M37754M8C-XXXGP, M37754M8C-XXXHP
M37754S4CGP, M37754S4CHP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M37754M8C-XXXGP is a single-chip microcomputer designed with high-performance CMOS silicon gate technology. This is housed in a 100-pin plastic molded QFP.

This microcomputer has a CPU and a bus interface unit. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing, and the bus interface unit enhances the memory access efficiency to execute instructions fast.

In addition to the 7700 Family basic instructions, the M37754M8C-XXXGP has 6 special instructions which contain instructions for signed multiplication/division; these added instructions improve the servo arithmetic performance to control hard disk drives and so on.

This microcomputer also include the ROM, RAM, multiple-function timers, motor control function, serial I/O, A-D converter, D-A converter, and so on.

The differences between M37754M8C-XXXGP, M37754M8C-XXXHP, M37754S4CGP and M37754S4CHP are listed in the table on the next page: the internal ROM, usable processor mode, and package. Therefore, the following descriptions will be for the M37754M8C-XXXGP unless otherwise noted.

DISTINCTIVE FEATURES

- Number of basic machine instructions 109
  (103 basic instructions of 7700 Family + 6 special instructions)
- Memory size ROM ..... 60 Kbytes
  RAM ..... 2048 bytes
- Instruction execution time
  The fastest instruction at 40 MHz frequency ..... 100 ns
- Single power supply ..... 5V ±10 %
- Low power dissipation (at 40 MHz frequency) ..... 125 mW (Typ.)
- Interrupts ..... 21 types, 7 levels
- Multiple-function 16-bit timer ..... 5+3
  (three-phase motor drive waveform or pulse motor control waveform output)
- Serial I/O (UART or clock synchronous) 2
- 10-bit A-D converter ..... 8-channel inputs
- 8-bit D-A converter ..... 2-channel outputs
- 12-bit watchdog timer
- Programmable input/output
  (ports P0—P11) 87
- Small package [M37754M8C-XXXHP]
  ..... 100-pin fine pitch QFP (read pitch : 0.5 mm)

APPLICATION

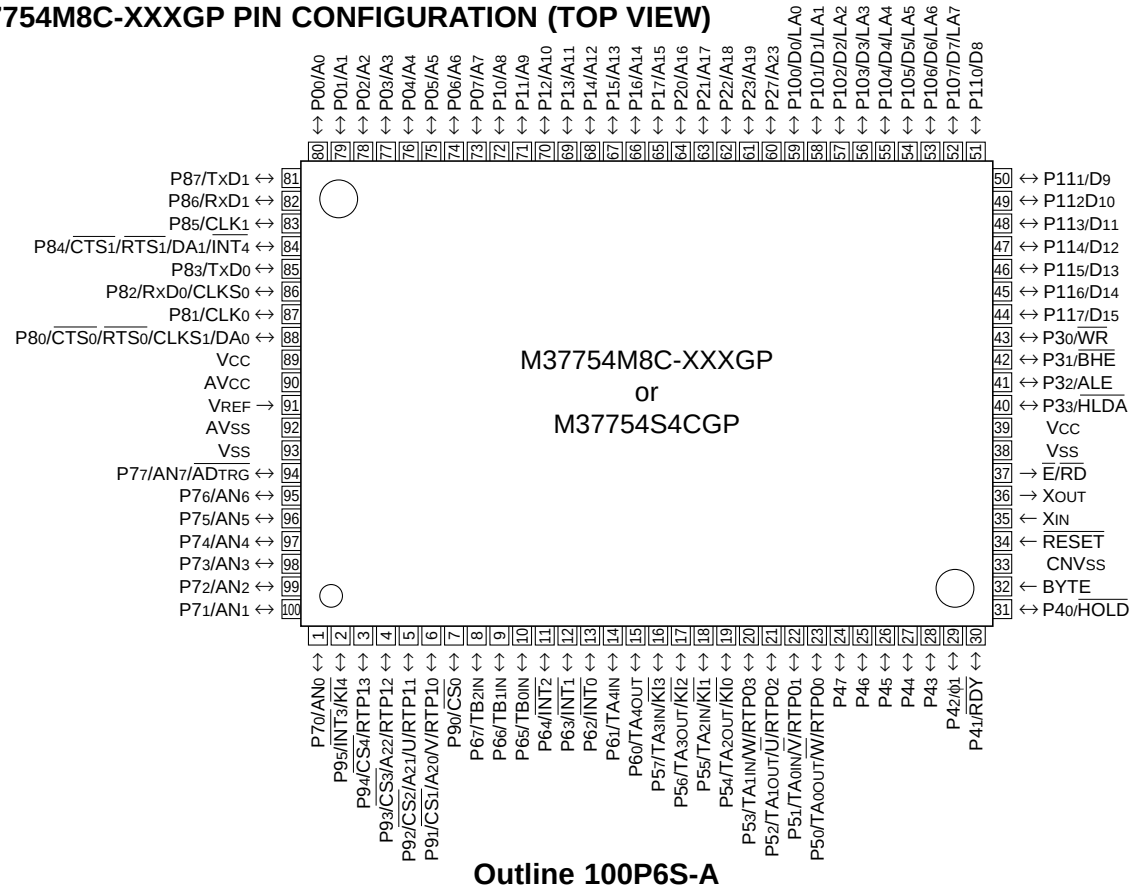
Control devices for personal computer peripheral equipment such as CD-ROM drives, hard disk drives, high density FDD, printers

Control devices for office equipment such as copiers and facsimiles

Control devices for industrial equipment such as communication and measuring instruments

Control devices for equipment required for motor control such as inverter air conditioner and general purpose inverter

M37754M8C-XXXGP PIN CONFIGURATION (TOP VIEW)

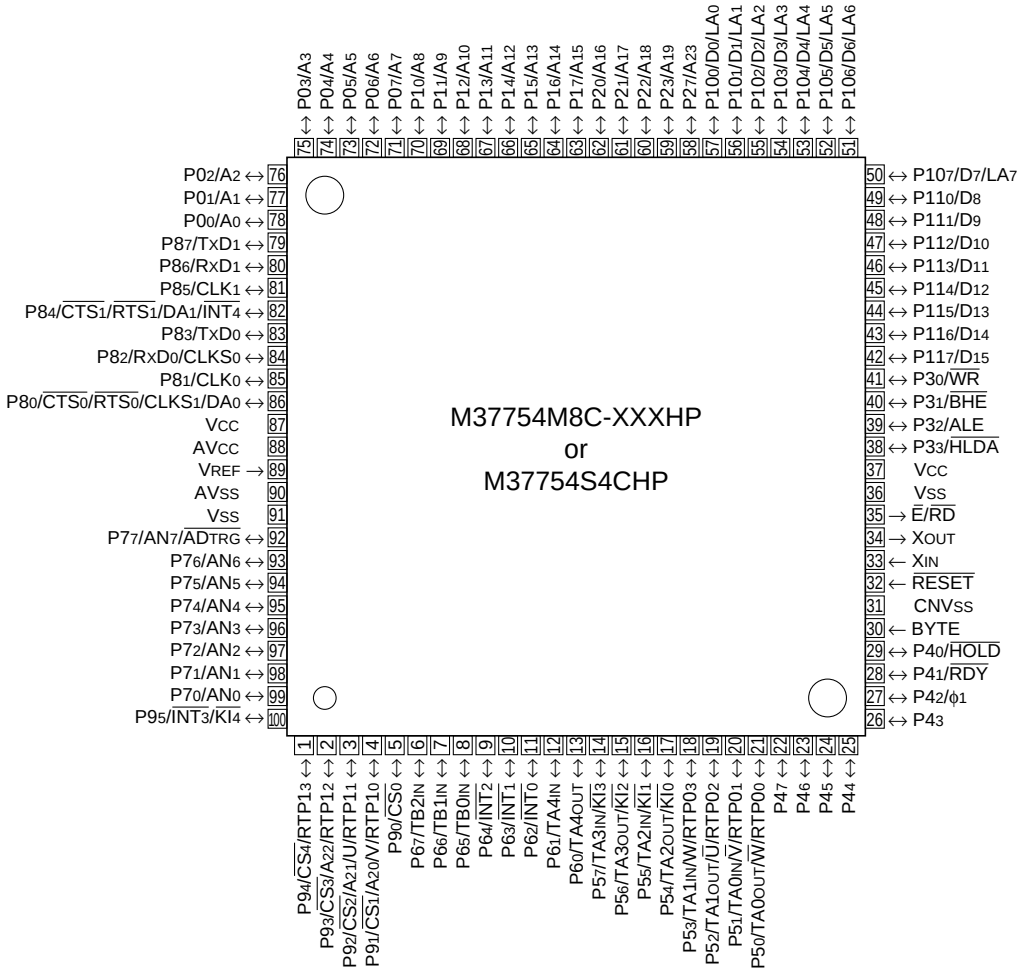


Outline 100P6S-A

M37754M8C-XXXGP, M37754M8C-XXXHP
M37754S4CGP, M37754S4CHP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

M37754M8C-XXXHP PIN CONFIGURATION (TOP VIEW)



Outline 100P6Q-A

Differences between M37754M8C-XXXGP, M37754M8C-XXXHP, M37754S4CGP, and M37754S4CHP

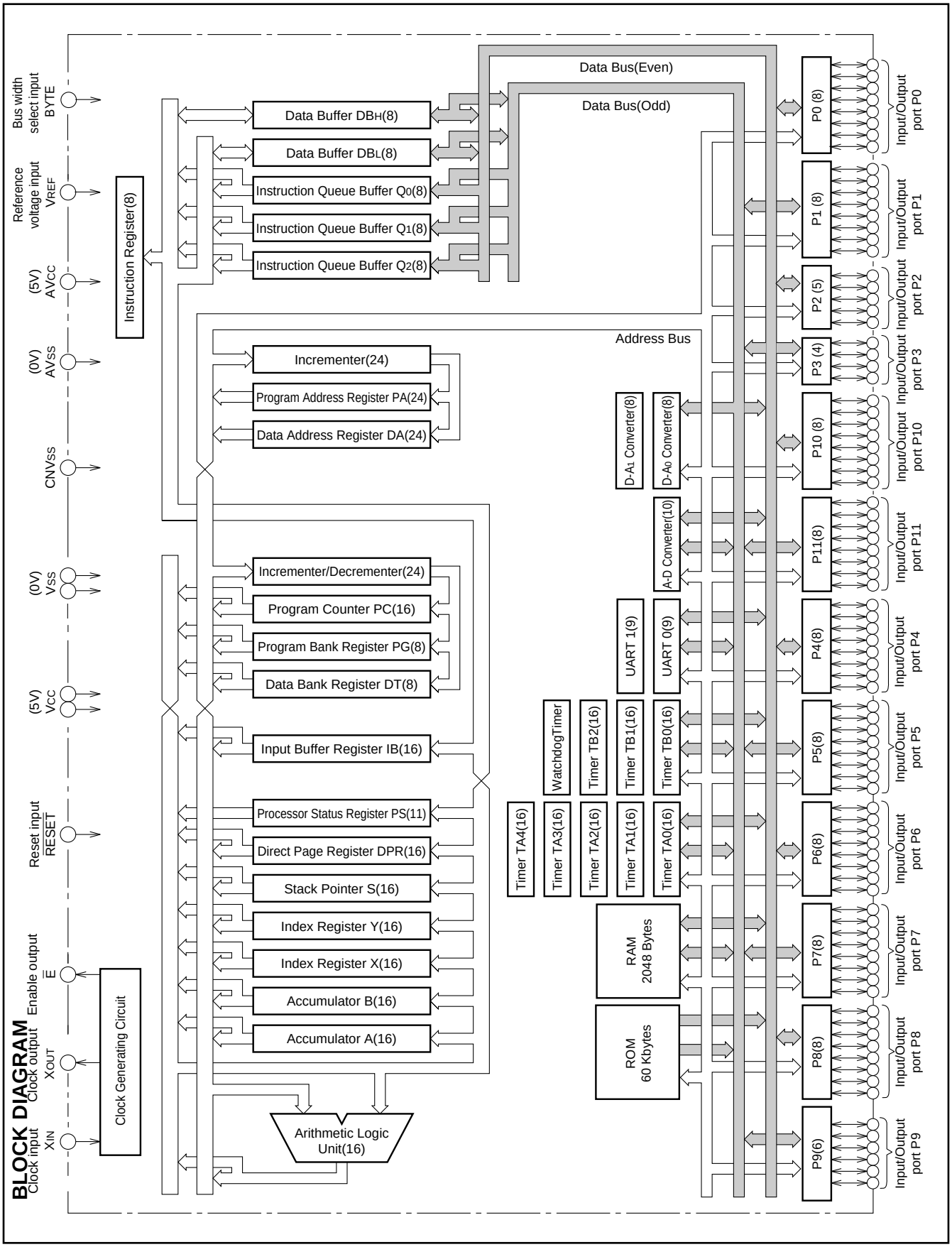
Product	Internal ROM	Usable processor mode	Package
M37754M8C-XXXGP	Equipped (60 Kbytes)	• Single-chip mode • Memory expansion mode • Microprocessor mode	100-pin QFP (100P6S-A)
M37754M8C-XXXHP			100-pin fine pitch QFP (100P6Q-A)
M37754S4CGP	Not equipped (External ROM)	• Microprocessor mode	100-pin QFP (100P6S-A)
M37754S4CHP			100-pin fine pitch QFP (100P6Q-A)

MITSUBISHI ELECTRIC

M37754M8C-XXXGP, M37754M8C-XXXHP
M37754S4CGP, M37754S4CHP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

BLOCK DIAGRAM



FUNCTIONS OF M37754M8C-XXXGP

Parameter		Functions
Number of basic machine instructions		109
Instruction execution time		100 ns (the fastest instruction at external clock 40 MHz frequency)
Memory size	ROM **(Note 1)**	60 Kbytes
	RAM	2048 bytes
Input/Output ports **(Note 2)**	P0, P1, P4 – P8, P10, P11	8-bit × 9
	P2	5-bit × 1
	P3	4-bit × 1
	P9	6-bit × 1
Multiple-function timers	TA0, TA1, TA2, TA3, TA4	16-bit × 5
	TB0, TB1, TB2	16-bit × 3
Serial I/O		(UART or clock synchronous serial I/O) × 2
A-D converter		10-bit × 1(8 channels)
D-A converter		8-bit × 2
Watchdog timer		12-bit × 1
Short-circuit prevention time set timer		8-bit × 3
Interrupts		5 external types, 16 internal types (Each interrupt can be set to priority levels 0 – 7.)
Clock generating circuit		Built-in (externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage		5 V±10 %
Power dissipation		125 mW(at external clock 40 MHz frequency)
Input/Output characteristic	Input/Output withstand voltage	5 V
	Output current	5 mA
Memory expansion		Maximum 16 Mbytes
Operating temperature range		–20 to 85 °C
Device structure		CMOS high-performance silicon gate process
Package		100-pin plastic molded QFP

Notes 1: The M37754S4CGP and the M37754S4CHP are not equipped with ROM.

2: Input/Output ports for the M37754S4CGP and the M37754S4CHP are as shown below :

- P5-P8, P11 (8-bit × 5)
- P4 (5-bit × 1)
- P9 (6-bit × 1)

PIN DESCRIPTION (MICROCOMPUTER MODE)

Pin	Name	Input/ Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5 V±10 % to V_{CC} and 0 V to V_{SS}.
CNV$_{SS}$	CNV$_{SS}$ input	Input	This pin controls the processor mode. Connect to V_{SS} for single-chip mode or memory expansion mode. Connect to V_{CC} for microprocessor mode and external ROM version.
$\overline{RESET}$	Reset input	Input	This is reset input pin. The microcomputer is reset when supplying “L” level to this pin.
X_{IN}	Clock input	Input	These are I/O pins of internal clock generating circuit. Connect a ceramic or quartz-crystal resonator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	This pin outputs enable signal E, which indicates access state of data bus for single-chip mode. This pin outputs $\overline{RD}$ signal for memory expansion mode or microprocessor mode.
BYTE **(Note)**	Bus width select input	Input	This pin determines whether the external data bus is 8-bit width or 16-bit width for memory expansion mode or microprocessor mode. The width is 16 bits when “L” signal inputs and 8 bits when “H” signal inputs.
AV_{CC}, AV_{SS}	Analog supply input		Power supply for the A-D converter and the D-A converter. Connect AV_{CC} to V_{CC} and AV_{SS} to V_{SS} externally.
V_{REF}	Reference voltage input	Input	This is reference voltage input pin for the A-D converter and the D-A converter.
$P0_0$ – $P0_7$	I/O port P0	I/O	In single-chip mode, port P0 is an 8-bit I/O port. This port has an I/O direction register and each pin can be programmed for input or output. These ports are in the input mode when reset. Address (A_0 - A_7) is output in memory expansion mode or microprocessor mode.
$P1_0$ – $P1_7$	I/O port P1	I/O	In single-chip mode, these pins have the same functions as port P0. Address (A_8 - A_{15}) is output in memory expansion mode or microprocessor mode.
$P2_0$ – $P2_3$, $P2_7$	I/O port P2	I/O	In single-chip mode, these pins have the same functions as port P0. Address (A_{16} - A_{19}, A_{23}) is output in memory expansion mode or microprocessor mode.
$P3_0$ – $P3_3$	I/O port P3	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, $\overline{WR}$, $\overline{BHE}$, ALE, and $\overline{HLDA}$ signals are output.
$P4_0$ – $P4_7$	I/O port P4	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or micro processor mode, $P4_0$, $P4_1$, and $P4_2$ become $\overline{HOLD}$ and $\overline{RDY}$ input pins, and clock ϕ_1 output pin respectively. Functions of other pins are the same as in single-chip mode. In memory expansion mode, $P4_2$ can be programmed as I/O port.
$P5_0$ – $P5_7$	I/O port P5	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as I/O pins for timer A0, timer A1, timer A2, timer A3, output pins for motor drive waveform, and input pins for key input interrupt.
$P6_0$ – $P6_7$	I/O port P6	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as I/O pins for timer A4, input pins for external interrupt input $\overline{INT_0}$, $\overline{INT_1}$, and $\overline{INT_2}$, and input pins for timer B0, timer B1, and timer B2.
$P7_0$ – $P7_7$	I/O port P7	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as input pins for A-D converter.
$P8_0$ – $P8_7$	I/O port P8	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as I/O pins for UART0, UART1, output pins for D-A converter, and input pin for $\overline{INT_4}$.
$P9_0$ – $P9_5$	I/O port P9	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as input pin for $\overline{INT_3}$, output pins for motor drive waveform. In memory expansion mode and microprocessor mode, these pins can be programmed as address (A_{20} - A_{22}) or output pins for $\overline{CS_0}$ – $\overline{CS_4}$

Note: It is impossible to change the input level of the BYTE pin in each bus cycle. In other words, bus width cannot be switched dynamically. Fix the input level of the BYTE pin to “H” or “L” according to the bus width used.

Pin	Name	Input/ Output	Functions
$P10_0$ – $P10_7$	I/O port P10	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, these pins become data I/O pins and operate as follows: (1) When using 16-bit width as external data bus width: ● Accessing external memory <When reading> Pins' value is input into low-order internal data bus (DB_0 to DB_7). <When writing> Value of low-order internal data bus (DB_0 to DB_7) is output to these pins. ● Accessing internal memory <When reading> These pins become high impedance. <When writing> Value of internal data bus is output to these pins. (2) When using 8-bit width as external data bus width: ● Accessing external memory <When reading> Pins' value is input into internal data bus. The value is input into low-order internal data bus (DB_0 to DB_7) when accessing an even address; it is input into high-order internal data bus (DB_8 to DB_{15}) when accessing an odd address. <When writing> Value of internal data bus is output to these pins. The value of low-order internal data bus (DB_0 to DB_7) is output when accessing an even address; the value of high-order internal data bus (DB_8 to DB_{15}) is output when accessing an odd address. ● Accessing internal memory <When reading> These pins become high impedance. <When writing> Value of internal data bus is output to these pins. When the external bus width is 8 bits, the mode where low-order address (LA_0 – LA_7) is output when $\overline{RD}$ or $\overline{WR}$ output is "H" and data (D_0 – D_7) is input/output when $\overline{RD}$ or $\overline{WR}$ output is "L" can be selected in specified external memory area access cycle.
$P11_0$ – $P11_7$	I/O port P11	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, these pins operate as follows: (1) When using 16-bit width as external data bus width ● Accessing external memory <When reading> The value is input into high-order internal data bus (DB_8 to DB_{15}) when accessing an odd address; these pins enter high impedance state when not accessing an odd address. <When writing> Value of high-order internal data bus (DB_8-DB_{15}) is output to these pins. ● Accessing internal memory <When reading> These pins enter high impedance state. <When writing> Value of internal data bus is output to these pins. (2) When using 8-bit width as external data bus width These pins become I/O port $P11_0$ – $P11_7$.

BASIC FUNCTION BLOCKS

The M37754M8C-XXXGP contains the following devices on a single chip: ROM, RAM, CPU, bus interface unit, timers, UART, A-D converter, D-A converter, I/O ports, clock generating circuit and others. Each of these devices is described below.

MEMORY

The memory map is shown in Figure 1. The address space is 16 Mbytes from addresses 0_{16} to FFFFFF$_{16}$. The address space is divided into 64-Kbyte units called banks. The banks are numbered from 0_{16} to FF$_{16}$.

Internal ROM, internal RAM, and control registers for internal peripheral devices are assigned to bank 0_{16}.

The 60-Kbyte area from addresses 1000_{16} to FFFF$_{16}$ is the internal ROM.

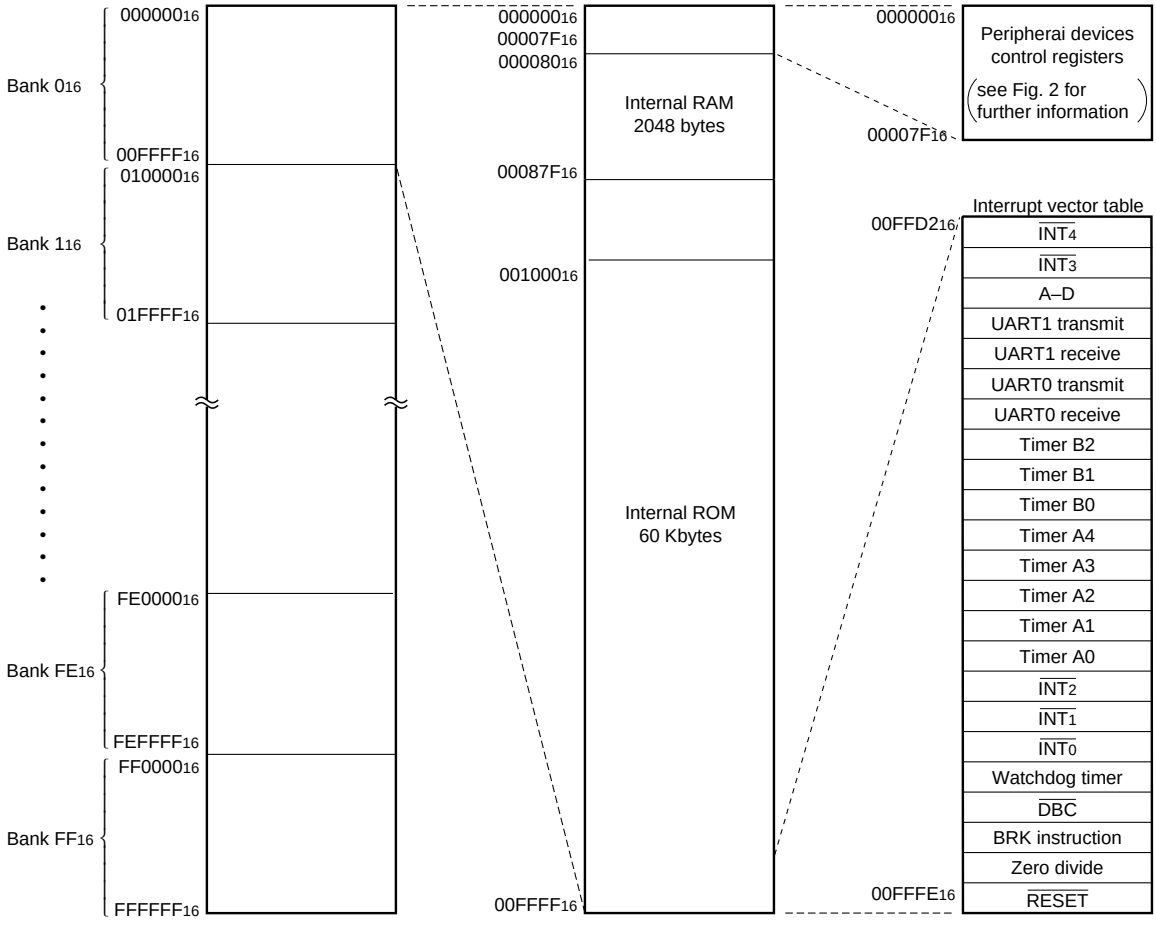
Addresses FFD2$_{16}$ to FFFF$_{16}$ are the RESET and interrupt vector addresses and contain the interrupt vectors. Refer to the section on interrupts for details.

The 2048-byte area from addresses 80_{16} to 87F$_{16}$ contains the internal RAM. In addition to storing data, the RAM is used as stack during a subroutine call, or interrupts.

Assigned to addresses 0_{16} to 7F$_{16}$ are peripheral devices such as I/O ports, A-D converter, D-A converter, UART, timer, and interrupt control registers.

Additionally the internal ROM area can be modified by software. Refer to the section on ROM area modification function for details.

A 256-byte direct page area can be allocated anywhere in bank 0_{16} using the direct page register DPR. In direct page addressing mode, the memory in the direct page area can be accessed with two words thus reducing program steps.



Note: Internal ROM area can be modified. (Refer to the section on ROM area modification function.)

Fig. 1 Memory map

Address (Hexadecimal notation)	
000000	
000001	
000002	Port P0 register
000003	Port P1 register
000004	Port P0 direction register
000005	Port P1 direction register
000006	Port P2 register
000007	Port P3 register
000008	Port P2 direction register
000009	Port P3 direction register
00000A	Port P4 register
00000B	Port P5 register
00000C	Port P4 direction register
00000D	Port P5 direction register
00000E	Port P6 register
00000F	Port P7 register
000010	Port P6 direction register
000011	Port P7 direction register
000012	Port P8 register
000013	Port P9 register
000014	Port P8 direction register
000015	Port P9 direction register
000016	Port P10 register
000017	Port P11 register
000018	Port P10 direction register
000019	Port P11 direction register
00001A	Waveform output mode register
00001B	Dead-time timer
00001C	Pulse output data register 1
00001D	Pulse output data register 0
00001E	A-D control register 0
00001F	A-D control register 1
000020	A-D register 0
000021	
000022	A-D register 1
000023	
000024	A-D register 2
000025	
000026	A-D register 3
000027	
000028	A-D register 4
000029	
00002A	A-D register 5
00002B	
00002C	A-D register 6
00002D	
00002E	A-D register 7
00002F	
000030	UART0 transmit/receive mode register
000031	UART0 baud rate register
000032	UART0 transmit buffer register
000033	
000034	UART0 transmit/receive control register 0
000035	UART0 transmit/receive control register 1
000036	UART0 receive buffer register
000037	
000038	UART1 transmit/receive mode register
000039	UART1 baud rate register
00003A	UART1 transmit buffer register
00003B	
00003C	UART1 transmit/receive control register 0
00003D	UART1 transmit/receive control register 1
00003E	UART1 receive buffer register
00003F	

Address (Hexadecimal notation)	
000040	Count start register
000041	
000042	One-shot start register
000043	
000044	Up-down register
000045	Timer A write register
000046	Timer A0 register
000047	
000048	Timer A1 register
000049	
00004A	Timer A2 register
00004B	
00004C	Timer A3 register
00004D	
00004E	Timer A4 register
00004F	
000050	Timer B0 register
000051	
000052	Timer B1 register
000053	
000054	Timer B2 register
000055	
000056	Timer A0 mode register
000057	Timer A1 mode register
000058	Timer A2 mode register
000059	Timer A3 mode register
00005A	Timer A4 mode register
00005B	Timer B0 mode register
00005C	Timer B1 mode register
00005D	Timer B2 mode register
00005E	Processor mode register 0
00005F	Processor mode register 1
000060	Watchdog timer register
000061	Watchdog timer frequency select register
000062	Chip select control register
000063	Chip select area register
000064	Comparator function select register
000065	Reserved area **(Note)**
000066	Comparator result register
000067	Reserved area **(Note)**
000068	D-A register 0
000069	
00006A	D-A register 1
00006B	
00006C	Particular function select register 0
00006D	Particular function select register 1
00006E	$\overline{INT_4}$ interrupt control register
00006F	$\overline{INT_3}$ interrupt control register
000070	A-D interrupt control register
000071	UART0 trasmit interrupt control register
000072	UART0 receive interrupt control register
000073	UART1 trasmit interrupt control register
000074	UART1 receive interrupt control register
000075	Timer A0 interrupt control register
000076	Timer A1 interrupt control register
000077	Timer A2 interrupt control register
000078	Timer A3 interrupt control register
000079	Timer A4 interrupt control register
00007A	Timer B0 interrupt control register
00007B	Timer B1 interrupt control register
00007C	Timer B2 interrupt control register
00007D	$\overline{INT_0}$ interrupt control register
00007E	$\overline{INT_1}$ interrupt control register
00007F	$\overline{INT_2}$ interrupt control register

Note: Do not write to this address.

Fig. 2 Location of peripheral devices and interrupt control registers

CENTRAL PROCESSING UNIT (CPU)

The CPU has ten registers and is shown in Figure 3. Each of these registers is described below.

ACCUMULATOR A (A)

Accumulator A is the main register of the microcomputer. It consists of 16 bits and the low-order 8 bits can be used separately. The data length flag m determines whether the register is used as 16-bit register or as 8-bit register. It is used as a 16-bit register when flag m is “0” and as an 8-bit register when flag m is “1”. Flag m is a part of the processor status register (PS) which is described later.

Data operations such as calculations, data transfer, input/output, etc., is executed mainly through the accumulator.

ACCUMULATOR B (B)

Accumulator B has the same functions as accumulator A, but the use of accumulator B requires more instruction bytes and execution cycles than accumulator A.

INDEX REGISTER X (X)

Index register X consists of 16 bits and the low-order 8 bits can be used separately. The index register length flag x determines whether the register is used as 16-bit register or as 8-bit register. It is used as a 16-bit register when flag x is “0” and as an 8-bit register when flag x is “1”. Flag x is a part of the processor status register (PS) which is described later.

In index addressing mode, register X is used as the index register and the contents of this address is added to obtain the real address. Index register X functions as a pointer register which indicates an address of data table in instructions MVP, MVN, RMPA (Repeat MultiPly and Accumulate).

INDEX REGISTER Y (Y)

Index register Y consists of 16 bits and the low-order 8 bits can be used separately. The index register length flag x determines whether the register is used as 16-bit register or as 8-bit register. It is used as a 16-bit register when flag x is “0” and as an 8-bit register when flag x is “1”. Flag x is a part of the processor status register (PS) which is described later.

In index addressing mode, register Y is used as the index register and the contents of this address is added to obtain the real address. Index register Y functions as a pointer register which indicates an address of data table in instructions MVP, MVN, RMPA (Repeat MultiPly and Accumulate).

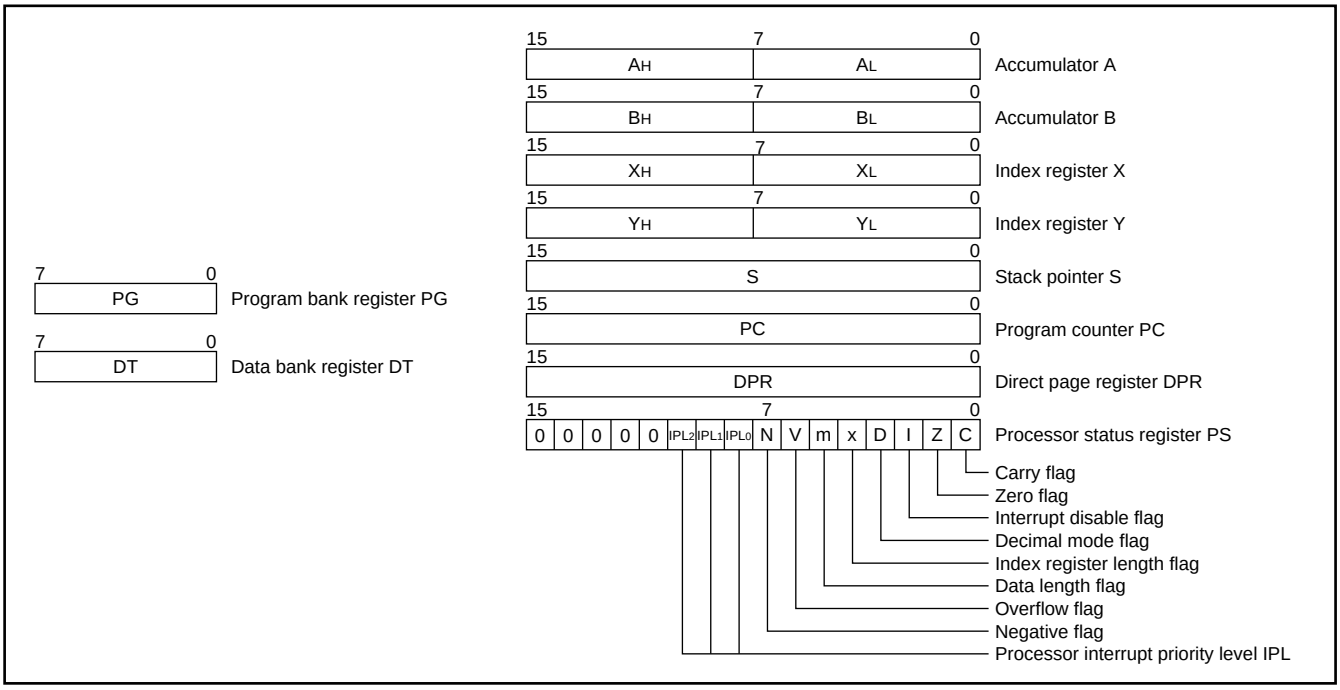


Fig. 3 Register structure

STACK POINTER (S)

Stack pointer (S) is a 16-bit register. It is used during a subroutine call or interrupts. It is also used during stack, stack pointer relative, or stack pointer relative indirect indexed Y addressing mode.

PROGRAM COUNTER (PC)

Program counter (PC) is a 16-bit counter that indicates the low-order 16 bits of the next program memory address to be executed. There is a bus interface unit between the program memory and the CPU, so that the program memory is accessed through bus interface unit. This is described later.

PROGRAM BANK REGISTER (PG)

Program bank register is an 8-bit register that indicates the high-order 8 bits of the next program memory address to be executed. When a carry occurs by incrementing the contents of the program counter, the contents of the program bank register (PG) is increased by 1. Also, when a carry or borrow occurs after adding or subtracting the offset value to or from the contents of the program counter (PC) using the branch instruction, the contents of the program bank register (PG) is increased or decreased by 1, so that programs can be written without worrying about bank boundaries.

DATA BANK REGISTER (DT)

Data bank register (DT) is an 8-bit register. With some addressing modes, the data bank register (DT) is used to specify a part of the memory address. The contents of data bank register (DT) is used as the high-order 8 bits of a 24-bit address. Addressing modes that use the data bank register (DT) are direct indirect, direct indexed X indirect, direct indirect indexed Y, absolute, absolute bit, absolute indexed X, absolute indexed Y, absolute bit relative, and stack pointer relative indirect indexed Y.

DIRECT PAGE REGISTER (DPR)

Direct page register (DPR) is a 16-bit register. Its contents is used as the base address of a 256-byte direct page area. The direct page area is allocated in bank 0_{16}, but when the contents of DPR is $FF01_{16}$ or greater, the direct page area spans across bank 0_{16} and bank 1_{16}. All direct addressing modes use the contents of the direct page register (DPR) to generate the data address. If the low-order 8 bits of the direct page register (DPR) is "00_{16}", the number of cycles required to generate an address is minimized.
Normally the low-order 8 bits of the direct page register (DPR) is set to "00_{16}".

PROCESSOR STATUS REGISTER (PS)

Processor status register (PS) is an 11-bit register. It consists of a flag to indicate the result of operation and CPU interrupt levels.
Branch operations can be performed by testing the flags C, Z, V, and N.
The details of each bit of the processor status register are described below.

1. Carry flag (C)

The carry flag contains the carry or borrow generated by the ALU after an arithmetic operation. This flag is also affected by shift and rotate instructions. This flag can be set and reset directly with the SEC and CLC instructions or with the SEP and CLP instructions.

2. Zero flag (Z)

The zero flag is set if the result of an arithmetic operation or data transfer is zero and reset if it is not. This flag can be set and reset directly with the SEP and CLP instructions.

3. Interrupt disable flag (I)

When the interrupt disable flag is set to "1", all interrupts except watchdog timer, $\overline{DBC}$, and software interrupt are disabled. This flag is set to "1" automatically when there is an interrupt. It can be set and reset directly with the SEI and CLI instructions or SEP and CLP instructions.

4. Decimal mode flag (D)

The decimal mode flag determines whether addition and subtraction are performed as binary or decimal. Binary arithmetic is performed when this flag is "0". If it is "1", decimal arithmetic is performed with each word treated as 2- or 4- digit decimal. Arithmetic operation is performed using four digits when the data length flag m is "0" and with two digits when it is "1". Decimal adjust is automatically performed. (Decimal operation is possible only with the ADC and SBC instructions.) This flag can be set and reset with the SEP and CLP instructions.

5. Index register length flag (x)

The index register length flag determines whether index register X and index register Y are used as 16-bit registers or as 8-bit registers. The registers are used as 16-bit registers when flag x is “0” and as 8-bit registers when it is “1”.
This flag can be set and reset with the SEP and CLP instructions.

6. Data length flag (m)

The data length flag determines whether the data length is 16-bit or 8-bit. The data length is 16-bit when flag m is “0” and 8-bit when it is “1”. This flag can be set and reset with the SEM and CLM instructions or with the SEP and CLP instructions.

7. Overflow flag (V)

The overflow flag is valid when addition or subtraction is performed with a word treated as a signed binary number. If data length flag m is “0”, the overflow flag is set when the result of addition or subtraction is outside the range between –32768 and +32767. If data length flag m is “1”, the overflow flag is set when the result of addition or subtraction is outside the range between –128 and +127. It is reset in all other cases. The overflow flag can also be set and reset directly with the SEP, and CLV or CLP instructions.
Additionally, the overflow flag is set when a result of unsigned/signed division exceeds the length of the register where the result is to be stored; the flag is also set when the addition result is outside range of –2147483648 to +2147483647 in the RMPA operation.

8. Negative flag (N)

The negative flag is set when the result of arithmetic operation or data transfer is negative (If data length flag m is “0”, data’s bit 15 is “1”. If data length flag m is “1”, data’s bit 7 is “1”.) It is reset in all other cases. It can also be set and reset with the SEP and CLP instructions.

9. Processor interrupt priority level (IPL)

The processor interrupt priority level (IPL) consists of 3 bits and determines the priority of processor interrupts from level 0 to level 7. Interrupt is enabled when the interrupt priority of the device requesting interrupt (set using the interrupt control register) is higher than the processor interrupt priority. When interrupt is enabled, the current processor interrupt priority level is saved in a stack and the processor interrupt priority level is replaced by the interrupt priority level of the device requesting the interrupt. Refer to the section on interrupts for more details.
Note: Fix bits 11 to 15 of the processor status register (PS) to “0”.

BUS INTERFACE UNIT

The CPU operates on the basis of internal clock ϕ_{CPU} frequency. In order to speed-up processing, a bus interface unit is used to pre-fetch instructions when the data bus is idle. The bus interface unit synchronizes the CPU and the bus and pre-fetches instructions. Figure 4 shows the relationship between the CPU and the bus interface unit.
The bus interface unit controls buses to access memories easily. Refer to BUS CYCLE on the following pages. The bus interface unit has a program address register, a 3-byte instruction queue buffer, a data address register, and a 2-byte data buffer.
The bus interface unit obtains an instruction code from memory and stores it in the instruction queue buffer, obtains data from memory and stores it in the data buffer, or writes the data form the data buffer to the memory.

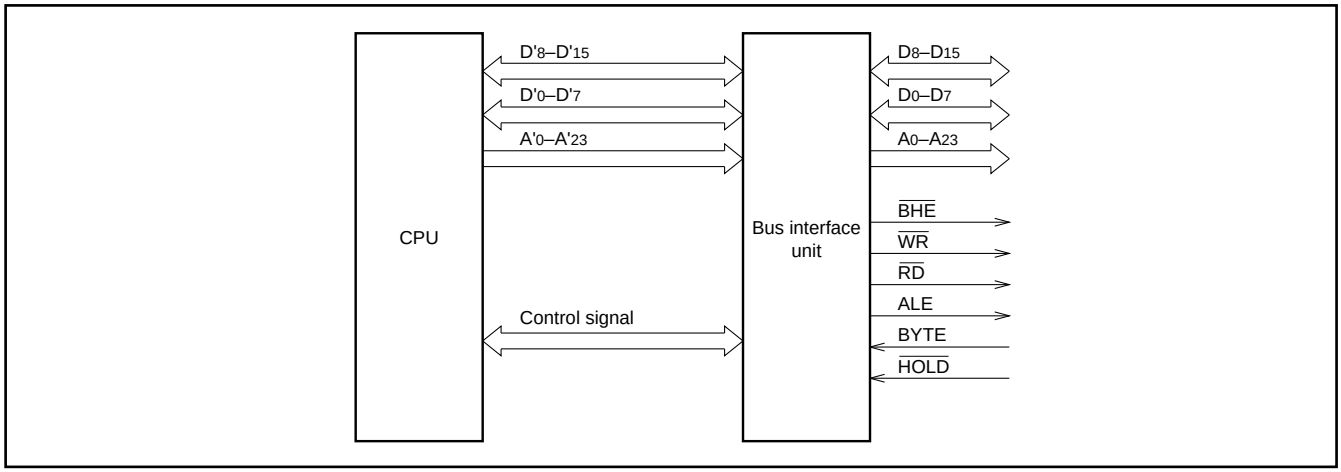


Fig. 4 Relationship between the CPU and the bus interface unit

Figure 5 shows basic waveforms of the bus interface unit. The $\overline{RD}$ signal becomes "L" when the bus interface unit reads an instruction code or data from memory. The $\overline{WR}$ signal becomes "L" when the bus interface unit writes data to memory.

Waveforms (1) and (3) in Figure 5 are used to access a single byte or two bytes simultaneously. To read or write two bytes simultaneously, the first address accessed must be even. Furthermore, when accessing an external memory area in memory expansion mode or microprocessor mode, set the bus width select input pin (BYTE) to "L" (external data bus width = 16 bits). The internal memory area is always treated as 16-bit bus width regardless of BYTE.

When performing 16-bit data read or write, if the conditions for simultaneously accessing two bytes are not satisfied, waveforms (2) and (4) are used to access each byte, one by one.

However, when prefetching the instruction code, if the address of the instruction code is odd, only one byte is read in the instruction queue buffer.

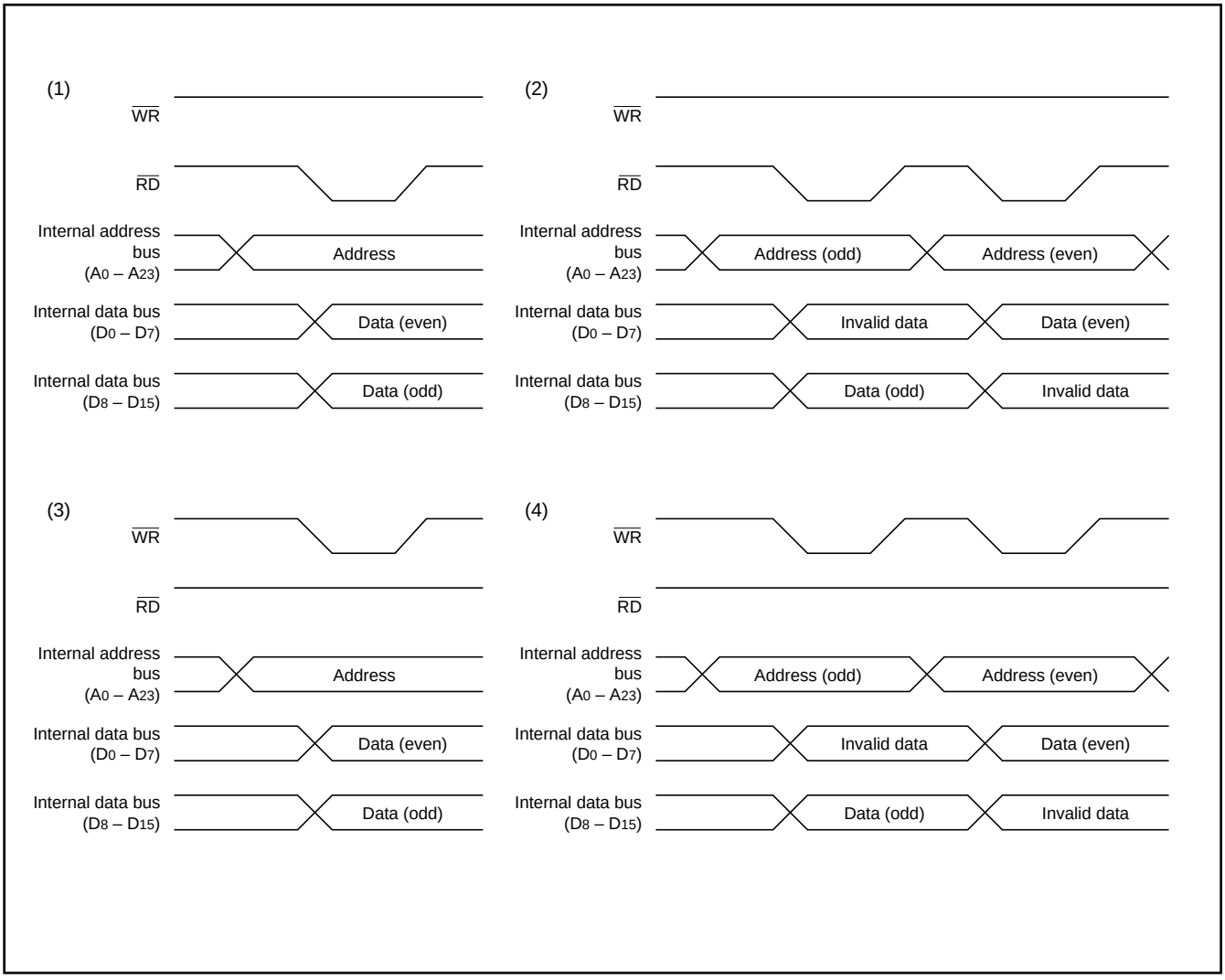


Fig. 5 Basic waveforms of bus interface unit

Instruction code read, data read, and data write are described below. Instruction code read will be described first.

The CPU obtains instruction codes from the instruction queue buffer and executes them. The CPU notifies the bus interface unit that CPU is requesting an instruction code during an instruction code request cycle. If the requested instruction code is not yet stored in the instruction queue buffer, the bus interface unit halts the CPU until it can store more instructions than requested in the instruction queue buffer.

Even if there is no instruction code request from the CPU, the bus interface unit reads instruction codes from memory and stores them in the instruction queue buffer when the instruction queue buffer is empty or when only one instruction code is stored and the bus is idle on the next cycle.

This is referred to as instruction pre-fetching.

Normally, when reading an instruction code from memory, if the accessed address is even, the next odd address is read together with the instruction code and stored in the instruction queue buffer.

However, in memory expansion mode or microprocessor mode, if the bus width select input (BYTE) is “H” and external data bus width is 8 bits, and if the address to be read is in external memory area or is odd, only one byte is read and stored in the instruction queue buffer.

Data read and write are described below.

The CPU notifies the bus interface unit when performing data read or write. At this time, the bus interface unit halts the CPU if the bus interface unit is already using the bus or if there is a request with higher priority. When data read or write is enabled, the bus interface unit performs data read or write.

During data read, the CPU waits until the entire data is stored in the data buffer. The bus interface unit sends the address sent from the CPU to the address bus. Then it reads the memory when the $\overline{RD}$ signal is “L” and stores the result in the data buffer.

During data write, the CPU writes the data in the data buffer and the bus interface unit writes it to memory. Therefore, the CPU can proceed to the next step without waiting for write to complete. The bus interface unit sends the address sent from the CPU to the address bus. Then, when the $\overline{WR}$ signal is “L”, the bus interface unit sends the data in the data buffer to the data bus and writes it to memory.

BUS CYCLE

The M37754M8C-XXXGP can select bus cycles shown in Figures 6 and 7.

Central processing unit (CPU) running speed can be selected from low-speed running (clock $\phi_1 \leq 12.5$ MHz) and high-speed running (clock $\phi_1 \leq 20$ MHz); it is selected by bit 3 of processor mode register 1 (see Figure 9).

When accessing the external memory, the bus cycle is selected by bits 4 and 5 of processor mode register 1.

When accessing the internal memory, the bus cycle is selected by bit 2 of processor mode register 0 (see Figure 14).

Figure 8 shows output signals at 3-ϕ access in high-speed running. The $\overline{BHE}$ signal becomes “L” when accessing the odd address.

Signals A_0 and $\overline{BHE}$ indicate the differences between 1-byte read in even address, 1-byte read in odd address, and simultaneous 2-byte read in even and odd address; these signals also indicate the differrences between 1-byte write in even address, 1-byte write in odd address, and simultaneous 2-byte write in even and odd address.

The A_0 signal, which is bit 0 of address, becomes “L” when accessing an even address.

Table 1. Signals A_0 and $\overline{BHE}$

Signal \ Access method	Simultaneous access of 2 bytes	Access of 1 byte in even address	Access of 1 byte in odd address
A_0	“L”	“L”	“H”
$\overline{BHE}$	“L”	“H”	“L”

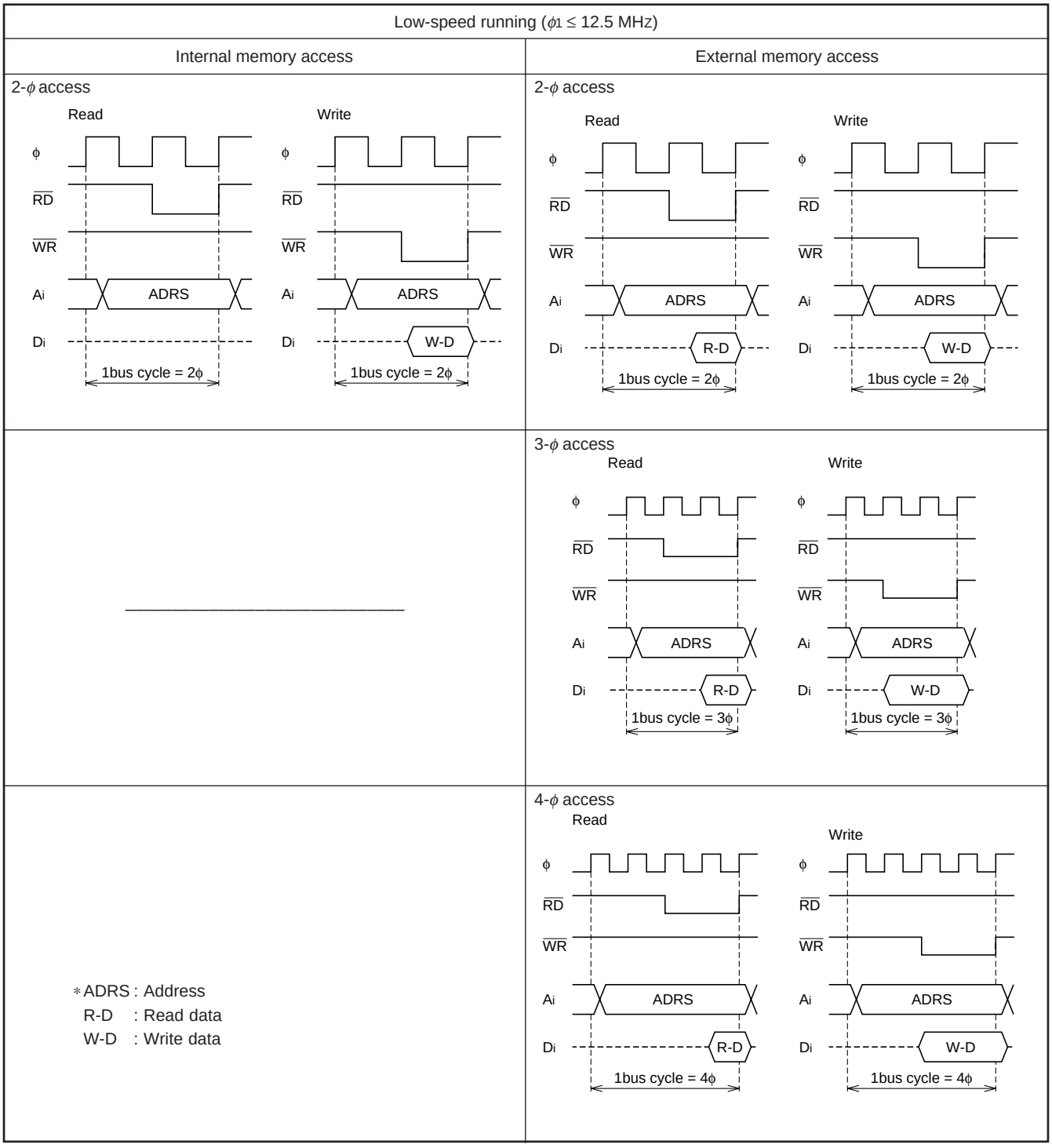


Fig. 6 Bus cycle selection (low-speed running)

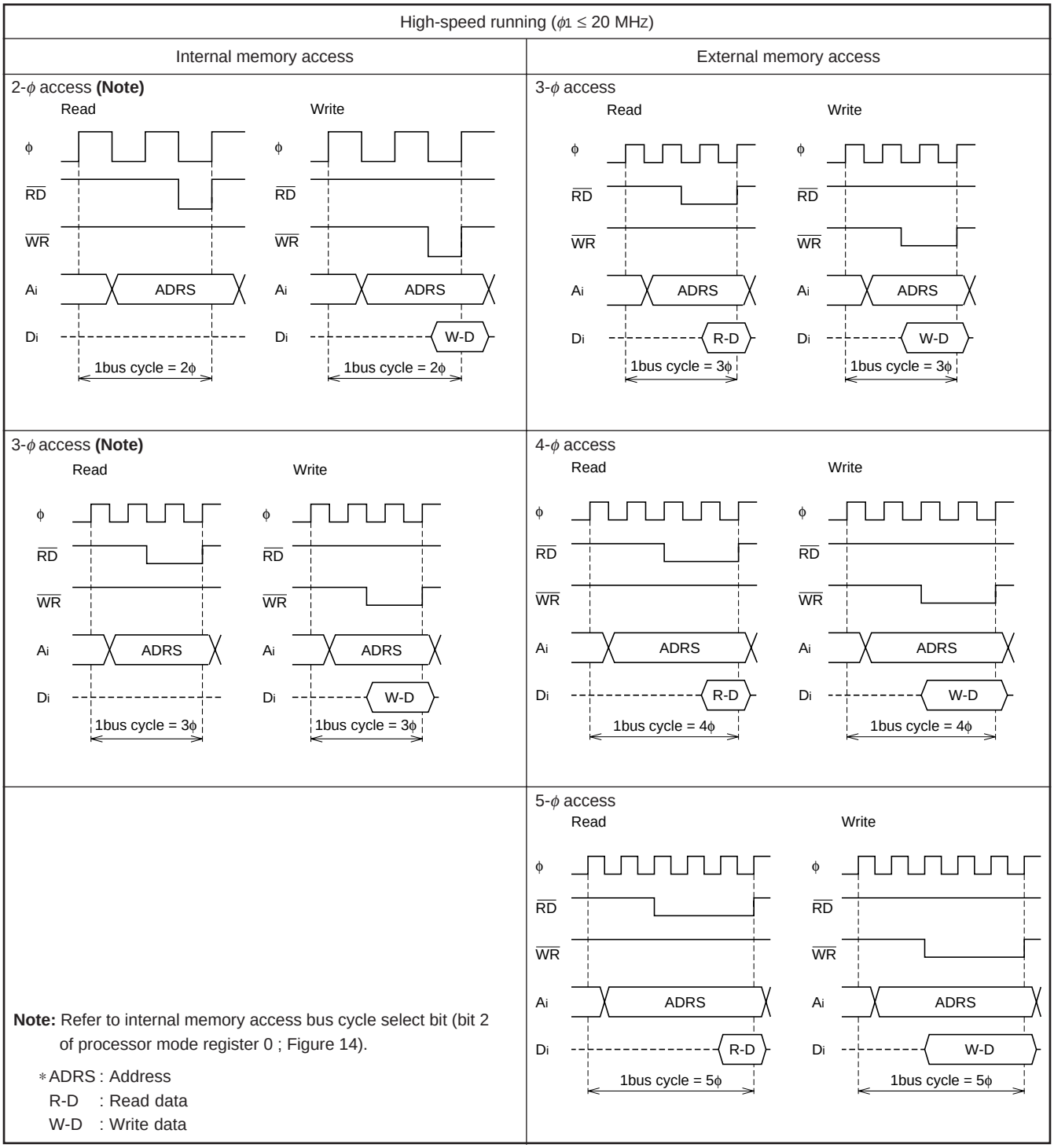
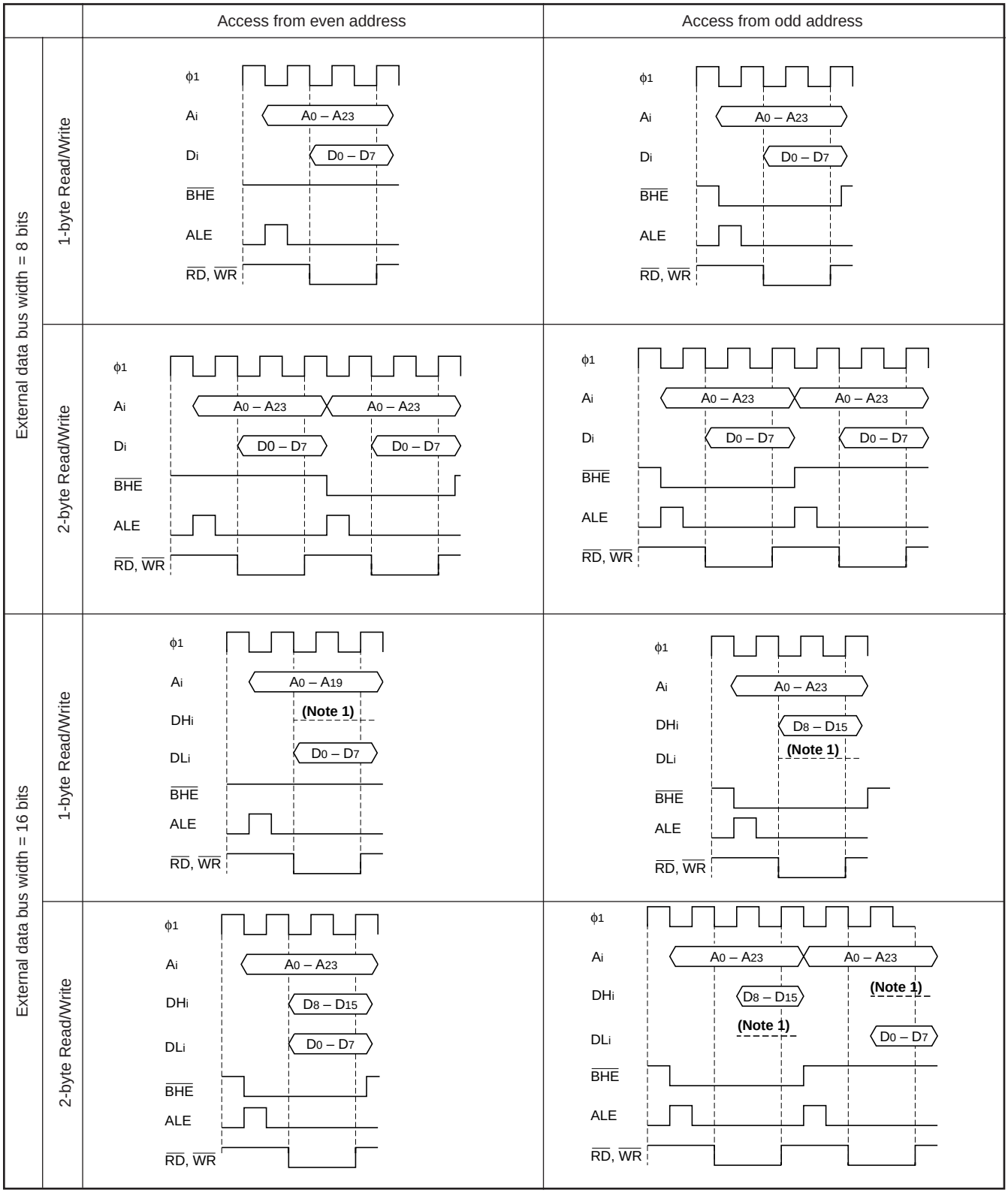
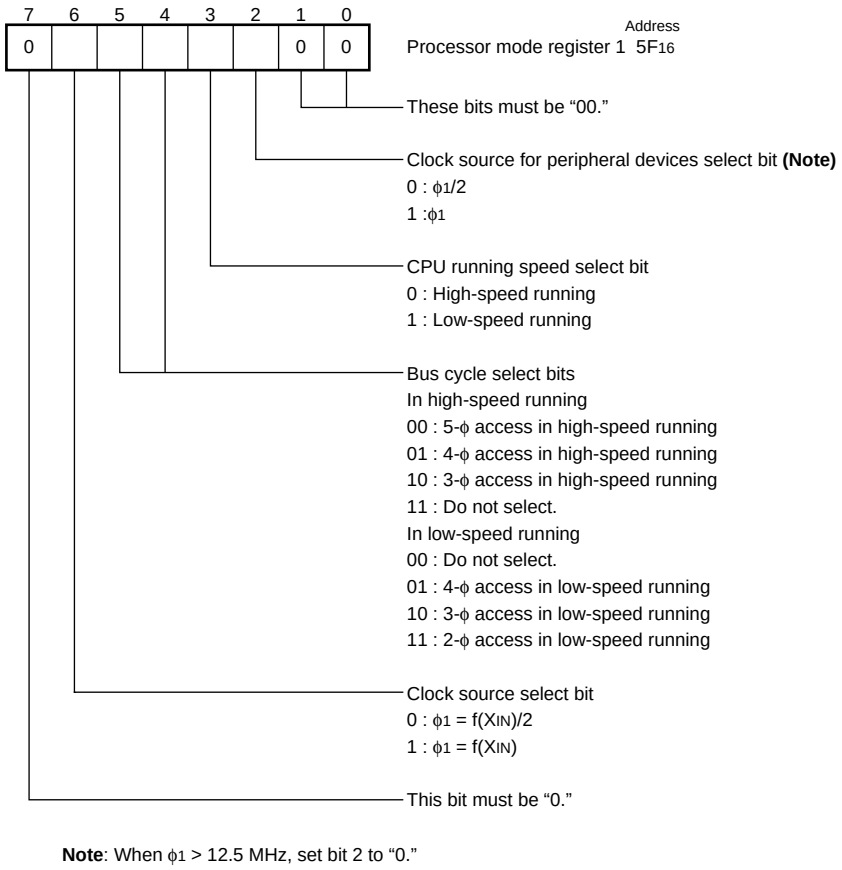


Fig. 7 Bus cycle selection (high-speed running)



Notes 1: It becomes Hi-Z when reading, and it outputs undefined data when writing.
2: When the external data bus width is 8 bits, the function to output the low-order address from the Di pin while $\overline{RD}$ or $\overline{WR}$ is "H" can be selected only in special area access cycle. Refer to the section on the processor mode for details.

Fig. 8 Output signals at 3-ϕ access in high-speed running



Note: When ϕ_1 > 12.5 MHz, set bit 2 to “0.”

Fig. 9 Processor mode register 1 bit configuration

INTERRUPTS

Table 2 shows the interrupt types and the corresponding interrupt vector addresses. Reset is also treated as a type of interrupt and is discussed in this section, too.

$\overline{DBC}$ is an interrupt used during debugging.

Interrupts other than reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction all have interrupt control registers. Table 3 shows the addresses of the interrupt control registers and Figure 10 shows the bit configuration of the interrupt control register.

The interrupt request bit is automatically cleared by the hardware during reset or when processing an interrupt. Also, interrupt request bits other than $\overline{DBC}$ and watchdog timer can be cleared by software.

$\overline{INT_4}$ to $\overline{INT_0}$ are external interrupts; whether to cause an interrupt at the input level (level sense) or at the edge (edge sense) can be selected with the level/edge select bit. Furthermore, the polarity of the interrupt input can be selected with the polarity select bit.

In the $\overline{INT_3}$ external interrupt, the $\overline{INT_3}$ input, $\overline{KI_3}$ to $\overline{KI_0}$ inputs, or $\overline{KI_4}$ to $\overline{KI_0}$ inputs can be selected with bits 7 and 6 of $\overline{INT_3}$ interrupt control register.

Timer and UART interrupts are described in the respective section.

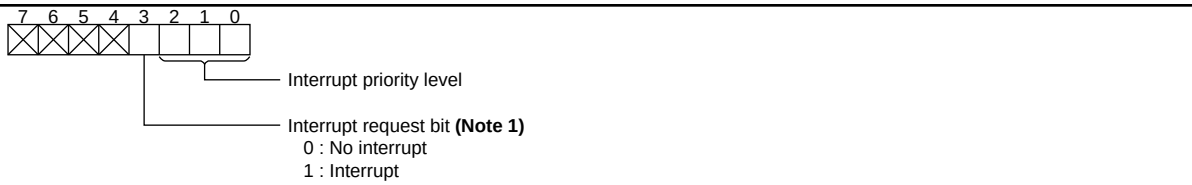
The priority of interrupts when multiple interrupts are caused simultaneously is partially fixed by hardware, but, it can also be adjusted by software as shown in Figure 11.

The hardware priority is fixed as the following:

reset > $\overline{DBC}$ > watchdog timer > other interrupts

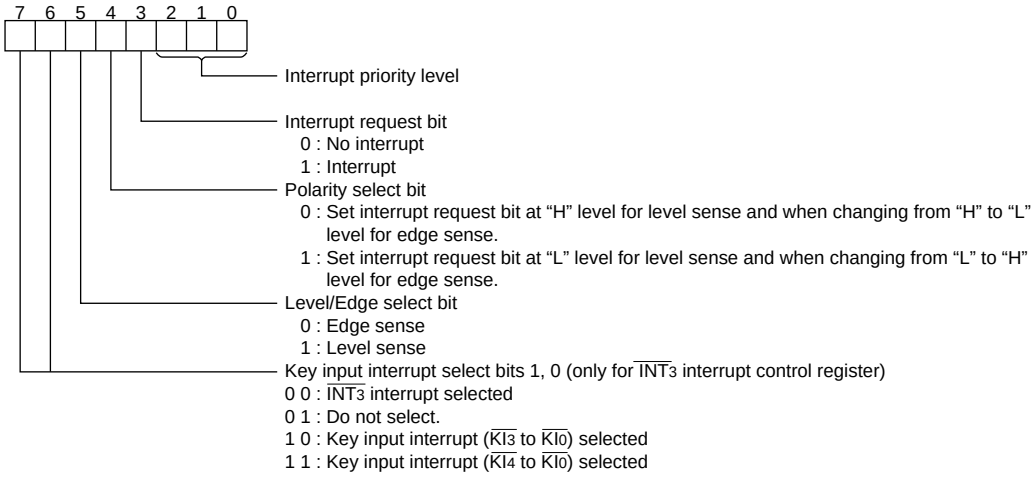
Table 2. Interrupt types and the interrupt vector addresses

Interrupts	Vector addresses
$\overline{INT_4}$ external interrupt	00FFD2$_{16}$ 00FFD3$_{16}$
$\overline{INT_3}$ external interrupt	00FFD4$_{16}$ 00FFD5$_{16}$
A-D	00FFD6$_{16}$ 00FFD7$_{16}$
UART1 transmit	00FFD8$_{16}$ 00FFD9$_{16}$
UART1 receive	00FFDA$_{16}$ 00FFDB$_{16}$
UART0 transmit	00FFDC$_{16}$ 00FFDD$_{16}$
UART0 receive	00FFDE$_{16}$ 00FFDF$_{16}$
Timer B2	00FFE0$_{16}$ 00FFE1$_{16}$
Timer B1	00FFE2$_{16}$ 00FFE3$_{16}$
Timer B0	00FFE4$_{16}$ 00FFE5$_{16}$
Timer A4	00FFE6$_{16}$ 00FFE7$_{16}$
Timer A3	00FFE8$_{16}$ 00FFE9$_{16}$
Timer A2	00FFEA$_{16}$ 00FFEB$_{16}$
Timer A1	00FFEC$_{16}$ 00FFED$_{16}$
Timer A0	00FFEE$_{16}$ 00FFEF$_{16}$
$\overline{INT_2}$ external interrupt	00FFF0$_{16}$ 00FFF1$_{16}$
$\overline{INT_1}$ external interrupt	00FFF2$_{16}$ 00FFF3$_{16}$
$\overline{INT_0}$ external interrupt	00FFF4$_{16}$ 00FFF5$_{16}$
Watchdog timer	00FFF6$_{16}$ 00FFF7$_{16}$
$\overline{DBC}$ (Do not select.)	00FFF8$_{16}$ 00FFF9$_{16}$
Break instruction	00FFFA$_{16}$ 00FFFB$_{16}$
Zero divide	00FFFC$_{16}$ 00FFFD$_{16}$
Reset	00FFFE$_{16}$ 00FFFF$_{16}$



Interrupt control register configuration for A-D converter, UART0, UART1, timer A0 to timer A4, and timer B0 to timer B2.

Note 1: The A-D conversion interrupt request bit becomes undefined after reset. Clear this bit to "0" before use of the A-D conversion interrupt.



Interrupt control register configuration for $\overline{INT_4}$– $\overline{INT_0}$ **(Note 2)**.

Note 2: The contents of $\overline{INT_4}$ interrupt control register after reset cannot be changed unless bit 5 of the particular function select register 1 (see Figure 15) is set to "1."

Fig. 10 Interrupt control register bit configuration

Table 3. Addresses of interrupt control registers

Interrupt control registers	Addresses
$\overline{INT_4}$ interrupt control register	00006E16
$\overline{INT_3}$ interrupt control register	00006F16
A-D interrupt control register	000070₁₆
UART0 transmit interrupt control register	000071₁₆
UART0 receive interrupt control register	000072₁₆
UART1 transmit interrupt control register	000073₁₆
UART1 receive interrupt control register	000074₁₆
Timer A0 interrupt control register	000075₁₆
Timer A1 interrupt control register	000076₁₆
Timer A2 interrupt control register	000077₁₆
Timer A3 interrupt control register	000078₁₆
Timer A4 interrupt control register	000079₁₆
Timer B0 interrupt control register	00007A16
Timer B1 interrupt control register	00007B16
Timer B2 interrupt control register	00007C16
$\overline{INT_0}$ interrupt control register	00007D16
$\overline{INT_1}$ interrupt control register	00007E16
$\overline{INT_2}$ interrupt control register	00007F16

Interrupts caused by a BRK instruction and when dividing by zero are software interrupts and are not included in this list.

Other interrupts previously mentioned are A-D converter, UART, etc. interrupts. The priority of these interrupts can be changed by changing the priority level in the corresponding interrupt control register by software.

Figure 12 shows a diagram of the interrupt priority detection circuit When an interrupt is caused, each interrupt device compares its own priority with the priority from above and if its own priority is higher, then it sends the priority below and requests the interrupt. If the priorities are the same, the one above has priority.

This comparison is repeated to select the interrupt with the highest priority among the interrupts that are being requested. Finally the selected interrupt is compared with the processor interrupt priority level (IPL) contained in the processor status register (PS) and the request is accepted if it is higher than IPL and the interrupt disable flag I is “0”. The request is not accepted if flag I is “1”. The reset, $\overline{DBC}$, and watchdog timer interrupts are not affected by the interrupt disable flag I.

When an interrupt is accepted, the contents of the processor status register (PS) is saved to the stack and the interrupt disable flag I is set to “1”.

Furthermore, the interrupt request bit of the accepted interrupt is cleared to “0” and the processor interrupt priority level (IPL) in the processor status register (PS) is replaced by the priority level of the accepted interrupt.

Therefore, multi-level priority interrupts are possible by resetting the interrupt disable flag I to “0” and enable further interrupts.

For reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction interrupts, which do not have an interrupt control register, the processor interrupt level (IPL) is set as shown in Table 4.

The interrupt request bit and the interrupt priority level of each interrupt source are sampled and latched at each operation code fetch cycle while ϕ_{BIU} is “H”. However, no sampling pulse is generated until the cycles whose number is selected by software has passed, even if the next operation code fetch cycle is generated. The detection of an interrupt which has the highest priority is performed during that time.

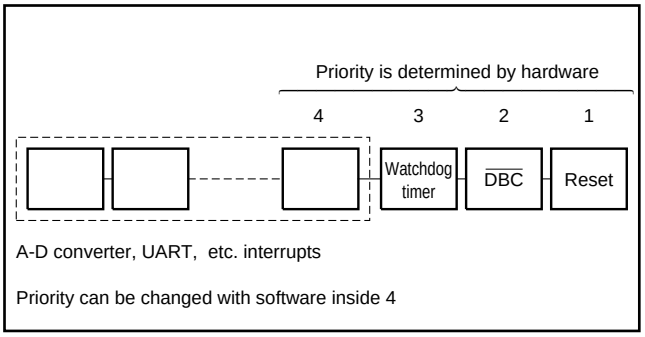


Fig. 11 Interrupt priority

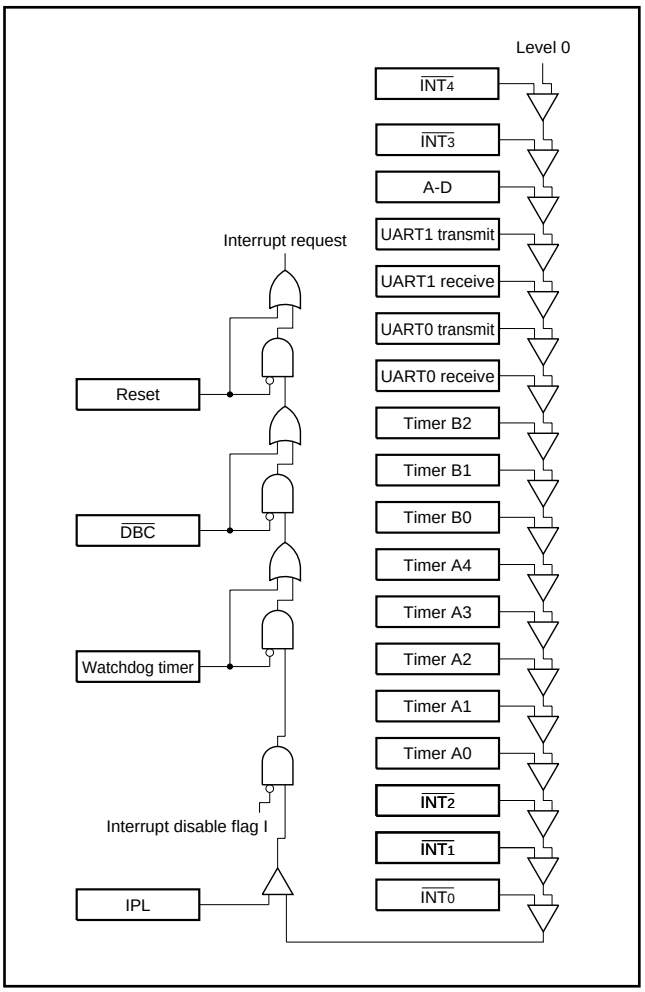


Fig. 12 Interrupt priority detection

As shown in Figure 13, there are three different interrupt priority detection time from which one is selected by software. After the selected time has elapsed, the highest priority is determined and is processed after the currently executing instruction has been completed.

The time is selected with bits 4 and 5 of the processor mode register 0 (address $5E_{16}$) shown in Figure 14. Table 5 shows the relationship between these bits and the number of cycles. After a reset, the processor mode register 0 is initialized to "00_{16}." Therefore, the longest time is automatically set, however, the shortest time must be selected by software.

Table 4. Value set in processor interrupt level (IPL) during an interrupt

Interrupt types	Setting value
Reset	0
$\overline{DBC}$	7
Watchdog timer	7
Zero divide	Not change value of IPL.
BRK instruction	Not change value of IPL.

Table 5. Relationship between interrupt priority detection time select bit and number of cycles

Priority detection time select bit		Number of cycles
Bit 5	Bit 4	
0	0	7 cycles of ϕBIU
0	1	4 cycles of ϕBIU
1	0	2 cycles of ϕBIU

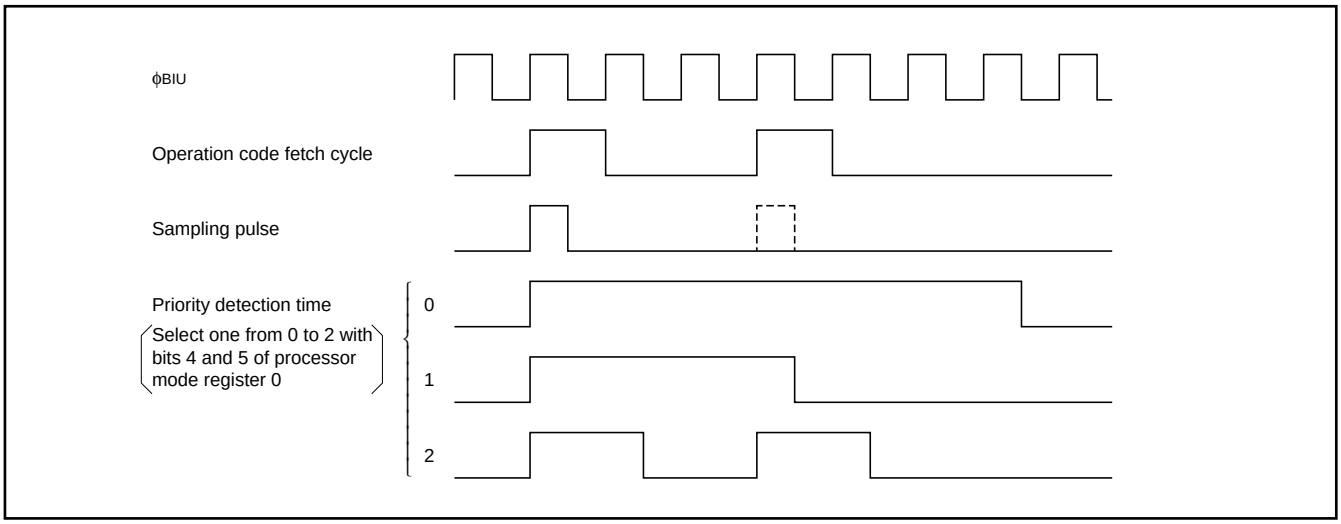


Fig. 13 Interrupt priority detection time

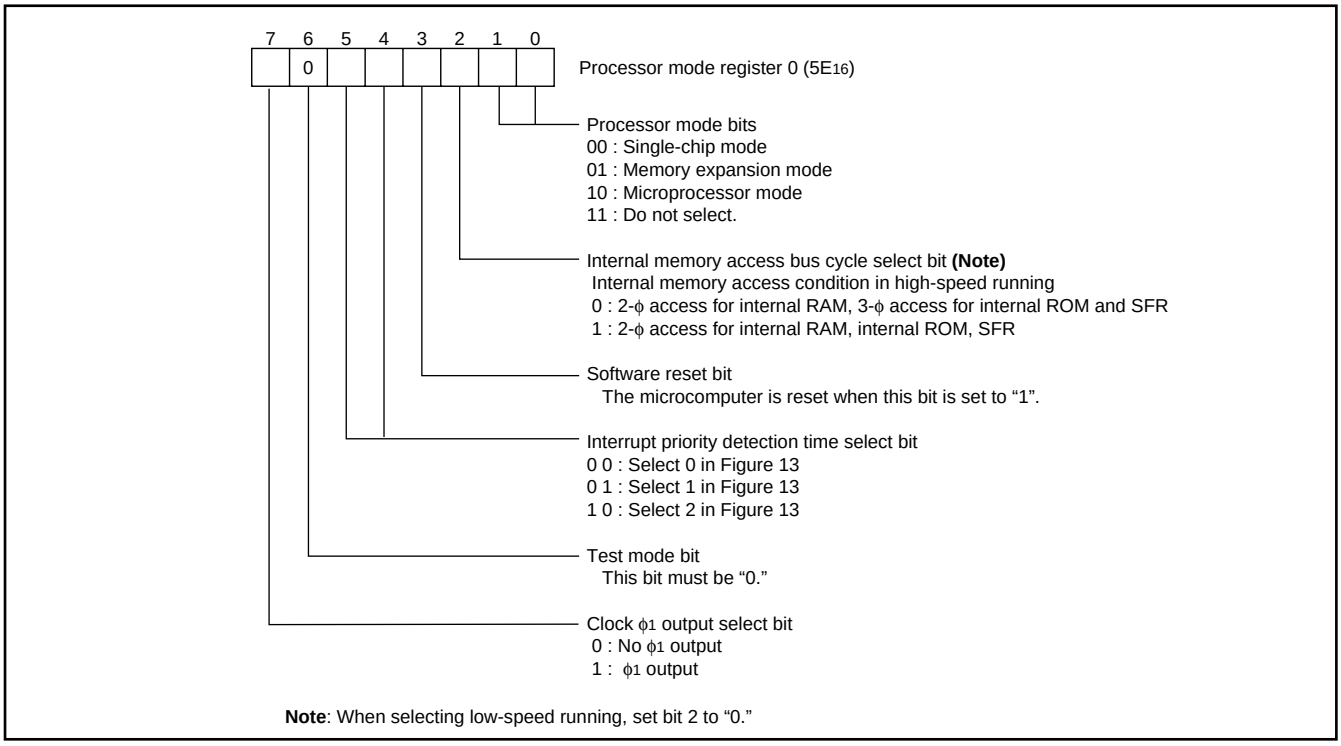


Fig. 14 Processor mode register 0 bit configuration

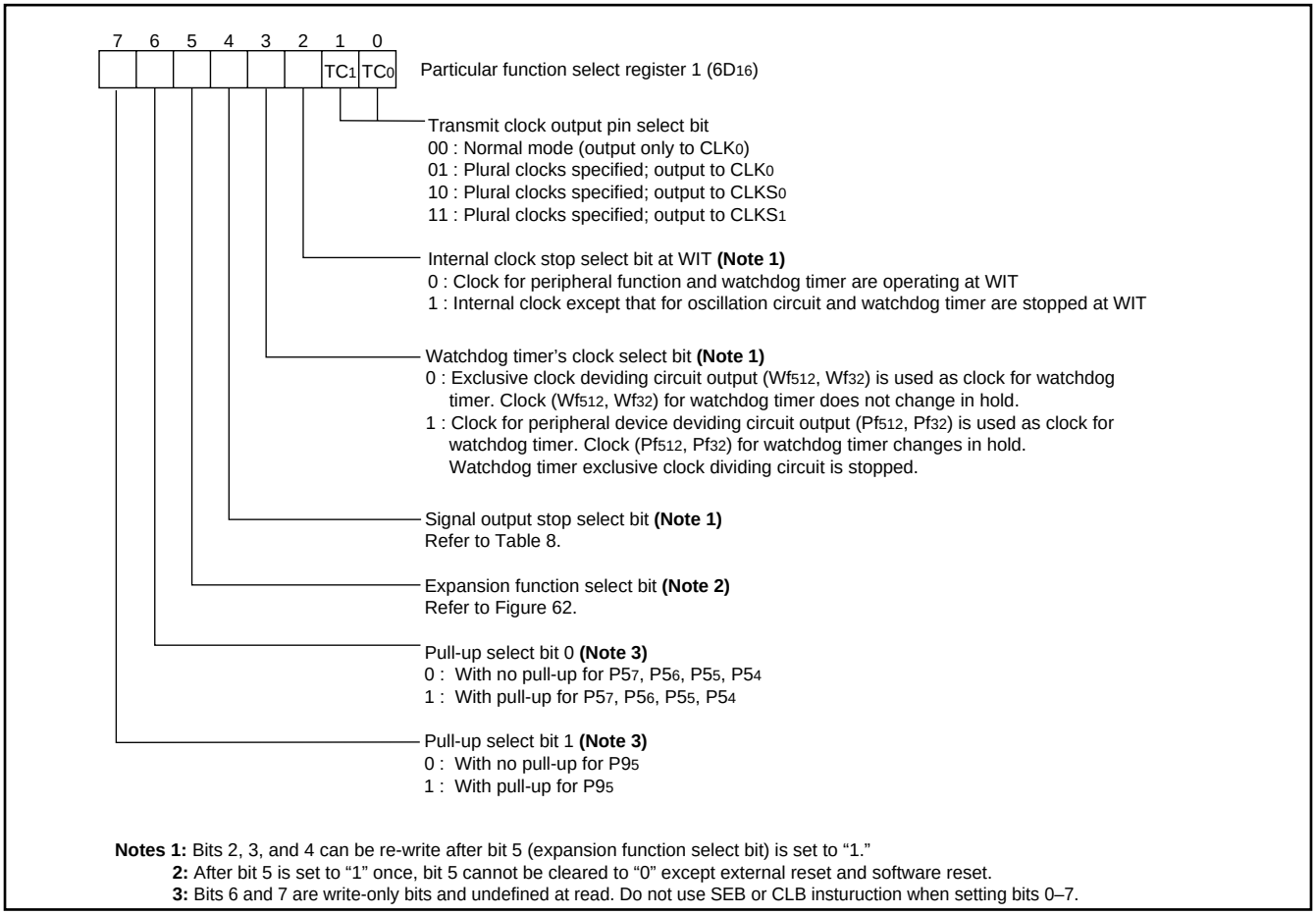


Fig. 15 Processor mode register 0 bit configuration

The $\overline{INT_3}$ interrupt can function as the key input interrupt by setting bits 7 and 6 of the $\overline{INT_3}$ interrupt control register. The key input interrupt uses inputs $\overline{KI_3}$ to $\overline{KI_0}$ or inputs $\overline{KI_4}$ to $\overline{KI_0}$. Figure 10 shows the interrupt control register bit configuration. Figure 15 shows the particular function select register 1 bit configuration, and Figure 16 shows the $\overline{INT_3}$/key input interrupt input circuit block diagram.

When the $\overline{INT_3}$ interrupt control register's bit 7 is "0" and its bit 6 is "0", a signal from the $\overline{INT_3}$ pin is connected to the $\overline{INT_3}$ interrupt control circuit and $\overline{INT_3}$ external interrupt is normally performed.

When the $\overline{INT_3}$ interrupt control register's bit 7 is "1" and its bit 6 is "0", signals from the $\overline{KI_3}$ to $\overline{KI_0}$ pins, which correspond to ports $P5_7$ to $P5_4$, are inverted and then the logical sum of these signals is connected to the $\overline{INT_3}$ interrupt control circuit. In this case, the external interrupt which uses the $\overline{KI_3}$ to $\overline{KI_0}$ pins is performed.

When the $\overline{INT_3}$ interrupt control register's bit 7 is "1" and its bit 6 is "1", signals from the $\overline{KI_4}$ pin, which corresponds to port $P9_5$, $\overline{KI_3}$ to $\overline{KI_0}$ pins, which correspond to ports $P5_7$ to $P5_4$, are inverted and then the logical sum of these signals is connected to the $\overline{INT_3}$ interrupt control circuit. In this case, the external interrupt which uses the $\overline{KI_4}$ to $\overline{KI_0}$ pins is performed.

When using the above key input interrupt, select the edge sense which uses the falling edge from "H" to "L" with the $\overline{INT_3}$ interrupt control register so that an interrupt request can occur by inputting "L" to each of the $\overline{KI_3}$ to $\overline{KI_0}$ pins or the $\overline{KI_4}$ to $\overline{KI_0}$ pins. The interrupt vector is common to the $\overline{INT_3}$ interrupt's one. Additionally, pull-up resistor (transistors) can be added to the $\overline{KI_4}$ to $\overline{KI_0}$ pins by setting the contents of the particular function select register 1's bits 7 and 6 and setting "0" to each bit of the corresponding port's direction register.

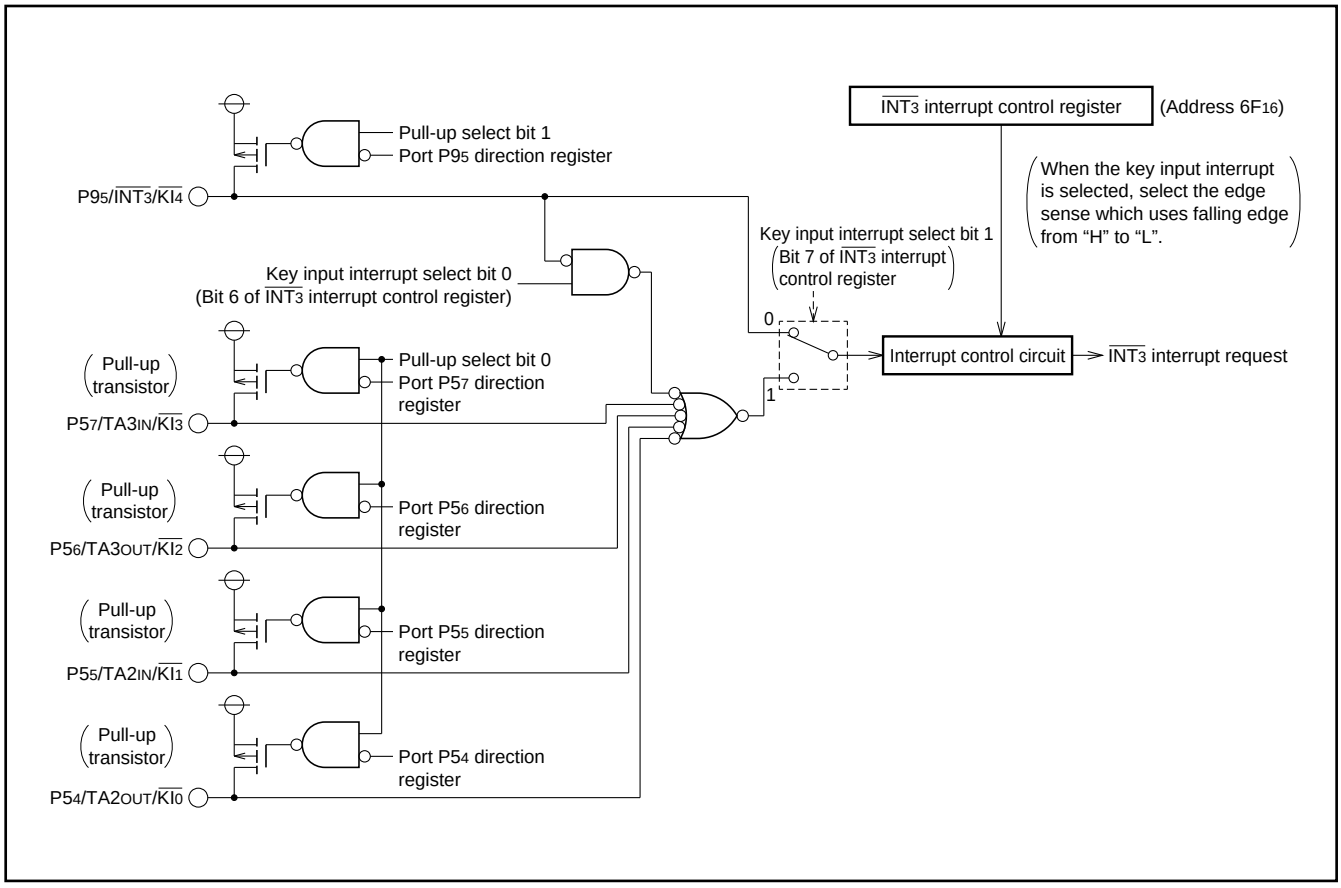


Fig. 16 $\overline{INT_3}$/key input interrupt input circuit block diagram

TIMER

There are eight 16-bit timers. They are divided by type into timer A(5) and timer B(3).

The timer I/O pins are multiplexed with I/O pins for port P5 and P6. To use these pins as timer input pins, the data direction register bit corresponding to the pin must be cleared to "0" to specify input mode.

TIMER A

Figure 17 shows a block diagram of timer A.

Timer A has four modes: timer mode, event counter mode, one-shot pulse mode, and pulse width modulation mode. The mode is selected with bits 0 and 1 of the timer Ai mode register (i = 0 to 4). Each of these modes is described below.

(1) Timer mode [00]

Figure 18 shows the bit configuration of the timer Ai mode register during timer mode. Bits 0 and 1 of the timer Ai mode register must be "0" in timer mode. Bits 3, 4, and 5 are used to select the gate function. Bits 4 and 5 must be "0" when not selecting the gate function. Bit 3 is ignored if bit 4 is "0".

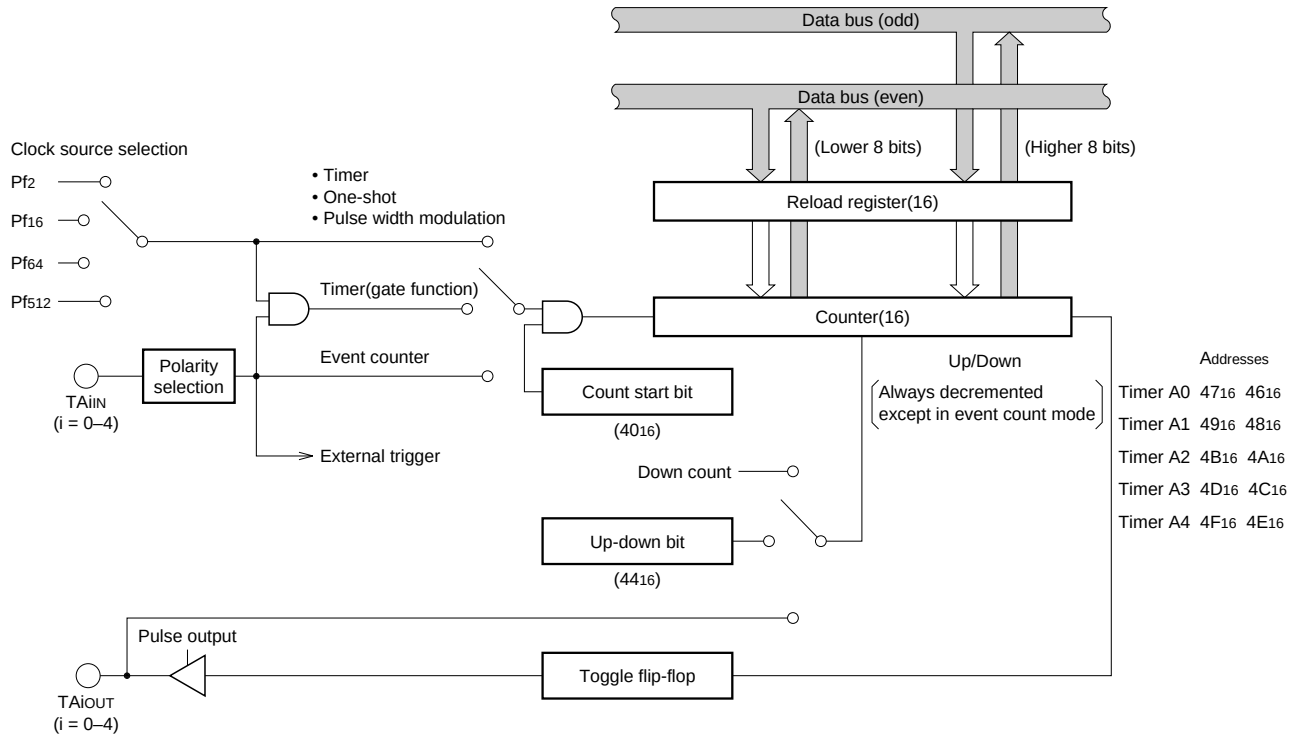
Bits 6 and 7 are used to select the timer counter source.

The counting of the selected clock starts when the count start bit is "1" and stops when it is "0".

Figure 19 shows the bit configuration of the count start bit. The counter is decremented, an interrupt is caused and the interrupt request bit in the timer Ai interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is transferred to the counter and count is continued.

When data is written to timer Ai register with timer Ai halted, the same data is also written to the reload register and the counter. When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The new data is reloaded from the reload register to the counter at the next reload time and counting continues. The contents of the counter can be read at any time.

When the value set in the timer Ai register is n, the timer frequency dividing ratio is 1/(n+1).



Note: Perform write and read to/from timer Ai register in the condition of 16-bit data length : data length flag (m) = "0".

Fig. 17 Block diagram of timer A

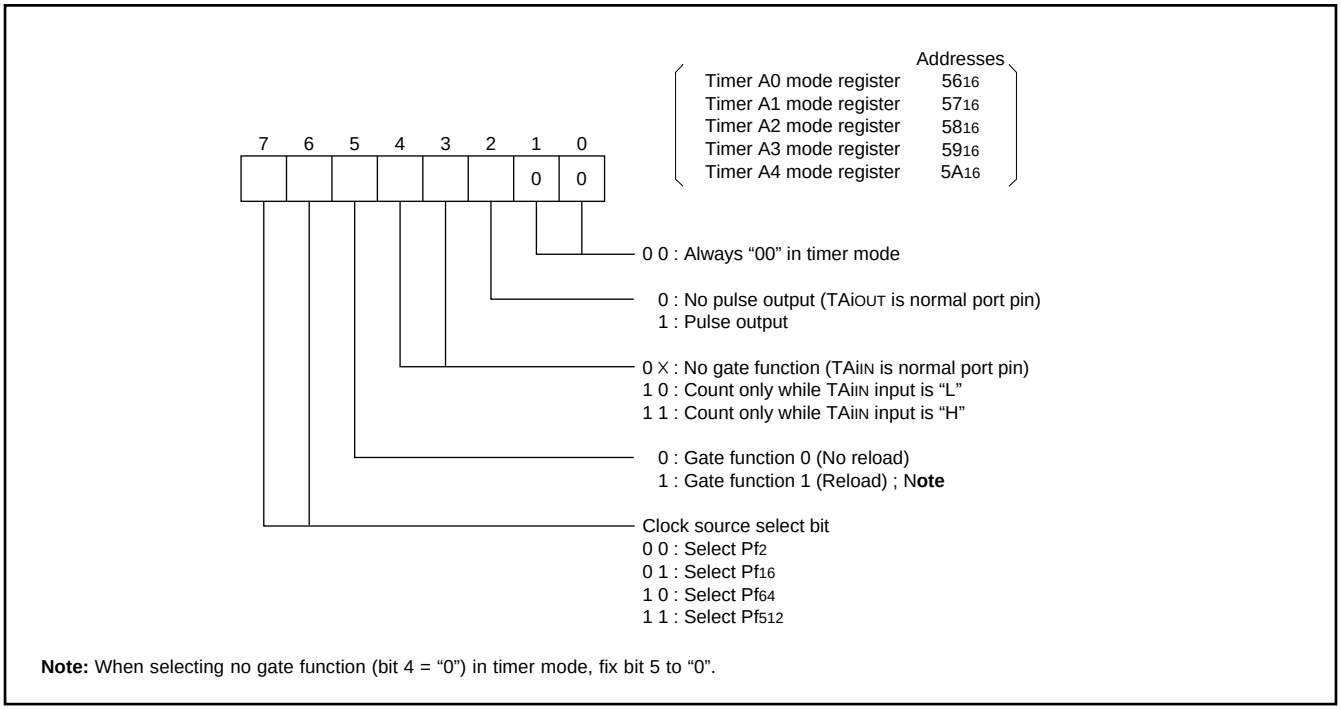
Pulse output function

When bit 2 of the timer Ai mode register is “1”, the output is generated from TAiOUT pin. The output is toggled each time the contents of the counter reaches to 0000$_{16}$. When the contents of the count start bit is “0”, “L” is output from TAiOUT pin.
When bit 2 is “0”, TAiOUT can be used as a normal port pin. When bit 4 is “0”, TAiIN can be used as a normal port pin.

Gate function

When bit 4 is “1”, counting is performed only while the input signal from the TAiIN pin is “H” or “L” as shown in Figure 20. Therefore, this can be used to measure the pulse width of the TAiIN input signal. Whether to count while the input signal is “H” or while it is “L” is determined by bit 3. If bit 3 is “1”, counting is performed while the TAiIN pin input signal is “H” and if bit 3 is “0”, counting is performed while it is “L”.

When bit 5 is “0, counting restarts from the value which is contained at restarting (gate function 0 [no reload]) and an overflow occurs (n + 1) cycles of the count source later. Figure 21 shows that operation.
When bit 5 is “1”, counting restarts from the value which is obtained by reload at restarting (gate function 1 [reload]) and the first overflow occurs (n + 2) cycles of the count source later. Figure 22 shows that operation. After that, while the input signal from the TAiIN pin keeps valid level, an overflow occurs at (n + 1)- cycle intervals. Make sure to set the value of 1 or more to n.
When gate functions are used, the duration of “H” or “L” on the TAiIN pin must be 2 or more cycles of the timer count source.



Note: When selecting no gate function (bit 4 = “0”) in timer mode, fix bit 5 to “0”.

Fig. 18 Timer Ai mode register bit configuration during timer mode

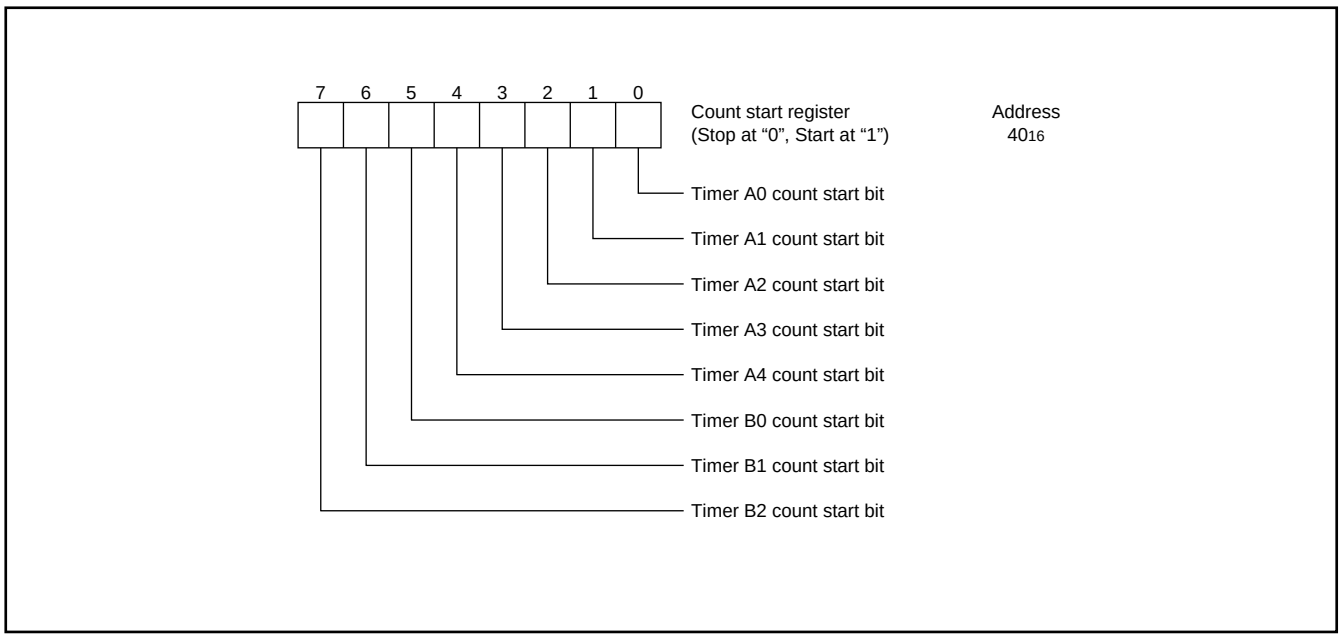


Fig. 19 Count start flag bit configuration

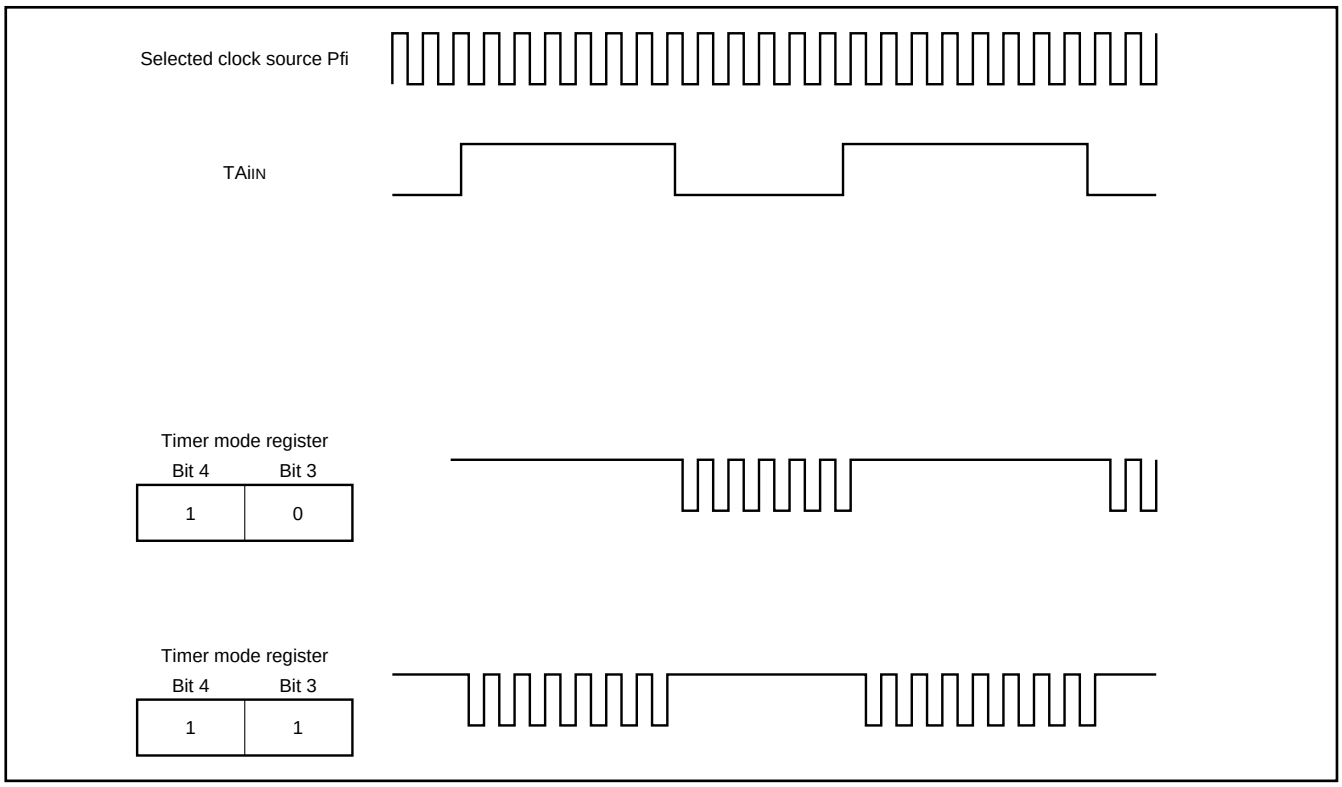


Fig. 20 Count waveform when gate function is available

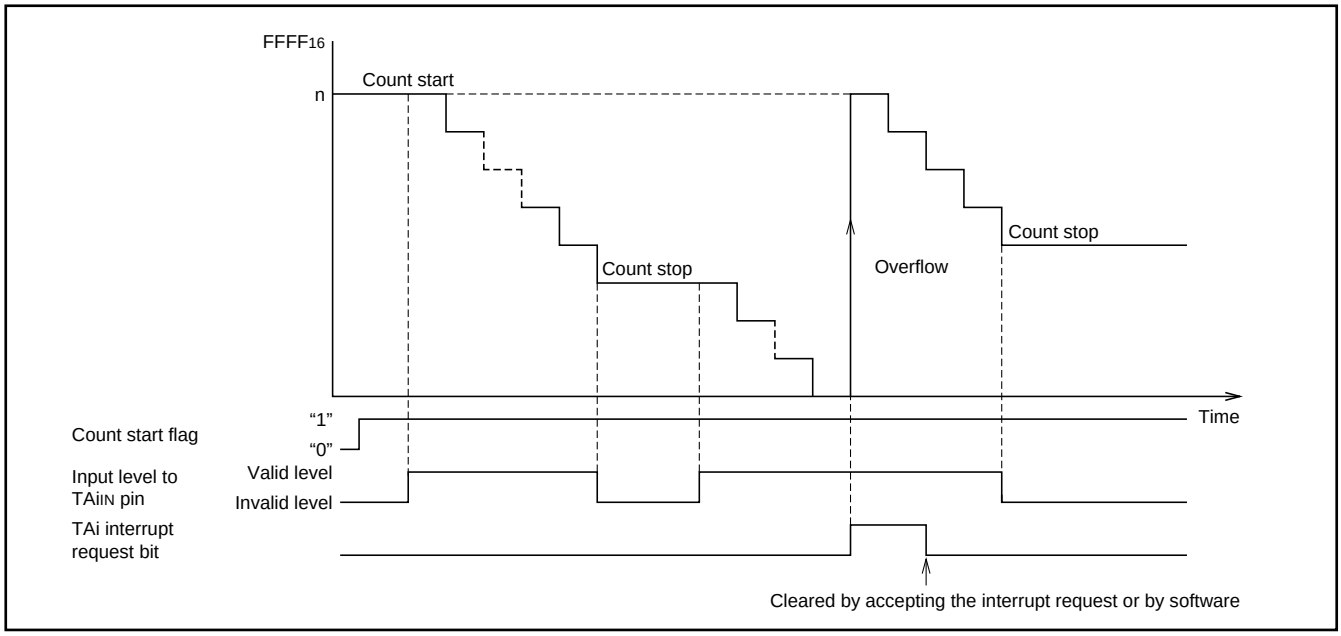


Fig. 21 Timer operation example with gate function 0 (no reload) selected

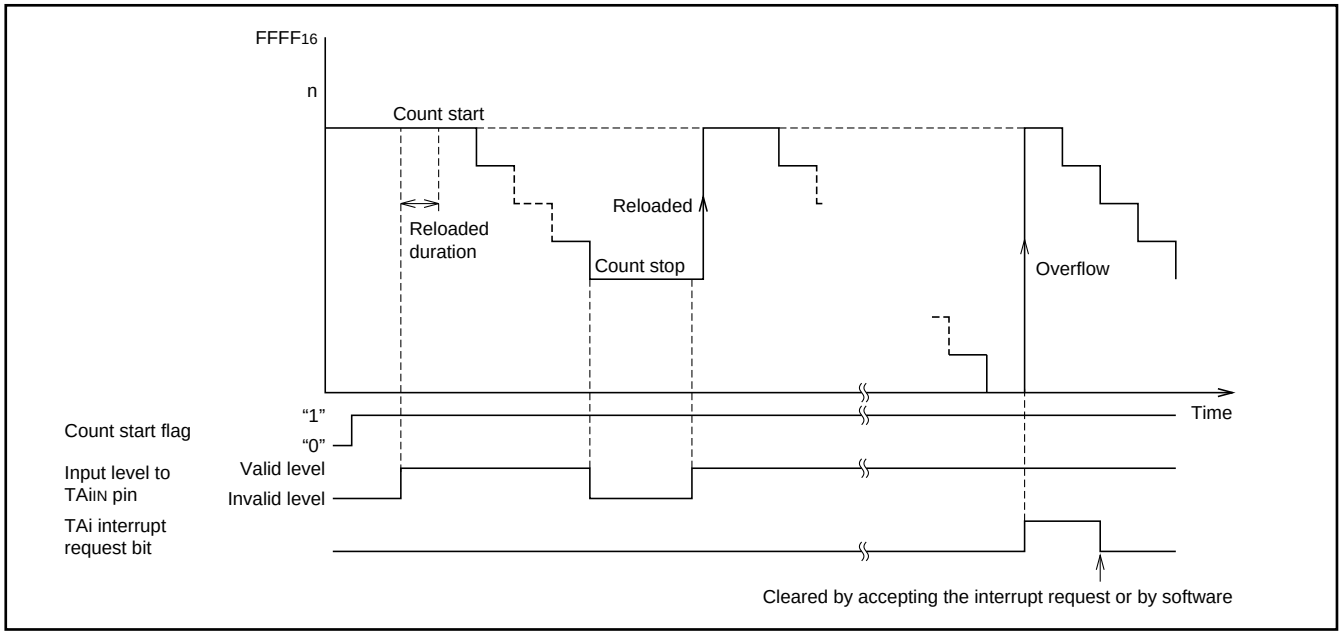


Fig. 22 Timer operation example with gate function 1 (reload) selected

(2) Event counter mode [01]

Figure 23 shows the bit configuration of the timer Ai mode register during event counter mode. In event counter mode, bit 0 of the timer Ai mode register must be "1" and bits 1 and 5 must be "0".

The input signal from the TAiIN pin is counted when the count start bit shown in Figure 19 is "1" and counting is stopped when it is "0".

Count is performed at the fall of the input signal when bit 3 is "0" and at the rise of the signal when it is "1".

In event counter mode, whether to increment or decrement the count can be selected with the up-down bit or the input signal from the TAiOUT pin.

When bit 4 of the timer Ai mode register is "0", the up-down bit is used to determine whether to increment or decrement the count (decrement when the bit is "0" and increment when it is "1"). Figure 24 shows the bit configuration of the up-down register.

When bit 4 of the timer Ai mode register is "1", the input signal from the TAiOUT pin is used to determine whether to increment or decrement the count. However, note that bit 2 must be "0" if bit 4 is "1." It is because if bit 2 is "1", TAiOUT pin becomes an output pin to output pulses.

The count is decremented when the input signal from the TAiOUT pin is "L" and incremented when it is "H". Determine the level of the input signal from the TAiOUT pin before a valid edge is input to the TAiIN pin.

An interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set when the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count). At the same time, the contents of the reload register is transferred to the counter and the count is continued.

When bit 2 is "1," each time the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$(increment count), the waveform's polarity is reversed and is output from TAiOUT pin.

If bit 2 is "0", TAiOUT pin can be used as a normal port pin.

However, if bit 4 is "1" and the TAiOUT pin is used as an output pin, the output from the pin changes the count direction. Therefore, bit 4 must be "0" unless the output from the TAiOUT pin is to be used to select the count direction.

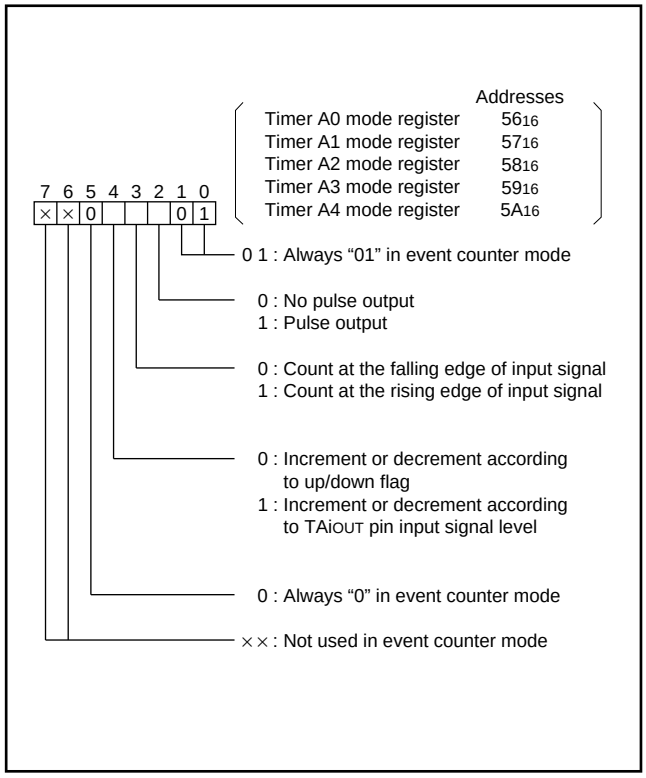


Fig. 23 Timer Ai mode register bit configuration during event counter mode

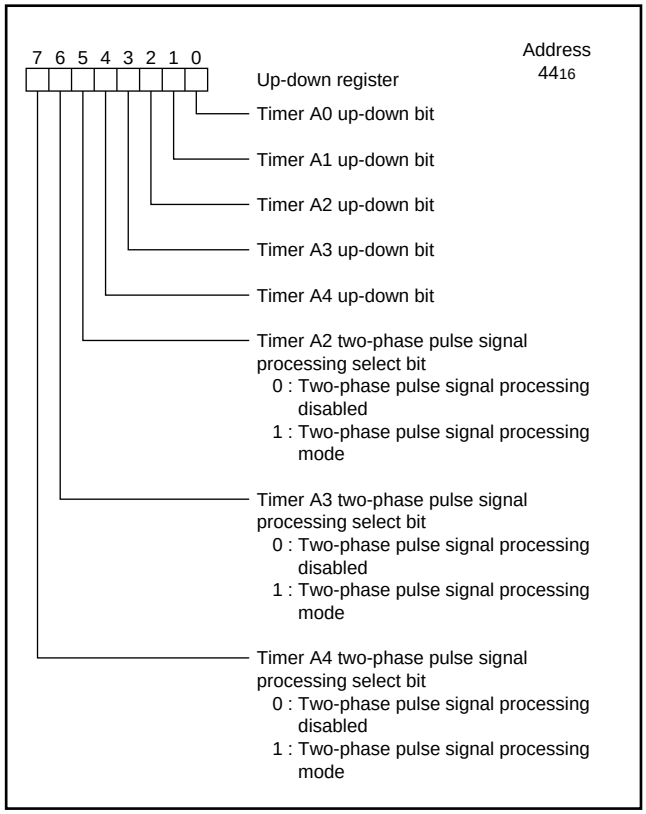


Fig. 24 Up-down register bit configuration

Data write and data read are performed in the same way as for timer mode. That is, when data is written to timer Ai halted, it is also written to the reload register and the counter. When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The counter can be read at any time.

Two-phase pulse processing

In event counter mode, whether to increment or decrement the counter can also be determined by supplying two kinds of pulses of which phases differ by 90° to timer A2, A3, or A4. There are two types of two-phase pulse processing operations. One uses timers A2 and A3, and the other uses timer A4. In both processing operations, two pulses described above are input to the TAjOUT (j = 2 to 4) pin and TAjIN pin respectively.

When timers A2 and A3 are used, as shown in Figure 25, the count is incremented when a rising edge is input to the TAkIN pin after the level of TAkOUT(k=2,3) pin changes from “L” to “H”, and when the falling edge is input, the count is decremented.

For timer A4, as shown in Figure 26, when a phase-related pulse with a rising edge input to the TA4IN pin is input after the level of TA4OUT pin changes from “L” to “H”, the count is incremented at the respective rising edge and falling edge of the TA4OUT pin and TA4IN pin.

When a phase-related pulse with a falling edge input to the TA4OUT pin is input after the level of TA4IN pin changes from “H” to “L”, the count is decremented at the respective rising edge and falling edge of the TA4IN pin and TA4OUT pin. When performing this two-phase pulse signal processing, timer Aj mode register bit 0 and bit 4 must be set to “1” and bits 1, 2, 3, and 5 must be “0”. Bits 6 and 7 are ignored. Note that bits 5, 6, and 7 of the up-down register (44_{16}) are the two-phase pulse signal processing select bits for timers A2, A3 and A4 respectively. Each timer operates in normal event counter mode when the corresponding bit is “0” and performs two-phase pulse signal processing when it is “1”.

Count is started by setting the count start bit to “1”. Data write and read are performed in the same way as for normal event counter mode. Note that the direction register of the input port must be set to input mode because two kinds of pulse signals, described above, are input. Also, there can be no pulse output in this mode.

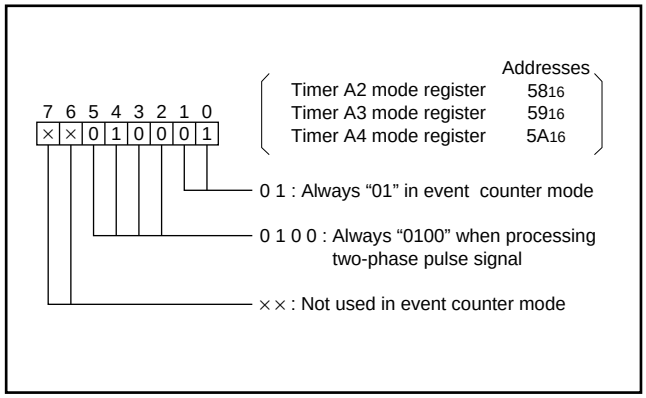


Fig. 27 Timer Aj mode register bit configuration when performing two-phase pulse signal processing in event counter mode

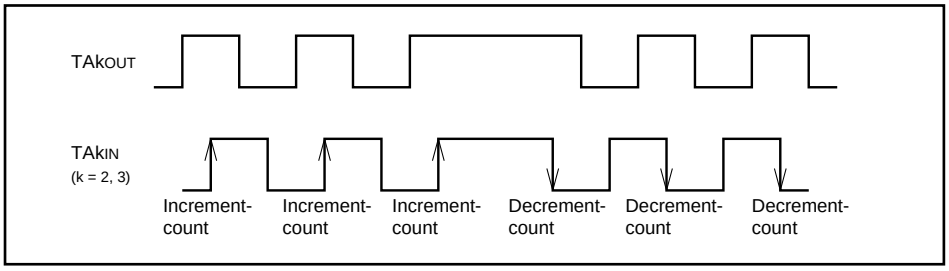


Fig. 25 Two-phase pulse processing operation of timers A2 and timer A3

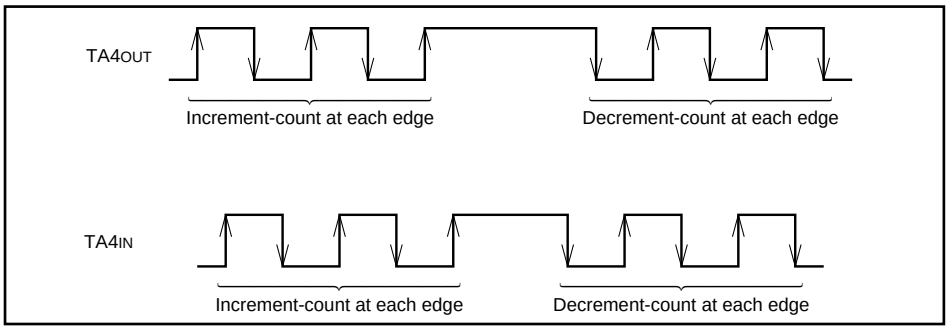


Fig. 26 Two-phase pulse processing operation of timer A4

(3) One-shot pulse mode [10]

Figure 28 shows the bit configuration of the timer Ai mode register during one-shot pulse mode. In one-shot pulse mode, bit 0 and bit 5 must be “0” and bit 1 and bit 2 must be “1”.

The trigger is enabled when the count start bit is “1”. The trigger can be generated by software or it can be input from the TAiIN pin. Software trigger is selected when bit 4 is “0” and the input signal from the TAiIN pin is used as the trigger when it is “1“.

Bit 3 is used to determine whether to trigger at the fall of the trigger signal or at the rise. The trigger is at the fall of the trigger signal when bit 3 is “0” and at the rise of the trigger signal when it is “1”.

Software trigger is generated by setting the bit in the one-shot start bit corresponding to each timer.

Figure 29 shows the bit configuration of the one-shot start register.

As shown in Figure 30, when a trigger signal is received, the counter counts the clock selected by bits 6 and 7.

If the contents of the counter is not 0000_{16}, the TAiOUT pin goes “H” when a trigger signal is received. The count direction is decrement. When the counter reaches 0001_{16}, The TAiOUT pin goes “L” and count is stopped. The contents of the reload register is transferred to the counter. At the same time, an interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set. This is repeated each time a trigger signal is received. The output pulse width is

$$\frac{1}{\text{pulse frequency of the selected clock}} \times (\text{counter's value at the time of trigger}).$$

If the count start flag is “0”, TAiOUT goes “L”. Therefore, the value corresponding to the desired pulse width must be written to timer Ai before setting the timer Ai count start bit.

As shown in Figure 31, a trigger signal can be received before the operation for the previous trigger signal is completed. In this case, the contents of the reload register is transferred to the counter by the trigger and then that value is decremented.

Except when retriggering while operating, the contents of the reload register is not transferred to the counter by triggering.

When retriggering, there must be at least one timer count source cycle before a new trigger can be issued.

Data write is performed in the same way as for timer mode.

When data is written in timer Ai halted, it is also written to the reload register and the counter.

When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time.

Undefined data is read when timer Ai is read.

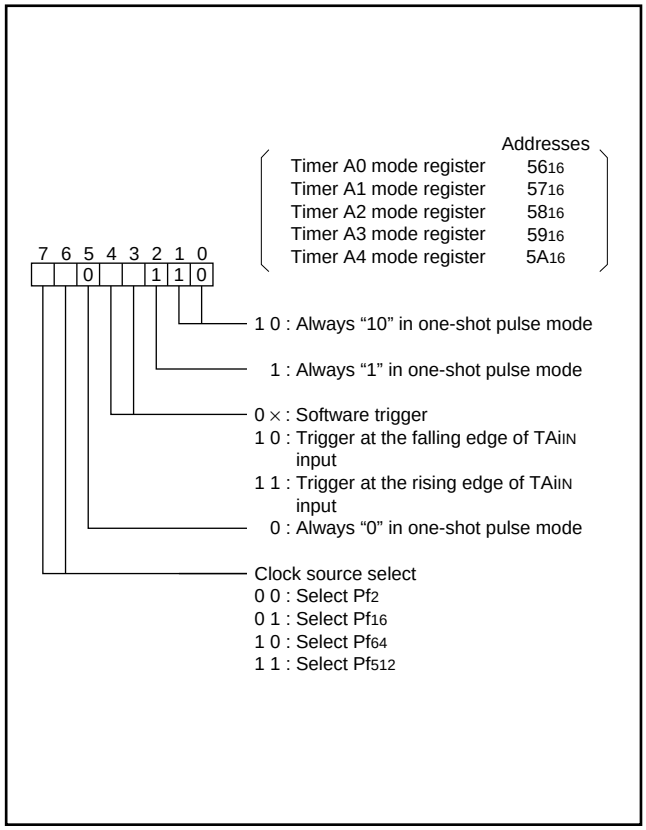


Fig. 28 Timer Ai mode register bit configuration during one-shot pulse mode

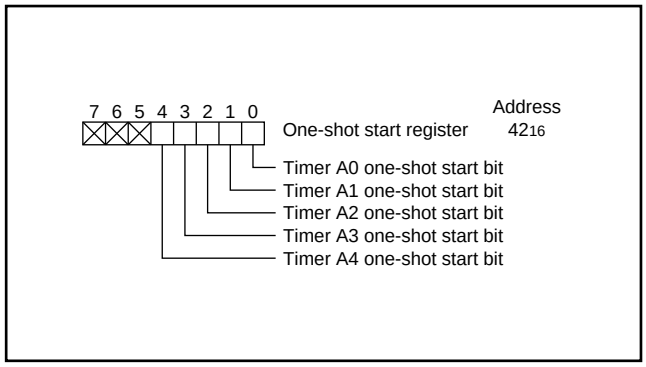


Fig. 29 One-shot start register bit configuration

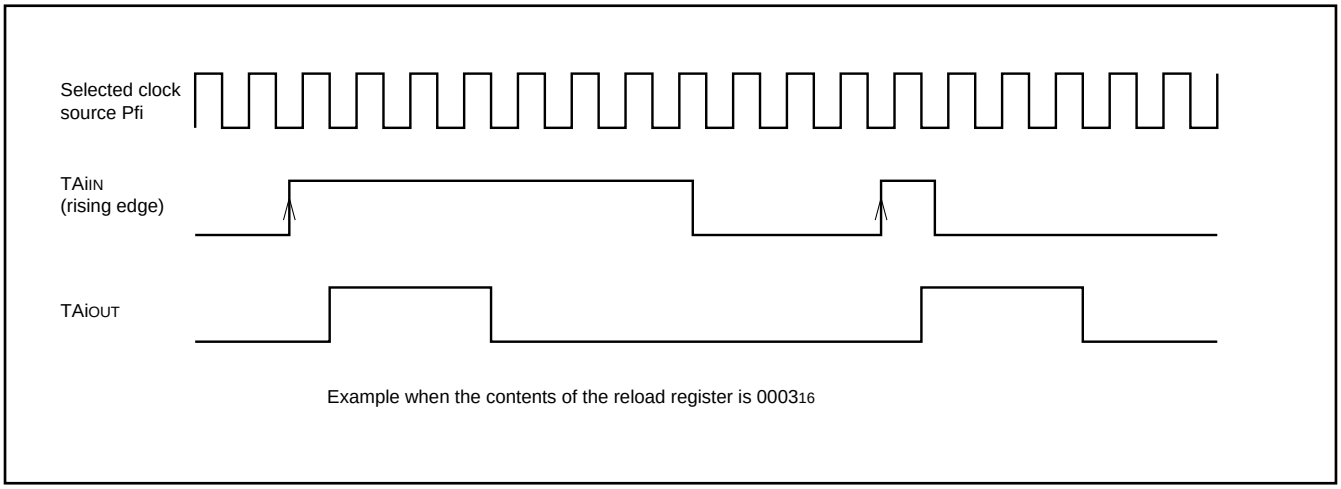


Fig. 30 Pulse output example when external rising edge is selected

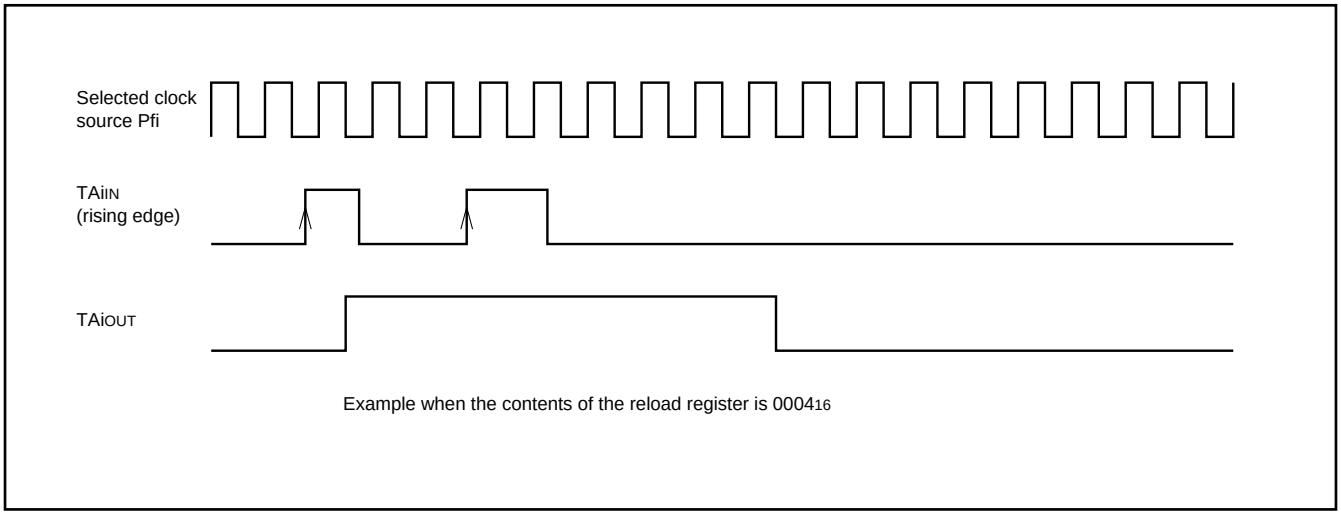


Fig. 31 Example when trigger is re-issued during pulse output

(4) Pulse width modulation mode [11]

Figure 32 shows the bit configuration of the timer Ai mode register during pulse width modulation mode. In pulse width modulation mode, bits 0, 1, and 2 must be set to “1”.

Bit 5 is used to determine whether to perform 16-bit length pulse width modulator or 8-bit length pulse width modulator. 16-bit length pulse width modulator is selected when bit 5 is “0” and 8-bit length pulse width modulator is selected when it is “1”. The 16-bit length pulse width modulator is described first.

The pulse width modulator can be started with a software trigger or with an input signal from a TAiIN pin (external trigger).

The software trigger mode is selected when bit 4 is “0”.

Pulse width modulator is started and a pulse is output from TAiOUT when the timer Ai start bit is set to “1”.

The external trigger mode is selected when bit 4 is “1”.

Pulse width modulation starts when a trigger signal is input from the TAiIN pin when the timer Ai start bit is “1”. Whether to trigger at the fall or rise of the trigger signal is determined by bit 3. The trigger is at the fall of the trigger signal when bit 3 is “0” and at the rise when it is “1”.

When data is written to timer Ai with the pulse width modulator halted, it is written to the reload register and the counter.

Then when the timer Ai start bit is set to “1” and a software trigger or an external trigger is issued to start modulation, the waveform shown in Figure 33 is output continuously.

Once modulation is started, triggers are not accepted. If the value in the reload register is m, the duration “H” of pulse is

$$\frac{1}{\text{selected clock frequency}} \times m$$

and the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (2^{16} - 1).$$

An interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set at each fall of the output pulse.

The width of the output pulse is changed by updating timer data. The update can be performed at any time. The output pulse width is changed at the rise of the pulse after data is written to the timer.

The contents of the reload register are transferred to the counter just before the rise of the next pulse so that the pulse width is changed from the next output pulse.

Undefined data is read when timer Ai is read.

The 8-bit length pulse width modulator is described next.

The 8-bit length pulse width modulator is selected when the timer Ai mode register bit 5 is “1”.

The reload register and the counter are both divided into 8-bit halves. The low-order 8 bits function as a prescaler and the high-order 8 bits function as the 8-bit length pulse width modulator. The prescaler counts the clock selected by bits 6 and 7. A pulse is generated when the counter reaches 0000_{16} as shown in Figure 34. At the same time, the contents of the reload register is transferred to the counter and count is continued.

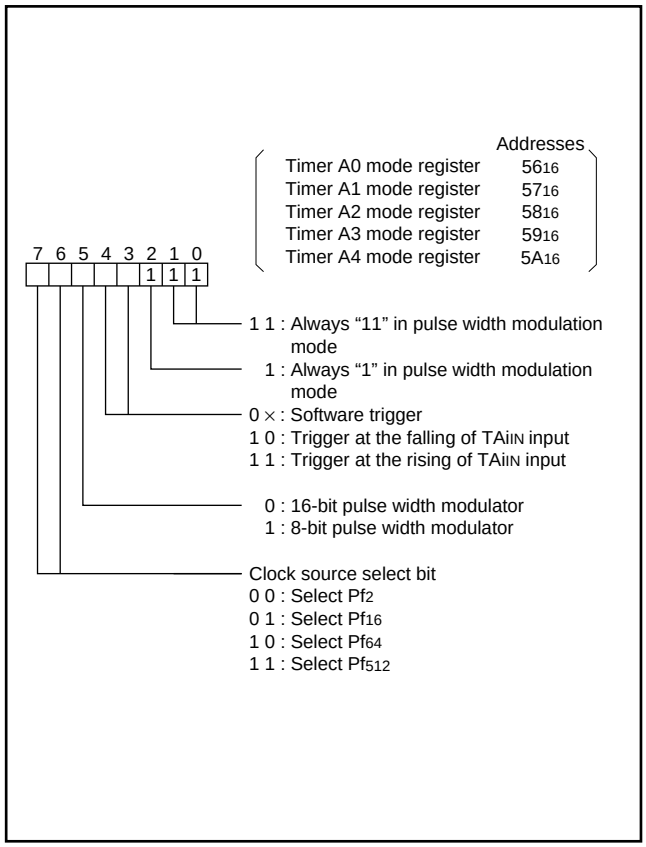


Fig. 32 Timer Ai mode register bit configuration during pulse width modulation mode

Therefore, if the low-order 8 bits of the reload register are n, the period of the generated pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n+1).$$

The high-order 8 bits function as an 8-bit length pulse width modulator using this pulse as input. The operation is the same as for 16-bit length pulse width modulator except that the length is 8 bits. If the high-order 8 bits of the reload register are m, the duration "H" of pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n+1) \times m.$$

And the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (n+1) \times (2^8-1).$$

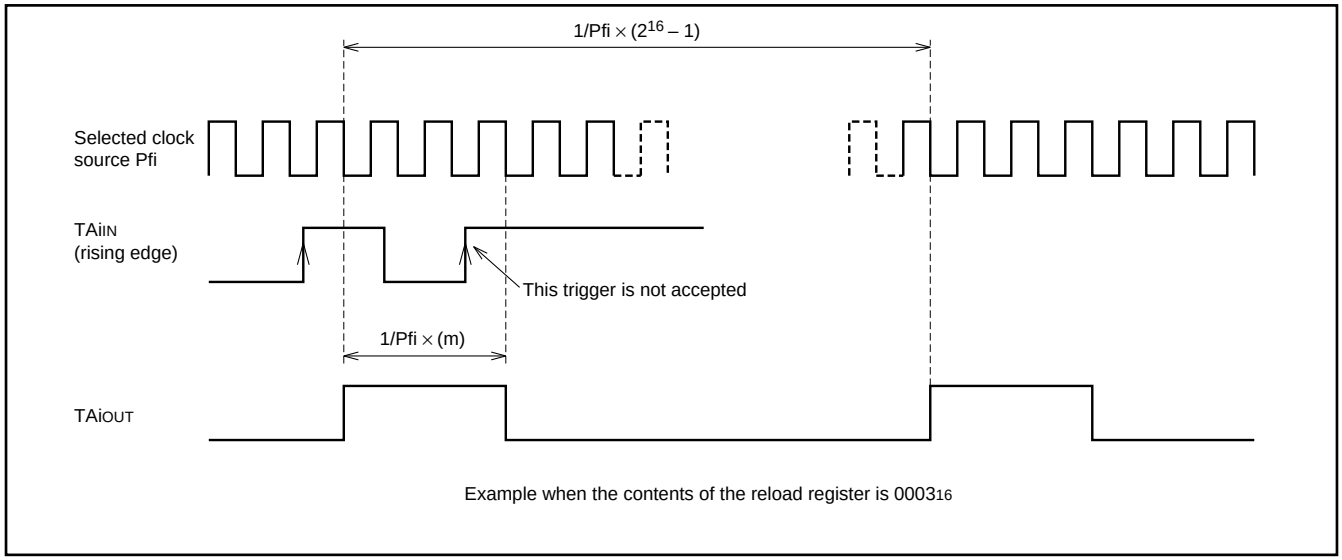


Fig. 33 16-bit length pulse width modulator output pulse example

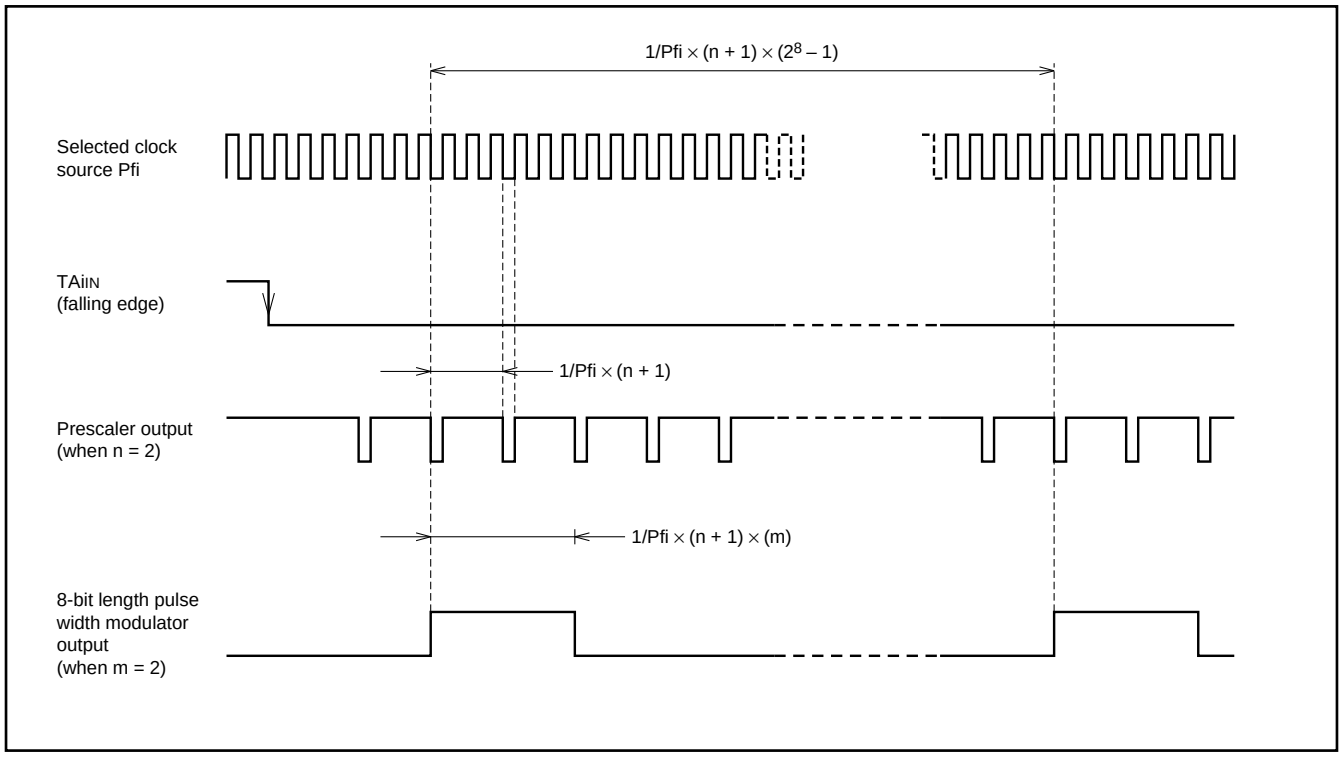


Fig. 34 8-bit length pulse width modulator output pulse example

TIMER B

Figure 35 shows a block diagram of timer B.

Timer B has three modes: timer mode, event counter mode, and pulse period measurement/pulse width measurement mode. The mode is selected with bits 0 and 1 of the timer Bi mode register (i=0 to 2). Each of these modes is described below.

(1) Timer mode [00]

Figure 36 shows the bit configuration of the timer Bi mode register during timer mode. Bits 0 and 1 of the timer Bi mode register must always be “0” in timer mode.

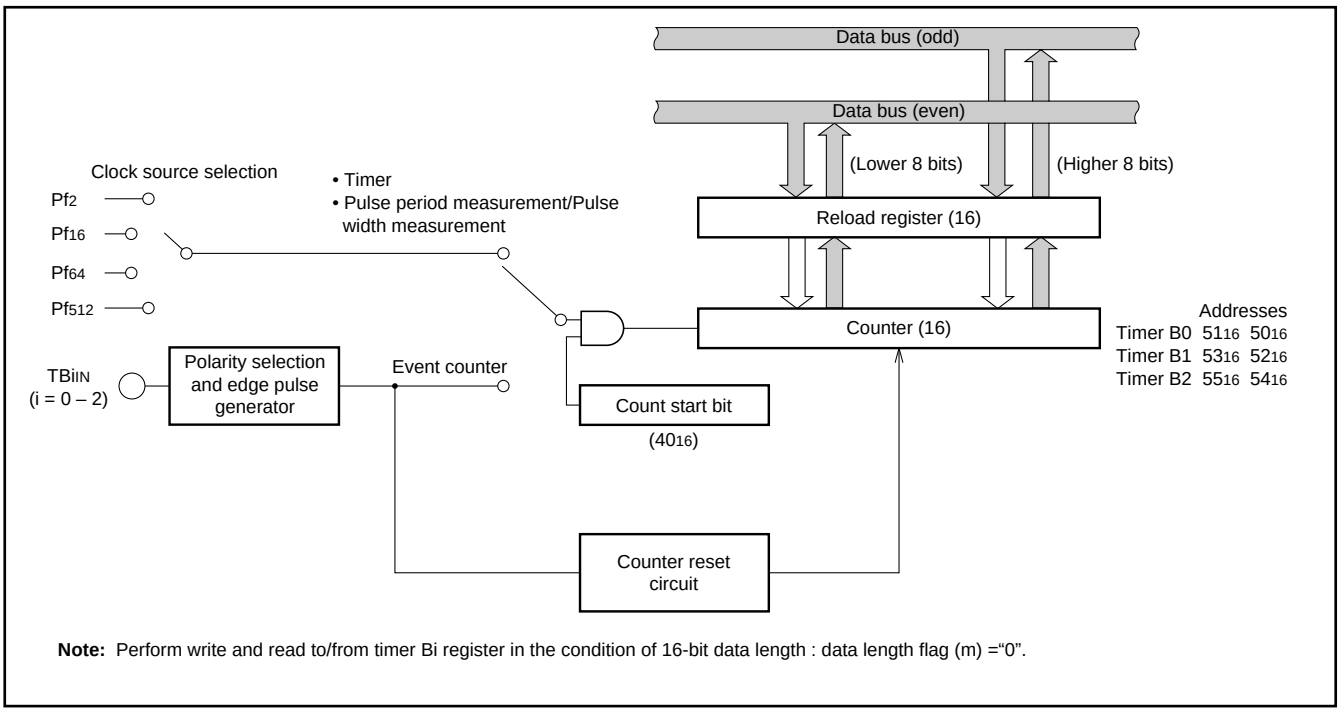
Bits 6 and 7 are used to select the clock source. The counting of the selected clock starts when the count start bit is “1” and stops when “0”.

As shown in Figure 19, the timer Bi count start bit is at the same address as the timer Ai count start bit. The count is decremented, an interrupt occurs, and the interrupt request bit in the timer Bi interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is stored in the counter and count is continued.

Timer Bi does not have a pulse output function or a gate function like timer A.

When data is written to timer Bi halted, it is written to the reload register and the counter. When data is written to timer Bi which is busy, the data is written to the reload register, but not to the counter. The new data is reloaded from the reload register to the counter at the next reload time and counting continues.

The contents of the counter can be read at any time.



Note: Perform write and read to/from timer Bi register in the condition of 16-bit data length : data length flag (m) =“0”.

Fig. 35 Timer B block diagram

(2) Event counter mode [01]

Figure 37 shows the bit configuration of the timer Bi mode register during event counter mode. In event counter mode, bit 0 in the timer Bi mode register must be "1" and bit 1 must be "0".

The input signal from the TBiIN pin is counted when the count start flag is "1" and counting is stopped when it is "0".

Count is performed at the fall of the input signal when bits 2, and 3 are "0" and at the rise of the input signal when bit 3 is "0" and bit 2 is "1".

When bit 3 is "1" and bit 2 is "0", count is performed at the rise and fall of the input signal.

Data write, data read and timer interrupt are performed in the same way as for timer mode.

(3) Pulse period measurement/pulse width measurement mode [10]

Figure 38 shows the bit configuration of the timer Bi mode register during pulse period measurement/pulse width measurement mode.

In pulse period measurement/pulse width measurement mode, bit 0 must be "0" and bit 1 must be "1". Bits 6 and 7 are used to select the clock source. The selected clock is counted when the count start flag is "1" and counting stops when it is "0".

The pulse period measurement mode is selected when bit 3 is "0". In pulse period measurement mode, the selected clock is counted during the interval starting at the fall of the input signal from the TBiIN pin to the next fall or at the rise of the input signal to the next rise; the result is stored in the reload register. In this case, the reload register acts as a buffer register.

When bit 2 is "0", the clock is counted from the fall of the input signal to the next fall. When bit 2 is "1", the clock is counted from the rise of the input signal to the next rise.

In the case of counting from the fall of the input signal to the next fall, counting is performed as follows. As shown in Figure 39, when the fall of the input signal from TBiIN pin is detected, the contents of the counter is transferred to the reload register. Next the counter is cleared and count is started from the next clock. When the fall of the next input signal is detected, the contents of the counter is transferred to the reload register once more, the counter is cleared, and the count is started. The period from the fall of the input signal to the next fall is measured in this way.

After the contents of the counter is transferred to the reload register, an interrupt request signal is generated and the interrupt request bit in the timer Bi interrupt control register is set. However, no interrupt request signal is generated when the contents of the counter is transferred first to the reload register after the count start bit is set to "1".

When bit 3 is "1", the pulse width measurement mode is selected. Pulse width measurement mode is the same as the pulse period measurement mode except that the clock is counted from the fall of the TBiIN pin input signal to the next rise or from the rise of the input signal to the next fall as shown in Figure 40.

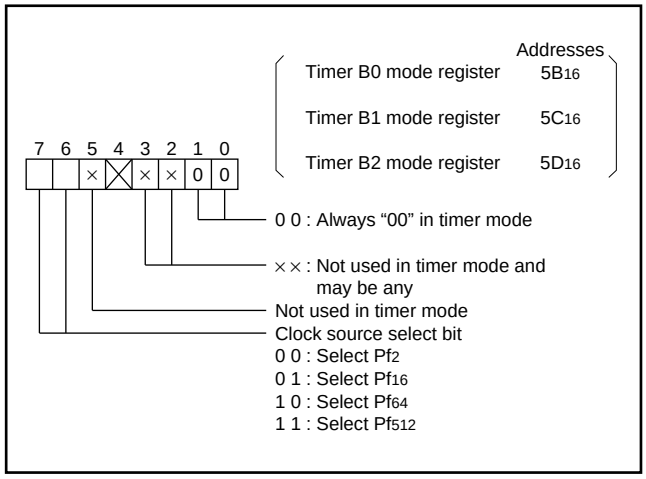


Fig. 36 Timer Bi mode register bit configuration during timer mode

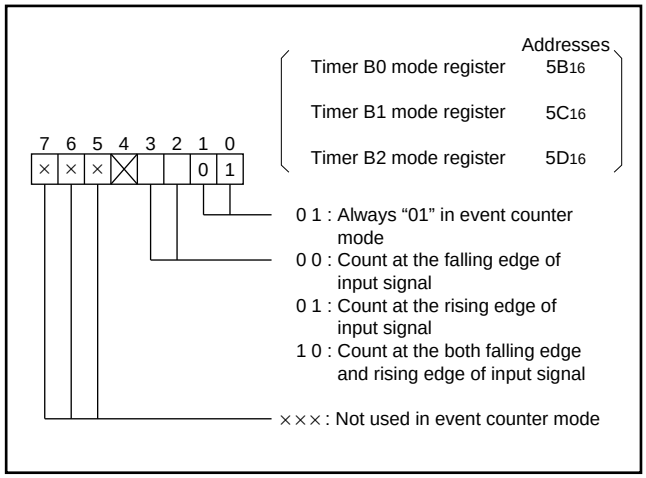


Fig. 37 Timer Bi mode register bit configuration during event counter mode

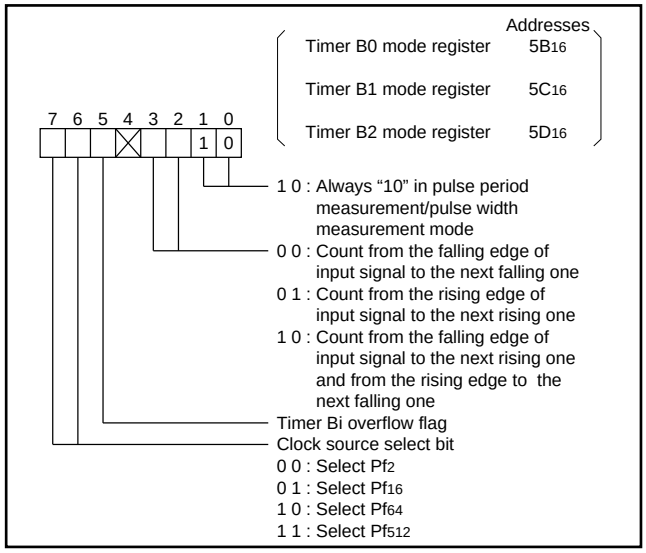


Fig. 38 Timer Bi mode register bit configuration during pulse period measurement/pulse width measurement mode

When timer Bi is read, the contents of the reload register is read.
Note that in this mode, the interval between the fall of the TBiIN pin input signal to the next rise or from the rise to the next fall must be at least two cycles of the timer count source.
Timer Bi overflow flag which is bit 5 of timer Bi mode register is set to "1" when the timer Bi counter reaches 0000_{16}, which indicates that a pulse width or pulse period is longer than that which can be measured by a 16-bit length.
This flag is cleared by writing data to the corresponding timer Bi mode register. This bit is set to "1"at reset.

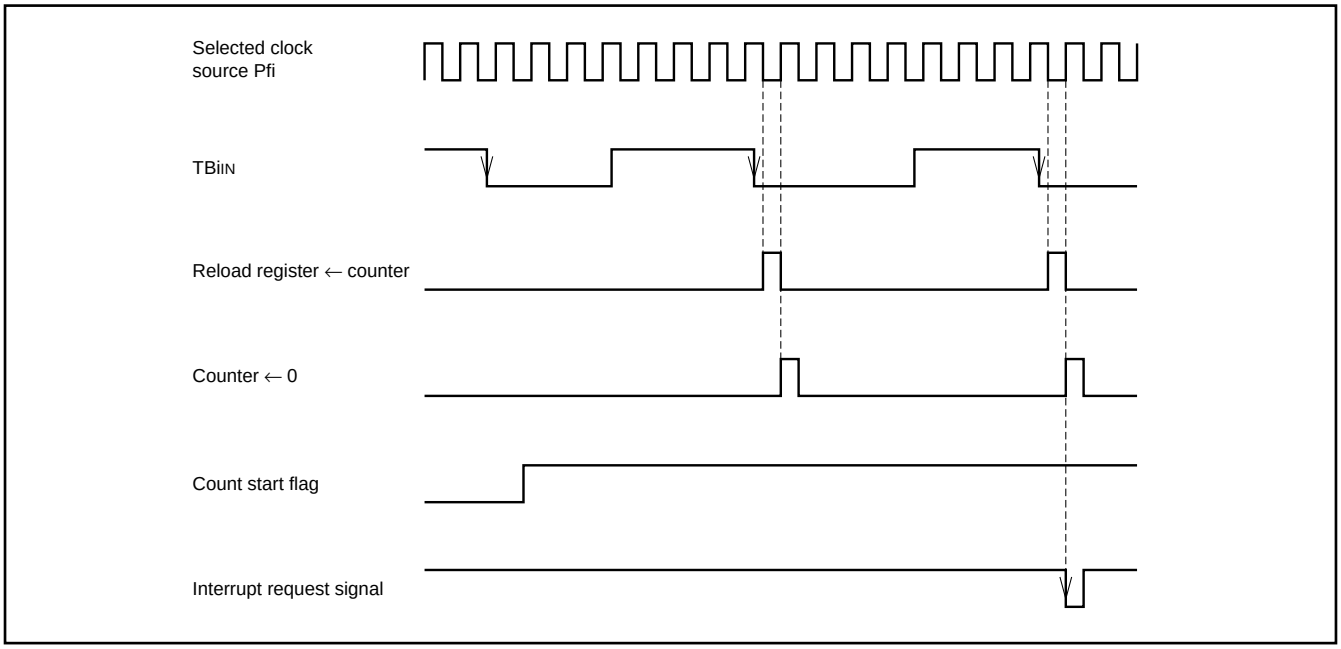


Fig. 39 Pulse period measurement mode operation (example of measuring the interval between the falling edge to next falling one)

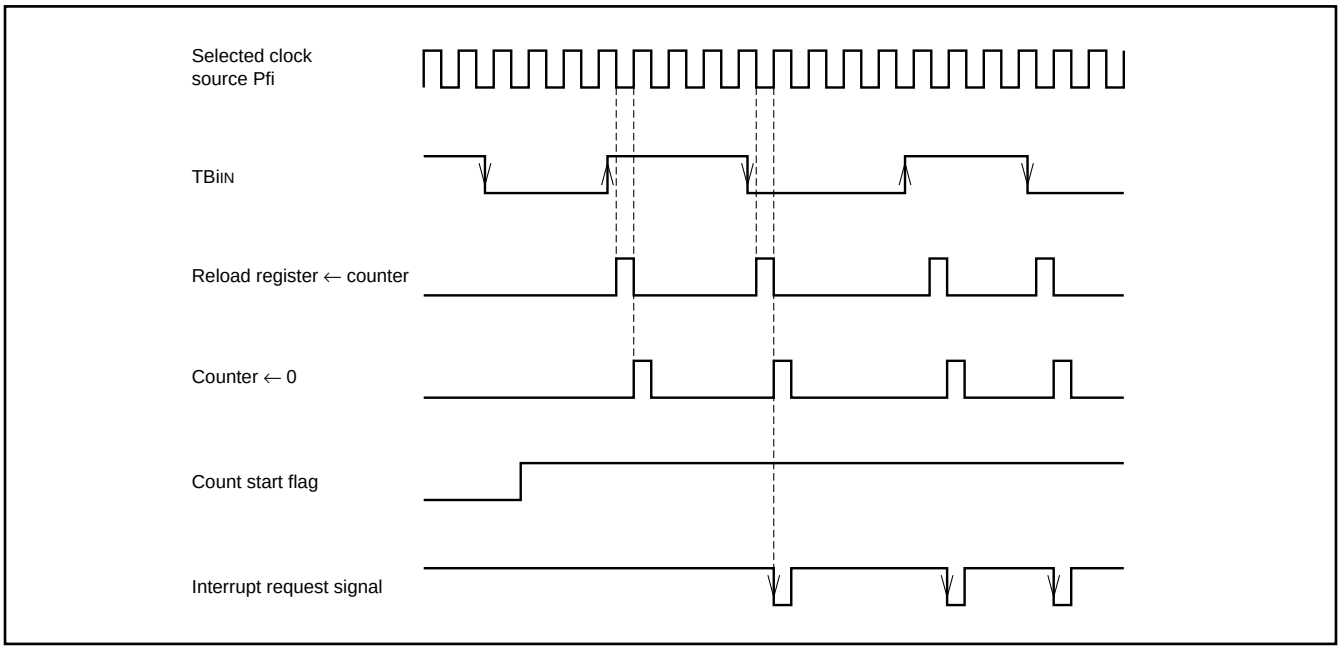


Fig. 40 Pulse width measurement mode operation

Timer function for motor control

Three-phase motor drive waveform and pulse motor drive waveform can be output by using plural internal timers A and B. Those modes are explained bellow.

Three-phase motor drive waveform output mode (three-phase waveform mode)

Three-phase waveform mode using four timers of the timers A0, A1, A2 and B4 is selected by setting the waveform output select bits of the waveform output mode register (address $1A_{16}$, Figure 41) to "100_2".

There are two types of the three-phase waveform mode: three-phase mode 0 and three-phase mode 1. Bit 4 of the waveform output mode register selects either mode. In three-phase waveform mode, set the corresponding timer mode registers of timers A0, A1, and A2 to select the one-shot pulse mode with the rising edge of external trigger; set the timer mode register of timer B2 to select the timer mode.

Figure 43 shows the three-phase waveform mode block diagram. The three-phase waveform mode outputs six waveforms, positive waveforms (U, V, W phases) and negative waveforms ($\overline{U}$, $\overline{V}$, $\overline{W}$ phases), from the respective ports with "L" level active.

Timer A2 controls U and $\overline{U}$ phases; timer A1 does $\overline{V}$ and V phases and timer A0 does W and $\overline{W}$ phases. Timer B2 controls those one-shot pulses' period of timers A2, A1 and A0.

In the waveform output, a short circuit prevention time can be set to prevent "L" level of positive waveforms (U, V, W phases) from overlapping with "L" level of their negative waveforms ($\overline{U}$, $\overline{V}$, $\overline{W}$ phases). The short circuit prevention time can be set with three 8-bit dead-time timers, sharing one reload register. The dead-time timer operates as a one-shot timer. As its start trigger, both the rising and falling edges of timers A0 to A2's one-shot pulses or their falling edge. Bit 6 of the waveform output mode register selects it. When that is "0", both the rising and falling edges become the start trigger; when that is "1", the falling edge becomes it.

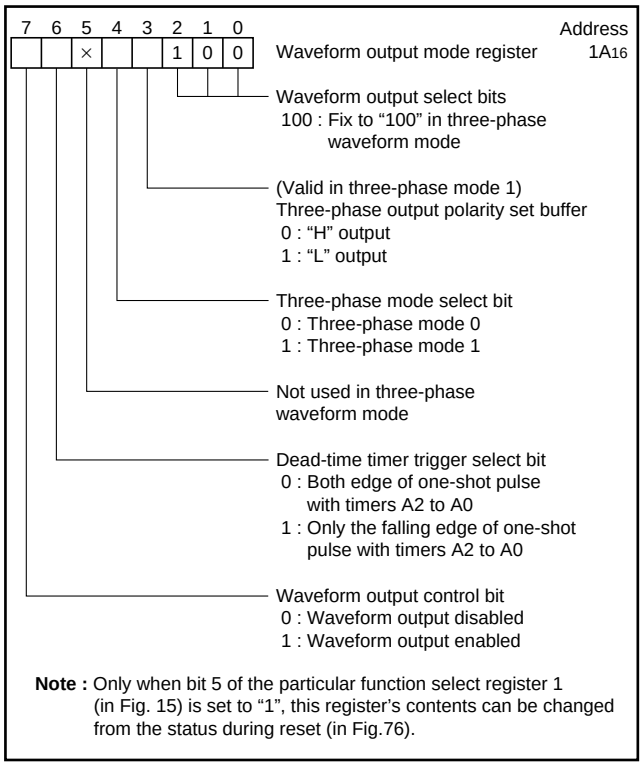


Fig. 41 Waveform output mode register bit configuration

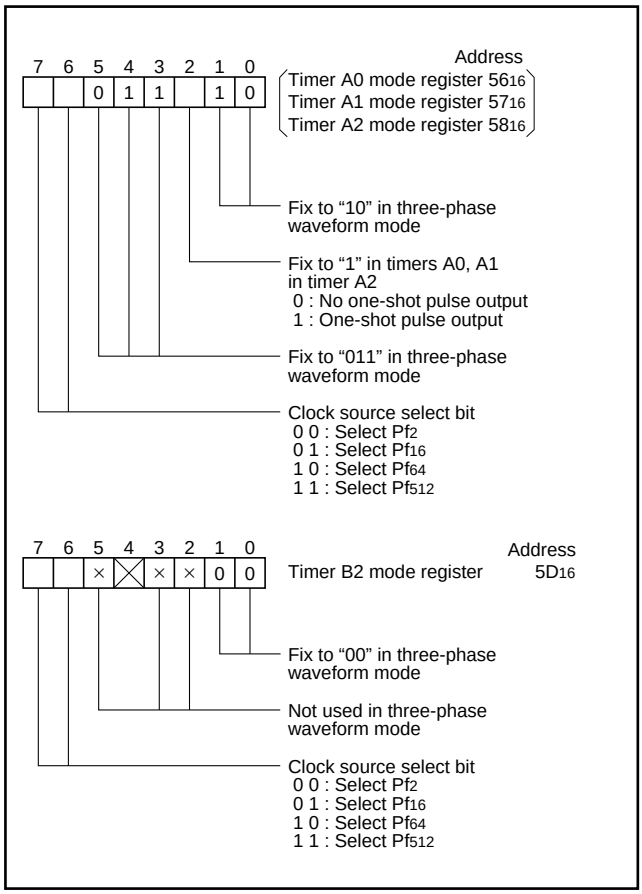


Fig. 42 Timer A0, A1, A2, mode register and timer B2 mode register bit configuration

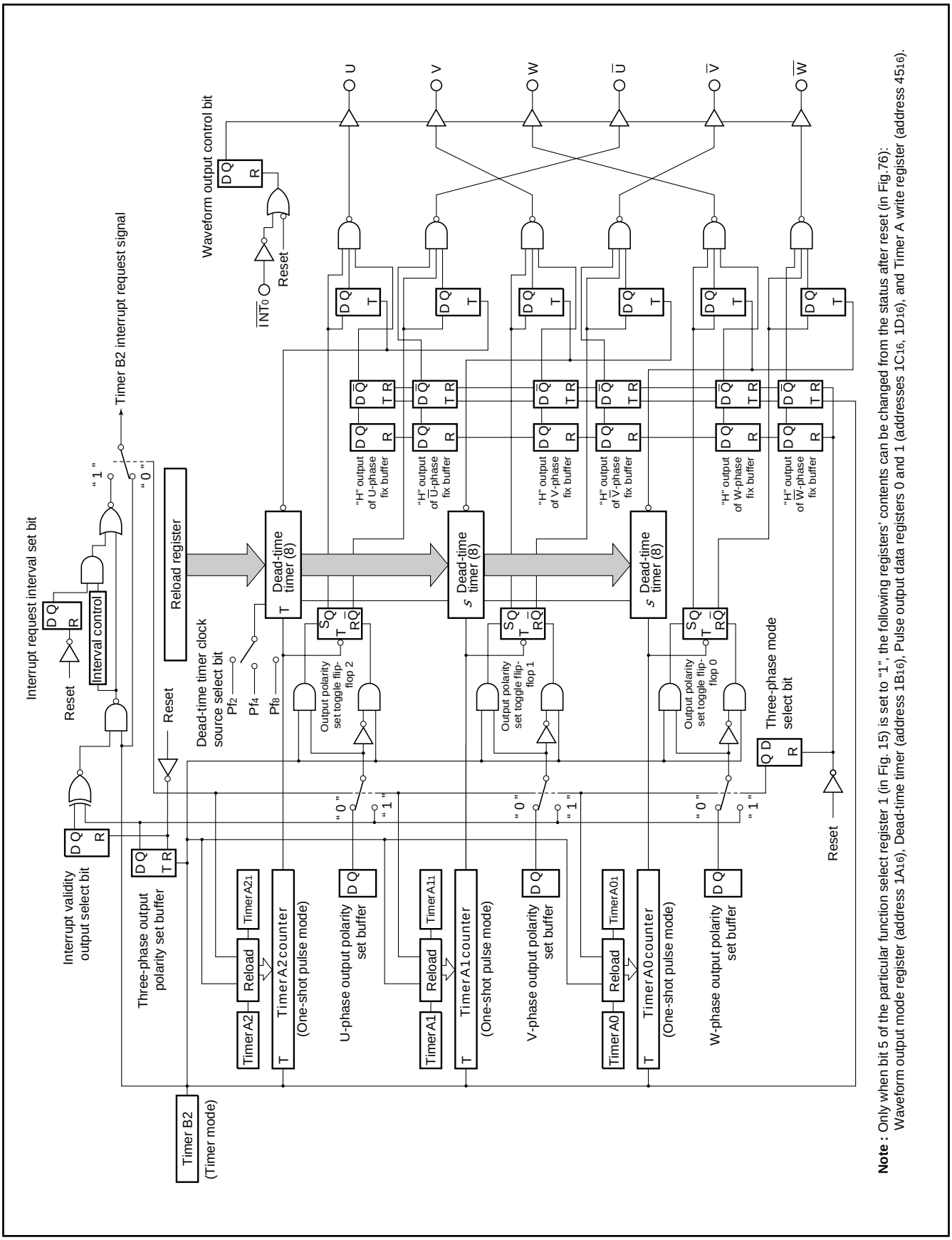


Fig. 43 Three-phase waveform mode block diagram

When writing data to the dead-time timer (address $1B_{16}$), the data is written to the reload register shared by three dead-time timers. When the dead-time timers catch the start trigger from the respective timers, the reload register contents are transferred to its counter and the dead-time timer decrements with the clock source selected by bits 6 and 7 of pulse output data register (address $1C_{16}$). Additionally, this timer can accept another trigger before completion of the preceding trigger operation. In this case, after transferring the reload register contents to the dead-time timer at acceptance of the trigger, the value is decremented.

The dead-time timer operates as a one-shot timer. Accordingly, this timer starts pulse output when the trigger is caught, and finishes pulse output and stops operation when its contents become "00_{16}", and waits next trigger.

In the three-phase waveform mode, setting bit 7 of the waveform output mode register (address $1A_{16}$) to "1" makes positive waveforms (U, V, W phases) and their negative waveforms ($\overline{U}$, $\overline{V}$, $\overline{W}$ phases) output from the respective ports. When that bit is "0", their ports are floating. That bit is cleared to "0" by inputting falling edge to the $\overline{INT_0}$ pin or reset other than clearing by an instruction..

Additionally, setting bits 5 to 3 of the pulse output data register 1 (address $1C_{16}$) to "1" makes the corresponding positive waveforms fixed to "H", and setting bits 7 to 5 of the pulse output data register 0 (address $1D_{16}$) to "1" makes the corresponding negative waveforms fixed to "H".

When selecting the three-phase waveform mode, $\overline{INT_0}$ pin become input-only pin.

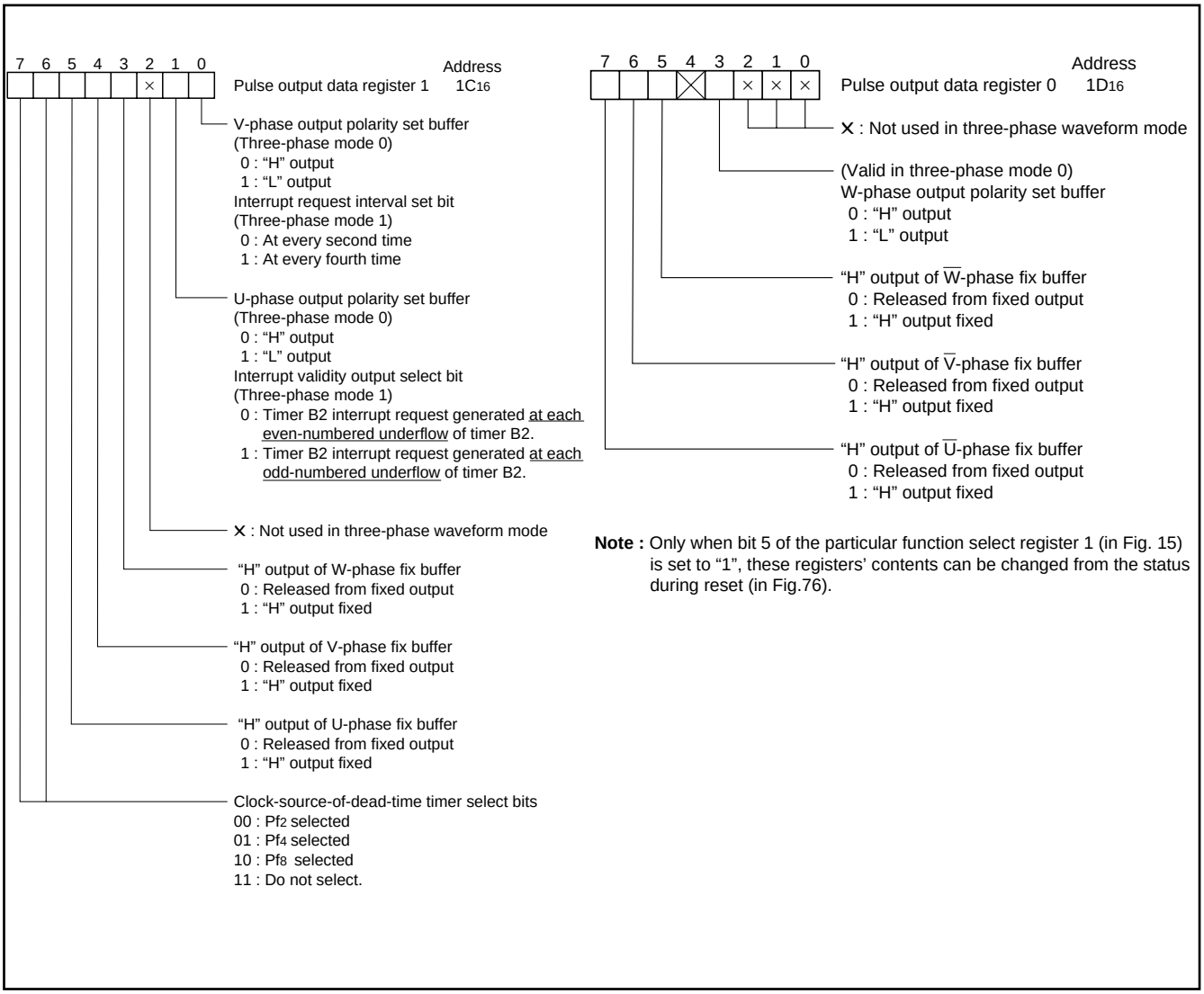


Fig. 44 Bit configuration of pulse output data registers 1 and 0 in three-phase waveform mode

Three-phase mode 0

In selecting three-phase waveform mode, three-phase mode 0 is selected by setting bit 4 of the waveform output mode register (address $1A_{16}$) to "0".

The output polarity of three-phase waveform depends on the output polarity set toggle flip-flop. The positive waveform of the three-phase waveform is "H" output when the toggle flip-flop is "0"; it is "L" output when the toggle flip-flop is "1". (Three-phase waveform is output as a negative waveform.)

Each output polarity set toggle flip-flop has the output polarity set buffer shown in Figure 44. When the timer B2's counter contents become 0000_{16}, the contents of output polarity set buffer are set into the output polarity set toggle flip-flop. After that, the polarity of the contents of output polarity set toggle flip-flop are reversed each time completion of one-shot pulse of timer (timers A2 to A0) corresponding to each phase.

Figure 45 shows an example of U-phase waveform and the output operation is explained. Three-phase mode 0 becomes valid when writing "0" to the U-phase output polarity set buffer (bit 1 at address $1C_{16}$) and actuating the timer B2. When the counter of timer B2 becomes 0000_{16}, the timer B2 interrupt request signal occurs and the timer A2 simultaneously starts one-shot pulse output. At this time, the contents of U-phase output polarity set buffer, "0" in this case, are set into the output polarity set toggle flip-flop 2.

When the one-shot pulse output of timer A2 is completed, the contents of output polarity set toggle flip-flop 2 is reversed from "0" to "1". Simultaneously, the one-shot pulse of the 8-bit dead-time timer is output for ensuring time not to overlap "L" levels of U phase waveform and its negative $\overline{U}$ phase waveform.

The U-phase waveform output keeps "H" level from the start until the one-shot pulse output of the dead-time timer is completed, even if the contents of output polarity set toggle flip-flop 2 are reversed from "0" to "1" owing to the timer A2's one-shot pulse output. When the one-shot pulse output of the dead-time timer is completed, "1" of output polarity set toggle flip-flop 2 which has been reversed becomes valid and the U phase waveform changes to "L" level.

Then, write "1" to the U-phase output polarity set buffer (bit 1 at address $1C_{16}$) before the counter of timer B2 becomes 0000_{16}.

After that, when the counter of timer B2 becomes 0000_{16}, the timer A2 starts one-shot pulse output. Simultaneously, the contents of U-phase output polarity set buffer, "1" in this case, are set into the output polarity set toggle flip-flop 2 and the U phase waveform remains "L" level.

When the one-shot pulse output of timer A2 is completed, the contents of output polarity set toggle flip-flop 2 is reversed from "1" to "0". Simultaneously, the one-shot pulse output of the dead-time timer starts.

When the contents of output polarity set toggle flip-flop 2 are reversed from "1" to "0", the U-phase waveform changes its output level from "L" to "H" without waiting for completion of the one-shot pulse output of the dead-time timer.

U-phase waveform is generated by repeating the operation above. The way to generate $\overline{U}$-phase waveform, which is the negative phase of U-phase, is the same as that for U-phase waveform except that the contents of output polarity set toggle flip-flop 2 are treated as the reversed signal from the case of U-phase waveform.

In this way, U-phase waveform and $\overline{U}$-phase waveform, having the negative phase of U-phase, are output from the pins so that their "L" levels do not overlap each other. The width of "L" level can be also modified by changing the value of timer B2 or timer A2.

V-, W-phase waveform and $\overline{V}$-, $\overline{W}$-phase waveform, having their negative phase, are similarly output according to the corresponding timer operation.

The explanation above is an example of three-phase waveform generating due to an triangular wave modulation. Three-phase waveform due to a saw-tooth-wave modulation can also be generated by fixing each beginning level of phases.

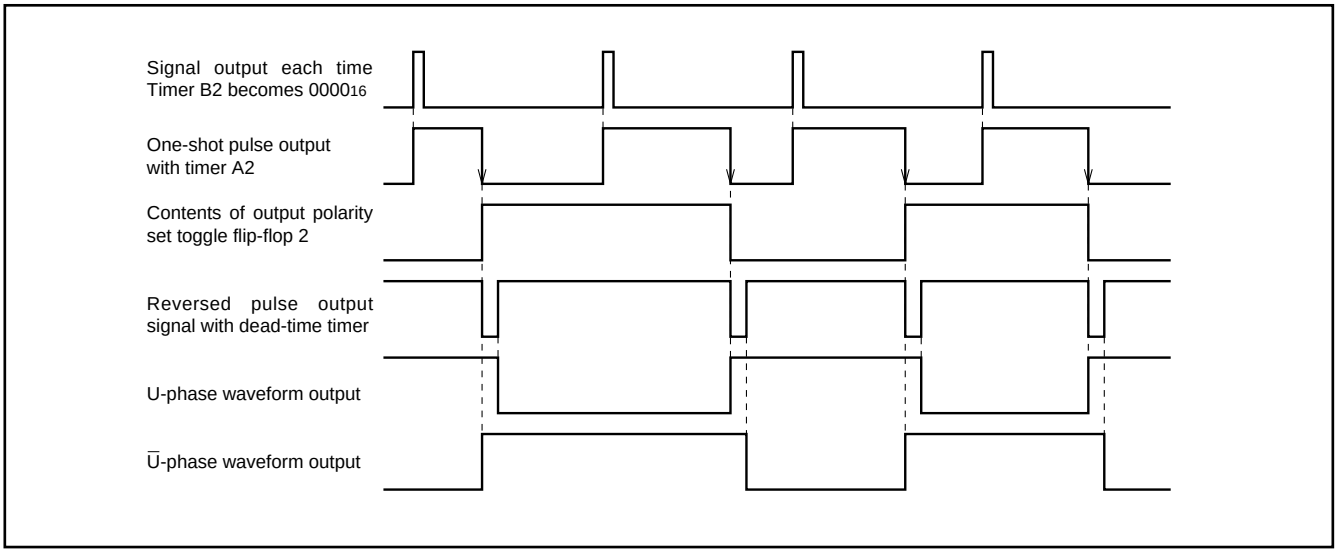


Fig. 45 U-phase waveform output example in three-phase mode 0 (triangular wave modulation)

Three-phase mode 1

In selecting three-phase waveform mode, three-phase mode 1 is selected by setting bit 4 of the waveform output mode register (address $1A_{16}$) to "1".

In this mode, each of timers A0 to A2 can have two timer registers and the contents of those registers are alternately reloaded into the counter each time the counter of timer B2 becomes 0000_{16}. About write operation to two timer registers, when rewriting to each timer register of timers A0, A1 and A2 after writing to each timer register of them, the data is written each to timers $A0_1$, $A1_1$ and $A2_1$. When writing to each timer register, the timer A write register (in Figure 46) indicates the timer to be intended for write.

The interrupt request normally occurs when the counter of timer B2 becomes 0000_{16}. However, this occurrence interval can be switched between "every second time" and "every fourth time." Bit 0 of the pulse output data register 1 (address $1C_{16}$) selects that.

Additionally, "0" or "1" of the three-phase output polarity set buffer can be used as the occurrence factor of timer B2 interrupt request. Bit 1 of the pulse output data register 1 (address $1C_{16}$) selects that. When the timer B2's counter contents become 0000_{16}, the contents of three-phase output polarity set buffer are set into the output polarity set toggle flip-flop on which .the output polarity of three-phase waveform depends. The contents of three-phase output polarity set buffer are reversed after that operation.

The polarity of the contents of output polarity set toggle flip-flop is reversed each time completion of one-shot pulse of timer (timers A2 to A0) corresponding to each phase.

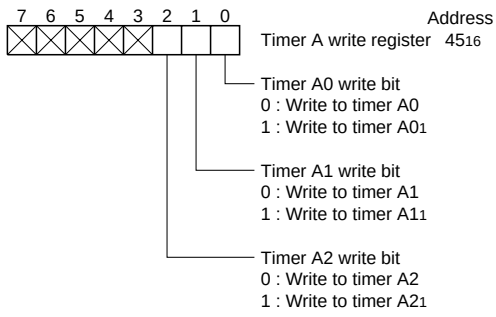
Figure 47 shows an example of U-phase waveform and the output operation is explained.

Write "0" to the three-phase output polarity set buffer (bit 3 at address $1A_{16}$). Clear the interrupt request interval set bit (bit 0 at address $1C_{16}$) to "0" so that the timer B2 interrupt request may occur at every second time. Additionally, clear the interrupt validity output select bit (bit 1 at address $1C_{16}$) so that the timer B2 interrupt request may occur at "0" of the three-phase output polarity set buffer.

After the procedure above, three-phase mode 1 starts operation when actuating the timer B2.

When the counter of timer B2 becomes 0000_{16}, the timer B2 interrupt request occurs and timer A2 simultaneously starts one-shot pulse output. At this time, the contents of three-phase output polarity set buffer, "0" in this case, are set into the output polarity set toggle flip-flop 2. The contents of three-phase output polarity set buffer are reversed from "0" to "1" after that operation.

When the timer A2 counter counts the value written into the timer A2 and the one-shot pulse output of timer A2 is completed, the contents of output polarity set toggle flip-flop 2 are reversed from "0" to "1". Simultaneously, the one-shot pulse of the 8-bit dead-time timer is output for ensuring time, so that "L" levels of U- and $\overline{U}$-phase waveforms do not overlap.



Note : Only when bit 5 of the particular function select register 1 (in Fig. 15) is set to "1", this register's contents can be changed from the status after reset (in Fig.76).

Fig. 46 Timer A write flag bit configuration

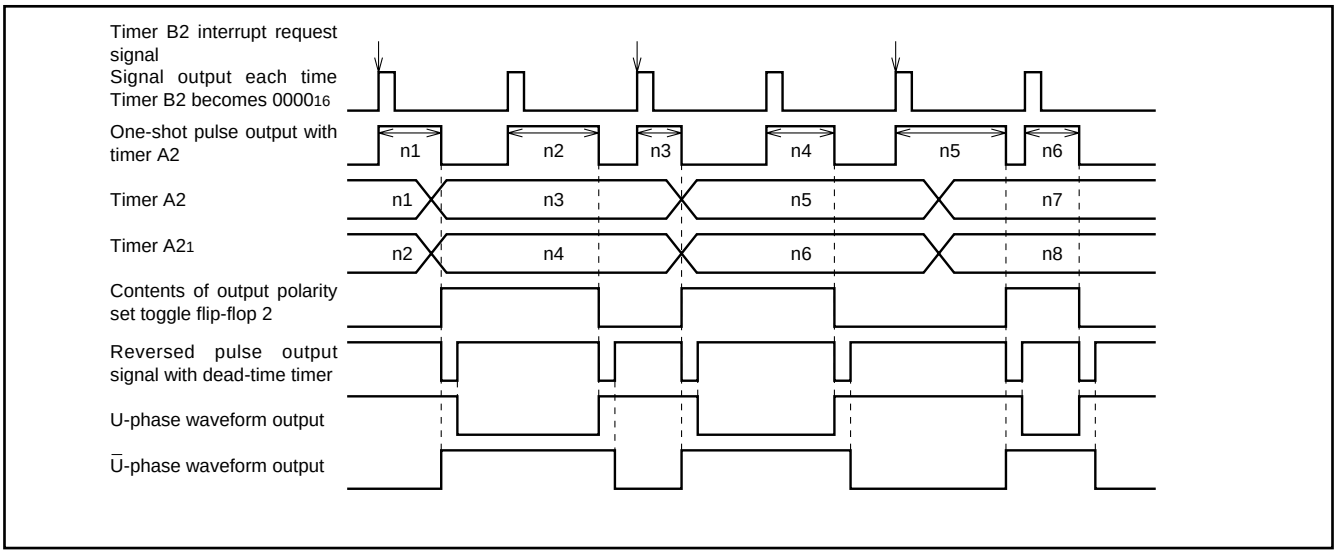


Fig. 47 U-phase waveform output example in three-phase mode 1 (triangular wave modulation)

The U-phase waveform output keeps “H” level from the start until the one-shot pulse output of the dead-time timer is completed, even if the contents of output polarity set toggle flip-flop 2 are reversed from “0” to “1” owing to the timer A2’s one-shot pulse output.
When the one-shot pulse output of the dead-time timer is completed, “1” of output polarity set toggle flip-flop 2 which has been reversed becomes valid and the U-phase waveform changes to “L” level.
Then, when the counter of timer B2 becomes 0000_{16}, the timer A2 counter counts the value written into timer A2 and timer A2 starts one-shot pulse output. Simultaneously, the contents of three-phase output polarity set buffer are set into the output polarity set toggle flip-flop 2. However, the U-phase waveform remains “L” level, because the value is the same (“1”).
The contents of three-phase output polarity set buffer are reversed from “1” to “0” after that operation.
When the one-shot pulse output of timer A2 is completed, the contents of output polarity set toggle flip-flop 2 is reversed from “1” to “0”. Simultaneously, the one-shot pulse output of the dead-time timer starts.
When the contents of output polarity set toggle flip-flop 2 is reversed from “1” to “0”, the U-phase waveform changes its output level from “L” to “H” without waiting for completion of the one-shot pulse output of the dead-time timer.
U-phase waveform is generated by repeating the operation above.
The way to generate $\overline{U}$-phase waveform, which is the negative phase of U-phase, is the same as that for U-phase waveform except that the contents of output polarity set toggle flip-flop 2 is treated as the reversed signal from the case of U-phase waveform.
In this way, U-phase waveform and $\overline{U}$-phase waveform, having the negative phase of U-phase, are output from the pins so that their “L” levels do not overlap each other. The width of “L” level can be also modified by changing the value of timer B2, timer A2 or timer $A2_1$.
V-, W-phase waveform and $\overline{V}$-, $\overline{W}$-phase waveform, having their negative phase, are similarly output according to the corresponding timer operation.

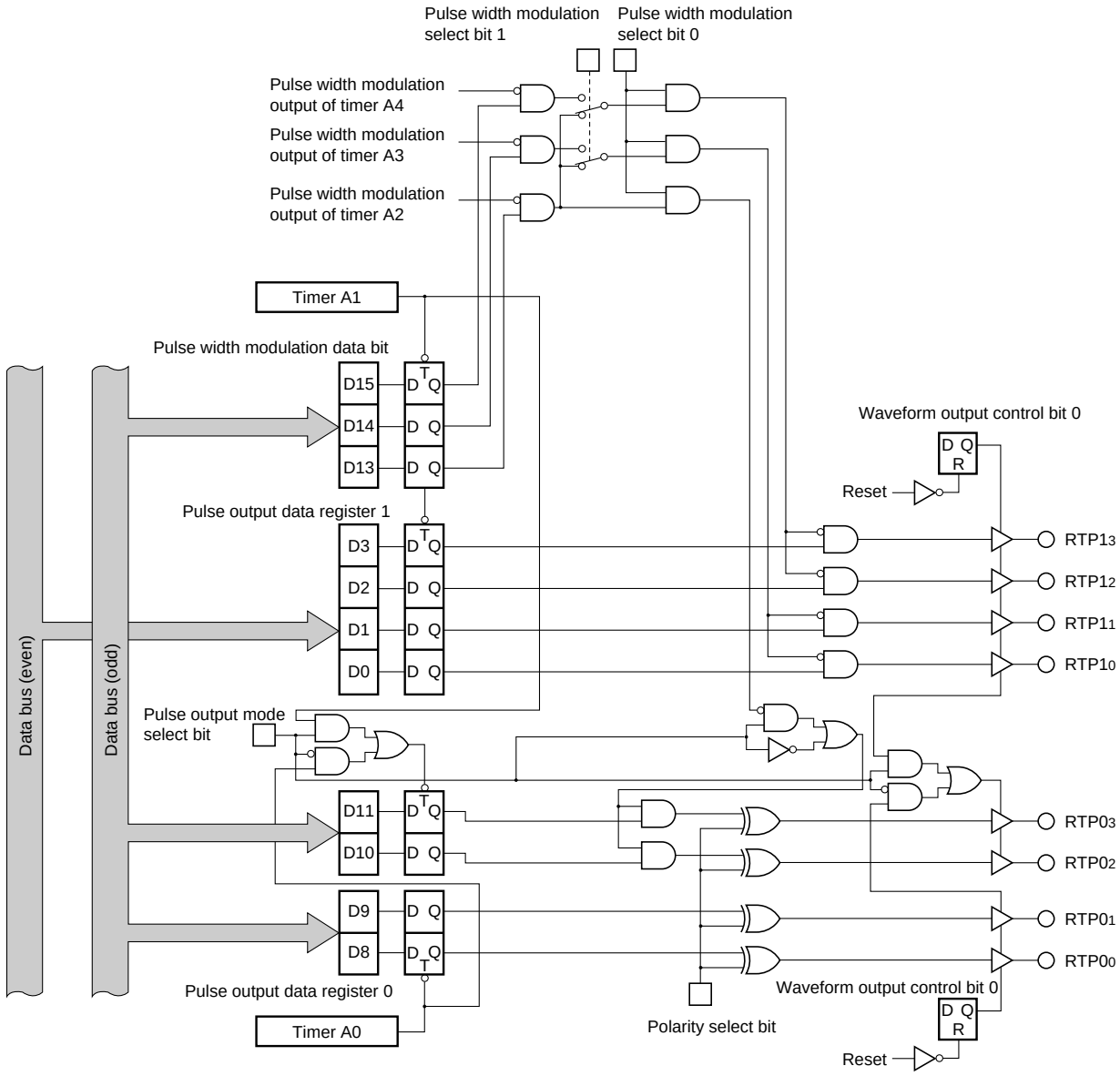
Pulse output port mode

Figure 48 shows the pulse output port mode block diagram.

This mode has an 8-bit pulse output port. The waveform output select bits (bits 0 to 2) of waveform output mode register (address $1A_{16}$, Figure 49) select use of pulse output port. The 8-bit pulse output port is divided into 4 bits and 4 bits, or 6 bits and 2 bits with the pulse output mode select bit (bit 4) of pulse output data register 1 (address $1C_{16}$, Figure 51) ; each of them can be individually controlled.

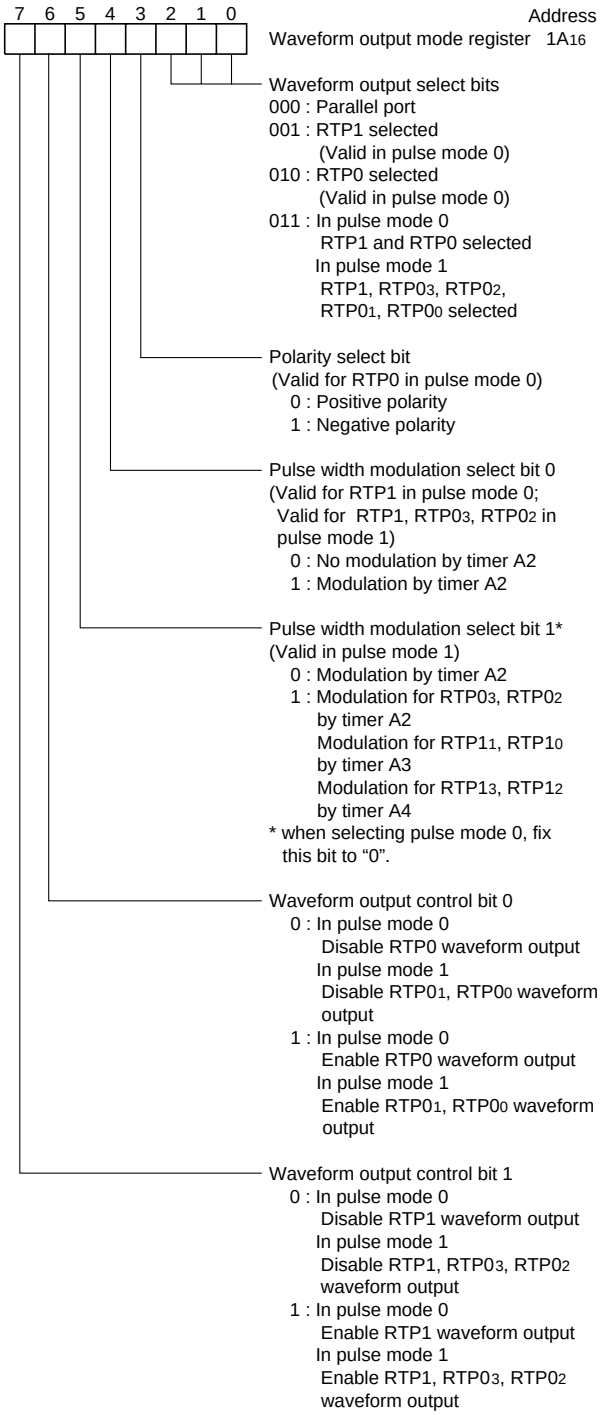
Set timers A1 and A0 to the timer mode because they are used in the pulse output mode. Additionally, set bit 2 of the corresponding timer Ai mode register to "1" to use a pulse output port because the pulse output port are multiplexed with the TAi_{OUT} (i = 0 to 4). Figure 50 shows the bit configuration of timer A1 and A0 mode registers in the pulse output port mode.

Timers A1 and A0 start count when setting the corresponding timer count start flag to "1", and they stop it when clearing that flag to "0".



Note : Only when bit 5 of the particular function select register 1 (in Fig. 15) is set to "1", the following registers' contents can be changed from the status after reset (in Fig.76) : Waveform output mode register (address $1A_{16}$) and Pulse output data registers 0 and 1 (addresses $1C_{16}$, $1D_{16}$).

Fig. 48 Pulse output port mode block diagram



Note : Only when bit 5 of the particular function select register 1 (in Fig. 15) is set to "1", this register's contents can be changed from the status after reset (in Fig.76).

Fig. 49 Bit configuration of waveform output mode register in pulse output port mode

Pulse mode 0

This mode divides a pulse output port into 4 bits and 4 bits and individually controls them.

When setting the pulse output mode select bit to "0", and setting bits 2 and 1 to "0" and bit 0 to "1" of the waveform output select bits, four of RTP13, RTP12, RTP11, and RTP10 become the pulse output ports with RTP1 selected.

When setting the pulse output mode select bit to "0", and setting bits 2 and 0 to "0" and bit 1 to "1" of the waveform output select bits, four of RTP03, RTP02, RTP01, RTP00 become the pulse output ports with RTP0 selected.

When setting the pulse output mode select bit to "0", and setting bit 2 to "0" and bits 1 and 0 to "1" of the waveform output select bits, the following two groups become the pulse output ports with RTP1 and RTP0 selected:

•Four of RTP13, RTP12, RTP11, RTP10

•Four of RTP03, RTP02, RTP01, RTP00.

Each time the contents of timer A1 counter become 0000$_{16}$, the contents of pulse output data register 1 (low-order 4 bits at address 1C$_{16}$) corresponding to RTP13, RTP12, RTP11, RTP10 are output from ports.

Each time the contents of timer A0 counter become 0000$_{16}$, the contents of pulse output data register 0 (low-order 4 bits at address 1D$_{16}$) corresponding to RTP03, RTP02, RTP01, RTP00 are output from ports.

When writing "0" to the specified bit of pulse output data register, "L" level is output from the pulse output port when the contents of corresponding timer counter become 0000$_{16}$; when writing "1" to it, "H" level is output from the pulse output port.

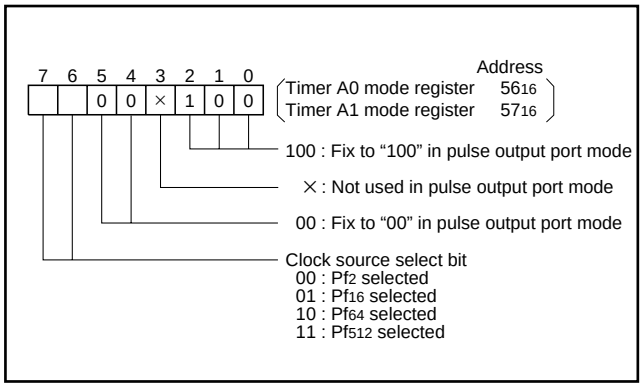


Fig. 50 Bit configuration of timer A1 and A0 mode registers in pulse output port mode

Additionally, pulse width modulation can be applied for the pulse output port RTP1. Because the timer A2 is used for pulse width modulation, actuate timer A2 in the pulse width modulation mode. When any bit of pulse output data is "1", the pulse to which pulse width modulation is applied is output from the pulse output port when the contents of timer A1 counter become 0000_{16}.

Pulse width modulation by timer A2 is applied when setting the pulse width modulation select bit 0 (bit 4) of waveform output mode register to "1", pulse width modulation select bit 1 (bit 5) to "0," and the pulse width modulation data bit of RTP1 (bit 5) of pulse output data register 0 to "1".

$RTP0_3$, $RTP0_2$, $RTP0_1$ and $RTP0_0$ can output the contents of pulse output data register 0 by setting the polarity select bit (bit 3) of waveform output mode register. When the polarity select bit is "1", the reversed contents of pulse output data register 0 is output; when that bit is "0", the contents of pulse output data register 0 are output as it is. Figure 52 shows example waveforms in the pulse mode 0.

In ports selecting the pulse mode 0, output of $RTP0_3$, $RTP0_2$, $RTP0_1$ and $RTP0_0$ is controlled by the waveform output control bit 0 (bit 6) of waveform output mode register; output of $RTP1_3$, $RTP1_2$, $RTP1_1$ and $RTP1_0$ is done by the waveform output control bit 1 (bit 7).

When setting the waveform output control bit to "1", waveform is output from the corresponding port. When clearing that bit to "0", waveform output from the corresponding port stops, and the port becomes floating. The waveform output control bits are cleared to "0" by reset other than clearing with instructions.

Pulse mode 1

This mode divides a pulse output port into 6 bits and 2 bits, and individually controls them.

When setting the pulse output mode select bit to "1", and setting bit 2 to "0" and bits 1 and 0 to "1" of the waveform output select bits, the following two groups become the pulse output ports:

•Six of $RTP1_3$, $RTP1_2$, $RTP1_1$, $RTP1_0$, $RTP0_3$, $RTP0_2$

•Two of $RTP0_1$, $RTP0_0$.

Timer A1 controls six of $RTP1_3$, $RTP1_2$, $RTP1_1$, $RTP1_0$, $RTP0_3$, and $RTP0_2$; timer A0 controls two of $RTP0_1$, $RTP0_0$.

Additionally, pulse width modulation can be applied for the pulse output ports (RTP1, $RTP0_3$, $RTP0_2$). The pulse width modulation select bit 1 (bit 5) of waveform output mode register selects the type of modulation: the common modulation to six of $RTP1_3$, $RTP1_2$, $RTP1_1$, $RTP1_0$, $RTP0_3$ and $RTP0_2$ or the modulation to every two ports of $RTP1_3$ and $RTP1_2$, $RTP1_1$ and $RTP1_0$, $RTP0_3$ and $RTP0_2$.

When setting that bit to "0", the common modulation to six ports is applied; when setting that bit to "1", the modulation to every two ports is applied. The timer A2 is used for the common modulation to six ports; the timers A2, A3 and A4 are used for the modulation to every two ports. Accordingly, actuate the respective timers in the pulse width modulation mode. When any bit of pulse output data is "1", the pulse to which pulse width modulation is applied is output from the pulse output port when the contents of timer A1 counter become 0000_{16}.

Pulse width modulation by corresponding timers is applied when setting the pulse width modulation select bit 0 of waveform output mode register to "1" and the corresponding pulse width modulation data bits (bits 7 to 5) of pulse output data register 0 to "1".

The polarity select bit (bit 3) of waveform output mode register must be "0" to select the positive polarity. The other operations are the same as that of pulse mode 0. Figure 53 shows example waveforms in the pulse mode 1.

In ports selecting the pulse mode 1, output of $RTP0_1$ and $RTP0_0$ is controlled by the waveform output control bit 0 (bit 6) of waveform output mode register; output of $RTP1_3$, $RTP1_2$, $RTP1_1$, $RTP1_0$, $RTP0_3$ and $RTP0_2$ is done by the waveform output control bit 1 (bit 7).

When setting the waveform output control bit to "1", waveform is output from the corresponding port. When clearing that bit to "0", waveform output from the corresponding port stops and the port becomes floating. The waveform output control bits are cleared to "0" by reset other than clearing with instructions.

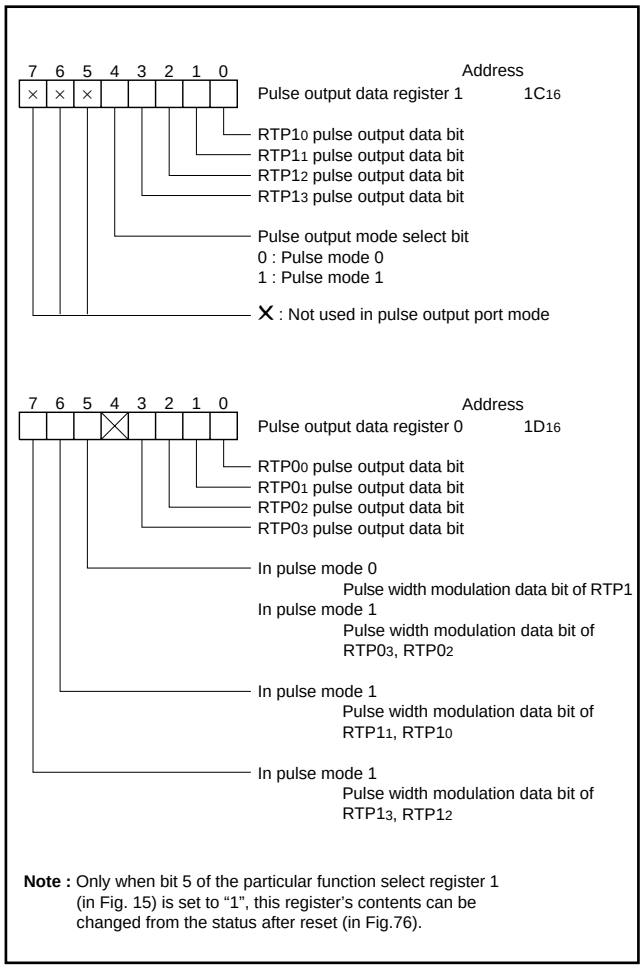


Fig. 51 Bit configuration of pulse output data registers 1 and 0 in pulse output port mode

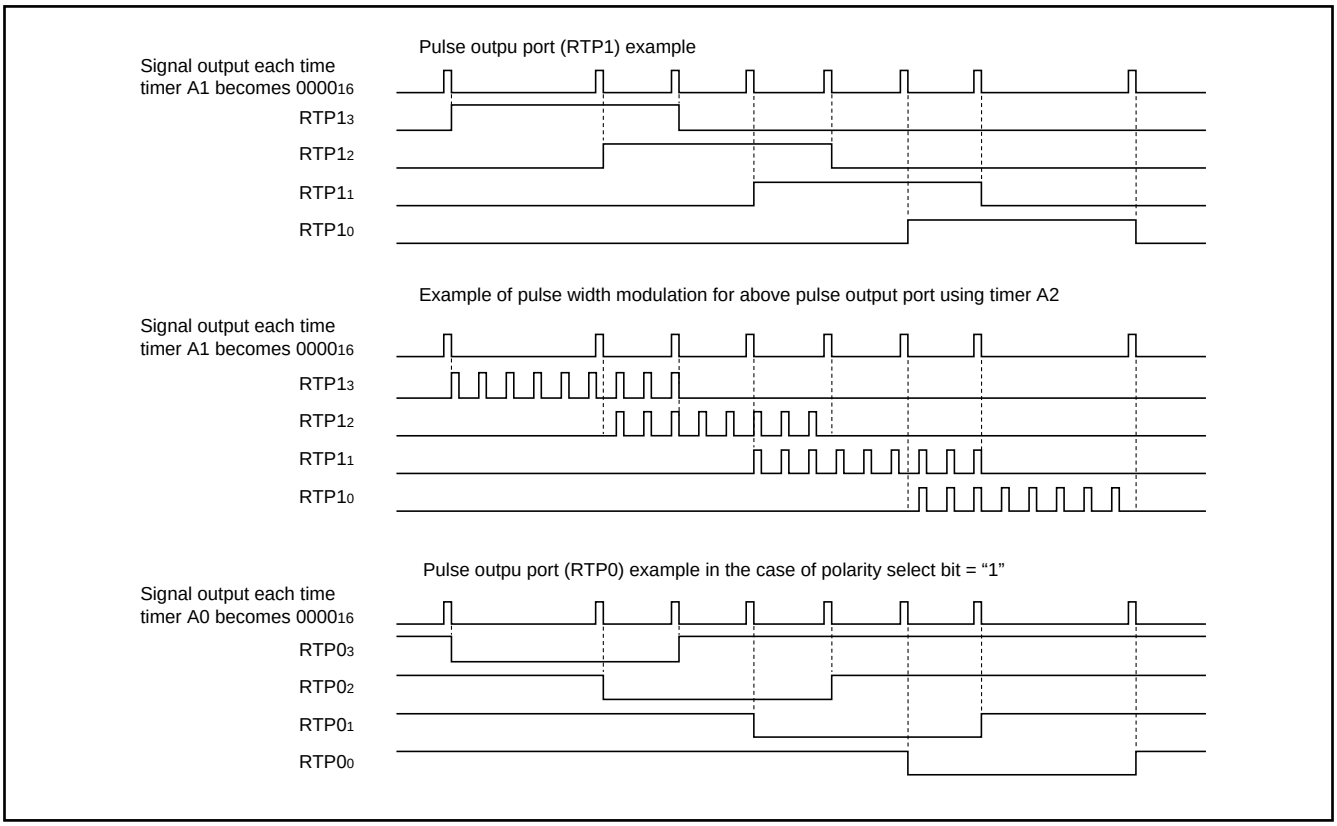


Fig. 52 Example waveforms in pulse mode 0

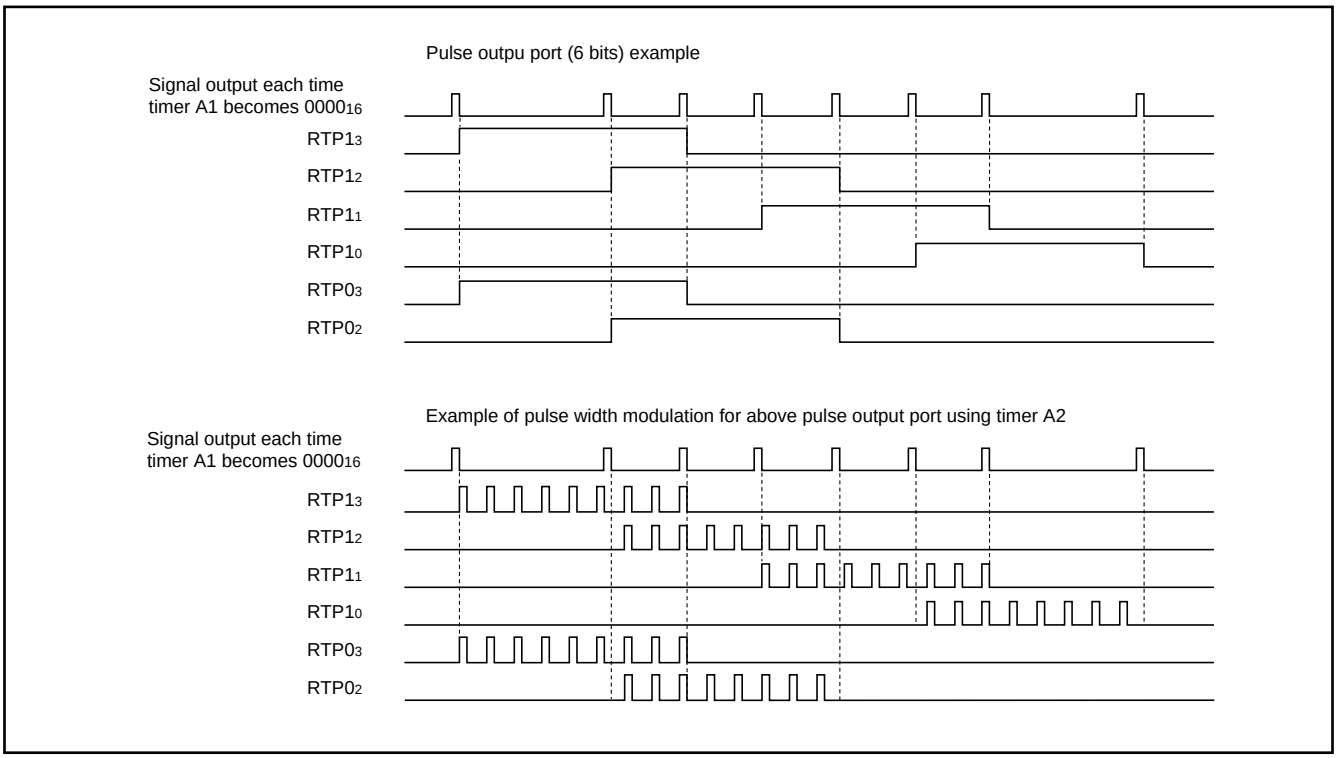


Fig. 53 Example waveforms in pulse mode 1

SERIAL I/O PORTS

Two independent serial I/O ports are provided. Figure 54 shows a block diagram of the serial I/O ports.

Bits 0, 1, and 2 of the UARTi(i = 0,1) Transmit/Receive mode register shown in Figure 55 are used to determine whether to use port P8 as parallel port, clock synchronous serial I/O port, or asynchronous (UART) serial I/O port using start and stop bits.

Figures 56 and 57 show the connections of receiver/transmitter according to the mode.

Figure 58 shows the bit configuration of the UARTi Transmit/Receive control register.

Each communication method is described below.

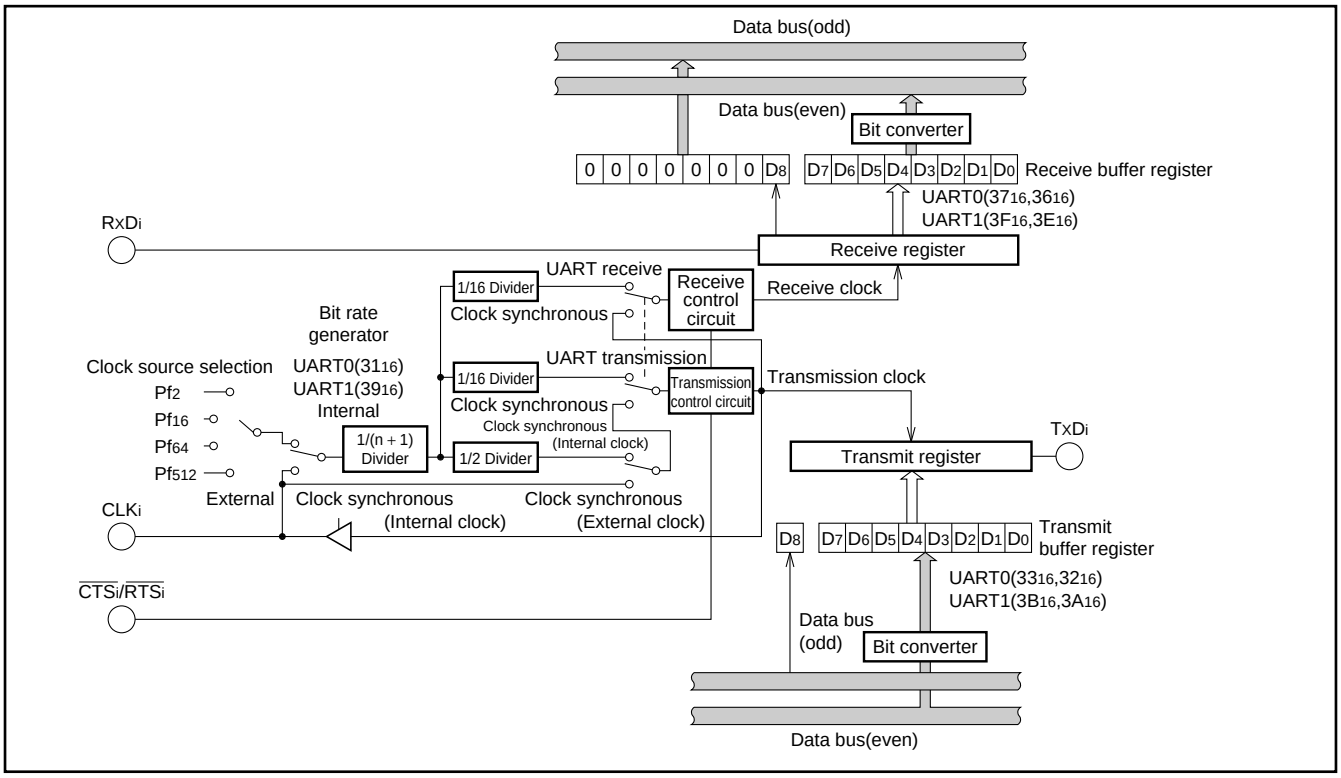


Fig. 54 Serial I/O port block diagram

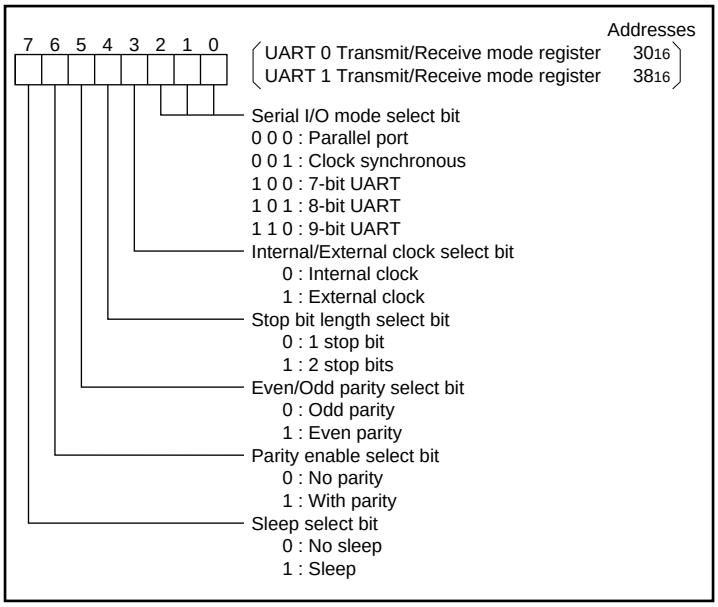


Fig. 55 UARTi Transmit/Receive mode register bit configuration

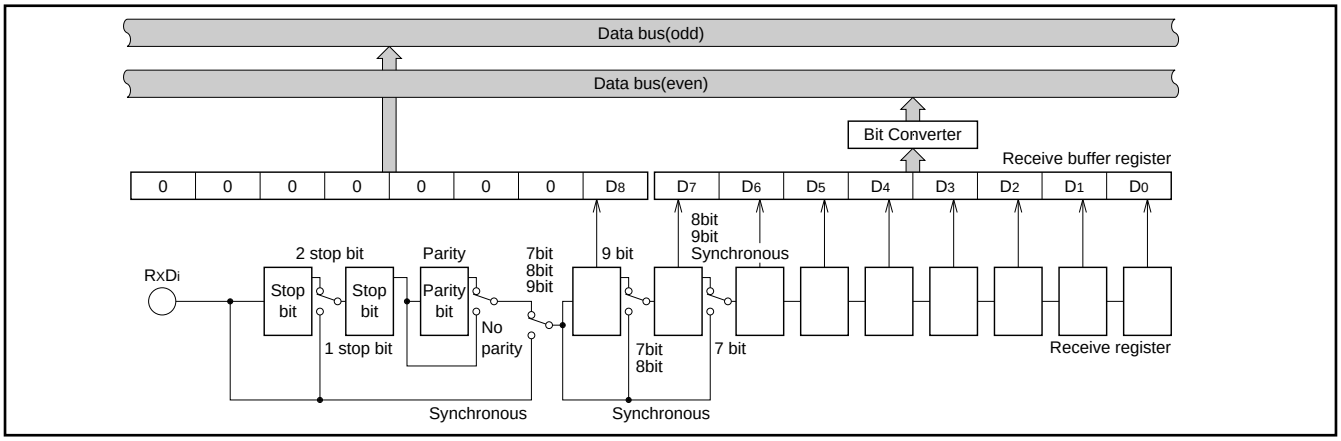


Fig. 56 Receiver block diagram

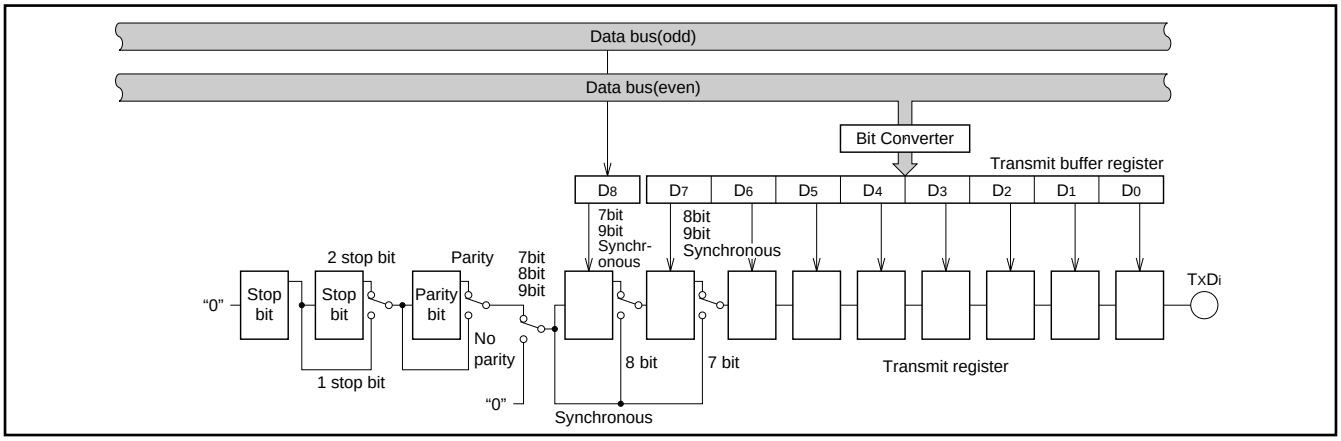


Fig. 57 Transmitter block diagram

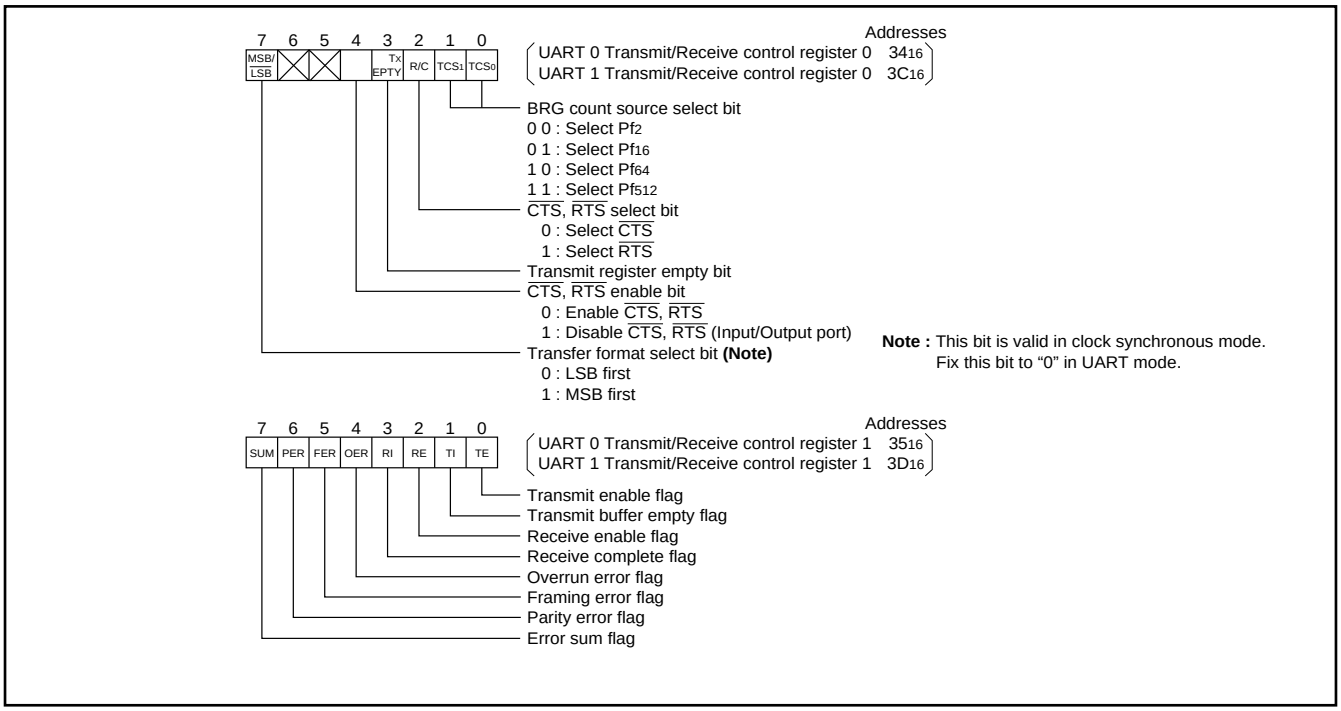


Fig. 58 UARTi Transmit/Receive control register bit configuration

CLOCK SYNCHRONOUS SERIAL COMMUNICATION

A case where communication is performed between two clock synchronous serial I/O ports as shown in Figure 59 will be described. (The transmission side will be denoted by subscript j and the receiving side will be denoted by subscript k.)

Bit 0 of the UARTj Transmit/Receive mode register and UARTk Transmit/Receive mode register must be set to "1" and bits 1 and 2 must be "0". The length of the transmission data is fixed at 8 bits.

Bit 3 of the UARTj Transmit/Receive mode register of the clock sending side is cleared to "0" to select the internal clock. Bit 3 of the UARTk Transmit/Receive mode register of the clock receiving side is set to "1" to select the external clock. Bits 4, 5 and 6 are ignored in clock synchronous mode. Bit 7 must always be "0".

The clock source is selected by bit 0 (TCS_0) and bit 1 (TCS_1) of the clock-sending-side UARTj Transmit/Receive control register 0. As shown in Figure 54, the selected clock is divided by (n+1), then by 2, is passed through a transmission control circuit, and is output as transmission clock CLKj. Therefore, when the selected clock is Pfi,

$$\text{Bit Rate} = Pf_i / \{(n+1) \times 2\}$$

On the clock receiving side, the TCS_0 and TCS_1 bits of the UARTk Transmit/Receive control register 0 are ignored because an external clock is selected.

Bit 2 of the clock-sending-side UARTj Transmit/Receive control register 0 is cleared to "0" to select $\overline{CTS}j$ input. Bit 2 of the clock receiving side is set to "1" to select $\overline{RTS}k$ output.

Bit 4 of the UART Transmit/Receive control register 0 is used to determine whether to use $\overline{CTS}$ or $\overline{RTS}$ signal. Bit 4 must be "0" when $\overline{CTS}$ or $\overline{RTS}$ signal is used. Bit 4 must be "1" when $\overline{CTS}$ and $\overline{RTS}$ signals are not used. When $\overline{CTS}$ and $\overline{RTS}$ signals are not used, $\overline{CTS}/\overline{RTS}$ pin can be used as a normal port. The case using $\overline{CTS}$ and $\overline{RTS}$ signals are explained below. However, when $\overline{CTS}$ and $\overline{RTS}$ signals are not used, there are no condition of $\overline{CTS}j$ input, and there is no $\overline{RTS}k$ output.

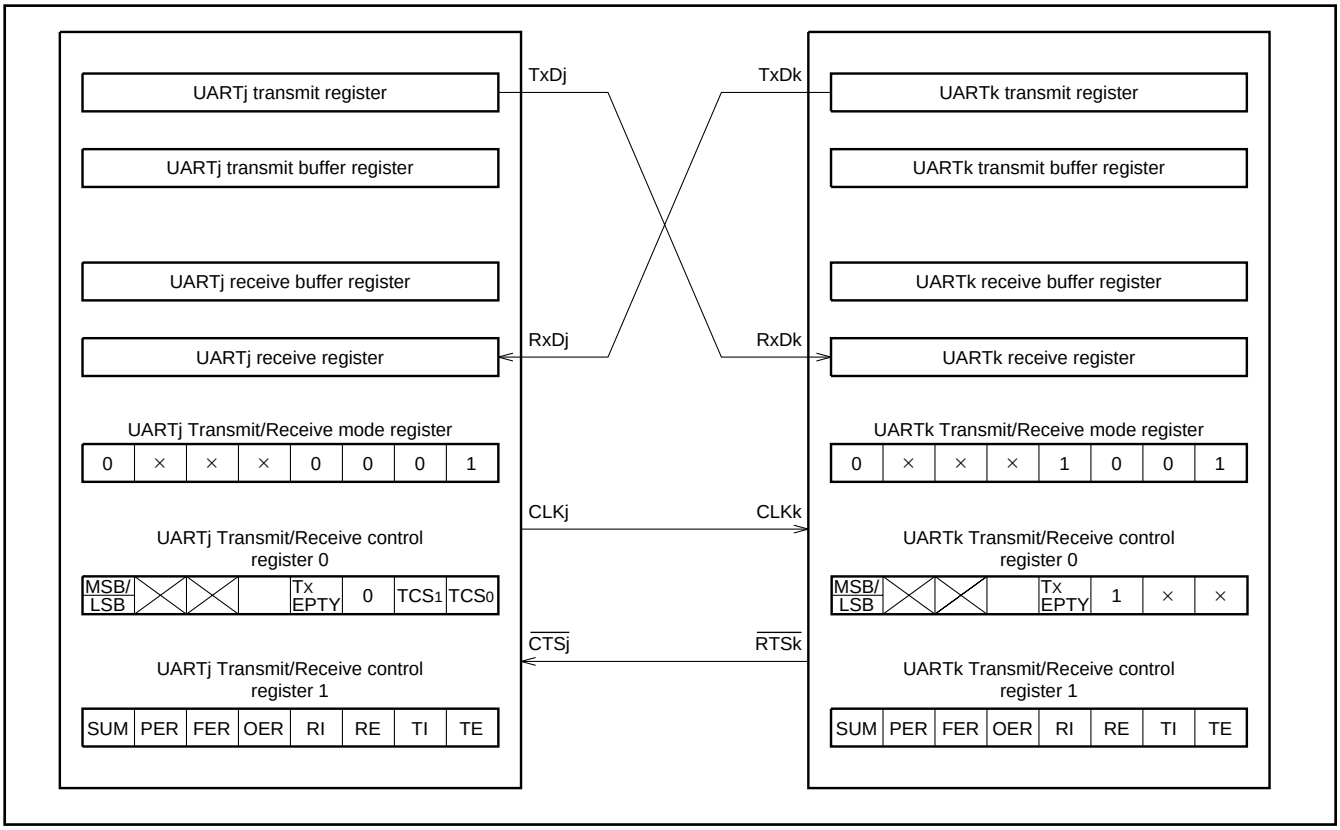


Fig. 59 Clock synchronous serial communication

Transmission

Transmission is started when bit 0 (TEj flag) of UARTj Transmit/Receive control register 1 is “1”, bit 1 (TIj flag) of one is “0”, and $\overline{CTS}j$ input is “L”. As shown in Figure 60, data is output from TxDj pin each time when transmission clock CLKj changes from “H” to “L”. The data is output from the least significant bit.

The TIj flag indicates whether the transmit buffer register is empty or not. It is cleared to “0” when data is written in the transmit buffer register and set to “1” when the contents of the transmit buffer register is transferred to the transmit register.

When the transmit register becomes empty after the contents has been transmitted, data is transferred automatically from the transmit buffer register to the transmit register if the next transmission start condition is satisfied. If bit 2 of UARTj Transmit/Receive control register 0 is “1”, $\overline{CTS}j$ input is ignored, and transmission start is controlled only by the TEj flag and TIj flag. Once transmission has started, the TEj flag, TIj flag, and $\overline{CTS}j$ signals are ignored until data transmission completes. Therefore, transmission is not interrupt when $\overline{CTS}j$ input is changed to “H” during transmission.

The transmission start condition indicated by TEj flag, TIj flag, and $\overline{CTS}j$ is checked while the TENDj signal (shown in Figure 60) is “H”. Therefore, data can be transmitted continuously if the next transmission data is written in the transmit buffer register and TIj flag is cleared to “0” before theTENDj signal goes “H”.

Bit 3 (TxEPTYj flag) of UARTj Transmit/Receive control register 0 changes to “1” at the next cycle just after the TENDj signal goes “H” and changes to “0” when transmission starts. Therefore, this flag can be used to determine whether data transmission has completed.

When the TIj flag changes from “0” to “1”, the interrupt request bit in the UARTj transmit interrupt control register is set to “1”.

In only UART0, data can be output to a maximum of 3 external receive devices. This is realized under the condition in which the internal clock is selected and the transmission clock is output from one of pins CLK0, CLKS0 (multiplexed with RxD0) and CLKS1 (multiplexed with $\overline{CTS}0$/$\overline{RTS}0$). Make sure that do not switch the selection of the clock during transmission. Figure 61 shows an external connection example.

Plural output of transmit clock mode is set with bits 1 and 0 of the particular function select register 1. Additionally, it is necessary to select the internal clock, disable $\overline{CTS}$ and $\overline{RTS}$, receive and D-A output with the UART0 Transmit/Receive mode register, UART0 Transmit/Receive control registers 0 and 1, and A-D control register 1. Figure 62 shows the other registers bit configuration in plural output of transmit clock mode and Figure 63 shows the particular function select register 1 bit configuration .

Table 6 shows the function of the particular function select register 1’s bits 1 and 0, which is the output pin of transmit clock select bits: TC1 and TC0. According to this table, select the CLK0, CLKS0 or CLKS1 pin corresponding to the contents of TC1 and TC0, and output the transmit clock.

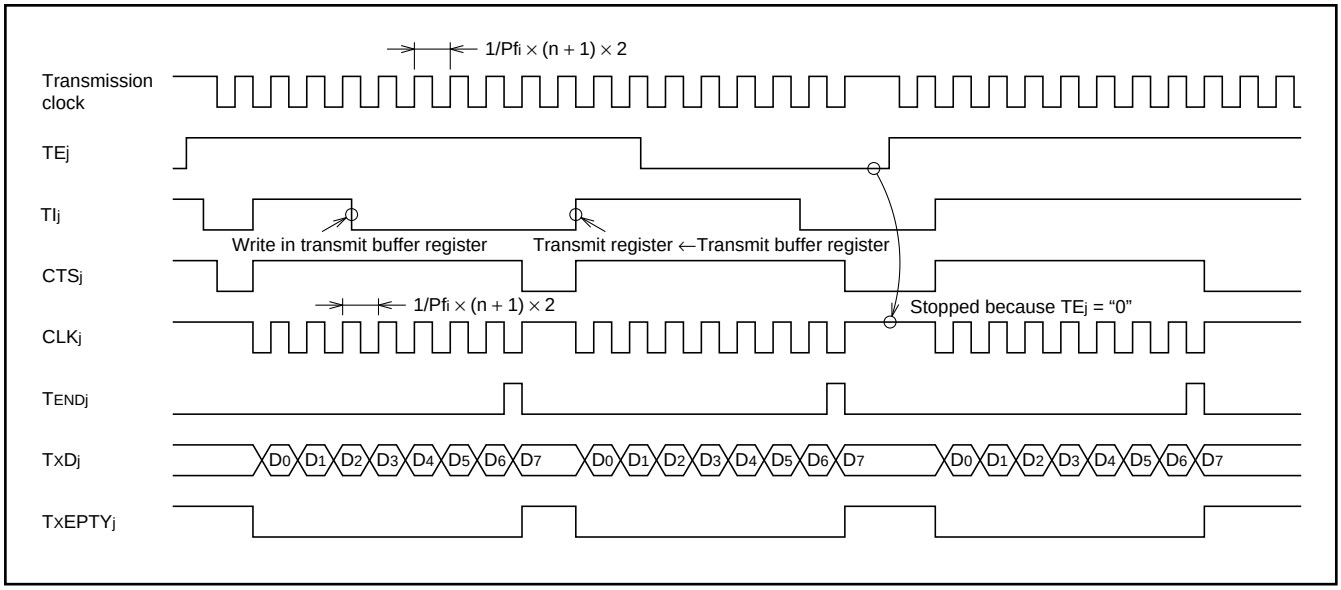
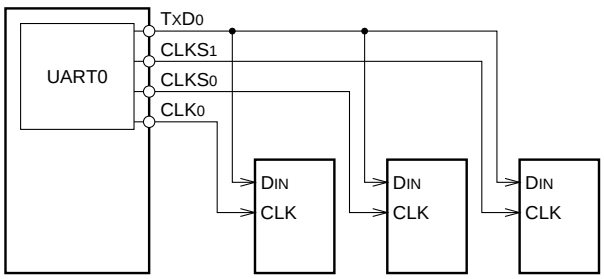


Fig. 60 Clock synchronous serial I/O timing



Note: This is available in clock synchronous serial I/O, using internal clock and transmission mode.

Fig. 61 External connection example in plural output of transmit clock mode

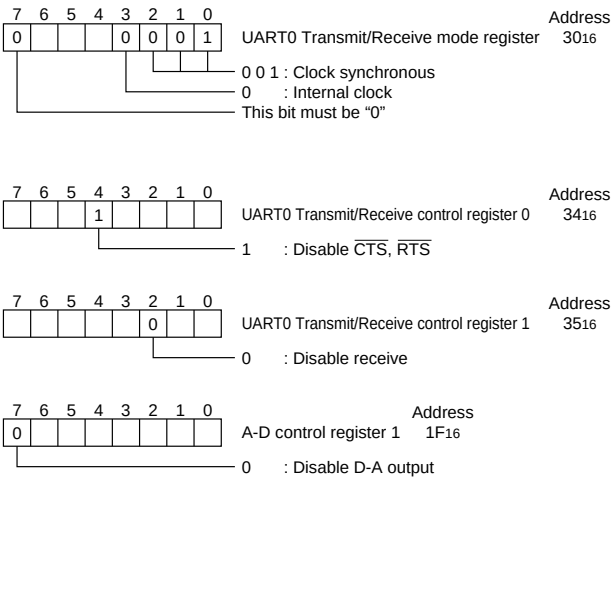


Fig. 62 Other registers except special function select register 1 bit configuration in plural output of transmit clock mode

Table 6. Output pin of transmit clock select bits and pins' function

Output pin of transmit clock select bits		Pin name		
TC1	TC0	P81/CLK0	P82/RxD0	P80/CTS0/RTS0/DA0
0	0	CLK0	RxD0	P80/CTS0/RTS0/DA0
0	1	CLK0	"H" **(Note)**	P80
1	0	"H"	CLKS2	P80
1	1	"H"	"H" **(Note)**	CLKS1

Note: It outputs "H" when bit 2 of the port P8 direction register is "1", and it becomes floating when bit 2 is "0".

Receive

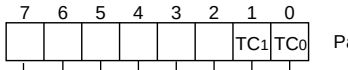
Receive starts when bit 2 (REk flag) of UARTk Transmit/Receive control register 1 is set to "1".

The $\overline{\text{RTSk}}$ output is "H" when the REk flag is "0" and goes "L" when the REk flag changed to "1" and the TIk flag did to "0". It goes back to "H" when receive starts. The TIk flag is cleared to "0" by write dummy data to the transmit buffer register. It is ready to receive when $\overline{\text{RTSk}}$ output is "L".

The data from the RxDk pin is retrieved and the contents of the receive register is shifted by 1 bit each time when the transmission clock CLKj changes from "L" to "H." When an 8-bit data is received, the contents of the receive register is transferred to the receive buffer register and bit 3 (RIk flag) of UARTk Transmit/Receive control register 1 is set to "1". In other words, the setting "1" to the RIk flag indicates that the receive buffer register contains the received data. When the RIk flag changes from "0" to "1", the interrupt request bit in the UARTk receive interrupt control register is set to "1". Bit 4 (OERk flag) of UARTk Transmit/Receive control register 1 is set to "1" when the next data is transferred from the receive register to the receive buffer register while RIk flag is "1", and indicates that the next data was transferred to the receive register before the contents of the receive buffer register was read. RIk flag is automatically cleared to "0" when the low-order byte of the receive buffer register is read or when the REk flag is cleared to "0". The OERk flag is cleared when the REk flag is cleared or port P8 is set to a parallel port. Bit 5 (FERk flag), bit 6 (PERk flag), and bit 7 (SUMk flag) are ignored in clock synchronous mode.

When reading the contents of the receive buffer register, the received data is pulled from the least significant bit (LSB) in the received order if bit 7 (TEM) of the UARTj Transmit/Receive control registers 0 is "0". If bit 7 (TEM) is "1", the received data is pulled from the most significant bit (MSB).

As shown in Figure 54, with clock synchronous serial communication, data cannot be received unless the transmitter is operating because the receive clock is created from the transmission clock. Therefore, the transmitter must be operating even when there is no need to sent data from UARTk to UARTj.



Transmit clock output pin select bit
00 : Normal mode (output only to CLK_0)
01 : Plural clocks specified; output to CLK_0
10 : Plural clocks specified; output to $CLKS_0$
11 : Plural clocks specified; output to $CLKS_1$

Internal clock stop select bit at WIT **(Note 1)**
0 : Clock for peripheral function and watchdog timer are operating at WIT
1 : Internal clock except that for oscillation circuit and watchdog timer are stopped at WIT

Watchdog timer's select bit **(Note 1)**
0 : Exclusive clock deviding circuit output (Wf_{512}, Wf_{32}) is used as clock for watchdog timer. Clock (Wf_{512}, Wf_{32}) for watchdog timer does not change in hold.
1 : Clock for peripheral device deviding circuit output (Pf_{512}, Pf_{32}) is used as clock for watchdog timer. Clock (Pf_{512}, Pf_{32}) for watchdog timer changes in hold. Watchdog timer exclusive clock dividing circuit is stopped.

Signal output stop select bit **(Note 1)**
Refer to Table 8.

Expansion function select bit **(Note 2)**
Refer to Figure 62.

Pull-up select bit 0 **(Note 3)**
0 : With no pull-up for $P5_7$, $P5_6$, $P5_5$, $P5_4$
1 : With pull-up for $P5_7$, $P5_6$, $P5_5$, $P5_4$

Pull-up select bit 1 **(Note 3)**
0 : With no pull-up for $P9_5$
1 : With pull-up for $P9_5$

Control bits affected by expansion function select bit

Register	Address	Bit
A-D control register 1	$1F_{16}$	5
Chip select area register	63_{16}	2, 5, 6, 7
Particular function select register 0	$6C_{16}$	0, 1, 5, 6
Particular function select register 1	$6D_{16}$	2, 3, 4

Control registers affected by expansion function select register

Register	Address
Waveform output mode register	$1A_{16}$
Dead-time timer	$1B_{16}$
Pulse output data register 1	$1C_{16}$
Pulse output data register 0	$1D_{16}$
Timer A write flag	45_{16}
$\overline{INT_4}$ interrupt control register	$6E_{16}$

Notes 1: Bits 2, 3, and 4 can be re-write after bit 5 (expansion function select bit) is set to "1."
2: After bit 5 is set to "1" once, bit 5 cannot be cleared to "0" except external reset and software reset.
3: Bits 6 and 7 are write-only bits and undefined at read. Do not use SEB or CLB insturuction when setting bits 0–7.

Fig. 63 Particular function select register 1 bit configuration

Selection of transfer format

In clock synchronous serial communication, transfer format can be selected by bit 7 of the Transmit/Receive control register 0. When bit 7 is “0”, transfer format is LSB first. When bit 7 is “1”, transfer format is MSB first.

This function is realized by changing connection relation between the transmit buffer register and the receive buffer register when writing transmit data to the transmit buffer register or reading receive data from the receive buffer register. Accordingly, the transmitter’s operation is the same in both transfer formats.

Figure 64 shows the connection relation.

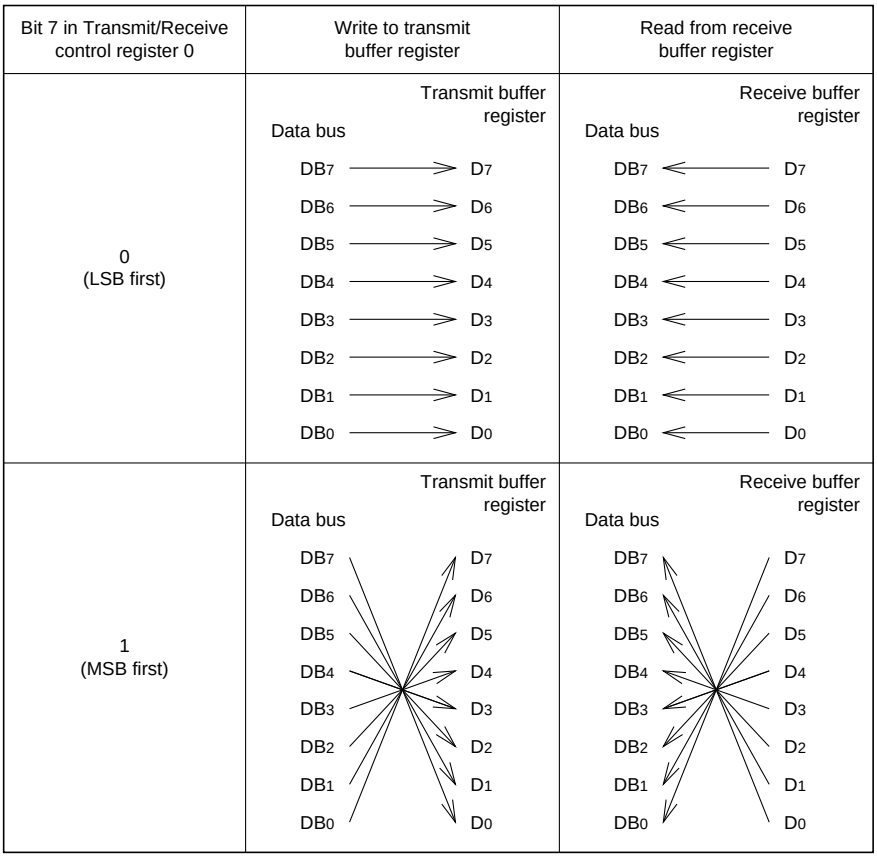


Fig. 64 Connection relation between transmit buffer register, receive buffer register, and data bus

ASYNCHRONOUS SERIAL COMMUNICATION

Asynchronous serial communication can be performed using 7-, 8-, or 9-bit length data. The operation is the same for all data lengths. The following is the description for 8-bit asynchronous communication.

With 8-bit asynchronous communication, bit 0 of UARTi Transmit/Receive mode register is "1", bit 1 is "0", and bit 2 is "1".

Bit 3 is used to select an internal clock or an external clock. If bit 3 is "0", an internal clock is selected and if bit 3 is "1", then external clock is selected. If an internal clock is selected, bit 0 (TCS_0) and bit 1 (TCS_1) of UARTi Transmit/Receive control register 0 are used to select the clock source. When an internal clock is selected for asynchronous serial communication, the CLKi pin can be used as a normal I/O pin.

The selected internal or external clock is divided by (n+1), then by 16, and is passed through a control circuit to create the UART transmission clock or UART receive clock.

Therefore, the transmission speed can be changed by changing the contents (n) of the bit rate generator. If the selected clock is an internal clock Pfi or an external clock f_{EXT},

Bit Rate = (Pfi or f_{EXT}) / {(n+1)×16}

Bit 4 is the stop bit length select bit to select 1 stop bit or 2 stop bits.

Bit 5 is a select bit of odd parity or even parity.

In the odd parity mode, the parity bit is adjusted so that the sum of 1s in the data and parity bit is always odd.

In the even parity mode, the parity bit is adjusted so that the sum of the 1s in the data and parity bit is always even.

Bit 6 is the parity bit select bit which indicates whether to add parity bit or not.

Bits 4 to 6 must be set or reset according to the data format used in the communicating devices.

Bit 7 is the sleep select bit. The sleep mode is described later.

The UARTi Transmit/Receive control register 0 bit 2 is used to determine whether to use $\overline{CTS}i$ input or $\overline{RTS}i$ output.

$\overline{CTS}i$ input is used if bit 2 is "0" and $\overline{RTS}i$ output is used if bit 2 is "1".

If $\overline{CTS}i$ input is selected, the user can control whether to stop or start transmission by external $\overline{CTS}i$ input.

Bit 4 of the UART Transmit/Receive control register 0 is used to determine whether to use $\overline{CTS}$ or $\overline{RTS}$ signal. Bit 4 must be "0" when $\overline{CTS}$ or $\overline{RTS}$ signal is used. Bit 4 must be "1" when $\overline{CTS}$ or $\overline{RTS}$ signal is not used. When $\overline{CTS}$ or $\overline{RTS}$ signal is not used, $\overline{CTS}$/$\overline{RTS}$ pin can be used as a normal port. The case using $\overline{CTS}$ and $\overline{RTS}$ signals are explained below. However, when $\overline{CTS}$ and $\overline{RTS}$ signals are not used, there are no condition of $\overline{CTS}i$ input, and there is no $\overline{RTS}i$ output.

Clear UARTj Transmit/Receive control register 0 bit 7 to "1" in asynchronous communication.

Transmission

Transmission is started when bit 0 (TEi flag) of UARTi Transmit/Receive control register 1 is "1", bit 1 (TIi flag) is "0", and $\overline{CTSi}$ input is "L" if $\overline{CTSi}$ input is selected. As shown in Figures 65 and 66, data is output from the TxDi pin with the stop bit or parity bit specified by bits 4 to 6 of UARTi Transmit/Receive mode register. The data is output from the least significant bit.

The TIi flag indicates whether the transmit buffer is empty or not. It is cleared to "0" when data is written in the transmit buffer, and is set to "1" when the contents of the transmit buffer register is transferred to the transmit register.

When the transmit register becomes empty after the contents has been transmitted, data is transferred automatically from the transmit buffer register to the transmit register if the next transmit start condition is satisfied.

Once transmission has started, the TEi flag, TIi flag, and $\overline{CTSi}$ signal (if $\overline{CTSi}$ input is selected) are ignored until data transmission is completed.

Therefore, transmission does not stop until it completes event if the TEi flag is cleared during transmission.

The transmission start condition indicated by TEi flag, TIi flag, and $\overline{CTSi}$ is checked while the TENDi signal shown in Figure 65 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmit buffer register and TIi flag is cleared to "0" before the TENDi signal goes "H".

Bit 3 (TxEPTYi flag) of UARTi Transmit/Receive control register 0 changes to "1" at the next cycle just after the TENDi signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmission is completed.

When the TIi flag changes from "0" to "1", the interrupt request bit of the UARTi transmit interrupt control register is set to "1".

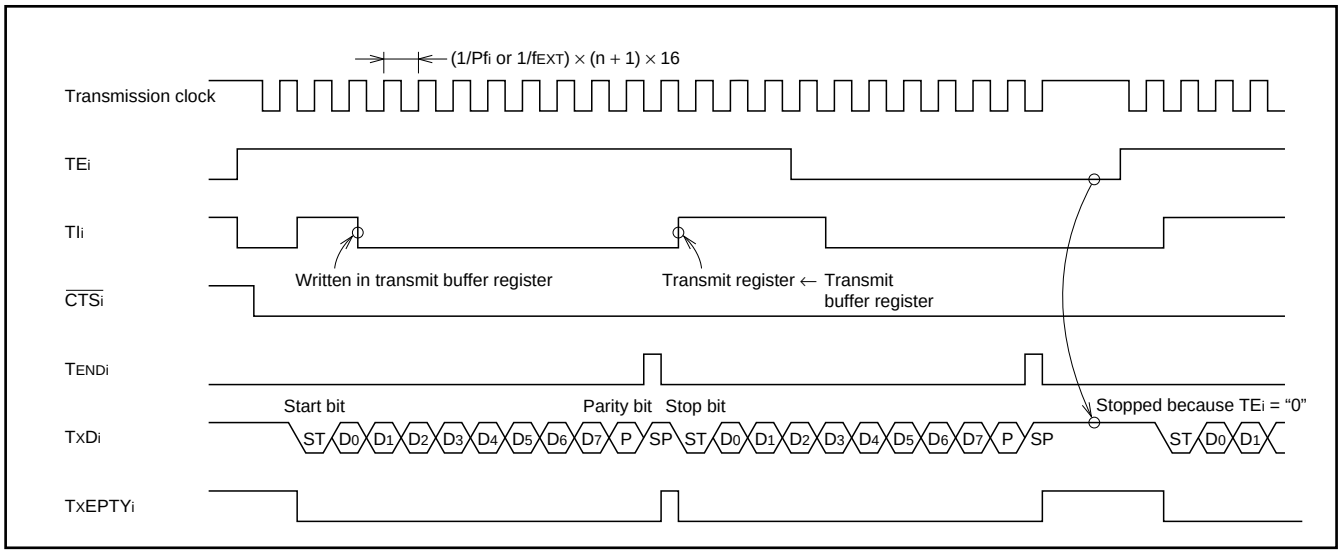


Fig. 65 Transmit timing example when 8-bit asynchronous communication with parity and 1 stop bit selected

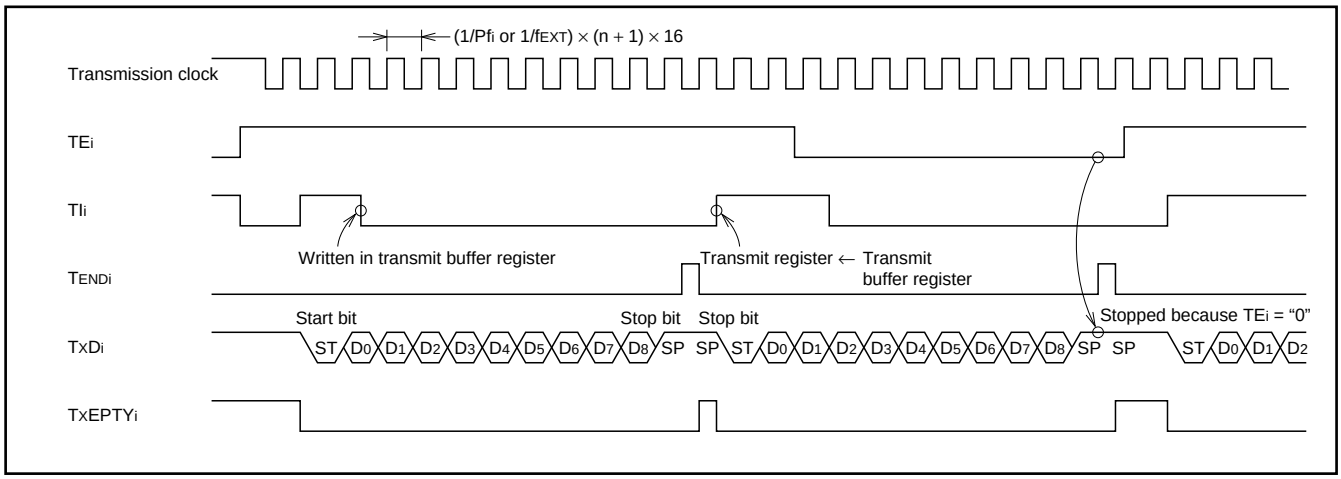


Fig. 66 Transmit timing example when 9-bit asynchronous communication with no parity and 2 stop bits selected

Receive

Receive is enabled when bit 2 (REi flag) of UARTi Transmit/Receive control register 1 is set to "1." As shown in Figure 67, the frequency divider circuit (1/16) at the receiving side begin to work when a start bit arrives and the data is received.

If $\overline{\text{RTSi}}$ output is selected by setting bit 2 of UARTi Transmit/Receive control register 0 to "1", the $\overline{\text{RTSi}}$ output is "H" when the REi flag is "0". When the REi flag changes to "1", the $\overline{\text{RTSi}}$ output goes "L" to indicate receive ready and returns to "H" once receive has started. In other words, $\overline{\text{RTSi}}$ output can be used to determine externally whether the receive register is ready to receive.

The entire transmission data bits are received when the start bit passes the final bit of the receive block shown in Figure 56. At this point, the contents of the receive register is transferred to the receive buffer register and bit 3 (RIi flag) of UARTi Transmit/Receive control register 1 is set to "1." In other words, the RIi flag indicates that the receive buffer register contains data when it is set to "1." If $\overline{\text{RTSi}}$ output is selected, $\overline{\text{RTSi}}$ output goes "L" to indicate that the register is ready to receive the next data.

The interrupt request bit of the UARTi receive interrupt control register is set to "1" when the RIi flag changes from "0" to "1".

Bit 4 (OERi flag) of UARTi Transmit/Receive control register 1 is set to "1" when the next data is transferred from the receive register to the receive buffer register while the RIi flag is "1", in other words, when an overrun error occurs. If the OERi flag is "1", it indicates that the next data has been transferred to the receive buffer register before the contents of the receive buffer register has been read.

Bit 5 (FERi flag) is set to "1" when the number of stop bits is less than required (framing error).

Bit 6 (PERi flag) is set to "1" when a parity error occurs.

Bit 7 (SUMi flag) is set to "1" when either the OERi flag, FERi flag, or the PERi flag is set to "1." Therefore, the SUMi flag can be used to determine whether there is an error.

The setting of the RIi flag, OERi flag, FERi flag, and the PERi flag is performed while transferring the contents of the receive register to the receive buffer register. The RIi, FERi, and PERi flags are cleared to "0" when reading the low-order byte of the receive buffer register or when writing "0" to the REi flag or when setting to a parallel port.

The OERi and SUMi flags are cleared to "0" when writing "0" to the REi flag or when setting to a parallel port.

The SUMi flag is cleared to "0" when the OERi, FERi, PERi flags are cleared to "0" all.

Sleep mode

The sleep mode is used to communicate only between certain microcomputers when multiple microcomputers are connected through serial I/O.

The microcomputer enters the sleep mode when bit 7 of UARTi Transmit/Receive mode register is set to "1."

The operation of the sleep mode for an 8-bit asynchronous communication is described below.

When sleep mode is selected, the contents of the receive register is not transferred to the receive buffer register if bit 7 (bit 6 if 7-bit asynchronous communication and bit 8 if 9-bit asynchronous communication) of the received data is "0". Also the RIi, OERi, FERi, PERi, and the SUMi flag are unchanged. Therefore, the interrupt request bit of the UARTi receive interrupt control register is also unchanged. Normal receive operation takes place when bit 7 of the received data is "1".

The following is an example of how the sleep mode can be used.

The main microcomputer first sends data: bit 7 is "1" and bits 0 to 6 are set to the address of the subordinate microcomputer to be communicated with. Then all subordinate microcomputers receive this data. Each subordinate microcomputer checks the received data, clears the sleep bit to "0" if bits 0 to 6 are its own address and sets the sleep bit to "1" if not. Next, the main microcomputer sends data with bit 7 cleared. Then the microcomputer which cleared the sleep bit will receive the data, but the microcomputers which set the sleep bit to "1" will not. In this way, the main microcomputer is able to communicate only with the designated microcomputer.

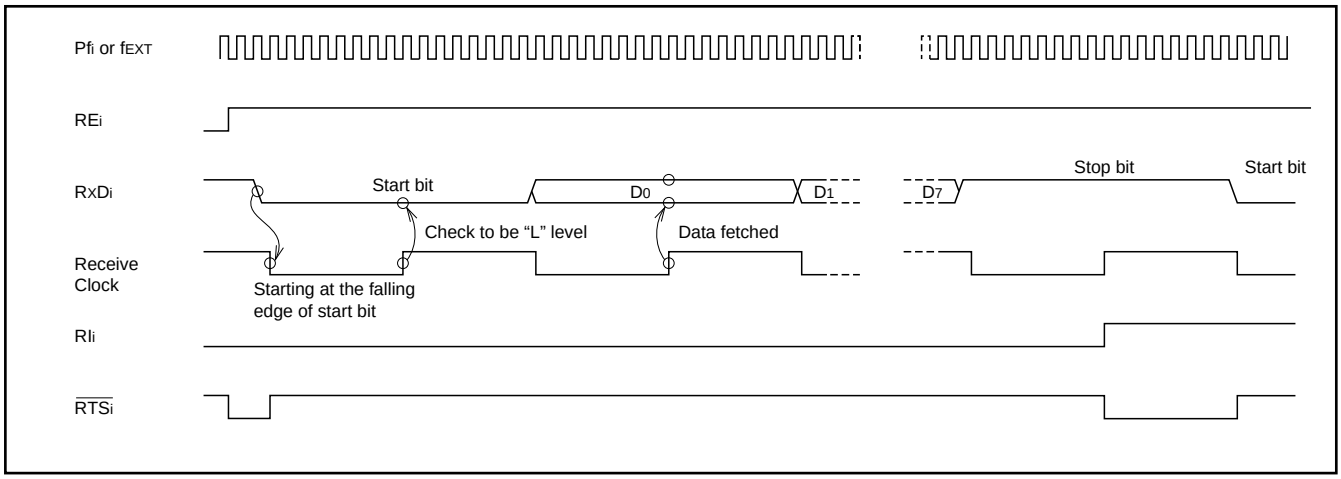


Fig. 67 Receive timing example when 8-bit asynchronous communication with no parity and 1 stop bit selected

A-D CONVERTER

The A-D converter is a 10-bit successive approximation converter. The use of A-D converter or the use of comparator can be selected for each A-D input pin. The contents of the comparator function select register specify it.

Figure 68 shows a block diagram of the A-D converter.

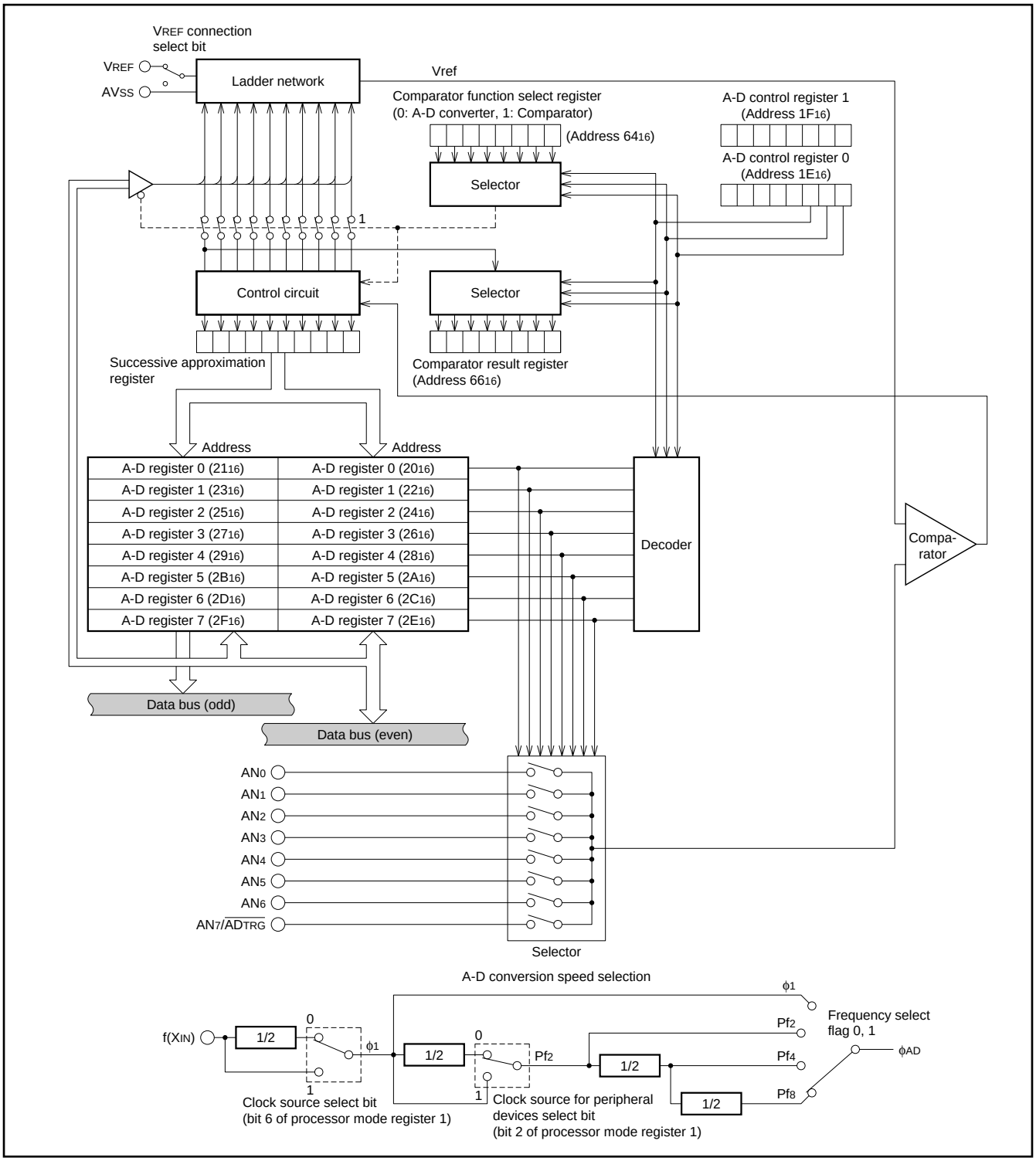


Fig. 68 A-D converter block diagram

Figure 69 shows the comparator function select register (address 64_{16}) bit configuration. Bits 7 to 0 correspond to channels 7 to 0 respectively. Each channel can be selected as either an A-D converter or a comparator. When the bit is "0", the channel corresponding to it functions as a 10-bit or an 8-bit A-D converter. When the bit is "1", the channel functions as a comparator.

When selecting an A-D converter, an input voltage to a selected analog input pin is A-D converted and the result is stored into the A-D register.

When selecting a comparator, D-A conversion is performed to the value of which high-order 8 bits are the value stored in an even address of the A-D converter and of which low-order 2 bits are "10_2." Then, this D-A converted value is compared with the voltage supplied to an analog input pin. After the comparison, when the voltage supplied to an analog input pin is higher, "1" is stored into the comparator result register (address 66_{16}) shown in Figure 70. When it is lower, "0" is stored into that register.

Be sure to perform only read to the A-D register of which channel is selected as an A-D converter, and perform only write to the A-D register of which channel is selected as a comparator. Additionally, do not write to the comparator function select register and the A-D register while an A-D converter or a comparator is operating.

Port direction register's bits corresponding to pins to be A-D converted must be "0" (input mode) because analog input ports are multiplexed with port P7.

Figure 71 shows the bit configuration of the A-D control register 0 (address $1E_{16}$) and the A-D control register 1 (address $1F_{16}$).

An operation clock (ϕ_{AD}) of an A-D converter or a comparator can be selected with bit 7 of the A-D control register 0 and bit 4 of the A-D control register 1.

When bit 4 (frequency select flag 1) of the A-D control register 1 is "0", ϕ_{AD} becomes Pf_8 when bit 7 (frequency select flag 0) of the A-D control register 0 is "0", ϕ_{AD} becomes Pf_4 when bit 7 of the A-D control register 0 is "1".

When the frequency select flag 1 is "1", ϕ_{AD} becomes Pf_2 when the frequency select flag 0 is "0", ϕ_{AD} becomes ϕ_1 when the frequency select flag 0 is "1". The last case is used when ϕ_1 is forcibly used as ϕ_{AD} in high-speed running ($f(X_{IN}) > \phi_1 > 12.5$ MHz). However, this selection is available only in 8-bit resolution mode.

ϕ_{AD} during A-D conversion must be 250 kHz or more because the comparator uses a capacity coupling amplifier.

Bit 3 of A-D control register 1 is used to select whether to regard the conversion result as 10-bit or as 8-bit data. The conversion result is regarded as 10-bit data when bit 3 is "1" and as 8-bit data when bit 3 is "0".

When the conversion result is used as 10-bit data, the low-order 8 bits of the conversion result is stored in the even address of the corresponding A-D register and the high-order 2 bits are stored in bits 0 and 1 at the odd address of the corresponding A-D register. Bits 2 to 7 of the A-D register odd address are "000000_2" when read.

When the conversion result is used as 8-bit data, the high-order 8 bits of the 10-bit A-D conversion result are stored in even address of the corresponding A-D register. In this case, the value at the A-D register's odd address is "00_{16}" when read.

Whether to connect the reference voltage input (V_{REF}) with the ladder network or not depends on bit 5 of the A-D control register 1. The V_{REF} pin is connected when bit 5 is "0" and is disconnected when bit 5 is "1" (High impedance state).

When A-D or D-A conversion is not performed, current from the V_{REF} pin to the ladder network can be cut off by disconnecting ladder network from the V_{REF} pin.

Before starting A-D or D-A conversion, wait for 1 μs or more after clearing bit 5 to "0".

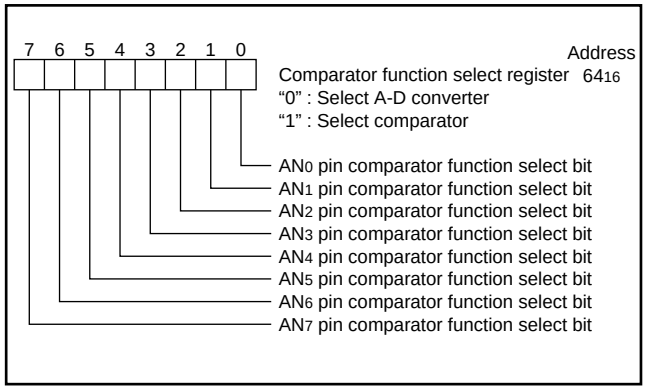


Fig. 69 Comparator function select register bit configuration

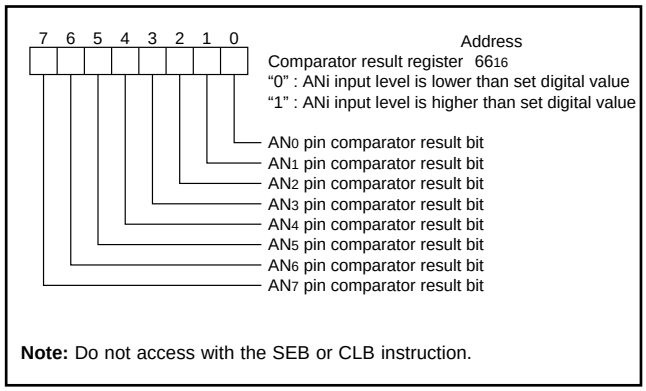


Fig. 70 Comparator result register bit configuration

Operation mode

The operation mode is selected by bits 3 and 4 of A-D control register 0 and bit 2 of A-D control register 1. The available operation modes are one-shot, repeat, single sweep, repeat sweep 0, and repeat sweep 1. Either an A-D converter or a comparator can be selected respectively for every pin in the following 5 modes. The following description applies to the case where the bit of the comparator function select register is "0" and an A-D converter is selected. It also applies to a comparator's operation except that an A-D conversion is changed to a comparator operation and the result of a comparison is stored into the comparator result register.

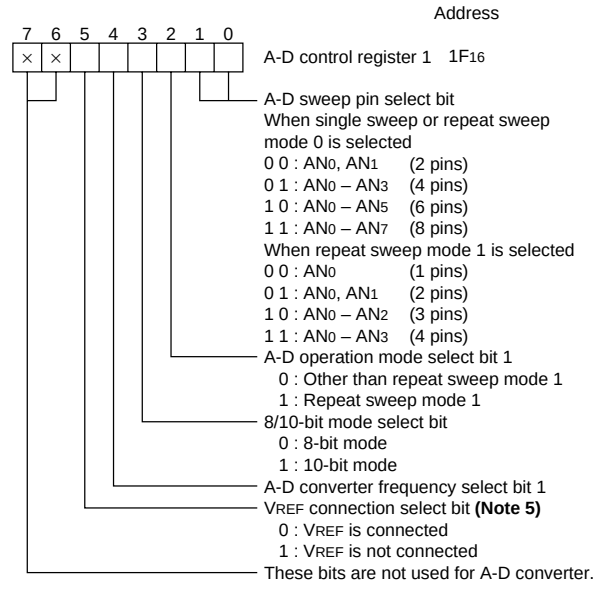
(1) One-shot mode

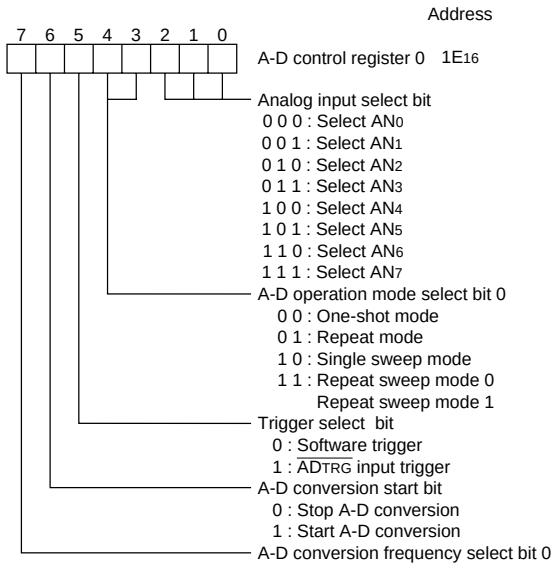
One-shot mode is selected when bits 3 and 4 of A-D control register 0 are "0" and bit 2 of A-D control register 1 is "0". The A-D conversion pins are selected with bits 0 to 2 of A-D control register 0. A-D conversion can be started by a software trigger or by an external trigger. A software trigger is selected when bit 5 of A-D control register 0 is "0" and an external trigger is selected when it is "1". When a software trigger is selected, A-D conversion or comparator operation is started when bit 6 (A-D conversion start flag) is set to "1."

When the bit of comparator function select register is "0" and bit 3 of A-D control register 1 is "1", A-D conversion ends after 59 ϕAD cycles, and the interrupt request bit of the A-D interrupt control register is set to "1." At the same time, A-D control register 0 bit 6 (A-D conversion start bit) is cleared to "0" and A-D conversion stops. The result of A-D conversion is stored in the A-D register corresponding to the selected pin.

When the bit of the comparator function select register is "1", a comparator operation ends after 14 ϕAD cycles and the interrupt request bit of the A-D interrupt control register is set to "1". At the same time, the A-D control register 0 bit 6 (A-D conversion start bit) is cleared to "0" and the comparator operation stops. The result of the comparison is stored into the bit of the comparator result register corresponding to the selected pin.

If an external trigger is selected, A-D conversion starts when the A-D conversion start bit is "1" and the $\overline{\text{ADTRG}}$ input changes from "H" to "L". In this case, the pins that can be used for A-D conversion are AN0 to AN6 because the $\overline{\text{ADTRG}}$ pin is multiplexed with analog voltage input pin AN7. This operation is the same as that for software trigger except that the A-D conversion start bit is not cleared after A-D conversion and a retrigger can be available during A-D conversion.





Bit 6 at address 5F16 (Note 1)	Bit 2 at address 5F16 (Note 2)	A-D conversion frequency select bit		ϕAD
		Bit 1	Bit 0	
0	0	0	0	f(XIN)/16
		0	1	f(XIN)/8
		1	0	f(XIN)/4
		1	1	f(XIN)/2 (Note 3)
	1	0	0	f(XIN)/8
		0	1	f(XIN)/4
		1	0	f(XIN)/2
		1	1	

Notes1, 2: Refer to Figure 9 Processor mode register 1 bit configuration.
3: When f(XIN) > 25 MHz, this can be selected only in 8-bit resolution mode.

Bit 6 at address 5F16 (Note 1)	Bit 2 at address 5F16 (Note 2)	A-D conversion frequency select bit		ϕAD
		Bit 1	Bit 0	
1	0	0	0	f(XIN)/8
		0	1	f(XIN)/4
		1	0	f(XIN)/2
		1	1	f(XIN) (Note 4)
	1	0	0	f(XIN)/4
		0	1	f(XIN)/2
		1	0	f(XIN)
		1	1	

Notes 4: When f(XIN) > 12.5 MHz, this can be selected only in 8-bit resolution mode.
5: When the expansion function select bit (bit 5 of particular function select register 1 ; refer to Fig. 62) is "1", bit 5 can be written and changed.

Fig. 71 A-D control register bit configuration

(2) Repeat mode

Repeat mode is selected when bit 3 of A-D control register 0 is "1", bit 4 is "0" and bit 2 of A-D control register 1 is "0".

The operation of this mode is the same as the operation of one-shot mode except that when A-D conversion of the selected pin is complete and the result is stored in the A-D register, conversion does not stop, but is repeated.

No interrupt request is generated in this mode. Furthermore, if software trigger is selected, the A-D conversion start bit is not cleared.

The contents of the A-D register can be read at any time.

Be sure not to write to the A-D register corresponding to the pins selected for a comparator during operation.

(3) Single sweep mode

Single sweep mode is selected when bit 3 of A-D control register 0 is "0", bit 4 is "1" and bit 2 of A-D control register 1 is "0".

In the single sweep mode, the number of analog input pins to be swept can be selected. Analog input pins are selected by bits 1 and 0 of the A-D control register 1 (address $1F_{16}$). Two pins, four pins, six pins, or eight pins can be selected as analog input pins, depending on the contents of these bits.

A-D conversion is performed only for selected input pins. After A-D conversion is performed for input of AN_0 pin, the conversion result is stored in A-D register 0, and in the same way, A-D conversion is performed for selected pins one after another. After A-D conversion is performed for all selected pins, the sweep is stopped.

A-D conversion can be started with a software trigger or with an external trigger input. A software trigger is selected when bit 5 is "0" and an external trigger is selected when it is "1".

When a software trigger is selected, A-D conversion is started when A-D control register 0 bit 6 (A-D conversion start bit) is set to "1."

When A-D conversion of all selected pins end, the interrupt request bit of the A-D conversion interrupt control register is set to "1." At the same time, A-D conversion start bit is cleared to "0" and A-D conversion stops.

When an external trigger is selected, A-D conversion starts when the A-D conversion start bit is "1" and the $\overline{ADTRG}$ input changes from "H" to "L". In this case, the A-D conversion result which is stored in the A-D register 7 becomes invalid because the $\overline{ADTRG}$ pin is multiplexed with AN_7 pin.

The operation by external trigger is the same as that by software trigger except that the A-D conversion start bit is not cleared to "0" after A-D conversion and a retrigger can be available during A-D conversion.

(4) Repeat sweep mode 0

Repeat sweep mode 0 is selected when bit 3 of A-D control register 0 is "1", bit 4 is "1" and bit 2 of A-D control register 1 is "0".

The difference from the single sweep mode is that A-D conversion does not stop after conversion for all selected pins, but repeats again from the AN_0 pin. The repeat is performed among the selected pins. Also, no interrupt request is generated. Furthermore, if software trigger is selected, the A-D convension start bit is not cleared.

The A-D register can be read at any time.

Be sure not to write to the A-D register corresponding to the pins selected for a comparator during operation.

(5) Repeat sweep mode 1

Repeat sweep mode 1 is selected when bit 3 of A-D control register 0 is "1", bit 4 is "1" and bit 2 of A-D control register 1 is "1".

The difference from the repeat sweep mode 0 is that A-D conversion for one unselected pin is performed each time when A-D conversion for selected pins is completed and A-D conversion is repeated once again from AN_0 pin. The number of analog input pins to be swept is also different.

The analog input pins for repeatedly sweep are selected with bits 1 and 0 of A-D control register 1. The contents of these pins are used to select one pin, two pins, three pins, or four pins.

The unselected pins are converted from the pin next to the pins selected as repeat sweep pins. No interrupt request is generated. Furthermore, if software trigger is selected, the A-D conversion start bit is not cleared.

The A-D register can be read at any time.

Be sure not to write to the A-D register corresponding to the pins selected for a comparator during operation.

Note: Clear the interrupt request bit of the A-D interrupt control register (bit 3 at address 70_{16}) before using the A-D interrupt. It is because the interrupt request bit is undefined just after reset.

D-A CONVERTER

The D-A converter is an 8-bit R-2R method D-A converter and consists of two independent D-A converters. Figure 72 shows the block diagram of the D-A converter and Figure 73 shows the bit configuration of A-D control register 1.

D-A conversion is performed by writing a value in the corresponding D-A register. The conversion result is output by bits 6 and 7 of A-D control register 1 (address $1F_{16}$). When bit 7 is "1", the conversion result is output from DA_0 pin.

When bit 6 is "1", the conversion result is output from DA_1pin.

The output analog voltage V is determined according to the value n ("n" is a decimal number) set in the D-A register.

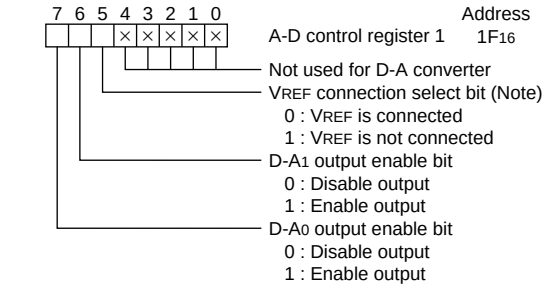
$V = V_{REF} \times n/256$ (n = 0 to 255)

V_{REF} : Reference voltage

The D-A output enable bit is cleared to "0" at reset. Whether to connect the reference voltage input (V_{REF}) with the ladder network or not depends on bit 5 of the A-D control register 1. The V_{REF} pin is connected when bit 5 is "0" and is disconnected when bit 5 is "1" (High impedance state).

When A-D or D-A conversion is not performed, current from the V_{REF} pin to the ladder network can be cut off by disconnecting ladder network from the V_{REF} pin.

Before starting A-D or D-A conversion, wait for 1 μs or more after clearing bit 5 to "0". An external buffer must be connected when connecting to a low impedance load because there is no built-in D-A output buffer.



Note : When the expansion function select bit (bit 5 of peripheral function select register 1 ; refer to Fig. 62) is "1," bit 5 can be written and changed.

Fig. 73 A-D control register 1 bit configuration

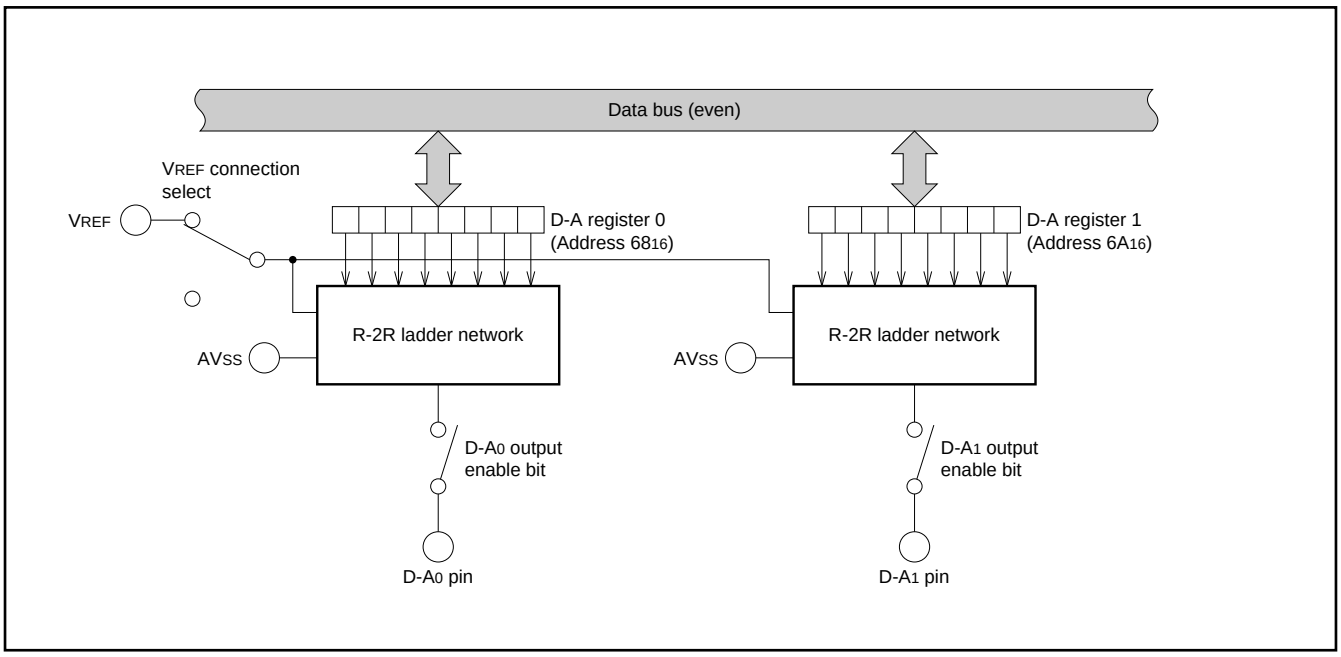


Fig. 72 D-A converter block diagram

WATCHDOG TIMER

The watchdog timer is used to detect unexpected execution sequence caused by software runaway and others. Figure 74 shows the block diagram of the watchdog timer.

The watchdog timer consists of a 12-bit binary counter.

The watchdog timer counts clock Wf_{32}/Pf_{32}, which is obtained by dividing the peripheral devices' clock Pf_2 by 16; or clock Wf_{512}/Pf_{512}, which is obtained by doing it by 256. The watchdog timer frequency select register shown in Figure 75 selects which clock is counted.

Wf_{512}/Pf_{512} is selected when its contents are "0", and Wf_{32}/Pf_{32} is selected when they are "1". They are cleared to "0" after reset.

The watchdog timer clock select bit (bit 3 of particular function select register 1; Figure 62) selects use of clock Wf_{512}/Wf_{32} or Pf_{512}/Pf_{32} as the clock source of watchdog timer. When selecting Wf_{512}/Wf_{32}, the clock source of watchdog timer (Wf_{512}/Wf_{32}) is not active during Hold state. When selecting Pf_{512}/Pf_{32}, the clock source of watchdog timer (Pf_{512}/Pf_{32}) is active during Hold state, however, current consumption can be reduced. It is because the Wf_{512}/Wf_{32} division circuit stops.

FFF_{16} is set in the watchdog timer when "L" or 2Vcc is applied to the $\overline{RESET}$ pin, STP instruction is executed, data is written to the watchdog timer, or the most significant bit of the watchdog timer becomes "0".

After FFF_{16} is set in the watchdog timer, when the watchdog timer counts the clock source by 2048 counts, the most significant bit of watchdog timer becomes "0", the watchdog timer interrupt request bit is set to "1", and FFF_{16} is set again in the watchdog timer.

Normally, a program is written so that data is written in the watchdog timer before the most significant bit of the watchdog timer becomes "0". If this routine is not executed owing to unexpected program execution and others, the most significant bit of the watchdog timer becomes "0" and an interrupt is generated.

The microcomputer can be reset by writing "1" to bit 3 (software reset bit) of processor mode register 0 in the interrupt routine, described in Figure 16 in the interrupt section, and generating a reset pulse.

The watchdog timer stops its function when the $\overline{RESET}$ pin voltage is raised to double the Vcc voltage.

The watchdog timer can also be used to return from when the clock is stopped by the STP instruction. Refer to the section on the clock generating circuit for more details.

The watchdog timer also becomes Hold state during Hold state and the clock input to it is stopped.

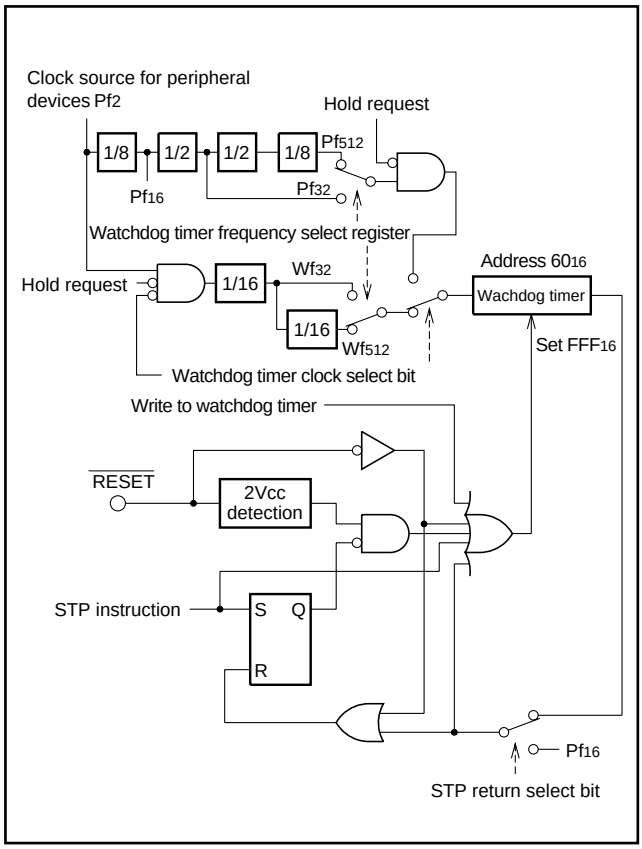


Fig. 74 Watchdog timer block diagram

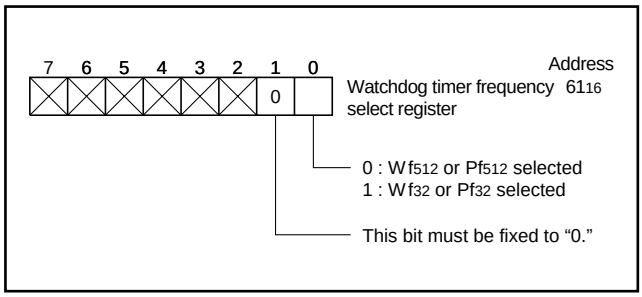


Fig. 75 Watchdog timer frequency select register bit configuration

RESET CIRCUIT

Reset is released when the $\overline{\text{RESET}}$ pin is returned to "H" level after holding it at "L" level while the supply voltage is at 5V ±10%. As the result, program execution starts at the address formed by setting the address A_{23}–A_{16} to 00_{16}, A_{15}–A_8 to the contents of address $FFFF_{16}$, and A_7–A_0 to the contents of address $FFFE_{16}$.

Figure 76 shows the status of the internal registers during reset.

Figure 77 shows an example of a reset circuit. When a stabilized clock is input from the external to the oscillation circuit, the reset input voltage must be held 0.9V or lower when the supply voltage reaches 4.5V. When connecting a resonator to the oscillation circuit, return the reset input voltage from "L" to "H" after the main-clock oscillation is fully stabilized

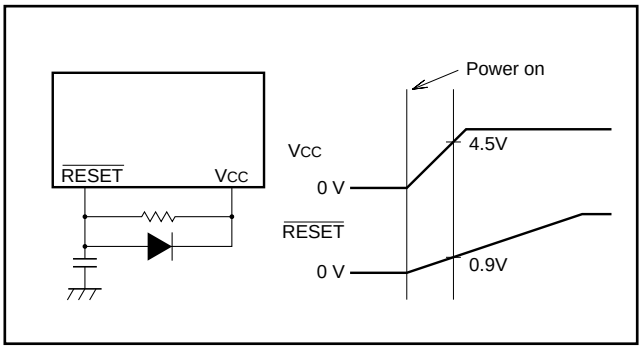


Fig. 77 Reset circuit example (perform careful evaluation at system design before using)

INPUT/OUTPUT PINS

Ports P0 to P11 all have the direction register and each bit can be programmed for input or output. A pin becomes an output pin when the corresponding bit of direction register is "1", and an input pin when it is "0".

When a pin is programmed for output, the data is written to its port latch and it is output to the output pin. When a pin is programmed for output, the contents of the port latch is read instead of the value of the pin. Accordingly, a previously output value can be read correctly even when the output "H" voltage is lowered or the output "L" voltage is raised owing to an external load and others.

A pin programmed as an input pin is floating, and the value input to the pin can be read. When a pin is programmed as an input pin, the data is written only in the port latch and the pin remains floating.

Additionally, ports $P9_5$, $P5_4$ to $P5_7$ include pull-up transistors. The pull-up function of ports is selected with bits 7 and 6 of the particular function select register 1. Refer to the section on Interrupts for the pull-up function.

Figures 78 and 79 show block diagrams of ports P0 to P11 in the single-chip mode and E output.

Ports P0 to P4, P10 and P11 are also used as pins of address, data and control signals. Refer to the section on Processor mode for more details.

	Address	
Port P0 direction register	(04$_{16}$)···	00$_{16}$
Port P1 direction register	(05$_{16}$)···	00$_{16}$
Port P2 direction register	(08$_{16}$)···	0 X X X 0 0 0 0
Port P3 direction register	(09$_{16}$)···	X X X X 0 0 0 0
Port P4 direction register	(0C$_{16}$)···	00$_{16}$
Port P5 direction register	(0D$_{16}$)···	00$_{16}$
Port P6 direction register	(10$_{16}$)···	00$_{16}$
Port P7 direction register	(11$_{16}$)···	00$_{16}$
Port P8 direction register	(14$_{16}$)···	00$_{16}$
Port P9 direction register	(15$_{16}$)···	X X 0 0 0 0 0 0
Port P10 direction register	(18$_{16}$)···	00$_{16}$
Port P11 direction register	(19$_{16}$)···	00$_{16}$
Waveform output mode register	(1A$_{16}$)···	00$_{16}$
Pulse output data register 1	(1C$_{16}$)···	00$_{16}$
Pulse output data register 0	(1D$_{16}$)···	0 0 0 X 0 0 0 0
A-D control register 0	(1E$_{16}$)···	0 0 0 0 0 ? ? ?
A-D control register 1	(1F$_{16}$)···	0 0 0 0 0 0 1 1
UART 0 Transmit/Receive mode register	(30$_{16}$)···	00$_{16}$
UART 1 Transmit/Receive mode register	(38$_{16}$)···	00$_{16}$
UART 0 Transmit/Receive control register 0	(34$_{16}$)···	0 X X 0 1 0 0 0
UART 1 Transmit/Receive control register 0	(3C$_{16}$)···	0 X X 0 1 0 0 0
UART 0 Transmit/Receive control register 1	(35$_{16}$)···	0 0 0 0 0 0 1 0
UART 1 Transmit/Receive control register 1	(3D$_{16}$)···	0 0 0 0 0 0 1 0
Count start register	(40$_{16}$)···	00$_{16}$
One-shot start register	(42$_{16}$)···	X X X 0 0 0 0 0
Up-down register	(44$_{16}$)···	0 0 0 0 0 0 0 0
Timer A write register	(45$_{16}$)···	X X X X X 0 0 0
Timer A0 mode register	(56$_{16}$)···	00$_{16}$
Timer A1 mode register	(57$_{16}$)···	00$_{16}$
Timer A2 mode register	(58$_{16}$)···	00$_{16}$
Timer A3 mode register	(59$_{16}$)···	00$_{16}$
Timer A4 mode register	(5A$_{16}$)···	00$_{16}$
Timer B0 mode register	(5B$_{16}$)···	0 0 1 X 0 0 0 0
Timer B1 mode register	(5C$_{16}$)···	0 0 1 X 0 0 0 0
Timer B2 mode register	(5D$_{16}$)···	0 0 1 X 0 0 0 0
Processor mode register 0	(5E$_{16}$)···	0 0 0 0 0 0 0 0
Processor mode register 1	(5F$_{16}$)···	00$_{16}$

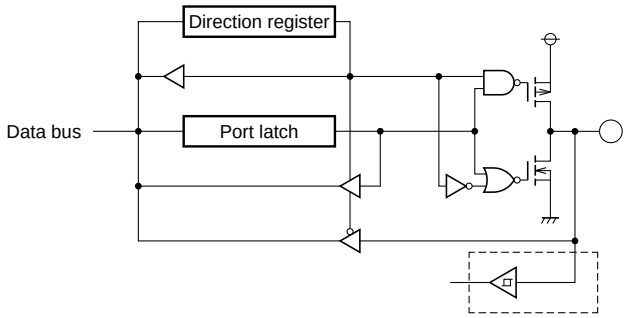
	Address	
Watchdog timer	(60$_{16}$)···	FFF$_{16}$
Watchdog timer frequency select register	(61$_{16}$)···	X X X X X X 0 0
Chip select control register	(62$_{16}$)···	0 0 0 0 0 0 0 0
Chip select area register	(63$_{16}$)···	0 0 0 X X 0 0 0
Comparator function select register	(64$_{16}$)···	0 0 0 0 0 0 0 0
Comparator result register	(66$_{16}$)···	0 0 0 0 0 0 0 0
D-A register 0	(68$_{16}$)···	0 0 0 0 0 0 0 0
D-A register 1	(6A$_{16}$)···	0 0 0 0 0 0 0 0
Particular function select register 0	(6C$_{16}$)···	0 0 0 0 0 0 0 0
Particular function select register 1	(6D$_{16}$)···	0 0 0 0 0 0 0 0
$\overline{\text{INT}}_4$ interrupt control register	(6E$_{16}$)···	X X 0 0 0 0 0 0
$\overline{\text{INT}}_3$ interrupt control register	(6F$_{16}$)···	0 0 0 0 0 0 0 0
A-D interrupt control register	(70$_{16}$)···	X X X X ? 0 0 0
UART 0 transmit interrupt control register	(71$_{16}$)···	X X X X 0 0 0 0
UART 0 receive interrupt control register	(72$_{16}$)···	X X X X 0 0 0 0
UART 1 transmit interrupt control register	(73$_{16}$)···	X X X X 0 0 0 0
UART 1 receive interrupt control register	(74$_{16}$)···	X X X X 0 0 0 0
Timer A0 interrupt control register	(75$_{16}$)···	X X X X 0 0 0 0
Timer A1 interrupt control register	(76$_{16}$)···	X X X X 0 0 0 0
Timer A2 interrupt control register	(77$_{16}$)···	X X X X 0 0 0 0
Timer A3 interrupt control register	(78$_{16}$)···	X X X X 0 0 0 0
Timer A4 interrupt control register	(79$_{16}$)···	X X X X 0 0 0 0
Timer B0 interrupt control register	(7A$_{16}$)···	X X X X 0 0 0 0
Timer B1 interrupt control register	(7B$_{16}$)···	X X X X 0 0 0 0
Timer B2 interrupt control register	(7C$_{16}$)···	X X X X 0 0 0 0
$\overline{\text{INT}}_0$ interrupt control register	(7D$_{16}$)···	X X 0 0 0 0 0 0
$\overline{\text{INT}}_1$ interrupt control register	(7E$_{16}$)···	X X 0 0 0 0 0 0
$\overline{\text{INT}}_2$ interrupt control register	(7F$_{16}$)···	X X 0 0 0 0 0 0
Processor status register PS		0 0 0 ? ? 0 0 0 1 ? ?
Program bank register PG		00$_{16}$
Program counter PC$_H$		Contents of FFFF$_{16}$
Program counter PC$_L$		Contents of FFFE$_{16}$
Direct page register DPR		0000$_{16}$
Data bank register DT		00$_{16}$

Contents of other registers and RAM are not initiallzed and must be initiallzed by software.

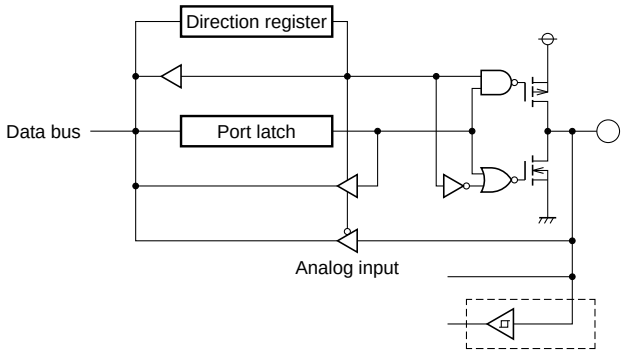
Note : Bit 0 of chip select control register (address 62$_{16}$) becomes "0" when CNVss pin level is "L"; that bit becomes "1" when the pin level is "H".

Fig. 76 Microcomputer internal registers status after reset

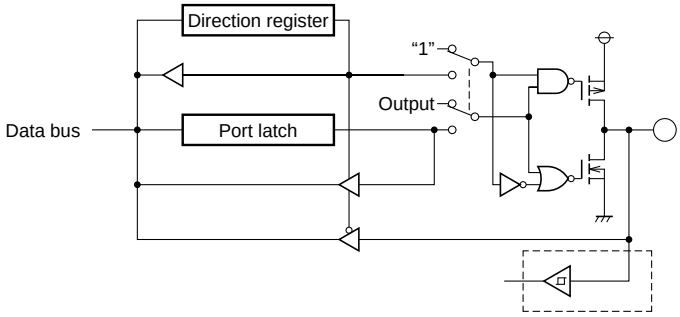
• Port $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_3$, $P2_7$, $P3_0$ to $P3_3$, $P4_3$ to $P4_6$, $P10_0$ to $P10_7$, $P11_0$ to $P11_7$ (Inside dotted-line not included)
Port $P4_0$, $P4_1$, $P4_7$, $P5_1$, $P5_3$, $P6_1$ to $P6_7$, $P8_6$ (Inside dotted-line included)



• Port $P7_0$ to $P7_6$ (Inside dotted-line not included)
• Port $P7_7$ (Inside dotted-line included)



• Port $P4_2$, $P8_3$, $P8_7$, $P9_0$ to $P9_4$ (Inside dotted-line not included)
Port $P5_0$, $P5_2$, $P6_0$, $P8_2$ (Inside dotted-line included)



• Port $P5_4$, $P5_6$

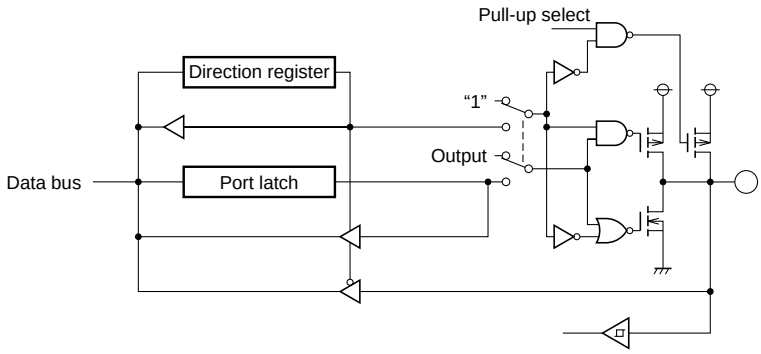


Fig. 78 Block diagram for ports P0 to P11 in single-chip mode and E output (1)

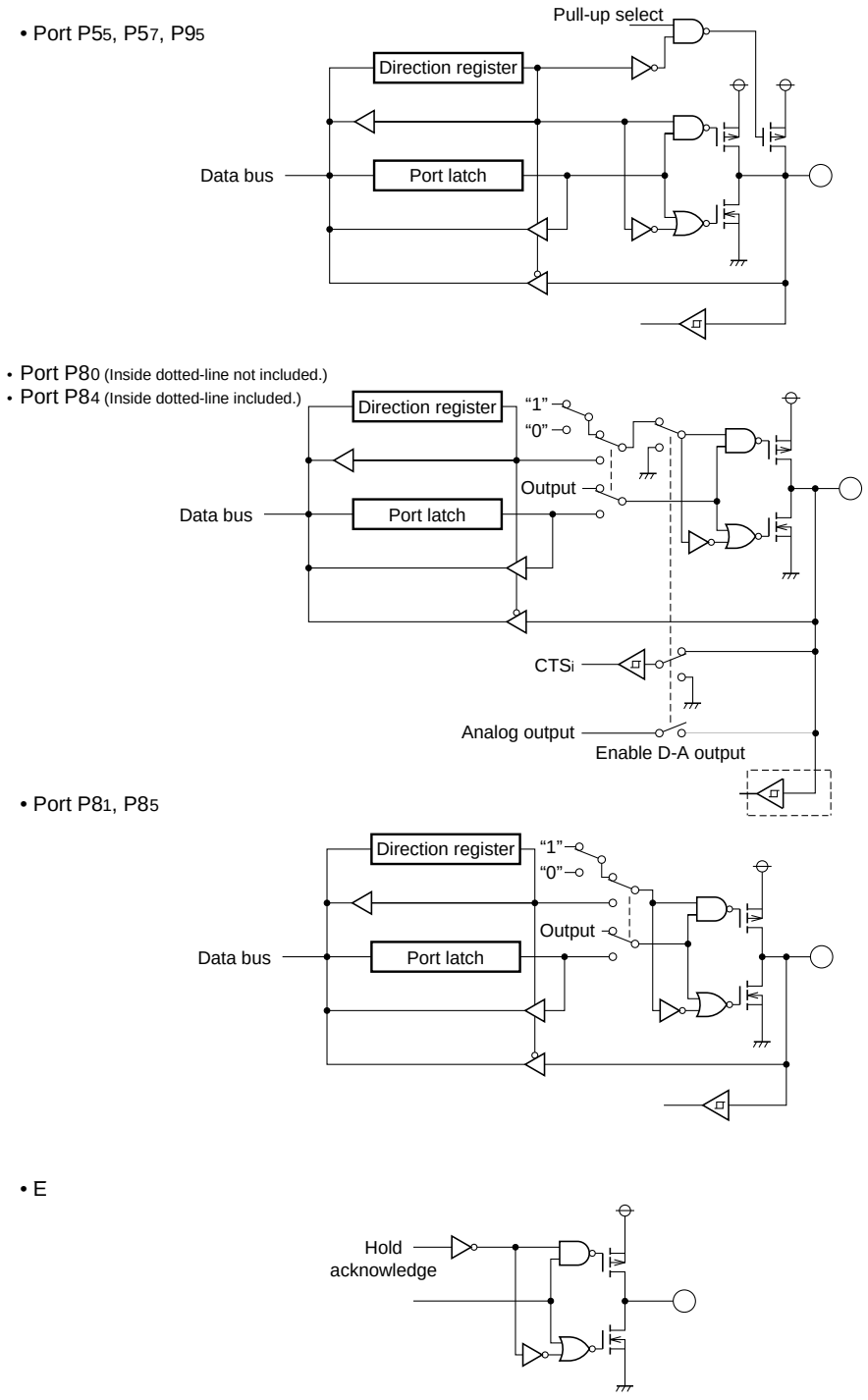


Fig. 79 Block diagram for ports P0 to P11 in single-chip mode and $\overline{E}$ output (2)

CLOCK GENERATING CIRCUIT

The clock generating circuit makes basic clocks, which activate the central processing unit (CPU), bus interface unit (BIU) and internal peripheral devices, of an oscillation circuit output. Figure 82 shows the block diagram of the clock generating circuit.

The clock source ϕ_1 to activate internal peripheral devices, the clock source ϕ_{BIU} to activate the bus interface unit and the clock source ϕ_{CPU} to activate the CPU are made of an clock input to the XIN pin. When bit 6 (clock source select bit) of processor mode register 1 is "0", the clock which is obtained by dividing an input clock to the XIN pin by 2 becomes the clock source ϕ_1. When bit 6 is "1", the clock which is an input clock to the XIN pin becomes the clock source ϕ_1 as it is. When bit 2 (clock source for peripheral devices select bit) is "0", the clock source ϕ_1 which is more divided by 2 becomes the standard clock for peripheral devices. When bit 2 is "1", the clock source ϕ_1 becomes the standard clock for peripheral devices as it is.

The standard clock is more divided with the division circuit shown in Figure 82 and the clocks having all kinds of frequencies are made. Each internal peripheral device can select one of 4 clocks, Pf_2, Pf_{16}, Pf_{64} and Pf_{512}, and use it.

Pf_2 means f(XIN), which is an oscillation circuit's frequency, divided by 2 when the clock source for peripheral devices select bit is "1". It means f(XIN) divided by 4 when that bit is "0". In the case of $\phi_1 > 12.5$ MHz, fix the bit to "0".

Figure 80 shows a circuit example using a ceramic (or quartz crystal) resonator. Use the manufactures' recommended values for constants such as capacitance which differs for each resonator.

Figure 81 shows a circuit example inputting clocks externally. When inputting clocks externally, setting bit 1 (clock external input select bit) of particular function select register 0 (in Figure 83) to "1" makes operation of the clock oscillation circuit stop, that is, the XOUT output stays at "H", and the current consumption reduce.

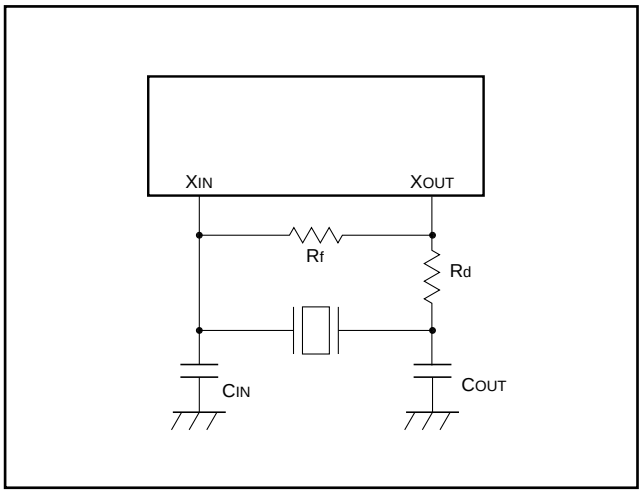


Fig. 80 Circuit example using a ceramic (or quartz crystal) resonator

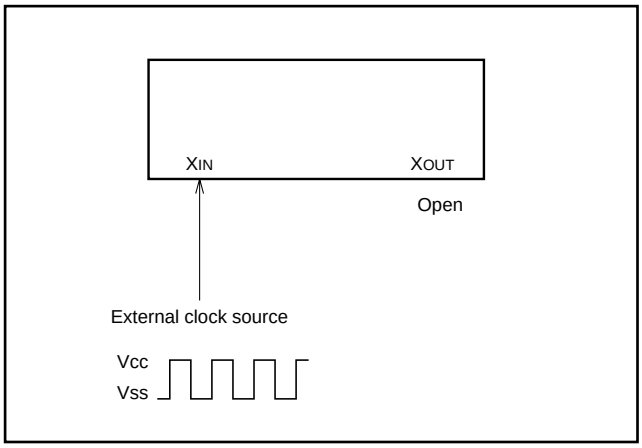


Fig. 81 Circuit example inputting clocks externally

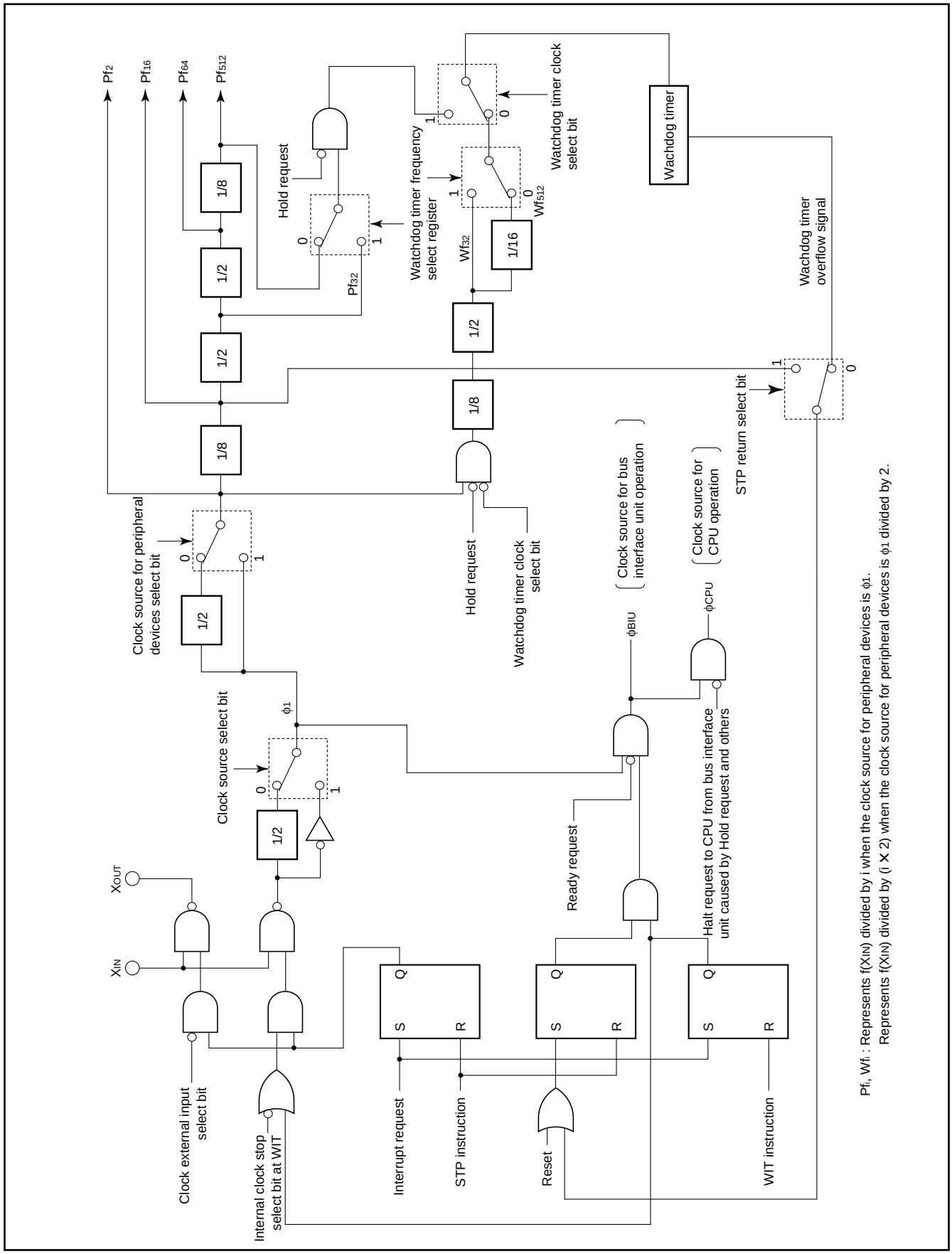
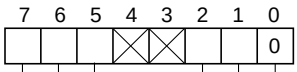


Fig. 82 Clock generating circuit block diagram



Particular function select register 0 Address $6C_{16}$

This bit must be fixed to "0."

External clock input select bit **(Notes 1, 2)**
0 : Actuated oscillation circuit; connecting resonator
1 : Stopped oscillation circuit; inputting externaly genarated clock

Memory allocation select bit **(Note 2)**
0 : ROM 60 Kbytes, RAM 2048 Bytes
(ROM : 001000_{16} to $00FFFF_{16}$, RAM : 000080_{16} to $00087F_{16}$)
1 : ROM 56 Kbytes, RAM 2048 Bytes
(ROM:002000_{16} to $00FFFF_{16}$, RAM:000080_{16} to $00087F_{16}$)

Standby state select bit 0 **(Notes 1, 3)**
; in execution of WIT or STP instruction in memory expansion or microprocessor mode
0 : External bus for P0 to P3, P10, P11
1 : Port Input/Output for P0 to P3, P10, P11

Standby state select bit 1 **(Notes 1, 4)**
; in execution of WIT or STP instruction
0 : "H" or "L" output for pins $\overline{E}/\overline{RD}$, $\overline{WR}$
1 : "H" output for pins $\overline{E}/\overline{RD}$, $\overline{WR}$

STP rerun select bit
0 : Wachdog timer is used when returning from Stop mode.
1 : Wachdog timer is not used when returning from Stop mode ; the maicrocomputer returns at once.

Notes 1 : After the expansion function select bit (bit 5 of particular function select register 1; Figure 62) is "1", bits 1, 5 and 6 can be written and changed.
2 : To set bits 1 and 2, continuous-twice-write operations to address $6C_{16}$ are required.
3 : When BYTE = "H" (8-bit external bus width), P11 becomes an input/output port independent of bit 5's contents.
4 : When the signal output disable select bit = "1" and bit 5 = "1", the $\overline{E}/\overline{RD}$ pin outputs "L" independent of bit 6's contents in execution of WIT or STP instruction.

Fig. 83 Particular function select register 0 bit configulation

STANDBY FUNCTION

The WIT and the STP instructions make the microcomputer standby state.

Table 7 shows the relation between standby state and each block's operation.

The WIT/STP state is terminated by interrupt acceptance or reset. Accordingly, it is necessary to prepare the state in which any interrupt can be accepted before the WIT/STP instruction is executed.

WIT instruction

When the WIT instruction is executed with the internal clock stop select bit at WIT (bit 2 of particular function select register 1; Figure 62) = "0", the clock sources ϕ BIU and ϕ CPU are stopped at "L", however, the oscillation circuit, the clock source ϕ 1, and the divided clocks Pf2 to Pf512, Wf32, Wf512 are not stopped. Accordingly, although the CPU and bus interface unit stop operation, internal peripheral devices which use these divided clocks can operate even at WIT state.

Otherwise, when the WIT instruction is executed with the internal clock stop select bit at WIT = "1", the oscillation circuit is not stopped, however, the clock source ϕ 1, divided clocks, and the clock sources ϕ BIU and ϕ CPU are stopped. Accordingly, in this case, all of the internal peripheral devices and the watchdog timer which use divided clocks Pf2 to Pf512, Wf32, and Wf512 are stopped.

When internal peripheral devices are not used in WIT state, the latter state (internal clock stop select bit at WIT = "1") is more effective to reduce current consumption.

Make sure to set the internal clock stop select bit at WIT to "1" immediately before the WIT instruction execution and clear the bit to "0" immediately after the WIT state is terminated.

The WIT state is terminated when an interrupt request is accepted, and the internal clock ϕ operation is restarted. Interrupt processing can immediately be executed because oscillation circuit's operation is not stopped during WIT state.

STP instruction

When the STP instruction is executed, the oscillation circuit is stopped and the clock sources ϕ 1, ϕ BIU and ϕ CPU are at "L". Furthermore, "FFF16" is automatically set into the watchdog timer, and its clock source is forced to connect with Wf32 when the watchdog timer clock select bit = "0", or Pf32 when the bit = "1". This connection is cut off when the most significant bit of the watchdog timer becomes "0" or the microcomputer is reset, and the clock source is connected with the input depending on the contents of the watchdog timer frequency select register and the watchdog timer clock select bit. In STP state, all of the internal peripheral devices and the watchdog timer which use divided clocks Pf2 to Pf512, Wf32, and Wf512 are stopped.

The STP state is terminated by reset or interrupt request acceptance, and then oscillation is restarted. At the same time, supply of the clock source ϕ 1 and divided clocks Pf2 to Pf512, Wf32 and Wf512 is restarted.

In that condition, when the STP return select bit (bit 7 of particular function select register 0) is "0", the clock sources ϕ BIU and ϕ CPU stop at "L" until the most significant bit of the watchdog timer decremented with divided clock Pf32 or Wf32 becomes "0". However, supply of the clock sources ϕ BIU and ϕ CPU is restarted immediately after the oscillation restarts by reset. Accordingly, in this case, wait for enough time to stabilize the oscillation before the reset input of "H".

Otherwise in that condition, when the STP return select bit is "1", supply of the clock sources ϕ BIU and ϕ CPU is restarted at the timing of the divided clock Pf16's "H" to "L" after the oscillation restarts. This function makes it possible to immediately return from STP state when the clock supply input to the XIN from the external is stabilized. Even though clocks are input from the external, make sure to clear the STP return select bit to "0" if the external clock is unstable for a short time when returning from STP state

Table 7 Relation between standby state and each block's operation.

Instruction	Internal clock stop bit at WIT	Operation at WIT/STP state					
		Oscillation circuit	ϕ 1	Pf2 to Pf512	Wf2, Wf512	ϕ BIU, ϕ CPU	Internal peripheral devices using Pf2 to Pf512, Wf32, Wf512
WIT	"0"	Operating **(Note 1)**	Operating	Operating	Operating **(Note 2)**	Stopped ("L")	Operation enabled (Watchdog timer operating)
	"1"	Operating **(Note 1)**	Stopped ("L")	Stopped ("H")	Stopped ("H")	Stopped ("L")	Operation disabled (Watchdog timer stopped)
STP	—	Stopped	Stopped ("L")	Stopped ("H")	Stopped ("H")	Stopped ("L")	Operation disabled (Watchdog timer stopped)

Notes 1 : When the clock external input select bit is "1", the clock oscillation circuit stops. An external clock can be input.
2 : When the watchdog timer clock select bit is "1", Wf32 and Wf512 stop. The watchdog timer operates with Pf32 or Pf512.

Bus cycle in WIT/STP

When the WIT/STP instruction is executed with the standby state select bit 1 (bit 6 of particular function select register 0) = "0", the clock sources ϕ BIU and ϕ CPU or oscillation stop without waiting for completion of the bus cycle being executed. Accordingly, the microcomputer may enter WIT/STP state during bus access in which output of pins $\overline{E}$, $\overline{RD}$ and $\overline{WR}$ is "L".

Otherwise, when the WIT/STP instruction is executed with the standby state select bit 1 = "1", the clock sources ϕ BIU and ϕ CPU or oscillation stop after completion of read or write in the bus access cycle being executed. Consequently, in WIT/STP state, the bus becomes the nonaccess state in which output of pins $\overline{E}$, $\overline{RD}$ and $\overline{WR}$ is "H".

Bus state in WIT/STP

Normally, pins for the address output, data input/output and bus control signal output in the memory expansion/microprocessor mode (ports P0 to P3, P10, P11 in single-chip mode; refer to section on Processor mode) retain the state as external bus pins when the clock sources ϕ BIU and ϕ CPU stop in WIT/STP state.

However, when the WIT/STP instruction is executed with the standby state select bit 0 (bit 5 of particular function select register 0) = "1", those pins function depending on the contents of each port direction register and port latch in WIT/STP state like ports in single-chip mode. That is, when setting arbitrary data to the port latch and the contents of direction register to "1", that data is output from the pin; when clearing the contents of direction register to "0", the pin becomes floating. This function makes the external bus arbitrary state in WIT/STP state. When making pins floating, take consideration with an external circuit to prevent their electric potential from becoming half level of the electric potential.

When writing to registers relevant to ports P0 to P3, P10, P11 in the memory expansion/microprocessor mode, set the standby state select bit 0 to "1" before that write. If that bit is "0", write is impossible, because addresses corresponding to registers relevant to ports P0 to P3, P10, P11, which are addresses 2_{16} to 9_{16}, 16_{16} to 19_{16}, are the external memory areas shown in Figure 86.

[Note]

Port P11 functions as an input/output port regardless of processor modes when inputting "L" to the BYTE pin.

The $\overline{RD}$ pin state can arbitrarily be selected in WIT/STP state in the memory expansion/microprocessor mode, too. Refer to the Table 8 for details.

Note that the function of arbitrary data output cannot be emulated using a debugger.

Table 8 Signal output disable select bit function (bit 4 of particular function select register 1; Figure 62)

Processor mode	Pin	Function	
		Signal output disable select bit = "0"	Signal output disable select bit = "1"
Single-chip mode	$\overline{E}$/$\overline{RD}$	Outputs enable signal $\overline{E}$.	Outputs "L".
Memory expansion mode Microprocessor mode	$\overline{RD}$, $\overline{WR}$	Outputs $\overline{RD}$/$\overline{WR}$ when accessing internal/ external memory area.	Outputs $\overline{RD}$/$\overline{WR}$ when accessing external memory area only.
	$\overline{RD}$	Outputs "H" or "L" after executing WIT/STP instruction Outputs "H" when standby state select bit 1 is "1".	Outputs "H" or "L" after executing WIT/STP instruction. Outputs "L" when standby state select bit 0 is "1". Outputs "H" when standby state select bit 1 is "1" and standby state select bit 0 is "0."
	ALE	Outputs ALE.	Outputs "L" when multiplex bus select bit = "0". Outputs ALE when multiplex bus select bit = "1".
Microprocessor mode	ϕ 1	Outputs clock ϕ 1 regardless of ϕ 1 output select bit.	Outputs contents of port $P4_2$ latch; necessary to set its direction register bit to "1".

Note : All functions of signal output disable select bit cannot be debugged using an debugger.

PROCESSOR MODE

Bits 0 and 1 of processor mode register 0 (address $5E_{16}$) shown in Figure 84 are used to select any mode of the single-chip mode, the memory expansion mode and the microprocessor mode.

Ports P0 to P3, P10, P11 and a part of port P4 are used as I/O pins of address, data, and control signals in the modes except the single-chip mode.

Figure 85 shows the functions of ports P0 to P4, P10 and P11 in each mode.

The external memory area depends on the mode. Figure 86 shows the memory map for each mode. Refer to Figure 1 for the addresses of RAM and ROM in the single-chip mode. The external memory area can be accessed in the modes except the single-chip mode. The access to the external memory is affected by the BYTE pin

• BYTE pin

When accessing the external memory, the level of the BYTE pin is used to determine whether to use the data bus as 8-bit width or 16-bit width.

The data bus has a width of 16 bits when the level of the BYTE pin is "L", and ports P10 and P11 become the data I/O pins.

The data bus has a width of 8 bits when level of the BYTE pin is "H", and port P10 becomes data I/O pins. Port P11 functions then as an input/output port similarly in the single-chip mode.

When accessing the internal memory, the data bus always has a width of 16 bits regardless of the BYTE pin level.

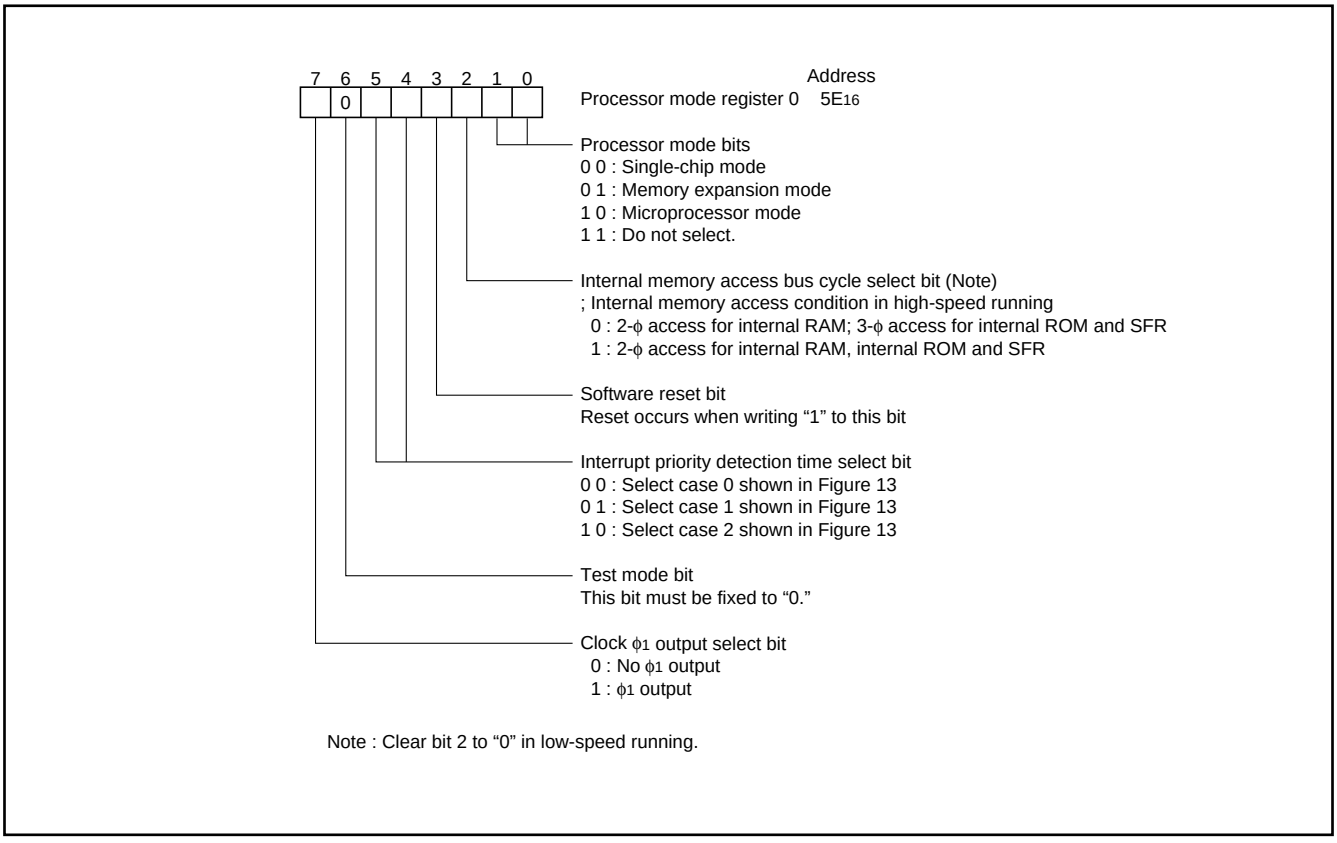


Fig. 84 Processor mode register 0 bit configuration

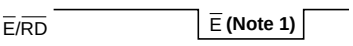
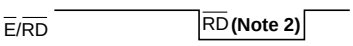
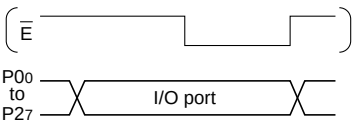
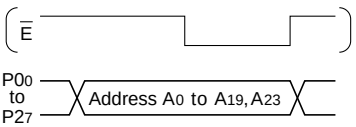
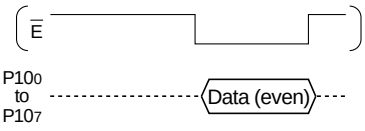
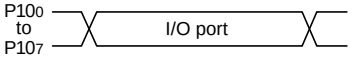
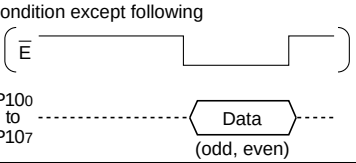
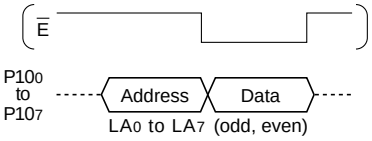
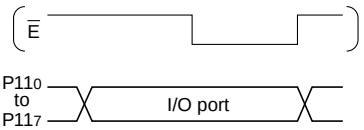
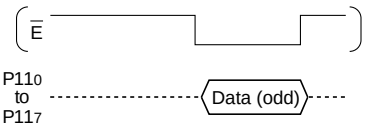
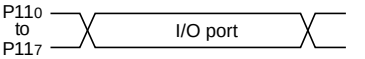
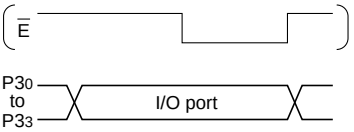
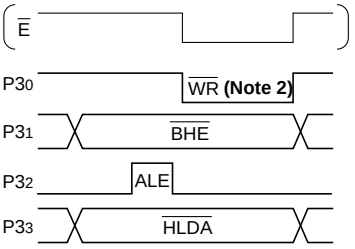
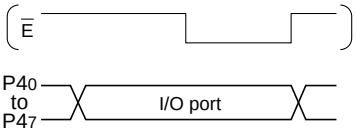
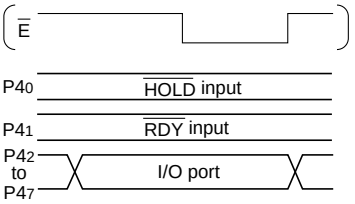
Port \ PM1 / PM0	PM1 = 0, PM0 = 0	PM1 = 0, PM0 = 1	PM1 = 1, PM0 = 0
	Single-chip mode	Memory expansion mode **(Note 1)**	Microprocessor mode **(Note 1)**
$\overline{E}/\overline{RD}$			Same as left
Port P0 Port P1 Port P2			Same as left
Port P10, BYTE = "L"			Same as left
Port P10, BYTE = "H"		• Condition except following 	Same as left
		• When multiplex bus select bit is "1" and accessing $\overline{CS4}$ area 	Same as left
Port P11, BYTE = "L"			Same as left
Port P11, BYTE = "H"			Same as left
Port P3			Same as left
Port P4	 ∗ Clock ϕ_1 is output from P42 when bit 7 of processor mode register 0 is "1".	 ∗ Clock ϕ_1 is output from P42 when bit 7 of processor mode register 0 is "1."	Clock ϕ_1 is output from P42 regardless of bit 7 of processor mode register 0 ; others are the same as left (Note 2)

Fig. 85 Processor modes and ports P0 to P4, P10 and P11

Notes 1 : $\overline{E}$ signal is not output in the memory expansion and microprocessor modes.
2 : The signal output stop disable bit (bit 4 of particular function select register 0) can stop $\overline{E}$ output in the single-chip mode and ϕ_1 output in the microprocessor mode. Similarly, when accessing the internal memory in the memory expansion and microprocessor modes, $\overline{RD}$ and $\overline{WR}$ output can be fixed to "H". Refer to Table 8 for details.

Processor modes are explained bellow.

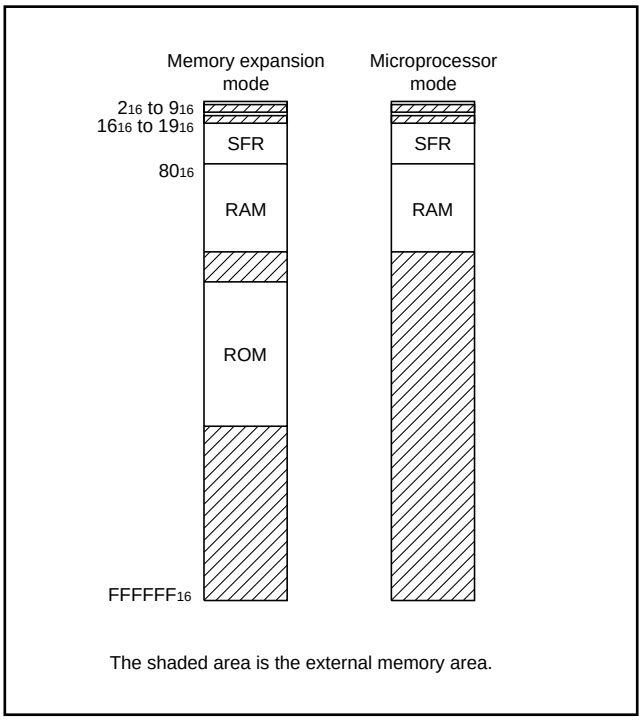


Fig. 86 External memory area for each mode

(1) Single-chip mode [00]

The microcomputer enters the single-chip mode by connecting the CNVss pin to Vss and starting from reset. Ports P0 to P4, P10 and P11 all function as normal I/O ports. Port $P4_2$ can output clock source ϕ_1 by setting bit 7 of the processor mode register 0 to "1".

In this mode, enable signal $\overline{E}$ is output from pin $\overline{E}/\overline{RD}$. Signal $\overline{E}$ output can be stopped by setting the signal output disable select bit (bit 4 of particular function select register 1) to "1", and it is possible to switch the output to "L" level. Table 8 shows the function of the signal output disable select bit's function.

(2) Memory expansion mode [01]

The microcomputer enters the memory expansion mode by setting the processor mode bits to "01" after connecting the CNVss pin to Vss and starting from reset.

Pin $\overline{E}/\overline{RD}$ becomes the $\overline{RD}$ output pin. $\overline{RD}$ is an read signal, and read is performed during it is "L" level. When the internal memory area is read, the $\overline{RD}$ output can be fixed to "H" by setting the signal output disable select bit to "1".

Ports P0, P1 and P2 become the output pins of addresses A_0 to A_{19} and A_{23}, and their I/O port function are lost.

Port P10 becomes I/O pins of data D_0 to D_7 and loses its I/O port function. When the BYTE pin's level is "L", those pins function as data I/O pins at an even address. When the level is "H", those pins function as data I/O pins at even and odd addresses. However, if an internal memory area is read, external data is not input

When the BYTE pin's level is "H" and the multiplex bus select bit (bit 5 of chip select area register; Figure 88) is "1", port P10 functions as follows during the bus cycle in which the external memory area corresponding to the chip select $\overline{CS_4}$ are accessed:

- Output pins of addresses LA_0 to LA_7, same as low-order addresses A_0 to A_7, during "H" of $\overline{RD}$ or $\overline{WR}$.
- Data input/output pins at even and odd addresses during "L" of $\overline{RD}$ or $\overline{WR}$.

That is, it functions as a multiplex bus during that bus cycle.

Port P11 has two functions depending on the level of the BYTE pin. When the BYTE pin level is "L", those pins function as data D_8 to D_{15} I/O pins at an odd address. The I/O port function is lost. However, if an internal memory area is read, external data is not input. When the BYTE pin level is "H", port P11 functions as a programmable port P11 similarly in the single-chip mode.

Ports $P3_0$, $P3_1$, $P3_2$, and $P3_3$ become $\overline{WR}$, $\overline{BHE}$, ALE, and $\overline{HLDA}$ output pins respectively and lose their I/O port functions.

$\overline{WR}$ is a write signal which indicates a write when it is "L".

$\overline{BHE}$ is a byte-high-enable signal which indicates that an odd address is accessed when it is "L".

Therefore, two bytes at even and odd addresses are accessed simultaneously when address A_0 is "L" and $\overline{BHE}$ is "L".

ALE is an address-latch-enable signal. The latch is open while ALE is "H", so that the address signal passes through; the address is held while ALE is "L".

$\overline{HLDA}$ is a hold-acknowledge signal and is used to indicate to the external that the microcomputer accepts $\overline{HOLD}$ input and enters Hold state.

Ports $P4_0$ and $P4_1$ become $\overline{HOLD}$ and $\overline{RDY}$ input pins, respectively, and their I/O port function are lost.

$\overline{HOLD}$ is a hold-request signal. It is an input signal used to make the microcomputer enter Hold state. $\overline{HOLD}$ input is accepted when the ϕ_{BIU} has fallen from "H" to "L" level while the bus is not used. In Hold state, ϕ_{CPU} stops at "L". A_0 to A_{19}, A_{23}, D_0 to D_7, D_8 to D_{15} (at BYTE = "L"), $\overline{RD}$, $\overline{WR}$ and $\overline{BHE}$ become floating then. These pins become floating one cycle of ϕ_{BIU} later than $\overline{HLDA}$ signal becomes "L" level. When terminating Hold state, these pins are terminated from floating state one cycle of ϕ_{BIU} later than $\overline{HLDA}$ signal becomes "H" level.

$\overline{RDY}$ is a ready signal. When this signal goes "L", ϕ_{CPU} and ϕ_{BIU} stop at "L". $\overline{RDY}$ is used when a slow external memory is connected and others.

Port $P4_2$ becomes a normal I/O port when bit 7 of the processor mode register 0 is "0" and becomes the clock ϕ_1 output pin when bit 7 is "1". The ϕ_1 output is independent of $\overline{RDY}$ and does not stop even when ϕ_{CPU} and ϕ_{BIU} stop owing to "L" input to the $\overline{RDY}$ pin.

(3) Microprocessor mode [10]

The microcomputer enters the microprocessor mode by connecting the CNVss pin to Vcc and starting from reset. It is possible to enter this mode by programming the processor mode bits to "10" after connecting the CNVss pin to Vss and starting from reset. This mode is the same as the memory expansion mode except the following: the internal ROM is disabled and an external memory is required, and clock ϕ_1 is always output from port P4$_2$ independent of bit 7 of the processor mode register 0.

As shown in Table 8, ϕ_1 output can also be stopped by setting the signal output disable select bit to "1". In this case, write "1" to the port P4$_2$ direction register bit.

Table 9 shows the relationship between the CNVss pin's input level and the processor modes.

Additionally, addresses A$_{20}$ to A$_{22}$ or chip select signals $\overline{CS_0}$ to $\overline{CS_4}$ can be output from port P9 regardless of processor modes. For details, refer to the following sections: output function of chip select signal and address output function.

Table 9. Relationship between CNV$_{SS}$ pin's input levels and processor modes

CNV$_{SS}$	Mode	Description
V$_{SS}$	• Single-chip • Memory expansion • Microprocessor	Single-chip mode upon starting after reset. Each mode can be selected by changing the processor mode bits by software.
V$_{CC}$	• Microprocessor	Microprocessor mode upon starting after reset.

OUTPUT FUNCTION OF CHIP SELECT SIGNAL

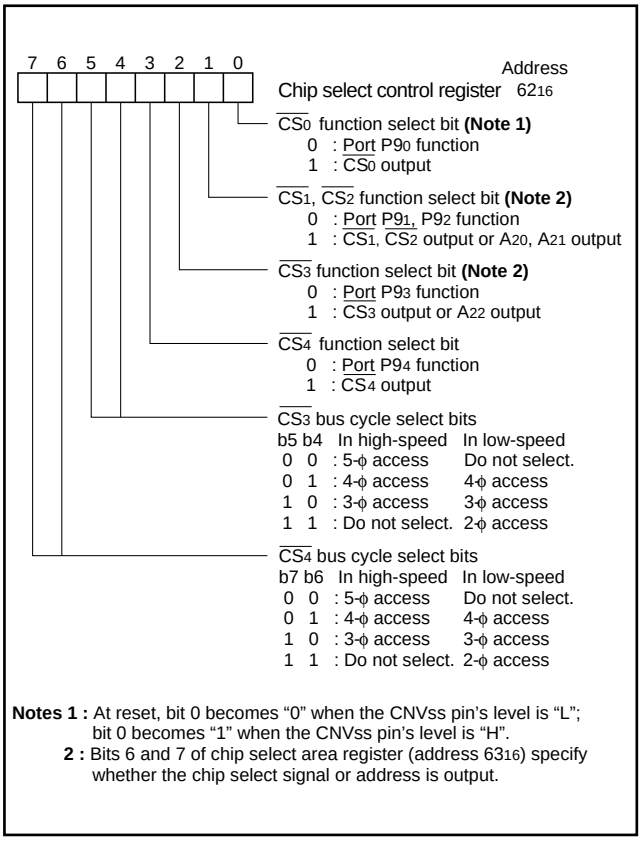
Ports P90 to P94 can output the chip select signals $\overline{CS_0}$ to $\overline{CS_4}$ according to the contents of chip select control register and chip select area register. Bits 0 to 3 of chip select control register select either chip select output (or addresses A20 to A22 output) or port function. Additionally, bits 0 to 2 of chip select area register select the area intended for each chip select signal.

Figure 87 shows the bit configuration of chip select control register and Figure 88 shows that of chip select area register. Figure 89 shows the chip select areas.

The bus cycle of $\overline{CS_3}$ and $\overline{CS_4}$ can be selected with bits 4 to 7 of chip select control register. That selection is valid regardless of the bus cycle select bits of processor mode register 1. Additionally, that bus cycle selection of $\overline{CS_3}$ and $\overline{CS_4}$ is valid when selecting port function with the $\overline{CS_3}$ and $\overline{CS_4}$ function select bits.

When accessing addresses in which the chip select area specified by bits 0 to 2 of chip select area register and the internal memory area overlap one another, chip select signals are not output. In this case, its bus cycle is the cycle of internal memory area access.

It is possible to make the chip select output floating during Hold state. That is realized by clearing the corresponding bit of port P9 direction register (address 15_{16}) to "0" and bits 0 to 2 of waveform output mode register (address $1A_{16}$) to "000". The timing of Hold start and termination is the same as that of addresses A0 to A19. (Refer to section on processor mode.)



Notes 1 : At reset, bit 0 becomes "0" when the CNVss pin's level is "L"; bit 0 becomes "1" when the CNVss pin's level is "H".
2 : Bits 6 and 7 of chip select area register (address 63_{16}) specify whether the chip select signal or address is output.

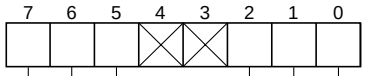
Fig. 87 Chip select control register bit configuration

ADDRESS OUTPUT FUNCTION

Port P91 to P93 can output the high-order addresses (A20 to A22) according to bits 1 and 2 of chip select control register, and bits 6 and 7 of chip select area register.

About signal pairs of A20 and $\overline{CS_1}$, A21 and $\overline{CS_2}$, and A22 and $\overline{CS_3}$, only one signal can be output. It is because chip select signals $\overline{CS_1}$ to $\overline{CS_3}$ output are common to ports P91 to P93 and addresses A20 to A22 output.

It is possible to make the address output floating during Hold state. That is realized by clearing the corresponding bit of port P9 direction register (address 15_{16}) to "0" and bits 0 to 2 of waveform output mode register (address $1A_{16}$) to "000". The timing of Hold start and termination is the same as that of addresses A0 to A19. (Refer to section on processor mode.)



Address
Chip select area register 63₁₆

Chip select area switch bits (Notes 1, 2)

000 : $\overline{CS_0}$ 001000₁₆ to 02FFFF₁₆ (188 Kbytes)
$\overline{CS_1}$ 030000₁₆ to 04FFFF₁₆ (128 Kbytes)
$\overline{CS_2}$ 050000₁₆ to 06FFFF₁₆ (128 Kbytes)
$\overline{CS_3}$ 000880₁₆ to 000DFF₁₆ (1408 bytes)
$\overline{CS_4}$ 000E00₁₆ to 000FFF₁₆ (512 bytes)

001 : $\overline{CS_0}$ 008000₁₆ to 02FFFF₁₆ (160 Kbytes)
$\overline{CS_1}$ 030000₁₆ to 04FFFF₁₆ (128 Kbytes)
$\overline{CS_2}$ 050000₁₆ to 06FFFF₁₆ (128 Kbytes)
$\overline{CS_3}$ 000880₁₆ to 000FFF₁₆ (1920 bytes)
$\overline{CS_4}$ 001000₁₆ to 007FFF₁₆ (28 Kbytes)

010 : $\overline{CS_0}$ 001000₁₆ to 02FFFF₁₆ (188 Kbytes)
$\overline{CS_1}$ 030000₁₆ to 04FFFF₁₆ (128 Kbytes)
$\overline{CS_2}$ 050000₁₆ to 06FFFF₁₆ (128 Kbytes)
$\overline{CS_3}$ 070000₁₆ to 077FFF₁₆ (32 Kbytes)
$\overline{CS_4}$ 078000₁₆ to 07FFFF₁₆ (32 Kbytes)

011 : $\overline{CS_0}$ 008000₁₆ to 02FFFF₁₆ (160 Kbytes)
$\overline{CS_1}$ 030000₁₆ to 04FFFF₁₆ (128 Kbytes)
$\overline{CS_2}$ 050000₁₆ to 06FFFF₁₆ (128 Kbytes)
$\overline{CS_3}$ 070000₁₆ to 077FFF₁₆ (32 Kbytes)
$\overline{CS_4}$ 078000₁₆ to 07FFFF₁₆ (32 Kbytes)

100 : $\overline{CS_0}$ 000880₁₆ to 02FFFF₁₆ (190 Kbytes)
$\overline{CS_1}$ 030000₁₆ to 04FFFF₁₆ (128 Kbytes)
$\overline{CS_2}$ 050000₁₆ to 06FFFF₁₆ (128 Kbytes)
$\overline{CS_3}$ 080000₁₆ to 3FFFFF₁₆ (3.5 Mbytes)
$\overline{CS_4}$ 070000₁₆ to 07FFFF₁₆ (64 Kbytes)

101 : $\overline{CS_0}$ 000880₁₆ to 02FFFF₁₆ (190 Kbytes)
$\overline{CS_1}$ 030000₁₆ to 04FFFF₁₆ (128 Kbytes)
$\overline{CS_2}$ 050000₁₆ to 06FFFF₁₆ (128 Kbytes)
$\overline{CS_3}$ 070000₁₆ to 07FFFF₁₆ (64 Kbytes)
$\overline{CS_4}$ 080000₁₆ to 7FFFFF₁₆ (7.5 Mbytes)

110 : $\overline{CS_0}$ 000880₁₆ to 06FFFF₁₆ (446 Kbytes)
$\overline{CS_1}$, $\overline{CS_2}$ Not available
$\overline{CS_3}$ 080000₁₆ to 3FFFFF₁₆ (3.5 Mbytes)
$\overline{CS_4}$ 070000₁₆ to 07FFFF₁₆ (64 Kbytes)

111 : $\overline{CS_0}$ 000880₁₆ to 06FFFF₁₆ (446 Kbytes)
$\overline{CS_1}$, $\overline{CS_2}$ Not available
$\overline{CS_3}$ 070000₁₆ to 07FFFF₁₆ (64 Kbytes)
$\overline{CS_4}$ 080000₁₆ to 7FFFFF₁₆ (7.5 Mbytes)

Multiplex bus select bit **(Note 1)**
0 : D_0 to D_7 input/output (separate bus)
1 : When BYTE pin input is "H" and accessing $\overline{CS_4}$ area
LA_0/D_0 to LA_7/D_7 input/output (multiplex bus)
In condition except above
D_0 to D_7 input/output (separate bus)

Expansion address output select bits **(Notes 1, 3)**
0 0 : $P9_1$.... $\overline{CS_1}$ output, $P9_2$...$\overline{CS_2}$ output, $P9_3$...$\overline{CS_3}$ output
0 1 : $P9_1$.... A_{20} output, $P9_2$...$\overline{CS_2}$ output, $P9_3$...$\overline{CS_3}$ output
1 0 : $P9_1$.... A_{20} output, $P9_2$...A_{21} output, $P9_3$...$\overline{CS_3}$ output
1 1 : $P9_1$.... A_{20} output, $P9_2$...A_{21} output, $P9_3$...A_{22} output

Notes 1 : When the expansion function select bit (bit 5 of particular function select register 1; Figure 62) is "1", bits 2, 5, 6 and 7 can be written and changed.
2 : When accessing the internal memory area, $\overline{CS_i}$ is not output. When only accessing the external area, $\overline{CS_i}$ output is valid.
3 : Select function of bits 6 and 7 is valid when both the $\overline{CS_1}$, $\overline{CS_2}$ function select bit and the $\overline{CS_3}$ function select bit (chip select control register) are "1".

Fig. 88 Chip select area register bit configuration

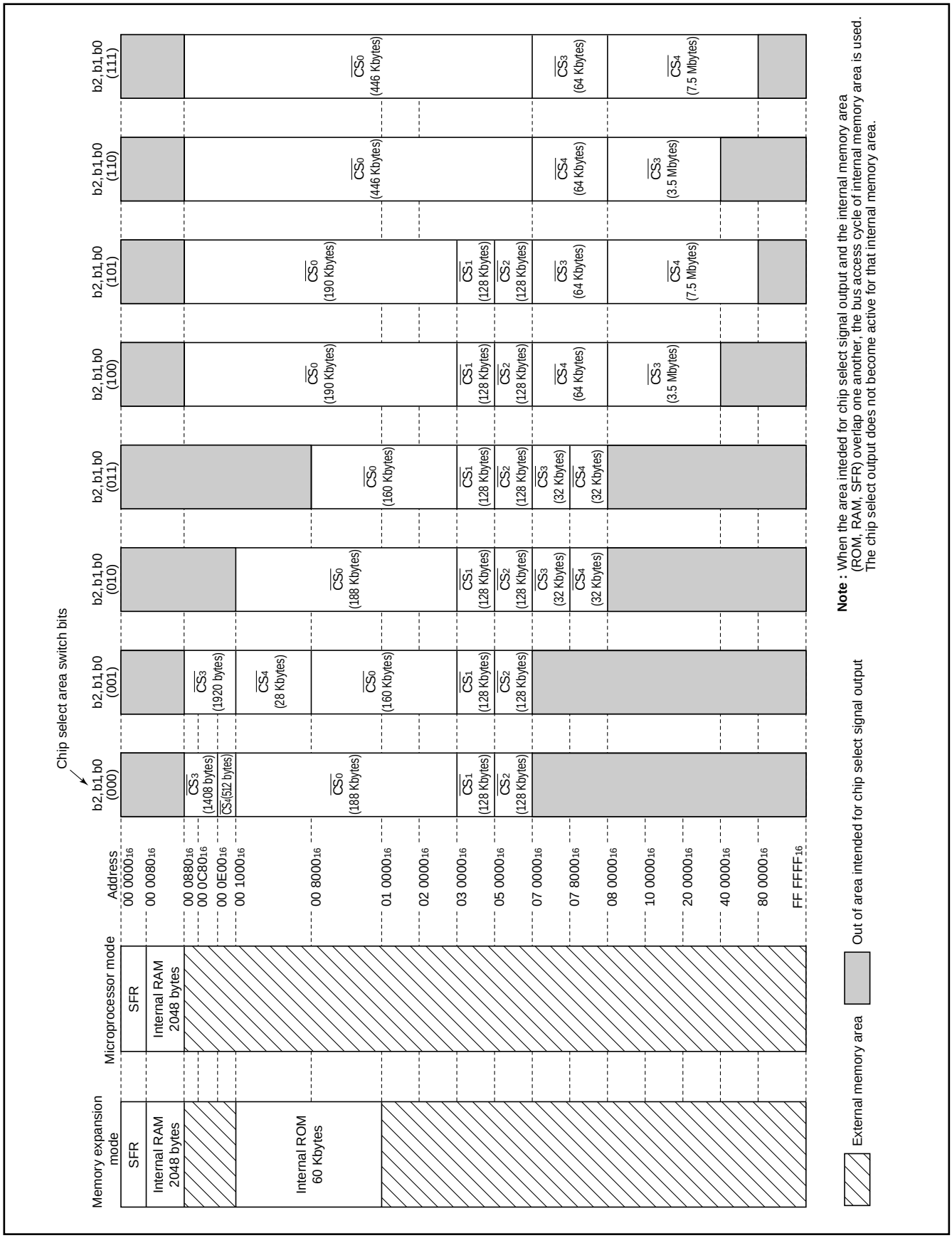


Fig. 89 Chip select areas

MEMORY MODIFICATION FUNCTION

The M37754M8C-XXXGP's internal memory size and address area can be modified by set of bit 2 (memory allocation select bit) of the particular function select register 0. Figure 90 shows the memory allocation when modifying the internal memory area.

When ordering a mask ROM, Mitsubishi Electric corp. produces the mask ROM using the data within 60 Kbytes (between addresses 001000_{16} to $00FFFF_{16}$). It is regardless of the selected ROM size (refer to MASK ROM ORDER CONFIRMATION FORM). Therefore, on the EPROM tendered for ordering a mask ROM, program data "FF_{16}" to addresses which correspond to the area out of the selected ROM area.

Additionally, address $00FFFF_{16}$ of the microcomputer corresponds to the lowest address of the tendered EPROM.

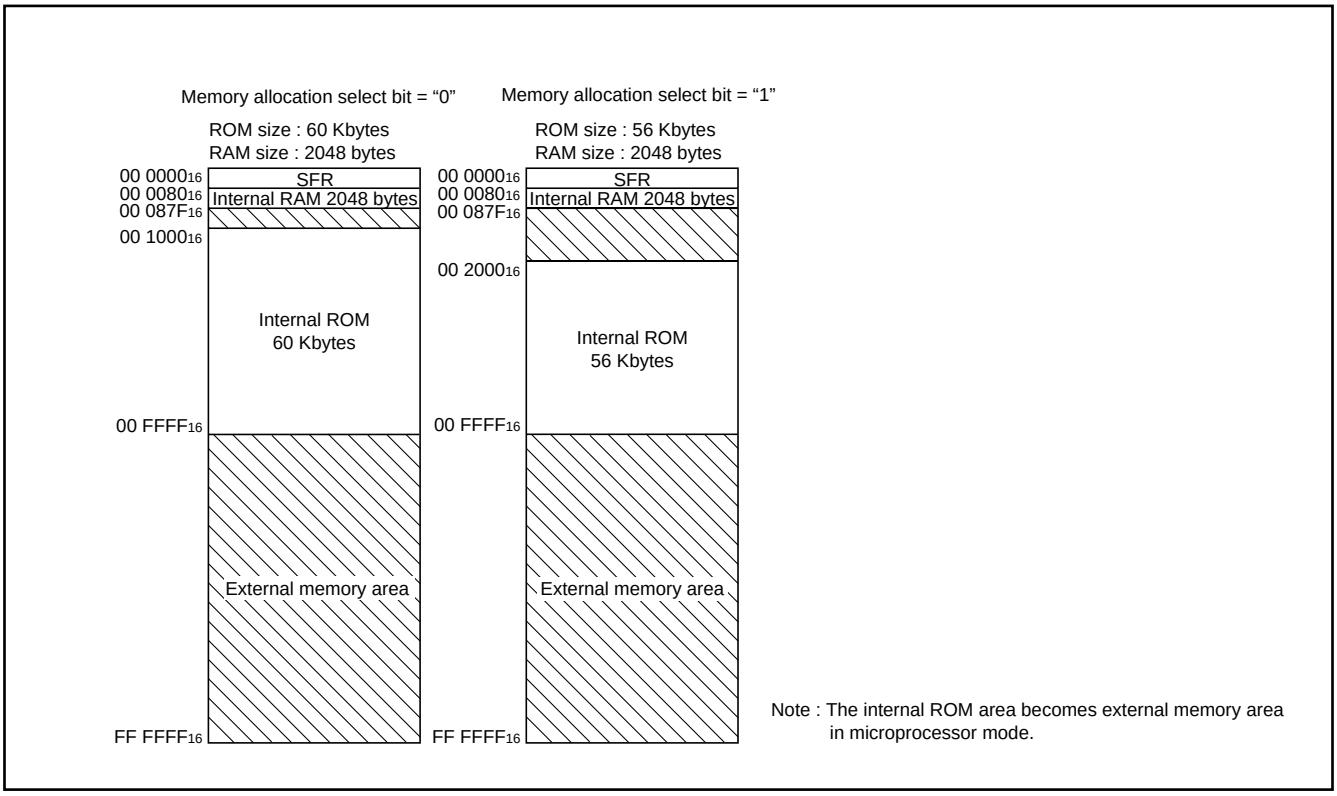


Fig. 90 Memory allocation when modifying internal memory area with memory allocation select bit

ADDRESSING MODES AND INSTRUCTION SET

The M37754M8C-XXXGP and M37754M8C-XXXHP have 29 powerful addressing modes; 1 addressing mode is added to the basis of the 7700 series. Refer to the “7751 Series Software Manual” for the details.

INSTRUCTION SET

The M37754M8C-XXXGP and M37754M8C-XXXHP have the extended instruction set; 6 instructions are added to the instruction set of 7700 series. The object code of this extended instruction set is upwards compatible to that of 7700 series instruction set.
Refer to the “7751 Series Software Manual” for the details.

SHORTENING NUMBER OF INSTRUCTION EXECUTION CYCLES

Shortening number of instruction execution cycles is realized in the M37754M8C-XXXGP and M37754M8C-XXXHP owing to modifications of the instruction execution algorithm and the CPU circuit, and others.
Refer to the “7751 Series Software Manual” about the number of instruction execution cycles.

DATA REQUIRED FOR MASK ROM ORDERING

Please send the following data for mask orders:
<M37754M8C-XXXGP>
(1) M37754M8C-XXXGP mask ROM order confirmation form
(2) 100P6S mark specification form
(3) ROM data (EPROM 3 sets)

<M37754M8C-XXXHP>
(1) M37754M8C-XXXHP mask ROM order confirmation form
(2) 100P6Q mark specification form
(3) ROM data (EPROM 3 sets)

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Ratings	Unit
V_{CC}	Power source voltage	–0.3 to 7	V
AV_{CC}	Analog power source voltage	–0.3 to 7	V
V_I	Input voltage $\overline{RESET}$, CNV_{SS}, BYTE	–0.3 to 12	V
V_I	Input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P9_0$–$P9_5$, $P10_0$–$P10_7$, $P11_0$–$P11_7$, V_{REF}, X_{IN}	–0.3 to V_{CC}+0.3	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P9_0$–$P9_5$, $P10_0$–$P10_7$, $P11_0$–$P11_7$, X_{OUT}, $\overline{E}$	–0.3 to V_{CC}+0.3	V
P_d	Power dissipation	300	mW
T_{opr}	Operating temperature	–20 to 85	°C
T_{stg}	Storage temerature	–40 to 150	°C

RECOMMENDED OPERATING CONDITIONS (V_{CC} = 5 V±10 %, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
V_{CC}	Supply voltage	4.5	5.0	5.5	V
AV_{CC}	Analog supply voltage		V_{CC}		V
V_{SS}	Supply voltage		0		V
AV_{SS}	Analog supply voltage		0		V
V_{IH}	High-level input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P9_0$–$P9_5$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	0.8 V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage $P10_0$–$P10_7$, $P11_0$–$P11_7$ (in single-chip mode)	0.8 V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage $P10_0$–$P10_7$, $P11_0$–$P11_7$ (in memory expansion mode and microprocessor mode)	0.5 V_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P9_0$–$P9_5$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	0		0.2 V_{CC}	V
V_{IL}	Low-level input voltage $P10_0$–$P10_7$, $P11_0$–$P11_7$ (in single-chip mode)	0		0.2 V_{CC}	V
V_{IL}	Low-level input voltage $P10_0$–$P10_7$, $P11_0$–$P11_7$ (in memory expansion mode and microprocessor mode)	0		0.16 V_{CC}	V
$I_{OH(peak)}$	High-level peak output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P9_0$–$P9_2$, $P9_5$, $P10_0$–$P10_7$, $P11_0$–$P11_7$			–10	mA
$I_{OH(peak)}$	$P9_3$, $P9_4$			–20	mA
$I_{OH(avg)}$	High-level average output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P9_0$–$P9_2$, $P9_5$, $P10_0$–$P10_7$, $P11_0$–$P11_7$			–5	mA
$I_{OH(avg)}$	$P9_3$, $P9_4$			–15	mA
$I_{OL(peak)}$	Low-level peak output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_4$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P9_0$, $P9_5$, $P10_0$–$P10_7$, $P11_0$–$P11_7$			10	mA
$I_{OL(peak)}$	$P5_0$–$P5_3$, $P9_1$–$P9_4$			20	mA
$I_{OL(avg)}$	Low-level average output current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_3$, $P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_4$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P9_0$, $P9_5$, $P10_0$–$P10_7$, $P11_0$–$P11_7$			5	mA
$I_{OL(avg)}$	$P5_0$–$P5_3$,$P9_1$–$P9_4$			15	mA
$f(X_{IN})$	External clock frequency input **(Note 3)** — Low-speed running			25	MHz
	External clock frequency input **(Note 3)** — High-speed running			40	MHz

Notes 1: Average output current is the averaage value of a 100 ms interval.
2: The sum of $I_{OL(peak)}$ for ports P0, P1, P2, P3, P8, P10, and P11 must be 80 mA or less, the sum of $I_{OH(peak)}$ for ports P0, P1, P2, P3, P8, P10, and P11 must be 80 mA or less, the sum of $I_{OL(peak)}$ for ports P4, P5, P6, P7, and P9 must be 110 mA or less, the sum of $I_{OH(peak)}$ for ports P4, P5, P6, P7, and P9 must be 80 mA or less.
3: When the clock source select bit is "1," $f(X_{IN})$'s maximum limit is 12.5 MHz at low-speed running and is 20 MHz at high-speed running.

ELECTRICAL CHARACTERISTICS (Vcc = 5 V, Vss = 0 V, Ta = –20 to 85 °C, f(XIN) = 40 MHz **(Note)**)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	High-level output voltage P0₀–P0₇, P1₀–P1₇, P2₀–P2₃, P2₇, P3₁, P3₃, P4₀–P4₇, P5₀–P5₇, P6₀–P6₇, P7₀–P7₇, P8₀–P8₇, P9₀–P9₂, P9₅, P10₀–P10₇, P11₀–P11₇	I_{OH} = –10 mA		3.4			V
V_{OH}	High-level output voltage P0₀–P0₇, P1₀–P1₇, P2₀–P2₃, P2₇, P3₁, P3₃, P9₀–P9₂, P10₀–P10₇, P11₀–P11₇	I_{OH} = –400 μA		4.8			V
V_{OH}	High-level output voltage E, P3₀, P3₂	I_{OH} = –10 mA		3.4			V
		I_{OH} = –400 μA		4.8			
V_{OH}	High-level output voltage P9₃, P9₄	I_{OH} = –15 mA		3.4			V
		I_{OH} = –600 μA		4.8			
V_{OL}	Low-level output voltage P0₀–P0₇, P1₀–P1₇, P2₀–P2₃, P2₇, P3₁, P3₃, P4₀–P4₇, P5₄–P5₇, P6₀–P6₇,P7₀–P7₇, P8₀–P8₇, P9₀, P9₅, P10₀–P10₇, P11₀–P11₇	I_{OL} = 10 mA				2	V
V_{OL}	Low-level output voltage P0₀–P0₇, P1₀–P1₇, P2₀–P2₃, P2₇, P3₁, P3₃, P9₀, P10₀–P10₇, P11₀–P11₇	I_{OL} = 2 mA				0.45	V
V_{OL}	Low-level output voltage E, P3₀, P3₂	I_{OL} = 10 mA				1.6	V
		I_{OL} = 2 mA				0.4	
V_{OL}	Low-level output voltage P5₀–P5₃, P9₁–P9₄	I_{OL} = 20 mA				2	V
		I_{OL} = 2 mA				0.4	
V_{T+} —V_{T-}	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, TA0IN–TA4IN, TB0IN–TB2IN, $\overline{INT0}$–$\overline{INT4}$, ADTRG, $\overline{CTS0}$, $\overline{CTS1}$, CLK0, CLK1, RxD0, RxD1			0.4		1	V
V_{T+} —V_{T-}	Hysteresis $\overline{RESET}$, $\overline{HOLD}$, $\overline{RDY}$			0.2		0.5	V
V_{T+} —V_{T-}	Hysteresis XIN			0.1		0.3	V
I_{IH}	High-level input current P0₀–P0₇, P1₀–P1₇, P2₀–P2₃, P2₇, P3₀–P3₃, P4₀–P4₇, P5₀–P5₇, P6₀–P6₇, P7₀–P7₇, P8₀–P8₇, P9₀–P9₅, $\overline{P10_0}$–$\overline{P10_7}$, P11₀–P11₇, XIN, $\overline{RESET}$, CNVSS, BYTE	V_I = 5 V				5	μA
I_{IL}	Low-level input current P0₀–P0₇, P1₀–P1₇, P2₀–P2₃, P2₇, P3₀–P3₃, P4₀–P4₇, P5₀–P5₃, P6₀–P6₇, P7₀–P7₇, P8₀–P8₇, P9₀–P9₅, $\overline{P10_0}$–$\overline{P10_7}$, P11₀–P11₇, XIN, $\overline{RESET}$, CNVSS, BYTE	V_I = 0 V				–5	μA
I_{IL}	Low-level input current P5₄–P5₇, P9₅	V_I = 0 V, No pull-up transistor				–5	μA
		V_I = 0 V, Pull-up transistor used		–0.25	–0.5	–1.0	mA
V_{RAM}	RAM hold voltage	When clock is stoped.		2			V
I_{CC}	Power supply current (target value)	Output-only pin is open and other pins are Vss during reset.	f(XIN) = 40 MHz, square waveform **(Note)**		25	50	mA
			Ta = 25 °C when clcock is stopped.			1	μA
			Ta = 85 °C when clcock is stopped.			20	

Note: f(XIN) = 20 MHz when the clock source select bit = "1."

A-D CONVERTER CHARACTERISTICS

(V_{CC} = AV_{CC} = 5 V ± 10 %, V_{SS} = AV_{SS} = 0 V, T_a = –20 to 85 °C, the clock source select bit = 0, unless otherwise noted)

Symbol	Parameter	Test conditions			Limits			Unit
					Min.	Typ.	Max.	
————	Resolution	V_{REF} = V_{CC}		A-D converter selected			10	Bits
				Comparator selected			$\frac{1}{256}V_{REF}$	V
————	Absolute accuracy	V_{REF} = V_{CC}	250 kHz ≤ ϕ_{AD} ≤ 12.5 MHz	10-bit mode			± 3	LSB
				8-bit mode			± 2	LSB
				Comparator			± 40	mV
			250 kHz ≤ ϕ_{AD} ≤ 20 MHz **(Note 1)**	8-bit mode			± 3	LSB
				Comparator			± 60	mV
R_{LADDER}	Ladder resistance	V_{REF} = V_{CC}			5		20	kΩ
t_{CONV}	Conversion time	High-speed running (f(X_{IN}) ≤ 40 MHz) **(Note 2)**	ϕ_{AD} = f(X_{IN})/4 selected	10-bit mode	5.9			μs
				8-bit mode	4.9			
				Comparator	1.4			
			ϕ_{AD} = f(X_{IN})/2 selected	8-bit mode	2.45			
				Comparator	0.7			
		Low-speed running (f(X_{IN}) ≤ 25 MHz) **(Note 2)**		10-bit mode	4.72			
				8-bit mode	3.92			
				Comparator	1.12			
V_{REF}	Reference voltage				2.7		V_{CC}	V
V_{IA}	Analog input voltage				0		V_{REF}	V

Notes 1: This is valid when the high-speed running is selected.
2: When the clock source select bit = 1, f(X_{IN}) is 20 MHz or less at the high-speed running, and f(X_{IN}) is 12.5 MHz or less at the low-speed running.

D-A CONVERTER CHARACTERISTICS

(V_{CC} = 5 V, V_{SS} = AV_{SS} = 0 V, V_{REF} = 5 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
——	Resolution				8	Bits
——	Absolute accuracy				± 1.0	%
t_{su}	Set time				3	μs
R_O	Output resistance		1	2.5	4	kΩ
I_{VREF}	Reference power supply input current	**(Note)**			3.2	mA

Note: The test conditions are as follows:
- One D-A converter is used.
- The D-A register value of the unused D-A converter is "00_{16}."
- The reference power supply input current of the ladder resistance of the A-D converter is excluded.

PERIPHERAL DEVICE INPUT/OUTPUT TIMING (V_{CC} = 5 V±10 %, V_{CC} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

✻ If the values depends on external clock frequency f(X_{IN}), formulas of the limits are shown below. Also, the values at f(X_{IN}) = 40 MHz in high-speed running and at f(X_{IN}) = 25 MHz in low-speed running are shown in ( ). At this time, the clock source select bit is "0." When the clock source select bit is "1", regard f(X_{IN}) in tables as 2·f(X_{IN}).

✻ The rise and fall time of input signal must be 100 ns or less respectively, unless otherwise noted.

Timer A input (Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TA_{iIN} input cycle time	80		ns
$t_{w(TAH)}$	TA_{iIN} input high-level pulse width	40		ns
$t_{w(TAL)}$	TA_{iIN} input low-level pulse width	40		ns

Timer A input (Gating input in timer mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TA)}$	TA_{iIN} input cycle time	f(X_{IN}) ≤ 40 MHz	$\frac{16 \times 10^9}{f(X_{IN})}$ (400)		ns
		(X_{IN}) ≤ 25 MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (320)		ns
$t_{w(TAH)}$	TA_{iIN} input high-level pulse width	f(X_{IN}) ≤ 40 MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (200)		ns
		f(X_{IN}) ≤ 25 MHz	$\frac{4 \times 10^9}{f(X_{IN})}$ (160)		ns
$t_{w(TAL)}$	TA_{iIN} input low-level pulse width	f(X_{IN}) ≤ 40 MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (200)		ns
		f(X_{IN}) ≤ 25 MHz	$\frac{4 \times 10^9}{f(X_{IN})}$ (160)		ns

Note : The TA_{iIN} input cycle time requires 4 or more cycles of count source. The TA_{iIN} input high-level pulse width and the TA_{iIN} input low-level pulse width respectively require 2 or more cycles of the count source. The limits in the table are the values when the count source is f(X_{IN})/4 in high-speed running (f(X_{IN}) ≤ 40 MHz) and when the count source is f(X_{IN})/2 in low-speed running (f(X_{IN}) ≤ 25 MHz). At this time, the clock source select bit is "0."

Timer A input (External trigger input in one-shot pulse mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TA)}$	TA_{iIN} input cycle time	f(X_{IN}) ≤ 40 MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (200)		ns
		f(X_{IN}) ≤ 25 MHz	$\frac{4 \times 10^9}{f(X_{IN})}$ (160)		ns
$t_{w(TAH)}$	TA_{iIN} input high-level pulse width		80		ns
$t_{w(TAL)}$	TA_{iIN} input low-level pulse width		80		ns

Timer A input (External trigger input in pulse width modulation mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(TAH)}$	TA_{iIN} input high-level pulse width	80		ns
$t_{w(TAL)}$	TA_{iIN} input low-level pulse width	80		ns

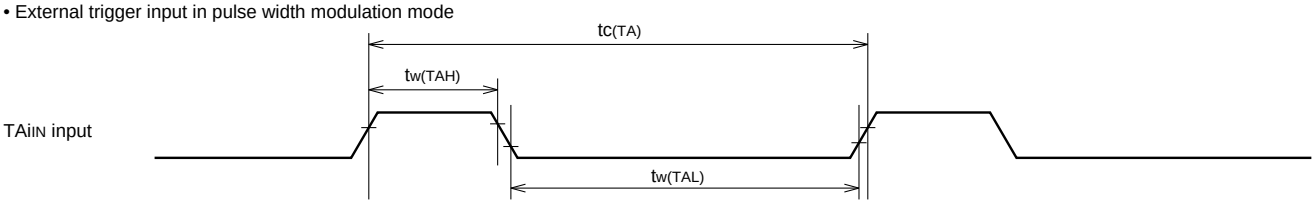
Timer A input (Up-down input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(UP)}$	TA_{iOUT} input cycle time	2000		ns
$t_{w(UPH)}$	TA_{iOUT} input high-level pulse width	1000		ns
$t_{w(UPL)}$	TA_{iOUT} input low-level pulse width	1000		ns
$t_{su(UP\text{-}TIN)}$	TA_{iOUT} input setup time	400		ns
$t_{h(TIN\text{-}UP)}$	TA_{iOUT} input hold time	400		ns

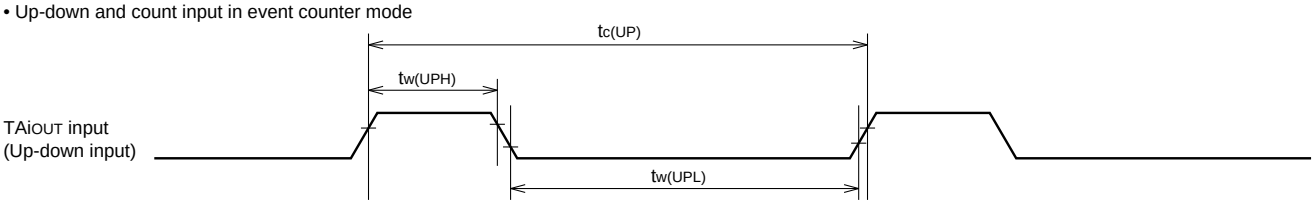
Timer A input (Two-phase pulse input in event counter mode)

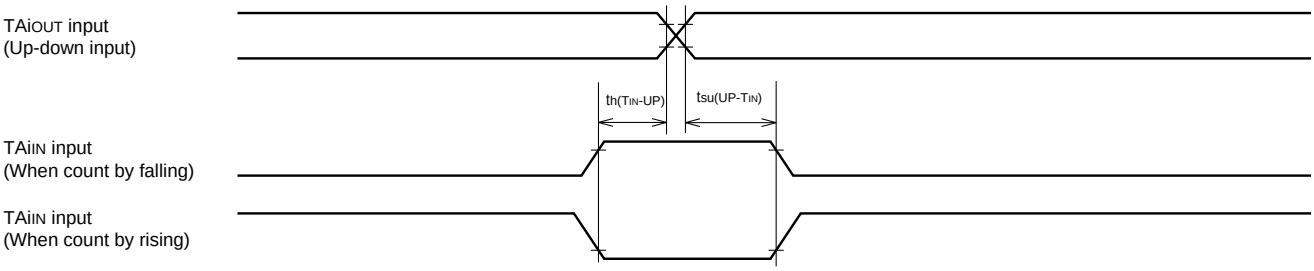
Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	800		ns
$t_{su(TAjIN-TAjOUT)}$	TAjIN input setup time	200		ns
$t_{su(TAjOUT-TAjIN)}$	TAjOUT input setup time	200		ns

- Count input in event counter mode
- Gating input in timer mode
- External trigger input in one-shot pulse mode
- External trigger input in pulse width modulation mode

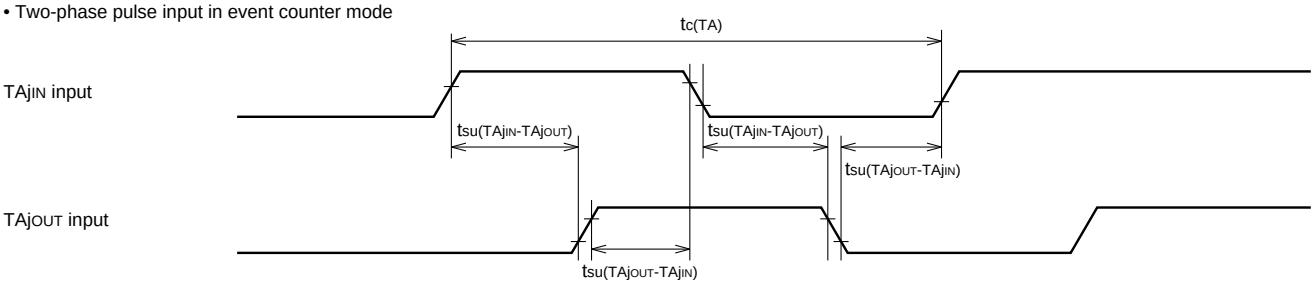


- Up-down and count input in event counter mode





- Two-phase pulse input in event counter mode



Test conditions
- V_{CC} = 5 V±10 %
- Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V

Timer B input (Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (one edge count)	80		ns
$t_{w(TBH)}$	TBiIN input high-level pulse width (one edge count)	40		ns
$t_{w(TBL)}$	TBiIN input low-level pulse width (one edge count)	40		ns
$t_{c(TB)}$	TBiIN input cycle time (both edge count)	160		ns
$t_{w(TBH)}$	TBiIN input high-level pulse width (both edge count)	80		ns
$t_{w(TBL)}$	TBiIN input low-level pulse width (both edge count)	80		ns

Timer B input (Pulse period measurement mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	$f(X_{IN}) \le 40$ MHz	$\frac{16 \times 10^9}{f(X_{IN})}$ (400)		ns
		$f(X_{IN}) \le 25$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (320)		ns
$t_{w(TBH)}$	TBiIN input high-level pulse width	$f(X_{IN}) \le 40$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (200)		ns
		$f(X_{IN}) \le 25$ MHz	$\frac{4 \times 10^9}{f(X_{IN})}$ (160)		ns
$t_{w(TBL)}$	TBiIN input low-level pulse width	$f(X_{IN}) \le 40$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (200)		ns
		$f(X_{IN}) \le 25$ MHz	$\frac{4 \times 10^9}{f(X_{IN})}$ (160)		ns

Note : The TBiIN input cycle time requires 4 or more cycles of count source. The TBiIN input high-level pulse width and the TBiIN input low-level pulse width respectively require 2 or more cycles of the count source. The limits in the table are the values when the count source is $f(X_{IN})/4$ in high-speed running ($f(X_{IN}) \le 40$ MHz) and when the count source is $f(X_{IN})/2$ in low-speed running ($f(X_{IN}) \le 25$ MHz). At this time, the clock source select bit is "0."

Timer B input (Pulse width measurement mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	$f(X_{IN}) \le 40$ MHz	$\frac{16 \times 10^9}{f(X_{IN})}$ (400)		ns
		$f(X_{IN}) \le 25$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (320)		ns
$t_{w(TBH)}$	TBiIN input high-level pulse width	$f(X_{IN}) \le 40$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (200)		ns
		$f(X_{IN}) \le 25$ MHz	$\frac{4 \times 10^9}{f(X_{IN})}$ (160)		ns
$t_{w(TBL)}$	TBiIN input low-level pulse width	$f(X_{IN}) \le 40$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (200)		ns
		$f(X_{IN}) \le 25$ MHz	$\frac{4 \times 10^9}{f(X_{IN})}$ (160)		ns

Note : The TBiIN input cycle time requires 4 or more cycles of count source. The TBiIN input high-level pulse width and the TBiIN input low-level pulse width respectively require 2 or more cycles of the count source. The limits in the table are the values when the count source is $f(X_{IN})/4$ in high-speed running ($f(X_{IN}) \le 40$ MHz) and when the count source is $f(X_{IN})/2$ in low-speed running ($f(X_{IN}) \le 25$ MHz). At this time, the clock source select bit is "0."

A-D trigger input

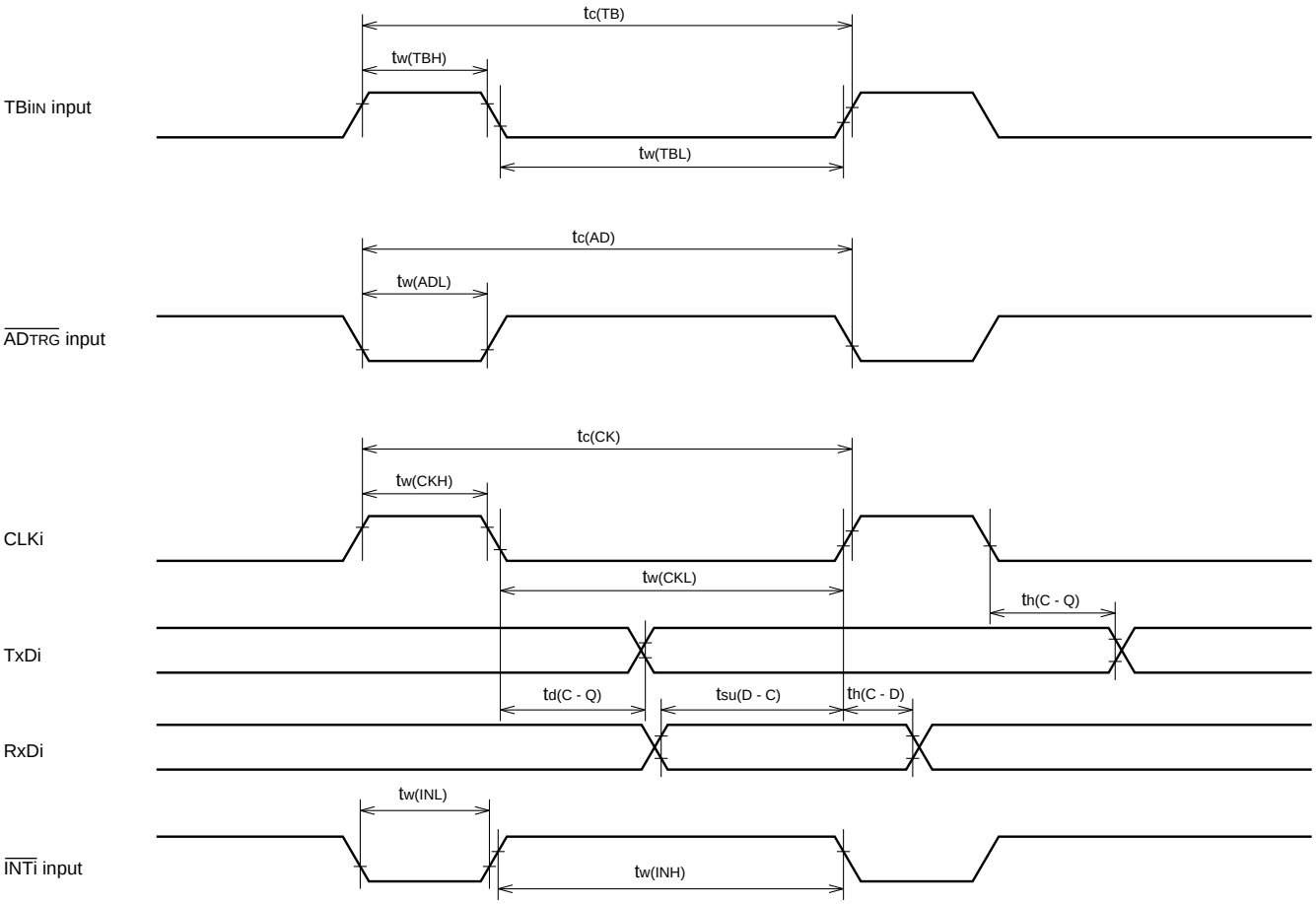
Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{AD_{TRG}}$ input cycle time (minimum allowable trigger)	1000		ns
$t_{w(ADL)}$	$\overline{AD_{TRG}}$ input low-level pulse width	125		ns

Serial I/O

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input high-level pulse width	100		ns
$t_{w(CKL)}$	CLKi input low-level 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	20		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

External interrupt $\overline{INT}_i$ input

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INT}_i$ input high-level pulse width	250		ns
$t_{w(INL)}$	$\overline{INT}_i$ input low-level pulse width	250		ns



Test conditions
- Vcc = 5 V±10 %
- Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V, C_L = 100 pF

READY, HOLD TIMING

Timing requirements

(V_{CC} = 5 V±10 %, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 40 MHz when the clock source select bit = "0"*, unless otherwise noted)

* The rise and fall time of input signal must be 100 ns or less respectively, unless otherwise noted.

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{su(RDY-\phi1)}$	$\overline{RDY}$ input setup time	42		ns
$t_{su(HOLD-\phi1)}$	$\overline{HOLD}$ input setup time	42		ns
$t_{h(\phi1-RDY)}$	$\overline{RDY}$ input hold time	0		ns
$t_{h(\phi1-HOLD)}$	$\overline{HOLD}$ input hold time	0		ns

*: f(X_{IN}) = 20 MHz when the clock source select bit = "1".

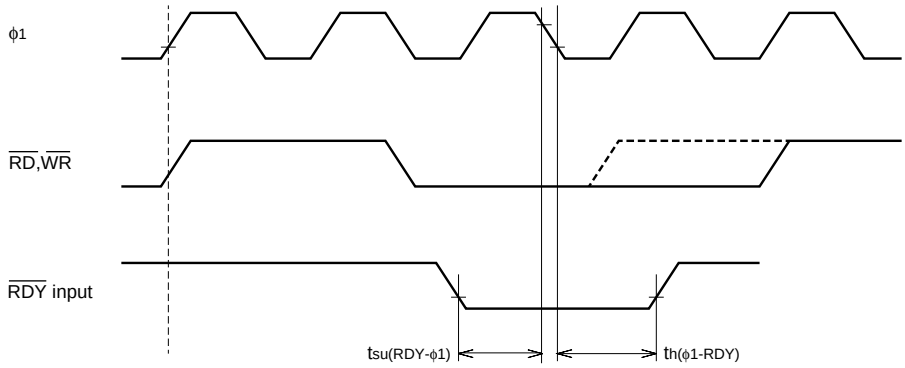
Switching characteristics

(V_{CC} = 5 V±10 %, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 40 MHz when the clock source select bit = "0"*, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{d(\phi1-HLDA)}$	$\overline{HLDA}$ output delay time		50	ns
$t_{pxz(HLDA-RDZ)}$	Floating start delay time (at hold state)		50	ns
$t_{pxz(HLDA-WRZ)}$	Floating start delay time (at hold state)		50	ns
$t_{pxz(HLDA-BHEZ)}$	Floating start delay time (at hold state)		50	ns
$t_{pxz(HLDA-AZ)}$	Floating start delay time (at hold state)		50	ns
$t_{pxz(HLDA-DLZ/DHZ)}$	Floating start delay time (at hold state)		50	ns
$t_{pzx(HLDA-RDZ)}$	Floating release delay time (at hold state)	0		ns
$t_{pzx(HLDA-WRZ)}$	Floating release delay time (at hold state)	0		ns
$t_{pzx(HLDA-BHEZ)}$	Floating release delay time (at hold state)	0		ns
$t_{pzx(HLDA-AZ)}$	Floating release delay time (at hold state)	0		ns
$t_{pzx(HLDA-DLZ/DHZ)}$	Floating release delay time (at hold state)	0		ns

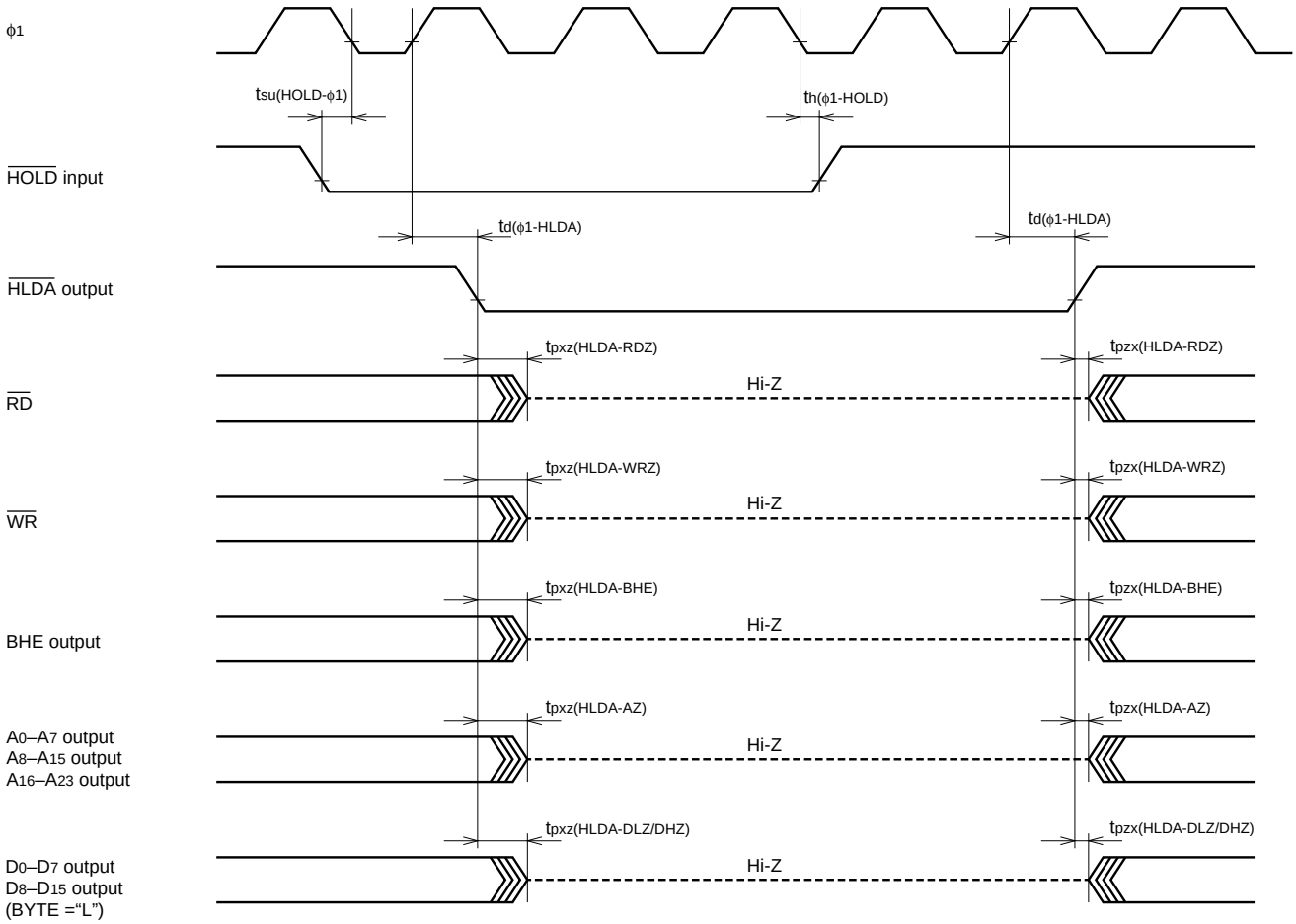
*: f(X_{IN}) = 20 MHz when the clock source select bit = "1".

$\overline{\text{RDY}}$ input (when 3-ϕ access in high-speed running)



✻ $\overline{\text{RDY}}$ input is always sampled at the falling edge of ϕ_1 just before the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals' rise regardless of the bus mode and the number of waits.

$\overline{\text{HOLD}}$ input



Test conditions

- V_{CC} = 5 V±10 %
- $\overline{\text{RDY}}$ input, $\overline{\text{HOLD}}$ input : V_{IL} = 1.0 V, V_{IH} = 4.0 V
- $\overline{\text{HLDA}}$ output : V_{OL} = 0.8 V, V_{OH} = 2.0 V, C_L = 100 pF

Timing requirements (V_{CC} = 5 V±10 %, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 40 MHz when the clock source select bit = "0"✻, unless otherwise noted)

✻ The rise and fall time of input signal must be 100 ns or less respectively, unless otherwise noted.

Single-chip mode

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_c	External clock input cycle time **(Note 1)**	25		ns
$t_{w(H)}$	External clock input high-level pulse width **(Note 2)**	t_c/2 – 8		ns
$t_{w(L)}$	External clock input low-level pulse width **(Note 2)**	t_c/2 – 8		ns
t_r	External clock rise time		8	ns
t_f	External clock fall time		8	ns
$t_{su(PiD–E)}$	Port Pi input setup time (i = 0—11)	60		ns
$t_{h(E–PiD)}$	Port Pi input hold time (i = 0—11)	0		ns

✻: f(X_{IN}) = 20 MHz when the clock source select bit = "1"

Notes 1: When the clock source select bit = "1", t_c's minimum limit is 50 ns.

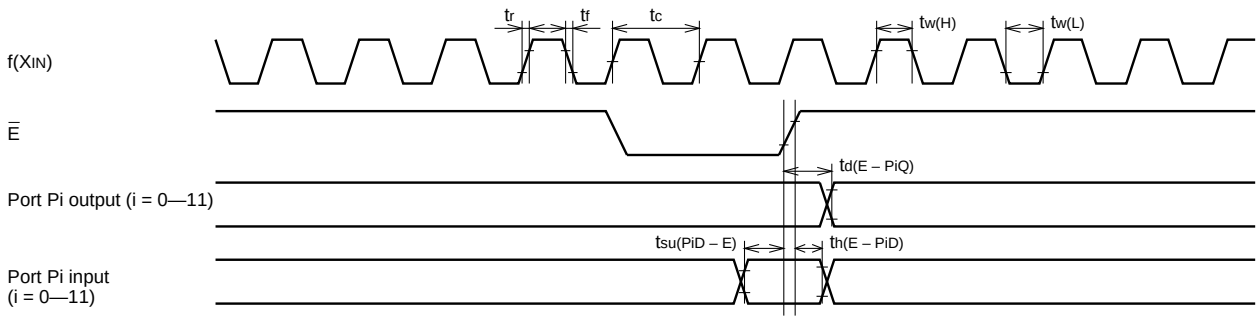
2: When the clock source select bit = "1", set $t_{w(H)}/t_c$ and $t_{w(L)}/t_c$ ratios to 45 to 55 %.

Switching characteristics (V_{CC} = 5 V±10 %, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 40 MHz when the clock source select bit = "0"✻, unless otherwise noted)

(Single-chip mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{d(E–PiQ)}$	Port Pi data output delay time (i = 0—11)		60	ns

✻: f(X_{IN}) = 20 MHz when the clock source select bit = "1"



Test conditions
- V_{CC} = 5 V±10 %
- Intput timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V, C_L = 100 pF

Timing requirements

(V_{CC} = 5 V±10 %, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 25 MHz when the clock source select bit = "0"*, unless otherwise noted)

✻ The rise and fall time of input signal must be 100 ns or less respectively, unless otherwise noted.

Memory expansion and Microprocessor mode : Low-speed running

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_c	External clock input cycle time **(Note 1)**	40		ns
$t_{w(H)}$	External clock input high-level pulse width **(Note 2)**	$t_c/2 - 8$		ns
$t_{w(L)}$	External clock input low-level pulse width **(Note 2)**	$t_c/2 - 8$		ns
t_r	External clock rise time		8	ns
t_f	External clock fall time		8	ns
$t_{su(DH-RD)}$	High-order data input setup time (BYTE = "L")	30		ns
$t_{su(DL-RD)}$	Low-order data input setup time	30		ns
$t_{su(PiD-RD)}$	Port Pi input setup time (i = 4—9, 11)	60		ns
$t_{h(RD-DH)}$	High-order data input hold time (BYTE = "L")	0		ns
$t_{h(RD-DL)}$	Low-order data input hold time	0		ns
$t_{h(RD-PiD)}$	Port Pi input hold time (i = 4—9, 11)	0		ns
$t_{su(A-DL/DH)}$	Data setup time with address stabilized **(Note 3)**		60 (2-ϕ access)	ns
			140 (3-ϕ access)	
			220 (4-ϕ access)	
$t_{su(CS-DL/DH)}$	Data setup time with chip select stabilized **(Note 3)**		60 (2-ϕ access)	ns
			140 (3-ϕ access)	
			220 (4-ϕ access)	
$t_{su(LA-DL)}$	Data setup time with address stabilized **(Note 3)**		55 (2-ϕ access)	ns
			135 (3-ϕ access)	
			215 (4-ϕ access)	

✻: f(X_{IN}) = 12.5 MHz when the clock source selet bit = "1"

Notes 1: When the clock source select bit = "1", t_c's minimum limit is 80 ns.

2: When the clock source select bit = "1", set $t_{w(H)}/t_c$ and $t_{w(L)}/t_c$ ratios to 45 to 55 %.

3: Since the values depend on external clock input frequency f(X_{IN}), calculate them using the bus timing data formula on the page after the next page.

Switching characteristics (V_{CC} = 5 V±10 %, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 25 MHz when the clock source select bit = "0"*, unless otherwise noted)

Memory expansion and Microprocessor mode : Low-speed running

Symbol	Parameter	2-ϕ access		3-ϕ access		4-ϕ access		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{w(\phi H)}$, $t_{w(\phi L)}$	ϕ high-level pulse width, ϕ low-level pulse width **(Note)**	20		20		20		ns
$t_{d(\phi 1-WR)}$	$\overline{WR}$ output delay time	–7	12	–7	12	–7	12	ns
$t_{d(\phi 1-RD)}$	$\overline{RD}$ output delay time	–7	12	–7	12	–7	12	ns
$t_{w(\overline{WR})}$	$\overline{WR}$ low-level pulse width **(Note)**	60		140		140		ns
$t_{w(\overline{RD})}$	RD low-level pulse width **(Note)**	60		140		140		ns
$t_{d(A-WR)}$	Address output delay time **(Note)**	15		15		95		ns
$t_{d(A-RD)}$	Address output delay time **(Note)**	15		15		95		ns
$t_{d(A-ALE)}$	Address output delay time **(Note)**	8		8		55		ns
$t_{d(BHE-WR)}$	$\overline{BHE}$ output delay time **(Note)**	15		15		95		ns
$t_{d(BHE-RD)}$	$\overline{BHE}$ output delay time **(Note)**	15		15		95		ns
$t_{d(BHE-ALE)}$	$\overline{BHE}$ output delay time **(Note)**	8		8		55		ns
$t_{d(CS-WR)}$	Chip select output delay time **(Note)**	15		15		95		ns
$t_{d(CS-RD)}$	Chip select output delay time **(Note)**	15		15		95		ns
$t_{d(CS-ALE)}$	Chip select output delay time **(Note)**	8		8		55		ns
$t_{d(WR-DLQ/DHQ)}$	Data output delay time		35		35		35	ns
$t_{pxz(WR-DLZ/DHZ)}$	Floating start delay time **(Note)**		30		30		30	ns
$t_{d(ALE-WR)}$	ALE output delay time	4		4		4		ns
$t_{d(ALE-RD)}$	ALE output delay time	4		4		4		ns
$t_{w(ALE)}$	ALE pulse width **(Note)**	22		22		62		ns
$t_{h(WR-A)}$	Address hold time **(Note)**	10		10		10		ns
$t_{h(RD-A)}$	Address hold time **(Note)**	10		10		10		ns
$t_{h(WR-BHE)}$	$\overline{BHE}$ hold time **(Note)**	10		10		10		ns
$t_{h(RD-BHE)}$	$\overline{BHE}$ hold time **(Note)**	10		10		10		ns
$t_{h(WR-CS)}$	Chip select hold time **(Note)**	10		10		10		ns
$t_{h(RD-CS)}$	Chip select hold time **(Note)**	10		10		10		ns
$t_{h(WR-DLQ/DHQ)}$	Data hold time **(Note)**	15		15		15		ns
$t_{pzx(WR-DLZ/DHZ)}$	Floating release delay time	0		0		0		ns
$t_{d(LA-WR)}$	Address output delay time **(Note)**	12		12		92		ns
$t_{d(LA-RD)}$	Address output delay time **(Note)**	12		12		92		ns
$t_{d(LA-ALE)}$	Address output delay time **(Note)**	5		5		52		ns
$t_{h(ALE-LA)}$	Address hold time	9		9		25 **(Note)**		ns
$t_{pxz(RD-DLZ)}$	Floating start delay time		5		5		5	ns
$t_{pzx(RD-DLZ)}$	Floating release delay time **(Note)**	18		18		18		ns
$t_{d(WR-PiQ)}$	Port Pi data output delay time (i = 4—9, 11)		60		60		60	ns

*: f(X_{IN}) = 12.5 MHz when the clock source selet bit = "1"

Note: Since the values depend on external clock input frequency f(X_{IN}), calculate them using the bus timing data formula on the next page.

Bus timing data formulas

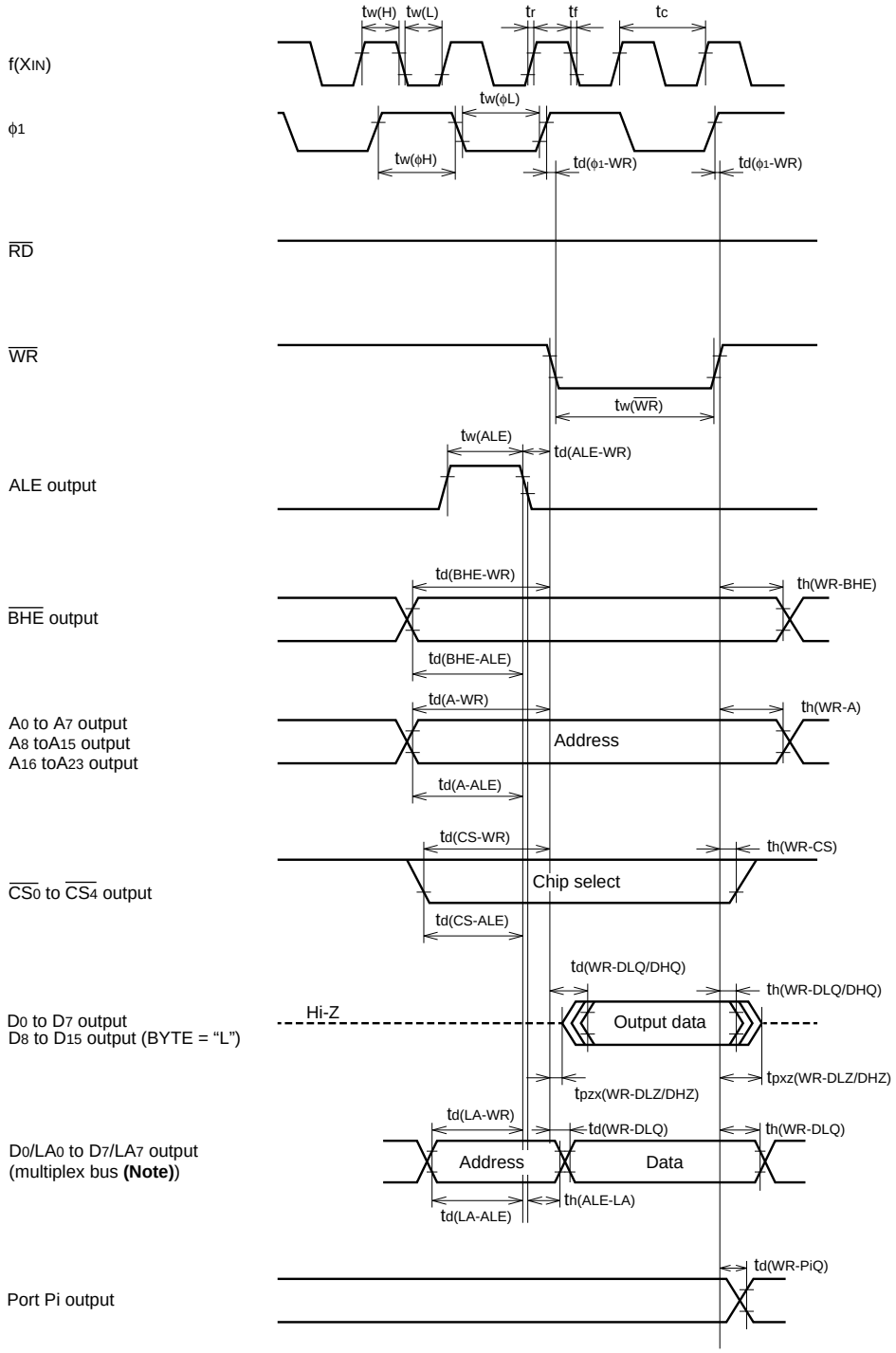
Memory expansion and Microprocessor mode : Low-speed running (V_{CC} = 5 V±10 %, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) ≤ 25 MHz when the clock source select bit = "0"*, unless otherwise noted)

Symbol	Parameter	2-ϕ access	3-ϕ access	4-ϕ access	Unit
$t_{su(A-DL/DH)}$	Data setup time with address stabilized	$\frac{3\times10^9}{f(X_{IN})}-60$	$\frac{5\times10^9}{f(X_{IN})}-60$	$\frac{7\times10^9}{f(X_{IN})}-60$	ns
$t_{su(CS-DL/DH)}$	Data setup time with chip select stabilized	$\frac{3\times10^9}{f(X_{IN})}-60$	$\frac{5\times10^9}{f(X_{IN})}-60$	$\frac{7\times10^9}{f(X_{IN})}-60$	ns
$t_{w(\phi H)}$, $t_{w(\phi L)}$	ϕ high-level pulse width, f low-level pulse width	$\frac{1\times10^9}{f(X_{IN})}-20$	←	←	ns
$t_{w(\overline{WR})}$, $t_{w(\overline{RD})}$	$\overline{WR}$, $\overline{RD}$ low-level pulse width	$\frac{2\times10^9}{f(X_{IN})}-20$	$\frac{4\times10^9}{f(X_{IN})}-20$	←	ns
$t_{d(A-WR)}$	Address output delay time	$\frac{1\times10^9}{f(X_{IN})}-25$	←	$\frac{3\times10^9}{f(X_{IN})}-25$	ns
$t_{d(A-RD)}$	Address output delay time	$\frac{1\times10^9}{f(X_{IN})}-25$	←	$\frac{3\times10^9}{f(X_{IN})}-25$	ns
$t_{d(A-ALE)}$	Address output delay time	$\frac{1\times10^9}{f(X_{IN})}-32$	←	$\frac{3\times10^9}{f(X_{IN})}-65$	ns
$t_{d(BHE-WR)}$	$\overline{BHE}$ output delay time	$\frac{1\times10^9}{f(X_{IN})}-25$	←	$\frac{3\times10^9}{f(X_{IN})}-25$	ns
$t_{d(BHE-RD)}$	$\overline{BHE}$ outupt delay time	$\frac{1\times10^9}{f(X_{IN})}-25$	←	$\frac{3\times10^9}{f(X_{IN})}-25$	ns
$t_{d(BHE-ALE)}$	$\overline{BHE}$ output delay time	$\frac{1\times10^9}{f(X_{IN})}-32$	←	$\frac{3\times10^9}{f(X_{IN})}-65$	ns
$t_{d(CS-WR)}$	Chip select output delay time	$\frac{1\times10^9}{f(X_{IN})}-25$	←	$\frac{3\times10^9}{f(X_{IN})}-25$	ns
$t_{d(CS-RD)}$	Chip select output delay time	$\frac{1\times10^9}{f(X_{IN})}-25$	←	$\frac{3\times10^9}{f(X_{IN})}-25$	ns
$t_{d(CS-ALE)}$	Chip select output delay time	$\frac{1\times10^9}{f(X_{IN})}-32$	←	$\frac{3\times10^9}{f(X_{IN})}-65$	ns
$t_{w(ALE)}$	ALE pulse width	$\frac{1\times10^9}{f(X_{IN})}-18$	←	$\frac{2\times10^9}{f(X_{IN})}-18$	ns
$t_{h(WR-A)}$	Address hold time	$\frac{1\times10^9}{f(X_{IN})}-30$	←	←	ns
$t_{h(RD-A)}$	Address hold time	$\frac{1\times10^9}{f(X_{IN})}-30$	←	←	ns
$t_{d(WR-BHE)}$	$\overline{BHE}$ hold time	$\frac{1\times10^9}{f(X_{IN})}-30$	←	←	ns
$t_{d(RD-BHE)}$	$\overline{BHE}$ hold time	$\frac{1\times10^9}{f(X_{IN})}-30$	←	←	ns
$t_{d(WR-CS)}$	Chip select hold time	$\frac{1\times10^9}{f(X_{IN})}-30$	←	←	ns
$t_{d(RD-CS)}$	Chip select holt time	$\frac{1\times10^9}{f(X_{IN})}-30$	←	←	ns
$t_{h(WR-DLQ/DHQ)}$	Data hold time	$\frac{1\times10^9}{f(X_{IN})}-25$	←	←	ns
$t_{pxz(WR-DLZ/DHZ)}$	Floating start delay time	$\frac{1\times10^9}{f(X_{IN})}-10$	←	←	ns
$t_{su(LA-DL)}$	Data setup time with address stabilized	$\frac{3\times10^9}{f(X_{IN})}-65$	$\frac{5\times10^9}{f(X_{IN})}-65$	$\frac{7\times10^9}{f(X_{IN})}-65$	ns
$t_{d(LA-WR)}$	Address output delay time	$\frac{1\times10^9}{f(X_{IN})}-28$	←	$\frac{3\times10^9}{f(X_{IN})}-28$	ns
$t_{d(LA-RD)}$	Address output delay time	$\frac{1\times10^9}{f(X_{IN})}-28$	←	$\frac{3\times10^9}{f(X_{IN})}-28$	ns
$t_{d(LA-ALE)}$	Address output delay time	$\frac{1\times10^9}{f(X_{IN})}-35$	←	$\frac{2\times10^9}{f(X_{IN})}-28$	ns
$t_{h(ALE-LA)}$	Address hold time	———	———	$\frac{1\times10^9}{f(X_{IN})}-15$	ns
$t_{pzx(RD-DLZ)}$	Floating release delay time	$\frac{1\times10^9}{f(X_{IN})}-22$	←	←	ns

*: f(X_{IN}) ≤ 12.5 MHz when the clock source select bit = "1"

Note: When the clock source select bit is "1", regard f(X_{IN}) in tables as 2·f(X_{IN}).

(when 2-ϕ access in low-speed running <Write>)

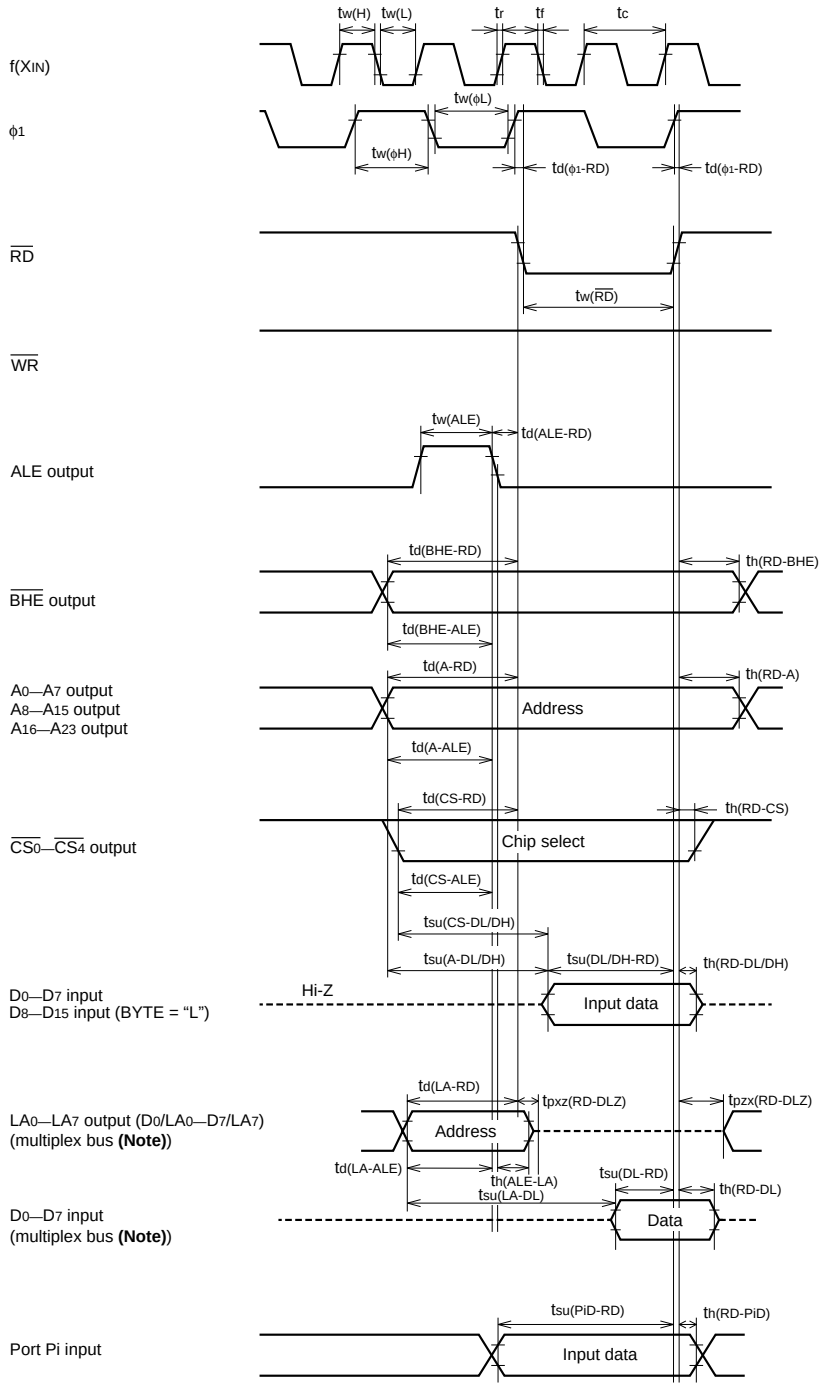


Note: These become a multiplex bus only when all of the following conditions are satisfied:
- BYTE = "H"
- Multiplex bus select bit = "1"
- While the address which corresponds to chip select signal $\overline{CS_4}$ is accessed

Test conditions (except Port Pi, f(XIN))
- VCC = 5 V±10 %
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF
- Data input : VIL = 0.8 V, VIH = 2.5 V

Test conditions (Port Pi, f(XIN))
- VCC = 5 V±10 %
- Input timing voltage : VIL = 1.0 V, VIH = 4.0 V
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF

(when 2-ϕ access in low-speed running <Read>)

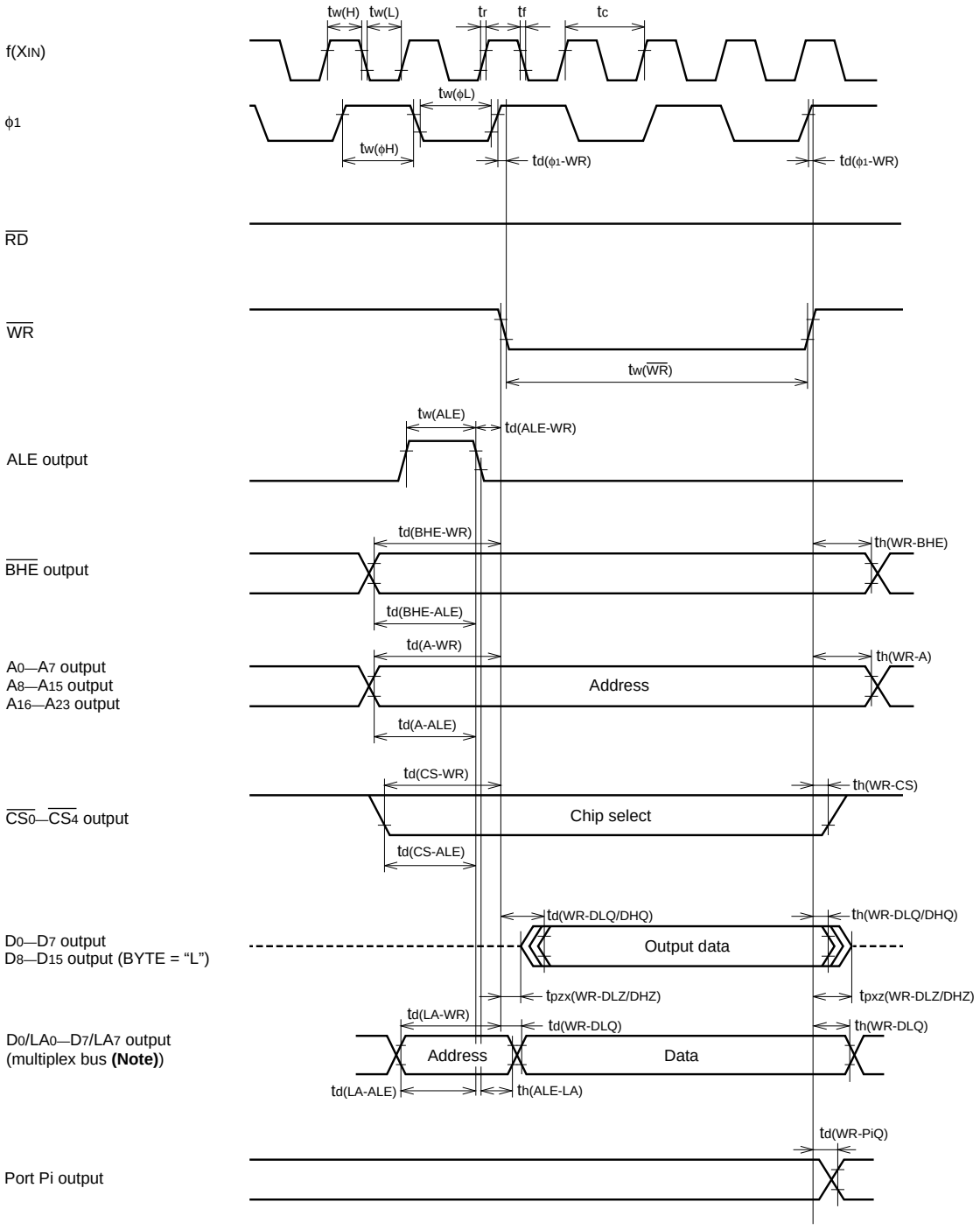


Note: These become a multiplex bus only when all of the following conditions are satisfied:
- BYTE = "H"
- Multiplex bus select bit = "1"
- While the address which corresponds to chip select signal $\overline{CS4}$ is accessed

Test conditions (except Port Pi, f(XIN))
- VCC = 5 V±10 %
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF
- Data input : VIL = 0.8 V, VIH = 2.5 V

Test conditions (Port Pi, f(XIN))
- VCC = 5 V±10 %
- Input timing voltage : VIL = 1.0 V, VIH = 4.0 V
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF

(when 3-ϕ access in low-speed running <Write>)

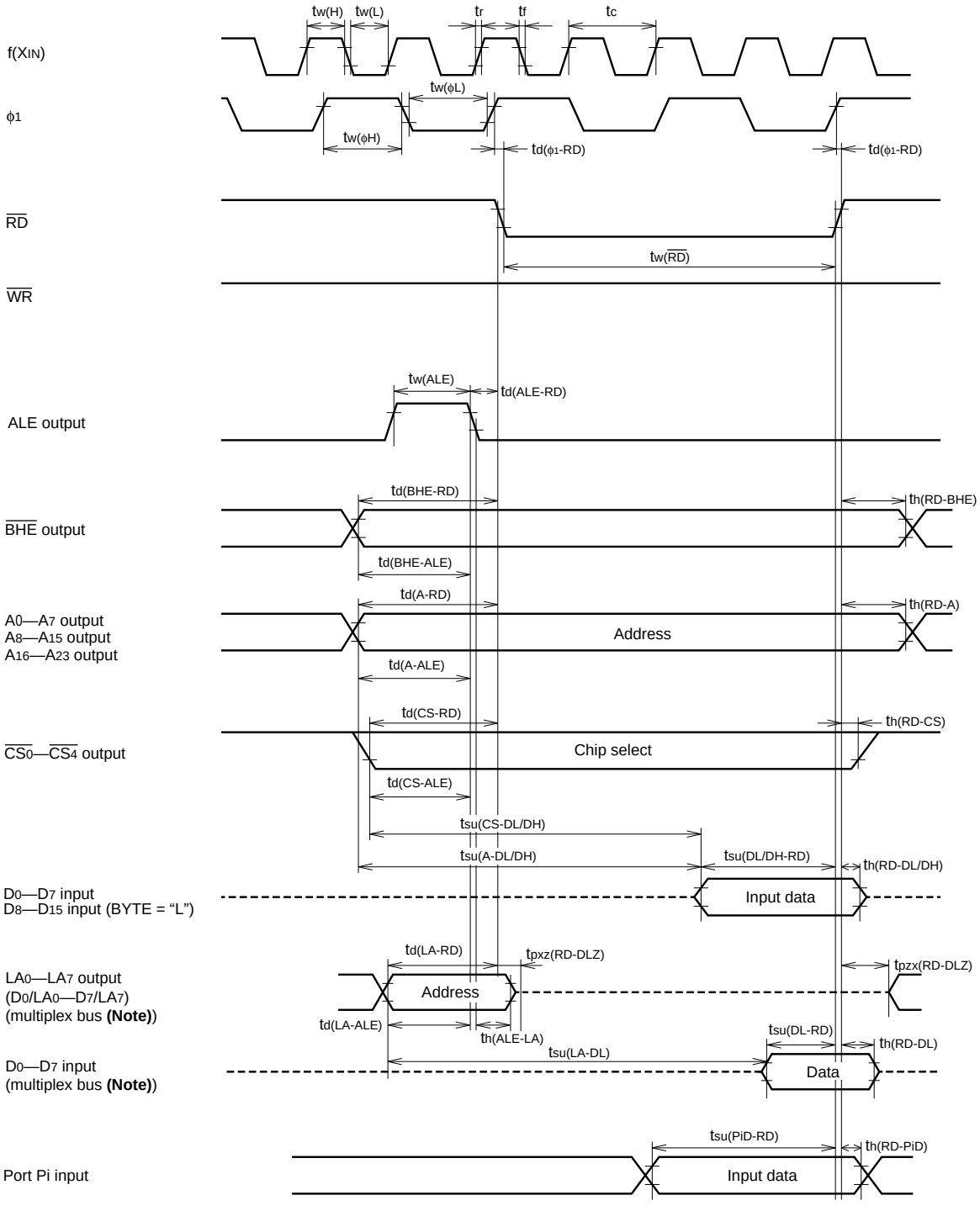


Note: These become a multiplex bus only when all of the following conditions are satisfied:
- •BYTE = "H"
- •Multiplex bus select bit = "1"
- •While the address which corresponds to chip select signal $\overline{CS4}$ is accessed

Test conditions (except Port Pi, f(XIN))
- V_{CC} = 5 V±10 %
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V, C_L = 100 pF
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

Test conditions (Port Pi, f(XIN))
- V_{CC} = 5 V±10 %
- Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V, C_L = 100 pF

(when 3-ϕ access in low-speed running <Read>)

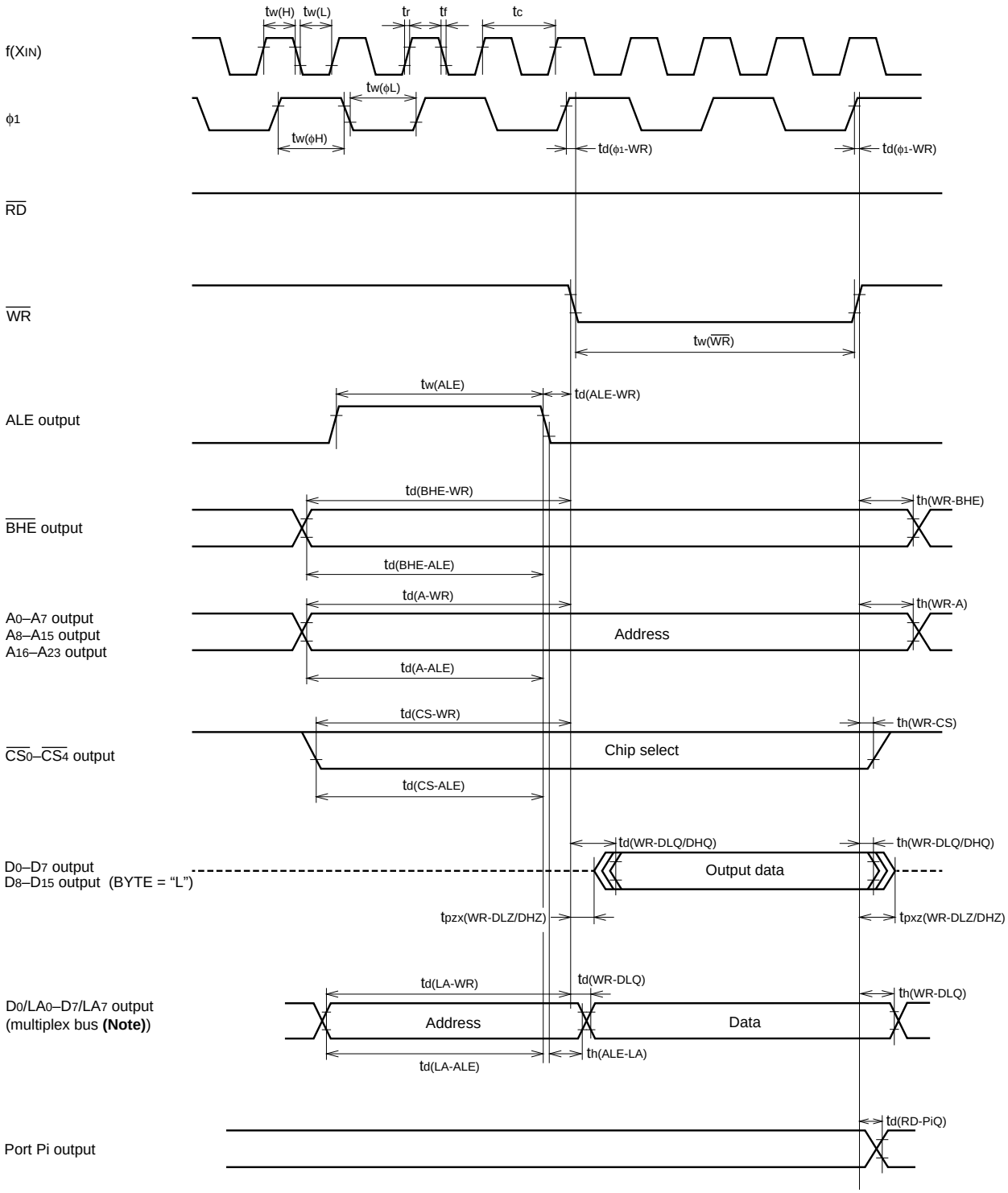


Note: These become a multiplex bus only when all of the following conditions are satisfied:
- BYTE = "H"
- Multiplex bus select bit = "1"
- While the address which corresponds to chip select signal $\overline{CS_4}$ is accessed

Test conditions (except Port Pi, f(XIN))
- VCC = 5 V±10 %
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF
- Data input : VIL = 0.8 V, VIH = 2.5 V

Test conditions (Port Pi, f(XIN))
- VCC = 5 V±10 %
- Input timing voltage : VIL = 1.0 V, VIH = 4.0 V
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF

(when 4-ϕ access in low-speed running <Write>)

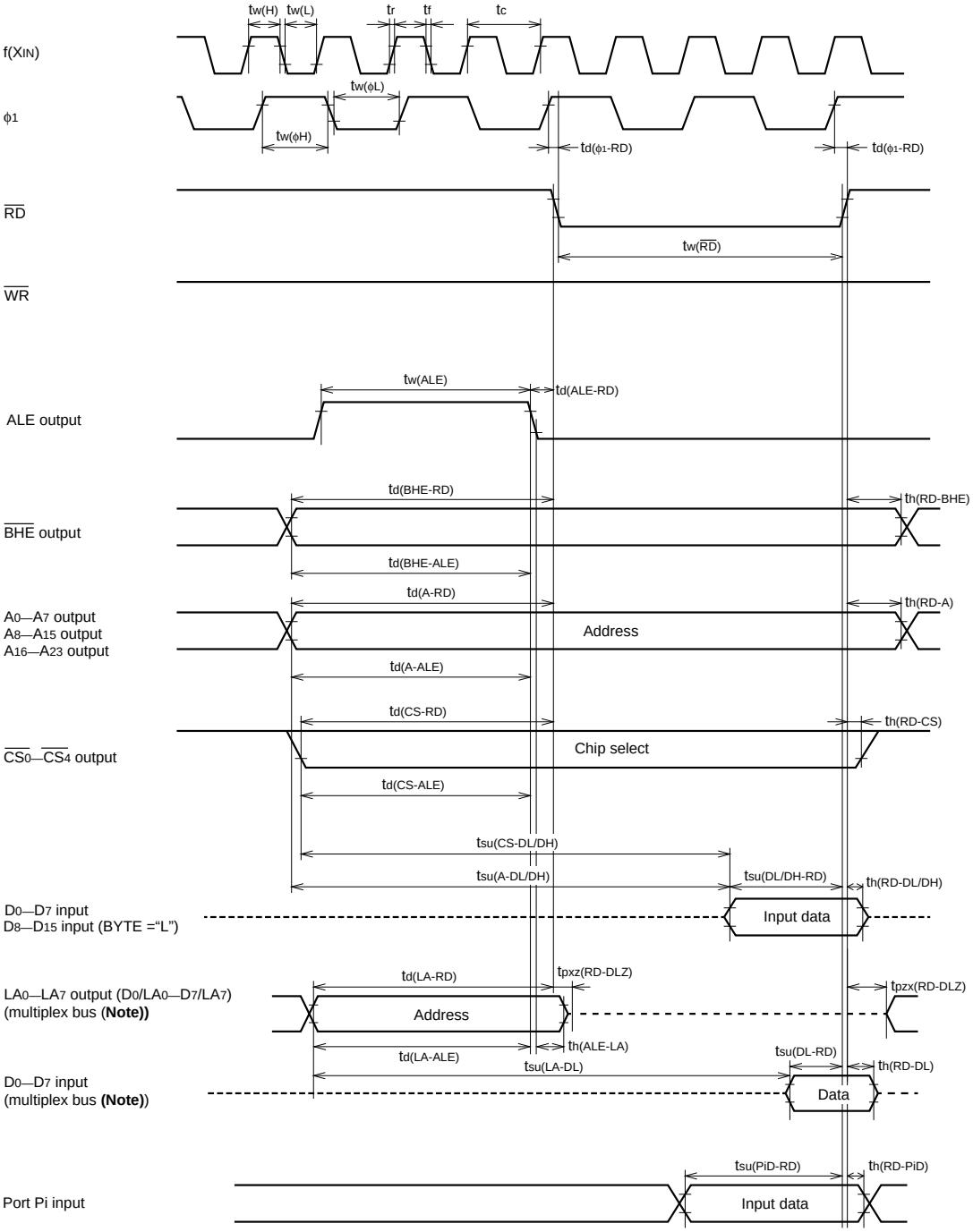


Note: These become a multiplex bus only when all of the following conditions are satisfied:
- BYTE = "H"
- Multiplex bus select bit = "1"
- While the address which corresponds to chip select signal $\overline{CS4}$ is accessed

Test conditions (except Port Pi, f(XIN))
- VCC = 5 V±10 %
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF
- Data input : VIL = 0.8 V, VIH = 2.5 V

Test conditions (Port Pi, f(XIN))
- VCC = 5 V±10 %
- Input timing voltage : VIL = 1.0 V, VIH = 4.0 V
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF

(when 4-ϕ access in low-speed running <Read>)



Note: These become a multiplex bus only when all of the following conditions are satisfied:
- BYTE = "H"
- Multiplex bus select bit = "1"
- While the address which corresponds to chip select signal $\overline{CS4}$ is accessed

Test conditions (except Port Pi, f(XIN))
- VCC = 5 V±10 %
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF
- Data input : VIL = 0.8 V, VIH = 2.5 V

Test conditions (Port Pi, f(XIN))
- VCC = 5 V±10 %
- Input timing voltage : VIL = 1.0 V, VIH = 4.0 V
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF

Timing requirements

(V_{CC} = 5 V±10 %, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN})=40 MHz when the clock source select bit = "0"*, unless otherwise noted)

✻ The rise and fall time of input signal must be 100 ns or less respectively, unless otherwise noted.

Memory expansion and Microprocessor mode : High-speed running

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_c	External clock input cycle time **(Note 1)**	25		ns
$t_{w(H)}$	External clock input high-level pulse width **(Note 2)**	$t_c/2 - 8$		ns
$t_{w(L)}$	External clock input low-level pulse width **(Note 2)**	$t_c/2 - 8$		ns
t_r	External clock rise time		8	ns
t_f	External clock fall time		8	ns
$t_{su(DH-RD)}$	High-order data input setup time (BYTE = "L")	30		ns
$t_{su(DL-RD)}$	Low-order data input setup time	30		ns
$t_{su(PiD-RD)}$	Port Pi input setup time (i = 4—9, 11)	60		ns
$t_{h(RD-DH)}$	High-order data input hold time (BYTE = "L")	0		ns
$t_{h(RD-DL)}$	Low-order data input hold time	0		ns
$t_{h(RD-PiD)}$	Port Pi input hold time (i = 4—9, 11)	0		ns
$t_{su(A-DL/DH)}$	Data setup time with address stabilized **(Note 3)**		65 (3-ϕ access)	ns
			110 (4-ϕ access)	
			160 (5-ϕ access)	
$t_{su(CS-DL/DH)}$	Data setup time with chip select stabilized **(Note 3)**		65 (3-ϕ access)	ns
			110 (4-ϕ access)	
			160 (5-ϕ access)	
$t_{su(LA-DL)}$	Data setup time with address stabilized **(Note 3)**		50 (3-ϕ access)	ns
			100 (4-ϕ access)	
			150 (5-ϕ access)	

✻: f(X_{IN}) = 20 MHz when the clock source selet bit = "1"

Notes 1: When the clock source select bit = "1", t_c's minimum limit is 50 ns.

2: When the clock source select bit = "1", set $t_{w(H)}/t_c$ and $t_{w(L)}/t_c$ ratios to 45 to 55 %.

3: Since the values depend on external clock input frequency f(X_{IN}), calculate them using the bus timing data formula on the page after the next page.

Switching characteristics

(V_{CC} = 5 V±10 %, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 40 MHz when the clock source select bit = "0"*, unless otherwise noted)

Memory expansion and Microprocessor mode : High-speed running

Symbol	Parameter		3-ϕ access		4–ϕ access		5-ϕ access		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{w(\phi H)}$, $t_{w(\phi L)}$	ϕ high-level pulse width, ϕ low-level pulse width	**(Note)**	5		5		5		ns
$t_{d(\phi 1-WR)}$	$\overline{WR}$ output delay time		–7	12	–7	12	–7	12	ns
$t_{d(\phi 1-RD)}$	$\overline{RD}$ output delay time		–7	12	–7	12	–7	12	ns
$t_{w(\overline{WR})}$	$\overline{WR}$ low-level pulse width	**(Note)**	55		80		130		ns
$t_{w(\overline{RD})}$	$\overline{RD}$ low-level pulse width	**(Note)**	55		80		130		ns
$t_{d(A-WR)}$	Address output delay time	**(Note)**	25		45		45		ns
$t_{d(A-RD)}$	Address output delay time	**(Note)**	25		45		45		ns
$t_{d(A-ALE)}$	Address output delay time	**(Note)**	10		35		35		ns
$t_{d(BHE-WR)}$	$\overline{BHE}$ output delay time	**(Note)**	25		45		45		ns
$t_{d(BHE-RD)}$	$\overline{BHE}$ output delay time	**(Note)**	25		45		45		ns
$t_{d(BHE-ALE)}$	$\overline{BHE}$ output delay time	**(Note)**	10		35		35		ns
$t_{d(CS-WR)}$	Chip select output delay time	**(Note)**	25		45		45		ns
$t_{d(CS-RD)}$	Chip select output delay time	**(Note)**	25		45		45		ns
$t_{d(CS-ALE)}$	Chip select output delay time	**(Note)**	10		35		35		ns
$t_{d(WR-DLQ/DHQ)}$	Data output delay time			35		35		35	ns
$t_{pxz(WR-DLZ/DHZ)}$	Floating start delay time	**(Note)**		30		30		30	ns
$t_{d(ALE-WR)}$	ALE output delay time		4		4		4		ns
$t_{d(ALE-RD)}$	ALE output delay time		4		4		4		ns
$t_{w(ALE)}$	ALE pulse width	**(Note)**	10		35		35		ns
$t_{h(WR-A)}$	Address hold time	**(Note)**	10		10		10		ns
$t_{h(RD-A)}$	Address hold time	**(Note)**	10		10		10		ns
$t_{h(WR-BHE)}$	$\overline{BHE}$ hold time	**(Note)**	10		10		10		ns
$t_{h(RD-BHE)}$	$\overline{BHE}$ hold time	**(Note)**	10		10		10		ns
$t_{h(WR-CS)}$	Chip select hold time	**(Note)**	10		10		10		ns
$t_{h(RD-CS)}$	Chip select hold time	**(Note)**	10		10		10		ns
$t_{h(WR-DLQ/DHQ)}$	Data hold time	**(Note)**	15		15		15		ns
$t_{pzx(WR-DLZ/DHZ)}$	Floating release delay time		0		0		0		ns
$t_{d(LA-WR)}$	Address output delay time	**(Note)**	15		40		40		ns
$t_{d(LA-RD)}$	Address output delay time	**(Note)**	15		40		40		ns
$t_{d(LA-ALE)}$	Address output delay time	**(Note)**	5		30		30		ns
$t_{h(ALE-LA)}$	Address hold time	**(Note)**	10		10		10		ns
$t_{PXZ(RD-DLZ)}$	Floating start delay time			5		5		5	ns
$t_{PZX(RD-DLZ)}$	Floating release delay time	**(Note)**	15		15		15		ns
$t_{d(WR-PiQ)}$	Port Pi data output delay time (i = 4—9, 11)			60		60		60	ns

*: f(X_{IN}) = 20 MHz when the clock source selet bit = "1"

Note: Since the values depend on external clock frequency f(X_{IN}), calculate them by using the bus timing data formulas on the next page.

Bus timing data formulas

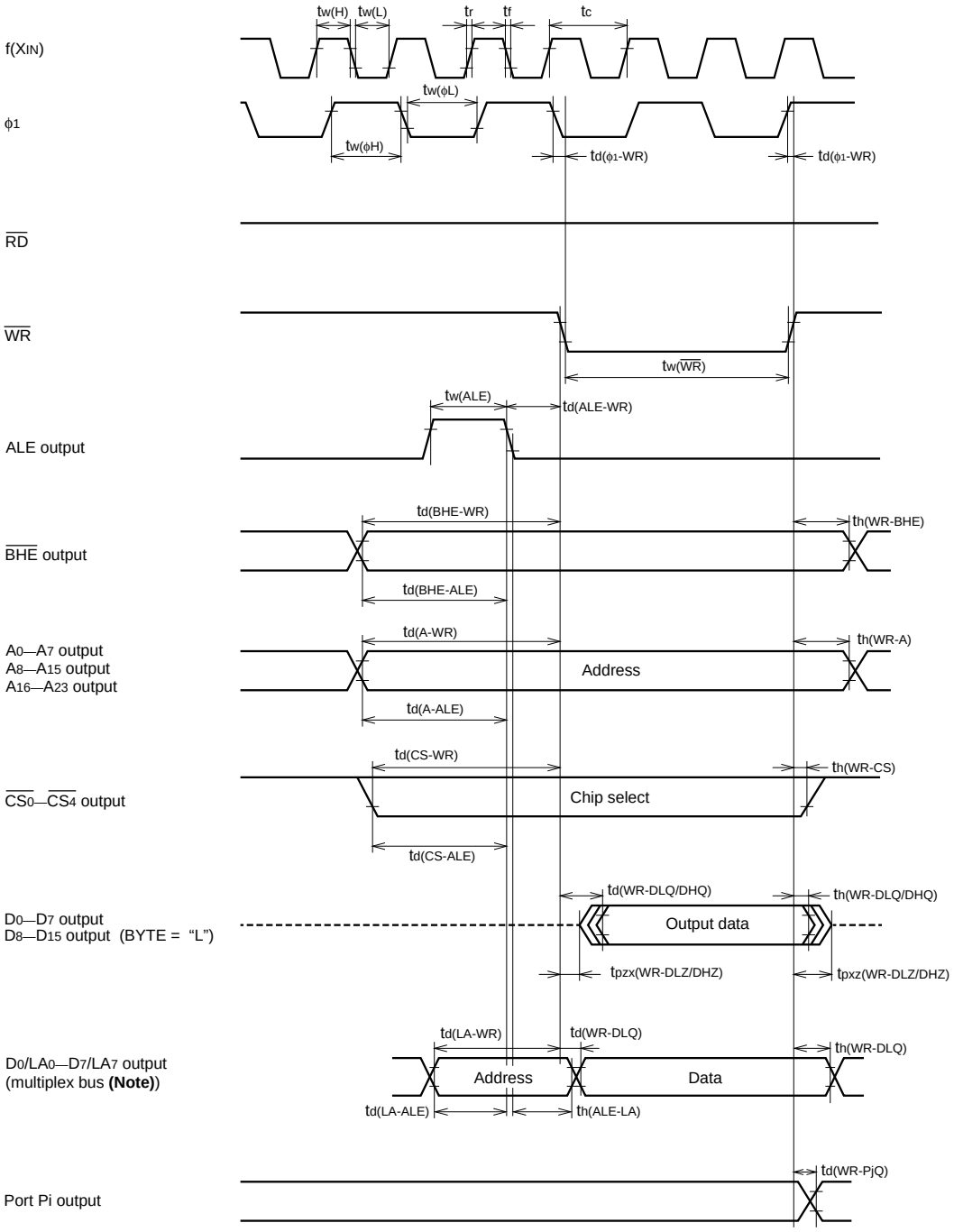
Memory expansion and Microprocessor mode : High-speed running (V_{CC} = 5 V±10 %, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) ≤ 40 MHz when the clock source select bit = "0"*, unless otherwise noted)

Symbol	Parameter	3-ϕ access	4-ϕ access	5-ϕ access	Unit
$t_{su(A-DL/DH)}$	Data setup time with address stabilized	$\frac{5\times10^9}{f(X_{IN})}$ – 60	$\frac{7\times10^9}{f(X_{IN})}$ – 65	$\frac{9\times10^9}{f(X_{IN})}$ – 65	ns
$t_{su(CS-DL/DH)}$	Data setup time with chip select stabilized	$\frac{5\times10^9}{f(X_{IN})}$ – 60	$\frac{7\times10^9}{f(X_{IN})}$ – 65	$\frac{9\times10^9}{f(X_{IN})}$ – 65	ns
$t_{w(\phi H)}$, $t_{w(\phi L)}$	ϕ high-level pulse width, ϕ low-level pulse width	$\frac{1\times10^9}{f(X_{IN})}$ – 20	←	←	ns
$t_{w(\overline{WR})}$, $t_{w(\overline{RD})}$	$\overline{WR}$, $\overline{RD}$ low-level pulse width	$\frac{3\times10^9}{f(X_{IN})}$ – 20	$\frac{4\times10^9}{f(X_{IN})}$ – 20	$\frac{6\times10^9}{f(X_{IN})}$ – 20	ns
$t_{d(A-WR)}$	Address output delay time	$\frac{2\times10^9}{f(X_{IN})}$ – 25	$\frac{3\times10^9}{f(X_{IN})}$ – 30	←	ns
$t_{d(A-RD)}$	Address output delay time	$\frac{2\times10^9}{f(X_{IN})}$ – 25	$\frac{3\times10^9}{f(X_{IN})}$ – 30	←	ns
$t_{d(A-ALE)}$	Address output delay time	$\frac{1\times10^9}{f(X_{IN})}$ – 15	$\frac{2\times10^9}{f(X_{IN})}$ – 15	←	ns
$t_{d(BHE-WR)}$	$\overline{BHE}$ outuput delay time	$\frac{2\times10^9}{f(X_{IN})}$ – 25	$\frac{3\times10^9}{f(X_{IN})}$ – 30	←	ns
$t_{d(BHE-RD)}$	$\overline{BHE}$ outuput delay time	$\frac{2\times10^9}{f(X_{IN})}$ – 25	$\frac{3\times10^9}{f(X_{IN})}$ – 30	←	ns
$t_{d(BHE-ALE)}$	$\overline{BHE}$ outuput delay time	$\frac{1\times10^9}{f(X_{IN})}$ – 15	$\frac{2\times10^9}{f(X_{IN})}$ – 15	←	ns
$t_{d(CS-WR)}$	Chip select output delay time	$\frac{2\times10^9}{f(X_{IN})}$ – 25	$\frac{3\times10^9}{f(X_{IN})}$ – 30	←	ns
$t_{d(CS-RD)}$	Chip select output delay time	$\frac{2\times10^9}{f(X_{IN})}$ – 25	$\frac{3\times10^9}{f(X_{IN})}$ – 30	←	ns
$t_{d(CS-ALE)}$	Chip select output delay time	$\frac{1\times10^9}{f(X_{IN})}$ – 15	$\frac{2\times10^9}{f(X_{IN})}$ – 15	←	ns
$t_{w(ALE)}$	ALE pulse width	$\frac{1\times10^9}{f(X_{IN})}$ – 15	$\frac{2\times10^9}{f(X_{IN})}$ – 15	←	ns
$t_{h(WR-A)}$	Address hold time	$\frac{1\times10^9}{f(X_{IN})}$ – 15	←	←	ns
$t_{h(RD-A)}$	Address hold time	$\frac{1\times10^9}{f(X_{IN})}$ – 15	←	←	ns
$t_{d(WR-BHE)}$	$\overline{BHE}$ hold time	$\frac{1\times10^9}{f(X_{IN})}$ – 15	←	←	ns
$t_{d(RD-BHE)}$	$\overline{BHE}$ hold time	$\frac{1\times10^9}{f(X_{IN})}$ – 15	←	←	ns
$t_{d(WR-CS)}$	Chip select hold time	$\frac{1\times10^9}{f(X_{IN})}$ – 15	←	←	ns
$t_{d(RD-CS)}$	Chip select hold time	$\frac{1\times10^9}{f(X_{IN})}$ – 15	←	←	ns
$t_{h(WR-DLQ/DHQ)}$	Data hold time	$\frac{1\times10^9}{f(X_{IN})}$ – 10	←	←	ns
$t_{pxz(WR-DLZ/DHZ)}$	Floating start delay time	$\frac{1\times10^9}{f(X_{IN})}$ + 5	←	←	ns
$t_{su(LA-DL)}$	Data setup time with address stabilized	$\frac{5\times10^9}{f(X_{IN})}$ – 75	$\frac{7\times10^9}{f(X_{IN})}$ – 75	$\frac{9\times10^9}{f(X_{IN})}$ – 75	ns
$t_{d(LA-WR)}$	Address outuput delay time	$\frac{2\times10^9}{f(X_{IN})}$ – 35	$\frac{3\times10^9}{f(X_{IN})}$ – 35	←	ns
$t_{d(LA-RD)}$	Address outuput delay time	$\frac{2\times10^9}{f(X_{IN})}$ – 35	$\frac{3\times10^9}{f(X_{IN})}$ – 35	←	ns
$t_{d(LA-ALE)}$	Address outuput delay time	$\frac{1\times10^9}{f(X_{IN})}$ – 20	$\frac{2\times10^9}{f(X_{IN})}$ – 20	←	ns
$t_{d(ALE-LA)}$	Address hold time	$\frac{1\times10^9}{f(X_{IN})}$ – 15	←	←	ns
$t_{pzx(RD-DLZ)}$	Floating release delay time	$\frac{1\times10^9}{f(X_{IN})}$ – 10	←	←	ns

*: f(X_{IN}) ≤ 20 MHz when the clock source select bit = "1"

Note: When the clock source select bit is "1", regard f(X_{IN}) in tables as 2·f(X_{IN}).

(when 3-ϕ access in high-speed running <Write>)

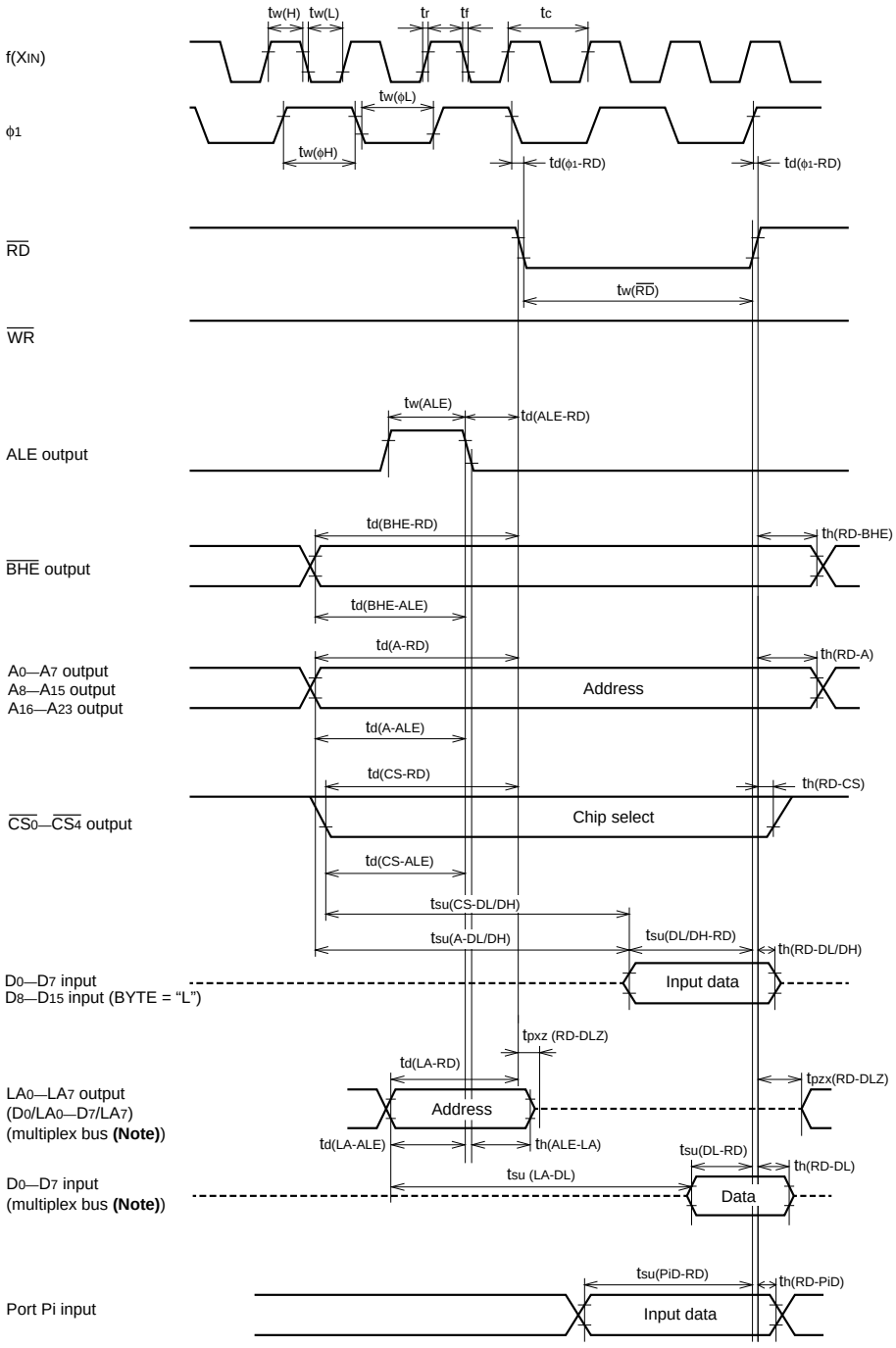


Note: These become a multiplex bus only when all of the following conditions are satisfied:
- BYTE = "H"
- Multiplex bus select bit = "1"
- While the address which corresponds to chip select signal $\overline{CS_4}$ is accessed

Test conditions (except Port Pi, f(XIN))
- VCC = 5 V±10 %
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF
- Data input : VIL = 0.8 V, VIH = 2.5 V

Test conditions (Port Pi, f(XIN))
- VCC = 5 V±10 %
- Input timing voltage : VIL = 1.0 V, VIH = 4.0 V
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF

(when 3-ϕ access in high-speed running <Read>)

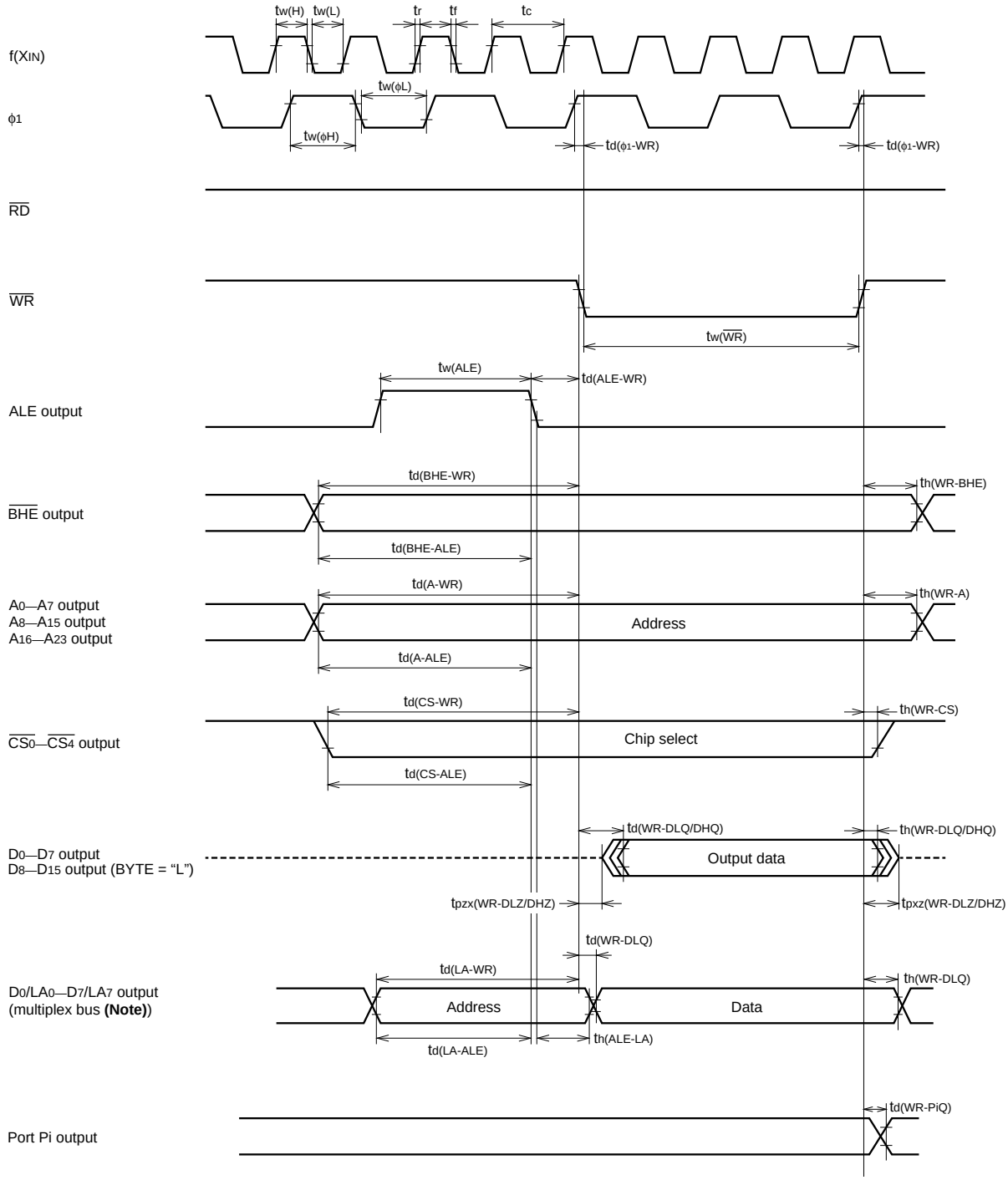


Note: These become a multiplex bus only when all of the following conditions are satisfied:
- BYTE = "H"
- Multiplex bus select bit = "1"
- While the address which corresponds to chip select signal $\overline{CS4}$ is accessed

Test conditions (except Port Pi, f(XIN))
- VCC = 5 V±10 %
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF
- Data input : VIL = 0.8 V, VIH = 2.5 V

Test conditions (Port Pi, f(XIN))
- VCC = 5 V±10 %
- Input timing voltage : VIL = 1.0 V, VIH = 4.0 V
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF

(when 4-ϕ access in high-speed running <Write>)

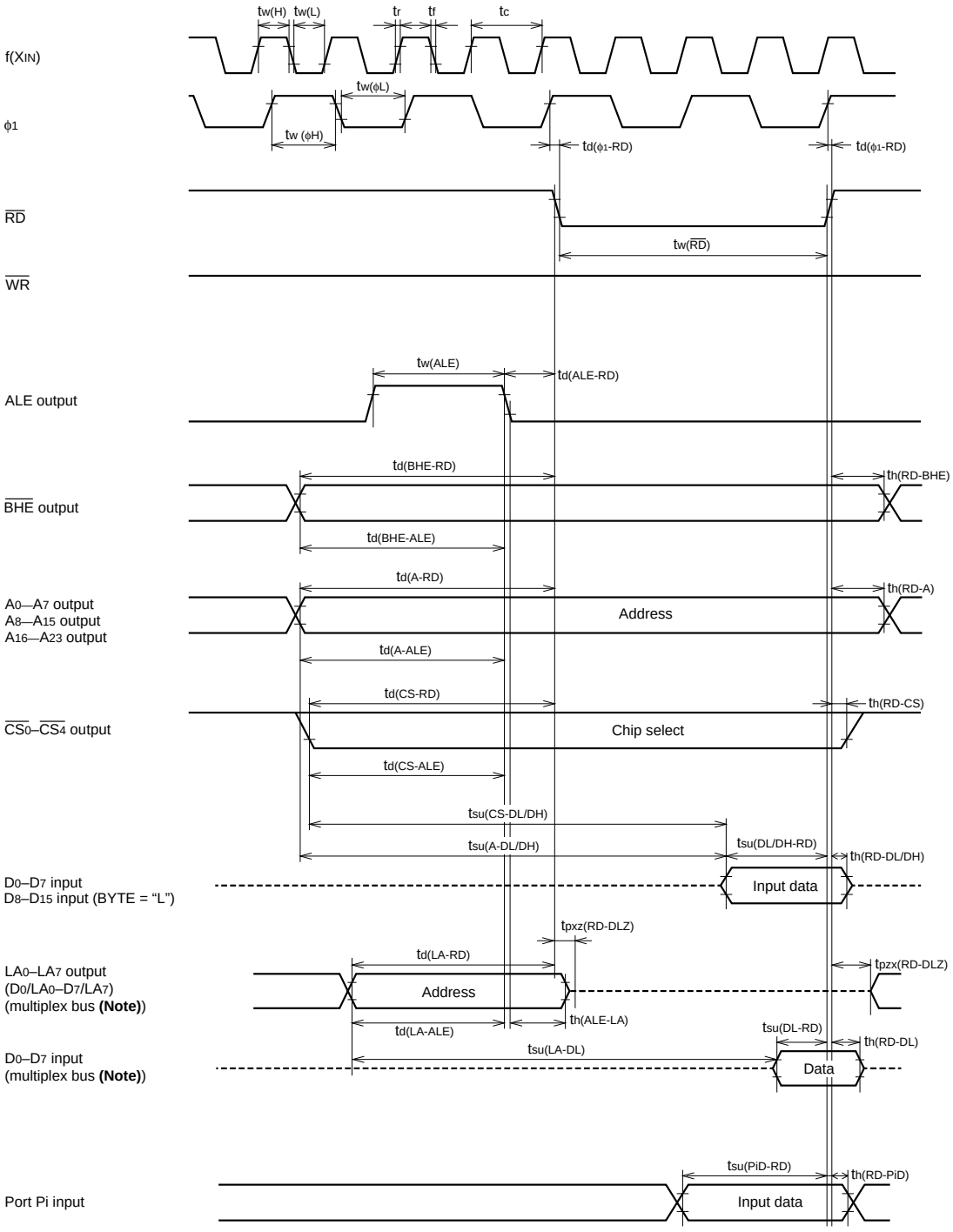


Note: These become a multiplex bus only when all of the following conditions are satisfied:
- BYTE = "H"
- Multiplex bus select bit = "1"
- While the address which corresponds to chip select signal $\overline{CS_4}$ is accessed

Test conditions (except Port Pi, f(XIN))
- VCC = 5 V±10 %
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF
- Data input : VIL = 0.8 V, VIH = 2.5 V

Test conditions (Port Pi, f(XIN))
- VCC = 5 V±10 %
- Input timing voltage : VIL = 1.0 V, VIH = 4.0 V
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF

(when 4-ϕ access in high-speed running <Read>)

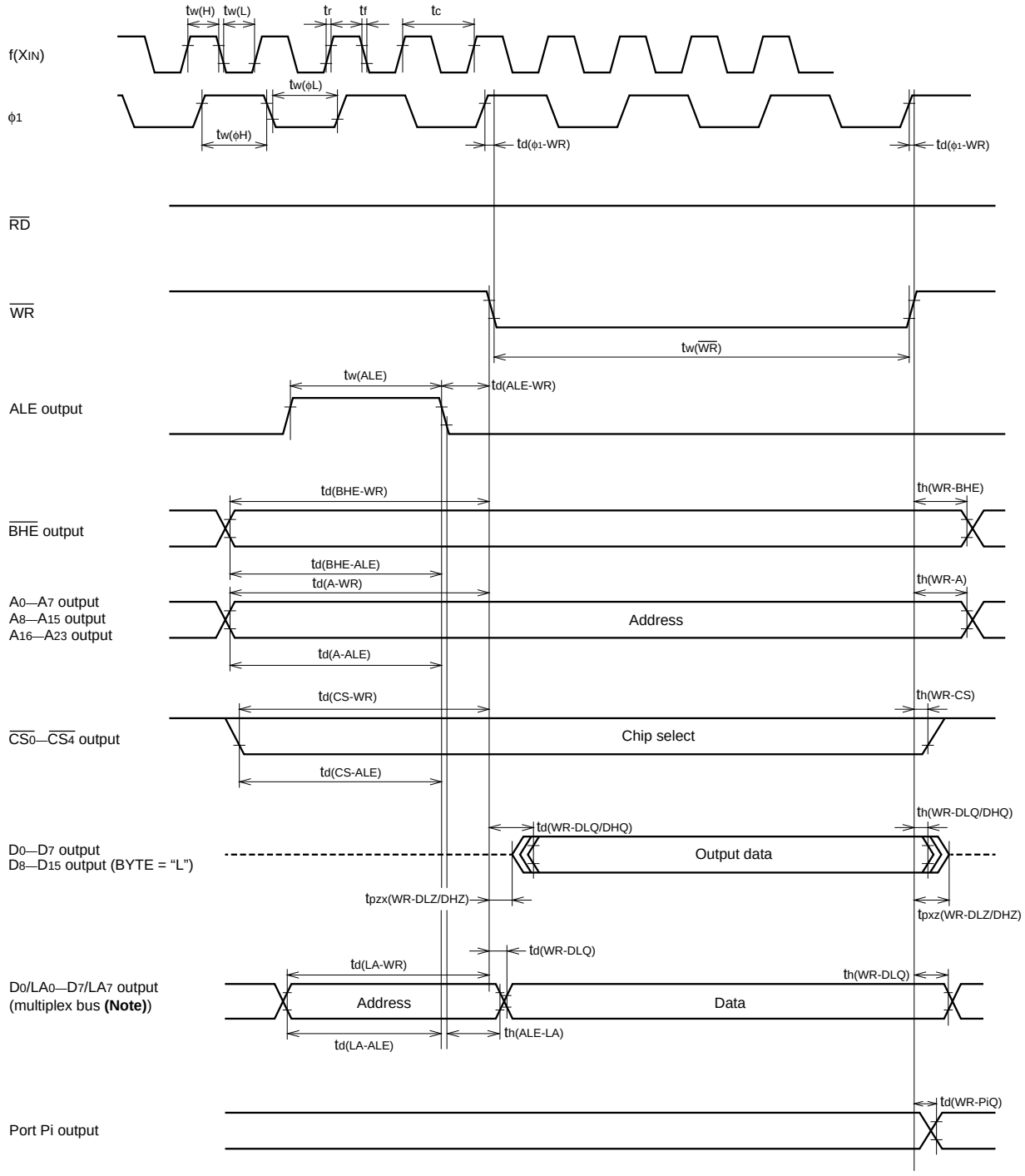


Note: These become a multiplex bus only when all of the following conditions are satisfied:
- BYTE = "H"
- Multiplex bus select bit = "1"
- While the address which corresponds to chip select signal $\overline{CS4}$ is accessed

Test conditions (except Port Pi, f(XIN))
- VCC = 5 V±10 %
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF
- Data input : VIL = 0.8 V, VIH = 2.5 V

Test conditions (Port Pi, f(XIN))
- VCC = 5 V±10 %
- Input timing voltage : VIL = 1.0 V, VIH = 4.0 V
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF

(when 5-ϕ access in high-speed running <Write>)

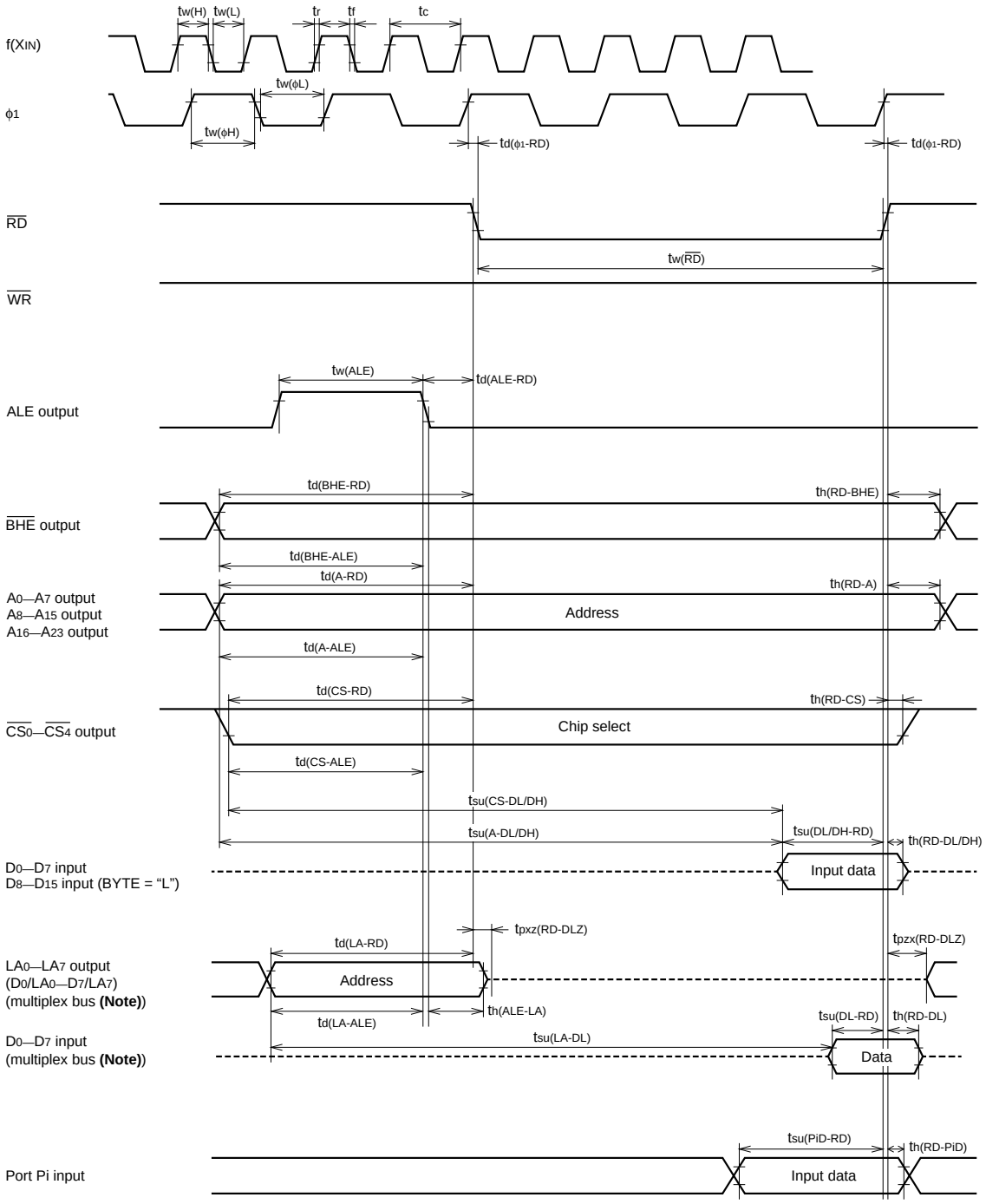


Note: These become a multiplex bus only when all of the following conditions are satisfied:
- BYTE = "H"
- Multiplex bus select bit = "1"
- While the address which corresponds to chip select signal $\overline{CS4}$ is accessed

Test conditions (except Port Pi, f(XIN))
- VCC = 5 V±10 %
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF
- Data input : VIL = 0.8 V, VIH = 2.5 V

Test conditions (Port Pi, f(XIN))
- VCC = 5 V±10 %
- Input timing voltage : VIL = 1.0 V, VIH = 4.0 V
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF

(when 5-ϕ access in high-speed running <Read>)



Note: These become a multiplex bus only when all of the following conditions are satisfied:
- BYTE = "H"
- Multiplex bus select bit = "1"
- While the address which corresponds to chip select signal $\overline{CS_4}$ is accessed

Test conditions (except Port Pi, f(XIN))
- VCC = 5 V±10 %
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF
- Data input : VIL = 0.8 V, VIH = 2.5 V

Test conditions (Port Pi, f(XIN))
- VCC = 5 V±10 %
- Input timing voltage : VIL = 1.0 V, VIH = 4.0 V
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 100 pF

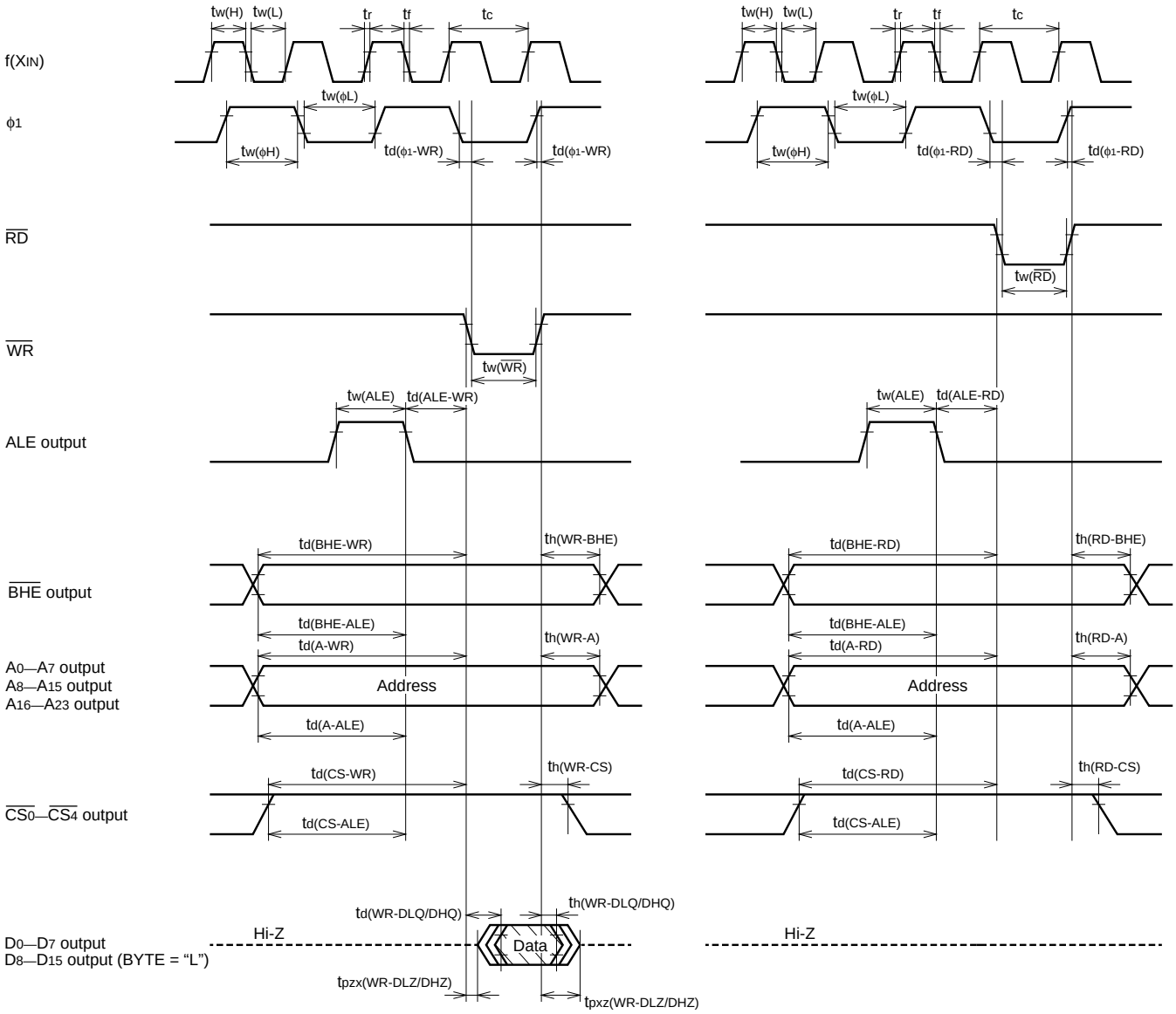
<NOTE> External bus timing when internal memory area is accessed (2-ϕ access) in high-speed running

(V_{CC} = 5 V±10 %, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) ≤ 40 MHz when the clock source select bit = "0"*)

Symbol	Parameter	f (X_{IN}) = 40 MHz**		Bus timing data formula	Unit
		Min.	Max.		
$t_{w(\phi H)}$, $t_{w(\phi L)}$	ϕ high-level pulse width, ϕ low-level pulse width	5		$\frac{1\times10^9}{f(X_{IN})}$ – 20	ns
$t_{d(\phi 1-WR)}$	$\overline{WR}$ output delay time	–7	12		ns
$t_{d(\phi 1-RD)}$	$\overline{RD}$ output delay time	–7	12		ns
$t_{w(\overline{WR})}$	$\overline{WR}$ low-level pulse width	5		$\frac{1\times10^9}{f(X_{IN})}$ – 20	ns
$t_{w(\overline{RD})}$	$\overline{RD}$ low-level pulse width	5		$\frac{1\times10^9}{f(X_{IN})}$ – 20	ns
$t_{d(A-WR)}$	Address output delay time	25		$\frac{2\times10^9}{f(X_{IN})}$ – 25	ns
$t_{d(A-RD)}$	Address output delay time	25		$\frac{2\times10^9}{f(X_{IN})}$ – 25	ns
$t_{d(A-ALE)}$	Address output delay time	10		$\frac{2\times10^9}{f(X_{IN})}$ – 40	ns
$t_{d(BHE-WR)}$	$\overline{BHE}$ output delay time	25		$\frac{2\times10^9}{f(X_{IN})}$ – 25	ns
$t_{d(BHE-RD)}$	$\overline{BHE}$ output delay time	25		$\frac{2\times10^9}{f(X_{IN})}$ – 25	ns
$t_{d(BHE-ALE)}$	$\overline{BHE}$ output delay time	10		$\frac{2\times10^9}{f(X_{IN})}$ – 40	ns
$t_{d(CS-WR)}$	Chip select output delay time	25		$\frac{2\times10^9}{f(X_{IN})}$ – 25	ns
$t_{d(CS-RD)}$	Chip select output delay time	25		$\frac{2\times10^9}{f(X_{IN})}$ – 25	ns
$t_{d(CS-ALE)}$	Chip select output delay time	10		$\frac{2\times10^9}{f(X_{IN})}$ – 40	ns
$t_{d(WR-DLQ/DHQ)}$	Data output delay time		35	————	ns
$t_{pxz(WR-DLZ/DHZ)}$	Floating start delay time	30		$\frac{1\times10^9}{f(X_{IN})}$ + 5	ns
$t_{d(ALE-WR)}$	ALE output delay time	4		————	ns
$t_{d(ALE-RD)}$	ALE output delay time	4		————	ns
$t_{w(ALE)}$	ALE pulse width	10		$\frac{1\times10^9}{f(X_{IN})}$ – 15	ns
$t_{h(WR-A)}$	Address hold time	10		$\frac{1\times10^9}{f(X_{IN})}$ – 15	ns
$t_{h(RD-A)}$	Address hold time	10		$\frac{1\times10^9}{f(X_{IN})}$ – 15	ns
$t_{d(WR-BHE)}$	$\overline{BHE}$ hold time	10		$\frac{1\times10^9}{f(X_{IN})}$ – 15	ns
$t_{d(RD-BHE)}$	$\overline{BHE}$ hold time	10		$\frac{1\times10^9}{f(X_{IN})}$ – 15	ns
$t_{d(WR-CS)}$	Chip select hold time	10		$\frac{1\times10^9}{f(X_{IN})}$ – 15	ns
$t_{d(RD-CS)}$	Chip select hold time	10		$\frac{1\times10^9}{f(X_{IN})}$ – 15	ns
$t_{h(WR-DLQ/DHQ)}$	Data hold time	15		$\frac{1\times10^9}{f(X_{IN})}$ – 10	ns
$t_{pzx(WR-DLZ/DHZ)}$	Floating release delay time	0		————	ns

*: f(X_{IN}) ≤ 20 MHz when the clock source select bit = "1".
**: f(X_{IN}) = 20 MHz when the clock source select bit = "1".

(External bus timing on internal RAM access (2-ϕ access) in high-speed running)



✻ The value of output data is undefined.

Test conditions
- V_{CC} = 5 V±10 %
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V, C_L = 100 pF

GZZ–SH00–85B<85A0>

Mask ROM number	

7700 FAMILY MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37754M8C-XXXGP
M37754M8C-XXXHP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ※

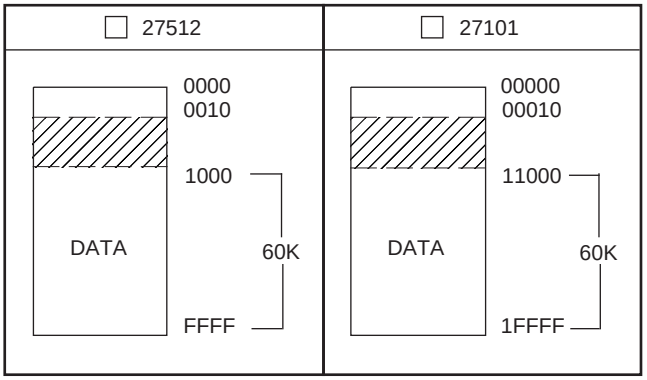
※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date:			

※1. Confirmation

Specify the name of the product being ordered.
Three sets of EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data.
We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data.
Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas [ ][ ][ ][ ] (hexadecimal notation)

EPROM Type :



(1) Set "FF16" in the shaded area.
(2) Address 016 to 1016 are the area for storing the data on model designation and options.This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

Address		Address		Address
4D	0	43	8	Option data 10
33	1	2D	9	
37	2	FF	A	
37	3	FF	B	
35	4	FF	C	
34	5	FF	D	
4D	6	FF	E	
38	7	FF	F	

※2. STP instruction option

One of the following sets of data should be written to the option data address (1016) of the EPROM you have ordered.
Check @ in the appropriate box.

☐ STP instruction enable 0116 Address 1016
☐ STP instruction disable 0016 Address 1016

※3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 100P6S Mark Specification Form (for M37754M8C-XXXGP), 100P6Q Mark Specification Form (for M37754M8C-XXXHP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

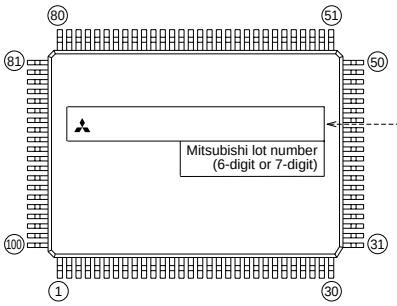
MITSUBISHI ELECTRIC

100P6S (100-PIN QFP) MARK SPECIFICATION FORM

Mitsubishi IC catalog name

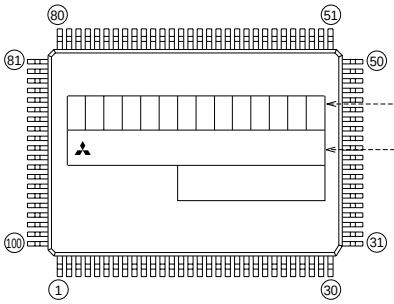
Please choose one of the marking types below (A, B, C), and enter the Mitsubishi catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark



Mitsubishi IC catalog name

B. Customer's Parts Number + Mitsubishi catalog name



Customer's Parts Number

Note : The fonts and size of characters are standard Mitsubishi type.

Mitsubishi IC catalog name

Note1 : The mark field should be written right aligned.

2 : The fonts and size of characters are standard Mitsubishi type.

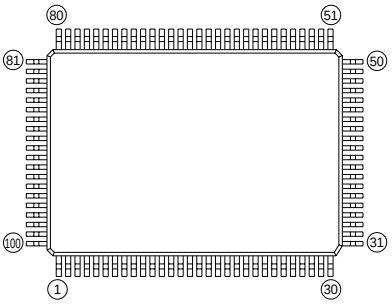
3 : Customer's Parts Number can be up to 14 characters : Only 0 ~ 9, A ~ Z, +, –, /, (,), &, ©, . (periods), , (commas) are usable.

4 : If the Mitsubishi logo is not required, check the box below.

Mitsubishi logo is not required

☐

C. Special Mark Required



Note1 : If the Special Mark is to be Printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated as close as possible.
Mitsubishi lot number (6-digit or 7-digit) and Mask ROM number (3-digit) are always marked.

2 : If the customer's trade mark logo must be used in the Special Mark, check the box below.
Please submit a clean original of the logo.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

Special logo required

☐

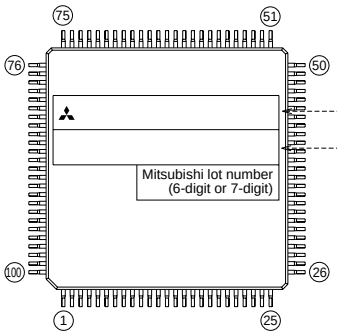
MITSUBISHI ELECTRIC

100P6Q (100-PIN LQFP) MARK SPECIFICATION FORM

Mitsubishi IC catalog name

Please choose one of the marking types below (A, B, C), and enter the Mitsubishi catalog name and the special mark (if needed).

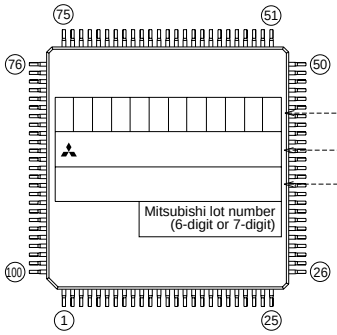
A. Standard Mitsubishi Mark



Mitsubishi IC catalog name

Mitsubishi IC catalog name

B. Customer's Parts Number + Mitsubishi catalog name



Customer's Parts Number

Note : The fonts and size of characters are standard Mitsubishi type.

Mitsubishi IC catalog name

Note1 : The mark field should be written right aligned.

2 : The fonts and size of characters are standard Mitsubishi type.

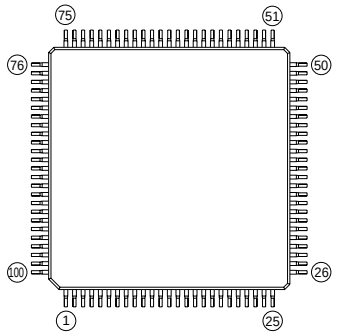
3 : Customer's Parts Number can be up to 12 characters : Only 0 ~ 9, A ~ Z, +, –, /, (,), &, ©, . (periods), , (commas) are usable.

4 : If the Mitsubishi logo is not required, check the box below.

Mitsubishi logo is not required

☐

C. Special Mark Required



Note1 : If the Special Mark is to be Printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated as close as possible.
Mitsubishi lot number (6-digit or 7-digit) and Mask ROM number (3-digit) are always marked.

2 : If the customer's trade mark logo must be used in the Special Mark, check the box below.
Please submit a clean original of the logo.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

Special logo required

☐

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 or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams and charts, represent 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.
- 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.

New publication, effective Apr. 1999.
Specifications subject to change without notice.

MITSUBISHI ELECTRIC

REVISION DESCRIPTION LIST	M37754M8C-XXXGP/HP DATA SHEET

Rev. No.	Revision Description	Rev. date
1.0	First Edition	971114
1.01	(1) Page 14 is updated. (The previous version of this page cannot be read in.) (2) The following are added: •MASK ROM ORDER CONFIRMATION FORM •MARK SPECIFICATION FORM	980602
2.00	(1) For the “valid output polarity select bit for interrupt request (bit 1 at address $1C_{16}$)” (three-phase mode 1), it’s name and function are corrected: • New bit name in three-phase mode 1: interrupt validity output select bit • Corrected function: 0: Timer B2 interrupt request generated at each even-numbered underflow of timer B2 1: Timer B2 interrupt request generated at each odd-numbered underflow of timer B2 • Related pages: pages 37, 38, 40 (2) For the following register, it’s internal status after reset is corrected: • Target register: processor mode register 0 (address $5E_{16}$) • Correction: the status of bit 1 is “0”. (Not “1”.) • Related page: page 63 (3) The names of registers at addresses $5C_{16}$, $5D_{16}$ are corrected: • Address $5C_{16}$: timer B1 mode register • Address $5D_{16}$: timer B2 mode register • Related page: page 63 (4) For the “timer A write flag (address 45_{16})”, it’s name and it’s bit name are corrected: • New register name: timer A write register • New bit name: timer Ai write bit (i = 0 to 2) • Related pages: pages 8, 37, 40, 63	990428