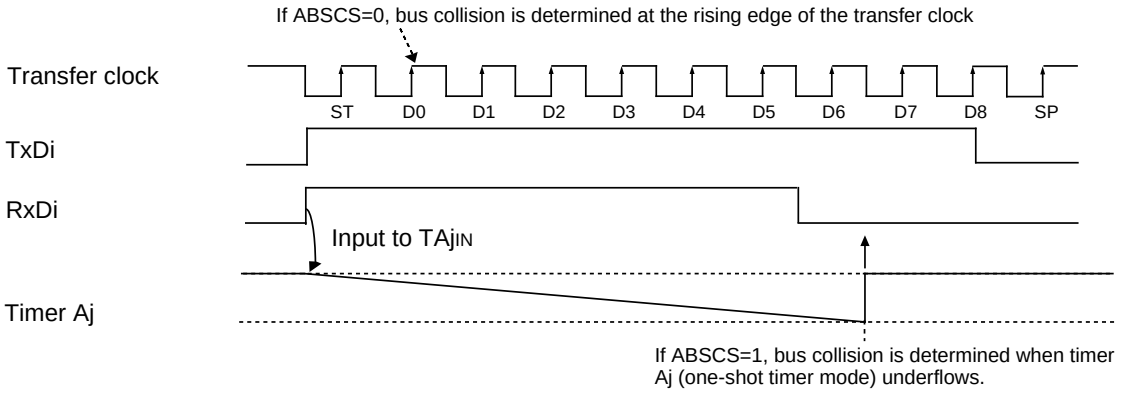
Use the IFSR2A register's IFSR26 and IFSR27 bits to enable the UART0/UART1 bus collision detect function.

Table 1. 20. 8. Registers to Be Used and Settings in IE Mode

Register	Bit	Function
UiTB	0 to 8	Set transmission data
UiRB(Note3)	0 to 8	Reception data can be read
	OER,FER,PER,SUM	Error flag
UiBRG	---	Set a transfer rate
UiMR	SMD2 to SMD0	Set to '110$_2$'
	CKDIR	Select the internal clock or external clock
	STPS	Set to "0"
	PRY	Invalid because PRYE=0
	PRYE	Set to "0"
	IOPOL	Select the TxD/RxD input/output polarity
UiC0	CLK1, CLK0	Select the count source for the UiBRG register
	CRS	Invalid because CRD=1
	TXEPT	Transmit register empty flag
	CRD	Set to "1"
	NCH	Select TxDi pin output mode (Note 2)
	CKPOL	Set to "0"
	UFORM	Set to "0"
UiC1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS (Note 1)	Select the source of UART2 transmit interrupt
	UiRRM (Note 1), UiLCH, UiERE	Set to "0"
UiSMR	0 to 3, 7	Set to "0"
	ABSCS	Select the sampling timing at which to detect a bus collision
	ACSE	Set this bit to "1" to use the auto clear function of transmit enable bit
	SSS	Select the transmit start condition
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
IFSR2A	IFSR26, IFSR27	Set to "1"
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set to "0"
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1,RCSP,7	Set to "0"

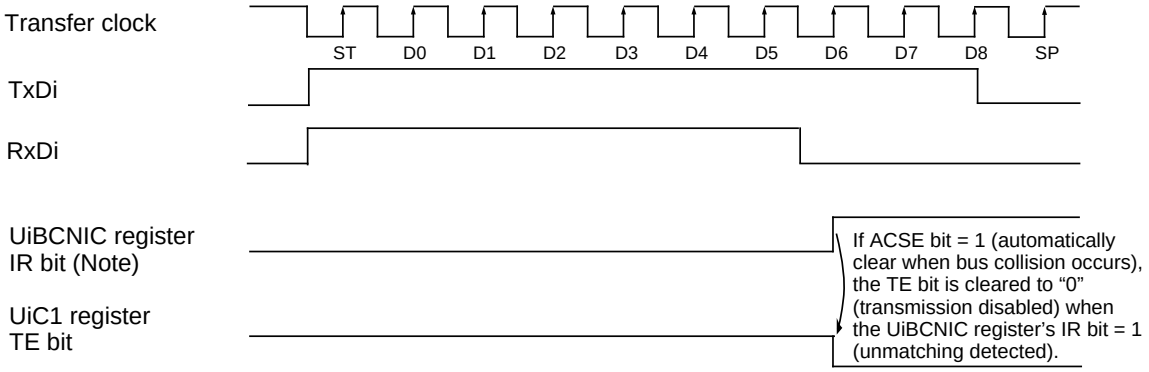
Note 1: Set the U0C0 and U1C1 registers bit 4 and bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.
Note 2: TxD2 pin is N channel open-drain output. Set the U2C0 register's NCH bit to "0".
Note 3: Not all register bits are described above. Set those bits to "0" when writing to the registers in IEmode. i= 0 to 2

(1) UiSMR register ABSCS bit (bus collision detect sampling clock select) (i=0 to 2)



Timer Aj: timer A3 when UART0; timer A4 when UART1; timer A0 when UART2

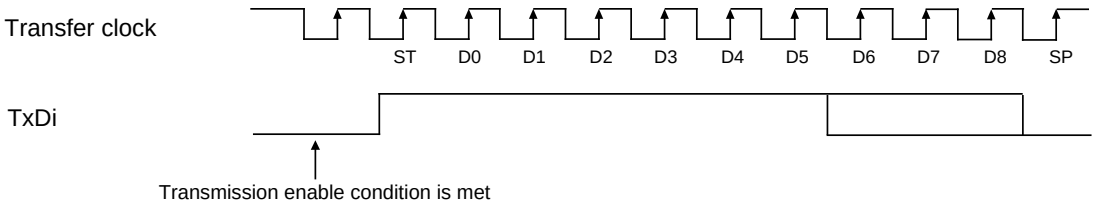
(2) UiSMR register ACSE bit (auto clear of transmit enable bit)



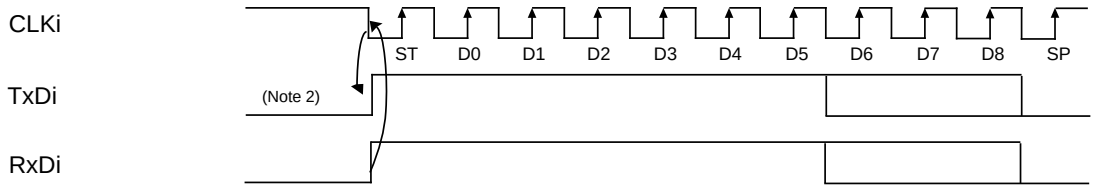
Note: BCNIC register when UART2.

(3) UiSMR register SSS bit (Transmit start condition select)

If SSS bit = 0, the serial I/O starts sending data one transfer clock cycle after the transmission enable condition is met.



If SSS bit = 1, the serial I/O starts sending data at the rising edge (Note 1) of RxDi



Note 1: The falling edge of RxDi when IOPOL=0; the rising edge of RxDi when IOPOL =1.
Note 2: The transmit condition must be met before the falling edge (Note 1) of RxD.

This diagram applies to the case where IOPOL=1 (reversed).

Figure 1.20.9. Bus Collision Detect Function-Related Bits

Special Mode 4 (SIM Mode) (UART2)

Based on UART mode, this is an SIM interface compatible mode. Direct and inverse formats can be implemented, and this mode allows to output a low from the TxD2 pin when a parity error is detected. Tables 1.20.9 lists the specifications of SIM mode. Table 1.20.10 lists the registers used in the SIM mode and the register values set.

Table 1.20.9. SIM Mode Specifications

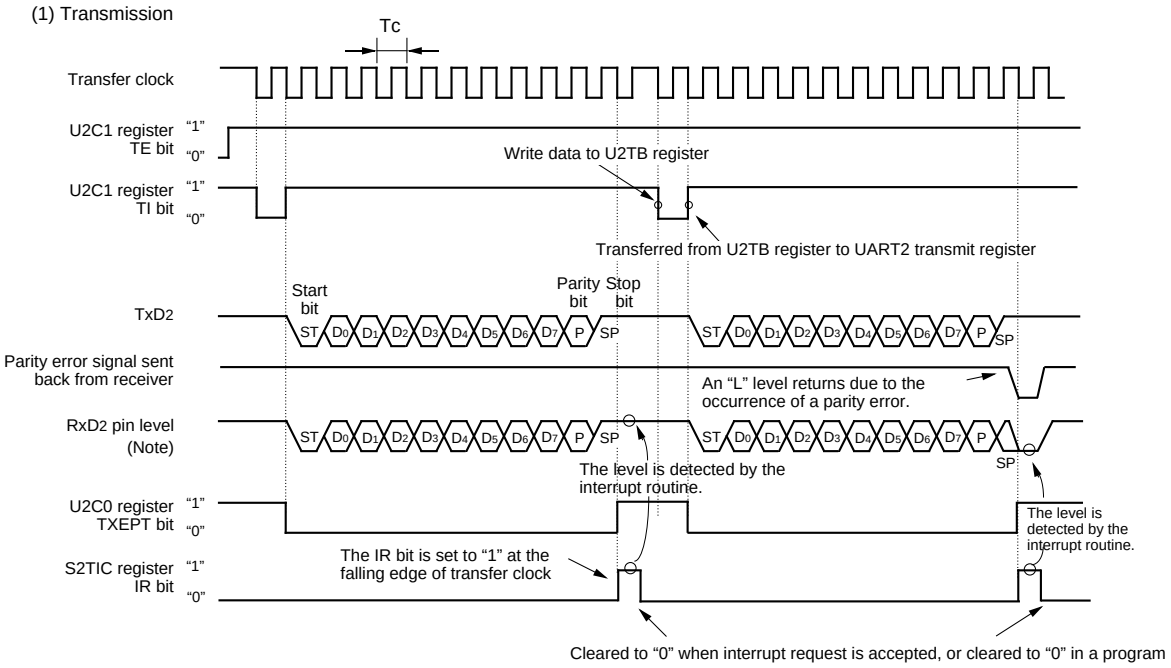
Item	Specification
Transfer data format	• Direct format • Inverse format
Transfer clock	• U2MR register's CKDIR bit = "0" (internal clock) : fi/ 16(n+1) fi = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of U2BRG register 00_{16} to FF_{16} • CKDIR bit = "1" (external clock) : f_{EXT}/16(n+1) f_{EXT}: Input from CLK_2 pin. n: Setting value of U2BRG register 00_{16} to FF_{16}
Transmission start condition	• Before transmission can start, the following requirements must be met – The TE bit of U2C1 register= 1 (transmission enabled) – The TI bit of U2C1 register = 0 (data present in U2TB register)
Reception start condition	• Before reception can start, the following requirements must be met – The RE bit of U2C1 register= 1 (reception enabled) – Start bit detection
Interrupt request generation timing	• For transmission When the serial I/O finished sending data from the U2TB transfer register (U2IRS bit =1) • For reception When transferring data from the UART2 receive register to the U2RB register (at completion of reception)
Error detection	• Overrun error (Note) This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the bit one before the last stop bit of the next data • Framing error This error occurs when the number of stop bits set is not detected • Parity error During reception, if a parity error is detected, parity error signal is output from the TxD_2 pin. During transmission, a parity error is detected by the level of input to the RxD_2 pin when a transmission interrupt occurs • Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered

Note: If an overrun error occurs, the value of U2RB register will be indeterminate. The IR bit of S2RIC register does not change.

Table 1. 20. 10. Registers to Be Used and Settings in SIM Mode

Register	Bit	Function
U2TB(Note)	0 to 7	Set transmission data
U2RB(Note)	0 to 7	Reception data can be read
	OER,FER,PER,SUM	Error flag
U2BRG	---	Set a transfer rate
U2MR	SMD2 to SMD0	Set to '101_2'
	CKDIR	Select the internal clock or external clock
	STPS	Set to "0"
	PRY	Set this bit to "1" for direct format or "0" for inverse format
	PRYE	Set to "1"
	IOPOL	Set to "0"
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because CRD=1
	TXEPT	Transmit register empty flag
	CRD	Set to "1"
	NCH	Set to "0"
	CKPOL	Set to "0"
	UFORM	Set this bit to "0" for direct format or "1" for inverse format
U2C1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS	Set to "1"
	U2RRM	Set to "0"
	U2LCH	Set this bit to "0" for direct format or "1" for inverse format
	U2ERE	Set to "1"
U2SMR(Note)	0 to 3	Set to "0"
U2SMR2	0 to 7	Set to "0"
U2SMR3	0 to 7	Set to "0"
U2SMR4	0 to 7	Set to "0"

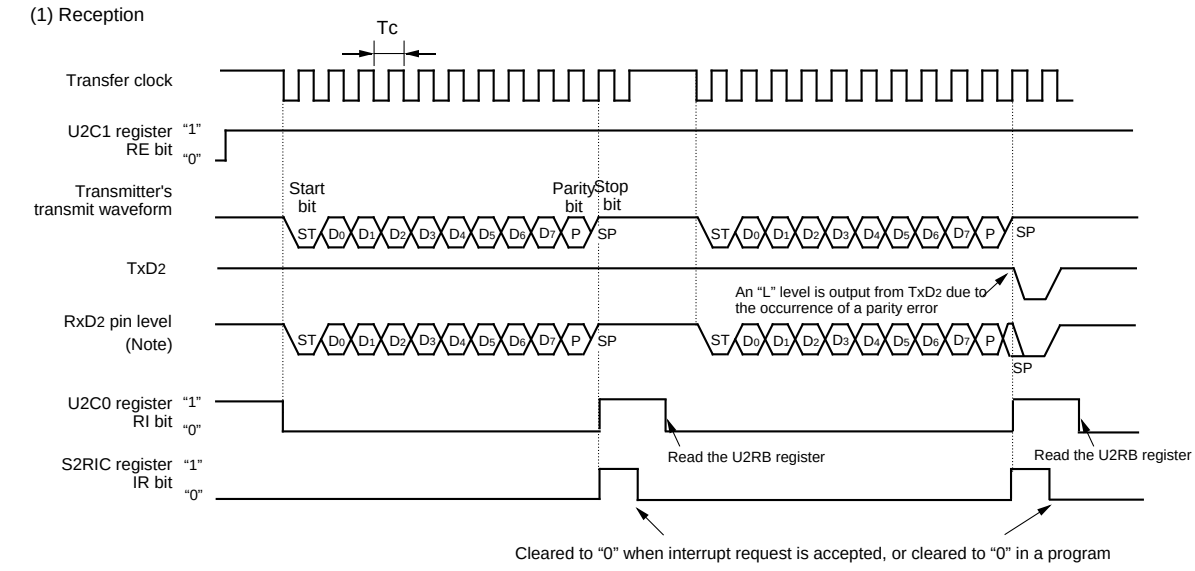
Note: Not all register bits are described above. Set those bits to "0" when writing to the registers in SIM mode.



The above timing diagram applies to the case where data is transferred in the direct format.
- U2MR register PRY bit = 1 (even)
- U2C0 register UFORM bit = 0 (LSB first)
- U2C1 register U2LCH bit = 0 (no reverse)

Tc = 16 (n + 1) / fi or 16 (n + 1) / f_{EXT}
fi : frequency of U2BRG count source (f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO})
f_{EXT} : frequency of U2BRG count source (external clock)
n : value set to U2BRG

Note : Because TxD2 and RxD2 are connected, this is composite waveform consisting of the TxD2 output and the parity error signal sent back from receiver.



The above timing diagram applies to the case where data is received in direct format.
- U2MR register PRY bit = 1 (even)
- U2C0 register UFORM bit = 0 (LSB first)
- U2C1 register U2LCH bit = 0 (no reverse)

Tc = 16 (n + 1) / fi or 16 (n + 1) / f_{EXT}
fi : frequency of U2BRG count source (f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO})
f_{EXT} : frequency of U2BRG count source (external clock)
n : value set to U2BRG

Note : Because TxD2 and RxD2 are connected, this is composite waveform consisting of the transmitter's transmit waveform and the parity error signal received.

Figure 1.20.10. Transmit and Receive Timing in SIM Mode

Figure 1.20.11 shows the example of connecting the SIM interface. Connect TxD2 and RxD2 and apply pull-up.

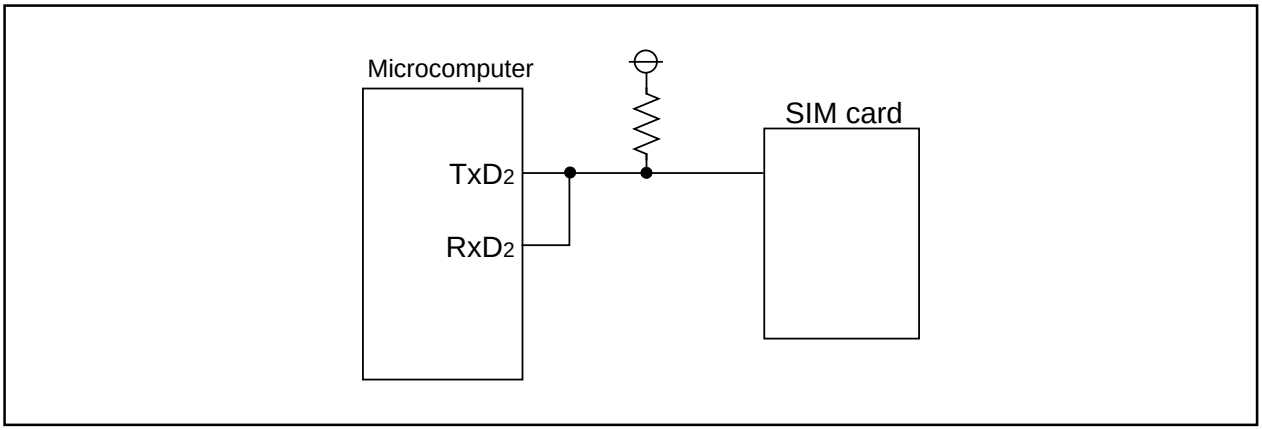


Figure 1.20.11. SIM Interface Connection

(a) Parity Error Signal Output

The parity error signal is enabled by setting the U2C1 register's U2ERE bit to "1".

• When receiving

The parity error signal is output when a parity error is detected while receiving data. This is achieved by pulling the TxD2 output low with the timing shown in Figure 1.20.12. If the R2RB register is read while outputting a parity error signal, the PER bit is cleared to "0" and at the same time the TxD2 output is returned high.

• When transmitting

A transmission-finished interrupt request is generated at the falling edge of the transfer clock pulse that immediately follows the stop bit. Therefore, whether a parity signal has been returned can be determined by reading the port that shares the RxD2 pin in a transmission-finished interrupt service routine.

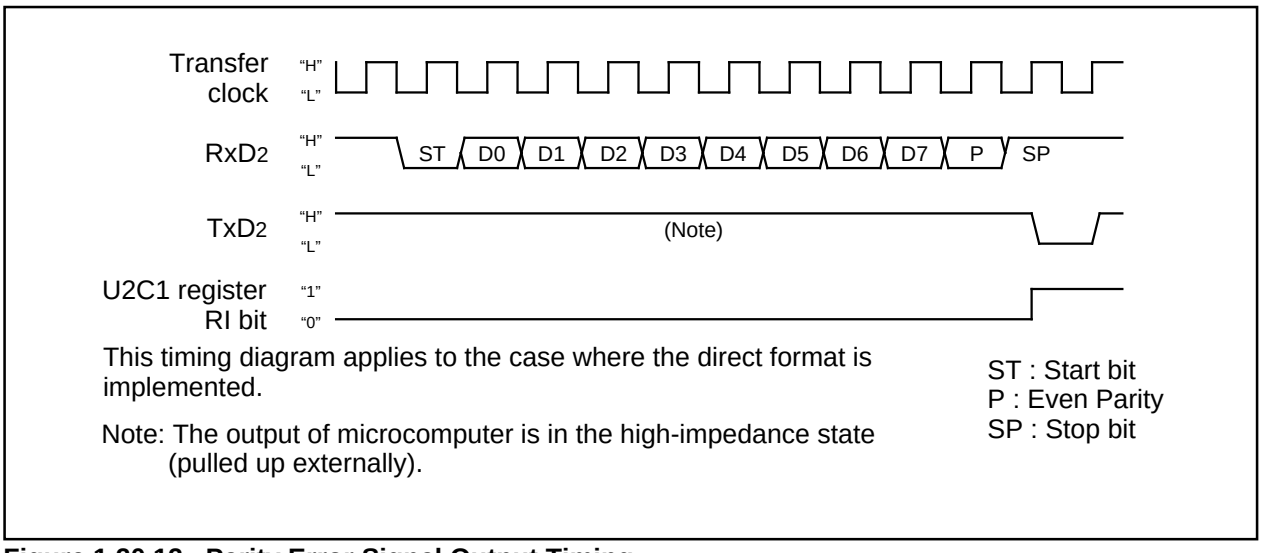


Figure 1.20.12. Parity Error Signal Output Timing

(b) Format

• Direct Format

Set the U2MR register's PRY bit to “1”, U2C0 register's UFORM bit to “0” and U2C1 register's U2LCH bit to “0”.

• Inverse Format

Set the PRY bit to “0”, UFORM bit to “1” and U2LCH bit to “1”.

Figure 1.20.13 shows the SIM interface format.

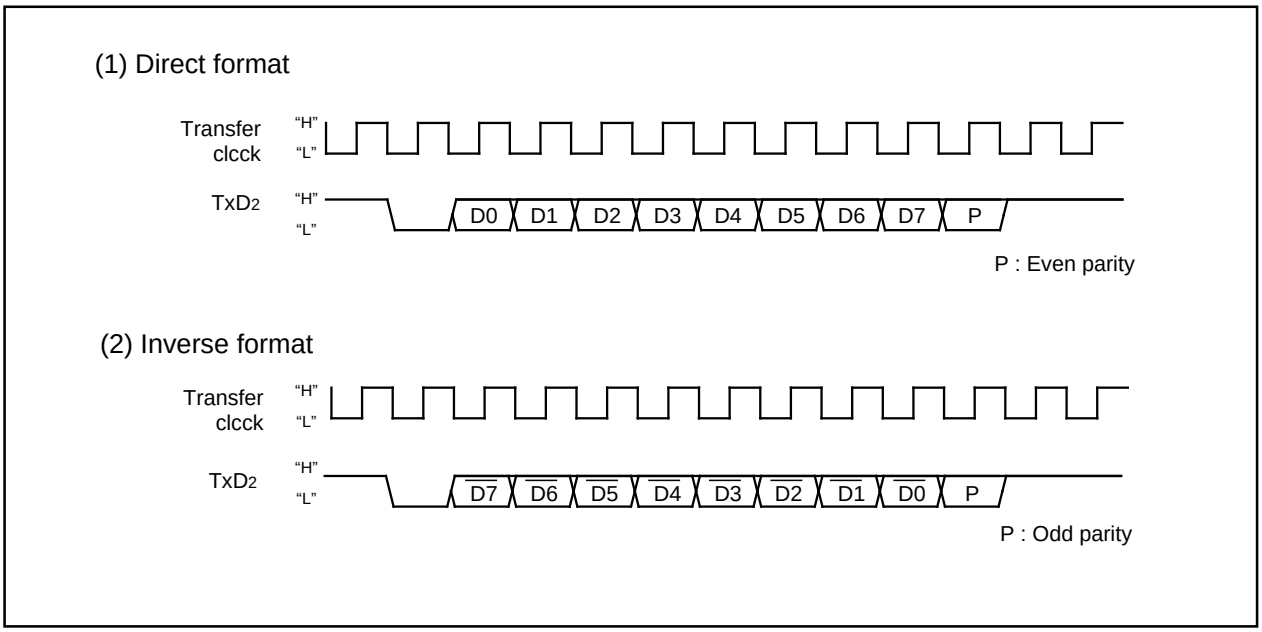


Figure 1.20.13. SIM Interface Format

SI/O3 and SI/O4

SI/O3 and SI/O4 are exclusive clock-synchronous serial I/Os.

Figure 1.21.1 shows the block diagram of SI/O3 and SI/O4, and Figure 1.21.2 shows the SI/O3 and SI/O4-related registers.

Table 1.21.1 shows the specifications of SI/O3 and SI/O4.

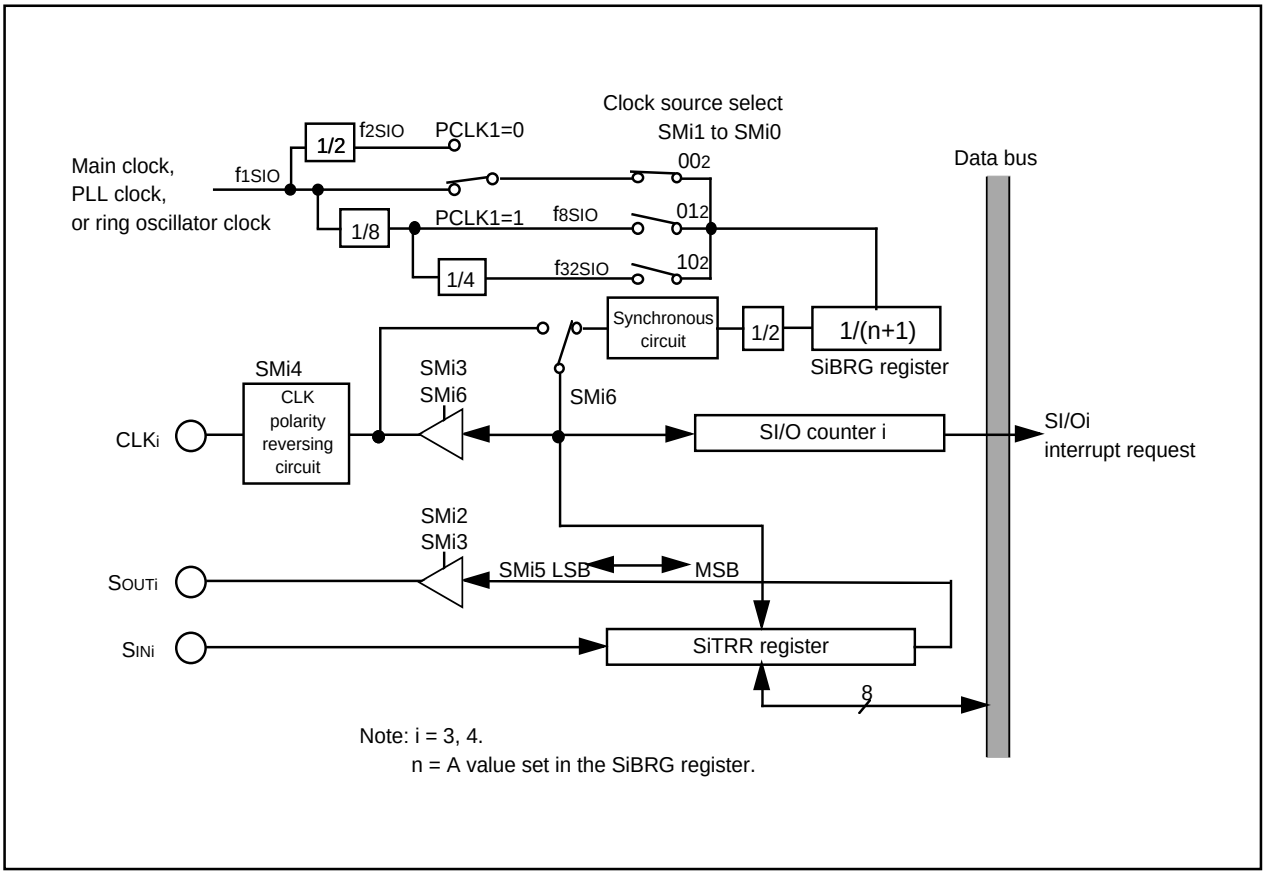
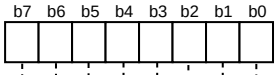


Figure 1.21.1. SI/O3 and SI/O4 Block Diagram

S I/Oi control register (i = 3, 4) (Note 1)

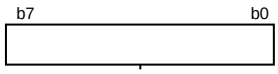


Symbol	Address	After reset
S3C	0362₁₆	01000000₁₆
S4C	0366₁₆	01000000₁₆

Bit symbol	Bit name	Description	RW
SMi0	Internal synchronous clock select bit	b1 b0 0 0 : Selecting f_{1SIO} or f_{2SIO} 0 1 : Selecting f_{8SIO} 1 0 : Selecting f_{32SIO} 1 1 : Must not be set.	RW
SMi1			RW
SMi2	SOUTi output disable bit (Note 4)	0 : SOUTi output 1 : SOUTi output disable(high impedance)	RW
SMi3	S I/Oi port select bit	0 : Input/output port 1 : SOUTi output, CLKi function	RW
SMi4	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	RW
SMi5	Transfer direction select bit	0 : LSB first 1 : MSB first	RW
SMi6	Synchronous clock select bit	0 : External clock (Note 2) 1 : Internal clock (Note 3)	RW
SMi7	SOUTi initial value set bit	Effective when SMi3 = 0 0 : "L" output 1 : "H" output	RW

Note 1: Make sure this register is written to by the next instruction after setting the PRCR register's PRC2 bit to "1" (write enable).
Note 2: Set the SMi3 bit to "1" (SOUTi output, CLKi function).
Note 3: Set the SMi3 bit to "1" and the corresponding port direction bit to "0" (input mode).
Note 4: Effective when SMi3 bit = 1.

SI/Oi bit rate generator (i = 3, 4) (Notes 1, 2)

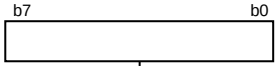


Symbol	Address	After reset
S3BRG	0363₁₆	Indeterminate
S4BRG	0367₁₆	Indeterminate

Description	Setting range	RW
Assuming that set value = n, BRGi divides the count source by n + 1	00₁₆ to FF₁₆	WO

Note 1: Write to this register while serial I/O is neither transmitting nor receiving.
Note 2: Use MOV instruction to write to this register.

SI/Oi transmit/receive register (i = 3, 4) (Note 1, 2)



Symbol	Address	After reset
S3TRR	0360₁₆	Indeterminate
S4TRR	0364₁₆	Indeterminate

Description	RW
Transmission/reception starts by writing transmit data to this register. After transmission/reception finishes, reception data can be read by reading this register.	RW

Note 1: Write to this register while serial I/O is neither transmitting nor receiving.
Note 2: To receive data, set the corresponding port direction bit for SINi to "0" (input mode).

Figure 1.21.2. S3C and S4C Registers, S3BRG and S4BRG Registers, and S3TRR and S4TRR Registers

Table 1.21.1. SI/O3 and SI/O4 Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• SiC (i=3, 4) register's SMi6 bit = "1" (internal clock) : fj/ 2(n+1) fj = f_{1SIO}, f_{8SIO}, f_{32SIO}. n=Setting value of SiBRG register 00_{16} to FF_{16}. • SMi6 bit = "0" (external clock) : Input from CLKi pin (Note 1)
Transmission/reception start condition	• Before transmission/reception can start, the following requirements must be met Write transmit data to the SiTRR register (Notes 2, 3)
Interrupt request generation timing	• When SiC register's SMi4 bit = 0 The rising edge of the last transfer clock pulse (Note 4) • When SMi4 = 1 The falling edge of the last transfer clock pulse (Note 4)
CLKi pin fucntion	I/O port, transfer clock input, transfer clock output
SOUTi pin function	I/O port, transmit data output, high-impedance
SINi pin function	I/O port, receive data input
Select function	• LSB first or MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Function for setting an SOUTi initial value set function When the SiC register's SMi6 bit = 0 (external clock), the SOUTi pin output level while not tranmitting can be selected. • CLK polarity selection Whether transmit data is output/input timing at the rising edge or falling edge of transfer clock can be selected.

Note 1: To set the SiC register's SMi6 bit to "0" (external clock), follow the procedure described below.
• If the SiC register's SMi4 bit = 0, write transmit data to the SiTRR register while input on the CLKi pin is high. The same applies when rewriting the SiC register's SMi7 bit.
• If the SMi4 bit = 1, write transmit data to the SiTRR register while input on the CLKi pin is low. The same applies when rewriting the SMi7 bit.
• Because shift operation continues as long as the transfer clock is supplied to the SI/Oi circuit, stop the transfer clock after supplying eight pulses. If the SMi6 bit = 1 (internal clock), the transfer clock automatically stops.

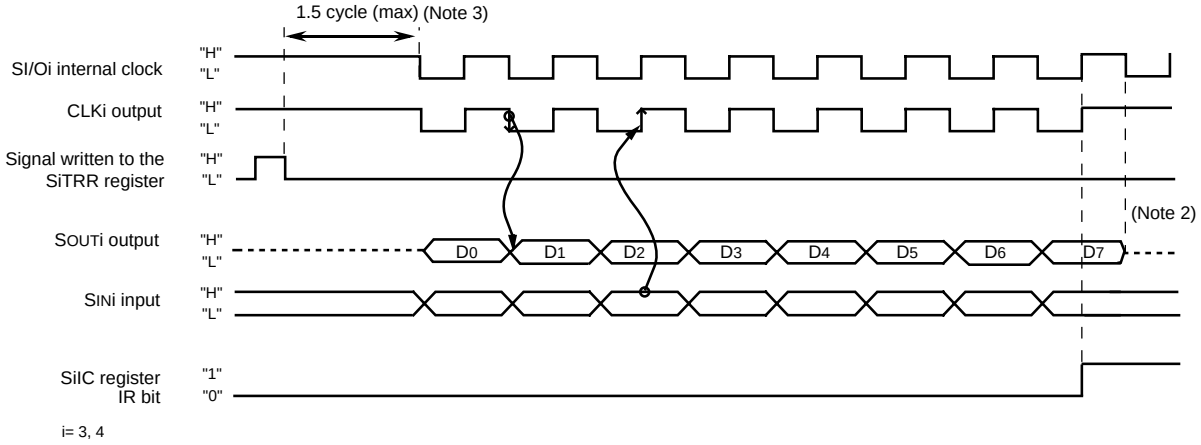
Note 2: Unlike UART0 to UART2, SI/Oi (i = 3 to 4) is not separated between the transfer register and buffer. Therefore, do not write the next transmit data to the SiTRR register during transmission.

Note 3: When the SiC register's SMi6 bit = 1 (internal clock), SOUTi retains the last data for a 1/2 transfer clock period after completion of transfer and, thereafter, goes to a high-impedance state. However, if transmit data is written to the SiTRR register during this period, SOUTi immediately goes to a high-impedance state, with the data hold time thereby reduced.

Note 4: When the SiC register's SMi6 bit = 1 (internal clock), the transfer clock stops in the high state if the SMi4 bit = 0, or stops in the low state if the SMi4 bit = 1.

(a) SI/Oi Operation Timing

Figure 1.21.3 shows the SI/Oi operation timing

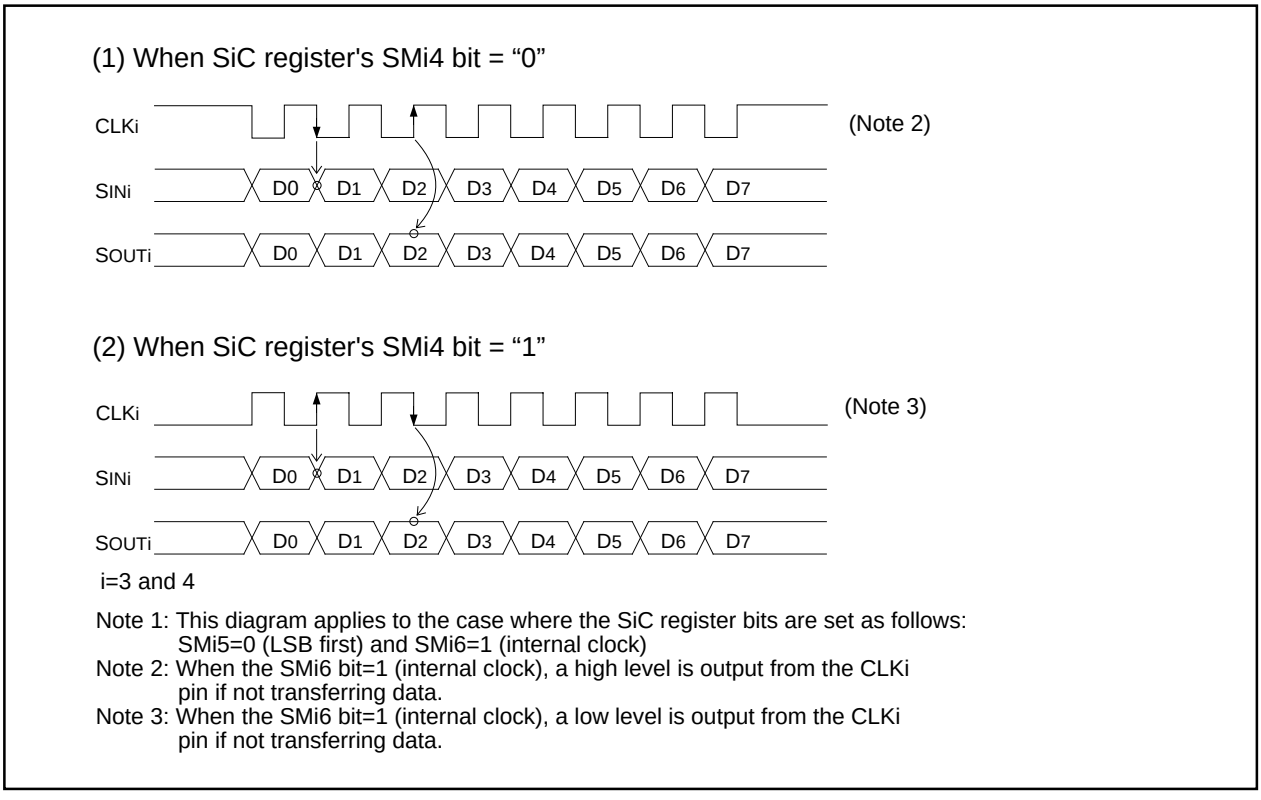


Note 1: This diagram applies to the case where the SiC register bits are set as follows:
SMi2=0 (SOUTi output), SMi3=1 (SOUTi output, CLKi function), SMi4=0 (transmit data output at the falling edge and receive data input at the rising edge of the transfer clock), SMi5=0 (LSB first) and SMi6=1 (internal clock)
Note 2: When the SMi6 bit = 1 (internal clock), the SOUTi pin is placed in the high-impedance state after the transfer finishes.
Note 3: If the SMi6 bit=0 (internal clock), the serial I/O starts sending or receiving data a maximum of 1.5 transfer clock cycles after writing to the SiTRR register.

Figure 1.21.3. SI/Oi Operation Timing

(b) CLK Polarity Selection

The SiC register's SMi4 bit allows selection of the polarity of the transfer clock. Figure 1.21.4 shows the polarity of the transfer clock.

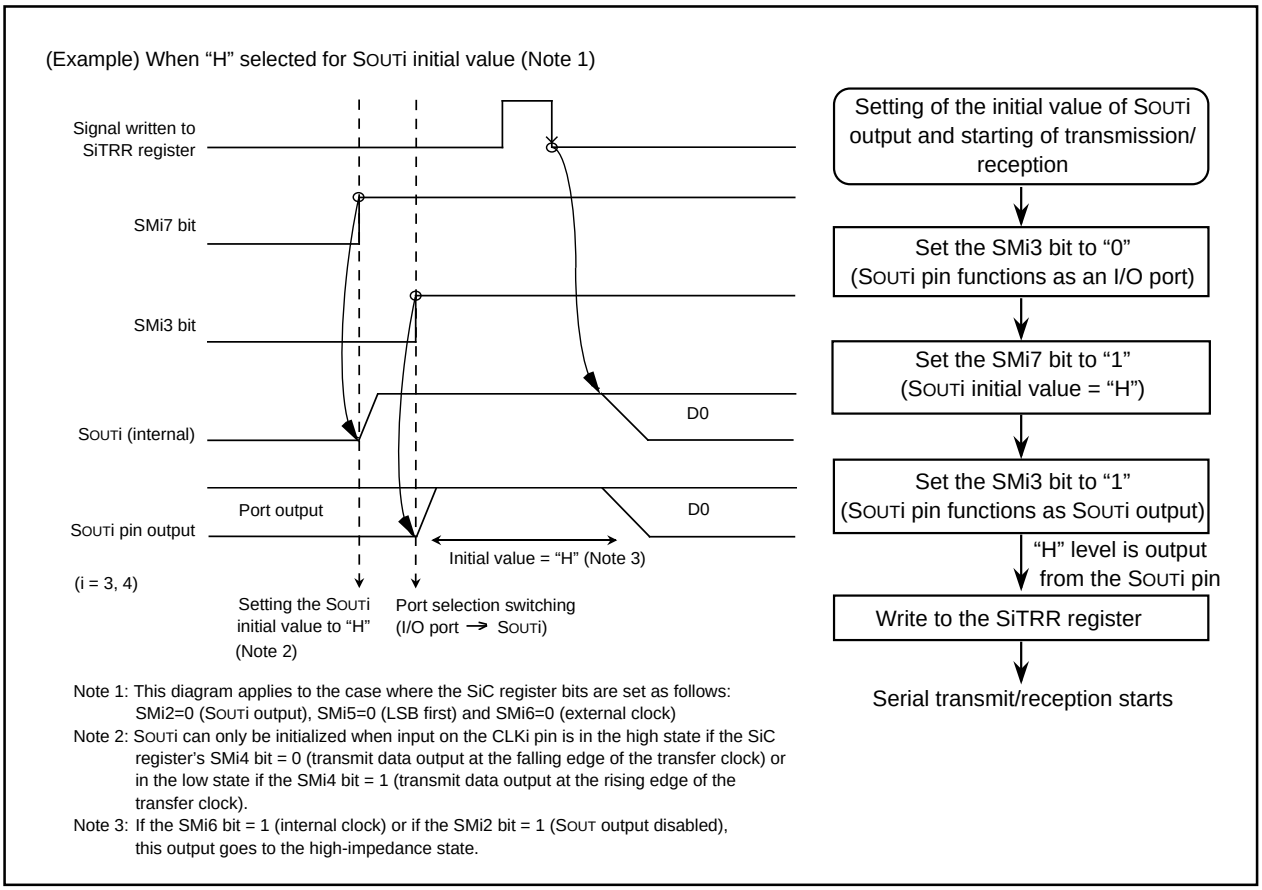


Note 1: This diagram applies to the case where the SiC register bits are set as follows:
SMi5=0 (LSB first) and SMi6=1 (internal clock)
Note 2: When the SMi6 bit=1 (internal clock), a high level is output from the CLKi pin if not transferring data.
Note 3: When the SMi6 bit=1 (internal clock), a low level is output from the CLKi pin if not transferring data.

Figure 1.21.4. Polarity of Transfer Clock

(c) Functions for Setting an SOUTi Initial Value

If the SiC register's SMi6 bit = 0 (external clock), the SOUTi pin output can be fixed high or low when not transferring. Figure 1.21.5 shows the timing chart for setting an SOUTi initial value and how to set it.



Note 1: This diagram applies to the case where the SiC register bits are set as follows: SMi2=0 (SOUTi output), SMi5=0 (LSB first) and SMi6=0 (external clock)
Note 2: SOUTi can only be initialized when input on the CLKi pin is in the high state if the SiC register's SMi4 bit = 0 (transmit data output at the falling edge of the transfer clock) or in the low state if the SMi4 bit = 1 (transmit data output at the rising edge of the transfer clock).
Note 3: If the SMi6 bit = 1 (internal clock) or if the SMi2 bit = 1 (SOUT output disabled), this output goes to the high-impedance state.

Figure 1.21.5. SOUTi's Initial Value Setting

A-D Converter

The microcomputer contains one A-D converter circuit based on 10-bit successive approximation method configured with a capacitive-coupling amplifier. The analog inputs share the pins with $P10_0$ to $P10_7$, $P9_5$, $P9_6$, $P0_0$ to $P0_7$, and $P2_0$ to $P2_7$. Similarly, $\overline{ADTRG}$ input shares the pin with $P9_7$. Therefore, when using these inputs, make sure the corresponding port direction bits are set to “0” (= input mode).

When not using the A-D converter, set the VCUT bit to “0” (= Vref unconnected), so that no current will flow from the V_{REF} pin into the resistor ladder, helping to reduce the power consumption of the chip.

The A-D conversion result is stored in the ADi register bits for ANi, AN0i, and AN2i pins (i = 0 to 7).

Table 1.22.1 shows the performance of the A-D converter. Figure 1.22.1 shows the block diagram of the A-D converter, and Figures 1.22.2 and 1.22.3 show the A-D converter-related registers.

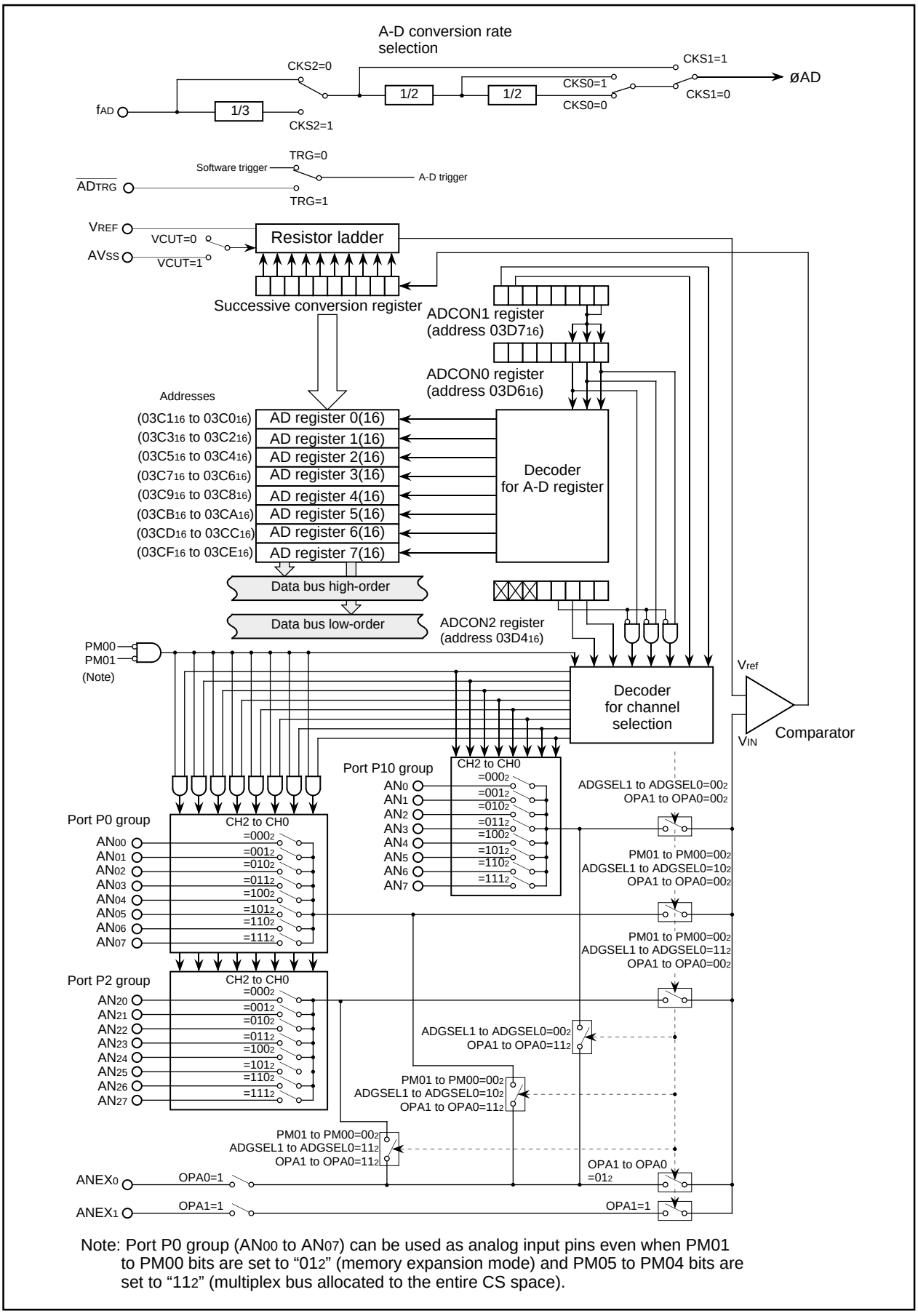
Table 1.22.1. Performance of A-D Converter

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AV_{CC} (V_{CC1})
Operating clock ϕ_{AD} (Note 2)	f_{AD}/divide-by-2 of f_{AD}/divide-by-3 of f_{AD}/divide-by-4 of f_{AD}/divide-by-6 of f_{AD}/divide-by-12 of f_{AD}
Resolution	8-bit or 10-bit (selectable)
Integral nonlinearity error	When AV_{CC} = V_{REF} = 5V • With 8-bit resolution: ±2LSB • With 10-bit resolution - AN_0 to AN_7 input : ±3LSB - AN_{00} to $AN0_7$ input and AN_{20} to AN_{27} input : ±7LSB - ANEX0 and ANEX1 input (including mode in which external operation amp is connected) : ±7LSB When AV_{CC} = V_{REF} = 3.3V • With 8-bit resolution: ±2LSB • With 10-bit resolution - AN_0 to AN_7 input : ±5LSB - AN_{00} to $AN0_7$ input and AN_{20} to AN_{27} input : ±7LSB - ANEX0 and ANEX1 input (including mode in which external operation amp is connected) : ±7LSB
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	8 pins (AN_0 to AN_7) + 2 pins (ANEX0 and ANEX1) + 8 pins (AN_{00} to $AN0_7$) + 8 pins (AN_{20} to AN_{27})
A-D conversion start condition	• Software trigger The ADCON0 register's ADST bit is set to “1” (A-D conversion starts) • External trigger (retriggerable) Input on the $\overline{ADTRG}$ pin changes state from high to low after the ADST bit is set to “1” (A-D conversion starts)
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: The f_{AD} frequency must be 10 MHz or less.
Without sample-and-hold function, limit the f_{AD} frequency to 250kHz or less.
With the sample and hold function, limit the f_{AD} frequency to 1MHz or less.

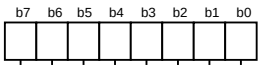
Note 3: If $V_{CC2} < V_{CC1}$, do not use AN_{00} to $AN0_7$ and AN_{20} to AN_{27} as analog input pins.



Note: Port P0 group (AN00 to AN07) can be used as analog input pins even when PM01 to PM00 bits are set to "012" (memory expansion mode) and PM05 to PM04 bits are set to "112" (multiplex bus allocated to the entire CS space).

Figure 1.22.1. A-D Converter Block Diagram

A-D control register 0 (Note)

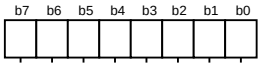


Symbol: ADCON0 | Address: 03D6$_{16}$ | After reset: 00000XXX$_2$

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit	Function varies with each operation mode	RW
CH1			RW
CH2			RW
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : $\overline{\text{AD}}_{\text{TRG}}$ trigger	RW
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	RW
CKS0	Frequency select bit 0	See Note 3 for the ADCON2 register	RW

Note: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.

A-D control register 1 (Note 1)



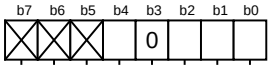
Symbol: ADCON1 | Address: 03D7$_{16}$ | After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SCAN0	A-D sweep pin select bit	Function varies with each operation mode	RW
SCAN1			RW
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1 1 : Repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1	See Note 3 for the ADCON2 register	RW
VCUT	Vref connect bit (Note 2)	0 : Vref not connected 1 : Vref connected	RW
OPA0	External op-amp connection mode bit	Function varies with each operation mode	RW
OPA1			RW

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: If the VCUT bit is reset from “0” (Vref unconnected) to “1” (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 1.22.2. ADCON0 to ADCON1 Registers

A-D control register 2 (Note 1)



Symbol	Address	After reset
ADCON2	03D4₁₆	00₁₆

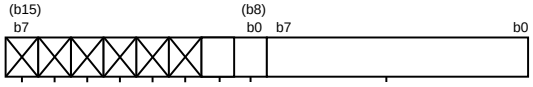
Bit symbol	Bit name	Function	RW
SMP	A-D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	RW
ADGSEL0	A-D input group select bit	b2 b1 0 0 : Port P10 group is selected 0 1 : Must not be set 1 0 : Port P0 group is selected (Note 3) 1 1 : Port P2 group is selected	RW
ADGSEL1			RW
(b3)	Reserved bit	Must always be set to "0"	RW
CKS2	Frequency select bit 2 (Note 3)	0: Selects f_{AD}, f_{AD} divided by 2, or f_{AD} divided by 4. 1: Selects f_{AD} divided by 3, f_{AD} divided by 6, or f_{AD} divided by 12.	RW
(b7-b5)	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—

Note 1: If the ADCON2 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: If $V_{CC2} < V_{CC1}$, do not use AN_{00} to AN_{07} and AN_{20} to AN_{27} as analog input pins.
Note 3: The $Ø_{AD}$ frequency must be 10 MHz or less. The selected $Ø_{AD}$ frequency is determined by a combination of the ADCON0 register's CKS0 bit, ADCON1 register's CKS1 bit, and ADCON2 register's CKS2 bit.

CKS0	CKS1	CKS2	$Ø_{AD}$
0	0	0	Divide-by-4 of f_{AD}
0	0	1	Divide-by-2 of f_{AD}
0	1	0	f_{AD}
0	1	1	
1	0	0	Ddivide-by-12 of f_{AD}
1	0	1	Divide-by-6 of f_{AD}
1	1	0	Divide-by-3 of f_{AD}
1	1	1	

A-D register i (i=0 to 7)

Symbol	Address	After reset
AD0	03C1₁₆ to 03C0₁₆	Indeterminate
AD1	03C3₁₆ to 03C2₁₆	Indeterminate
AD2	03C5₁₆ to 03C4₁₆	Indeterminate
AD3	03C7₁₆ to 03C6₁₆	Indeterminate
AD4	03C9₁₆ to 03C8₁₆	Indeterminate
AD5	03CB₁₆ to 03CA₁₆	Indeterminate
AD6	03CD₁₆ to 03CC₁₆	Indeterminate
AD7	03CF₁₆ to 03CE₁₆	Indeterminate



Function		RW
When the ADCON1 register's BITS bit is "1" (10-bit mode)	When the ADCON1 register's BITS bit is "0" (8-bit mode)	
Eight low-order bits of A-D conversion result	A-D conversion result	RO
Two high-order bits of A-D conversion result	When read, the content is indeterminate	RO
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—

Figure 1.22.3. ADCON2 Register, and AD0 to AD7 Registers

(1) One-shot Mode

In this mode, the input voltage on one selected pin is A-D converted once. Table 1.22.2 shows the specifications of one-shot mode. Figure 1.22.4 shows the ADCON0 to ADCON1 registers in one-shot mode.

Table 1.22.2. One-shot Mode Specifications

Item	Specification
Function	The input voltage on one pin selected by the ADCON0 register's CH2 to CH0 bits and ADCON2 register's ADGSEL1 to ADGSEL0 bits or the ADCON1 register's OPA1 to OPA0 bits is A-D converted once.
A-D conversion start condition	• When the ADCON0 register's TRG bit is “0” (software trigger) The ADCON0 register's ADST bit is set to “1” (A-D conversion starts) • When the TRG bit is “1” ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to “1” (A-D conversion starts)
A-D conversion stop condtision	• Completion of A-D conversion (If a software trigger is selected, the ADST bit is cleared to “0” (A-D conversion halted).) • Set the ADST bit to “0”
Interrupt request generation timing	Completion of A-D conversion
Analog input pin (Note)	Select one pin from AN_0 to AN_7, AN_{00} to AN_{07}, AN_{20} to AN_{27}, ANEX0 to ANEX1
Reading of result of A-D converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

Note: If $V_{CC2} < V_{CC1}$, do not use AN_{00}–AN_{07} and AN_{20}–AN_{27} as analog input pins.

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	0			

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

Bit symbol	Bit name	Function	RW
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 (Note 2) 1 1 1 : AN$_7$ is selected (Note 3)	RW
CH1			RW
CH2			RW
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode (Note 3)	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	RW
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	RW
CKS0	Frequency select bit 0	See Note 3 for the ADCON2 register	RW

Note 1: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: AN$_{00}$ to AN$_{07}$, and AN$_{20}$ to AN$_{27}$ can be used in the same way as AN$_0$ to AN$_7$. Use the ADCON2 register's ADGSEL1 to ADGSEL0 bits to select the desired pin. However, if $V_{CC2} < V_{CC1}$, do not use AN$_{00}$ to AN$_{07}$ and AN$_{20}$ to AN$_{27}$ as analog input pins.
Note 3: After rewriting the MD1 to MD0 bits, set the CH2 to CH0 bits over again using another instruction.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		1			0		

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

Bit symbol	Bit name	Function	RW
SCAN0	A-D sweep pin select bit	Invalid in one-shot mode	RW
SCAN1			RW
MD2	A-D operation mode select bit 1	Set to "0" when one-shot mode is selected	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit1	See Note 3 for the ADCON2 register	RW
VCUT	Vref connect bit (Note 2)	1 : Vref connected	RW
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	RW
OPA1			RW

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: If the VCUT bit is reset from "0" (Vref unconnected) to "1" (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 1.22.4. ADCON0 Register and ADCON1 Register (One-shot Mode)

(2) Repeat mode

In this mode, the input voltage on one selected pin is A-D converted repeatedly. Table 1.22.3 shows the specifications of repeat mode. Figure 1.22.5 shows the ADCON0 to ADCON1 registers in repeat mode.

Table 1.22.3. Repeat Mode Specifications

Item	Specification
Function	The input voltage on one pin selected by the ADCON0 register's CH2 to CH0 bits and ADCON2 register's ADGSEL1 to ADGSEL0 bits or the ADCON1 register's OPA1 to OPA0 bits is A-D converted repeatdly.
A-D conversion start condition	• When the ADCON0 register's TRG bit is “0” (software trigger) The ADCON0 register's ADST bit is set to “1” (A-D conversion starts) • When the TRG bit is “1” ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to “1” (A-D conversion starts)
A-D conversion stop condtision	Set the ADST bit to “0” (A-D conversion halted)
Interrupt request generation timing	None generated
Analog input pin (Note)	Select one pin from AN_0 to AN_7, AN_{00} to AN_{07}, AN_{20} to AN_{27}, ANEX0 to ANEX1
Reading of result of A-D converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

Note: If $V_{CC2} < V_{CC1}$, do not use AN_{00}–AN_{07} and AN_{20}–AN_{27} as analog input pins.

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	1			

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

Bit symbol	Bit name	Function	RW
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 (Note 2) 1 1 1 : AN$_7$ is selected (Note 3)	RW
CH1			RW
CH2			RW
MD0	A-D operation mode select bit 0	b4 b3 0 1 : Repeat mode (Note 3)	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	RW
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	RW
CKS0	Frequency select bit 0	See Note 3 for the ADCON2 register	RW

Note 1: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: AN$_{00}$ to AN$_{07}$, and AN$_{20}$ to AN$_{27}$ can be used in the same way as AN$_0$ to AN$_7$. Use the ADCON2 register's ADGSEL1 to ADGSEL0 bits to select the desired pin. However, if $V_{CC2} < V_{CC1}$, do not use AN$_{00}$ to AN$_{07}$ and AN$_{20}$ to AN$_{27}$ as analog input pins.
Note 3: After rewriting the MD1 to MD0 bits, set the CH2 to CH0 bits over again using another instruction.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		1			0		

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

Bit symbol	Bit name	Function	RW
SCAN0	A-D sweep pin select bit	Invalid in repeat mode	RW
SCAN1			RW
MD2	A-D operation mode select bit 1	Set to "0" when this mode is selected	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1	See Note 3 for the ADCON2 register	RW
VCUT	Vref connect bit (Note 2)	1 : Vref connected	RW
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	RW
OPA1			RW

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: If the VCUT bit is reset from "0" (Vref unconnected) to "1" (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 1.22.5. ADCON0 Register and ADCON1 Register (Repeat Mode)

(3) Single Sweep Mode

In this mode, the input voltages on selected pins are A-D converted, one pin at a time. Table 1.22.4 shows the specifications of single sweep mode. Figure 1.22.6 shows the ADCON0 to ADCON1 registers in single sweep mode.

Table 1.22.4. Single Sweep Mode Specifications

Item	Specification
Function	The input voltages on pins selected by the ADCON1 register's SCAN1 to SCAN0 bits and ADCON2 register's ADGSEL1 to ADGSEL0 bits are A-D converted, one pin at a time.
A-D conversion start condition	• When the ADCON0 register's TRG bit is “0” (software trigger) The ADCON0 register's ADST bit is set to “1” (A-D conversion starts) • When the TRG bit is “1” ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to “1” (A-D conversion starts)
A-D conversion stop condtision	• Completion of A-D conversion (If a software trigger is selected, the ADST bit is cleared to “0” (A-D conversion halted).) • Set the ADST bit to “0”
Interrupt request generation timing	Completion of A-D conversion
Analog input pin	Select from AN_0 to AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), AN_0 to AN_7 (8 pins) (Note)
Reading of result of A-D converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

Note: AN_{00} to AN_{07}, and AN_{20} to AN_{27} can be used in the same way as AN_0 to AN_7. However, if $V_{CC2} < V_{CC1}$, do not use AN_{00}–AN_{07} and AN_{20}–AN_{27} as analog input pins.

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
			1	0			

Symbol: ADCON0 — Address: 03D6$_{16}$ — After reset: 00000XXX$_2$

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit	Invalid in single sweep mode	RW
CH1			RW
CH2			RW
MD0	A-D operation mode select bit 0	b4 b3 1 0 : Single sweep mode	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	RW
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	RW
CKS0	Frequency select bit 0	See Note 3 for the ADCON2 register	RW

Note: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		1			0		

Symbol: ADCON1 — Address: 03D7$_{16}$ — After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SCAN0	A-D sweep pin select bit	When single sweep mode is selected b1 b0 0 0 : AN$_0$ to AN$_1$ (2 pins) 0 1 : AN$_0$ to AN$_3$ (4 pins) 1 0 : AN$_0$ to AN$_5$ (6 pins) 1 1 : AN$_0$ to AN$_7$ (8 pins) (Note 2)	RW
SCAN1			RW
MD2	A-D operation mode select bit 1	Set to “0” when single sweep mode is selected	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1	See Note 3 for the ADCON2 register	RW
VCUT	Vref connect bit (Note 3)	1 : Vref connected	RW
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : Must not be set 1 0 : Must not be set 1 1 : External op-amp connection mode	RW
OPA1			RW

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: AN$_{00}$ to AN$_{07}$, and AN$_{20}$ to AN$_{27}$ can be used in the same way as AN$_0$ to AN$_7$. Use the ADCON2 register’s ADGSEL1 to ADGSEL0 bits to select the desired pin. However, if $V_{CC2} < V_{CC1}$, do not use AN$_{00}$ to AN$_{07}$ and AN$_{20}$ to AN$_{27}$ as analog input pins.
Note 3: If the VCUT bit is reset from “0” (Vref unconnected) to “1” (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 1.22.6. ADCON0 Register and ADCON1 Register (Single Sweep Mode)

(4) Repeat Sweep Mode 0

In this mode, the input voltages on selected pins are A-D converted repeatedly. Table 1.22.5 shows the specifications of repeat sweep mode 0. Figure 1.22.7 shows the ADCON0 to ADCON1 registers in repeat sweep mode 0.

Table 1.22.5. Repeat Sweep Mode 0 Specifications

Item	Specification
Function	The input voltages on pins selected by the ADCON1 register's SCAN1 to SCAN0 bits and ADCON2 register's ADGSEL1 to ADGSEL0 bits are A-D converted repeatdly.
A-D conversion start condition	• When the ADCON0 register's TRG bit is “0” (software trigger) The ADCON0 register's ADST bit is set to “1” (A-D conversion starts) • When the TRG bit is “1” ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to “1” (A-D conversion starts)
A-D conversion stop condtision	Set the ADST bit to “0” (A-D conversion halted)
Interrupt request generation timing	None generated
Analog input pin	Select from AN_0 to AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), AN_0 to AN_7 (8 pins) (Note)
Reading of result of A-D converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

Note: AN_{00} to AN_{07}, and AN_{20} to AN_{27} can be used in the same way as AN_0 to AN_7. However, if $V_{CC2} < V_{CC1}$, do not use AN_{00}–AN_{07} and AN_{20}–AN_{27} as analog input pins.

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
			1	1			

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

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit	Invalid in repeat sweep mode 0	RW
CH1			RW
CH2			RW
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : AD_{TRG} trigger	RW
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	RW
CKS0	Frequency select bit 0	See Note 3 for the ADCON2 register	RW

Note: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		1			0		

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

Bit symbol	Bit name	Function	RW
SCAN0	A-D sweep pin select bit	When repeat sweep mode 0 is selected b1 b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0 to AN_3 (4 pins) 1 0 : AN_0 to AN_5 (6 pins) 1 1 : AN_0 to AN_7 (8 pins) (Note 2)	RW
SCAN1			RW
MD2	A-D operation mode select bit 1	Set to “0” when repeat sweep mode 0 is selected	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1	See Note 3 for the ADCON2 register	RW
VCUT	Vref connect bit (Note 3)	1 : Vref connected	RW
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : Must not be set 1 0 : Must not be set 1 1 : External op-amp connection mode	RW
OPA1			RW

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: AN_{00} to AN_{07}, and AN_{20} to AN_{27} can be used in the same way as AN_0 to AN_7. Use the ADCON2 register’s ADGSEL1 to ADGSEL0 bits to select the desired pin. However, if $V_{CC2} < V_{CC1}$, do not use AN_{00} to AN_{07} and AN_{20} to AN_{27} as analog input pins.
Note 3: If the VCUT bit is reset from “0” (Vref unconnected) to “1” (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 1.22.7. ADCON0 Register and ADCON1 Registers (Repeat Sweep Mode 0)

(5) Repeat Sweep Mode 1

In this mode, the input voltages on all pins are A-D converted repeatedly, with priority given to the selected pins. Table 1.22.6 shows the specifications of repeat sweep mode 1. Figure 1.22.8 shows the ADCON0 to ADCON1 registers in repeat sweep mode 1.

Table 1.22.6. Repeat Sweep Mode 1 Specifications

Item	Specification
Function	The input voltages on all pins selected by the ADCON2 register's ADGSEL1 to ADGSEL0 bits are A-D converted repeatdly, with priority given to pins selected by the ADCON1 register's SCAN1 to SCAN0 bits and ADGSEL1 to ADGSEL0 bits. Example : If AN_0 selected, input voltages are A-D converted in order of $AN_0 \rightarrow AN_1 \rightarrow AN_0 \rightarrow AN_2 \rightarrow AN_0 \rightarrow AN_3$, and so on.
A-D conversion start condition	• When the ADCON0 register's TRG bit is “0” (software trigger) The ADCON0 register's ADST bit is set to “1” (A-D conversion starts) • When the TRG bit is “1” ($\overline{ADTRG}$ trigger) Input on the $\overline{ADTRG}$ pin changes state from high to low after the ADST bit is set to “1” (A-D conversion starts)
A-D conversion stop condtision	Set the ADST bit to “0” (A-D conversion halted)
Interrupt request generation timing	None generated
Analog input pins to be given priority when A-D converted	Select from AN_0 (1 pins), AN_0 to AN_1 (2 pins), AN_0 to AN_2 (3 pins), AN_0 to AN_3 (4 pins) (Note)
Reading of result of A-D converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

Note: AN_{00} to AN_{07}, and AN_{20} to AN_{27} can be used in the same way as AN_0 to AN_7. However, if $V_{CC2} < V_{CC1}$, do not use AN_{00}–AN_{07} and AN_{20}–AN_{27} as analog input pins.

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
			1	1			

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

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit	Invalid in repeat sweep mode 1	RW
CH1			RW
CH2			RW
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : AD_{TRG} trigger	RW
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	RW
CKS0	Frequency select bit 0	See Note 3 for the ADCON2 register	RW

Note: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		1			1		

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

Bit symbol	Bit name	Function	RW
SCAN0	A-D sweep pin select bit	When repeat sweep mode 1 is selected b1 b0 0 0 : AN_0 (1 pin) 0 1 : AN_0, AN_1 (2 pins) 1 0 : AN_0 to AN_2 (3 pins) 1 1 : AN_0 to AN_3 (4 pins) (Note 2)	RW
SCAN1			RW
MD2	A-D operation mode select bit 1	Set to "1" when repeat sweep mode 1 is selected	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1	See Note 3 for the ADCON2 register	RW
VCUT	Vref connect bit (Note 3)	1 : Vref connected	RW
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : Must not be set 1 0 : Must not be set 1 1 : External op-amp connection mode	RW
OPA1			RW

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.
Note 2: AN_{00} to AN_{07}, and AN_{20} to AN_{27} can be used in the same way as AN_0 to AN_7. Use the ADCON2 register's ADGSEL1 to ADGSEL0 bits to select the desired pin. However, if $V_{CC2} < V_{CC1}$, do not use AN_{00} to AN_{07} and AN_{20} to AN_{27} as analog input pins.
Note 3: If the VCUT bit is reset from "0" (Vref unconnected) to "1" (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 1.22.8. ADCON0 Register and ADCON1 Register (Repeat Sweep Mode 1)

(a) Resolution Select Function

The desired resolution can be selected using the ADCON1 register's BITS bit. If the BITS bit is set to "1" (10-bit conversion accuracy), the A-D conversion result is stored in the ADi register (i = 0 to 7)'s bit 0 to bit 9. If the BITS bit is set to "0" (8-bit conversion accuracy), the A-D conversion result is stored in the ADi register's bit 0 to bit 7.

(b) Sample and Hold

If the ADCON2 register's SMP bit is set to "1" (with sample-and-hold), the conversion speed per pin is increased to 28 ØAD cycles for 8-bit resolution or 33 ØAD cycles for 10-bit resolution. Sample-and-hold is effective in all operation modes. Select whether or not to use the sample-and-hold function before starting A-D conversion.

(c) Extended Analog Input Pins

In one-shot and repeat modes, the ANEX0 and ANEX1 pins can be used as analog input pins. Use the ADCON1 register's OPA1 to OPA0 bits to select whether or not use ANEX0 and ANEX1.

The A-D conversion results of ANEX0 and ANEX1 inputs are stored in the AD0 and AD1 registers, respectively.

(d) External Operation Amp Connection Mode

Multiple analog inputs can be amplified using a single external op-amp via the ANXE0 and ANEX1 pins. Set the ADCON1 register's OPA1 OPA0 bits to '11_2' (external op-amp connection mode). The inputs from ANi (i = 0 to 7) (Note 1) are output from the ANEX0 pin. Amplify this output with an external op-amp before sending it back to the ANEX1 pin. The A-D conversion result is stored in the corresponding ADi register. The A-D conversion speed depends on the response characteristics of the external op-amp. Note that the ANXE0 and ANEX1 pins cannot be directly connected to each other. Figure 1.22.9 is an example of how to connect the pins in external operation amp.

Note: AN0i and AN2i can be used the same as ANi. However, if VCC2 < VCC1, do not use AN0i and AN2i as analog input pins.

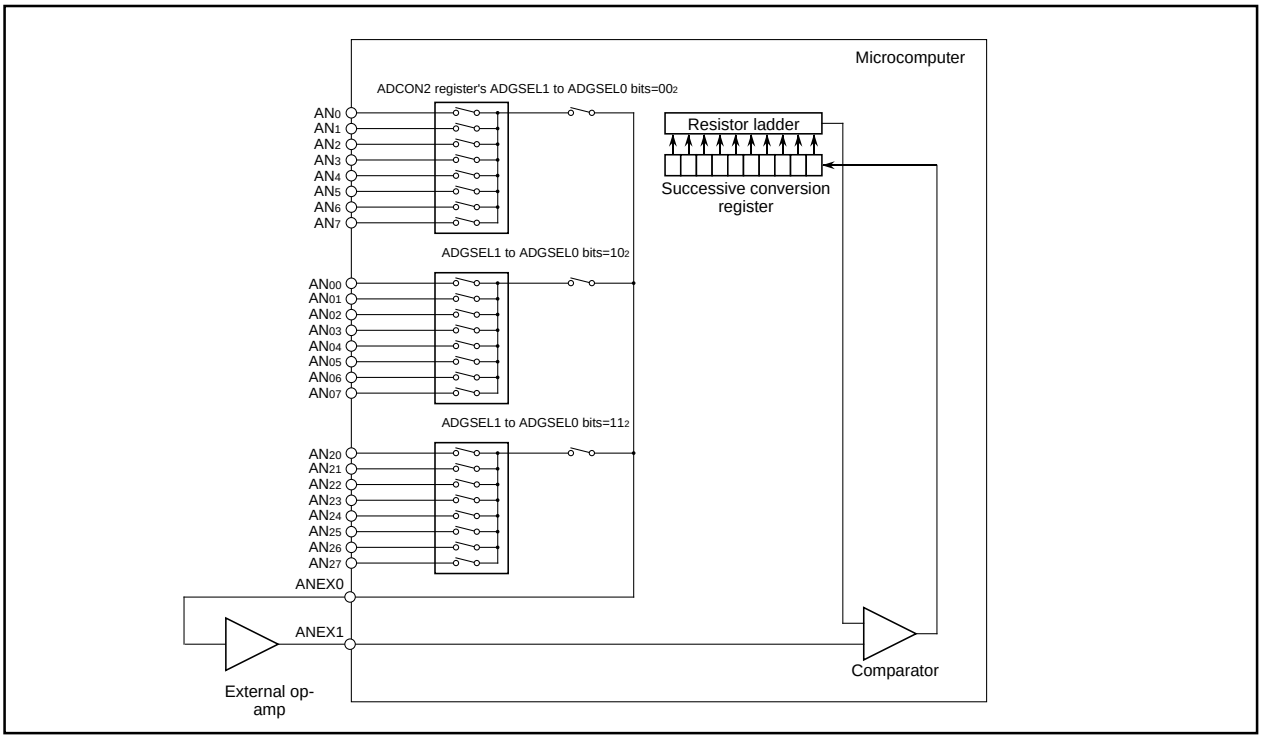


Figure 1.22.9. External Op-amp Connection

(e) Current Consumption Reducing Function

When not using the A-D converter, its resistor ladder and reference voltage input pin (V_{REF}) can be separated using the ADCON1 register's VCUT bit. When separated, no current will flow from the V_{REF} pin into the resistor ladder, helping to reduce the power consumption of the chip.

To use the A-D converter, set the VCUT bit to "1" (V_{REF} connected) and then set the ADCON0 register's ADST bit to "1" (A-D conversion start). The VCUT and ADST bits cannot be set to "1" at the same time.

Nor can the VCUT bit be set to "0" (V_{REF} unconnected) during A-D conversion.

Note that this does not affect V_{REF} for the D-A converter (irrelevant).

(f) Analog Input Pin and External Sensor Equivalent Circuit Example

Figure 1.22.10 shows analog input pin and external sensor equivalent circuit example.

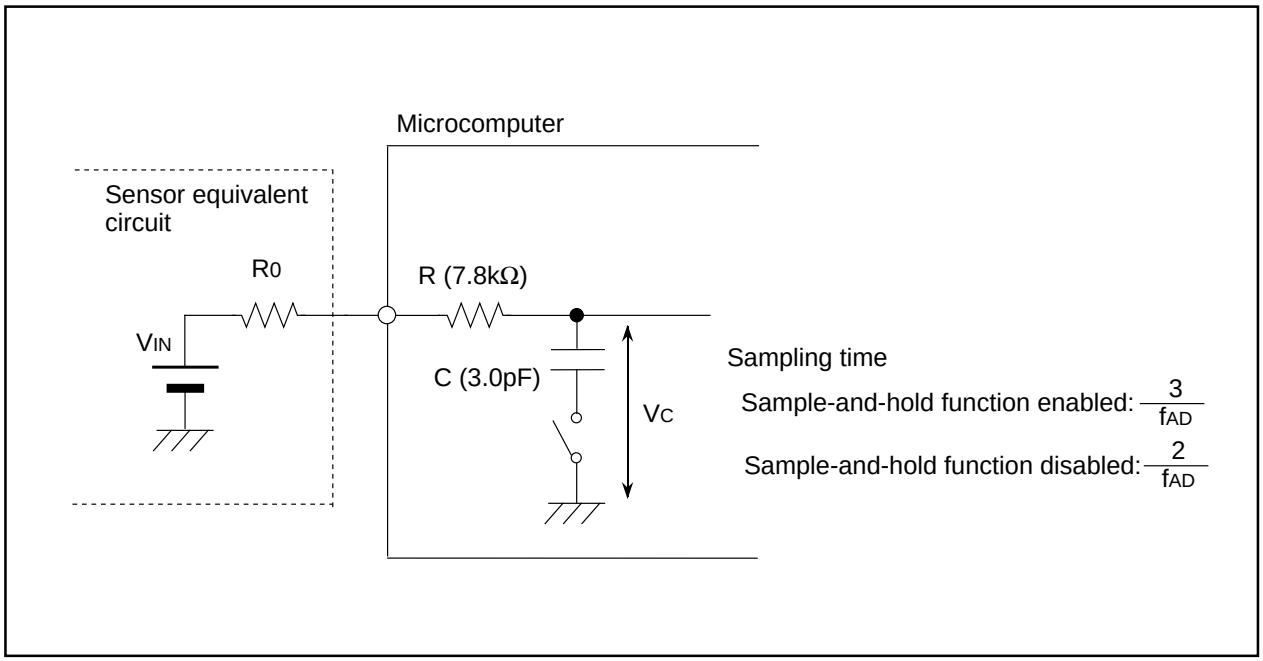


Figure 1.22.10. Analog Input Pin and External Sensor Equivalent Circuit

(g) Caution of Using A-D Converter

(1) Make sure the port direction bits for those pins that are used as analog inputs are set to “0” (input mode). Also, if the ADCON0 register’s TGR bit = 1 (external trigger), make sure the port direction bit for the $\overline{\text{ADTRG}}$ pin is set to “0” (input mode).

(2) When using key input interrupts, do not use any of the four AN_4 to AN_7 pins as analog inputs. (A key input interrupt request is generated when the A-D input voltage goes low.)

(3) To prevent noise-induced device malfunction or latchup, as well as to reduce conversion errors, insert capacitors between the AV_{CC}, V_{REF}, and analog input pins (ANi (i=0 to 7), AN_{0i}, and AN_{2i}) each and the AV_{SS} pin. Similarly, insert a capacitor between the V_{CC} pin and the V_{SS} pin. Figure 1.22.11 is an example connection of each pin.

(4) If $V_{CC2} < V_{CC1}$, do not use AN_{00} to AN_{07} and AN_{20} to AN_{27} as analog input pins.

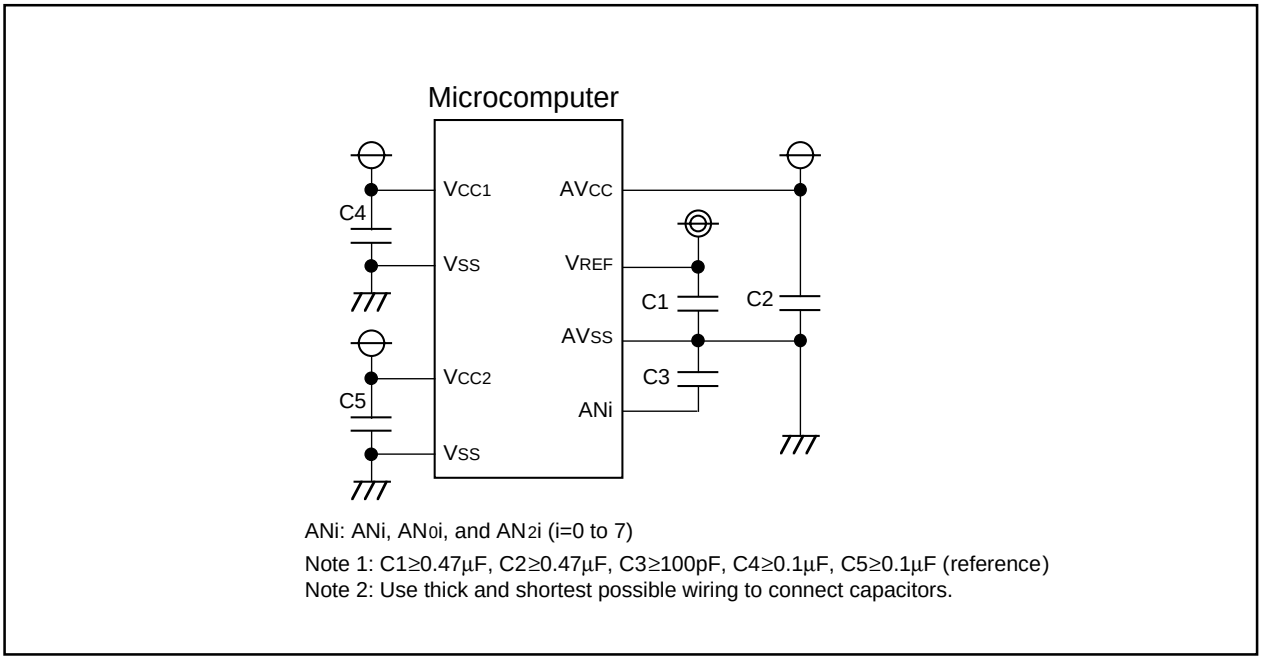
(5) If the CPU reads the ADi register (i = 0 to 7) at the same time the conversion result is stored in the ADi register after completion of A-D conversion, an incorrect value may be stored in the ADi register. This problem occurs when a divide-by-n clock derived from the main clock or a subclock is selected for CPU clock.

- When operating in one-shot or single-sweep mode

  Check to see that A-D conversion is completed before reading the target ADi register. (Check the IR bit in the ADIC register to see if A-D conversion is completed.)
- When operating in repeat mode or repeat sweep mode 0 or 1

  Use the main clock for CPU clock directly without dividing it.

(6) If A-D conversion is forcibly terminated while in progress by setting the ADCON0 register’s ADST bit to “0” (A-D conversion halted), the conversion result of the A-D converter is indeterminate. The contents of ADi registers irrelevant to A-D conversion may also become indeterminate. If while A-D conversion is underway the ADST bit is cleared to “0” in a program, ignore the values of all ADi registers.



ANi: ANi, AN_{0i}, and AN_{2i} (i=0 to 7)

Note 1: C1≥0.47µF, C2≥0.47µF, C3≥100pF, C4≥0.1µF, C5≥0.1µF (reference)
Note 2: Use thick and shortest possible wiring to connect capacitors.

Figure 1.22.11. V_{CC}, V_{SS}, AV_{CC}, AV_{SS}, V_{REF} and ANi Connection

D-A Converter

This is an 8-bit, R-2R type D-A converter. These are two independent D-A converters.

D-A conversion is performed by writing to the DAi register (i = 0 to 1). To output the result of conversion, set the DACON register's DAiE bit to "1" (output enabled). Before D-A conversion can be used, the corresponding port direction bit must be cleared to "0" (input mode). Setting the DAiE bit to "1" removes a pull-up from the corresponding port.

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

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

V_{REF} : reference voltage

Table 1.23.1 lists the performance of the D-A converter. Figure 1.23.1 shows the block diagram of the D-A converter. Figure 1.23.2 shows the D-A converter related registers. Figure 1.23.3 shows the D-A converter equivalent circuit.

Table 1.23.1. D-A Converter Performance

Item	Performance
D-A conversion method	R-2R method
Resolution	8 bits
Analog output pin	2 (DA0 and DA1)

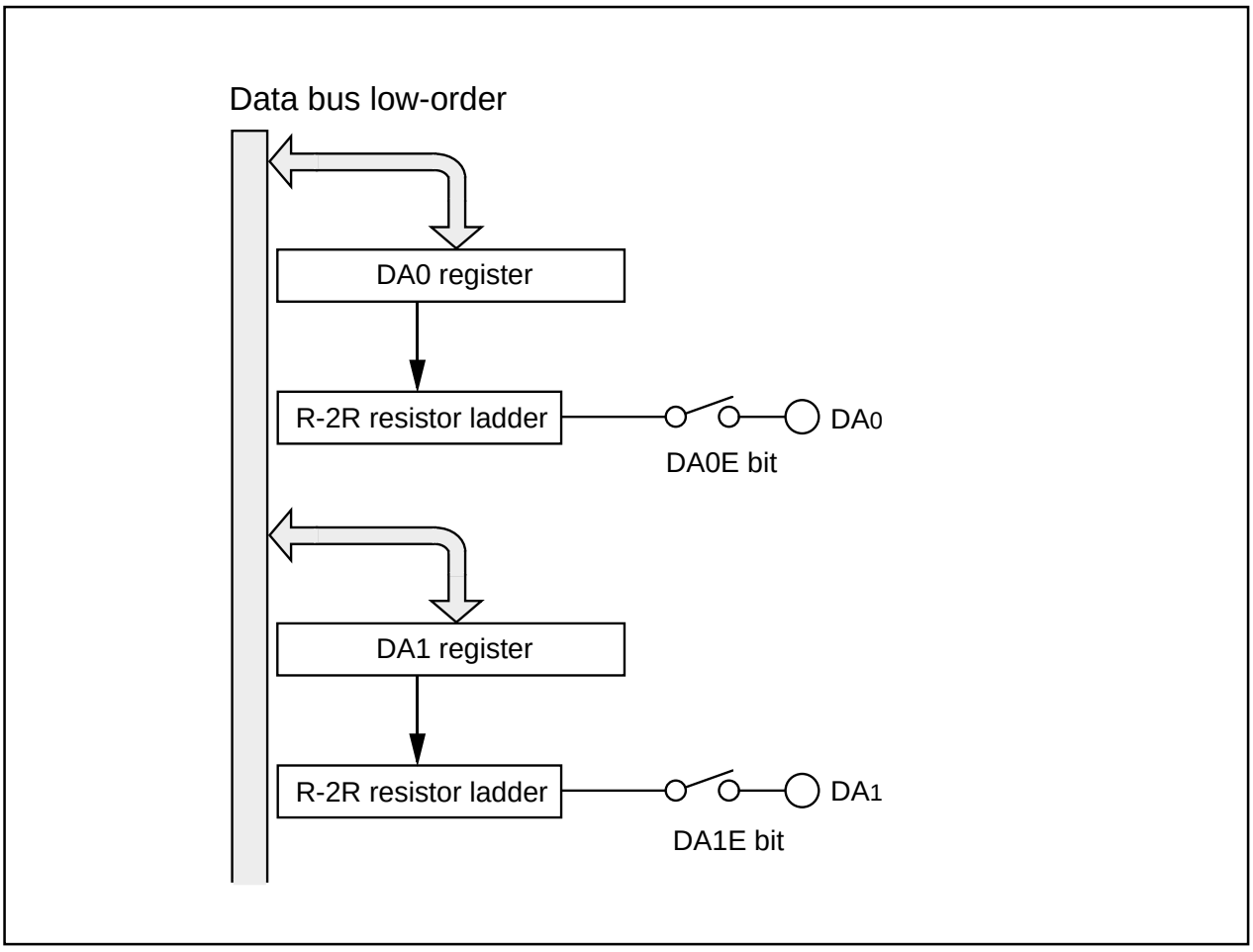
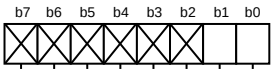


Figure 1.23.1. D-A Converter Block Diagram

D-A control register (Note)

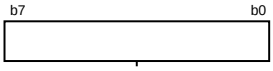


Symbol	Address	After reset
DACON	03DC16	0016

Bit symbol	Bit name	Function	RW
DA0E	D-A0 output enable bit	0 : Output disabled 1 : Output enabled	RW
DA1E	D-A1 output enable bit	0 : Output disabled 1 : Output enabled	RW
— (b7-b2)	Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”		RW

Note: When not using the D-A converter, clear the DAiE bit (i = 0 to 1) to “0” (output disabled) to reduce the unnecessary current consumption in the chip and set the DAi register to ‘0016’ to prevent current from flowing into the R-2R resistor ladder.

D-Ai register (Note) (i= 0 to 1)

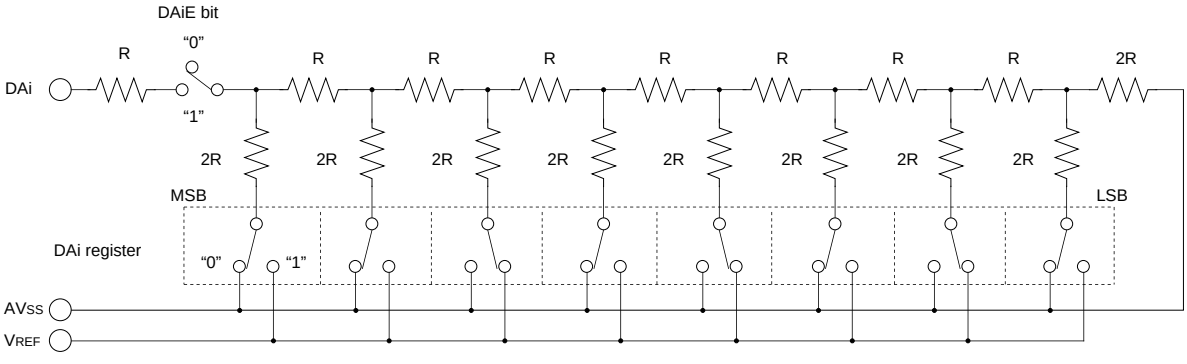


Symbol	Address	After reset
DA0	03D816	Indeterminate
DA1	03DA16	Indeterminate

Function	RW
Output value of D-A conversion	RW

Note: When not using the D-A converter, clear the DAiE bit (i = 0 to 1) to “0” (output disabled) to reduce the unnecessary current consumption in the chip and set the DAi register to ‘0016’ to prevent current from flowing into the R-2R resistor ladder.

Figure 1.23.2. DACON Register, DA0 Register, and DA1 Register



Note: The above diagram shows an instance in which the DA0 register is assigned “2A16”.

Figure 1.23.3. D-A Converter Equivalent Circuit

CRC Calculation

The Cyclic Redundancy Check (CRC) operation detects an error in data blocks. The microcomputer uses a generator polynomial of CRC_CCITT ($X^{16} + X^{12} + X^5 + 1$) to generate CRC code.

The CRC code consists of 16 bits which are generated for each data block in given length, separated in 8 bit units. After the initial value is set in the CRCD register, the CRC code is set in that register each time one byte of data is written to the CRCIN register. CRC code generation for one-byte data is finished in two cycles.

Figure 1.24.1 shows the block diagram of the CRC circuit. Figure 1.24.2 shows the CRC-related registers. Figure 1.24.3 shows the calculation example using the CRC operation.

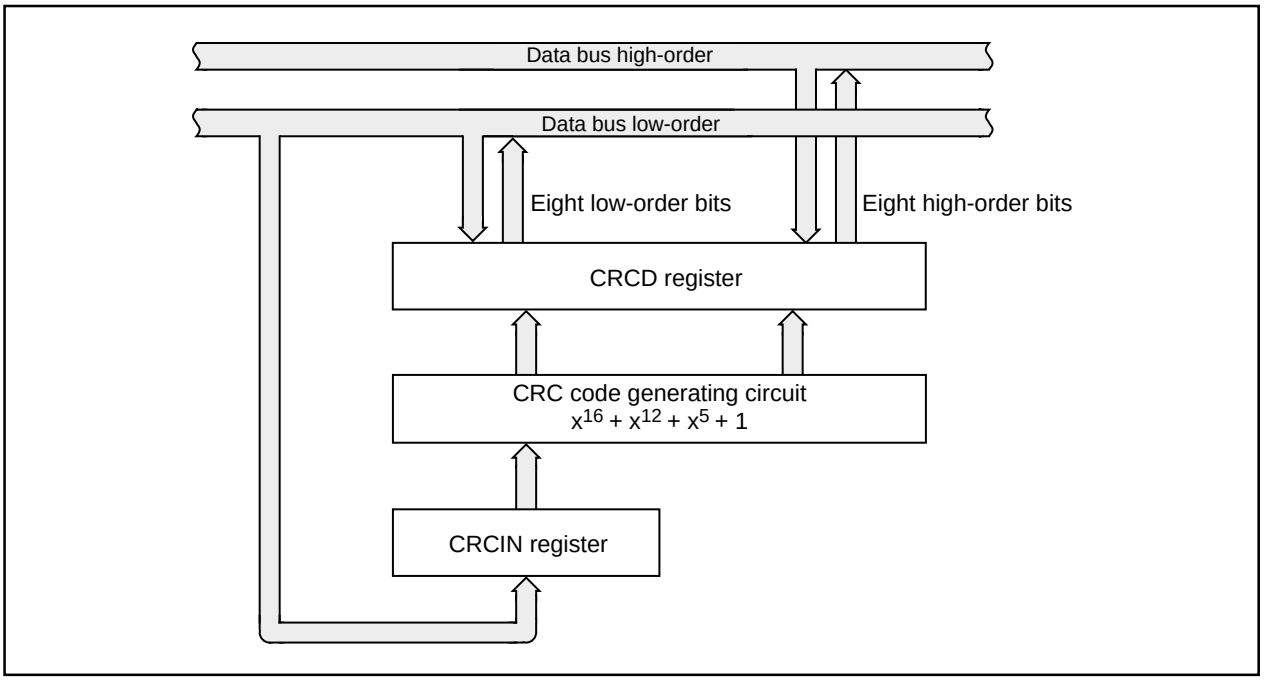


Figure 1.24.1. CRC Circuit Block Diagram

CRC data register

(b15) (b8) b7 b0 b7 b0

Symbol	Address	After reset
CRCD	03BD16 to 03BC16	Indeterminate

Function	Setting range	RW
When data is written to the CRCIN register after setting the initial value in the CRCD register, the CRC code can be read out from the CRCD register.	000016 to FFFF16	RW

CRC input register

b7 b0

Symbo	Address	After reset
CRCIN	03BE16	Indeterminate

Function	Setting range	RW
Data input	0016 to FF16	RW

Figure 1.24.2. CRCD Register and CRCIN Register

Setup procedure and CRC operation when generating CRC code "80C4$_{16}$"

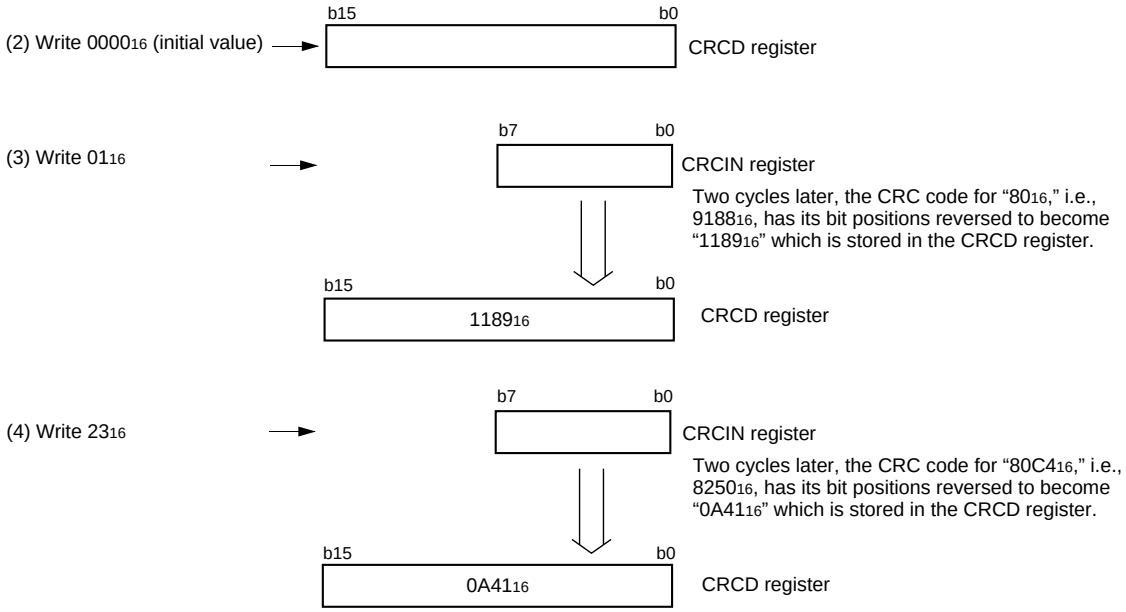
(a) CRC operation performed by the M16C

CRC code: Remainder of a division in which the value written to the CRCIN register with its bit positions reversed is divided by the generator polynomial

Generator polynomial: $X^{16} + X^{12} + X^5 + 1$ (1 0001 0000 0010 0001$_2$)

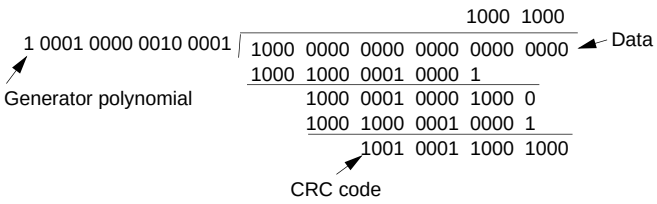
(b) Setting procedure

(1) Reverse the bit positions of the value "80C4$_{16}$" bytewise in a program.
"80$_{16}$" $\rightarrow$ "01$_{16}$", "C4$_{16}$" $\rightarrow$ "23$_{16}$"



(c) Details of CRC operation

In the case of (3) above, the value written to the CRCIN register "01$_{16}$ (00000001$_2$)" has its bit positions reversed to become "10000000$_2$." The value "1000 0000 0000 0000 0000 0000$_2$" derived from that by adding 16 digits and the CRCD register's initial value "0000$_{16}$" are added, the result of which is divided by the generator polynomial using modulo-2 arithmetic.



Modulo-2 operation is operation that complies with the law given below.

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0
-1 = 1

The value "0001 0001 1000 1001$_2$ (1189$_{16}$)" derived from the remainder "1001 0001 1000 1000$_2$ (9188$_{16}$)" by reversing its bit positions may be read from the CRCD register.

If operation (4) above is performed subsequently, the value written to the CRCIN register "23$_{16}$ (00100011$_2$)" has its bit positions reversed to become "11000100$_2$. The value "1100 0100 0000 0000 0000 0000$_2$" derived from that by adding 16 digits and the remainder in (3) "1001 0001 1000 1000$_2$" which is left in the CRCD register are added, the result of which is divided by the generator polynomial using modulo-2 arithmetic.
The value "0000 1010 0100 0001$_2$ (0A41$_{16}$)" derived from the remainder by reversing its bit positions may be read from the CRCD register.

Figure 1.24.3. CRC Calculation

Programmable I/O Ports

The programmable input/output ports (hereafter referred to simply as “I/O ports”) consist of 87 lines P0 to P10 (except $P8_5$) for the 100-pin version, or 113 lines P0 to P14 (except $P8_5$) for the 128-pin version. Each port can be set for input or output every line by using a direction register, and can also be chosen to be or not be pulled high every 4 lines. $P8_5$ is an input-only port and does not have a pull-up resistor. Port $P8_5$ shares the pin with $\overline{NMI}$, so that the $\overline{NMI}$ input level can be read from the P8 register P8_5 bit.

Figures 1.25.1 to 1.25.4 show the I/O ports. Figure 1.25.5 shows the I/O pins.

Each pin functions as an I/O port, a peripheral function input/output, or a bus control pin.

For details on how to set peripheral functions, refer to each functional description in this manual. If any pin is used as a peripheral function input or D-A converter output pin, set the direction bit for that pin to “0” (input mode). Any pin used as an output pin for peripheral functions other than the D-A converter is directed for output no matter how the corresponding direction bit is set.

When using any pin as a bus control pin, refer to “Bus Control.”

P0 to P5, P12, and P13 are capable of V_{CC2}-level input/output; P6 to P11 and P14 are capable of V_{CC1}-level input/output.

(1) Port Pi Direction Register (PDi Register, i = 0 to 13)

Figure 1.25.6 shows the direction registers.

This register selects whether the I/O port is to be used for input or output. The bits in this register correspond one for one to each port.

During memory extension and microprocessor modes, the PDi registers for the pins functioning as bus control pins (A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL/WR}$, $\overline{WRH/BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$, and BCLK) cannot be modified.

No direction register bit for $P8_5$ is available.

(2) Port Pi Register (Pi Register, i = 0 to 13)

Figure 1.25.7 and 1.25.8 show the Pi registers.

Data input/output to and from external devices are accomplished by reading and writing to the Pi register. The Pi register consists of a port latch to hold the input/output data and a circuit to read the pin status. For ports set for input mode, the input level of the pin can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register.

For ports set for output mode, the port latch can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register. The data written to the port latch is output from the pin. The bits in the Pi register correspond one for one to each port.

During memory extension and microprocessor modes, the PDi registers for the pins functioning as bus control pins (A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL/WR}$, $\overline{WRH/BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$, and BCLK) cannot be modified.

(3) Pull-up Control Register 0 to Pull-up Control Register 2 (PUR0 to PUR2 Registers)

Figure 1.25.9 shows the PUR0 to PUR2 registers.

The PUR0 to PUR2 register bits can be used to select whether or not to pull the corresponding port high in 4 bit units. The port chosen to be pulled high has a pull-up resistor connected to it when the direction bit is set for input mode.

However, the pull-up control register has no effect on P0 to P3, $P4_0$ to $P4_3$, and P5 during memory extension and microprocessor modes. Although the register contents can be modified, no pull-up resistors are connected.

(4) Port Control Register

Figure 1.25.10 shows the port control register.

When the P1 register is read after setting the PCR register’s PCR0 bit to “1”, the corresponding port latch can be read no matter how the PD1 register is set.

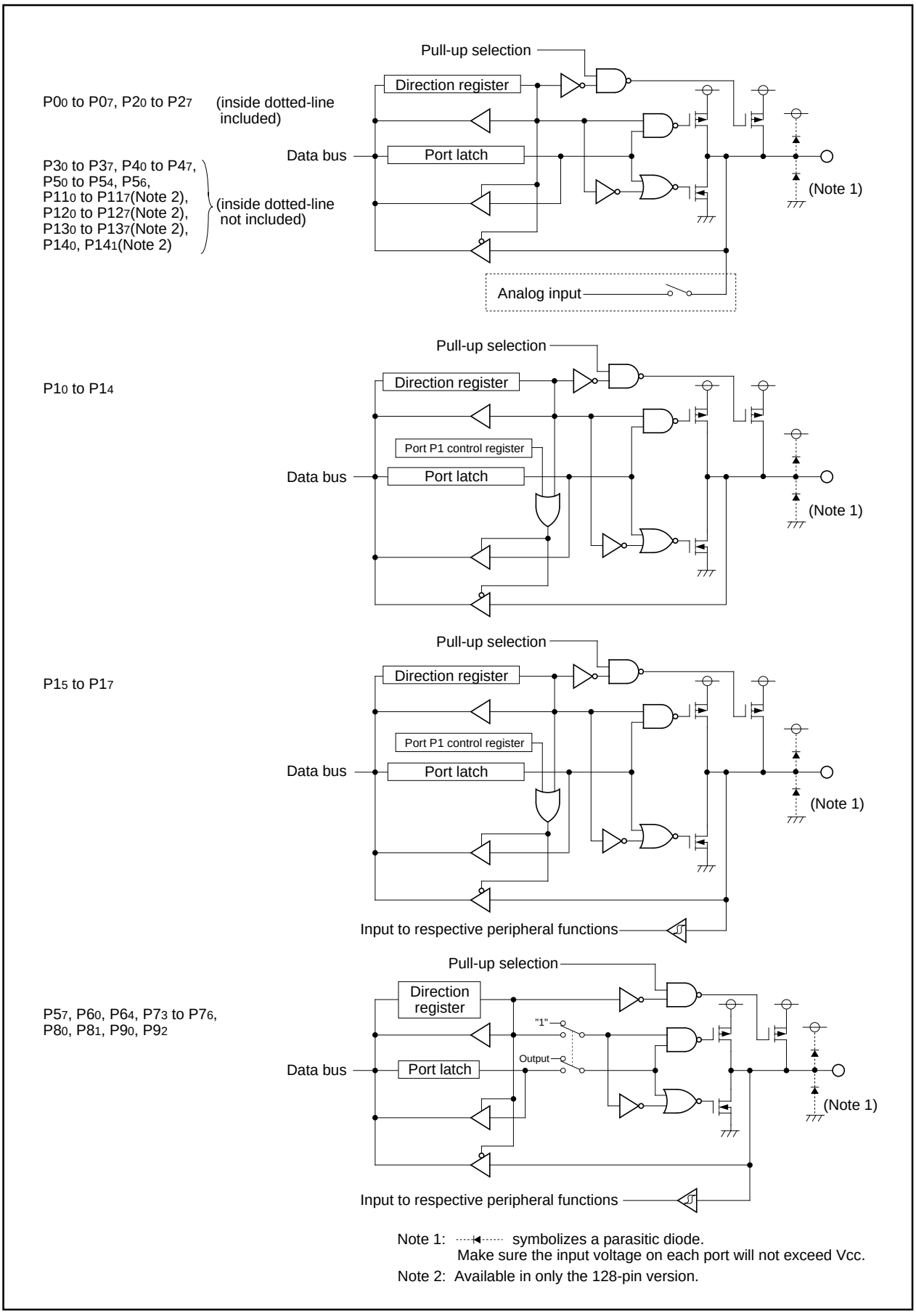
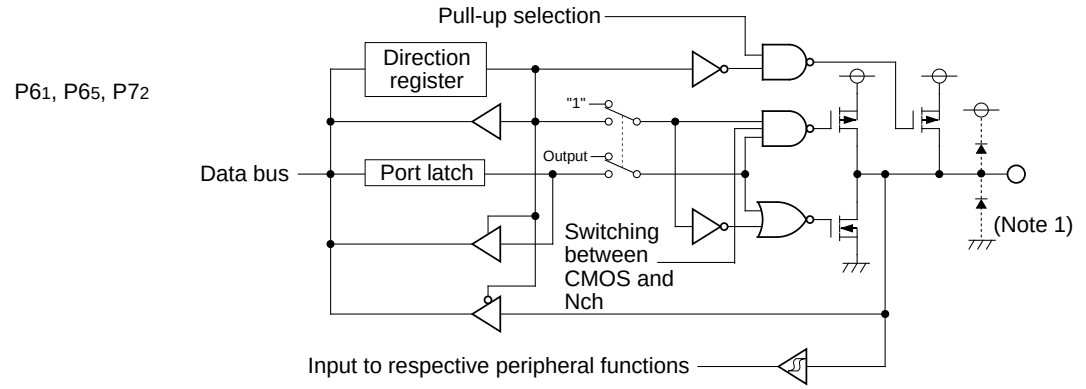
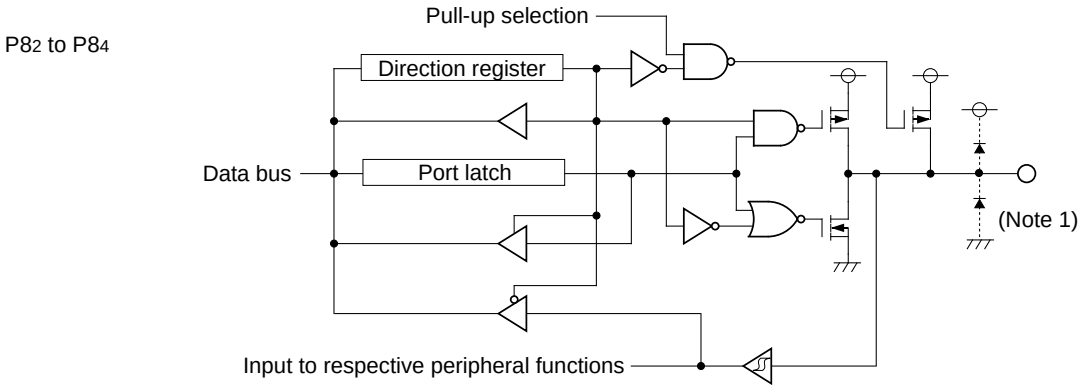
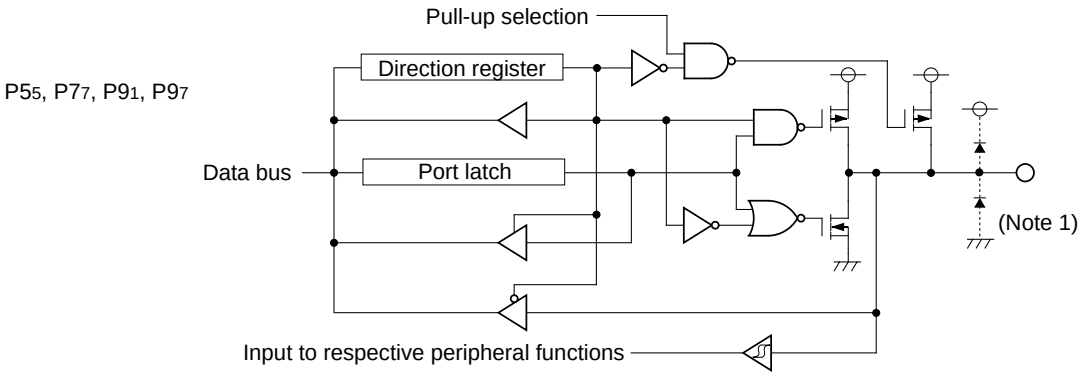


Figure 1.25.1. I/O Ports (1)







Note 1: ⋯◂⋯ symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.
Note 2: Available in only the 128-pin version.

Figure 1.25.2. I/O Ports (2)

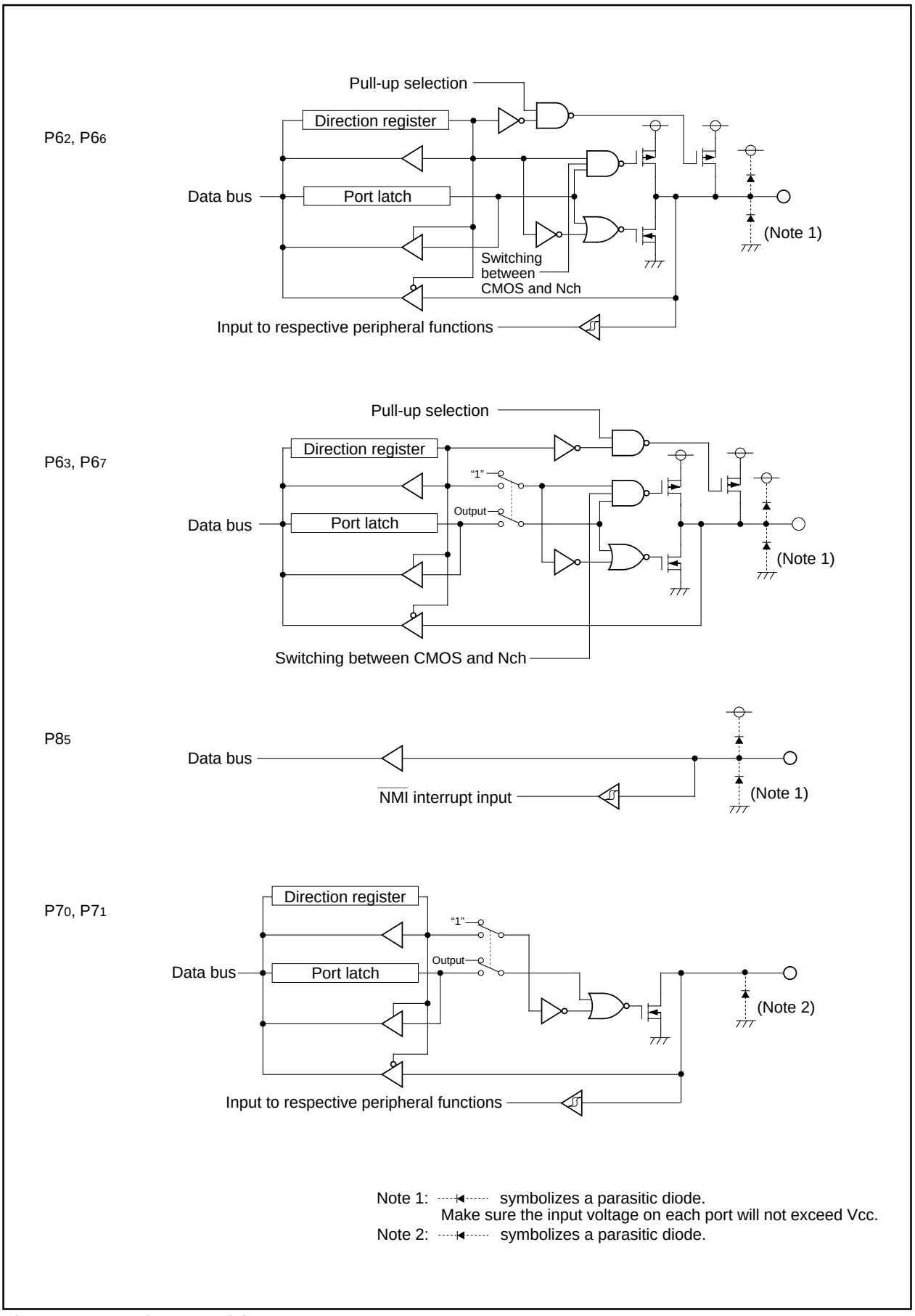


Figure 1.25.3. I/O Ports (3)

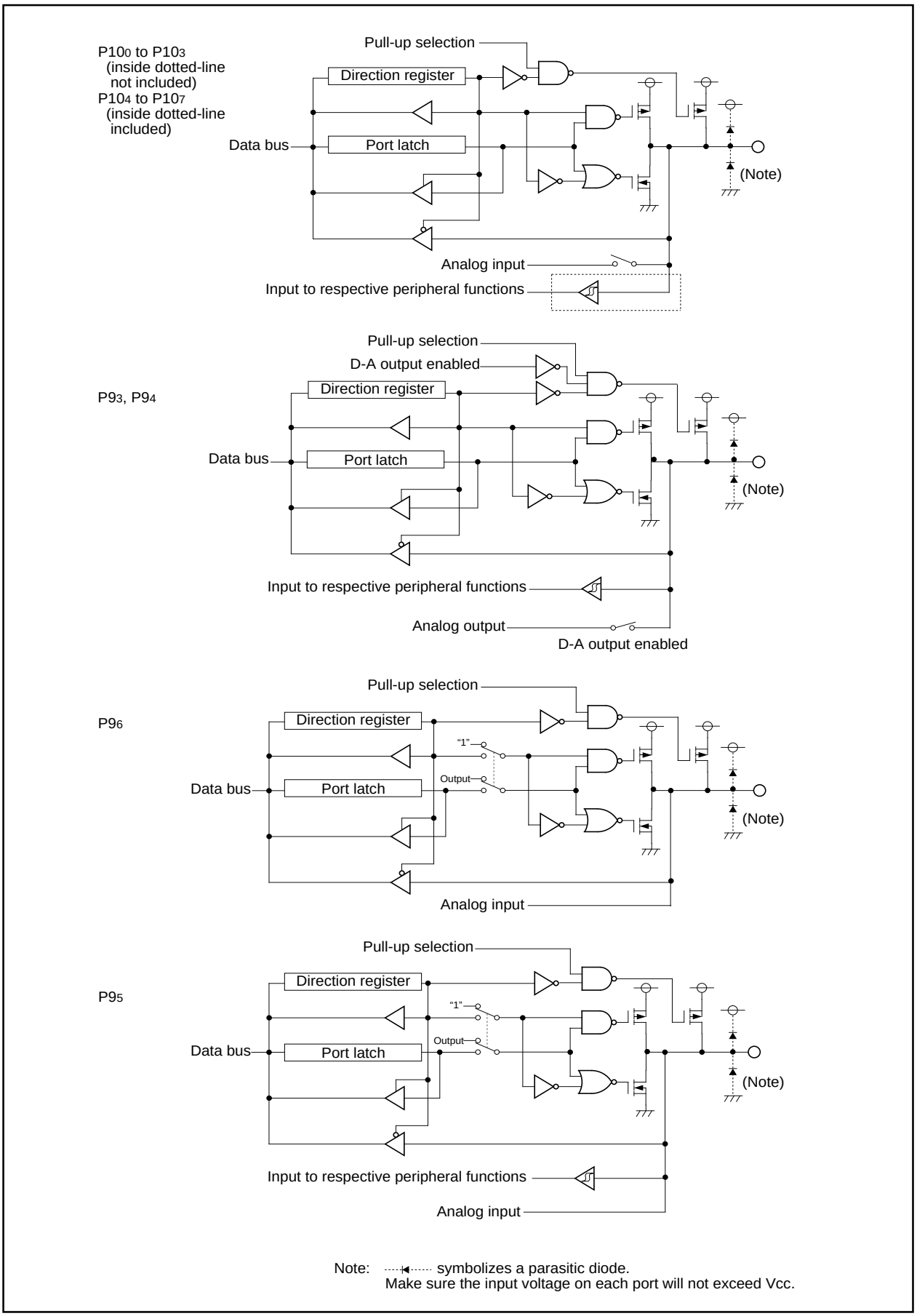


Figure 1.25.4. I/O Ports (4)

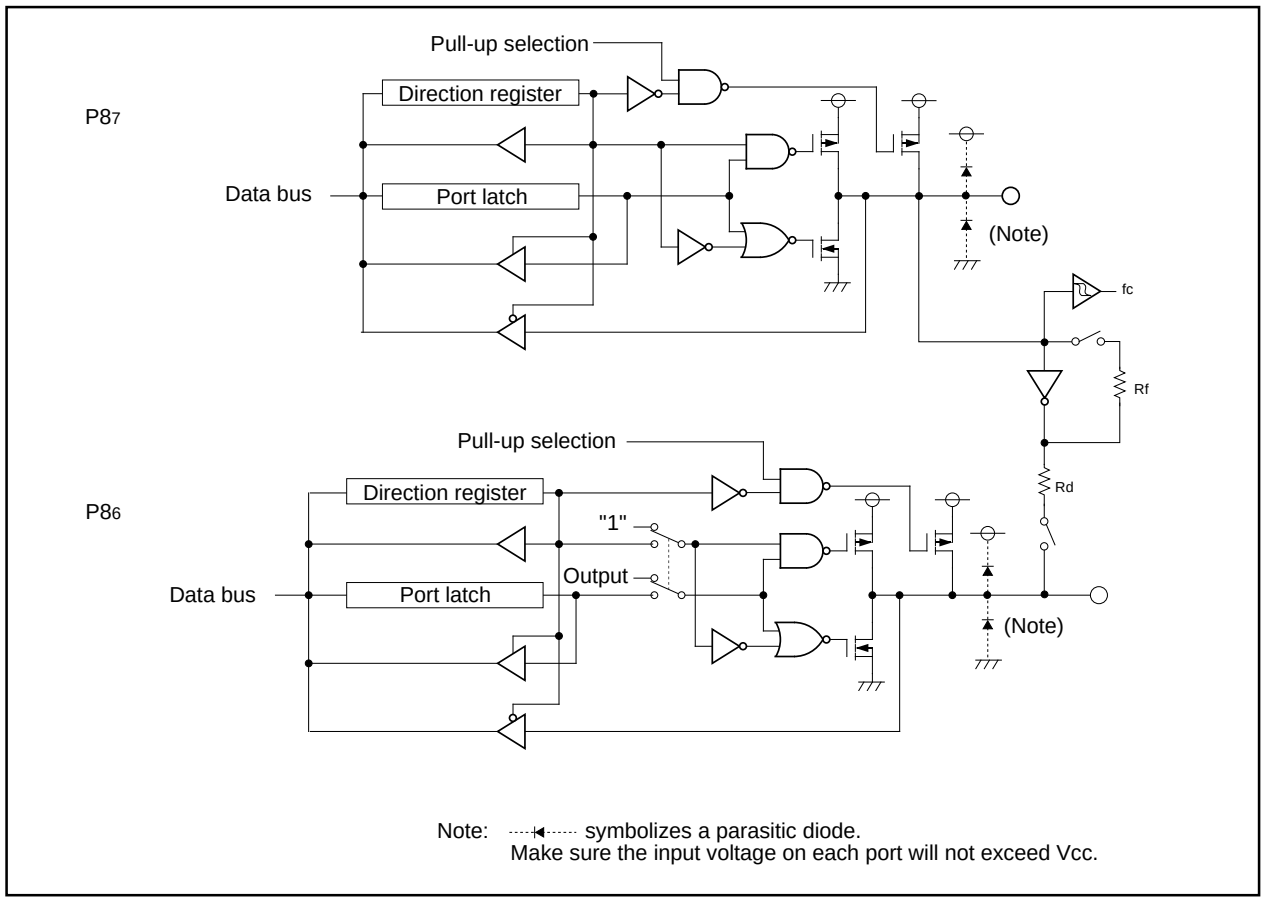


Figure 1.25.5. I/O Ports (5)

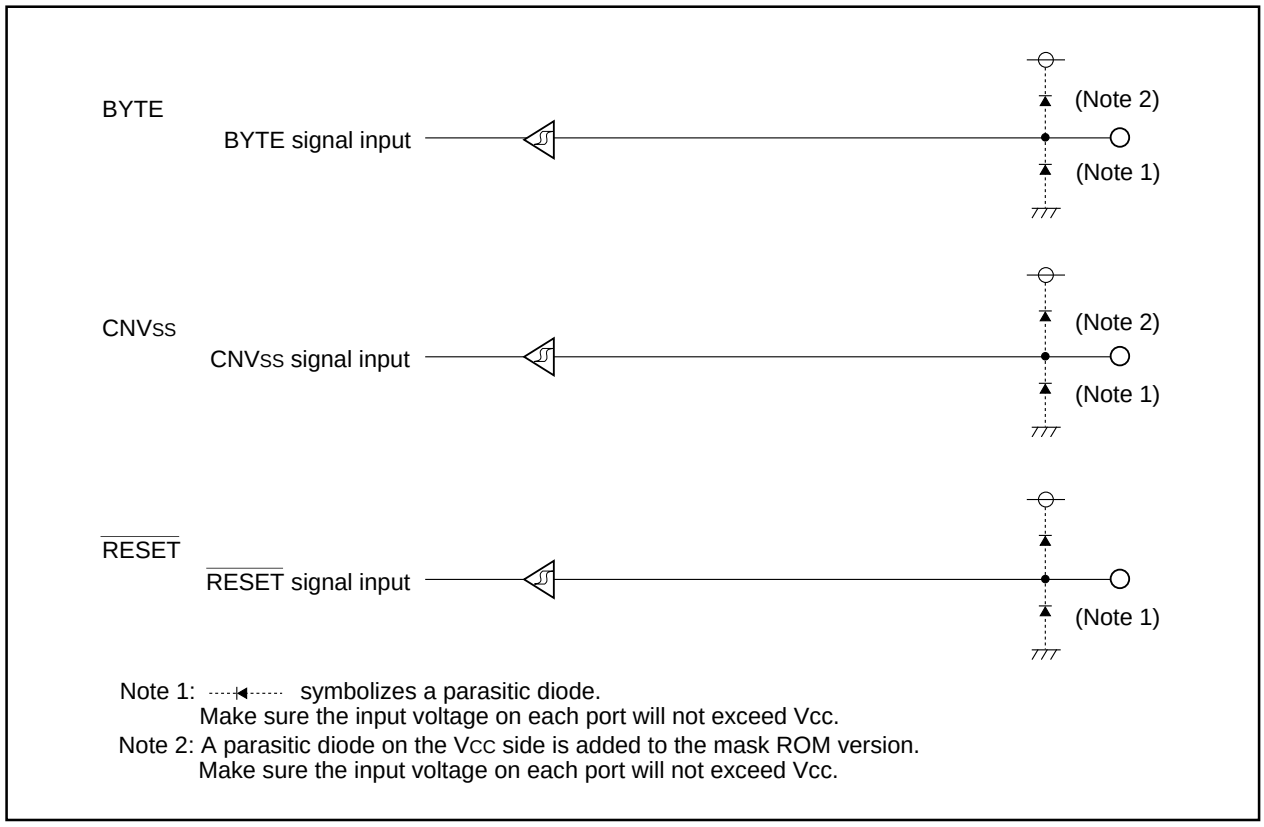
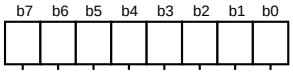


Figure 1.25.6. I/O Pins

Port Pi direction register (i=0 to 7 and 9 to 13) (Note 1, 2, 3)

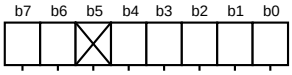


Symbol	Address	After reset
PD0 to PD3	$03E2_{16}$, $03E3_{16}$, $03E6_{16}$, $03E7_{16}$	00_{16}
PD4 to PD7	$03EA_{16}$, $03EB_{16}$, $03EE_{16}$, $03EF_{16}$	00_{16}
PD9 to PD12	$03F3_{16}$, $03F6_{16}$, $03F7_{16}$, $03FA_{16}$	00_{16}
PD13	$03FB_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
PDi_0	Port Pi_0 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 0 to 7 and 9 to 13)	RW
PDi_1	Port Pi_1 direction bit		RW
PDi_2	Port Pi_2 direction bit		RW
PDi_3	Port Pi_3 direction bit		RW
PDi_4	Port Pi_4 direction bit		RW
PDi_5	Port Pi_5 direction bit		RW
PDi_6	Port Pi_6 direction bit		RW
PDi_7	Port Pi_7 direction bit		RW

Note 1: Make sure the PD9 register is written to by the next instruction after setting the PRCR register's PRC2 bit to "1" (write enabled).
Note 2: During memory extension and microprocessor modes, the PD register for the pins functioning as bus control 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.
Note 3: To use ports P11 to P14, set the PUR3 register's PU37 bit to "1" (enable). If this bit is set to "0" (disable), the P11 to P14 pins are placed in the high-impedance state.

Port P8 direction register

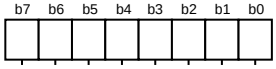


Symbol	Address	After reset
PD8	$03F2_{16}$	$00X00000_2$

Bit symbol	Bit name	Function	RW
PD8_0	Port $P8_0$ direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD8_1	Port $P8_1$ direction bit		RW
PD8_2	Port $P8_2$ direction bit		RW
PD8_3	Port $P8_3$ direction bit		RW
PD8_4	Port $P8_4$ direction bit		RW
— (b5)	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 bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD8_7	Port $P8_7$ direction bit		RW

Figure 1.25.7. PD0 to PD13 Registers

Port Pi register (i=0 to 7 and 9 to 13) (Note 2, 3)

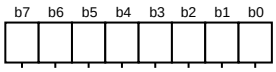


Symbol	Address	After reset
P0 to P3	03E0$_{16}$, 03E1$_{16}$, 03E4$_{16}$, 03E5$_{16}$	Indeterminate
P4 to P7	03E8$_{16}$, 03E9$_{16}$, 03EC$_{16}$, 03ED$_{16}$	Indeterminate
P9 to P12	03F1$_{16}$, 03F4$_{16}$, 03F5$_{16}$, 03F8$_{16}$	Indeterminate
P13	03F9$_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
Pi_0	Port Pi$_0$ bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0 : “L” level 1 : “H” level (Note 1) (i = 0 to 7 and 9 to 13)	RW
Pi_1	Port Pi$_1$ bit		RW
Pi_2	Port Pi$_2$ bit		RW
Pi_3	Port Pi$_3$ bit		RW
Pi_4	Port Pi$_4$ bit		RW
Pi_5	Port Pi$_5$ bit		RW
Pi_6	Port Pi$_6$ bit		RW
Pi_7	Port Pi$_7$ bit		RW

Note 1: Since P7$_0$ and P7$_1$ are N-channel open drain ports, the data is high-impedance.
Note 2: During memory extension and microprocessor modes, the Pi register for the pins functioning as bus control 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.
Note 3: To use ports P11 to P14, set the PUR3 register’s PU37 bit to “1” (enable). If this bit is set to “0” (disable), the P11 to P14 registers are cleared to ‘00$_{16}$’ and the P11 to P14 pins are placed in the high-impedance state.

Port P8 register

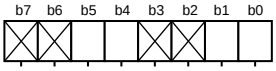


Symbol	Address	After reset
P8	03F0$_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
P8_0	Port P8$_0$ bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register (except for P8$_5$) 0 : “L” level 1 : “H” level	RW
P8_1	Port P8$_1$ bit		RW
P8_2	Port P8$_2$ bit		RW
P8_3	Port P8$_3$ bit		RW
P8_4	Port P8$_4$ bit		RW
P8_5	Port P8$_5$ bit		RO
P8_6	Port P8$_6$ bit		RW
P8_7	Port P8$_7$ bit		RW

Figure 1.25.8. P0 to P13 Registers

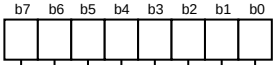
Port P14 control register (128-pin package)



Symbol	Address	After reset
PC14	$03DE_{16}$	$XX00XXXX_2$

Bit symbol	Bit name	Function	RW
P140	Port $P14_0$ bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0 : "L" level 1 : "H" level	RW
P141	Port $P14_1$ bit		RW
— (b3-b2)	Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		—
PD140	Port $P14_0$ direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD141	Port $P14_1$ direction bit		RW
— (b7-b6)	Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		—

Pull-up control register 3 (128-pin package)



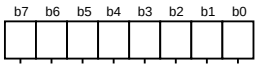
Symbol	Address	After reset
PUR3	$03DF_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
PU30	$P11_0$ to $P11_3$ pull-up	0 : Not pulled high 1 : Pulled high (Note 1)	RW
PU31	$P11_4$ to $P11_7$ pull-up		RW
PU32	$P12_0$ to $P12_3$ pull-up		RW
PU33	$P12_4$ to $P12_7$ pull-up		RW
PU34	$P13_0$ to $P13_3$ pull-up		RW
PU35	$P13_4$ to $P13_7$ pull-up		RW
PU36	$P14_0$, $P14_1$ pull-up		RW
PU37	P11 to P14 enabling bit	0 : Unusable (Note 2) 1 : Usable	RW

Note 1: The pin for which this bit is "1" (pulled high) and the direction bit is "0" (input mode) is pulled high.
Note 2: If the PU37 bit is set to "0" (unusable), the P11 to P14 pins are placed in the high-impedance state and the P11 to P14 registers are cleared to "0".

Figure 1.25.9. PC14 Register and PUR3 Register

Pull-up control register 0 (Note 1)

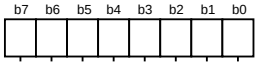


Symbol	Address	After reset
PUR0	$03FC_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
PU00	$P0_0$ to $P0_3$ pull-up	0 : Not pulled high 1 : Pulled high (Note 2)	RW
PU01	$P0_4$ to $P0_7$ pull-up		RW
PU02	$P1_0$ to $P1_3$ pull-up		RW
PU03	$P1_4$ to $P1_7$ pull-up		RW
PU04	$P2_0$ to $P2_3$ pull-up		RW
PU05	$P2_4$ to $P2_7$ pull-up		RW
PU06	$P3_0$ to $P3_3$ pull-up		RW
PU07	$P3_4$ to $P3_7$ pull-up		RW

Note 1: During memory extension and microprocessor modes, the pins are not pulled high although their corresponding register contents can be modified.
Note 2: The pin for which this bit is “1” (pulled high) and the direction bit is “0” (input mode) is pulled high.

Pull-up control register 1



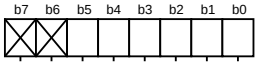
Symbol	Address	After reset(Note 5)
PUR1	$03FD_{16}$	00000000_2 00000010_2

Bit symbol	Bit name	Function	RW
PU10	$P4_0$ to $P4_3$ pull-up (Note 2)	0 : Not pulled high 1 : Pulled high (Note 3)	RW
PU11	$P4_4$ to $P4_7$ pull-up (Note 4)		RW
PU12	$P5_0$ to $P5_3$ pull-up (Note 2)		RW
PU13	$P5_4$ to $P5_7$ pull-up (Note 2)		RW
PU14	$P6_0$ to $P6_3$ pull-up		RW
PU15	$P6_4$ to $P6_7$ pull-up		RW
PU16	$P7_2$ to $P7_3$ pull-up (Note 1)		RW
PU17	$P7_4$ to $P7_7$ pull-up		RW

Note 1: The $P7_0$ and $P7_1$ pins do not have pull-ups.
Note 2: During memory extension and microprocessor modes, the pins are not pulled high although the contents of these bits can be modified.
Note 3: The pin for which this bit is “1” (pulled high) and the direction bit is “0” (input mode) is pulled high.
Note 4: If the PM01 to PM00 bits are set to “01_2” (memory expansion mode) or “11_2” (microprocessor mode) in a program during single-chip mode, the PU11 bit becomes “1”.
Note 5: The values after hardware reset 1 and 2 are as follows:
- 00000000_2 when input on CNVss pin is “L“
- 00000010_2 when input on CNVss pin is “H“

The values after software reset, watchdog timer reset and oscillation stop detection reset are as follows:
- 00000000_2 when PM 01 to PM00 bits of PM0 register are “00_2“ (single-chip mode)
- 00000010_2 when PM 01 to PM00 bits of PM0 register are “01_2“ (memory expansion mode) or “11_2“ (microprocessor mode)

Pull-up control register 2



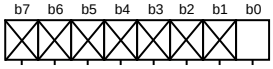
Symbol	Address	After reset
PUR2	$03FE_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
PU20	$P8_0$ to $P8_3$ pull-up	0 : Not pulled high 1 : Pulled high (Note 1)	RW
PU21	$P8_4$ to $P8_7$ pull-up (Note 2)		RW
PU22	$P9_0$ to $P9_3$ pull-up		RW
PU23	$P9_4$ to $P9_7$ pull-up		RW
PU24	$P10_0$ to $P10_3$ pull-up		RW
PU25	$P10_4$ to $P10_7$ pull-up		RW
— (b7-b6)	Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.		—

Note 1: The pin for which this bit is “1” (pulled high) and the direction bit is “0” (input mode) is pulled high.
Note 2: The $P8_5$ pin does not have pull-up.

Figure 1.25.10. PUR0 to PUR2 Registers

Port control register



Symbpl: PCR
Address: $03FF_{16}$
After reset: 00_{16}

Bit symbol	Bit name	Function	RW
PCR0	Port P1 control bit	Operation performed when the P1 register is read 0: When the port is set for input, the input levels of P10 to P17 pins are read. When set for output, the port latch is read. 1: The port latch is read regardless of whether the port is set for input or output.	RW
— (b7-b1)	Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.		—

Figure 1.25.11. PCR Register

Table 1.25.1. Unassigned Pin Handling in Single-chip Mode

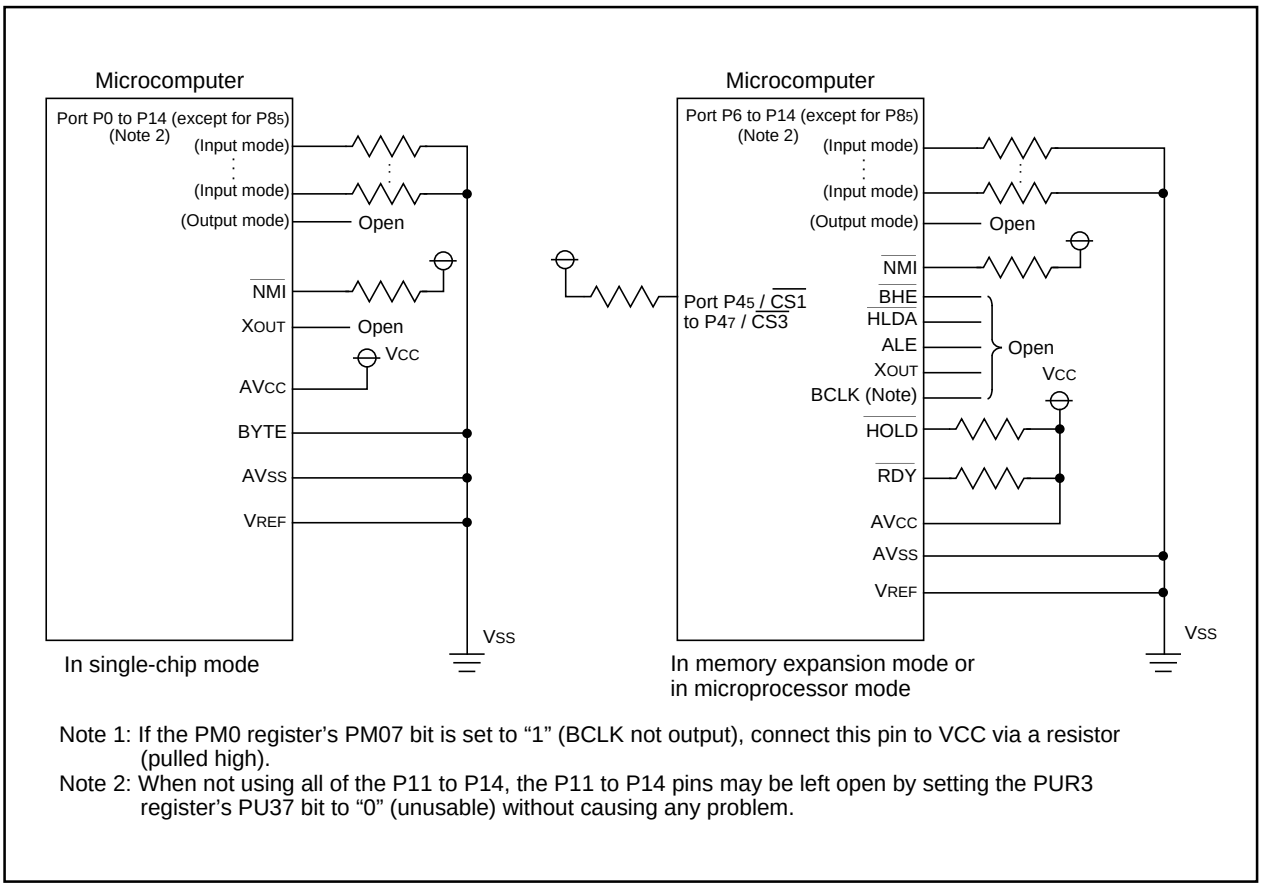
Pin name	Connection
Ports P0 to P14 (excluding P85)	After setting for input mode, connect every pin to VSS via a resistor(pull-down); or after setting for output mode, leave these pins open. (Note 2)
XOUT (Note 1)	Open
$\overline{\text{NMI}}$	Connect via resistor to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF, BYTE	Connect to VSS

Note 1: With external clock input to XIN pin.
Note 2: When not using all of the P11 to P14, the P11 to P14 pins may be left open by setting the PUR3 register's PU37 bit to "0" (unusable) without causing any problem.

Table 1.25.2. Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode

Pin name	Connection
Ports P6 to P10 (excluding P85)	After setting for input mode, connect every pin to VSS via a resistor (pull-down); or after setting for output mode, leave these pins open.
P45 / $\overline{\text{CS1}}$ to P47 / $\overline{\text{CS3}}$	Connect to VCC via a resistor (pulled high) by setting the PD4 register's corresponding direction bit for $\overline{\text{CSi}}$ (i=1 to 3) to "0" (input mode) and the CSR register's CSi bit to "0" (chip select disabled).
$\overline{\text{BHE}}$, ALE, $\overline{\text{HLDA}}$, XOUT (Note 1), BCLK (Note 2)	Open
$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$, $\overline{\text{NMI}}$	Connect via resistor to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF	Connect to VSS

Note 1: With external clock input to XIN pin.
Note 2: If the PM0 register's PM07 bit is set to "1" (BCLK not output), connect this pin to VCC via a resistor (pulled high).
Note 3: When not using all of the P11 to P14, the P11 to P14 pins may be left open by setting the PUR3 register's PU37 bit to "0" (unusable) without causing any problem.



Note 1: If the PM0 register's PM07 bit is set to "1" (BCLK not output), connect this pin to VCC via a resistor (pulled high).
Note 2: When not using all of the P11 to P14, the P11 to P14 pins may be left open by setting the PUR3 register's PU37 bit to "0" (unusable) without causing any problem.

Figure 1.25.12. Unassigned Pins Handling

Electrical Characteristics

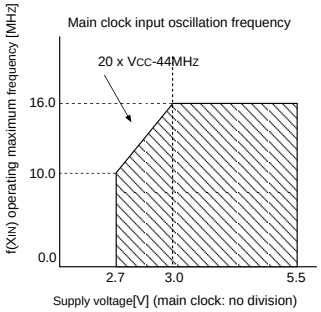
Table 1.26.1. Absolute Maximum Ratings

Symbol	Parameter		Condition	Rated value	Unit
V_{CC1}, V_{CC2}	Supply voltage		V_{CC1}=AVCC	-0.3 to 6.5	V
V_{CC2}	Supply voltage		V_{CC2}	-0.3 to V_{CC1}+0.1	V
AVCC	Analog supply voltage		V_{CC1}=AVCC	-0.3 to 6.5	V
V_I	Input voltage	$\overline{RESET}$, CNV_{SS}, BYTE, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, $P11_0$ to $P11_7$, $P14_0$, $P14_1$, V_{REF}, X_{IN}		-0.3 to V_{CC1}+0.3	V
		$P0_0$ 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 $P5_7$, $P12_0$ to $P12_7$, $P13_0$ to $P13_7$		-0.3 to V_{CC2}+0.3	V
		$P7_0$, $P7_1$		-0.3 to 6.5	V
V_O	Output voltage	$P6_0$ 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 $P10_7$, $P11_0$ to $P11_7$, $P14_0$, $P14_1$, X_{OUT}		-0.3 to V_{CC1}+0.3	V
		$P0_0$ 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 $P5_7$, $P12_0$ to $P12_7$, $P13_0$ to $P13_7$		-0.3 to V_{CC2}+0.3	V
		$P7_0$, $P7_1$		-0.3 to 6.5	V
P_d	Power dissipation		Topr=25 °C	300	mW
T_{opr}	Operating ambient temperature			-20 to 85 / -40 to 85	°C
T_{stg}	Storage temperature			-65 to 150	°C

Table 1.26.2. Recommended Operating Conditions (Note 1)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V_{CC1}, V_{CC2}	Supply voltage($V_{CC1} \geq V_{CC2}$)		2.7	5.0	5.5	V
AVcc	Analog supply voltage			V_{CC1}		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 $P5_7$, $P12_0$ to $P12_7$, $P13_0$ to $P13_7$	0.8V_{CC2}		V_{CC2}	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (during single-chip mode)	0.8V_{CC2}		V_{CC2}	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)	0.5V_{CC2}		V_{CC2}	V
		$P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, $P11_0$ to $P11_7$, $P14_0$, $P14_1$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	0.8V_{CC1}		V_{CC1}	V
		$P7_0$, $P7_1$	0.8V_{CC1}		6.5	V
V_{IL}	LOW input voltage	$P3_1$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P12_0$ to $P12_7$, $P13_0$ to $P13_7$	0		0.2V_{CC2}	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (during single-chip mode)	0		0.2V_{CC2}	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		0.16V_{CC2}	V
		$P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, $P11_0$ to $P11_7$, $P14_0$, $P14_1$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	0		0.2V_{CC1}	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 $P10_7$, $P11_0$ to $P11_7$, $P12_0$ to $P12_7$, $P13_0$ to $P13_7$, $P14_0$, $P14_1$			-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 $P10_7$, $P11_0$ to $P11_7$, $P12_0$ to $P12_7$, $P13_0$ to $P13_7$, $P14_0$, $P14_1$			-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 $P10_7$, $P11_0$ to $P11_7$, $P12_0$ to $P12_7$, $P13_0$ to $P13_7$, $P14_0$, $P14_1$			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 $P10_7$, $P11_0$ to $P11_7$, $P12_0$ to $P12_7$, $P13_0$ to $P13_7$, $P14_0$, $P14_1$			5.0	mA
f (X_{IN})	Main clock input oscillation frequency (Note 4)	V_{CC}=3.0 to 5.5V	0		16	MHz
		V_{CC}=2.7 to 3.0V	0		20 X V_{CC}-44	MHz
f (X_{CIN})	Sub-clock oscillation frequency			32.768	50	kHz
f (Ring)	Ring oscillation frequency			1		MHz
f (PLL)	PLL clock oscillation frequency (Note 4)	V_{CC}=3.0 to 5.5V	10		24	MHz
		V_{CC}=2.7 to 3.0V	10		46.67 X V_{CC}-116	MHz
f (BCLK)	CPU operation clock		0		24	MHz
T_{SU}(PLL)	PLL frequency synthesizer stabilization wait time	V_{CC}=5.0V			20	ms
		V_{CC}=3.0V			50	ms

Note 1: Referenced to Vcc = V_{CC1} = V_{CC2} = 2.7 to 5.5V at Topr = -20 to 85 °C / -40 to 85 °C unless otherwise specified.
Note 2: The mean output current is the mean value within 100ms.
Note 3: The total I_{OL} (peak) for ports P0, P1, P2, $P8_6$, $P8_7$, P9, P10, P11, $P14_0$ and $P14_1$ must be 80mA max. The total I_{OL} (peak) for ports P3, P4, P5, P6, P7, $P8_0$ to $P8_4$, P12, and P13 must be 80mA max. The total I_{OH} (peak) for ports P0, P1, and P2 must be -40mA max. The total I_{OH} (peak) for ports P3, P4, P5, P12, and P13 must be -40mA max. The total I_{OH} (peak) for ports P6, P7, and $P8_0$ to $P8_4$ must be -40mA max. The total I_{OH} (peak) for ports $P8_6$, $P8_7$, P9, P10, P11, $P14_0$, and $P14_1$ must be -40mA max.
Note 4: Relationship between main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.



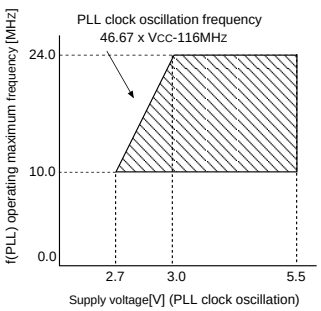


Table 1.26.3. A-D Conversion Characteristics (Note 1)

Symbol	Parameter		Measuring condition		Standard Min.	Standard Typ.	Standard Max.	Unit
–	Resolution		V_{REF} =V_{CC1}				10	Bits
INL	Integral non-linearity error	10 bit	V_{REF}= V_{CC1}= 5V	AN_0 to AN_7 input			±3	LSB
				ANEX0, ANEX1 input External operation amp connection mode AN_{00} to AN_{07} input AN_{20} to AN_{27} input			±7	LSB
			V_{REF}= V_{CC1}= 3.3V	AN_0 to AN_7 input			±5	LSB
				ANEX0, ANEX1 input External operation amp connection mode AN_{00} to AN_{07} input AN_{20} to AN_{27} input			±7	LSB
		8 bit	V_{REF} =V_{CC1}=3.3V				±2	LSB
DNL	Differential non-linearity error						±1	LSB
–	Offset error						±3	LSB
–	Gain error						±3	LSB
R_{LADDER}	Ladder resistance		V_{REF} =V_{CC1}		10		40	kΩ
t_{CONV}	Conversion time(10bit), Sample & hold function available		V_{REF} =V_{CC1}=5V, $ø_{AD}$=10MHz		3.3			µs
t_{CONV}	Conversion time(8bit), Sample & hold function available		V_{REF} =V_{CC1}=5V, $ø_{AD}$=10MHz		2.8			µs
t_{SAMP}	Sampling time				0.3			µs
V_{REF}	Reference voltage				2.0		V_{CC1}	V
V_{IA}	Analog input voltage				0		V_{REF}	V

Note 1: Referenced to V_{CC1}=AV_{CC}=V_{REF}=3.3 to 5.5V, V_{SS}=AV_{SS}=0V at Topr = -20 to 85 °C / -40 to 85 °C unless otherwise specified.
Note 2: If V_{CC1} > V_{CC2}, do not use AN_{00} to AN_{07} and AN_{20} to AN_{27} as analog input pins.
Note 3: AD operation clock frequency ($Ø_{AD}$ frequency) must be 10 MHz or less. And divide the f_{AD} if V_{CC1} is less than 4.2V, and make $Ø_{AD}$ frequency equal to or lower than f_{AD}/2.
Note 4: A case without sample & hold function turn $Ø_{AD}$ frequency into 250 kHz or more in addition to a limit of Note 3.
A case with sample & hold function turn $Ø_{AD}$ frequency into 1MHz or more in addition to a limit of Note 3.

Table 1.26.4. D-A Conversion Characteristics (Note 1)

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

Note 1: Referenced to V_{CC1}=V_{REF}=3.3 to 5.5V, V_{SS}=AV_{SS}=0V at Topr = -20 to 85 °C / -40 to 85 °C unless otherwise specified.
Note 2: This applies when using one D-A converter, with the D-A register for the unused D-A converter set to "00_{16}". The A-D converter's ladder resistance is not included. Also, when D-A register contents are not "00_{16}", the current I_{VREF} always flows even though Vref may have been set to be unconnected by the A-D control register.

Table 1.26.5. Flash Memory Version Electrical Characteristics (Note 1)

Parameter	Standard Min.	Standard Typ.	Standard Max	Unit
Word program time		30	200	µs
Block erase time		1	4	s
Erase all unlocked blocks time		1 X n	4 X n	s
Lock bit program time		30	200	µs

Note 1: Referenced to V_{CC1}=4.5 to 5.5V, 3.0 to 3.6V at Topr = 0 to 60 °C unless otherwise specified.
Note 2: n denotes the number of block erases.

Table 1.26.6. Flash Memory Version Program/Erase Voltage and Read Operation Voltage Characteristics (at Topr = 0 to 60°C)

Flash program, erase voltage	Flash read operation voltage
V_{CC1} = 3.3 V ± 0.3 V or 5.0 V ± 0.5 V	V_{CC1}=2.7 to 5.5 V

Table 1.26.7. Low Voltage Detection Circuit Electrical Characteristics (Note 1)

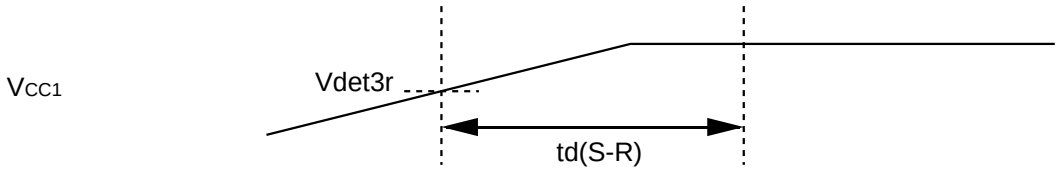
Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
Vdet4	Power supply down detection voltage (Notes 1, 2)	V_{CC1}=0.8 to 5.5V	3.3	3.8	4.4	V
Vdet3	Reset level detection voltage (Notes 1, 2)		2.2	2.8	3.6	V
Vdet3s	Low voltage reset retention voltage		0.8			V
Vdet3r	Low voltage reset release voltage (Note 3)		2.2	2.9	4.0	V
Vdet2	RAM retention limit detection voltage (Notes 1, 2)		1.4	2.0	2.7	V

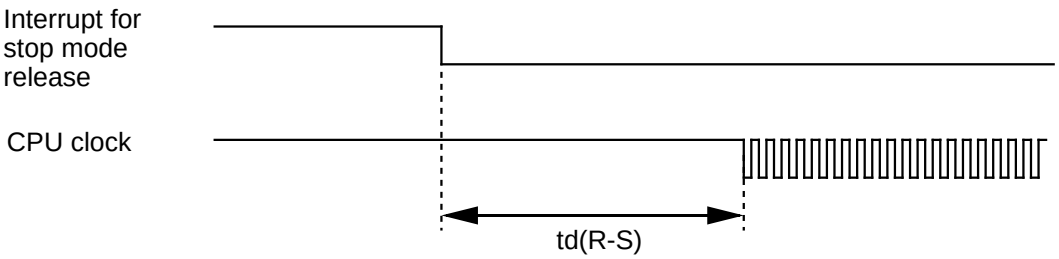
Note 1: Vdet4 > Vdet3 > Vdet2
Note 2: Where reset level detection voltage is less than 2.7 V, if the supply power voltage is greater than the reset level detection voltage, the operation at f(BCLK) ≤ 10MHz is guaranteed.
Note 3: Vdet3r > Vdet3 is not guaranteed.

Table 1.26.8. Power Supply Circuit Timing Characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Time for internal power supply stabilization during powering-on	V_{CC1}=2.7 to 5.5V			2	ms
td(R-S)	STOP release time				150	µs
td(M-L)	Time for internal power supply stabilization when main clock oscillation starts				50	µs
td(S-R)	Hardware reset 2 release wait time	V_{CC1}=Vdet3r to 5.5V		6 (Note)	20	ms
td(E-A)	Low voltage detection circuit operation start time	V_{CC1}=2.7 to 5.5V			20	µs

Note : When Vcc1 = 5V





Electrical Characteristics (V_{CC1} = V_{CC2} = 5V)

V_{CC1} = V_{CC2} = 5V

Table 1.26.9. Electrical Characteristics (Note 1)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	$P6_0$ 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 $P10_7$,$P11_0$ to $P11_7$,$P14_0$,$P14_1$	I_{OH}=-5mA	V_{CC1}-2.0		V_{CC1}	V
		$P0_0$ 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 $P5_7$,$P12_0$ to $P12_7$,$P13_0$ to $P13_7$	I_{OH}=-5mA(Note 2)	V_{CC2}-2.0		V_{CC2}	
V_{OH}	HIGH output voltage	$P6_0$ 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 $P10_7$,$P11_0$ to $P11_7$,$P14_0$,$P14_1$	I_{OH}=-200µA	V_{CC1}-0.3		V_{CC1}	V
		$P0_0$ 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 $P5_7$,$P12_0$ to $P12_7$,$P13_0$ to $P13_7$	I_{OH}=-200µA(Note 2)	V_{CC2}-0.3		V_{CC2}	
V_{OH}	HIGH output voltage	X_{OUT} HIGHPOWER	I_{OH}=-1mA	V_{CC1}-2.0		V_{CC1}	V
		X_{OUT} LOWPOWER	I_{OH}=-0.5mA	V_{CC1}-2.0		V_{CC1}	
	HIGH output voltage	X_{COUT} HIGHPOWER	With no load applied		2.5		V
		X_{COUT} LOWPOWER	With no load applied		1.6		
V_{OL}	LOW output voltage	$P6_0$ 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 $P10_7$,$P11_0$ to $P11_7$,$P14_0$,$P14_1$	I_{OL}=5mA			2.0	V
		$P0_0$ 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 $P5_7$,$P12_0$ to $P12_7$,$P13_0$ to $P13_7$	I_{OL}=5mA(Note 2)			2.0	
V_{OL}	LOW output voltage	$P6_0$ 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 $P10_7$,$P11_0$ to $P11_7$,$P14_0$,$P14_1$	I_{OL}=200µA			0.45	V
		$P0_0$ 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 $P5_7$,$P12_0$ to $P12_7$,$P13_0$ to $P13_7$	I_{OL}=200µA(Note 2)			0.45	
V_{OL}	LOW output voltage	X_{OUT} HIGHPOWER	I_{OL}=1mA			2.0	V
		X_{OUT} LOWPOWER	I_{OL}=0.5mA			2.0	
	LOW output voltage	X_{COUT} HIGHPOWER	With no load applied		0		V
		X_{COUT} LOWPOWER	With no load applied		0		
V_{T+}-V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$ to $TA4_{IN}$, $TB0_{IN}$ to $TB5_{IN}$, $\overline{INT0}$ to $\overline{INT5}$, $\overline{NMI}$, $\overline{AD_{TRG}}$, $\overline{CTS0}$ to $\overline{CTS2}$, SCL, SDA, $CLK0$ to $CLK4$,$TA2_{OUT}$ to $TA4_{OUT}$, $\overline{KI0}$ to $\overline{KI3}$, $RxD0$ to $RxD2$, S_{IN3}, S_{IN4}		0.2		1.0	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$		0.2		2.2	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 $P10_7$,$P11_0$ to $P11_7$, $P12_0$ to $P12_7$,$P13_0$ to $P13_7$,$P14_0$,$P14_1$, X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I=5V			5.0	µA
I_{IL}	LOW input current	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_7$,$P3_0$ 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 $P10_7$,$P11_0$ to $P11_7$, $P12_0$ to $P12_7$,$P13_0$ to $P13_7$,$P14_0$,$P14_1$, X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I=0V			-5.0	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 $P10_7$, $P11_0$ to $P11_7$,$P12_0$ to $P12_7$,$P13_0$ to $P13_7$,$P14_0$,$P14_1$	V_I=0V	30	50	170	kΩ
R_{fXIN}	Feedback resistance	X_{IN}			1.5		MΩ
R_{fXCIN}	Feedback resistance	X_{CIN}			15		MΩ
V_{RAM}	RAM retention voltage		At stop mode	2.0			V

Note 1: Referenced to V_{CC}=V_{CC1}=V_{CC2}=4.2 to 5.5V, V_{SS}=0V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK)=24MHz unless otherwise specified.
Note 2: Where the product is used at V_{CC1} = 5 V and V_{CC2} = 3 V, refer to the 3 V version value for the pin specified value on the V_{CC2} port side.

V_{CC1} = V_{CC2} = 5V

Table 1.26.10. Electrical Characteristics (2) (Note 1)

Symbol	Parameter		Measuring condition		Standard			Unit
					Min.	Typ.	Max.	
I_{CC}	Power supply current (V_{CC}=4.0 to 5.5V)	In single-chip mode, the output pins are open and other pins are V_{SS}	Mask ROM	f(BCLK)=24MHz, No division, PLL operation		14	20	mA
				No division, Ring oscillation		1		mA
			Flash memory	f(BCLK)=24MHz, No division, PLL operation		18	27	mA
				No division, Ring oscillation		1.8		mA
			Flash memory Program	f(BCLK)=10MHz, V_{CC}=5.0V		15		mA
			Flash memory Erase	f(BCLK)=10MHz, V_{CC}=5.0V		25		mA
			Mask ROM	f(X_{CIN})=32kHz, Low power dissipation mode, ROM(Note 3)		25		µA
			Flash memory	f(BCLK)=32kHz, Low power dissipation mode, RAM(Note 3)		25		µA
				f(BCLK)=32kHz Low power dissipation mode, Flash memory(Note 3)		420		µA
				Ring oscillation, Wait mode		50		µA
			Mask ROM Flash memory	f(BCLK)=32kHz, Wait mode (Note 2), Oscillation capacity High		7.5		µA
				f(BCLK)=32kHz, Wait mode(Note 2), Oscillation capacity Low		2.0		µA
				Stop mode, T_{opr}=25°C		0.8	3.0	µA
Idet4	Power supply down detection dissipation current (Note 4)					0.7	4	µA
Idet3	Reset area detection dissipation current (Note 4)					1.2	8	µA
Idet2	RAM retention limit detection dissipation current (Note 4)					1.1	6	µA

Note 1: Referenced to V_{CC}=V_{CC1}=V_{CC2}=4.2 to 5.5V, V_{SS}=0V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK)=24MHz unless otherwise specified.
Note 2: With one timer operated using f_{C32}.
Note 3: This indicates the memory in which the program to be executed exists.
Note 4: Idet is dissipation current when the following bit is set to "1" (detection circuit enabled).
Idet4: VC27 bit of VCR2 register
Idet3: VC26 bit of VCR2 register
Idet2: VC25 bit of VCR2 register

$V_{CC1} = V_{CC2} = 5V$

Timing Requirements

($V_{CC1} = V_{CC2} = 5V$, $V_{SS} = 0V$, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 1.26.11. 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 1.26.12. Memory Expansion Mode and Microprocessor Mode

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{ac1(RD-DB)}$	Data input access time (for setting with no wait)		(Note 1)	ns
$t_{ac2(RD-DB)}$	Data input access time (for setting with wait)		(Note 2)	ns
$t_{ac3(RD-DB)}$	Data input access time (when accessing multiplex bus area)		(Note 3)	ns
$t_{su(DB-RD)}$	Data input setup time	40		ns
$t_{su(RDY-BCLK)}$	$\overline{RDY}$ input setup time	30		ns
$t_{su(HOLD-BCLK)}$	$\overline{HOLD}$ input setup time	40		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 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 45 \quad [ns]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 45 \quad [ns]$$

n is “2” for 1-wait setting, “3” for 2-wait setting and “4” for 3-wait setting.

Note 3: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 45 \quad [ns]$$

n is “2” for 2-wait setting, “3” for 3-wait setting.

V_{CC1} = V_{CC2} = 5V

Timing Requirements

(V_{CC1} = V_{CC2} = 5V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 1.26.13. 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 1.26.14. 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 1.26.15. 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 1.26.16. 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 1.26.17. Timer A Input (Counter Increment/decrement 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

Table 1.26.18. Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi_{IN} input cycle time	800		ns
$t_{su(TAIN-TAOUT)}$	TAi_{OUT} input setup time	200		ns
$t_{su(TAOUT-TAIN)}$	TAi_{IN} input setup time	200		ns

V_{CC1} = V_{CC2} = 5V

Timing Requirements

(V_{CC1} = V_{CC2} = 5V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 1.26.19. Timer B Input (Counter Input in Event Counter Mode)

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

Table 1.26.20. Timer B Input (Pulse Period Measurement Mode)

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

Table 1.26.21. Timer B Input (Pulse Width Measurement Mode)

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

Table 1.26.22. 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 1.26.23. Serial I/O

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

Table 1.26.24. External Interrupt $\overline{INTi}$ Input

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

V_{CC1} = V_{CC2} = 5V

Switching Characteristics

(V_{CC1} = V_{CC2} = 5V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C, CM15="1" unless otherwise specified)

Table 1.26.25. Memory Expansion and Microprocessor Modes (for setting with no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.26.1		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (refers to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (refers to RD)		0		ns
$t_{h(WR-AD)}$	Address output hold time (refers to WR)		(Note 2)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (refers to BCLK)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		–4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			25	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (refers to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (refers to BCLK)		4		ns
$t_{d(DB-WR)}$	Data output delay time (refers to WR)		(Note 1)		ns
$t_{h(WR-DB)}$	Data output hold time (refers to WR)(Note 3)		(Note 2)		ns

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

$$\frac{0.5 \times 10^9}{f(BCLK)} - 40 \quad [ns]$$

Note 2: Calculated according to the BCLK frequency as follows:

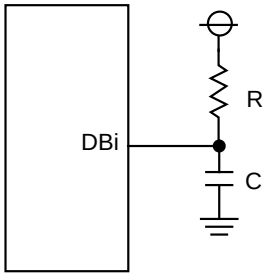
$$\frac{0.5 \times 10^9}{f(BCLK)} \quad [ns]$$

Note 3: This standard value shows the timing when the output is off, and does not show hold time of data bus.
Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in

$$t = -CR \times \ln (1 - V_{OL} / V_{CC2})$$

by a circuit of the right figure.
For example, when V_{OL} = 0.2V_{CC2}, C = 30pF, R = 1kΩ, hold time of output "L" level is

$$t = -30pF \times 1k\Omega \times \ln (1 - 0.2V_{CC2} / V_{CC2})$$
$$= 6.7ns.$$



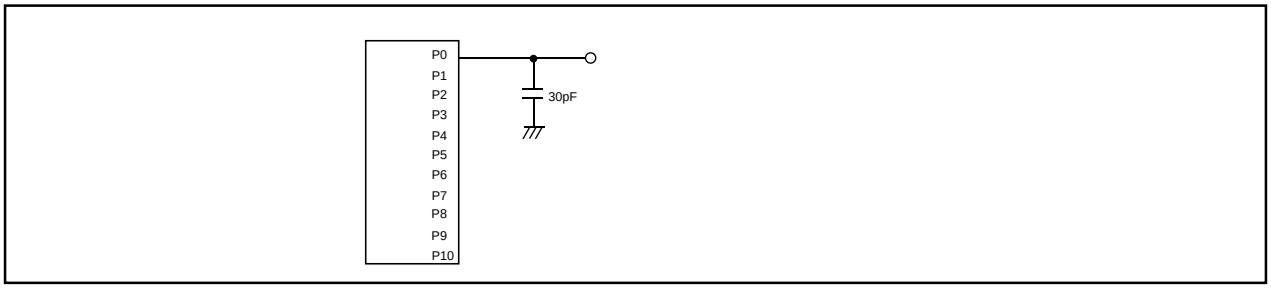


Figure 1.26.1. Ports P0 to P10 Measurement Circuit

VCC1 = VCC2 = 5V

Switching Characteristics

(VCC1 = VCC2 = 5V, VSS = 0V, at Topr = – 20 to 85°C / – 40 to 85°C, CM15="1" unless otherwise specified)

Table 1.26.26. Memory Expansion and Microprocessor Modes (for 1- to 3-wait setting and external area access)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.26.1		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (refers to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (refers to RD)		0		ns
$t_{h(WR-AD)}$	Address output hold time (refers to WR)		(Note 2)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (refers to BCLK)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		–4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			25	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (refers to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (refers to BCLK)		4		ns
$t_{d(DB-WR)}$	Data output delay time (refers to WR)		(Note 1)		ns
$t_{h(WR-DB)}$	Data output hold time (refers to WR)(Note 3)		(Note 2)		ns

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

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 40 \quad [ns]$$

n is "1" for 1-wait setting, "2" for 2-wait setting and "3" for 3-wait setting.

Note 2: Calculated according to the BCLK frequency as follows:

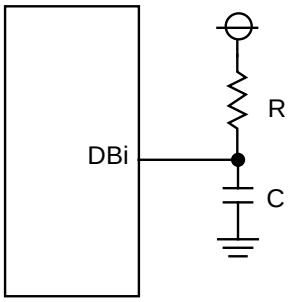
$$\frac{0.5 \times 10^9}{f(BCLK)} \quad [ns]$$

Note 3: This standard value shows the timing when the output is off, and does not show hold time of data bus.
Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in

$$t = -CR \times \ln (1 - V_{OL} / V_{CC2})$$

by a circuit of the right figure.
For example, when $V_{OL} = 0.2V_{CC2}$, C = 30pF, R = 1kΩ, hold time of output "L" level is

$$t = -30pF \times 1k\Omega \times \ln (1 - 0.2V_{CC2} / V_{CC2})$$
$$= 6.7ns.$$



VCC1 = VCC2 = 5V

Switching Characteristics

(VCC1 = VCC2 = 5V, VSS = 0V, at Topr = – 20 to 85°C / – 40 to 85°C, CM15="1" unless otherwise specified)

Table 1.26.27. Memory Expansion and Microprocessor Modes
(for 2- to 3-wait setting, external area access and multiplex bus selection)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.26.1		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (refers to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (refers to RD)		(Note 1)		ns
$t_{h(WR-AD)}$	Address output hold time (refers to WR)		(Note 1)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (refers to BCLK)		4		ns
$t_{h(RD-CS)}$	Chip select output hold time (refers to RD)		(Note 1)		ns
$t_{h(WR-CS)}$	Chip select output hold time (refers to WR)		(Note 1)		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			25	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (refers to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (refers to BCLK)		4		ns
$t_{d(DB-WR)}$	Data output delay time (refers to WR)		(Note 2)		ns
$t_{h(WR-DB)}$	Data output hold time (refers to WR)		(Note 1)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time (refers to BCLK)			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time (refers to BCLK)		– 4		ns
$t_{d(AD-ALE)}$	ALE signal output delay time (refers to Address)		(Note 3)		ns
$t_{h(ALE-AD)}$	ALE signal output hold time (refers to Adderss)		30		ns
$t_{d(AD-RD)}$	RD signal output delay from the end of Adress		0		ns
$t_{d(AD-WR)}$	WR signal output delay from the end of Adress		0		ns
$t_{dZ(RD-AD)}$	Address output floating start time			8	ns

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

$$\frac{0.5 \times 10^9}{f(BCLK)} \quad [ns]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 40 \quad [ns]$$

n is "2" for 2-wait setting, "3" for 3-wait setting.

Note 3: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 25 \quad [ns]$$

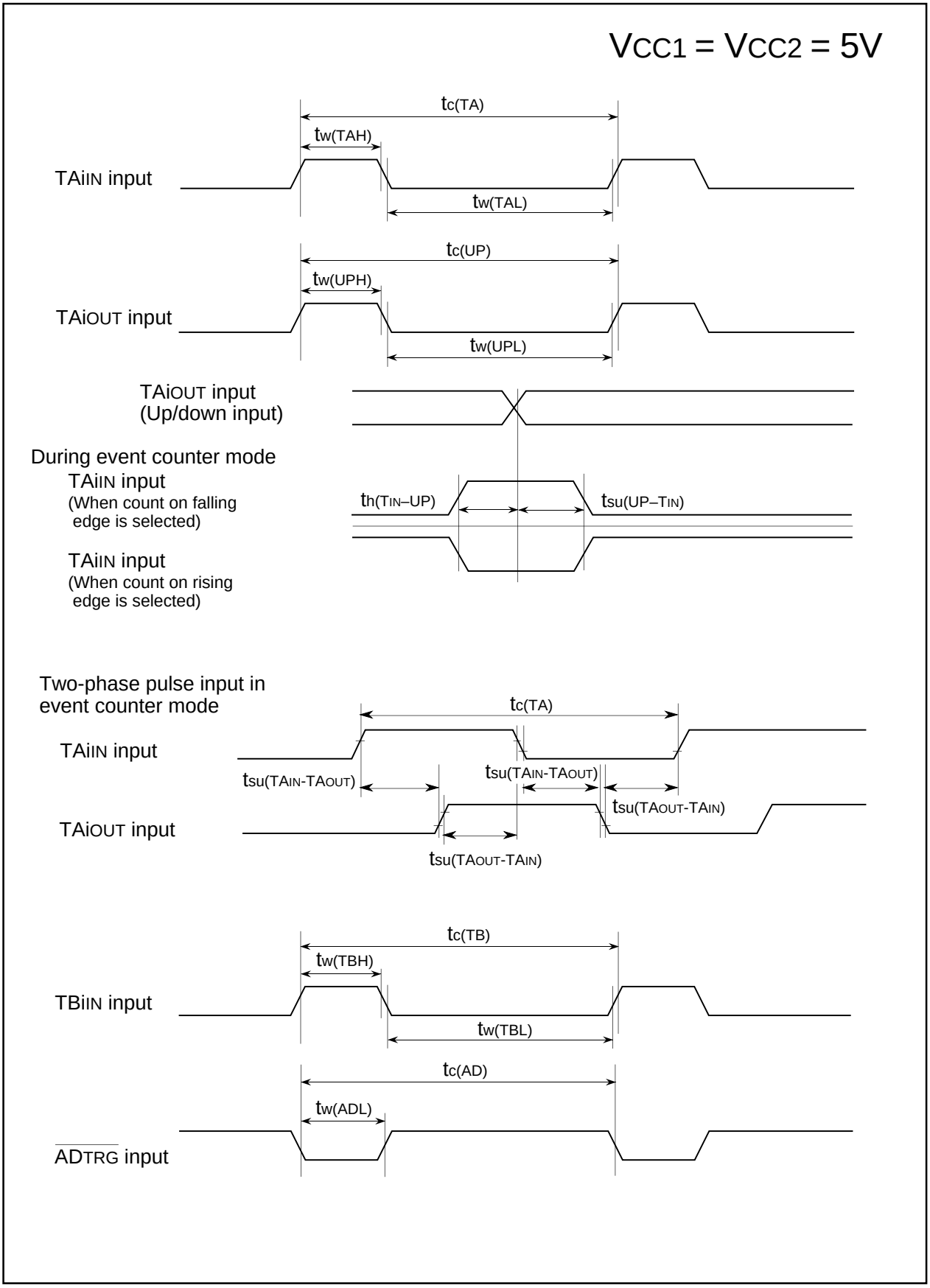


Figure 1.26.2. Timing Diagram (1)

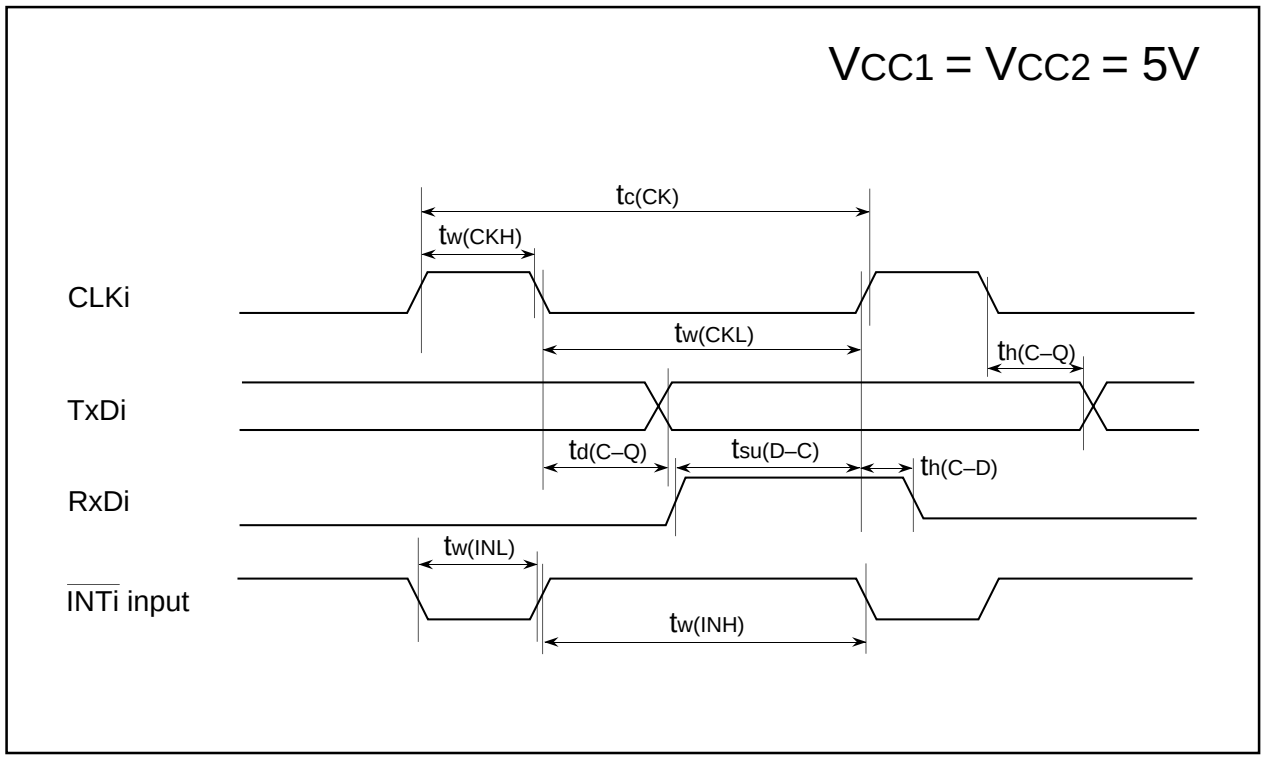
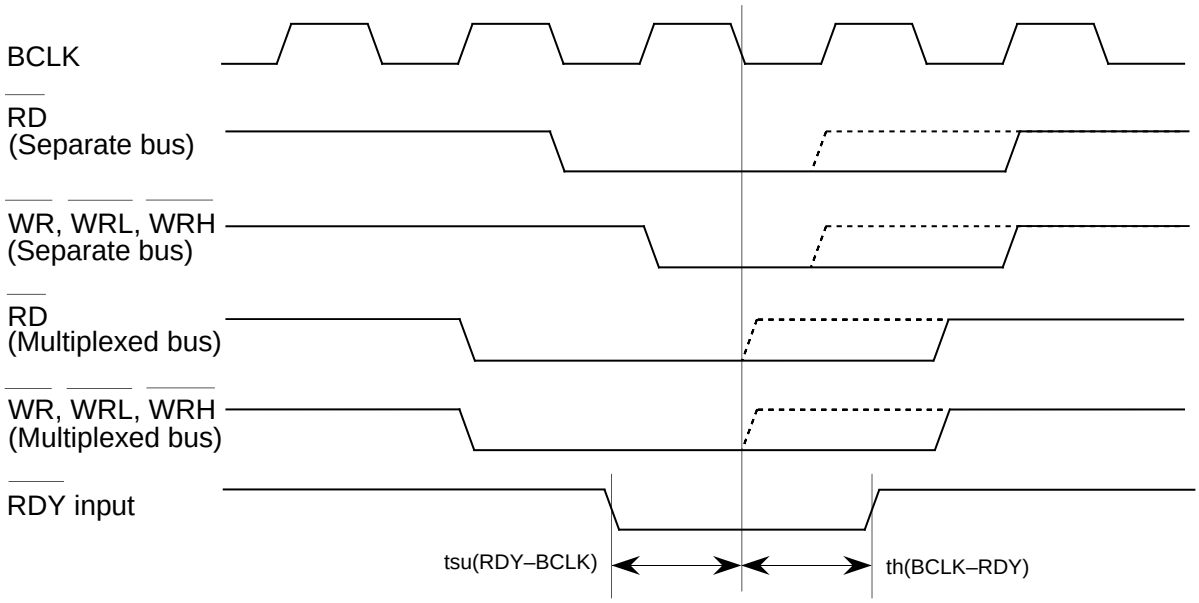


Figure 1.26.3. Timing Diagram (2)

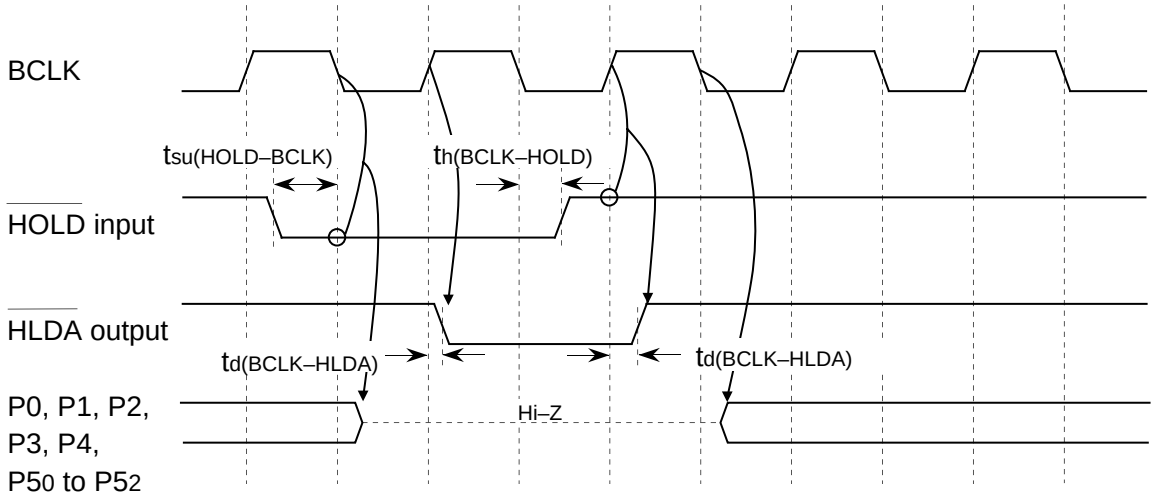
$V_{CC1} = V_{CC2} = 5V$

Memory Expansion Mode, Microprocessor Mode

(Effective for setting with wait**)**



(Common to setting with wait and setting without wait**)**



Note: The above pins are set to high-impedance regardless of the input level of the BYTE pin, PM06 bit in PM0 register and PM11 bit in PM1 register.

Measuring conditions :

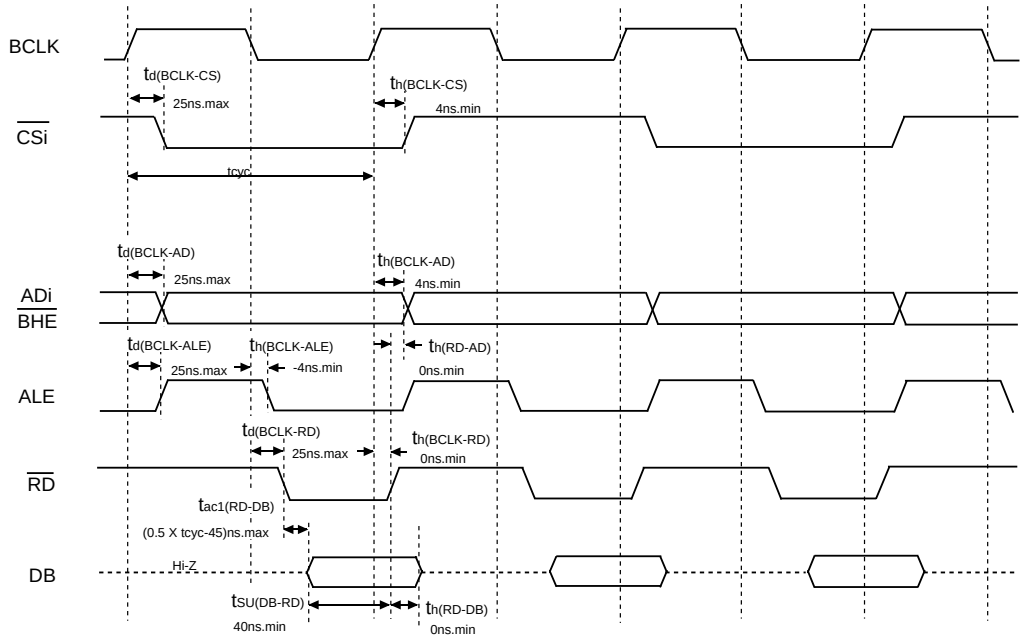
- V_{CC1}=V_{CC2}=5V
- Input timing voltage : Determined with V_{IL}=1.0V, V_{IH}=4.0V
- Output timing voltage : Determined with V_{OL}=2.5V, V_{OH}=2.5V

Figure 1.26.4. Timing Diagram (3)

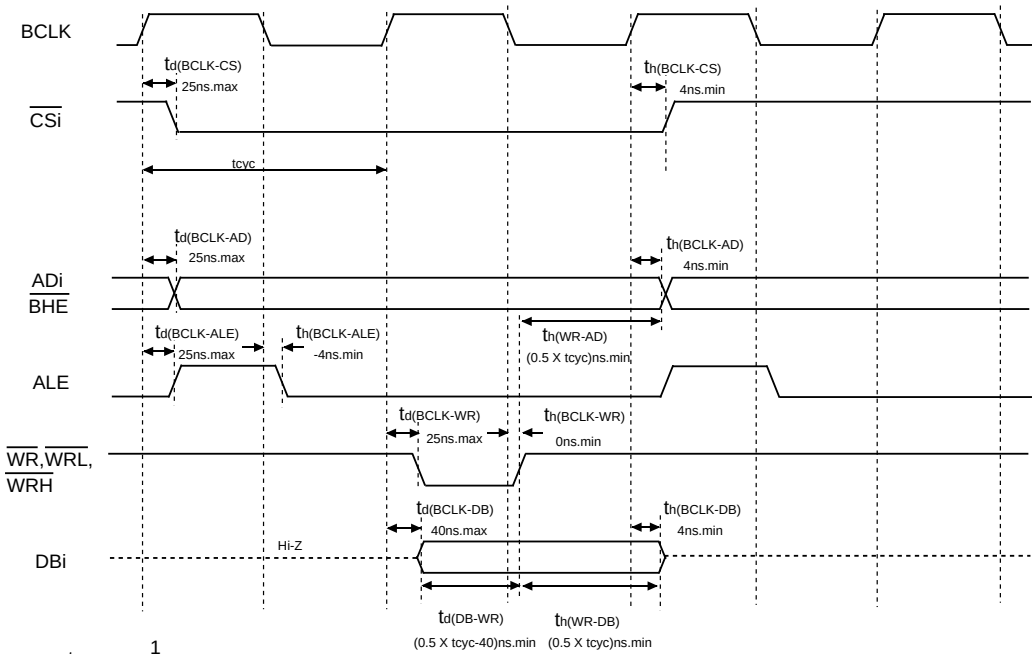
$V_{CC1} = V_{CC2} = 5V$

Memory Expansion Mode, Microprocessor Mode
(For setting with no wait**)**

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions
- V_{CC1}=V_{CC2}=5V
- Input timing voltage : V_{IL}=0.8V, V_{IH}=2.0V
- Output timing voltage : V_{OL}=0.4V, V_{OH}=2.4V

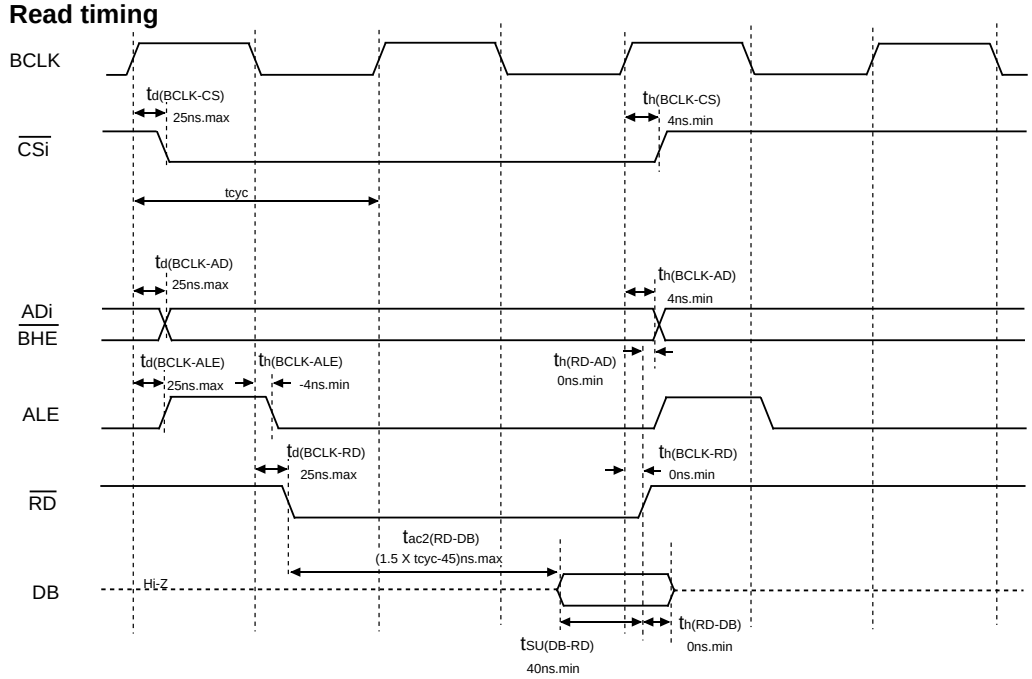
Figure 1.26.5. Timing Diagram (4)

$V_{CC1} = V_{CC2} = 5V$

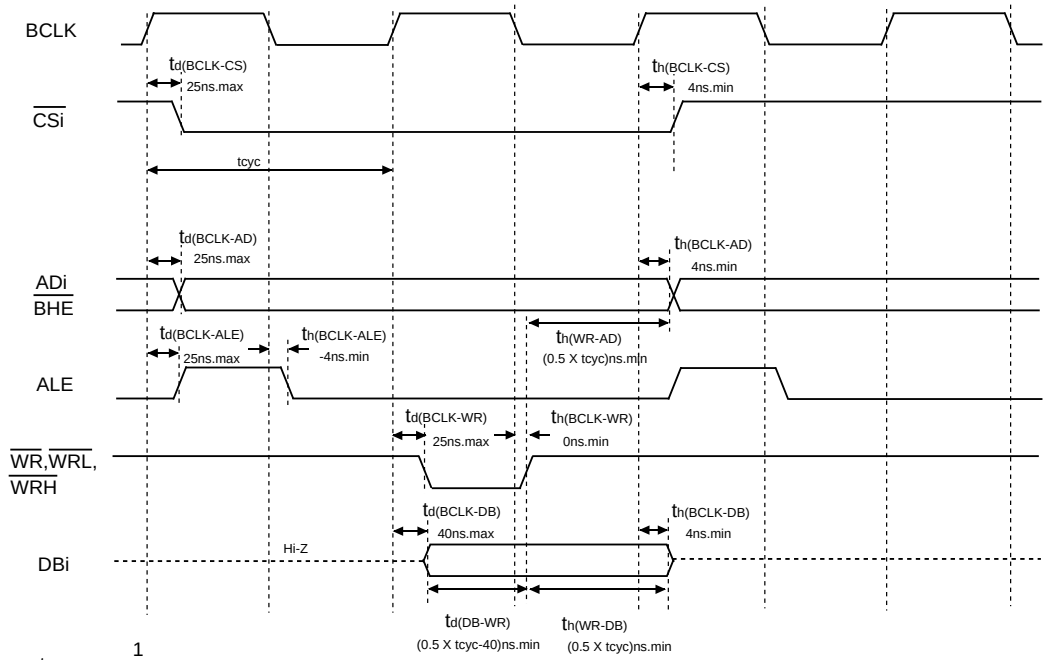
Memory Expansion Mode, Microprocessor Mode

(for 1-wait setting and external area access**)**

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions

- V_{CC1}=V_{CC2}=5V
- Input timing voltage : V_{IL}=0.8V, V_{IH}=2.0V
- Output timing voltage : V_{OL}=0.4V, V_{OH}=2.4V

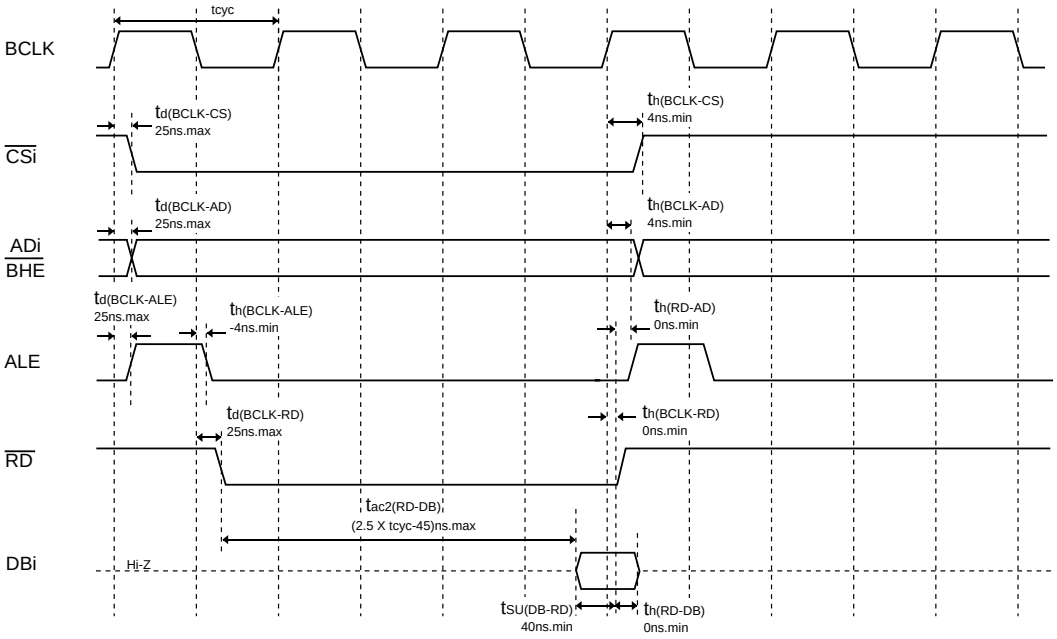
Figure 1.26.6. Timing Diagram (5)

V_{CC1} = V_{CC2} = 5V

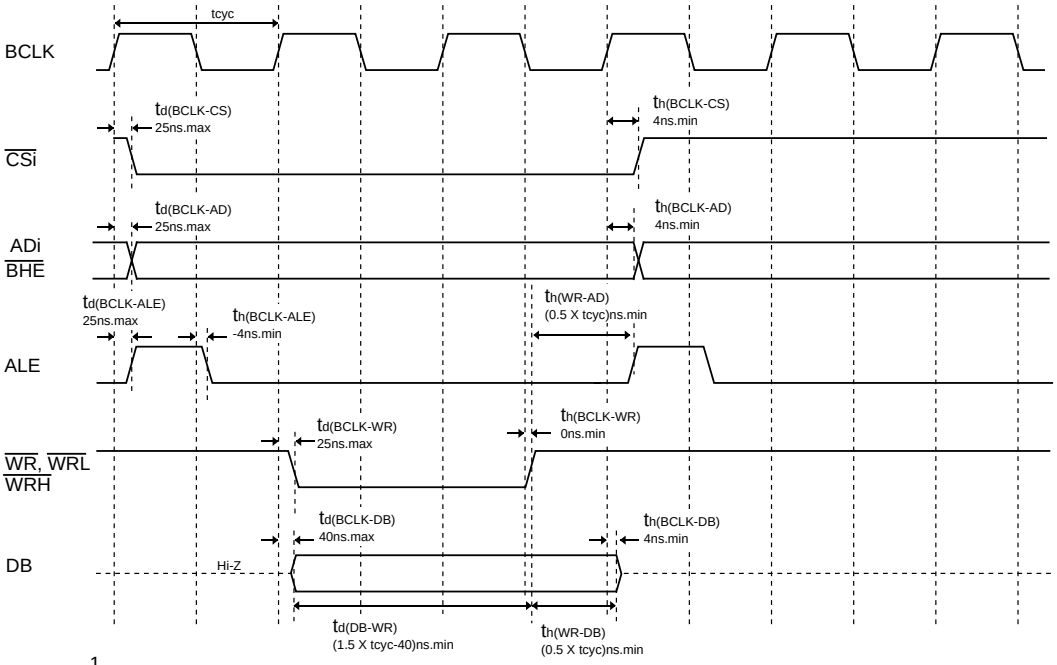
Memory Expansion Mode, Microprocessor Mode

(for 2-wait setting and external area access)

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions
- V_{CC1}=V_{CC2}=5V
- Input timing voltage : V_{IL}=0.8V, V_{IH}=2.0V
- Output timing voltage : V_{OL}=0.4V, V_{OH}=2.4V

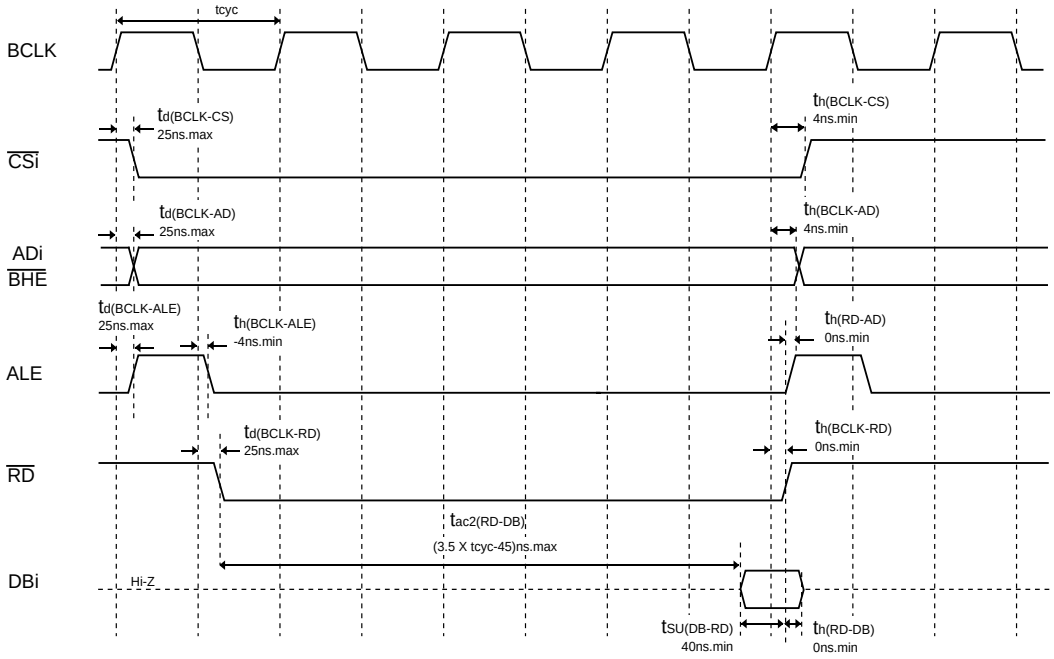
Figure 1.26.7. Timing Diagram (6)

V_{CC1} = V_{CC2} = 5V

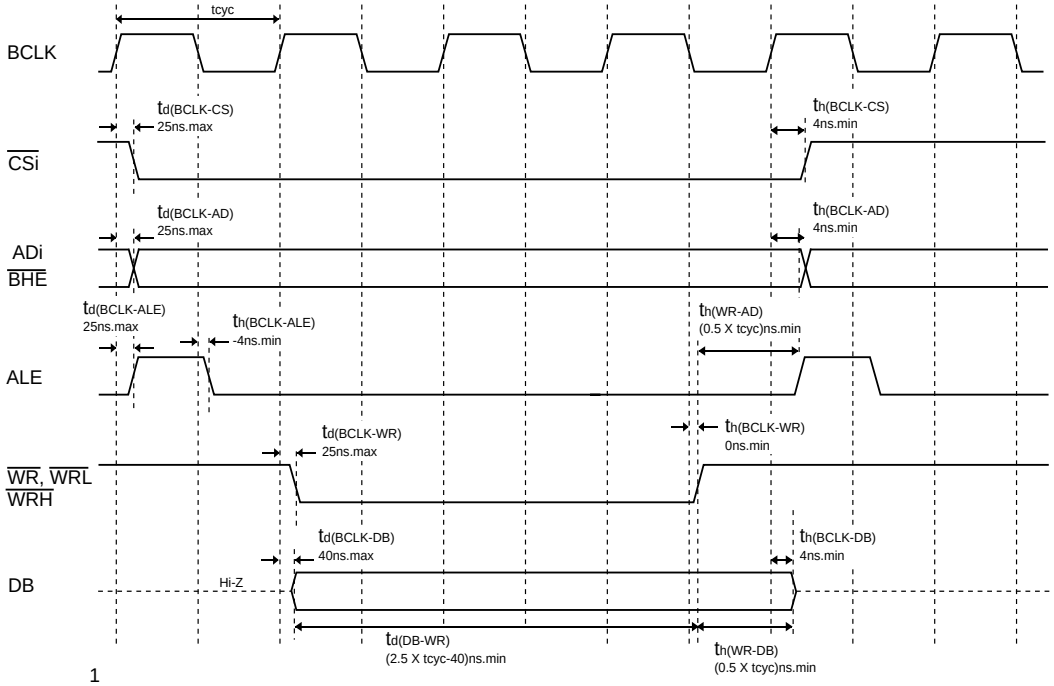
Memory Expansion Mode, Microprocessor Mode

(for 3-wait setting and external area access**)**

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions
- V_{CC1}=V_{CC2}=5V
- Input timing voltage : V_{IL}=0.8V, V_{IH}=2.0V
- Output timing voltage : V_{OL}=0.4V, V_{OH}=2.4V

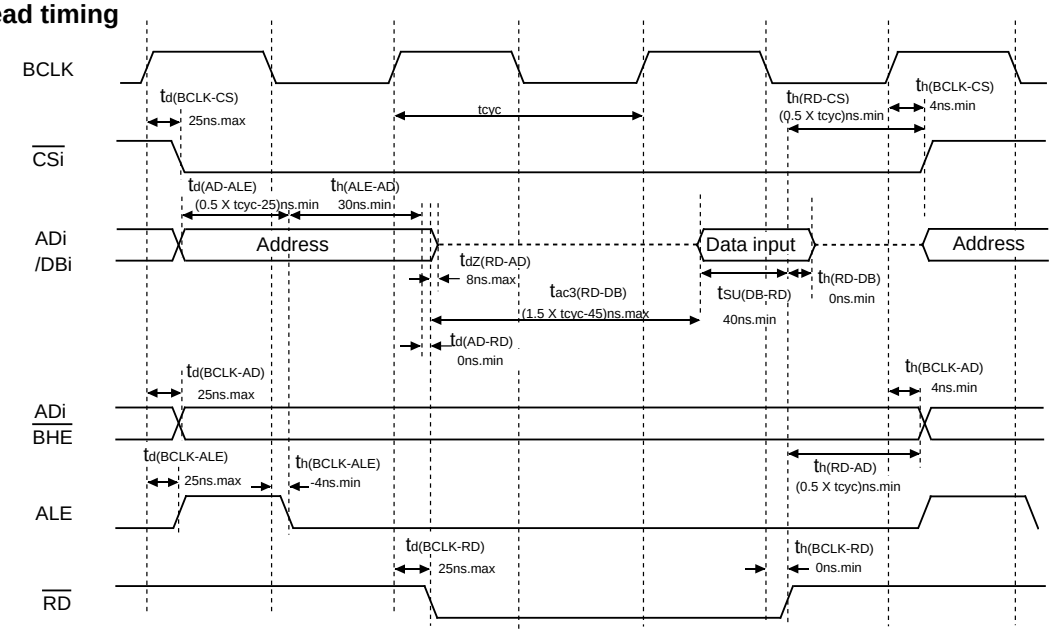
Figure 1.26.8. Timing Diagram (7)

$V_{CC1} = V_{CC2} = 5V$

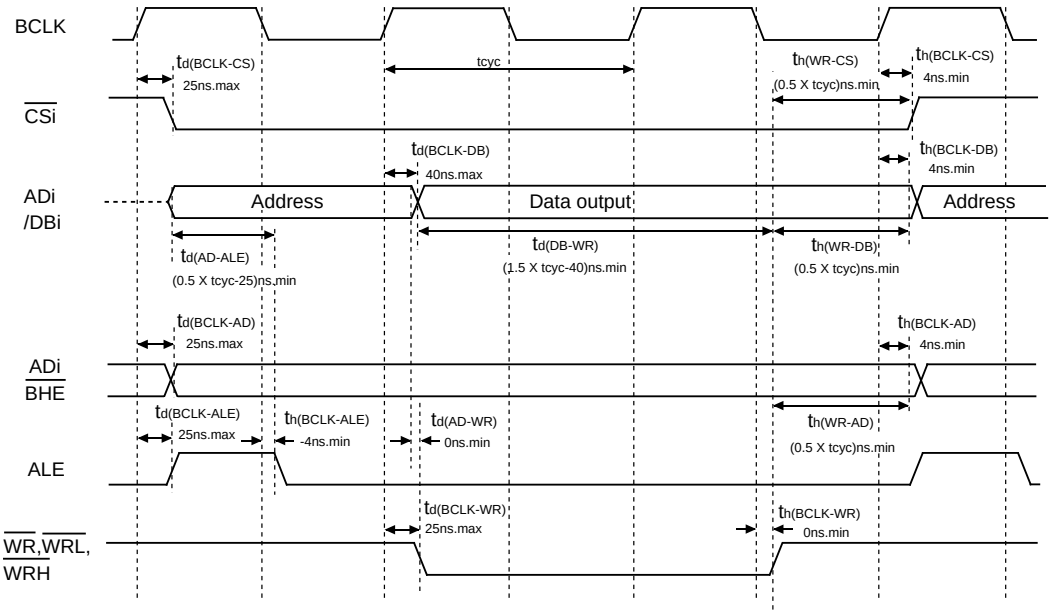
Memory Expansion Mode, Microprocessor Mode

(For 1- or 2-wait setting, external area access and multiplex bus selection**)**

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions

- V_{CC1}=V_{CC2}=5V
- Input timing voltage : V_{IL}=0.8V, V_{IH}=2.0V
- Output timing voltage : V_{OL}=0.4V, V_{OH}=2.4V

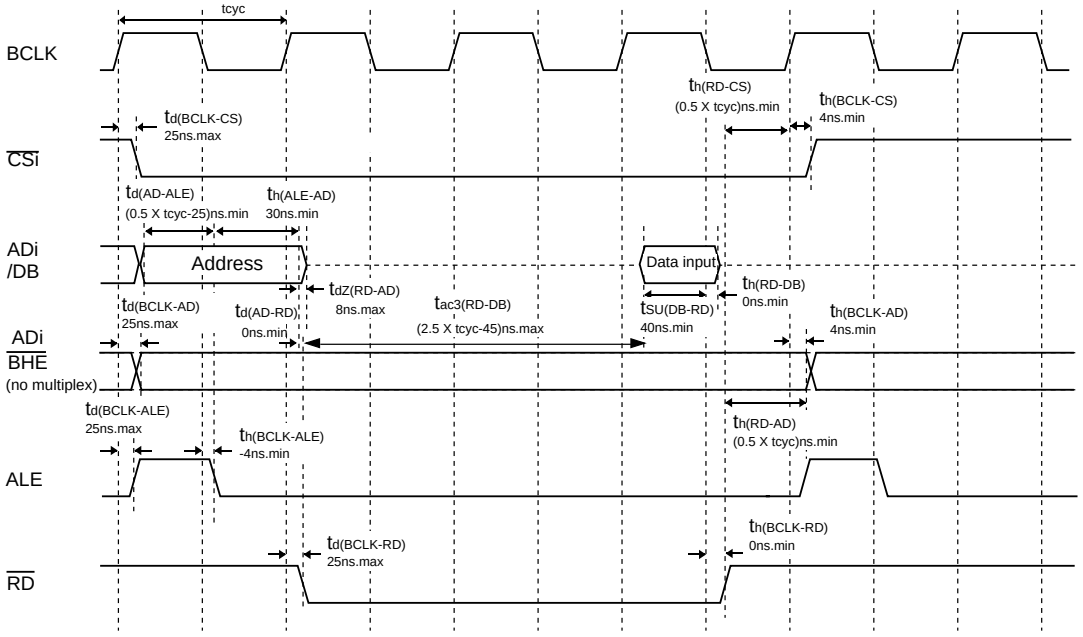
Figure 1.26.9. Timing Diagram (8)

VCC1 = VCC2 = 5V

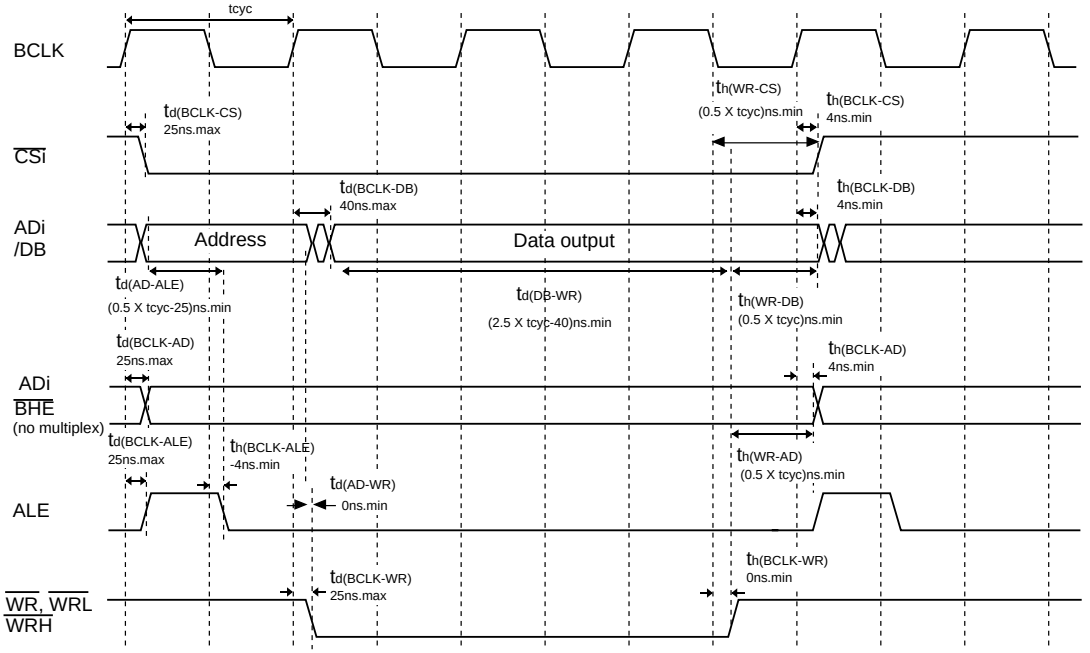
Memory Expansion Mode, Microprocessor Mode

(For 3-wait setting, external area access and multiplex bus selection**)**

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions
- V_{CC1}=V_{CC2}=5V
- Input timing voltage : V_{IL}=0.8V, V_{IH}=2.0V
- Output timing voltage : V_{OL}=0.4V, V_{OH}=2.4V

Figure 1.26.10. Timing Diagram (9)

Electrical Characteristics (V_{CC1} = V_{CC2} = 3V)

V_{CC1} = V_{CC2} = 3V

Table 1.26.28. Electrical Characteristics (Note)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_7$,$P3_0$ 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 $P10_7$, $P11_0$ to $P11_7$,$P12_0$ to $P12_7$,$P13_0$ to $P13_7$,$P14_0$,$P14_1$	I_{OH}=-1mA	Vcc-0.5		Vcc	V
V_{OH}	HIGH output voltage X_{OUT}	HIGHPOWER	I_{OH}=-0.1mA	Vcc-0.5		Vcc	V
		LOWPOWER	I_{OH}=-50µA	Vcc-0.5		Vcc	
	HIGH output voltage X_{COUT}	HIGHPOWER	With no load applied		2.5		V
		LOWPOWER	With no load applied		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 $P10_7$, $P11_0$ to $P11_7$,$P12_0$ to $P12_7$,$P13_0$ to $P13_7$,$P14_0$,$P14_1$	I_{OL}=1mA			0.5	V
V_{OL}	LOW output voltage X_{OUT}	HIGHPOWER	I_{OL}=0.1mA			0.5	V
		LOWPOWER	I_{OL}=50µA			0.5	
	LOW output voltage X_{COUT}	HIGHPOWER	With no load applied		0		V
		LOWPOWER	With no load applied		0		
V_{T+}-V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$ to $TA4_{IN}$, $TB0_{IN}$ to $TB5_{IN}$, $\overline{INT0}$ to $\overline{INT5}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS0}$ to $\overline{CTS2}$, SCL, SDA, $CLK0$ to $CLK4$, $TA2_{OUT}$ to $TA4_{OUT}$, $\overline{KI0}$ to $\overline{KI3}$, $RxD0$ to $RxD2$, $SIN3$,$SIN4$		0.2		0.8	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$		0.2	(0.7)	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 $P10_7$,$P11_0$ to $P11_7$, $P12_0$ to $P12_7$,$P13_0$ to $P13_7$,$P14_0$,$P14_1$, X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I=3V			4.0	µA
I_{IL}	LOW input current	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_7$,$P3_0$ 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 $P10_7$,$P11_0$ to $P11_7$, $P12_0$ to $P12_7$,$P13_0$ to $P13_7$,$P14_0$,$P14_1$, X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I=0V			-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 $P10_7$, $P11_0$ to $P11_7$,$P12_0$ to $P12_7$,$P13_0$ to $P13_7$,$P14_0$,$P14_1$	V_I=0V	66	160	500	kΩ
R_{fXIN}	Feedback resistance X_{IN}				3.0		MΩ
R_{fXCIN}	Feedback resistance X_{CIN}				25		MΩ
V_{RAM}	RAM retention voltage		At stop mode	2.0			V

Note : Referenced to V_{CC}=V_{CC1}=V_{CC2}=2.7 to 3.3V, V_{SS}=0V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK)=10MHz unless otherwise specified.

$V_{CC1} = V_{CC2} = 3V$

Table 1.26.29. Electrical Characteristics (2) (Note 1)

Symbol	Parameter		Measuring condition		Standard			Unit
					Min.	Typ.	Max.	
I_{CC}	Power supply current (V_{CC}=2.7 to 3.6V)	In single-chip mode, the output pins are open and other pins are V_{SS}	Mask ROM	f(BCLK)=10MHz, No division		8	11	mA
				No division, Ring oscillation		1		mA
			Flash memory	f(BCLK)=10MHz, No division		8	13	mA
				No division, Ring oscillation		1.8		mA
			Flash memory Program	f(BCLK)=10MHz, V_{CC1}=3.0V		12		mA
			Flash memory Erase	f(BCLK)=10MHz, V_{CC1}=3.0V		22		mA
			Mask ROM	f(X_{CIN})=32kHz, Low power dissipation mode, ROM(Note 3)		25		µA
			Flash memory	f(BCLK)=32kHz, Low power dissipation mode, RAM(Note 3)		25		µA
				f(BCLK)=32kHz, Low power dissipation mode, Flash memory(Note 3)		420		µA
				Ring oscillation, Wait mode		45		µA
			Mask ROM Flash memory	f(BCLK)=32kHz, Wait mode (Note 2), Oscillation capacity High		6.0		µA
				f(BCLK)=32kHz, Wait mode (Note 2), Oscillation capacity Low		1.8		µA
				Stop mode, T_{opr}=25°C		0.7	3.0	µA
Idet4	Power supply down detection dissipation current (Note 4)					0.6	4	µA
Idet3	Reset level detection dissipation current (Note 4)					0.4	2	µA
Idet2	RAM retention limit detection dissipation current (Note 4)					0.9	4	µA

Note 1: Referenced to V_{CC}=V_{CC1}=V_{CC2}=2.7 to 3.3V, V_{SS}=0V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK)=10MHz unless otherwise specified.
Note 2: With one timer operated using f_{C32}.
Note 3: This indicates the memory in which the program to be executed exists.
Note 4: Idet is dissipation current when the following bit is set to "1" (detection circuit enabled).
Idet4: VC27 bit of VCR2 register
Idet3: VC26 bit of VCR2 register
Idet2: VC25 bit of VCR2 register

$V_{CC1} = V_{CC2} = 3V$

Timing Requirements

(V_{CC1} = V_{CC2} = 3V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 1.26.30. External Clock Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	100		ns
$t_{w(H)}$	External clock input HIGH pulse width	40		ns
$t_{w(L)}$	External clock input LOW pulse width	40		ns
t_r	External clock rise time		18	ns
t_f	External clock fall time		18	ns

Table 1.26.31. Memory Expansion and Microprocessor Modes

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{ac1(RD-DB)}$	Data input access time (for setting with no wait)		(Note 1)	ns
$t_{ac2(RD-DB)}$	Data input access time (for setting with wait)		(Note 2)	ns
$t_{ac3(RD-DB)}$	Data input access time (when accessing multiplex bus area)		(Note 3)	ns
$t_{su(DB-RD)}$	Data input setup time	50		ns
$t_{su(RDY-BCLK)}$	$\overline{RDY}$ input setup time	40		ns
$t_{su(HOLD-BCLK)}$	$\overline{HOLD}$ input setup time	50		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 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 60 \quad [ns]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 60 \quad [ns]$$

n is “2” for 1-wait setting, “3” for 2-wait setting and “4” for 3-wait setting.

Note 3: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 60 \quad [ns]$$

n is “2” for 2-wait setting, “3” for 3-wait setting.

$V_{CC1} = V_{CC2} = 3V$

Timing Requirements

($V_{CC1} = V_{CC2} = 3V$, $V_{SS} = 0V$, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 1.26.32. Timer A Input (Counter Input in Event Counter Mode)

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

Table 1.26.33. Timer A Input (Gating Input in Timer Mode)

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

Table 1.26.34. 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	300		ns
$t_{w(TAH)}$	TAi_{IN} input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAi_{IN} input LOW pulse width	150		ns

Table 1.26.35. 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	150		ns
$t_{w(TAL)}$	TAi_{IN} input LOW pulse width	150		ns

Table 1.26.36. Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

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

Table 1.26.37. Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi_{IN} input cycle time	2		µs
$t_{su(TAIN-TAOUT)}$	TAi_{OUT} input setup time	500		ns
$t_{su(TAOUT-TAIN)}$	TAi_{IN} input setup time	500		ns

V_{CC1} = V_{CC2} = 3V

Timing Requirements

(V_{CC1} = V_{CC2} = 3V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 1.26.38. Timer B Input (Counter Input in Event Counter Mode)

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

Table 1.26.39. Timer B Input (Pulse Period Measurement Mode)

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

Table 1.26.40. Timer B Input (Pulse Width Measurement Mode)

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

Table 1.26.41. A-D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (trigger able minimum)	1500		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input LOW pulse width	200		ns

Table 1.26.42. Serial I/O

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

Table 1.26.43. External Interrupt $\overline{INTi}$ Input

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

$V_{CC1} \geq V_{CC2}$ = 3V

Switching Characteristics

(V_{CC1} = V_{CC2} = 3V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C, CM15="1" unless otherwise specified)

Table 1.26.44. Memory Expansion, Microprocessor Modes (for setting with no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.26.11		30	ns
$t_{h(BCLK-AD)}$	Address output hold time (refers to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (refers to RD)		0		ns
$t_{h(WR-AD)}$	Address output hold time (refers to WR)		(Note 2)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			30	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (refers to BCLK)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			30	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		–4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			30	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			30	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (refers to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (refers to BCLK)		4		ns
$t_{d(DB-WR)}$	Data output delay time (refers to WR)		(Note 1)		ns
$t_{h(WR-DB)}$	Data output hold time (refers to WR)(Note 3)		(Note 2)		ns

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

$$\frac{0.5 \times 10^9}{f(BCLK)} - 40 \quad [ns]$$

Note 2: Calculated according to the BCLK frequency as follows:

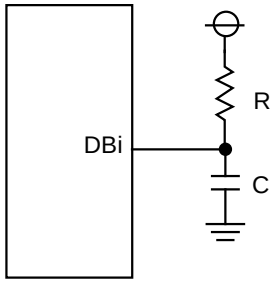
$$\frac{0.5 \times 10^9}{f(BCLK)} \quad [ns]$$

Note 3: This standard value shows the timing when the output is off, and does not show hold time of data bus.
Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in

$t = -CR \times \ln(1 - V_{OL} / V_{CC2})$

by a circuit of the right figure.
For example, when $V_{OL} = 0.2V_{CC2}$, C = 30pF, R = 1kΩ, hold time of output "L" level is

$t = -30pF \times 1k\Omega \times \ln(1 - 0.2V_{CC2} / V_{CC2})$
$= 6.7ns.$



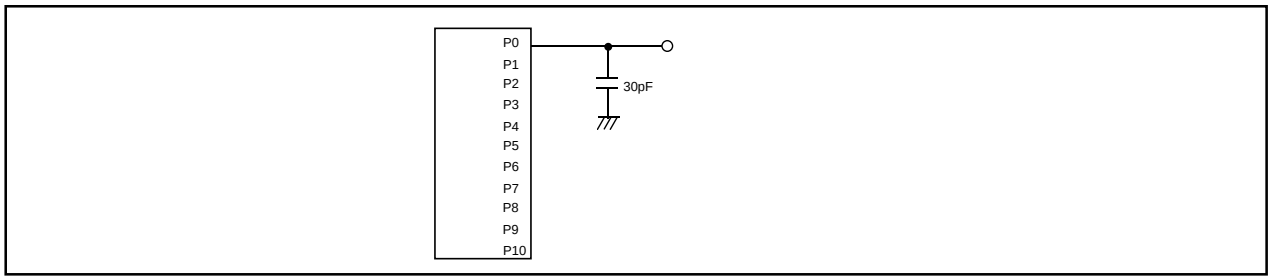


Figure 1.26.11. Ports P0 to P10 Measurement Circuit

$V_{CC1} \geq V_{CC2}$ = 3V

Switching Characteristics

(V_{CC1} = V_{CC2} = 3V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C, CM15="1" unless otherwise specified)

Table 1.26.45. Memory expansion and Microprocessor Modes (for 1- to 3-wait setting and external area access)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.26.11		30	ns
$t_{h(BCLK-AD)}$	Address output hold time (refers to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (refers to RD)		0		ns
$t_{h(WR-AD)}$	Address output hold time (refers to WR)		(Note 2)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			30	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (refers to BCLK)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			30	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		–4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			30	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			30	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (refers to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (refers to BCLK)		4		ns
$t_{d(DB-WR)}$	Data output delay time (refers to WR)		(Note 1)		ns
$t_{h(WR-DB)}$	Data output hold time (refers to WR)(Note 3)		(Note 2)		ns

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

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 40 \quad [ns]$$

n is "1" for 1-wait setting, "2" for 2-wait setting and "3" for 3-wait setting.

Note 2: Calculated according to the BCLK frequency as follows:

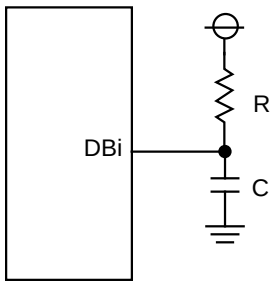
$$\frac{0.5 \times 10^9}{f(BCLK)} \quad [ns]$$

Note 3: This standard value shows the timing when the output is off, and does not show hold time of data bus.
Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in

$t = -CR \times \ln (1 - V_{OL} / V_{CC2})$

by a circuit of the right figure.
For example, when $V_{OL} = 0.2V_{CC2}$, C = 30pF, R = 1kΩ, hold time of output "L" level is

$t = -30pF \times 1k\Omega \times \ln (1 - 0.2V_{CC2} / V_{CC2})$
$= 6.7ns.$



$V_{CC1} \geq V_{CC2}$ = 3V

Switching Characteristics

(V_{CC1} = V_{CC2} = 3V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C, CM15="1" unless otherwise specified)

Table 1.26.46. Memory expansion and Microprocessor Modes
(for 2- to 3-wait setting, external area access and multiplex bus selection)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.26.11		50	ns
$t_{h(BCLK-AD)}$	Address output hold time (refers to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (refers to RD)		(Note 1)		ns
$t_{h(WR-AD)}$	Address output hold time (refers to WR)		(Note 1)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			50	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (refers to BCLK)		4		ns
$t_{h(RD-CS)}$	Chip select output hold time (refers to RD)		(Note 1)		ns
$t_{h(WR-CS)}$	Chip select output hold time (refers to WR)		(Note 1)		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 (refers to BCLK)			50	ns
$t_{h(BCLK-DB)}$	Data output hold time (refers to BCLK)		4		ns
$t_{d(DB-WR)}$	Data output delay time (refers to WR)		(Note 2)		ns
$t_{h(WR-DB)}$	Data output hold time (refers to WR)		(Note 1)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time (refers to BCLK)			40	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time (refers to BCLK)		– 4		ns
$t_{d(AD-ALE)}$	ALE signal output delay time (refers to Address)		(Note 3)		ns
$t_{h(ALE-AD)}$	ALE signal output hold time (refers to Adderss)		30		ns
$t_{d(AD-RD)}$	RD signal output delay from the end of Address		0		ns
$t_{d(AD-WR)}$	WR signal output delay from the end of Address		0		ns
$t_{dZ(RD-AD)}$	Address output floating start time			8	ns

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

$$\frac{0.5 \times 10^9}{f(BCLK)} \quad [ns]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 50 \quad [ns]$$

n is "2" for 2-wait setting, "3" for 3-wait setting.

Note 3: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 40 \quad [ns]$$

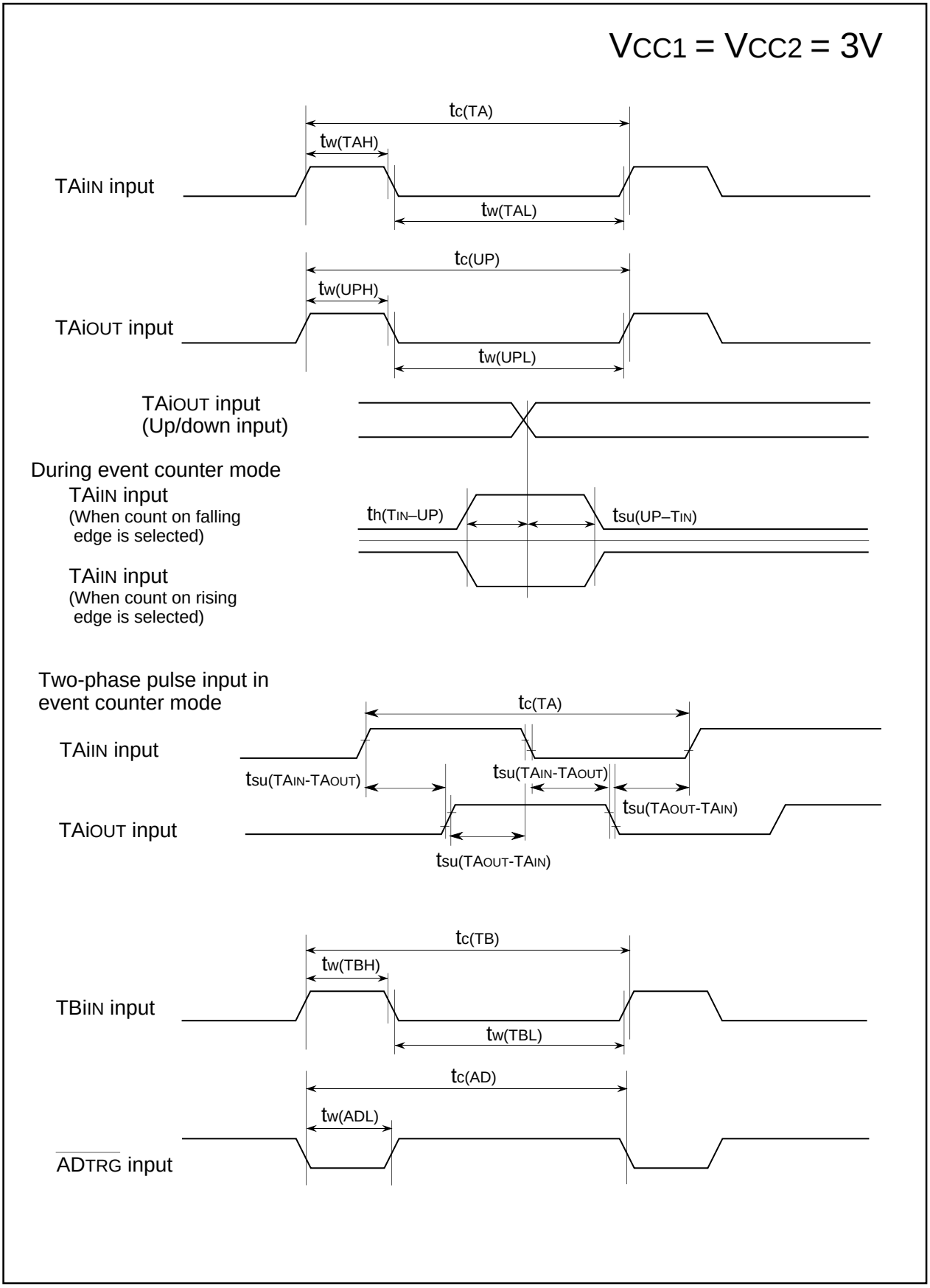


Figure 1.26.12. Timing Diagram (1)

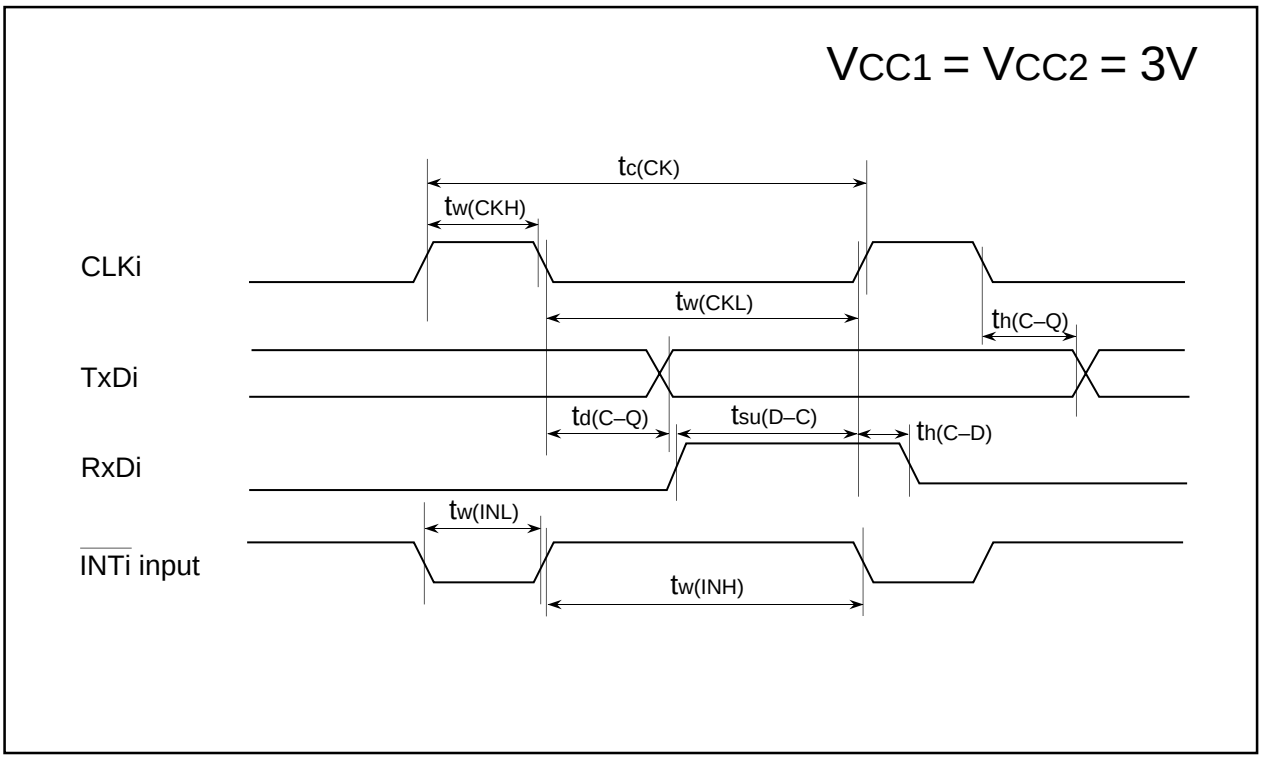
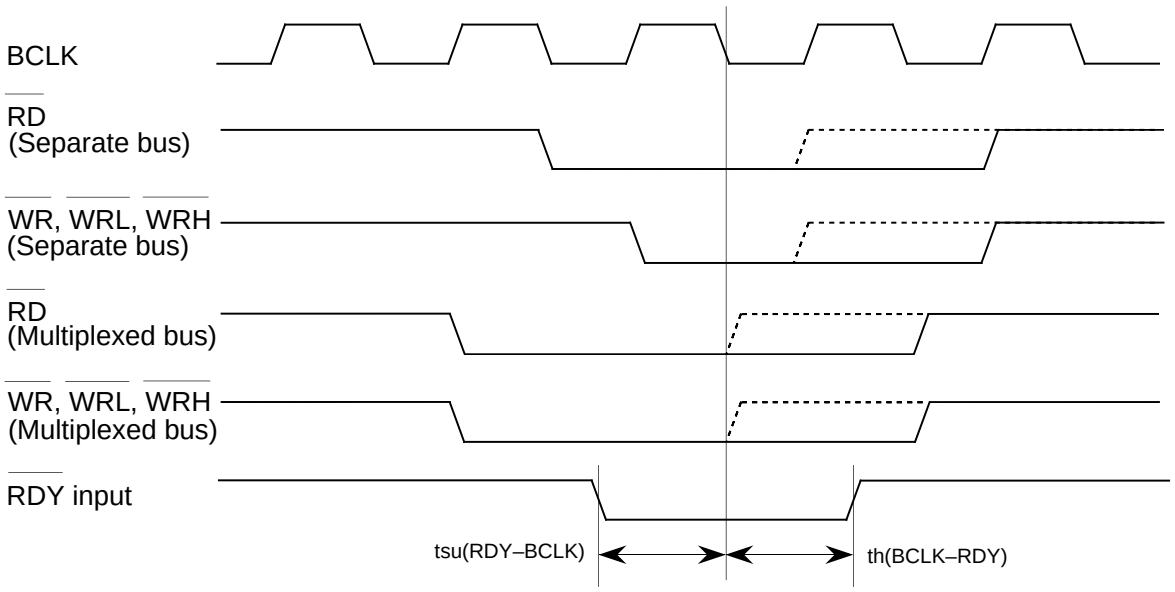


Figure 1.26.13. Timing Diagram (2)

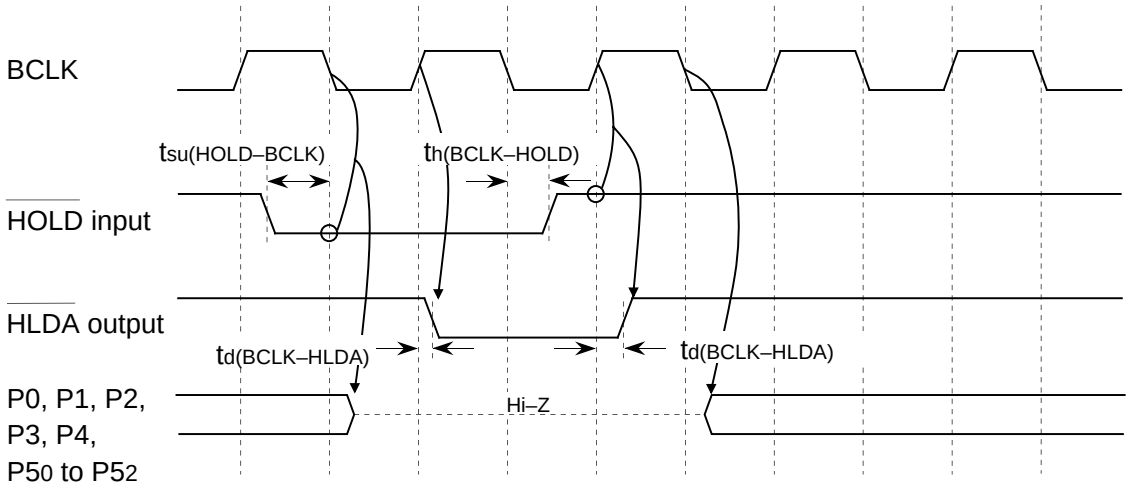
$V_{CC1} = V_{CC2} = 3V$

Memory Expansion Mode, Microprocessor Mode

(Effective for setting with wait**)**



(Common to setting with wait and setting without wait**)**



Note: The above pins are set to high-impedance regardless of the input level of the BYTE pin, PM06 bit of PM0 register and PM11 bit of PM1 register.

Measuring conditions :

- V_{CC1}=V_{CC2}=3V
- Input timing voltage : Determined with V_{IL}=0.6V, V_{IH}=2.4V
- Output timing voltage : Determined with V_{OL}=1.5V, V_{OH}=1.5V

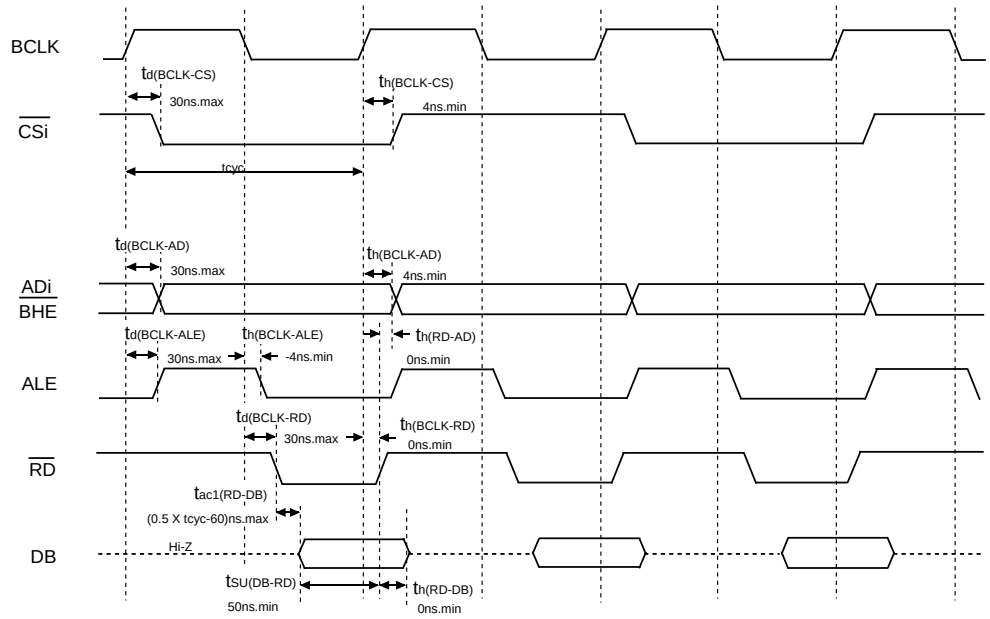
Figure 1.26.14. Timing Diagram (3)

$V_{CC1} \geq V_{CC2}$ = 3V

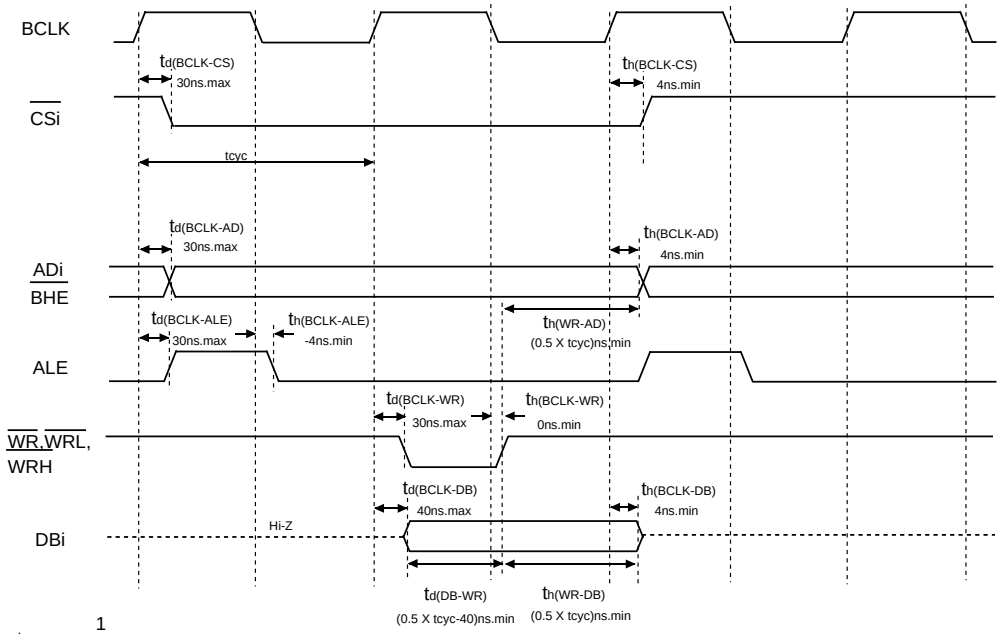
Memory Expansion Mode, Microprocessor Mode

(For setting with no wait**)**

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions

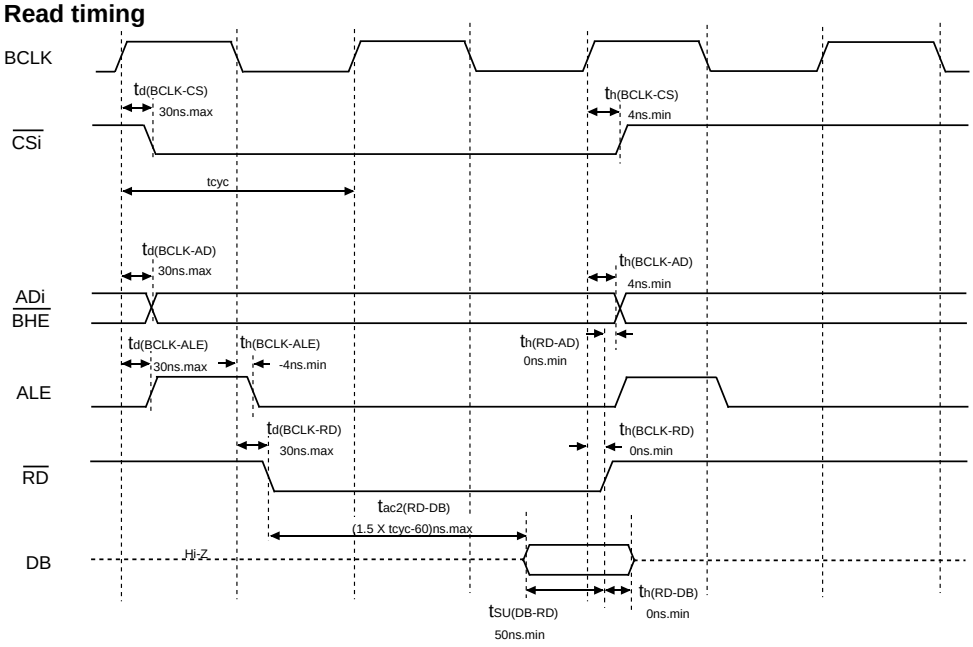
- V_{CC1}=V_{CC2}=3V
- Input timing voltage : V_{IL}=0.6V, V_{IH}=2.4V
- Output timing voltage : V_{OL}=1.5V, V_{OH}=1.5V

Figure 1.26.15. Timing Diagram (4)

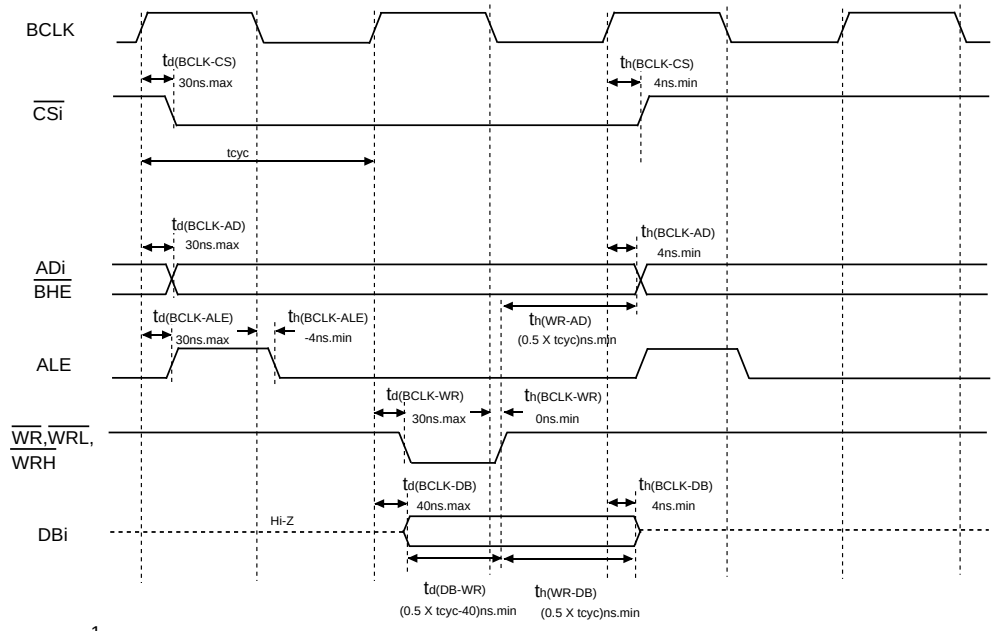
$V_{CC1} \geq V_{CC2}$ = 3V

Memory Expansion Mode, Microprocessor Mode
(for 1-wait setting and external area access**)**

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

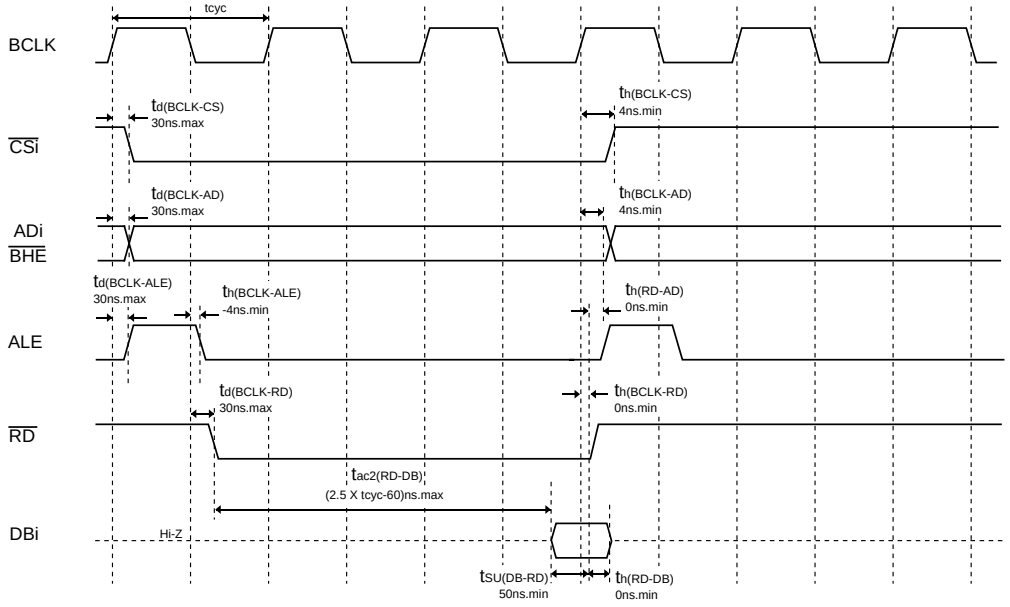
Measuring conditions
- V_{CC1}=V_{CC2}=3V
- Input timing voltage : V_{IL}=0.6V, V_{IH}=2.4V
- Output timing voltage : V_{OL}=1.5V, V_{OH}=1.5V

Figure 1.26.16. Timing Diagram (5)

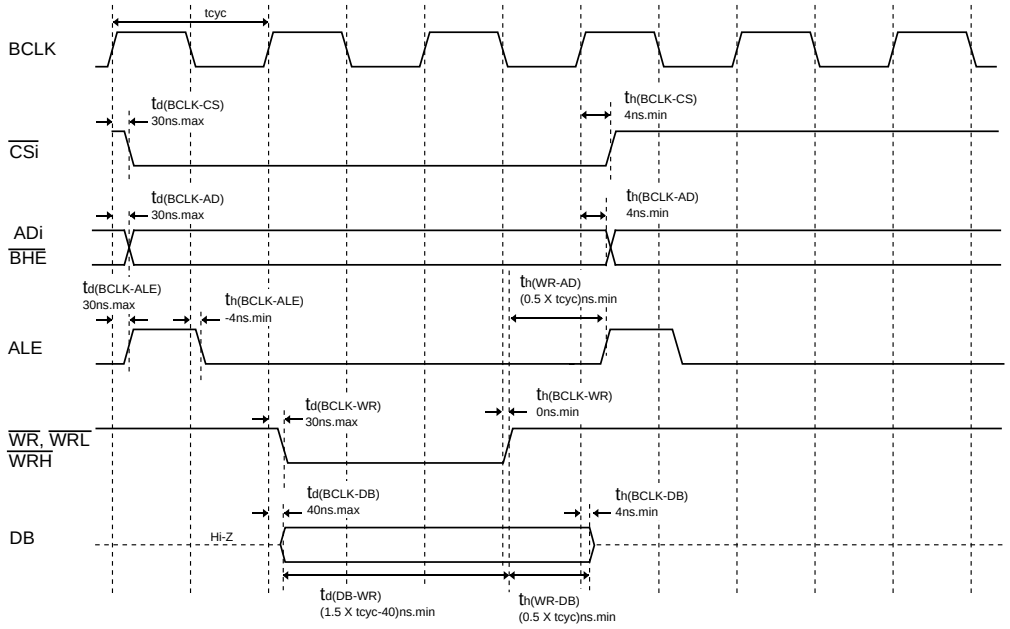
$V_{CC1} \geq V_{CC2}$ = 3V

Memory Expansion Mode, Microprocessor Mode
(for 2-wait setting and external area access**)**

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions
- V_{CC1}=V_{CC2}=3V
- Input timing voltage : V_{IL}=0.6V, V_{IH}=2.4V
- Output timing voltage : V_{OL}=1.5V, V_{OH}=1.5V

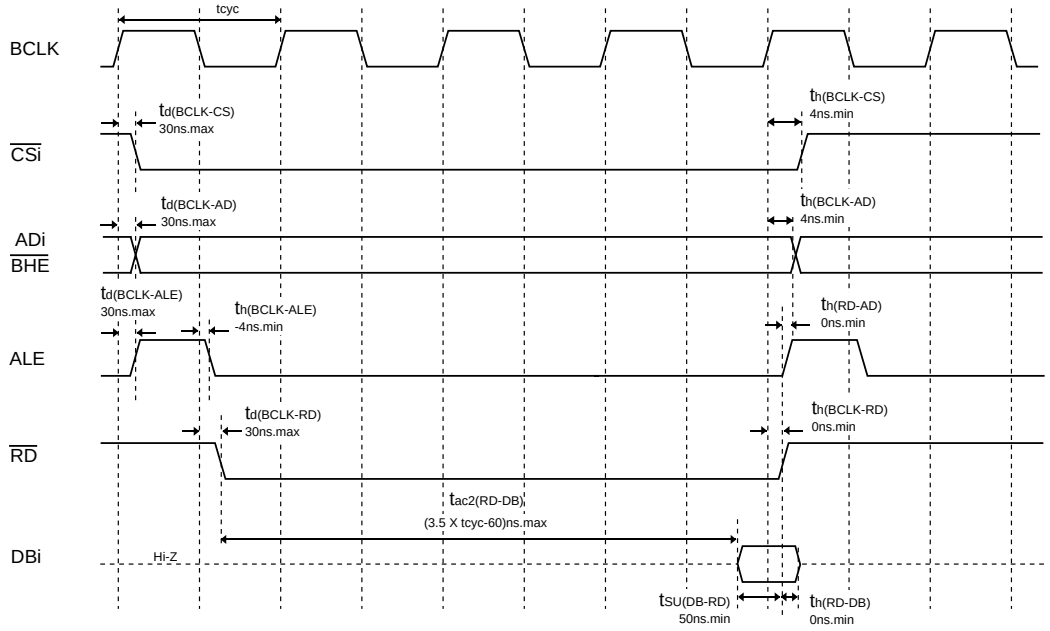
Figure 1.26.17. Timing Diagram (6)

$V_{CC1} \geq V_{CC2}$ = 3V

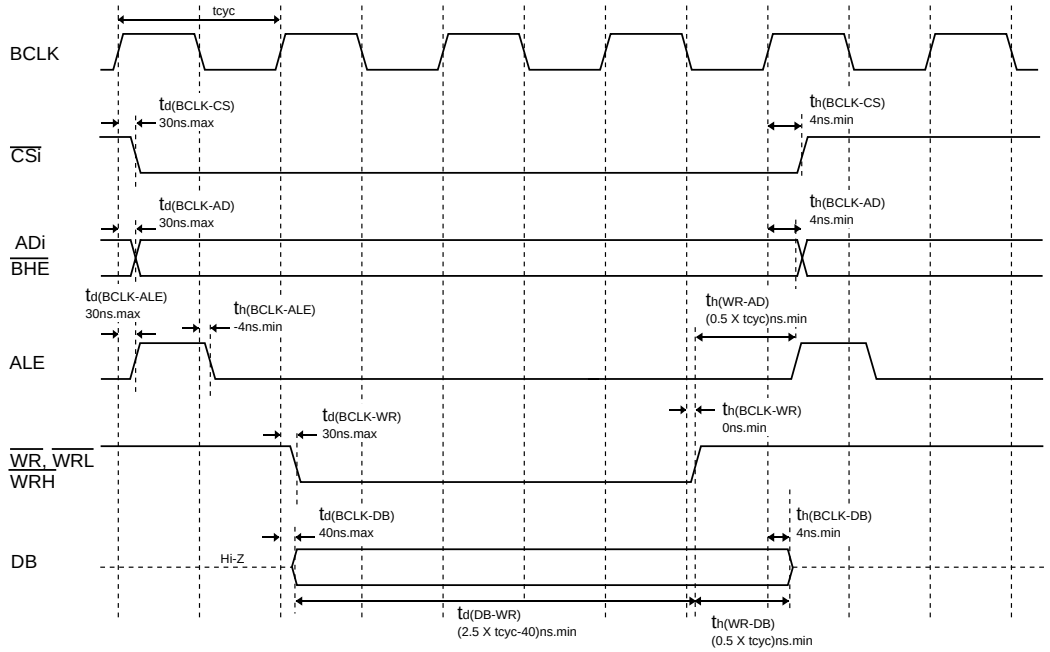
Memory Expansion Mode, Microprocessor Mode

(for 3-wait setting and external area access**)**

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions
- V_{CC1}=V_{CC2}=3V
- Input timing voltage : V_{IL}=0.6V, V_{IH}=2.4V
- Output timing voltage : V_{OL}=1.5V, V_{OH}=1.5V

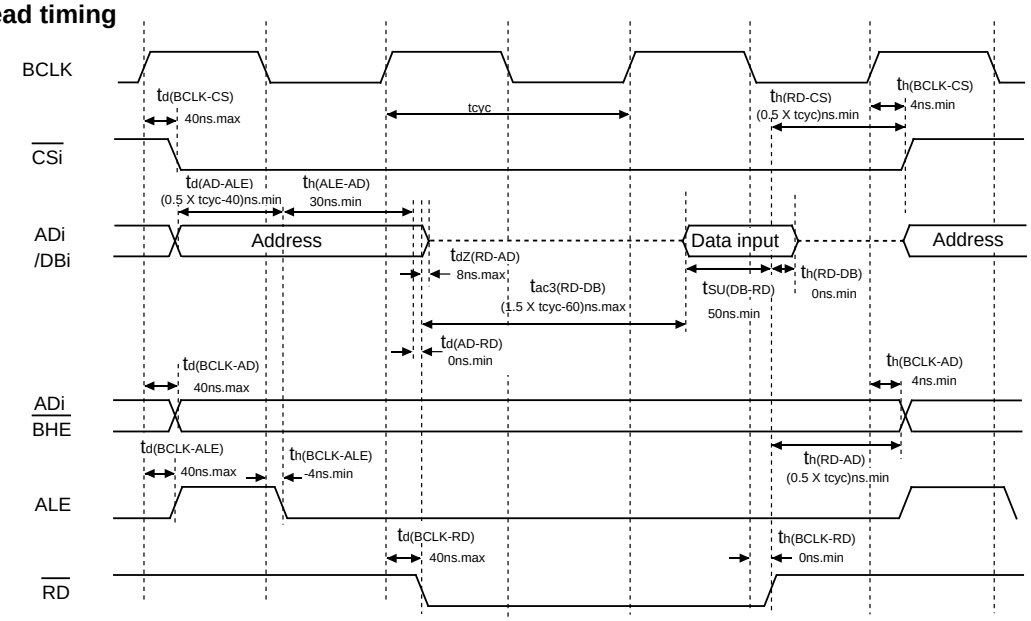
Figure 1.26.18. Timing Diagram (7)

$V_{CC1} \geq V_{CC2}$ = 3V

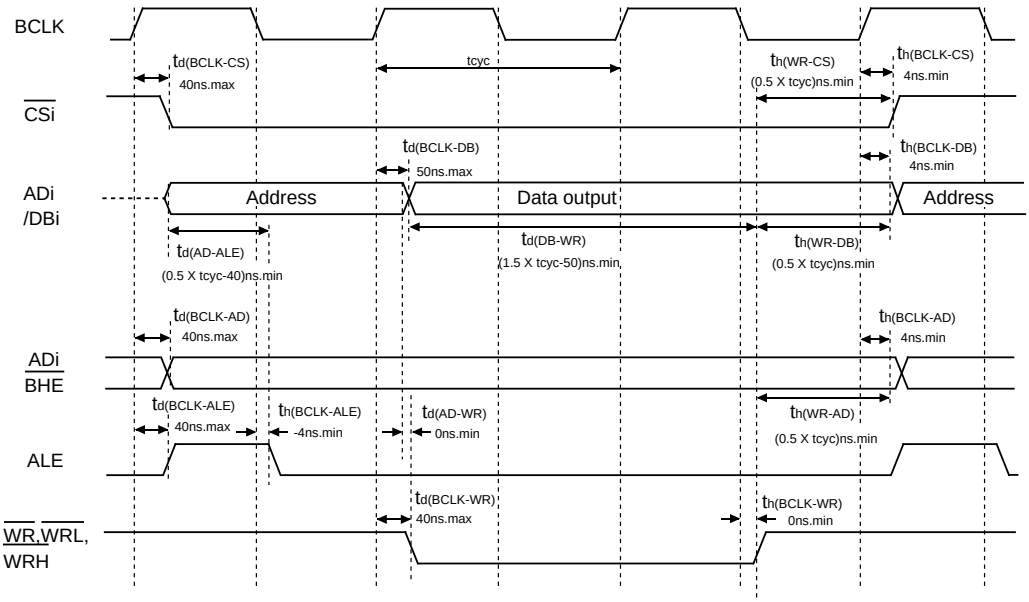
Memory Expansion Mode, Microprocessor Mode

(For 2-wait setting, external area access and multiplex bus selection)

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions
- V_{CC1}=V_{CC2}=3V
- Input timing voltage : V_{IL}=0.6V, V_{IH}=2.4V
- Output timing voltage : V_{OL}=1.5V, V_{OH}=1.5V

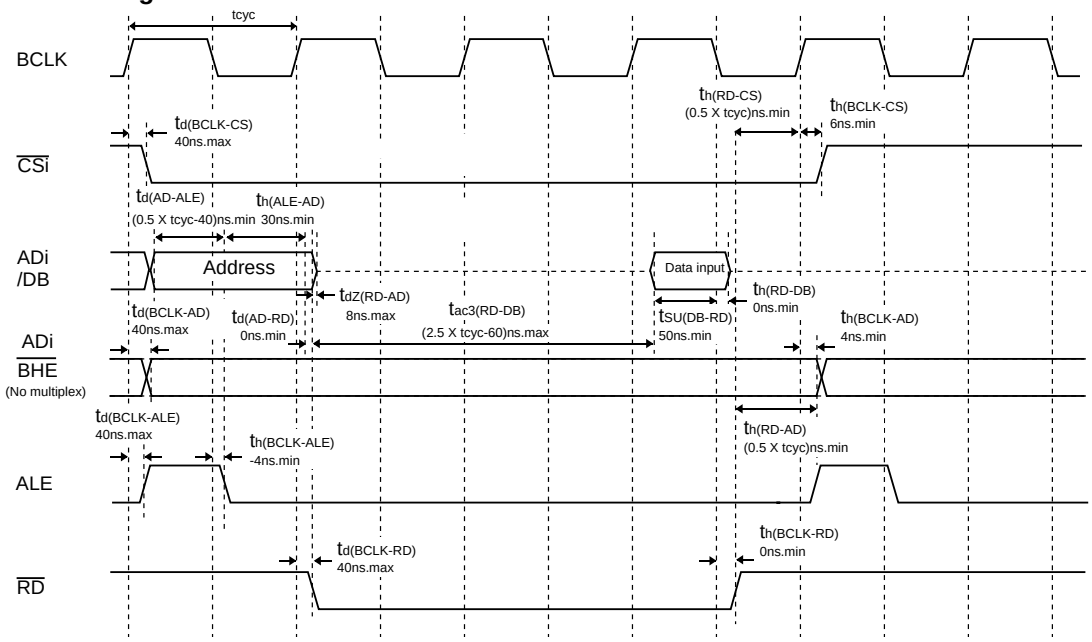
Figure 1.26.19. Timing Diagram (8)

$V_{CC1} \geq V_{CC2}$ = 3V

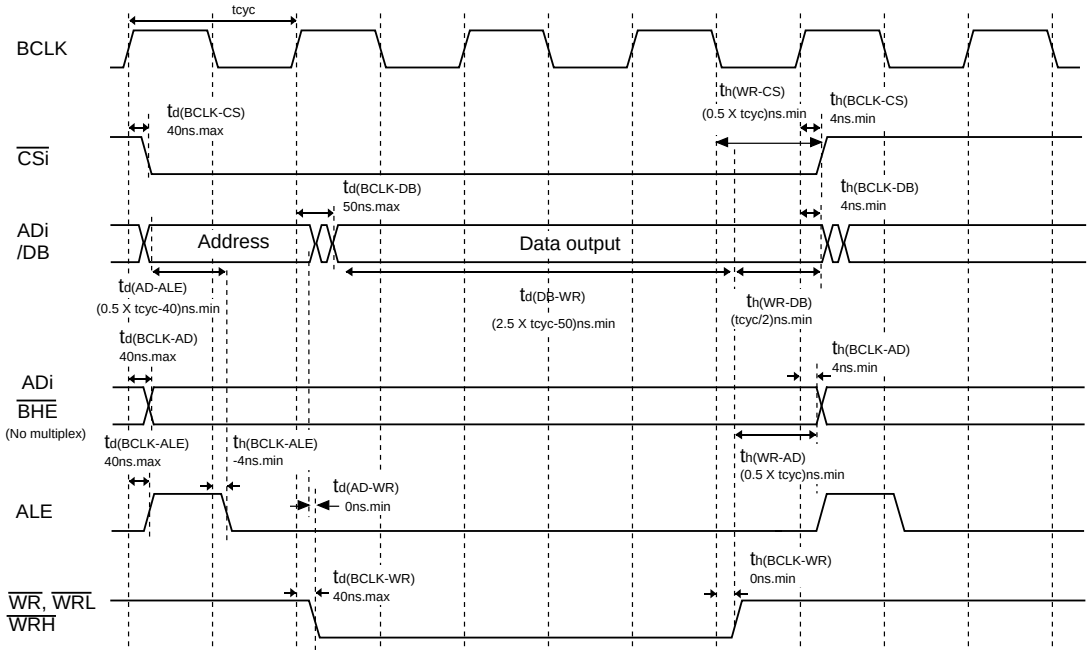
Memory Expansion Mode, Microprocessor Mode

(For 3-wait setting, external area access and multiplex bus selection)

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions
- V_{CC1}=V_{CC2}=3V
- Input timing voltage : V_{IL}=0.6V, V_{IH}=2.4V
- Output timing voltage : V_{OL}=1.5V, V_{OH}=1.5V

Figure 1.26.20. Timing Diagram (9)

Flash Memory Version

Flash Memory Performance

The flash memory version is functionally the same as the mask ROM version except that it internally contains flash memory.

The flash memory version has three modes—CPU rewrite, standard serial input/output, and parallel input/output modes—in which its internal flash memory can be operated on.

Table 1.27.1 shows the outline performance of flash memory version (see Table 1.1.1 for the items not listed in Table 1.27.1.).

Table 1.27.1. Flash Memory Version Specifications

Item		Specification
Flash memory operating mode		3 modes (CPU rewrite, standard serial I/O, parallel I/O)
Erase block	User ROM area	See Figure 1.27.1
	Boot ROM area	1 block (4 Kbytes) (Note 1)
Method for program		In units of word, in units of byte (Note 2)
Method for erasure		Collective erase, block erase
Program, erase control method		Program and erase controlled by software command
Protect method		Protected for each block by lock bit
Number of commands		8 commands
Number of program and erasure		100 times
Data Retention		10 years
ROM code protection		Parallel I/O and standard serial I/O modes are supported.

Note 1: The boot ROM area contains a standard serial I/O mode rewrite control program which is stored in it when shipped from the factory. This area can only be rewritten in parallel input/output mode.
Note 2: Can be programmed in byte units in only parallel input/output mode.

Table 1.27.2. Flash Memory Rewrite Modes Overview

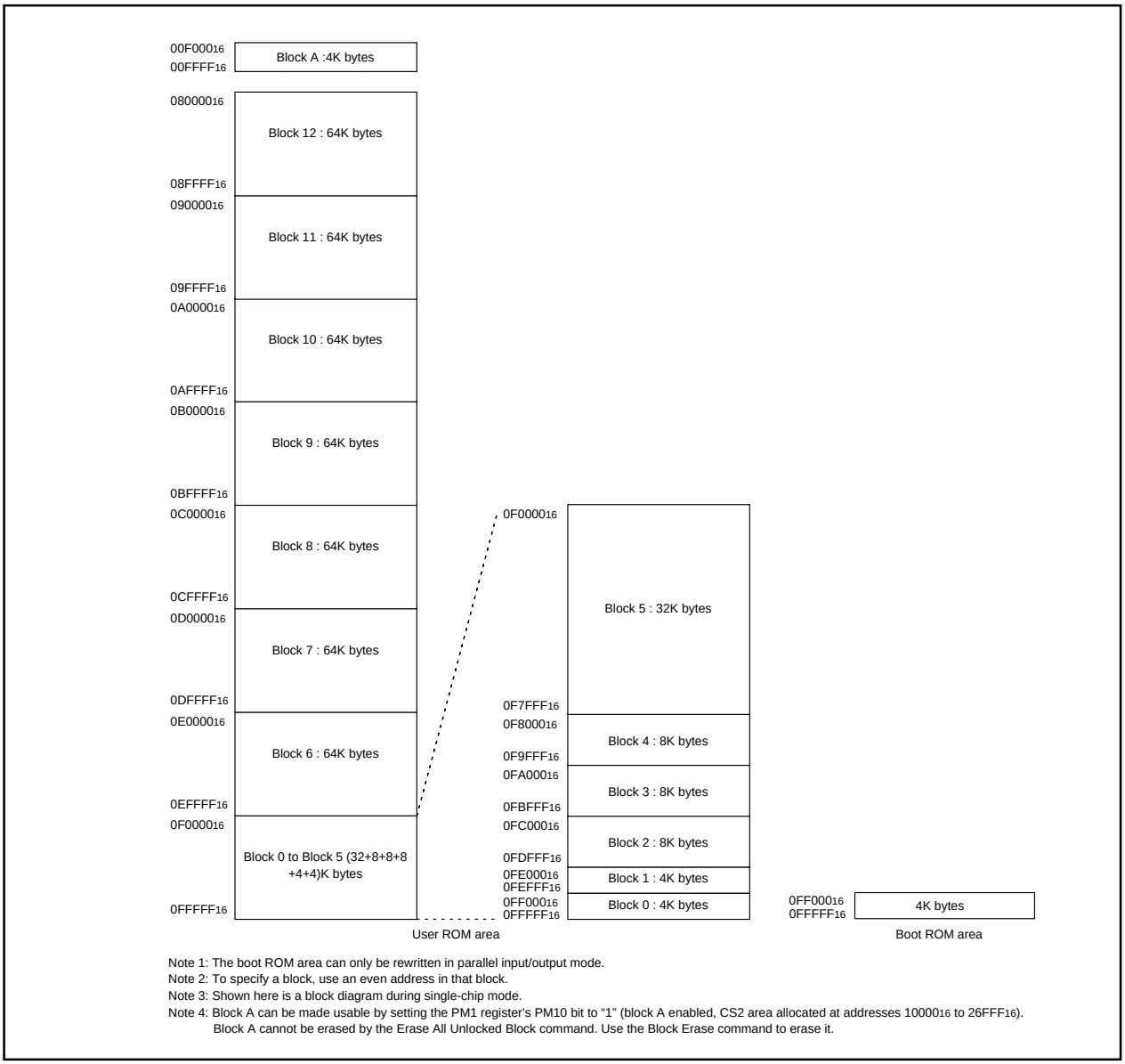
Flash memory rewrite mode	CPU rewrite mode	Standard serial I/O mode	Parallel I/O mode
Function	The user ROM area is rewritten by executing software commands from the CPU. EW0 mode: Can be rewritten in any area other than the flash memory EW1 mode: Can be rewritten in the flash memory	The user ROM area is rewritten by using a dedicated serial programmer. Standard serial I/O mode 1: Clock sync serial I/O Standard serial I/O mode 2: UART	The boot ROM and user ROM areas are rewritten by using a dedicated parallel programmer.
Areas which can be rewritten	User ROM area	User ROM area	User ROM area Boot ROM area
Operation mode	Single chip mode Memory expansion mode (EW0 mode) Boot mode (EW0 mode)	Boot mode	Parallel I/O mode
ROM programmer	None	Serial programmer	Parallel programmer

1. Memory Map

The ROM in the flash memory version is separated between a user ROM area and a boot ROM area. Figure 1.27.1 shows the block diagram of flash momoery. The user ROM area has a 4K-byte block A, in addition to the area that stores a program for microcomputer operation during singe-chip or memory expansion mode.

The user ROM area is divided into several blocks, each of which can individually be protected (locked) against programming or erasure. The user ROM area can be rewritten in all of CPU rewrite, standard serial input/output, and parallel input/output modes. Block A is enabled for use by setting the PM1 register's PM10 bit to "1" (block A enabled, CS2 area at addresses 10000_{16} to $26FFF_{16}$).

The boot ROM area is located at addresses that overlap the user ROM area, and can only be rewritten in parallel input/output mode. After a hardware reset that is performed by applying a high-level signal to the CNVSS and $P5_0$ pins and a low-level signal to the $P5_5$ pin, the program in the boot ROM area is executed. After a hardware reset that is performed by applying a low-level signal to the CNVSS pin, the program in the user ROM area is executed (but the boot ROM area cannot be read).



Note 1: The boot ROM area can only be rewritten in parallel input/output mode.
Note 2: To specify a block, use an even address in that block.
Note 3: Shown here is a block diagram during single-chip mode.
Note 4: Block A can be made usable by setting the PM1 register's PM10 bit to "1" (block A enabled, CS2 area allocated at addresses 10000_{16} to $26FFF_{16}$). Block A cannot be erased by the Erase All Unlocked Block command. Use the Block Erase command to erase it.

Figure 1.27.1. Flash Memory Block Diagram

Boot Mode

After a hardware reset which is performed by applying a low-level signal to the $P5_5$ pin and a high-level signal to the CNV_{SS} and $P5_0$ pins, the microcomputer is placed in boot mode, thereby executing the program in the boot ROM area.

During boot mode, the boot ROM and user ROM areas are switched over by the FMR05 bit in the FMR0 register.

The boot ROM area contains a standard serial input/output mode based rewrite control program which was stored in it when shipped from the factory.

The boot ROM area can be rewritten in parallel input/output mode. Prepare an EW0 mode based rewrite control program and write it in the boot ROM area, and the flash memory can be rewritten as suitable for the system.

Functions To Prevent Flash Memory from Rewriting

To prevent the flash memory from being read or rewritten easily, parallel input/output mode has a ROM code protect and standard serial input/output mode has an ID code check function.

• ROM Code Protect Function

The ROM code protect function inhibits the flash memory from being read or rewritten during parallel input/output mode. Figure 1.27.2 shows the ROMCP register.

The ROMCP register is located in the user ROM area.The ROMCP1 bit consists of two bits. The ROM code protect function is enabled by clearing one or both of two ROMCP1 bits to “0” when the ROMCR bits are not ‘00_2,’ with the flash memory thereby protected against reading or rewriting. Conversely, when the ROMCR bits are ‘00_2’ (ROM code protect removed), the flash memory can be read or rewritten. Once the ROM code protect function is enabled, the ROMCR bits cannot be changed during parallel input/output mode. Therefore, use standard serial input/output or other modes to rewrite the flash memory.

• ID Code Check Function

Use this function in standard serial input/output mode. Unless the flash memory is blank, the ID codes sent from the programmer and the ID codes written in the flash memory are compared to see if they match. If the ID codes do not match, the commands sent from the programmer are not accepted. The ID code consists of 8-bit data, the areas of which, beginning with the first byte, are $0FFFDF_{16}$, $0FFFE3_{16}$, $0FFFEB_{16}$, $0FFFEF_{16}$, $0FFFF3_{16}$, $0FFFF7_{16}$, and $0FFFFB_{16}$. Prepare a program in which the ID codes are preset at these addresses and write it in the flash memory.

ROM code protect control address

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

Symbol: ROMCP
Address: 0FFFFF16
Value when shipped: FF16 (Note 4)

Bit symbol	Bit name	Function	RW
———	Reserved bit	Set this bit to “1”	RW
———	Reserved bit	Set this bit to “1”	RW
———	Reserved bit	Set this bit to “1”	RW
———	Reserved bit	Set this bit to “1”	RW
ROMCR	ROM code protect reset bit (Note 2, Note 4)	b5 b4 00: Removes protect 01: } Enables ROOMCP1 bit 10: } Enables ROOMCP1 bit 11: } Enables ROOMCP1 bit	RW RW
ROMCP1	ROM code protect level 1 set bit (Note 1, Note 3, Note 4)	b7 b6 00: } Protect enabled 01: } Protect enabled 10: } Protect enabled 11: Protect disabled	RW RW

Note 1: If the ROMCR bits are set to other than ‘002’ and the ROMCP1 bits are set to other than ‘112’ (ROM code protect enabled), the flash memory is disabled against reading and rewriting in parallel input/output mode.
Note 2: If the ROMCR bits are set to ‘002’ when the ROMCR bits are other than ‘002’ and the ROMCP1 bits are other than ‘112,’ ROM code protect level 1 is removed. However, because the ROMCR bits cannot be modified during parallel input/output mode, they need to be modified in standard serial input/output or other modes.
Note 3: The ROMCP1 bits are effective when the ROMCR bits are ‘012,’ ‘102,’ or ‘112.’
Note 4: Once any of these bits is cleared to “0”, it cannot be set back to “1”. If a memory block that contains the ROMCP register is erased, the ROMCP register is set to ‘FF16.’

Figure 1.27.2. ROMCP Register

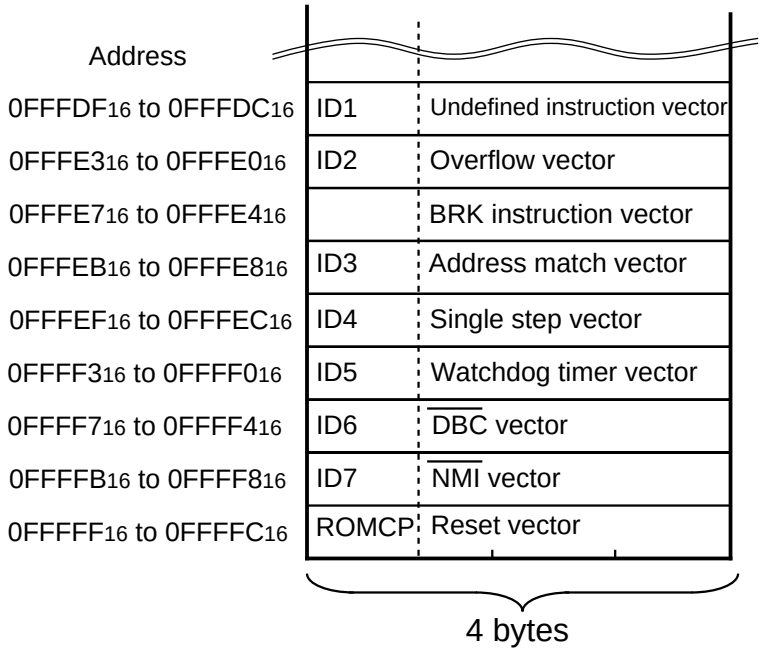


Figure 1.27.3. Address for ID Code Stored

CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten by executing software commands from the CPU. Therefore, the user ROM area can be rewritten directly while the microcomputer is mounted on-board without having to use a ROM programmer, etc.

In CPU rewrite mode, only the user ROM area shown in Figure 1.27.1 can be rewritten and the boot ROM area cannot be rewritten. Make sure the Program and the Block Erase commands are executed only on each block in the user ROM area.

During CPU rewrite mode, the user ROM area be operated on in either Erase Write 0 (EW0) mode or Erase Write 1 (EW1) mode. Table 1.27.3 lists the differences between Erase Write 0 (EW0) and Erase Write 1 (EW1) modes.

Table 1.27.3. EW0 Mode and EW1 Mode

Item	EW0 mode	EW1 mode
Operation mode	• Single chip mode • Memory expansion mode • Boot mode	Single chip mode
Areas in which a rewrite control program can be located	• User ROM area • Boot ROM area	User ROM area
Areas in which a rewrite control program can be executed	Must be transferred to any area other than the flash memory (e.g., RAM) before being executed	Can be executed directly in the user ROM area
Areas which can be rewritten	User ROM area	User ROM area However, this does not include the area in which a rewrite control program exists
Software command limitations	None	• Program, Block Erase command Cannot be executed on any block in which a rewrite control program exists • Erase All Unlocked Block command Cannot be executed when the lock bit for any block in which a rewrite control program exists is set to “1” (unlocked) or the FMR0 register’s FMR02 bit is set to “1” (lock bit disabled) • Read Status Register command Cannot be executed
Modes after Program or Erase	Read Status Register mode	Read Array mode
CPU status during Auto Write and Auto Erase	Operating	Hold state (I/O ports retain the state in which they were before the command was executed)(Note)
Flash memory status detection	• Read the FMR0 register's FMR00, FMR06, and FMR07 bits in a program • Execute the Read Status Register command to read the status register's SR7, SR5, and SR4 flags.	Read the FMR0 register's FMR00, FMR06, and FMR07 bits in a program

Note: Make sure no interrupts (except $\overline{\text{NMI}}$ and watchdog timer interrupts) and DMA transfers will occur.

- **EW0 Mode**

  The microcomputer is placed in CPU rewrite mode by setting the FMR0 register's FMR01 bit to "1" (CPU rewrite mode enabled), ready to accept commands. In this case, because the FMR1 register's FMR11 bit = 0, EW0 mode is selected. The FMR01 bit can be set to "1" by writing "0" and then "1" in succession.
  Use software commands to control program and erase operations. Read the FMR0 register or status register to check the status of program or erase operation at completion.

- **EW1 Mode**

  EW1 mode is selected by setting FMR11 bit to "1" (by writing "0" and then "1" in succession) after setting the FMR01 bit to "1" (by writing "0" and then "1" in succession).
  Read the FMR0 register to check the status of program or erase operation at completion. The status register cannot be read during EW1 mode.

Figure 1.27.4 shows the FIDR, FMR0 and FMR1 registers.

FMR00 Bit

This bit indicates the operating status of the flash memory. The bit is “0” when the Program, Erase, or Lock Bit program is running; otherwise, the bit is “1”.

FMR01 Bit

The microcomputer is made ready to accept commands by setting the FMR01 bit to “1” (CPU rewrite mode). During boot mode, make sure the FMR05 bit also is “1” (user ROM area access).

FMR02 Bit

The lock bit set for each block can be disabled by setting the FMR02 bit to “1” (lock bit disabled). (Refer to the description of the data protect function.) The lock bits set are enabled by setting the FMR02 bit to “0”. The FMR02 bit only disables the lock bit function and does not modify the lock bit data (lock bit status flag). However, if the Erase command is executed while the FMR02 bit is set to “1”, the lock bit data changes state from “0” (locked) to “1” (unlocked) after Erase is completed.

FMSTP Bit

This bit is provided for initializing the flash memory control circuits, as well as for reducing the amount of current consumed in the flash memory. The internal flash memory is disabled against access by setting the FMSTP bit to “1”. Therefore, the FMSTP bit must be written to by a program in other than the flash memory.

In the following cases, set the FMSTP bit to “1”:

- When flash memory access resulted in an error while erasing or programming in EW0 mode (FMR00 bit not reset to “1” (ready))
- When entering low power mode or ring low power mode

Figure 1.27.7 shows a flow chart to be followed before and after entering low power mode.

Note that when going to stop or wait mode, the FMR0 register does not need to be set because the power for the internal flash memory is automatically turned off and is turned back on again after returning from stop or wait mode.

FMR05 Bit

This bit switches between the boot ROM and user ROM areas during boot mode. Set this bit to “0” when accessing the boot ROM area (for read) or “1” (user ROM access) when accessing the user ROM area (for read, write, or erase).

FMR06 Bit

This is a read-only bit indicating the status of auto program operation. The bit is set to “1” when a program error occurs; otherwise, it is cleared to “0”. For details, tefer to the description of the full status check.

FMR07 Bit

This is a read-only bit indicating the status of auto erase operation. The bit is set to “1” when an erase error occurs; otherwise, it is cleared to “0”. For details, tefer to the description of the full status check.

Figure 1.27.5 and 1.27.6 show the setting and resetting of EW0 mode and EW1 mode, respectively.

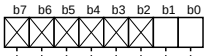
FMR11 Bit

Setting this bit to “1” places the microcomputer in EW1 mode.

FMR16 Bit

This is a read-only bit indicating the execution result of the Read Lock Bit Status command.

Flash identification register

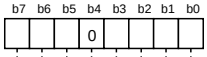


Symbol	Address	After reset
FIDR	$01B4_{16}$	$XXXXXX00_2$

Bit symbol	Bit name	Function	RW
FIDR0	Flash module type identification value	b1 b0 0 0: M16C/62N, M3062GF8N type flash module 1 0: M16C/62P type flash module 1 1: M16C/62M, M16C/62A type flash module	RO
FIDR1			RO
(b7-b2)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

Note: This register identifies on-chip flash module type of M16C/62 group. Note, however, no chip version is known by this register. Follow the procedure described below for the identification.
(1) Write FF_{16} to FIDR register
(2) Read FIDR register
(3) Check two low-order bits of read value
Make sure no access to external memories or other SFRs or no interrupts or DMA transfers will occur between the above two instructions no. 1 and no. 2.

Flash memory control register 0

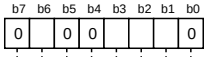


Symbol	Address	After reset
FMR0	$01B7_{16}$	$XX000001_2$

Bit symbol	Bit name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0: Busy (being written or erased) 1: Ready	RO
FMR01	CPU rewrite mode select bit (Note 1)	0: Disables CPU rewrite mode 1: Inables CPU rewrite mode	RW
FMR02	Lock bit disable select bit (Note 2)	0: Inables lock bit 1: Disables lock bit	RW
FMSTP	Flash memory stop bit (Note 3, Note 5))	0: Enables flash memory operation 1: Stops flash memory operation (placed in low power mode, flash memory initialized)	RW
(b4)	Reserved bit	Must always be set to “0”	RW
FMR05	User ROM area select bit (Note 3) (Effective in only boot mode)	0: Boot ROM area is accessed 1: User ROM area is accessed	RW
FMR06	Program status flag (Note 4)	0: Terminated normally 1: Terminated in error	RO
FMR07	Erase status flag (Note 4)	0: Terminated normally 1: Terminated in error	RO

Note 1: To set this bit to “1”, write “0” and then “1” in succession. Make sure no interrupts or DMA transfers will occur before writing “1” after writing “0”.
Write to this bit when the NMI pin is in the high state. Also, while in EW0 mode, write to this bit from a program in other than the flash memory.
Note 2: To set this bit to “1”, write “0” and then “1” in succession when the FMR01 bit = 1. Make sure no interrupts or no DMA transfers will occur before writing “1” after writing “0”.
Note 3: Write to this bit from a program in other than the flash memory.
Note 4: This flag is cleared to “0” by executing the Clear Status command.
Note 5: Effective when the FMR01 bit = 1 (CPU rewrite mode). If the FMR01 bit = 0, although the FMR03 bit can be set to “1” by writing “1” in a program, the flash memory is neither placed in low power mode nor initialized.
Note 6: This status includes writing or reading with the Lock Bit Program or Read Lock Bit Status command.

Flash memory control register 1

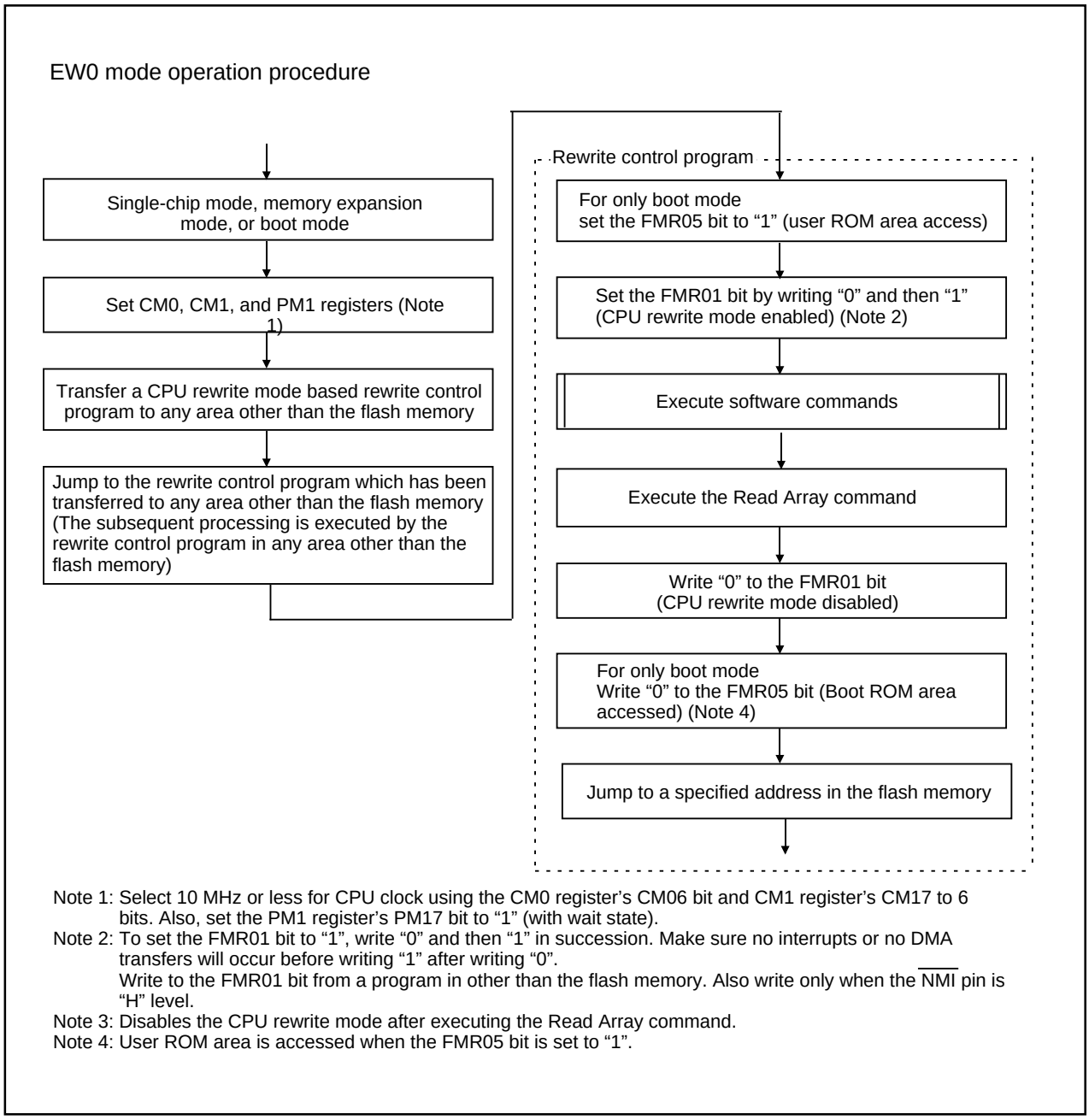


Symbol	Address	After reset
FMR1	$01B5_{16}$	$0X00XX0X_2$

Bit symbol	Bit name	Function	RW
(b0)	Reserved bit	The value in this bit when read is indeterminate.	RO
FMR11	EW1 mode select bit (Note)	0: EW0 mode 1: EW1 mode	RW
(b3-b2)	Reserved bit	The value in this bit when read is indeterminate.	RO
(b5-b4)	Reserved bit	Must always be set to “0”	RW
FMR06	Lock bit status flag	0: Lock 1: Unlock	RO
(b7)	Reserved bit	Must always be set to “0”	RW

Note : To set this bit to “1”, write “0” and then “1” in succession when the FMR01 bit = 1. Make sure no interrupts or no DMA transfers will occur before writing “1” after writing “0”.
The FMR01 and FMR11 bits both are cleared to “0” by setting the FMR01 bit to “0”.

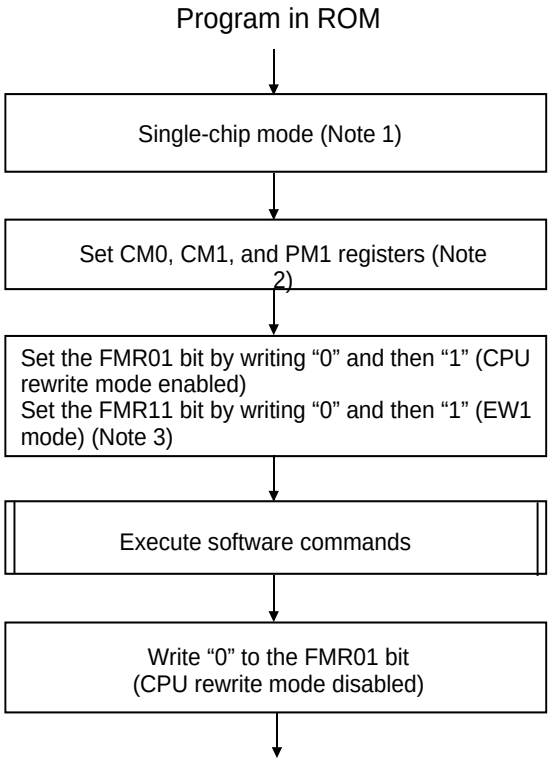
Figure 1.27.4. FIDR Register and FMR0 and FMR1 Registers



Note 1: Select 10 MHz or less for CPU clock using the CM0 register's CM06 bit and CM1 register's CM17 to 6 bits. Also, set the PM1 register's PM17 bit to "1" (with wait state).
Note 2: To set the FMR01 bit to "1", write "0" and then "1" in succession. Make sure no interrupts or no DMA transfers will occur before writing "1" after writing "0".
Write to the FMR01 bit from a program in other than the flash memory. Also write only when the $\overline{\text{NMI}}$ pin is "H" level.
Note 3: Disables the CPU rewrite mode after executing the Read Array command.
Note 4: User ROM area is accessed when the FMR05 bit is set to "1".

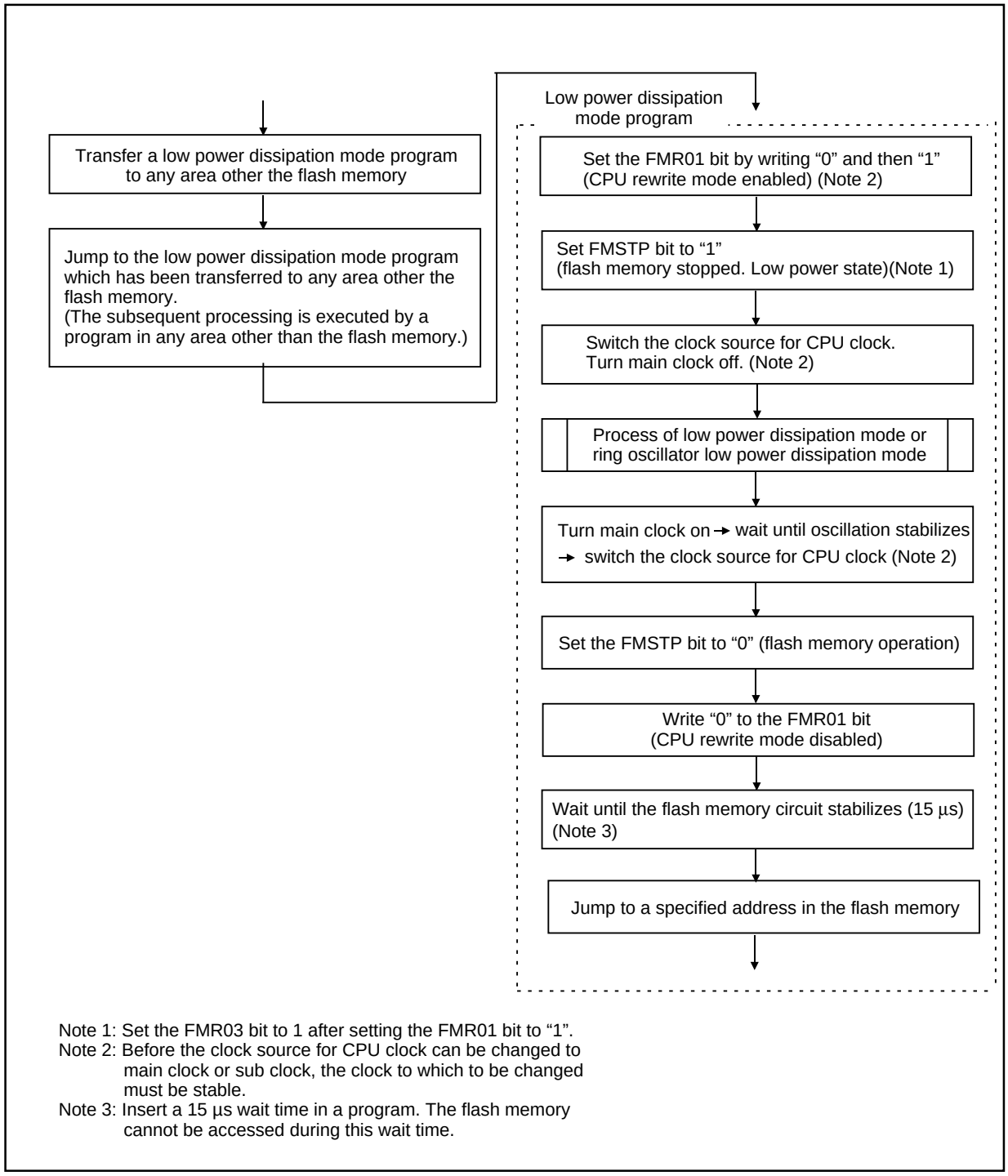
Figure 1.27.5. Setting and Tesetting of EW0 Mode

EW1 mode operation procedure



Note 1: In EW1 mode, do not set the microcomputer in memory expansion or boot mode.
Note 2: Select 10 MHz or less for CPU clock using the CM0 register’s CM06 bit and CM1 register’s CM17 to 6 bits. Also, set the PM1 register’s PM17 bit to “1” (with wait state).
Note 2: To set the FMR01 bit to “1”, write “0” and then “1” in succession. Make sure no interrupts or no DMA transfers will occur before writing “1” after writing “0”.
Write to the FMR01 bit from a program in other than the flash memory. Also write only when the NMI pin is “H” level.

Figure 1.27.6. Setting and Resetting of EW1 Mode



Note 1: Set the FMR03 bit to 1 after setting the FMR01 bit to “1”.
Note 2: Before the clock source for CPU clock can be changed to main clock or sub clock, the clock to which to be changed must be stable.
Note 3: Insert a 15 µs wait time in a program. The flash memory cannot be accessed during this wait time.

Figure 1.27.7. Processing Before and After Low Power Sissipation Mode

Precautions on CPU Rewrite Mode

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

(1) Operation Speed

Before entering CPU rewrite mode (EW0 or EW1 mode), select 10 MHz or less for BCLK using the CM06 bit in the CM0 register and the CM17 to CM16 bits in the CM1 register. Also, set the PM17 bit in the PM1 register to “1” (with wait state).

(2) Instructions to Prevent from Using

The following instructions cannot be used in EW0 mode because the flash memory’s internal data is referenced: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

(3) Interrupts

EW0 Mode

- Any interrupt which has a vector in the variable vector table can be used providing that its vector is transferred into the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts can be used because the FMR0 register and FMR1 register are initialized when one of those interrupts occurs. The jump addresses for those interrupt service routines should be set in the fixed vector table.

  Because the rewrite operation is halted when a $\overline{\text{NMI}}$ or watchdog timer interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.
- The address match interrupt cannot be used because the flash memory’s internal data is referenced.

EW1 Mode

- Make sure that any interrupt which has a vector in the variable vector table or address match interrupt will not be accepted during the auto program or auto erase period.
- Avoid using watchdog timer interrupts.
- The $\overline{\text{NMI}}$ interrupt can be used because the FMR0 register and FMR1 register are initialized when this interrupt occurs. The jump address for the interrupt service routine should be set in the fixed vector table.

  Because the rewrite operation is halted when a $\overline{\text{NMI}}$ interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

(4) How to Access

To set the FMR01, FMR02, or FMR11 bit to “1”, write “0” and then “1” in succession. This is necessary to ensure that no interrupts or DMA transfers will occur before writing “1” after writing “0”. Also only when $\overline{\text{NMI}}$ pin is “H” level.

(5) Writing in the User ROM Space

EW0 Mode

- If the power supply voltage drops while rewriting any block in which the rewrite control program is stored, a problem may occur that the rewrite control program is not correctly rewritten and, consequently, the flash memory becomes unable to be rewritten thereafter. In this case, standard serial I/O or parallel I/O mode should be used.

EW1 Mode

- Avoid rewriting any block in which the rewrite control program is stored.

(6) DMA Transfer

In EW1 mode, make sure that no DMA transfers will occur while the FMR0 register's FMR00 bit = 0 (during the auto program or auto erase period).

(7) Writing Command and Data

Write the command code and data at even addresses.

(8) Wait Mode

When shifting to wait mode, set the FMR01 bit to "0" (CPU rewrite mode disabled) before executing the WAIT instruction.

(9) Stop Mode

When shifting to stop mode, the following settings are required:

• Set the FMR01 bit to "0" (CPU rewrite mode disabled) and disable DMA transfers before setting the CM10 bit to "1" (stop mode).

• Execute the JMP.B instruction subsequent to the instruction which sets the CM10 bit to "1" (stop mode)

```
Example program    BSET    0, CM1    ; Stop mode
                   JMP.B   L1
               L1:
                   Program after returning from stop mode
```

(10) Low Power Dissipation Mode and Ring Oscillator Low Power Dissipation Mode

If the CM05 bit is set to "1" (main clock stop), the following commands must not be executed.

• Program
• Block erase
• Erase all unlocked blocks
• Lock bit program

Software Commands

Software commands are described below. The command code and data must be read and written in 16-bit units, to and from even addresses in the user ROM area. When writing command code, the 8 high-order bits (D_{1t}–D_8) are ignored.

Table 1.27.4. Software Commands

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read array	Write	X	xxFF16			
Read status register	Write	X	xx7016	Read	X	SRD
Clear status register	Write	X	xx5016			
Program	Write	WA	xx4016	Write	WA	WD
Block erase	Write	X	xx2016	Write	BA	xxD016
Erase all unlocked block(Note)	Write	X	xxA716	Write	X	xxD016
Lock bit program	Write	BA	xx7716	Write	BA	xxD016
Read lock bit status	Write	X	xx7116	Write	BA	xxD016

Note: It is only blocks 0 to 12 that can be erased by the Erase All Unlocked Block command.
Block A cannot be erased. Use the Block Erase command to erase block A.
SRD: Status register data (D_7 to D_0)
WA: Write address (Make sure the address value specified in the the first bus cycle is the same even address as the write address specified in the second bus cycle.)
WD: Write data (16 bits)
BA: Uppermost block address (even address, however)
X: Any even address in the user ROM area
x: High-order 8 bits of command code (ignored)

Read Array Command (FF16)

This command reads the flash memory.

Writing 'xxFF16' in the first bus cycle places the microcomputer in read array mode. Enter the read address in the next or subsequent bus cycles, and the content of the specified address can be read in 16-bit units.

Because the microcomputer remains in read array mode until another command is written, the contents of multiple addresses can be read in succession.

Read Status Register Command (7016)

This command reads the status register.

Write 'xx7016' in the first bus cycle, and the status register can be read in the second bus cycle. (Refer to "Status Register.") When reading the status register too, specify an even address in the user ROM area.

Do not execute this command in EW1 mode.

Clear Status Register Command (50_{16})

This command clears the status register to “0”.

Write ‘xx50_{16}’ in the first bus cycle, and the FMR06 to FMR07 bits in the FMR0 register and SR4 to SR5 in the status register will be cleared to “0”.

Program Command (40_{16})

This command writes data to the flash memory in 1 word (2 byte) units.

Write ‘xx40_{16}’ in the first bus cycle and write data to the write address in the second bus cycle, and an auto program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same even address as the write address specified in the second bus cycle.

Check the FMR00 bit in the FMR0 register to see if auto programming has finished. The FMR00 bit is “0” during auto programming and set to “1” when auto programming is completed.

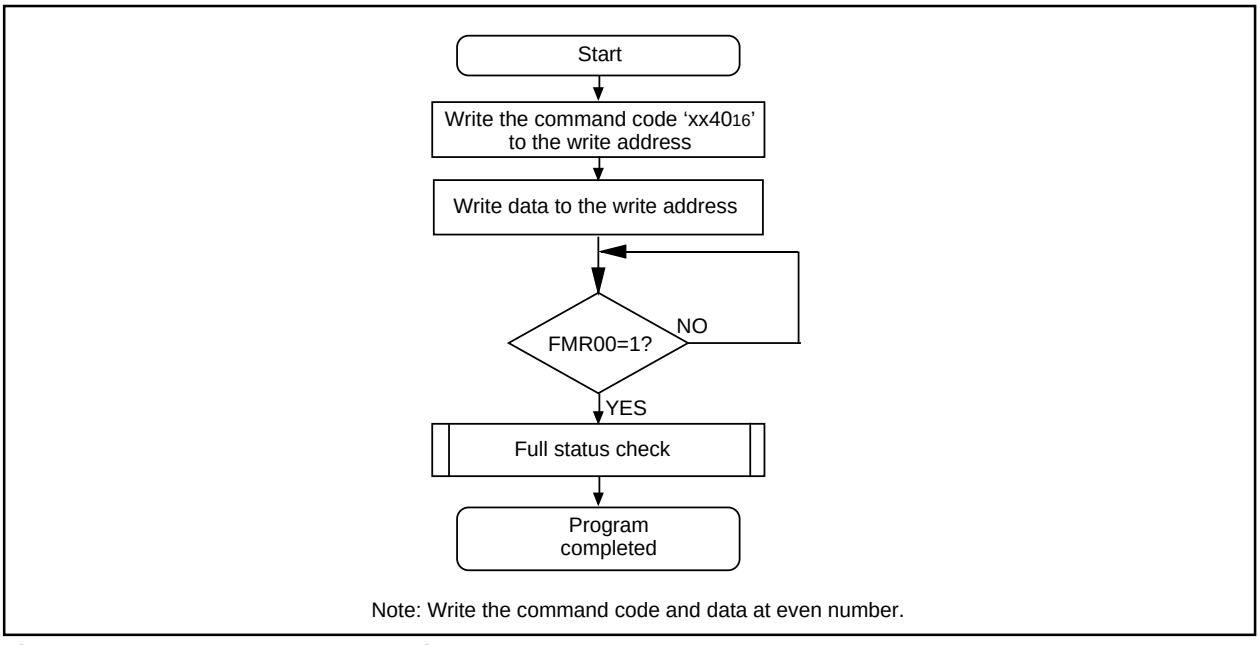
Check the FMR06 bit in the FMR0 register after auto programming has finished, and the result of auto programming can be known. (Refer to “Full Status Check.”)

Each block can be protected against programming by a lock bit. (Refer to “Data Protect Function.”)

Writing over already programmed addresses is inhibited.

In EW1 mode, do not execute this command on any address at which the rewrite control program is located.

In EW0 mode, the microcomputer goes to read status register mode at the same time auto programming starts, making it possible to read the status register. The status register bit 7 (SR7) is cleared to “0” at the same time auto programming starts, and set back to “1” when auto programming finishes. In this case, the microcomputer remains in read status register mode until a read command is written next. The result of auto programming can be known by reading the status register after auto programming has finished.

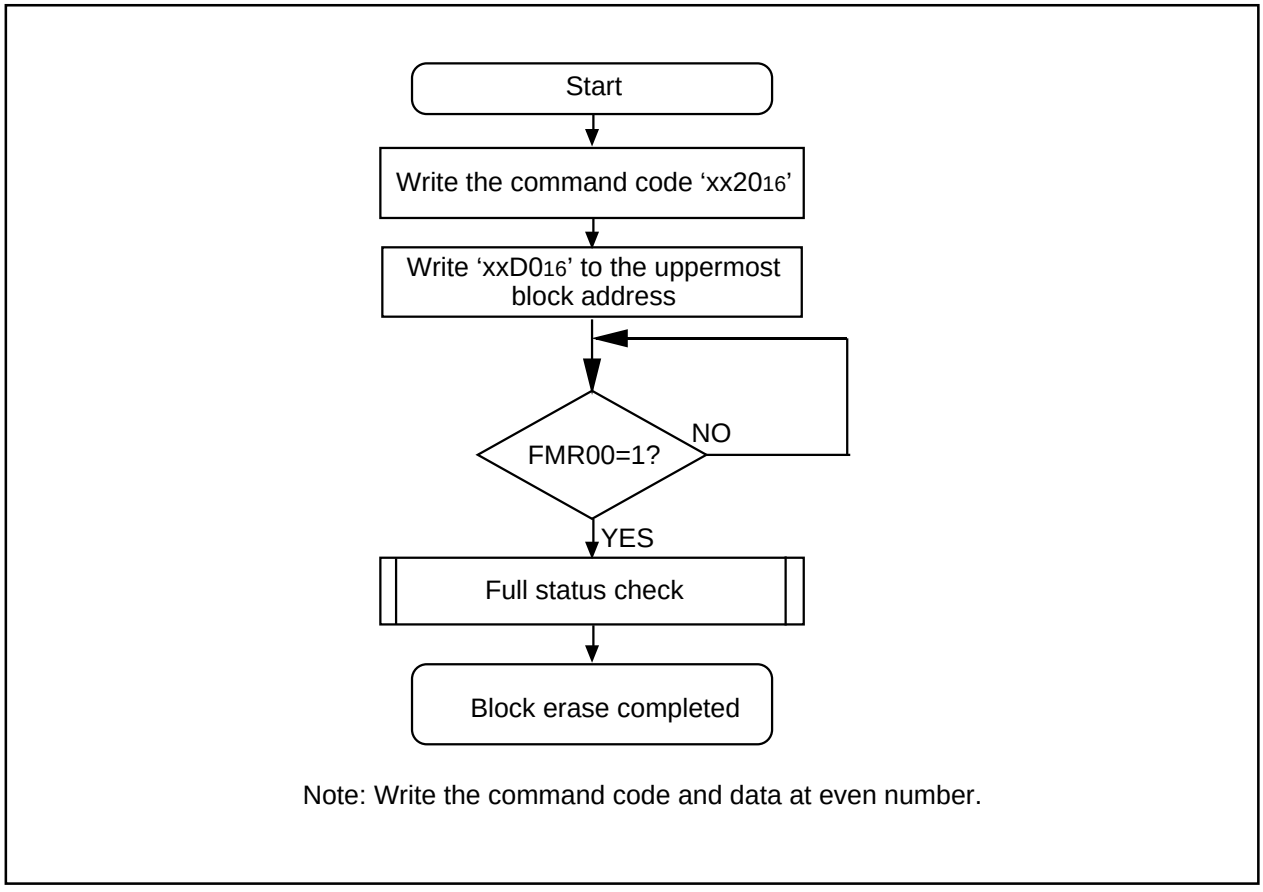


Note: Write the command code and data at even number.

Figure 1.27.8. Program Command

Block Erase

Write ‘xx20$_{16}$’ in the first bus cycle and write ‘xxD0$_{16}$’ to the uppermost address of a block (even address, however) in the second bus cycle, and an auto erase operation (erase and verify) will start.
Check the FMR0 register’s FMR00 bit to see if auto erasing has finished.
The FMR00 bit is “0” during auto erasing and set to “1” when auto erasiing is completed.
Check the FMR0 register’s FMR07 bit after auto erasing has finished, and the result of auto erasing can be known. (Refer to “Full Status Check.”)
Figure 1.27.9 shows an example of a block erase flowchart.
Each block can be protected against erasing by a lock bit. (Refer to “Data Protect Function.”)
Writing over already programmed addresses is inhibited.
In EW1 mode, do not execute this command on any address at which the rewrite control program is located.
In EW0 mode, the microcomputer goes to read status register mode at the same time auto erasing starts, making it possible to read the status register. The status register bit 7 (SR7) is cleared to “0” at the same time auto erasing starts, and set back to “1” when auto erasing finishes. In this case, the microcomputer remains in read status register mode until the Read Array or Read Lock Bit Status command is written next.



Note: Write the command code and data at even number.

Figure 1.27.9. Block Erase Command

Erase All Unlocked Block

Write 'xxA7$_{16}$' in the first bus cycle and write 'xxD0$_{16}$' in the second bus cycle, and all blocks except block A will be erased successively, one block at a time.

Check the FMR0 register's FMR00 bit to see if auto erasing has finished. The result of the auto erase operation can be known by inspecting the FMR0 register's FMR07 bit.

Each block can be protected against erasing by a lock bit. (Refer to "Data Protect Function.")

In EW1 mode, do not execute this command when the lock bit for any block = 1 (unlocked) in which the rewrite control program is stored, or when the FMR0 register's FMR02 bit = 1 (lock bit disabled).

In EW0 mode, the microcomputer goes to read status register mode at the same time auto erasing starts, making it possible to read the status register. The status register bit 7 (SR7) is cleared to "0" at the same time auto erasing starts, and set back to "1" when auto erasing finishes. In this case, the microcomputer remains in read status register mode until the Read Array or Read Lock Bit Status command is written next.

Note that only blocks 0 to 12 can be erased by the Erase All Unlocked Block command. Block A cannot be erased. Use the Block Erase command to erase block A.

Lock Bit Program Command (77$_{16}$/D0$_{16}$)

This command sets the lock bit for a specified block to "0" (locked).

Write 'xx77$_{16}$' in the first bus cycle and write 'xxD0$_{16}$' to the uppermost address of a block (even address, however) in the second bus cycle, and the lock bit for the specified block is cleared to "0". Make sure the address value specified in the first bus cycle is the same uppermost block address that is specified in the second bus cycle.

Figure 1.27.10 shows an example of a lock bit program flowchart. The lock bit status (lock bit data) can be read using the Read Lock Bit Status command.

Check the FMR0 register's FMR00 bit to see if writing has finished.

For details about the lock bit function, and on how to set the lock bit to "1", refer to "Data Protect Function."

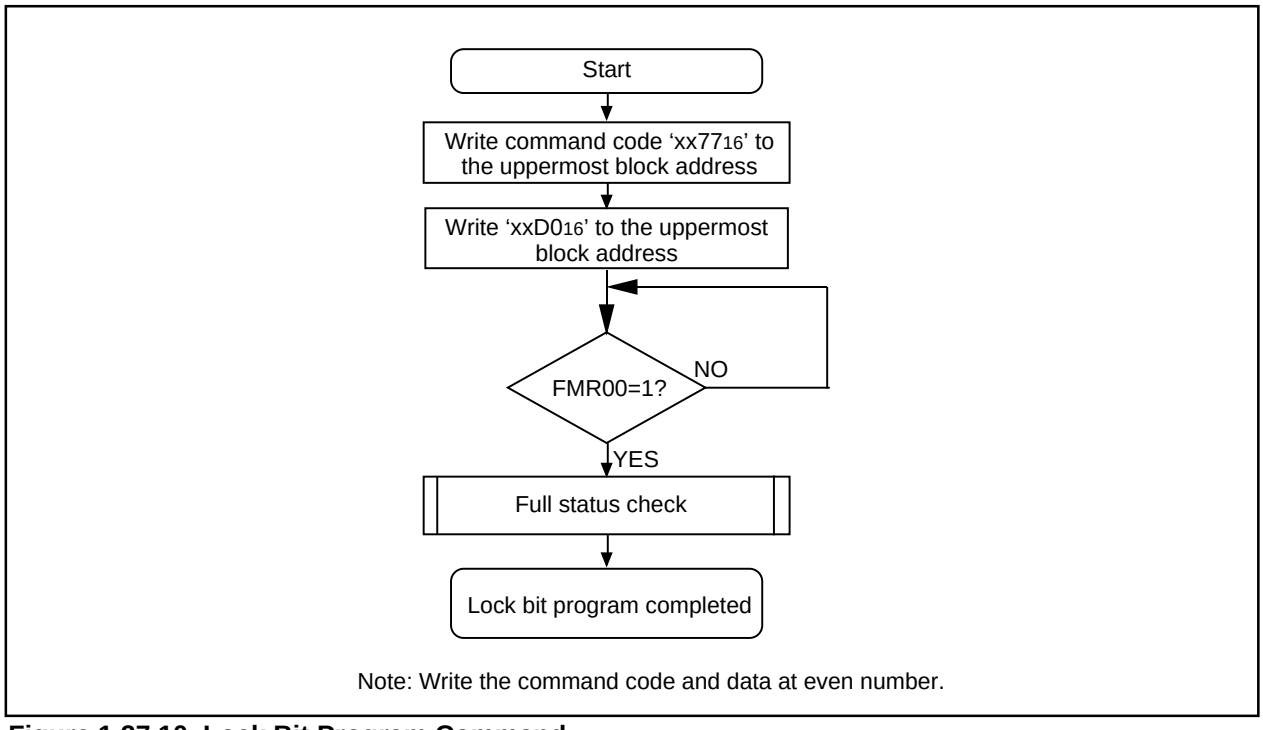


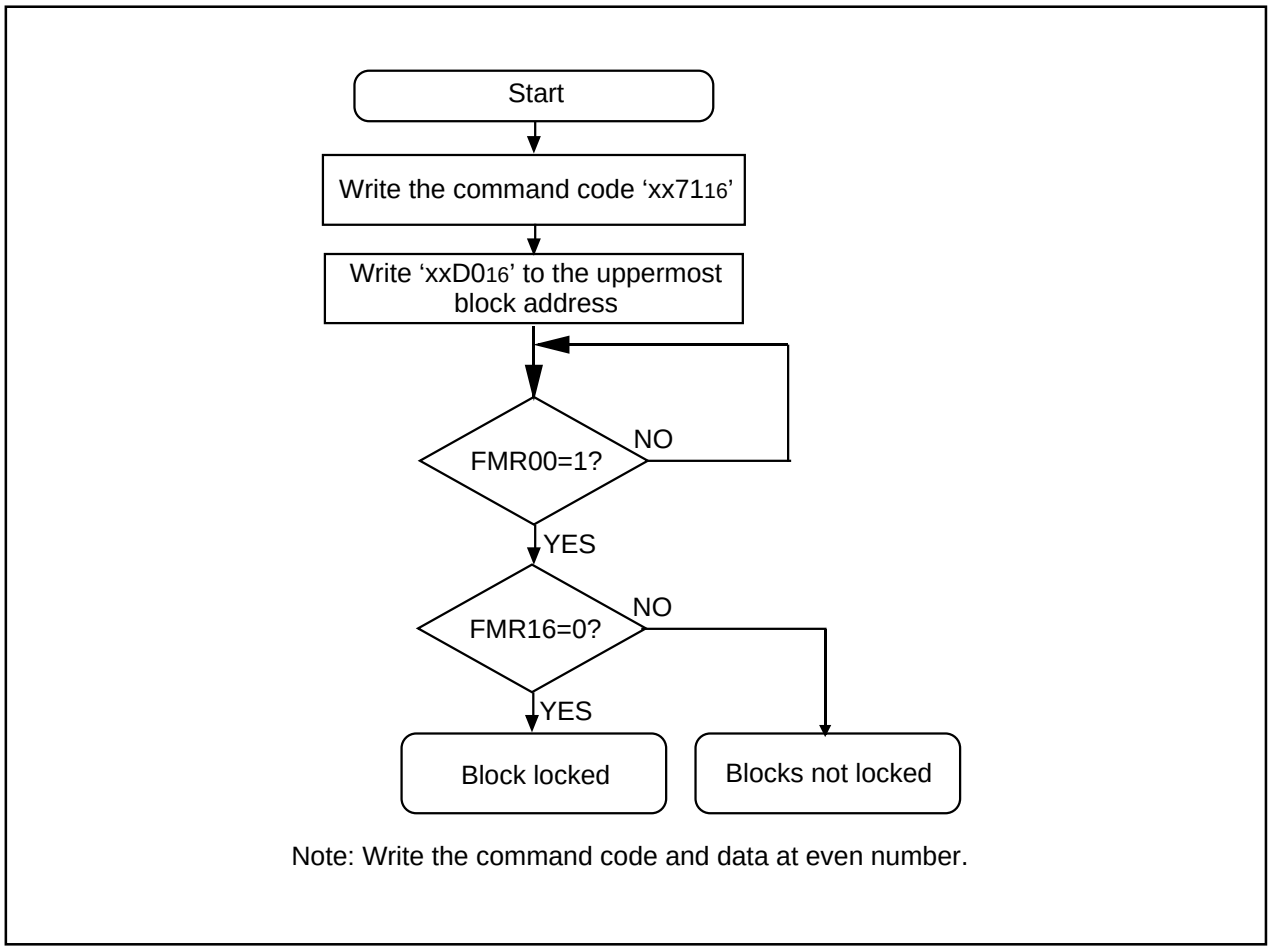
Figure 1.27.10. Lock Bit Program Command

Read Lock Bit Status Command (71$_{16}$)

This command reads the lock bit status of a specified block.

Write 'xx71$_{16}$' in the first bus cycle and write 'xxD0$_{16}$' to the uppermost address of a block (even address, however) in the second bus cycle, and the lock bit status of the specified block is stored in the FMR1 register's FMR16 bit. Read the FMR16 bit after the FMR0 register's FMR00 bit is set to "1" (ready).

Figure 1.27.11 shows an example of a read lock bit status flowchart.



Note: Write the command code and data at even number.

Figure 1.27.11. Read Lock Bit Status Command

Data Protect Function

Each block in the flash memory has a nonvolatile lock bit. The lock bit is effective when the FMR02 bit = 0 (lock bit enabled). The lock bit allows each block to be individually protected (locked) against programming and erasure. This helps to prevent data from inadvertently written to or erased from the flash memory. The following shows the relationship between the lock bit and the block status.

- When the lock bit = 0, the block is locked (protected against programming and erasure).
- When the lock bit = 1, the block is not locked (can be programmed or erased.

The lock bit is cleared to “0” (locked) by executing the Lock Bit Program command, and is set to “1” (unlocked) by erasing the block. The lock bit cannot be set to “1” by a command.
The lock bit status can be read using the Read Lock Bit Status command

The lock bit function is disabled by setting the FMR02 bit to “1”, with all blocks placed in an unlocked state. (The lock bit data itself does not change state.) Setting the FMR02 bit to “0” enables the lock bit function (lock bit data retained).
If the Block Erase or Erase All Unlocked Block command is executed while the FMR02 bit = 1, the target block or all blocks are erased irrespective of how the lock bit is set. The lock bit for each block is set to “1” after completion of erasure.
For details about the commands, refer to “Software Commands.”

Status Register

The status register indicates the operating status of the flash memory and whether an erase or programming operation terminated normally or in error. The status of the status register can be known by reading the FMR0 register’s FMR00, FMR06, and FMR07 bits.
Table 1.27.5 shows the status register.
In EW0 mode, the status register can be read in the following cases:

(1) When a given even address in the user ROM area is read after writing the Read Status Register command
(2) When a given even address in the user ROM area is read after executing the Program, Block Erase, Erase All Unlocked Block, or Lock Bit Program command but before executing the Read Array command.

Sequencer Status (SR7 and FMR00 Bits)

The sequence status indicates the operating status of the flash memory. SR7 = 0 (busy) during auto programming, auto erase, and lock bit write, and is set to “1” (ready) at the same time the operation finishes.

Erase Status (SR5 and FMR07 Bits)

Refer to “Full Status Check.”

Program Status (SR4 and FMR06 Bits)

Refer to “Full Status Check.”

Table 1.27.5. Status Register

Status register bit	FMR0 register bit	Status name	Contents		Value after reset
			"0"	"1"	
SR7 (D_7)	FMR00	Sequencer status	Busy	Ready	1
SR6 (D_6)	——	Reserved	-	-	——
SR5 (D_5)	FMR07	Erase status	Terminated normally	Terminated in error	0
SR4 (D_4)	FMR06	Program status	Terminated normally	Terminated in error	0
SR3 (D_3)	——	Reserved	-	-	——
SR2 (D_2)	——	Reserved	-	-	——
SR1 (D_1)	——	Reserved	-	-	——
SR0 (D_0)	——	Reserved	-	-	——

• D_0 to D_7: Indicates the data bus which is read out when the Read Status Register command is executed.

• The FMR07 bit (SR5) and FMR06 bit (SR4) are cleared to “0” by executing the Clear Status Register command.

• When the FMR07 bit (SR5) or FMR06 bit (SR4) = 1, the Program, Block Erase, Erase All Unlocked Block, and Lock Bit Program commands are not accepted.

Full Status Check

When an error occurs, the FMR0 register's FMR06 to FMR07 bits are set to "1", indicating occurrence of each specific error. Therefore, execution results can be verified by checking these status bits (full status check). Table 1.27.6 lists errors and FMR0 register status. Figure 1.27.12 shows a full status check flowchart and the action to be taken when each error occurs.

Table 1.27.6. Errors and FMR0 Register Status

FRM00 register (status register) status		Error	Error occurance condition
FMR07 (SR5)	FMR06 (SR4)		
1	1	Command sequence error	• When any command is not written correctly • When invalid data was written other than those that can be written in the second bus cycle of the Lock Bit Program, Block Erase, or Erase All Unlocked Block command (i.e., other than 'xxD0$_{16}$' or 'xxFF$_{16}$') (Note 1)
1	0	Erase error	• When the Block Erase command was executed on locked blocks (Note 2) • When the Block Erase or Erase All Unlocked Block command was executed on unlocked blocks but the blocks were not automatically erased correctly
0	1	Program error	• When the Block Erase command was executed on locked blocks (Note 2) • When the Program command was executed on unlocked blocks but the blocks were not automatically programmed correctly. • When the Lock Bit Program command was executed but not programmed correctly

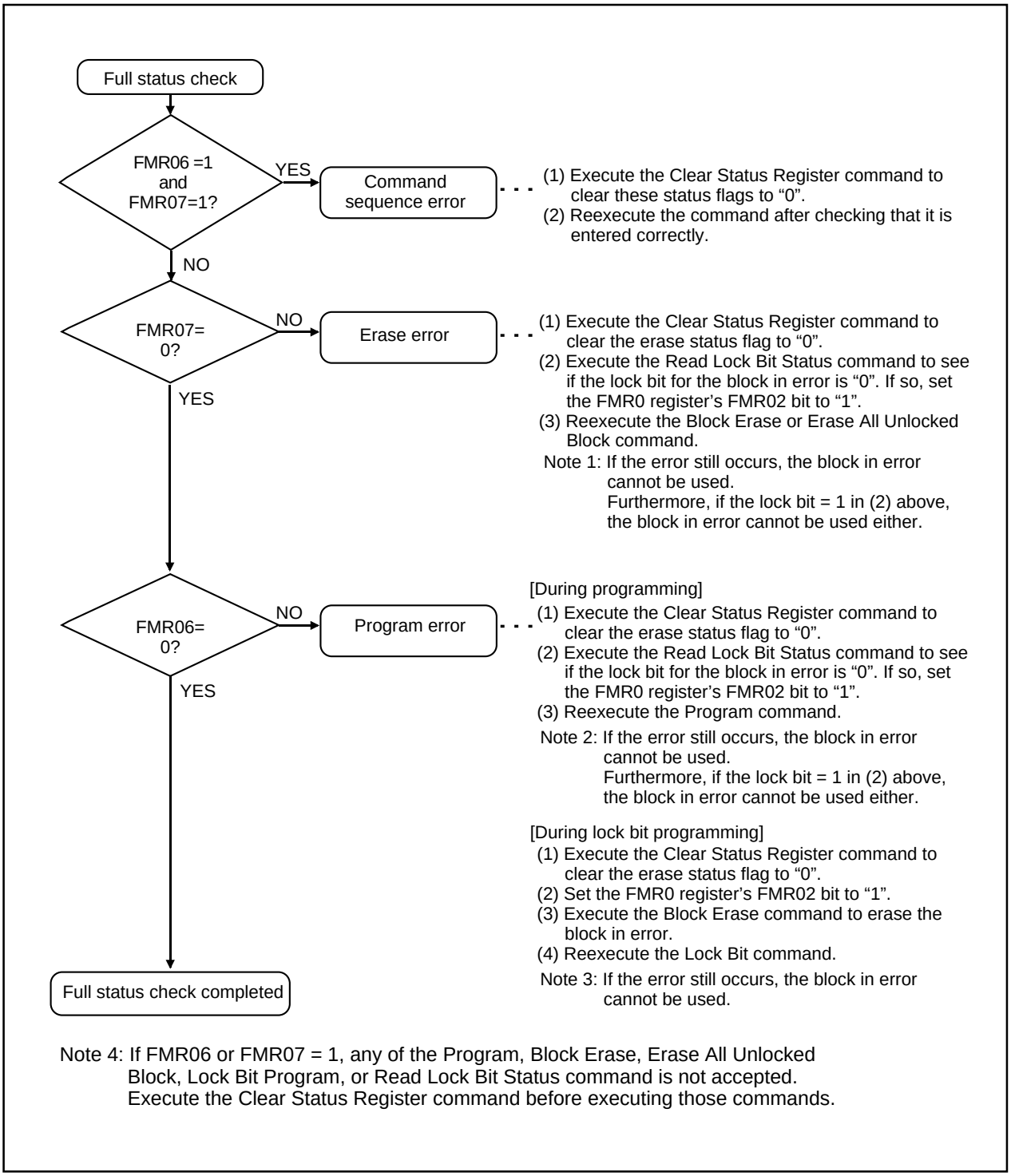


Figure 1.27.12. Full Status Check and Handling Procedure for Each Error

Standard Serial I/O Mode

In standard serial input/output mode, the user ROM area can be rewritten while the microcomputer is mounted on-board by using a serial programmer suitable for the M16C/62P group. For more information about serial programmers, contact the manufacturer of your serial programmer. For details on how to use, refer to the user's manual included with your serial programmer.

Table 1.27.7 lists pin functions (flash memory standard serial input/output mode). Figures 1.27.13 to 1.27.15 show pin connections for serial input/output mode.

ID Code Check Function

This function determines whether the ID codes sent from the serial programmer and those written in the flash memory match. (Refer to the desctiption of the functions to inhibit rewriting flash memory version.)

Table 1.27.7. Pin Functions (Flash Memory Standard Serial I/O Mode)

Pin	Name	I/O	Description
VCC,VSS	Power input		Apply the voltage guaranteed for Program and Erase to Vcc pin and 0 V to Vss pin.
CNVSS	CNVSS	I	Connect to Vcc pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While $\overline{\text{RESET}}$ pin is "L" level, input a 20 cycle or longer clock to XIN pin.
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock output	O	
BYTE	BYTE	I	Connect this pin to Vcc or Vss.
AVCC, AVSS	Analog power supply input		Connect AVss to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for AD from this pin.
P00 to P07	Input port P0	I	Input "H" or "L" level signal or open.
P10 to P17	Input port P1	I	Input "H" or "L" level signal or open.
P20 to P27	Input port P2	I	Input "H" or "L" level signal or open.
P30 to P37	Input port P3	I	Input "H" or "L" level signal or open.
P40 to P47	Input port P4	I	Input "H" or "L" level signal or open.
P51 to P54, P56, P57	Input port P5	I	Input "H" or "L" level signal or open.
P50	$\overline{\text{CE}}$ input	I	Input "H" level signal.
P55	$\overline{\text{EPM}}$ input	I	Input "L" level signal.
P60 to P63	Input port P6	I	Input "H" or "L" level signal or open.
P64	BUSY output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitors the boot program operation check signal output pin.
P65	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Input "L".
P66	RxD input	I	Serial data input pin
P67	TxD output	O	Serial data output pin (Note 1)
P70 to P77	Input port P7	I	Input "H" or "L" level signal or open.
P80 to P84, P86, P87	Input port P8	I	Input "H" or "L" level signal or open.
P85	$\overline{\text{NMI}}$ input	I	Connect this pin to Vcc.
P90 to P97	Input port P9	I	Input "H" or "L" level signal or open. (Note 2)
P100 to P107	Input port P10	I	Input "H" or "L" level signal or open. (Note 2)
P110 to P117	Input port P11	I	Input "H" or "L" level signal or open. (Note 2)
P120 to P127	Input port P12	I	Input "H" or "L" level signal or open. (Note 2)
P130 to P137	Input port P13	I	Input "H" or "L" level signal or open. (Note 2)
P140 to P147	Input port P14	I	Input "H" or "L" level signal or open. (Note 2)

Note 1: When using standard serial input/output mode 1, the TxD pin must be held high while the $\overline{\text{RESET}}$ pin is pulled low. Therefore, connect this pin to VCC via a resistor. Because this pin is directed for data output after reset, adjust the pull-up resistance value in the system so that data transfers will not be affected.

Note 2: Available in only the 128-pin version.

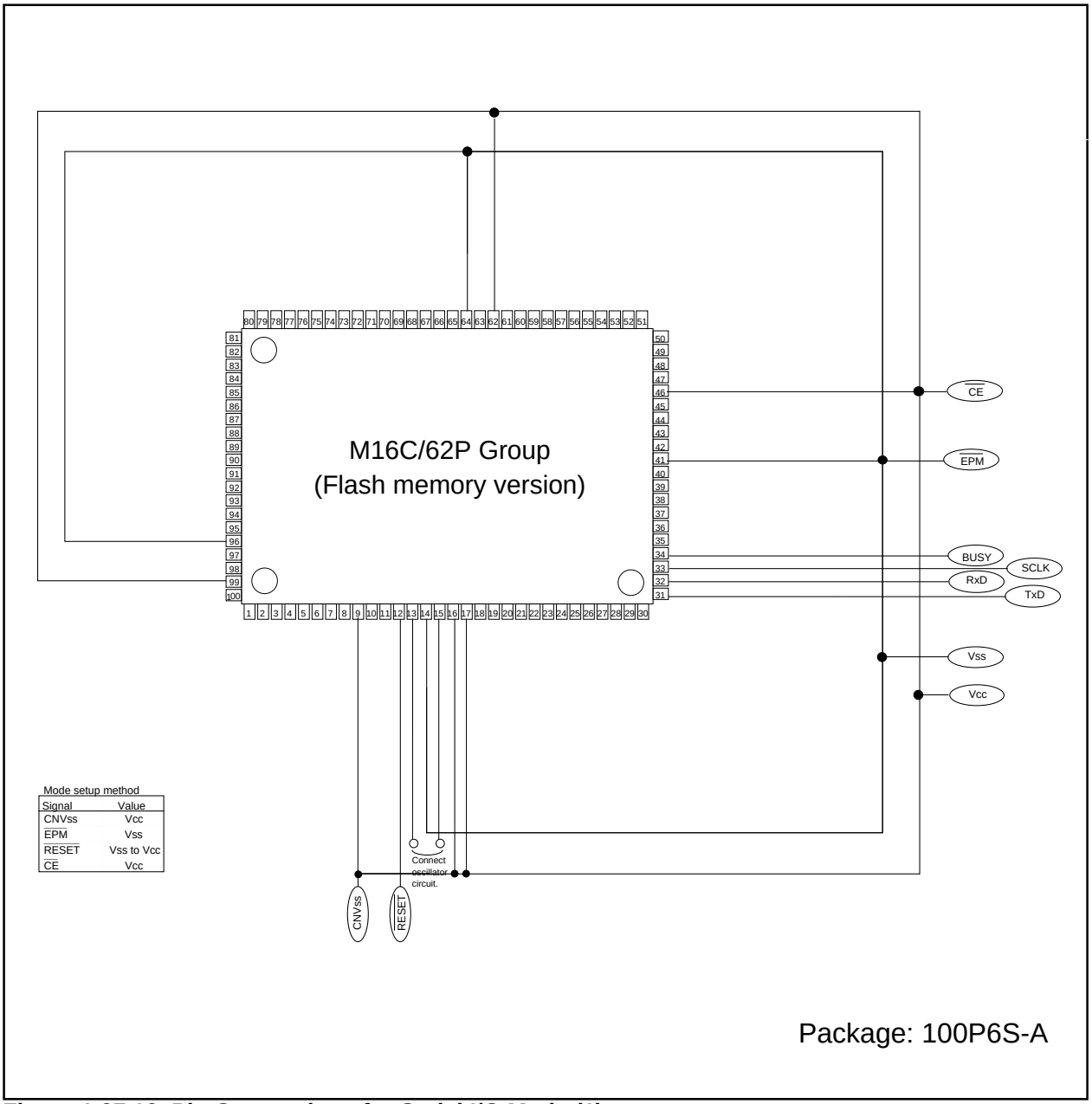


Figure 1.27.13. Pin Connections for Serial I/O Mode (1)

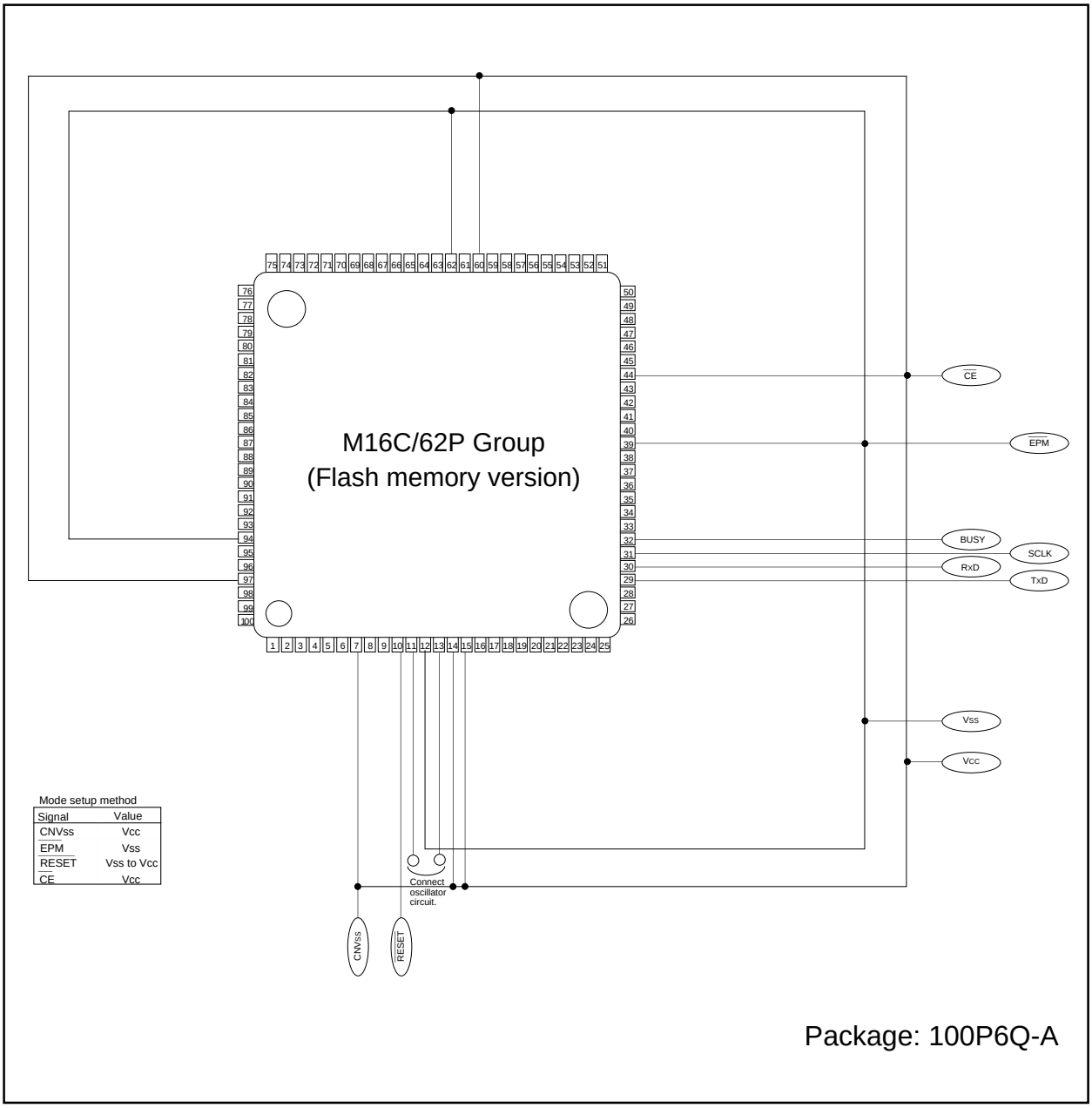


Figure 1.27.14. Pin Connections for Serial I/O Mode (2)

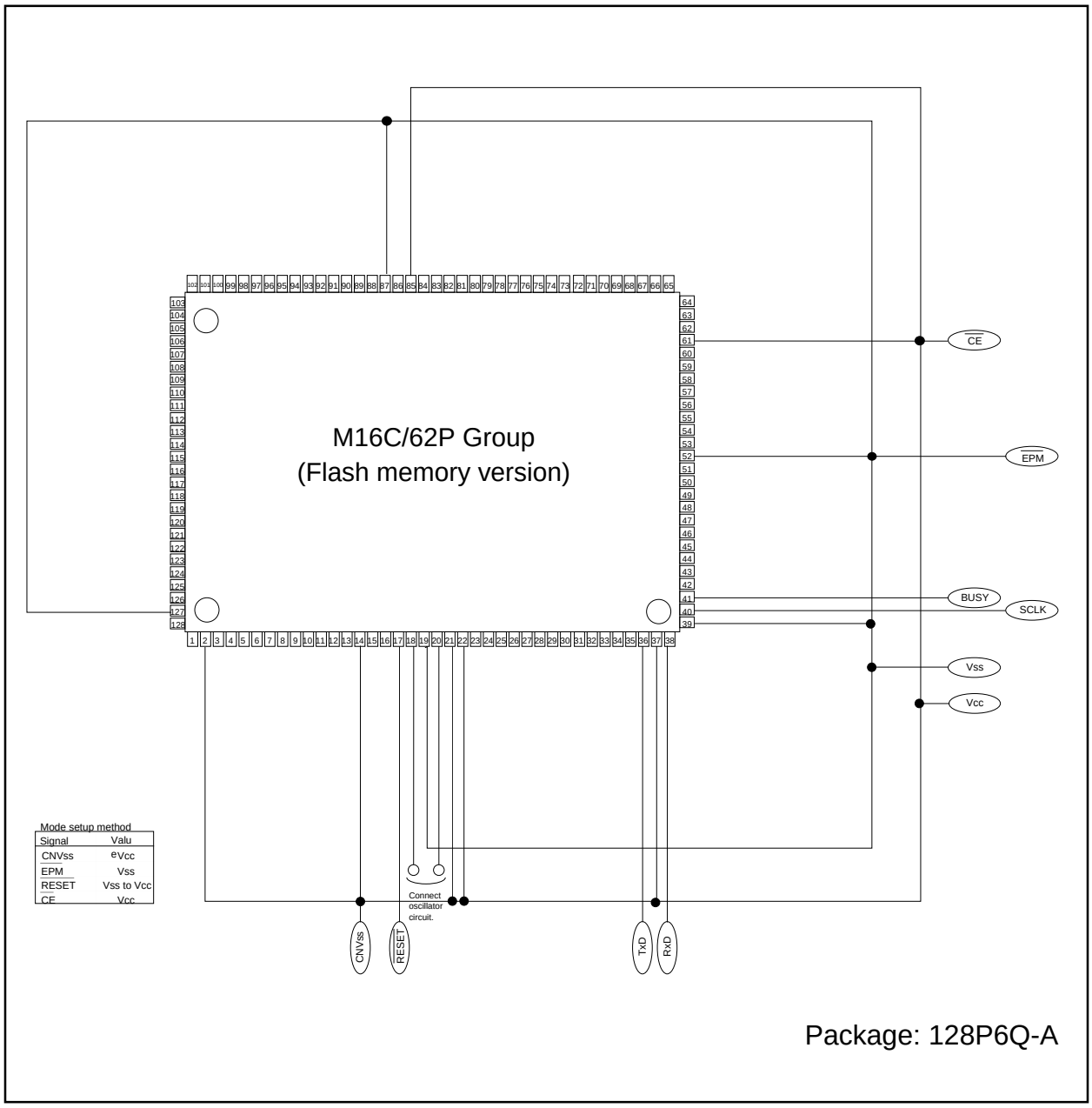
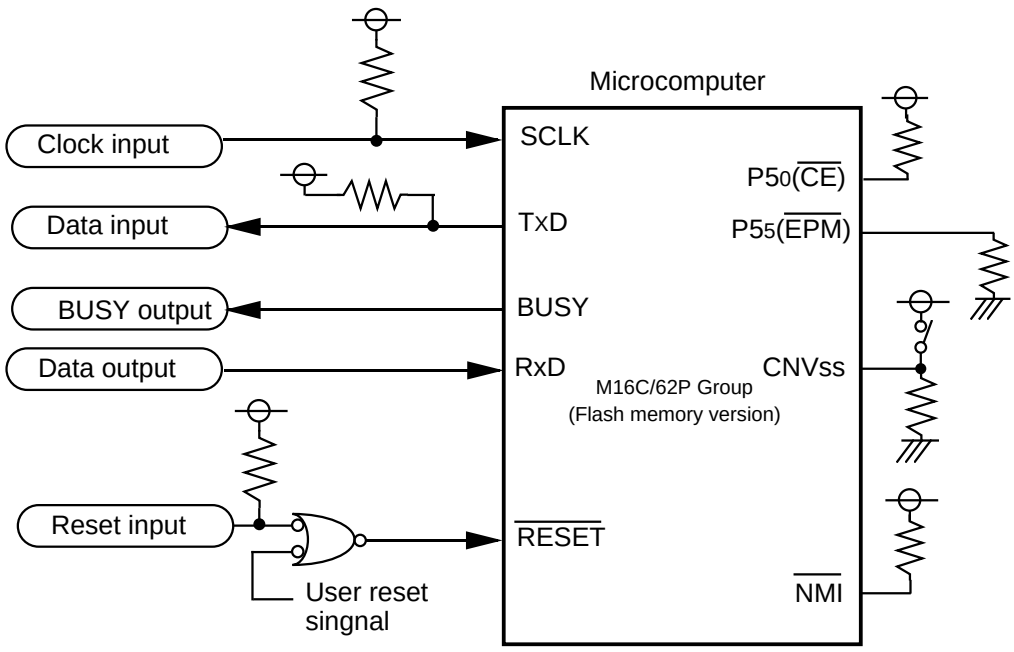


Figure 1.27.15. Pin Connections for Serial I/O Mode (3)

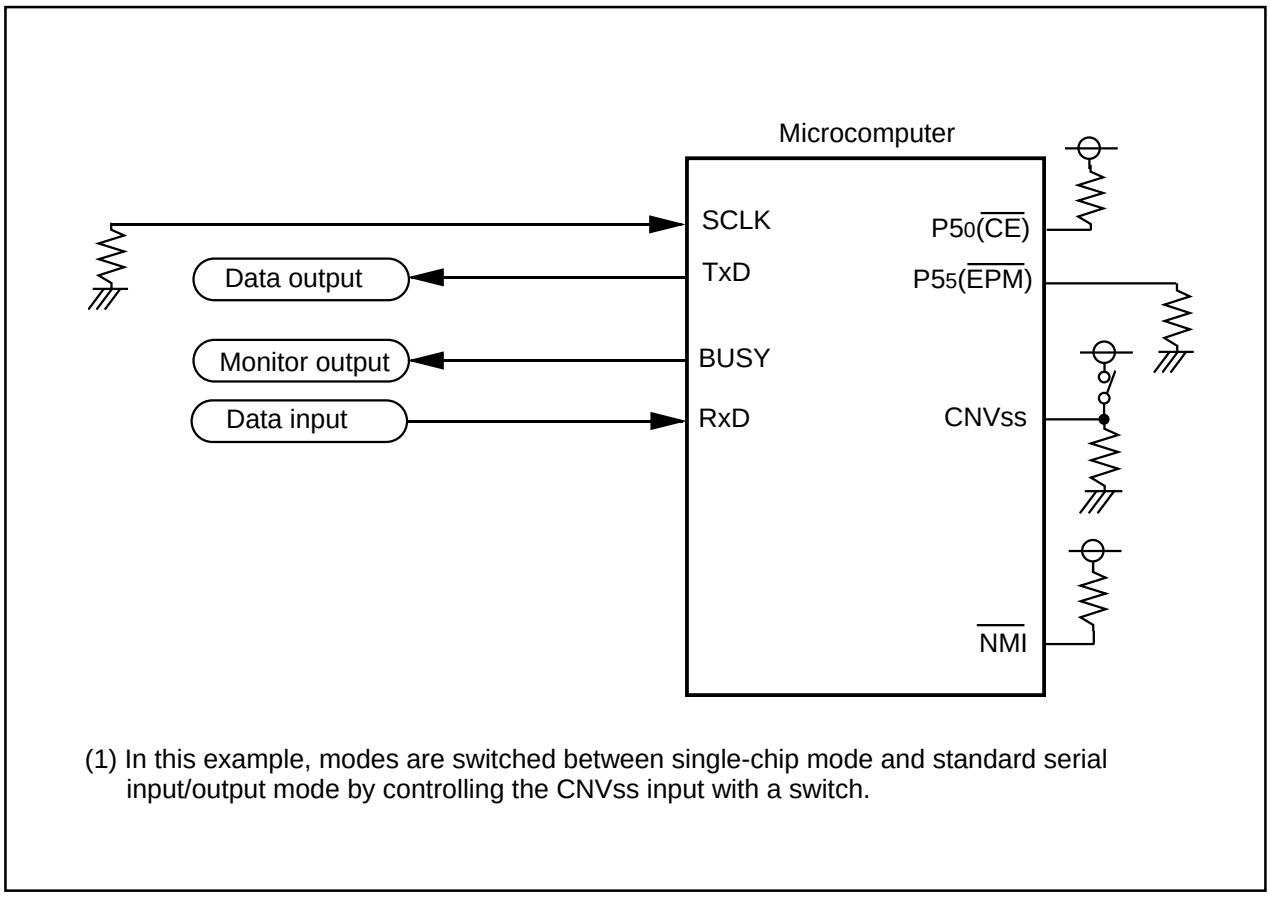
Example of Circuit Application in the Standard Serial I/O Mode

Figure 1.27.16 and 1.27.17 show example of circuit application in standard serial I/O mode 1 and mode 2, respectively. Refer to the user's manual for serial writer to handle pins controlled by a serial writer.



(1) Control pins and external circuitry will vary according to programmer. For more information, see the programmer manual.
(2) In this example, modes are switched between single-chip mode and standard serial input/output mode by controlling the CNVss input with a switch.
(3) If in standard serial input/output mode 1 there is a possibility that the user reset signal will go low during serial input/output mode, break the connection between the user reset signal and $\overline{\text{RESET}}$ pin by using, for example, a jumper switch.

Figure 1.27.16. Circuit Application in Standard Serial I/O Mode 1



(1) In this example, modes are switched between single-chip mode and standard serial input/output mode by controlling the CNVss input with a switch.

Figure 1.27.17. Circuit Application in Standard Serial I/o Mode 2

Parallel I/O Mode

In parallel input/output mode, the user ROM and boot ROM areas can be rewritten by using a parallel programmer suitable for the M16C/62P group. For more information about parallel programmers, contact the manufacturer of your parallel programmer. For details on how to use, refer to the user's manual included with your parallel programmer.

User ROM and Boot ROM Areas

In the boot ROM area, an erase block operation is applied to only one 4 Kbyte block. The boot ROM area contains a standard serial input/output mode based rewrite control program which was written in it when shipped from the factory. Therefore, when using a serial programmer, be careful not to rewrite the boot ROM area.

When in parallel output mode, the boot ROM area is located at addresses $0FF000_{16}$ to $0FFFFF_{16}$. When rewriting the boot ROM area, make sure that only this address range is rewritten. (Do not access other than the addresses $0FF000_{16}$ to $0FFFFF_{16}$.)

ROM Code Protect Function

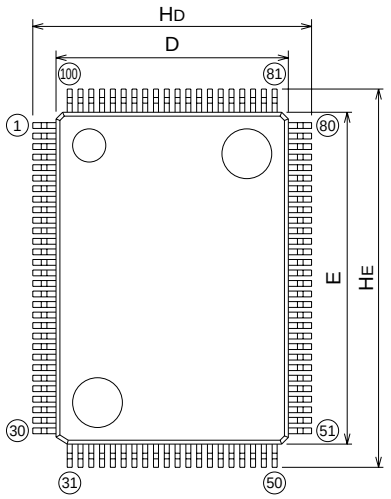
The ROM code protect function inhibits the flash memory from being read or rewritten. (Refer to the description of the functions to inhibit rewriting flash memory version.)

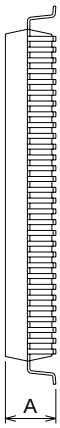
Package Outline

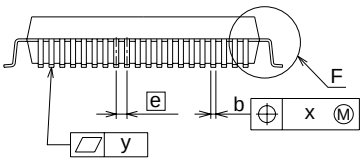
100P6S-A (MMP)

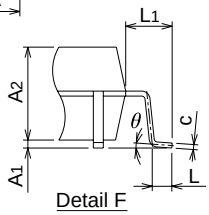
Plastic 100pin 14×20mm body QFP

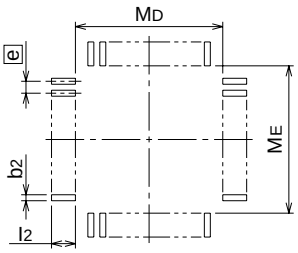
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP100-P-1420-0.65	–	1.58	Alloy 42











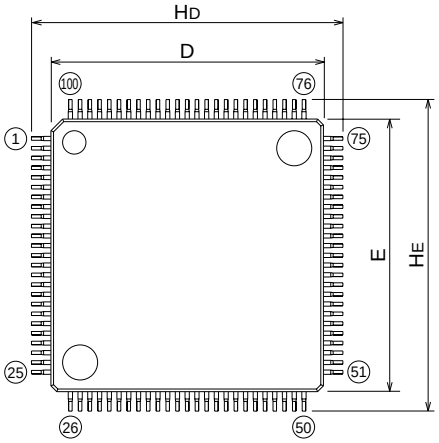
Recommended Mount Pad

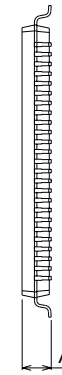
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	3.05
A1	0	0.1	0.2
A2	–	2.8	–
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	–	0.65	–
HD	16.5	16.8	17.1
HE	22.5	22.8	23.1
L	0.4	0.6	0.8
L1	–	1.4	–
x	–	–	0.13
y	–	–	0.1
θ	0°	–	10°
b2	–	0.35	–
l2	1.3	–	–
MD	–	14.6	–
ME	–	20.6	–

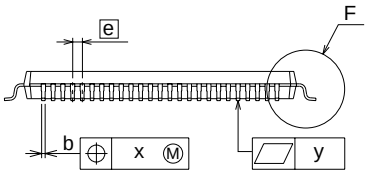
100P6Q-A (MMP)

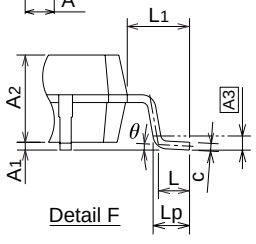
Plastic 100pin 14×14mm body LQFP

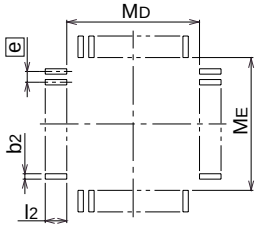
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP100-P-1414-0.50	–	0.63	Cu Alloy











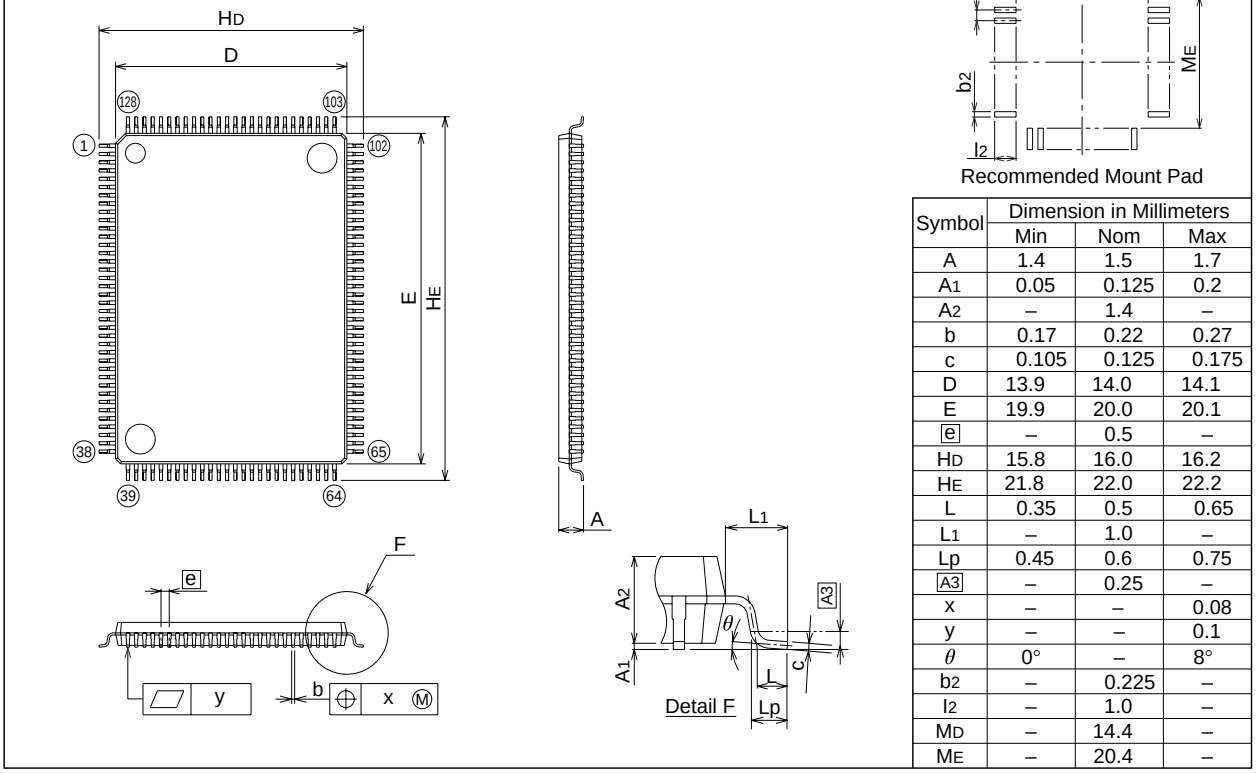
Recommended Mount Pad

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	–

128P6Q-A (MMP)

Plastic 128pin 14×20mm body LQFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP128-P-1420-0.50	–	–	Cu Alloy



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	1.4	1.5	1.7
A_1	0.05	0.125	0.2
A_2	–	1.4	–
b	0.17	0.22	0.27
c	0.105	0.125	0.175
D	13.9	14.0	14.1
E	19.9	20.0	20.1
e	–	0.5	–
H_D	15.8	16.0	16.2
H_E	21.8	22.0	22.2
L	0.35	0.5	0.65
L_1	–	1.0	–
Lp	0.45	0.6	0.75
A3	–	0.25	–
x	–	–	0.08
y	–	–	0.1
θ	0°	–	8°
b_2	–	0.225	–
l_2	–	1.0	–
M_D	–	14.4	–
M_E	–	20.4	–

Differences Between M16C/62P and M16C/62A

Differences in Mask ROM Version and Flash Memory Version (1) (Note)

Item	M16C/62P	M16C/62A
Shortest instruction execution time	41.7ns (f(BCLK)=24MHz, V_{CC1}=3.0 to 5.5V) 100ns (f(BCLK)=10MHz, V_{CC1}=2.7 to 5.5V)	62.5ns (f(X_{IN})=16MHz, V_{CC}=4.2V to 5.5V) 100ns (f(X_{IN})=10MHz, V_{CC}=2.7V to 5.5V with software one-wait)
Supply voltage	V_{CC1}=3.0 to 5.5V, V_{CC2}=3.0V to V_{CC1} (f(BCLK)=24MHz) V_{CC1}=V_{CC2}=2.7 to 5.5V (f(BCLK)=10MHz)	4.2V to 5.5V (f(X_{IN})=16MHz, without software wait) 2.7V to 5.5V (f(X_{IN})=10MHz, with software one-wait)
I/O power supply	Double (V_{CC1}, V_{CC2})	Single (V_{CC})
Package	100-pin, 128-pin plastic mold QFP	80-pin, 100-pin plastic mold QFP
Voltage detection circuit	Built-in Vdet2, Vdet3, Vdet4 detect Power supply voltage down detect interrupt Hardware reset 2	None
Clock Generating Circuit	PLL, X_{IN}, X_{CIN}, ring oscillator Main clock division rate when main clock is stopped: Divide-by-8 frequency X_{IN} drive capacity when main clock is stopped: HIGH	X_{IN}, X_{CIN} Main clock division rate when main clock is stopped: No change X_{IN} drive capacity when main clock is stopped: No change
System clock protective function	Built-in	None (protected by protect register)
Oscillation stop, re-oscillation detection function	Built-in	None
Low power consumption	18mA (V_{CC1}=V_{CC2}=5V, f(BCLK)=24MHz) 8mA (V_{CC1}=V_{CC2}=3V, f(BCLK)=10MHz) 1.8μA (V_{CC1}=V_{CC2}=3V, f(X_{CIN})=32kHz, when wait mode)	32.5mA (V_{CC}=5V, f(X_{IN})=16MHz) 8.5mA (V_{CC}=3V, f(X_{CIN})=10MHz with software one-wait) 0.9μA (V_{CC}=3V, f(X_{CIN})=32kHz, when wait mode)
Memory area	Memory area expandable (4 Mbytes)	1 Mbytes fixed
External device connect area	04000_{16}–$07FFF_{16}$(PM13=0) 08000_{16}–$0FFFF_{16}$(PM10=0) 10000_{16}–$26FFF_{16}$ 28000_{16}–$7FFFF_{16}$ 80000_{16}–$CFFFF_{16}$(PM13=0) $D0000_{16}$–$FFFFF_{16}$(Microprocessor mode)	04000_{16}–$05FFF_{16}$(PM13=0) 06000_{16}–$CFFFF_{16}$ $D0000_{16}$–$FFFFF_{16}$(Microprocessor mode)
Upper address in memory expansion mode and microprocessor mode	$P4_0$ to $P4_3$ (A_{16} to A_{19}), $P3_4$ to $P3_7$ (A_{12} to A_{15}) : Switchable between address bus and port	$P4_0$ to $P4_3$ (A_{16} to A_{19}) : Switchable between address bus and port
Access to SFR	Variable (1 to 2 waits)	1 wait fixed
Software wait to external area	Variable (0 to 3 waits)	Variable (0 to 1 wait)
Protect	Can be set for PM0, PM1, PM2, CM0, CM1, CM2, PLC0, INVC0, INVC1, PD9, S3C, S4C, TB2SC, PCLKR, VCR2, D4INT registers	Can be set for PM0, PM1, CM0, CM1, PD9, S3C, S4C registers
Watchdog timer	Watchdog timer interrupt or watchdog timer reset is selected Count source protective mode is available	Watchdog timer interrupt No count source protective mode
Address match interrupt	4	2

Note: About the details and the electric characteristics, refer to data sheet.

Differences in Mask ROM version and Flash memory version (2) (Note)

Item	M16C/62P	M16C/62A
Timers A, B count source	Selectable: f_1, f_2, f_8, f_{32}, f_{C32}	Selectable: f_1, f_8, f_{32}, f_{C32}
Timer A two-phase pulse signal processing	Z-phase (counter reset) input is available	No Z-phase (counter reset input
Timer functions for three-phase motor control	Function protect by protect register Count source is selectable: f_1, f_2, f_8, f_{32}, f_{C32} Dead time timer count source is selectable: f_1, f_1 divided by 2, f_2 , f_2 divided by 2 Output polarity is selectable Carrier wave phase detectable Three-phase output port $\overline{\text{NMI}}$ control	No function protect by protect register Count source is selectable: f_1, f_8, f_{32}, f_{C32} Dead time timer count source is fixed at $f_1/2$
Serial I/O (UART0 to UART2)	(UART, clock synchronous, I^2C bus, IE bus) x 3	(UART, clock synchronous,) x 2 (UART, clock synchronous, IIC bus, IE bus) x 1
UART0 to UART2, SI/O3, SI/O4 count source	Selectable: f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}	Selectable: f_1, f_8, f_{32}
Serial I/O $\overline{\text{RTS}}$ timing	Assert low when receive buffer is read	Assert low when reception is completed
$\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function	Have	None
UART2 data transmit timing	After data was written, transfer starts at the 2nd BRG overflow timing (same as UART0 and UART1)	After data was written, transfer starts at the 1st BRG overflow timing (Output starts one cycle of BRG overflow earlier than UART0 and UART1)
Serial I/O sleep function	None	Have
Serial I/O I^2C mode	Start condition, stop condition: Auto-generationable	Start condition, stop condition: Not auto-generationable
Serial I/O I^2C mode SDA delay	Only digital delay is selected as SDA delay SDA digital delay count source: BRG	Analog or digital delay is selected as SDA delay SDA digital delay count source: 1/ f(X_{IN})
SI/O3, SI/O4 clock polarity selection	Selectable	Not selectable
A-D converter	10 bits X 8 channels Expandable up to 26 channels	10 bits X 8 channels Expandable up to 10 channels
A-D converter operation clock	Selectable: f_{AD}, f_{AD} divided by 2, 3, 4, 6, 12	Selectable: f_{AD}, $f_{AD}/2$, $f_{AD}/4$
A-D converter input pin	Selectable: ports P0, P2, P10	Fixed at port P10

Note: About the details and the electric characteristics, refer to data sheet.

Differences in Flash memory version(Note)

Item	M16C/62P	M16C/62A
User ROM blocks	14 blocks: 4 Kbytes x 3, 8 Kbytes x 3, 32 Kbytes x1, 64 Kbytes x 7 (Flash memory: max. 512 Kbytes)	7 blocks: 8 Kbytes x 2, 16 Kbytes x1, 32 Kbytes x 1, 64 Kbytes x 3 (Flash memory: max. 256 Kbytes)
Program manner	Word	Page
Program command (software command)	Page program command: none Program command: have (program method: in units of word, in units of byte)	Page program command: have Program command: none (program method: in units of page)
Block status after program function	Have	None
CPU rewrite mode	EW1 mode is available	No EW1 mode

Note: About the details and the electric characteristics, refer to data sheet.

Register Index

U

V

W

Rev.	Date	Description	
		Page	Summary
1.0	Jan/31/Y03 (Continued)	1	Applications are partly revised.
		2	Table 1.1.1 is partly revised.
		5	Table 1.1.3 is partly revised.
		5	Figure 1.1.2 is partly revised.
		11	Explanation of “Memory” is partly revised.
		20	Explanation of “Hardware Reset 1” is partly revised.
		21	Figure 1.5.1 is partly revised.
		22	Figure 1.5.2 is partly revised.
		24	Figure 1.5.4 is partly revised.
		25	VCR2 Register in Figure 1.5.6 is partly revised.
		26	Figure 1.5.6 is partly revised.
		27	Explanation of “Power Supply Down Detection Interrupt” is partly revised.
		30	Figure 1.6.1 is partly revised.
		31	Figure 1.6.2 is partly revised.
		39	Table 1.7.5 is partly revised.
		41	Table 1.7.7 is partly revised.
		43	Figure 1.7.8 is partly revised.
		44	Explanation of “4 Mbyte Mode” is partly revised.
		53	Notes 12 and 13 in Figure 1.9.2 is partly revised.
		54	Notes 2 and 5 in Figure 1.9.3 is partly revised.
		55	Figure 1.9.4 is partly revised.
		57	Note 4 in Figure 1.9.6 is partly revised.
		60	Explanation of “PLL Clock” is partly revised.
		61	Figure 1.9.9 is partly revised.
		62	Explanation of “CPU Clock and BCLK” is partly revised.
		63	Explanation of “Low-speed Mode” is partly revised.
		63	Explanation of “Low Power Dissipation Mode” is partly revised.
		64	Explanation of “Ring Oscillator Low Power Dissipation Mode” is partly revised.
		64	Table 1.9.3 is partly revised.
		65	Table 1.9.5 is partly revised.
		68	Figure 1.9.10 is partly revised.
		69	Figure 1.9.11 is partly revised.
		70	Table 1.9.7 is added.
		71	Explanation of “System Clock Protective Function” is partly revised.
		77	Explanation of “Power Supply Down Detection Interrupt” is partly revised.
		78	Table 1.11.1 is partly revised.
		88	Figure 1.11.9 is partly revised.
		96	WDTS Register in Figure 1.12.2 is partly revised.
		99	Figure 1.13.2 is partly revised.
		100	Figure 1.13.3 is partly revised.
		103	Figure 1.13.5 is partly revised.
		104	Table 1.13.3 is partly revised.
		105	Explanation of “DMA Enable” is partly revised.
		109	Figure 1.14.3 is partly revised.
		115	Table 1.14.3 is partly revised.
		117	Explanation of “Counter Initialization by Two-Phase Pulse Signal Processing” is partly revised.
		117	Figure 1.14.10 is partly revised.
		122	Figure 1.14.14 is partly revised.
		122	Figure 1.14.15 is partly revised.

Rev.	Date	Description	
		Page	Summary
1.0	Jan/31/Y03 (Continued)	124	Figure 1.15.3 is partly revised.
		128	Figure 1.15.7 is partly revised.
		128	Figure 1.15.8 is partly revised.
		130	Figure 1.16.1 is partly revised.
		132	Figure 1.16.3 is partly revised.
		134	Note 7 is added to TAi, TAi1 Register in Figure 1.16.5.
		137	Figure 1.16.8 is partly revised.
		146	UiSMR2 Register in Figure 1.17.7 is partly revised.
		163	Figure 1.20.1 is partly revised.
		164, 165	Table 1.20.2 and Table 1.20.3 are partly revised.
		169	Figure 1.20.4 is partly revised.
		169	Explanation of “Arbitration” is partly revised.
		170	Explanation of “Transfer Clock” is partly revised.
		171	Explanation of “ACK and NACK” is partly revised.
		179	Explanation of “Special Mode 4 (SIM Mode)” is partly revised.
		179	Table 1.20.9 is partly revised.
		184	Figure 1.21.1 is partly revised.
		187	Figure 1.21.4 is partly revised.
		203	Explanation of “External Operation Amp Connection Mode” is partly revised.
		205	Explanation of “Caution of Using A-D Converter” is partly revised.
		205	Figure 1.22.11 is partly revised
		206	Table 1.23.1 is partly revised.
		207	Figure 1.23.3 is partly revised.
		218	Figure 1.25.9 is partly revised.
		223	Table 1.26.1 is partly revised.
		224	Table 1.26.2 is partly revised.
		225	Note 1 of Table 1.26.3 is partly revised.
		225	Note 1 of Table 1.26.4 is partly revised.
		225	Table 1.26.6 is partly revised.
		227	Note 1 of Table 1.26.9 is partly revised.
		228	Note 1 of Table 1.26.10 is partly revised.
		229	Measurement conditions of timing requirements are partly revised.
		229	Table 1.26.11 is partly revised.
		230	Measurement conditions of timing requirements are partly revised.
		230	Table 1.26.18 is added.
		231	Measurement conditions of timing requirements are partly revised.
		232	Measurement conditions of switching characteristics are partly revised.
		233	Measurement conditions of switching characteristics are partly revised.
		234	Measurement conditions of switching characteristics are partly revised.
		235	Figure 1.26.2 is partly revised.
		242	Figure 1.26.9 is partly revised.
		244	Note of Table 1.26.28 is partly revised.
		245	Figure 1.26.29 is partly revised.
		246	Measurement conditions of timing requirements are partly revised.
		246	Table 1.26.30 is partly revised.
		247	Measurement conditions of timing requirements are partly revised.
		247	Table 1.26.37 is added.
		248	Measurement conditions of timing requirements are partly revised.
		249	Measurement conditions of switching characteristics are partly revised.
		250	Measurement conditions of switching characteristics are partly revised.

Rev.	Date	Description	
		Page	Summary
1.0	Jan/31/Y03 (Continued)	251	Measurement conditions of switching characteristics are partly revised.
		252	Figure 1.26.12 is partly revised.
		255	Figure 1.26.15 is partly revised.
		256	Figure 1.26.16 is partly revised.
		257	Figure 1.26.17 is partly revised.
		258	Figure 1.26.18 is partly revised.
		259	Figure 1.26.19 is partly revised.
		260	Figure 1.26.20 is partly revised.
		262	Explanation of “Memory Map” is partly revised.
		263	Explanation of “Boot Mode” is partly revised.
		264	Figure 1.27.3 is partly revised.
		268	Note of FIDR Register in Figure 1.27.4 is partly revised.
		271	Figure 1.27.7 is partly revised.
		272	Explanation of “Interrupts” is partly revised.
		272	Explanation of “Writing in the User ROM Space” is partly revised.
		274	Table 1.27.4 is partly revised.
		274	Explanation of “Read Array Command” is partly revised.
		278	Explanation of “Program Command” is partly revised.
		287	Figure 1.27.15 is partly revised.
		293	Partly revised.

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

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

Printed in Japan (RODJ) 11
New publication, effective February 2003.
Specifications subject to change without notice.

RENESAS
Renesas Technology Corp.

MITSUBISHI 16-BIT SINGLE-CHIP MICROCOMPUTER
M16C FAMILY / M16C/60 SERIES

M16C/62

(M16C/62P)

Group

Usage Notes Reference Book

http://www.infomicom.maec.co.jp/indexe.htm

Before using this material, please visit the above website to confirm that this is the most current document available.

RENESAS
Renesas Technology Corp.

Revision date: February 14, 2003

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

Preface

This book describes the M16C/62 (M16C/62P) group's precautions for use, which contains paragraphs describing precautions of the user's manual and technical news relevant to these paragraphs. Please refer to this book when developing your systems. However, all of precautions are not contained in this book, please perform sufficient evaluation under systems development.

1. Usage Precaution

1.1 Precautions for Interrupts

1.1.1 Reading address 00000_{16}

Do not read the address 00000_{16} in a program. When a maskable interrupt request is accepted, the CPU reads interrupt information (interrupt number and interrupt request priority level) from the address 00000_{16} during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to “0”. If the address 00000_{16} is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to “0”. This causes a problem that the interrupt is canceled, or an unexpected interrupt is generated.

1.1.2 Setting the SP

Set any value in the SP before accepting an interrupt. The SP is cleared to '0000$_{16}$' after reset. Therefore, if an interrupt is accepted before setting any value in the SP, the program may go out of control. Especially when using $\overline{\text{NMI}}$ interrupt, set a value in the SP at the beginning of the program. For the first and only the first instruction after reset, all interrupts including $\overline{\text{NMI}}$ interrupt are disabled.

1.1.3 The $\overline{\text{NMI}}$ Interrupt

1. The $\overline{\text{NMI}}$ interrupt cannot be disabled. If this interrupt is unused, connect the $\overline{\text{NMI}}$ pin to V_{CC} via a resistor (pull-up).
2. The input level of the $\overline{\text{NMI}}$ pin can be read by accessing the P8 register's P8_5 bit. Note that the P8_5 bit can only be read when determining the pin level after an $\overline{\text{NMI}}$ interrupt is generated.
3. Stop mode cannot be entered into while input on the $\overline{\text{NMI}}$ pin is low. This is because while input on the $\overline{\text{NMI}}$ pin is low the CM1 register's CM10 bit is fixed to "0".
4. Do not go to wait mode while input on the $\overline{\text{NMI}}$ pin is low. This is because when input on the $\overline{\text{NMI}}$ pin goes low, the CPU stops but CPU clock remains active; therefore, the current consumption in the chip does not drop. In this case, normal condition is restored by an interrupt generated thereafter.
5. The low and high level durations of the input signal to the $\overline{\text{NMI}}$ pin must each be 2 CPU clock cycles + 300 ns or more.

1.1.4 $\overline{\text{INT}}$ Interrupt

1. Either an “L” level or an “H” level of at least 250 ns width is necessary for the signal input to pins $\overline{\text{INT}}_0$ through $\overline{\text{INT}}_5$ regardless of the CPU operation clock.
2. When the polarity of the $\overline{\text{INT}}_0$ to $\overline{\text{INT}}_5$ pins is changed or the interrupt request cause of the software interrupt numbers 8 to 9 is changed, the IR bit is sometimes set to “1” (interrupt request). After these changes were made, set the interrupt request bit to “0” (no interrupt request). Figure 1.1.1 shows the procedure for changing the $\overline{\text{INT}}$ interrupt generate factor.

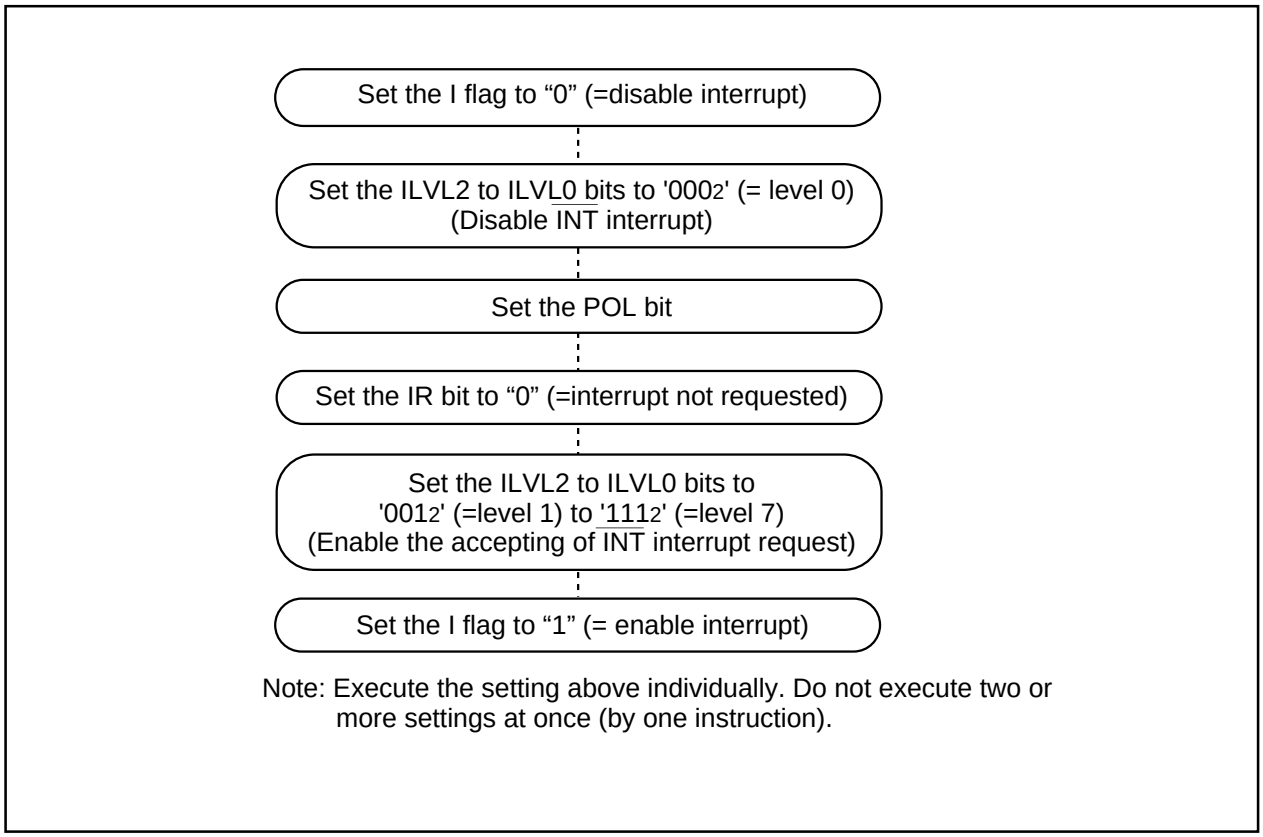


Figure 1.1.1. Procedure for Changing the $\overline{\text{INT}}$ Interrupt Generate Factor

1.1.5 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt occurs.

1.1.6 Rewrite the Interrupt Control Register

Each interrupt control register can only be modified while no interrupt requests corresponding to that register are generated. If interrupt requests managed by any interrupt control register are likely to occur, disable the interrupts before modifying the register. A sample program is shown below.

Example 1:

```
INT_SWITCH1:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Set the TA0IC register to "00₁₆".
    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     ; Set the TA0IC register to "00₁₆".
    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     ; Set the TA0IC register to "00₁₆".
    POPC    FLG             ; Enable interrupts.
```

Why the FSET I instruction is preceded by two NOP instructions (four when using HOLD function) in Example 1 and why the FSET I instruction is preceded by a dummy read in Example 2
This is to prevent the I flag from being set to "1" before writing to the interrupt control register for reasons of the instruction queue buffer.

To modify any interrupt control register after disabling interrupts, be careful with the instructions used.

(1) Modifying Other Than the IR Bit

If an interrupt request corresponding to that register is generated while executing the instruction, the IR bit may not be set to "1" (= interrupt requested), with the result that the interrupt request is ignored. If this presents a problem, use the following instructions to modify the register.
Instructions to use: AND, OR, BCLR, BSET

(2) Modifying the IR Bit

Even when the IR bit is cleared to "0" (= interrupt not requested), it may not actually be cleared to "0" depending on the instruction used. Therefore, use the MOV instruction to clear the IR bit.

1.2 Precautions for Protect

Set the PRC2 bit to “1” (write enabled) and then write to any address, and the PRC2 bit will be cleared to “0” (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to “1”. Make sure no interrupts or DMA transfers will occur between the instruction in which the PRC2 bit is set to “1” and the next instruction.

1.3 Precautions for DMAC

1.3.1 Write to DMAE Bit in DMiCON Register

When both of the conditions below are met, follow the steps below.

Conditions

- The DMAE bit is set to “1” again while it remains set (DMAi is in an active state).
- A DMA request may occur simultaneously when the DMAE bit is being written.

Step 1: Write “1” to the DMAE bit and DMAS bit in DMiCON register simultaneously(*1).
Step 2: Make sure that the DMAi is in an initial state(*2) in a program.
If the DMAi is not in an initial state, the above steps should be repeated.

Notes:

*1. The DMAS bit remains unchanged even if “1” is written. However, if “0” is written to this bit, it is set to “0” (DMA not requested). In order to prevent the DMAS bit from being modified to “0”, “1” should be written to the DMAS bit when “1” is written to the DMAE bit. In this way the state of the DMAS bit immediately before being written can be maintained.
Similarly, when writing to the DMAE bit with a read-modify-write instruction, “1” should be written to the DMAS bit in order to maintain a DMA request which is generated during execution.

*2. Read the TCRi register to verify whether the DMAi is in an initial state. If the read value is equal to a value which was written to the TCRi register before DMA transfer start, the DMAi is in an initial state. (If a DMA request occurs after writing to the DMAE bit, the value written to the TCRi register is “1”.) If the read value is a value in the middle of transfer, the DMAi is not in an initial state.

1.4 Precautions for Timers

1.4.1 Timers A and B

This section describes precautions for timers A and B. Precautions for each mode should be referred as well.

1. After reset, timers stop. After setting mode, count source or counter value, the TAiS bit (i=0 to 4) or TBjS bit (j=0 to 5) in the TABSR or TBSR register should be set to “1” (starts counting). Make sure that the TAiS bit or TBjS bit is set to “0” (stops counting) before changing the registers and bits listed below.
   - TAiMR register and TBjMR register
   - TAi register and TBj register
   - UDF register
   - TAZIE, TA0TGL and TA0TGH bits in ONSF register
   - TRGSR register

1.4.2 Timer A

1.4.2.1 Timer A (Timer Mode)

1. After reset, the TABSR register TAiS bit (i = 0 to 4) is cleared to “0” (stops counting). Select operation mode and set a value in the TAi register before setting the TAiS bit to “1” (starts counting).

2. While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the counter is read at the same time it is reloaded, the value “FFFF$_{16}$” is read. Also, if the counter is read before it starts counting after a value is set in the TAi register while not counting, the set value is read.

3. If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), the TA1$_{OUT}$, TA2$_{OUT}$ and TA4$_{OUT}$ pins go to a high-impedance state.

1.4.2.2 Timer A (Event Counter Mode)

1. After reset, the TABSR register TAiS bit (i = 0 to 4) is cleared to “0” (stopped counting). Select operation mode and set a value in the TAi register before setting the TAiS bit to “1” (start counting).

2. While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, “$FFFF_{16}$” can be read in underflow, while reloading, and “0000_{16}” in overflow. When setting TAi register to a value during a counter stop, the setting value can be read before a counter starts counting. Also, if the counter is read before it starts counting after a value is set in the TAi register while not counting, the set value is read.

3. If a low-level signal is applied to the $\overline{NMI}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{NMI}$ pin enabled), the $TA1_{OUT}$, $TA2_{OUT}$ and $TA4_{OUT}$ pins go to a high-impedance state.

1.4.2.3 Timer A (One-shot Timer Mode)

1. After reset, the TABSR register TAiS bit (i = 0 to 4) is cleared to “0” (stopped counting). Select operation mode and set a value in the TAi register before setting the TAiS bit to “1” (start counting).

2. When setting TABSR register to “0” (count stop), the followings occur:
   • A counter stops counting and a content of reload register is reloaded.
   • TAi$_{OUT}$ pin outputs “L”.
   • After one cycle of the CPU clock, the IR bit of TAiIC register is set to “1” (interrupt request).

3. Output in one-shot timer mode synchronizes with a count source internally generated. When an external trigger has been selected, one-cycle delay of a count source as maximum occurs between a trigger input to TAi$_{IN}$ pin and output in one-shot timer mode.

4. The IR bit is set to “1” when timer operation mode is set with any of the following procedures:
   • Select one-shot timer mode after reset.
   • Change an operation mode from timer mode to one-shot timer mode.
   • Change an operation mode from event counter mode to one-shot timer mode.
   To use the timer Ai interrupt (the IR bit), set the IR bit to “0” after the changes listed above have been made.

5. When a trigger occurs, while counting, a counter reloads the reload register to continue counting after generating a re-trigger and counting down once. To generate a trigger while counting, generate a second trigger between occurring the previous trigger and operating longer than one cycle of a timer count source.

6. If a low-level signal is applied to the $\overline{NMI}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{NMI}$ pin enabled), the TA1$_{OUT}$, TA2$_{OUT}$ and TA4$_{OUT}$ pins go to a high-impedance state.

1.4.2.4 Timer A (Pulse Width Modulation Mode)

1. After reset, the TABSR register TAiS bit (i = 0 to 4) is cleared to “0” (stopped counting). Select operation mode and set a value in the TAi register before setting the TAiS bit to “1” (start counting).

2. The IR bit is set to “1” when setting a timer operation mode with any of the following procedures:
   • Select the PWM mode after reset.
   • Change an operation mode from timer mode to PWM mode.
   • Change an operation mode from event counter mode to PWM mode.
   To use the timer Ai interrupt (interrupt request bit), set the IR bit to “0” by program after the above listed changes have been made.

3. When setting TAiS register to “0” (count stop) during PWM pulse output, the following action occurs:
   • Stop counting.
   • When TAiOUT pin is output “H”, output level is set to “L” and the IR bit is set to “1”.
   • When TAiOUT pin is output “L”, both output level and the IR bit remains unchanged.

4. If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

1.4.3 Timer B

1.4.3.1 Timer B (Timer Mode and Event Counter Mode)

1. After reset, the TBiS bit (i = 0 to 5) is cleared to “0” (stopped counting). Select operation mode and set a value in the TBi register before setting the TBiS bit to “1” (start counting).

The TB0S to TB2S bits are the bits 5 to 7 of TABSR register, the TB3S to TB5S bits are the bits 5 to 7 of TBSR register.

2. A value of a counter, while counting, can be read in TBi register at any time. “$FFFF_{16}$” is read while reloading. Setting value is read between setting values in TBi register at count stop and starting a counter.

1.4.3.2 Timer B (Pulse Period/pulse Width Measurement Mode)

1. The IR bit of TBiIC register (i=0 to 5) goes to “1” (overflow), when an effective edge of a measurement pulse is input or timer Bi is overflowed. The factor of interrupt request can be determined by use of the MR3 bit of TBiMR register within the interrupt routine.

2. If the source of interrupt cannot be identified by the MR3 bit such as when the measurement pulse input and a timer overflow occur at the same time, use another timer to count the number of times timer B has overflowed.

3. To set the MR3 bit to “0” (no overflow), set TBiMR register with setting the TBiS bit to “1” and counting the next count source after setting the MR3 bit to “1” (overflow).

4. Use the IR bit of TBiIC register to detect only overflows. Use the MR3 bit only to determine the interrupt factor within the interrupt routine.

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

6. A value of the counter is indeterminate at the beginning of a count. MR3 may be set to “1” and timer Bi interrupt request may be generated between a count start and an effective edge input.

7. When changing the MR1 to MR0 bits of TBiMR after a count is started, the IR bit of TBiIC register may be set to “1” (interrupt request). Note that the IR bit does not change if the same value as before is written to the MR1 to MR0 bits.

8. For pulse width measurement, pulse widths are successively measured. Use program to check whether the measurement result is an “H” level width or an “L” level width.

1.5 Precautions for Serial I/O (Clock-synchronous Serial I/O)

1.5.1 Transmission/reception

1. With an external clock selected, and choosing the $\overline{\text{RTS}}$ function, the output level of the $\overline{\text{RTSi}}$ pin goes to “L” when the data-receivable status becomes ready, which informs the transmission side that the reception has become ready. The output level of the $\overline{\text{RTSi}}$ pin goes to “H” when reception starts. So if the $\overline{\text{RTSi}}$ pin is connected to the $\overline{\text{CTSi}}$ pin on the transmission side, the circuit can transmission and reception data with consistent timing. With the internal clock, the $\overline{\text{RTS}}$ function has no effect.

2. If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), the $\overline{\text{RTS2}}$ and CLK2 pins go to a high-impedance state.

1.5.2 Transmission

When an external clock is selected, the conditions must be met while if the UiC0 register's CKPOL bit = "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the UiC0 register's CKPOL bit = "1" (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.

- The TE bit of UiC1 register= "1" (transmission enabled)
- The TI bit of UiC1 register = "0" (data present in UiTB register)
- If $\overline{\text{CTS}}$ function is selected, input on the $\overline{\text{CTS}}$i pin = "L"

1.5.3 Reception

1. In operating the clock-synchronous serial I/O, operating a transmitter generates a shift clock. Fix settings for transmission even when using the device only for reception. Dummy data is output to the outside from the TxDi pin when receiving data.

2. When an internal clock is selected, set the UiC1 register (i = 0 to 2)'s TE bit to 1 (transmission enabled) and write dummy data to the UiTB register, and the shift clock will thereby be generated. When an external clock is selected, set the UiC1 register (i = 0 to 2)'s TE bit to 1 and write dummy data to the UiTB register, and the shift clock will be generated when the external clock is fed to the CLKi input pin.

3. When successively receiving data, if all bits of the next receive data are prepared in the UARTi receive register while the UiC1 register (i = 0 to 2)'s RE bit = "1" (data present in the UiRB register), an overrun error occurs and the UiRB register OER bit is set to "1" (overrun error occurred). In this case, because the content of the UiRB register is indeterminate, a corrective measure must be taken by programs on the transmit and receive sides so that the valid data before the overrun error occurred will be retransmitted. Note that when an overrun error occurred, the SiRIC register IR bit does not change state.

4. To receive data in succession, set dummy data in the lower-order byte of the UiTB register every time reception is made.

5. When an external clock is selected, the conditions must be met while if the CKPOL bit = "0", the external clock is in the high state; if the CKPOL bit = "1", the external clock is in the low state.
   - The RE bit of UiC1 register= "1" (reception enabled)
   - The TE bit of UiC1 register= "1" (transmission enabled)
   - The TI bit of UiC1 register= "0" (data present in the UiTB register)

1.6 Precautions for Serial I/O (UART Mode, Special Mode 2)

1. If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), the $\overline{\text{RTS}_2}$ and CLK_2 pins go to a high-impedance state.

1.7 Precautions for A-D Converter

1. Set ADCON0 (except bit 6), ADCON1 and ADCON2 registers when A-D conversion is stopped (before a trigger occurs).

2. When the VCUT bit of ADCON1 register is changed from “0” (Vref not connected) to “1” (Vref connected), start A-D conversion after passing 1 µs or longer.

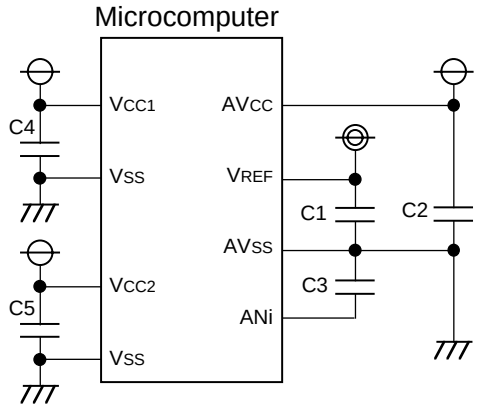
3. To prevent noise-induced device malfunction or latchup, as well as to reduce conversion errors, insert capacitors between the AVCC, VREF, and analog input pins (ANi) each and the AVSS pin. Similarly, insert a capacitor between the VCC pin and the VSS pin. Figure 1.7.1 is an example connection of each pin.

4. Make sure the port direction bits for those pins that are used as analog inputs are set to “0” (input mode). Also, if the ADCON0 register’s TGR bit = 1 (external trigger), make sure the port direction bit for the $\overline{\text{ADTRG}}$ pin is set to “0” (input mode).

5. When using key input interrupts, do not use any of the four AN4 to AN7 pins as analog inputs. (A key input interrupt request is generated when the A-D input voltage goes low.)

6. The ϕAD frequency must be 10 MHz or less. Without sample-and-hold function, limit the ϕAD frequency to 250kHz or more. With the sample and hold function, limit the ϕAD frequency to 1MHz or more.

7. When changing an A-D operation mode, select analog input pin again in the CH2 to CH0 bits of ADCON0 register and the SCAN1 to SCAN0 bits of ADCON1 register.



ANi: ANi, AN0i, and AN2i (i=0 to 7)

Note 1: C1≥0.47µF, C2≥0.47µF, C3≥100pF, C4≥0.1µF, C5≥0.1µF (reference)
Note 2: Use thick and shortest possible wiring to connect capacitors.

Figure 1.7.1. Use of capacitors to reduce noise

8. If the CPU reads the ADi register (i = 0 to 7) at the same time the conversion result is stored in the ADi register after completion of A-D conversion, an incorrect value may be stored in the ADi register. This problem occurs when a divide-by-n clock derived from the main clock or a subclock is selected for CPU clock.
   - When operating in one-shot or single-sweep mode
     Check to see that A-D conversion is completed before reading the target ADi register. (Check the ADiIC register's IR bit to see if A-D conversion is completed.)
   - When operating in repeat mode or repeat sweep mode 0 or 1
     Use the main clock for CPU clock directly without dividing it.

9. If A-D conversion is forcibly terminated while in progress by setting the ADCON0 register's ADST bit to "0" (A-D conversion halted), the conversion result of the A-D converter is indeterminate. The contents of ADi registers irrelevant to A-D conversion may also become indeterminate. If while A-D conversion is underway the ADST bit is cleared to "0" in a program, ignore the values of all ADi registers.

10. If $V_{CC2} < V_{CC1}$, do not use AN_{00} to AN_{07} and AN_{20} to AN_{27} as analog input pins.

1.8 Precautions for Power Control

1. When exiting stop mode by hardware reset, set $\overline{\text{RESET}}$ pin to “L” until a main clock oscillation is stabilized.

2. Insert more than four NOP instructions after an WAIT instruction or a instruction to set the CM10 bit of CM1 register to “1”. When shifting to wait mode or stop mode, an instruction queue reads ahead to the next instruction to halt a program by an WAIT instruction and an instruction to set the CM10 bit to “1” (all clocks stopped). The next instruction may be executed before entering wait mode or stop mode, depending on a combination of instruction and an execution timing.

3. Wait until the $t_{su(M-L)}$ elapses or main clock oscillation stabilization time, whichever is longer, before switching the clock source for CPU clock to the main clock.
Similarly, wait until the sub clock oscillates stably before switching the clock source for CPU clock to the sub clock.

4. Suggestions to reduce power consumption

(a) Ports

The processor retains the state of each I/O port even when it goes to wait mode or to stop mode. A current flows in active I/O ports. A pass current flows in input ports that high-impedance state. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

(b) A-D converter

When A-D conversion is not performed, set the VCUT bit of ADiCON1 register to “0” (no V_{REF} connection). When A-D conversion is performed, start the A-D conversion at least 1 μs or longer after setting the VCUT bit to “1” (V_{REF} connection).

(c) D-A converter

When not performing D-A conversion, set the DAi bit (i=0, 1) of DACON register to “0” (input inhibited) and DAi register to “00_{16}”.

(d) Stopping peripheral functions

Use the CM0 register CM02 bit to stop the unnecessary peripheral functions during wait mode. However, because the peripheral function clock (f_{C32}) generated from the sub-clock does not stop, this measure is not conducive to reducing the power consumption of the chip. During low speed mode and low power dissipation mode, do not set the CM02 bit to “1” (peripheral function clock stopped when in wait mode) before entering wait mode.

(e) Switching the oscillation-driving capacity

Set the driving capacity to “LOW” when oscillation is stable.

(f) External clock

When using an external clock input for the CPU clock, set the CM0 register CM05 bit to “1” (stop). Setting the CM05 bit to “1” disables the X_{OUT} pin from functioning, which helps to reduce the amount of current drawn in the chip. (When using an external clock input, note that the clock remains fed into the chip regardless of how the CM05 bit is set.)

1.9 Precautions for External Bus

1. The external ROM version can operate only in the microprocessor mode, connect the CNVSS pin to VCC.

2. When resetting CNVss pin with "H" input, contents of internal ROM cannot be read out.

1.10 Electric Characteristic Differences Between Mask ROM and Flash Memory Version Microcomputers

Flash memory version and mask ROM version may have different characteristics, operating margin, noise tolerated dose, noise width dose in electrical characteristics due to internal ROM, different layout pattern, etc. When switching to the mask ROM version, conduct equivalent tests as system evaluation tests conducted in the flush memory version.

1.11 Precautions for Flash Memory Version

1.11.1 Precautions for Functions to Inhibit Rewriting Flash Memory Rewrite

ID codes are stored in addresses 0FFFDF16, 0FFFE316, 0FFFEB16, 0FFFEF16, 0FFFF316, 0FFFF716, and 0FFFFB16. If wrong data are written to theses addresses, the flash memory cannot be read or written in standard serial I/O mode.

The ROMCP register is mapped in address 0FFFFF16. If wrong data is written to this address, the flash memory cannot be read or written in parallel I/O mode.

In the flash memory version of microcomputer, these addresses are allocated to the vector addresses (H) of fixed vectors.

1.11.2 Precautions for Program Command

Write 'xx40$_{16}$' in the first bus cycle and write data to the write address in the second bus cycle, and an auto program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same even address as the write address specified in the second bus cycle.

1.11.3 Precautions for Lock Bit Program Command

Write 'xx77$_{16}$' in the first bus cycle and write 'xxD0$_{16}$' to the uppermost address of a block (even address, however) in the second bus cycle, and the lock bit for the specified block is cleared to "0". Make sure the address value specified in the first bus cycle is the same uppermost block address that is specified in the second bus cycle.

1.11.4 Precautions for Stop mode

When shifting to stop mode, the following settings are required:

• Set the FMR01 bit to “0” (CPU rewrite mode disabled) and disable DMA transfers before setting the CM10 bit to “1” (stop mode).

• Execute the JMP.B instruction subsequent to the instruction which sets the CM10 bit to “1” (stop mode)

```
Example program   BSET    0, CM1    ; Stop mode
                  JMP.B   L1
            L1:
                  Program after returning from stop mode
```

1.11.5 Precautions for Wait mode

When shifting to wait mode, set the FMR01 bit to “0” (CPU rewrite mode diabled) before executing the WAIT instruction.

1.11.6 Precautions for CPU Rewrite Mode

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

1.11.6.1 Operation speed

Before entering CPU rewrite mode (EW0 or EW1 mode), select 10 MHz or less for BCLK using the CM0 register's CM06 bit and CM1 register's CM17–6 bits. Also, set the PM1 register's PM17 bit to 1 (with wait state).

1.11.6.2 Instructions inhibited against use

The following instructions cannot be used in EW0 mode because the flash memory's internal data is referenced: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

1.11.6.3 Interrupts

EW0 Mode

- Any interrupt which has a vector in the variable vector table can be used providing that its vector is transferred into the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts can be used because the FMR0 register and FMR1 register are initialized when one of those interrupts occurs. The jump addresses for those interrupt service routines should be set in the fixed vector table.

  Because the rewrite operation is halted when a $\overline{\text{NMI}}$ or watchdog timer interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.
- The address match interrupt cannot be used because the flash memory's internal data is referenced.

EW1 Mode

- Make sure that any interrupt which has a vector in the variable vector table or address match interrupt will not be accepted during the auto program or auto erase period.
- Avoid using watchdog timer interrupts.
- The $\overline{\text{NMI}}$ interrupt can be used because the FMR0 register and FMR1 register are initialized when this interrupt occurs. The jump address for the interrupt service routine should be set in the fixed vector table.

  Because the rewrite operation is halted when a $\overline{\text{NMI}}$ interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

1.11.6.4 How to access

To set the FMR01, FMR02, or FMR11 bit to “1”, write “0” and then “1” in succession. This is necessary to ensure that no interrupts or DMA transfers will occur before writing “1” after writing “0”. Also only when $\overline{\text{NMI}}$ pin is “H” level.

1.11.6.5 Writing in the user ROM area

EW0 Mode

- If the power supply voltage drops while rewriting any block in which the rewrite control program is stored, a problem may occur that the rewrite control program is not correctly rewritten and, consequently, the flash memory becomes unable to be rewritten thereafter. In this case, standard serial I/O or parallel I/O mode should be used.

EW1 Mode

- Avoid rewriting any block in which the rewrite control program is stored.

1.11.6.6 DMA transfer

In EW1 mode, make sure that no DMA transfers will occur while the FMR0 register's FMR00 bit = 0 (during the auto program or auto erase period).

1.11.6.7 Writing command and data

Write the command code and data at even addresses.

1.11.7 Precautions for Low power dissipation mode, ring oscillator low power dissipation mode

If the CM05 bit is set to “1” (main clock stop), the following commands must not be executed.

- Program
- Block erase
- Erase all unlocked blocks
- Lock bit program

1.12 Precautions for PLL Frequency Synthesizer

Make the supply voltage stable to use the PLL frequency synthesizer.

For ripple with the supply voltage 5V, keep below 10kHz as frequency, below 0.5V (peak to peak) as voltage fluctuation band and below 1V/mS as voltage fluctuation rate.

For ripple with the supply voltage 3V, keep below 10kHz as frequency, below 0.3V (peak to peak) as voltage fluctuation band and below 0.6V/mS as voltage fluctuation rate.

1.13 Precautions for Programmable I/O Ports

1. If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the TB2SC register IVPCR1 bit = “1” (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), the $P7_2$ to $P7_5$, $P8_0$ and $P8_1$ pins go to a high-impedance state.

2. Differences Made Depending on Manufactured Time

2.1 Vdet2 Detection

The present version of the products may not detect the Vdet2 voltage in the voltage detection circuit properly. Therefore, the followings should be noted.

(1) When the VC25 bit in the VCR2 register is set to “1” (enabling the RAM retention limit detection circuit), the present version may not be reset even if the voltage at the Vcc1 input pin drops below Vdet2.

(2) The WD5 bit in the WDC register may not change properly.

Supplementary Explanation

Normally, during the stop mode, the Vdet3 voltage is not detected, and thus no reset is generated even when the input voltage at the VCC1 pin drops to Vdet3 or less. Therefore, if the microcomputer is not reset when the VCC1 voltage drops below Vdet2 due to the reason described in the above No.1, the microcomputer cannot get out of the stop mode with Hardware Reset 2.

2.2 RESET Input

Ensure that pin RESET must hold valid-low state during powering-up.

When using a reset IC, use a CMOS type IC. When using an open-drain type reset IC, insert a capacitor between the reset input and Vss and a resistor between the input and Vcc respectively. The R-C time constant of the capacitor and resistor must provide a low state at least 10 times longer than the Vcc rise time.

Rev.	Date	Description	
		Page	Summary
1.0	Jan/31/Y03	1	Figure 1.1.1 is partly revised.
		8	The section “1.3 Precautions for DMAC” is added.
		9	The section “1.4.1 Timers A and B” is added.
		15	The section “1.4.3.2 Timer B (Pulse Period/Pulse Width Measurement Mode” is partly revised.
		18	The section “1.5.3 Reception” is partly revised.
		19	The section “1.6 Precautions for Serial I/O (UART Mode, Special Mode 2)” is partly revised.
		22	The section “1.8 Precautions for Power Control” is partly revised.
		25	The section “1.11.1 Precautions for Functions to Inhibit Rewriting Flash Memory Rewrite” is partly revised.
		26	The section “1.11.2 Precautions for Program Command” is partly revised.
		38	The section “1.12 Precautions for PLL Frequency Synthesizer” is partly revised.

MITSUBISHI SEMICONDUCTORS
USAGE NOTES REFERENCE BOOK
M16C/62 (M16C/62P) Group

February First Edition 2003

Editioned by
Committee of editing of Mitsubishi Semiconductor Usage Notes Reference Book

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

Usage Notes Reference Book
M16C/62 (M16C/62P) Group

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

New publication, effective February 2003.
Specifications subject to change without notice.