To all our customers

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

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

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

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

MITSUBISHI 16-BIT SINGLE-CHIP MICROCOMPUTER
7700 FAMILY / 7751 SERIES

7751
Group

User's Manual

RENESAS
Renesas Technology Corp.

Preface

This manual describes the hardware of the Mitsubishi CMOS 16-bit microcomputers 7751 Group. After reading this manual, the users will be able to understand the functions, so that they can utilize their capabilities fully.

For details concerning the software, refer to the 7751 Series Software Manual. For details concerning the development support tools (assembler, emulation pods), refer to the respective user's manuals.

Table of Contents

CHAPTER 1. DESCRIPTION

CHAPTER 2. CENTRAL PROCESSING UNIT (CPU)

CHAPTER 3. INPUT/OUTPUT PINS

CHAPTER 4. INTERRUPTS

CHAPTER 5. TIMER A

CHAPTER 6. TIMER B

CHAPTER 7. SERIAL I/O

CHAPTER 8. A-D CONVERTER

CHAPTER 9. WATCHDOG TIMER

CHAPTER 10. STOP MODE

CHAPTER 11. WAIT MODE

CHAPTER 12. CONNECTION WITH EXTERNAL DEVICES

CHAPTER 13. RESET

CHAPTER 14. CLOCK GENERATING CIRCUIT

CHAPTER 15. ELECTRICAL CHARACTERISTICS

CHAPTER 16. STANDARD CHARACTERISTICS

CHAPTER 17. APPLICATIONS

CHAPTER 18. PROM VERSION

CHAPTER 19. FLASH MEMORY VERSION

APPENDIX

MEMORANDUM

CHAPTER 1

DESCRIPTION

DESCRIPTION

The 16-bit single-chip microcomputers 7751 Group is suitable for office, business, and industrial equipment controllers that require high-speed processing of large amounts of data.
These microcomputers develop with the M37751M6C-XXXFP as the base chip. This manual describes the functions about the M37751M6C-XXXFP unless there is a specific difference and refers to the M37751M6C-XXXFP as “M37751.”

Notes 1: About details concerning each microcomputer’s development status of the 7751 Group, inquire of **“CONTACT ADDRESSES FOR FURTHER INFORMATION”** described last.
2: How the 7751 Group’s type name see is described below.

M 3 77 51 M 6 C–XXX FP

- M — Mitsubishi integrated prefix
- 3 — Represent an original single-chip microcomputer
- 77 — Series designation using 2 digits
- 51 — Circuit function identification code using 2 digits
- M — Memory identification code using a digit
  - M: Mask ROM
  - E: EPROM
  - F: Flash memory
  - S: External ROM
- 6 — Memory size identification code using a digit
- C — Difference of electrical characteristics identification code using a digit
- XXX — Mask ROM number
- FP — Package style
  - FP: Plastic molded QFP
  - FS: Ceramic QFN

1.1 Performance overview

Table 1.1.1 shows the performance overview of the M37751.

Table 1.1.1 M37751 performance overview

Parameters		Functions
Number of basic instructions		109
Instruction execution time		100 ns (the minimum instruction at $f(X_{IN})$ = 40 MHz)
Operating clock frequency $f(X_{IN})$		40 MHz (maximum at high-speed running)
Memory size	ROM	49152 bytes
	RAM	2048 bytes
Programmable Input/Output ports	P0–P2, P4–P8	8 bits × 8
	P3	4 bits × 1
Multifunction timers	TA0–TA4	16 bits × 5
	TB0–TB2	16 bits × 3
Serial I/O	UART0, UART1	(UART or clock synchronous serial I/O) × 2
A-D converter		10-bit successive approximation method × 1 (8 channels)
Watchdog timer		12 bits × 1
Interrupts		3 external, 16 internal (priority levels 0 to 7 can be set for each interrupt with software)
Clock generating circuit		Built-in (externally connected to a ceramic resonator or a quartz-crystal oscillator)
Supply voltage		5 V ±10 %
Power dissipation		125 mW (at $f(X_{IN})$ = 40 MHz frequency, typ.)
Port Input/Output characteristics	Input/Output withstand voltage	5 V
	Output current	5 mA
Memory expansion		Maximum 16 Mbytes
Operating temperature range		–20°C to 85°C
Device structure		CMOS high-performance silicon gate process
Package		80-pin plastic molded QFP

Note: All of the 7751 Group microcomputers are the same except for the package type, memory type, memory size, and electric characteristics.

1.2 Pin configuration

Figure 1.2.1 shows the M37751 pin configuration.

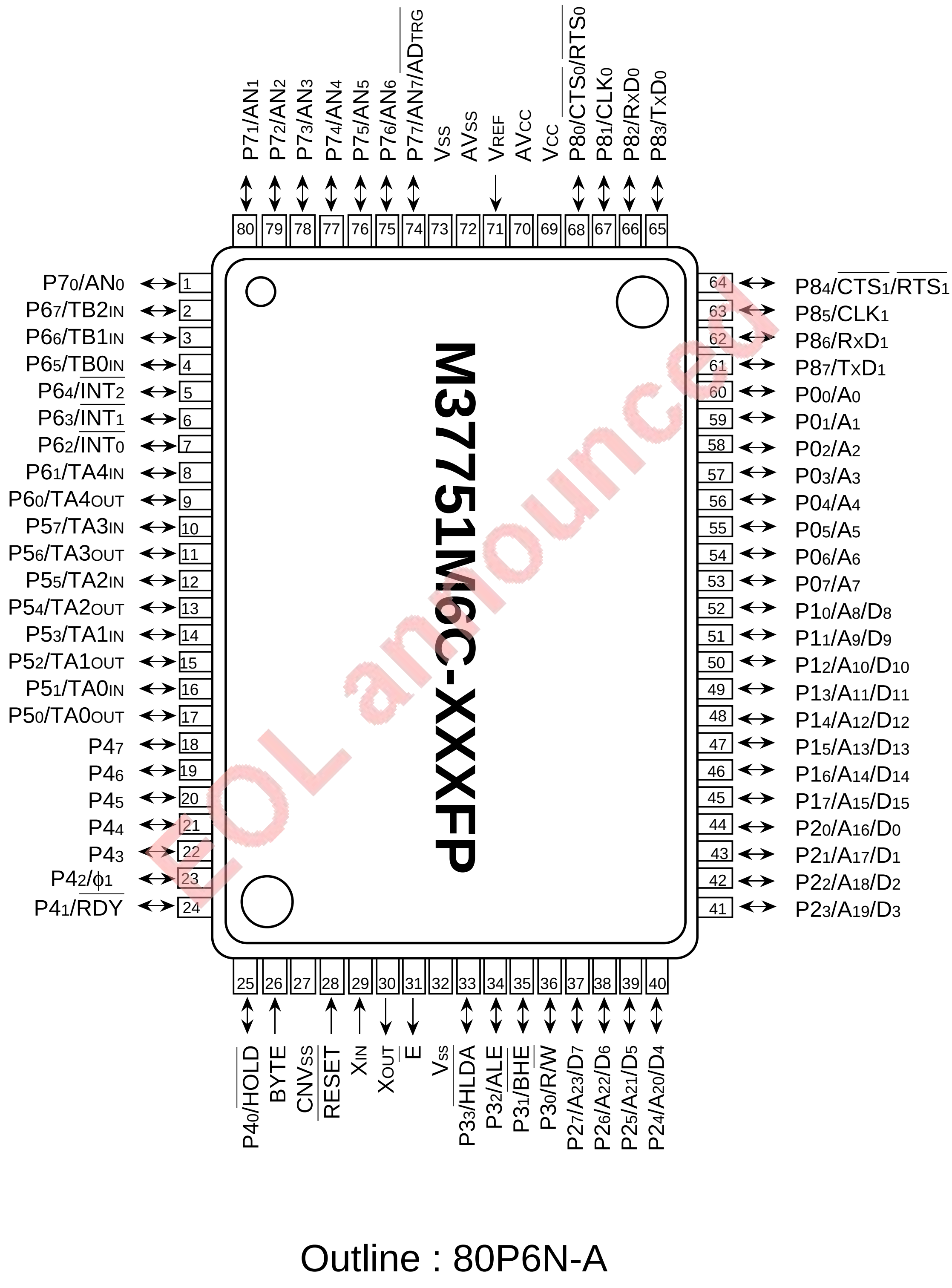


Fig. 1.2.1 M37751 pin configuration (top view)

1.3 Pin description

Tables 1.3.1 to 1.3.3 list the pin description.

Table 1.3.1 Pin description (1)

Pin	Name	Input/Output	Functions
Vcc, Vss	Power supply		Supply 5 V ±10 % to Vcc pin and 0 V to Vss pin.
CNVss	CNVss	Input	This pin controls the processor mode. **[Single-chip mode] [Memory expansion mode]** Connect to Vss pin. **[Microprocessor mode]** Connect to Vcc pin.
$\overline{\text{RESET}}$	Reset input	Input	The microcomputer is reset when supplying “L” level to this pin.
X_{IN}	Clock input	Input	These are I/O pins of the internal clock generating circuit. Connect a ceramic resonator or quartz-crystal oscillator between X_{IN} and X_{OUT} pins. When using an external clock, the clock source should be input to X_{IN} pin and X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{\text{E}}$	Enable output	Output	This pin outputs $\overline{\text{E}}$ signal. Data/instruction code read or data write is performed when output from this pin is “L” level.
BYTE	External data bus width selection input	Input	**[Memory expansion mode] [Microprocessor mode]** Input level to this pin determines whether the external data bus has a 16-bit width or 8-bit width. The width is 16 bits when the level is “L”, and 8 bits when the level is “H.”
AVcc	Analog supply input		The power supply pin for the A-D converter. Externally connect AVcc to Vcc pin.
AVss			The power supply pin for the A-D converter. Externally connect AVss to Vss pin.
V_{REF}	Reference voltage input	Input	This is a reference voltage input pin for the A-D converter.

1.3 Pin description

Table 1.3.2 Pin description (2)

Pin	Name	Input/Output	Functions
$P0_0$–$P0_7$	I/O port P0	I/O	**[Single-chip mode]** Port P0 is an 8-bit CMOS I/O port. This port has an I/O direction register and each pin can be programmed for input or output.
A_0–A_7		Output	**[Memory expansion mode] [Microprocessor mode]** Low-order 8 bits (A_0–A_7) of the address are output.
$P1_0$–$P1_7$	I/O port P1	I/O	**[Single-chip mode]** Port P1 is an 8-bit I/O port with the same function as P0.
A_8/D_8–A_{15}/D_{15}			**[Memory expansion mode] [Microprocessor mode]** ●External bus width = 8 bits (When the BYTE pin is "H" level) Middle-order 8 bits (A_8–A_{15}) of the address are output. ●External bus width = 16 bits (When the BYTE pin is "L" level) Data (D_8 to D_{15}) input/output and output of the middle-order 8 bits (A_8–A_{15}) of the address are performed with the time sharing system.
$P2_0$–$P2_7$	I/O port P2	I/O	**[Single-chip mode]** Port P2 is an 8-bit I/O port with the same function as P0.
A_{16}/D_0–A_{23}/D_7			**[Memory expansion mode] [Microprocessor mode]** Data (D_0 to D_7) input/output and output of the high-order 8 bits (A_{16}–A_{23}) of the address are performed with the time sharing system.
$P3_0$–$P3_3$	I/O port P3	I/O	**[Single-chip mode]** Port P3 is a 4-bit I/O port with the same function as P0.
R/$\overline{W}$, $\overline{BHE}$, ALE, $\overline{HLDA}$		Output	**[Memory expansion mode] [Microprocessor mode]** $P3_0$–$P3_3$ respectively output R/$\overline{W}$, $\overline{BHE}$, ALE, and $\overline{HLDA}$ signals. ●R/$\overline{W}$ The Read/Write signal indicates the data bus state. The state is read while this signal is "H" level, and write while this is "L" level. ●$\overline{BHE}$ "L" level is output when an odd-numbered address is accessed. ●ALE This is used to obtain only the address from address and data multiplex signals. ●$\overline{HLDA}$ This is the signal to externally indicate the state when the microcomputer is in Hold state. "L" level is output during Hold state.

Table 1.3.3 Pin description (3)

Pin	Name	Input/Output	Functions
P4$_0$–P4$_7$	I/O port P4	I/O	**[Single-chip mode]** Port P4 is an 8-bit I/O port with the same function as P0. P4$_2$ can be programmed as the clock ϕ_1 output pin.
$\overline{HOLD}$, $\overline{RDY}$, P4$_2$–P4$_7$		Input Input I/O	**[Memory expansion mode]** P4$_0$ functions as the $\overline{HOLD}$ input pin, P4$_1$ as the $\overline{RDY}$ input pin. The microcomputer is in Hold state while "L" level is input to the $\overline{HOLD}$ pin. The microcomputer is in Ready state while "L" level is input to the $\overline{RDY}$ pin. P4$_2$–P4$_7$ function as I/O ports with the same functions as P0. P4$_2$ can be programmed for the clock ϕ_1 output pin.
$\overline{HOLD}$, $\overline{RDY}$, ϕ_1, P4$_3$–P4$_7$		Input Input Output I/O	**[Microprocessor mode]** P4$_0$ functions as the $\overline{HOLD}$ input pin, P4$_1$ as the $\overline{RDY}$ input pin. P4$_2$ always functions as the clock ϕ_1 output pin. P4$_3$–P4$_7$ function as I/O ports with the same functions as P0.
P5$_0$–P5$_7$	I/O port P5	I/O	Port P5 is an 8-bit I/O port with the same function as P0. These pins can be programmed as I/O pins for Timers A0–A3.
P6$_0$–P6$_7$	I/O port P6	I/O	Port P6 is an 8-bit I/O port with the same function as P0. These pins can be programmed as I/O pins for Timer A4, input pins for external interrupt and input pins for Timers B0–B2.
P7$_0$–P7$_7$	I/O port P7	I/O	Port P7 is an 8-bit I/O port with the same function as P0. These pins can be programmed as input pins for A-D converter.
P8$_0$–P8$_7$	I/O port P8	I/O	Port P8 is an 8-bit I/O port with the same function as P0. These pins can be programmed as I/O pins for serial I/O.

1.4 Block diagram

Figure 1.4.1 shows the M37751 block diagram.

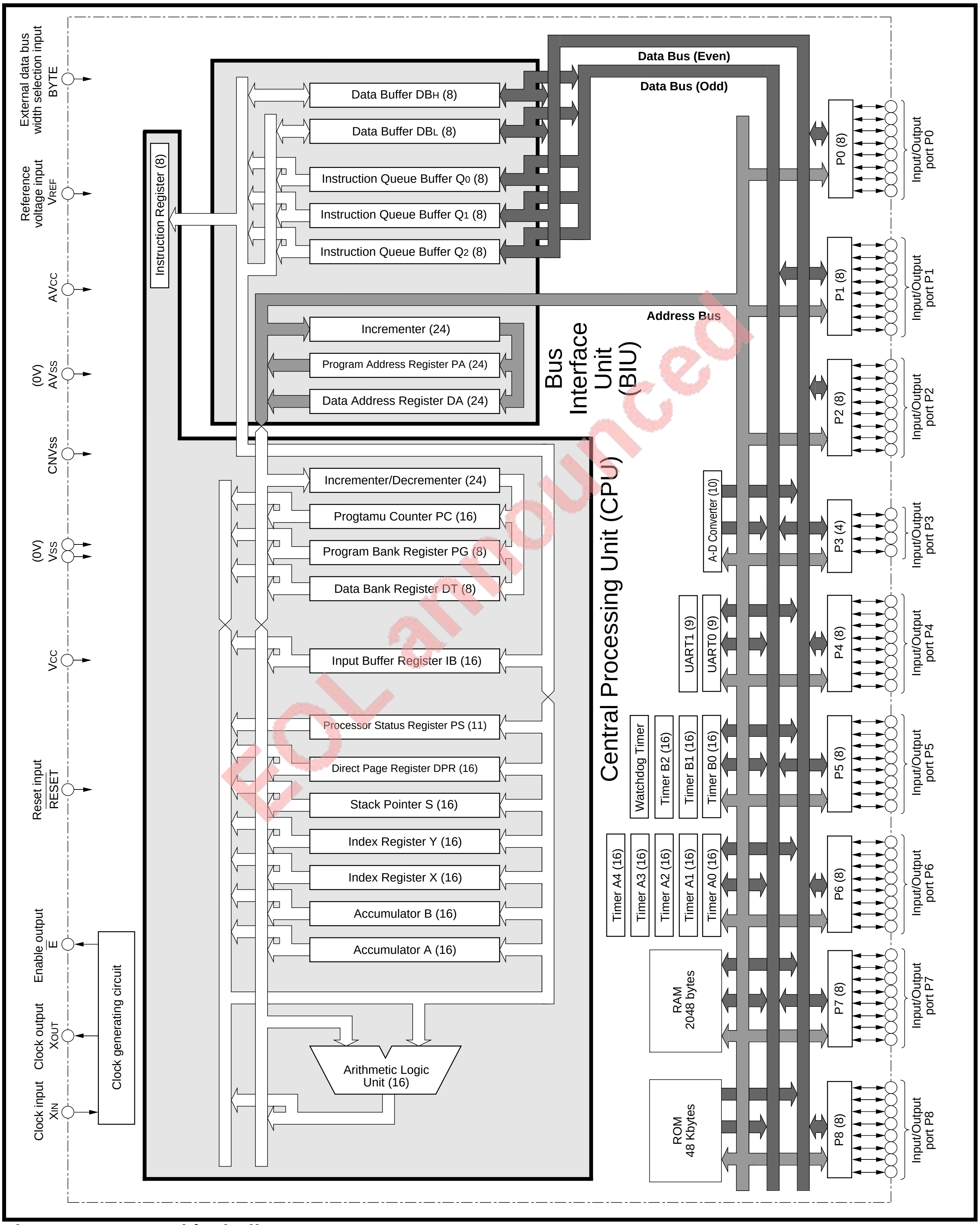


Fig. 1.4.1 M37751 block diagram

CHAPTER 2

CENTRAL PROCESSING UNIT (CPU)

2.1 Central processing unit

The CPU (Central Processing Unit) has the ten registers as shown in Figure 2.1.1.

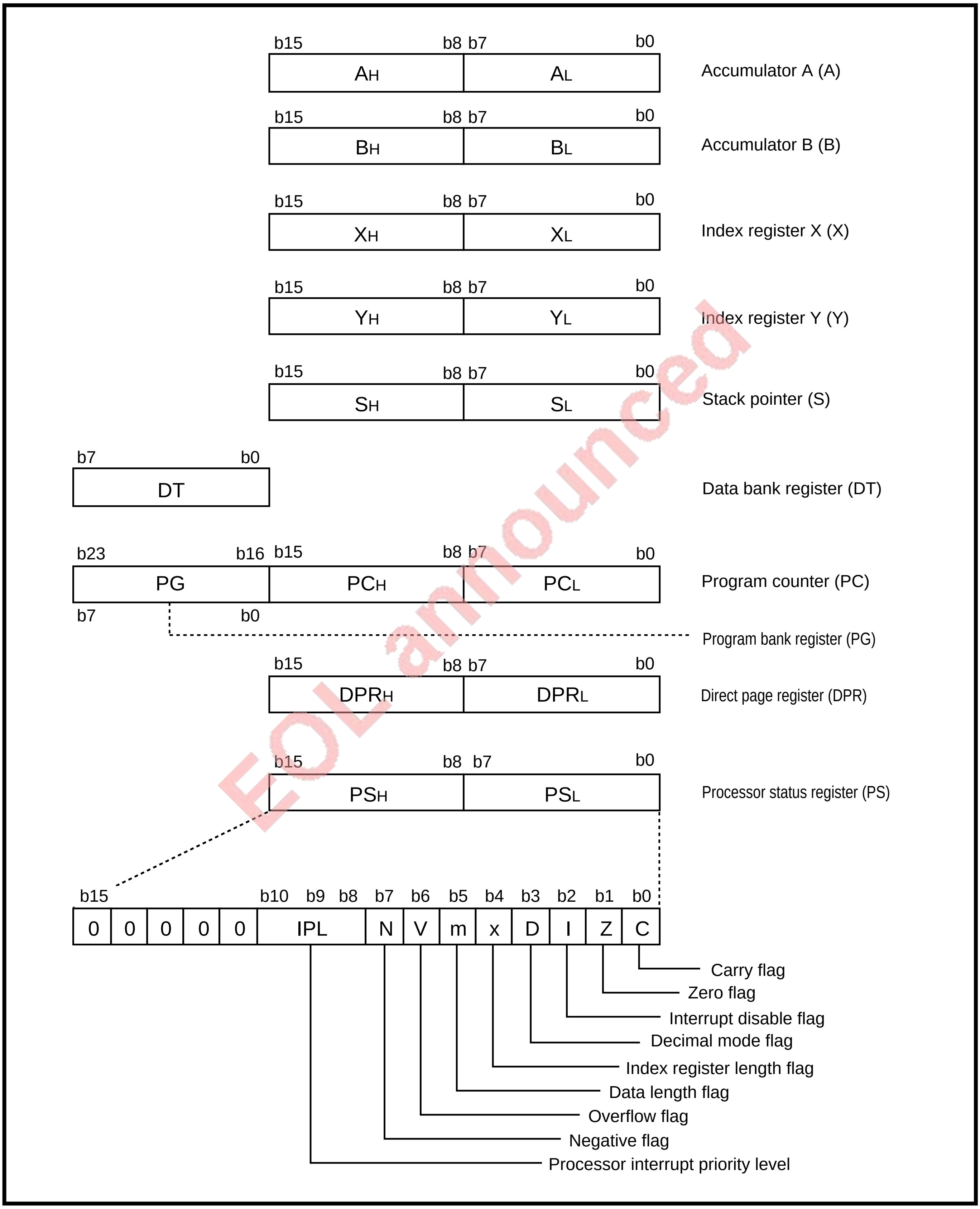


Fig. 2.1.1 CPU registers structure

2.1.1 Accumulator (Acc)
Accumulators A and B are available.

(1) Accumulator A (A)
Accumulator A is the main register of the microcomputer. The transaction of data such as calculation, data transfer, and input/output are performed mainly through accumulator A. It consists of 16 bits, and the low-order 8 bits can also be used separately. The data length flag (m) determines whether the register is used as a 16-bit register or as an 8-bit register. Flag m is a part of the processor status register which is described later. When an 8-bit register is selected, only the low-order 8 bits of accumulator A are used and the contents of the high-order 8 bits is unchanged.

(2) Accumulator B (B)
Accumulator B is a 16-bit register with the same function as accumulator A. Accumulator B can be used instead of accumulator A. The use of accumulator B, however except for some instructions, requires more instruction bytes and execution cycles than that of accumulator A. Accumulator B is also controlled by the data length flag (m) just as in accumulator A.

2.1.2 Index register X (X)
Index register X consists of 16 bits and the low-order 8 bits can also be used separately. The index register length flag (x) determines whether the register is used as a 16-bit register or as an 8-bit register. Flag x is a part of the processor status register which is described later. When an 8-bit register is selected, only the low-order 8 bits of index register X are used and the contents of the high-order 8 bits is unchanged. In an addressing mode in which index register X is used as an index register, the address obtained by adding the contents of this register to the operand's contents is accessed.

In the **MVP** or **MVN** instruction, a block transfer instruction, the contents of index register X indicate the low-order 16 bits of the source address. The third byte of the instruction is the high-order 8 bits of the source address.
In the **RMPA** instruction, a Repeat MultiPly and Accumulate instruction, the contents of index register X indicate the low-order 16 bits of address in which multiplicands are stored.

Note: Refer to **"7751 Series Software Manual"** for addressing modes.

2.1.3 Index register Y (Y)
Index register Y is a 16-bit register with the same function as index register X. Just as in index register X, the index register length flag (x) determines whether this register is used as a 16-bit register or as an 8-bit register.
In the **MVP** or **MVN** instruction, a block transfer instruction, the contents of index register Y indicate the low-order 16 bits of the destination address. The second byte of the instruction is the high-order 8 bits of the destination address.
In the **RMPA** instruction, a Repeat MultiPly and Accumulate instruction, the contents of index register Y indicate the low-order 16 bits of address in which multipliers are stored.

2.1.4 Stack pointer (S)

The stack pointer (S) is a 16-bit register. It is used for a subroutine call or an interrupt. It is also used when addressing modes using the stack are executed. The contents of S indicate an address (stack area) for storing registers during subroutine calls and interrupts. Bank 0_{16} is specified for the stack area. (Refer to **"2.1.6 Program bank register (PG)."**)

When an interrupt request is accepted, the microcomputer stores the contents of the program bank register (PG) at the address indicated by the contents of S and decrements the contents of S by 1. Then the contents of the program counter (PC) and the processor status register (PS) are stored. The contents of S after accepting an interrupt request is equal to the contents of S decremented by 5 before the accepting of the interrupt request. (Refer to Figure 2.1.2.)

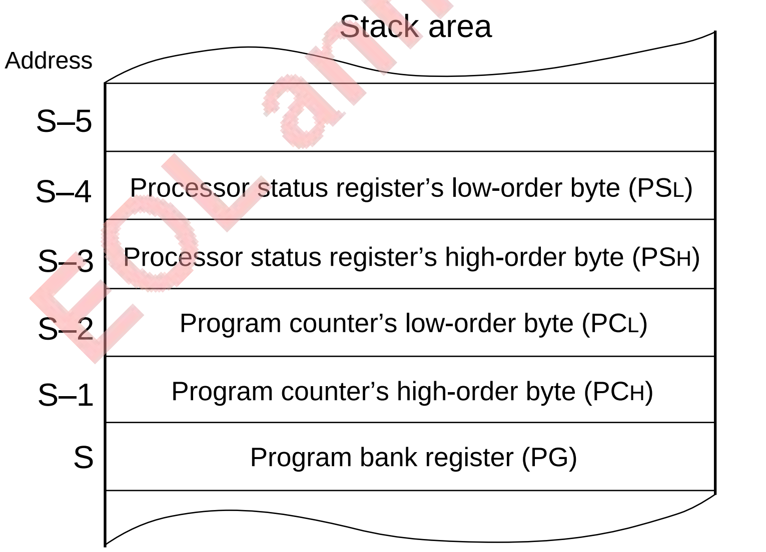
When completing the process in the interrupt routine and returning to the original routine, the contents of registers stored in the stack area are restored into the original registers in the reverse sequence (PS→PC→PG) by executing the **RTI** instruction. The contents of S is returned to the state before accepting an interrupt request.

The same operation is performed during a subroutine call, however, the contents of PS is not automatically stored. (The contents of PG may not be stored. This depends on the addressing mode.)

The user should store registers other than those described above with software when the user needs them during interrupts or subroutine calls.

Additionally, initialize S at the beginning of the program because its contents are undefined at reset. The stack area changes when subroutines are nested or when multiple interrupt requests are accepted. Therefore, make sure of the subroutine's nesting depth not to destroy the necessary data.

Note: Refer to **"7751 Series Software Manual"** for addressing modes.



● "S" is the initial address that the stack pointer (S) indicates at accepting an interrupt request.
The S's contents become "S–5" after storing the above registers.

Fig. 2.1.2 Stored registers of the stack area

2.1.5 Program counter (PC)

The program counter is a 16-bit counter that indicates the low-order 16 bits of the address (24 bits) at which an instruction to be executed next (in other words, an instruction to be read out from an instruction queue buffer next) is stored. The contents of the high-order program counter (PC_H) become "FF_{16}," and the low-order program counter (PC_L) becomes "FE_{16}" at reset. The contents of the program counter becomes the contents of the reset's vector address (addresses $FFFE_{16}$, $FFFF_{16}$) immediately after reset.
Figure 2.1.3 shows the program counter and the program bank register.

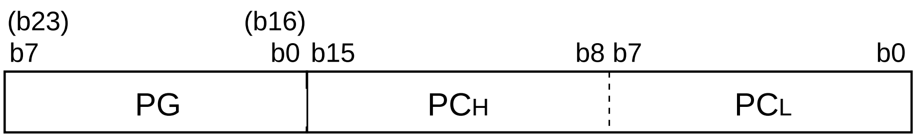


Fig. 2.1.3 Program counter and program bank register

2.1.6 Program bank register (PG)

The program bank register is an 8-bit register. This register indicates the high-order 8 bits (bank) of the address (24 bits) at which an instruction to be executed next (in other words, an instruction to be read out from an instruction queue buffer next) is stored. These 8 bits are called bank.
When a carry occurs after adding the contents of the program counter or adding the offset value to the contents of the program counter in the branch instruction and others, the contents of the program bank register is automatically incremented by 1. When a borrow occurs after subtracting the contents of the program counter, the contents of the program bank register is automatically decremented by 1. Accordingly, there is no need to consider bank boundaries in programming, usually.

In the single-chip mode, make sure to prevent the program bank register from being set to the value other than "00_{16}" by executing the branch instructions and others. It is because the access space of the single-chip mode is the internal area within the bank 0_{16}.
This register is cleared to "00_{16}" at reset.

2.1.7 Data bank register (DT)

The data bank register is an 8-bit register. In the following addressing modes using the data bank register, the contents of this register is used as the high-order 8 bits (bank) of a 24-bit address to be accessed. Use the **LDT** instruction to set a value to this register.

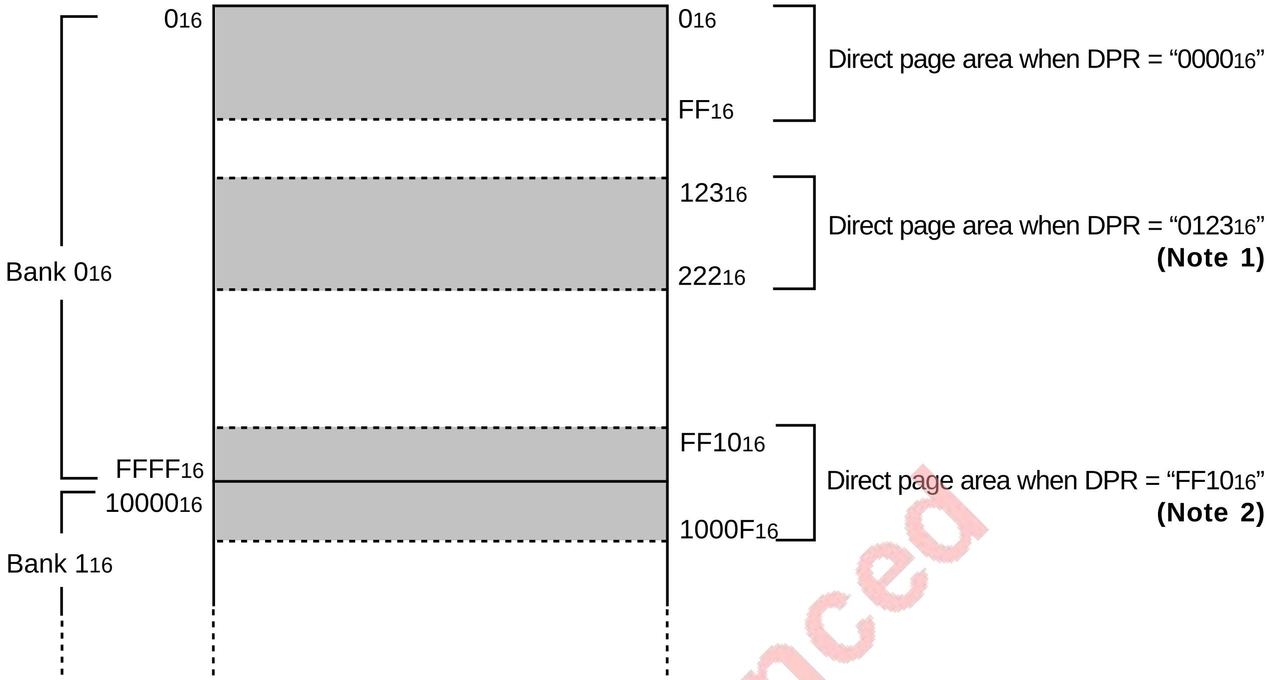
In the single-chip mode, make sure to fix this register to “00_{16}”. It is because the access space of the single-chip mode is the internal area within the bank 0_{16}.
This register is cleared to “00_{16}” at reset.

●Addressing modes using data bank register
- Direct indirect
- Direct indexed X indirect
- Direct indirect indexed Y
- Absolute
- Absolute bit
- Absolute indexed X
- Absolute indexed Y
- Absolute bit relative
- Stack pointer relative indirect indexed Y
- Multiplied accumulation

2.1.8 Direct page register (DPR)

The direct page register is a 16-bit register. The contents of this register indicate the direct page area which is allocated in bank 0_{16} or in the space across banks 0_{16} and 1_{16}. The following addressing modes use the direct page register.
The contents of the direct page register indicate the base address (the lowest address) of the direct page area. The space which extends to 256 bytes above that address is specified as a direct page.
The direct page register can contain a value from “0000_{16}” to “$FFFF_{16}$.” When it contains a value equal to or more than “$FF01_{16}$,” the direct page area spans the space across banks 0_{16} and 1_{16}.
When the contents of low-order 8 bits of the direct page register is “00_{16},” the number of cycles required to generate an address is 1 cycle smaller than the number when its contents are not “00_{16}.” Accordingly, the access efficiency can be enhanced in this case.
This register is cleared to “0000_{16}” at reset.
Figure 2.1.4 shows a setting example of the direct page area.

●Addressing modes using direct page register
- Direct
- Direct bit
- Direct indexed X
- Direct indexed Y
- Direct indirect
- Direct indexed X indirect
- Direct indirect indexed Y
- Direct indirect long
- Direct indirect long indexed Y
- Direct bit relative



Notes 1 : The number of cycles required to generate an address is 1 cycle smaller when the low-order 8 bits of the DPR are "00_{16}."

2 : The direct page area spans the space across banks 0_{16} and 1_{16} when the DPR is "$FF01_{16}$" or more.

Fig. 2.1.4 Setting example of direct page area

2.1.9 Processor status register (PS)

The processor status register is an 11-bit register.
Figure 2.1.5 shows the structure of the processor status register.

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	IPL			N	V	m	x	D	I	Z	C

Processor status register (PS)

Note: Fix bits 11–15 to “0.”

Fig. 2.1.5 Processor status register structure

(1) Bit 0: Carry flag (C)

It retains a carry or a borrow generated in the arithmetic and logic unit (ALU) during an arithmetic operation. This flag is also affected by shift and rotate instructions. When the **BCC** or **BCS** instruction is executed, this flag’s contents determine whether the program causes a branch or not.
Use the **SEC** or **SEP** instruction to set this flag to “1,” and use the **CLC** or **CLP** instruction to clear it to “0.”

(2) Bit 1: Zero flag (Z)

It is set to “1” when a result of an arithmetic operation or data transfer is “0,” and cleared to “0” when otherwise. When the **BNE** or **BEQ** instruction is executed, this flag’s contents determine whether the program causes a branch or not.
Use the **SEP** instruction to set this flag to “1,” and use the **CLP** instruction to clear it to “0.”

Note: This flag is invalid in the decimal mode addition (the **ADC** instruction).

(3) Bit 2: Interrupt disable flag (I)

It disables all maskable interrupts (interrupts other than watchdog timer, the **BRK** instruction, and zero division). Interrupts are disabled when this flag is “1.” When an interrupt request is accepted, this flag is automatically set to “1” to avoid multiple interrupts. Use the **SEI** or **SEP** instruction to set this flag to “1,” and use the **CLI** or **CLP** instruction to clear it to “0.” This flag is set to “1” at reset.

(4) Bit 3: Decimal mode flag (D)

It determines whether addition and subtraction are performed in binary or decimal. Binary arithmetic is performed when this flag is “0.” When it is “1,” decimal arithmetic is performed with each word treated as two or four digits decimal (determined by the data length flag). Decimal adjust is automatically performed. Decimal operation is possible only with the **ADC** and **SBC** instructions. Use the **SEP** instruction to set this flag to “1,” and use the **CLP** instruction to clear it to “0.” This flag is cleared to “0” at reset.

(5) Bit 4: Index register length flag (x)

It determines whether each of index register X and index register Y is used as a 16-bit register or an 8-bit register. That register is used as a 16-bit register when this flag is “0,” and as an 8-bit register when it is “1.” Use the **SEP** instruction to set this flag to “1,” and use the **CLP** instruction to clear it to “0.” This flag is cleared to “0” at reset.

Note: When transferring data between registers which are different in bit length, the data is transferred with the length of the destination register, but except for the **TXA**, **TYA**, **TXB**, **TYB**, and **TXS** instructions. Refer to **“7751 Series Software Manual”** for details.

(6) Bit 5: Data length flag (m)

It determines whether to use a data as a 16-bit unit or as an 8-bit unit. A data is treated as a 16-bit unit when this flag is “0,” and as an 8-bit unit when it is “1.”
Use the **SEM** or **SEP** instruction to set this flag to “1,” and use the **CLM** or **CLP** instruction to clear it to “0.” This flag is cleared to “0” at reset.

Note: When transferring data between registers which are different in bit length, the data is transferred with the length of the destination register, but except for the **TXA**, **TYA**, **TXB**, **TYB**, and **TXS** instructions. Refer to **“7751 Series Software Manual”** for details.

(7) Bit 6: Overflow flag (V)

It is used when adding or subtracting with a word regarded as signed binary. When the data length flag (m) is “0,” the overflow flag is set to “1” when the result of addition or subtraction exceeds the range between –32768 and +32767, and cleared to “0” in all other cases. When the data length flag (m) is “1,” the overflow flag is set to “1” when the result of addition or subtraction exceeds the range between –128 and +127, and cleared to “0” in all other cases.
The overflow flag is also set to “1” when a result of division exceeds the register length to be stored in the **DIV** or **DIVS** instruction, a division instruction with unsigned or signed; and when a result of addition exceeds the range between –2147483648 and +2147483647 in the **RMPA** instruction, a Repeat MultiPly and Accumulate instruction.
When the **BVC** or **BVS** instruction is executed, this flag’s contents determine whether the program causes a branch or not.
Use the **SEP** instruction to set this flag to “1,” and use the **CLV** or **CLP** instruction to clear it to “0.”

Note: This flag is invalid in the decimal mode.

(8) Bit 7: Negative flag (N)

It is set to “1” when a result of arithmetic operation or data transfer is negative. (Bit 15 of the result is “1” when the data length flag (m) is “0,” or bit 7 of the result is “1” when the data length flag (m) is “1.”) It is cleared to “0” in all other cases. When the **BPL** or **BMI** instruction is executed, this flag determines whether the program causes a branch or not. Use the **SEP** instruction to set this flag to “1,” and use the **CLP** instruction to clear it to “0.”

Note: This flag is invalid in the decimal mode.

(9) Bits 10 to 8: Processor interrupt priority level (IPL)

These three bits can determine the processor interrupt priority level to one of levels 0 to 7. The interrupt is enabled when the interrupt priority level of a required interrupt, which is set in each interrupt control register, is higher than IPL. When an interrupt request is accepted, IPL is stored in the stack area, and IPL is replaced by the interrupt priority level of the accepted interrupt request. There are no instruction to directly set or clear the bits of IPL. IPL can be changed by storing the new IPL into the stack area and updating the processor status register with the **PUL** or **PLP** instruction. The contents of IPL is cleared to “000_2” at reset.

2.2 Bus interface unit

A bus interface unit (BIU) is built-in between the central processing unit (CPU) and memory•I/O devices. BIU's function and operation are described below.
When externally connecting devices, refer to **"Chapter 12. CONNECTION WITH EXTERNAL DEVICES."**

2.2.1 Overview

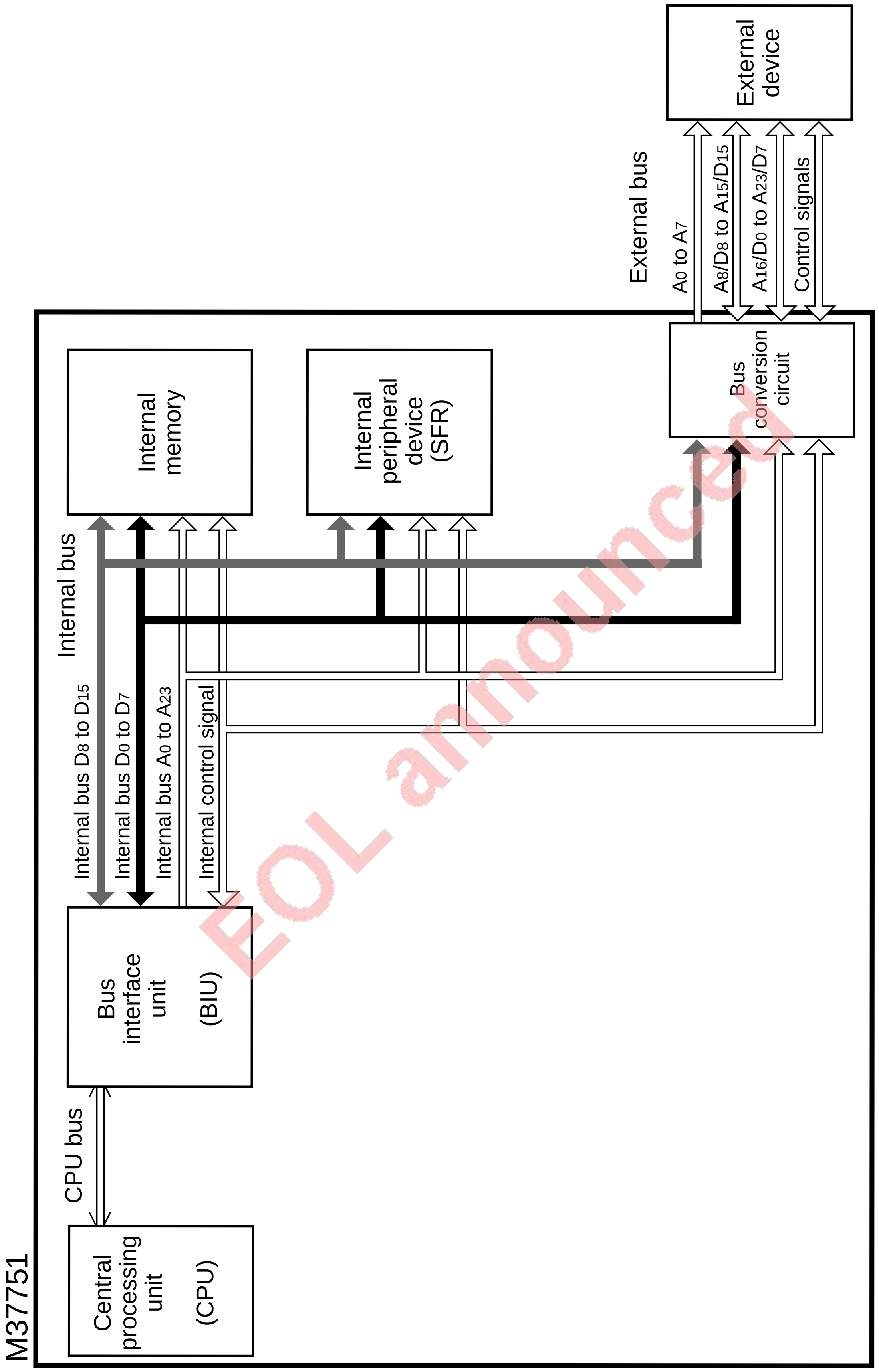
Transfer operation between the CPU and memory•I/O devices is always performed via the BIU.
Figure 2.2.1 shows the bus and bus interface unit (BIU).

① The BIU reads an instruction from the memory before the CPU executes it.

② When the CPU reads data from the memory•I/O device, the CPU first specifies the address from which data is read to the BIU. The BIU reads data from the specified address and passes it to the CPU.

③ When the CPU writes data to the memory•I/O device, the CPU first specifies the address to which data is written to the BIU and write data. The BIU writes the data to the specified address.

④ To perform the above operations ① to ③, the BIU inputs and outputs the control signals, and control the bus.



SFR : Special Function Register

Notes 1: The CPU bus, internal bus, and external bus are independent of one another.
2: Refer to **"Chapter 12. CONNECTION WITH EXTERNAL DEVICES"** about control signals of the external bus.

Fig. 2.2.1 Bus and bus interface unit (BIU)

2.2.2 Functions of bus interface unit (BIU)

The bus interface unit (BIU) consists of four registers shown in Figure 2.2.2. Table 2.2.1 lists the functions of each register.

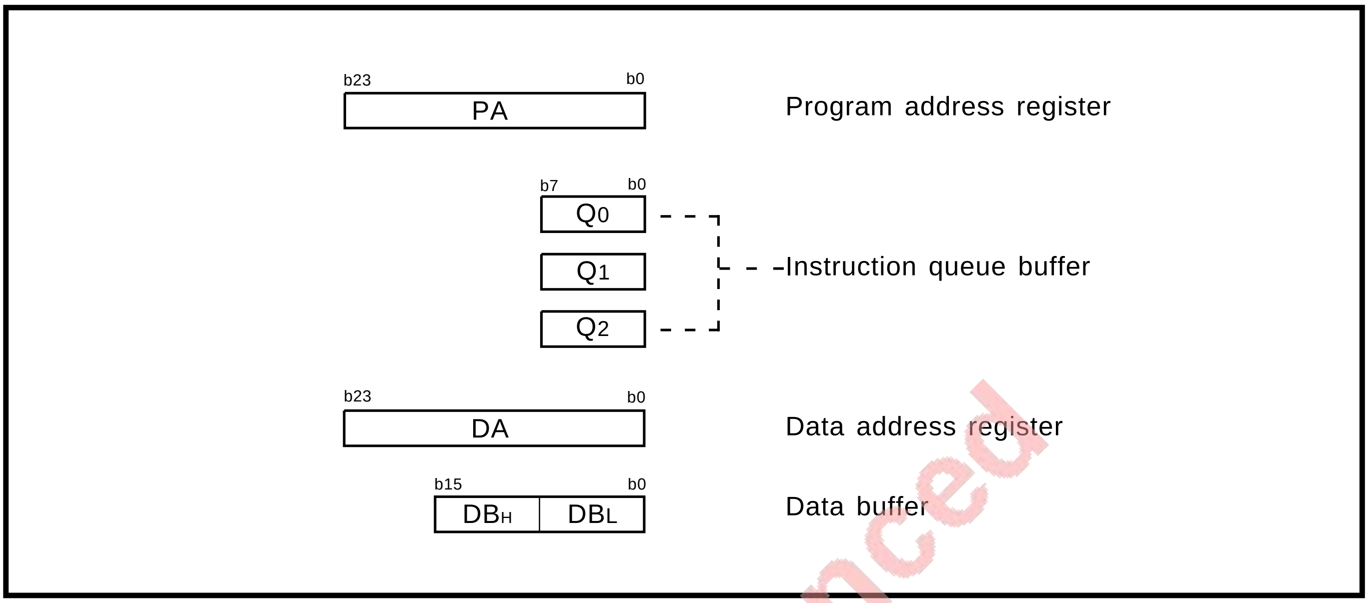


Fig. 2.2.2 Register structure of bus interface unit (BIU)

Table 2.2.1 Functions of each register

Name	Functions
Program address register	Indicates the storage address for the instruction which is next taken into the instruction queue buffer.
Instruction queue buffer	Temporarily stores the instruction which has been taken in.
Data address register	Indicates the address for the data which is next read from or written to.
Data buffer	Temporarily stores the data which is read from the memory•I/O device by the BIU or which is written to the memory•I/O device by the CPU.

The CPU and the bus send or receive data via BIU because each operates based on different clocks **(Note)**. The BIU allows the CPU to operate at high speed without waiting for access to the memory • I/O devices that require a long access time.
The BIU's functions are described bellow.

Note: The CPU operates based on ϕCPU. The period of ϕCPU is normally the same as that of ϕ. The internal bus operates based on the $\overline{E}$ signal. The period of the $\overline{E}$ signal is twice that of ϕ at a minimum.

(1) Reading out instruction (Instruction prefetch)

When the CPU does not require to read or write data, that is, when the bus is not in use, the BIU reads instructions from the memory and stores them in the instruction queue buffer. This is called instruction prefetch.
The CPU reads instructions from the instruction queue buffer and executes them, so that the CPU can operate at high speed without waiting for access to the memory which requires a long access time.
When the instruction queue buffer becomes empty or contains only 1 byte of an instruction, the BIU performs instruction prefetch. The instruction queue buffer can store instructions up to 3 bytes.
The contents of the instruction queue buffer is initialized when a branch or jump instruction is executed, and the BIU reads a new instruction from the destination address.
When instructions in the instruction queue buffer are insufficient for the CPU's needs, the BIU extends the pulse duration of clock ϕCPU in order to keep the CPU waiting until the BIU fetches the required number of instructions or more.

(2) Reading data from memory•I/O device

The CPU specifies the storage address of data to be read to the BIU's data address register, and requires data. The CPU waits until data is ready in the BIU.
The BIU outputs the address received from the CPU onto the address bus, reads contents at the specified address, and takes it into the data buffer.
The CPU continues processing, using data in the data buffer.
However, if the BIU uses the bus for instruction prefetch when the CPU requires to read data, the BIU keeps the CPU waiting.

(3) Writing data to memory•I/O device

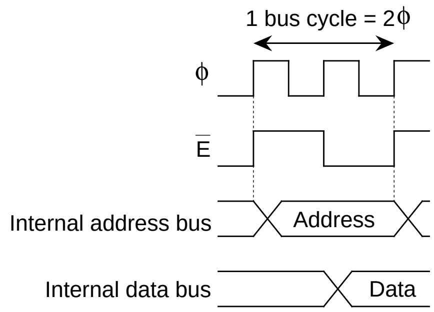
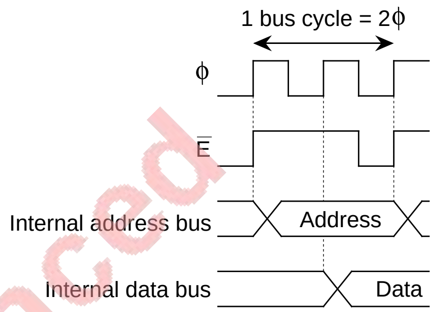
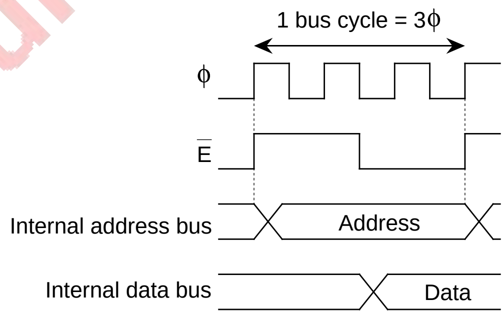
The CPU specifies the address of data to be written to the BIU's data address register. Then, the CPU writes data into the data buffer. The BIU outputs the address received from the CPU onto the address bus and writes data in the data buffer into the specified address.
The CPU advances to the next processing without waiting for completion of BIU's write operation.
However, if the BIU uses the bus for instruction prefetch when the CPU requires to write data, the BIU keeps the CPU waiting.

(4) Bus control

To perform the above operations (1) to (3), the BIU inputs and outputs the control signals, and controls the address bus and the data bus. The cycle in which the BIU controls the bus and accesses the memory•I/O device is called the bus cycle. Table 2.2.2 shows the bus cycle at accessing the internal area.

Refer to "**Chapter 12. CONNECTION WITH EXTERNAL DEVICES**" about the bus cycle at accessing the external devices.

Table 2.2.2 Bus cycle at accessing internal area

<table>
<tr><th></th><th>In low-speed running (f(XIN) ≤ 25 MHz)</th><th>In high-speed running (f(XIN) ≤ 40 MHz)</th></tr>
<tr><td>RAM</td><td rowspan="3">
</td><td>
</td></tr>
<tr><td>ROM</td><td rowspan="2">
</td></tr>
<tr><td>SFR</td></tr>
</table>

2.2.3 Operation of bus interface unit (BIU)

Figure 2.2.3 shows the basic operating waveforms of the bus interface unit (BIU).
About signals which are input/output externally when accessing external devices, refer to **"Chapter 12. CONNECTION WITH EXTERNAL DEVICES."**

(1) When fetching instructions into the instruction queue buffer

① When the instruction which is next fetched is located at an even address, the BIU fetches 2 bytes at a time with the timing of waveform (a).
However, when accessing an external device which is connected with the 8-bit external data bus width (BYTE = "H"), only 1 byte is fetched.

② When the instruction which is next fetched is located at an odd address, the BIU fetches only 1 byte with the timing of waveform (a). The contents at the even address are not taken.

(2) When reading or writing data to and from the memory•I/O device

① When accessing a 16-bit data which begins at an even address, waveform (a) is applied. The 16 bits of data are accessed at a time.

② When accessing a 16-bit data which begins at an odd address, waveform (b) is applied. The 16 bits of data are accessed separately in 2 operations, 8 bits at a time. Invalid data is not fetched into the data buffer.

③ When accessing an 8-bit data at an even address, waveform (a) is applied. The data at the odd address is not fetched into the data buffer.

④ When accessing an 8-bit data at an odd address, waveform (a) is applied. The data at the even address is not fetched into the data buffer.

For instructions that are affected by the data length flag (m) and the index register length flag (x), operation ① or ② is applied when flag m or x = "0"; operation ③ or ④ is applied when flag m or x = "1."

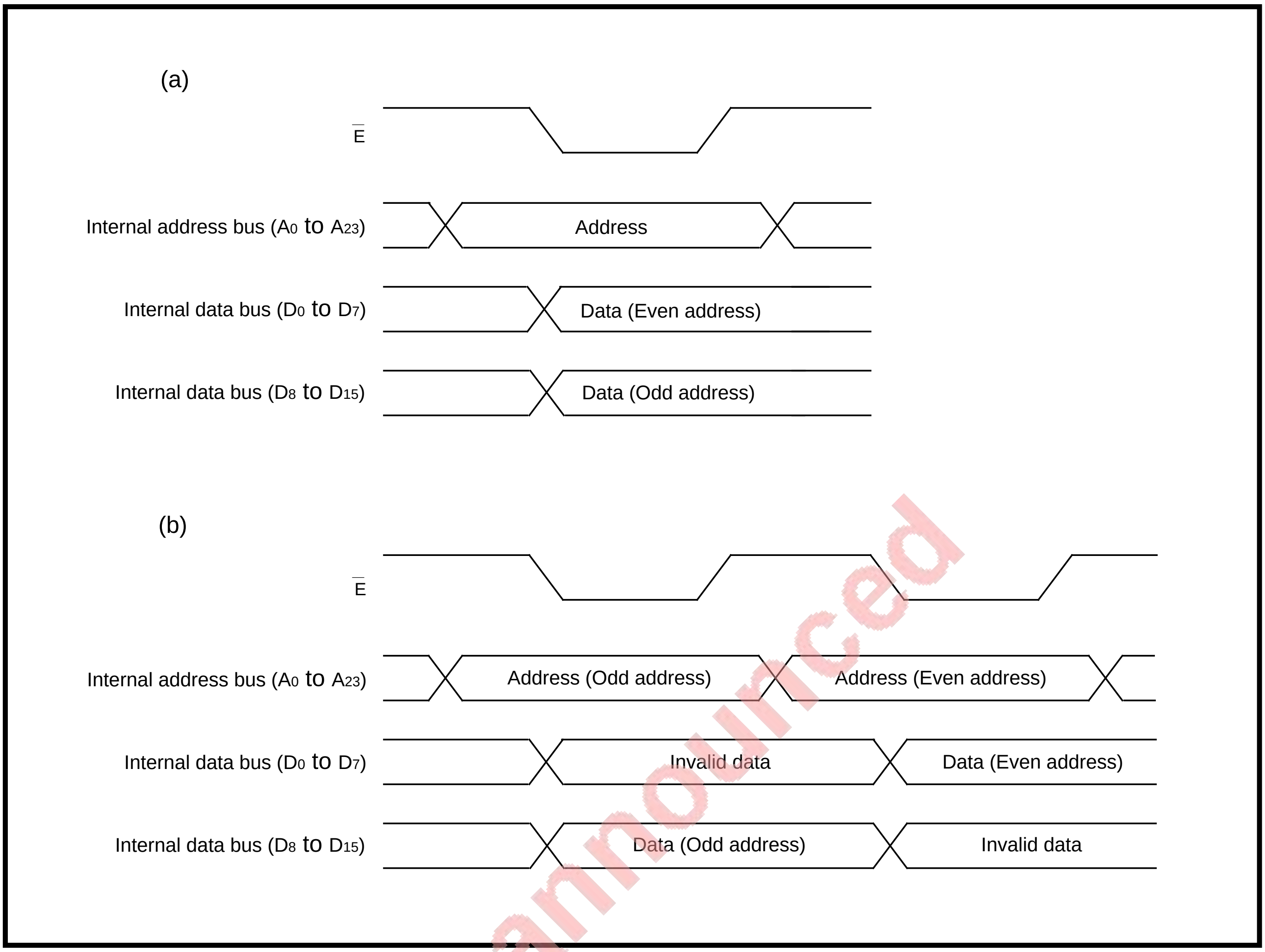


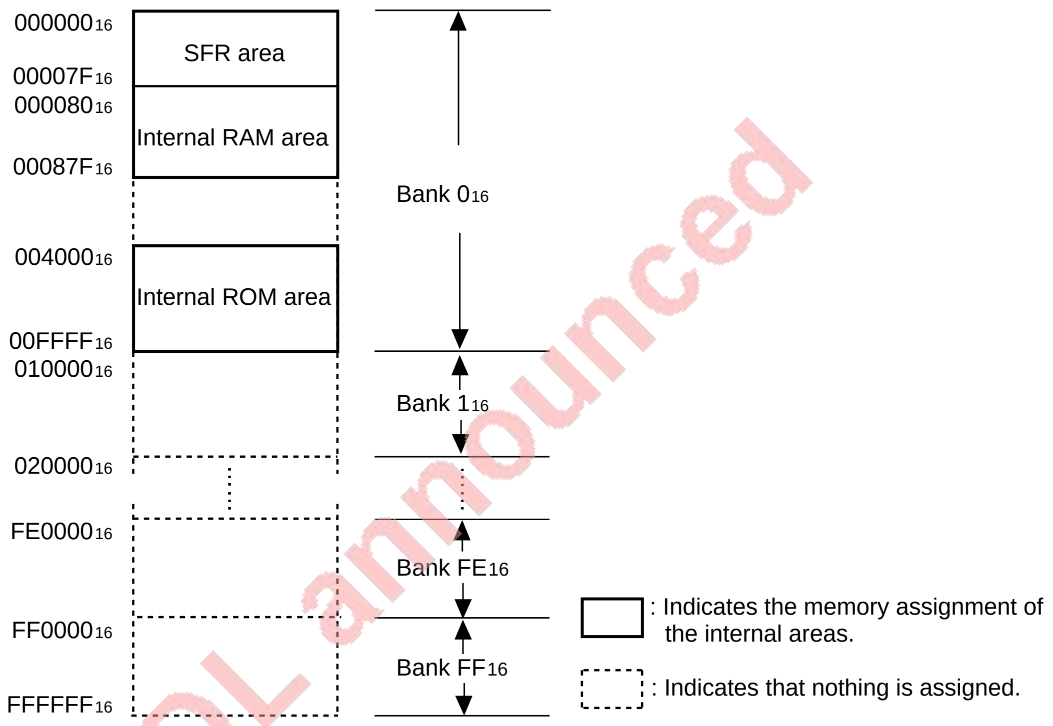
Fig. 2.2.3 Basic operating waveforms of bus interface unit (BIU)

2.3 Access space

Figure 2.3.1 shows the M37751's access space.

By combination of the program counter (PC), which is 16 bits of structure, and the program bank register (PG), a 16-Mbyte space from addresses 0_{16} to $FFFFFF_{16}$ can be accessed. For details about access of an external area, refer to **"Chapter 12. CONNECTION WITH EXTERNAL DEVICES."**

The memory and I/O devices are allocated in the same access space. Accordingly, it is possible to perform transfer and arithmetic operations using the same instructions without discrimination of the memory from I/O devices.



Note : Memory assignment of internal area varies according to the type of microcomputer. This figure shows the case of the M37751M6C-XXXFP.
Refer to **"Appendix 1. Memory assignment"** for other products.

SFR : Special Function Register

Fig. 2.3.1 M37751's access space

2.3 Access space

2.3.1 Banks

The access space is divided in units of 64 Kbytes. This unit is called “bank.” The high-order 8 bits of address (24 bits) indicate a bank, which is specified by the program bank register (PG) or data bank register (DT). Each bank can be accessed efficiently by using an addressing mode that uses the data bank register (DT).

If the program counter (PC) overflows at a bank boundary, the contents of the program bank register (PG) is incremented by 1. If a borrow occurs in the program counter (PC) as a result of subtraction, the contents of the program bank register (PG) is decremented by 1. Normally, accordingly, the user can program without concern for bank boundaries.

SFR (Special Function Register), internal RAM, and internal ROM are assigned in bank 0_{16}. For details, refer to section **“2.4 Memory assignment.”**

2.3.2 Direct page

A 256-byte space specified by the direct page register (DPR) is called “direct page.” A direct page is specified by setting the base address (the lowest address) of the area to be specified as a direct page into the direct page register (DPR).

By using a direct page addressing mode, a direct page can be accessed with less instruction cycles than otherwise.

Note: Refer also to section **“2.1 Central processing unit.”**

2.4 Memory assignment

This section describes the internal area's memory assignment. For more information about the external area, refer also to section **"2.5 Processor modes."**

2.4.1 Memory assignment in internal area

SFR (Special Function Register), internal RAM, and internal ROM are assigned in the internal area. Figure 2.4.1 shows the internal area's memory assignment.

(1) SFR area

The registers for setting internal peripheral devices are assigned at addresses 0_{16} to $7F_{16}$. This area is called SFR (Special Function Register). Figure 2.4.2 shows the SFR area's memory assignment. For each register in the SFR area, refer to each functional description in this manual.
For the state of the SFR area immediately after a reset, refer to section **"13.1.2 State of CPU, SFR area, and internal RAM area."**

(2) Internal RAM area

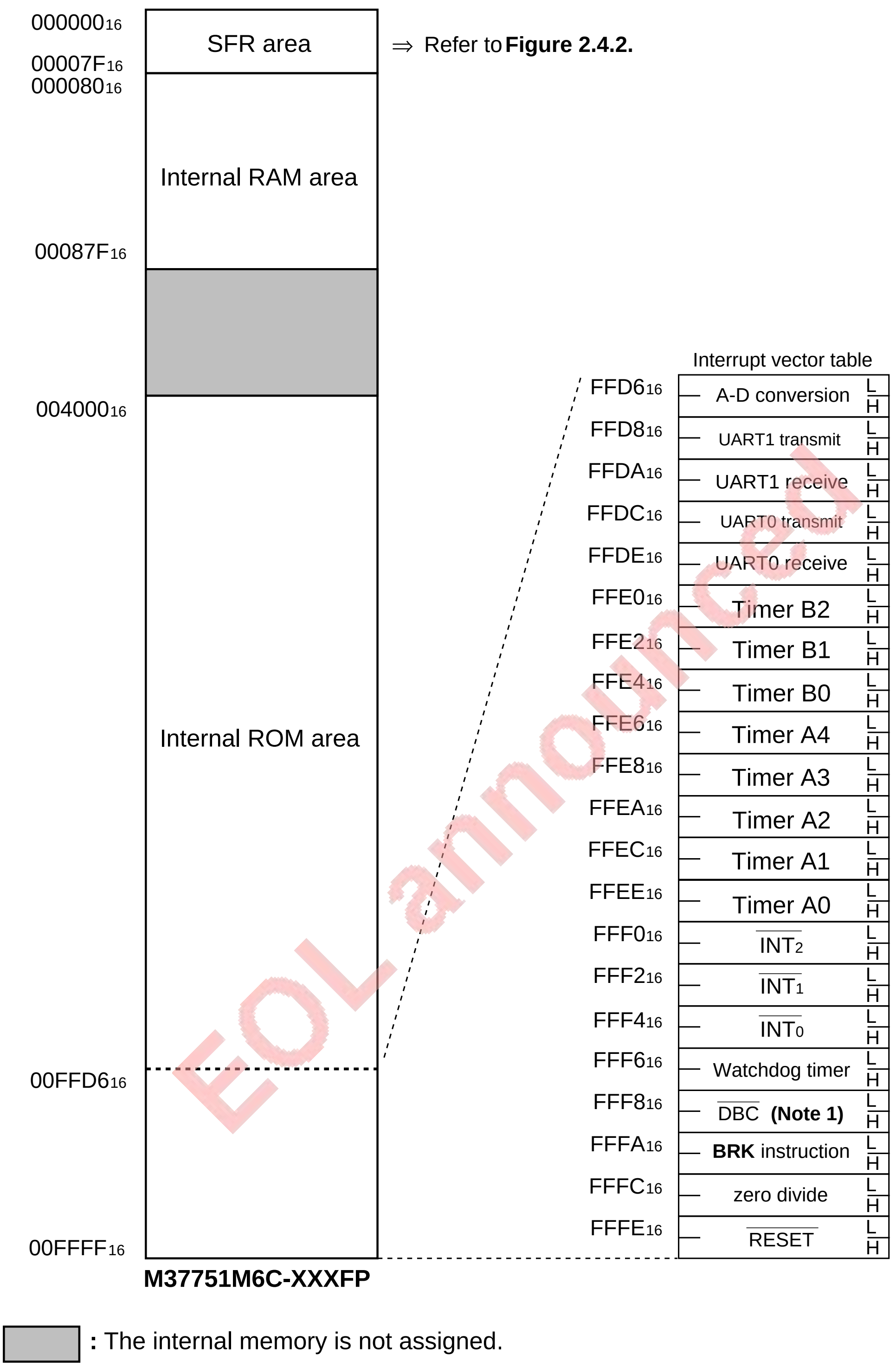
The M37751M6C-XXXFP (See **Note**) assigns the 2048-byte static RAM at addresses 80_{16} to $87F_{16}$. The internal RAM area is used as a stack area, as well as an area to store data. Accordingly, note that set the nesting depth of a subroutine and multiple interrupts' level not to destroy the necessary data.

(3) Internal ROM area

The M37751M6C-XXXFP (See **Note**) assigns the 48-Kbyte mask RAM at addresses 4000_{16} to $FFFF_{16}$. Its addresses $FFD6_{16}$ to $FFFF_{16}$ are the vector addresses, which are called the interrupt vector table, for reset and interrupts. In the microprocessor mode where use of the internal ROM area is inhibited, assign a ROM at addresses $FFD6_{16}$ to $FFFF_{16}$.

Note : Refer to "**Appendix 1. Memory assignment**" for other products.

2.4 Memory assignment



Notes 1: DBC is an interrupt only for debugging; do not use this interrupt.
2: Access to the internal ROM area is disabled in the microprocessor mode.
(Refer to section **"2.5 Processor modes."**)
3: Memory assignment of internal area varies according to the type of microcomputer.
Refer to **"Appendix 1. Memory assignment"** for other products.

Fig. 2.4.1 Internal area's memory assignment

Address	Register	Address	Register
0_{16}		40_{16}	Count start register
1_{16}		41_{16}	
2_{16}	Port P0 register	42_{16}	One-shot start register
3_{16}	Port P1 register	43_{16}	
4_{16}	Port P0 direction register	44_{16}	Up-down register
5_{16}	Port P1 direction register	45_{16}	
6_{16}	Port P2 register	46_{16}	Timer A0 register
7_{16}	Port P3 register	47_{16}	
8_{16}	Port P2 direction register	48_{16}	Timer A1 register
9_{16}	Port P3 direction register	49_{16}	
A_{16}	Port P4 register	$4A_{16}$	Timer A2 register
B_{16}	Port P5 register	$4B_{16}$	
C_{16}	Port P4 direction register	$4C_{16}$	Timer A3 register
D_{16}	Port P5 direction register	$4D_{16}$	
E_{16}	Port P6 register	$4E_{16}$	Timer A4 register
F_{16}	Port P7 register	$4F_{16}$	
10_{16}	Port P6 direction register	50_{16}	Timer B0 register
11_{16}	Port P7 direction register	51_{16}	
12_{16}	Port P8 register	52_{16}	Timer B1 register
13_{16}		53_{16}	
14_{16}	Port P8 direction register	54_{16}	Timer B2 register
15_{16}		55_{16}	
16_{16}		56_{16}	Timer A0 mode register
17_{16}		57_{16}	Timer A1 mode register
18_{16}		58_{16}	Timer A2 mode register
19_{16}		59_{16}	Timer A3 mode register
$1A_{16}$		$5A_{16}$	Timer A4 mode register
$1B_{16}$		$5B_{16}$	Timer B0 mode register
$1C_{16}$		$5C_{16}$	Timer B1 mode register
$1D_{16}$		$5D_{16}$	Timer B2 mode register
$1E_{16}$	A-D control register 0	$5E_{16}$	Processor mode register 0
$1F_{16}$	A-D control register 1	$5F_{16}$	Processor mode register 1
20_{16}	A-D register 0	60_{16}	Watchdog timer register
21_{16}		61_{16}	Watchdog timer frequency select register
22_{16}	A-D register 1	62_{16}	
23_{16}		63_{16}	
24_{16}	A-D register 2	64_{16}	
25_{16}		65_{16}	
26_{16}	A-D register 3	66_{16}	
27_{16}		67_{16}	
28_{16}	A-D register 4	68_{16}	
29_{16}		69_{16}	
$2A_{16}$	A-D register 5	$6A_{16}$	
$2B_{16}$		$6B_{16}$	
$2C_{16}$	A-D register 6	$6C_{16}$	
$2D_{16}$		$6D_{16}$	
$2E_{16}$	A-D register 7	$6E_{16}$	
$2F_{16}$		$6F_{16}$	
30_{16}	UART0 transmit/receive mode register	70_{16}	A-D conversion interrupt control register
31_{16}	UART0 baud rate register (BRG0)	71_{16}	UART0 transmit interrupt control register
32_{16}	UART0 transmit buffer register	72_{16}	UART0 receive interrupt control register
33_{16}		73_{16}	UART1 transmit interrupt control register
34_{16}	UART0 transmit/receive control register 0	74_{16}	UART1 receive interrupt control register
35_{16}	UART0 transmit/receive control register 1	75_{16}	Timer A0 interrupt control register
36_{16}	UART0 receive buffer register	76_{16}	Timer A1 interrupt control register
37_{16}		77_{16}	Timer A2 interrupt control register
38_{16}	UART1 transmit/receive mode register	78_{16}	Timer A3 interrupt control register
39_{16}	UART1 baud rate register (BRG1)	79_{16}	Timer A4 interrupt control register
$3A_{16}$	UART1 transmit buffer register	$7A_{16}$	Timer B0 interrupt control register
$3B_{16}$		$7B_{16}$	Timer B1 interrupt control register
$3C_{16}$	UART1 transmit/receive control register 0	$7C_{16}$	Timer B2 interrupt control register
$3D_{16}$	UART1 transmit/receive control register 1	$7D_{16}$	$\overline{INT_0}$ interrupt control register
$3E_{16}$	UART1 receive buffer register	$7E_{16}$	$\overline{INT_1}$ interrupt control register
$3F_{16}$		$7F_{16}$	$\overline{INT_2}$ interrupt control register

Fig. 2.4.2 SFR area's memory map

2.5 Processor modes

The M37751 can operate in 3 processor modes: single-chip mode, memory expansion mode, and microprocessor mode. Some pins' functions, memory assignment, and access space vary according to the processor modes. This section describes the differences between the processor modes. Figure 2.5.1 shows a memory assignment in each processor mode.

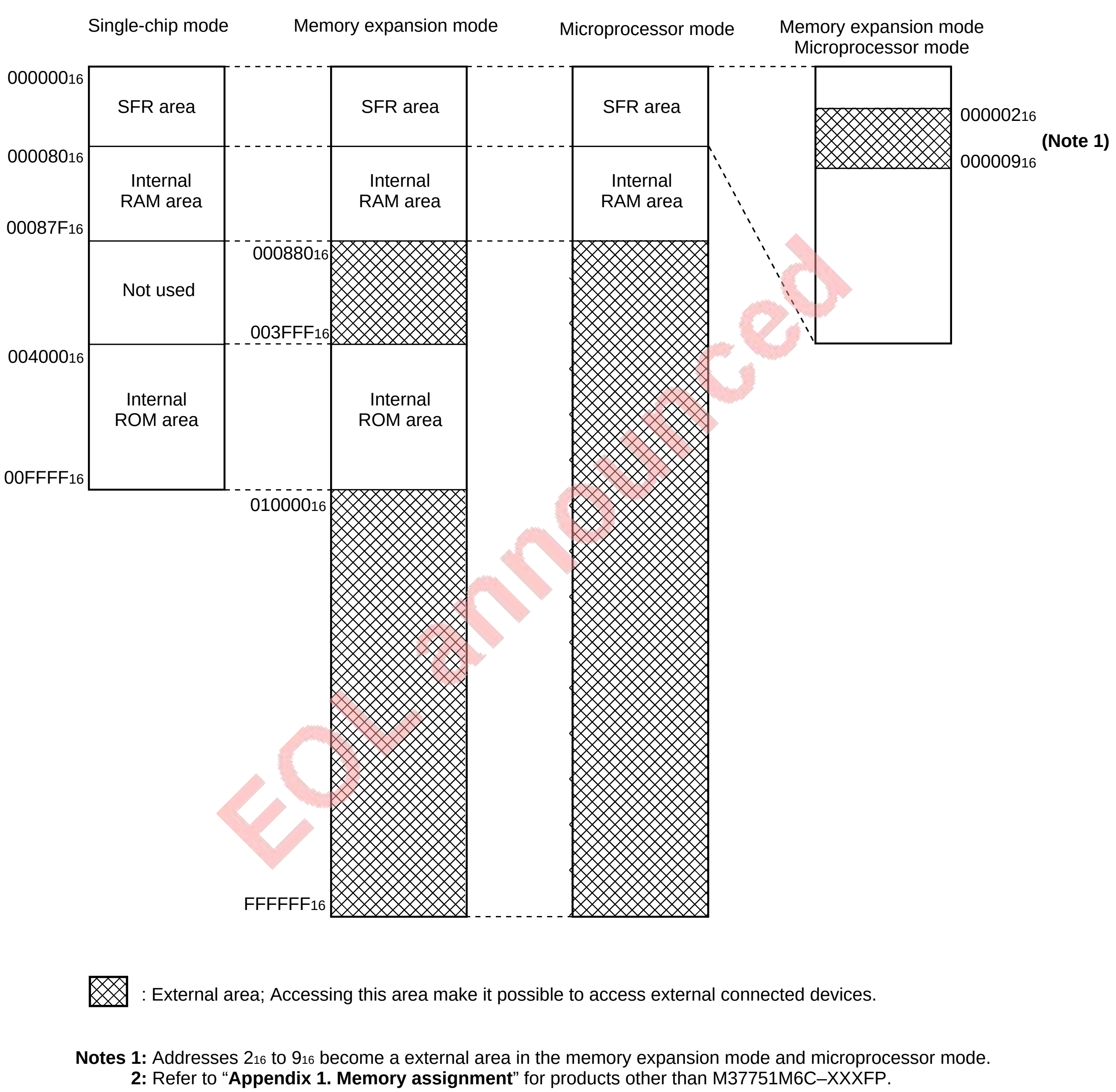


Fig. 2.5.1 Memory assignment in each processor mode for M37751M6C-XXXFP

2.5.1 Single-chip mode

Use this mode when not using external devices. In this mode, ports P0 to P8 function as programmable I/O ports (when using an internal peripheral device, they function as its I/O pins).
In the single-chip mode, only the internal area (SFR, internal RAM, and internal ROM) can be accessed.

2.5.2 Memory expansion and microprocessor modes

Use these modes when connecting devices externally. In these modes, an external device can be connected to any required location in the 16-Mbyte access space. For access to external devices, refer to **"Chapter 12. CONNECTION WITH EXTERNAL DEVICES."**
The memory expansion and microprocessor modes have the same functions except for the following:

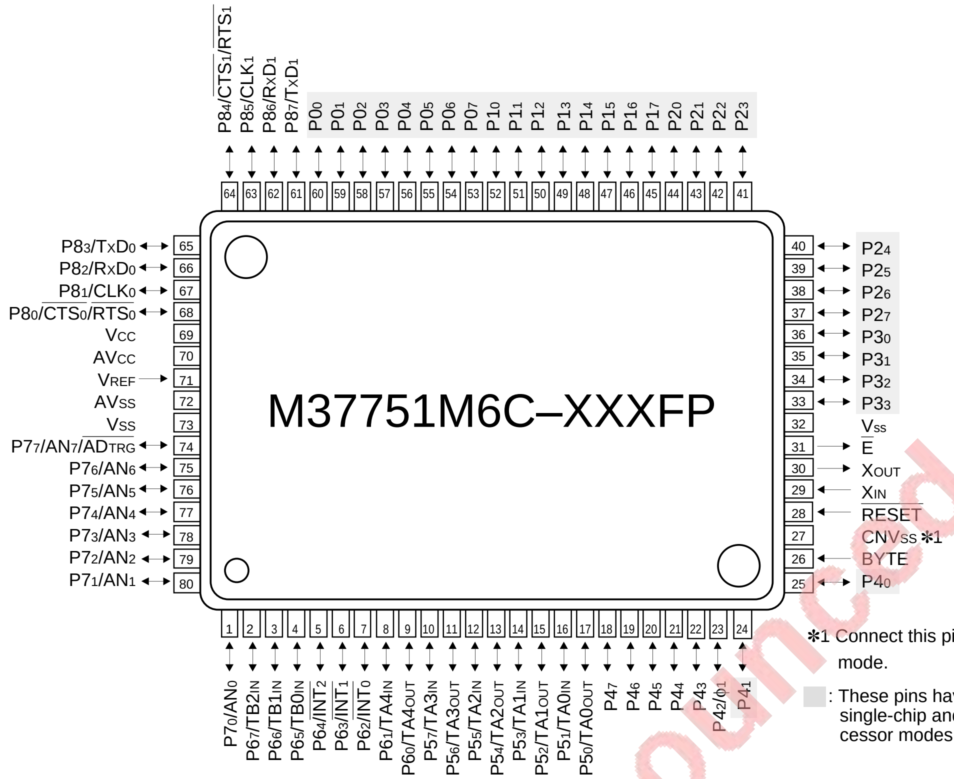
- In the microprocessor mode, access to the internal ROM area is disabled by force, and the internal ROM area is handled as an external area.
- In the microprocessor mode, port $P4_2$ always functions as the clock ϕ_1 output pin.

In the memory expansion and microprocessor modes, P0 to P3, $P4_0$, and $P4_1$ function as the I/O pins for the signals required for accessing external devices. Consequently, these pins cannot be used as programmable I/O ports.
If an external device is connected with an area with which the internal area overlaps, when this overlapping area is read, data in the internal area is taken in the CPU, but data in the external area is not taken in. If data is written to an overlapping area, the data is written to the internal area, and a signal is output externally at the same timing as writing to the internal area.

Figure 2.5.2 shows a pin configuration in each processor mode. Table 2.5.1 lists the functions of P0 to P4 in each processor mode.
For the function of each pin, refer to section **"1.3 Pin description," "Chapter 3. INPUT/OUTPUT PINS,"** and **"Chapter 12. CONNECTION WITH EXTERNAL DEVICES."**

2.5 Processor modes

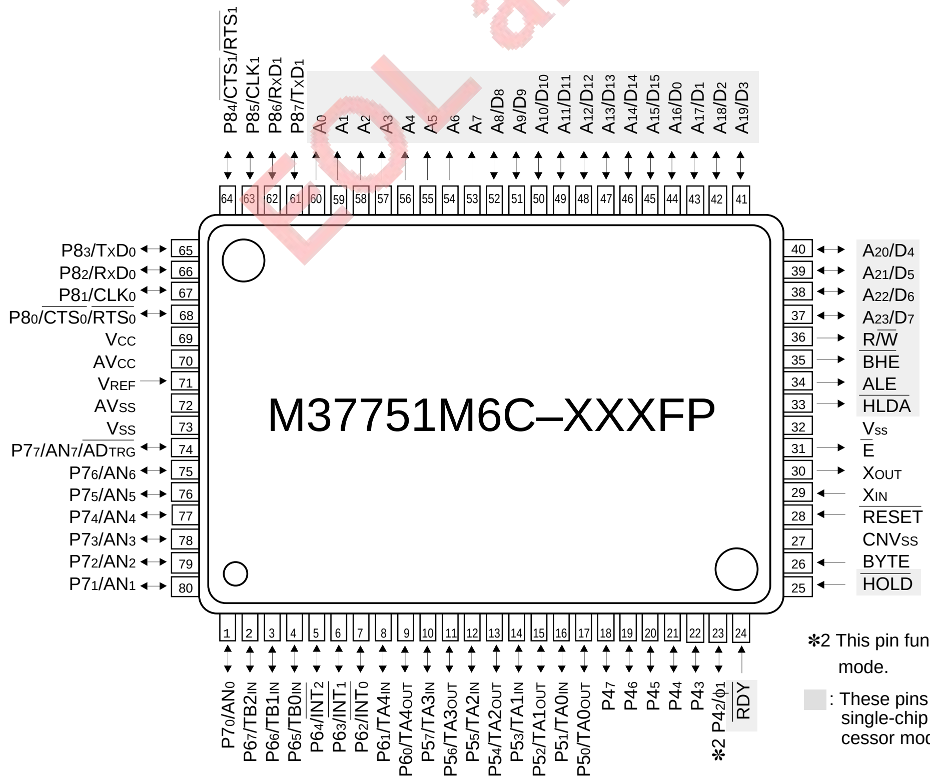
●Single-chip mode



✻1 Connect this pin to Vss pin in the single-chip mode.

: These pins have different functions between the single-chip and the memory expansion/microprocessor modes.

●Memory expansion/Microprocessor mode



✻2 This pin functions as ϕ_1 in the microprocessor mode.

: These pins have different functions between the single-chip and the memory expansion/microprocessor modes.

Fig. 2.5.2 Pin configuration in each processor mode (top view)

Table 2.5.1 Functions of ports P0 to P4 in each processor mode

Pins \ Processor modes	Single-chip mode	Memory expansion/Microprocessor mode
P0	P P: Functions as a programmable I/O port.	A_0 – A_7
P1	P P: Functions as a programmable I/O port.	• When external data bus width is 16 bits (BYTE = "L") A_8 – A_{15} / D(odd) D (odd): Data at odd address • When external data bus width is 8 bits (BYTE = "H") A_8 – A_{15}
P2	P P: Functions as a programmable I/O port.	• When external data bus width is 16 bits (BYTE = "L") A_{16} – A_{23} / D(even) D (even): Data at even address • When external data bus width is 8 bits (BYTE = "H") A_{16} – A_{23} / D D : Data
P3	P P: Functions as a programmable I/O port.	$P3_3$: $\overline{HLDA}$ $P3_2$: ALE $P3_1$: $\overline{BHE}$ $P3_0$: $R/\overline{W}$
P4	P P: Functions as a programmable I/O port. **(Note 1)**	$P4_3$ – $P4_7$: P P: Functions as a programmable I/O port. $P4_2$: ϕ_1 **(Note 2)** $P4_1$: $\overline{RDY}$ $P4_0$: $\overline{HOLD}$

Notes 1: $P4_2$ also functions as the clock ϕ_1 output pin. (Refer to "**Chapter 12. CONNECTION WITH EXTERNAL DEVICES.**")

2: $P4_2$ functions as a programmable I/O port in the memory expansion mode, and that functions as the clock ϕ_1 output pin by software selection. (Refer to "**Chapter 12. CONNECTION WITH EXTERNAL DEVICES.**")

3: This table lists a switch of pins' functions by switching the processor mode. Refer to the following section about the input/output timing of each signal:
- "**Chapter 12. CONNECTION WITH EXTERNAL DEVICES.**"
- "**Chapter 15. ELECTRICAL CHARACTERISTICS.**"

2.5.3 Setting processor modes

The voltage supplied to the CNVss pin and the processor mode bits (bits 1 and 0 at address $5E_{16}$) set the processor mode.

● When Vss level is supplied to CNVss pin

After a reset, the microcomputer starts operating in the single-chip mode. The processor mode is switched by the processor mode bits after the microcomputer starts operating. When the processor mode bits are set to "01_2," the microcomputer enters the memory expansion mode; when these bits are set to "10_2," the microcomputer enters the microprocessor mode.

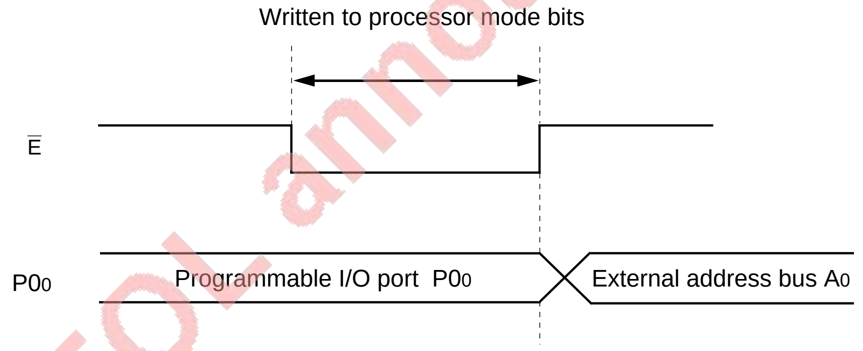
The processor mode is switched at the rising edge of signal $\overline{E}$ after writing to the processor mode bits. Figure 2.5.3 shows the timing when pin functions are switched by switching the processor mode from the single-chip mode to the memory expansion or microprocessor mode with the processor mode bits.

When the processor mode is switched during the program execution, the contents of the instruction queue buffer is not initialized. (Refer to **"Appendix 9. Q & A."**)

● When Vcc level is supplied to CNVss pin

After a reset, the microcomputer starts operating in the microprocessor mode. In this case, the microcomputer cannot operate in the other modes. (Fix the processor mode bits to "10_2.")

Table 2.5.2 lists the methods for setting processor modes. Figure 2.5.4 shows the structure of processor mode register 0 (address $5E_{16}$).



Note: Functions of pins $P0_1$ to $P0_7$, P1 to P3, $P4_0$ to $P4_2$ are switched at the same timing shown above. Function of pin $P4_2$ is, however, switched only when the processor mode is switched to the microprocessor mode.

Fig. 2.5.3 Timing when pin functions are switched

Table 2.5.2 Methods for setting processor modes

Processor mode	CNVss pin level	Processor mode bits	
		b1	b0
Single-chip mode	Vss (0 V) (**Note 1**)	0	0
Memory expansion mode	Vss (0 V) (**Note 1**)	0	1
Microprocessor mode	Vss (0 V) (**Note 1**)	1	0
	Vcc (5 V) (**Note 2**)		

Notes 1: The microcomputer starts operating in the single-chip mode after a reset. The microcomputer can be switched to the other processor modes by setting the processor mode bits.

2: The microcomputer starts operating in the microprocessor mode after a reset. The microcomputer cannot operate in the other modes, so that fix the processor mode bits as follows:
•b1 = “1” and b0 = “0.”

b7	b6	b5	b4	b3	b2	b1	b0
	0				0		

Processor mode register 0 (Address $5E_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Memory expansion mode 1 0 : Microprocessor mode 1 1 : Not selected	0	RW
1			0 (Note 1)	RW
2	Fix this bit to “0.”		0	RW
3	Software reset bit	The microcomputer is reset by writing “1” to this bit. The value is “0” at reading.	0	WO
4	Interrupt priority detection time select bits	b5 b4 0 0 : 7 cycles of ϕ 0 1 : 4 cycles of ϕ 1 0 : 2 cycles of ϕ 1 1 : Not selected	0	RW
5			0	RW
6	Fix this bit to “0.”		0	RW
7	Clock ϕ_1 output select bit **(Note 2)**	0 : Clock ϕ_1 output disabled (P4$_2$ functions as a programmable I/O port.) 1 : Clock ϕ_1 output enabled (P4$_2$ functions as a clock ϕ_1 output pin.)	0	RW

Notes 1: While supplying the Vcc level to the CNVss pin, this bit becomes “1” after a reset. (Fixed to “1.”)
2: This bit is ignored in the microprocessor mode. (It may be either “0” or “1.”)

(shaded) : Bits 7 to 2 are not used for setting of the processor mode.

Fig. 2.5.4 Structure of processor mode register 0

[Precautions when operating in single-chip mode]

The bus cycle select bits (bits 4 and 5 at address $5F_{16}$) is not used in the single-chip mode. However, do not make those bits state of not selected in all cases. Especially in low-speed running, rewrite both bits at the same time to “01_2,” “10_2” or “11_2.” These bits are cleared to “00_2” at reset.

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

Processor mode register 1 (Address $5F_{16}$)

Bit	Bit name	Functions	At reset	RW
1, 0	Fix these bits to “0.”		0	RW
2	Clock source for peripheral devices select bit **(Note)**	0 : φ divided by 2 1 : φ	0	RW
3	CPU running speed select bit **(Note)**	0 : High-speed running 1 : Low-speed running	0	RW
4	Bus cycle select bits	In high-speed running b5 b4 0 0 : 5φ access in high-speed running 0 1 : 4φ access in high-speed running 1 0 : 3φ access in high-speed running 1 1 : Not selected	0	RW
5		In low-speed running b5 b4 0 0 : Not selected 0 1 : 4φ access in low-speed running 1 0 : 3φ access in low-speed running 1 1 : 2φ access in low-speed running	0	RW
7, 6	Fix these bits to “0.”		0	RW

Note: Fix this bit to “0” when $f(X_{IN}) > 25$ MHz.

Fig. 2.5.5 Structure of processor mode register 1

CHAPTER 3

INPUT/OUTPUT PINS

This chapter describes the programmable I/O ports in the single-chip mode. For P0 to P4, which change their functions according to the processor mode, refer also to the section “**2.5 Processor modes**” and “**Chapter 12. CONNECTION WITH EXTERNAL DEVICES.**”

3.1 Programmable I/O ports

The 7751 Group has 68 programmable I/O ports, P0 to P8.
The programmable I/O ports have direction registers and port registers in the SFR area. Figure 3.1.1 shows the memory map of direction registers and port registers.
$P4_2$ and P5 to P8 also function as the I/O pins of the internal peripheral devices. For the functions, refer to the section “**3.2 I/O pins of internal peripheral devices**” and relevant sections of each internal peripheral devices.

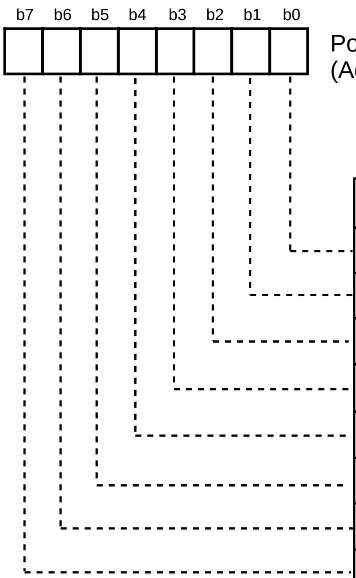
Addresses	
2_{16}	Port P0 register
3_{16}	Port P1 register
4_{16}	Port P0 direction register
5_{16}	Port P1 direction register
6_{16}	Port P2 register
7_{16}	Port P3 register
8_{16}	Port P2 direction register
9_{16}	Port P3 direction register
A_{16}	Port P4 register
B_{16}	Port P5 register
C_{16}	Port P4 direction register
D_{16}	Port P5 direction register
E_{16}	Port P6 register
F_{16}	Port P7 register
10_{16}	Port P6 direction register
11_{16}	Port P7 direction register
12_{16}	Port P8 register
13_{16}	
14_{16}	Port P8 direction register

Fig. 3.1.1 Memory map of direction registers and port registers

3.1.1 Direction register

This register determines the input/output direction of the programmable I/O port. Each bit of this register corresponds one for one to each pin of the microcomputer.

Figure 3.1.2 shows the structure of port Pi (i = 0 to 8) direction register.



Port Pi direction register (i = 0 to 8)
(Addresses 4_{16}, 5_{16}, 8_{16}, 9_{16}, C_{16}, D_{16}, 10_{16}, 11_{16}, 14_{16})

Bit	Bit name	Functions	At reset	RW
0	Port Pi_0 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	0	RW
1	Port Pi_1 direction bit		0	RW
2	Port Pi_2 direction bit		0	RW
3	Port Pi_3 direction bit		0	RW
4	Port Pi_4 direction bit		0	RW
5	Port Pi_5 direction bit		0	RW
6	Port Pi_6 direction bit		0	RW
7	Port Pi_7 direction bit		0	RW

Note: Bits 7 to 4 of the port P3 direction register cannot be written and are fixed to “0” at reading.

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Corresponding pin	Pi_7	Pi_6	Pi_5	Pi_4	Pi_3	Pi_2	Pi_1	Pi_0

Fig. 3.1.2 Structure of port Pi (i = 0 to 8) direction register

3.1.2 Port register

Data is input/output to/from externals by writing/reading data to/from the port register. The port register consists of a port latch which holds the output data and a circuit which reads the pin state. Each bit of the port register corresponds one for one to each pin of the microcomputer. Figure 3.1.3 shows the structure of the port Pi (i = 0 to 8) register.

● **When outputting data from programmable I/O ports set to output mode**

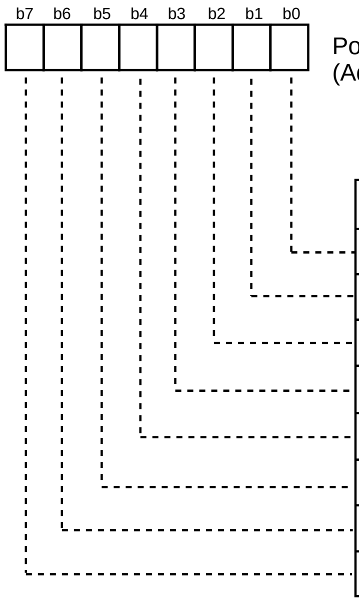
① By writing data to the corresponding bit of the port register, the data is written into the port latch.
② The data is output from the pin according to the contents of the port latch.

By reading the port register of a port set to output mode, the contents of the port latch is read out, instead of the pin state. Accordingly, the output data is correctly read without being affected by an external load. (Refer to Figures 3.1.4 and 3.1.5.)

● **When inputting data from programmable I/O ports set to input mode**

① The pin which is set to input mode enters the floating state.
② By reading the corresponding bit of the port register, the data which is input from the pin can be read out.

By writing data to the port register of a programmable I/O port set to input mode, the data is only written into the port latch and is not output to externals. The pin retains floating.



Port Pi register (i = 0 to 8)
(Addresses 2_{16}, 3_{16}, 6_{16}, 7_{16}, A_{16}, B_{16}, E_{16}, F_{16}, 12_{16})

Bit	Bit name	Functions	At reset	RW
0	Port Pi_0	Data is input/output to/from a pin by reading/writing from/to the corresponding bit. 0 : "L" level 1 : "H" level	Undefined	RW
1	Port Pi_1		Undefined	RW
2	Port Pi_2		Undefined	RW
3	Port Pi_3		Undefined	RW
4	Port Pi_4		Undefined	RW
5	Port Pi_5		Undefined	RW
6	Port Pi_6		Undefined	RW
7	Port Pi_7		Undefined	RW

Note: Bits 7 to 4 of the port P3 register cannot be written and are fixed to "0" at reading.

Fig. 3.1.3 Port Pi (i = 0 to 8) register structure

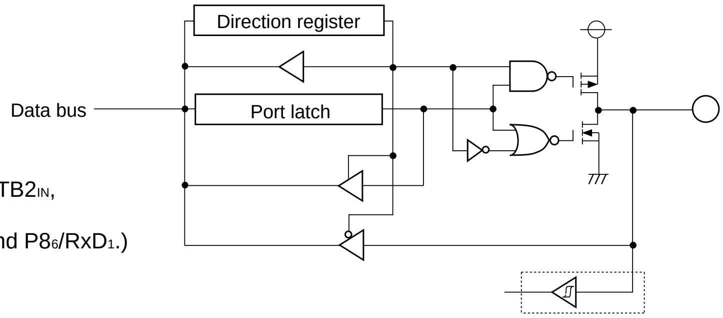
Figures 3.1.4 and 3.1.5 show the port peripheral circuits.

[Inside dotted-line not included]

Ports $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_3$, $P4_3$ to $P4_6$

[Inside dotted-line included]

Ports $P4_0$, $P4_1$, $P4_7$, $P5_1$/$TA0_{IN}$, $P5_3$/$TA1_{IN}$, $P5_5$/$TA2_{IN}$, $P5_7$/$TA3_{IN}$, $P6_1$/$TA4_{IN}$, $P6_2$/$\overline{INT_0}$ to $P6_4$/$\overline{INT_2}$, $P6_5$/$TB0_{IN}$ to $P6_7$/$TB2_{IN}$, $P8_2$/RxD_0, $P8_6$/RxD_1
(There is no hysteresis for $P8_2$/RxD_0 and $P8_6$/RxD_1.)

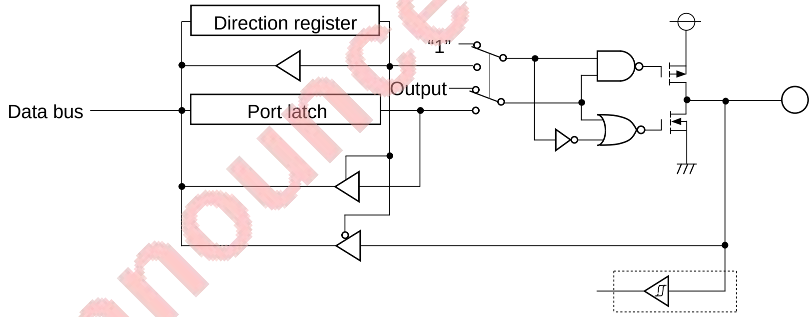


[Inside dotted-line not included.]

Ports $P4_2$/ϕ_1, $P8_3$/TxD_0, $P8_7$/TxD_1

[Inside dotted-line included.]

Ports $P5_0$/$TA0_{OUT}$, $P5_2$/$TA1_{OUT}$, $P5_4$/$TA2_{OUT}$, $P5_6$/$TA3_{OUT}$, $P6_0$/$TA4_{OUT}$



[Inside dotted-line not included]

Ports $P7_0$/AN_0 to $P7_6$/AN_6

[Inside dotted-line included]

Port $P7_7$/AN_7/$\overline{AD_{TRG}}$

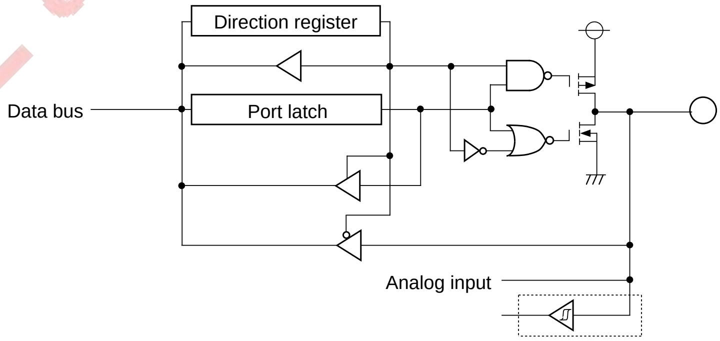
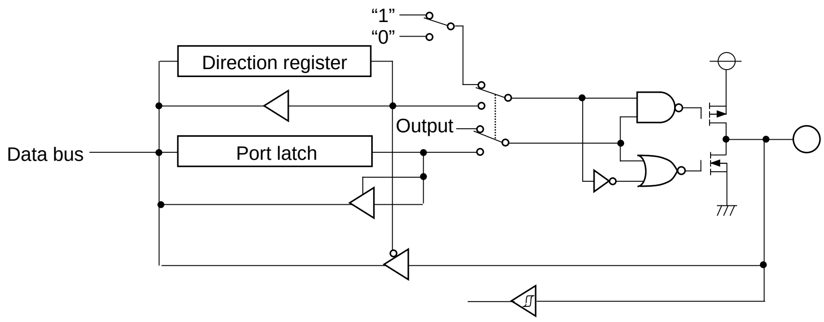


Fig. 3.1.4 Port peripheral circuits (1)

Ports P8$_0$/$\overline{\text{CTS}_0}$/$\overline{\text{RTS}_0}$, P8$_1$/CLK$_0$,
P8$_4$/CTS$_1$/RTS$_1$, P8$_5$/CLK$_1$



$\overline{\text{E}}$ output pin

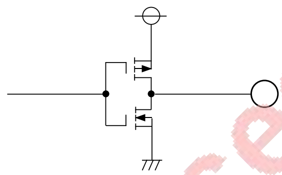


Fig. 3.1.5 Port peripheral circuits (2)

3.2 I/O pins of internal peripheral devices ($P4_2$, P5–P8)

$P4_2$ and P5 to P8 also function as the I/O pins of the internal peripheral devices. Table 3.2.1 lists I/O pins for the internal peripheral devices.
For their functions, refer to relevant sections of each internal peripheral device. For the clock ϕ_1 output pin, refer to "**Chapter 12. CONNECTION WITH EXTERNAL DEVICES.**"

Table 3.2.1 I/O pins for internal peripheral devices

Port	I/O pins for internal peripheral devices
$P4_2$	Clock ϕ_1 output pin
P5	I/O pins of Timer A
$P6_0$, $P6_1$	
$P6_2$ to $P6_4$	Input pins of external interrupts
$P6_5$ to $P6_7$	Input pins of Timer B
P7	Input pins of A-D converter
P8	I/O pins of Serial I/O

CHAPTER 4

INTERRUPTS

The suspension of the current operation in order to perform another operation owing to a certain factor is referred to as "Interrupt." This chapter describes the interrupts.

4.1 Overview

The M37751 has 19 interrupt sources to generate interrupt requests.
Figure 4.1.1 shows the interrupt processing sequence.
When an interrupt request is accepted, a branch is made to the start address of the interrupt routine set in the interrupt vector table (addresses $FFD6_{16}$ to $FFFF_{16}$). Set the start address of each interrupt routine at each interrupt vector address in the interrupt vector table.

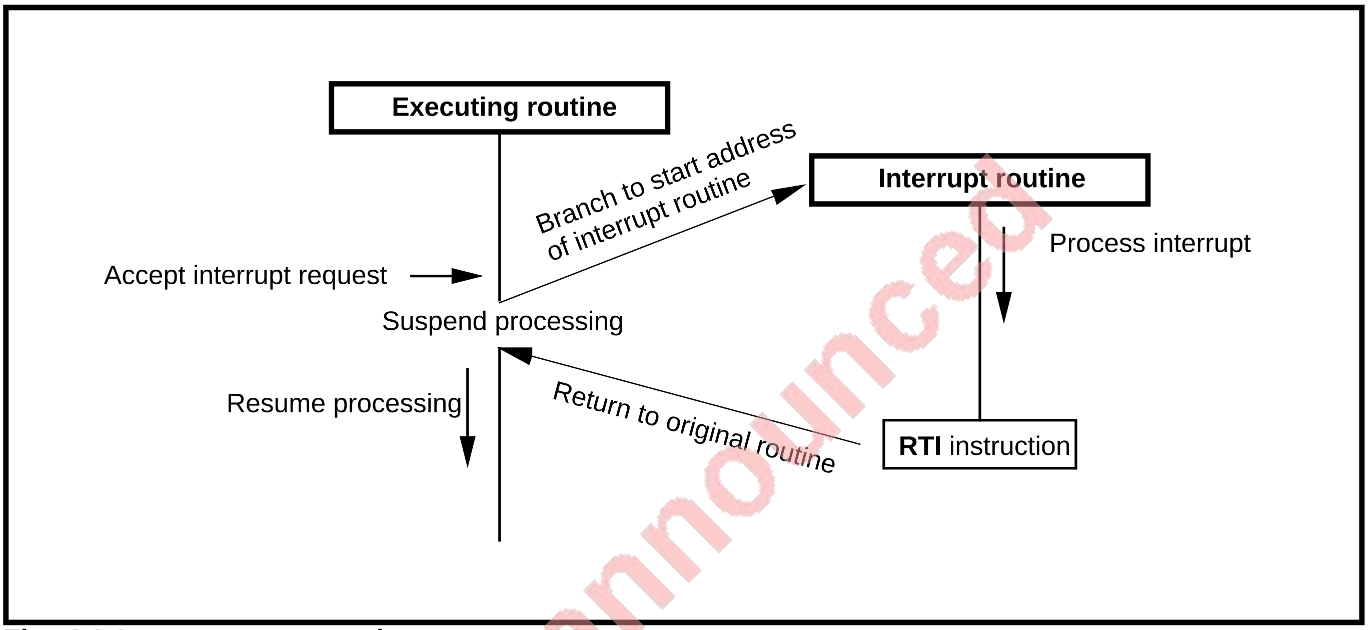


Fig. 4.1.1 Interrupt processing sequence

When an interrupt request is accepted, the contents of the registers listed below immediately preceding the acceptance of the interrupt request are automatically saved to the stack area in order of registers ①→②→③.

① Program bank register (PG)
② Program counter (PC_L, PC_H)
③ Processor status register (PS_L, PS_H)

Figure 4.1.2 shows the state of the stack area just before entering the interrupt routine.
Execute the **RTI** instruction at the end of this interrupt routine to return to the routine that the microcomputer was executing before the interrupt request was accepted. As the **RTI** instruction is executed, the register contents saved in the stack area are restored in order of registers ③→②→①, and a return is made to the routine executed before the acceptance of interrupt request and processing is resumed from it.
When an interrupt request is accepted and the **RTI** instruction is executed, the only above registers ① to ③ are automatically saved and restored. When there are any other registers of which contents are necessary to be kept, use software to save and restore them.

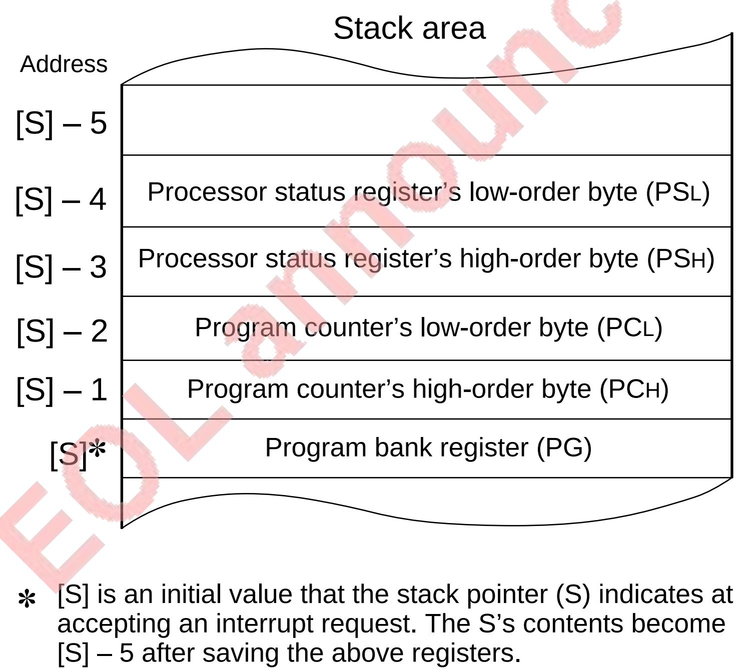


Fig. 4.1.2 State of stack area just before entering interrupt routine

4.2 Interrupt sources

Table 4.2.1 lists the interrupt sources and the interrupt vector addresses. When programming, set the start address of each interrupt routine at the vector addresses listed in this table.

Table 4.2.1 Interrupt sources and interrupt vector addresses

Interrupt source	Interrupt vector address		Remarks
	High-order address	Low-order address	
Reset	$FFFF_{16}$	$FFFE_{16}$	Non-maskable
Zero division	$FFFD_{16}$	$FFFC_{16}$	Non-maskable software interrupt
BRK instruction	$FFFB_{16}$	$FFFA_{16}$	Non-maskable software interrupt
$\overline{DBC}$ **(Note)**	$FFF9_{16}$	$FFF8_{16}$	Not used usually
Watchdog timer	$FFF7_{16}$	$FFF6_{16}$	Non-maskable interrupt
$\overline{INT_0}$	$FFF5_{16}$	$FFF4_{16}$	External interrupt due to $\overline{INT_0}$ pin input signal
$\overline{INT_1}$	$FFF3_{16}$	$FFF2_{16}$	External interrupt due to $\overline{INT_1}$ pin input signal
$\overline{INT_2}$	$FFF1_{16}$	$FFF0_{16}$	External interrupt due to $\overline{INT_2}$ pin input signal
Timer A0	$FFEF_{16}$	$FFEE_{16}$	Internal interrupt from Timer A0
Timer A1	$FFED_{16}$	$FFEC_{16}$	Internal interrupt from Timer A1
Timer A2	$FFEB_{16}$	$FFEA_{16}$	Internal interrupt from Timer A2
Timer A3	$FFE9_{16}$	$FFE8_{16}$	Internal interrupt from Timer A3
Timer A4	$FFE7_{16}$	$FFE6_{16}$	Internal interrupt from Timer A4
Timer B0	$FFE5_{16}$	$FFE4_{16}$	Internal interrupt from Timer B0
Timer B1	$FFE3_{16}$	$FFE2_{16}$	Internal interrupt from Timer B1
Timer B2	$FFE1_{16}$	$FFE0_{16}$	Internal interrupt from Timer B2
UART0 receive	$FFDF_{16}$	$FFDE_{16}$	Internal interrupt from UART0
UART0 transmit	$FFDD_{16}$	$FFDC_{16}$	
UART1 receive	$FFDB_{16}$	$FFDA_{16}$	Internal interrupt from UART1
UART1 transmit	$FFD9_{16}$	$FFD8_{16}$	
A-D conversion	$FFD7_{16}$	$FFD6_{16}$	Internal interrupt from A-D converter

Note: The $\overline{DBC}$ interrupt source is used exclusively for debugger control.

Table 4.2.2 lists occurrence factors of internal interrupt request, which occur due to internal operation.

Table 4.2.2 Occurrence factors of internal interrupt request

Interrupt	Interrupt request occurrence factors
Zero division interrupt	Occurs when “0” is specified as the divisor for the **DIV** instruction (Division instruction). (Refer to **“7751 Series Software Manual.”**)
BRK instruction interrupt	Occurs when the **BRK** instruction is executed. (Refer to **“7751 Series Software Manual.”**)
Watchdog timer interrupt	Occurs when the most significant bit of the watchdog timer becomes “0.” (Refer to **“Chapter 9. WATCHDOG TIMER.”**)
Timer Ai interrupt (i = 0 to 4)	Differs according to the timer Ai’s operating modes. (Refer to **“Chapter 5. TIMER A.”**)
Timer Bi interrupt (i = 0 to 2)	Differs according to the timer Bi’s operating modes. (Refer to **“Chapter 6. TIMER B.”**)
UARTi receive interrupt (i = 0, 1)	Occurs at serial data reception. (Refer to **“Chapter 7. SERIAL I/O.”**)
UARTi transmit interrupt (i = 0, 1)	Occurs at serial data transmission. (Refer to **“Chapter 7. SERIAL I/O.”**)
A-D conversion interrupt	Occurs when A-D conversion is completed. (Refer to **“Chapter 8. A-D CONVERTER.”**)

4.3 Interrupt control

The enabling and disabling of maskable interrupts are controlled by the following :

•Interrupt request bit
•Interrupt priority level select bits
•Processor interrupt priority level (IPL)
•Interrupt disable flag (I)

The interrupt disable flag (I) and the processor interrupt priority level (IPL) are assigned to the processor status register (PS). The interrupt request bit and the interrupt priority level select bits are assigned to the interrupt control register of each interrupt.
Figure 4.3.1 shows the memory assignment of the interrupt control registers, and Figure 4.3.2 shows their structure.

●**Maskable interrupt:** An interrupt of which request's acceptance can be disabled by software.
●**Non-maskable interrupt** (including Zero division, **BRK** instruction, Watchdog timer interrupts)**:**
An interrupt which is certain to be accepted when its request occurs. These interrupts do not have their interrupt control registers and are independent of the interrupt disable flag (I).

Address	
70_{16}	A-D conversion interrupt control register
71_{16}	UART0 transmit interrupt control register
72_{16}	UART0 receive interrupt control register
73_{16}	UART1 transmit interrupt control register
74_{16}	UART1 receive interrupt control register
75_{16}	Timer A0 interrupt control register
76_{16}	Timer A1 interrupt control register
77_{16}	Timer A2 interrupt control register
78_{16}	Timer A3 interrupt control register
79_{16}	Timer A4 interrupt control register
$7A_{16}$	Timer B0 interrupt control register
$7B_{16}$	Timer B1 interrupt control register
$7C_{16}$	Timer B2 interrupt control register
$7D_{16}$	$\overline{INT_0}$ interrupt control register
$7E_{16}$	$\overline{INT_1}$ interrupt control register
$7F_{16}$	$\overline{INT_2}$ interrupt control register

Fig. 4.3.1 Memory assignment of interrupt control registers

b7	b6	b5	b4	b3	b2	b1	b0

A-D conversion, UART0 and 1 transmit, UART0 and 1 receive, timers A0 to A4, timers B0 to B2 interrupt control registers (Addresses 70_{16} to $7C_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 Low level 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7 High level	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	RW
7 to 4	Nothing is assigned.		Undefined	–

b7	b6	b5	b4	b3	b2	b1	b0

$\overline{INT_0}$ to $\overline{INT_2}$ interrupt control registers (Addresses $7D_{16}$ to $7F_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 Low level 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7 High level	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit **(Note)**	0 : No interrupt request 1 : Interrupt request	0	RW
4	Polarity select bit	0 : Set the interrupt request bit at "H" level for level sense and at falling edge for edge sense. 1 : Set the interrupt request bit at "L" level for level sense and at rising edge for edge sense.	0	RW
5	Level sense/Edge sense select bit	0 : Edge sense 1 : Level sense	0	RW
7, 6	Nothing is assigned.		Undefined	–

Note: The $\overline{INT_0}$ to $\overline{INT_2}$ interrupt request bits are invalid when selecting the level sense.

Fig. 4.3.2 Structure of interrupt control register

4.3.1 Interrupt disable flag (I)

All maskable interrupts can be disabled by this flag. When this flag is set to “1,” all maskable interrupts are disabled; when the flag is cleared to “0,” those interrupts are enabled. Because this flag is set to “1” at reset, clear the flag to “0” when enabling interrupts.

4.3.2 Interrupt request bit

When an interrupt request occurs, this bit is set to “1.” The bit remains set to “1” until the interrupt request is accepted, and it is cleared to “0” when the interrupt request is accepted.
This bit also can be set to “0” or “1” by software.
For the $\overline{\text{INT}_i}$ interrupt request bit (i = 0 to 2), when using the $\overline{\text{INT}_i}$ interrupt with level sense, the bit is ignored.

4.3.3 Interrupt priority level select bits and processor interrupt priority level (IPL)

The interrupt priority level select bits are used to determine the priority level of each interrupt. Use the **SEB** or **CLB** instruction to set these bits.
When an interrupt request occurs, its interrupt priority level is compared with the processor interrupt priority level (IPL). The requested interrupt is enabled only when the comparison result meets the following condition. Accordingly, an interrupt can be disabled by setting its interrupt priority level to 0.

Each interrupt priority level > Processor interrupt priority level (IPL)

Table 4.3.1 lists the setting of interrupt priority level, and Table 4.3.2 lists the interrupt enabled level corresponding to IPL contents.

All the interrupt disable flag (I), interrupt request bit, interrupt priority level select bits, and processor interrupt priority level (IPL) are independent of one another; they do not affect one another. Interrupt requests are accepted only when the following conditions are satisfied.

- Interrupt disable flag (I) = “0”
- Interrupt request bit = “1”
- Interrupt priority level > Processor interrupt priority level (IPL)

Table 4.3.1 Setting of interrupt priority level

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

Table 4.3.2 Interrupt enabled level corresponding to IPL contents

IPL_2	IPL_1	IPL_0	Enabled interrupt priority level
0	0	0	Enable level 1 and above interrupts.
0	0	1	Enable level 2 and above interrupts.
0	1	0	Enable level 3 and above interrupts.
0	1	1	Enable level 4 and above interrupts.
1	0	0	Enable level 5 and above interrupts.
1	0	1	Enable level 6 and level 7 interrupts.
1	1	0	Enable only level 7 interrupt.
1	1	1	Disable all maskable interrupts.

IPL_0: Bit 8 in processor status register (PS)
IPL_1: Bit 9 in processor status register (PS)
IPL_2: Bit 10 in processor status register (PS)

4.4 Interrupt priority level

When two or more interrupt requests are detected at the same sampling timing, at which whether an interrupt request exists or not is checked, in the case of the interrupt disable flag (I) = "0" (interrupts enabled); they are accepted in order of priority levels, with the highest priority interrupt request accepted first.
Among a total of 19 interrupt sources, the user can set the desired priority levels for 16 interrupt sources except software interrupts (zero division and **BRK** instruction interrupts) and the watchdog timer interrupt. Use the interrupt priority level select bits to set their priority levels. Additionally, the reset, which is handled as one that has the highest priority of all interrupts, and the watchdog timer interrupt have their priority levels set by hardware. Figure 4.4.1 shows the interrupt priority levels set by hardware.
Note that software interrupts are not affected by interrupt priority levels. Whenever the instruction is executed, a branch is certain to be made to the interrupt routine.

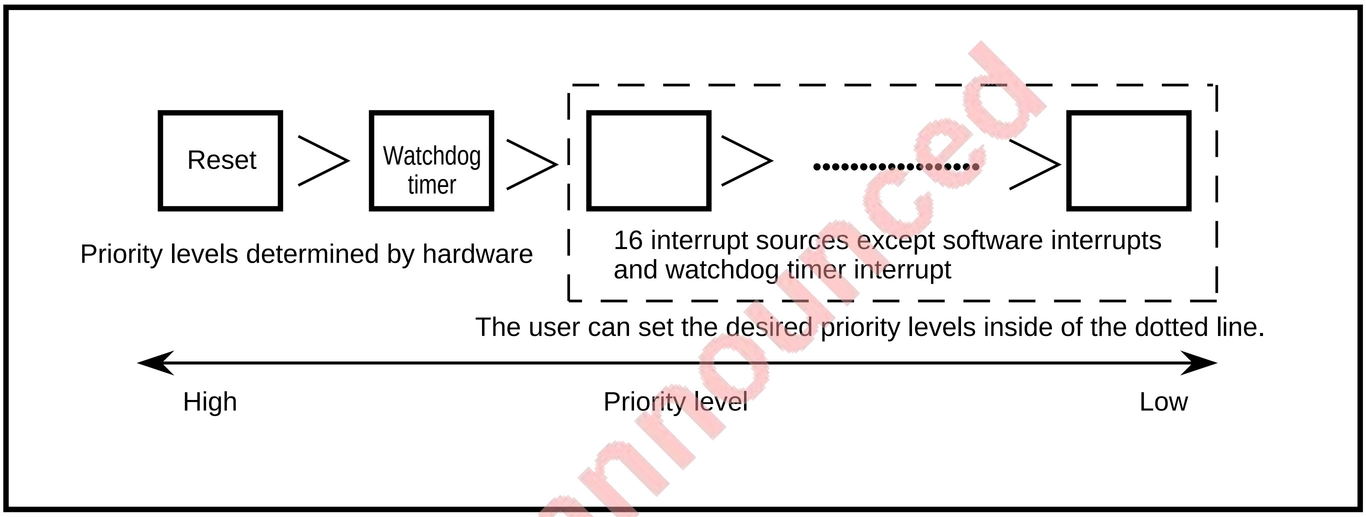


Fig. 4.4.1 Interrupt priority levels set by hardware

4.5 Interrupt priority level detection circuit

The interrupt priority level detection circuit selects the interrupt having the highest priority level when more than one interrupt request occurs at the same sampling timing. Figure 4.5.1 shows the interrupt priority level detection circuit.

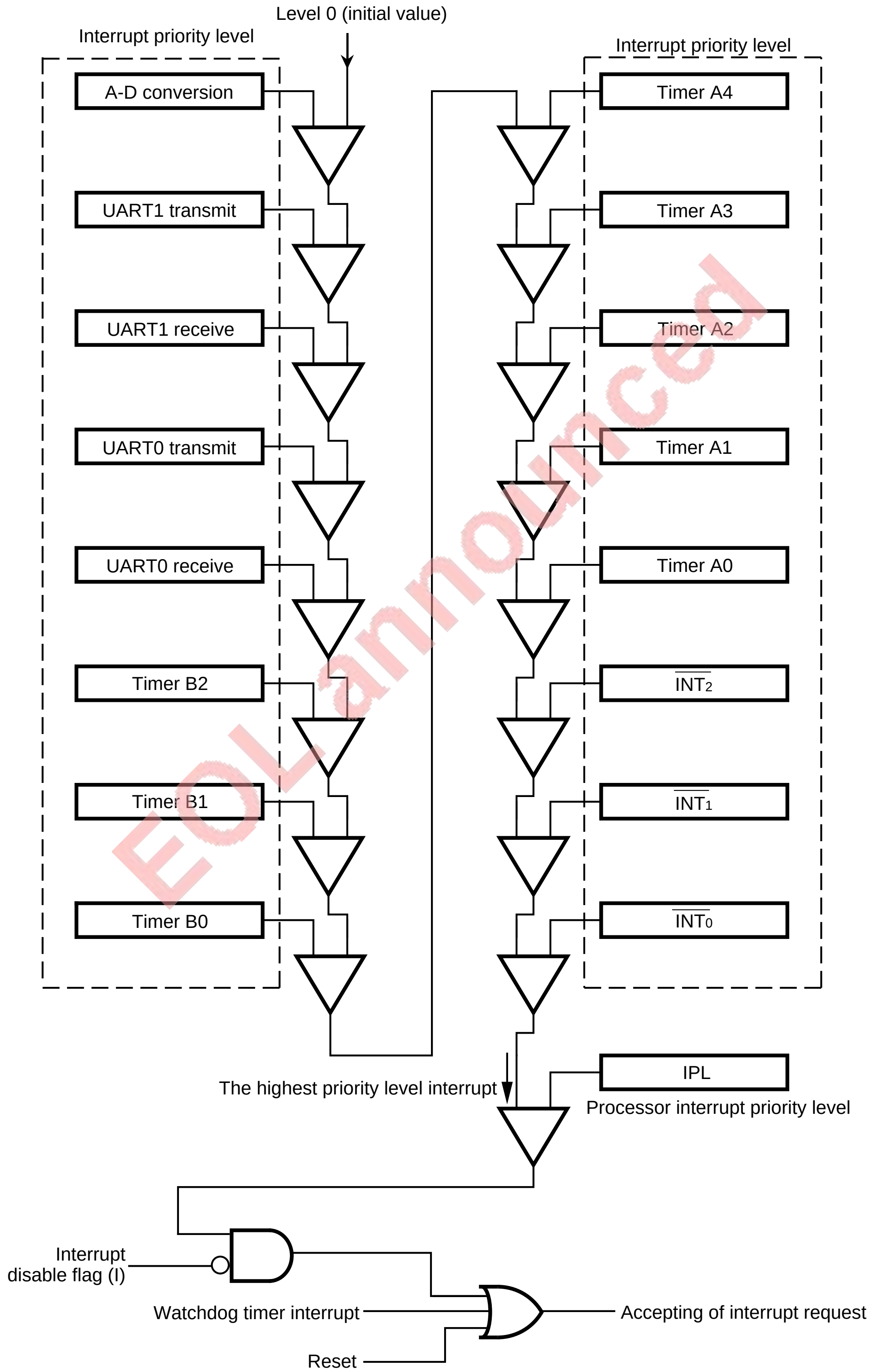


Fig. 4.5.1 Interrupt priority level detection circuit

4.5 Interrupt priority level detection circuit

The following explains the operation of the interrupt priority detection circuit using Figure 4.5.2.
The interrupt priority level of a requested interrupt (Y in Figure 4.5.2) is compared with the resultant priority level sent from the preceding comparator (X in Figure 4.5.2); whichever interrupt of the higher priority level is sent to the next comparator (Z in Figure 4.5.2). (Initial comparison value is “0.”) For interrupts for which no interrupt request occurs, the priority level sent from the preceding comparator is forwarded to the next comparator. When the two priority levels are found the same by comparison, the priority level sent from the preceding comparator is forwarded to the next comparator. Accordingly, when the same priority level is set by software, the interrupt requests are subject to the following relation about priority:

A-D conversion > UART1 transmit > UART1 receive > UART0 transmit > UART0 receive > Timer B2 > Timer B1 > Timer B0 > Timer A4 > Timer A3 > Timer A2 > Timer A1 > Timer A0 > $\overline{INT_2}$ > $\overline{INT_1}$ > $\overline{INT_0}$

Among the multiple interrupt requests sampled at the same time, one that has the highest priority level is detectedd by the above comparison.

Then this highest interrupt priority level is compared with the processor interrupt priority level (IPL). When this interrupt priority level is higher than the processor interrupt priority level (IPL) and the interrupt disable flag (I) is “0,” the interrupt request is accepted. A interrupt request which is not accepted here is retained until it is accepted or its interrupt request bit is cleared to “0” by software.
The interrupt priority is detected when the CPU fetches an op code, which is called the CPU’s op-code fetch cycle. However, when an op-code fetch cycle is generated during detection of an interrupt priority, new detection of that does not start. (Refer to Figure 4.6.1.) Since the state of the interrupt request bit and interrupt priority levels are latched during detection of interrupt priority, even if the bit state and priority levels change, the detection is performed on the previous state before it has changed.

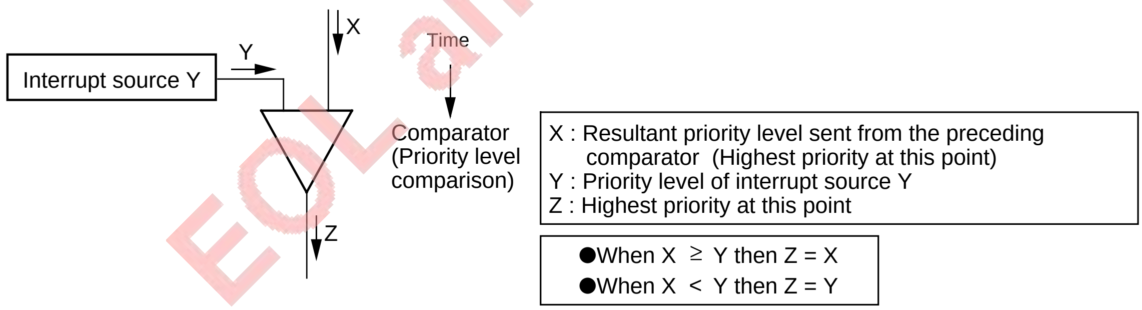
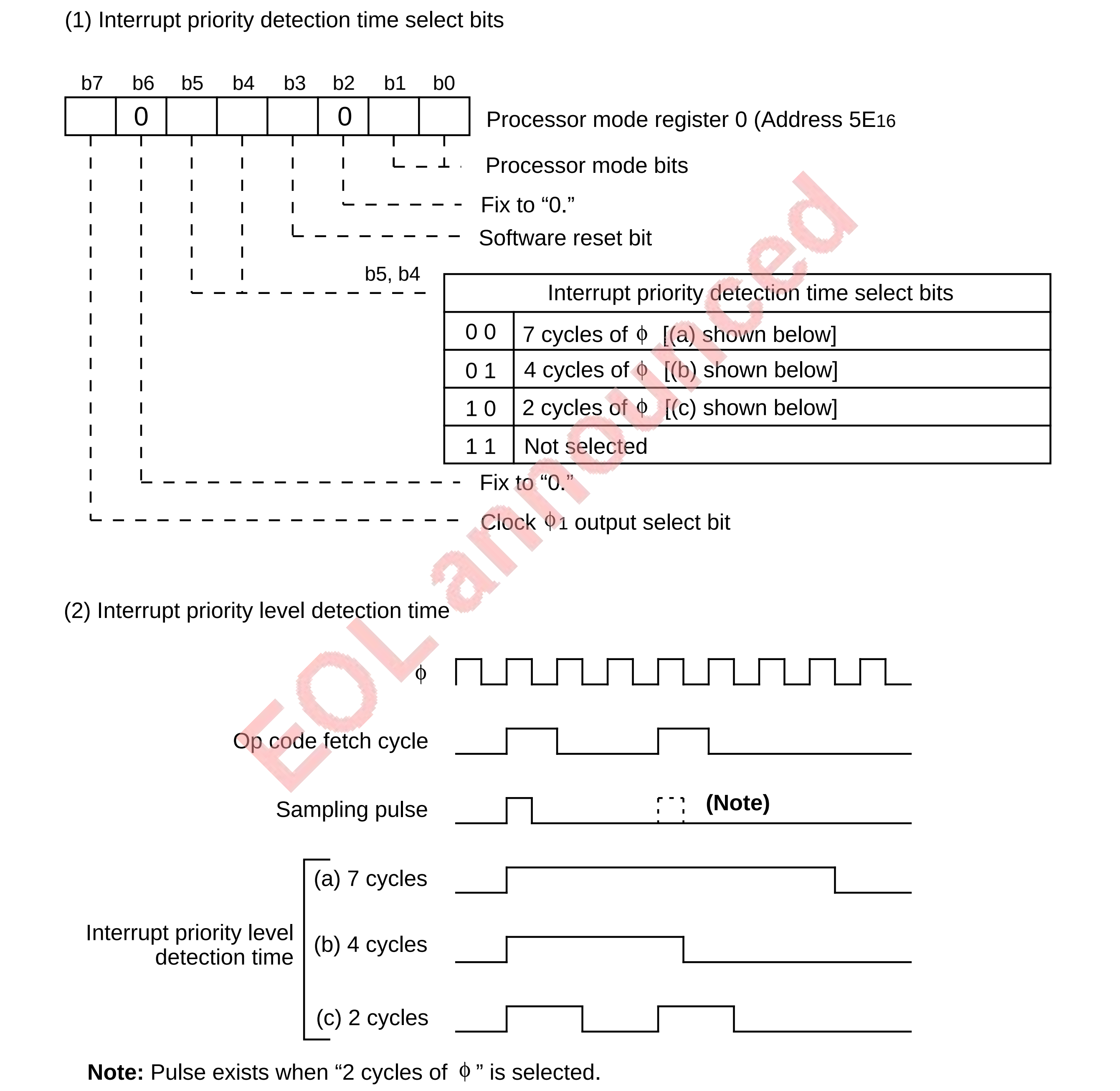


Fig. 4.5.2 Interrupt priority level detection model

4.6 Interrupt priority level detection time

After sampling had started, an interrupt priority level detection time has elapses before an interrupt request is accepted. The interrupt priority level detection time can be selected by software. Figure 4.6.1 shows the interrupt priority level detection time.
As the interrupt priority level detection time, normally select “2 cycles of internal clock ϕ.”



b5, b4	Interrupt priority detection time select bits
0 0	7 cycles of φ [(a) shown below]
0 1	4 cycles of φ [(b) shown below]
1 0	2 cycles of φ [(c) shown below]
1 1	Not selected

Fig. 4.6.1 Interrupt priority level detection time

4.7 Sequence from acceptance of interrupt request to execution of interrupt routine

The sequence from the acceptance of interrupt request to the execution of the interrupt routine is described below.

When an interrupt request is accepted, the interrupt request bit which corresponds to the accepted interrupt is cleared to “0,” and then the interrupt processing starts from the next cycle of completion of the instruction which is being executed at accepting the interrupt request. Figure 4.7.1 shows the sequence from acceptance of interrupt request to execution of interrupt routine.

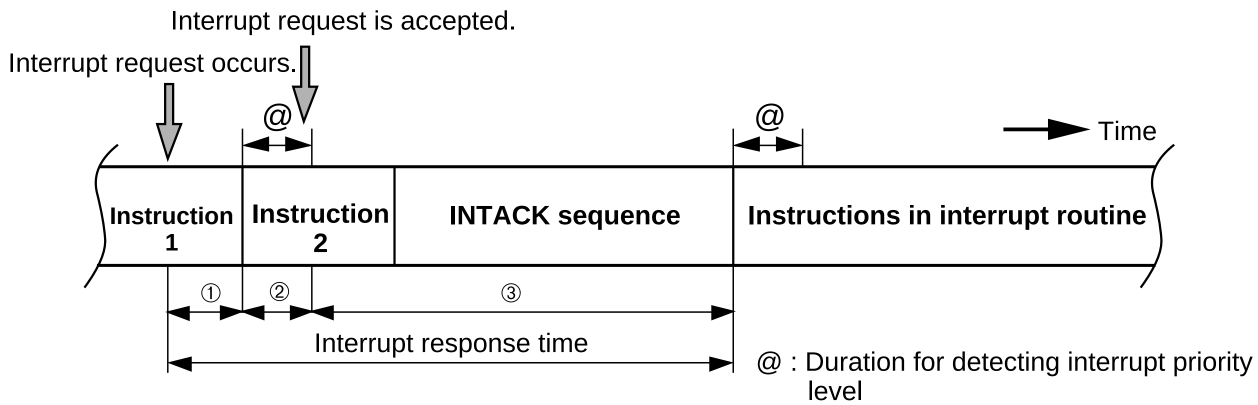
After execution of an instruction at accepting the interrupt request is completed, an INTACK (Interrupt Acknowledge) sequence is executed, and a branch is made to the start address of the interrupt routine allocated in addresses 0_{16} to $FFFF_{16}$.

The INTACK sequence is automatically performed in the following order.

① The contents of the program bank register (PG) just before performing the INTACK sequence are stored to stack.
② The contents of the program counter (PC) just before performing the INTACK sequence are stored to stack.
③ The contents of the processor status register (PS) just before performing the INTACK sequence is stored to stack.
④ The interrupt disable flag (I) is set to “1.”
⑤ The interrupt priority level of the accepted interrupt is set into the processor interrupt priority level (IPL).
⑥ The contents of the program bank register (PG) are cleared to “00_{16},” and the contents of the interrupt vector address are set into the program counter (PC).

Performing the INTACK sequence requires at least 15 cycles of internal clock ϕ. Figure 4.7.2 shows the INTACK sequence timing.

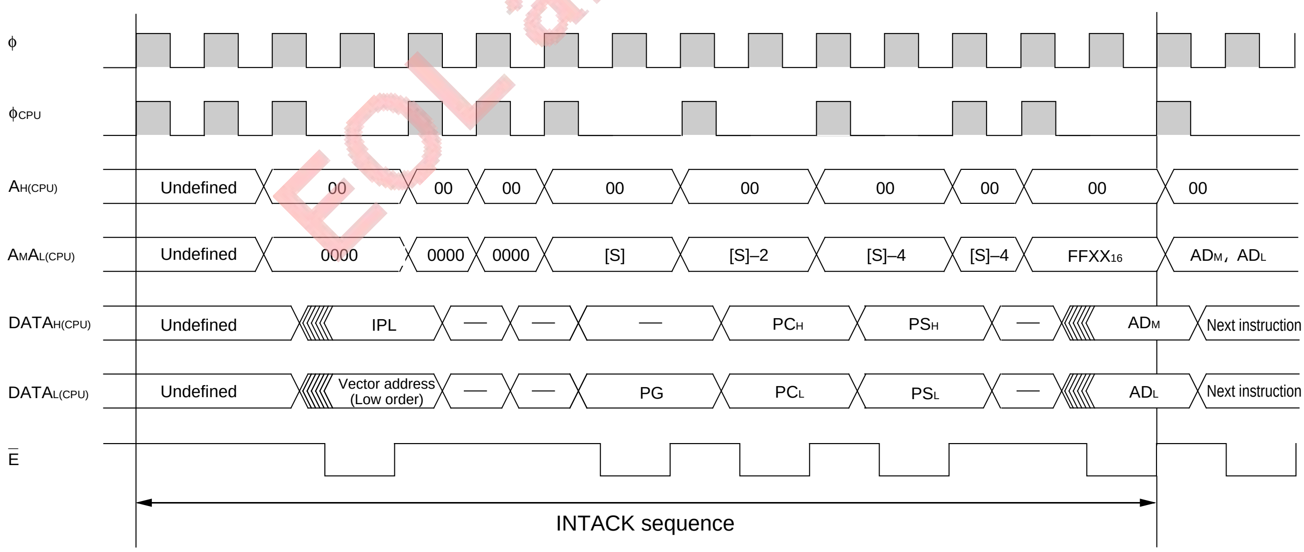
Execution is started beginning with an instruction at the start address of the interrupt routine after completing the INTACK sequence.



① Time from the occurrence of an interrupt request until the completion of executing an instruction which is being executed at the occurrence.
② Time from the instruction next to ① **(Note)** until the completion of executing an instruction which is being done at the end of priority detection
Note : At this time, interrupt priority detection starts.
③ Time required to execute the INTACK sequence (15 cycles of ϕ at minimum)

Fig. 4.7.1 Sequence from acceptance of interrupt request to execution of interrupt routine

●When 2ϕ access in low-speed running and stack pointer(S)'s content is even



[S] : Contents of stack pointer (S)
FFXX16 : Vector address

Fig. 4.7.2 INTACK sequence timing (at minimum)

4.7 Sequence from acceptance of interrupt request to execution of interrupt routine

4.7.1 Change in IPL at acceptance of interrupt request

When an interrupt request is accepted, the processor interrupt priority level (IPL) is replaced with the interrupt priority level of the accepted interrupt. This results in easy control of multiple interrupts. (Refer to section **"4.9 Multiple interrupts."**)

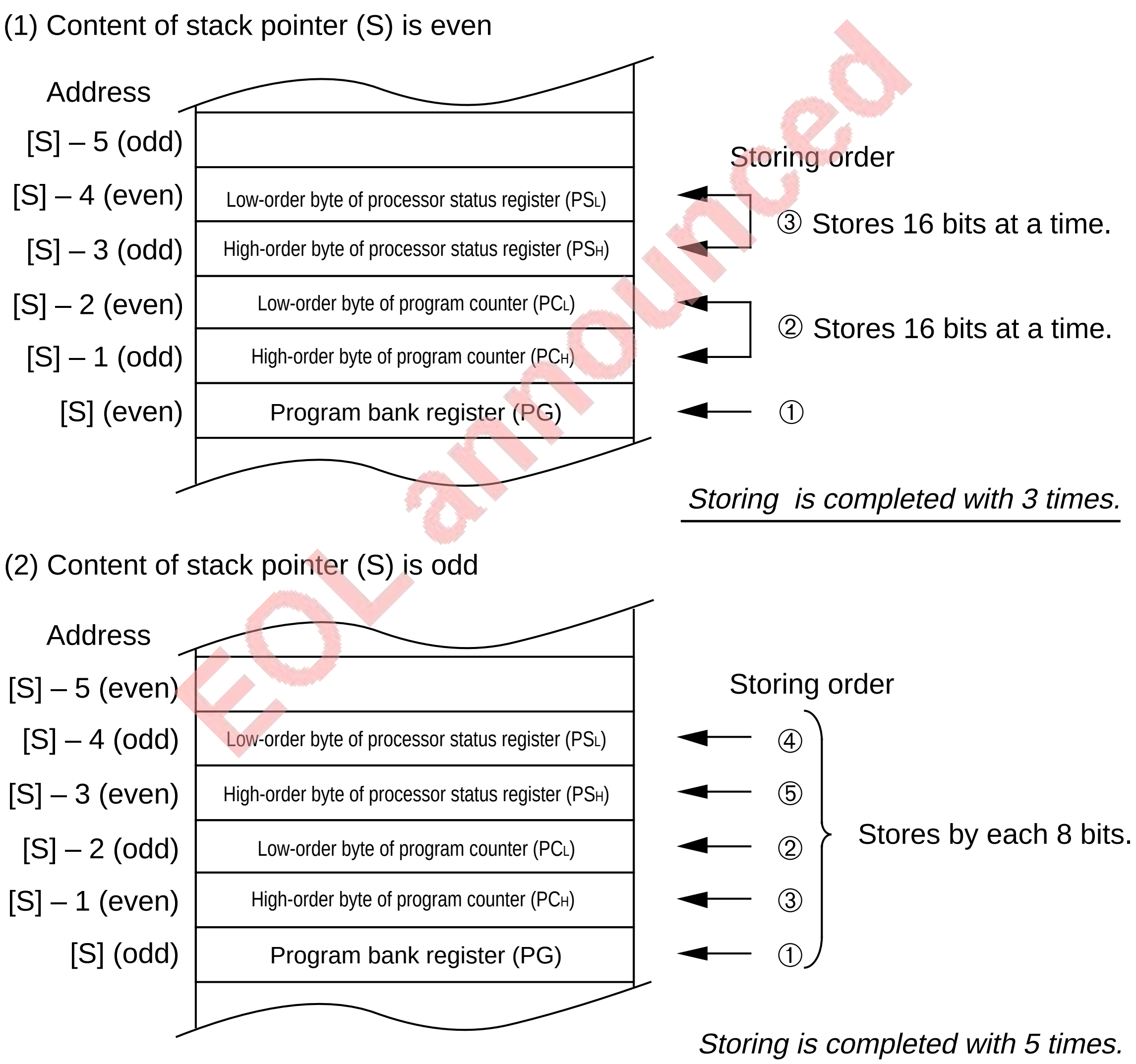
When at reset or the watchdog timer or the software interrupt is accepted, the value shown in Table 4.7.1 is set in the IPL.

Table 4.7.1 Change in IPL at interrupt request acceptance

Interrupt source	Change in IPL
Reset	Level 0 ("000_2") is set.
Watchdog timer	Level 7 ("111_2") is set.
Zero division	No change
BRK instruction	No change
Other interrupts	Interrupt priority level of the accepted interrupt request is set.

4.7.2 Storing registers

The register storing operation performed during INTACK sequence depends on whether the contents of the stack pointer (S) at accepting interrupt request are even or odd.
When the contents of the stack pointer (S) are even, the contents of the program counter (PC) and the processor status register (PS) are stored as a 16-bit unit simultaneously at each other. When the contents of the stack pointer (S) are odd, they are stored with twice by an 8-bit unit for each. Figure 4.7.3 shows the register storing operation.
In the INTACK sequence, only the contents of the program bank register (PG), program counter (PC), and processor status register (PS) are stored to the stack area. The other necessary registers must be stored by software at the beginning of the interrupt routine.
Using the **PSH** instruction can store all CPU registers except the stack pointer (S).



✻ [S] is an initial value that the stack pointer (S) indicates at accepting an interrupt request. The S's contents become [S] – 5 after storing the above registers.

Fig. 4.7.3 Register storing operation

4.8 Return from interrupt routine

When the **RTI** instruction is executed at the end of the interrupt routine, the contents of the program bank register (PG), program counter (PC), and processor status register (PS) immediately before performing the INTACK sequence, which were saved to the stack area, are automatically restored, and control returns to the routine executed before the acceptance of interrupt request and processing is resumed from it left off. For any register that is saved by software in the interrupt routine, restore it with the same data length and same register length as it was saved by using the **PUL** instruction and others before executing the **RTI** instruction.

4.9 Multiple interrupts

When a branch is made to the interrupt routine, the microcomputer becomes the following situation:

- Interrupt disable flag (I) = “1” (interrupts disabled)
- Interrupt request bit of the accepted interrupt = “0”
- Processor interrupt priority level (IPL) = interrupt priority level of the accepted interrupt

Accordingly, as long as the IPL remains unchanged, the microcomputer can accept the interrupt request that has higher priority than the interrupt request being executed now by clearing the interrupt disable flag (I) to “0” in the interrupt routine. This is multiple interrupts.
Figure 4.9.1 shows the multiple interrupt mechanism.
The interrupt requests that have not been accepted owing to their low priority levels are retained. When the **RTI** instruction is executed, the interrupt priority level of the routine that the microcomputer was executing before accepting the interrupt request is restored to the IPL. Therefore, one of the interrupt requests being retained is accepted when the following condition is satisfied at next detection of interrupt priority level:

Interrupt priority level of interrupt request being retained **>** Processor interrupt priority level (IPL)

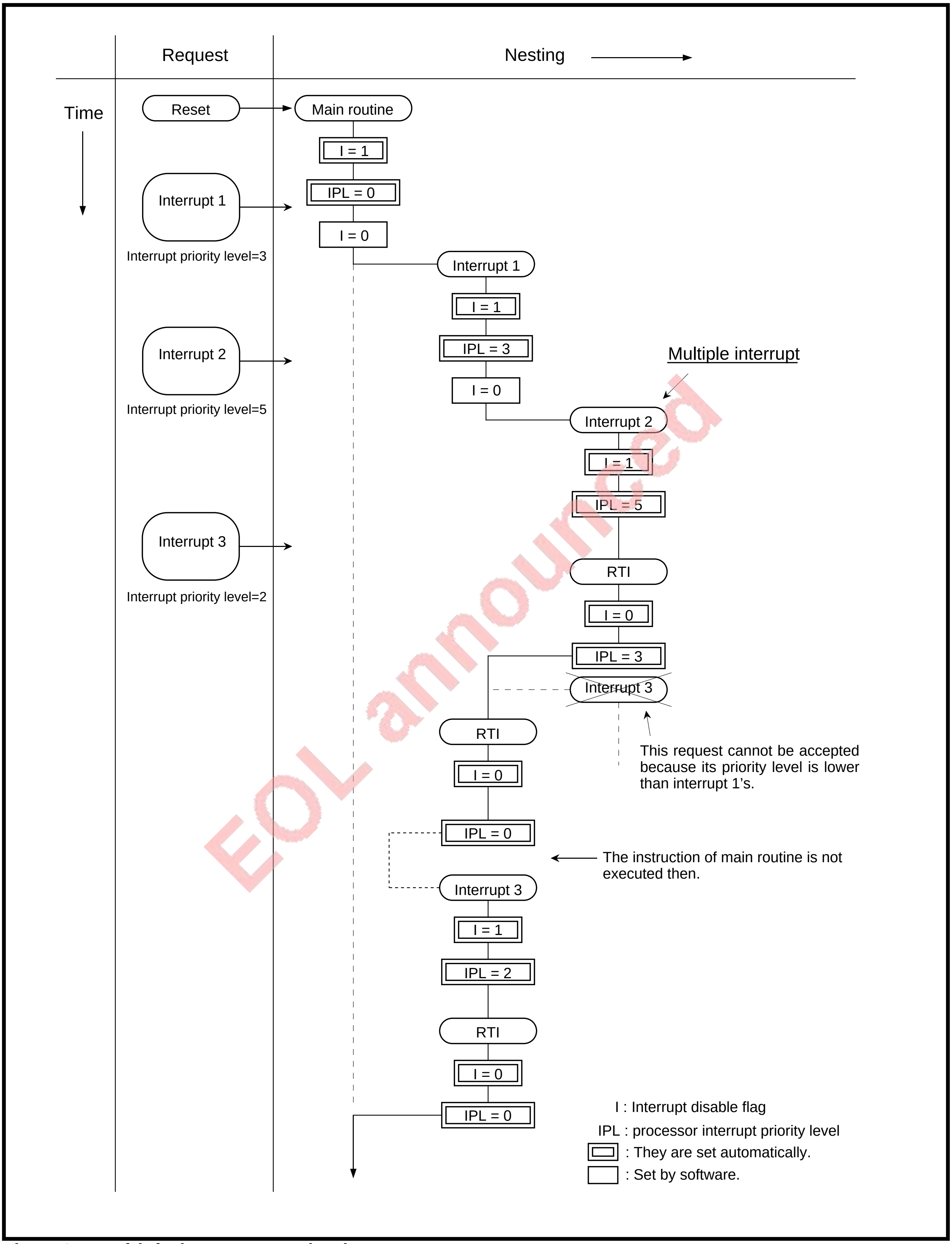


Fig. 4.9.1 Multiple interrupt mechanism

4.10 External interrupts ($\overline{INT_i}$ interrupt)

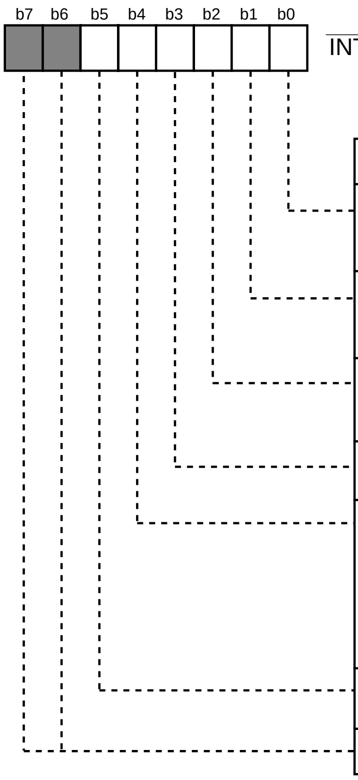
An external interrupt request occurs by input signals to the $\overline{INT_i}$ (i = 0 to 2) pin. The occurrence factor of interrupt request can be selected by the level sense/edge sense select bit and the polarity select bit (bits 5 and 4 at addresses $7D_{16}$ to $7F_{16}$) shown in Figure 4.10.1. Table 4.10.1 lists the occurrence factor of $\overline{INT_i}$ interrupt request.
When using $P6_2/\overline{INT_0}$ to $P6_4/\overline{INT_2}$ pins as input pins of external interrupts, set the corresponding bits at address 10_{16} (port P6 direction register) to “0.” (Refer to Figure 4.10.2.)
The signals input to the $\overline{INT_i}$ pin require “H” or “L” level width of 250 ns or more independent of the $f(X_{IN})$.
Additionally, even when using the pins $P6_2/\overline{INT_0}$ to $P6_4/\overline{INT_2}$ as the input pins of external interrupt, the user can obtain the pin’s state by reading bits 2 to 4 at address E_{16} (port P6 register).

Note: When selecting an input signal’s falling or “L” level as the occurrence factor of an interrupt request, make sure that the input signal is held “L” for 250 ns or more. When selecting an input signal’s rising or “H” level as that, make sure that the input signal is held “H” for 250 ns or more.

Table 4.10.1 Occurrence factor of $\overline{INT_i}$ interrupt request

b5	b4	$\overline{INT_i}$ interrupt request occurrence factor
0	0	Interrupt request occurs at falling of the signal input to the $\overline{INT_i}$ pin (edge sense).
0	1	Interrupt request occurs at rising of the signal input to the $\overline{INT_i}$ pin (edge sense).
1	0	Interrupt request occurs while the $\overline{INT_i}$ pin level is “H” (level sense).
1	1	Interrupt request occurs while the $\overline{INT_i}$ pin level is “L” (level sense).

The $\overline{INT_i}$ interrupt request occurs by always detecting the $\overline{INT_i}$ pin’s state. Accordingly, when the user does not use the $\overline{INT_i}$ interrupt, set the $\overline{INT_i}$ interrupt’s priority level to level 0.



$\overline{\text{INT}}_0$ to $\overline{\text{INT}}_2$ interrupt control registers (Addresses $7D_{16}$ to $7F_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 (Low level) 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7 (High level)	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit **(Note)**	0 : No interrupt request 1 : Interrupt request	0	RW
4	Polarity select bit	0 : Set the interrupt request bit at "H" level for level sense and at falling edge for edge sense. 1 : Set the interrupt request bit at "L" level for level sense and at rising edge for edge sense.	0	RW
5	Level sense/Edge sense select bit	0 : Edge sense 1 : Level sense	0	RW
7, 6	Nothing is assigned.		Undefined	–

Note: The $\overline{\text{INT}}_0$ to $\overline{\text{INT}}_2$ interrupt request bits are invalid when selecting the level sense.

Fig. 4.10.1 Structure of $\overline{\text{INT}}_i$ (i=0 to 2) interrupt control register

4.10 External interrupts ($\overline{INT_i}$ interrupt)

b7 b6 b5 b4 b3 b2 b1 b0

Port P6 direction register (Address 10_{16})

Bit	Corresponding pin	Functions	At reset	RW
0	$TA4_{OUT}$ pin	0 : Input mode 1 : Output mode When using pins as external interrupt input pins,set the corresponding bits to “0.”	0	RW
1	$TA4_{IN}$ pin		0	RW
2	$\overline{INT_0}$ pin		0	RW
3	$\overline{INT_1}$ pin		0	RW
4	$\overline{INT_2}$ pin		0	RW
5	$TB0_{IN}$ pin		0	RW
6	$TB1_{IN}$ pin		0	RW
7	$TB2_{IN}$ pin		0	RW

(shaded) : Bits 0, 1 and bits 5 to 7 are not used for external interrupts.

Fig. 4.10.2 Relationship between port P6 direction register and input pins of external interrupt

4.10.1 Function of $\overline{INT_i}$ interrupt request bit

(1) Selecting edge sense mode

The interrupt request bit has the same function as that of internal interrupts. That is, when an interrupt request occurs, the interrupt request bit is set to “1.” The bit remains set to “1” until the interrupt request is accepted; it is cleared to “0” when the interrupt request is accepted. By software, this bit also can be set to “0” in order to clear the interrupt request or “1” in order to generate the interrupt request.

(2) Selecting level sense mode

The $\overline{INT_i}$ interrupt request bit becomes ignored.
In this case, the interrupt request occurs continuously while the level of the $\overline{INT_i}$ pin is valid level*1.
When the $\overline{INT_i}$ pin level changes from the valid level to the invalid level*2 before the $\overline{INT_i}$ interrupt request is accepted, this interrupt request is not retained. (Refer to Figure 4.10.4.)

Valid level*1: This means the level which is selected by the polarity select bit (bit 4 at addresses $7D_{16}$ to $7F_{16}$).
Invalid level*2: This means the reversed level of a valid level.

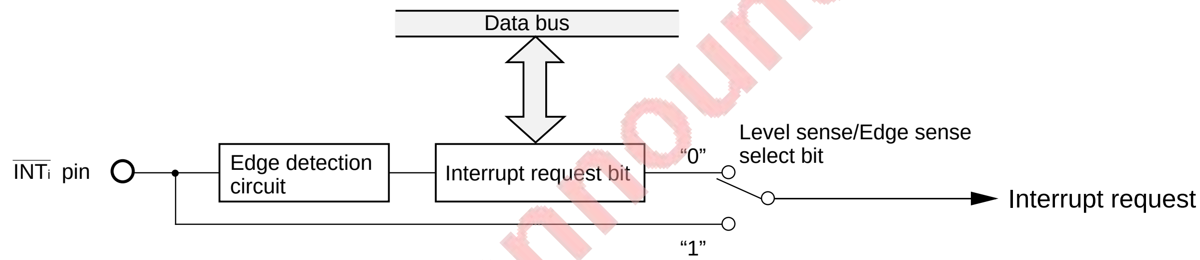


Fig. 4.10.3 Circuit of $\overline{INT_i}$ Interrupt

4.10 External interrupts ($\overline{INT_i}$ interrupt)

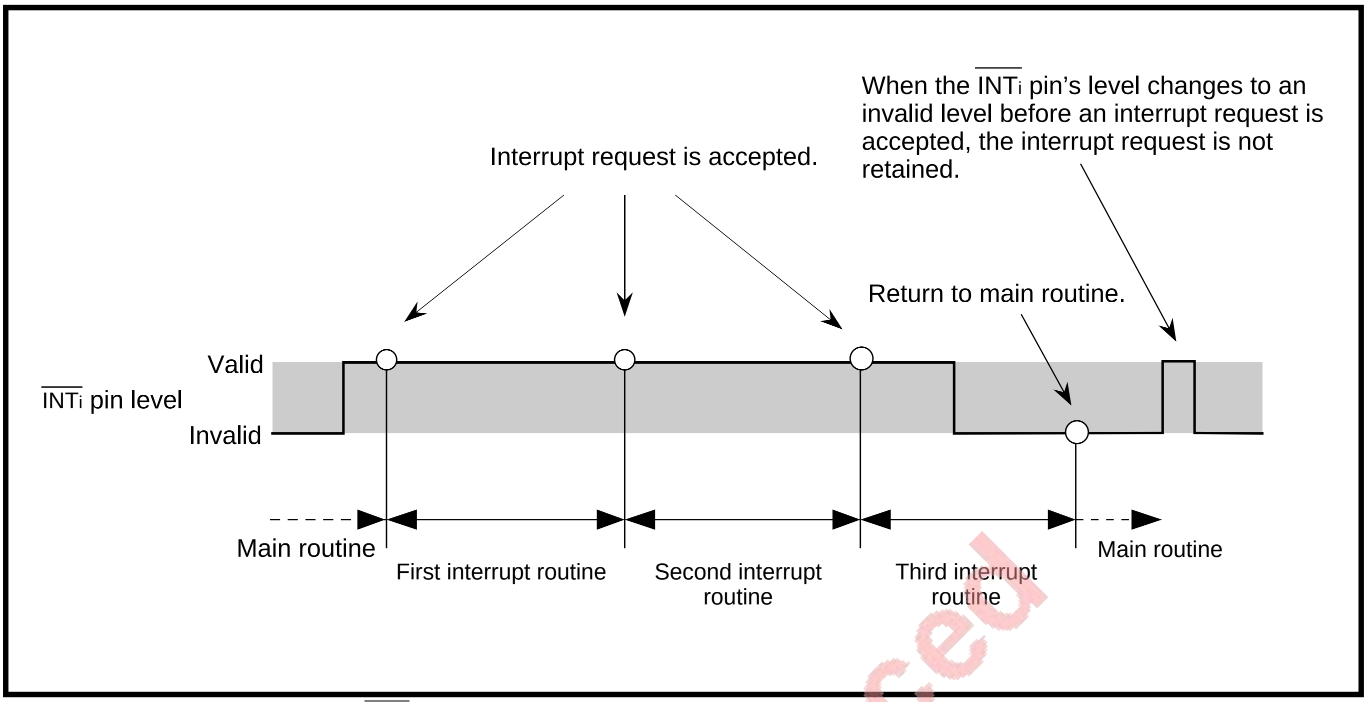
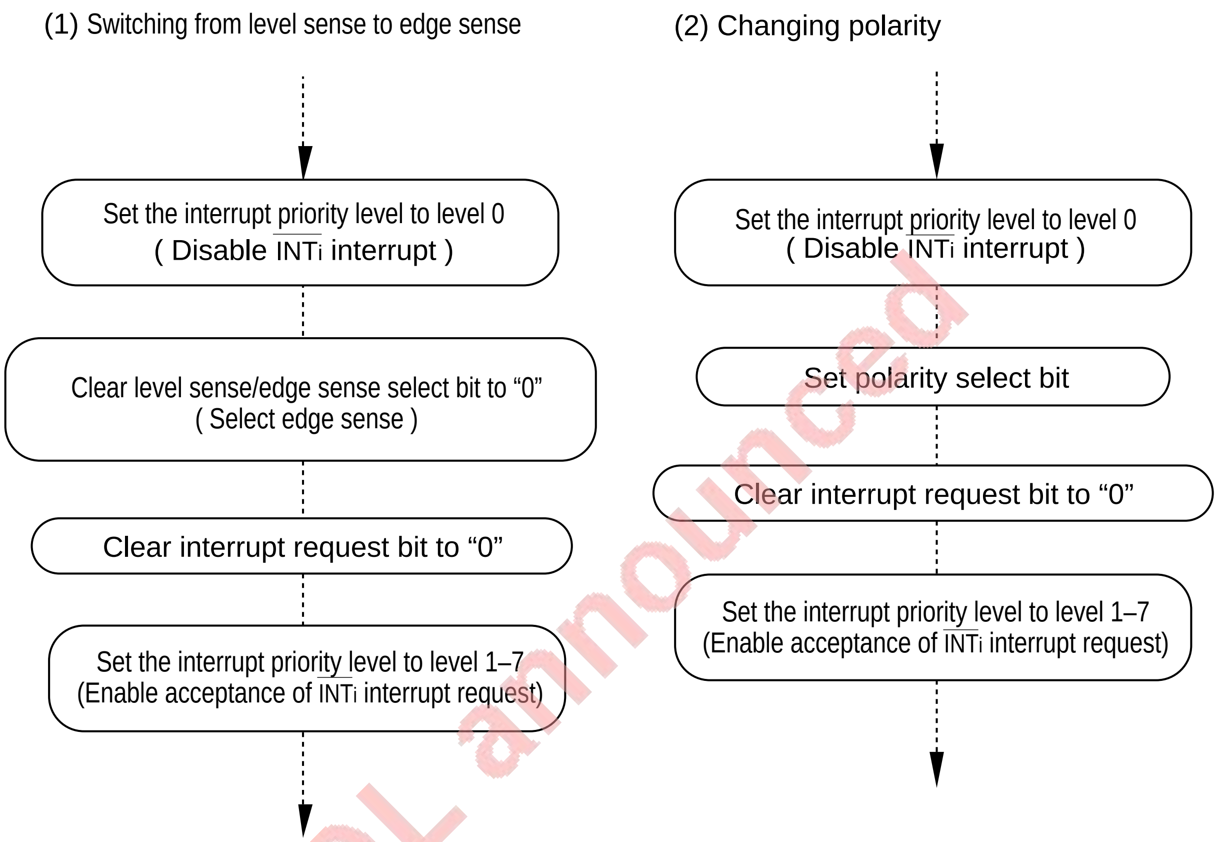


Fig. 4.10.4 Occurrence of $\overline{INT_i}$ interrupt request in level sense mode

4.10.2 Switch of occurrence factor of $\overline{\mathrm{INT}_i}$ interrupt request

To switch the occurrence factor of $\overline{\mathrm{INT}_i}$ interrupt request from the level sense to the edge sense, set the $\overline{\mathrm{INT}_i}$ interrupt control register in the sequence shown in Figure 4.10.5 (1). To change the polarity, set the $\overline{\mathrm{INT}_i}$ interrupt control register in the sequence shown in Figure 4.10.5 (2).



Note: Follow the above procedure each. Do not perform 2 or more setting at the same time, with 1 instruction.

Fig. 4.10.5 Switching flow of occurrence factor of $\overline{\mathrm{INT}_i}$ interrupt request

4.11 Precautions when using interrupts

To change the interrupt priority level select bits (bits 0 to 2 at addresses 70_{16} to $7F_{16}$), 2 to 7 cycles of ϕ are required after executing an write-instruction until completion of the interrupt priority level's change. Accordingly, it is necessary to reserve enough time by software when changing the interrupt priority level of which interrupt source is the same within a very short execution time consisting of a few instructions.
Figure 4.11.1 shows a program example to reserve time required for changing interrupt priority level. The time for change depends on the interrupt priority detection timer select bits (bits 4 and 5 at address $5E_{16}$). Table 4.11.1 lists the relation between the number of instructions to be inserted with program example of Figure 4.11.1 and the interrupt priority detection time select bits.

```
:
LDM.B #0XH, 007XH  ; Write to interrupt priority level select bits
NOP                ; Insert NOP instruction (Note)
NOP                ;
NOP                ;
LDM.B #0XH, 007XH  ; Write to interrupt priority level select bits
:
```

Note: All instructions (other than instructions for writing to address $7X_{16}$) which have the same cycles as **NOP** instruction can also be inserted. Confirm the number of instructions to be inserted by Table 4.11.1.

Fig. 4.11.1 Program example to reserve time required for changing interrupt priority level

Table 4.11.1 Relation between number of instructions to be inserted with program example of Figure 4.11.1 and interrupt priority detection time select bits

Interrupt priority detection time select bits **(Note)**		Interrupt priority level detection time	Number of inserted instructions
b5	b4		
0	0	7 cycles of ϕ	**NOP** instruction 4 or more
0	1	4 cycles of ϕ	**NOP** instruction 2 or more
1	0	2 cycles of ϕ	**NOP** instruction 1 or more
1	1	Do not select.	

Note: We recommend [b5 = "1", b4 = "0"].

CHAPTER 5

TIMER A

Timer A is used primarily for output to externals. It consists of five counters, timers A0 to A4, each equipped with a 16-bit reload function. Timers A0 to A4 operate independently of one another.

5.1 Overview

Timer Ai (i = 0 to 4) has four operating modes listed below. Except for the event counter mode, Timers A0 to A4 all have the same functions.

● **Timer mode**

The timer counts an internally generated count source. Following functions can be used in this mode:
•Gate function
•Pulse output function

● **Event counter mode**

The timer counts an external signal. Following functions can be used in this mode:
•Pulse output function
•Two-phase pulse signal processing function (Timers A2, A3, and A4)

● **One-shot pulse mode**

The timer outputs a pulse which has an arbitrary width once.

● **Pulse width modulation (PWM) mode**

Timer outputs pulses which have an arbitrary width in succession. The timer functions as which pulse width modulator as follows:
•16-bit pulse width modulator
•8-bit pulse width modulator

5.2 Block description

Figure 5.2.1 shows the block diagram of Timer A. Explanation of relevant registers to Timer A is described below.

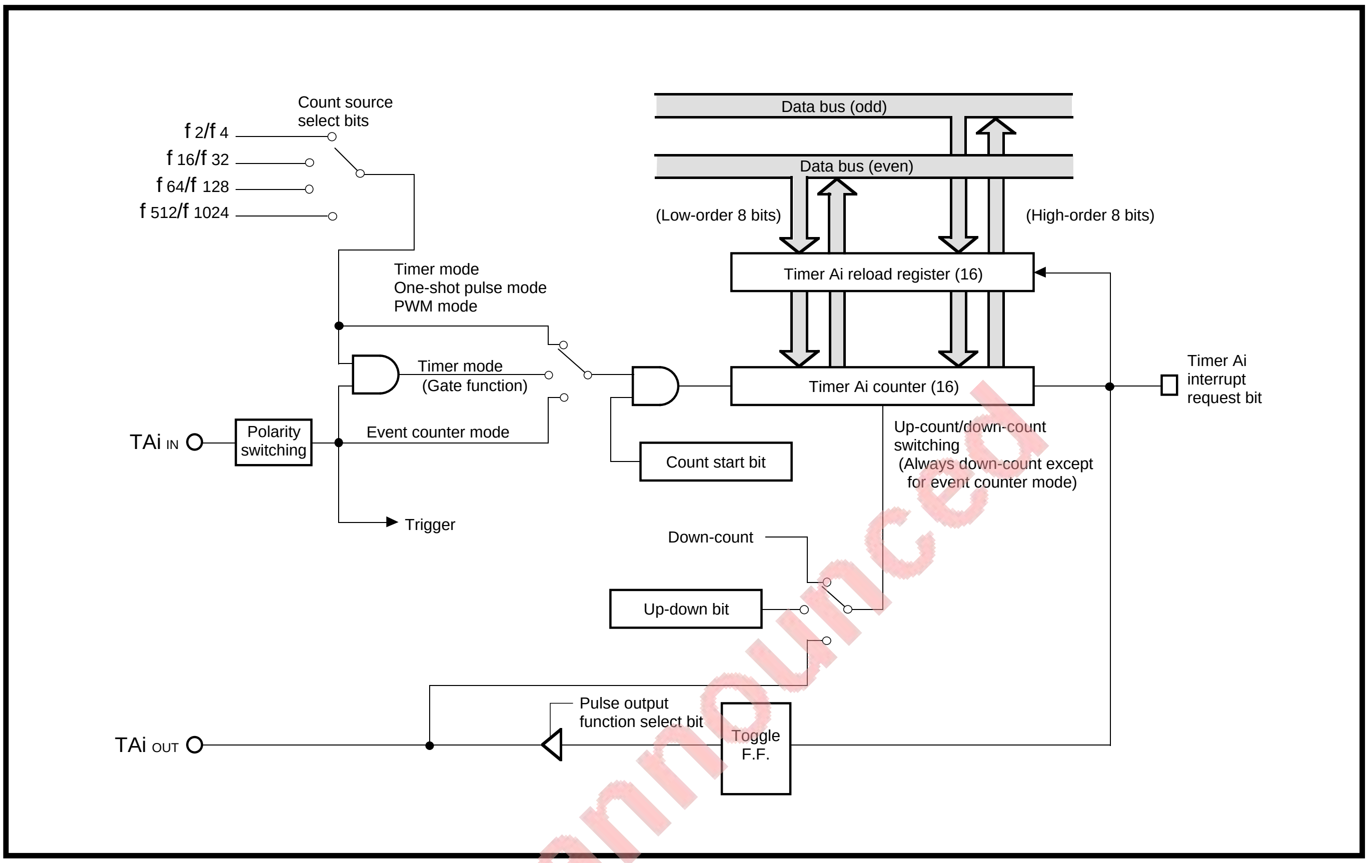


Fig. 5.2.1 Block diagram of Timer A

5.2 Block description

5.2.1 Counter and reload register (timer Ai register)

Each of timer Ai counter and reload register consists of 16 bits.
The counter down-counts each time the count source is input. In the event counter mode, it can also function as an up-counter.
The reload register is used to store the initial value of the counter. When the counter underflows or overflows, the reload register's contents are reloaded into the counter.
Values are set to the counter and reload register by writing a value to the timer Ai register. Table 5.2.1 lists the memory assignment of the timer Ai register.
The value written into the timer Ai register when counting is not in progress is set to the counter and reload register. The value written into the timer Ai register when counting is in progress is set to only the reload register. In this case, the reload register's updated contents are transferred to the counter at the next reload time. The value got when reading out the timer Ai register varies according to the operating mode.
Table 5.2.2 lists reading and writing from and to the timer Ai register.

Table 5.2.1 Memory assignment of timer Ai register

Timer Ai register	High-order byte	Low-order byte
Timer A0 register	Address 47_{16}	Address 46_{16}
Timer A1 register	Address 49_{16}	Address 48_{16}
Timer A2 register	Address $4B_{16}$	Address $4A_{16}$
Timer A3 register	Address $4D_{16}$	Address $4C_{16}$
Timer A4 register	Address $4F_{16}$	Address $4E_{16}$

Note: When reset, the contents of the timer Ai register are undefined.

Table 5.2.2 Reading and writing from and to timer Ai register

Operating mode	Read	Write
Timer mode	Counter value is read out. (**Note 1**)	<During counting> Written to only reload register. <When not counting> Written to both counter and reload register.
Event counter mode		
One-shot pulse mode	Undefined value is read out.	
Pulse width modulation (PWM) mode		

Notes 1: Also refer to **"[Precautions when operating in timer mode]"** and **"[Precautions when operating in event counter mode]."**
2: When reading and writing to/from the timer Ai register, perform them in a unit of 16 bits.

5.2.2 Count start register

This register is used to start and stop counting. Each bit of this register corresponds to each timer. Figure 5.2.2 shows the structure of the count start register.

b7 b6 b5 b4 b3 b2 b1 b0

Count start register (Address 40_{16})

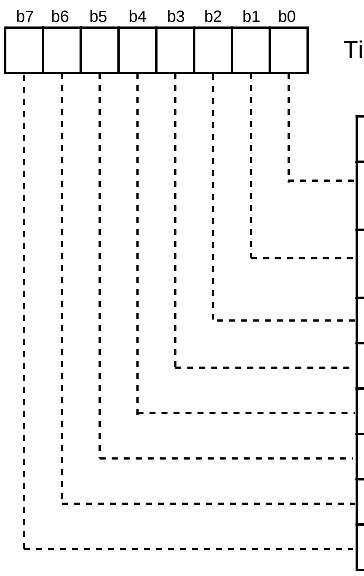
Bit	Bit name	Functions	At reset	RW
0	Timer A0 count start bit	0 : Stop counting 1 : Start counting	0	RW
1	Timer A1 count start bit		0	RW
2	Timer A2 count start bit		0	RW
3	Timer A3 count start bit		0	RW
4	Timer A4 count start bit		0	RW
5	Timer B0 count start bit		0	RW
6	Timer B1 count start bit		0	RW
7	Timer B2 count start bit		0	RW

(shaded) : Bits 7 to 5 are not used for Timer A.

Fig. 5.2.2 Structure of count start register

5.2.3 Timer Ai mode register

Figure 5.2.3 shows the structure of the timer Ai mode register. Operating mode select bits are used to select the operating mode of timer Ai. Bits 2 to 7 have different functions according to the operating mode. These bits are described in the paragraph of each operating mode.



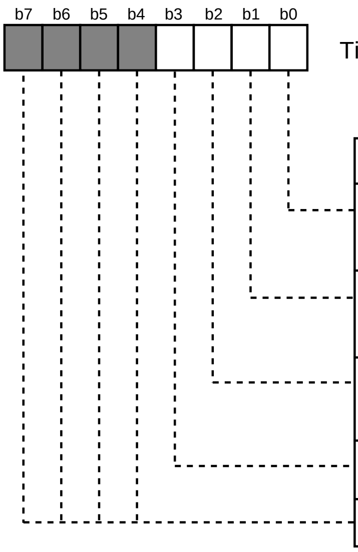
Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot pulse mode 1 1 : Pulse width modulation (PWM) mode	0	RW
1			0	RW
2	These bits have different functions according to the operating mode.		0	RW
3			0	RW
4			0	RW
5			0	RW
6			0	RW
7			0	RW

Fig. 5.2.3 Structure of timer Ai mode register

5.2.4 Timer Ai interrupt control register

Figure 5.2.4 shows the structure of the timer Ai interrupt control register. For details about interrupts, refer to **"Chapter 4. INTERRUPTS."**



Timer Ai interrupt control registers (i = 0 to 4) (Addresses 75_{16} to 79_{16})

Bit	Bit name	Functions	At reset	RW
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 (Low level) 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7 (High level)	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	RW
7 to 4	Nothing is assigned.		Undefined	–

Fig. 5.2.4 Structure of timer Ai interrupt control register

(1) Interrupt priority level select bits (bits 2 to 0)

These bits select a timer Ai interrupt's priority level. When using timer Ai interrupts, select priority levels 1 to 7. When a timer Ai interrupt request occurs, its priority level is compared with the processor interrupt priority level (IPL), so that the requested interrupt is enabled only when its priority level is higher than the IPL. (However, this applies when the interrupt disable flag (I) = "0.") To disable timer Ai interrupts, set these bits to "000_2" (level 0).

(2) Interrupt request bit (bit 3)

This bit is set to "1" when the timer Ai interrupt request occurs. This bit is automatically cleared to "0" when the timer Ai interrupt request is accepted. This bit can be set to "1" or "0" by software.

5.2.5 Port P5 and port P6 direction registers

The I/O pins of Timers A0 to A3 are shared with port P5, and the I/O pins of Timer A4 are shared with port P6. When using these pins as Timer Ai's input pins, set the corresponding bits of the port P5 and port P6 direction registers to "0" to set these ports for the input mode. When used as Timer Ai's output pins, these pins are forcibly set to output pins of Timer Ai regardless of the direction registers's contents. Figure 5.2.5 shows the relationship between the port P5 and port P6 direction registers and the Timer Ai's I/O pins.

b7 b6 b5 b4 b3 b2 b1 b0

Port P5 direction register (Address D_{16})

Bit	Corresponding pin name	Functions	At reset	RW
0	$TA0_{OUT}$ pin	0: Input mode 1: Output mode When using these pins as Timer Ai's input pins, set the corresponding bits to "0."	0	RW
1	$TA0_{IN}$ pin		0	RW
2	$TA1_{OUT}$ pin		0	RW
3	$TA1_{IN}$ pin		0	RW
4	$TA2_{OUT}$ pin		0	RW
5	$TA2_{IN}$ pin		0	RW
6	$TA3_{OUT}$ pin		0	RW
7	$TA3_{IN}$ pin		0	RW

b7 b6 b5 b4 b3 b2 b1 b0

Port P6 direction register (Address 10_{16})

Bit	Corresponding pin name	Functions	At reset	RW
0	$TA4_{OUT}$ pin	0: Input mode 1: Output mode When using these pins as Timer Ai's input pins, set the corresponding bits to "0."	0	RW
1	$TA4_{IN}$ pin		0	RW
2	$\overline{INT_0}$ pin		0	RW
3	$\overline{INT_1}$ pin		0	RW
4	$\overline{INT_2}$ pin		0	RW
5	$TB0_{IN}$ pin		0	RW
6	$TB1_{IN}$ pin		0	RW
7	$TB2_{IN}$ pin		0	RW

(shaded) : Bits 7 to 2 are not used for Timer A.

Fig. 5.2.5 Relationship between port P5 and port P6 direction registers and Timer Ai's I/O pins

5.3 Timer mode

In this mode, the timer counts an internally generated count source. (Refer to Table 5.3.1.) Figure 5.3.1 shows the structures of the timer Ai mode register and timer Ai register in the timer mode.

Table 5.3.1 Specifications of timer mode

Item	Specifications
Count source	f_2/f_4, f_{16}/f_{32}, f_{64}/f_{128}, or f_{512}/f_{1024}
Count operation	• Down-count • When the counter underflows, reload register's contents are reloaded and counting continues.
Count start condition	When count start bit is set to "1."
Count stop condition	When count start bit is cleared to "0."
Interrupt request occurrence timing	When the counter underflows.
TAi_{IN} pin function	Programmable I/O port or gate input
TAi_{OUT} pin function	Programmable I/O port or pulse output
Read from timer Ai register	Counter value can be read out.
Write to timer Ai register	● While counting is stopped When a value is written to timer Ai register, it is written to both reload register and counter. ● While counting is in progress When a value is written to timer Ai register, it is written to only reload register. (Transferred to counter at next reload timing.)

b7 b6 b5 b4 b3 b2 b1 b0

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 0 : Timer mode	0	RW
1			0	RW
2	Pulse output function select bit	0 : No pulse output (TAi$_{OUT}$ pin functions as a programmable I/O port.) 1 : Pulse output (TAi$_{OUT}$ pin functions as a pulse output pin.)	0	RW
3	Gate function select bits	b4 b3 0 0 : } No gate function 0 1 : } (TAi$_{IN}$ pin functions as a programmable I/O port.) 1 0 : Gate function (Counter counts only while TAi$_{IN}$ pin's input signal is "L" level.) 1 1 : Gate function (Counter counts only while TAi$_{IN}$ pin's input signal is "H" level.)	0	RW
4			0	RW
5	Fix this bit to "0" in the timer mode.		0	RW
6	Count source select bits	b7 b6 0 0 : f_2/f_4 0 1 : f_{16}/f_{32} 1 0 : f_{64}/f_{128} 1 1 : f_{512}/f_{1024}	0	RW
7			0	RW

(b15) (b8)
b7 b0 b7 b0

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

Bit	Functions	At reset	RW
15 to 0	These bits can be set to "0000_{16}" to "$FFFF_{16}$." Assuming that the set value = n, the counter divides the count source frequency by n + 1. When reading, the register indicates the counter value.	Undefined	RW

Fig. 5.3.1 Structures of timer Ai mode register and timer Ai register in timer mode

5.3.1 Setting for timer mode

Figures 5.3.2 and 5.3.3 show an initial setting example for registers relevant to the timer mode.
Note that when using interrupts, set up to enable the interrupts. For details, refer to section **"Chapter 4. INTERRUPTS."**

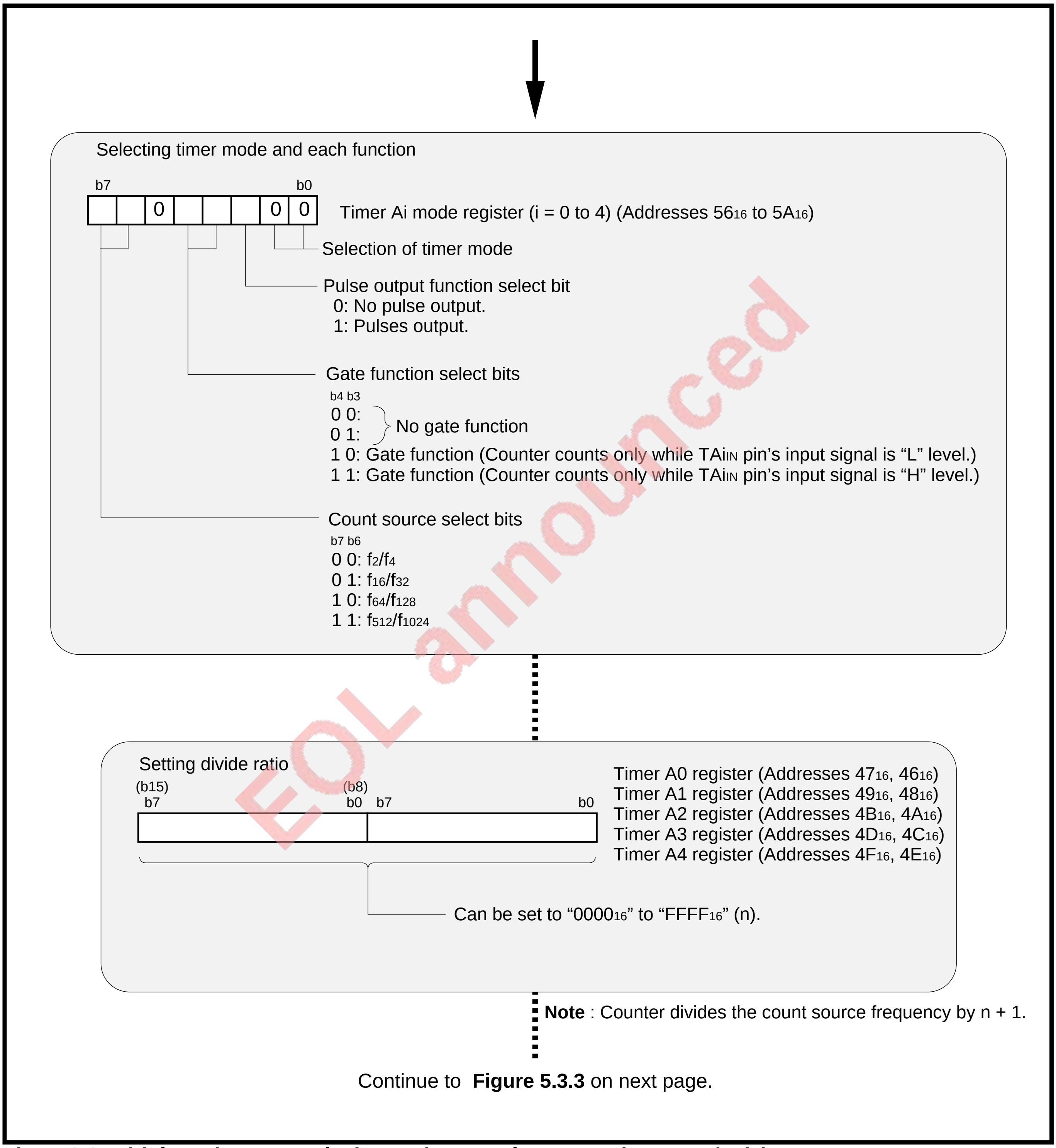


Fig. 5.3.2 Initial setting example for registers relevant to timer mode (1)

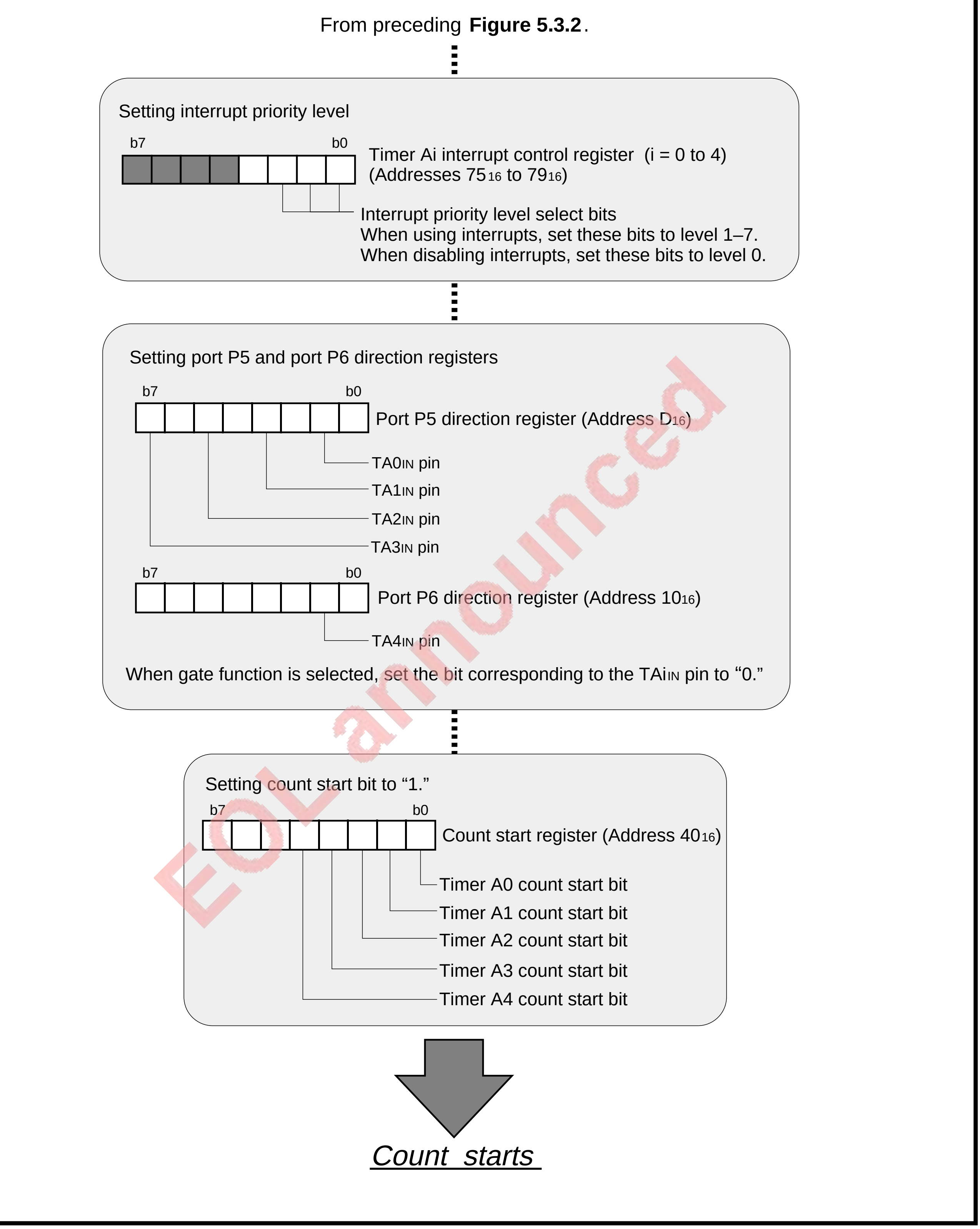


Fig. 5.3.3 Initial setting example for registers relevant to timer mode (2)

5.3.2 Count source

In the timer mode, the count source select bits (bits 6 and 7 at addresses 56_{16} to $5A_{16}$) select the count source. Table 5.3.2 lists the count source frequency.

Table 5.3.2 Count source frequency

Count source select bits		$f(X_{IN})$ = 25 MHz				$f(X_{IN})$ = 40 MHz	
		Clock source for peripheral devices select bit = "0"		Clock source for peripheral devices select bit = "1"		Clock source for peripheral devices select bit = "0"	
b7	b6	Count source	Frequency	Count source	Frequency	Count source	Frequency
0	0	f_4	6.25 MHz	f_2	12.5 MHz	f_4	10 MHz
0	1	f_{32}	781.25 kHz	f_{16}	1.5625 MHz	f_{32}	1.25 MHz
1	0	f_{128}	195.3125 kHz	f_{64}	390.625 kHz	f_{128}	312.5 kHz
1	1	f_{1024}	24.4141 kHz	f_{512}	48.8281 kHz	f_{1024}	39.0625 kHz

Clock source for peripheral devices select bit : bit 2 at address $5F_{16}$

5.3.3 Operation in timer mode

① When the count start bit is set to "1," the counter starts counting of the count source.
② When the counter underflows, the reload register's contents are reloaded and counting continues.
③ The timer Ai interrupt request bit is set to "1" when the counter underflows in ②. The interrupt request bit remains set to "1" until the interrupt request is accepted or the interrupt request bit is cleared to "0" by software.

Figure 5.3.4 shows an example of operation in the timer mode.

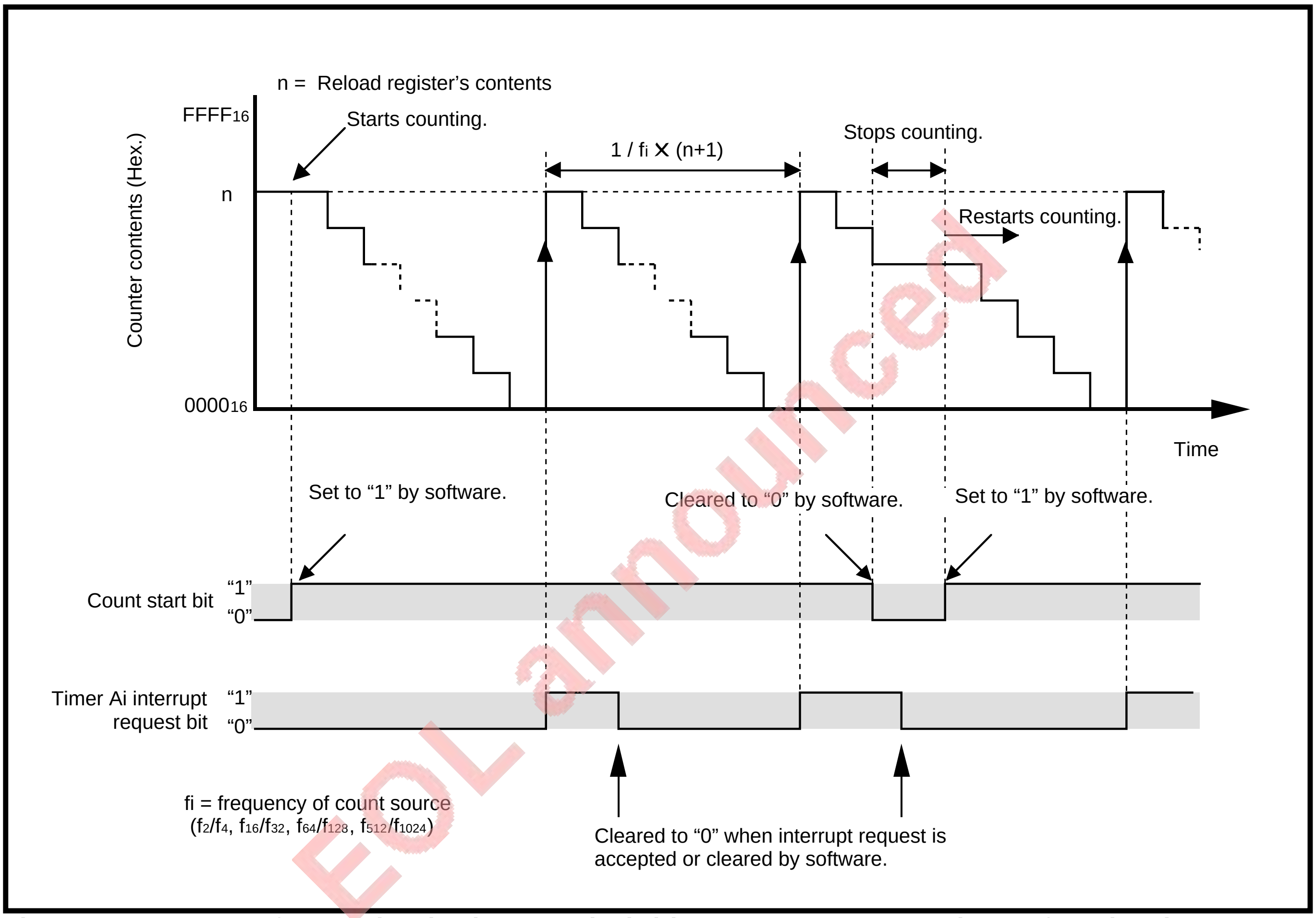


Fig. 5.3.4 Example of operation in timer mode (without pulse output and gate functions)

5.3.4 Select function

The following describes the selective gate and pulse output functions.

(1) Gate function

The gate function is selected by setting the gate function select bits (bits 4 and 3 at addresses 56_{16} to $5A_{16}$) to "10_2" or "11_2." The gate function makes it possible to start or stop counting depending on the TAi_{IN} pin's input signal. Table 5.3.3 lists the count valid levels.

Figure 5.3.5 shows an example of operation selecting the gate function.

When selecting the gate function, set the port P5 and port P6 direction registers' bits which correspond to the TAi_{IN} pin for the input mode. Additionally, make sure that the TAi_{IN} pin's input signal has a pulse width equal to or more than two cycles of the count source.

Table 5.3.3 Count valid levels

Gate function select bits		Count valid level (Duration when counter counts)
b4	b3	
1	0	While TAi_{IN} pin's input signal is "L" level
1	1	While TAi_{IN} pin's input signal is "H" level

Note: The counter does not count while the TAi_{IN} pin's input signal is not at the count valid level.

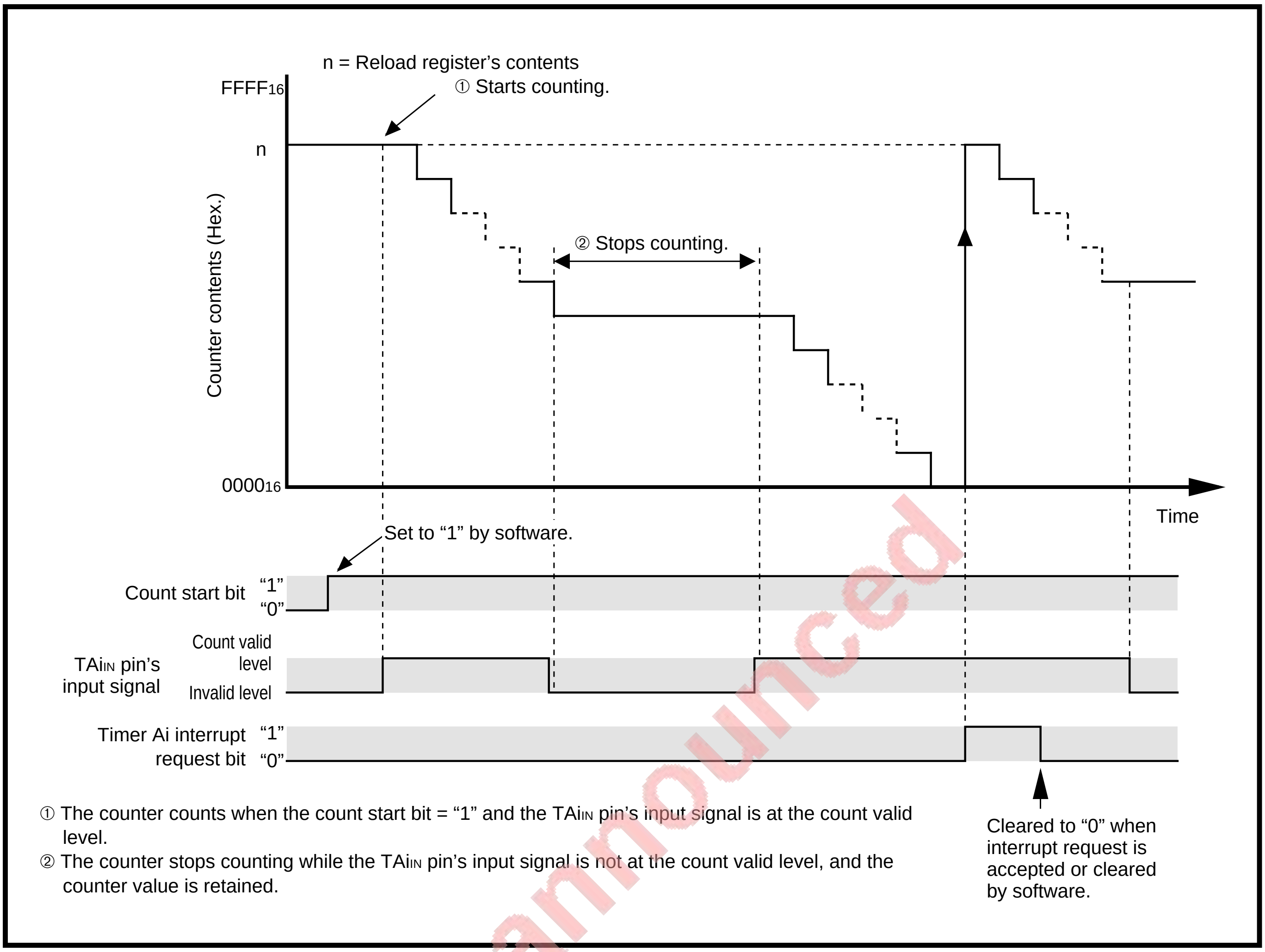


Fig. 5.3.5 Example of operation selecting gate function

(2) Pulse output function

The pulse output function is selected by setting the pulse output function select bit (bit 2 at addresses 56_{16} to $5A_{16}$) to "1." When this function is selected, the TAi_{OUT} pin is forcibly set for the pulse output pin regardless of the corresponding bits of the port P5 and port P6 direction registers. The TAi_{OUT} pin outputs pulses of which polarity is inverted each time the counter underflows.
When the count start bit (address 40_{16}) is "0" (count stopped), the TAi_{OUT} pin outputs "L" level. Figure 5.3.6 shows an example of operation selecting the pulse output function.

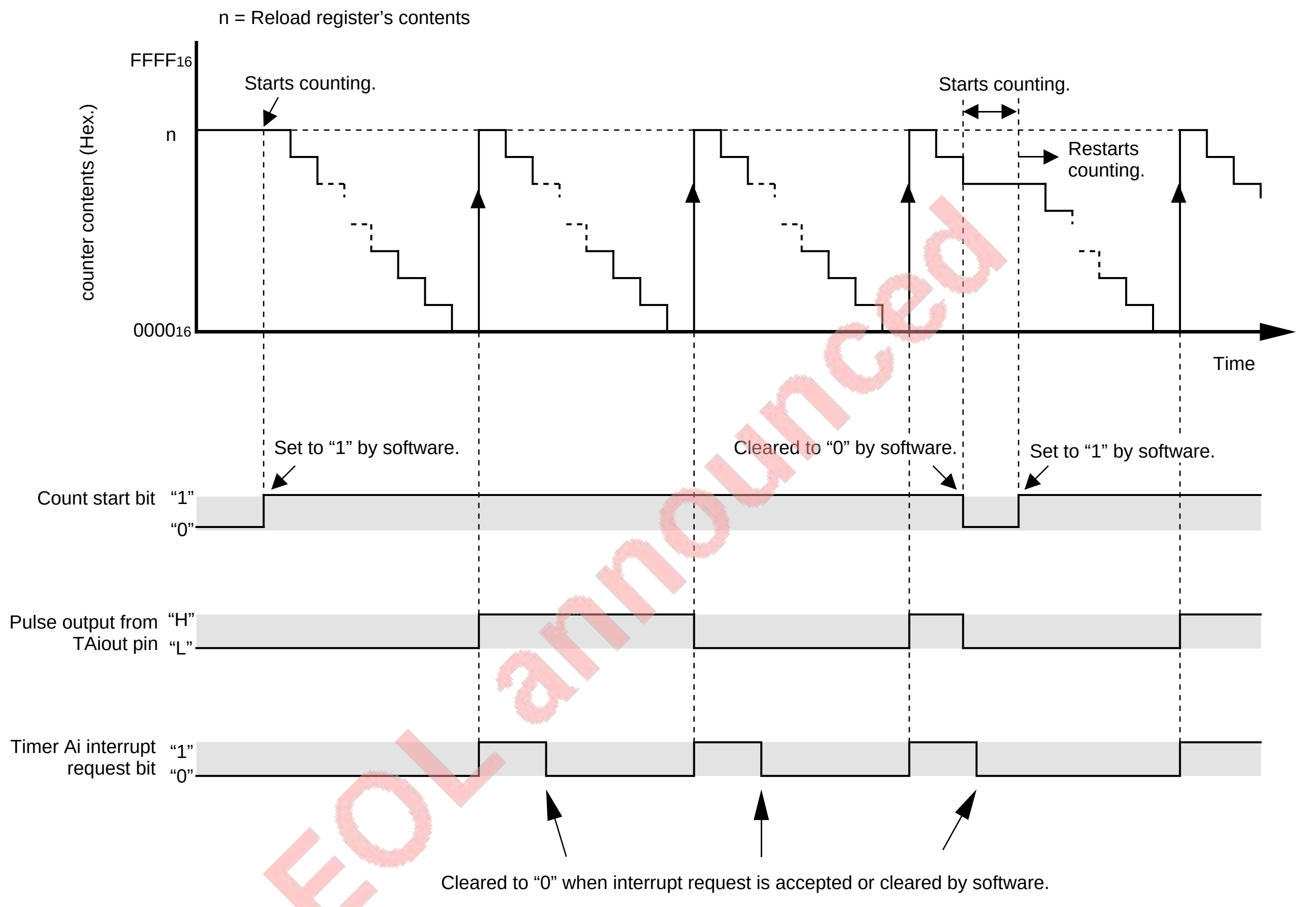


Fig. 5.3.6 Example of operation selecting pulse output function

[Precautions when operating in timer mode]

By reading the timer Ai register, the counter value can be read out at any timing while counting is in progress. However, if the timer Ai register is read at the reload timing shown in Figure 5.3.7, the value "$FFFF_{16}$" is read out. When reading the timer Ai register after setting a value to the register while counting is not in progress and before the counter starts counting, the set value is read out correctly.

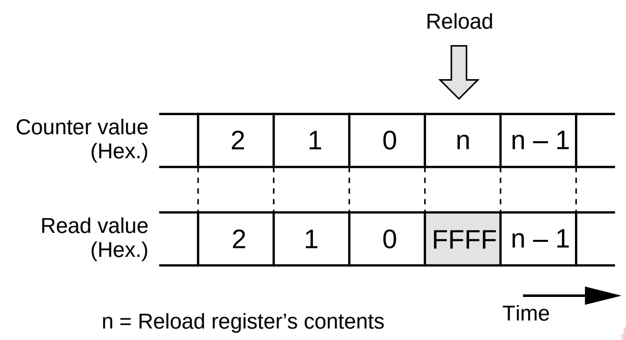


Fig. 5.3.7 Reading timer Ai register

5.4 Event counter mode

In this mode, the timer counts an external signal. (Refer to Tables 5.4.1 and 5.4.2.) Figure 5.4.1 shows the structures of the timer Ai mode register and timer Ai register in the event counter mode.

Table 5.4.1 Specifications of event counter mode (when not using two-phase pulse signal processing function)

Item	Specifications
Count source	● External signal input to the TAi_{IN} pin ● The count source's valid edge can be selected between the falling and the rising edges by software.
Count operation	● Up-count or down-count can be switched by external signal or software. ● When the counter overflows or underflows, reload register's contents are reloaded and counting continues.
Count start condition	When count start bit is set to "1."
Count stop condition	When count start bit is cleared to "0."
Interrupt request occurrence timing	When the counter overflows or underflows.
TAi_{IN} pin function	Count source input
TAi_{OUT} pin function	Programmable I/O port, pulse output, or up-count/down-count switch signal input
Read from timer Ai register	Counter value can be read out.
Write to timer Ai register	● While counting is stopped When a value is written to timer Ai register, it is written to both reload register and counter. ● While counting is in progress When a value is written to timer Ai register, it is written to only reload register. (Transferred to counter at next reload time.)

5.4 Event counter mode

Table 5.4.2 Specifications of event counter mode (when using two-phase pulse signal processing function with timers A2, A3, and A4)

Item	Specifications
Count source	External signal (two-phase pulse) input to the TAj_{IN} or TAj_{OUT} pin (j = 2 to 4)
Count operation	● Up-count or down-count can be switched by external signal (two-phase pulse). ● When the counter overflows or underflows, reload register's contents are reloaded and counting is continued.
Count start condition	When count start bit is set to "1."
Count stop condition	When count start bit is cleared to "0."
Interrupt request occurrence timing	When the counter overflows or underflows.
TAj_{IN}, TAj_{OUT} (j = 2 to 4) pin function	Two-phase pulse input
Read from timer Aj register	Counter value can be read out.
Write to timer Aj register	● While counting is stopped When a value is written to timer A2, A3, or A4 register, it is written to both reload register and counter. ● While counting is in progress When a value is written to timer A2, A3, or A4 register, it is written to only reload register. (Transferred to counter at next reload time.)

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

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 1 : Event counter mode	0	RW
1			0	RW
2	Pulse output function select bit	0 : No pulse output (TAi_{OUT} pin functions as a programmable I/O port.) 1 : Pulse output (TAi_{OUT} pin functions as a pulse output pin.)	0	RW
3	Count polarity select bit	0 : Counts at falling edge of external signal 1 : Counts at rising edge of external signal	0	RW
4	Up-down switching factor select bit	0 : Contents of up-down register 1 : Input signal to TAi_{OUT} pin	0	RW
5	Fix this bit to “0” in event counter mode.		0	RW
6	These bits are ignored in event counter mode.		0	RW
7			0	RW

X : It may be either “0” or “1.”

(b15) b7	(b8) b0	b7	b0

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

Bit	Functions	At reset	RW
15 to 0	These bits can be set to “0000_{16}” to “$FFFF_{16}$.” Assuming that the set value = n, the counter divides the count source frequency by n + 1 when down-counting, or by $FFFF_{16}$ – n + 1 when up-counting. When reading, the register indicates the counter value.	Undefined	RW

Fig. 5.4.1 Structures of timer Ai mode register and timer Ai register in event counter mode

5.4 Event counter mode

5.4.1 Setting for event counter mode

Figures 5.4.2 and 5.4.3 show an initial setting example for registers relevant to the event counter mode. Note that when using interrupts, set up to enable the interrupts. For details, refer to **"Chapter 4. INTERRUPTS."**

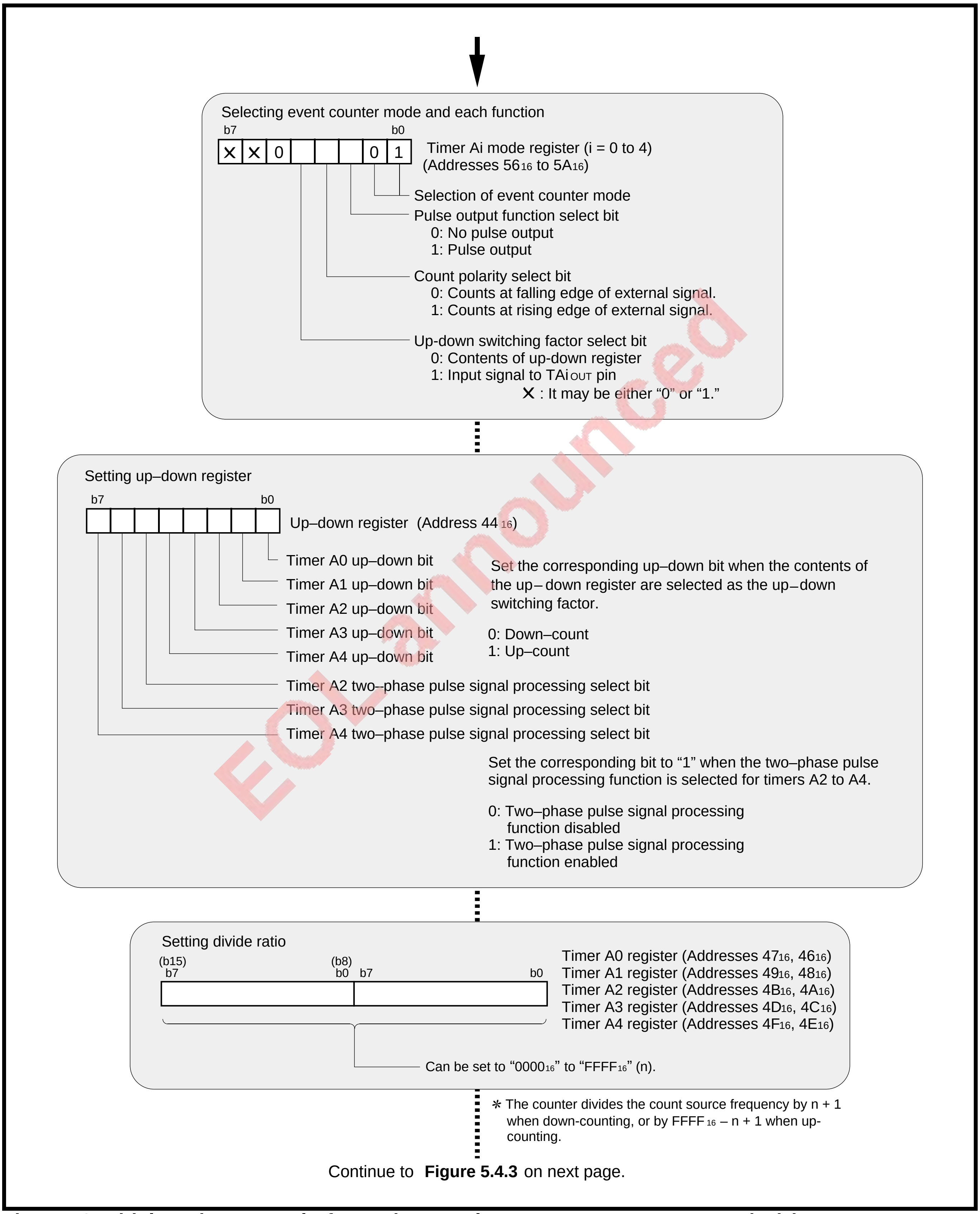


Fig. 5.4.2 Initial setting example for registers relevant to event counter mode (1)

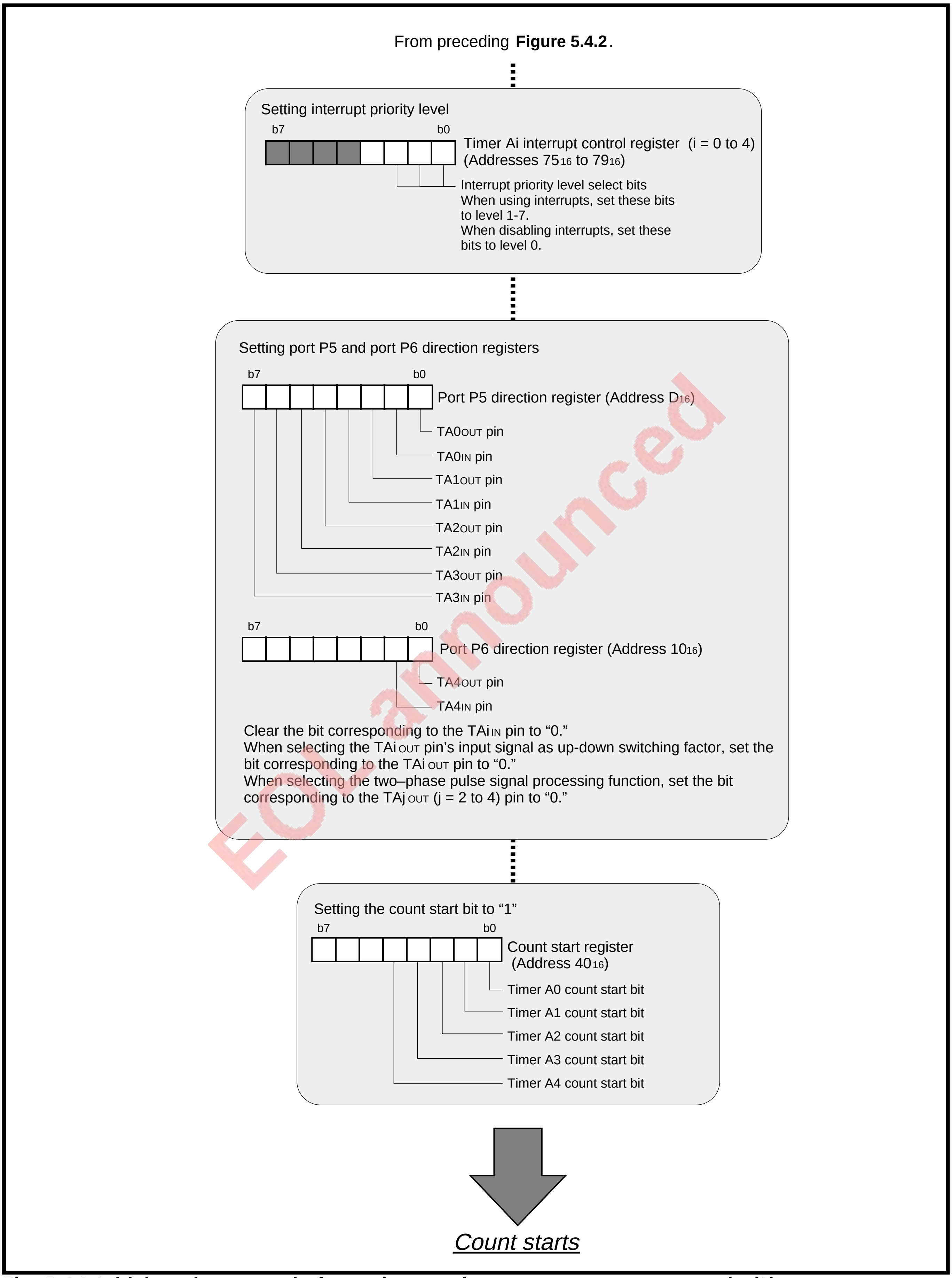
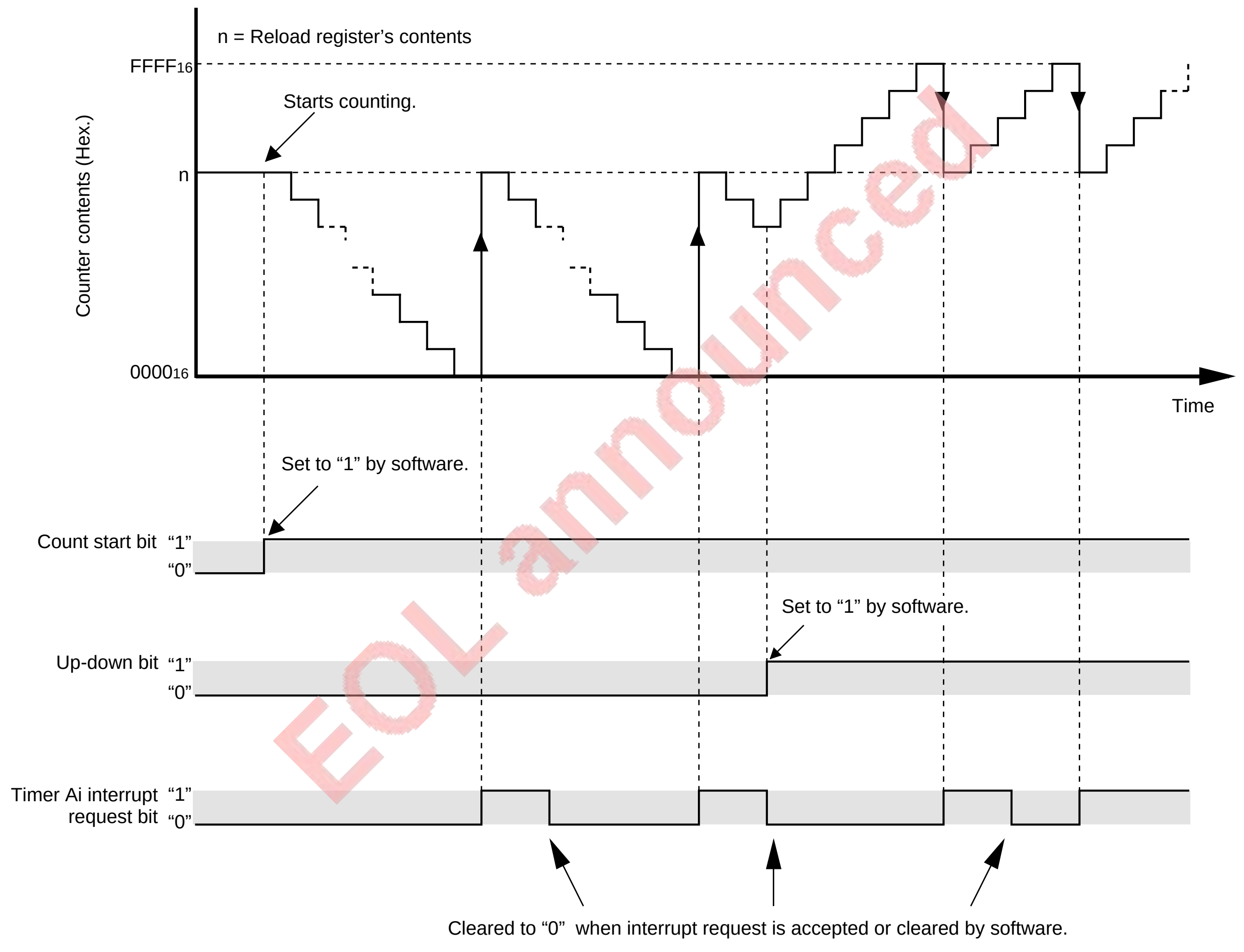


Fig. 5.4.3 Initial setting example for registers relevant to event counter mode (2)

5.4.2 Operation in event counter mode

① When the count start bit is set to "1," the counter starts counting of the count source.
② The counter counts the count source's valid edges.
③ When the counter underflows or overflows, the reload register's contents are reloaded and counting continues.
④ The timer Ai interrupt request bit is set to "1" when the counter underflows or overflows in ③. The interrupt request bit remains set to "1" until the interrupt request is accepted or the interrupt request bit is cleared to "0" by software.

Figure 5.4.4 shows an example of operation in the event counter mode.



Note: The above applies when the up-down bit's contents are selected as the up-down switching factor (i.e., up-down switching factor select bit = "0").

Fig. 5.4.4 Example of operation in event counter mode (without pulse output function and two-phase pulse signal processing function)

(1) Switching between up-count and down-count

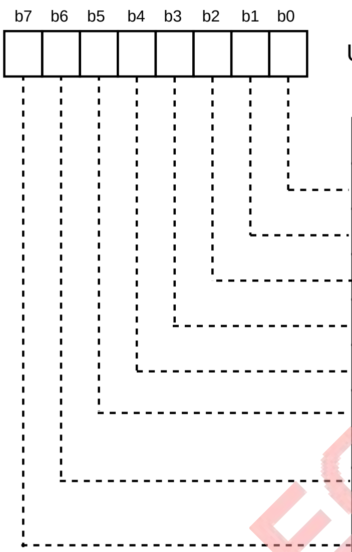
The up-down register (address 44_{16}) or the input signal from the TAi_{OUT} pin is used to switch the up-count from and to the down-count. This switching is performed by the up-down bit when the up-down switching factor select bit (bit 4 at addresses 56_{16} to $5A_{16}$) is "0," and by the input signal from the TAi_{OUT} pin when the up-down switching factor select bit is "1."
When switching the up-count/down-count, this switching is actually performed when the count source's next valid edge is input.

●Switching by up-down bit

The counter down-counts when the up-down bit is "0," and up-counts when the up-down bit is "1."
Figure 5.4.5 shows the structure of the up-down register.

●Switching by TAi_{OUT} pin's input signal

The counter down-counts when the TAi_{OUT} pin's input signal is at "L" level, and up-counts when the TAi_{OUT} pin's input signal is at "H" level.
When using the TAi_{OUT} pin input signal to switch the up-count/down-count, set the port P5 and P6 direction registers' bits which correspond to the TAi_{OUT} pin for the input mode.



Up-down register (Address 44_{16})

Bit	Bit name	Functions	At reset	RW
0	Timer A0 up-down bit	0 : Down-count 1 : Up-count This function is valid when the contents of the up-down register are selected as the up-down switching factor.	0	RW
1	Timer A1 up-down bit		0	RW
2	Timer A2 up-down bit		0	RW
3	Timer A3 up-down bit		0	RW
4	Timer A4 up-down bit		0	RW
5	Timer A2 two-phase pulse signal processing select bit **(Note)**	0 : Disabled Two-phase pulse signal processing function 1 : Enabled Two-phase pulse signal processing function When not using the two-phase pulse signal processing function, make sure to set the bit to "0." The value is "0" at reading.	0	WO
6	Timer A3 two-phase pulse signal processing select bit **(Note)**		0	WO
7	Timer A4 two-phase pulse signal processing select bit **(Note)**		0	WO

Note: Use the **LDM** or **STA** instruction when writing to bits 5 to 7.

Fig. 5.4.5 Structure of up-down register

5.4.3 Select functions

The following describes the selective pulse output, and two-phase pulse signal processing functions.

(1) Pulse output function

The pulse output function is selected by setting the pulse output function select bit (bit 2 at addresses 56_{16} to $5A_{16}$) to "1." When this function is selected, the TAi_{OUT} pin is forcibly set for the pulse output pin regardless of the corresponding bits of the port P5 and port P6 direction registers. The TAi_{OUT} pin outputs pulses of which polarity is inverted each time the counter underflows or overflows. (Refer to Figure 5.3.6.)

When the count start bit (address 40_{16}) is "0" (count stopped), the TAi_{OUT} pin outputs "L" level.

(2) Two-phase pulse signal processing function (Timers A2 to A4)

For timers A2 to A4, the two-phase pulse signal processing function is selected by setting the two-phase pulse signal processing select bits (bits 5 to 7 at address 44_{16}) to "1." (Refer to Figure 5.4.5.)

Figure 5.4.6 shows the timer A2, A3, and A4 mode registers when the two-phase pulse signal processing function is selected.

With timers selecting the two-phase pulse signal processing function, the timer counts two kinds of pulses of which phases differ by 90 degrees. There are two types of the two-phase pulse signal processing: normal processing and quadruple processing. In timers A2 and A3, normal processing is performed; in timer A4, quadruple processing is performed.

For some bits of the port P5 and P6 direction registers correspond to pins used for two-phase pulse input, set these bits for the input mode.

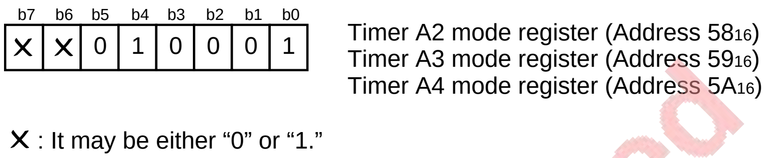


Fig. 5.4.6 Timer A2, A3, and A4 mode registers when two-phase pulse signal processing function is selected

●Normal processing

The timer up-counts the rising edges to the TAk_{IN} pin when the phase has the relationship that the TAk_{IN} pin's input signal level goes from "L" to "H" while the TAk_{OUT} (k = 2 and 3) pin's input signal is "H" level.

The timer down-counts the falling edges to the TAk_{IN} pin when the phase has the relationship that the TAk_{IN} pin's input signal level goes from "H" to "L" while the TAk_{OUT} pin's input signal is "H" level. (Refer to Figure 5.4.7.)

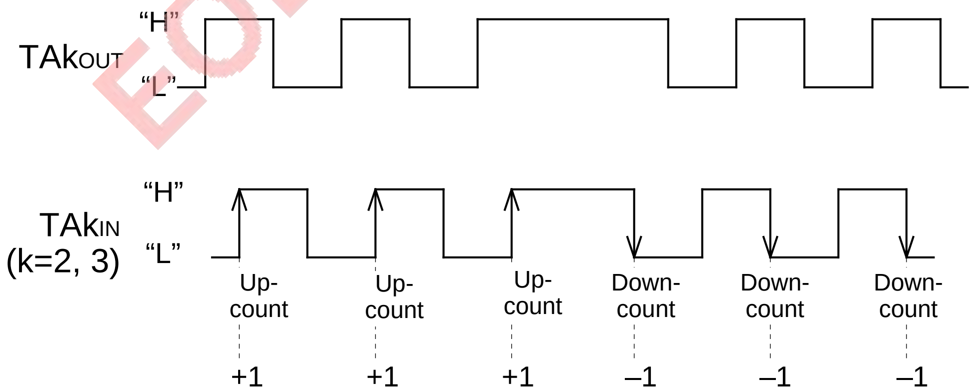


Fig. 5.4.7 Normal processing

●Quadruple processing

The timer up-counts all rising and falling edges to the TA4OUT and TA4IN pins when the phase has the relationship that the TA4IN pin's input signal level goes from "L" to "H" while the TA4OUT pin's input signal is "H" level.

The timer down-counts all rising and falling edges to the TA4OUT and TA4IN pins when the phase has the relationship that the TA4IN pin's input signal level goes from "H" to "L" while the TA4OUT pin's input signal is "H" level. (Refer to Figure 5.4.8.)

Table 5.4.3 lists the input signals to the TA4OUT and TA4IN pins when the quadruple processing is selected.

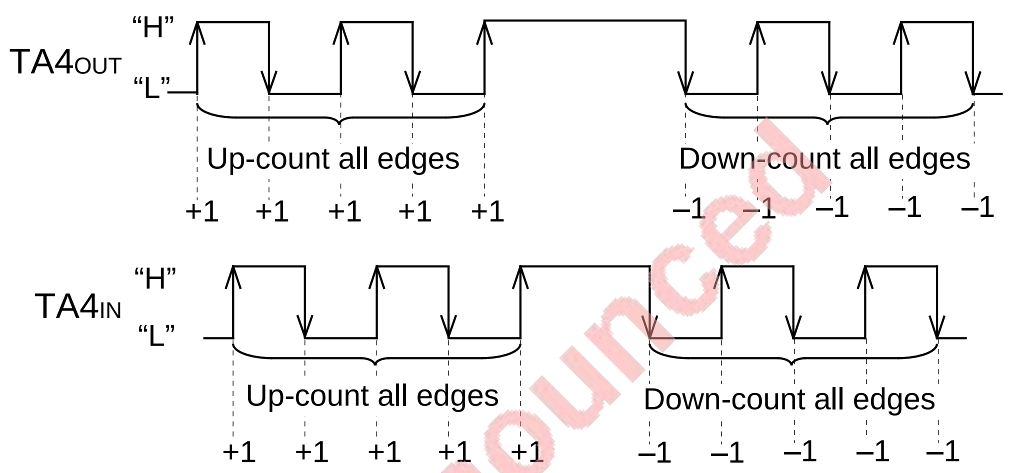


Fig. 5.4.8 Quadruple processing

Table 5.4.3 TA4OUT and TA4IN pins' input signals when quadruple operation is selected

	Input signal to TA4OUT pin	Input signal to TA4IN pin
Up-count	"H" level	Rising
	"L" level	Falling
	Rising	"L" level
	Falling	"H" level
Down-count	"H" level	Falling
	"L" level	Rising
	Rising	"H" level
	Falling	"L" level

[Precautions when operating in event counter mode]

1. By reading the timer Ai register, the counter value can be read out at any timing while counting is in progress. However, when the timer Ai register is read at the reload timing shown in Figure 5.4.9, a value "$FFFF_{16}$" (at the underflow) or "0000_{16}" (at the overflow) is read out. When reading the timer Ai register after setting a value to the register while counting is not in progress and before the counter starts counting, the set value is read out correctly.

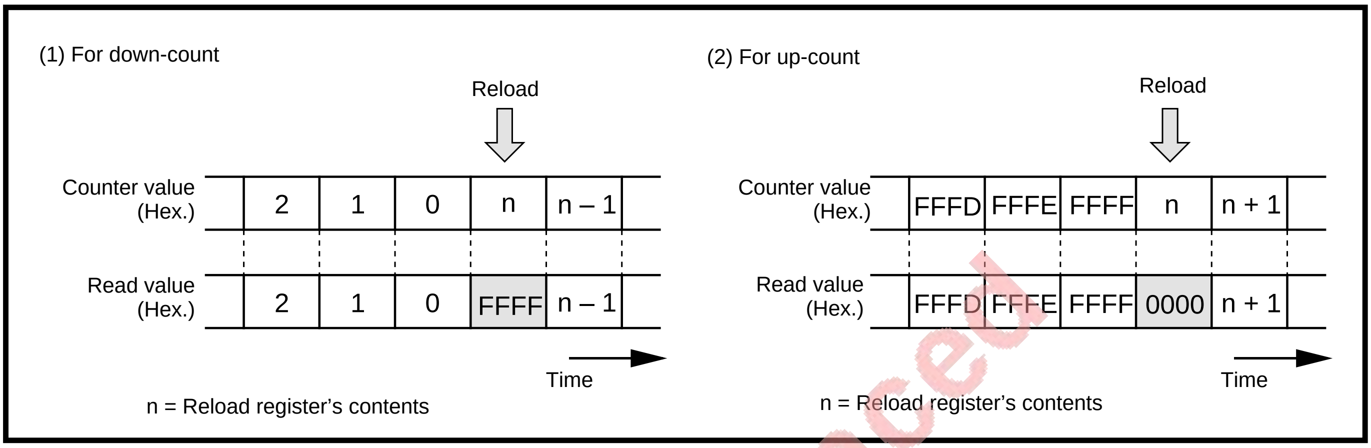


Fig. 5.4.9 Reading timer Ai register

2. The TAi_{OUT} pin is used for all functions listed below. Accordingly, only one of these functions can be selected for each timer.
   •Switching between up-count and down-count by TAi_{OUT} pin's input signal
   •Pulse output function
   •Two-phase pulse signal processing function for timers A2 to A4

5.5 One-shot pulse mode

In this mode, the timer outputs a pulse which has an arbitrary width once. (Refer to Table 5.5.1.) When a trigger occurs, the timer outputs "H" level from the TAi_{OUT} pin for an arbitrary time. Figure 5.5.1 shows the structures of the timer Ai mode register and timer Ai register in the one-shot pulse mode.

Table 5.5.1 Specifications of one-shot pulse mode

Item	Specifications
Count source	f_2/f_4, f_{16}/f_{32}, f_{64}/f_{128}, or f_{512}/f_{1024}
Count operation	● Down-count ● When the counter value becomes "0000_{16}," reload register's contents are reloaded and counting stops. ● If a trigger occurs during counting, reload register's contents are reloaded then and counting continues.
Count start condition	● When a trigger occurs. (**Note**) ● Internal or external trigger can be selected by software.
Count stop condition	● When the counter value becomes "0000_{16}" ● When count start bit is cleared to "0"
Interrupt request occurrence timing	When counting stops.
TAi_{IN} pin function	Programmable I/O port or trigger input
TAi_{OUT} pin function	One-shot pulse output
Read from timer Ai register	An undefined value is read out.
Write to timer Ai register	● While counting is stopped When a value is written to timer Ai register, it is written to both reload register and counter. ● While counting is in progress When a value is written to timer Ai register, it is written to only reload register. (Transferred to counter at next reload time.)

Note: The trigger is generated with the count start bit = "1."

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

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 1 0 : One-shot pulse mode	0	RW
1			0	RW
2	Fix this bit to "1" in one-shot pulse mode.		0	RW
3	Trigger select bits	b4 b3 0 0 : } Writing "1" to one-shot start bit 0 1 : } (TAiIN pin functions as a programmable I/O port.) 1 0 : Falling edge of TAiIN pin's input signal 1 1 : Rising edge of TAiIN pin's input signal	0	RW
4			0	RW
5	Fix this bit to "0" in one-shot pulse mode.		0	RW
6	Count source select bits	b7 b6 0 0 : f_2/f_4 0 1 : f_{16}/f_{32} 1 0 : f_{64}/f_{128} 1 1 : f_{512}/f_{1024}	0	RW
7			0	RW

(b15) b7 (b8) b0 b7 b0

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

Bit	Functions	At reset	RW
15 to 0	These bits can be set to "0000_{16}" to "$FFFF_{16}$." Assuming that the set value = n, the "H" level width of the one-shot pulse output from the TAiOUT pin is expressed as follows : n / f_i.	Undefined	WO

f_i: Frequency of count source (f_2/f_4, f_{16}/f_{32}, f_{64}/f_{128}, or f_{512}/f_{024})

Fig. 5.5.1 Structures of timer Ai mode register and timer Ai register in one-shot pulse mode

5.5 One-shot pulse mode

5.5.1 Setting for one-shot pulse mode

Figures 5.5.2 and 5.5.3 show an initial setting example for registers relevant to the one-shot pulse mode. Note that when using interrupts, set up to enable the interrupts. For details, refer to **"Chapter 4. INTERRUPTS."**

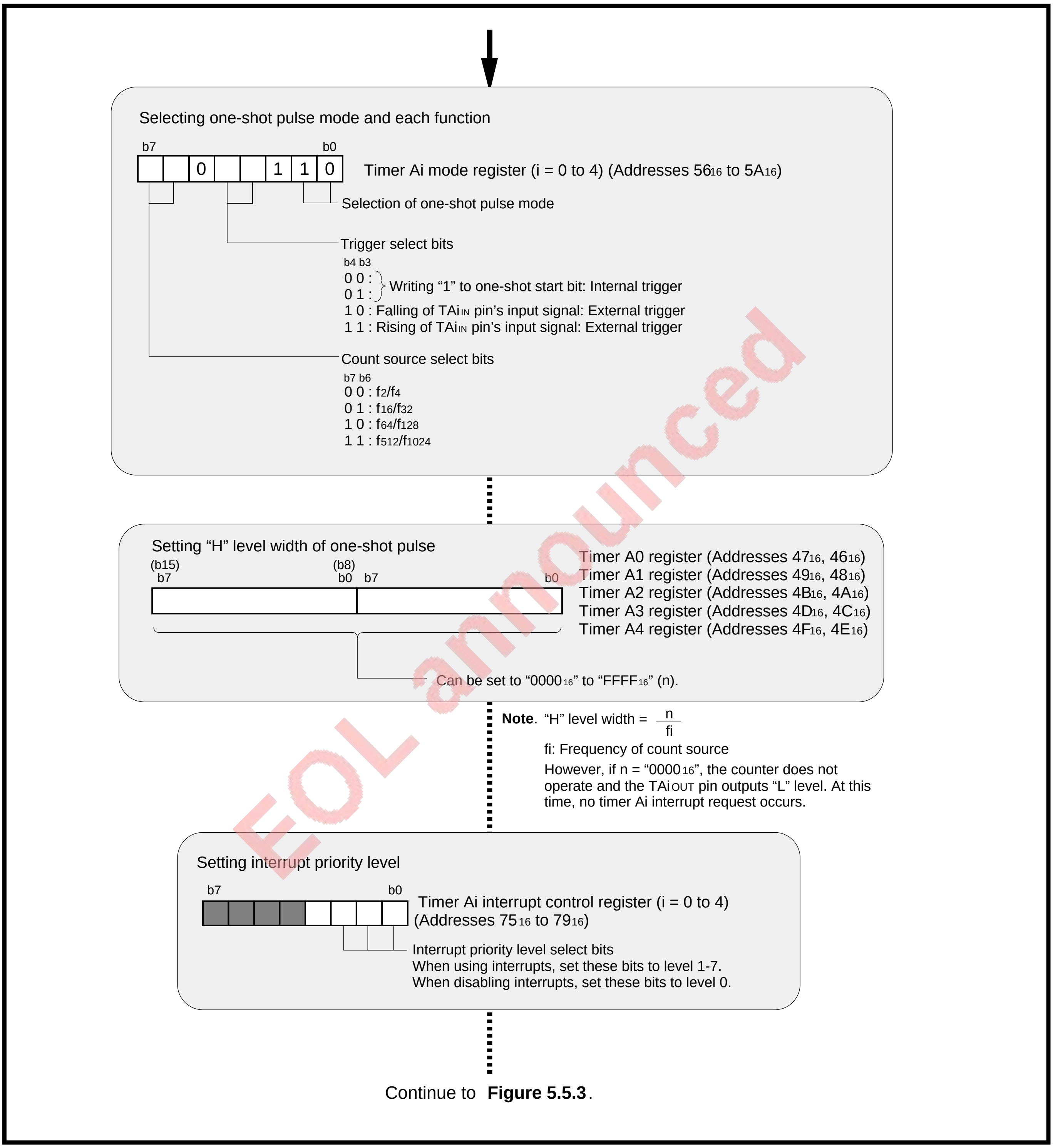


Fig. 5.5.2 Initial setting example for registers relevant to one-shot pulse mode (1)

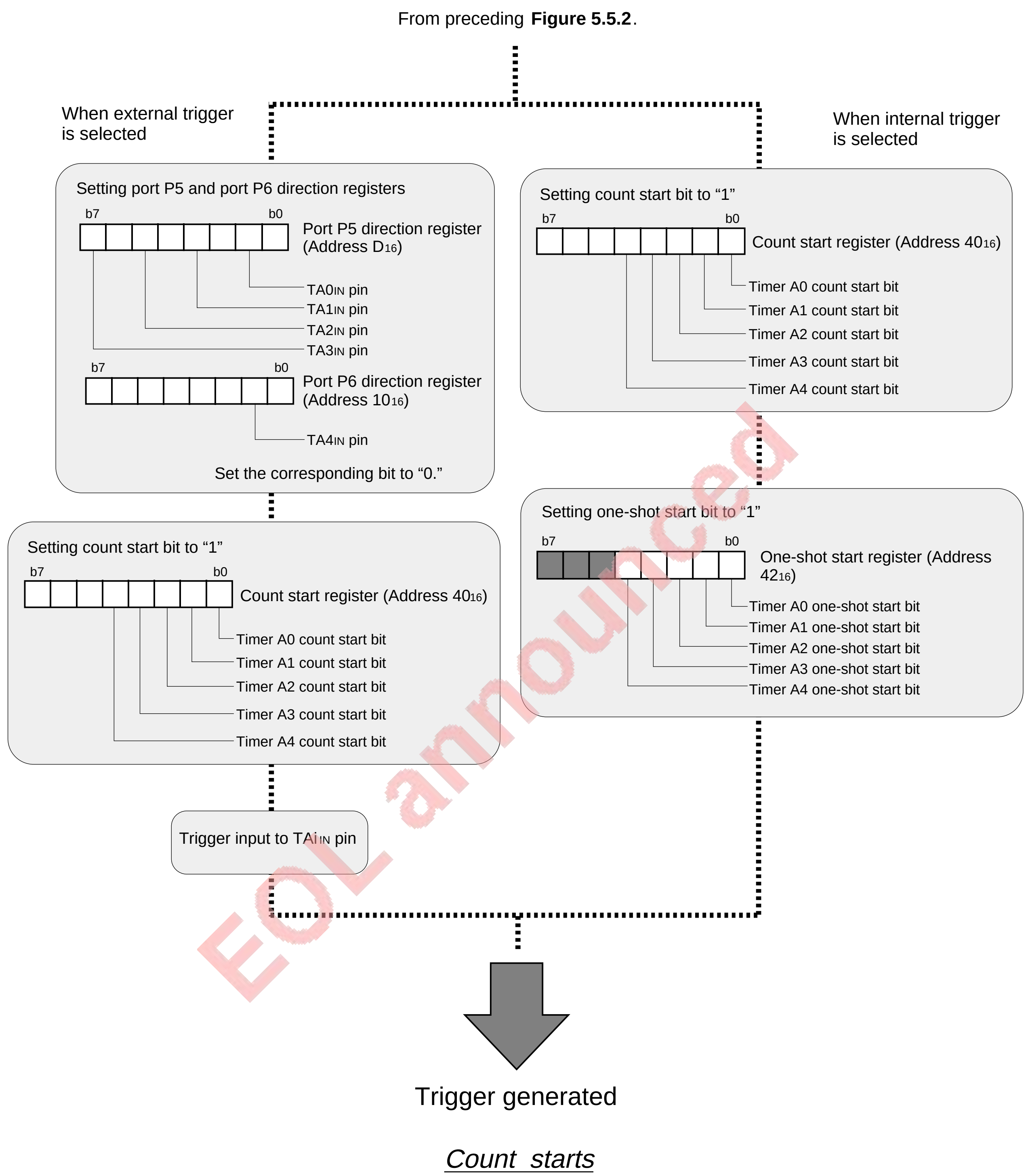


Fig. 5.5.3 Initial setting example for registers relevant to one-shot pulse mode (2)

5.5 One-shot pulse mode

5.5.2 Count source

In the one-shot pulse mode, the count source select bits (bits 6 and 7 at addresses 56_{16} to $5A_{16}$) select the count source. Table 5.5.2 lists the count source frequency.

Table 5.5.2 Count source frequency

Count source select bits		f(XIN) = 25 MHz				f(XIN) = 40 MHz	
		Clock source for peripheral devices select bit = "0"		Clock source for peripheral devices select bit = "1"		Clock source for peripheral devices select bit = "0"	
b7	b6	Count source	Frequency	Count source	Frequency	Count source	Frequency
0	0	f_4	6.25 MHz	f_2	12.5 MHz	f_4	10 MHz
0	1	f_{32}	781.25 kHz	f_{16}	1.5625 MHz	f_{32}	1.25 MHz
1	0	f_{128}	195.3125 kHz	f_{64}	390.625 kHz	f_{128}	312.5 kHz
1	1	f_{1024}	24.4141 kHz	f_{512}	48.8281 kHz	f_{1024}	39.0625 kHz

Clock source for peripheral devices select bit : bit 2 at address $5F_{16}$

5.5.3 Trigger

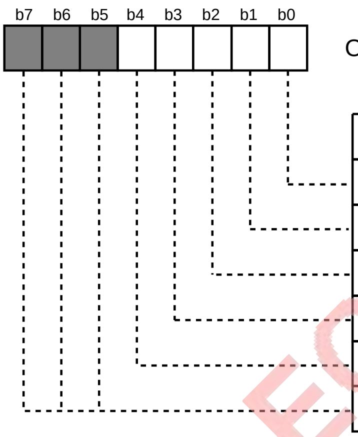
The counter is enabled for counting when the count start bit (address 40_{16}) is set to "1." The counter starts counting when a trigger is generated after it has been enabled. An internal or an external trigger can be selected as that trigger.
An internal trigger is selected when the trigger select bits (bits 4 and 3 at addresses 56_{16} to $5A_{16}$) are "00_2" or "01_2"; an external trigger is selected when the bits are "10_2" or "11_2."
If a trigger is generated during counting, the reload register's contents are reloaded and the counter continues counting. If generating a trigger during counting, make sure that a certain time which is equivalent to one cycle of the timer's count source or more has passed between the previous generated trigger and a new generated trigger.

(1) When selecting internal trigger

A trigger is generated when writing "1" to the one-shot start bit (address 42_{16}). Figure 5.5.4 shows the structure of the one-shot start register.

(2) When selecting external trigger

A trigger is generated at the falling of the TAi_{IN} pin's input signal when bit 3 at addresses 56_{16} to $5A_{16}$ is "0," or at its rising when bit 3 is "1."
When using an external trigger, set the port P5 and P6 direction registers' bits which correspond to the TAi_{IN} pins for the input mode.



One-shot start register (Address 42_{16})

Bit	Bit name	Functions	At reset	RW
0	Timer A0 one-shot start bit	1 : Start outputting one-shot pulse (valid when selecting internal trigger.) The value is "0" at reading.	0	WO
1	Timer A1 one-shot start bit		0	WO
2	Timer A2 one-shot start bit		0	WO
3	Timer A3 one-shot start bit		0	WO
4	Timer A4 one-shot start bit		0	WO
7 to 5	Nothing is assigned.		Undefined	–

Fig. 5.5.4 Structure of one-shot start register

5.5.4 Operation in one-shot pulse mode

① When the one-shot pulse mode is selected with the operating mode select bits, the TAi_{OUT} pin outputs "L" level.

② When the count start bit is set to "1," the counter is enabled for counting. After that, counting starts when a trigger is generated.

③ When the counter starts counting, the TAi_{OUT} pin outputs "H" level. (However, if the timer Ai register has a value "0000_{16}" set in it, the counter does not operate and the output from the TAi_{OUT} pin remains "L." The timer Ai interrupt request does not occur.)

④ When the counter value becomes "0000_{16}," the output from the TAi_{OUT} pin becomes "L" level. Additionally, the reload register's contents are reloaded and the counter stops counting there.

⑤ Simultaneously at ④, the timer Ai interrupt request bit is set to "1."
This interrupt request bit remains set to "1" until the interrupt request is accepted or the interrupt request bit is cleared to "0" by software.

Figure 5.5.5 shows an example of operation in the one-shot pulse mode.
When a trigger is generated after ④ above, the counter and TAi_{OUT} pin perform the same operations beginning from ② again. Furthermore, if a trigger is generated during counting, the counter down-counts once after this generated new trigger, and it continues counting with the reload register's contents reloaded. If generating a trigger during counting, make sure that a certain time which is equivalent to one cycle of the timer's count source or more has passed between the previous generated trigger and a new generated trigger.

The one-shot pulse output from the TAi_{OUT} pin can be disabled by clearing the timer Ai mode register's bit 2 to "0." Accordingly, timer Ai can be also used as an internal one-shot timer that does not perform the pulse output. In this case, the TAi_{OUT} pin functions as a programmable I/O port.

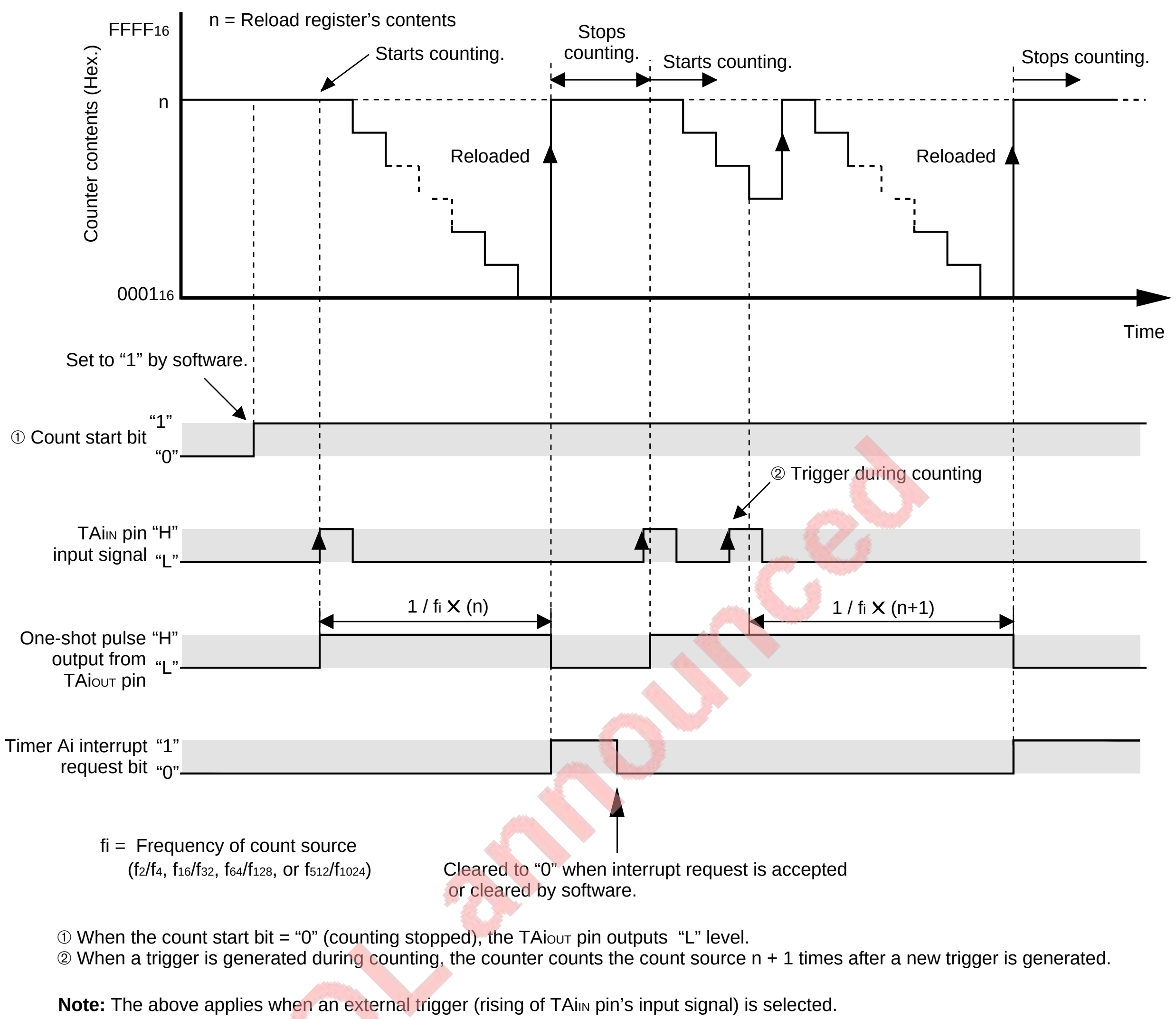


Fig. 5.5.5 Example of operation in one-shot pulse mode (selecting external trigger)

[Precautions when operating in one-shot pulse mode]

1. If the count start bit is cleared to "0" during counting, the counter stops counting and the TAi_{OUT} pin's output level becomes "L." At the same time, the timer Ai interrupt request bit is set to "1."

2. A one-shot pulse is output synchronously with an internally generated count source. Accordingly, when selecting an external trigger, there will be a delay equivalent to one cycle of count source at maximum from when a trigger is input to the TAi_{IN} pin till when a one-shot pulse is output.

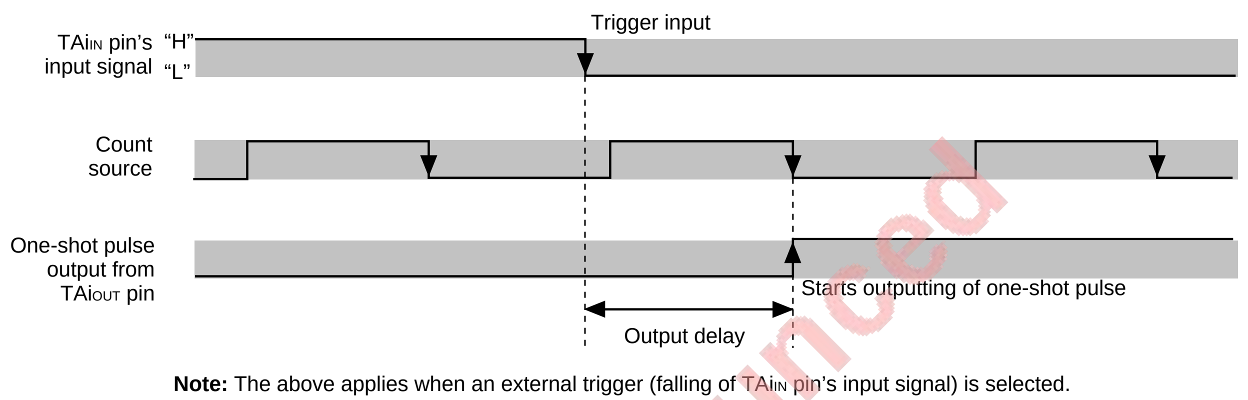


Fig. 5.5.6 Output delay in one-shot pulse output

3. When setting the timer's operating mode in one of the followings, the timer Ai interrupt request bit is set to "1."
   - ●When the one-shot pulse mode is selected after a reset
   - ●When the operating mode is switched from the timer mode to the one-shot pulse mode
   - ●When the operating mode is switched from the event counter mode to the one-shot pulse mode

   Therefore, when using the timer Ai interrupt (interrupt request bit), be sure to clear the timer Ai interrupt request bit to "0" after above setting.

5.6 Pulse width modulation (PWM) mode

In this mode, the timer continuously outputs pulses which have an arbitrary width. (Refer to Table 5.6.1.) Figure 5.6.1 shows the structures of the timer Ai mode register and timer Ai register in the PWM mode.

Table 5.6.1 Specifications of PWM mode

Item	Specifications
Count source	f_2/f_4, f_{16}/f_{32}, f_{64}/f_{128}, or f_{512}/f_{1024}
Count operation	● Down-count (operating as an 8-bit or 16-bit pulse width modulator) ● Reload register's contents are reloaded at rising of PWM pulse and counting continues. ● A trigger generated during counting does not affect the counting.
Count start condition	● When a trigger is generated. ● Internal or external trigger can be selected by software.
Count stop condition	When count start bit is cleared to "0."
Interrupt request occurrence timing	At falling of PWM pulse
TAi_{IN} pin function	Programmable I/O port or trigger input
TAi_{OUT} pin function	PWM pulse output
Read from timer Ai register	An undefined value is read out.
Write to timer Ai register	● While counting is stopped When a value is written to timer Ai register, it is written to both reload register and counter. ● While counting is in progress When a value is written to timer Ai register, it is written to only reload register. (Transferred to counter at next reload time.)

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

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 1 1 : PWM mode	0	RW
1			0	RW
2	Fix this bit to “1” in PWM mode.		0	RW
3	Trigger select bits	b4 b3 0 0 : } Writing “1” to count start bit 0 1 : } (TAi$_{IN}$ pin functions as a programmable I/O port.) 1 0 : Falling edge of TAi$_{IN}$ pin’s input signal 1 1 : Rising edge of TAi$_{IN}$ pin’s input signal	0	RW
4			0	RW
5	16/8-bit PWM mode select bit	0 : As a 16-bit pulse width modulator 1 : As an 8-bit pulse width modulator	0	RW
6	Count source select bits	b7 b6 0 0 : f_2/f_4 0 1 : f_{16}/f_{32} 1 0 : f_{64}/f_{128} 1 1 : f_{512}/f_{1024}	0	RW
7			0	RW

<When operating as a 16-bit pulse width modulator>

(b15) b7 ... (b8) b0 | b7 ... b0

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

Bit	Functions	At reset	RW
15 to 0	These bits can be set to “0000_{16}” to “$FFFE_{16}$.” Assuming that the set value = n, the “H” level width of the PWM pulse output from the TAi$_{OUT}$ pin is expressed as follows: $\frac{n}{f_i}$	Undefined	WO

f_i: Frequency of count source (f_2/f_4, f_{16}/f_{32}, f_{64}/f_{128}, or f_{512}/f_{1024})

<When operating as an 8-bit pulse width modulator>

(b15) b7 ... (b8) b0 | b7 ... b0

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

Bit	Functions	At reset	RW
7 to 0	These bits can be set to “00_{16}” to “FF_{16}.” Assuming that the set value = m, PWM pulse’s period output from the TAi$_{OUT}$ pin is expressed as follows: $\frac{(m + 1)(2^8 - 1)}{f_i}$	Undefined	WO
15 to 8	These bits can be set to “00_{16}” to “FE_{16}.” Assuming that the set value = n, the “H” level width of the PWM pulse output from the TAi$_{OUT}$ pin is expressed as follows: $\frac{n(m + 1)}{f_i}$	Undefined	WO

f_i: Frequency of count source (f_2/f_4, f_{16}/f_{32}, f_{64}/f_{128}, or f_{512}/f_{1024})

Fig. 5.6.1 Structures of timer Ai mode registers and timer Ai registers in PWM mode

5.6.1 Setting for PWM mode

Figures 5.6.2 and 5.6.3 show an initial setting example for registers relevant to the PWM mode. Note that when using interrupts, set up to enable the interrupts. For details, refer to **"Chapter 4. INTERRUPTS."**

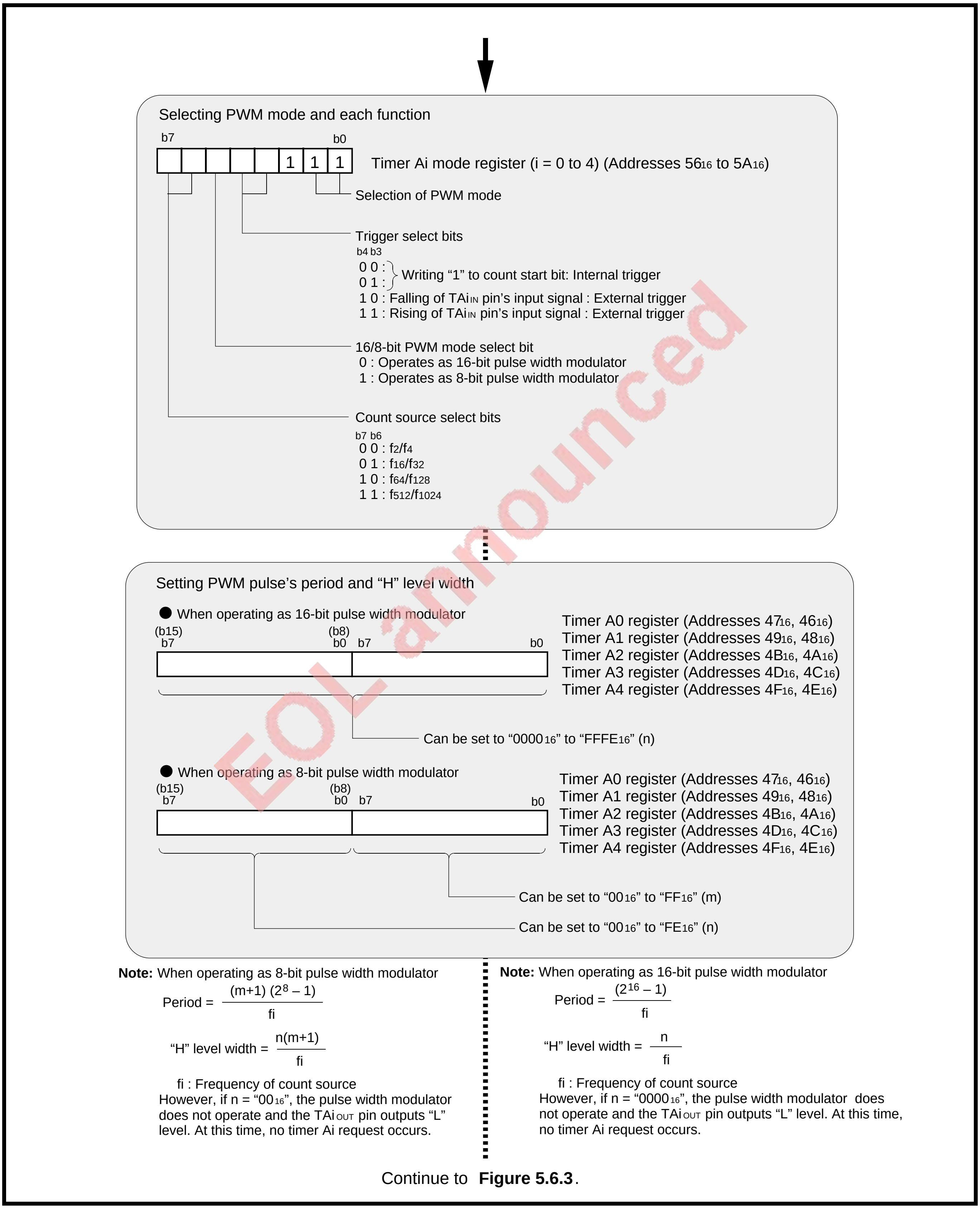


Fig. 5.6.2 Initial setting example for registers relevant to PWM mode (1)

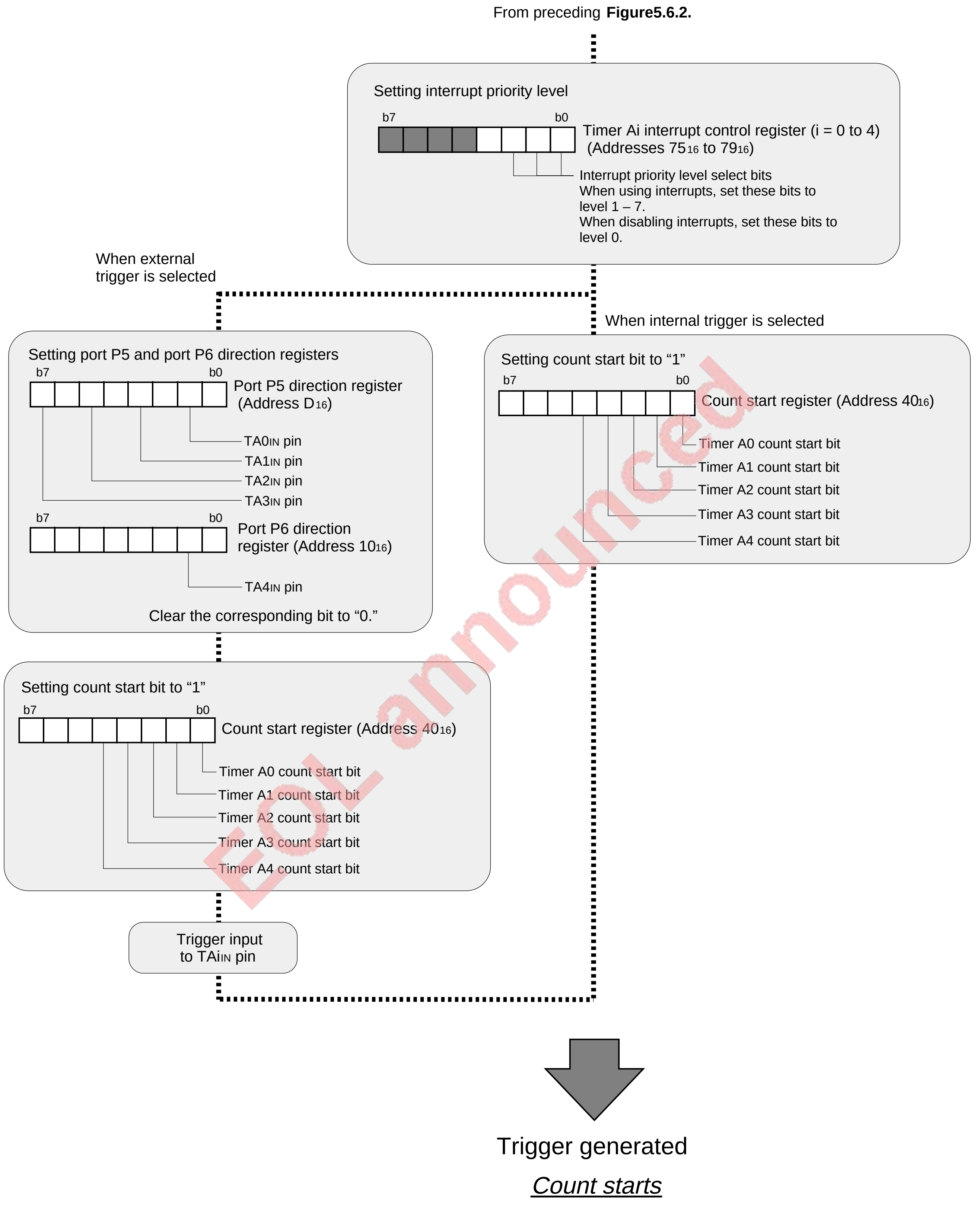


Fig. 5.6.3 Initial setting example for registers relevant to PWM mode (2)

5.6.2 Count source

In the PWM mode, the count source select bits (bits 6 and 7 at addresses 56_{16} to $5A_{16}$) select the count source. Table 5.6.2 lists the count source frequency.

Table 5.6.2 Count source frequency

Count source select bits		f(XIN) = 25 MHz				f(XIN) = 40 MHz	
		Clock source for peripheral devices select bit = "0"		Clock source for peripheral devices select bit = "1"		Clock source for peripheral devices select bit = "0"	
b7	b6	Count source	Frequency	Count source	Frequency	Count source	Frequency
0	0	f_4	6.25 MHz	f_2	12.5 MHz	f_4	10 MHz
0	1	f_{32}	781.25 kHz	f_{16}	1.5625 MHz	f_{32}	1.25 MHz
1	0	f_{128}	195.3125 kHz	f_{64}	390.625 kHz	f_{128}	312.5 kHz
1	1	f_{1024}	24.4141 kHz	f_{512}	48.8281 kHz	f_{1024}	39.0625 kHz

Clock source for peripheral devices select bit : bit 2 at address $5F_{16}$

5.6.3 Trigger

When a trigger is generated, the TAi_{OUT} pin starts outputting PWM pulses. An internal or an external trigger can be selected as that trigger.
An internal trigger is selected when the trigger select bits (bits 4 and 3 at addresses 56_{16} to $5A_{16}$) are "00_2" or "01_2"; an external trigger is selected when the bits are "10_2" or "11_2."
A trigger generated during outputting of PWM pulses is ignored and it does not affect the pulse output operation.

(1) When selecting internal trigger

A trigger is generated when writing "1" to the count start bit (at address 40_{16}).

(2) When selecting external trigger

A trigger is generated at the falling of the TAi_{IN} pin's input signal when bit 3 at addresses 56_{16} to $5A_{16}$ is "0," or at its rising when bit 3 is "1." However, the trigger input is accepted only when the count start bit is "1."
When using an external trigger, set the port P5 and P6 direction registers' bits which correspond to the TAi_{IN} pins for the input mode.

5.6 Pulse width modulation (PWM) mode

5.6.4 Operation in PWM mode

① When the PWM mode is selected with the operating mode select bits, the TAi_{OUT} pin outputs "L" level.

② When a trigger is generated, the counter (pulse width modulator) starts counting and the TAi_{OUT} pin outputs a PWM pulse (**Notes 1 and 2**).

③ The timer Ai interrupt request bit is set to "1" each time the PWM pulse level goes from "H" to "L." The interrupt request bit remains set to "1" until the interrupt request is accepted or the interrupt request bit is cleared to "0" by software.

④ Each time a PWM pulse has been output for one period, the reload register's contents are reloaded and the counter continues counting.

The following explains operation of the pulse width modulator.

[16-bit pulse width modulator]

When the 16/8-bit PWM mode select bit is set to "0," the counter operates as a 16-bit pulse width modulator. Figures 5.6.4 and 5.6.5 show operation examples of the 16-bit pulse width modulator.

[8-bit pulse width modulator]

When the 16/8-bit PWM mode select bit is set to "1," the counter is divided into 8-bit halves. Then, the high-order 8 bits operate as an 8-bit pulse width modulator, and the low-order 8 bits operate as an 8-bit prescaler. Figures 5.6.6 and 5.6.7 show operation examples of the 8-bit pulse width modulator.

Notes 1: If a value "0000_{16}" is set into the timer Ai register when the counter operates as a 16-bit pulse width modulator, the pulse width modulator does not operate and the output from the TAi_{OUT} pin remains "L" level. The timer Ai interrupt request does not occur. Similarly, if a value "00_{16}" is set into the high-order 8 bits of the timer Ai register when the counter operates as an 8-bit pulse width modulator, the same is performed.

2: When the counter operates as an 8-bit pulse width modulator, the TAi_{OUT} pin outputs "L" level of the PWM pulse which has the same width as set "H" level of the PWM pulse after a trigger generated. After that, the PWM pulse output starts from the TAi_{OUT} pin.

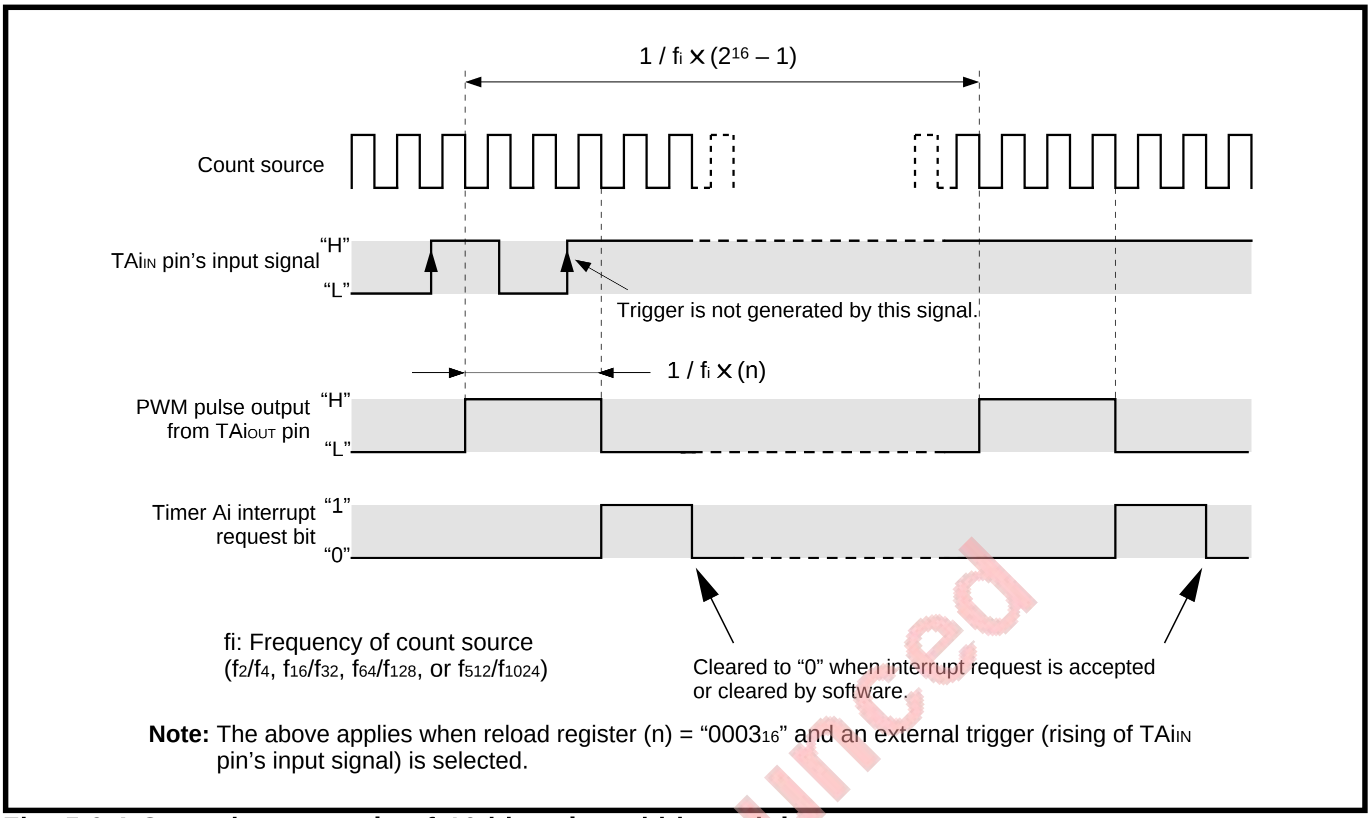


Fig. 5.6.4 Operation example of 16-bit pulse width modulator

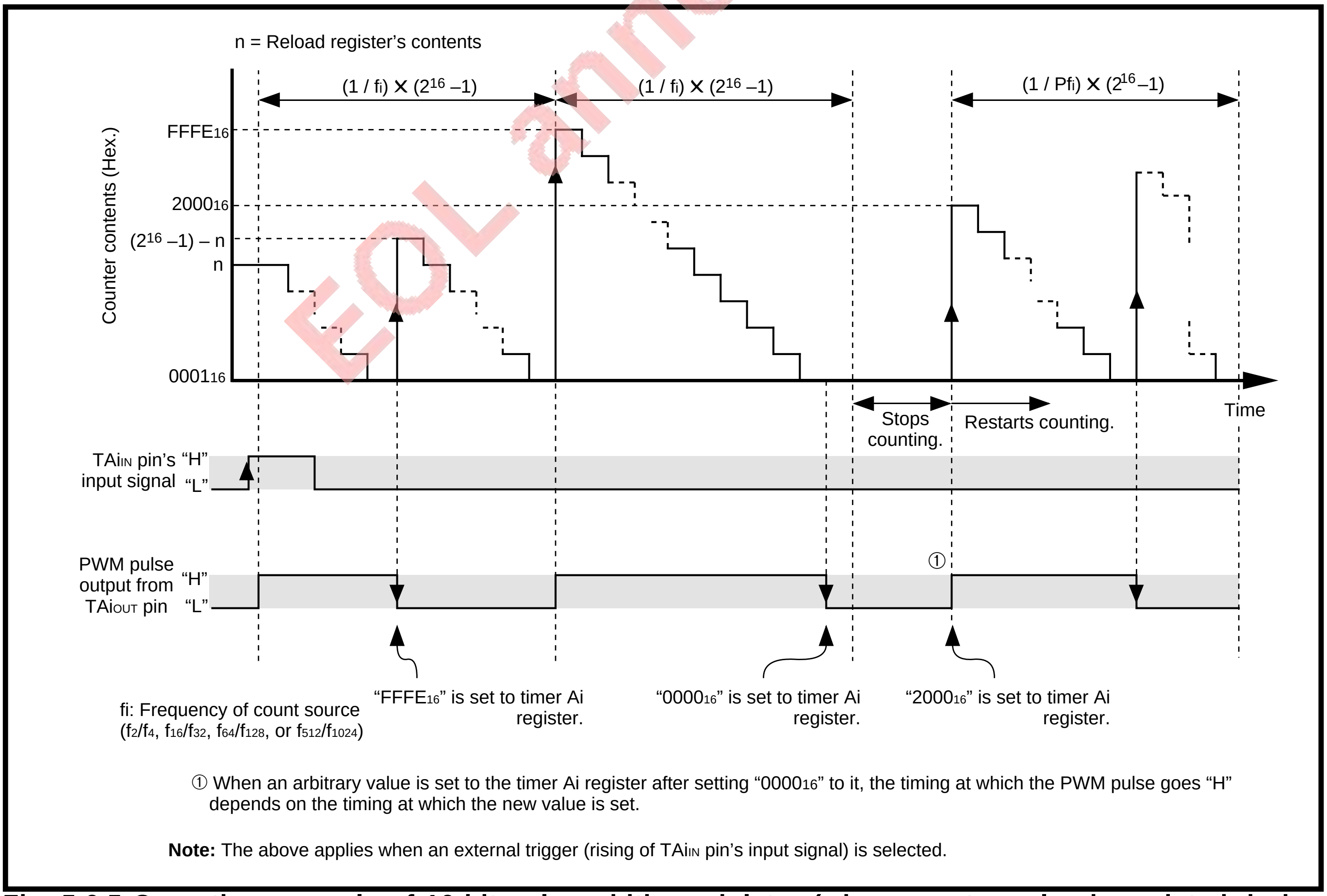
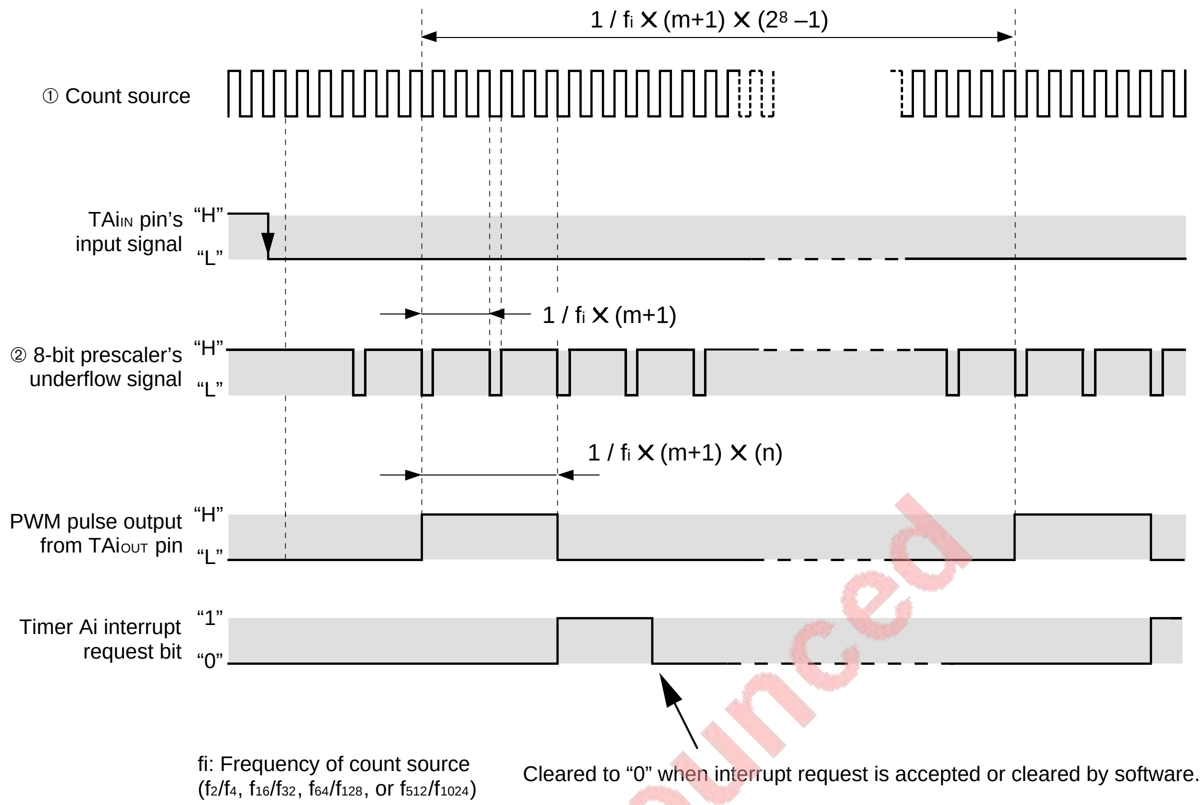


Fig. 5.6.5 Operation example of 16-bit pulse width modulator (when counter value is updated during pulse output)

5.6 Pulse width modulation (PWM) mode



① The 8-bit prescaler counts the count source.
② The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.

Note: The above applies when the reload register's high-order 8 bits (n) = "02_{16}" and low-order 8 bits (m) = "02_{16}" and an external trigger (falling of TAi_{IN} pin input signal) is selected.

Fig. 5.6.6 Operation example of 8-bit pulse width modulator

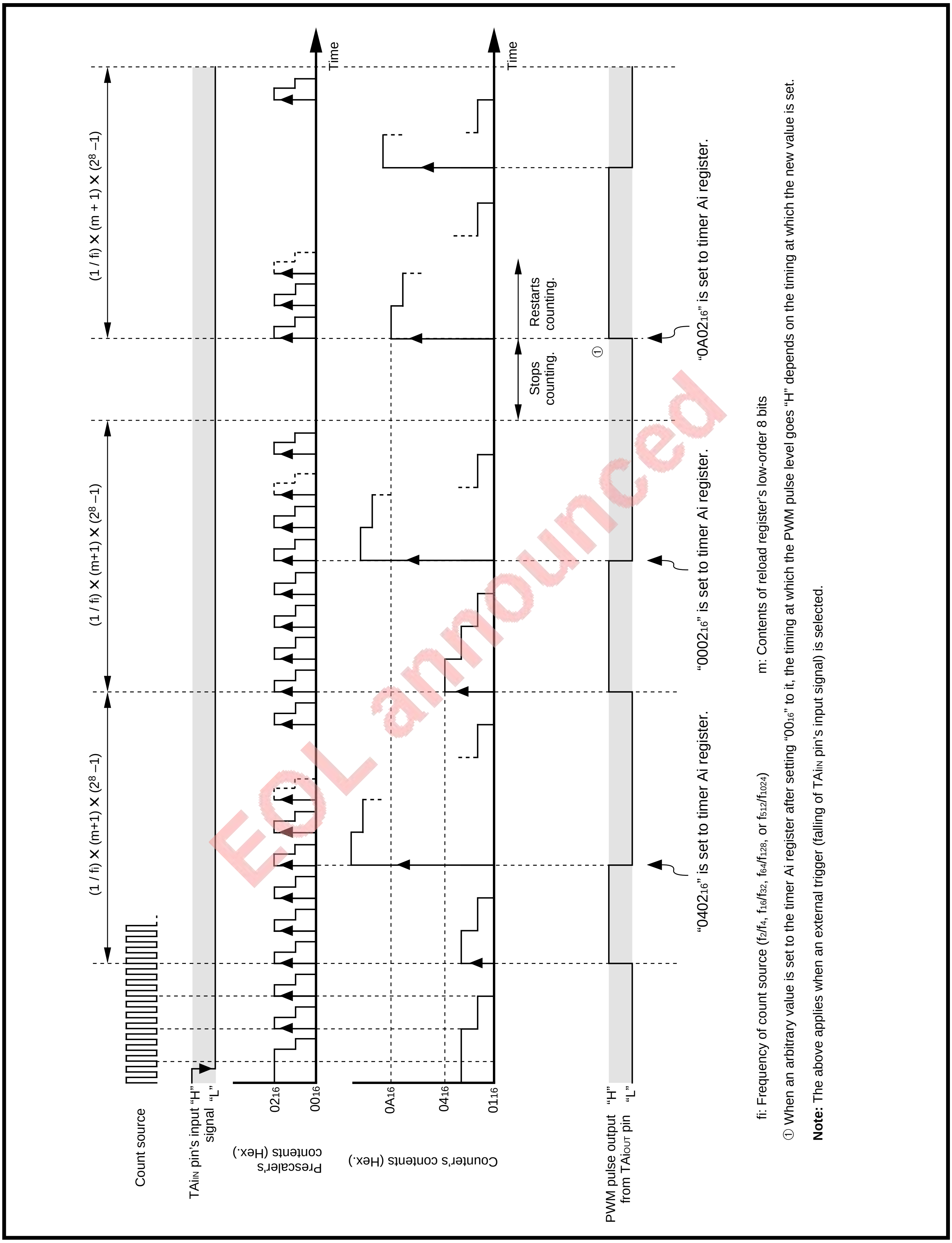


Fig. 5.6.7 Operation example of 8-bit pulse width modulator (when counter value is updated during pulse output)

5.6 Pulse width modulation (PWM) mode

[Precautions when operating in PWM mode]

1. If the count start bit is cleared to “0” while outputting PWM pulses, the counter stops counting. When the TAi_{OUT} pin was outputting “H” level at that time, the output level becomes “L” and the timer Ai interrupt request bit is set to “1.” When the TAi_{OUT} pin was outputting “L” level, the output level does not change and the timer Ai interrupt request does not occur.

2. When setting the timer’s operating mode in one of the followings, the timer Ai interrupt request bit is set to “1.”
   ●When the PWM mode is selected after a reset
   ●When the operating mode is switched from the timer mode to PWM mode
   ●When the operating mode is switched from the event counter mode to the PWM mode

   Therefore, when using the timer Ai interrupt (interrupt request bit), be sure to clear the timer Ai interrupt request bit to “0” after the above setting.

CHAPTER 6

TIMER B

Timer B consists of three counters (Timers B0 to B2) each equipped with a 16-bit reload function. Timers B0 to B2 have identical functions and operate independently with each other.

6.1 Overview

Timer Bi (i = 0 to 2) has three operating modes listed below.

- **Timer mode**
  The timer counts an internally generated count source.

- **Event counter mode**
  The timer counts an external signal.

- **Pulse period/pulse width measurement mode**
  The timer measures an external signal's pulse period or pulse width.

6.2 Block description

Figure 6.2.1 shows the block diagram of Timer B. Explanation of registers relevant to Timer B is described below.

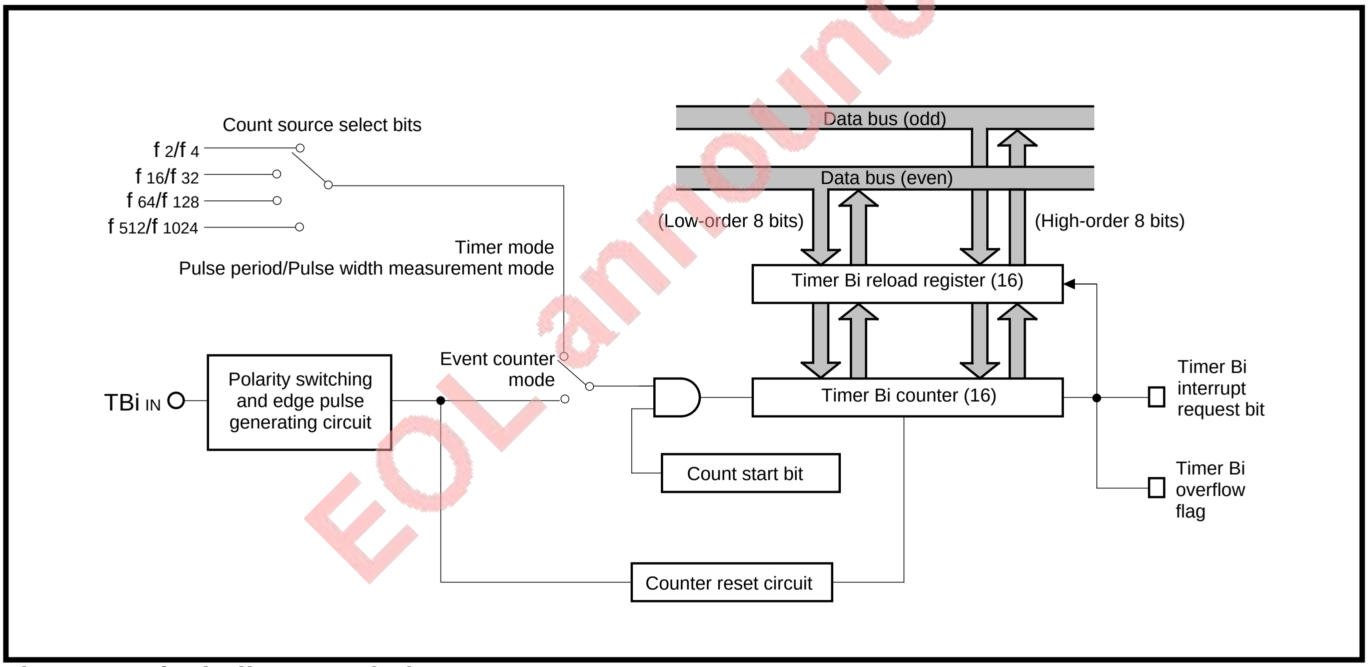


Fig. 6.2.1 Block diagram of Timer B

6.2.1 Counter and reload register (timer Bi register)

Each of timer Bi counter and reload register consists of 16 bits and has the following functions.

(1) Functions in timer mode and event counter mode

The counter down-counts each time count source is input. The reload register is used to store the initial value of the counter. When the counter underflows, the reload register's contents are reloaded into the counter.

Values are set to the counter and reload register by writing a value to the timer Bi register. Table 6.2.1 lists the memory assignment of the timer Bi register.

The value written into the timer Bi register when the counting is not in progress is set to the counter and reload register. The value written into the timer Bi register when the counting is in progress is set to only the reload register. In this case, the reload register's updated contents are transferred to the counter when the counter underflows next time. The counter value is read out by reading out the timer Bi register.

Note: When reading and writing from/to the timer Bi register, perform them in a unit of 16 bits. For more information about the value got by reading the timer Bi register, refer to **"[Precautions when operating in timer mode]"** and **"[Precautions when operating in event counter mode]."**

(2) Functions in pulse period/pulse width measurement mode

The counter up-counts each time count source is input. The reload register is used to retain the pulse period or pulse width measurement result. When a valid edge is input to the TBi_{IN} pin, the counter value is transferred to the reload register. In this mode, the value got by reading the timer Bi register is the reload register's contents, so that the measurement result is obtained.

Note: When reading from the timer Bi register, perform it in a unit of 16 bits.

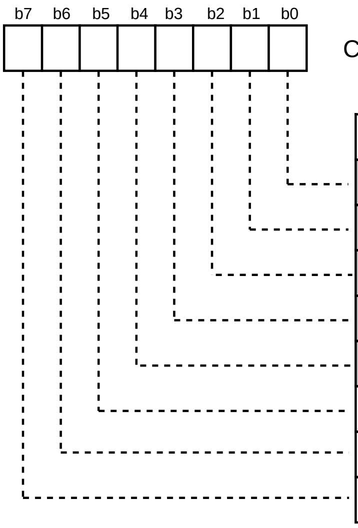
Table 6.2.1 Memory assignment of timer Bi registers

Timer Bi register	High-order byte	Low-order byte
Timer B0 register	Address 51_{16}	Address 50_{16}
Timer B1 register	Address 53_{16}	Address 52_{16}
Timer B2 register	Address 55_{16}	Address 54_{16}

Note: When reset, the contents of the timer Bi register are undefined.

6.2.2 Count start register

This register is used to start and stop counting. Each bit of this register corresponds each timer. Figure 6.2.2 shows the structure of the count start register.



Count start register (Address 40_{16})

Bit	Bit name	Functions	At reset	RW
0	Timer A0 count start bit	0 : Stop counting 1 : Start counting	0	RW
1	Timer A1 count start bit		0	RW
2	Timer A2 count start bit		0	RW
3	Timer A3 count start bit		0	RW
4	Timer A4 count start bit		0	RW
5	Timer B0 count start bit		0	RW
6	Timer B1 count start bit		0	RW
7	Timer B2 count start bit		0	RW

(shaded) : Bits 0 to 4 are not used for Timer B.

Fig. 6.2.2 Structure of count start register

6.2.3 Timer Bi mode register

Figure 6.2.3 shows the structure of the timer Bi mode register. The operating mode select bits are used to select the operating mode of Timer Bi. Bits 2 and 3 and bits 5 to 7 have different functions according to the operating mode. These bits are described in the paragraph of each operating mode.

b7 b6 b5 b4 b3 b2 b1 b0

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

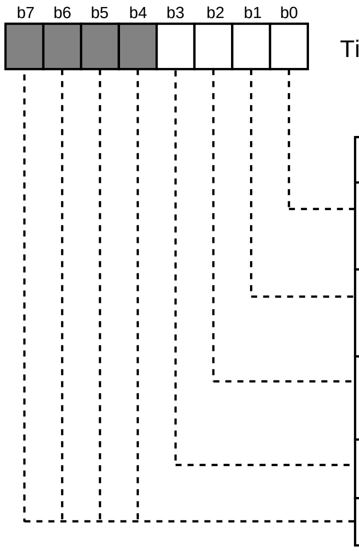
Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/Pulse width measurement mode 1 1 : Not selected	0	RW
1			0	RW
2	These bits have different functions according to the operating mode.		0	RW
3			0	RW
4	Nothing is assigned.		Undefined	–
5	These bits have different functions according to the operating mode.		Undefined	RO **(Note)**
6			0	RW
7			0	RW

Note: Bit 5 is ignored in the timer mode and event counter mode; its value is undefined at reading.

Fig. 6.2.3 Structure of timer Bi mode register

6.2.4 Timer Bi interrupt control register

Figure 6.2.4 shows the structure of the timer Bi interrupt control register. For details about interrupts, refer to **"Chapter 4. INTERRUPTS."**



Timer Bi interrupt control register (i = 0 to 2) (Addresses $7A_{16}$ to $7C_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 Low level 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7 High level	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	RW
7 to 4	Nothing is assigned.		Undefined	–

Fig. 6.2.4 Structure of timer Bi interrupt control register

(1) Interrupt priority level select bits (bits 2 to 0)

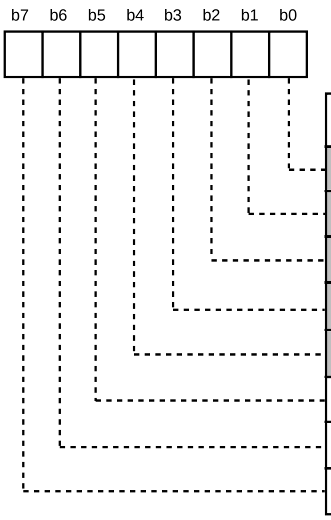
These bits select a timer Bi interrupt's priority level. When using timer Bi interrupts, select priority levels 1 to 7. When the timer Bi interrupt request occurs, its priority level is compared with the processor interrupt priority level (IPL), so that the requested interrupt is enabled only when its priority level is higher than the IPL. (However, this applies when the interrupt disable bit (I) = "0.") To disable timer Bi interrupts, set these bits to "000_2" (level 0).

(2) Interrupt request bit (bit 3)

This bit is set to "1" when the timer Bi interrupt request occurs. This bit is automatically cleared to "0" when the timer Bi interrupt request is accepted. This bit can be set to "1" or cleared to "0" by software.

6.2.5 Port P6 direction register

Timer Bi's input pins are shared with port P6. When using these pins as Timer Bi's input pins, set the corresponding bits of the port P6 direction register to "0" to set these pins for the input mode. Figure 6.2.5 shows the relationship between port P6 direction register and Timer Bi's input pins.



Port P6 direction register (Address 10_{16})

Bit	Corresponding pin name	Functions	At reset	RW
0	$TA4_{OUT}$ pin	0: Input mode 1: Output mode When using these pins as Timer Bi's input pins, set the corresponding bits to "0."	0	RW
1	$TA4_{IN}$ pin		0	RW
2	$\overline{INT_0}$ pin		0	RW
3	$\overline{INT_1}$ pin		0	RW
4	$\overline{INT_2}$ pin		0	RW
5	$TB0_{IN}$ pin		0	RW
6	$TB1_{IN}$ pin		0	RW
7	$TB2_{IN}$ pin		0	RW

(shaded) : Bits 0 to 4 are not used for Timer B.

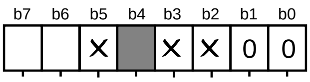
Fig. 6.2.5 Relationship between port P6 direction register and Timer Bi's input pins

6.3 Timer mode

In this mode, the timer counts an internally generated count source. (Refer to Table 6.3.1.) Figure 6.3.1 shows the structures of the timer Bi mode register and timer Bi register in the timer mode.

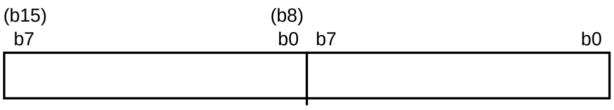
Table 6.3.1 Specifications of timer mode

Item	Specifications
Count source	f_2/f_4, f_{16}/f_{32}, f_{64}/f_{128}, or f_{512}/f_{1024}
Count operation	•Down-count •When the counter underflows, reload register's contents are reloaded and counting continues.
Count start condition	When count start bit is set to "1."
Count stop condition	When count start bit is cleared to "0."
Interrupt request occurrence timing	When the counter underflows.
TBi_{IN} pin function	Programmable I/O port
Read from timer Bi register	Counter value can be read out.
Write to timer Bi register	● While counting is stopped When a value is written to the timer Bi register, it is written to both reload register and counter. ● While counting is in progress When a value is written to the timer Bi register, it is written to only reload register. (Transferred to counter at next reload time.)



Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 0 : Timer mode	0	RW
1			0	RW
2	These bits are ignored in timer mode.		0	RW
3			0	RW
4	Nothing is assigned.		Undefined	–
5	This bit is ignored in timer mode.		Undefined	–
6	Count source select bits	b7 b6 0 0 : f_2/f_4 0 1 : f_{16}/f_{32} 1 0 : f_{64}/f_{128} 1 1 : f_{512}/f_{1024}	0	RW
7			0	RW



Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

Bit	Functions	At reset	RW
15 to 0	These bits can be set to "0000_{16}" to "$FFFF_{16}$." Assuming that the set value = n, the counter divides the count source frequency by n + 1. When reading, the register indicates the counter value.	Undefined	RW

Fig. 6.3.1 Structures of timer Bi mode register and timer Bi register in timer mode

6.3.1 Setting for timer mode

Figure 6.3.2 shows an initial setting example for registers relevant to the timer mode.
Note that when using interrupts, set up to enable the interrupts. For details, refer to **"Chapter 4. INTERRUPTS."**

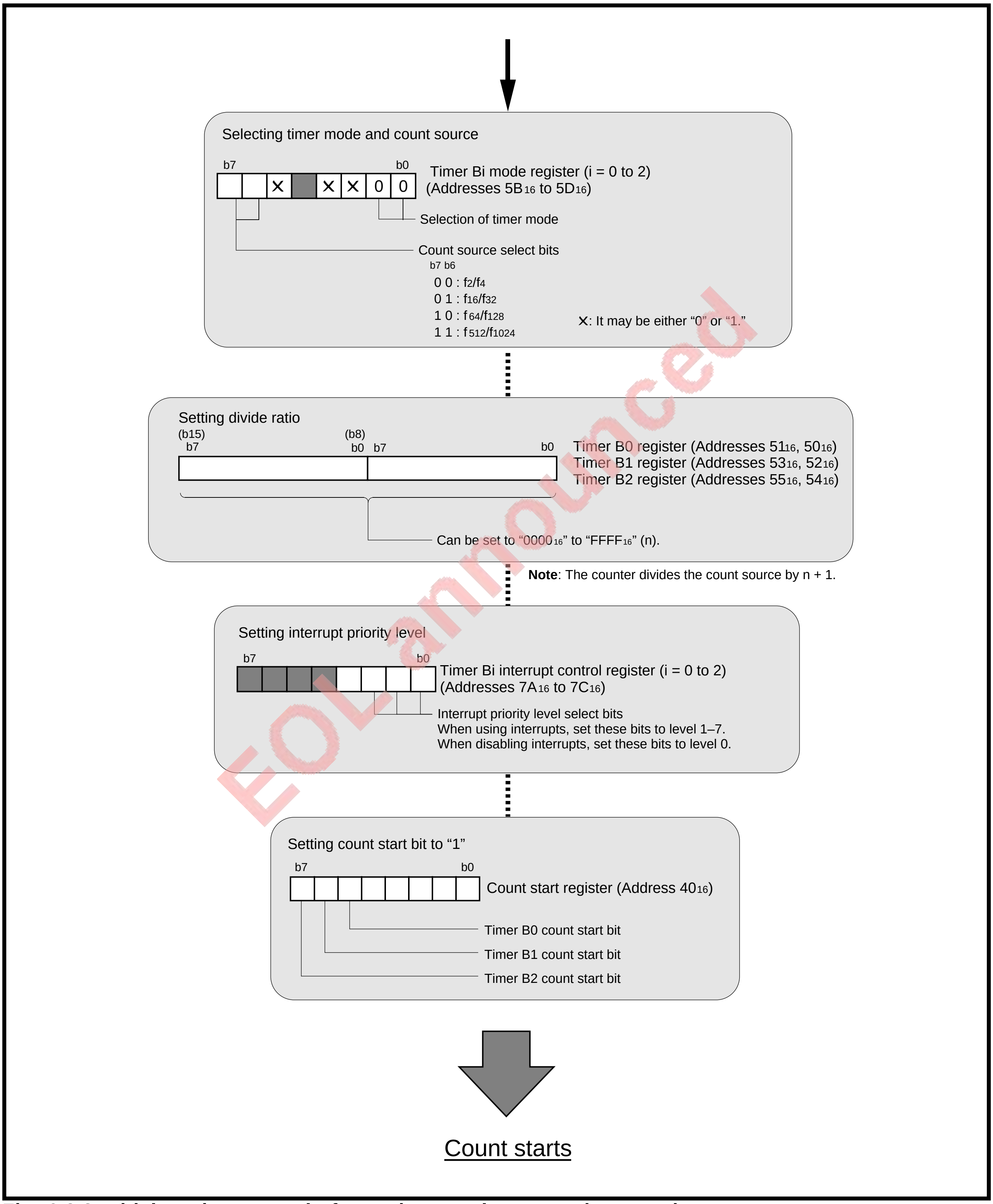


Fig. 6.3.2 Initial setting example for registers relevant to timer mode

6.3.2 Count source

In the timer mode, the count source select bits (bits 6 and 7 at addresses $5B_{16}$ to $5D_{16}$) select the count source. Table 6.3.2 lists the count source frequency.

Table 6.3.2 Count source frequency

Count source select bits		f(XIN) = 25 MHz				f(XIN) = 40 MHz	
		Clock source for peripheral devices select bit = “0”		Clock source for peripheral devices select bit = “1”		Clock source for peripheral devices select bit = “0”	
b7	b6	Count source	Frequency	Count source	Frequency	Count source	Frequency
0	0	f_4	6.25 MHz	f_2	12.5 MHz	f_4	10 MHz
0	1	f_{32}	781.25 kHz	f_{16}	1.5625 MHz	f_{32}	1.25 MHz
1	0	f_{128}	195.3125 kHz	f_{64}	390.625 kHz	f_{128}	312.5 kHz
1	1	f_{1024}	24.4141 kHz	f_{512}	48.8281 kHz	f_{1024}	39.0625 kHz

Clock source for peripheral devices select bit : bit 2 at address $5F_{16}$

6.3.3 Operation in timer mode

① When the count start bit is set to "1," the counter starts counting of the count source.
② When the counter underflows, the reload register's contents are reloaded and counting continues.
③ The timer Bi interrupt request bit is set to "1" when the counter underflows in ②. The interrupt request bit remains set to "1" until the interrupt request is accepted or the interrupt request bit is cleared to "0" by software.

Figure 6.3.3 shows an example of operation in the timer mode.

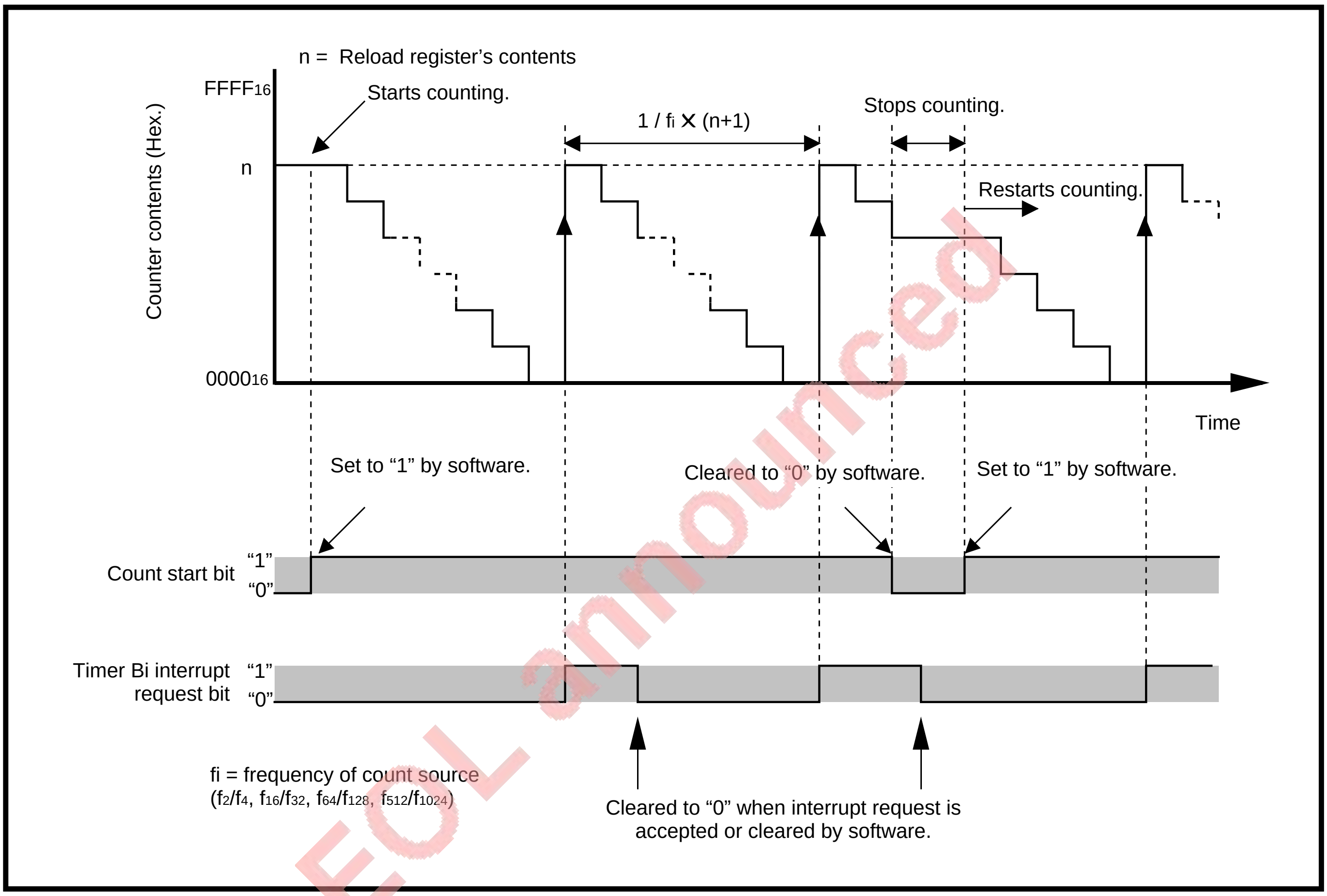


Fig. 6.3.3 Example of operation in timer mode

[Precautions when operating in timer mode]

By reading the timer Bi register, the counter value can be read out at any timing while counting is in progress. However, if the timer Bi register is read at the reload timing shown in Figure 6.3.4, the value "$FFFF_{16}$" is read out. When reading the timer Bi register after setting a value to the register while counting is not in progress and before the counter starts counting, the set value can be read out correctly.

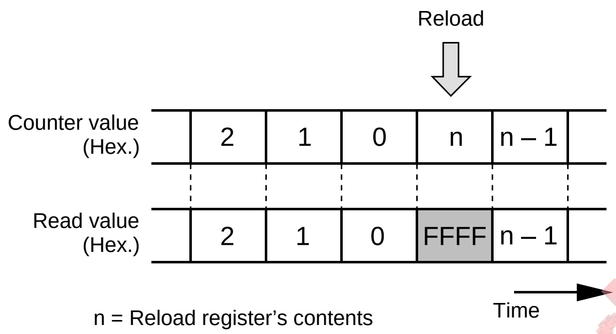


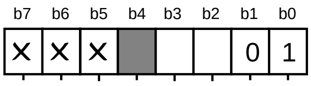
Fig. 6.3.4 Reading timer Bi register

6.4 Event counter mode

In this mode, the timer counts an external signal. (Refer to Table 6.4.1.) Figure 6.4.1 shows the structures of the timer Bi mode register and the timer Bi register in the event counter mode.

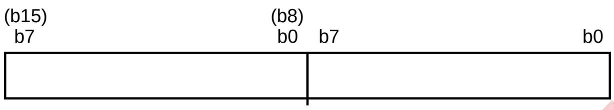
Table 6.4.1 Specifications of event counter mode

Item	Specifications
Count source	•External signal input to the TBiIN pin •The count source's valid edge can be selected from the falling edge, the rising edge, or both of the falling and rising edges by software.
Count operation	•Down-count •When the counter underflows, reload register's contents are reloaded and counting continues.
Count start condition	When count start bit is set to "1."
Count stop condition	When count start bit is cleared to "0."
Interrupt request occurrence timing	When the counter underflows.
TBiIN pin function	Count source input
Read from timer Bi register	Counter value can be read out.
Write to timer Bi register	● While counting is stopped When a value is written to the timer Bi register, it is written to both reload register and counter. ● While counting is in progress When a value is written to the timer Bi register, it is written to only reload register. (Transferred to counter at next reload time.)



Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 1 : Event counter mode	0	RW
1			0	RW
2	Count polarity select bit	b3 b2 0 0 : Count at falling edge of external signal 0 1 : Count at rising edge of external signal 1 0 : Counts at both falling and rising edges of external signal 1 1 : Not selected	0	RW
3			0	RW
4	Nothing is assigned.		Undefined	—
5	This bit is ignored in event counter mode.		Undefined	—
6	These bits are ignored in event counter mode.		0	RW
7			0	RW



Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

Bit	Functions	At reset	RW
15 to 0	These bits can be set to "0000_{16}" to "$FFFF_{16}$." Assuming that the set value = n, the counter divides the count source frequency by n + 1. When reading, the register indicates the counter value.	Undefined	RW

Fig. 6.4.1 Structures of timer Bi mode register and timer Bi register in event counter mode

6.4 Event counter mode

6.4.1 Setting for event counter mode

Figure 6.4.2 shows an initial setting example for registers relevant to the event counter mode.
Note that when using interrupts, set up to enable the interrupts. For details, refer to section **"Chapter 4. INTERRUPTS."**

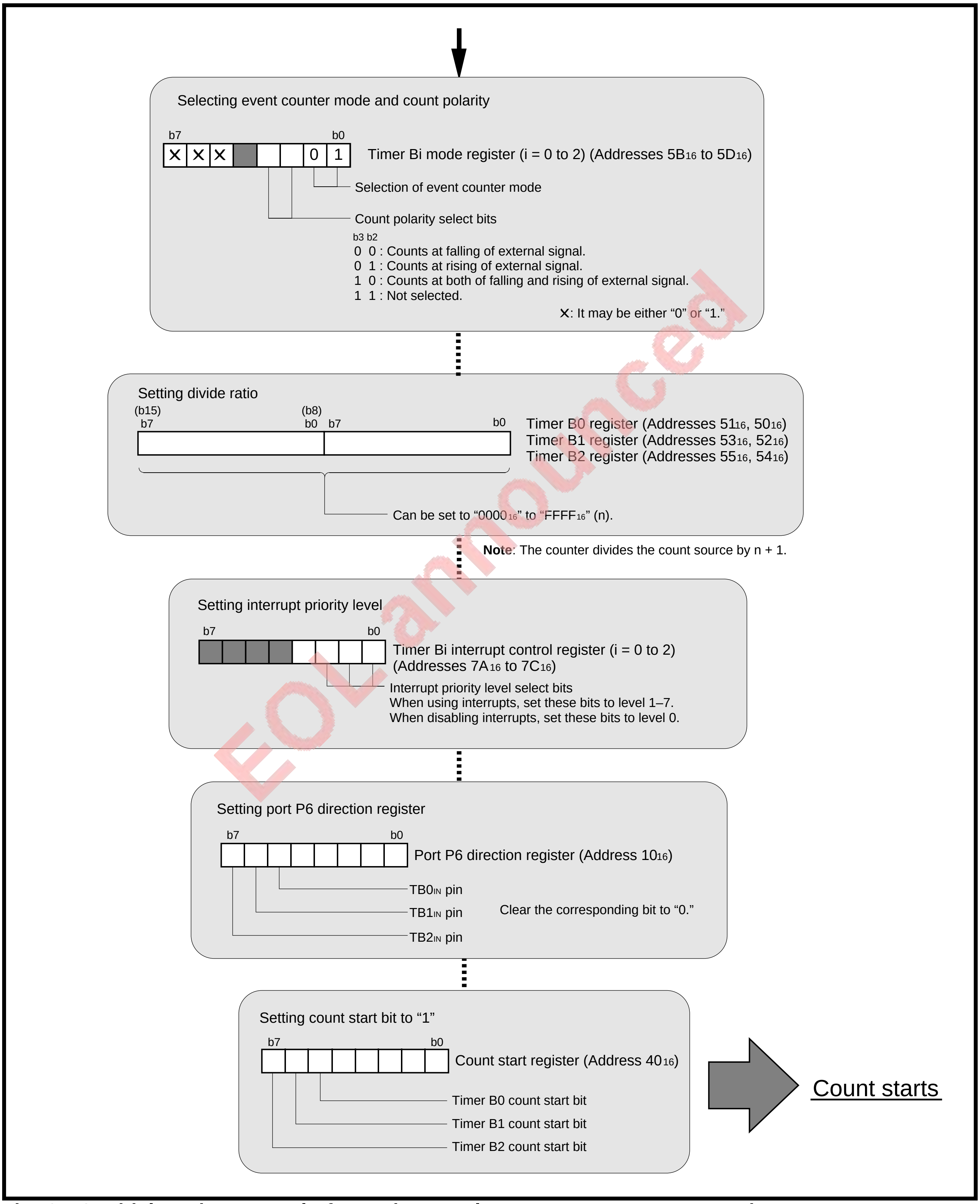


Fig. 6.4.2 Initial setting example for registers relevant to event counter mode

6.4.2 Operation in event counter mode

① When the count start bit is set to "1," the counter starts counting of the count source.
② The counter counts the count source's valid edges.
③ When the counter underflows, the reload register's contents are reloaded and counting continues.
④ The timer Bi interrupt request bit is set to "1" when the counter underflows in ③.
The interrupt request bit remains set to "1" until the interrupt request is accepted or the interrupt request bit is cleared to "0" by software.

Figure 6.4.3 shows an example of operation in the event counter mode.

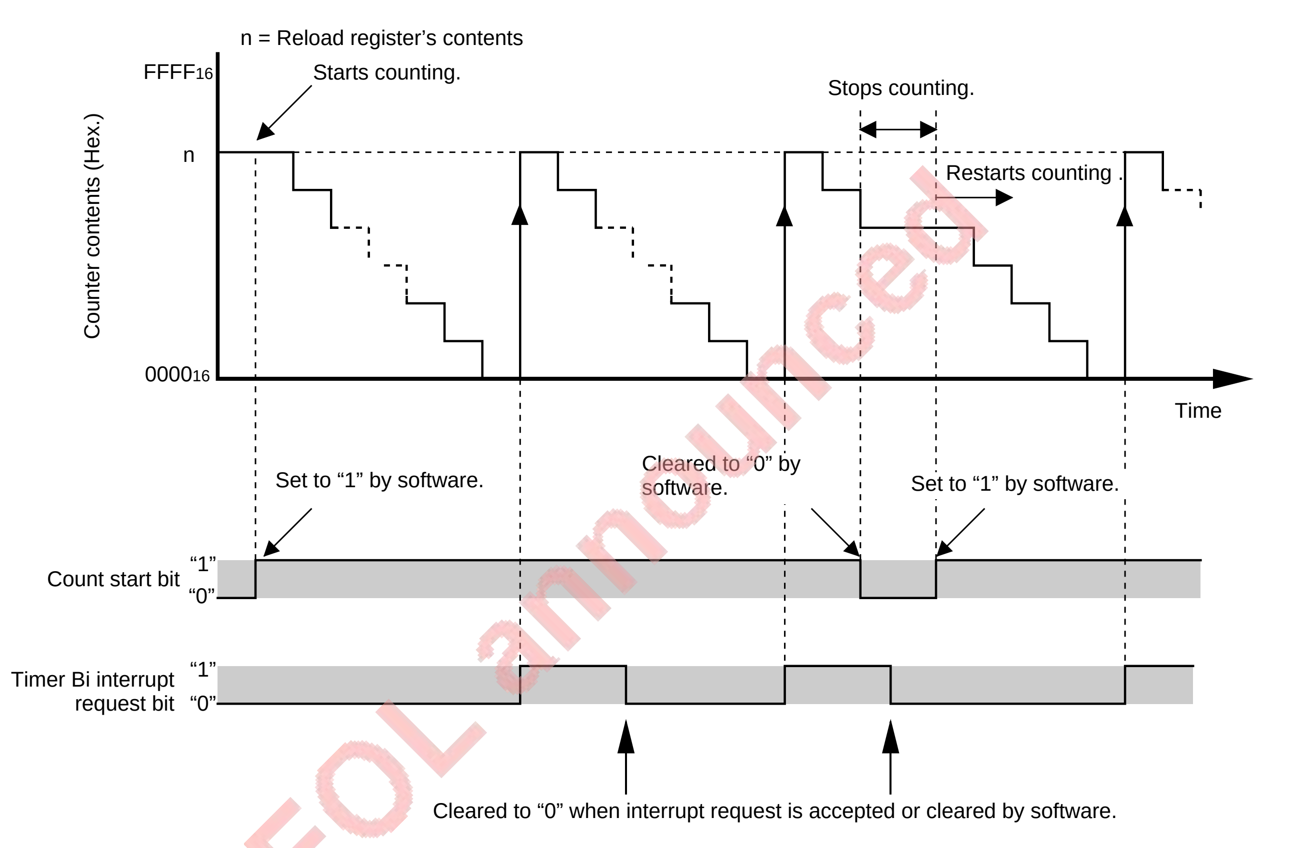


Fig. 6.4.3 Example of operation in event counter mode

[Precautions when operating in event counter mode]

By reading the timer Bi register, the counter value can be read out at any timing while counting is in progress. However, if the timer Bi register is read at the reload timing shown in Figure 6.4.4, the value "$FFFF_{16}$" is read out. When reading the timer Bi register after setting a value to the register while counting is not in progress and before the counter starts counting, the set value can be read out correctly.

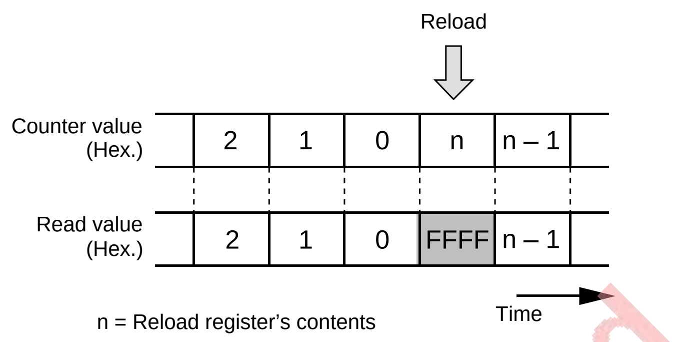


Fig. 6.4.4 Reading timer Bi register

6.5 Pulse period/pulse width measurement mode

In these mode, the timer measures an external signal's pulse period or pulse width. (Refer to Table 6.5.1.) Figure 6.5.1 shows the structures of the timer Bi mode register and timer Bi register in the pulse period/ pulse width measurement mode.

● **Pulse period measurement**
The timer measures the pulse period of the external signal that is input to the TBi_{IN} pin.

● **Pulse width measurement**
The timer measures the pulse width ("L" level and "H" level widths) of the external signal that is input to the TBi_{IN} pin.

Table 6.5.1 Specifications of pulse period/pulse width measurement mode

Item	Specifications
Count source	f_2/f_4, f_{16}/f_{32}, f_{64}/f_{128}, or f_{512}/f_{1024}
Count operation	● Up-count ● Counter value is transferred to reload register at valid edge of measurement pulse, and counting continues after clearing the counter value to "0000_{16}."
Count start condition	When count start bit is set to "1."
Count stop condition	When count start bit is cleared to "0."
Interrupt request occurrence timing	● When valid edge of measurement pulse is input (**Note 1**). ● When counter overflows (Timer Bi overflow flag* is set to "1" simultaneously).
TBi_{IN} pin function	Measurement pulse input
Read from timer Bi register	The value got by reading timer Bi register is the reload register's contents, measurement result (**Note 2**).
Write to timer Bi register	Impossible.

Timer Bi overflow flag*: The bit used to identify the source of an interrupt request occurrence.

Notes 1: This interrupt request does not occur when the first valid edge is input after the timer starts counting.

2: The value read out from the timer Bi register is undefined until the second valid edge is input after the timer starts counting.

6.5 Pulse period/pulse width measurement mode

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

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 1 0 : Pulse period/Pulse width measurement mode	0	RW
1			0	RW
2	Measurement mode select bits	b3 b2 0 0 : Pulse period measurement (Interval between falling edges of measurement pulse) 0 1 : Pulse period measurement (Interval between rising edges of measurement pulse) 1 0 : Pulse width measurement (Interval from falling edge to rising edge, and from rising edge to falling edge of measurement pulse) 1 1 : Not selected	0	RW
3			0	RW
4	Nothing is assigned.		Undefined	–
5	Timer Bi overflow flag **(Note)**	0 : No overflow 1 : Overflow	Undefined	RO
6	Count source select bits	b7 b6 0 0 : f_2/f_4 0 1 : f_{16}/f_{32} 1 0 : f_{64}/f_{128} 1 1 : f_{512}/f_{1024}	0	RW
7			0	RW

Note: The timer Bi overflow flag is cleared to “0” by writing to the timer Bi mode register with the count start bit = “1.”

(b15) b7 … (b8) b0 | b7 … b0

Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

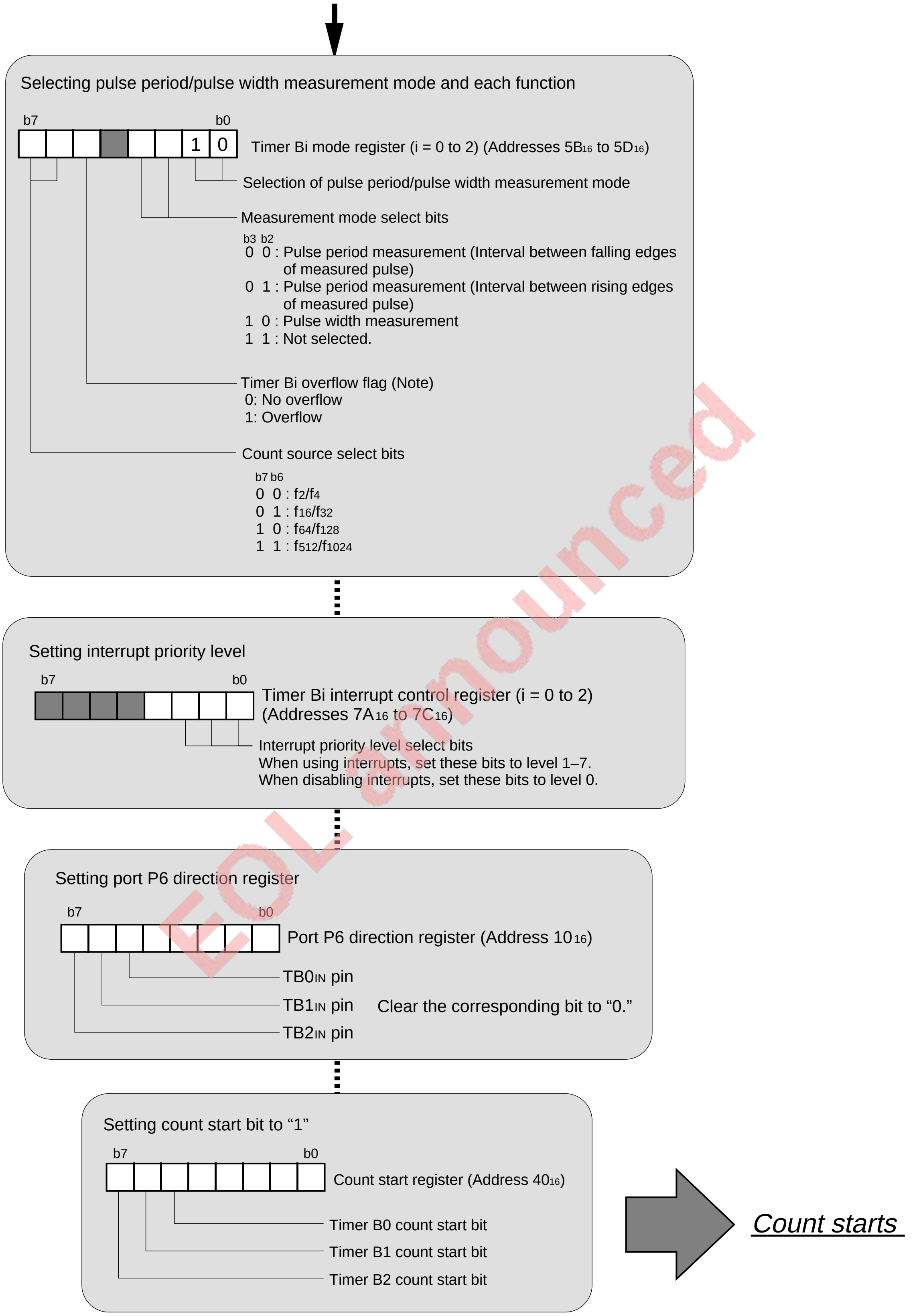
Bit	Functions	At reset	RW
15 to 0	The measurement result of pulse period or pulse width is read out.	Undefined	RO

Fig. 6.5.1 Structures of timer Bi mode register and timer Bi register in pulse period/pulse width measurement mode

6.5.1 Setting for pulse period/pulse width measurement mode

Figure 6.5.2 shows an initial setting example for registers relevant to the pulse period/pulse width measurement mode.
Note that when using interrupts, set up to enable the interrupts. For details, refer to **"Chapter 4. INTERRUPTS."**

6.5 Pulse period/pulse width measurement mode



Note: The timer Bi overflow flag is a read-only bit. This bit is undefined after reset. This bit is cleared to "0" by writing to the timer Bi mode register with the count start bit = "1."

Fig. 6.5.2 Initial setting example for registers relevant to pulse period/pulse width measurement mode

6.5.2 Count source

In the pulse period/pulse width measurement mode, the count source select bits (bits 6 and 7 at addresses $5B_{16}$ to $5D_{16}$) select the count source.
Table 6.5.2 lists the count source frequency.

Table 6.5.2 Count source frequency

Count source select bits		f(XIN) = 25 MHz				f(XIN) = 40 MHz	
		Clock source for peripheral devices select bit = "0"		Clock source for peripheral devices select bit = "1"		Clock source for peripheral devices select bit = "0"	
b7	b6	Count source	Frequency	Count source	Frequency	Count source	Frequency
0	0	f_4	6.25 MHz	f_2	12.5 MHz	f_4	10 MHz
0	1	f_{32}	781.25 kHz	f_{16}	1.5625 MHz	f_{32}	1.25 MHz
1	0	f_{128}	195.3125 kHz	f_{64}	390.625 kHz	f_{128}	312.5 kHz
1	1	f_{1024}	24.4141 kHz	f_{512}	48.8281 kHz	f_{1024}	39.0625 kHz

Clock source for peripheral devices select bit : bit 2 at address $5F_{16}$

6.5.3 Operation in pulse period/pulse width measurement mode

① When the count start bit is set to “1,” the counter starts counting of the count source.
② The counter value is transferred to the reload register when an valid edge of the measurement pulse is detected. (Refer to section **“(1) Pulse period/pulse width measurement.”**)
③ The counter value is cleared to “0000_{16}” after the transfer in ②, and the counter continues counting.
④ The timer Bi interrupt request bit is set to “1” when the counter value is cleared to “0000_{16}” in ③ (**Note**). The interrupt request bit remains set to “1” until the interrupt request is accepted or the interrupt request bit is cleared to “0” by software.
⑤ The timer repeats operations ② to ④ above.

Note: The timer Bi interrupt request does not occur when the first valid edge is input after the timer starts counting.

(1) Pulse period/pulse width measurement

The measurement mode select bits (bits 2 and 3 at addresses $5B_{16}$ to $5D_{16}$) specify whether the pulse period of an external signal is measured or its pulse width is done. Table 6.5.3 lists the relationship between the measurement mode select bits and the pulse period/pulse width measurements.
Make sure that the measurement pulse interval from the falling to the rising, and from the rising to the falling are two cycles of the count source or more. Additionally, use software to identify whether the measurement result indicates the “H” level or the “L” level width.

Table 6.5.3 Relationship between measurement mode select bits and pulse period/pulse width measurements

b3	b2	Pulse period/pulse width measurement	Measurement interval (Valid edges)
0	0	Pulse period measurement	From falling to falling (Falling)
0	1		From rising to rising (Rising)
1	0	Pulse width measurement	From falling to rising, and from rising to falling (Falling and rising)
1	1	Not selected	

(2) Timer Bi overflow flag

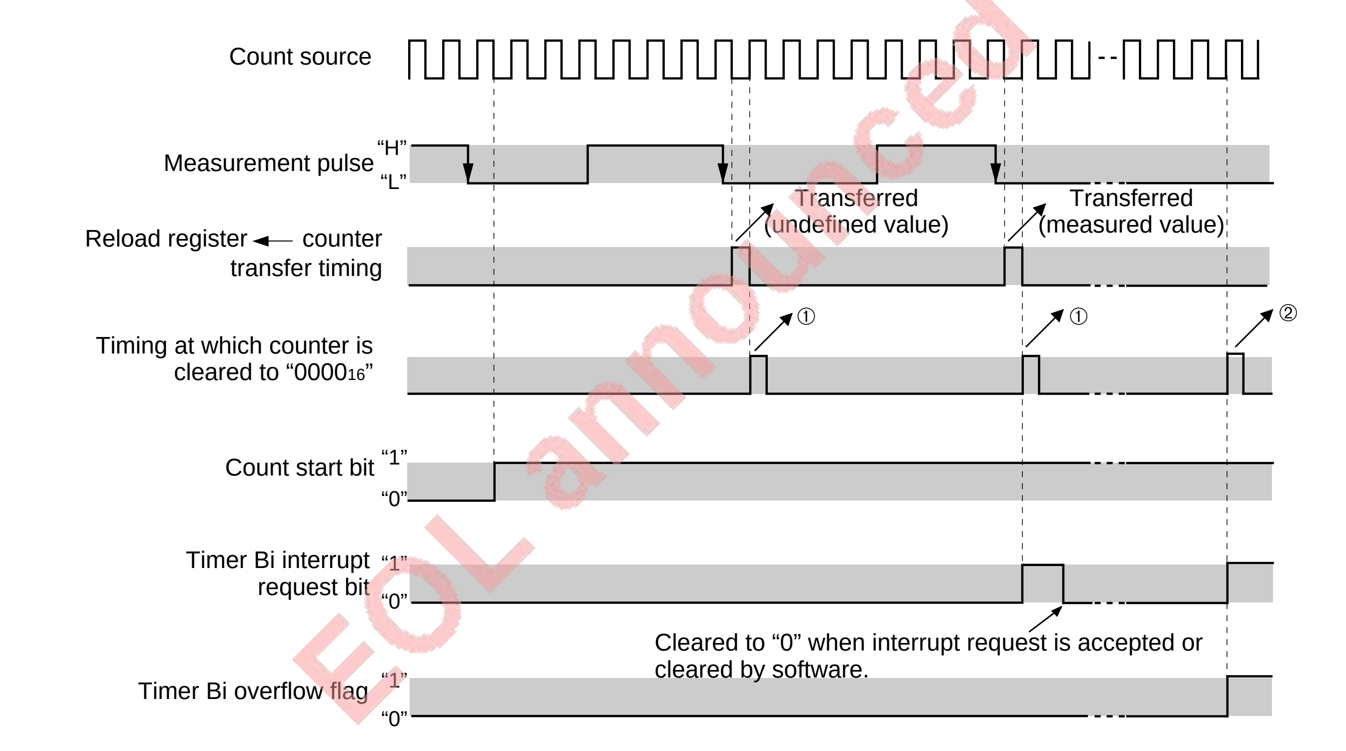
The timer Bi interrupt request occurs when the measurement pulse's valid edge is input or the counter overflows. The timer Bi overflow flag is used to identify the cause of the interrupt request, that is, whether it is an overflow occurrence or an effective edge input.

The timer Bi overflow flag is set to "1" by an overflow. Accordingly, the cause of the interrupt request occurrence is identified by checking the timer Bi overflow flag in the interrupt routine. When a value is written to the timer Bi mode register with the count start bit = "1," the timer Bi overflow flag is cleared to "0" at the next count timing of the count source

The timer Bi overflow flag is a read-only bit.

Use the timer Bi interrupt request bit to detect the overflow timing. Do not use the timer Bi overflow flag to do that.

Figure 6.5.3 shows the operation during pulse period measurement. Figure 6.5.4 shows the operation during pulse width measurement.



① Counter is initialized by completion of measurement.
② Counter overflow.

Note: The above applies when measurement is performed for an interval from one falling to the next falling of the measurement pulse.

Fig. 6.5.3 Operation during pulse period measurement

6.5 Pulse period/pulse width measurement mode

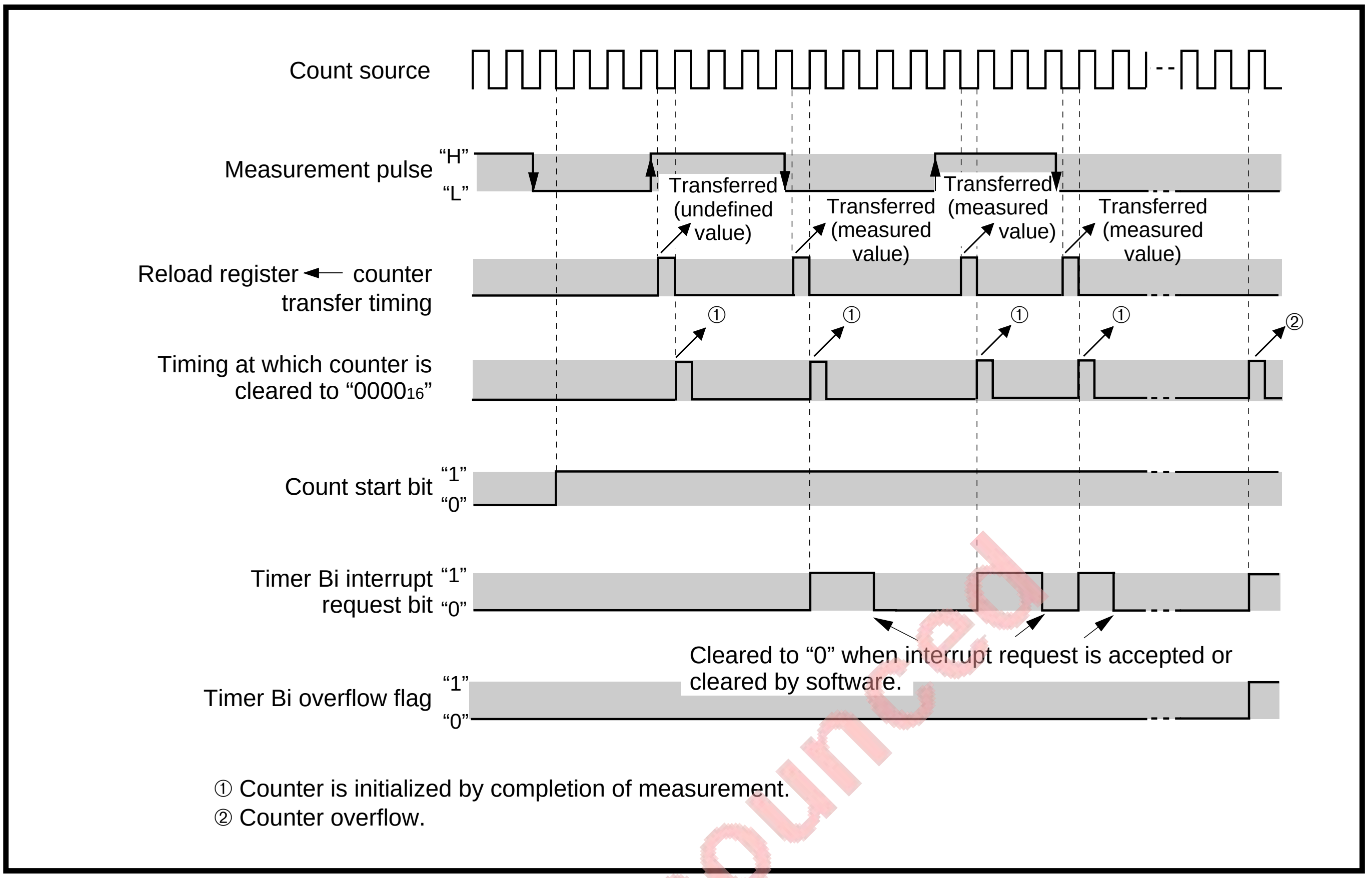


Fig. 6.5.4 Operation during pulse width measurement

[Precautions when operating in pulse period/pulse width measurement mode]

1. The timer Bi interrupt request occurs by the following two causes:
   - Input of measured pulse's valid edge
   - Counter overflow

   When the overflow is the cause of the interrupt request occurrence, the timer Bi overflow flag is set to "1."

2. After reset, the timer Bi overflow flag is undefined. When writing to the timer Bi mode register with the count start bit = "1," this flag can be cleared to "0" at the next count timing of the count source.

3. An undefined value is transferred to the reload register when the first valid edge is input after the counter starts counting. In this case, the timer Bi interrupt request does not occur.

4. The counter value at start of counting is undefined. Accordingly, the timer Bi interrupt request may occur by the overflow immediately after the counter starts counting.

5. If the contents of the measurement mode select bits are changed after the counter starts counting, the timer Bi interrupt request bit is set to "1." When writing the same value which has been set yet to the measurement mode select bits, the timer Bi interrupt request bit is not changed, that is, the bit retains the state.

6. If the input signal to the TBi_{IN} pin is affected by noise, etc., the counter may not perform the exact measurement. We recommend to verify, by software, that the measurement values are within a constant range.

MEMORANDUM

CHAPTER 7
SERIAL I/O

This chapter describes the Serial I/O.
The Serial I/O consists of 2 channels: UART0 and UART1. They each have a transfer clock generating timer for the exclusive use of them and can operate independently. UART0 and UART1 have the same functions.

7.1 Overview

UARTi (i = 0 and 1) has the following 2 operating modes:

- ●Clock synchronous serial I/O mode
  Transmitter and receiver use the same clock as the transfer clock. Transfer data has the length of 8 bits.

- ●Clock asynchronous serial I/O (UART) mode
  Transfer rate and transfer data format can arbitrarily be set. The user can select a 7-bit, 8-bit, or 9-bit length as the transfer data length.

●Clock synchronous serial I/O mode
- **Transfer data length of 8 bits (LSB first)**
- **Transfer data length of 8 bits (MSB first)**

●UART mode
- **Transfer data length of 7 bits**
- **Transfer data length of 8 bits**
- **Transfer data length of 9 bits**

Fig. 7.1.1 Transfer data formats in each operating mode

7.2 Block description

Figure 7.2.1 shows the block diagram of Serial I/O. Registers relevant to Serial I/O are described below.

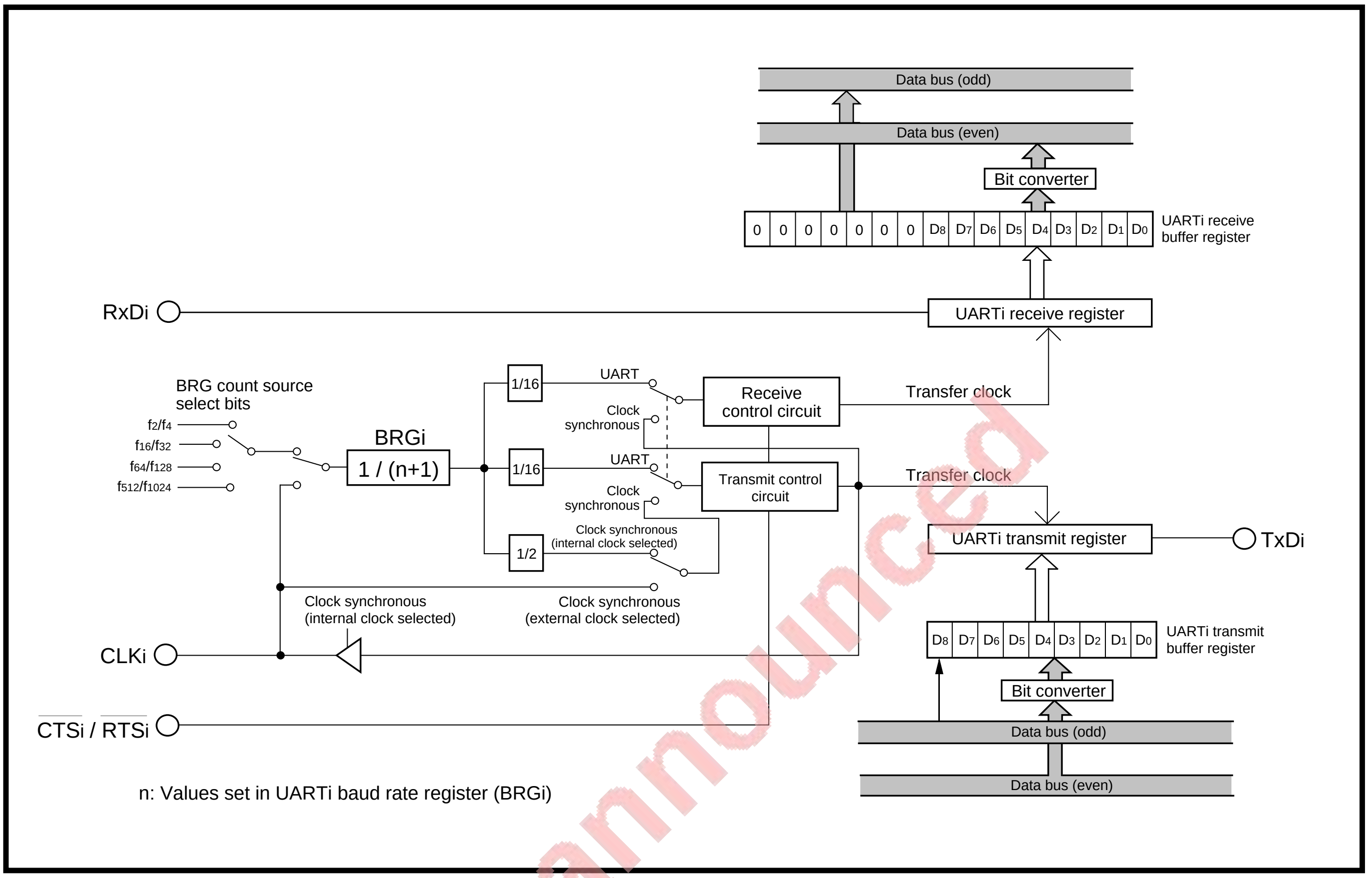
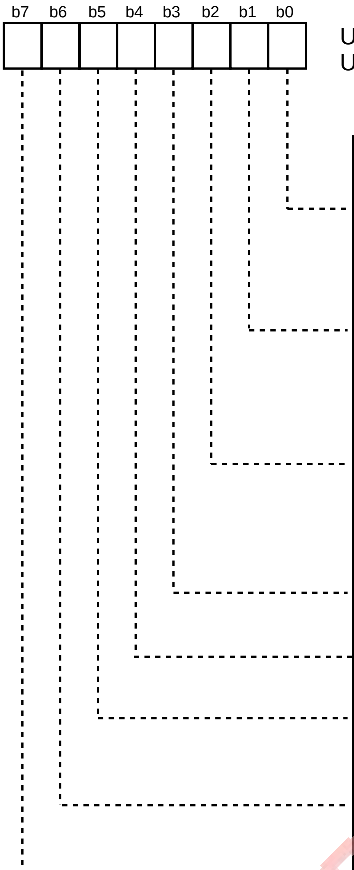


Fig. 7.2.1 Block diagram of Serial I/O

7.2 Block description

7.2.1 UARTi transmit/receive mode register

Figure 7.2.2 shows the structure of UARTi transmit/receive mode register. The serial I/O mode select bits is used to select UARTi's operating mode. Bits 4 to 6 are described in the section **"7.4.2 Transfer data format**," and bit 7 is done in the section "**7.4.8 Sleep mode.**"



UART0 transmit/receive mode register (Address 30_{16})
UART1 transmit/receive mode register (Address 38_{16})

Bit	Bit name	Functions	At reset	RW
0	Serial I/O mode select bits	b2 b1 b0 0 0 0 : Serial I/O disabled (P8 functions as a programmable I/O port.) 0 0 1 : Clock synchronous serial I/O mode 0 1 0 : Not selected 0 1 1 : Not selected 1 0 0 : UART mode (Transfer data length = 7 bits) 1 0 1 : UART mode (Transfer data length = 8 bits) 1 1 0 : UART mode (Transfer data length = 9 bits) 1 1 1 : Not selected	0	RW
1			0	RW
2			0	RW
3	Internal/External clock select bit	0 : Internal clock 1 : External clock	0	RW
4	Stop bit length select bit (Valid in UART mode) (**Note**)	0 : One stop bit 1 : Two stop bits	0	RW
5	Odd/Even parity select bit (Valid in UART mode when parity enable bit is "1") (**Note**)	0 : Odd parity 1 : Even parity	0	RW
6	Parity enable bit (Valid in UART mode) (**Note**)	0 : Parity disabled 1 : Parity enabled	0	RW
7	Sleep select bit (Valid in UART mode) (**Note**)	0 : Sleep mode cleared (ignored) 1 : Sleep mode selected	0	RW

Note: Bits 4 to 6 are ignored in the clock synchronous serial I/O mode. (They may be either "0" or "1.") Additionally, fix bit 7 to "0."

Fig. 7.2.2 Structure of UARTi transmit/receive mode register

(1) Internal/External clock select bit (bit 3)

[Clock synchronous serial I/O mode]

By clearing this bit to “0” in order to select an internal clock, the clock which is selected with the BRG count source select bits (bits 0 and 1 at addresses 34_{16}, $3C_{16}$) becomes the count source of BRGi (described later). The BRGi output of which frequency is divided by 2 becomes the transfer clock. Additionally, the transfer clock is output from the CLK_i pin.

By setting this bit to “1” in order to select an external clock, the clock input to the CLK_i pin becomes the transfer clock.

[UART mode]

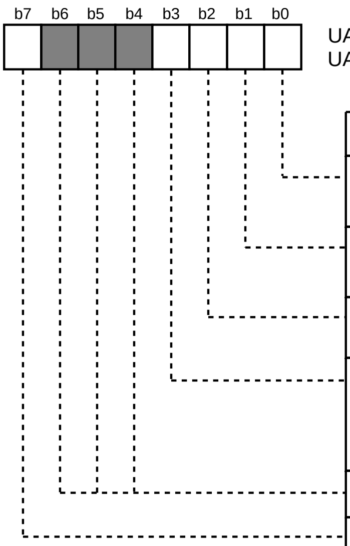
By clearing this bit to “0” in order to select an internal clock, the clock which is selected with the BRG count source select bits (bits 0 and 1 at addresses 34_{16}, $3C_{16}$) becomes the count source of the BRGi (described later). Then, the CLK_i pin functions as a programmable I/O port.

By setting this bit to “1” in order to select an external clock, the clock input to the CLK_i pin becomes the count source of BRGi.

Always in the UART mode, the BRGi output of which frequency is divided by 16 is the transfer clock.

7.2.2 UARTi transmit/receive control register 0

Figure 7.2.3 shows the structure of UARTi transmit/receive control register 0. For bits 0 and 1, refer to **“7.2.1 (1) Internal/External clock select bit.”** For bit 7, refer to **“7.2.2 transfer data format.”**



UART0 transmit/receive control register 0 (Address 34_{16})
UART1 transmit/receive control register 0 (Address $3C_{16}$)

Bit	Bit name	Functions	At reset	RW
0	BRG count source select bits	b1 b0 0 0 : f_2 / f_4 0 1 : f_{16} / f_{32} 1 0 : f_{64} / f_{128} 1 1 : f_{512} / f_{1024}	0	RW
1			0	RW
2	$\overline{CTS}$/$\overline{RTS}$ select bit	0 : $\overline{CTS}$ function selected. 1 : $\overline{RTS}$ function selected.	0	RW
3	Transmit register empty flag	0 : Data present in transmit register. (During transmitting) 1 : No data present in transmit register. (Transmitting completed)	1	RO
6 to 4	Nothing is assigned.		Undefined	—
7	Transfer format select bit (Used in clock synchronous serial I/O mode) **(Note)**	0 : LSB (Least Significant Bit) first 1 : MSB (Most Significant Bit) first	0	RW

Note: Fix bit 7 to “0” in the UART mode or when Serial I/O is ignored.

Fig. 7.2.3 Structure of UARTi transmit/receive control register 0

(1) $\overline{CTS}$/$\overline{RTS}$ select bit (bit 2)

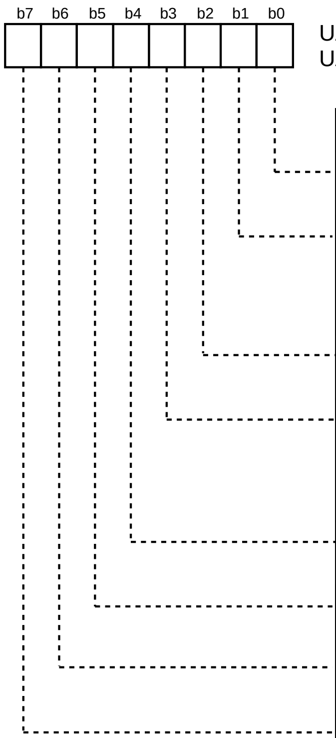
By clearing this bit to “0” in order to select the $\overline{CTS}$ function, pins $P8_0$ and $P8_4$ function as $\overline{CTS}$ input pins, and the input signal of “L” level to these pins becomes one of the transmission conditions.
By setting this bit to “1” in order to select the $\overline{RTS}$ function, pins $P8_0$ and $P8_4$ become $\overline{RTS}$ output pins. When the receive enable bit (bit 2 at addresses 35_{16}, $3D_{16}$) is “0” (reception disabled), the $\overline{RTS}$ output pin outputs “H” level.
The output level of this pin becomes “L” when the receive enable bit is set to “1.” It becomes “H” when reception starts and it becomes “L” when reception is completed.

(2) Transmit register empty flag (bit 3)

This flag is cleared to “0” when the UARTi transmit buffer register’s contents are transferred to the UARTi transmit register. When transmission is completed and the UARTi transmit register becomes empty, this flag is set to “1.”

7.2.3 UARTi transmit/receive control register 1

Figure 7.2.4 shows the structure of UARTi transmit/receive control register 1. For bits 4 to 7, refer to each operation mode's description.



UART0 transmit/receive control register 1 (Address 35_{16})
UART1 transmit/receive control register 1 (Address $3D_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0	RW
1	Transmit buffer empty flag	0 : Data present in transmit buffer register. 1 : No data present in transmit buffer register.	1	RO
2	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0	RW
3	Receive complete flag	0 : No data present in receive buffer register. 1 : Data present in receive buffer register.	0	RO
4	Overrun error flag **(Note 1)**	0 : No overrun error 1 : Overrun error detected	0	RO
5	Framing error flag **(Notes 1, 2)** (Valid in UART mode)	0 : No framing error 1 : Framing error detected	0	RO
6	Parity error flag **(Notes 1, 2)** (Valid in UART mode)	0 : No parity error 1 : Parity error detected	0	RO
7	Error sum flag **(Notes 1, 2)** (Valid in UART mode)	0 : No error 1 : Error detected	0	RO

Notes 1: Bit 4 is cleared to "0" when clearing the receive enable bit to "0."
Bits 5 and 6 are cleared to "0" when one of the following is performed:
•clearing the receive enable bit to "0."
•reading the low-order byte of the UARTi receive buffer register (addresses 36_{16}, $3E_{16}$) out.
Bit 7 is cleared to "0" when all of bits 4 to 6 become "0."
2: Bits 5 to 7 are ignored in the clock synchronous serial I/O mode.

Fig. 7.2.4 Structure of UARTi transmit/receive control register 1

(1) Transmit enable bit (bit 0)

By setting this bit to “1,” UARTi enters the transmission enable state. By clearing this bit to “0” during transmission, UARTi enters the transmission disable state after the transmission which is performed at that time is completed.

(2) Transmit buffer empty flag (bit 1)

This flag is set to “1” when data set in the UARTi transmit buffer register is transferred from the UARTi transmit buffer register to the UARTi transmit register. This flag is cleared to “0” when data is set in the UARTi transmit buffer register.

(3) Receive enable bit (bit 2)

By setting this bit to “1,” UARTi enters the reception enable state. By clearing this bit to “0” during reception, UARTi quits the reception then and enters the reception disable state.

(4) Receive complete flag (bit 3)

This flag is set to “1” when data is ready in the UARTi receive register and that is transferred to the UARTi receive buffer register (i.e., when reception is completed). This flag is cleared to “0” when the low-order byte of the UARTi receive buffer register is read out or when the receive enable bit (bit 2) is cleared to “0.”

7.2.4 UARTi transmit register and UARTi transmit buffer register

Figure 7.2.5 shows the block diagram of transmit section; Figure 7.2.6 shows the structure of UARTi transmit buffer register.

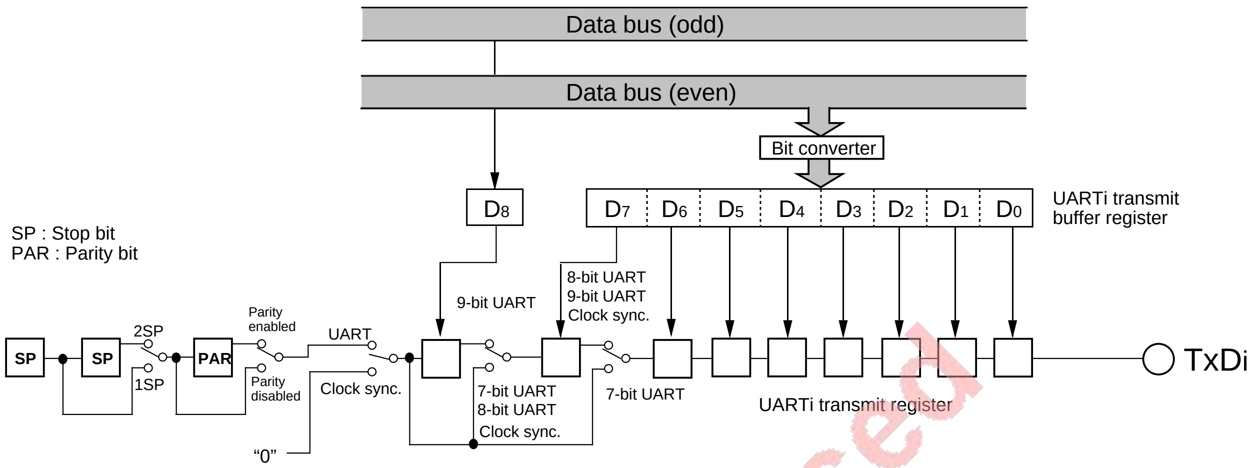
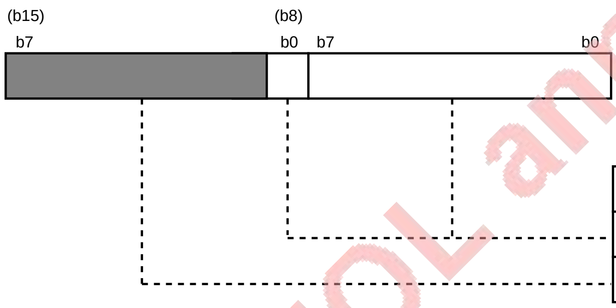


Fig. 7.2.5 Block diagram of transmit section



UART0 transmit buffer register (Addresses 33_{16}, 32_{16})
UART1 transmit buffer register (Addresses $3B_{16}$, $3A_{16}$)

Bit	Functions	At reset	RW
8 to 0	Transmit data is set.	Undefined	WO
15 to 9	Nothing is assigned.	Undefined	–

Fig. 7.2.6 Structure of UARTi transmit buffer register

7.2 Block description

The UARTi transmit buffer register is used to set transmit data. Set the transmit data into the low-order byte of this register when operating in the clock synchronous serial I/O mode or when a 7-bit or 8-bit length of transfer data is selected in the UART mode. When a 9-bit length of transfer data is selected in the UART mode, set the transmit data into the UARTi transmit buffer register as follows:

•Bit 8 of the transmit data into bit 0 of high-order byte of this register.
•Bits 7 to 0 of the transmit data into the low-order byte of this register.

The transmit data which is set in the UARTi transmit buffer register is transferred to the UARTi transmit register when the transmission conditions are satisfied, and then it is output from the TxD_i pin synchronously with the transfer clock. The UARTi transmit buffer register becomes empty when the data which is set in the UARTi transmit buffer register is transferred to the UARTi transmit register. Accordingly, the user can set next transmit data.

When selecting the “MSB first” in the clock synchronous serial I/O mode, the data of which bit position was reversed is written, as a transmit data, into the UARTi transmit buffer register. (Refer to section **“7.3.2 Transfer data format.”**) Transmission operation itself is the same whichever format is selected, “LSB first” or “MSB first.”

When quitting the transmission which is in progress and setting the UARTi transmit buffer register again, follow the procedure described bellow:

① Clear the serial I/O mode select bits (bits 2 to 0 at addresses 30_{16}, 38_{16}) to “000_2” (Serial I/O disabled).
② Set the serial I/O mode select bits again.
③ Set the transmit enable bit (bit 0 at addresses 35_{16}, $3D_{16}$) to “1” (transmission enabled) and set transmit data in the UARTi transmit buffer register.

7.2.5 UARTi receive register and UARTi receive buffer register

Figure 7.2.7 shows the block diagram of receive section; Figure 7.2.8 shows the structure of UARTi receive buffer register.

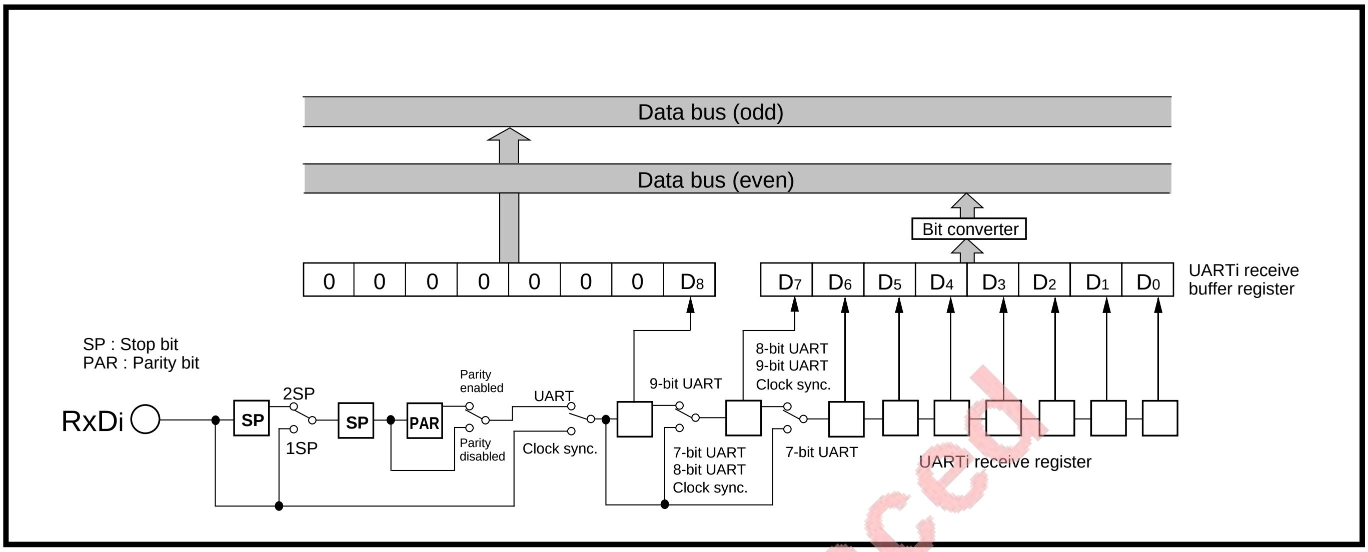
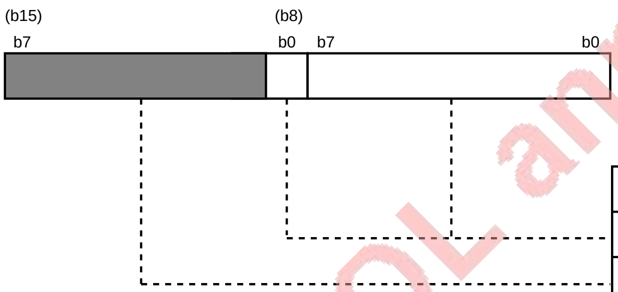


Fig. 7.2.7 Block diagram of receive section



UART0 receive buffer register (Addresses 37_{16}, 36_{16})
UART1 receive buffer register (Addresses $3F_{16}$, $3E_{16}$)

Bit	Functions	At reset	RW
8 to 0	Receive data is read out from here.	Undefined	RO
15 to 9	Nothing is assigned. The value is “0” at reading.	0	–

Fig. 7.2.8 Structure of UARTi receive buffer register

The UARTi receive register is used to convert serial data which is input to the RxD_i pin into parallel data. This register takes in the input signal to the RxD_i pin synchronously with the transfer clock, one bit at a time.
The UARTi receive buffer register is used to read out receive data. When reception is completed, receive data which is taken in the UARTi receive register is automatically transferred to the UARTi receive buffer register. The contents of UARTi receive buffer register is updated when the next data is ready before reading out the data which has been transferred to the UARTi receive buffer register (i.e., an overrun error occurs).
When selecting the "MSB first" in the clock synchronous serial I/O mode, bit position of data in the UARTi receive buffer register is reversed, and then the data of which bit position was reversed is read out, as receive data. (Refer to section **"7.3.2 Transfer data format."**) Reception operation itself is the same whichever format is selected, "LSB first" or "MSB first."
The UARTi receive buffer register is initialized by setting the receive enable bit (bit 2 at addresses 35_{16}, $3D_{16}$) to "1" after clearing it to "0."
Figure 7.2.9 shows the contents of UARTi receive buffer register when reception is completed.

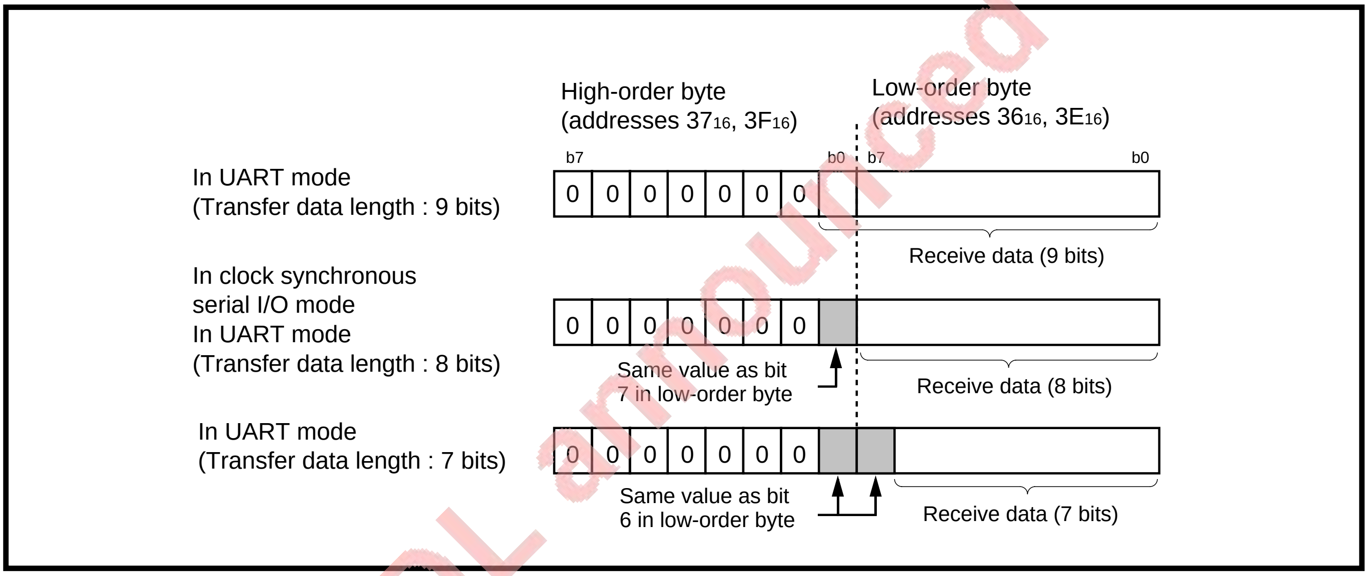


Fig. 7.2.9 Contents of UARTi receive buffer register when reception is completed

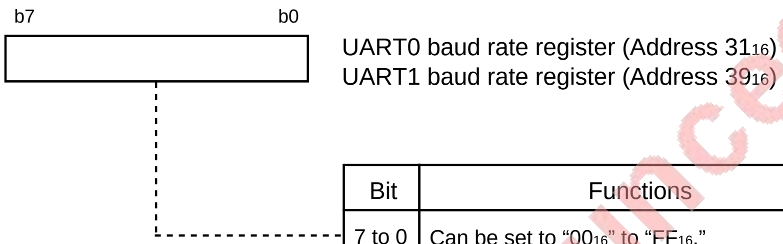
7.2.6 UARTi baud rate register (BRGi)

The UARTi baud rate register (BRGi) is an 8-bit timer exclusively used for UARTi to generate a transfer clock. It has a reload register. Assuming that a value set in the BRGi is "n" (n = "00_{16}" to "FF_{16}"), the BRGi divides the count source frequency by n + 1.

In the clock synchronous serial I/O mode, the BRGi is valid when an internal clock is selected, and a clock of which frequency is the BRGi output's frequency divided by 2 becomes the transfer clock. In the UART mode, the BRGi is always valid, and a clock of which frequency is the BRGi output's frequency divided by 16 becomes the transfer clock.

The data which is written to the addresses 31_{16} and 39_{16} is written to both the timer register and the reload register whether transmission/reception is stopped or in progress. Accordingly, writing to their addresses, perform it while that is stopped.

Figure 7.2.10 shows the structure of the UARTi baud rate register (BRGi); Figure 7.2.11 shows the block diagram of transfer clock generating section.



Bit	Functions	At reset	RW
7 to 0	Can be set to "00_{16}" to "FF_{16}." Assuming that the set value = n, BRGi divides the count source frequency by n + 1.	Undefined	WO

Fig. 7.2.10 Structure of UARTi baud rate register (BRGi)

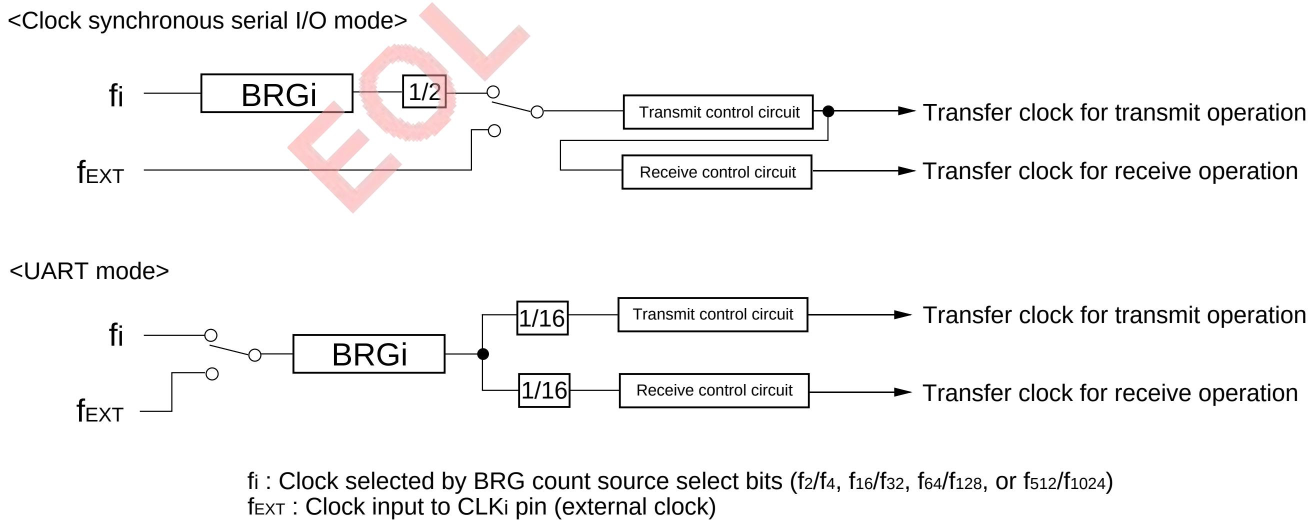
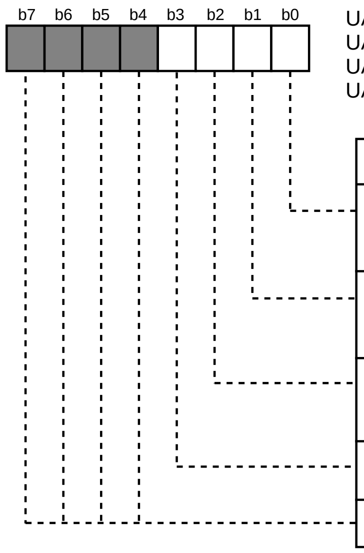


Fig. 7.2.11 Block diagram of transfer clock generating section

7.2.7 UARTi transmit interrupt control and UARTi receive interrupt control registers

When using UARTi, 2 types of interrupts, which are UARTi transmit and UARTi receive interrupts, can be used. Each interrupt has its corresponding interrupt control register. Figure 7.2.12 shows the structure of UARTi transmit interrupt control and UARTi receive interrupt control registers.
For details about interrupts, refer to **"Chapter 4. INTERRUPTS."**



UART0 transmit interrupt control register (Address 71_{16})
UART0 receive interrupt control register (Address 72_{16})
UART1 transmit interrupt control register (Address 73_{16})
UART1 receive interrupt control register (Address 74_{16})

Bit	Bit name	Functions	At reset	RW
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 (Low level) 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7 (High level)	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	RW
7 to 4	Nothing is assigned.		Undefined	–

Fig. 7.2.12 Structure of UARTi transmit interrupt control and UARTi receive interrupt control registers

(1) Interrupt priority level select bits (bits 0 to 2)

These bits select the priority level of the UARTi transmit interrupt or UARTi receive interrupt. When using UARTi transmit/receive interrupt, select priority levels 1 to 7. When the UARTi transmit/receive interrupt request occurs, its priority level is compared with the processor interrupt priority level (IPL), so that the requested interrupt is enabled only when its priority level is higher than the IPL. (However, this applies when the interrupt disable flag (I) = “0.”) To disable the UARTi transmit/receive interrupt, set these bits to “000_2” (level 0).

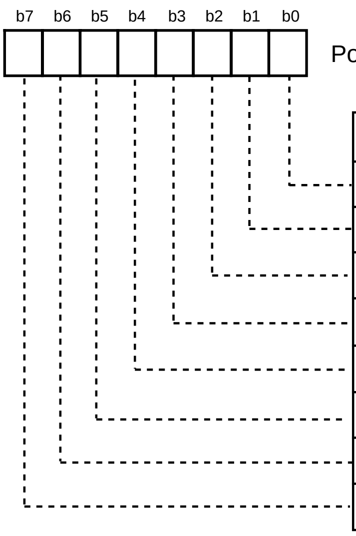
(2) Interrupt request bit (bit 3)

The UARTi transmit interrupt request bit is set to “1” when data is transferred from the UARTi transmit buffer register to the UARTi transmit register. The UARTi receive interrupt request bit is set to “1” when data is transferred from the UARTi receive register to the UARTi receive buffer register. However, when an overrun error occurs, it does not change.

Each interrupt request bit is automatically cleared to “0” when its corresponding interrupt request is accepted. This bit can be set to “1” or “0” by software.

7.2.8 Port P8 direction register

I/O pins of UARTi are shared with port P8. When using pins $P8_2$ and $P8_6$ as serial data input pins (RxD_i), set the corresponding bits of the port P8 direction register to "0" to set these pins for the input mode. When using pins $P8_0$, $P8_1$, $P8_3$ to $P8_5$ and $P8_7$ as I/O pins ($\overline{CTS_i}/\overline{RTS_i}$, CLK_i, TxD_i) of UARTi, these pins are forcibly set as I/O pins of UARTi regardless of port P8 direction register's contents. Figure 7.2.13 shows the relationship between the port P8 direction register and UARTi's I/O pins.



Port P8 direction register (Address 14_{16})

Bit	Corresponding pin	Functions	At reset	RW
0	$\overline{CTS_0}/\overline{RTS_0}$ pin	0 : Input mode 1 : Output mode When using pins $P8_2$ and $P8_6$ as serial data input pins (RxD_0, RxD_1), set the corresponding bits to "0."	0	RW
1	CLK_0 pin		0	RW
2	RxD_0 pin		0	RW
3	TxD_0 pin		0	RW
4	$\overline{CTS_1}/\overline{RTS_1}$ pin		0	RW
5	CLK_1 pin		0	RW
6	RxD_1 pin		0	RW
7	TxD_1 pin		0	RW

Fig. 7.2.13 Relationship between port P8 direction register and UARTi's I/O pins

7.3 Clock synchronous serial I/O mode

Table 7.3.1 lists the performance overview in the clock synchronous serial I/O mode, and Table 7.3.2 lists the functions of I/O pins in this mode.

Table 7.3.1 Performance overview in clock synchronous serial I/O mode

Item		Functions
Transfer data format		Transfer data has a length of 8 bits. LSB first or MSB first can be selected by software.
Transfer rate	When selecting internal clock	Clock which is BRGi output's divided by 2.
	When selecting external clock	Maximum 5 Mbps
Transmit/Receive control		$\overline{CTS}$ function or $\overline{RTS}$ function can be selected by software.

Table 7.3.2 Functions of I/O pins in clock synchronous serial I/O mode

Pin name	Functions	Method of selection
TxD_i ($P8_3$, $P8_7$)	Serial data output	Fixed (Dummy data is output when performing only reception.)
RxD_i ($P8_2$, $P8_6$)	Serial data input	Port P8 direction register*1's corresponding bit = "0"
CLK_i ($P8_1$, $P8_5$)	Transfer clock output	Internal/External clock select bit*2 = "0"
	Transfer clock input	Internal/External clock select bit = "1"
$\overline{CTS_i}$/$\overline{RTS_i}$ ($P8_0$, $P8_4$)	$\overline{CTS}$ input	$\overline{CTS}$/$\overline{RTS}$ select bit*3 = "0"
	$\overline{RTS}$ output	$\overline{CTS}$/$\overline{RTS}$ select bit = "1"

Port P8 direction register*1: Address 14_{16}
Internal/External clock select bit*2: bit 3 at addresses 30_{16}, 38_{16}
$\overline{CTS}$/$\overline{RTS}$ select bit*3: bit 2 at addresses 34_{16}, $3C_{16}$

Notes 1: The TxD_i pin outputs "H" level until transmission starts after UARTi's operating mode is selected.
2: The RxD_i pin can be used as a programmable I/O port when performing only transmission.

7.3 Clock synchronous serial I/O mode

7.3.1 Transfer clock (synchronizing clock)

Data transfer is performed synchronously with the transfer clock. For the transfer clock, the user can select whether to generate the transfer clock internally or to input it from an external.
The transfer clock is generated by operation of the transmit control circuit. Accordingly, even when performing only reception, set the transmit enable bit to "1," and set dummy data in the UARTi transmit buffer register in order to make the transmit control circuit active.

(1) Generating transfer clock internally

The count source selected with the BRG count source select bits is divided by the BRGi, and its BRGi output is further divided by 2. This is the transfer clock. The transfer clock is output from the CLK_i pin.

[Setting relevant registers]

•Select an internal clock (bit 3 at addresses 30_{16}, 38_{16} = "0").
•Select the BRGi's count source (bits 0 and 1 at addresses 34_{16}, $3C_{16}$)
•Set "divide value – 1" (= n; 00_{16} to FF_{16}) to the BRGi (addresses 31_{16}, 39_{16}).

$$\text{Transfer clock frequency} = \frac{f_i}{2\,(n+1)}$$

f_i: Frequency of BRGi's count source (f_2/f_4, f_{16}/f_{32}, f_{64}/f_{128} , f_{512}/f_{1024})

•Enable transmission (bit 0 at addresses 35_{16}, $3D_{16}$ = "1").
•Set data to the UARTi transmit buffer register (addresses 32_{16}, $3A_{16}$)

[Pin's state]

•A transfer clock is output from the CLK_i pin.
•Serial data is output from the TxDi pin. (Dummy data is output when performing only reception.)

(2) Inputting transfer clock from an external

A clock input from the CLK_i pin is the transfer clock.

[Setting relevant registers]

•Select an external clock (bit 3 at addresses 30_{16}, 38_{16} = "1").
•Enable transmission (bit 0 at addresses 35_{16}, $3D_{16}$ = "1").
•Set data to the UARTi transmit buffer register (addresses 32_{16}, $3A_{16}$).

[Pin's state]

•A transfer clock is input from the CLK_i pin.
•Serial data is output from the TxD_i pin. (Dummy data is output when performing only reception.)

7.3.2 Transfer data format

LSB first or MSB first can be selected as the transfer data format. Table 7.3.3 lists the relationship between the transfer data format and writing/reading to and from the UARTi transmit/receive buffer register.

The transfer format select bit (bit 7 at addresses 34_{16}, $3C_{16}$) selects the transfer data format. When this bit is cleared to "0," the set data is written to the UARTi transmit buffer register as the transmit data as it is. Similarly, the data in the UARTi receive buffer register is read out as the receive data as it is. (Refer to the upper row in Table 7.3.3.)

When this bit is set to "1," each bit's position of set data is reversed, and the resultant data is written to the UARTi transmit buffer register as the transmit data. Similarly, each bit's position of data in the UARTi receive buffer register is reversed, and the resultant data is read out as the receive data. (Refer to the lower row in Table 7.3.3.)

Note that only the method of writing/reading to and from the UARTi transmit/receive buffer register is affected by selection of the transfer data format, and that the transmit/receive operation is unaffected by it.

Table 7.3.3 Relationship between transfer data format and writing/reading to and from UARTi transmit/ receive buffer register

Transfer format select bit	Transfer data format	Writing to UARTi transmit buffer register	Reading from UARTi receive buffer register
0	LSB (Least Significant Bit) first	Data bus → UARTi transmit buffer register DB7 → D_7 DB6 → D_6 DB5 → D_5 DB4 → D_4 DB3 → D_3 DB2 → D_2 DB1 → D_1 DB0 → D_0	Data bus ← UARTi receive buffer register DB7 ← D_7 DB6 ← D_6 DB5 ← D_5 DB4 ← D_4 DB3 ← D_3 DB2 ← D_2 DB1 ← D_1 DB0 ← D_0
1	MSB (Most Significant Bit) first	Data bus → UARTi transmit buffer register DB7 → D_0 DB6 → D_1 DB5 → D_2 DB4 → D_3 DB3 → D_4 DB2 → D_5 DB1 → D_6 DB0 → D_7	Data bus ← UARTi receive buffer register DB7 ← D_0 DB6 ← D_1 DB5 ← D_2 DB4 ← D_3 DB3 ← D_4 DB2 ← D_5 DB1 ← D_6 DB0 ← D_7

7.3.3 Method of transmission

Figures 7.3.1 shows an initial setting example for relevant registers when transmitting. Transmission is started when all of the following conditions (① to ③) are satisfied. When an external clock is selected, satisfy conditions ① to ③ with the following precondition satisfied.

<Precondition>
The CLK_i pin's input is "H" level
Note: When an internal clock is selected, above precondition is ignored.

<Transmission conditions>
① Transmission is enabled (transmit enable bit = "1").
② Transmit data is present in the UARTi transmit buffer register (transmit buffer empty flag = "0")
③ $\overline{CTS_i}$ pin's input is "L" level (when $\overline{CTS}$ function selected).
Note: When the $\overline{CTS}$ function is not selected, this condition is ignored.

When using interrupts, it is necessary to set the relevant register to enable interrupts. For details, refer to **"Chapter 4. INTERRUPTS."**
Figure 7.3.2 shows writing data after start of transmission, and Figure 7.3.3 shows detection of transmission's completion.

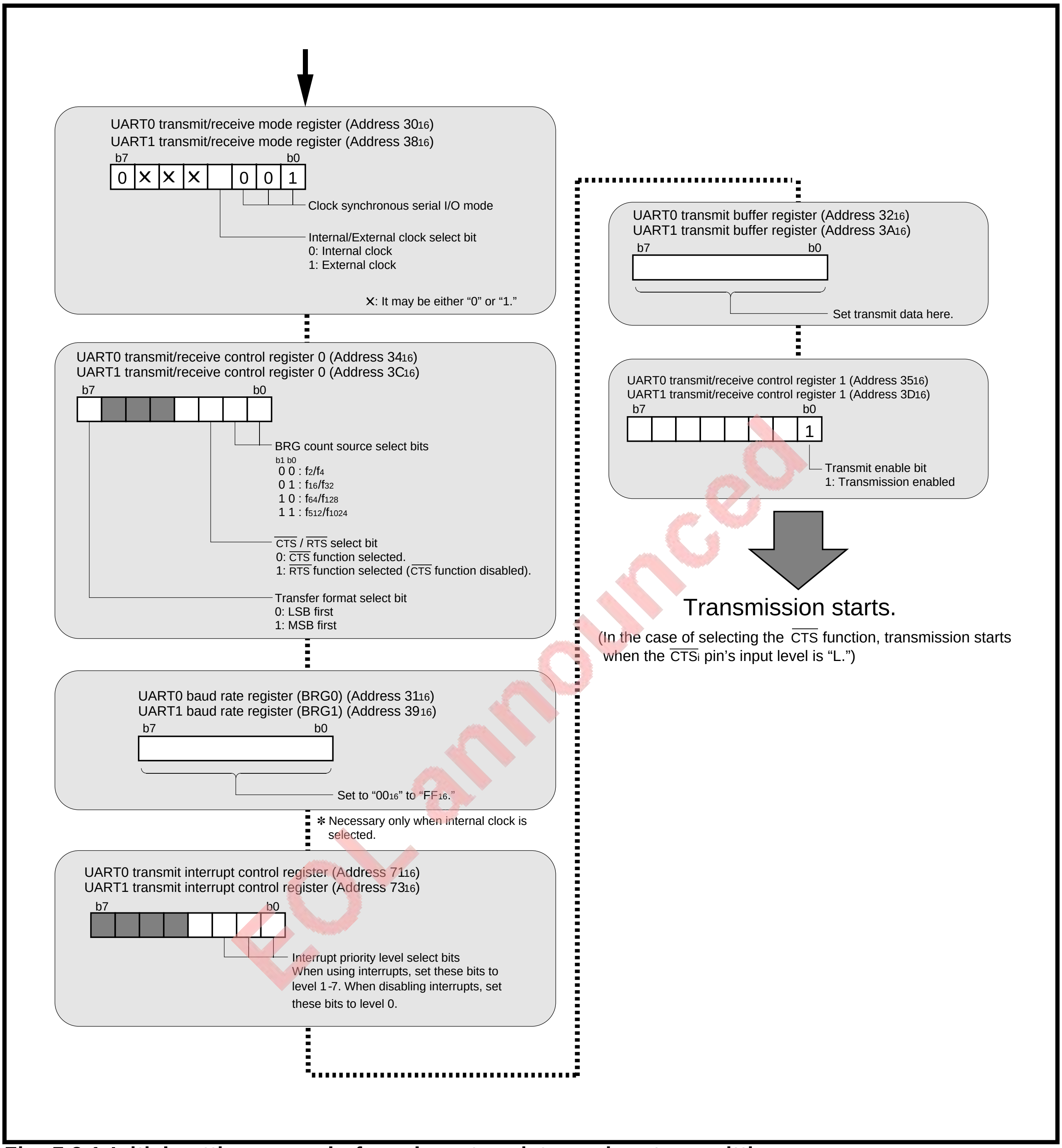


Fig. 7.3.1 Initial setting example for relevant registers when transmitting

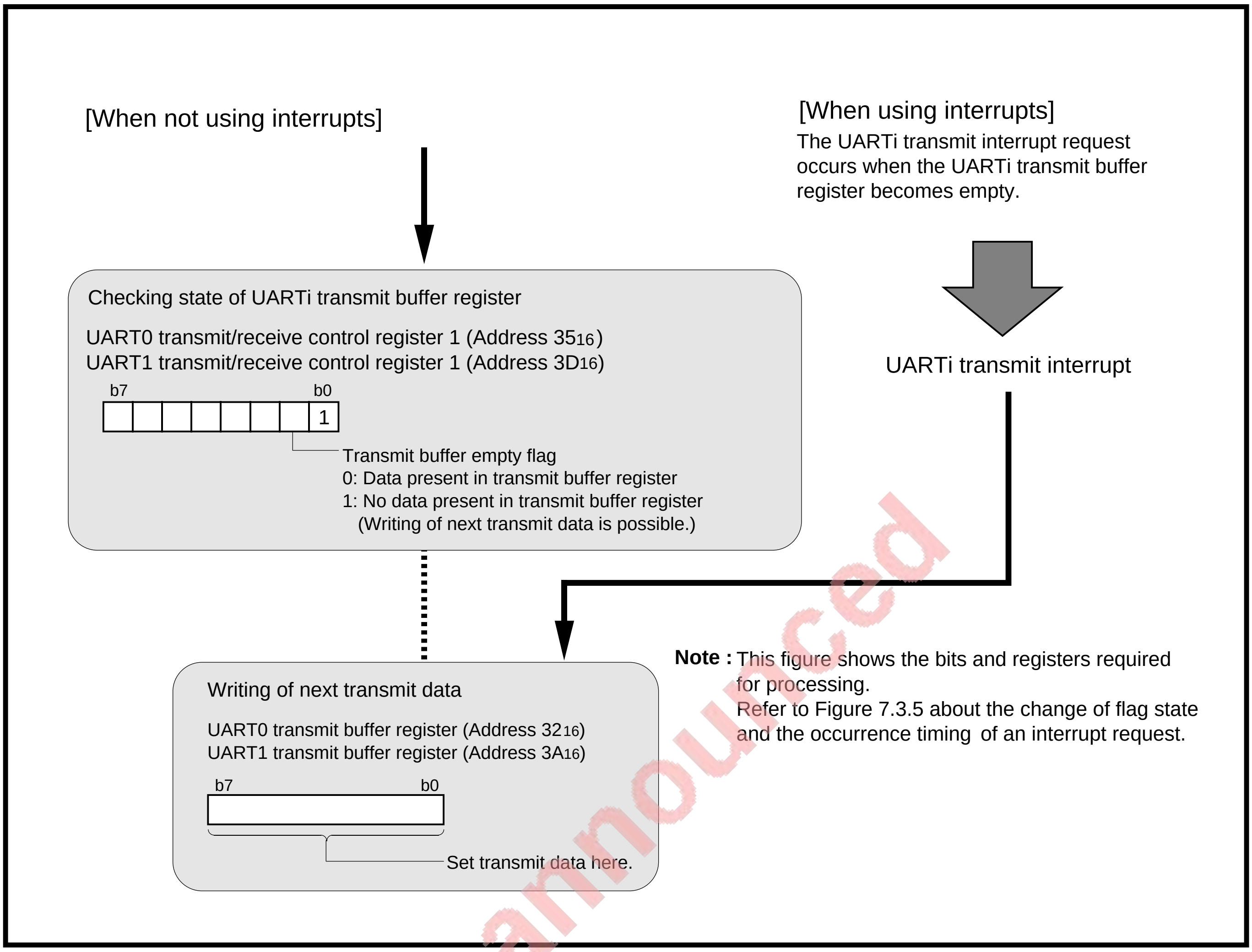


Fig. 7.3.2 Writing data after start of transmission

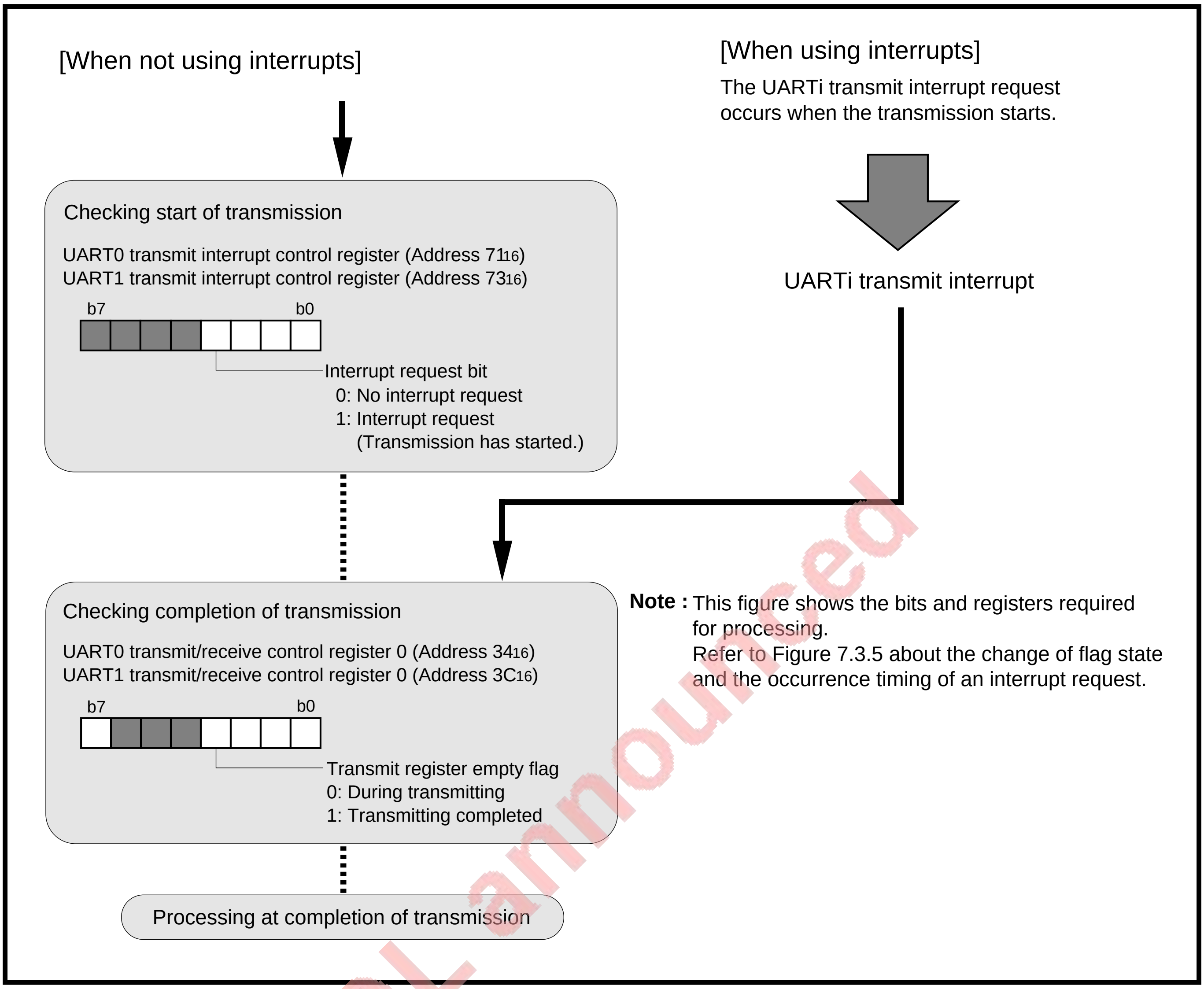


Fig. 7.3.3 Detection of transmission's completion

7.3.4 Transmit operation

When the transmit conditions described in page 7-20 are satisfied, the following operations are automatically performed simultaneously.

- •The UARTi transmit buffer register's contents are transferred to the UARTi transmit register.
- •8 transfer clocks are generated (when an internal clock is selected).
- •The transmit buffer empty flag is set to "1."
- •The transmit register empty flag is cleared to "0."
- •The UARTi transmit interrupt request occurs, and the interrupt request bit is set to "1."

The transmit operations are described below.

① Data in the UARTi transmit register is transmitted from the TxD_i pin synchronously with the falling of the transfer clock.
② This data is transmitted bit by bit sequentially beginning with the least significant bit.
③ When 1-byte data has been transmitted, the transmit register empty flag is set to "1," indicating completion of the transmission.

Figure 7.3.4 shows the transmit operation.
In the case of an internal clock is selected, when the transmit conditions for the next data are satisfied at completion of the transmission, the transfer clock is generated continuously. Accordingly, when performing transmission continuously, set the next transmit data to the UARTi transmit buffer register during transmission (when the transmit register empty flag = "0"). When the transmit conditions for the next data are not satisfied, the transfer clock stops at "H" level.
Figures 7.3.5 shows an example of transmit timing (when selecting an internal clock).

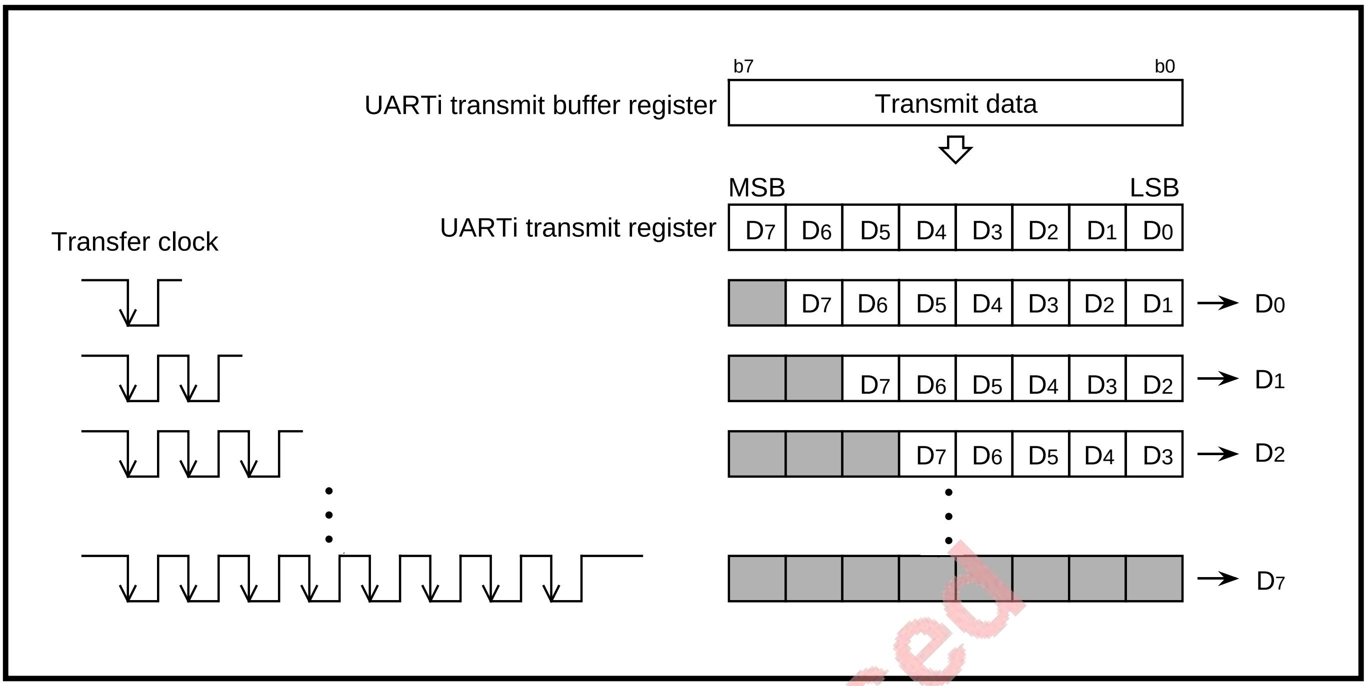


Fig. 7.3.4 Transmit operation

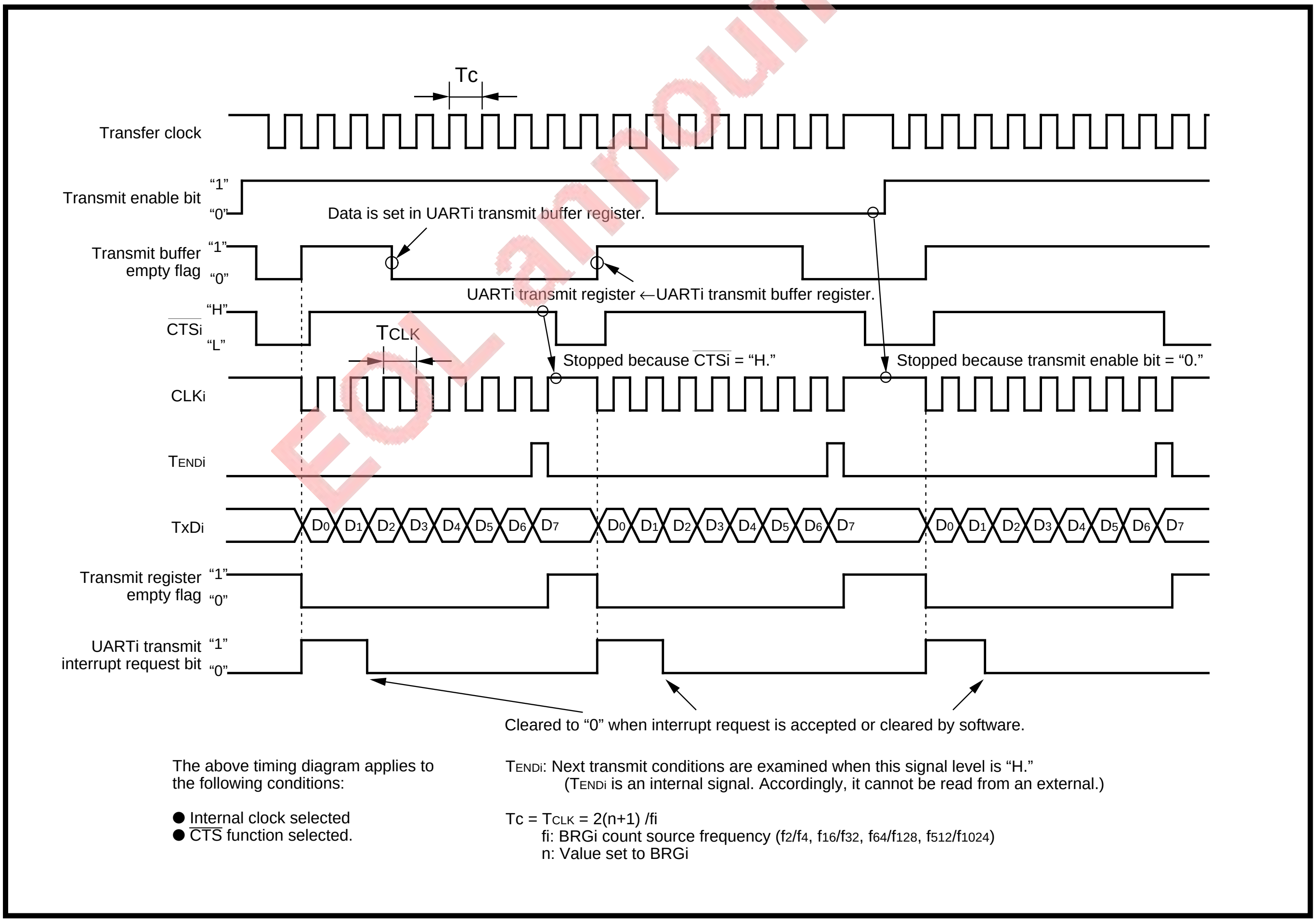


Fig. 7.3.5 Example of transmit timing (when selecting internal clock)

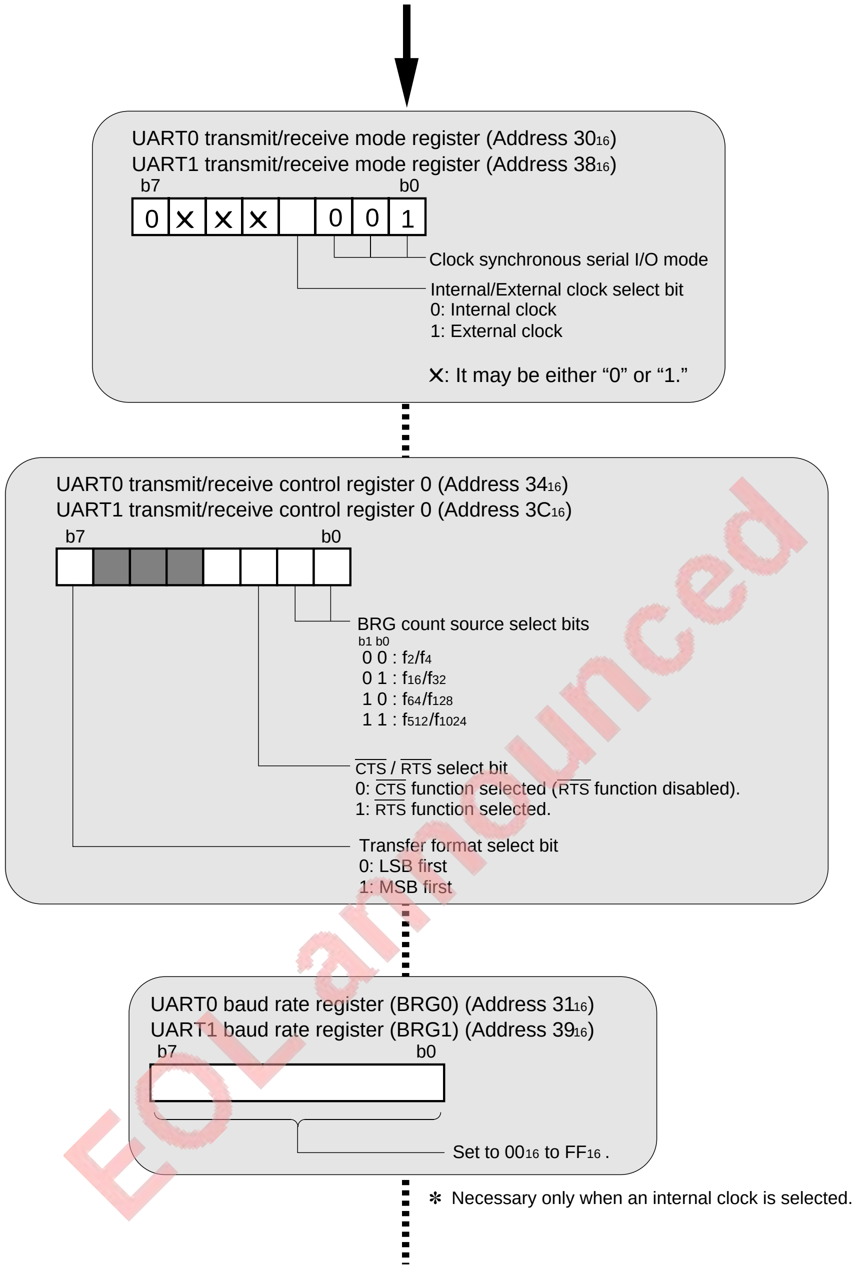
7.3.5 Method of reception

Figures 7.3.6 and 7.3.7 show initial setting examples for relevant registers when receiving. Reception is started when all of the following conditions (① to ③) are satisfied. When an external clock is selected, satisfy conditions ① to ③ with the following precondition satisfied.

<Precondition>
The CLKi pin's input is "H" level.
Note: When an internal clock is selected, above precondition is ignored.

<Reception conditions>
① Reception is enabled (receive enable bit = "1").
② Transmission is enabled (transmit enable bit = "1").
③ Dummy data is present in the UARTi transmit buffer register (transmit buffer empty flag = "0")

When using interrupts, it is necessary to set the relevant register to enable interrupts. For details, refer to **"Chapter 4. INTERRUPTS."**
Figure 7.3.8 shows processing after reception's completion.



Continued to **Figure 7.3.7** on next page.

Fig. 7.3.6 Initial setting example for relevant registers when receiving (1)

7.3 Clock synchronous serial I/O mode

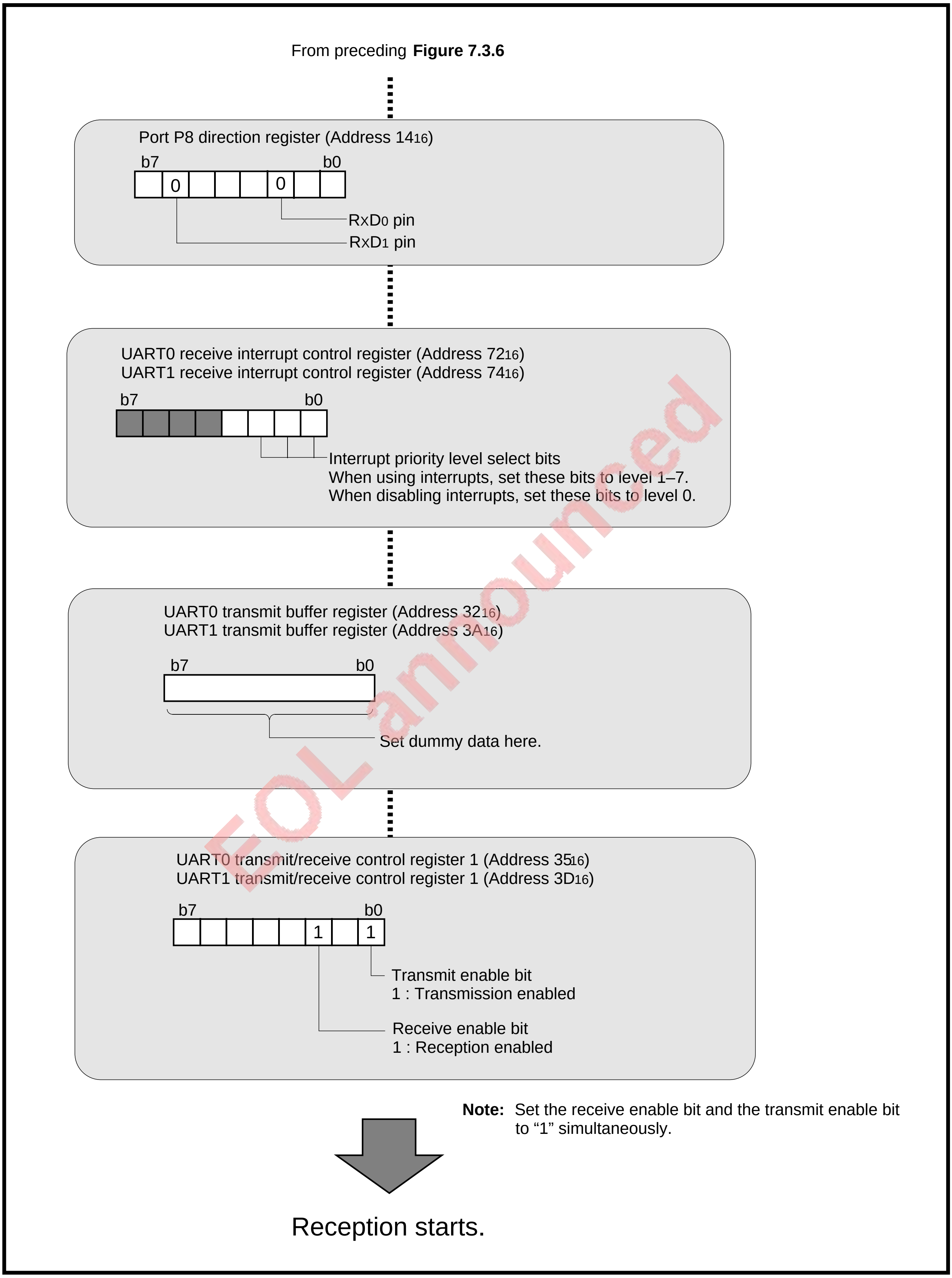
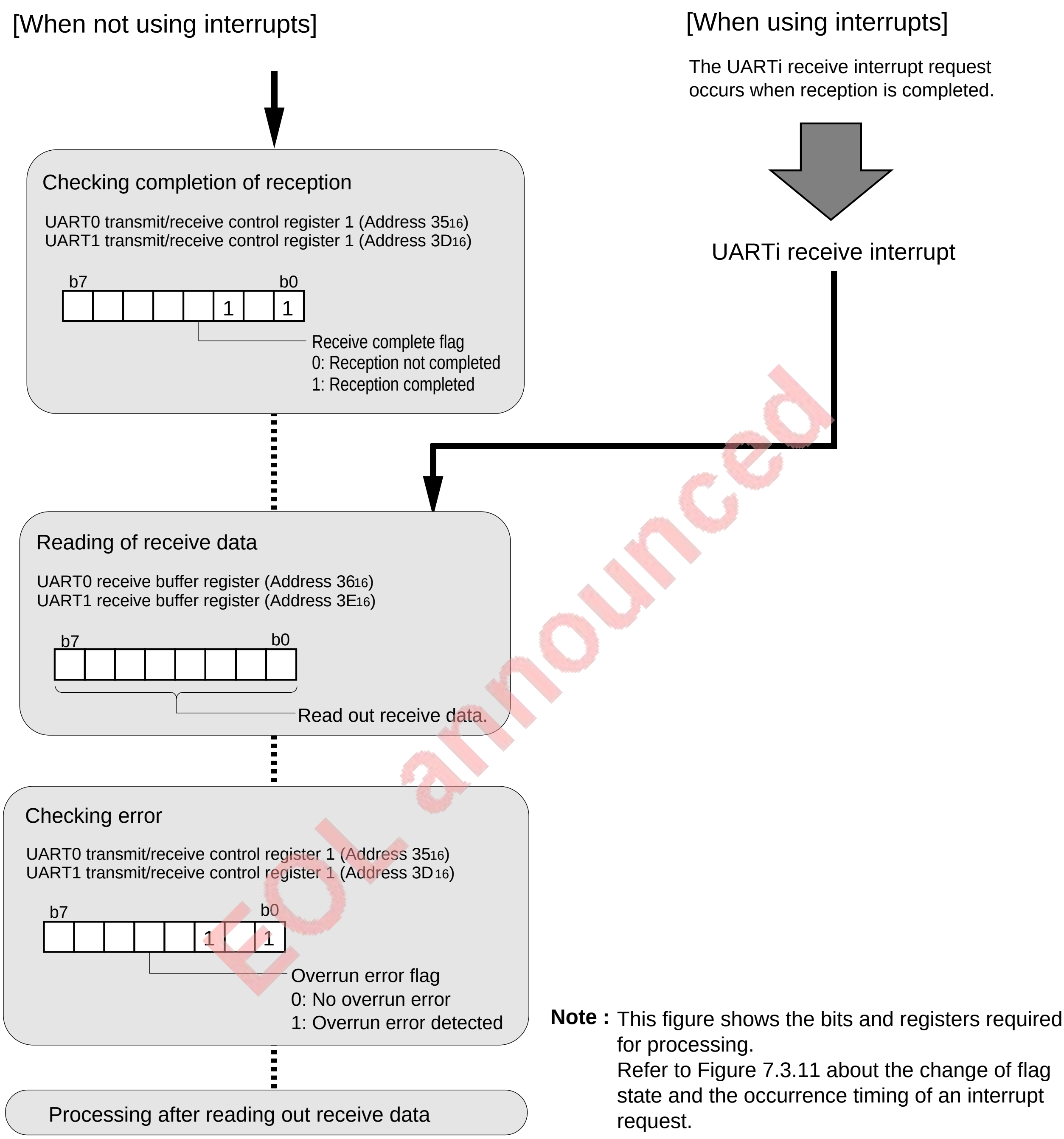


Fig. 7.3.7 Initial setting example for relevant registers when receiving (2)



Note : This figure shows the bits and registers required for processing.
Refer to Figure 7.3.11 about the change of flag state and the occurrence timing of an interrupt request.

Fig. 7.3.8 Processing after reception's completion

7.3.6 Receive operation

When the receive conditions listed on page 7-26 are satisfied, the UARTi enters the receive enable state.

The receive operations are described below.

① The input signal of the RxD_i pin is taken into the most significant bit of the UARTi receive register synchronously with the rising of the transfer clock.
② The contents of the UARTi receive register are shifted by 1 bit to the right.
③ Steps ① and ② are repeated at each rising of the transfer clock.
④ When 1-byte data is prepared in the UARTi receive register, the contents of this register are transferred to the UARTi receive buffer register.
⑤ Simultaneously with step ④, the receive complete flag is set to "1," and the UARTi receive interrupt request occurs and its interrupt request bit is set to "1."

The receive complete flag is cleared to "0" when the low-order byte of the UARTi receive buffer register is read out. Figure 7.3.10 shows the receive operation, and Figure 7.3.11 shows an example of receive timing (when selecting an external clock).
When the transfer format select bit = "1" (MSB first), each bit's position of this register's contents is reversed and the resultant data is read out.

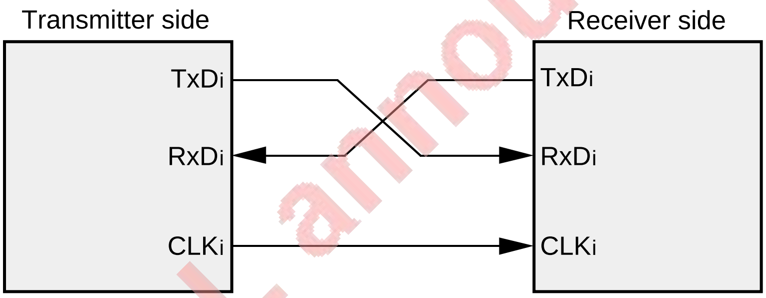


Fig. 7.3.9 Connection example

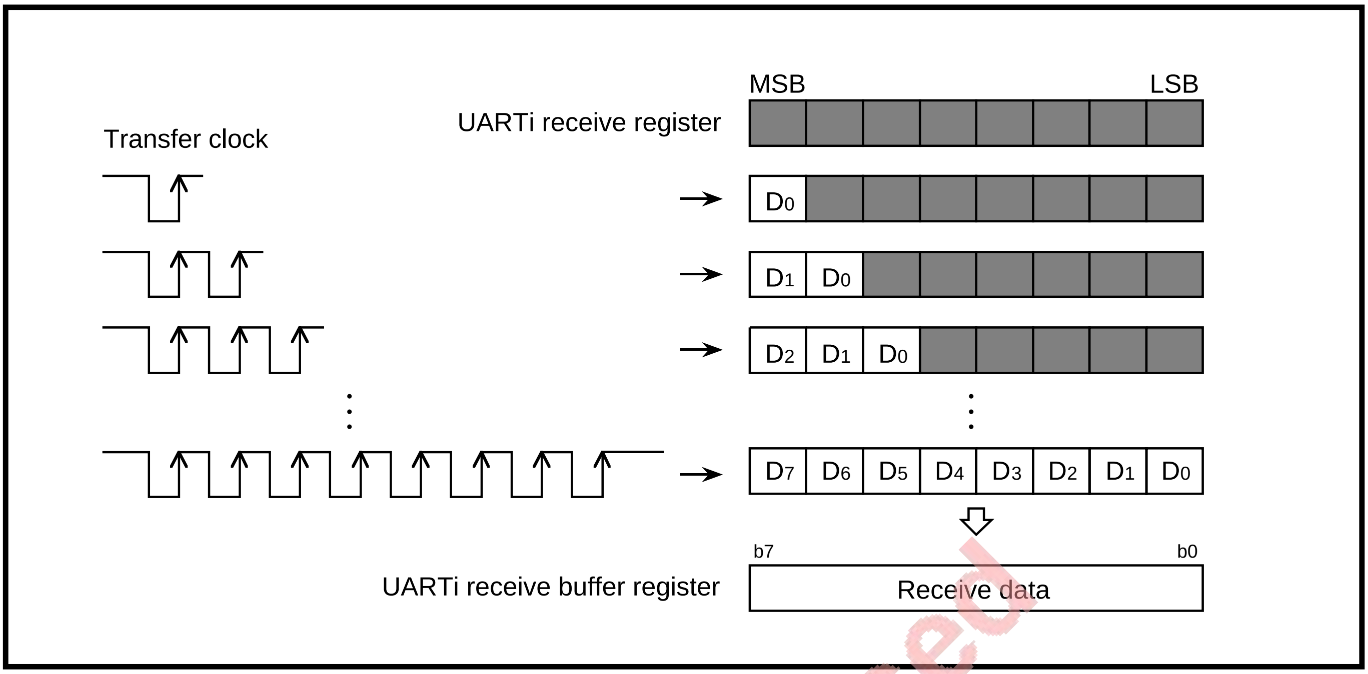
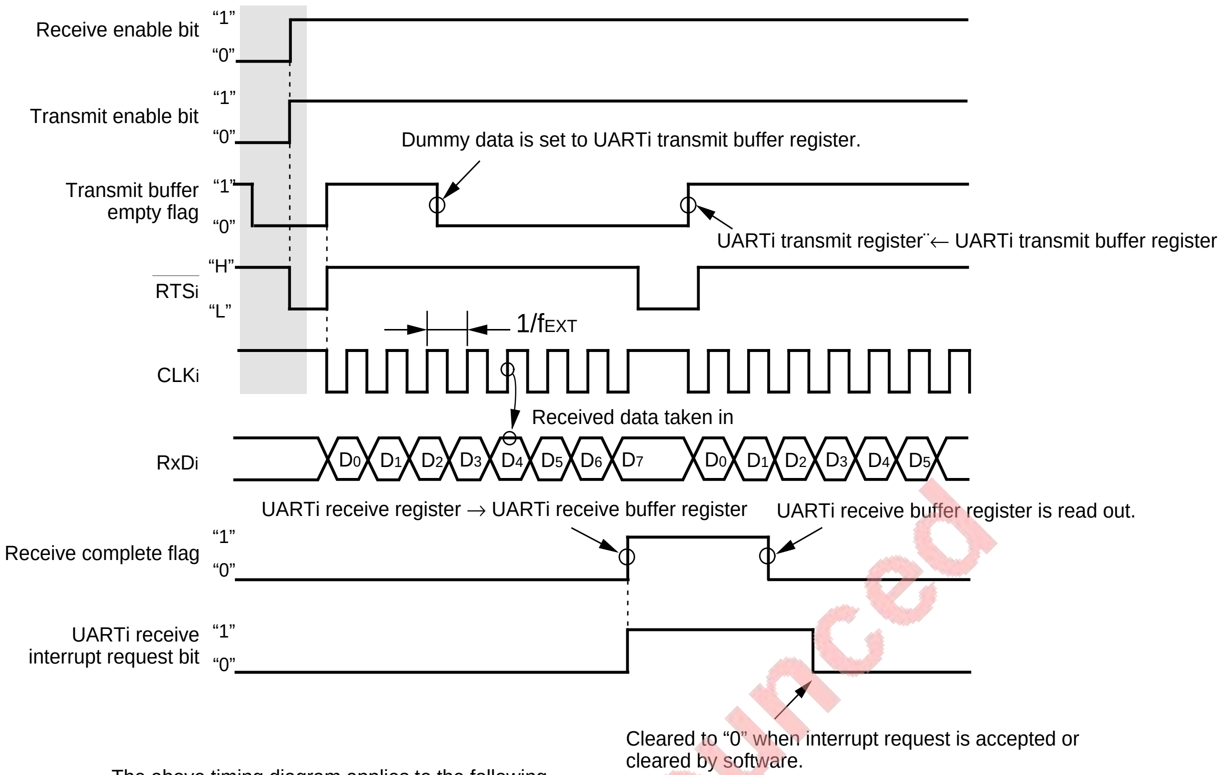


Fig. 7.3.10 Receive operation



The above timing diagram applies to the following setting conditions:
- External clock selected.
- RTS function selected.

f_{EXT}: Frequency of external clock

(Shaded area): When the CLKi pin's input level is "H," satisfy the following cinditions:
- Transmit enable bit → "1"
- Receive enable bit → "1"
- Writing of dummy data to UARTi transmit buffer register

Fig. 7.3.11 Example of receive timing (when selecting external clock)

7.3.7 Process on detecting overrun error

In the clock synchronous serial I/O mode, an overrun error can be detected.
An overrun error occurs when the next data is prepared in the UARTi receive register with the receive complete flag = “1” (data is present in the UARTi receive buffer register) and that is transferred to the receive buffer register, in other words, when the next data is prepared before reading out the contents of the UARTi receive buffer register. When an overrun error occurs, the next receive data is written into the UARTi receive buffer register, and the UARTi receive interrupt request bit is not changed.
An overrun error is detected when data is transferred from the UARTi receive register to the UARTi receive buffer register and the overrun error flag is set to “1.” The overrun error flag is cleared to “0” by clearing the receive enable bit to “0.”

When an overrun error occurs during reception, initialize the overrun error flag and the UARTi receive buffer register before performing reception again. When it is necessary to perform retransmission owing to an overrun error which occurs in the receiver side, set the UARTi transmit buffer register again before starting transmission again.
The method of initializing the UARTi receive buffer register and that of setting the UARTi transmit buffer register again are described below.

(1) Method of initializing UARTi receive buffer register

① Clear the receive enable bit to “0” (reception disabled).
② Set the receive enable bit to “1” again (reception enabled).

(2) Method of setting UARTi transmit buffer register again

① Clear the serial I/O mode select bits to “000_2” (serial I/O ignored).
② Set the serial I/O mode select bits to “001_2” again.
③ Set the transmit enable bit to “1” (transmission enabled), and set the transmit data to the UARTi transmit buffer register.

[Precautions when operating in clock synchronous serial I/O mode]

1. The transfer clock is generated by operation of the transmit control circuit. Accordingly, even when performing only reception, transmit operation (setting for transmission) must be performed. In this case, dummy data is output from the TxD_i pin.

2. When receiving, simultaneously set the receive enable bit and the transmit enable bit to “1.”

3. When selecting an external clock, satisfy the following 3 conditions with the input to CLK_i pin = “H” level.

   <When transmitting>
   ① Set the transmit enable bit to “1.”
   ② Write transmit data to the UARTi transmit buffer register.
   ③ Input “L” level to the $\overline{CTS_i}$ pin (when selecting the $\overline{CTS}$ function).

   <When receiving>
   ① Set the receive enable bit to “1.”
   ② Set the transmit enable bit to “1.”
   ③ Write dummy data to the UARTi transmit buffer register.

4. When receiving data, write dummy data to the low-order byte of the UARTi transmission buffer register for each reception of 1-byte data.

5. The output level of the $\overline{RTS_i}$ pin becomes “L” simultaneously at setting the receive enable bit to “1.” The output level of this pin becomes “H” when receive starts, and it becomes “L” when receive is completed. The output level of this pin changes regardless of the contents of the transmit enable bit, the transmission buffer empty flag, and the receive complete flag.

7.4 Clock asynchronous serial I/O (UART) mode

Table 7.4.1 lists the performance overview in the UART mode, and Table 7.4.2 lists the functions of I/O pins in this mode.

Table 7.4.1 Performance overview in UART mode

Item		Functions
Transfer data format	Start bit	1 bit
	Character bit (Transfer data)	7 bits, 8 bits, or 9 bits
	Parity bit	0 bit or 1 bit (Odd or even can be selected.)
	Stop bit	1 bit or 2 bits
Transfer rate	When selecting internal clock	Clock of BRGi output divided by 16
	When selecting external clock	Maximum 312.5 kbps
Error detection		4 types (Overrun, Framing, Parity, and Summing) Presence of error can be detected only by checking error sum flag.

Table 7.4.2 Functions of I/O pins in UART mode

Pin name	Functions	Method of selection
TxD_i ($P8_3$, $P8_7$)	Serial data output	Fixed
RxD_i ($P8_2$, $P8_6$)	Serial data input	Port P8 direction register*1's corresponding bit = "0"
CLK_i ($P8_1$, $P8_5$)	BRGi's count source input	Internal/External clock select bit*2 = "1"
$\overline{CTS_i}/\overline{RTS_i}$ ($P8_0$, $P8_4$)	$\overline{CTS}$ input	$\overline{CTS}/\overline{RTS}$ select bit*3 = "0"
	$\overline{RTS}$ output	$\overline{CTS}/\overline{RTS}$ select bit = "1"

Port P8 direction register*1: Address 14_{16}
Internal/External clock select bit*2: bit 3 at addresses 30_{16}, 38_{16}
$\overline{CTS}/\overline{RTS}$ select bit*3: bit 2 at addresses 34_{16}, $3C_{16}$

Notes 1: The TxD_i pin outputs "H" level while not transmitting after selecting UARTi's operating mode.
2: The RxD_i pin can be used as a programmable I/O port when performing only transmission.
3: The CLK_i pin can be used as a programmable I/O port when selecting internal clock.
4: The $\overline{CTS_i}/\overline{RTS_i}$ pin can be used as a input port when performing only reception and not using $\overline{RTS}$ function (when selecting $\overline{CTS}$ function).

7.4 Clock asynchronous serial I/O (UART) mode

7.4.1 Transfer rate (frequency of transfer clock)

The transfer rate is determined by the BRGi (addresses 31_{16}, 39_{16}).
When setting "n" into BRGi (n = "00_{16}" to "FF_{16}"), BRGi divides the count source frequency by n + 1. The divided clock by BRGi is further divided by 16 and the resultant clock becomes the transfer clock. Accordingly, the value "n" is expressed by the following formula.

$$n = \frac{F}{16 \times B} - 1$$

n: Value set into BRGi (00_{16} to FF_{16})
F: BRGi's count source frequency
B: Transfer rate (bps)

An internal clock or an external clock can be selected as the BRGi's count source with the internal/external clock select bit (bit 3 at addresses 30_{16}, 38_{16}). When an internal clock is selected, the clock selected with the BRG count source select bits (bits 0 and 1 at addresses 34_{16}, $3C_{16}$) becomes the BRGi's count source. When an external clock is selected, the clock input to the CLK_i pin becomes the BRGi's count source.
Tables 7.4.3 to 7.4.5 are list the setting examples of transfer rate. Set the same transfer rate between the transmitter and the receiver.

Table 7.4.3 Setting examples of transfer rate (1)

Transfer rate (bps)	f(X_{IN}) = 25 MHz					
	Clock source for peripheral devices select bit = "1"			Clock source for peripheral devices select bit = "0"		
	BRGi count source	BRGi setting value : n	Actual time (bps)	BRGi count source	BRGi setting value : n	Actual time (bps)
150	f_{64}	162 ($A2_{16}$)	149.78	f_{128}	80 (50_{16})	150.70
300	f_{64}	80 (50_{16})	301.41	f_{32}	162 ($A2_{16}$)	299.56
600	f_{16}	162 ($A2_{16}$)	599.12	f_{32}	80 (50_{16})	602.82
1200	f_{16}	80 (50_{16})	1205.63	f_{32}	40 (28_{16})	1190.93
2400	f_{16}	40 (28_{16})	2381.86	f_4	162 ($A2_{16}$)	2396.47
4800	f_2	162 ($A2_{16}$)	4792.94	f_4	80 (50_{16})	4822.53
9600	f_2	80 (50_{16})	9645.06	f_4	40 (28_{16})	9527.44
19200	f_2	40 (28_{16})	19054.88			
31250	f_2	24 (18_{16})	31250.00			

Clock source for peripheral devices select bit : bit 2 at address $5F_{16}$

Table 7.4.4 Setting examples of transfer rate (2)

Transfer rate (bps)	f(X_{IN}) = 24.576 MHz					
	Clock source for peripheral devices select bit = "1"			Clock source for peripheral devices select bit = "0"		
	BRGi count source	BRGi setting value : n	Actual time (bps)	BRGi count source	BRGi setting value : n	Actual time (bps)
150	f_{64}	159 ($9F_{16}$)	150.00	f_{128}	79 ($4F_{16}$)	150.00
300	f_{64}	79 ($4F_{16}$)	300.00	f_{32}	159 ($9F_{16}$)	300.00
600	f_{16}	159 ($9F_{16}$)	600.00	f_{32}	79 ($4F_{16}$)	600.00
1200	f_{16}	79 ($4F_{16}$)	1200.00	f_{32}	39 (27_{16})	1200.00
2400	f_{16}	39 (27_{16})	2400.00	f_{4}	159 ($9F_{16}$)	2400.00
4800	f_{2}	159 ($9F_{16}$)	4800.00	f_{4}	79 ($4F_{16}$)	4800.00
9600	f_{2}	79 ($4F_{16}$)	9600.00	f_{4}	39 (27_{16})	9600.00
19200	f_{2}	39 (27_{16})	19200.00	f_{4}	19 (13_{16})	19200.00
31250						

Clock source for peripheral devices select bit : bit 2 at address $5F_{16}$

Table 7.4.5 Setting examples of transfer rate (3)

Transfer rate (bps)	Clock source for peripheral devices select bit = "0"					
	f(X_{IN}) = 39.3216 MHz			f(X_{IN}) = 40 MHz		
	BRGi count source	BRGi setting value : n	Actual time (bps)	BRGi count source	BRGi setting value : n	Actual time (bps)
150	f_{128}	127 ($7F_{16}$)	150.00	f_{128}	129 (81_{16})	150.24
300	f_{32}	255 (FF_{16})	300.00	f_{128}	64 (40_{16})	300.48
600	f_{32}	127 ($7F_{16}$)	600.00	f_{32}	129 (81_{16})	600.96
1200	f_{32}	63 ($3F_{16}$)	1200.00	f_{32}	64 (40_{16})	1201.92
2400	f_{4}	255 (FF_{16})	2400.00	f_{32}	32 (20_{16})	2367.42
4800	f_{4}	127 ($7F_{16}$)	4800.00	f_{4}	129 (81_{16})	4807.69
9600	f_{4}	63 ($3F_{16}$)	9600.00	f_{4}	64 (40_{16})	9615.38
19200	f_{4}	31 ($1F_{16}$)	19200.00	f_{4}	32 (20_{16})	18939.39
31250				f_{4}	19 (13_{16})	31250.00

Clock source for peripheral devices select bit : bit 2 at address $5F_{16}$

7.4.2 Transfer data format

The transfer data format can be selected from formats shown in Figure 7.4.1. Bits 4 to 6 at addresses 30_{16} and 38_{16} select the transfer data format. (Refer to Figure 7.1.1.) Set the same transfer data format for both transmitter and receiver sides.

Figure 7.4.2 shows an example of transfer data format. Table 7.4.6 lists each bit in transmit data.

Transfer data length of 7 bits
- 1ST—7DATA——1SP
- 1ST—7DATA——2SP
- 1ST—7DATA—1PAR—1SP
- 1ST—7DATA—1PAR—2SP

Transfer data length of 8 bits
- 1ST—8DATA——1SP
- 1ST—8DATA——2SP
- 1ST—8DATA—1PAR—1SP
- 1ST—8DATA—1PAR—2SP

Transfer data length of 9 bits
- 1ST—9DATA——1SP
- 1ST—9DATA——2SP
- 1ST—9DATA—1PAR—1SP
- 1ST—9DATA—1PAR—2SP

ST : Start bit
DATA : Character bit (transfer data)
PAR : Parity bit
SP : Stop bit

Fig. 7.4.1 Transfer data format

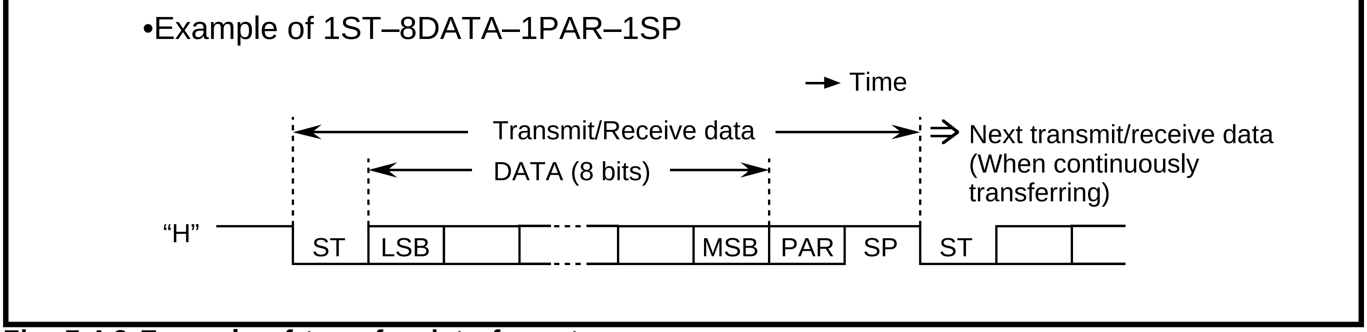


Fig. 7.4.2 Example of transfer data format

Table 7.4.6 Each bit in transmit data

Name	Functions
ST Start bit	"L" signal equivalent to 1 character bit which is added immediately before the character bits. It indicates start of data transmission.
DATA Character bit	Transmit data which is set in the UARTi transmit buffer register.
PAR Parity bit	A signal that is added immediately after the character bits in order to improve data reliability. The level of this signal changes according to selection of odd/even parity in such a way that the sum of "1"s in this bit and character bits is always an odd or even number.
ST Stop bit	"H" level signal equivalent to 1 or 2 character bits which is added immediately after the character bits (or parity bit when parity is enabled). It indicates finish of data transmission.

7.4.3 Method of transmission

Figure 7.4.3 shows an initial setting example for relevant registers when transmitting.
The difference due to selection of transfer data length (7 bits, 8 bits, or 9 bits) is only that data length. When selecting a 7- or 8-bit data length, set the transmit data into the low-order byte of the UARTi transmit buffer register. When selecting a 9-bit data length, set the transmit data into that low-order byte and bit 0 of that high-order byte.
Transmission is started when all of the following conditions (① to ③) are satisfied:

① Transmit is enabled (transmit enable bit = “1”).
② Transmit data is present in the UARTi transmit buffer register (transmit buffer empty flag = “0”).
③ $\overline{\text{CTS}}$i pin’s input is “L” level (when $\overline{\text{CTS}}$ function selected).
Note: When the $\overline{\text{CTS}}$ function is not selected, this condition is ignored.

When using interrupts, it is necessary to set the corresponding register to enable interrupts. For details, refer to **“Chapter 4. INTERRUPTS.”**
Figure 7.4.4 shows writing data after start of transmission, and Figure 7.4.5 shows detection of transmission’s completion.

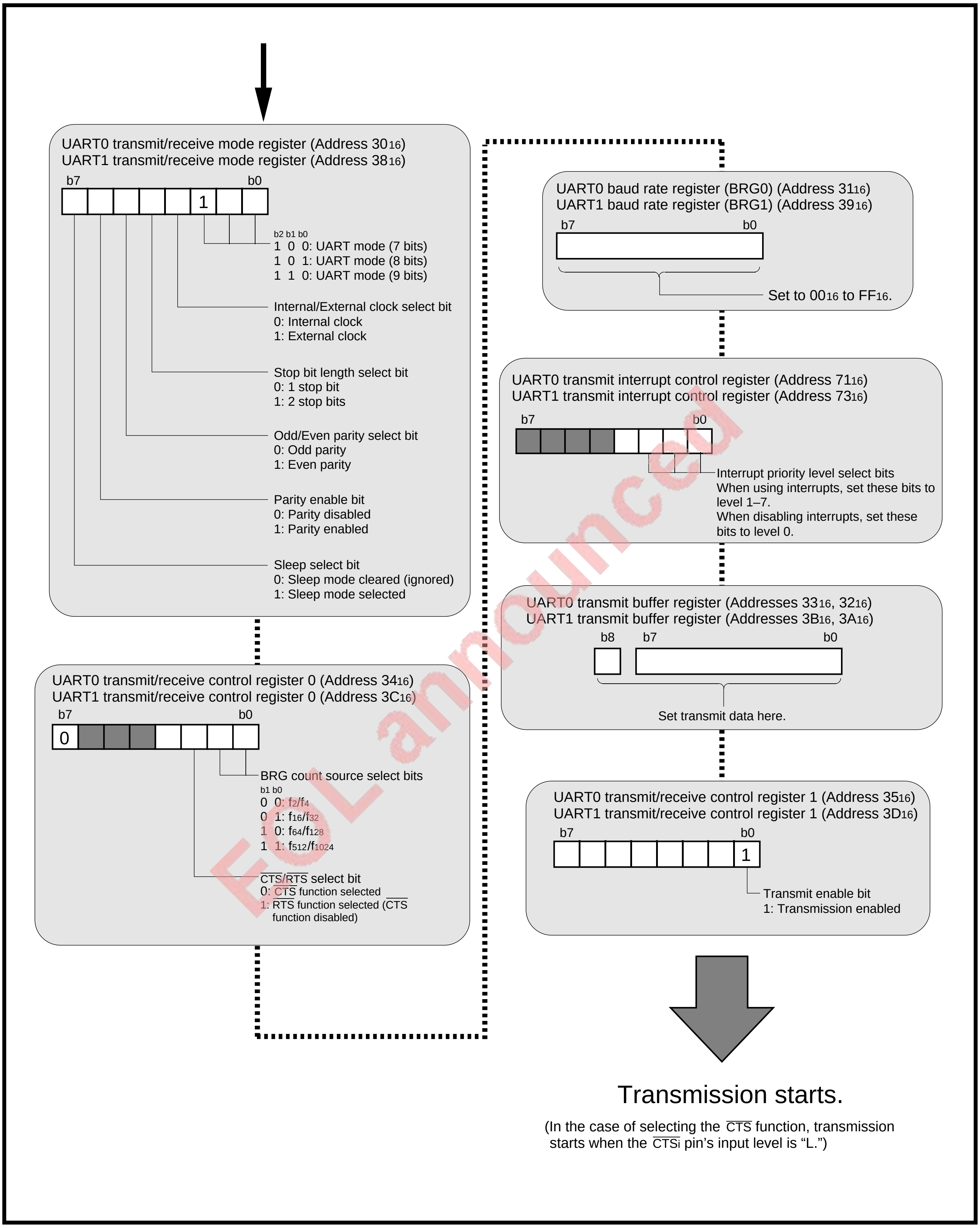


Fig. 7.4.3 Initial setting example for relevant registers when transmitting

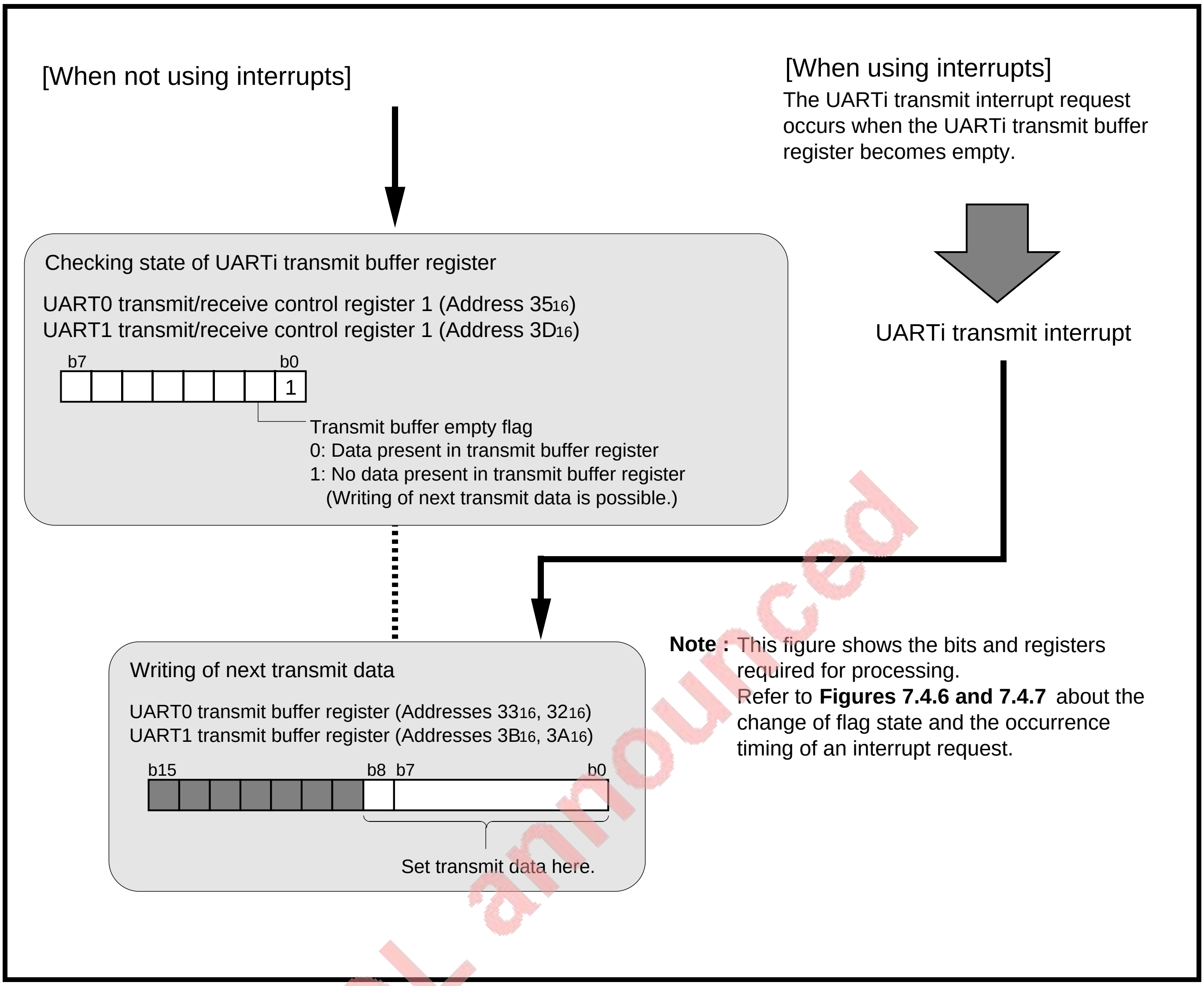


Fig. 7.4.4 Writing data after start of transmission

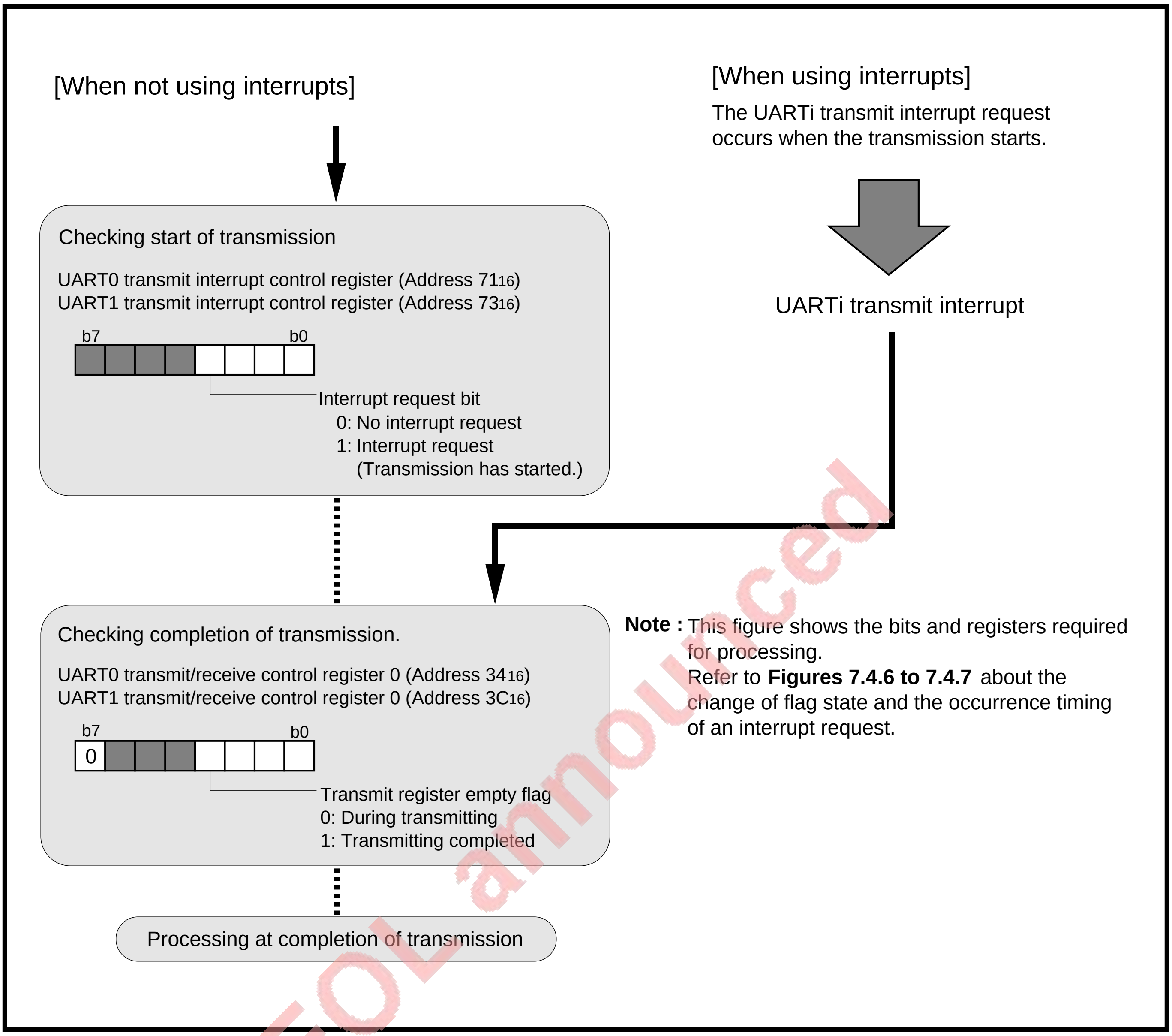


Fig. 7.4.5 Detection of transmission's completion

7.4.4 Transmit operation

Simultaneously when the transmit conditions listed on page 7-40 are satisfied, the following operations are automatically performed.

•The UARTi transmit buffer register's contents are transferred to the UARTi transmit register.
•The transmit buffer empty flag is set to "1."
•The transmit register empty flag is cleared to "0."
•The UARTi transmit interrupt request occurs and the interrupt request bit is set to "1."

The transmit operations are described below.

① Data in the UARTi transmit register is transmitted from the TxD_i pin.
② This data is transmitted bit by bit sequentially in order of ST→DATA (LSB)→•••→DATA (MSB)→PAR →SP according to the set transfer data format.
③ When the stop bit has been transmitted, the transmission register empty flag is set to "1," indicating completion of transmission.

When the transmit conditions for the next data are satisfied at completion of transmission, the start bit is generated following the stop bit, and the next data is transmitted. When performing transmission continuously, set the next transmit data in the UARTi transmit buffer register during transmission (when the transmit register empty flag = "0"). When the transmit conditions for the next data are not satisfied, the TxD_i pin outputs "H" level.
Figures 7.4.6 shows example of transmit timing when the transfer data length is 8 bits, and Figure 7.4.7 shows an example of transmit timing when the transfer data length is 9 bits.

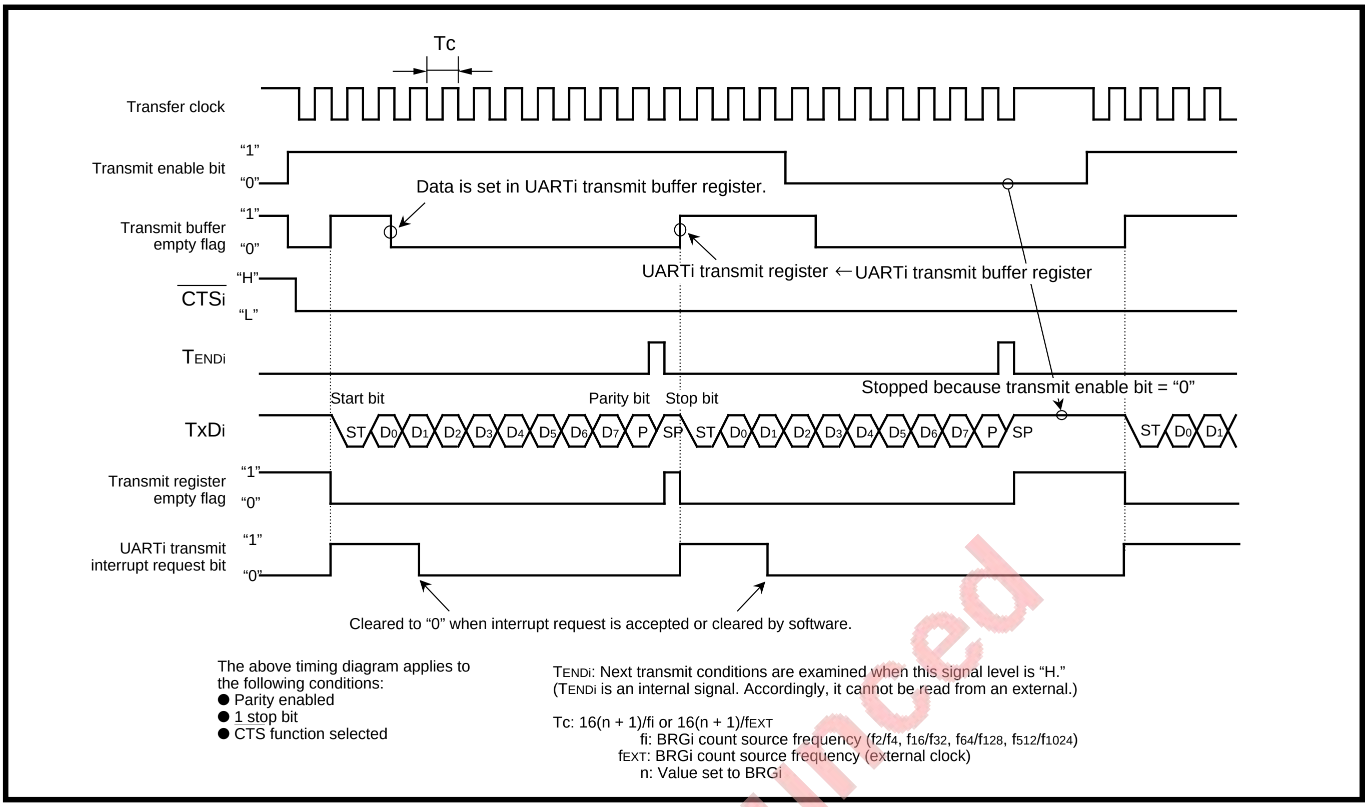


Fig. 7.4.6 Example of transmit timing when transfer data length is 8 bits (when parity enabled, selecting 1 stop bit)

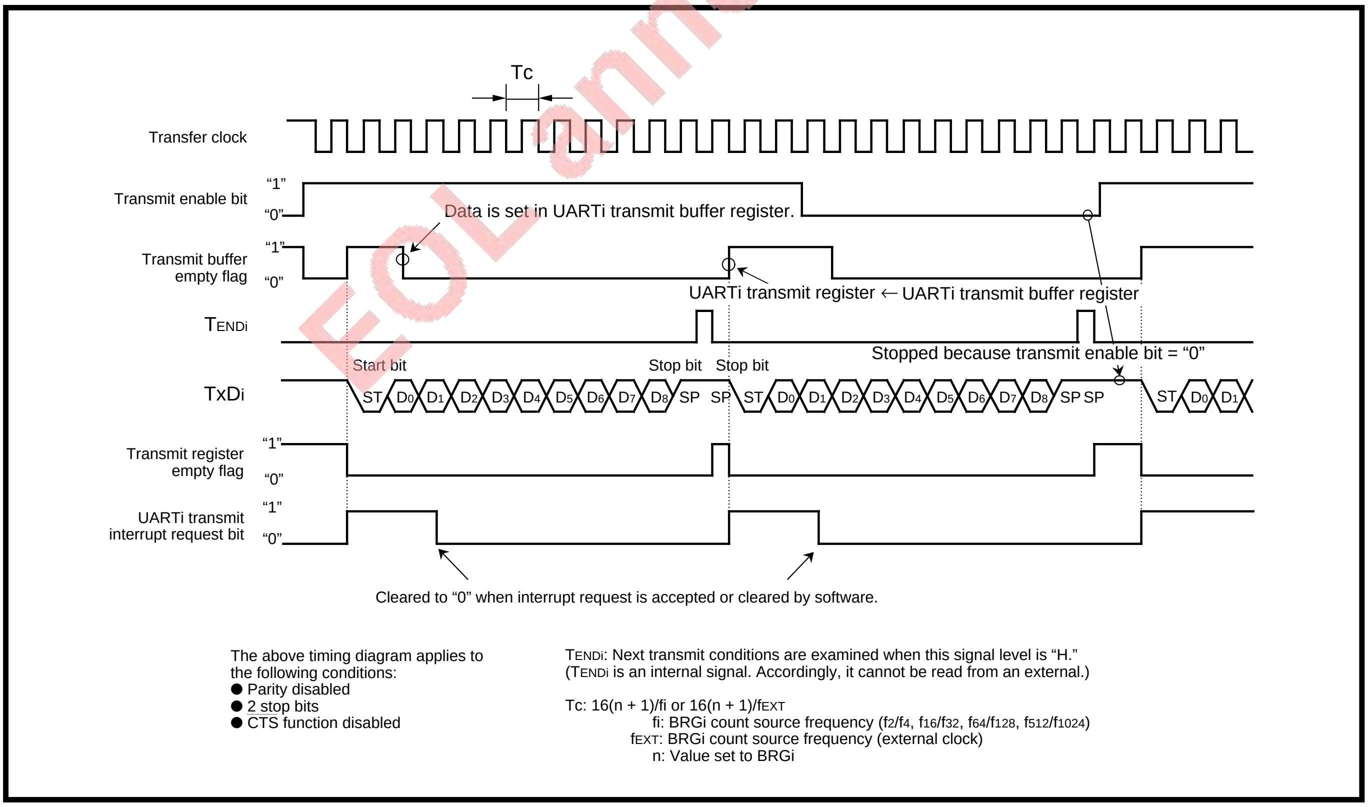


Fig. 7.4.7 Example of transmit timing when transfer data length is 9 bits (when parity disabled, selecting 2 stop bits)

7.4.5 Method of reception

Figure 7.4.8 shows an initial setting example for relevant registers when receiving. Reception is started when all of the following conditions (① and ②) are satisfied:

① Reception is enabled (receive enable bit = “1”).
② The start bit is detected.

When using interrupts, it is necessary to set the corresponding register to enable interrupts. For details, refer to **“Chapter 4. INTERRUPTS.”**
Figure 7.4.9 shows processing after reception’s completion.

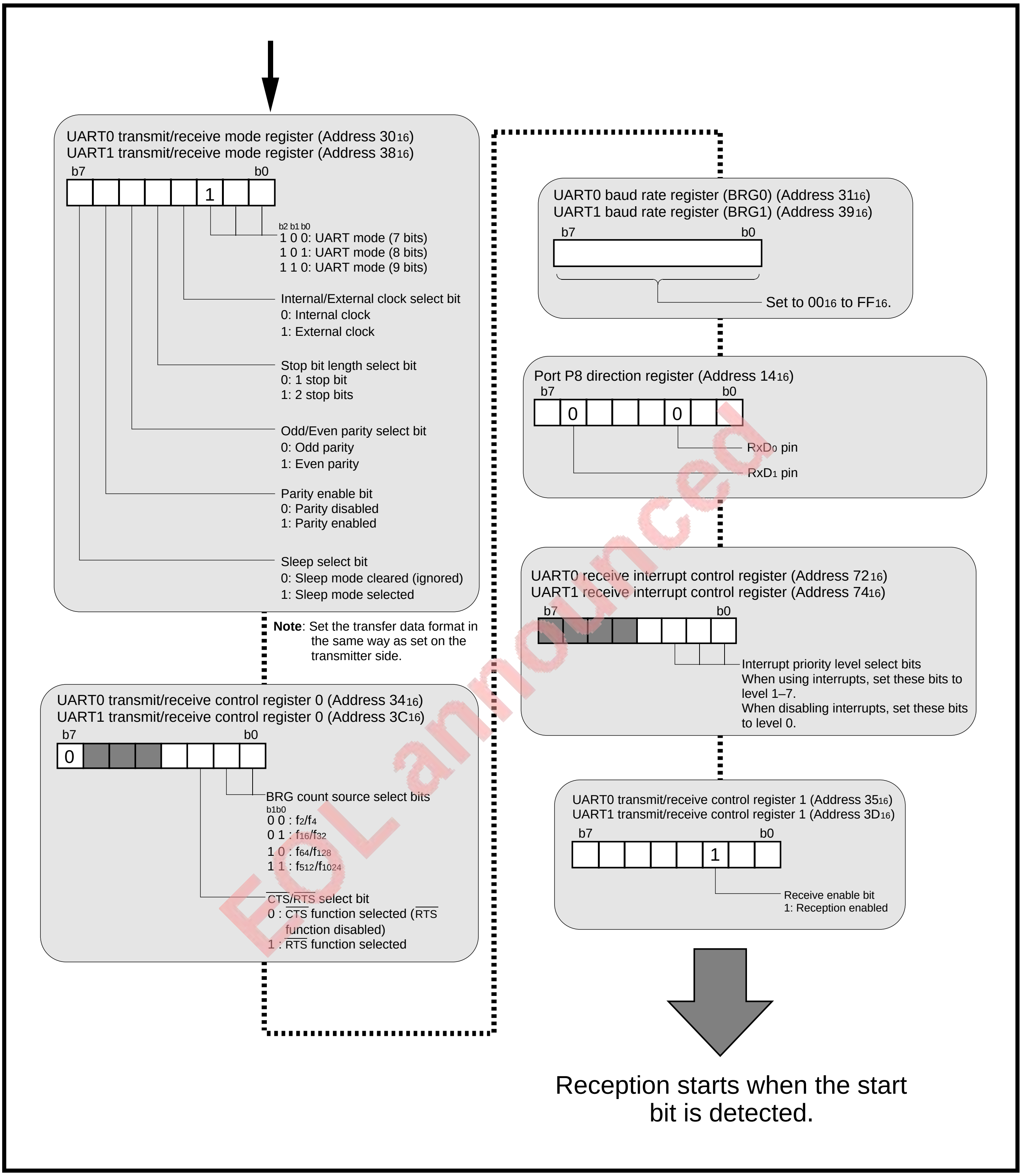


Fig. 7.4.8 Initial setting example for relevant registers when receiving

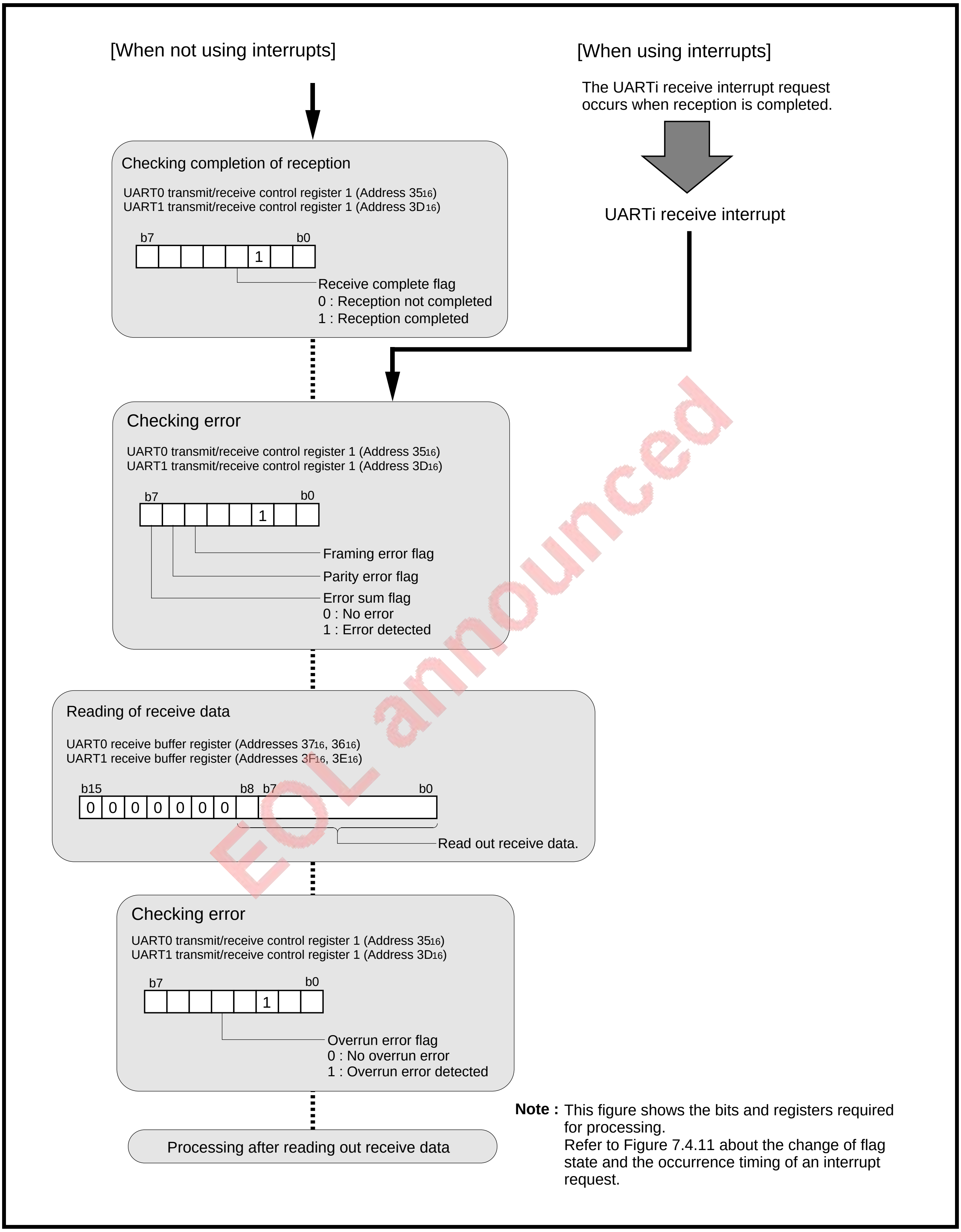


Fig. 7.4.9 Processing after reception's completion

7.4.6 Receive operation

When the receive enable bit is set to “1,” the UARTi enters the reception enabled state and reception starts at detecting ST. The receive operation is described below.

① The input signal of the RxD_i pin is taken into the most significant bit of the UARTi receive register synchronously with the transfer clock’s rising.
② The contents of UARTi receive register are shifted by 1 bit to the right.
③ Steps ① and ② are repeated at each rising of the transfer clock.
④ When one set of data has been prepared, in other words, the shift according to the selected data format has been completed; the UARTi receive register’s contents are transferred to the UARTi receive buffer register.
⑤ Simultaneously with step ④, the receive complete flag is set to “1,” and the UARTi receive interrupt request occurs and its interrupt request bit is set to “1.”

The receive complete flag is cleared to “0” when the low-order byte of the UARTi receive buffer register is read out. Figure 7.4.11 shows an example of receive timing when the transfer data length is 8 bits.

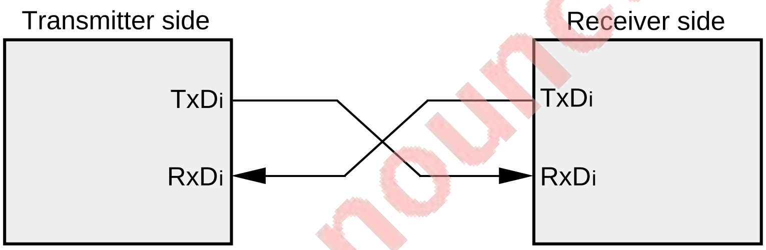


Fig. 7.4.10 Connection example

7.4 Clock asynchronous serial I/O (UART) mode

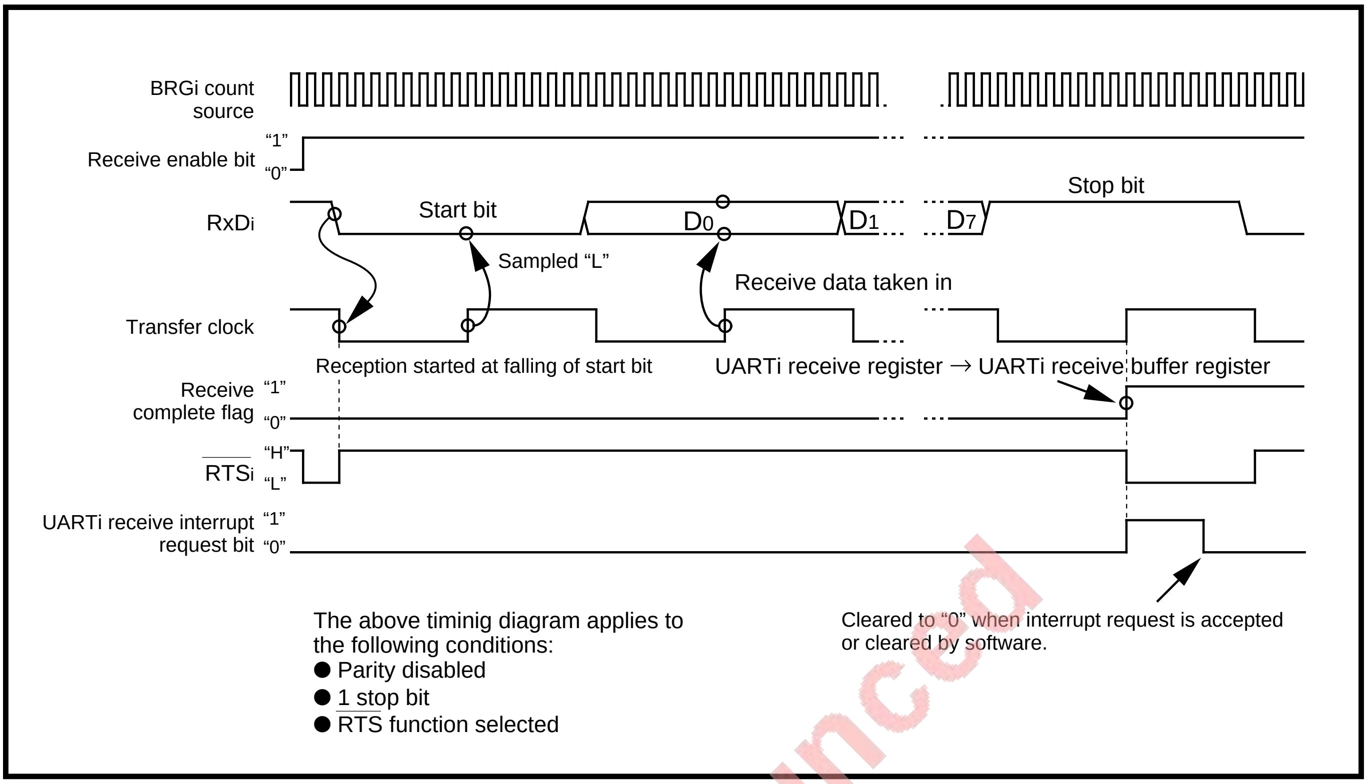


Fig. 7.4.11 Example of receive timing when transfer data length is 8 bits (when parity disabled, selecting 1 stop bit)

7.4.7 Process on detecting error

Errors listed below can be detected in the UART mode:

● Overrun error

An overrun error occurs when the next data is prepared in the UARTi receive register with the receive completion flag = "1" (that is, data present in the UARTi receive buffer register) and that data is transferred to the UARTi receive buffer register. In other words, when the next data is prepared before the contents of the UARTi receive buffer register is read out, an overrun error occurs. When an overrun error occurs, the next receive data is written into the UARTi receive buffer register, and the UARTi receive interrupt request bit is not changed.

● Framing error

A framing error occurs when the number of detected stop bits does not match the number of stop bits set. (The UARTi interrupt request bit becomes "1.")

● Parity error

A parity error occurs when the sum of "1"s in the parity bit and character bits does not match the number of "1"s set. (The UARTi interrupt request bit becomes "1.")

Each error is detected when data is transferred from the UARTi receive register to the UARTi receive buffer register, and the corresponding error flag is set to "1." Furthermore, when any of the above errors occurs, the error sum flag is set to "1." Accordingly, the error sum flag informs the user whether any error has occurred or not.

The overrun error flag is cleared to "0" by clearing the receive enable bit to "0."

The framing error flag and the parity error flag are cleared to "0" by reading the contents of the UARTi receive buffer register low-order byte or clearing the receive enable bit to "0." The error sum flag is cleared to "0" by clearing the all error flags, which are overrun, framing, and parity.

When errors occur during reception, initialize the error flags and the UARTi receive buffer register, and then perform reception again. When it is necessary to perform retransmission owing to an error which occurs in the receiver side, set the UARTi transmit buffer register again, and then starts transmission again.

The method of initializing the UARTi receive buffer register and that of setting the UARTi transmit buffer register again are described below.

(1) Method of initializing UARTi receive buffer register

① Clear the receive enable bit to "0" (reception disabled).

② Set the receive enable bit to "1" again (reception enabled).

(2) Method of setting UARTi transmit buffer register again

① Clear the serial I/O mode select bits to "000_2" (serial I/O ignored).

② Set the serial I/O mode select bits again.

③ Set the transmit enable bit to "1" (transmission enabled), and set the transmit data to the UARTi transmit buffer register.

7.4.8 Sleep mode

This mode is used to transfer data between the specified microcomputers, which are connected by using UARTi. The sleep mode is selected by setting the sleep select bit (bit 7 at addresses 30_{16}, 38_{16}) to “1” when receiving.

In the sleep mode, receive operation is performed when the MSB (D_8 when the transfer data length is 9 bits, D_7 when it is 8 bits, D_6 when it is 7 bits) of the receive data is “1.” Receive operation is not performed when the MSB is “0.” (The UARTi receive register’s contents are not transferred to the UARTi receive buffer register. Additionally, the receive complete flag and error flags do not change and the UARTi receive interrupt request does not occur.)

The following shows an usage example of sleep mode when the transfer data length is 8 bits.

① Set the same transfer data format for the master and slave microcomputers. Select the sleep mode for the slave microcomputers.

② Transmit data, which has “1” in bit 7 and the address of the slave microcomputer with which communicates in bits 0 to 6, from the master microcomputer to all slave microcomputers.

③ All slave microcomputers receive data of step ②. (At this time, the UARTi receive interrupt request occurs.)

④ In all slave microcomputers, check in the interrupt routine whether bits 0 to 6 in the receive data match their addresses.

⑤ In the slave microcomputer of which address matches bits 0 to 6 in the receive data, clear the sleep mode. (Do not clear the sleep mode for the other slave microcomputers.)
By performing steps ② to ⑤, “specification of the microcomputer performing transfer” is realized.

⑥ Transmit data, which has “0” in bit 7, from the master microcomputer. (Only the microcomputer specified in steps ② to ⑤ can receive this data. The other microcomputers do not receive this data.)

⑦ By repeating step ⑥, transfer can be performed between the same microcomputers continuously. When communicating with another microcomputer, perform steps ② to ⑤ in order to specify the new slave microcomputer.

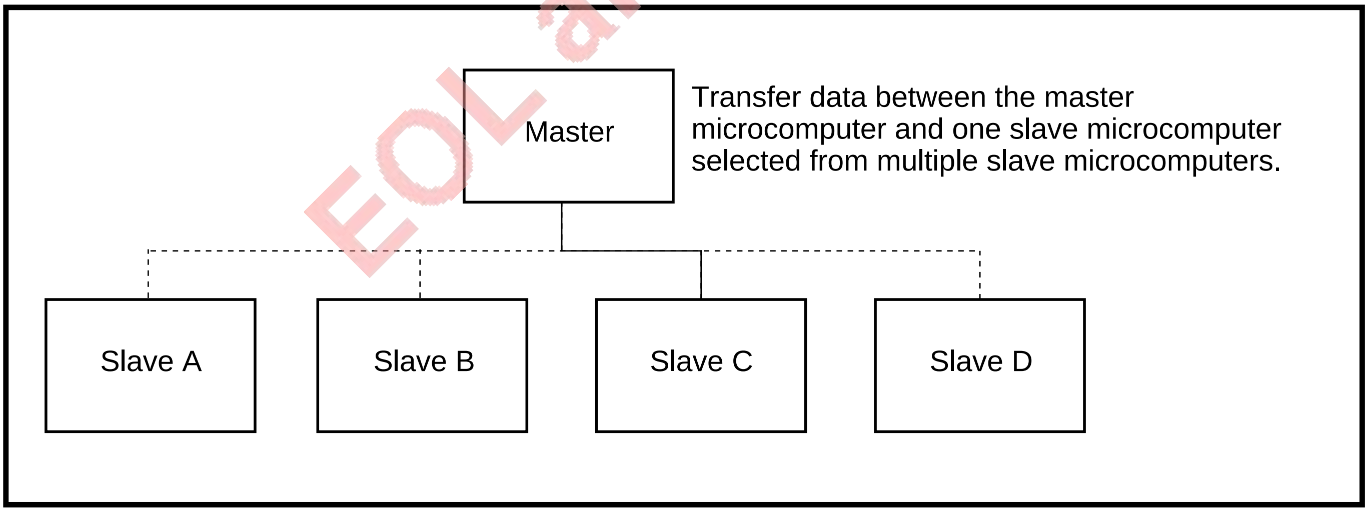


Fig. 7.4.12 Sleep mode

CHAPTER 8

A-D CONVERTER

8.1 Overview

The A-D converter has the performance specifications listed in Table 8.1.1.

Table 8.1.1 Performance specifications of A-D converter

Item	Performance specifications
A-D conversion method	Successive approximation conversion method
Resolution	Either 8 bits or 10 bits can be selected by software
Absolute accuracy	8-bit mode: ±2 LSB
	10-bit mode: ±3 LSB
Analog input pin	8 pins (AN_0 to AN_7)
Conversion rate per analog input pin	8-bit mode: 49 ϕ_{AD}* cycles
	10-bit mode: 59 ϕ_{AD}* cycles

ϕ_{AD}*: A-D converter's operation clock

The A-D converter has the 5 operation modes listed below.

- •One-shot mode
  This mode is used to perform the operation once for a voltage input from one selected analog input pin.
- •Repeat mode
  This mode is used to perform the operation repeatedly for a voltage input from one selected analog input pin.
- •Single sweep mode
  This mode is used to perform the operation for voltages input from multiple selected analog input pins, one at a time.
- •Repeat sweep mode 0
  This mode is used to perform the operation repeatedly for voltages input from multiple selected analog input pins.
- •Repeat sweep mode 1
  This mode is used to perform the operation repeatedly for voltages input from all analog input pins. In this mode, analog input pins are separated into two groups according to the frequency of use. One is the group for more frequencies of use, and the other is the group for fewer frequencies of use.

8.2 Block description

Figure 8.2.1 shows the block diagram of the A-D converter. Registers relevant to the A-D converter are described below.

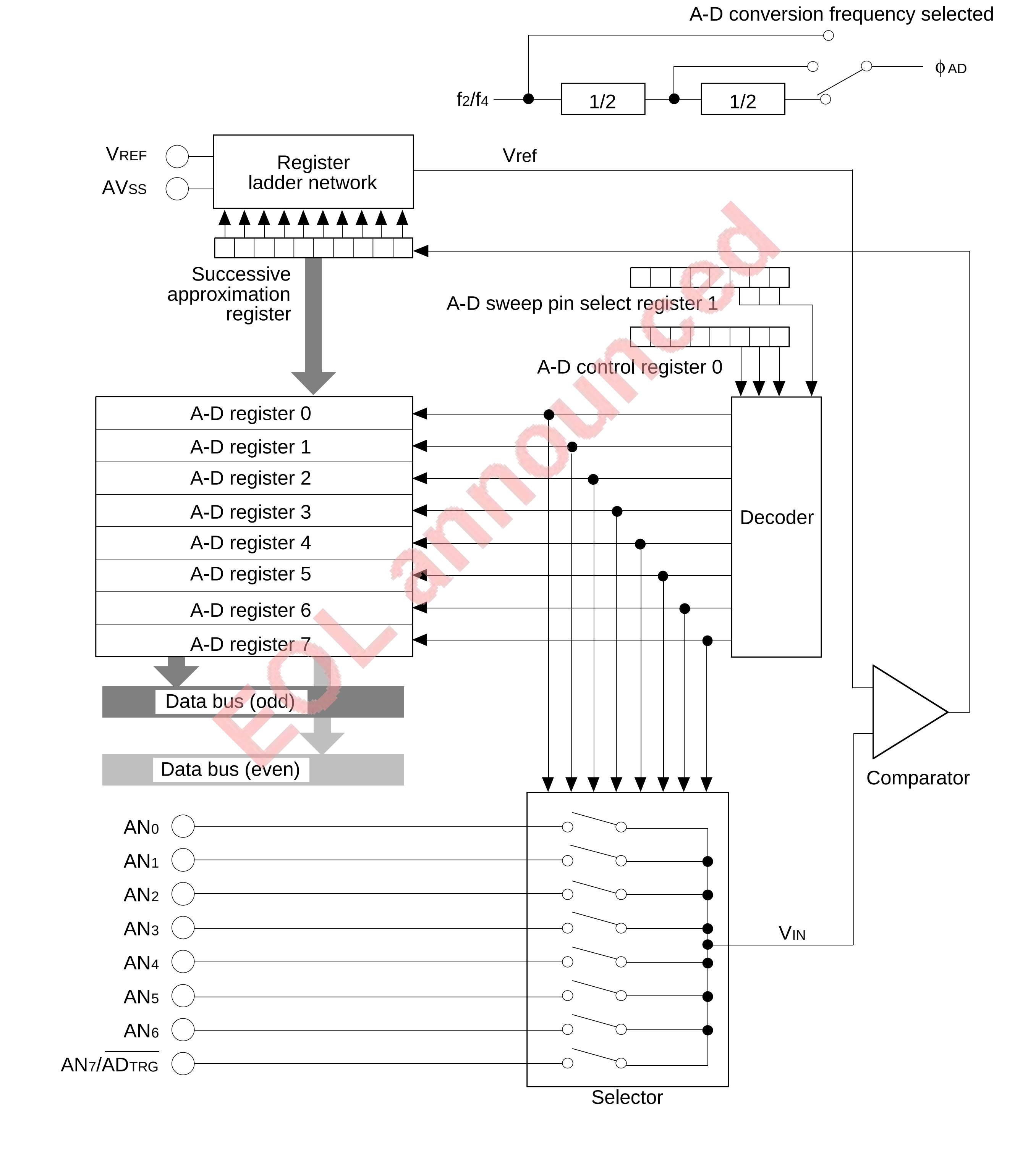
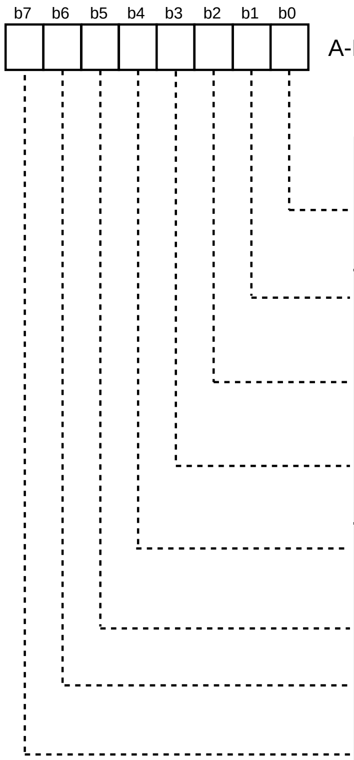


Fig. 8.2.1 Block diagram of A-D converter

8.2.1 A-D control register 0

Figure 8.2.2 shows the structure of the A-D control register 0. The A-D operation mode select bits 0 select the operation mode of the A-D converter. The other bits are described below.



A-D control register 0 (Address $1E_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Analog input select bits (Valid in one-shot and repeat modes) **(Note 1)**	b2 b1 b0 0 0 0 : AN_0 selected 0 0 1 : AN_1 selected 0 1 0 : AN_2 selected 0 1 1 : AN_3 selected 1 0 0 : AN_4 selected 1 0 1 : AN_5 selected 1 1 0 : AN_6 selected 1 1 1 : AN_7 selected **(Note 2)**	Undefined	RW
1			Undefined	RW
2			Undefined	RW
3	A-D operation mode select bits 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 / Repeat sweep mode 1 **(Note 3)**	0	RW
4			0	RW
5	Trigger select bit	0 : Internal trigger 1 : External trigger	0	RW
6	A-D conversion start bit	0 : Stop A-D conversion 1 : Start A-D conversion	0	RW
7	A-D conversion frequency (ϕ_{AD}) select bit 0	Refer to **Table 8.2.1**	0	RW

Notes 1: These bits are ignored in the single sweep and repeat sweep mode 0 and repeat sweep mode 1. (They may be either "0" or "1.")
2: When selecting an external trigger, the AN_7 pin cannot be used as an analog input pin.
3: Use the A-D operation mode select bit 1 (bit 2 at address $1F_{16}$) to select either the repeat sweep mode 0 or repeat sweep mode 1.
4: Writing to each bit (except bit 6) of the A-D control register 0 must be performed while the A-D converter halts.

Fig. 8.2.2 Structure of A-D control register 0

(1) Analog input select bits (bits 2 to 0)

These bits are used to select an analog input pin in the one-shot mode and repeat mode. Pins which are not selected as analog input pins function as programmable I/O ports.

These bits must be set again when the user switches the A-D operation mode to the one-shot mode or repeat mode after performing the operation in the single sweep mode, repeat sweep mode 0 or repeat sweep mode 1.

(2) Trigger select bit (bit 5)
This bit is used to select the source of trigger occurrence. (Refer to **"(3) A-D conversion start bit."**)

(3) A-D conversion start bit (bit 6)

● **When internal trigger is selected**

Setting this bit to "1" generates a trigger, causing the A-D converter to start operating. Clearing this bit to "0" causes the A-D converter to stop operating.
In the one-shot mode or single sweep mode, this bit is cleared to "0" after the operation is completed. In the repeat mode, repeat sweep mode 0 or repeat sweep mode 1, the A-D converter continues operating until this bit is cleared to "0" by software.

● **When external trigger is selected**

When the $\overline{\text{ADTRG}}$ pin level goes from "H" to "L" with this bit = "1," a trigger occurs, causing the A-D converter to start operating. The A-D converter stops when this bit is cleared to "0."
In the one-shot mode or single sweep mode, this bit remains set to "1" even after the operation is completed. In the repeat mode, repeat sweep mode 0 or repeat sweep mode 1, the A-D converter continues operating until this bit is cleared to "0" by software.

(4) A-D conversion frequency (ϕ_{AD}) select bit 0 (bit 7)
The operating time of the A-D converter varies depending on the selected operating clock (ϕ_{AD}) by this bit and the A-D conversion frequency (ϕ_{AD}) select bit 1 (bit 4 at address $1F_{16}$; refer to Figure 8.2.3) as listed in Table 8.2.3.
Since the A-D converter's comparator consists of capacity coupling amplifiers, keep that $\phi_{AD} \geq 250$ kHz during A-D conversion.

Table 8.2.1 Time for performance to one analog input pin (unit: μs)

Clock source for peripheral devices select bit		0			1		
A-D conversion frequency (ϕ_{AD}) select bit 1		0	0	1	0	0	1
A-D conversion frequency (ϕ_{AD}) select bit 0		0	1	0	0	1	0
ϕ_{AD}		f_4 divided by 4	f_4 divided by 2	f_4	f_2 divided by 4	f_2 divided by 2	f_2
$f(X_{IN})$ = 25 MHz	8-bit resolution	31.36	15.68	7.84	15.68	7.84	3.92
	10-bit resolution	37.76	18.88	9.44	18.88	9.44	4.72
$f(X_{IN})$ = 40 MHz	8-bit resolution	19.60	9.80	4.90	—	—	—
	10-bit resolution	23.60	11.80	5.90	—	—	—

8.2.2 A-D control register 1

Figure 8.2.3 shows the structure of the A-D control register 1.

The A-D operation mode select bit 1 is used to select the operation mode of the A-D converter. The 8/10-bit mode select bit is used to select the resolution. Refer to Table 8.2.1 for the A-D conversion frequency (ϕ_{AD}) select bit 1.

b7 b6 b5 b4 b3 b2 b1 b0

A-D control register 1 (Address $1F_{16}$)

Bit	Bit name	Functions	At reset	RW
0	A-D sweep pin select bits (Valid in single sweep, repeat sweep mode 0 and repeat sweep mode 1) **(Note 1)**	●Single sweep mode/Repeat sweep mode 0 b1 b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0 to AN_3 (4 pins) 1 0 : AN_0 to AN_5 (6 pins) 1 1 : AN_0 to AN_7 (8 pins) **(Note 2)**	1	RW
1		●Repeat sweep mode 1 **(Note 3)** b1 b0 0 0 : AN_0 (1 pin) 0 1 : AN_0, AN_1 (2 pins) 1 0 : AN_0 to AN_2 (3 pins) 1 1 : AN_0 to AN_3 (4 pins)	1	RW
2	A-D operation mode select bit 1 (Use in repeat sweep mode 0 and repeat sweep mode 1) **(Note 4)**	0 : Repeat sweep mode 0 1 : Repeat sweep mode 1	0	RW
3	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	0	RW
4	A-D conversion frequency (ϕ_{AD}) select bit 1	Refer to A-D conversion frequency (ϕ_{AD}) select bit 0 (bit 7 at address $1E_{16}$) ; See **Table 8.2.1**	0	RW
7 to 5	Nothing is assigned.		Undefined	—

Notes 1: These bits are invalid in the one-shot and repeat modes. (They may be either "0" or "1.")
2: When selecting an external trigger, the AN_7 pin cannot be used as an analog input pin.
3: Analog input pins which are frequently A-D converted are selected in the repeat sweep mode 1.
4: Fix this bit to "0" in the one-shot, repeat, and single sweep modes.
5: Writing to each bit of the A-D control register 1 must be performed while the A-D converter halts.

Fig. 8.2.3 Structure of A-D control register 1

(1) A-D sweep pin selection bits (bits 1 and 0)

These bits are used to select analog input pins in the single sweep mode, repeat sweep mode 0 and repeat sweep mode 1. In the single sweep mode and repeat sweep mode 0, pins which are not selected as analog input pins function as programmable I/O ports.

8.2.3 A-D register i (i = 0 to 7)

Figure 8.2.4 shows the structure of the A-D register i. When the A-D conversion is completed, the conversion result (contents of the successive approximation register) is stored into this register. Each A-D register i corresponds to an analog input pin (AN_i). Table 8.2.2 lists the correspondence of an analog input pin to A-D register i.

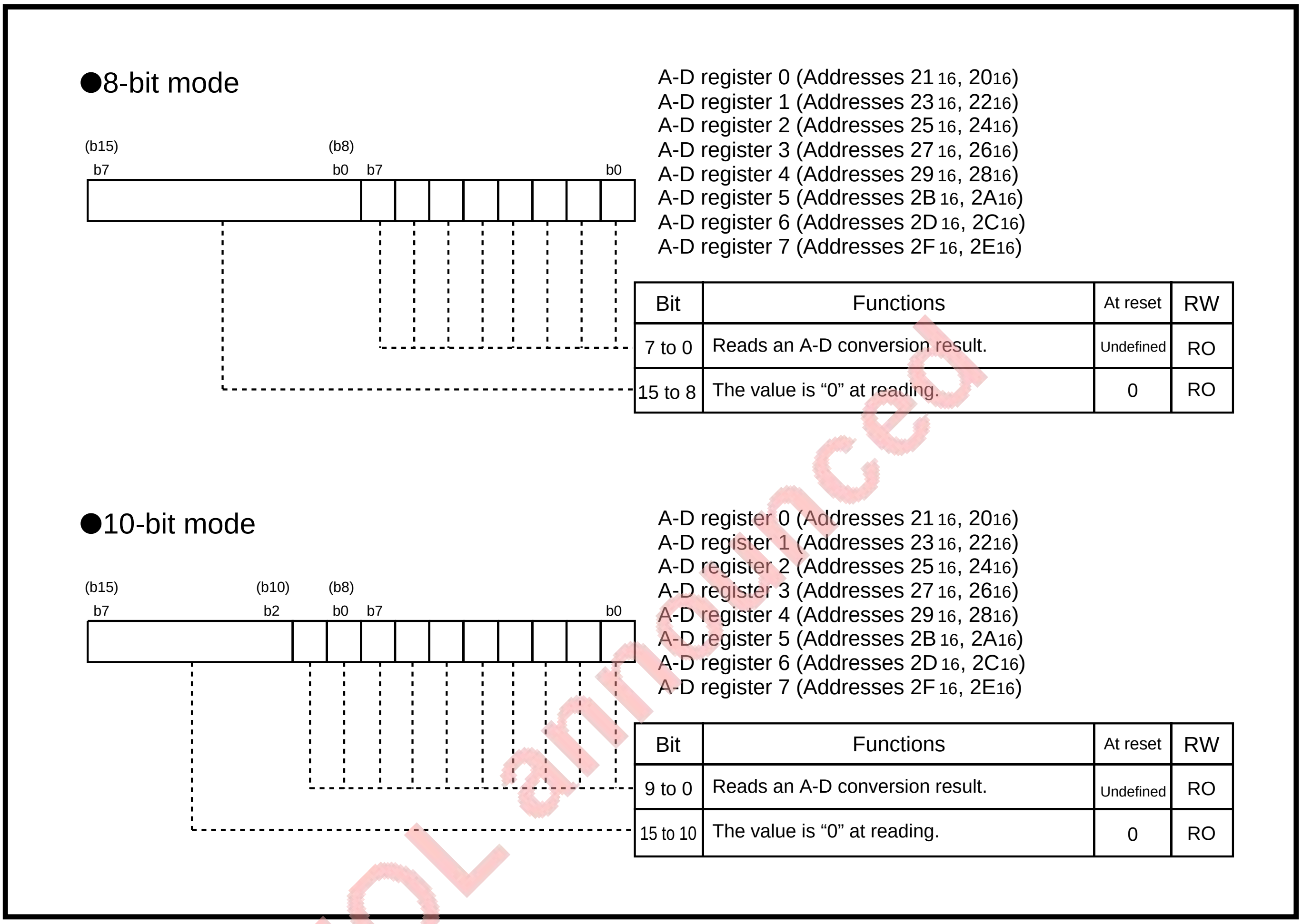


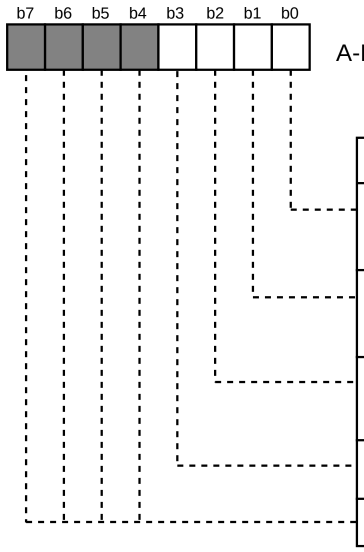
Fig. 8.2.4 Structure of A-D register i

Table 8.2.2 Correspondence of analog input pin and A-D register i

Analog input pin	A-D register i where conversion result is stored
AN_0 pin	A-D register 0
AN_1 pin	A-D register 1
AN_2 pin	A-D register 2
AN_3 pin	A-D register 3
AN_4 pin	A-D register 4
AN_5 pin	A-D register 5
AN_6 pin	A-D register 6
AN_7 pin	A-D register 7

8.2.4 A-D conversion interrupt control register

Figure 8.2.5 shows the structure of the A-D conversion interrupt control register. For details about interrupts, refer to **"Chapter 4. INTERRUPTS."**



A-D conversion interrupt control register (Address 70_{16})

Bit	Bit name	Functions	At reset	RW
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 Low level 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7 High level	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit	0 : No interrupt request 1 : Interrupt request	Undefined **(Note)**	RW
7 to 4	Nothing is assigned.		Undefined	–

Note : Clear the bit 3 to "0" by software when using A-D conversion interrupt.

Fig. 8.2.5 Structure of A-D conversion interrupt control register

(1) Interrupt priority level select bits (bits 2 to 0)

These bits select the A-D conversion interrupt's priority level. When using A-D conversion interrupts, select priority levels 1 to 7. When an A-D conversion interrupt request occurs, its priority level is compared with the processor interrupt priority level (IPL) and the requested interrupt is enabled only when its priority level is higher than the IPL. (However, this applies when the interrupt disable flag (I) = "0.") To disable the A-D conversion interrupt, set these bits to "000_2" (level 0).

(2) Interrupt request bit (bit 3)

This bit is set to "1" when an A-D conversion interrupt request occurs. This bit is automatically cleared to "0" when the A-D conversion interrupt request is accepted. This bit can be set to "1" or cleared to "0" by software.

8.2.5 Port P7 direction register

The A-D converter and port P7 use the same pins in common. When using these pins as the A-D converter's input pins, set the corresponding bits of the port P7 direction register to "0" to set these ports for the input mode. Figure 8.2.6 shows the relationship between the port P7 direction register and A-D converter's input pins.

b7 b6 b5 b4 b3 b2 b1 b0

Port P7 direction register (Address 11_{16})

Bit	Corresponding pin	Functions	At reset	RW
0	AN_0 pin	0 : Input mode 1 : Output mode When using these pins as A-D converter's input pins, set the corresponding bits to "0."	0	RW
1	AN_1 pin		0	RW
2	AN_2 pin		0	RW
3	AN_3 pin		0	RW
4	AN_4 pin		0	RW
5	AN_5 pin		0	RW
6	AN_6 pin		0	RW
7	$AN_7/\overline{AD_{TRG}}$ pin		0	RW

Fig. 8.2.6 Relationship between port P7 direction register and A-D converter's input pins

8.3 A-D conversion method (successive approximation conversion method)

The A-D converter compares the comparison voltage (V_{ref}), which is internally generated according to the contents of the successive approximation register, with the analog input voltage (V_{IN}), which is input from the analog input pin (AN_i). By reflecting the comparison result on the successive approximation register, V_{IN} is converted into a digital value. When a trigger is generated, the A-D converter performs the following processing:

① **Determining bit 9 of the successive approximation register**
The A-D converter compares V_{ref} with V_{IN}. At this point, the contents of the successive approximation register are "1000000000_2" (initial value).
Bit 9 of the successive approximation register changes according to the comparison result as follows:
When $V_{ref} < V_{IN}$, bit 9 = "1"
When $V_{ref} > V_{IN}$, bit 9 = "0"

② **Determining bit 8 of the successive approximation register**
After setting bit 8 of the successive approximation register to "1," the A-D converter compares V_{ref} with V_{IN}. Bit 8 changes according to the comparison result as follows:
When $V_{ref} < V_{IN}$, bit 8 = "1"
When $V_{ref} > V_{IN}$, bit 8 = "0"

③ **Determining bits 7 to 0 of the successive approximation register**
Operation in ② are performed for bits 7 to 0 in the 10-bit mode.
Operation in ② are performed for bits 7 to 2 in the 8-bit mode.
When the LSB is determined, the contents (conversion result) of the successive approximation register are transferred to the A-D register i.

The comparison voltage (V_{ref}) is generated according to the latest contents of the successive approximation register. Table 8.3.1 lists the relationship between the successive approximation register's contents and V_{ref}. Table 8.3.2 and Table 8.3.3 list changes of the successive approximation register and V_{ref} during the A-D conversion. Figure 8.3.1 shows the ideal A-D conversion characteristics in the 10-bit mode.

Table 8.3.1 Relationship between successive approximation register's contents and V_{ref}

Successive approximation register's contents: n	V_{ref} (V)
0	0
1 to 1023	$\frac{V_{REF}*}{1024} \times (n - 0.5)$

V_{REF}*: Reference voltage

Table 8.3.2 Change in successive approximation register and Vref during A-D conversion in 8-bit mode

	Successive approximation register (b9 … b0)	Change of Vref
A-D converter halt	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$ [V]
1st comparison ↓	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{2048}$ [V]
2nd comparison ↓	n_9 1 0 0 0 0 0 0 0 0 (↑ 1st comparison result)	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{2048}$ [V] (•n_9=1 $+\frac{V_{REF}}{4}$; •n_9=0 $-\frac{V_{REF}}{4}$)
3rd comparison ↓	n_9 n_8 1 0 0 0 0 0 0 0 (↑ 2nd comparison result)	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{2048}$ [V] (•n_8=1 $+\frac{V_{REF}}{8}$; •n_8=0 $-\frac{V_{REF}}{8}$)
⋮	⋮	⋮
8th comparison ↓	n_9 n_8 n_7 n_6 n_5 n_4 n_3 1 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} \pm \ldots\ldots \pm \frac{V_{REF}}{256} - \frac{V_{REF}}{2048}$ [V]
Conversion complete	n_9 n_8 n_7 n_6 n_5 n_4 n_3 n_2 0 0	

Table 8.3.3 Change in successive approximation register and Vref during A-D conversion in 10-bit mode

	Successive approximation register (b9 … b0)	Change of Vref
A-D converter halt	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$ [V]
1st comparison ↓	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{2048}$ [V]
2nd comparison ↓	n_9 1 0 0 0 0 0 0 0 0 (↑ 1st comparison result)	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{2048}$ [V] (•n_9=1 $+\frac{V_{REF}}{4}$; •n_9=0 $-\frac{V_{REF}}{4}$)
3rd comparison ↓	n_9 n_8 1 0 0 0 0 0 0 0 (↑ 2nd comparison result)	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{2048}$ [V] (•n_8=1 $+\frac{V_{REF}}{8}$; •n_8=0 $-\frac{V_{REF}}{8}$)
⋮	⋮	⋮
10th comparison ↓	n_9 n_8 n_7 n_6 n_5 n_4 n_3 n_2 n_1 1	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} \pm \ldots\ldots \pm \frac{V_{REF}}{1024} - \frac{V_{REF}}{2048}$ [V]
Conversion complete	n_9 n_8 n_7 n_6 n_5 n_4 n_3 n_2 n_1 n_0	

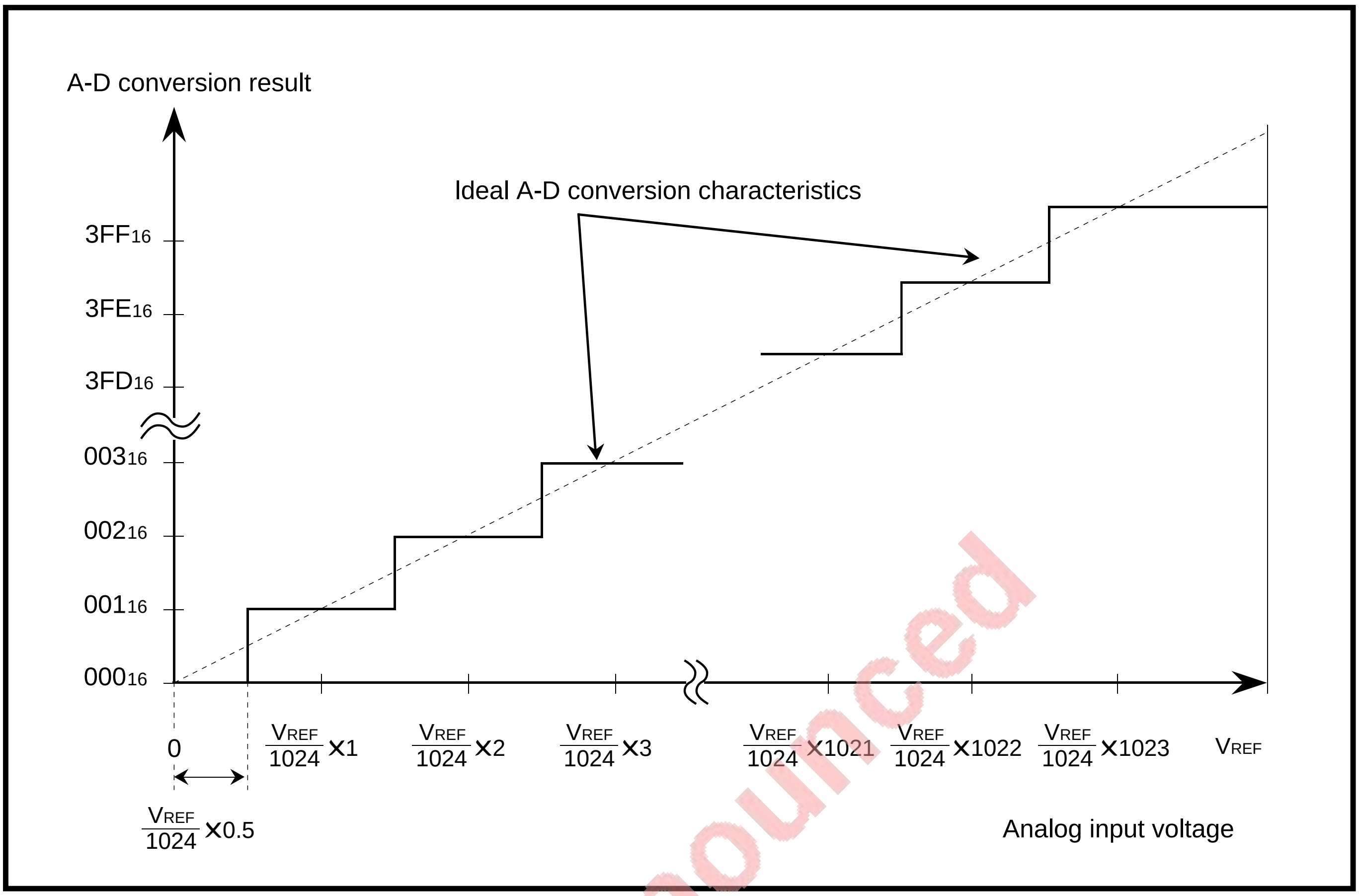


Fig. 8.3.1 Ideal A-D conversion characteristics in 10-bit mode

8.4 Absolute accuracy and differential non-linearity error

8.4.1 Absolute accuracy

The absolute accuracy is the difference expressed in the LSB between the actual A-D conversion result and the output code of an A-D converter with ideal characteristics. The analog input voltage when measuring the accuracy is assumed to be the mid point of the input voltage width that outputs the same output code from an A-D converter with ideal characteristics. For example, in the case of the 10-bit mode, when V_{REF}=5.12 V, 1 LSB width is 5 mV, and 0 mV, 5 mV, 10 mV, 15 mV, 20 mV, ... are selected as the analog input voltages.

The absolute accuracy = ±3 LSB indicates that when the analog input voltage is 25 mV, the output code expected from an ideal A-D conversion characteristics is "005_{16}," but the actual A-D conversion result is between "002_{16}" to "008_{16}."

The absolute accuracy includes the zero error and the full-scale error.

The absolute accuracy degrades when V_{REF} is lowered. The output code for analog input voltages V_{REF} to AV_{CC} is "$3FF_{16}$."

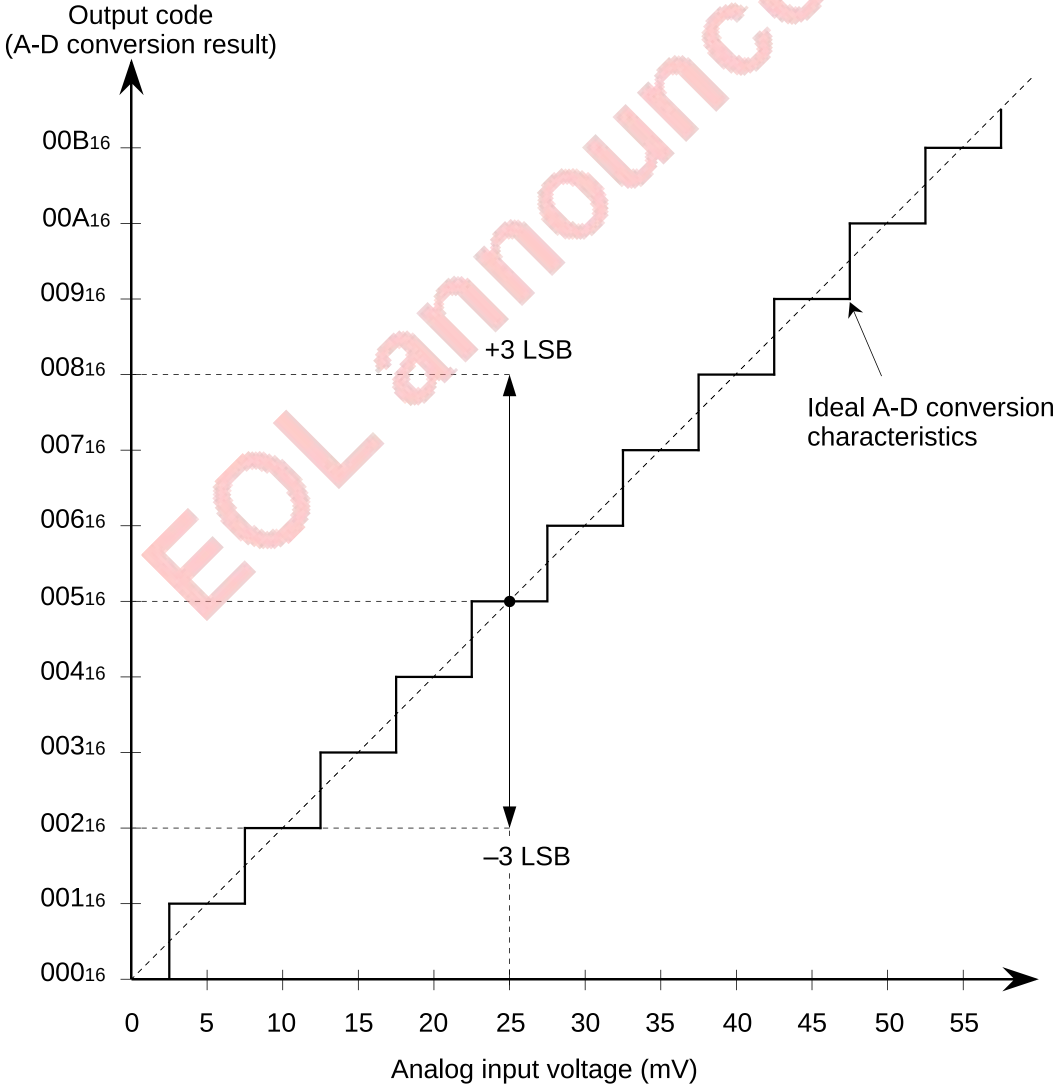


Fig. 8.4.1 Absolute accuracy of A-D converter in 10-bit mode

8.4 Absolute accuracy and differential non-linearity error

8.4.2 Differential non-linearity error

The differential non-linearity error indicates the difference between the 1 LSB step width (the ideal analog input voltage width while the same output code is expected to output) of an A-D converter with ideal characteristics and the actual measured step width (the actual analog input voltage width while the same output code is output). For example, in the case of the 10-bit mode, when V_{REF}=5.12 V, the 1 LSB width of an A-D converter with ideal characteristics is 5 mV, but if the differential non-linearity error is ±1 LSB, the actual measured 1 LSB width is 0 to 10 mV.

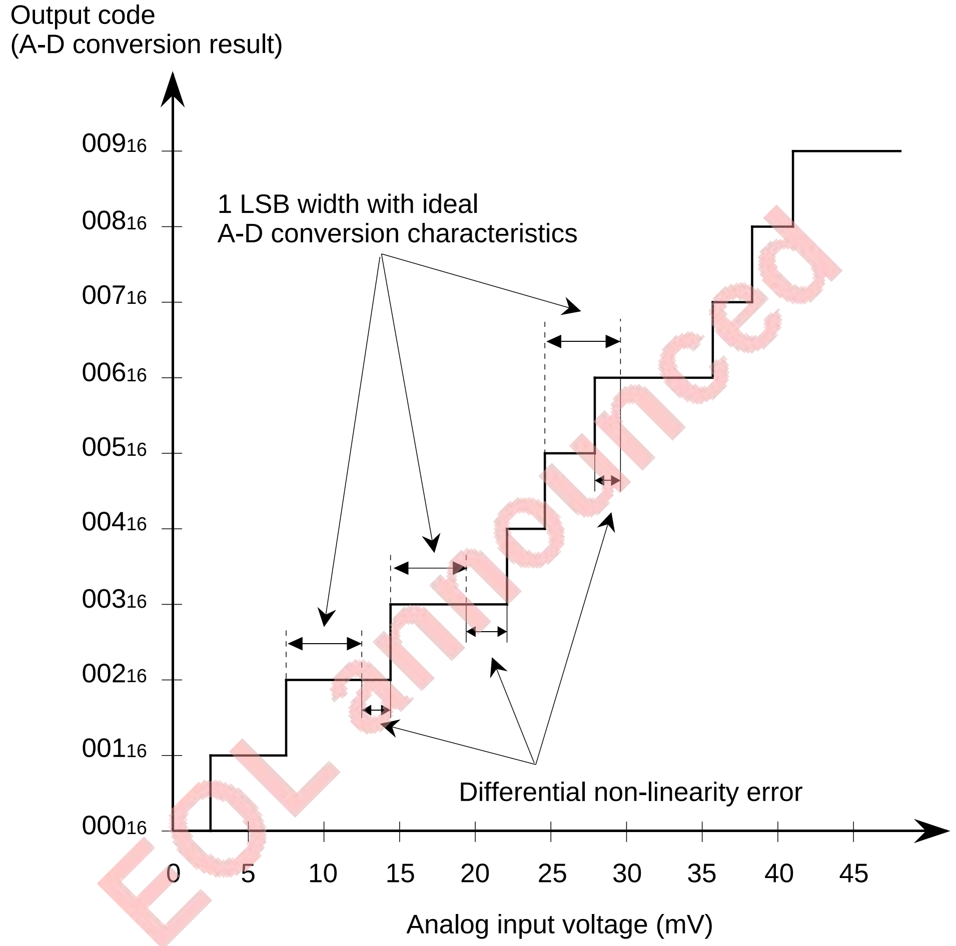


Fig. 8.4.2 Differential non-linearity error in 10-bit mode

8.5 Comparison voltage in 8-bit mode

In the 8-bit mode, the M37751 treats the high-order 8 bits of the 10-bit successive approximation register as the conversion result. Accordingly, when compared with the 8-bit A-D converter, the A-D conversion of the M37751 is performed by using a comparison reference voltage that is different by $3V_{REF}/2048$ (refer to the underlined portions in the Table 8.5.1). The difference of the output code change point is generated as shown in Figure 8.5.1.

Table 8.5.1 Comparison reference voltage of the M37751's 8-bit mode and 8-bit A-D converter

	M37751's 8-bit mode	8-bit A-D converter
Comparison reference voltage Vref	$V_{REF}/2^8 \times n - \underline{V_{REF}/2^{10} \times 0.5}$	$V_{REF}/2^8 \times n - \underline{V_{REF}/2^8 \times 0.5}$

V_{REF}: Reference voltage
n : Contents of successive approximation register

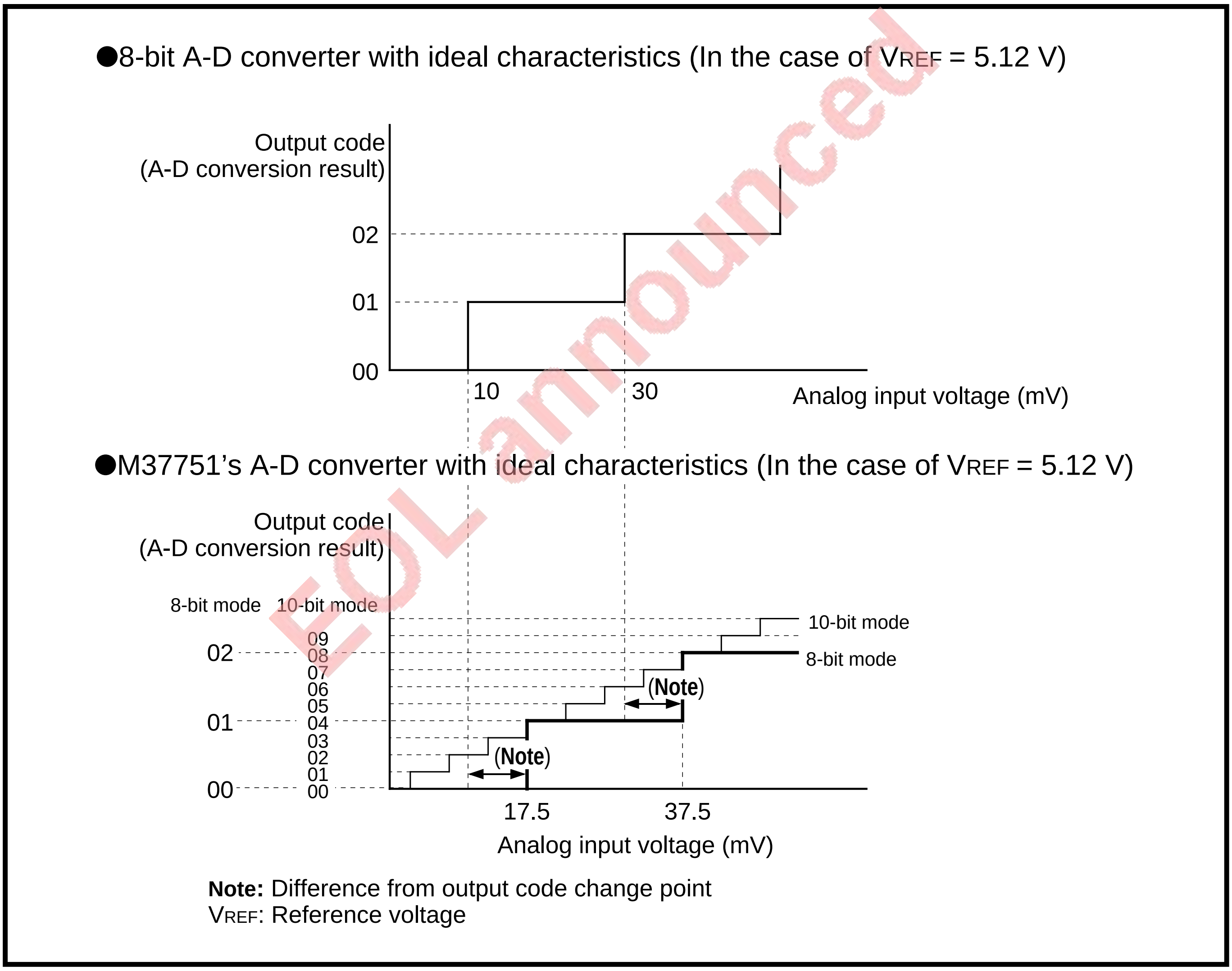


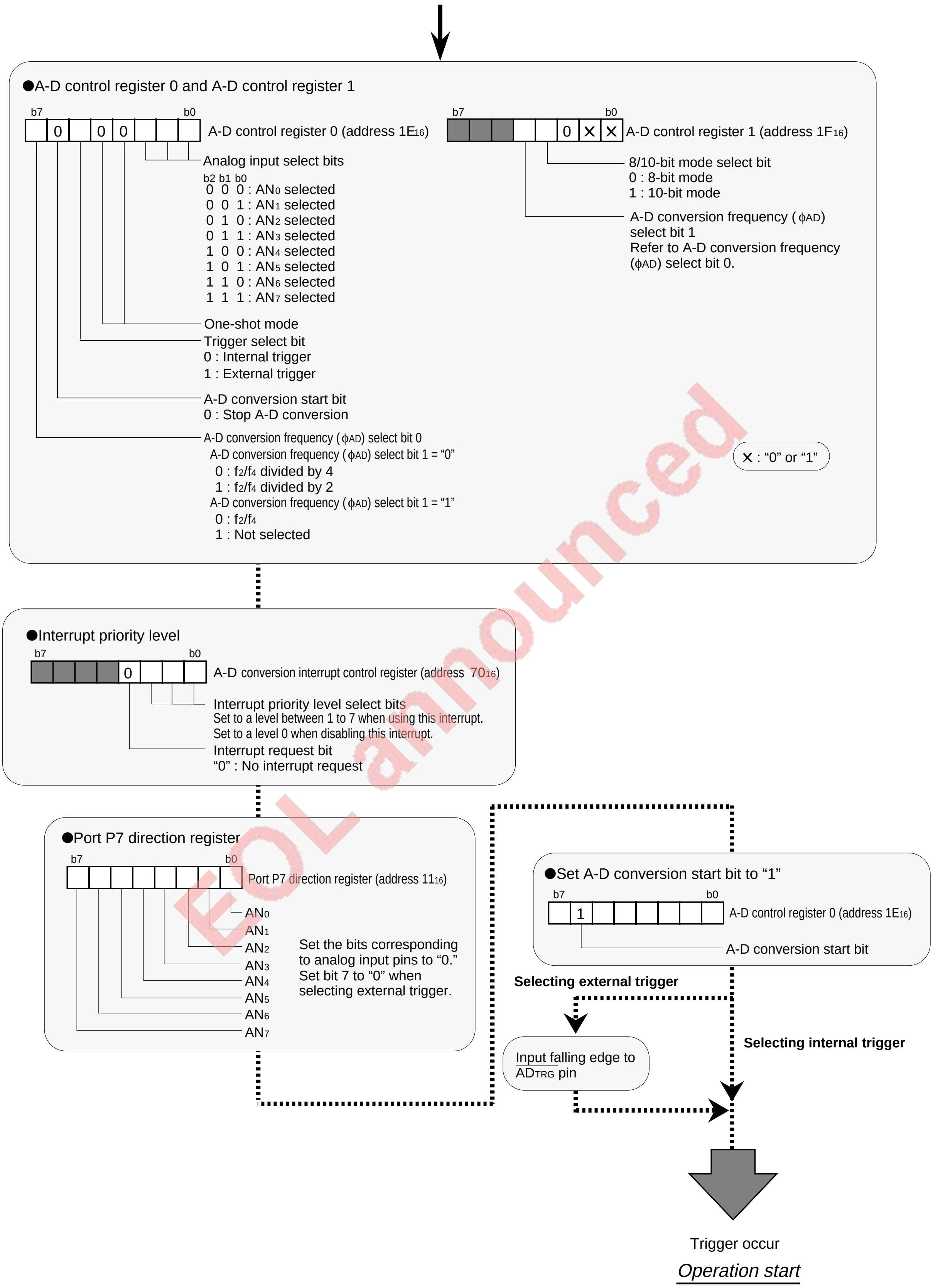
Fig. 8.5.1 Difference of output code change point

8.6 One-shot mode

In the one-shot mode, the operation for the input voltage from the one selected analog input pin is performed once, and an A-D conversion interrupt request occurs when the operation is completed.

8.6.1 Settings for one-shot mode

Figure 8.6.1 shows an initial setting example of the one-shot mode.
When using an interrupt, it is necessary to set the corresponding register to enable interrupts. Refer to **"Chapter 4. INTERRUPTS"** for more descriptions.



Note: Write the following registers when the A-D conversion stops (before trigger occurs).
- Each bit of A-D control register 0 (except bit 6)
- Each bit of A-D control register 1

Fig. 8.6.1 Initial setting example of one-shot mode

8.6.2 One-shot mode operation description

(1) When an internal trigger is selected

① The A-D converter starts operation when the A-D conversion start bit is set to "1."

② The A-D conversion is completed after 49 cycles of ϕ_{AD} in the 8-bit mode, or 59 cycles of ϕ_{AD} in the 10-bit mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register i.

③ At the same time as step ②, the A-D conversion interrupt request bit is set to "1."

④ The A-D conversion start bit is cleared to "0" and the A-D converter stops operation.

(2) When an external trigger is selected

① The A-D converter starts operation when the input level to the $\overline{AD_{TRG}}$ pin changes from "H" to "L" while the A-D conversion start bit is "1."

② The A-D conversion is completed after 49 cycles of ϕ_{AD} in the 8-bit mode, or 59 cycles of ϕ_{AD} in the 10-bit mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register i.

③ At the same time as step ②, the A-D conversion interrupt request bit is set to "1."

④ The A-D conversion stops operation.

The A-D conversion start bit remains set to "1" after the operation is completed. Accordingly, the operation of the A-D converter can be performed again from step ① when the level of the $\overline{AD_{TRG}}$ pin changes from "H" to "L."
When the level of the $\overline{AD_{TRG}}$ pin changes from "H" to "L" during operation, the operation at that point is cancelled and is restarted from step ①.

Figure 8.6.2 shows the conversion operation in the one-shot mode.

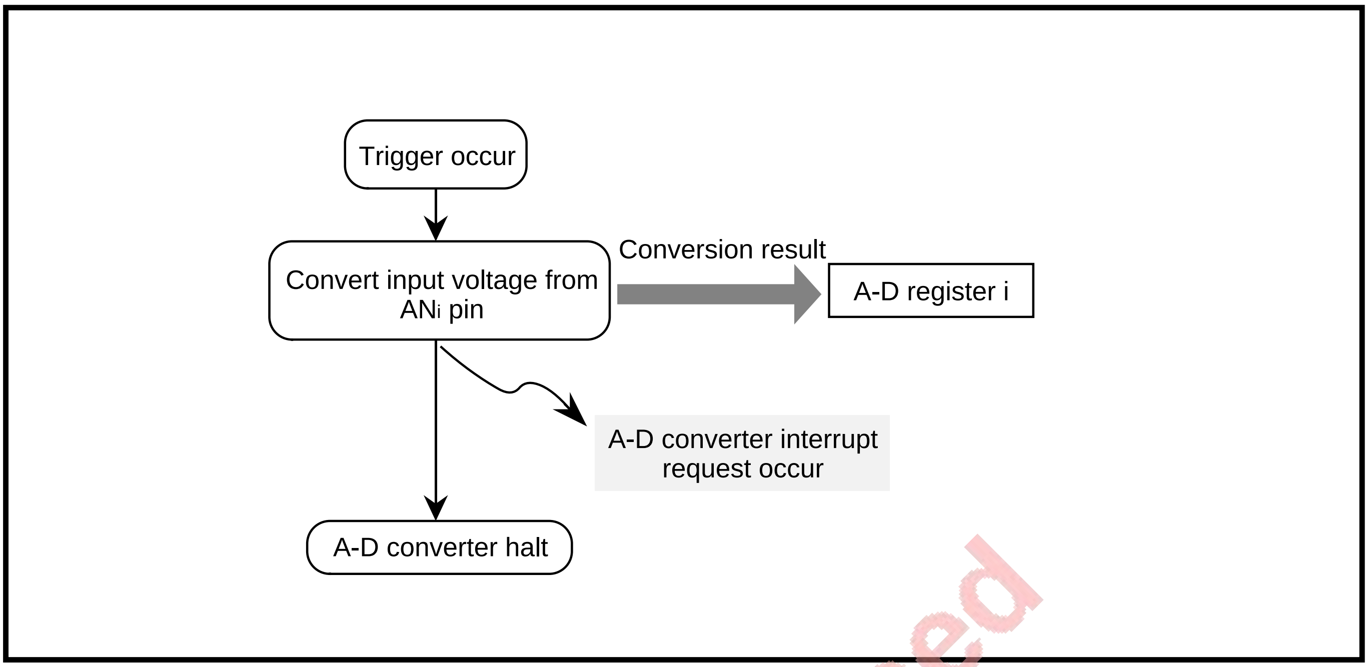


Fig. 8.6.2 Conversion operation in one-shot mode

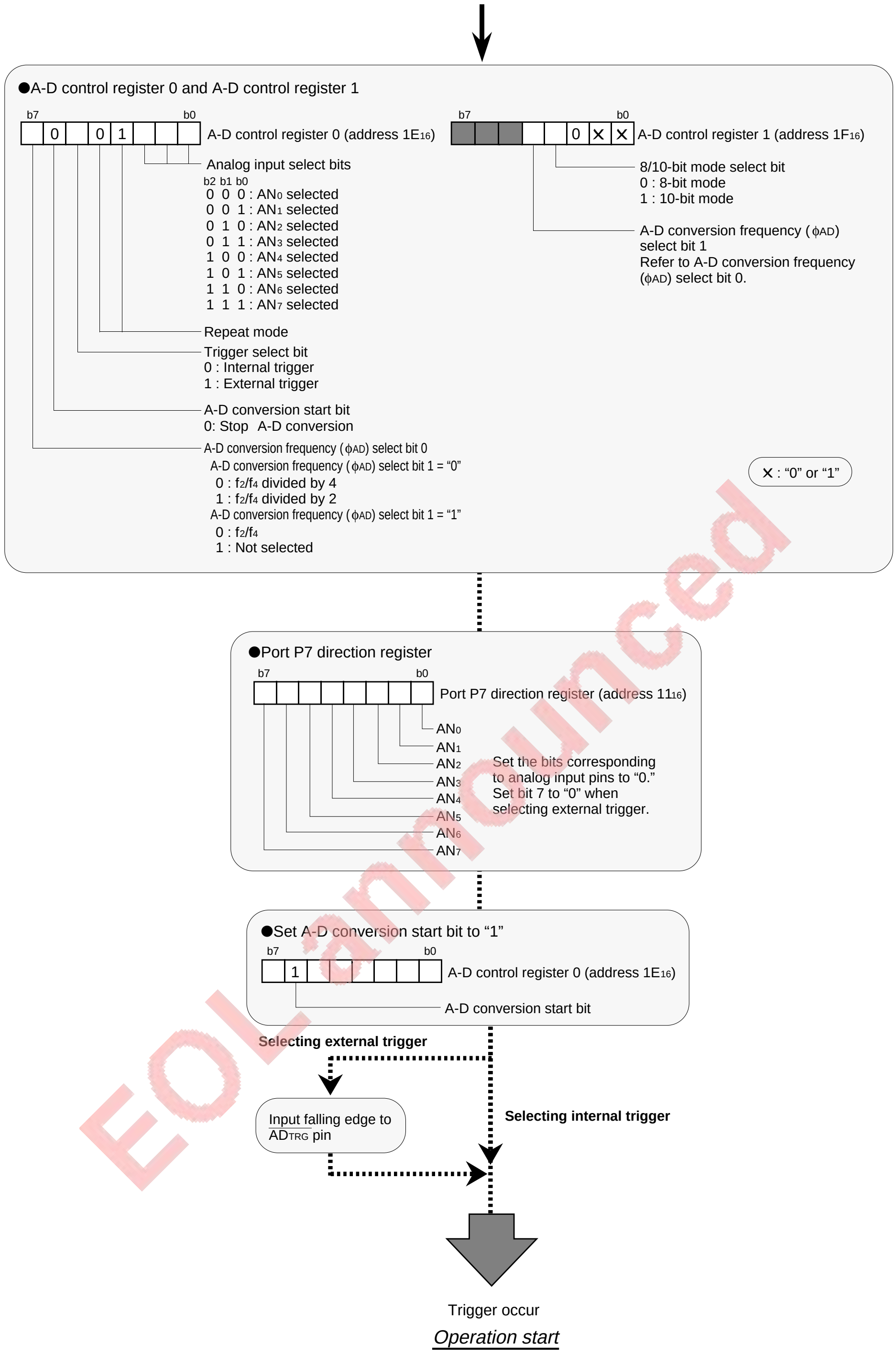
8.7 Repeat mode

In the repeat mode, the operation for the input voltage from the one selected analog input pin is performed repeatedly.

In this mode, no A-D conversion interrupt request occurs. Additionally, the A-D conversion start bit (bit 6 at address $1E_{16}$) remains set to "1" until it is cleared to "0" by software, and the operation is performed repeatedly while the A-D conversion start bit is "1."

8.7.1 Settings for repeat mode

Figure 8.7.1 shows an initial setting example of repeat mode.



Note: Write the following registers when the A-D conversion stops (before trigger occurs).
- Each bit of A-D control register 0 (except bit 6)
- Each bit of A-D control register 1

Fig. 8.7.1 Initial setting example of repeat mode

8.7.2 Repeat mode operation description

(1) When an internal trigger is selected

① The A-D converter starts operation when the A-D conversion start bit is set to "1."

② The first A-D conversion is completed after 49 cycles of ϕ_{AD} in the 8-bit mode, or 59 cycles of ϕ_{AD} in the 10-bit mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register i.

③ The A-D converter repeats operation until the A-D conversion start bit is cleared to "0" by software. The conversion result is transferred to the A-D register i each time the conversion is completed.

(2) When an external trigger is selected

① The A-D converter starts operation when the input level to the $\overline{AD_{TRG}}$ pin changes from "H" to "L" while the A-D conversion start bit is "1."

② The first A-D conversion is completed after 49 cycles of ϕ_{AD} in the 8-bit mode, or 59 cycles of ϕ_{AD} in the 10-bit mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register i.

③ The A-D converter repeats operation until the A-D conversion start bit is cleared to "0" by software. The conversion result is transferred to the A-D register i each time the conversion is completed. When the comparator function is selected, the comparison result is stored in the AN_i pin comparator result bit each time the comparison is completed.

When the level of the $\overline{AD_{TRG}}$ pin changes from "H" to "L" during operation, the operation at that point is cancelled and is restarted from step ①.

Figure 8.7.2 shows the conversion operation in the repeat mode.

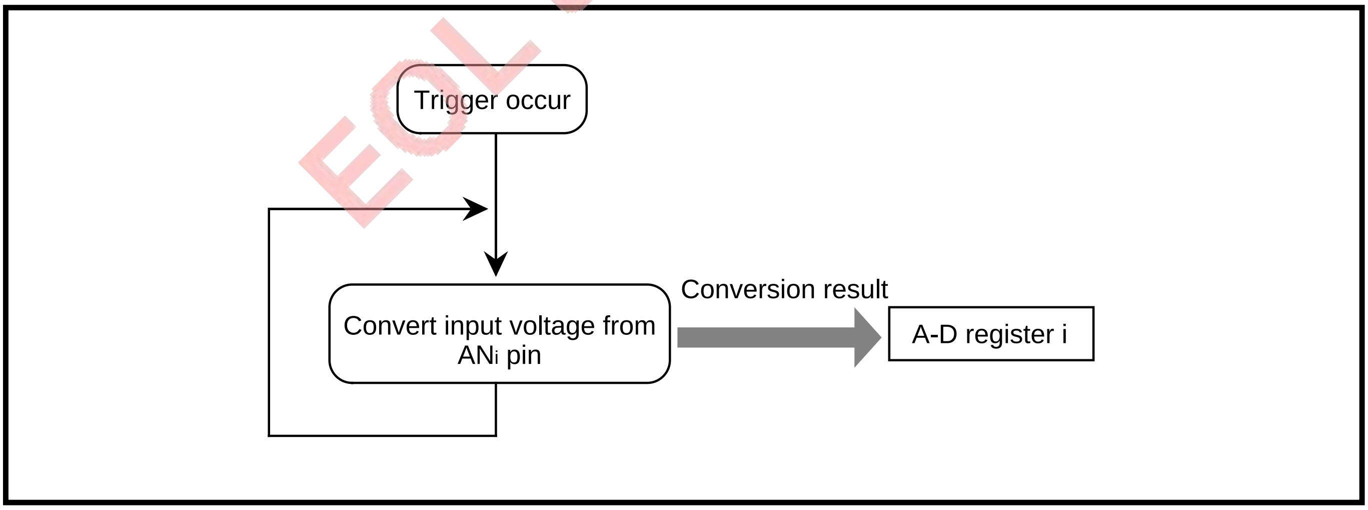


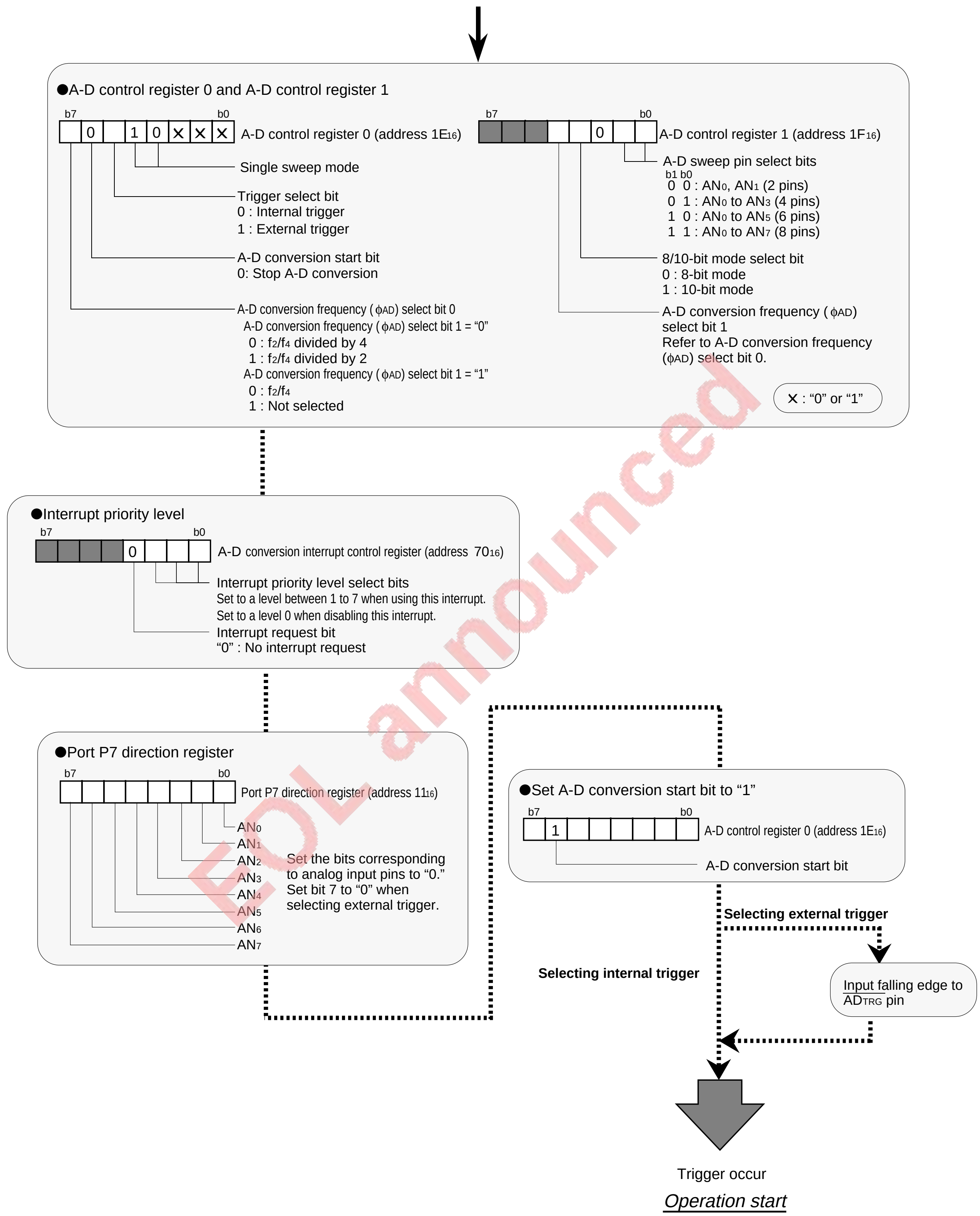
Fig. 8.7.2 Conversion operation in repeat mode

8.8 Single sweep mode

In the single sweep mode, the operation for the input voltage from multiple selected analog input pins is performed, one at a time. The A-D converter is operated in ascending sequence from the AN_0 pin. An A-D conversion interrupt request occurs when the operation for all selected input pins are completed.

8.8.1 Settings for single sweep mode

Figure 8.8.1 shows an initial setting example of single sweep mode.
When using an interrupt, it is necessary to set the corresponding register to enable interrupts. Refer to **"Chapter 4. INTERRUPTS"** for more information.



Note: Write the following registers when the A-D conversion stops (before trigger occurs).
- Each bit of A-D control register 0 (except bit 6)
- Each bit of A-D control register 1

Fig. 8.8.1 Initial setting example of single sweep mode

8.8.2 Single sweep mode operation description

(1) When an internal trigger is selected

① The A-D converter starts conversion for the input voltage from the AN_0 pin starts when the A-D conversion start bit is set to "1."

② The A-D conversion of the input voltage from the AN_0 pin is completed after 49 cycles of ϕ_{AD} in the 8-bit mode, or 59 cycles of ϕ_{AD} in the 10-bit mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register 0.

③ The operation to all selected analog input pins is performed.
The conversion result is transferred to the A-D register i each time each pin is converted.

④ When the step ③ is completed, the A-D conversion interrupt request bit is set to "1."

⑤ The A-D conversion start bit is cleared to "0" and the A-D converter stops operation.

(2) When an external trigger is selected

① The A-D converter starts conversion for the input voltage from the AN_0 pin when the input level to the $\overline{AD_{TRG}}$ pin changes from "H" to "L" while the A-D conversion start bit is "1."

② The A-D conversion of the input voltage from the AN_0 pin is completed after 49 cycles of ϕ_{AD} in the 8-bit mode, or 59 cycles of ϕ_{AD} in the 10-bit mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register 0.

③ The operation to all selected analog input pins is performed.
The conversion result is transferred to the A-D register i each time each pin is converted.

④ When the step ③ is completed, the A-D conversion interrupt request bit is set to "1."

⑤ The A-D conversion stops operation.

The A-D conversion start bit remains set to "1" after the operation is completed. Accordingly, the operation of the A-D converter can be performed again from step ① when the level of the $\overline{AD_{TRG}}$ pin changes from "H" to "L."
When the level of the $\overline{AD_{TRG}}$ pin changes from "H" to "L" during operation, the operation at that point is cancelled and is restarted from step ①.

Figure 8.8.2 shows the conversion operation in the single sweep mode.

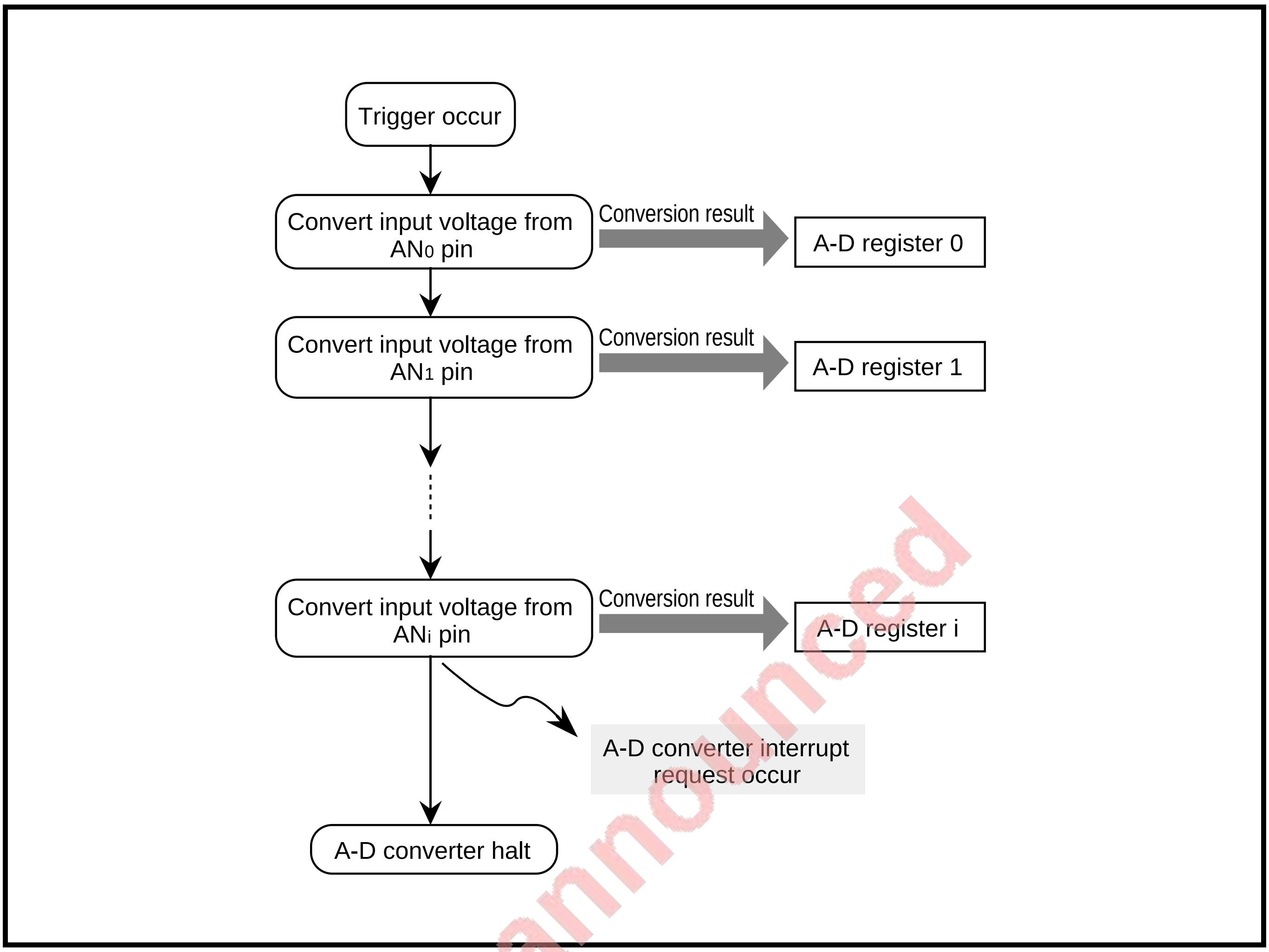


Fig. 8.8.2 Conversion operation in single sweep mode

8.9 Repeat sweep mode 0

In the repeat sweep mode 0, the operation for the input voltage from the multiple selected analog input pins is performed repeatedly. The A-D converter is operated in ascending sequence from the AN_0 pin.
In this mode, no A-D conversion interrupt request occurs. Additionally, the A-D conversion start bit (bit 6 at address $1E_{16}$) remains set to “1” until it is cleared to “0” by software, and the operation is performed repeatedly while the A-D conversion start bit is “1.”

8.9.1 Settings for repeat sweep mode 0

Figure 8.9.1 shows an initial setting example of repeat sweep mode 0.

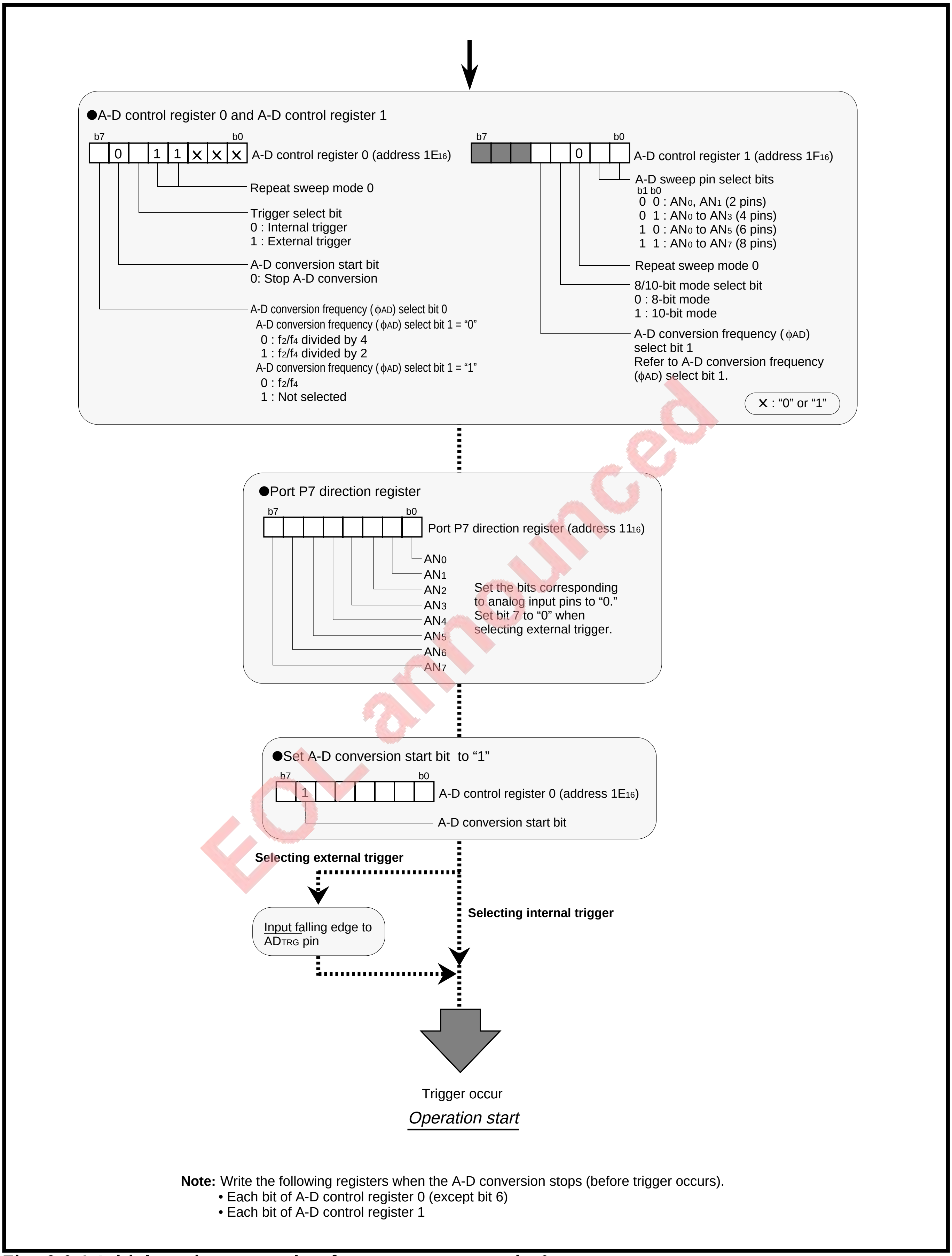


Fig. 8.9.1 Initial setting example of repeat sweep mode 0

8.9.2 Repeat sweep mode 0 operation description

(1) When an internal trigger is selected

① The A-D converter starts conversion for the input voltage from the AN_0 pin starts when the A-D conversion start bit is set to "1."

② The A-D conversion of the input voltage from the AN_0 pin is completed after 49 cycles of ϕ_{AD} in the 8-bit mode, or 59 cycles of ϕ_{AD} in the 10-bit mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register 0.

③ The operation to all selected analog input pins is performed.
The conversion result is transferred to the A-D register i each time each pin is converted.

④ The operation to all selected analog input pins is performed again.

⑤ The operation is performed repeatedly until the A-D conversion start bit is cleared to "0" by software.

(2) When an external trigger is selected

① The A-D converter starts conversion for the input voltage from the AN_0 pin when the input level to the $\overline{AD_{TRG}}$ pin changes from "H" to "L" while the A-D conversion start bit is "1."

② The A-D conversion of the input voltage from the AN_0 pin is completed after 49 cycles of ϕ_{AD} in the 8-bit mode, or 59 cycles of ϕ_{AD} in the 10-bit mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register 0.

③ The operation to all selected analog input pins is performed.
The conversion result is transferred to the A-D register i each time each pin is converted.

④ The operation to all selected analog input pins is performed again.

⑤ The operation is performed repeatedly until the A-D conversion start bit is cleared to "0" by software.

When the level of the $\overline{AD_{TRG}}$ pin changes from "H" to "L" during operation, the operation at that point is cancelled and is restarted from step ①.

Figure 8.9.2 shows the conversion operation in the repeat sweep mode 0.

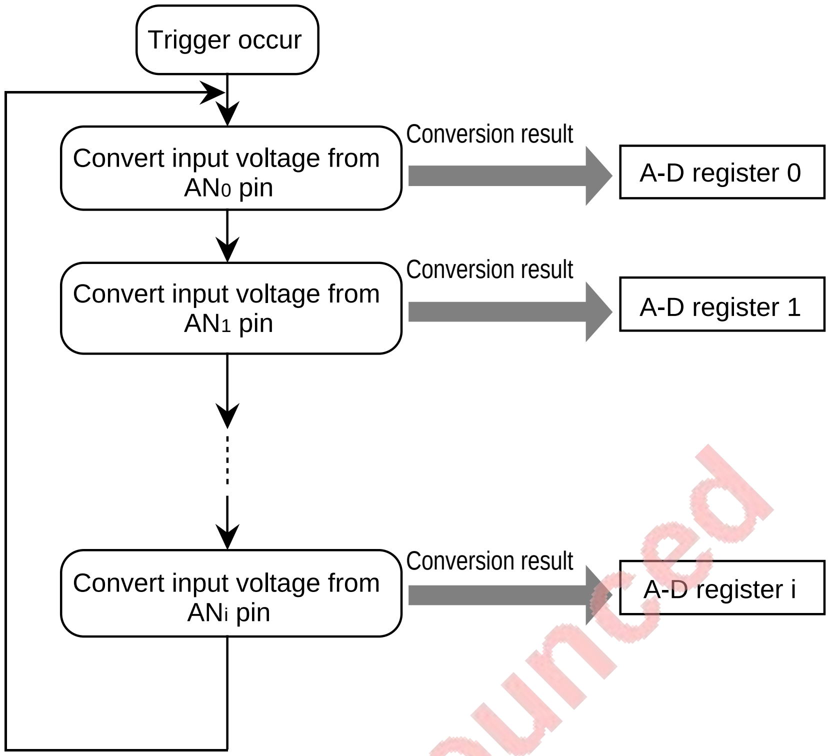


Fig. 8.9.2 Conversion operation in repeat sweep mode 0

8.10 Repeat sweep mode 1

In the repeat sweep mode 1, the operation for the input voltage from all selected analog input pins is performed repeatedly.
In this mode, analog input pins are separated into two groups according to the frequency of use. One is the group for more frequencies of use. The other is the group for few frequencies of use. First, the operations to all analog input pins in the group of more frequencies of use are performed. Next, the operation to one of analog input pins in the group of fewer frequencies of use is operated.
Figure 8.10.1 shows the analog input pin sweep operation. As shown in Figure 8.10.1, the pin to be executed in the group of fewer frequencies changes sequently.
In this mode, no A-D conversion interrupt request occurs. Additionally, the A-D conversion start bit (bit 6 at address $1E_{16}$) remains set to "1" until it is cleared to "0" by software, and the operation is performed repeatedly while the A-D conversion start bit is "1."

8.10.1 Settings for repeat sweep mode 1

Figure 8.10.2 shows an initial setting example of repeat sweep mode 1.
Select the analog input pins in the group of more frequencies of use by the A-D sweep pin select bits (bits 1 and 0 at address $1F_{16}$). Pins which are not selected by the A-D sweep pin select bits belong to the group of fewer frequencies of use.

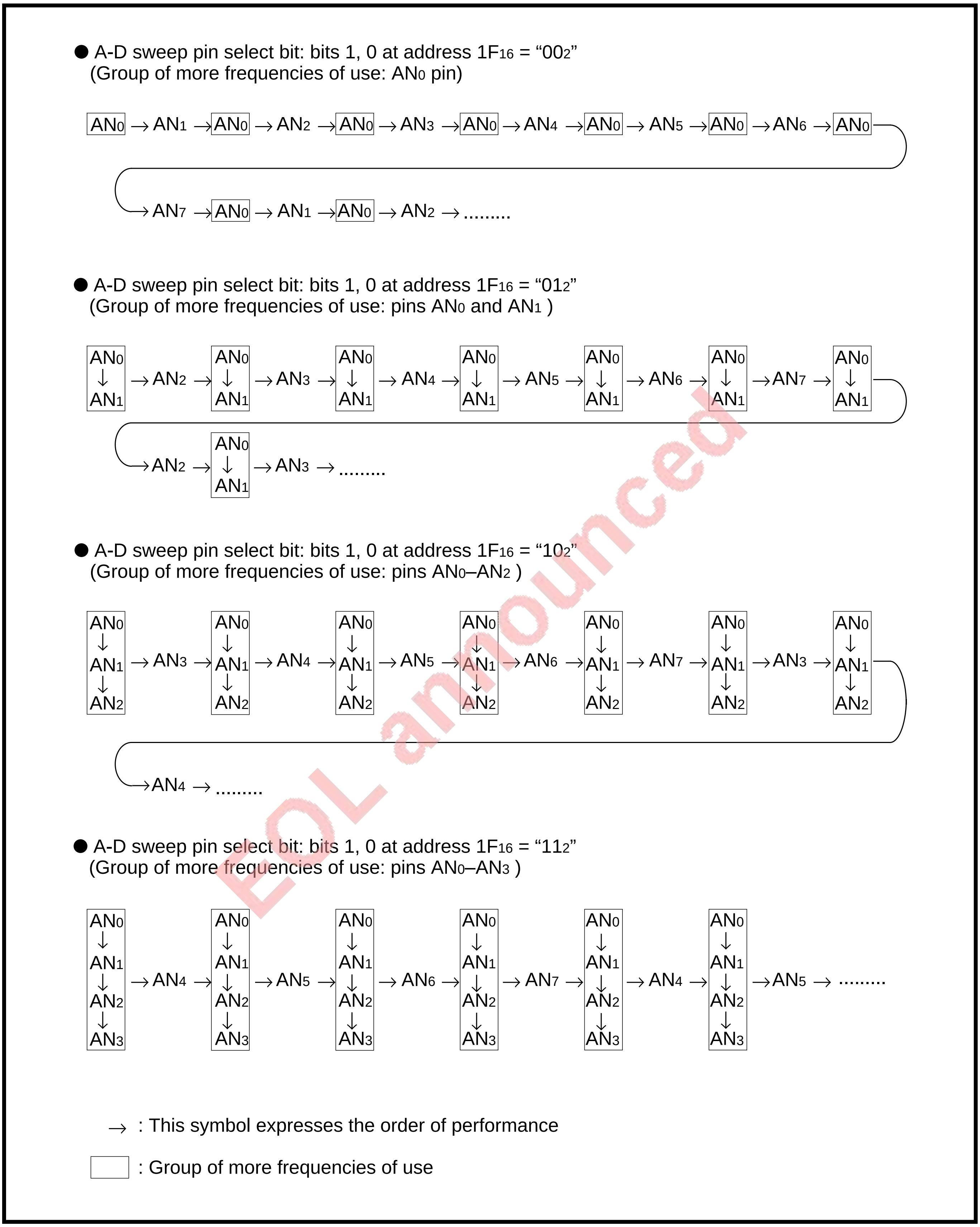


Fig. 8.10.1 Analog input pin sweep operation in repeat sweep mode 1

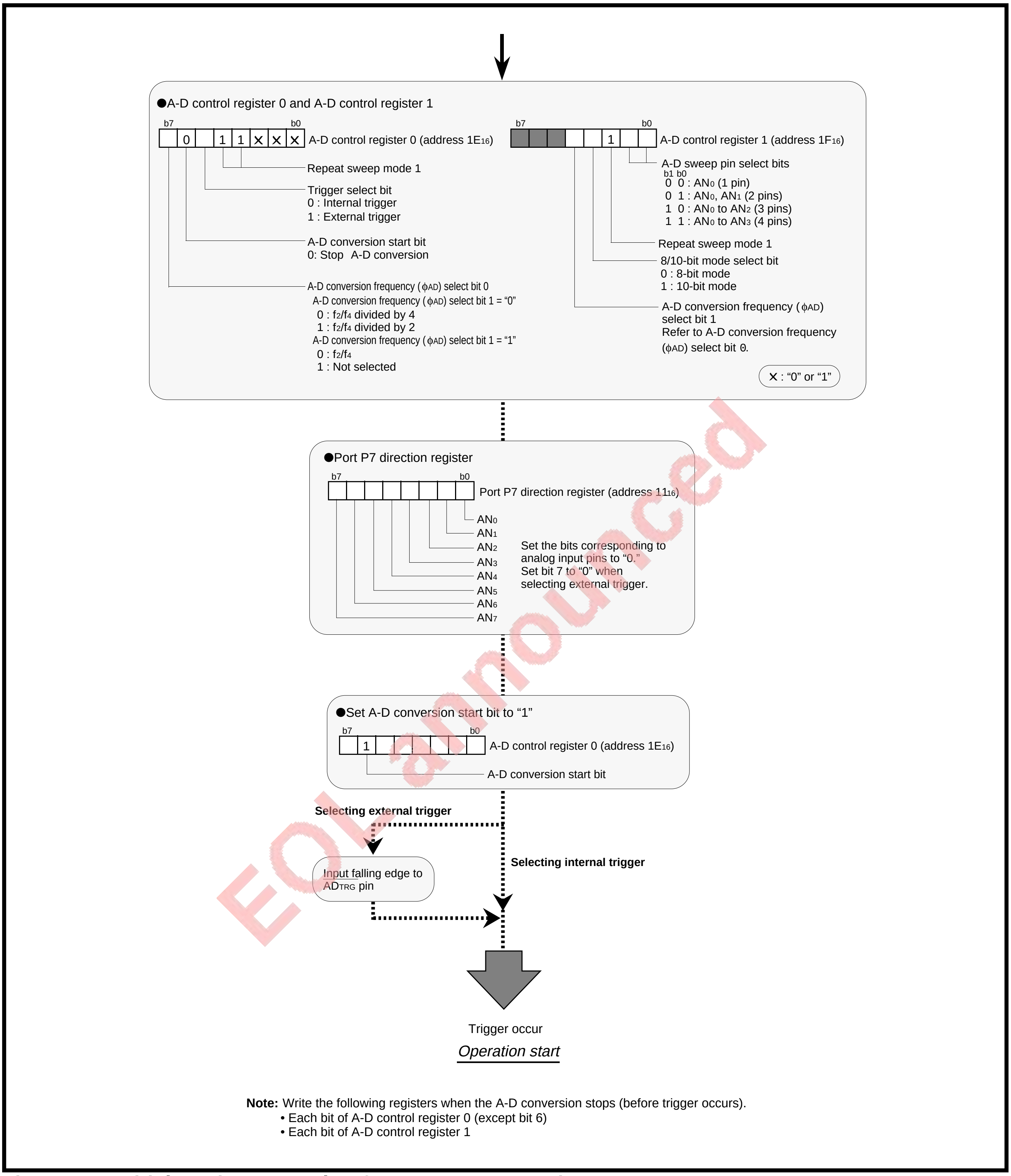


Fig. 8.10.2 Initial setting example of repeat sweep mode 1

8.10.2 Repeat sweep mode 1 operation description

(1) When an internal trigger is selected

① The A-D converter starts conversion for the input voltage from the AN_0 pin when the A-D conversion start bit is set to "1."

② The A-D conversion of the input voltage from the AN_0 pin is completed after 49 cycles of ϕ_{AD} in the 8-bit mode, or 59 cycles of ϕ_{AD} in the 10-bit mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register 0.

③ The operations to all analog input pins in the group of more frequencies of use are performed. The conversion result is transferred to the A-D register i each time each pin is converted.

④ The operation to one (refer to Figure 8.10.1) of analog input pins in the group of fewer frequencies of use is performed.

⑤ The operations to all analog input pins in the group of more frequencies of use are performed again.

⑥ The operation to another one, which is different from the one selected in step ④, of analog input pins in the group of fewer frequencies of use is performed. (Refer to Figure 8.10.1.)

⑦ The operation is performed repeatedly until the A-D conversion start bit is cleared to "0" by software.

(2) When an external trigger is selected

① The A-D converter starts conversion for the input voltage from the AN_0 pin when the input level to the $\overline{AD_{TRG}}$ pin changes from "H" to "L" while the A-D conversion start bit is set to "1."

② The A-D conversion of the input voltage from the AN_0 pin is completed after 49 cycles of ϕ_{AD} in the 8-bit mode, or 59 cycles of ϕ_{AD} in the 10-bit mode. Then, the contents of the successive approximation register (conversion result) are transferred to the A-D register 0.

③ The operations to all analog input pins in the group of more frequencies of use are performed. The conversion result is transferred to the A-D register i each time each pin is converted.

④ The operation to one (refer to Figure 8.10.1) of analog input pins in the group of fewer frequencies of use is performed.

⑤ The operations to all analog input pins in the group of more frequencies of use are performed again.

⑥ The operation to another one, which is different from the one selected in step ④, of analog input pins in the group of fewer frequencies of use is performed. (Refer to Figure 8.10.1.)

⑦ The operation is performed repeatedly until the A-D conversion start bit is cleared to "0" by software.

When the level of the $\overline{AD_{TRG}}$ pin changes from "H" to "L" during operation, the operation at that point is cancelled and is restarted from step ①.

[Precautions when using A-D converter]

1. Write to the following bits and registers before a trigger occurs (while the A-D converter stops operation).
- A-D control register 0 (except bit 6)
- A-D control register 1

2. When an external trigger is selected, the $AN_7/\overline{AD_{TRG}}$ pin cannot be used as the analog input pin. It is because the $AN_7/\overline{AD_{TRG}}$ pin is not connected to the comparator. When an external trigger is selected and the AN_7 pin is selected as the analog input pin, the A-D converter is operated and the A-D register 7 contains an undefined value.

3. Refer to **"Appendix.8 Examples of noise immunity improvement"** when using the A-D converter.

MEMORANDUM

CHAPTER 9

WATCHDOG TIMER

WATCHDOG TIMER

This chapter describes Watchdog timer.
Watchdog timer has the following functions:

- Detection of a program runaway.
- Measurement of a certain time when oscillation starts owing to terminating Stop mode.
  (Refer to **"Chapter 10. STOP MODE."**)

9.1 Block description

Figure 9.1.1 shows the block diagram of the watchdog timer.

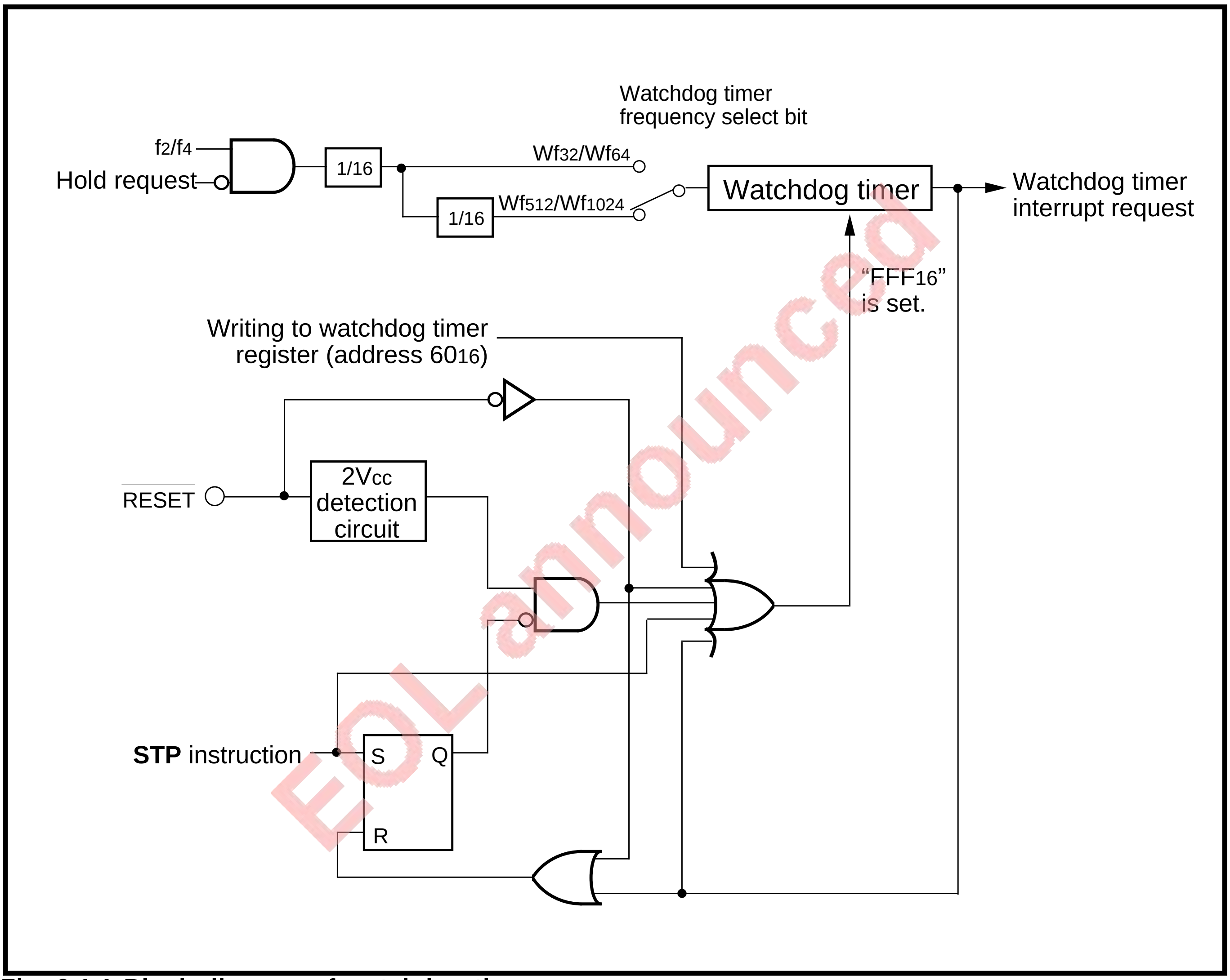


Fig. 9.1.1 Block diagram of watchdog timer

9.1.1 Watchdog timer

Watchdog timer is a 12-bit counter that down-counts the count source which is selected with the watchdog timer frequency select bit (bit 0 at address 61_{16}). A value “FFF_{16}” is automatically set in Watchdog timer in the cases listed below. An arbitrary value cannot be set to Watchdog timer.

- When dummy data is written to the watchdog timer register (Refer to Figure 9.1.2.)
- When the most significant bit of Watchdog timer becomes “0”
- When the **STP** instruction is executed (Refer to **“Chapter 10. STOP MODE.”**)
- At reset

b7 b0

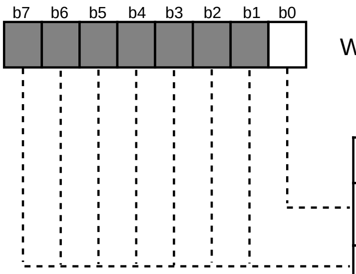
Watchdog timer register (Address 60_{16})

Bit	Functions	At reset	RW
7 to 0	Initializes the watchdog timer. When a dummy data is written to this register, the watchdog timer’s value is initialized to “FFF_{16}.” (Dummy data: 00_{16} to FF_{16})	Undefined	–

Fig. 9.1.2 Structure of watchdog timer register

9.1.2 Watchdog timer frequency select register

This is used to select the watchdog timer's count source. Figure 9.1.3 shows the structure of the watchdog timer frequency select register.



Watchdog timer frequency select register (Address 61_{16})

Bit	Bit name	Functions	At reset	RW
0	Watchdog timer frequency select bit	0 : Wf_{512}/Wf_{1024} 1 : Wf_{32}/Wf_{64}	0	RW
7 to 1	Nothing is assigned.		Undefined	–

Fig. 9.1.3 Structure of watchdog timer frequency select register

9.2 Operation description

The operation of Watchdog timer is described below.

9.2.1 Basic operation

① Watchdog timer starts down-counting from "FFF_{16}."
② When the Watchdog timer's most significant bit becomes "0" (counted 2048 times), the watchdog timer interrupt request occurs. (Refer to Table 9.2.1.)
③ When the interrupt request occurs at above ②, a value "FFF_{16}" is set to Watchdog timer.

The watchdog timer interrupt is a nonmaskable interrupt. When the watchdog timer interrupt request is accepted, the processor interrupt priority level (IPL) is set to "111_2."

Table 9.2.1 Occurrence interval of watchdog timer interrupt request

Watchdog timer frequency select bit	f(X_{IN}) = 25 MHz				f(X_{IN}) = 40 MHz	
	Clock source for peripheral devices select bit = "0"		Clock source for peripheral devices select bit = "1"		Clock source for peripheral devices select bit = "0"	
	Count source	Occurrence interval	Count source	Occurrence interval	Count source	Occurrence interval
0	Wf_{1024}	83.89 ms	Wf_{512}	41.94 ms	Wf_{1024}	52.43 ms
1	Wf_{64}	5.24 ms	Wf_{32}	2.62 ms	Wf_{64}	3.28 ms

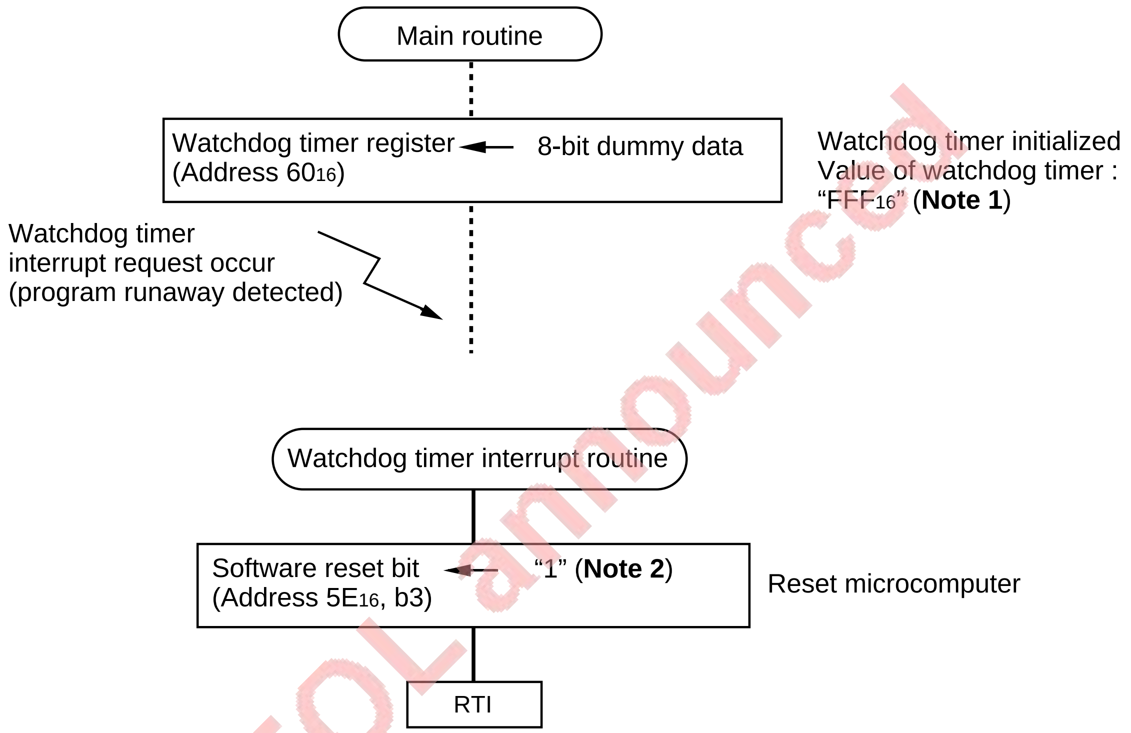
Clock source for peripheral devices select bit : bit 2 at address $5F_{16}$

9.2 Operation description

(1) Example of program runaway detection

Write to the address 60_{16} (watchdog timer register) before the most significant bit of Watchdog timer becomes "0." In the case that Watchdog timer is used to detect a program runaway, if writing to address 60_{16} is not performed owing to a program runaway, the watchdog timer interrupt request occurs when the most significant bit of Watchdog timer becomes "0." It means that a program runaway has occurred.

To reset the microcomputer after a program runaway, write "1" to the software reset bit (bit 3 at address $5E_{16}$) in the watchdog timer interrupt routine.



Notes 1: Initialize (write to address 60_{16}) Watchdog timer before the most significant bit of Watchdog timer becomes "0" (the watchdog timer interrupt request occurs).

2: When the program runaway occurs, values of the data bank register (DT) and direct page register (DPR) may be changed. When "1" is written to the software reset bit by the addressing mode using DT and DPR, set values to DT and DPR again.

Fig. 9.2.1 Example of program runaway detection by Watchdog timer

9.2.2 Operation in Stop mode

In Stop mode, Watchdog timer stops operating. Immediately after Stop mode is terminated, Watchdog timer operates as follows.

(1) When Stop mode is terminated by a hardware reset

Supply of the ϕ_{CPU} and ϕ_{BIU} starts immediately after Stop mode is terminated, and the microcomputer performs the "operation after a reset." (Refer to **"Chapter 13. RESET."**) The watchdog timer frequency select bit becomes "0," and Watchdog timer starts counting of Wf_{1024} from "FFF_{16}."

(2) When Stop mode is terminated by an interrupt request occurrence

Immediately after Stop mode is terminated, Watchdog timer starts counting of the count source Wf_{32}/Wf_{64} from "FFF_{16}." Supply of the ϕ_{CPU} and ϕ_{BIU} starts when the Watchdog timer's most significant bit becomes "0." (At this time, the watchdog timer interrupt request does not occur.)

Supply of the ϕ_{CPU} and ϕ_{BIU} starts immediately after Stop mode is terminated, and the microcomputer executes the routine of the interrupt which is used to terminate Stop mode. Watchdog timer restarts counting of the count source **(Note)** from "FFF_{16}."

Note: Clock Wf_{32}/Wf_{64} or Wf_{512}/Wf_{1024} which was counted just before executing the **STP** instruction.

9.2.3 Operation in Hold state

Watchdog timer stops operating in Hold state. When Hold state* is terminated, Watchdog timer restarts counting in the same state where it stopped operating.

Hold state*: Refer to section **"12.4 Hold function."**

9.3 Precautions when using watchdog timer

1. When a dummy data is written to address 60_{16} with the 16-bit data length, writing to address 61_{16} is simultaneously performed. Accordingly, when the user does not want to change a value of the watchdog timer frequency select bit (bit 0 at address 61_{16}), write the previous value to the bit simultaneously with writing to address 60_{16}.

2. When the **STP** instruction (refer to **"Chapter 10. STOP MODE"**) is executed, Watchdog timer stops. When Watchdog timer is used to detect the program runaway, select "**STP** instruction disable" with mask option.

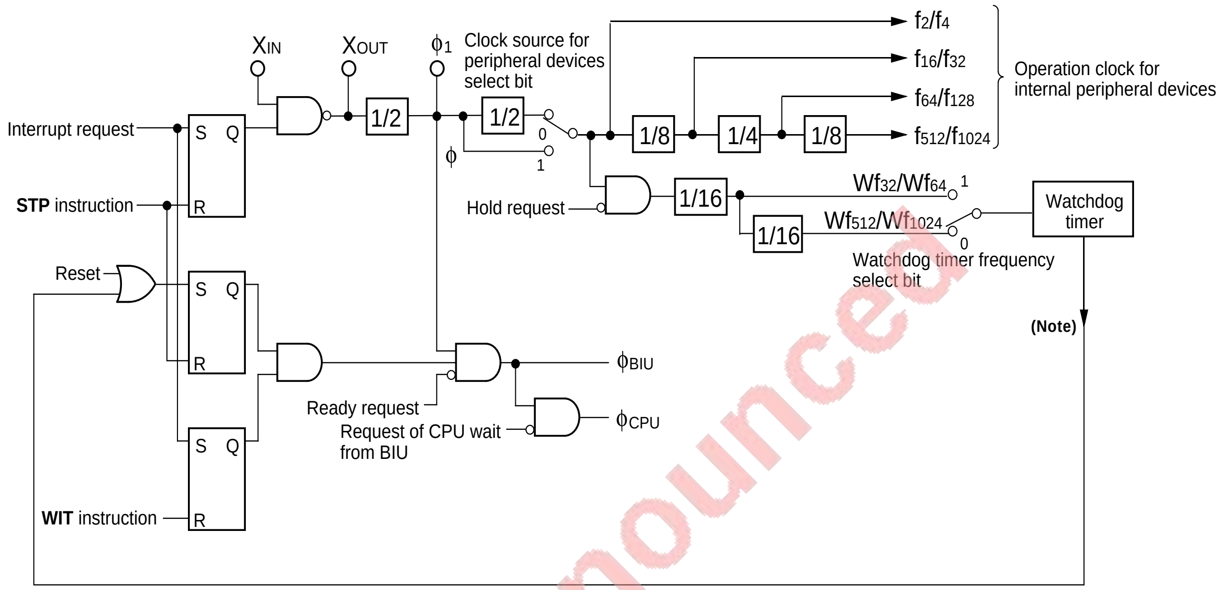
CHAPTER 10

STOP MODE

This chapter describes Stop mode.
Stop mode is used to stop oscillation when there is no need to operate the central processing unit (CPU). The microcomputer enters Stop mode when the **STP** instruction is executed.
Stop mode can be terminated by an interrupt request occurrence or the hardware reset.

10.1 Clock generating circuit

Figure 10.1.1 shows the clock generating circuit.



Clock source for peripheral devices select bit : Bit 2 at address 5F16
Watchdog timer frequency select bit : Bit 0 at address 6116

CPU : Central processing unit
BIU : Bus interface unit

Note: This is the signal generated when the watchdog timer's most significant bit becomes "0."

Fig. 10.1.1 Clock generating circuit

10.2 Operation description

When the **STP** instruction is executed, the oscillator stops oscillating. This state is called "Stop mode." In Stop mode, the contents of the internal RAM can be retained intact when the Vcc, power source voltage, is 2 V or more. Additionally, the microcomputer's power consumption is reduced. It is because the CPU and all internal peripheral devices using clocks f_2/f_4 to f_{512}/f_{1024} stop the operation.
Table 10.2.1 lists the microcomputer state and operation in and after Stop mode.

Table 10.2.1 Microcomputer state and operation in and after Stop mode

Item			State and Operation
State in Stop mode	Oscillation		Stopped
	ϕ_{CPU}, ϕ_{BIU}, ϕ, clock ϕ_1, f_2/f_4 to f_{512}/f_{1024}, Wf_{32}/Wf_{64}, Wf_{512}/Wf_{1024}		
	Internal peripheral devices	Timer A	Operating enabled only in event counter mode
		Timer B	
		Serial I/O	Operating enabled only when selecting external clock
		A-D converter	Stopped
		Watchdog timer	
		Pins	Retains the same state in which the **STP** instruction was executed
Operation after terminating Stop mode	By interrupt request occurrence		Supply of ϕ_{CPU} and ϕ_{BIU} starts after a certain time measured by watchdog timer has passed.
	By hardware reset		Operates in the same way as hardware reset

10.2 Operation description

10.2.1 Termination by interrupt request occurrence

When terminating Stop mode by interrupt request occurrence, instructions are executed after a certain time measured by the watchdog timer has passed.

① When an interrupt request occurs, the oscillator starts oscillating. Simultaneously, supply of ϕ, clock ϕ_1, f_2/f_4 to f_{512}/f_{1024}, Wf_{32}/Wf_{64}, and Wf_{512}/Wf_{1024} starts.
② The watchdog timer starts counting owing to the oscillation start. The watchdog timer counts Wf_{32}/Wf_{64}.
③ When the watchdog timer's MSB becomes "0," supply of ϕ_{CPU}, ϕ_{BIU} starts. At the same time, the watchdog timer's count source returns to Wf_{32}/Wf_{64} or Wf_{512}/Wf_{1024} that is selected by the watchdog timer frequency select bit (bit 0 at address 61_{16}).
④ The interrupt request which occurs in ① is accepted.

Table 10.2.2 lists the interrupts used to terminate Stop mode.

Table 10.2.2 Interrupts used to terminate Stop mode

Interrupt	Conditions for using each function to generate interrupt request
$\overline{INTi}$ interrupt (i = 0 to 2)	————
Timer Ai interrupt (i = 0 to 4)	Enabled in event counter mode
Timer Bi interrupt (i = 0 to 2)	
UARTi transmit interrupt (i = 0, 1)	Enabled when selecting external clock
UARTi receive interrupt (i = 0, 1)	

Notes 1: Since the oscillator has stopped oscillating, each function does not work unless they are operated under the above condition. Also, the A-D converter does not work.
2: Since the oscillator has stopped oscillating, no interrupts other than those above can be used.
3: Refer to **"Chapter 4. INTERRUPT"** and the description of each internal peripheral device for details about each interrupt.

Before executing the **STP** instruction, enable interrupts used to terminate Stop mode.
In addition, the interrupt priority level of the interrupt used to terminate Stop mode must be higher than the processor interrupt priority level (IPL) of the routine where the **STP** instruction is executed. When multiple interrupts in Table 10.2.2 are enabled, Stop mode is terminated by the first interrupt request.
There is possibility that all interrupt requests occur after the oscillation starts in ① and until supply of ϕ_{CPU} and ϕ_{BIU} starts in ③. The interrupt requests which occur during this time are accepted in order of priority **(Note)** after the watchdog timer's MSB becomes "0."
For interrupts not to be accepted, set their interrupt priority levels to level 0 (interrupt disabled) before executing the **STP** instruction.

Note : The interrupt request which has the highest priority is accepted first.

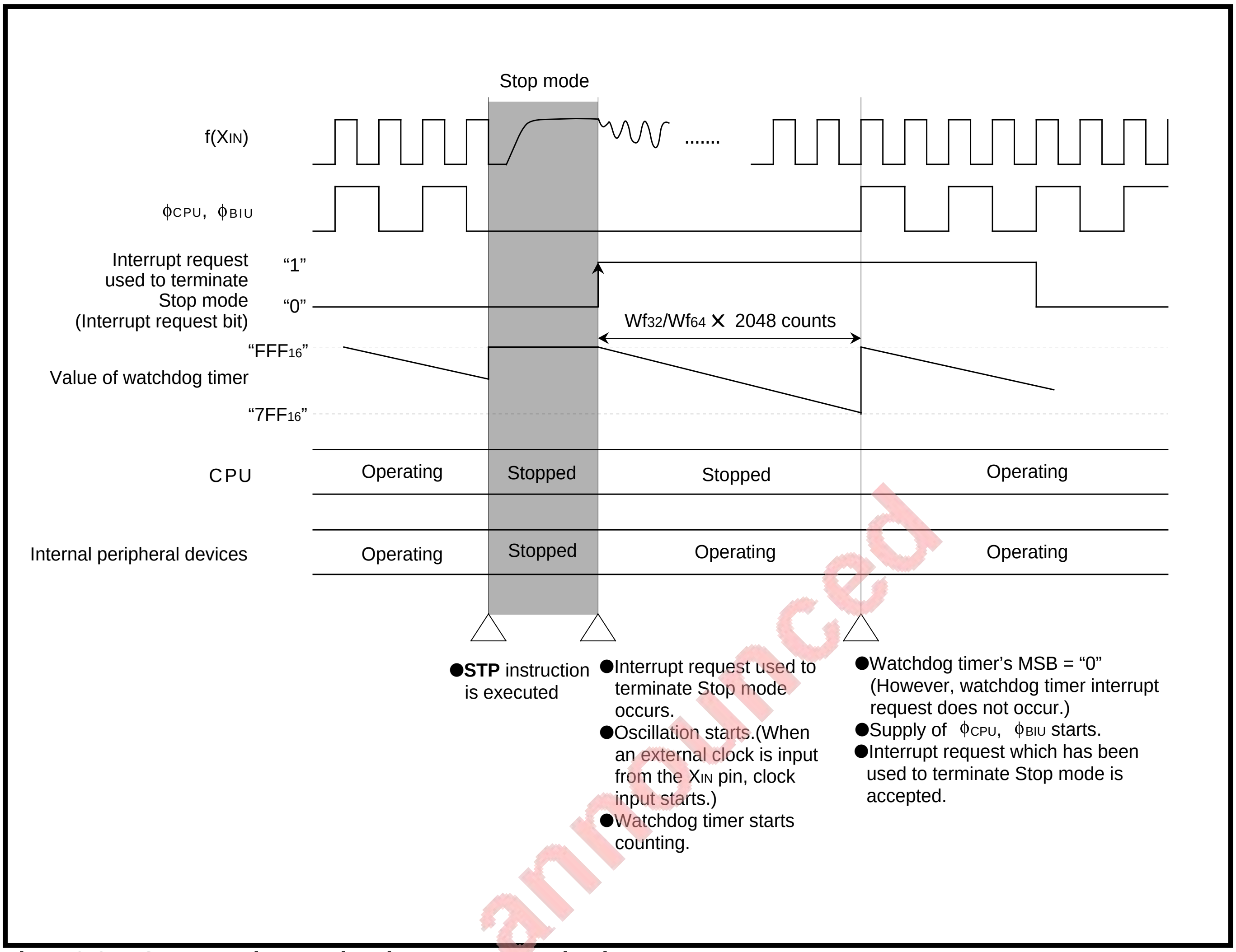


Fig. 10.2.1 Stop mode terminating sequence by interrupt request occurrence

10.2.2 Termination by hardware reset

Supply "L" level to the RESET pin by using the external circuit until the oscillation of the oscillator is stabilized.

The CPU and the SFR area are initialized in the same way as the system reset. However, the internal RAM area retains the same contents as that before executing the **STP** instruction. The termination sequence is the same as the internal processing sequence which is performed after a reset.

To determine whether a hardware reset was performed to terminate Stop mode or a system reset was performed, use software after a reset.

Refer to **"Chapter 13. RESET"** for details about a reset.

10.3 Precautions for Stop mode

1. When using the **STP** instruction with the mask ROM version, select "**STP** instruction enable" with the **STP** instruction option on the MASK ROM ORDER CONFIRMATION FORM.
The **STP** instruction is always enabled in the built-in PROM version and the flash memory version.

2. When executing the **STP** instruction after writing to the internal area or an external area, the three **NOP** instructions must be inserted to complete the write operation before the **STP** instruction is executed.

```
STA  A, XXXX ;  Writing instruction
NOP          ;  NOP instruction insertion
NOP          ;
NOP          ;
STP          ;  STP instruction
```

Fig. 10.3.1 NOP instruction insertion example

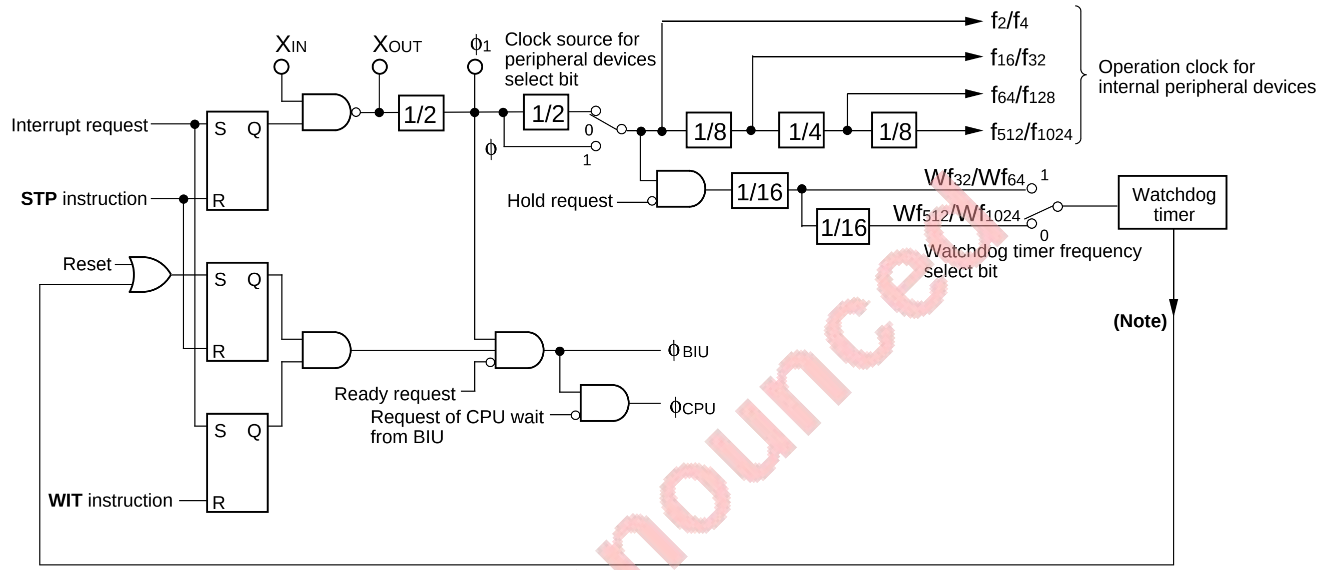
CHAPTER 11

WAIT MODE

This chapter describes Wait mode.
Wait mode is used to stop ϕ_{CPU} and ϕ_{BIU} when there is no need to operate the central processing unit (CPU).
The microcomputer enters Wait mode when the **WIT** instruction is executed.
Wait mode can be terminated by an interrupt request occurrence or the hardware reset.

11.1 Clock generating circuit

Figure 11.1.1 shows the clock generating circuit.



Clock source for peripheral devices select bit : Bit 2 at address $5F_{16}$
Watchdog timer frequency select bit : Bit 0 at address 61_{16}

CPU : Central processing unit
BIU : Bus interface unit

Note : This is the signal generated when the watchdog timer's most significant bit becomes "0."

Fig. 11.1.1 Clock generating circuit

11.2 Operation description

When the **WIT** instruction is executed, ϕ_{CPU} and ϕ_{BIU} stop. The oscillator’s oscillation is not stopped. This state is called “Wait mode.”
In Wait mode, the microcomputer’s power consumption is reduced though the Vcc, power source voltage, is maintained.
Table 11.2.1 lists the microcomputer state and operation in and after Wait mode.

Table 11.2.1 Microcomputer state and operation in and after Wait mode

Item			State and Operation
State in Wait mode	Oscillation		Operating
	ϕ_{CPU}, ϕ_{BIU}		Stopped
	Clock ϕ, ϕ_1, f_2/f_4 to f_{512}/f_{1024}, Wf_{32}/Wf_{64}, Wf_{512}/Wf_{1024}		Operating
	Internal peripheral devices	Timer A	Operating
		Timer B	
		Serial I/O	
		A-D converter	
		Watchdog timer	
		Pins	Retains the same state in which the **WIT** instruction was executed
Operation after terminating Wait mode	By interrupt request occurrence		Supply of ϕ_{CPU} and ϕ_{BIU} starts just after the termination.
	By hardware reset		Operates in the same way as hardware reset

11.2 Operation description

11.2.1 Termination by interrupt request occurrence

① When an interrupt request occurs, supply of clock ϕ_{CPU} and ϕ_{BIU} starts.
② The interrupt request which occurs in ① is accepted.
Table 11.2.2 shows the interrupts used to terminate Wait mode.
The occurrence of the watchdog timer interrupt request also terminates Wait mode.

Table 11.2.2 Interrupts used to terminate Wait mode

Interrupt
•$\overline{INT_i}$ interrupt (i = 0 to 2)
•Timer Ai interrupt (i = 0 to 4)
•Timer Bi interrupt (i = 0 to 2)
•UARTi transmit interrupt (i = 0, 1)
•UARTi receive interrupt (i = 0, 1)
•A-D converter interrupt

Note : Refer to "**Chapter 4. INTERRUPTS**" and each functional description about interrupts.

Before executing the **WIT** instruction, enable interrupts used to terminate Wait mode.
In addition, the interrupt priority level of the interrupt used to terminate Wait mode must be higher than the processor interrupt priority level (IPL) of the routine where the **WIT** instruction is executed. When the multiple interrupts in Table 11.2.2 are enabled, Wait mode is terminated by the first interrupt request.

11.2.2 Termination by hardware reset

The CPU and the SFR area are initialized in the same way as a system reset. However, the internal RAM area retains the same contents as that before executing the **WIT** instruction. The termination sequence is the same as the internal processing sequence which is performed after a reset.
To determine whether a hardware reset was performed to terminate Wait mode or a system reset was performed, use software after a reset.
Refer to **"Chapter 13. RESET"** for details about a reset.

11.3 Precautions for Wait mode

When executing the **WIT** instruction after writing to the internal area or an external area, the three **NOP** instructions must be inserted to complete the write operation before the **WIT** instruction is executed.

```
STA  A, XXXX ;  Writing instruction
NOP          ;  NOP instruction insertion
NOP          ;
NOP          ;
WIT          ;  WIT instruction
```

Fig. 11.3.1 NOP instruction insertion example

MEMORANDUM

CHAPTER 12

CONNECTION WITH EXTERNAL DEVICES

This chapter describes functions to connect devices externally.

12.1 Signals required for accessing external devices

The functions and operation of the signals which are required for accessing external devices are described below.

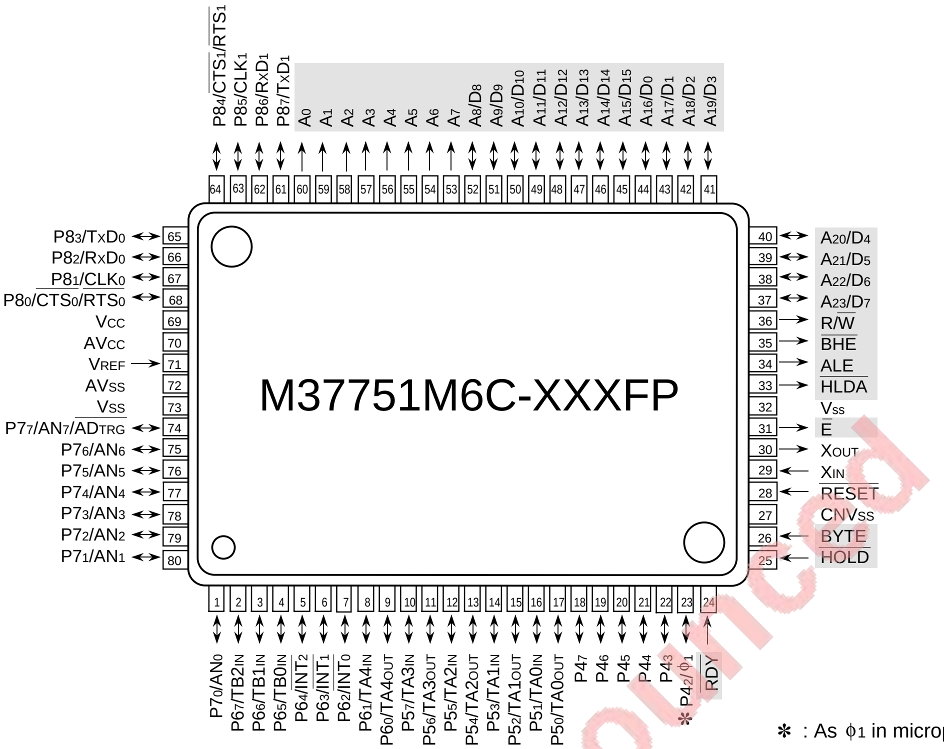
When connecting an external device that requires a long access time, refer to sections **"12.2 Bus cycle," "12.3 Ready function,"** and **"12.4 Hold function,"** as well as this section.

12.1.1 Descriptions of signals

When an external device is connected, operate the microcomputer in the memory expansion or microprocessor mode. (Refer to section **"2.5 Processor modes."**) In these modes, pins P0 to P4 and the $\overline{E}$ pin function as I/O pins for the signals required for accessing external devices.

Figure 12.1.1 shows the pin configuration in the memory expansion and microprocessor modes. Table 12.1.1 lists the functions of pins P0 to P4 and the $\overline{E}$ pin in the memory expansion and the microprocessor modes.

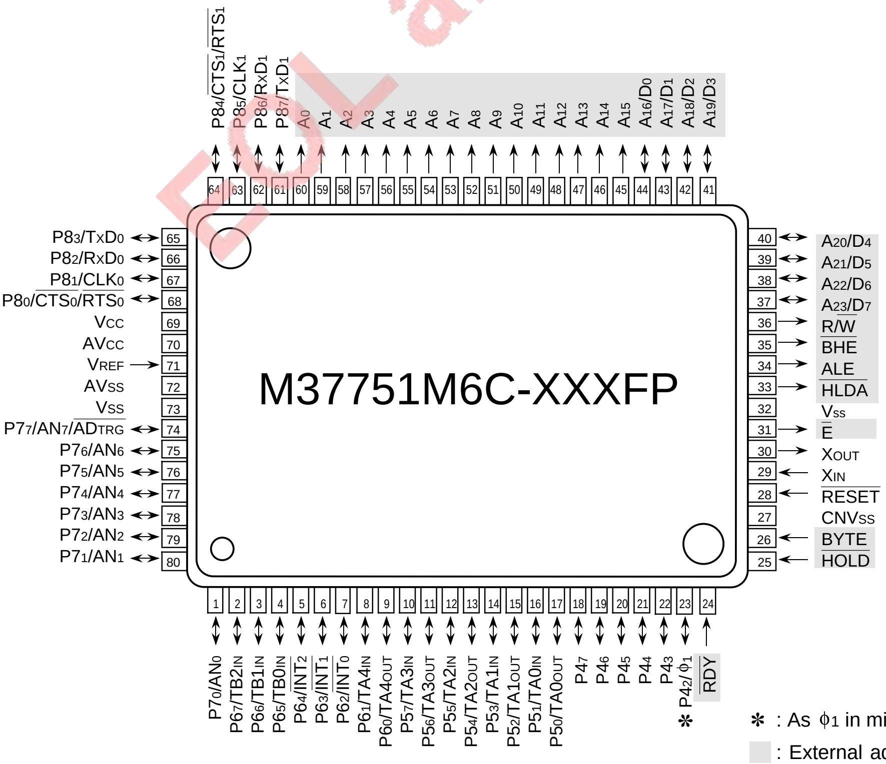
●External data bus width = 16 bits (BYTE = "L")



* : As ϕ_1 in microprocessor mode

: External address bus, external data bus, bus control signal

●External data bus width = 8 bits (BYTE = "H")



* : As ϕ_1 in microprocessor mode

: External address bus, external data bus, bus control signal

Fig. 12.1.1 Pin configuration in memory expansion and microprocessor modes (top view)

CONNECTION WITH EXTERNAL DEVICES

12.1 Signals required for accessing external devices

Table 12.1.1 Functions of pins P0 to P4 and $\overline{E}$ pin in memory expansion and microprocessor modes

Pin \ External data bus width	16 bits (BYTE = "L")	8 bits (BYTE = "H")
A_7 to A_0 (P0)	A_7 — A_0: A_7 — A_0	
A_{15}/D_{15} to A_8/D_8 (P1)	A_{15}/D_{15} — A_8/D_8: A_{15} — A_8, D(odd) D(odd): Data at odd address	A_{15} — A_8: A_{15} — A_8
A_{23}/D_7 to A_{16}/D_0 (P2)	A_{23}/D_7 — A_{16}/D_0: A_{23} — A_{16}, D(even) D(even): Data at even address	A_{23}/D_7 — A_{16}/D_0: A_{23} — A_{16}, D D: Data
$\overline{HLDA}$ ($P3_3$) ALE ($P3_2$) $\overline{BHE}$ ($P3_1$) $R/\overline{W}$ ($P3_0$)	$\overline{HLDA}$: $\overline{HLDA}$ ALE: ALE $\overline{BHE}$: $\overline{BHE}$ $R/\overline{W}$: $R/\overline{W}$	
$P4_7$ to $P4_3$ ϕ_1 ($P4_2$) $\overline{RDY}$ ($P4_1$) $\overline{HOLD}$ ($P4_0$)	$P4_7$ — $P4_3$: P P: Functions as a programmable I/O port. ϕ_1: ϕ_1 **(Note 1)** $\overline{RDY}$: $\overline{RDY}$ $\overline{HOLD}$: $\overline{HOLD}$	
$\overline{E}$	$\overline{E}$: $\overline{E}$	

Notes 1: In the memory expansion mode, this pin functions as a programmable I/O port and can be programmed as the clock ϕ_1 output pin by software.

2: This table shows the pins' functions. Refer to the following about the input/output timing of each signal: "**12.1.2 Operation of bus interface unit (BIU)**"; "**12.2 Bus cycle**"; "**12.3 Ready function**"; "**12.4 Hold function**"; "**Chapter 15. ELECTRICAL CHARACTERISTICS**."

(1) External bus (A_0 to A_7, A_8/D_8 to A_{15}/D_{15}, A_{16}/D_0 to A_{23}/D_7)

External areas are specified by the address (A_0 to A_{23}) output. Figure 12.1.2 shows the external area. Pins A_8 to A_{23} of the external address bus and pins D_0 to D_{15} of the external data bus are assigned to the same pins. When the BYTE pin level, described later, is "L" (i.e., external data bus width is 16 bits), the A_8/D_8 to A_{15}/D_{15} and A_{16}/D_0 to A_{23}/D_7 pins perform address output and data input/output with time-sharing. When the BYTE pin level is "H" (i.e., external data bus width is 8 bits), the A_{16}/D_0 to A_{23}/D_7 pins perform address output and data input/output with time-sharing, and pins A_8 to A_{15} output addresses.

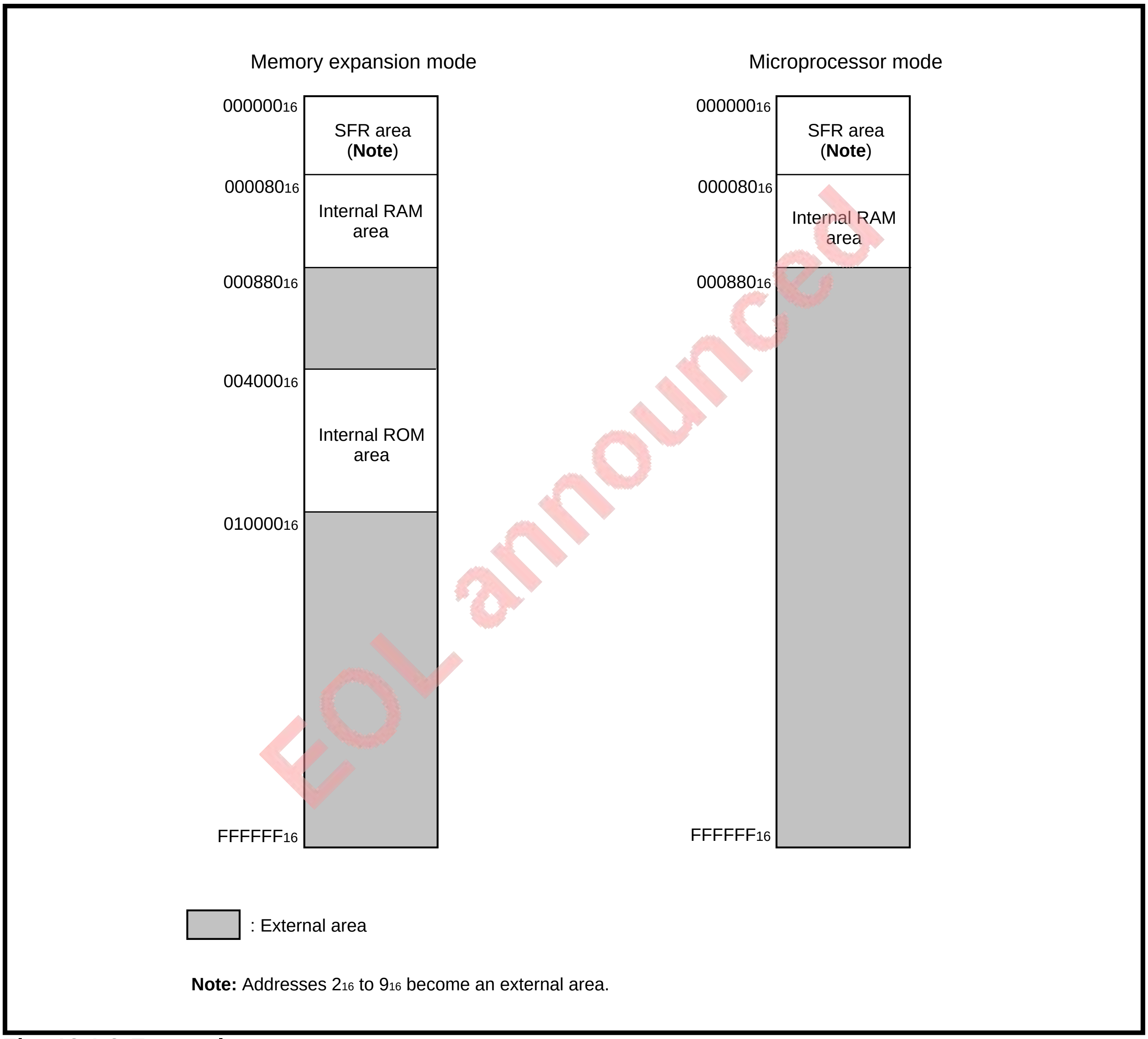


Fig. 12.1.2 External area

(2) External data bus width switching signal (BYTE pin level)

This signal is used to select the external data bus width between 8 bits and 16 bits. When this signal level is “L,” the external data bus width is 16 bits; when the level is “H,” the bus width is 8 bits (refer to Table 12.1.1.)

Fix this signal to either “H” or “L” level.

This signal is valid only for the external areas. When accessing the internal areas, the data bus width is always 16 bits.

(3) Enable signal ($\overline{E}$)

This signal becomes “L” level while reading or writing data to and from the data bus. (See Table 12.1.2.)

(4) Read/Write signal (R/$\overline{W}$)

This signal indicates the state of the data bus. This signal becomes “L” level while writing to the data bus. Table 12.1.2 lists the state of the data bus indicated with the $\overline{E}$ and R/$\overline{W}$ signals.

Table 12.1.2 State of data bus indicated with $\overline{E}$ and R/$\overline{W}$ signals

$\overline{E}$	R/$\overline{W}$	State of data bus
H	H	Not used
	L	
L	H	Read data
	L	Write data

(5) Byte high enable signal ($\overline{BHE}$)

This signal indicates the access to an odd address. This signal becomes “L” level when accessing an only odd address or when simultaneously accessing odd and even addresses.

This signal is used to connect memories or I/O devices of which data bus width is 8 bits when the external data bus width is 16 bits.

Table 12.1.3 lists levels of the external address bus A_0 and the $\overline{BHE}$ signal and access addresses.

Table 12.1.3 Levels of A_0 and $\overline{BHE}$ signal and access addresses

Access address	Even and odd addresses (Simultaneous 2-byte access)	Even address (1-byte access)	Odd address (1-byte access)
A_0	L	L	H
$\overline{BHE}$	L	H	L

(6) Address latch enable signal (ALE)

This signal is used to obtain the address from the multiplexed signal of address and data that is input and output to and from the A_8/D_8 to A_{15}/D_{15} and A_{16}/D_0 to A_{23}/D_7 pins. Make sure that when this signal is “H,” latch the address and simultaneously output the addresses. When this signal is “L,” retain the latched address.

(7) Ready function-related signal ($\overline{RDY}$)

This is the signal to use the Ready function. (Refer to section **“12.3 Ready function.”**)

(8) Hold function-related signals ($\overline{HOLD}$, $\overline{HLDA}$)

These are the signals to use the Hold function. (Refer to section **“12.4 Hold function.”**)

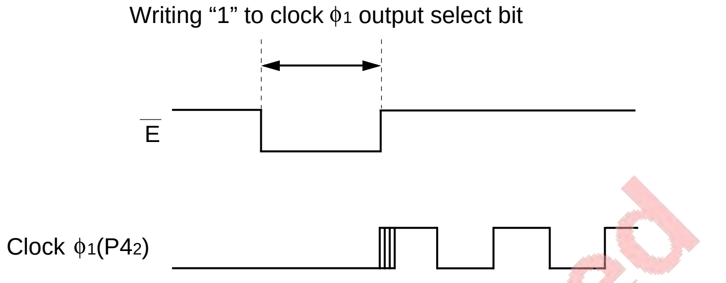
(9) Clock ϕ_1

This signal has the same period as ϕ.

In the memory expansion mode, this signal is output externally by setting the clock ϕ_1 output select bit (bit 7 at address $5E_{16}$) to “1.” Figure 12.1.3 shows the output start timing of clock ϕ_1.

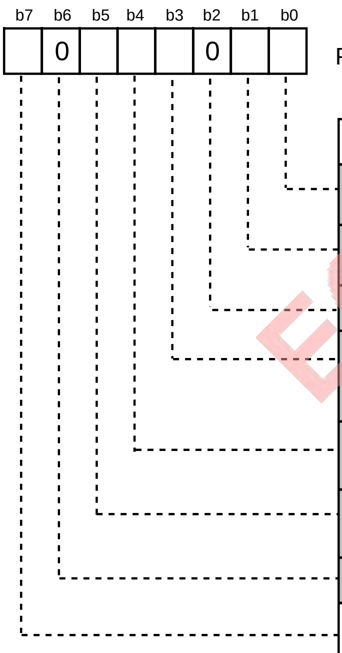
In the microprocessor mode, this signal is always output externally.

Note: Even in the single-chip mode, the clock ϕ_1 can be output externally. This signal is output externally by setting the clock ϕ_1 output select bit to “1” just as in the memory expansion mode.



Notes 1: The 1st cycle of clock ϕ_1 may be shortened; indicated by ⎍⎍⎍.
2: This applies when writing to clock ϕ_1 output select bit while $P4_2$ pin is outputting “L” level.

Fig. 12.1.3 Output start timing of clock ϕ_1



Processor mode register 0 (Address $5E_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Memory expansion mode 1 0 : Microprocessor mode 1 1 : Not selected	0	RW
1			0 **(Note 1)**	RW
2	Fix this bit to “0.”		0	RW
3	Software reset bit	The microcomputer is reset by writing “1” to this bit. The value is “0” at reading.	0	WO
4	Interrupt priority detection time select bits	b5 b4 0 0 : 7 cycles of ϕ 0 1 : 4 cycles of ϕ 1 0 : 2 cycles of ϕ 1 1 : Not selected	0	RW
5			0	RW
6	Fix this bit to “0.”		0	RW
7	Clock ϕ_1 output select bit **(Note 2)**	0 : Clock ϕ_1 output disabled ($P4_2$ functions as a programmable I/O port.) 1 : Clock ϕ_1 output enabled ($P4_2$ functions as a clock ϕ_1 output pin.)	0	RW

Notes 1: While supplying the Vcc level to the CNVss pin, this bit becomes “1.” (Fixed to “1.”)
2: This bit is ignored in the microprocessor mode. (It may be either “0” or “1.”)

(shaded) : Bits 0 to 6 are not used for setting of clock ϕ_1 output.

Fig. 12.1.4 Structure of processor mode register

12.1.2 Operation of bus interface unit (BIU)

Figures 12.1.5 and 12.1.6 show the examples of operating waveforms of the signals input and output to /from externals when accessing external devices. The following explains these waveforms compared with the basic operating waveform (refer to section **"2.2.3 Operation of bus interface unit (BIU)."**)

(1) When fetching instructions into instruction queue buffer

① When the instruction which is next fetched is located at an even address in the 16-bit external data bus width, the BIU fetches 2 bytes at a time with the waveform (a). When in the 8-bit external data bus width, the BIU fetches only 1 byte with the first half of waveform (e).

② When the instruction which is next fetched is located at an odd address in the 16-bit external data bus width, the BIU fetches only 1 byte with the waveform (d). When in the 8-bit external data bus width, the BIU fetches only 1 byte with the first half of waveform (f).

When a branch to an odd address is caused by a branch instruction and others in the 16-bit external data bus width, the BIU first fetches 1 byte in waveform (d), and after that, fetches each two bytes at a time in waveform (a).

(2) When reading or writing data to and from memory•I/O device

① When accessing 16-bit data which begins at an even address, waveform (a) or (e) is applied.

② When accessing 16-bit data which begins at an odd address, waveform (b) or (f) is applied.

③ When accessing 8-bit data at an even address, waveform (c) or the first half of (e) is applied.

④ When accessing 8-bit data at an odd address, waveform (d) or the first half of (f) is applied.

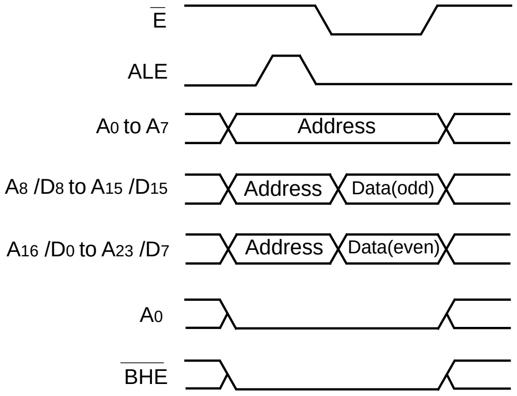
For instructions that are affected by the data length flag (m) and the index register length flag (x), operation ① or ② is applied when flag m or x = "0"; operation ③ or ④ is applied when flag m or x = "1."

The setup of flags m and x and the selection of the external data bus width do not affect each other.

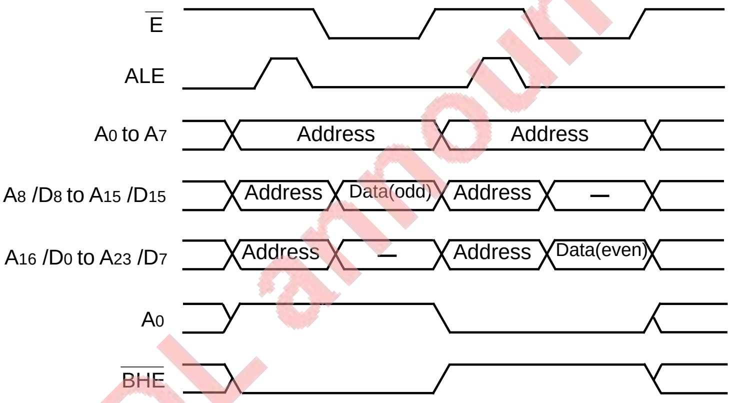
● External data bus width = 16 bits (BYTE = “L”)

<16-bit data access>

(a) Access from even address

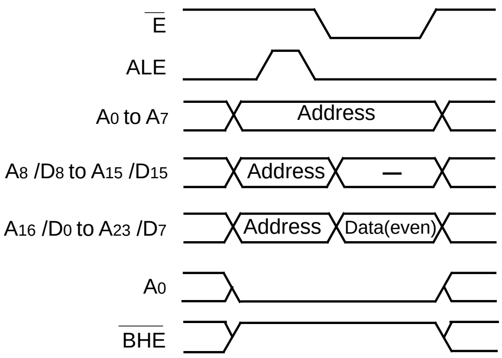


(b) Access from odd address



<8-bit data access>

(c) Access to even address



(d) Access to odd address

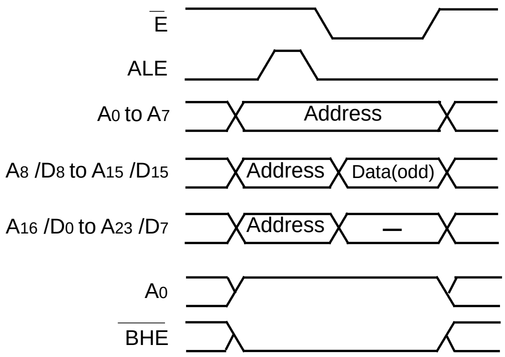
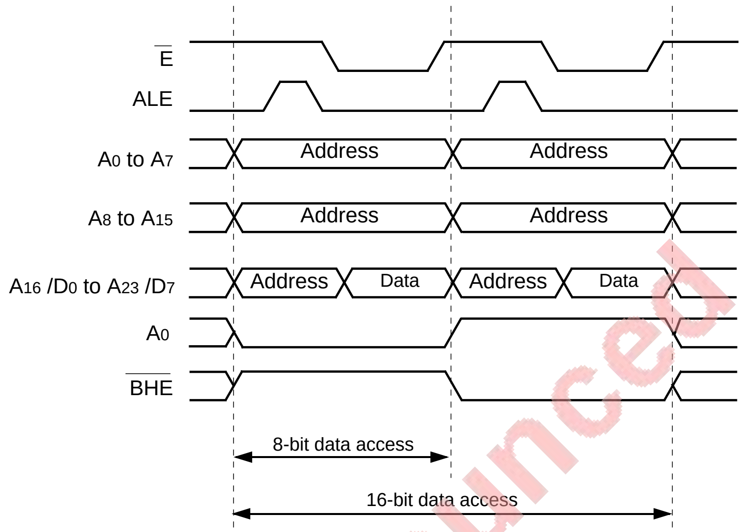


Fig. 12.1.5 Example of operating waveforms of signals input and output to/from externals (1)

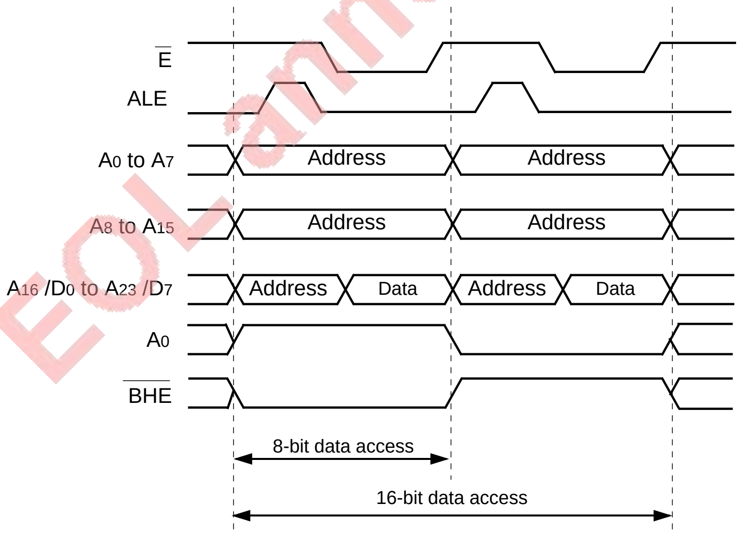
● External data bus width = 8 bits (BYTE = “H”)

<8/16-bit data access>

(e) Access from even address



(f) Access from odd address



Note: When accessing 16-bit data, 2 times of access are performed in the sequence of the low-order 8 bits and high-order 8 bits.

Fig. 12.1.6 Example of operating waveforms of signals input and output to/from externals (2)

12.2 Bus cycle

The bus cycle can be selected to make it easy to access the external devices which require a long access time. The bus cycle is selected with the bus cycle select bits (bits 4 and 5 at address $5F_{16}$).
The selectable bus cycle depends on the CPU running speed. The CPU running speed is selected with the CPU running speed select bit (bit 3 at address $5F_{16}$).
Table 12.2.1 lists the selection of CPU running speed and bus cycle. Figure 12.2.1 shows the structure of the processor mode register 1 (address $5F_{16}$). Table 12.2.2 lists each bus cycle.

The selection of bus cycle is valid only for external areas.
For the internal area, the access is performed with the fixed bus cycle.

Table 12.2.1 Selection of CPU running speed and bus cycle

Processor mode register 1 (address $5F_{16}$)			Access to internal area	Access to external area (**Note**)
b5	b4	b3		
1	1	1	2 ϕ access in low-speed running	2 ϕ access in low-speed running
1	0	1		3 ϕ access in low-speed running
0	1	1		4 ϕ access in low-speed running
1	0	0	High-speed running	3 ϕ access in high-speed running
0	1	0	RAM: 2 ϕ access	4 ϕ access in high-speed running
0	0	0	ROM, SFR: 3 ϕ access	5 ϕ access in high-speed running
1	1	0	Not selected	
0	0	1		

12.2 Bus cycle

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

Processor mode register 1 (Address $5F_{16}$)

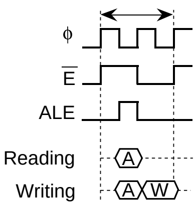
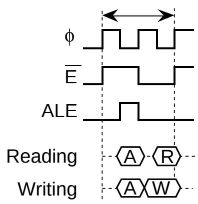
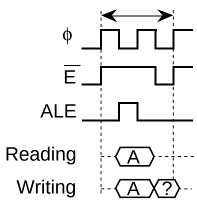
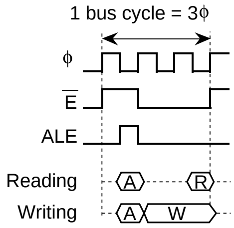
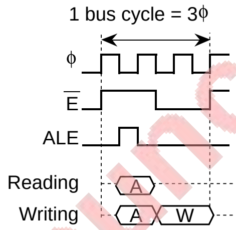
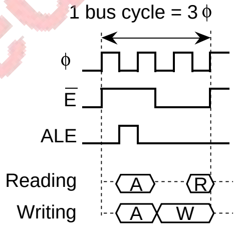
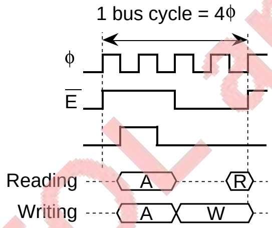
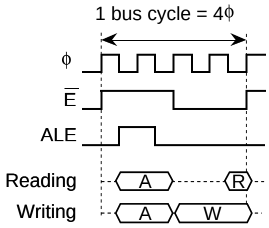
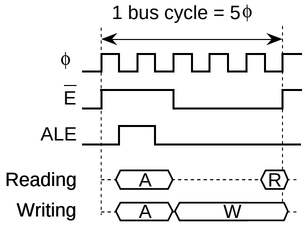
Bit	Bit name	Functions	At reset	RW
1, 0	Fix these bits to "0."		0	RW
2	Clock source for peripheral devices select bit **(Note)**	0 : φ divided by 2 1 : φ	0	RW
3	CPU running speed select bit **(Note)**	0 : High-speed running 1 : Low-speed running	0	RW
4	Bus cycle select bits	In high-speed running b5 b4 0 0 : 5φ access in high-speed running 0 1 : 4φ access in high-speed running 1 0 : 3φ access in high-speed running 1 1 : Not selected	0	RW
5		In low-speed running b5 b4 0 0 : Not selected 0 1 : 4φ access in low-speed running 1 0 : 3φ access in low-speed running 1 1 : 2φ access in low-speed running	0	RW
7, 6	Fix these bits to "0."		0	RW

Note: Fix this bit to "0" when $f(X_{IN})$ > 25 MHz.

(shaded) : Bits 0 to 2, 6 and 7 are not used to select the bus cycle.

Fig. 12.2.1 Structure of the processor mode register 1

Table 12.2.2 Bus cycle

Low-speed running [f (XIN) ≤ 25 MHz]		High-speed running [f (XIN) ≤ 40 MHz]	
Internal area access (Note)	External area access	Internal area access (Note)	External area access
2ϕ access in low-speed running 1 bus cycle = 2ϕ 	2ϕ access in low-speed running 1 bus cycle = 2ϕ 	2ϕ access in high-speed running (RAM) 1 bus cycle = 2ϕ 	
	3ϕ access in low-speed running 1 bus cycle = 3ϕ 	3ϕ access in high-speed running (ROM, SFR) 1 bus cycle = 3ϕ 	3ϕ access in high-speed running 1 bus cycle = 3 ϕ 
	4ϕ access in low-speed running 1 bus cycle = 4ϕ 		4ϕ access in high-speed running 1 bus cycle = 4ϕ 
			5ϕ access in high-speed running 1 bus cycle = 5ϕ 

Note : Signals when accessing an internal area means signals which are output from pins externally when accessing an internal area in the memory expansion mode.

A: Address **W**: Data to be written
R: Data to be read **?**: Undefined value

12.3 Ready function

Ready function provides the function to facilitate access to external devices that require a long access time. By supplying “L” level to the $\overline{RDY}$ pin in the memory expansion or microprocessor mode, the microcomputer enters Ready state and retains this state while the $\overline{RDY}$ pin is at “L” level. Table 12.3.1 lists the microcomputer’s state in Ready state.
In Ready state, the oscillator’s oscillation does not stop, so that the internal peripheral devices can operate.
<u>Ready function is valid for the internal and external areas.</u>

Table 12.3.1 Microcomputer’s state in Ready state

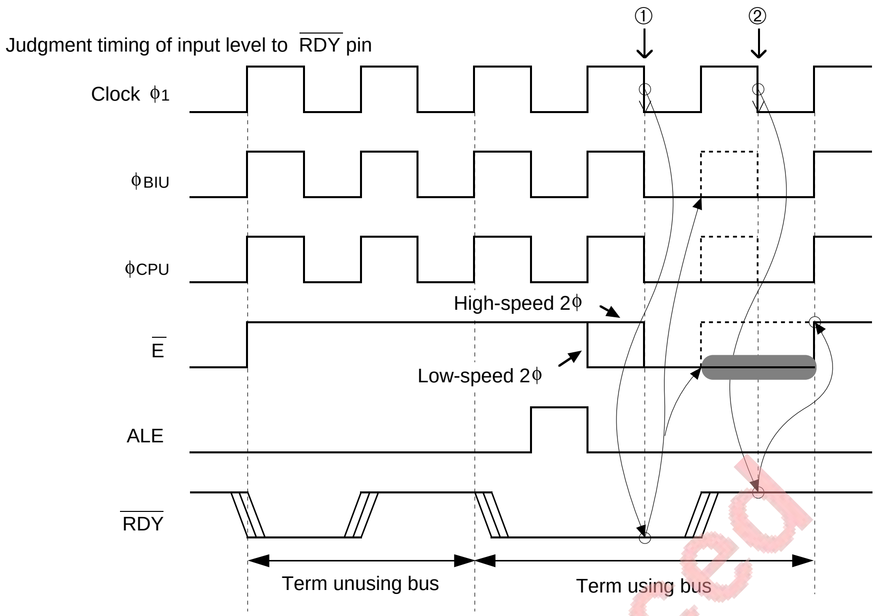
Item	State
Oscillation, ϕ	Operating
ϕ_{CPU}, ϕ_{BIU}, $\overline{E}$	Stopped at “L”
Pins A_0 to A_7, A_8/D_8 to A_{15}/D_{15}, A_{16}/D_0 to A_{23}/D_7, $R/\overline{W}$, $\overline{BHE}$, $\overline{HLDA}$, ALE	Retains the state when Ready request was accepted.
Pins $P4_3$ to $P4_7$, P5 to P8 **(Note)**	
$P4_2/\phi_1$	In the memory expansion mode: •When clock ϕ_1 output select bit* = “1,” this pin outputs clock ϕ_1. •When clock ϕ_1 output select bit = “0,” this pin retains the state when Ready request was accepted. In the microprocessor mode: •This pin outputs clock ϕ_1.
Watchdog timer	Operating

Clock ϕ_1 output select bit* : Bit 7 at address $5E_{16}$
Note: When this functions as a programmable I/O port.

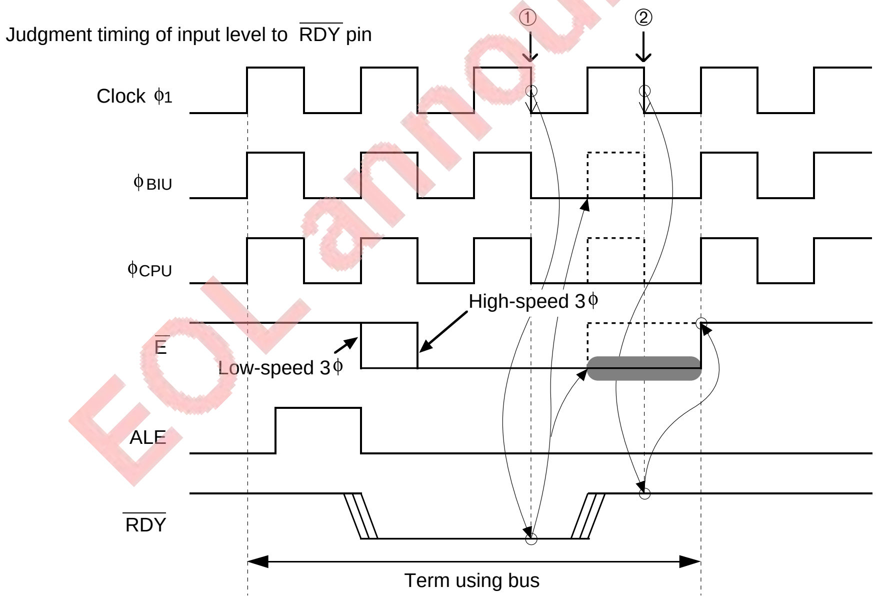
12.3.1 Operation description

The input level of the $\overline{RDY}$ pin is judged at the last falling of the clock ϕ_1 in each bus cycle. Then, when "L" level is detected, the microcomputer enters Ready state. (This is called acceptance of Ready request.)
In Ready state, the input level of the $\overline{RDY}$ pin is judged at every falling of the clock ϕ_1. Then, when "H" level is detected, the microcomputer terminates Ready state next rising of the clock ϕ_1.
Figures 12.3.1 and 12.3.2 show timing of acceptance of Ready request and termination of Ready state.
Refer also to section "**17.1 Memory expansion**" about usage of the Ready function.

● 2φ access in low-speed running, 2φ access in high-speed running



● 3φ access in low-speed running, 3φ access in high-speed running



① By accepting an Ready request, "L" level of E signal stops for 1 cycle with the clock φ1, indicated by (shaded bar), and clocks φBIU and φCPU stop at "L" level.
② Ready state is terminated.

✻ Input level to the RDY pin is not judged during the term unusing the bus or before the condition above ①.

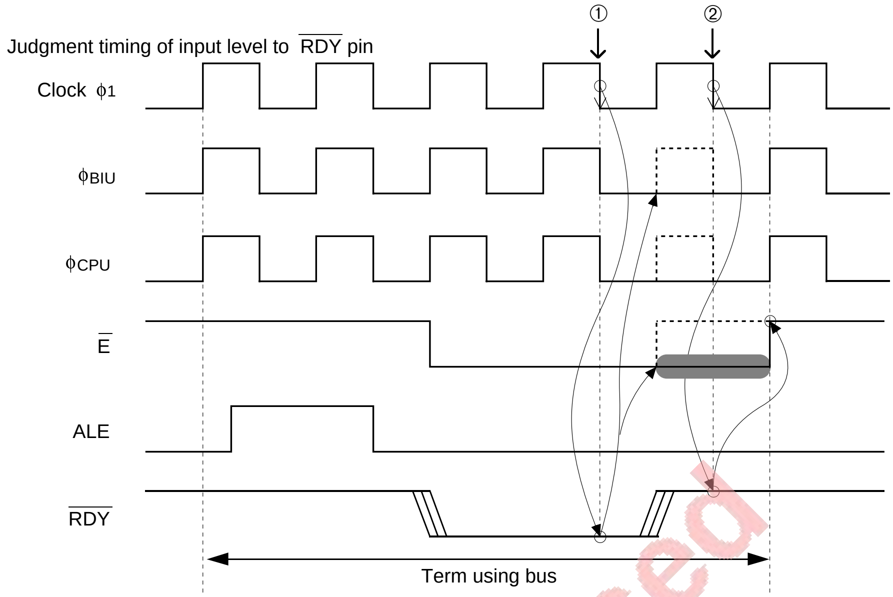
Notes 1: The timing of ALE signal differs depending on low-speed running or high-speed running, and accessing an internal area or an external area. For more information, refer to section "**Chapter 15. ELECTRICAL CHARACTERISTICS**."

2: The dotted lines of signals φBIU, φCPU and E indicate the waveform when input level to the RDY pin is "H", no Ready request.

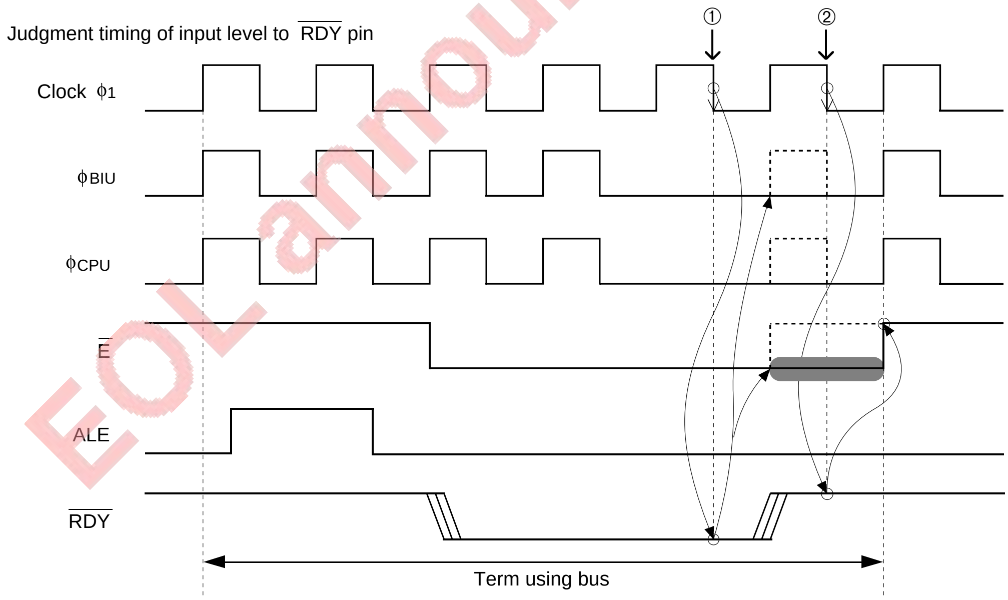
3: In high-speed running, the internal RAM is accessed by 2φ access in high-speed running.

Fig. 12.3.1 Timings of acceptance of Ready request and termination of Ready state (1)

● 4ϕ access in low-speed running, 4ϕ access in high-speed running



● 5ϕ access in high-speed running



① By accepting an Ready request, "L" level of E signal stops for 1 cycle with the clock ϕ1, indicated by [shaded bar], and clocks ϕBIU and ϕCPU stop at "L" level.
② Ready state is terminated.

✻ Input level to the RDY pin is not judged during the term unusing the bus or before the condition above ①.

Notes 1: The timing of ALE signal differs depending on low-speed running or high-speed running, and accessing an internal area or an external area. For more information, refer to section "**Chapter 15. ELECTRICAL CHARACTERISTICS** ."
2: The dotted lines of signals ϕBIU, ϕCPU and E indicate the waveform when input level to the RDY pin is "H", no Ready request.

Fig. 12.3.2 Timings of acceptance of Ready request and termination of Ready state (2)

12.4 Hold function

When composing the external circuit (DMA) which accesses the bus without using the central processing unit (CPU), the Hold function is used to generate a timing for transferring the right to use the bus from the CPU to the external circuit.

In the memory expansion or microprocessor mode, the microcomputer enters Hold state by input of "L" level to the $\overline{HOLD}$ pin and retains this state while the level of the $\overline{HOLD}$ pin is at "L." Table 12.4.1 lists the microcomputer's state in Hold state.

In Hold state, the oscillation of the oscillator does not stop. Accordingly, the internal peripheral devices can operate. However, Watchdog timer stops operating.

Table 12.4.1 Microcomputer's state in Hold state

Item	State
Oscillation	Operating
ϕ_{CPU}	Stopped at "L"
ϕ_{BIU}, ϕ	Operating
$\overline{E}$	Stopped at "H"
Pins A_0 to A_7, A_8/D_8 to A_{15}/D_{15}, A_{16}/D_0 to A_{23}/D_7, $R/\overline{W}$, $\overline{BHE}$	Floating
Pins $\overline{HLDA}$, ALE	Outputs "L" level.
Pin $P4_2/\phi_1$	In the memory expansion mode: •When clock ϕ_1 output select bit* = "1," this pin outputs clock ϕ_1. •When clock ϕ_1 output select bit = "0," this pin retains the state when Hold request was accepted. In the microprocessor mode: •This pin outputs clock ϕ_1.
Pins $P4_3$ to $P4_7$, P5 to P8 **(Note)**	Retains the state when Hold request was accepted.
Watchdog timer	Stopped

Clock ϕ_1 output select bit* : Bit 7 at address $5E_{16}$

Note: When this functions as a programmable I/O port.

12.4.1 Operation description

Judgment timing of the input level of the $\overline{\text{HOLD}}$ pin depends on the state using the bus. While the bus is not in use, the judgment is performed at every falling of ϕ_{BIU}. While the bus is in use, the judgment timing depends on the bus cycle. Table 12.4.2 lists the judgment timing of the input level of the $\overline{\text{HOLD}}$ pin during the used bus.
Additionally, when accessing word data beginning from an odd address with 2-bus cycle, the judgment is performed only at the second bus cycle. (See Figure 12.4.1.)
When "L" level is detected at judgment of the input level, the microcomputer enters Hold state. (This is called acceptance of Hold request.)

When the Hold request is accepted, ϕ_{CPU} stops next rising of ϕ_{BIU}. At the same time, the $\overline{\text{HLDA}}$ pin's level changes "H" to "L". When 1 cycle of ϕ_{BIU} has passed after the level of $\overline{\text{HLDA}}$ pin becomes "L", pins R/$\overline{\text{W}}$, $\overline{\text{BHE}}$, and the external bus become floating state.

In Hold state, the input level of the $\overline{\text{HOLD}}$ pin is judged at every falling of ϕ_{BIU}. Then, when "H" level is detected, the $\overline{\text{HLDA}}$ pin's level changes "L" to "H" next rising of ϕ_{BIU}. When 1 cycle of ϕ_{BIU} has passed after the level of $\overline{\text{HLDA}}$ pin becomes "H", the microcomputer terminates Hold state.
Figures 12.4.2 to 12.4.4 show timing of acceptance of Hold request and termination of Hold state.

Note: ϕ_{BIU} has a same polarity and a same frequency as the clock ϕ_1. However, ϕ_{BIU} stops by acceptance of the Ready request, or executing the **STP** or **WIT** instruction. Accordingly, judgment of the input level of the $\overline{\text{HOLD}}$ pin is not performed during Ready state.

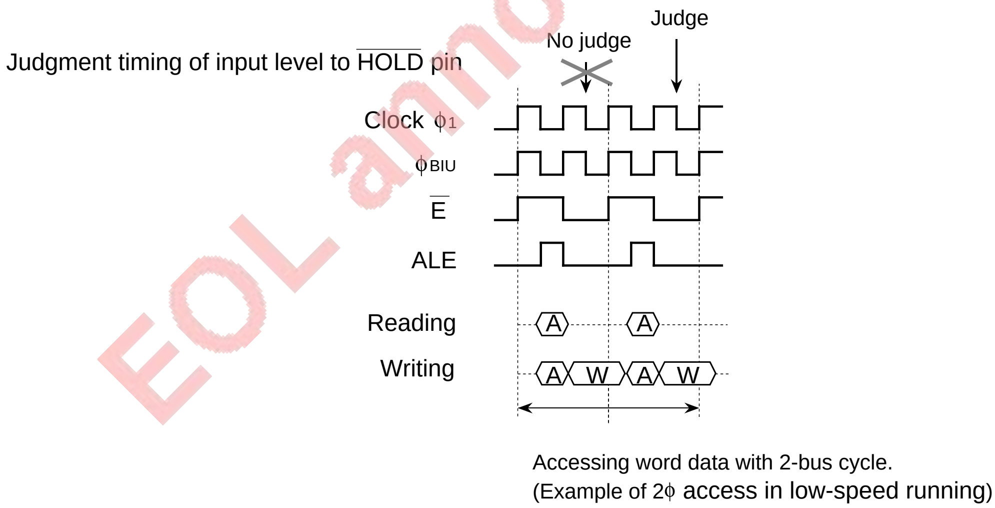
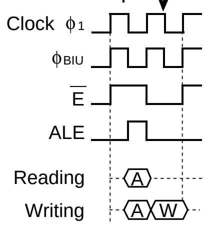
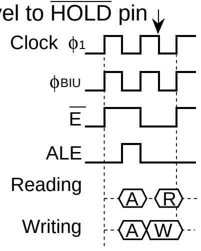
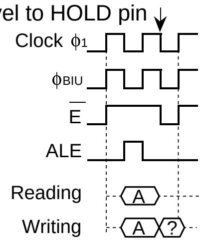
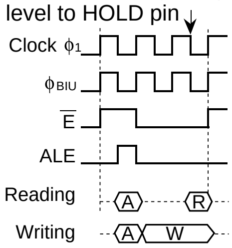
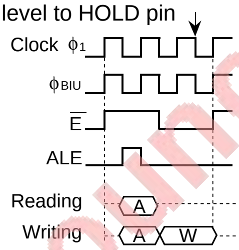
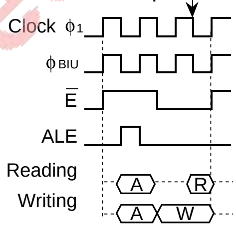
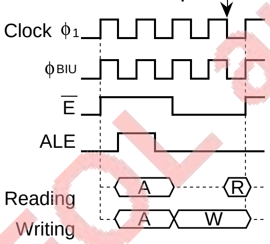
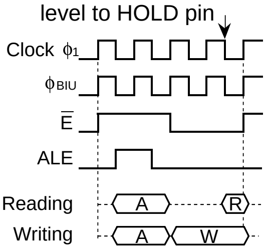
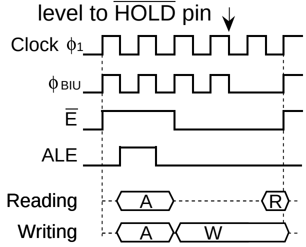


Fig. 12.4.1 Judgment when accessing word data beginning from odd address with 2-bus cycle

Table 12.4.2 Judgment timing of input level of $\overline{HOLD}$ pin during used bus

Low-speed running [f (XIN) ≤ 25 MHz]		High-speed running [f (XIN) ≤ 40 MHz]	
Internal area access (Note)	External area access	Internal area access (Note)	External area access
2φ access in low-speed running Judgment timing of input level to $\overline{HOLD}$ pin 	2φ access in low-speed running Judgment timing of input level to $\overline{HOLD}$ pin 	2φ access in high-speed running (RAM) Judgment timing of input level to $\overline{HOLD}$ pin 	
	3φ access in low-speed running Judgment timing of input level to $\overline{HOLD}$ pin 	3φ access in high-speed running (ROM, SFR) Judgment timing of input level to $\overline{HOLD}$ pin 	3φ access in high-speed running Judgment timing of input level to $\overline{HOLD}$ pin 
	4φ access in low-speed running Judgment timing of input level to $\overline{HOLD}$ pin 		4φ access in high-speed running Judgment timing of input level to $\overline{HOLD}$ pin 
			5φ access in high-speed running Judgment timing of input level to $\overline{HOLD}$ pin 

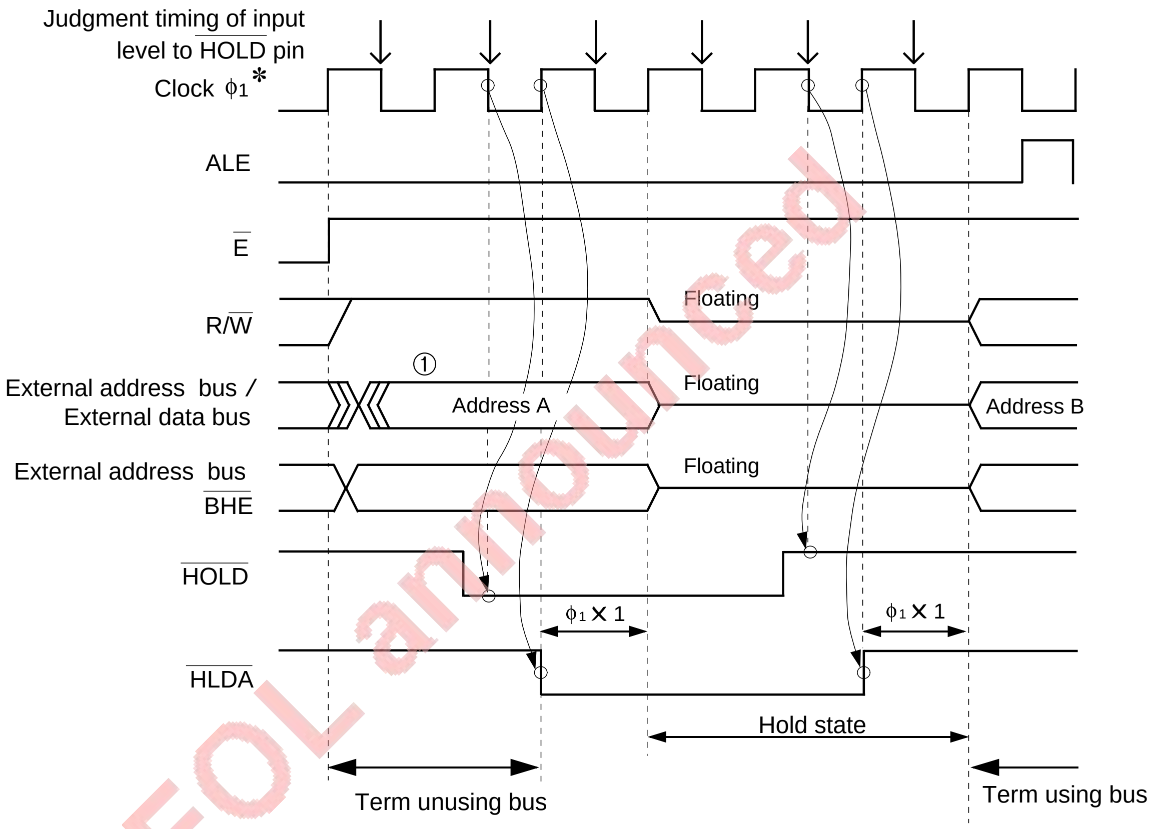
Note : Signals when accessing an internal area means signals which are output from pins externally when accessing an internal area in the memory expansion mode.

A: Address **W**: Data to be written
R: Data to be read **?**: Undefined value

<When inputting “L” level to $\overline{HOLD}$ pin during term unusing bus>

● State when inputting “L” level to $\overline{HOLD}$ pin

External data bus	Data length	External data bus width
Unused	8	8, 16
	16	8, 16



① This is the term in which the bus is not used, so that not a new address but an address output just before is output again.

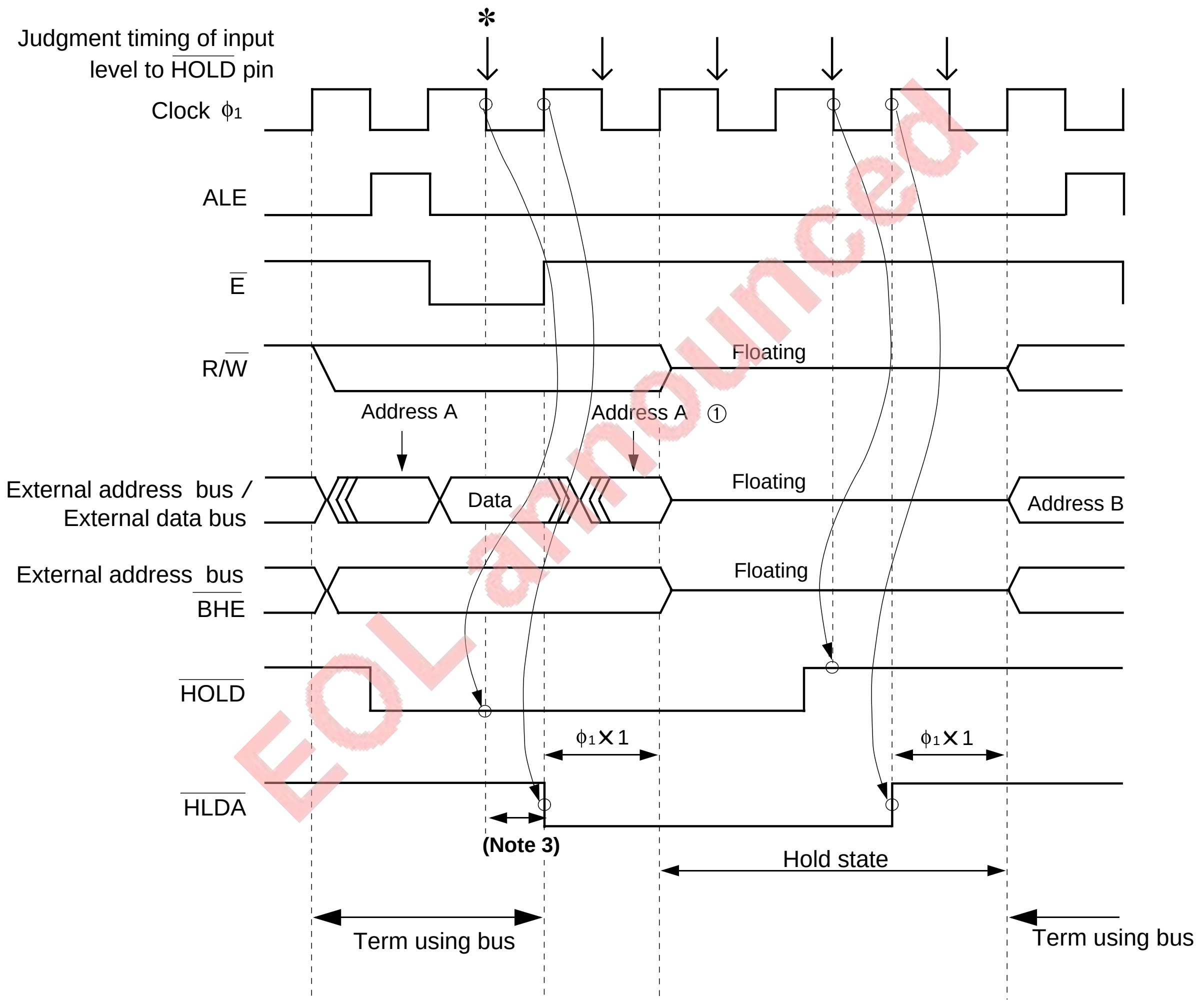
✻ Clock ϕ_1 has the same polarity and the same frequency as ϕ_{BIU}.
Signals timing to be input or output externally is ordained by clock ϕ_1 as a basis.

Fig. 12.4.2 Timing of acceptance of Hold request and termination of Hold state (1)

<When inputting "L" level to $\overline{HOLD}$ pin during term using bus; when data access is completed with 1-bus cycle>

● State when inputting "L" level to $\overline{HOLD}$ pin

External data bus	Data length	External data bus width
Using	8	8, 16
	16	16 (Access from even address)



① When accepting a Hold request, not a new address but an address output just before is output again.

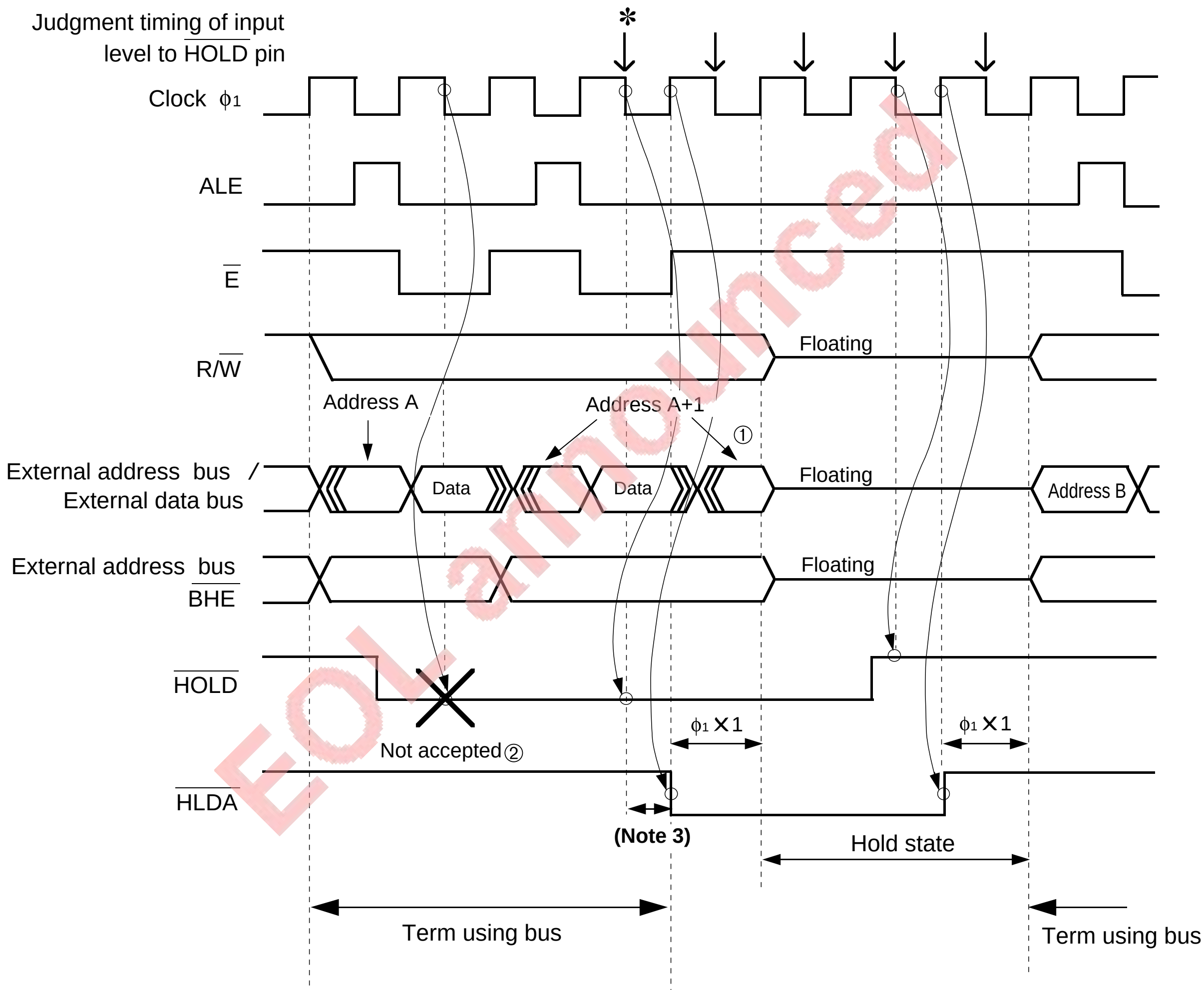
Notes 1: This figure shows the case of 2φ access in low-speed running.
2: Clock $φ_1$ has the same polarity and the same frequency as $φ_{BIU}$.
Signals timing to be input or output externally is ordained by clock $φ_1$ as a basis.
3: This term indicated by **Note 3** becomes 1.5 cycles in 5φ access in high-speed running. It is because the level judgment timing becomes the 1.5 cycles before the end of the term using bus (See Table 12.4.2.)

Fig. 12.4.3 Timing of acceptance of Hold request and termination of Hold state (2)

<When inputting "L" level to $\overline{HOLD}$ pin during term using bus; when data access is completed with continuous 2-bus cycle>

● State when inputting "L" level to $\overline{HOLD}$ pin

External data bus	Data length	External data bus width
Using	16	8
		16 (Access from odd address)



① When accepting a Hold request, not a new address but an address output just before is output again.
② Hold request cannot be accepted before input/output of 16-bit data is completed.

Notes 1: This figure shows the case of 2φ access in low-speed running.
2: Clock φ1 has the same polarity and the same frequency as $φ_{BIU}$.
Signals timing to be input or output externally is ordained by clock φ1 as a basis.
3: This term indicated by **Note 3** becomes 1.5 cycles in 5φ access in high-speed running. It is because the level judgment timing becomes the 1.5 cycles before the end of the term using bus (See Table 12.4.2.)

Fig. 12.4.4 Timing of acceptance of Hold request and termination of Hold state (3)

MEMORANDUM

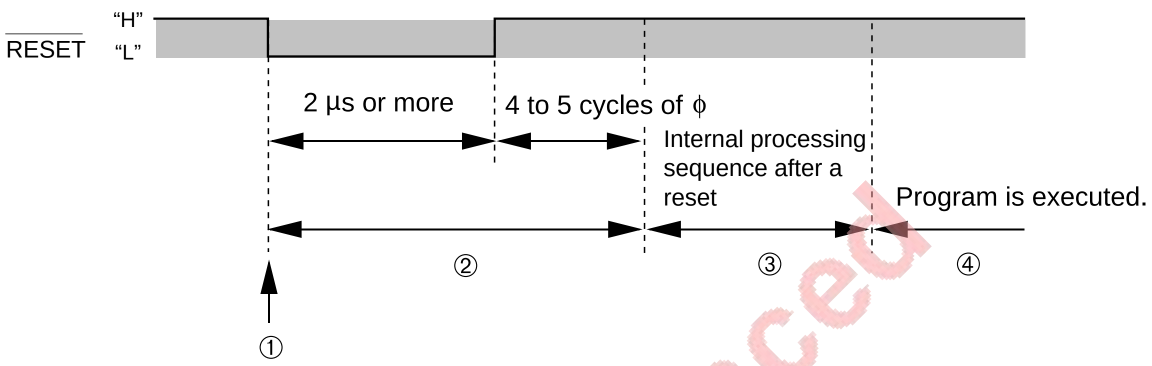
CHAPTER 13

RESET

This chapter describes the method to reset the microcomputer. There are two methods to do that: Hardware reset and Software reset.

13.1 Hardware reset

When the power source voltage satisfies the microcomputer's recommended operating conditions, the microcomputer is reset by supplying "L" level to the $\overline{\text{RESET}}$ pin. This is called a hardware reset. Figure 13.1.1 shows an example of hardware reset timing.



Note: When the clock is stably supplied. (Refer to **"13.1.4 Time supplying "L" level to $\overline{\text{RESET}}$ pin."**)

Fig. 13.1.1 Example of hardware reset timing

The following explains how the microcomputer operates for terms ① to ④ above.

① After supplying "L" level to the $\overline{\text{RESET}}$ pin, the microcomputer initializes pins within a term of several ten ns. (Refer to Table 13.1.1.)

② While the $\overline{\text{RESET}}$ pin is "L" level and within the term of 4 to 5 cycles of the internal clock ϕ after the $\overline{\text{RESET}}$ pin goes from "L" to "H," the microcomputer initializes the central processing unit (CPU) and SFR area. At this time, the contents of the internal RAM area become undefined (except when Stop or Wait mode is terminated). (Refer to Figures 13.1.2 to 13.1.6.)

③ After ②, the microcomputer performs "Internal processing sequence after reset." (Refer to Figure 13.1.7.)

④ The microcomputer executes a program beginning with the address set into the reset vector addresses which are $FFFE_{16}$ and $FFFF_{16}$.

13.1.1 Pin state

Table 13.1.1 lists the microcomputer's pin state while the $\overline{RESET}$ pin is "L" level.

Table 13.1.1 Pin state while $\overline{RESET}$ pin is "L" level

	CNVss pin level	Pin (Port) name	Pin state
Mask ROM version	V_{SS} or V_{CC}	P0 to P8	Floating.
		$\overline{E}$	Outputs "H" level.
PROM version (Including One time PROM and EPROM versions)	V_{SS}	P0 to P8	Floating.
		$\overline{E}$	Outputs "H" level.
	V_{CC} (**Note 1**)	P0, P1, P3 to P8	Floating.
		P2	Floating while supplying "H" level to two pins of $P5_1$ and $P5_2$, or one of them. Outputs "H" or "L" level while supplying "L" level to two pins of $P5_1$ and $P5_2$.
		$\overline{E}$	Outputs "H" level.
Flash memory version	V_{SS}	P0 to P8	Floating.
		$\overline{E}$	Outputs "H" level.
	V_{CC} (**Note 2**)	P0, P1, P3 to P8	Floating.
		P2	Floating while supplying "H" level to two pins of $P5_1$ and $P5_2$, or one of them. Outputs "H" or "L" level while supplying "L" level to two pins of $P5_1$ and $P5_2$.
		$\overline{E}$	Outputs "H" level.
	V_{PPH} (**Note 2**)	P0, P1, P3, $P4_0$, $P4_1$, $P4_3$,$P4_5$ to $P4_7$, P5 to P8	Floating.
		P2	Floating while supplying "H" level to two pins of $P5_1$ and $P5_2$, or one of them. Outputs "H" or "L" level while supplying "L" level to two pins of $P5_1$ and $P5_2$.
		$P4_2$	Outputs clock ϕ_1.
		$P4_4$	Floating while supplying "L" level to one or more pins of $P4_5$, $P4_6$ and $P5_1$. Outputs "H" or "L" level while supplying "H" level to three pins of $P4_5$, $P4_6$ and $P5_1$.
		$\overline{E}$	Outputs "H" level.

Notes 1: Each pin becomes the above state. It is because the microcomputer enters the EPROM mode. Refer to "**Chapter 18. PROM VERSION**."

2: Each pin becomes the above state. It is because the microcomputer enters the Flash memory mode. Refer to "**Chapter 19. FLASH MEMORY VERSION**."

13.1.2 State of CPU, SFR area, and internal RAM area

Figure 13.1.2 shows the state of the CPU registers immediately after reset. Figures 13.1.3 to 13.1.6 show the state of the SFR area and internal RAM areas immediately after reset.

0 : "0" immediately after a reset.
1 : "1" immediately after a reset.
? : Undefined immediately after a reset.
0 (hatched) : "0" immediately after a reset. Fix to "0."
(Do not write "1" into this.

Register name	State immediately after a reset: b15–b8	b7–b0
Accumulator A (A)	?	?
Accumulator B (B)	?	?
Index register X (X)	?	?
Index register Y (Y)	?	?
Stack pointer (S)	?	?
Data bank register (DT)		00_{16}
Program bank register (PG)		00_{16}
Program counter (PC)	Contents at address $FFFF_{16}$	Contents at address $FFFE_{16}$
Direct page register (DPR)	00_{16}	00_{16}

Processor status register (PS):

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0 (hatched)	0 (hatched)	0 (hatched)	0 (hatched)	0 (hatched)	0	0	0	?	?	0	0	0	1	?	?
					IPL	IPL	IPL	N	V	m	x	D	I	Z	C

Fig. 13.1.2 State of CPU registers immediately after reset

●SFR area (0_{16} to $7F_{16}$)

RW : It is possible to read the bit state at reading. The written value becomes valid data.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid data. It is not possible to read the bit state.
(shaded) : Nothing is assigned. It is not possible to read the bit state. The written value becomes invalid.

0 : "0" immediately after a reset.
1 : "1" immediately after a reset.
? : Undefined immediately after a reset.

[0] : Always "0" at reading.
[?] : Always undefined at reading.
[0] (dark) : "0" immediately after a reset. Fix to "0." (Do not write "1" into this.)

Address	Register name	Access characteristics b7							b0	State immediately after a reset b7							b0
0_{16}										?							
1_{16}										?							
2_{16}	Port P0 register	RW								?							
3_{16}	Port P1 register	RW								?							
4_{16}	Port P0 direction register	RW								00_{16}							
5_{16}	Port P1 direction register	RW								00_{16}							
6_{16}	Port P2 register	RW								?							
7_{16}	Port P3 register					RW				0	0	0	0	?			
8_{16}	Port P2 direction register	RW								00_{16}							
9_{16}	Port P3 direction register					RW				0	0	0	0	0	0	0	0
A_{16}	Port P4 register	RW								?							
B_{16}	Port P5 register	RW								?							
C_{16}	Port P4 direction register	RW								00_{16}							
D_{16}	Port P5 direction register	RW								00_{16}							
E_{16}	Port P6 register	RW								?							
F_{16}	Port P7 register	RW								?							
10_{16}	Port P6 direction register	RW								00_{16}							
11_{16}	Port P7 direction register	RW								00_{16}							
12_{16}	Port P8 register	RW								?							
13_{16}										?							
14_{16}	Port P8 direction register	RW								00_{16}							
15_{16}										?							
16_{16}										?							
17_{16}										?							
18_{16}										?							
19_{16}										?							
$1A_{16}$										?							
$1B_{16}$										?							
$1C_{16}$										?							
$1D_{16}$										?							
$1E_{16}$	A-D control register 0	RW								0	0	0	0	0	?		
$1F_{16}$	A-D control register 1				RW					?	?	?	0	0	0	1	1

Fig. 13.1.3 State of SFR and internal RAM areas immediately after reset (1)

Address	Register name	Access characteristics (b7 … b0)	State immediately after a reset (b7 … b0)
20_{16}	A-D register 0	RO	?
21_{16}		RO	0 0 0 0 0 0 ?
22_{16}	A-D register 1	RO	?
23_{16}		RO	0 0 0 0 0 0 ?
24_{16}	A-D register 2	RO	?
25_{16}		RO	0 0 0 0 0 0 ?
26_{16}	A-D register 3	RO	?
27_{16}		RO	0 0 0 0 0 0 ?
28_{16}	A-D register 4	RO	?
29_{16}		RO	0 0 0 0 0 0 ?
$2A_{16}$	A-D register 5	RO	?
$2B_{16}$		RO	0 0 0 0 0 0 ?
$2C_{16}$	A-D register 6	RO	?
$2D_{16}$		RO	0 0 0 0 0 0 ?
$2E_{16}$	A-D register 7	RO	?
$2F_{16}$		RO	0 0 0 0 0 0 ?
30_{16}	UART0 transmit/receive mode register	RW	00_{16}
31_{16}	UART0 baud rate register	WO	?
32_{16}	UART0 transmit buffer register	WO	?
33_{16}		(b7–b1 shaded) WO	?
34_{16}	UART0 transmit/receive control register 0	RW (b6–b4 shaded) RO RW	0 ? ? ? 1 0 0 0
35_{16}	UART0 transmit/receive control register 1	RO RW RO RW	0 0 0 0 0 0 1 0
36_{16}	UART0 receive buffer register	RO	?
37_{16}		(b7–b1 shaded) RO	0 0 0 0 0 0 0 ?
38_{16}	UART1 transmit/receive mode register	RW	00_{16}
39_{16}	UART1 baud rate register	WO	?
$3A_{16}$	UART1 transmit buffer register	WO	?
$3B_{16}$		(b7–b1 shaded) WO	?
$3C_{16}$	UART1 transmit/receive control register 0	RW (b6–b4 shaded) RO RW	0 ? ? ? 1 0 0 0
$3D_{16}$	UART1 transmit/receive control register 1	RO RW RO RW	0 0 0 0 0 0 1 0
$3E_{16}$	UART1 receive buffer register	RO	?
$3F_{16}$		(b7–b1 shaded) RO	0 0 0 0 0 0 0 ?

Fig. 13.1.4 State of SFR and internal RAM areas immediately after reset (2)

Address	Register name	Access characteristics (b7 … b0)	State immediately after a reset (b7 … b0)
40_{16}	Count start register	RW	00_{16}
41_{16}			?
42_{16}	One-shot start register	b7–b5: —, b4–b0: WO	? 0 0 0 0 0
43_{16}			?
44_{16}	Up-down register	b7–b5: WO, b4–b0: RW	0 0 0 0 0 0 0 0
45_{16}			?
46_{16}	Timer A0 register	(Note 1)	?
47_{16}		(Note 1)	?
48_{16}	Timer A1 register	(Note 1)	?
49_{16}		(Note 1)	?
$4A_{16}$	Timer A2 register	(Note 1)	?
$4B_{16}$		(Note 1)	?
$4C_{16}$	Timer A3 register	(Note 1)	?
$4D_{16}$		(Note 1)	?
$4E_{16}$	Timer A4 register	(Note 1)	?
$4F_{16}$		(Note 1)	?
50_{16}	Timer B0 register	(Note 2)	?
51_{16}		(Note 2)	?
52_{16}	Timer B1 register	(Note 2)	?
53_{16}		(Note 2)	?
54_{16}	Timer B2 register	(Note 2)	?
55_{16}		(Note 2)	?
56_{16}	Timer A0 mode register	RW	00_{16}
57_{16}	Timer A1 mode register	RW	00_{16}
58_{16}	Timer A2 mode register	RW	00_{16}
59_{16}	Timer A3 mode register	RW	00_{16}
$5A_{16}$	Timer A4 mode register	RW	00_{16}
$5B_{16}$	Timer B0 mode register	b7–b6: RW, b5: (Note 3), b4: —, b3–b0: RW	0 0 ? ? 0 0 0 0
$5C_{16}$	Timer B1 mode register	b7–b6: RW, b5: (Note 3), b4: —, b3–b0: RW	0 0 ? ? 0 0 0 0
$5D_{16}$	Timer B2 mode register	b7–b6: RW, b5: (Note 3), b4: —, b3–b0: RW	0 0 ? ? 0 0 0 0
$5E_{16}$	Processor mode register 0	b7–b4: RW, b3: WO, b2: RW, b1: (Note 4), b0: RW	0 0 0 0 0 0 (Note 4) 0
$5F_{16}$	Processor mode register 1	RW	0 0 0 0 0 0 0 0

Notes 1: The access characteristics at addresses 46_{16} to $4F_{16}$ vary according to Timer A's operating mode. (Refer to **"Chapter 5. TIMER A."**)
2: The access characteristics at addresses 50_{16} to 55_{16} vary according to Timer B's operating mode. (Refer to **"Chapter 6. TIMER B."**)
3: The access characteristics for bit 5 at addresses $5B_{16}$ to $5D_{16}$ vary according to Timer B's operating mode. (Refer to **"Chapter 6. TIMER B."**)
4: The access characteristics for bit 1 at address $5E_{16}$ and its state immediately after a reset vary according to the voltage level supplied to the CNVss pin. (Refer to section **"2.5 Processor modes."**)

Fig. 13.1.5 State of SFR and internal RAM areas immediately after reset (3)

Address	Register name	Access b7							Access b0	State b7							State b0
60_{16}	Watchdog timer register	(Note 1)								?(Note 2)							
61_{16}	Watchdog timer frequency select register								RW	?							0
62_{16}										?							
63_{16}										?							
64_{16}										?							
65_{16}										?							
66_{16}										?							
67_{16}										?							
68_{16}										?							
69_{16}										?							
$6A_{16}$										?							
$6B_{16}$										?							
$6C_{16}$										?							
$6D_{16}$										?							
$6E_{16}$										?							
$6F_{16}$										?							
70_{16}	A-D conversion interrupt control register					RW				?				?	0	0	0
71_{16}	UART0 transmit interrupt control register					RW				?				0	0	0	0
72_{16}	UART0 receive interrupt control register					RW				?				0	0	0	0
73_{16}	UART1 transmit interrupt control register					RW				?				0	0	0	0
74_{16}	UART1 receive interrupt control register					RW				?				0	0	0	0
75_{16}	Timer A0 interrupt control register					RW				?				0	0	0	0
76_{16}	Timer A1 interrupt control register					RW				?				0	0	0	0
77_{16}	Timer A2 interrupt control register					RW				?				0	0	0	0
78_{16}	Timer A3 interrupt control register					RW				?				0	0	0	0
79_{16}	Timer A4 interrupt control register					RW				?				0	0	0	0
$7A_{16}$	Timer B0 interrupt control register					RW				?				0	0	0	0
$7B_{16}$	Timer B1 interrupt control register					RW				?				0	0	0	0
$7C_{16}$	Timer B2 interrupt control register					RW				?				0	0	0	0
$7D_{16}$	$\overline{INT_0}$ interrupt control register			RW						?	0	0	0	0	0	0	0
$7E_{16}$	$\overline{INT_1}$ interrupt control register			RW						?	0	0	0	0	0	0	0
$7F_{16}$	$\overline{INT_2}$ interrupt control register			RW						?	0	0	0	0	0	0	0

Notes 1: By writing dummy data to address 60_{16}, a value "FFF_{16}" is set to the watchdog timer. The dummy data is not retained anywhere.

2: The value "FFF_{16}" is set to the watchdog timer. (Refer to "**Chapter 9. WATCHDOG TIMER**.")

●Internal RAM area; addresses 80_{16} to $87F_{16}$ in M37751M6C-XXXFP)

- At hardware reset (Except the case that Stop or Wait mode is terminated)... Undefined.
- At software reset... Retaining the state immediately before a reset
- At terminating Stop or Wait mode (Hardware reset is used to terminate it)............... Retaining the state immediately before the **STP** or **WIT** instruction is executed

Fig. 13.1.6 State of SFR and internal RAM areas immediately after reset (4)

13.1.3 Internal processing sequence after reset

Figure 13.1.7 shows the internal processing sequence after reset.

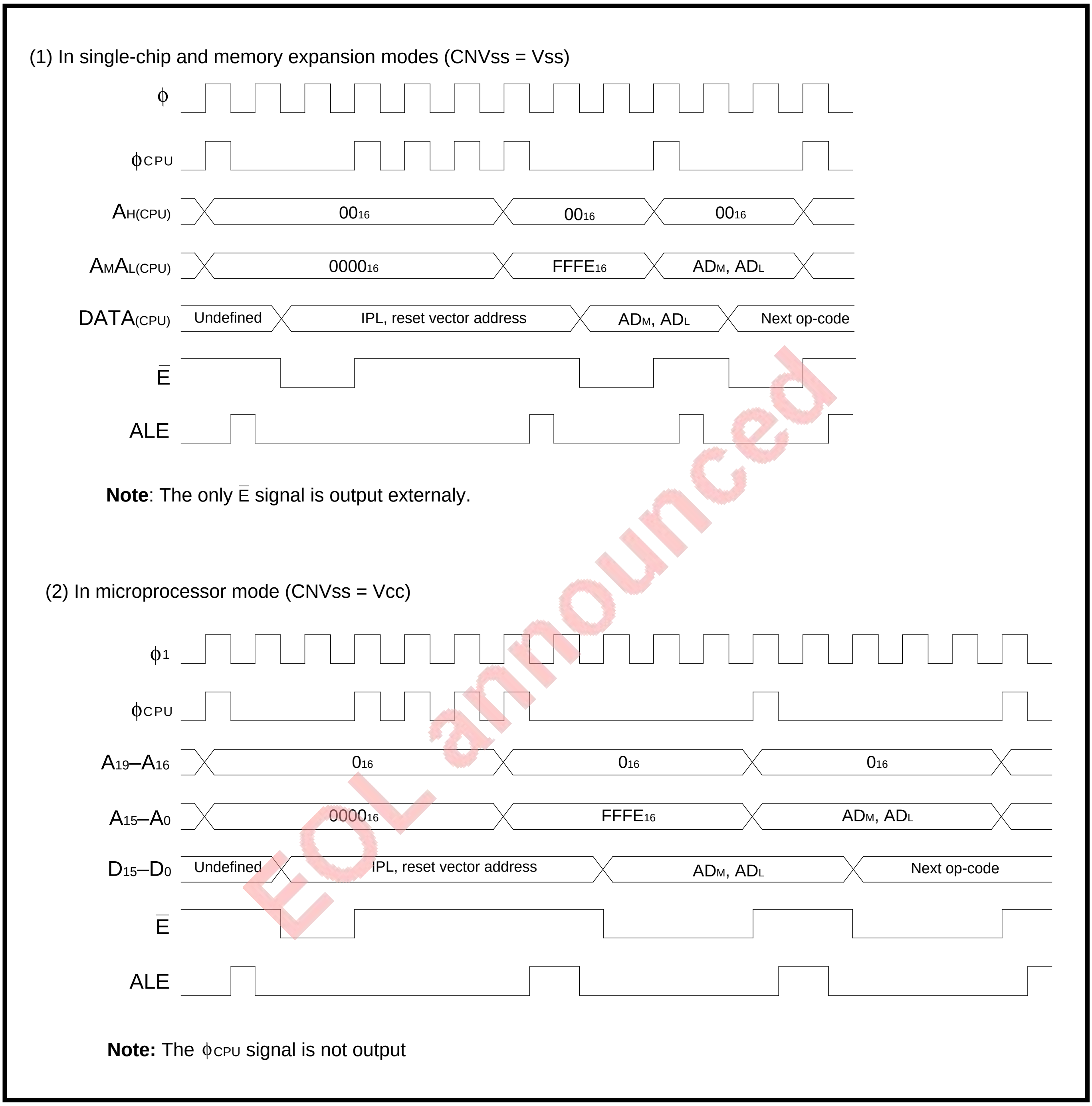


Fig. 13.1.7 Internal processing sequence after reset

13.1 Hardware reset

13.1.4 Time supplying “L” level to $\overline{\text{RESET}}$ pin

Time supplying “L” level to the $\overline{\text{RESET}}$ pin varies according to the state of the clock oscillation circuit.

●When the oscillator is stably oscillating or a stable clock is input from the X_{IN} pin, supply “L” level for 2 μs or more.

●If the oscillator is not stably oscillating (including a power-on reset and In Stop mode), supply “L” level until the oscillation is stabilized.
The time to stabilize oscillation varies according to the oscillator. For details, contact the oscillator manufacturer.
Figure 13.1.8 shows the power-on reset condition. Figure 13.1.9 shows an example of a power-on reset circuit.

✻ For details about Stop mode, refer to **“Chapter 10. STOP MODE.”** For details about clocks, refer to **“Chapter 14. CLOCK GENERATING CIRCUIT.”**

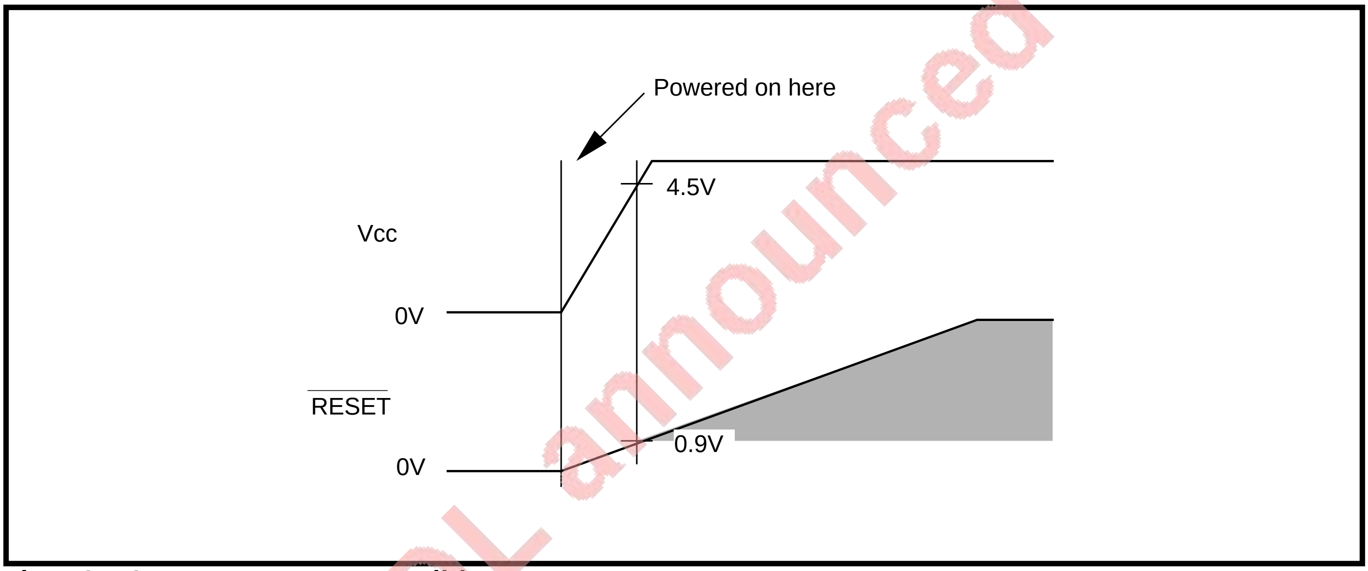
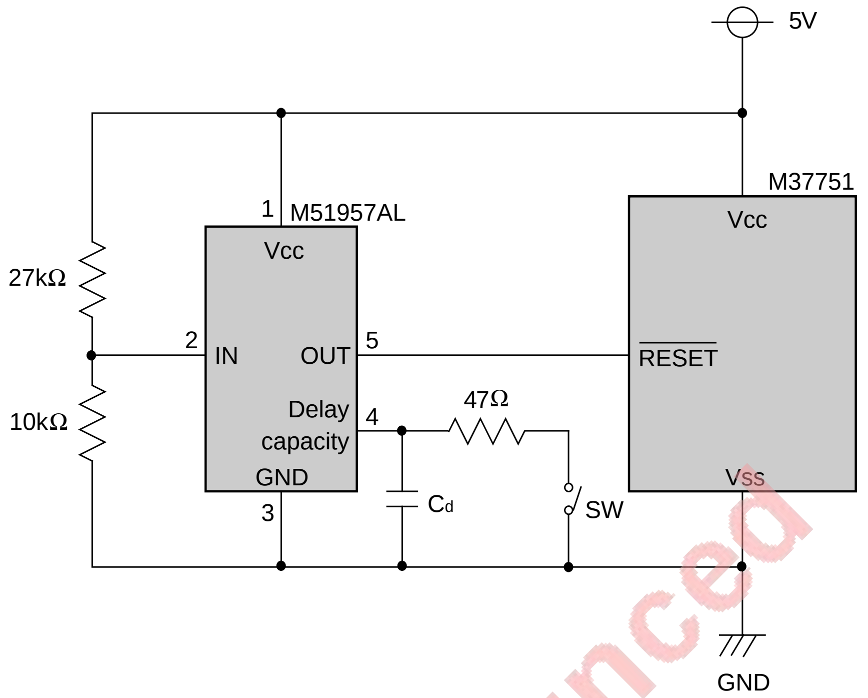


Fig. 13.1.8 Power-on reset condition



✻ The delay time is about 11 ms when C_d = 0.033 µF.

$t_d \approx 0.34 \times C_d$ [µs], C_d: [pF]

Fig. 13.1.9 Example of power-on reset circuit

13.2 Software reset

When the power source voltage satisfies the microcomputer's recommended operating conditions, the microcomputer is reset by writing "1" to the software reset bit (bit 3 at address $5E_{16}$). This is called a software reset. In this case, the microcomputer initializes pins, CPU, and SFR area just as in the case of a hardware reset. However, the microcomputer retains the contents of the internal RAM area. (Refer to Table 13.1.1 and Figures 13.1.2 to 13.1.6.) Figure 13.2.1 shows the structure of processor mode register 0. After completing initialization, the microcomputer performs the internal processing sequence after a reset. (Refer to Figure 13.1.7.) After that, it executes a program beginning from the address set into the reset vector addresses which are $FFFE_{16}$ and $FFFF_{16}$.

b7	b6	b5	b4	b3	b2	b1	b0
	0				0		

Processor mode register 0 (Address $5E_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Memory expansion mode 1 0 : Microprocessor mode 1 1 : Not selected	0	RW
1			0 **(Note 1)**	RW
2	Fix this bit to "0."		0	RW
3	Software reset bit	The microcomputer is reset by writing "1" to this bit. The value is "0" at reading.	0	WO
4	Interrupt priority detection time select bits	b5 b4 0 0 : 7 cycles of φ 0 1 : 4 cycles of φ 1 0 : 2 cycles of φ 1 1 : Not selected	0	RW
5			0	RW
6	Fix this bit to "0."		0	RW
7	Clock ϕ_1 output select bit **(Note 2)**	0 : Clock ϕ_1 output disabled (P4₂ functions as a programmable I/O port.) 1 : Clock ϕ_1 output enabled (P4₂ functions as a clock ϕ_1 output pin.)	0	RW

Notes 1: While supplying the Vcc level to the CNVss pin, this bit becomes "1." (Fixed to "1.")
2: This bit is ignored in the microprocessor mode. (It may be either "0" or "1.")

(shaded box) : Bits 0 to 2, and bits 4 to 7 are not used for software reset.

Fig. 13.2.1 Structure of processor mode register 0

CHAPTER 14

CLOCK GENERATING CIRCUIT

This chapter describes a clock generating circuit which supplies the operating clock of the central processing unit (CPU), bus interface unit (BIU), or internal peripheral devices.
The clock generating circuit contains the oscillation circuit.

14.1 Oscillation circuit example

To the oscillation circuit, a ceramic resonator or a quartz-crystal oscillator can be connected, or the clock which is externally generated can be input. The example of the oscillation circuit is described below.

14.1.1 Connection example using resonator/oscillator

Figure 14.1.1 shows an example when connecting a ceramic resonator/quartz-crystal oscillator between pins X_{IN} and X_{OUT}.
The circuit constants such as R_f, R_d, C_{IN}, and C_{OUT} (shown in Figure 14.1.1) depend on the resonator/oscillator. These values shall be set to the resonator/oscillator manufacturer's recommended values.

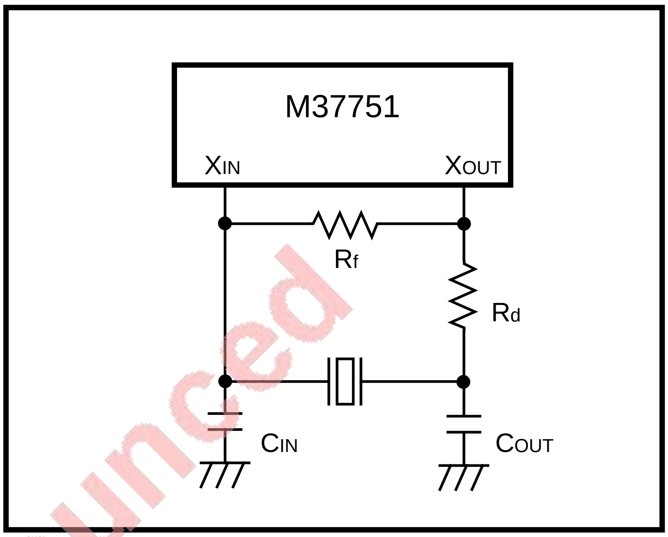


Fig. 14.1.1 Connection example using resonator/oscillator

14.1.2 Input example of externally generated clock

Figure 14.1.2 shows an input example of the clock which is externally generated. The external clock must be input from the X_{IN} pin, and the X_{OUT} pin must be left open.

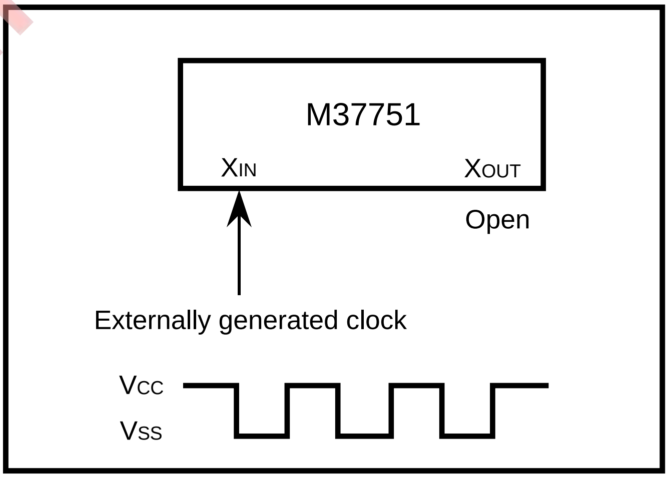
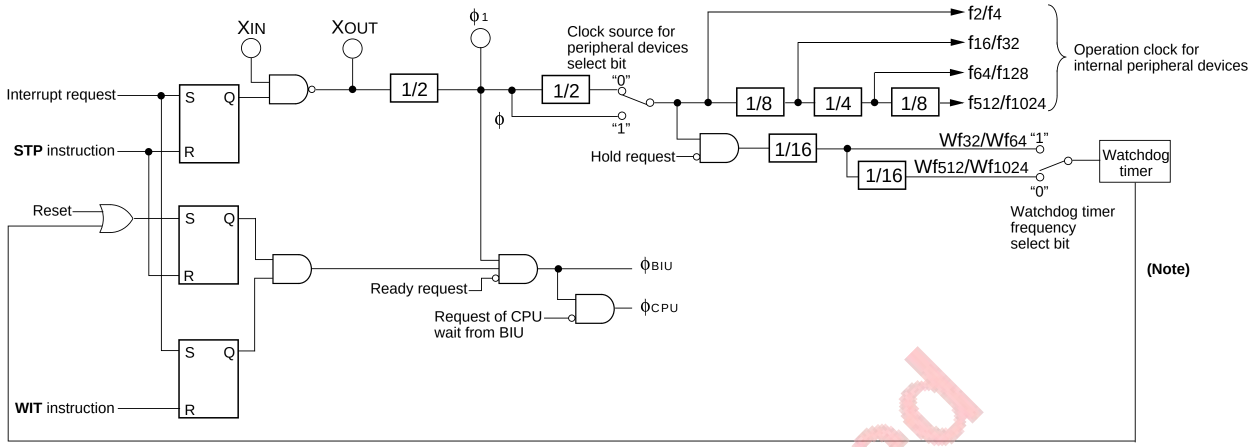


Fig. 14.1.2 Externally generated clock input example

14.2 Clock

Figure 14.2.1 shows the clock generating circuit block diagram.



CPU	: Central Processing Unit
BIU	: Bus Interface Unit
Clock source for peripheral devices select bit	: Bit 2 at address $5F_{16}$
Watchdog timer frequency select bit	: Bit 0 at address 61_{16}

Note: This is the signal generated when the watchdog timer's most significant bit becomes "0."

Fig. 14.2.1 Clock generating circuit block diagram

14.2.1 Clock generated in clock generating circuit

(1) ϕ

This is the clock source of ϕ_{CPU}, ϕ_{BIU} clock ϕ_1, f_2/f_4 to f_{512}/f_{1024}, Wf_{32}/Wf_{64} and Wf_{512}/Wf_{1024}.

(2) ϕ_{CPU}

This is the operation clock of CPU.

(3) ϕ_{BIU}

This is the operation clock of BIU.

(4) Clock ϕ_1

This has the same period as ϕ and is output to the external.

(5) f_2/f_4 **to** f_{512}/f_{1024}

Each of them is the operation clock for the internal peripheral devices, and its clock source is ϕ or ϕ divided by 2.
(Refer to "**14.2.2 Operation clock for internal peripheral devices**.")

Table 14.2.1 Operation clock for internal peripheral devices

Operation clock	Clock source for peripheral devices select bit (See Fig. 14.2.2)	
	1	0
f_2/f_4	f_2	f_4
f_{16}/f_{32}	f_{16}	f_{32}
f_{64}/f_{128}	f_{64}	f_{128}
f_{512}/f_{1024}	f_{512}	f_{1024}

(6) Wf_{32}/Wf_{64}**,** Wf_{512}/Wf_{1024}

This is the operation clock of Watchdog timer, and its clock source is ϕ or ϕ divided by 2.
(Refer to "**14.2.2 Operation clock for internal peripheral devices**.")

Table 14.2.2 Operation clock for Watchdog timer

Operation clock	Clock source for peripheral devices select bit (See Fig. 14.2.2)	
	1	0
Wf_{32}/Wf_{64}	Wf_{32}	Wf_{64}
Wf_{512}/Wf_{1024}	Wf_{512}	Wf_{1024}

14.2.2 Operation clock for internal peripheral devices

The operation clock for the internal peripheral devices uses ϕ or ϕ divided by 2 as its clock source.
The clock source of the operation clock for internal peripheral devices is selected by the clock source for peripheral devices select bit (bit 2 at address $5F_{16}$).
Figure 14.2.2 shows the structure of processor mode register 1 (address $5F_{16}$).

When $f(X_{IN})$ > 25 MHz, fix the clock source for peripheral devices select bit to "0."

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

Processor mode register 1 (Address $5F_{16}$)

Bit	Bit name	Functions	At reset	RW
1, 0	Fix these bits to "0."		0	RW
2	Clock source for peripheral devices select bit **(Note)**	0 : ϕ divided by 2 1 : ϕ	0	RW
3	CPU running speed select bit **(Note)**	0 : High-speed running 1 : Low-speed running	0	RW
4	Bus cycle select bits	In high-speed running b5 b4 0 0 : 5ϕ access in high-speed running 0 1 : 4ϕ access in high-speed running 1 0 : 3ϕ access in high-speed running 1 1 : Not selected	0	RW
5		In low-speed running b5 b4 0 0 : Not selected 0 1 : 4ϕ access in low-speed running 1 0 : 3ϕ access in low-speed running 1 1 : 2ϕ access in low-speed running	0	RW
7, 6	Fix these bits to "0."		0	RW

Note: Fix this bit to "0" when $f(X_{IN})$ > 25 MHz.

(shaded) : Bits 0, 1, and bits 3 to 7 are not used for the clock generating circuit.

Fig. 14.2.2 Structure of processor mode register 1

MEMORANDUM

CHAPTER 15

ELECTRICAL CHARACTERISTICS

This chapter describes electrical characteristics of the M37751M6C-XXXFP.
For the latest data, inquire of addresses described last (☞**"CONTACT ADDRESSES FOR FURTHER INFORMATION")**.

15.1 Absolute maximum ratings

Absolute maximum ratings

Symbol	Parameter		Conditions	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	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, V_{REF}, X_{IN}		–0.3 to V_{CC}+0.3	V
V_O	Output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, X_{OUT}, $\overline{E}$		–0.3 to V_{CC}+0.3	V
P_d	Power dissipation		Ta = 25 °C	300	mW
T_{opr}	Operating temperature			–20 to 85	°C
T_{stg}	Storage temperature			–40 to 150	°C

15.2 Recommended operating conditions

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}	Power source voltage		4.5	5.0	5.5	V
AV_{CC}	Analog power source voltage			V_{CC}		V
V_{SS}	Power source voltage			0		V
AV_{SS}	Analog power source voltage			0		V
V_{IH}	High-level input voltage	$P0_0$–$P0_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	0.8 V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (in single-chip mode)	0.8 V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (in memory expansion mode and microprocessor mode)	0.5 V_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage	$P0_0$–$P0_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	0		0.2 V_{CC}	V
V_{IL}	Low-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (in single-chip mode)	0		0.2 V_{CC}	V
V_{IL}	Low-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (in memory expansion mode and microprocessor mode)	0		0.16 V_{CC}	V
$I_{OH (peak)}$	High-level peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			–10	mA
$I_{OH (avg)}$	High-level average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			–5	mA
$I_{OL (peak)}$	Low-level peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			10	mA
$I_{OL (avg)}$	Low-level average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			5	mA
$f(X_{IN})$	Operating clock frequency				40	MHz

Notes 1: Average output current is the average value of a 100 ms interval.

2: The sum of I_{OL}(peak) for ports P0, P1, P2, P3, and P8 must be 80 mA or less, the sum of I_{OH}(peak) for ports P0, P1, P2, P3, and P8 must be 80 mA or less, the sum of I_{OL}(peak) for ports P4, P5, P6, and P7 must be 80 mA or less, and the sum of I_{OH}(peak) for ports P4, P5, P6, and P7 must be 80 mA or less.

15.3 Electrical characteristics

Electrical characteristics (V_{CC} = 5 V, V_{SS} = 0 V, Ta = –20 to 85 °C, f(X_{IN}) = 40 MHz, unless otherwise noted)

Symbol	Parameter		Test conditions		Limits			Unit
					Min.	Typ.	Max.	
V_{OH}	High-level output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	I_{OH} = –10 mA		3			V
V_{OH}	High-level output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OH} = –400 μA		4.7			V
V_{OH}	High-level output voltage	$P3_2$	I_{OH} = –10 mA		3.1			V
			I_{OH} = –400 μA		4.8			
V_{OH}	High-level output voltage	$\overline{E}$	I_{OH} = –10 mA		3.4			V
			I_{OH} = –400 μA		4.8			
V_{OL}	Low-level output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	I_{OL} = 10 mA				2	V
V_{OL}	Low-level output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OL} = 2 mA				0.45	V
V_{OL}	Low-level output voltage	$P3_2$	I_{OL} = 10 mA				1.9	V
			I_{OL} = 2 mA				0.43	
V_{OL}	Low-level output voltage	$\overline{E}$	I_{OL} = 10 mA				1.6	V
			I_{OL} = 2 mA				0.4	
V_{T+}–V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$–$TA4_{IN}$, $TB0_{IN}$–$TB2_{IN}$, $\overline{INT_0}$–$\overline{INT_2}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1			0.4		1	V
V_{T+}–V_{T-}	Hysteresis	$\overline{RESET}$			0.2		0.5	V
V_{T+}–V_{T-}	Hysteresis	X_{IN}			0.1		0.3	V
I_{IH}	High-level input current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I = 5 V				5	μA
I_{IL}	Low-level input current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I = 0 V				–5	μA
V_{RAM}	RAM hold voltage		When clock is stopped.		2			V
I_{CC}	Power source current		In single-chip mode, output pins are open, and the other pins are connected to V_{SS}.	f(X_{IN}) = 40 MHz		25	50	mA
				Ta = 25°C, when clock is stopped			1	μA
				Ta = 85°C, when clock is stopped			20	μA

15.4 A-D converter characteristics

A-D CONVERTER CHARACTERISTICS (V_{CC} = AV_{CC} = 5 V±10%, V_{SS} = AV_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits Min.	Typ.	Max.	Unit
—	Resolution	V_{REF} = V_{CC}				10	Bits
—	Absolute accuracy	V_{REF} = V_{CC}	Resolution 10 bit			±3	LSB
			Resolution 8 bit			±2	
R_{LADDER}	Ladder resistance	V_{REF} = V_{CC}		5			kΩ
t_{CONV}	Conversion time	f(X_{IN}) = 40 MHz	Resolution 10 bit	5.9			µs
			Resolution 8 bit	4.9			
		f(X_{IN}) = 25 MHz	Resolution 10 bit	4.72			
			Resolution 8 bit	3.92			
V_{REF}	Reference voltage			2		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

15.5 Internal peripheral devices

Timing requirements (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Timer A input (Count input in event counter mode)

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

Timer A input (Gating input in timer mode)

Symbol	Parameter	Data formula (Min.)		Limits Min.	Limits Max.	Unit
$t_{c(TA)}$	TAi_{IN} 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 when ϕ divided by 2 selected as clock source for peripheral devices	$\frac{16 \times 10^9}{f(X_{IN})}$	640		
		$f(X_{IN}) \le$ 25 MHz when ϕ selected as clock source for peripheral devices	$\frac{8 \times 10^9}{f(X_{IN})}$	320		
$t_{w(TAH)}$	TAi_{IN} 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 when ϕ divided by 2 selected as clock source for peripheral devices	$\frac{8 \times 10^9}{f(X_{IN})}$	320		
		$f(X_{IN}) \le$ 25 MHz when ϕ selected as clock source for peripheral devices	$\frac{4 \times 10^9}{f(X_{IN})}$	160		
$t_{w(TAL)}$	TAi_{IN} 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 when ϕ divided by 2 selected as clock source for peripheral devices	$\frac{8 \times 10^9}{f(X_{IN})}$	320		
		$f(X_{IN}) \le$ 25 MHz when ϕ selected as clock source for peripheral devices	$\frac{4 \times 10^9}{f(X_{IN})}$	160		

Notes 1: TAi_{IN} input cycle time must be 4 cycles or more of count source,
TAi_{IN} input high-level pulse width must be 2 cycles or more of count source,
TAi_{IN} input low-level pulse width must be 2 cycles or more of count source.

2: The limits in the upper row of the table are the values when $f(X_{IN})$ is 40 MHz and the count source is f_4.
The limits in the middle row of the table are the values when $f(X_{IN})$ is 25 MHz and the count source is f_4.
The limits in the lower row of the table are the values when $f(X_{IN})$ is 25 MHz and the count source is f_2.

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

Symbol	Parameter	Data formula (Min.)		Limits		Unit
				Min.	Max.	
$t_{c(TA)}$	TAi_{IN} input cycle time (**Note**)	$f(X_{IN}) \leq 40$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$	200		ns
		$f(X_{IN}) \leq 25$ MHz when ϕ divided by 2 selected as clock source for peripheral devices	$\frac{8 \times 10^9}{f(X_{IN})}$	320		
		$f(X_{IN}) \leq 25$ MHz when ϕ selected as clock source for peripheral devices	$\frac{4 \times 10^9}{f(X_{IN})}$	160		
$t_{w(TAH)}$	TAi_{IN} input high-level pulse width			80		ns
$t_{w(TAL)}$	TAi_{IN} input low-level pulse width			80		ns

Note: The limits in the upper row of the table are the values when $f(X_{IN})$ is 40 MHz and the count source is f_4.
The limits in the middle row of the table are the values when $f(X_{IN})$ is 25 MHz and the count source is f_4.
The limits in the lower row of the table are the values when $f(X_{IN})$ is 25 MHz and the count source is f_2.

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

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

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

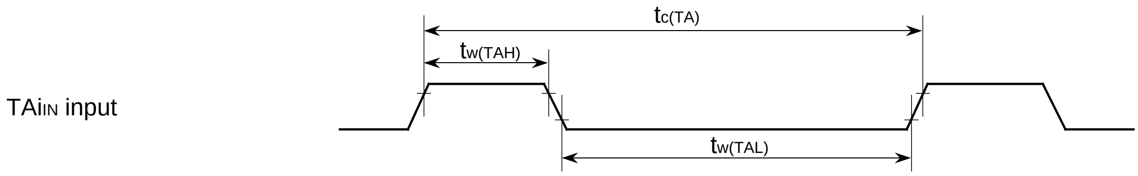
Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(UP)}$	TAi_{OUT} input cycle time	2000		ns
$t_{w(UPH)}$	TAi_{OUT} input high-level pulse width	1000		ns
$t_{w(UPL)}$	TAi_{OUT} input low-level pulse width	1000		ns
$t_{su(UP-T_{IN})}$	TAi_{OUT} input setup time	400		ns
$t_{h(T_{IN}-UP)}$	TAi_{OUT} input hold time	400		ns

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

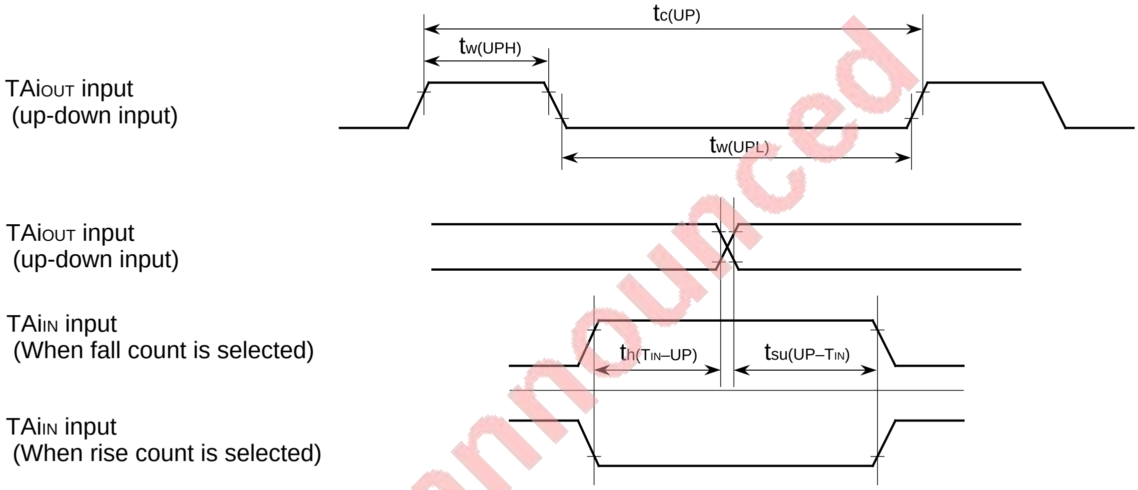
Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{su(TAj_{IN}-TAj_{OUT})}$	TAj_{IN} input setup time	200		ns
$t_{su(TAj_{OUT}-TAj_{IN})}$	TAj_{OUT} input setup time	200		ns

Internal peripheral devices

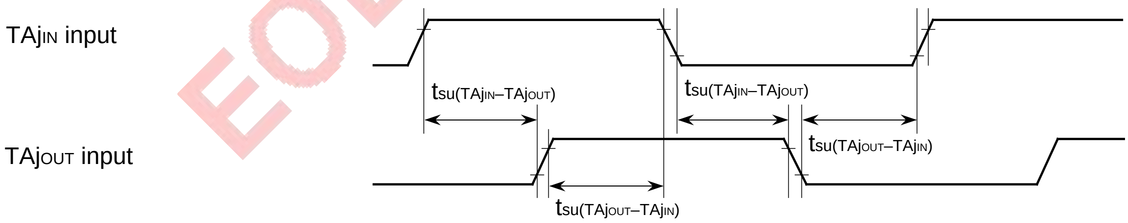
- ●Count input in event counter mode
- ●Gating input in timer mode
- ●External trigger input in one-shot pulse mode
- ●External trigger input in pulse width modulation mode



●Up-down input 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 Min.	Limits Max.	Unit
$t_{c(TB)}$	TBi$_{IN}$ input cycle time (one edge count)	80		ns
$t_{w(TBH)}$	TBi$_{IN}$ input high-level pulse width (one edge count)	40		ns
$t_{w(TBL)}$	TBi$_{IN}$ input low-level pulse width (one edge count)	40		ns
$t_{c(TB)}$	TBi$_{IN}$ input cycle time (both edges count)	160		ns
$t_{w(TBH)}$	TBi$_{IN}$ input high-level pulse width (both edges count)	80		ns
$t_{w(TBL)}$	TBi$_{IN}$ input low-level pulse width (both edges count)	80		ns

Timer B input (pulse period measurement mode)

Symbol	Parameter	Data formula (Min.)		Limits Min.	Limits Max.	Unit
$t_{c(TB)}$	TBi$_{IN}$ input cycle time	$f(X_{IN}) \leq$ 40 MHz	$\frac{16 \times 10^9}{f(X_{IN})}$	400		ns
		$f(X_{IN}) \leq$ 25 MHz when ϕ divided by 2 selected as clock source for peripheral devices	$\frac{16 \times 10^9}{f(X_{IN})}$	640		
		$f(X_{IN}) \leq$ 25 MHz when ϕ selected as clock source for peripheral devices	$\frac{8 \times 10^9}{f(X_{IN})}$	320		
$t_{w(TBH)}$	TBi$_{IN}$ input high-level pulse width	$f(X_{IN}) \leq$ 40 MHz	$\frac{8 \times 10^9}{f(X_{IN})}$	200		ns
		$f(X_{IN}) \leq$ 25 MHz when ϕ divided by 2 selected as clock source for peripheral devices	$\frac{8 \times 10^9}{f(X_{IN})}$	320		
		$f(X_{IN}) \leq$ 25 MHz when ϕ selected as clock source for peripheral devices	$\frac{4 \times 10^9}{f(X_{IN})}$	160		
$t_{w(TBL)}$	TBi$_{IN}$ input low-level pulse width	$f(X_{IN}) \leq$ 40 MHz	$\frac{8 \times 10^9}{f(X_{IN})}$	200		ns
		$f(X_{IN}) \leq$ 25 MHz when ϕ divided by 2 selected as clock source for peripheral devices	$\frac{8 \times 10^9}{f(X_{IN})}$	320		
		$f(X_{IN}) \leq$ 25 MHz when ϕ selected as clock source for peripheral devices	$\frac{4 \times 10^9}{f(X_{IN})}$	160		

Notes 1: TBi$_{IN}$ input cycle time must be 4 cycles or more of count source,
TBi$_{IN}$ input high-level pulse width must be 2 cycles or more of count source,
TBi$_{IN}$ input low-level pulse width must be 2 cycles or more of count source.

2: The limits in the upper row of the table are the values when $f(X_{IN})$ is 40 MHz and the count source is f_4.
The limits in the middle row of the table are the values when $f(X_{IN})$ is 25 MHz and the count source is f_4.
The limits in the lower row of the table are the values when $f(X_{IN})$ is 25 MHz and the count source is f_2.

15.5 Internal peripheral devices

Timer B input (Pulse width measurement mode)

Symbol	Parameter	Data formula (Min.)		Limits		Unit
				Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	$f(X_{IN}) \leq 40$ MHz	$\frac{16 \times 10^9}{f(X_{IN})}$	400		ns
		$f(X_{IN}) \leq 25$ MHz when ϕ divided by 2 selected as clock source for peripheral devices	$\frac{16 \times 10^9}{f(X_{IN})}$	640		
		$f(X_{IN}) \leq 25$ MHz when ϕ selected as clock source for peripheral devices	$\frac{8 \times 10^9}{f(X_{IN})}$	320		
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width	$f(X_{IN}) \leq 40$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$	200		ns
		$f(X_{IN}) \leq 25$ MHz when ϕ divided by 2 selected as clock source for peripheral devices	$\frac{8 \times 10^9}{f(X_{IN})}$	320		
		$f(X_{IN}) \leq 25$ MHz when ϕ selected as clock source for peripheral devices	$\frac{4 \times 10^9}{f(X_{IN})}$	160		
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width	$f(X_{IN}) \leq 40$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$	200		ns
		$f(X_{IN}) \leq 25$ MHz when ϕ divided by 2 selected as clock source for peripheral devices	$\frac{8 \times 10^9}{f(X_{IN})}$	320		
		$f(X_{IN}) \leq 25$ MHz when ϕ selected as clock source for peripheral devices	$\frac{4 \times 10^9}{f(X_{IN})}$	160		

Notes 1: TBi_{IN} input cycle time must be 4 cycles or more of count source,
TBi_{IN} input high-level pulse width must be 2 cycles or more of count source,
TBi_{IN} input low-level pulse width must be 2 cycles or more of count source.

2: The limits in the upper row of the table are the values when $f(X_{IN})$ is 40 MHz and the count source is f_4.
The limits in the middle row of the table are the values when $f(X_{IN})$ is 25 MHz and the count source is f_4.
The limits in the lower row of the table are the values when $f(X_{IN})$ is 25 MHz and the count source is f_2.

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

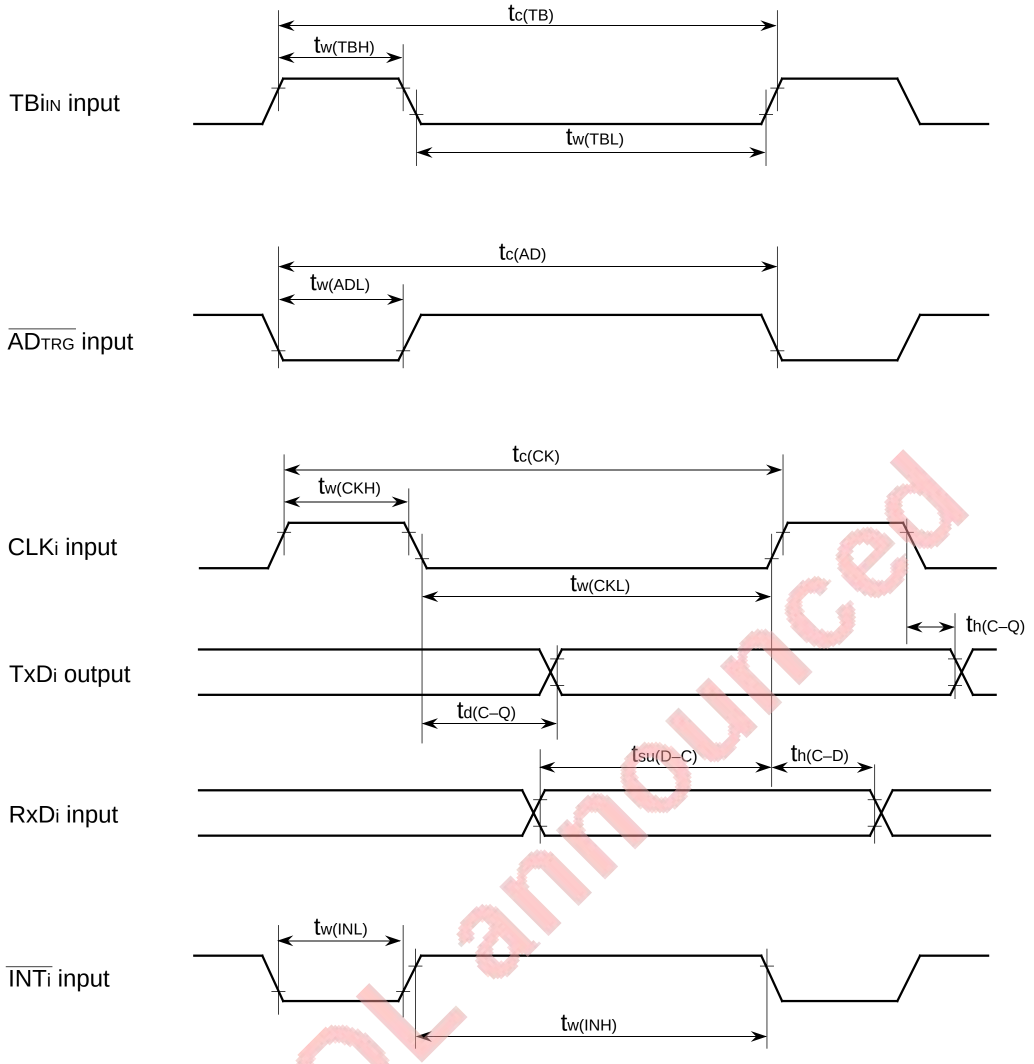
Serial I/O

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(CK)}$	CLK_i input cycle time	200		ns
$t_{w(CKH)}$	CLK_i input high-level pulse width	100		ns
$t_{w(CKL)}$	CLK_i input low-level pulse width	100		ns
$t_{d(C-Q)}$	TxD_i output delay time		80	ns
$t_{h(C-Q)}$	TxD_i hold time	0		ns
$t_{su(D-C)}$	RxD_i input setup time	20		ns
$t_{h(C-D)}$	RxD_i 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

Internal peripheral devices



Test conditions

- 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

15.6 Ready and Hold

Timing requirements (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, f(X_{IN}) = 40 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{su(RDY-\phi_1)}$	$\overline{RDY}$ input setup time	40		ns
$t_{su(HOLD-\phi_1)}$	$\overline{HOLD}$ input setup time	40		ns
$t_{h(\phi_1-RDY)}$	$\overline{RDY}$ input hold time	0		ns
$t_{h(\phi_1-HOLD)}$	$\overline{HOLD}$ input hold time	0		ns

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, f(X_{IN}) = 40 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time		50	ns

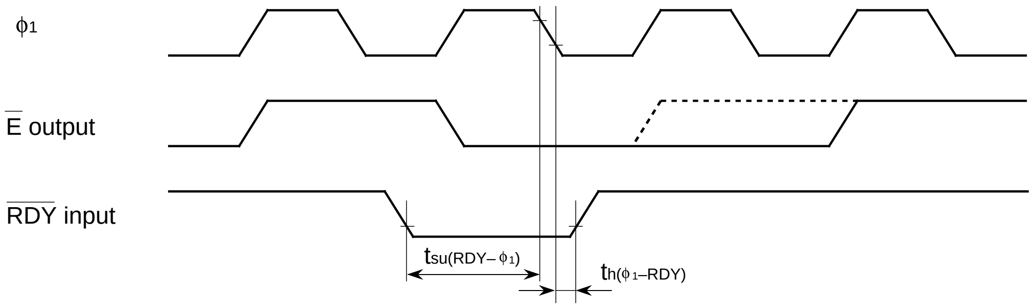
Note: For test conditions, refer to Figure 15.15.1.

ELECTRICAL CHARACTERISTICS

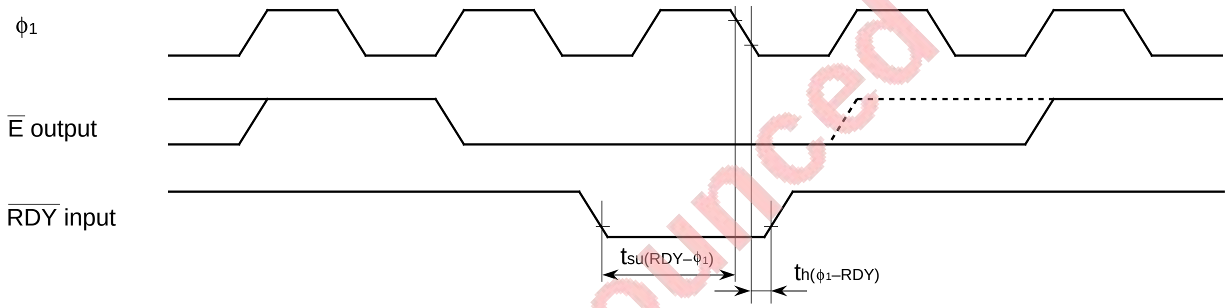
15.6 Ready and Hold

●Ready function

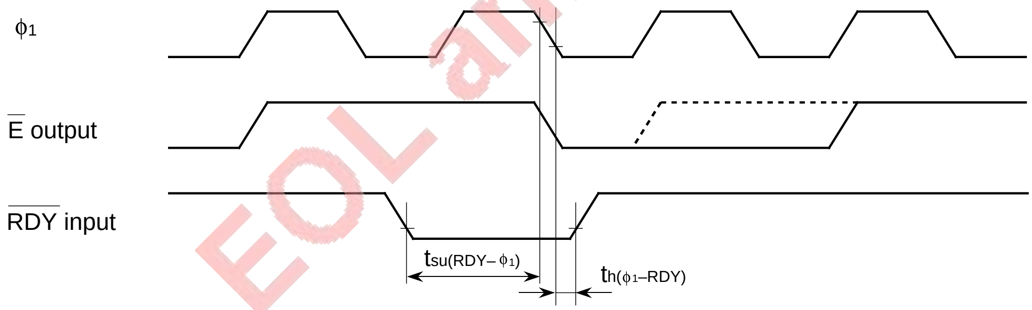
When 2-ϕ access in low-speed running



When 3-ϕ access and 4-ϕ access in low-speed running, and 4-ϕ access in high-speed running



When 2-ϕ access in high-speed running

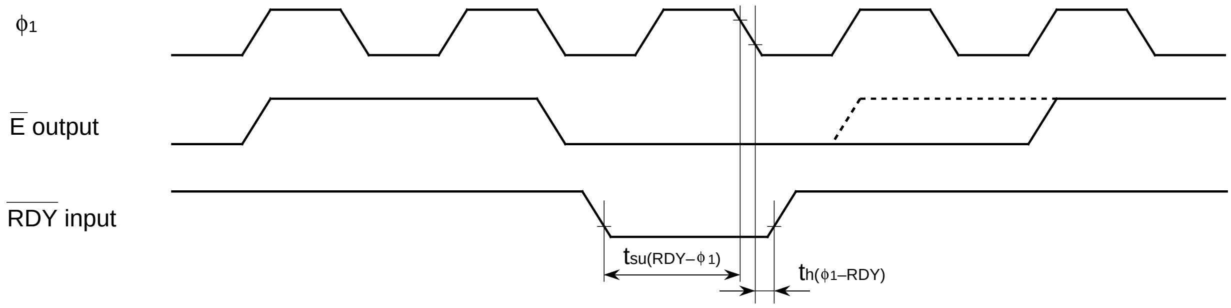


Test conditions

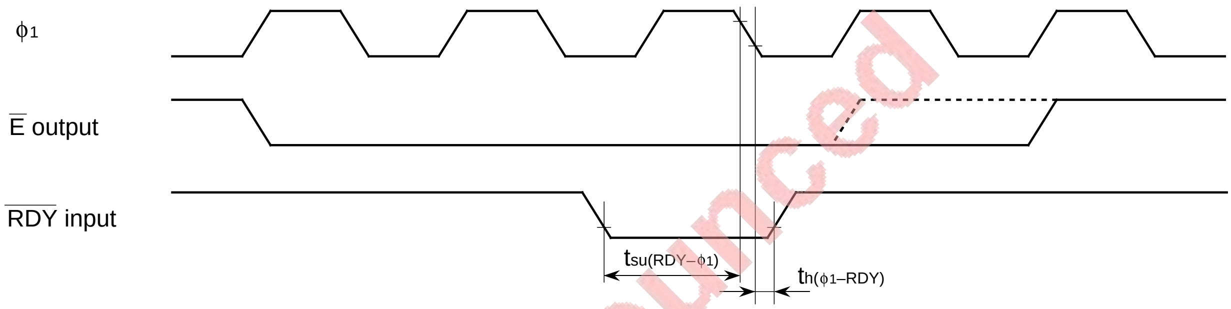
- V_{CC} = 5 V±10%
- Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V

●Ready function

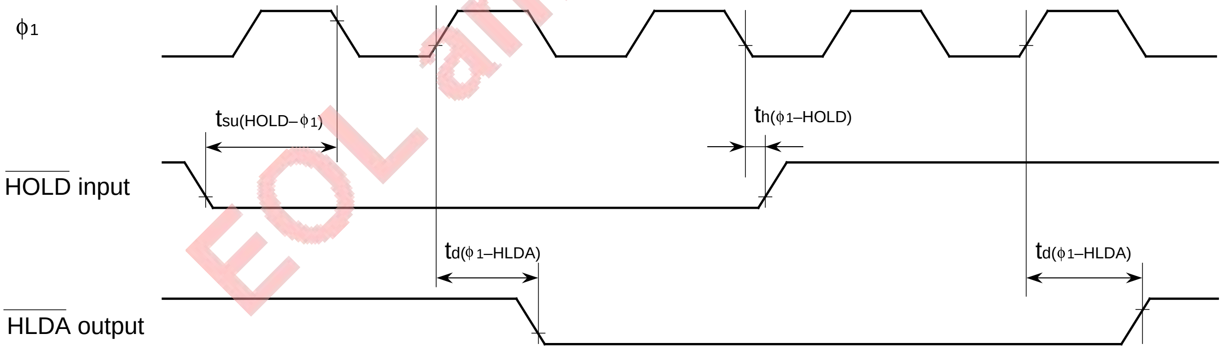
When 3-φ access in high-speed running



When 5-φ access in high-speed running



●Hold function



Test conditions
- 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

15.7 Single-chip mode

Timing requirements (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

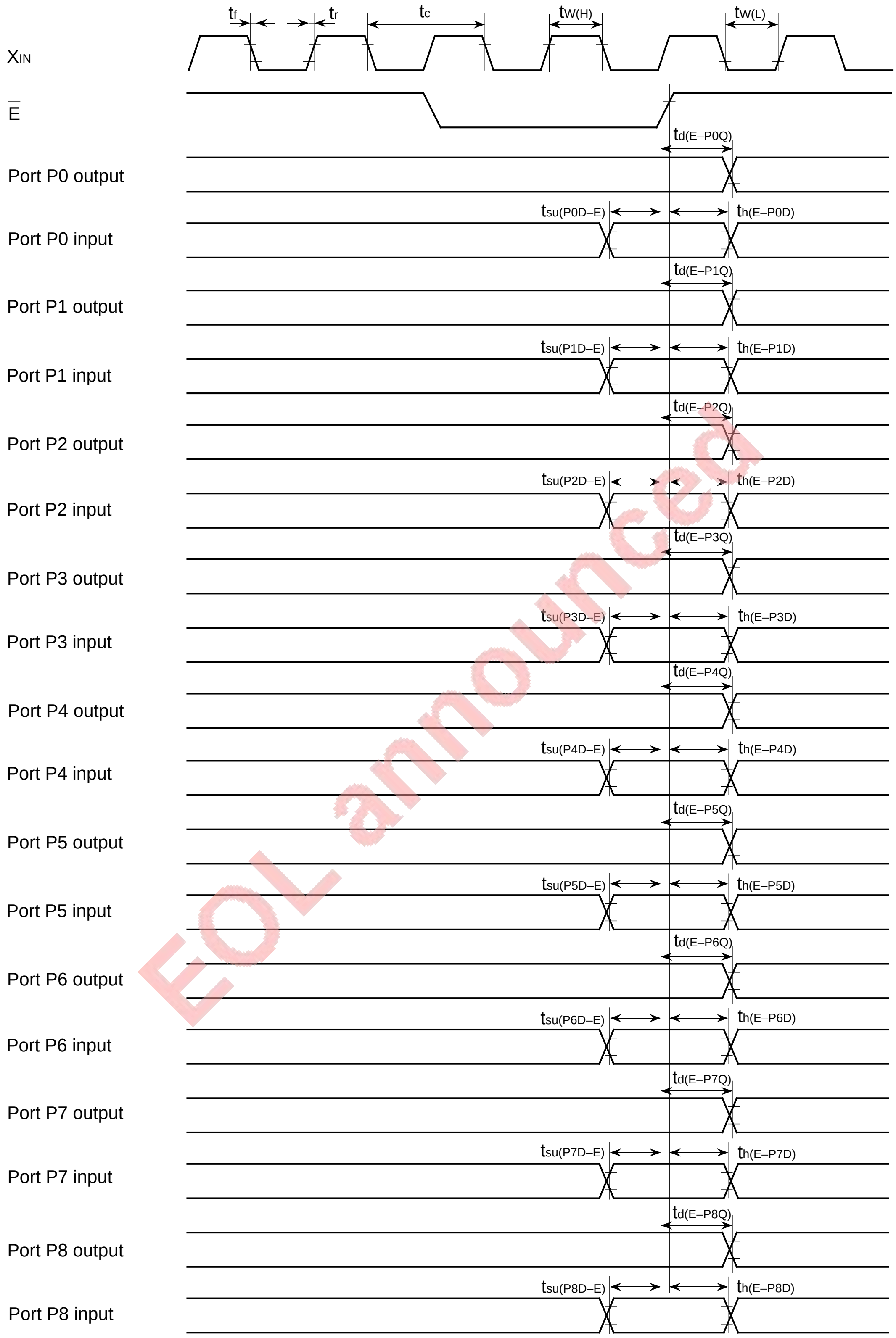
Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_c	External clock input cycle time	25		ns
$t_{w(H)}$	External clock input high-level pulse width	t_c/2–8		ns
$t_{w(L)}$	External clock input low-level pulse width	t_c/2–8		ns
t_r	External clock rise time		8	ns
t_f	External clock fall time		8	ns
$t_{su(P0D-E)}$	Port P0 input setup time	60		ns
$t_{su(P1D-E)}$	Port P1 input setup time	60		ns
$t_{su(P2D-E)}$	Port P2 input setup time	60		ns
$t_{su(P3D-E)}$	Port P3 input setup time	60		ns
$t_{su(P4D-E)}$	Port P4 input setup time	60		ns
$t_{su(P5D-E)}$	Port P5 input setup time	60		ns
$t_{su(P6D-E)}$	Port P6 input setup time	60		ns
$t_{su(P7D-E)}$	Port P7 input setup time	60		ns
$t_{su(P8D-E)}$	Port P8 input setup time	60		ns
$t_{h(E-P0D)}$	Port P0 input hold time	0		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		ns
$t_{h(E-P3D)}$	Port P3 input hold time	0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		ns

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{d(E-P0Q)}$	Port P0 data output delay time		60	ns
$t_{d(E-P1Q)}$	Port P1 data output delay time		60	ns
$t_{d(E-P2Q)}$	Port P2 data output delay time		60	ns
$t_{d(E-P3Q)}$	Port P3 data output delay time		60	ns
$t_{d(E-P4Q)}$	Port P4 data output delay time		60	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		60	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		60	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time		60	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		60	ns

Note: For test conditions, refer to Figure 15.15.1.

Single-chip mode



Test conditions
- 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

15.8 Memory expansion mode and microprocessor mode : When 2-ϕ access in low-speed running

Timing requirements (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 25 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Max.)	Limits		Unit
			Min.	Max.	
t_c	External clock input cycle time		40		ns
$t_{w(H)}$	External clock input high-level pulse width		t_c/2–8		ns
$t_{w(L)}$	External clock input low-level pulse width		t_c/2–8		ns
t_r	External clock rise time			8	ns
t_f	External clock fall time			8	ns
$t_{su(P1D-E)}$	Port P1 input setup time		30		ns
$t_{su(P2D-E)}$	Port P2 input setup time		30		ns
$t_{su(P4D-E)}$	Port P4 input setup time		60		ns
$t_{su(P5D-E)}$	Port P5 input setup time		60		ns
$t_{su(P6D-E)}$	Port P6 input setup time		60		ns
$t_{su(P7D-E)}$	Port P7 input setup time		60		ns
$t_{su(P8D-E)}$	Port P8 input setup time		60		ns
$t_{h(E-P1D)}$	Port P1 input hold time		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time		0		ns
$t_{su(P0A/P1A/P2A-P1D/P2D)}$	Port Pi data setup time with address stabilized	$\frac{3 \times 10^9}{f(X_{IN})} - 65$		55	ns

15.8 Memory expansion mode and microprocessor mode : When 2-ϕ access in low-speed running

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, f(X_{IN}) = 25 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Min.)	Limits Min.	Limits Max.	Unit
$t_{d(E-P4Q)}$	Port P4 data output delay time			60	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			60	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			60	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			60	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			60	ns
$t_{w(\phi H)}$	ϕ high-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns
$t_{w(\phi L)}$	ϕ low-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns
$t_{d(E-\phi 1)}$	ϕ_1 output delay time		0	18	ns
$t_{w(EL)}$	$\overline{E}$ low-level pulse width	$\frac{2 \times 10^9}{f(X_{IN})} - 25$	55		ns
$t_{d(P0A-E)}$	Port P0 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 28$	12		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE = “L”)			35	ns
$t_{pxz(E-P1Z)}$	Port P1 floating start delay time (BYTE = “L”)			5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 28$	12		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 35$	5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			35	ns
$t_{pxz(E-P2Z)}$	Port P2 floating start delay time			5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 28$	12		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 35$	5		ns
$t_{d(E-ALE)}$	ALE output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns
$t_{d(ALE-E)}$	ALE output delay time		4		ns
$t_{w(ALE)}$	ALE pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 18$	22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns

Note: For test conditions, refer to Figure 15.15.1.

15.8 Memory expansion mode and microprocessor mode : When 2-ϕ access in low-speed running

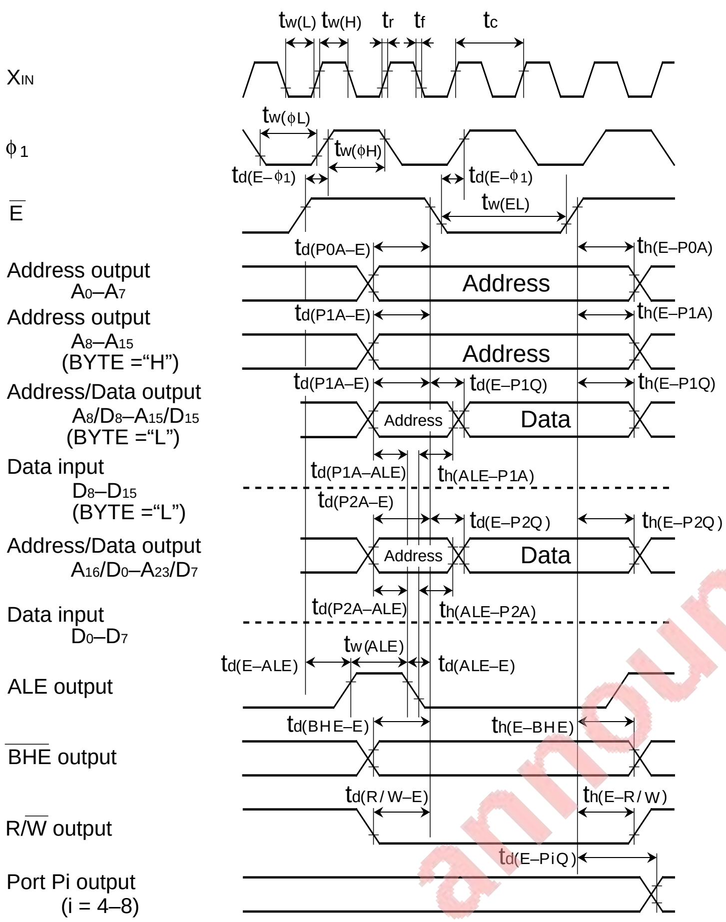
Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 25 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Min.)	Limits Min.	Limits Max.	Unit
$t_{h(E-P0A)}$	Port P0 address hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE = “L”)		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{pzx(E-P1Z)}$	Port P1 floating release delay time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE = “H”)	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{pzx(E-P2Z)}$	Port P2 floating release delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(E-RW)}$	R/$\overline{W}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns

Note: For test conditions, refer to Figure 15.15.1.

Memory expansion mode and Microprocessor mode
: When 2-ϕ access in low-speed running

<Write>



Test conditions (ϕ_1, $\overline{E}$, P0–P3)

- V_{CC} = 5 V±10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

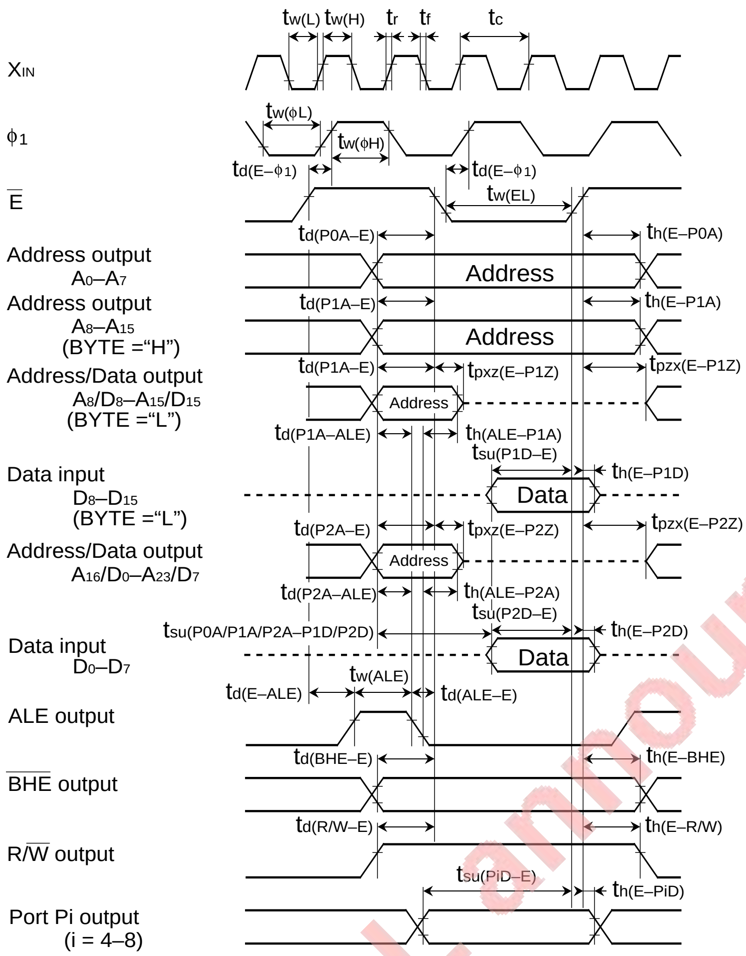
Test conditions (P4–P8)

- 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

15.8 Memory expansion mode and microprocessor mode : When 2-ϕ access in low-speed running

Memory expansion mode and Microprocessor mode
: When 2-ϕ access in low-speed running

<Read>



Test conditions (ϕ_1, $\overline{E}$, P0–P3)
- V_{CC} = 5 V±10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

Test conditions (P4–P8)
- 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

15.9 Memory expansion mode and microprocessor mode : When 3-ϕ access in low-speed running

Timing requirements (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 25 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Max.)	Limits		Unit
			Min.	Max.	
t_c	External clock input cycle time		40		ns
$t_{w(H)}$	External clock input high-level pulse width		t_c/2–8		ns
$t_{w(L)}$	External clock input low-level pulse width		t_c/2–8		ns
t_r	External clock rise time			8	ns
t_f	External clock fall time			8	ns
$t_{su(P1D–E)}$	Port P1 input setup time		30		ns
$t_{su(P2D–E)}$	Port P2 input setup time		30		ns
$t_{su(P4D–E)}$	Port P4 input setup time		60		ns
$t_{su(P5D–E)}$	Port P5 input setup time		60		ns
$t_{su(P6D–E)}$	Port P6 input setup time		60		ns
$t_{su(P7D–E)}$	Port P7 input setup time		60		ns
$t_{su(P8D–E)}$	Port P8 input setup time		60		ns
$t_{h(E–P1D)}$	Port P1 input hold time		0		ns
$t_{h(E–P2D)}$	Port P2 input hold time		0		ns
$t_{h(E–P4D)}$	Port P4 input hold time		0		ns
$t_{h(E–P5D)}$	Port P5 input hold time		0		ns
$t_{h(E–P6D)}$	Port P6 input hold time		0		ns
$t_{h(E–P7D)}$	Port P7 input hold time		0		ns
$t_{h(E–P8D)}$	Port P8 input hold time		0		ns
$t_{su(P0A/P1A/P2A–P1D/P2D)}$	Port Pi data setup time with address stabilized	$\frac{5 \times 10^9}{f(X_{IN})} - 65$		135	ns

15.9 Memory expansion mode and microprocessor mode : When 3-ϕ access in low-speed running

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 25 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Min.)	Limits		Unit
			Min.	Max.	
$t_{d(E-P4Q)}$	Port P4 data output delay time			60	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			60	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			60	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			60	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			60	ns
$t_{w(\phi H)}$	ϕ high-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns
$t_{w(\phi L)}$	ϕ low-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	18	ns
$t_{w(EL)}$	$\overline{E}$ low-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})} - 25$	135		ns
$t_{d(P0A-E)}$	Port P0 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 28$	12		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE = “L”)			35	ns
$t_{pxz(E-P1Z)}$	Port P1 floating start delay time (BYTE = “L”)			5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 28$	12		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 35$	5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			35	ns
$t_{pxz(E-P2Z)}$	Port P2 floating start delay time			5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 28$	12		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 35$	5		ns
$t_{d(E-ALE)}$	ALE output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns
$t_{d(ALE-E)}$	ALE output delay time		4		ns
$t_{w(ALE)}$	ALE pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 18$	22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns

Note: For test conditions, refer to Figure 15.15.1.

15.9 Memory expansion mode and microprocessor mode : When 3-ϕ access in low-speed running

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 25 MHz, unless otherwise noted)

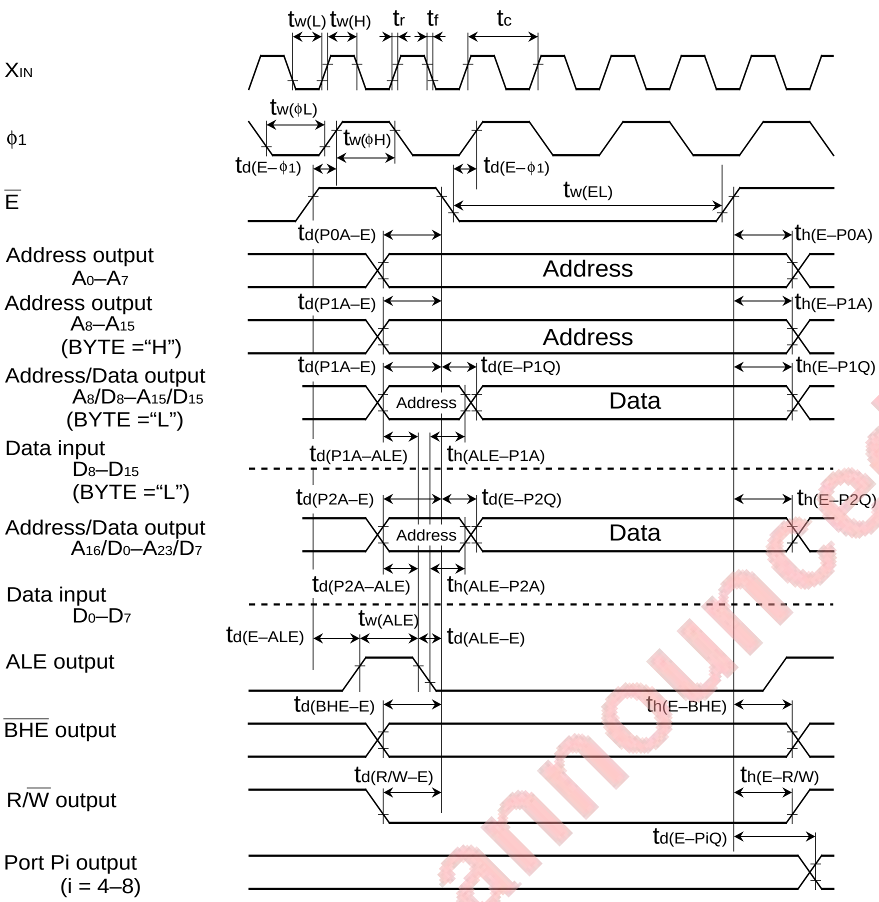
Symbol	Parameter	Data formula (Min.)	Limits		Unit
			Min.	Max.	
$t_{h(E-P0A)}$	Port P0 address hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE = "L")		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE = "L")	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{pzx(E-P1Z)}$	Port P1 floating release delay time (BYTE = "L")	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE = "H")	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{pzx(E-P2Z)}$	Port P2 floating release delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(E-RW)}$	R/$\overline{W}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns

Note: For test conditions, refer to Figure 15.15.1.

15.9 Memory expansion mode and microprocessor mode : When 3-ϕ access in low-speed running

Memory expansion mode and Microprocessor mode
: When 3-ϕ access in low-speed running

<Write>



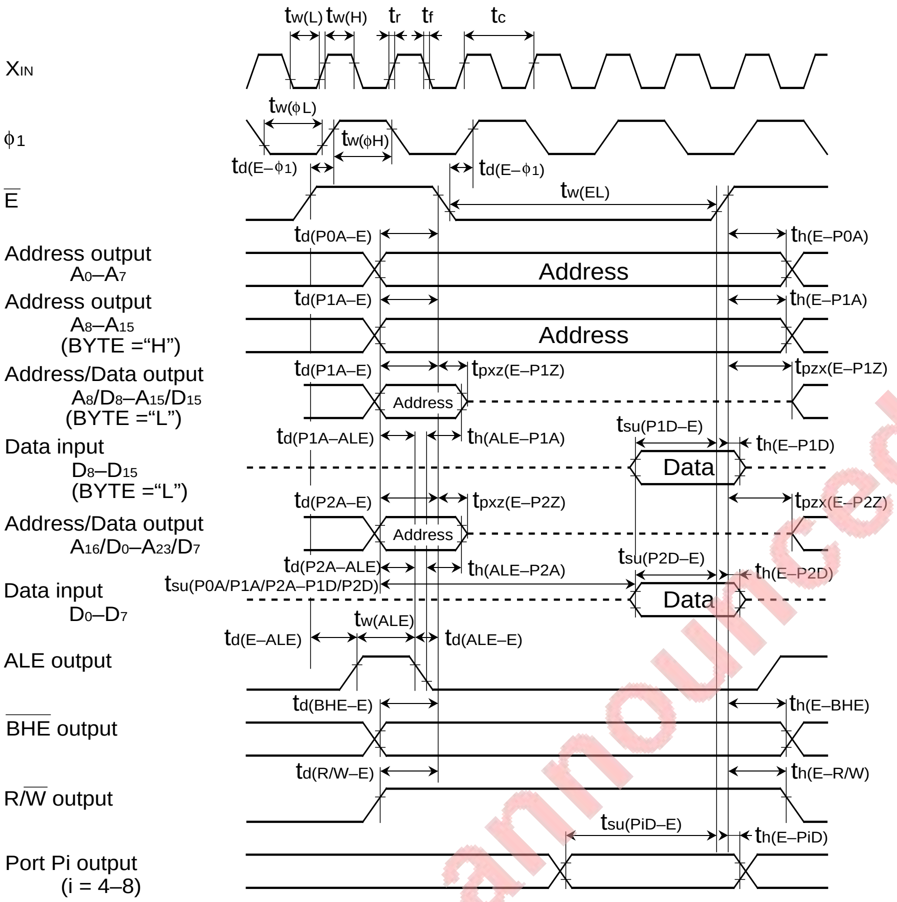
Test conditions (ϕ1, $\overline{E}$, P0–P3)
- V_{CC} = 5 V±10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

Test conditions (P4–P8)
- 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

15.9 Memory expansion mode and microprocessor mode : When 3-ϕ access in low-speed running

Memory expansion mode and Microprocessor mode
: When 3-ϕ access in low-speed running

<Read>



Test conditions (ϕ_1, $\overline{E}$, P0–P3)

•V_{CC} = 5 V±10%

•Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V

•Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

Test conditions (P4–P8)

•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

15.10 Memory expansion mode and microprocessor mode : When 4-φ access in low-speed running

Timing requirements (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, f(X_{IN}) = 25 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Max.)	Limits		Unit
			Min.	Max.	
t_c	External clock input cycle time		40		ns
$t_{w(H)}$	External clock input high-level pulse width		tc/2–8		ns
$t_{w(L)}$	External clock input low-level pulse width		tc/2–8		ns
t_r	External clock rise time			8	ns
t_f	External clock fall time			8	ns
$t_{su(P1D-E)}$	Port P1 input setup time		30		ns
$t_{su(P2D-E)}$	Port P2 input setup time		30		ns
$t_{su(P4D-E)}$	Port P4 input setup time		60		ns
$t_{su(P5D-E)}$	Port P5 input setup time		60		ns
$t_{su(P6D-E)}$	Port P6 input setup time		60		ns
$t_{su(P7D-E)}$	Port P7 input setup time		60		ns
$t_{su(P8D-E)}$	Port P8 input setup time		60		ns
$t_{h(E-P1D)}$	Port P1 input hold time		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time		0		ns
$t_{su(P0A/P1A/P2A-P1D/P2D)}$	Port Pi data setup time with address stabilized	$\frac{7 \times 10^9}{f(X_{IN})}-65$		215	ns

15.10 Memory expansion mode and microprocessor mode : When 4-ϕ access in low-speed running

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, f(X_{IN}) = 25 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Min.)	Limits		Unit
			Min.	Max.	
$t_{d(E-P4Q)}$	Port P4 data output delay time			60	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			60	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			60	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			60	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			60	ns
$t_{w(\phi H)}$	ϕ high-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns
$t_{w(\phi L)}$	ϕ low-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns
$t_{d(E-\phi 1)}$	ϕ_1 output delay time		0	18	ns
$t_{w(EL)}$	$\overline{E}$ low-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})} - 25$	135		ns
$t_{d(P0A-E)}$	Port P0 address output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 28$	92		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE = "L")			35	ns
$t_{pxz(E-P1Z)}$	Port P1 floating start delay time (BYTE = "L")			5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 28$	92		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 28$	52		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			35	ns
$t_{pxz(E-P2Z)}$	Port P2 floating start delay time			5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 28$	92		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 28$	52		ns
$t_{d(E-ALE)}$	ALE output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	20		ns
$t_{d(ALE-E)}$	ALE output delay time		4		ns
$t_{w(ALE)}$	ALE pulse width	$\frac{2 \times 10^9}{f(X_{IN})} - 18$	62		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 20$	100		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 20$	100		ns

Note: For test conditions, refer to Figure 15.15.1.

ELECTRICAL CHARACTERISTICS

15.10 Memory expansion mode and microprocessor mode : When 4-ϕ access in low-speed running

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 25 MHz, unless otherwise noted)

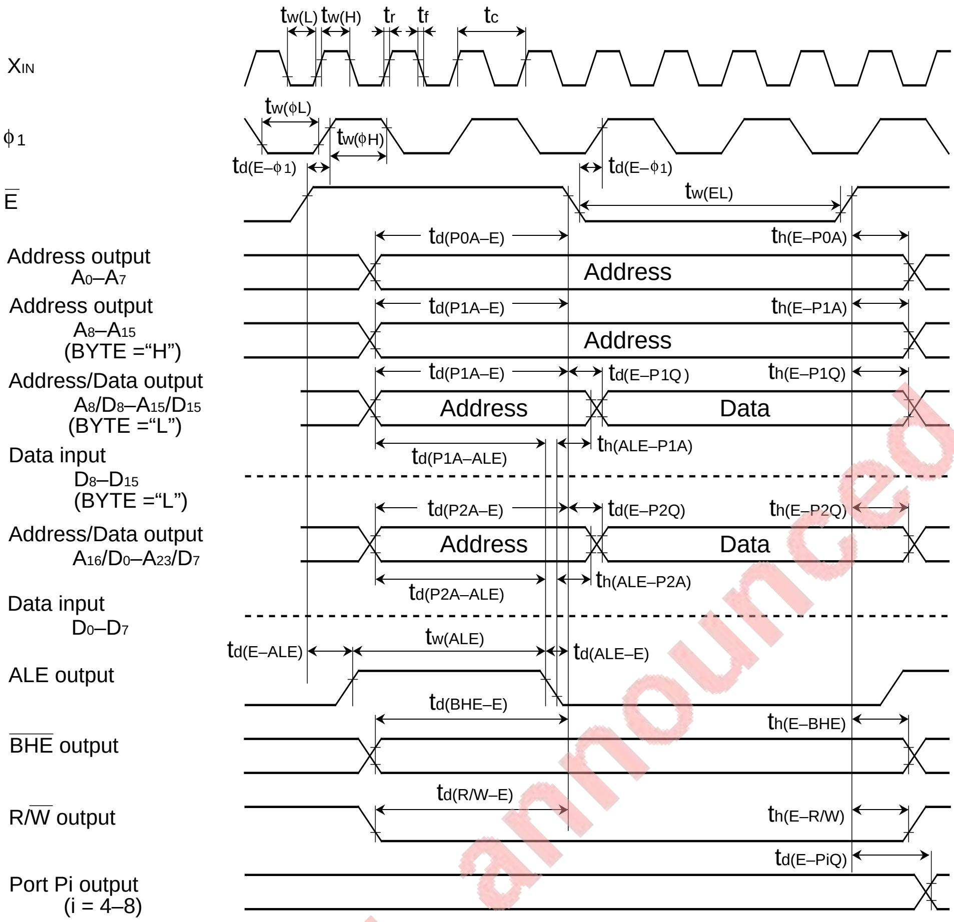
Symbol	Parameter	Data formula (Min.)	Limits		Unit
			Min.	Max.	
$t_{h(E-P0A)}$	Port P0 address hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	25		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{pzx(E-P1Z)}$	Port P1 floating release delay time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE = “H”)	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	25		ns
$t_{h(E-P2Q)}$	Port P2 data hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{pzx(E-P2Z)}$	Port P2 floating release delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns
$t_{h(E-RW)}$	R/$\overline{W}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	18		ns

Note: For test conditions, refer to Figure 15.15.1.

15.10 Memory expansion mode and microprocessor mode : When 4-ϕ access in low-speed runninge

Memory expansion mode and Microprocessor mode
: When 4-ϕ access in low-speed running

<Write>



Test conditions (ϕ_1, $\overline{E}$, P0–P3)

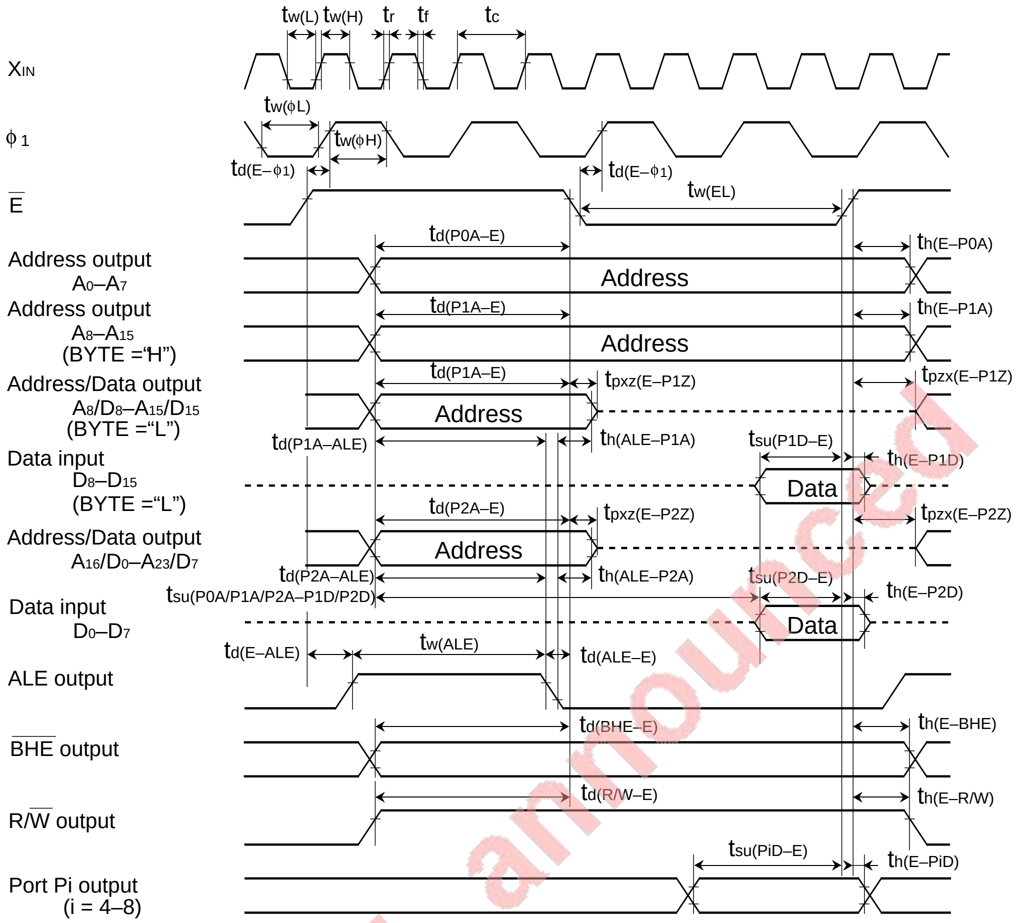
- V_{CC} = 5 V±10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

Test conditions (P4–P8)

- 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

15.10 Memory expansion mode and microprocessor mode : When 4-ϕ access in low-speed runninge

Memory expansion mode and Microprocessor mode
: When 4-ϕ access in low-speed running
<Read>



Test conditions (ϕ_1, $\overline{E}$, P0–P3)
- V_{CC} = 5 V±10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

Test conditions (P4–P8)
- 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

15.11 Memory expansion mode and microprocessor mode : When 3-ϕ access in high-speed running

Timing requirements (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, f(X_{IN}) = 40 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Max.)	Limits		Unit
			Min.	Max.	
t_c	External clock input cycle time		25		ns
$t_{w(H)}$	External clock input high-level pulse width		t_c/2–8		ns
$t_{w(L)}$	External clock input low-level pulse width		t_c/2–8		ns
t_r	External clock rise time			8	ns
t_f	External clock fall time			8	ns
$t_{su(P1D-E)}$	Port P1 input setup time		30		ns
$t_{su(P2D-E)}$	Port P2 input setup time		30		ns
$t_{su(P4D-E)}$	Port P4 input setup time		60		ns
$t_{su(P5D-E)}$	Port P5 input setup time		60		ns
$t_{su(P6D-E)}$	Port P6 input setup time		60		ns
$t_{su(P7D-E)}$	Port P7 input setup time		60		ns
$t_{su(P8D-E)}$	Port P8 input setup time		60		ns
$t_{h(E-P1D)}$	Port P1 input hold time		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time		0		ns
$t_{su(P0A/P1A/P2A-P1D/P2D)}$	Port Pi data setup time with address stabilized	$\frac{5 \times 10^9}{f(X_{IN})} - 75$		50	ns

15.11 Memory expansion mode and microprocessor mode : When 3-ϕ access in high-speed running

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, f(X_{IN}) = 40 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Min.)	Limits		Unit
			Min.	Max.	
$t_{d(E-P4Q)}$	Port P4 data output delay time			60	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			60	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			60	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			60	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			60	ns
$t_{w(\phi H)}$	ϕ high-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{w(\phi L)}$	ϕ low-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	18	ns
$t_{w(EL)}$	$\overline{E}$ low-level pulse width	$\frac{3 \times 10^9}{f(X_{IN})} - 25$	50		ns
$t_{d(P0A-E)}$	Port P0 address output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 35$	15		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE = "L")			35	ns
$t_{pxz(E-P1Z)}$	Port P1 floating start delay time (BYTE = "L")			5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 35$	15		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			35	ns
$t_{pxz(E-P2Z)}$	Port P2 floating start delay time			5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 35$	15		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{d(E-ALE)}$	ALE output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{d(ALE-E)}$	ALE output delay time	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 7.5$	5		ns
$t_{w(ALE)}$	ALE pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 30$	20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 30$	20		ns

Note: For test conditions, refer to Figure 15.15.1.

15.11 Memory expansion mode and microprocessor mode : When 3-ϕ access in high-speed running

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 40 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Min.)	Limits		Unit
			Min.	Max.	
$t_{h(E-P0A)}$	Port P0 address hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{pzx(E-P1Z)}$	Port P1 floating release delay time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE = “H”)	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{h(E-P2Q)}$	Port P2 data hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{pzx(E-P2Z)}$	Port P2 floating release delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(E-RW)}$	R/$\overline{W}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns

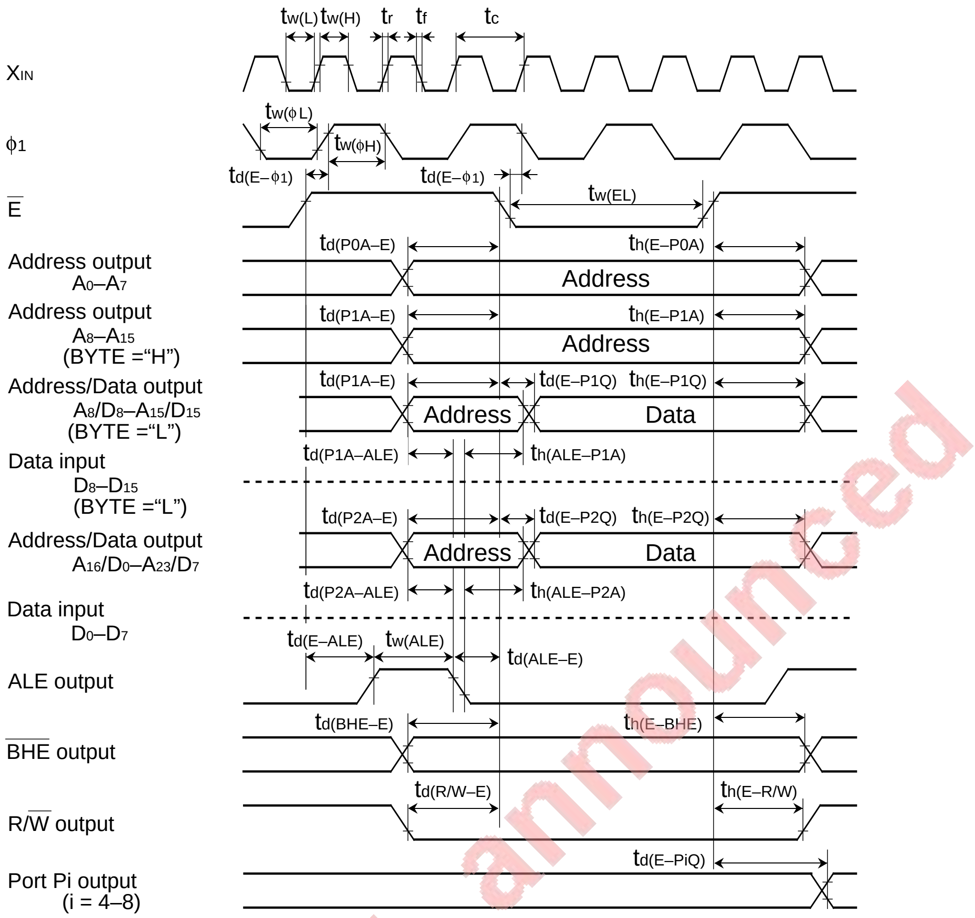
Note: For test conditions, refer to Figure 15.15.1.

ELECTRICAL CHARACTERISTICS

15.11 Memory expansion mode and microprocessor mode : When 3-ϕ access in high-speed running

Memory expansion mode and Microprocessor mode
: When 3-ϕ access in high-speed running

<Write>



Test conditions (φ1, E, P0–P3)

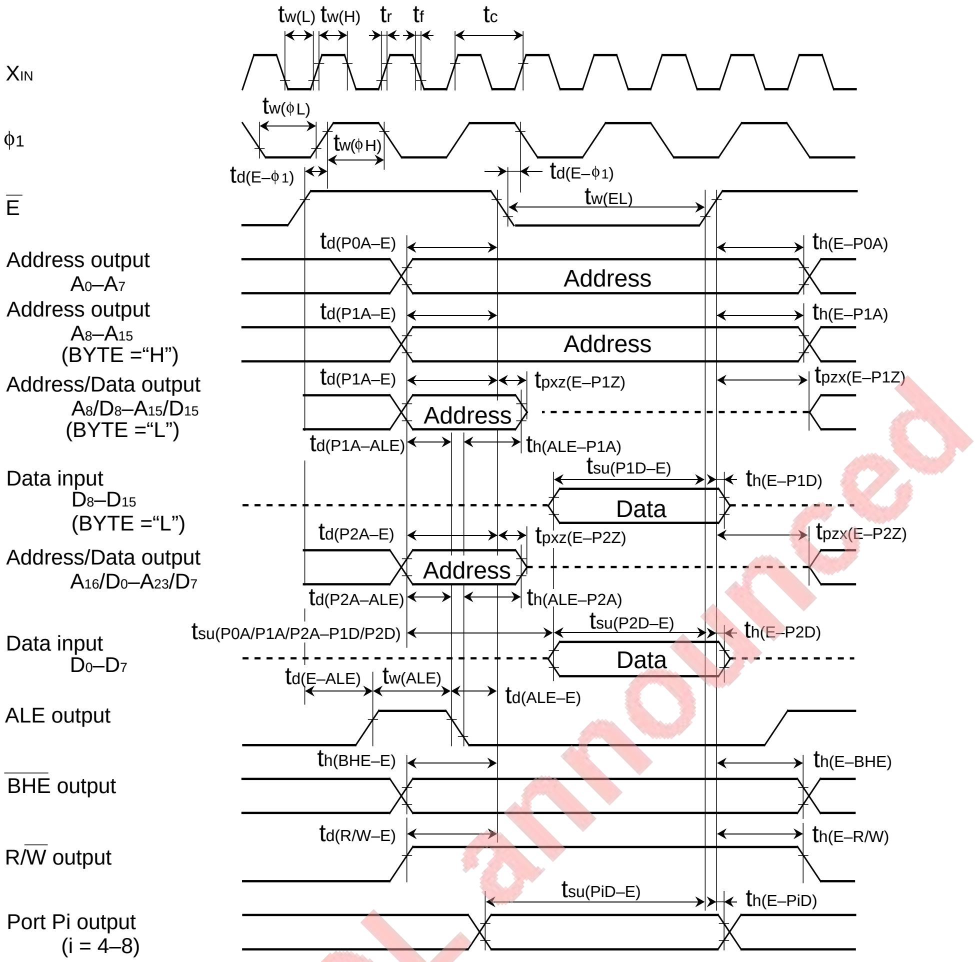
- V_{CC} = 5 V±10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

Test conditions (P4–P8)

- 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

Memory expansion mode and Microprocessor mode
: When 3-ϕ access in high-speed running

<Read>



Test conditions (ϕ_1, $\overline{E}$, P0–P3)
- V_{CC} = 5 V±10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

Test conditions (P4–P8)
- 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

15.12 Memory expansion mode and microprocessor mode : When 4-ϕ access in high-speed running

Timing requirements (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 40 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Max.)	Limits		Unit
			Min.	Max.	
t_c	External clock input cycle time		25		ns
$t_{w(H)}$	External clock input high-level pulse width		t_c/2–8		ns
$t_{w(L)}$	External clock input low-level pulse width		t_c/2–8		ns
t_r	External clock rise time			8	ns
t_f	External clock fall time			8	ns
$t_{su(P1D-E)}$	Port P1 input setup time		30		ns
$t_{su(P2D-E)}$	Port P2 input setup time		30		ns
$t_{su(P4D-E)}$	Port P4 input setup time		60		ns
$t_{su(P5D-E)}$	Port P5 input setup time		60		ns
$t_{su(P6D-E)}$	Port P6 input setup time		60		ns
$t_{su(P7D-E)}$	Port P7 input setup time		60		ns
$t_{su(P8D-E)}$	Port P8 input setup time		60		ns
$t_{h(E-P1D)}$	Port P1 input hold time		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time		0		ns
$t_{su(P0A/P1A/P2A-P1D/P2D)}$	Port Pi data setup time with address stabilized	$\frac{7 \times 10^9}{f(X_{IN})} - 75$		100	ns

15.12 Memory expansion mode and microprocessor mode : When 4-ϕ access in high-speed running

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, f(X_{IN}) = 40 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Min.)	Limits		Unit
			Min.	Max.	
$t_{d(E-P4Q)}$	Port P4 data output delay time			60	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			60	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			60	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			60	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			60	ns
$t_{w(\phi H)}$	ϕ high-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{w(\phi L)}$	ϕ low-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	18	ns
$t_{w(EL)}$	$\overline{E}$ low-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})} - 25$	75		ns
$t_{d(P0A-E)}$	Port P0 address output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 35$	40		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE = “L”)			35	ns
$t_{pxz(E-P1Z)}$	Port P1 floating start delay time (BYTE = “L”)			5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 35$	40		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 20$	30		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			35	ns
$t_{pxz(E-P2Z)}$	Port P2 floating start delay time			5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 35$	40		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 20$	30		ns
$t_{d(E-ALE)}$	ALE output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{d(ALE-E)}$	ALE output delay time	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 7.5$	5		ns
$t_{w(ALE)}$	ALE pulse width	$\frac{2 \times 10^9}{f(X_{IN})} - 15$	35		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 30$	45		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 30$	45		ns

Note: For test conditions, refer to Figure 15.15.1.

15.12 Memory expansion mode and microprocessor mode : When 4-ϕ access in high-speed running

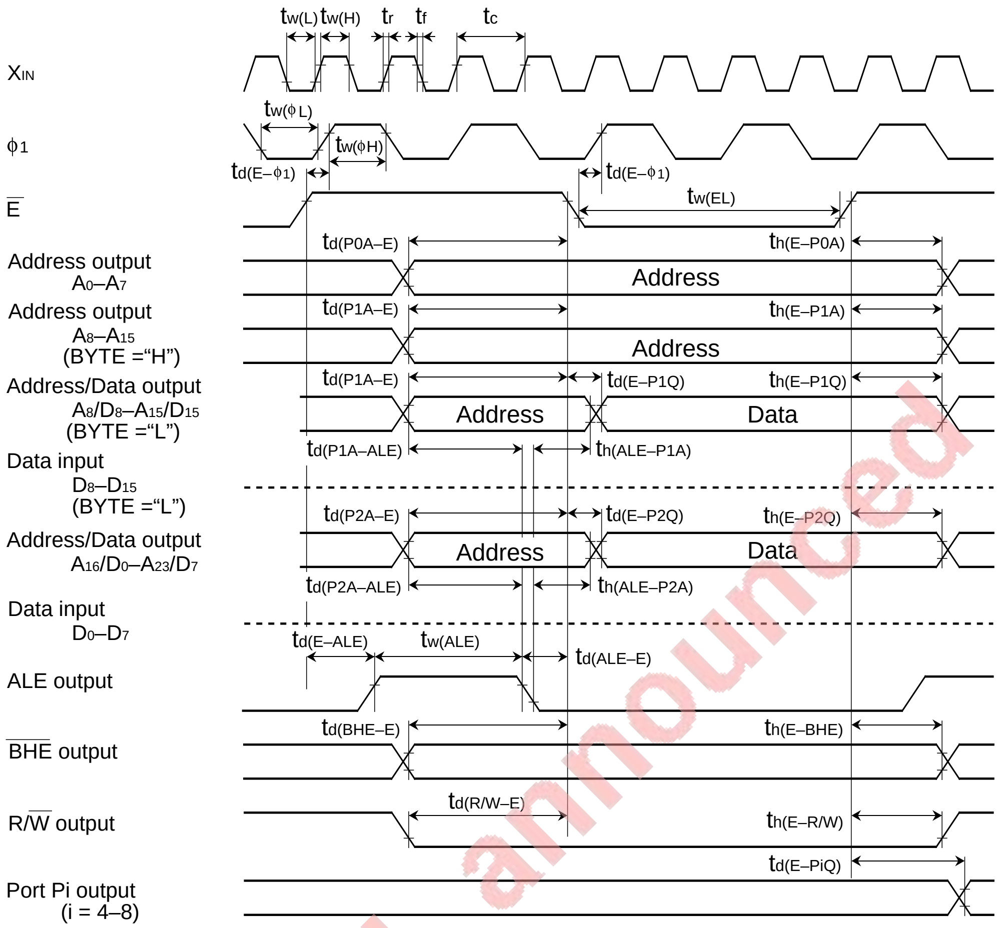
Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 40 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Min.)	Limits		Unit
			Min.	Max.	
$t_{h(E-P0A)}$	Port P0 address hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{pzx(E-P1Z)}$	Port P1 floating release delay time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE = “H”)	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{h(E-P2Q)}$	Port P2 data hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{pzx(E-P2Z)}$	Port P2 floating release delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(E-RW)}$	R/$\overline{W}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns

Note: For test conditions, refer to Figure 15.15.1.

15.12 Memory expansion mode and microprocessor mode : When 4-ϕ access in high-speed running

Memory expansion mode and Microprocessor mode
: When 4-ϕ access in high-speed running
<Write>



Test conditions (ϕ1, E, P0–P3)

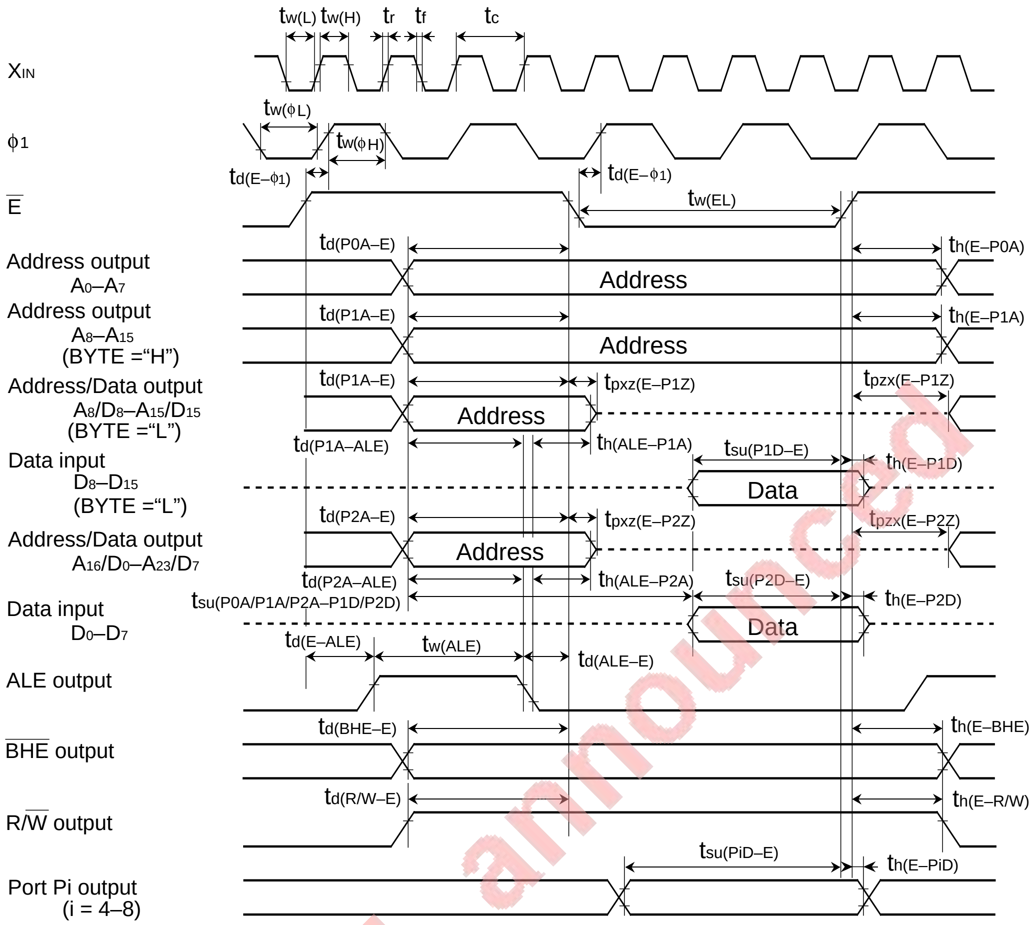
- Vcc = 5 V±10%
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V
- Data input : VIL = 0.8 V, VIH = 2.5 V

Test conditions (P4–P8)

- 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

15.12 Memory expansion mode and microprocessor mode : When 4-ϕ access in high-speed running

Memory expansion mode and Microprocessor mode
: When 4-ϕ access in high-speed running
<Read>



Test conditions (ϕ_1, $\overline{E}$, P0–P3)
- V_{CC} = 5 V±10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

Test conditions (P4–P8)
- 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

15.13 Memory expansion mode and microprocessor mode : When 5-ϕ access in high-speed running

Timing requirements (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 40 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Max.)	Limits Min.	Limits Max.	Unit
t_c	External clock input cycle time		25		ns
$t_{w(H)}$	External clock input high-level pulse width		t_c/2–8		ns
$t_{w(L)}$	External clock input low-level pulse width		t_c/2–8		ns
t_r	External clock rise time			8	ns
t_f	External clock fall time			8	ns
$t_{su(P1D–E)}$	Port P1 input setup time		30		ns
$t_{su(P2D–E)}$	Port P2 input setup time		30		ns
$t_{su(P4D–E)}$	Port P4 input setup time		60		ns
$t_{su(P5D–E)}$	Port P5 input setup time		60		ns
$t_{su(P6D–E)}$	Port P6 input setup time		60		ns
$t_{su(P7D–E)}$	Port P7 input setup time		60		ns
$t_{su(P8D–E)}$	Port P8 input setup time		60		ns
$t_{h(E–P1D)}$	Port P1 input hold time		0		ns
$t_{h(E–P2D)}$	Port P2 input hold time		0		ns
$t_{h(E–P4D)}$	Port P4 input hold time		0		ns
$t_{h(E–P5D)}$	Port P5 input hold time		0		ns
$t_{h(E–P6D)}$	Port P6 input hold time		0		ns
$t_{h(E–P7D)}$	Port P7 input hold time		0		ns
$t_{h(E–P8D)}$	Port P8 input hold time		0		ns
$t_{su(P0A/P1A/P2A–P1D/P2D)}$	Port Pi data setup time with address stabilized	$\frac{9 \times 10^9}{f(X_{IN})} - 75$		150	ns

15.13 Memory expansion mode and microprocessor mode : When 5-ϕ access in high-speed running

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 40 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Min.)	Limits		Unit
			Min.	Max.	
$t_{d(E-P4Q)}$	Port P4 data output delay time			60	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			60	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			60	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			60	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			60	ns
$t_{w(\phi H)}$	ϕ high-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{w(\phi L)}$	ϕ low-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	18	ns
$t_{w(EL)}$	$\overline{E}$ low-level pulse width	$\frac{6 \times 10^9}{f(X_{IN})} - 25$	125		ns
$t_{d(P0A-E)}$	Port P0 address output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 35$	40		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE = “L”)			35	ns
$t_{pxz(E-P1Z)}$	Port P1 floating start delay time (BYTE = “L”)			5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 35$	40		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 20$	30		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			35	ns
$t_{pxz(E-P2Z)}$	Port P2 floating start delay time			5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 35$	40		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 20$	30		ns
$t_{d(E-ALE)}$	ALE output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{d(ALE-E)}$	ALE output delay time	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 7.5$	5		ns
$t_{w(ALE)}$	ALE pulse width	$\frac{2 \times 10^9}{f(X_{IN})} - 15$	35		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 30$	45		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	$\frac{3 \times 10^9}{f(X_{IN})} - 30$	45		ns

Note: For test conditions, refer to Figure 15.15.1.

15.13 Memory expansion mode and microprocessor mode : When 5-ϕ access in high-speed running

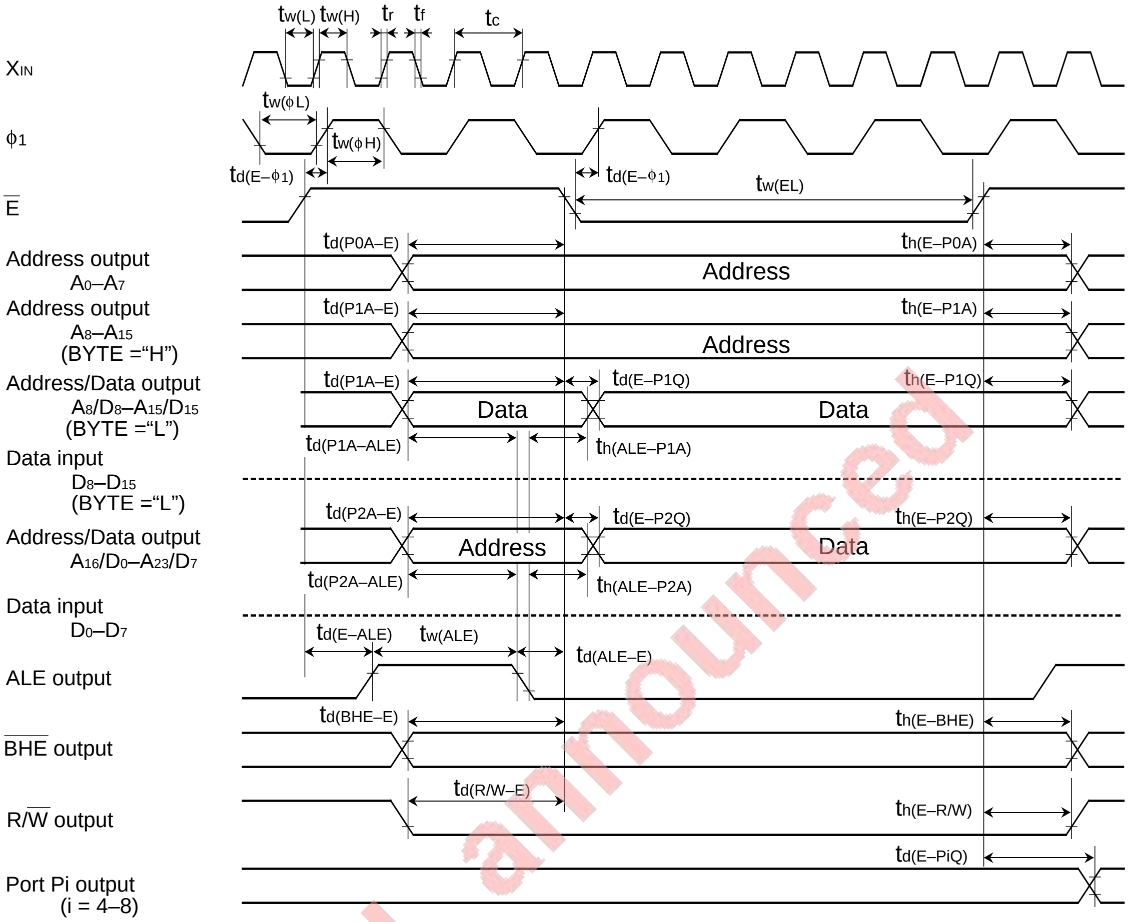
Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 40 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Min.)	Limits		Unit
			Min.	Max.	
$t_{h(E-P0A)}$	Port P0 address hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{pzx(E-P1Z)}$	Port P1 floating release delay time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE = “H”)	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{h(E-P2Q)}$	Port P2 data hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{pzx(E-P2Z)}$	Port P2 floating release delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(E-RW)}$	R/$\overline{W}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns

Note: For test conditions, refer to Figure 15.15.1.

15.13 Memory expansion mode and microprocessor mode : When 5-ϕ access in high-speed running

Memory expansion mode and Microprocessor mode
: When 5-ϕ access in high-speed running
<Write>

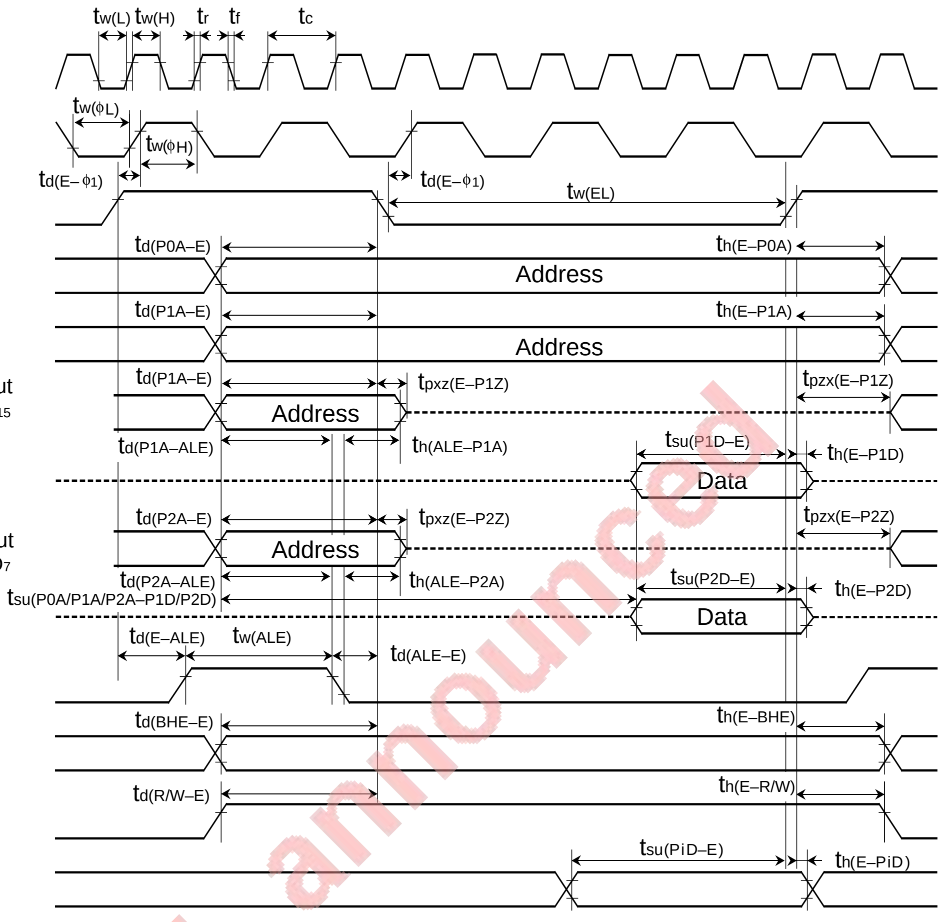


Test conditions (ϕ_1, $\overline{E}$, P0–P3)
- V_{CC} = 5 V±10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

Test conditions (P4–P8)
- 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

15.13 Memory expansion mode and microprocessor mode : When 5-ϕ access in high-speed running

Memory expansion mode and Microprocessor mode
: When 5-ϕ access in high-speed running
<Read>



Test conditions (ϕ_1, $\overline{E}$, P0–P3)
- V_{CC} = 5 V±10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

Test conditions (P4–P8)
- 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

15.14 Memory expansion mode and microprocessor mode : When 2-ϕ access in high-speed running (Internal RAM access)

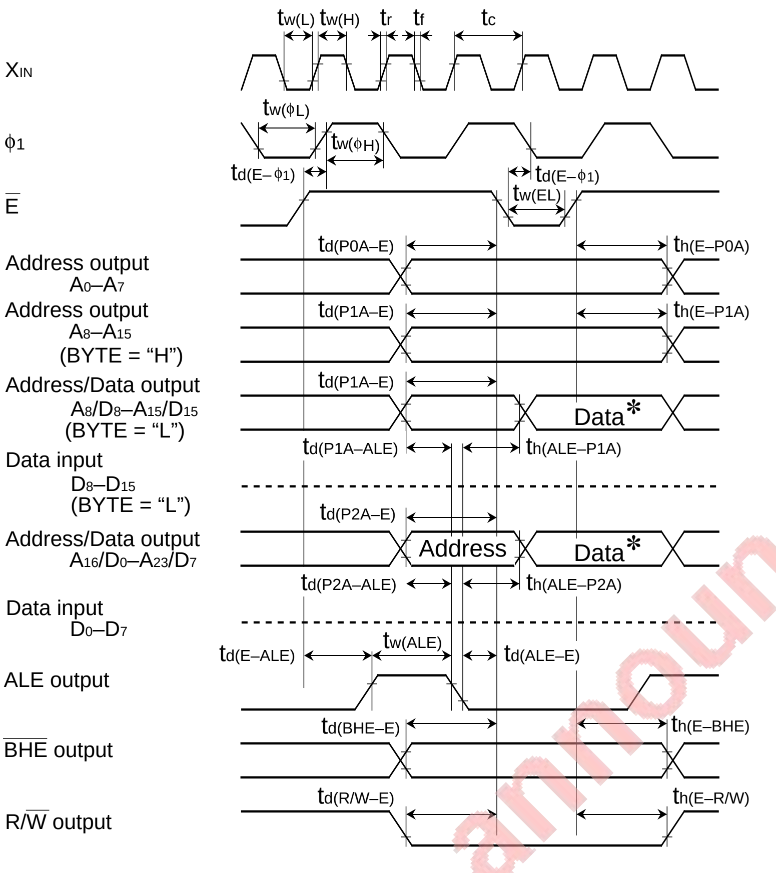
Timing requirements (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, $f(X_{IN})$ = 40 MHz, unless otherwise noted)

Symbol	Parameter	Data formula (Min.)	Limits Min.	Limits Max.	Unit
$t_{w(\phi H)}$	ϕ high-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{w(\phi L)}$	ϕ low-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	18	ns
$t_{w(EL)}$	$\overline{E}$ low-level pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{d(P0A-E)}$	Port P0 address output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 35$	15		ns
$t_{pxz(E-P1Z)}$	Port P1 floating start delay time (BYTE = “L”)			5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 35$	15		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{pxz(E-P2Z)}$	Port P2 floating start delay time			5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 35$	15		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 20$	5		ns
$t_{d(E-ALE)}$	ALE output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{d(ALE-E)}$	ALE output delay time	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 7.5$	5		ns
$t_{w(ALE)}$	ALE pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 30$	20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	$\frac{2 \times 10^9}{f(X_{IN})} - 30$	20		ns
$t_{h(E-P0A)}$	Port P0 address hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{pzx(E-P1Z)}$	Port P1 floating release delay time (BYTE = “L”)	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE = “H”)	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 15$	10		ns
$t_{pzx(E-P2Z)}$	Port P2 floating release delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	15		ns

15.14 Memory expansion mode and microprocessor mode : When 2-ϕ access in high-speed running (Internal RAM access)

Memory expansion mode and Microprocessor mode
: When 2-ϕ access in high-speed running (Internal RAM access)

<Write>



✻ The undefined value is output.

Test conditions (ϕ_1, $\overline{E}$, P0–P3)

•V_{CC} = 5 V±10%

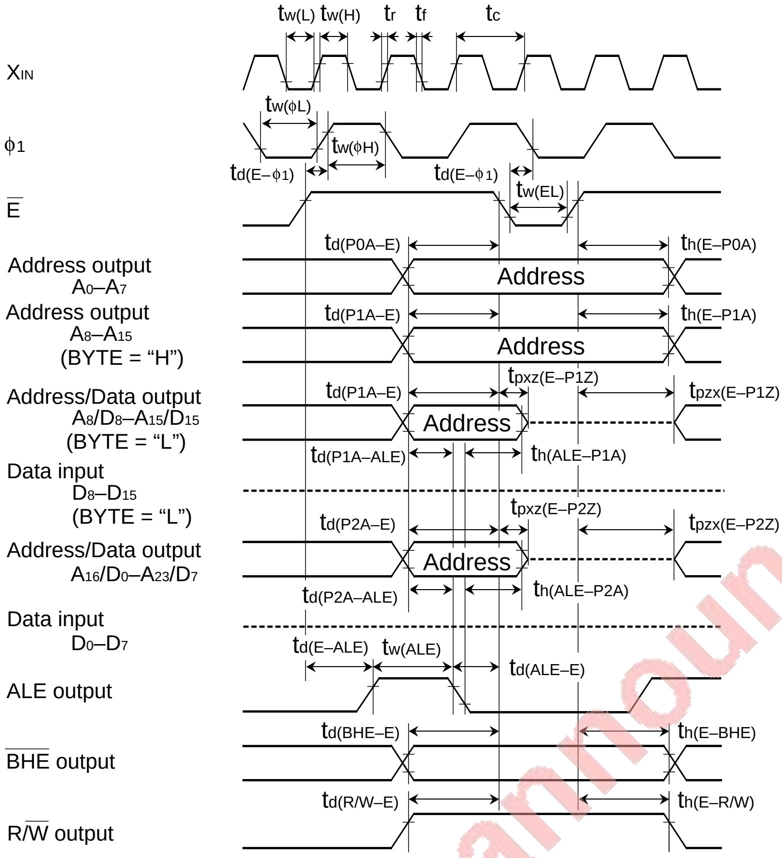
•Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V

•Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

15.14 Memory expansion mode and microprocessor mode : When 2-φ access in high-speed running (Internal RAM access)

Memory expansion mode and Microprocessor mode
: When 2-φ access in high-speed running (Internal RAM access)

<Read>



✻ The contents of external data bus cannot be read into the internal.

Test conditions (φ1, E, P0–P3)

- V_{CC} = 5 V±10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

15.15 Testing circuit for ports P0 to P8, ϕ_1, and $\overline{E}$

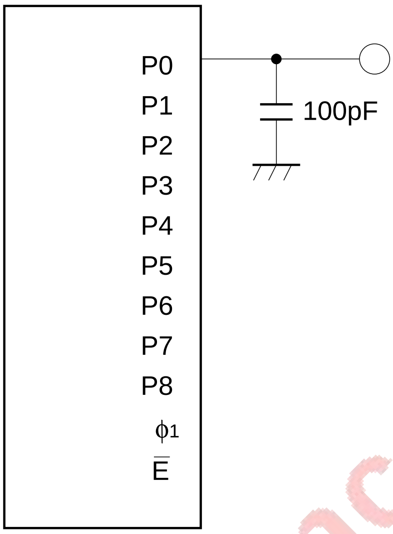


Fig. 15.15.1 Testing circuit for ports P0 to P8, ϕ_1, and $\overline{E}$

MEMORANDUM

CHAPTER 16

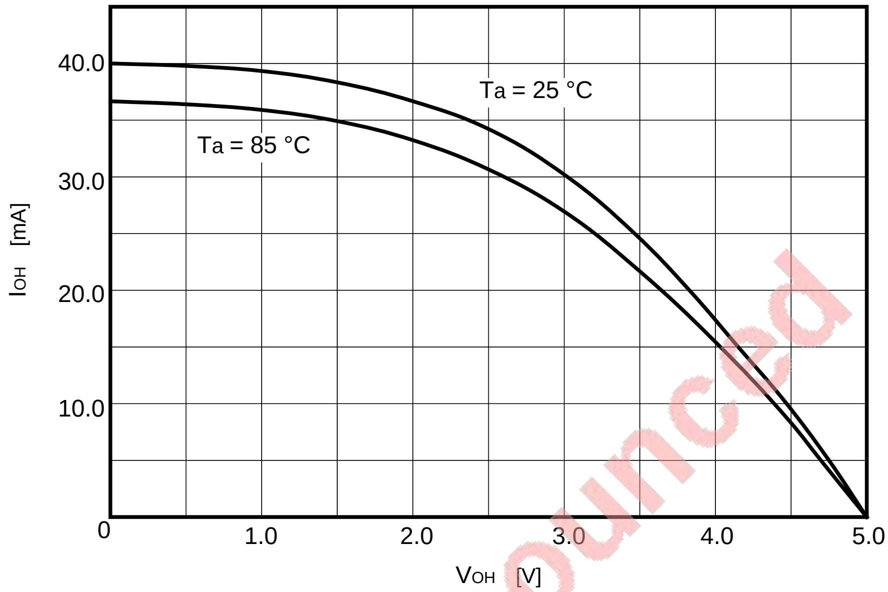
STANDARD CHARACTERISTICS

16.1 Standard characteristics

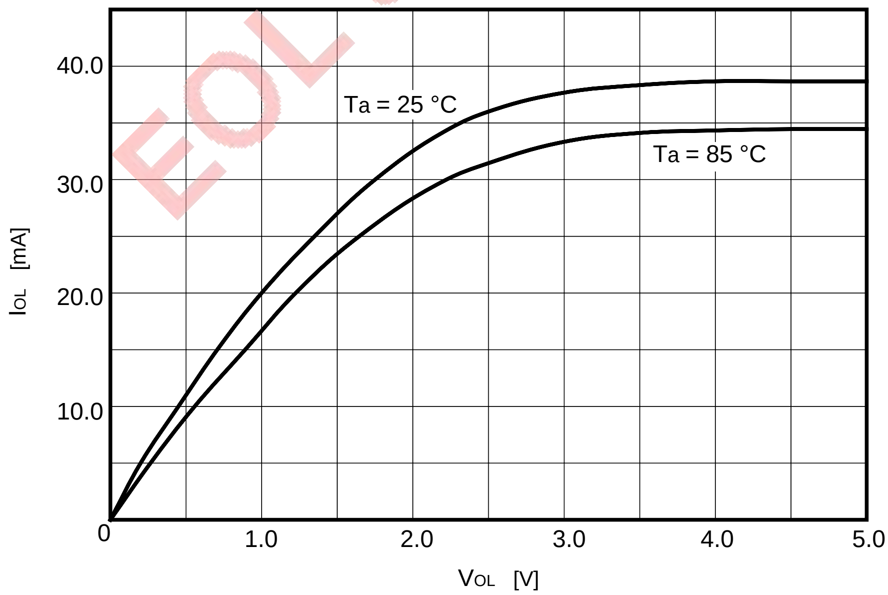
Standard characteristics described below are just examples of the M37751M6C-XXXFP's characteristics and are not guaranteed. For rated values, refer to **"Chapter 15. ELECTRICAL CHARACTERISTICS."**

16.1.1 Programmable I/O port (CMOS output) standard characteristics

(1) P-channel I_{OH}–V_{OH} characteristics



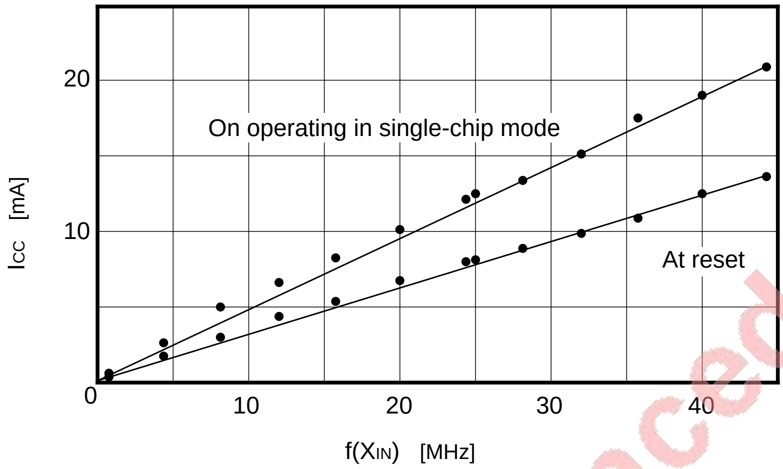
(2) N-channel I_{OL}–V_{OL} characteristics



16.1.2 Icc–f(XIN) standard characteristics

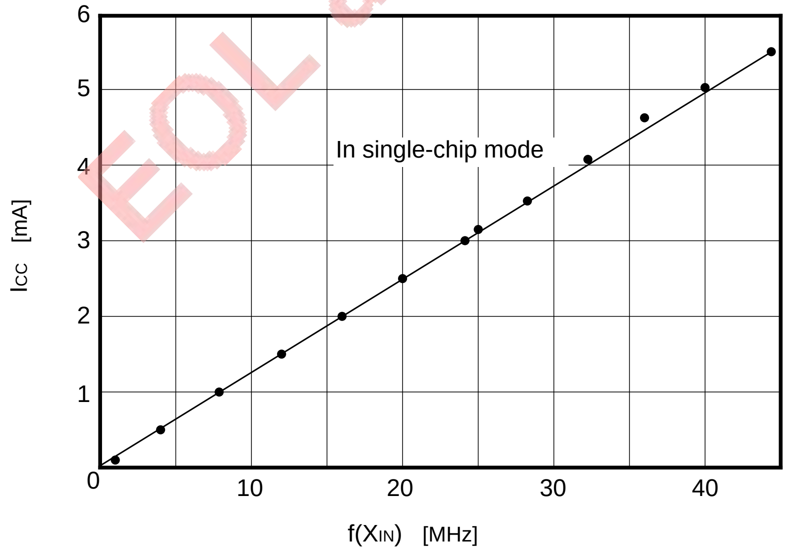
(1) Icc–f(XIN) standard characteristics on operating and at reset

Measuring conditions (Vcc = 5.0 V, Ta = 25 °C, f(XIN) ; square waveform)



(2) Icc–f(XIN) standard characteristics during wait mode

Measuring conditions (Vcc = 5.0 V, Ta = 25 °C, f(XIN) ; square waveform)

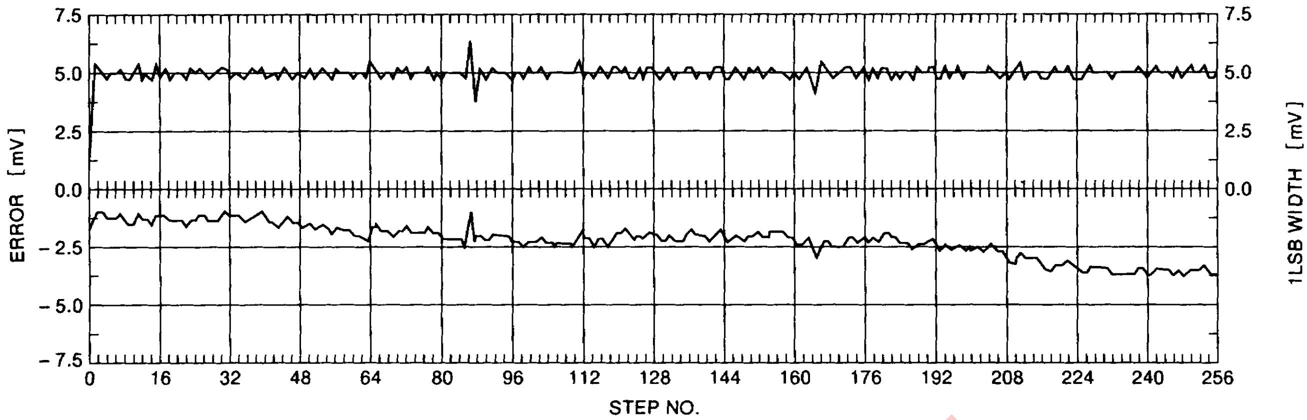


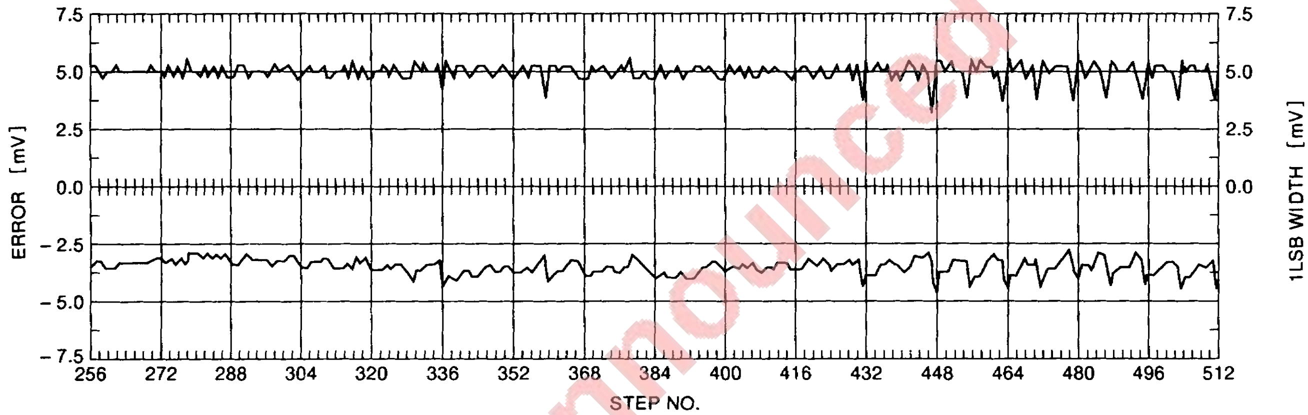
16.1.3 A-D converter standard characteristics

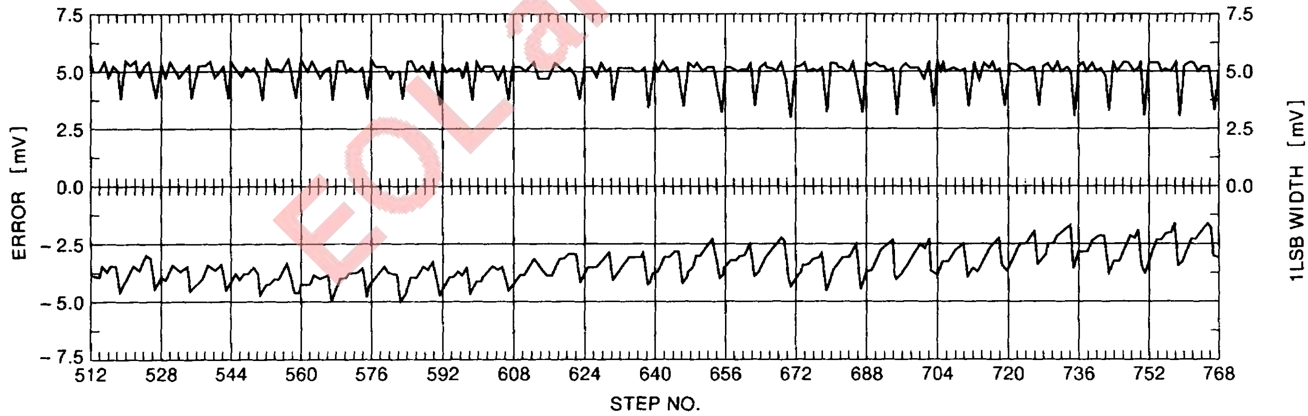
The lower line of the graph indicates the absolute precision errors. These are expressed as the deviation from the ideal value when the output code changes. For example, the change in output code from 15 to 16 should occurs at 77.5 mV, but the measured value is –1.2 mV. Accordingly, the measured point of change is 77.5 – 1.2 = 76.3 mV.

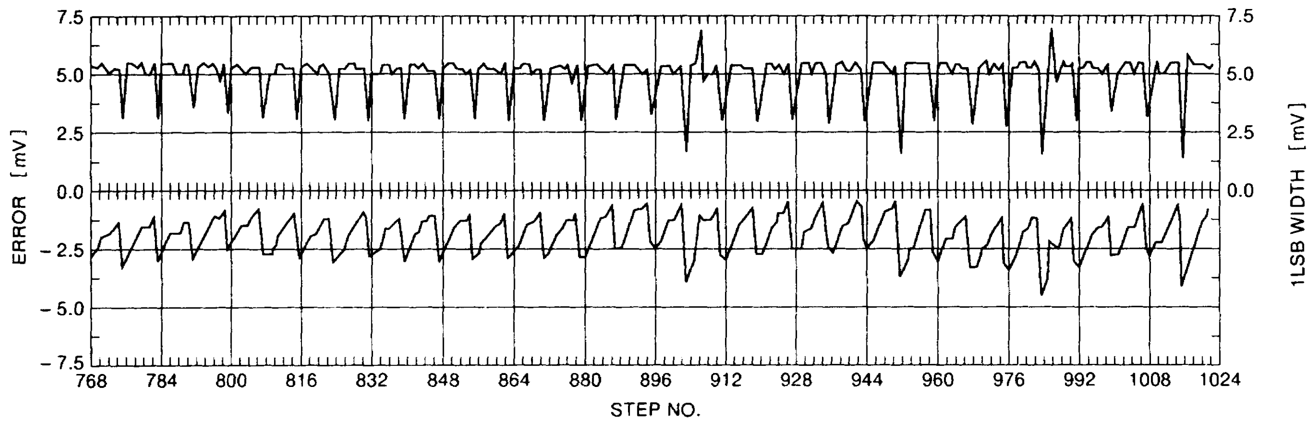
The upper line of the graph indicates the input voltage width for which the output code is constant. For example, the measured input voltage width for which the output code is 16 is 4.9 mV, so that the differential non-linear error is 4.9 – 5 = –0.1 mV (–0.02 LSB).

[Measuring conditions]
•Vcc = 5.12 V, •VREF = 5.12 V, •f(XIN) = 40 MHz, •Ta = 25 °C









MEMORANDUM

CHAPTER 17

APPLICATIONS

17.1 Memory expansion

This section shows examples for memory and I/O expansion. Refer to **"Chapter 12. CONNECTION WITH EXTERNAL DEVICES"** for details about the functions and operation of used pins when expanding a memory or I/O. Refer to **"Chapter 15. ELECTRICAL CHARACTERISTICS"** for timing requirements of the microcomputer. Application shown here are just examples. The user shall modify them according to the actual application and test them.

17.1.1 Memory expansion model

Memory expansion to the external is possible in the memory expansion mode or the microprocessor mode. The level of the external data bus width select signal makes it possible to select the four memory expansion models shown in Table 17.1.1.

(1) Minimum model

This is an expansion model of which external data bus width is 8 bits and accessible area is expanded up to 64 Kbytes. It is unnecessary to connect the address latch externally. This is an expansion model which is suited to having priority the cost when connecting the memory of which external data bus width is 8 bits.

(2) Medium model A

This is an expansion model of which external data bus width is 8 bits and accessible area is expanded up to 16 Mbytes. In this expansion model, the high-order 8 bits of the external address bus (A_{23} to A_{16}) are multiplexed with the external data bus. Therefore, an n-bit ($n \leq 8$) address latch is required for latching address (n bits of A_{23} to A_{16}).

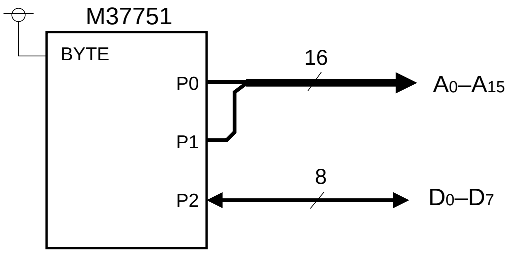
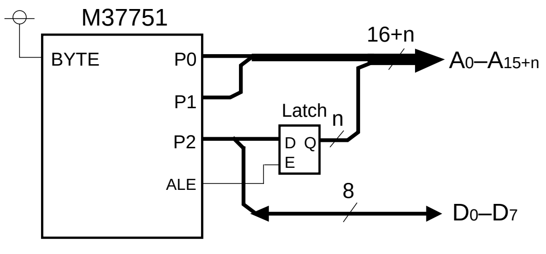
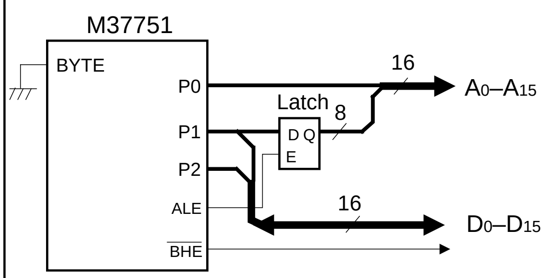
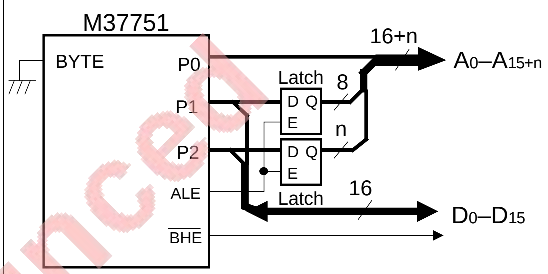
(3) Medium model B

This is an expansion model of which external data bus width is 16 bits and accessible area is expanded up to 64 Kbytes. This expansion model is used when having priority the rate performance. In this expansion model, the middle-order 8 bits of the external address bus (A_{15} to A_8) are multiplexed with the external data bus. Therefore, an 8-bit address latch is required for latching address (A_{15} to A_8).

(4) Maximum model

This is an expansion model of which external data bus width is 16 bits and accessible area is expanded up to 16 Mbytes. In this expansion model, the high- and middle-order 16 bits of the external address bus (A_{23} to A_8) are multiplexed with the external data bus. Therefore, an 8-bit address latch for latching A_{15} to A_8 and an n-bit ($n \leq 8$) address latch for latching n bits of A_{23} to A_{16} are required.

Table 17.1.1 Memory expansion model

Access area / External data bus width	Maximum 64 Kbytes	Maximum 16 Mbytes
8-bit width; BYTE = "H"	 Memory expansion model: Minimum model	 Memory expansion model: Medium model A
16-bit width; BYTE = "L"	 Memory expansion model: Medium model B	 Memory expansion model: Maximum model

Notes 1: Refer to **"Chapter 12. CONNECTION WITH EXTERNAL DEVICES"** for details about the functions and operation of used pins when expanding a memory. Refer to **"Chapter 15. ELECTRICAL CHARACTERISTICS"** for timing requirements.

2: Because the address bus width is used as maximum 24 bits when expanding a memory, strengthen the M37751's Vss line. (Refer to **"Appendix 8. Examples of noise immunity improvement."**)

17.1.2 How to calculate timing

When expanding a memory, use a memory of which standard specifications satisfy the address access time and the data setup time for write. The following describes how to calculate each timing.

① **External memory's address access time; $t_{a(AD)}$**

$t_{a(AD)} = t_{su(P0A/P1A/P2A-P1D/P2D)}$ – (address decode time*1 + address latch delay time*2)

Address decode time*1: Time required for the chip select signal to be enabled after decoding address
Address latch delay time*2: Delay time required when latching address (Unnecessary in minimum model)

② **External memory's data setup time for write; $t_{su(D)}$**

$t_{su(D)} = t_{w(EL)} - t_{d(E-P2Q/P1Q)}$

$t_{d(E-P2Q/P1Q)}$: $t_{d(E-P2Q)}$ or $t_{d(E-P1Q)}$

Table 17.1.2 lists the data or the calculation formulas for each parameter. Figure 17.1.1 shows the bus timing diagram. Figures 17.1.2 and 17.1.4 show the relationship between $t_{su(P0A/P1A/P2A-P1D/P2D)}$ and $f(X_{IN})$; Figures 17.1.3 and 17.1.5 show the relationship between $t_{su(D)}$ and $f(X_{IN})$.

Table 17.1.2 Data or calculation formulas for each parameter (unit: ns)

Parameter \ Bus cycle	Low-speed running 2ϕ access	Low-speed running 3ϕ access	Low-speed running 4ϕ access	High-speed running 3ϕ access	High-speed running 4ϕ access	High-speed running 5ϕ access
$t_{su(P0A/P1A/P2A-P1D/P2D)}$	$\frac{3 \times 10^9}{f(X_{IN})} - 65$	$\frac{5 \times 10^9}{f(X_{IN})} - 65$	$\frac{7 \times 10^9}{f(X_{IN})} - 65$	$\frac{5 \times 10^9}{f(X_{IN})} - 75$	$\frac{7 \times 10^9}{f(X_{IN})} - 75$	$\frac{9 \times 10^9}{f(X_{IN})} - 75$
$t_{w(EL)}$	$\frac{2 \times 10^9}{f(X_{IN})} - 25$	$\frac{4 \times 10^9}{f(X_{IN})} - 25$	$\frac{4 \times 10^9}{f(X_{IN})} - 25$	$\frac{3 \times 10^9}{f(X_{IN})} - 25$	$\frac{4 \times 10^9}{f(X_{IN})} - 25$	$\frac{6 \times 10^9}{f(X_{IN})} - 25$
$t_{d(E-P2Q)}$ $t_{d(E-P1Q)}$	35	35	35	35	35	35

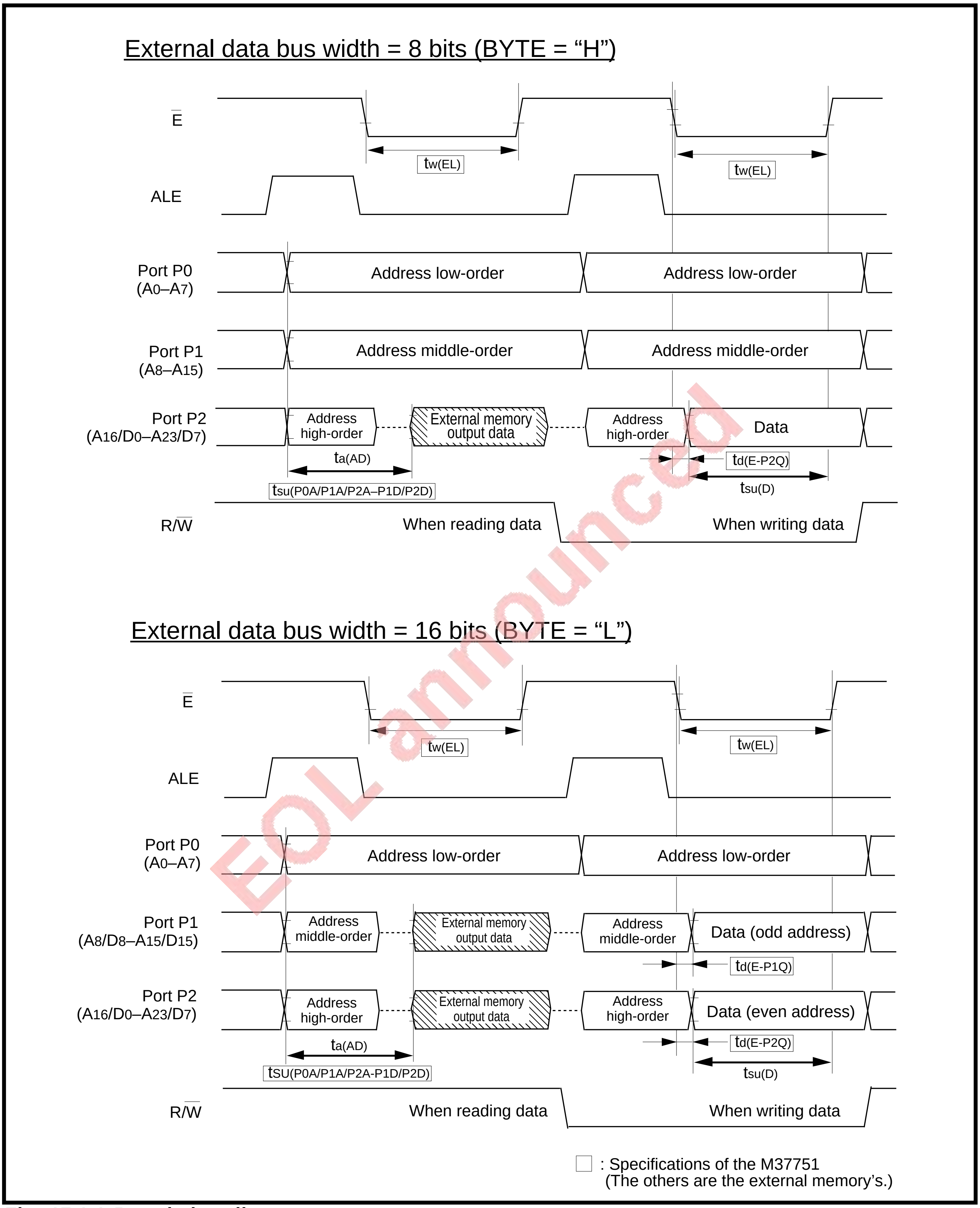


Fig. 17.1.1 Bus timing diagrams

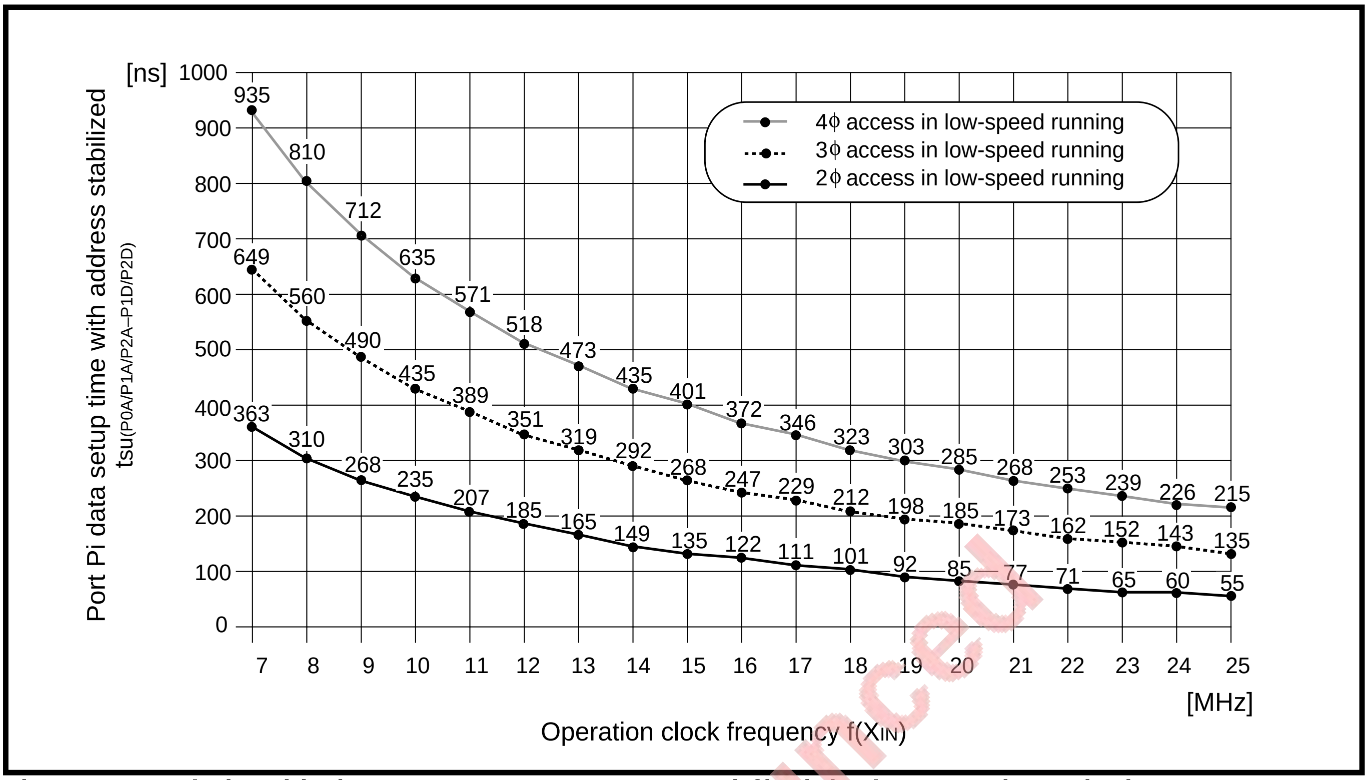


Fig. 17.1.2 Relationship between $t_{su(P0A/P1A/P2A-P1D/P2D)}$ and $f(X_{IN})$ (at low-speed running)

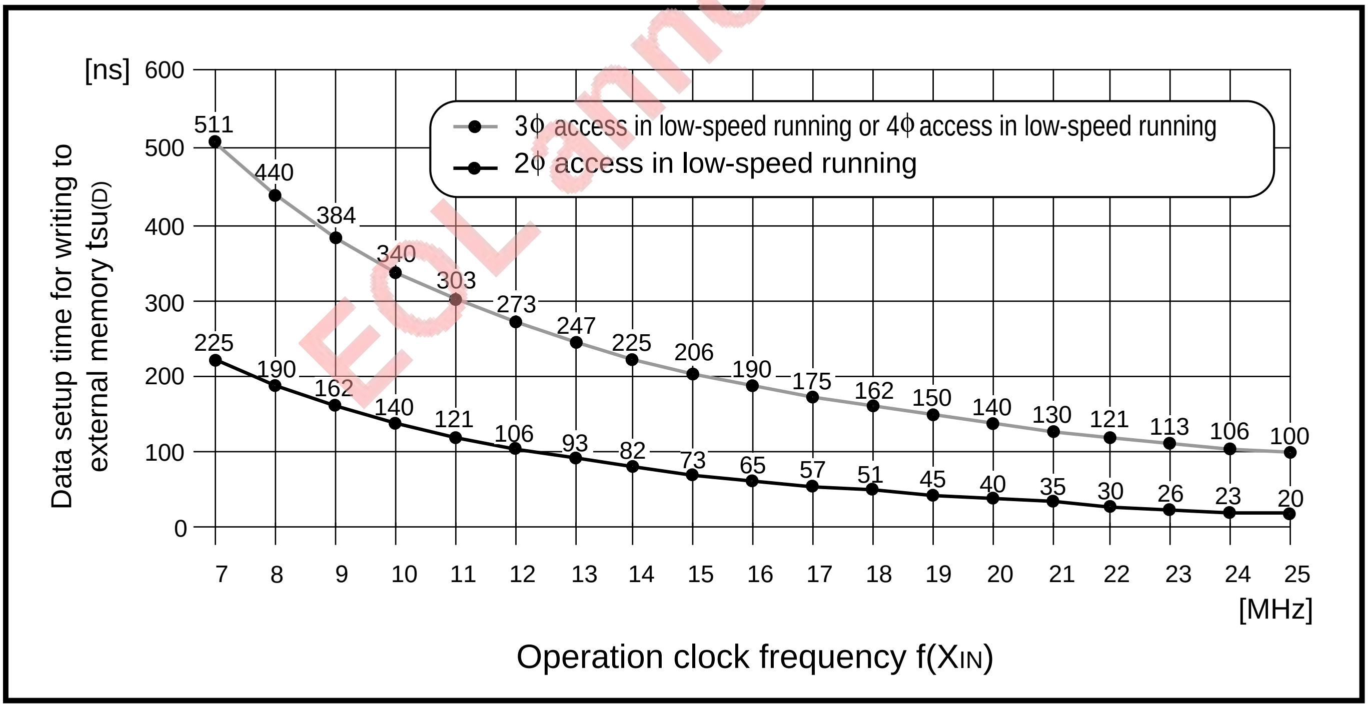


Fig. 17.1.3 Relationship between $t_{su(D)}$ and $f(X_{IN})$ (at low-speed running)

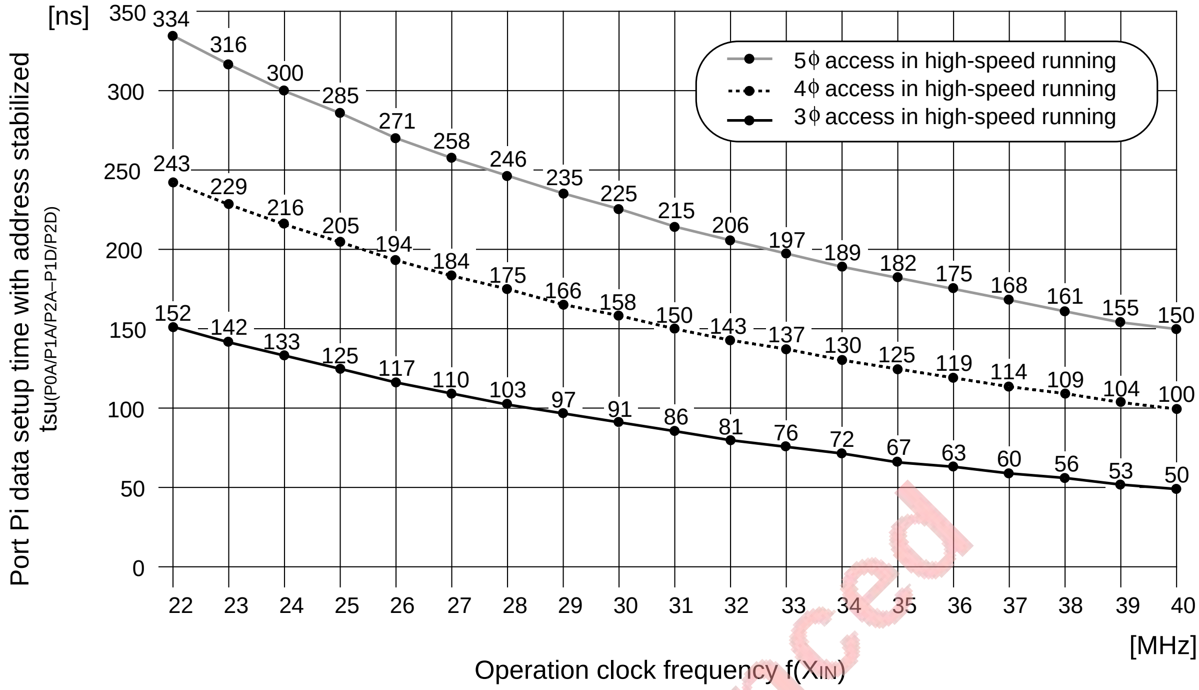


Fig. 17.1.4 Relationship between $t_{su(P0A/P1A/P2A–P1D/P2D)}$ and $f(X_{IN})$ (at high-speed running)

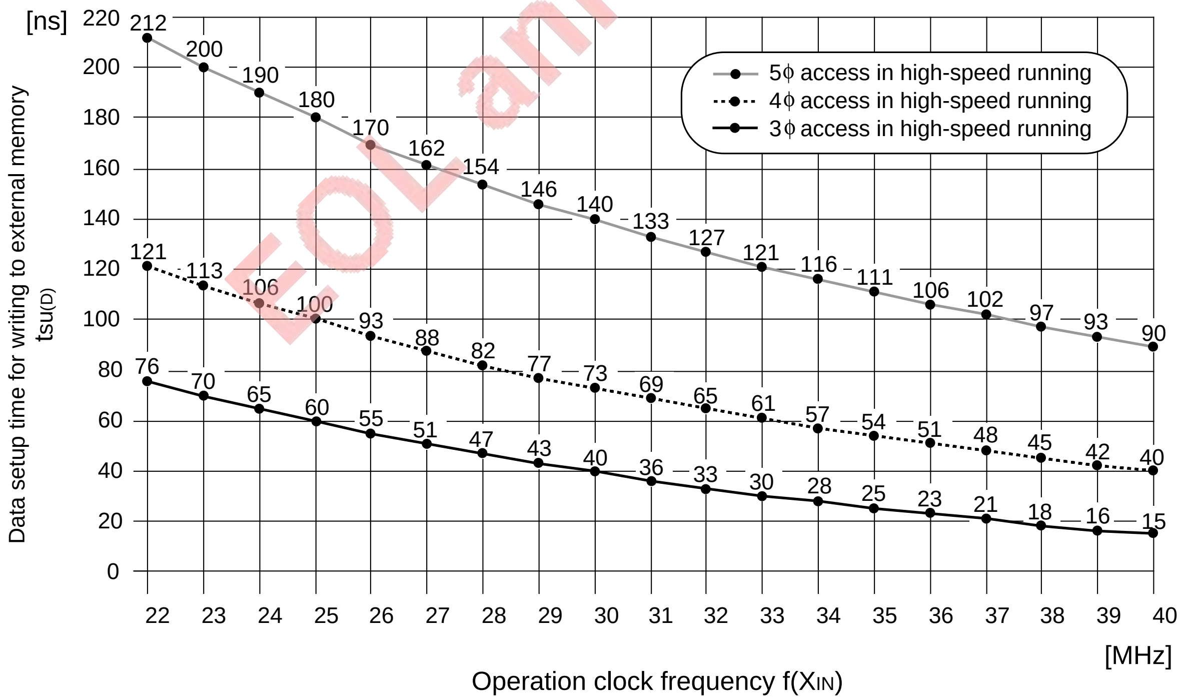


Fig. 17.1.5 Relationship between $t_{su(D)}$ and $f(X_{IN})$ (at high-speed running)

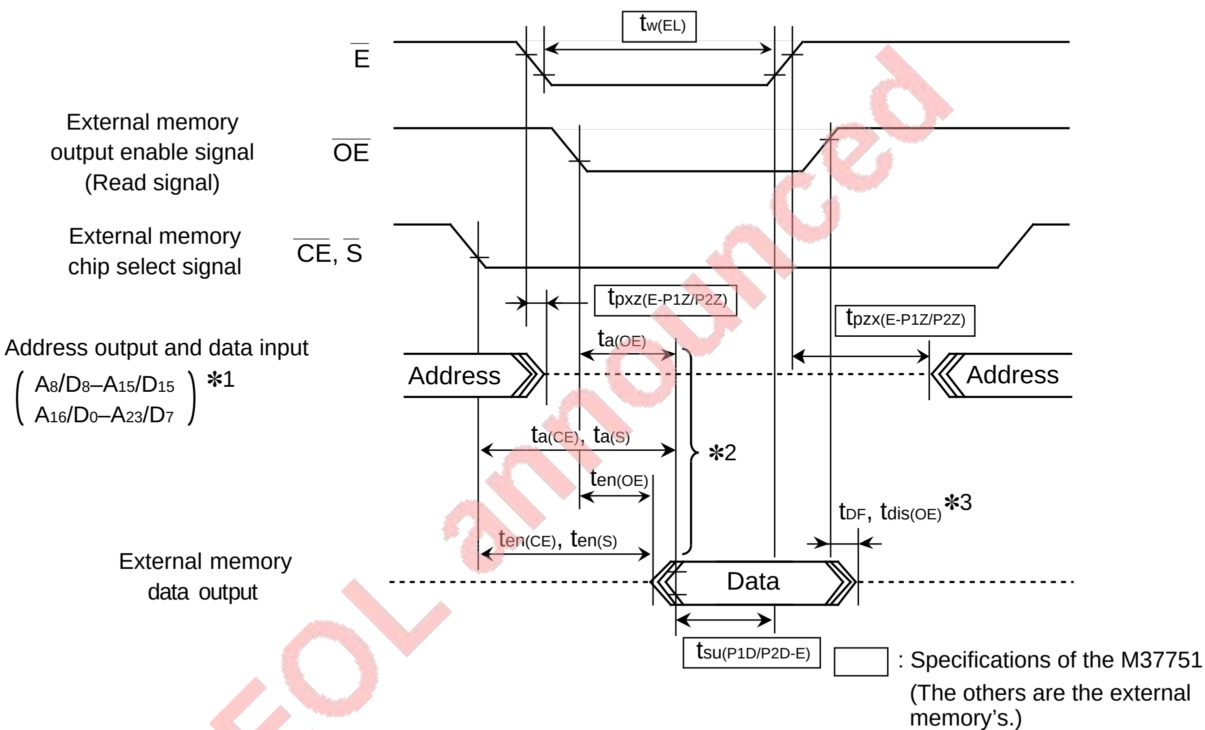
17.1.3 Points in memory expansion

(1) Reading data

Figure 17.1.6 shows the timing at which data is read from an external memory.

When reading data, the external data bus is placed in a floating state, and data is read from the external memory. This floating state is maintained from $t_{pxz(E-P1Z/P2Z)}$ after falling of the $\overline{E}$ signal till $t_{pzx(E-P1Z/P2Z)}$ after rising of the $\overline{E}$ signal. Table 17.1.3 lists the values of $t_{pxz(E-P1Z/P2Z)}$ and the formulas to calculate $t_{pzx(E-P1Z/P2Z)}$.

Consider timing during data read to avoid collision between the data being read–in and the preceding or following address output because the external data bus is multiplexed with the external address bus. (Refer to **"(3) Precautions on memory expansion."**)



*1: This applies when the external data bus has a width of 16 bits (BYTE = "L").

*2: If one of the external memory's specifications is smaller than $t_{pxz(E-P1Z/P2Z)}$, there is a possibility of the tail of address colliding with the head of data. → Refer to **"(3) Precautions on memory expansion."**

*3: If one of the external memory's specifications is greater than $t_{pzx(E-P1Z/P2Z)}$, there is a possibility of the tail of data colliding with the head of address. → Refer to **"(3) Precautions on memory expansion."**

Fig. 17.1.6 Timing at which data is read from external memory

Table 17.1.3 Values of $t_{pxz(E-P1Z/P2Z)}$ and formulas to calculate $t_{pzx(E-P1Z/P2Z)}$ (unit : ns)

Parameter \ Bus cycle	Low-speed running 2φ access	Low-speed running 3φ access	Low-speed running 4φ access	High-speed running 3φ access	High-speed running 4φ access	High-speed running 5φ access
$t_{pxz(E-P1Z)}$ $t_{pxz(E-P2Z)}$	5	5	5	5	5	5
$t_{pzx(E-P1Z)}$ $t_{pzx(E-P2Z)}$	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	$\frac{1 \times 10^9}{f(X_{IN})} - 10$

17.1 Memory expansion

(2) Writing data

Figure 17.1.7 shows the timing at which data is written to an external memory.
When writing data, the output data is validated after $t_{d(E\text{-}P1Q/P2Q)}$ passes from falling of the $\overline{E}$ signal. Its validated data is output continuously until $t_{h(E\text{-}P1Q/P2Q)}$ passes from rising of the $\overline{E}$ signal. Table 17.1.4 lists the data of $t_{d(E\text{-}P1Q/P2Q)}$ and the calculation formulas of $t_{h(E\text{-}P1Q/P2Q)}$.
Data output at writing data must satisfy the data set up time, $t_{su(D)}$, and the data hold time, $t_{h(D)}$, for write to an external memory.

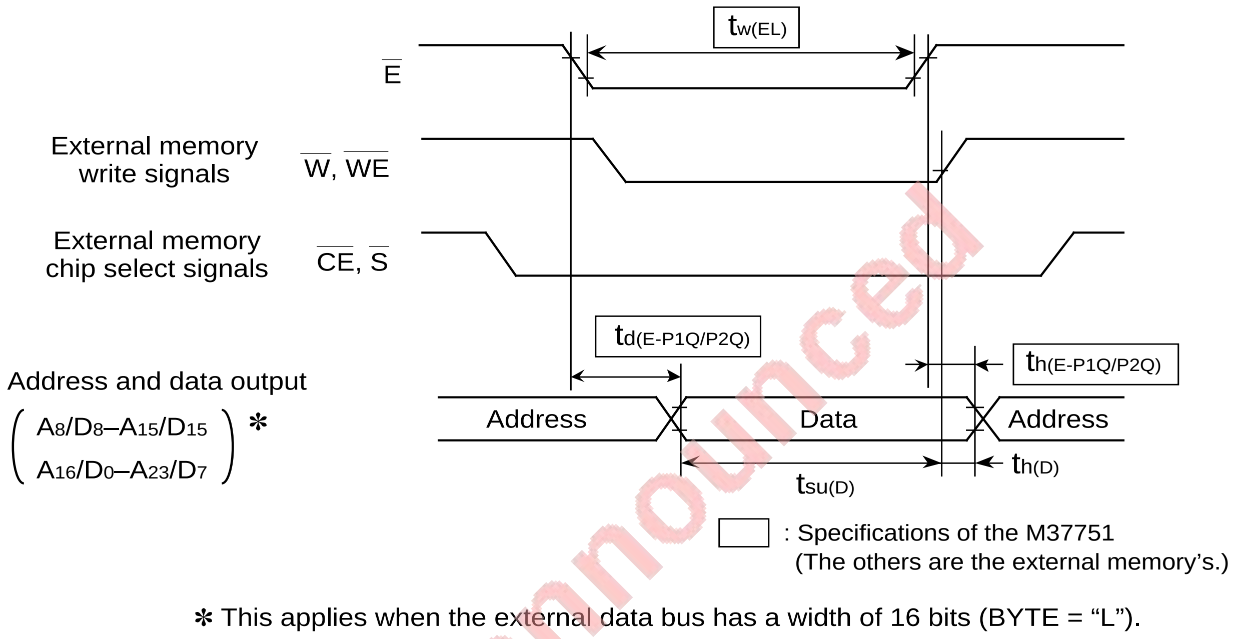


Fig. 17.1.7 Timing at which data is written to external memory

Table 17.1.4 Data of $t_{d(E\text{-}P1Q/P2Q)}$ and calculation formulas of $t_{h(E\text{-}P1Q/P2Q)}$ (unit: ns)

Parameter \ Bus cycle	Low-speed running 2ϕ access	Low-speed running 3ϕ access	Low-speed running 4ϕ access	High-speed running 3ϕ access	High-speed running 4ϕ access	High-speed running 5ϕ access
$t_{d(E—P1Q)}$ $t_{d(E—P2Q)}$	35	35	35	35	35	35
$t_{h(E—P1Q)}$ $t_{hE—P2Q)}$	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	$\frac{1 \times 10^9}{f(X_{IN})} - 22$	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	$\frac{1 \times 10^9}{f(X_{IN})} - 10$	$\frac{1 \times 10^9}{f(X_{IN})} - 10$

(3) Precautions on memory expansion

As described in ① to ③ below, if specifications of the external memory do not match those of the M37751, some considerations must be incorporated into circuit design as in the following cases:

① When using an external memory that requires a long access time, $t_{a(AD)}$
② When using an external memory that outputs data within $t_{pxz(E-P1Z/P2Z)}$ after falling of the $\overline{E}$ signal
③ When using an external memory that outputs data for more than $t_{pzx(E-P1Z/P2Z)}$ after rising of the $\overline{E}$ signal

① When using an external memory that requires a long access time, $t_{a(AD)}$

If the M37751's $t_{su(P1D/P2D-E)}$ cannot be satisfied because the external memory requires a long access time, $t_{a(AD)}$, examine the method described below:

- Lower $f(X_{IN})$.
- Select a long bus cycle by software. (Refer to section **"12.2 Bus cycle."**)
- Use Ready function. (Refer to section **"12.3 Ready function."**)

Figure 17.1.8 shows an example of using Ready function (at 2ϕ access in low–speed running).
Figure 17.1.9 shows an example of using Ready function (at 3ϕ access in low–speed running).
Figure 17.1.10 shows an example of using Ready function (at 3ϕ access in high–speed running).
Figure 17.1.11 shows an example of using Ready function (at 4ϕ access in high–speed running).
Ready function is available for the internal areas, so that the circuit in Figures 17.1.8 to 17.1.11 use the chip select signal ($\overline{CS_2}$) to specify the area where Ready function is available.

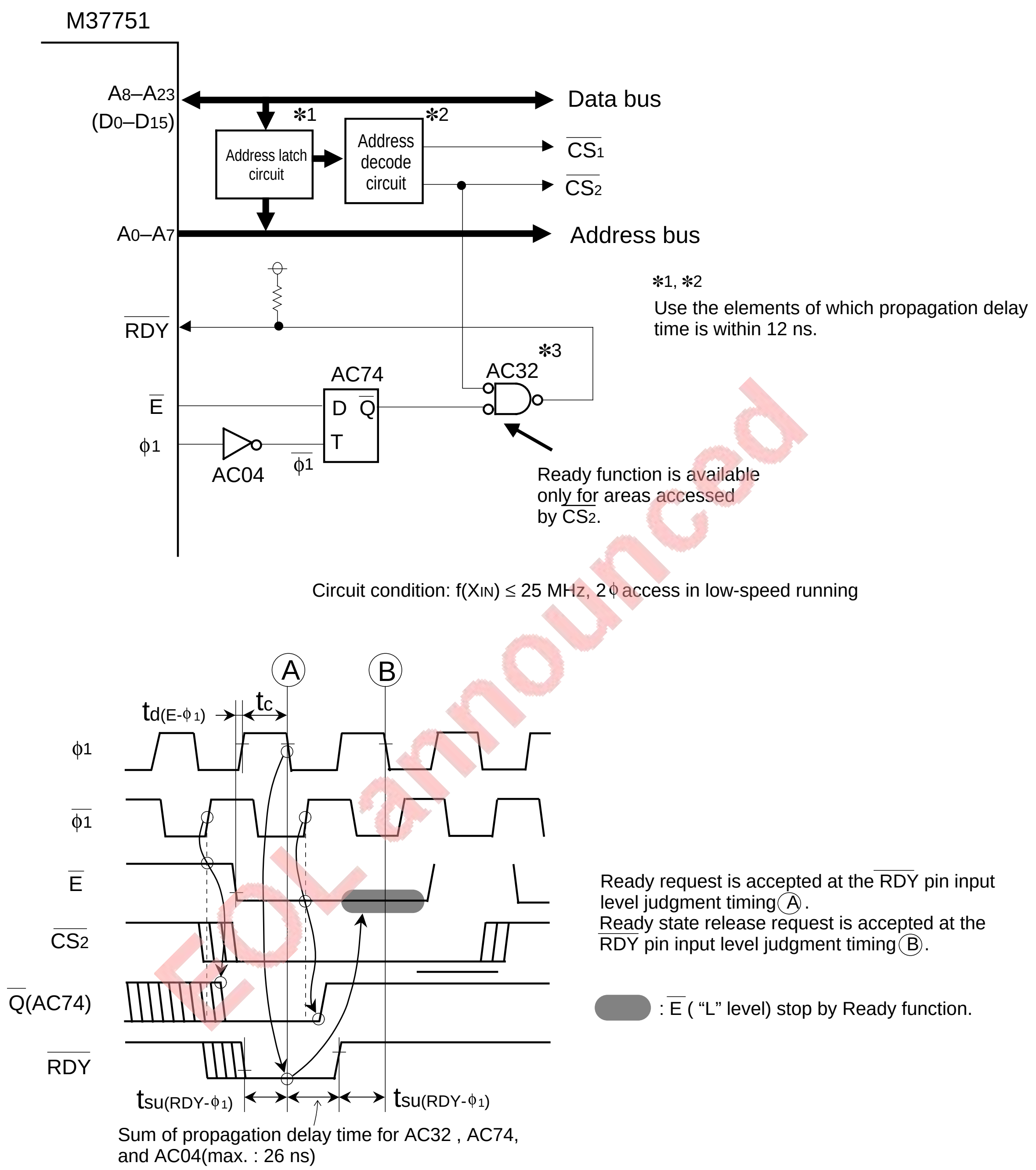
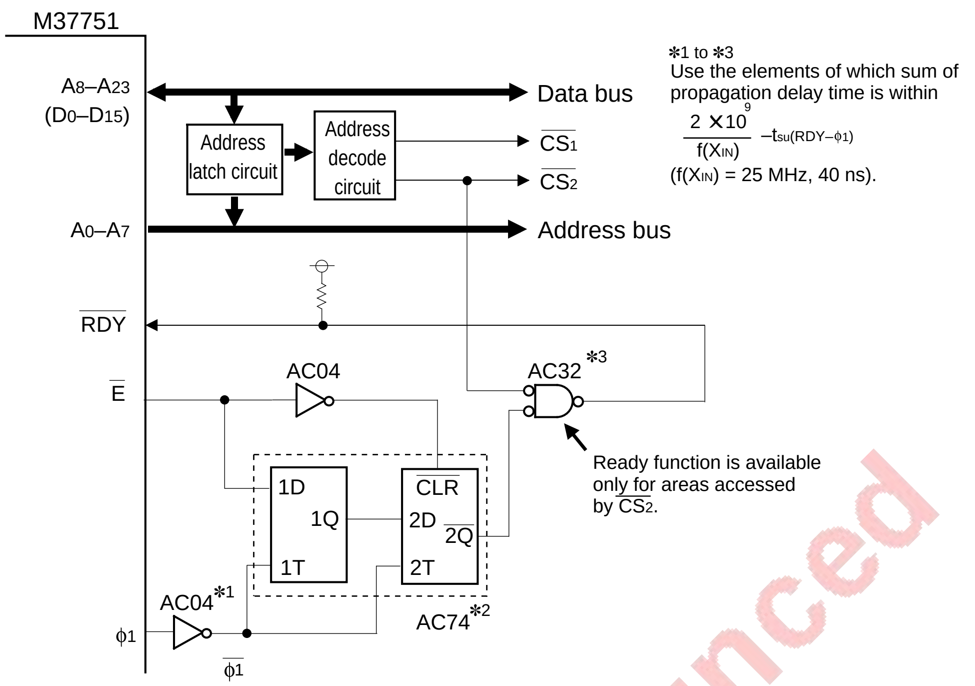
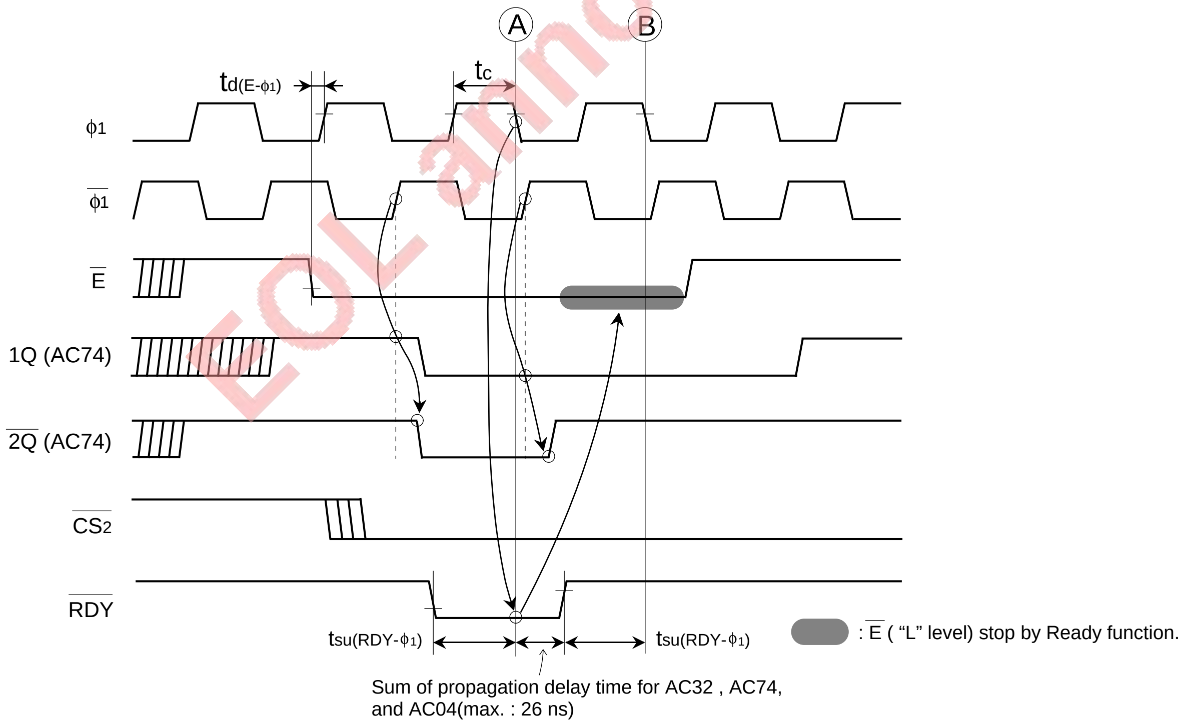


Fig. 17.1.8 Example of using Ready function (at 2ϕ access in low–speed running)



Circuit condition: $f(X_{IN}) \leq 25$ MHz, 3φ access in low–speed running



Ready request is accepted at the RDY pin input level judgment timing Ⓐ.
Ready state release request is accepted at the RDY pin input level judgment timing Ⓑ.

Fig. 17.1.9 Example of using Ready function (at 3φ access in low–speed running)

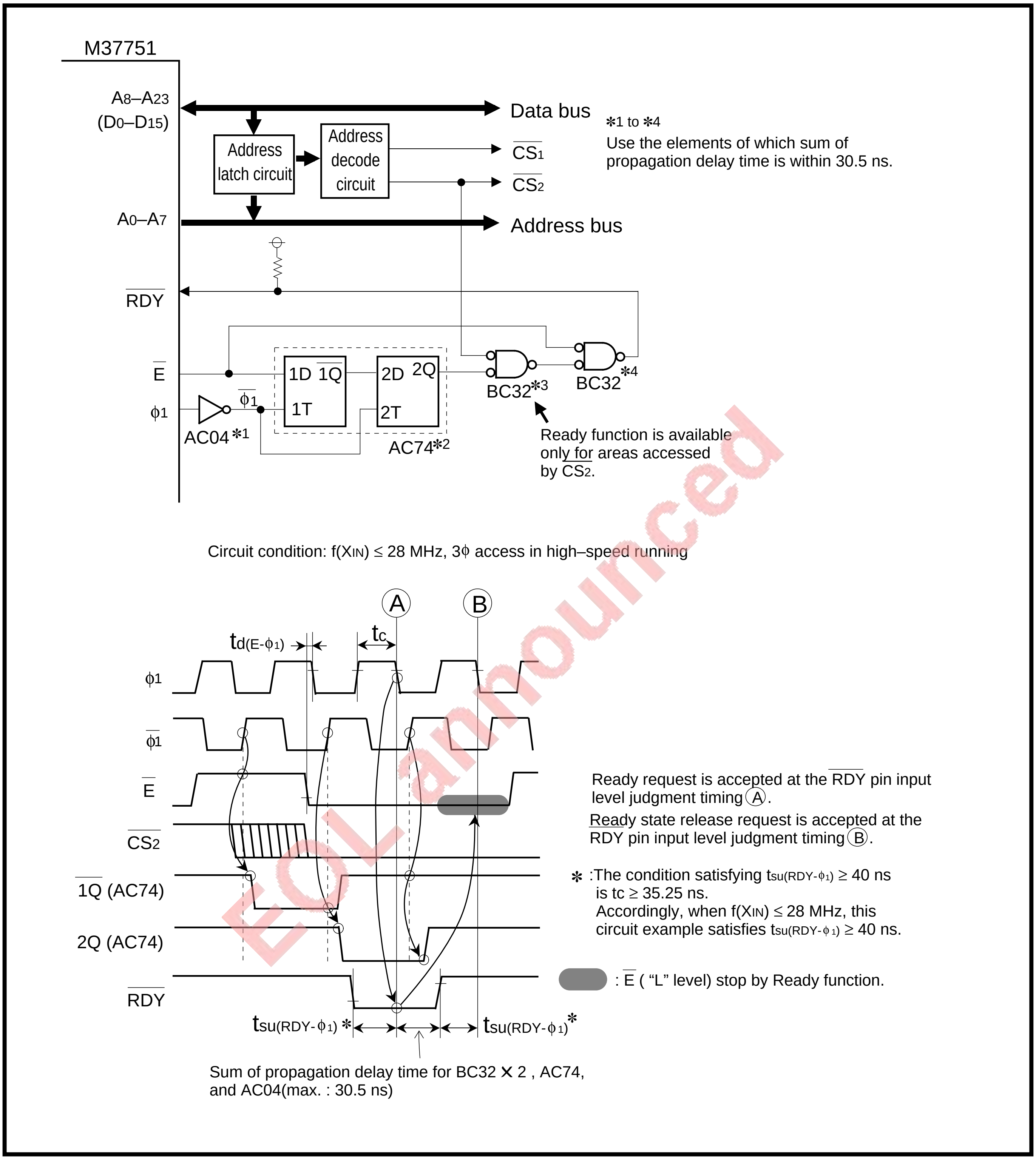


Fig. 17.1.10 Example of using Ready function (at 3ϕ access in high–speed running)

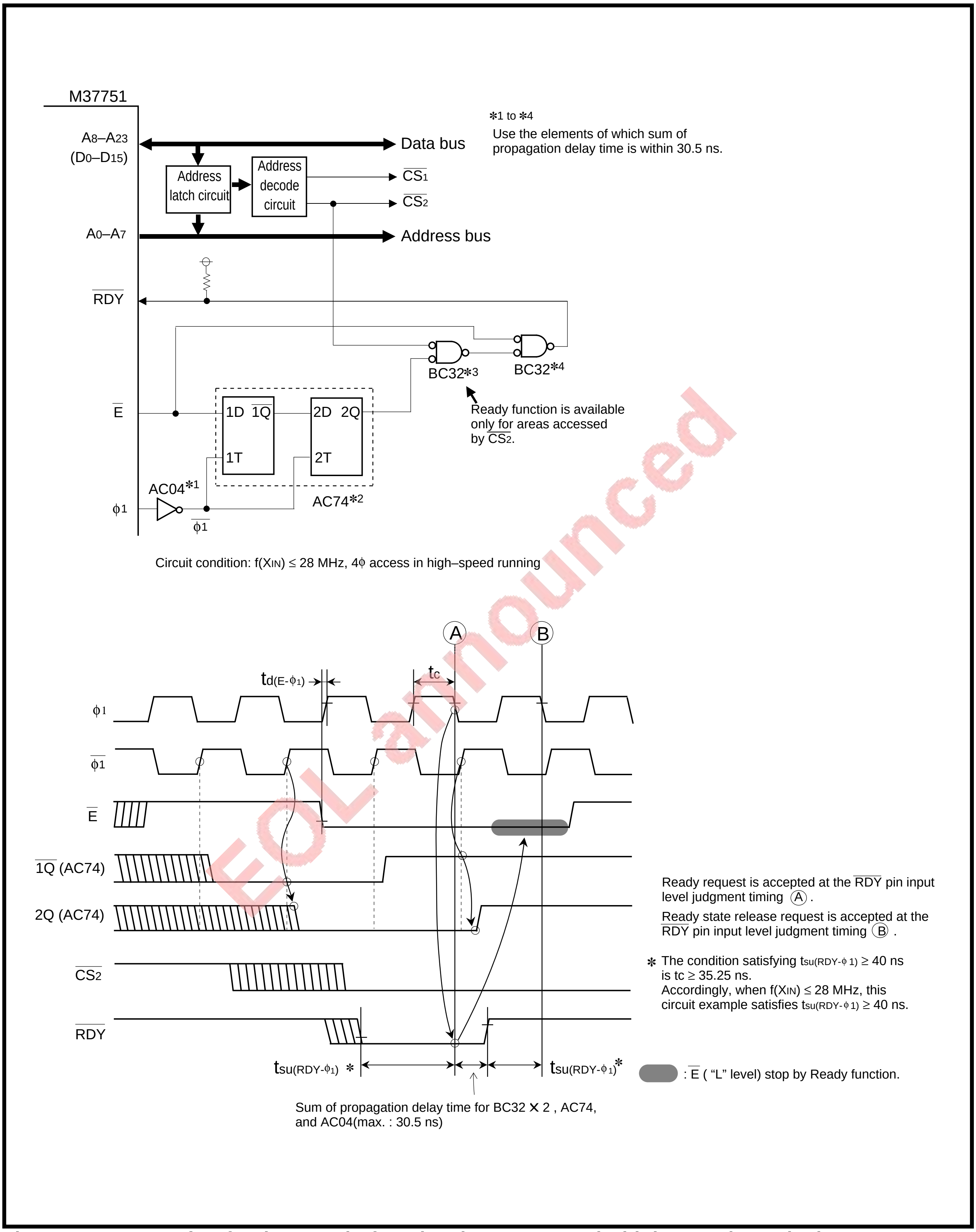
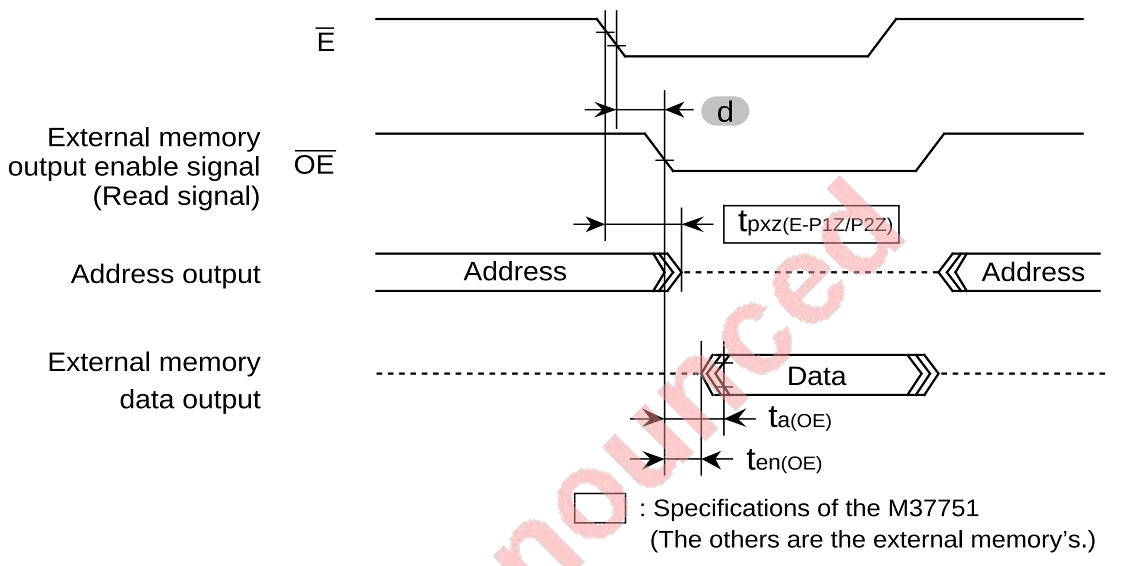


Fig. 17.1.11 Example of using Ready function (at 4φ access in high–speed running)

② **When using an external memory that outputs data within $t_{pxz(E-P1Z/P2Z)}$ after falling of the $\overline{E}$ signal**

Because the external memory outputs data within $t_{pxz(E-P1Z/P2Z)}$ after falling of the $\overline{E}$ signal, there will be a possibility of the tail of address colliding with the head of data. In such a case, generate the memory read signal ($\overline{OE}$) with delay only the leading edge of the fall of the $\overline{E}$. (Refer to Figure 17.1.12.)



Note: Satisfy $t_{pxz(E-P1Z/P2Z)} \le t_{en(OE)}+d$.

If $t_{en(OE)} \le t_{pxz(E-P1Z/P2Z)}$ (= 5 ns), secure a certain time (i.e., 'd' in this diagram) from falling of $\overline{E}$ to the falling of $\overline{OE}$.

Fig. 17.1.12 Example of causing to delay data output timing

③ **When using external memory that outputs data for more than $t_{pzx(E-P1Z/P2Z)}$ after rising of $\overline{E}$ signal**

Because the external memory outputs data for more than $t_{pzx(E-P1Z/P2Z)}$ after rising of the $\overline{E}$ signal, there will be a possibility of the tail of data colliding with the head of address. In such a case, examine the method described below:

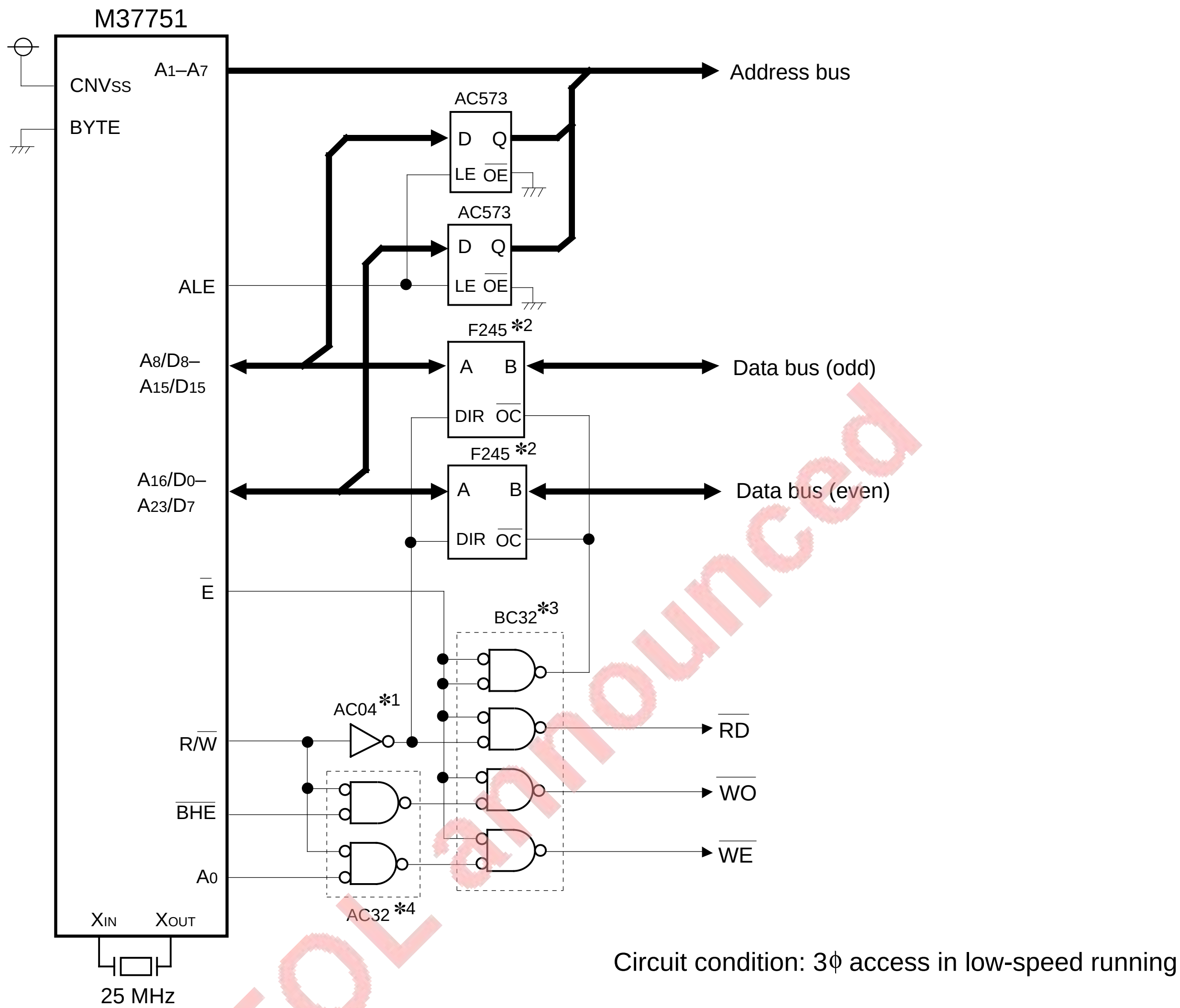
- Cut the tail of data output from the external memory by using, for example, a bus buffer.
- Use the Mitsubishi's memory chips that can be connected without a bus buffer.

Figures 17.1.13 to 17.1.20 show examples for how to use a bus buffer and the timing charts. Table 17.1.5 lists the memory chips that can be connected a without bus buffer. When using one of these memory chips, the user can connect it to the user's microcomputer without a bus buffer because timing parameters t_{DF} and $t_{dis(OE)}$ listed below are guaranteed. (However, the read signal must go high within 10 ns after rising of $\overline{E}$ signal.)

Table 17.1.5 Memory chips that can be connected without bus buffer

Memory	Type description	$t_{DF}/t_{dis(OE)}$	Conditions
EPROM	M5M27C256AK-85, -10, -12, -15 M5M27C512AK-10, -12, -15 M5M27C100K-12. -15 M5M27C101K-12, -15 M5M27C102K-12, -15 M5M27C201K, JK-10, -12, -15 M5M27C202K, JK-10, -12, -15	(Maximum) 15 ns (when guaranteeing by kit) **(Note)**	$f(X_{IN}) \leq 20$ MHz, at low–speed running
One-time PROM	M5M27C256AP, FP, VP, RV-12, -15 M5M27C512AP, FP-15 M5M27C100P-15 M5M27C101P, FP, J, VP, RV-15 M5M27C102P, FP, J, VP, RV-15 M5M27C201P, FP, J, VP, RV-12, -15 M5M27C202P, FP, J, VP, RV-12, -15		
Flash memory	M5M28F101P, FP, J, VP, RV-10, -12, -15 M5M28F102FP, J, VP, RV-10, -12, -15		
SRAM	M5M5256CP, FP, KP, VP, RV-55LL, -55XL, -70LL, -70XL, -85LL, -85XL, -10LL, -10XL		
	M5M5278CP, FP, J-20, -20L	8 ns	$f(X_{IN}) \leq 40$ MHz, at high–speed running $f(X_{IN}) \leq 25$ MHz, at low–speed running
	M5M5278CP, FP, J-25, -25L	10 ns	$f(X_{IN}) \leq 25$ MHz, at low–speed running
	M5M5278DP, J-12	6 ns	$f(X_{IN}) \leq 40$ MHz, at high–speed running $f(X_{IN}) \leq 25$ MHz, at low–speed running
	M5M5278DP, FP, J-15, -15L	7 ns	
	M5M5278DP, FP, J-20, -20L	8 ns	

Note: When the user want specifications of the memory chips listed above, add a comment "$t_{DF}/t_{dis(OE)}$ 15 ns product, microcomputer and kit."



✻1: Use the elements of which propagation delay time is within 20 ns.
✻2, ✻3 : Use the elements of which sum of output disable time in ✻2 and propagation delay time in ✻3 is within 18 ns and the sum of output enable time in ✻2 and propagation delay time in ✻3 is 5 ns or more.
✻4: Use the elements of which propagation delay time is within 12 ns.

Fig. 17.1.13 Example for using bus buffer (at low–speed running–1)

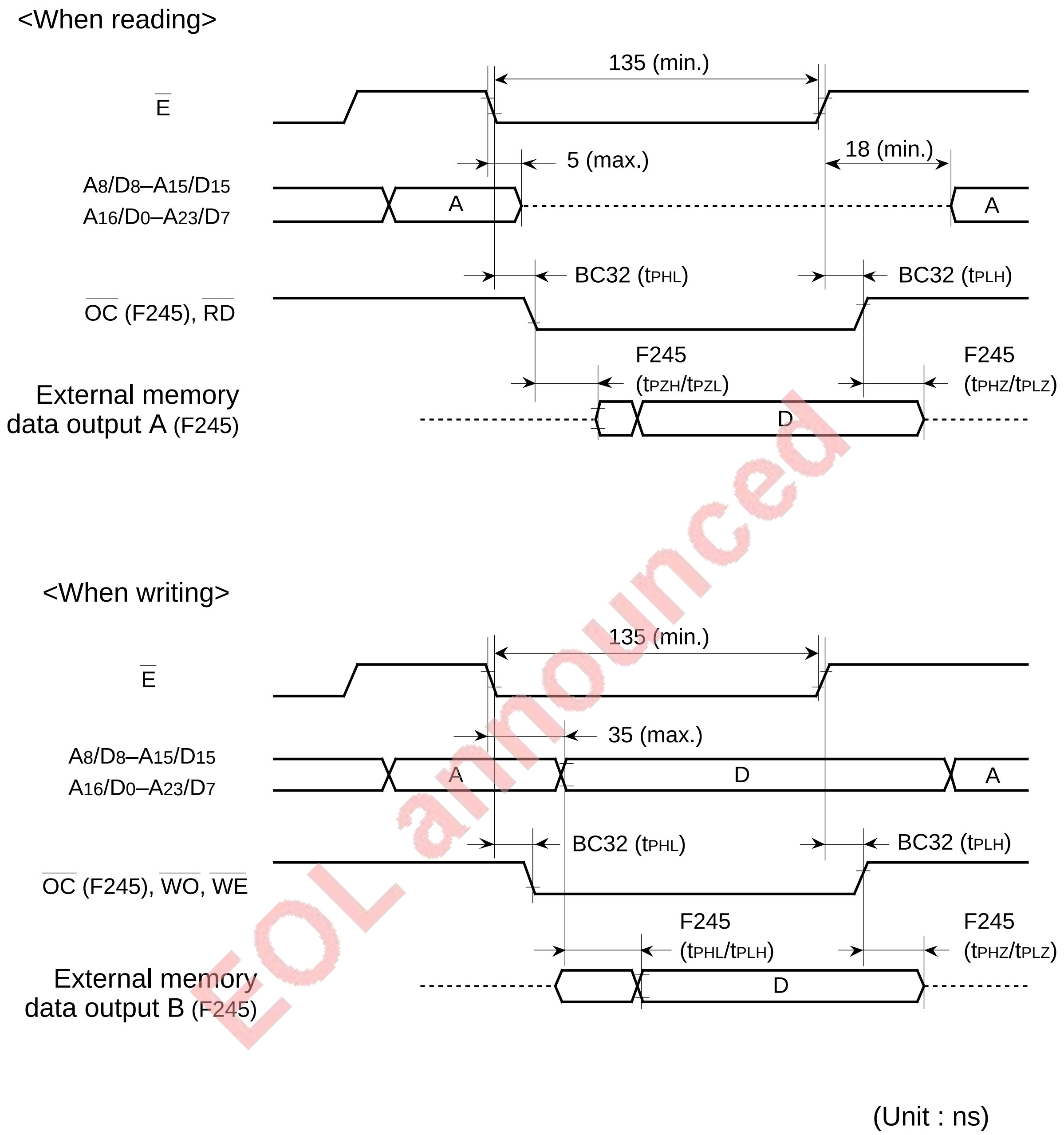
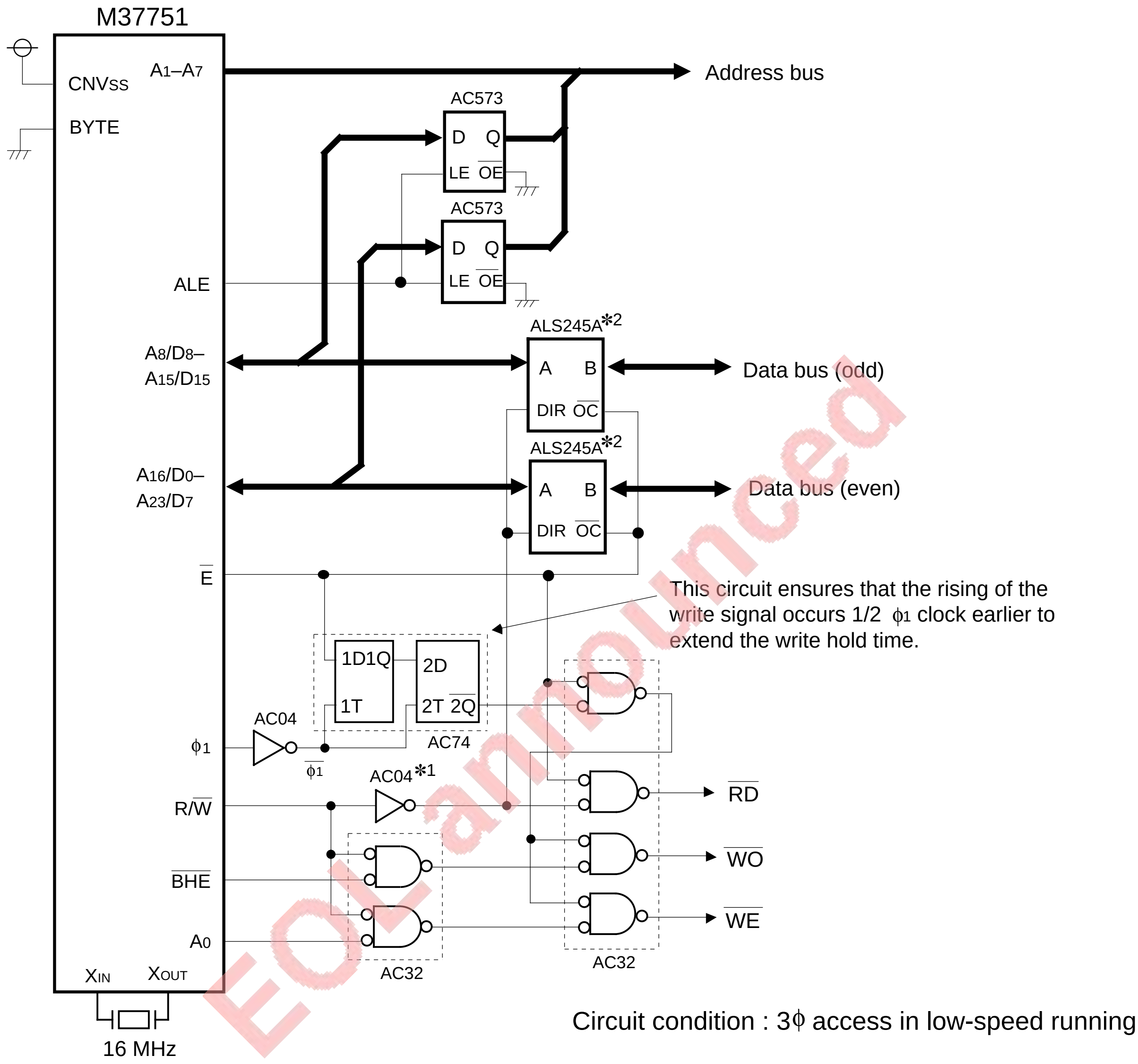


Fig. 17.1.14 Timing chart for sample circuit using bus buffers (at low–speed running-1)



*1: Use the elements of which propagation delay time is within 42.5 ns.
*2: Use the elements of which output enable time is 5 ns or more and output disable time is within 40.5 ns.

Fig. 17.1.15 Example for using bus buffer (at low-speed running-2 : connecting with memory requiring long hold time for write)

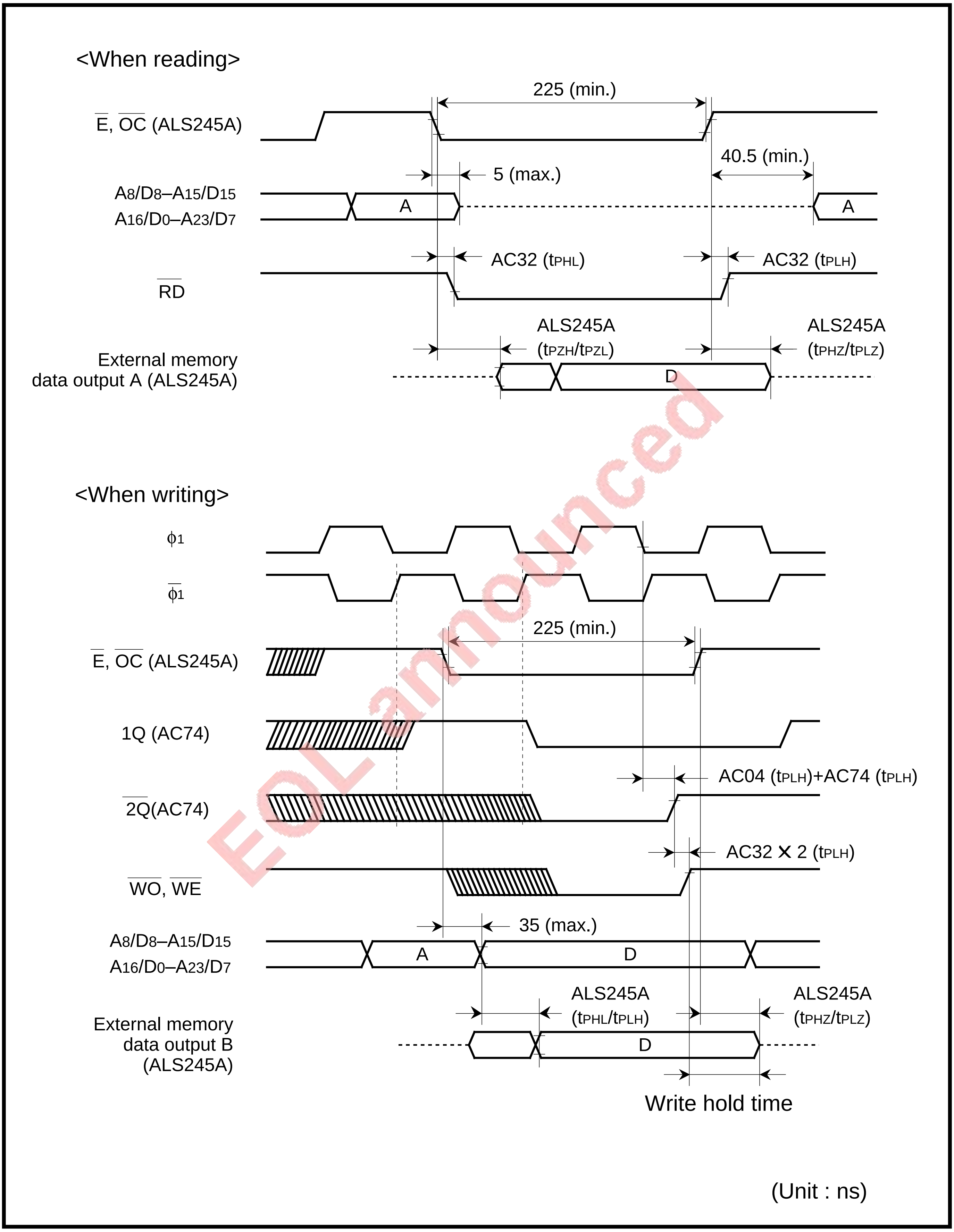
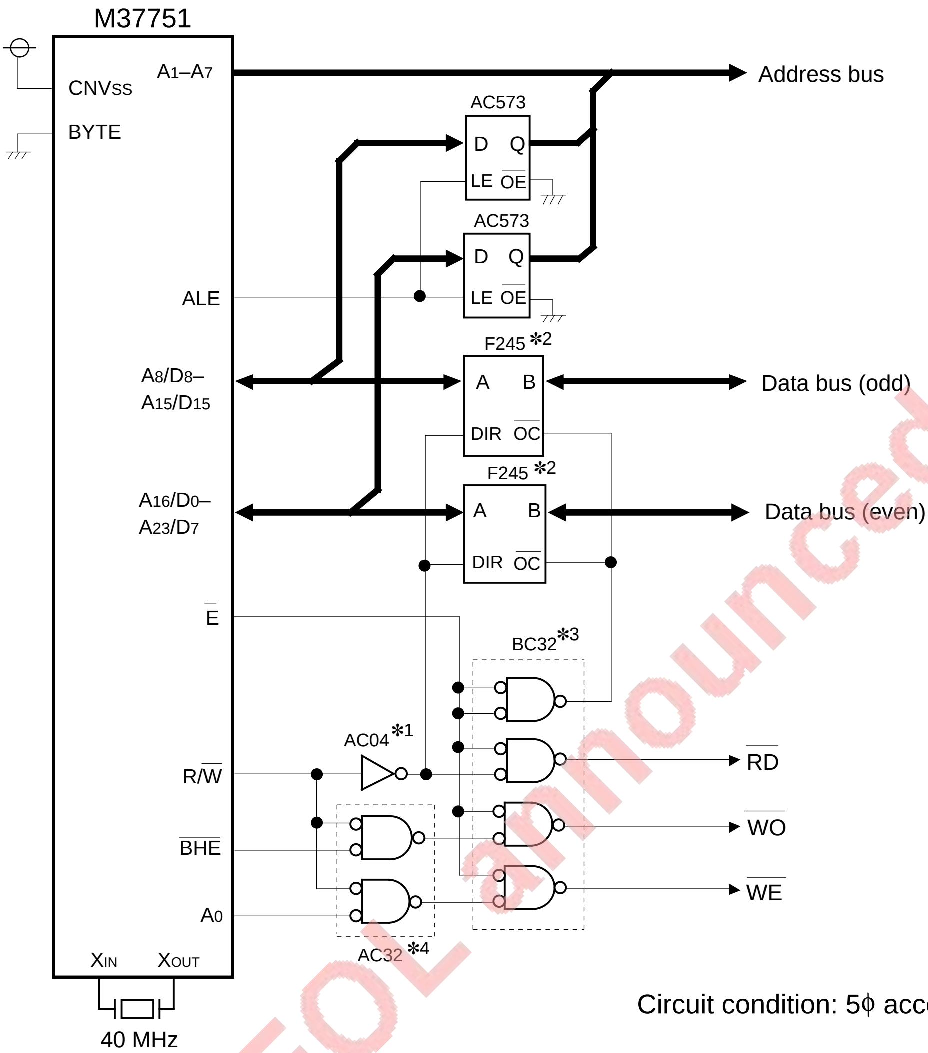


Fig. 17.1.16 Timing chart for sample circuit using bus buffers (at low-speed running-2)



Circuit condition: 5φ access in high-speed running

*1: Use the elements of which propagation delay time is within 45 ns.
*2, *3 : Use the elements of which sum of output disable time in *2 and propagation delay time in *3 is within 15 ns, and the sum of output enable time in *2 and propagation delay time in *3 is 5 ns or more.
*4: Use the elements of which propagation delay time is within 40 ns.

Fig. 17.1.17 Example for using bus buffer (at high-speed running-1)

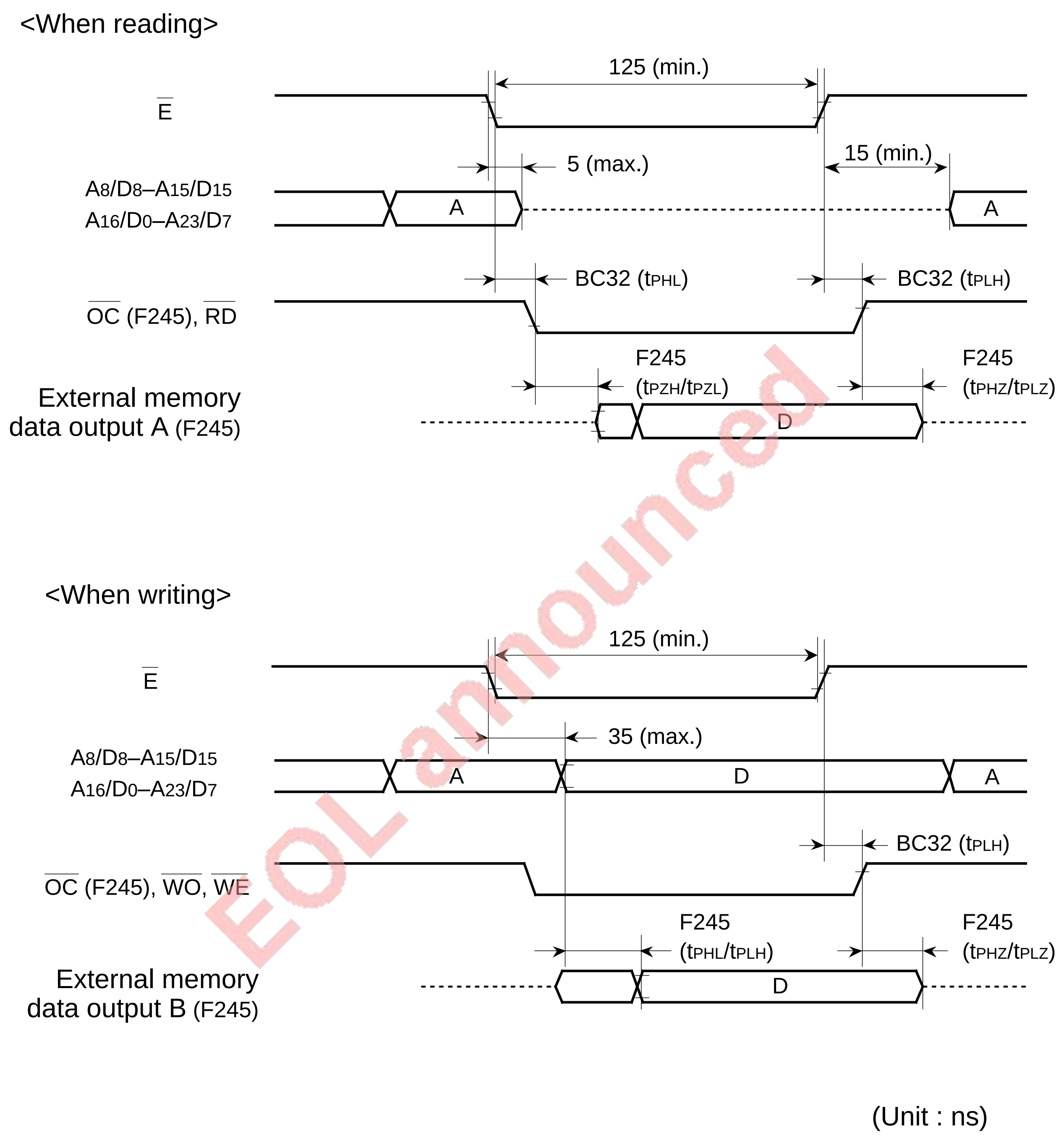
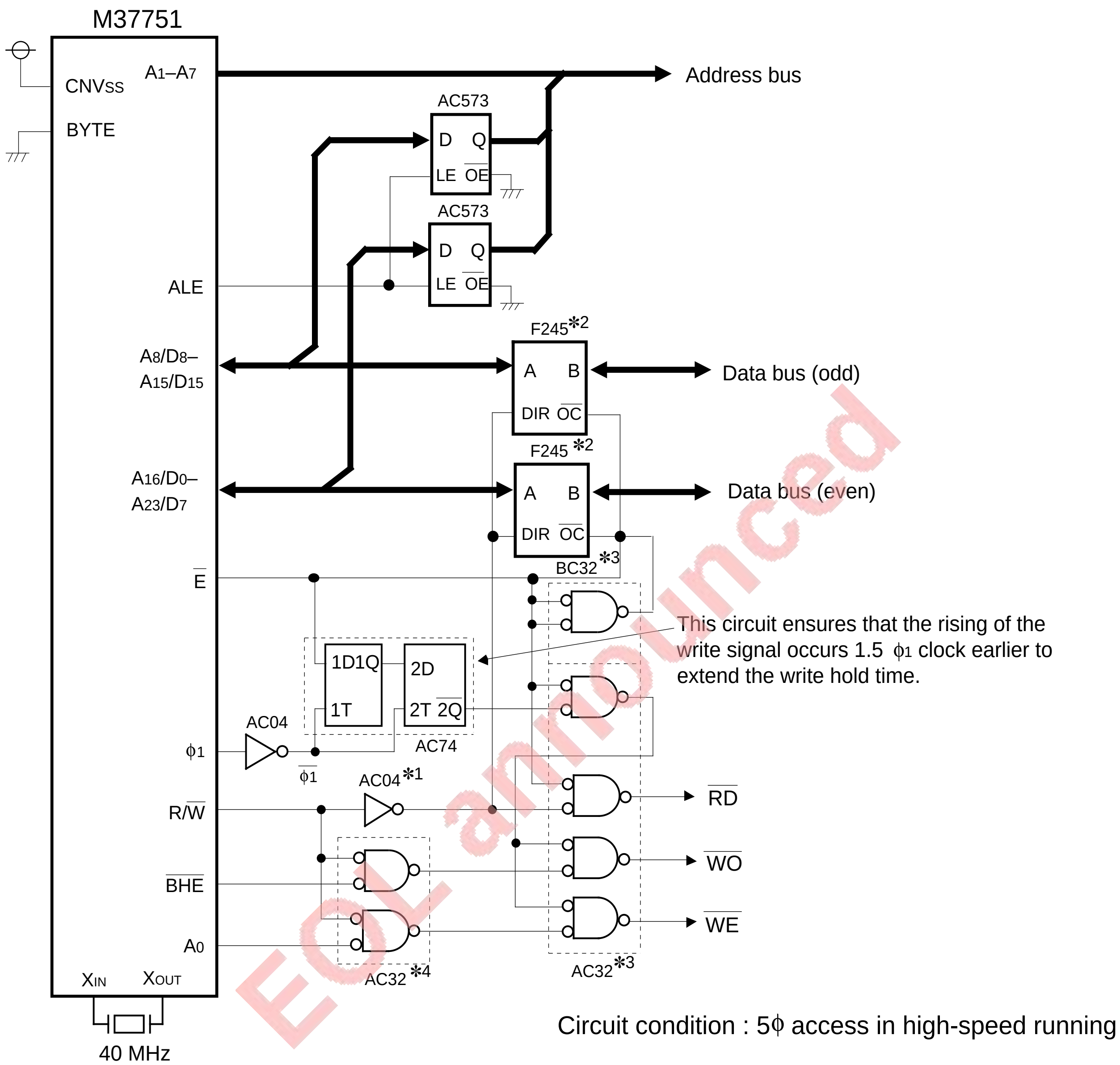


Fig. 17.1.18 Timing chart for sample circuit using bus buffers (at high-speed running-1)

17.1 Memory expansion



Circuit condition : 5φ access in high-speed running

✻1: Use the elements of which propagation delay time is within 45 ns.
✻2, ✻3 : Use the elements of which sum of output disable time in ✻2 and propagation delay time in ✻3 is within 15 ns, and the sum of output enable time in ✻2 and propagation delay time in ✻3 is 5 ns or more.
✻4: Use the elements of which propagation delay time is within 40 ns.

Fig. 17.1.19 Example for using bus buffer (at high-speed running-2 : connecting with memory requiring long hold time for write)

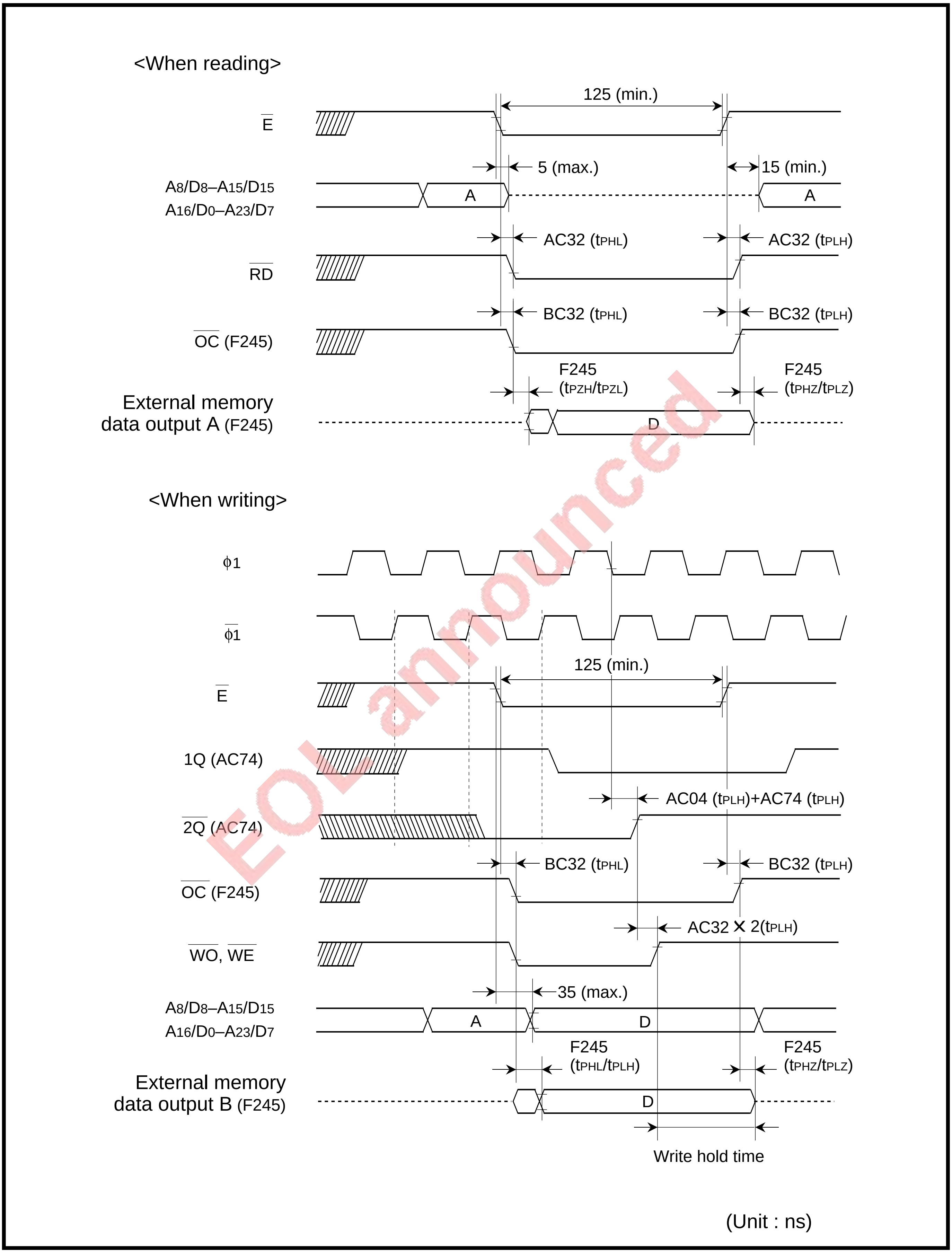


Fig. 17.1.20 Timing chart for sample circuit using bus buffers (at high-speed running-2)

17.1.4 Example of memory expansion

(1) Example of SRAM expansion (minimum model)

Figure 17.1.21 shows a memory expansion example (minimum model) using a 32-Kbyte SRAM in the memory expansion mode at the low-speed running. Figure 17.1.22 shows the timing chart for this example.

Figure 17.1.23 shows a memory expansion example (minimum model) using a 32-Kbyte SRAM in the memory expansion mode at the high-speed running. Figure 17.1.24 shows the timing chart for this example.

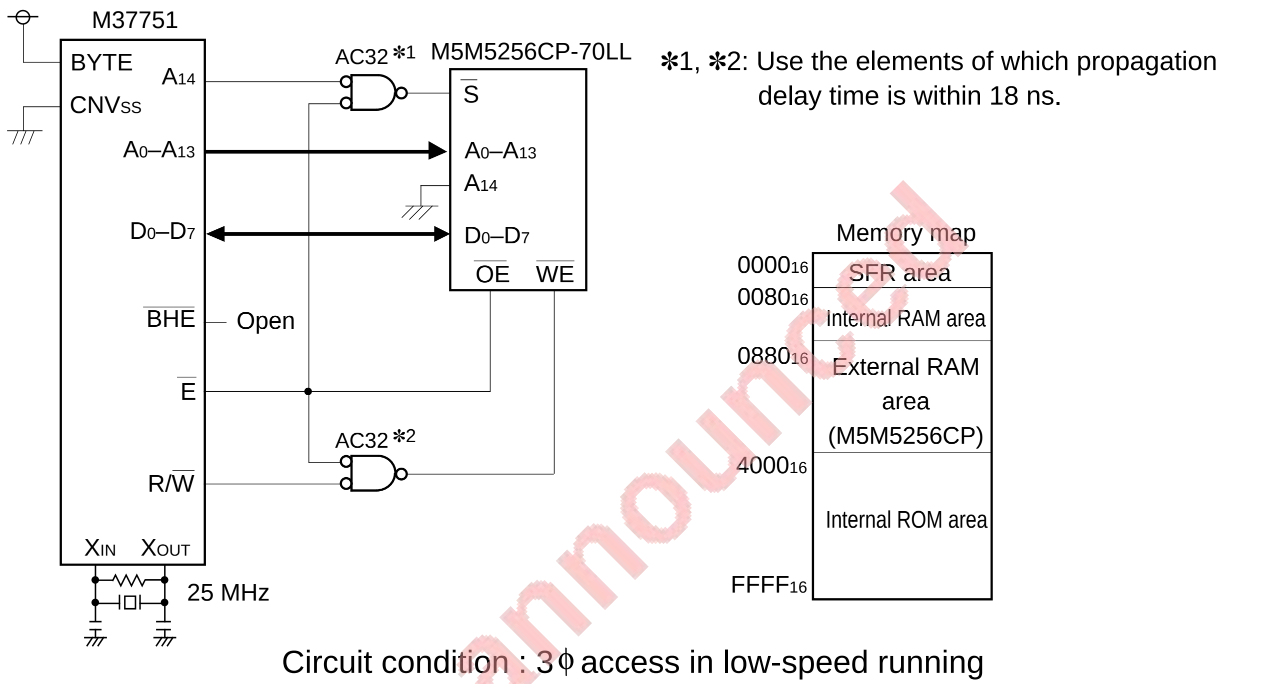


Fig. 17.1.21 Example of SRAM expansion (minimum model at low-speed running)

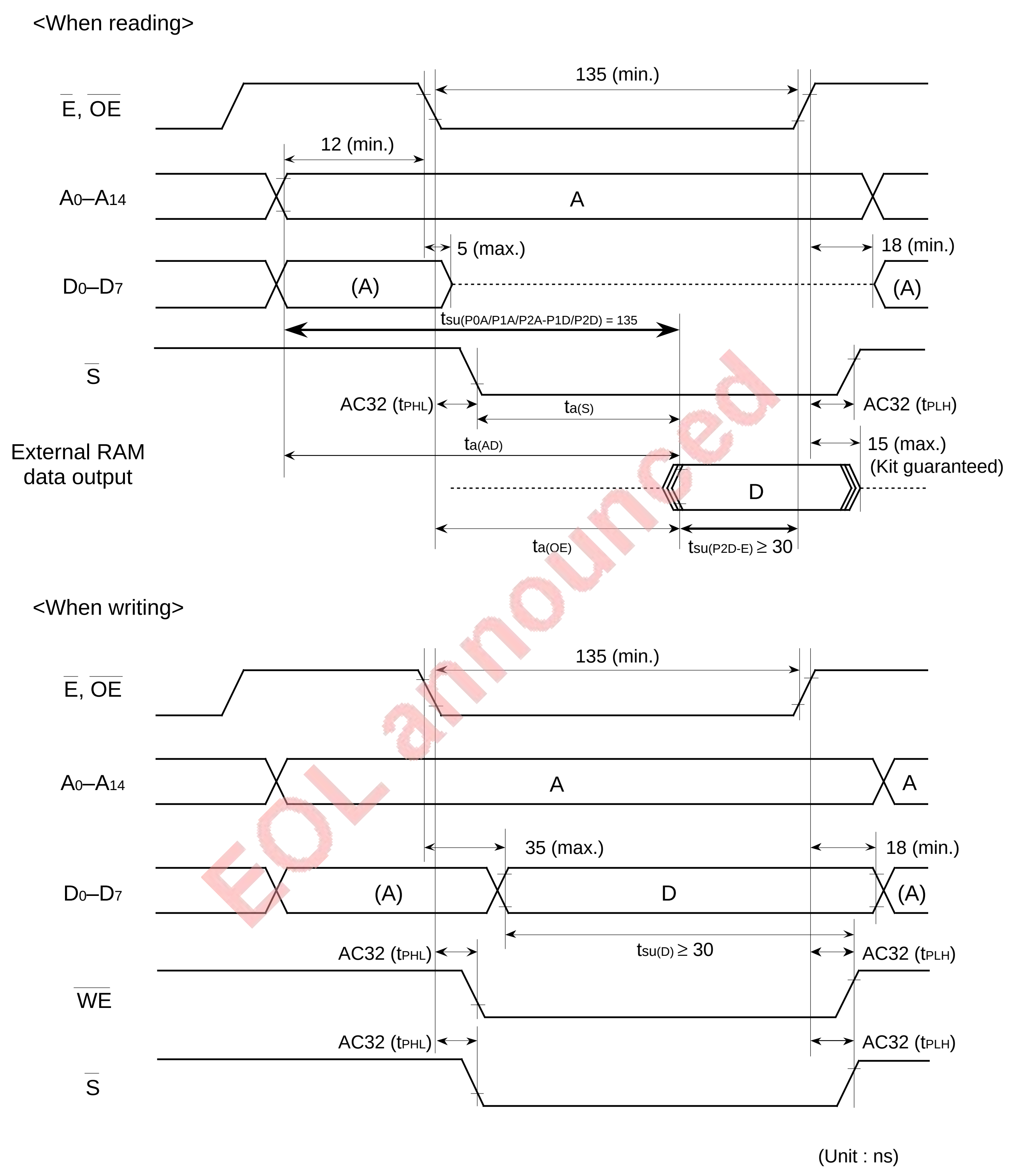


Fig. 17.1.22 Timing chart for SRAM expansion example (minimum model at low-speed running)

17.1 Memory expansion

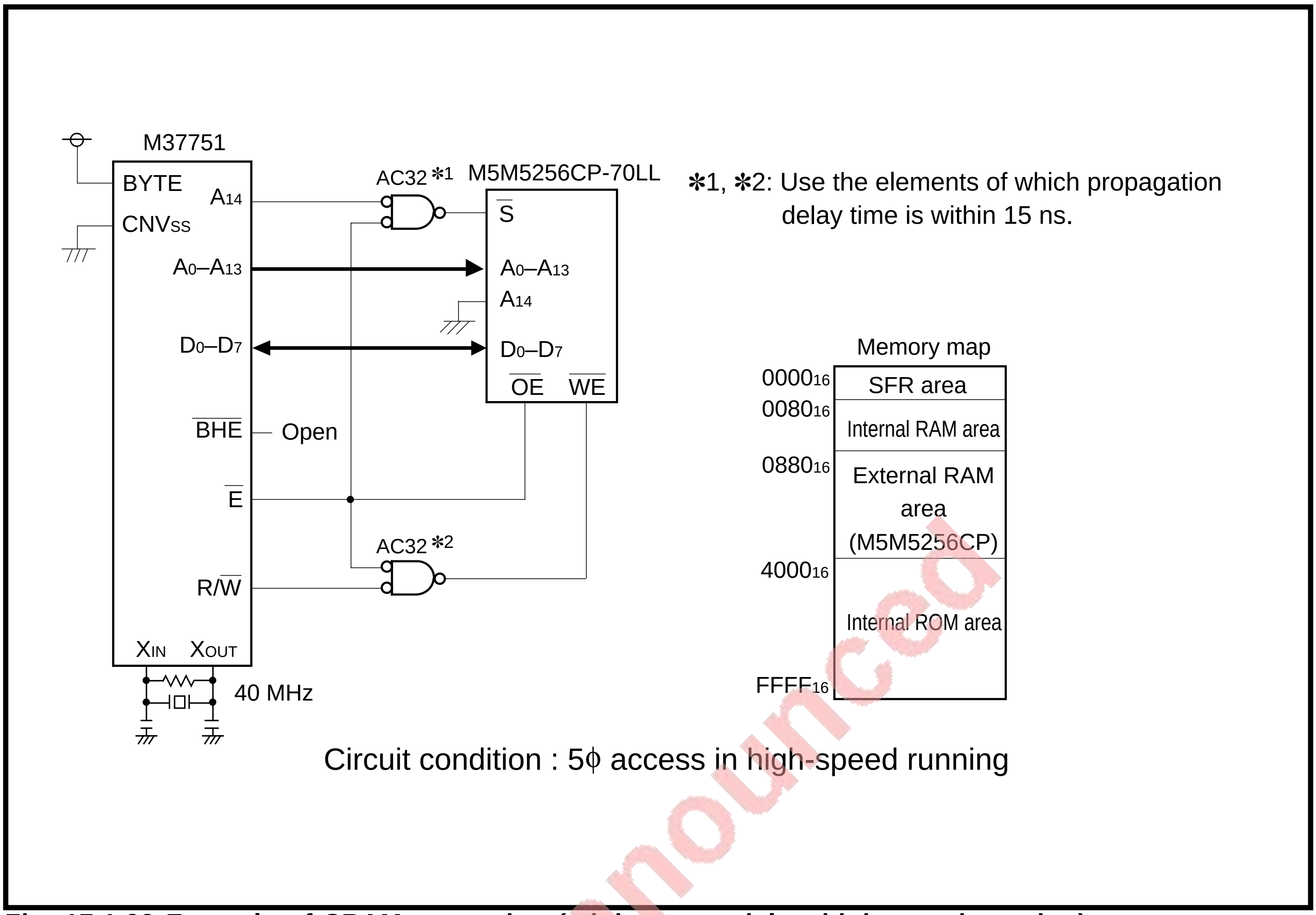


Fig. 17.1.23 Example of SRAM expansion (minimum model at high-speed running)

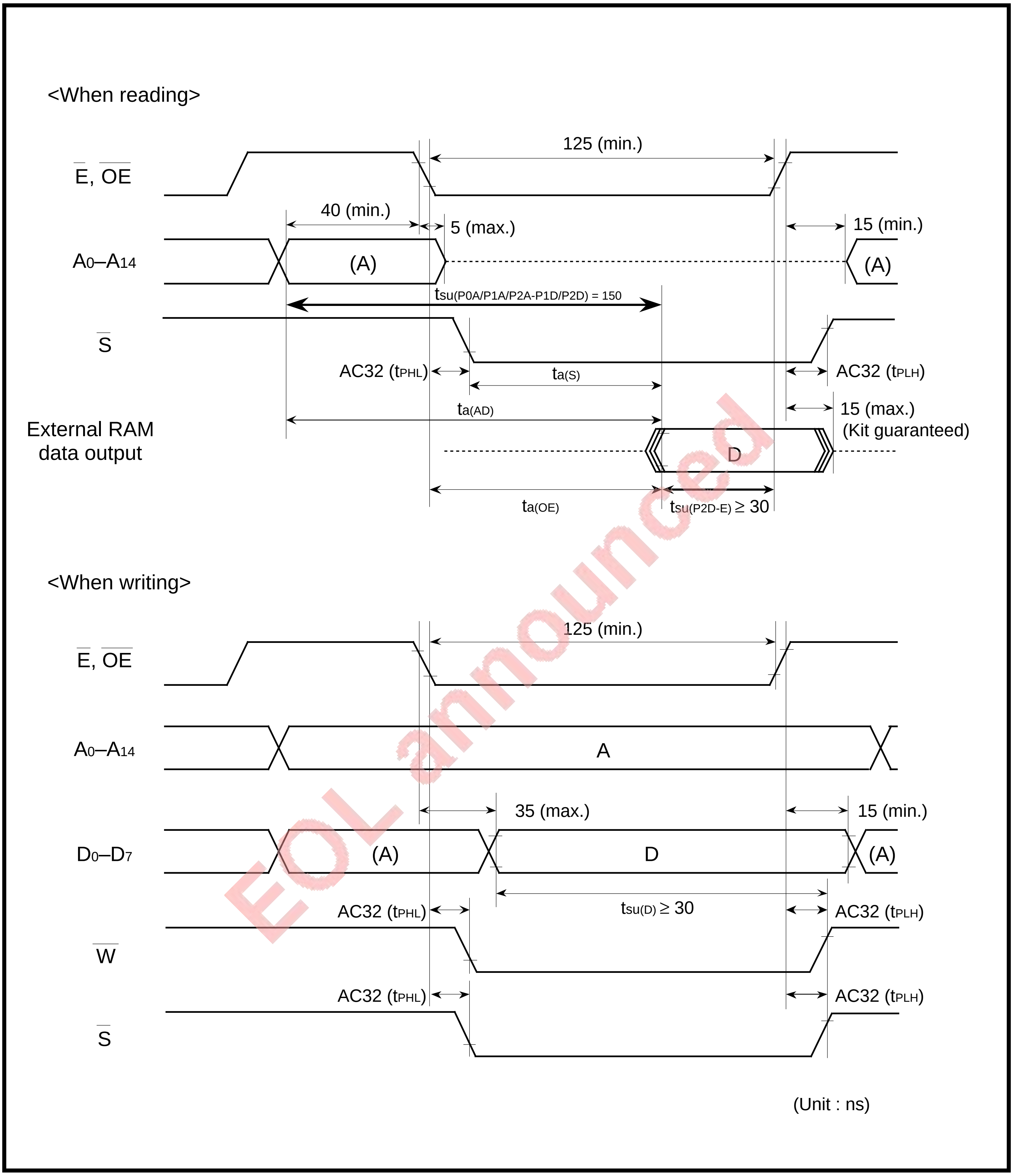


Fig. 17.1.24 Timing chart for SRAM expansion example (minimum model at high-speed running)

(2) Example of ROM expansion (maximum model)

Figure 17.1.25 shows a memory expansion example (maximum model) using a 1-Mbits ROM in the microprocessor mode. Figure 17.1.26 shows the timing chart for this example.

Figure 17.1.27 shows a memory expansion example (maximum model) using a 1-Mbits ROM in the microprocessor mode. Figure 17.1.28 shows the timing chart for this example.

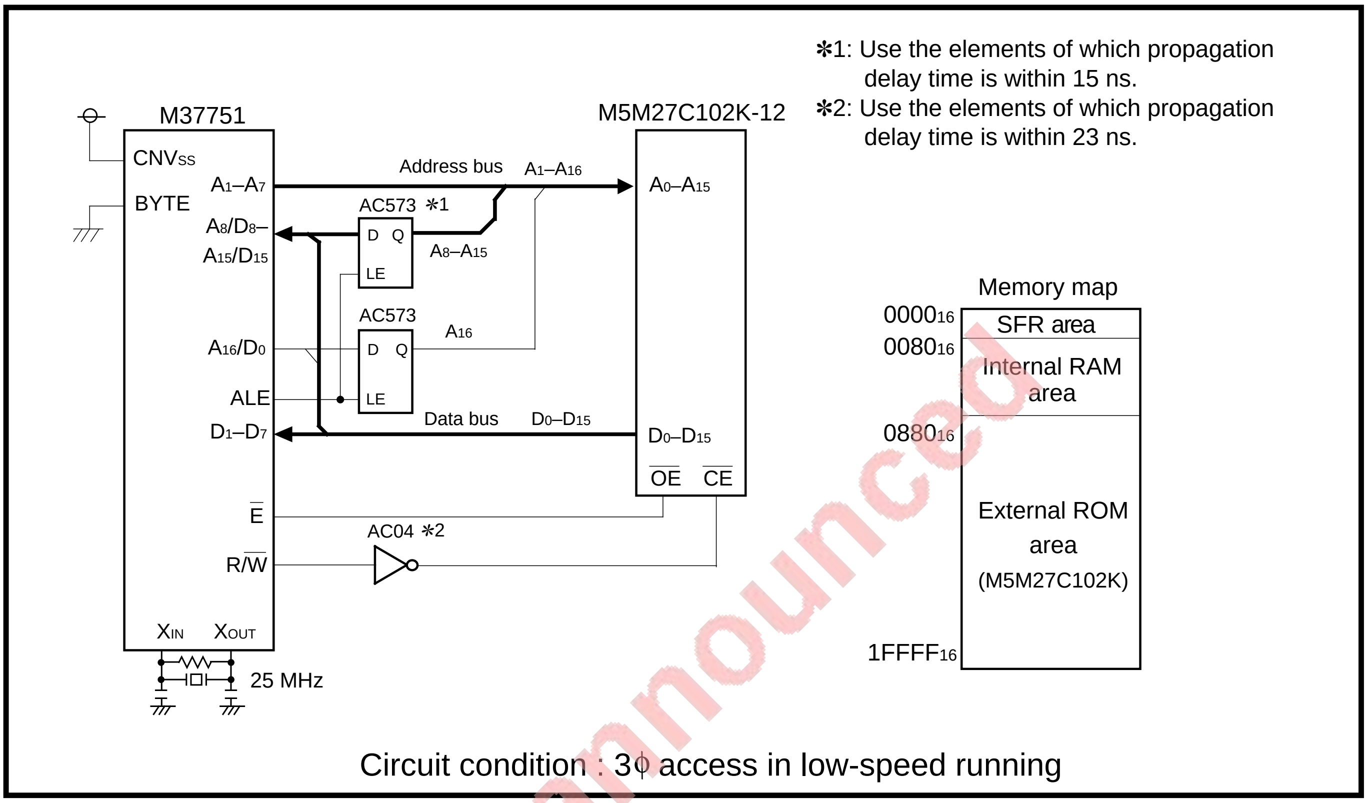


Fig. 17.1.25 Example of ROM expansion (maximum model at low-speed running)

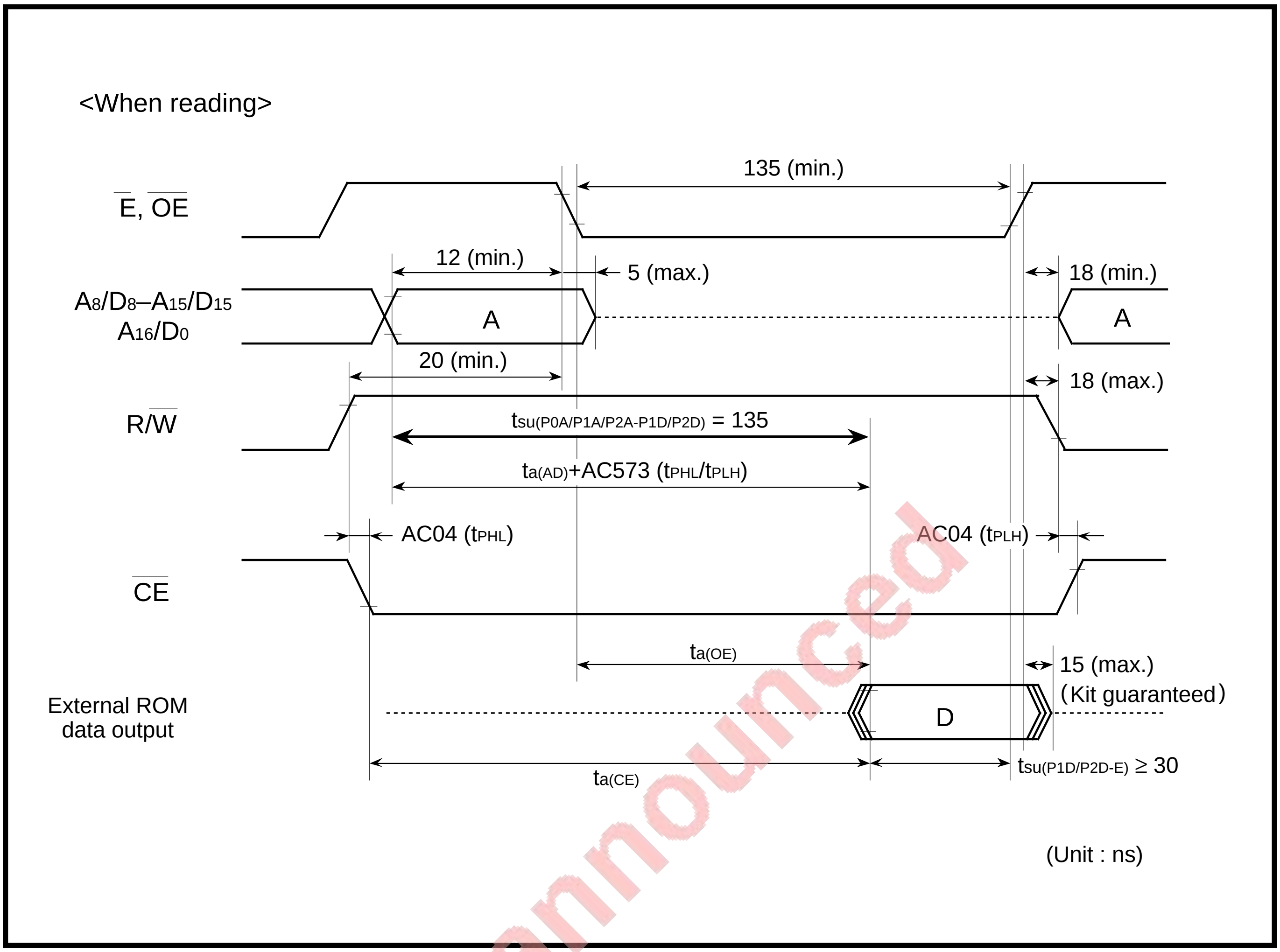


Fig. 17.1.26 Timing chart for ROM expansion example (maximum model at low-speed running)

17.1 Memory expansion

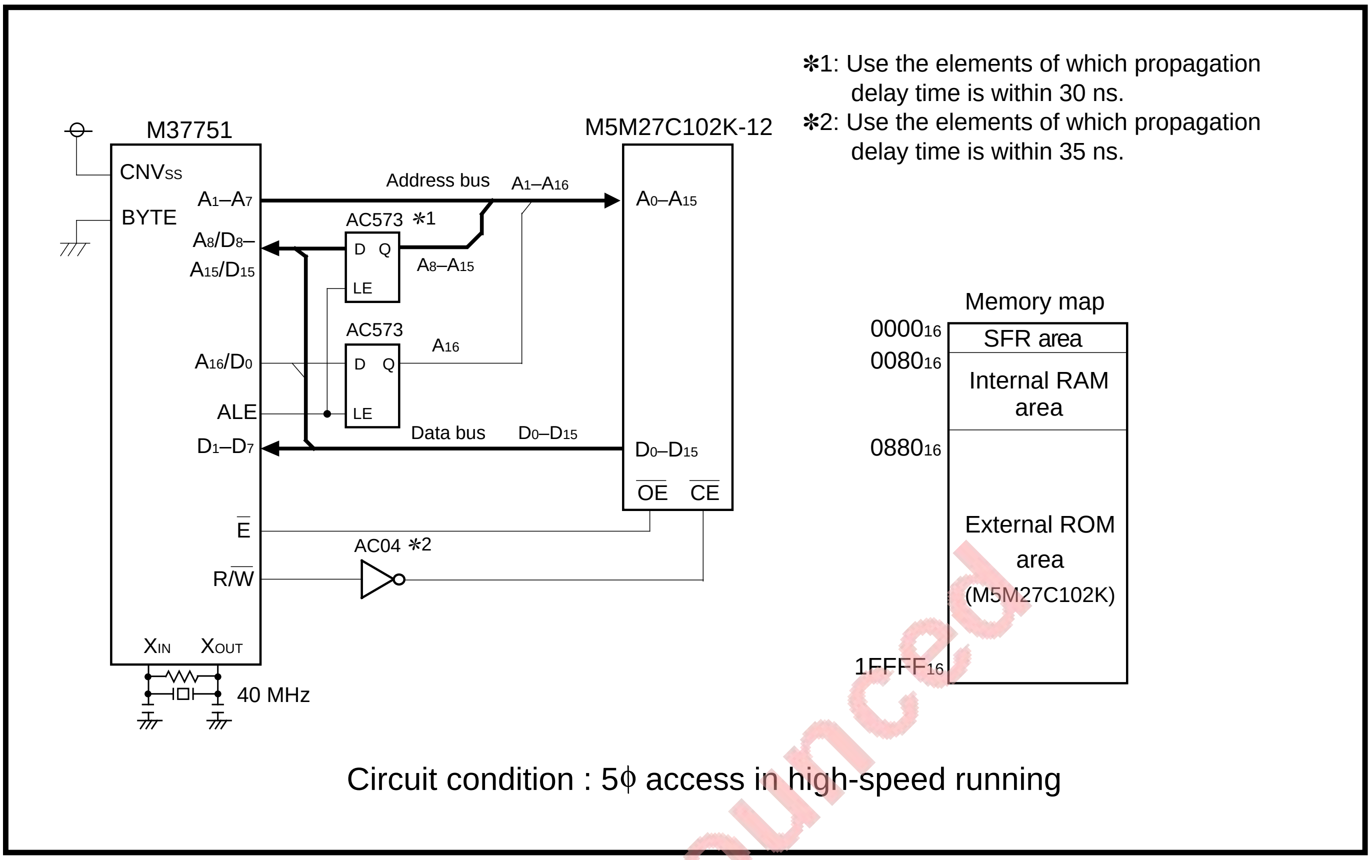


Fig. 17.1.27 Example of ROM expansion (maximum model at high-speed running)

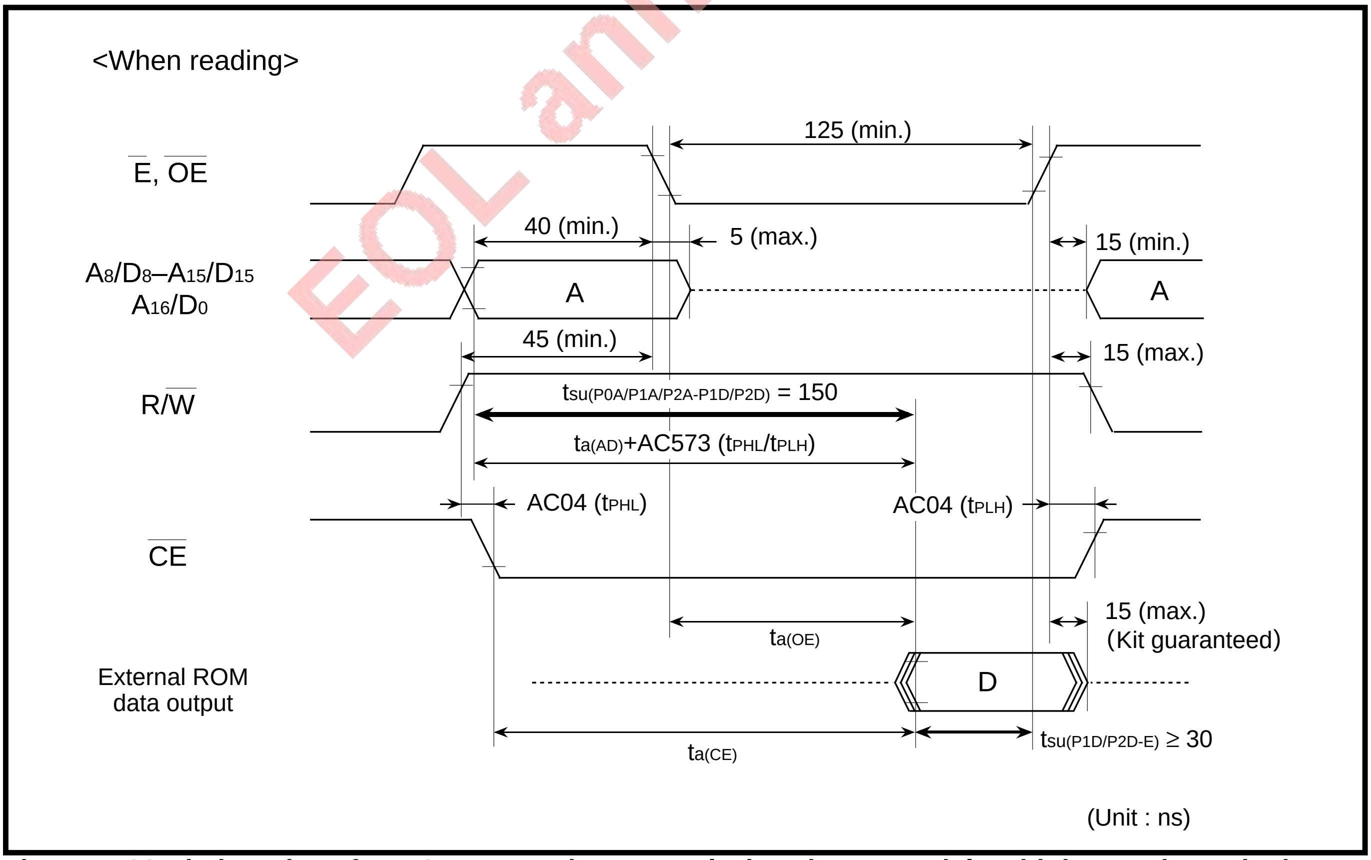


Fig. 17.1.28 Timing chart for ROM expansion example (maximum model at high-speed running)

(3) Example of ROM and SRAM expansion (maximum model)

Figure 17.1.29 shows a memory expansion example (maximum model) using two 32-Kbytes ROM and two 32-Kbytes SRAM in the microprocessor mode at the low-speed running. Figure 17.1.30 shows the timing chart for this example.

Figure 17.1.31 shows a memory expansion example (maximum model) using two 32-Kbytes ROM and two 32-Kbytes SRAM in the microprocessor mode at the high-speed running. Figure 17.1.32 shows the timing chart for this example.

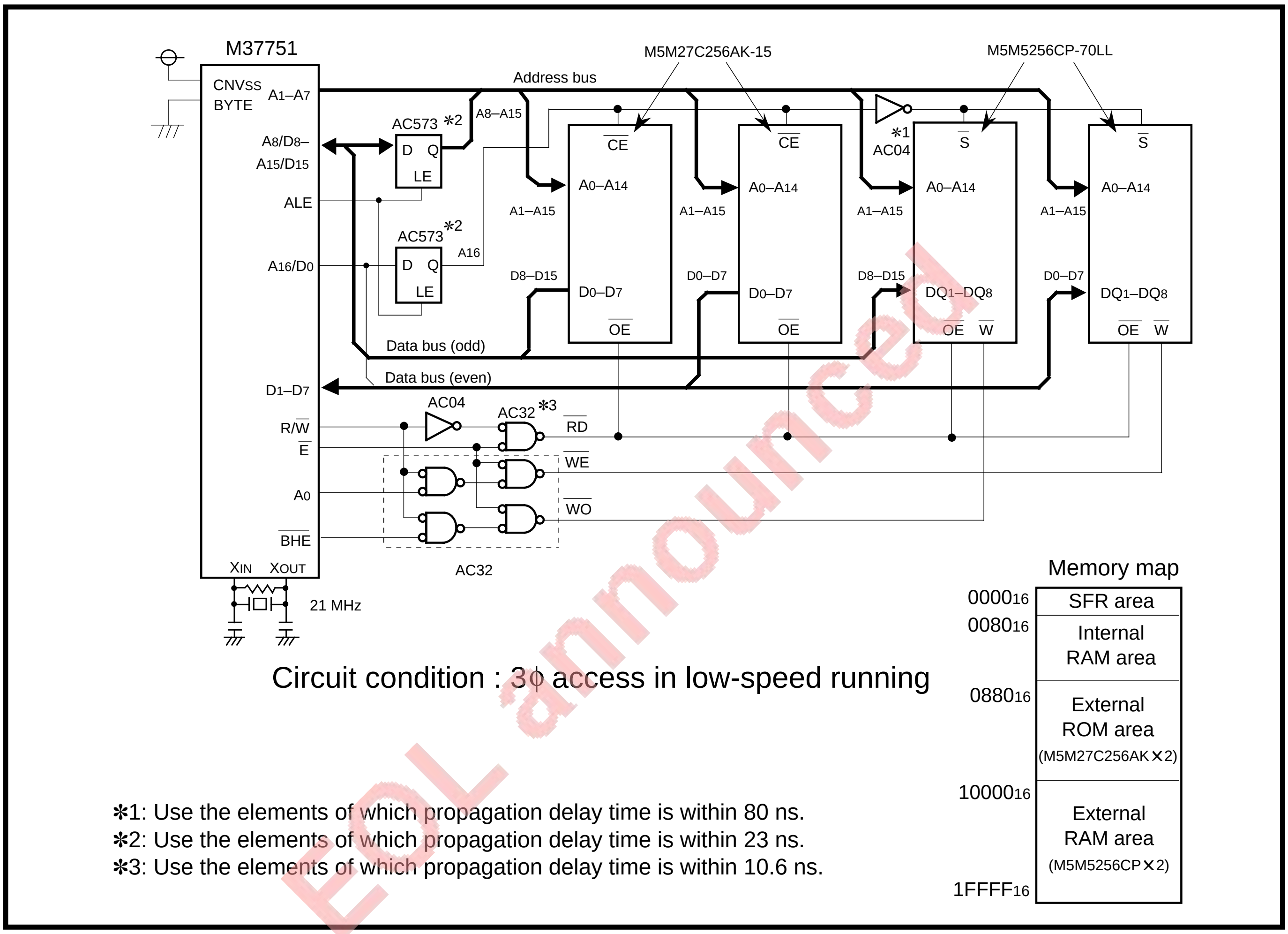


Fig. 17.1.29 Example of ROM and SRAM expansion (maximum model at low-speed running)

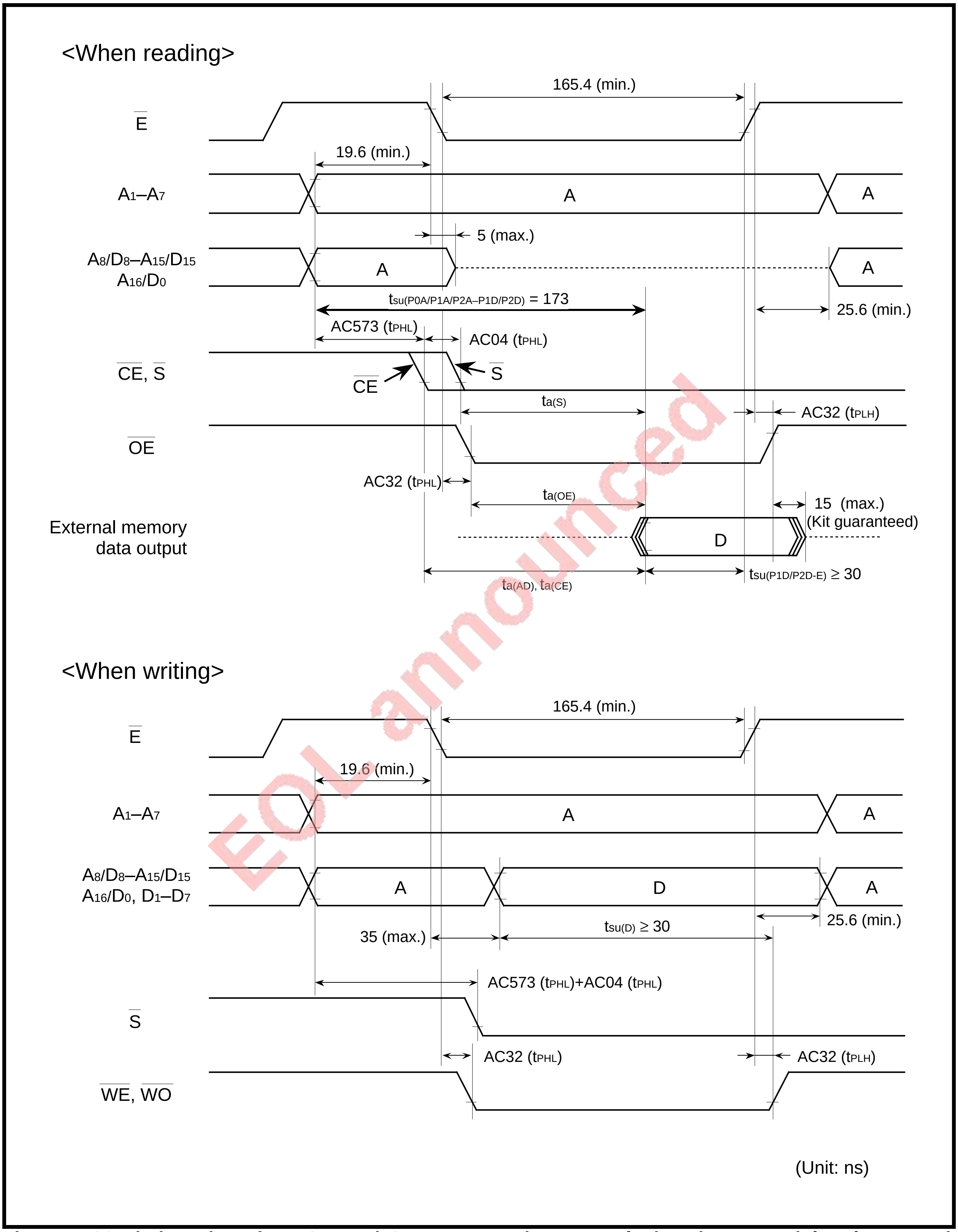
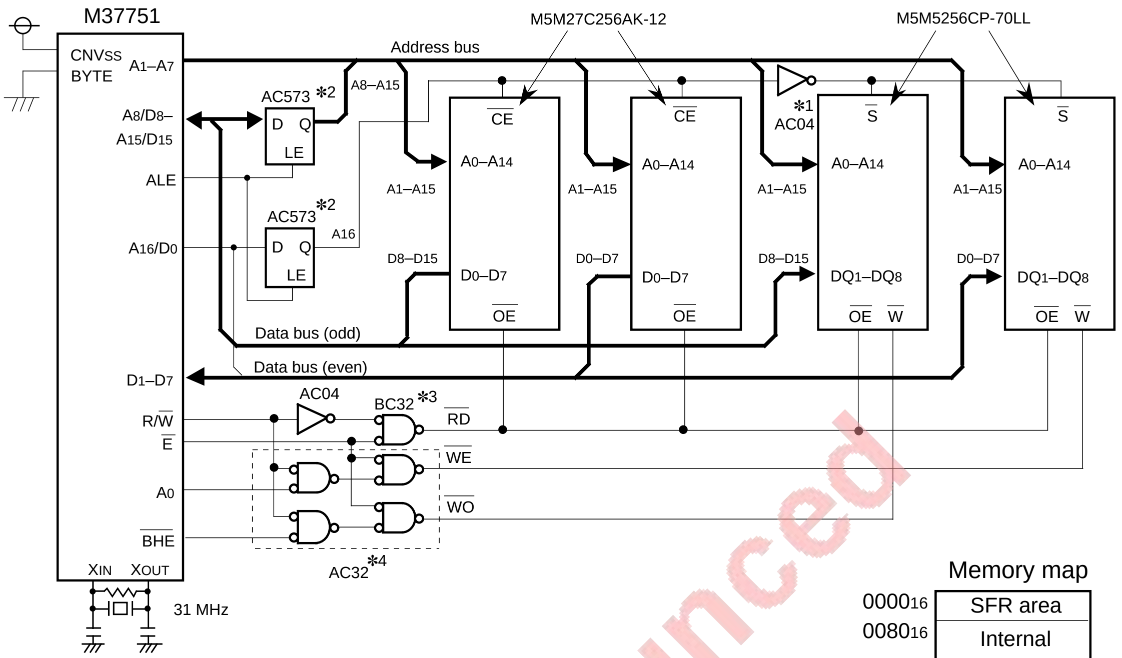


Fig. 17.1.30 Timing chart for ROM and SRAM expansion example (maximum model at low-speed running)



Circuit condition : 4φ access in high-speed running

Memory map

000016	SFR area
008016	Internal RAM area
088016	External ROM area (M5M27C256AK×2)
1000016	External RAM area (M5M5256CP×2)
1FFFF16	

✻1: Use the elements of which propagation delay time is within 50 ns.
✻2: Use the elements of which propagation delay time is within 30 ns.
✻3: Use the elements of which propagation delay time is within 7.2 ns.
✻4: Use the elements of which propagation delay time is within 22.2 ns.

Fig. 17.1.31 Example of ROM and SRAM expansion (maximum model at high-speed running)

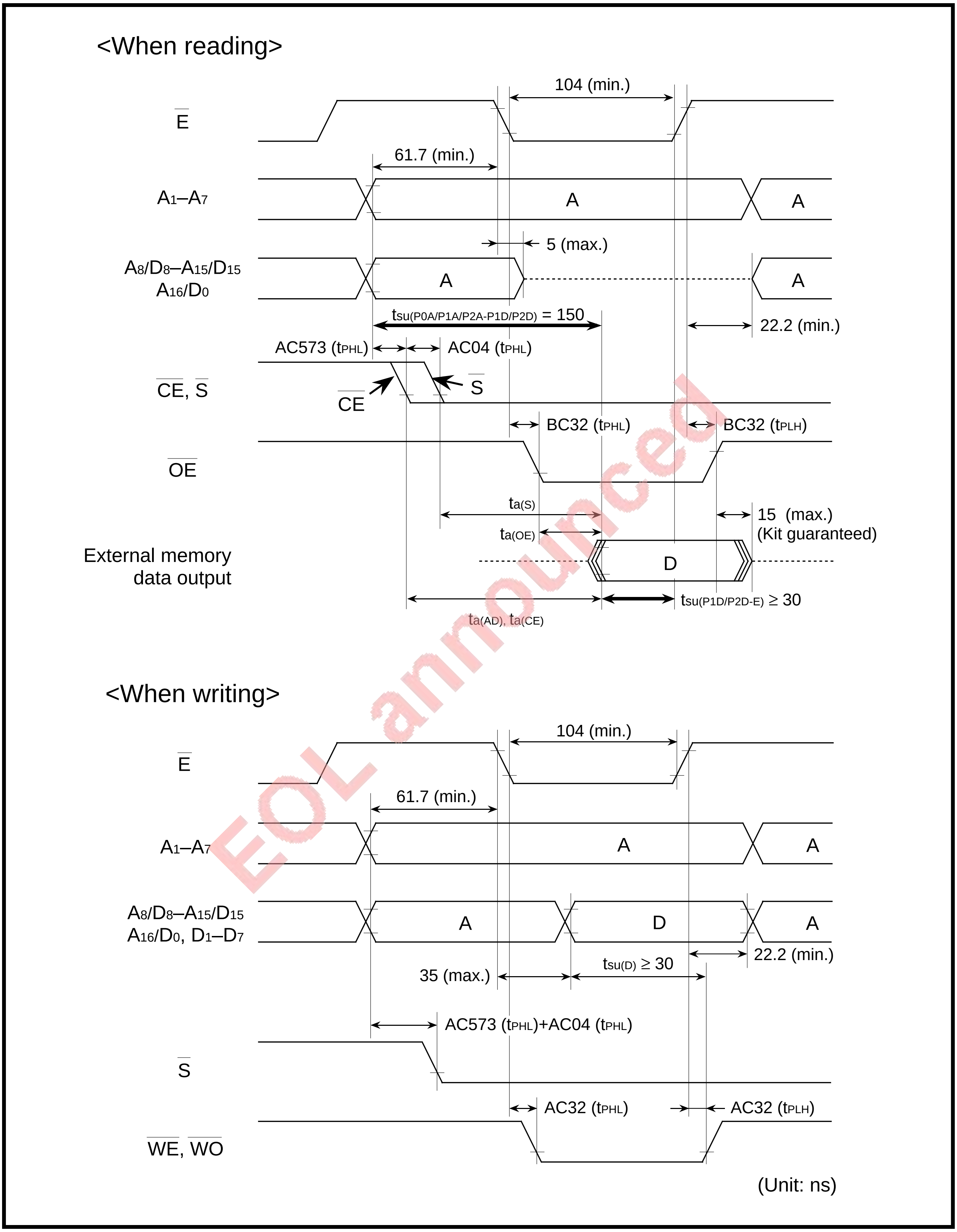


Fig. 17.1.32 Timing chart for ROM and SRAM expansion example (maximum model at high-speed running)

17.1.5 Example of I/O expansion

(1) Example of port expansion circuit using M66010FP

Figure 17.1.33 shows an example of a port expansion circuit using the M66010FP. Although Figure 17.1.33 is an expansion example in the high-speed running, when using 1.923 MHz or less frequency for Serial I/O transfer clock, the same expansion is possible regardless of the bus cycle.
About Serial I/O control in this expansion example is described below.
In this example, 8-bit data transmission/reception is performed 3 times by using UART0 and 24-bit port expansion is realized. Setting of UART0 is described below:

- Clock synchronous serial I/O mode: Transmission/Reception enable state
- Internal clock is selected. Transfer clock frequency is 1.66 MHz.
- LSB first

The control process is described below:

① Output "L" level from port $P4_5$. (Expansion I/O ports of M66010FP become floating state by this signal.)
② Output "H" level from port $P4_5$.
③ Output "L" level from port $P4_4$.
④ Transmit/Receive 24-bit data by using UART0.
⑤ Output "H" level from port $P4_4$.

Figure 17.1.34 shows serial transfer timing between M37751 and M66010FP.

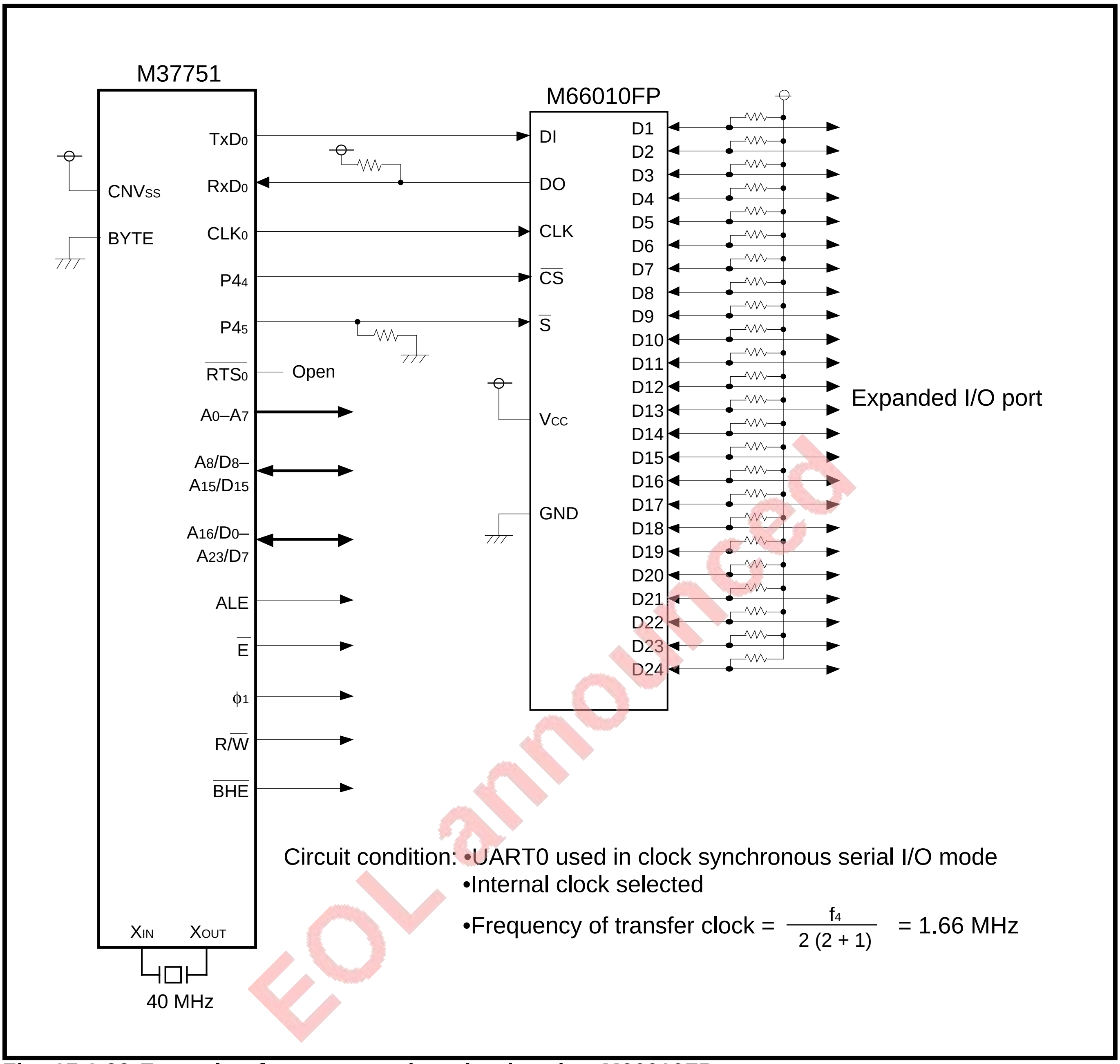


Fig. 17.1.33 Example of port expansion circuit using M66010FP

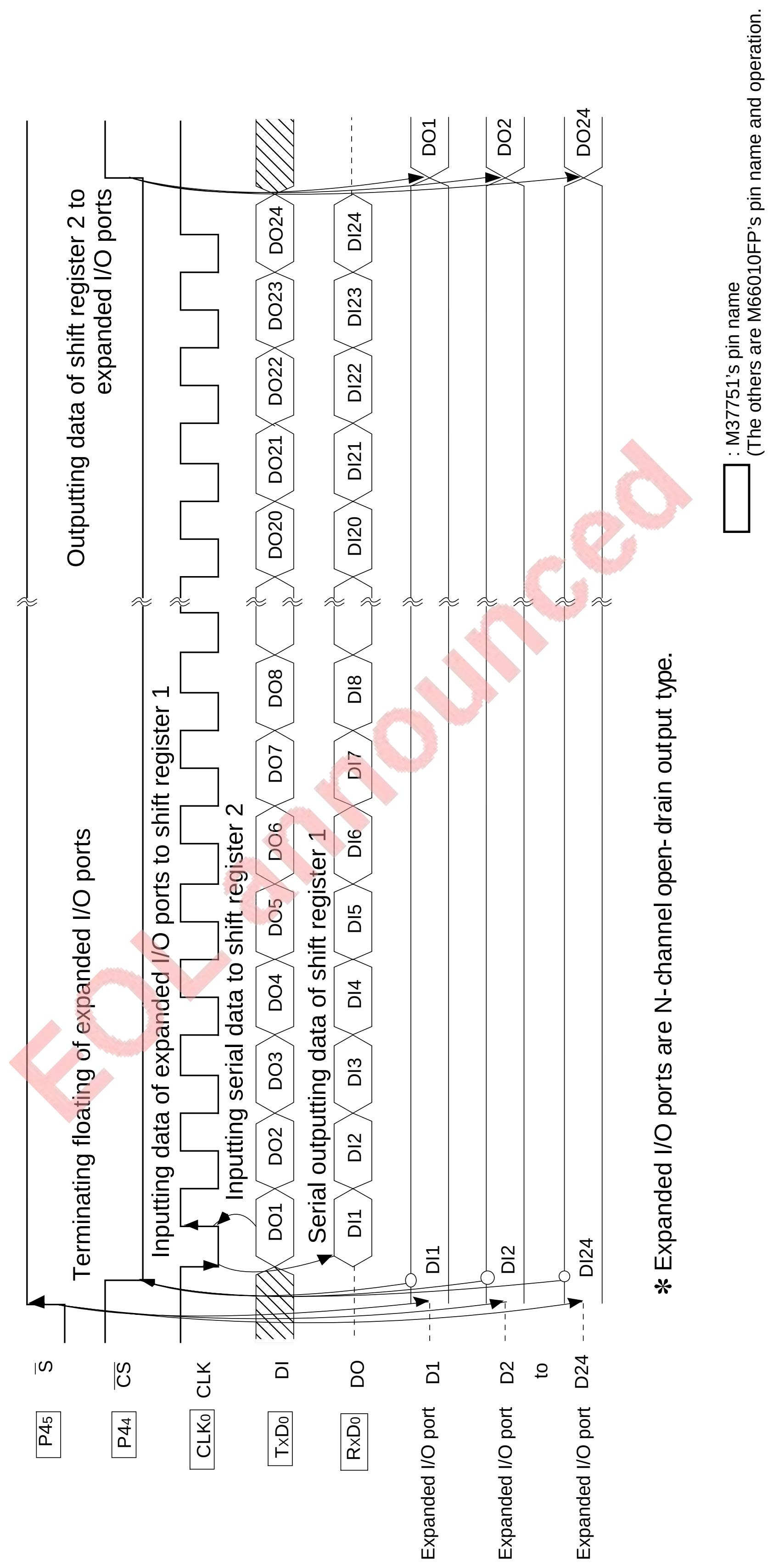


Fig. 17.1.34 Serial transfer timing between M37751 and M66010FP

MEMORANDUM

CHAPTER 18

PROM VERSION

PROM VERSION

In the PROM version, programming/reading to and from the built-in PROM can be performed by using a general-purpose PROM programmer and a programming adapter.
The PROM version has the following two types :

- ●One time PROM version
  Programming to the PROM can be performed once.
  This version is suitable for a small quantity of and various productions.
- ●EPROM version
  Programming to the PROM can be performed repeatedly because a program can be erased by exposing the erase window on the top of the package to an ultraviolet light source.
  This version can be used only for program development, evaluation only.

The PROM version have the same functions as the mask ROM version except that the former have a built-in PROM. Table 18.1.1 lists the product expansion of the PROM version.

Table 18.1.1 Product expansion of PROM version

Type name	PROM size	RAM size
M37751E6C-XXXFP (M37751E6CFP)	One time PROM 49152 bytes	2048 bytes
M37751E6CFS	EPROM 49152 bytes	

18.1 EPROM mode

The PROM version can select the normal operating mode which performs the same operation as that of the mask ROM version, or the EPROM mode which enables to program/read to/from the built-in PROM.
When "L" level is input to the $\overline{\text{RESET}}$ pin, the PROM version enters the EPROM mode.

18.1.1 Pin description

Table 18.1.2 lists the pin description in the EPROM mode.

Table 18.1.2 Pin description in EPROM mode

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power source input	––	Apply 5 V ± 10% to V_{CC} pin, and 0 V to V_{SS} pin.
CNV_{SS}	V_{PP} input	Input	Apply V_{PP} level when programming or verifying.
BYTE			
$\overline{\text{RESET}}$	Reset input	Input	Connect to V_{SS} pin.
X_{IN}	Clock input	Input	Connect a ceramic resonator or a quartz-crystal oscillator between X_{IN} and X_{OUT} pins. When an external generated clock is input, the clock must be input to X_{IN} pin, and X_{OUT} pin must be left open.
X_{OUT}	Clock output	Output	
$\overline{\text{E}}$	Enable output	Output	Open.
AV_{CC}, AV_{SS}	Analog power source input	––	Connect AV_{CC} pin to V_{CC} pin and AV_{SS} pin to V_{SS} pin.
V_{REF}	Reference voltage input	Input	Connect to V_{SS} pin.
$P0_0$–$P0_7$	Address input (A_0–A_7)	Input	Input pins for A_0–A_7 of addresses.
$P1_0$–$P1_7$	Address input (A_8–A_{15})	Input	Input pins for A_8–A_{15} of addresses.
$P2_0$–$P2_7$	Data input/output (D_0–D_7)	I/O	I/O pins for data D_0–D_7.
$P3_0$–$P3_3$	Input port P3	Input	Connect to V_{SS} pin.
$P4_0$–$P4_7$	Input port P4	Input	Connect to V_{SS} pin.
$P5_0$	Control input	Input	$P5_0$ functions as $\overline{\text{PGM}}$ input pin.
$P5_1$			$P5_1$ functions as $\overline{\text{OE}}$ input pin.
$P5_2$			$P5_2$ functions as $\overline{\text{CE}}$ input pin.
$P5_3$–$P5_6$	Input port P5		Connect to V_{CC} pin.
$P5_7$			Connect to V_{SS} pin.
$P6_0$–$P6_7$	Input port P6	Input	Connect to V_{SS} pin.
$P7_0$–$P7_7$	Input port P7	Input	
$P8_0$–$P8_7$	Input port P8	Input	

18.1 EPROM mode

18.1.2 Programming/reading

EPROM mode can perform programming/reading to and from the built-in PROM with the same manner as M5M27C101K. However, there is no device identification code. Accordingly, programming conditions must be set carefully. Perform the programming to addresses 14000_{16} to $1FFFF_{16}$.

Table 18.1.3 lists the pin correspondence with M5M27C101K. Figure 18.1.1 shows the pin connections in EPROM mode. Table 18.1.4 lists the built-in PROM states in EPROM mode.

Table 18.1.3 Pin correspondence with M5M27C101K

	M37751E6C-XXXFP (M37751E6CFP) M37751E6CFS	M5M27C101K
V_{CC}	V_{CC}	V_{CC}
V_{PP} input	CNVss, BYTE	V_{PP}
V_{SS}	V_{SS}	V_{SS}
Address input	P0, P1	A_0–A_{15}
Data I/O	P2	D_0–D_7
$\overline{CE}$ input	$P5_2$	$\overline{CE}$
$\overline{OE}$ input	$P5_1$	$\overline{OE}$
$\overline{PGM}$ input	$P5_0$	$\overline{PGM}$

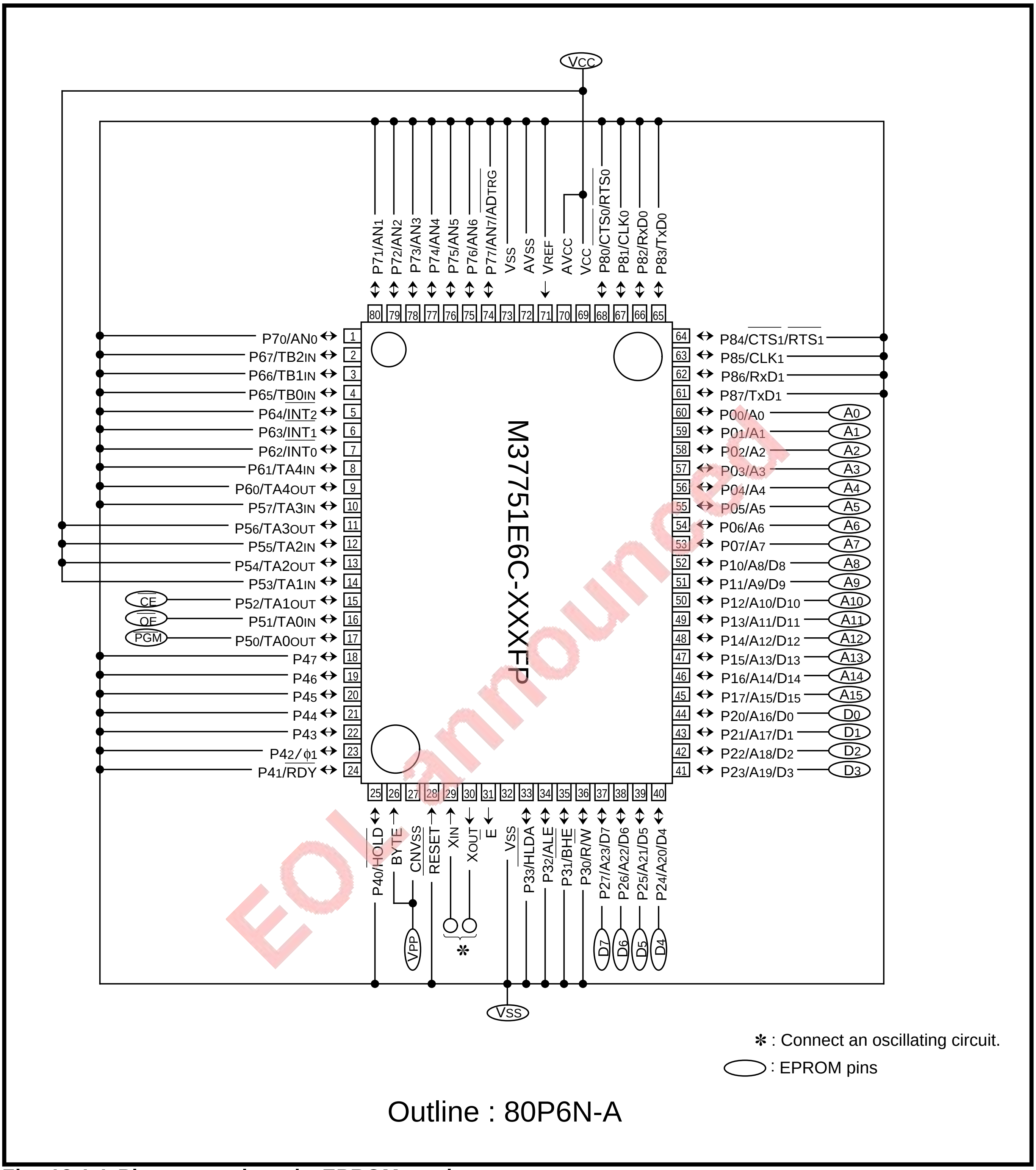


Fig. 18.1.1 Pin connections in EPROM mode

18.1 EPROM mode

Table 18.1.4 Built-in PROM state in EPROM mode

Mode \ Pin name	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	V_{PP}	V_{CC}	Data I/O
Read-out	V_{IL}	V_{IL}	×	5 V	5 V	Output
Output disable	V_{IL}	V_{IH}	×	5 V	5 V	Floating
	V_{IH}	×	×	5 V	5 V	Floating
Program	V_{IL}	V_{IH}	V_{IL}	12.5 V	6 V	Input
Program verify	V_{IL}	V_{IL}	V_{IH}	12.5 V	6 V	Output
Program disable	V_{IH}	V_{IH}	V_{IH}	12.5 V	6 V	Floating

× : It may be V_{IL} or V_{IH}.

(1) Read

When $\overline{CE}$ and $\overline{OE}$ pins are set to “L” level and an address is input to address input pins, the data of the specified address, input address, is output externally from data I/O pins.
When $\overline{CE}$ and $\overline{OE}$ pins are set to “H” level, data I/O pins enter the floating state.

(2) Program (Write)

When $\overline{CE}$ pin is set to “L” level and $\overline{OE}$ pin is set to “H” level and V_{PP} level is applied to V_{PP} pin, programming to the built-in PROM becomes possible.
Input an address to address input pins and supply data to be programmed to data I/O pins in 8-bit parallel. In this condition, when $\overline{PGM}$ pin is set to “L” level, the data is programmed at the specified address, input address, into the built-in PROM.

(3) Erase (Possible only in EPROM version)

The contents of the built-in PROM is erased by exposing the glass window on top of the package to an ultraviolet light which has a wave length of 2537 Angstrom. The light must be 15 J/cm^2 or more.

18.1.3 Programming algorithm of built-in PROM

① Set Vcc = 6 V, VPP = 12.5 V, and address to 14000_{16}.
② After applying a programming pulse of 0.2 ms, check whether data can be read or not.
③ If the data cannot be read, apply a programming pulse of 0.2 ms again.
④ Repeat the procedure, which consists of applying a programming pulse of 0.2 ms and read check, until the data can be read. Additionally, record the number of applied pulses (χ) before the data has been read.
⑤ Apply χ pulse (0.2 × χ ms) (described in ④) as additional programming pulses.
⑥ When this procedure (① to ⑤) is completed, increment the address and repeat the above procedure until the last address is reached.
⑦ After programming to the last address, read data when Vcc = VPP = 5 V (or Vcc = VPP = 5.5 V).

Figure 18.1.2 shows the programming algorithm flowchart.

18.1 EPROM mode

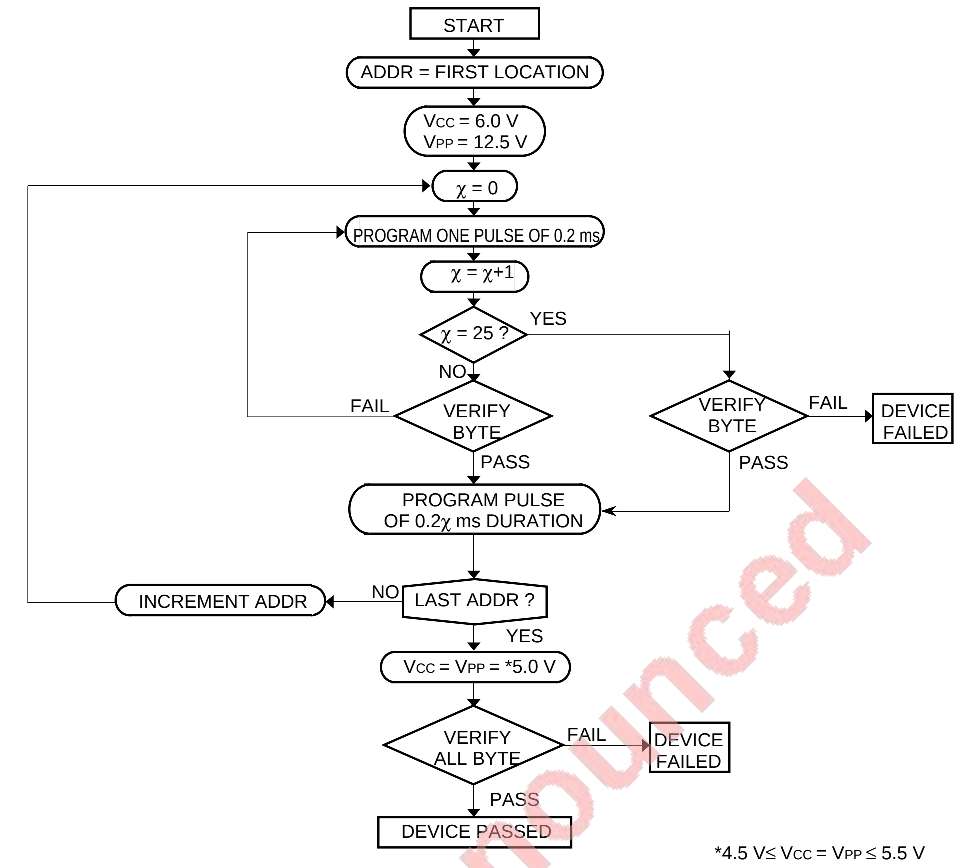


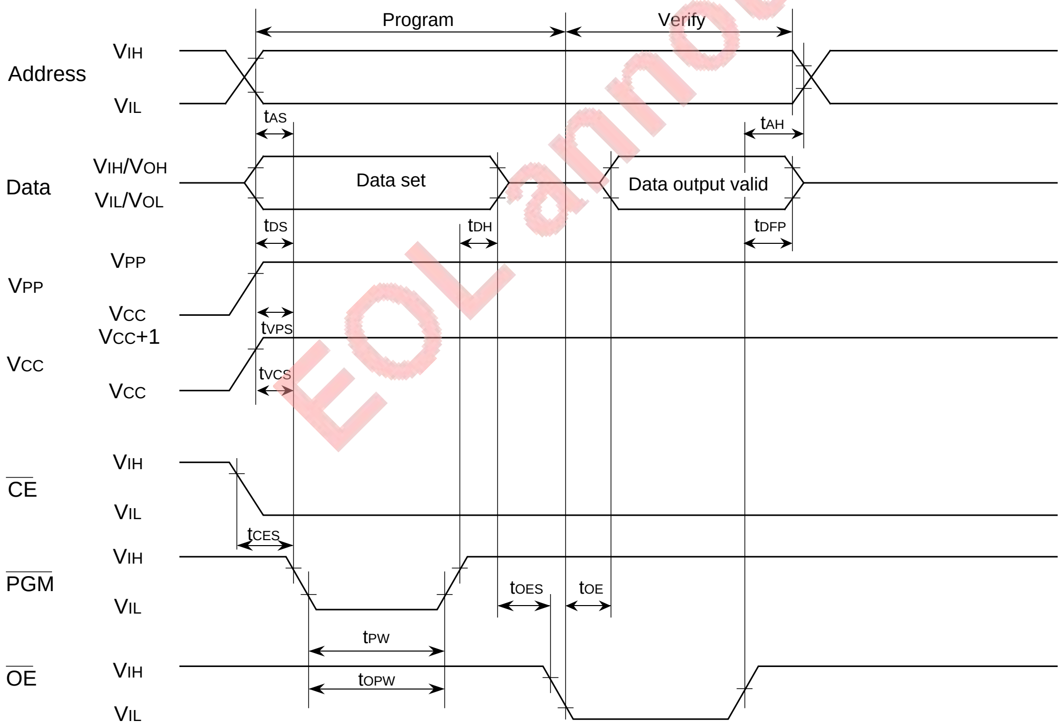
Fig. 18.1.2 Programming algorithm flow chart

18.1.4 Electrical characteristics of programming algorithm

AC electrical characteristics (Ta = 25±5 °C, Vcc = 6±0.25 V, V_{PP} = 12.5±0.3 V, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_{AS}	Address setup time	2			µs
t_{OES}	$\overline{OE}$ setup time	2			µs
t_{DS}	Data setup time	2			µs
t_{AH}	Address hold time	0			µs
t_{DH}	Data hold time	2			µs
t_{DFP}	Output floating delay time after $\overline{OE}$	0		130	ns
t_{VCS}	Vcc setup time	2			µs
t_{VPS}	V_{PP} setup time	2			µs
t_{PW}	$\overline{PGM}$ pulse width	0.19	0.2	0.21	ms
t_{OPW}	Additional $\overline{PGM}$ pulse width	0.19		5.25	ms
t_{CES}	$\overline{CE}$ setup time	2			µs
t_{OE}	Data delay time after $\overline{OE}$			150	ns

Programming timing diagram



Switching characteristics measuring conditions
●Input voltage : V_{IL} = 0.45 V, V_{IH} = 2.4 V
●Input signal rise/fall time (10 % – 90 %) : ≤ 20 ns
●Reference voltage in timing measurement : Input/output “L” = 0.8 V, “H” = 2 V

18.2 Usage precaution

18.2.1 Precautions on all PROM versions

●When programming to the built-in PROM, high voltage is required. Accordingly, be careful not to apply excessive voltage to the microcomputer. Furthermore, be especially careful during power-on.

●Noise gets in easily because the built-in PROM is wired directly from CNV_{SS} (V_{PP}) pin. To prevent noise, the wiring of CNV_{SS} (V_{PP}) pin is performed below. Figure 18.2.1 shows the wiring of CNV_{SS} (V_{PP}) pin.

<In single-chip or memory expansion mode>

Connect CNV_{SS} (V_{PP}) pin to the microcomputer's V_{SS} pin in the shortest possible distance.
If the wiring cannot be shortened, insert a resistor of about 5 kohms as close to CNV_{SS} (V_{PP}) pin as possible. By way of this resistor, connect CNV_{SS} (V_{PP}) pin to V_{SS} pin.

<In microprocessor mode>

Connect CNV_{SS} (V_{PP}) pin to the microcomputer's V_{CC} pin in the shortest possible distance.

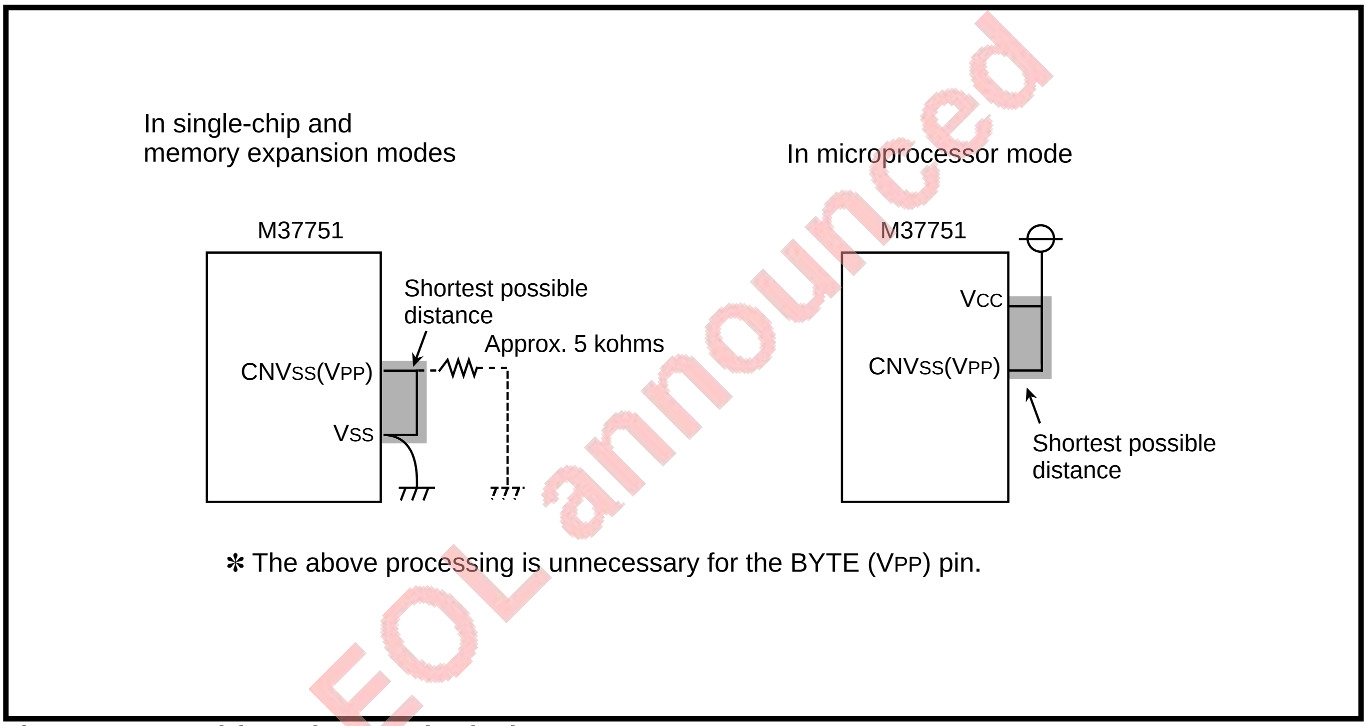
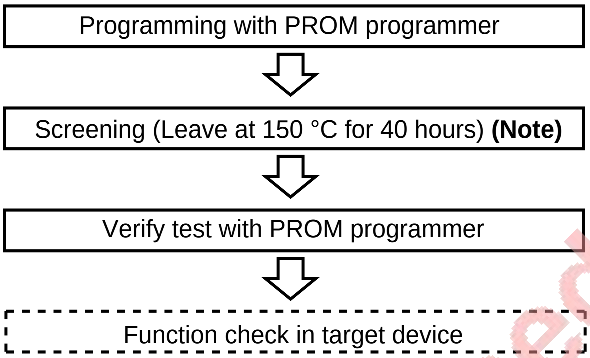


Figure 18.2.1 Wiring of CNV_{SS} (V_{PP}) pin

18.2.2 Precautions on one time PROM version

One time PROM version shipped in a blank (M37751E6CFP), of which built-in PROM is programmed by users, is also provided.

For the microcomputer, a programming test and screening are not performed in the assembly process and the following processes. To improve their reliability after programming, we recommend to program and test as the flow shown in Figure 18.2.2 before use.



Note: Never expose to 150 °C exceeding 100 hours.

Fig. 18.2.2 Programming and test flow for One Time PROM version

18.2.3 Precautions on EPROM version

- Cover the transparent glass window with a shield or others during the read mode because exposing to sun light or fluorescent lamp can cause erasing the programmed data.
  Be careful that the shield does not touch the EPROM lead pins.
  A shield to cover the transparent window is available from Mitsubishi Electric Corporation.
- Clean the transparent glass before erasing. There is a possibility that fingers' fat and paste disturb the passage of ultraviolet rays and affect badly the erasure capability.
- The EPROM version is a tool only for program development, evaluation only, and do not use it for the mass product run.

MEMORANDUM

CHAPTER 19

FLASH MEMORY VERSION

FLASH MEMORY VERSION

In the flash memory version M37751F6CFP, to perform program, read, and erase operations for the built-in flash memory is possible. The M37751F6CFP has the same function as the mask ROM version except for the built–in flash memory **(Note)**.
The M37751F6CFP can select the microcomputer mode, which is performed the same operation as the mask ROM version, or the flash memory mode, which enables to access to the built–in flash memory. When inputting “L” level to the $\overline{RESET}$ pin, the M37751F6CFP enters the flash memory mode. In the flash memory mode, there are two modes: the parallel input/output mode and the serial input/output mode.

Note: Ports $P4_5$ and $P4_6$ peripheral circuits are different from those of mask ROM version.

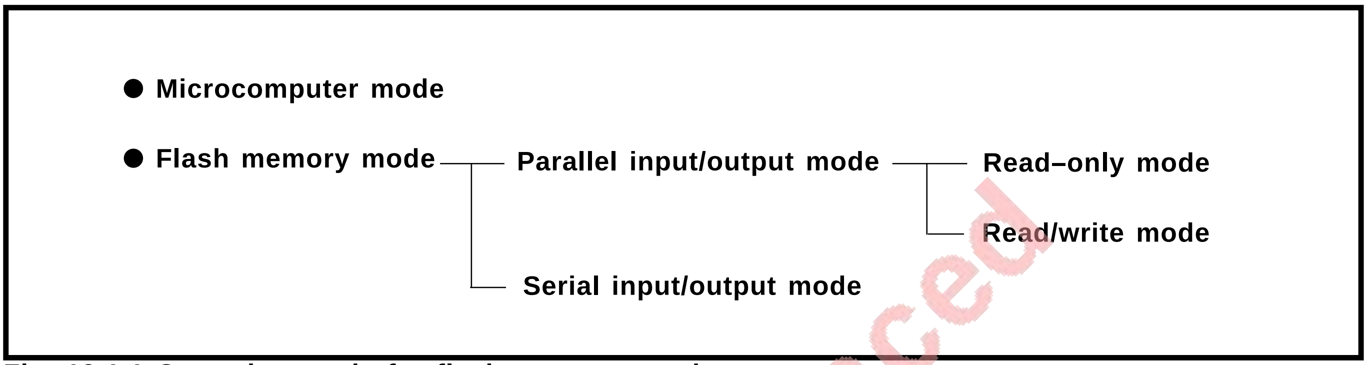


Fig. 19.1.1 Operation mode for flash memory version

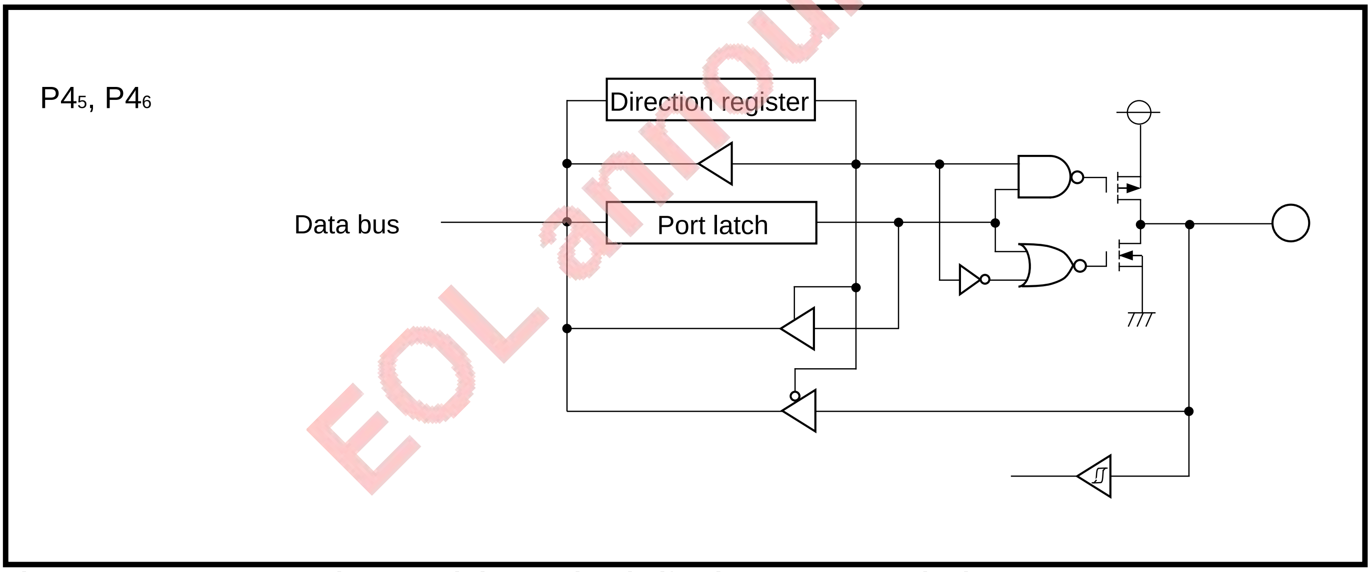


Fig. 19.1.2 Ports $P4_5$ and $P4_6$ peripheral circuit (flash memory version)

19.1 Parallel input/output mode

The built-in flash memory can be accessed by using a general purpose ROM programmer in the parallel I/O mode. In this mode, the read–only mode or the read/write mode (software command control mode) can be selected as the built–in flash memory mode with the voltage applied to the V_{PP} (CNV_{SS}) pin.

FLASH MEMORY VERSION

19.1 Parallel input/output mode

19.1.1 Pin description

Table 19.1.1 lists the pin description in the parallel I/O mode.

Table 19.1.1 Pin description in parallel I/O mode

Pin	Name	Input/Output	Functions
Vcc, Vss	Power supply		Supply 5 V ±10 % to Vcc pin and 0 V to Vss pin.
CNVss	V_{PP} input	Input	**[Read-only mode]** Supply Vcc to Vcc +1.0 V. **[Read/write mode]** Supply 12 V ±5 %.
BYTE	External data bus width select input	Input	Connect to Vss pin.
$\overline{RESET}$	Reset input	Input	Connect to Vss pin.
X_{IN}	Clock input	Input	Connect a ceramic resonator or quartz-crystal oscillator between X_{IN} and X_{OUT} pins. When using an external clock, the clock source must be input to X_{IN} pin and X_{OUT} pin must be left open.
X_{OUT}	Clock output	Output	Connect a ceramic resonator or quartz-crystal oscillator between X_{IN} and X_{OUT} pins. When using an external clock, the clock source must be input to X_{IN} pin and X_{OUT} pin must be left open.
$\overline{E}$	Enable output	Output	Left open.
AVcc	Analog supply input		Connect to Vcc pin.
AVss	Analog supply input		Connect to Vss pin.
V_{REF}	Reference voltage input	Input	Connect to Vss pin.
$P0_0$–$P0_7$	Address input A_0 to A_7	Input	These are address A_0–A_7 input pins.
$P1_0$–$P1_7$	Address input A_8 to A_{15}	Input	These are address A_8–A_{15} input pins.
$P2_0$–$P2_7$	Data input/output D_0 to D_7	Input/Output	These are data D_0–D_7 input/output pins.
$P3_0$–$P3_3$	Input port P3	Input	Connect to Vss pin.
$P4_0$, $P4_1$	Input port P4	Input	Connect to Vss pin.
$P4_2$	Input port P4	Input	Left open.
$P4_3$ to $P4_7$	Input port P4	Input	Connect to Vss pin.
$P5_0$	Control signal input	Input	This is $\overline{WE}$ signal input pin.
$P5_1$	Control signal input	Input	This is $\overline{OE}$ signal input pin.
$P5_2$	Control signal input	Input	This is $\overline{CE}$ signal input pin.
$P5_3$	Input port P5	Input	Connect to Vcc pin.
$P5_4$	Address input A_{16}	Input	This is address A_{16} input pin.
$P5_5$ to $P5_7$	Input port P5	Input	Connect to Vss pin.
$P6_0$ to $P6_7$	Input port P6	Input	Connect to Vss pin.
$P7_0$ to $P7_7$	Input port P7	Input	Connect to Vss pin.
$P8_0$ to $P8_7$	Input port P8	Input	Connect to Vss pin.

19.1.2 Access to built–in flash memory

In the parallel I/O mode, the built–in flash memory can be accessed with the same operation as CMOS flash memory M5M28F101. However, because the built–in flash memory has a capacity of 48 Kbytes, use addresses 04000_{16} to $0FFFF_{16}$ for programming and write "FF_{16}" to addresses 00000_{16} to $03FFF_{16}$ and 10000_{16} to $1FFFF_{16}$. The M37751F6CFP does not contain a facility to read out a device identification code by applying a high voltage to A_9 ($P1_1$) pin. Do not erratically set program conditions etc..
Table 19.1.2 lists the pin correspondence of the M37751F6CFP and the M5M28F101.
Figure 19.1.3 shows the pin connection in the parallel I/O mode.

Table 19.1.2 Pin correspondence of M37751F6CFP and M5M28F101 (parallel I/O mode)

	M37751F6CFP	M5M28F101
Vcc	Vcc	Vcc
V_{PP} input	CNVss	V_{PP}
Vss	Vss	Vss
Address input	P0, P1, $P5_4$	A_0 to A_{16}
Data I/O	P2	D_0 to D_7
$\overline{CE}$ signal input	$P5_2$	$\overline{CE}$
$\overline{OE}$ signal input	$P5_1$	$\overline{OE}$
$\overline{WE}$ signal input	$P5_0$	$\overline{WE}$

19.1 Parallel input/output mode

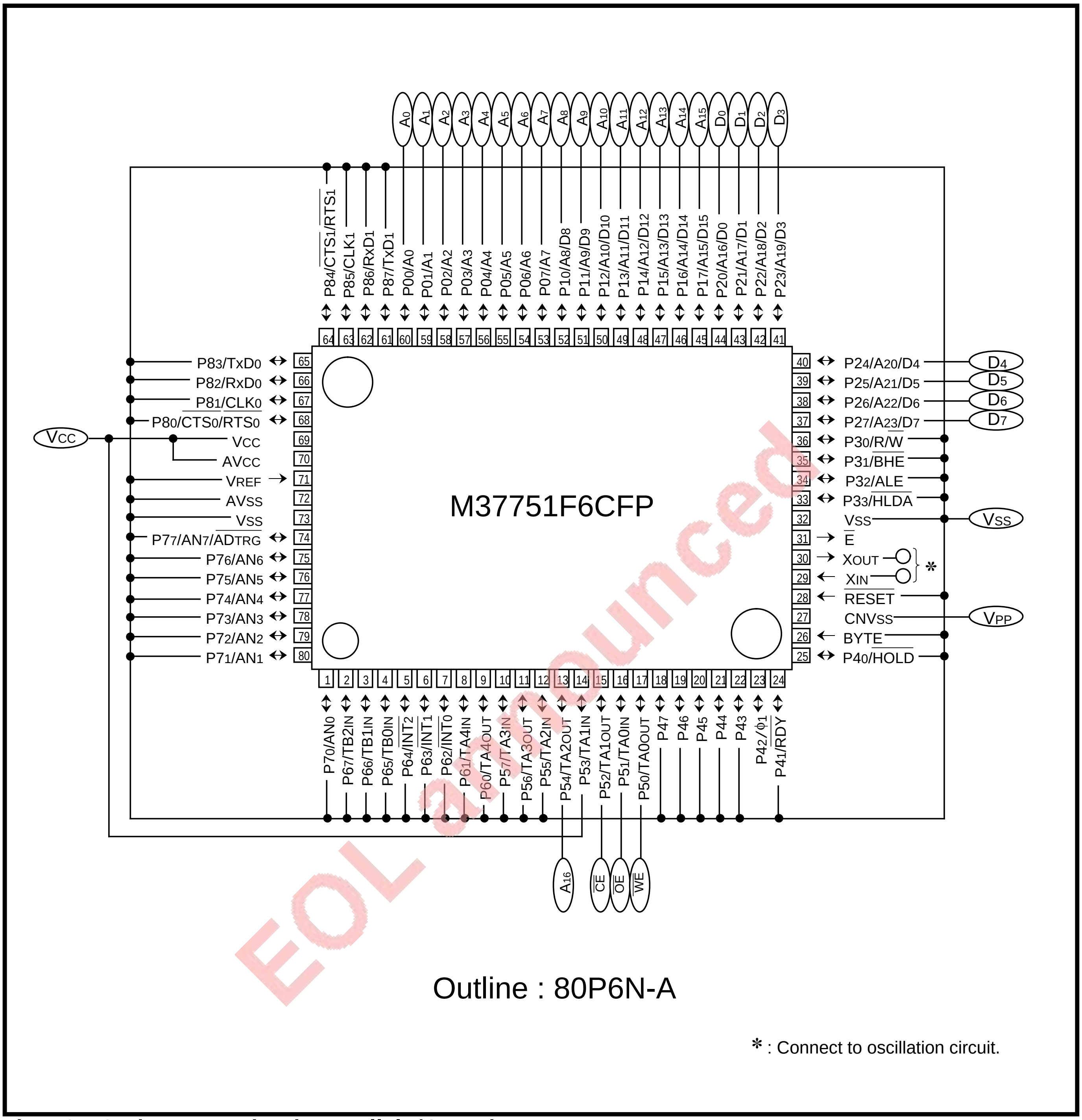


Fig. 19.1.3 Pin connection in parallel I/O mode

19.1.3 Read–only mode

When connecting shown in Figure 19.1.3 and $V_{PP}L$ level is applied to the V_{PP} pin, the built–in flash memory operates at the read–only mode. In the read–only mode, the built–in flash memory becomes read, output disable, or standby state depending on the control signals. In this mode, the contents of the built–in flash memory can be read. Table 19.1.3 lists the states of the built–in flash memory.

Table 19.1.3 States of control signals and built–in flash memory in read–only mode

State \ Pin	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	V_{PP}	Data I/O
Read	V_{IL}	V_{IL}	V_{IH}	$V_{PP}L$	Output
Output disable	V_{IL}	V_{IH}	V_{IH}	$V_{PP}L$	Floating
Standby	V_{IH}	×	×	$V_{PP}L$	Floating

Note: × can be V_{IL} or V_{IH}.

(1) Read

When inputting the address of a memory location to be read and the control signals at the timing shown in Figure 19.1.4, data of the specified address (input address) is output to an external.

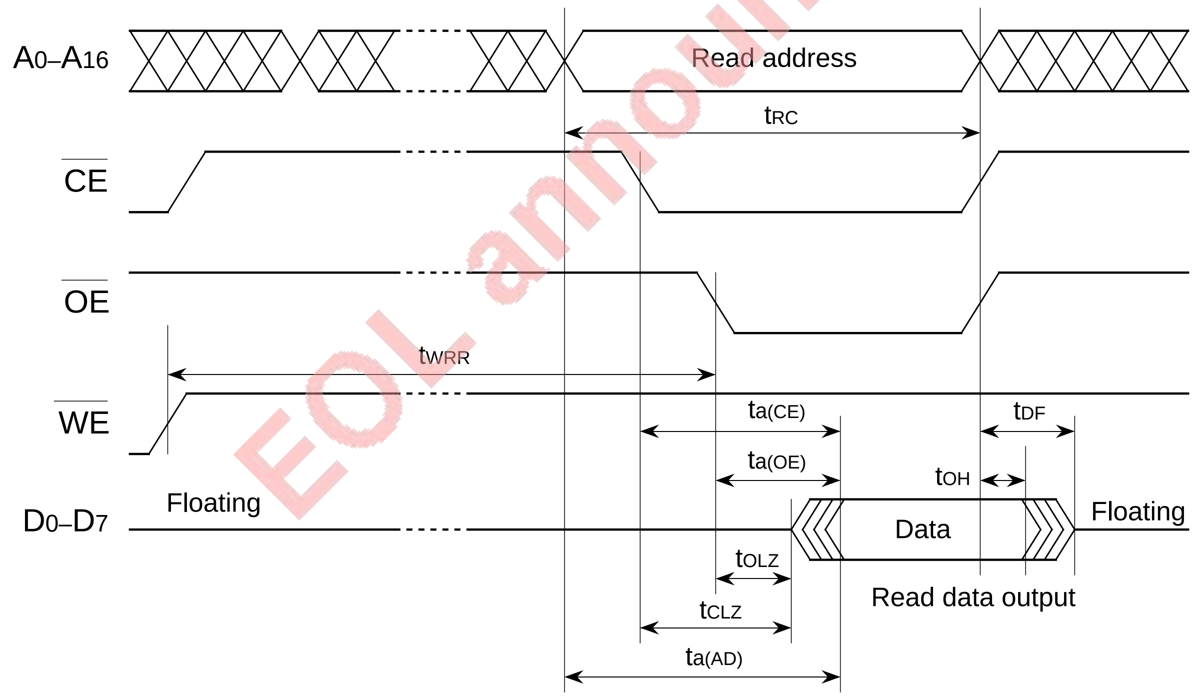


Fig. 19.1.4 Read timing

(2) Output disable
The microcomputer enters the read disable state.

(3) Standby
The microcomputer enters the power–saving state and the supply current decreases.

19.1.4 Read/write (software command control) mode

When connecting shown in Figure 19.1.3 and V_{PP}H level is applied to the V_{PP} pin, the built–in flash memory operates at the read/write mode. In the read/write mode, the built–in flash memory becomes read, output disable, standby or program state depending on the control signals. In this mode, program, read, and erase operations can be performed to the built–in flash memory. Table 19.1.4 lists the states of the built–in flash memory.

Table 19.1.4 States of control signals and built–in flash memory in read/write mode

State \ Pin	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	V_{PP}	Data I/O
Read	V_{IL}	V_{IL}	V_{IH}	V_{PP}H	Output
Output disable	V_{IL}	V_{IH}	V_{IH}	V_{PP}H	Floating
Standby	V_{IH}	×	×	V_{PP}H	Floating
Program	V_{IL}	V_{IH}	V_{IL}	V_{PP}H	Input

Notes 1: × can be V_{IL} or V_{IH}.
2: Refer to **"(5) Software command"** for read and write states.

(1) Read
When executing the read command or program verify command etc., the read mode is used. (Refer to **"(5) Software command."**)

(2) Output disable
The microcomputer enters the read disable state.

(3) Standby
The microcomputer enters the power–saving state and the supply current decreases.

(4) Program
When inputting the command code or program data etc., the program mode is used. (Refer to **"(5) Software command."**)

(5) Software command

In the read/write mode, the built–in flash memory is accessed by input (execution) of the software command.

Table 19.1.5 lists the software command. The software command is executed by data input/output in the first and second cycles. The command code is input to select the operation of the built–in flash memory in the first cycle. The data etc. are input/output in the second cycle.

The following explains each software command.

Table 19.1.5 Software command and input/output information

Software command	First cycle		Second cycle	
	Address input	Data (command code) input	Address input	Data I/O
Read	×	00_{16}	Read address	Read data output
Program	×	40_{15}	Program address	Program data input
Program verify	×	$C0_{16}$	×	Verify data output
Erase	×	20_{16}	×	20_{16} (command code) input
Erase verify	Verify address	$A0_{16}$	×	Verify data output
Reset	×	FF_{16}	×	FF_{16} (command code) input
Device identification	×	90_{16}	ADI	DDI output

Note: × can be V_{IL} or V_{IH}.

ADI (Device identification address) : Manufacture's code 00000_{16}; device code 00001_{16}

DDI (Device identification data) : Manufacture's code $1C_{16}$; device code $D0_{16}$

● **Read command**

Figure 19.1.5 shows the read command execution timing.

The command code is latched into the internal command latch at the rising edge of the $\overline{WE}$ signal by inputting the control signals and the command code "00_{16}" in the first cycle.

The data of the specified address (input address) is output to an external by inputting the address and control signals in the second cycle.

The read command code which is latched into the command latch is retained until any other command code is latched into the command latch. Accordingly, when the second cycle input over again after the read command code is input in the first cycle, the read command is executed over again.

The read command code is latched into the command latch after power–on.

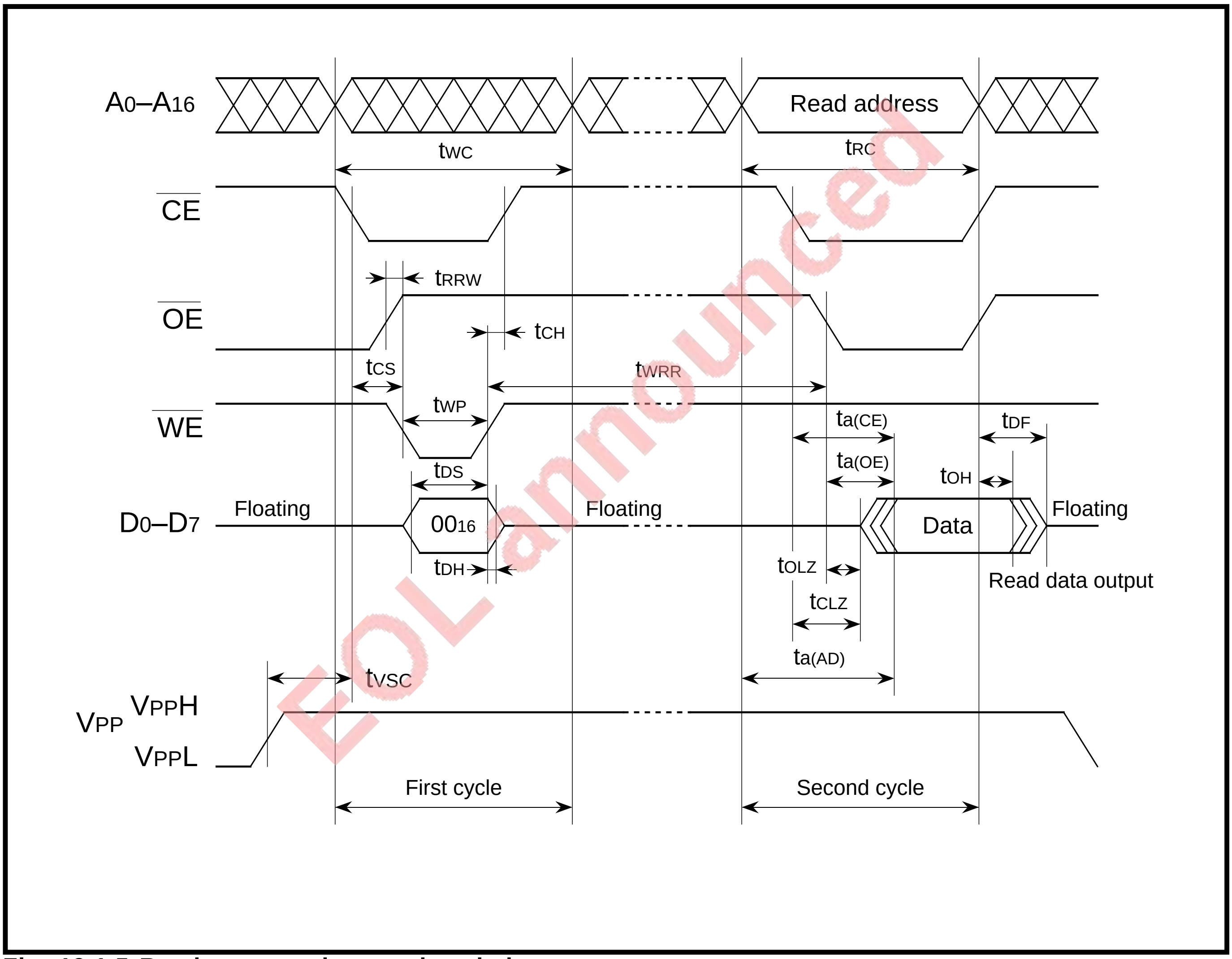


Fig. 19.1.5 Read command execution timing

Note: When executing any command other than the read command, input the command code (input from the first cycle) each time the execution.

● **Program command**

Figure 19.1.6 shows the program command and the program verify command execution timing. The command code is latched into the internal command latch at the rising edge of the $\overline{WE}$ signal by inputting the control signals and the command code "40_{16}" in the first cycle.

The address is latched into the internal at the falling edge of the $\overline{WE}$ signal and the data is latched at the rising edge of the $\overline{WE}$ signal by inputting the address, data, and control signals in the second cycle.

The program is started at the rising edge of the $\overline{WE}$ signal in the second cycle and the input data is programmed to the specified address (input address) within 10 μs as measured by its internal timer. Programming is performed by the byte unit.

Note: Be sure to execute a program verify command after executing the program command. If this verification fails, execute repeatedly the program command and the program verify command until the verification passes. (Refer to **"19.1.6 Program/erase algorithm flow chart."**)

● **Program verify command**

This command is executed to verify the program data after executing the program command. The command code is latched into the internal command latch at the rising edge of the $\overline{WE}$ signal by inputting the control signals and the command code "$C0_{16}$" in the first cycle.

The data of the address where the program command is executed is output to an external by inputting the control signals in the second cycle.

Since the address is internally latched when the program command is executed, there is no need to input it when the program verify command is executed.

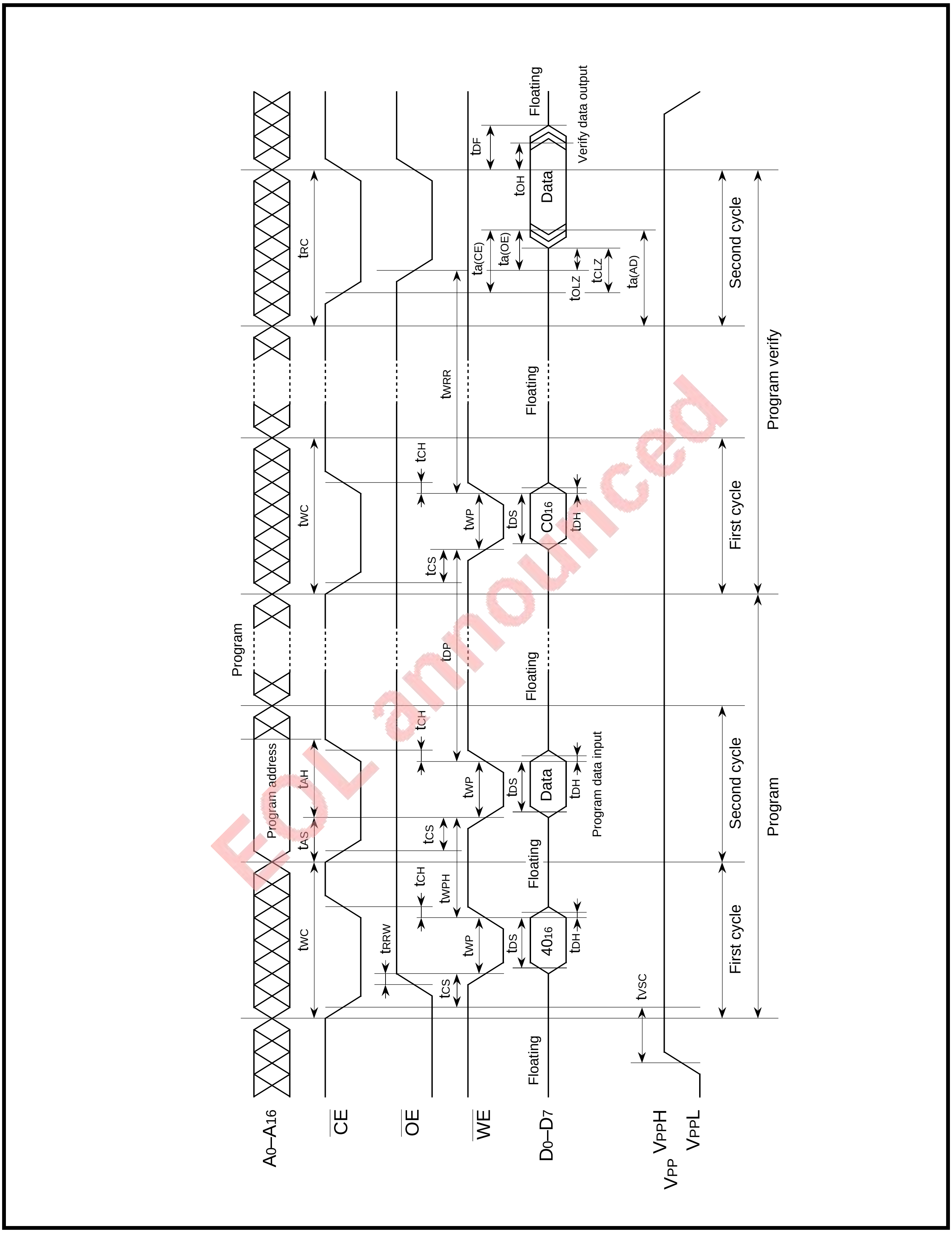


Fig. 19.1.6 Program command and program verify command execution timing

● **Erase command**

Figure 19.1.7 shows the erase command and the erase verify command execution timing.
The command code is latched into the internal command latch at the rising edge of the $\overline{WE}$ signal by inputting the control signals and the command code "20_{16}" in the first cycle.
The command code is latched into the internal command latch again at the rising edge of the $\overline{WE}$ signal by inputting the control signals and the command code "20_{16}" again in the second cycle.
The erase operation is started at the rising edge of the $\overline{WE}$ signal in the second cycle, and the built–in flash memory contents are collectively erased within 9.5 ms as measured by the internal timer.
Write "00_{16}" to all the built–in flash memory area before executing the erase command.

Note: Be sure to execute a erase verify command after executing the erase command. If this verification fails, execute repeatedly the erase command and the erase verify command until the verification passes. (Refer to **"19.1.6 Program/erase algorithm flow chart."**)
When executing again the erase command after executing the erase verify command and the verification fails, there is no need to write "00_{16}" to the built–in flash memory.

● **Erase verify command**

This command is executed to verify whether or not all contents of the built–in flash memory have been erased after executing the erase command.
The address is latched internally at the falling edge of the $\overline{WE}$ signal by inputting the address, the control signals, and the command code "$A0_{16}$" in the first cycle. The command code is latched into the internal command latch at the rising edge of the $\overline{WE}$ signal.
The data of the specified address (input address) is output to an external by inputting the control signals in the second cycle.

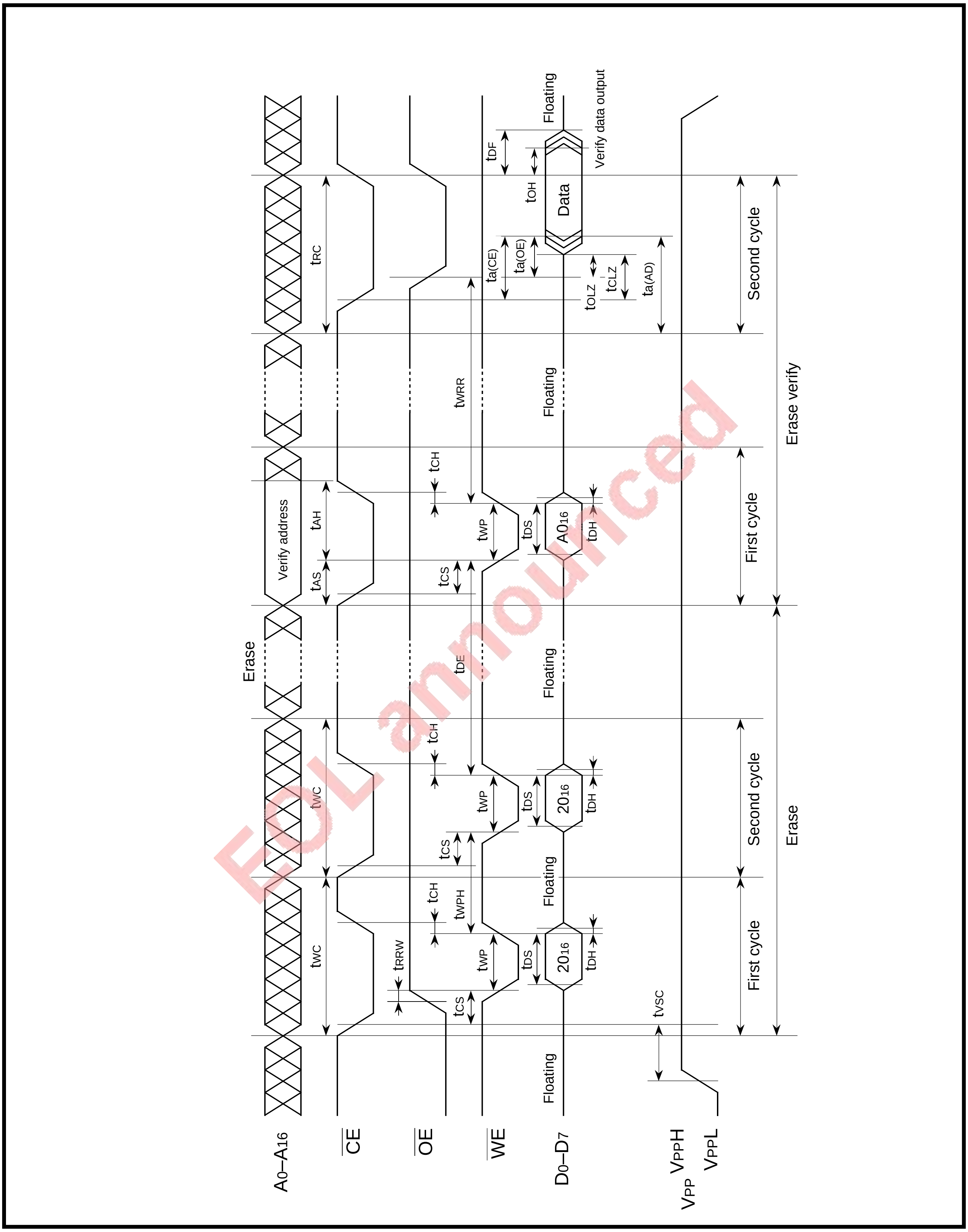


Fig. 19.1.7 Erase command and erase verify command execution timing

19.1 Parallel input/output mode

● **Reset command**

This command is used to stop executing of program or erase safely after inputting the program or erase command code that is, after the command code is latched into the internal command latch in the first cycle.

Figure 19.1.8 shows the reset command execution timing.

When inputting the control signals and the command code "FF_{16}" in the first cycle after the program or erase command code is latched into the command latch, the command code is latched into the internal command latch at the rising edge of the $\overline{WE}$ signal.

When inputting the control signals and command code "FF_{16}" again in the second cycle, the command latch is cleared to "00_{16}" and becomes the state where the read command code is latched. Then, program or erase is not executed. (The contents of the built–in flash memory is not changed.)

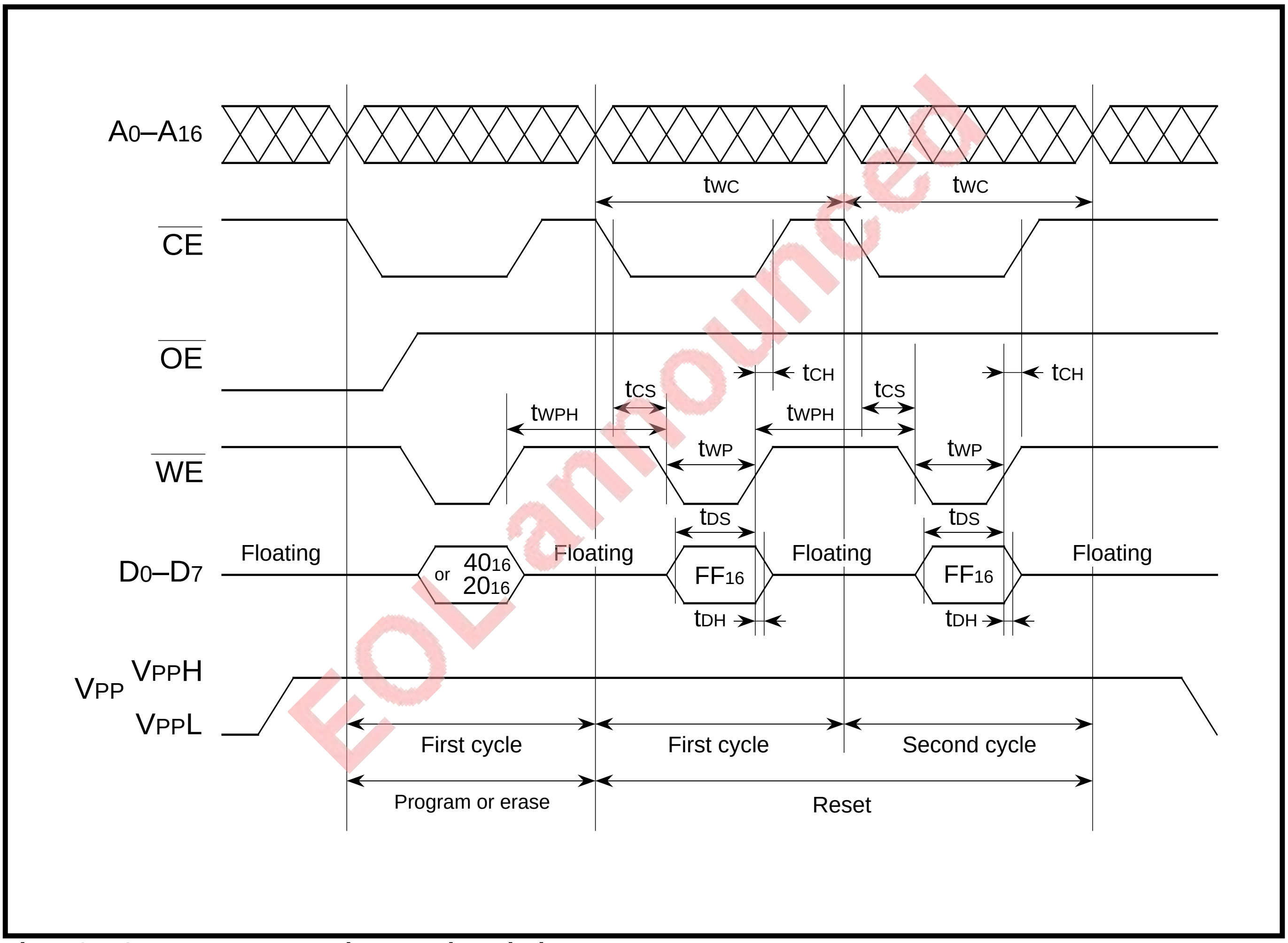


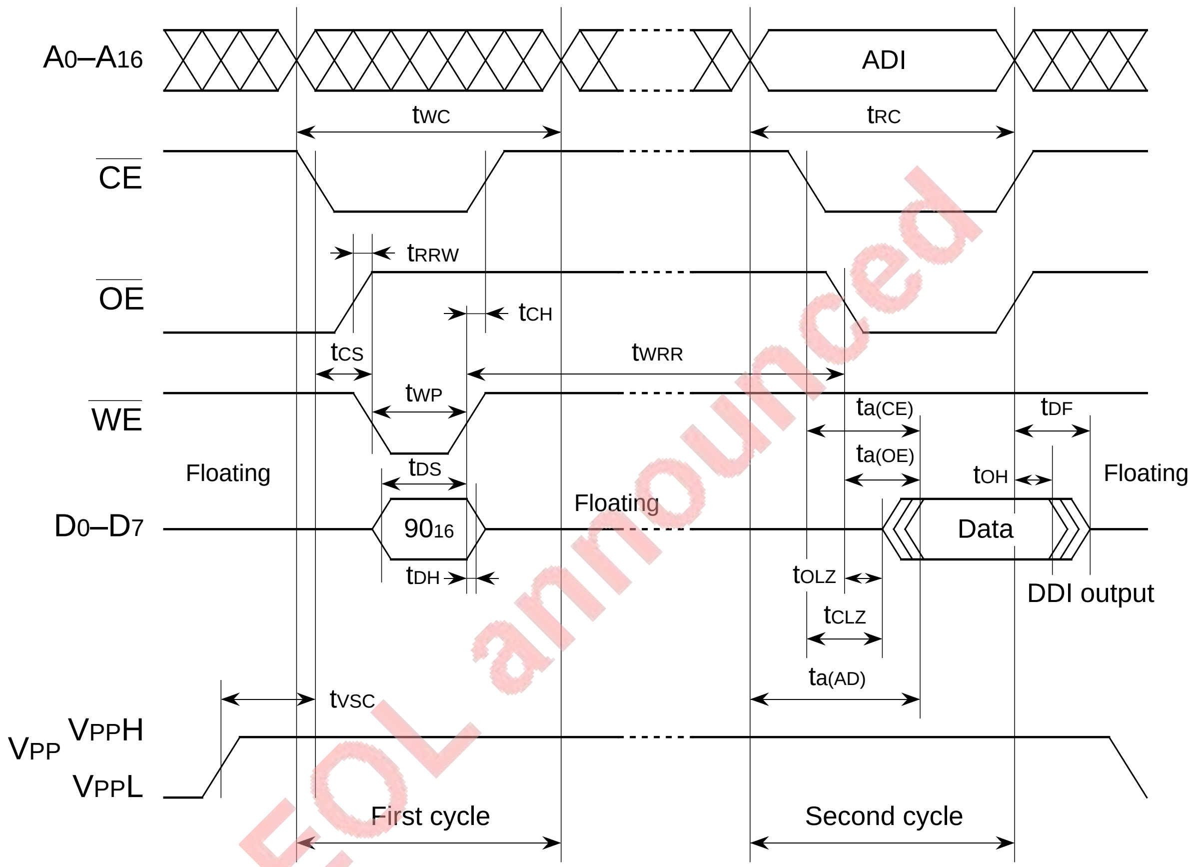
Fig. 19.1.8 Reset command execution timing

● **Device identification command**

Figure 19.1.9 shows the device identification command execution timing.

The command code is latched into the internal command latch at the rising edge of the $\overline{WE}$ signal by inputting the control signals and the command code "90_{16}" in the first cycle.

The manufacture's code "$1C_{16}$" (i.e., MITSUBISHI) is output externally when inputting an address "00000_{16}" and the control signals in the second cycle. The device code "$D0_{16}$" (i.e., 1M–bit flash memory) is output externally when inputting an address "00001_{16}" and the control signals.



ADI (Device identification address) : Manufacturer's code 00000_{16}; device code 00001_{16}
DDI (Device identification data) : Manufacturer's code $1C_{16}$; device code $D0_{16}$

Fig. 19.1.9 Device identification command execution timing

19.1.5 Electrical characteristics

DC electrical characteristics (Ta = 25 °C, V_{CC} = 5 V±10%, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
I_{SB1}	Vcc supply current (at standby)	Vcc = 5.5 V, $\overline{CE}$ = V_{IH}			1	mA
I_{SB2}		Vcc = 5.5 V, $\overline{CE}$ = Vcc±0.2 V			100	µA
I_{CC1}	Vcc supply current (at read)	Vcc = 5.5 V, $\overline{CE}$ = V_{IL}, t_{RC} = 150 ns, Iout = 0 mA			30	mA
I_{CC2}	Vcc supply current (at program)	V_{PP} = V_{PP}H			30	mA
I_{CC3}	Vcc supply current (at erase)	V_{PP} = V_{PP}H			30	mA
I_{PP1}	V_{PP} supply current (at read)	0 ≤ V_{PP} ≤ Vcc+1.0 V			10	µA
		V_{PP} = V_{PP}H			100	µA
		V_{PP} = V_{PP}H			100	µA
I_{PP2}	V_{pp} supply current (at program)	V_{PP} = V_{PP}H			30	mA
I_{PP3}	V_{pp} supply current (at erase)				30	mA
V_{PP}L	V_{pp} supply voltage (read-only mode)		Vcc		Vcc+1.0	V
V_{PP}H	V_{pp} supply voltage (read/write mode)		11.4	12.0	12.6	V

Note: V_{IH}/V_{IL}, V_{OH}/V_{OL}, and I_{IH}/I_{IL} for the control input, address input, and data input/output pins conform to standards for microcomputer modes (memory expansion and microprocessor modes).

AC electrical characteristics (Ta = 25 °C, V_{CC} = 5 V±10%, unless otherwise noted)

Read–only mode

Symbol	Parameter	Limits Min.	Limits Max.	Unit
t_{RC}	Read cycle time	150		ns
$t_{a(AD)}$	Address access time		150	ns
$t_{a(CE)}$	$\overline{CE}$ access time		150	ns
$t_{a(OE)}$	$\overline{OE}$ access time		55	ns
t_{CLZ}	Output enable time (after $\overline{CE}$)	0		ns
t_{OLZ}	Output enable time (after $\overline{OE}$)	0		ns
t_{DF}	Output floating time (after $\overline{OE}$)		35	ns
t_{OH}	Output efficiency time (after $\overline{CE}$, $\overline{OE}$, address)	0		ns
t_{WRR}	Write recovery time (before read)	6		µs

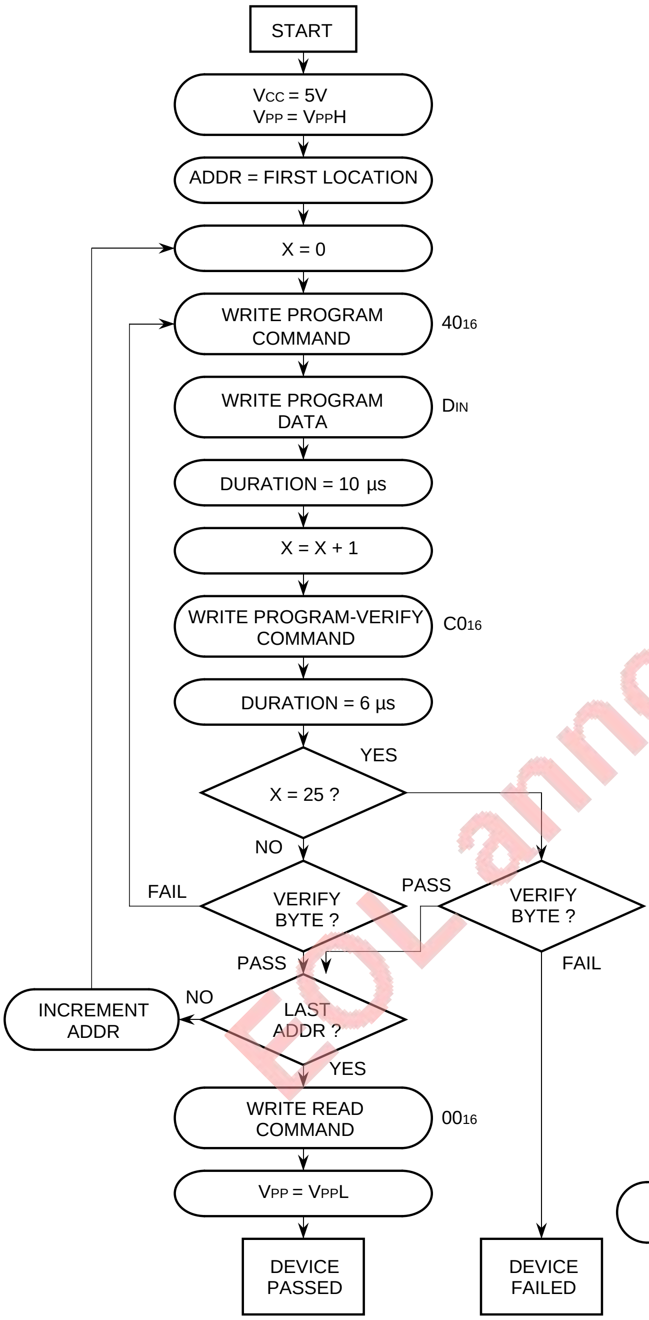
Read/write mode

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_{WC}	Write cycle time	150		ns
t_{AS}	Address setup time	0		ns
t_{AH}	Address hold time	60		ns
t_{DS}	Data setup time	50		ns
t_{DH}	Data hold time	10		ns
t_{WRR}	Write recovery time (before read)	6		µs
t_{RRW}	Read recovery time (before write)	0		µs
t_{CS}	$\overline{CE}$ setup time	20		ns
t_{CH}	$\overline{CE}$ hold time	0		ns
t_{WP}	Write pulse time	60		ns
t_{WPH}	Write pulse waiting time	20		ns
t_{DP}	Program time	10		µs
t_{DE}	Erase time	9.5		ms
t_{VSC}	V_{PP} setup time	1		µs

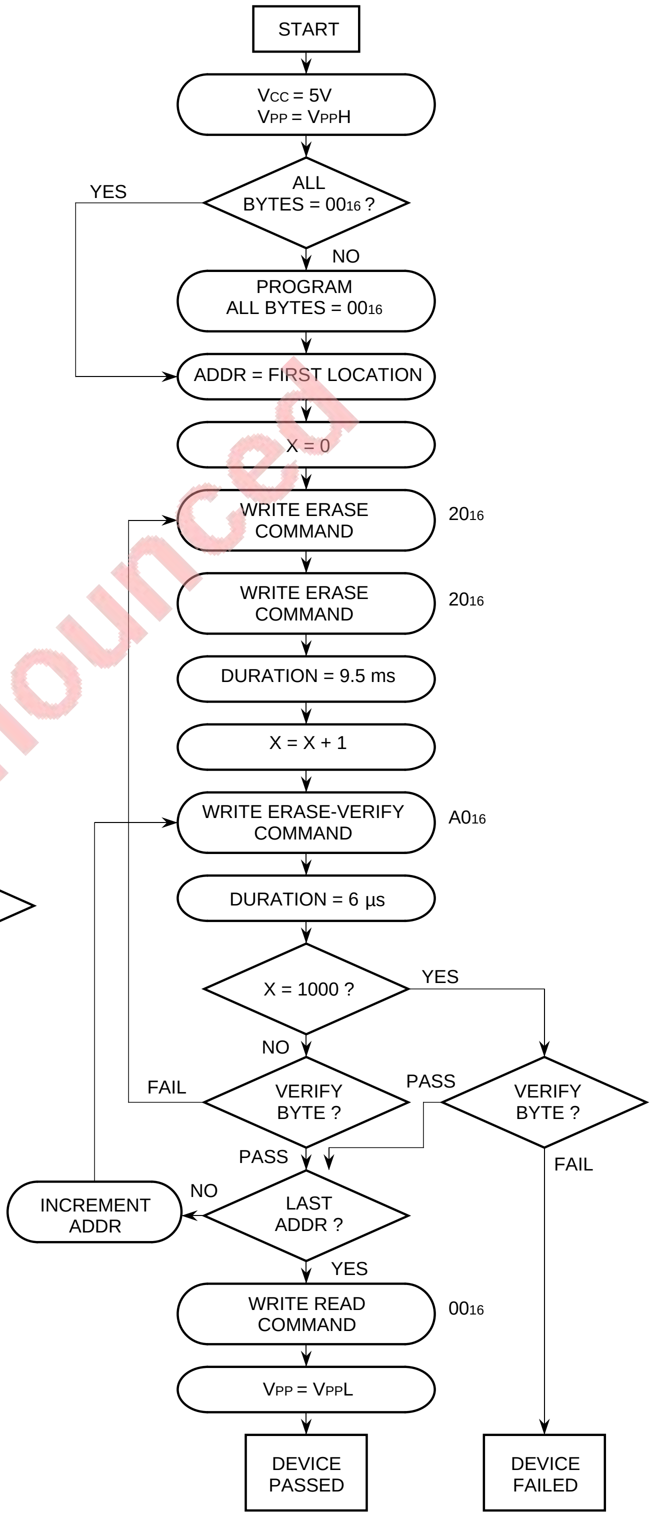
Note: The read timing is same as the read only mode.

19.1.6 Program/erase algorithm flow chart

Program



Erase



19.2 Serial input/output mode

In the serial I/O mode, the contents of the built–in flash memory can be reprogrammed with the state mounting the microcomputer on the board.

19.2.1 Pin description

Table 19.2.1 lists the pin description in the serial I/O mode.

19.2 Serial input/output mode

Table 19.2.1 Pin description in serial I/O mode

Pin	Name	Input/Output	Functions
Vcc, Vss	Power supply		Supply 5 V ±10 % to Vcc pin and 0 V to Vss pin.
CNVss	V_{PP} input	Input	Supply 12 V ±5 %.
BYTE	External data bus width select input	Input	Connect to Vss pin or Vcc pin.
$\overline{RESET}$	Reset input	Input	Connect to Vss pin.
X_{IN}	Clock input	Input	Connect a ceramic resonator or quartz-crystal oscillator between X_{IN} and X_{OUT} pins. When using an external clock, the clock source must be input to X_{IN} pin and X_{OUT} pin must be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	"H" level is output.
AVcc	Analog supply input		Connect to Vcc pin.
AVss			Connect to Vss pin.
V_{REF}	Reference voltage input	Input	Input level between Vss and Vcc or open.
P0$_0$–P0$_7$	Input port P0	Input	Input "H" or "L" level, or open.
P1$_0$–P1$_7$	Input port P1	Input	
P2$_0$–P2$_7$	Input port P2	Input	
P3$_0$–P3$_3$	Input port P3	Input	
P4$_0$	Input port P4	Input	Input "H" or "L" level, or open.
P4$_1$			
P4$_2$			Clock ϕ_1 is output.
P4$_3$			Input "H" or "L" level, or open.
P4$_4$	BUSY output	Output	This pin is BUSY signal output.
P4$_5$	SDA I/O	I/O	This pin is serial data I/O.
P4$_6$	SCLK input	Input	This pin is serial clock input.
P4$_7$	Input port P4		Input "H" or "L" level, or open.
P5$_0$	Input port P5	Input	Input "H" or "L" level, or open.
P5$_1$	Control signal input		This pin is $\overline{OE}$ signal input.
P5$_2$ to P5$_7$	Input port P5		Input "H" or "L" level, or open.
P6$_0$ to P6$_7$	Input port P6	Input	Input "H" or "L" level, or open.
P7$_0$ to P7$_7$	Input port P7	Input	
P8$_0$ to P8$_7$	Input port P8	Input	

19.2.2 Access to built–in flash memory

Figure 19.2.1 shows the pin connection in the serial I/O mode.
When inputting "H" level to the SDA ($P4_5$), SCLK ($P4_6$), and $\overline{OE}$ signal input ($P5_1$) pins, and after that, applying the $V_{PP}H$ level to the V_{PP} (CNV_{SS}), the built–in flash memory operates in the serial I/O mode. The software command, address, and data required for operation of the built–in flash memory are input/output by the clock synchronous serial transfer in this mode. The software command, address, and program data are taken from SDA pin to the inside synchronously with the rising edge of the serial clock inputting to SCLK pin. The read data, verify data, and error code are externally output from SDA pin synchronously with the falling edge of the serial clock. The transfer is performed at 8–bit length and LSB first.
In the serial I/O mode, the built–in flash memory is accessed by inputting (execution) of the software command. Table 19.2.2 lists the software command. To execute the software command requires twice or four times of the transfer. In the first transfer, the command code is input for selecting the built–in flash memory's operation. In the second to fourth transfer, address and data etc. are input/output.
Each software command is described below.
As the capacity of the built-in flash memory is 48 Kbytes, specify addresses 4000_{16} to $FFFF_{16}$. If the addresses except addresses 4000_{16} to $FFFF_{16}$ are specified, the error occurs.

Table 19.2.2 Software command and input/output information

Software command	First transfer (command code input)	Second transfer	Third transfer	Forth transfer
Read	00_{16}	Low-order 8 bits of read address input	High-order 8 bits of read address input	Read data output
Program	40_{16}	Low-order 8 bits of program address input	High-order 8 bits of program address input	Program data input
Program verify	$C0_{16}$	Verify data output	———	———
Auto erase	30_{16}	30_{16} (command code) input	———	———
Error check	80_{16}	Error code output	———	———

19.2 Serial input/output mode

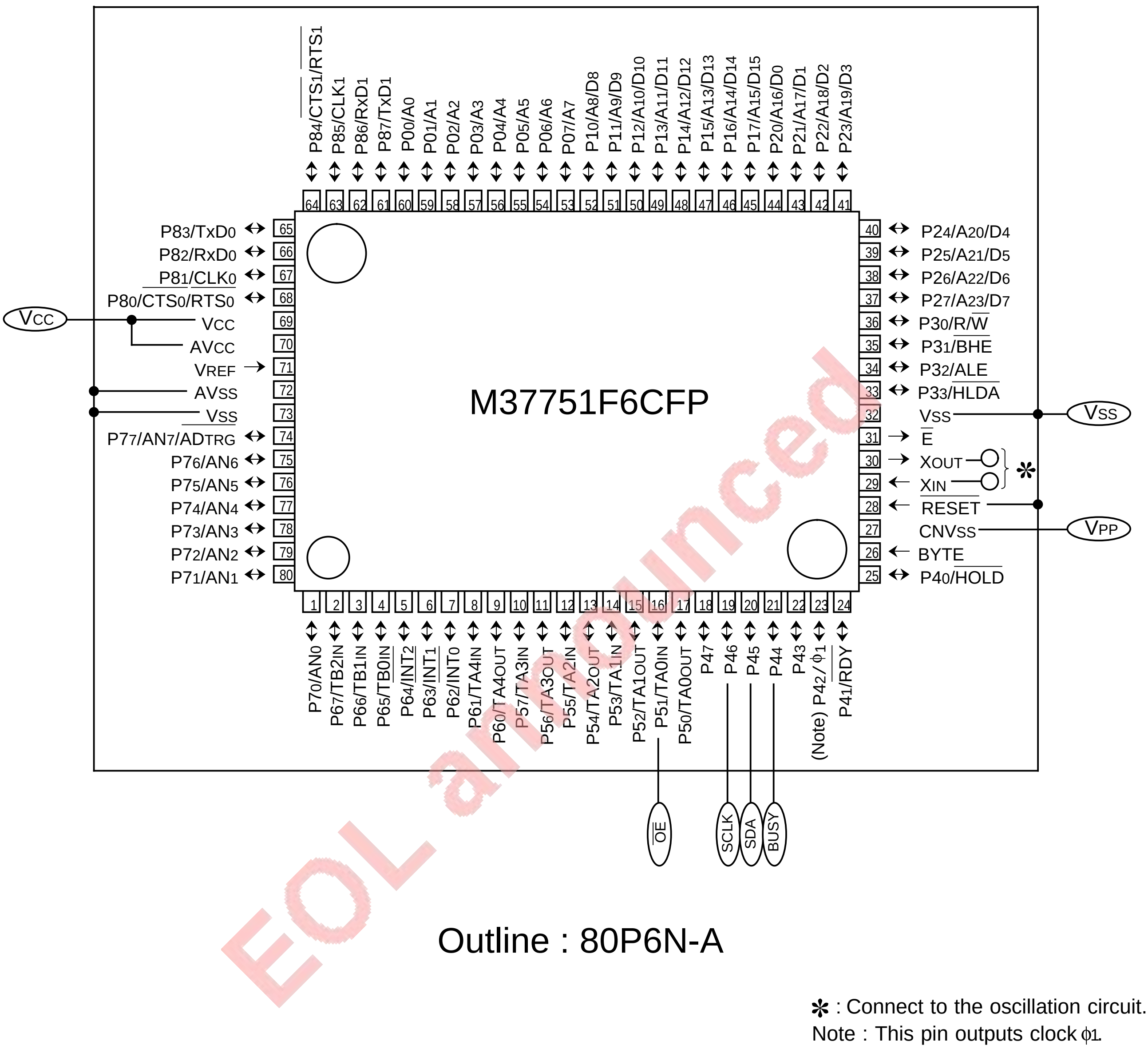


Fig. 19.2.1 Pin connection in serial I/O mode

● **Read command**

Figure 19.2.2 shows the read command execution timing.
The command code "00_{16}" is input at the first transfer.
The low–order 8 bits and the high–order 8 bits are input at the second and third transfer.
When setting "L" level to the $\overline{OE}$ signal, the data of the specified address (input address) is read out and latched up to the internal data latch.
When returning "H" level to the $\overline{OE}$ signal and inputting the serial clock, the data which is latched up to the data latch is output externally.

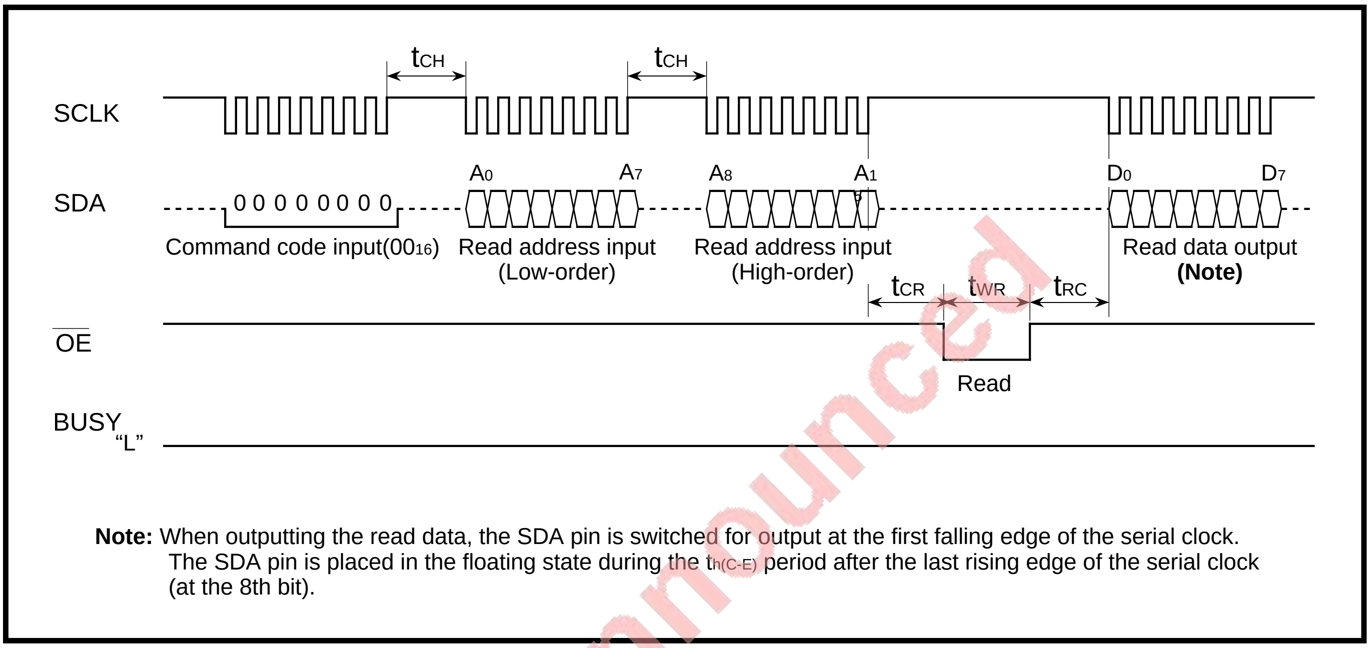


Fig. 19.2.2 Read command execution timing

● **Program command**

Figure 19.2.3 shows the program command execution timing.

The command code “40_{16}” is input at the first transfer.

The low–order 8 bits and the high–order 8 bits of the address are input at the second and third transfer.

The data is input at the forth transfer.

Programming is started at the last rising edge of the forth transfer serial clock and the BUSY signal becomes “H” level. The input data is programmed to the specified address (input address) within 10 µs as measured by the built–in timer and the BUSY signal becomes “L” level. Programming is performed by the byte unit.

Note: Be sure to execute a program verify command after executing the program command. If this verification fails, execute repeatedly the program and program verify commands until the verification passes. (Refer to **“19.2.4 Program algorithm flow chart.”**)

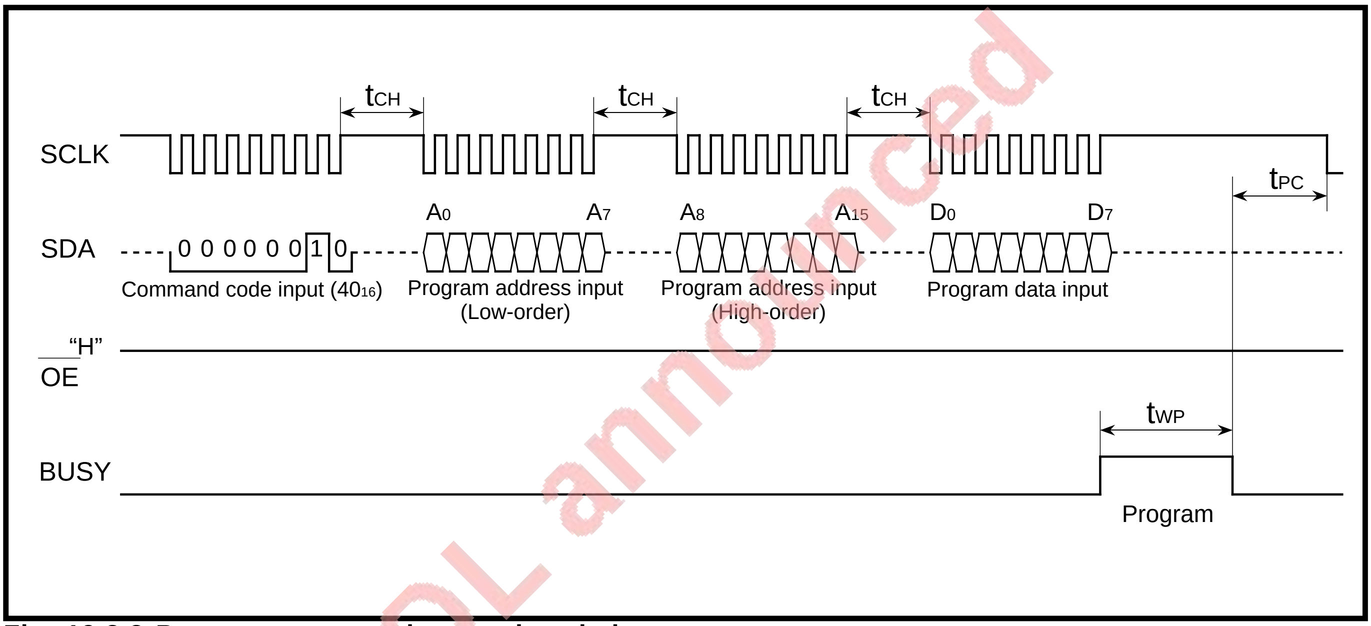


Fig. 19.2.3 Program command execution timing

● **Program verify command**

Figure 19.2.4 shows the program verify command execution timing.

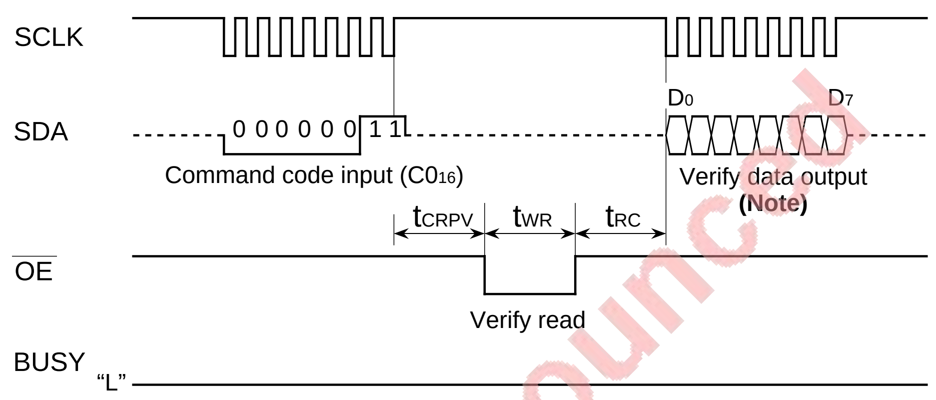
This command is executed to verify data of address where the program command has been executed after executing the program command.

The command code "$C0_{16}$" is input at the first transfer.

When setting the $\overline{OE}$ signal to "L" level, data of address where the program command has been executed is read out and latched to the internal data latch.

When returning the $\overline{OE}$ signal to "H" level and inputting the serial clock, the data which is latched to the data latch is output externally.

Since the address is internally latched when the program command is executed, there is no need to input it when the program verify command is executed.



Note: When outputting the verify data, the SDA pin is switched for output at the first falling edge of the serial clock. The SDA pin is switched in the floating state during the $t_{h(C-E)}$ period after the last rising edge of the serial clock (at the 8th bit).

Fig. 19.2.4 Program verify command execution timing

● **Auto erase command**

Figure 19.2.5 shows the auto erase command execution timing.
The command code "30_{16}" is input at the first transfer.
The command code "30_{16}" is input again at the second transfer.
Erasing is started at the last rising edge of the second transfer serial clock and the BUSY signal becomes "H" level. The BUSY signal becomes "L" by erasing all the contents of the built–in flash memory.

Note: When executing the auto erase command once, "erase → erase verify" is performed repeatedly internally and automatically after programming "00_{16}" to all memory area until erasing all the contents of the built–in flash memory.
Accordingly, erasing is completed by executing the auto erase command once.

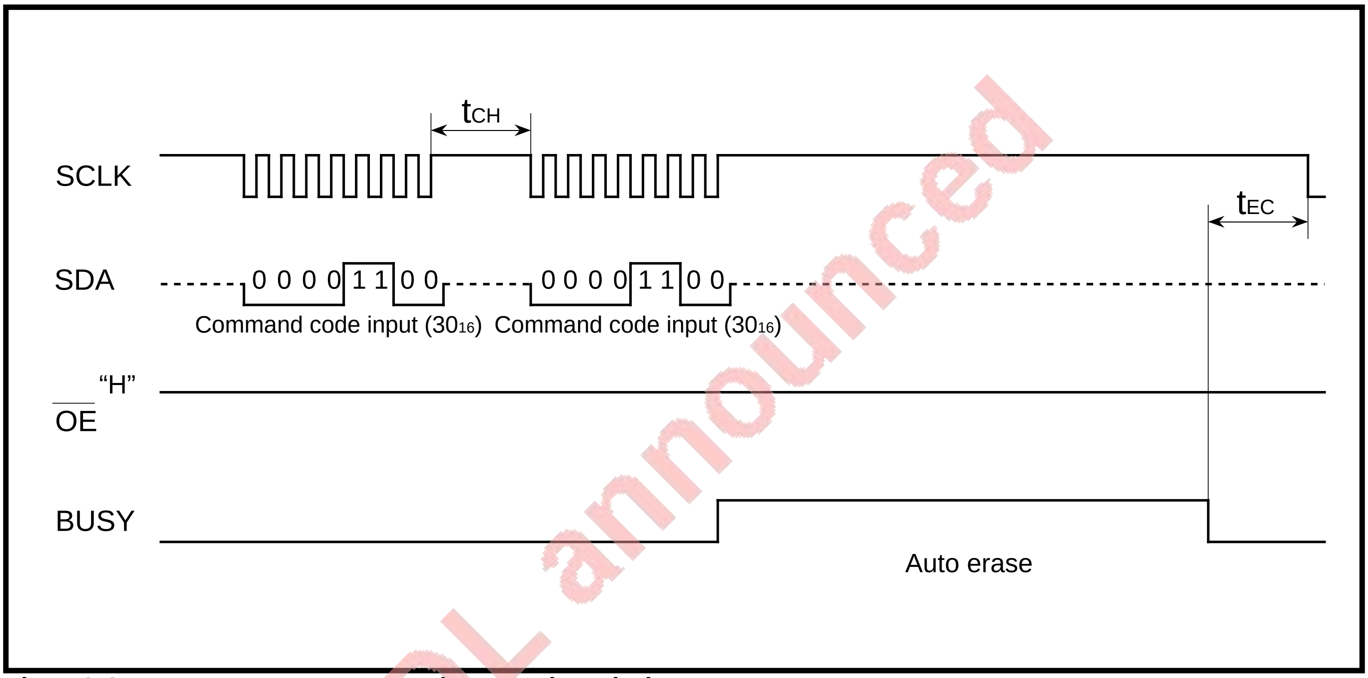
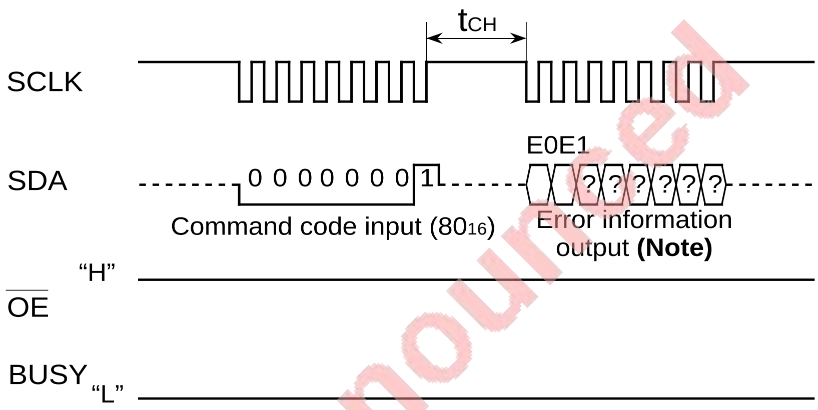


Fig. 19.2.5 Auto erase command execution timing

● **Error check command**

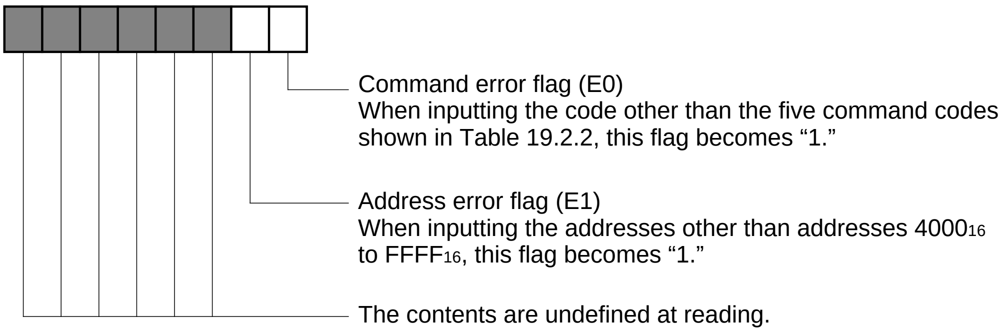
Figure 19.2.6 shows the error check command execution timing.
The command code "80_{16}" is input at the first transfer.
When inputting the serial clock, the error information is output externally.
When an error occurs, the serial communication circuit sets the corresponding error flag to "1" and stops operating, and the serial clock and data are not accepted (even including an error check command). Accordingly, apply the $V_{PP}L$ level to the V_{PP} pin to clear the serial I/O mode and then apply the $V_{PP}H$ level again to select the serial I/O mode and initialize the serial communication circuit. The error information is output when first executing the error check command after initializing.
Figure 19.2.7 shows the error information.
The error flag becomes "0" by executing the error check command. Be sure to execute the error check command because the error flag is undefined after power–on.



Note: When outputting the error information, the SDA pin is switched for output at the first falling edge of the serial clock. The SDA pin is switched in the floating state during the $t_{h(C-E)}$ period after the last rising edge of the serial clock (at the 8th bit).

Fig. 19.2.6 Error check command execution timing

19.2 Serial input/output mode



✻ The error flags are undefined after power-on.
✻ When executing the error check command, the error flags become “0.”

Fig. 19.2.7 Error information

19.2.3 Electrical characteristics

DC electrical characteristics (Ta = 25 °C, V_{CC} = 5 V±10%, V_{PP} = 12 V±5%, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
I_{CC1}	Vcc supply current (at read)	Vcc = 5.5 V, t_{WR} = 320 ns, I_{out} = 0 mA			30	mA
I_{CC2}	Vcc supply current (at program)				30	mA
I_{CC3}	Vcc supply current (at erase)				30	mA
I_{PP1}	V_{PP} supply current (at read)				100	µA
I_{PP2}	V_{PP} supply current (at program)				30	mA
I_{PP3}	V_{PP} supply current (at erase)				30	mA
$V_{PP}H$	V_{PP} supply voltage (at serial I/O mode)		11.4	12.0	12.6	V

Note: V_{IH}/V_{IL}, V_{OH}/V_{OL}, and I_{IH}/I_{IL} for the control signal input, BUSY output, SDA I/O, and SCLK input pins conform to standards for microcomputer modes.

AC electrical characteristics (Ta = 25 °C, V_{CC} = 5 V±10%, V_{PP} = 12 V±5%, f(X_{IN}) = 40 MHz, unless otherwise noted)

Symbol	Parameter	Limits Min.	Limits Max.	Unit
t_{CH}	Serial transmission interval time	400 **(Note 1)**		ns
t_{CR}	Read waiting time after transmission	400 **(Note 1)**		ns
t_{WR}	Read pulse width	320 **(Note 2)**		ns
t_{RC}	Transfer waiting time after read	400 **(Note 1)**		ns
t_{CRPV}	Waiting time before program verify	6		µs
t_{WP}	Programming time		10	µs
t_{PC}	Transfer waiting time after programming	400 **(Note 1)**		ns
t_{EC}	Transfer waiting time after erase	400 **(Note 1)**		ns
$t_{C(CK)}$	SCLK input cycle time	250		ns
$t_{W(CKH)}$	SCLK “H” pulse width	100		ns
$t_{W(CKL)}$	SCLK “L” pulse width	100		ns
$t_{r(CK)}$	SCLK rise time	20		ns
$t_{f(CK)}$	SCLK fall time	20		ns
$t_{d(C-Q)}$	SDA output delay time	0	90	ns
$t_{h(C-Q)}$	SDA output hold time	0		ns
$t_{h(C-E)}$	SDA output hold time (only the 8th bit)	120 **(Note 3)**	200 **(Note 4)**	ns
$t_{su(D-C)}$	SDA input setup time	30		ns
$t_{h(C-D)}$	SDA input hold time	90		ns

Notes 1: When f(X_{IN}) = 25 MHz or less, calculate the minimum value as the following formula 1.

Formula 1 : $\frac{1 \times 10}{f(X_{IN})} \times 10^9$

2: When f(X_{IN}) = 25 MHz or less, calculate the minimum value as the following formula 2.

Formula 2 : $\frac{1 \times 8}{f(X_{IN})} \times 10^9$

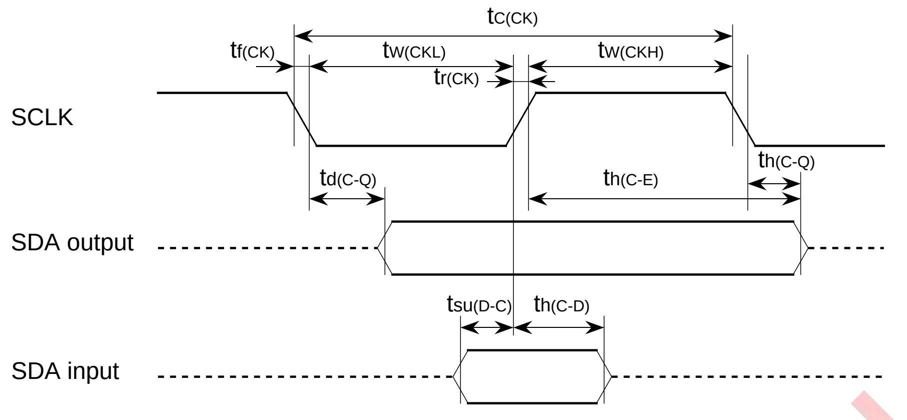
3: When f(X_{IN}) = 25 MHz or less, calculate the minimum value as the following formula 3.

Formula 3 : $\frac{1 \times 3}{f(X_{IN})} \times 10^9$

4: When f(X_{IN}) = 25 MHz or less, calculate the maximum value as the following formula 4.

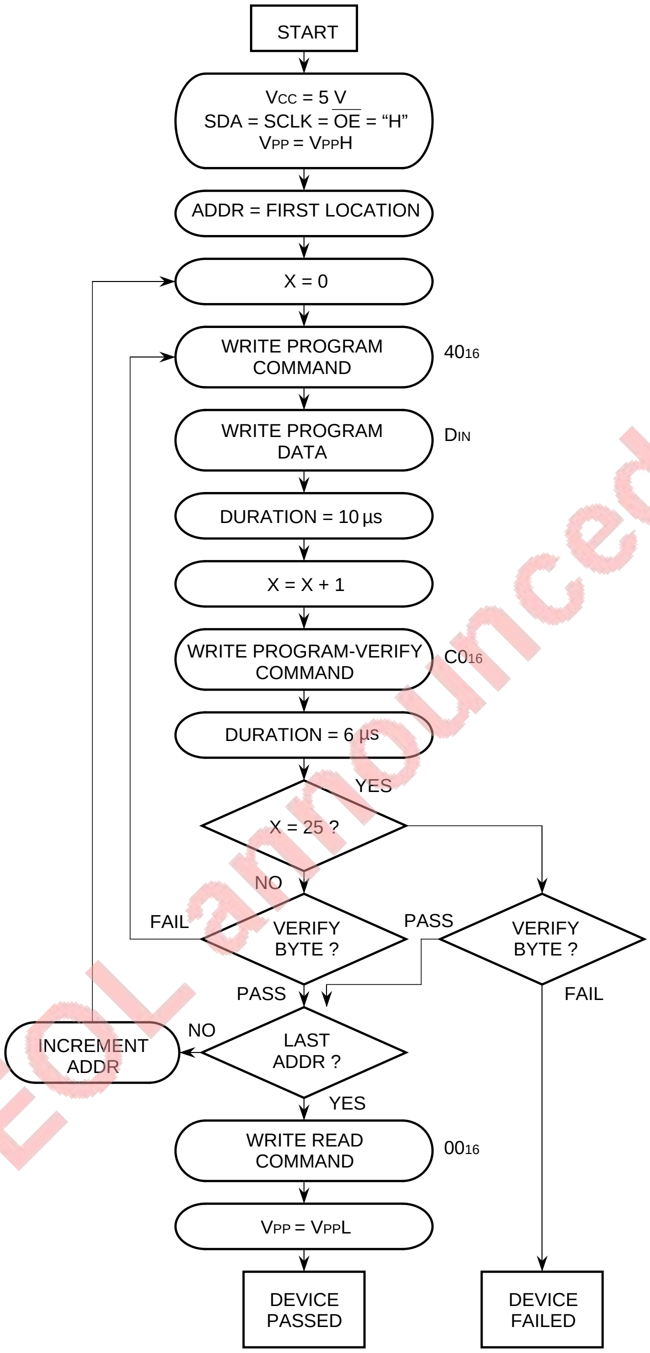
Formula 4 : $\frac{1 \times 5}{f(X_{IN})} \times 10^9$

Timing



Test conditions
•Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
•Input timing voltage : V_{IL} = 0.2 V_{CC}, V_{IH} = 0.8 V_{CC}

19.2.4 Program algorithm flow chart

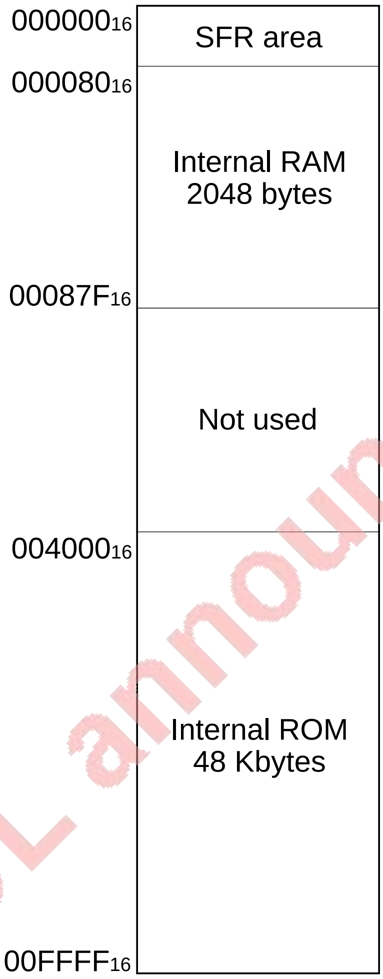


MEMORANDUM

APPENDIX

Appendix 1. Memory assignment

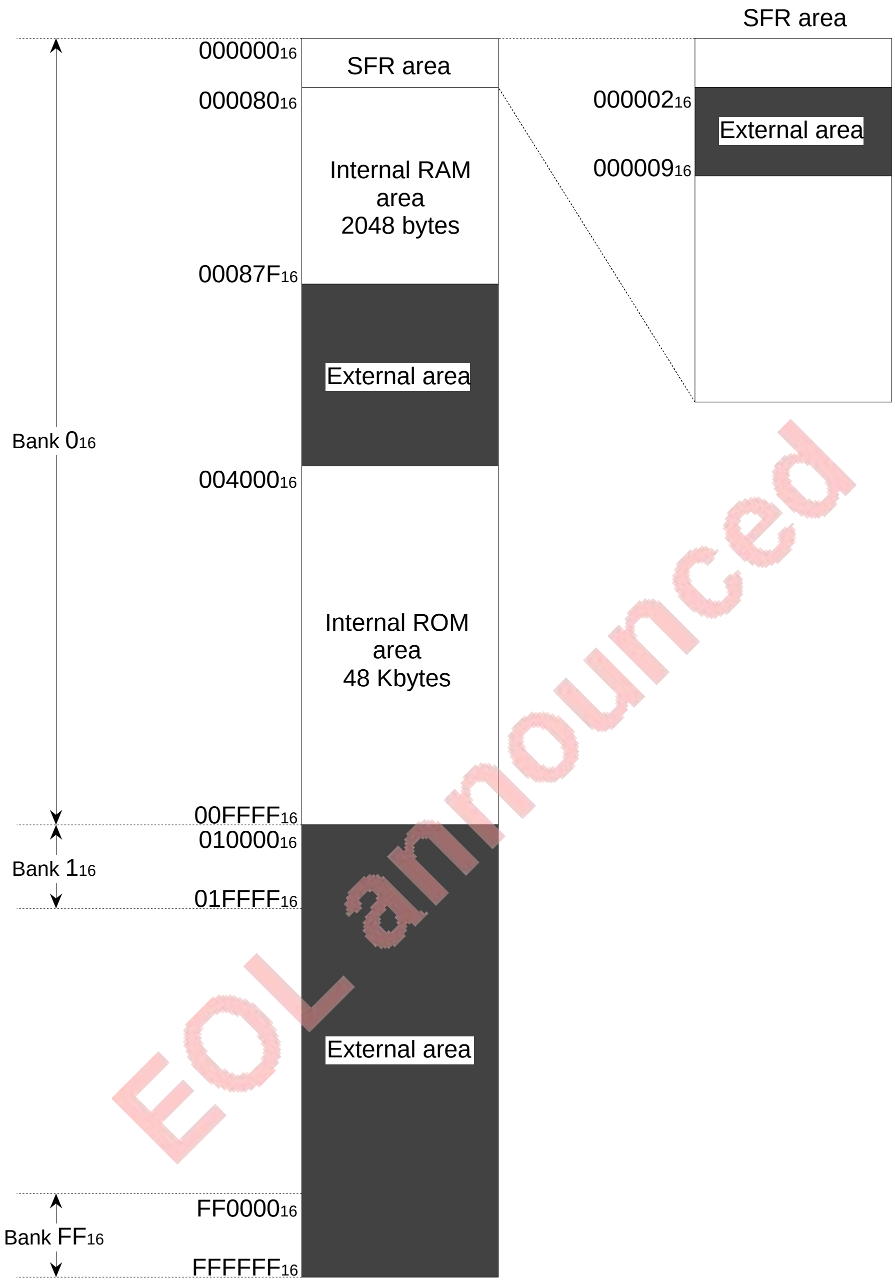
1. During single-chip mode



Type name	M37751M6C-XXXFP
	M37751E6C-XXXFP
	M37751E6CFS
	M37751F6CFP

Fig. 1. Memory assignment during single-chip mode

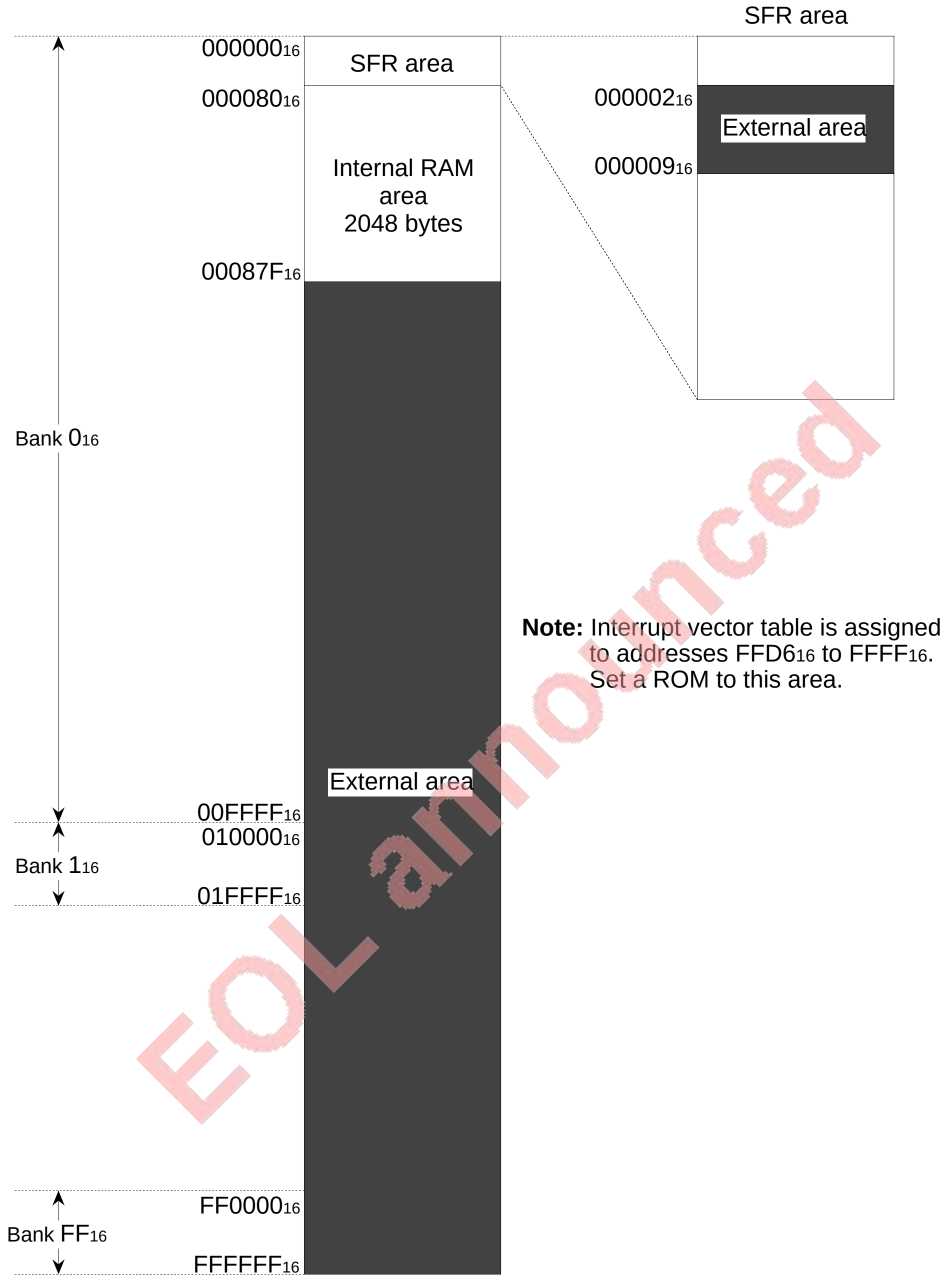
2. During memory expansion mode



Type name	M37751M6C-XXXFP
	M37751E6C-XXXFP
	M37751E6CFS
	M37751F6CFP

Fig. 2. Memory assignment during memory expansion mode

3. During microprocessor mode



Type name	
Type name	M37751M6C-XXXFP
	M37751E6C-XXXFP
	M37751E6CFS
	M37751F6CFP

Fig. 3. Memory assignment during microprocessor mode

Appendix 2. Memory assignment in SFR area

Access characteristics

RW : It is possible to read the bit state at reading. The written value becomes valid data.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid data. It is impossible to read the bit state.
(shaded) : Nothing is assigned. It is impossible to read the bit state. The written value is ignored.

State immediately after a reset

0 : "0" immediately after a reset.
1 : "1" immediately after a reset.
? : Undefined immediately after a reset.
0 (shaded) : Always "0" at reading
? (shaded) : Always undefined at reading
0 (hatched) : "0" immediately after a reset. Fix this bit to "0."

Address	Register name	Access characteristics b7							b0	State immediately after a reset b7							b0
0_{16}										?							
1_{16}										?							
2_{16}	Port P0 register	RW								?							
3_{16}	Port P1 register	RW								?							
4_{16}	Port P0 direction register	RW								00_{16}							
5_{16}	Port P1 direction register	RW								00_{16}							
6_{16}	Port P2 register	RW								?							
7_{16}	Port P3 register					RW				0	0	0	0	?			
8_{16}	Port P2 direction register	RW								00_{16}							
9_{16}	Port P3 direction register					RW				0	0	0	0	0	0	0	0
A_{16}	Port P4 register	RW								?							
B_{16}	Port P5 register	RW								?							
C_{16}	Port P4 direction register	RW								00_{16}							
D_{16}	Port P5 direction register	RW								00_{16}							
E_{16}	Port P6 register	RW								?							
F_{16}	Port P7 register	RW								?							
10_{16}	Port P6 direction register	RW								00_{16}							
11_{16}	Port P7 direction register	RW								00_{16}							
12_{16}	Port P8 register	RW								?							
13_{16}										?							
14_{16}	Port P8 direction register	RW								00_{16}							
15_{16}										?							
16_{16}										?							
17_{16}										?							
18_{16}										?							
19_{16}										?							
$1A_{16}$										?							
$1B_{16}$										?							
$1C_{16}$										?							
$1D_{16}$										?							
$1E_{16}$	A-D control register 0	RW								0	0	0	0	0	?	?	?
$1F_{16}$	A-D control register 1				RW					?	?	?	0	0	0	1	1

Access characteristics

RW : It is possible to read the bit state at reading. The written value becomes valid data.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid data. It is impossible to read the bit state.
(shaded) : Nothing is assigned. It is impossible to read the bit state. The written value is ignored.

State immediately after a reset

0 : "0" immediately after a reset.
1 : "1" immediately after a reset.
? : Undefined immediately after a reset.

(shaded 0) : Always "0" at reading
(shaded ?) : Always undefined at reading
(hatched 0) : "0" immediately after a reset. Fix this bit to "0."

Address	Register name	Access characteristics (b7 … b0)	State immediately after a reset (b7 … b0)
20_{16}	A-D register 0	RO	?
21_{16}		RO	0 0 0 0 0 0 (shaded) ?
22_{16}	A-D register 1	RO	?
23_{16}		RO	0 0 0 0 0 0 (shaded) ?
24_{16}	A-D register 2	RO	?
25_{16}		RO	0 0 0 0 0 0 (shaded) ?
26_{16}	A-D register 3	RO	?
27_{16}		RO	0 0 0 0 0 0 (shaded) ?
28_{16}	A-D register 4	RO	?
29_{16}		RO	0 0 0 0 0 0 (shaded) ?
$2A_{16}$	A-D register 5	RO	?
$2B_{16}$		RO	0 0 0 0 0 0 (shaded) ?
$2C_{16}$	A-D register 6	RO	?
$2D_{16}$		RO	0 0 0 0 0 0 (shaded) ?
$2E_{16}$	A-D register 7	RO	?
$2F_{16}$		RO	0 0 0 0 0 0 (shaded) ?
30_{16}	UART0 transmit/receive mode register	RW	00_{16}
31_{16}	UART0 baud rate register	WO	? (shaded)
32_{16}	UART0 transmit buffer register	WO	? (shaded)
33_{16}		b7–b1: (shaded), b0: WO	? (shaded)
34_{16}	UART0 transmit/receive control register 0	RW (shaded) (shaded) (shaded) RO RW RW RW	0 ? ? ? (shaded) 1 0 0 0
35_{16}	UART0 transmit/receive control register 1	RO RO RO RO RO RW RO RW	0 0 0 0 0 0 1 0
36_{16}	UART0 receive buffer register	RO	?
37_{16}		b7–b1: (shaded), b0: RO	0 0 0 0 0 0 0 (shaded) ?
38_{16}	UART1 transmit/receive mode register	RW	00_{16}
39_{16}	UART1 baud rate register	WO	? (shaded)
$3A_{16}$	UART1 transmit buffer register	WO	? (shaded)
$3B_{16}$		b7–b1: (shaded), b0: WO	? (shaded)
$3C_{16}$	UART1 transmit/receive control register 0	RW (shaded) (shaded) (shaded) RO RW RW RW	0 ? ? ? (shaded) 1 0 0 0
$3D_{16}$	UART1 transmit/receive control register 1	RO RO RO RO RO RW RO RW	0 0 0 0 0 0 1 0
$3E_{16}$	UART1 receive buffer register	RO	?
$3F_{16}$		b7–b1: (shaded), b0: RO	0 0 0 0 0 0 0 (shaded) ?

Access characteristics

RW : It is possible to read the bit state at reading. The written value becomes valid data.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid data. It is impossible to read the bit state.
(shaded) : Nothing is assigned. It is impossible to read the bit state. The written value is ignored.

State immediately after a reset

0 : "0" immediately after a reset.
1 : "1" immediately after a reset.
? : Undefined immediately after a reset.

(shaded 0) : Always "0" at reading
(shaded ?) : Always undefined at reading
(hatched 0) : "0" immediately after a reset. Fix this bit to "0."

Address	Register name	Access characteristics (b7 … b0)	State immediately after a reset (b7 … b0)
40_{16}	Count start register	RW	00_{16}
41_{16}		(shaded)	(shaded) ?
42_{16}	One-shot start register	(shaded) (shaded) (shaded) | WO (b4–b0)	(shaded) ? (b7–b5) | 0 0 0 0 0
43_{16}		(shaded)	(shaded) ?
44_{16}	Up-down register	WO (b7–b5) | RW (b4–b0)	0 0 0 0 0 0 0 0
45_{16}		(shaded)	(shaded) ?
46_{16}	Timer A0 register	✻	?
47_{16}		✻	?
48_{16}	Timer A1 register	✻	?
49_{16}		✻	?
$4A_{16}$	Timer A2 register	✻	?
$4B_{16}$		✻	?
$4C_{16}$	Timer A3 register	✻	?
$4D_{16}$		✻	?
$4E_{16}$	Timer A4 register	✻	?
$4F_{16}$		✻	?
50_{16}	Timer B0 register	✻	?
51_{16}		✻	?
52_{16}	Timer B1 register	✻	?
53_{16}		✻	?
54_{16}	Timer B2 register	✻	?
55_{16}		✻	?
56_{16}	Timer A0 mode register	RW	00_{16}
57_{16}	Timer A1 mode register	RW	00_{16}
58_{16}	Timer A2 mode register	RW	00_{16}
59_{16}	Timer A3 mode register	RW	00_{16}
$5A_{16}$	Timer A4 mode register	RW	00_{16}
$5B_{16}$	Timer B0 mode register	RW (b7–b6) | ✻ | (shaded) | RW (b3–b0)	0 0 ? (shaded ?) 0 0 0 0
$5C_{16}$	Timer B1 mode register	RW (b7–b6) | ✻ | (shaded) | RW (b3–b0)	0 0 ? (shaded ?) 0 0 0 0
$5D_{16}$	Timer B2 mode register	RW (b7–b6) | ✻ | (shaded) | RW (b3–b0)	0 0 ? (shaded ?) 0 0 0 0
$5E_{16}$	Processor mode register 0	RW (b7–b4) | WO | RW | ✻ | RW	0 (hatched 0) 0 0 (shaded 0) (hatched 0) ✻ 0
$5F_{16}$	Processor mode register 1	RW	(hatched 0) (hatched 0) 0 0 0 0 (hatched 0) (hatched 0)

✻ The access characteristics at addresses 46_{16} to $4F_{16}$ varies according to Timer A's operating mode. (Refer to **"Chapter 5. TIMER A."**)
✻ The access characteristics at addresses 50_{16} to 55_{16} varies according to Timer B's operating mode. (Refer to **"Chapter 6. TIMER B."**)
✻ The access characteristics of bit 5 at addresses $5B_{16}$ to $5D_{16}$ varies according to Timer B's operating mode. (Refer to **"Chapter 6. TIMER B."**)
✻ The access characteristics of bit 1 at address $5E_{16}$ and its state immediately after a reset vary according to the voltage level supplied to the CNVss pin. (Refer to section **"2.5 Processor modes."**)

Access characteristics

RW : It is possible to read the bit state at reading. The written value becomes valid data.
RO : It is possible to read the bit state at reading. The written value becomes invalid.
WO : The written value becomes valid data. It is impossible to read the bit state.
(shaded box) : Nothing is assigned. It is impossible to read the bit state. The written value is ignored.

State immediately after a reset

0 : "0" immediately after a reset.
1 : "1" immediately after a reset.
? : Undefined immediately after a reset.

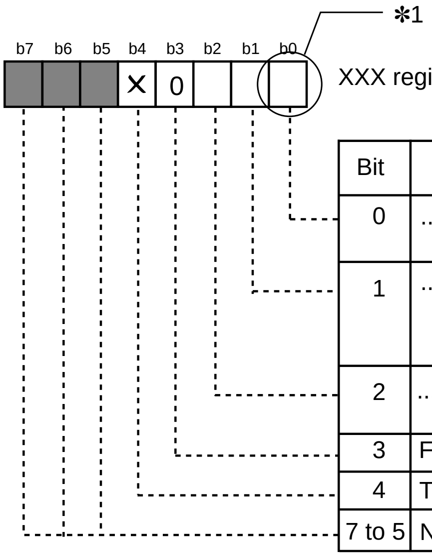
(shaded box with 0) : Always "0" at reading
(shaded box with ?) : Always undefined at reading
(hatched box with 0) : "0" immediately after a reset. Fix this bit to "0."

Address	Register name	Access b7							Access b0	State b7							State b0
60_{16}	Watchdog timer register	**(Note 1)**								? **(Note 2)**							
61_{16}	Watchdog timer frequency select register								RW	?							0
62_{16}										?							
63_{16}										?							
64_{16}										?							
65_{16}										?							
66_{16}										?							
67_{16}										?							
68_{16}										?							
69_{16}										?							
$6A_{16}$										?							
$6B_{16}$										?							
$6C_{16}$										?							
$6D_{16}$										?							
$6E_{16}$										?							
$6F_{16}$										?							
70_{16}	A-D conversion interrupt control register					RW				?				?	0	0	0
71_{16}	UART0 transmit interrupt control register					RW				?				0	0	0	0
72_{16}	UART0 receive interrupt control register					RW				?				0	0	0	0
73_{16}	UART1 transmit interrupt control register					RW				?				0	0	0	0
74_{16}	UART1 receive interrupt control register					RW				?				0	0	0	0
75_{16}	Timer A0 interrupt control register					RW				?				0	0	0	0
76_{16}	Timer A1 interrupt control register					RW				?				0	0	0	0
77_{16}	Timer A2 interrupt control register					RW				?				0	0	0	0
78_{16}	Timer A3 interrupt control register					RW				?				0	0	0	0
79_{16}	Timer A4 interrupt control register					RW				?				0	0	0	0
$7A_{16}$	Timer B0 interrupt control register					RW				?				0	0	0	0
$7B_{16}$	Timer B1 interrupt control register					RW				?				0	0	0	0
$7C_{16}$	Timer B2 interrupt control register					RW				?				0	0	0	0
$7D_{16}$	$\overline{INT_0}$ interrupt control register			RW						?		0	0	0	0	0	0
$7E_{16}$	$\overline{INT_1}$ interrupt control register			RW						?		0	0	0	0	0	0
$7F_{16}$	$\overline{INT_2}$ interrupt control register			RW						?		0	0	0	0	0	0

Notes 1: By writing dummy data to address 60_{16}, a value "FFF_{16}" is set to the watchdog timer. The dummy data is not retained anywhere.
2: The value "FFF_{16}" is set tot the watchdog timer. (Refer to **"Chapter 9. WATCHDOG TIMER."**)

Appendix 3. Control registers

The register structure of each control register assignment in the SFR area are shown on the following pages. The view of the register structure is described below.



XXX register (Address XX_{16})

✻2 ✻3

Bit	Bit name	Functions	At reset	RW
0	... select bit	0 : ... 1 : ...	0	RW
1	... select bit	0 : ... 1 : ... The value is “0” at reading.	Undefined	WO
2	... flag	0 : ... 1 : ...	0	RO
3	Fix this bit to “0.”		0	RW
4	This bit is ignored in ... mode.		0	RW
7 to 5	Nothing is assigned.		Undefined	–

✻4

✻1

- Blank : Set to “0” or “1” to according to the usage.
- 0 : Set to “0” at writing.
- 1 : Set to “1” at writing.
- × : Ignored depending on the specific mode or state. It may be either “0” or “1.”
- (shaded box) : Nothing is assigned.

✻2

- 0 : “0” immediately after a reset.
- 1 : “1” immediately after a reset.
- Undefined : Undefined immediately after a reset.

✻3

- RW : It is possible to read the bit state at reading. The written value becomes valid.
- RO : It is possible to read the bit state at reading. The written value becomes invalid. Accordingly, the written value may be “0” or “1.”
- WO : The written value becomes valid. It is impossible to read the bit state. The value is undefined at reading. However, when [“0” is at reading”] is indicated in the “Function” or “Note” column, the bit is always “0” at reading. (See ✻4 above.)
- — : It is impossible to read the bit state. The value is undefined at reading. However, when [“0” is at reading”] is indicated in the “Function” or “Note” column, the bit is always “0” at reading. (See ✻4 above.) The written value becomes invalid. Accordingly, the written value may be “0” or “1.”

Port Pi register

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi register (i = 0 to 8)
(Addresses 2_{16}, 3_{16}, 6_{16}, 7_{16}, A_{16}, B_{16}, E_{16}, F_{16}, 12_{16})

Bit	Bit name	Functions	At reset	RW
0	Port Pi_0	Data is input/output to/from a pin by reading/writing from/to the corresponding bit. 0 : "L" level 1 : "H" level	Undefined	RW
1	Port Pi_1		Undefined	RW
2	Port Pi_2		Undefined	RW
3	Port Pi_3		Undefined	RW
4	Port Pi_4		Undefined	RW
5	Port Pi_5		Undefined	RW
6	Port Pi_6		Undefined	RW
7	Port Pi_7		Undefined	RW

Note: Bits 7 to 4 of the port P3 register cannot be written and are fixed to "0" at reading.

Port Pi direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi direction register (i = 0 to 8)
(Addresses 4_{16}, 5_{16}, 8_{16}, 9_{16}, C_{16}, D_{16}, 10_{16}, 11_{16}, 14_{16})

Bit	Bit name	Functions	At reset	RW
0	Port Pi_0 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	0	RW
1	Port Pi_1 direction bit		0	RW
2	Port Pi_2 direction bit		0	RW
3	Port Pi_3 direction bit		0	RW
4	Port Pi_4 direction bit		0	RW
5	Port Pi_5 direction bit		0	RW
6	Port Pi_6 direction bit		0	RW
7	Port Pi_7 direction bit		0	RW

Note: Bits 7 to 4 of the port P3 direction register cannot be written and are fixed to "0" at reading.

A-D control register 0

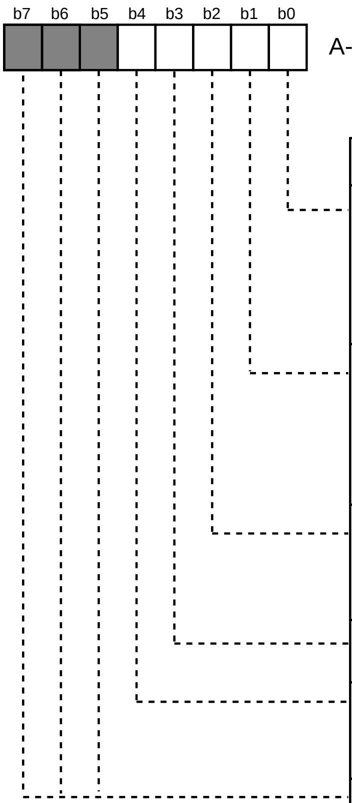
b7 b6 b5 b4 b3 b2 b1 b0

A-D control register 0 (Address $1E_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Analog input select bits (Valid in one-shot and repeat modes) **(Note 1)**	b2 b1 b0 0 0 0 : AN_0 selected 0 0 1 : AN_1 selected 0 1 0 : AN_2 selected 0 1 1 : AN_3 selected 1 0 0 : AN_4 selected 1 0 1 : AN_5 selected 1 1 0 : AN_6 selected 1 1 1 : AN_7 selected **(Note 2)**	Undefined	RW
1			Undefined	RW
2			Undefined	RW
3	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 / Repeat sweep mode 1 **(Note 3)**	0	RW
4			0	RW
5	Trigger select bit	0 : Internal trigger 1 : External trigger	0	RW
6	A-D conversion start bit	0 : Stop A-D conversion 1 : Start A-D conversion	0	RW
7	A-D conversion frequency (ϕ_{AD}) select bit 0	When A-D conversion frequency (ϕ_{AD}) select bit 1 (bit 4 at address $1F_{16}$) = "0," 0 : f_2 divided by 4, or f_4 divided by 4 1 : f_2 divided by 2, or f_4 divided by 2 When A-D conversion frequency (ϕ_{AD}) select bit 1 (bit 4 at address $1F_{16}$) = "1," 0 : f_2 or f_4 1 : Not selected	0	RW

Notes 1: These bits are ignored in the single sweep, repeat sweep mode 0, and repeat sweep mode 1. (They may be either "0" or "1.")
2: When selecting an external trigger, the AN_7 pin cannot be used as an analog input pin.
3: Use the A-D operation mode select bit 1 (bit 2 at address $1F_{16}$) to select either the repeat sweep mode 0 or repeat sweep mode 1.
4: Writing to each bit (except bit 6) of the A-D control register 0 must be performed while the A-D converter halts.

A-D control register 1



A-D control register 1 (Address $1F_{16}$)

Bit	Bit name	Functions	At reset	RW
0	A-D sweep pin select bits (Valid in single sweep, repeat sweep mode 0 and repeat sweep mode 1) **(Note 1)**	●Single sweep mode/Repeat sweep mode 0 b1 b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0 to AN_3 (4 pins) 1 0 : AN_0 to AN_5 (6 pins) 1 1 : AN_0 to AN_7 (8 pins) **(Note 2)**	1	RW
1		●Repeat sweep mode 1 **(Note 3)** b1 b0 0 0 : AN_0 (1 pin) 0 1 : AN_0, AN_1 (2 pins) 1 0 : AN_0 to AN_2 (3 pins) 1 1 : AN_0 to AN_3 (4 pins)	1	RW
2	A-D operation mode select bit 1 (Use in repeat sweep mode 0 and repeat sweep mode 1) **(Note 4)**	0 : Repeat sweep mode 0 1 : Repeat sweep mode 1	0	RW
3	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	0	RW
4	A-D conversion frequency (ϕ_{AD}) select bit 1	Refer to A-D conversion frequency (ϕ_{AD}) select bit 0 (bit 7 at address $1E_{16}$)	0	RW
7 to 5	Nothing is assigned.		Undefined	–

Notes 1: These bits are invalid in the one-shot and repeat modes. (They may be either “0” or “1.”)

2: When selecting an external trigger, the AN_7 pin cannot be used as an analog input pin.

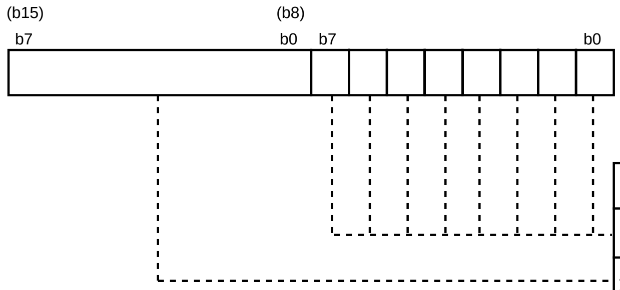
3: Analog input pins which are frequently A-D converted are selected in the repeat sweep mode 1.

4: Fix this bit to “0” in the one-shot, repeat, and single sweep modes.

5: Writing to each bit of the A-D control register 1 must be performed while the A-D converter halts.

A-D register i

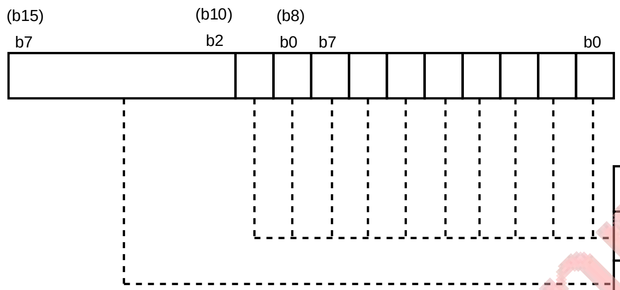
●8-bit mode



A-D register 0 (Addresses 21_{16}, 20_{16})
A-D register 1 (Addresses 23_{16}, 22_{16})
A-D register 2 (Addresses 25_{16}, 24_{16})
A-D register 3 (Addresses 27_{16}, 26_{16})
A-D register 4 (Addresses 29_{16}, 28_{16})
A-D register 5 (Addresses $2B_{16}$, $2A_{16}$)
A-D register 6 (Addresses $2D_{16}$, $2C_{16}$)
A-D register 7 (Addresses $2F_{16}$, $2E_{16}$)

Bit	Functions	At reset	RW
7 to 0	Reads an A-D conversion result.	Undefined	RO
15 to 8	The value is “0” at reading.	0	RO

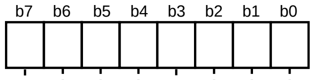
●10-bit mode



A-D register 0 (Addresses 21_{16}, 20_{16})
A-D register 1 (Addresses 23_{16}, 22_{16})
A-D register 2 (Addresses 25_{16}, 24_{16})
A-D register 3 (Addresses 27_{16}, 26_{16})
A-D register 4 (Addresses 29_{16}, 28_{16})
A-D register 5 (Addresses $2B_{16}$, $2A_{16}$)
A-D register 6 (Addresses $2D_{16}$, $2C_{16}$)
A-D register 7 (Addresses $2F_{16}$, $2E_{16}$)

Bit	Functions	At reset	RW
9 to 0	Reads an A-D conversion result.	Undefined	RO
15 to 10	The value is “0” at reading.	0	RO

UARTi transmit/receive mode register



UART0 transmit/receive mode register (Address 30_{16})
UART1 transmit/receive mode register (Address 38_{16})

Bit	Bit name	Functions	At reset	RW
0	Serial I/O mode select bits	b2 b1 b0 0 0 0 : Serial I/O disabled (P8 functions as a programmable I/O port.) 0 0 1 : Clock synchronous serial I/O mode 0 1 0 : Not selected 0 1 1 : Not selected 1 0 0 : UART mode (Transfer data length = 7 bits) 1 0 1 : UART mode (Transfer data length = 8 bits) 1 1 0 : UART mode (Transfer data length = 9 bits) 1 1 1 : Not selected	0	RW
1			0	RW
2			0	RW
3	Internal/External clock select bit	0 : Internal clock 1 : External clock	0	RW
4	Stop bit length select bit (Valid in UART mode) (**Note**)	0 : One stop bit 1 : Two stop bits	0	RW
5	Odd/Even parity select bit (Valid in UART mode when parity enable bit is "1") (**Note**)	0 : Odd parity 1 : Even parity	0	RW
6	Parity enable bit (Valid in UART mode) (**Note**)	0 : Parity disabled 1 : Parity enabled	0	RW
7	Sleep select bit (Valid in UART mode) (**Note**)	0 : Sleep mode cleared (ignored) 1 : Sleep mode selected	0	RW

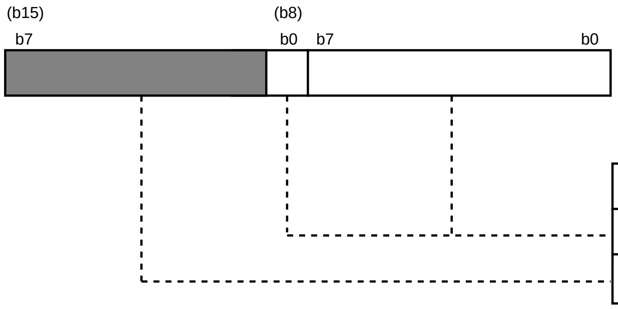
Note: Bits 4 to 6 are ignored in the clock synchronous serial I/O mode. (They may be either "0" or "1.") Additionally, fix bit 7 to "0."

UARTi baud rate register (BRGi)

UART0 baud rate register (Address 31_{16})
UART1 baud rate register (Address 39_{16})

Bit	Functions	At reset	RW
7 to 0	Can be set to "00_{16}" to "FF_{16}." Assuming that the set value = n, BRGi divides the count source frequency by n + 1.	Undefined	WO

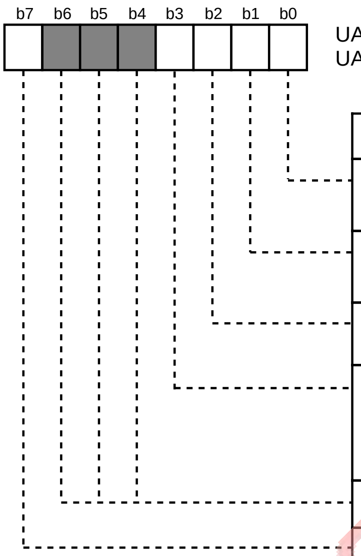
UARTi transmit buffer register



UART0 transmit buffer register (Addresses 33_{16}, 32_{16})
UART1 transmit buffer register (Addresses $3B_{16}$, $3A_{16}$)

Bit	Functions	At reset	RW
8 to 0	Transmit data is set.	Undefined	WO
15 to 9	Nothing is assigned.	Undefined	–

UARTi transmit/receive control register 0

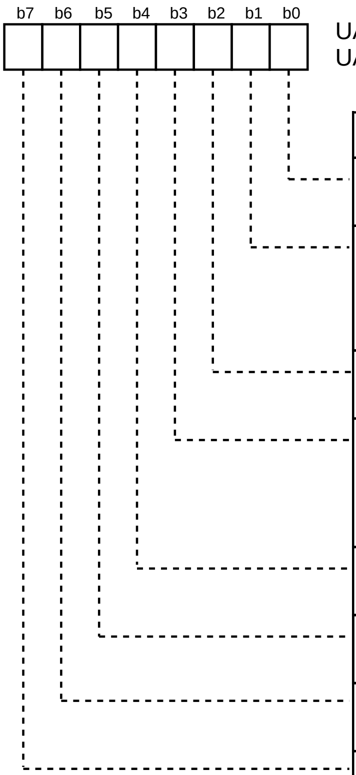


UART0 transmit/receive control register 0 (Address 34_{16})
UART1 transmit/receive control register 0 (Address $3C_{16}$)

Bit	Bit name	Functions	At reset	RW
0	BRG count source select bits	b1 b0 0 0 : f_2 / f_4 0 1 : f_{16} / f_{32} 1 0 : f_{64} / f_{128} 1 1 : f_{512} / f_{1024}	0	RW
1			0	RW
2	$\overline{CTS}$/$\overline{RTS}$ select bit	0 : $\overline{CTS}$ function selected. 1 : RTS function selected.	0	RW
3	Transmit register empty flag	0 : Data present in transmit register. (During transmitting) 1 : No data present in transmit register. (Transmitting completed)	1	RO
6 to 4	Nothing is assigned.		Undefined	–
7	Transfer format select bit (Used in clock synchronous serial I/O mode) **(Note)**	0 : LSB (Least Significant Bit) first 1 : MSB (Most Significant Bit) first	0	RW

Note: Fix bit 7 to "0" in the UART mode or when Serial I/O is ignored.

UARTi transmit/receive control register 1

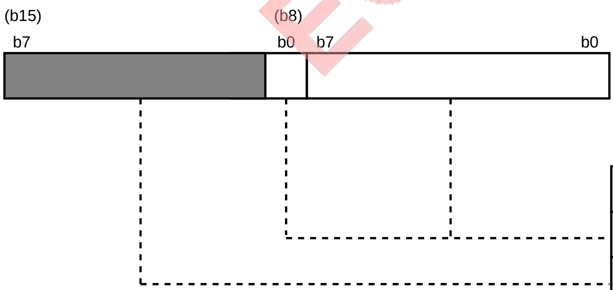


UART0 transmit/receive control register 1 (Address 35_{16})
UART1 transmit/receive control register 1 (Address $3D_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0	RW
1	Transmit buffer empty flag	0 : Data present in transmit buffer register. 1 : No data present in transmit buffer register.	1	RO
2	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0	RW
3	Receive complete flag	0 : No data present in receive buffer register. 1 : Data present in receive buffer register.	0	RO
4	Overrun error flag **(Note 1)**	0 : No overrun error 1 : Overrun error detected	0	RO
5	Framing error flag **(Notes 1, 2)** (Valid in UART mode)	0 : No framing error 1 : Framing error detected	0	RO
6	Parity error flag **(Notes 1, 2)** (Valid in UART mode)	0 : No parity error 1 : Parity error detected	0	RO
7	Error sum flag **(Notes 1, 2)** (Valid in UART mode)	0 : No error 1 : Error detected	0	RO

Notes 1: Bit 4 is cleared to “0” when clearing the receive enable bit to “0.”
Bits 5 and 6 are cleared to “0” when one of the following is performed:
•clearing the receive enable bit to “0.”
•reading the low-order byte of the UARTi receive buffer register (addresses 36_{16}, $3E_{16}$) out.
Bit 7 is cleared to “0” when all of bits 4 to 6 become “0.”
2: Bits 5 to 7 are ignored in the clock synchronous serial I/O mode.

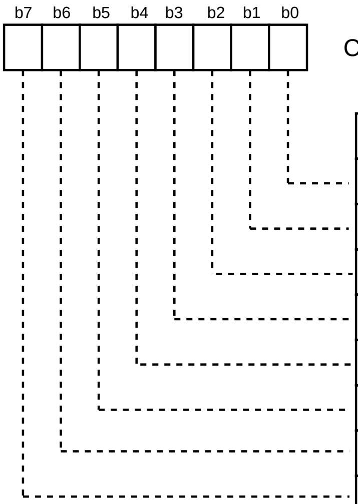
UARTi receive buffer register



UART0 receive buffer register (Addresses 37_{16}, 36_{16})
UART1 receive buffer register (Addresses $3F_{16}$, $3E_{16}$)

Bit	Functions	At reset	RW
8 to 0	Receive data is read out from here.	Undefined	RO
15 to 9	Nothing is assigned. The value is “0” at reading.	0	–

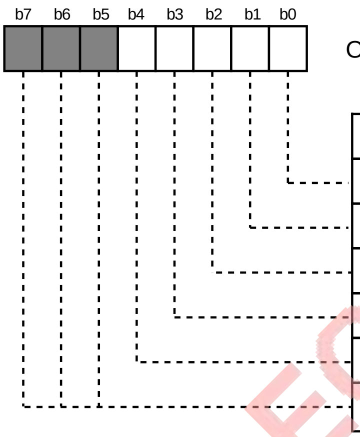
Count start register



Count start register (Address 40_{16})

Bit	Bit name	Functions	At reset	RW
0	Timer A0 count start bit	0 : Stop counting 1 : Start counting	0	RW
1	Timer A1 count start bit		0	RW
2	Timer A2 count start bit		0	RW
3	Timer A3 count start bit		0	RW
4	Timer A4 count start bit		0	RW
5	Timer B0 count start bit		0	RW
6	Timer B1 count start bit		0	RW
7	Timer B2 count start bit		0	RW

One-shot start register



One-shot start register (Address 42_{16})

Bit	Bit name	Functions	At reset	RW
0	Timer A0 one-shot start bit	1 : Start outputting one-shot pulse (valid when selecting internal trigger.) The value is “0” at reading.	0	WO
1	Timer A1 one-shot start bit		0	WO
2	Timer A2 one-shot start bit		0	WO
3	Timer A3 one-shot start bit		0	WO
4	Timer A4 one-shot start bit		0	WO
7 to 5	Nothing is assigned.		Undefined	–

Up-down register

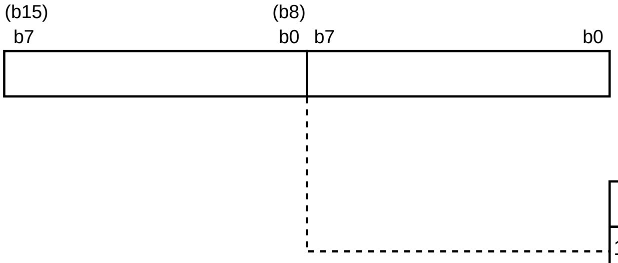
b7 b6 b5 b4 b3 b2 b1 b0

Up-down register (Address 44_{16})

Bit	Bit name	Functions	At reset	RW
0	Timer A0 up-down bit	0 : Down-count 1 : Up-count This function is valid when the contents of the up-down register is selected as the up-down switching factor.	0	RW
1	Timer A1 up-down bit		0	RW
2	Timer A2 up-down bit		0	RW
3	Timer A3 up-down bit		0	RW
4	Timer A4 up-down bit		0	RW
5	Timer A2 two-phase pulse signal processing select bit **(Note)**	0 : Two-phase pulse signal processing function disabled 1 : Two-phase pulse signal processing function enabled When not using the two-phase pulse signal processing function, set the bit to "0." The value is "0" at reading.	0	WO
6	Timer A3 two-phase pulse signal processing select bit **(Note)**		0	WO
7	Timer A4 two-phase pulse signal processing select bit **(Note)**		0	WO

Note: Use the **LDM** or **STA** instruction when writing to bits 5 to 7.

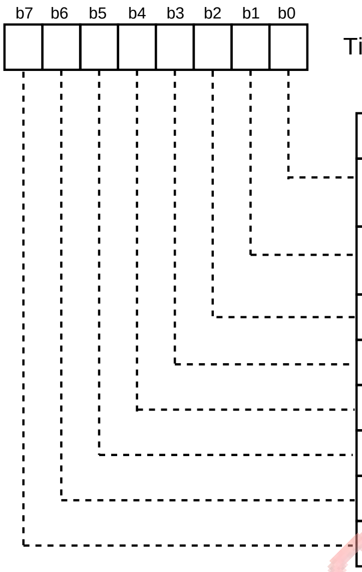
Timer Ai register



Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

Bit	Functions	At reset	RW
15 to 0	These bits have different functions according to the operating mode.	Undefined	RW

Timer Ai mode register



Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot pulse mode 1 1 : Pulse width modulation (PWM) mode	0	RW
1			0	RW
2	These bits have different functions according to the operating mode.		0	RW
3			0	RW
4			0	RW
5			0	RW
6			0	RW
7			0	RW

Timer Mode

(b15) b7 … (b8) b0 | b7 … b0

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

Bit	Functions	At reset	RW
15 to 0	These bits can be set to "0000_{16}" to "$FFFF_{16}$." Assuming that the set value = n, the counter divides the count source frequency by n + 1. When reading, the register indicates the counter value.	Undefined	RW

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 0 : Timer mode	0	RW
1			0	RW
2	Pulse output function select bit	0 : No pulse output (TAi$_{OUT}$ pin functions as a programmable I/O port.) 1 : Pulse output (TAi$_{OUT}$ pin functions as a pulse output pin.)	0	RW
3	Gate function select bits	b4 b3 0 0 : } No gate function 0 1 : } (TAi$_{IN}$ pin functions as a programmable I/O port.) 1 0 : Counter counts only while TAi$_{IN}$ pin's input signal is "L" level. 1 1 : Counter counts only while TAi$_{IN}$ pin's input signal is "H" level.	0	RW
4			0	RW
5	Fix this bit to "0" in the timer mode.		0	RW
6	Count source select bits	b7 b6 0 0 : f_2 / f_4 0 1 : f_{16} / f_{32} 1 0 : f_{64} / f_{128} 1 1 : f_{512} / f_{1024}	0	RW
7			0	RW

Event counter mode

(b15) b7	(b8) b0	b7	b0

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

Bit	Functions	At reset	RW
15 to 0	These bits can be set to "0000_{16}" to "$FFFF_{16}$." Assuming that the set value = n, the counter divides the count source frequency by n + 1 when down-counting, or by $FFFF_{16}$ – n + 1 when up-counting. When reading, the register indicates the counter value.	Undefined	RW

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

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 1 : Event counter mode	0	RW
1			0	RW
2	Pulse output function select bit	0 : No pulse output (TAiOUT pin functions as a programmable I/O port.) 1 : Pulse output (TAiOUT pin functions as a pulse output pin.)	0	RW
3	Count polarity select bit	0 : Counts at falling edge of external signal 1 : Counts at rising edge of external signal	0	RW
4	Up-down switching factor select bit	0 : Contents of up-down register 1 : Input signal to TAiOUT pin	0	RW
5	Fix this bit to "0" in event counter mode.		0	RW
6	These bits are ignored in event counter mode.		0	RW
7			0	RW

X : It may be either "0" or "1."

One-shot pulse mode

(b15) b7 … (b8) b0 | b7 … b0

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

Bit	Functions	At reset	RW
15 to 0	These bits can be set to "0000_{16}" to "$FFFF_{16}$." Assuming that the set value = n, the "H" level width of the one-shot pulse output from the TAi_{OUT} pin is expressed as follows : $\frac{n}{f_i}$.	Undefined	WO

f_i: Frequency of count source (f_2 / f_4, f_{16}/ f_{32}, f_{64}/ f_{128}, or f_{512}/ f_{1024})

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

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 1 0 : One-shot pulse mode	0	RW
1			0	RW
2	Fix this bit to "1" in one-shot pulse mode.		0	RW
3	Trigger select bits	b4 b3 0 0 : } Writing "1" to one-shot start register 0 1 : } (TAi_{IN} pin functions as a programmable I/O port.) 1 0 : Falling edge of TAi_{IN} pin's input signal 1 1 : Rising edge of TAi_{IN} pin's input signal	0	RW
4			0	RW
5	Fix this bit to "0" in one-shot pulse mode.		0	RW
6	Count source select bits	b7 b6 0 0 : f_2 / f_4 0 1 : f_{16} / f_{32} 1 0 : f_{64} / f_{128} 1 1 : f_{512} / f_{1024}	0	RW
7			0	RW

Pulse width modulation (PWM) mode

<When operating as a 16-bit pulse width modulator>

(b15) b7 | (b8) b0 | b7 | b0

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

Bit	Functions	At reset	RW
15 to 0	These bits can be set to "0000_{16}" to "$FFFE_{16}$." Assuming that the set value = n, the "H" level width of the PWM pulse output from the TAi_{OUT} pin is expressed as follows: $\frac{n}{f_i}$	Undefined	WO

f_i: Frequency of count source (f_2 / f_4, f_{16} / f_{32}, f_{64} / f_{128}, or f_{512} / f_{1024})

<When operating as an 8-bit pulse width modulator>

(b15) b7 | (b8) b0 | b7 | b0

Timer A0 register (Addresses 47_{16}, 46_{16})
Timer A1 register (Addresses 49_{16}, 48_{16})
Timer A2 register (Addresses $4B_{16}$, $4A_{16}$)
Timer A3 register (Addresses $4D_{16}$, $4C_{16}$)
Timer A4 register (Addresses $4F_{16}$, $4E_{16}$)

Bit	Functions	At reset	RW
7 to 0	These bits can be set to "00_{16}" to "FF_{16}." Assuming that the set value = m, PWM pulse's period output from the TAi_{OUT} pin is expressed as follows: $\frac{(m + 1)(2^8 - 1)}{f_i}$	Undefined	WO
15 to 8	These bits can be set to "00_{16}" to "FE_{16}." Assuming that the set value = n, the "H" level width of the PWM pulse output from the TAi_{OUT} pin is expressed as follows: $\frac{n(m + 1)}{f_i}$	Undefined	WO

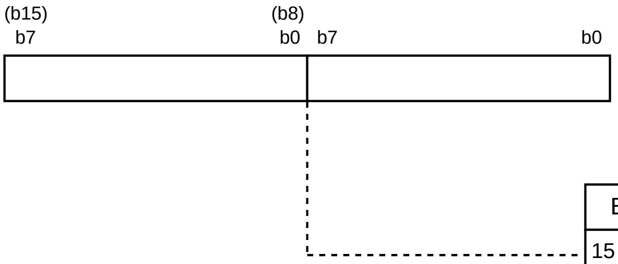
f_i: Frequency of count source (f_2 / f_4, f_{16} / f_{32}, f_{64} / f_{128}, or f_{512} / f_{1024})

b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 1 1 : PWM mode	0	RW
1			0	RW
2	Fix this bit to "1" in PWM mode.		0	RW
3	Trigger select bits	b4 b3 0 0 : } Writing "1" to count start register 0 1 : } (TAi_{IN} pin functions as a programmable I/O port.) 1 0 : Falling edge of TAi_{IN} pin's input signal 1 1 : Rising edge of TAi_{IN} pin's input signal	0	RW
4			0	RW
5	16/8-bit PWM mode select bit	0 : As a 16-bit pulse width modulator 1 : As an 8-bit pulse width modulator	0	RW
6	Count source select bits	b7 b6 0 0 : f_2 / f_4 0 1 : f_{16} / f_{32} 1 0 : f_{64} / f_{128} 1 1 : f_{512} / f_{1024}	0	RW
7			0	RW

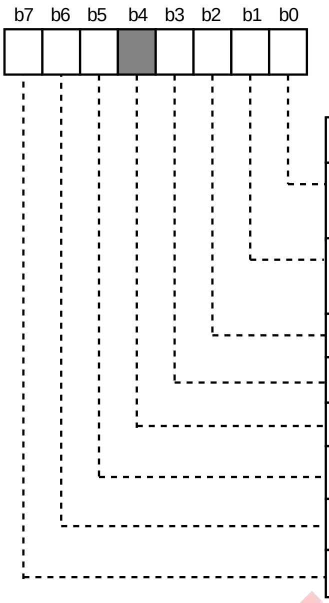
Timer Bi register



Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

Bit	Functions	At reset	RW
15 to 0	These bits have different functions according to the operating mode.	Undefined	RW

Timer Bi mode register



Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/Pulse width measurement mode 1 1 : Not selected	0	RW
1			0	RW
2	These bits have different functions according to the operating mode.		0	RW
3			0	RW
4	Nothing is assigned.		Undefined	–
5	These bits have different functions according to the operating mode.		Undefined	RO **(Note)**
6			0	RW
7			0	RW

Note: Bit 5 is ignored in the timer and event counter modes; its value is undefined at reading.

Timer mode

(b15) b7	(b8) b0	b7	b0

Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

Bit	Functions	At reset	RW
15 to 0	These bits can be set to "0000_{16}" to "$FFFF_{16}$." Assuming that the set value = n, the counter divides the count source frequency by n + 1. When reading, the register indicates the counter value.	Undefined	RW

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

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 0 : Timer mode	0	RW
1			0	RW
2	These bits are ignored in timer mode.		0	RW
3			0	RW
4	Nothing is assigned.		Undefined	–
5	This bit is ignored in timer mode.		Undefined	–
6	Count source select bits	b7 b6 0 0 : f_2 /f_4 0 1 : f_{16} /f_{32} 1 0 : f_{64} /f_{128} 1 1 : f_{512} /f_{1024}	0	RW
7			0	RW

X : It may be either "0" or "1."

Event counter mode

(b15) b7 … (b8) b0 | b7 … b0

Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

Bit	Functions	At reset	RW
15 to 0	These bits can be set to "0000_{16}" to "$FFFF_{16}$." Assuming that the set value = n, the counter divides the count source frequency by n + 1. When reading, the register indicates the counter value.	Undefined	RW

b7	b6	b5	b4	b3	b2	b1	b0
×	×	×	■			0	1

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 1 : Event counter mode	0	RW
1			0	RW
2	Count polarity select bit	b3 b2 0 0 : Count at falling edge of external signal 0 1 : Count at rising edge of external signal 1 0 : Counts at both falling and rising edges of external signal 1 1 : Not selected	0	RW
3			0	RW
4	Nothing is assigned.		Undefined	—
5	This bit is ignored in event counter mode.		Undefined	—
6	These bits are ignored in event counter mode.		0	RW
7			0	RW

× : It may be either "0" or "1."

Pulse period/pulse width measurement mode

(b15) b7 ... (b8) b0 | b7 ... b0

Timer B0 register (Addresses 51_{16}, 50_{16})
Timer B1 register (Addresses 53_{16}, 52_{16})
Timer B2 register (Addresses 55_{16}, 54_{16})

Bit	Functions	At reset	RW
15 to 0	The measurement result of pulse period or pulse width is read out.	Undefined	RO

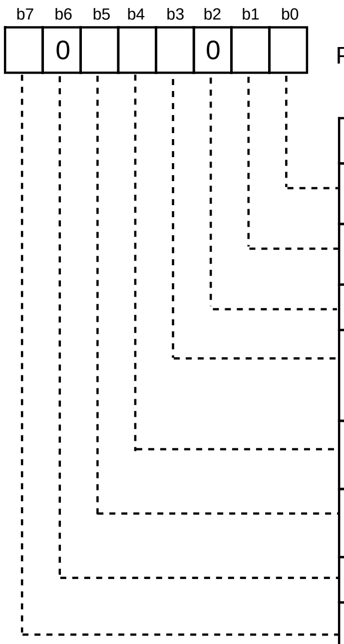
b7 b6 b5 b4 b3 b2 b1 b0
[ ][ ][ ][ ][ ][ ][1][0]

Timer Bi mode register (i = 0 to 2) (Addresses $5B_{16}$ to $5D_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 1 0 : Pulse period/Pulse width measurement mode	0	RW
1			0	RW
2	Measurement mode select bits	b3 b2 0 0 : Pulse period measurement (Interval between falling edges of measurement pulse) 0 1 : Pulse period measurement (Interval between rising edges of measurement pulse) 1 0 : Pulse width measurement (Interval from a falling edge to a rising edge, and from a rising edge to a falling edge of measurement pulse) 1 1 : Not selected	0	RW
3			0	RW
4	Nothing is assigned.		Undefined	–
5	Timer Bi overflow flag **(Note)**	0 : No overflow 1 : Overflowed	Undefined	RO
6	Count source select bits	b7 b6 0 0 : f_2 / f_4 0 1 : f_{16} / f_{32} 1 0 : f_{64} / f_{128} 1 1 : f_{512} / f_{1024}	0	RW
7			0	RW

Note: The timer Bi overflow flag is cleared to “0” by writing to the timer Bi mode register with the count start bit = “1”. The timer Bi overflow flag cannot be set to “1” by software.

Processor mode register 0



Processor mode register 0 (Address $5E_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Memory expansion mode 1 0 : Microprocessor mode 1 1 : Not selected	0	RW
1			0 **(Note 1)**	RW
2	Fix this bit to “0.”		0	RW
3	Software reset bit	The microcomputer is reset by writing “1” to this bit. The value is “0” at reading.	0	WO
4	Interrupt priority detection time select bits	b5 b4 0 0 : 7 cycles of ϕ 0 1 : 4 cycles of ϕ 1 0 : 2 cycles of ϕ 1 1 : Not selected	0	RW
5			0	RW
6	Fix this bit to “0.”		0	RW
7	Clock ϕ_1 output select bit **(Note 2)**	0 : Clock ϕ_1 output disabled (P4$_2$ functions as a programmable I/O port.) 1 : Clock ϕ_1 output enabled (P4$_2$ functions as a clock ϕ_1 output pin.)	0	RW

Notes 1: While supplying the Vcc level to the CNVss pin, this bit becomes “1.” (Fixed to “1.”)
2: This bit is ignored in the microprocessor mode. (It may be either “0” or “1.”)

Processor mode register 1

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

Processor mode register 1 (Address $5F_{16}$)

Bit	Bit name	Functions	At reset	RW
1, 0	Fix these bits to "0."		0	RW
2	Clock source for peripheral devices select bit **(Note)**	0 : ϕ divided by 2 1 : ϕ	0	RW
3	CPU running speed select bit **(Note)**	0 : High-speed running 1 : Low-speed running	0	RW
4	Bus cycle select bits	In high-speed running b5 b4 0 0 : 5ϕ access in high-speed running 0 1 : 4ϕ access in high-speed running 1 0 : 3ϕ access in high-speed running 1 1 : Not selected	0	RW
5		In low-speed running b5 b4 0 0 : Not selected 0 1 : 4ϕ access in low-speed running 1 0 : 3ϕ access in low-speed running 1 1 : 2ϕ access in low-speed running	0	RW
7, 6	Fix these bits to "0."		0	RW

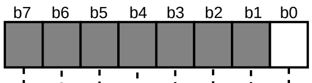
Note: Fix this bit to "0" when $f(X_{IN})$ > 25 MHz.

Watchdog timer register

Watchdog timer register (Address 60_{16})

Bit	Functions	At reset	RW
7 to 0	Initializes the watchdog timer. When a dummy data is written to this register, the watchdog timer's value is initialized to "FFF_{16}." (Dummy data: 00_{16} to FF_{16})	Undefined	–

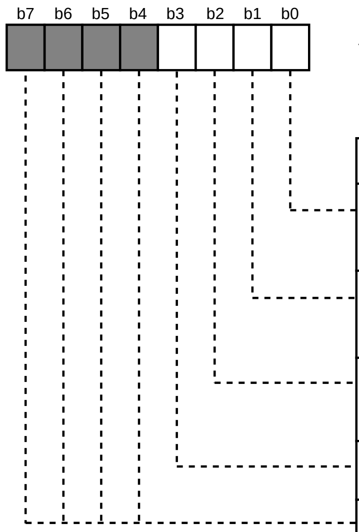
Watchdog timer frequency select register



Watchdog timer frequency select register (Address 61_{16})

Bit	Bit name	Functions	At reset	RW
0	Watchdog timer frequency select bit	0 : Wf_{512} / Wf_{1024} 1 : Wf_{32} / Wf_{64}	0	RW
7 to 1	Nothing is assigned.		Undefined	–

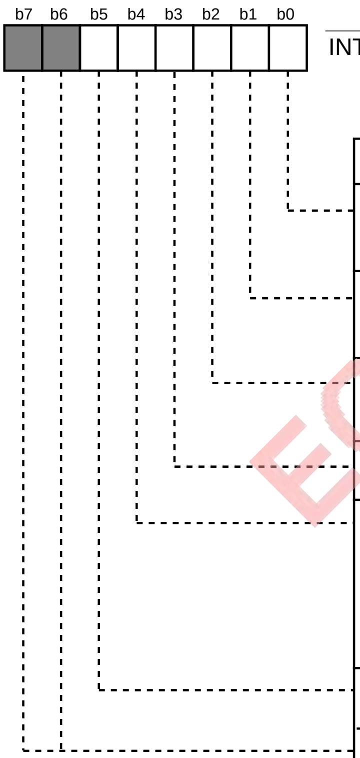
Interrupt control register



A-D conversion, UART0 and 1 transmit, UART0 and 1 receive, timers A0 to A4, timers B0 to B2 interrupt control registers (Addresses 70_{16} to $7C_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 Low level 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7 High level	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit	0 : No interrupt request 1 : Interrupt request	0 **(Note)**	RW
7 to 4	Nothing is assigned.		Undefined	–

Note: The A-D conversion interrupt request bit becomes undefined after reset.



$\overline{INT_0}$ to $\overline{INT_2}$ interrupt control registers (Addresses $7D_{16}$ to $7F_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Interrupt priority level select bits	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 Low level 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7 High level	0	RW
1			0	RW
2			0	RW
3	Interrupt request bit **(Note)**	0 : No interrupt request 1 : Interrupt request	0	RW
4	Polarity select bit	0 : Set the interrupt request bit at "H" level for level sense and at falling edge for edge sense. 1 : Set the interrupt request bit at "L" level for level sense and at rising edge for edge sense.	0	RW
5	Level sense/Edge sense select bit	0 : Edge sense 1 : Level sense	0	RW
7, 6	Nothing is assigned.		Undefined	–

Note: The $\overline{INT_0}$ to $\overline{INT_2}$ interrupt request bits are invalid when selecting the level sense.

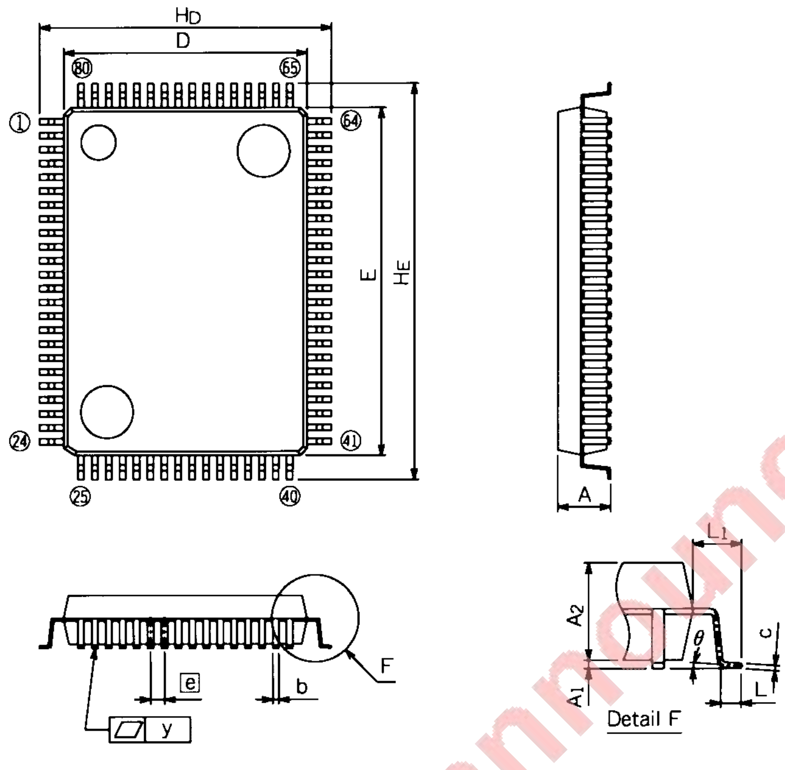
Appendix 4. Package outlines

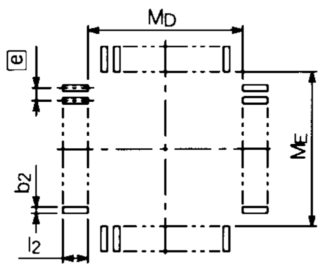
80P6N-A

Plastic 80pin 14×20mm body QFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP80-P-1420-0.80	–	1.58	Alloy 42

Scale :





Recommended Mount Pad

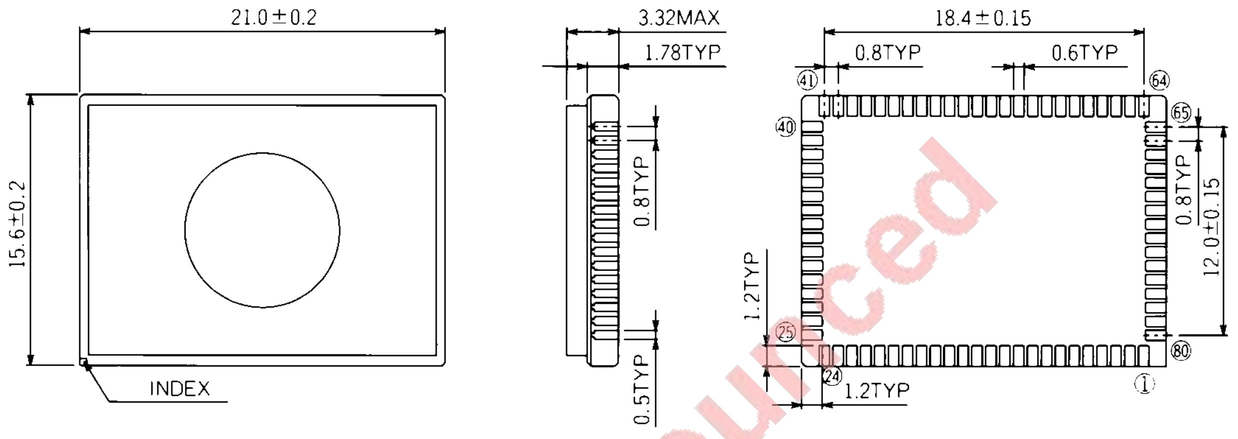
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	3.05
A_1	0	0.1	0.2
A_2	–	2.8	–
b	0.3	0.35	0.45
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	–	0.8	–
H_D	16.5	16.8	17.1
H_E	22.5	22.8	23.1
L	0.4	0.6	0.8
L_1	–	1.4	–
y	–	–	0.1
θ	0°	–	10°
b_2	–	0.5	–
l_2	1.3	–	–
M_D	–	14.6	–
M_E	–	20.6	–

80D0

Glass seal 80pin QFN

EIAJ Package Code	JEDEC Code	Weight(g)
–	–	

Scale :



Appendix 5. Example for processing unused pins

Table 1 Example for processing unused pins in single-chip mode

Pin name	Example of processing
Ports P0 to P8	Set for input mode and connect these pins to Vcc or Vss via a resistor; or set for output mode and leave these pins open. **(Note 1)**
$\overline{E}$	Leave it open.
X_{OUT} **(Note 2)**	
AVcc	Connect this pin to Vcc.
AVss, V_{REF}, BYTE	Connect these pins to Vss.

Notes 1: When setting these ports to the output mode and leave them open, they remain set to the input mode until they are switched to the output mode by software after reset. While ports remain set to the input mode, consequently, voltage levels of pins are unstable, and a power source current can increase.
The contents of the direction register can be changed by noise or a program runaway generated by noise. To improve its reliability, we recommend to periodically set the contents of the direction register by software.
When processing unused pins, use the possible shortest wiring (within 20 mm from the microcomputer).

2: This applies when a clock externally generated is input to the X_{IN} pin.

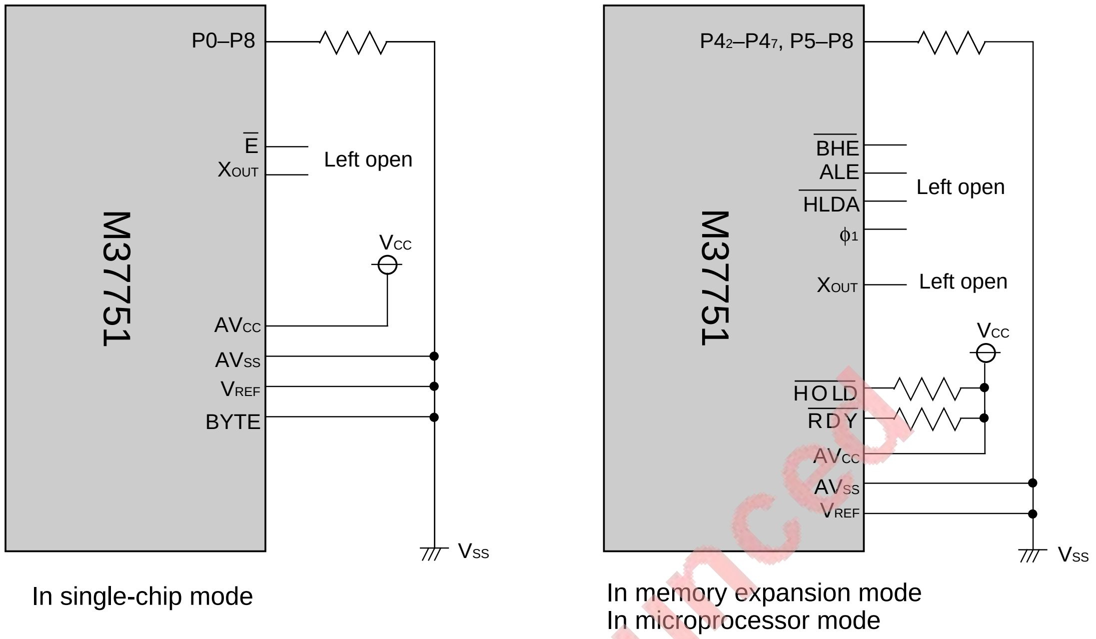
Table 2 Example for processing unused pins in memory expansion mode or microprocessor mode

Pin name	Example of processing
Ports $P4_2$ to $P4_7$, P5 to P8	Set for input mode and connect these pins to Vcc or Vss via a resistor; or set for output mode and leave these pins open. **(Note 1)**
$\overline{\text{BHE}}$ (**Note 2**)	Leave them open. **(Note 4)**
ALE (**Note 3**)	
$\overline{\text{HLDA}}$, ϕ_1	
X_{OUT} (**Note 5**)	Leave it open.
$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$	Connect these pins to Vcc via a resistor (pull-up).
AVcc	Connect this pin to Vcc.
AVss, V_{REF}	Connect these pins to Vss.

Notes 1: When setting these ports to the output mode and leave them open, they remain set to the input mode until they are switched to the output mode by software after reset. While ports remain set to the input mode, consequently, voltage levels of pins are unstable, and a power source current can increase.
The contents of the direction register can be changed by noise or a program runaway generated by noise. To improve its reliability, we recommend to periodically set the contents of the direction register by software.
When processing unused pins, use the possible shortest wiring (within 20 mm from the microcomputer).
2: This applies when "H" level is input to the BYTE pin.
3: This applies when "H" level is input to the BYTE pin and the access space is 64 Kbytes.
4: When supplying Vss level to the CNVss pin, these pins remain set to the input mode until they are switched to the output mode by software after reset (until the pin function is switched in the case of the ϕ_1 pin in the memory expansion mode). While pins remain set to the input mode, consequently, voltage levels of pins are unstable, and a power source current can increase.
5: This applies when a clock externally generated is input to the X_{IN} pin.

Appendix 5. Example for processing unused pins

● When setting ports for input mode



● When setting ports for output mode

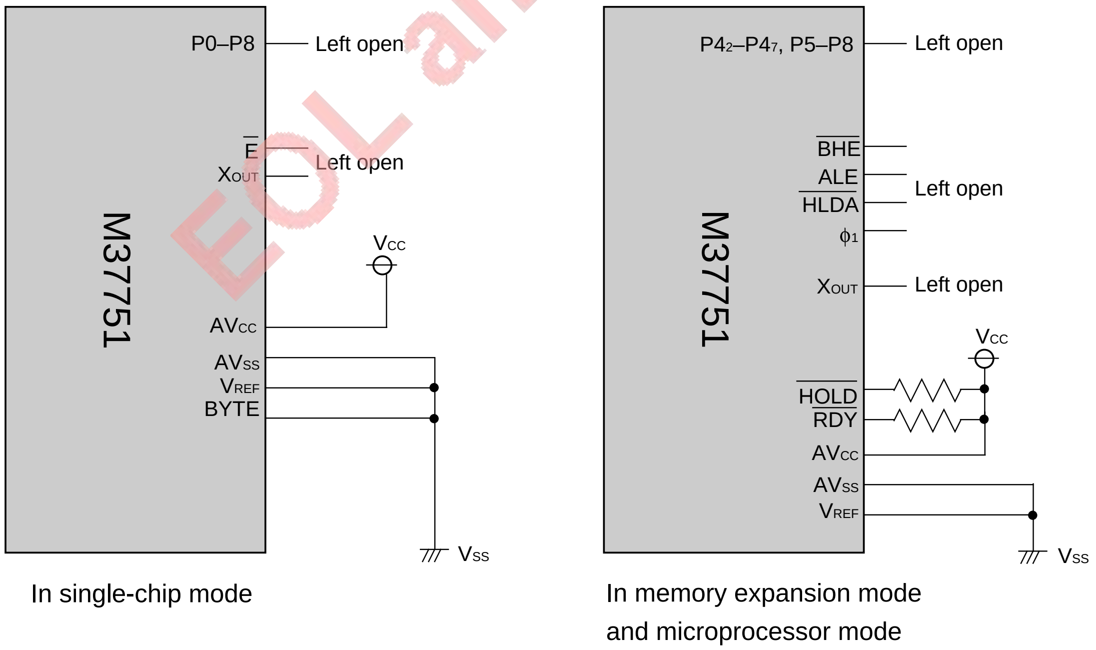


Fig. 4. Example for processing unused pins

Appendix 6. Hexadecimal instruction code table

7751 SERIES INSTRUCTION CODE TABLE-1

D_7–D_4 \ D_3–D_0		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	Hexadecimal notation	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA A,(DIR,X)		ORA A,SR	SEB DIR,b	ORA A,DIR	ASL DIR	ORA A,L(DIR)	PHP	ORA A,IMM	ASL A	PHD	SEB ABS,b	ORA A,ABS	ASL ABS	ORA A,ABL
0001	1	BPL	ORA A,(DIR),Y	ORA A,(DIR)	ORA A,(SR),Y	CLB DIR,b	ORA A,DIR,X	ASL DIR,X	ORA A,L(DIR),Y	CLC	ORA A,ABS,Y	DEC A	TAS	CLB ABS,b	ORA A,ABS,X	ASL ABS,X	ORA A,ABL,X
0010	2	JSR ABS	AND A,(DIR,X)	JSR ABL	AND A,SR	BBS DIR,b,R	AND A,DIR	ROL DIR	AND A,L(DIR)	PLP	AND A,IMM	ROL A	PLD	BBS ABS,b,R	AND A,ABS	ROL ABS	AND A,ABL
0011	3	BMI	AND A,(DIR),Y	AND A,(DIR)	AND A,(SR),Y	BBC DIR,b,R	AND A,DIR,X	ROL DIR,X	AND A,L(DIR),Y	SEC	AND A,ABS,Y	INC A	TSA	BBC ABS,b,R	AND A,ABS,X	ROL ABS,X	AND A,ABL,X
0100	4	RTI	EOR A,(DIR,X)	Note 1	EOR A,SR	MVP	EOR A,DIR	LSR DIR	EOR A,L(DIR)	PHA	EOR A,IMM	LSR A	PHG	JMP ABS	EOR A,ABS	LSR ABS	EOR A,ABL
0101	5	BVC	EOR A,(DIR),Y	EOR A,(DIR)	EOR A,(SR),Y	MVN	EOR A,DIR,X	LSR DIR,X	EOR A,L(DIR),Y	CLI	EOR A,ABS,Y	PHY	TAD	JMP ABL	EOR A,ABS,X	LSR ABS,X	EOR A,ABL,X
0110	6	RTS	ADC A,(DIR,X)	PER	ADC A,SR	LDM DIR	ADC A,DIR	ROR DIR	ADC A,L(DIR)	PLA	ADC A,IMM	ROR A	RTL	JMP (ABS)	ADC A,ABS	ROR ABS	ADC A,ABL
0111	7	BVS	ADC A,(DIR),Y	ADC A,(DIR)	ADC A,(SR),Y	LDM DIR,X	ADC A,DIR,X	ROR DIR,X	ADC A,L(DIR),Y	SEI	ADC A,ABS,Y	PLY	TDA	JMP (ABS,X)	ADC A,ABS,X	ROR ABS,X	ADC A,ABL,X
1000	8	BRA REL	STA A,(DIR,X)	BRA REL	STA A,SR	STY DIR	STA A,DIR	STX DIR	STA A,L(DIR)	DEY	Note 2	TXA	PHT	STY ABS	STA A,ABS	STX ABS	STA A,ABL
1001	9	BCC	STA A,(DIR),Y	STA A,(DIR)	STA A,(SR),Y	STY DIR,X	STA A,DIR,X	STX DIR,Y	STA A,L(DIR),Y	TYA	STA A,ABS,Y	TXS	TXY	LDM ABS	STA A,ABS,X	LDM ABS,X	STA A,ABL,X
1010	A	LDY IMM	LDA A,(DIR,X)	LDX IMM	LDA A,SR	LDY DIR	LDA A,DIR	LDX DIR	LDA A,L(DIR)	TAY	LDA A,IMM	TAX	PLT	LDY ABS	LDA A,ABS	LDX ABS	LDA A,ABL
1011	B	BCS	LDA A,(DIR),Y	LDA A,(DIR)	LDA A,(SR),Y	LDY DIR,X	LDA A,DIR,X	LDX DIR,Y	LDA A,L(DIR),Y	CLV	LDA A,ABS,Y	TSX	TYX	LDY ABS,X	LDA A,ABS,X	LDX ABS,Y	LDA A,ABL,X
1100	C	CPY IMM	CMP A,(DIR,X)	CLP IMM	CMP A,SR	CPY DIR	CMP A,DIR	DEC DIR	CMP A,L(DIR)	INY	CMP A,IMM	DEX	WIT	CPY ABS	CMP A,ABS	DEC ABS	CMP A,ABL
1101	D	BNE	CMP A,(DIR),Y	CMP A,(DIR)	CMP A,(SR),Y	PEI	CMP A,DIR,X	DEC DIR,X	CMP A,L(DIR),Y	CLM	CMP A,ABS,Y	PHX	STP	JMP L(ABS)	CMP A,ABS,X	DEC ABS,X	CMP A,ABL,X
1110	E	CPX IMM	SBC A,(DIR,X)	SEP IMM	SBC A,SR	CPX DIR	SBC A,DIR	INC DIR	SBC A,L(DIR)	INX	SBC A,IMM	NOP	PSH	CPX ABS	SBC A,ABS	INC ABS	SBC A,ABL
1111	F	BEQ	SBC A,(DIR),Y	SBC A,(DIR)	SBC A,(SR),Y	PEA	SBC A,DIR,X	INC DIR,X	SBC A,L(DIR),Y	SEM	SBC A,ABS,Y	PLX	PUL	JSR (ABS,X)	SBC A,ABS,X	INC ABS,X	SBC A,ABL,X

Notes 1. 42_{16} specifies the contents of the INSTRUCTION CODE TABLE-2. About the second word's codes, refer to the INSTRUCTION CODE TABLE-2.
2. 89_{16} specifies the contents of the INSTRUCTION CODE TABLE-3. About the second word's codes, refer to the INSTRUCTION CODE TABLE-3.

Appendix 6. Hexadecimal instruction code table

7751 SERIES INSTRUCTION CODE TABLE-2 (The first word's code of each instruction is 42_{16}.)

D7–D4 \ D3–D0	Hexadecimal notation	0000 (0)	0001 (1)	0010 (2)	0011 (3)	0100 (4)	0101 (5)	0110 (6)	0111 (7)	1000 (8)	1001 (9)	1010 (A)	1011 (B)	1100 (C)	1101 (D)	1110 (E)	1111 (F)
0000	0		ORA B,(DIR,X)		ORA B,SR		ORA B,DIR		ORA B,L(DIR)	ASR B	ORA B,IMM	ASL B			ORA B,ABS		ORA B,ABL
0001	1		ORA B,(DIR),Y	ORA B,(DIR)	ORA B,(SR),Y		ORA B,DIR,X		ORA B,L(DIR),Y		ORA B,ABS,Y	DEC B	TBS		ORA B,ABS,X		ORA B,ABL,X
0010	2		AND B,(DIR,X)		AND B,SR		AND B,DIR		AND B,L(DIR)		AND B,IMM	ROL B			AND B,ABS		AND B,ABL
0011	3		AND B,(DIR),Y	AND B,(DIR)	AND B,(SR),Y		AND B,DIR,X		AND B,L(DIR),Y		AND B,ABS,Y	INC B	TSB		AND B,ABS,X		AND B,ABL,X
0100	4		EOR B,(DIR,X)		EOR B,SR		EOR B,DIR		EOR B,L(DIR)	PHB	EOR B,IMM	LSR B			EOR B,ABS		EOR B,ABL
0101	5		EOR B,(DIR),Y	EOR B,(DIR)	EOR B,(SR),Y		EOR B,DIR,X		EOR B,L(DIR),Y		EOR B,ABS,Y		TBD		EOR B,ABS,X		EOR B,ABL,X
0110	6		ADC B,(DIR,X)		ADC B,SR		ADC B,DIR		ADC B,L(DIR)	PLB	ADC B,IMM	ROR B			ADC B,ABS		ADC B,ABL
0111	7		ADC B,(DIR),Y	ADC B,(DIR)	ADC B,(SR),Y		ADC B,DIR,X		ADC B,L(DIR),Y		ADC B,ABS,Y		TDB		ADC B,ABS,X		ADC B,ABL,X
1000	8		STA B,(DIR,X)		STA B,SR		STA B,DIR		STA B,L(DIR)			TXB	EXTS B		STA B,ABS		STA B,ABL
1001	9		STA B,(DIR),Y	STA B,(DIR)	STA B,(SR),Y		STA B,DIR,X		STA B,L(DIR),Y	TYB	STA B,ABS,Y				STA B,ABS,X		STA B,ABL,X
1010	A		LDA B,(DIR,X)		LDA B,SR		LDA B,DIR		LDA B,L(DIR)	TBY	LDA B,IMM	TBX	EXTZ B		LDA B,ABS		LDA B,ABL
1011	B		LDA B,(DIR),Y	LDA B,(DIR)	LDA B,(SR),Y		LDA B,DIR,X		LDA B,L(DIR),Y		LDA B,ABS,Y				LDA B,ABS,X		LDA B,ABL,X
1100	C		CMP B,(DIR,X)		CMP B,SR		CMP B,DIR		CMP B,L(DIR)		CMP B,IMM				CMP B,ABS		CMP B,ABL
1101	D		CMP B,(DIR),Y	CMP B,(DIR)	CMP B,(SR),Y		CMP B,DIR,X		CMP B,L(DIR),Y		CMP B,ABS,Y				CMP B,ABS,X		CMP B,ABL,X
1110	E		SBC B,(DIR,X)		SBC B,SR		SBC B,DIR		SBC B,L(DIR)		SBC B,IMM				SBC B,ABS		SBC B,ABL
1111	F		SBC B,(DIR),Y	SBC B,(DIR)	SBC B,(SR),Y		SBC B,DIR,X		SBC B,L(DIR),Y		SBC B,ABS,Y				SBC B,ABS,X		SBC B,ABL,X

7751 SERIES INSTRUCTION CODE TABLE-3 (The first word's code of each instruction is 89 16.)

D7–D4 \ D3–D0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0		MPY (DIR,X)		MPY SR		MPY DIR	ASR DIR	MPY L(DIR)	ASR A	MPY IMM				MPY ABS	ASR ABS	MPY ABL
0001	1		MPY (DIR),Y	MPY (DIR)	MPY (SR),Y		MPY DIR,X	ASR DIR,X	MPY L(DIR),Y		MPY ABS,Y				MPY ABS,X	ASR ABS,X	MPY ABL,X
0010	2		DIV (DIR,X)		DIV SR		DIV DIR		DIV L(DIR)	XAB	DIV IMM				DIV ABS		DIV ABL
0011	3		DIV (DIR),Y	DIV (DIR)	DIV (SR),Y		DIV DIR,X		DIV L(DIR),Y		DIV ABS,Y				DIV ABS,X		DIV ABL,X
0100	4										RLA IMM						
0101	5																
0110	6																
0111	7																
1000	8		MPYS (DIR,X)		MPYS SR		MPYS DIR		MPYS L(DIR)		MPYS IMM		EXTS A		MPYS ABS		MPYS ABL
1001	9		MPYS (DIR),Y	MPYS (DIR)	MPYS (SR),Y		MPYS DIR,X		MPYS L(DIR),Y		MPYS ABS,Y				MPYS ABS,X		MPYS ABL,X
1010	A		DIVS (DIR,X)		DIVS SR		DIVS DIR		DIVS L(DIR)		DIVS IMM		EXTZ A		DIVS ABS		DIVS ABL
1011	B		DIVS (DIR),Y	DIVS (DIR)	DIVS (SR),Y		DIVS DIR,X		DIVS L(DIR),Y		DIVS ABS,Y				DIVS ABS,X		DIVS ABL,X
1100	C			LDT IMM													
1101	D																
1110	E			RMPA Multiplied accumula-tion													
1111	F																

Appendix 7. Machine instructions

7751 SERIES MACHINE INSTRUCTIONS

Symbol	Function	Details
ADC (Note 1,2)	Acc, C←Acc+M+C	Adds the carry, the accumulator and the memory contents. The result is entered into the accumulator. When the D flag is "0", binary additions is done, and when the D flag is "1", decimal addition is done.
AND (Note 1,2)	Acc←Acc∧M	Obtains the logical product of the contents of the accumulator and the contents of the memory. The result is entered into the accumulator.
ASL (Note 1)	m=0 C ← b15···b0 ← 0 m=1 C ← b7···b0 ← 0	Shifts the accumulator or the memory contents one bit to the left. "0" is entered into bit 0 of the accumulator or the memory. The contents of bit 15 (bit 7 when the m flag is "1") of the accumulator or memory before shift is entered into the C flag.
ASR (Note 1)	m=0 →b15···b0 →C m=1 →b7···b0 →C	Shifts the accumulator or the memory contents one bit to the right. The bit 0 of the accumulator or memory is entered into the C flag. The contents of bit 15 (bit 7 when the m flag is "1") of the accumulator or memory before shift is entered into bit 15 (bit 7).
BBC (Note 4)	Mb=0?	Tests the specified bit of the memory. Branches when all the contents of the specified bit is "0".
BBS (Note 4)	Mb=1?	Tests the specified bit of the memory. Branches when all the contents of the specified bit is "1".
BCC	C=0?	Branches when the contents of the C flag is "0".
BCS	C=1?	Branches when the contents of the C flag is "1".
BEQ	Z=1?	Branches when the contents of the Z flag is "1".
BMI	N=1?	Branches when the contents of the N flag is "1".
BNE	Z=0?	Branches when the contents of the Z flag is "0".
BPL	N=0?	Branches when the contents of the N flag is "0".

Symbol	Addressing mode: IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y			L(DIR)		
	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
ADC				69	2 2	2				65	4 4	2				75	5 6	2				72	6 6	2	61	7 8	2	71	8 9	2	67	8 8	2
				42 69	4 4	3				42 65	6 6	3				42 75	7 8	3				42 72	8 8	3	42 61	9 10	3	42 71	10 11	3	42 67	10 10	3
AND				29	2 2	2				25	4 4	2				35	5 6	2				32	6 6	2	21	7 8	2	31	8 9	2	27	8 8	2
				42 29	4 4	3				42 25	6 6	3				42 35	7 8	3				42 32	8 8	3	42 21	9 10	3	42 31	10 11	3	42 27	10 10	3
ASL							0A	2 2	1	06	7 8	2				16	7 8	2															
							42 0A	4 4	2																								
ASR							89 08	4 4	2	89 06	9 10	3				89 16	9 10	3															
							42 08	4 4	2																								
BBC																																	
BBS																																	
BCC																																	
BCS																																	
BEQ																																	
BMI																																	
BNE																																	
BPL																																	

Addressing mode

L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL		
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
77	10 11	2	6D	4 4	3				7D	6 7	3	79	6 7	3	6F	6 7	4
42 77	12 13	3	42 6D	6 6	4				42 7D	8 9	4	42 79	8 9	4	42 6F	8 9	5
37	10 11	2	2D	4 4	3				3D	6 7	3	39	6 7	3	2F	6 7	4
42 37	12 13	3	42 2D	6 6	4				42 3D	8 9	4	42 39	8 9	4	42 2F	8 9	5
			0E	7 8	3				1E	8 9	3						
			89 0E	9 10	4				89 1E	10 11	4						

ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL		
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
7F	7 9	4															
42 7F	9 11	5															
3F	7 9	4															
42 3F	9 11	5															
															90	4 4	2
															B0	4 4	2
															F0	4 4	2
															30	4 4	2
															D0	4 4	2
															10	4 4	2

DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			Multiplied accumulation		
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
						63	5 6	2	73	8 9	2						
						42 63	7 8	3	42 73	10 11	3						
						23	5 6	2	33	8 9	2						
						42 23	7 8	3	42 33	10 11	3						
34	7 8	4	3C	8 9	5												
24	7 8	4	2C	8 9	5												

Processor status register

10	9	8	7	6	5	4	3	2	1	0
IPL			N	V	m	x	D	I	Z	C
•	•	•	N	V	•	•	•	•	Z	C
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	C
•	•	•	N	•	•	•	•	•	Z	C
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																																
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y			L(DIR)		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
BRA (Note 3)	PC←PC±offset PG←PG+1 (carry occurred) PG←PG−1 (borrow occurred)	Jumps to the address indicated by the program counter plus the offset value.																																	
BRK	PC←PC+2 M(S)←PG S←S−1 M(S)←PCH S←S−1 M(S)←PCL S←S−1 M(S)←PSH S←S−1 M(S)←PSL S←S−1 I←1 PCL←ADL PCH←ADM PG←0016	Executes software interruption.	00	15 15	2																														
BVC	V=0?	Branches when the contents of the V flag is "0".																																	
BVS	V=1?	Branches when the contents of the V flag is "1".																																	
CLB (Note 4)	Mb←0	Makes the contents of the specified bit in the memory "0".													14	8 9	3																		
CLC	C←0	Makes the contents of the C flag "0".	18	2 2	1																														
CLI	I←0	Makes the contents of the I flag "0".	58	2 2	1																														
CLM	m←0	Makes the contents of the m flag "0".	D8	2 2	1																														
CLP	PSb←0	Specifies the bit position in the processor status register by the bit pattern of the second byte in the instruction, and sets "0" in that bit.				C2	4 4	2																											
CLV	V←0	Makes the contents of the V flag "0".	B8	2 2	1																														
CMP (Note 1,2)	ACC−M	Compares the contents of the accumulator with the contents of the memory.				C9	2 2	2				C5	4 4	2				D5	5 6	2				D2	6 6	2	C1	7 8	2	D1	8 9	2	C7	8 8	2
						42 C9	4 4	3				42 C5	6 6	3				42 D5	7 8	3				42 D2	8 8	3	42 C1	9 10	3	42 D1	10 11	3	42 C7	10 10	3
CPX (Note 11)	X−M	Compares the contents of the index register X with the contents of the memory.				E0	2 2	2				E4	4 4	2																					
CPY (Note 11)	Y−M	Compares the contents of the index register Y with the contents of the memory.				C0	2 2	2				C4	4 4	2																					
DEC (Note 11)	ACC←ACC−1 or M←M−1	Decrements the contents of the accumulator or memory by 1.							1A	2 2	1	C6	7 8	2				D6	7 8	2															
									42 1A	4 4	2																								
DEX	X←X−1	Decrements the contents of the index register X by 1.	CA	2 2	1																														

Addressing mode																																																						Processor status register										
L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			Multiplied accumulation			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																	80	3 3	2																			·	·	·	·	·	·	·	·	·	·	·
																																	82	3 3	3																													
																																																						·	·	·	·	·	·	·	·	·	·	·
																																	50	4 4	2																			·	·	·	·	·	·	·	·	·	·	·
																																	70	4 4	2																			·	·	·	·	·	·	·	·	·	·	·
						1C	9 11	4																																														·	·	·	·	·	·	·	·	·	·	·
																																																						·	·	·	·	·	·	·	·	·	·	0
																																																						·	·	·	·	·	·	·	·	0	·	·
																																																						·	·	·	·	·	0	·	·	·	·	·
																																																						·	·	·	Specified flag becomes "0".							
																																																						·	·	·	·	0	·	·	·	·	·	·
D7	10 11	2	CD	4 4	3				DD	6 7	3	D9	6 7	3	CF	6 7	4	DF	7 9	4																						C3	5 6	2	D3	8 9	2							·	·	·	N	·	·	·	·	·	Z	C
42 D7	12 13	3	42 CD	6 6	4				42 DD	8 9	4	42 D9	8 9	4	42 CF	8 9	5	42 DF	9 11	5																						42 C3	7 8	3	42 D3	10 11	3																	
			EC	4 4	3																																																	·	·	·	N	·	·	·	·	·	Z	C
			CC	4 4	3																																																	·	·	·	N	·	·	·	·	·	Z	C
			CE	7 8	3				DE	8 9	3																																											·	·	·	N	·	·	·	·	·	Z	·
																																																						·	·	·	N	·	·	·	·	·	Z	·

Symbol	Function	Details	Addressing mode																																
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y			L(DIR)		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
DEY	Y←Y−1	Decrements the contents of the index register Y by 1.	88	2 2	1																														
DIV (Note 2,9,13)	A(quotient)←B, A÷M B(remainder)	The numeral that places the contents of accumulator B to the higher order and the contents of accumulator A to the lower order is divided by the contents of the memory. The quotient is entered into accumulator A and the remainder into accumulator B.				89 29	21 21	3				89 25	23 23	3				89 35	24 25	3				89 32	25 25	3	89 21	26 27	3	89 31	27 28	3	89 27	27 27	3
DIVS (Note 2,9,14)	A(quotient)←B, A÷M (with sign) B(remainder)	The numeral with sign that places the contents of accumulator B to the higher order and the contents of accumulator A to the lower order is divided by the contents of the memory. The quotient is entered into accumulator A and the remainder into accumulator B.				89 A9	23 23	3				89 A5	25 25	3				89 B5	26 27	3				89 B2	27 27	3	89 A1	28 29	3	89 B1	29 30	3	89 A7	29 29	3
EOR (Note 1,2)	ACC←ACC∀M	Logical exclusive sum is obtained of the contents of the accumulator and the contents of the memory. The result is placed into the accumulator.				49	2 2	2				45	4 4	2				55	5 6	2				52	6 6	2	41	7 8	2	51	8 9	2	47	8 8	2
						42 49	4 4	3				42 45	6 6	3				42 55	7 8	3				42 52	8 8	3	42 41	9 10	3	42 51	10 11	3	42 47	10 10	3
EXTS (Note 1)	Bit 7 of ACC=1 b15 b7 b0 11111111 1 Bit 7 of ACC=0 b15 b7 b0 00000000 0	The signed 8-bit data stored in the low-order byte of the accumulator is extended to a 16-bit data.							89 8B	4 4	2																								
									42 8B	4 4	2																								
EXTZ (Note 1)	ACC b15 b8 b7 b0 00000000	The 8-bit data stored in the low-order byte of the accumulator is extended to a 16-bit data. Bits 8 to 15 of the accumulator are set to "0".							89 AB	4 4	2																								
									42 AB	4 4	2																								
INC (Note 1)	ACC←ACC+1 or M←M+1	Increments the contents of the accumulator or memory by 1.							3A	2 2	1	E6	7 8	2				F6	7 8	2															
									42 3A	4 4	2																								
INX	X←X+1	Increments the contents of the index register X by 1.	E8	2 2	1																														
INY	Y←Y+1	Increments the contents of the index register Y by 1.	C8	2 2	1																														
JMP	ABS PCL←ADL PCH←ADM ABL PCL←ADL PCH←ADM PG←ADH (ABS) PCL←(ADM, ADL) PCH←(ADM, ADL+1) L(ABS) PCL←(ADM, ADL) PCH←(ADM, ADL+1) PG←(ADM, ADL+2) (ABS, X) PCL←(ADM, ADL+X) PCH←(ADM, ADL+X+1)	Places a new address into the program counter and jumps to that new address.																																	

Addressing mode																																																						Processor status register										
L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			Multiplied accumulation			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																																						•	•	•	N	•	•	•	•	•	Z	•
89 37	29 / 30	3	89 2D	23 / 23	4				89 3D	25 / 26	4	89 39	25 / 26	4	89 2F	25 / 26	5	89 3F	26 / 28	5																						89 23	24 / 25	3	89 33	27 / 28	3							•	•	•	N	V	•	•	•	•	Z	C
89 B7	31 / 32	3	89 AD	25 / 25	4				89 BD	27 / 28	4	89 B9	27 / 28	4	89 AF	27 / 28	5	89 BF	28 / 30	5																						89 A3	26 / 27	3	89 B3	29 / 30	5							•	•	•	N	V	•	•	•	•	Z	C
57	10 / 11	2	4D	4 / 4	3				5D	6 / 7	3	59	6 / 7	3	4F	6 / 7	4	5F	7 / 9	4																						43	5 / 6	2	53	8 / 9	2							•	•	•	N	•	•	•	•	•	Z	•
42 57	12 / 13	3	42 4D	6 / 6	4				42 5D	8 / 9	4	42 59	8 / 9	4	42 4F	8 / 9	5	42 5F	9 / 11	5																						42 43	7 / 8	3	42 53	10 / 11	3																	
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																							•	•	•	0	•	•	•	•	•	Z	•
			EE	7 / 8	3				FE	8 / 9	3																																											•	•	•	N	•	•	•	•	•	Z	•
																																																							•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
			4C	2 / 2	3										5C	4 / 5	4				6C	4 / 4	3	DC	6 / 6	3	7C	6 / 7	3																									•	•	•	•	•	•	•	•	•	•	•

Appendix 7. Machine instructions

Symbol	Function	Details	IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y			L(DIR)		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
JSR	ABS M(S)←PC_H S←S−1 M(S)←PC_L S←S−1 PC_L←AD_L PC_H←AD_M ABL M(S)←PG S←S−1 M(S)←PC_H S←S−1 M(S)←PC_L S←S−1 PC_L←AD_L PC_H←AD_M PG←AD_H (ABS, X) M(S)←PC_H S←S−1 M(S)←PC_L S←S−1 PC_L←(AD_M, AD_L+X) PC_H←(AD_M, AD_L+X+1)	Saves the contents of the program counter (also the contents of the program bank register for ABL) into the stack, and jumps to the new address.																																		
LDA (Note 1,2)	Acc←M	Enters the contents of the memory into the accumulator.				A9	2 2	2				A5	4 4	2				B5	5 6	2				B2	6 6	2	A1	7 8	2	B1	8 9	2	A7	8 8	2
						42 A9	4 4	3				42 A5	6 6	3				42 B5	7 8	3				42 B2	8 8	3	42 A1	9 10	3	42 B1	10 11	3	42 A7	10 10	3
LDM (Note 4)	M←IMM	Enters the immediate value into the memory.										64	4 5	3				74	5 6	3															
LDT	DT←IMM	Enters the immediate value into the data bank register.				89 C2	5 5	3																											
LDX (Note 11)	X←M	Enters the contents of the memory into index register X.				A2	2 2	2				A5	4 4	2							B6	5 6	2												
LDY (Note 11)	Y←M	Enters the contents of the memory into index register Y.				A0	2 2	2				A4	4 4	2				B4	5 6	2															
LSR (Note 1)	m=0 0 → [b15 ··· b0] → C m=1 0 → [b7 ··· b0] → C	Shifts the contents of the accumulator or the contents of the memory one bit to the right. The bit 0 of the accumulator or the memory is entered into the C flag. "0" is entered into bit 15 (bit 7 when the m flag is "1".)							4A	2 2	1	46	7 8	2				56	7 8	2															
									42 4A	4 4	2																								
MPY (Note 2,10)	B, A←A×M	Multiplies the contents of accumulator A and the contents of the memory. The higher order of the result of operation are entered into accumulator B, and the lower order into accumulator A.				89 09	8 8	3				89 05	10 10	3				89 15	11 12	3				89 12	12 12	3	89 01	13 14	3	89 11	14 15	3	89 07	14 14	3
MPYS (Note 2,10)	B, A←A×M (with sign)	The content of the accumulator A is multiplied by the content of memory as signed data. The result is a 32-bit data which is placed in the accumulators B (upper 16 bits of the result) and A (lower 16 bits of the result).				89 89	8 8	3				89 85	10 10	3				89 95	11 12	3				89 92	12 12	3	89 81	13 14	3	89 91	14 15	3	89 87	14 14	3
MVN (Note 7)	M(Y+k)←M(X+k) k=0 to i−1	Transmits the data block. The transmission is done from the lower order address of the block.																																	

Addressing mode																																																						Processor status register										
L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			Multiplied accumulation			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
			20	6 7	3										22	8 9	4										FC	8 9	3																									·	·	·	·	·	·	·	·	·	·	·
B7	10 11	2	AD	4 4	3				BD	6 7	3	B9	6 7	3	AF	6 7	4	BF	7 9	4																						A3	5 6	2	B3	8 9	2							·	·	·	N	·	·	·	·	·	Z	·
42 B7	12 13	3	42 AD	6 6	4				42 BD	8 9	4	42 B9	8 9	4	42 AF	8 9	5	42 BF	9 11	5																						42 A3	7 8	3	42 B3	10 11	3																	
			9C	5 6	4				9E	6 8	4																																											·	·	·	·	·	·	·	·	·	·	·
																																																						·	·	·	·	·	·	·	·	·	·	·
			AE	4 4	3							BE	6 7	3																																								·	·	·	N	·	·	·	·	·	Z	·
			AC	4 4	3				BC	6 7	3																																											·	·	·	N	·	·	·	·	·	Z	·
			4E	7 8	3				5E	8 9	3																																											·	·	·	0	·	·	·	·	·	Z	C
89 17	16 17	3	89 0D	10 10	4				89 1D	12 13	4	89 19	12 13	4	89 0F	12 13	5	89 1F	13 15	5																						89 03	11 12	3	89 13	14 15	3							·	·	·	N	·	·	·	·	·	Z	0
89 97	16 17	3	89 8D	10 10	4				89 9D	12 13	4	89 99	12 13	4	89 8F	12 13	5	89 9F	13 15	5																						89 83	11 12	3	89 93	14 15	3							·	·	·	N	·	·	·	·	·	Z	0
																																																54 $\frac{i}{2}$	5 + ×	3 7				·	·	·	·	·	·	·	·	·	·	·
																																																$\frac{i}{2}$	5 + ×	7														

Appendix 7. Machine instructions

Symbol	Function	Details	Addressing mode																																
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y			L(DIR)		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
MVP (Note 8)	M(Y–K)←M(X–k) k=0~i–1	Transmits the data block. Transmission is done form the higher order address of the data block.																																	
NOP	PC←PC+1	Advances the program counter, but performs nothing else.	EA	2 2	1																														
ORA (Note 1,2)	Acc←Acc∨M	Logical sum per bit of the contents of the accumulator and the contents of the memory is obtained. The result is entered into the accumulator.				09	2 2	2				05	4 4	2				15	5 6	2				12	6 6	2	01	7 8	2	11	8 9	2	07	8 8	2
						42 09	4 4	3				42 05	6 6	3				42 15	7 8	3				42 12	8 8	3	42 01	9 10	3	42 11	10 11	3	42 07	10 10	3
PEA	M(S)←IMM_2 S←S–1 M(S)←IMM_1 S←S–1	The 3rd and the 2nd bytes of the instruction are saved into the stack, in this order.																																	
PEI	M(S)←M((DPR)+IMM+1) S←S–1 M(S)←M((DPR)+IMM) S←S–1	Specifies 2 sequential bytes in the direct page in the 2nd byte of the instruction, and saves the contents into the stack.																																	
PER	EAR←PC+IMM_2, IMM_1 M(S)←EAR_H S←S–1 M(S)←EAR_L S←S–1	Regards the 2nd and 3rd bytes of the instruction as 16-bit numerals, adds them to the program counter, and saves the result into the stack.																																	
PHA	m=0 M(S)←A_H S←S–1 M(S)←A_L S←S–1 m=1 M(S)←A_L S←S–1	Saves the contents of accumulator A into the stack.																																	
PHB	m=0 M(S)←B_H S←S–1 M(S)←B_L S←S–1 m=1 M(S)←B_L S←S–1	Saves the contents of accumulator B into the stack.																																	
PHD	M(S)←DPR_H S←S–1 M(S)←DPR_L S←S–1	Saves the contents of the direct page register into the stack.																																	
PHG	M(S)←PG S←S–1	Saves the contents of the program bank register into the stack.																																	
PHP	M(S)←PS_H S←S–1 M(S)←PS_L S←S–1	Saves the contents of the program status register into the stack.																																	
PHT	M(S)←DT S←S–1	Saves the contents of the data bank register into the stack.																																	

Addressing mode: L(DIR),Y (op n #)	ABS (op n #)	ABS,b (op n #)	ABS,X (op n #)	ABS,Y (op n #)	ABL (op n #)	ABL,X (op n #)	(ABS) (op n #)	L(ABS) (op n #)	(ABS,X) (op n #)	STK (op n #)	REL (op n #)	DIR,b,R (op n #)	ABS,b,R (op n #)	SR (op n #)	(SR),Y (op n #)	BLK (op n #)	Multiplied accumulation (op n #)	PSR 10 (IPL)	9 (IPL)	8 (IPL)	7 N	6 V	5 m	4 x	3 D	2 I	1 Z	0 C
																44 9+ i/2 ×7 / 9+ i/2 ×7 3		•	•	•	•	•	•	•	•	•	•	•
																		•	•	•	•	•	•	•	•	•	•	•
17 10/11 2	0D 4/4 3		1D 6/7 3	19 6/7 3	0F 6/7 4	1F 7/9 4								03 5/6 2	13 8/9 2			•	•	•	N	•	•	•	•	•	Z	•
42 17 12/13 3	42 0D 6/6 4		42 1D 8/9 4	42 19 8/9 4	42 0F 8/9 5	42 1F 9/11 5								42 03 7/8 3	42 13 10/11 3													
										F4 5/7 3								•	•	•	•	•	•	•	•	•	•	•
										D4 6/7 2								•	•	•	•	•	•	•	•	•	•	•
										62 5/6 3								•	•	•	•	•	•	•	•	•	•	•
										48 4/4 1								•	•	•	•	•	•	•	•	•	•	•
										42 48 6/6 2								•	•	•	•	•	•	•	•	•	•	•
										0B 4/4 1								•	•	•	•	•	•	•	•	•	•	•
										4B 3/3 1								•	•	•	•	•	•	•	•	•	•	•
										08 4/4 1								•	•	•	•	•	•	•	•	•	•	•
										8B 3/3 1								•	•	•	•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																																
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y			L(DIR)		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
PHX	x=0 $M(S) \leftarrow X_H$ $S \leftarrow S-1$ $M(S) \leftarrow X_L$ $S \leftarrow S-1$ x=1 $M(S) \leftarrow X_L$ $S \leftarrow S-1$	Saves the contents of the index register X into the stack.																																	
PHY	x=0 $M(S) \leftarrow Y_H$ $S \leftarrow S-1$ $M(S) \leftarrow Y_L$ $S \leftarrow S-1$ x=1 $M(S) \leftarrow Y_L$ $S \leftarrow S-1$	Saves the contents of the index register Y into the stack.																																	
PLA	m=0 $S \leftarrow S+1$ $A_L \leftarrow M(S)$ $S \leftarrow S+1$ $A_H \leftarrow M(S)$ m=1 $S \leftarrow S+1$ $A_L \leftarrow M(S)$	Restores the contents of the stack on the accumulator A.																																	
PLB	m=0 $S \leftarrow S+1$ $B_L \leftarrow M(S)$ $S \leftarrow S+1$ $B_H \leftarrow M(S)$ m=1 $S \leftarrow S+1$ $B_L \leftarrow M(S)$	Restores the contents of the stack on the accumulator B.																																	
PLD	$S \leftarrow S+1$ $DPR_L \leftarrow M(S)$ $S \leftarrow S+1$ $DPR_H \leftarrow M(S)$	Restores the contents of the stack on the direct page register.																																	
PLP	$S \leftarrow S+1$ $PS_L \leftarrow M(S)$ $S \leftarrow S+1$ $PS_H \leftarrow M(S)$	Restores the contents of the stack on the processor status register.																																	
PLT	$S \leftarrow S+1$ $DT \leftarrow M(S)$	Restores the contents of the stack on the data bank register.																																	
PLX	x=0 $S \leftarrow S+1$ $X_L \leftarrow M(S)$ $S \leftarrow S+1$ $X_H \leftarrow M(S)$ x=1 $S \leftarrow S+1$ $X_L \leftarrow M(S)$	Restores the contents of the stack on the index register X.																																	
PLY	x=0 $S \leftarrow S+1$ $Y_L \leftarrow M(S)$ $S \leftarrow S+1$ $Y_H \leftarrow M(S)$ x=1 $S \leftarrow S+1$ $Y_L \leftarrow M(S)$	Restores the contents of the stack on the index register Y.																																	

Addressing mode																																																									Processor status register										
L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			Multiplied accumulation			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																														DA	4 4	1																						·	·	·	·	·	·	·	·	·	·	·
																														5A	4 4	1																						·	·	·	·	·	·	·	·	·	·	·
																														68	5 5	1																						·	·	·	N	·	·	·	·	·	Z	·
																														42 68	7 7	2																						·	·	·	N	·	·	·	·	·	Z	·
																														2B	5 5	1																						·	·	·	·	·	·	·	·	·	·	·
																														28	6 6	1																						Value saved in stack.										
																														AB	6 6	1																						·	·	·	N	·	·	·	·	·	Z	·
																														FA	5 5	1																						·	·	·	N	·	·	·	·	·	Z	·
																														7A	5 5	1																						·	·	·	N	·	·	·	·	·	Z	·

Symbol	Function	Details	Addressing mode																																
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y			L(DIR)		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
PSH (Note 5)	M(S)←A, B, X···	Saves the registers among accumulator, index register, direct page register, data bank register, program bank register, or processor status register, specified by the bit pattern of the second byte of the instruction into the stack.																																	
PUL (Note 6)	A, B, X...←M(S)	Restores the contents of the stack to the registers among accumulator, index register, direct page register, data bank register, or processor status register, specified by the bit pattern of the second byte of the instruction.																																	
RLA (Note 12)	m=0 i bit rotate left ← b15 ··· b0 ← m=1 i bit rotate left ← b7 ··· b0 ←	Rotates the contents of the accumulator A, i bits to the left.				89 49	6+i 6+i	3																											
RMPA (Note 15)	m=0 Repeat B, A←B, A+M(DT, X)× M(DT, Y)(with sign) X←X+2 Y←Y+2 i←i−1 Until i=0 m=1 Repeat BL,AL←BL,AL+M(DT,X)× M(DT, Y)(with sign) X←X+1 Y←Y+1 i←i−1 Until i=0	Performs signed multiplication of the data in the memory specified by index register X and data bank register, and the data in the memory specified by index register Y and data bank register. The multiplication result is added as binary addition to the data of which high-order is the contents of accumulator B, and of which low-order is the contents of accumulator A. The high-order result is stored in accumulator B, and the low-order result is stored in accumulator A again. After the addition, when the data length flag (m) is "0", each of the contents of index register X and index register Y is incremented by 2. Additionally, the number of multiplication and addition is decremented by 1. When the data length flag (m) is "1", each of the contents of index register X and index register Y is incremented by 1. Additionally, the number of multiplication and addition is decremented by 1. The above multiplication and addition are repeated until the number of multiplication and addition is "0".																																	
ROL (Note 1)	m=0 ← b15 ··· b0 ← C ← m=1 ← b7 ··· b0 ← C ←	Links the accumulator or the memory to C flag, and rotates result to the left by 1 bit.							2A 42 2A	2 / 2 4 / 4	1 2	26	7 / 8	2				36	7 / 8	2															
ROR (Note 2)	m=0 → C → b15 ··· b0 → m=1 → C → b7 ··· b0 →	Links the accumulator or the memory to C flag, and rotates result to the right by 1 bit.							6A 42 6A	2 / 2 4 / 4	1 2	66	7 / 8	2				76	7 / 8	2															

Addressing mode																																																						Processor status register										
L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			Multiplied accumulation			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																														EB	$11 + 2i_1 + i_2$ $11 + 2i_1 + i_2$	2																						·	·	·	·	·	·	·	·	·	·	·
																														FB	$12 + 3i_1 + 4i_2$ $12 + 3i_1 + 4i_2$	2																						If restored the contents of PS, it becomes its value. And the other case is no change.										
																																																					·	·	·	·	·	·	·	·	·	·	·
																																																		89 E2	$6 + i \times 16$ $6 + i \times 16$	3	·	·	·	N	V	·	·	·	·	Z	C
			2E	7 8	3				3E	8 9	3																																											·	·	·	N	·	·	·	·	·	Z	C
			6E	7 8	3				7E	8 9	3																																										·	·	·	N	·	·	·	·	·	Z	C

Symbol	Function	Details	Addressing mode																																
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y			L(DIR)		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
RTI	S←S+1 PSL←M(S) S←S+1 PSH←M(S) S←S+1 PCL←M(S) S←S+1 PCH←M(S) S←S+1 PG←M(S)	Returns from the interruption routine.	40	9 9	1																														
RTL	S←S+1 PCL←M(S) S←S+1 PCH←M(S) S←S+1 PG←M(S)	Returns from the subroutine. The contents of the program bank register are also restored.	6B	7 7	1																														
RTS	S←S+1 PCL←M(S) S←S+1 PCH←M(S)	Returns from the subroutine. The contents of the program bank register are not restored.	60	5 5	1																														
SBC (Note 1, 2)	ACC, C←ACC–M–$\overline{C}$	Subtracts the contents of the memory and the borrow from the contents the accumulator.				E9	2 2	2				E5	4 4	2				F5	5 6	2				F2	6 6	2	E1	7 8	2	F1	8 9	2	E7	8 8	2
						42 E9	4 4	3				42 E5	6 6	3				42 F5	7 8	3				42 F2	8 8	3	42 E1	9 10	3	42 F1	10 11	3	42 E7	10 10	3
SEB (Note 4)	Mb ←1	Makes the contents of the specified bit in the memory "1".													04	8 9	3																		
SEC	C←1	Makes the contents of the C flag "1".	38	2 2	1																														
SEI	I←1	Makes the contents of the I flag "1".	78	2 2	1																														
SEM	m←1	Makes the contents of the m flag "1".	F8	2 2	1																														
SEP	PSb←1	Set the specified bit of the processor status register's lower byte (PSL) to "1".				E2	3 3	2																											
STA (Note 1)	M←ACC	Stores the contents of the accumulator into the memory.										85	4 5	2				95	5 5	2				92	7 8	2	81	7 8	2	91	7 8	2	87	9 10	2
												42 85	6 7	3				42 95	7 7	3				42 92	9 10	3	42 81	9 10	3	42 91	9 10	3	42 87	11 12	3
STP		Stops the oscillation of the oscillator.	DB	– –	1																														
STX	M←X	Stores the contents of the index register X into the memory.										86	4 5	2							96	5 5	2												
STY	M←Y	Stores the contents of the index register Y into the memory.										84	4 5	2				94	5 5	2															
TAD	DPR←A	Transmits the contents of the accumulator A to the direct page register.	5B	2 2	1																														
TAS	S←A	Transmits the contents of the accumulator A to the stack pointer.	1B	2 2	1																														

Addressing mode (each cell: op, n, #)

L(DIR),Y	ABS	ABS,b	ABS,X	ABS,Y	ABL	ABL,X	(ABS)	L(ABS)	(ABS,X)	STK	REL	DIR,b,R	ABS,b,R	SR	(SR),Y	BLK	Multiplied accumulation
F7, 10/11, 2	ED, 4/4, 3		FD, 6/7, 3	F9, 6/7, 3	EF, 6/7, 4	FF, 7/9, 4								E3, 5/6, 2	F3, 8/9, 2		
42 F7, 12/13, 3	42 ED, 6/6, 4		42 FD, 8/9, 4	42 F9, 8/9, 4	42 EF, 8/9, 5	42 FF, 9/11, 5								42 E3, 7/8, 3	42 F3, 10/11, 3		
		0C, 9/11, 4															
97, 9/10, 2	8D, 5/6, 3		9D, 5/5, 3	99, 5/5, 3	8F, 6/8, 4	9F, 7/9, 4								83, 5/5, 2	93, 8/9, 2		
42 97, 11/12, 3	42 8D, 7/8, 4		42 9D, 7/8, 4	42 99, 7/8, 4	42 8F, 8/10, 5	42 9F, 9/10, 5								42 83, 7/7, 3	42 93, 10/11, 3		
	8E, 5/6, 3																
	8C, 5/6, 3																

Processor status register

10	9	8	7	6	5	4	3	2	1	0
IPL			N	V	m	x	D	I	Z	C
Value saved in stack.										
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	N	V	•	•	•	•	Z	C
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	1
•	•	•	•	•	•	•	•	1	•	•
•	•	•	•	•	1	•	•	•	•	•
•	•	•	Specified flag becomes "1".							
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•

Symbol	Function	Details	IMP op	IMP n	IMP #	IMM op	IMM n	IMM #	A op	A n	A #	DIR op	DIR n	DIR #	DIR,b op	DIR,b n	DIR,b #	DIR,X op	DIR,X n	DIR,X #	DIR,Y op	DIR,Y n	DIR,Y #	(DIR) op	(DIR) n	(DIR) #	(DIR,X) op	(DIR,X) n	(DIR,X) #	(DIR),Y op	(DIR),Y n	(DIR),Y #	L(DIR) op	L(DIR) n	L(DIR) #
TAX	X←A	Transmits the contents of the accumulator A to the index register X.	AA	2 / 2	1																														
TAY	Y←A	Transmits the contents of the accumulator A to the index register Y.	A8	2 / 2	1																														
TBD	DPR←B	Transmits the contents of the accumulator B to the direct page register.	42 5B	4 / 4	2																														
TBS	S←B	Transmits the contents of the accumulator B to the stack pointer.	42 1B	4 / 4	2																														
TBX	X←B	Transmits the contents of the accumulator B to the index register X.	42 AA	4 / 4	2																														
TBY	Y←B	Transmits the contents of the accumulator B to the index register Y.	42 A8	4 / 4	2																														
TDA	A←DPR	Transmits the contents of the direct page register to the accumulator A.	7B	2 / 2	1																														
TDB	B←DPR	Transmits the contents of the direct page register to the accumulator B.	42 7B	4 / 4	2																														
TSA	A←S	Transmits the contents of the stack pointer to the accumulator A.	3B	2 / 2	1																														
TSB	B←S	Transmits the contents of the stack pointer to the accumulator B.	42 3B	4 / 4	2																														
TSX	X←S	Transmits the contents of the stack pointer to the index register X.	BA	2 / 2	1																														
TXA	A←X	Transmits the contents of the index register X to the accumulator A.	8A	2 / 2	1																														
TXB	B←X	Transmits the contents of the index register X to the accumulator B.	42 8A	4 / 4	2																														
TXS	S←X	Transmits the contents of the index register X to the stack pointer.	9A	2 / 2	1																														
TXY	Y←X	Transmits the contents of the index register X to the index register Y.	9B	2 / 2	1																														
TYA	A←Y	Transmits the contents of the index register Y to the accumulator A.	98	2 / 2	1																														
TYB	B←Y	Transmits the contents of the index register Y to the accumulator B.	42 98	4 / 4	2																														
TYX	X←Y	Transmits the contents of the index register Y to the index register X.	BB	2 / 2	1																														
WIT		Stops the ϕCPU, ϕBIU.	CB	– / –	1																														
XAB	A⇔B	Exchanges the contents of the accumulator A and the contents of the accumulator B.	89 28	5 / 5	2																														

Addressing mode																																																					
L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			Multiplied accumulation		
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#

Processor status register										
10	9	8	7	6	5	4	3	2	1	0
IPL			N	V	m	x	D	I	Z	C
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	N	•	•	•	•	•	Z	•

Notes for machine instructions table

A number of cycles on the upper row is the number when fetching instructions at 2 ϕ access in low-speed running under the condition of f (XIN) ≦ 25 MHz. A number of cycles on the lower row is the number when fetching instructions at 3 ϕ access in high-speed running under the condition of 25 MHz < f (XIN) ≦ 40 MHz.
The cycles' number of addressing modes concerning DPR is the number of the case of DPR="0". When DPR ≠ "0", the number of cycles is incremented by 1.
The number of cycles shown in the table differs according to the bytes fetched into the instruction queue buffer, or according to whether the memory accessed is odd address or even address. It also differs when the external area is accessed by BYTE="H". This table shows the fastest number of cycles for each instruction.

Note 1. The operation code at the upper row is used for accumulator A, and the operation at the lower row is used for accumulator B.

Note 2. When setting flag m=0 to handle the data as 16-bit data in the immediate addressing mode, the number of bytes increments by 1.

Note 3. The operation code on the upper row is used for branching in the range of –128 to +127, and the operation code on the lower row is used for branching in the range of –32768 to +32767.

Note 4. When handling 16-bit data with flag m=0, the byte in the table is incremented by 1.

Note 5.

Type of register	A	B	X	Y	DPR	DT	PG	PS
Number of cycles	2	2	2	2	2	1	1	2

The number of cycles corresponding to the register to be pushed are added. The number of cycles when no pushing is done is 11. i_1 indicates the number of registers among A, B, X, Y, DPR, and PS to be saved. i_2 indicates the number of registers among DT and PG to be saved.

Note 6.

Type of register	A	B	X	Y	DPR	DT	PS
Number of cycles	3	3	3	3	4	3	3

The number of cycles corresponding to the register to be pulled are added. The number of cycles when no pulling is done is 12. i_1 indicates the number of registers among A, B, X, Y, DT, and PS to be restored. i_2=1 when DPR is to be restored, and i_2=0 when DPR is not to be restored.

Note 7. The number of cycles is the case when the number of bytes to be transferred is even.
When the number of bytes to be transferred is odd, the number is calculated as;

$$5 + (i / 2) \times 7 + 6$$

Note that, (i/2) shows the integer part when i is divided by 2.

Note 8. The number of cycles is the case when the number of bytes to be transfered is even.
When the number of bytes to be transfered is odd, the number is calculated as;

$$9 + (i / 2) \times 7 + 8$$

Note that, (i/2) shows the integer part when i is divided by 2.

Note 9. The number of cycles is the case in the 16-bit ÷ 8-bit operation. The number of cycles is incremented by 8 for 32-bit ÷ 16-bit operation.

Note 10. The number of cycles is the case in the 8-bit × 8-bit operation. The number of cycles is incremented by 4 for 16-bit × 16-bit operation.

Note 11. When setting flag x=0 to handle the data as 16-bit data in the immediate addressing mode, the number of bytes increments by 1.

Note 12. When flag m is 0, the byte in the table is incremented by 1.

Note 13. When a zero division interrupt occurs, the number of cycles is the number when it does not occur decremented by 3. It is regardless of the data length.

Note 14. When a zero division interrupt occurs, the number of cycles is the number when it does not occur decremented by 5. It is regardless of the data length.

Note 15. The number of cycles is the case when flag m is 1.
When flag m=0, the number is calculated as;

$$6 + 20 \times i$$

Appendix 7. Machine instructions

Symbols in machine instructions table

Symbol	Description	Symbol	Description
IMP	Implied addressing mode	∀	Exclusive OR
IMM	Immediate addressing mode	—	Negation
A	Accumulator addressing mode	←	Movement to the arrow direction
DIR	Direct addressing mode	→	
DIR, b	Direct bit addressing mode	⇔	
DIR, X	Direct indexed X addressing mode	ACC	Accumulator
DIR, Y	Direct indexed Y addressing mode	A	Accumulator A
(DIR)	Direct indirect addressing mode	A_H	Accumulator A's upper 8 bits
(DIR, X)	Direct indexed X indirect addressing mode	A_L	Accumulator A's lower 8 bits
(DIR), Y	Direct indirect indexed Y addressing mode	B	Accumulator B
L (DIR)	Direct indirect long addressing mode	B_H	Accumulator B's upper 8 bits
L (DIR), Y	Direct indirect long indexed Y addressing mode	B_L	Accumulator B's lower 8 bits
ABS	Absolute addressing mode	X	Index register X
ABS, b	Absolute bit addressing mode	X_H	Index register X's upper 8 bits
ABS, X	Absolute indexed X addressing mode	X_L	Index register X's lower 8 bits
ABS, Y	Absolute indexed Y addressing mode	Y	Index register Y
ABL	Absolute long addressing mode	Y_H	Index register Y's upper 8 bits
ABL, X	Absolute long indexed X addressing mode	Y_L	Index register Y's lower 8 bits
(ABS)	Absolute indirect addressing mode	S	Stack pointer
L (ABS)	Absolute indirect long addressing mode	PC	Program counter
(ABS, X)	Absolute indexed X indirect addressing mode	PC_H	Program counter's upper 8 bits
STK	Stack addressing mode	PC_L	Program counter's lower 8 bits
REL	Relative addressing mode	PG	Program bank register
DIR, b R	Direct bit relative addressing mode	DT	Data bank register
ABS, b, R	Absolute bit relative addressing mode	DPR	Direct page register
SR	Stack pointer relative addressing mode	DPR_H	Direct page register's upper 8 bits
(SR), Y	Stack pointer relative indirect indexed Y addressing mode	DPR_L	Direct page register's lower 8 bits
		PS	Processor status register
BLK	Block transfer addressing mode	PS_H	Processor status register's upper 8 bits
Multiplied accumulation	Multiply and accumulate addressing mode	PS_L	Processor status register's lower 8 bits
		PS_b	Bit in processor status register
op	Operation code	M	Memory
n	Number of cycle	M(S)	Contents of memory at address indicated by stack pointer
#	Number of byte		
C	Carry flag	M_b	Bit in memory location
Z	Zero flag	AD_H	Value of 24-bit address's upper 8-bit (A_{23}–A_{16})
I	Interrupt disable flag	AD_M	Value of 24-bit address's middle 8-bit (A_{15}–A_8)
D	Decimal operation mode flag	AD_L	Value of 24-bit address's lower 8-bit (A_7–A_0)
x	Index register length selection flag	IMM	Immediate value
m	Data length selection flag	EAR	Executed address (16 bits)
V	Overflow flag	EAR_H	Upper 8-bit address executed
N	Negative flag	EAR_L	Lower 8-bit address executed
IPL	Processor interrupt priority level	bn	Bit position of accumulator or memory indicated by n
+	Addition		
−	Subtraction	i	Number of transfer byte, rotation or repeated operation
×	Multiplication		
÷	Division	i_1, i_2	Number of registers pushed or pulled
∧	Logical AND		
∨	Logical OR		

Appendix 8. Examples of noise immunity improvement

Generally effective examples of noise immunity improvements are described below. Although the effect of these countermeasure depends on each system, refer to the following when an noise-related problem occurs.

1. Short wiring length

The wiring on a printed circuit board may function as an antenna which feeds noise into the microcomputer. The shorter the total wiring length (by mm unit), the less possibility of noise insertion into the microcomputer.

(1) Wiring for $\overline{\text{RESET}}$ pin

Make the length of wiring connected to $\overline{\text{RESET}}$ pin as short as possible.
In particular, connect a capacitor between $\overline{\text{RESET}}$ pin and Vss pin with the shortest possible wiring (within 20 mm).

Reason: If noise is input to $\overline{\text{RESET}}$ pin, the microcomputer restarts operation before the internal state of the microcomputer is completely initialized. This may cause a program runaway.

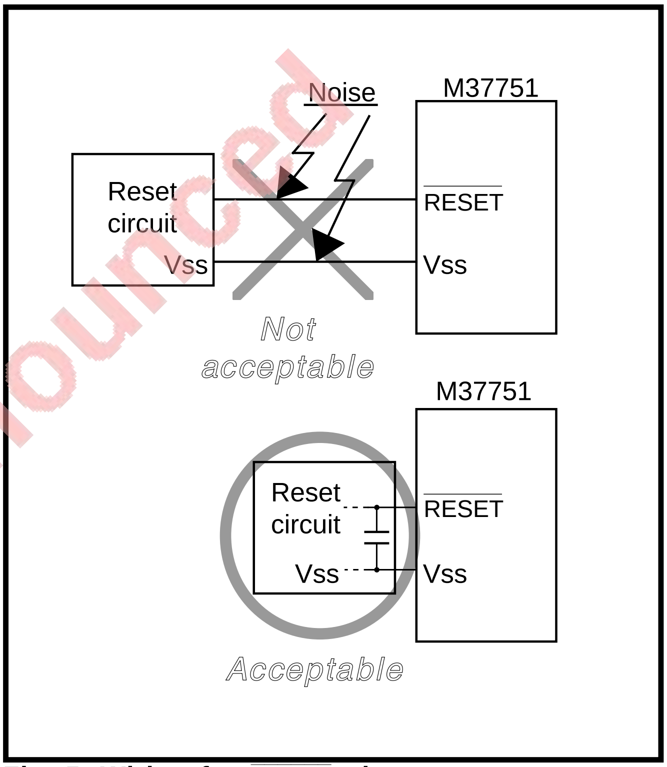


Fig. 5. Wiring for $\overline{\text{RESET}}$ pin

(2) Wiring for clock input/output pins

- Make the length of wiring connected to the clock input/output pins as short as possible.
- Make the length of wiring between the grounding lead of the capacitor, which is connected to the oscillator and Vss pin of the microcomputer, as short as possible (within 20 mm).
- Separate the Vss pattern for oscillation from all other Vss patterns. (Refer to Figure 14.)

Reason: The microcomputer's operation synchronizes with a clock generated by the oscillation circuit.
If noise enters clock I/O pins, clock waveforms may be deformed. This may cause a malfunction or a program runaway.
Also, if the noise causes a potential difference between the Vss level of the microcomputer and the Vss level of an oscillator, the correct clock will not be input in the microcomputer.

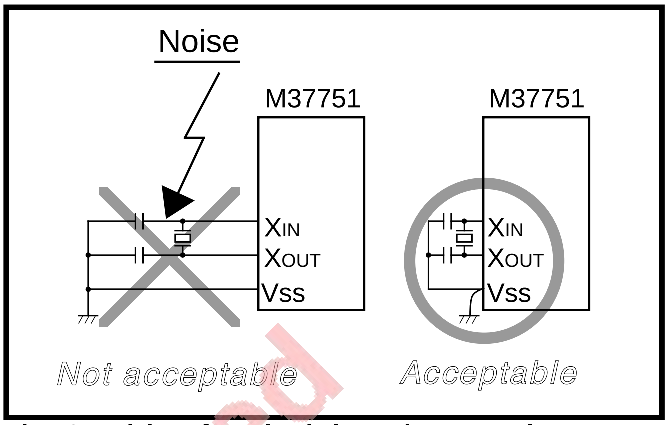


Fig. 6. Wiring for clock input/output pins

(3) Wiring for CNVss pin

Connect CNVss pin to Vss pin with the shortest possible wiring.

Reason: The processor mode of the microcomputer is influenced by a potential at CNVss pin when CNVss and Vss pins are connected.
If the noise causes a potential difference between CNVss and Vss pins, the processor mode may become unstable. This may cause a microcomputer malfunction or a program runaway.

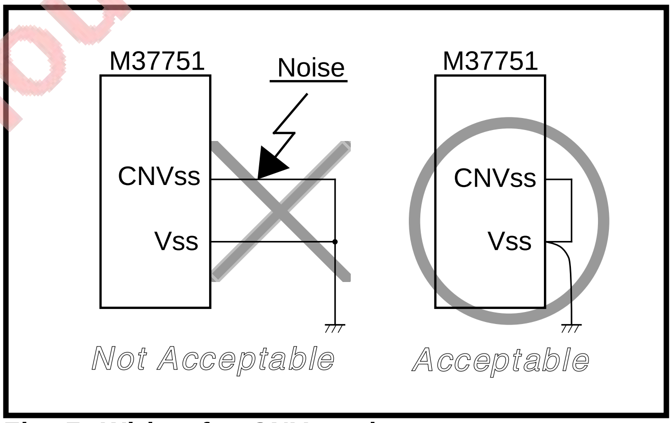


Fig. 7. Wiring for CNVss pin

(4) Wiring for CNVss (V_{PP}) pin of built-in PROM version

< In single-chip or memory expansion modes>

- Connect CNVss (V_{PP}) to Vss pin of the microcomputer with the shortest possible wiring.
- If the above countermeasure can not be taken, insert an approximate 5 kΩ resistor between CNVss (V_{PP}) and Vss pins and be sure to make the distance between the resistor and CNVss (V_{PP}) pin as short as possible.

< In microprocessor mode>

- Connect CNVss (V_{PP}) pin to Vcc pin with the shortest possible wiring.

Reason: CNVss (V_{PP}) pin is connected to the internal ROM in the low-impedance state. (Noise is easily fed to the pin in this condition.)
If noise enters the CNVss (V_{PP}) pin, incorrect instruction codes or data is fetched from the built-in PROM. This may cause a program runaway.

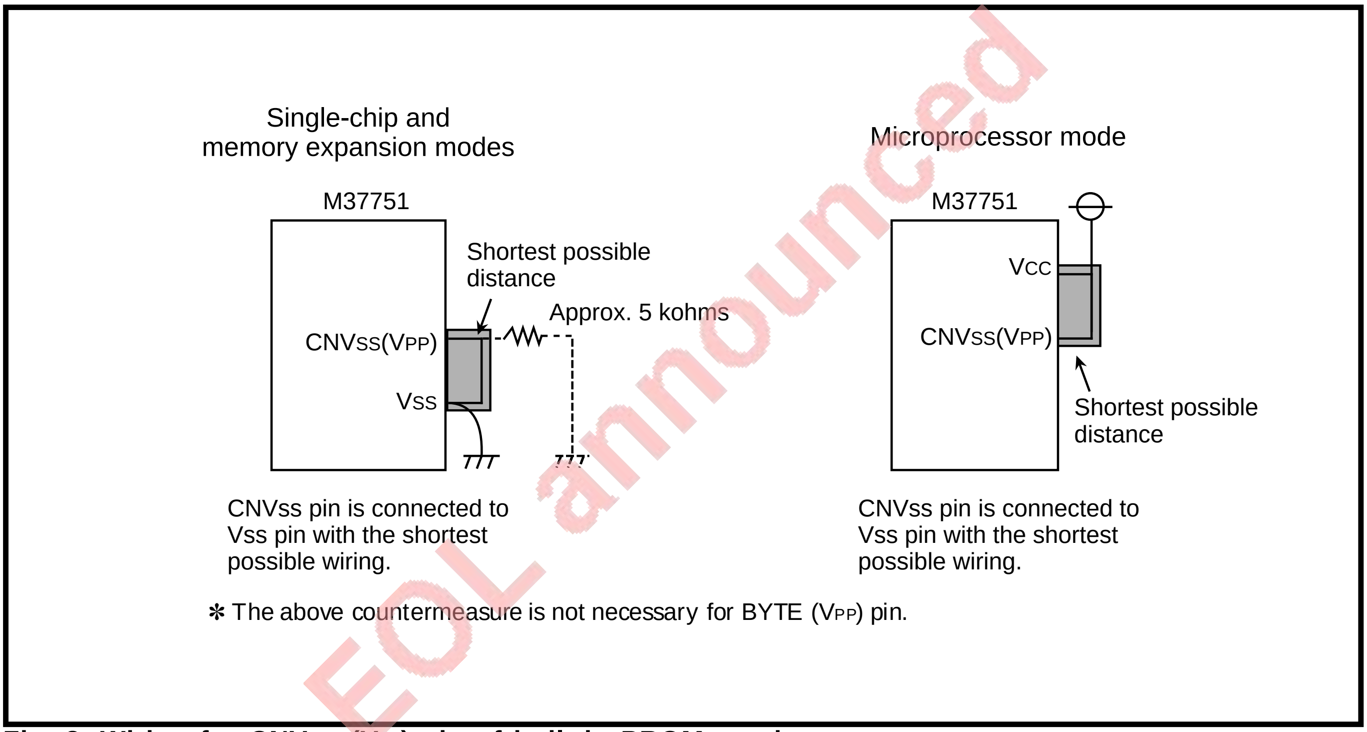


Fig. 8. Wiring for CNVss (V_{PP}) pin of built-in PROM version

Appendix 8. Examples of noise immunity improvement

2. Connection of bypass capacitor between Vss and Vcc lines

Connect an approximate 0.1 μF bypass capacitor as follows:

- Connect a bypass capacitor between the Vss and Vcc pins, at equal lengths.
- The wiring connecting the bypass capacitor between the Vss and Vcc pins should be as short as possible.
- Use thicker wiring for the Vss and Vcc lines than the other signal lines.

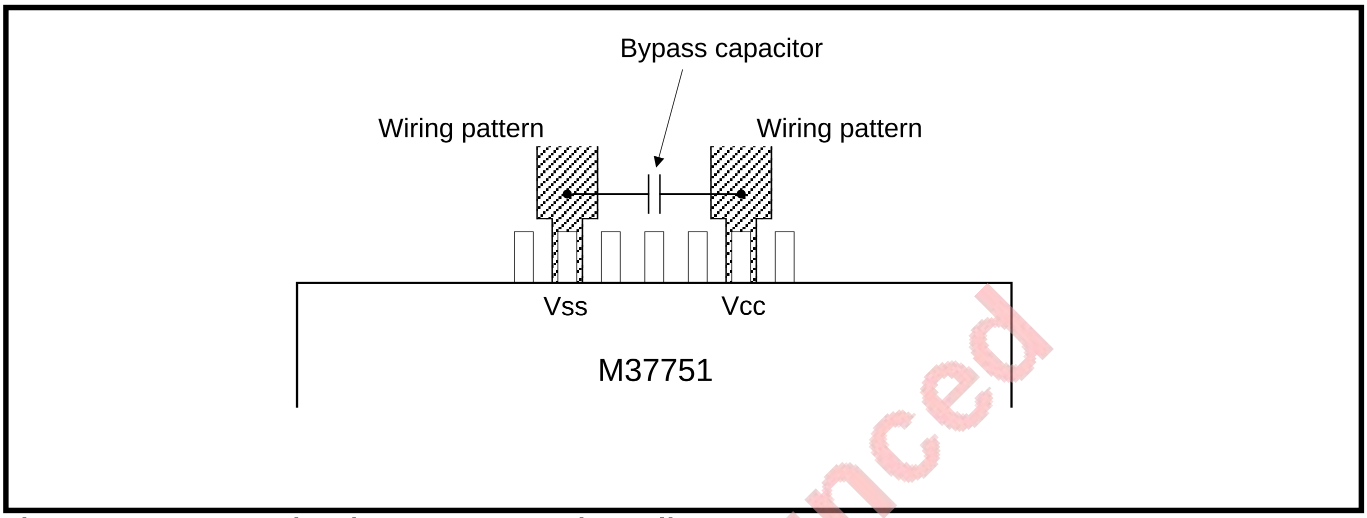


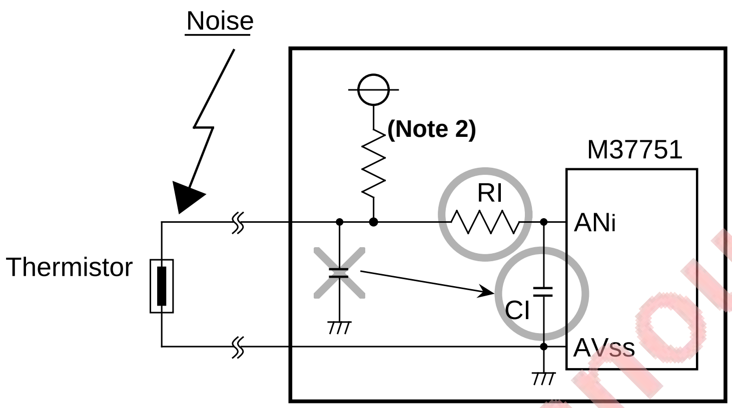
Fig. 9. Bypass capacitor between Vss and Vcc lines

3. Wiring for analog input pins, analog power source pins, etc.

(1) Processing analog input pins

- Connect a resistor to the analog signal line, which is connected to an analog input pin, and make the connection as close to the microcomputer as possible.
- Connect a capacitor between the analog input pin and AVss pin, as close to the AVss pin as possible.

Reason: A signal which is input to the analog input pin is usually an output signal from a sensor. The sensor which measures changes in status tends to be installed far from the microcomputer printed circuit board. The result is long wiring that becomes an antenna which picks up noise and feeds it into the microcomputer analog input pin.
If a capacitor between an analog input pin and AVss pin is grounded far away from AVss pin, noise on the GND line may enter the microcomputer through the capacitor.



Reference value
RI : Approximately 100 to 1000 Ω
CI : Approximately 100 to 1000 pF

Notes 1 : Design an external circuit for ANi pin so that charge/discharge is available within 1 cycle of ϕ_{AD}.

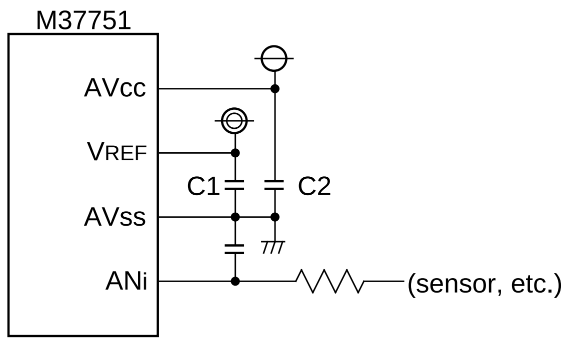
2 : This resistor and thermistor are used to divide resistance.

Fig. 10. Example of noise immunity improvement using thermistor

(2) Processing analog power source pins, etc.

- Use independent power sources for Vcc, AVcc and V_{REF} pins.
- Insert capacitors between the AVcc and AVss pins, and between the V_{REF} and AVss pins.

Reasons: Prevents the A-D converter from noise on the Vcc line.



Reference value
C1 ≥ 0.47 µF
C2 ≥ 0.47 µF

Note : Connect capacitors using the thickest, shortest wiring possible.

Fig. 11. Processing analog power source pins, etc.

4. Oscillator protection

The oscillator which generates the basic clock for the microcomputer operations must be protected from the affect of other signals.

(1) Distance oscillator from signal lines with large current flows

Install the microcomputer, especially the oscillator, as far as possible from signal lines which handle currents larger than the microcomputer current value tolerance.

Reason: A microcomputer is used in systems which contain signal lines for controlling motors, LEDs, thermal heads, etc. Noise occurs due to mutual inductance when a large current flows through the signal lines.

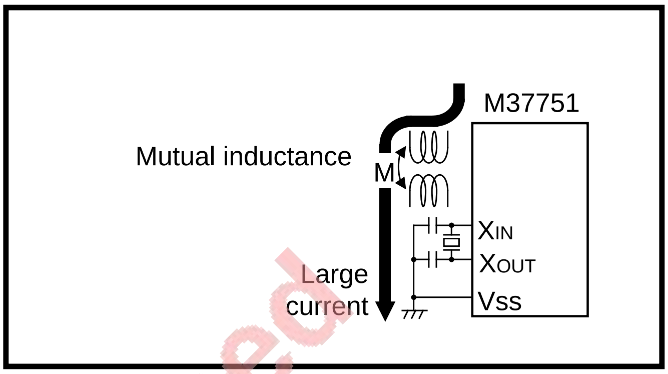


Fig. 12. Connection of signal wires where a large current flows

(2) Distance oscillator from signal lines with frequent potential level changes

- Install an oscillator and a connecting pattern of an oscillator away from signal lines in which potential levels change frequently.
- Do not cross the signal lines over the clock-related or noise-sensitive signal lines.

Reason: Signals lines with frequently changing potential levels may affect other signal lines at a rising or falling edge. In particular, if the lines cross over a clock-related signal line, clock waveforms may be deformed, which causes a microcomputer malfunction or a program runaway.

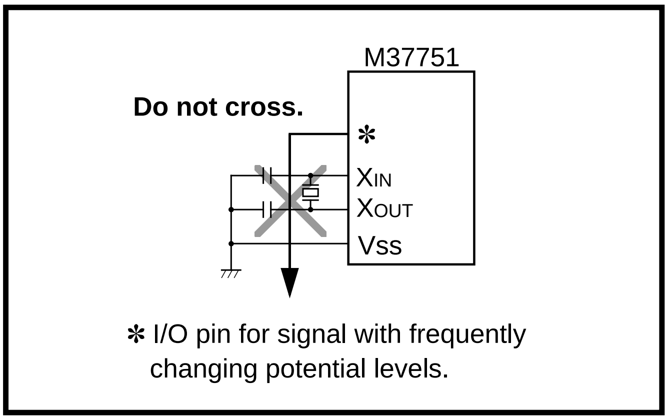


Fig. 13. Wiring of rapidly level changing signal wire

(3) Oscillator protection using Vss pattern

Print a Vss pattern on the bottom (soldering side) of a double-sided printed circuit board, under the oscillator mount position.

Connect the Vss pattern to Vss pin of the microcomputer with the shortest possible wiring, separating it from other Vss patterns.

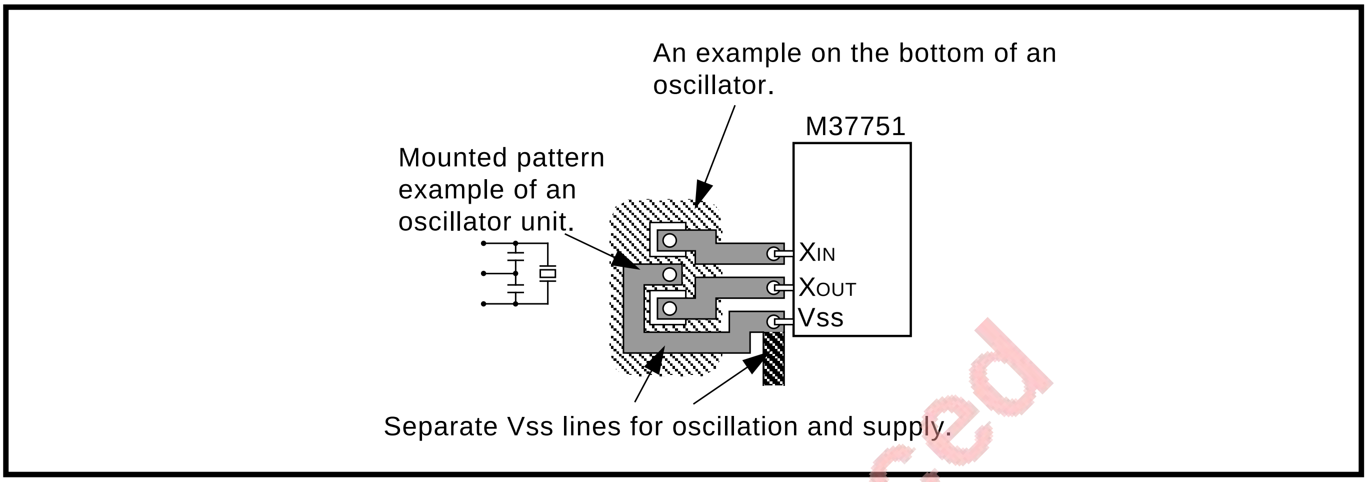


Fig. 14. Vss pattern underneath mounted oscillator

5. Setup for I/O ports

Setup I/O ports by hardware and software as follows:

<Hardware protection>

- Connect a resistor of 100 ohms or more to an I/O port in series.

<Software protection>

- As for an input port, read data several times for checking whether input levels are equal or not.
- As for an output port, since the output data may reverse because of noise, rewrite data to its port Pi register periodically.
- Rewrite data to port Pi direction registers periodically.

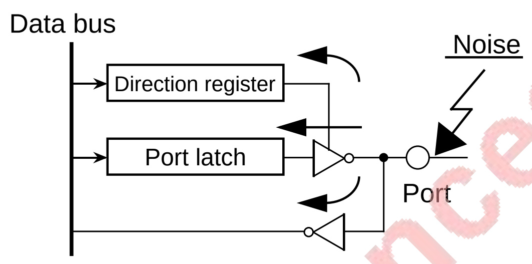


Fig. 15. Setup for I/O ports

6. Reinforcement of the power source line

- For the Vss and Vcc lines, use thicker wiring than that of other signal lines.
- When using a multilayer printed circuit board, the Vss and Vcc patterns must each be one of the middle layers.
- The following is necessary for double-sided printed circuit boards:
  On one side, the microcomputer is installed at the center, and the Vss line is looped or meshed around it. The vacant area is filled with the Vss line.
  On the opposite side, the Vcc line is wired the same as the Vss line.
  The power source lines of external devices which are connected by bus to the microcomputer must be connected to the microcomputer's power source lines with the shortest possible wiring.

Reasons: With external devices connected to the microcomputer, the levels of many of the signal lines (total external address buses: 24 bits) may change simultaneously, causing noise on the power source line.

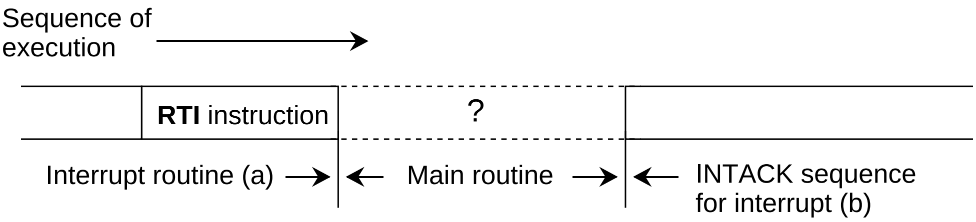
Appendix 9. Q & A

Information which may be helpful in fully utilizing the 7751 Group is provided in Q & A format.
In Q & A, as a rule, one question and its answer are summarized within one page. The upper box on each page is a question, and a box below the question is its answer. (If a question or an answer extends to two or more pages, there is a page number at the lower right corner.)
At the upper right corner of each page, the main function related to the contents of description in that page is listed.

Interrupt

Q

If an interrupt request (b) occurs while executing an interrupt routine (a), is the main routine is not executed before the INTACK sequence for the next interrupt (b) is executed after the interrupt routine (a) under execution is completed?



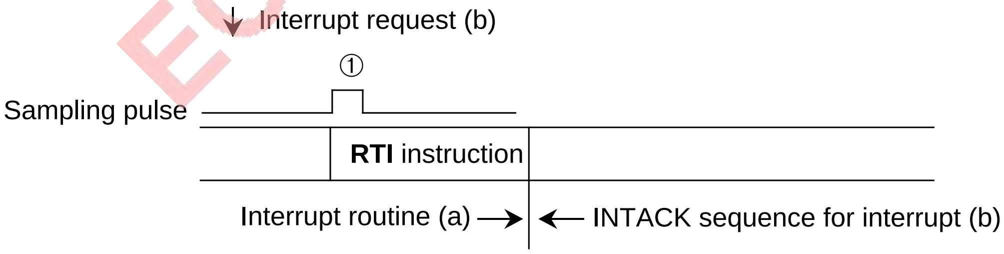
Condition

- I is cleared to "0" with the **RTI** instruction.
- The interrupt priority level of the interrupt (b) is higher than the main routine IPL.
- The interrupt priority detection time is 2 cycles of ϕ.

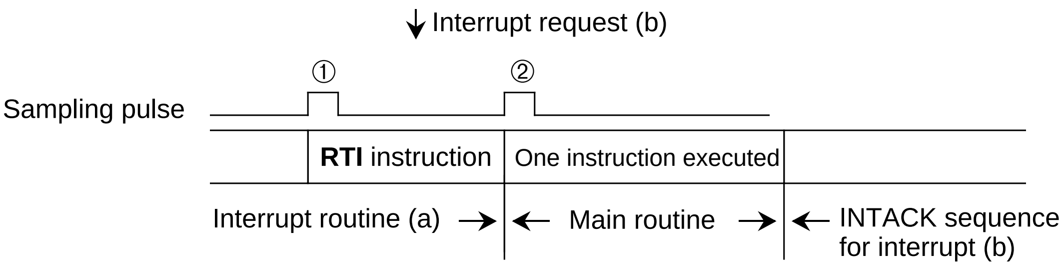
A

Sampling for interrupt requests are performed by sampling pulses generated synchronously with the CPU's op-code fetch cycles.

(1) If the next interrupt request (b) occurs before the sampling pulse (①) for the **RTI** instruction is generated, the microcomputer executes the INTACK sequence for (b) without executing the main routine (not even one instruction) because sampling is completed while executing the **RTI** instruction.



(2) If the next interrupt request (b) occurs immediately after generating of the sampling pulse ① , the microcomputer executes one instruction of the main routine before executing the INTACK sequence for (b) because the interrupt request is sampled by the next sampling pulse ②.



Interrupt

Q

There is a routine where a certain interrupt request should not be accepted (with enabled acceptance of all other interrupt requests). Accordingly, the program set the interrupt priority level select bits of the interrupt to be not accepted to "000_2" in order to disable it before executing the routine. However, the interrupt request of that interrupt has been accepted immediately after the priority level had been changed. Why did this occur and what can I do about it?

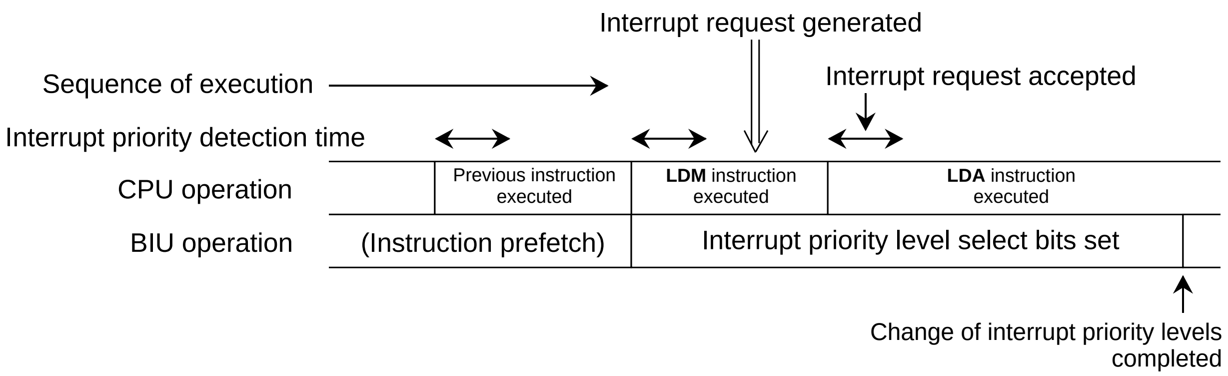
```
                         :
Interrupt request is  LDM #00H, XXXIC  ; Writes "0002" to interrupt priority level select bits.
accepted  in  this →                   ; Clears interrupt request bit to "0."
interval              LDA A,DATA       ; Instruction at the beginning of the routine that
                                         should not accept one certain interrupt request.
                         :             ;
```

A

When changing the interrupt priority level, the microcomputer can behave "as if the interrupt request is accepted immediately after it is disabled " if the next instruction (the **LDA** instruction in the above case) is already stored in the BIU's instruction queue buffer and conditions to accept the interrupt request which should not be accepted are met immediately before executing the instruction which is in that buffer.

When writing to a memory or an I/O, the CPU passes the address and data to the BIU. Then, the CPU executes the next instruction in the instruction queue buffer while the BIU is writing data into the actual address. Detection of interrupt priority level is performed at the beginning of each instruction.

In the above case, in the interrupt priority detection which is performed simultaneously with the execution of the next instruction, the interrupt priority level before changing it is detected and the interrupt request is accepted. It is because the CPU executes the next instruction before the BIU finishes changing the interrupt priority levels.



(1/2)

Interrupt

A

To prevent this problem, use software to execute the routine that should not accept a certain interrupt request after change of interrupt priority level is completed. The following shows a sample program.

[Sample program]

After an instruction which writes "000_2" to the interrupt priority level select bits, fill the instruction queue buffer with the **NOP** instruction to make the next instruction not be executed before the writing is completed.

```
    :
LDM #00H, XXXIC  ; Sets the interrupt priority level select bits to "0002."
NOP              ;
NOP              ;
NOP              ;
LDA  A,DATA      ; Instruction at the beginning of the routine that should not accept a certain
                   interrupt request
```

(2/2)

Interrupt

Q

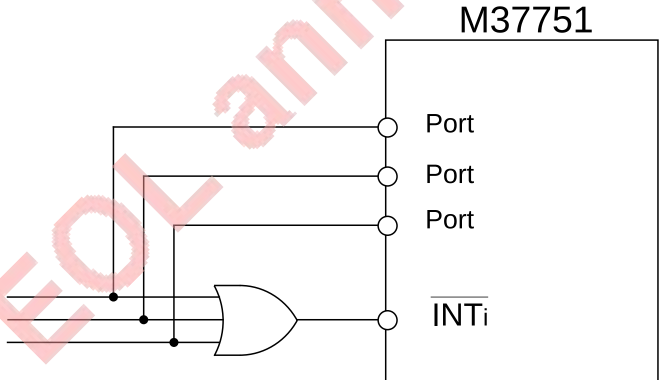
(1) Which timing of clock ϕ_1 is the external interrupts (input signals to the $\overline{INT_i}$ pin) detected?
(2) How can four or more external interrupt input pins ($\overline{INT_i}$) be used?

A

(1) In both the edge sense and level sense, external interrupt requests occur when the input signal to the $\overline{INT_i}$ pin changes its level regardless of clock ϕ_1.
In the edge sense, the interrupt request bit is set to "1" at this time.

(2) There are two methods: one uses external interrupt's level sense, and the other uses the timer's event counter mode.

① **Using external interrupt's level sense**
In hardware, input a logical sum of multiple interrupt signals (e.g., 'a', 'b', and 'c') to the $\overline{INT_i}$ pin, and input each signal to each corresponding port.
In software, check the port's input levels in the $\overline{INT_i}$ interrupt routine to determine that which of the signals 'a', 'b', and 'c' is input.



② **Using timer's event counter mode**
In hardware, input interrupt signals to the TAi_{IN} pins or TBi_{IN} pins.
In software, set the timer's operating mode to the event counter mode and a value "0000_{16}" into the timer register to the effective edge.
The timer's interrupt request occurs when an interrupt signal (selected effective edge) is input.

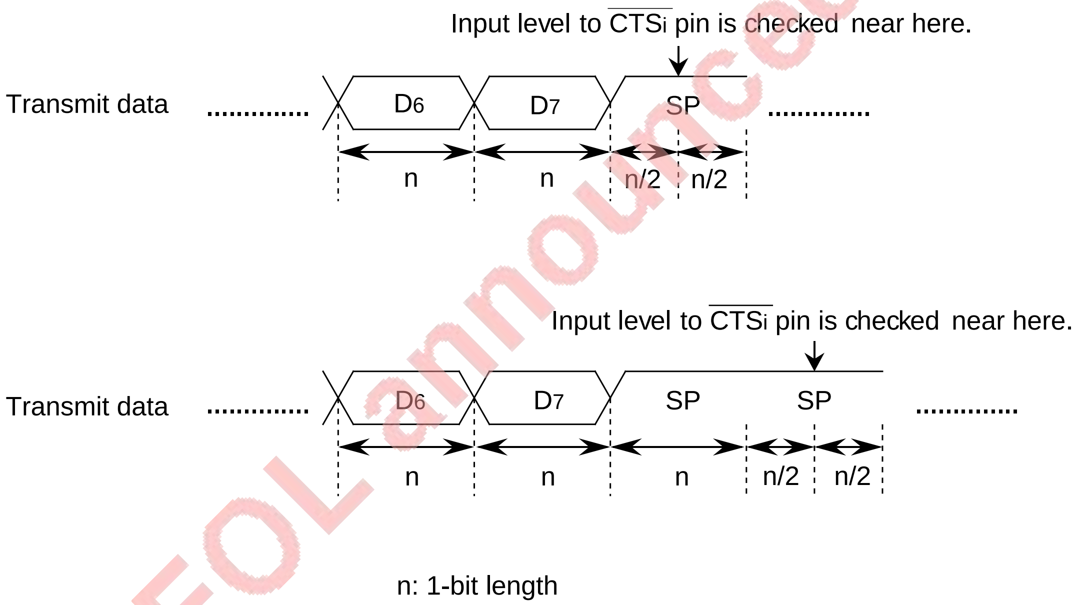
Serial I/O (UART mode)

Q

In the case selecting the $\overline{CTS}$ function in UART (clock asynchronous serial I/O) mode, when the transmitting side check the $\overline{CTS}$ input level ?

A

It is check near the middle of the stop bit (when two stop bits are selected, the second stop bit).



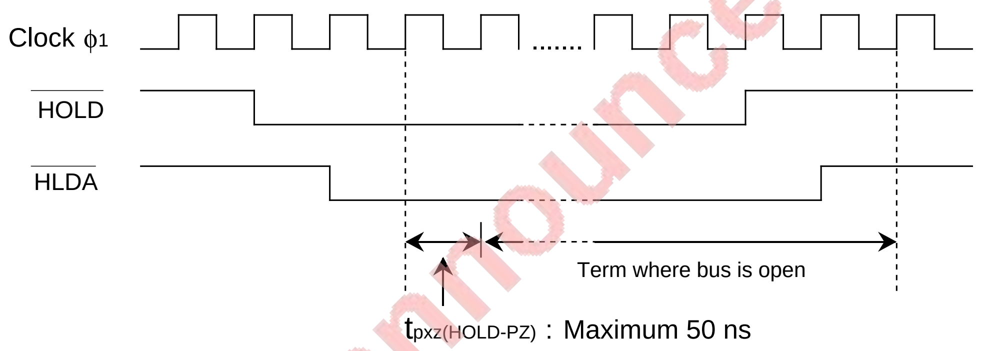
Hold function

Q

When “L” level is input to the $\overline{\text{HOLD}}$ pin, how long is the bus actually opened ?

A

The bus is opened after 50 ns at maximum has passed from the rising edge of next clock ϕ_1 when the $\overline{\text{HLDA}}$ pin output becomes “L” level.



Q

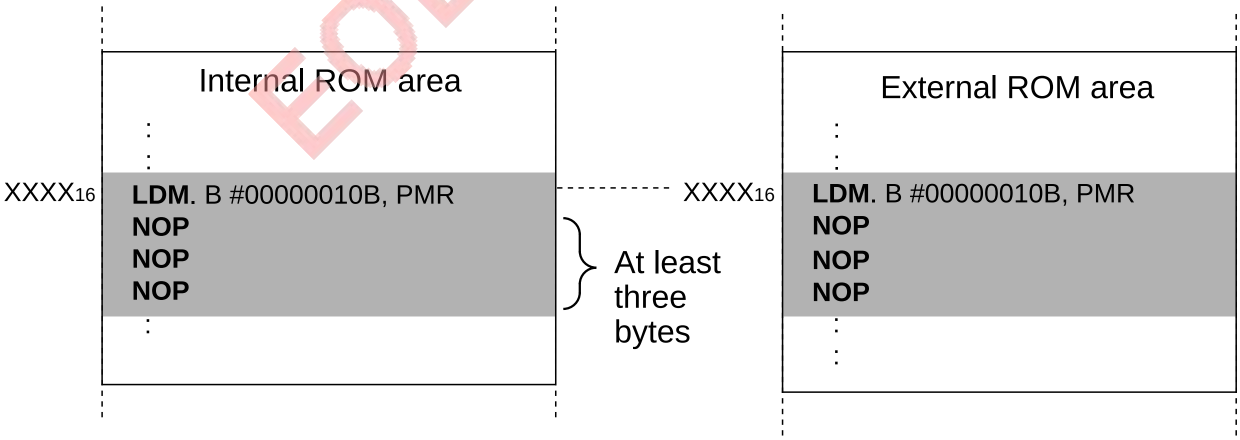
If the processor mode is switched as described below by using the processor mode bits (bits 1 and 0 at address $5E_{16}$) during program execution, is there any precaution in software?

- Single-chip mode → Microprocessor mode
- Memory expansion mode → Microprocessor mode

A

If the processor mode is switched as described above by using the processor mode bits, the mode is switched simultaneously when the cycle to write to the processor mode bits is completed. Then, the program counter indicates the address next to the address (address $XXXX_{16}$) that contains the write instruction for the processor mode bits. Additionally, access to the internal ROM area is disabled. However, since the instruction queue buffer can prefetch up to three instructions, the address in the external ROM area and is accessed first after the mode is switched is one of $XXXX_{16}$ + 1 to $XXXX_{16}$ + 4. The instructions at addresses $XXXX_{16}$ + 1 to $XXXX_{16}$ + 3 in the internal ROM area can be executed. To prevent this problem, process the following by software.

① Write the write instruction for the processor mode bits and next instructions (at least three bytes) at the same addresses both in the internal ROM and external ROM areas. (See below.)



② Transfer the write instruction for the processor mode bits to an internal RAM area and make a branch to there in order to execute the write instruction. After that, make a branch to the program address in the external ROM area. (Contents of the instruction queue buffer is initialized by a branch instruction.)

SFR

Q

Is there any SFR for which instructions that can be used to set registers or bits are limited?

A

Use the **STA** or **LDM** instruction to set the registers or the bits listed below. Do not use read-modify-write instructions (i.e., **CLB, SEB, INC, DEC, ASL, ASR, LSR, ROL,** and **ROR**).

UART0 baud rate register (address 31_{16})
UART1 baud rate register (address 39_{16})
UART0 transmit buffer register (addresses 33_{16}, 32_{16})
UART1 transmit buffer register (addresses $3B_{16}$, $3A_{16}$)
Timer A4 two-phase pulse signal processing select bit (bit 7 at address 44_{16})
Timer A3 two-phase pulse signal processing select bit (bit 6 at address 44_{16})
Timer A2 two-phase pulse signal processing select bit (bit 5 at address 44_{16})

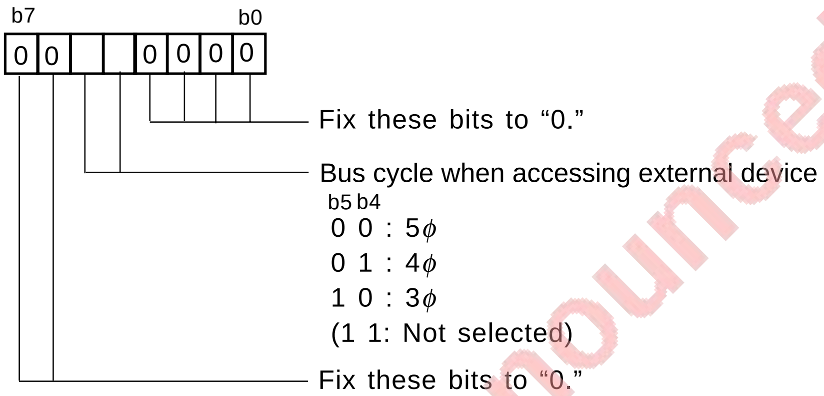
Clock

Q

Is there any precaution when $f(X_{IN})$ > 25 MHz ?

A

Set the processor mode register 1 (address $5F_{16}$) to the following.



The microcomputer becomes the following state by the setting above.

- f_4, f_{32}, f_{128}, or f_{1024} can be selected for the operating clock of internal peripheral devices such as timer.
- SFR and internal ROM area are accessed at 3ϕ bus cycle. Internal RAM area is accessed at 2ϕ bus cycle.
- 3ϕ, 4ϕ, or 5ϕ can be selected for the bus cycle when accessing an external device. 2ϕ cannot be selected for the bus cycle.

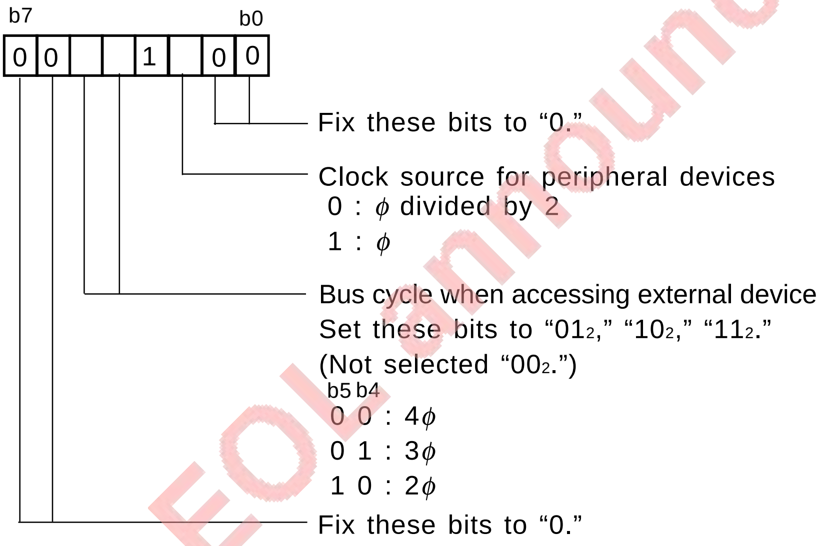
Clock

Q

Is there any precaution when $f(X_{IN}) \leq 25$ MHz ?

A

When setting the CPU running speed select bit (bit 3 at address $5F_{16}$) to “1,” SFR and internal ROM access become faster than this bit is “0.” Accordingly, we recommend to set this bit to “1.”
However, do not set bits 5 to 3 at address $5F_{16}$ to “001_2.” When setting bit 3 at address $5F_{16}$ to “1,” set bit 5, bit 4, or both bits 5 and 4 to “1” at the same time because bits 5 and 4 at address $5F_{16}$ become “00_2” at reset.



Substitute for 7700 Series/7750 Series

Q

Are there precautions when the 7751 Series substitutes for the 7700 Series or the 7750 Series?

A

The common precautions are described below. Refer to the relevant chapter for details.

- Fix the processor status register (PS) bits 15 to 11 to “0.” Do not set these bits to “1.”
- There are the structure differences in the processor mode register 0 (address $5E_{16}$) and the processor mode register 1 (address $5F_{16}$).
- The A-D conversion interrupt request bit (bit 3 at address 70_{16}) is undefined at reset. Set this bit to “0” by software before use.
- Clear the receive enable bit (bit 2 at addresses 35_{16}, $3D_{16}$) to “0” when clearing the overrun error flag (bit 4 at addresses 35_{16}, $3D_{16}$) to “0.”
  This is only method that the overrun error flag is cleared to “0”
- There are instructions of which number of the instruction cycle is decreased. Accordingly, it is possible that the instruction execution timing become faster.
- Part of the electrical characteristics, Ready function, Hold function, and the bus timing are different.

Watchdog timer

Q

When detecting the software runaway by the watchdog timer, if not software reset but setting the same value as the contents of the reset bector address to the watchdog timer interrupt bector address is processed, how does it result in?
When branching to the reset branch address within the watchdog timer interrupt routine, how does it result in?

A

The CPU registers and the SFR are not initialized in the above-mentioned way. Accordingly, the user must perform the initial setting for these all by software.
The processor interrupt priority level (IPL) retains “7” of the watchdog timer interrupt priority level, and that is not initialized. Consequently, all interrupt requests are not accepted.
When rewriting the IPL by software, save once the 16-bit immediate value to the stack area and next restore that 16-bit immediate value to all bits of the processor status register (PS).

We recommend software reset in order to initialize the microcomputer for software runaway.

MEMORANDUM

GLOSSARY

GLOSSARY

This section briefly explains the terms used in this user's manual. The terms defined here apply to this manual only.

Term	Meaning	Relevant term
Access	Means performing read, write, or read and write.	
Access space	An accessible memory space of up to 16 Mbytes.	Access
Access characteristics	Means whether accessible or not.	Access
Baud rate	Means a transfer rate of Serial I/O.	
Branch	Means moving the program's execution point (= address) to another location.	
Bus control signal	A generic name for ALE, $\overline{E}$, $\overline{BHE}$, R/$\overline{W}$, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$ and BYTE signals.	
Count source	A signal that is counted by Timers A and B, the UARTi baud rate register (BRGi) and Watchdog timer. That is f_2/f_4, f_{16}/f_{32}, f_{64}/f_{128}, f_{512}/f_{1024} selected by the count source select bits and others.	
Down-count	Means decreasing by 1 and counting.	Up-count
External area	An accessible area for external devices connected in the memory expansion or microprocessor mode. It is up to 16-Mbyte external area.	Internal area
External bus	A generic name for the external address bus and the data bus.	
External device	Devices connected externally to the microcomputer. A generic name for a memory, an I/O device and a peripheral IC.	
Internal area	An accessible internal area. A generic name for areas of the internal RAM, internal ROM and the SFR.	External area
Interrupt routine	A routine that is automatically executed when an interrupt request is accepted. Set the start address of this routine into the interrupt vector address.	
LSB first	Means a transfer data format of Serial I/O; LSB is transferred first.	MSB first
MSB first	Means a transfer data format of Serial I/O; MSB is transferred first.	LSB first
Overflow	A state where the up-count resultant is greater than the counter resolution.	Under flow Up-count
Read-modify-write instruction	An instruction that reads the memory contents, modifies them and writes back to the same address. Relevant instructions are the **ASL**, **ASR**, **CLB**, **DEC**, **INC**, **LSR**, **ROL**, **ROR**, **SEB** instructions.	
Signal required for access to external device	A generic name for bus control, address bus, and data bus signals.	Bus control signal
Stop mode	A state where the oscillation circuit halts and the program execution is stopped. By executing the **STP** instruction, the microcomputer enters Stop mode.	Wait mode
UART	Clock asynchronous serial I/O. When used to designate the name of a functional block, this term also means the serial I/O which can be switched to the cock synchronous serial I/O.	Clock synchronous serial I/O.
Under flow	A state where the down-count resultant is greater than the counter resolution.	Overflow Down-count

Term	Meaning	Relevant term
Up-count	Means increasing by 1 and counting.	Down-count
Wait mode	A state where the oscillation circuit is operating, however, the program execution is stopped. By executing the **WIT** instruction, the microcomputer enters Wait mode.	Stop mode

MEMORANDUM

MITSUBISHI SEMICONDUCTORS
USER'S MANUAL
7751 Group

Jul. First Edition 1997

Editioned by
Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division