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 / 7700 SERIES

7702/7703 Group

User's Manual

Preface

This manual describes the hardware of the Mitsubishi CMOS 16-bit microcomputers 7702 Group and 7703 Group. After reading this manual, the user will be able to understand the functions, so that they can utilize their capabilities fully.

For details concerning the software, refer to the 7700 Family Software Manual.

BEFORE USING THIS MANUAL

1. Constitution

This user's manual consists of the following chapters. Refer to the chapters relevant to used products and the processor mode.

●Chapter 1. DESCRIPTION to Chapter 17. APPLICATION

Functions which are common to all products and all processor modes are explained, using the M37702M2BXXXFP as an example.
When there are functional differences between the low voltage version, PROM version and the 7703 Group, the referential section is indicated. Refer to that section about differences and to "**Chapter. 1 to Chapter. 17**" about the common functions.

●Chapter 18. LOW VOLTAGE VERSION

Refer to this chapter when using the products of which difference of electrical characteristics identification code (see on page 1–2) is "L," the M37702M2LXXXGP for example. This chapter mainly explains the differences from the M37702M2BXXXFP, using the M37702M2LXXXGP as an example.

●Chapter 19. PROM VERSION

Refer to this chapter when using the products of which memory identification code (see on page 1–2) is "E," the M37702E2BXXXFP for example. This chapter mainly explains the differences from the M37702M2BXXXFP, using the M37702E2BXXXFP as an example.

●Chapter 20. 7703 GROUP

Refer to this chapter when using the 7703 Group. This chapter mainly explains the differences from the 7702 Group, using the M37703M2BXXXSP as an example.

●Appendix

Useful information for 7702 and 7703 Groups usage is shown.

2. Remark

●25 MHz version and 16 MHz version

The 25 MHz version products are distinguished from the 16 MHz version products in part of Chapters as the case may be. Refer to it as follows:

•Products of which difference of electrical characteristics identification code is "B," M37702M2BXXXFP as an example...Column of "25 MHz version"
•Products of which difference of electrical characteristics identification code is "A," M37702M2AXXXFP as an example...Column of "16 MHz version"

●Product expansion

See the latest data book and data sheets. Additionally, ask the contact addresses on the last page.

●Electrical characteristics

See also the latest data book or data sheet.

●Development support tools

See the latest data book and data sheet.

●Software

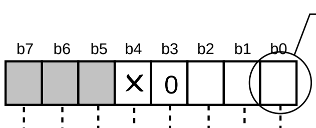
See "7700 Family Software Manual."

●**Mask ROM Confirmation Form, PROM Confirmation Form, Mark Specification Form**

Copy the form in the latest data book and use it. Or, ask the contact addresses on the last page.

3. Register structure

The view of the register structure is described below:



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	–

✻1

Blank : Set to “0” or “1” to meet the purpose.
0 : Set to “0” at writing.
1 : Set to “1” at writing.
× : This bit is not used in the specific mode or state. It may be either “0” or “1.”
: 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 data.

RO : It is possible to read the bit state at reading. The written value becomes invalid. Accordingly, the written value may be either “0” or “1.”

WO : The written value becomes valid data. It is not possible to read the bit state. The value is undefined at reading. However, the bit with the commentaries of “ The value is “0” at reading” in the functions column or the notes is always “0” at reading.(See to ✻4 above.)

— : It is no possible to read the bit state. The value is undefined at reading. However, the bit with the commentaries of “The value is “0” at reading” in the functions column or the notes is always “0” at reading.(See to ✻4 above.) The written value becomes invalid. Accordingly, the written value may be “0” or “1.”

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

Table of Contents

CHAPTER 13. RESET

CHAPTER 14. CLOCK GENERATING CIRCUIT

CHAPTER 15. ELECTRICAL CHARACTERISTICS

CHAPTER 16. STANDARD CHARACTERISTICS

CHAPTER 17. APPLICATIONS

CHAPTER 18. LOW VOLTAGE VERSION

APPENDIX

GLOSSARY

MEMORANDUM

CHAPTER 1

DESCRIPTION

DESCRIPTION

The 16-bit single-chip microcomputers 7702 Group and 7703 Group are suitable for office, business, and industrial equipment controllers that require high-speed processing of large amounts of data.
These microcomputers develop with the M37702M2BXXXFP as the base chip. This manual describes the functions about the M37702M2BXXXFP unless there is a specific difference and refers to the M37702M2BXXXXFP as “M37702.”

Notes 1: About details concerning each microcomputer’s development status of the 7702/7703 Group, inquire of **“CONTACT ADDRESSES FOR FURTHER INFORMATION”** described last.

Notes 2: How the 7702/7703 Group’s type name see is described below.

M 3 77 02 M 2 B XXX FP

- M — Mitsubishi integrated prefix
- 3 — Represent an original single-chip microcomputer
- 77 — Series designation using 2 digits
- 02 — Circuit function identification code using 2 digits
- M — Memory identification code using a digit
  - M: Mask ROM
  - E: EPROM
  - S: External ROM
- 2 — Memory size identification code using a digit
- B — Difference of electrical characteristics identification code using a digit
- XXX — Mask ROM number
- FP — Package style
  - FP: Plastic molded QFP
  - GP: Plastic molded QFP
  - HP: Plastic molded fine-pitch QFP
  - SP: Plastic molded SDIP
  - FS: Ceramic QFN

1.1 Performance overview

Table 1.1.1 shows the performance overview of the M37702.

7703 Group
Refer to "**Chapter 20. 7703 GROUP.**"

Table 1.1.1 M37702 performance overview

Parameters		Functions
Number of basic instructions		103
Instruction execution time	M37702M2BXXXFP	160 ns (the minimum instruction at f(X_{IN}) = 25 MHz)
	M37702M2AXXXFP	250 ns (the minimum instruction at f(X_{IN}) = 16 MHz)
External clock input frequency f(X_{IN})	M37702M2BXXXFP	25 MHz (maximum)
	M37702M2AXXXFP	16 MHz (maximum)
Memory size	ROM	16384 bytes
	RAM	512 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		8-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		60 mW (at f(X_{IN}) = 16 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

Notes 1: All of the 7702 Group microcomputers are the same except for the package type, memory type, memory size, and electric characteristics.
2: For the low voltage version, refer to **"Chapter 18. LOW VOLTAGE VERSION."**

1.2 Pin configuration

Figure 1.2.1 shows the M37702M2BXXXFP pin configuration. Figure 1.2.2 shows the M37702M2BXXXHP pin configuration.

Note: For the low voltage version of the 7702 Group, refer to **"Chapter 18. LOW VOLTAGE VERSION."**

7703 Group

Refer to **"Chapter 20. 7703 GROUP."**

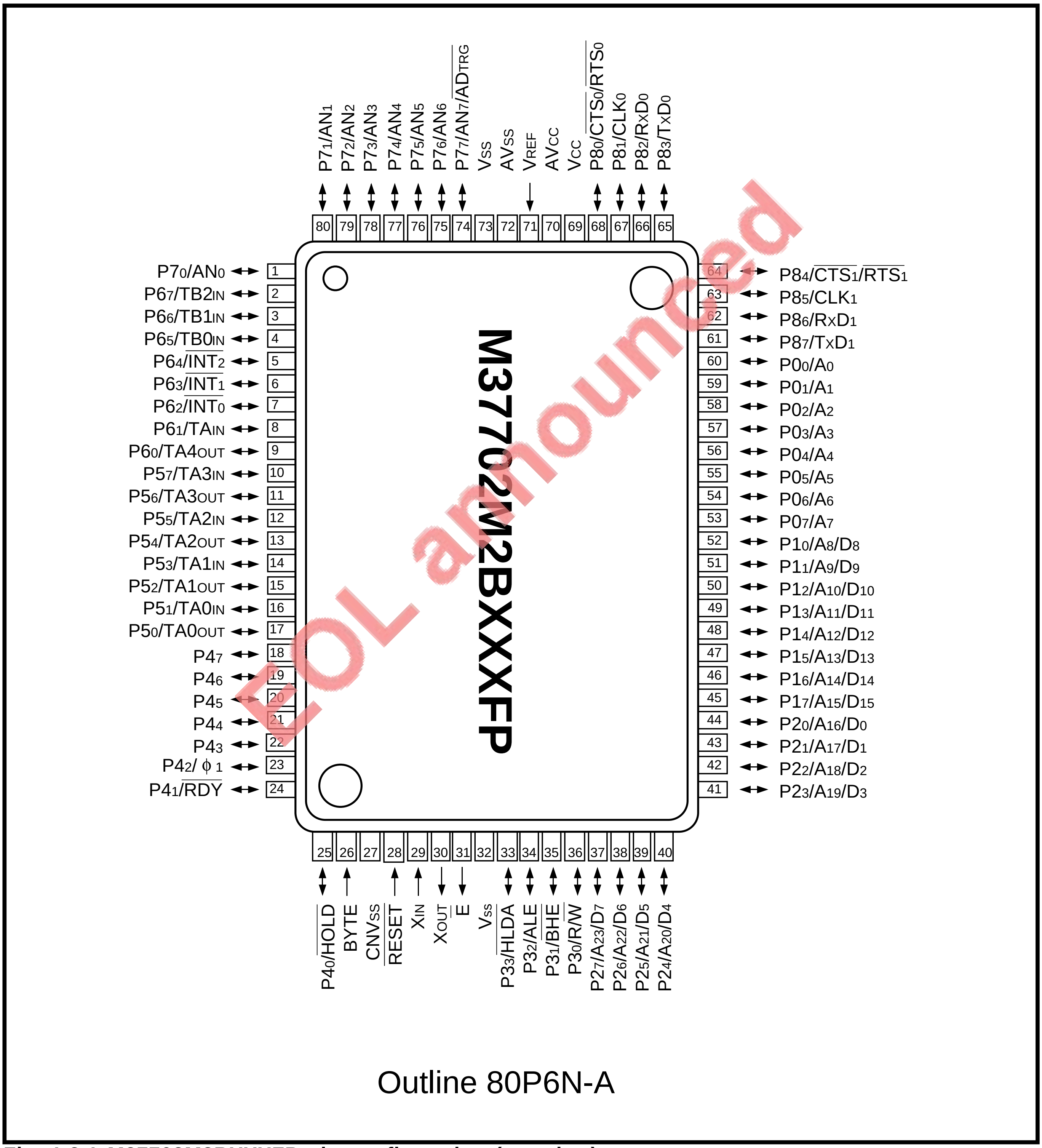
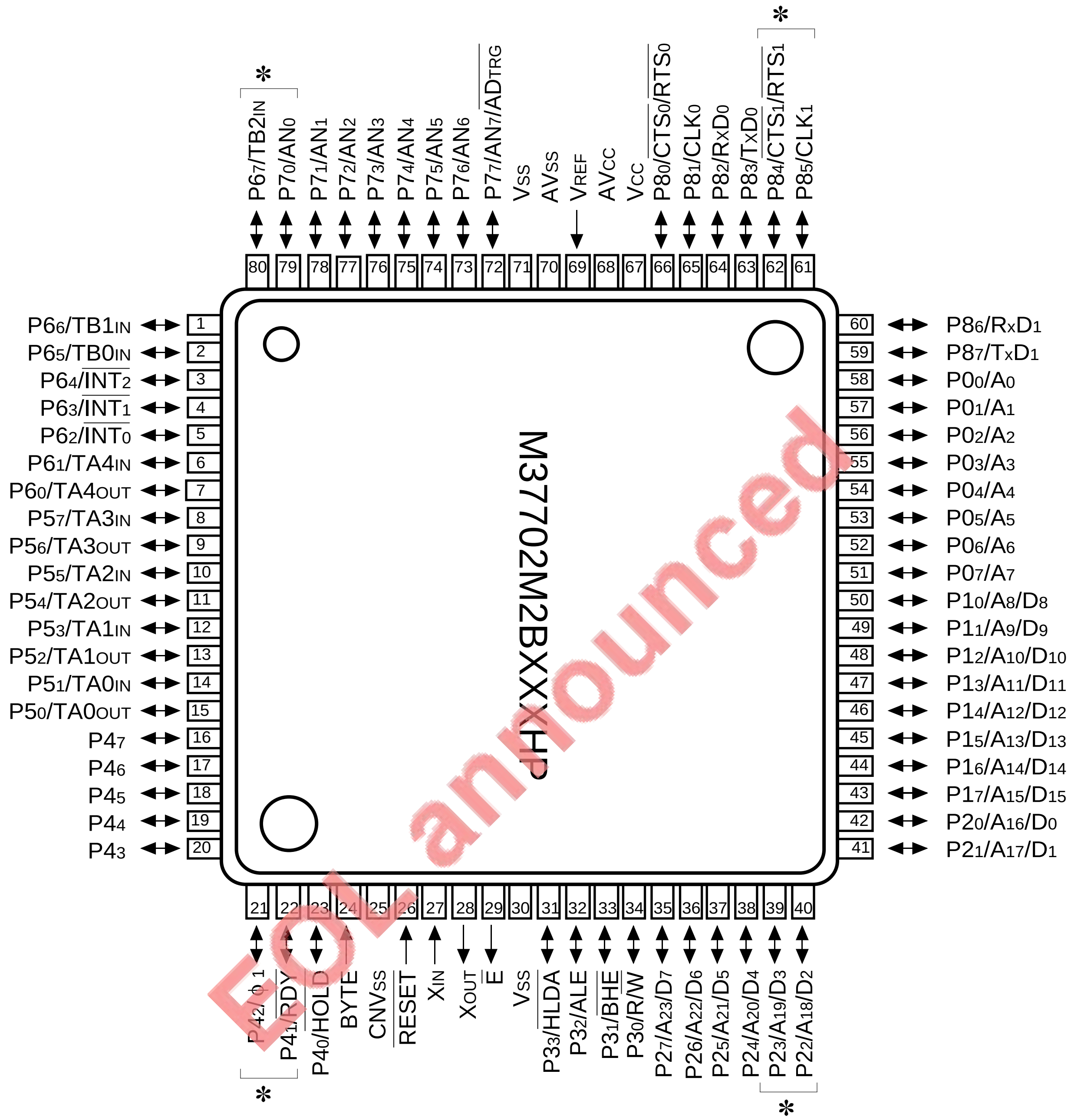


Fig. 1.2.1 M37702M2BXXXFP pin configuration (top view)



Outline 80P6D-A

✻ : The M37702M2BXXXHP have the pin configuration shifted to 2 pins assignment from the M37702M2BXXXFP.

Fig. 1.2.2 M37702M2BXXXHP pin configuration (top view)

1.3 Pin description

Tables 1.3.1 to 1.3.3 list the pin description. However, the pin description in the EPROM mode of the built-in PROM version is described to section **"19.2 EPROM mode."**

7703 Group

The 7703 Group does not have part of pins. Refer to **"Chapter 20. 7703 GROUP."**

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.
$\overline{RESET}$	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 pins X_{IN} and X_{OUT}. 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	
$\overline{E}$	Enable output	Output	This pin outputs $\overline{E}$ signal. Data/instruction code read or data write is performed when output from this pin is "L" level.
BYTE	Bus width selection input	Input	**[Single-chip mode]** Connect to Vss. **[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		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.

✻ : In the low voltage version, supply 2.7–5.5V to Vcc.

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 signal 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.

* : The 7703 Group does not have the $P3_3/\overline{HLDA}$ pin.

DESCRIPTION

1.3 Pin description

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.

* : The 7703 Group does not have the $P4_3$–$P4_6$, $P6_0$, $P6_1$, $P6_6$, $P6_7$, $P7_3$–$P7_6$, $P8_4$, and $P8_5$ pins.

1.3.1 Example for processing unused pins

Examples for processing unused pins are described below. These descriptions are just examples. The user shall modify them according to the user's actual application and test them.

(1) In single-chip mode

Table 1.3.4 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. **(Notes 1, 3)**
$\overline{E}$	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.

3: In the 7703 Group, the following ports does not have the corresponding pins and have only the direction registers. Fix the bit of these direction registers to "1" (output mode).

•Ports $P3_3$, $P4_3$–$P4_6$, $P6_0$, $P6_1$, $P6_6$, $P6_7$, $P7_3$–$P7_6$, $P8_4$, $P8_5$

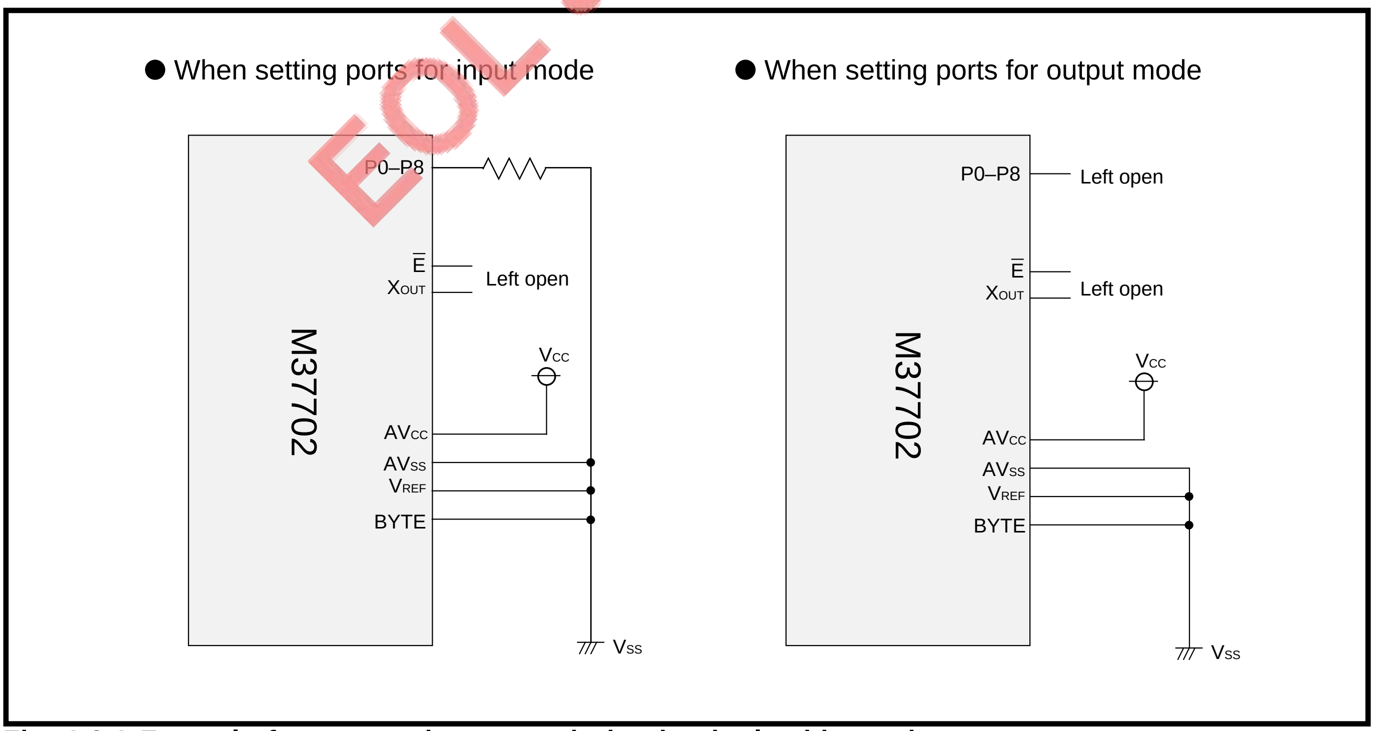


Fig. 1.3.1 Example for processing unused pins in single-chip mode

DESCRIPTION

1.3 Pin description

(2) In memory expansion mode

Table 1.3.5 Example for processing unused pins in memory expansion 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. **(Notes 1, 6, 7)**
$\overline{BHE}$ (**Note 2**)	Leave them open. **(Note 4)**
ALE (**Note 3**)	
$\overline{HLDA}$ (**Note 6**)	
X_{OUT} (**Note 5**)	Leave it open.
$\overline{HOLD}$, $\overline{RDY}$ **(Note 8)**	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. 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.

6: In the 7703 Group, the following ports does not have the corresponding pins and have only the direction registers. Fix the bit of these direction registers to "1" (output mode).
•Ports $P4_3$–$P4_6$, $P6_0$, $P6_1$, $P6_6$, $P6_7$, $P7_3$–$P7_6$, $P8_4$, $P8_5$
There is not the $\overline{HLDA}$ pin.

7: Set the $P4_2/\phi_1$ pin to the $P4_2$ function (clock ϕ_1 output disabled), and perform the same processing as ports $P4_3$–$P4_7$, P5–P8.

8: When processing unused pins, use the possible shortest wiring (within 20 mm from the microcomputer).

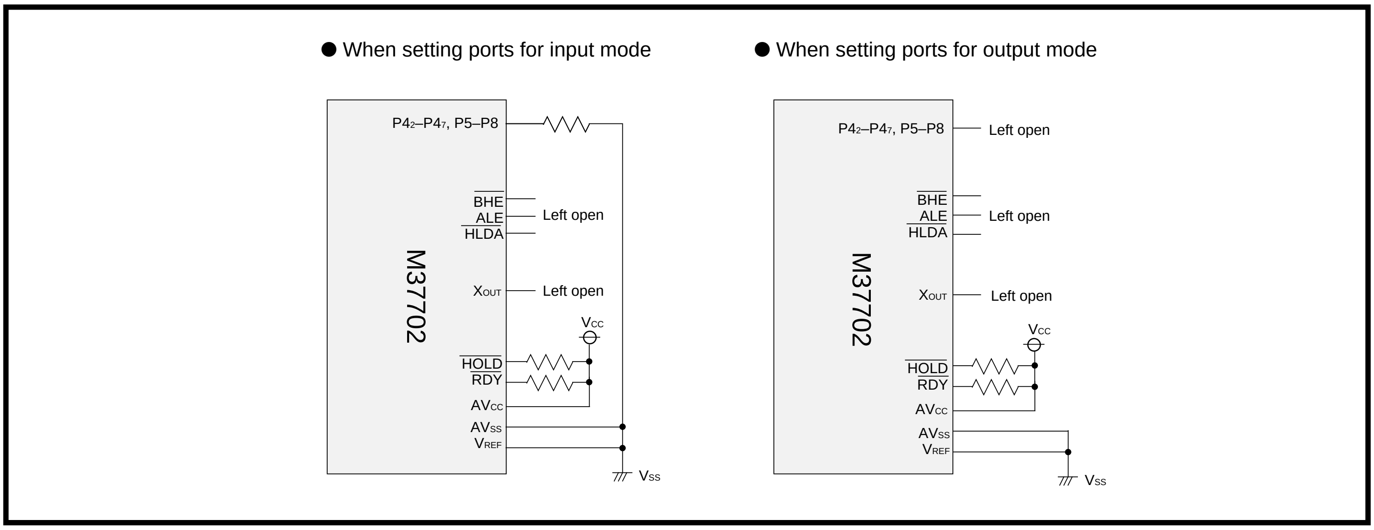


Fig. 1.3.2 Example for processing unused pins in memory expansion mode

(3) In microprocessor mode

Table 1.3.6 Example for processing unused pins in microprocessor mode

Pin name	Example of processing
Ports $P4_3$ 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. **(Notes 1, 6)**
$\overline{\text{BHE}}$ (**Note 2**)	Leave it open. **(Note 4)**
ALE (**Note 3**)	
$\overline{\text{HLDA}}$, ϕ_1 (**Note 6**)	
X_{OUT} (**Note 5**)	Leave it open.
$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$ (**Note 7**)	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 $\overline{\text{BYTE}}$ pin.

3: This applies when "H" level is input to the $\overline{\text{BYTE}}$ 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. 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.

6: In the 7703 Group, the following ports does not have the corresponding pins and have only the direction registers. Fix the bit of these direction registers to "1" (output mode).
•Ports $P4_3$–$P4_6$, $P6_0$, $P6_1$, $P6_6$, $P6_7$, $P7_3$–$P7_6$, $P8_4$, $P8_5$
There is not the $\overline{\text{HLDA}}$ pin.

7: When processing unused pins, use the possible shortest wiring (within 20 mm from the microcomputer).

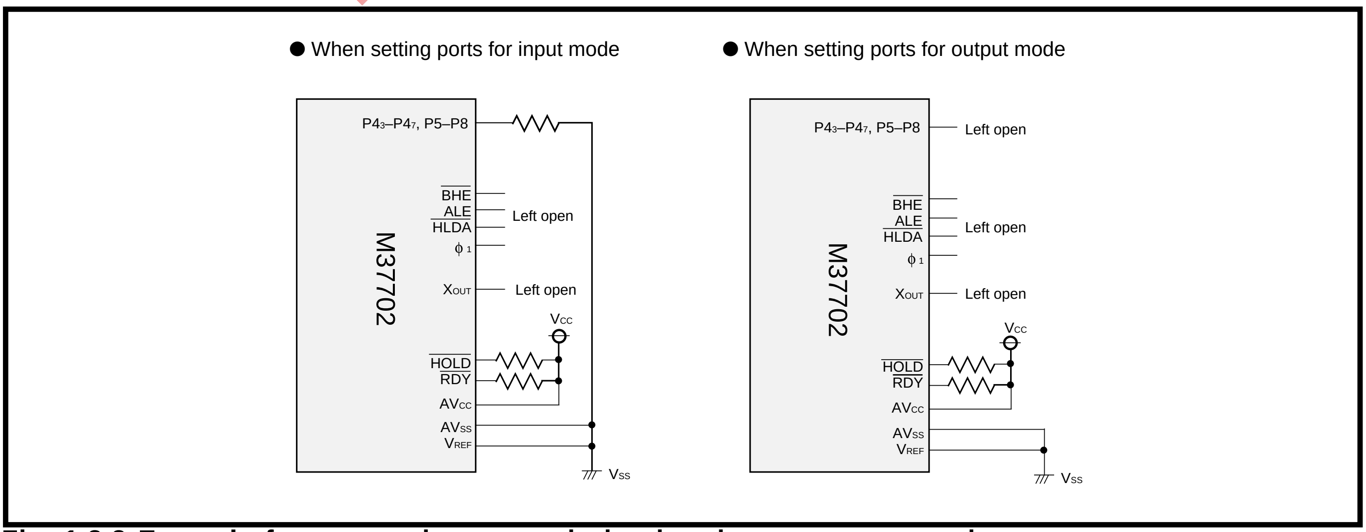
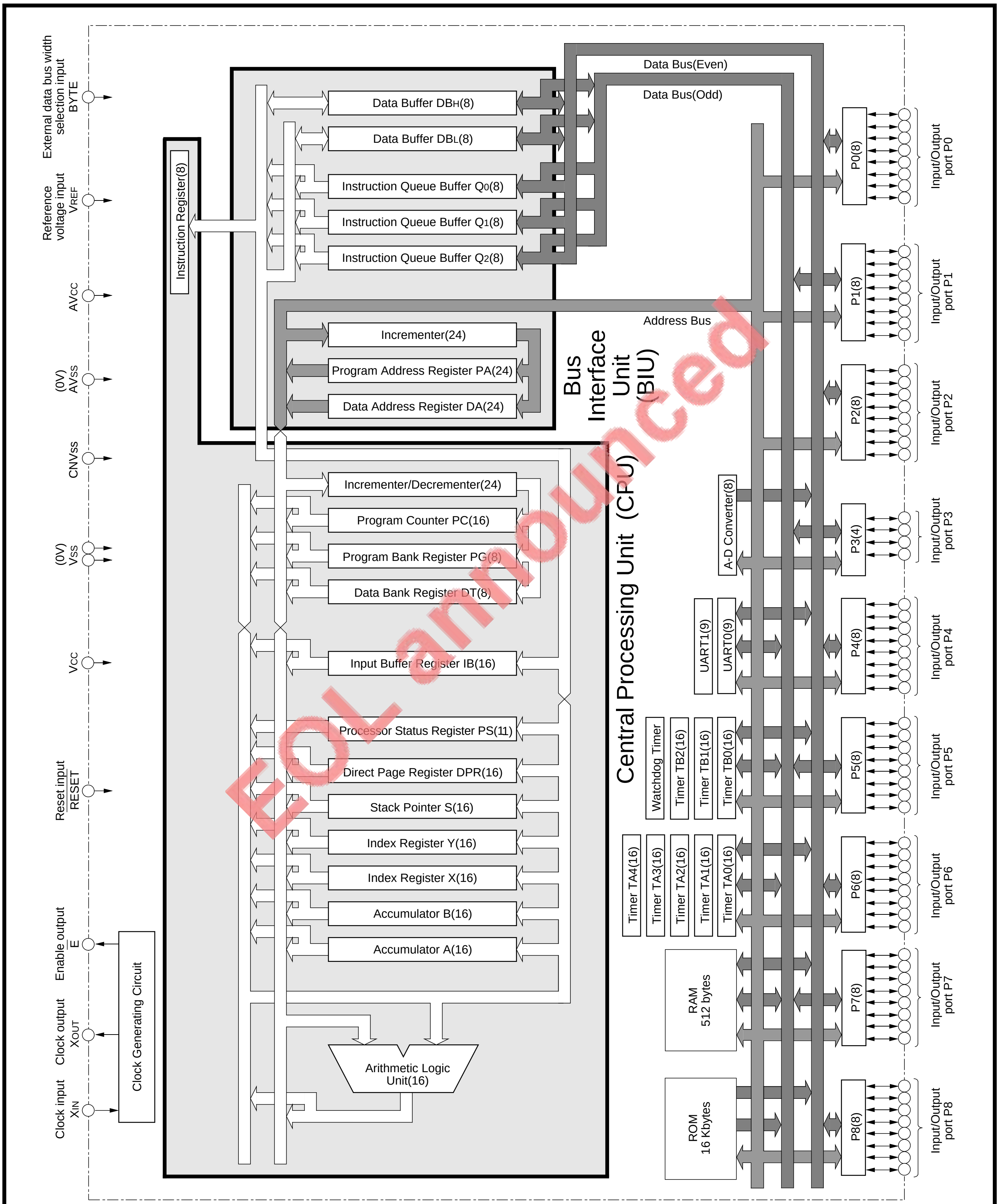


Fig. 1.3.3 Example for processing unused pins in microprocessor mode

1.4 Block diagram

Figure 1.4.1 shows the M37702 block diagram.



Note: The 7703 Group does not have the $P3_3$, $P4_3$–$P4_6$, $P6_0$, $P6_1$, $P6_6$, $P6_7$, $P7_3$–$P7_6$, $P8_4$, and $P8_5$ pins.

Fig. 1.4.1 M37702 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.

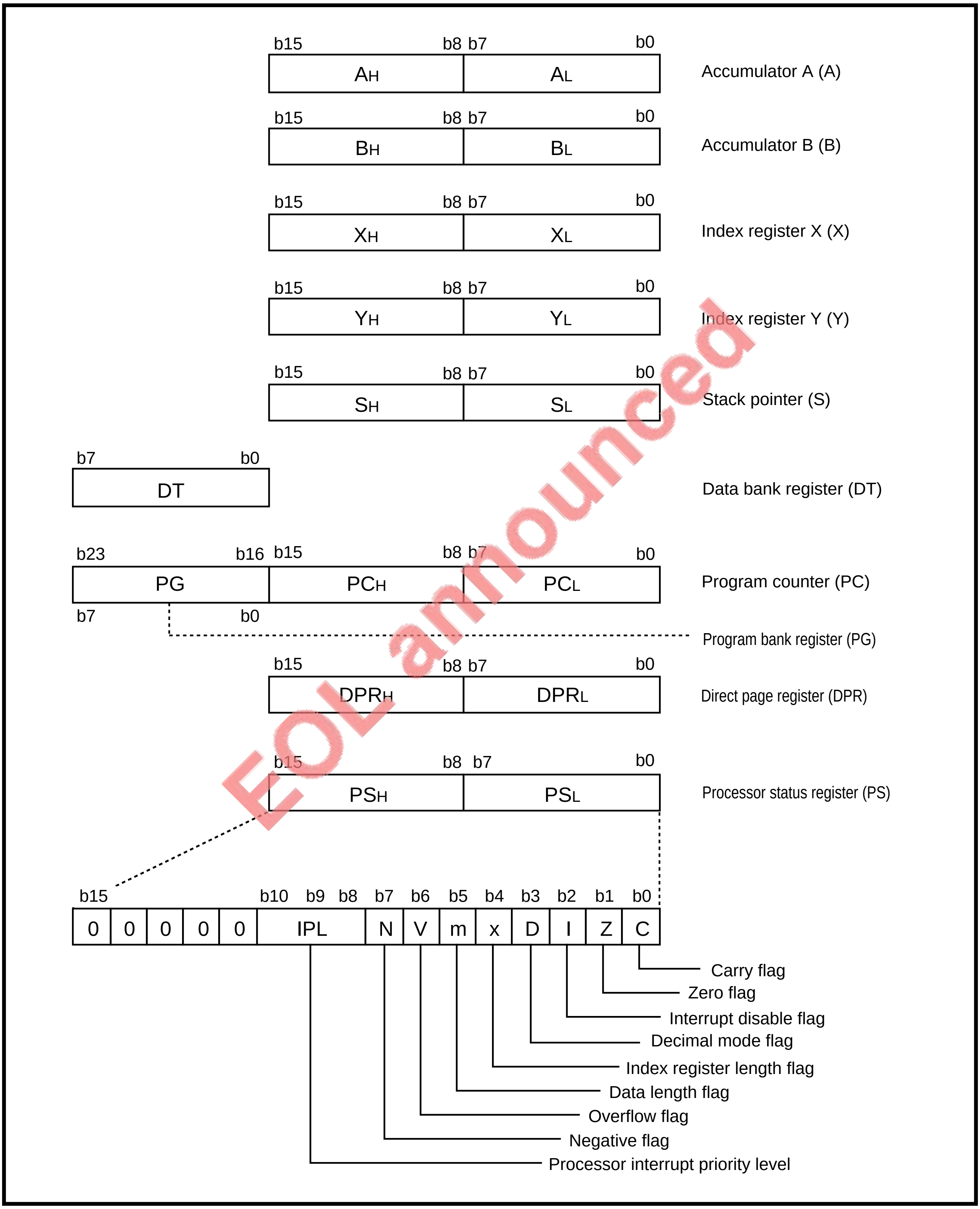


Fig. 2.1.1 CPU registers structure

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 indicates 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.

Note: Refer to **"7700 Family 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.

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.)

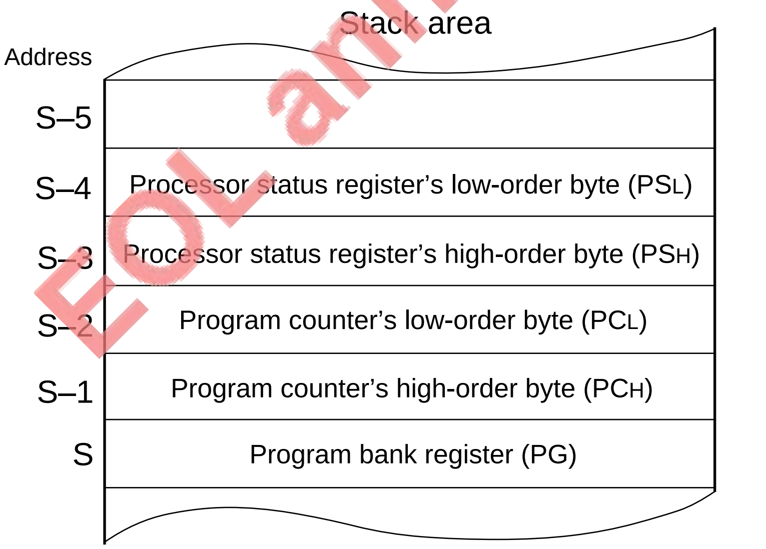
When completing the process in the interrupt routine and returning to the original routine, the contents of registers stored in the stack area are restored into the original registers in the reverse sequence (PS→PC→PG) by executing the **RTI** instruction. The contents of S is returned to the state before 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 **"7700 Family 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.

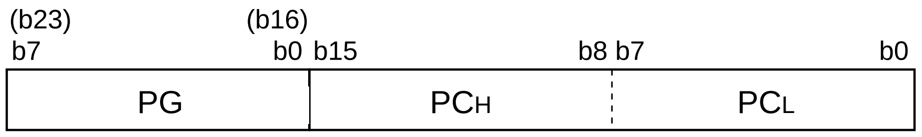


Fig. 2.1.3 Program counter and program bank register

2.1.6 Program bank register (PG)

The program bank register is an 8-bit register. This register indicates the high-order 8 bits 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.

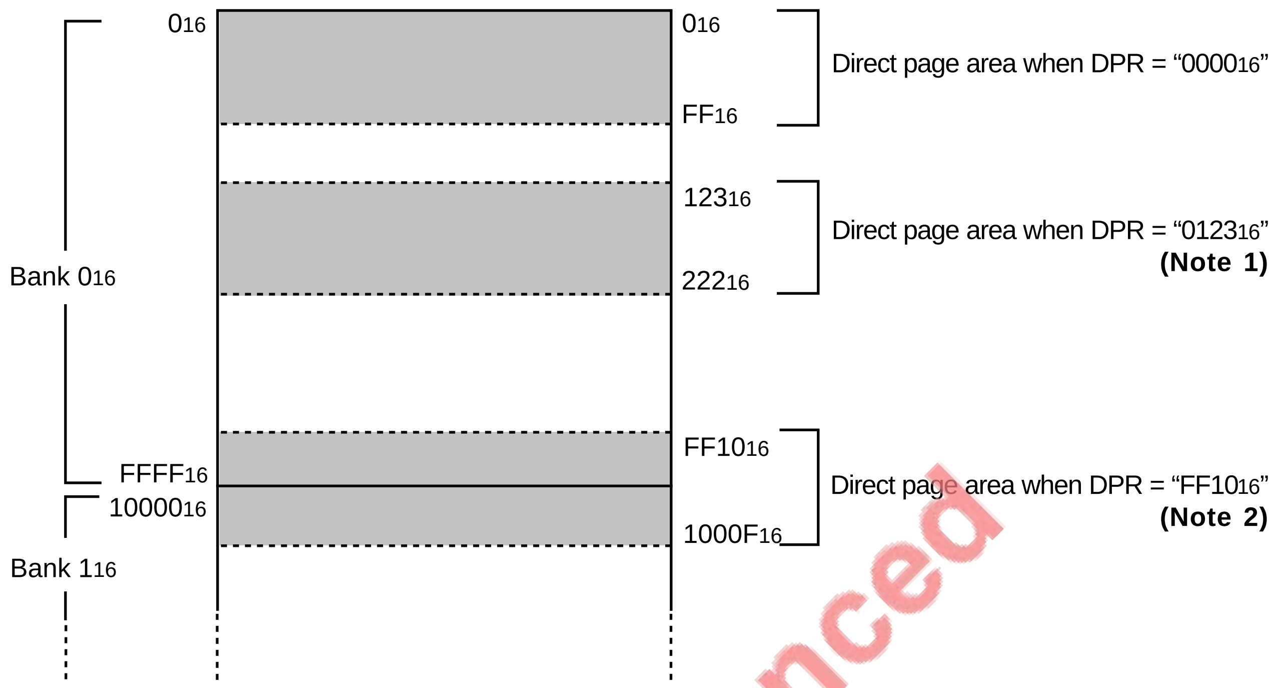
In the single-chip mode, make sure to fix this register to “00_{16}”. It is because the access space of the single-chip mode is the internal area within the bank 0_{16}.
This register is cleared to “00_{16}” at reset.

●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

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 staus register (PS)

Note: "0" is always read from each of bits 15–11.

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 **"7700 Family 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 **“7700 Family 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** instruction, a division 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

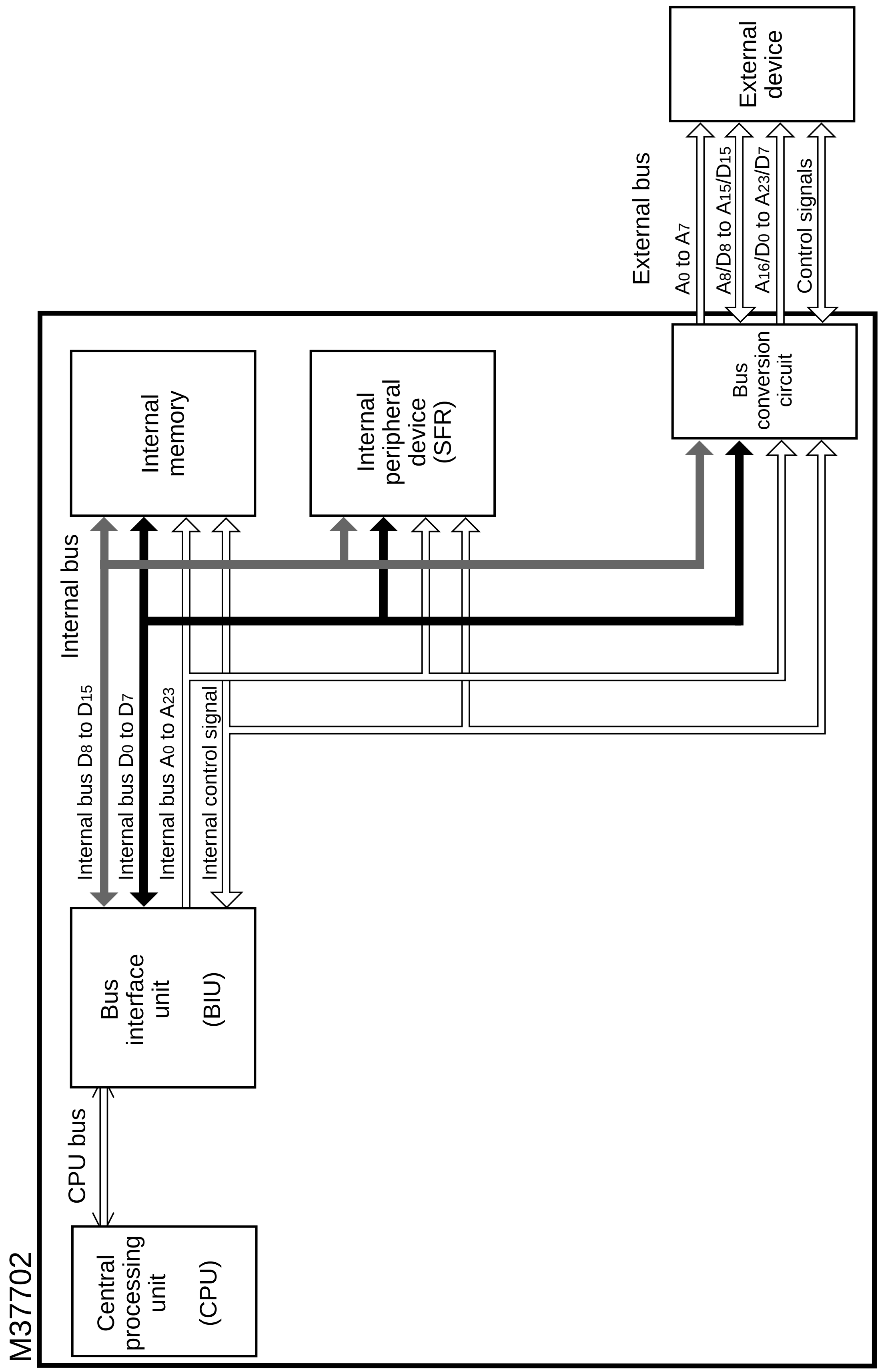
Transfer operation between the CPU and memory•I/O devices is always performed via the BIU.
Figure 2.2.1 shows the bus and bus interface unit (BIU).

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 Bus interface unit

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.

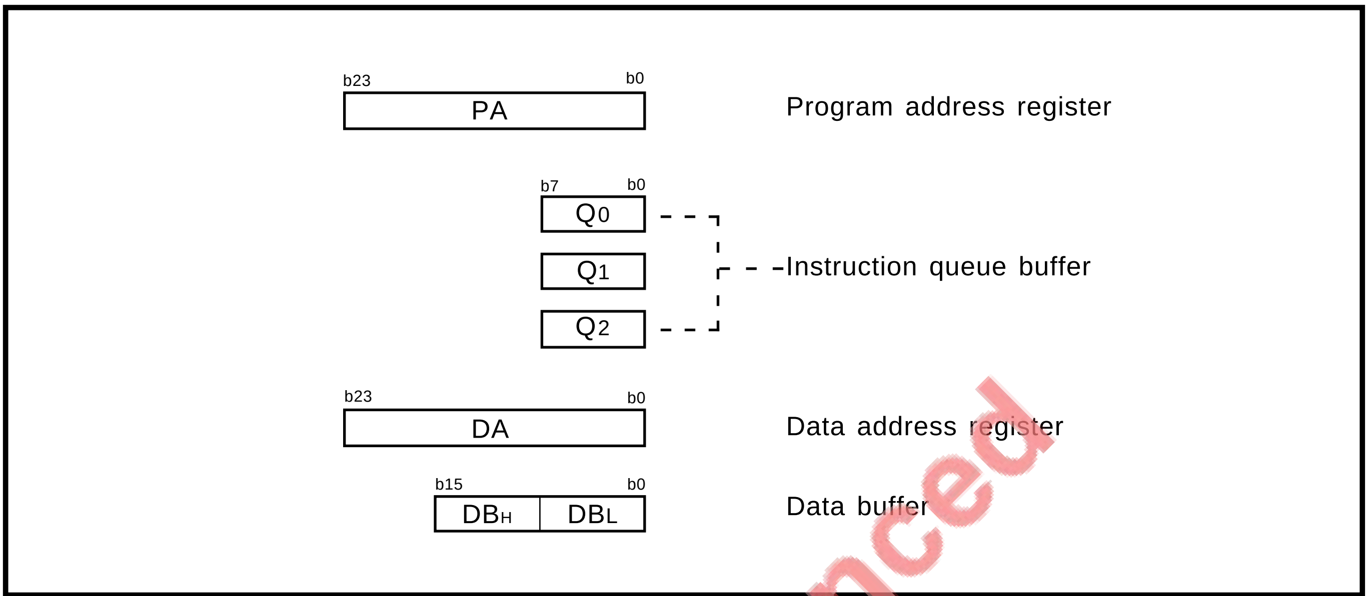


Fig. 2.2.2 Register structure of bus interface unit (BIU)

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 ϕ_{CPU}. The period of ϕ_{CPU} is normally the same as that of the internal clock ϕ. The internal bus operates based on the $\overline{E}$ signal. The period of the $\overline{E}$ signal is twice that of the internal clock 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 ϕ_{CPU} 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.
Refer to "**Chapter 12. CONNECTION WITH EXTERNAL DEVICES**" about the bus cycle at accessing the external devices.

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."

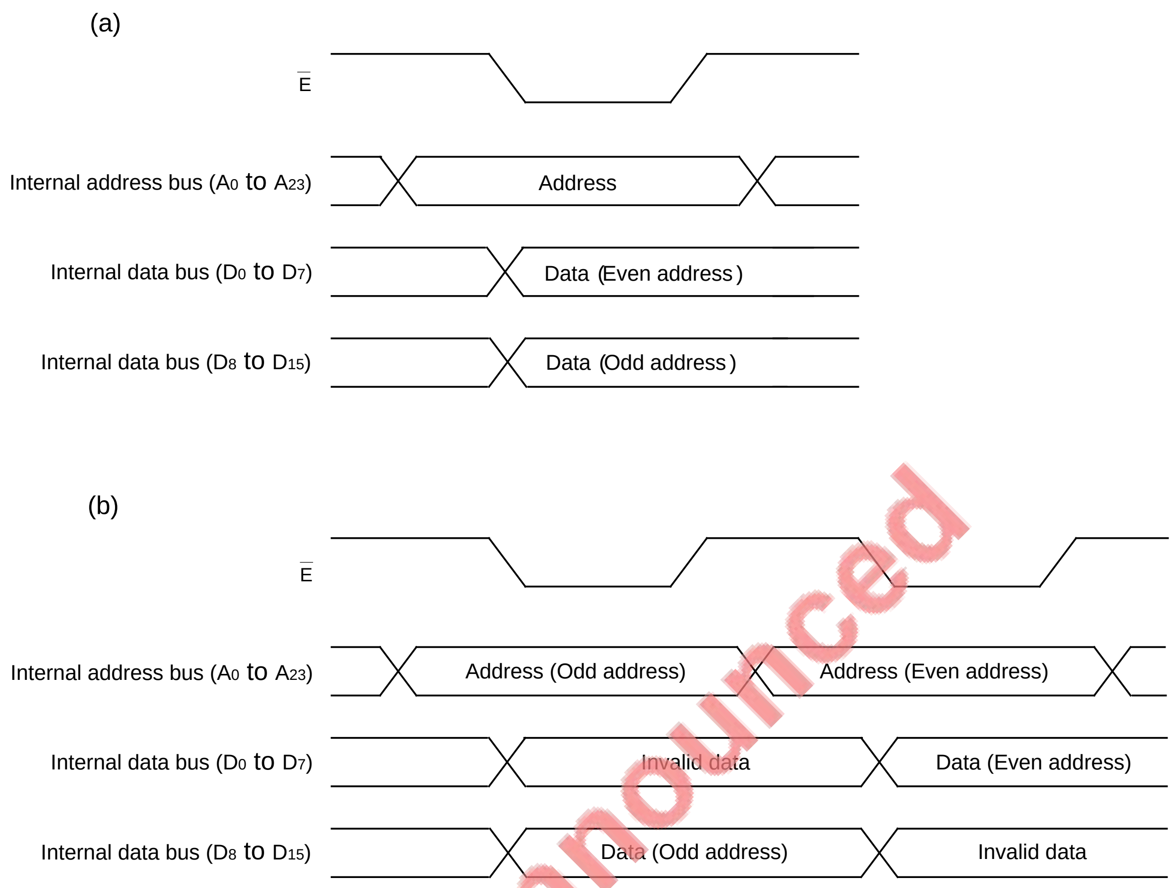
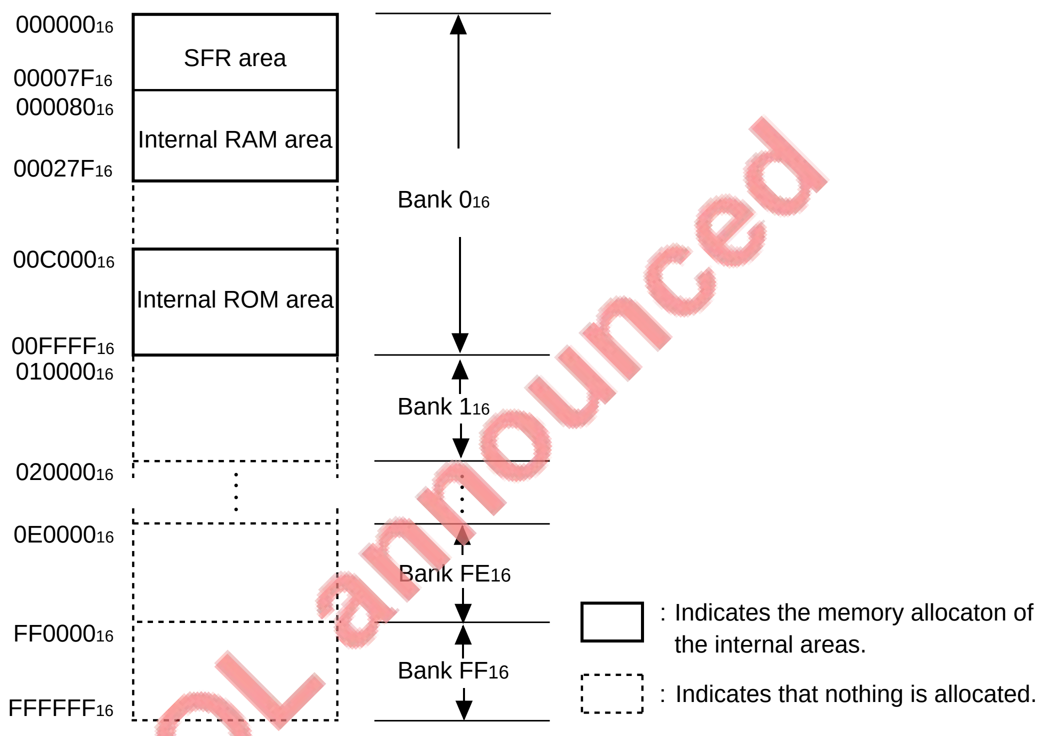


Fig. 2.2.3 Basic operating waveforms of bus interface unit (BIU)

2.3 Access space

Figure 2.3.1 shows the M37702'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 000000_{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 M37702M2BXXXFP.
Refer to **"Appendix 1. Memory assignment"** for other products.

SFR : Special Function Register

Fig. 2.3.1 M37702's 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."**

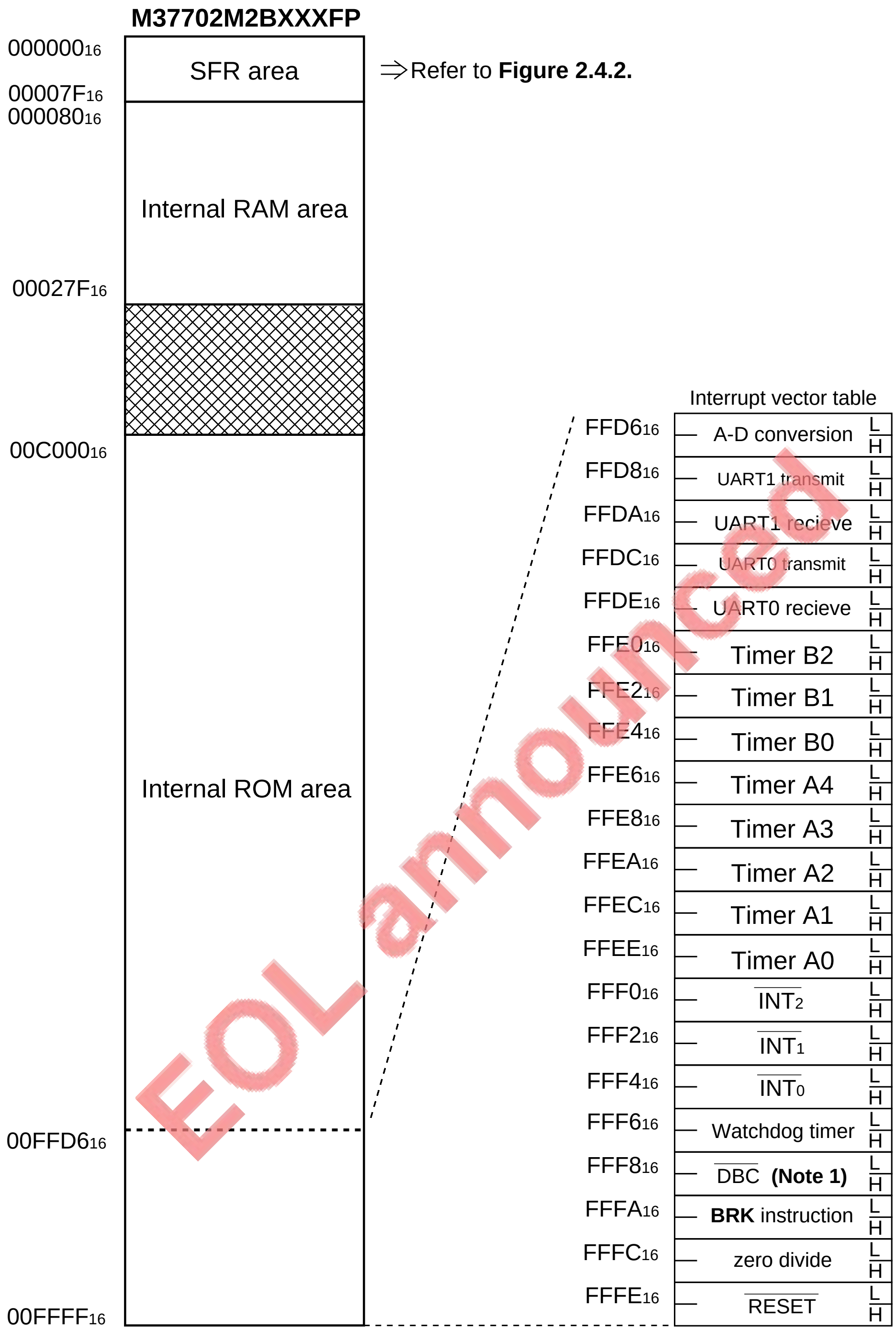
(2) Internal RAM area

The M37702M2BXXXFP (See **Note**) assigns the 512-byte static RAM at addresses 80_{16} to $27F_{16}$. 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 M37702M2BXXXFP (See **Note**) assigns the 16-Kbyte mask RAM at addresses $C000_{16}$ 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 and the external ROM version 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.



▨ **:** The internal memory is not allocated.

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

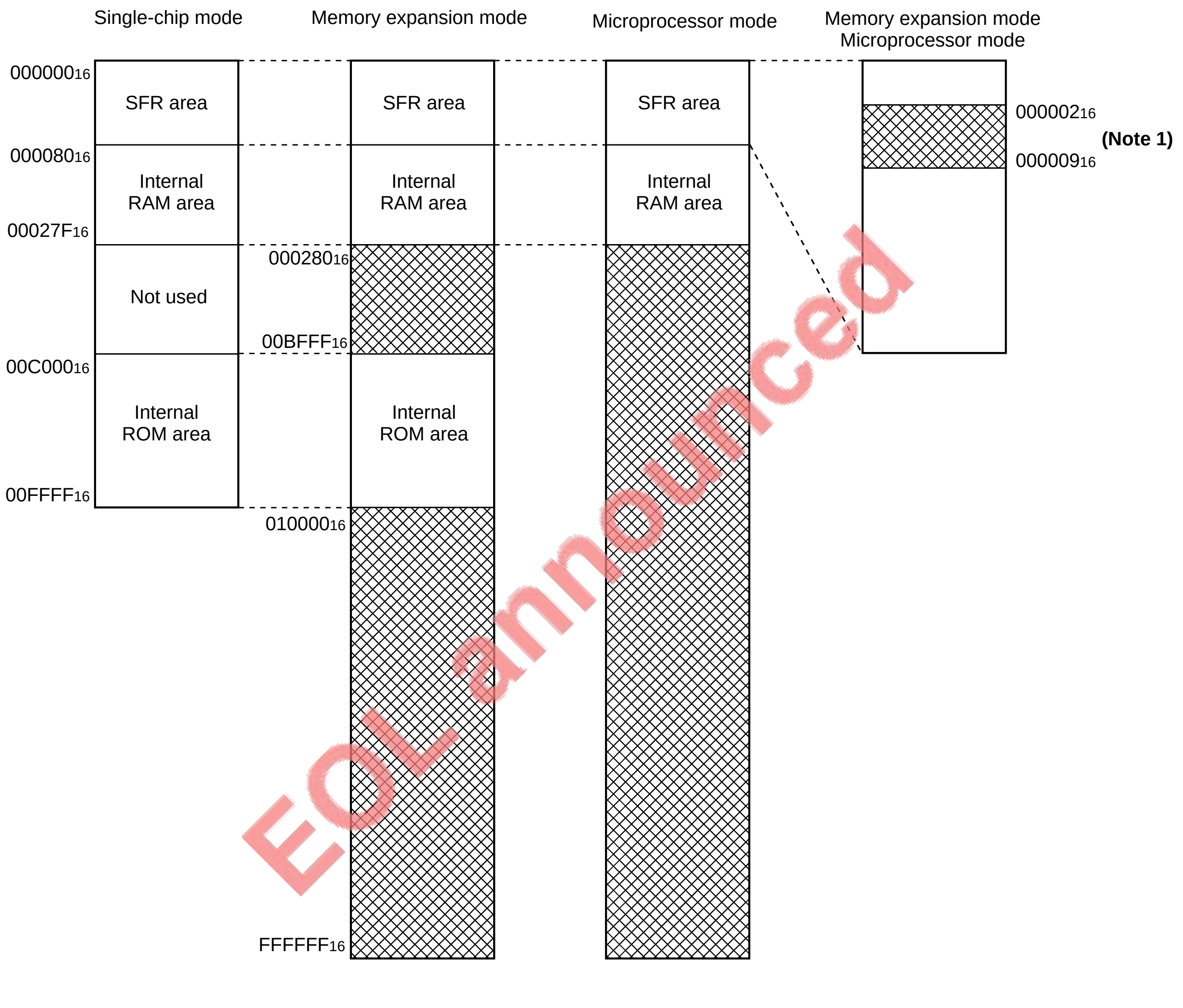
Address	
0_{16}	
1_{16}	
2_{16}	Port P0 register
3_{16}	Port P1 register
4_{16}	Port P0 direction register
5_{16}	Port P1 direction register
6_{16}	Port P2 register
7_{16}	Port P3 register
8_{16}	Port P2 direction register
9_{16}	Port P3 direction register
A_{16}	Port P4 register
B_{16}	Port P5 register
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
15_{16}	
16_{16}	
17_{16}	
18_{16}	
19_{16}	
$1A_{16}$	
$1B_{16}$	
$1C_{16}$	
$1D_{16}$	
$1E_{16}$	A-D control register
$1F_{16}$	A-D sweep pin select register
20_{16}	A-D register 0
21_{16}	
22_{16}	A-D register 1
23_{16}	
24_{16}	A-D register 2
25_{16}	
26_{16}	A-D register 3
27_{16}	
28_{16}	A-D register 4
29_{16}	
$2A_{16}$	A-D register 5
$2B_{16}$	
$2C_{16}$	A-D register 6
$2D_{16}$	
$2E_{16}$	A-D register 7
$2F_{16}$	
30_{16}	UART0 transmit/receive mode register
31_{16}	UART0 baud rate register (BRG0)
32_{16}, 33_{16}	UART0 transmit buffer register
34_{16}	UART0 transmit/receive control register 0
35_{16}	UART0 transmit/receive control register 1
36_{16}, 37_{16}	UART0 receive buffer register
38_{16}	UART1 transmit/receive mode register
39_{16}	UART1 baud rate register (BRG1)
$3A_{16}$, $3B_{16}$	UART1 transmit buffer register
$3C_{16}$	UART1 transmit/receive control register 0
$3D_{16}$	UART1 transmit/receive control register 1
$3E_{16}$, $3F_{16}$	UART1 receive buffer register

Address	
40_{16}	Count start register
41_{16}	
42_{16}	One-shot start register
43_{16}	
44_{16}	Up-down register
45_{16}	
46_{16}, 47_{16}	Timer A0 register
48_{16}, 49_{16}	Timer A1 register
$4A_{16}$, $4B_{16}$	Timer A2 register
$4C_{16}$, $4D_{16}$	Timer A3 register
$4E_{16}$, $4F_{16}$	Timer A4 register
50_{16}, 51_{16}	Timer B0 register
52_{16}, 53_{16}	Timer B1 register
54_{16}, 55_{16}	Timer B2 register
56_{16}	Timer A0 mode register
57_{16}	Timer A1 mode register
58_{16}	Timer A2 mode register
59_{16}	Timer A3 mode register
$5A_{16}$	Timer A4 mode register
$5B_{16}$	Timer B0 mode register
$5C_{16}$	Timer B1 mode register
$5D_{16}$	Timer B2 mode register
$5E_{16}$	Processor mode register
$5F_{16}$	
60_{16}	Watchdog timer register
61_{16}	Watchdog timer frequency select register
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
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. 2.4.2 SFR area's memory map

2.5 Processor modes

The M37702 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.



: External area; Accessing this area make it possible to access external connected devices.

Notes 1: Addresses 2_{16} to 9_{16} become a external area in the memory expansion mode and microprocessor mode.
2: Refer to "**Appendix 1. Memory assignment**" for products other than M37702M2BXXXFP.

Fig. 2.5.1 Memory assignment in each processor mode for M37702M2BXXXFP

2.5 Processor modes

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

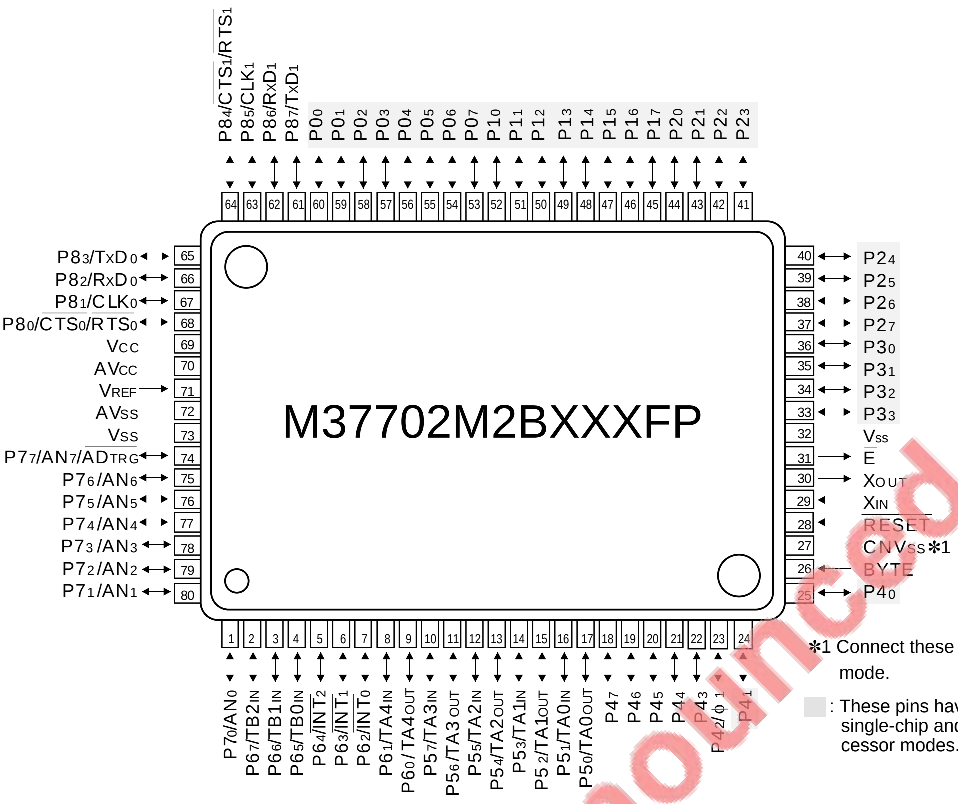
Use these modes when connecting devices externally. In these modes, an external device can be connected to any required location in the 16-Mbyte access space. For access to external devices, refer to **"Chapter 12. CONNECTION WITH EXTERNAL DEVICES."**
The memory expansion and microprocessor modes have 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 $P4_1$ when the external data bus width is 16 bits 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,"** each descriptions of internal peripheral devices and **"Chapter 12. CONNECTION WITH EXTERNAL DEVICES."**

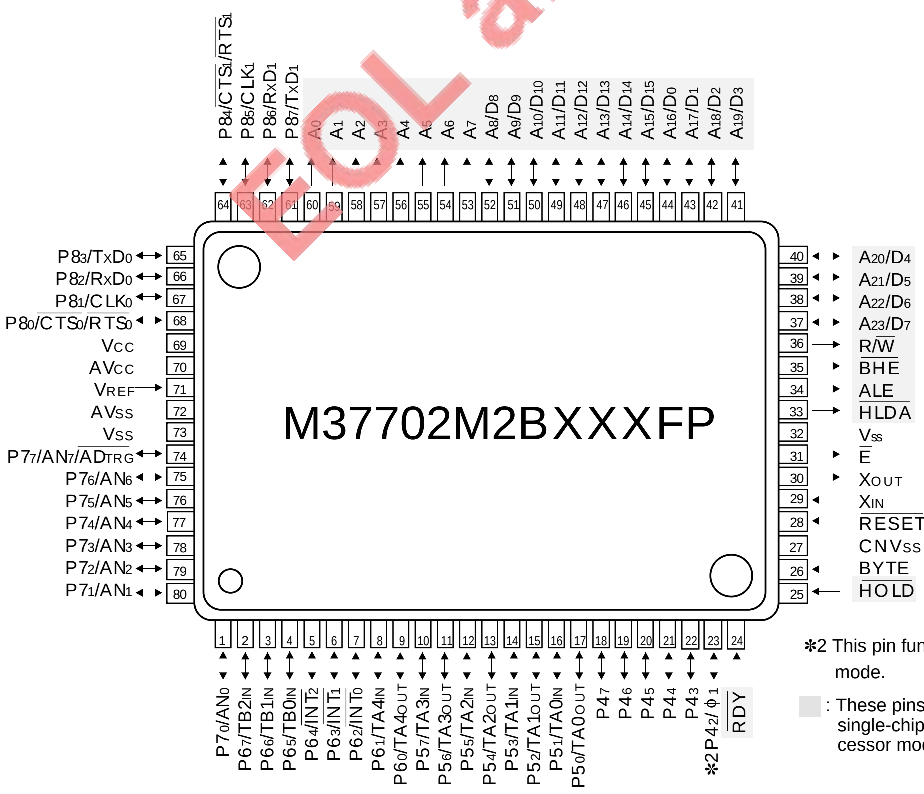
●Single-chip mode



*1 Connect these pins 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 φ1 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.	$P3_3$: HLDA (Note 4) $P3_2$: ALE $P3_1$: BHE $P3_0$: R/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) $P4_1$: RDY $P4_0$: 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.**"

4: The 7703 group does not have $P3_3$/HLDA pin.

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.

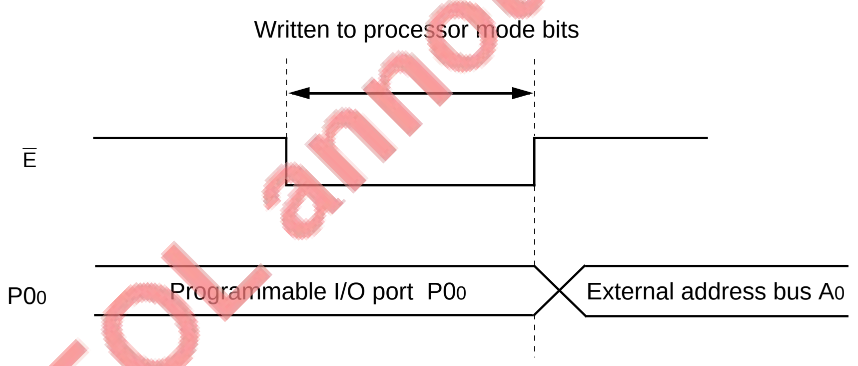
The processor mode is switched at the rising edge of signal $\overline{E}$ after writing to the processor mode bits. Figure 2.5.3 shows the timing when pin functions are switched by switching the processor mode from the single-chip mode to the memory expansion or microprocessor mode with the processor 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 6. 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 (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.

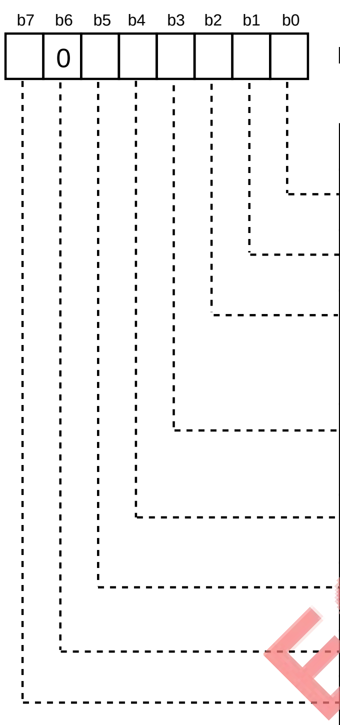
Fig. 2.5.3 Timing when pin functions are switched

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."



Processor mode register (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 **(Note1)**	RW
2	Wait bit	0 : Software Wait is inserted when accessing external area. 1 : No software wait is inserted when accessing external area.	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" 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

[Precautions when selecting processor mode]

1. For the products operating only in the single-chip mode, be sure to set the following:
   •Connect the CNVss pin with Vss.
   •Fix the processor mode bits (bits 1 and 0 at address $5E_{16}$) to "00_2."

2. The external ROM version is only for the microprocessor mode. Accordingly, be sure to set the following:
   •Connect the CNVss pin with Vcc.
   •Fix the processor mode bits (bits 1 and 0 at address $5E_{16}$) to "10_2."

3. When using the memory expansion mode or microprocessor mode, be sure to set bits 0 and 1 of the port P4 direction register to "0."
   Set the above setting whether using $P4_0/\overline{HOLD}$ pin as $\overline{HOLD}$ pin and $P4_1/\overline{RDY}$ pin as $\overline{RDY}$ pin. For also the external ROM version, set the above setting. Additionally, it is not need to set the port P0 to P3 direction registers.

MEMORANDUM

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.**"
$P4_2$ 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.

7703 Group
The 7703 Group varies with the 7702 Group in the number of pins, pins' assignment and others. Refer to the section "**Chapter 20. 7703 GROUP**."

3.1 Programmable I/O ports

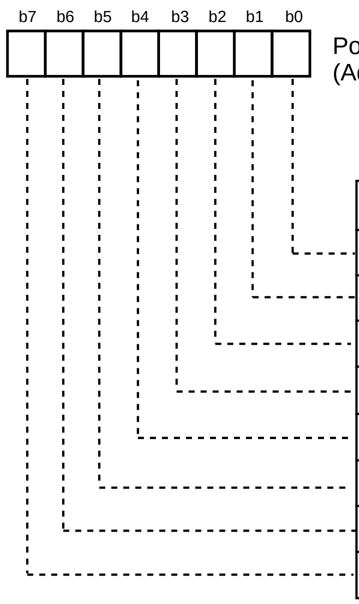
The 7702 Group has 68 programmable I/O ports, P0 to P8.
The programmable I/O ports have direction registers and port registers in the SFR area. Figure 3.1.1 shows the memory map of direction registers and port registers.

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

Notes 1: Bits 7 to 4 of the port P3 direction register cannot be written (they may be either "0" or "1") and are fixed to "0" at reading.
2: In the memory expansion mode or the microprocessor mode, fix bits 0 and 1 of the port P4 direction register to "0".

7703 Group

Fix the following bits which do not have the corresponding pin to "1".
- Bit 3 of port P3 direction register
- Bits 3 to 6 of port P4 direction register
- Bits 0, 1, 6, and 7 of port P6 direction register
- Bits 3 to 6 of port P7 direction register
- Bits 4 and 5 of port P8 direction register

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**

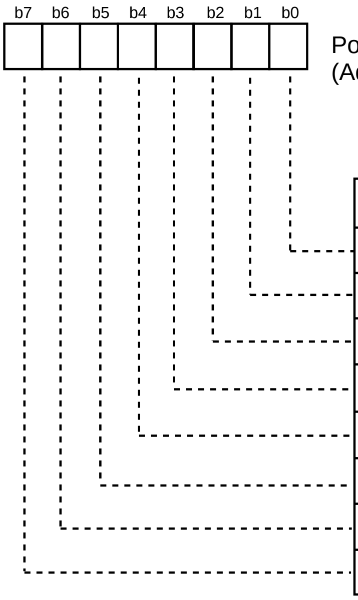
① By writing data to the corresponding bit of the port register, the data is written into the port latch.
② The data is output from the pin according to the contents of the port latch.

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 (they may be either “0” or “1”) and are fixed to “0” at reading.

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

3.1 Programmable I/O ports

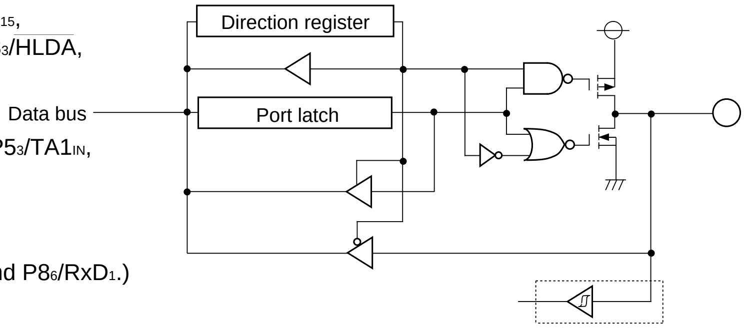
Figures 3.1.4 and 3.1.5 show the port peripheral circuits.

[Inside dotted-line not included]

Ports $P0_0/A_0$ to $P0_7/A_7$, $P1_0/A_8/D_8$ to $P1_7/A_{15}/D_{15}$, $P2_0/A_{16}/D_0$ to $P2_7/A_{23}/D_7$, $P3_0/R/\overline{W}$ to $P3_3/\overline{HLDA}$, $P4_3$ to $P4_6$

[Inside dotted-line included]

Ports $P4_0/\overline{HOLD}$, $P4_1/\overline{RDY}$, $P4_7$, $P5_1/TA0_{IN}$, $P5_3/TA1_{IN}$, $P5_5/TA2_{IN}$, $P5_7/TA3_{IN}$, $P6_1/TA4_{IN}$, $P6_2/\overline{INT_0}$ 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 $P8_6/RxD_1$.)

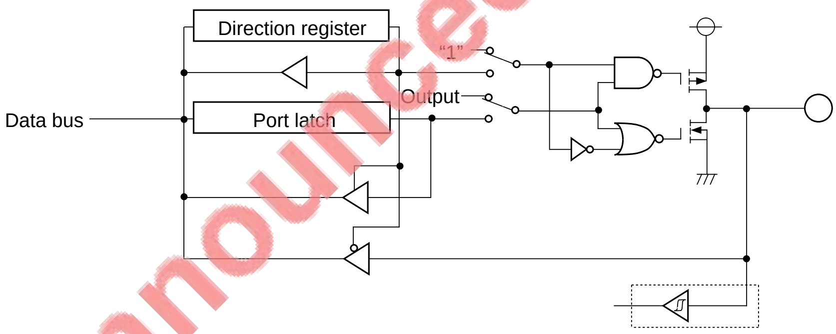


[Inside dotted-line not included.]

Ports $P4_2/\phi_1$, $P8_3/TxD_0$, $P8_7/TxD_1$

[Inside dotted-line included.]

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

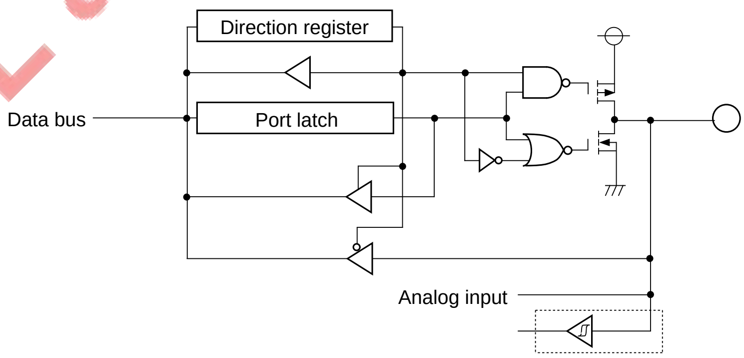


[Inside dotted-line not included.]

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

[Inside dotted-line included.]

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

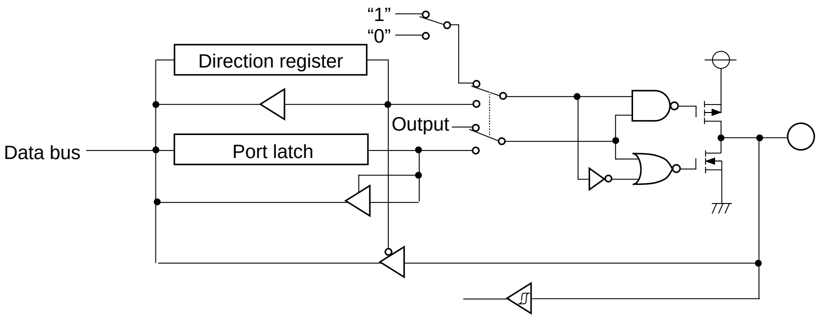


7703 Group

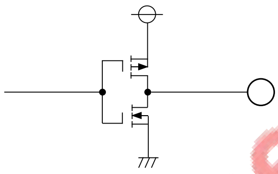
There are not pins P33, P43 – P46, P60, P61, P66, P67, P73 – P76.

Fig. 3.1.4 Port peripheral circuits (1)

Ports $P8_0/\overline{CTS_0}/\overline{RTS_0}$, $P8_1/CLK_0$,
$P8_4/\overline{CTS_1}/\overline{RTS_1}$, $P8_5/CLK_1$



$\overline{E}$ output pin



7703 Group

There are not pins $P8_4$ and $P8_5$.

Fig. 3.1.5 Port peripheral circuits (2)

3.2 I/O pins of internal peripheral devices

$P4_2$ 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 devices. 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

INTERRUPTS

4.1 Overview

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 M37702 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.

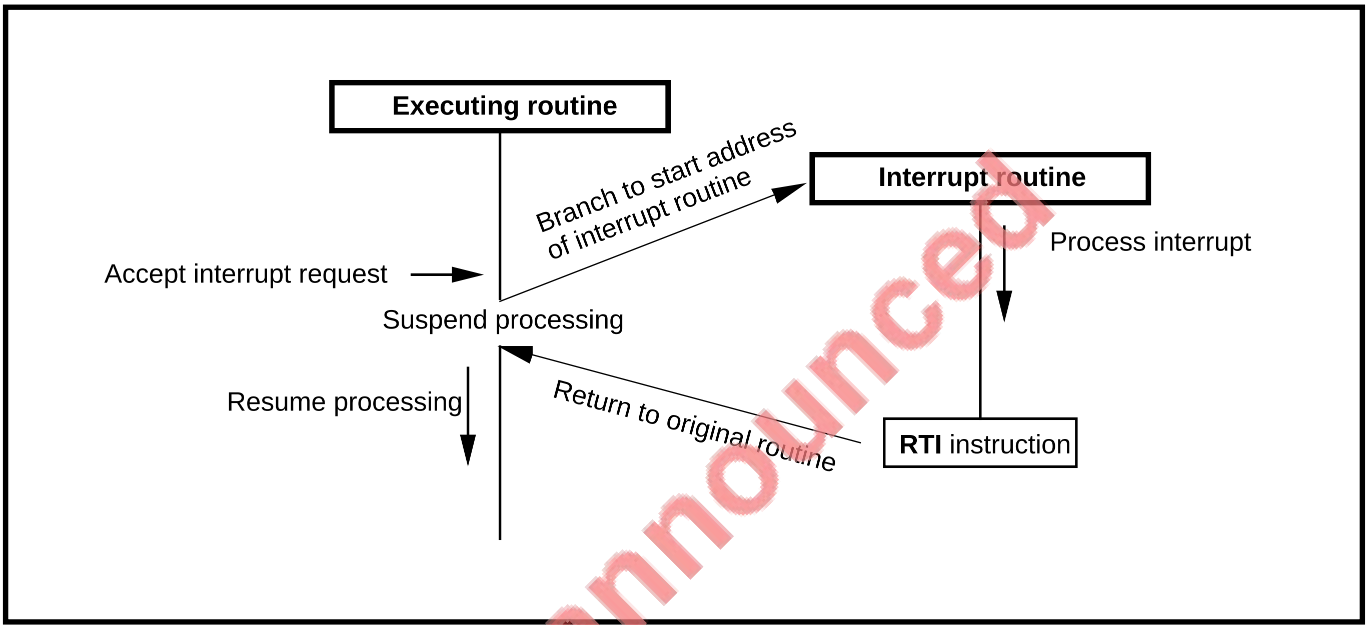
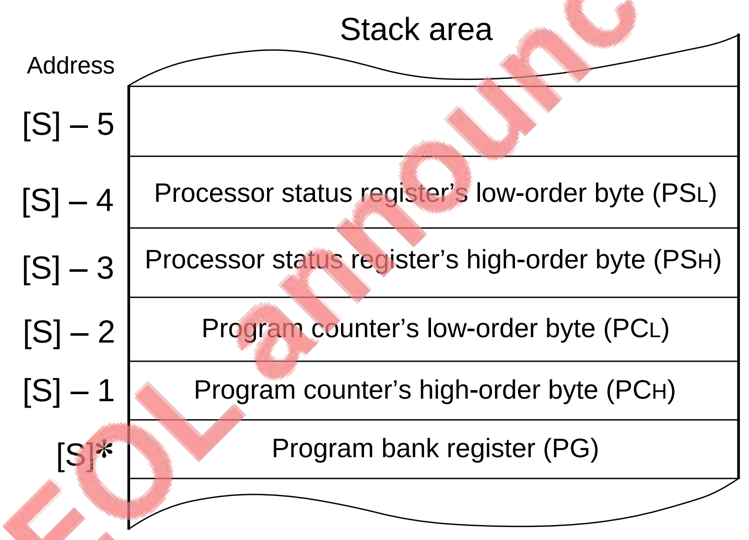


Fig. 4.1.1 Interrupt processing sequence

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.



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

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 **"7700 Family Software Manual."**)
BRK instruction interrupt	Occurs when the **BRK** instruction is executed. (Refer to **"7700 Family 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 allocated.		Undefined	–

Note: Use the **SEB** or **CLB** instruction to set each interrupt control register.

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 1)**	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 allocated.		Undefined	–

Notes 1: The $\overline{INT_0}$ to $\overline{INT_2}$ interrupt request bits are invalid when selecting the level sense.
2: Use the **SEB** or the **CLB** instruction to set the $\overline{INT_0}$ to $\overline{INT_2}$ interrupt control registers.

Fig. 4.3.2 Structure of interrupt control register

4.3 Interrupt control

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. Use the **SEB** or **CLB** instruction to set this bit.
For the $\overline{INT_i}$ interrupt request bit (i = 0 to 2), when using the $\overline{INT_i}$ 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, <u>an interrupt can be disabled by setting its interrupt priority level to 0.</u>

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.

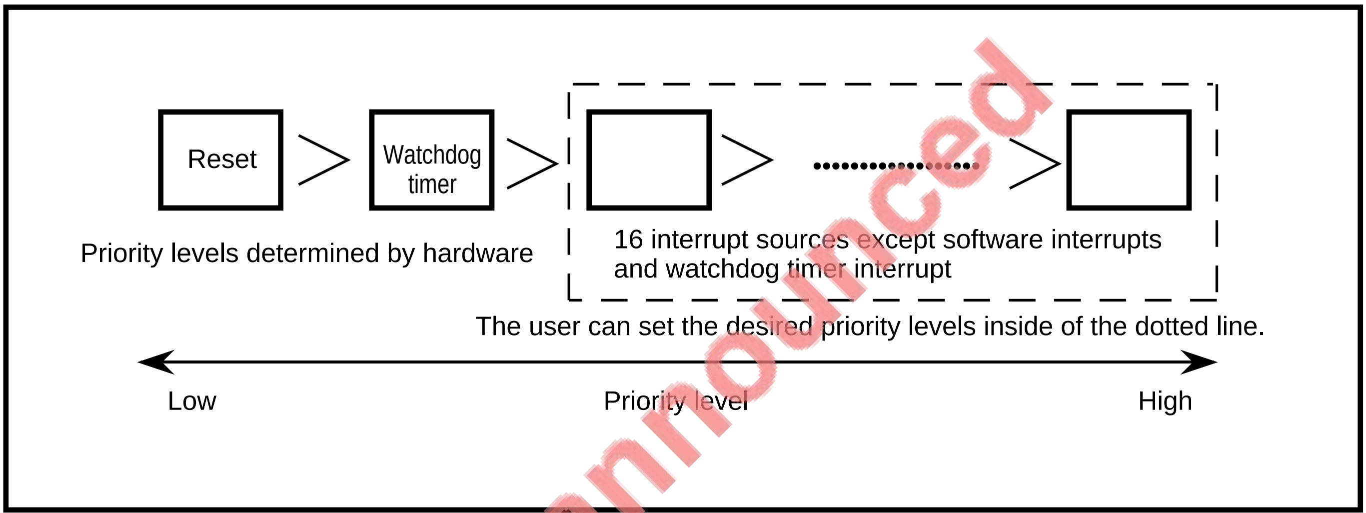


Fig. 4.4.1 Interrupt priority levels set by hardware

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.

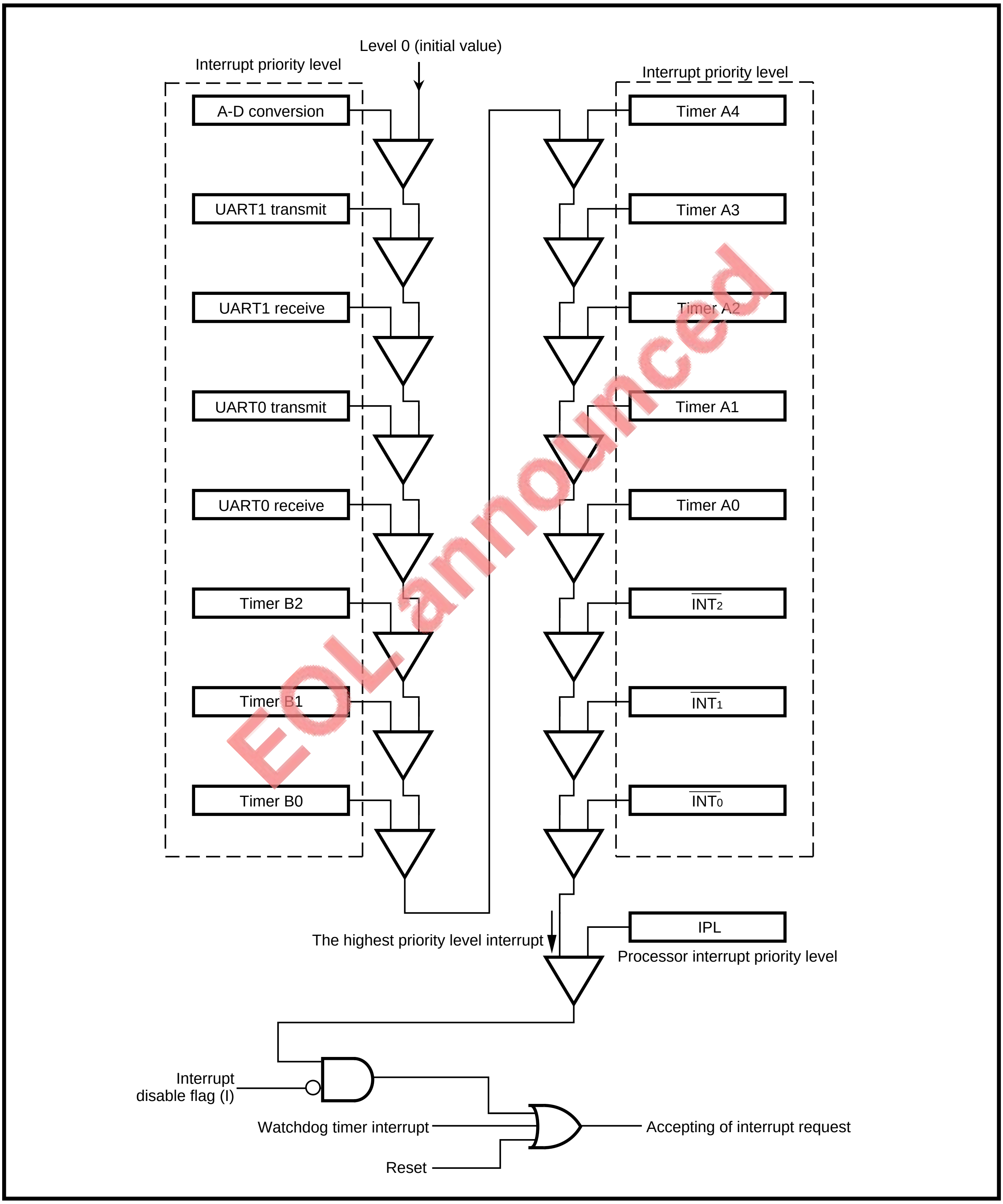


Fig. 4.5.1 Interrupt priority level detection circuit

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.

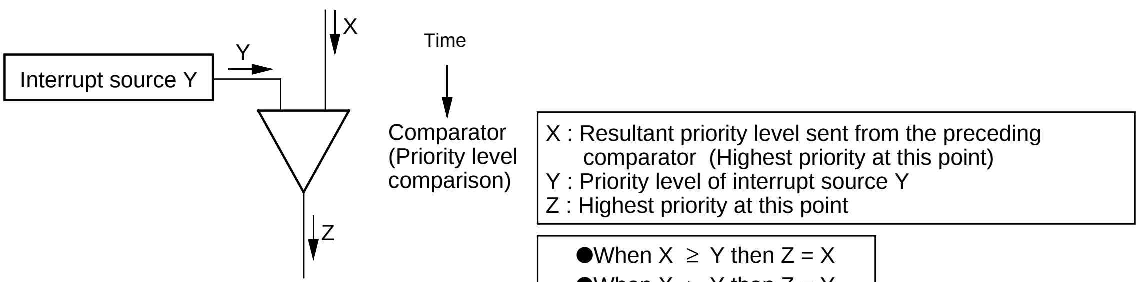


Fig. 4.5.2 Interrupt priority level detection model

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."

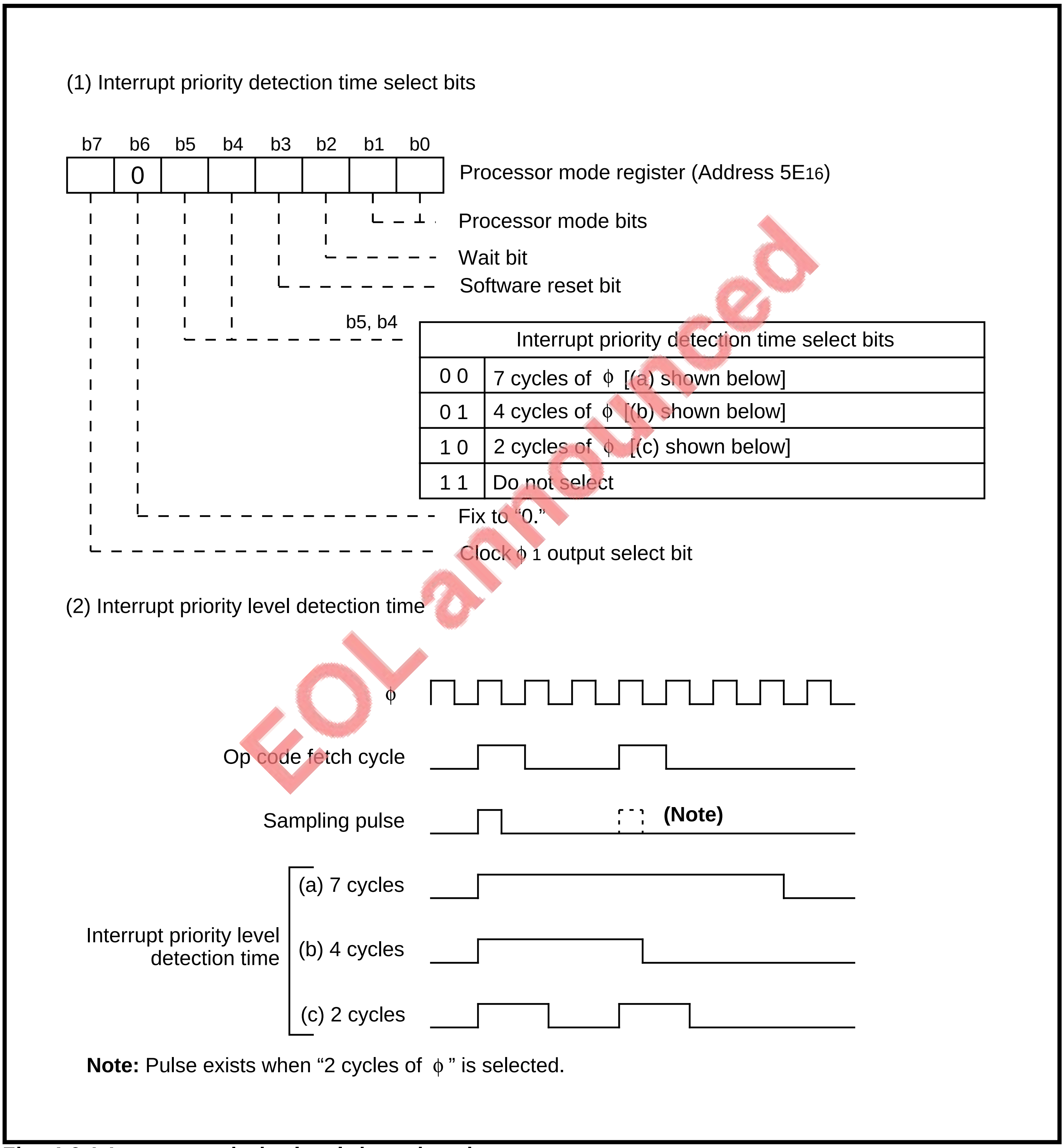


Fig. 4.6.1 Interrupt priority level detection time

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.

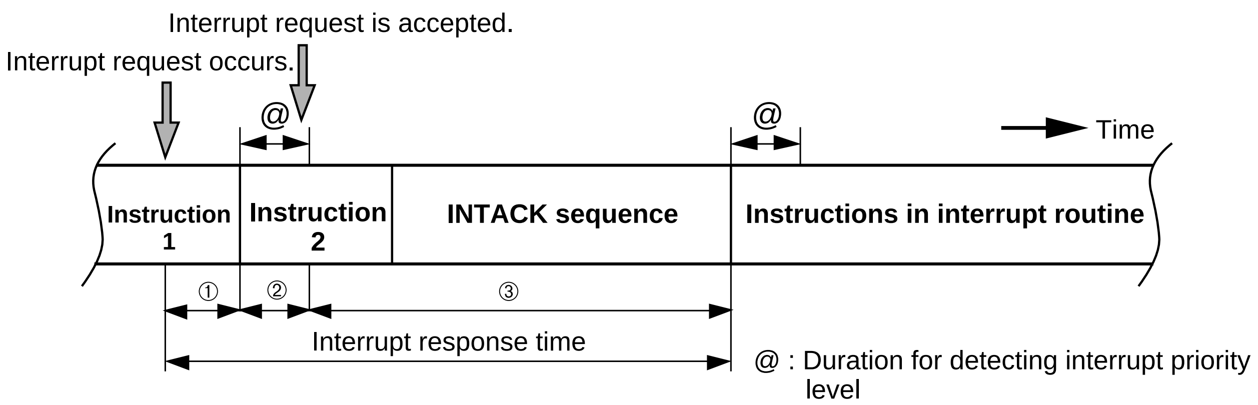
After execution of an instruction at accepting the interrupt request is completed, an INTACK (Interrupt Acknowledge) sequence is executed, and a branch is made to the start address of the interrupt routine allocated in addresses 0_{16} to $FFFF_{16}$.

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 13 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.

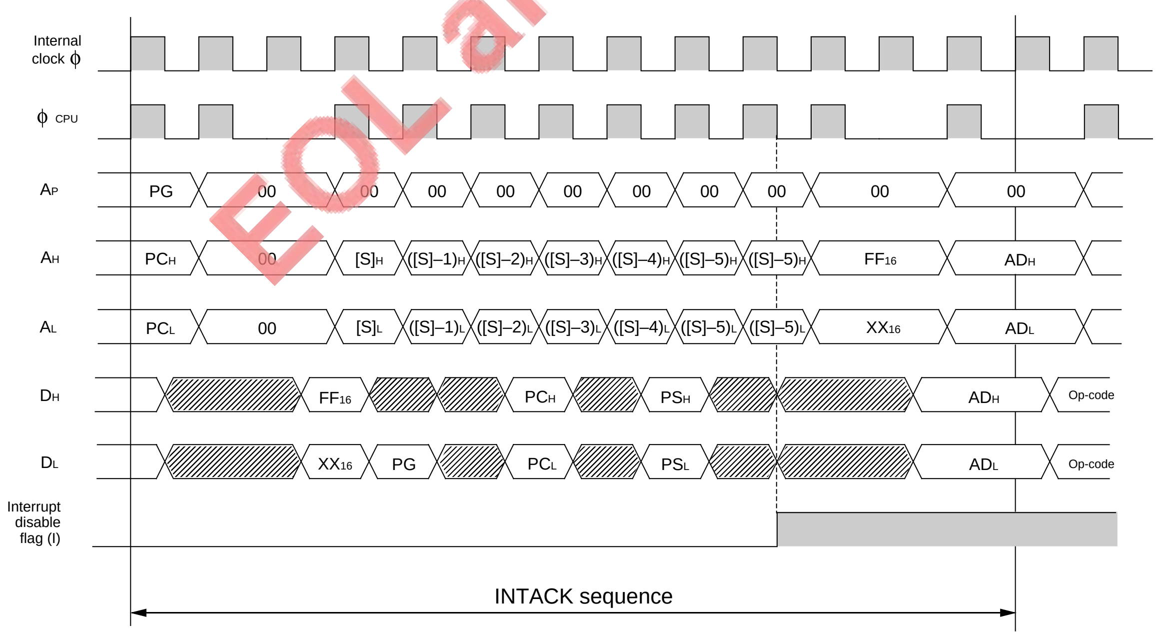
② Time from the instruction next to ① **(Note)** until the completion of executing an instruction which is being done at the end of priority detection

Note : At this time, interrupt priority detection starts.

③ Time required to execute the INTACK sequence (13 cycles of ϕ at minimum)

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

●When stack pointer (S)'s contents is even and no Wait



ϕ CPU : CPU standard clock
AP : High-order 8 bits of CPU internal address bus
AH : Middle-order 8 bits of CPU internal address bus
AL : Low-order 8 bits of CPU internal address bus
DH : CPU internal data bus for odd address
DL : CPU internal data bus for even address

(hatched) : Not used
[S] : Contents of stack pointer (S)
XX16 : Low-order 8 bits of vector address
ADH : Contents of vector address (High-order address)
ADL : Contents of vector address (Low-order 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

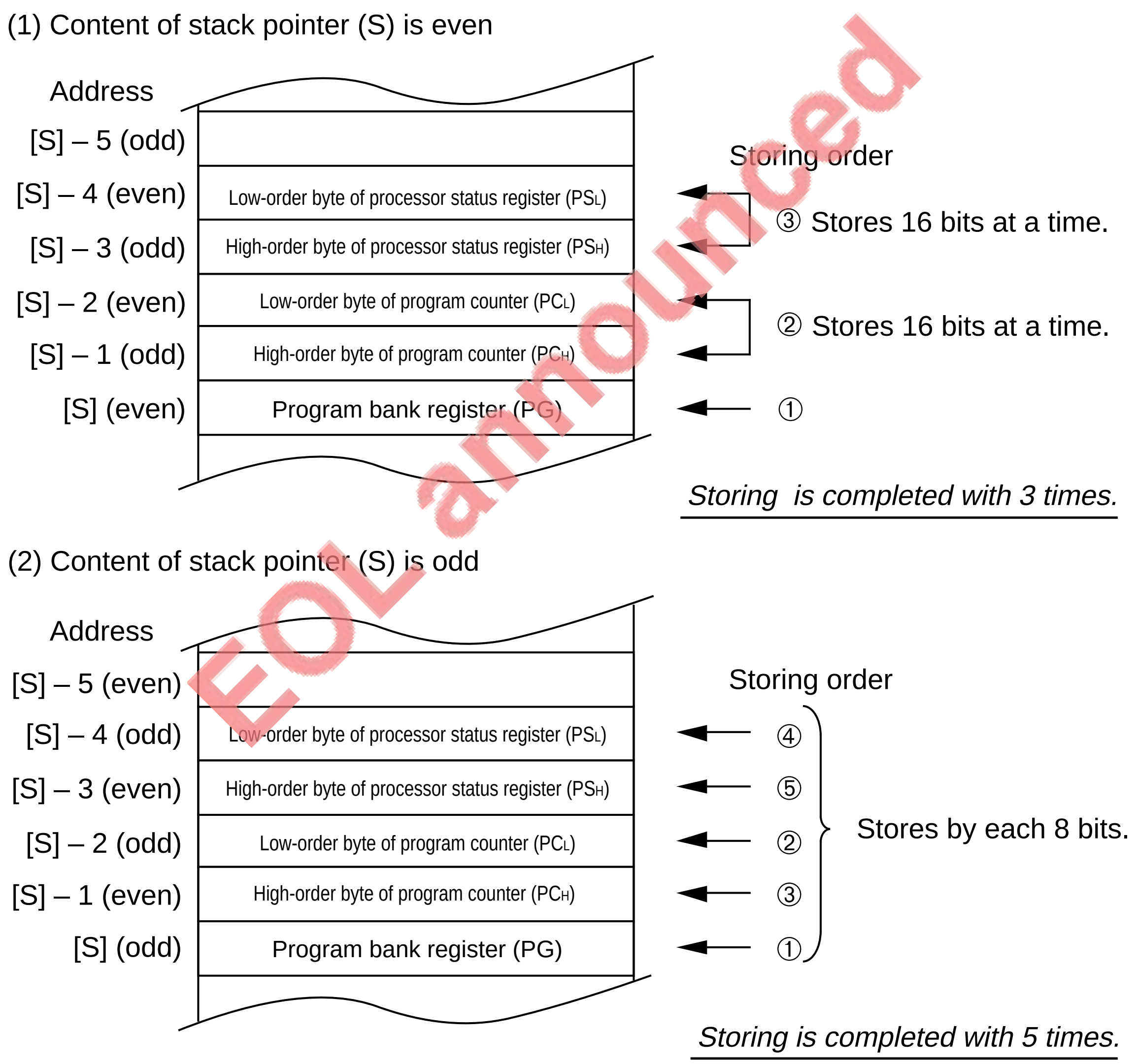
When an interrupt request is accepted, the processor interrupt priority level (IPL) is replaced with the interrupt priority level of the accepted interrupt. This results in easy control of multiple interrupts. (Refer to section **"4.9 Multiple interrupts."**)
When at reset or the watchdog timer or the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

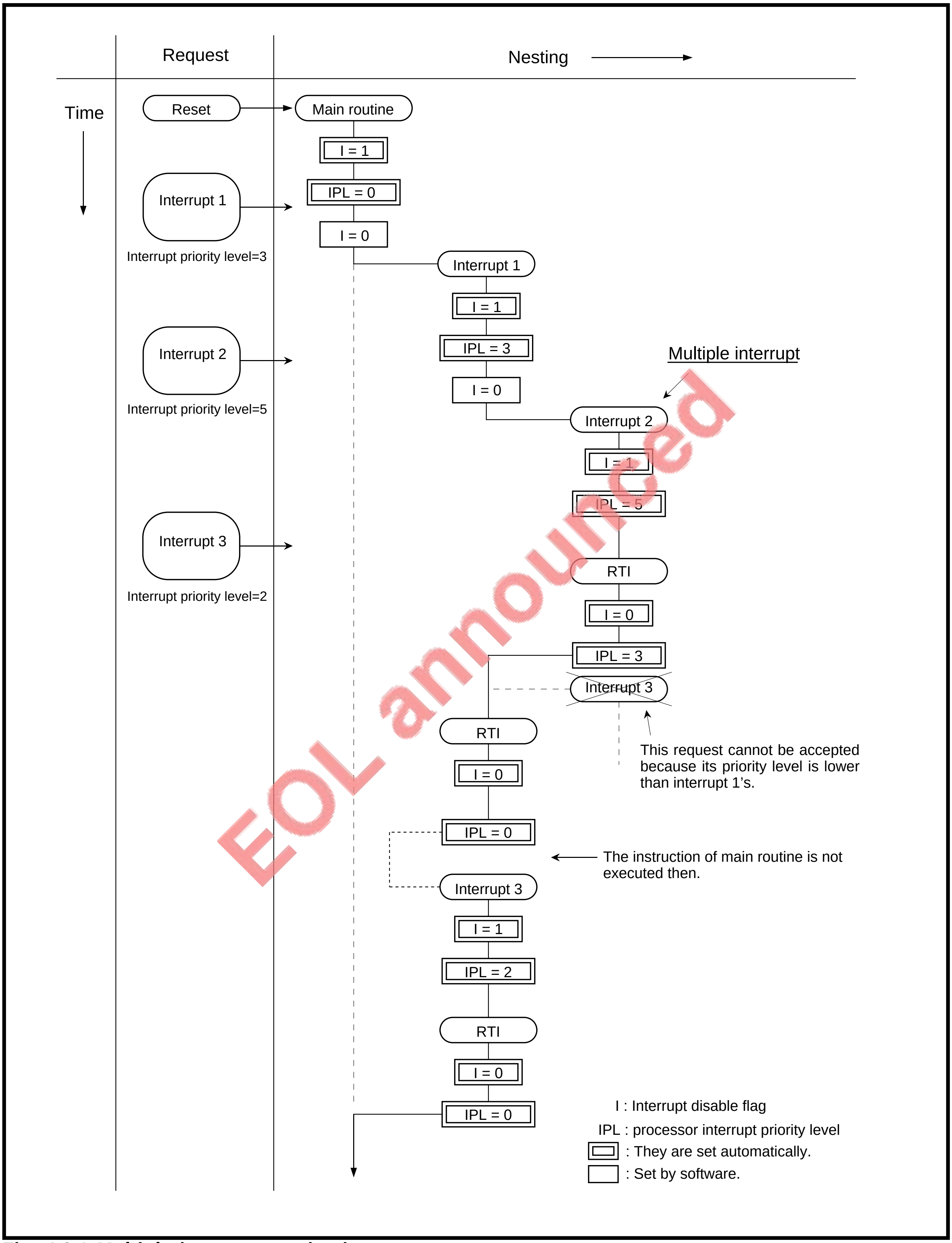


Fig. 4.9.1 Multiple interrupt mechanism

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

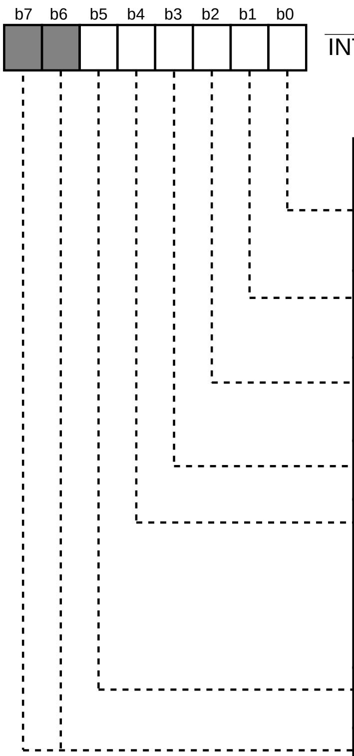
An external interrupt request occurs by input signals to the $\overline{INT_i}$ (i = 0 to 2) pin. The occurrence factor of interrupt request can be selected by the level sense/edge sense select bit and the polarity select bit (bits 5 and 4 at addresses $7D_{16}$ to $7F_{16}$) shown in Figure 4.10.1. 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.



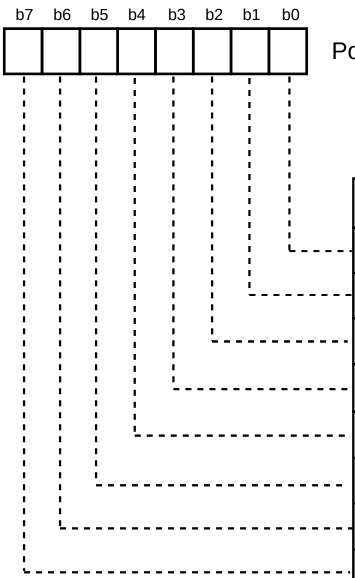
$\overline{INT_0}$ to $\overline{INT_2}$ interrupt control registers (Addresses $7D_{16}$ to $7F_{16}$)

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 1)**	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 allocated.		Undefined	–

Notes 1: The $\overline{INT_0}$ to $\overline{INT_2}$ interrupt request bits are invalid when selecting the level sense.
2: Use the **SEB** or **CLB** instruction to set the $\overline{INT_0}$ to $\overline{INT_2}$ interrupt control registers.

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

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



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 $\overline{\text{INT}}_i$ 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 $\overline{\text{INT}}_i$ interrupt request bit becomes ignored.
In this case, the interrupt request occurs continuously while the level of the $\overline{\text{INT}}_i$ pin is valid level*1. When the $\overline{\text{INT}}_i$ pin level changes from the valid level to the invalid level*2 before the $\overline{\text{INT}}_i$ 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.

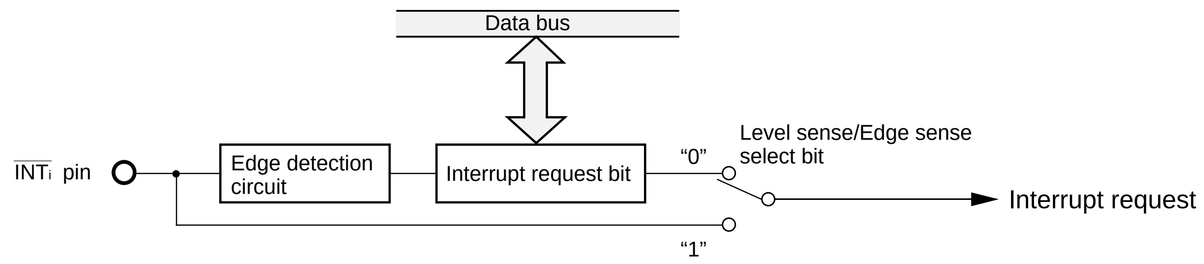


Fig. 4.10.3 Circuit of $\overline{\text{INT}}_i$ Interrupt

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

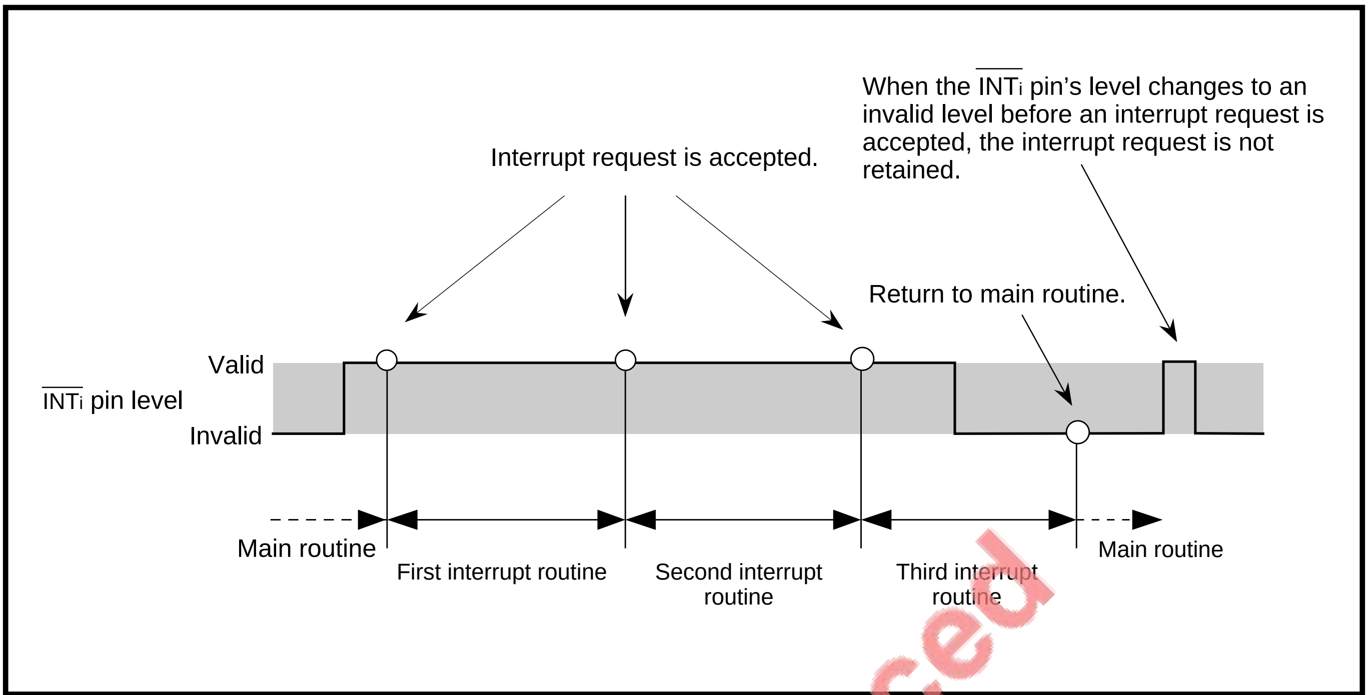
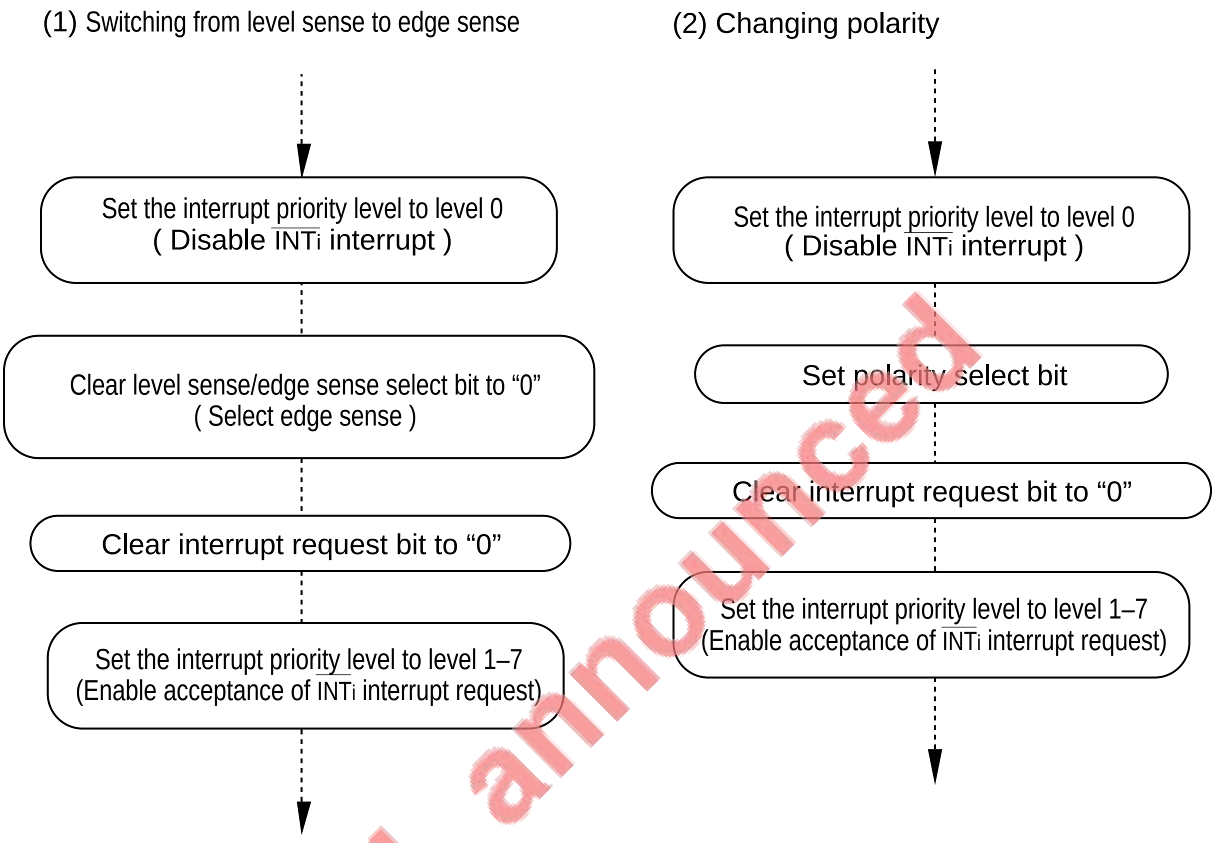


Fig. 4.10.4 Occurrence of $\overline{INT_i}$ interrupt request in level sense mode

4.10.2 Switch of occurrence factor of $\overline{INT_i}$ interrupt request

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



Notes 1: Use the **SEB** or **CLB** instruction when setting the $\overline{INT_i}$ interrupt control register (i = 0 to 2).

2: Perform the above setting separately. Do not perform 2 or more setting at the same time, with 1 instruction.

Fig. 4.10.5 Switching flow of occurrence factor of $\overline{INT_i}$ interrupt request

4.11 Precautions when using interrupts

1. Use the **SEB** or **CLB** instruction when setting the interrupt control registers (addresses 70_{16} to $7F_{16}$.)
2. 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.

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

5.1 Overview

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.

7703 Group

Timer A4's function of the 7703 Group varies from the 7702 Group's. Refer to "**Chapter 20. 7703 GROUP**."

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.
However, for the following registers, refer to the relevant section:

•Up-down register (address 44_{16}) .... "**5.4.2 Operation in event counter mode**"
•One-shot start register (address 42_{16}) .... "**5.5.3 Trigger**"

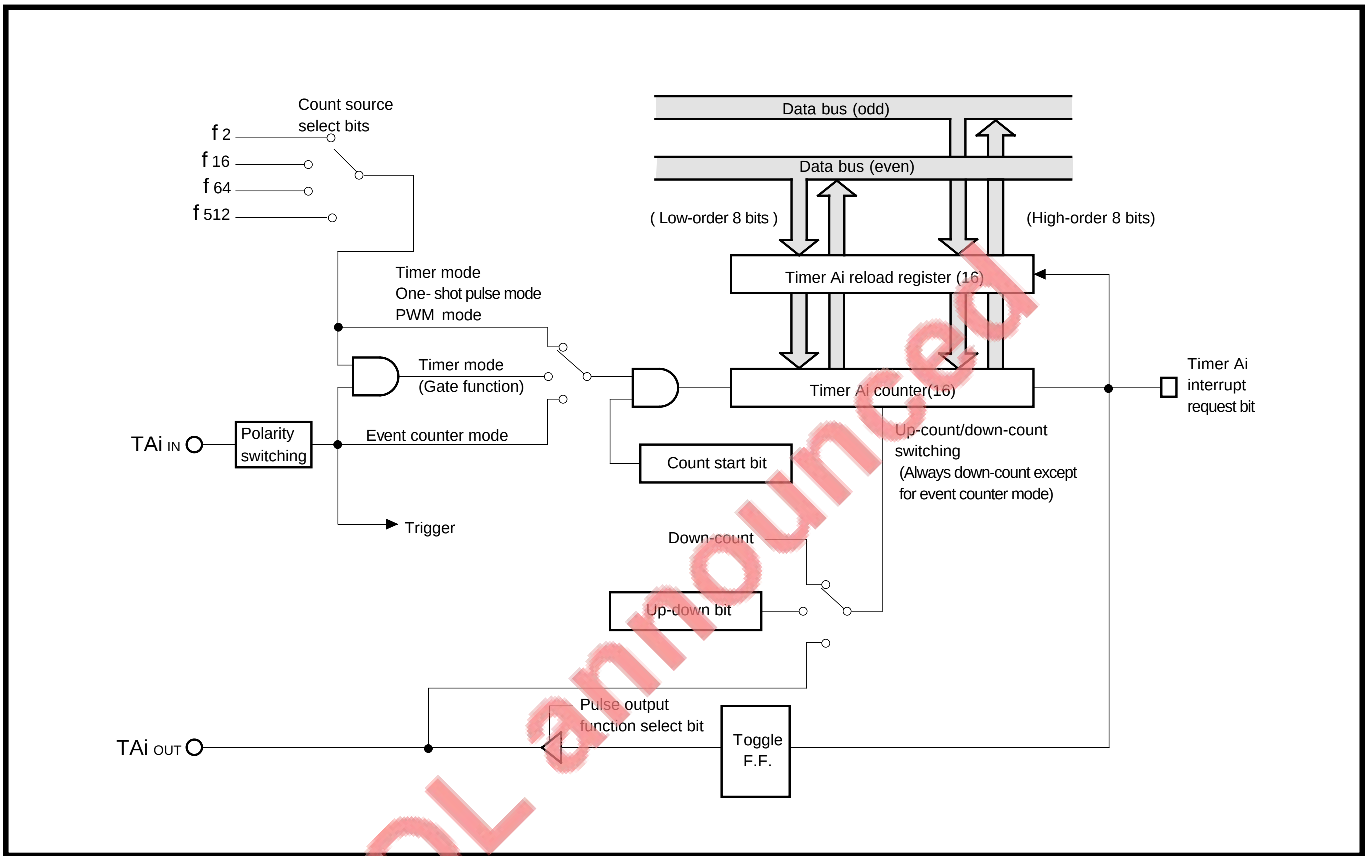


Fig. 5.2.1 Block diagram of Timer A

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 an 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 Block description

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.

b7 b6 b5 b4 b3 b2 b1 b0

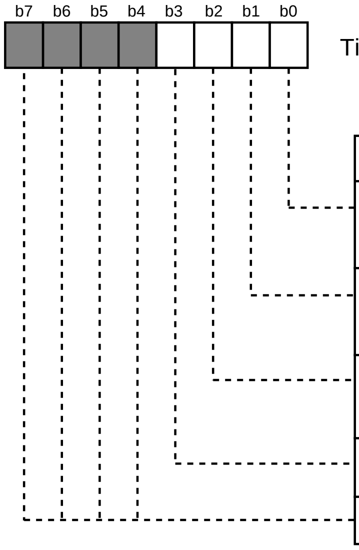
Timer Ai mode register (i = 0 to 4) (Addresses 56_{16} to $5A_{16}$)

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	–

Note: Use the **SEB** or **CLB** instruction to set each interrupt control register.

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

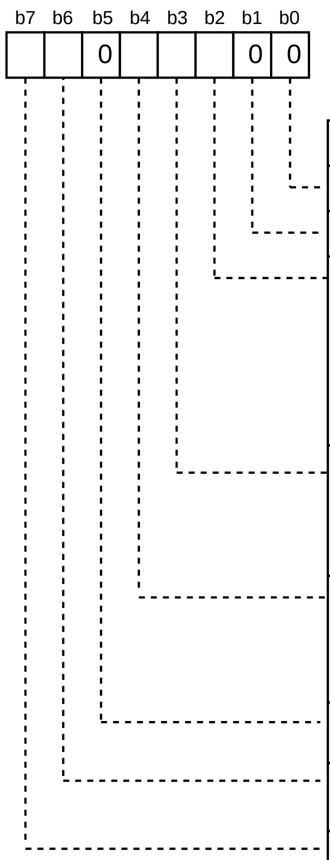
5.3 Timer mode

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

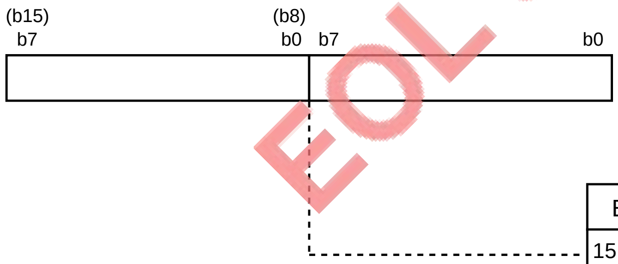
Item	Specifications
Count source	f_2, f_{16}, f_{64}, or f_{512}
Count operation	• Down-count • When the counter underflows, reload register's contents are reloaded and counting continues.
Divide ratio	$\frac{1}{(n+1)}$ n : Timer Ai register setting value
Count start condition	When count start bit is 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.)

5.3 Timer 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	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	Gate function select bits	b4 b3 0 0 : } No gate function 0 1 : } (TAiIN pin functions as a programmable I/O port.) 1 0 : Gate function (Counter counts only while TAiIN pin's input signal is "L" level.) 1 1 : Gate function (Counter counts only while TAiIN 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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	0	RW
7			0	RW



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."**

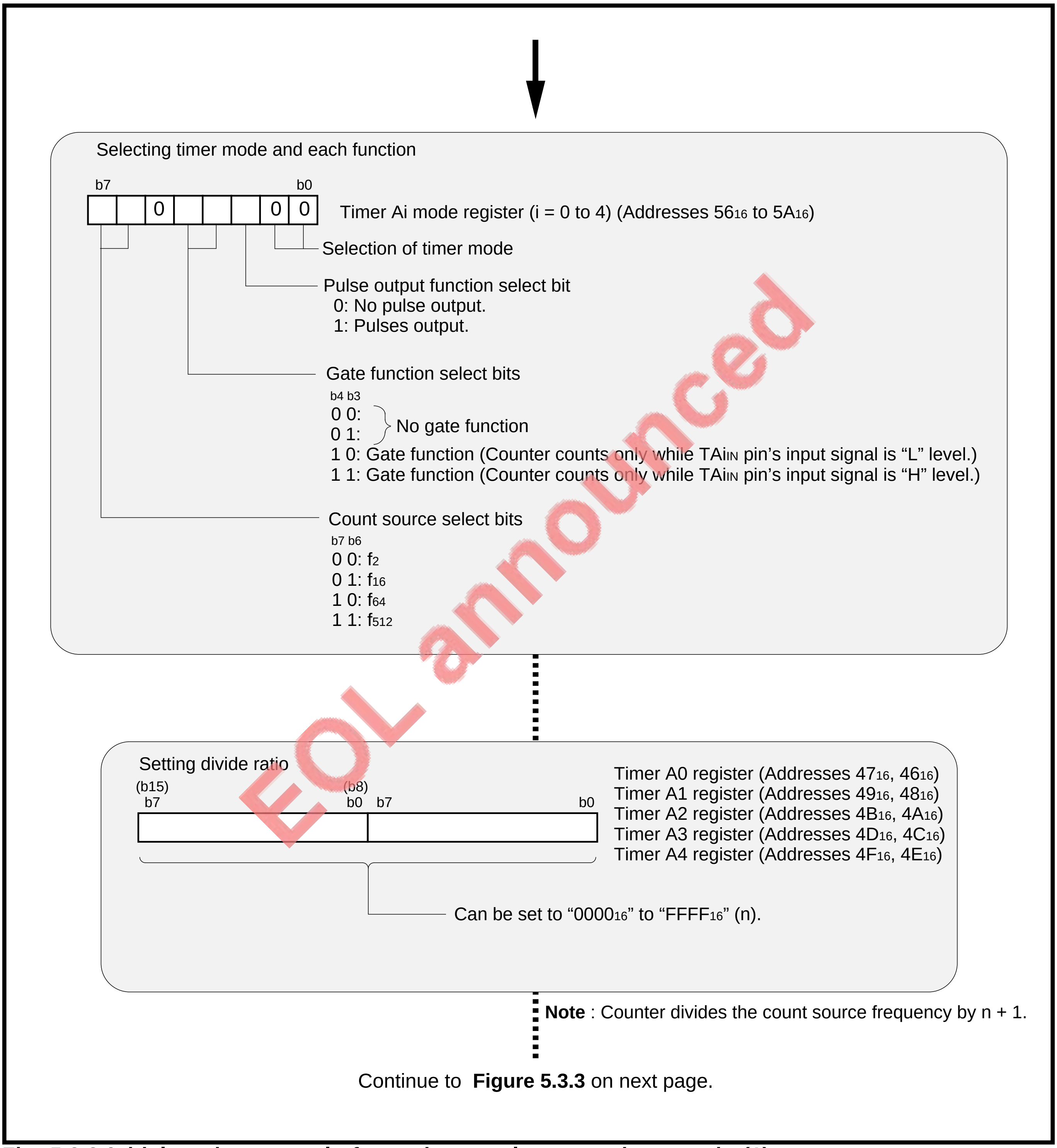


Fig. 5.3.2 Initial setting example for registers relevant to timer mode (1)

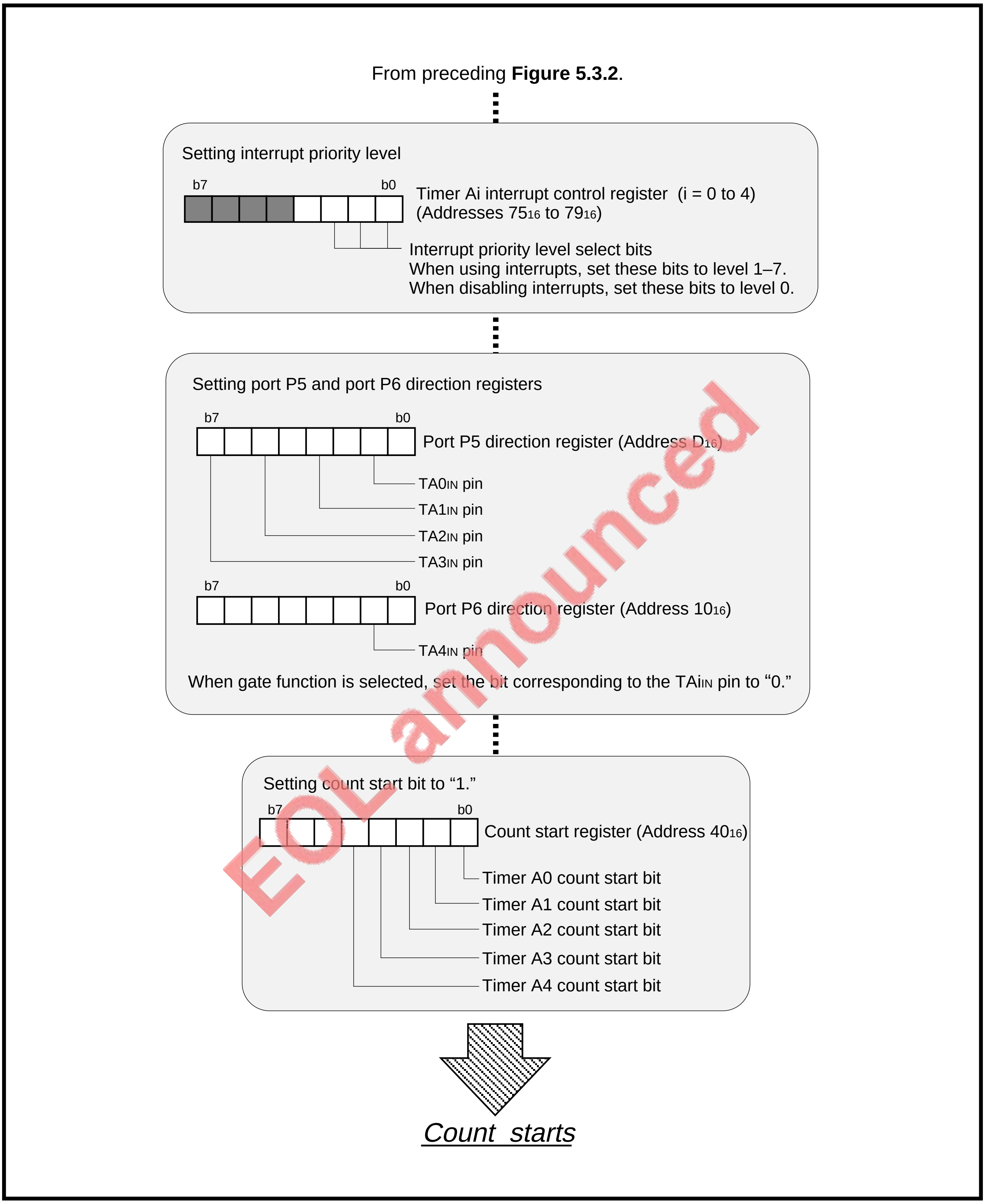


Fig. 5.3.3 Initial setting example for registers relevant to timer mode (2)

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 b7	Count source select bits b6	Count source	Count source frequency f(X_{IN}) = 8 MHz	f(X_{IN}) = 16 MHz	f(X_{IN}) = 25 MHz
0	0	f_2	4 MHz	8 MHz	12.5 MHz
0	1	f_{16}	500 kHz	1 MHz	1.5625 MHz
1	0	f_{64}	125 kHz	250 kHz	390.625 kHz
1	1	f_{512}	15625 Hz	31250 Hz	48.8281 kHz

5.3 Timer mode

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.

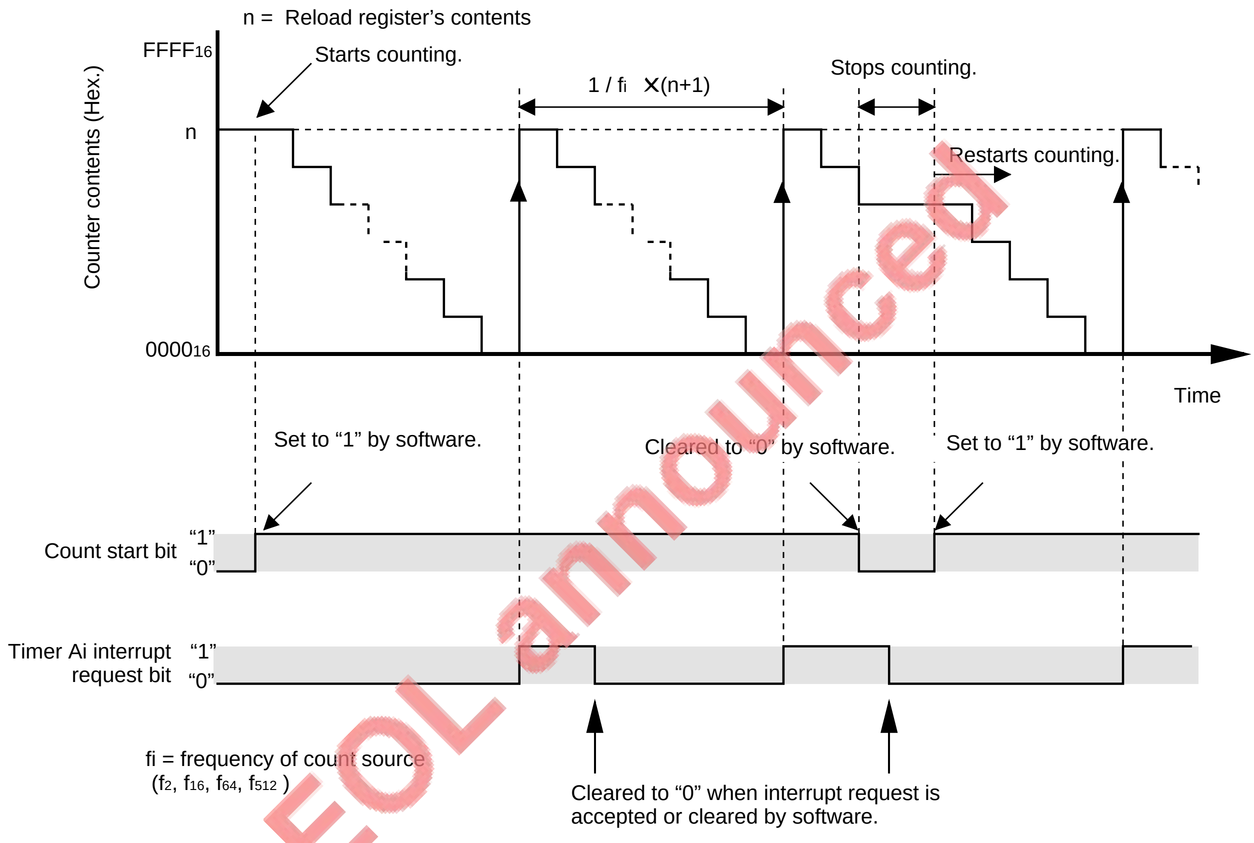


Fig. 5.3.4 Example of operation in timer mode (without pulse output and gate functions)

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.

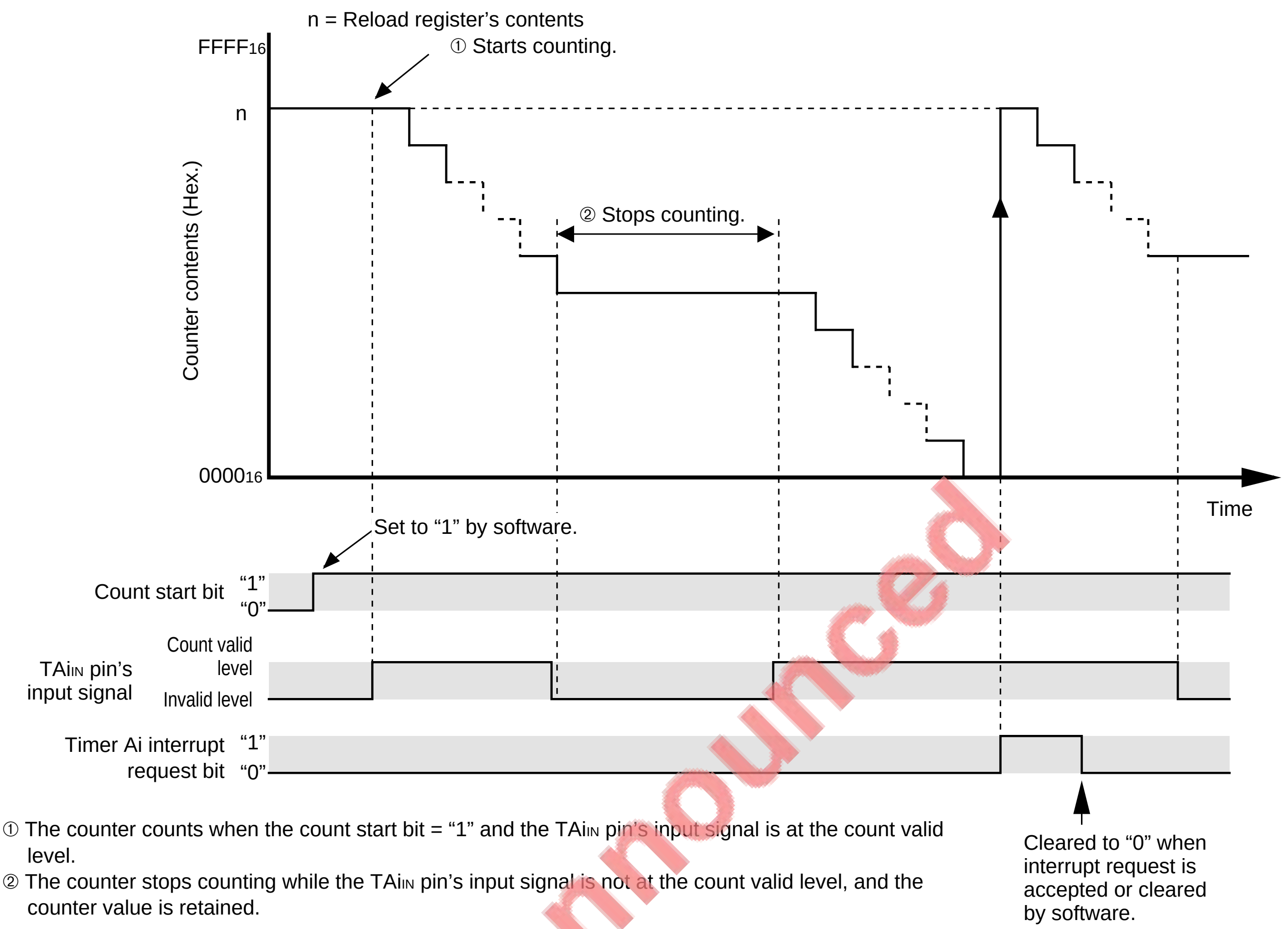
Figure 5.3.5 shows an example of operation selecting the gate function.

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.



① The counter counts when the count start bit = "1" and the TAiIN pin's input signal is at the count valid level.

② The counter stops counting while the TAiIN pin's input signal is not at the count valid level, and the counter value is retained.

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.

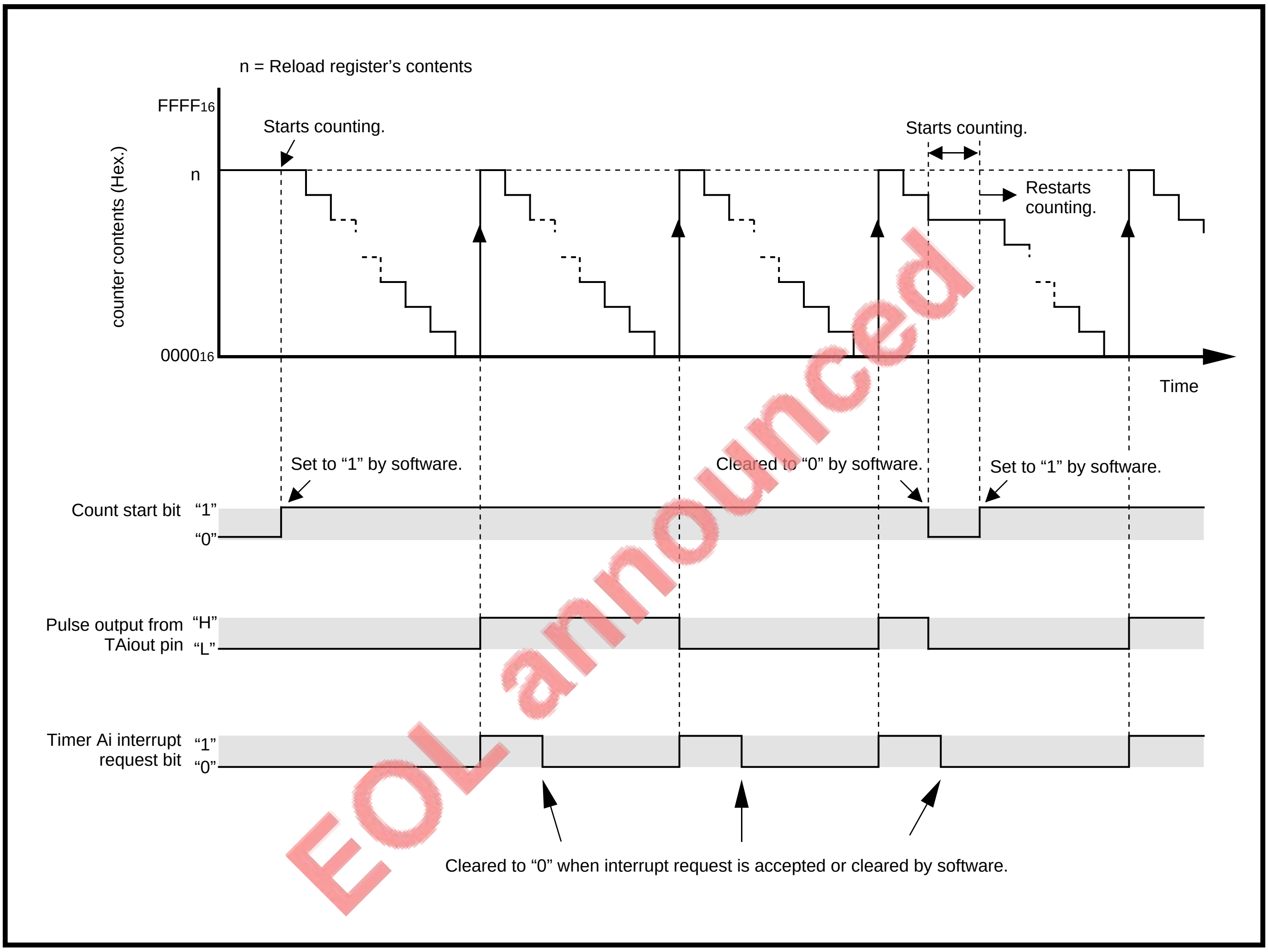


Fig. 5.3.6 Example of operation selecting pulse output function

5.3 Timer mode

[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.

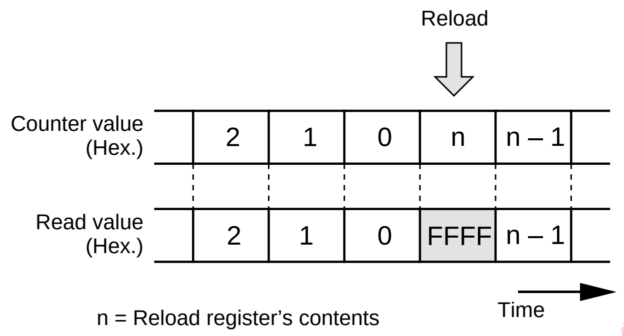


Fig. 5.3.7 Reading timer Ai register

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.

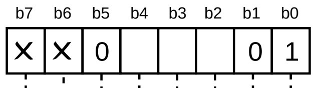
Table 5.4.1 Specifications of event counter mode (when not using two-phase pulse signal processing function)

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.
Divide ratio	● For down-count $\frac{1}{(n+1)}$ ● For up-count $\frac{1}{(FFFF_{16} - n + 1)}$ n: Timer Ai register setting value
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)

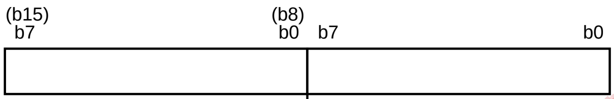
Item	Specifications
Count source	External signal (two-phase pulse) input to the TAj_{IN} or TAj_{OUT} pin (j = 2 to 4)
Count operation	● Up-count or down-count can be switched by external signal (two-phase pulse). ● When the counter overflows or underflows, reload register's contents are reloaded and counting is continued.
Divide ratio	● For down-count $\frac{1}{(n + 1)}$ ● For up-count $\frac{1}{(FFFF_{16} - n + 1)}$ n: Timer Aj register setting value
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.)



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."



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."**

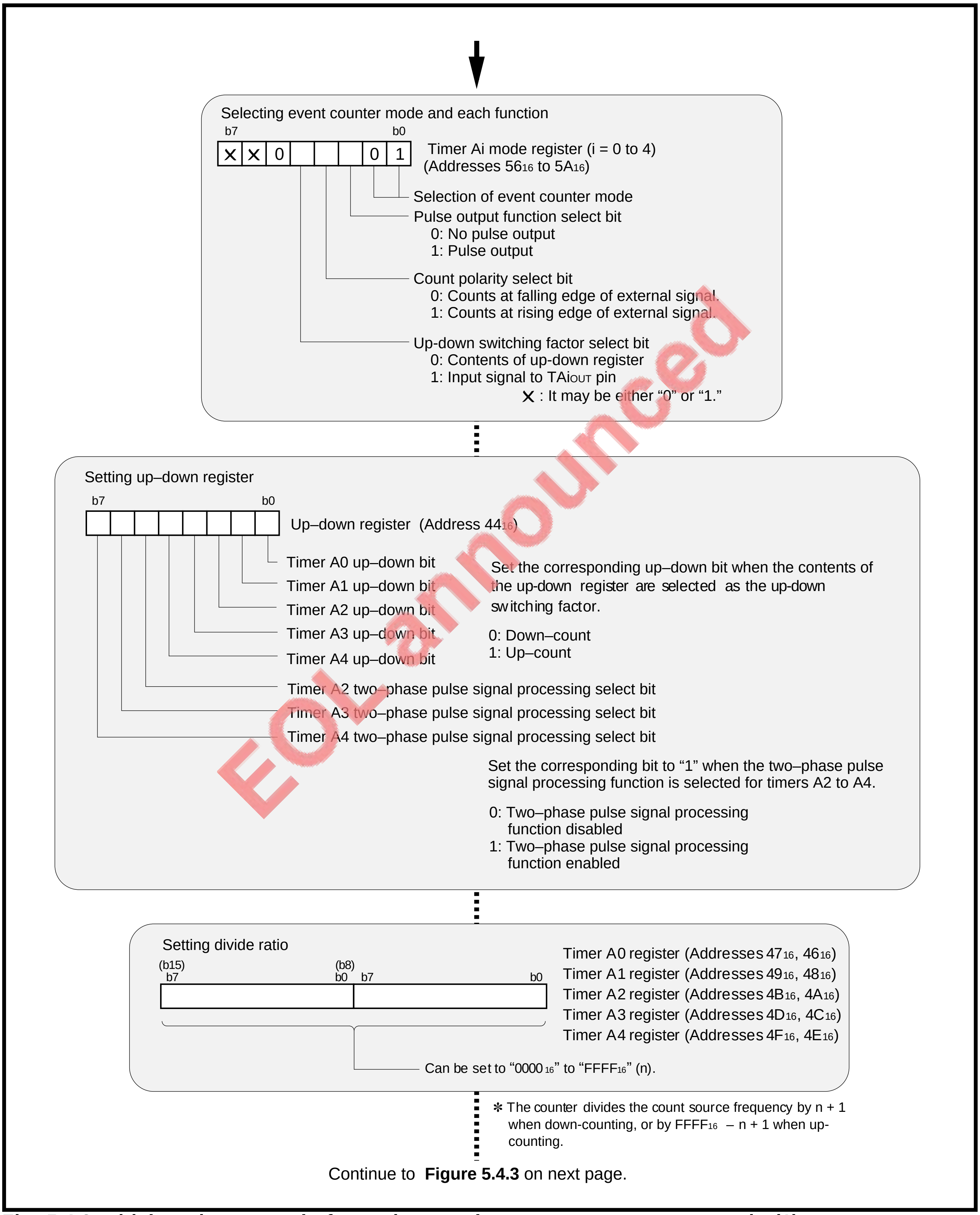


Fig. 5.4.2 Initial setting example for registers relevant to event counter mode (1)

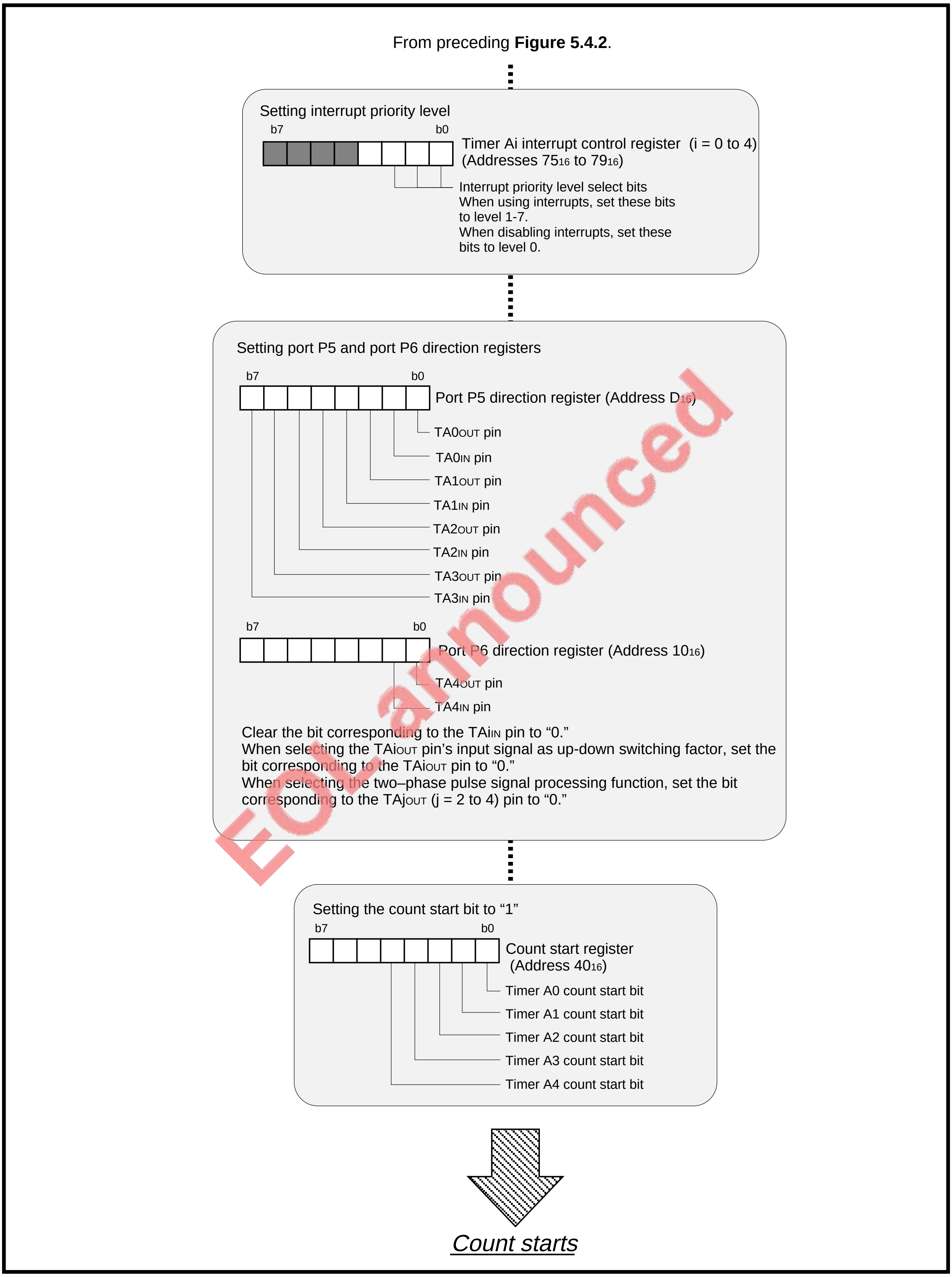
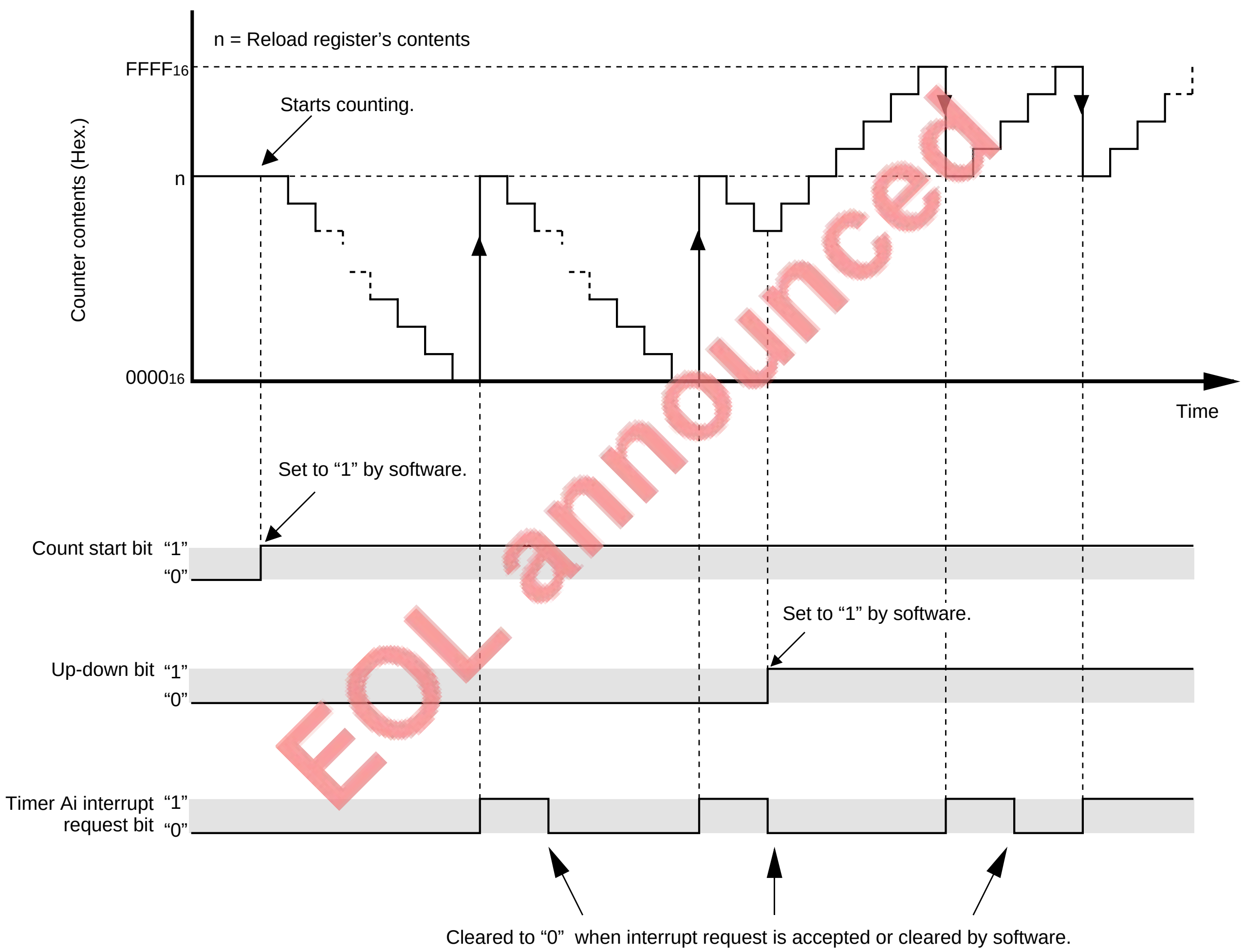


Fig. 5.4.3 Initial setting example for registers relevant to event counter mode (2)

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

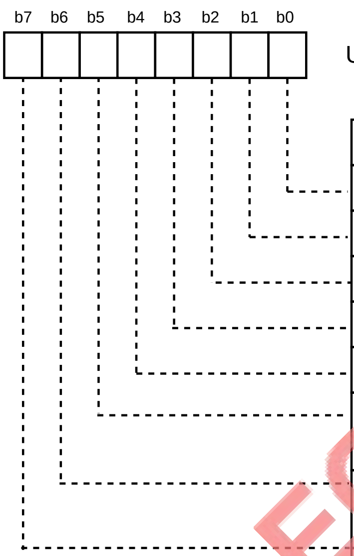
The up-down register (address 44_{16}) or the input signal from the TAi_{OUT} pin is used to switch the up-count from and to the down-count. This switching is performed by the up-down bit when the up-down switching factor select bit (bit 4 at addresses 56_{16} to $5A_{16}$) is "0," and by the input signal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

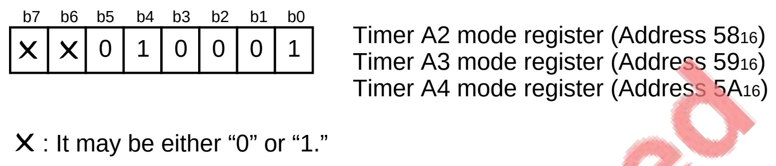


Fig. 5.4.6 Timer A2, A3, and A4 mode registers when two-phase pulse signal processing function is selected

●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.)

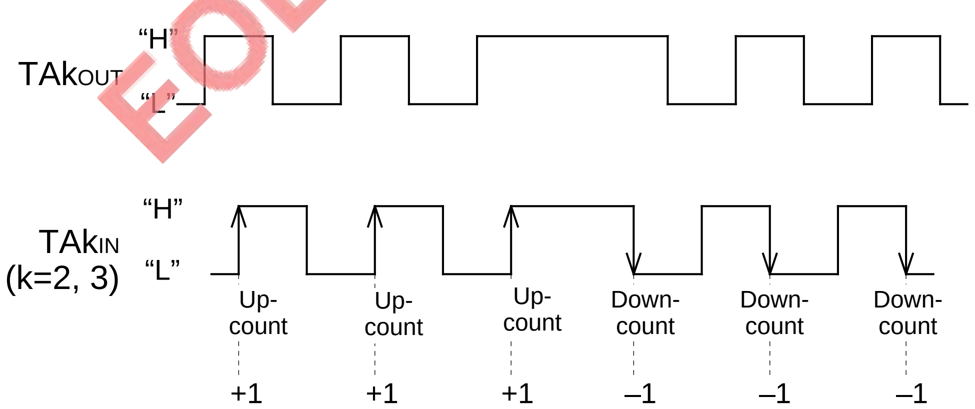


Fig. 5.4.7 Normal processing

●**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.

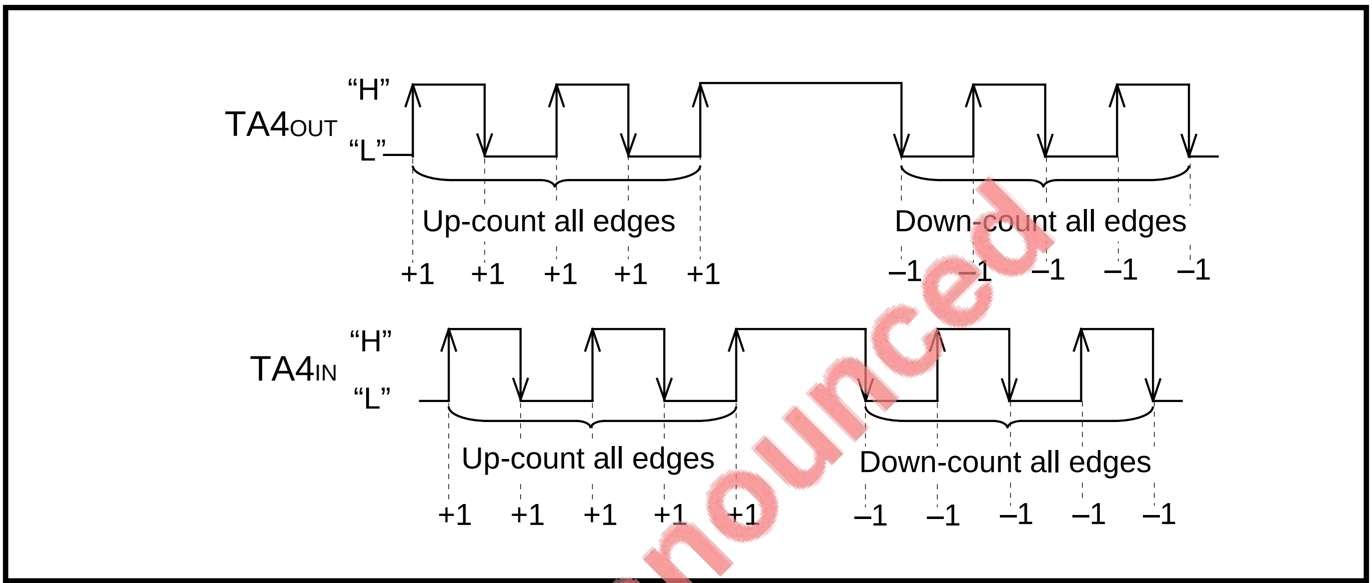


Fig. 5.4.8 Quadruple processing

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.

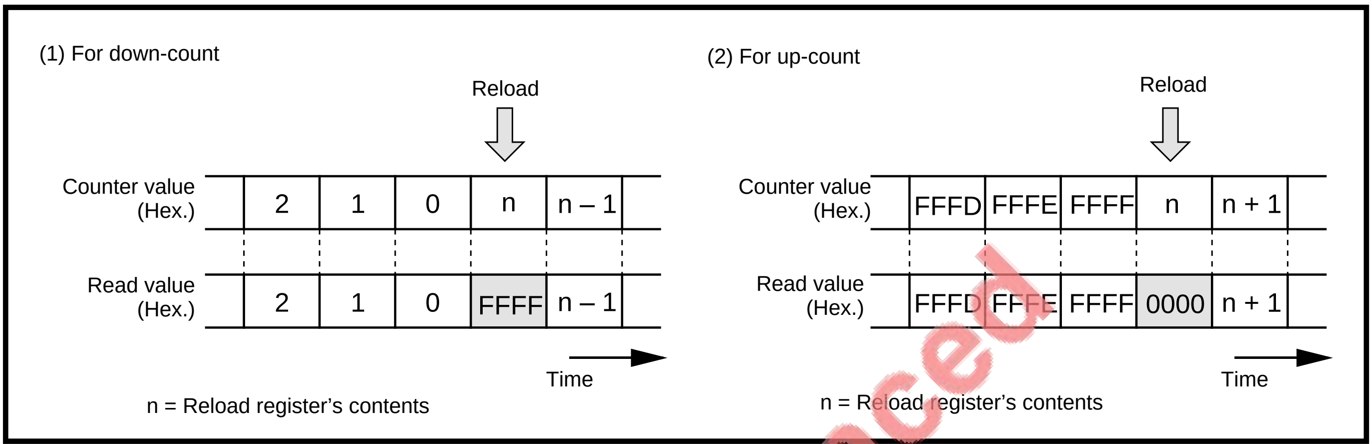


Fig. 5.4.9 Reading timer Ai register

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_{16}, f_{64}, or f_{512}
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.
Output pulse width ("H")	$\frac{n}{f_i}$ [s] n : Timer Ai register setting value
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 : } (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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	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 “0001_{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 : n / f_i.	Undefined	WO

f_i: Frequency of count source (f_2, f_{16}, f_{64}, or f_{512})

Fig. 5.5.1 Structures of timer Ai mode register and timer Ai register in 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."**

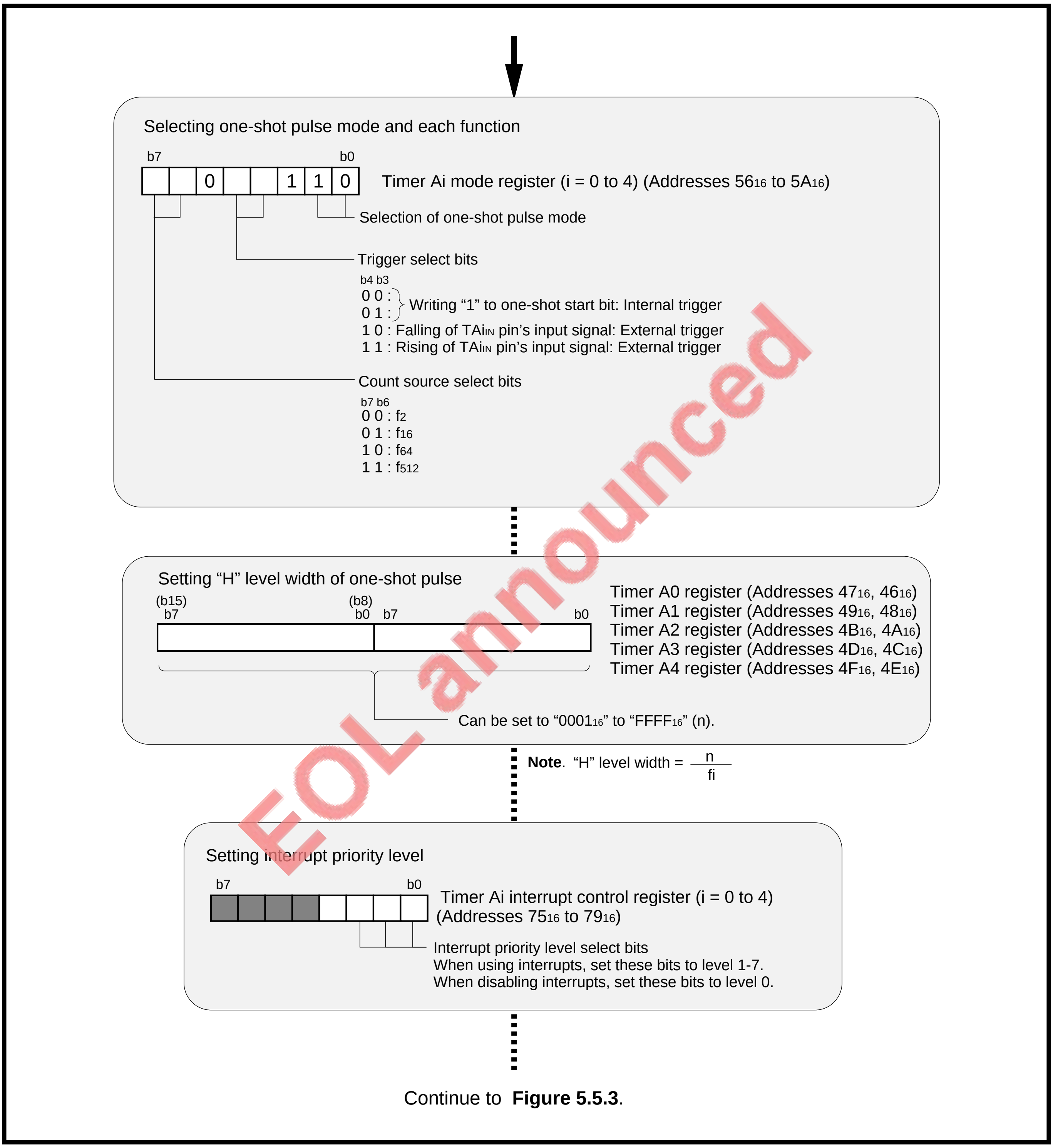


Fig. 5.5.2 Initial setting example for registers relevant to one-shot pulse mode (1)

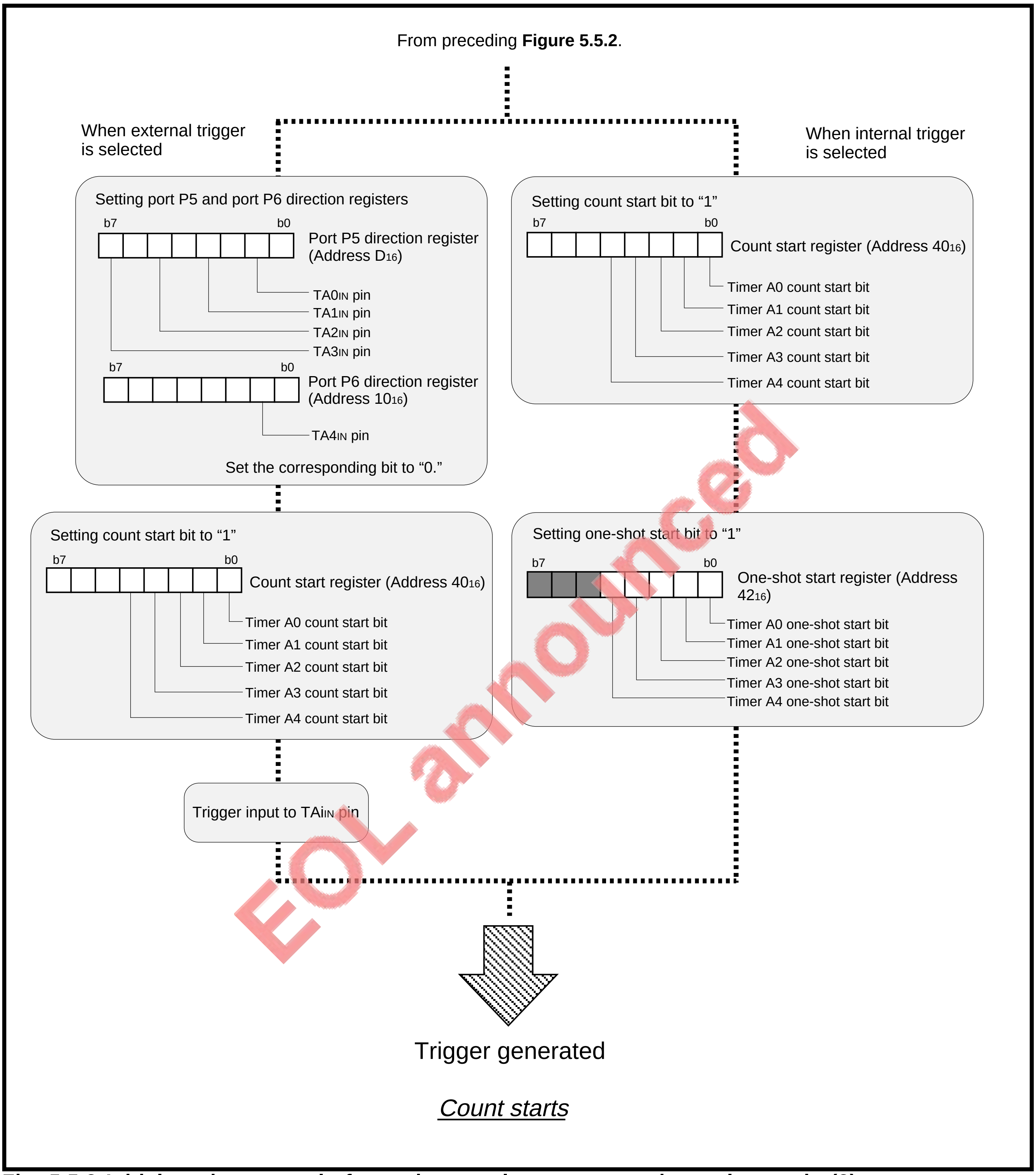


Fig. 5.5.3 Initial setting example for registers relevant to one-shot pulse mode (2)

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		Count source	Count source frequency		
b7	b6		$f(X_{IN})$ = 8 MHz	$f(X_{IN})$ = 16 MHz	$f(X_{IN})$ = 25 MHz
0	0	f_2	4 MHz	8 MHz	12.5 MHz
0	1	f_{16}	500 kHz	1 MHz	1.5625 MHz
1	0	f_{64}	125 kHz	250 kHz	390.625 kHz
1	1	f_{512}	15625 Hz	31250 Hz	48.8281 kHz

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.

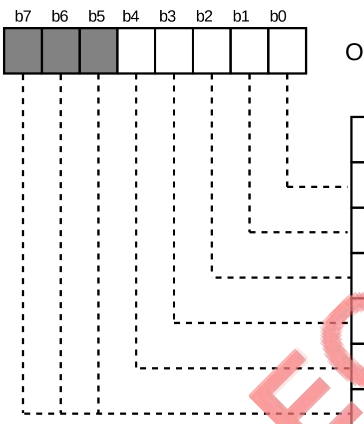
(1) When selecting internal trigger

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 One-shot pulse mode

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.

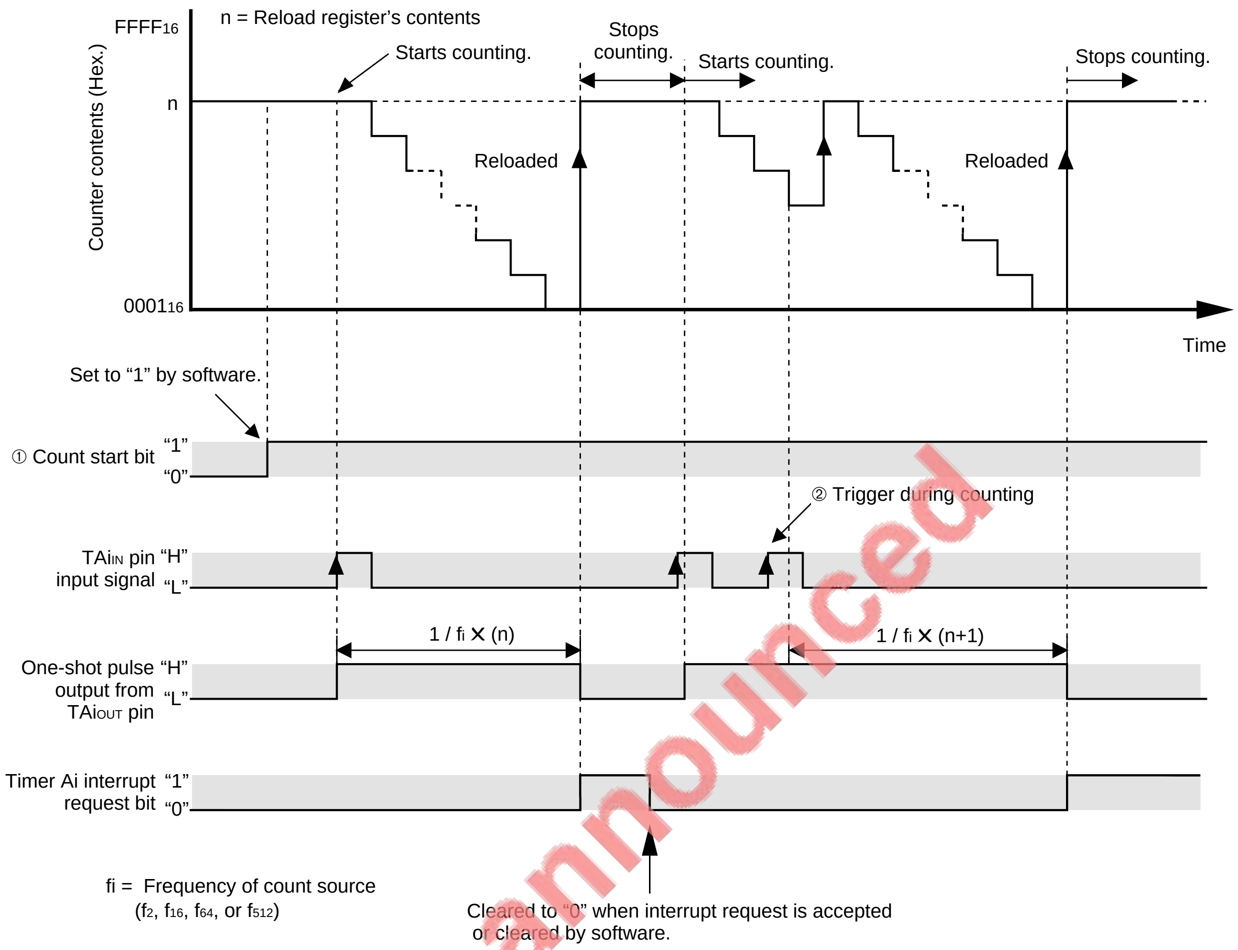
③ When the counter starts counting, the TAi_{OUT} pin outputs "H" level.

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.



fi = Frequency of count source
(f_2, f_{16}, f_{64}, or f_{512})

① When the count start bit = "0" (counting stopped), the TAiOUT pin outputs "L" level.
② When a trigger is generated during counting, the counter counts the count source n + 1 times after a new trigger is generated.

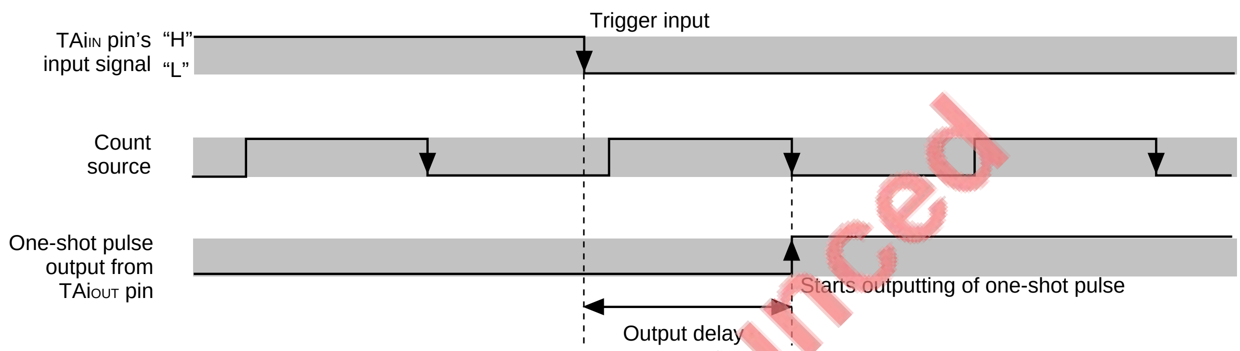
Note: The above applies when an external trigger (rising of TAiIN pin's input signal) is selected.

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 reload register's contents are reloaded into the counter, 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.



Note: The above applies when an external trigger (falling of TAi_{IN} pin's input signal) is selected.

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.

4. Do not set "0000_{16}" to the timer Ai register.

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_{16}, f_{64}, or f_{512}
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.
PMW period/"H" level width	<16-bit pulse width modulator> Period = $\frac{(2^{16}-1)}{f_i}$ [s] "H" level width = $\frac{n}{f_i}$ [s] n: Timer Ai register setting value <8-bit pulse width modulator> Period = $\frac{(m+1)(2^8-1)}{f_i}$ [s] "H" level width = $\frac{n(m+1)}{f_i}$ [s] m: Timer Ai register low-order 8 bits setting value n: Timer Ai register high-order 8 bits setting value
Count start condition	● When a trigger is generated. **(Note)** ● 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
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT 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.)

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

5.6 Pulse width modulation (PWM) mode

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 TAiIN pin's input signal 1 1 : Rising edge of TAiIN 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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	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 TAiOUT pin is expressed as follows: $\frac{n}{f_i}$ PWM pulse's period is expressed as follows: $\frac{n^{16}-1}{f_i}$	Undefined	WO

f_i: Frequency of count source (f_2, f_{16}, f_{64}, or f_{512})

<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 $4F_{16}$, $4E_{16}$)

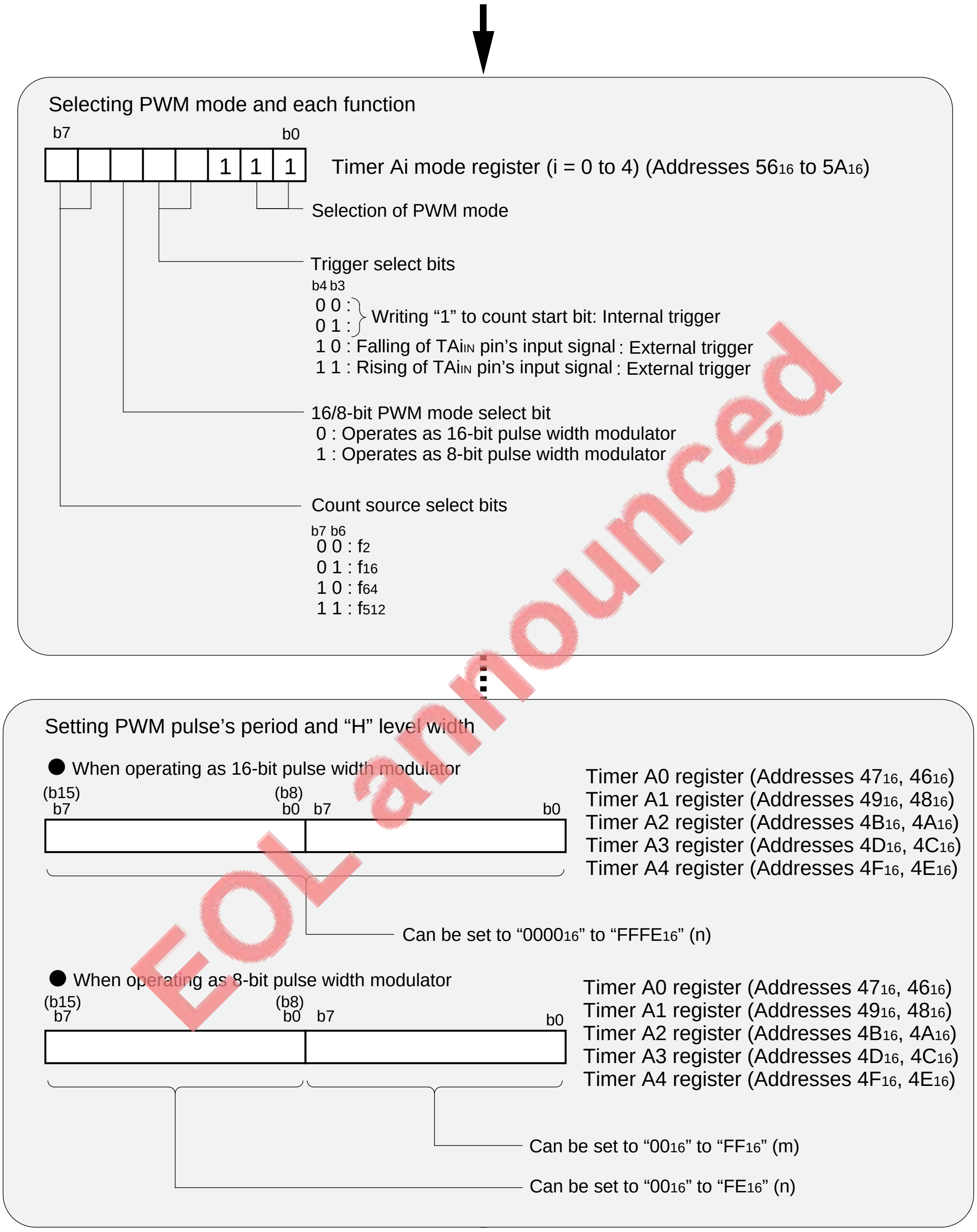
Bit	Functions	At reset	RW
7 to 0	These bits can be set to “00_{16}” to “FF_{16}.” Assuming that the set value = m, PWM pulse's period output from the TAiOUT pin is expressed as follows: $\frac{(m+1)(2^8-1)}{f_i}$	Undefined	WO
15 to 8	These bits can be set to “00_{16}” to “FE_{16}.” Assuming that the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

f_i: Frequency of count source (f_2, f_{16}, f_{64}, or f_{512})

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."**



Note: When operating as 8-bit pulse width modulator

$$\text{Period} = \frac{(m+1)(2^8 - 1)}{fi}$$

$$\text{"H" level width} = \frac{n(m+1)}{fi}$$

fi : Frequency of count source

However, if n = "00_{16}", the pulse width modulator does not operate and the TAi_{OUT} pin outputs "L" level. At this time, no timer Ai request occurs.

Note: When operating as 16-bit pulse width modulator

$$\text{Period} = \frac{(2^{16} - 1)}{fi}$$

$$\text{"H" level width} = \frac{n}{fi}$$

fi : Frequency of count source

However, if n = "0000_{16}", the pulse width modulator does not operate and the TAi_{OUT} pin outputs "L" level. At this time, no timer Ai request occurs.

Continue to **Figure 5.6.3**.

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

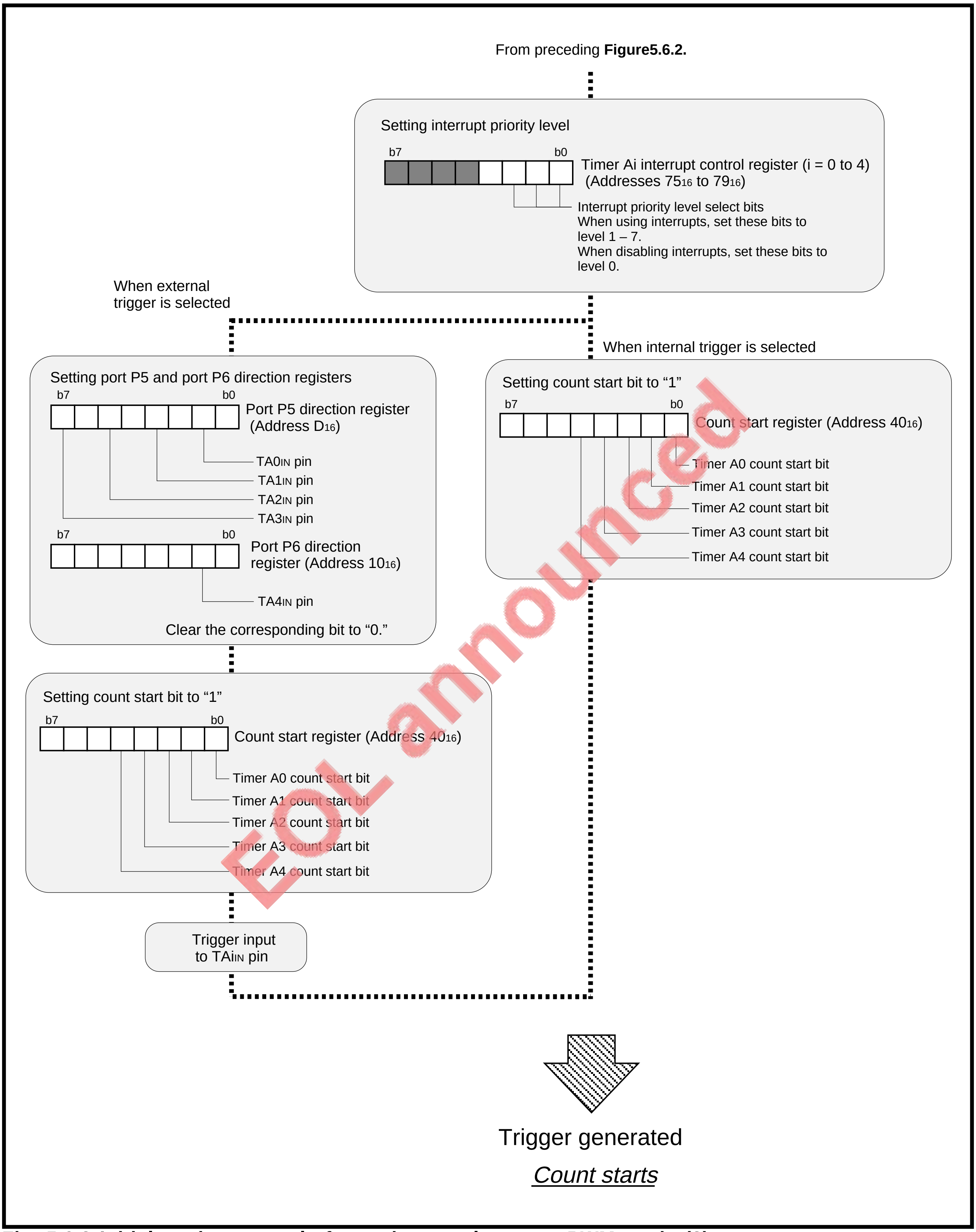


Fig. 5.6.3 Initial setting example for registers relevant to PWM mode (2)

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		Count source	Count source frequency		
b7	b6		$f(X_{IN})$ = 8 MHz	$f(X_{IN})$ = 16 MHz	$f(X_{IN})$ = 25 MHz
0	0	f_2	4 MHz	8 MHz	12.5 MHz
0	1	f_{16}	500 kHz	1 MHz	1.5625 MHz
1	0	f_{64}	125 kHz	250 kHz	390.625 kHz
1	1	f_{512}	15625 Hz	31250 Hz	48.8281 kHz

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.

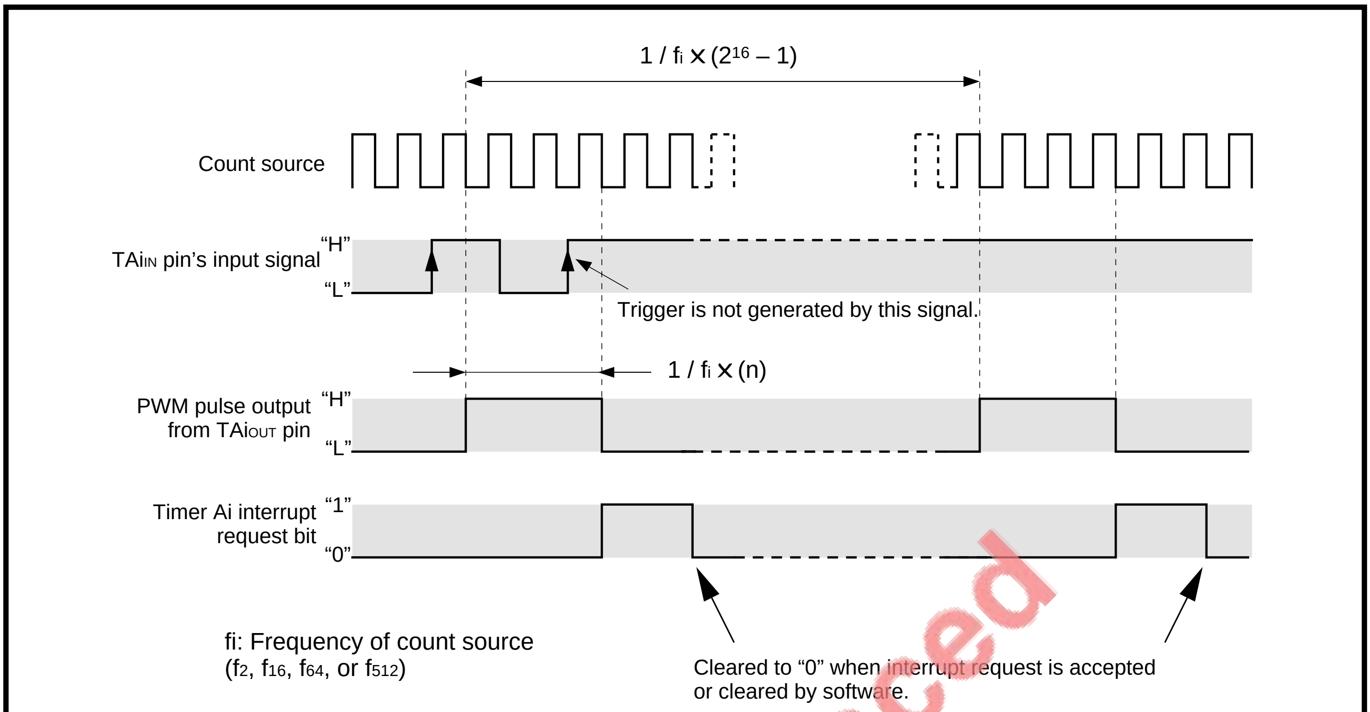
The following explains operation of the pulse width modulator.

[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.

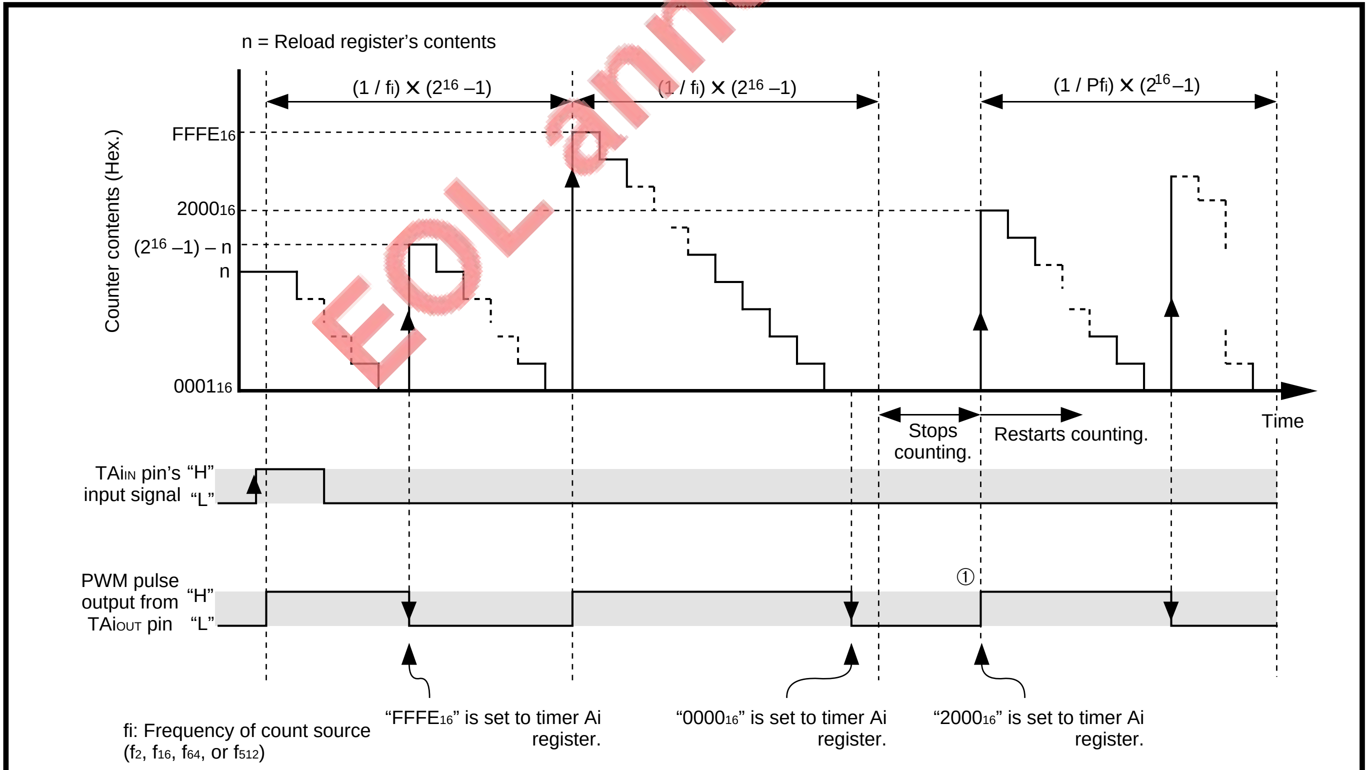
Notes 1: If a value “0000_{16}” is set into the timer Ai register when the counter operates as a 16-bit pulse width modulator, the pulse width modulator does not operate and the output from the TAi_{OUT} pin remains “L” level. The timer Ai interrupt request does not occur. Similarly, if a value “00_{16}”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



Note: The above applies when reload register (n) = "0003$_{16}$" and an external trigger (rising of TAi$_{IN}$ pin's input signal) is selected.

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

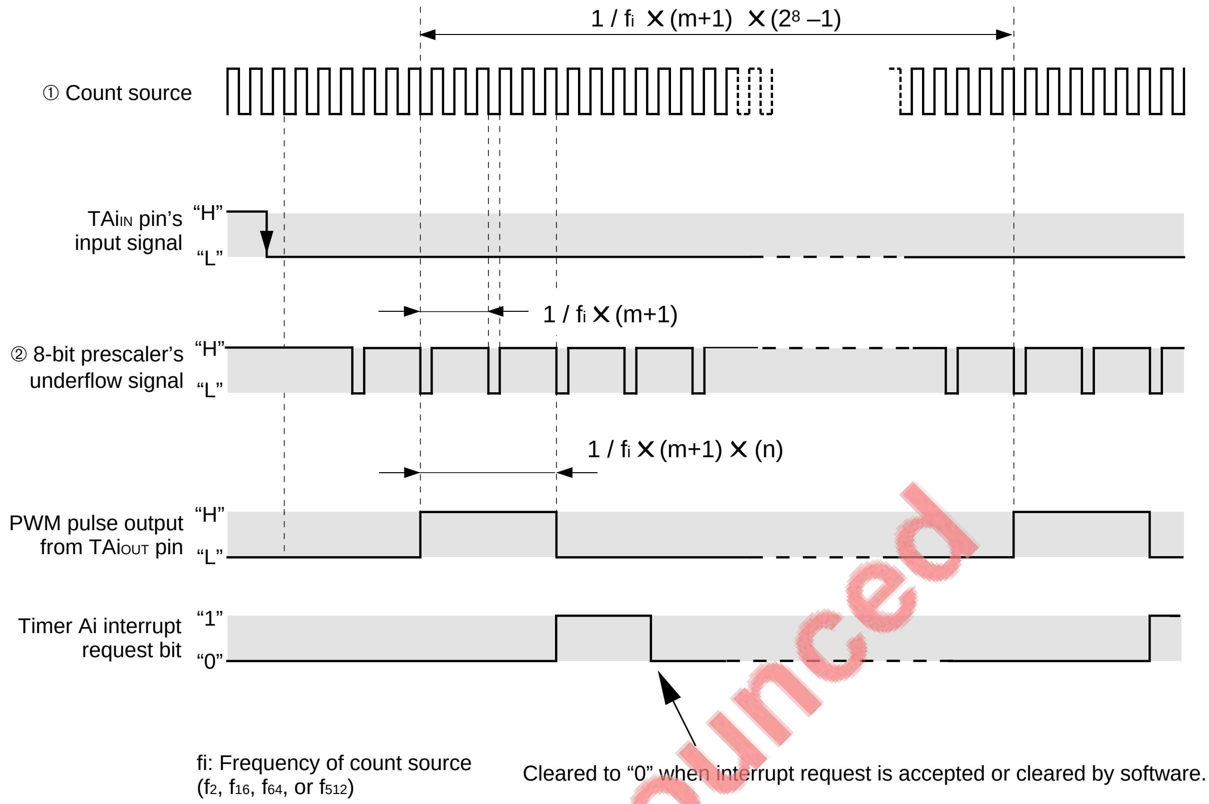


① When an arbitrary value is set to the timer Ai register after setting "0000$_{16}$" to it, the timing at which the PWM pulse goes "H" depends on the timing at which the new value is set.

Note: The above applies when an external trigger (rising of TAi$_{IN}$ pin's input signal) is selected.

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

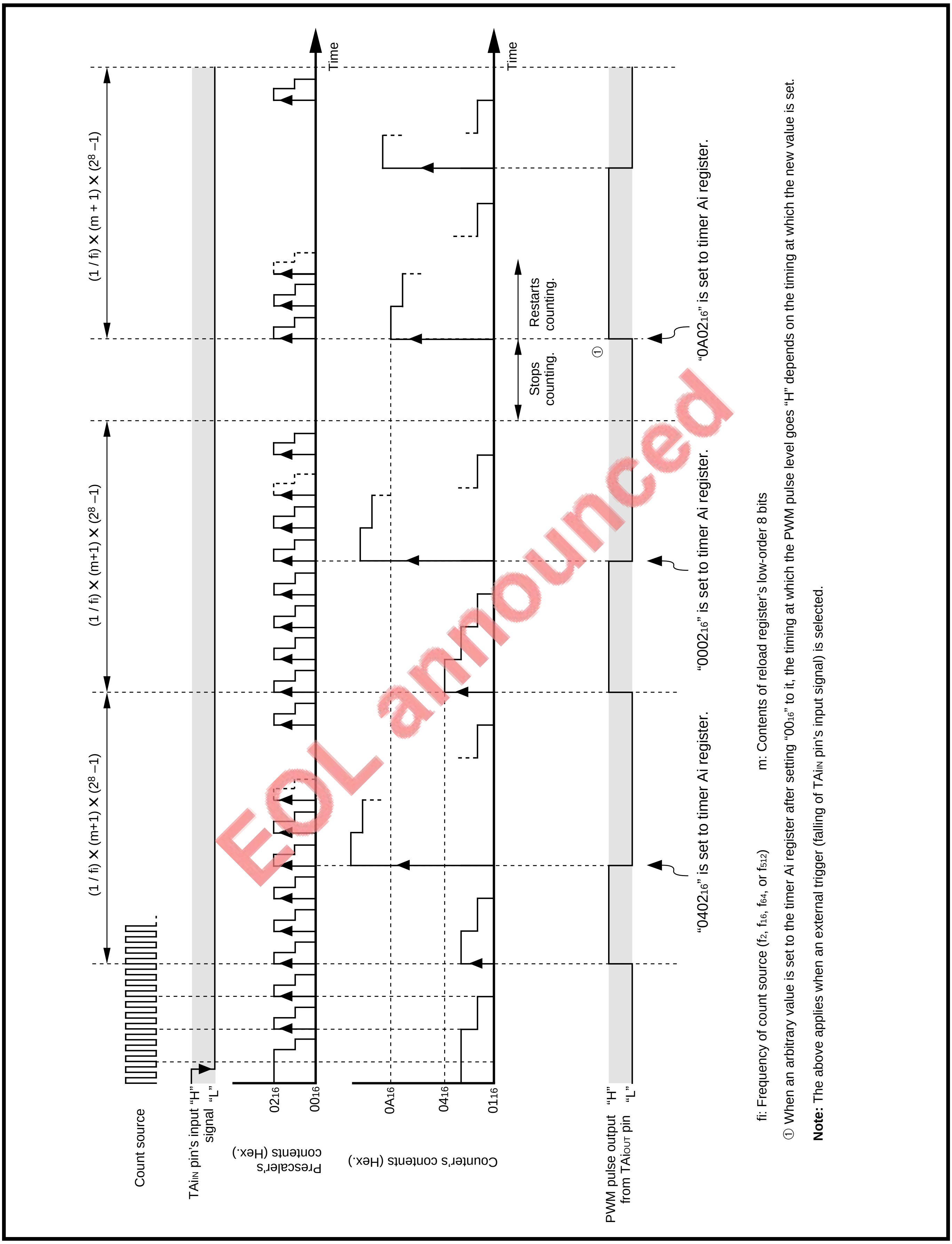


Fig. 5.6.7 Operation example of 8-bit pulse width modulator (when counter value is updated during pulse output)

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

6.1 Overview 6.2 Block description

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 of each other.

7703 Group
Timers B1 and B2s' function of the 7703 Group varies from the 7702 Group's. Refer to "**Chapter 20. 7703 GROUP**."

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.

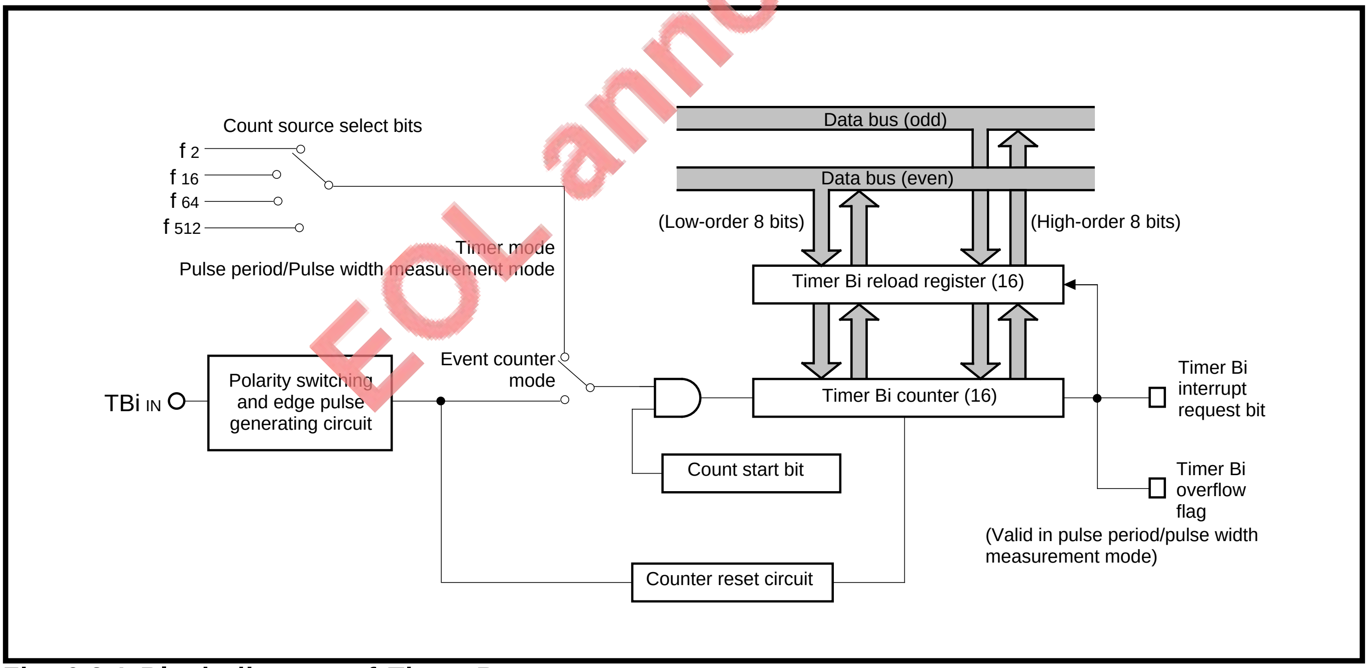


Fig. 6.2.1 Block diagram of Timer B

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 an 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 hold 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 an 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 Block description

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.

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 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 $5D_{16}$)

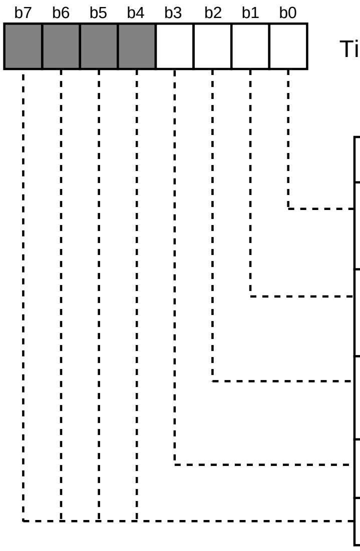
Bit	Bit name	Functions	At reset	RW
0	Operating mode select bits	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/Pulse width measurement mode 1 1 : Not selected	0	RW
1			0	RW
2	These bits have different functions according to the operating mode.		0	RW
3			0	RW
4	Nothing is assigned.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

Note: Use the **SEB** or **CLB** instruction to set each interrupt control register.

Fig. 6.2.4 Structure of timer Bi interrupt control register

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

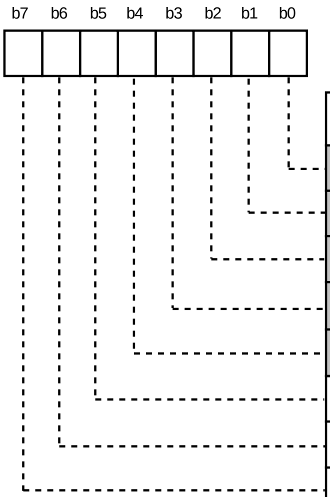
These bits select a timer Bi interrupt's priority level. When using timer Bi interrupts, select priority levels 1 to 7. When the timer Bi interrupt request occurs, its priority level is compared with the processor interrupt priority level (IPL), so that the requested interrupt is enabled only when its 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.

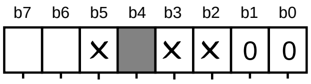
Fig. 6.2.5 Relationship between port P6 direction register and Timer Bi's input pins

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.

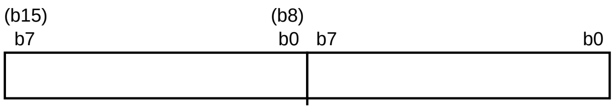
Table 6.3.1 Specifications of timer mode

Item	Specifications
Count source	f_2, f_{16}, f_{64}, or f_{512}
Count operation	•Down-count •When the counter underflows, reload register's contents are reloaded and counting continues.
Divide ratio	$\frac{1}{(n+1)}$ n: Timer Bi register setting value
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	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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	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 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."**

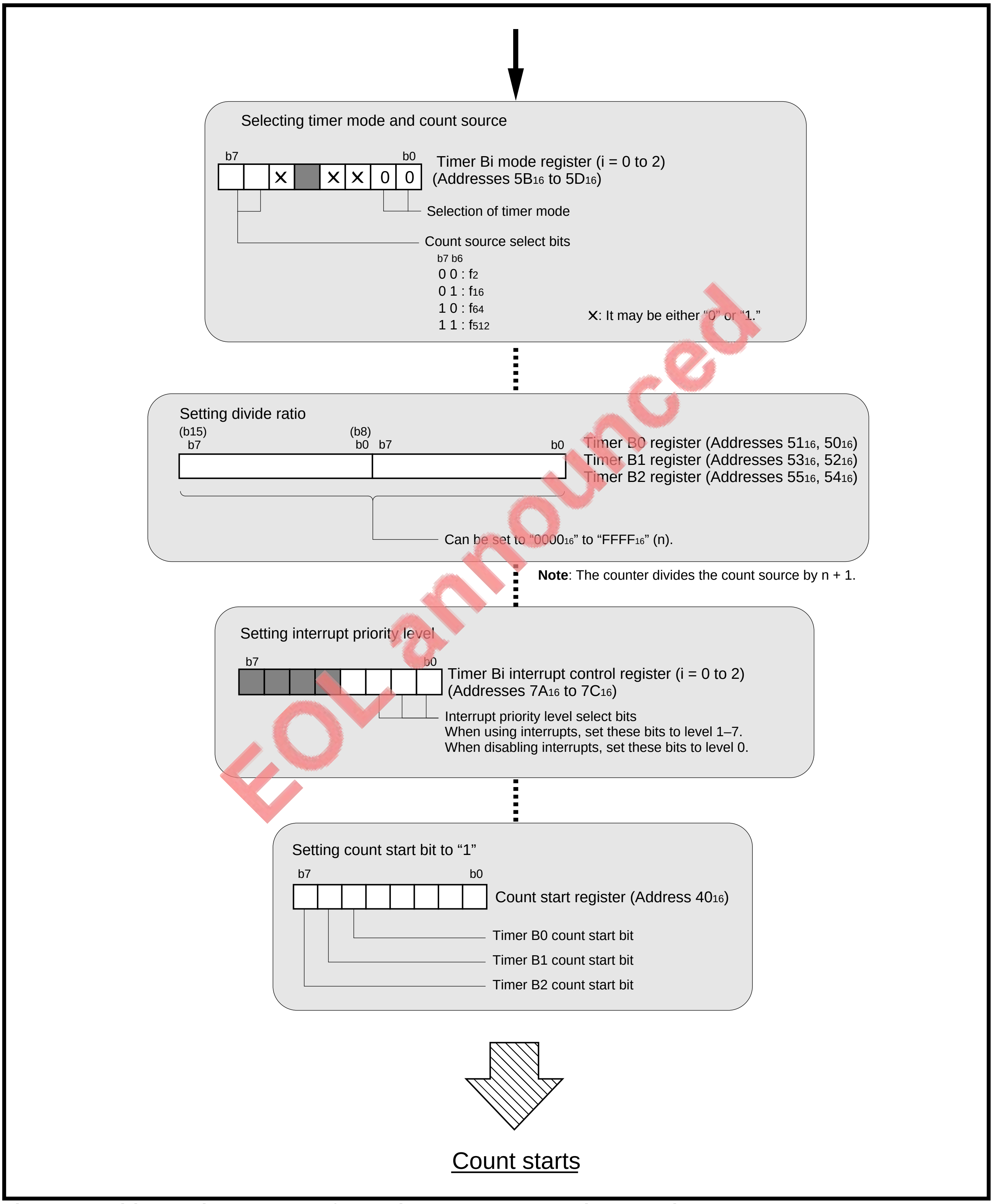


Fig. 6.3.2 Initial setting example for registers relevant to timer mode

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		Count source	Count source frequency		
b7	b6		$f(X_{IN})$ = 8 MHz	$f(X_{IN})$ = 16 MHz	$f(X_{IN})$ = 25 MHz
0	0	f_2	4 MHz	8 MHz	12.5 MHz
0	1	f_{16}	500 kHz	1 MHz	1.5625 MHz
1	0	f_{64}	125 kHz	250 kHz	390.625 kHz
1	1	f_{512}	15625 Hz	31250 Hz	48.8281 kHz

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.

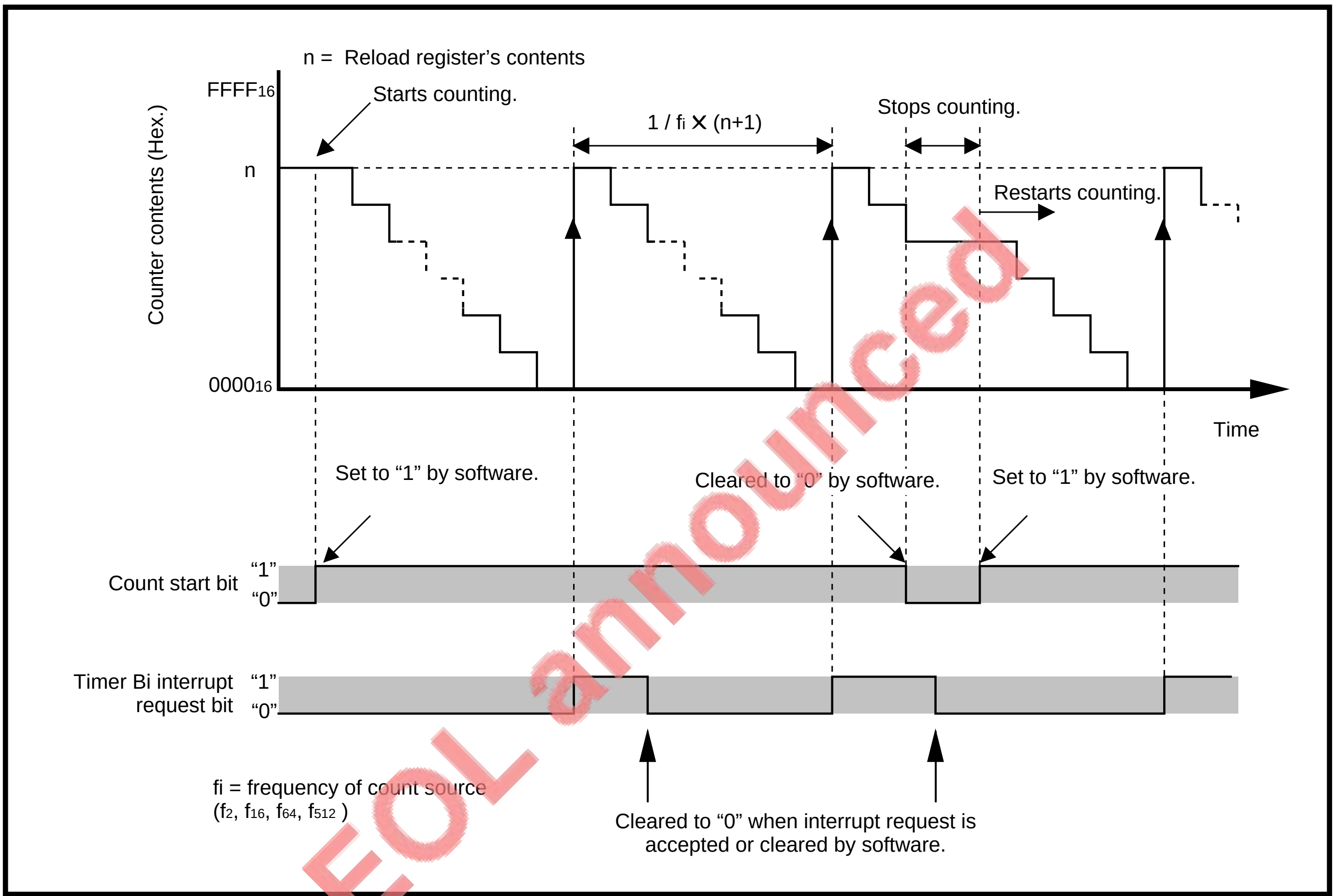


Fig. 6.3.3 Example of operation in timer mode

[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 "FFFF16" 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.

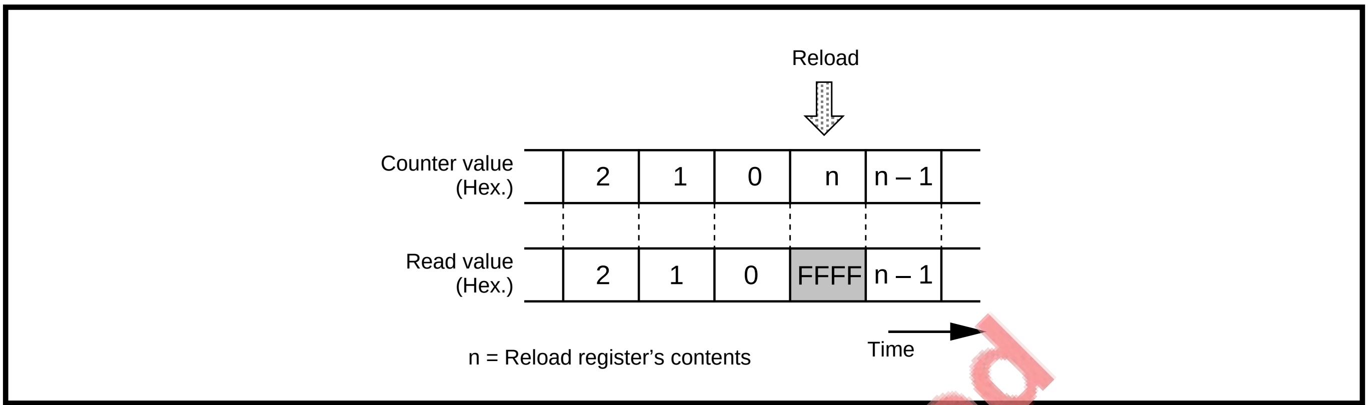


Fig. 6.3.4 Reading timer Bi register

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 effective 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.
Divide ratio	$\frac{1}{(n+1)}$ n: Timer Bi register setting value
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.)

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

(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

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."**

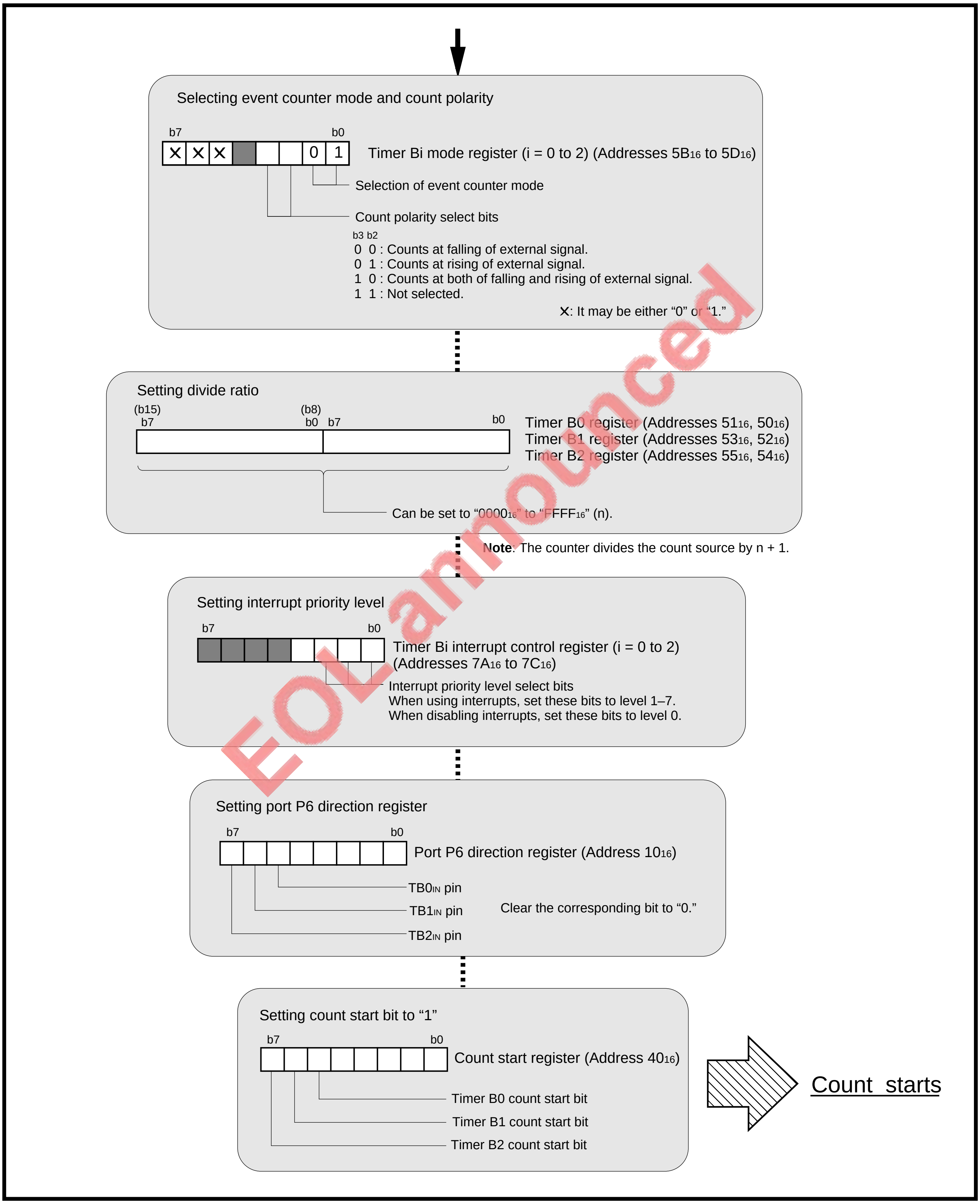


Fig. 6.4.2 Initial setting example for registers relevant to event counter mode

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.

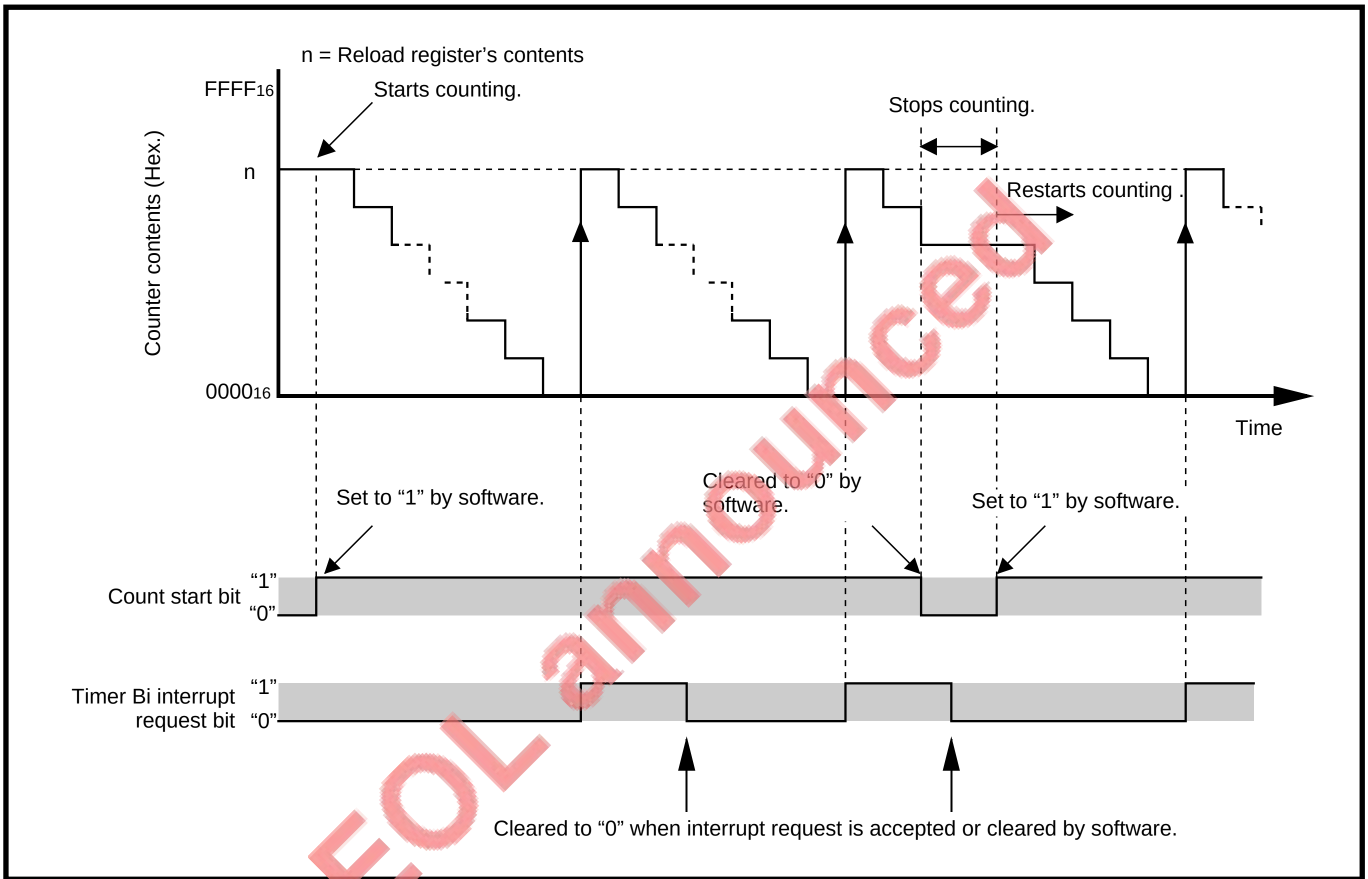


Fig. 6.4.3 Example of operation in event counter mode

[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 "FFFF16" 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.

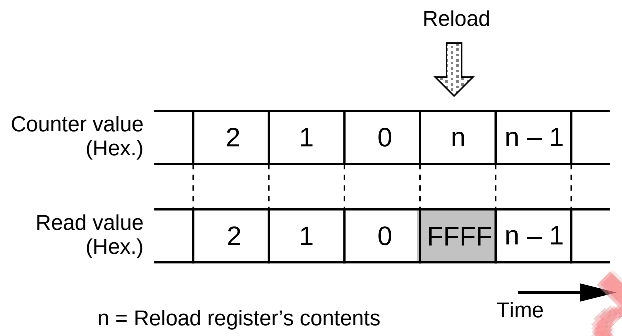


Fig. 6.4.4 Reading timer Bi register

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_{16}, f_{64}, or f_{512}
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 (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.

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

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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	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."**

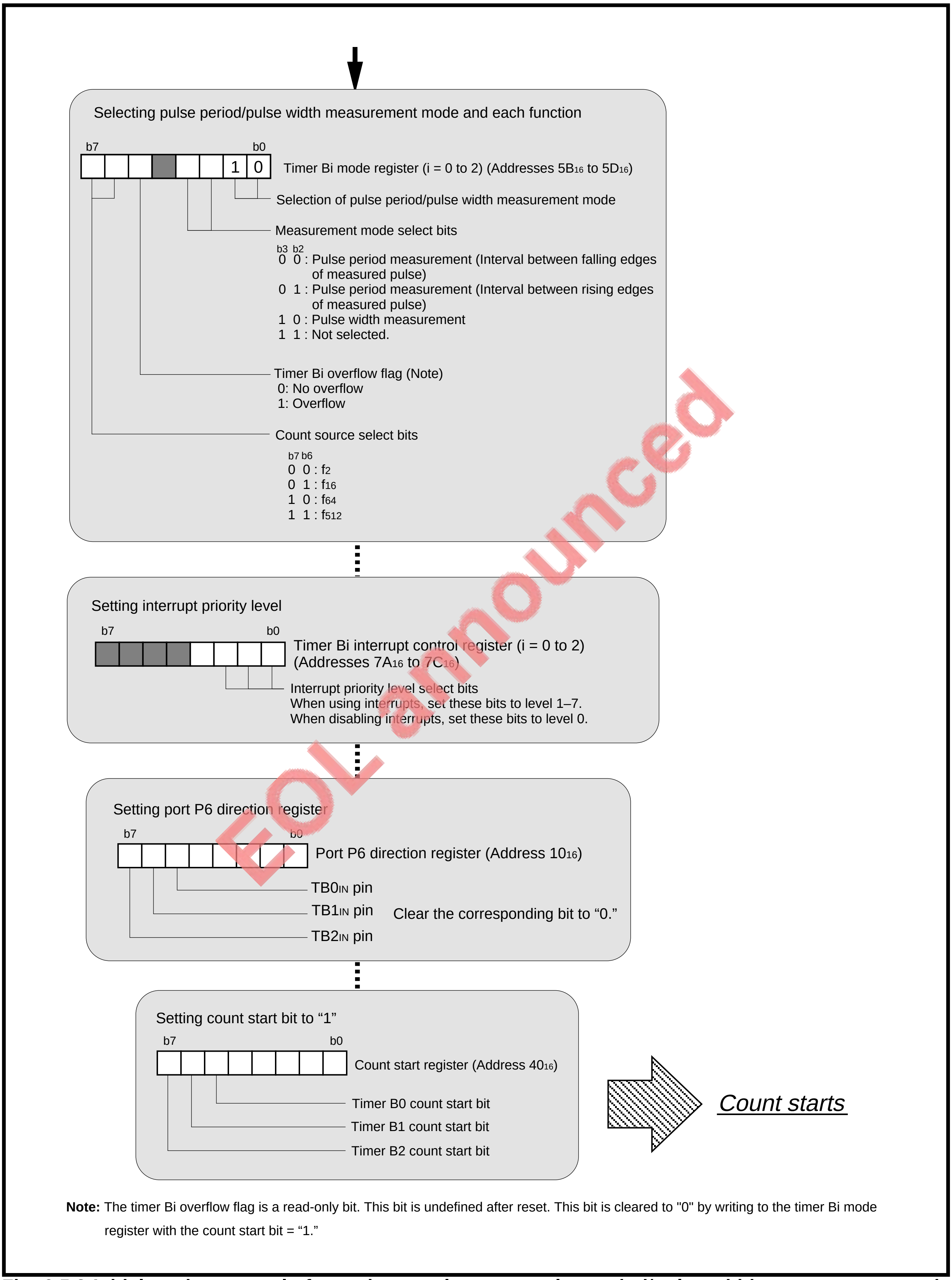


Fig. 6.5.2 Initial setting example for registers relevant to pulse period/pulse width measurement mode

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		Count source	Count source frequency		
b7	b6		$f(X_{IN})$ = 8 MHz	$f(X_{IN})$ = 16 MHz	$f(X_{IN})$ = 25 MHz
0	0	f_2	4 MHz	8 MHz	12.5 MHz
0	1	f_{16}	500 kHz	1 MHz	1.5625 MHz
1	0	f_{64}	125 kHz	250 kHz	390.625 kHz
1	1	f_{512}	15625 Hz	31250 Hz	48.8281 kHz

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.

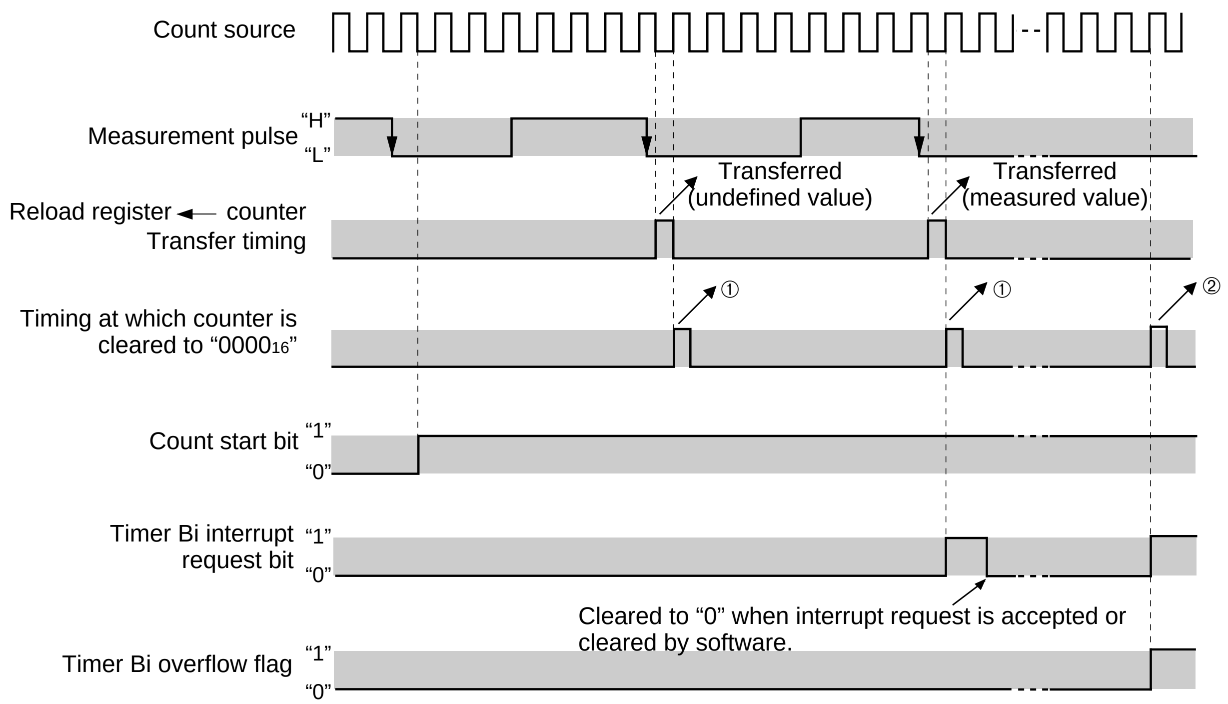
Table 6.5.3 Relationship between measurement mode select bits and pulse period/pulse width measurements

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.

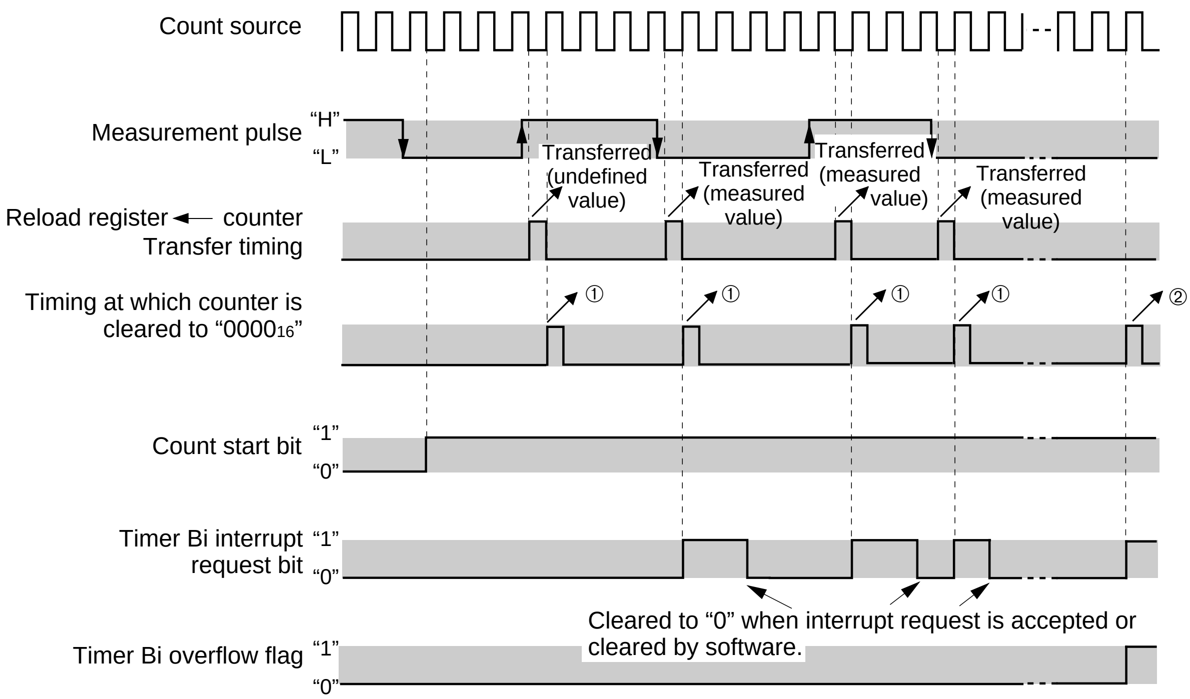


① Counter is initialized by completion of measurement.
② Counter overflow.

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



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

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

SERIAL I/O

7.1 Overview

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.

7703 Group

UART1's function of the 7703 Group varies from the 7702 Group's. Refer to "**Chapter 20. 7703 GROUP.**"

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.

Figure 7.1.1 shows the transfer data formats in each operating mode.

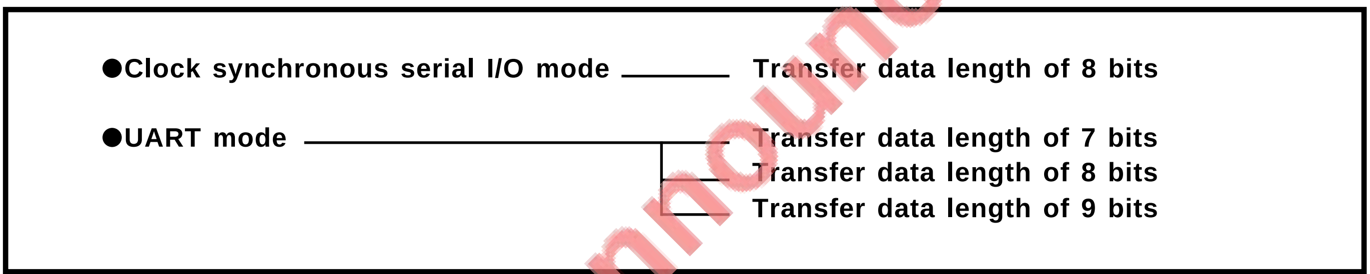


Fig. 7.1.1 Transfer data formats in each operating mode

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.

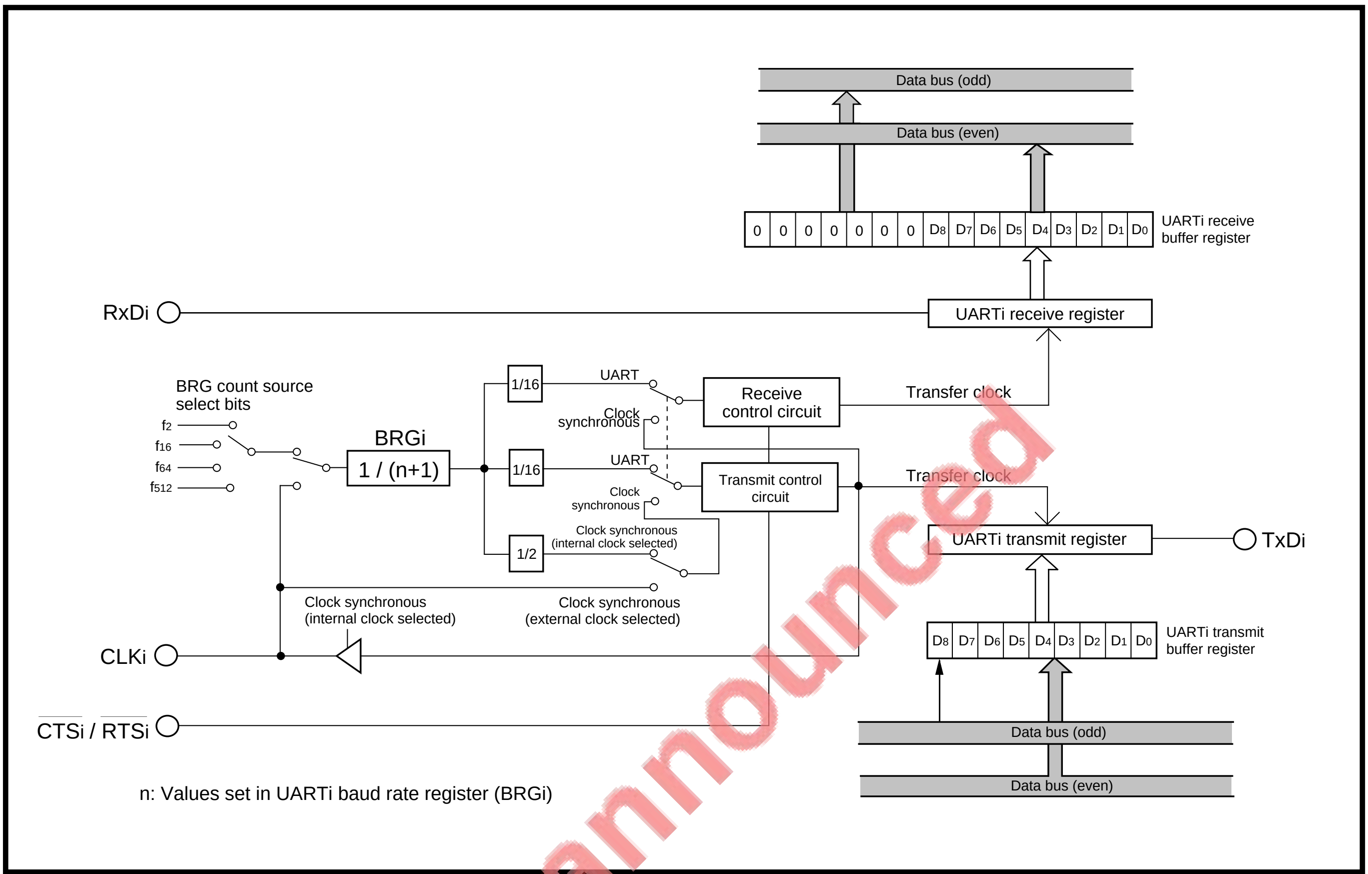
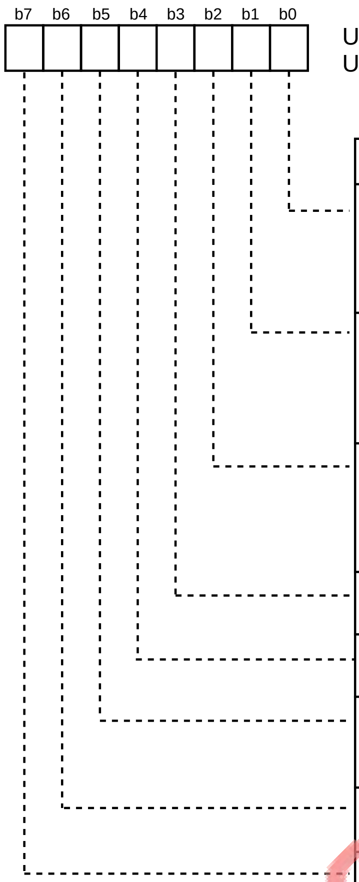


Fig. 7.2.1 Block diagram of Serial I/O

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.

BRGi: UARTi baud rate register (Refer to section "**7.2.6 UARTi baud rate register (BRGi)**.")

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."**

b7 b6 b5 b4 b3 b2 b1 b0

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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	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
7 to 4	Nothing is assigned.		Undefined	—

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.

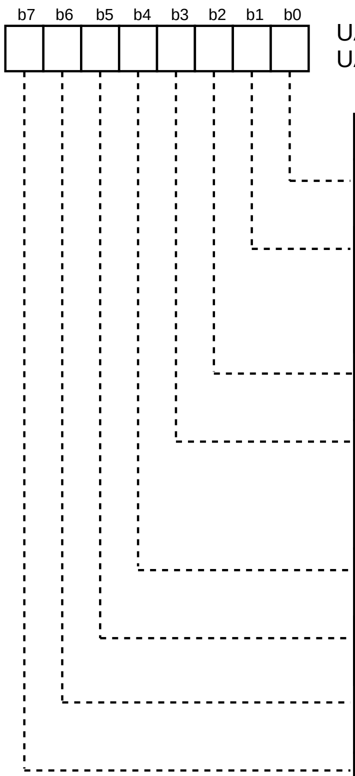
The output level of this pin becomes "L" when the receive enable bit is set to "1." It becomes "H" when reception starts and it becomes "L" when reception is completed.

(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: Bits 7 to 4 are cleared to "0" when clearing the receive enable bit to "0" or when reading the low-order byte of the UARTi receive buffer register (addresses 36_{16}, $3E_{16}$) out.
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

7.2 Block description

(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.

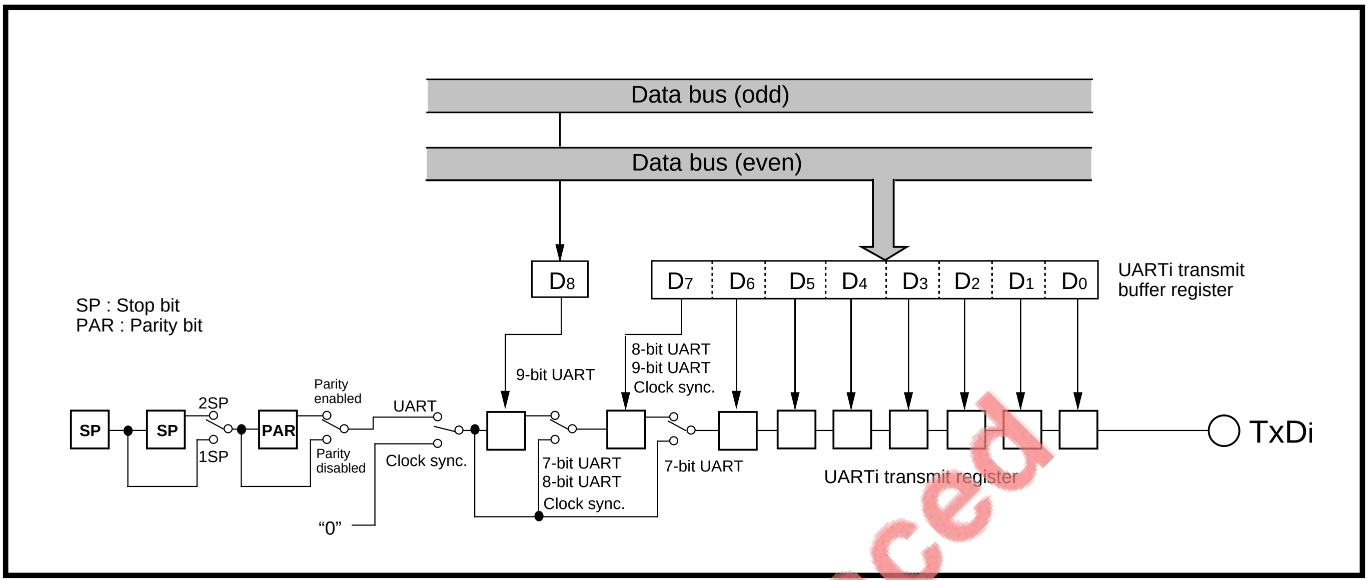
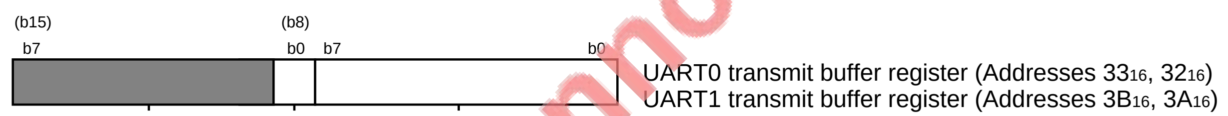


Fig. 7.2.5 Block diagram of transmit section



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 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.

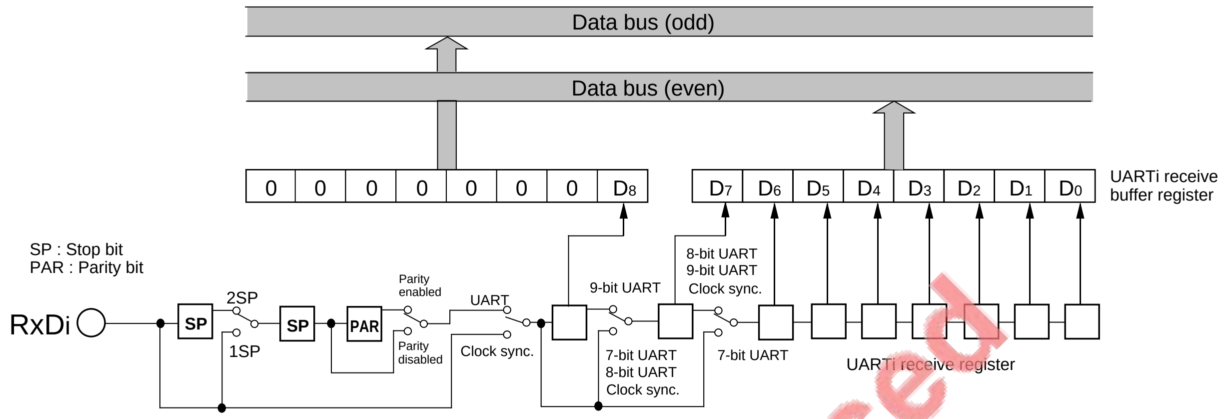
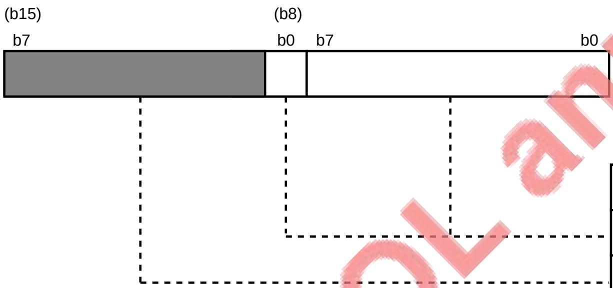


Fig. 7.2.7 Block diagram of receive section



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

7.2 Block description

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).

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.

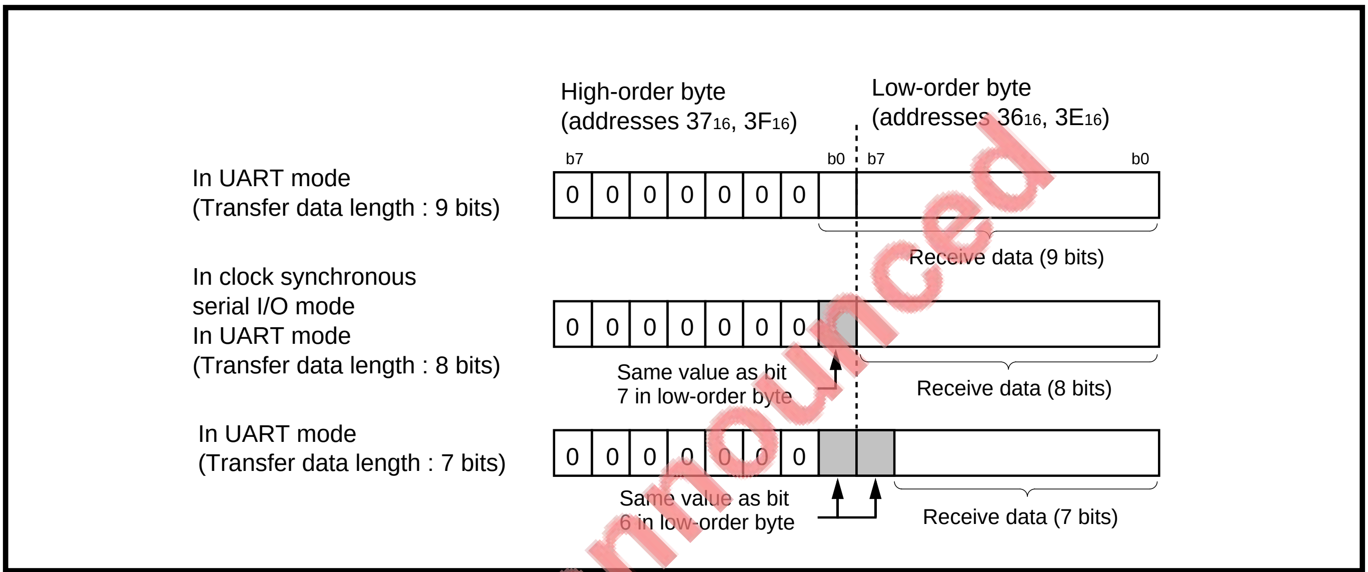


Fig. 7.2.9 Contents of UARTi receive buffer register when reception is completed

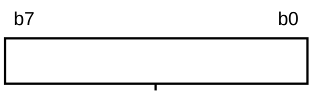
7.2.6 UARTi baud rate register (BRGi)

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.



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

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

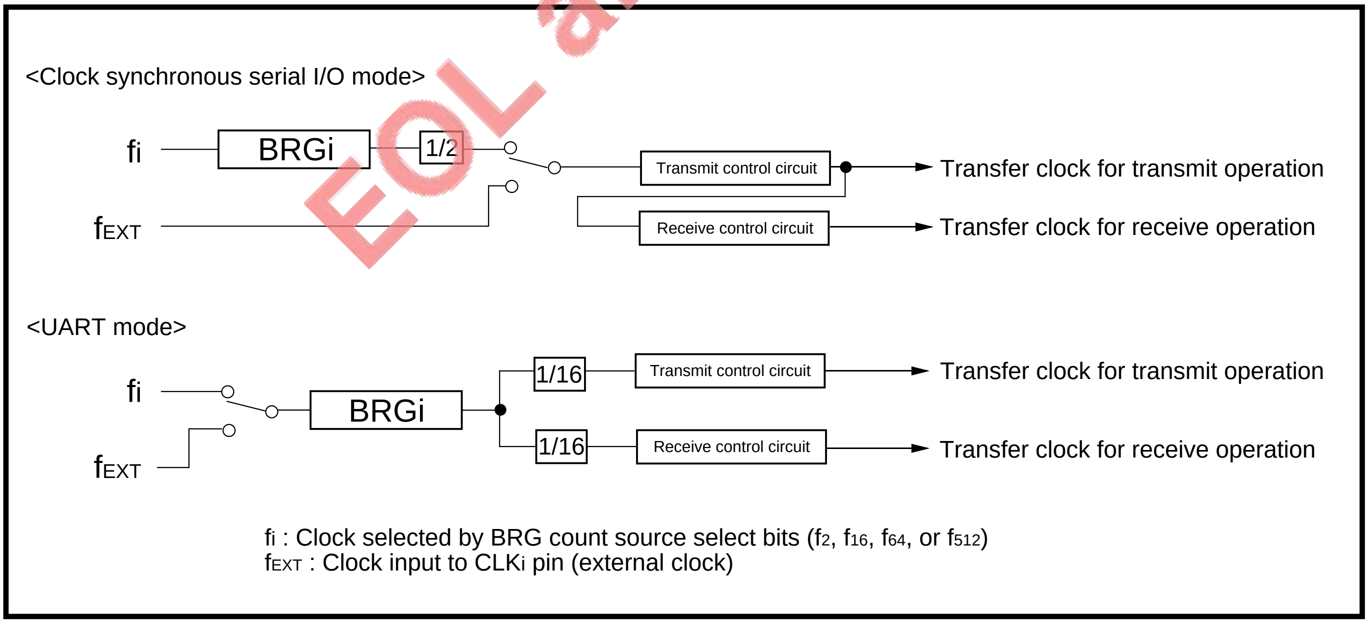
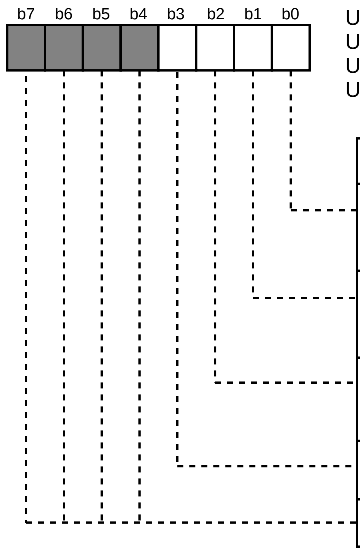


Fig. 7.2.11 Block diagram of transfer clock generating section

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 allocated.		Undefined	–

Note: Use the **SEB** or **CLB** instruction to set each interrupt control registers.

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).

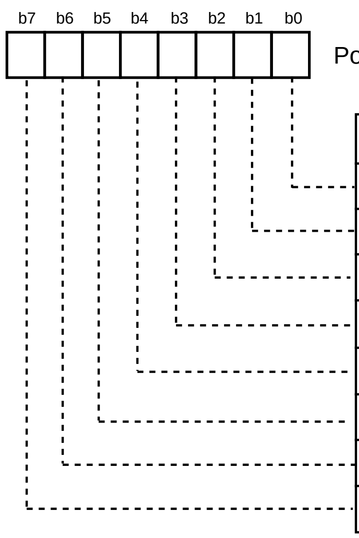
(2) Interrupt request bit (bit 3)

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
Transfer rate	When selecting internal clock	Clock which is BRGi output's divided by 2.
	When selecting external clock	Maximum 5 Mbps ($f(X_{IN})$ = 25 MHz) Maximum 4 Mbps ($f(X_{IN})$ = 16 MHz) Maximum 2 Mbps ($f(X_{IN})$ = 8 MHz)
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" to the BRGi (addresses 31_{16}, 39_{16}).

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

n: Setting value to BRGi
f_i: Frequency of BRGi's count source (f_2, f_{16}, f_{64}, f_{512})

•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 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 (external clock selected).
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.

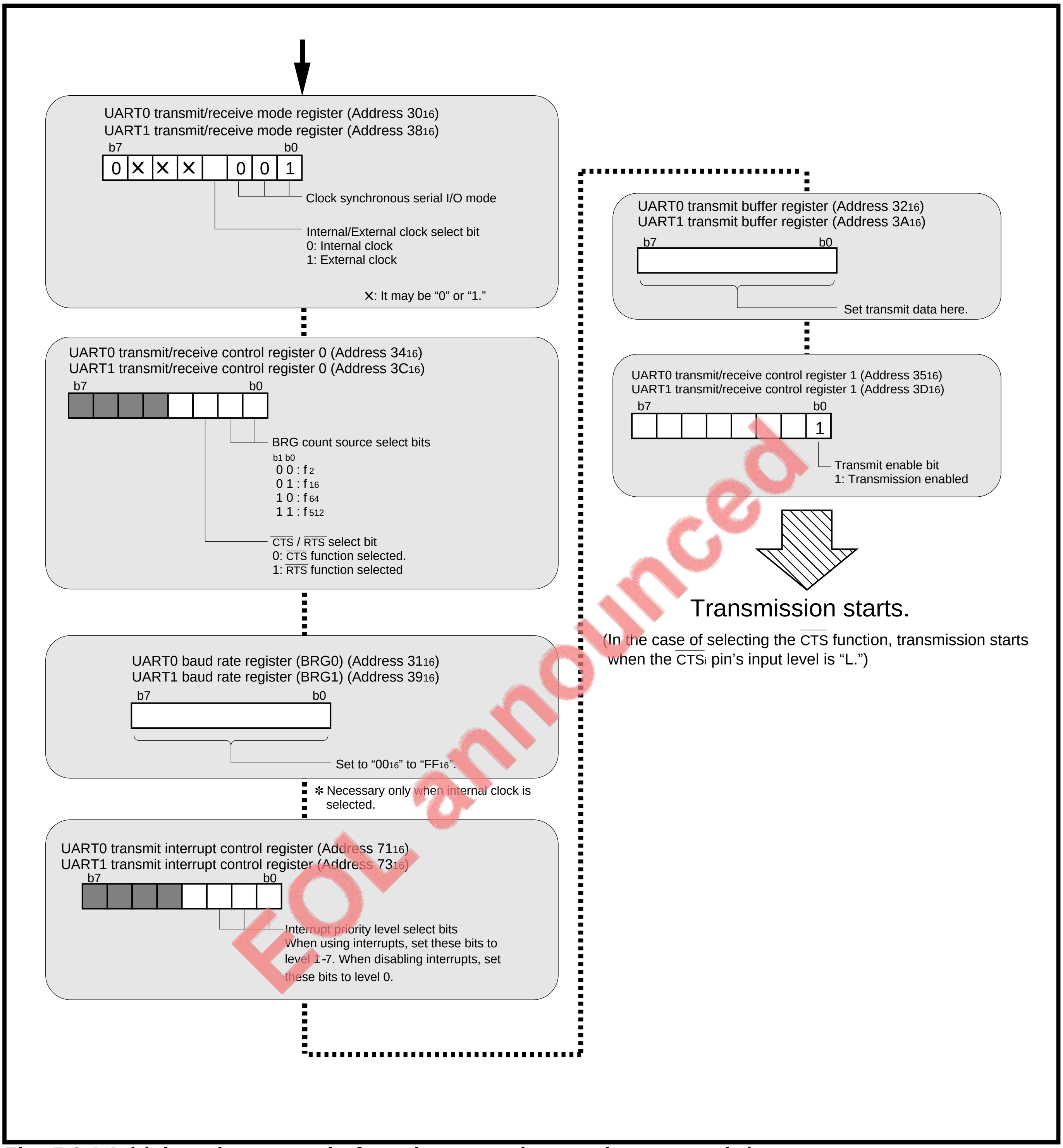


Fig. 7.3.1 Initial setting example for relevant registers when transmitting

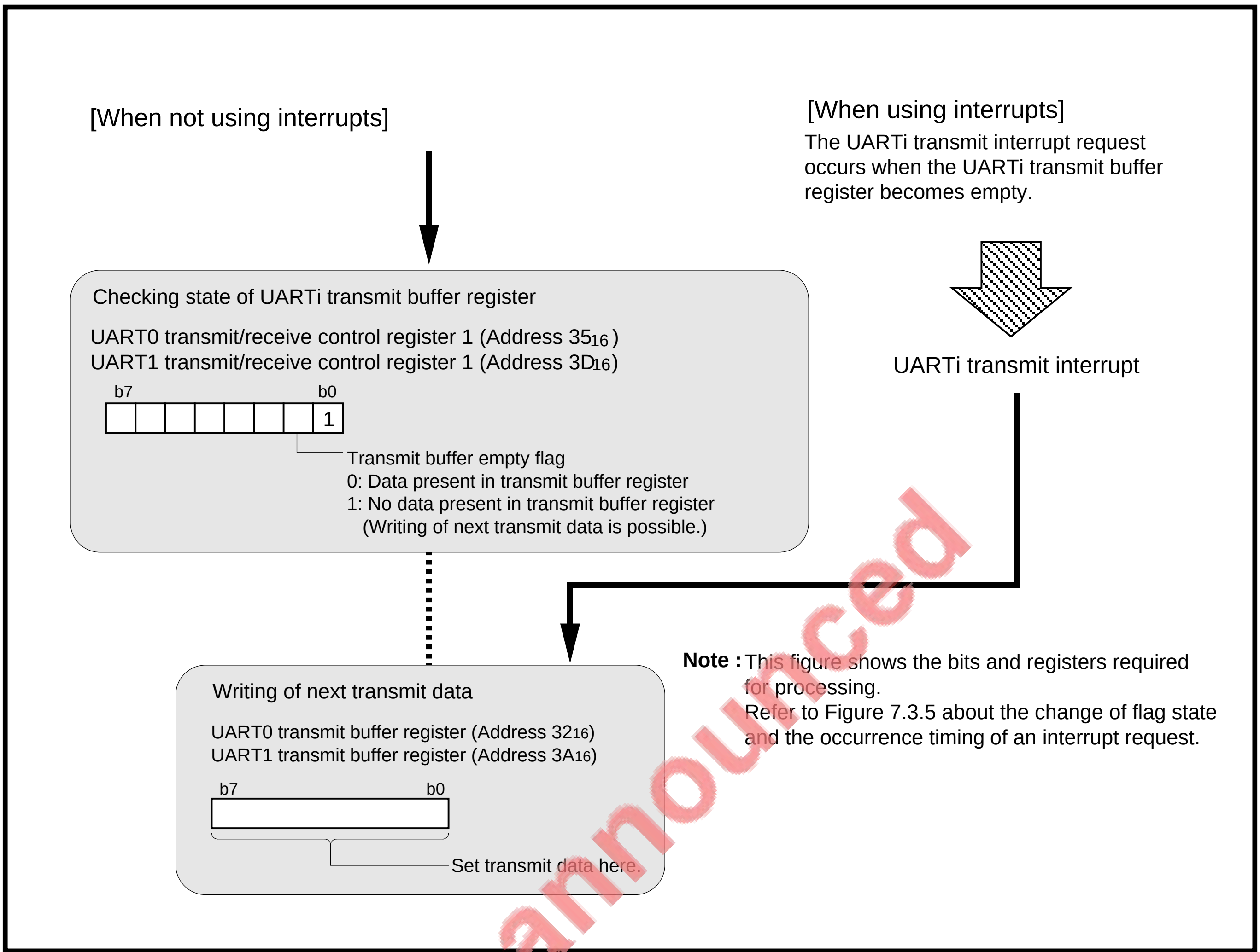


Fig. 7.3.2 Writing data after start of transmission

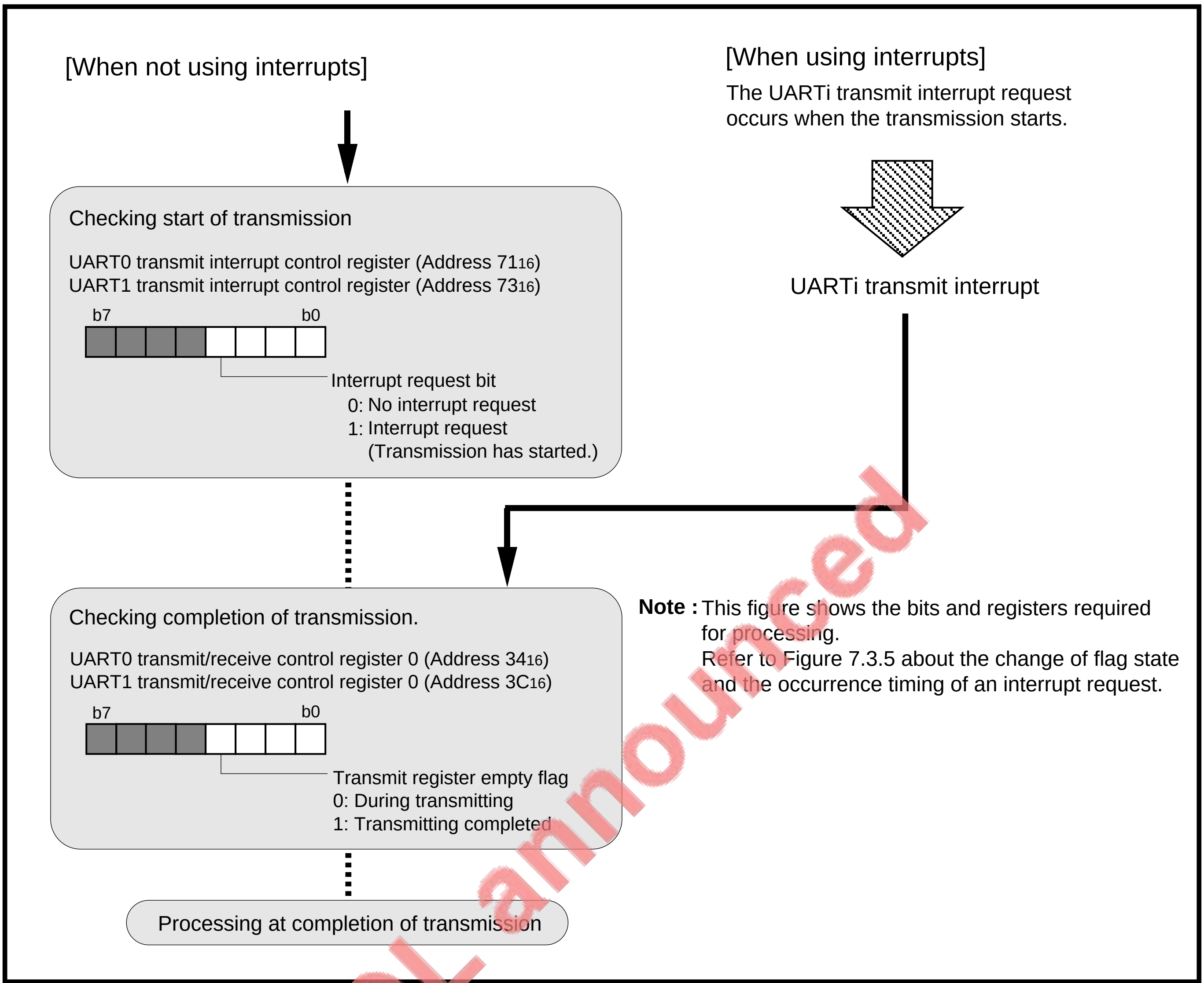


Fig. 7.3.3 Detection of transmission's completion

7.3.3 Transmit operation

When the transmit conditions described in page 7-19 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).

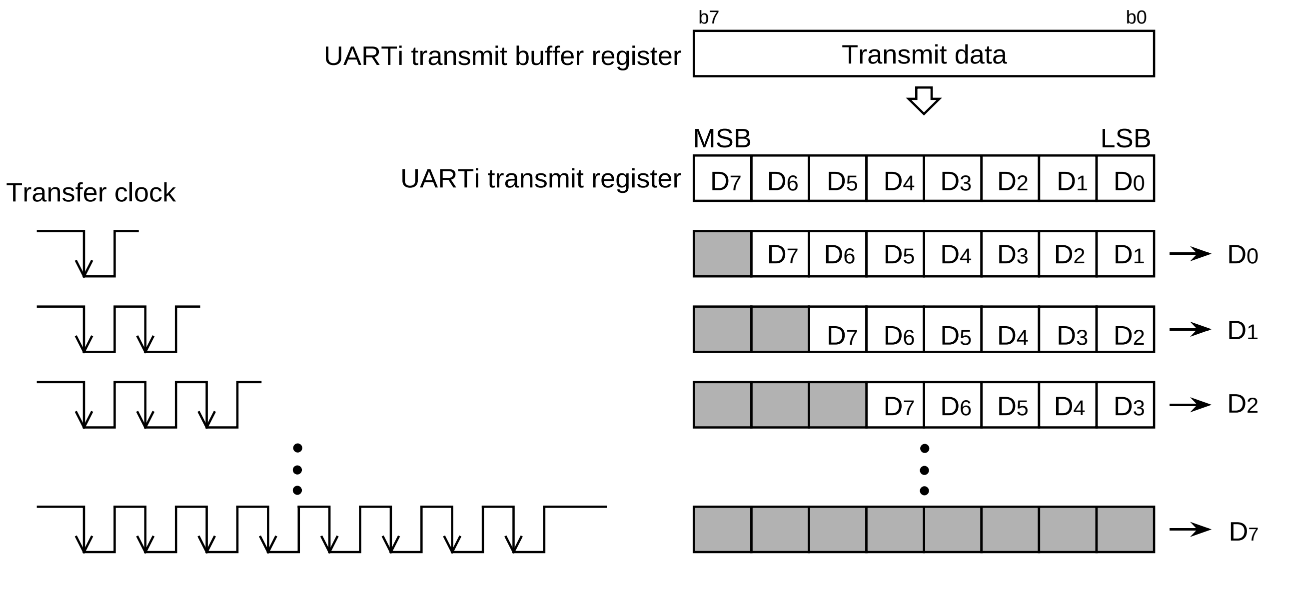


Fig. 7.3.4 Transmit operation

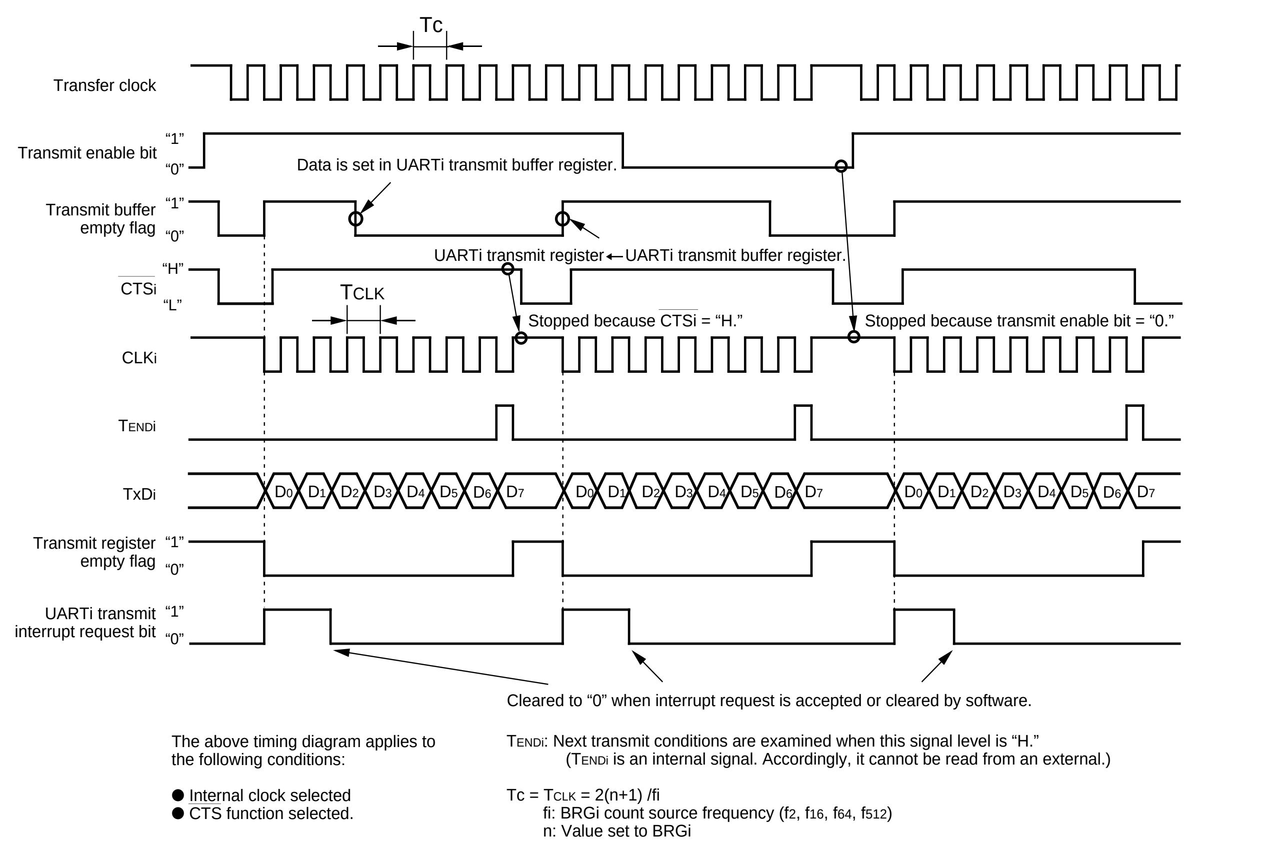


Fig. 7.3.5 Example of transmit timing (when selecting internal clock)

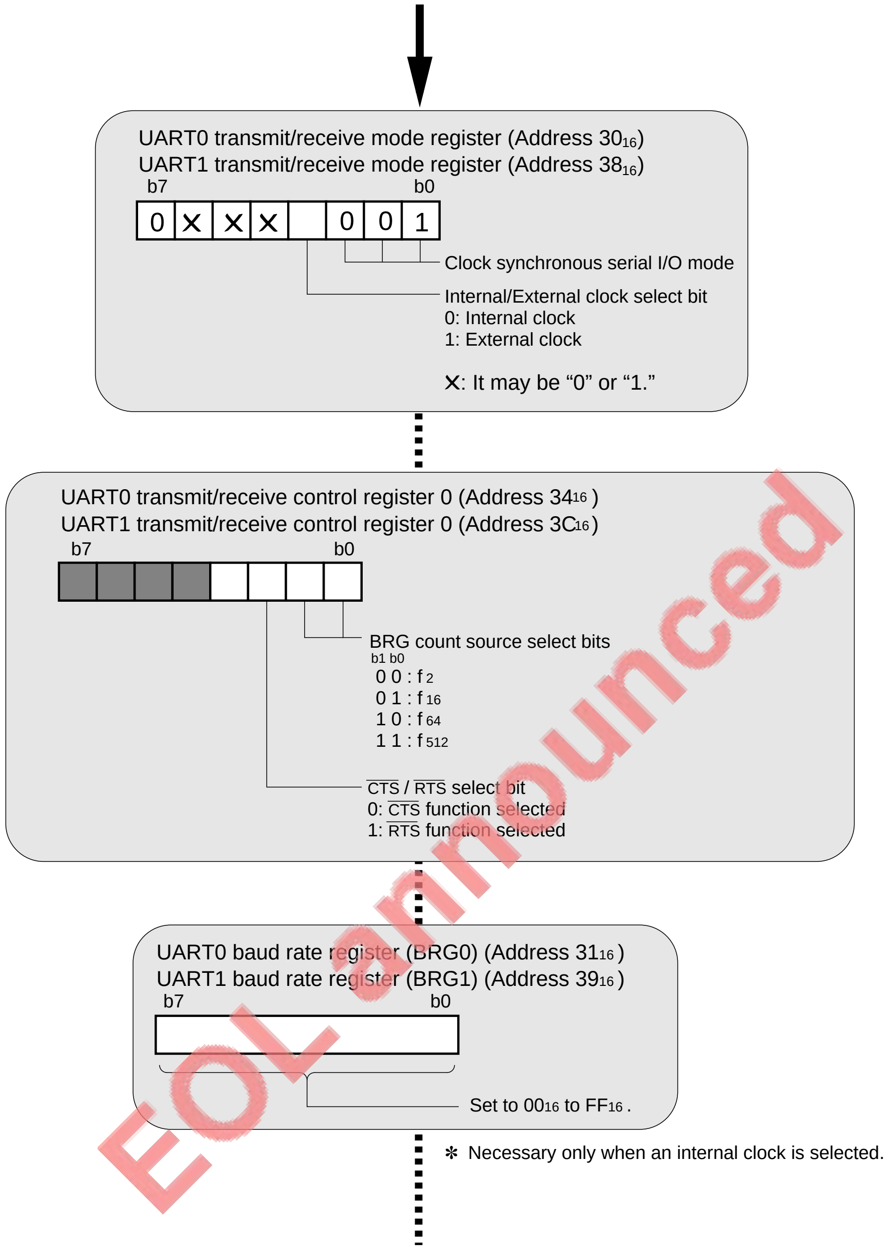
7.3.4 Method of reception

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)

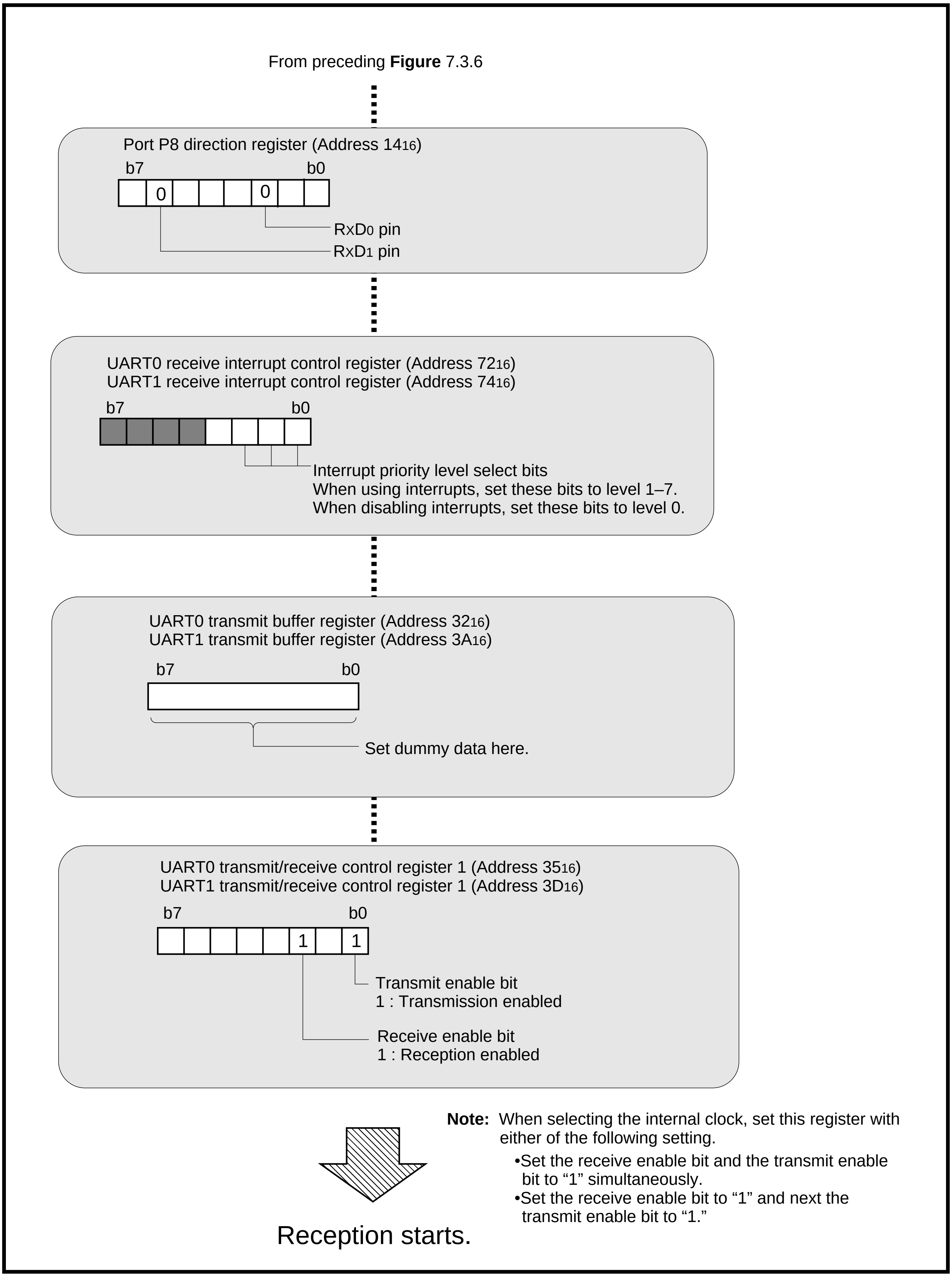


Fig. 7.3.7 Initial setting example for relevant registers when receiving (2)

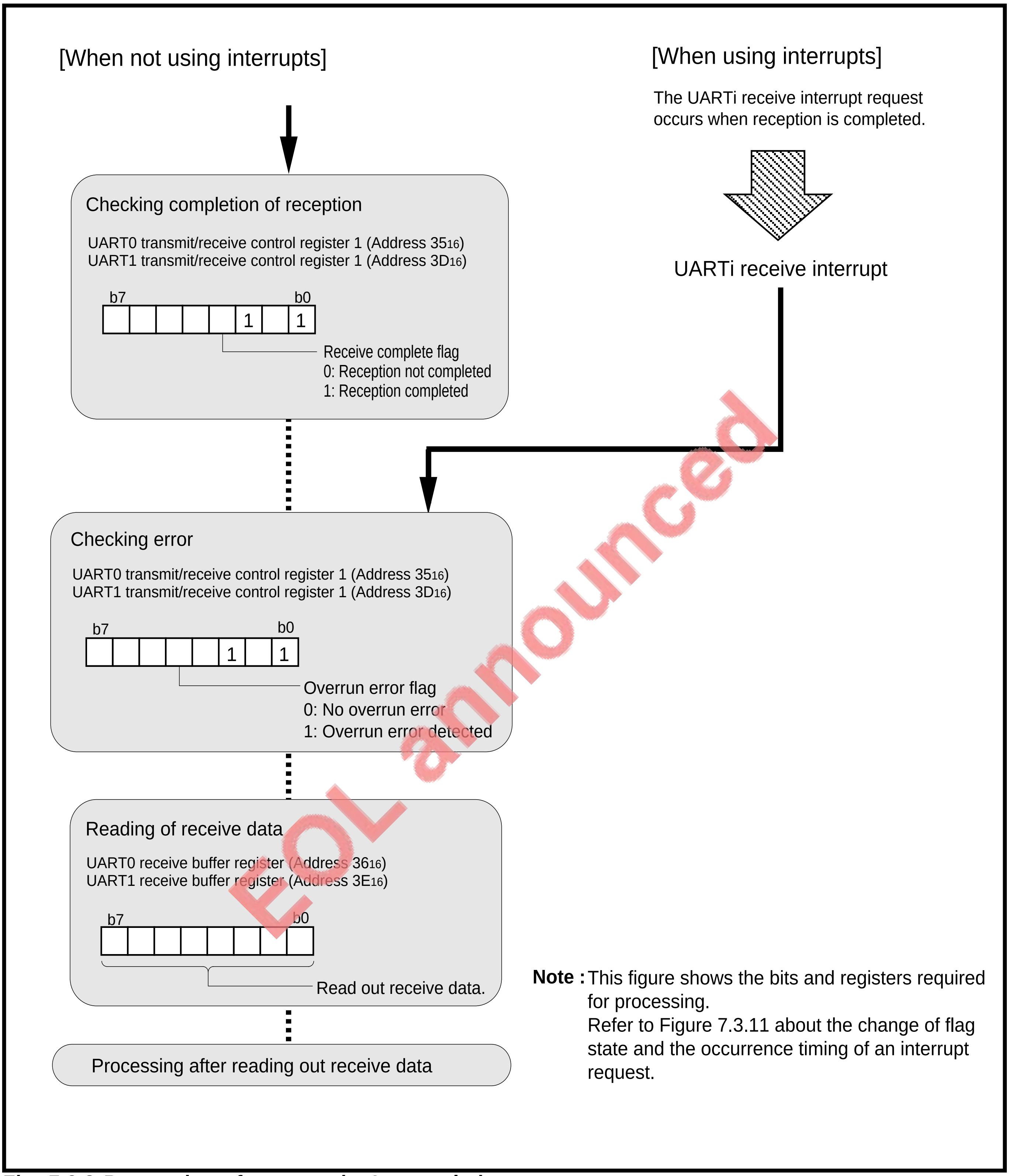


Fig. 7.3.8 Processing after reception's completion

7.3.5 Receive operation

When the receive conditions listed on page 7-25 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 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).

7.3 Clock synchronous serial I/O mode

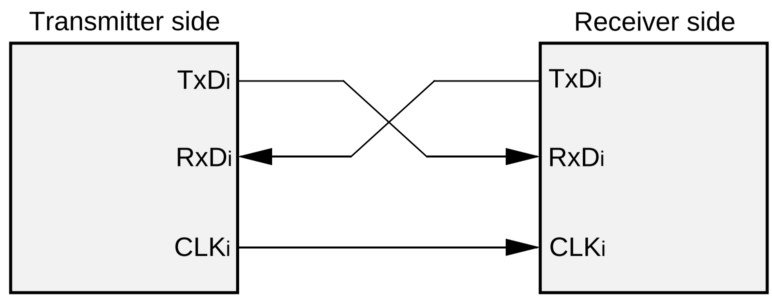


Fig. 7.3.9 Connection example

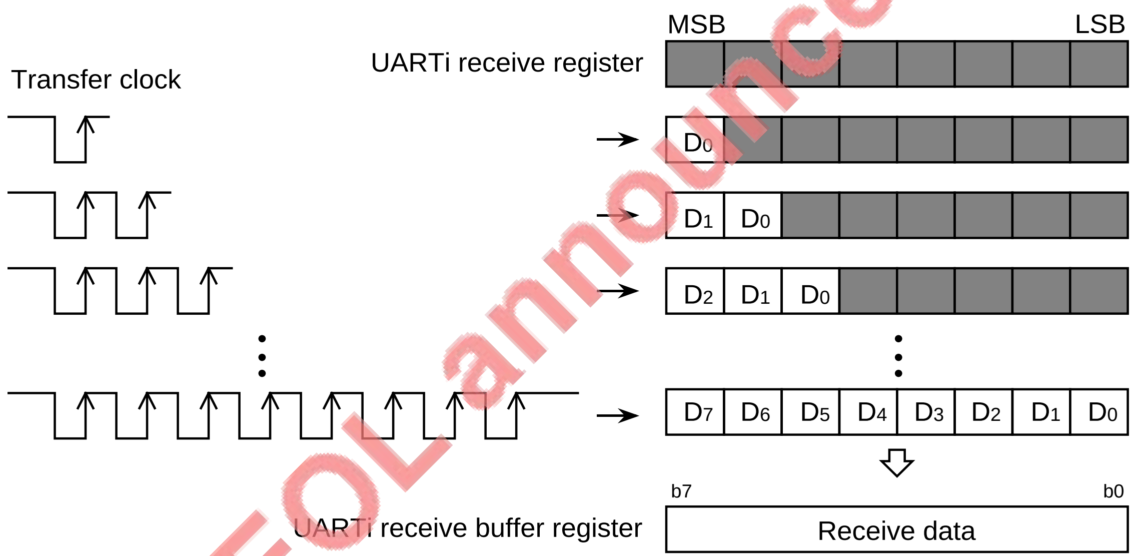
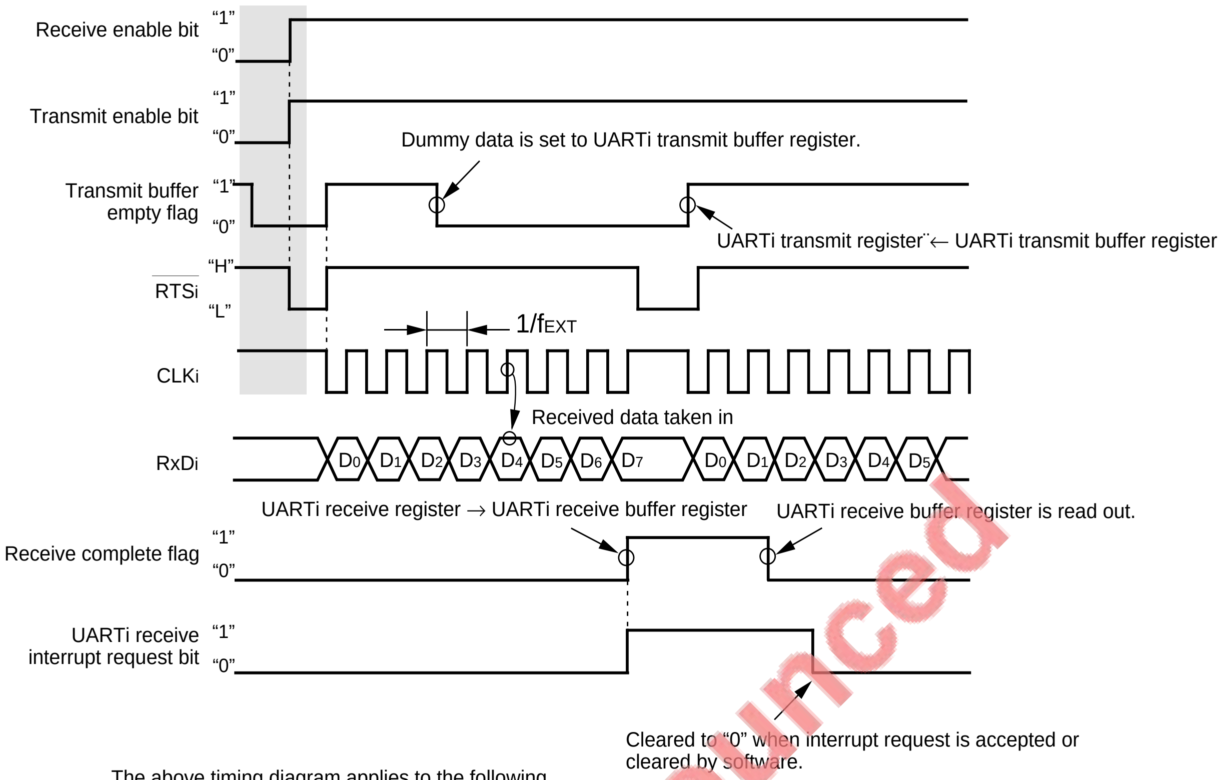


Fig. 7.3.10 Receive operation



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

f_{EXT}: Frequency of external clock

: When the CLKi pin’s input level is “H,” safisfy 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)

7.3.6 Process on detecting overrun error

In the clock synchronous serial I/O mode, an overrun error can be detected. (However it is impossible to detect an overrun error as the case may be. Refer to **6** in **"[*Precautions when operating in clock synchronous serial I/O mode*]."**

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 reading out the low-order byte of the UARTi receive buffer register or 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 an internal clock is selected during reception, the transfer clock is generated by setting the transmit enable bit to "1" (transmission enabled) and setting dummy data to the UARTi transmission buffer register. When an external clock is selected , the transfer clock is generated by setting the transmit enable bit to "1" and inputting a clock to the CLK_i pin after setting dummy data to the UARTi transmission buffer register.

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).

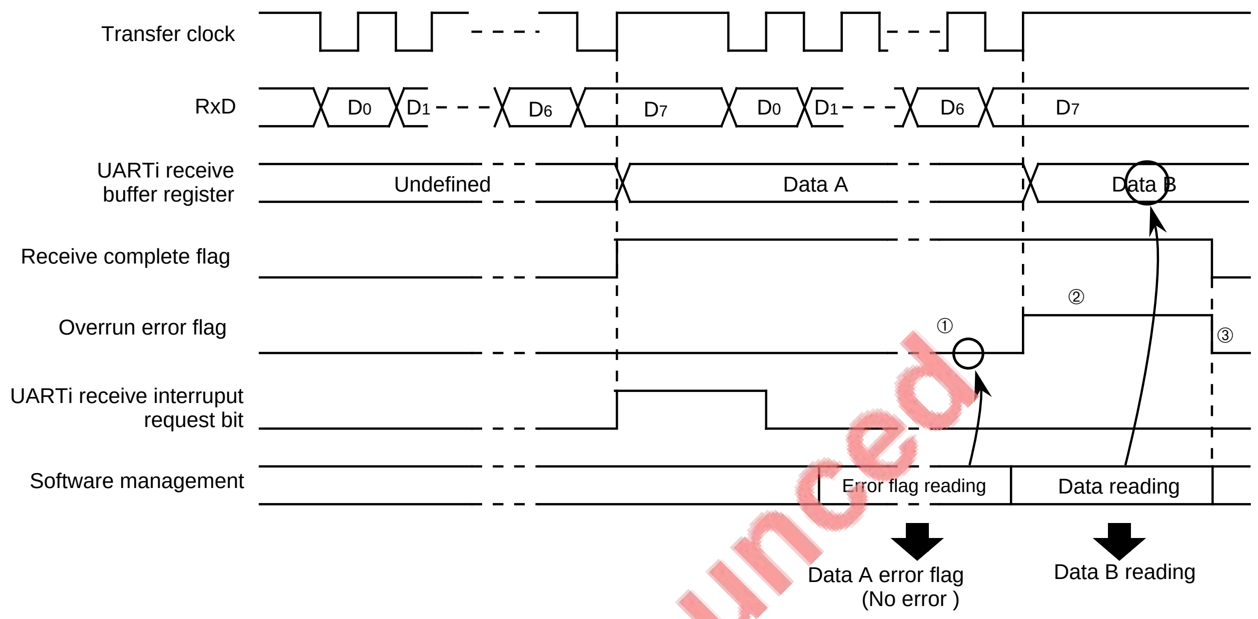
   <When receiving>
   ① Set the receive enable bit to "1."
   ② Set the transmit enable bit to "1."
   ③ Write dummy data to the UARTi transmit buffer register.

4. When receiving data, write dummy data to the low-oreder 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.

7.3 Clock synchronous serial I/O mode

6. When receiving data continuously, an overrun error cannot be detected in the following situation: when the next data reception is completed between reading the error flag by software and reading the UARTi receive buffer register.



① When checking this error flag by software, the microcomputer judges errors nothing because errors do not have occurred at data A receiving.
② When receiving the data B, the data B is written to the UARTi receive buffer register and the data A is cleard and the overrun error flag becomes “1” simultaneously. The UARTi receive interrupt request bit does not change.
③ When reading the UARTi receive buffer register by software, the data B is read and the overrun error flag becomes “0” simultaneously. Accordingly, the overrun error cannot may be detected and it is possible that the data B is managed as the data A.

Fig. 7.3.12 Case of overrun error cannot be detect (using clock synchronous seriai I/O mode)

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 ($f(X_{IN})$ = 25 MHz) Maximum 250 kbps ($f(X_{IN})$ = 16 MHz) Maximum 125 kbps ($f(X_{IN})$ = 8 MHz)
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).

7.4.1 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
F: BRGi's count source frequency
B: Transfer rate

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 and 7.4.4 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)	$f(X_{IN})$ = 8 MHz			$f(X_{IN})$ = 16 MHz		
	BRGi count source	BRGi setting value : n	Actual time (bps)	BRGi count source	BRGi setting value : n	Actual time (bps)
75	f_{512}	12 ($0C_{16}$)	75.12	f_{512}	25 (19_{16})	75.12
110	f_{64}	70 (46_{16})	110.04	f_{64}	141 ($8D_{16}$)	110.04
134.5	f_{64}	57 (39_{16})	134.70	f_{64}	115 (73_{16})	134.70
150	f_{64}	51 (33_{16})	150.24	f_{64}	103 (67_{16})	150.24
300	f_{64}	25 (19_{16})	300.48	f_{64}	51 (33_{16})	300.48
600	f_{64}	12 ($0C_{16}$)	600.96	f_{64}	25 (19_{16})	600.96
1200	f_{16}	25 (19_{16})	1201.92	f_{16}	51 (33_{16})	1201.92
2400	f_{16}	12 ($0C_{16}$)	2403.85	f_{16}	25 (19_{16})	2403.85
4800	f_2	51 (33_{16})	4807.69	f_2	103 (67_{16})	4807.69
9600	f_2	25 (19_{16})	9615.39	f_2	51 (33_{16})	9615.39
19200	f_2	12 ($0C_{16}$)	19230.77	f_2	25 (19_{16})	19230.77
31250	f_2	7 (07_{16})	31250.00	f_2	15 ($0F_{16}$)	31250.00
62500	f_2	3 (03_{16})	62500.00	f_2	7 (07_{16})	62500.00
125000	f_2	1 (01_{16})	125000.00	f_2	3 (03_{16})	125000.00
250000	f_2	0 (00_{16})	250000.00	f_2	1 (01_{16})	250000.00
500000	f_2	———	———	f_2	0 (00_{16})	500000.00

Table 7.4.4 Setting examples of transfer rate (2)

Transfer rate (bps)	f(XIN) = 24.576 MHz			f(XIN) = 25 MHz		
	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_{64}	162 ($A2_{16}$)	149.78
300	f_{64}	79 ($4F_{16}$)	300.00	f_{64}	80 (50_{16})	301.41
600	f_{16}	159 ($9F_{16}$)	600.00	f_{16}	162 ($A2_{16}$)	599.12
1200	f_{16}	79 ($4F_{16}$)	1200.00	f_{16}	80 (50_{16})	1205.63
2400	f_{16}	39 (27_{16})	2400.00	f_{16}	40 (28_{16})	2381.86
4800	f_2	159 ($9F_{16}$)	4800.00	f_2	162 ($A2_{16}$)	4792.94
9600	f_2	79 ($4F_{16}$)	9600.00	f_2	80 (50_{16})	9645.06
19200	f_2	39 (27_{16})	19200.00	f_2	40 (28_{16})	19054.88
31250				f_2	24 (18_{16})	31250.00

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 7.2.2.) Set the same transfer data format for both transmitter and receiver sides.
Figure 7.4.2 shows an example of transfer data format. Table 7.4.5 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

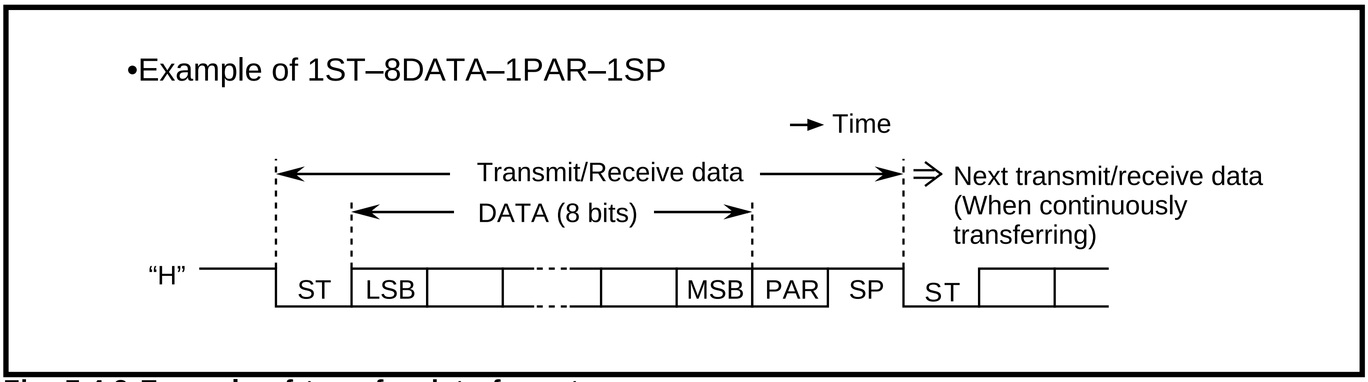


Fig. 7.4.2 Example of transfer data format

Table 7.4.5 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 Clock asynchronous serial I/O (UART) mode

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 $\overline{CTSi}$ 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 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.

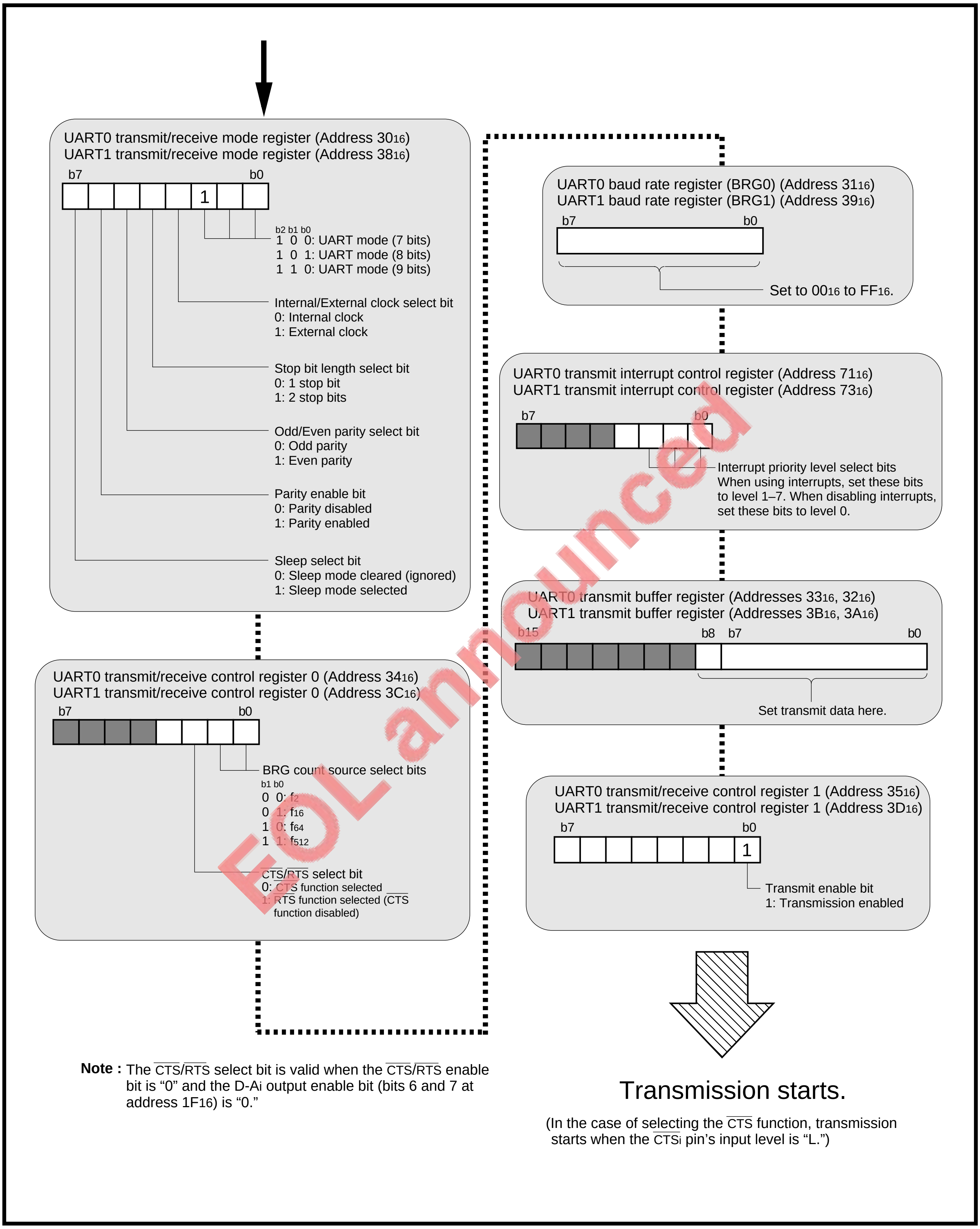


Fig. 7.4.3 Initial setting example for relevant registers when transmitting

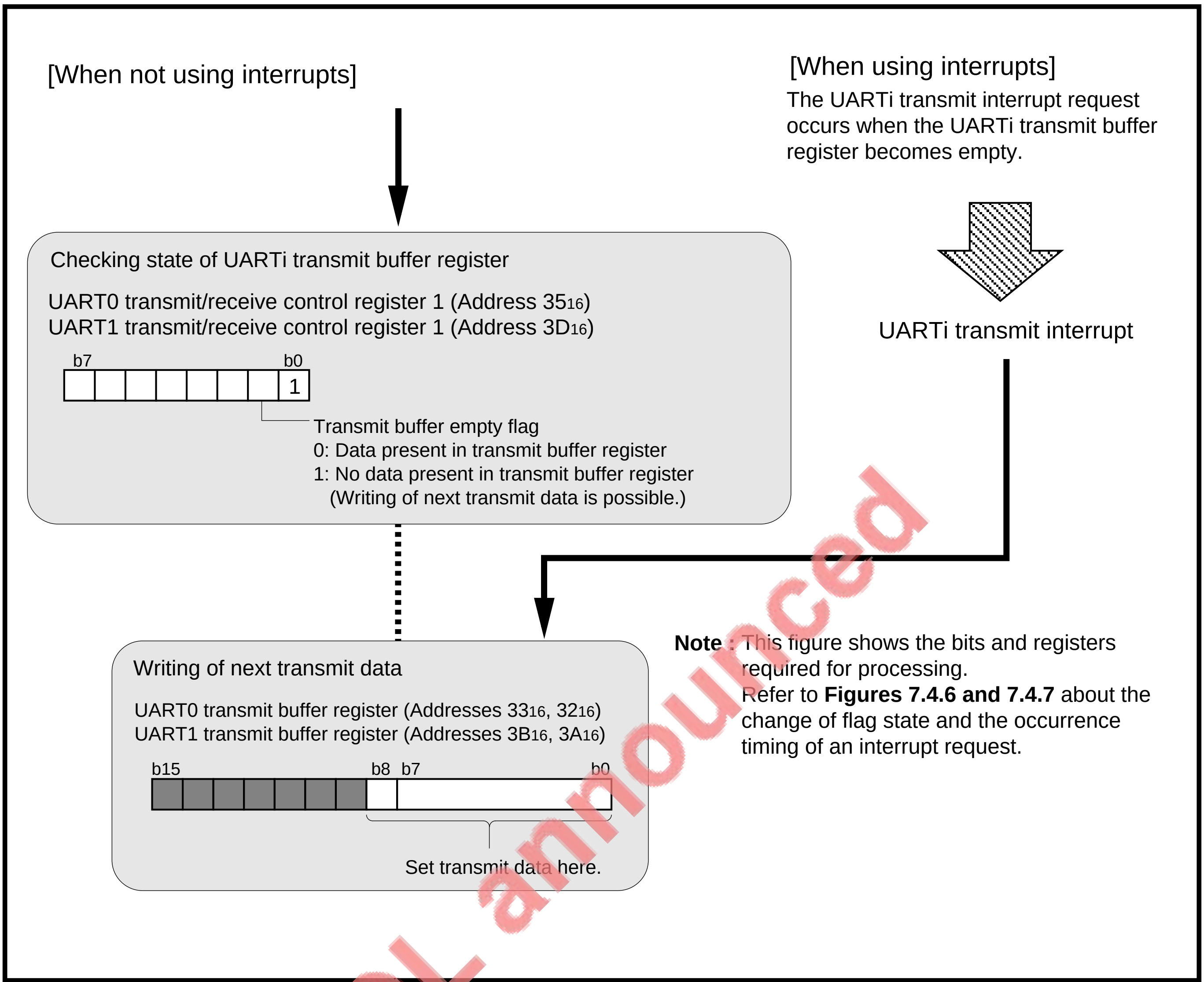


Fig. 7.4.4 Writing data after start of transmission

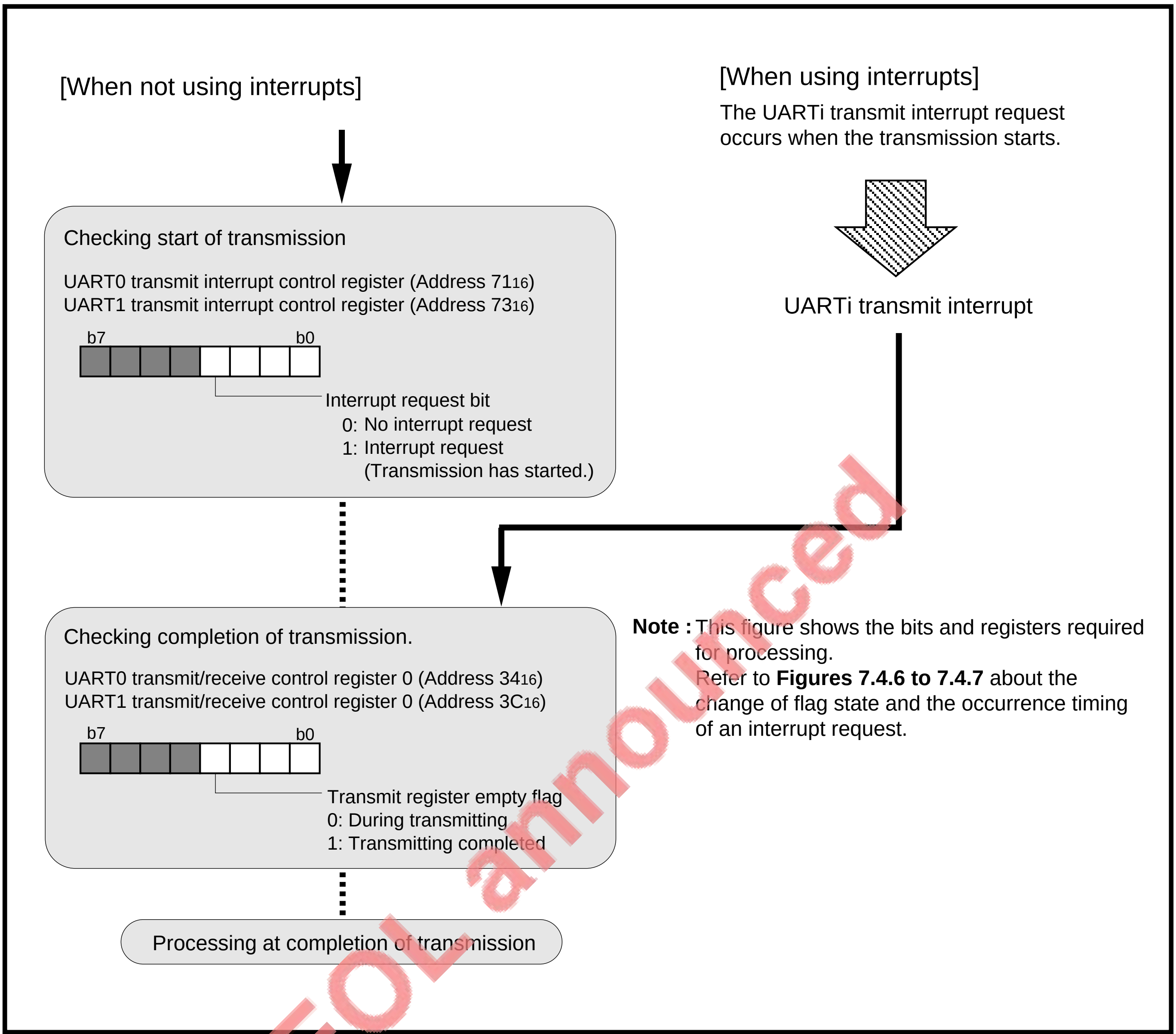


Fig. 7.4.5 Detection of transmission's completion

7.4 Clock asynchronous serial I/O (UART) mode

7.4.4 Transmit operation

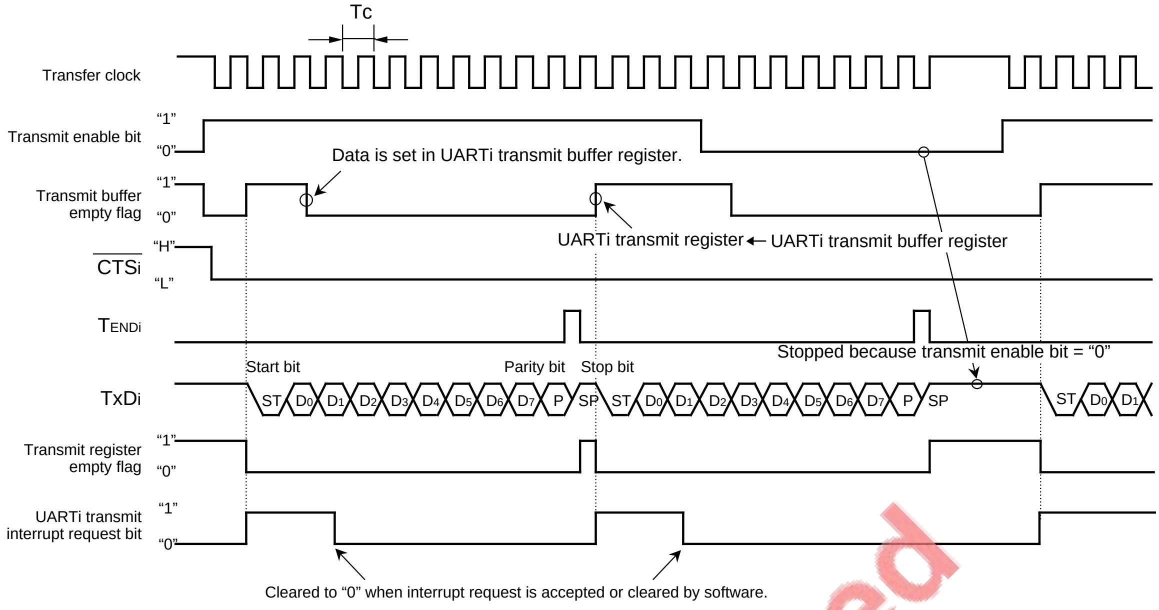
Simultaneously when the transmit conditions listed on page 7-40 are satisfied, the following operations are automatically performed.

- 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.

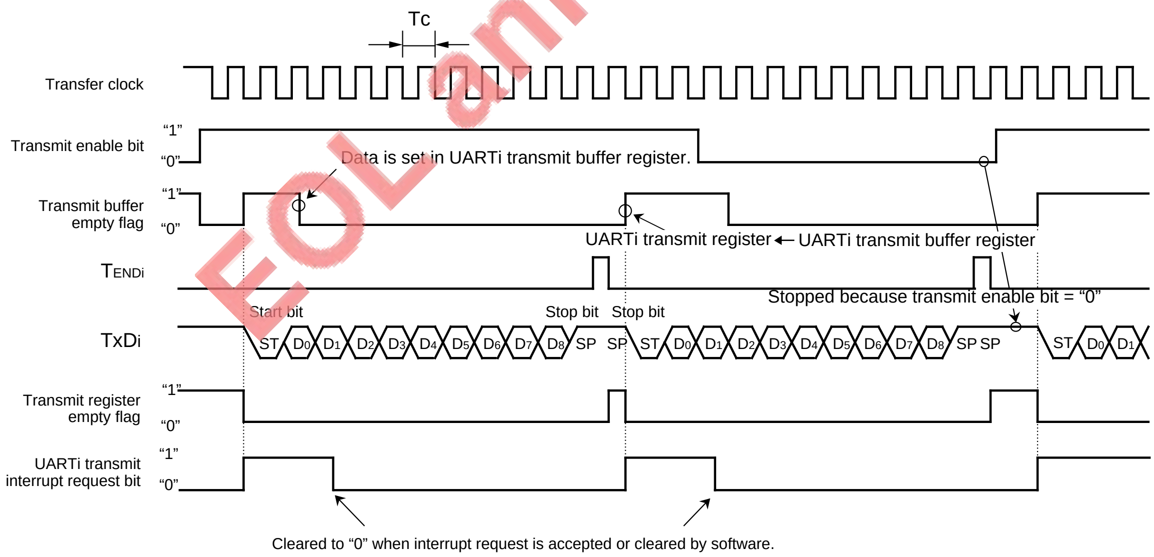


The above timing diagram applies to the following conditions:
- Parity enabled
- 1 stop bit
- CTS function selected

TENDi: Next transmit conditions are examined when this signal level is "H."
(TENDi is an internal signal. Accordingly, it cannot be read from an external.)

Tc: 16(n + 1)/fi or 16(n + 1)/fEXT
fi: BRGi count source frequency (f2, f16, f64, f512)
fEXT: BRGi count source frequency (external clock)
n: Value set to BRGi

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



The above timing diagram applies to the following conditions:
- Parity disabled
- 2 stop bits
- CTS function disabled

TENDi: Next transmit conditions are examined when this signal level is "H."
(TENDi is an internal signal. Accordingly, it cannot be read from an external.)

Tc: 16(n + 1)/fi or 16(n + 1)/fEXT
fi: BRGi count source frequency (f2, f16, f64, f512)
fEXT: BRGi count source frequency (external clock)
n: Value set to BRGi

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.

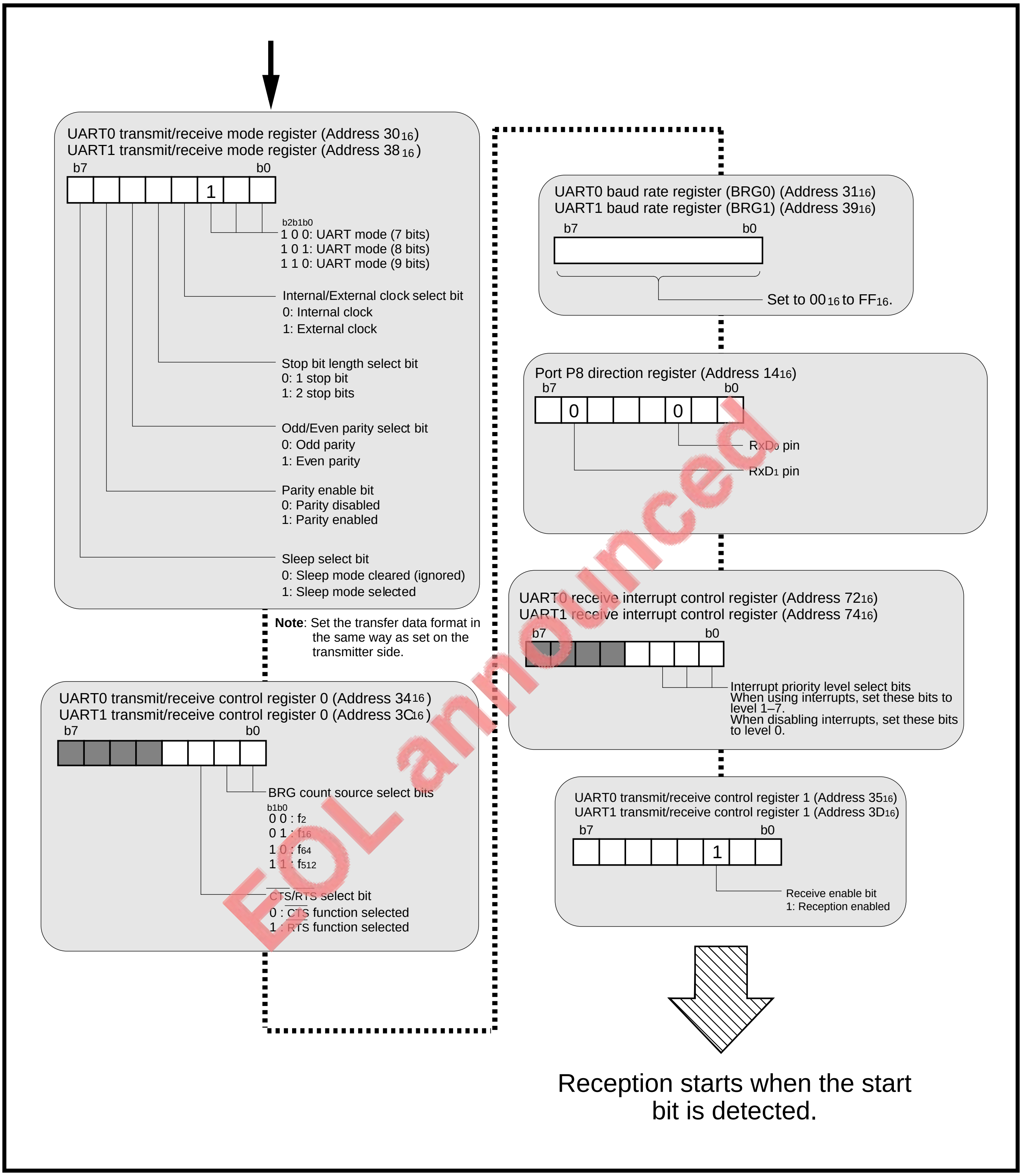


Fig. 7.4.8 Initial setting example for relevant registers when receiving

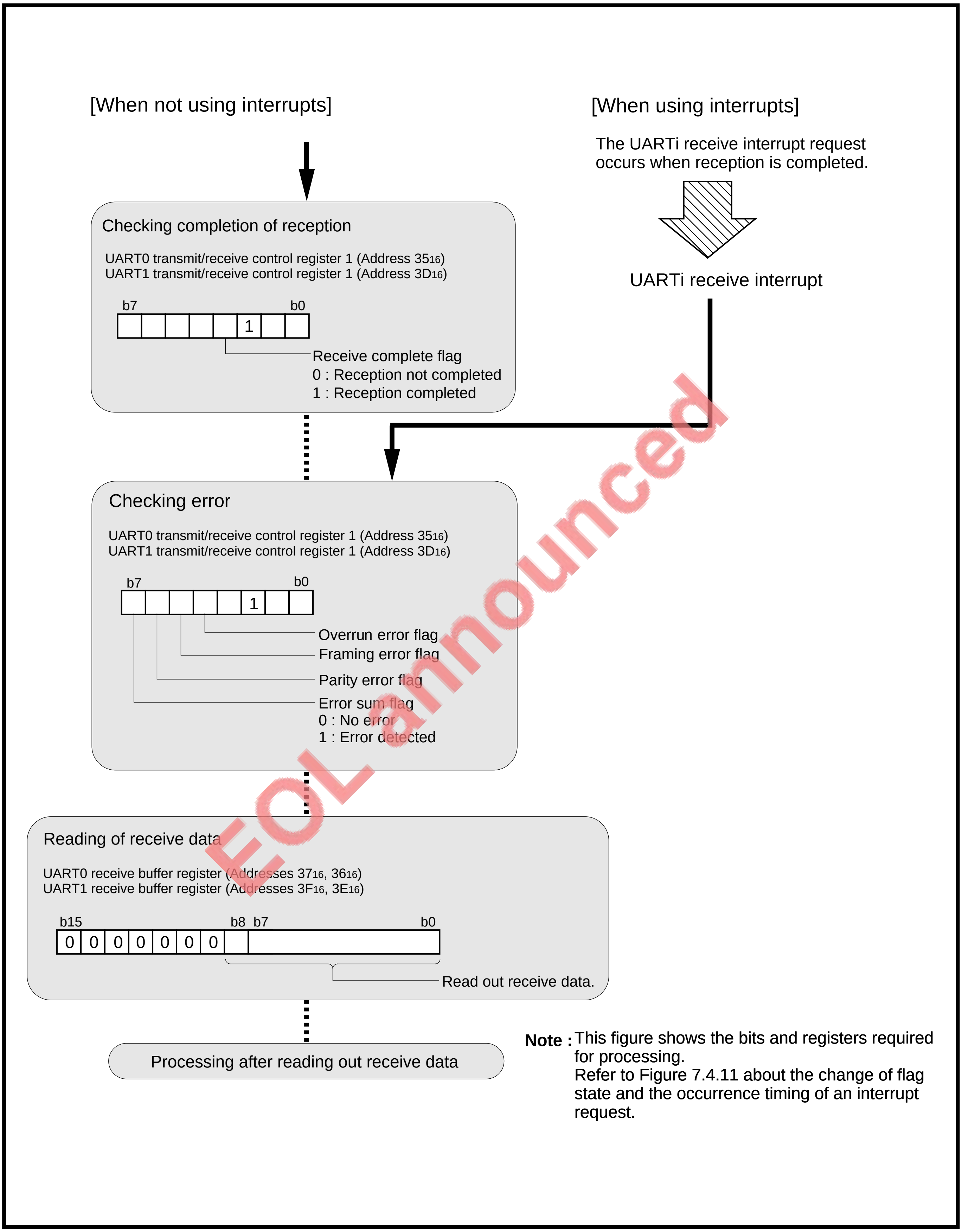


Fig. 7.4.9 Processing after reception's completion

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 RxDi 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.

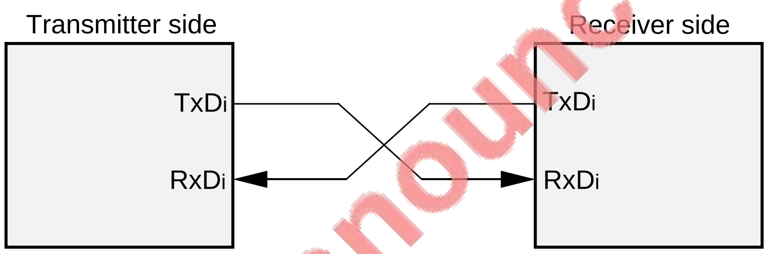


Fig. 7.4.10 Connection example

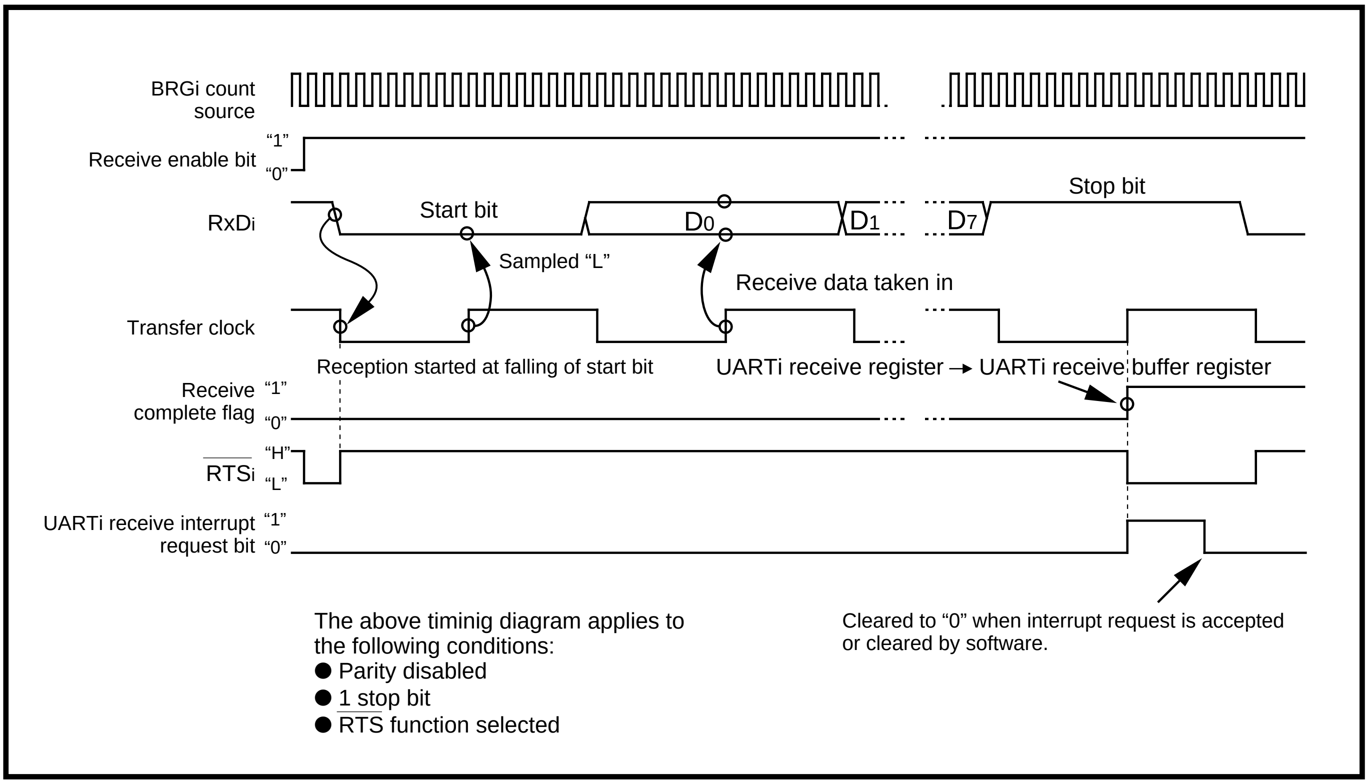


Fig. 7.4.11 Example of receive timing when transfer data length is 8 bits (when parity disabled, selecting 1 stop bit)

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. However it is impossible to detect an overrun error as the case may be. Refer to **1** in **"*[Precautions when operating in clock asynchronous serial I/O mode]*."**

●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.
Error flags such as the overrun error flag, the framing error flag, the parity error flag, the error sum 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."

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.

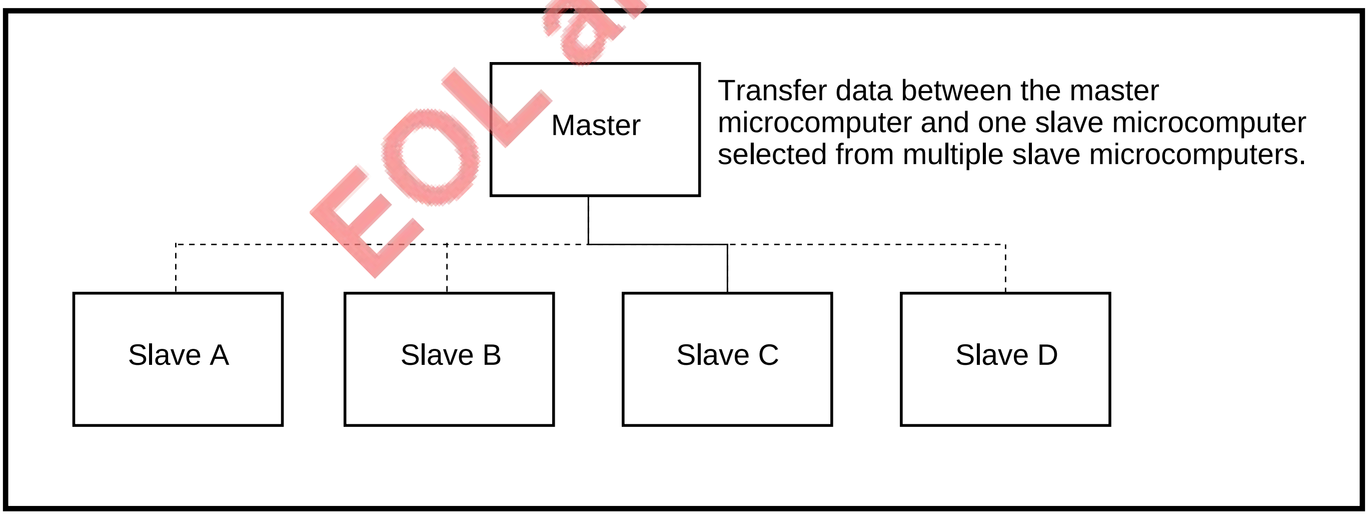
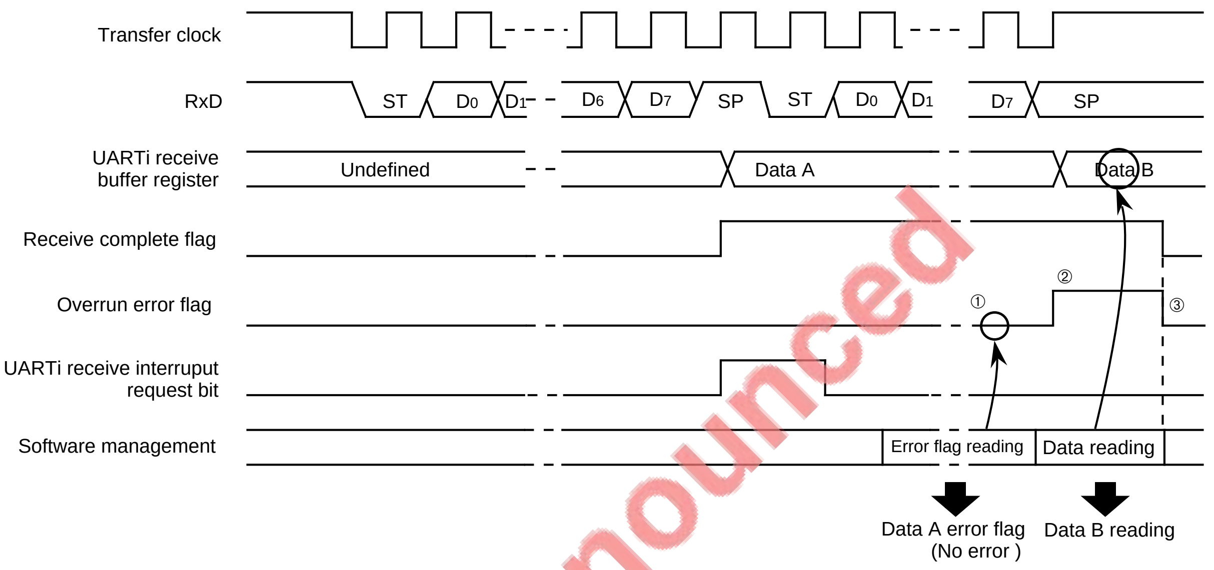


Fig. 7.4.12 Sleep mode

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

When receiving data continuously, an overrun error cannot be detected in the following situation: when the next data reception is completed between reading the error flag by software and reading the UARTi receive buffer register.

●8-bit data length, parity disabled, 1 stop bit



① When checking this error flag by software, the microcomputer judges errors nothing because errors do not have occurred at data A receiving.
② When receiving the data B, the data B is written to the UARTi receive buffer register and the data A is cleard and the overrun error flag becomes “1” simultaneously. The UARTi receive interrupt request bit does not change.
③ When reading the UARTi receive buffer register by software, the data B is read and the overrun error flag becomes “0” simultaneously. Accordingly, the overrun error cannot may be detected and it is possible that the data B is managed as the data A.

Fig. 7.4.13 Case of overrun error cannot be detect (using clock asynchronous seriai I/O mode)

MEMORANDUM

CHAPTER 8

A-D CONVERTER

This chapter describes the A-D converter.
The 7702 Group has a built-in 8-bit A-D converter. The A-D converter performs successive approximation conversion. The 7702 Group has the 8 analog input pins.

7703 Group

The number of the 7703 Group's analog input pins is different from the 7702 Group's. Refer to "**Chapter 20. 7703 GROUP**" for more information.

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	8 bits
Absolute accuracy	±3 LSB
Analog input pin	8 pins (AN_0 to AN_7) (**Note**)
Conversion rate per analog input pin	57 ϕ_{AD}* cycles

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

Note: In the 7703 Group, the analog input pins are 4 pins, AN_0 to AN_2, AN_7. Refer to "**Chapter 20. 7703 GROUP**" for more information.

The A-D converter has the 4 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
  This mode is used to perform the operation repeatedly for voltages input from multiple selected analog input pins.

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.

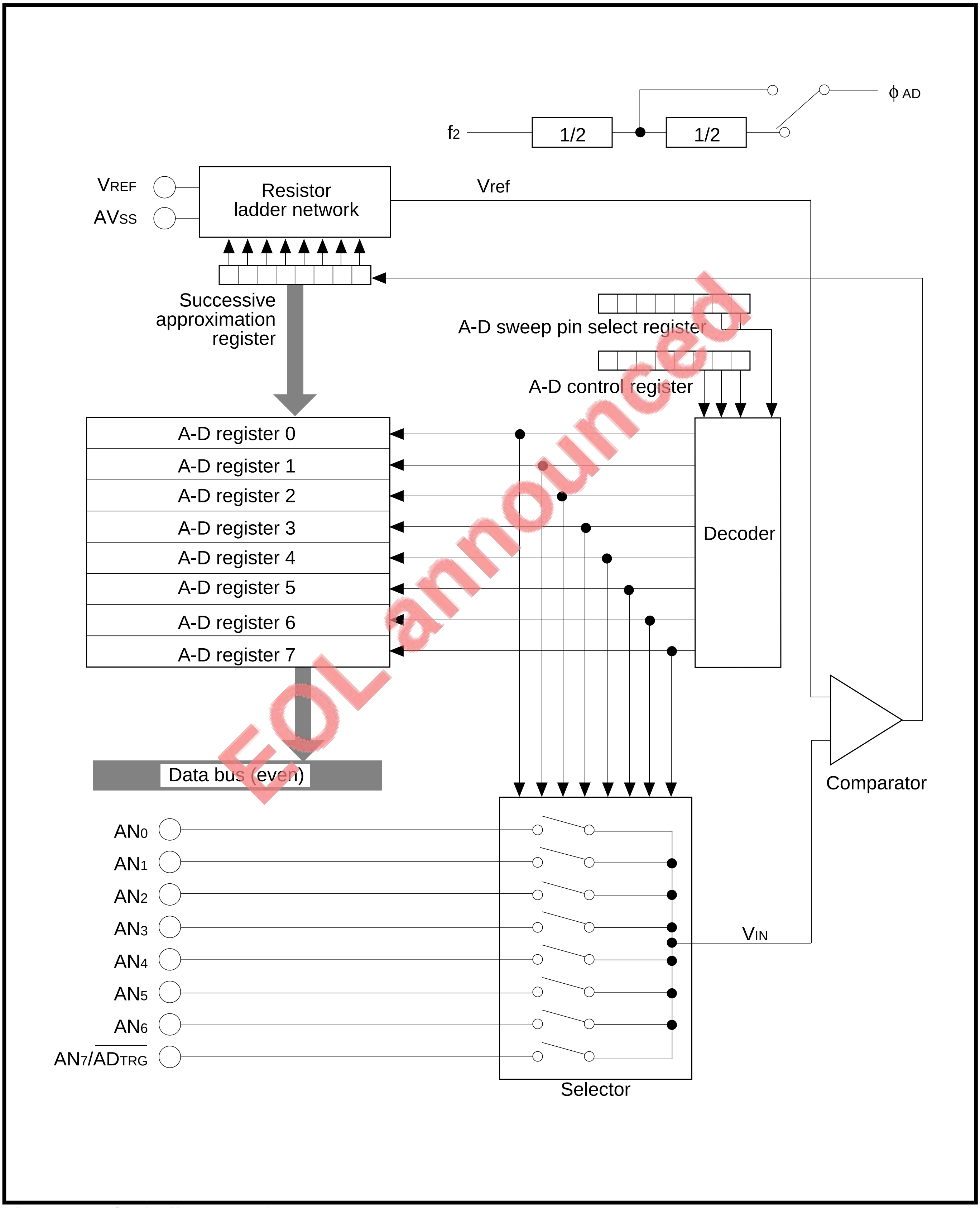
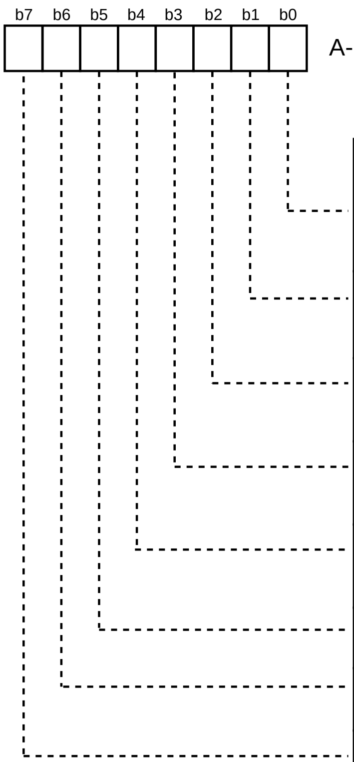


Fig. 8.2.1 Block diagram of A-D converter

8.2 Block description

8.2.1 A-D control register

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



A-D control register (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	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode	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 : f_2 divided by 4 1 : f_2 divided by 2	0	RW

Notes 1: These bits are ignored in the single sweep and repeat sweep mode. (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: Writing to each bit (except bit 6) of the A-D control register must be performed while the A-D converter halts.

Fig. 8.2.2 Structure of A-D control register

(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 or repeat sweep mode.

(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 or repeat sweep mode, the A-D converter continues operating until this bit is cleared to "0" by software.

● **When external trigger is selected**

When the $\overline{ADTRG}$ 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 or repeat sweep mode, the A-D converter continues operating until this bit is cleared to "0" by software.

(4) A-D conversion frequency (ϕ_{AD}) select bit (bit 7)

As shown in Table 8.2.1, the operating time of the A-D converter varies depending on the selected operating clock (ϕ_{AD}) by this bit.

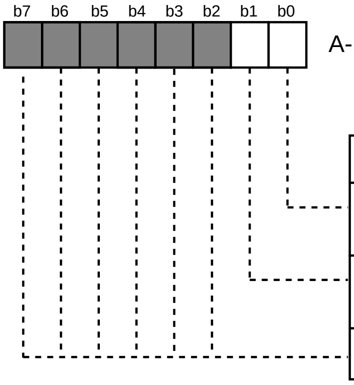
Since the A-D converter's comparator consists of capacity coupling amplifiers, keep that $\phi_{AD} \geq 250$ kHz during A-D conversion.

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

A-D conversion frequency (ϕ_{AD}) select bit		0	1
ϕ_{AD}		$f_2/4$	$f_2/2$
Conversion time	$f(X_{IN})$ = 8 MHz	57.0	28.5
	$f(X_{IN})$ = 16 MHz	28.5	14.25
	$f(X_{IN})$ = 25 MHz	18.24	9.12

8.2.2 A-D sweep pin select register

Figure 8.2.3 shows the structure of the A-D sweep pin select register.



A-D sweep pin select register (Address $1F_{16}$)

Bit	Bit name	Functions	At reset	RW
0	A-D sweep pin select bits (Valid in single sweep and repeat sweep mode) **(Note 1)**	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			1	RW
7 to 2	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: Writing to each bit of the A-D sweep pin select register must be performed while the A-D converter halts.

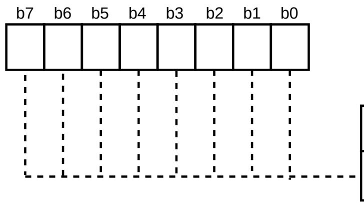
Fig. 8.2.3 Structure of A-D control register 1

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

These bits are used to select analog input pins in the single sweep mode or repeat sweep mode. In the single sweep mode and repeat sweep mode, 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.



A-D register 0 (Addresses 20_{16})
A-D register 1 (Addresses 22_{16})
A-D register 2 (Addresses 24_{16})
A-D register 3 (Addresses 26_{16})
A-D register 4 (Addresses 28_{16})
A-D register 5 (Addresses $2A_{16}$)
A-D register 6 (Addresses $2C_{16}$)
A-D register 7 (Addresses $2E_{16}$)

Bit	Functions	At reset	RW
7 to 0	Reads an A-D conversion result.	Undefined	RO

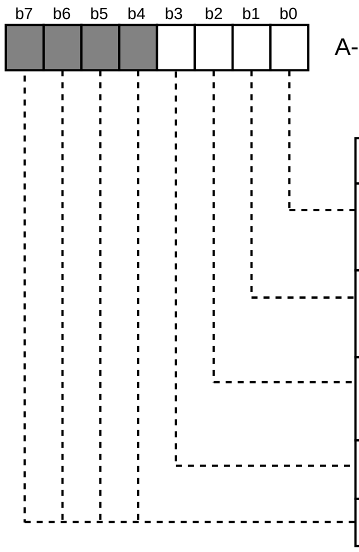
Fig. 8.2.4 Structure of A-D register i

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	0	RW
7 to 4	Nothing is assigned.		Undefined	–

Note : Use the **SEB** or **CLB** instruction to set the A-D conversion interrupt control register.

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

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 7 of the successive approximation register**
The A-D converter compares V_{ref} with V_{IN}. At this time, the contents of the successive approximation register is "10000000_2" (initial value).
Bit 7 of the successive approximation register changes according to the comparison result as follows:
When $V_{ref} < V_{IN}$, bit 7 = "1"
When $V_{ref} > V_{IN}$, bit 7 = "0"

② **Determining bit 6 of the successive approximation register**
After setting bit 6 of the successive approximation register to "1," the A-D converter compares V_{ref} with V_{IN}. Bit 6 changes according to the comparison result as follows:
When $V_{ref} < V_{IN}$, bit 6 = "1"
When $V_{ref} > V_{IN}$, bit 6 = "0"

③ **Determining bits 5 to 0 of the successive approximation register**
Operations in ② are performed for bits 5 to 0.
When bit 0 is determined, the contents (conversion result) of the successive approximation register is 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 lists 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.

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 255	$\frac{V_{REF}^*}{256} \times (n - 0.5)$

V_{REF}*: Reference voltage

Table 8.3.2 Change in successive approximation register and V_{ref} during A-D conversion

	Successive approximation register								Change of V_{ref}
	b7							b0	
A-D converter halt	1	0	0	0	0	0	0	0	$\frac{V_{REF}}{2}$ [V]
1st comparison	1	0	0	0	0	0	0	0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{512}$ [V]
2nd comparison	n_7 (↑ 1st comparison result)	1	0	0	0	0	0	0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{512}$ [V] (•$n_7=1$ $+\frac{V_{REF}}{4}$; •$n_7=0$ $-\frac{V_{REF}}{4}$)
3rd comparison	n_7	n_6 (↑ 2nd comparison result)	1	0	0	0	0	0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{512}$ [V] (•$n_6=1$ $+\frac{V_{REF}}{8}$; •$n_6=0$ $-\frac{V_{REF}}{8}$)
⋮	⋮								⋮
8th comparison	n_7	n_6	n_5	n_4	n_3	n_2	n_1	1	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} \pm \ldots\ldots \pm \frac{V_{REF}}{256} - \frac{V_{REF}}{512}$ [V]
Conversion complete	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0	

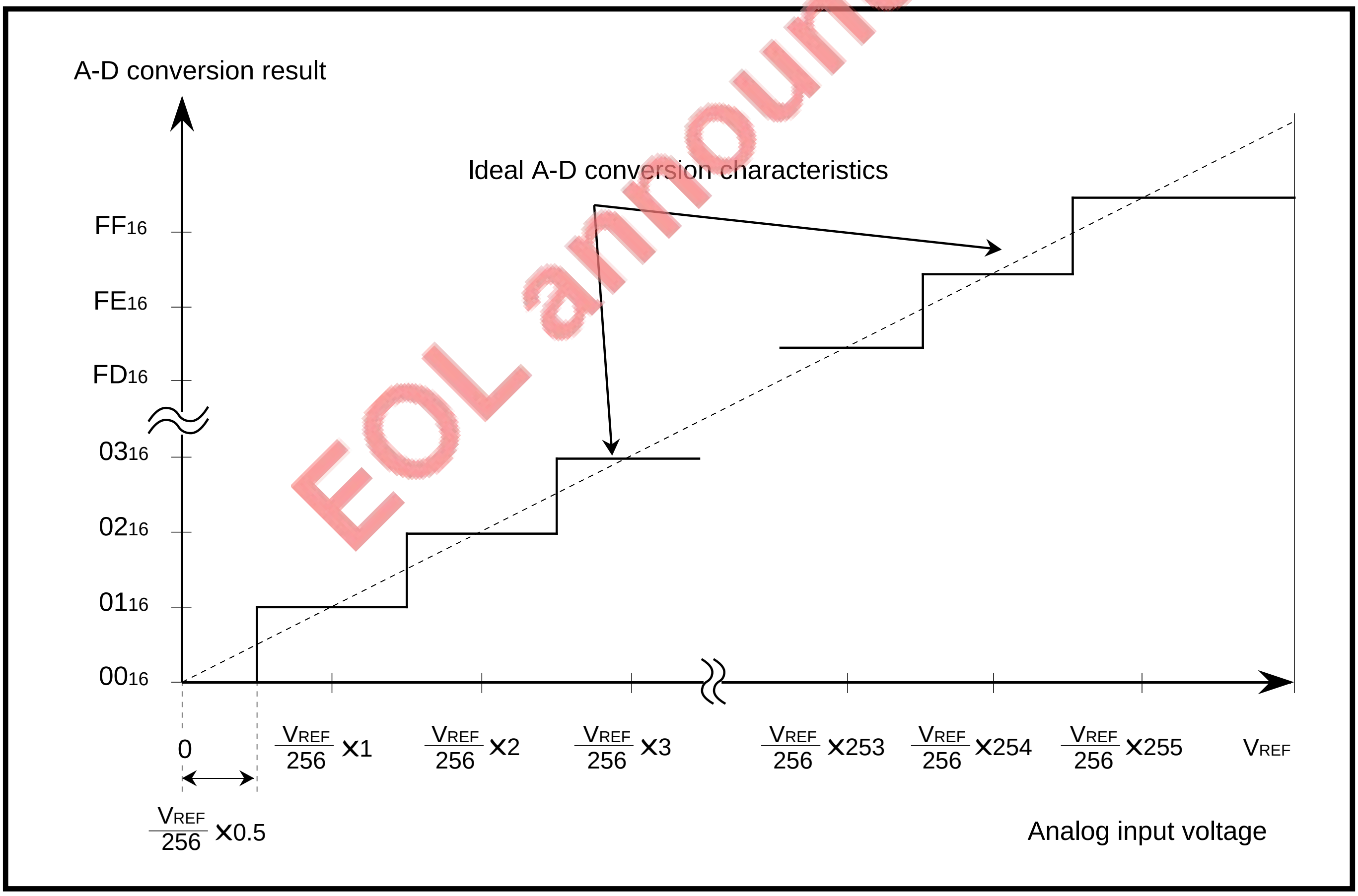


Fig. 8.3.1 Ideal A-D conversion characteristics

8.4 Absolute accuracy and differential non-linearity error

The A-D converter's accuracy is described below.

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, when V_{REF} = 5.12 V, 1 LSB width is 20 mV, and 0 mV, 20 mV, 40 mV, 60 mV, 80 mV, ... are selected as the analog input voltages.

The absolute accuracy = ±3 LSB indicates that when the analog input voltage is 100 mV, the output code expected from an ideal A-D conversion characteristics is "005_{16}," however 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 "FF_{16}."

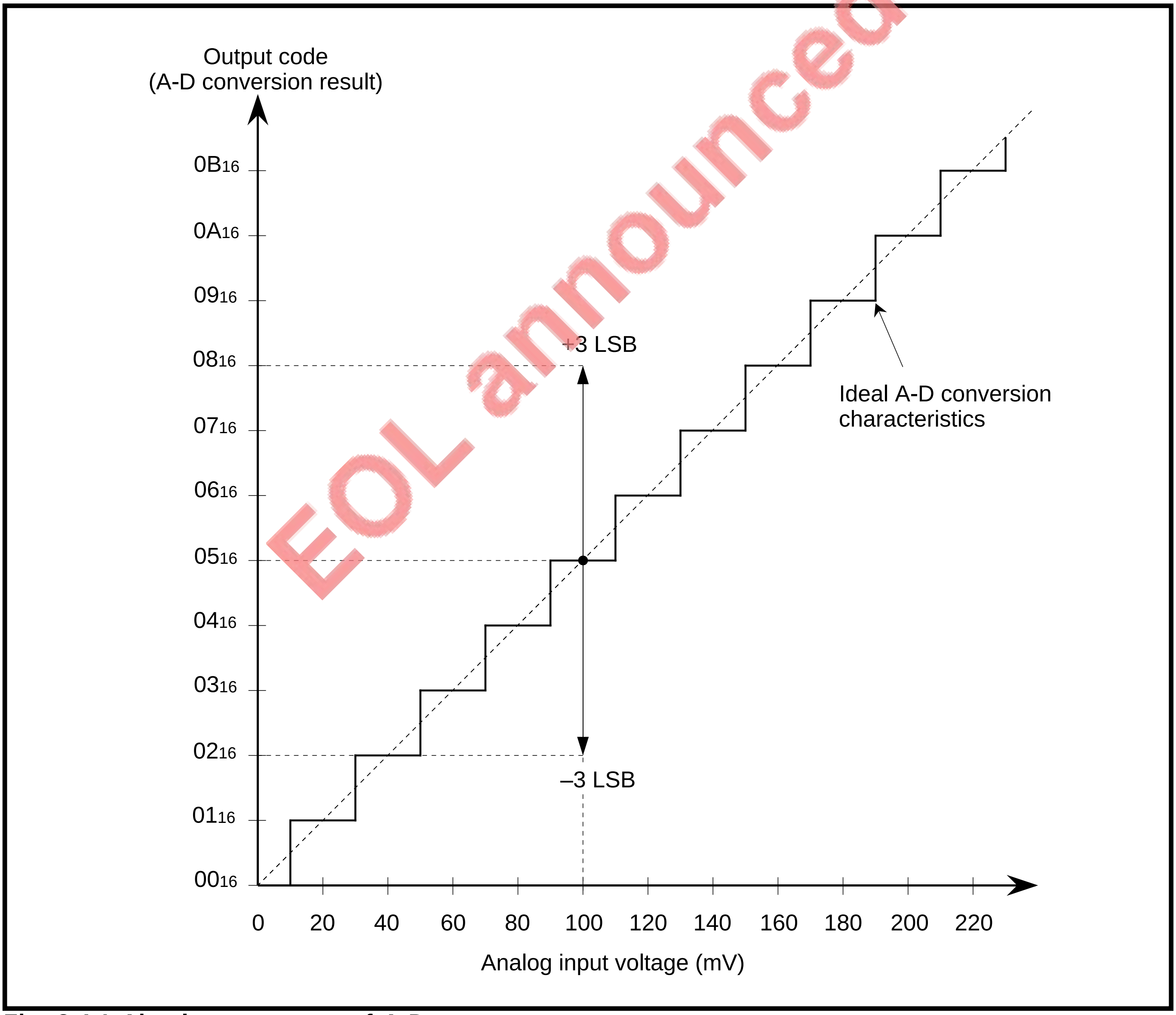


Fig. 8.4.1 Absolute accuracy of A-D converter

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, when V_{REF} = 5.12 V, the 1 LSB width of an A-D converter with ideal characteristics is 20 mV, however when the differential non-linearity error is ±1 LSB, the actual measured 1 LSB width is 0 to 40 mV. (Refer to section "**16.1.3 A-D converter standard characteristics**.")

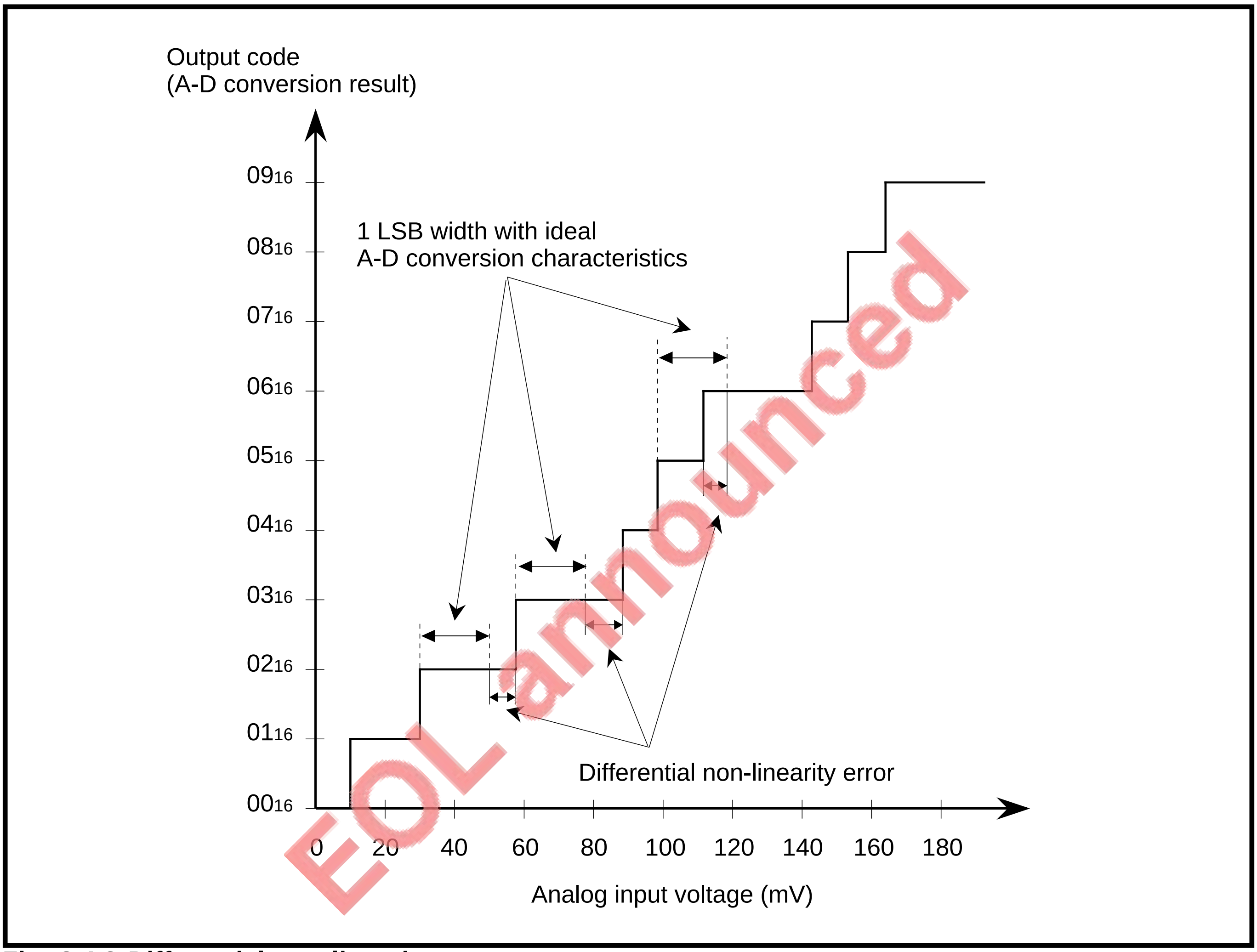


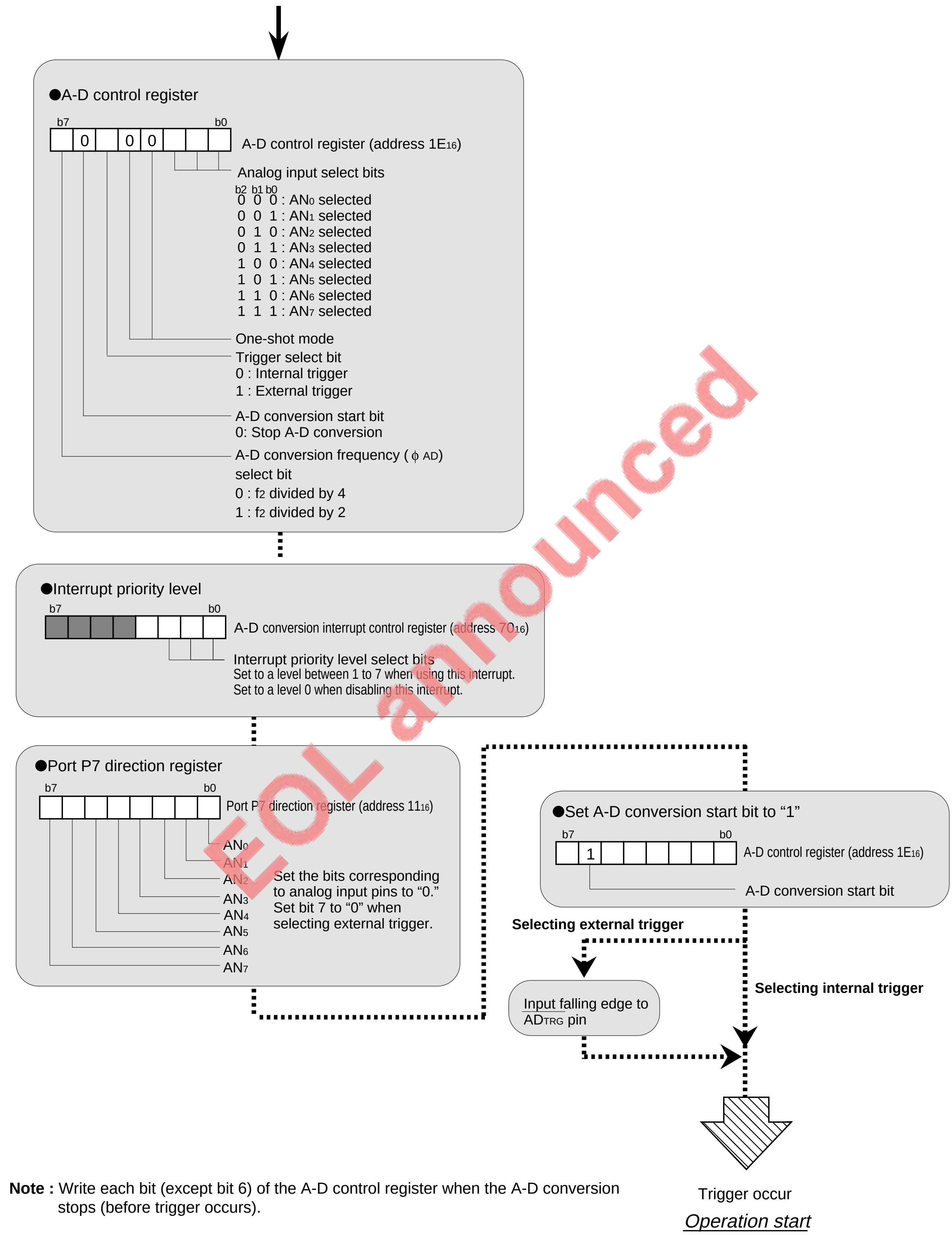
Fig. 8.4.2 Differential non-linearity error

8.5 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 the A-D conversion interrupt request occurs when the operation is completed.

8.5.1 Settings for one-shot mode

Figure 8.5.1 shows an initial setting example of the one-shot mode.
When using an interrupt, it is necessary to set the relevant registers to enable the interrupt. Refer to **“Chapter 4. INTERRUPTS”** for more descriptions.



Note : Write each bit (except bit 6) of the A-D control register when the A-D conversion stops (before trigger occurs).

Fig. 8.5.1 Initial setting example of one-shot mode

8.5.2 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 57 cycles of ϕ_{AD}. 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 57 cycles of ϕ_{AD}. 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 8.5.2 shows the conversion operation in the one-shot mode.

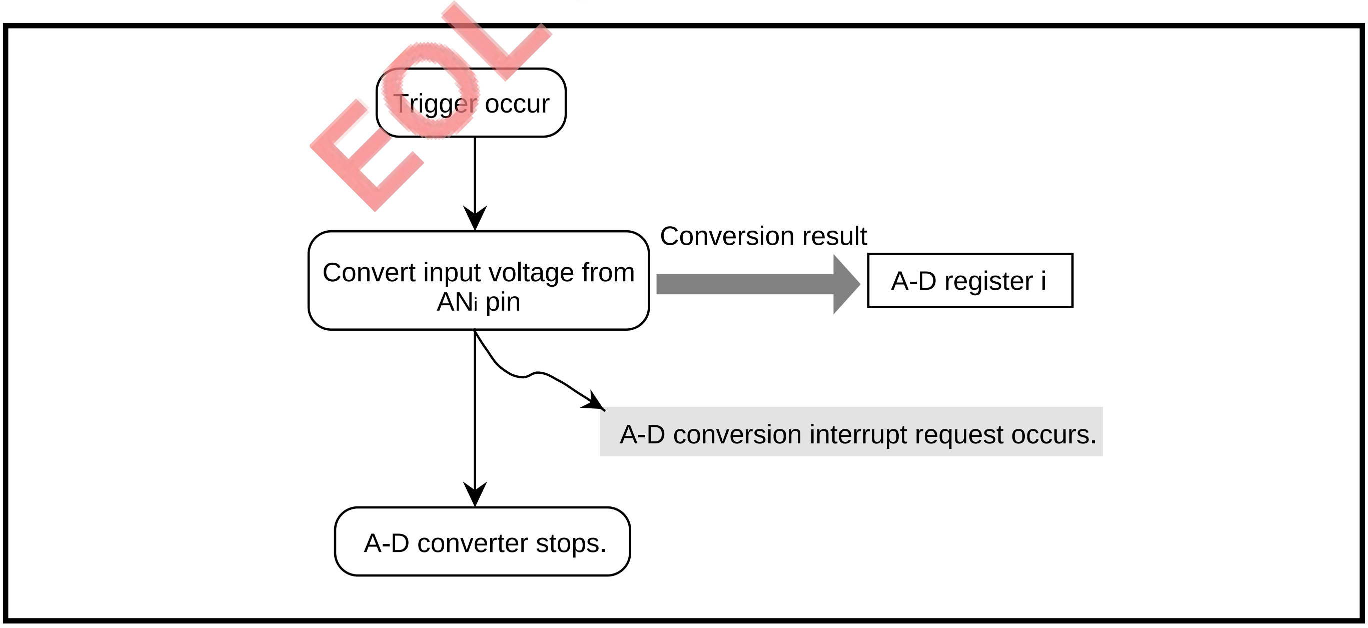


Fig. 8.5.2 Conversion operation in one-shot mode

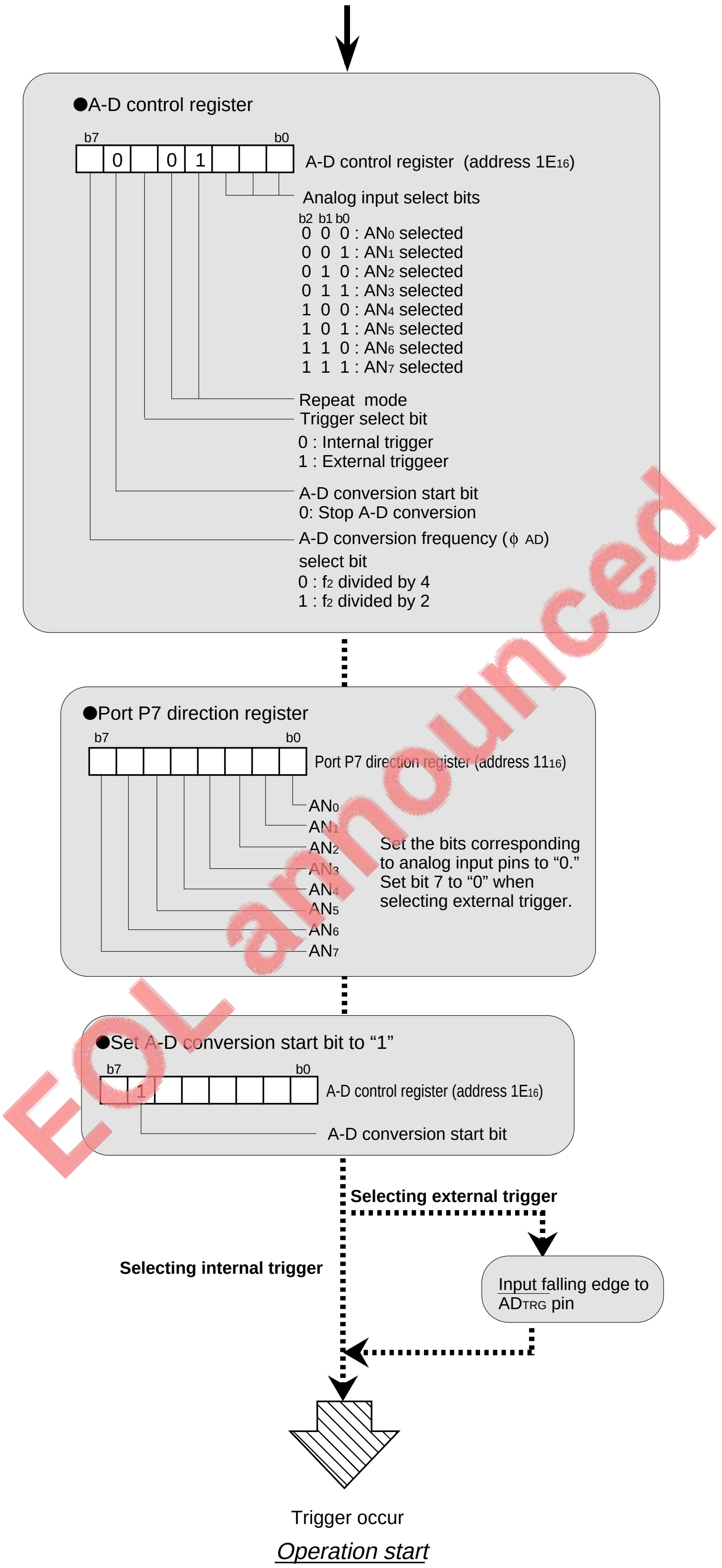
8.6 Repeat mode

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."

8.6.1 Settings for repeat mode

Figure 8.6.1 shows an initial setting example of repeat mode.



Note : Write the each bit (except bit 6) of the A-D control regiter when the A-D conversion stops (before trigger occurs).

Fig. 8.6.1 Initial setting example of repeat mode

8.6.2 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 57 cycles of ϕ_{AD}. 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 57 cycles of ϕ_{AD}. 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 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 repeat mode.

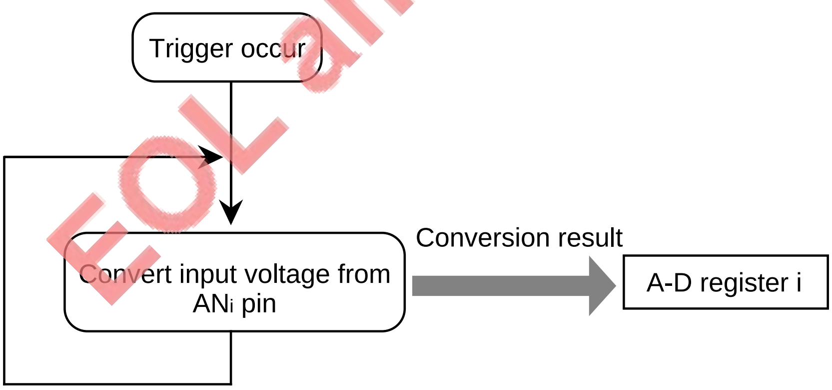


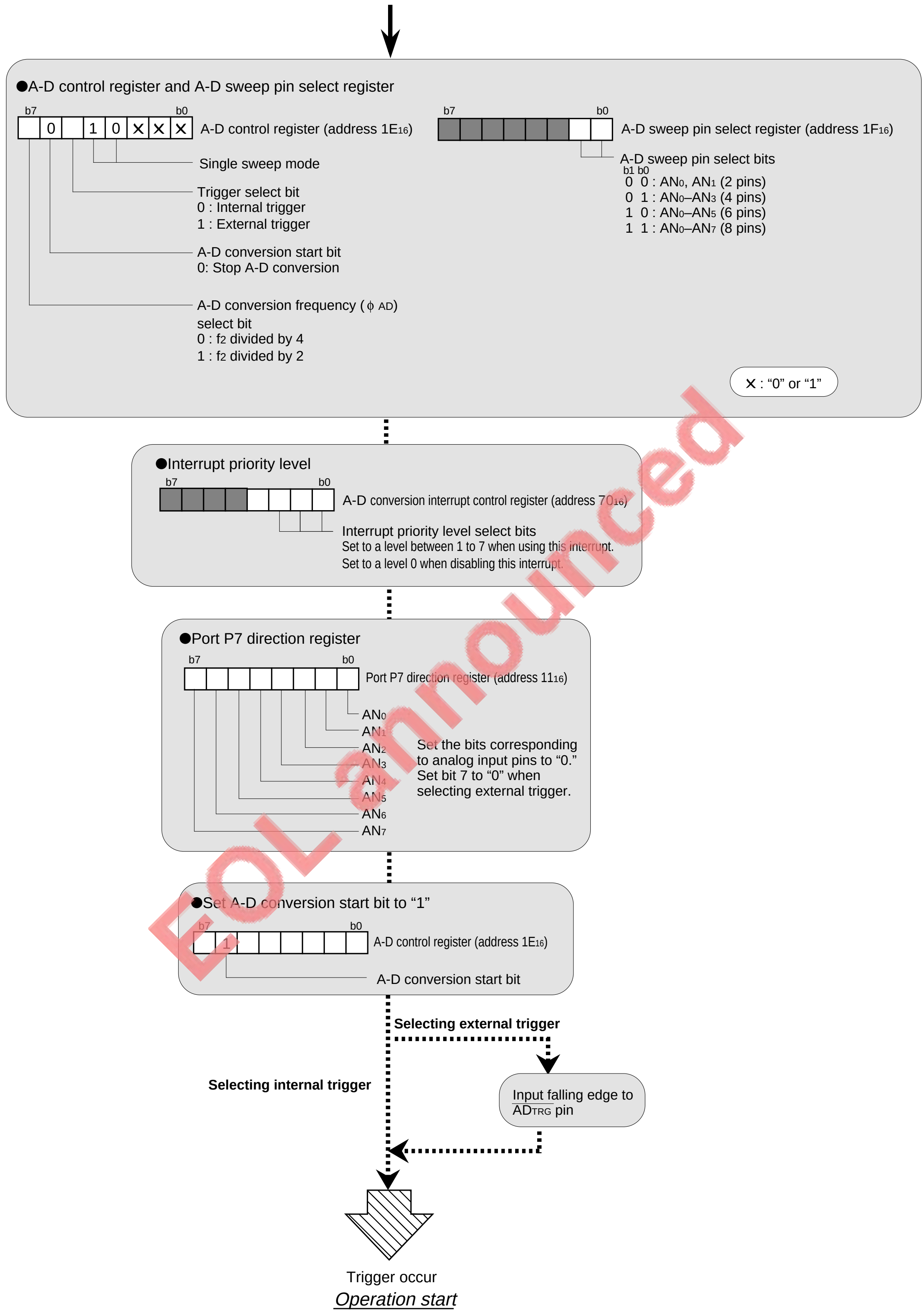
Fig. 8.6.2 Conversion operation in repeat mode

8.7 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. The A-D conversion interrupt request occurs when the operation for all selected input pins are completed.

8.7.1 Settings for single sweep mode

Figure 8.7.1 shows an initial setting example of single sweep mode.
When using an interrupt, it is necessary to set the relevant registers to enable the interrupt. Refer to **“Chapter 4. INTERRUPTS”** for more information.



Note : Write each bit (except bit 6) of the A-D control register and each bit of the A-D sweep pin select register when the A-D conversion stops (before trigger occurs).

Fig. 8.7.1 Initial setting example of single sweep mode

A-D CONVERTER

8.7 Single sweep mode

8.7.2 Single sweep mode operation description

(1) When an internal trigger is selected

① The operation 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 57 cycles of ϕ_{AD}. 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 operation 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 57 cycles of ϕ_{AD}. 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 8.7.2 shows the conversion operation in the single sweep mode.

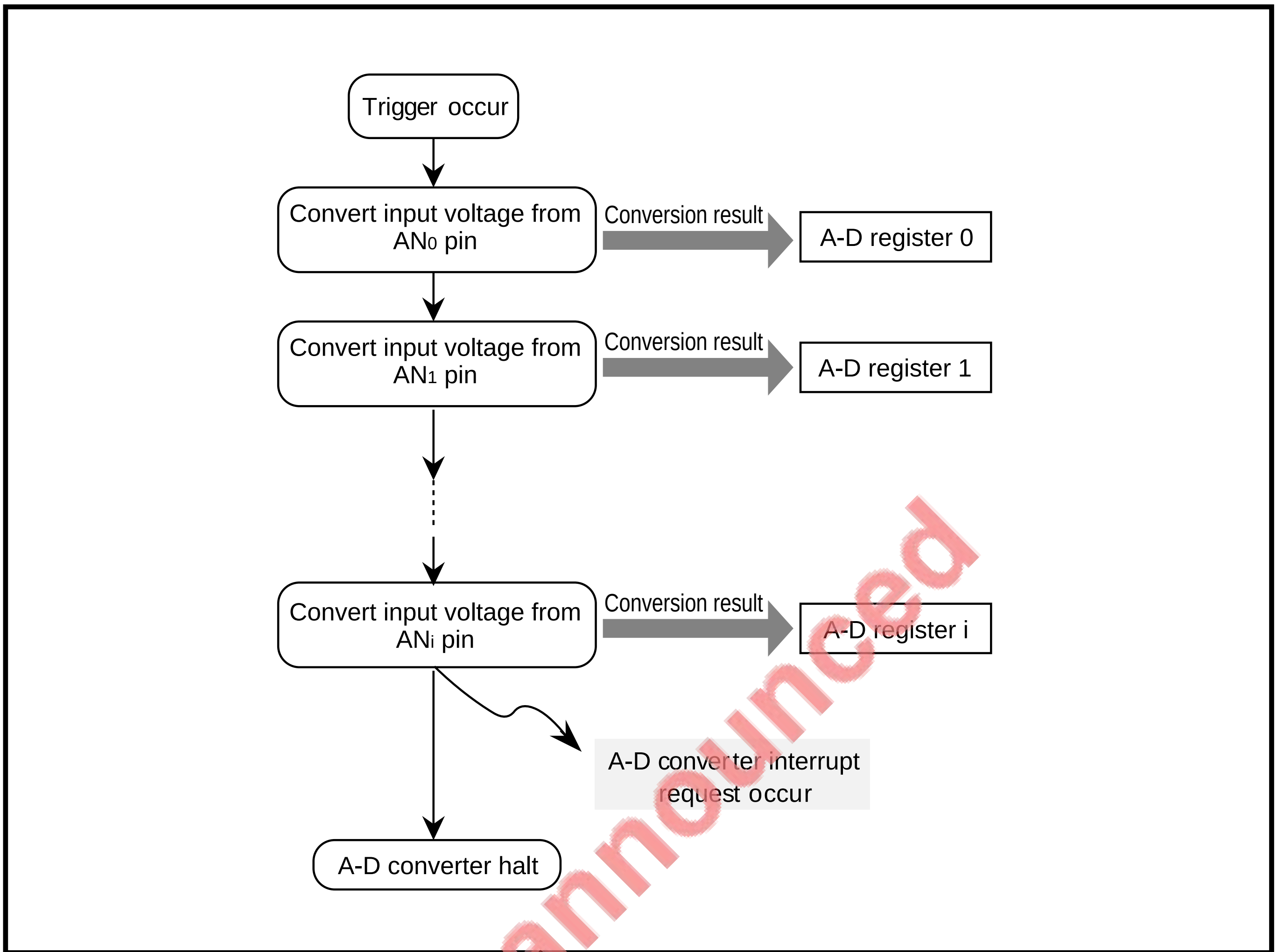


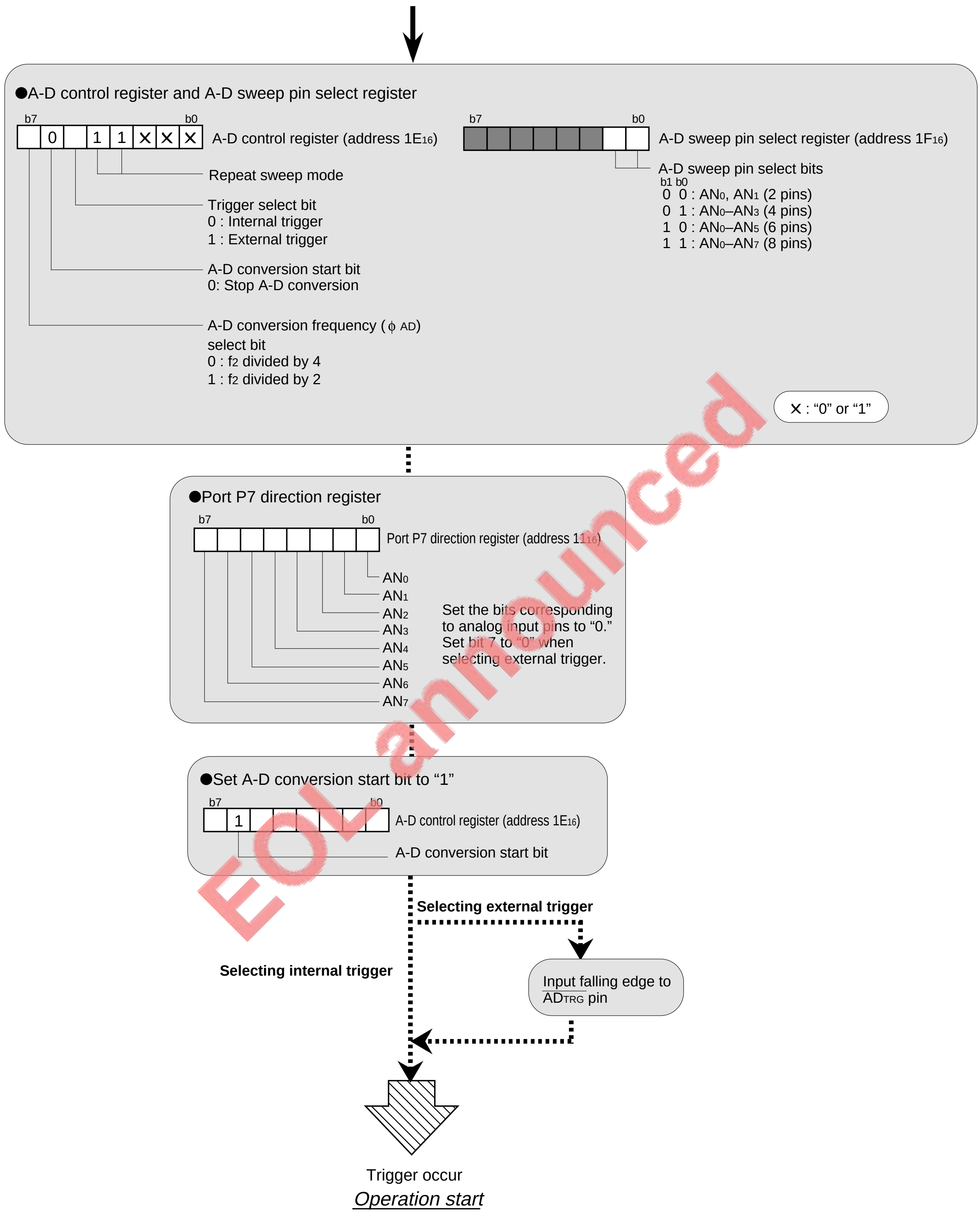
Fig. 8.7.2 Conversion operation in single sweep mode

8.8 Repeat sweep mode

In the repeat sweep mode, 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."

8.8.1 Settings for repeat sweep mode

Figure 8.8.1 shows an initial setting example of repeat sweep mode.



Note : Write each bit (except bit 6) of the A-D control register and each bit of the A-D sweep pin select register when the A-D conversion stops (before trigger occurs).

Fig. 8.8.1 Initial setting example of repeat sweep mode

8.8.2 Repeat sweep mode operation description

(1) When an internal trigger is selected

① The operation 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 57 cycles of ϕ_{AD}. 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 operation 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 57 cycles of ϕ_{AD}. 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 8.8.2 shows the conversion operation in the repeat sweep mode.

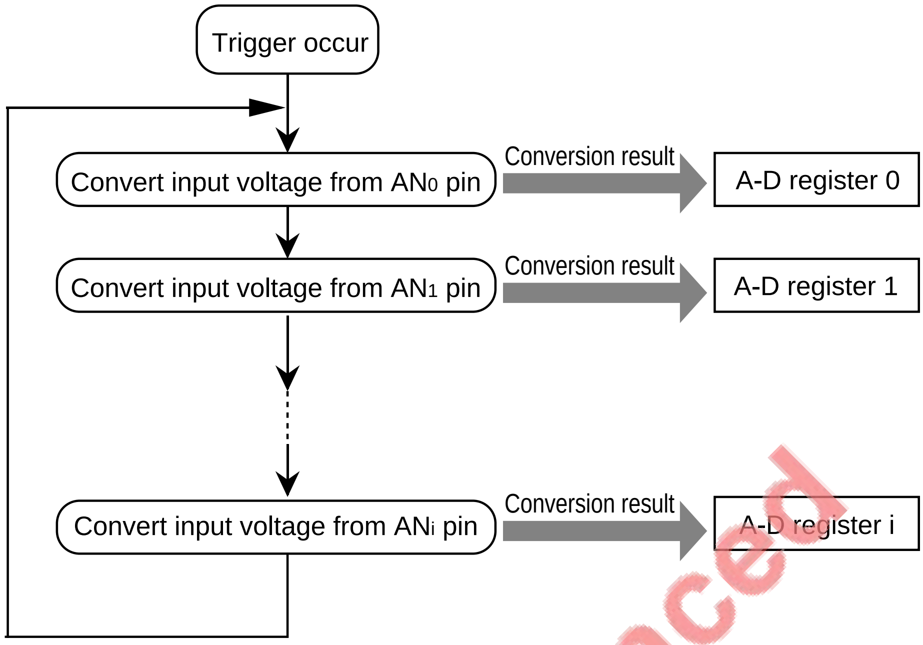


Fig. 8.8.2 Conversion operation in repeat sweep mode

8.9 Precautions when using A-D converter

1. Write to each bit (except bit 6) of the A-D control regisrer and each bit of the A-D sweep pin select register before a trigger occurs (while the A-D converter stops operation).

2. When selecting the AN_7 pin as an analog input pin while an external trigger is selected, A-D conversion is performed for a trigger input, which is the input voltage on the $\overline{AD_{TRG}}$ pin, and the conversion result is stored into the A-D register 7. Consequently, the user cannot use the AN_7 pin as an analog input pin while an external trigger is selected.

3. Refer to **“Appendix.6 Countermeasures against noise”** when using the A-D converter.

CHAPTER 9

WATCHDOG TIMER

WATCHDOG TIMER

9.1 Block description

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.

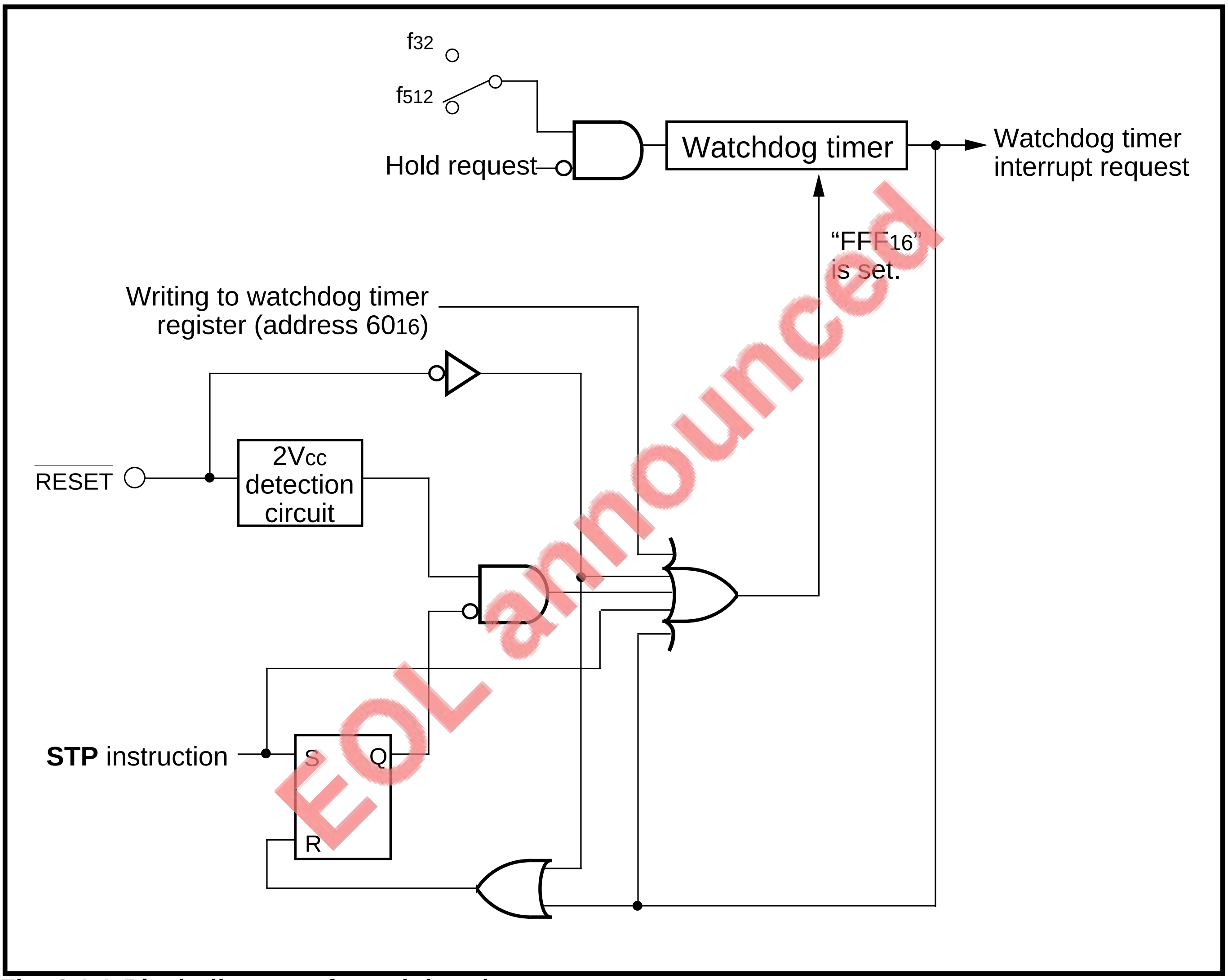


Fig. 9.1.1 Block diagram of watchdog timer

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})

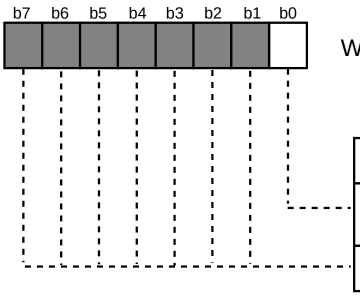
Bit	Functions	At reset	RW
7 to 0	Initializes the watchdog timer. When a dummy data is written to this register, the watchdog timer's value is initialized to "FFF_{16}." (Dummy data: 00_{16} to FF_{16})	Undefined	–

Fig. 9.1.2 Structure of watchdog timer register

9.1 Block description

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 : f_{512} 1 : f_{32}	0	RW
7 to 1	Nothing is allocated.		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

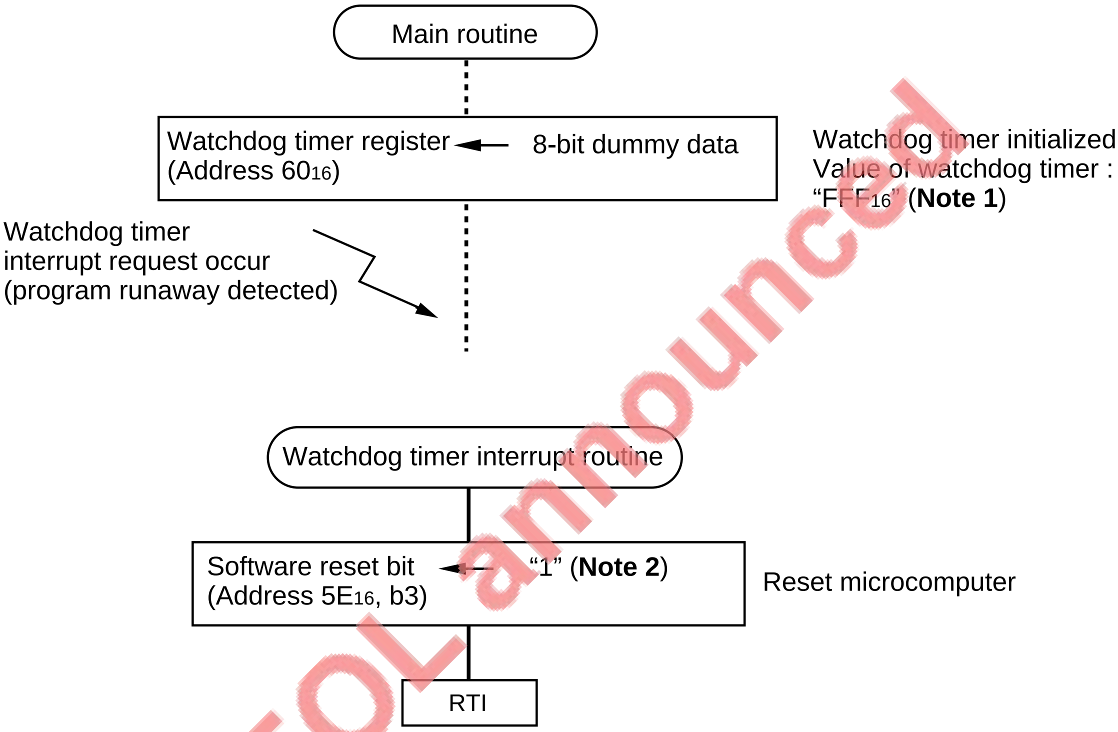
Watchdog timer frequency select bit	f(X_{IN}) = 25 MHz	
	Count source	Occurrence interval
0	f_{512}	41.94 ms
1	f_{32}	2.62 ms

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 and ϕ_{CPU} 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 f_{512} from "FFF_{16}."

(2) When Stop mode is terminated by an interrupt request occurrence

Immediately after the stop mode is terminated, Watchdog timer starts counting of the count source f_{32} from "FFF_{16}." Supply of the ϕ and ϕ_{CPU} 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} 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 f_{32} or f_{512} 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."**

WATCHDOG TIMER

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.

3. To stop Watchdog timer in Hold state, the count source which is actually counted by Watchdog timer is the logical AND product of two signals. One is the inverted signal input from the $\overline{\text{HOLD}}$ pin, and the other is the count source (f_{32} or f_{512})(Note). Accordingly, when the $\overline{\text{HOLD}}$ pin's input signal level changes in a duration which is shorter than 1 cycle of the count source **(Note)**, counting by Watchdog timer can be performed. (Refer to Figure 9.3.1.)

Note: It is selected with the watchdog timer frequency select bit.

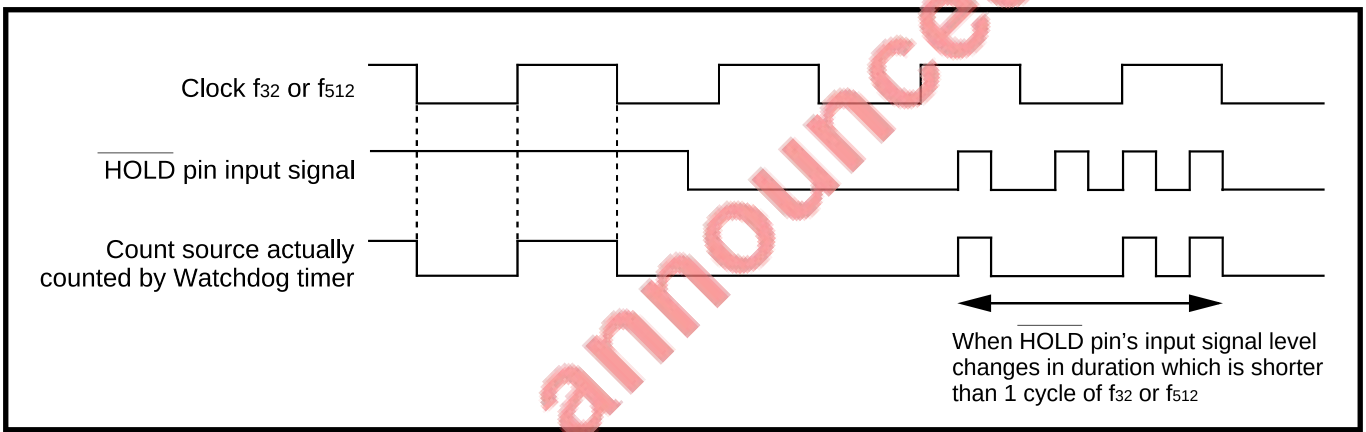


Fig. 9.3.1 Watchdog timer's count source

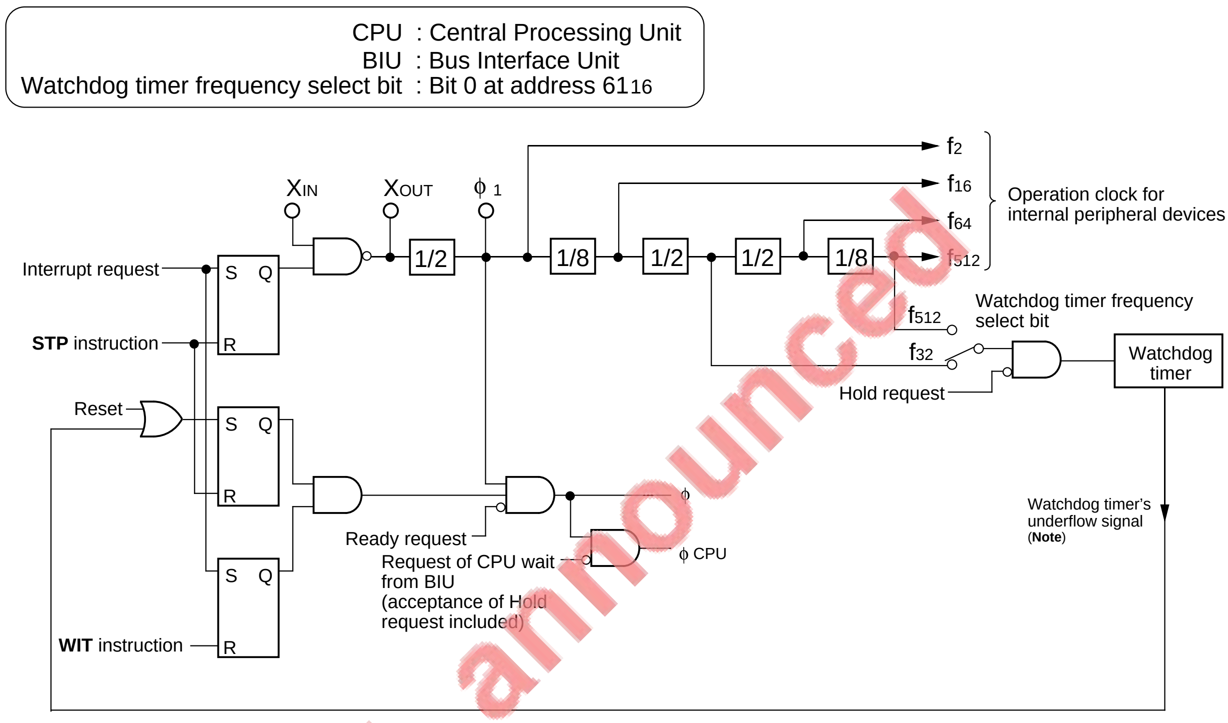
CHAPTER 10
STOP MODE

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.



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 to f_{512} 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, clock ϕ_1, f_2 to f_{512}		
	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 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 clock ϕ_1, f_2 to f_{512} starts.
② The watchdog timer starts counting owing to the oscillation start. The watchdog timer counts f_{32}.
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 f_{32} or f_{512} 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.

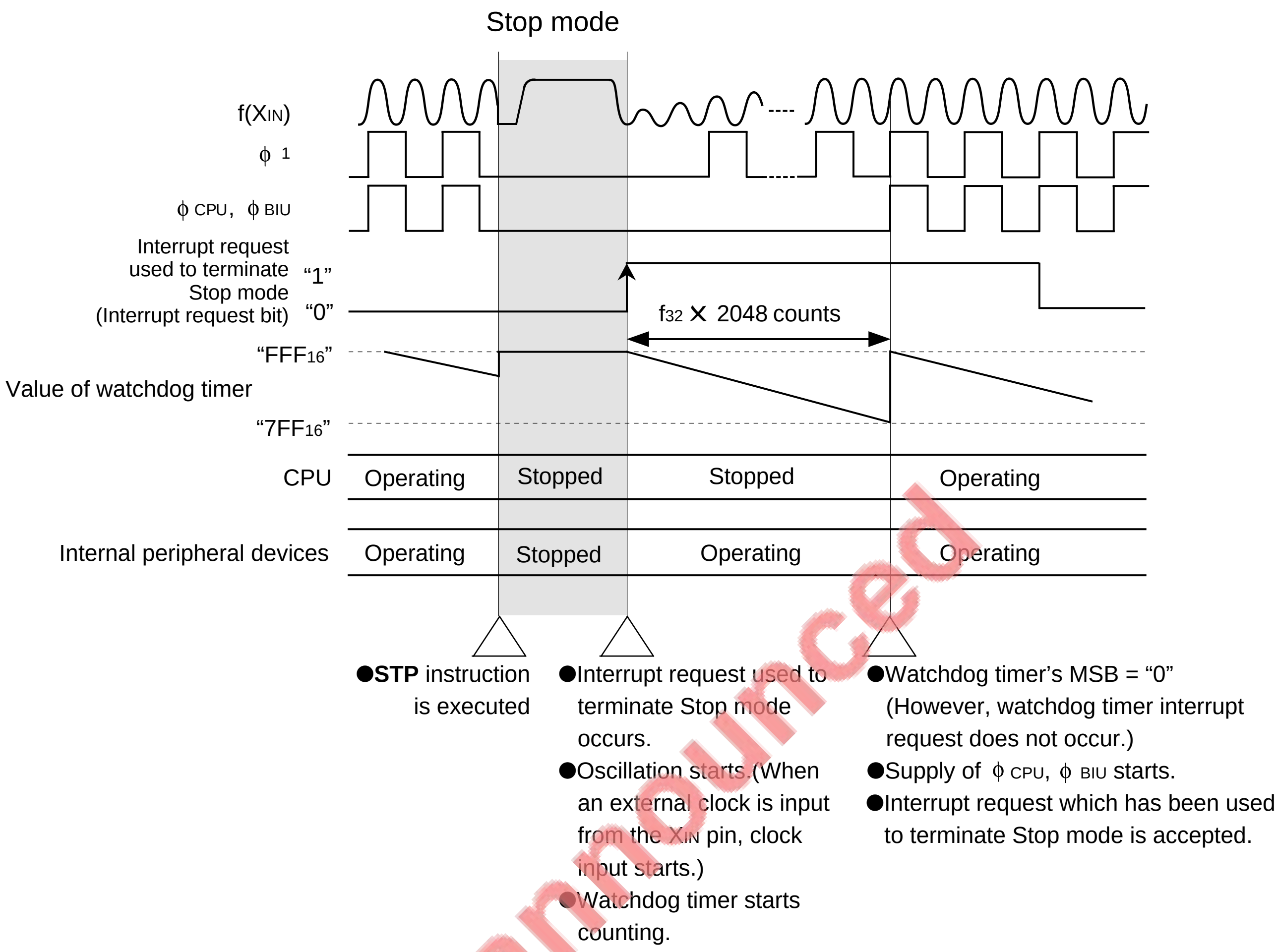


Fig. 10.2.1 Stop mode terminating sequence by interrupt request occurrence

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 a 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 external ROM 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

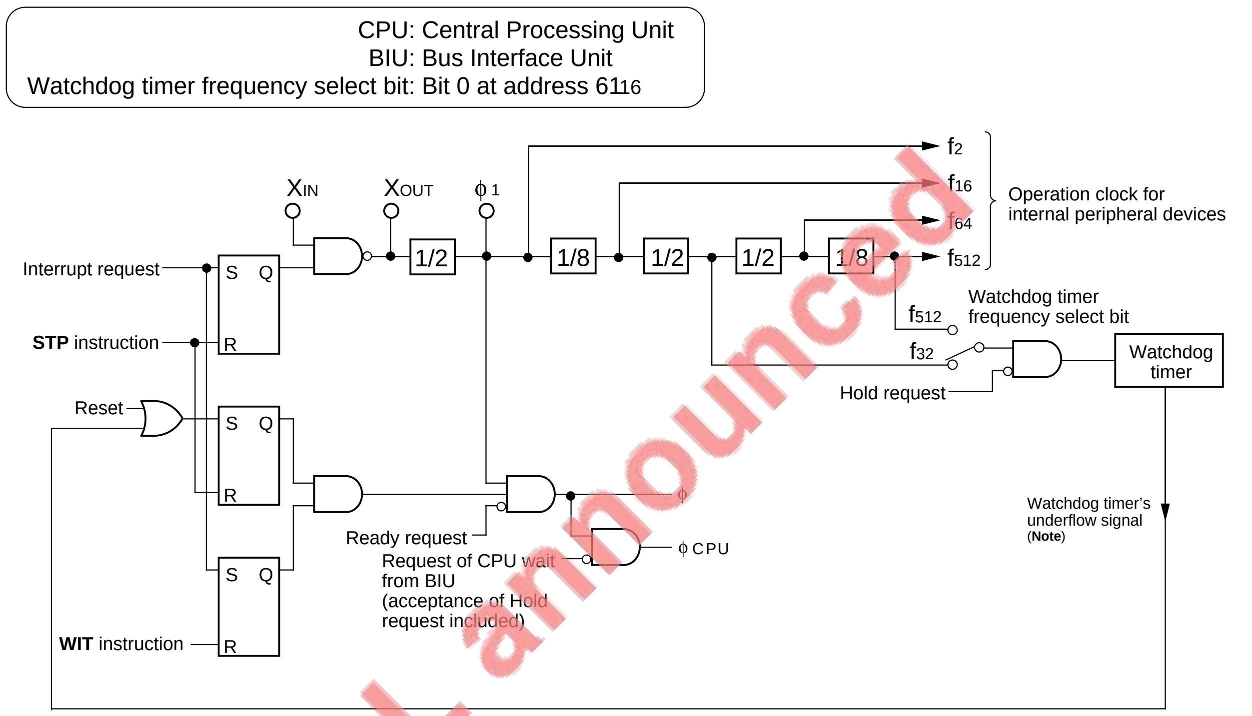
CHAPTER 11

WAIT MODE

This chapter describes Wait mode.
Wait mode is used to stop ϕ_{CPU} and ϕ when there is no need to operate the central processing unit (CPU). The oscillator continues its oscillation. 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.



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 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 is, power source voltage, is maintained.

Table 11.2.1 lists the microcomputers 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		Stopped
	Clock ϕ_1, f_2 to f_{512}		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 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 starts.
② The interrupt request which occurs in ① is accepted.
The following interrupts are used to terminate Wait mode.
The occurrence of the watchdog timer interrupt request also terminates Wait mode.

- $\overline{INT_i}$ 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 above multiple interrupts 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.

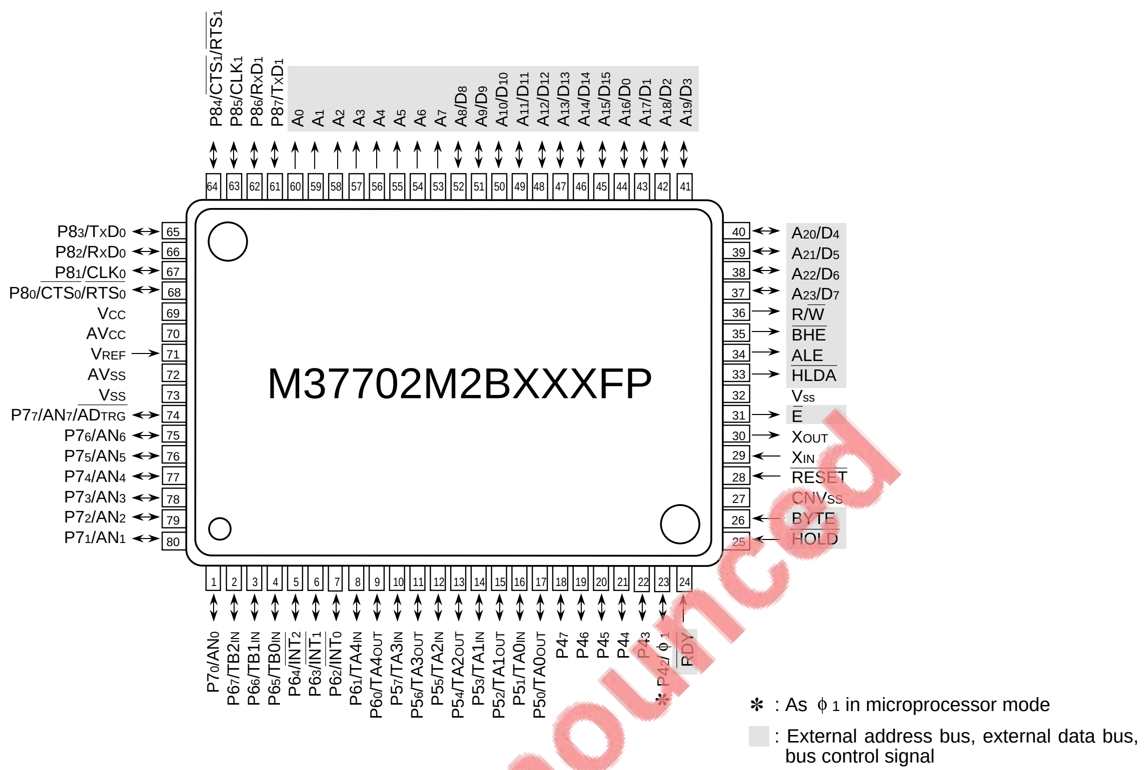
12.1 Signals required for accessing external devices

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 Software Wait," "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”)



●External data bus width = 8 bits (BYTE = “H”)

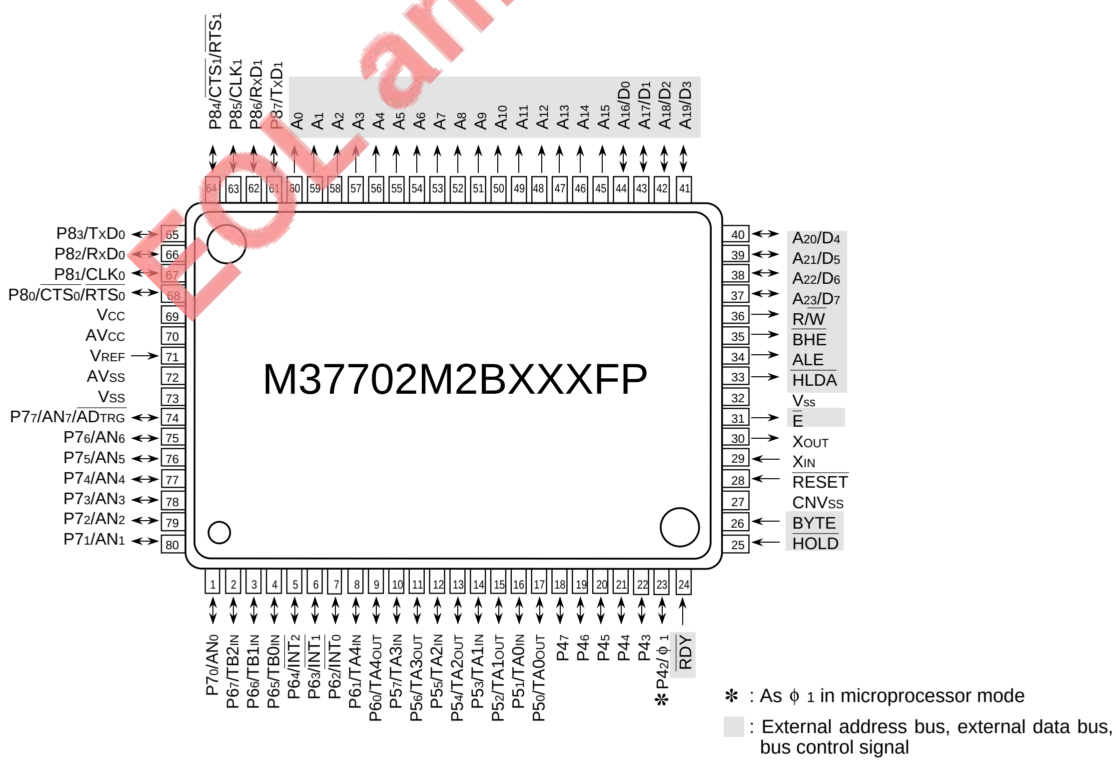


Fig. 12.1.1 Pin configuration in memory expansion and microprocessor modes (top view)

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 — A_0 (P0)	A_7 — A_0: A_7 — A_0	
A_{15}/D_{15} — 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 — 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/$\overline{W}$ ($P3_0$)	$\overline{HLDA}$: $\overline{HLDA}$ **(Note 1)** ALE: ALE $\overline{BHE}$: $\overline{BHE}$ R/$\overline{W}$: R/$\overline{W}$	
$P4_7$ — $P4_3$ 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 2)** $\overline{RDY}$: $\overline{RDY}$ **(Note 4)** $\overline{HOLD}$: $\overline{HOLD}$ **(Note 4)**	
$\overline{E}$	$\overline{E}$: $\overline{E}$	

Notes 1:The 7703 Group does not have the $\overline{HLDA}$ pin.

2: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.

3: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 Software Wait**"; "**12.3 Ready function**"; "**12.4 Hold function**"; "**Chapter 15. Electrical characteristics**."

4:Fix bits 0 and 1 of the Port P4 direction register to "0." Perform the setup regardless of whether using the $P4_0$/$\overline{HOLD}$ and $P4_1$/$\overline{RDY}$ pins as the $\overline{HOLD}$ or $\overline{RDY}$ pins or not. For the external ROM version, perform the same setup.

(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.

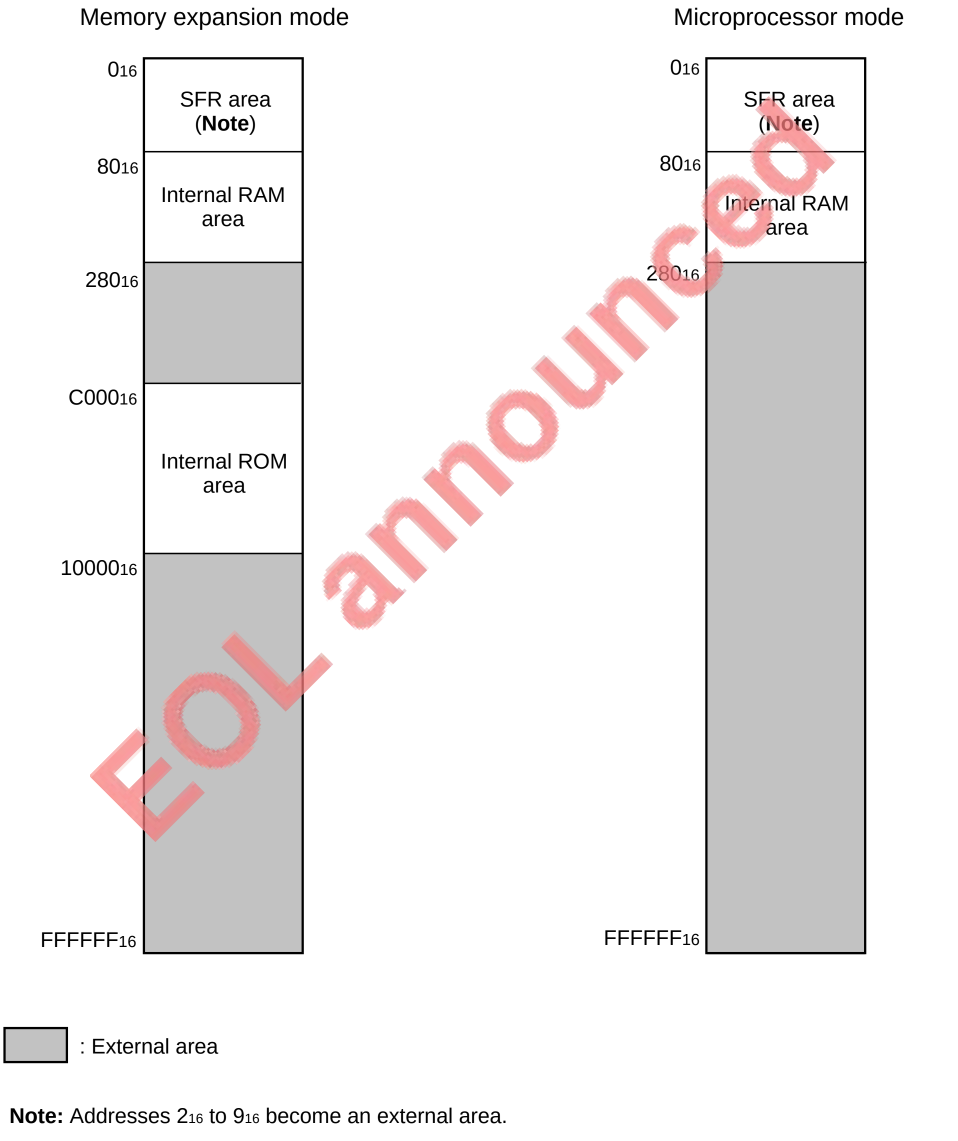


Fig. 12.1.2 External area

(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."**)

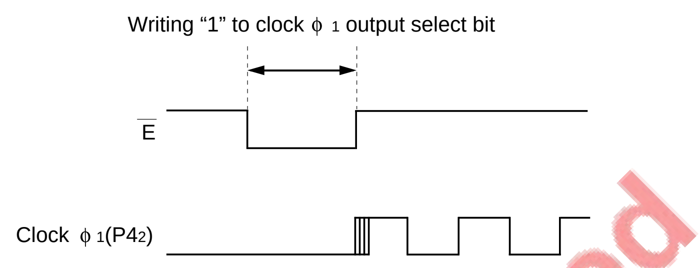
(9) Clock ϕ_1

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 P4₂ pin is outputting “L” level.

Fig. 12.1.3 Output start timing of clock ϕ_1

b7 b6 b5 b4 b3 b2 b1 b0 — bit 6: 0

Processor mode register (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	Wait bit	0 : Software Wait is inserted when accessing external area. 1 : No software Wait is inserted when accessing external area.	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” after a reset. (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 Signals required for accessing external devices

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.

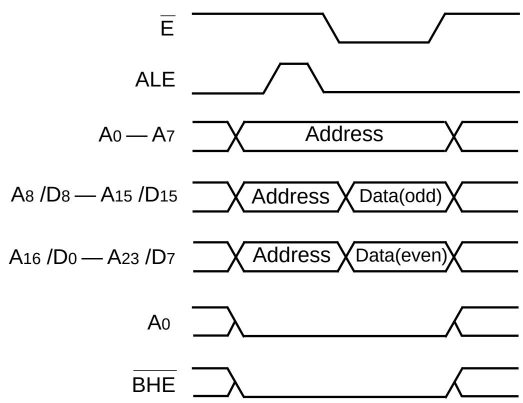
For instructions that are affected by the data length flag (m) and the index register length flag (x), operation ① or ② is applied when flag m or x = "0"; operation ③ or ④ is applied when flag m or x = "1."

The setup of flags m and x and the selection of the external data bus width do not affect each other.

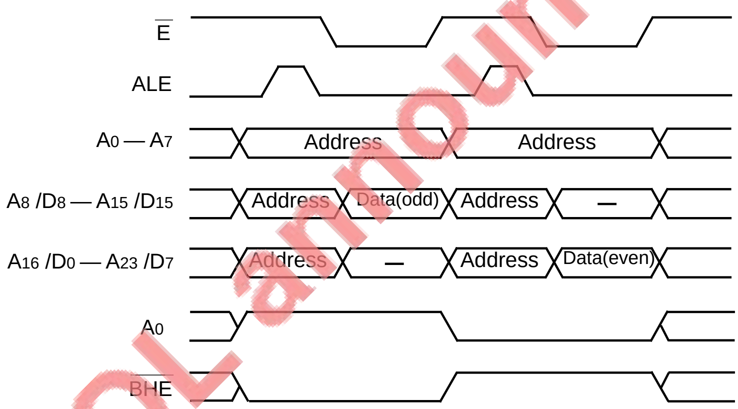
● External data bus width = 16 bits (BYTE = "L")

<16-bit data access>

(a) Access from even address

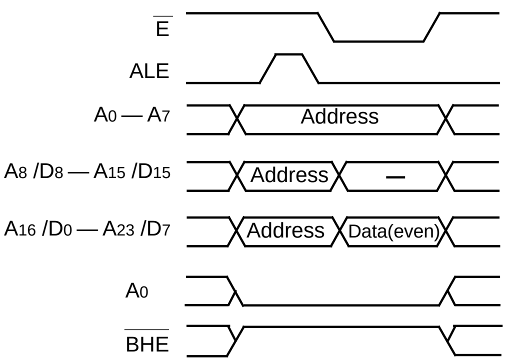


(b) Access from odd address



<8-bit data access>

(c) Access to even address



(d) Access to odd address

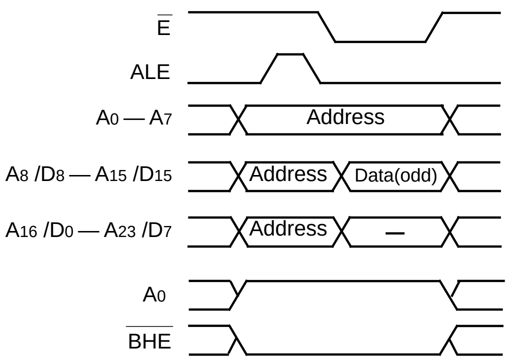
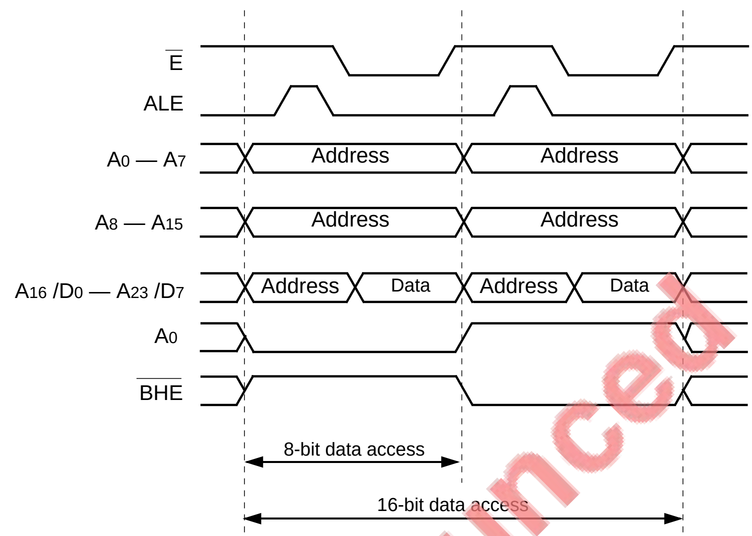


Fig. 12.1.5 Example of operating waveforms of signals input and output to/from externals (1)

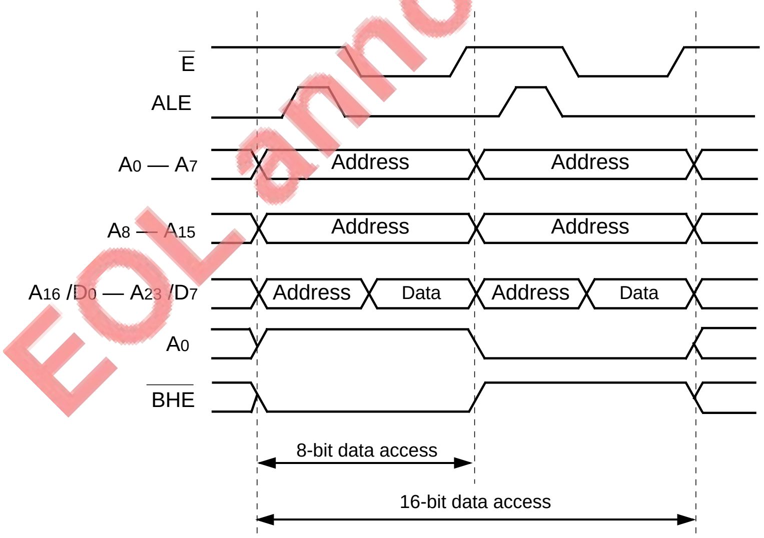
● External data bus width = 8 bits (BYTE = "H")

<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 Software Wait

Software Wait provides a function to facilitate access to external devices that require a long access time. To select the software Wait, use the wait bit (bit 2 at address $5E_{16}$). Figure 12.2.1 shows the structure of the processor mode register (address $5E_{16}$). Figure 12.2.2 shows an example of bus timing when the software Wait is used.

Software Wait is valid only for the external area. The internal areas is always accessed with no Wait.

b7	b6	b5	b4	b3	b2	b1	b0
	0						

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

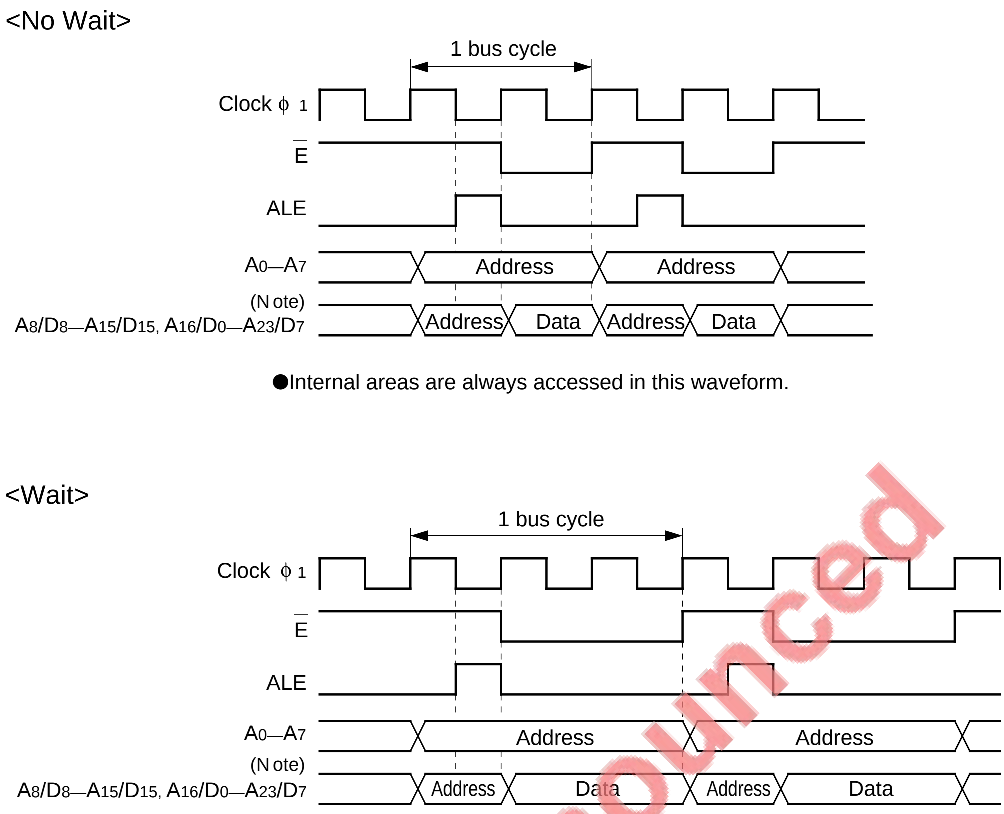
Bit	Bit name	Functions	At reset	RW
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Memory expansion mode 1 0 : Microprocessor mode 1 1 : Not selected	0	RW
1			0 **(Note1)**	RW
2	Wait bit	0 : Software Wait is inserted when accessing external area. 1 : No software Wait is inserted when accessing external area.	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, bit 1 becomes "1" after a reset. (Fixed to "1.")

2: Bit 7 is ignored in the microprocessor mode. (It may be either "0" or "1.")

(Shaded) : Bits 3 to 6 are not used when accessing the external area.

Fig. 12.2.1 Structure of processor mode register



Note: When the external data bus is 8 bits width (BYTE = "H"), A_8/D_8 to A_{15}/D_{15} operate with the same bus timing as A_0 to A_7.

Fig. 12.2.2 Example of bus timing when software Wait is used (BYTE = "L")

12.3 Ready function

Ready function provides a function to facilitate access to external devices that require a long access time. Fix bit 1 of the port P4 direction register to "0."
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.
Ready function is valid for the internal and external areas.

Table 12.3.1 Microcomputer's state in Ready state

Item	State
Oscillation	Operating
ϕ_{CPU}, ϕ	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, $\overline{E}$, $R/\overline{W}$, $\overline{BHE}$, $\overline{HLDA}$ **(Note 1)**, ALE	Retains the state when Ready request was accepted.
Pins $P4_3$ to $P4_7$, P5 to P8 **(Note 2)**	
$P4_2/\phi_1$	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 $5E_{16}$
Notes 1: The 7703 Group does not have the $\overline{HLDA}$ pin.
2: When this functions as a programmable I/O port.

12.3 Ready function

12.3.1 Operation description

The input level of the $\overline{\text{RDY}}$ pin is judged at the falling of the clock ϕ_1. 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 $\overline{\text{RDY}}$ 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 shows timing of acceptance of Ready request and termination of Ready state. Refer also to section "**17.1 Memory expansion**" about use of the Ready function.

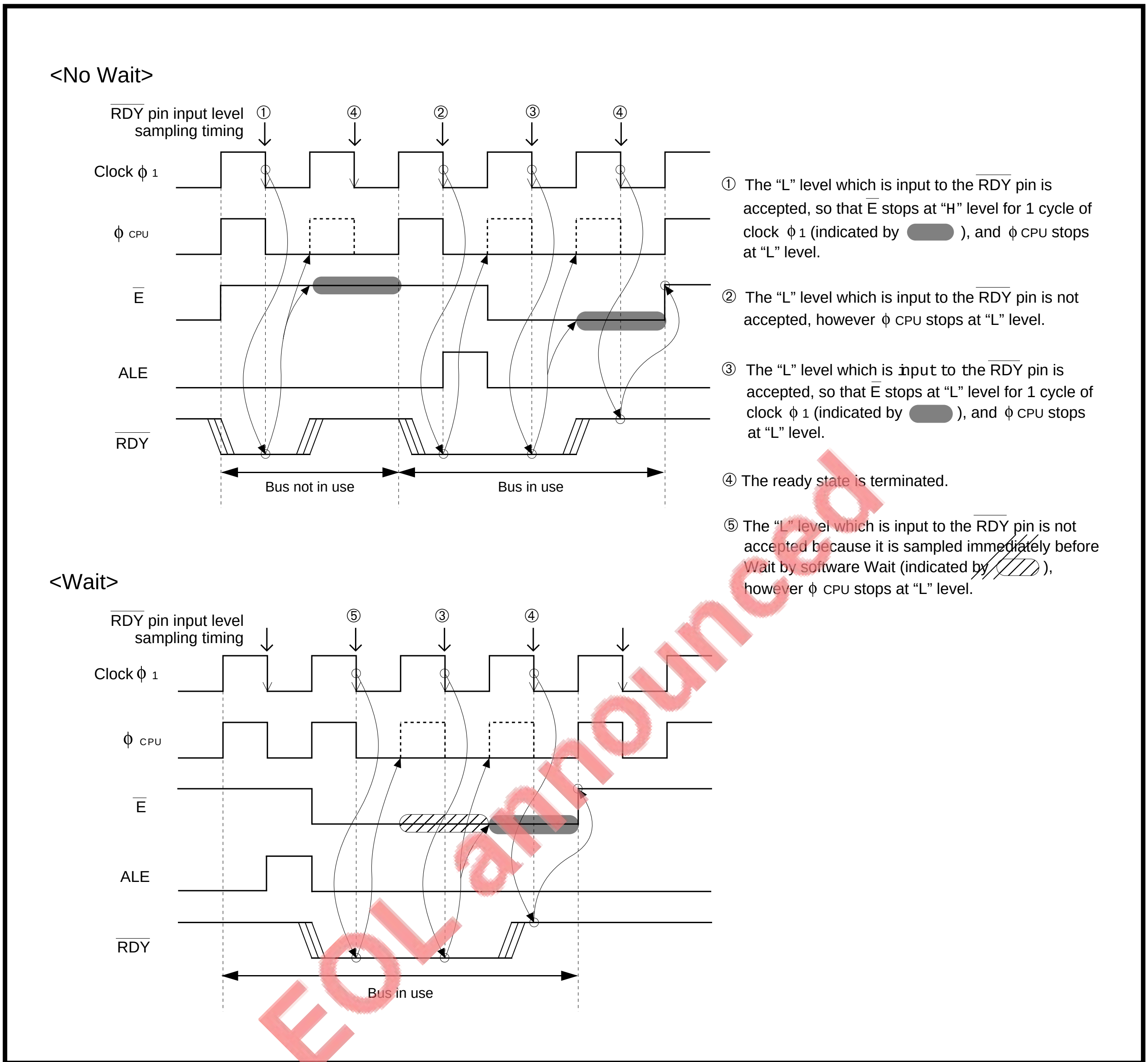


Fig. 12.3.1 Timings of acceptance of Ready request and termination of Ready state

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. Fix bit 0 of the port P4 direction register to "0."

In the memory expansion or microprocessor mode, the microcomputer enters Hold state by input of "L" level to the $\overline{\text{HOLD}}$ pin and retains this state while the level of the $\overline{\text{HOLD}}$ 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	Operating
ϕ_{CPU}	Stopped at "L"
$\overline{\text{E}}$	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/$\overline{\text{W}}$, $\overline{\text{BHE}}$	Floating
Pins $\overline{\text{HLDA}}$ **(Note 1)**, 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 2)**	Retains the state when Hold request was accepted.
Watchdog timer	Stopped

Clock ϕ_1 output select bit* : Bit 7 at address $5E_{16}$

Notes 1: The 7703 Group does not have the $\overline{\text{HLDA}}$ pin.

2: When this functions as a programmable I/O port.

12.4.1 Operation description

Judgment timing of the input level of the $\overline{HOLD}$ pin depends on the state using the bus. While the bus is not in use, the judgment is performed at every falling of ϕ. While the bus is in use, judgment is performed at the falling of the last ϕ in each bus cycle. Additionally, when accessing word data starting 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. At the same time, the $\overline{HLDA}$ pin's level changes "H" to "L". When 1 cycle of ϕ has passed after the level of $\overline{HLDA}$ pin becomes "L", pins R/$\overline{W}$, $\overline{BHE}$, and the external bus become floating state.

In Hold state, the input level of the $\overline{HOLD}$ pin is judged at every falling of ϕ. Then, when "H" level is detected, the $\overline{HLDA}$ pin's level changes "L" to "H" next rising of ϕ. When 1 cycle of ϕ has passed after the level of $\overline{HLDA}$ 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 has a same polarity and a same frequency as the clock ϕ_1. However, ϕ stops by the Ready request, or executing the **STP** or **WIT** instruction. Accordingly, judgment of the input level of the $\overline{HOLD}$ pin is not performed during Ready state.

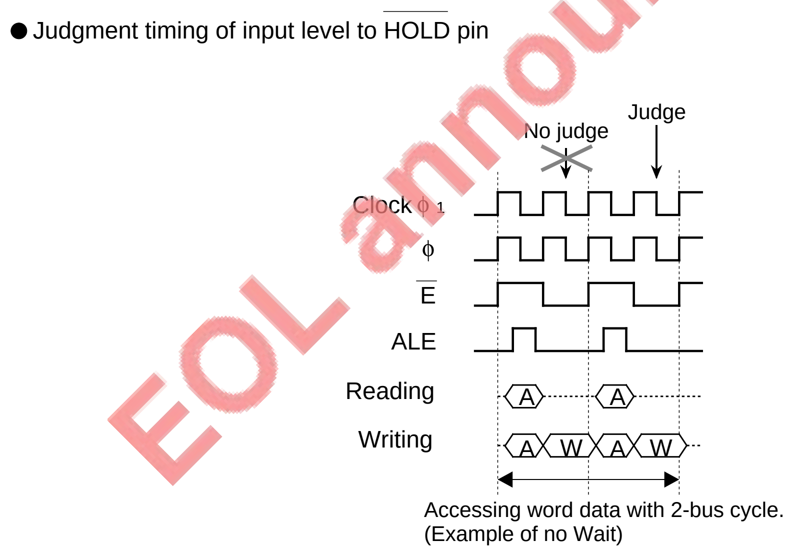
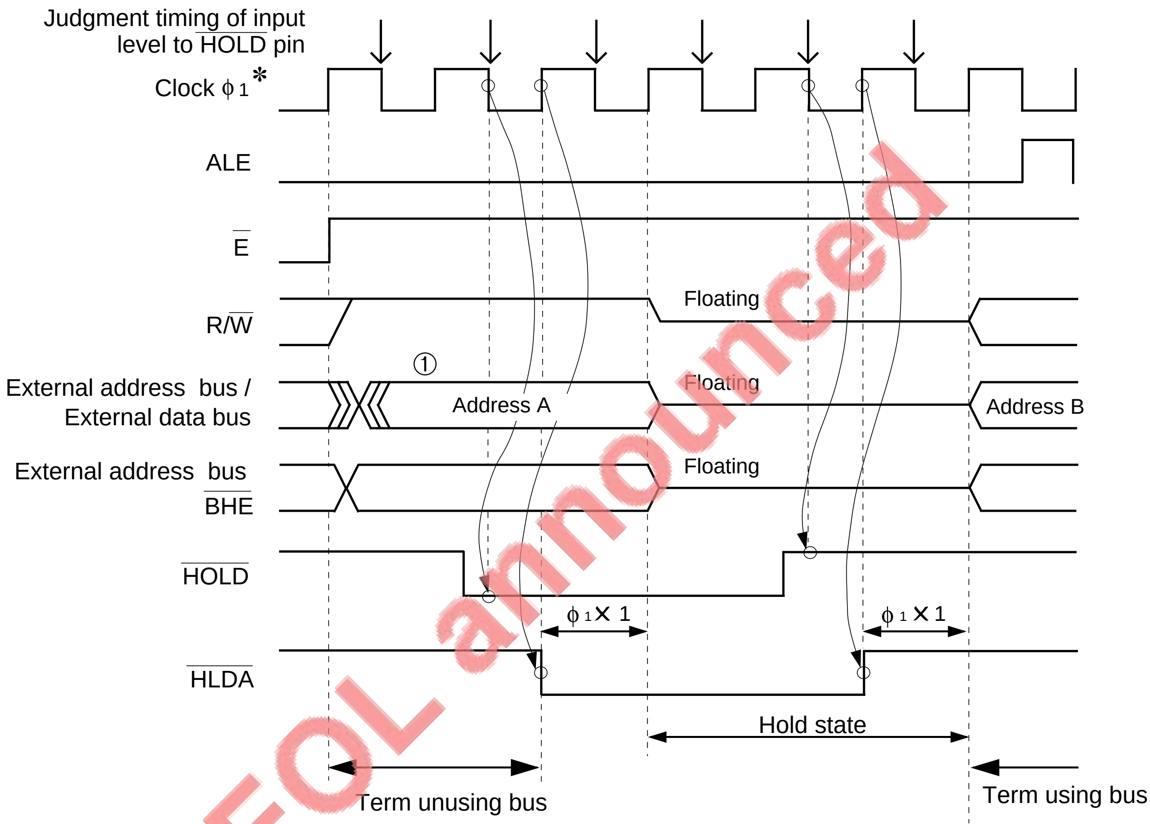


Fig. 12.4.1 Judgment when accessing word data beginning from odd address with 2-bus cycle

<When inputting “L” level to $\overline{\text{HOLD}}$ pin during term unusing bus>

● State when inputting “L” level to $\overline{\text{HOLD}}$ 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.

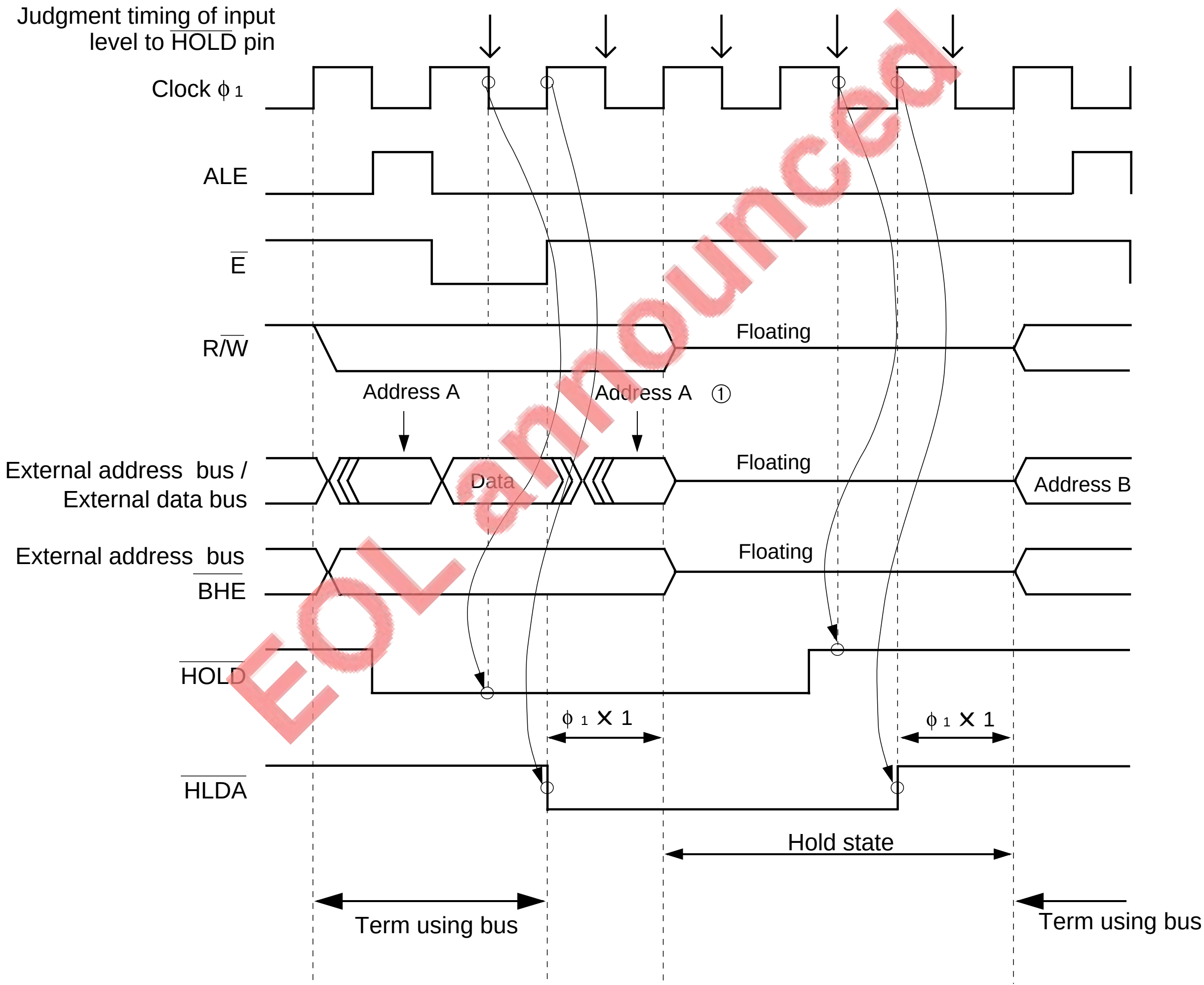
✻ Clock ϕ_1 has the same polarity and the same frequency as ϕ .
Signals timing to be input or output externally is ordained by clock ϕ_1 as a basis.

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 no Wait.

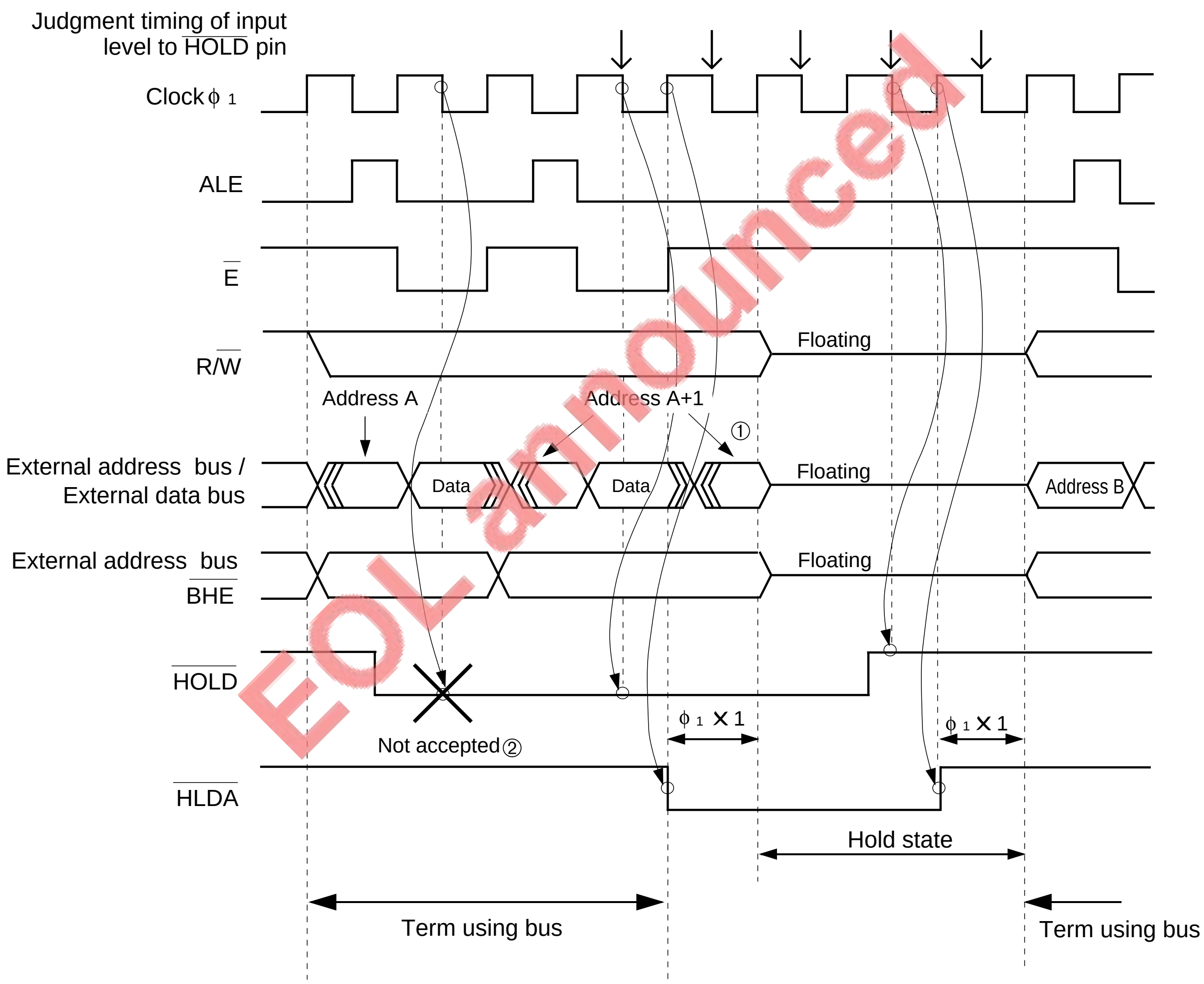
2: Clock φ1 has the same polarity and the same frequency as φ.
Signals timing to be input or output externally is ordained by clock φ1 as a basis.

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 no Wait.
2: Clock ϕ_1 has the same polarity and the same frequency as ϕ .
Signals timing to be input or output externally is ordained by clock ϕ_1 as a basis.

Fig. 12.4.4 Timing of acceptance of Hold request and termination of Hold state (3)

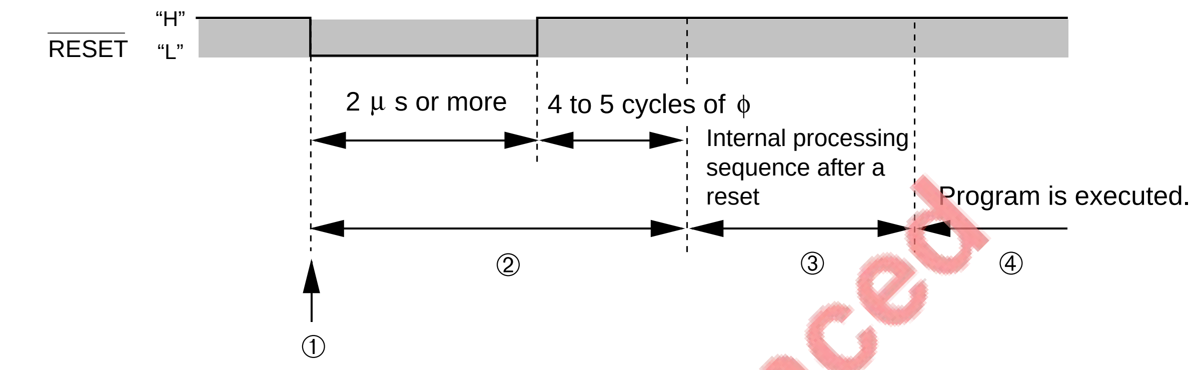
CHAPTER 13

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 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 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 RESET pin, the microcomputer initializes pins within a term of several ten ns. (Refer to Table 13.1.1.)

② While the RESET pin is "L" level and within the term of 4 to 5 cycles of the internal clock ϕ after the 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 $\overline{\text{RESET}}$ pin is "L" level.

Table 13.1.1 Pin state while $\overline{\text{RESET}}$ pin is "L" level

	Identification*	CNVss pin level	Pin (Port) name	Pin state
Mask ROM version	M6 M8	Vss or Vcc	P0 to P8	Floating.
			$\overline{E}$	Outputs "H" level.
	M3 MD	Vss	P0 to P8	Floating.
			$\overline{E}$	Outputs "H" level.
	M2 M4	Vss	P0 to P8	Floating.
			$\overline{E}$	Outputs "H" level.
		Vcc	P0, P1, P2, $P3_1$	Outputs "H" or "L" level.
			$P3_0$, $P3_3$, $\overline{E}$	Outputs "H" level.
			$P3_2$	Outputs "L" level.
			$P4_2$	Outputs ϕ_1.
			$P4_0$, $P4_1$, $P4_3$–$P4_7$, P5 to P8	Floating.
External ROM version	S1 S4	Vcc	A_0–A_7, A_8/D_8–A_{15}/D_{15}, A_{16}/D_0–A_{23}/D_7, $\overline{\text{BHE}}$	Outputs "H" or "L" level.
			R/$\overline{\text{W}}$, $\overline{\text{HLDA}}$, $\overline{E}$	Outputs "H" level.
			ALE	Outputs "L" level.
			ϕ_1	Outputs ϕ_1.
			$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$, $P4_3$–$P4_7$, P5 to P8	Floating.
PROM version (Including One time PROM and EPROM versions)		Vss	P0 to P8	Floating.
			$\overline{E}$	Outputs "H" level.
		Vcc (**Note**)	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.

Identification* : This expresses the internal memory type and its size identification. Refer to "**Chapter 1. DESCRIPTION**."

Note: Each pin becomes the above state. It is because the microcomputer enters the EPROM mode. Refer to "**Chapter 19. PROM 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 area immediately after reset.

0 : "0" immediately after a reset.
1 : "1" immediately after a reset.
? : Undefined immediately after a reset.

(shaded) : Always "0" at reading.

Register name	State immediately after a reset: b15 – b8	State immediately after a reset: 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
(shaded)	(shaded)	(shaded)	(shaded)	(shaded)	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 hatched] : "0" immediately after a reset. Fix to "0."

Address	Register name	Access characteristics (b7 … b0)	State immediately after a reset (b7 … b0)	
0_{16}		(nothing assigned)	?	
1_{16}		(nothing assigned)	?	
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	b7–b4: nothing assigned; b3–b0: RW	0 0 0 0 ?	
8_{16}	Port P2 direction register	RW	00_{16}	
9_{16}	Port P3 direction register	b7–b4: nothing assigned; b3–b0: RW	0 0 0 0 0 0 0 0	(Note)
A_{16}	Port P4 register	RW	?	
B_{16}	Port P5 register	RW	?	
C_{16}	Port P4 direction register	RW	00_{16}	(Note)
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}	(Note)
11_{16}	Port P7 direction register	RW	00_{16}	(Note)
12_{16}	Port P8 register	RW	?	
13_{16}		(nothing assigned)	?	
14_{16}	Port P8 direction register	RW	00_{16}	(Note)
15_{16}		(nothing assigned)	?	
16_{16}		(nothing assigned)	?	
17_{16}		(nothing assigned)	?	
18_{16}		(nothing assigned)	?	
19_{16}		(nothing assigned)	?	
$1A_{16}$		(nothing assigned)	?	
$1B_{16}$		(nothing assigned)	?	
$1C_{16}$		(nothing assigned)	?	
$1D_{16}$		(nothing assigned)	?	
$1E_{16}$	A-D control register	RW	0 0 0 0 0 ?	
$1F_{16}$	A-D sweep pin select register	b7–b2: nothing assigned; b1–b0: RW	? ? ? ? ? ? 1 1	

Note : In the 7703 Group, after a reset, set "1" to the bits which do not have corresponding pins. (Refer to section "**20.4.1 I/O pin.**")

Fig. 13.1.3 State of SFR and internal RAM areas immediately after reset (1)

RESET

13.1 Hardware reset

Address	Register name	Access characteristics (b7 … b0)	State immediately after a reset (b7 … b0)
20_{16}	A-D register 0	RO	?
21_{16}			?
22_{16}	A-D register 1	RO	?
23_{16}			?
24_{16}	A-D register 2	RO	?
25_{16}			?
26_{16}	A-D register 3	RO	?
27_{16}			?
28_{16}	A-D register 4	RO	?
29_{16}			?
$2A_{16}$	A-D register 5	RO	?
$2B_{16}$			?
$2C_{16}$	A-D register 6	RO	?
$2D_{16}$			?
$2E_{16}$	A-D register 7	RO	?
$2F_{16}$			?
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}		b0: WO	?
34_{16}	UART0 transmit/receive control register 0	b3: RO; b2–b0: RW	? ? ? ? 1 0 0 0
35_{16}	UART0 transmit/receive control register 1	b7–b3: RO; b2: RW; b1: RO; b0: RW	0 0 0 0 0 0 1 0
36_{16}	UART0 receive buffer register	RO	?
37_{16}		b0: 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	?
$3B_{16}$		b0: WO	?
$3C_{16}$	UART1 transmit/receive control register 0	b3: RO; b2–b0: RW	? ? ? ? 1 0 0 0
$3D_{16}$	UART1 transmit/receive control register 1	b7–b3: RO; b2: RW; b1: RO; b0: RW	0 0 0 0 0 0 1 0
$3E_{16}$	UART1 receive buffer register	RO	?
$3F_{16}$		b0: 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 b7							Access b0	State b7							State b0
40_{16}	Count start register	RW								00_{16}							
41_{16}										?							
42_{16}	One-shot start register				WO					?			0	0	0	0	0
43_{16}										?							
44_{16}	Up-down register	WO			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	RW		(Note 3)		RW				0	0	?	?	0	0	0	0
$5C_{16}$	Timer B1 mode register	RW		(Note 3)		RW				0	0	?	?	0	0	0	0
$5D_{16}$	Timer B2 mode register	RW		(Note 3)		RW				0	0	?	?	0	0	0	0
$5E_{16}$	Processor mode register	RW				WO	RW	RW (Note4)	RW	0	0	0	0	0	0	(Note 4)	0
$5F_{16}$										?							

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	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}, the 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 $27F_{16}$ in M37702M2BXXXFP)

- •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.

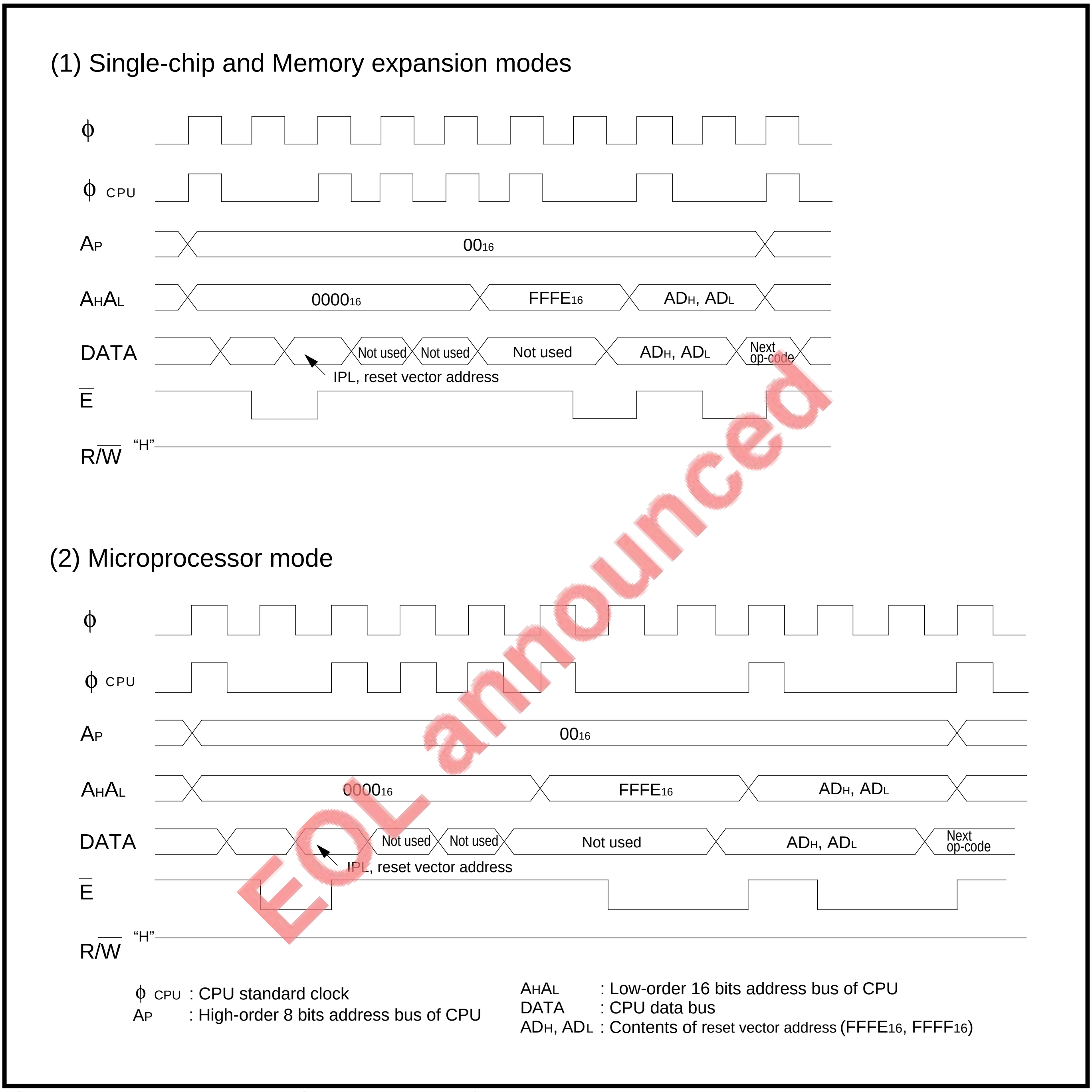


Fig. 13.1.7 Internal processing sequence after reset

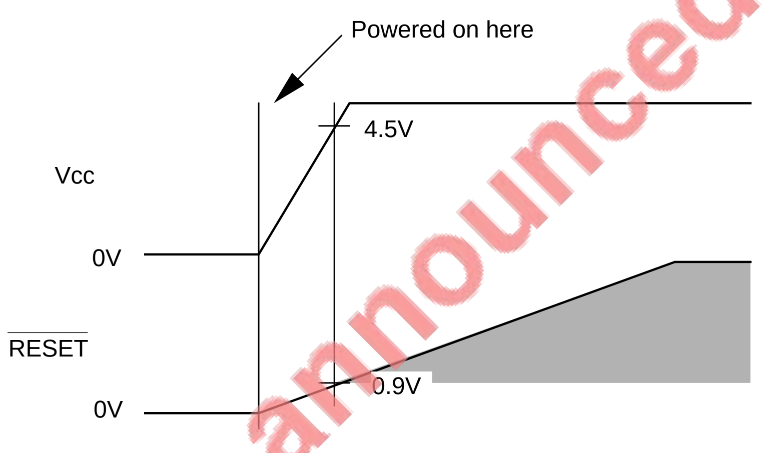
13.1.4 Time supplying "L" level to $\overline{\text{RESET}}$ pin

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.

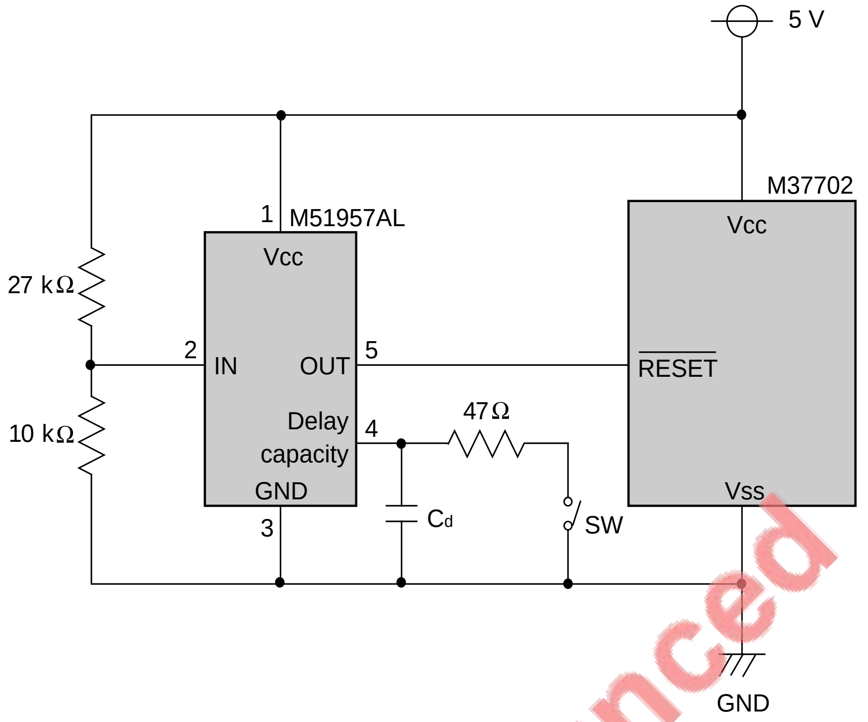
● If the oscillator is not stably oscillating (including a power-on reset and In Stop mode), supply "L" level until the oscillation is stabilized.
The time to stabilize oscillation varies according to the oscillator. For details, contact the oscillator manufacturer.
Figure 13.1.8 shows the power-on reset 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."**



Note : Refer to "**Figure 18.3.1 Power-on reset conditions**" for the low supply voltage version.

Fig. 13.1.8 Power-on reset condition



✻ The delay time is about 11 ms when C_d = 0.033 µF.

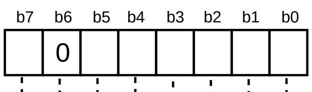
$t_d \approx 0.34 \times C_d$ [µs], C_d: [pF]

Note : Refer to "**Figure 18.3.2 Example of power-on reset circuit**" for the low supply voltage version.

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. 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 $FFFF_{16}$.



Processor mode register (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	Wait bit	0 : Software Wait is inserted when accessing external area. 1 : No software Wait is inserted when accessing external area.	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" after a reset. (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 4 to 7 are not used at software reset.

Fig. 13.2.1 Structure of processor mode register

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.

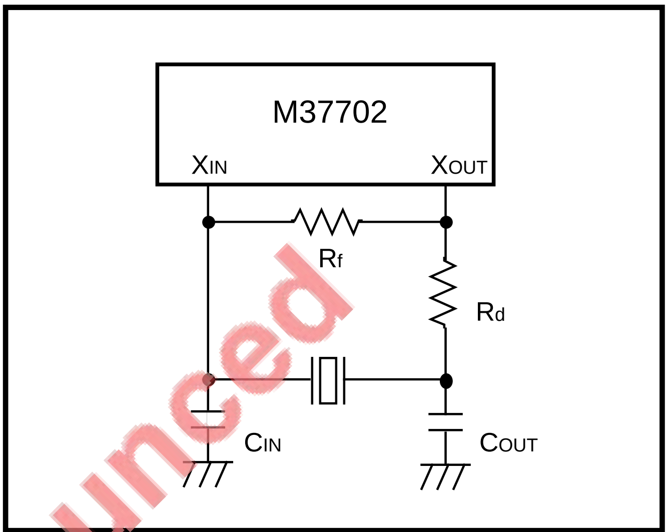


Fig. 14.1.1 Connection example using resonator/oscillator

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.

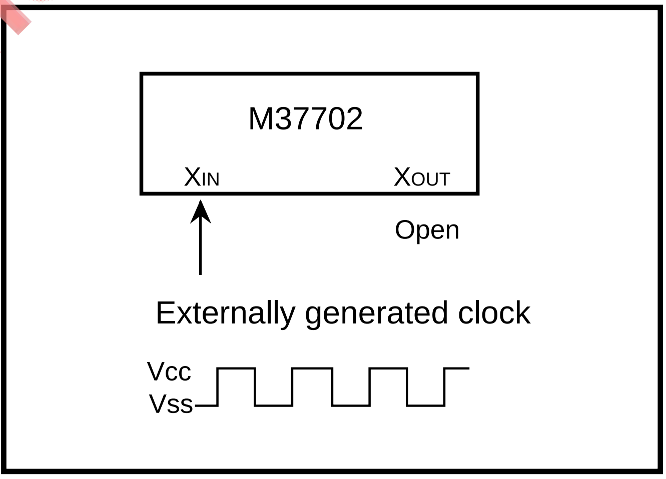
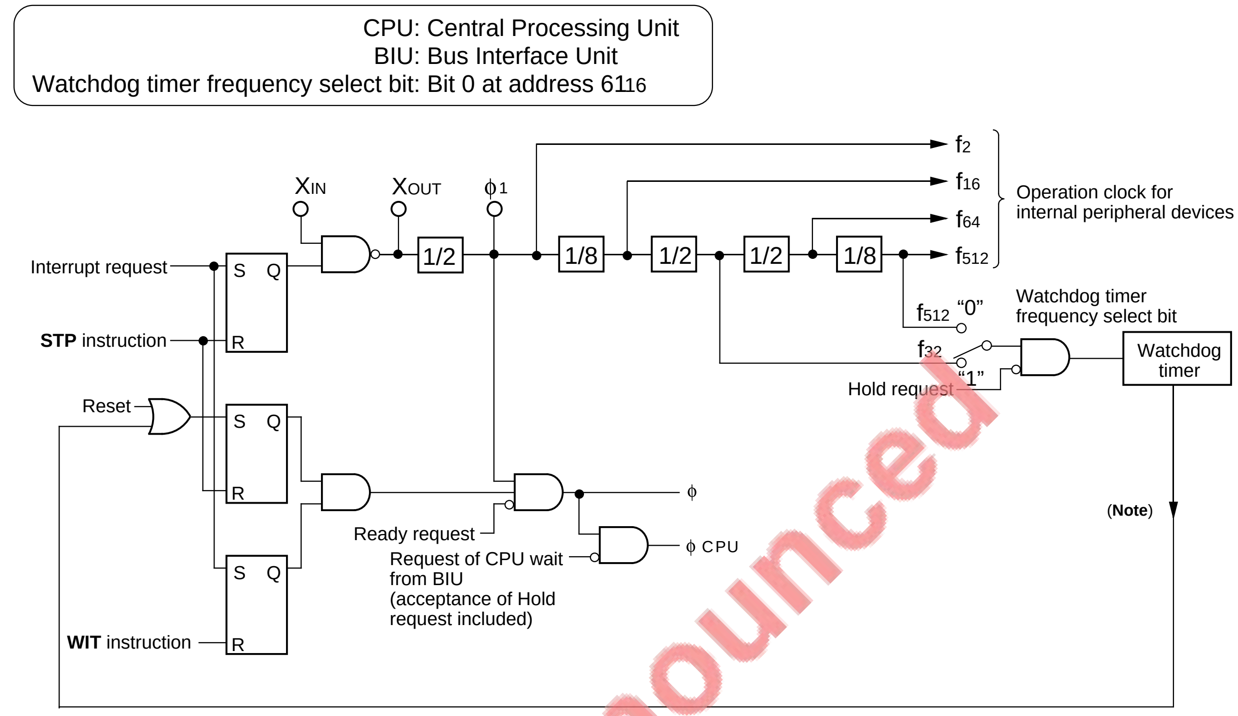


Fig. 14.1.2 Externally generated clock input example

14.2 Clock

Figure 14.2.1 shows the clock generating circuit block diagram.



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 Clock

14.2.1 Clock generated in clock generating circuit

(1) ϕ

It is the operation clock of BIU. It is also the clock source of ϕ_{CPU}.
The ϕ stops by Ready request or execution of the **STP** or **WIT** instruction. It is not stopped by acceptance of Hold request.

(2) ϕ_{CPU}

It is the operation clock of CPU. The ϕ_{CPU} stops by the following:
•Execution of the **STP** or **WIT** instruction,
•Ready request; “L” level input to $\overline{\text{RDY}}$ pin
•Wait request from BIU; Hold request acceptance included

(3) Clock ϕ_1

It has the same period as ϕ and is output to the external from the ϕ_1 pin. The clock ϕ_1 stops by execution of the **STP** instruction.
It is not stopped by Ready request or acceptance of Hold request, or execution of the **WIT** instruction.

(4) f_2 to f_{512}

Each of them is the internal peripheral devices’ operating clock.

Note: Refer to each functional description for details:
•Execution of **STP** instruction .... “**Chapter 10. STOP MODE**”
•Execution of **WIT** instruction .... “**Chapter 11. WAIT MODE**”
•Ready .... “**Paragraph. 12.3 Ready function**”
•Hold .... “**Paragraph. 12.4 Hold function**”

CHAPTER 15
ELECTRICAL CHARACTERISTICS

ELECTRICAL CHARACTERISTICS

15.1 Absolute maximum ratings

This chapter describes electrical characteristics of the M37702M2BXXXFP and M37702M2AXXXFP.
For the low voltage version, refer to section "**18.4 Electrical characteristics.**"
The 7703 Group's available pins varies from that of the 7702 Group. Refer to **"Chapter 20. 7703 GROUP."**
For the latest data, inquire of addresses described last (☞**"CONTACT ADDRESSES FOR FURTHER INFORMATION")**.

In a part of the standard indicated in this chapter, there are the limits depending on each microcomputer product or used external clock input frequency. Distinguish it described below.

•Limits depending on each microcomputer
(Example) M37702M2BXXXFP
When this sign is 'A,' refer to the column of "16 MHz."
When this sign is 'B,' refer to the column of "25 MHz."

•Limits depending on used external clock input frequency
The calculation formula is described in the table. When the microcomputer is 16 MHz version, the limits is the value in the case of $f(X_{IN})$ = 16 MHz. When the microcomputer is 25 MHz version, the limits is the value in the case of $f(X_{IN})$ = 25 MHz.

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 **(Note)**	mW
T_{opr}	Operating temperature		–20 to 85	°C
T_{stg}	Storage temperature		–40 to 150	°C

Note: In the 7703 Group, this value is 1000 mW.

15.2 Recommended operating conditions

Recommended operating conditions (V_{CC} = 5 V±10%, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits Min.	Limits Typ.	Limits Max.	Unit
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	0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage	P1$_0$–P1$_7$, P2$_0$–P2$_7$ (in single-chip mode)	0.8V_{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)	0.5V_{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		0.2V_{CC}	V
V_{IL}	Low-level input voltage	P1$_0$–P1$_7$, P2$_0$–P2$_7$ (in single-chip mode)	0		0.2V_{CC}	V
V_{IL}	Low-level input voltage	P1$_0$–P1$_7$, P2$_0$–P2$_7$ (in memory expansion mode and microprocessor mode)	0		0.16V_{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$_3$, 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$_3$, P5$_4$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_7$			5	mA
$f(X_{IN})$	External clock input frequency	M37702M2BXXXFP			25	MHz
		M37702M2AXXXFP			15	

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, unless otherwise noted)

Symbol	Parameter		Test conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
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$_{IN}$, $\overline{INT_0}$–$\overline{INT_2}$, 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}) = 25 MHz		19	38	mA
				f(X_{IN}) = 16 MHz		12	24	
				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			Unit
			Min.	Typ.	Max.	
—	Resolution	V_{REF} = V_{CC}			8	Bits
—	Absolute accuracy	V_{REF} = V_{CC}			±3	LSB
R_{LADDER}	Ladder resistance	V_{REF} = V_{CC}	2		10	kΩ
t_{CONV}	Conversion time	f(X_{IN}) = 25 MHz	9.12			μs
		f(X_{IN}) = 16 MHz	14.25			
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 16 MHz Min.	Limits 16 MHz Max.	Limits 25 MHz Min.	Limits 25 MHz Max.	Unit
$t_{c(TA)}$	TAi_{IN} input cycle time	125		80		ns
$t_{w(TAH)}$	TAi_{IN} input high-level pulse width	62		40		ns
$t_{w(TAL)}$	TAi_{IN} input low-level pulse width	62		40		ns

Timer A input (gating input in timer mode)

Symbol	Parameter	Data formula	Limits 16 MHz Min.	Limits 16 MHz Max.	Limits 25 MHz Min.	Limits 25 MHz Max.	Unit
$t_{c(TA)}$	TAi_{IN} input cycle time	$\frac{8 \times 10^9}{f(X_{IN})}$	500		320		ns
$t_{w(TAH)}$	TAi_{IN} input high-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})}$	250		160		ns
$t_{w(TAL)}$	TAi_{IN} input low-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})}$	250		160		ns

Note: 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.

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

Symbol	Parameter	Data formula	Limits 16 MHz Min.	Limits 16 MHz Max.	Limits 25 MHz Min.	Limits 25 MHz Max.	Unit
$t_{c(TA)}$	TAi_{IN} input cycle time	$\frac{4 \times 10^9}{f(X_{IN})}$	250		160		ns
$t_{w(TAH)}$	TAi_{IN} input high-level pulse width		150		80		ns
$t_{w(TAL)}$	TAi_{IN} input low-level pulse width		150		80		ns

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

Symbol	Parameter	Limits 16 MHz Min.	Limits 16 MHz Max.	Limits 25 MHz Min.	Limits 25 MHz Max.	Unit
$t_{w(TAH)}$	TAi_{IN} input high-level pulse width	125		80		ns
$t_{w(TAL)}$	TAi_{IN} input low-level pulse width	125		80		ns

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

Symbol	Parameter	Limits 16 MHz Min.	Limits 16 MHz Max.	Limits 25 MHz Min.	Limits 25 MHz Max.	Unit
$t_{c(UP)}$	TAi_{OUT} input cycle time	2500		2000		ns
$t_{w(UPH)}$	TAi_{OUT} input high-level pulse width	1250		1000		ns
$t_{w(UPL)}$	TAi_{OUT} input low-level pulse width	1250		1000		ns
$t_{su(UP-T_{IN})}$	TAi_{OUT} input setup time	500		400		ns
$t_{h(T_{IN}-UP)}$	TAi_{OUT} input hold time	500		400		ns

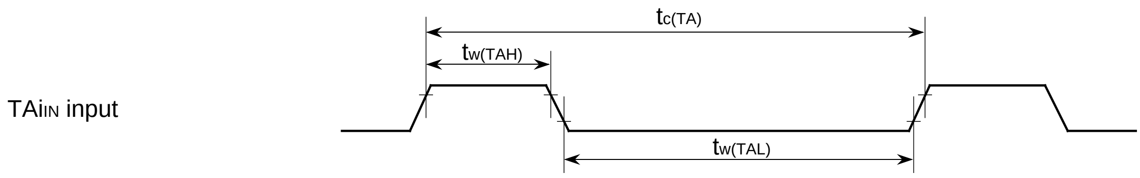
Timer A input (Two-phase pulse input in event counter mode)

Symbol	Parameter	Limits				Unit
		16 MHz		25 MHz		
		Min.	Max.	Min.	Max.	
$t_{c(TA)}$	TAj_{IN} input cycle time	1000		800		ns
$t_{su(TAj_{IN}-TAj_{OUT})}$	TAj_{IN} input setup time	250		200		ns
$t_{su(TAj_{OUT}-TAj_{IN})}$	TAj_{OUT} input setup time	250		200		ns

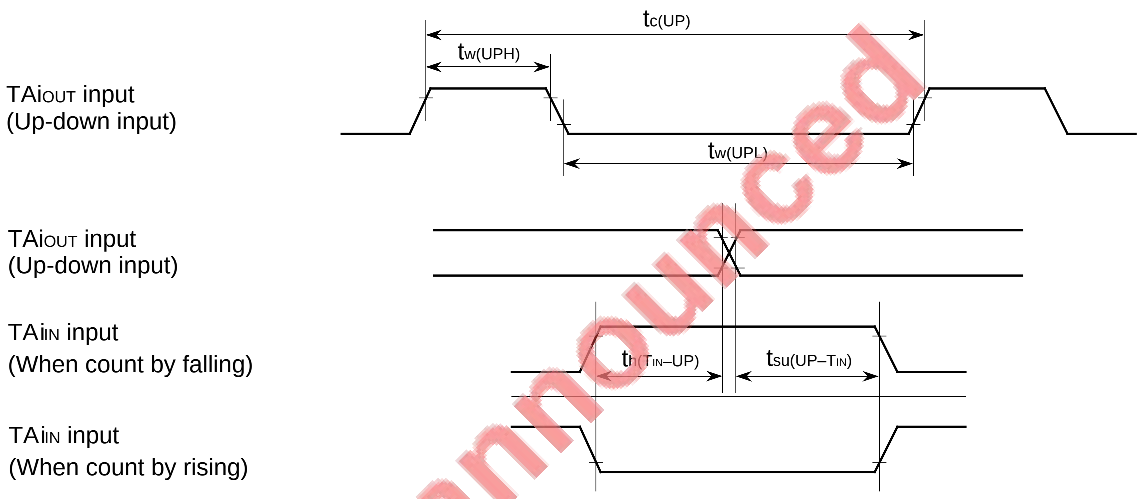
15.5 Internal peripheral devices

Internal peripheral devices

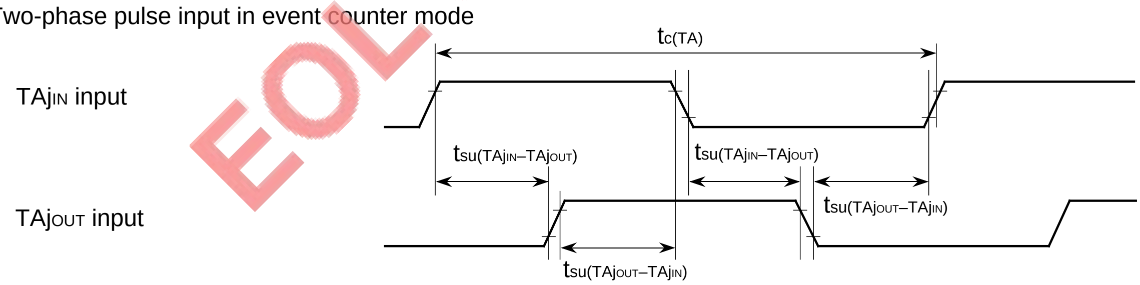
- Count input in event counter mode
- Gating input in timer mode
- External trigger input in one-shot pulse mode
- External trigger input in pulse width modulation mode



- Up-down input, 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 16 MHz Min.	Limits 16 MHz Max.	Limits 25 MHz Min.	Limits 25 MHz Max.	Unit
$t_{c(TB)}$	TBi_{IN} input cycle time (one edge count)	125		80		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width (one edge count)	62		40		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width (one edge count)	62		40		ns
$t_{c(TB)}$	TBi_{IN} input cycle time (both edges count)	250		160		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width (both edges count)	125		80		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width (both edges count)	125		80		ns

Timer B input (pulse period measurement mode)

Symbol	Parameter	Data formula	Limits 16 MHz Min.	Limits 16 MHz Max.	Limits 25 MHz Min.	Limits 25 MHz Max.	Unit
$t_{c(TB)}$	TBi_{IN} input cycle time	$\frac{8 \times 10^9}{f(X_{IN})}$	500		320		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})}$	250		160		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})}$	250		160		ns

Note: 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.

Timer B input (pulse width measurement mode)

Symbol	Parameter	Data formula	Limits 16 MHz Min.	Limits 16 MHz Max.	Limits 25 MHz Min.	Limits 25 MHz Max.	Unit
$t_{c(TB)}$	TBi_{IN} input cycle time	$\frac{8 \times 10^9}{f(X_{IN})}$	500		320		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})}$	250		160		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})}$	250		160		ns

Note: 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.

A-D trigger input

Symbol	Parameter	Limits 16 MHz Min.	Limits 16 MHz Max.	Limits 25 MHz Min.	Limits 25 MHz Max.	Unit
$t_{c(AD)}$	$\overline{AD_{TRG}}$ input cycle time (minimum allowable trigger)	1000		1000		ns
$t_{w(ADL)}$	$\overline{AD_{TRG}}$ input low-level pulse width	125		125		ns

ELECTRICAL CHARACTERISTICS

15.5 Internal peripheral devices

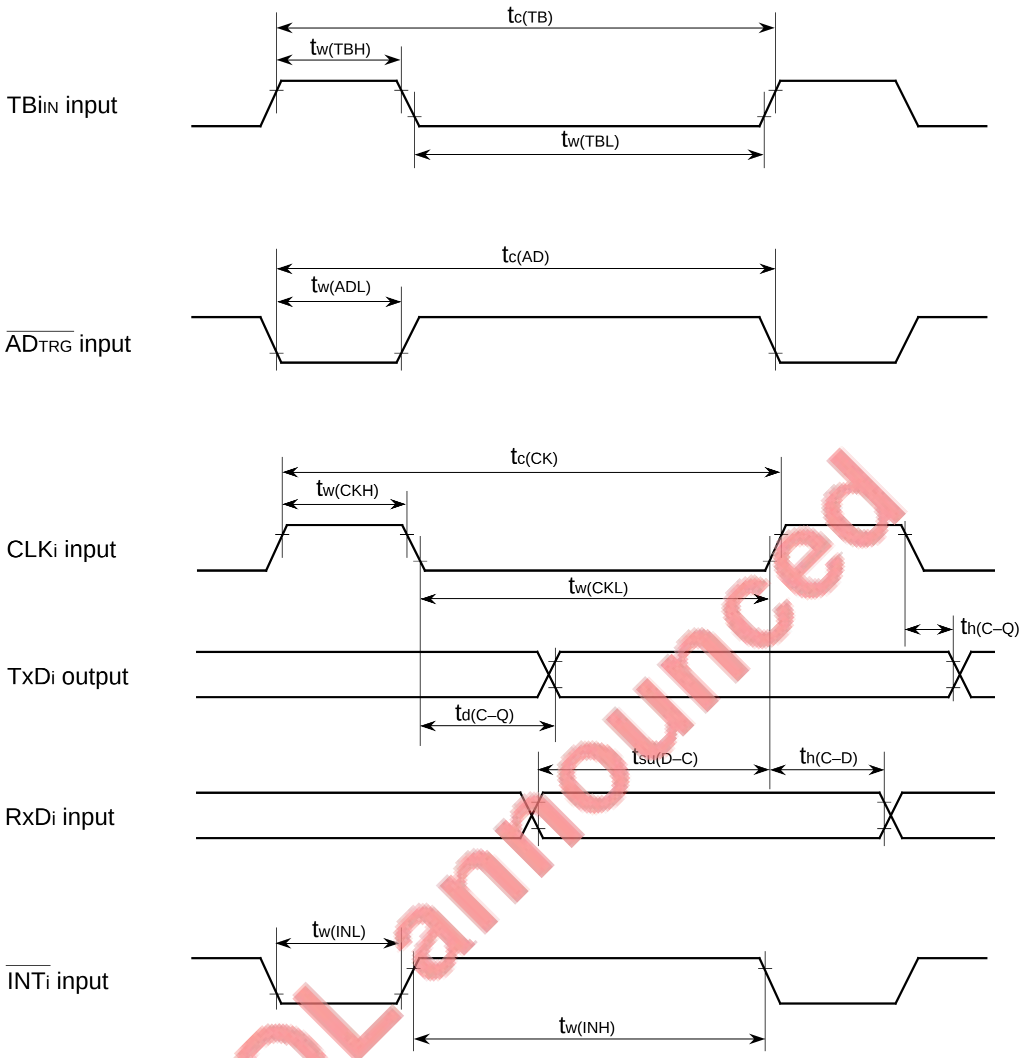
Serial I/O

Symbol	Parameter	Limits				Unit
		16 MHz		25 MHz		
		Min.	Max.	Min.	Max.	
$t_{c(CK)}$	CLK_i input cycle time	250		200		ns
$t_{w(CKH)}$	CLK_i input high-level pulse width	125		100		ns
$t_{w(CKL)}$	CLK_i input low-level pulse width	125		100		ns
$t_{d(C-Q)}$	TxD_i output delay time		90		80	ns
$t_{h(C-Q)}$	TxD_i hold time	0		0		ns
$t_{su(D-C)}$	RxD_i input setup time	30		30		ns
$t_{h(C-D)}$	RxD_i input hold time	90		90		ns

External interrupt $\overline{INT_i}$ input

Symbol	Parameter	Limits				Unit
		16 MHz		25 MHz		
		Min.	Max.	Min.	Max.	
$t_{w(INH)}$	$\overline{INT_i}$ input high-level pulse width	250		250		ns
$t_{w(INL)}$	$\overline{INT_i}$ input low-level pulse width	250		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, unless otherwise noted)

Symbol	Parameter	Limits				Unit
		16 MHz		25 MHz		
		Min.	Max.	Min.	Max.	
$t_{su(RDY-\phi_1)}$	$\overline{RDY}$ input setup time	60		55		ns
$t_{su(HOLD-\phi_1)}$	$\overline{HOLD}$ input setup time	60		55		ns
$t_{h(\phi_1-RDY)}$	$\overline{RDY}$ input hold time	0		0		ns
$t_{h(\phi_1-HOLD)}$	$\overline{HOLD}$ input hold time	0		0		ns

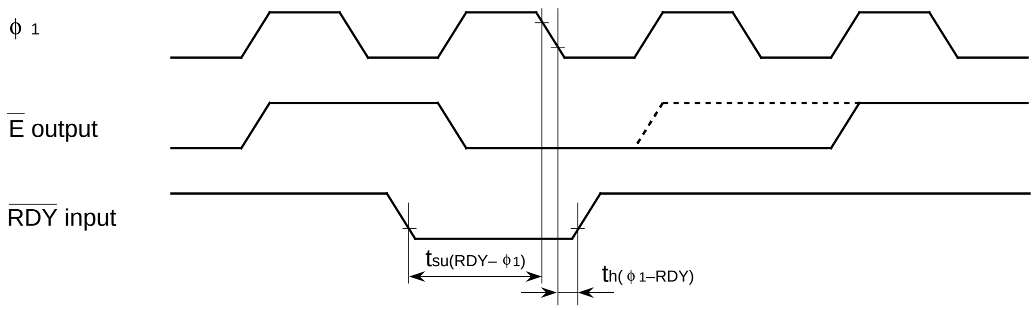
Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits				Unit
		16 MHz		25 MHz		
		Min.	Max.	Min.	Max.	
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time		50		50	ns

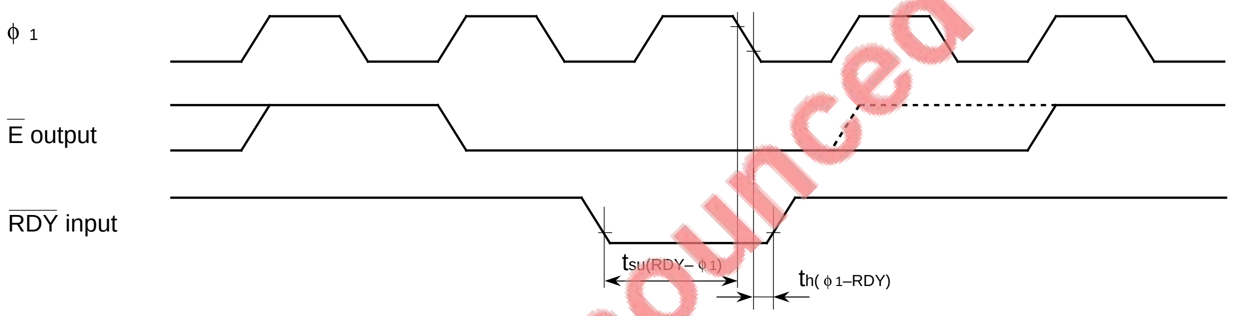
Note: For test conditions, refer to Figure 15.10.1.

● Ready function

With no Wait



With Wait



Test conditions

•V_{CC} = 5 V±10%

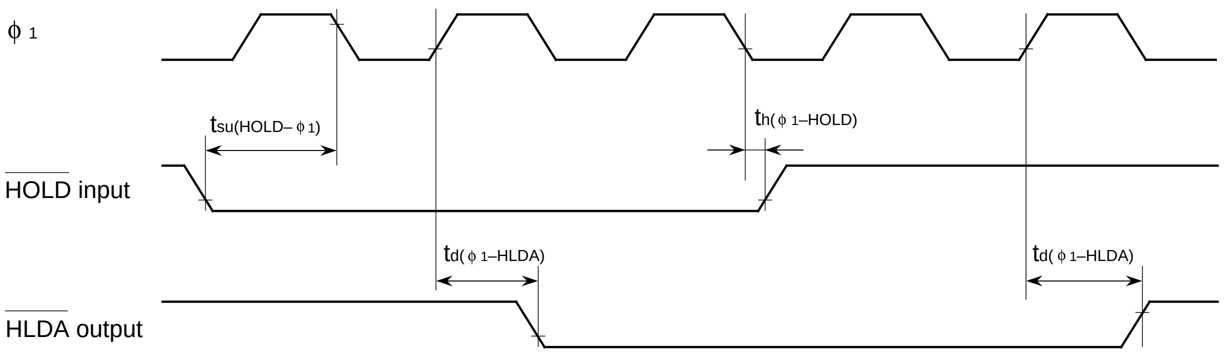
•Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V

•Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V

ELECTRICAL CHARACTERISTICS

15.6 Ready and Hold

● 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
		16 MHz		25 MHz		
		Min.	Max.	Min.	Max.	
t_c	External clock input cycle time	62		40		ns
$t_{w(H)}$	External clock input high-level pulse width	25		15		ns
$t_{w(L)}$	External clock input low-level pulse width	25		15		ns
t_r	External clock rise time		10		8	ns
t_f	External clock fall time		10		8	ns
$t_{su(P0D-E)}$	Port P0 input setup time	100		60		ns
$t_{su(P1D-E)}$	Port P1 input setup time	100		60		ns
$t_{su(P2D-E)}$	Port P2 input setup time	100		60		ns
$t_{su(P3D-E)}$	Port P3 input setup time	100		60		ns
$t_{su(P4D-E)}$	Port P4 input setup time	100		60		ns
$t_{su(P5D-E)}$	Port P5 input setup time	100		60		ns
$t_{su(P6D-E)}$	Port P6 input setup time	100		60		ns
$t_{su(P7D-E)}$	Port P7 input setup time	100		60		ns
$t_{su(P8D-E)}$	Port P8 input setup time	100		60		ns
$t_{h(E-P0D)}$	Port P0 input hold time	0		0		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		ns
$t_{h(E-P3D)}$	Port P3 input hold time	0		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		0		ns

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

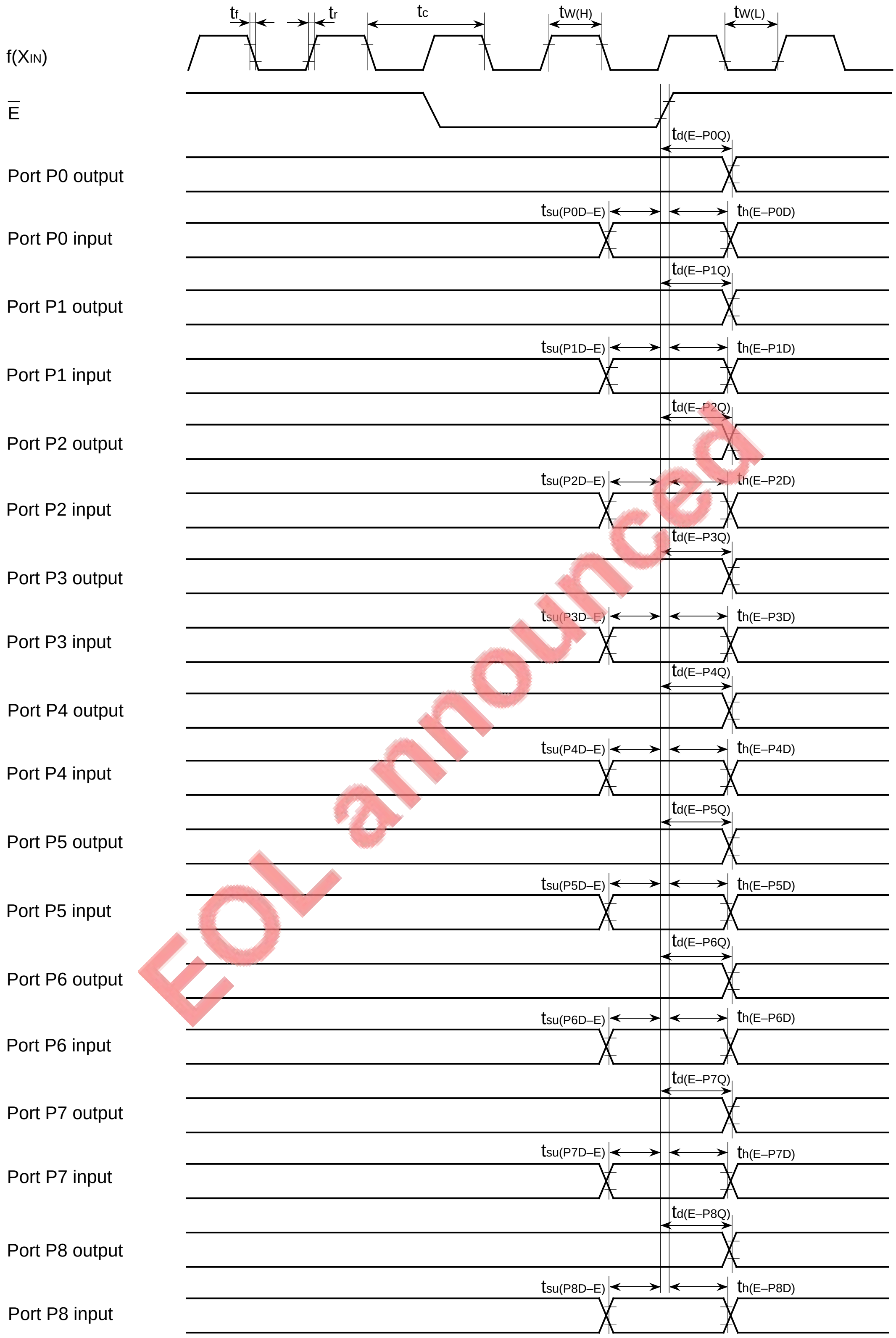
Symbol	Parameter	Limits				Unit
		16 MHz		25 MHz		
		Min.	Max.	Min.	Max.	
$t_{d(E-P0Q)}$	Port P0 data output delay time		100		80	ns
$t_{d(E-P1Q)}$	Port P1 data output delay time		100		80	ns
$t_{d(E-P2Q)}$	Port P2 data output delay time		100		80	ns
$t_{d(E-P3Q)}$	Port P3 data output delay time		100		80	ns
$t_{d(E-P4Q)}$	Port P4 data output delay time		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		100		80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		100		80	ns

Note: For test conditions, refer to Figure 15.10.1.

ELECTRICAL CHARACTERISTICS

15.7 Single-chip mode

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 : with no Wait

Timing requirements (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits 16 MHz Min.	Limits 16 MHz Max.	Limits 25 MHz Min.	Limits 25 MHz Max.	Unit
t_c	External clock input cycle time	62		40		ns
$t_{w(H)}$	External clock input high-level pulse width	25		15		ns
$t_{w(L)}$	External clock input low-level pulse width	25		15		ns
t_r	External clock rise time		10		8	ns
t_f	External clock fall time		10		8	ns
$t_{su(P1D-E)}$	Port P1 input setup time	45		30		ns
$t_{su(P2D-E)}$	Port P2 input setup time	45		30		ns
$t_{su(P4D-E)}$	Port P4 input setup time	100		60		ns
$t_{su(P5D-E)}$	Port P5 input setup time	100		60		ns
$t_{su(P6D-E)}$	Port P6 input setup time	100		60		ns
$t_{su(P7D-E)}$	Port P7 input setup time	100		60		ns
$t_{su(P8D-E)}$	Port P8 input setup time	100		60		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		0		ns

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits 16 MHz Min.	Limits 16 MHz Max.	Limits 25 MHz Min.	Limits 25 MHz Max.	Unit
$t_{d(E-P4Q)}$	Port P4 data output delay time		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		100		80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		100		80	ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time	0	20	0	18	ns
$t_{w(EL)}$	$\overline{E}$ low-level pulse width	95 *		50 *		ns
$t_{d(P0A-E)}$	Port P0 address output delay time	30 *		12 *		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE = "L")		70		45	ns
$t_{pxz(E-P1Z)}$	Port P1 floating start delay time (BYTE = "L")		5		5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	30 *		12 *		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	24 *		5 *		ns
$t_{h(E-P2Q)}$	Port P2 data output delay time		70		45	ns
$t_{pxz(E-P2Z)}$	Port P2 floating start delay time		5		5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	30 *		12 *		ns
$t_{h(P2A-ALE)}$	Port P2 address output delay time	24 *		5 *		ns
$t_{d(ALE-E)}$	ALE output delay time	4		4		ns
$t_{w(ALE)}$	ALE pulse width	35 *		22 *		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	30 *		20 *		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	30 *		20 *		ns

Note: For test conditions, refer to Figure 15.10.1.
* This is the value depending on $f(X_{IN})$. For data formula, refer to Table 15.8.1.

ELECTRICAL CHARACTERISTICS

15.8 Memory expansion mode and microprocessor mode : with no Wait

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits				Unit
		16 MHz		25 MHz		
		Min.	Max.	Min.	Max.	
$t_{h(E-P0A)}$	Port P0 address hold time	25*		18 *		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE = “L”)	9		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE = “L”)	25*		18 *		ns
$t_{pzx(E-P1Z)}$	Port P1 floating release delay time (BYTE = “L”)	36* (Note 1)		18 *		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE = “H”)	25*		18 *		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time	9		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time	25*		18 *		ns
$t_{pzx(E-P2Z)}$	Port P2 floating release delay time	36* (Note 1)		18 *		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	18		18		ns
$t_{h(E-RW)}$	R/$\overline{W}$ hold time	18		18		ns

Notes 1: For the M37702E2AXXXFP, M37702E2AFS, M37702E4AXXXFP, and M37702E4AFS, refer to section **“19.5.4 Bus timing and EPROM mode.”** For the M37703E2AXXXSP and M37703E4AXXXSP, refer to section **“20.6.2 Bus timing and EPROM mode.”**

2: For test conditions, refer to Figure 15.10.1.

*: This is the value depending on $f(X_{IN})$. For data formula, refer to Table 15.8.1.

Table 15.8.1 Bus timing data formula

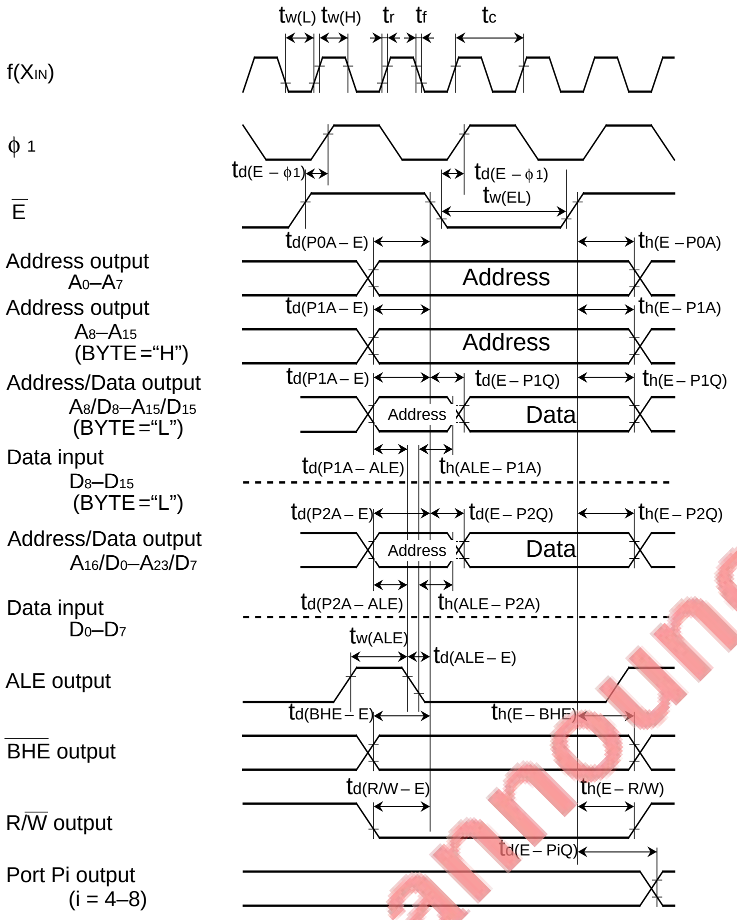
Sign \ $f(X_{IN})$	$f(X_{IN}) \leq 8$ MHz	8 MHz < $f(X_{IN}) \leq 16$ MHz	16 MHz < $f(X_{IN}) \leq 25$ MHz
$t_{w(EL)}$	$\frac{2 \times 10^9}{f(X_{IN})} - 30$		
$t_{d(P0A-E)}$, $t_{d(P1A-E)}$, $t_{d(P2A-E)}$	$100 + \frac{1 \times 10^9}{f(X_{IN})} - 125$	$30 + \frac{1.2 \times 10^9}{f(X_{IN})} - 75$	$12 + \frac{1 \times 10^9}{f(X_{IN})} - 40$
$t_{d(P1A-ALE)}$, $t_{d(P2A-ALE)}$	$\frac{1 \times 10^9}{f(X_{IN})} - 45$	$\frac{1 \times 10^9}{f(X_{IN})} - 38.5$	$\frac{1 \times 10^9}{f(X_{IN})} - 35$
$t_{w(ALE)}$	$\frac{1 \times 10^9}{f(X_{IN})} - 35$	$\frac{1 \times 10^9}{f(X_{IN})} - 27.5$	$\frac{1 \times 10^9}{f(X_{IN})} - 18$
$t_{d(BHE-E)}$, $t_{d(R/W-E)}$	$100 + \frac{1 \times 10^9}{f(X_{IN})} - 125$	$30 + \frac{1.2 \times 10^9}{f(X_{IN})} - 75$	$20 + \frac{1 \times 10^9}{f(X_{IN})} - 40$
$t_{h(E-P0A)}$, $t_{h(E-P1A)}$, $t_{h(E-P1Q)}$, $t_{h(E-P2Q)}$	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 12.5$	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 6.25$	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 2$
$t_{pzx(E-P1Z)}$, $t_{pzx(E-P2Z)}$ **(Note)**	$\frac{1 \times 10^9}{f(X_{IN})} - 30$	$\frac{1 \times 10^9}{f(X_{IN})} - 26$	$\frac{1 \times 10^9}{f(X_{IN})} - 22$

Note: For the M37702E2AXXXFP, M37702E2AFS, M37702E4AXXXFP, and M37702E4AFS, refer to section **“19.5.4 Bus timing and EPROM mode.”** For the M37703E2AXXXSP and M37703E4AXXXSP, refer to section **“20.6.2 Bus timing and EPROM mode.”**

15.8 Memory expansion mode and microprocessor mode : with no Wait

Memory expansion mode and microprocessor mode ; With no Wait

<Write>



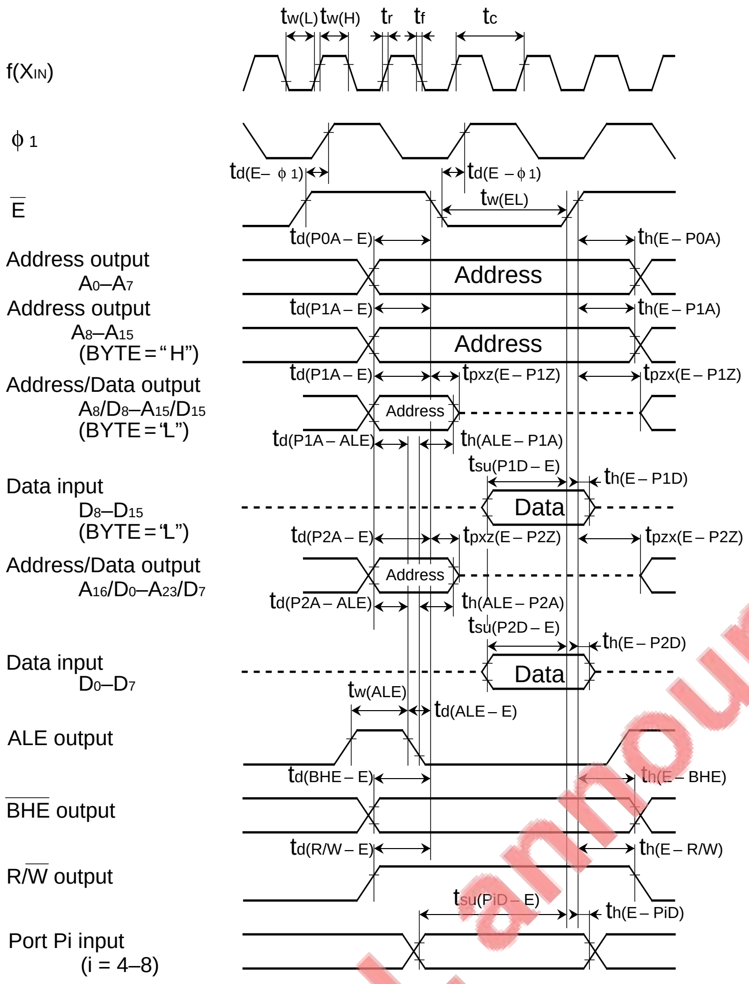
Test conditions (φ 1, E, P0–P3)
- V_{CC} = 5 V ± 10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

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} = 2.0V

15.8 Memory expansion mode and microprocessor mode : with no Wait

Memory epxansion mode and microprocessor mode ; With no Wait

<Read>



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

15.9 Memory expansion mode and microprocessor mode : with Wait

Timing requirements (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits				Unit
		16 MHz		25 MHz		
		Min.	Max.	Min.	Max.	
t_c	External clock input cycle time	62		40		ns
$t_{w(H)}$	External clock input high-level pulse width	25		15		ns
$t_{w(L)}$	External clock input low-level pulse width	25		15		ns
t_r	External clock rise time		10		8	ns
t_f	External clock fall time		10		8	ns
$t_{su(P1D-E)}$	Port P1 input setup time	45		30		ns
$t_{su(P2D-E)}$	Port P2 input setup time	45		30		ns
$t_{su(P4D-E)}$	Port P4 input setup time	100		60		ns
$t_{su(P5D-E)}$	Port P5 input setup time	100		60		ns
$t_{su(P6D-E)}$	Port P6 input setup time	100		60		ns
$t_{su(P7D-E)}$	Port P7 input setup time	100		60		ns
$t_{su(P8D-E)}$	Port P8 input setup time	100		60		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		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
		16 MHz		25 MHz		
		Min.	Max.	Min.	Max.	
$t_{d(E-P4Q)}$	Port P4 data output delay time		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		100		80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		100		80	ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time	0	20	0	18	ns
$t_{w(EL)}$	$\overline{E}$ low-pulse width	220 *		130 *		ns
$t_{d(P0A-E)}$	Port P0 address output delay time	30 *		12 *		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE = “L”)		70		45	ns
$t_{pxz(E-P1Z)}$	Port P1 floating start delay time (BYTE = “L”)		5		5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	30 *		12 *		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	24 *		5 *		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time		70		45	ns
$t_{pxz(E-P2Z)}$	Port P2 floating start delay time		5		5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	30 *		12 *		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time	24 *		5 *		ns
$t_{d(ALE-E)}$	ALE output delay time	4		4		ns
$t_{w(ALE)}$	ALE pulse width	35 *		22 *		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	30 *		20 *		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	30 *		20 *		ns

Note: For test conditions, refer to Figure 15.10.1.
*: This is the value depending on $f(X_{IN})$. For data formula, refer to Table 15.9.1.

ELECTRICAL CHARACTERISTICS

15.9 Memory expansion mode and microprocessor mode : with Wait

Switching characteristics (V_{CC} = 5 V±10%, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits 16 MHz Min.	Limits 16 MHz Max.	Limits 25 MHz Min.	Limits 25 MHz Max.	Unit
$t_{h(E-P0A)}$	Port P0 address hold time	25*		18 *		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE = “L”)	9		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE = “L”)	25*		18 *		ns
$t_{pzx(E-P1Z)}$	Port P1 floating release delay time (BYTE = “L”)	36* (Note 1)		18 *		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE = “H”)	25*		18 *		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time	9		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time	25*		18 *		ns
$t_{pzx(E-P2Z)}$	Port P2 floating release delay time	36* (Note 1)		18 *		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time	18		18		ns

Notes 1: For the M37702E2AXXXFP, M37702E2AFS, M37702E4AXXXFP, and M37702E4AFS, refer to section **“19.5.4 Bus timing and EPROM mode.”** For the M37703E2AXXXSP and M37703E4AXXXSP, refer to section **“20.6.2 Bus timing and EPROM mode.”**

2: For test conditions, refer to Figure 15.10.1.

*: This is the value depending on $f(X_{IN})$. For data formula, refer to Table 15.9.1.

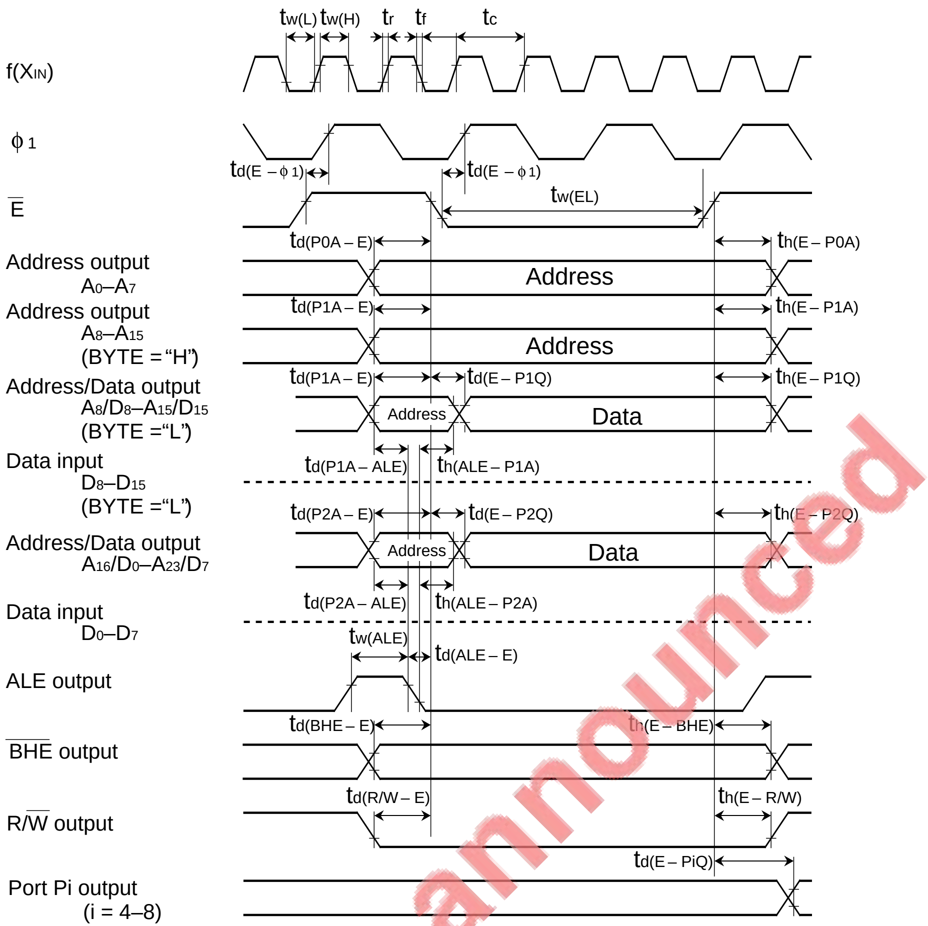
Table 15.9.1 Bus timing data formula

Sign \ $f(X_{IN})$	$f(X_{IN}) \le 8$ MHz	8 MHz < $f(X_{IN}) \le 16$ MHz	16 MHz < $f(X_{IN}) \le 25$ MHz
$t_{w(EL)}$	$\frac{4 \times 10^9}{f(X_{IN})} - 30$		
$t_{d(P0A-E)}$, $t_{d(P1A-E)}$, $t_{d(P2A-E)}$	$100 + \frac{1 \times 10^9}{f(X_{IN})} - 125$	$30 + \frac{1.2 \times 10^9}{f(X_{IN})} - 75$	$12 + \frac{1 \times 10^9}{f(X_{IN})} - 40$
$t_{d(P1A-ALE)}$, $t_{d(P2A-ALE)}$	$\frac{1 \times 10^9}{f(X_{IN})} - 45$	$\frac{1 \times 10^9}{f(X_{IN})} - 38.5$	$\frac{1 \times 10^9}{f(X_{IN})} - 35$
$t_{w(ALE)}$	$\frac{1 \times 10^9}{f(X_{IN})} - 35$	$\frac{1 \times 10^9}{f(X_{IN})} - 27.5$	$\frac{1 \times 10^9}{f(X_{IN})} - 18$
$t_{d(BHE-E)}$, $t_{d(R/W-E)}$	$100 + \frac{1 \times 10^9}{f(X_{IN})} - 125$	$30 + \frac{1.2 \times 10^9}{f(X_{IN})} - 75$	$20 + \frac{1 \times 10^9}{f(X_{IN})} - 40$
$t_{h(E-P0A)}$, $t_{h(E-P1A)}$, $t_{h(E-P1Q)}$, $t_{h(E-P2Q)}$	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 12.5$	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 6.25$	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 2$
$t_{pzx(E-P1Z)}$, $t_{pzx(E-P2Z)}$ **(Note)**	$\frac{1 \times 10^9}{f(X_{IN})} - 30$	$\frac{1 \times 10^9}{f(X_{IN})} - 26$	$\frac{1 \times 10^9}{f(X_{IN})} - 22$

Note: For the M37702E2AXXXFP, M37702E2AFS, M37702E4AXXXFP, and M37702E4AFS, refer to section **“19.5.4 Bus timing and EPROM mode.”** For the M37703E2AXXXSP and M37703E4AXXXSP, refer to section **“20.6.2 Bus timing and EPROM mode.”**

Memory expansion mode and microprocessor mode ; With Wait

<Write>



Test conditions (φ1, $\overline{E}$, P0–P3)

- V_{CC} = 5 V ± 10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0V
- Data input : V_{IL} = 0.8 V, V_{IH} = 2.5 V

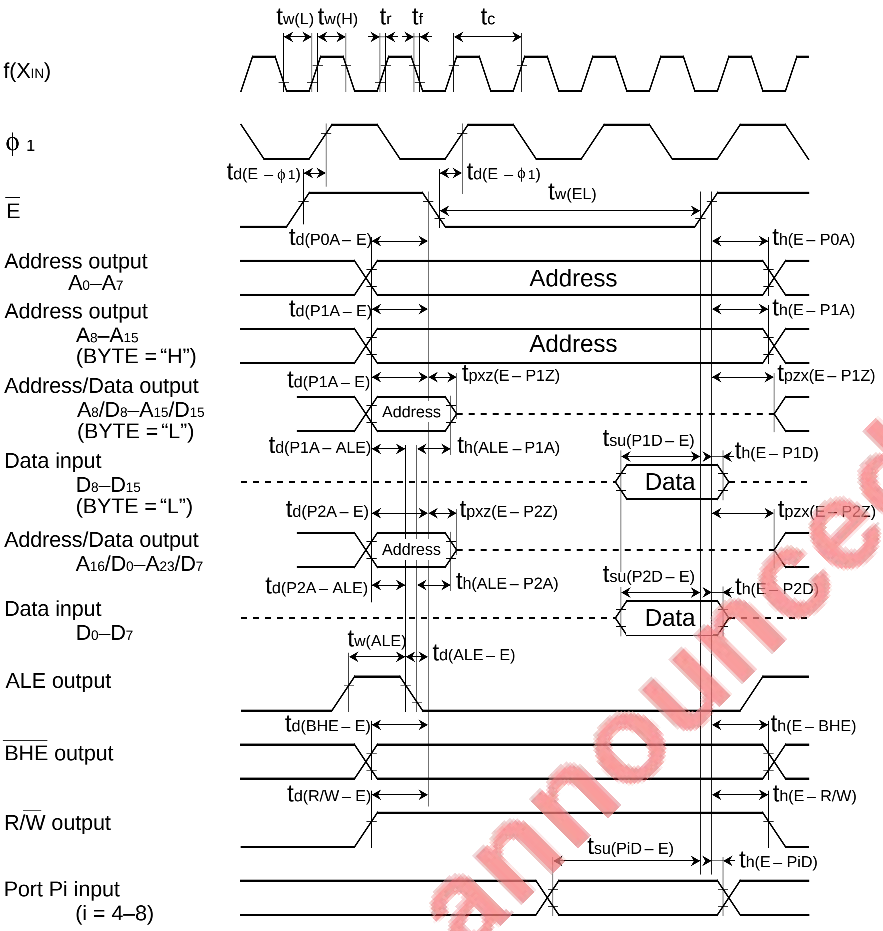
Test conditions (P4–P8)

- V_{CC} = 5V ± 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 : with Wait

Memory expansion mode and microprocessor mode ; With Wait

<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 Testing circuit for ports P0 to P8, ϕ_1, and $\overline{E}$

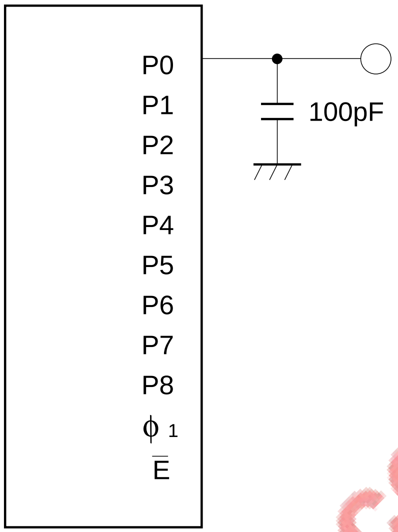


Fig. 15.10.1 Testing circuit for ports P0 to P8, ϕ_1, and $\overline{E}$

15.10 Testing circuit for ports P0 to P8, ϕ_1, and $\overline{E}$

MEMORANDUM

CHAPTER 16

STANDARD CHARACTERISTICS

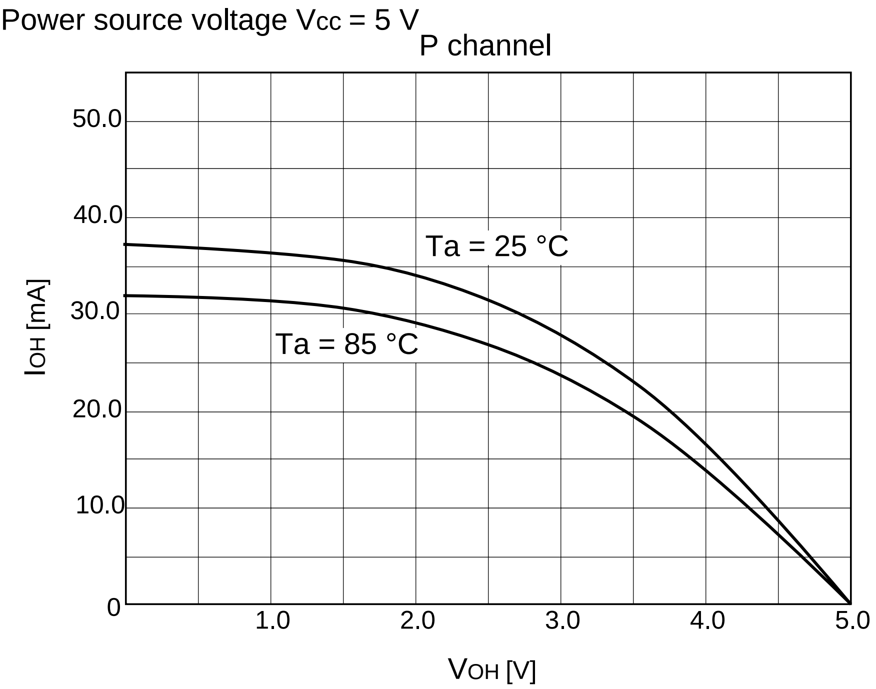
STANDARD CHARACTERISTICS

16.1 Standard characteristics

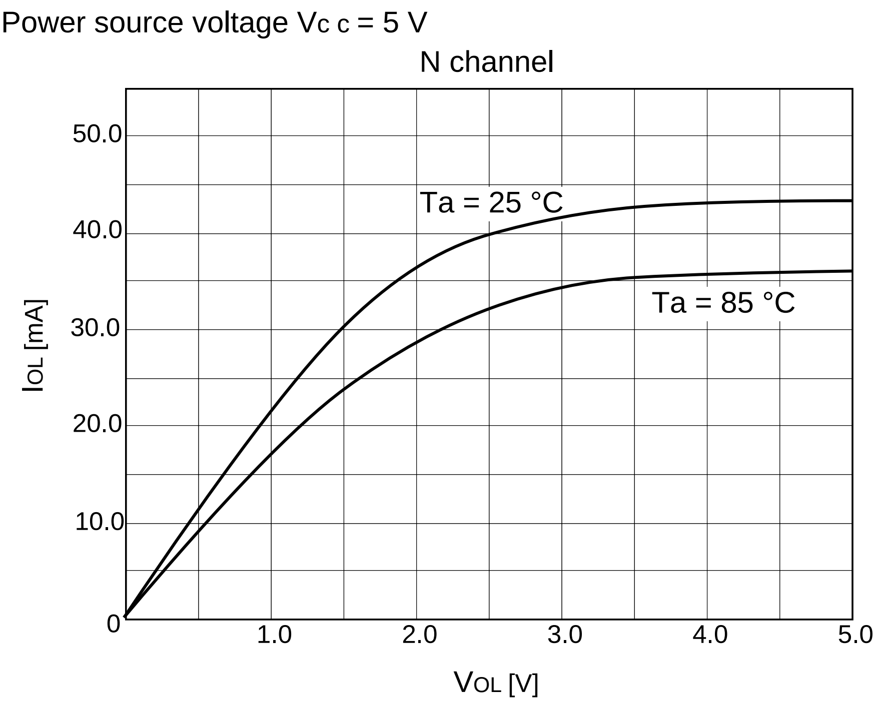
The data described below are characteristic examples for M37702M2BXXXFP. The data is not guaranteed value. Refer to “**Chapter 15. ELECTRICAL CHARACTERISTICS**” for rated value.

16.1.1 Port standard characteristics

(1) Programmable I/O port (CMOS output) P channel I_{OH}–V_{OH} characteristics



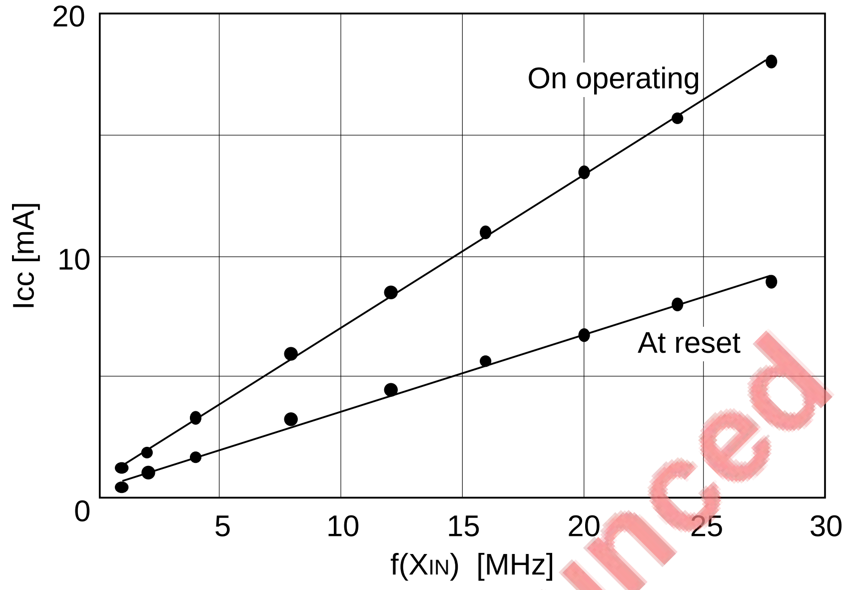
(2) Programmable I/O port (CMOS output) N channel I_{OL}–V_{OL} characteristics



16.1.2 I_{CC}–f(X_{IN}) standard characteristics

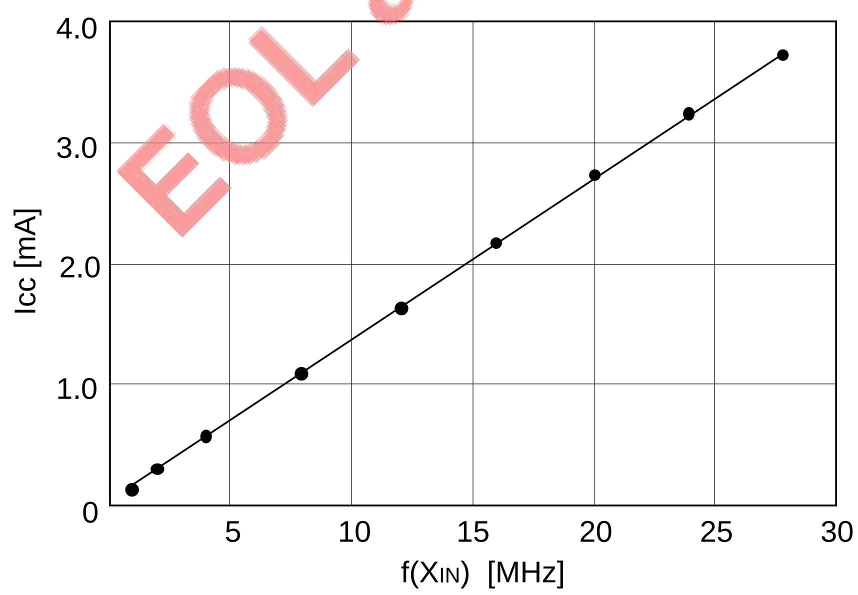
(1) I_{CC}–f(X_{IN}) characteristics on operating and at reset

Measurement condition (V_{CC} = 5 V, Ta = 25 °C, f(X_{IN}) : square waveform input, single-chip mode)



(2) I_{CC}–f(X_{IN}) characteristics during wait

Measurement condition (V_{CC} = 5 V, Ta = 25 °C, f(X_{IN}) : square waveform input, single-chip mode)



STANDARD CHARACTERISTICS

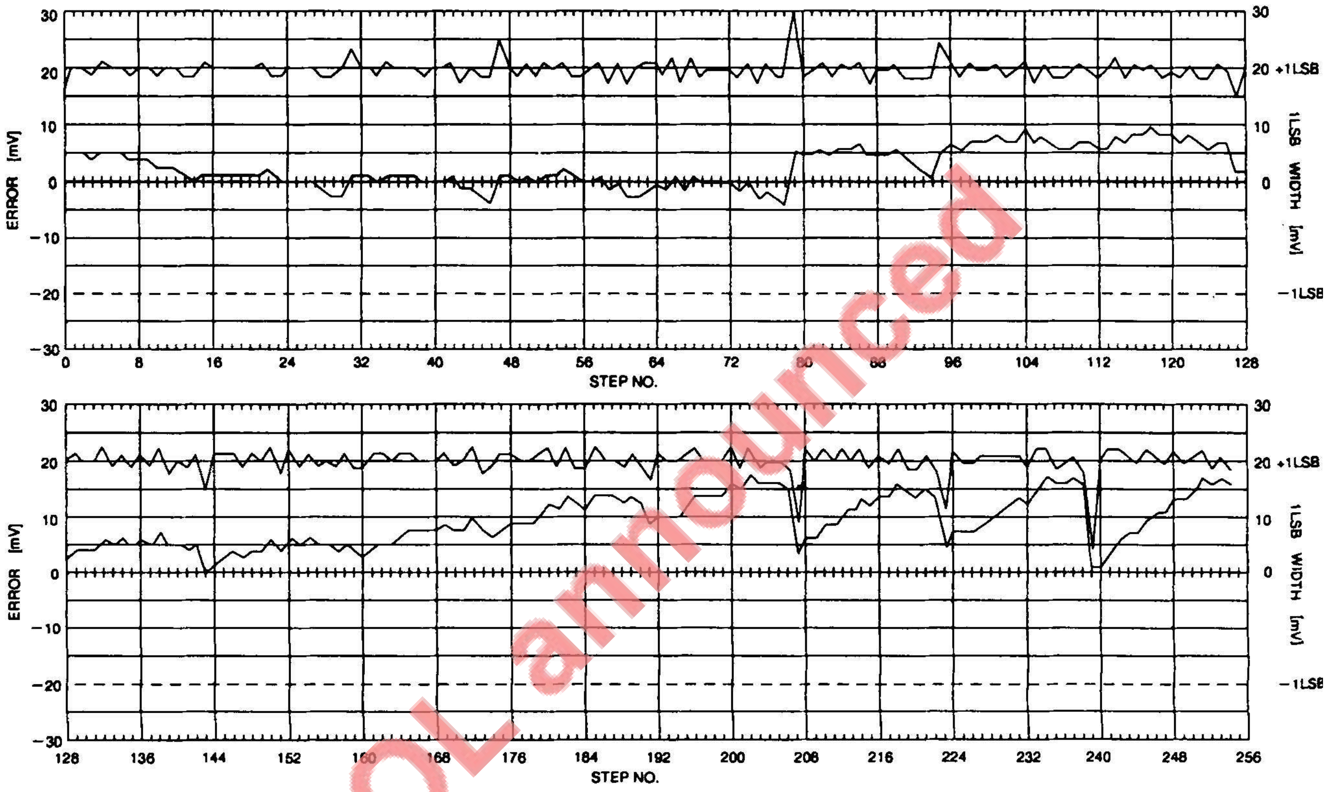
16.1 Standard characteristics

16.1.3 A–D converter standard characteristics

The lower lines of the graph indicate 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 00_{16} to 01_{16} should occur at 10 mV, but the measured value is 5 mV. Therefore, the measured point of change is 10 + 5 = 15 mV.

The upper lines of the graph indicate the input voltage width for which the output code is constant. For example, the measured input voltage width for which the output code is $0F_{16}$ is 22 mV. Therefore, the differential non-linear error is 22 – 20 = 2 mV (0.1LSB).

Measurement condition (V_{CC} = 5.12 V, X_{IN} = 25 MHz, Temp. = 25°C)



CHAPTER 17
APPLICATION

This chapter describes application. Application shown here is just an example. The user shall modify them according to the actual application and test them.

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. Refer to **"Chapter 18. LOW VOLTAGE VERSION"** for timing requirements and application of the low voltage version.

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 having the cost priority which is suited for 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. Accordingly, an n-bit ($n \leq 8$) address latch is required for latching addresses (n bits of A_{23} to A_{16}).

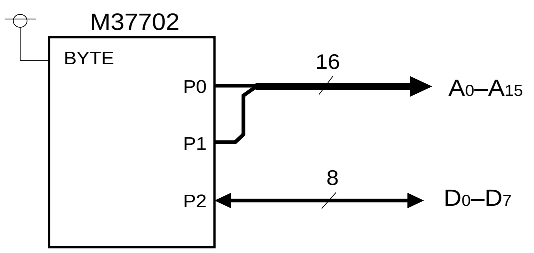
(3) Medium model B

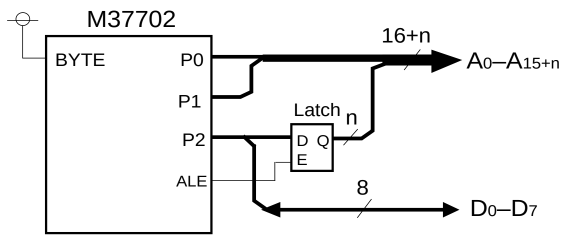
This is an expansion model of which external data bus width is 16 bits and accessible area is expanded up to 64 Kbytes. This expansion model is used when having the speed performance priority. In this expansion model, the middle-order 8 bits of the external address bus (A_{15} to A_8) are multiplexed with the external data bus. Accordingly, an 8-bit address latch is required for latching address (A_{15} to A_8).

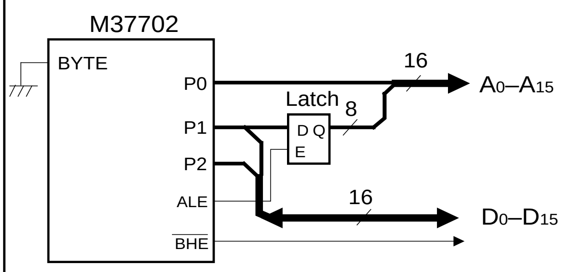
(4) Maximum model

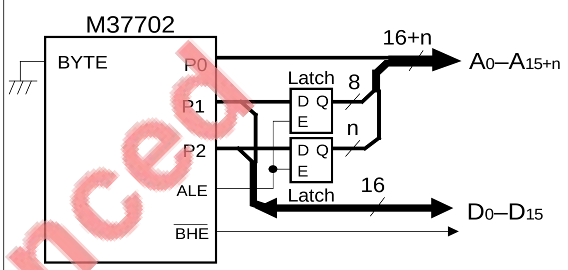
This is an expansion model of which external data bus width is 16 bits and accessible area is expanded up to 16 Mbytes. In this expansion model, the high- and middle-order 16 bits of the external address bus (A_{23} to A_8) are multiplexed with the external data bus. Accordingly, an 8-bit address latch 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

<table>
<tr><th>Access area
External data bus width</th><th>Maximum 64 Kbytes</th><th>Maximum 16 Mbytes</th></tr>
<tr><td>8-bit width;
BYTE = “H”</td><td>

Memory expansion model: Minimum model</td><td>

Memory expansion model: Medium model A</td></tr>
<tr><td>16-bit width;
BYTE = “L”</td><td>

Memory expansion model: Medium model B</td><td>

Memory expansion model: Maximum model</td></tr>
</table>

Notes 1: Refer to **“Chapter 12. CONNECTION WITH EXTERNAL DEVICES”** 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 M37702’s Vss line. (Refer to **“Appendix 5. Countermeasures against noise.”**)

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; $t_{a(AD)}$**

$t_{a(AD)} = t_{d(P0A/P1A/P2A-E)} + t_{w(EL)} - t_{su(P2D/P1D-E)} -$ (address decode time*1 + address latch delay time*2)

$t_{d(P0A/P1A/P2A-E)}$: $t_{d(P0A-E)}$, $t_{d(P1A-E)}$, or $t_{d(P2A-E)}$

$t_{su(P2D/P1D-E)}$: $t_{su(P2D-E)}$ or $t_{su(P1D-E)}$

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 calculation formulas for each parameter; Table 17.1.3 lists the data of each parameter; Figure 17.1.1 shows the bus timing diagrams.
Figures 17.1.2 and 17.1.4 show the relationship between $t_{a(A-D)}$ 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 Calculation formulas for each parameter (unit: ns)

Parameter \ $f(X_{IN})$	$f(X_{IN}) \le 8$ MHz		8 MHz $< f(X_{IN}) \le 16$ MHz		16 MHz $< f(X_{IN}) \le 25$ MHz	
	No Wait	Wait	No Wait	Wait	No Wait	Wait
$t_{d(P0A-E)}$, $t_{d(P1A-E)}$, $t_{d(P2A-E)}$	$100 + \frac{1 \times 10^9}{f(X_{IN})} - 125$		$30 + \frac{1.2 \times 10^9}{f(X_{IN})} - 75$		$12 + \frac{1 \times 10^9}{f(X_{IN})} - 40$	
$t_{w(EL)}$	$\frac{2 \times 10^9}{f(X_{IN})} - 30$	$\frac{4 \times 10^9}{f(X_{IN})} - 30$	$\frac{2 \times 10^9}{f(X_{IN})} - 30$	$\frac{4 \times 10^9}{f(X_{IN})} - 30$	$\frac{2 \times 10^9}{f(X_{IN})} - 30$	$\frac{4 \times 10^9}{f(X_{IN})} - 30$

Table 17.1.3 Data of each parameter (unit: ns)

Parameter \ Type	16 MHz version	25 MHz version
$t_{su(P1D-E)}$, $t_{su(P2D-E)}$	45	30
$t_{d(E-P1Q)}$, $t_{d(E-P2Q)}$	70	45

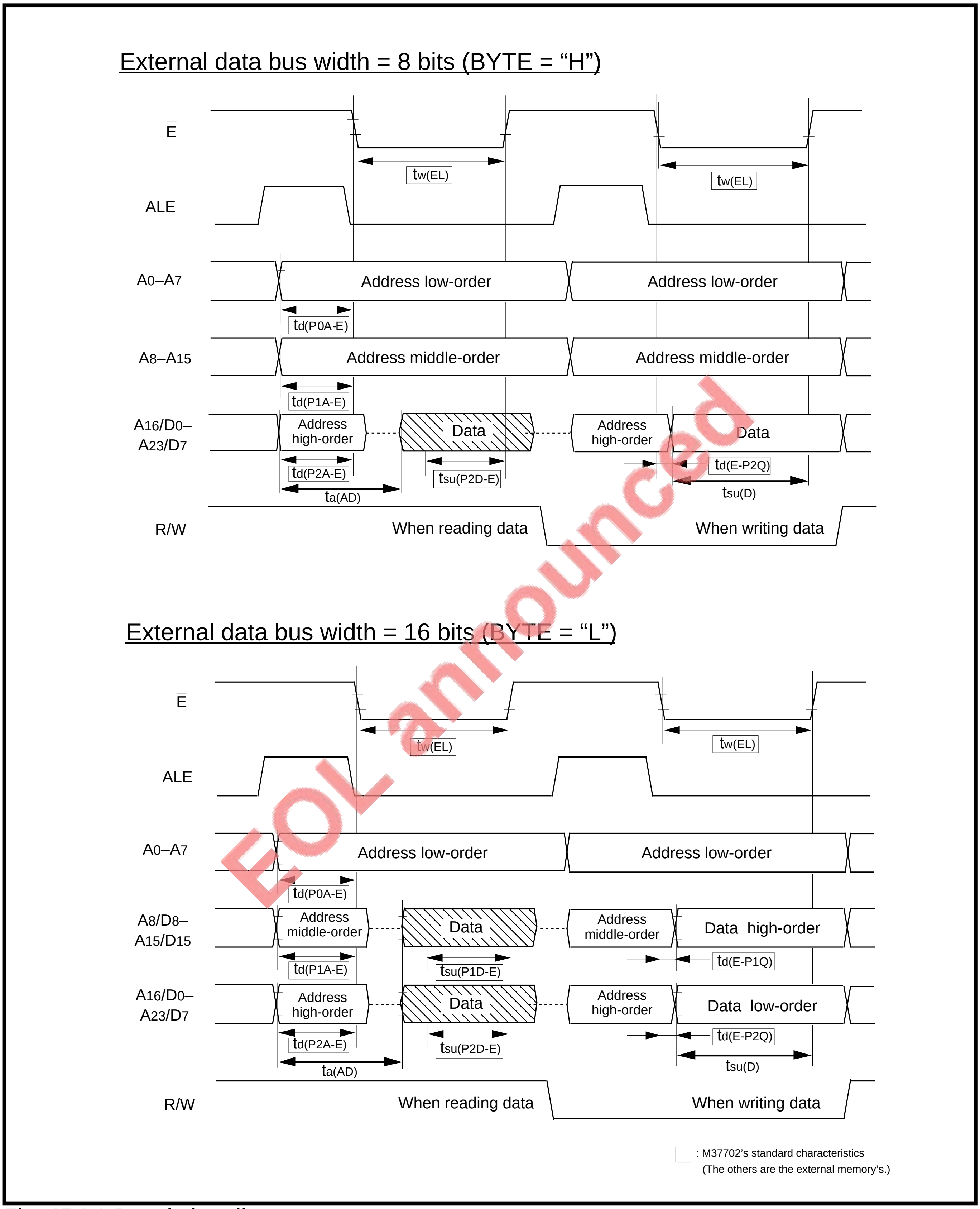


Fig. 17.1.1 Bus timing diagrams

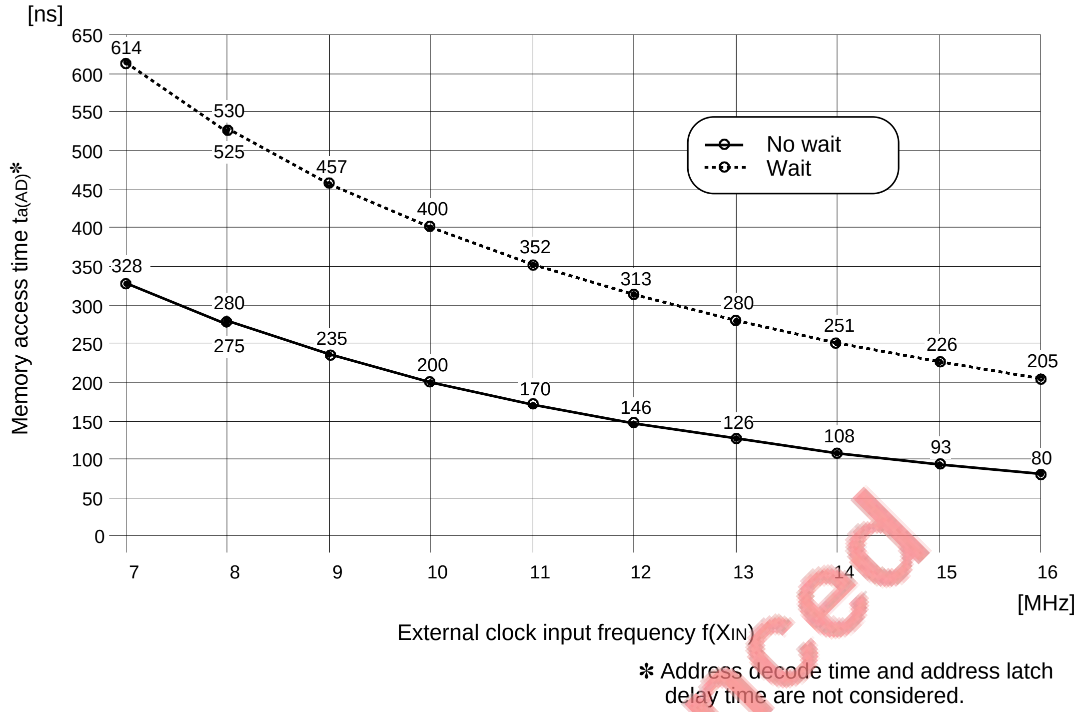


Fig. 17.1.2 Relationship between $t_{a(AD)}$ and $f(X_{IN})$ (16 MHz version)

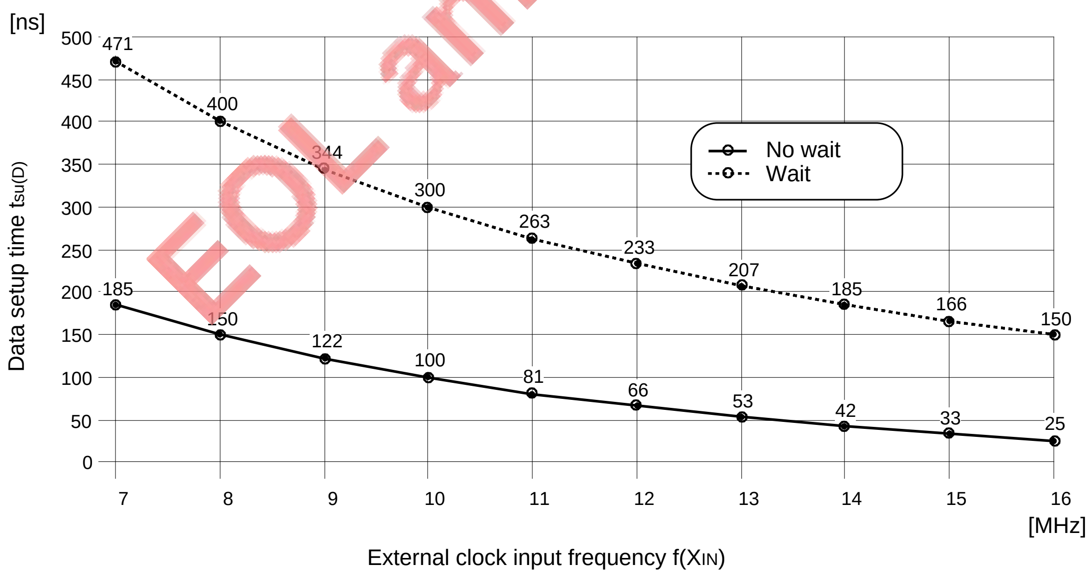
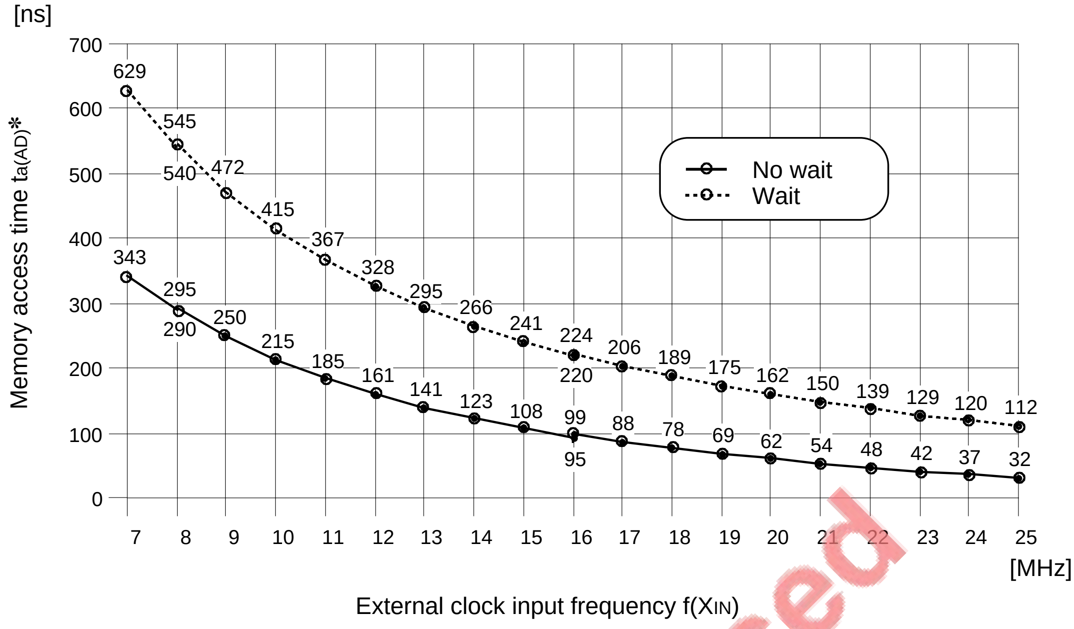


Fig. 17.1.3 Relationship between $t_{su(D)}$ and $f(X_{IN})$ (16 MHz version)



✻ Address decode time and address latch delay time are not considered.

Fig. 17.1.4 Relationship between $t_{a(AD)}$ and $f(X_{IN})$ (25 MHz version)

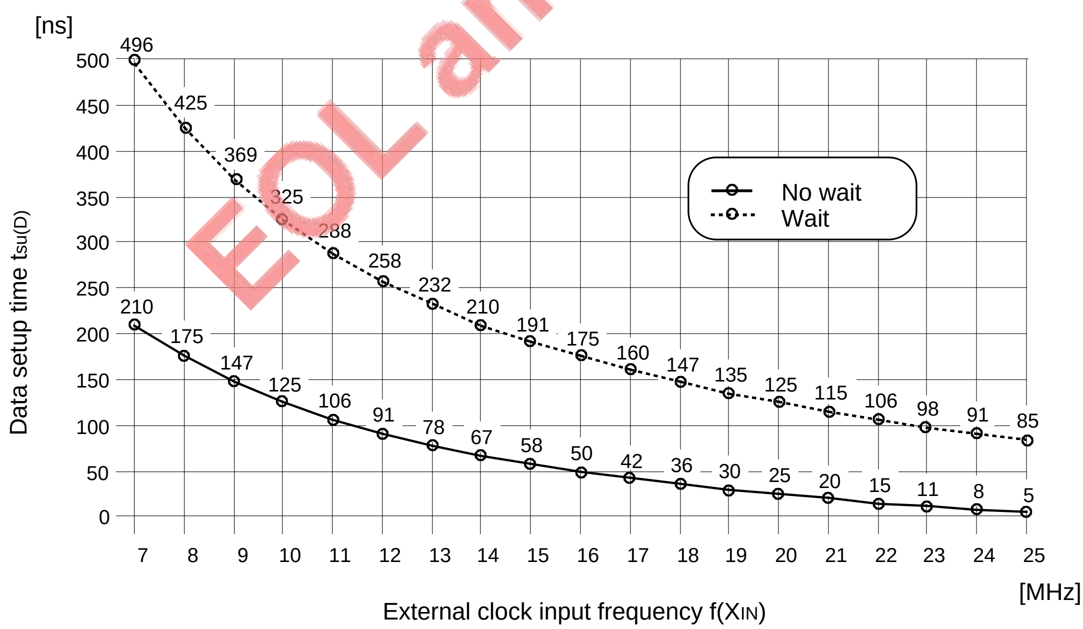


Fig. 17.1.5 Relationship between $t_{su(D)}$ and $f(X_{IN})$ (25 MHz version)

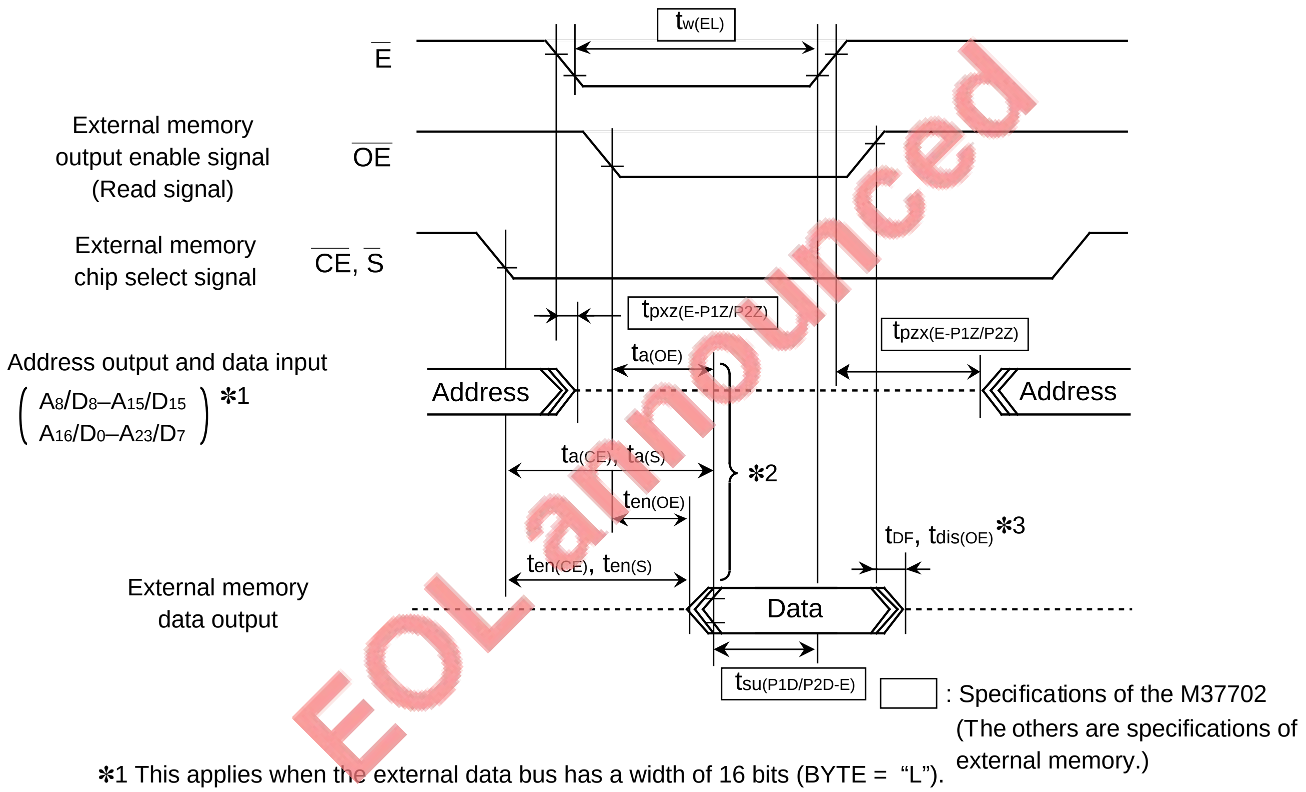
17.1.3 Points in memory expansion

(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 $t_{pxz(E-P1Z/P2)}$ after the falling edge of the $\overline{E}$ signal till $t_{pzx(E-P1Z/P2Z)}$ after the rising edge of the $\overline{E}$ signal. Table 17.1.4 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 an external memory

Table 17.1.4 Values of $t_{pxz(E-P1Z/P2Z)}$ and formulas to calculate $t_{pzx(E-P1Z/P2Z)}$ (unit : ns)

Parameter \ $f(X_{IN})$	$f(X_{IN}) \leq 8$ MHz	8 MHz $< f(X_{IN}) \leq 16$ MHz	16 MHz $< f(X_{IN}) \leq 25$ MHz
$t_{pxz(E-P1Z)}$ $t_{pxz(E-P2Z)}$	5	5	5
$t_{pzx(E-P1Z)}$ $t_{pzx(E-P2Z)}$ **(Note)**	$\frac{1 \times 10^9}{f(X_{IN})} - 30$	$\frac{1 \times 10^9}{f(X_{IN})} - 26$	$\frac{1 \times 10^9}{f(X_{IN})} - 22$

Note: In the M37702E2AXXXFP, the M37702E2AFS, the M37702E4AXXXFP, the M37702E4AFS, the M37703E2AXXXSP, and the M37703E4AXXXSP, refer to section **"19.5.4 Bus timing and EPROM mode."**

APPLICATION

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 starts after $t_{d(E-P1Q/P2Q)}$ passes from falling of the $\overline{E}$ signal. Its validated data is output continuously until $t_{h(E-P1Q/P2Q)}$ passes from rising of the $\overline{E}$ signal.

Table 17.1.5 lists the calculation formulas of $t_{h(E-P1Q/P2Q)}$. Table 17.1.6 lists the constants of $t_{d(E-P1Q/P2Q)}$.

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.

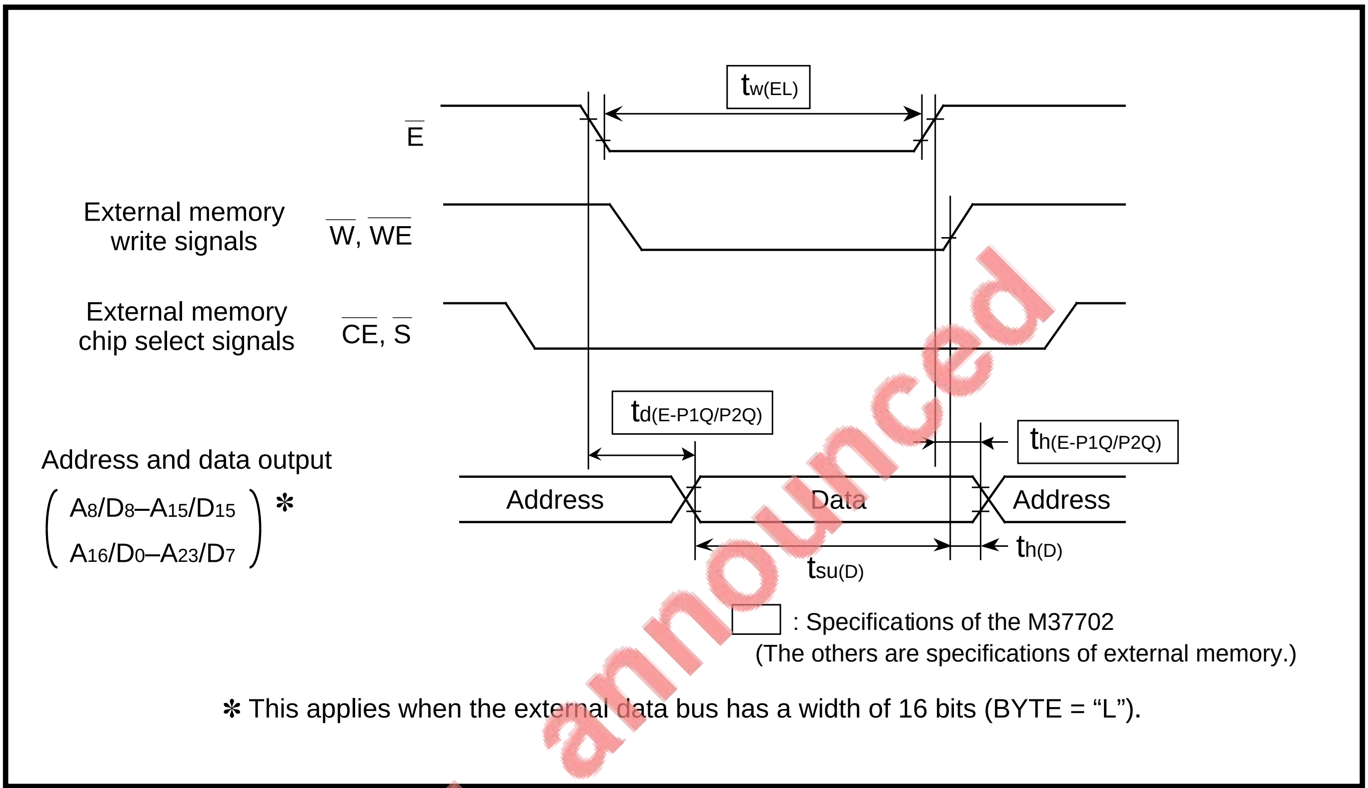


Fig. 17.1.7 Timing at which data is written to an external memory

Table 17.1.5 Calculation formulas of $t_{h(E-P1Q/P2Q)}$ (unit: ns)

Parameter \ $f(X_{IN})$	$f(X_{IN}) \leq 8$ MHz	8 MHz $< f(X_{IN}) \leq 16$ MHz	16 MHz $< f(X_{IN}) \leq 25$ MHz
$t_{h(E-P1Q)}$ $t_{h(E-P2Q)}$	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 12.5$	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 6.25$	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 2$

Table 17.1.6 Constants of $t_{d(E-P1Q/P2Q)}$ (unit: ns)

Parameter \ Microcomputer type	16 MHz version	25 MHz version
$t_{d(E-P1Q)}$ $t_{d(E-P2Q)}$	70	45

(3) Precautions on memory expansion

As described in ① to ③ below, if specifications of the external memory do not match those of the M37702, 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 the falling edge of the $\overline{E}$ signal
③ When using an external memory that outputs data for more than $t_{pzx(E-P1Z/P2Z)}$ after the rising edge of the $\overline{E}$ signal

① When using external memory that requires long access time, $t_{a(AD)}$

If the M37702'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 software Wait. (Refer to section **"12.2 Software Wait."**)
- Use Ready function. (Refer to section **"12.3 Ready function."**)

Figure 17.1.8 shows an example of a Ready signal generating circuit (no Wait). Figure 17.1.9 shows an example of a Ready signal generating circuit (with Wait).
Ready function is valid for the internal areas, so that the circuits in Figures 17.1.8 and 17.1.9 use the chip select signal ($\overline{CS_2}$) to specify the area where Ready function is valid.

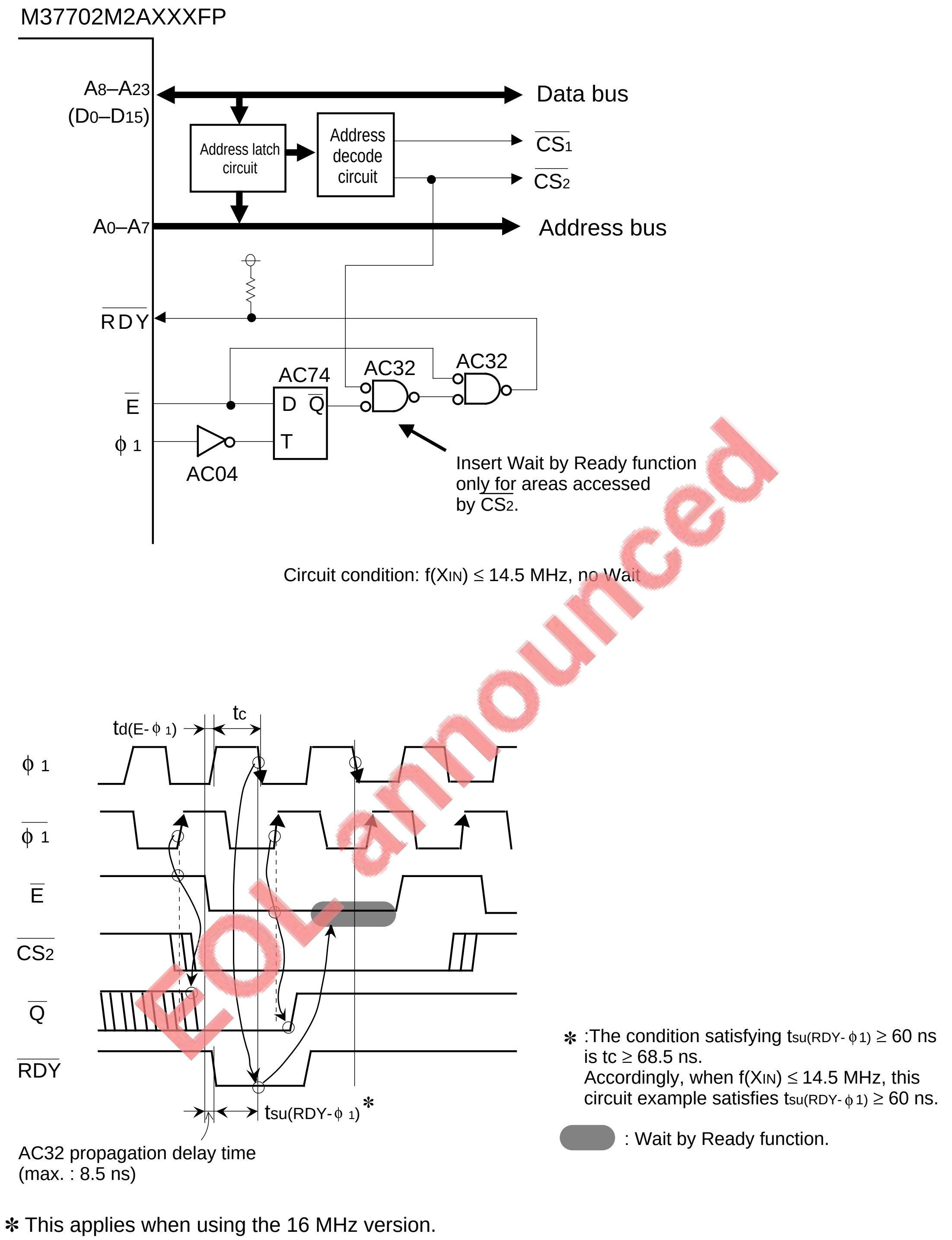
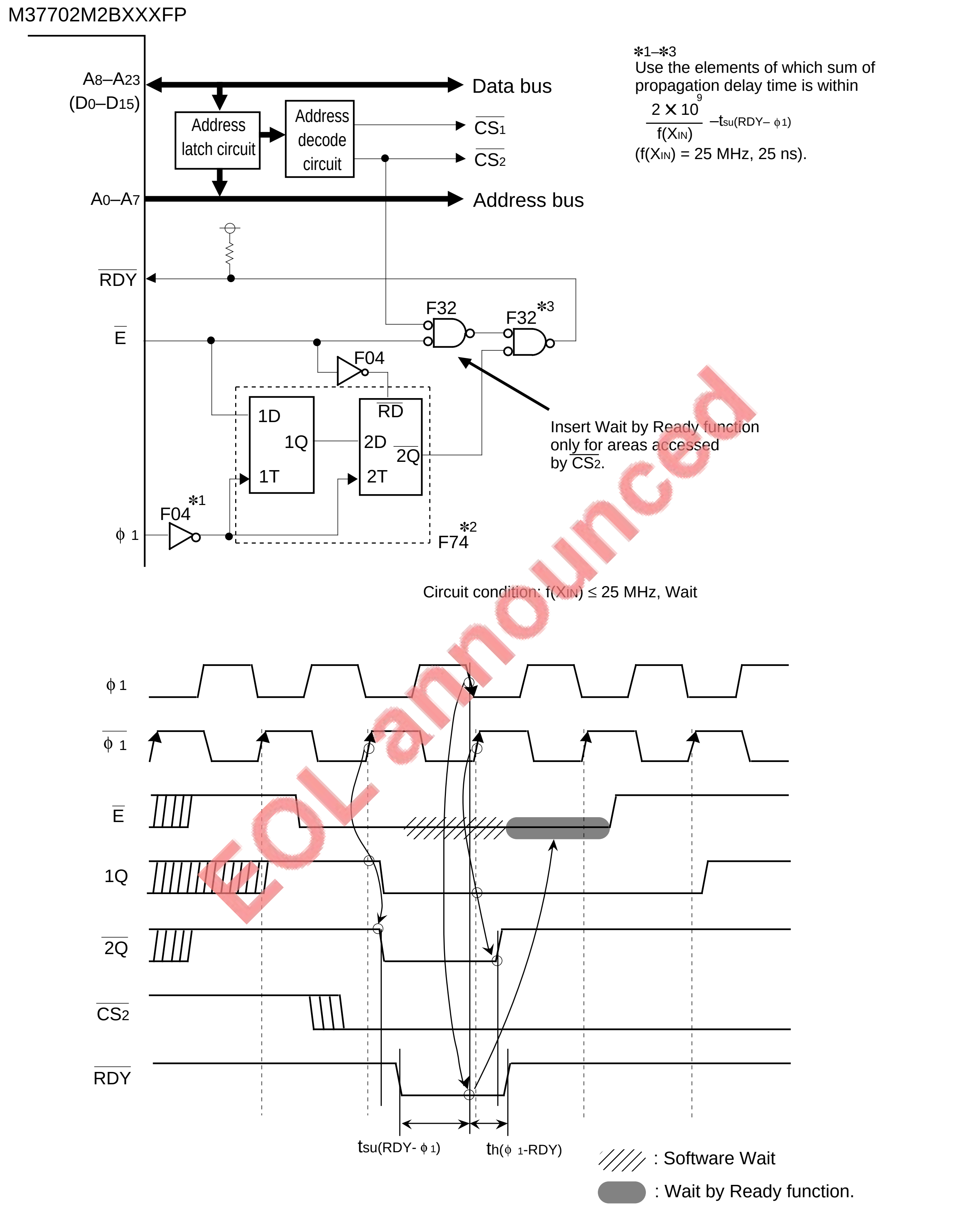


Fig. 17.1.8 Example of Ready signal generating circuit (no Wait)

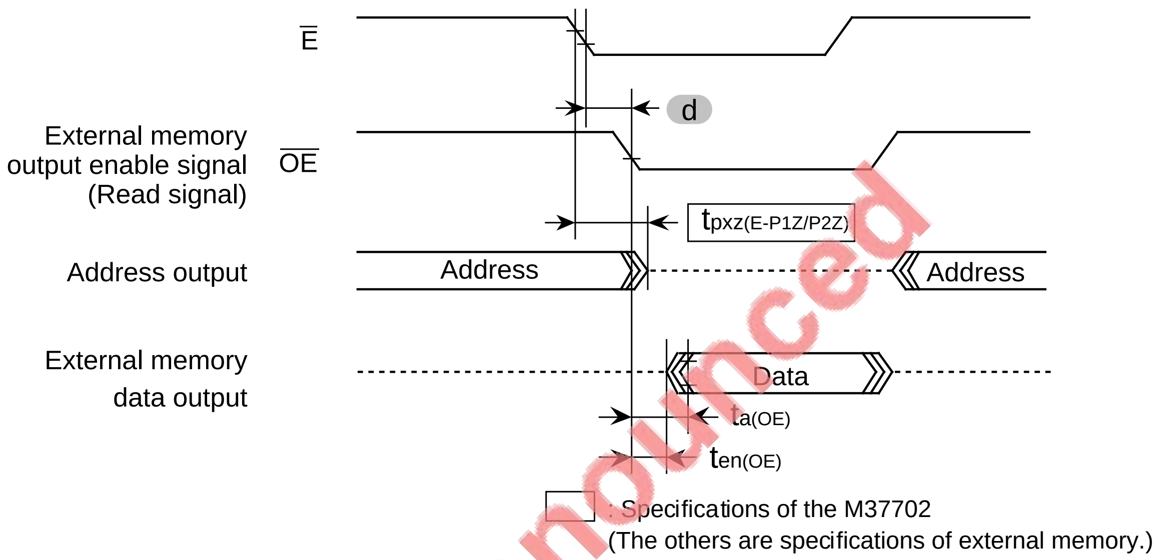


* This applies when using the 25 MHz version.

Fig. 17.1.9 Example of Ready signal generating circuit (Wait)

② **When using external memory that outputs data within $t_{pxz(E-P1Z/P2Z)}$ after falling edge of $\overline{E}$ signal**

Because the external memory outputs data within $t_{pxz(E-P1Z/P2Z)}$ after the falling edge of the $\overline{E}$ signal, there will be a possibility of the tail of address colliding with the head of data. In this case, generate the memory read signal ($\overline{OE}$) by delaying only the leading edge of the fall of the $\overline{E}$. (Refer to Figure 17.1.10.)



Note: Satisfy $t_{pxz(E-P1Z/P2Z)} \le t_{en(OE)}+d$.

If $t_{en(OE)} \le t_{pxz(E-P1Z/P2Z)}$ (= 5 ns), ensure a certain time (i.e., 'd' in this diagram) by delaying the falling edge of $\overline{OE}$ after the falling edge of $\overline{E}$.

Fig. 17.1.10 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 edge of $\overline{E}$ signal**

Because the external memory outputs data for more than $t_{pzx(E-P1Z/P2Z)}$ after the rising edge of the $\overline{E}$ signal, there will be a possibility of the tail of data colliding with the head of address. In this case, examine the method described below.

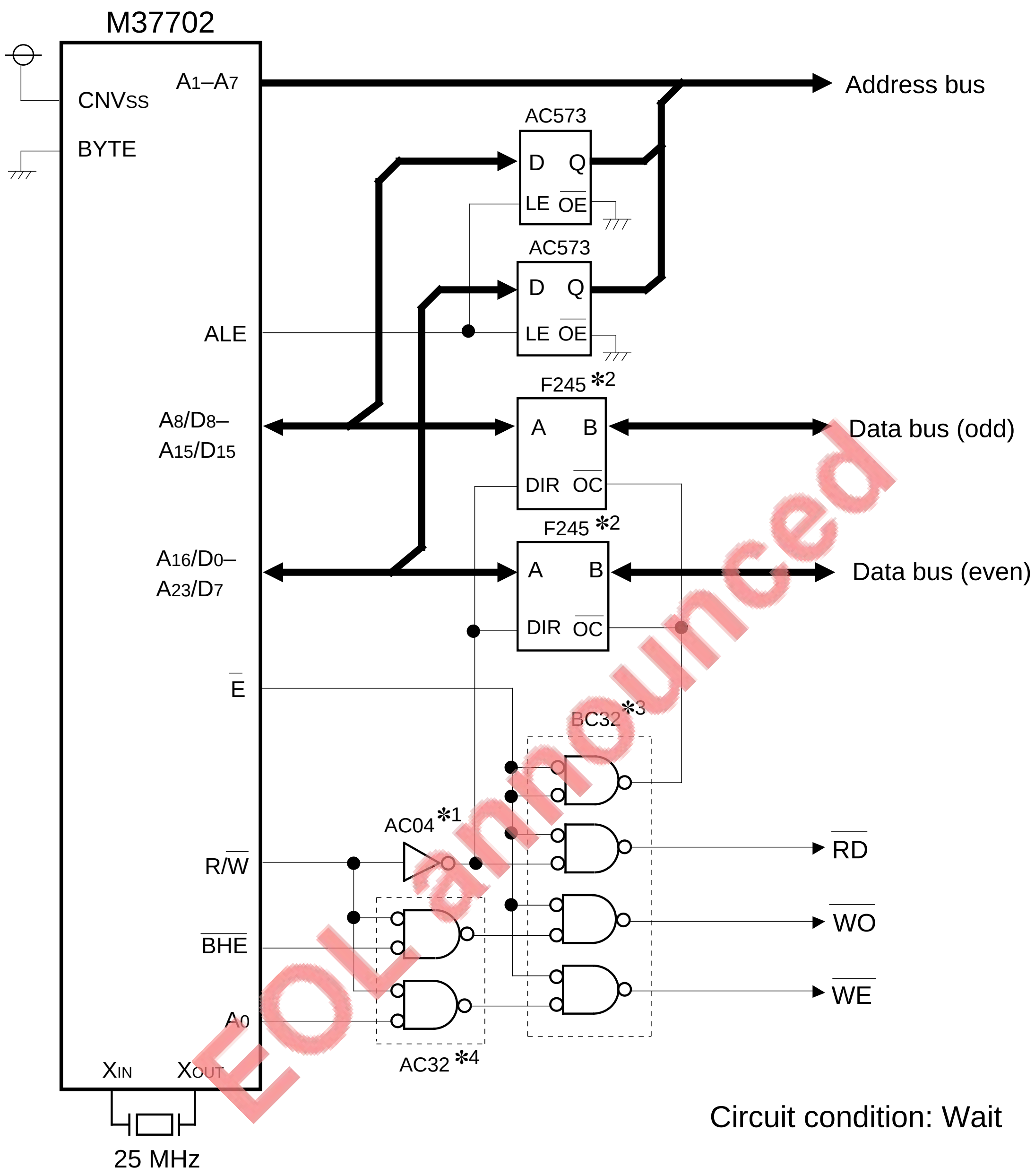
- Cut the tail of data output from the external memory by using a bus buffer and others.
- Use the Mitsubishi's memories that can be connected without a bus buffer.

Figures 17.1.11 to 17.1.14 show examples for how to use a bus buffer and the timing diagrams. Table 17.1.7 lists the memories that can be connected without a bus buffer. These memories do not require 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 the rising edge of $\overline{E}$ signal.)

Table 17.1.7 Memories that can be connected without bus buffer

Memory	Type description	$t_{DF}/t_{dis(OE)}$ (Maximum)	Conditions
EPROM	M5M27C256AK-85, -10, -12, -15	15 ns	$f(X_{IN}) \leq 20$ MHz
	M5M27C512AK-10, -12, -15	(Guaranteed by kit) **(Note)**	
	M5M27C100K-12. -15		
	M5M27C101K-12, -15		
	M5M27C102K-12, -15		
	M5M27C201K, JK-10, -12, -15		
	M5M27C202K, JK-10, -12, -15		
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		
Frash 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	8 ns	
	M5M5278CP, FP, J-20, -20L	10 ns	$f(X_{IN}) \leq 25$ MHz
	M5M5278CP, FP, J-25, -25L	6 ns	
	M5M5278DP, J-12	7 ns	
	M5M5278DP, FP, J-15, -15L	8 ns	
	M5M5278DP, FP, J-20, -20L		

Note: When the user needs a specification of the memories listed above, add the 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. 17.1.11 Example for using bus buffer (1)

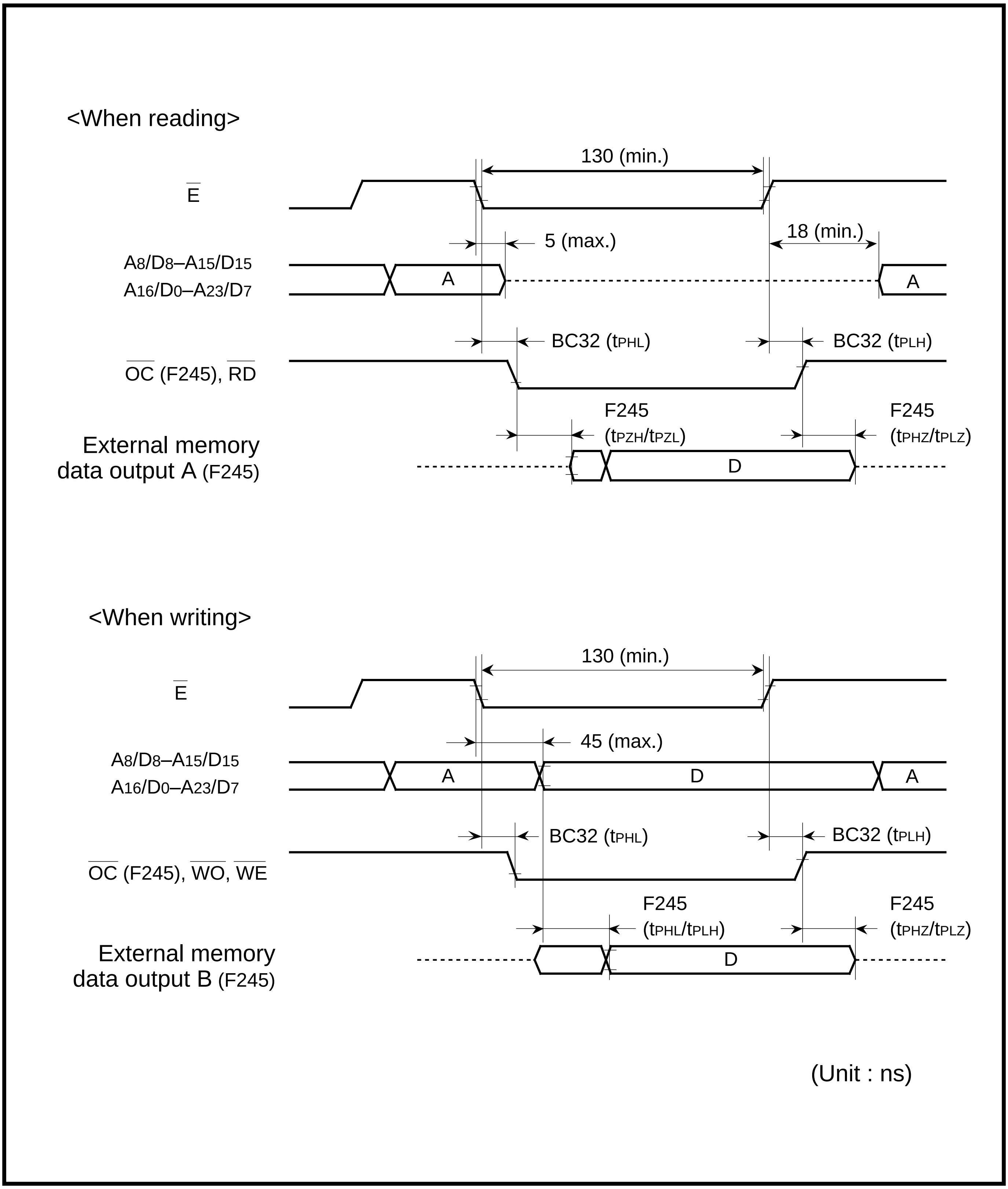
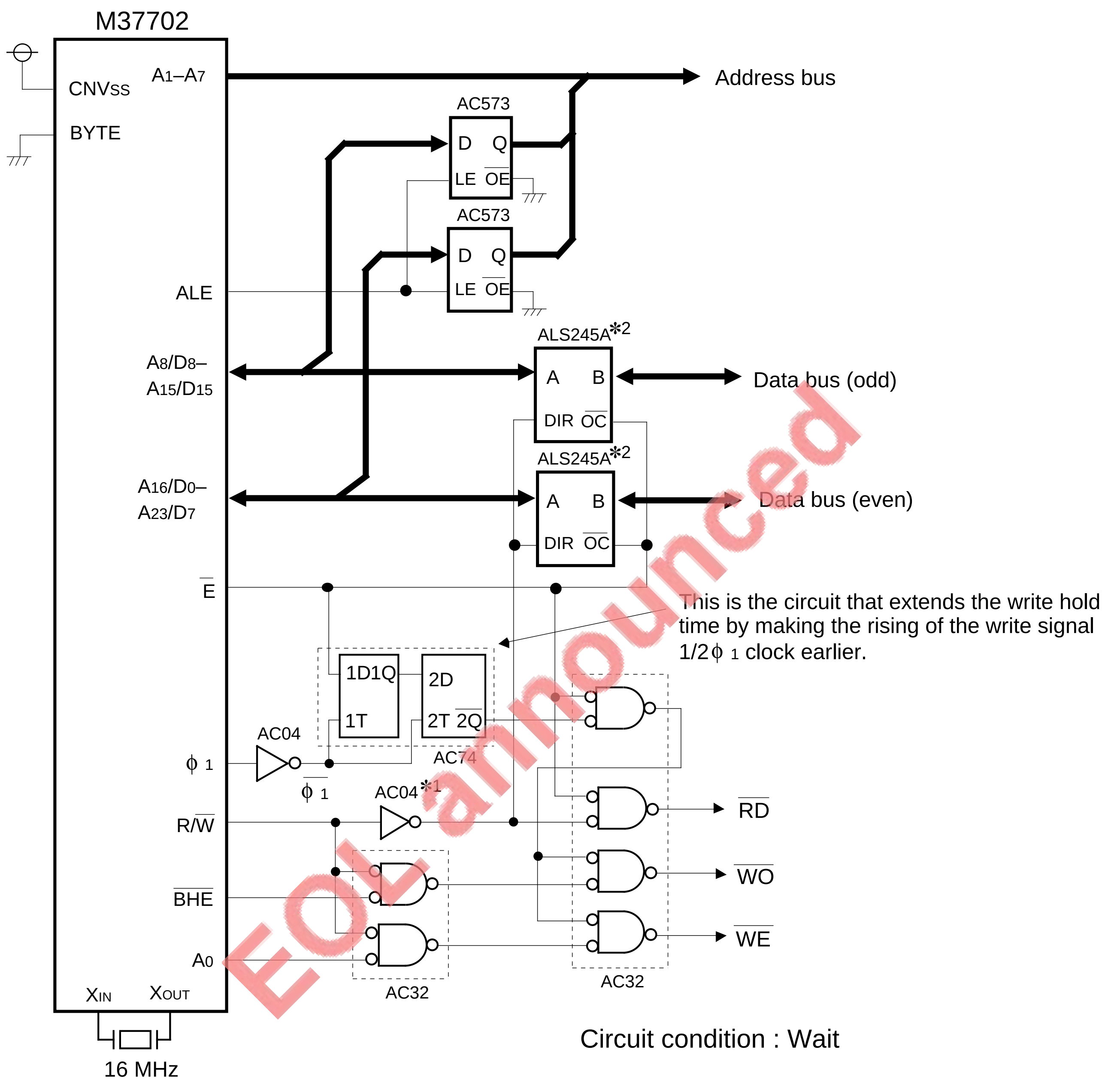


Fig. 17.1.12 Timing chart for sample circuit using bus buffers (1)



*1: Use the elements of which propagation delay time is within 30 ns.
*2: Use the elements of which output enable time is 5 ns or more and output disable time is within 36 ns.

Fig. 17.1.13 Example for using bus buffer (connecting with memory requiring a long hold time for write)

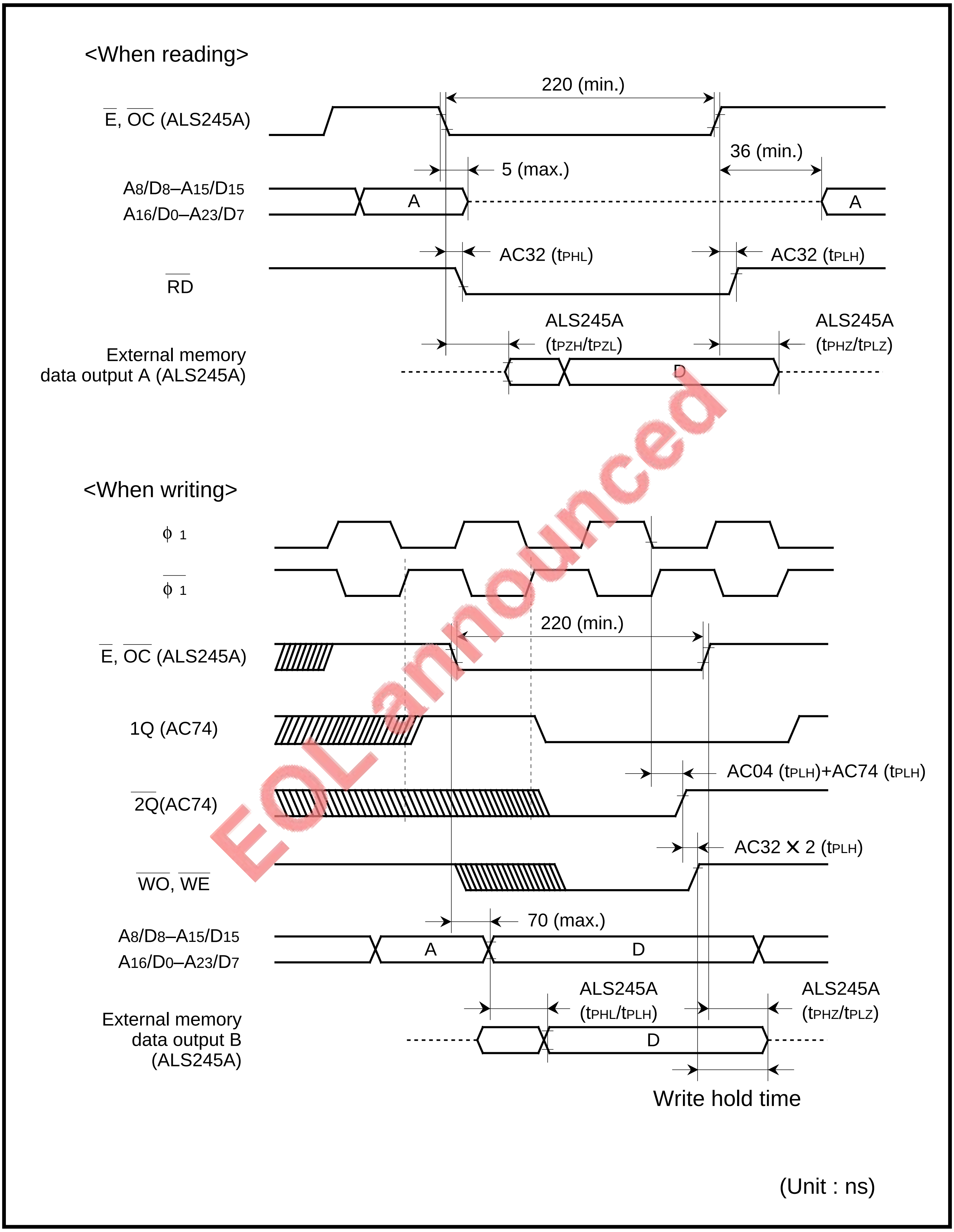


Fig. 17.1.14 Timing chart for sample circuit using bus buffers (2)

17.1.4 Example of memory expansion

(1) Example of SRAM expansion (minimum model)

Figure 17.1.15 shows a memory expansion example (minimum model) using a 32-Kbyte SRAM in the memory expansion mode. Figure 17.1.16 shows the timing chart for this example.

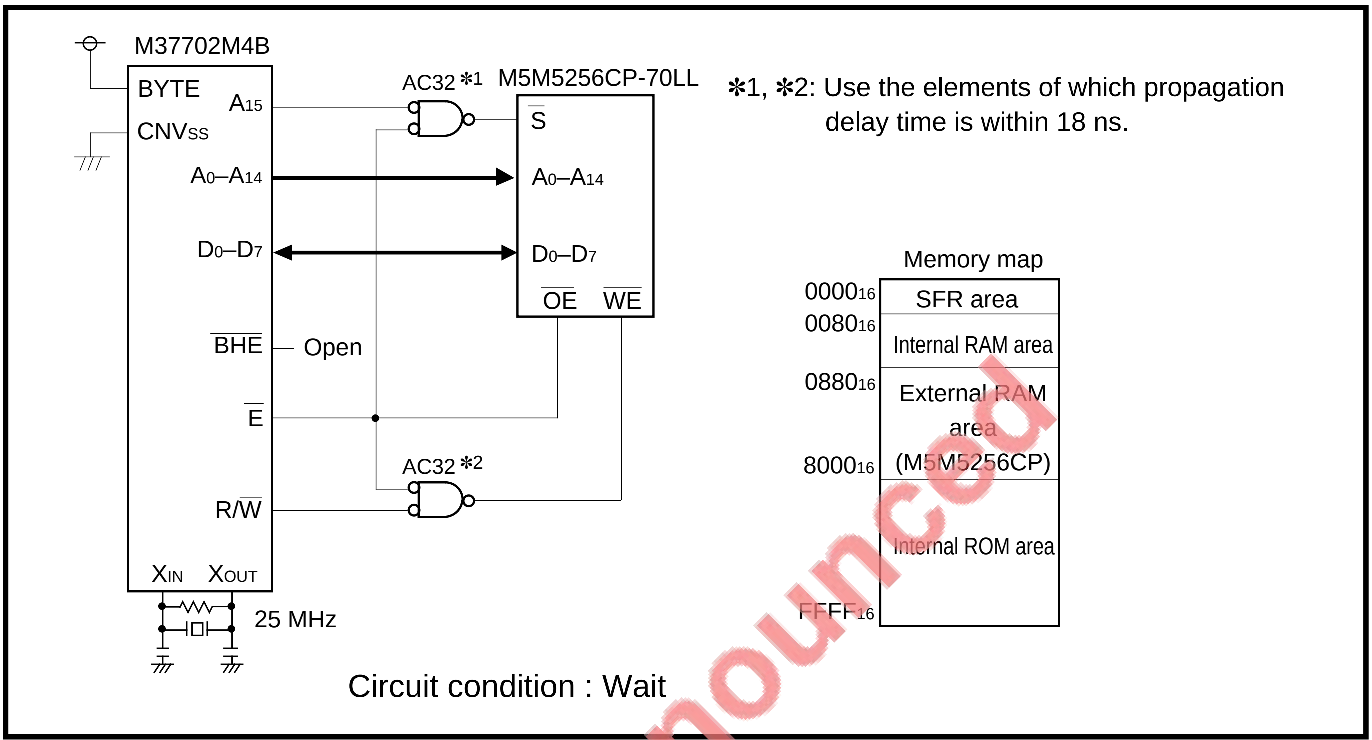


Fig. 17.1.15 Example of SRAM expansion (minimum model)

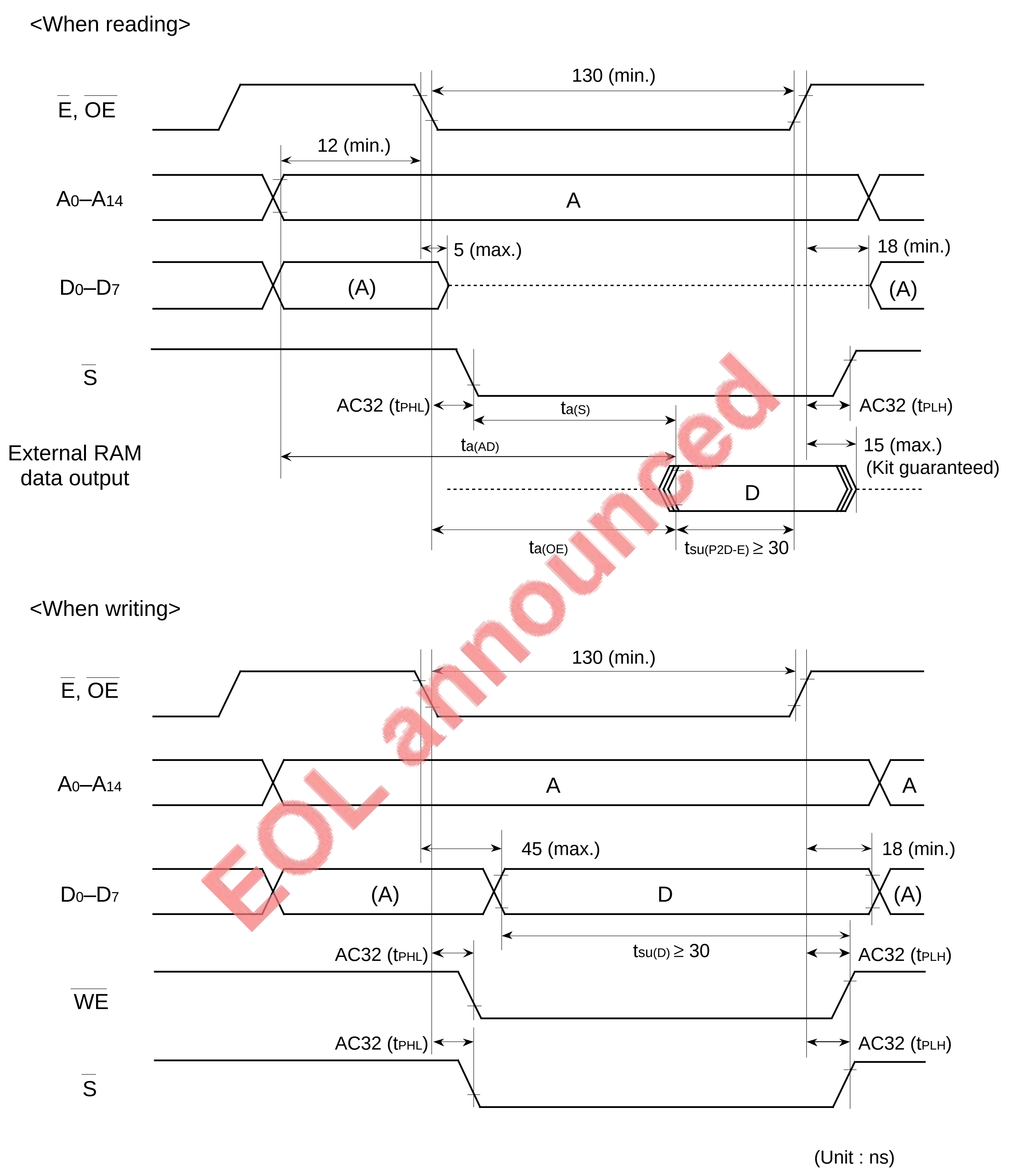


Fig. 17.1.16 Timing chart for SRAM expansion example (minimum model)

(2) Example of ROM expansion (maximum model)

Figure 17.1.17 shows a memory expansion example (maximum model) using a 2-Mbits ROM in the microprocessor mode. Figure 17.1.18 shows the timing chart for this example.

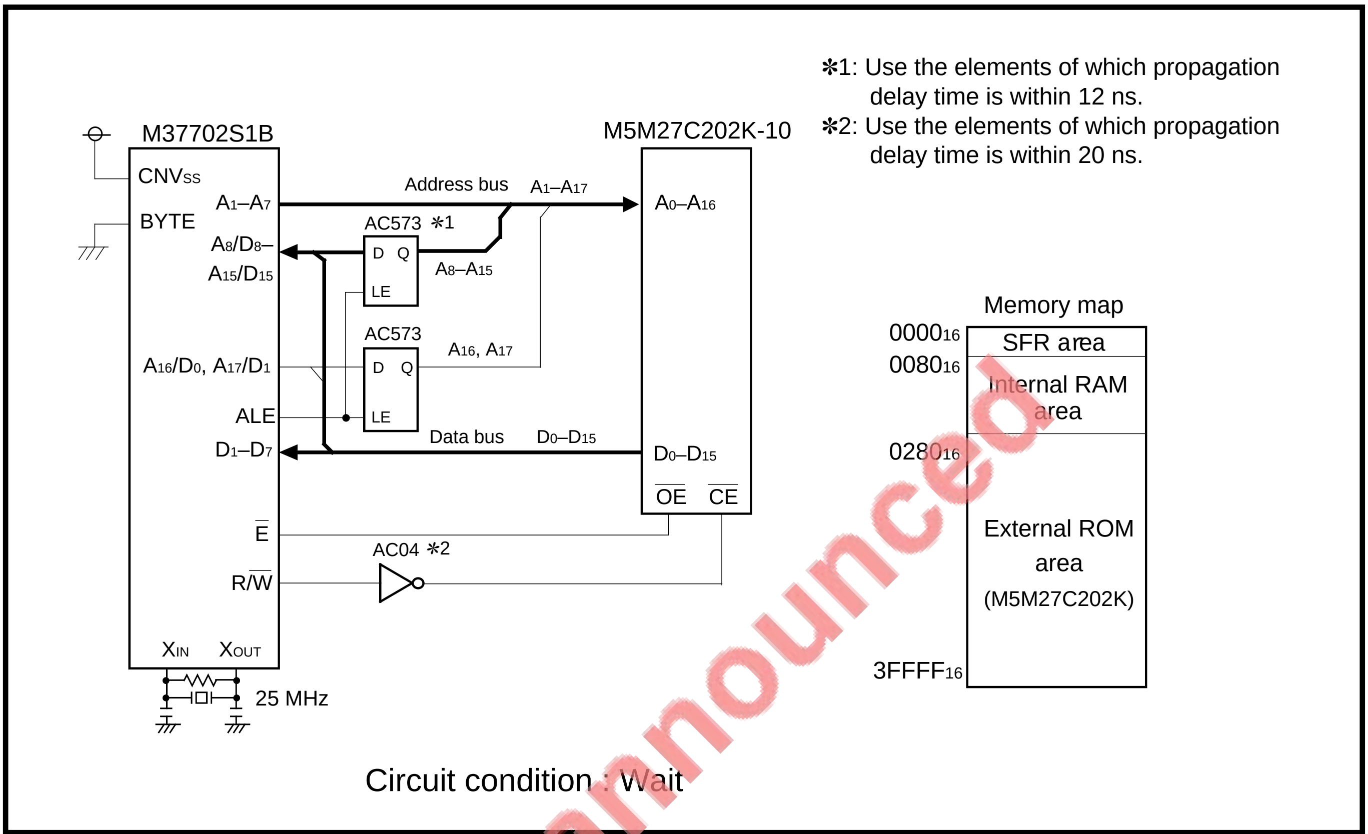


Fig. 17.1.17 Example of ROM expansion (maximum model)

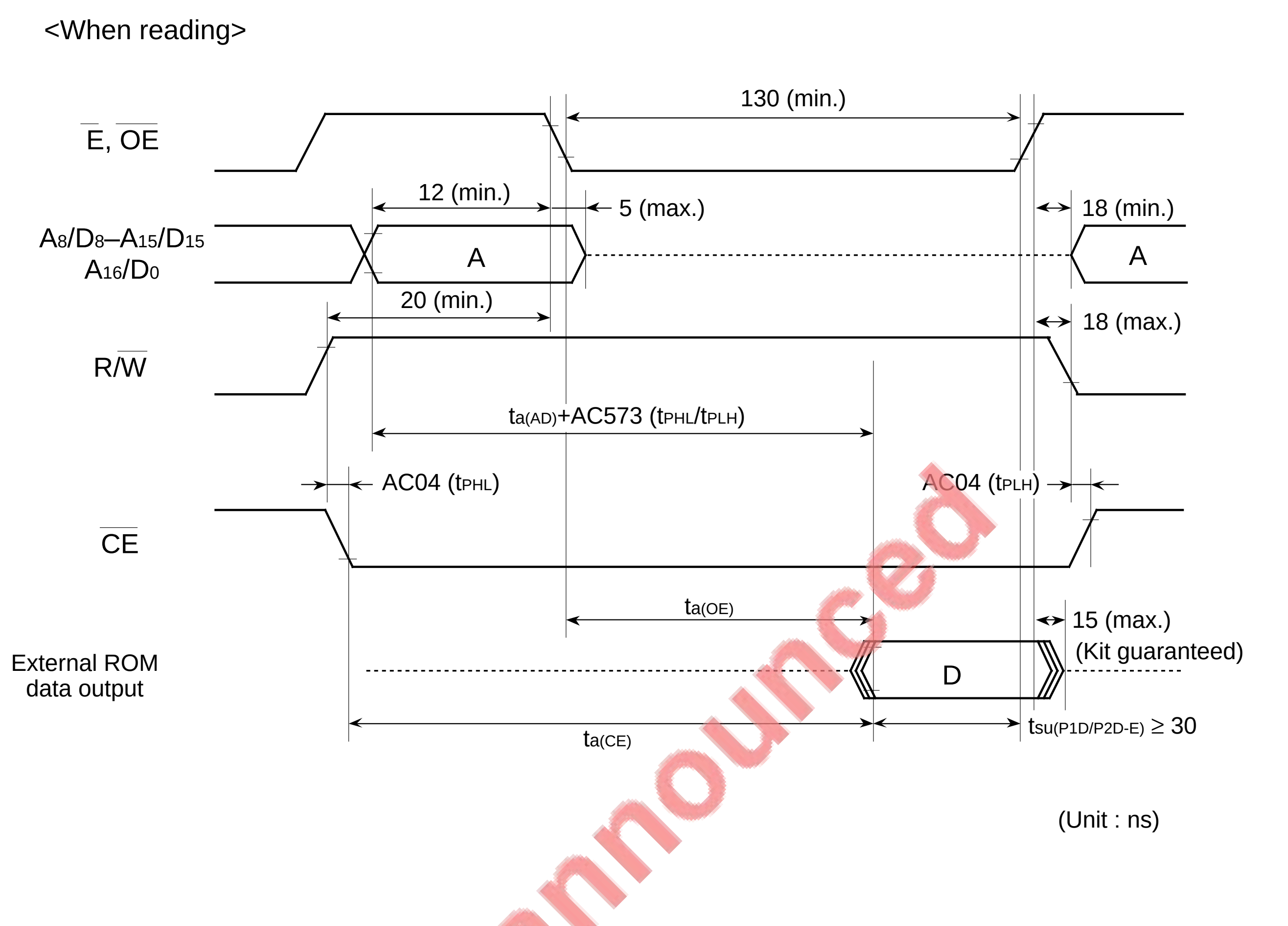


Fig. 17.1.18 Timing chart for ROM expansion example (maximum model)

17.1 Memory expansion

(3) Example of ROM and SRAM expansion (maximum model)

Figure 17.1.19 shows a memory expansion example (maximum model) using two 32-Kbyte ROM and two 32-Kbyte SRAM in the microprocessor mode. Figure 17.1.20 shows the timing diagram for this example.

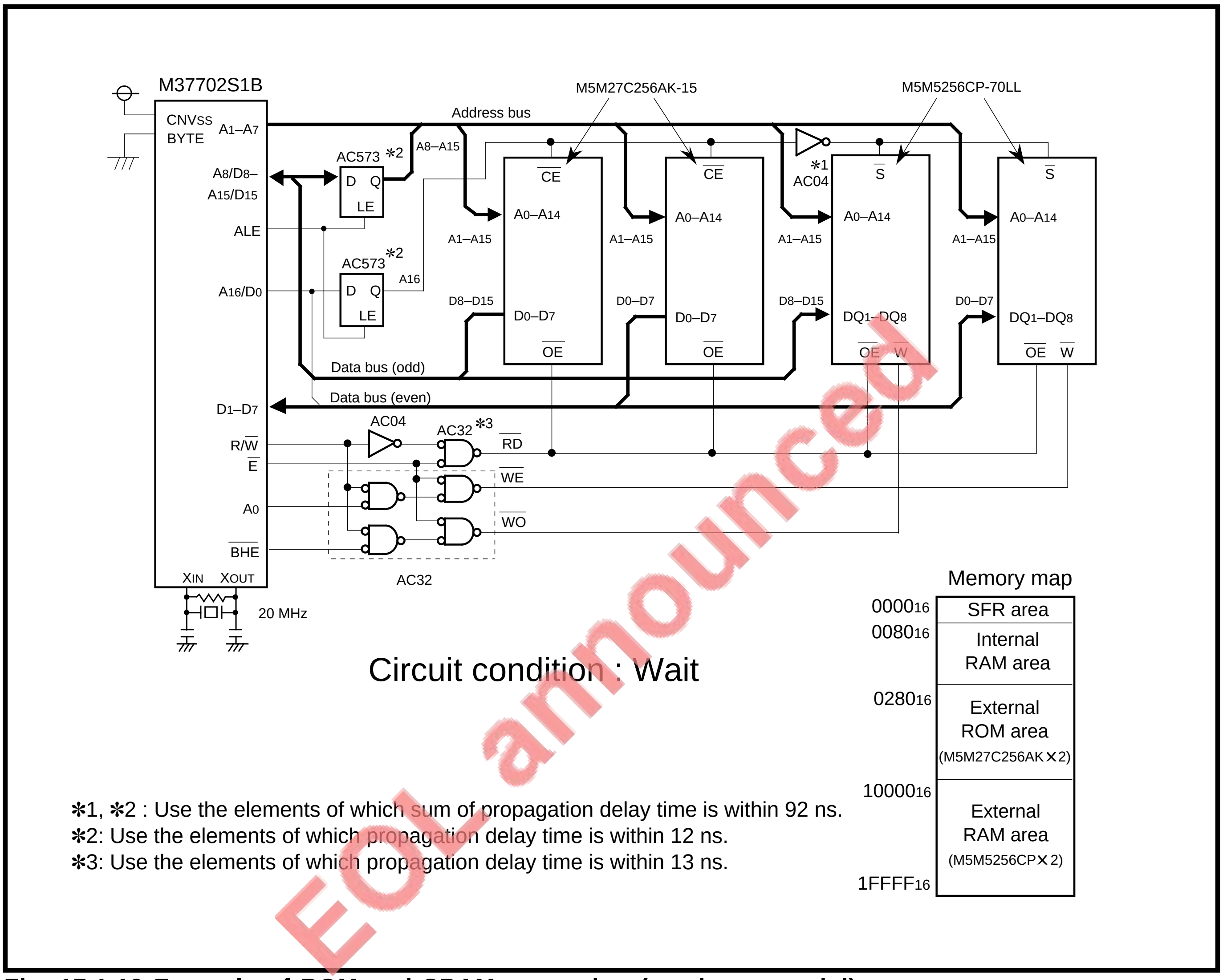


Fig. 17.1.19 Example of ROM and SRAM expansion (maximum model)

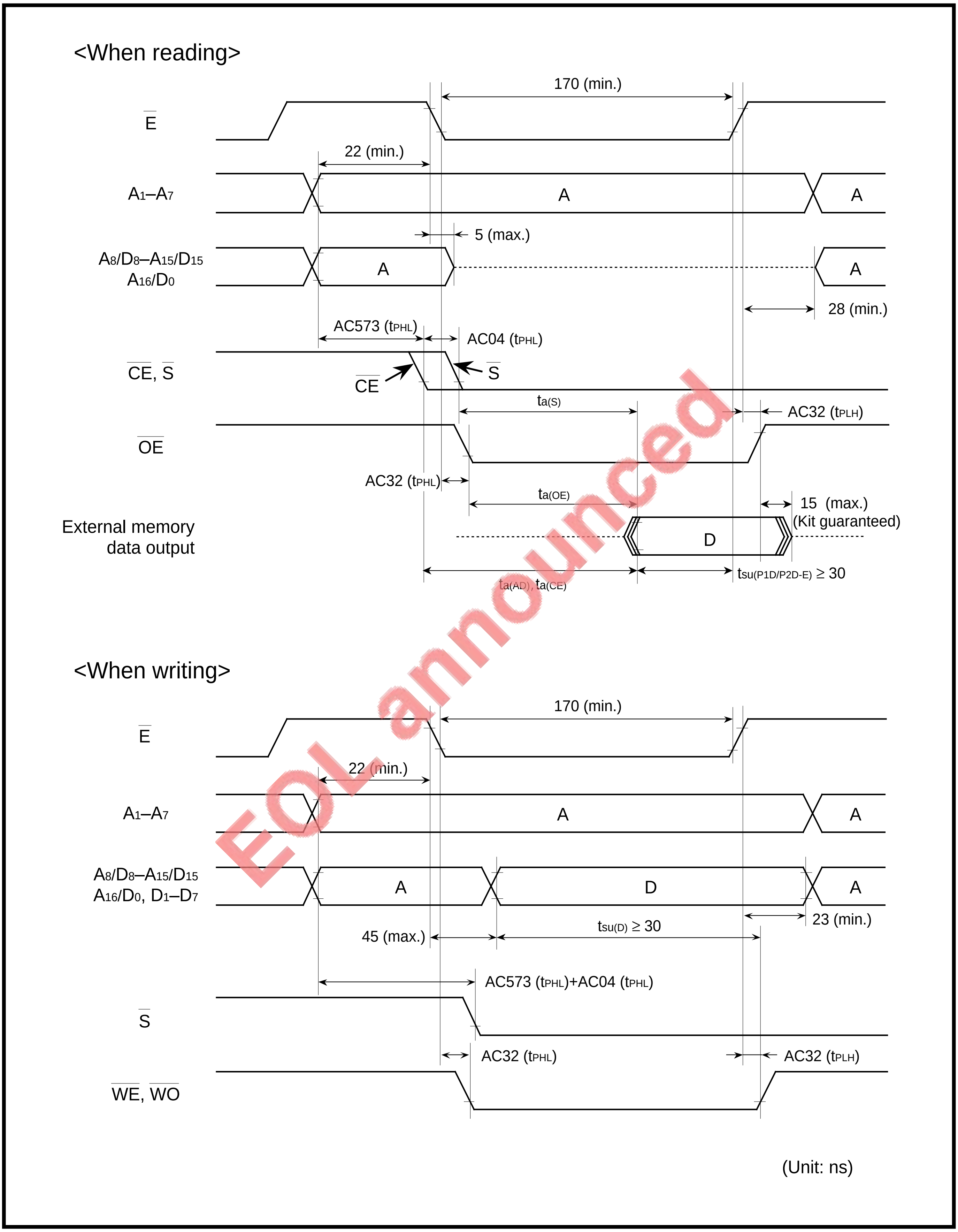


Fig. 17.1.20 Timing diagram for ROM and SRAM expansion example (maximum model)

17.1.5 Example of I/O expansion

(1) Example of port expansion circuit using M66010FP

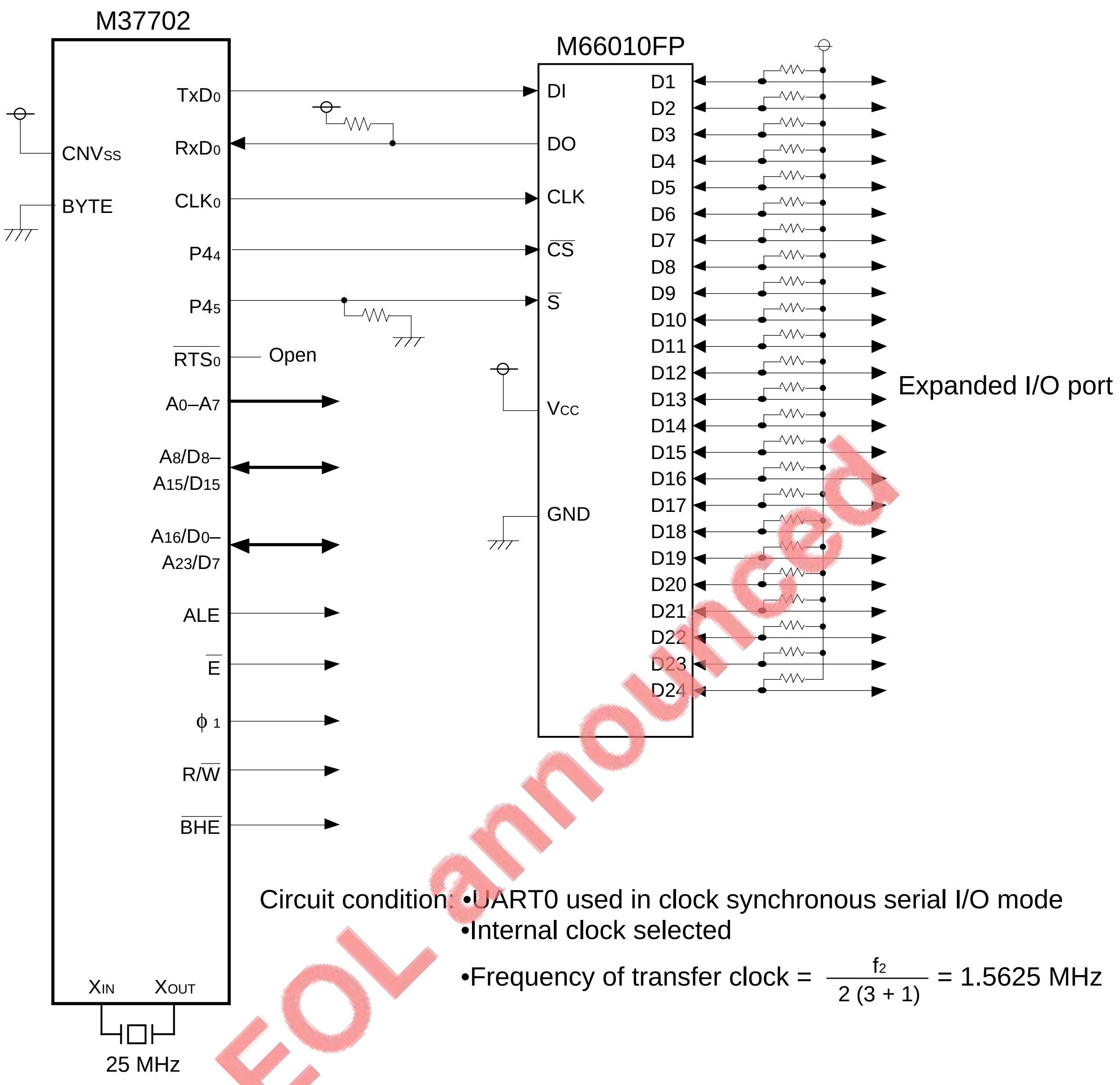
Figure 17.1.21 shows an example of a port expansion circuit using the M60010FP. Use 1.923 MHz or less frequency for Serial I/O transfer clock.
Serial I/O control in this expansion example is described below.
In this example, 8-bit data transmission/reception is performed 3 times by using UART0 and 24-bit port expansion is realized. Setting of UART0 is described below.

- Clock synchronous serial I/O mode; Transmission/Reception enable state
- Selected internal clock. Transfer clock frequency of 1.5625 MHz.
- LSB first

The control procedure 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 17.1.22 shows serial transfer timing between M37702 and M66010FP.



Circuit condition: •UART0 used in clock synchronous serial I/O mode
•Internal clock selected
•Frequency of transfer clock = $\frac{f_2}{2\ (3 + 1)}$ = 1.5625 MHz

Fig. 17.1.21 Example of port expansion circuit using M60010FP

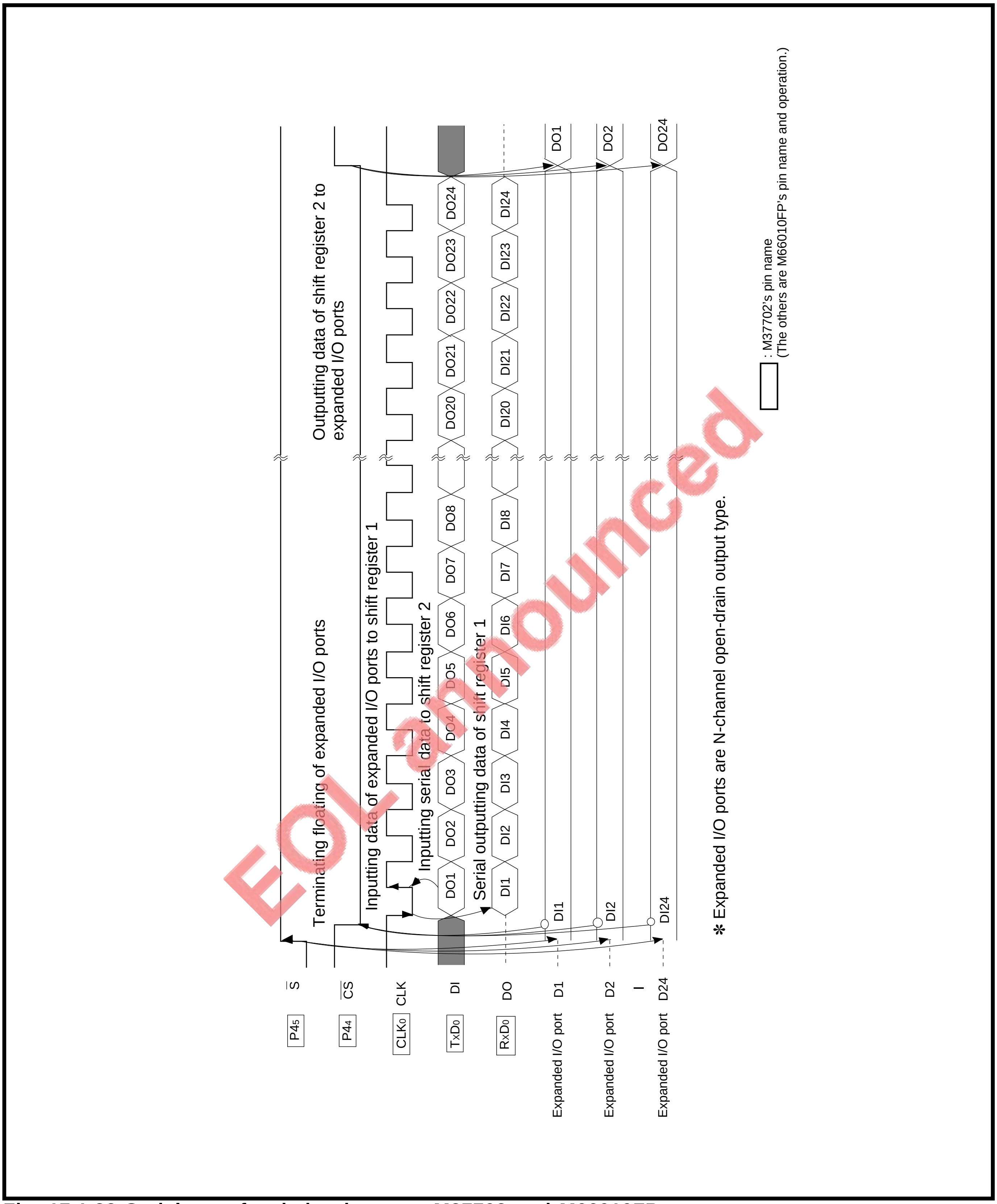


Fig. 17.1.22 Serial transfer timing between M37702 and M66010FP

17.2 Sample program execution rate comparison

Sample program execution rates are compared in this paragraph.
The execution time ratio depends on the program or the usage conditions.

17.2.1 Difference depending on data bus width and software Wait

Internal areas are always accessed at 16-bit data bus width and without software Wait. In the external areas, the external data bus width and software Wait are selectable. Table 17.2.1 lists the sample program (refer to Figure 17.2.1) execution time ratio depending on these selection and used memory areas.

Table 17.2.1 Sample program execution time ratio (external data bus width and software Wait)

Memory area		External data bus width (bit)	Software Wait	Sample program execution time ratio	
RAM	ROM			Sample A	Sample B
Internal	Internal	(16)	(Nothing)	1.00	1.00
Internal	External	16	Nothing	1.00	1.00
			Inserted	1.17	1.10
		8	Nothing	1.19	1.08
			Inserted	1.67	1.46
External	External	16	Nothing	1.00	1.00
			Inserted	1.25	1.17
		8	Nothing	1.19	1.13
			Inserted	1.78	1.65
Calculation value*				0.92	0.90

Calculation value* : The value is calculated from the shortest execution cycle number of each instruction described in the software manual.

17.2 Sample program execution rate comparison

Sample A

```
        SEP      M,X
        LDA.B    A,#0
        STA      A,DEST+64
        STA      A,DEST+65
        STA      A,DEST+66
        LDX.B    #63
ITALIC: LDA      A,SOUR,X
        TAY
        AND.B    A,#00000011B
        STA      A,DEST,X
        TYA
        AND.B    A,#00001100B
        ORA      A,DEST+1,X
        STA      A,DEST+1,X
        TYA
        AND.B    A,#00110000B
        ORA      A,DEST+2,X
        STA      A,DEST+2,X
        TYA
        AND.B    A,#11000000B
        ORA      A,DEST+3,X
        STA      A,DEST+3,X
        DEX
        BPL      ITALIC
```

Sample B

```
        SEP      X
        CLM
        .DATA    16
        .INDEX   8
        LDY      #69
LOOP0:  LDX      #69
LOOP1:  ASL      SOUR,X
        SEM
        .DATA    8
        ROL      SOUR+2,X
        ROL      B
        CLM
        .DATA    16
        ROR      A
        DEX
        DEX
        DEX
        BNE      LOOP1
        STA      A,DEST,Y
        SEM
        .DATA    8
        STA      B,DEST+2,Y
        CLM
        .DATA    16
        DEY
        DEY
        DEY
        BNE      LOOP0
```

✻ SOUR, DEST : Work area
(Direct page area : Access this area at the following mode.)
- Direct addressing mode
- Direct Indexed X addressing mode
- Absolute Indexed Y addressing mode

Fig. 17.2.1 Sample program list

17.2.2 Comparison software Wait ($f(X_{IN})$ = 20 MHz) with software Wait + Ready ($f(X_{IN})$ = 25 MHz)

The following condiitons ① and ② are compared. Refer to Figure 17.2.1 about executed sample program. The execution time ratio depends on the program or the usage conditions.

Condition ① : When selecting software Wait and $f(X_{IN})$ = 20 MHz
Condition ② : When selecting software Wait and $f(X_{IN})$ = 25 MHz and inserting a Wait which is 1 cycle of ϕ (inserting total Wait of 2 cycles of ϕ).

Table 17.2.2 Comparison condition

Item	Condition ①	Condition ②
Processor mode	Microprocessor mode	Microprocessor mode
$f(X_{IN})$	20 MHz	25 MHz
External data bus width	16 bits	16 bits
Software Wait	Inserted	Inserted
Ready	Invalid	Valid only to external EPROM areas
Program area	External EPROM	External EPROM
Work area	Internal or External SRAM	Internal or External SRAM

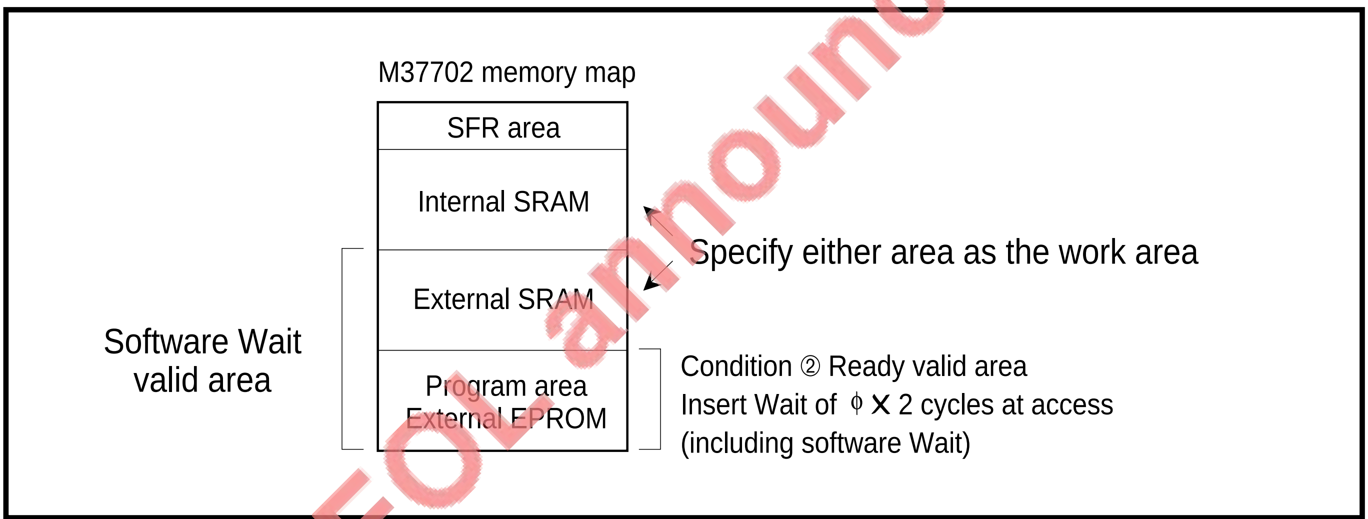


Fig. 17.2.2 Memory allocation at execution rate comparison

17.2 Sample program execution rate comparison

Figure 17.2.3 shows that there is almost no difference between conditions ① and ② about the execution time.

The bus buffers become unnecessary by using the specific memory. (See Table 17.1.7.) Consequently, the case selecting $f(X_{IN})$ = 20 MHz and inserting software Wait is superior in the cost performance.

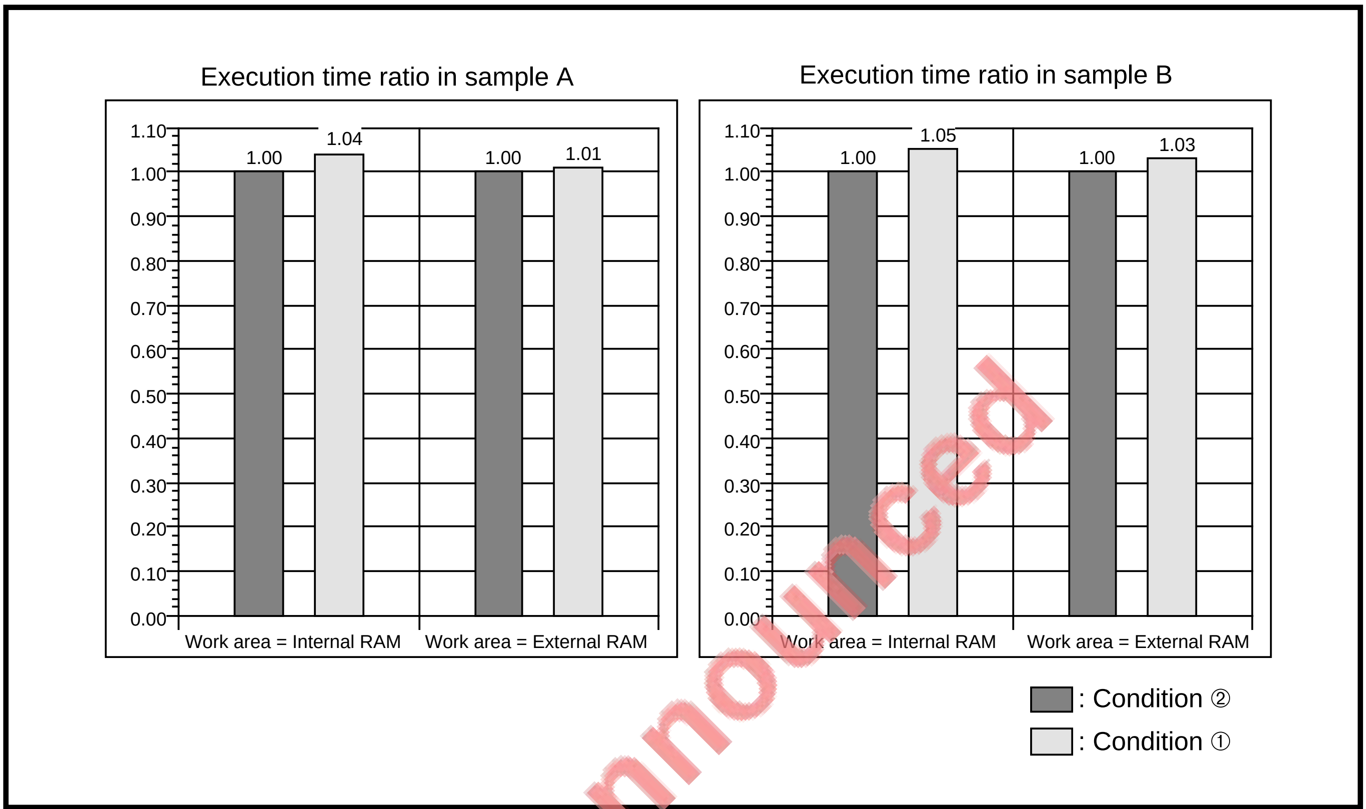


Fig. 17.2.3 Execution time ratio

CHAPTER 18

LOW VOLTAGE VERSION

LOW VOLTAGE VERSION

The low voltage version has the following characteristics:

- Low power source voltage (2.7 to 5.5 V)
- Wide operating temperature range (–40 to 85 °C)

The low voltage version is suitable to control equipment which is required to process a large amount of data with a low power dissipation, for example portable equipment which is driven by a battery and OA equipment.

Differences between the M37702M2LXXXGP and the M37702M2BXXXFP are mainly described below.
For the EPROM mode of the PROM version, refer to **"Chapter 19. PROM VERSION."**

18.1 Performance overview

Table 18.1.1 shows the performance overview of the M37702M2LXXXGP.

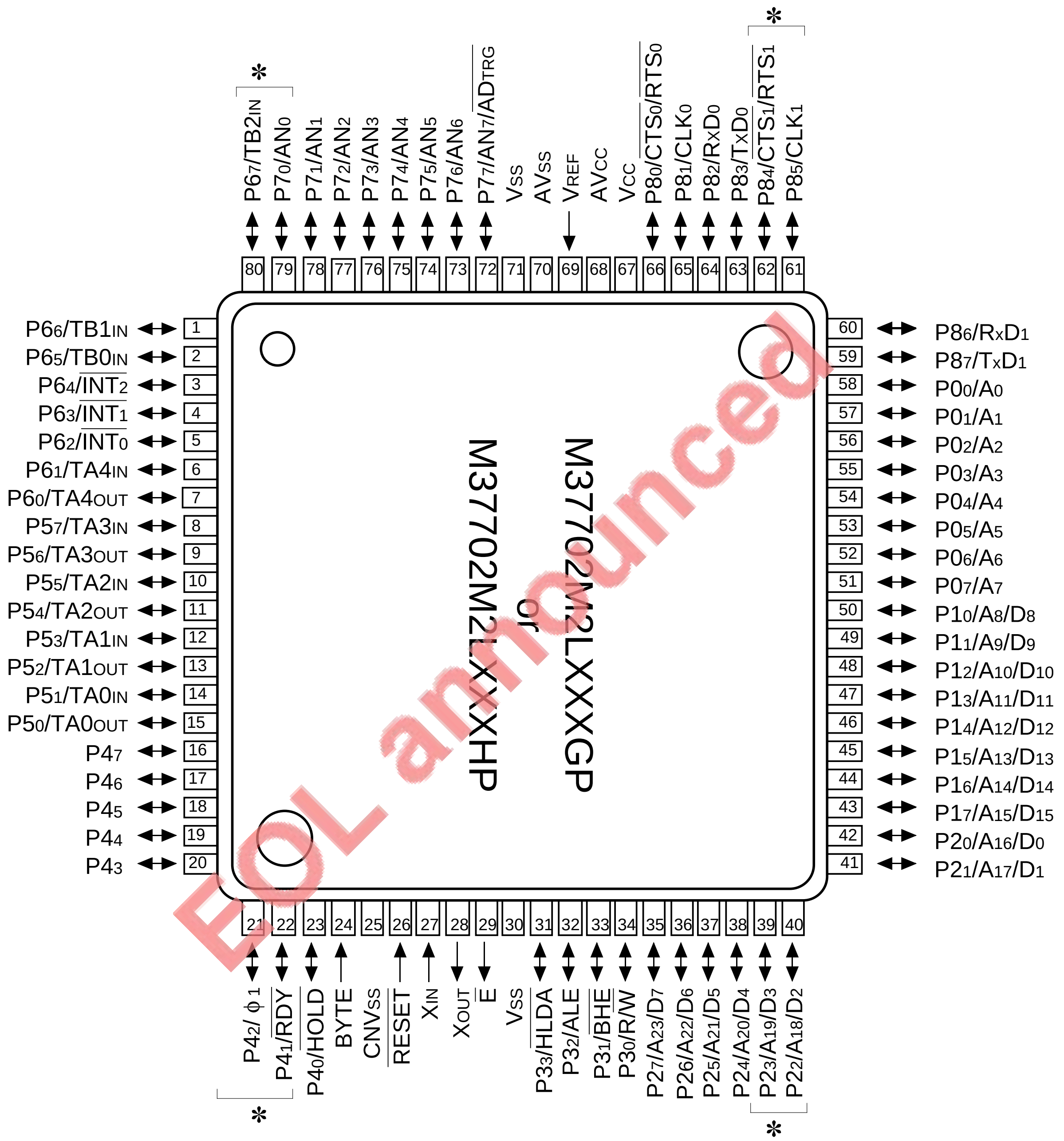
Table 18.1.1 M37702M2LXXXGP performance overview

Parameters		Functions
Number of basic instructions		103
Instruction execution time		500 ns (the minimum instruction at f(X_{IN}) = 8 MHz)
External clock input frequency f(X_{IN})		8 MHz (maximum)
Memory size	ROM	16384 bytes
	RAM	512 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		8-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		2.7 – 5.5 V
Power dissipation		12 mW (at supply voltage = 3 V, f(X_{IN}) = 8 MHz frequency)
		30 mW (at supply voltage = 5 V, f(X_{IN}) = 8 MHz frequency)
Port Input/Output characteristics	Input/Output withstand voltage	5 V
	Output current	5 mA
Memory expansion		Maximum 16 Mbytes
Operating temperature range		–40°C to 85°C
Device structure		CMOS high-performance silicon gate process
Package		80-pin plastic molded QFP

Note: Low voltage versions except the M37702M2LXXXGP are the same except for the package type, memory type, and memory size.

18.2 Pin configuration

Figure 18.2.1 shows the M37702M2LXXXGP and the M37702M2LXXXHP pin configuration. Figure 18.2.2 shows the M37702M4LXXXFP pin configuration.



Outline M37702M2LXXXGP •••••• 80P6S-A
Outline M37702M2LXXXHP •••••• 80P6D-A

✻ : The M37702M2LXXXGP and the M37702M2LXXXHP have the pin configuration shifted to 2 pins assignment from the M37702M2BXXXFP.

Fig. 18.2.1 M37702M2LXXXGP and M37702M2LXXXHP pin configuration (top view)

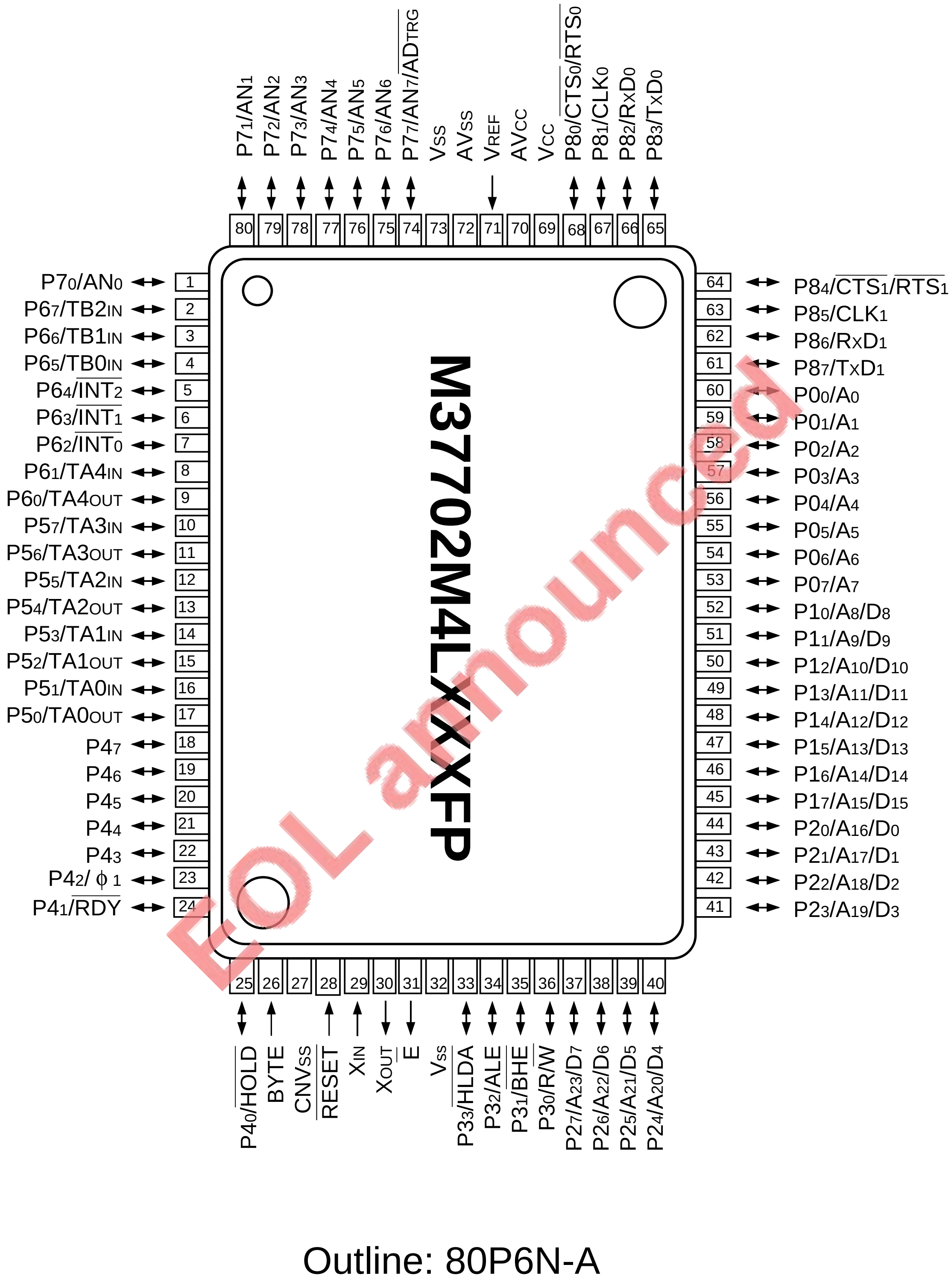


Fig. 18.2.2 M37702M4LXXXFP pin configuration (top view)

18.3 Functional description

The M37702M2LXXXGP has the same functions as the M37702M2BXXXFP except for the power-on reset conditions. Power-on reset conditions are described below.
For the other functions, refer to chapters **"2. CENTRAL PROCESSING UNIT"** to **"14. CLOCK GENERATING CIRCUIT."**

18.3.1 Power-on reset conditions

Figure 18.3.1 shows the power-on reset conditions and Figure 18.3.2 shows an example of power-on reset circuit. For details of reset, refer to **"Chapter 13. RESET."**

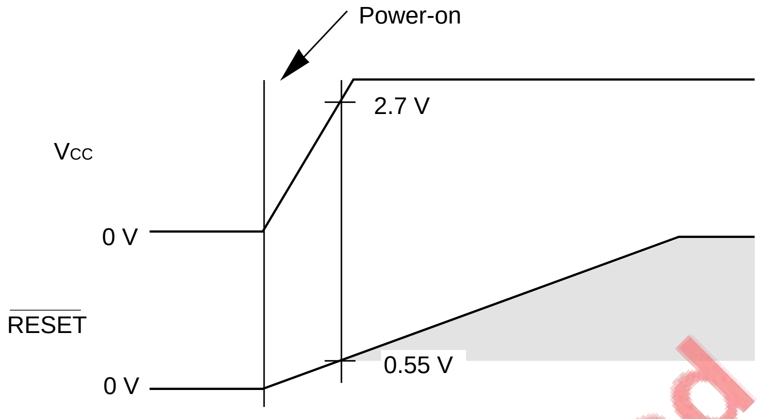


Fig. 18.3.1 Power-on reset conditions

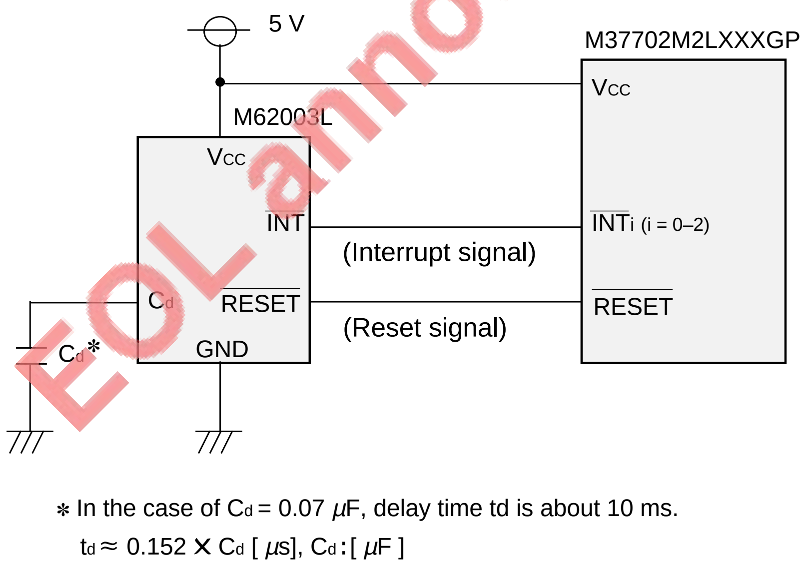


Fig. 18.3.2 Example of power-on reset circuit

18.4 Electrical characteristics

The electrical characteristics of M37702M2LXXXGP and M37702M2LXXXHP is described below. For the latest data, inquire of addresses described last (**“CONTACT ADDRESSES FOR FURTHER INFORMATION”)**.

18.4.1 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 Vcc+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 Vcc+0.3	V
P_d	Power dissipation	M37702M2LXXXGP	Ta = 25 °C	300	mW
		M37702M2LXXXHP	Ta = 25 °C	200	
T_{opr}	Operating temperature			–40 to 85	°C
T_{stg}	Storage temperature			–65 to 150	°C

18.4.2 Recommended operating conditions

Recommended operating conditions (V_{CC} = 2.7 – 5.5 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{CC}	Power source voltage		2.7		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	$0.8V_{CC}$		V_{CC}	V
V_{IH}	High-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (in single-chip mode)	$0.8V_{CC}$		V_{CC}	V
V_{IH}	High-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (in memory expansion mode and microprocessor mode)	$0.5V_{CC}$		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		$0.2V_{CC}$	V
V_{IL}	Low-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (in single-chip mode)	0		$0.2V_{CC}$	V
V_{IL}	Low-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (in memory expansion mode and microprocessor mode)	0		$0.16V_{CC}$	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	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_3$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			10	mA
$I_{OL (avg)}$	Low-level average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_3$, $P5_4$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			5	mA
$f(X_{IN})$	External clock input frequency				8	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.

LOW VOLTAGE VERSION

18.4 Electrical characteristics

18.4.3 Electrical characteristics

Electrical characteristics (V_{CC} = 5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	High-level output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	V_{CC} = 5 V, I_{OH} = –10 mA	3			V
			V_{CC} = 3 V, I_{OH} = –1 mA	2.5			V
V_{OH}	High-level output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_3$	V_{CC} = 5 V, I_{OH} = –400 μA	4.7			V
V_{OH}	High-level output voltage	$P3_2$	V_{CC} = 5 V, I_{OH} = –10 mA	3.1			V
			V_{CC} = 5 V, I_{OH} = –400 μA	4.8			
			V_{CC} = 3 V, I_{OH} = –1mA	2.6			
V_{OH}	High-level output voltage	$\overline{E}$	V_{CC} = 5 V, I_{OH} = –10 mA	3.4			V
			V_{CC} = 5 V, I_{OH} = –400 μA	4.8			
			V_{CC} = 3 V, I_{OH} = –1mA	2.6			
V_{OL}	Low-level output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	V_{CC} = 5 V, I_{OL} = 10 mA			2	V
			V_{CC} = 3 V, I_{OL} = 1 mA			0.5	
V_{OL}	Low-level output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_3$	V_{CC} = 5 V, I_{OL} = 2 mA			0.45	V
V_{OL}	Low-level output voltage	$P3_2$	V_{CC} = 5 V, I_{OL} = 10 mA			1.9	V
			V_{CC} = 5 V, I_{OL} = 2 mA			0.43	
			V_{CC} = 3 V, I_{OL} = 1 mA			0.4	
V_{OL}	Low-level output voltage	$\overline{E}$	V_{CC} = 5 V, I_{OL} = 10 mA			1.6	V
			V_{CC} = 5 V, I_{OL} = 2 mA			0.4	
			V_{CC} = 3 V, I_{OL} = 1 mA			0.4	
V_{T+}–V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$–$TA4_{IN}$, $TB0_{IN}$–$TB2_{IN}$, $\overline{INT_0}$–$\overline{INT_2}$, AD_{TRG}, CTS_0, CTS_1, CLK_0, CLK_1	V_{CC} = 5 V	0.4		1	V
			V_{CC} = 3 V	0.1		0.7	
V_{T+}–V_{T-}	Hysteresis	$\overline{RESET}$	V_{CC} = 5 V	0.2		0.5	V
			V_{CC} = 3 V	0.1		0.4	
V_{T+}–V_{T-}	Hysteresis	X_{IN}	V_{CC} = 5 V	0.1		0.3	V
			V_{CC} = 3 V	0.06		0.2	
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}, RESET, CNV_{SS}, BYTE	V_{CC} = 5 V, V_I = 5 V			5	μA
			V_{CC} = 3 V, V_I = 3 V			4	
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_{CC} = 5 V, V_I = 0 V			–5	μA
			V_{CC} = 3 V, V_I = 0 V			–4	
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}) = 8 MHz, V_{CC} = 5 V		6	12	mA
			In single-chip mode, output pins are open, and the other pins are connected to V_{SS}. f(X_{IN}) = 8 MHz, V_{CC} = 3 V		4	8	
			In single-chip mode, output pins are open, and the other pins are connected to V_{SS}. Ta = 25 °c, when clock is stopped			1	μA
			In single-chip mode, output pins are open, and the other pins are connected to V_{SS}. Ta = 85 °c, when clock is stopped			20	μA

18.4.4 A-D converter characteristics

A-D CONVERTER CHARACTERISTICS (V_{CC} = AV_{CC} = 2.7 – 5.5 V, V_{SS} = AV_{SS} = 0 V, Ta = –40 to 85 °C, f(X_{IN}) = 8 MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	V_{REF} = V_{CC}			8	Bits
—	Absolute accuracy	V_{REF} = V_{CC}			±3	LSB
R_{LADDER}	Ladder resistance	V_{REF} = V_{CC}	2		10	kΩ
t_{CONV}	Conversion time		28.5			µs
V_{REF}	Reference voltage		2.7		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

LOW VOLTAGE VERSION

18.4 Electrical characteristics

18.4.5 Internal peripheral devices

Timing requirements (V_{CC} = 2.7 – 5.5 V, V_{SS} = 0 V, Ta = –40 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	250		ns
$t_{w(TAH)}$	TAi_{IN} input high-level pulse width	125		ns
$t_{w(TAL)}$	TAi_{IN} input low-level pulse width	125		ns

Timer A input (gating input in timer mode)

Symbol	Parameter	Data formula (minimum)	Limits Min.	Limits Max.	Unit
$t_{c(TA)}$	TAi_{IN} input cycle time	$\frac{8 \times 10^9}{f(X_{IN})}$	1000		ns
$t_{w(TAH)}$	TAi_{IN} input high-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})}$	500		ns
$t_{w(TAL)}$	TAi_{IN} input low-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})}$	500		ns

Note: 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.

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

Symbol	Parameter	Data formula (minimum)	Limits Min.	Limits Max.	Unit
$t_{c(TA)}$	TAi_{IN} input cycle time	$\frac{4 \times 10^9}{f(X_{IN})}$	500		ns
$t_{w(TAH)}$	TAi_{IN} input high-level pulse width		250		ns
$t_{w(TAL)}$	TAi_{IN} input low-level pulse width		250		ns

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

Symbol	Parameter	Limits Min.	Limits Max.	Unit
$t_{w(TAH)}$	TAi_{IN} input high-level pulse width	250		ns
$t_{w(TAL)}$	TAi_{IN} input low-level pulse width	250		ns

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

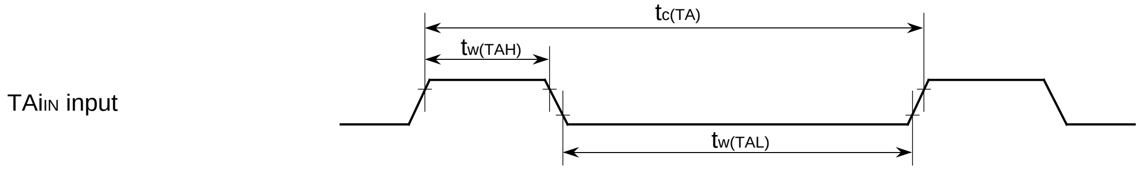
Symbol	Parameter	Limits Min.	Limits Max.	Unit
$t_{c(UP)}$	TAi_{OUT} input cycle time	5000		ns
$t_{w(UPH)}$	TAi_{OUT} input high-level pulse width	2500		ns
$t_{w(UPL)}$	TAi_{OUT} input low-level pulse width	2500		ns
$t_{su(UP-T_{IN})}$	TAi_{OUT} input setup time	1000		ns
$t_{h(T_{IN}-UP)}$	TAi_{OUT} input hold time	1000		ns

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

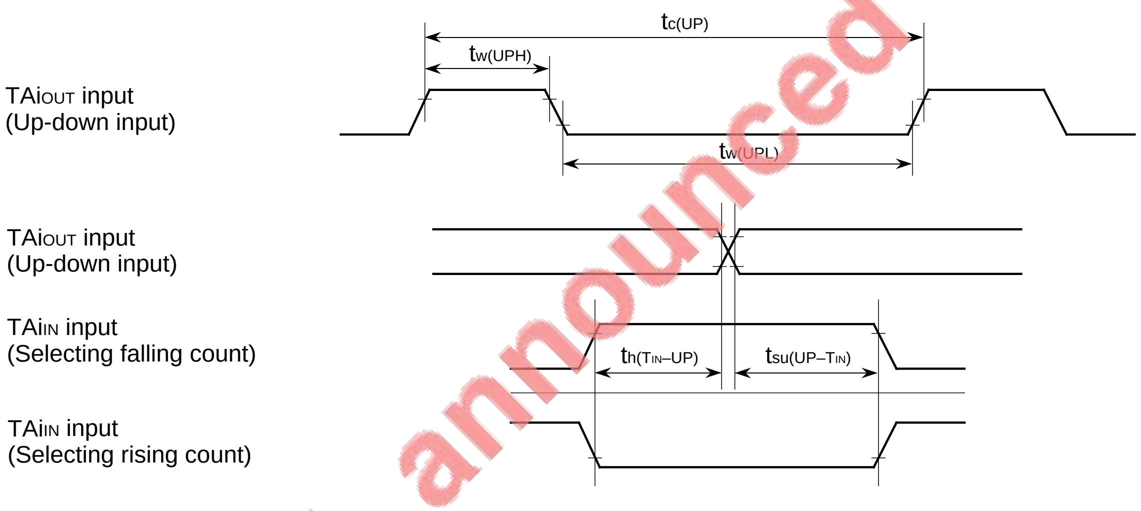
Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAj_{IN} input cycle time	2000		ns
$t_{su(TAj_{IN}-TAj_{OUT})}$	TAj_{IN} input setup time	500		ns
$t_{su(TAj_{OUT}-TAj_{IN})}$	TAj_{OUT} input setup time	500		ns

Internal peripheral devices

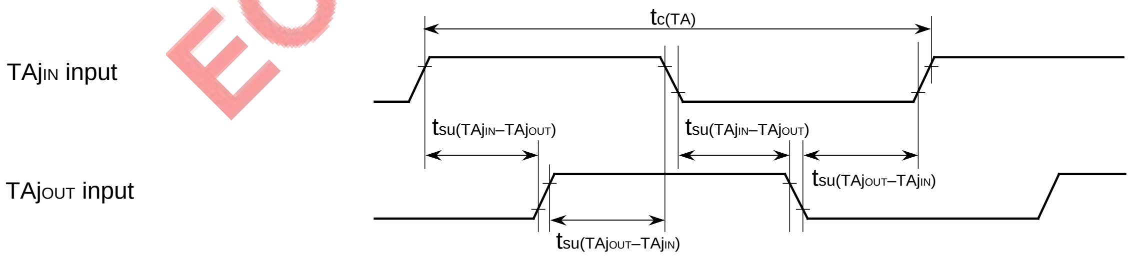
- Count input in event counter mode
- Gating input in timer mode
- External trigger input in one-shot pulse mode
- External trigger input in pulse width modulation mode



- Up-down input, count input in event counter mode



- Two-phase pulse input in event counter mode



Test conditions
- V_{CC} = 2.7–5.5 V
- Input timing voltage : V_{IL} = 0.2 V, V_{IH} = 0.8 V

Timer B input (count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time (one edge count)	250		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width (one edge count)	125		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width (one edge count)	125		ns
$t_{c(TB)}$	TBi_{IN} input cycle time (both edges count)	500		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width (both edges count)	250		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width (both edges count)	250		ns

Timer B input (pulse period measurement mode)

Symbol	Parameter	Data formula	Limits		Unit
			Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	$\frac{8 \times 10^9}{f(X_{IN})}$	1000		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})}$	500		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})}$	500		ns

Note: 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.

Timer B input (pulse width measurement mode)

Symbol	Parameter	Data formula	Limits		Unit
			Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	$\frac{8 \times 10^9}{f(X_{IN})}$	1000		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})}$	500		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width	$\frac{4 \times 10^9}{f(X_{IN})}$	500		ns

Note: 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.

A-D trigger input

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{AD_{TRG}}$ input cycle time (minimum allowable trigger)	2000		ns
$t_{w(ADL)}$	$\overline{AD_{TRG}}$ input low-level pulse width	250		ns

LOW VOLTAGE VERSION

18.4 Electrical characteristics

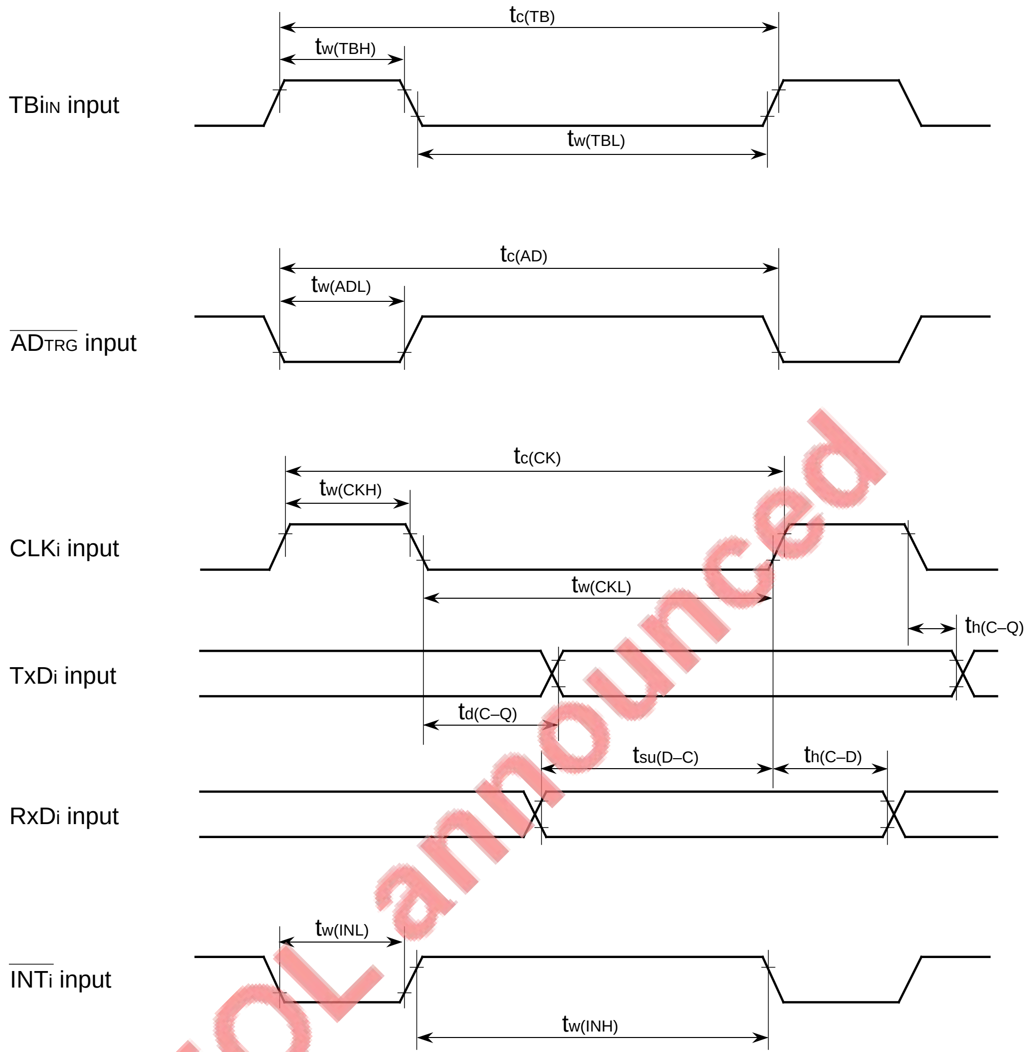
Serial I/O

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(CK)}$	CLK_i input cycle time	500		ns
$t_{w(CKH)}$	CLK_i input high-level pulse width	250		ns
$t_{w(CKL)}$	CLK_i input low-level pulse width	250		ns
$t_{d(C-Q)}$	TxD_i output delay time		170	ns
$t_{h(C-Q)}$	TxD_i hold time	0		ns
$t_{su(D-C)}$	RxD_i input setup time	80		ns
$t_{h(C-D)}$	RxD_i input hold time	100		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} = 2.7–5.5 V
- Input timing voltage : V_{IL} = 0.2 V, V_{IH} = 0.8 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V

18.4 Electrical characteristics

18.4.6 Ready and Hold

Timing requirements (V_{CC} = 2.7 – 5.5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{su(RDY-\phi_1)}$	$\overline{RDY}$ input setup time	90		ns
$t_{su(HOLD-\phi_1)}$	$\overline{HOLD}$ input setup time	90		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

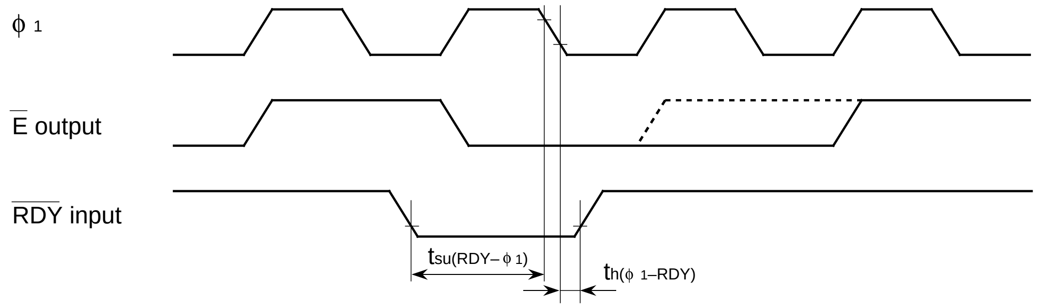
Switching characteristics (V_{CC} = 2.7 – 5.5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time		120	ns

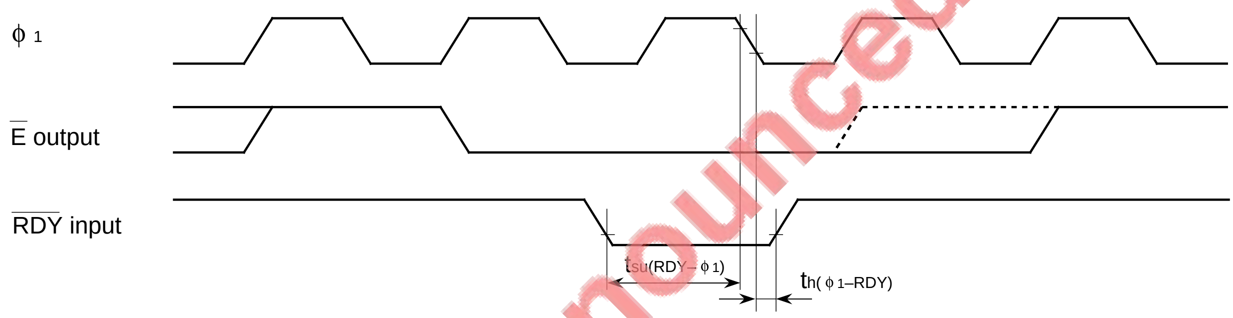
Note: For test conditions, refer to Figure 18.4.1.

●Ready

With no Wait



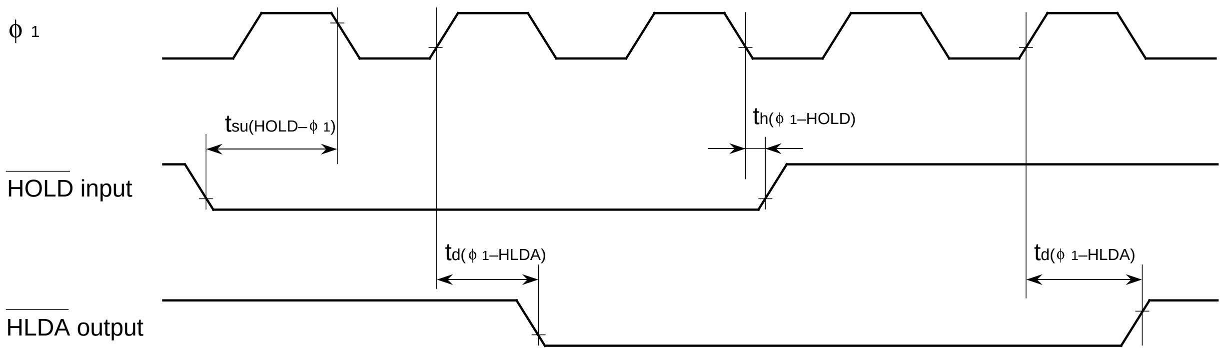
With Wait



Test conditions
- V_{CC} = 2.7–5.5 V
- Input timing voltage: V_{IL} = 0.2 V, V_{IH} = 0.8 V
- Output timing voltage: V_{OL} = 0.8 V, V_{OH} = 2.0 V

18.4 Electrical characteristics

●Hold



Test conditions
- V_{CC} = 2.7–5.5 V
- Input timing voltage : V_{IL} = 0.2 V, V_{IH} = 0.8 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V

18.4.7 Single-chip mode

Timing requirements (V_{CC} = 2.7 – 5.5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_c	External clock input cycle time	125		ns
$t_{w(H)}$	External clock input high-level pulse width	50		ns
$t_{w(L)}$	External clock input low-level pulse width	50		ns
t_r	External clock rise time		20	ns
t_f	External clock fall time		20	ns
$t_{su(P0D-E)}$	Port P0 input setup time	300		ns
$t_{su(P1D-E)}$	Port P1 input setup time	300		ns
$t_{su(P2D-E)}$	Port P2 input setup time	300		ns
$t_{su(P3D-E)}$	Port P3 input setup time	300		ns
$t_{su(P4D-E)}$	Port P4 input setup time	300		ns
$t_{su(P5D-E)}$	Port P5 input setup time	300		ns
$t_{su(P6D-E)}$	Port P6 input setup time	300		ns
$t_{su(P7D-E)}$	Port P7 input setup time	300		ns
$t_{su(P8D-E)}$	Port P8 input setup time	300		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

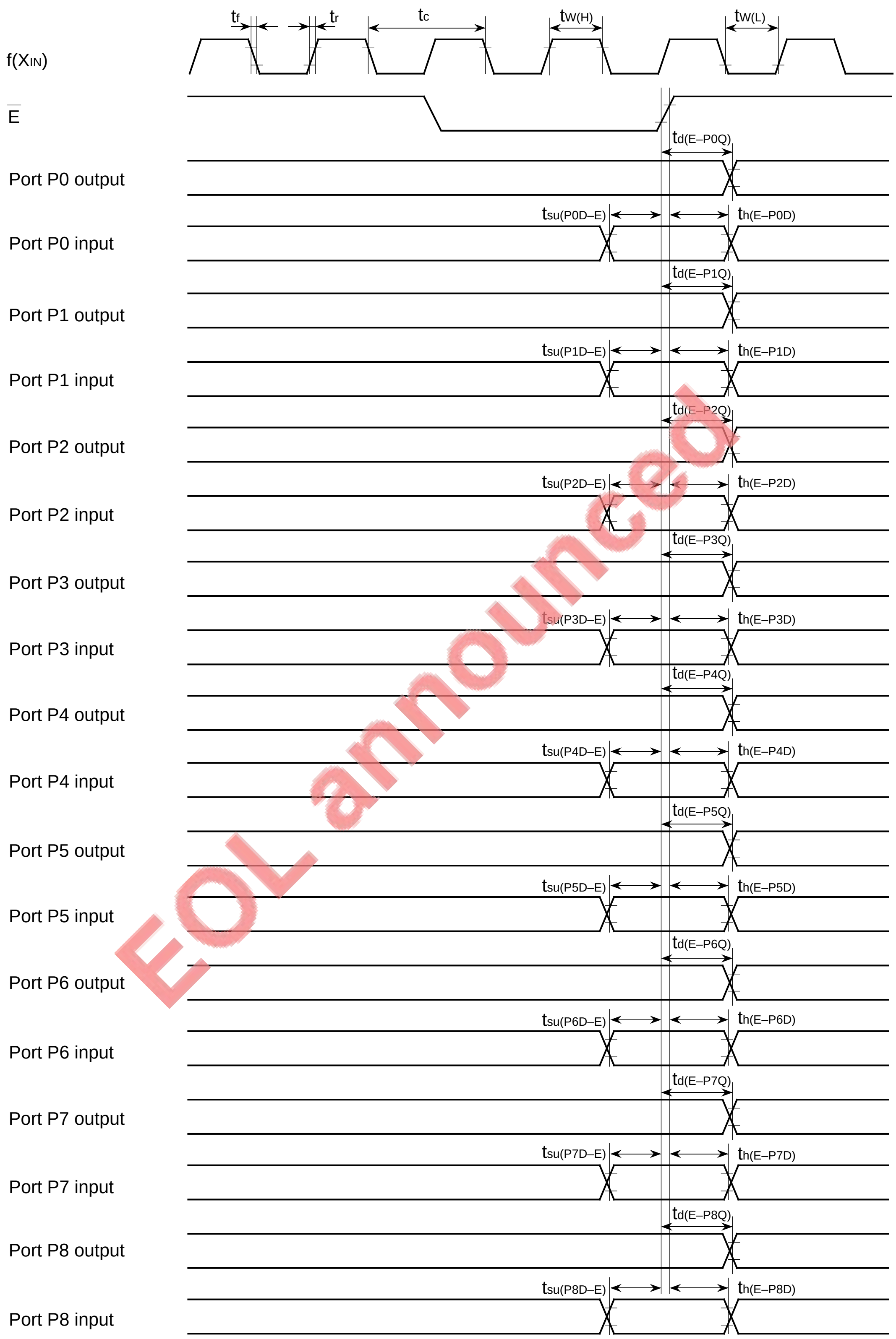
Switching characteristics (V_{CC} = 2.7 – 5.5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{d(E-P0Q)}$	Port P0 data output delay time		300	ns
$t_{d(E-P1Q)}$	Port P1 data output delay time		300	ns
$t_{d(E-P2Q)}$	Port P2 data output delay time		300	ns
$t_{d(E-P3Q)}$	Port P3 data output delay time		300	ns
$t_{d(E-P4Q)}$	Port P4 data output delay time		300	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		300	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		300	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time		300	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		300	ns

Note: For test conditions, refer to Figure 18.4.1.

18.4 Electrical characteristics

Single-chip mode



Test conditions

- V_{CC} = 2.7–5.5 V
- Input timing voltage : V_{IL} = 0.2 V, V_{IH} = 0.8 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V

18.4.8 Memory expansion mode and microprocessor mode : with no Wait

Timing requirements (V_{CC} = 2.7 – 5.5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, f(X_{IN}) = 8 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_c	External clock input cycle time	125		ns
$t_{w(H)}$	External clock input high-level pulse width	50		ns
$t_{w(L)}$	External clock input low-level pulse width	50		ns
t_r	External clock rise time		20	ns
t_f	External clock fall time		20	ns
$t_{su(P1D-E)}$	Port P1 input setup time	80		ns
$t_{su(P2D-E)}$	Port P2 input setup time	80		ns
$t_{su(P4D-E)}$	Port P4 input setup time	300		ns
$t_{su(P5D-E)}$	Port P5 input setup time	300		ns
$t_{su(P6D-E)}$	Port P6 input setup time	300		ns
$t_{su(P7D-E)}$	Port P7 input setup time	300		ns
$t_{su(P8D-E)}$	Port P8 input setup time	300		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

Switching characteristics (V_{CC} = 2.7–5.5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, f(X_{IN}) = 8 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{d(E-P4Q)}$	Port P4 data output delay time		300	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		300	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		300	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time		300	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		300	ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time	0	40	ns
$t_{w(EL)}$	$\overline{E}$ low-level pulse width	210 *		ns
$t_{d(P0A-E)}$	Port P0 address output delay time	50 *		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE = "L")		130	ns
$t_{pxz(E-P1Z)}$	Port P1 floating start delay time (BYTE = "L")		10	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	50 *		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	40 *		ns
$t_{h(E-P2Q)}$	Port P2 data output delay time		130	ns
$t_{pxz(E-P2Z)}$	Port P2 floating start delay time		10	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	50 *		ns
$t_{h(P2A-ALE)}$	Port P2 address output delay time	40 *		ns
$t_{d(ALE-E)}$	ALE output delay time	4		ns
$t_{w(ALE)}$	ALE pulse width	60 *		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	50 *		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	50 *		ns

Note: For test conditions, refer to Figure 18.4.1.
* This is the value depending on f(X_{IN}). For data formula, refer to Table 18.4.1.

LOW VOLTAGE VERSION

18.4 Electrical characteristics

Switching characteristics (V_{CC} = 2.7–5.5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, $f(X_{IN})$ = 8 MHz, unless otherwise noted)

Symbol	Parameter	Limits Min.	Limits Max.	Unit
$t_{h(E-P0A)}$	Port P0 address hold time	50 *		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")	50 *		ns
$t_{pzx(E-P1Z)}$	Port P1 floating release delay time (BYTE = "L")	95 *		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE = "H")	50 *		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time	9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time	50 *		ns
$t_{pzx(E-P2Z)}$	Port P2 floating release delay time	95 *		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	18		ns
$t_{h(E-RW)}$	R/$\overline{W}$ hold time	18		ns

Notes 1: For test conditions, refer to Figure 18.4.1.
*: This is depending on $f(X_{IN})$. For data formula, refer to Table 18.4.1.

Table 18.4.1 Bus timing data formula

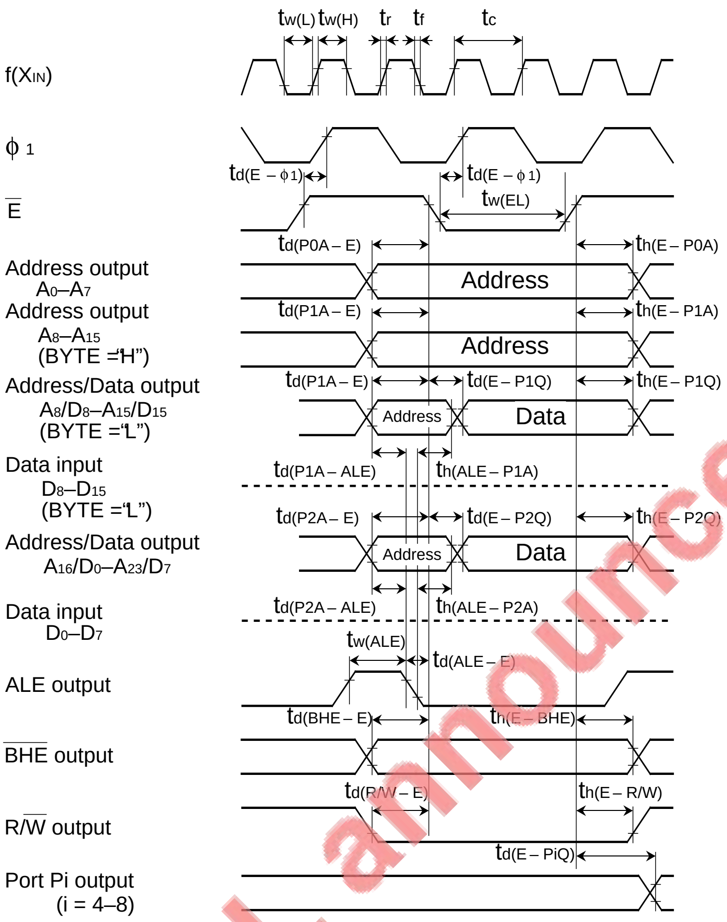
Symbol	Parameter	$f(X_{IN}) \leq$ 8 MHz
$t_{w(EL)}$	$\overline{E}$ pulse width	$\frac{2 \times 10^9}{f(X_{IN})} - 40$
$t_{d(P0A-E)}$ **(Note)**	Port P0 address output delay time	$50 + \frac{1 \times 10^9}{f(X_{IN})} - 125$
$t_{d(P1A-E)}$ **(Note)**	Port P1 address output delay time	
$t_{d(P2A-E)}$ **(Note)**	Port P2 address output delay time	
$t_{d(P1A-ALE)}$	Port P1 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 85$
$t_{d(P2A-ALE)}$	Port P2 address output delay time	
$t_{w(ALE)}$	ALE pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 65$
$t_{d(BHE-E)}$ **(Note)**	$\overline{BHE}$ output delay time	$50 + \frac{1 \times 10^9}{f(X_{IN})} - 125$
$t_{d(R/W-E)}$ **(Note)**	R/$\overline{W}$ output delay time	
$t_{h(E-P0A)}$	Port P0 address hold time	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 12.5$
$t_{h(E-P1A)}$	Port P1 address hold time	
$t_{h(E-P1Q)}$	Port P1 data hold time	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 12.5$
$t_{h(E-P2Q)}$	Port P2 data hold time	
$t_{pzx(E-P1Z)}$ **(Note)**	Port P1 floating start delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 30$
$t_{pzx(E-P2Z)}$ **(Note)**	Port P2 floating start delay time	

Unit : ns

Note: For the M37702E2LXXXGP and the M37702E4LXXXFP, refer to section **"19.5.4 Bus timing and EPROM mode."**

Memory expansion mode and microprocessor mode ; With no Wait

<Write>



Test conditions (φ 1, E, P0–P3)

•VCC = 2.7–5.5 V

•Output timing voltage : VOL = 0.8 V, VOH = 2.0 V

•Data input : VIL = 0.16 V, VIH = 0.5 V

Test conditons (P4–P8)

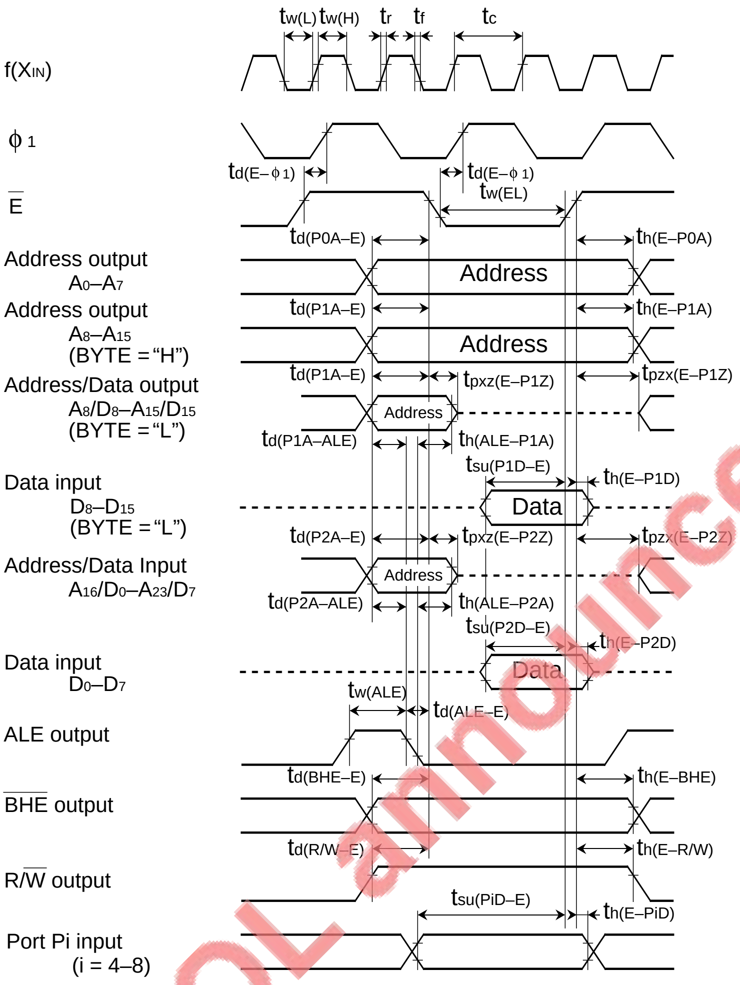
•VCC = 2.7–5.5 V

•Input timing voltage: VIL = 0.2 V, VIH = 0.8 V

•Output timing voltage: VOL = 0.8 V, VOH = 2.0 V

Memory epxansion mode and microprocessor mode ; With no Wait

<Read>



Test conditions (φ1, E, P0–P3)
- V_{CC} = 2.7–5.5 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.16 V, V_{IH} = 0.5 V

Test conditions (P4–P8)
- V_{CC} = 2.7–5.5 V
- Input timing voltage : V_{IL} = 0.2 V, V_{IH} = 0.8 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V

18.4.9 Memory expansion mode and microprocessor mode : with Wait

Timing requirements (V_{CC} = 2.7–5.5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, f(X_{IN}) = 8 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_c	External clock input cycle time	1		ns
$t_{w(H)}$	External clock input high-level pulse width			ns
$t_{w(L)}$	External clock input low-level pulse width			ns
t_r	External clock rise time		20	ns
t_f	External clock fall time		20	ns
$t_{su(P1D-E)}$	Port P1 input setup time	80		ns
$t_{su(P2D-E)}$	Port P2 input setup time	80		ns
$t_{su(P4D-E)}$	Port P4 input setup time	300		ns
$t_{su(P5D-E)}$	Port P5 input setup time	300		ns
$t_{su(P6D-E)}$	Port P6 input setup time	300		ns
$t_{su(P7D-E)}$	Port P7 input setup time	300		ns
$t_{su(P8D-E)}$	Port P8 input setup time	300		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

Switching characteristics (V_{CC} = 2.7–5.5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, f(X_{IN}) = 8 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{d(E-P4Q)}$	Port P4 data output delay time		300	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		300	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		300	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time		300	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		300	ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time	0	40	ns
$t_{w(EL)}$	$\overline{E}$ low-pulse width	460 *		ns
$t_{d(P0A-E)}$	Port P0 address output delay time	50 *		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE = “L”)		130	ns
$t_{pxz(E-P1Z)}$	Port P1 floating start delay time (BYTE = “L”)		10	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	50 *		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	40 *		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time		130	ns
$t_{pxz(E-P2Z)}$	Port P2 floating start delay time		10	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	50 *		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time	40 *		ns
$t_{d(ALE-E)}$	ALE output delay time	4		ns
$t_{w(ALE)}$	ALE pulse width	60 *		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	50 *		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	50 *		ns

Note: For test conditions, refer to Figure 18.4.1.
✻: This is depending on f(X_{IN}). For data formula, refer to Table 18.4.2.

LOW VOLTAGE VERSION

18.4 Electrical characteristics

Switching characteristics (V_{CC} = 2.7–5.5 V, V_{SS} = 0 V, Ta = –40 to 85 °C, $f(X_{IN})$ = 8 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{h(E-P0A)}$	Port P0 address hold time	50 *		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”)	50 *		ns
$t_{pzx(E-P1Z)}$	Port P1 floating release delay time (BYTE = “L”)	95 *		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE = “H”)	50 *		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time	9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time	50 *		ns
$t_{pzx(E-P2Z)}$	Port P2 floating release delay time	95 *		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time	18		ns

Note: For test conditions, refer to Figure 18.4.1.
*: This is depending on $f(X_{IN})$. For data formula, refer to Table 18.4.2.

Table 18.4.2 Bus timing data formula

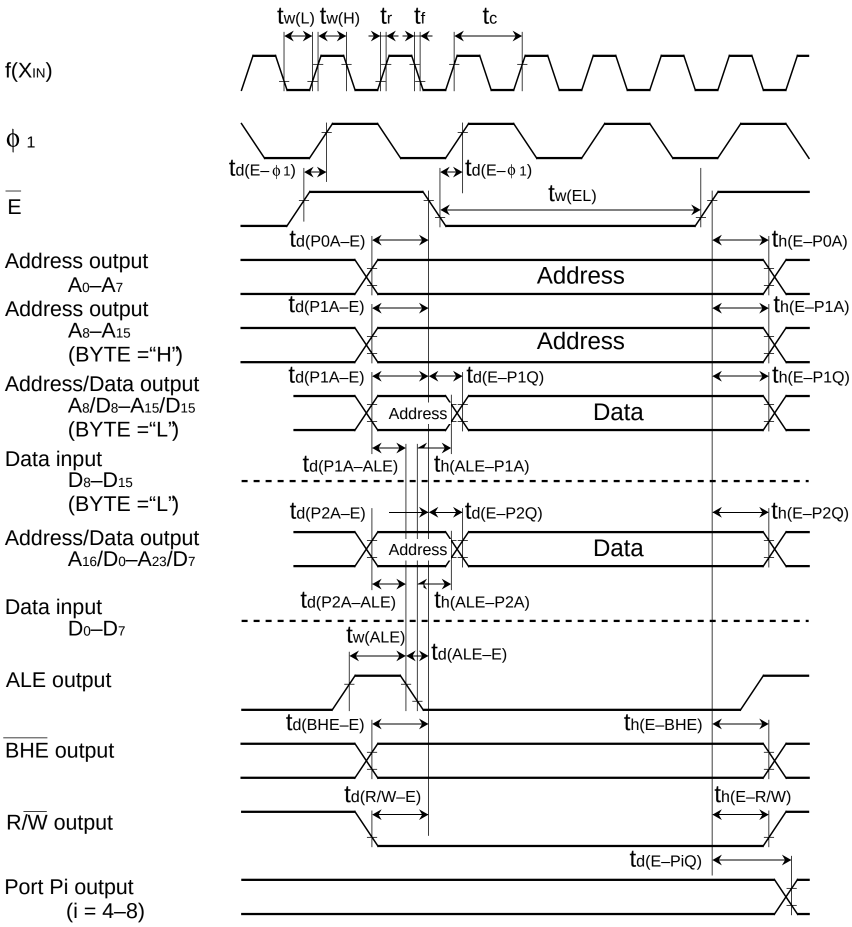
Symbol	Parameter	$f(X_{IN}) \leq$ 8 MHz
$t_{w(EL)}$	$\overline{E}$ pulse width	$\frac{4 \times 10^9}{f(X_{IN})} - 40$
$t_{d(P0A-E)}$ **(Note)**	Port P0 address output delay time	$50 + \frac{1 \times 10^9}{f(X_{IN})} - 125$
$t_{d(P1A-E)}$ **(Note)**	Port P1 address output delay time	
$t_{d(P2A-E)}$ **(Note)**	Port P2 address output delay time	
$t_{d(P1A-ALE)}$	Port P1 address output delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 85$
$t_{d(P2A-ALE)}$	Port P2 address output delay time	
$t_{w(ALE)}$	ALE pulse width	$\frac{1 \times 10^9}{f(X_{IN})} - 65$
$t_{d(BHE-E)}$ **(Note)**	$\overline{BHE}$ output delay time	$50 + \frac{1 \times 10^9}{f(X_{IN})} - 125$
$t_{d(R/W-E)}$ **(Note)**	R/$\overline{W}$ output delay time	
$t_{h(E-P0A)}$	Port P0 address hold time	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 12.5$
$t_{h(E-P1A)}$	Port P1 address hold time	
$t_{h(E-P1Q)}$	Port P1 data hold time	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 12.5$
$t_{h(E-P2Q)}$	Port P2 data hold time	
$t_{pzx(E-P1Z)}$ **(Note)**	Port P1 floating start delay time	$\frac{1 \times 10^9}{f(X_{IN})} - 30$
$t_{pzx(E-P2Z)}$ **(Note)**	Port P2 floating start delay time	

Unit : ns

Note: For the M37702E2LXXXGP and the M37702E4LXXXFP, refer to section **“19.5.4 Bus timing and EPROM mode.”**

Memory expansion mode and microprocessor mode ; With Wait

<Write>



Test conditions (φ 1, E, P0–P3)

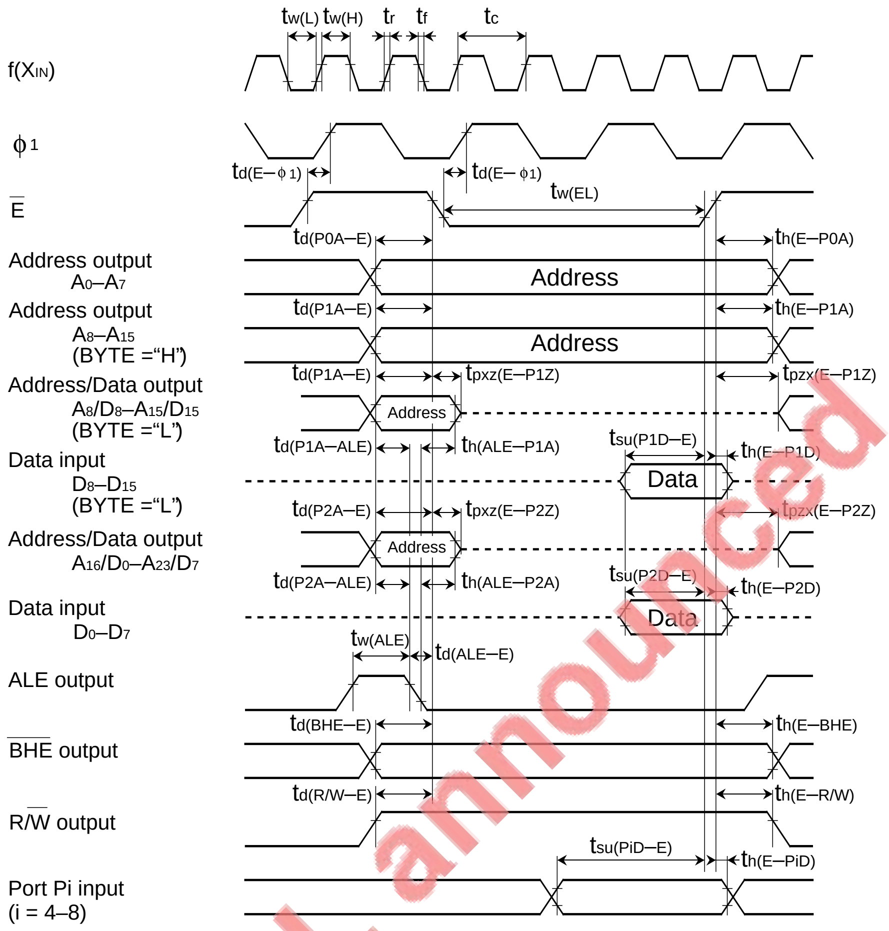
- V_{CC} = 2.7 – 5.5 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.16 V, V_{IH} =0.5 V

Test conditions (P4–P8)

- V_{CC} = 2.7–5.5 V
- Input timing voltage : V_{IL} = 0.2 V, V_{IH} = 0.8 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V

Memory expansion mode and microprocessor mode ; With Wait

<Read>



Test conditions (ϕ1, E, P0–P3)

- V_{CC} = 2.7 – 5.5 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input : V_{IL} = 0.16 V, V_{IH} = 0.5 V

Test conditons (P4–P8)

- V_{CC} = 2.7–5.5 V
- Input timing voltage : V_{IL} = 0.2 V, V_{IH} = 0.8 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V

18.4.10 Testing circuit for ports P0 to P8, ϕ_1, and $\overline{E}$

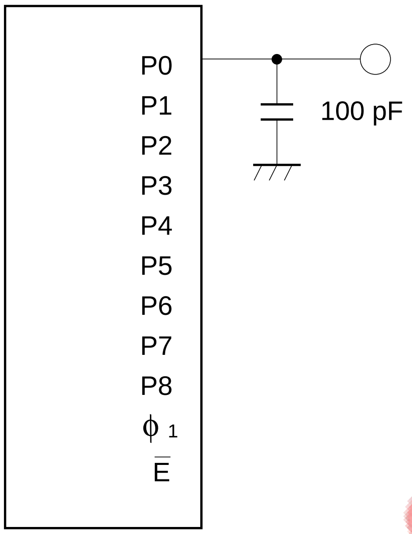


Fig. 18.4.1 Testing circuit for ports P0 to P8, ϕ_1, and $\overline{E}$

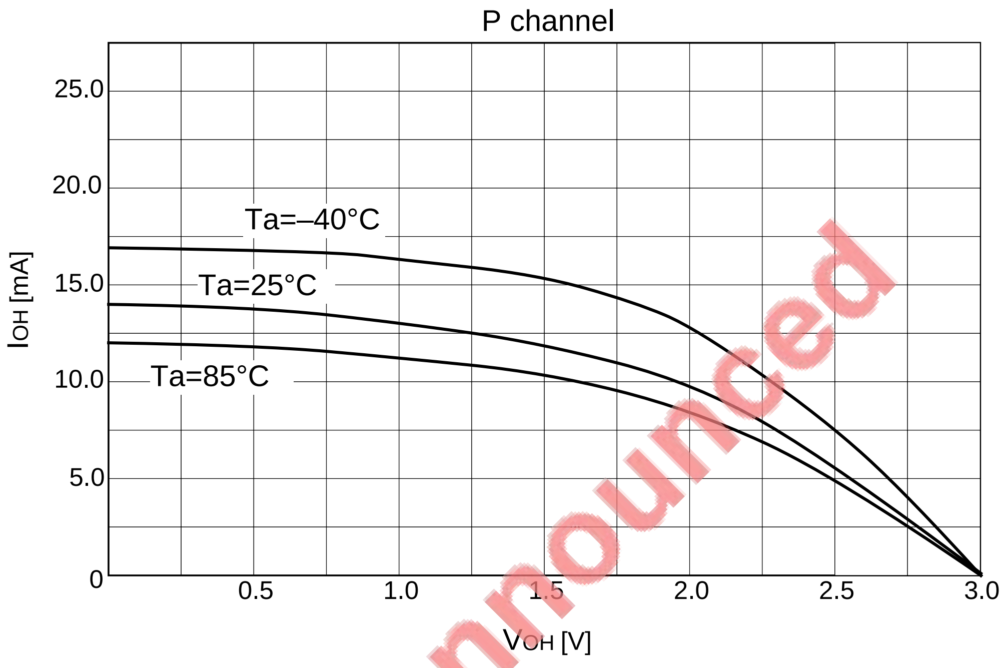
18.5 Standard characteristics

The data described below are characteristic examples for M37702M2LXXXGP. The data is not guaranteed value. Refer to section **"18.4 Electrical characteristics"** for rated value.

18.5.1 Port standard characteristics

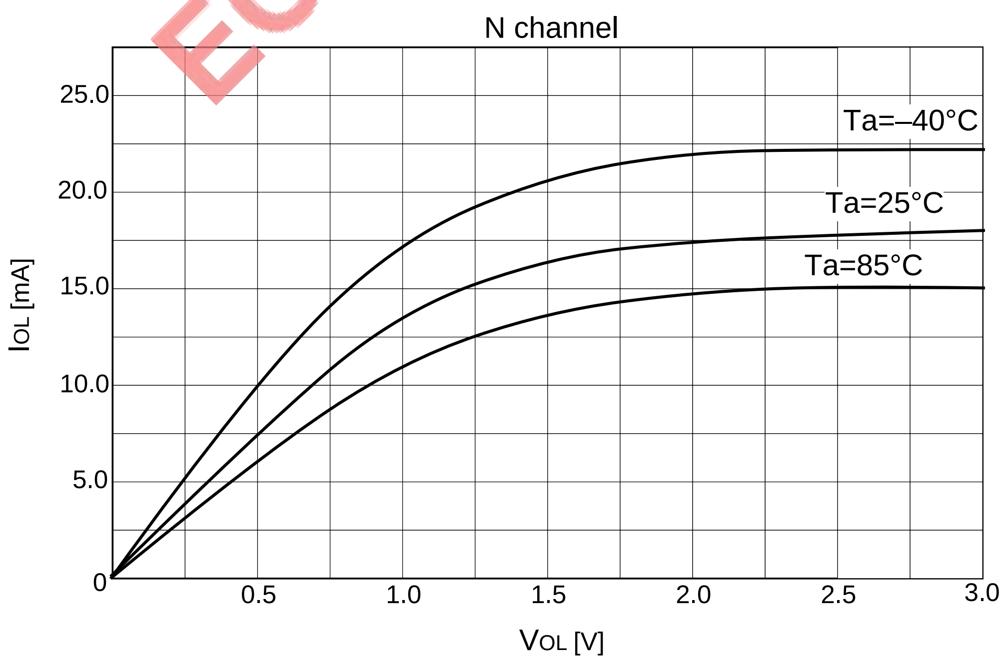
(1) Programmable I/O port (CMOS output) P channel I_{OH}–V_{OH} characteristics

Power source voltage V_{CC} = 3 V



(2) Programmable I/O port (CMOS output) N channel I_{OL}–V_{OL} characteristics

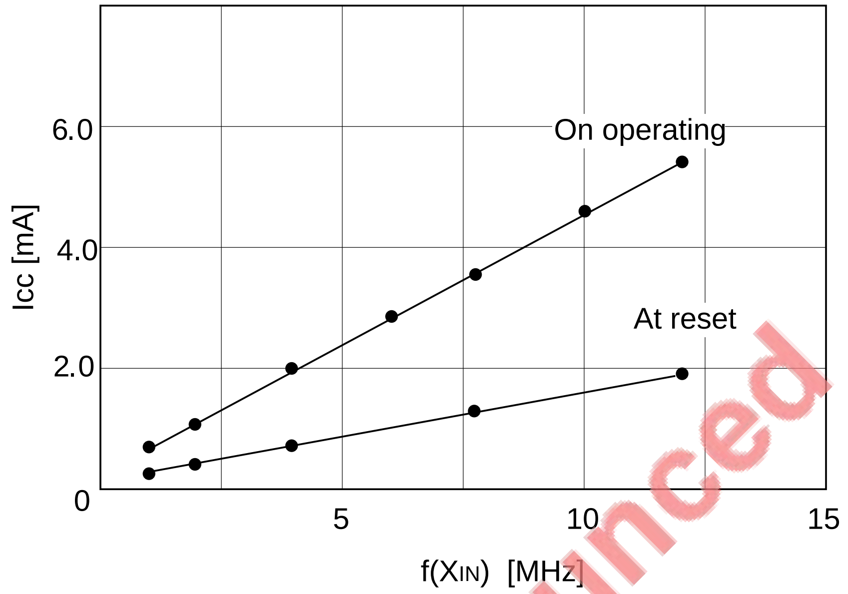
Power source voltage V_{CC} = 3 V



18.5.2 I_{CC}–f(X_{IN}) standard characteristics

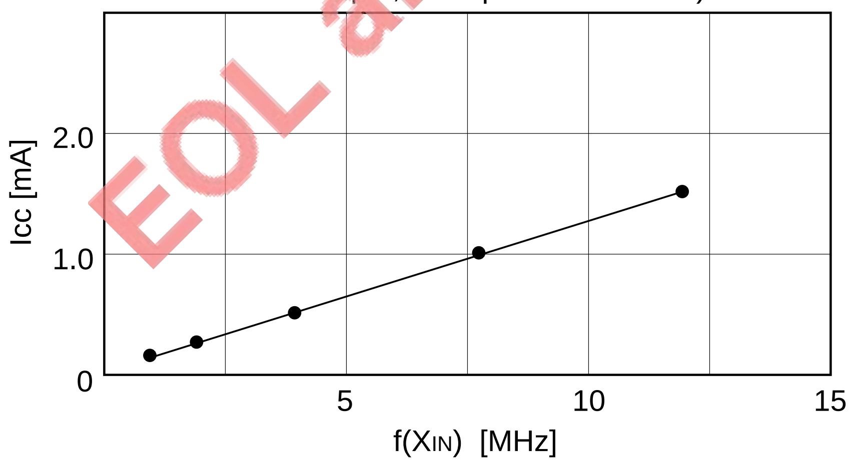
(1) I_{CC}–f(X_{IN}) characteristics on operating and at reset

Measurement condition (V_{CC} = 3 V, Ta = 25 °C, f(X_{IN}) : square waveform input, microprocessor mode)



(2) I_{CC}–f(X_{IN}) characteristics during Wait

Measurement condition (V_{CC} = 3 V, Ta = 25 °C, f(X_{IN}) : square waveform input, microprocessor mode)



18.5 Standard characteristics

18.5.3 A–D converter standard characteristics

The lower lines of the graph indicate 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 04_{16} to 05_{16} should occur at 52.7 mV, but the measured value is 2.9 mV. Therefore, the measured point of change is 52.7 + 2.9 = 55.6 mV.

The upper lines of the graph indicate the input voltage width for which the output code is constant. For example, the measured input voltage width for which the output code is $0F_{16}$ is 12.4 mV. Therefore, the differential non-linear error is 12.4 – 11.7 = 0.7 mV (0.06LSB).

Measurement condition (V_{CC} = 3 V, f(X_{IN}) = 8 MHz, Temp. = 25°C)

18.6 Application

Some application examples of connecting external memorys for the low voltage version are described bellow.
Applications shown here are just examples. Modify the desired application to suit the user's need and make sufficient evaluation before actually using it.

18.6.1 Memory expansion

The following items of the low voltage version are the same as those of section **"17.1 Memory expansion."** However, a part of the formulas and constants for parameters is different.

•Memory expansion model
•Formulas for address access time of external memory
•Bus timing
•Memory expansion method

① **Address access time of external memory $t_{a(AD)}$**

$t_{a(AD)} = t_{d(P0A/P1A/P2A–E)} + t_{w(EL)} – t_{su(P2D/P1D-E)}$ – (address decode time*1 + address latch delay time*2)

$t_{d(P0A/P1A/P2A–E)}$: $t_{d(P0A–E)}$, $t_{d(P1A–E)}$, or $t_{d(P2A–E)}$

$t_{su(P2D/P1D–E)}$: $t_{su(P2D–E)}$, or $t_{su(P1D–E)}$

address decode time*1 : time necessary for validating a chip select signal after an address is decoded
address latch delay time*2 : delay time necessary for latching an address
(This is not necessary on the minimum model.)

② **Data setup time of external memory for writing data $t_{su(D)}$**

$t_{su(D)} = t_{w(EL)} – t_{d(E–P2Q/P1Q)}$

$t_{d(E–P2Q/P1Q)}$: $t_{d(E–P2Q)}$, or $t_{d(E–P1Q)}$

Table 18.6.1 lists the calculation formulas and constants for each parameter of the low voltage version. Figure 18.6.1 shows the relationship between $t_{a(AD)}$ and $f(X_{IN})$. Figure 18.6.2 shows the relationship between $t_{su(D)}$ and $f(X_{IN})$.

Table 18.6.1 Calculation formulas and constants for each parameter (Unit : ns)

Parameter \ $f(X_{IN})$	$f(X_{IN}) \leq 8$ MHz	
	No wait	Wait
$t_{d(P0A-E)}$ $t_{d(P1A-E)}$ $t_{d(P2A-E)}$ **(Note)**	$50 + \frac{1 \times 10^9}{f(X_{IN})} - 125$	
$t_{w(EL)}$	$\frac{2 \times 10^9}{f(X_{IN})} - 40$	$\frac{4 \times 10^9}{f(X_{IN})} - 40$
$t_{su(P1D-E)}$ $t_{su(P2D-E)}$	80	
$t_{d(E-P1Q)}$ $t_{d(E-P2Q)}$	130	
$t_{pxz(E-P1Z)}$ $t_{pxz(E-P2Z)}$	10	
$t_{pzx(E-P1Z)}$ $t_{pzx(E-P2Z)}$ **(Note)**	$\frac{1 \times 10^9}{f(X_{IN})} - 30$	

Note: For M37702E2LXXXGP and M37702E4LXXXFP, refer to section **"19.5.4 Bus timing and EPROM mode."**

18.6 Application

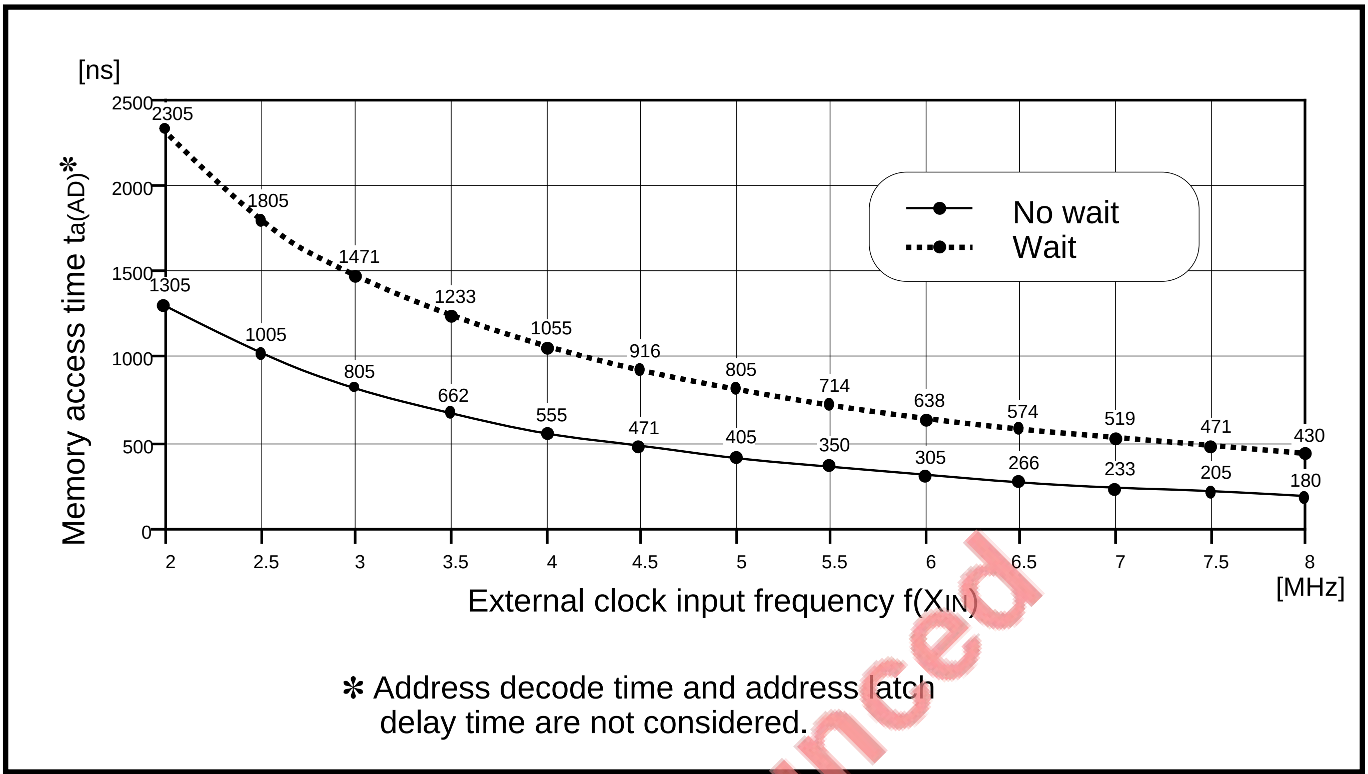


Fig. 18.6.1 Relationship between $t_{a(AD)}$ and $f(X_{IN})$

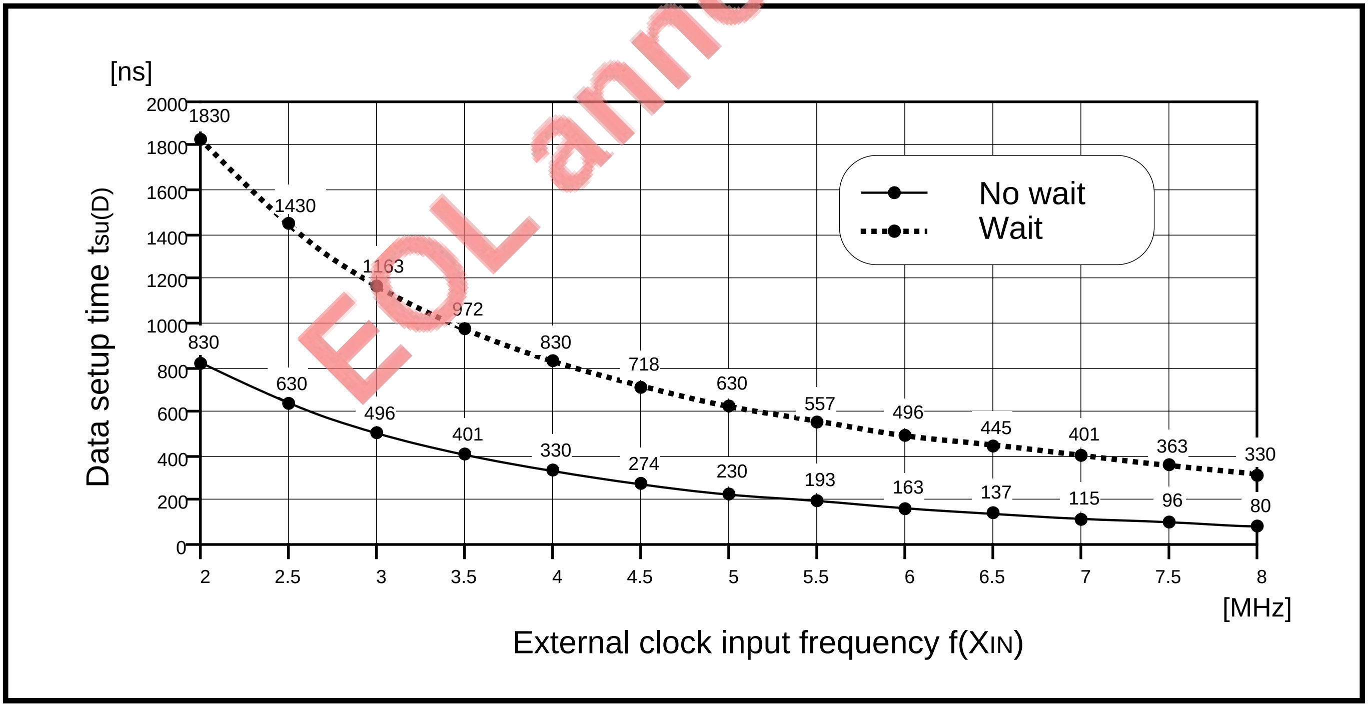
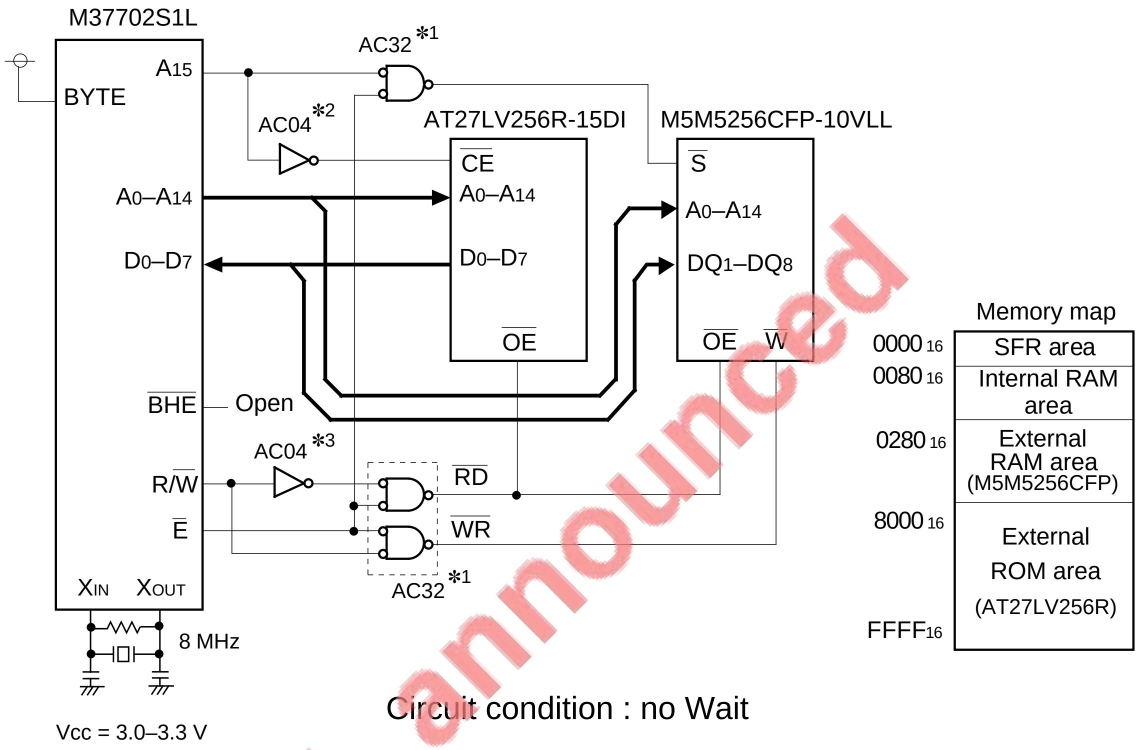


Fig. 18.6.2 Relationship between $t_{su(D)}$ and $f(X_{IN})$

18.6.2 Memory expansion example on minimum model

Figure 18.6.3 shows a memory expansion example on the minimum model (with external RAM) and Figure 18.6.4 shows the corresponding timing diagram. In this example, an Atmel company's EPROM (AT27LV256R) is used as the external ROM.

In Figure 18.6.3, the circuit condition is "No Wait."



*1, *2: Use the elements of which propagation delay time is within 30 ns.

*3: Use the elements of which propagation delay time is within 50 ns.

Fig. 18.6.3 Memory expansion example on minimum model

18.6 Application

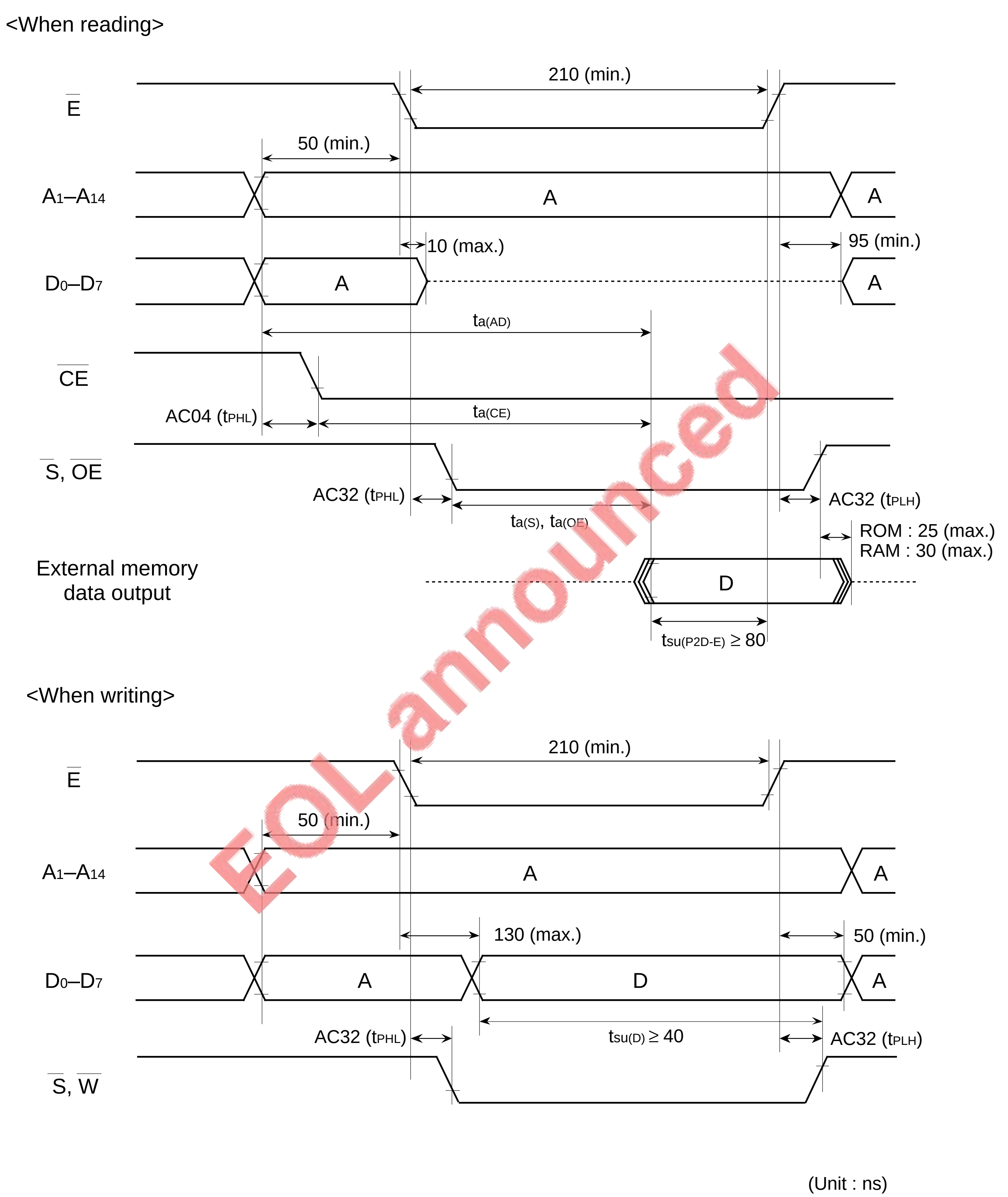


Fig. 18.6.4 Timing diagram on minimum model

18.6.3 Memory expansion example on medium model A

Figure 18.6.5 shows a memory expansion example on the medium model A of mask ROM version and PROM version. Figure 18.6.6 shows the corresponding timing diagram.

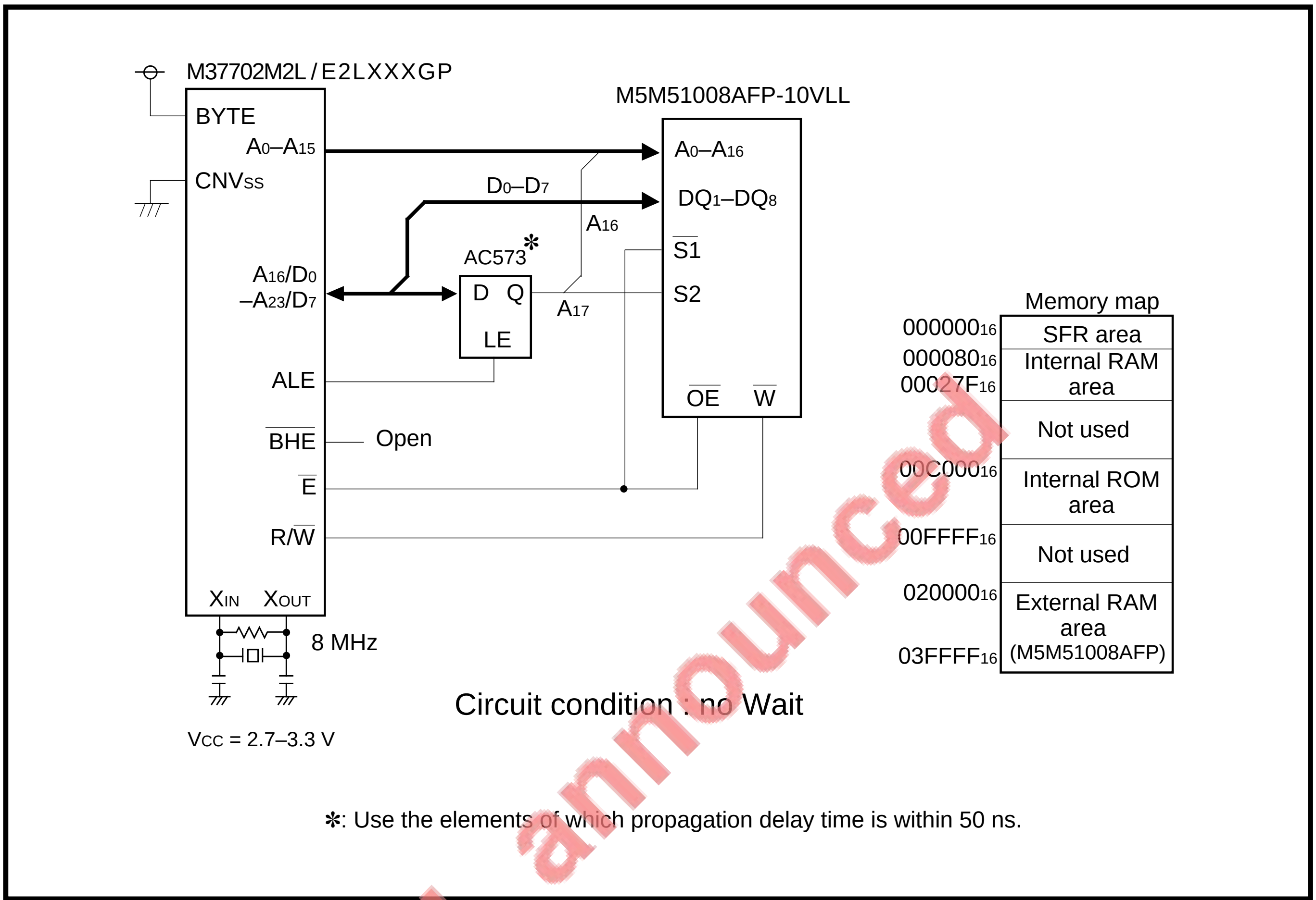


Fig. 18.6.5 Memory expansion example on medium model A

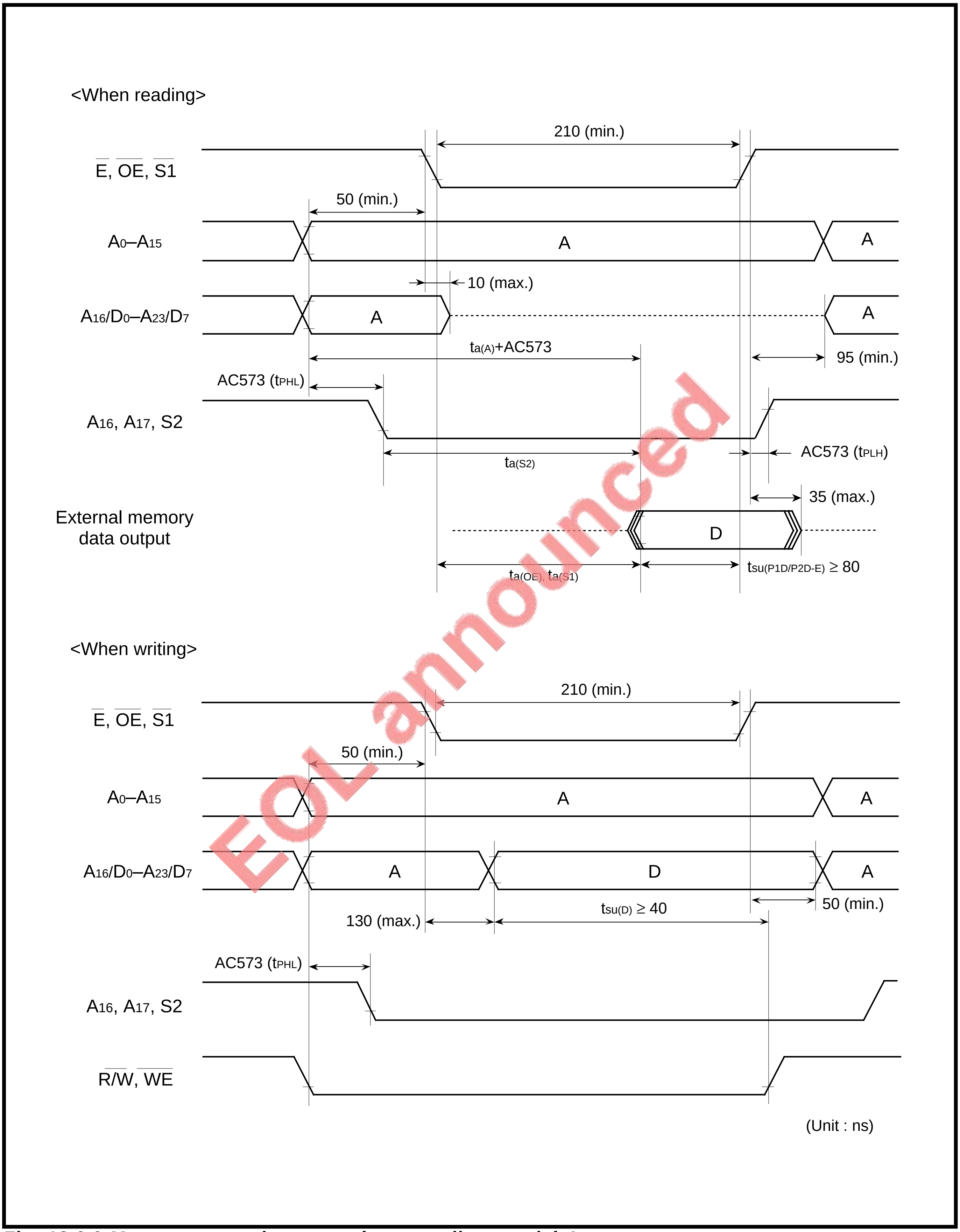


Fig. 18.6.6 Memory expansion example on medium model A

18.6.4 Memory expansion example on maximum model

Figure 18.6.7 shows a memory expansion example on the maximum model. Figure 18.6.8 shows the corresponding timing diagram. In this example, Atmel company's EPROMs (AT27LV256R) are used as the external ROMs.
In Figure 18.6.7, the circuit condition is "No Wait."

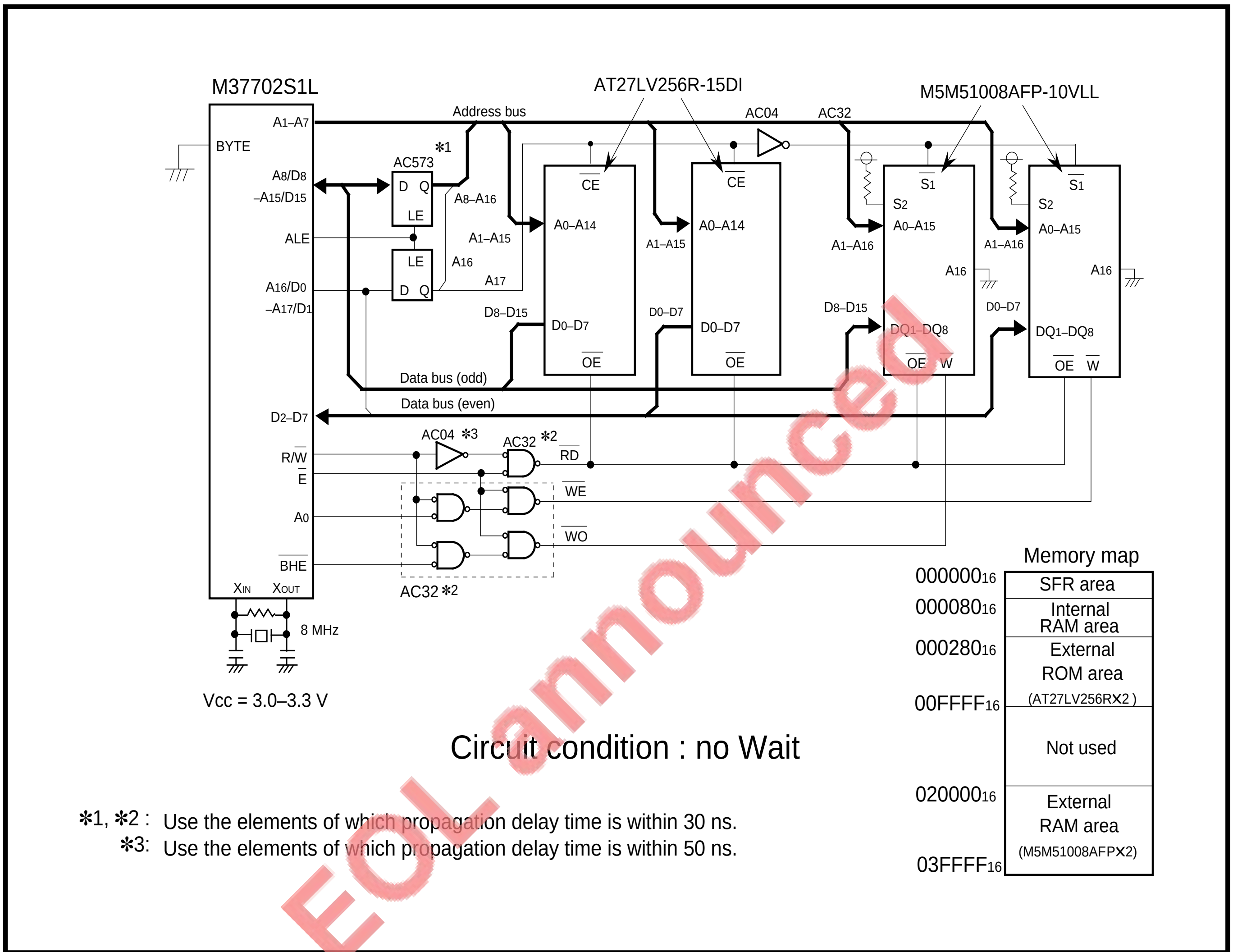


Fig. 18.6.7 Memory expansion example on maximum model

18.6 Application

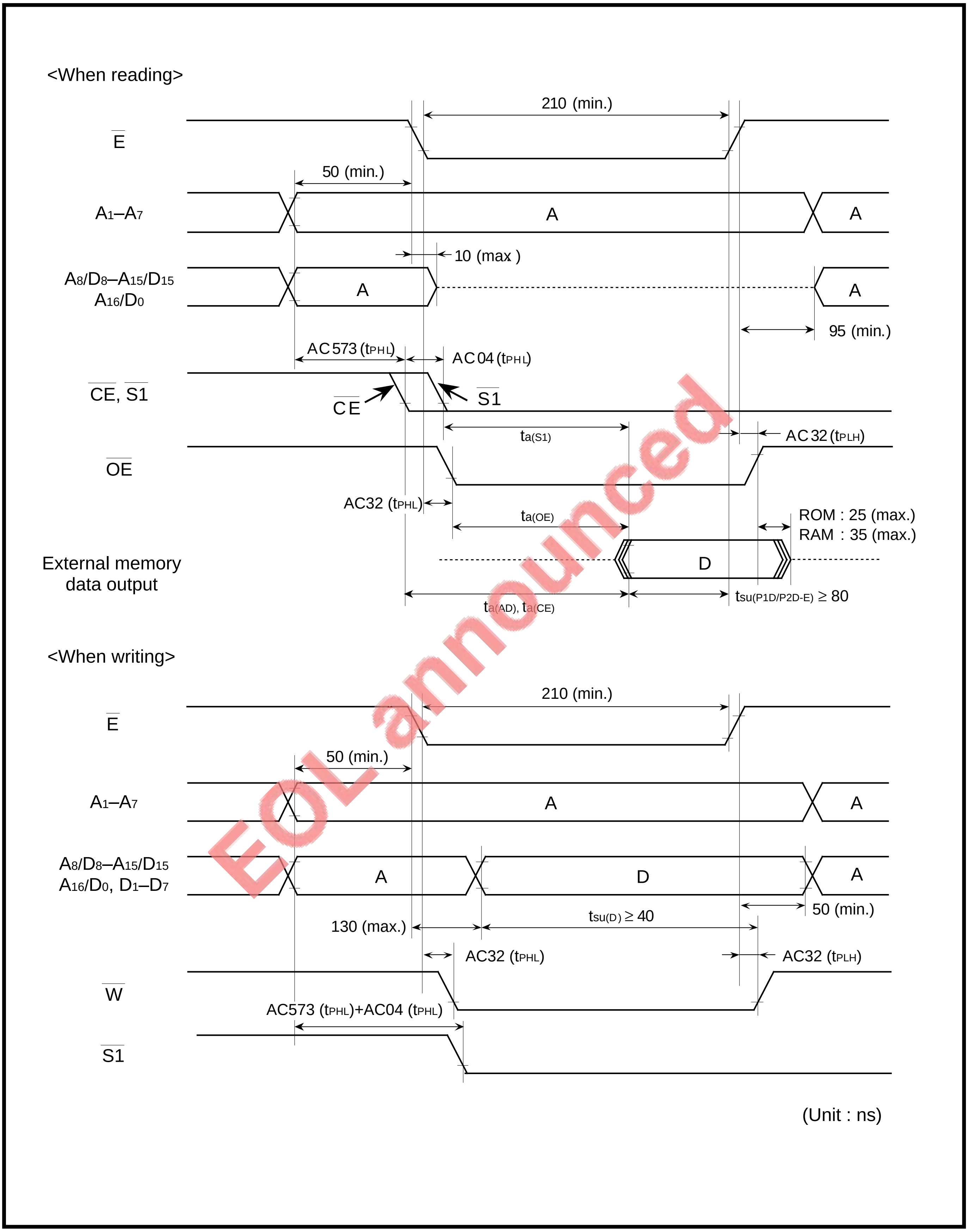


Fig. 18.6.8 Timing diagram on maximum model

18.6.5 Ready generating circuit example

When validating "Wait" only for a certain area (for example, ROM area) in Figures 18.6.3 to 18.6.8, use Ready function.

Figure 18.6.9 shows a Ready generating circuit example.

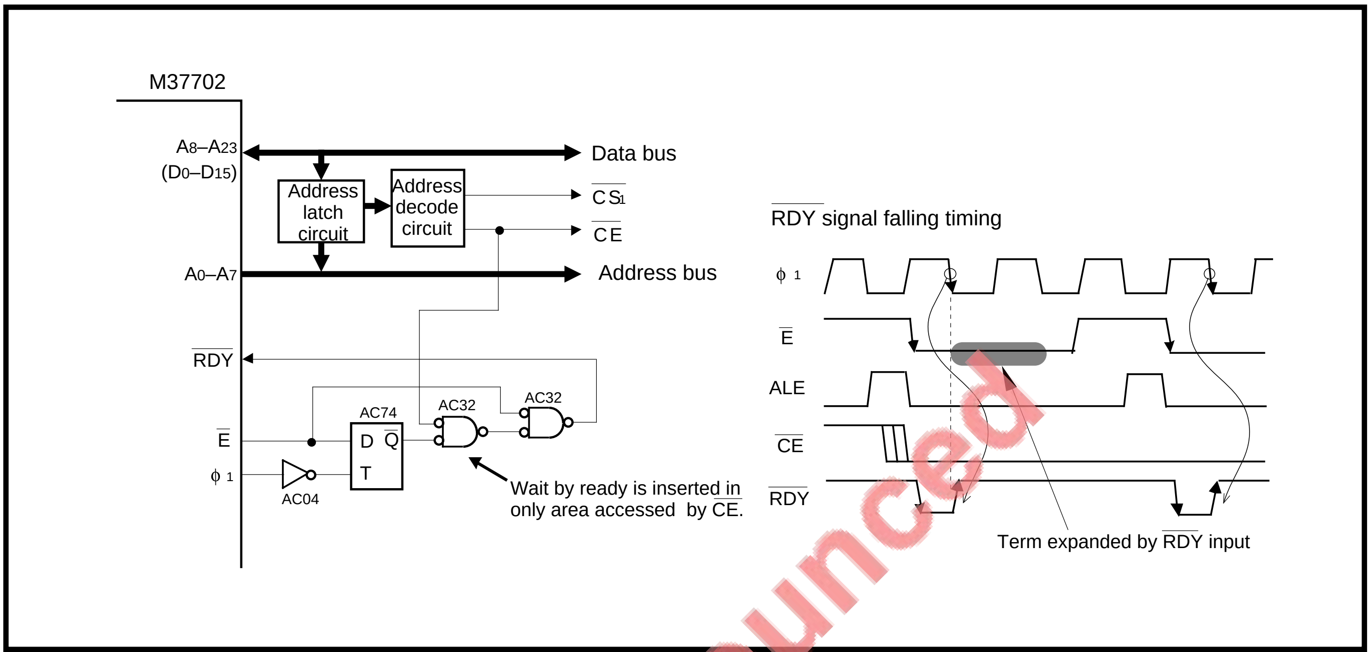


Fig. 18.6.9 Ready generating circuit example

MEMORANDUM

CHAPTER 19

PROM VERSION

PROM VERSION

19.1 Overview

This chapter describes the PROM version including the PROM.
The PROM version can be used with the program written into the built-in PROM.

7703 Group
Refer to “**Chapter 20. 7703 GROUP**” about the pin connections and others.

19.1 Overview

In the PROM version, programming to the built-in PROM can be performed by using a general-purpose PROM programmer and a programming adapter, which is suitable for the used microcomputer.
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 built-in PROM version has the same functions as the mask ROM version except that the former has a built-in PROM.

Table 19.1.1 Write address of PROM version

	Type name	PROM size (Byte)	RAM size (Byte)	Write address 1M mode	Write address 256K mode
One time PROM version	M37702E2BXXXFP	16K	512	1C000$_{16}$ to 1FFFF$_{16}$	4000$_{16}$ to 7FFF$_{16}$
	M37702E2BXXXHP				
	M37702E2AXXXFP (**Note 1**)				
	M37702E2LXXXGP (**Note 1**)				
	M37702E2LXXXHP			1C000$_{16}$ to 1FFFF$_{16}$	
	M37702E4BXXXFP	32K	2048	18000$_{16}$ to 1FFFF$_{16}$	0000$_{16}$ to 7FFF$_{16}$
	M37702E4AXXXFP (**Note 1**)				
	M37702E4LXXXFP (**Note 1**)				
	M37702E4LXXXGP			18000$_{16}$ to 1FFFF$_{16}$	
	M37702E6BXXXFP	48K	2048	14000$_{16}$ to 1FFFF$_{16}$	
	M37702E6LXXXFP				
	M37702E8BXXXFP	60K	2048	11000$_{16}$ to 1FFFF$_{16}$	
	M37702E8BXXXHP				
	M37702E8LXXXFP				
	M37702E8LXXXHP				
EPROM version	M37702E2BFS	16K	512	1C000$_{16}$ to 1FFFF$_{16}$	4000$_{16}$ to 7FFF$_{16}$
	M37702E2AFS (**Note 1**)				
	M37702E4BFS	32K	2048	18000$_{16}$ to 1FFFF$_{16}$	0000$_{16}$ to 7FFF$_{16}$
	M37702E4AFS (**Note 1**)				
	M37702E6BFS	48K	2048	14000$_{16}$ to 1FFFF$_{16}$	
	M37702E8BFS	60K	2048	11000$_{16}$ to 1FFFF$_{16}$	

Notes 1: Refer also to section "**19.5.4 Bus timing and EPROM mode**."

2: A blank product of the one time PROM version does not have the ROM number, which is printed on the **XXX** position. For example, M37702E2BFP.

19.2 EPROM mode

The built-in PROM version has the following two modes :

● Normal operating mode
  This mode has the same function as the mask ROM version.

● EPROM mode
  The built-in PROM can be programmed and read in this mode. The PROM version enters this mode when "L" level is input to the $\overline{RESET}$ pin

19.2.1 Write method

There are 2 types of the EPROM mode: 1M mode and 256K mode.
256K mode is recommended to write data deeply for the one time PROM version of which internal PROM size is 32 Kbytes or less. 1M mode is recommended for the EPROM version owing to its write velocity faster than 256K mode. It is because to write and erase is repeated for the EPROM version. However, the M37702E2AFS and M37702E4AFS cannot use 1M mode.
Additionally, use 1M mode to write and read in the built-in PROM version of which PROM size is 32 Kbytes or more.

19.2.2 Pin description

Table 19.2.1 lists the pin description in the EPROM mode.
In the normal operating mode, each pin has the same function as the mask ROM version.

Table 19.2.1 Pin description in EPROM mode

Pin	Name	Input/Output	Functions
Vcc, Vss	Power source input	—	Apply 5 V ± 10% to pin Vcc, and 0 V to pin Vss.
CNVss	V_{PP} input	Input	Apply V_{PP} level when programming or verifying.
BYTE			
$\overline{RESET}$	Reset input	Input	Connect to pin Vss.
X_{IN}	Clock input	Input	Connect a ceramic resonator or a quartz-crystal oscillator between pins X_{IN} and X_{OUT}. When an external generated clock is input, the clock must be input to pin X_{IN}, and pin X_{OUT} must be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Open.
AVcc, AVss	Analog power source input	—	Connect pin AVcc to Vcc and pin AVss to Vss.
V_{REF}	Reference voltage input	Input	Connect to pin Vss.
$P0_0$–$P0_7$	Address input (A_0–A_7)	Input	Input pins for A_0–A_7 of address.
$P1_0$–$P1_7$	Address input (A_8–A_{15})	Input	Input pins for A_8–A_{15} of address. Connect $P1_7$ to Vcc in 256K mode.
$P2_0$–$P2_7$	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 Vss.
$P4_0$–$P4_7$	Input port P4	Input	Connect to Vss.
$P5_0$	Control input	Input	$P5_0$ functions as $\overline{PGM}$ input pin in 1M mode. Connect to Vcc in 256K mode.
$P5_1$, $P5_2$			$P5_1$ functions as $\overline{OE}$ input pin and $P5_2$ does as $\overline{CE}$ input pin.
$P5_3$–$P5_5$	Input port P5	Input	Connect to Vcc.
$P5_6$			Connect to Vcc in 1M mode or to Vss in 256K mode.
$P5_7$			Connect to Vss.
$P6_0$–$P6_7$	Input port P6	Input	Connect to Vss.
$P7_0$–$P7_7$	Input port P7	Input	Connect to Vss.
$P8_0$–$P8_7$	Input port P8	Input	Connect to Vss.

19.3 1M mode

1M mode can perform reading/programming from and to the built-in PROM with the same manner as M5M27C101K. However, there is no device identification code. Accordingly, programming conditions must be set carefully.

Table 19.3.1 lists the pin correspondence with M5M27C101K. Figures 19.3.1 and 19.3.2 show the pin connections in 1M mode.

Table 19.3.1 Pin correspondence with M5M27C101K

	M37702E2BXXXFP (M37702E2BFP) M37702E2BFS	M5M27C101K
Vcc	Vcc	Vcc
V_{PP} input	CNVss, BYTE	V_{PP}
Vss	Vss	Vss
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	$P5_0$	$\overline{PGM}$

●1M mode (top view)

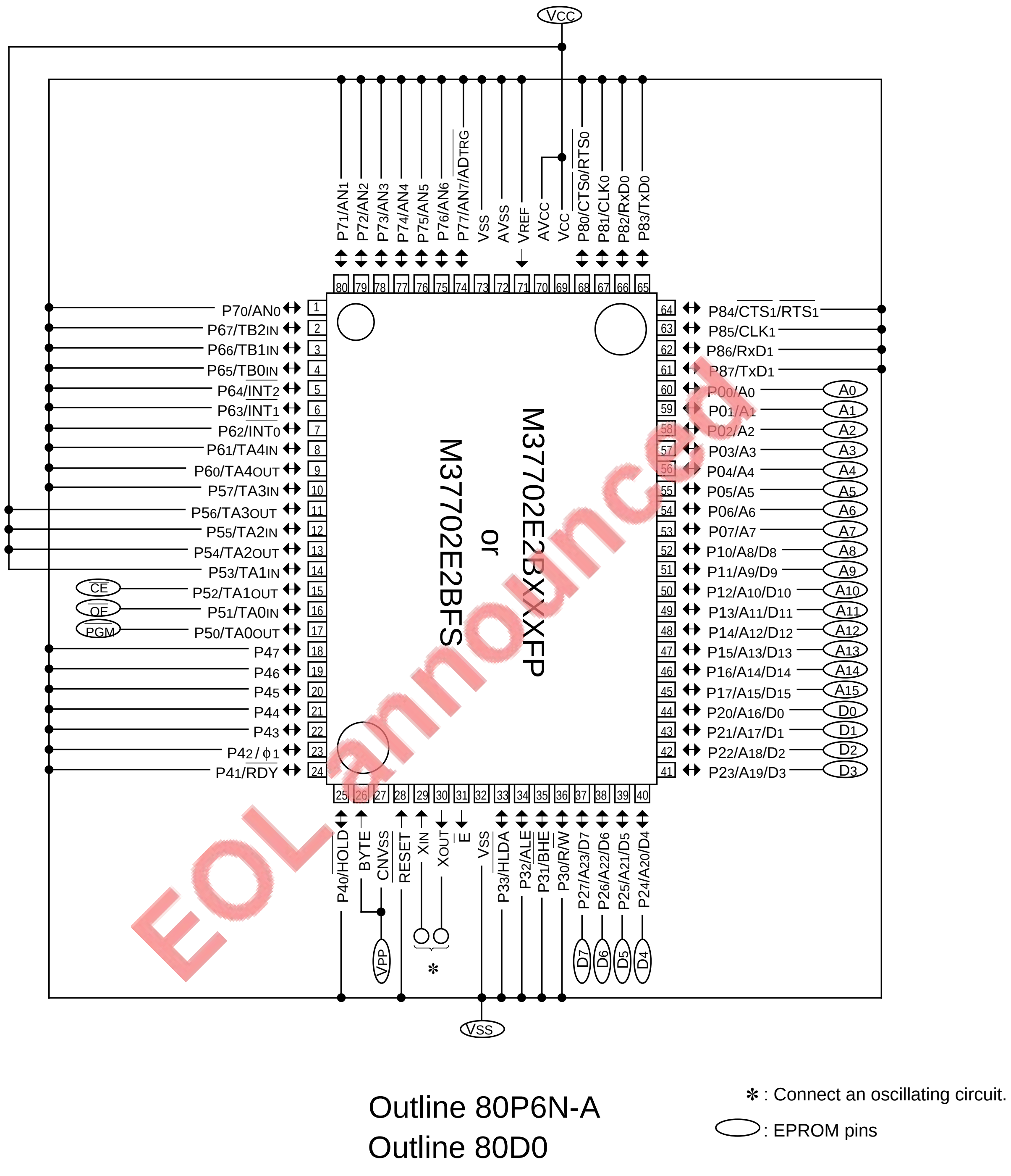


Fig. 19.3.1 Pin connections in 1M mode (1)

●1M mode (top view)

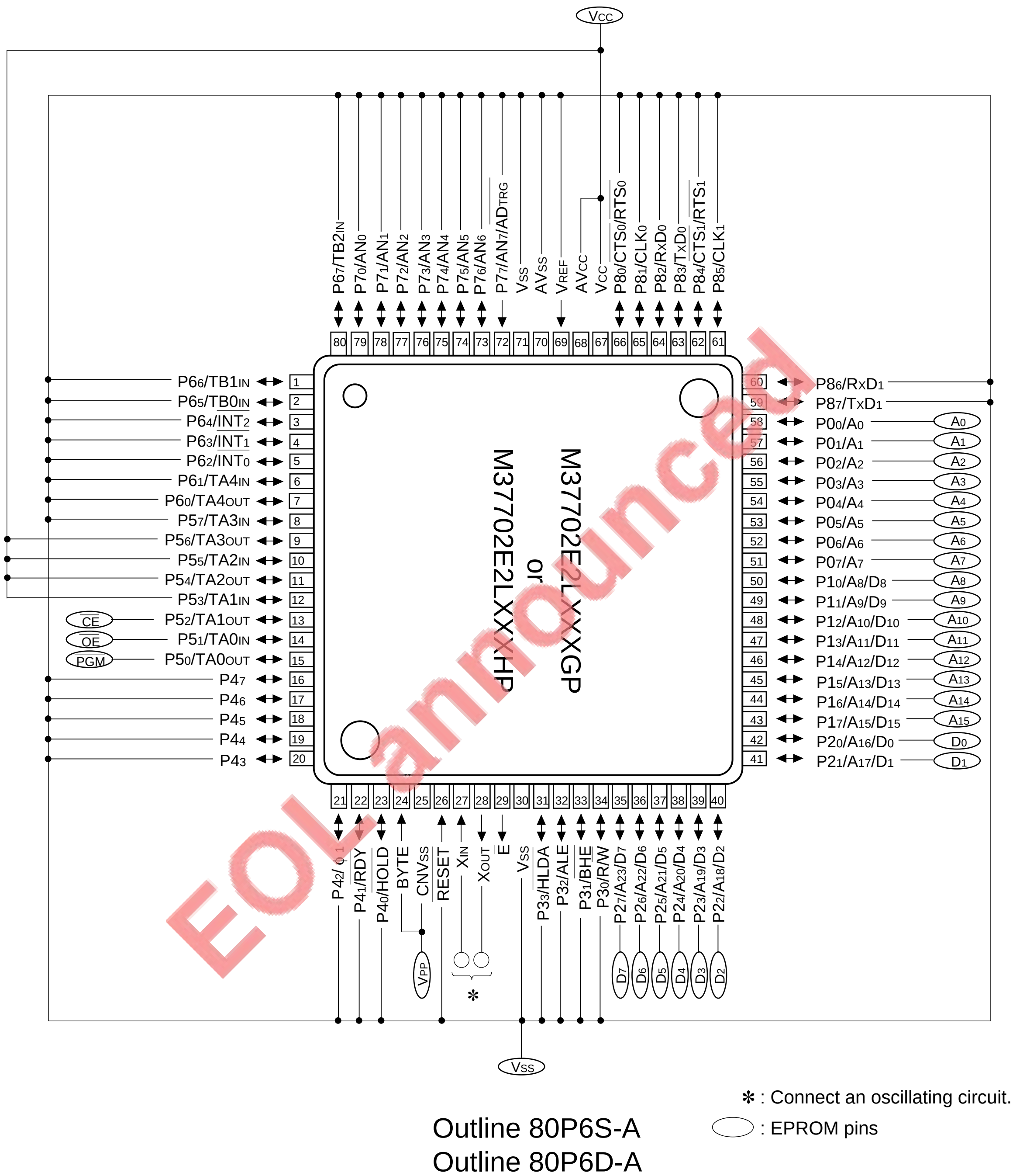


Fig. 19.3.2 Pin connections in 1M mode (2)

19.3.1 Read/Program/Erase

Table 19.3.2 lists the built-in PROM state in 1M mode and each mode is described bellow.

(1) Read

When pins $\overline{CE}$ and $\overline{OE}$ are set to "L" level and an address is input to address input pins, the contents of the built-in PROM can be output from data I/O pins and read.
When pins $\overline{CE}$ and $\overline{OE}$ are set to "H" level, data I/O pins enter the floating state.

(2) Program (Write)

When pin $\overline{CE}$ is set to "L" level and pin $\overline{OE}$ is set to "H" level and V_{PP} level is applied to pin V_{PP}, 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 pin $\overline{PGM}$ 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.

Table 19.3.2 Built-in PROM state in 1M mode

Mode \ Pin name	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	V_{PP}	Vcc	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}.

19.3.2 Programming algorithm of 1M mode

Figure 19.3.3 shows the programming algorithm flow chart of 1M mode.

① Set Vcc = 6 V, VPP = 12.5 V, and address to $1C000_{16}$*.
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 pulses (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).

✻ : This applies to the M37702E2BFS. Refer to **Table 19.1.1** about each write address of other products.

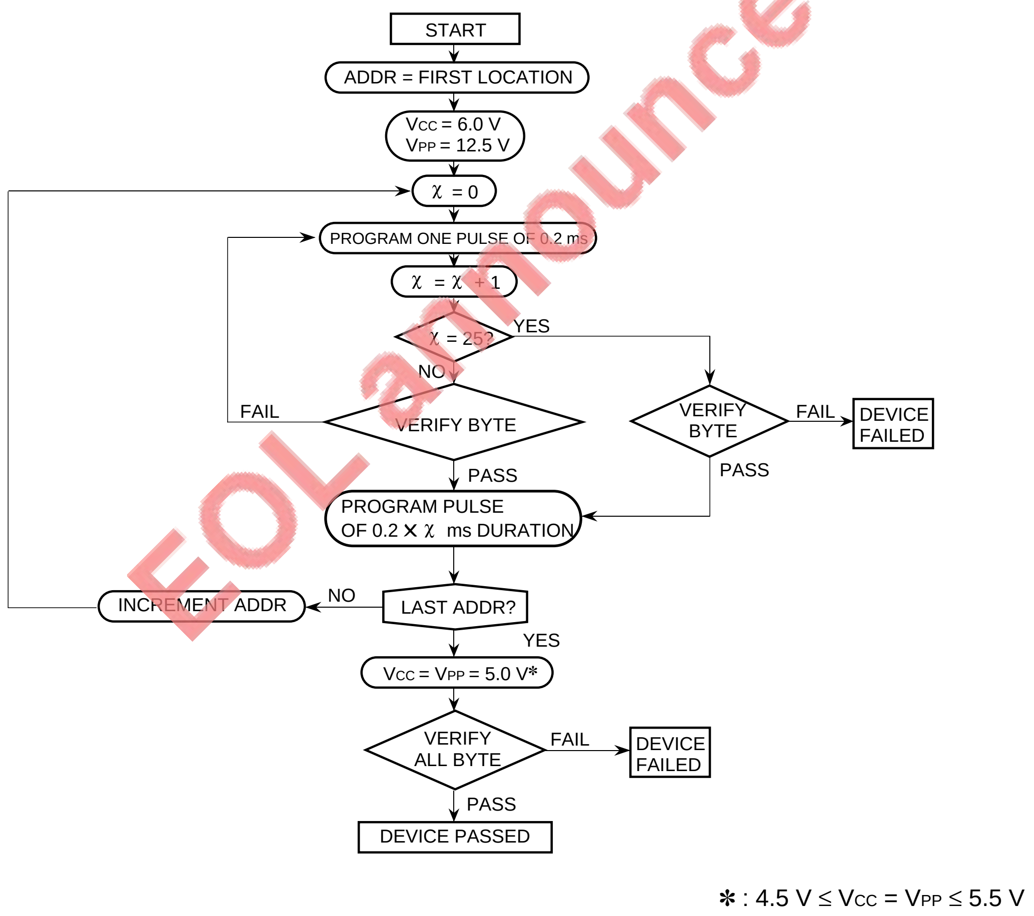


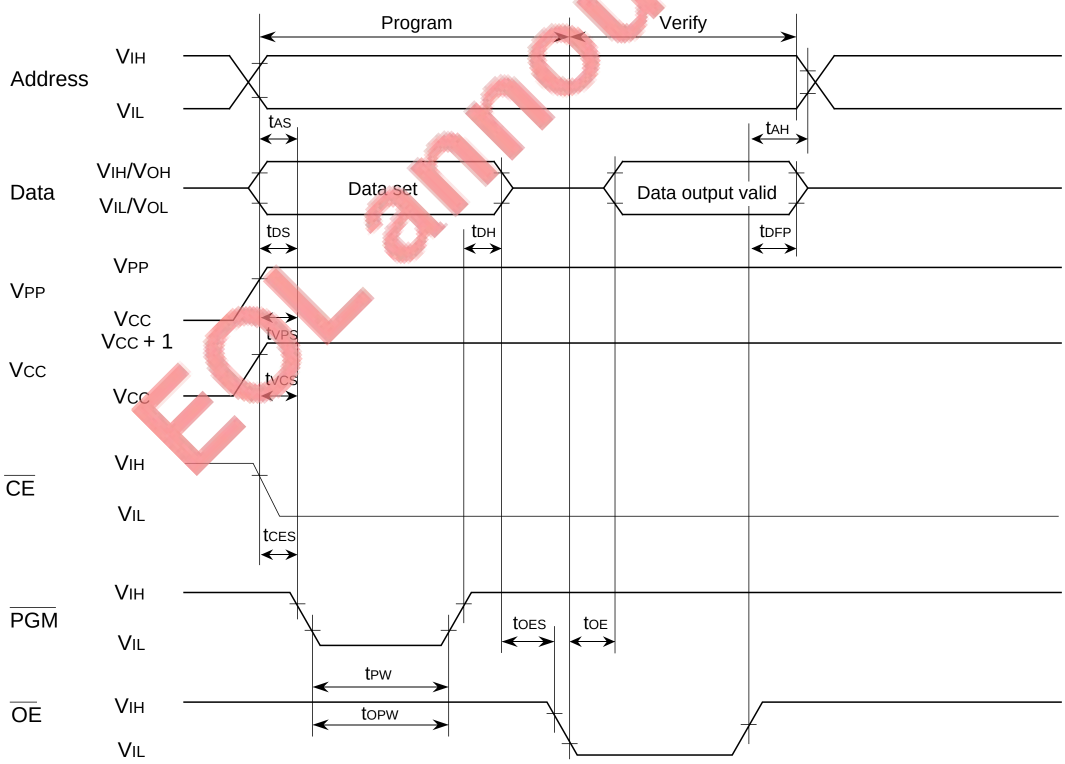
Fig. 19.3.3 Programming algorithm flow chart of 1M mode

19.3.3 Electrical characteristics of programming algorithm in 1M mode

AC electrical characteristics (Ta = 25 ± 5 °C, Vcc = 6 V ± 0.25 V, VPP = 12.5 ± 0.3 V, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
tAS	Address setup time	2			µs
tOES	OE setup time	2			µs
tDS	Data setup time	2			µs
tAH	Address hold time	0			µs
tDH	Data hold time	2			µs
tDFP	Output floating delay time after OE	0		130	ns
tVCS	Vcc setup time	2			µs
tVPS	VPP setup time	2			µs
tPW	PGM pulse width	0.19	0.2	0.21	ms
tOPW	Additional PGM pulse width	0.19		5.25	ms
tCES	CE setup time	2			µs
tOE	Data delay time after OE			150	ns

Programming timing diagram



Switching characteristics measuring conditions
- Input voltage : VIL = 0.45 V, VIH = 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

19.4 256K mode

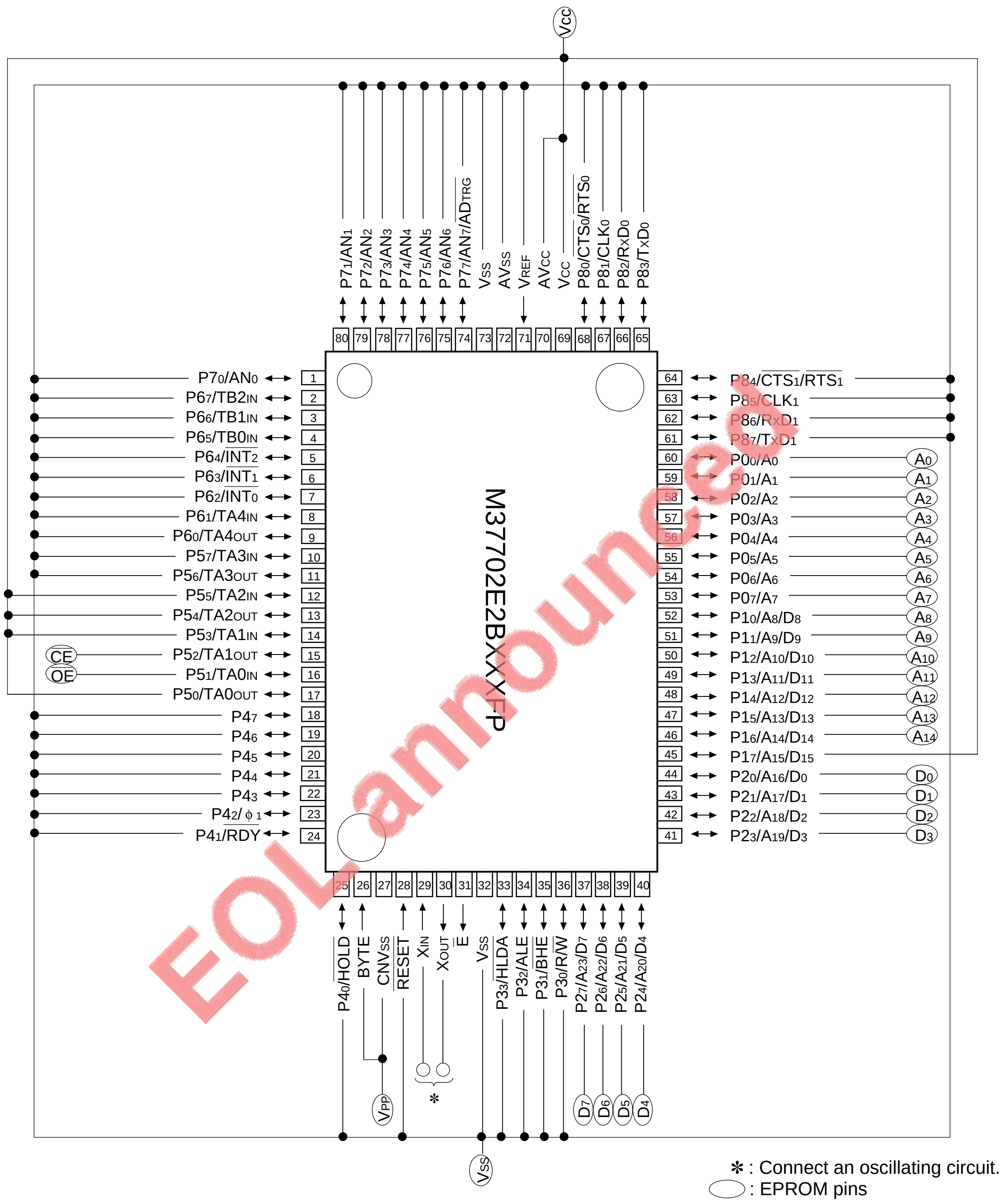
256K mode can perform reading/programming from and to the built-in PROM with the same manner as M5M27C256K. However, there is no device identification code. Accordingly, programming conditions must be set carefully.

Table 19.4.1 lists the pin correspondence with M5M27C256K. Figures 19.4.1 and 19.4.2 show the pin connections in 256K mode.

Table 19.4.1 Pin correspondence with M5M27C256K

	M37702E2BXXXFP (M37702E2BFP) M37702EHBFS	M5M27C256K
Vcc	Vcc	Vcc
V_{PP} input	CNVss, BYTE	V_{PP}
Vss	Vss	Vss
Address input	P0, P1	A_0–A_{14}
Data I/O	P2	D_0–D_7
$\overline{CE}$	$P5_2$	$\overline{CE}$
$\overline{OE}$	$P5_1$	$\overline{OE}$

●256K mode (top view)

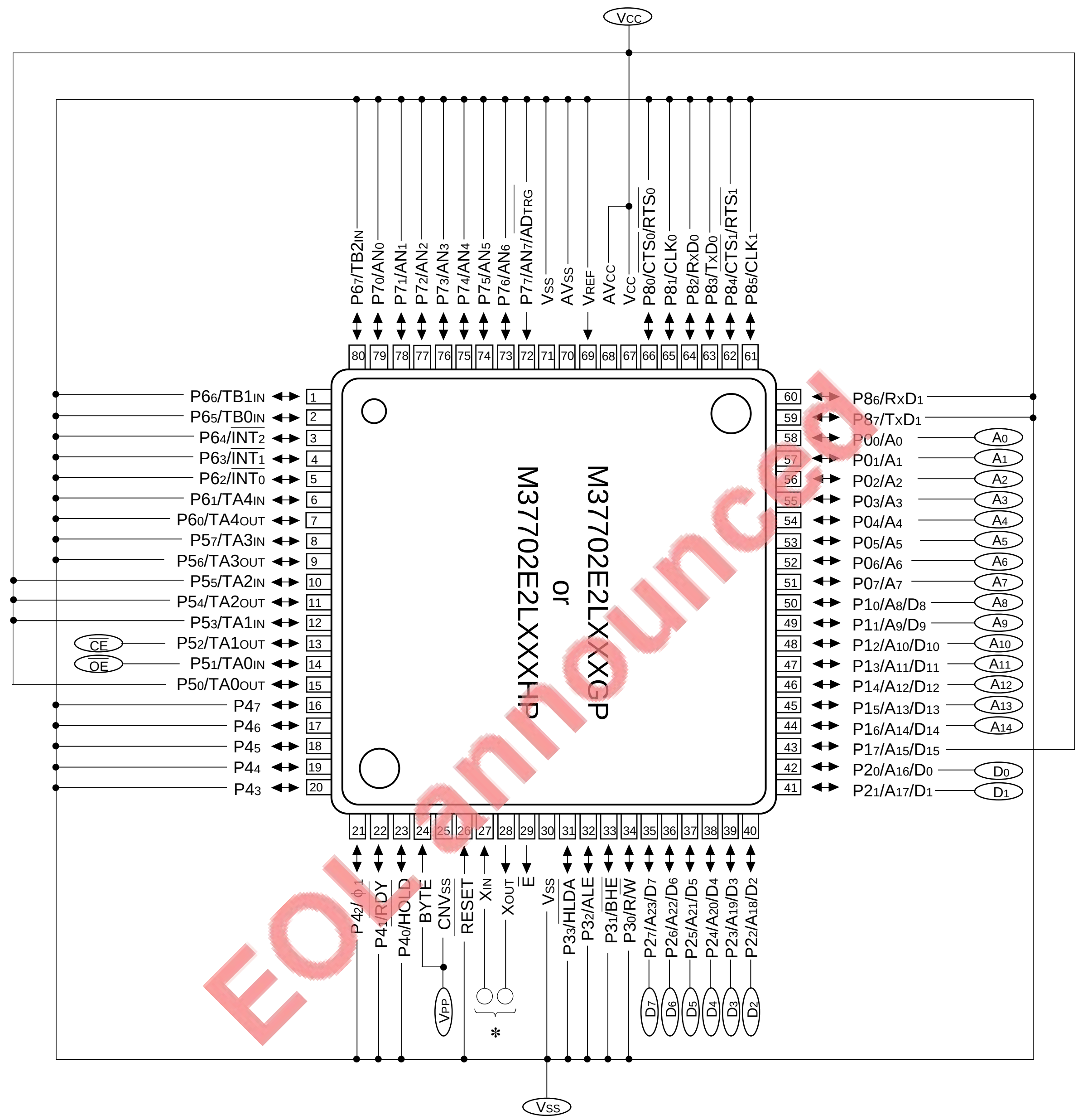


Outline 80P6N-A

Fig. 19.4.1 Pin connections in 256K mode (1)

19.4 256K mode

●256K mode (top view)



Outline 80P6S-A
Outline 80P6D-A

✻ : Connect an oscillating circuit.
⬭ : EPROM pins

Fig. 19.4.2 Pin connections in 256K mode (2)

19.4.1 Read/Program/Erase

Table 19.4.2 lists the built-in PROM state in 256K mode and each mode is described bellow.

(1) Read

When pins $\overline{CE}$ and $\overline{OE}$ are set to "L" level and an address is input to address input pins, the contents of the built-in PROM can be output from data I/O pins and read.
When pins $\overline{CE}$ and $\overline{OE}$ are set to "H" level, data I/O pins enter the floating state.

(2) Program (Write)

When pin $\overline{OE}$ is set to "H" level and V_{PP} level is applied to pin V_{PP}, 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 pin $\overline{CE}$ 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.

Table 19.4.2 Built-in PROM state in 256K mode

Mode \ Pin name	$\overline{CE}$	$\overline{OE}$	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}	12.5 V	6 V	Input
Program verify	V_{IH}	V_{IL}	12.5 V	6 V	Output
Program disable	V_{IH}	V_{IH}	12.5 V	6 V	Floating

× : It may be V_{IL} or V_{IH}.

19.4.2 Programming algorithm of 256K mode

Figure 19.4.3 shows the programming algorithm flow chart of 256K mode.

① Set Vcc = 6 V, VPP = 12.5 V, and address to 4000_{16}*.
② After applying a programming pulse of 1 ms, check whether data can be read or not.
③ If the data cannot be read, apply a programming pulse of 1 ms again.
④ Repeat the procedure, which consists of applying a programming pulse of 1 ms and read check, until the data can be read. Additionally, record the number of pulses applied (χ) before the data has been read.
⑤ Apply three times as many numbers as χ pulses (described in ④), that is, 3 × χ ms 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).

✻ : This applies to the M37702E2BXXXFP. Refer to **Table 19.1.1** about each write address of other products.

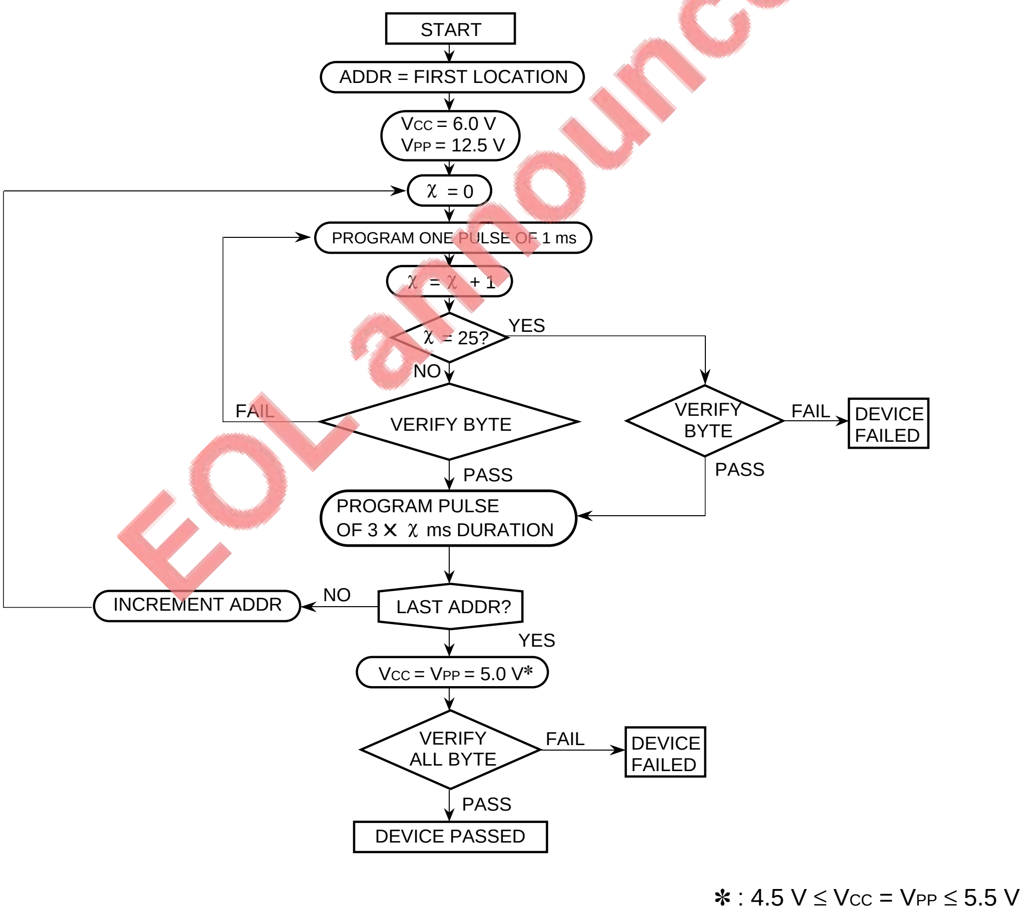


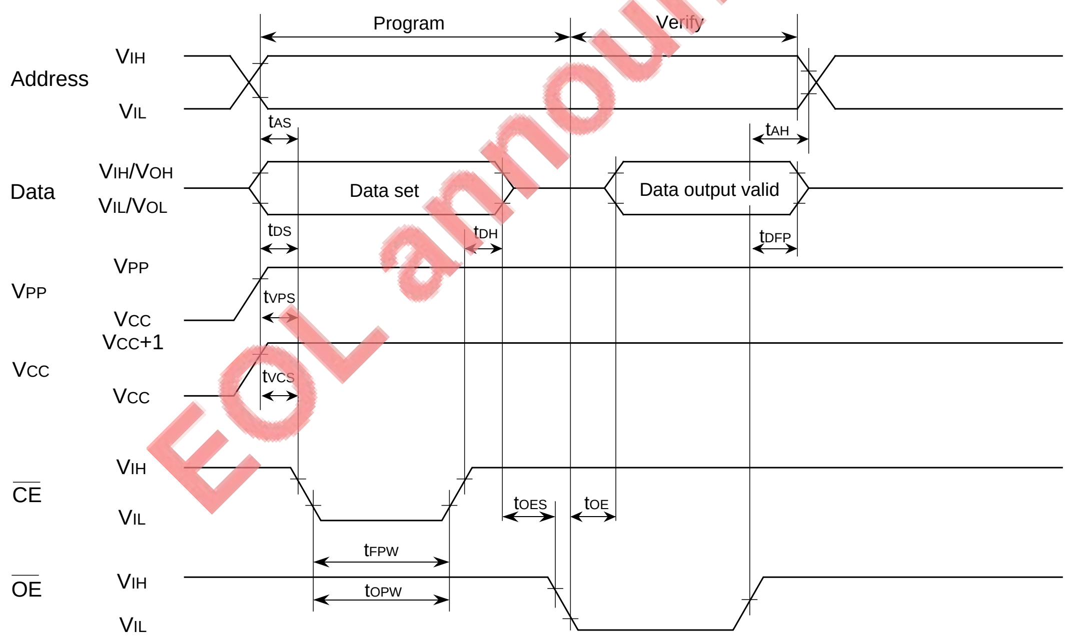
Fig. 19.4.3 Programming algorithm flow chart of 256K mode

19.4.3 Electrical characteristics of programming algorithm in 256K mode

AC electrical characteristics (Ta = 25 ± 5 °C, Vcc = 6 V ± 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_{FPW}	$\overline{CE}$ initial program pulse width	0.95	1	1.05	ms
t_{OPW}	Additional $\overline{CE}$ pulse width	2.85		78.75	ms
t_{OE}	Data delay time after $\overline{OE}$			150	ns

Programming timing diagram



19.5 Usage precaution

The usage precaution of PROM version is described bellow.

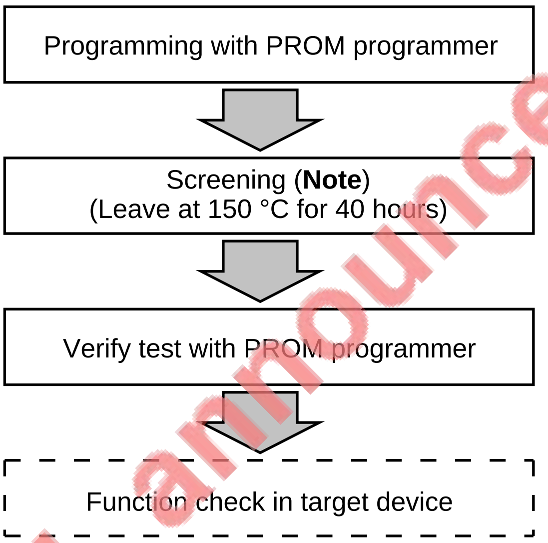
19.5.1 Precautions on all PROM versions

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.

19.5.2 Precautions on One time PROM version

One time PROM versions shipped in a blank, of which built-in PROMs are programmed by users, are also provided.

For these microcomputers, 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 19.5.1 before use.



Note: Never expose to 150 °C exceeding 100 hours.

Fig. 19.5.1 Programming and test flow for One Time PROM version

19.5.3 Precautions on EPROM version

(1) Cover transparent glass window

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.

A shield to cover the transparent window is available from Mitsubishi Electric Corporation. Be careful that the shield does not touch the EPROM lead pins.

(2) Erase

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.

(3) Usage

The EPROM version is a tool only for program development, evaluation only, and do not use it for the mass product run.

19.5.4 Bus timing and EPROM mode

The PROM versions shown in Tables 19.5.1 and 19.5.2 have the different bus timing from other PROM versions, mask ROM, external ROM versions. Additionally, they can use 256K mode as EPROM mode though its PROM size is 32 Kbytes or less.

Table 19.5.1 PROM versions having peculiar bus timing (16MHz version)

Type name \ Bus timing	$t_{pzx(E-P1Z)}$, $t_{pzx(E-P2Z)}$	
	$f(X_{IN}) \leq 8$ MHz	8MHz $< f(X_{IN}) \leq 16$ MHz
M37702E2AXXXFP M37702E2AFS M37702E4AXXXFP M37702E4AFS	Limits: 50 ns Formulas: $\frac{1 \times 10^9}{2 \times f(X_{IN})} - 12.5$	Limits: 25 ns Formulas: $\frac{1 \times 10^9}{2 \times f(X_{IN})} - 6.25$

Table 19.5.2 PROM versions having peculiar bus timing (Low voltage version)

Type name \ Bus timing	$t_{pzx(E-P1Z)}$, $t_{pzx(E-P2Z)}$	$t_{d(P0A-E)}$, $t_{d(P1A-E)}$, $t_{d(P2A-E)}$, $t_{d(BHE-E)}$, $t_{d(R/W-E)}$
M37702E2LXXXGP M37702E4LXXXFP	Limits: 50 ns Formulas: $\frac{1 \times 10^9}{2 \times f(X_{IN})} - 12.5$	Limits: 50 ns Formulas: $50 + \frac{1 \times 10^9}{2 \times f(X_{IN})} - 62.5$

(1) Bus timing

The limits and formulas of the PROM versions having the peculiar bus timing which is different from other PROM versions are shown in Tables 19.5.1 and 19.5.2.

When the user is planning to use the product shown in Tables 19.5.1 and 19.5.2 for evaluation or in early production and replace it later with the mask ROM version, we recommend to use the substitute shown in Table 19.5.3 for evaluation or in early production.

However, the substitute for the low voltage version has the larger ROM and RAM size. Make sure of its memory usage. The substitute for the 16 MHz version has the higher frequency of external clock input. There are no precaution about its operation.

Table 19.5.3 Substitutes

Type name to be used	Substitute	Remark
M37702E2AXXXFP	M37702E2BXXXFP	The substitute has the higher frequency of external clock input.
M37702E2AFS	M37702E2BFS	
M37702E4AXXXFP	M37702E4BXXXFP	
M37702E4AFS	M37702E4BFS	
M37702E2LXXXGP	M37702E4LXXXGP	The substitute has the larger ROM and RAM size.
M37702E4LXXXFP	M37702E6LXXXFP	

(2) EPROM mode

The products shown in Table 19.5.1 can use only 256K mode as the EPROM mode. Do not use 1M mode.

MEMORANDUM

CHAPTER 20

7703 GROUP

7703 GROUP

20.1 Description

This chapter describes the 7703 Group.
The 7703 Group has the same functions as the 7702 Group except for some functions. This chapter mainly describes the differences between the 7703 and 7702 Groups. Refer to the relevant descriptions of the 7702 Group about the common functions.

20.1 Description

The 16-bit single-chip microcomputers 7703 Group is suitable for office, business, and industrial equipment controllers that require high-speed processing.
These microcomputers develop with the M37703M2BXXXSP as the base chip. This manual describes the functions about the M37703M2BXXXSP unless there is a specific difference and the M37703M2BXXXXSP is referred to as “M37703.”

20.2 Performance overview

Table 20.2.1 lists the performance overview of the M37703.

Table 20.2.1 M37703 performance overview

Parameters		Functions
Number of basic instructions		103
Instruction execution time	M37703M2BXXXSP	160 ns (the minimum instruction at $f(X_{IN})$ = 25 MHz)
	M37703M2AXXXSP	250 ns (the minimum instruction at $f(X_{IN})$ = 16 MHz)
External clock input frequency $f(X_{IN})$	M37703M2BXXXSP	25 MHz (maximum)
	M37703M2AXXXSP	16 MHz (maximum)
Memory size	ROM	16384 bytes
	RAM	512 bytes
Programmable Input/Output ports	P0, P1, P2, P5	8 bits × 4
	P8	6 bits × 1
	P4, P6, P7	4 bits × 3
	P3	3 bits × 1
Multifunction timers	TA0–TA4	16 bits × 5; With I/O function× 4
	TB0–TB2	16 bits × 3; With Input function× 1
Serial I/O	UART0, UART1	UART × 2 (UART0 also as clock synchronous serial I/O)
A-D converter		8-bit successive approximation method × 1 (4 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		95 mW (at $f(X_{IN})$ = 25 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 SDIP

20.3 Pin configuration

Figure 20.3.1 shows the M37703M2BXXXSP pin configuration.

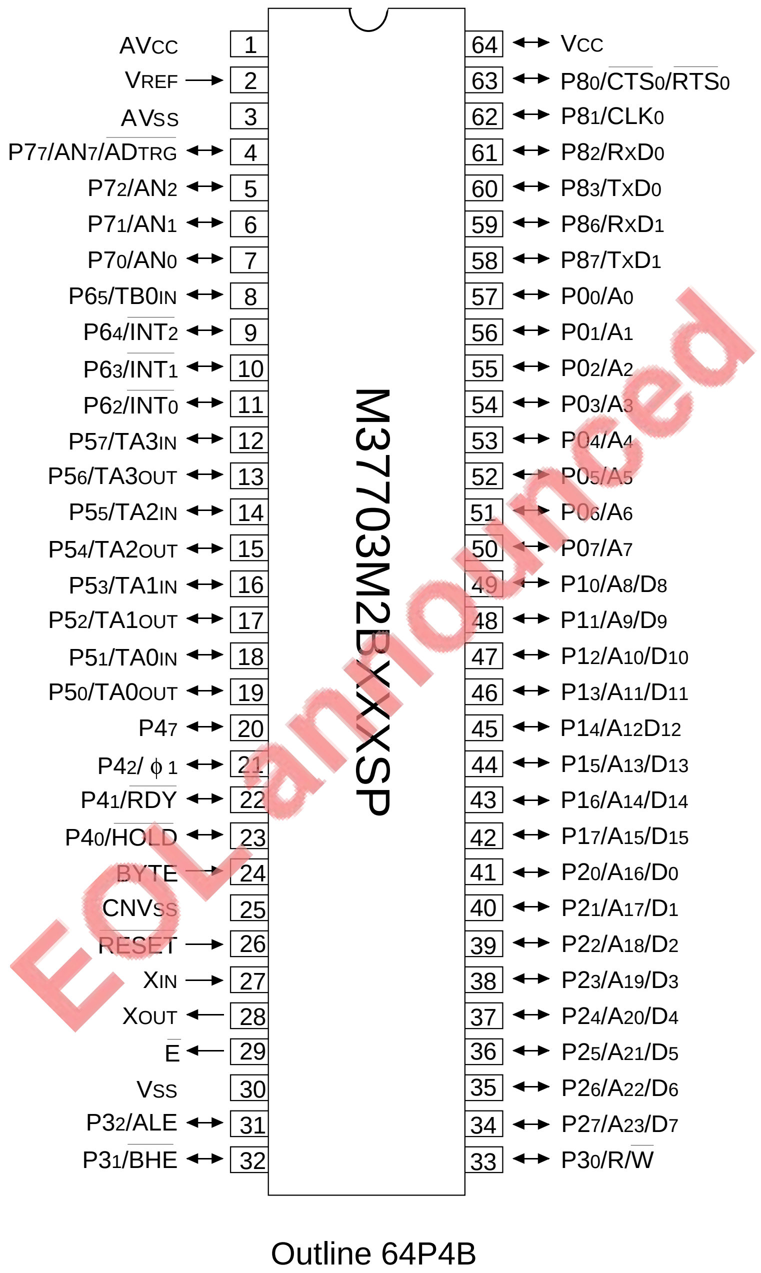


Fig. 20.3.1 M37703M2BXXXSP pin configuration (top view)

20.4 Functional description

The M37703 has the same internal circuit as the M37702. The control registers in the SFR area and the memory assignment are also the same. However, part of the M37703 functions varies from the M37702's, because the number of M37703's pins is 64 pins.
Table 20.4.1 lists the differences between the M37703 and M37702.

This paragraph describes the differences from the M37702. Refer to the relevant functional descriptions of the M37702 about others.

Table 20.4.1 Differences between the M37703 and M37702

Parameters		M37703M2BXXXSP	M37702M2BXXXFP
Programmable I/O port		53 (In single-chip mode)	68 (In single-chip mode)
	Port P0	8 bits	8 bits
	Port P1	8 bits	8 bits
	Port P2	8 bits	8 bits
	Port P3	3 bits; Without $P3_3/\overline{HLDA}$ pin	4 bits
	Port P4	4 bits; Without $P4_3$ to $P4_6$ pins	8 bits
	Port P5	8 bits	8 bits
	Port P6	4 bits; Without $P6_0$, $P6_1$, $P6_6$, and $P6_7$ pins	8 bits
	Port P7	4 bits; Without $P7_3$ to $P7_6$ pins	8 bits
	Port P8	6 bits; Without $P8_4$ and $P8_5$ pins	8 bits
Timer		16 bits × 8	16 bits × 8
	TA0	With timer I/O pins: Input pin (TAi_{IN}); Output pin (TAi_{OUT}) (i = 0 to 3)	With timer I/O pins: Input pin (TAj_{IN}); Output pin (TAj_{OUT}) (j = 0 to 4)
	TA1		
	TA2		
	TA3		
	TA4	Internal timer; Without I/O pins	
	TB0	With timer input pin ($TB0_{IN}$)	With timer input pins: Input pin (TBk_{IN}) (k = 0 to 2)
	TB1	Internal timer; Without I/O pins	
	TB2		
Serial I/O		2	2
	UART0	Clock synchronous or Clock asynchronous	Clock synchronous or Clock asynchronous
	UART1	Clock asynchronous	Clock synchronous or Clock asynchronous
A-D converter		Resolution 8 bits × 1 Analog input pin 4 channels: AN_0, AN_1, AN_2, AN_7 pins; Without AN_3 to AN_6 pins)	Resolution 8 bits × 1 Analog input pin 8 channels: AN_0 to AN_7 pins
Package		64-pin plastic molded SDIP; 64P4B	80-pin plastic molded QFP; 80P6N-A

20.4.1 I/O pin

The M37703 does not have the following pins of the M37702:

•Port $P3_3$
•Ports $P4_3$ to $P4_6$
•Ports $P6_0$, $P6_1$, $P6_6$, $P6_7$
•Ports $P7_3$ to $P7_6$
•Ports $P8_4$, $P8_5$

(1) Port direction register

Fix the bits of port Pi (i = 3, 4, 6, 7, 8) direction register which do not have the corresponding pins to “1.” All products of the M37703 need this procedure. Do it regardless of the product type and the used mode.
All bits of port Pi direction register are cleared to “0” after reset. Accordingly, follow the procedure shown by Figure 20.4.1 in the initial setting program after reset. Do not write “0” after that to the bits to be fixed to “1.”

Paragraph “1.3.1 Example for processing unused pins” explains the examples when there are pins, however, those pins are not used. The above explanation is independent of that example explanation.

(2) Memory expansion and Microprocessor modes

The M37703 does not have the $\overline{HLDA}$ pin, so that the $\overline{HLDA}$ signal cannot be used in those modes.

●Be sure to set “1” to the bit indicated by using “1”.
Though these bits do not have the corresponding pins, follow the above procedure.
The above procedure is necessary whether or not other programmable I/O ports are used.

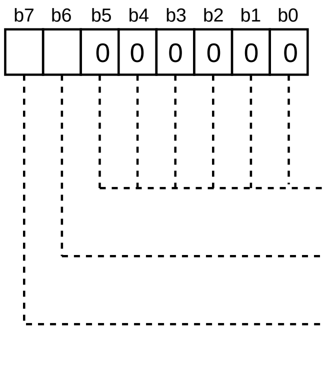
b7	b6	b5	b4	b3	b2	b1	b0	
(shaded)	(shaded)	(shaded)	(shaded)	1				Port P3 direction register (address 9_{16})
	1	1	1	1				Port P4 direction register (address C_{16})
1	1					1	1	Port P6 direction register (address 10_{16})
	1	1	1	1				Port P7 direction register (address 11_{16})
		1	1					Port P8 direction register (address 14_{16})

Notes 1: When executing the instruction to write to bits 4 to 7 of Port P3 direction register, the value cannot be written into them.
When reading to those bits, “0” is read.
2: The bits which are not indicated by using “1” and bits 4 to 7 of Port P3 direction register function as a programmable I/O port. Just as in ports P0–P2 and P5, set “0” when using as an input port, and set “1” when using as an output port.

Fig. 20.4.1 Procedure of port Pi (i = 3, 4, 6, 7, 8) direction register

20.4.2 Timer A

The M37703 does not have the I/O functions of Timer A4. It can be used only in the timer mode. Fix bits 5 to 0 of the timer A4 mode register to "000000_2."
Figure 20.4.2 shows the structure of the timer A4 mode register.



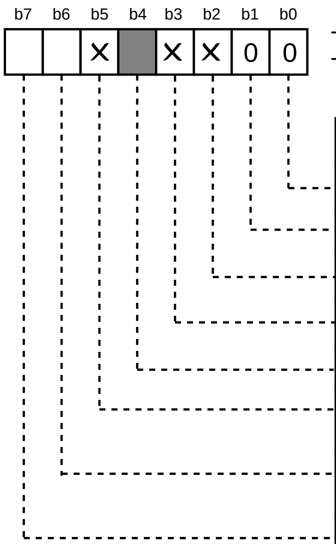
Timer A4 mode register (Addresses $5A_{16}$)

Bit	Bit name	Functions	At reset	RW
5 to 0	Fix these bits to "0". In timer mode, without pulse output and gate functions.		0	RW
6	Count source select bits	b7 b6 0 0 : f_2 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	0	RW
7			0	RW

Fig. 20.4.2 Structure of timer A4 mode register

20.4.3 Timer B

The M37703 does not have the input functions of Timers B1 and B2. They can be used only in the timer mode. Fix bits 1 and 0 of the timer B1 and B2 mode registers to "00_2."
Figure 20.4.3 shows the structure of the timer B1 and B2 mode registers.



Timer B1 mode register (Addresses $5C_{16}$)
Timer B2 mode register (Addresses $5D_{16}$)

Bit	Bit name	Functions	At reset	RW
0	Fix these bits to "0". In 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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	0	RW
7			0	RW

Fig. 20.4.3 Structure of timer B1 and B2 mode registers

20.4.4 Serial I/O

The M37703's UART1 can be used only in the clock asynchronous serial I/O mode, UART mode. It cannot be used in the clock synchronous serial I/O mode. Do not set the serial I/O mode select bits (bits 2 to 0 at address 38_{16}) to "001_2" to select the clock synchronous serial I/O mode.
Figure 20.4.4 shows the structure of the UART1 transmit/receive mode register.

(1) CLK1 pin

The M37703 does not have the CLK_1 pin. Set the internal/external clock select bit (bit 3 at address 38_{16}) to "0" to select the internal clock.

b7 b6 b5 b4 b3 b2 b1 b0
(b3 = 0)

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: Not selected 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	Fix these bits to "0". (To select internal 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. 20.4.4 Structure of UART1 transmit/receive mode register

(2) $\overline{CTS_1}$/$\overline{RTS_1}$ pin

The M37703 does not have the $\overline{CTS_1}$/$\overline{RTS_1}$ pin. Fix the $\overline{CTS}$/$\overline{RTS}$ select bit (bit 2 at address $3C_{16}$) to "1".

Figure 20.4.5 shows the structure of the UART1 transmit/receive control register 0 and Figure 20.4.6 shows the structure of the port P8 direction register when using UART1.

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

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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	0	RW
1			0	RW
2	Fix this bit to "1."		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
7 to 4	Nothing is assigned.		Undefined	–

Fig. 20.4.5 Structure of UART1 transmit/receive control register 0

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

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 pin $P8_2$ as serial data input pin (RxD_0),set bit 2 to "0."	0	RW
1	CLK_0 pin		0	RW
2	RxD_0 pin		0	RW
3	TxD_0 pin		0	RW
4	Fix these bits to "1."		0	RW
5			0	RW
6	RxD_1 pin	When using pin $P8_6$ as serial data input pin (RxD_1),set bit 6 to "0."	0	RW
7	TxD_1 pin		0	RW

(shaded) : Bits 0 to 3 are not used in UART1.

Fig. 20.4.6 Structure of port P8 direction register when using UART1

20.4.5 A-D converter

The M37703's analog inputs are 4 channels: AN_0 to AN_2 and AN_7.

(1) One-shot and Repeat modes

Set the analog input select bits (bits 2 to 0 at address $1E_{16}$) to one of "000_2", "001_2", "010_2" and "111_2."

Set the bits of the port P7 direction register which do not have pins corresponding to analog inputs AN_3 to AN_6 to "1" to make them output mode.

Figure 20.4.7 shows the structure of the A-D control register and Figure 20.4.8 shows the structure of the port P7 direction register when using A-D converter.

(2) Single sweep and Repeat sweep modes

Set the bits of the port P7 direction register corresponding to AN_0 to AN_2 and AN_7 pins to "0" to make them input mode.

Set the bits of the port P7 direction register which do not have pins corresponding to analog inputs AN_3 to AN_6 to "1" to make them output mode. The A-D register contents corresponding to analog inputs AN_3 to AN_6, which do not have their pins, become undefined.

b7 b6 b5 b4 b3 b2 b1 b0

A-D control register (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 : Not selected 1 0 0 : Not selected 1 0 1 : Not selected 1 1 0 : Not selected 1 1 1 : AN_7 selected **(Note 2)**	Undefined	RW
1			Undefined	RW
2			Undefined	RW
3	A-D operation mode select bits	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode	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 : f_2 divided by 4 1 : f_2 divided by 2	0	RW

Notes 1: These bits are ignored in the single sweep and repeat sweep 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: Writing to each bit except bit 6 of the A-D control register must be performed while the A-D converter halts.

Fig. 20.4.7 Structure of A-D control register

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

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

Bit	Corresponding bit name	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	Fix these bits to "1."		0	RW
4			0	RW
5			0	RW
6			0	RW
7	AN_7 pin/$\overline{AD_{TRG}}$ pin	0: Input mode 1: Output mode When using this pin as A-D converter's input pin or external trigger input pin, set this bit to "0."	0	RW

Fig. 20.4.8 Structure of the port P7 direction register when using A-D converter

20.5 Electrical characteristics

The M37703 electrical characteristics is the same as the M37702's except for the absolute maximum ratings shown in Table 20.5.1 and the parameters of not existing pins of the M37703.
Refer to "**Chapter 15. ELECTRICAL CHARACTERISTICS**."

Additionally, the M37703 standard characteristics is the same as the M37702's and refer to "**Chapter 16. STANDARD CHARACTERISTICS**."

Table 20.5.1 Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
P_d	Power dissipation	Ta = 25 °C	1000	mW

Note: The electrical characteristics except above is the same as the M37702's.

20.6 PROM version

In the PROM version, programming to the built-in PROM can be performed by using a general-purpose PROM programmer and a programming adapter, which is suitable for the used microcomputer.
The PROM version of M37703 is the one time PROM version. Programming to the PROM can be performed once in this version.
The one time PROM version has the same functions as the mask ROM version except that the former has a built-in PROM. Table 20.6.1 lists the write address of PROM version.

The M37703 does not have the EPROM version. Use the EPROM version of M37702 with a pitch converter for the M37703 evaluation.

Table 20.6.1 Write address of PROM version

Type name	PROM size (Byte)	RAM size (Byte)	Write address 1M mode	Write address 256K mode
M37703E2BXXXSP	16K	512	$1C000_{16}$ to $1FFFF_{16}$	4000_{16} to $7FFF_{16}$
M37703E2AXXXSP (**Note 1**)				
M37703E4BXXXSP	32K	2048	18000_{16} to $1FFFF_{16}$	0000_{16} to $7FFF_{16}$
M37703E4AXXXSP (**Note 1**)				

Notes 1: Refer also to section "**20.6.2 Bus timing and EPROM mode**."
2: A blank product of the one time PROM version does not have the ROM number, which is printed on the **XXX** position. For example, M37703E2BSP.

20.6.1 EPROM mode

The EPROM mode of M37703 is the same as the M37702's. Refer to section "**19.2 EPROM mode**."
The pin connections vary from the M37702's. Figure 20.6.1 shows the pin connections in EPROM mode.

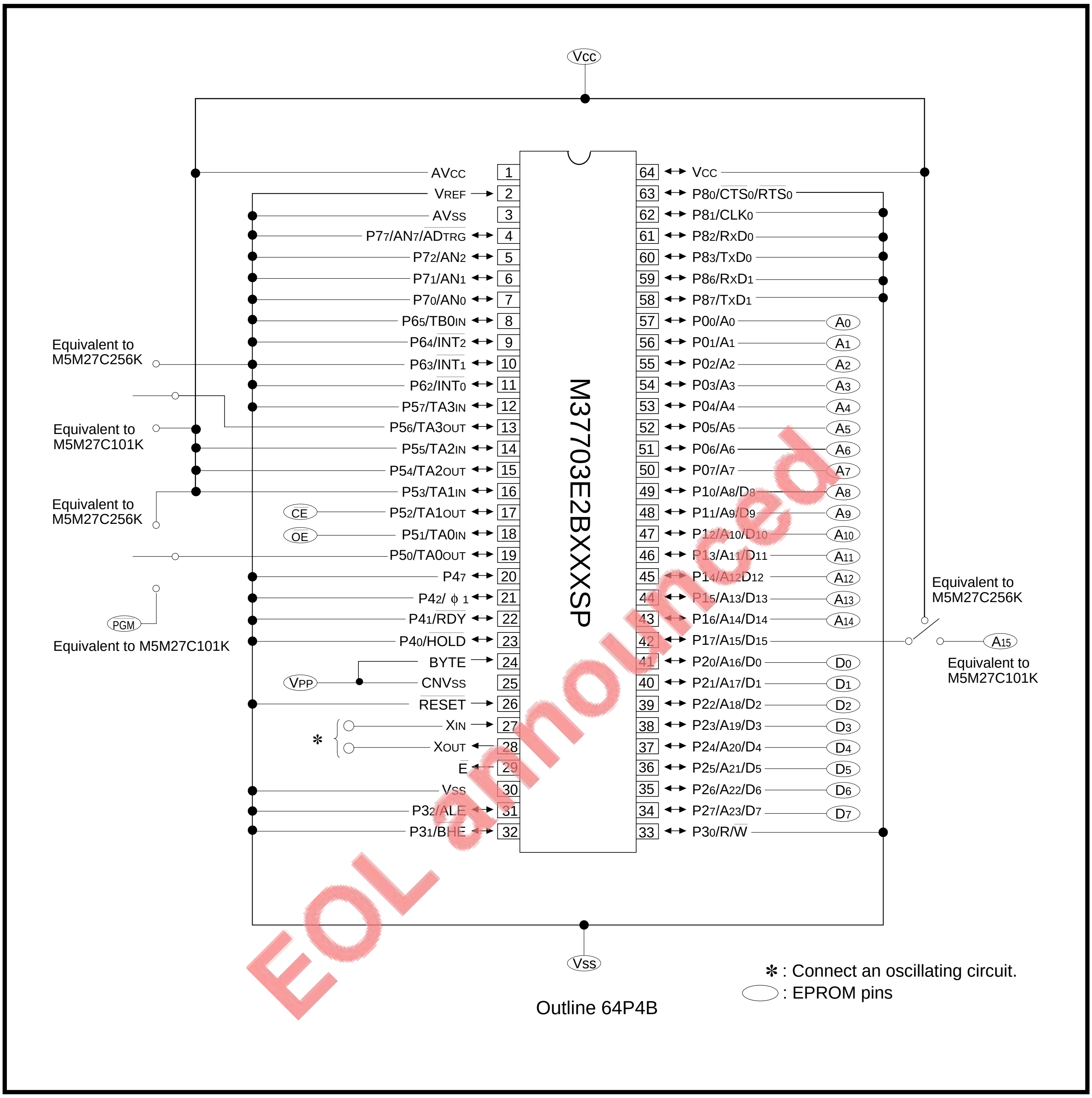


Fig. 20.6.1 Pin connections in EPROM mode

20.6.2 Bus timing and EPROM mode

The PROM versions shown in Table 20.6.2 have the different bus timing from other PROM versions, mask ROM, external ROM versions. Additionally, they can use only 256K mode as the EPROM mode though its PROM size is 32 Kbytes or less.

Table 20.6.2 PROM versions having peculiar bus timing

Type name \ Bus timing	$t_{pzx(E - P1Z)}$, $t_{pzx(E - P2Z)}$	
	$f(X_{IN}) \leq 8$ MHz	8MHz $< f(X_{IN}) \leq 16$ MHz
M37703E2AXXXSP	Limits: 50 ns	Limits: 25 ns
M37703E4AXXXSP	Formulas: $\frac{1 \times 10^9}{2 \times f(X_{IN})} - 12.5$	Formulas: $\frac{1 \times 10^9}{2 \times f(X_{IN})} - 6.25$

(1) Bus timing

The limits and formulas of the PROM versions having the peculiar bus timing which is different from other PROM versions are shown in Table 20.6.2.
When the user is planning to use the product shown in Table 20.6.2 for evaluation or in early production and replace it later with the mask ROM version, we recommend to use the substitute shown in Table 20.6.3 for evaluation or in early production.
However, the substitute version has the higher frequency of external clock input. There are no precaution about its operation.

Table 20.6.3 Substitutes

Type name to be used	Substitute	Remark
M37703E2AXXXSP	M37703E2BXXXSP	The substitute has the higher frequency of external clock input.
M37703E4AXXXSP	M37703E4BXXXSP	

(2) EPROM mode

The products shown in Table 20.6.2 can use only 256K mode as the EPROM mode. Do not use 1M mode.

MEMORANDUM

APPENDIX

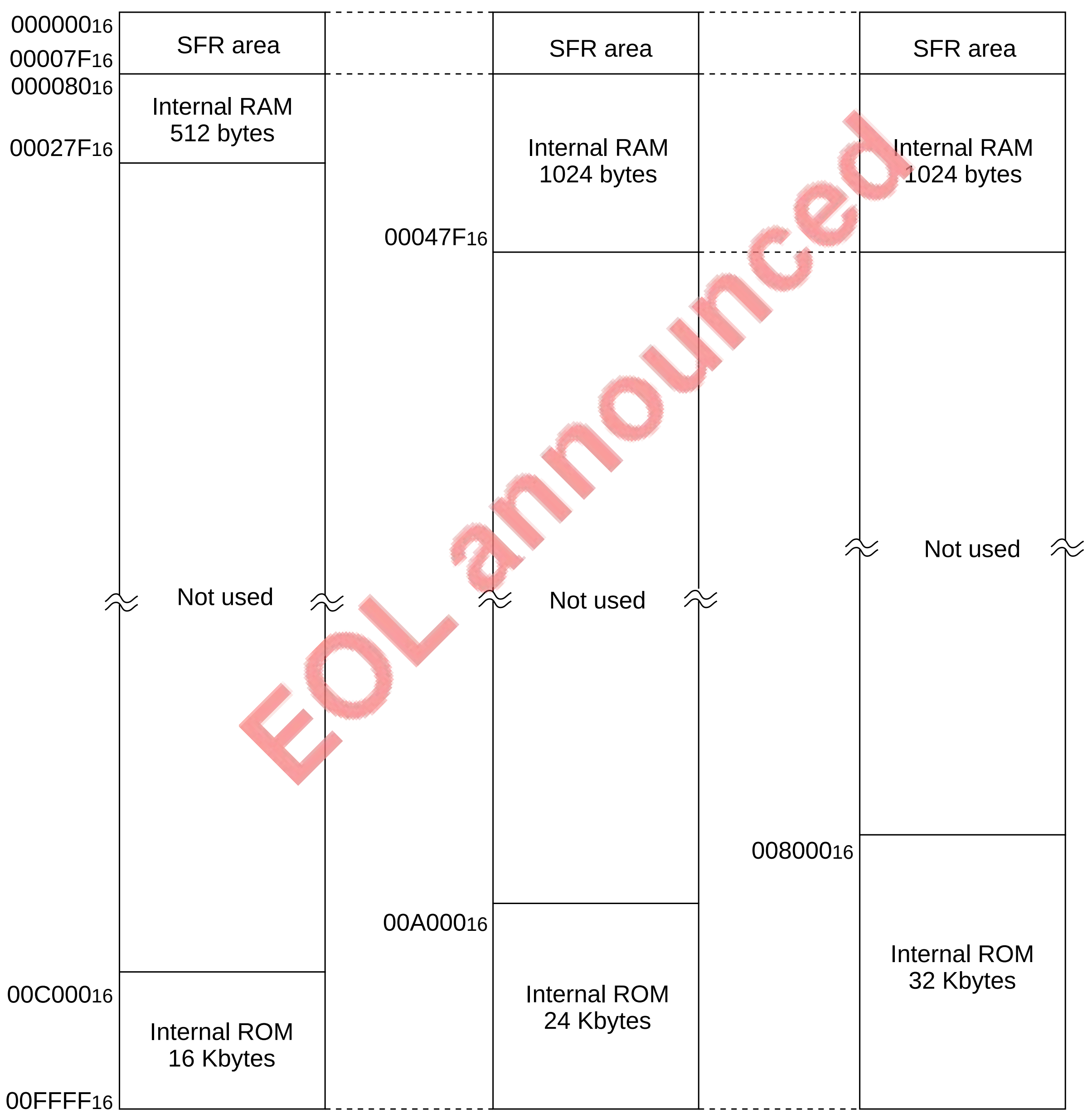
Appendix 1. Memory assignment

Figure 1 to Figure 5 show the memory assignment of the M37702 and the M37703 in each processor mode. Refer to the memory assignment whose type name show suitable memory type and memory size.

M37702M2BXXXFP
└ **Memory type and memory size**

● Single-chip mode

- Memory size/type......M2, E2
- Memory size/type......M3 **(Note)**
- Memory size/type......MD **(Note)**



Note: These can be used only in the single-chip mode.

Fig. 1 Memory assignment during single-chip mode (1)

● Single-chip mode

• Memory size/type......M4, E4 • Memory size/type......M6, E6 • Memory size/type......M8, E8

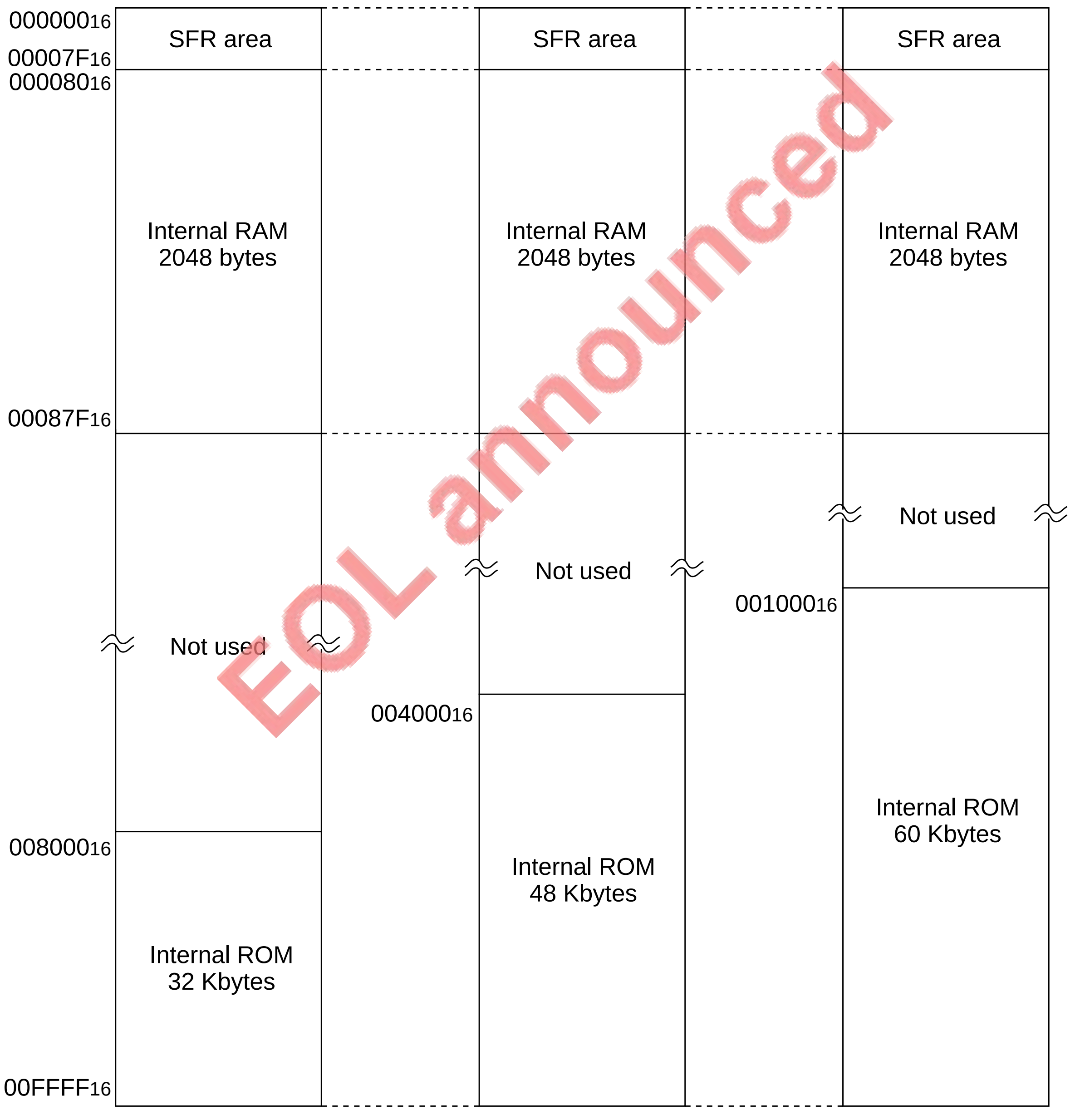


Fig. 2 Memory assignment during single-chip mode (2)

● Memory expansion mode

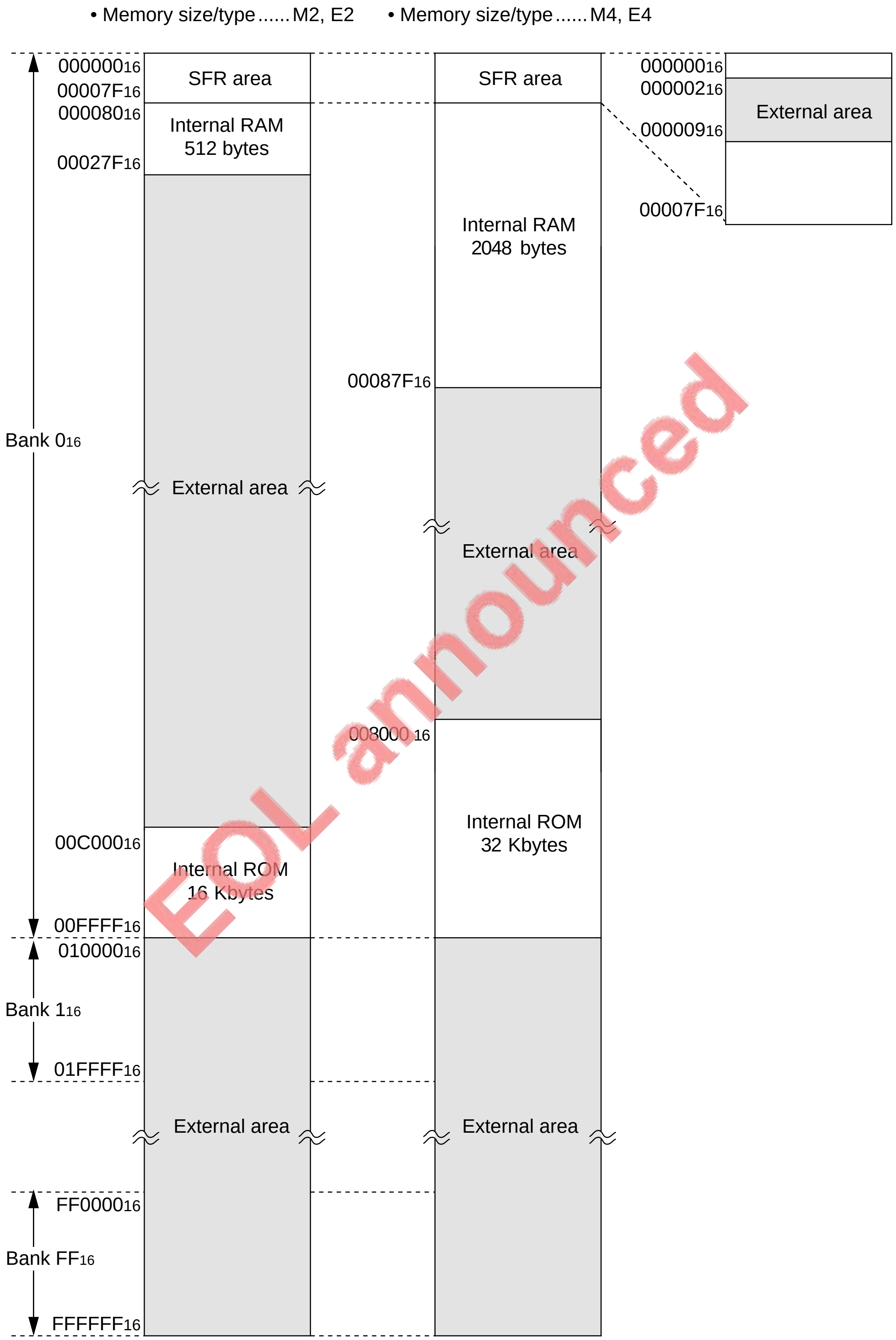


Fig. 3 Memory assignment during memory expansion mode (1)

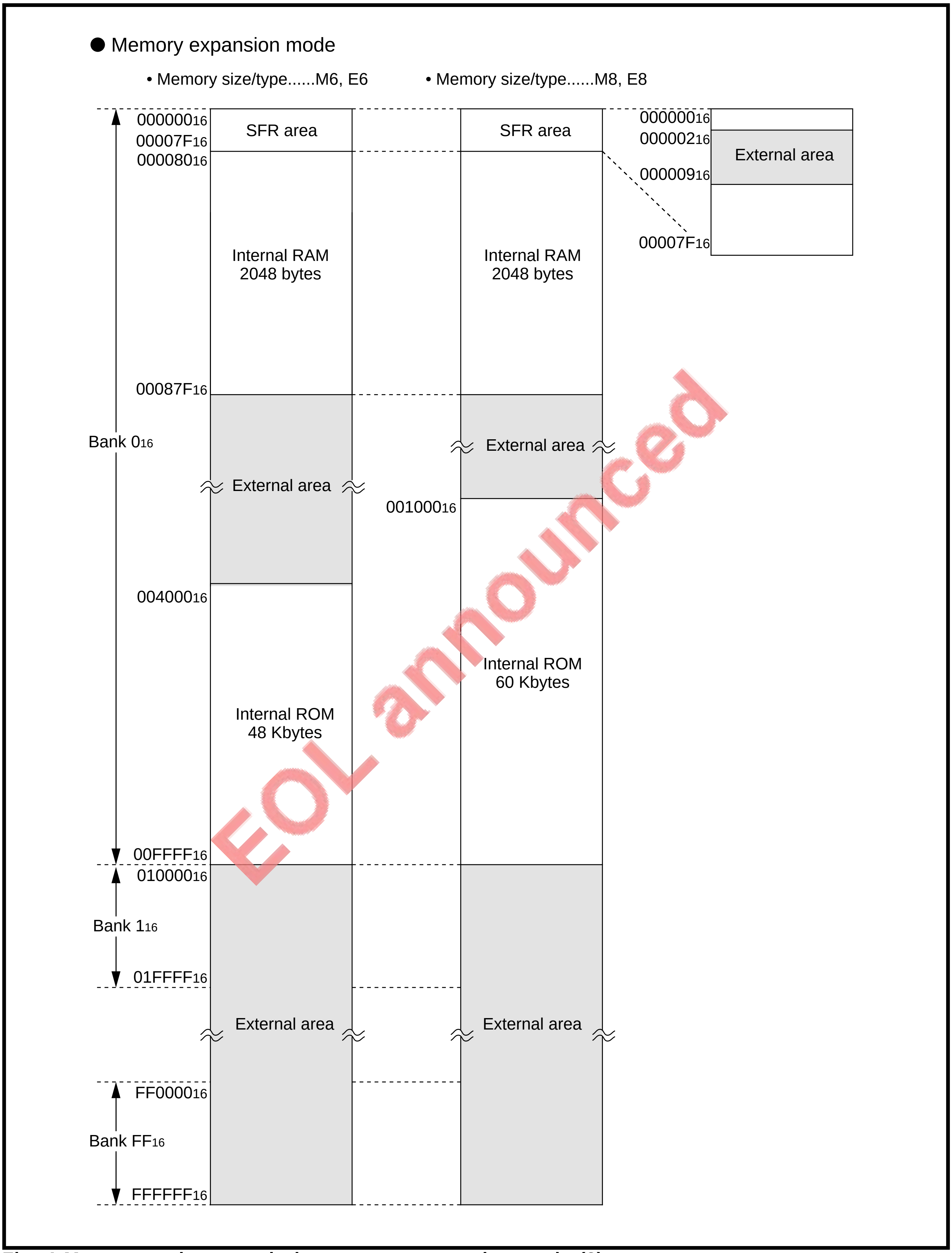


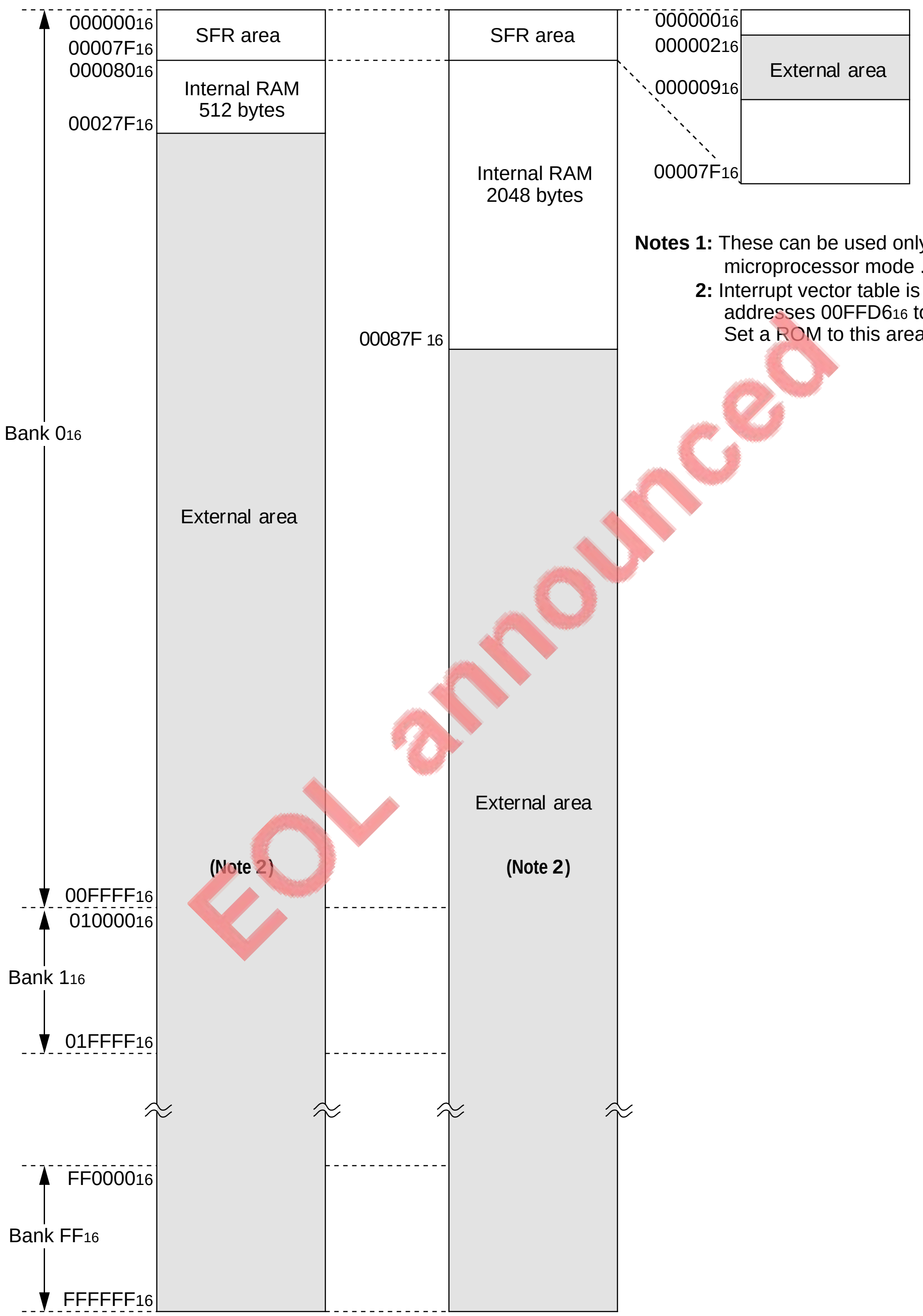
Fig. 4 Memory assignment during memory expansion mode (2)

Appendix 1. Memory assignment

● Microprocessor mode

• Memory size/type ...... M2, E2, S1 (Note 1)

• Memory size/type ...... M4, E4, M6, E6, M8, E8, S4 (Note 1)



Notes 1: These can be used only in the microprocessor mode .

2: Interrupt vector table is assigned to addresses $00FFD6_{16}$ to $00FFFF_{16}$. Set a ROM to this area.

Fig. 5 Memory assignment during microprocessor mode

Appendix 2. Memory assignment in SFR area

Figures 6 to 9 show the memory assignment in SFR area.
The significations which are used in Figures 6 to 9 is described below.

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 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)	
0_{16}		(nothing assigned)	?	
1_{16}		(nothing assigned)	?	
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	b7–b4: nothing assigned; b3–b0: RW	0 0 0 0 ?	
8_{16}	Port P2 direction register	RW	00_{16}	
9_{16}	Port P3 direction register	b7–b4: nothing assigned; b3–b0: 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}		(nothing assigned)	?	
14_{16}	Port P8 direction register	RW	00_{16}	✻
15_{16}		(nothing assigned)	?	
16_{16}		(nothing assigned)	?	
17_{16}		(nothing assigned)	?	
18_{16}		(nothing assigned)	?	
19_{16}		(nothing assigned)	?	
$1A_{16}$		(nothing assigned)	?	
$1B_{16}$		(nothing assigned)	?	
$1C_{16}$		(nothing assigned)	?	
$1D_{16}$		(nothing assigned)	?	
$1E_{16}$	A-D control register	RW	0 0 0 0 0 ? ? ?	
$1F_{16}$	A-D sweep pin select register	b7–b2: nothing assigned; b1–b0: RW	? ? ? ? ? ? 1 1	

✻ : In the 7703 Group, set "1" to the bit of which corresponding pin is nothing. (Refer to section **"20.4.1 Input/Output pins."**)

Fig. 6 Memory assignment in SFR area (1)

APPENDIX

Appendix 2. Memory assignment in SFR area

Address	Register name	Access characteristics (b7 … b0)	State immediately after a reset (b7 … b0)
20_{16}	A-D register 0	RO	?
21_{16}			?
22_{16}	A-D register 1	RO	?
23_{16}			?
24_{16}	A-D register 2	RO	?
25_{16}			?
26_{16}	A-D register 3	RO	?
27_{16}			?
28_{16}	A-D register 4	RO	?
29_{16}			?
$2A_{16}$	A-D register 5	RO	?
$2B_{16}$			?
$2C_{16}$	A-D register 6	RO	?
$2D_{16}$			?
$2E_{16}$	A-D register 7	RO	?
$2F_{16}$			?
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: –, b0: WO	?
34_{16}	UART0 transmit/receive control register 0	b7–b4: –, b3: RO, b2–b0: RW	? ? ? ? 1 0 0 0
35_{16}	UART0 transmit/receive control register 1	b7–b3: RO, b2: RW, b1: RO, b0: RW	0 0 0 0 0 0 1 0
36_{16}	UART0 receive buffer register	RO	?
37_{16}		b7–b1: –, b0: 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: –, b0: WO	?
$3C_{16}$	UART1 transmit/receive control register 0	b7–b4: –, b3: RO, b2–b0: RW	? ? ? ? 1 0 0 0
$3D_{16}$	UART1 transmit/receive control register 1	b7–b3: RO, b2: RW, b1: RO, b0: RW	0 0 0 0 0 0 1 0
$3E_{16}$	UART1 receive buffer register	RO	?
$3F_{16}$		b7–b1: –, b0: RO	0 0 0 0 0 0 0 ?

Fig. 7 Memory assignment in SFR area (2)

Address	Register name	Access characteristics (b7 … b0)	State immediately after a reset (b7 … b0)
40₁₆	Count start register	RW	00₁₆
41₁₆		(shaded)	?
42₁₆	One-shot start register	b7–b5: (shaded); b4–b0: WO	b7–b5: ?; b4–b0: 0 0 0 0 0
43₁₆		(shaded)	?
44₁₆	Up-down register	b7–b5: WO; b4–b0: RW	0 0 0 0 0 0 0 0
45₁₆		(shaded)	?
46₁₆	Timer A0 register	✻1	?
47₁₆		✻1	?
48₁₆	Timer A1 register	✻1	?
49₁₆		✻1	?
4A₁₆	Timer A2 register	✻1	?
4B₁₆		✻1	?
4C₁₆	Timer A3 register	✻1	?
4D₁₆		✻1	?
4E₁₆	Timer A4 register	✻1	?
4F₁₆		✻1	?
50₁₆	Timer B0 register	✻2	?
51₁₆		✻2	?
52₁₆	Timer B1 register	✻2	?
53₁₆		✻2	?
54₁₆	Timer B2 register	✻2	?
55₁₆		✻2	?
56₁₆	Timer A0 mode register	RW	00₁₆
57₁₆	Timer A1 mode register	RW	00₁₆
58₁₆	Timer A2 mode register	RW	00₁₆
59₁₆	Timer A3 mode register	RW	00₁₆
5A₁₆	Timer A4 mode register	RW	00₁₆
5B₁₆	Timer B0 mode register	b7–b6: RW; b5: ✻3; b4: (shaded); b3–b0: RW	0 0 ? ? 0 0 0 0
5C₁₆	Timer B1 mode register	b7–b6: RW; b5: ✻3; b4: (shaded); b3–b0: RW	0 0 ? ? 0 0 0 0
5D₁₆	Timer B2 mode register	b7–b6: RW; b5: ✻3; b4: (shaded); b3–b0: RW	0 0 ? ? 0 0 0 0
5E₁₆	Processor mode register	b7–b4: RW; b3: WO; b2: RW; b1: ✻4; b0: RW	0 0 0 0 0 0 ✻4 0
5F₁₆		(shaded)	?

✻1: The access characteristics at addresses 46₁₆ to 4F₁₆ varies according to Timer A's operating mode. (Refer to **"Chapter 5. TIMER A."**)
✻2: The access characteristics at addresses 50₁₆ to 55₁₆ varies according to Timer B's operating mode. (Refer to **"Chapter 6. TIMER B."**)
✻3: The access characteristics of bit 5 at addresses 5B₁₆ to 5D₁₆ varies according to Timer B's operating mode. (Refer to **"Chapter 6. TIMER B."**)
✻4: The access characteristics of bit 1 at address 5E₁₆ 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. 8 Memory assignment in SFR area (3)

Address	Register name	Access b7							Access b0	State b7							State b0
60_{16}	Watchdog timer register	✻5								? **(Note)**							
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	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

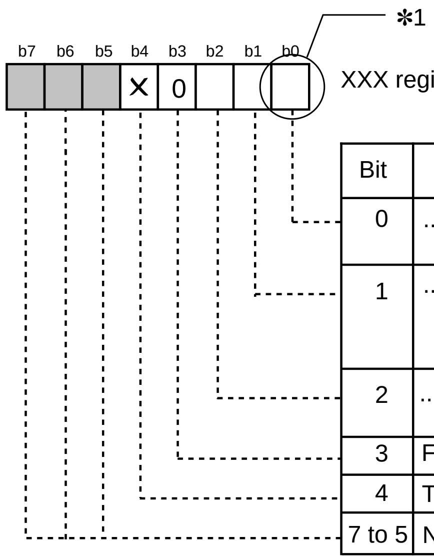
✻5 : By writing dummy data to address 60_{16}, a value "FFF_{16}" is set to the watchdog timer.
The dummy data is not retained anywhere.

Note: A value "FFF_{16}" is set to the watchdog timer. (Refer to "**Chapter 9. WATCHDOG TIMER**.")

Fig. 9 Memory assignment in SFR area (4)

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 meet the purpose.
- 0 : Set to “0” at writing.
- 1 : Set to “1” at writing.
- × : This bit is not used in the specific mode or state. It may be either “0” or “1.”
- : 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 data.
- RO : It is possible to read the bit state at reading. The written value becomes invalid. Accordingly, the written value may be either “0” or “1.”
- WO : The written value becomes valid data. It is impossible to read the bit state. The value is undefined at reading. However, the bit with the commentaries of “The value is “0” at reading” in the functions column or the notes is always “0” at reading.(See ✻4 above.)
- — : It is impossible to read the bit state. The value is undefined at reading. However, the bit with the commentaries of “The value is “0” at reading” in the functions column or the notes 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 (they may be either "0" or "1") 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

Notes 1: Bits 7 to 4 of the port P3 direction register cannot be written (they may be either "0" or "1") and are fixed to "0" at reading.
2: In the memory expansion mode or the microprocessor mode, fix bits 0 and 1 of the port P4 direction register to "0."

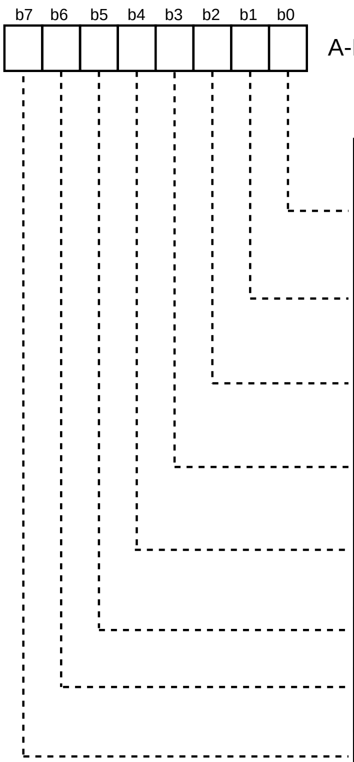
7703 Group

Fix the following bits which do not have the corresponding pin to "1."

- Bit 3 of port P3 direction register
- Bits 3 to 6 of port P4 direction register
- Bits 0, 1, 6, and 7 of port P6 direction register
- Bits 3 to 6 of port P7 direction register
- Bits 4 and 5 of port P8 direction register

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

A-D control register

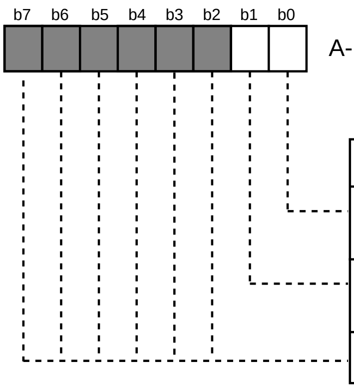


A-D control register (Address $1E_{16}$)

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	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode	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 : f_2 divided by 4 1 : f_2 divided by 2	0	RW

Notes 1: These bits are ignored in the single sweep and repeat sweep mode. (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: Writing to each bit (except bit 6) of the A-D control register must be performed while the A-D converter halts.

A-D sweep pin select register

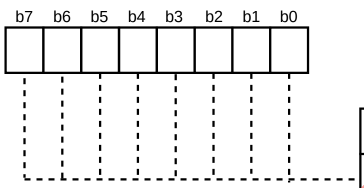


A-D sweep pin select register (Address $1F_{16}$)

Bit	Bit name	Functions	At reset	RW
0	A-D sweep pin select bits (Valid in single sweep and repeat sweep mode) **(Note 1)**	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			1	RW
7 to 2	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: Writing to each bit of the A-D sweep pin select register must be performed while the A-D converter halts.

A-D register i



A-D register 0 (Addresses 20_{16})
A-D register 1 (Addresses 22_{16})
A-D register 2 (Addresses 24_{16})
A-D register 3 (Addresses 26_{16})
A-D register 4 (Addresses 28_{16})
A-D register 5 (Addresses $2A_{16}$)
A-D register 6 (Addresses $2C_{16}$)
A-D register 7 (Addresses $2E_{16}$)

Bit	Functions	At reset	RW
7 to 0	Reads an A-D conversion result.	Undefined	RO

UARTi transmit/receive mode register

b7 b6 b5 b4 b3 b2 b1 b0

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."

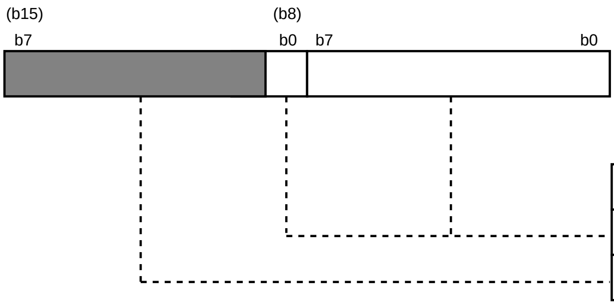
UARTi baud rate register (BRGi)

b7 b0

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

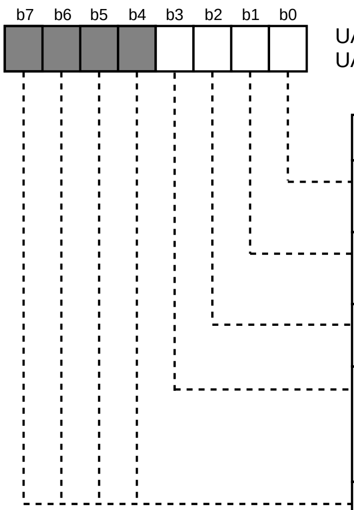
UARTi transmit buffer register



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	–

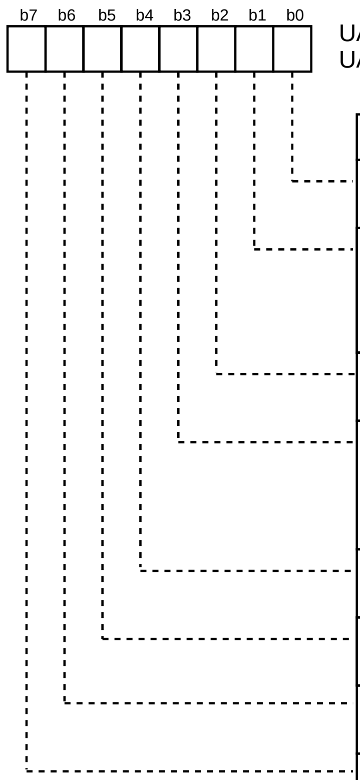
UARTi transmit/receive control register 0



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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	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
7 to 4	Nothing is assigned.		Undefined	–

UARTi transmit/receive control register 1

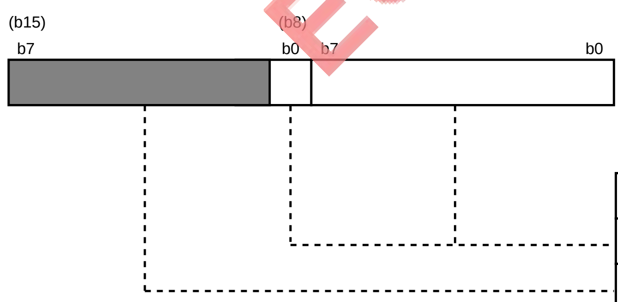


UART0 transmit/receive control register 1 (Address 35_{16})
UART1 transmit/receive control register 1 (Address $3D_{16}$)

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: Bits 7 to 4 are cleared to "0" when clearing the receive enable bit to "0" or when reading the low-order byte of the UARTi receive buffer register (addresses 36_{16}, $3E_{16}$) out.
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

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

One-shot start register

b7 b6 b5 b4 b3 b2 b1 b0

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

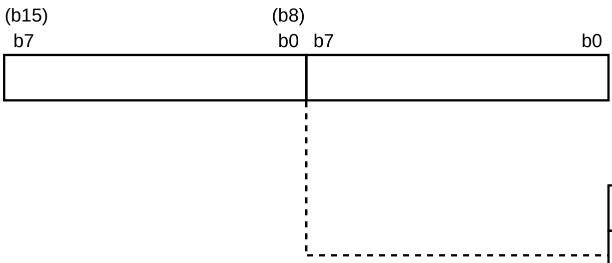
b7 b6 b5 b4 b3 b2 b1 b0

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, 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.

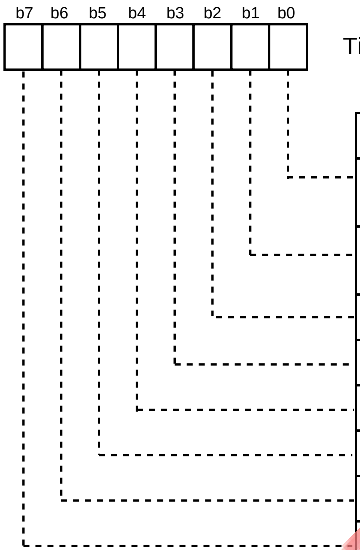
Timer Ai register



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

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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	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 counter divides the count source frequency by n + 1. When reading, the register indicates the counter value.	Undefined	RW

Event counter mode

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

One-shot pulse mode

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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	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 “0001_{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 : n / fi.	Undefined	WO

fi: Frequency of count source (f_2, f_{16}, f_{64}, or f_{512})

Pulse width modulation (PWM) mode

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 : } $_{IN}$ 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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	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_{16}, f_{64}, or f_{512})

<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_{16}, f_{64}, or f_{512})

Timer Bi register

(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 have different functions according to the operating mode.	Undefined	RW

Timer Bi mode register

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.

Timer mode

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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	0	RW
7			0	RW

X : It may be either "0" or "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	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

Event counter mode

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X				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 bits	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

X : It may be either "0" or "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	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

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 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 0 1 : f_{16} 1 0 : f_{64} 1 1 : f_{512}	0	RW
7			0	RW

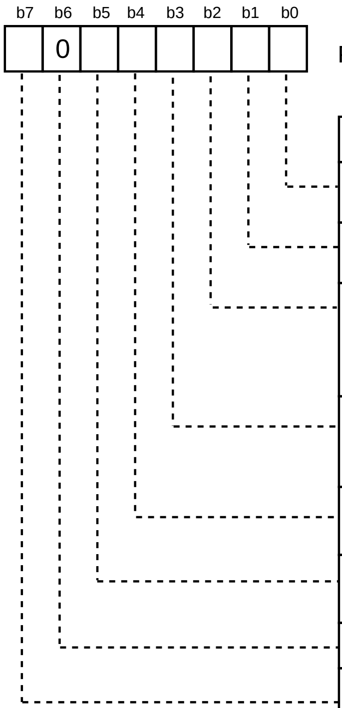
Note: The timer Bi overflow flag is cleared to "0" by writing to the timer Bi mode register with the count start bit = "1".

(b15) (b8)
b7 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

Processor mode register

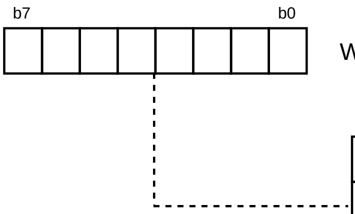


Processor mode register (Address $5E_{16}$)

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	Wait bit	0 : Software Wait is inserted when accessing external area. 1 : No software Wait is inserted when accessing external area.	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” after a reset. (Fixed to “1.”)
2: This bit is ignored in the microprocessor mode. (It may be either “0” or “1.”)

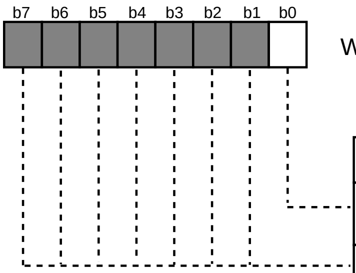
Watchdog timer register



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 : f_{512} 1 : f_{32}	0	RW
7 to 1	Nothing is assigned.		Undefined	–

Interrupt control register

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	–

Note: Use the **SEB** or **CLB** instruction to set each interrupt control register.

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 1)**	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	–

Notes 1: The $\overline{INT_0}$ to $\overline{INT_2}$ interrupt request bits are invalid when selecting the level sense.
2: Use the **SEB** or **CLB** instruction to set the $\overline{INT_0}$ to $\overline{INT_2}$ interrupt interrupt control registers.

APPENDIX

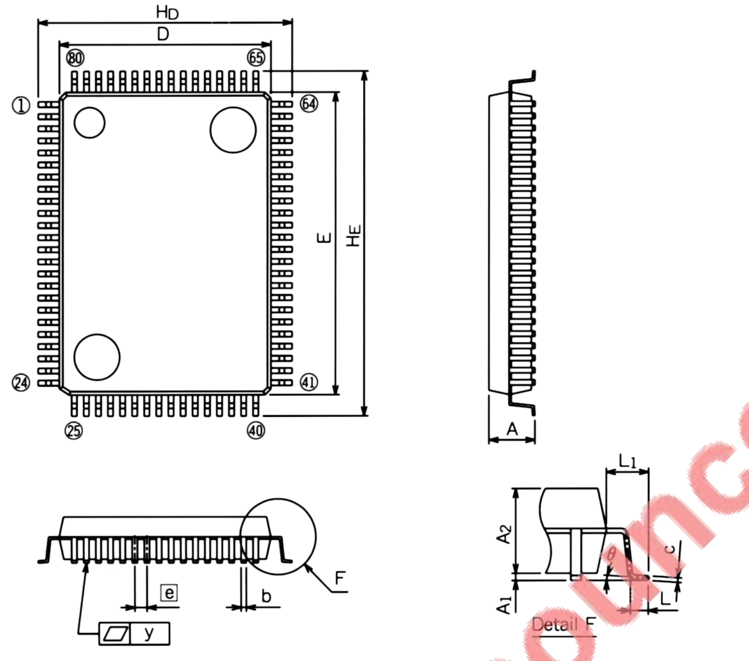
Appendix 4. Package outlines

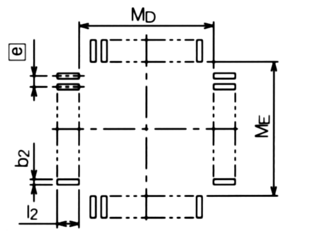
Appendix 4. Package outlines

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 : 2/1





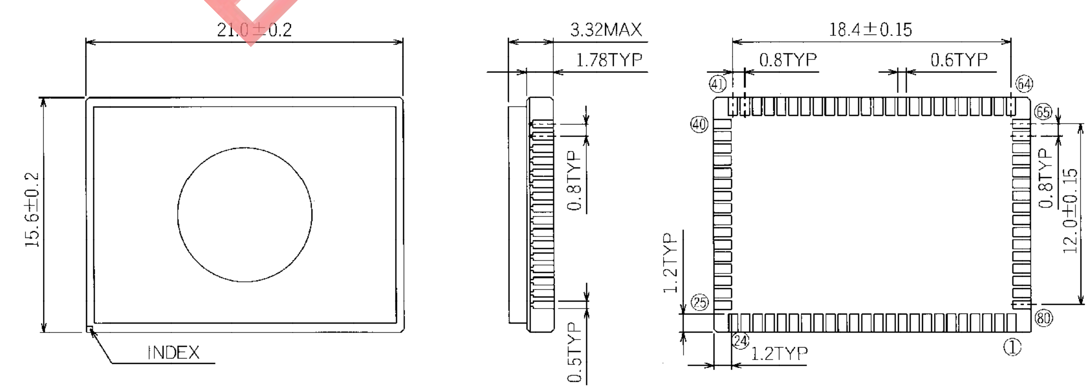
Recommended Mount Pad

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 : 2/1

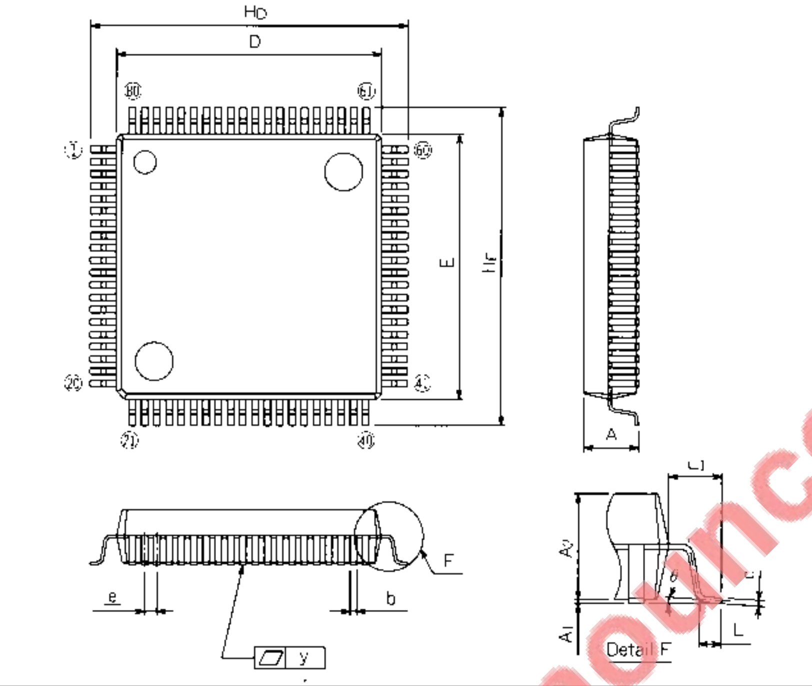


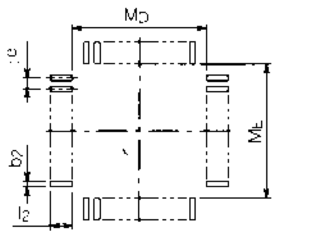
80P6S-A

Plastic 80pin 14×14mm body QFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP80-P-1414-0.65		1.11	Alloy 42

Scale :





Recommended Mount Pad

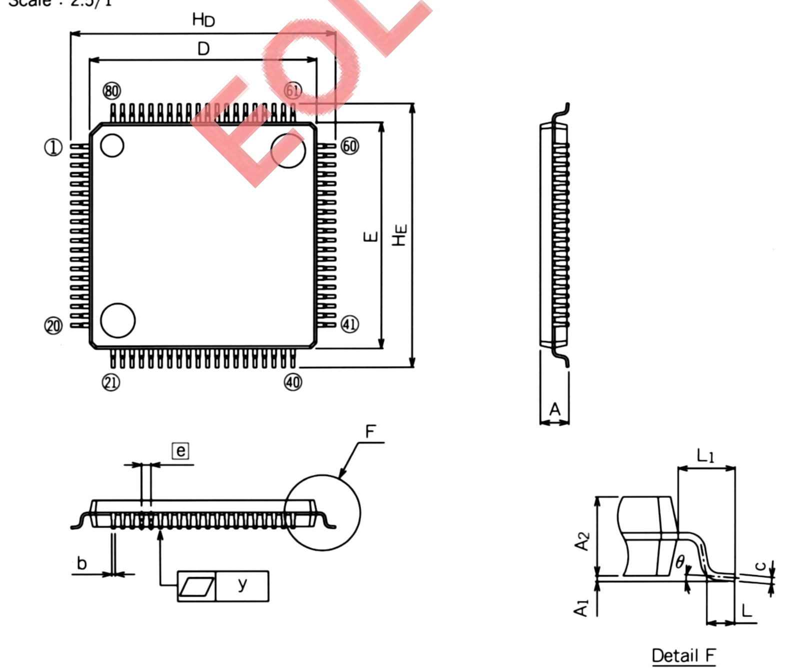
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–		3.05
A_1	0	0.1	0.2
A_2	–	2.8	–
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	13.8	14.0	14.2
e		0.65	
H_D	16.5	16.8	17.1
H_E	16.5	16.8	17.1
L	0.4	0.6	0.8
L_1		1.4	
y	–		0.1
θ	0	–	10
b_2	–	0.35	–
l_2	1.3		
M_D	–	14.6	–
M_E	–	14.6	–

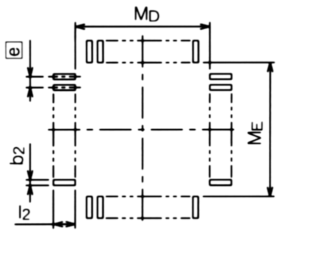
80P6D-A

Plastic 80pin 12×12mm body LQFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP80-P-1212-0.50	–	0.44	Alloy 42

Scale : 2.5/1





Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	1.7
A_1	0	0.1	0.2
A_2	–	1.4	–
b	0.13	0.18	0.28
c	0.105	0.125	0.175
D	11.9	12.0	12.1
E	11.9	12.0	12.1
e	–	0.5	–
H_D	13.8	14.0	14.2
H_E	13.8	14.0	14.2
L	0.3	0.5	0.7
L_1	–	1.0	–
y	–	–	0.1
θ	0°	–	10°
b_2	–	0.225	–
l_2	1.0	–	–
M_D	–	12.4	–
M_E	–	12.4	–

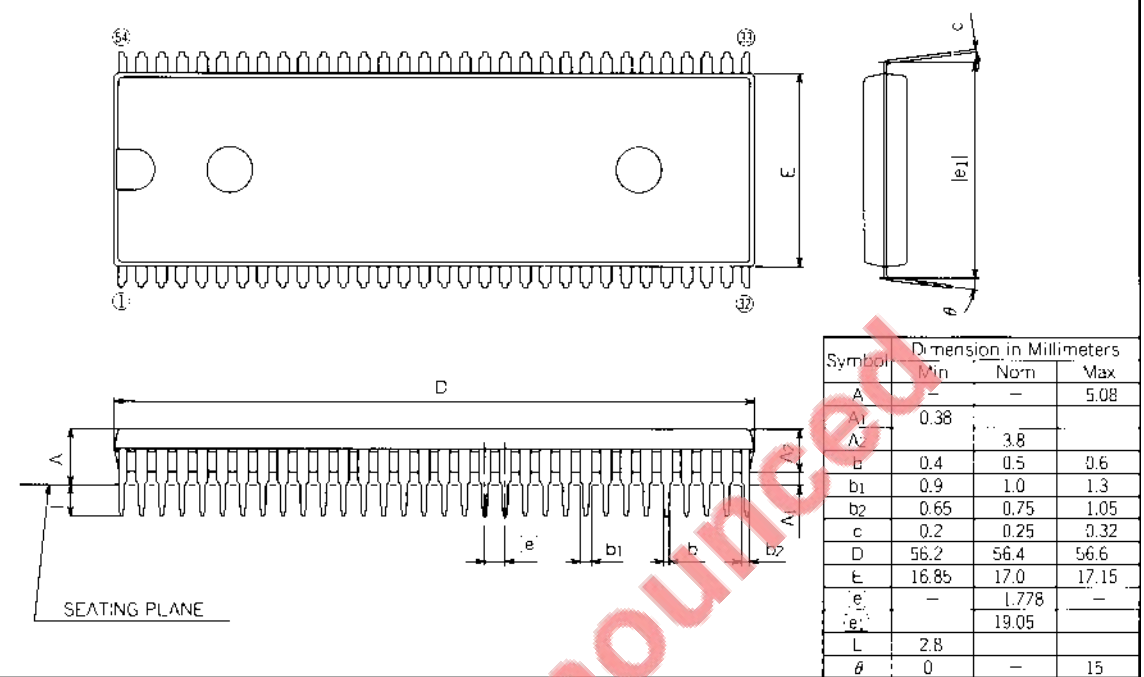
APPENDIX

Appendix 4. Package outlines

64P4B Plastic 64pin 750mil SDIP

EIAJ Package Code	JEDEC Code	Weight (g)	Lead Material
SDIP64-P-750-1.78	–	7.9	Alloy 42

Scale :



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	5.08
A_1	0.38		
A_2		3.8	
b	0.4	0.5	0.6
b_1	0.9	1.0	1.3
b_2	0.65	0.75	1.05
c	0.2	0.25	0.32
D	56.2	56.4	56.6
E	16.85	17.0	17.15
e	–	1.778	–
e_1		19.05	
L	2.8		
θ	0	–	15

Appendix 5. Countermeasures against noise

The following describes some examples of countermeasures against noise.
Although the effect depends on the system, refer to the following if the problem being relevant to noise occurs.

1. Reduction in wiring length

Wiring on a circuit board can serve as an antenna that pulls in noise into the microcomputer. Shorter the total length of wiring (in mm), the smaller the possibility of pulling in noise into the microcomputer.

(1) Wiring of $\overline{\text{RESET}}$ pin

Reduce the length of wiring connected to the $\overline{\text{RESET}}$ pin.
Especially, a capacitor that is inserted between the $\overline{\text{RESET}}$ and Vss pins must be connected to these pins in the shortest possible distance (within 20 mm).

Reasons: If noise gets into the $\overline{\text{RESET}}$ pin, the microcomputer will restart operating before its internal state is completely initialized, which can cause a program runaway.

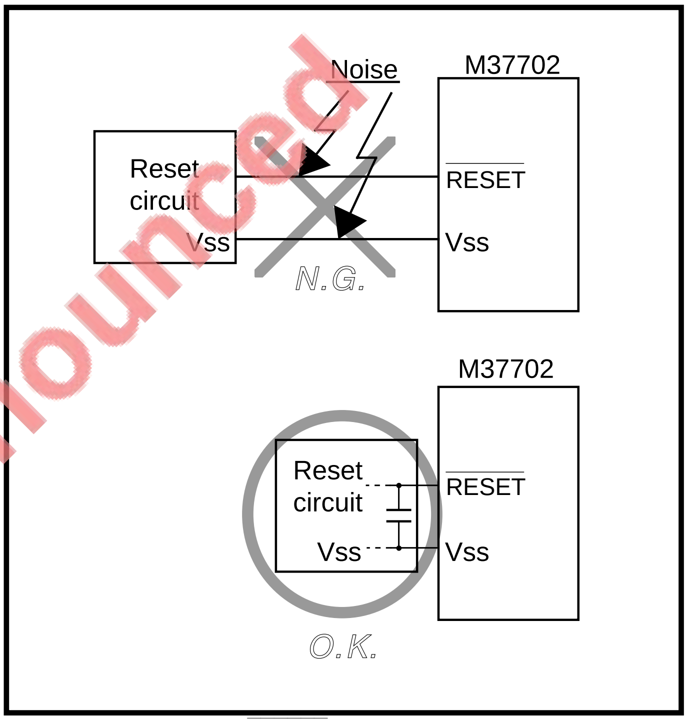


Fig. 10 Wiring of $\overline{\text{RESET}}$ pin

Appendix 5. Countermeasures against noise

(2) Wiring of clock input/output pins

- Reduce the length of wiring connected to the clock input/output pins.
- Connect the lead wire on the ground side of a capacitor connected to the oscillator and the microcomputer's Vss pin in the shortest possible distance (within 20 mm).
- Separate the Vss pattern for oscillation purpose from the other Vss patterns. (Refer to Figure 19.)

Reasons: The microcomputer operates synchronously with the clock generated by the oscillation circuit. If noise gets into the clock input/output pins, the clock waveform is disturbed, which can cause the microcomputer to malfunction or a program runaway.
Furthermore, if noise causes a potential difference between the microcomputer's Vss level and the oscillator's Vss level, the oscillator cannot generate an exact clock.

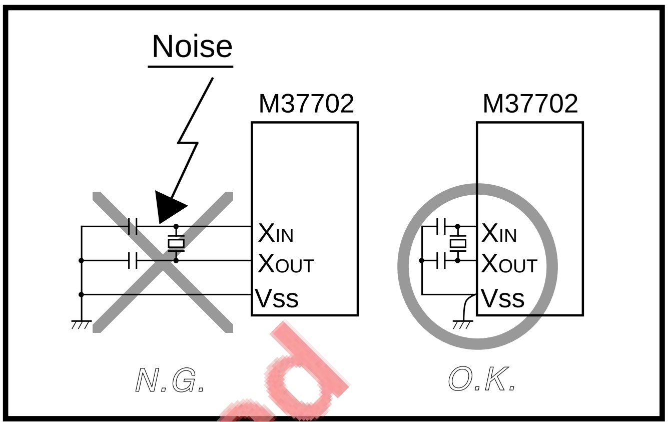


Fig. 11 Wiring of clock input/output pins

(3) Wiring of CNVss pin

When connecting the CNVss and Vss pins, connect them in the shortest possible distance.

Reasons: The voltage level on the CNVss pin affects the selection of microcomputer's processor modes. If noise causes a potential difference between the voltage levels of the CNVss and Vss pins when these pins are connected, the microcomputer's processor mode will become unstable, causing the microcomputer to malfunction or a program runaway.

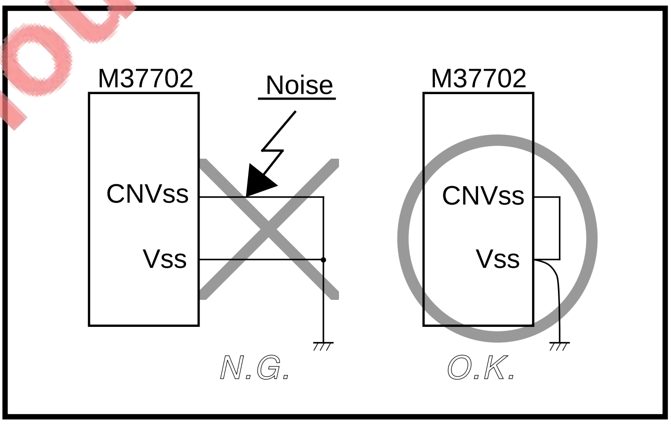


Fig. 12 Wiring of CNVss pin

(4) Wiring of CNVss (V_{PP}) pin of built-in PROM version

< In single-chip or memory expansion mode>

- Connect the CNVss (V_{PP}) pin to the microcomputer's Vss pin in the shortest possible distance.
- If the wiring cannot be shortened, insert a resistor of about 5 kohms as close to the CNVss (V_{PP}) pin as possible. By way of this resistor, connect the CNVss (V_{PP}) pin to the Vss pin.

< In microprocessor mode>

- Connect the CNVss (V_{PP}) and Vcc pins in the shortest possible distance.

Reasons: The CNVss (V_{PP}) pin serves as a power source input pin for the built-in PROM, and this pin has a reduced impedance to allow a programming current to flow in when programming to the built-in PROM. (This means that noise gets in easily.)
If noise gets into the CNVss (V_{PP}) pin, abnormal instruction codes or data will be read out from the built-in PROM, causing a program runaway.

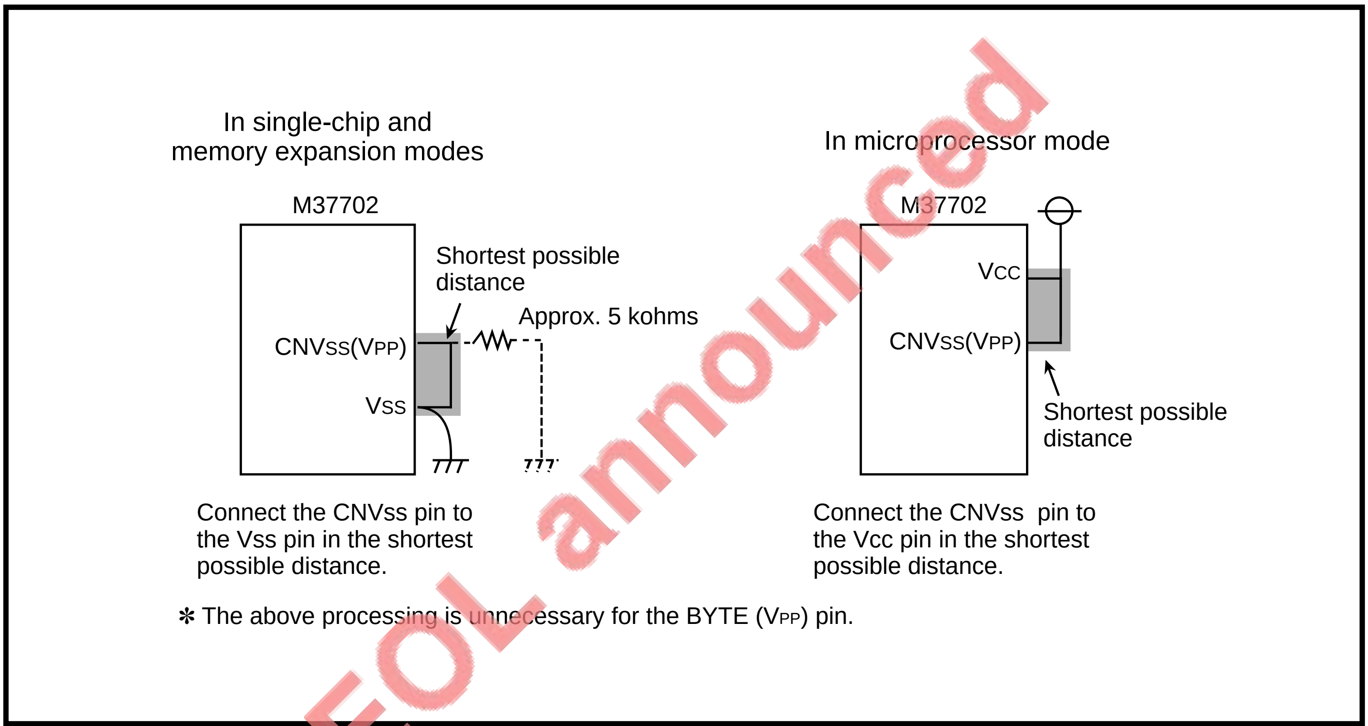


Fig. 13 Wiring of CNVss (V_{PP}) pin of built-in PROM version

Appendix 5. Countermeasures against noise

2. Inserting bypass capacitor between Vss and Vcc lines

Insert a bypass capacitor of about 0.1 μF between the Vss and Vcc lines. When inserting this bypass capacitor, make sure that the following conditions are satisfied.

- Wiring length between the Vss pin and the bypass capacitor equals that between the Vcc pin and the bypass capacitor.
- Wiring between the Vss pin and the bypass capacitor and that between the Vcc pin and the bypass capacitor have the shortest possible length.
- The Vss and Vcc lines both have broader wiring width than the other signal wires.

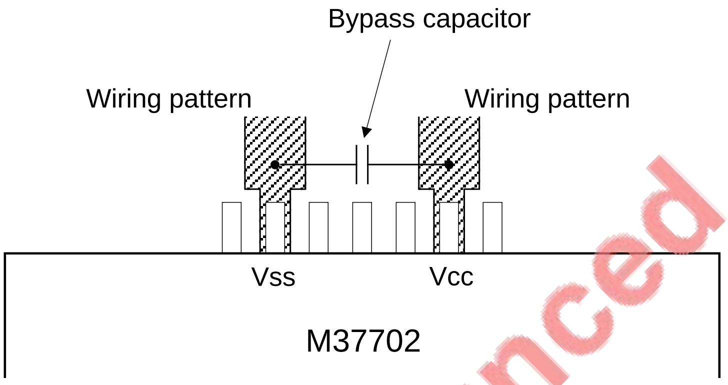


Fig. 14 Bypass capacitor between Vss and Vcc lines

3. Wiring processing of analog input pin, analog power source pin and others

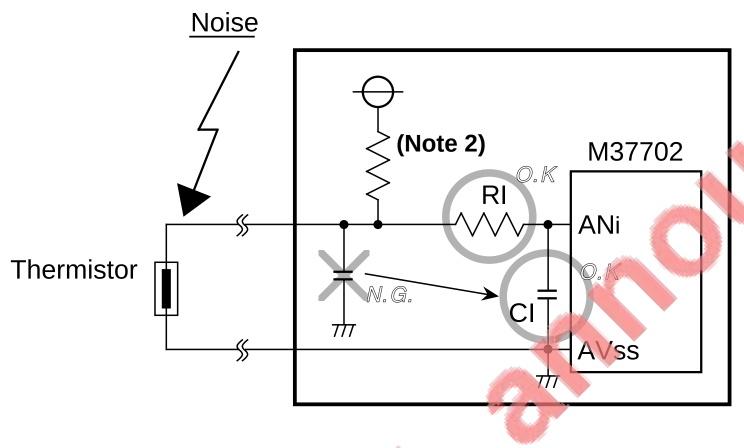
(1) Processing of analog input pin

- Connect a resistor in series to the analog signal wire connecting to an analog input pin at the position closest possible to the microcomputer.
- Insert a capacitor between the analog input pin and AVss pin at a position closest possible to the AVss pin.

Reasons: Normally, the signal which is input to the analog input pin is an output signal from a sensor.

A sensor used to detect changes in event is in many cases located away from the board on which the microcomputer is mounted. Accordingly, wiring from the sensor to the analog input pin inevitably becomes long. This long wiring can serve as an antenna that pulls in noise into the microcomputer, letting noise get into the analog input pin easily.

Additionally, if the capacitor between the analog input pin and AVss pin is grounded away from the AVss pin, noise on that ground can get into the microcomputer via the capacitor.



Reference value
RI : Approximately 100 to 1000 Ω
CI : Approximately 100 to 1000 pF

Notes 1 : Make sure that the external circuit of the ANi pin is designed so that the ANi pin can be charged/discharged within 1 cycle of ϕ_{AD}.

2 : This resistor is used to divide resistance from the thermistor.

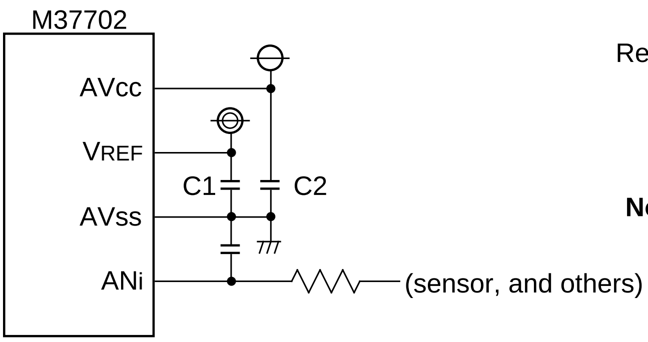
Fig. 15 Example for protecting analog input pin against noise by using thermistor

Appendix 5. Countermeasures against noise

(2) Processing of analog power source pins and others

- For each of the Vcc, AVcc, and V_{REF} pins, use separated power sources.
- Insert capacitors between the AVcc and AVss pin, and between the V_{REF} and AVss pin, respectively.

Reasons: Avoids affecting the A-D converter due to noise on Vcc.



Reference value

C1 ≥ 0.47 µF

C2 ≥ 0.47 µF

Note : Connect capacitors with the thickest possible wiring in the shortest possible distance.

Fig. 16 Processing of analog power source pin and others

4. Consideration to oscillator

The oscillator that generates the fundamental clock of the microcomputer's operation requires careful consideration not to be affected by the other signals.

(1) Isolation from signal wires where a large current flows

The signal wires where a large current exceeding the microcomputer's current limits accepted flows must be located as far away from the microcomputer (especially the oscillator) as possible.

Reasons: A system using a microcomputer contains signal wires to control, for example, motors, LEDs, and thermal heads. When a large current flows in these signal wires, noise due to mutual inductance is generated.

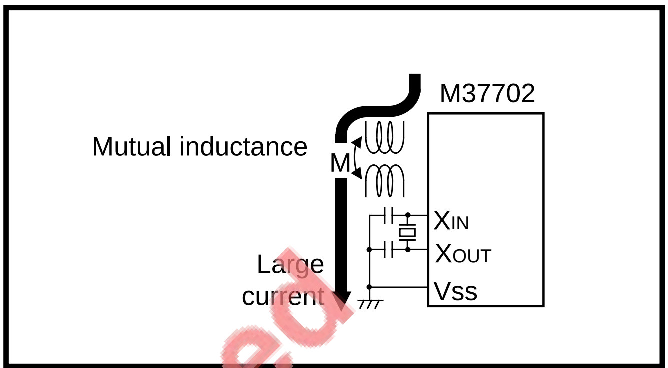


Fig. 17 Connection of signal wires where a large current flows

(2) Isolation from signal wires whose levels change rapidly

- The signal wires whose levels change rapidly must be located as far away from the oscillator as possible.
- Make sure that signal wires whose levels change rapidly do not cross any other clock-related or noise-susceptible signal wires.

Reasons: The signal wires whose voltage levels change rapidly tend to affect other signal wires as the signal level changes from high to low or from low to high. Especially if these signal wires cross a clock-related signal wire, they can disturb the clock waveform, causing the microcomputer to malfunction or a program runaway.

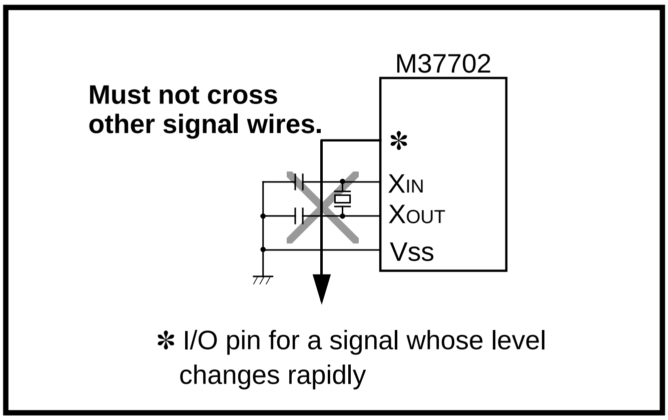


Fig. 18 Wiring of rapidly level changing signal wire

Appendix 5. Countermeasures against noise

(3) Protection with Vss pattern

For double-sided boards in which the oscillator is mounted on one side (mount side), make sure that there is a Vss pattern at the same position as the oscillator on the reverse side (solder side) of the board. This Vss pattern must be connected to the microcomputer's Vss pin in the shortest possible distance and must be located away from the other Vss patterns.

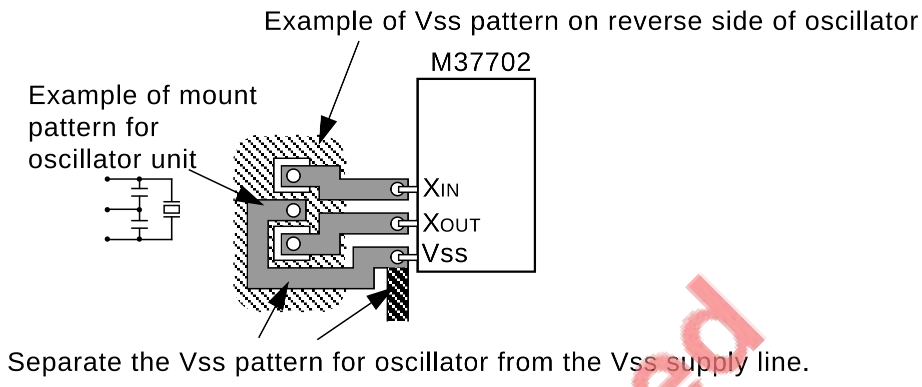


Fig. 19 Vss pattern on reverse side of oscillator

5. Processing of ports

Take protective measures for ports in both hardware and software.

<Hardware protection>

● Insert a resistor of 100 ohms or more in series.

<Software protection>

● For ports in the input mode, try reading in several times to detect whether their levels are matched or not.

● For ports in the output mode, since the output data can reverse owing to noise, periodically set the port Pi register.

● Set the port Pi direction register again at stated periods.

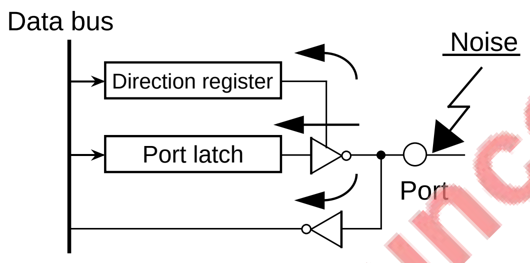


Fig. 20 Processing of ports

6. Reinforcement of the power supply line

- Use the broader wiring width than that of the other signal wire for the Vss and Vcc lines.
- When using the multilayer boards, make sure that one of the middle layer is Vss side, and the other one of middle layer is Vcc side.
- When using the double-sided boards, one side must be located with looped or mesh form to the Vss line centering the microcomputer. The vacant space must be filled with the Vss line. The other side must be located with the Vcc line just as in the above-mentioned Vss line.
  Connect the power supply line of external devices connected to the microcomputer with the bus and the power supply line of the microcomputer in the shortest possible distance.

Reasons: The level of many wiring among 24 pieces of external address bus will change at the same time when connecting external devices. That may causes noise of the power supply line.

Appendix 6. Q & A

Information which may be helpful in fully utilizing the 7702 Group and the 7703 Group are provided in Q & A format.

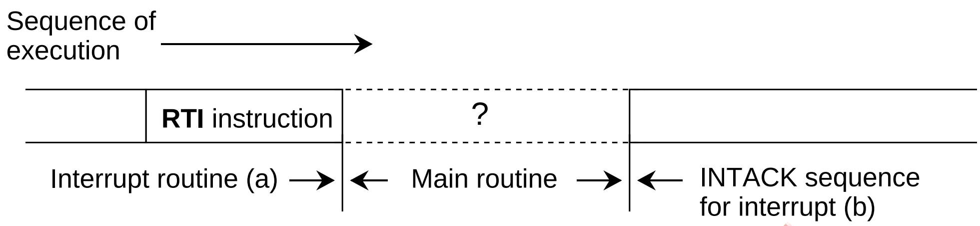
In Q & A, as a rule, one question and its answer are summarized within one page. The upper box on each page is a question, and a box below the question is its answer. (If a question or an answer extends to two or more pages, there is a page number at the lower right corner.)

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?



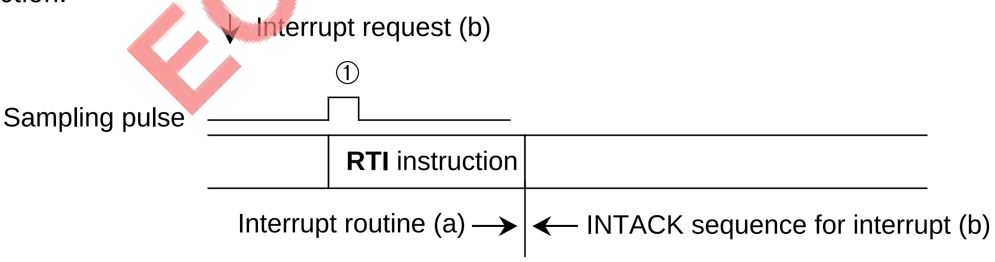
Condition

- 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.

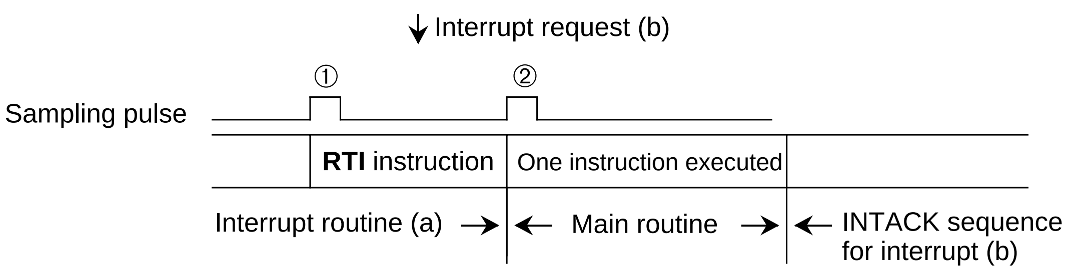
A

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?

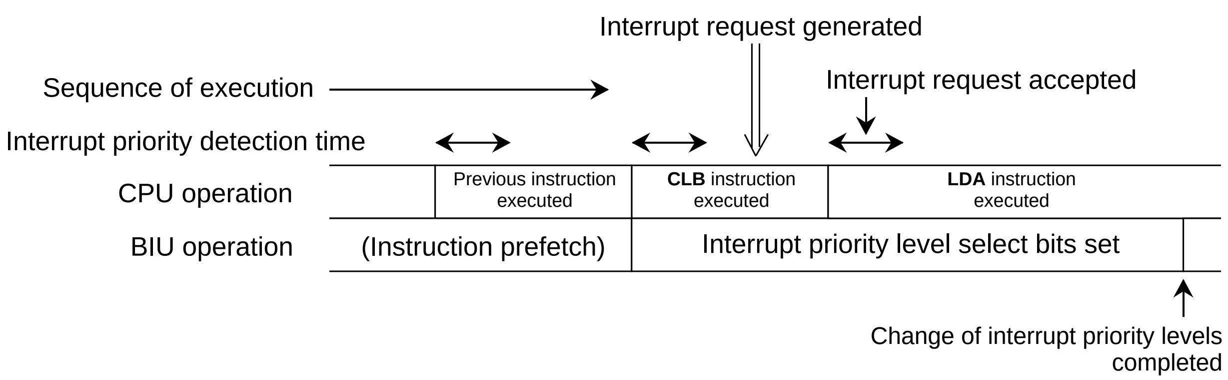
```
                              :
Interrupt request is     CLB #07H, XXXIC   ; Writes "0002" to interrupt priority level select bits.
accepted in this  →                        ; Clears interrupt request bit to "0."
interval                 LDA A,DATA        ; Instruction at the beginning of the routine that
                                             should not accept one certain interrupt.
                              :            ;
```

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)

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.

```
    :
CLB #07H, 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

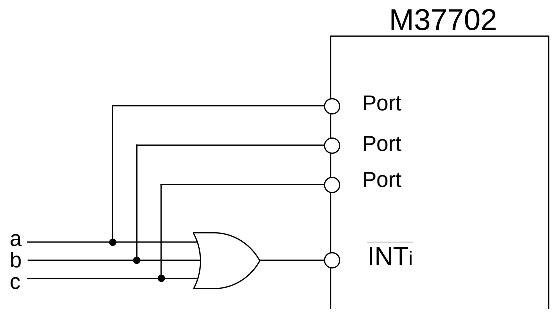
(1) Which timing of clock ϕ_1 is the external interrupts (input signals to the $\overline{INT_i}$ pin) detected?
(2) How can four or more external interrupt input pins ($\overline{INT_i}$) be used?

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 ports' 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.

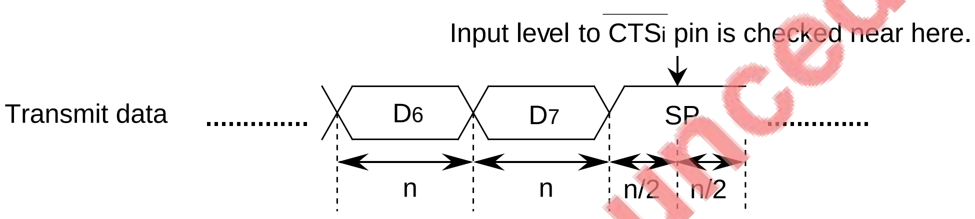
Serial I/O (UART mode)

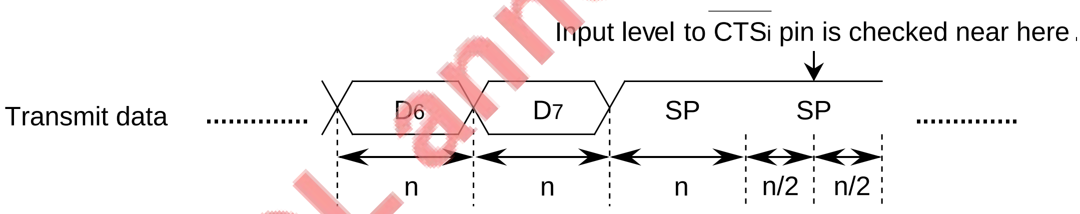
Q

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).





n: 1-bit length

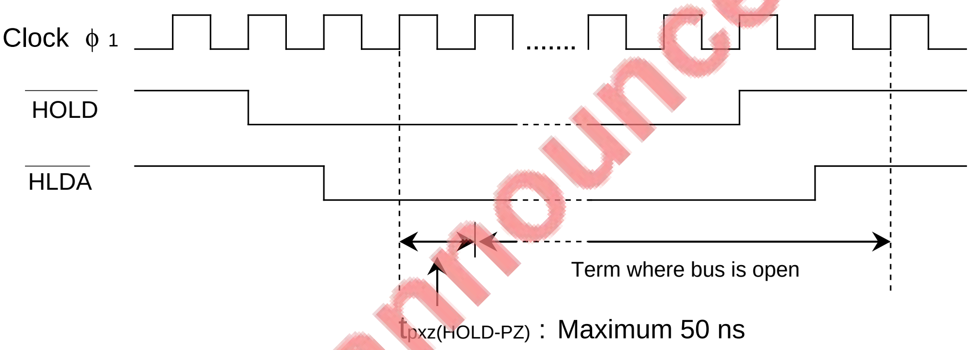
Hold function

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.



Note: The 7703 Group does not have the $\overline{\text{HLDA}}$ pin.

Q

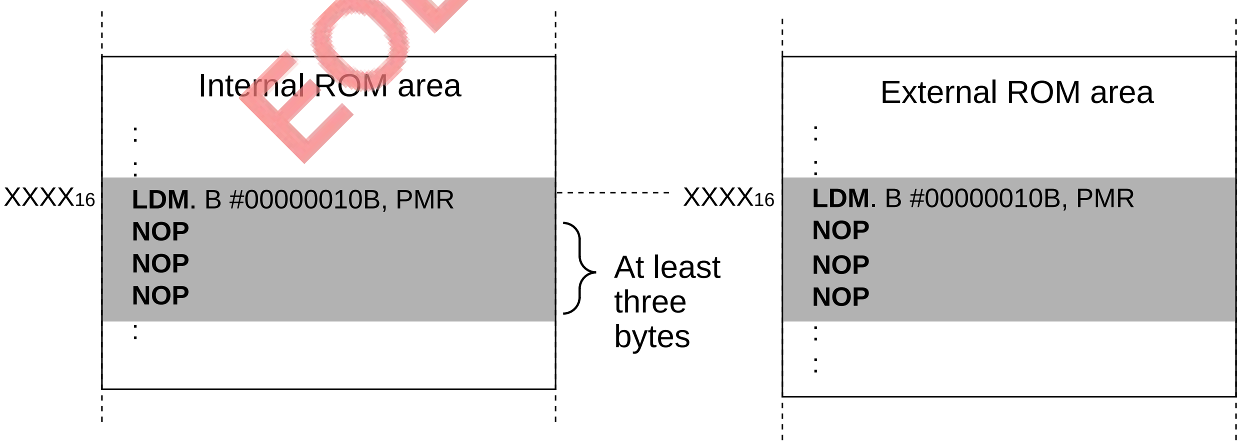
If the processor mode is switched as described below by using the processor mode bits (bits 1 and 0 at address $5E_{16}$) during program execution, is there any precaution in software?

- 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 <u>the address next to the address (address $XXXX_{16}$)</u> 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, <u>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</u>. 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

(1) 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})

(2) Use the **SEB** and **CLB** instructions to set interrupt control registers (addresses $7F_{16}$ to 70_{16}).

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 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, it is necessary that you 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, store once the 16-bit immediate value to the stack area and next return 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.

Appendix 7. Hexadecimal instruction code table

INSTRUCTION CODE TABLE-1

$D_3 \sim D_0$		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
$D_7 \sim D_4$	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

Note 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 third word's codes, refer to the INSTRUCTION CODE TABLE-2.

APPENDIX

Appendix 7. Hexadecimal instruction code table

INSTRUCTION CODE TABLE-2 (The first word's code of each instruction is 42_{16})

$D_7 \sim D_4$	Hexadecimal notation \ $D_3 \sim D_0$	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		ORA B,(DIR,X)		ORA B,SR		ORA B,DIR		ORA B,L(DIR)		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			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			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

INSTRUCTION CODE TABLE-3 (The first word's code of each instruction is 89_{16})

$D_7 \sim D_4$	$D_3 \sim D_0$ / 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		MPY L(DIR)		MPY IMM				MPY ABS		MPY ABL
0001	1		MPY (DIR),Y	MPY (DIR)	MPY (SR),Y		MPY DIR,X		MPY L(DIR),Y		MPY ABS,Y				MPY 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																
1001	9																
1010	A																
1011	B																
1100	C			LDT IMM													
1101	D																
1110	E																
1111	F																

APPENDIX

Appendix 8. Machine instructions

Appendix 8. Machine instructions

MACHINE INSTRUCTIONS

Symbol	Function	Details	IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
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.				69 42 69	2 4	2 3				65 42 65	4 6	2 3				75 42 75	5 7	2 3				72 42 72	6 8	2 3	61 42 61	7 9	2 3	71 42 71	8 10	2 3
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.				29 42 29	2 4	2 3				25 42 25	4 6	2 3				35 42 35	5 7	2 3				32 42 32	6 8	2 3	21 42 21	7 9	2 3	31 42 31	8 10	2 3
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.							0A 42 0A	2 4	1 2	06	7	2				16	7	2												
BBC (Note 3,5)	Mb=0 ?	Tests the specified bit of the memory. Branches when all the contents of the specified bit is "0".																														
BBS (Note 3,5)	Mb=1 ?	Tests the specified bit of the memory. Branches when all the contents of the specified bit is "1".																														
BCC (Note 3)	C=0 ?	Branches when the contents of the C flag is "0".																														
BCS (Note 3)	C=1 ?	Branches when the contents of the C flag is "1".																														
BEQ (Note 3)	Z=1 ?	Branches when the contents of the Z flag is "1".																														
BMI (Note 3)	N=1 ?	Branches when the contents of the N flag is "1".																														
BNE (Note 3)	Z=0 ?	Branches when the contents of the Z flag is "0".																														
BPL (Note 3)	N=0 ?	Branches when the contents of the N flag is "0".																														
BRA (Note 4)	PC ← PC±offset PG ← PG+1 (carry occured) PG ← PG−1 (borrow occured)	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) ← PC_H S ← S−1 M(S) ← PC_L S ← S−1 M(S) ← PS_H S ← S−1 M(S) ← PS_L S ← S−1 I ← 1 PC_L ← AD_L PC_H ← AD_H PG ← 00_{16}	Executes software interruption.	00	15	2																											
BVC (Note 3)	V=0 ?	Branches when the contents of the V flag is "0".																														
BVS (Note 3)	V=1 ?	Branches when the contents of the V flag is "1".																														
CLB (Note 5)	Mb ← 0	Makes the contents of the specified bit in the memory "0".													14	8	3															
CLC	C ← 0	Makes the contents of the C flag "0".	18	2	1																											
CLI	I ← 0	Makes the contents of the I flag "0".	58	2	1																											
CLM	*m* ← 0	Makes the contents of the *m* flag "0".	D8	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	2																								
CLV	V ← 0	Makes the contents of the V flag "0".	B8	2	1																											
CMP (Note 1,2)	Acc−M	Compares the contents of the accumulator with the contents of the memory.				C9 42 C9	2 4	2 3				C5 42 C5	4 6	2 3				D5 42 D5	5 7	2 3				D2 42 D2	6 8	2 3	C1 42 C1	7 9	2 3	D1 42 D1	8 10	2 3

Addressing mode																																																						Processor status register										
L(DIR)			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			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
67	10	2	77	11	2	6D	4	3				7D	6	3	79	6	3	6F	6	4	7F	7	4																						63	5	2	73	8	2				•	•	•	N	V	•	•	•	•	Z	C
42 67	12	3	42 77	13	3	42 6D	6	4				42 7D	8	4	42 79	8	4	42 6F	8	5	42 7F	9	5																						42 63	7	3	42 73	10	3														
27	10	2	37	11	2	2D	4	3				3D	6	3	39	6	3	2F	6	4	3F	7	4																						23	5	2	33	8	2				•	•	•	N	•	•	•	•	•	Z	•
42 27	12	3	42 37	13	3	42 2D	6	4				42 3D	8	4	42 39	8	4	42 2F	8	5	42 3F	9	5																						42 23	7	3	42 33	10	3														
						0E	7	3				1E	8	3																																								•	•	•	N	•	•	•	•	•	Z	C
																																							34	7	4	3C	8	5										•	•	•	•	•	•	•	•	•	•	•
																																							24	7	4	2C	8	5										•	•	•	•	•	•	•	•	•	•	•
																																				90	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				B0	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				F0	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				30	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				D0	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				10	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				80	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				82	4	3																										
																																																						•	•	•	•	•	•	•	•	•	•	•
																																				50	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				70	4	2																•	•	•	•	•	•	•	•	•	•	•
									1C	9	4																																											•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	•	•	•	•	•	•	•	0
																																																						•	•	•	•	•	•	•	•	0	•	•
																																																						•	•	•	•	•	0	•	•	•	•	•
																																																						•	•	•	Specified flag becomes "0".							
																																																						•	•	•	•	0	•	•	•	•	•	•
C7	10	2	D7	11	2	CD	4	3				DD	6	3	D9	6	3	CF	6	4	DF	7	4																						C3	5	2	D3	8	2				•	•	•	N	•	•	•	•	•	Z	C
42 C7	12	3	42 D7	13	3	42 CD	6	4				42 DD	8	4	42 D9	8	4	42 CF	8	5	42 DF	9	5																						42 C3	7	3	42 D3	10	3														

APPENDIX

Appendix 8. Machine instructions

Symbol	Function	Details	Addressing mode																													
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
CPX (Note 2)	X−M	Compares the contents of the index register X with the contents of the memory.				E0	2	2				E4	4	2																		
CPY (Note 2)	Y−M	Compares the contents of the index register Y with the contents of the memory.				C0	2	2				C4	4	2																		
DEC (Note 1)	$A_{CC} \leftarrow A_{CC}-1$ or $M \leftarrow M-1$	Decrements the contents of the accumulator or memory by 1.							1A 42 1A	2 4	1 2	C6	7	2				D6	7	2												
DEX	$X \leftarrow X-1$	Decrements the contents of the index register X by 1.	CA	2	1																											
DEY	$Y \leftarrow Y-1$	Decrements the contents of the index register Y by 1.	88	2	1																											
DIV (Note 2,10)	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	27	3				89 25	29	3				89 35	30	3				89 32	31	3	89 21	32	3	89 31	33	3
EOR (Note 1,2)	$A_{CC} \leftarrow A_{CC} \forall M$	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 42 49	2 4	2 3				45 42 45	4 6	2 3				55 42 55	5 7	2 3				52 42 52	6 8	2 3	41 42 41	7 9	2 3	51 42 51	8 10	2 3
INC (Note 1)	$A_{CC} \leftarrow A_{CC}+1$ or $M \leftarrow M+1$	Increments the contents of the accumulator or memory by 1.							3A 42 3A	2 4	1 2	E6	7	2				F6	7	2												
INX	$X \leftarrow X+1$	Increments the contents of the index register X by 1.	E8	2	1																											
INY	$Y \leftarrow Y+1$	Increments the contents of the index register Y by 1.	C8	2	1																											
JMP	ABS $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ ABL $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ $PG \leftarrow AD_G$ (ABS) $PC_L \leftarrow (AD_H, AD_L)$ $PC_H \leftarrow (AD_H, AD_L+1)$ L(ABS) $PC_L \leftarrow (AD_H, AD_L)$ $PC_H \leftarrow (AD_H, AD_L+1)$ $PG \leftarrow (AD_H, AD_L+2)$ (ABS, X) $PC_L \leftarrow (AD_H, AD_L+X)$ $PC_H \leftarrow (AD_H, AD_L+X+1)$	Places a new address into the program counter and jumps to that new address.																														
JSR	ABS $M(S) \leftarrow PC_H$ $S \leftarrow S-1$ $M(S) \leftarrow PC_L$ $S \leftarrow S-1$ $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ ABL $M(S) \leftarrow PG$ $S \leftarrow S-1$ $M(S) \leftarrow PC_H$ $S \leftarrow S-1$ $M(S) \leftarrow PC_L$ $S \leftarrow S-1$ $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ $PG \leftarrow AD_G$ (ABS, X) $M(S) \leftarrow PC_H$ $S \leftarrow S-1$ $M(S) \leftarrow PC_L$ $S \leftarrow S-1$ $PC_L \leftarrow (AD_H, AD_L+X)$ $PC_H \leftarrow (AD_H, AD_L+X+1)$	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.																														

Addressing mode																										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)		
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
						EC	4	3																		
						CC	4	3																		
						CE	7	3				DE	8	3												
89 27	35	3	89 37	36	3	89 2D	29	4				89 3D	31	4	89 39	31	4	89 2F	31	5	89 3F	32	5			
47	10	2	57	11	2	4D	4	3				5D	6	3	59	6	3	4F	6	4	5F	7	4			
42 47	12	3	42 57	13	3	42 4D	6	4				42 5D	8	4	42 59	8	4	42 4F	8	5	42 5F	9	5			
						EE	7	3				FE	8	3												
						4C	2	3										5C	4	4				6C	4	3
						20	6	3										22	8	4						

Addressing mode (continued)																										
L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK		
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
																		89 23	30	3	89 33	33	3			
																		43	5	2	53	8	2			
																		42 43	7	3	42 53	10	3			
DC	8	3	7C	6	3																					
			FC	8	3																					

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	C
•	•	•	N	•	•	•	•	•	Z	C
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	V	•	•	•	•	Z	C
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•

APPENDIX

Appendix 8. Machine instructions

Symbol	Function	Details	Addressing mode																													
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
LDA (Note 1,2)	A_{CC} ← M	Enters the contents of the memory into the accumulator.				A9 42 A9	2 4	2 3				A5 42 A5	4 6	2 3				B5 42 B5	5 7	2 3				B2 42 B2	6 8	2 3	A1 42 A1	7 9	2 3	B1 42 B1	8 10	2 3
LDM (Note 5)	M ← IMM	Enters the immediate value into the memory.										64	4	3				74	5	3												
LDT	DT ← IMM	Enters the immediate value into the data bank register.				89 C2	5	3																								
LDX (Note 2)	X ← M	Enters the contents of the memory into index register X.				A2	2	2				A6	4	2							B6	5	2									
LDY (Note 2)	Y ← M	Enters the contents of the memory into index register Y.				A0	2	2				A4	4	2				B4	5	2												
LSR (Note 1)	m=0 0→[b_{15} … b_0]→C m=1 0→[b_7 … b_0]→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 42 4A	2 4	1 2	46	7	2				56	7	2												
MPY (Note 2,11)	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	16	3				89 05	18	3				89 15	19	3				89 12	20	3	89 01	21	3	89 11	22	3
MVN (Note 8)	Mn+i←Mm+i	Transmits the data block. The transmission is done from the lower order address of the block.																														
MVP (Note 9)	Mn−i←Mm−i	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	1																											
ORA (Note 1,2)	A_{CC}←A_{CC}∨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 42 09	2 4	2 3				05 42 05	4 6	2 3				15 42 15	5 7	2 3				12 42 12	6 8	2 3	01 42 01	7 9	2 3	11 42 11	8 10	2 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.																														

Addressing mode

L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X		
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
A7	10	2	B7	11	2	AD	4	3				BD	6	3	B9	6	3	AF	6	4	BF	7	4
42 A7	12	3	42 B7	13	3	42 AD	6	4				42 BD	8	4	42 B9	8	4	42 AF	8	5	42 BF	9	5
						9C	5	4				9E	6	4									
						AE	4	3							BE	6	3						
						AC	4	3				BC	6	3									
						4E	7	3				5E	8	3									
89 07	24	3	89 17	25	3	89 0D	18	4				89 1D	20	4	89 19	20	4	89 0F	20	5	89 1F	21	5
07	10	2	17	11	2	0D	4	3				1D	6	3	19	6	3	0F	6	4	1F	7	4
42 07	12	3	42 17	13	3	42 0D	6	4				42 1D	8	4	42 19	8	4	42 0F	8	5	42 1F	9	5

(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK		
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
																					A3	5	2	B3	8	2			
																					42 A3	7	3	42 B3	10	3			
																					89 03	19	3	89 13	22	3			
																											54	7 + ½×7	3
																											44	9 + ½×7	3
																					03	5	2	13	8	2			
																					42 03	7	3	42 13	10	3			
									F4	5	3																		
									D4	6	2																		
									62	5	3																		
									48	4	1																		
									42 48	6	2																		

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	•
•	•	•	0	•	•	•	•	•	Z	C
•	•	•	N	•	•	•	•	•	Z	0
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	N	•	•	•	•	•	Z	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•

APPENDIX

Appendix 8. Machine instructions

Symbol	Function	Details	Addressing mode																													
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
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.																														
PHX	x=0 M(S)←X_H S←S−1 M(S)←X_L S←S−1 x=1 M(S)←X_L S←S−1	Saves the contents of the index register X into the stack.																														
PHY	x=0 M(S)←Y_H S←S−1 M(S)←Y_L S←S−1 x=1 M(S)←Y_L S←S−1	Saves the contents of the index register Y into the stack.																														
PLA	m=0 S←S+1 A_L←M(S) S←S+1 A_H←M(S) m=1 S←S+1 A_L←M(S)	Restores the contents of the stack on the accumulator A.																														
PLB	m=0 S←S+1 B_L←M(S) S←S+1 B_H←M(S) m=1 S←S+1 B_L←M(S)	Restores the contents of the stack on the accumulator B.																														
PLD	S←S+1 DPR_L←M(S) S←S+1 DPR_H←M(S)	Restores the contents of the stack on the direct page register.																														
PLP	S←S+1 PS_L←M(S) S←S+1 PS_H←M(S)	Restores the contents of the stack on the processor status register.																														
PLT	S←S+1 DT←M(S)	Restores the contents of the stack on the data bank register.																														
PLX	x=0 S←S+1 X_L←M(S) S←S+1 X_H←M(S) x=1 S←S+1 X_L←M(S)	Restores the contents of the stack on the index register X.																														

Addressing mode																																																						Processor status register										
L(DIR)			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			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
																																	0B	4	1																			•	•	•	•	•	•	•	•	•	•	•
																																	4B	3	1																			•	•	•	•	•	•	•	•	•	•	•
																																	08	4	1																			•	•	•	•	•	•	•	•	•	•	•
																																	8B	3	1																			•	•	•	•	•	•	•	•	•	•	•
																																	0A	4	1																			•	•	•	•	•	•	•	•	•	•	•
																																	5A	4	1																			•	•	•	•	•	•	•	•	•	•	•
																																	68	5	1																			•	•	•	N	•	•	•	•	•	Z	•
																																	42 68	7	2																			•	•	•	N	•	•	•	•	•	Z	•
																																	2B	5	1																			•	•	•	•	•	•	•	•	•	•	•
																																	28	6	1																			Value saved in stack.										
																																	AB	6	1																			•	•	•	N	•	•	•	•	•	Z	•
																																	FA	5	1																			•	•	•	N	•	•	•	•	•	Z	•

APPENDIX

Appendix 8. Machine instructions

Symbol	Function	Details	Addressing mode																													
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
PLY	x=0 S←S+1 Y_L←M(S) S←S+1 Y_H←M(S) x=1 S←S+1 Y_L←M(S)	Restores the contents of the stack on the index register Y.																														
PSH (Note 6)	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 7)	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 13)	m=0 n bit rotate left [b15 ··· b0] m=1 n bit rotate left [b7 ··· b0]	Rotates the contents of the accumulator A, n bits to the left.				89 49	6 + i	3																								
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 4	1 2	26	7	2				36	7	2												
ROR (Note 1)	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 4	1 2	66	7	2				76	7	2												
RTI	S←S+1 PS_L←M(S) S←S+1 PS_H←M(S) S←S+1 PC_L←M(S) S←S+1 PC_H←M(S) S←S+1 PG←M(S)	Returns from the interruption routine.	40	11	1																											
RTL	S←S+1 PC_L←M(S) S←S+1 PC_H←M(S) S←S+1 PG←M(S)	Returns from the subroutine. The contents of the program bank register are also restored.	6B	8	1																											
RTS	S←S+1 PC_L←M(S) S←S+1 PC_H←M(S)	Returns from the subroutine. The contents of the program bank register are not restored.	60	5	1																											
SBC (Note 1,2)	A_{CC}, C←A_{CC}−M−$\overline{C}$	Subtracts the contents of the memory and the borrow from the contents of the accumulator.				E9 42 E9	2 4	2 3				E5 42 E5	4 6	2 3				F5 42 F5	5 7	2 3				F2 42 F2	6 8	2 3	E1 42 E1	7 9	2 3	F1 42 F1	8 10	2 3

Addressing mode

L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X		
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
						2E	7	3				3E	8	3									
						6E	7	3				7E	8	3									
E7	10	2	F7	11	2	ED	4	3				FD	6	3	F9	6	3	EF	6	4	FF	7	4
42 E7	12	3	42 F7	13	3	42 ED	6	4				42 FD	8	4	42 F9	8	4	42 EF	8	5	42 FF	9	5

(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK		
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
									7A	5	1																		
									EB	12 + $2i_1 + i_2$	2																		
									FB	14 + $3i_1 + 4i_2$	2																		
																					E3	5	2	F3	8	2			
																					42 E3	7	3	42 F3	10	3			

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	•
•	•	•	•	•	•	•	•	•	•	•
If restored the contents of PS, it becomes its value. And the other case is no change.										
•	•	•	•	•	•	•	•	•	•	•
•	•	•	N	•	•	•	•	•	Z	C
•	•	•	N	•	•	•	•	•	Z	C
Value saved in stack.										
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
•	•	•	N	V	•	•	•	•	Z	C

APPENDIX

Appendix 8. Machine instructions

Symbol	Function	Details	Addressing mode																													
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
SEB (Note 5)	Mb←1	Makes the contents of the specified bit in the memory "1".													04	8	3															
SEC	C←1	Makes the contents of the C flag "1".	38	2	1																											
SEI	I←1	Makes the contents of the I flag "1".	78	2	1																											
SEM	m←1	Makes the contents of the m flag "1".	F8	2	1																											
SEP	PSb←1	Set the specified bit of the processor status register's lower byte (PS_L) to "1".				E2	3	2																								
STA (Note 1)	M←A_{CC}	Stores the contents of the accumulator into the memory.										85 42 85	4 6	2 3				95 42 95	5 7	2 3				92 42 92	7 9	2 3	81 42 81	7 9	2 3	91 42 91	7 9	2 3
STP		Stops the oscillation of the oscillator.	DB	3	1																											
STX	M←X	Stores the contents of the index register X into the memory.										86	4	2							96	5	2									
STY	M←Y	Stores the contents of the index register Y into the memory.										84	4	2				94	5	2												
TAD	DPR←A	Transmits the contents of the accumulator A to the direct page register.	5B	2	1																											
TAS	S←A	Transmits the contents of the accumulator A to the stack pointer.	1B	2	1																											
TAX	X←A	Transmits the contents of the accumulator A to the index register X.	AA	2	1																											
TAY	Y←A	Transmits the contents of the accumulator A to the index register Y.	A8	2	1																											
TBD	DPR←B	Transmits the contents of the accumulator B to the direct page register.	42 5B	4	2																											
TBS	S←B	Transmits the contents of the accumulator B to the stack pointer.	42 1B	4	2																											
TBX	X←B	Transmits the contents of the accumulator B to the index register X.	42 AA	4	2																											
TBY	Y←B	Transmits the contents of the accumulator B to the index register Y.	42 A8	4	2																											
TDA	A←DPR	Transmits the contents of the direct page register to the accumulator A.	7B	2	1																											
TDB	B←DPR	Transmits the contents of the direct page register to the accumulator B.	42 7B	4	2																											
TSA	A←S	Transmits the contents of the stack pointer to the accumulator A.	3B	2	1																											
TSB	B←S	Transmits the contents of the stack pointer to the accumulator B.	42 3B	4	2																											
TSX	X←S	Transmits the contents of the stack pointer to the index register X.	BA	2	1																											
TXA	A←X	Transmits the contents of the index register X to the accumulator A.	8A	2	1																											
TXB	B←X	Transmits the contents of the index register X to the accumulator B.	42 8A	4	2																											
TXS	S←X	Transmits the contents of the index register X to the stack pointer.	9A	2	1																											
TXY	Y←X	Transmits the contents of the index register X to the index register Y.	9B	2	1																											
TYA	A←Y	Transmits the contents of the index register Y to the accumulator A.	98	2	1																											
TYB	B←Y	Transmits the contents of the index register Y to the accumulator B.	42 98	4	2																											
TYX	X←Y	Transmits the contents of the index register Y to the index register X.	BB	2	1																											
WIT		Stops the internal clock.	CB	3	1																											
XAB	A⇄B	Exchanges the contents of the accumulator A and the contents of the accumulator B.	89 28	6	2																											

Addressing mode																																																						Processor status register										
L(DIR)			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			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
									0C	9	4																																											•	•	•	•	•	•	•	•	•	•	•
																																																				•	•	•	•	•	•	•	•	•	•	1
																																																				•	•	•	•	•	•	•	•	1	•	•
																																																				•	•	•	•	•	1	•	•	•	•	•
																																																				•	•	•	Specified flag becomes "1".							
87	10	2	97	11	2	8D	5	3				9D	5	3	99	5	3	8F	6	4	9F	7	4																						83	5	2	93	8	2				•	•	•	•	•	•	•	•	•	•	•
42 87	12	3	42 97	13	3	42 8D	7	4				42 9D	7	4	42 99	7	4	42 8F	8	5	42 9F	9	5																						42 83	7	3	42 93	10	3														
																																																					•	•	•	•	•	•	•	•	•	•	•
						8E	5	3																																														•	•	•	•	•	•	•	•	•	•	•
						8C	5	3																																														•	•	•	•	•	•	•	•	•	•	•
																																																				•	•	•	•	•	•	•	•	•	•	•
																																																				•	•	•	•	•	•	•	•	•	•	•
																																																				•	•	•	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	•

APPENDIX

Appendix 8. Machine instructions

Symbols in machine instructions table

Symbol	Description
IMP	Implied addressing mode
IMM	Immediate addressing mode
A	Accumulator addressing mode
DIR	Direct addressing mode
DIR, b	Direct bit addressing mode
DIR, X	Direct indexed X addressing mode
DIR, Y	Direct indexed Y addressing mode
(DIR)	Direct indirect addressing mode
(DIR, X)	Direct indexed X indirect addressing mode
(DIR), Y	Direct indirect indexed Y addressing mode
L (DIR)	Direct indirect long addressing mode
L (DIR), Y	Direct indirect long indexed Y addressing mode
ABS	Absolute addressing mode
ABS, b	Absolute bit addressing mode
ABS, X	Absolute indexed X addressing mode
ABS, Y	Absolute indexed Y addressing mode
ABL	Absolute long addressing mode
ABL, X	Absolute long indexed X addressing mode
(ABS)	Absolute indirect addressing mode
L (ABS)	Absolute indirect long addressing mode
(ABS, X)	Absolute indexed X indirect addressing mode
STK	Stack addressing mode
REL	Relative addressing mode
DIR, b, REL	Direct bit relative addressing mode
ABS, b, REL	Absolute bit relative addressing mode
SR	Stack pointer relative addressing mode
(SR), Y	Stack pointer relative indirect indexed Y addressing mode
BLK	Block transfer addressing mode
C	Carry flag
Z	Zero flag
I	Interrupt disable flag
D	Decimal operation mode flag
x	Index register length selection flag
m	Data length selection flag
V	Overflow flag
N	Negative flag
IPL	Processor interrupt priority level
+	Addition
−	Subtraction
*	Multiplication
/	Division
$\wedge$	Logical AND
$\vee$	Logical OR

Symbol	Description
$\veebar$	Exclusive OR
$\overline{\ \ }$	Negation
←	Movement to the arrow direction
A_{CC}	Accumulator
A_{CCH}	Accumulator's upper 8 bits
A_{CCL}	Accumulator's lower 8 bits
A	Accumulator A
A_H	Accumulator A's upper 8 bits
A_L	Accumulator A's lower 8 bits
B	Accumulator B
B_H	Accumulator B's upper 8 bits
B_L	Accumulator B's lower 8 bits
X	Index register X
X_H	Index register X's upper 8 bits
X_L	Index register X's lower 8 bits
Y	Index register Y
Y_H	Index register Y's upper 8 bits
Y_L	Index register Y's lower 8 bits
S	Stack pointer
PC	Program counter
PC_H	Program counter's upper 8 bits
PC_L	Program counter's lower 8 bits
PG	Program bank register
DT	Data bank register
DPR	Direct page register
DPR_H	Direct page register's upper 8 bits
DPR_L	Direct page register's lower 8 bits
PS	Processor status register
PS_H	Processor status register's upper 8 bits
PS_L	Processor status register's lower 8 bits
PS_b	Processor status register's b-th bit
M(S)	Contents of memory at address indicated by stack pointer
M_b	b-th memory location
AD_G	Value of 24-bit address's upper 8-bit (A_{23}~A_{16})
AD_H	Value of 24-bit address's middle 8-bit (A_{15}~A_8)
AD_L	Value of 24-bit address's lower 8-bit (A_7~A_0)
op	Operation code
n	Number of cycle
#	Number of byte
i	Number of transfer byte or rotation
i_1, i_2	Number of registers pushed or pulled

The number of cycles shown in the table is described in case of the fastest mode for each instruction. The number of cycles shown in the table is calculated for $DPR_L=0$. The number of cycles in the addressing mode concerning the DPR when $DPR_L \neq 0$ must be 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 read/write address is odd or even. It also differs when the external region memory is accessed by BYTE="H".

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 number of cycles increments by 2 when branching.

Note 4. The operation code on the upper row is used for branching in the range of −128~+127, and the operation code on the lower row is used for branching in the range of −32768~+32767.

Note 5. When handling 16-bit data with flag m=0, the byte in the table is incremented by 1.

Note 6.

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 12. i_1 indicates the number of registers among A, B, X, Y, DPR, and PS to be saved, while i_2 indicates the number of registers among DT and PG to be saved.

Note 7.

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 14. i_1 indicates the number of registers among A, B, X, Y, DT, and PS to be restored, while $i_2=1$ when DPR is to be restored.

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;

$$7+(i/2)\times 7+4$$

Note that, (i/2) shows the integer part when i is divided by 2.

Note 9. 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;

$$9+(i/2)\times 7+5$$

Note that, (i/2) shows the integer part when i is divided by 2.

Note 10. The number of cycles is the case in the 16-bit÷8-bit operation. The number of cycles is incremented by 16 for 32-bit÷16-bit operation.

Note 11. The number of cycles is the case in the 8-bit×8-bit operation. The number of cycles is incremented by 8 for 16-bit ×16-bit operation.

Note 12. 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 13. When flag m is 0, the byte in the table is incremented by 1.

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 area	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_{16}, f_{64}, f_{512} selected by the count source select bits and others.	
Counter contents (values)	Means a value read when reading the timer Ai and Bi registers.	
Down-count	Means decreasing by 1 and counting.	Up-count
Event counter mode	Means the mode of Timers which can count the number of external pulses exactly without a divider.	
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.	
Gate function of Timer	Means the function that the user can control input of the timer count source.	
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 table.	
LSB first	Means a transfer data format of Serial I/O;LSB is transferred first.	
Overflow	A state where the up-count resultant is greater than the counter resolution.	Under flow Up-count
Power saving	Means reducing a power dissipation by Stop mode, Wait mode or others	Stop mode Wait mode
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**, **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
Synchronizing clock	Means a transfer clock of the clock synchronous serial I/O.	

Term	Meaning	Relevant term
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
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
7702/7703 Group

Mar. First Edition 1997

Editioned by
Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

User's Manual
7702/7703 Group

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